SIEMENS

# ICs for Communications

Extended Line Card Interface Controller
ELIC®

PEB 20550
PEF 20550

Versions 1.3

User's Manual 01.96

T2055-0V13-M1-7600

**Edition 01.96**

This edition was realized using the software system FrameMaker®.

**Published by Siemens AG, Bereich Halbleiter, Marketing-Kommunikation, Balanstraße 73, 81541 München**

© Siemens AG 1996.
All Rights Reserved.

**Attention please!**

As far as patents or other rights of third parties are concerned, liability is only assumed for components, not for applications, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery and prices please contact the Semiconductor Group Offices in Germany or the Siemens Companies and Representatives worldwide (see address list).

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Siemens Office, Semiconductor Group.

Siemens AG is an approved CECC manufacturer.

**Packing**

Please use the recycling operators known to you. We can also help you – get in touch with your nearest sales office. By agreement we will take packing material back, if it is sorted. You must bear the costs of transport.

For packing material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

**Components used in life-support devices or systems must be expressly authorized for such purpose!**

Critical components[1] of the Semiconductor Group of Siemens AG, may only be used in life-support devices or systems[2] with the express written approval of the Semiconductor Group of Siemens AG.

1 A critical component is a component used in a life-support device or system whose failure can reasonably be expected to cause the failure of that life-support device or system, or to affect its safety or effectiveness of that device or system.

2 Life support devices or systems are intended (a) to be implanted in the human body, or (b) to support and/or maintain and sustain human life. If they fail, it is reasonable to assume that the health of the user may be endangered.

| **PEB 20550**<br>**PEF 20550**<br>**Revision History:** | | **User's Manual 01.96** |
|---|---|---|
| Previous Release: | | Technical Manual 9.93 |
| Page (in Previous Release) | Page (in User's Manual) | Subjects (major changes since last revision) |
| – | 13 | PEF 20550 (ext. temperature range; new) |
| – | 38 | System Integration and Application (DECT added) |
| 29 | 46 | Boundary scan number 22 = 110 (correction) |
| 29 | 46 | Boundary scan number 9: ID code for V1.3 added |
| 31 | 49 | Boundary scan ID code for V1.3 added |
| – | 57 | DMA-transfers, figure 31 (new) |
| – | 60 | Support of the HDLC protocol by SACCO, figure 35 (new) |
| 51 | 76 | SACCO clock mode 2 description (extended) |
| 53 | 80 | Extensions for V1.3 |
| 55 | 82 | Arbiter state machine description (extended) |
| 58 | 85 | Table 14: Control channel delay examples (extended) |
| 65 | 95 | Internal reference clock RCL replaced by CFI reference clock CRCL |
| – | 101 | Interrupt driven transmission sequence example, figure 50 (new) |
| 82 | 114 | Internal reference clock RCL replaced by CFI reference clock CRCL |
| 85 | 118 | Register address arrangement (extended) |
| – | 129 | EMOD: ECMD2 restriction 5 (new) |
| 93 | 130 | PMOD: PMD1..0 description (data rate stepping corrected) |
| 101 | 140 | CMD2: CXF, CRR description (corrected) |
| 104 | 144 | MACR description (extended) |
| 114 | 154 | TIMR: SSR (correction) |
| 121 | 162 | VNSR: VN3..0 = V1.2 (correction) |
| 124 | 167 | EXIR: XMR description (extended) |
| 128 | 172 | CCR1: ODS description (extended for V1.3) |
| 132 | 177 | SACCO RSTA: C/R description (new) |
| 140 | 185 | VSTR: VN3..0 value for V1.3 added |
| 142 | 187 | SCV: SCV7...0 description (extended) |
| – | 191 | Application Hints (new) |
| 148 | 380 | $t_{ALS\ min}$ = 8 ns, $t_{DRH\ max}$ = 65 ns, $t_{AH\ min}$ = 0 ns (correction) |
| – | 395 | Package outlines (new) |
| – | 396 | Appendix (new) |

**Table of Contents** **Page**


| | | |
|---|---|---|
| **1** | **Overview** | 11 |
| 1.1 | Features | 13 |
| 1.2 | Pin Configuration (top view) | 15 |
| 1.3 | Pin Definitions and Functions | 16 |
| 1.4 | Logic Symbol | 23 |
| 1.5 | Functional Block Diagram | 24 |
| 1.6 | System Integration and Application | 25 |
| 1.6.1 | Digital Line Card | 25 |
| 1.6.1.1 | Switching, Layer-1 Control, Group Controller Signaling | 25 |
| 1.6.1.2 | Decentralized D-Channel Processing, Multiplexed HDLC-Controller. | 27 |
| 1.6.1.3 | Decentralized D-Channel Processing, Dedicated HDLC-Controller per Subscriber | 31 |
| 1.6.1.4 | Decentralized D-Channel Processing, Multiplexed plus Dedicated HDLC-Control | 31 |
| 1.6.1.5 | Central D-Channel Processing | 33 |
| 1.6.1.6 | Mixed D-Channel Processing, Signaling Decentralized, Packet Data Centralized | 34 |
| 1.6.2 | Key Systems | 36 |
| 1.6.3 | Analog Line Card | 37 |
| 1.6.4 | DECT Applications | 38 |
| 1.6.4.1 | Adaptation of a DECT System to an Existing PBX | 38 |
| 1.6.4.2 | DECT Line Card Design for an Existing PBX | 40 |
| **2** | **Functional Description** | 41 |
| 2.1 | General Functions and Device Architecture | 41 |
| 2.2 | Functional Blocks | 41 |
| 2.2.1 | Bus Interface | 41 |
| 2.2.2 | Parallel Ports | 42 |
| 2.2.3 | Watchdog Timer | 43 |
| 2.2.4 | Reset Logic | 43 |
| 2.2.5 | Boundary Scan Support | 45 |
| 2.2.5.1 | Boundary Scan | 45 |
| 2.2.5.2 | TAP-Controller | 48 |
| 2.2.6 | EPIC®-1 | 49 |
| 2.2.6.1 | PCM-Interface | 49 |
| 2.2.6.2 | Configurable Interface | 50 |
| 2.2.6.3 | Memory Structure and Switching | 50 |
| 2.2.6.4 | Pre-processed Channels, Layer-1 Support | 51 |
| 2.2.6.5 | Special Functions | 52 |
| 2.2.7 | SACCO | 52 |
| 2.2.7.1 | Block Diagram | 52 |
| 2.2.7.2 | Parallel Interface | 53 |
| 2.2.7.3 | FIFO-Structure | 57 |

**Table of Contents** **Page**


2.2.7.4 Protocol Support . . . . . 60
2.2.7.5 Special Functions . . . . . 75
2.2.7.6 Serial Interface . . . . . 76
2.2.7.7 Serial Port Configuration . . . . . 79
2.2.7.8 Test Mode . . . . . 80
2.2.8 D-Channel Arbiter . . . . . 81
2.2.8.1 Upstream Direction . . . . . 82
2.2.8.2 Downstream Direction . . . . . 86
2.2.8.3 Control Channel Delay . . . . . 87
2.2.8.4 D-Channel Arbiter Co-operating with QUAT-S Circuits . . . . . 88

**3 Operational Description** . . . . . 89
3.1 Microprocessor Interface Operation . . . . . 89
3.2 Interrupt Structure and Logic . . . . . 90
3.3 Clocking . . . . . 91
3.4 Reset . . . . . 92
3.5 EPIC®-1 Operation . . . . . 92
3.5.1 PCM-Interface . . . . . 93
3.5.2 Configurable Interface . . . . . 94
3.5.3 Switching Functions . . . . . 95
3.5.4 Special Functions . . . . . 99
3.6 SACCO-A/B . . . . . 99
3.6.1 Data Transmission in Interrupt Mode . . . . . 99
3.6.2 Data Transmission in DMA-Mode . . . . . 101
3.6.3 Data Reception in Interrupt Mode . . . . . 102
3.6.4 Data Reception in DMA-Mode . . . . . 103
3.7 D-Channel Arbiter . . . . . 104
3.7.1 SACCO-A Transmission . . . . . 104
3.7.2 SACCO-A Reception . . . . . 105
3.8 Initialization Procedure . . . . . 106
3.8.1 Hardware Reset . . . . . 106
3.8.2 EPIC®-1 Initialization . . . . . 106
3.8.2.1 Register Initialization . . . . . 106
3.8.2.2 Control Memory Reset . . . . . 106
3.8.2.3 Initialization of Pre-processed Channels . . . . . 107
3.8.2.4 Initialization of the Upstream Data Memory (DM) Tristate Field . . . . . 109
3.8.3 SACCO-Initialization . . . . . 109
3.8.4 Initialization of D-Channel Arbiter . . . . . 111
3.8.5 Activation of the PCM- and CFI-Interfaces . . . . . 112
3.8.6 Initialization Example . . . . . 113
3.8.6.1 EPIC®-1 Initialization Example . . . . . 114
3.8.6.2 SACCO-A Initialization Example . . . . . 116
3.8.6.3 D-Channel Arbiter Initialization Example . . . . . 116

**Table of Contents** **Page**


3.8.6.4 PCM- and CFI-Interface Activation Example . . . . . 117
3.8.6.5 SACCO-B Initialization Example . . . . . 117

**4 Detailed Register Description** . . . . . 118
4.1 Register Address Arrangement . . . . . 118
4.2 Interrupt Top Level . . . . . 124
4.2.1 Interrupt Status Register (ISTA) . . . . . 124
4.2.2 Mask Register (MASK) . . . . . 125
4.3 Parallel Ports . . . . . 126
4.3.1 PORT0 Data Register (PORT0) . . . . . 126
4.3.2 PORT1 Data Register (PORT1) . . . . . 126
4.3.3 Port1 Configuration Register (PCON1) . . . . . 127
4.4 Watchdog Timer . . . . . 127
4.4.1 Watchdog Control Register (WTC) . . . . . 127
4.5 ELIC® Mode Register / Version Number Register (EMOD) . . . . . 128
4.6 EPIC®-1 . . . . . 130
4.6.1 PCM-Mode Register (PMOD) . . . . . 130
4.6.2 Bit Number per PCM-Frame (PBNR) . . . . . 132
4.6.3 PCM-Offset Downstream Register (POFD) . . . . . 132
4.6.4 PCM-Offset Upstream Register (POFU) . . . . . 133
4.6.5 PCM-Clock Shift Register (PCSR) . . . . . 134
4.6.6 PCM-Input Comparison Mismatch (PICM) . . . . . 135
4.6.7 Configurable Interface Mode Register 1 (CMD1) . . . . . 136
4.6.8 Configurable Interface Mode Register 2 (CMD2) . . . . . 138
4.6.9 Configurable Interface Bit Number Register (CBNR) . . . . . 141
4.6.10 Configurable Interface Time Slot Adjustment Register (CTAR) . . . . . 141
4.6.11 Configurable Interface Bit Shift Register (CBSR) . . . . . 142
4.6.12 Configurable Interface Subchannel Register (CSCR) . . . . . 143
4.6.13 Memory Access Control Register (MACR) . . . . . 144
4.6.14 Memory Access Address Register (MAAR) . . . . . 147
4.6.15 Memory Access Data Register (MADR) . . . . . 148
4.6.16 Synchronous Transfer Data Register (STDA) . . . . . 148
4.6.17 Synchronous Transfer Data Register B (STDB) . . . . . 149
4.6.18 Synchronous Transfer Receive Address Register A (SARA) . . . . . 149
4.6.19 Synchronous Transfer Receive Address Register B (SARB) . . . . . 150
4.6.20 Synchronous Transfer Transmit Address Register A (SAXA) . . . . . 150
4.6.21 Synchronous Transfer Transmit Address Register B (SAXB) . . . . . 151
4.6.22 Synchronous Transfer Control Register (STCR) . . . . . 151
4.6.23 MF-Channel Active Indication Register (MFAIR) . . . . . 152
4.6.24 MF-Channel Subscriber Address Register (MFSAR) . . . . . 153
4.6.25 Monitor/Feature Control Channel FIFO (MFFIFO) . . . . . 153
4.6.26 Signaling FIFO (CIFIFO) . . . . . 154
4.6.27 Timer Register (TIMR) . . . . . 154

**Table of Contents** **Page**


4.6.28 Status Register EPIC®-1 (STAR_E) . . . . . 155
4.6.29 Command Register EPIC®-1 (CMDR_E) . . . . . 156
4.6.30 Interrupt Status Register EPIC®-1 (ISTA_E) . . . . . 158
4.6.31 Mask Register EPIC®-1 (MASK_E) . . . . . 159
4.6.32 Operation Mode Register (OMDR) . . . . . 160
4.6.33 Version Number Status Register (VNSR) . . . . . 162
4.7 SACCO . . . . . 163
4.7.1 Receive FIFO (RFIFO) . . . . . 163
4.7.2 Transmit FIFO (XFIFO) . . . . . 164
4.7.3 Interrupt Status Register (ISTA_A/B) . . . . . 165
4.7.4 Mask Register (MASK_A/B) . . . . . 166
4.7.5 Extended Interrupt Register (EXIR_A/B) . . . . . 166
4.7.6 Command Register (CMDR) . . . . . 168
4.7.7 Mode Register (MODE) . . . . . 170
4.7.8 Channel Configuration Register 1 (CCR1) . . . . . 171
4.7.9 Channel Configuration Register 2 (CCR2) . . . . . 173
4.7.10 Receive Length Check Register (RLCR) . . . . . 174
4.7.11 Status Register (STAR) . . . . . 175
4.7.12 Receive Status Register (RSTA) . . . . . 176
4.7.13 Receive HDLC-Control Register (RHCR) . . . . . 178
4.7.14 Transmit Address Byte 1 (XAD1) . . . . . 178
4.7.15 Transmit Address Byte 2 (XAD2) . . . . . 179
4.7.16 Receive Address Byte Low Register 1 (RAL1) . . . . . 179
4.7.17 Receive Address Byte Low Register 2 (RAL2) . . . . . 180
4.7.18 Receive Address Byte High Register 1 (RAH1) . . . . . 180
4.7.19 Receive Address Byte High Register 2 (RAH2) . . . . . 181
4.7.20 Receive Byte Count Low (RBCL) . . . . . 181
4.7.21 Receive Byte Count High (RBCH) . . . . . 182
4.7.22 Transmit Byte Count Low (XBCL) . . . . . 182
4.7.23 Transmit Byte Count High (XBCH) . . . . . 183
4.7.24 Time Slot Assignment Register Transmit (TSAX) . . . . . 183
4.7.25 Time Slot Assignment Register Receive (TSAR) . . . . . 184
4.7.26 Transmit Channel Capacity Register (XCCR) . . . . . 184
4.7.27 Receive Channel Capacity Register (RCCR) . . . . . 185
4.7.28 Version Status Register (VSTR) . . . . . 185
4.8 D-Channel Arbiter . . . . . 186
4.8.1 Arbiter Mode Register (AMO) . . . . . 186
4.8.2 Arbiter State Register (ASTATE) . . . . . 187
4.8.3 Suspend Counter Value Register (SCV) . . . . . 187
4.8.4 D-Channel Enable Register IOM-Port 0 (DCE0) . . . . . 188
4.8.5 D-Channel Enable Register IOM-Port 1 (DCE1) . . . . . 188
4.8.6 D-Channel Enable Register IOM-Port 2 (DCE2) . . . . . 188

**Table of Contents** **Page**


4.8.7 D-Channel Enable Register IOM-Port 3 (DCE3) . . . . . 189
4.8.8 Transmit D-Channel Address Register (XDC) . . . . . 189
4.8.9 Broadcast Group IOM-port 0 (BCG0) . . . . . 190
4.8.10 Broadcast Group IOM-port 1 (BCG1) . . . . . 190
4.8.11 Broadcast Group IOM-port 2 (BCG2) . . . . . 190
4.8.12 Broadcast Group IOM-port 3 (BCG3) . . . . . 190

**5 Application Hints** . . . . . 191
5.1 Introduction . . . . . 191
5.1.1 IOM® and SLD Functions . . . . . 191
5.2 Configuration of Interfaces . . . . . 198
5.2.1 PCM Interface Configuration . . . . . 198
5.2.1.1 PCM Interface Signals . . . . . 198
5.2.1.2 PCM Interface Registers . . . . . 198
5.2.1.3 PCM Interface Characteristics . . . . . 199
5.2.2 Configurable Interface Configuration . . . . . 211
5.2.2.1 CFI Interface Signals . . . . . 211
5.2.2.2 CFI Registers . . . . . 211
5.2.2.3 CFI Characteristics . . . . . 213
5.3 Data and Control Memories . . . . . 239
5.3.1 Memory Structure . . . . . 239
5.3.2 Indirect Register Access . . . . . 240
5.3.3 Memory Access Commands . . . . . 244
5.3.3.1 Access to the Data Memory Data Field . . . . . 244
5.3.3.2 Access to the Data Memory Code (Tristate) Field . . . . . 248
5.3.3.3 Access to the Control Memory Data Field . . . . . 251
5.3.3.4 Access to the Control Memory Code Field . . . . . 253
5.4 Switched Channels . . . . . 260
5.4.1 CFI - PCM Timeslot Assignment . . . . . 261
5.4.2 Subchannel Switching . . . . . 265
5.4.3 Loops . . . . . 270
5.4.3.1 CFI - CFI Loops . . . . . 270
5.4.3.2 PCM - PCM Loops . . . . . 273
5.4.4 Switching Delays . . . . . 275
5.4.4.1 Internal Procedures at the Serial Interfaces . . . . . 276
5.4.4.2 How to Determine the Delay . . . . . 279
5.4.4.3 Example: Switching of Wide Band ISDN Channels with the ELIC® . . . . . 281
5.5 Preprocessed Channels . . . . . 284
5.5.1 Initialization of Preprocessed Channels . . . . . 285
5.5.2 Control/Signaling (CS) Handler . . . . . 298
5.5.2.1 Registers used in Conjunction with the CS Handler . . . . . 299
5.5.2.2 Access to Downstream C/I and SIG Channels . . . . . 301
5.5.2.3 Access to the Upstream C/I and SIG Channels . . . . . 302

**Table of Contents** **Page**


5.5.3 Monitor/Feature Control (MF) Handler . . . . . 304
5.5.3.1 Registers used in Conjunction with the MF Handler . . . . . 306
5.5.3.2 Description of the MF Channel Commands . . . . . 311
5.6 µP Channels . . . . . 319
5.7 Synchronous Transfer Utility . . . . . 323
5.7.1 Registers Used in Conjunction with the Synchronous Transfer Utility . . . 326
5.8 Supervision Functions . . . . . 331
5.8.1 Hardware Timer . . . . . 331
5.8.2 PCM Input Comparison . . . . . 333
5.8.3 PCM Framing Supervision . . . . . 336
5.8.4 Power and Clock Supply Supervision/Chip Version . . . . . 338
5.9 Applications . . . . . 339
5.9.1 Analog IOM®-2 Line Card with SICOFI®-4 as Codec/Filter Device . . . . 339
5.9.2 IOM®-2 Trunk Line Applications . . . . . 343
5.9.2.1 PBX With Multiple ISDN Trunk Lines . . . . . 344
5.9.2.2 Small PBX . . . . . 349
5.9.3 Miscellaneous . . . . . 351
5.9.3.1 Interfacing the ELIC® to a MUSAC™ . . . . . 351
5.9.3.2 Space and Time Switch for 16 kBit/s Channels . . . . . 353

**6 Application Notes** . . . . . 355
6.1 Example of ELIC® Operation in a Digital PBX . . . . . 355
6.1.1 Introduction . . . . . 355
6.1.2 Basic Initialization . . . . . 356
6.1.2.1 EPIC® Interface Initialization . . . . . 357
6.1.2.2 SACCO-A Initialization . . . . . 357
6.1.2.3 Basic D-Channel Arbiter Initialization . . . . . 358
6.1.2.4 SACCO-B Initialization . . . . . 358
6.1.3 ELIC® CM and OCTAT-P Initialization . . . . . 359
6.1.3.1 Resetting the CM . . . . . 360
6.1.3.2 Initializing the CM . . . . . 360
6.1.3.3 CFI Activation . . . . . 361
6.1.3.4 PCM Interface Activation . . . . . 361
6.1.3.5 Deactivating the OCTAT-P . . . . . 361
6.1.4 Line Activation by Subscriber A . . . . . 362
6.1.4.1 Handling of C/I Interrupt . . . . . 362
6.1.4.2 Confirmation of Line Activation . . . . . 363
6.1.4.3 Enabling the Arbiter . . . . . 363
6.1.4.4 Build-up of Layer 2 . . . . . 363
6.1.4.5 Build-up of Layer 3 . . . . . 364
6.1.5 Dialling the Desired Link . . . . . 364
6.1.5.1 Reception of Dialled Numbers at SACCO-A . . . . . 364
6.1.5.2 Preparing to Loop Data from Terminal A to Terminal B . . . . . 365

**Table of Contents** **Page**


6.1.6 Calling up Subscriber B . . . . . 365
6.1.6.1 Activating the Line to Subscriber B . . . . . 365
6.1.6.2 Enabling the Arbiter . . . . . 366
6.1.6.3 Build-up of Layer 2 . . . . . 366
6.1.6.4 Build-up of Layer 3 . . . . . 367
6.1.7 Completing the Call . . . . . 367
6.1.7.1 Receiving the Hook-off Information at the ELIC® . . . . . 367
6.1.7.2 Closing the Data Loop . . . . . 368
6.1.7.3 Giving both Terminals the ‘Go-Ahead’ to Transceive Data . . . . . 368
6.2 D-Channel Delay Due to Arbitration . . . . . 369
6.3 Behaviour of the SACCO-A when a RFIFO Overflow Occurs . . . . . 375

**7 Electrical Characteristics** . . . . . 377

**8 Package Outlines** . . . . . 395

**9 Appendix** . . . . . 396
9.1 Differences between EPIC®-1 (PEB 2055) and the ELIC®-EPIC® . . . . 396
9.2 Working Sheets . . . . . 396
9.2.1 Register Summary for EPIC® Initialization . . . . . 397
9.2.2 Switching of PCM Time Slots to the CFI Interface (data downstream) . . 401
9.2.3 Switching of CFI Time Slots to the PCM Interface (data upstream) . . . . 402
9.2.4 Preparing EPIC®s C/I Channels . . . . . 403
9.2.5 Receiving and Transmitting IOM®-2 C/I-Codes . . . . . 404
9.3 Development Tools . . . . . 405
9.3.1 SIPB 5000 Mainboard . . . . . 405
9.3.2 SIPB 5122 IOM®-2 Line Card Module (ELIC®) . . . . . 406

**10 Lists** . . . . . 407
10.1 Glossary . . . . . 407


IOM®, IOM®-1, IOM®-2, SICOFI®, SICOFI®-2, SICOFI®-4, SICOFI®-4µC, SLICOFI®, ARCOFI® , ARCOFI®-BA, ARCOFI®-SP, EPIC®-1, EPIC®-S, ELIC®, IPAT®-2, ITAC®, ISAC®-S, ISAC®-S TE, ISAC®-P, ISAC®-P TE, IDEC®, SICAT®, OCTAT®-P, QUAT®-S are registered trademarks of Siemens AG.

MUSAC™-A, FALC™54, IWE™, SARE™, UTPT™, ASM™, ASP™ are trademarks of Siemens AG.

Purchase of Siemens $I^2C$ components conveys a license under the Philips' $I^2C$ patent to use the components in the $I^2C$-system provided the system conforms to the $I^2C$ specifications defined by Philips. Copyright Philips 1983.

# 1 Overview

The PEB 20550 (**E**xtended **Li**ne Card **C**ontroller) is a highly integrated controller circuit optimized for line card and key system applications. It combines all functional blocks necessary to manage up to 32 digital (ISDN or proprietary) or 64 analog subscribers.

The switching and layer-1 control capability of the EPIC-1 (PEB 2055) constitutes a major functional block of the ELIC.

For layer-2 support, two independent Special Application Communication Controllers (SACCO) are available. One typically handles the communication with the group controller, the other is used to serve the subscriber terminals. A D-channel arbiter is employed to multiplex the HDLC controller between multiple subscribers while maintaining full duplex signaling protocols (e.g. LAPD).

Additionally, typical line card glue logic functions such as a power-up reset generator, a watchdog timer and two parallel ports are integrated.

The ELIC is implemented in a Siemens advanced 1.0-µm CMOS-technology and manufactured in a P-MQFP-80-1 package.

The ELIC is a member of a new chip family supporting D-channel multiplexing on the line card and in the subscriber terminal. This concept allows an highly economical implementation of digital subscriber lines.

**Chip Family**

Line Cards:

| | | |
|---|---|---|
| PEB 20550 | Extended Line Card Controller | (ELIC) |
| PEB 2096 | Octal $U_{PN}$ Transceiver | (OCTAT-P) |
| PEB 2095 | ISDN Burst Transceiver Circuit | (IBC) |
| PEB 2084 | QUAD $S_0$ Transceiver | (QUAT-S) |
| PEB 2465 | QUAD DSP based Codec Filter | (SICOFI-4) |
| PEB 2075 | ISDN D-Channel Exchange Controller | (IDEC) |

Terminals:

| | | |
|---|---|---|
| PSB 2196 | Digital Subscriber Access Controller for $U_{PN}$ Interface | (ISAC-P TE) |
| PEB 2081 (V3.2) | S/T-Bus Interface Circuit Extended | (SBCX) |

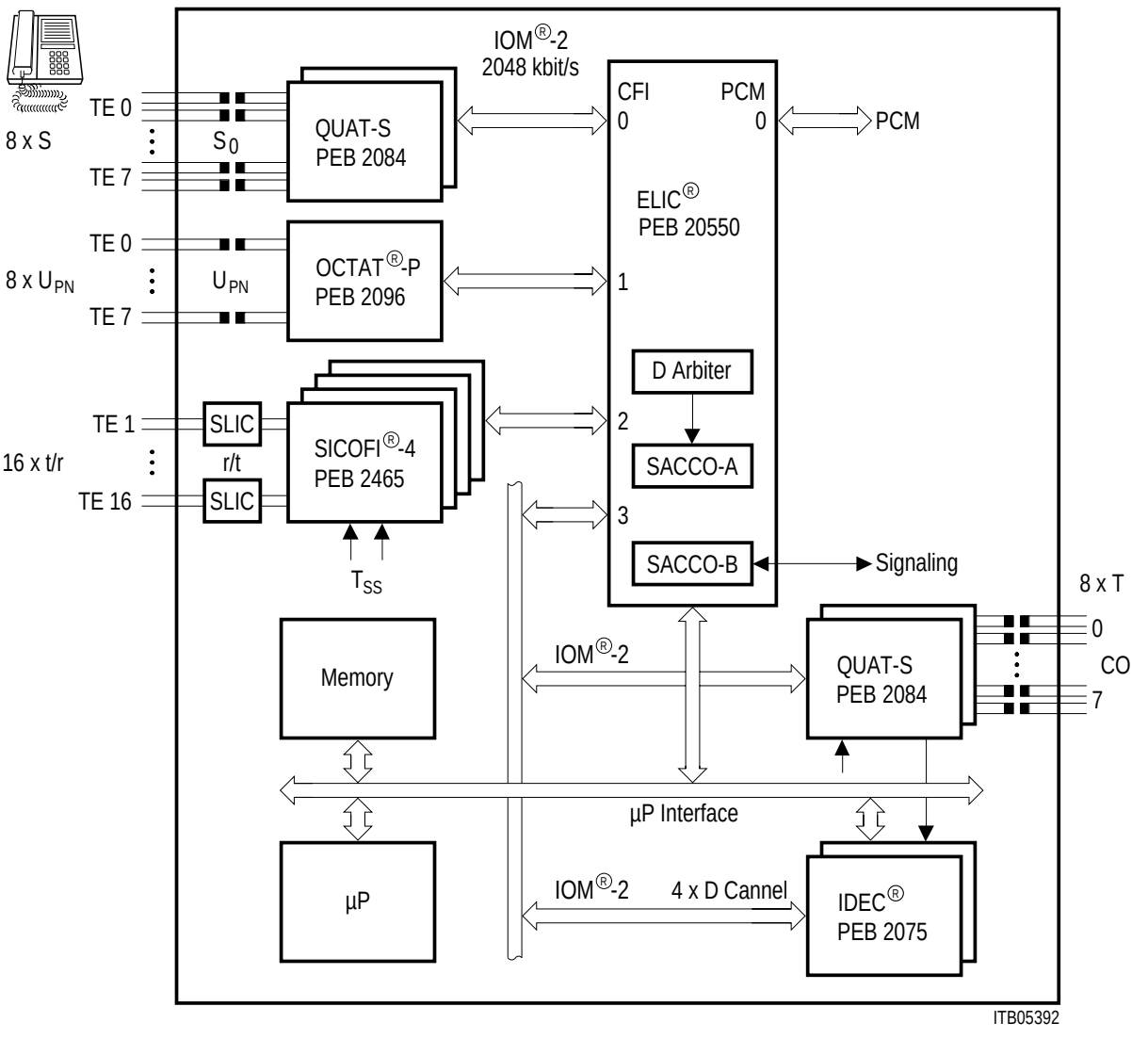



**Figure 1**
**Example for an Integrated Analog / Digital PBX**

SIEMENS

# Extended Line Card Interface Controller ELIC®

**PEB 20550**
**PEF 20550**

**Versions 1.3** **CMOS**

## 1.1 Features



**P-MQFP-80-1**

### Switching (EPIC®-1)

- Non-blocking switch for 32 digital (e.g. ISDN) or 64 voice subscribers
  - Bandwidth 16, 32, or 64 kbit/s
  - Two consecutive 64-bit/s channels can be switched as a single 128-kbit/s channel
- Freely programmable time slot assignment for all subscribers
- Synchronous µP-access to two selected channels
- Two types of serial interfaces independently programmable over a wide data range (128 - 8192 kbit/s)
  - PCM-interface
    Tristate control signals for external drivers
    Programmable clock shift
    Single or double data clock
  - Configurable interface
    Configurable for IOM-, SLD- and PCM-applications
    High degree of flexibility for datastream adaption
    Programmable clockshift
    Single or double data clock

| Type | Ordering Code | Package |
|---|---|---|
| PEB 20550 | Q67101-H6484 | P-MQFP-80-1 (SMD) |
| PEF 20550 | Q67101-H6605 | P-MQFP-80-1 (SMD) |

### Handling of Layer-1 Functions (EPIC®-1)

- Change detection for C/I-channel (IOM-configuration) or feature control (SLD-configuration)
- Additional last-look logic for feature control (SLD-configuration)
- Buffered monitor (IOM-configuration) or signaling channel (SLD-configuration)

### Handling of Layer-2 Functions (SACCO)

- Two independent full duplex HDLC-channels
  - Serial interface
    Data rate up to 4 Mbit/s
    Independent time slot assignment for each channel with programmable time slot length (1-256 bits)
    Support of bus configuration with collision resolution
    Continuous transmission of 1 to 32 bytes possible
  - Protocol support
    Auto-mode, fully compatible to PEB 2050 (PBC) protocol
    Non-auto mode, address recognition capability
    Transparent mode, HDLC-framing only
    Extended transparent mode, fully transparent without HDLC-framing
  - 64-bytes FIFO’s per HDLC-channel and direction

### D-channel Multiplexing (D-channel arbiter)

- Serving of multiple subscribers with one HDLC-controller
- Full duplex signaling protocols (e.g. LAPD or proprietary) supported
- Programmable priority scheme
- Broadcast transmission

### Line Card Glue Logic

- Power-up reset generator
- Watchdog timer
- Parallel ports (8-bit input, 4-bit I/O)

### Boundary Scan Support

- Fully IEEE 1149.1 compatible
- 32-bit device identification register

### Bus Interface

- Siemens/Intel or Motorola type µP-interface
- 8-bit demultiplexed bus interface
- FIFO-access interrupt or DMA controlled

## 1.2 Pin Configuration

(top view)

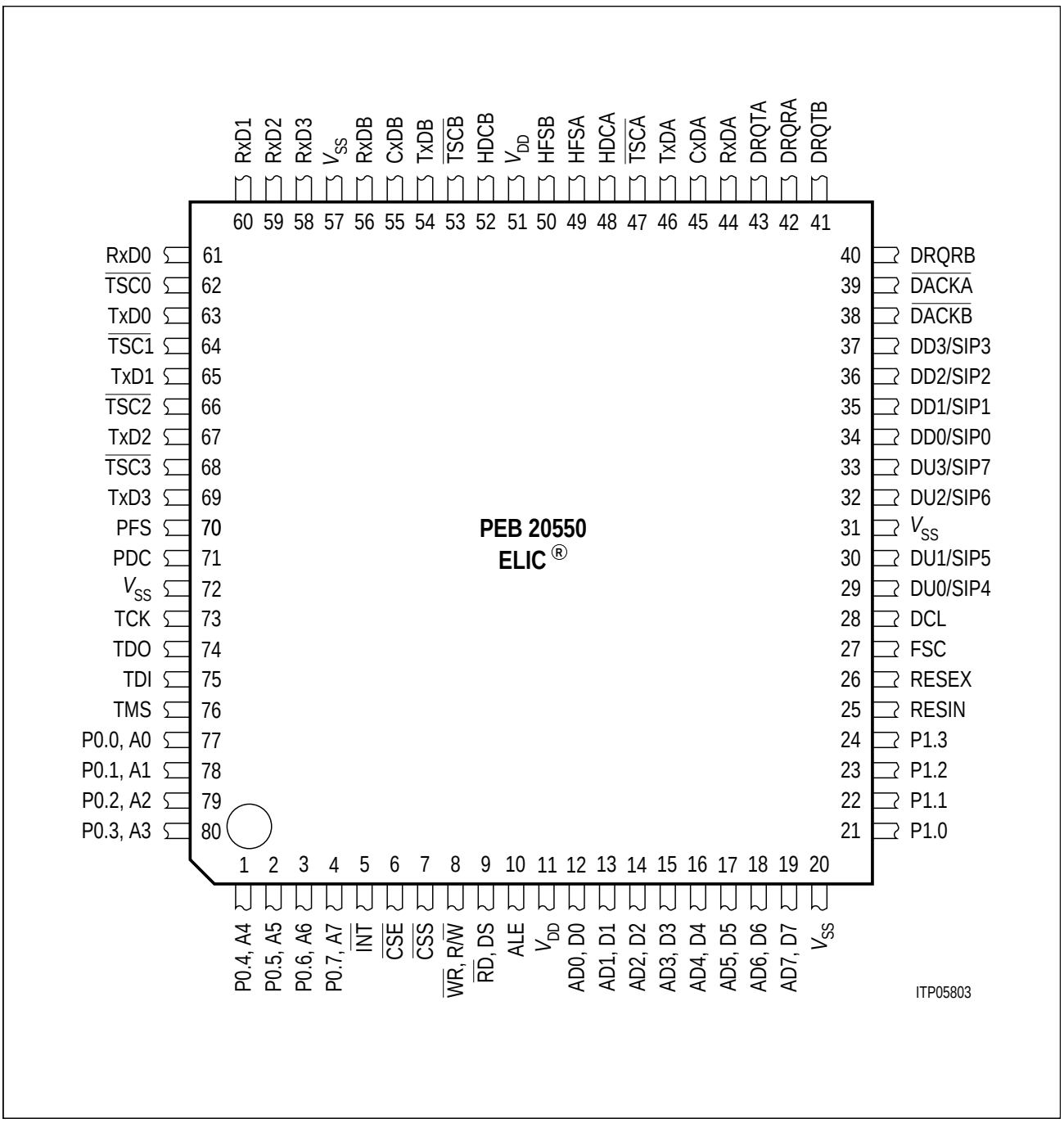



**Figure 2**

## 1.3 Pin Definitions and Functions

### µ-Processor Interface

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---|---|---|---|
| 6 | $\overline{\text{CSE}}$ | I | **C**hip **S**elect **EPIC-1**; active low. A "low" on this line selects all registers (excluding the SACCO-registers) for read/write operations. |
| 7 | $\overline{\text{CSS}}$ | I | **C**hip **S**elect **SACCO**; active low. A "low" on this line selects the SACCO-registers for read/write operations. |
| 8 | $\overline{\text{WR}}$, R/$\overline{\text{W}}$ | I | **W**rite, active low, Siemens/Intel bus mode.<br>When "low", a write operation is indicated.<br>**R**ead/**W**rite, Motorola bus mode.<br>When "high" a valid µP-access identifies a read operation, when "low" it identifies a write access. |
| 9 | $\overline{\text{RD}}$, DS | I | **R**ead, active low, Siemens/Intel bus mode.<br>When "low" a read operation is indicated.<br>**D**ata **S**trobe, Motorola bus mode.<br>A rising edge marks the end of a read or write operation. |
| 12<br>13<br>14<br>15<br>16<br>17<br>18<br>19 | AD0, D0<br>AD1, D1<br>AD2, D2<br>AD3, D3<br>AD4, D4<br>AD5, D5<br>AD6, D6<br>AD7, D7 | I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O | **A**ddress/**D**ata **B**us; multiplexed bus mode.<br>Transfers addresses from the µP-system to the ELIC and data between the µP and the ELIC.<br>**D**ata **B**us; demultiplexed bus mode.<br>Transfers data between the µP and the ELIC.<br>**When driving data the pins have push pull characteristic, otherwise they are in the state high impedance.** |
| 10 | ALE | I | **A**ddress **L**atch **E**nable<br>ALE controls the on chip address latch in multiplexed bus mode. While ALE is "high", the latch is transparent. The falling edge latches the current address. During the first read/write access following reset ALE is evaluated to select the bus mode. |

**Pin Definitions and Functions** (cont'd)

**µ-Processor Interface**

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---|---|---|---|
| 77<br>78<br>79<br>80<br>1<br>2<br>3<br>4 | P0.0,A0<br>P0.1,A1<br>P0.2,A2<br>P0.3,A3<br>P0.4,A4<br>P0.5,A5<br>P0.6,A6<br>P0.7,A7 | I<br>I<br>I<br>I<br>I<br>I<br>I<br>I | **A**ddress **B**us, demultiplexed bus mode.<br>Transfers addresses from the µP-system to the ELIC.<br>**P**ort **0**, multiplexed bus mode.<br>Parallel input port. The current data is latched with the falling edge of $\overline{RD}$, DS. |
| 21<br>22<br>23<br>24 | P1.0<br>P1.1<br>P1.2<br>P1.3 | I/O<br>I/O<br>I/O<br>I/O | **P**ort **1**<br>4-bit I/O port. Every pin can be configured individually as input or output. For inputs the current data is latched with the falling edge of $\overline{RD}$, DS. |
| 5 | $\overline{INT}$ | O<br>(OD) | **I**nterrupt **R**equest, active low.<br>This signal is activated when the ELIC requests an interrupt. Due to the open drain (OD) characteristic of $\overline{INT}$ multiple interrupt sources can be connected together. |
| 25 | RESIN | O | **R**eset **I**ndication<br>This pin is set to "high", when the ELIC executes either a power-up reset, a watchdog timer reset, an external reset (RESEX) or a software system reset. |
| 26 | RESEX | I | **R**eset **E**xternal<br>A "high" forces the ELIC into reset state. |

**Pin Definitions and Functions** (cont'd)

**EPIC®-1 Interface**

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---|---|---|---|
| 70 | PFS | I | **PCM**-**I**nterface **F**rames **S**ynchronization |
| 71 | PDC | I | **PCM**-**I**nterface **D**ata **C**lock<br>Single or double data rate. |
| 61<br>60<br>59<br>58 | RxD0<br>RxD1<br>RxD2<br>RxD3 | I<br>I<br>I<br>I | **R**eceive **PCM**-**I**nterface **D**ata<br>Time-slot oriented data is received on this pins and forwarded into the downstream data memory of the EPIC-1. |
| 63<br>65<br>67<br>69 | TxD0<br>TxD1<br>TxD2<br>TxD3 | O<br>O<br>O<br>O | **T**ransmit **PCM**-**I**nterface **D**ata<br>Time-slot oriented data is shifted out of the EPIC-1s upstream data memory on this lines. **For time-slots which are flagged in the tristate data memory or when bit OMDR:PSB is reset the pins are set in the state high impedance.** |
| 62<br>64<br>66<br>68 | $\overline{\text{TSC0}}$<br>$\overline{\text{TSC1}}$<br>$\overline{\text{TSC2}}$<br>$\overline{\text{TSC3}}$ | O<br>O<br>O<br>O | **T**ristate **C**ontrol<br>Supplies a control signal for an external driver. These lines are "low" when the corresponding TxD-outputs are valid. During reset these lines are "high". |
| 27 | FSC | I/O | **F**rame **S**ynchronization<br>Input or output in IOM-configuration. Direction indication signal in SLD-mode. |
| 28 | DCL | I/O | **D**ata **C**lock<br>Input or output in IOM, slave clock in SLD configuration. In IOM-configuration single or double data rate, single data rate in SLD-mode. |

**Pin Definitions and Functions** (cont'd)

**EPIC®-1 Interface**

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---|---|---|---|
| 29<br>30<br>32<br>33 | DU0/SIP4<br>DU1/SIP5<br>DU2/SIP6<br>DU3/SIP7 | I/IO (OD)<br>I/IO (OD)<br>I/IO (OD)<br>I/IO (OD) | **D**ata **U**pstream Input; IOM- or PCM-configuration.<br>**S**erial **I**nterface **P**ort, SLD-configuration.<br>**Depending on the bit OMDR:COS these lines have push pull or open drain characteristic. For unassigned channels or when bit OMDR:CSB is reset the pins are in the state high impedance.** |
| 34<br>35<br>36<br>37 | DD0/SIP0<br>DD1/SIP1<br>DD2/SIP2<br>DD3/SIP3 | O/IO (OD)<br>O/IO (OD)<br>O/IO (OD)<br>O/IO (OD) | **D**ata **D**ownstream Output, IOM- or PCM-configuration.<br>**S**erial **I**nterface **P**ort, SLD-configuration.<br>**Depending on the bit OMDR:COS these lines have push pull or open drain characteristic. For unassigned channels or when bit OMDR:CSB is reset the pins are in the state high impedance.** |

**Pin Definitions and Functions** (cont'd)

**SACCO-Interface**

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---|---|---|---|
| 49<br>50 | HFSA<br>HFSB | I<br>I | **H**DLC-Interface **F**rame **S**ynchronization Channel **A/B**<br>Frame synchronization pulse in clock mode 2, data strobe in clock mode 1. |
| 48<br>52 | HDCA<br>HDCB | I<br>I | **H**DLC-Interface **D**ata **C**lock **C**hannel **A/B**. Single or double data rate. |
| 44<br>56 | RxDA<br>RxDB | I<br>I | **R**eceive **S**erial **D**ata **HDLC**-**C**hannel **A/B**<br>The serial data received on this lines is forwarded into the corresponding HDLC-receive channel. Data is sampled on the<br>– falling edge of HDC (CCR2:RDS = 0) or<br>– rising edge of HDC (CCR2:RDS = 1). |
| 46<br>54 | TxDA<br>TxDB | O (OD)<br>O (OD) | **T**ransmit **S**erial **D**ata **HDLC**-**C**hannel **A/B**.<br>Data output lines of the corresponding HDLC-transmit channel. **Depending on the bit CCR1:ODS the pins have push pull or open drain characteristic. When transmission is disabled ($\overline{\text{TSCA}}$ or B = 1) or when bit CCR2:TXDE is reset the pins are in the state high impedance.** |
| 47<br>53 | $\overline{\text{TSCA}}$<br>$\overline{\text{TSCB}}$ | O<br>O | **T**ristate **C**ontrol **HDLC**-**C**hannel **A/B**, active low.<br>Supplies a control signal for an external driver. When low the corresponding TxD-outputs are valid. The detailed functionality is defined programming the SACCO-registers CCR2:SOC1,SOC0. During reset these lines are high. |
| 45<br>55 | CxDA<br>CxDB | I<br>I | **C**ollision **D**ata **HDLC**-**C**hannel **A/B**<br>In a bus configuration, the external serial bus must be connected to the respective CxD-pin for collision detection.<br>In point-to-point configurations the pin provides a "clear to send" function. When '0'/'1' the transmit channel is enabled/disabled. **If this function is not needed CxDA/B has to be tied to $V_{SS}$.** |

**Pin Definitions and Functions** (cont'd)

**SACCO-Interface**

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---|---|---|---|
| 42<br>40 | DRQRA<br>DRQRB | O<br>O | **DMA**-**R**equest **R**eceiver **C**hannel **A/B**<br>The receiver of HDLC-channel **A/B** requests a DMA-data transfer by activating this lines. The DRQR-pin remains "high" as long as the receiver FIFO requires data transfers. Only blocks of 32, 16, 8 or 4 bytes are transferred. |
| 43<br>41 | DRQTA<br>DRQTB | O<br>O | **DMA**-**R**equest **T**ransmitter **C**hannel **A/B**<br>The transmitter of HDLC-channel **A/B** requests a DMA-data transfer by activating this lines. The DRQT-pin remains "high" as long as the transmit FIFO requires data transfers. The number of data bytes to be transferred from system memory to the FIFO must be written first into the XBCH, XBCL registers (byte count registers). |
| 39<br>38 | $\overline{\text{DACKA}}$<br>$\overline{\text{DACKB}}$ | I<br>I | **DMA**-**A**cknowledge **HDLC**-Channel **A/B**, active low.<br>When "low", this lines notifies the HDLC-channel, that the requested DMA-cycle is in progress. Together with $\overline{\text{RD}}$ (DRQR) or $\overline{\text{WR}}$(DRQT) $\overline{\text{DACK}}$ works like $\overline{\text{CS}}$ to enable a read or write operation to the top of the receive or the transmit FIFO. When $\overline{\text{DACK}}$ is active, the address lines are ignored and the FIFOs are implicitly selected. When $\overline{\text{DACK}}$ is not used it has to be connected to $V_{DD}$. |

**Pin Definitions and Functions** (cont'd)

**Boundary Scan Interface, according to IEEE Std. 1149.1**

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---|---|---|---|
| 76 | TMS | I (internal pull-up) | **T**est **M**ode **S**elect<br>A 0 -> 1 transition on this pin is required to step through the TAP-controller state machine. |
| 75 | TDI | I (internal pull-up) | **T**est **D**ata **I**nput<br>In the appropriate TAP-controller state test data or a instruction is shifted in via this line |
| 74 | TDO | O | **T**est **D**ata **O**utput<br>In the appropriate TAP-controller state test data or a instruction is shifted out via this line. |
| 73 | TCK | I | **T**est **C**lock<br>Single rate test data clock. |

*Note: Pin 75 (TDI) and pin 76 (TMS) are internally connected to $V_{DD}$ via pull-up resistors.*

## 1.4 Logic Symbol

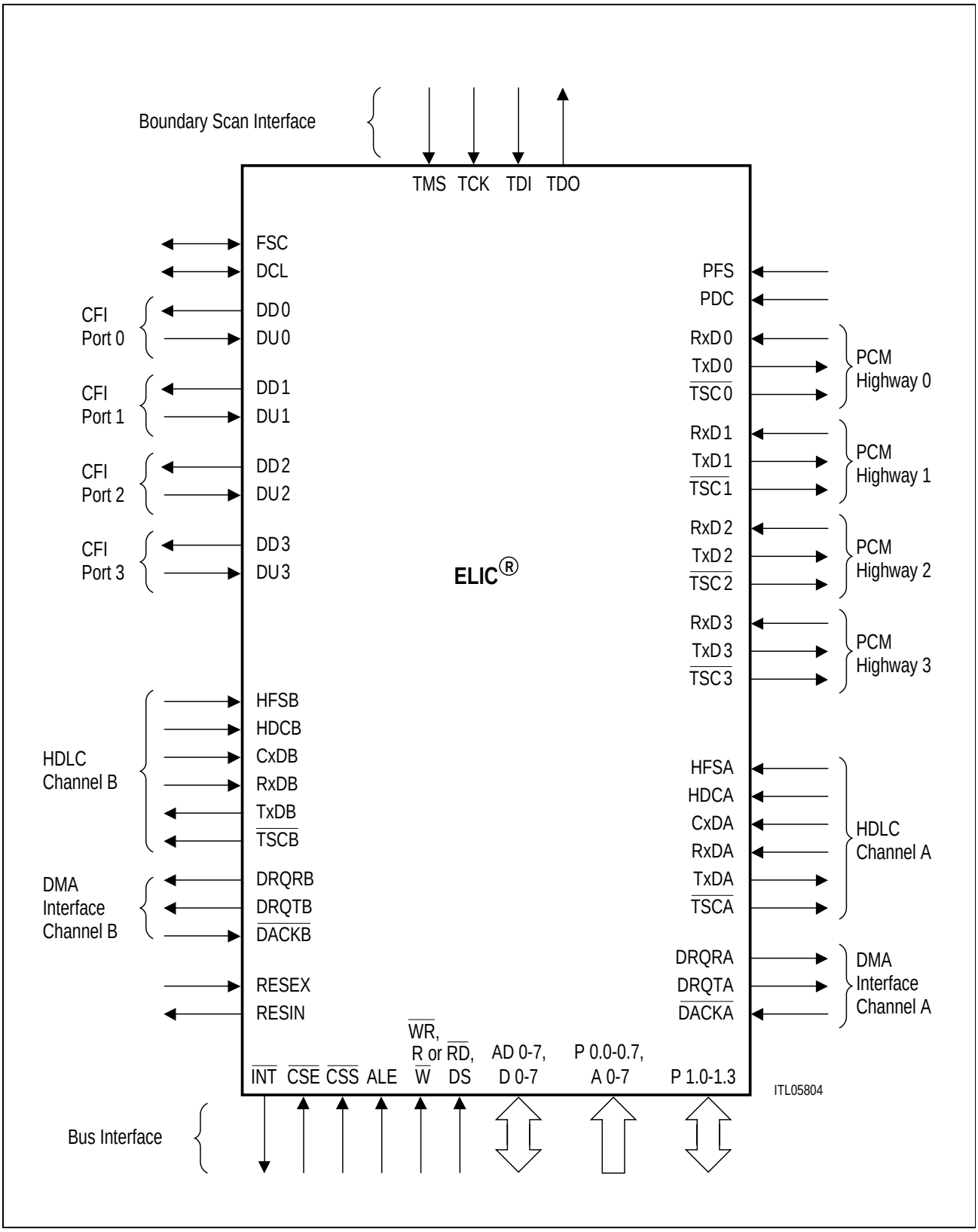



**Figure 3**

## 1.5 Functional Block Diagram

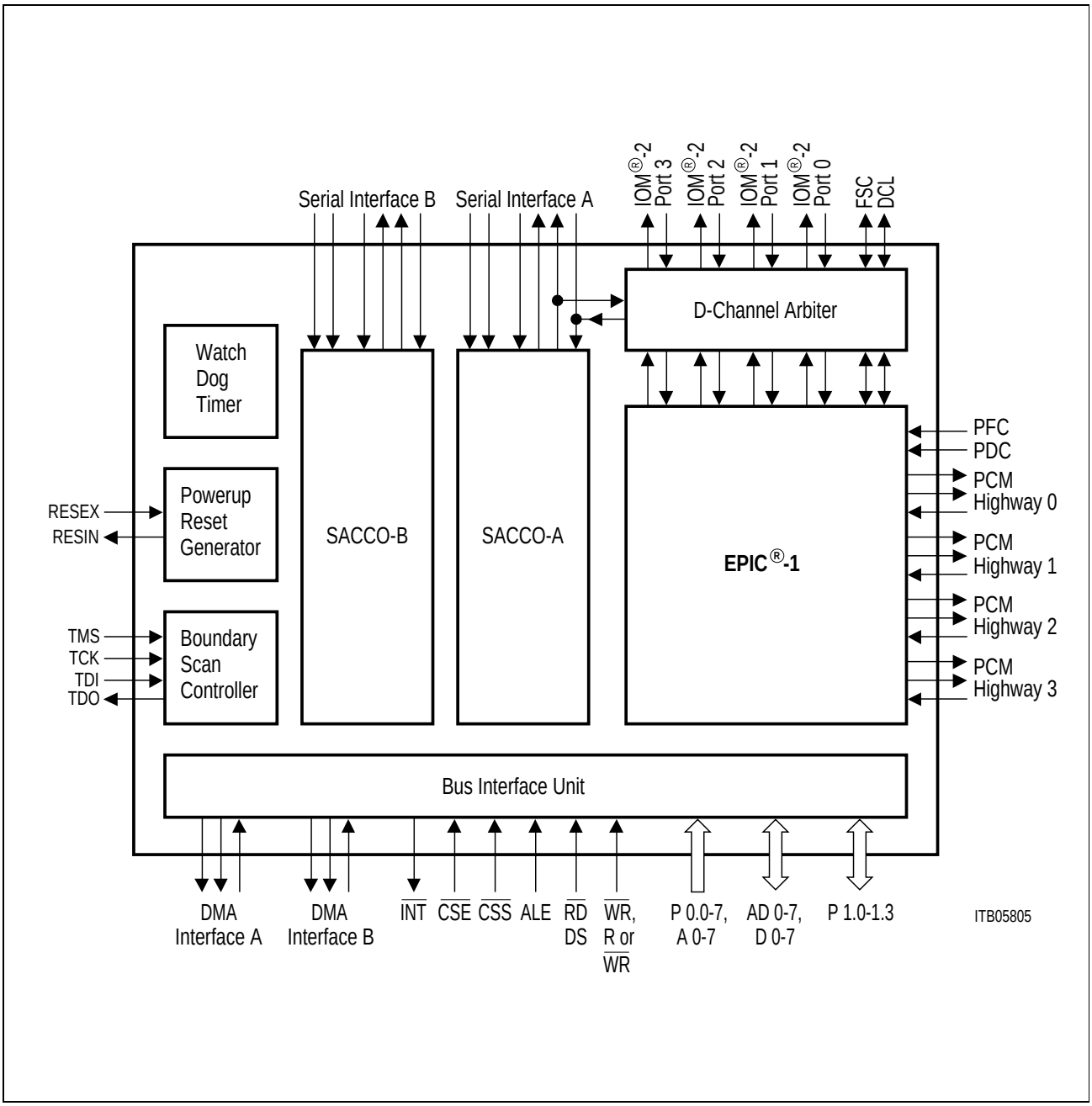



**Figure 4**

## 1.6 System Integration and Application

The main application fields of the ELIC are:

- Digital line cards, different architectures are supported,
- Central control units of key systems,
- Analog line cards,
- DECT line cards.

### 1.6.1 Digital Line Card

#### 1.6.1.1 Switching, Layer-1 Control, Group Controller Signaling

The ELIC performs a switching capability for up to 32 digital subscribers between the PCM- system highway and the IOM-2 interface (64 B-channels). Typically it switches 64-kbit/s channels between the PCM and the IOM-interfaces. Moreover it is able to handle also 16-, 32- and 128-kbit/s channels.

The signaling handler supports the command/indication (C/I) channel which is used to exchange predefined layer-1 information with the transceiver device.

A monitor handler supports the handshake protocol defined on the IOM-monitor channel. It allows programming of layer-1 devices which do not have a dedicated µP interface.

The communication between the line card and the group controller is performed by one of the SACCO-channels. Its auto-mode is optimized for this application and implements a slave station behaviour in normal response mode. The auto-mode is compatible with the PBC (PEB 2050) but due to the large FIFO-size the response time requirements compared to the PBC are reduced drastically.

The data exchange between the line card and the group controller board can take place on a separate signalling highway or on the PCM-highway (due to the time slot capability of the SACCO) (see **figure 5**).

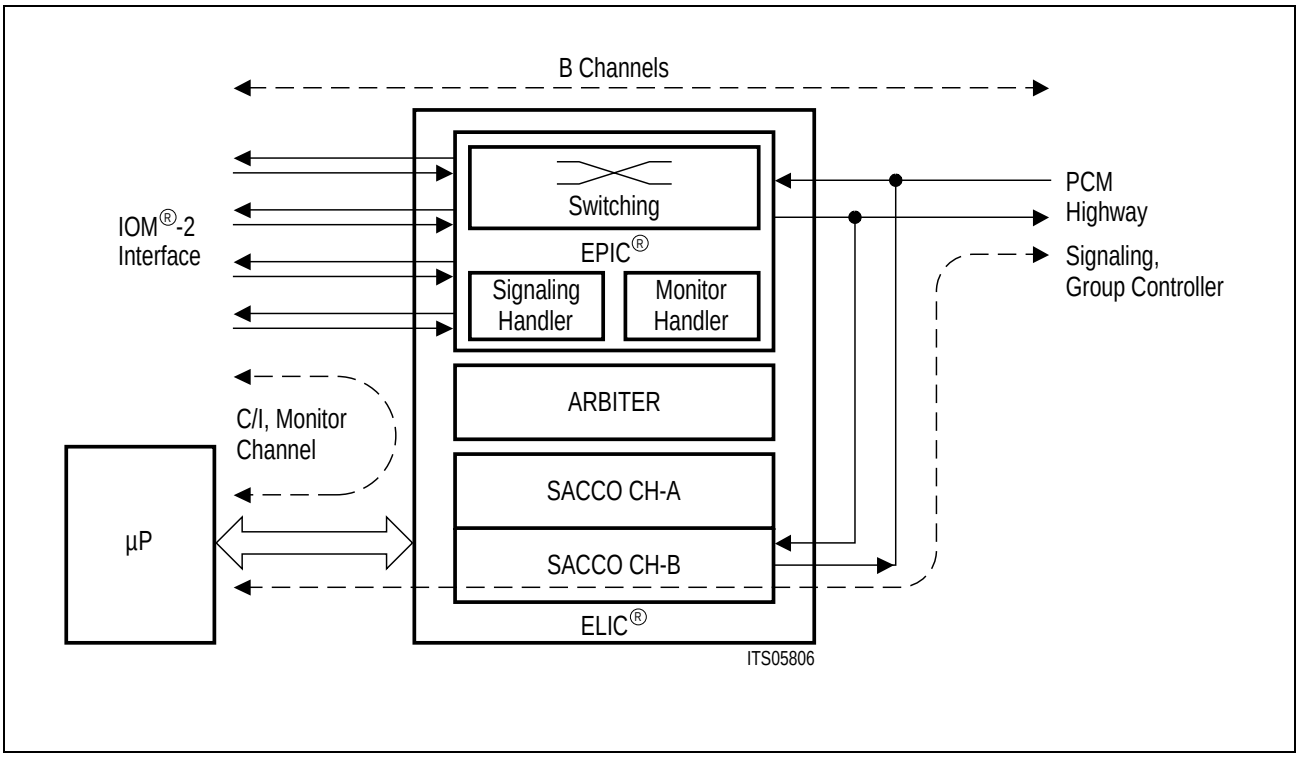



**Figure 5**
**Data Flow - B-channels, Layer-1 Control, Group Controller Signaling**

Another possibility to handle the point-to-multi-point configuration between a group controller and several line cards is a bus structure. The collision detection/resolution function of the SACCO perfectly supports this architecture and allows the application of balanced protocols (see **figure 6**).

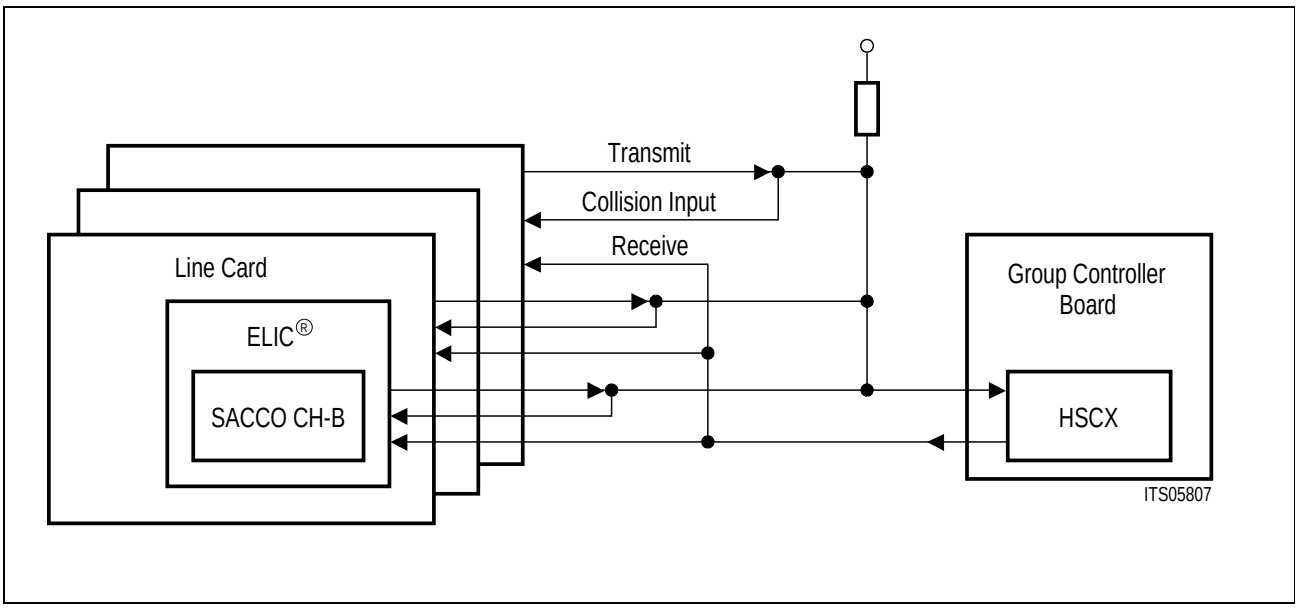



**Figure 6**
**Group Controller Signaling with Bus Structure for Balanced Protocols**

D-channel processing is supported by multiple different architectures:

### 1.6.1.2 Decentralized D-Channel Processing, Multiplexed HDLC-Controller.

Typically the D-channel load has a very bursty characteristic. Taking this into account, the ELIC provides the capability to multiplex one HDLC-controller among several subscribers. This feature results in a drastical reduction of hardware requirements while maintaining all benefits of HDLC based signaling.

A D-channel arbiter is used to assign the receive and transmit HDLC-channel independently to the subscriber terminals.

In downstream direction the arbiter links the transmit channel to one or more (broadcast) programmable IOM-2 D-channels (ports).

In upstream direction the arbiter assigns the HDLC-receive channel to a requesting subscriber and indicates to all other subscribers that their D-channels are blocked, using a control channel.

This configuration supports full duplex layer-2 protocols with bus capability e.g. LAPD or proprietary implementations. Consequently no polling overhead is necessary providing the full 16-kbit/s bandwidth of the D-channel for data exchange.

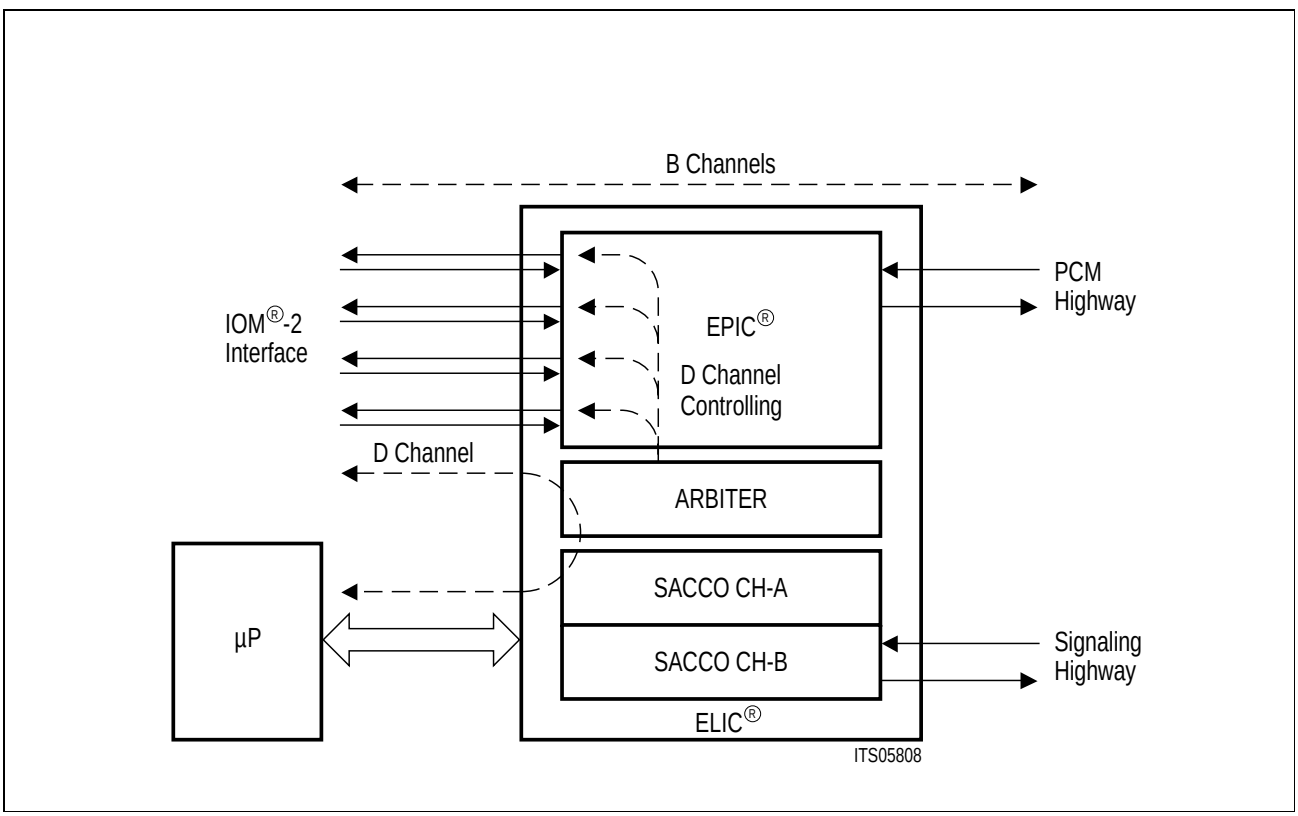



**Figure 7**
**D-Channel Handling with a Multiplexed HDLC-controller**

The control channel is unidirectional and forwards the status information of the corresponding D-channel (blocked or available) towards the subscriber terminal.

Different existing channel structures are used to implement the control channel between the HDLC-controllers on the line card and in the subscriber terminal.

### Control Channel Implementation on the $U_{PN}$-Interface

On an $U_{PN}$-line card, the control channel is either integrated in the C/I-channel or uses the MR-bit, depending on the connected layer-1 device (OCTAT-P -> C/I channel, IBC -> MR-bit).

The $U_{PN}$-transceiver uses the T-channel to transmit the control channel information to the terminal. The T-channel is a sub channel of the $U_{PN}$-interface with a bandwidth of 2 kbit/s.

In the subscriber terminal the control channel is included again in the IOM-2 protocol.

Depending on the terminal configuration two alternatives can be selected in the terminal transceiver device.

The blocked/available information is translated directly into the S/G-bit (Stop/Go) when no subsequent transceiver circuit is present in the terminal. The S/G-bit is evaluated by the terminal HDLC-controller ICC. It stops data transmission immediately when the S/G-bit is set to 1.

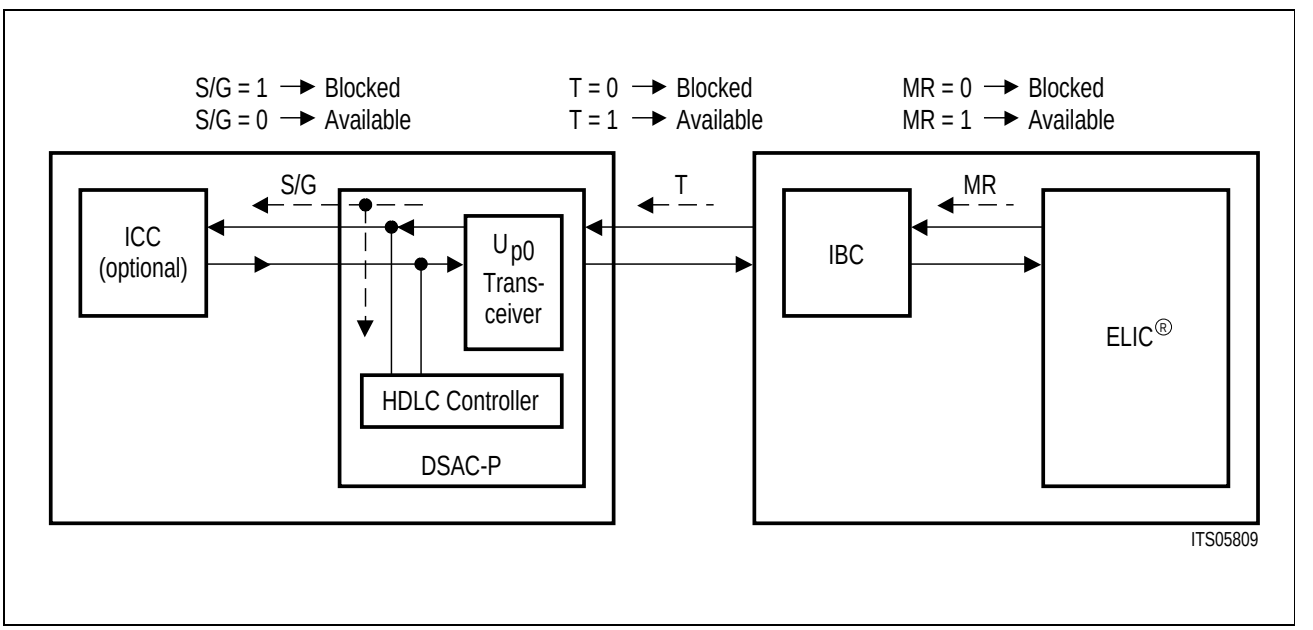



**Figure 8**
**Control Channel Implementation with IBC (PEB 2095) as Line Card Transceiver**

In **figure 9** a Control Channel Implementation with OCTAT-P as line card transceiver can be seen.

When an additional transceiver device is integrated in the terminal (e.g. an $S_0$-adapter, PEB 2081 (SBCX)) the control channel is translated into the A/B-bit (bit5, 4th byte, IOM-channel 2, downstream). The A/B-bit is monitored by the SBCX. A/B = 1 indicates that the corresponding D-channel is available (A/B = 0 blocked). Depending on this information, the SBCX controls the E-bit on the $S_0$-bus and the S/G-bit on the IOM-2 interface. When A/B = 0 the E-bit is forced in the inverted D-bit state, the S/G-bit is set to high. As a result all active transmitters in the terminal and on the $S_0$-bus are forced to abandon their messages.

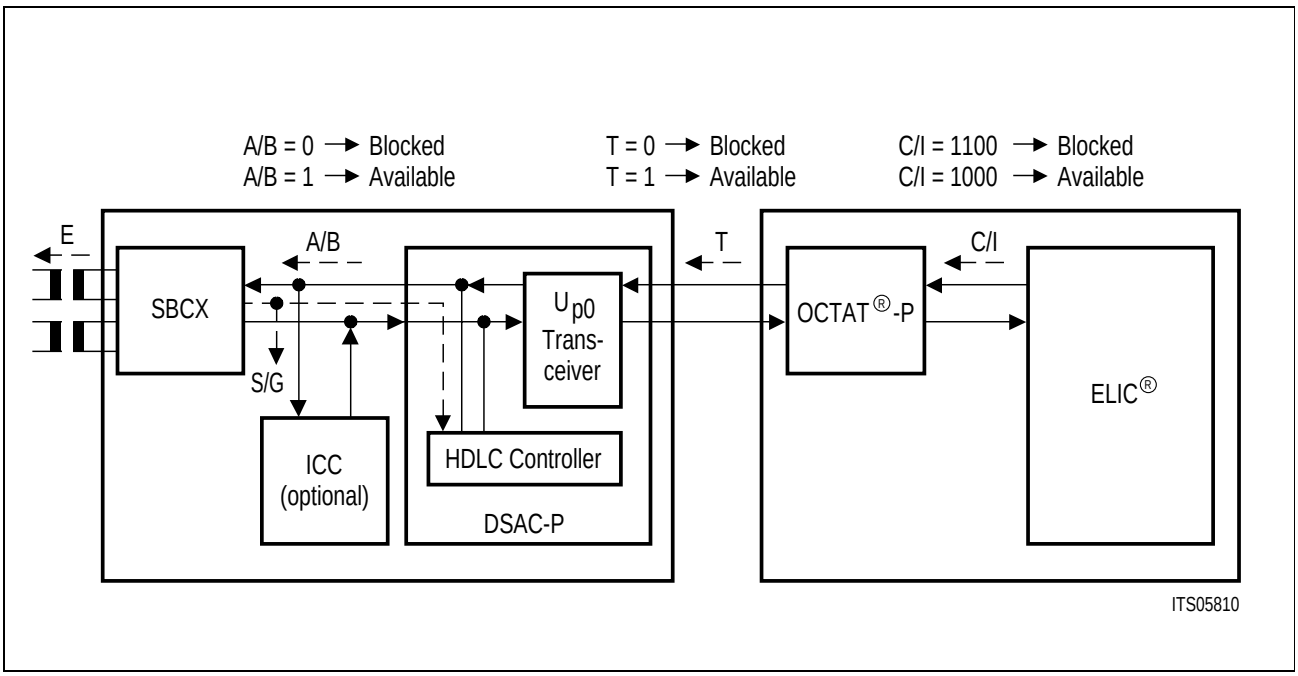



**Figure 9**
**Control Channel Implementation with OCTAT®-P (PEB 2096) as Line Card Transceiver and $S_0$-Adapter.**

### Control Channel Implementation on the $S_0$-Interface

When using the ELIC on a $S_0$-line card the structure is much simpler because the $S_0$-interface provides contention resolution as a standard feature. In this structure the QUAT-S modifies the E-bit on the line card, i.e. standard $S_0$-phones can be connected. The control channel on the line card is included in the C/I-channel.

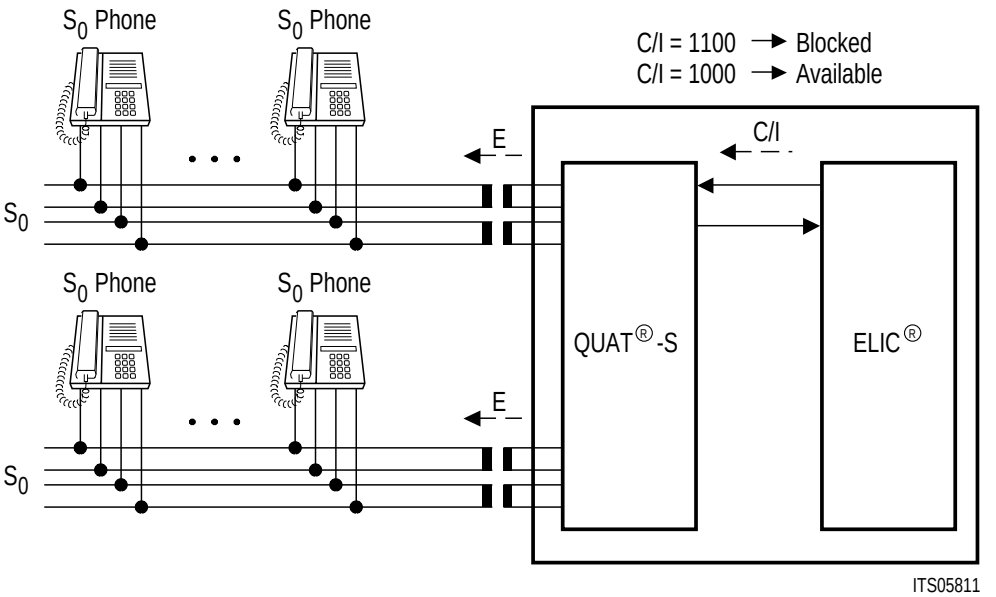



**Figure 10**
**Control Channel Implementation on a $S_0$-Line Card**

Even with a multiplexed HDLC controller signaling and packet data can be mixed on a $S_0$ line card. The priority scheme of the $S_0$ bus (2 priority classes) guarantees, that signal data is not delayed by data packets.

### 1.6.1.3 Decentralized D-Channel Processing, Dedicated HDLC-Controller per Subscriber

In this configuration IDECs (ISDN D-channel exchange controller, PEB 2075) handle the layer-2 functions for signaling and data packets in the D-channel. The extracted data is separated and sent via the µP and the SACCO to the system interface. In this configuration signaling data is transferred on the PCM-highway, for packet data a dedicated bus system with collision resolution is used.

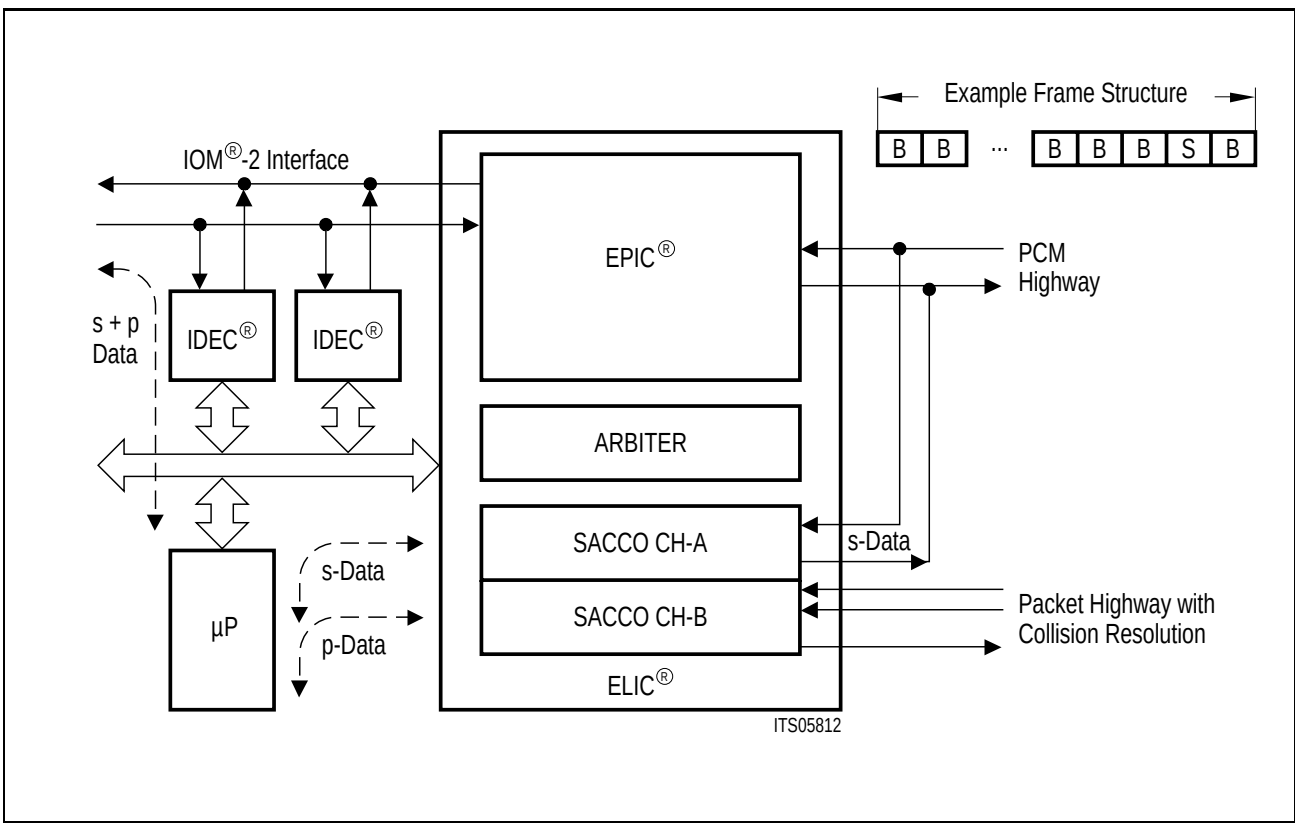



**Figure 11**
**Line Card Architecture for Completely Decentralized D-Channel Processing**

### 1.6.1.4 Decentralized D-Channel Processing, Multiplexed plus Dedicated HDLC-Control

Especially when packet data is supported in the D-channel one multiplexed HDLC controller may create a bottleneck situation. One solution to overcome this problem is the combination of the multiplexing scheme with additional layer-2 controllers which can be temporarily assigned to individual subscribers on request.

In normal operation all subscribers are managed by the D-channel arbiter and share SACCO-A. When a subscriber requests a special type of service, the system can switch a dedicated HDLC controller and exclude this subscriber temporarily from the arbitration. For small systems SACCO-B, for bigger systems IDECs may be used as an assignable controller resource.

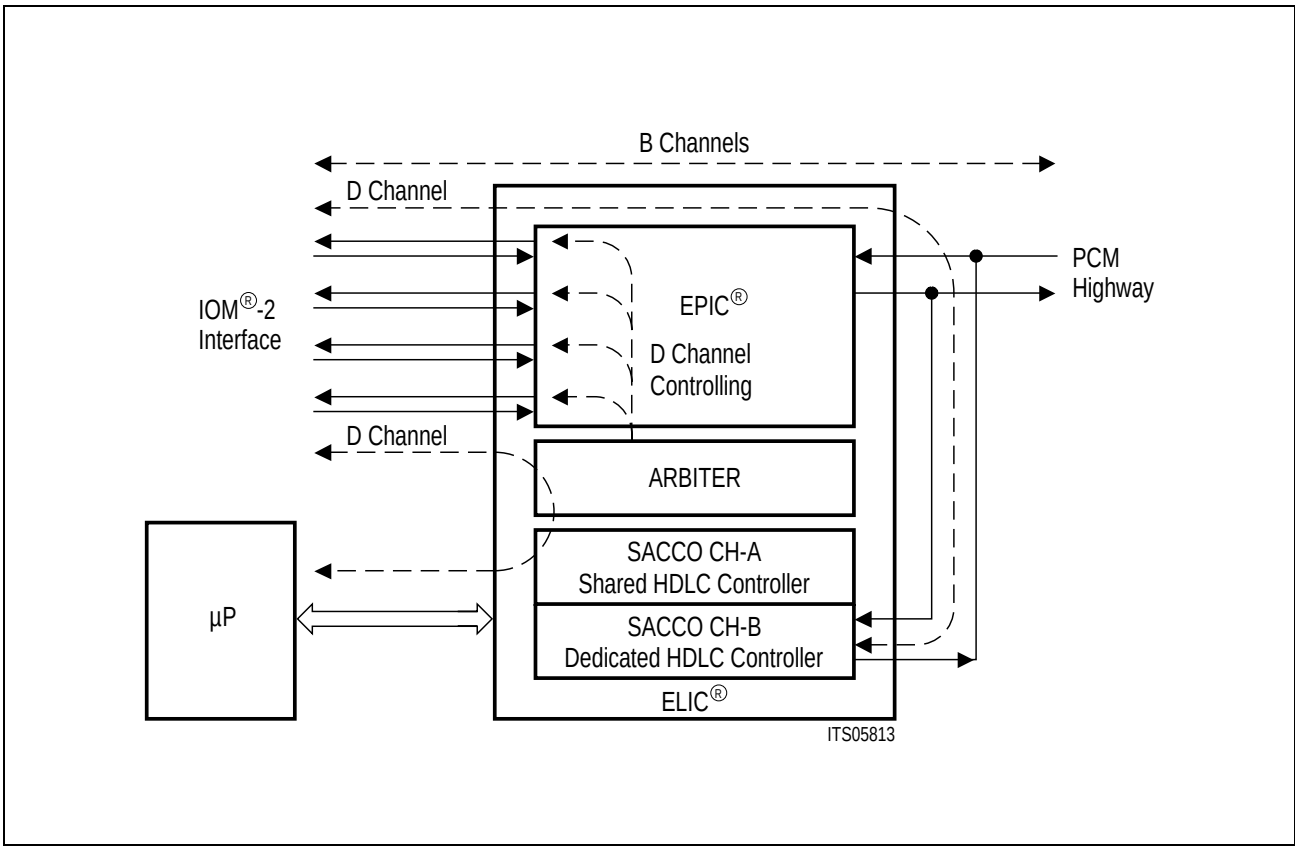



**Figure 12**
**SACCO-B as Assignable HDLC-Controller**

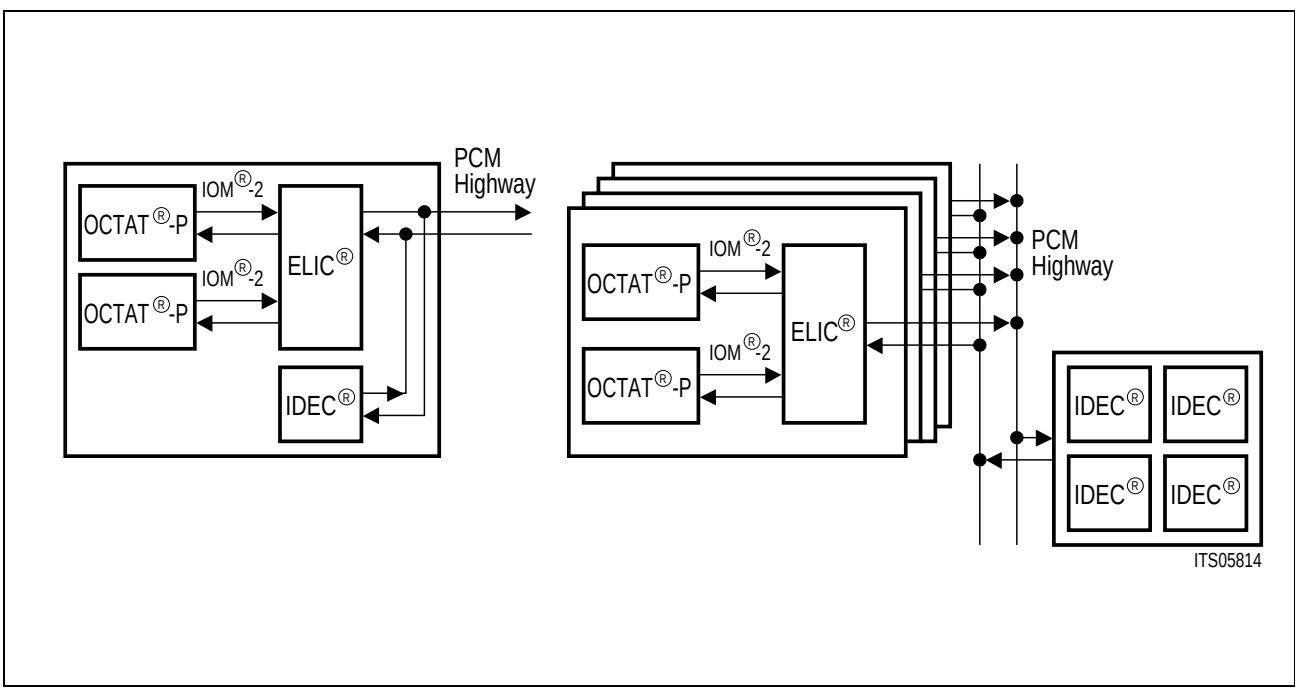



**Figure 13**
**IDEC®-S as Assignable HDLC-Controller Resources**

### 1.6.1.5 Central D-Channel Processing

In this application the EPIC-1 not only switches the B-channels and performs the C/I- and monitor channel control function, but switches also the D-channel data onto the system highway. In upstream direction the EPIC-1 can combine up to four 16-kbit/s D-channels into one 64-kbit/s channel. In downstream direction it provides the capability to distribute one 64-kbit/s channel in four 16-kbit/s channels.

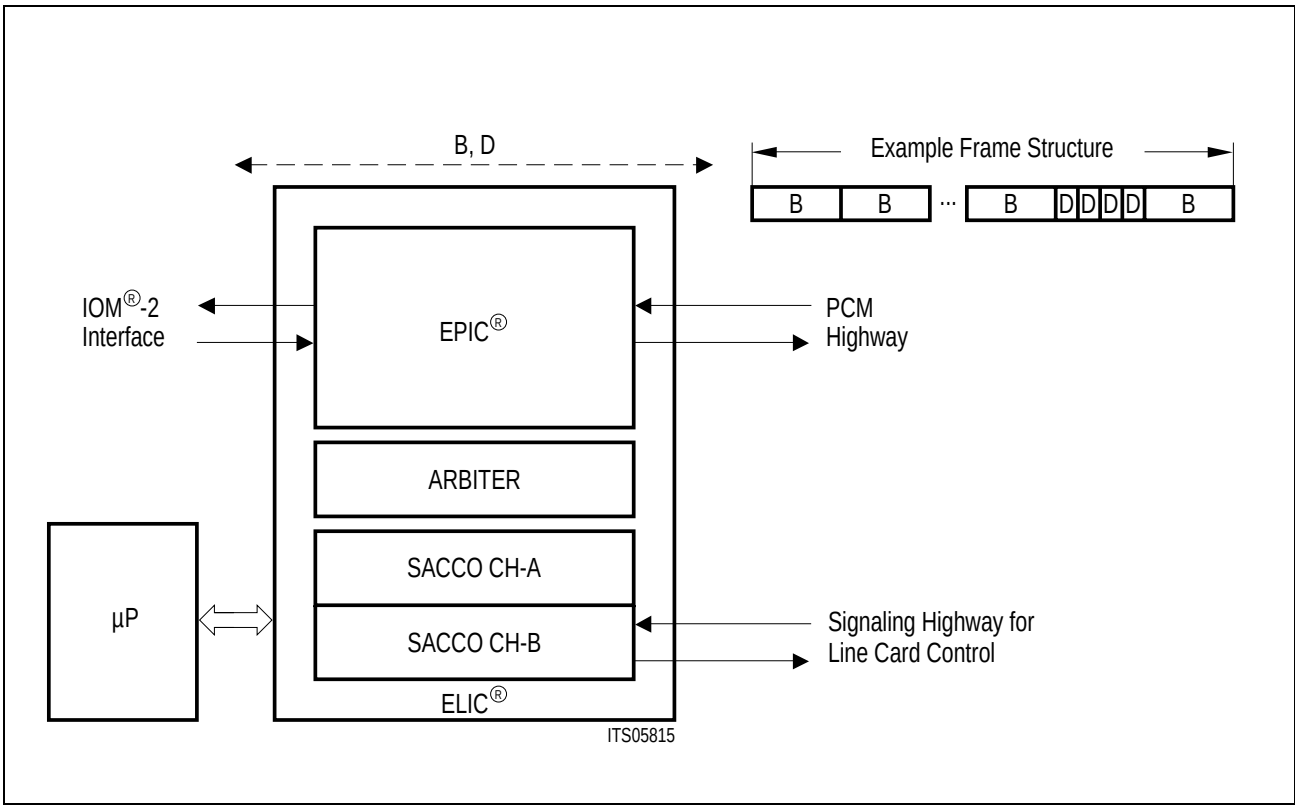



**Figure 14**
**Line Card Architecture for Completely Centralized D-Channel Processing**

### 1.6.1.6 Mixed D-Channel Processing, Signaling Decentralized, Packet Data Centralized

Another possibility is a mixed architecture with centralized packet data and decentralized signaling handling. This is a very flexible architecture which reduces the dynamic load of central processing units by evaluating the signaling information on the line card, but does not require resources for packet data handling. Any increase of packet data traffic does not necessitate a change in the line card architecture, the central packet handling unit can be expanded.

Also in this application IDECs are employed to handle the data on the D-channel. The IDECs separate signaling information from data packets. The signaling messages are transferred to the µP, which in turn hands them over to the group controller using the SACCO. The packet data is processed differently. Together with the collision resolution information it is transferred to one IOM-2 interface of the ELIC. The EPIC-1 switches the channels to the PCM-highway, optionally combining four D-channels to one 64-kbit/s channel. In this configuration one IOM-2 interface is occupied by IDECs, reducing the total switching capability of the EPIC-1 to 24 ISDN-subscribers.

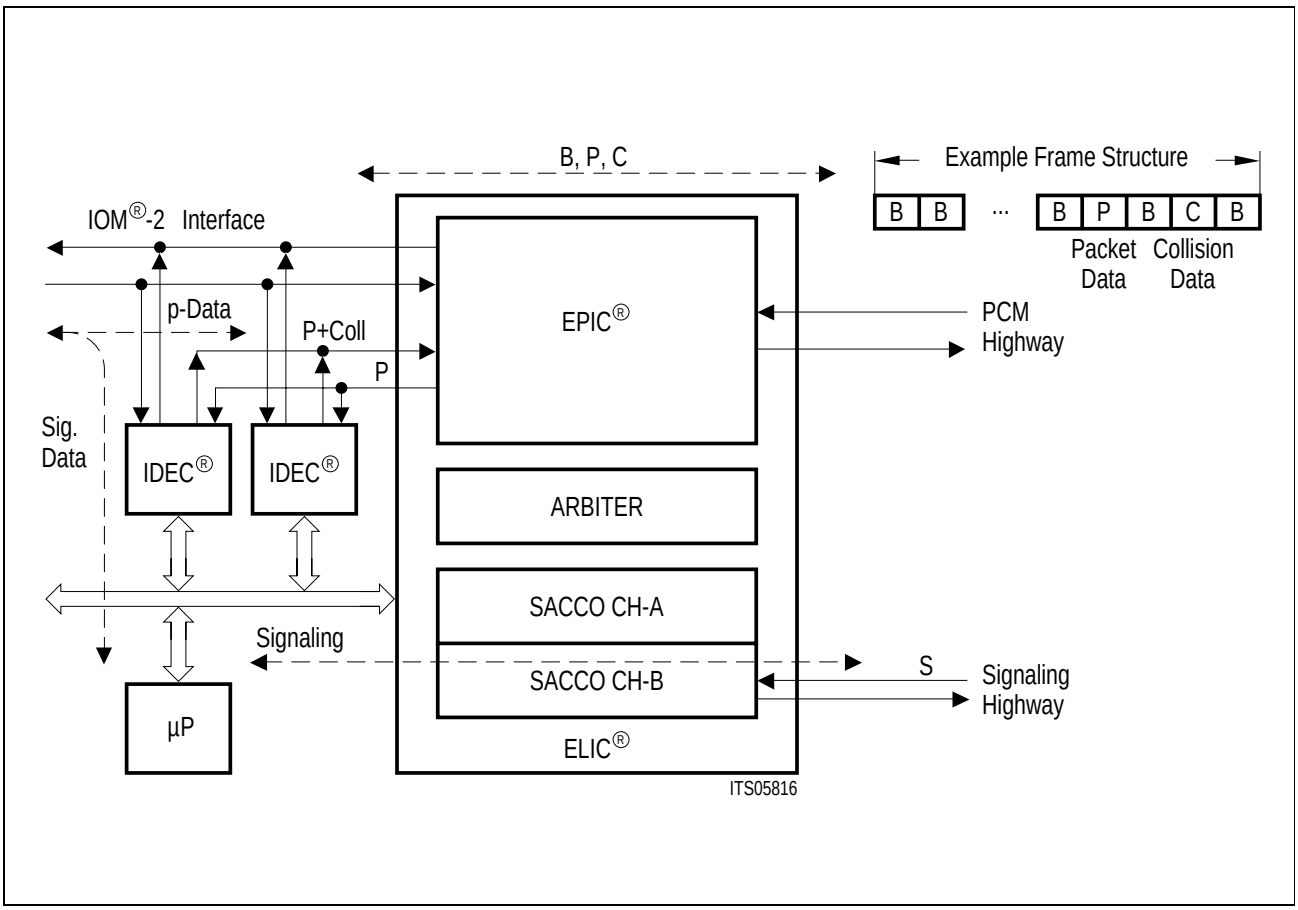



**Figure 15**
**Line Card Architecture for Mixed D-Channel Processing**

Alternatively, the packet and collision data can be directly exchanged between the IDECs and the PCM-highway. Thus, the full 32 subscriber switching capability of the EPIC-1 is retained.

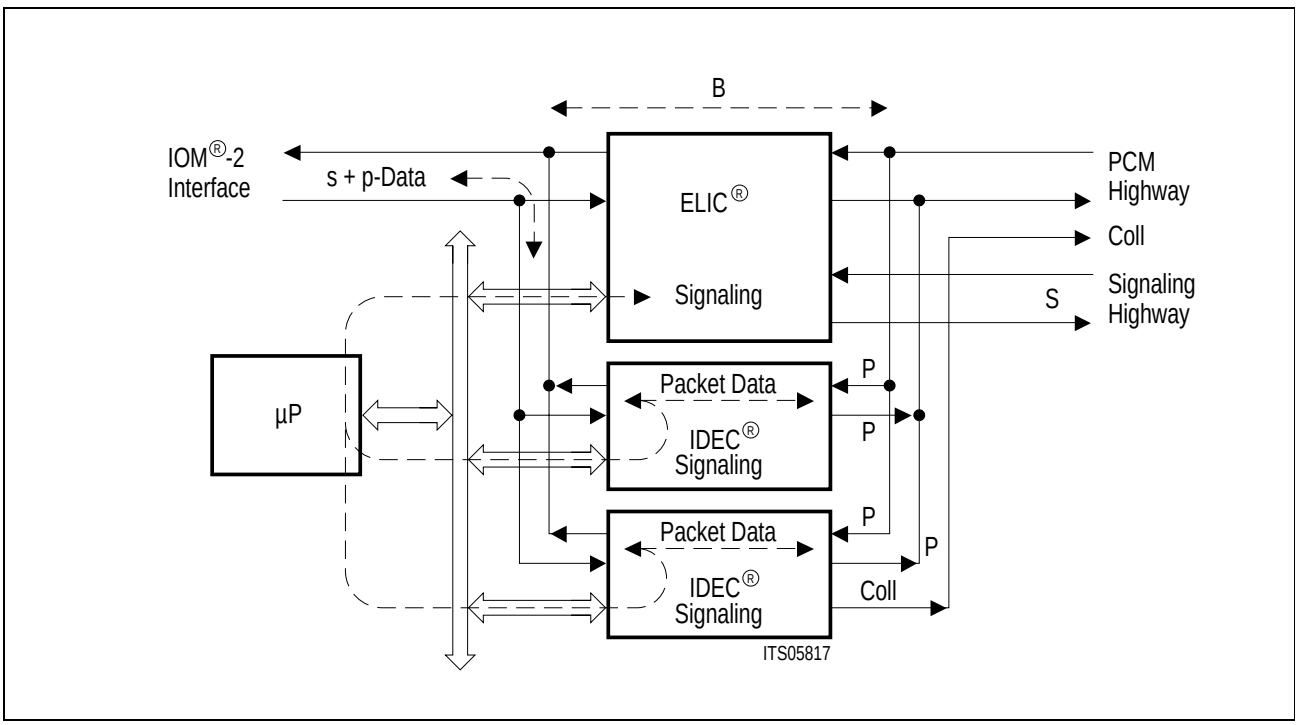



**Figure 16**
**Line Card Architecture for Mixed D-Channel Processing**

### 1.6.2 Key Systems

The ELIC is an optimal solution for key systems like a PBX. When selecting the multiplexed D-channel architecture, the ELIC covers switching, layer-1 and layer-2 control for the entire system. Together with the IOM-2 compatible Siemens transceiver circuits, a complete key system can be build with a few devices.

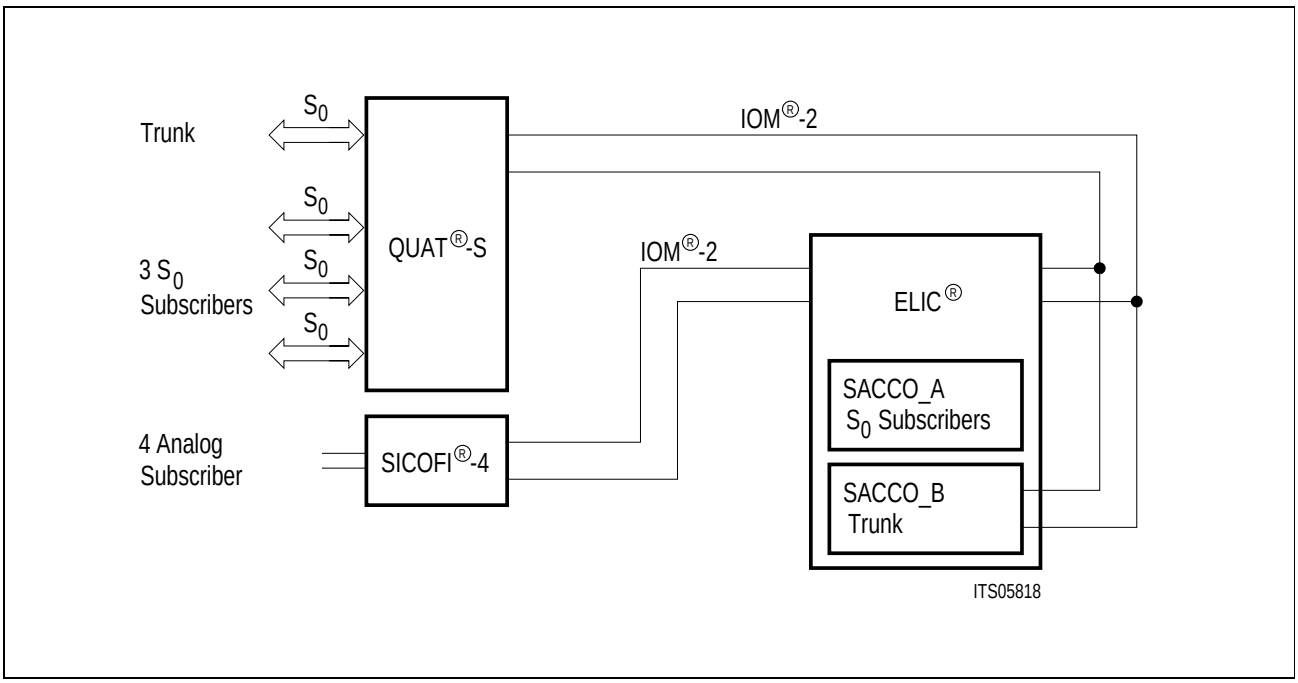



**Figure 17**
**Key System Architecture, Small Size**

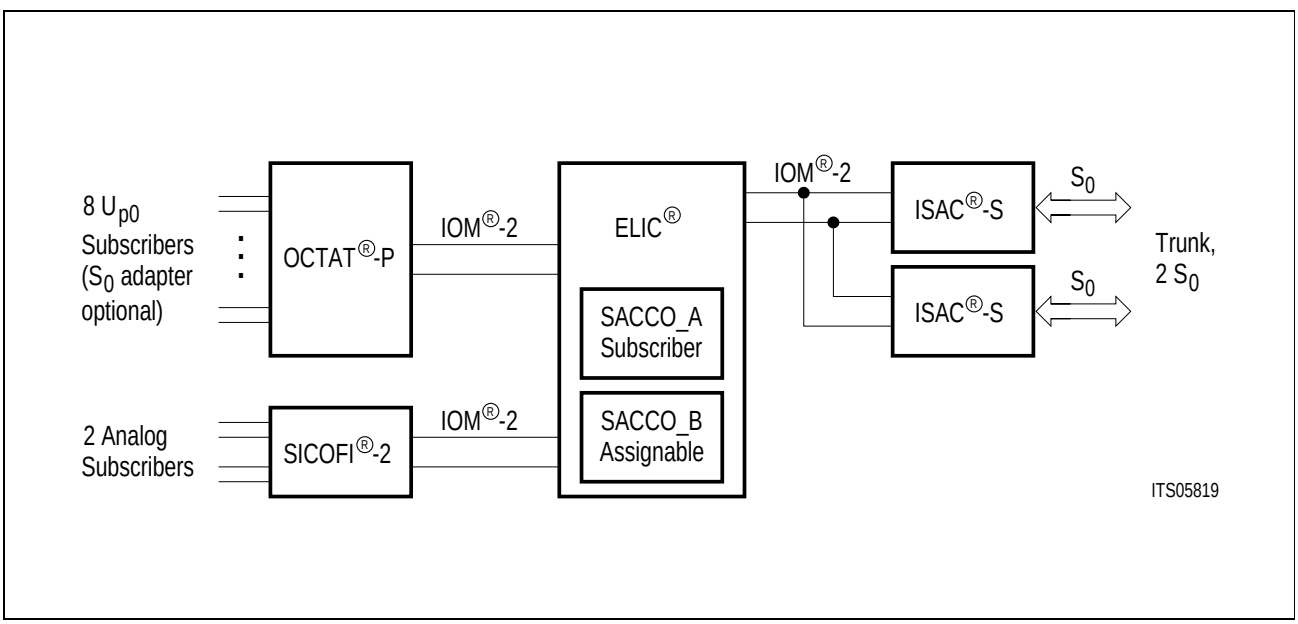



**Figure 18**
**Key System Architecture, Medium Size**

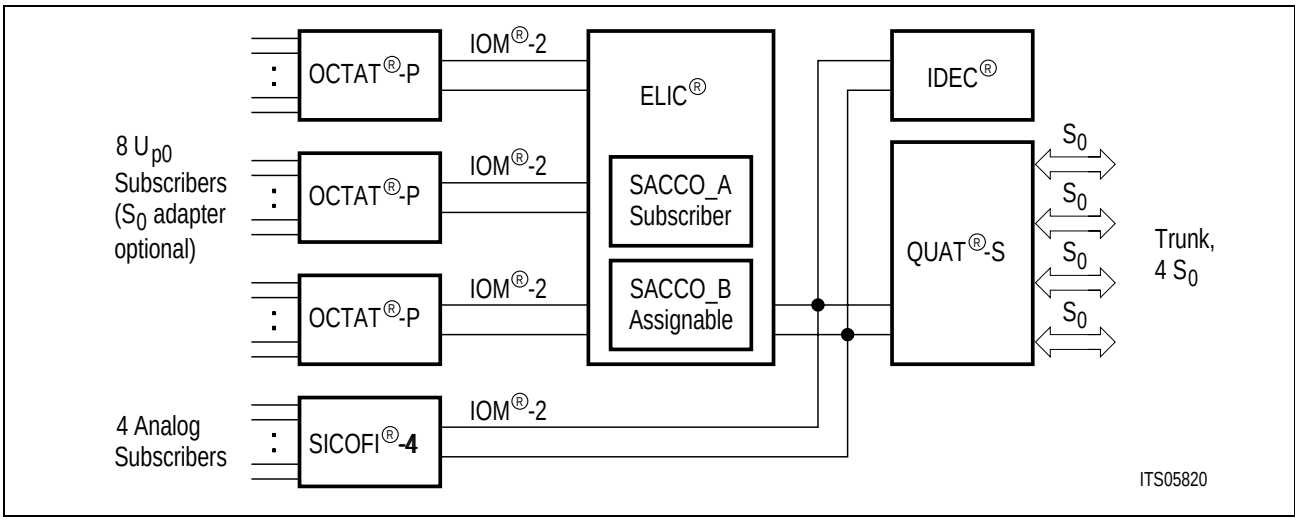



**Figure 19**
**Key System Architecture, Maximum Size**

### 1.6.3 Analog Line Card

Together with the highly flexible Siemens codec filter circuits SLICOFI, SICOFI, SICOFI-2 or SICOFI-4 the ELIC constitutes an optimized analog subscriber board architecture.

The EPIC-1 part of the ELIC handles the signalling and voice data for up to 64 subscriber channels with 64 kbit/s. The SACCO establishes the link to the group controller board.

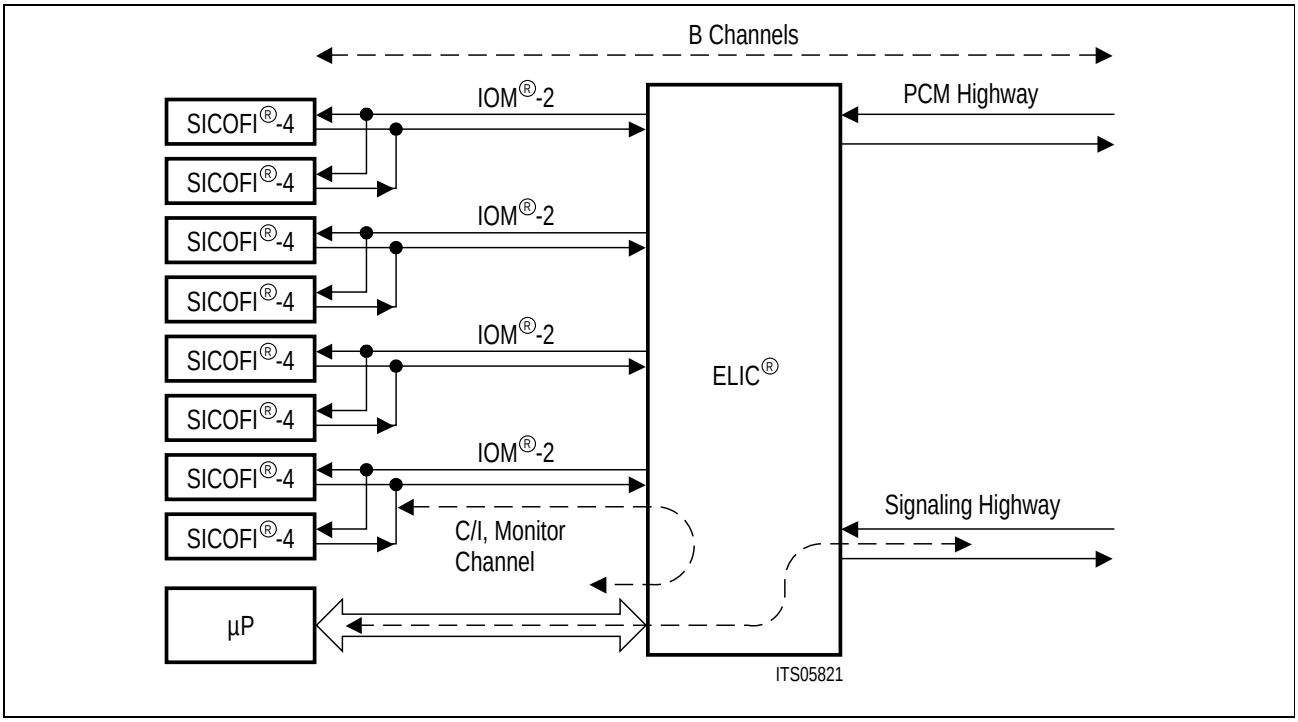



**Figure 20**
**Line Card Architecture for Analog Subscribers**

### 1.6.4 DECT Applications

#### 1.6.4.1 Adaptation of a DECT System to an Existing PBX

When adding a DECT system to an existing PBX, the line interface of the DECT system must provide the PBX with PCM-coded voice data.

Depending on the DECT controller the voice information is carried in different formats (4 bit ADPCM or 8 bit PCM ).

Therefore a base station offering 8 bit PCM coded data can be connected directly to any PBX, whereas a base station delivering 4 bit ADPCM coded data needs an ADPCM to PCM converter. Such an adapter is called Common Control Fixed Part (CCFP).

An example for a CCFP realized with the ELIC (serving up to 32 handhelds in operation at a time) is given in the **figure 21**.

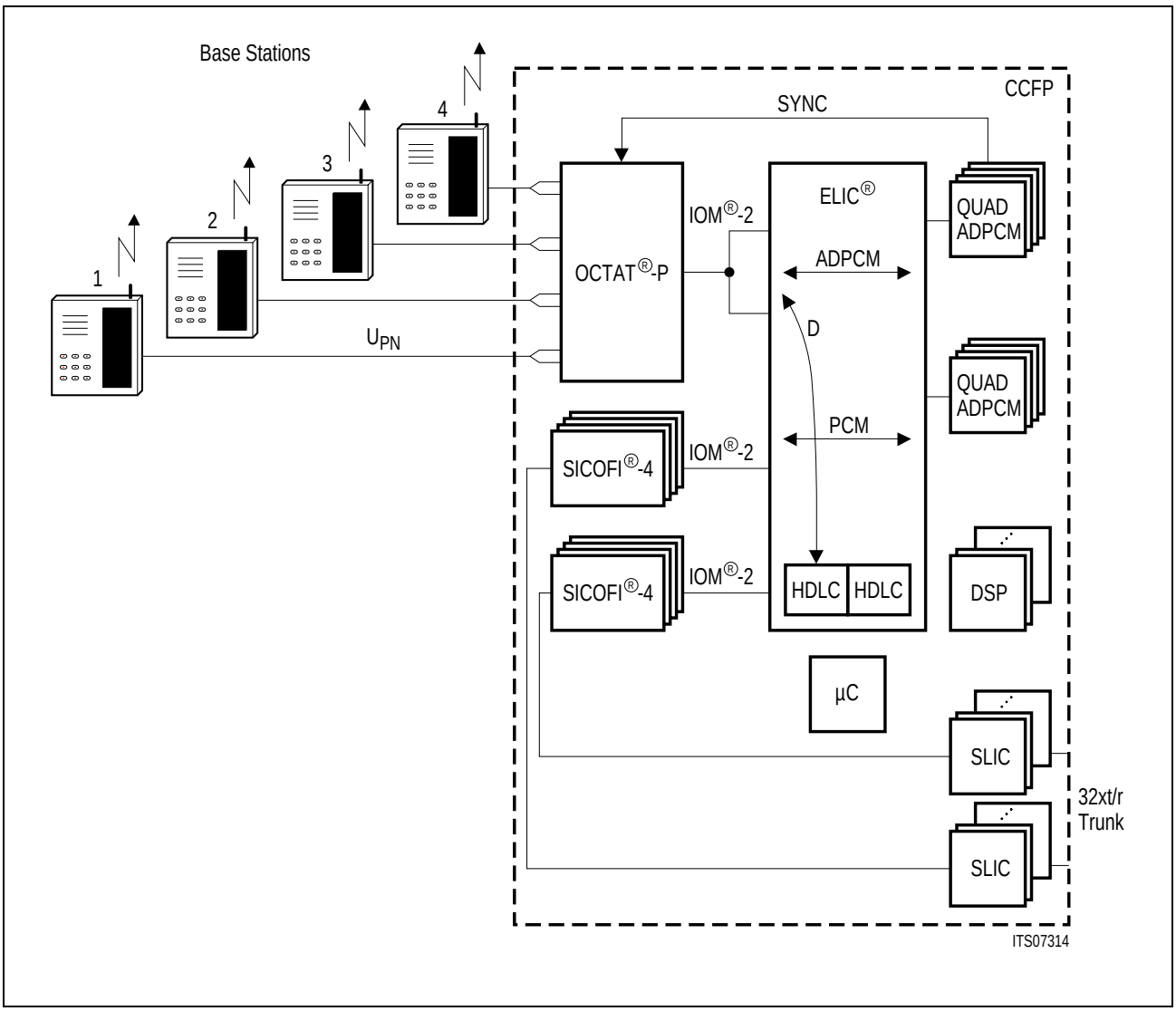



**Figure 21**
**DECT Application**

In this configuration the base stations are connected to the line interface of the CCFP via $U_{PN}$ (OCTAT-P). The 4 bit ADPCM voice channels provided by the base stations are switched (by the ELIC) to the PCM - ADPCM converter (QUAD ADPCM), expanded to an 8 bit ADPCM value and then switched (by the ELIC) to the analog trunk interface (SICOFI-4 + SLIC).

The additional DSPs are necessary, to compensate short end echoes occurring at analog nodes

The line card controller ELIC (PEB 20550) fulfills four major tasks:

- Layer-1 monitoring and controlling via IOM-2 C/I and MONITOR channel
- Signaling control (HDLC controller multiplexed to the subscribers)
- 4 bit switching of the PCM4 channels
- 8 bit switching of the PCM channels

### 1.6.4.2 DECT Line Card Design for an Existing PBX

Today most of the PBX´s have a modular design, meaning they can be extended by adding an analog or digital line card. This enables a user to integrate a DECT system into his PBX by simply inserting a DECT line card that behaves like a digital line card.

To communicate over the existing PCM highway, a PCM4 to PCM converter must be integrated onto the line card.

Compared to a digital line card a DECT line card requires additional efforts to synchronize all line cards.

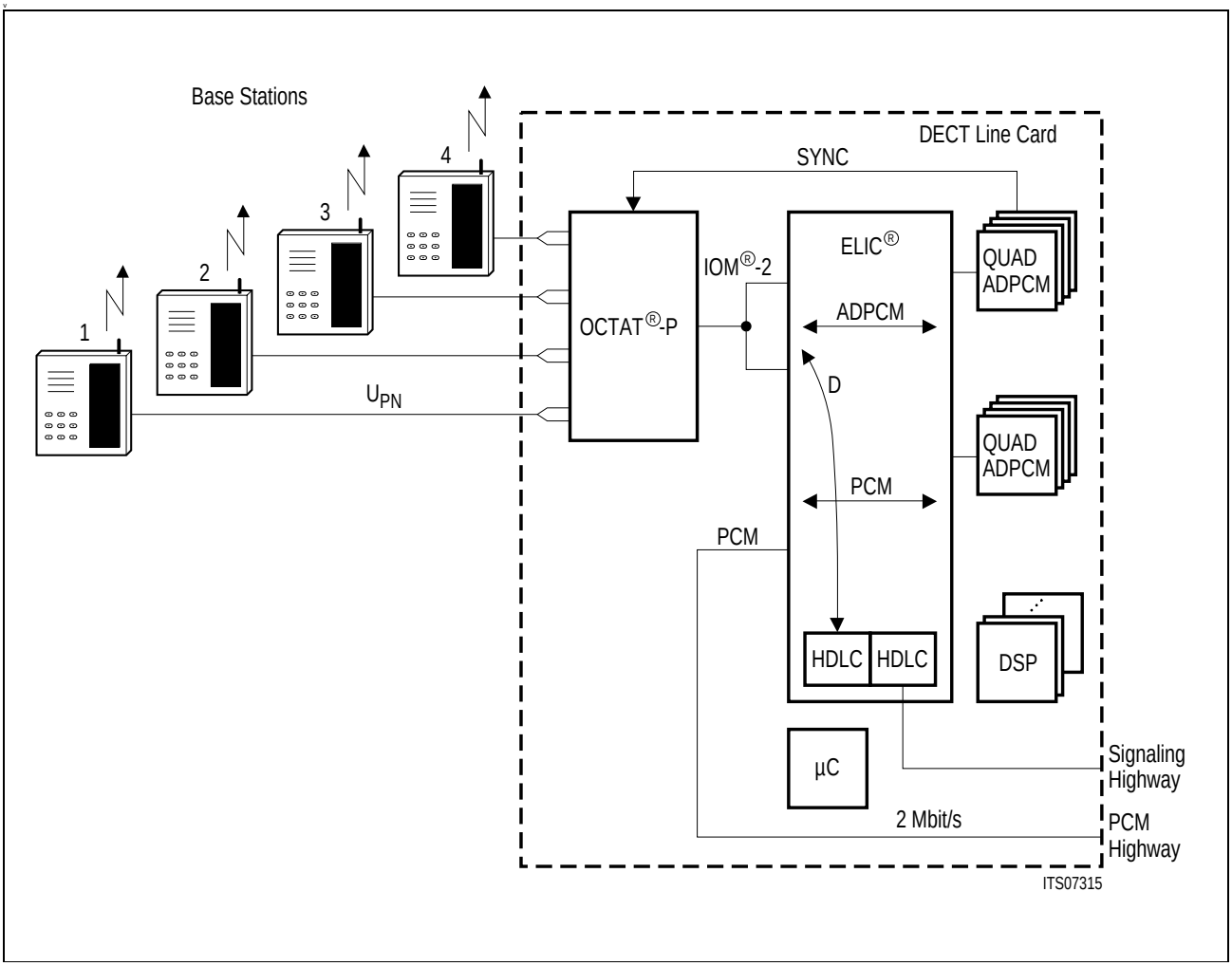



**Figure 22**
**Line Card Architecture for DECT Subscribers**

The tasks of the ELIC are:

- Layer-1 monitoring and controlling via IOM-2 C/I and MONITOR channel
- Signaling control (HDLC controller multiplexed to the subscribers)
- 4 bit switching of the PCM4 channels
- 8 bit switching of the PCM channels
- Signaling control to the group processor

# 2 Functional Description

## 2.1 General Functions and Device Architecture

The ELIC integrates the existing Siemens device PEB 2055 (EPIC-1), a two channel HDLC-Controller (SACCO: Special Application Communication Controller) with a PEB 2050 (PBC) compatible auto-mode, a D-channel arbiter, a configurable bus interface and typical system glue logic into one chip. It covers all control functions on digital and analog line cards and can be combined via IOM-2 interface with layer-1 circuits or special application devices (e.g. ADPCM/PCM-converters). Due to its flexible bus interface it fits perfectly into Siemens / Intel or Motorola microprocessor architectures.

## 2.2 Functional Blocks

### 2.2.1 Bus Interface

All registers and the FIFOs of the ELIC are accessible via the flexible bus interface supporting Siemens / Intel and Motorola type microprocessors. Depending on the register functionality a read, write or read/write access is possible.

The bus interface consists of the following elements

- Data bus, 8-bit wide, AD0-7, D0-7
- Address bus, 8-bit wide, P0.0-0.7, A0-7
- Two chip select lines, $\overline{\text{CSE}}$ and $\overline{\text{CSS}}$
- Address latch enable, ALE
- Two read/write control lines, $\overline{\text{RD}}$, DS and $\overline{\text{WR}}$, R or $\overline{\text{W}}$

The ALE-line is used to control the bus structure and interface type.

**Table 1**
**Selectable Bus Configurations**

| ALE | Interface | Bus Structure | Pin 9 | Pin 8 |
|---|---|---|---|---|
| Fixed to $V_{DD}$ | Motorola | demultiplexed | DS | R or $\overline{\text{W}}$ |
| Fixed to ground | Siemens / Intel | demultiplexed | $\overline{\text{RD}}$ | $\overline{\text{WR}}$ |
| Switching | Siemens / Intel | multiplexed | $\overline{\text{RD}}$ | $\overline{\text{WR}}$ |

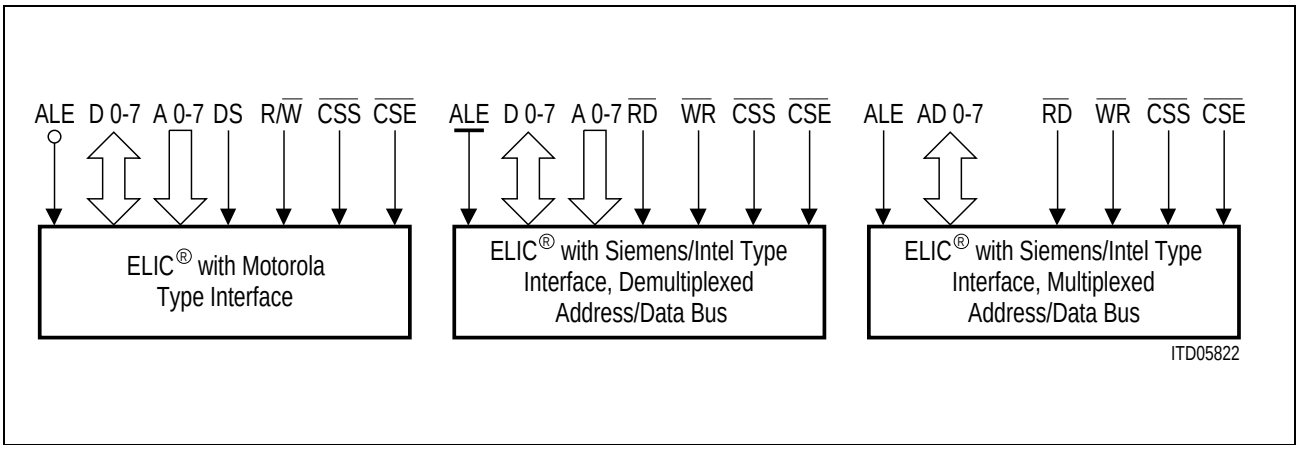



**Figure 23**
**Selectable Bus Interface Structures**

In order to simplify the use of 8- and 16-bit Siemens / Intel type CPUs, different register addresses are defined in multiplexed and demultiplexed bus mode (see **chapter 3.1**). In the multiplexed mode even addresses are used (AD0 always 0), if EMODE:DMXAD = 0. ELIC-data is always transferred in the low data byte.

### 2.2.2 Parallel Ports

The ELIC provides a 4-bit wide I/O-port. A programmable configuration register (PCON1) controls whether the individual bits are used as inputs or outputs. The port is read/written like a on chip register (PORT1).

If port 1 is to be configured as an output, please note that after reset the port is an input. The PORT1 register thus reflects the state of port 1 before it is configured as an output. If it is required that port 1 puts out a defined value immediately on being set as output, large (e.g. > 10 kΩ) pull-up or pull-down resistors should be applied.

After the port has been configured as output, its value can of course simply be set via the PORT1-register.

Additionally, when the bus interface is used in multiplexed bus mode (ALE switching), the pins A0,P0.0 - A7,P0.7 constitute a parallel 8-bit wide input port. The port is read like an on chip register (PORT0). The current values on the input port is latched with the falling edge of $\overline{RD}$, DS.

### 2.2.3 Watchdog Timer

To allow recovery from software or hardware failure, a watchdog timer is provided.

After reset the watchdog timer is disabled. When setting bit SWT in the watchdog timer control register WTC it is enabled. The only possibility to disable the watchdog timer is a ELIC-reset (power-up or RESEX).The timer period is 1024 PFS-cycles assuming that also PDC is active, i.e. a PFS of 8-kHz results in a timer period of 128 ms.

During that period, the bits WTC1 and WTC2 in the register WTC have to be written in the following sequence:

**Table 2**
**Watchdog Timer Programming**

| Activity | WTC:WTC1 | WTC:WTC2 |
|---|---|---|
| 1. | 1 | 0 |
| 2. | 0 | 1 |

**The minimum required interval between the two write accesses is 2 PDC-periods.**

When the software fails to follow these requirements, a timer overflow occurs and a IWD-interrupt is generated. Additionally an external reset indication (RESIN) is activated. The internal ELIC-status is not changed.

### 2.2.4 Reset Logic

After power-up the ELIC is latched into the "Resetting" state. A microprocessor access is not possible in the "Resetting" state. The ELIC is released from the power-up "Resetting" state when provided with PFS- and PDC-signals for 8 PFS-periods.

The ELIC can also be reset by applying a RESEX-pulse for at least 4 PDC-periods. Note that such an external RESEX has priority over a power-on reset. It is thus possible to kill the 8-frame reset duration after power-up.

Upon activation of the power supply an integrated power-up reset generator is provided. It is generated when $V_{DD}$ is in the range between 1 V and 3 V. Additionally an external reset input (RESEX) and an reset indication output (RESIN) are available.

During reset all ELIC-outputs with the exception of RESIN and TDO + DRQRA/B + DRQTA/B + SACCO are in the state high impedance. The tristate control signals of the EPIC-1 PCM-interface ($\overline{\text{TSC}}$[3:0]) $\overline{\text{TSCA}}/\overline{\text{B}}$ are not tristated during a chip reset. Instead they are high during reset, thus containing the correct tristate information for external drivers.

RESIN is set upon power up, RESEX and the expiring of the watchdog timer. It may be used as a system reset. RESIN is activated for 8 PFS-periods (assuming an active PDC-input) or it has the same pulse width as RESEX. RESEX has priority over internal

generated resets with respect to the RESIN pulse width. The activation of RESEX causes an immediate activation of RESIN. Upon the deactivation of RESEX however, RESIN is deactivated only with the next rising PDC-edge. A PFS-frequency of 8-kHz results in a RESIN-period of 1 ms.

When setting bit VNSR:SWRX RESIN is also activated but the ELIC itself is not reset. This feature supports a proper reset procedure for devices which require dedicated clocking during reset. The sequence required is as follows:

1. Initialize EPIC-1 for a timer interrupt
2. Set bit VNSR:SWRX to "1", RESIN is activated
3. When the timer interrupt occurs, RESIN is deactivated
4. Set bit VNSR:SWRX to "0"
5. Read ISTA_E, in order to deactivate timer interrupt

**Table 3**
**Reset Activities**

| | Internal ELIC Reset | RESIN Activation | RESIN Pulse Width |
|---|---|---|---|
| Power up | X | X | 8 PFS |
| Watchdog timer under flow | – | X | 8 PFS |
| External reset (RESEX) | X | X | RESEX |
| Setting of bit SWRX | – | X | Programmable |

When $V_{DD}$ drops under normal operation the reset logic has the following behavior:

**Table 4**
**Behavior of the Reset Logic in the Case of Voltage Drop**

| $V_{DD}$ | Behavior |
|---|---|
| > 3 V | No internal reset, no RESIN |
| < 1 V | Internal reset and RESIN after $V_{DD}$ goes up again |
| 1 V ≤ $V_{DD}$ ≤ 3 V | Not defined |

*Note: The power-up reset generator must not be used as a supply voltage control element.*

### 2.2.5 Boundary Scan Support

The ELIC provides fully IEEE Std. 1149.1 compatible boundary scan support consisting of:

– a complete boundary scan
– a test access port controller (TAP)
– four dedicated pins (TCK, TMS, TDI, TDO)
– a 32-bit IDCODE-register

#### 2.2.5.1 Boundary Scan

All ELIC-pins except power supply and ground are included in the boundary scan. Depending on the pin functionality one, two or three boundary scan cells are provided.

**Table 5**
**Boundary Scan Cell Types**

| Pin Type | Number of Boundary Scan Cells | Usage |
|---|---|---|
| Input | 1 | Input |
| Output | 2 | Output, enable |
| I/O | 3 | Input, output, enable |

When the TAP-controller is in the appropriate mode data is shifted into/out of the boundary scan via the pins TDI/TDO using the 6.25-MHz clock on pin TCK.

The ELIC-pins are included in the following sequence in the boundary scan:

**Table 6**
**Boundary Scan Sequence**

| Boundary Scan Number TDI → | Pin Number | Pin Name | Type | Number of Scan Cells | Default Value |
|---|---|---|---|---|---|
| 1 | 77 | P0.0,A0 | I | 1 | 0 |
| 2 | 78 | P0.1,A1 | I | 1 | 0 |
| 3 | 79 | P0.2,A2 | I | 1 | 0 |
| 4 | 80 | P0.3,A3 | I | 1 | 0 |
| 5 | 1 | P0.4,A4 | I | 1 | 0 |
| 6 | 2 | P0.5,A5 | I | 1 | 0 |
| 7 | 3 | P0.6,A6 | I | 1 | 0 |

**Table 6**
**Boundary Scan Sequence** (cont'd)

| Boundary Scan Number TDI → | Pin Number | Pin Name | Type | Number of Scan Cells | Default Value |
|---|---|---|---|---|---|
| 8 | 4 | P0.7,A7 | I | 1 | 0 |
| 9 | 5 | $\overline{INT}$ | O | 2 | 01<br>10 for V1.3 |
| 10 | 6 | $\overline{CSE}$ | I | 1 | 0 |
| 11 | 7 | $\overline{CSS}$ | I | 1 | 0 |
| 12 | 8 | $\overline{WR}$, R or $\overline{W}$ | I | 1 | 0 |
| 13 | 9 | $\overline{RD}$, DS | I | 1 | 0 |
| 14 | 10 | ALE | I | 1 | 0 |
| 15 | 12 | AD0,D0 | I/O | 3 | 000 |
| 16 | 13 | AD1,D1 | I/O | 3 | 000 |
| 17 | 14 | AD2,D2 | I/O | 3 | 100 |
| 18 | 15 | AD3,D3 | I/O | 3 | 110 |
| 19 | 16 | AD4,D4 | I/O | 3 | 000 |
| 20 | 17 | AD5,D5 | I/O | 3 | 100 |
| 21 | 18 | AD6,D6 | I/O | 3 | 000 |
| 22 | 19 | AD7,D7 | I/O | 3 | 110 |
| 23 | 21 | P1.0 | I/O | 3 | 000 |
| 24 | 22 | P1.1 | I/O | 3 | 000 |
| 25 | 23 | P1.2 | I/O | 3 | 000 |
| 26 | 24 | P1.3 | I/O | 3 | 000 |
| 27 | 25 | RESIN | O | 2 | 00 |
| 28 | 26 | RESEX | I | 1 | 0 |
| 29 | 27 | FSC | I/O | 3 | 000 |
| 30 | 28 | DCL | I/O | 3 | 000 |
| 31 | 29 | DU0 | I/O | 3 | 000 |
| 32 | 30 | DU1 | I/O | 3 | 000 |
| 33 | 32 | DU2 | I/O | 3 | 000 |
| 34 | 33 | DU3 | I/O | 3 | 000 |

**Table 6**
**Boundary Scan Sequence** (cont'd)

| Boundary Scan Number TDI → | Pin Number | Pin Name | Type | Number of Scan Cells | Default Value |
|---|---|---|---|---|---|
| 35 | 34 | DD0 | I/O | 3 | 000 |
| 36 | 35 | DD1 | I/O | 3 | 000 |
| 37 | 36 | DD2 | I/O | 3 | 000 |
| 38 | 37 | DD3 | I/O | 3 | 000 |
| 39 | 38 | $\overline{\text{DACKB}}$ | I | 1 | 0 |
| 40 | 39 | $\overline{\text{DACKA}}$ | I | 1 | 0 |
| 41 | 40 | DRQRB | O | 2 | 00 |
| 42 | 41 | DRQTB | O | 2 | 00 |
| 43 | 42 | DRQAR | O | 2 | 00 |
| 44 | 43 | DRQTA | O | 2 | 00 |
| 45 | 44 | RxDA | I | 1 | 0 |
| 46 | 45 | CxDA | I | 1 | 0 |
| 47 | 46 | TxDA | O | 2 | 00 |
| 48 | 47 | $\overline{\text{TSCA}}$ | O | 2 | 00 |
| 49 | 48 | HDCA | I | 1 | 0 |
| 50 | 49 | HFSA | I | 1 | 0 |
| 51 | 50 | HFSB | I | 1 | 0 |
| 52 | 52 | HDCB | I | 1 | 0 |
| 53 | 53 | $\overline{\text{TSCB}}$ | O | 2 | 00 |
| 54 | 54 | TxDB | O | 2 | 00 |
| 55 | 55 | CxDB | I | 1 | 0 |
| 56 | 56 | RxDB | I | 1 | 0 |
| 57 | 58 | RxD3 | I | 1 | 0 |
| 58 | 59 | RxD2 | I | 1 | 0 |
| 59 | 60 | RxD1 | I | 1 | 0 |
| 60 | 61 | RxD0 | I | 1 | 0 |
| 61 | 62 | $\overline{\text{TSC0}}$ | O | 2 | 00 |
| 62 | 63 | TxD0 | O | 2 | 00 |

**Table 6**
**Boundary Scan Sequence** (cont'd)

| Boundary Scan Number TDI → | Pin Number | Pin Name | Type | Number of Scan Cells | Default Value |
|---|---|---|---|---|---|
| 63 | 64 | $\overline{\text{TSC1}}$ | O | 2 | 00 |
| 64 | 65 | TxD1 | O | 2 | 00 |
| 65 | 66 | $\overline{\text{TSC2}}$ | O | 2 | 00 |
| 66 | 67 | TxD2 | O | 2 | 00 |
| 67 | 68 | $\overline{\text{TSC3}}$ | O | 2 | 00 |
| 68 | 69 | TxD3 | O | 2 | 00 |
| 69 | 70 | PFS | I | 1 | 0 |
| 70 | 71 | PDC | I | 1 | 0 |

### 2.2.5.2 TAP-Controller

The Test Access Port (TAP) controller implements the state machine defined in the JTAG-standard: IEEE Std. 1149.1. Transitions on the pin TMS cause the TAP-controller to perform a state change. Following the standard definition five instructions are executable.

**Table 7**
**TAP-Controller Instructions**

| Code | Instruction | Function |
|---|---|---|
| 000 | EXTEST | External testing |
| 001 | INTEST | Internal testing |
| 010 | SAMPLE/PRELOAD | Snap-shot testing |
| 011 | IDCODE | Reading ID-code |
| 111 | BYPASS | Bypass operation |
| Others | – | Bypass operation |

**EXTEST** is used to examine the board interconnections.

When the TAP-controller is in the state "update DR", all output pins are updated with the falling edge of TCK. When it has entered state "capture DR" the levels of all input pins are latched with the rising edge of TCK. The in/out shifting of the scan vectors is typically done using the instruction SAMPLE/PRELOAD.

**INTEST** supports internal chip testing.

When the TAP-controller is in the state "update DR", all inputs are updated internally with the falling edge of TCK. When it has entered state "capture DR" the levels of all outputs are latched with the rising edge of TCK. The in/out shifting of the scan vectors is typically done using the instruction SAMPLE/PRELOAD.

*Note: 001 (INTEST) is the default value of the instruction register.*

**SAMPLE/PRELOAD** provides a snap-shot of the pin level during normal operation or is used to preload (TDI)/shift out (TDO) the boundary scan with a test vector. Both activities are transparent to the system functionality.

**IDCODE**, the 32-bit identification register is serially read out via TDO. It contains the version number (4 bit), the device code (16 bit) and the manufacturer code (11 bits). The LSB is fixed to "1".

| | | | | | |
|---|---|---|---|---|---|
| TDI → | 00**01** | 0000 0000 0001 0011 | 0000 1000 001 | 1 | → TDO |
| | 00**10** | 0000 0000 0001 0011 | 0000 1000 001 | 1 | for V1.3 |

*Note: In the state "test logic reset" the code "011" is loaded into the instruction code register*

**BYPASS**, a bit entering TDI is shifted to TDO after one TCK-clock cycle.

### 2.2.6 EPIC®-1

The EPIC-1 is fully compatible to the Siemens PEB 2055 (EPIC-1, Version A3). It includes the following functional enhancements:

– Direct access to all registers also in demultiplexed mode
– PCM-mode 3
– Software activation of external reset
– Error correction
– Additional clock shift features PCM (register PCSR)

For detailed information refer to **appendix 9.1**.

#### 2.2.6.1 PCM-Interface

The PCM-interface formats the data transmitted or received at the PCM-highways. It can be configured as one (max. 8192 kbit/s), two (max. 4096 kbit/s) or four (max. 2048 kbit/s) PCM-ports, consisting each of a data receive (RxD#), a data transmit (TxD#) and an output tristate indication line ($\overline{\text{TSC}}$#).

Port configuration, data rates, clock shift and sampling conditions are programmable.

The newly implemented PCM-mode 3 is similar to mode 1 (two PCM-highways). Unlike mode 1 the pins TxD1, TxD3 are not tristated but drive the inverted values of TxD0, TxD2.

### 2.2.6.2 Configurable Interface

In order to optimize the on-board interchip communication, a very flexible serial interface is available. It formats the data transmitted or received at the DDn-, DUn- or SIPn-lines. Although it is typically used in IOM-2 or SLD-configuration to connect layer-1 devices, application specific frame structures can be defined (e.g. to interface ADPCM-converters or maintenance blocks).

### 2.2.6.3 Memory Structure and Switching

The memory block of the EPIC-1 performs the switching functionality.
It consists of four sub blocks:

– Upstream data memory
– Downstream data memory
– Upstream control memory
– Downstream control memory.

The PCM-interface reads periodically from the upstream (writes periodically to the downstream) data memory (cyclical access), see **figure 24.**

The CFI reads periodically the control memory and uses the extracted values as a pointers to write to the upstream (read from the downstream) data memory (random access). In the case of C/I- or signaling channel applications the corresponding data is stored in the control memory. In order to select the application of choice, the control memory provides a code portion.

The control memory is accessible via the µP-interface. In order to establish a connection between CFI time slot A and PCM-interface time slot B, the B-pointer has to be loaded into the control memory location A.

### 2.2.6.4 Pre-processed Channels, Layer-1 Support

The EPIC-1 supports the monitor/feature control and control/signaling channels according to SLD- or IOM-2 interface protocol.

The monitor handler controls the data flow on the monitor/feature control channel either with or without active handshake protocol. To reduce the dynamic load of the CPU a 16-byte transmit/receive FIFO is provided.

The signaling handler supports different schemes (D-channel + C/I-channel, 6-bit signaling, 8-bit signaling).

In downstream direction the relevant content of the control memory is transmitted in the appropriate CFI time slot. In the case of centralized ISDN D-channel handling, a 16-kbit/s D-channel received at the PCM-interface is included.

In upstream direction the signaling handler monitors the received data. Upon a change it generates an interrupt, the channel address is stored in the 9-byte deep C/I FIFO and the actual value is stored in the control memory. In 6-bit and 8-bit signaling schemes a double last look check is provided.

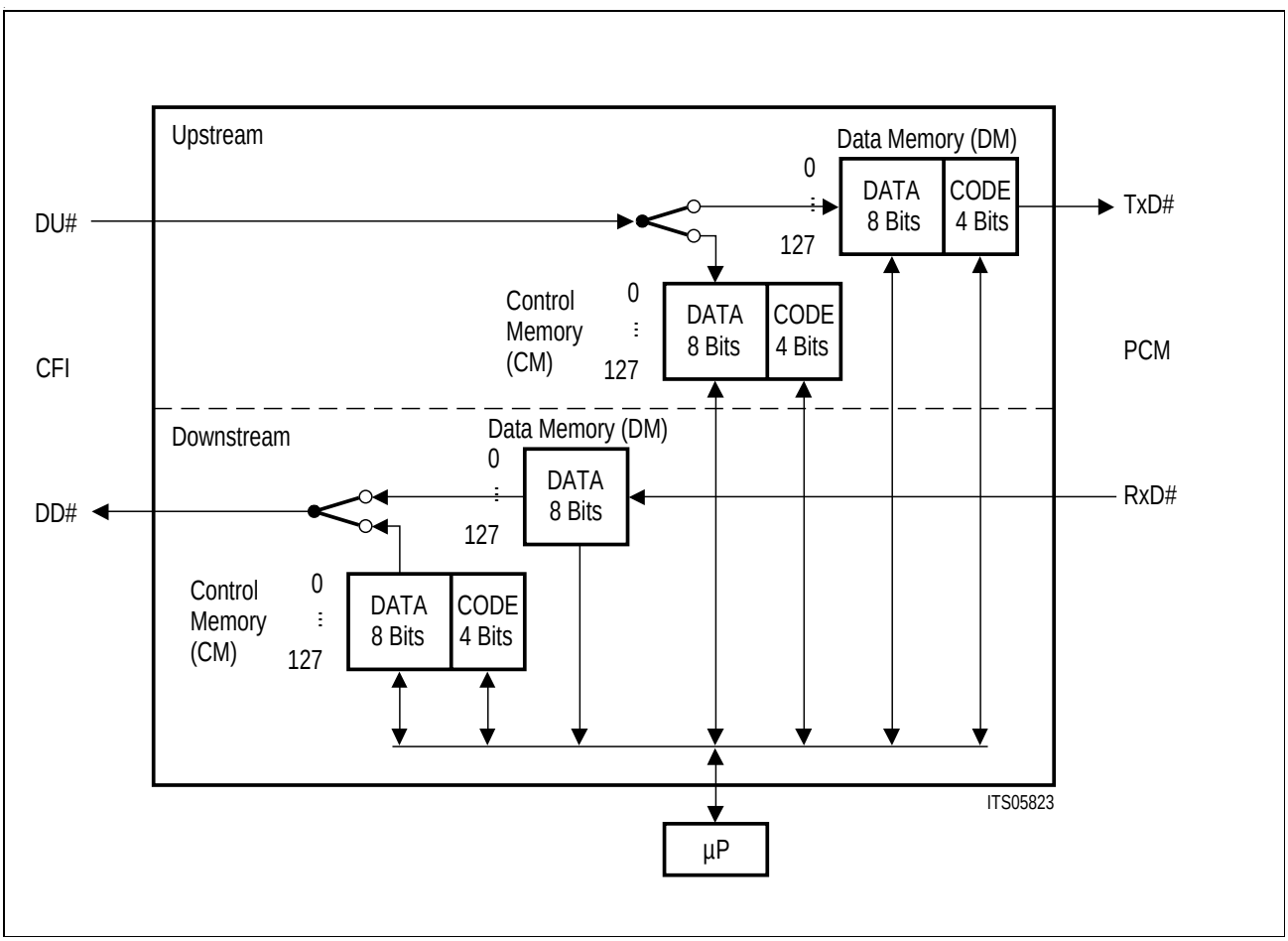



**Figure 24**
**EPIC®-1 Memory Structure**

#### 2.2.6.5 Special Functions

– Synchronous transfer.
  This utility allows the synchronous µP-access to two independent channels on the PCM- or CFI-interface. Interrupts are generated to indicate the appropriate access windows.
– 7-bit hardware timer.
  The timer can be used to cyclically interrupt the CPU, to determine the double last look period, to generate a proper CFI-multiframe synchronization signal or to generate a defined RESIN pulse width.
– Frame length checking.
  The PFS-period is internally checked against the programmed frame length.
– Alternative input functions.
  In PCM-mode 1 and 2, the unused ports can be used for redundancy purposes. In these modes, for every active input port a second input port exists which can be connected to a redundant PCM-line. Additionally the two lines are checked for mismatches.

### 2.2.7 SACCO

The SACCO (Special Application Communication Controller) is a high level serial communication controller consisting of two independent HDLC-channels (A + B). It is a derivative product of the Siemens SAB 82525 (HSCX).

The SACCO essentially reduces the hardware and software overhead for serial synchronous communication. SACCO channel A can be multiplexed by the D-channel arbiter to serve multiple subscribers.

In the following section one SACCO channel is described referring to as "SACCO".

#### 2.2.7.1 Block Diagram

The SACCO (one channel) provides two independent 64-byte FIFOs for receive and transmit direction and a sophisticated protocol support. It is optimized for line card applications in digital exchange systems and offers special features to support:

– Communication between a line card and a group controller
– Communication between terminal equipment and a line card

The SACCO consists of the following logical blocks:

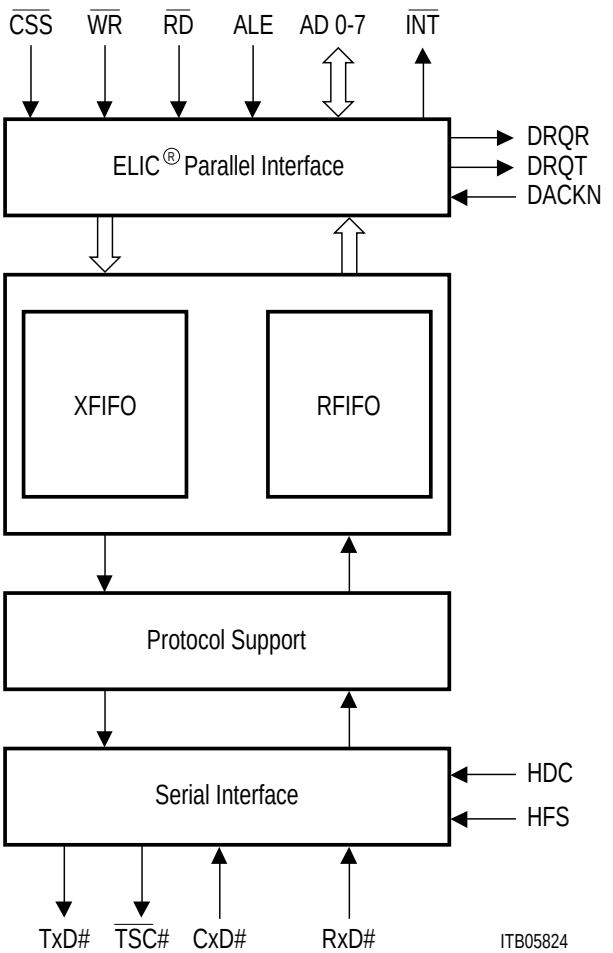



**Figure 25**
**SACCO-Block Diagram (one channel)**

### 2.2.7.2 Parallel Interface

All registers and the FIFOs are accessible via the ELIC parallel µP-interface. The chip select signal $\overline{CSS}$ selects the SACCO for read/write access. The FIFOs allocate an address space of 32 bytes each. The data in the FIFOs can be managed by the CPU- or a DMA-controller.

To enable the use of block move instructions, the top of FIFO-byte is selected by any address in the reserved range.

### Interrupts

The SACCO indicates special events by issuing an interrupt request. The cause of a request can be determined by reading the interrupt status register ISTA_A/B or EXIR_A/B. The related register is flagged in the top level ISTA (refer to **figure 46**).

Three indications are available in ISTA_A/B, another five in the extended interrupt register EXIR_A/B. An interrupt which is masked in the MASK_A/B is not indicated in the top level register and the $\overline{\text{INT}}$-line is not activated. The interrupt is also not visible in the local registers ISTA_A/B but remains stored internally and will be indicated again when the corresponding MASK_A/B-bit is reset.

The SACCO-interrupt sources can be splitted in three logical groups:

- Receive interrupts (RFS, RPF, RME, EHC)
- Transmit interrupts (XPR, XMR)
- Special condition interrupts (XDU/EXE, RFO)

For further information refer to **chapter 3.6.1** (**Data Transmission in Interrupt Mode**) and **chapter 3.6.3** (**Data Reception in Interrupt Mode**).

### DMA-Interface

To support efficient data exchange between system memory and the FIFOs an additional DMA-interface is provided. The FIFOs have separate DMA-request lines (DRQRA/B for RFIFO, DRQTA/B for XFIFO) and a common DMA-acknowledge input. The DMA-controller has to operate in the level triggered, demand transfer mode. If the DMA-controller provides a DMA-acknowledge signal, each bus cycle implicitly selects the top of FIFO and neither address nor chip select is evaluated. If no $\overline{\text{DACK}}$ signal is supplied, normal read/write operations (providing addresses) must be performed (memory to memory transfer).

The SACCO activates the DRQT/R-lines as long as data transfers are needed from/to the specific FIFOs.

A special timing scheme is implemented to guarantee safe DMA-transfers regardless of DMA-controller speed.

If in transmit direction a DMA-transfer of n bytes is necessary (n < 32 or the remainder of a long message), the DRQT-pin is active up to the rising edge of $\overline{\text{WR}}$ of DMA-transfer (n-1). If $n \geq 32$ the same behavior applies additionally to transfers 31, 63, …, ((k $\times$ 32) 1). DRQT is activated again with the next rising edge of $\overline{\text{DACK}}$ (or $\overline{\text{CSS}}$), if there are further bytes to transfer (**figure 27**). When a fast DMA-controller is used (> 16 MHz), byte n (or bytes k $\times$ 32) will be transferred before DRQT is deactivated from the SACCO. In this case pin DRQT is not activated any more up to the next block transfer (**figure 26**).

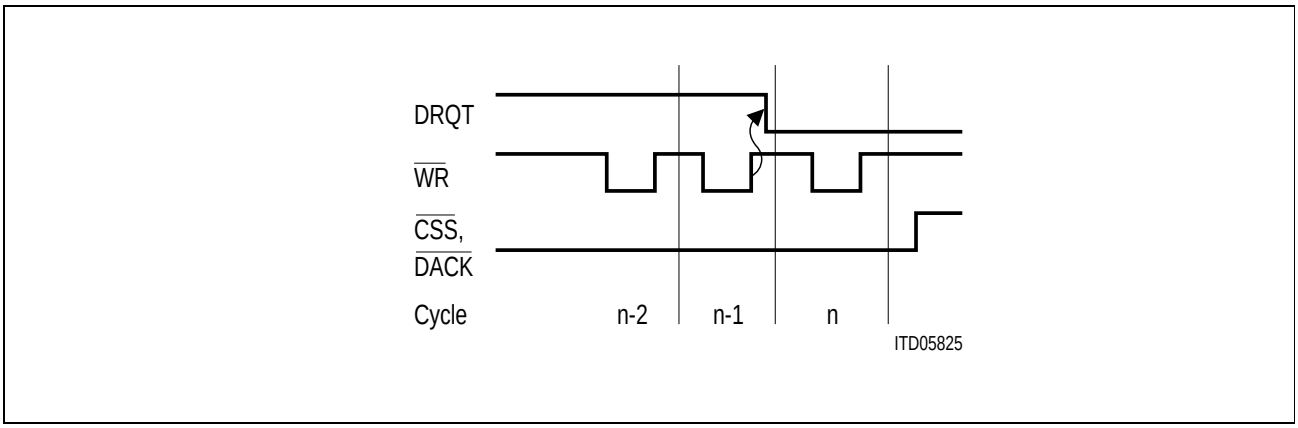



**Figure 26**
**Timing Diagram for DMA-Transfers (fast) Transmit (n < 32, remainder of a long message or n = k × 32)**

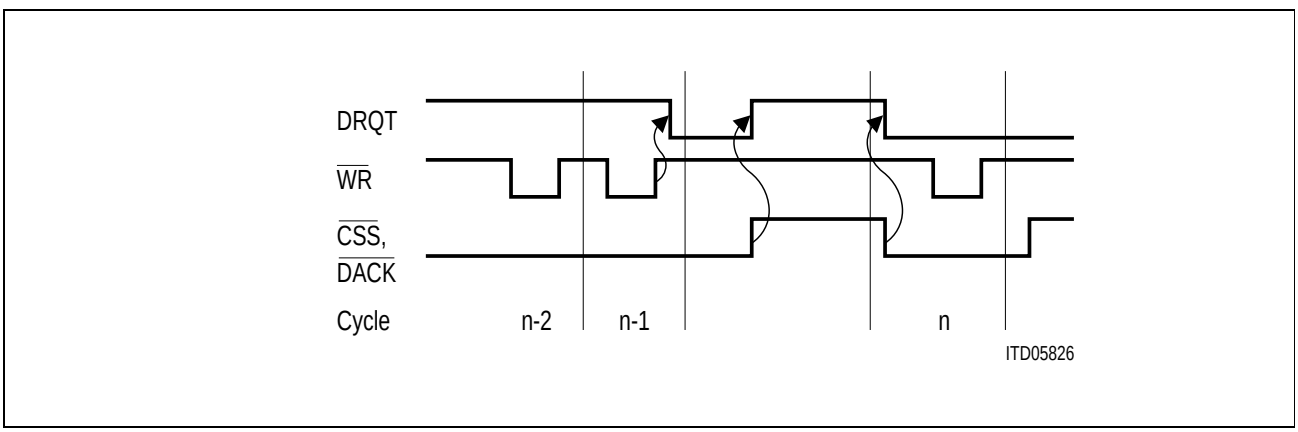



**Figure 27**
**Timing Diagram for DMA-Transfers (slow) Transmit (n < 32, remainder of a long message or n = k × 32)**

In receive direction the behavior of pin DRQR is implemented correspondingly. If k × 32 bytes are transferred, pin DRQR is deactivated with the rising edge of $\overline{RD}$ of DMA-transfer ((k × 32)  1) and it is activated again with the next rising edge of $\overline{DACK}$ (or $\overline{CSS}$), if there are further bytes to transfer (**figure 29**). When a fast DMA-controller is used (> 16 MHz), byte n (or bytes k × 32) will be transferred immediately (**figure 28**).

However, if 4, 8, 16 or 32 bytes have to be transferred (only these discrete values are possible in receive direction), DRQR is deactivated with the falling edge of $\overline{RD}$ (**figure 30**).

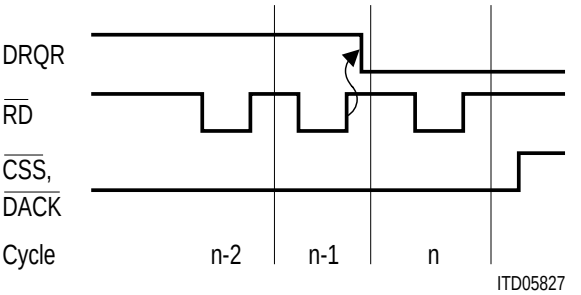



**Figure 28**
**Timing Diagram for DMA-Transfer (fast) Receive (n = k × 32)**

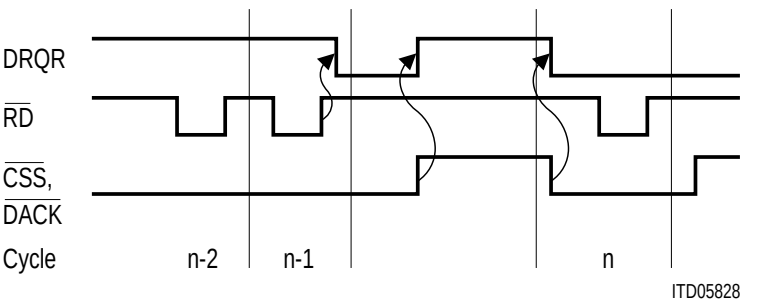



**Figure 29**
**Timing Diagram for DMA-Transfers (slow) Receive (n = k × 32)**

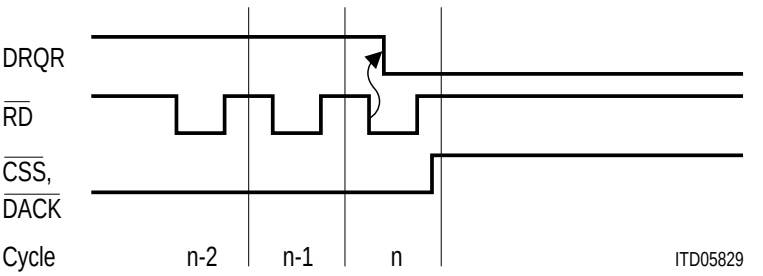



**Figure 30**
**Timing Diagram for DMA-Transfers (slow or fast) Receive (n = 4, 8 or 16)**

Generally it is the responsibility of the DMA-controller to perform the correct bus cycles as long as a request line is active.

For further information refer to **chapter 3.6.2** (**Data Transmission in DMA-Mode**) and **chapter 3.6.4** (**Data Reception in DMA-Mode**).

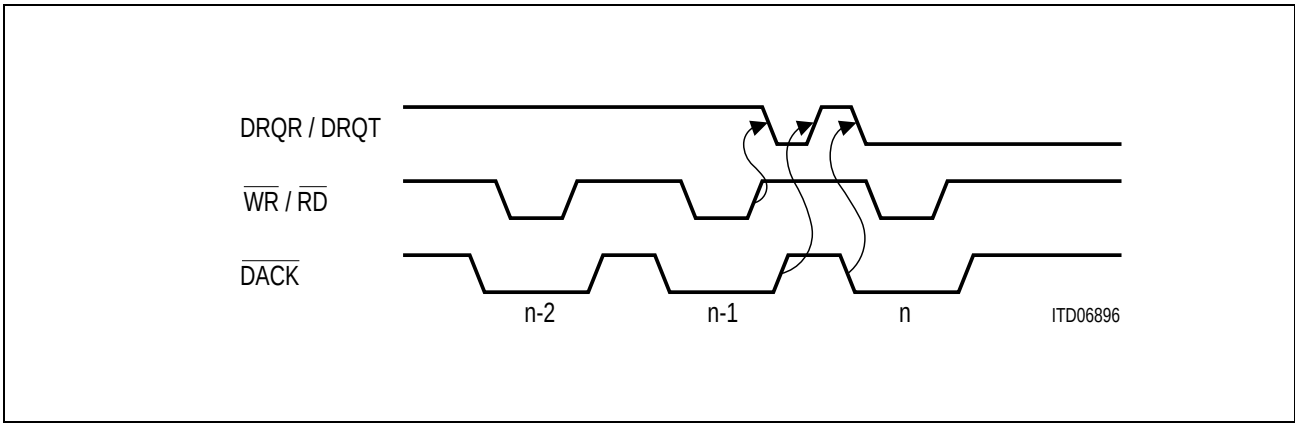



**Figure 31**
**DMA-Transfers with Pulsed DACK (read or write)**

If a pulsed DACK-signal is used the DRQR/DRQT-signal will be deactivated with the rising edge of RD/WR-operation (n-1) but activated again with the following rising edge of DACK. With the next falling edge of DACK (DACK 'n') it will be deactivated again (see **figure 31**).

This behavior might cause a short negative pulse on the DRQR/DRQT-line depending on the timing of DACK vs. RD/WR.

### 2.2.7.3 FIFO-Structure

Two independent 64-byte deep FIFOs for transmit and receive direction are provided. They enable an intermediate storage of data between the serial and the parallel (CPU) interface. The FIFOs are divided into two halves of 32 bytes each, where only one half is accessible by the CPU- or DMA-controller.

#### Receive FIFO

The receive FIFO (RFIFO) is organized in two parts of 32 bytes each, of which only one part is accessible for the CPU.

When a frame with **up to 64 bytes** is received, the complete frame may be stored in RFIFO. After the first 32 bytes have been received, the SACCO prompts to read the data block by means of interrupt or DMA-request (RPF-interrupt or activation of DRQR-line).

The data block remains in the RFIFO until a confirmation is given to the SACCO-acknowledging the reception of the data. This confirmation is either a RMC- (Receive Message Complete) command in interrupt mode or it is implicitly achieved in DMA-mode after 32 bytes have been read. As a result it is possible in interrupt mode to read out the data block any number of times until the RMC-command is executed. Upon the confirmation the second data block is shifted into the accessible RFIFO-part and an

RME-interrupt is generated. The configuration of the RFIFO prior to and after acknowledgment is shown in **figure 32** (**left**). If frames longer than 64 bytes are received, the SACCO will repeatedly prompt to read out 32-byte data blocks via interrupt or DMA.

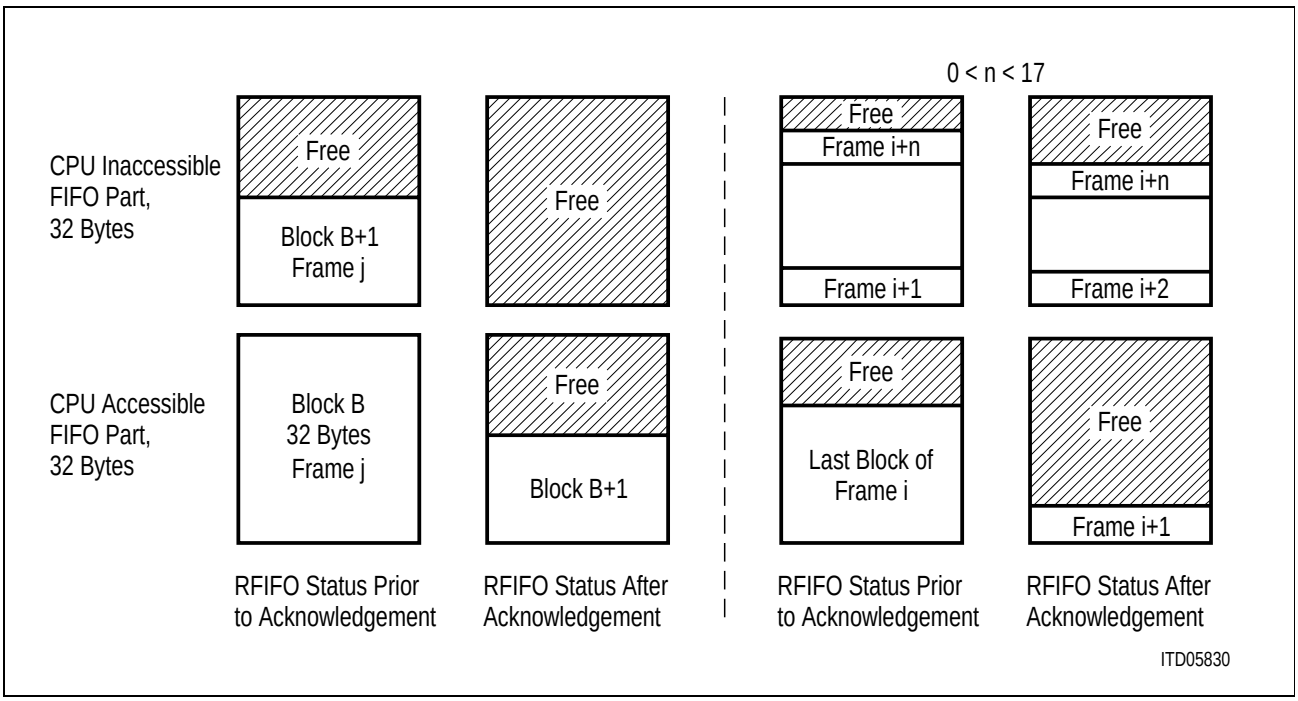



**Figure 32**
**Frame Storage in RFIFO (single frame / multiple frames)**

In the case of **several shorter frames**, up to 17 frames may be stored in the RFIFO. Nevertheless, only one frame is stored in the CPU accessible part of the RFIFO. E.g., if frame i (or the last part of frame i) is stored in the accessible RFIFO-part, up to 16 short frames may be stored in the other half (i + 1, i + 2, …, i + n, $n \leq 16$). This behavior is illustrated in **figure 32** (**right**).

*Note: After every frame a receive status byte is appended, specifying the status of the frame (e.g. if the CRC-check is o.k.).*

When using the DMA-mode, the SACCO requests fixed size block transfers (4, 8, 16 or 32 bytes). The valid byte count is determined by reading the registers RBCH, RBCL following the RME-interrupt.

### Transmit FIFO

The transmit FIFO (XFIFO) provides a $2 \times 32$ bytes capability to intermediately store transmit data.

In interrupt mode the user loads the data and then executes a transmit command.
When the frames are longer than 32 bytes, a XPR-interrupt is issued as soon as the accessible XFIFO-part is available again.

The status of the bit MODE:CFT (continuous frame transmission) defines whether a new frame can be loaded as soon as the XFIFO is available or after the current transmission was terminated.

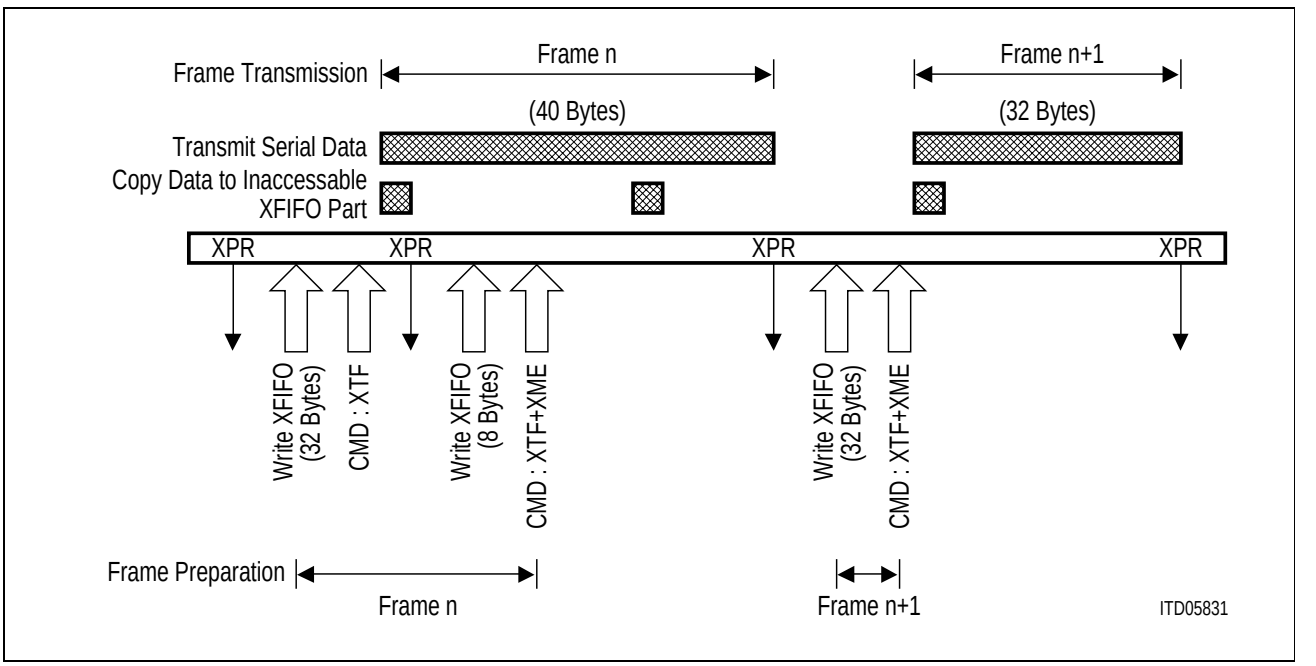



**Figure 33**
**XFIFO Loading, Continuous Frame Transmission Disabled (CFT = 0)**

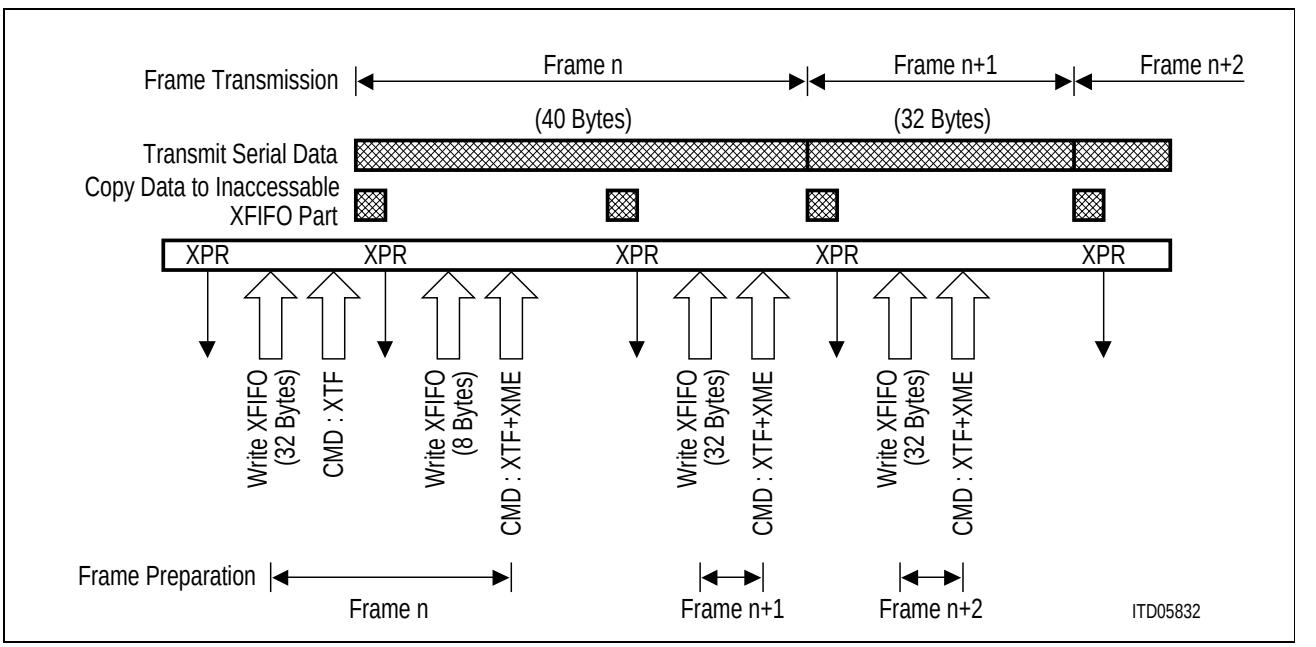



**Figure 34**
**XFIFO Loading, Continuous Frame Transmission Enabled (CFT = 1)**

When using the DMA-mode, prior to the data transfer the actual byte count to be transmitted must be written to the registers XBCH, XBCL (transmit byte count high, low).

If the data transfer is initiated via the proper command, the SACCO automatically requests the correct amount of block data transfers (n × 32 + remainder, n = 0, 1, 2, …) by activating the DRQT-line.

Refer to **chapter 2.2.7.2** for a detailed description of the DMA transfer timing.

### 2.2.7.4 Protocol Support

The SACCO supports the following fundamental HDLC functions:

– Flag insertion/deletion,
– Bit stuffing,
– CRC-generation and checking,
– Address recognition.

Further more it provides six different operating modes, which can be set via the MODE register. These are:

– Auto Mode,
– Non-Auto Mode,
– Transparent Mode 0 and 1,
– Extended Transparent Mode 0 and 1.

These modes provide different levels of HDLC processing. An overview is given in **figure 35**.

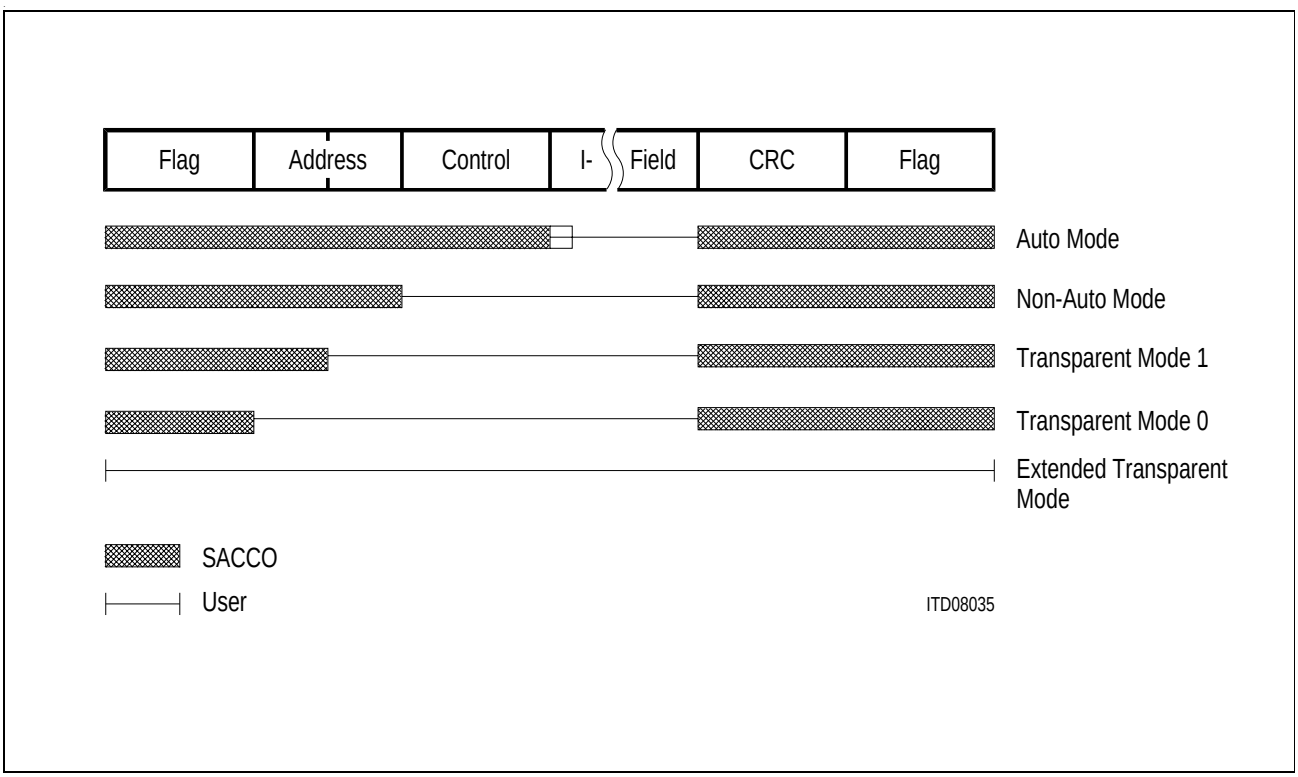



**Figure 35**
**Support of the HDLC Protocol by the SACCO**

### Address Recognition

Address recognition is performed in three operating modes (auto-mode, non-auto-mode and transparent mode 1). Two pairs of compare registers (RAH1, RAH2: high byte compare, RAL1, RAL2: low byte compare) are provided. RAL2 may be used for a broadcast address. In auto-mode and non-auto-mode 1- or 2-byte address fields are supported, transparent mode 1 is restricted on high byte recognition. The high byte address is additionally compared with the LAPD group address ($FC_H$, $FE_H$).

Depending on the operating mode the following combinations are considered valid addresses:

**Table 8**
**Address Recognition**

| Operating Mode | Compare Value High Byte | Compare Value Low Byte | Activity |
|---|---|---|---|
| Auto-mode, 2-byte address field | <RAH1> | <RAL1> | Processed, following the auto-mode protocol |
| | <RAH2> | <RAL1> | Frame is stored transparently in RFIFO |
| | FCH | <RAL1> | |
| | FEH | <RAL1> | |
| | <RAH1> | <RAL2> | |
| | <RAH2> | <RAL2> | |
| | FCH | <RAL2> | |
| | FEH | <RAL2> | |
| Auto-mode, 1-byte address field | – | <RAL1> | Processed, following the auto-mode protocol |
| | – | <RAL2> | Frame is stored transparently in RFIFO |
| Non-auto mode, 2-byte address field | <RAH1> | <RAL1> | Frame is stored transparently in RFIFO |
| | <RAH2> | <RAL1> | |
| | FCH | <RAL1> | |
| | FEH | <RAL1> | |
| | <RAH1> | <RAHL2> | |
| | <RAH2> | <RAL2> | |
| | FCH | <RAL2> | |
| | FEH | <RAL2> | |

**Table 8**
**Address Recognition** (cont'd)

| Operating Mode | Compare Value High Byte | Compare Value Low Byte | Activity |
|---|---|---|---|
| Non-auto mode, 1-byte address field | – | <RAL1> | Frame is stored transparently in RFIFO |
| | – | <RAL2> | |
| Transparent mode 1 | <RAH1> | – | Frame is stored transparently in RFIFO |
| | <RAH2> | – | |
| | FCH | – | |
| | FEH | – | |

**Auto-Mode (MODE:MDS1,MDS0 = 00)**

Characteristics: HDLC formatted, NRM-type protocol, 1-byte/2-byte address field, address recognition, any message length, automatic response generation for RR- and I-frames, window size 1.

The auto-mode is optimized to communicate with a group controller following a NRM-(Normal Response Mode) type protocol. Its functionality guarantees a minimum response time and avoids the interruption of the CPU in many cases.

The SACCO auto-mode is compatible to a PEB 2050 (PBC) behavior in secondary mode.

Following the PBC-conventions, two data types are supported in auto-mode.

**Table 9**
**Auto-Mode Data Types**

| Data Types | Meaning |
|---|---|
| Direct data | Data exchanged in normal operation mode between the local µP and the group controller, typically signaling data. |
| Prepared data | Data request by or send to the group controller for maintenance purposes. |

*Note: In many applications only direct data is used, nevertheless both data types are supported because of compatibility reasons.*

### Receive Direction

In auto-mode the SACCO provides address recognition for 2- and 1-byte address fields. The auto-mode protocol is only applied when RAL1 respectively RAH1/RAL1 match. With any other matching combination, the frame is transferred transparently into the RFIFO and an interrupt (RPF or RME) is issued.

If no address match occurs, the frame is skipped. The auto-mode protocol processes RR- and I-frames automatically. On the reception of any other frame type an EHC-interrupt (extended HDLC frame) is generated. No data is stored in the RFIFO but due to the internal hardware structure the HDLC-control field is temporarily stored in register RHCR. In the PBC-protocol an extended HDLC-frame does not contain any data.

**Table 10**
**HDLC-Control Field in Auto-mode**

| HDLC-Control Byte | Frame Type |
|---|---|
| xxxP xxx0 | I-frame |
| xxxP xx01 | RR-frame |
| xxxx xx11 | Extended HDLC-frame |

### RR-frames

RR-frames are processed automatically and are not stored in RFIFO.
When a RR-frame with poll bit set (control field = xxx10001) is received, it is interpreted as a request to transmit direct data.

Depending on the status of the XFIFO an I-frame (data available) or a RR-response (no data available) is issued.

This behavior guarantees minimum response times and supports a fast cyclical polling of signaling data in a point-to-multi-point configuration.

A RR-frame with poll bit = 0 is interpreted as an acknowledgment for a previously transmitted I-frame: the XFIFO is cleared, a XPR interrupt is emitted, no response is generated.

The polling of a frame can be repeated an unlimited number of times until the frame is acknowledged. Depending on the status of the bit MODE:AREP (auto repeat), the transmission is repeated without or with the intervention of the CPU (XMR interrupt).
The auto repeat mode must not be selected, when the frame length exceeds 32 bytes.
In DMA mode, when using the auto repeat mode, the control response will not be compatible to the PBC.

### I-frames

When an I-frame is received in auto-mode the first data byte is interpreted as a command byte according to the PEB 2050 (PBC) protocol.

Depending on the value of the command byte one of the following actions is performed.

**Table 11**
**Auto-mode Command Byte Interpretation**

| Command Byte = 1. Data Byte | Stored in RFIFO | Interrupt | Additional Activities | Condition |
|---|---|---|---|---|
| 00 - $9F_H$<br>B0 - $CF_H$<br>F0 - $FF_H$ | yes | RPF, RME | Response generation when poll bit set | – |
| A0-$AF_H$ | no | no | Response generation when poll bit set<br>I-frame with XFIFO-Data | –<br>Command XPD executed |
| D0 - $EF_H$ | no | XPR | Response generation when poll bit set,<br>reset XFIFO | Command XPD executed |
| | no | no | Response generation when poll bit set | Command XPD not executed |

When a I-frame is stored in RFIFO the command byte has to be interpreted by software. Depending on the subset of PBC commands used in the individual application, the implementation may be limited to the necessary functions. In case XPD is executed (with or without data in XFIFO) the SACCO will generate an XPR interrupt upon the reception of a command $D0_H$, …, $EF_H$, even if the data has not been polled previously.

*Note: In auto-mode I-frames with wrong CRC or aborted frames are stored in RFIFO. In the attached RSTA-byte the CRC and RAB-bits are set accordingly to indicate this situation. In these cases no response is generated.*

### Transmit Direction, Response Generation

In auto-mode frames are only transmitted after the reception of a RR- or I-frame with poll bit set.

**Table 12**
**Auto-Mode Response Generation**

| Received Frame | Response | Condition |
|---|---|---|
| RR-poll<br>poll bit set | I-frame with XFIFO-data | Command XDD executed |
| | RR-response | Command XDD not executed |
| I-frame, first byte = AxH poll bit set | I-frame with XFIFO-data | Command XPD executed |
| | I-frame, data byte = control response | Command XPD not executed |
| I-frame, first data byte not AxH,<br>poll bit set | I-frame, data byte = control response | |

### RR-Response

The RR-response is generated automatically.
It has the following structure.

| flag | address | control byte | CRC-word | flag |
|---|---|---|---|---|

The address is defined by the value stored in XAD1 (1-byte address) or XAD1 and XAD2 (2-byte address). The control byte is fixed to $11_H$ (RR-frame, final bit = 1).

### Control Response

The control response is generated automatically.
It has the following structure.

| flag | address | control byte | control resp. | CRC-word | flag |
|---|---|---|---|---|---|

The address is defined by the value stored in XAD1 (1-byte address) or XAD1 and XAD2 (2-byte address). The control byte is fixed to $10_H$.

According to the PBC conventions, the control response byte has the following structure:

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | AREP | 0 | 0 | DOV | 1 |

bit7 … 6 : 10 : response to an I-frame, no further data follows
bit5 : 1 : µP connected (PBC operates optionally in stand alone mode)
bit4 : AREP : 1/0: autorepeating is enabled/disabled
(Read back value of CMDR:AREP)
bit3 … 2 : 00 : SACCO FIFO available for data reception
bit1 : DOV : inverted status of the bit RSTA:RDO (RFIFO overflow)
bit0 : 1 : fixed value, no functionality.

### I-Frame with Data

| flag | address | control byte | data | CRC-word | flag |
|---|---|---|---|---|---|

The address is defined by the value stored in XAD1 (1-byte address) or XAD1 and XAD2 (2-byte address). The control byte is fixed to $10_H$ (I-frame, final bit = 1). The data field contains the XFIFO contents.

*Note: The control response byte has to be generated by software.*

### Data Transfer

### Polling of Direct Data

When direct data was loaded (XDD executed) an I-frame is generated as a response to a RR-poll.

After checking STAR:XFW, blocks of up to 32 bytes may be entered in XFIFO. When more than 32 bytes are to be transmitted the XPR-interrupt is used to indicate that the CPU accessible XFIFO-part is free again. A maximum of 64 bytes may be stored before the actual transmission is started.

A RR-acknowledge (poll bit = 0) causes an ISTA:XPR interrupt, XFIFO is cleared and STAR:XFW is set.

When the SACCO receives a RR-poll frame and no data was loaded in XFIFO it generates automatically a RR-response.

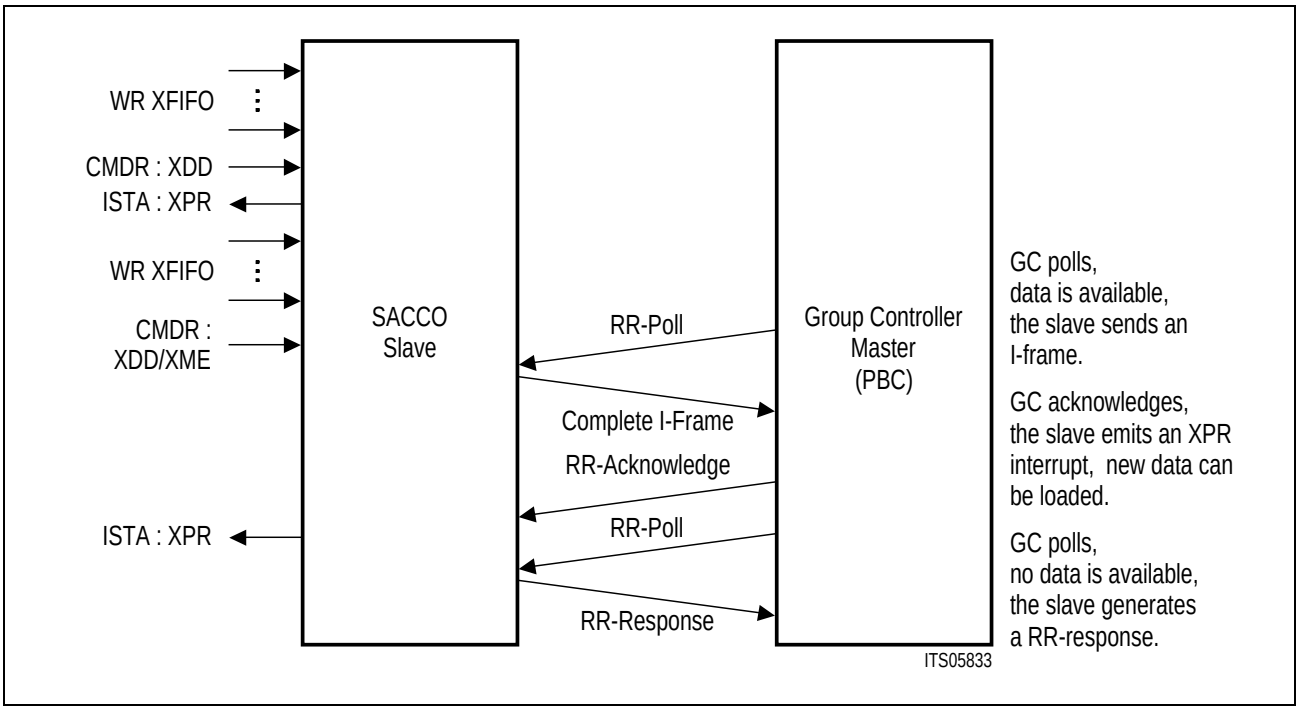



**Figure 36**
**Polling of up to 64 Bytes Direct Data**

If more than 64 bytes are transmitted, the XFIFO is used as an intermediate buffer. Data has to be reloaded after transmission was started.

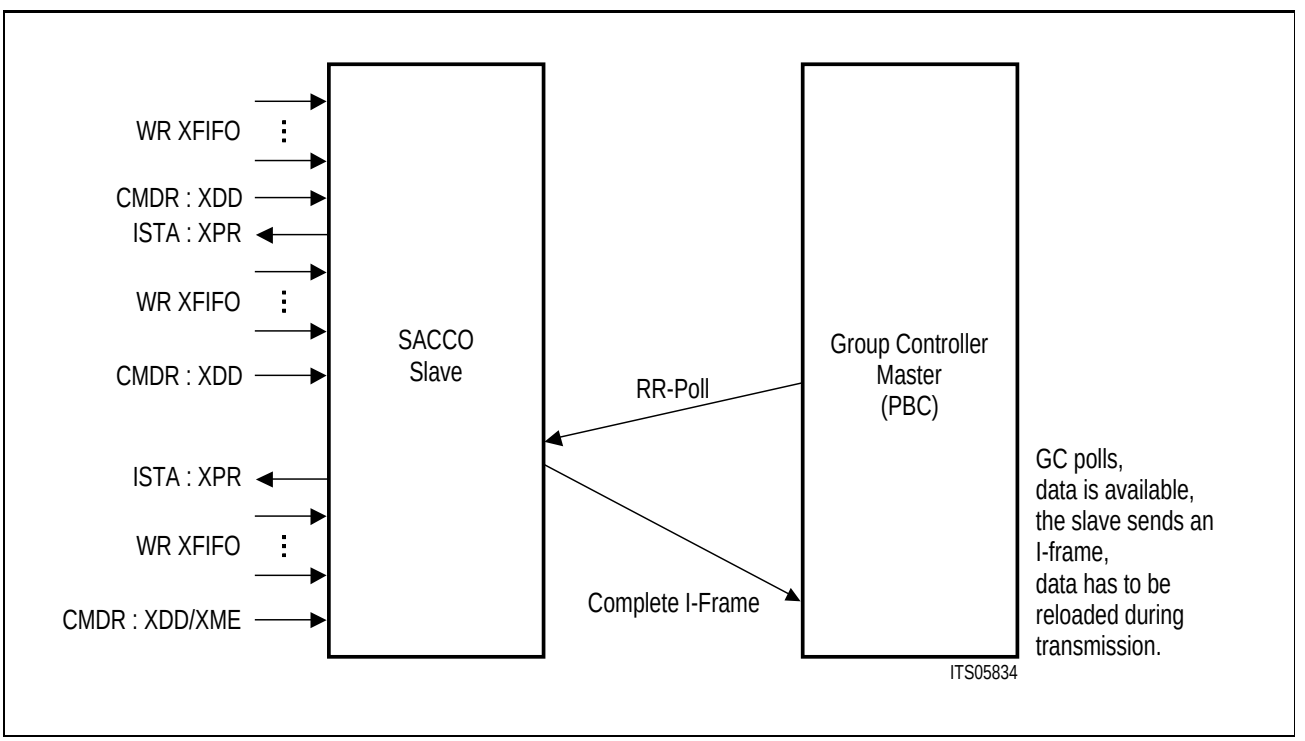



**Figure 37**
**Polling More than 64 bytes of Direct Data (e.g. 96 bytes)**

When the group controller wants the SACCO to re-transmit a frame (e.g. due to a CRC-error) it does not answer with a RR-acknowledge but emits a second RR-poll.

The SACCO then generates an XMR-interrupt (transmit message repeat) indicating the CPU that the previously transmitted frame has to be loaded again. For frames which are not longer then 32 bytes the SACCO offers an auto repeat function allowing the automatic re-transmission of a frame without interrupting the CPU.

*Note: For frames which are longer than 32 bytes the auto repeat function must not be used.*

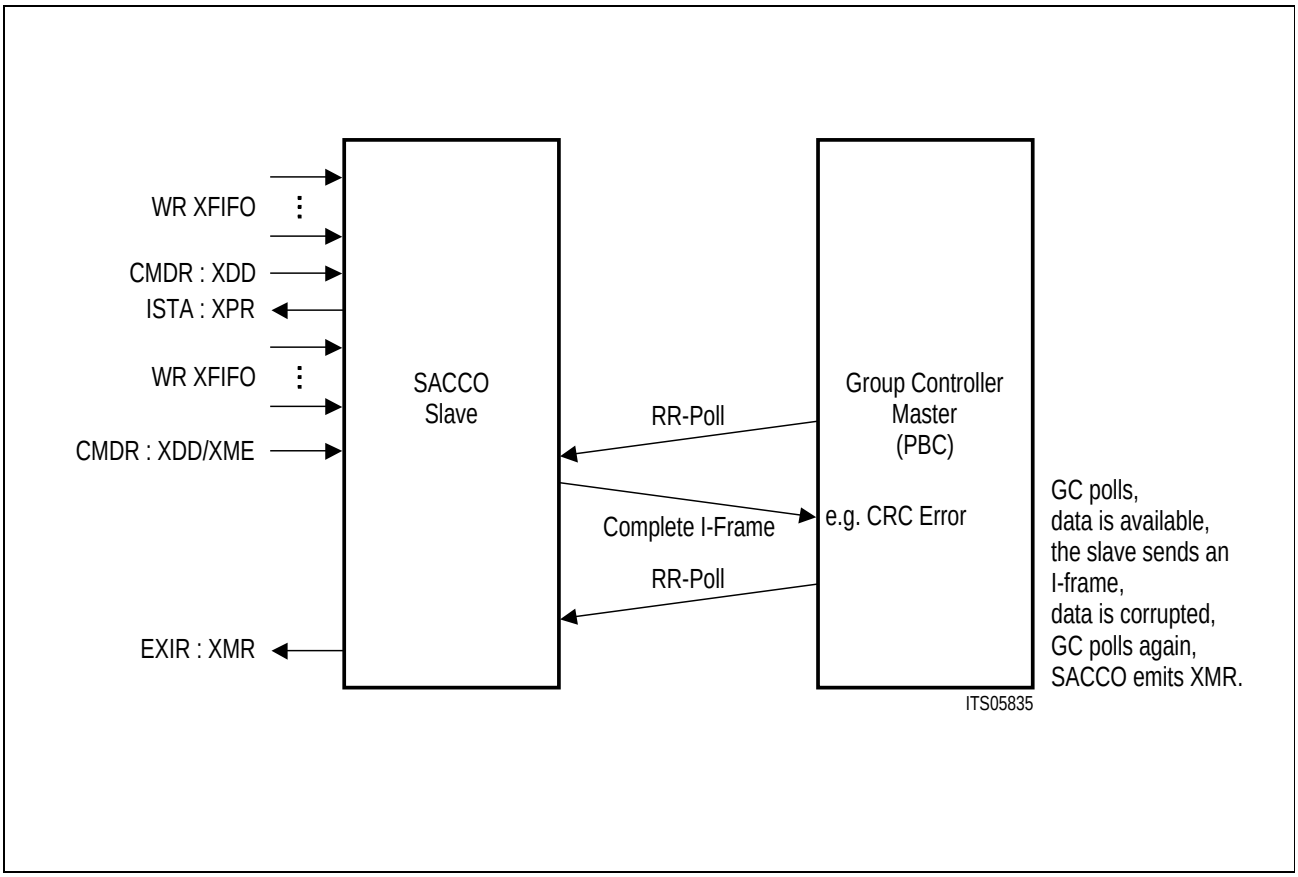



**Figure 38**
**Re-transmission of a Frame**

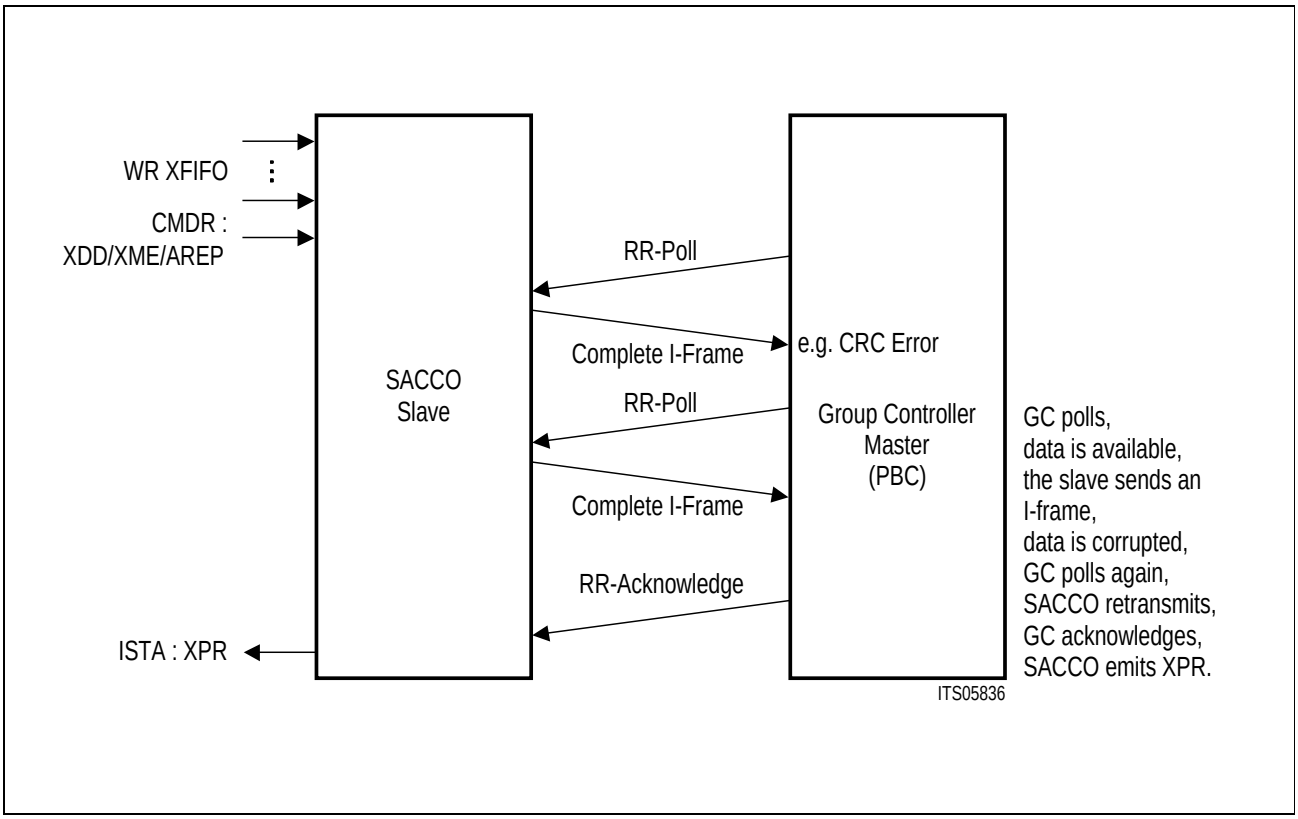



**Figure 39**
**Re-transmission of a Frame with Auto-Repeat Function**

### Polling of Prepared Data

If polling "prepared data" a different procedure is used. The group controller issues an I-frame with a set poll bit and the first data byte is interpreted as command byte.

When prepared data was loaded into the XFIFO (CMDR:XPD/XME was set) the reception of a command byte equal to AxH initiates the transmission of an I-frame.

For "prepared data" the auto repeat function must be selected! Due to this the polling can be repeated without interrupting the CPU.

An I-frame with a data byte equal to D0H-EFH is interpreted as an acknowledgment for previously transmitted data. An XPR-interrupt is issued and the XFIFO is reset.

All other I-frames are stored in the RFIFO and a RME-interrupt is generated. The local µP can read and interpret the received data (e.g. following the PBC-protocol). A PBC compatible control response is generated automatically.

E.g., if the local µP recognizes the request to "prepare data" it may load the XFIFO and set CMDR:XPD/XME.

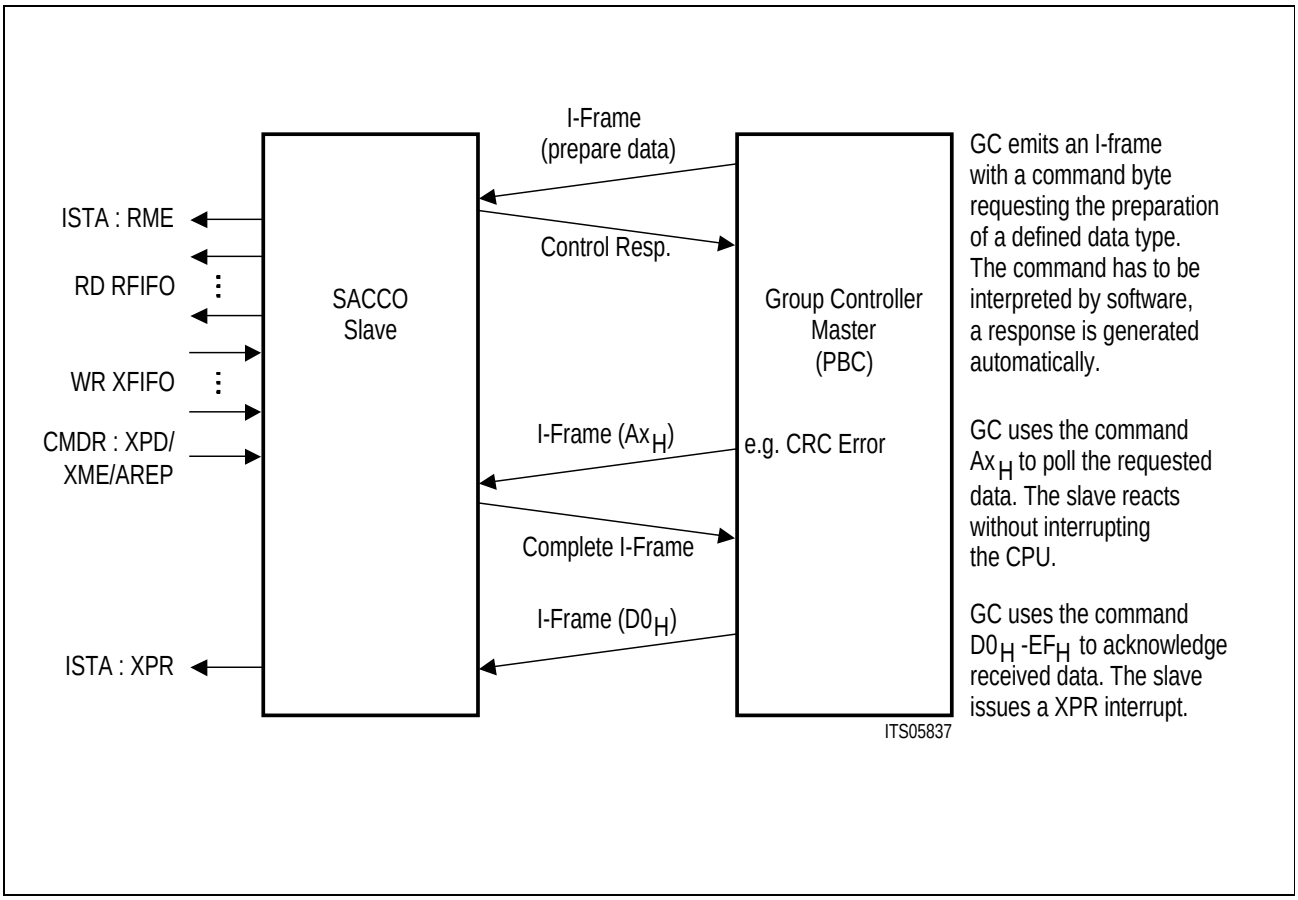



**Figure 40**
**Polling of Prepared Data**

### Behavior of SACCO when a RFIFO Overflow Occurs in Auto-mode

When the RFIFO overflows during the reception of an I-frame, a control response with overflow indication is transmitted, the overflow information is stored in the corresponding receive status byte. When additional poll frames are received while the RFIFO is still occupied, an RFO (receive frame overflow) interrupt is generated. Depending on the type of the received poll frame different responses are generated:

I-frame: – control response with overflow indication
(exception: when the command "transmit prepared data" (AxH) is received and prepared data is available in the XFIFO, an I-frame (with data) is issued)

RR-poll: – RR-response, when no direct data was stored in the XFIFO
– I-frame, when direct data was stored in the XFIFO

Depending on the number of bytes to be stored in the RFIFO the following behavior occurs:

| RFIFO Handling/Steps | Case 1 | Case 2 | Case 3 |
|---|---|---|---|
| Receive frame | Total frame length: 63 data bytes | Total frame length: 64 data bytes | Total frame length: 65 data bytes or more |
| After 32 bytes are received | A RPF-interrupt is issued, the RFIFO is not acknowledged | | |
| After next 31/32 bytes are received | Control response, no overflow indication | Control response with overflow indication | Control response with overflow indication |
| Additional I-poll | RFO-interrupt, I-response with overflow indication or I-data if stored in XFIFO as prepared data | | |
| Additional RR-poll | RFO-interrupt, RR-response or I-data if stored in XFIFO as direct data | | |
| Read and acknowledge RFIFO | RME-interrupt | RPF-interrupt | RPF-interrupt |
| Read and acknowledge RFIFO | RDO-bit is not set, frame is complete | RME-interrupt | RME-interrupt |
| Read and acknowledge RFIFO | | RDO-bit is set, **frame is complete but indicated as incomplete** | RDO-bit is set, frame is not complete |

Multiple shorter frames results in the same behavior, e.g.

frame 1: 1 - 31 bytes
frame 2 - n: total of 31 bytes including receive status bytes for frame 2 - (n – 1)

cause the case 1.

### Non-Auto-Mode (MODE:MDS1, MDS0 = 01)

Characteristics: HDLC formatted, 1-byte/2-byte address field, address recognition, any message length, any window size.

All frames with valid address fields are stored in the RFIFO and an interrupt (RPF, RME) is issued.

The HDLC-control field, data in the I-field and an additional status byte are stored in RFIFO. The HDLC-control field and the status byte can also be read from the registers RHCR, RSTA (currently received frame only!).

According to the selected address mode, the SACCO can perform 2-byte or 1-byte address recognition.

### Transparent Mode 1 (MODE:MDS1, MDS0, ADM = 101)

Characteristics: HDLC formatted, high byte address recognition, any message length, any window size.

Only the high byte address field is compared with RAH1, RAH2 and the group address ($FC_H$, $FE_H$). The whole frame except the first address byte is stored in RFIFO. RAL1 contains the second and RHCR the third byte following the opening flag (currently received frame only). When using LAPD the high byte address recognition feature can be used to restrict the frame reception to the selected SAPI-type.

### Transparent Mode 0 (MODE:MDS1, MDS0, ADM = 100)

Characteristics: HDLC formatted, no address recognition, any message length, any window size.

No address recognition is performed and each frame is stored in the RFIFO. RAL1 contains the first and RHCR the second byte following the opening flag (currently received frame only).

*Note: In non-auto-mode and transparent mode I-frames with wrong CRC or aborted frames are stored in RFIFO. In the attached RSTA-byte the CRC and RAB-bits are set accordingly to indicate this situation.*

### Extended Transparent Mode 0 (MODE:MDS1, MDS0, ADM = 110)

Characteristics: fully transparent without HDLC framing, any message length, any window size.
Data is stored in register RAL1.

In extended transparent mode, fully transparent data transmission/reception without HDLC-framing is performed, i.e. without FLAG-generation/recognition, CRC-generation/check, bit stuffing mechanism. This allows user specific protocol variations or can be used for test purposes (e.g. to generate frames with wrong CRC-words).

Data transmission is always performed out of the XFIFO. Data reception is done via register RAL1, which contains the actual data byte assembled at the RxD pin.

### Extended Transparent Mode 1 (MODE:MDS1, MDS0, ADM = 111)

Characteristics: fully transparent without HDLC-framing, any message length, any window size. Data is stored in register RAL1 and RFIFO.
Identical behavior as extended transparent mode 0 but the received data is shifted additionally into the RFIFO.

### Receive Data Flow (summary)

The following figure gives an overview of the management of the received HDLC-frames depending on the selected operating mode.

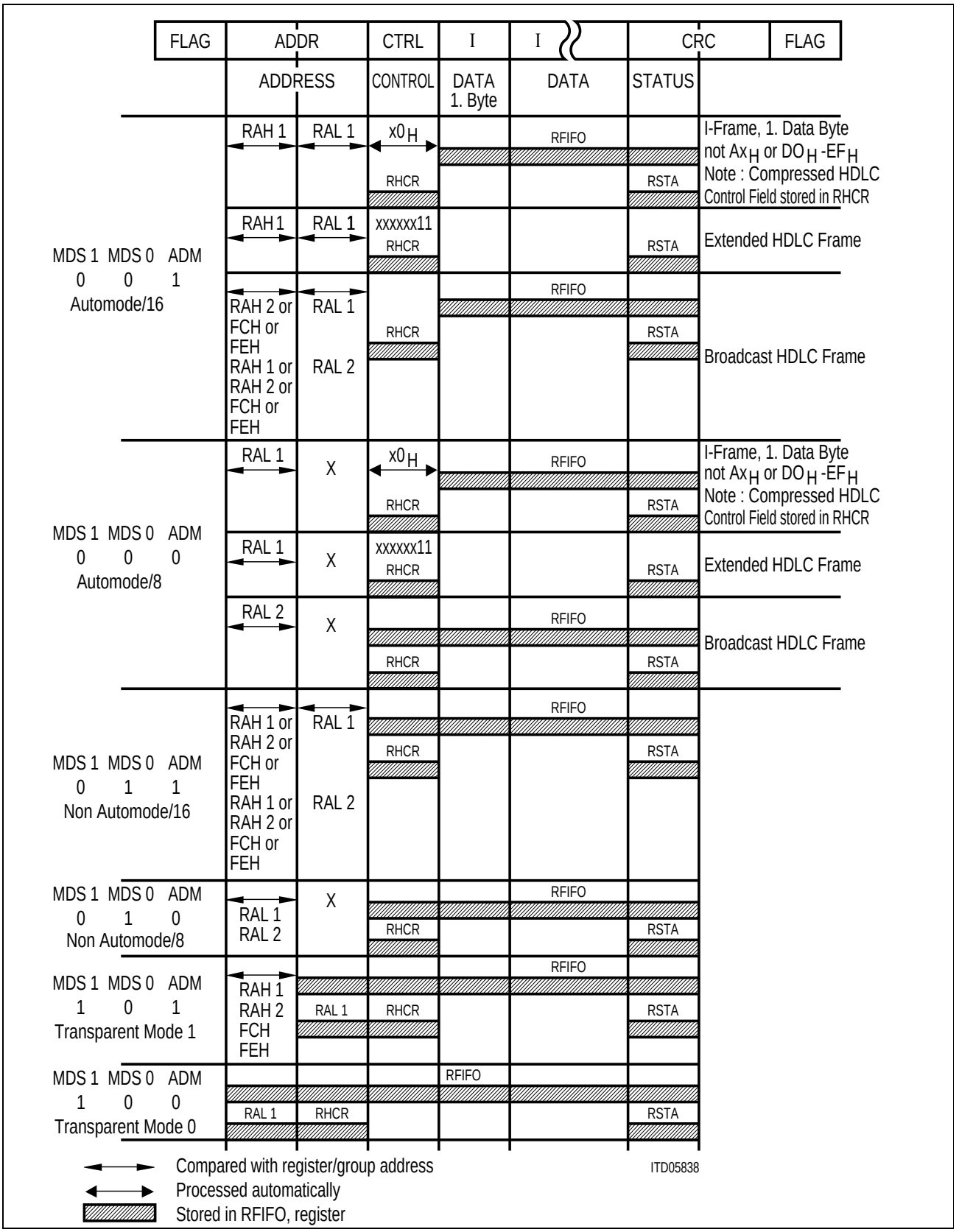



**Figure 41**
**Receive Data Flow**

*Note: RR-frames and I-frame with first data byte equal to $Ax_H$ or $D0_H$ - $EF_H$ are processed automatically. They are not stored in RFIFO and no interrupt is issued.*

### 2.2.7.5 Special Functions

#### Cyclical Transmission (fully transparent)

When the extended transparent mode is selected, the SACCO supports the continuous transmission of the XFIFO-contents.

After having written 1 to 32 bytes to the XFIFO, the command XREP/XTF/XME (XREP/XTF in DMA-mode) is executed. Consequently the SACCO repeatedly transmits the XFIFO-data via pin TxD.

The cyclical transmission continues until the command (CMDR:XRES) is executed or the bit XREP is reset. The inter frame timefill pattern is issued afterwards.
When resetting XREP, data transmission is stopped after the next XFIFO-cycle is completed, the XRES-command terminates data transmission immediately.

*Note: Bit MODE:CFT must be set to "0".*

#### Continuous Transmission (DMA-mode only)

If data transfer from system memory to the SACCO is done by DMA (DMA bit in XBCH set), the number of bytes to be transmitted is usually defined via the transmit byte count registers XBCH, XBCL. Setting the "transmit continuously" bit (XC) in XBCH, however, the byte count value is ignored and the DMA-interface of the SACCO will continuously request for transmit data any time 32 bytes can be stored in the XFIFO.

This feature can be used e.g. to

- continuously transmit voice or data onto a PCM-highway (clock mode 2, ext. transp. mode)
- transmit frames exceeding the byte count programmable in XBCH, XBCL (> 4095 bytes).

*Note: If the XC-bit is reset during continuous transmission, the transmit byte count becomes valid again, and the SACCO will request the amount of DMA-transfers programmed in XBC11 … XBC0. Otherwise the continuous transmission is stopped when a data underrun condition occurs in the XFIFO, i.e. the DMA-controller does not transfer further data to the SACCO. In this case an abort sequence (min. 7 '1's) followed by the inter frame timefill pattern is transmitted (no CRC-word is appended).*

#### Receive Length Check

The SACCO offers the possibility to supervise the maximum length of received frames and to terminate data reception in case this length is exceeded.

This feature is enabled by setting the RC- (receive check) bit in RLCR and programming the maximum frame length via bits RL6…RL0.

According to the value written to RL6…RL0, the maximum receive length can be adjusted in multiples of 32-byte blocks as follows: max. frame length = (RL + 1) × 32.

All frames exceeding this length are treated as if they have been aborted from the opposite station, i.e. the CPU is informed via a

– RME-interrupt, and the
– RAB-bit in RSTA register is set (clock mode 0 - 2)

To distinguish between frames really aborted from the opposite station, the receive byte count (readable from registers RBCH, RBCL) exceeds the maximum receive length (via RL6…RL0) by one or two bytes in this case.

#### 2.2.7.6 Serial Interface

**Clock Modes**

The SACCO uses a single clock for transmit and receive direction. Three different clock modes are provided to adapt the serial interface to different requirements.

**Clock Mode 0**

Serial data is transferred on RxD/TxD, an external generated clock (double or single data rate) is forwarded via pin HDC.

**Clock Mode 1**

Serial data is transferred on RxD/TxD, an external generated clock (double or single data rate) is forwarded via pin HDC. Additionally a receive/transmit strobe provided on pin HFS is evaluated.

**Clock Mode 2**

This operation mode has been designed for applications in time slot oriented PCM-systems. The SACCO receives and transmits only during a certain time slot of programmable width (1 … 256 bits) and location with respect to a frame synchronization signal, which must be delivered via pin HFS.

The position of the time slot can be determined applying the formula in **figure 42**.

**TSN:** Defines the number of 8 bit time slots between the start of the frame (HFS edge) and the beginning of the time slot for the HDLC channel. The values for TSN are written to the registers TSAR:7…2 and TSN:7…2.

**CS:** Additionally a clock shift of 0…7 bits can be defined using register bits TSAR:RSC2…1, TSAX:XCS2…1 and CCR2:XCS0, CCR2:RCS0.

Together TSN and CS provide 9 bits to determine the location of the time slot for the HDLC channel.

One of up to 64 time slots can be programmed independently for receive and transmit direction via the registers TSAR and TSAX.

According to the value programmed via those bits, the receive/transmit window (time slot) starts with a delay of 1 (minimum delay) up to 512 clock periods following the frame synchronization signal and is active during the number of clock periods programmed via RCCR, XCCR (number of bits to be received/transmitted within a time slot) as shown in **figure 42**.

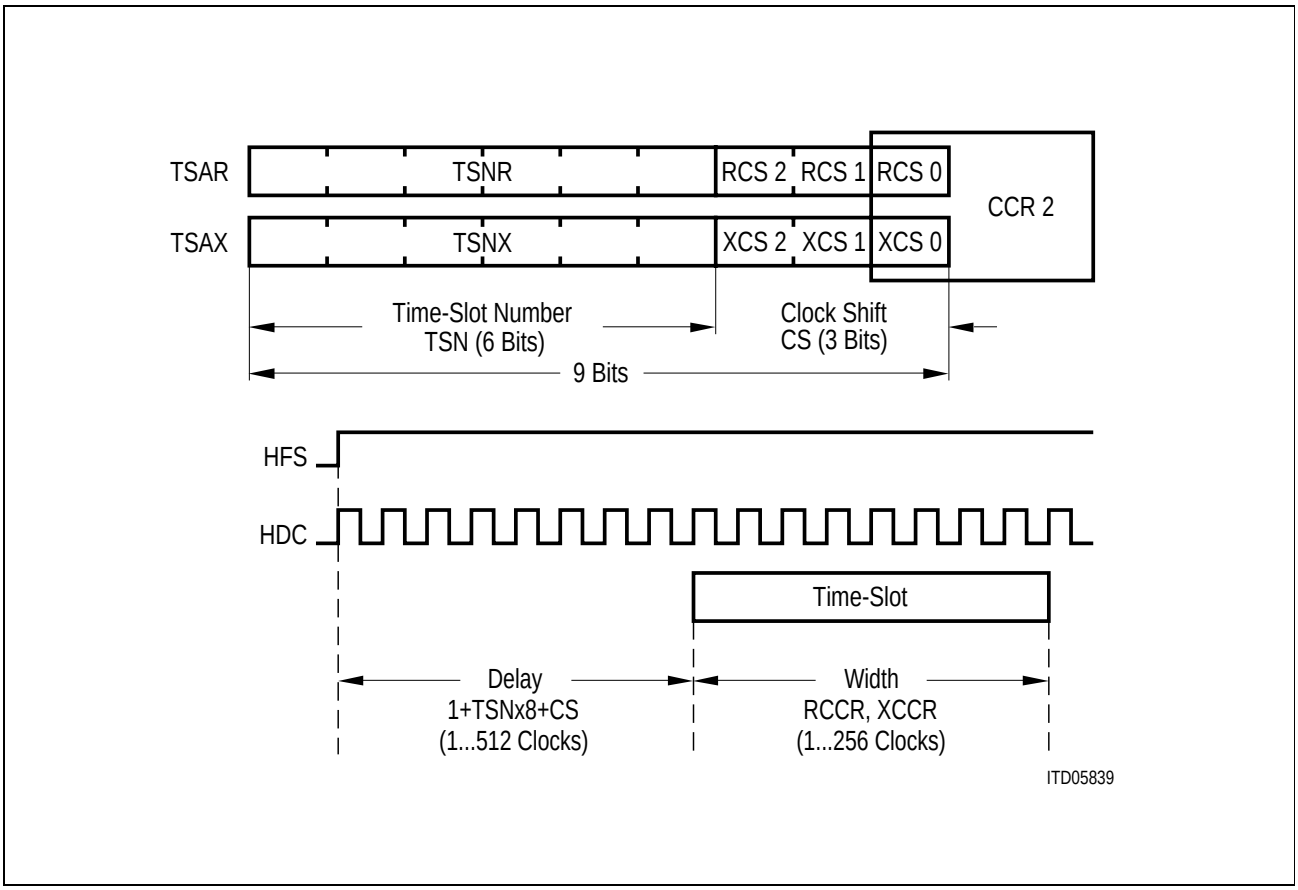



**Figure 42**
**Location of Time Slots**

*Note: In extended transparent mode the width of the time slot has to be n × 8 bit.*

### Clock Mode 3

In clock mode 3 SACCO-A is multiplexed among multiple subscribers under the control of the D-channel arbiter. It must be used only in combination with transparent mode 0.

Serial data is transferred on (received from) the D-channels of the EPIC-1 IOM-2 interfaces. The data clock is derived from DCL. The D-channel arbiter generates the receive and transmit strobes.

When bit CCR2:TXDE is set, the transmitted D-channel data can additionally be monitored on pin TxDA delayed by 1 bit. The timing is identical to clock mode 1 assuming a transmit strobe during the transmission of the third and fourth bit following the rising FSC-edge.

### Receive Status Byte in Clock Mode 3

In clock mode 3 the receive status byte is modified when it is copied into RFIFO. It contains the following information:

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| VFR | RDO | CRC | CHAD4 | CHAD3 | CHAD2 | CHAD1 | CHAD0 |

**VFR** Valid Frame.
Indicates whether the received frame is valid (’1’) or not (’0’ invalid).
A frame is invalid when
– its length is not an integer multiple of 8 bits (n $\times$ 8 bits), e.g. 25 bit,
– it is too short, depending on the selected operation mode (transparent mode 0: 2 bytes minimum),
– the frame was aborted from the transmitting station.

**RDO** Receive Data Overflow.
A '1' indicates, that a RFIFO-overflow has occurred within the actual frame.

**CRC** CRC Compare Check.
0: CRC check failed, received frame contains errors.
1: CRC check o.k., received frame is error free.

**CHAD4..0** Channel Address 4…0.
CHAD4..0 identifies on with IOM-port/channel the corresponding frame was received:
CHAD4..3: IOM-port number (3 - 0)
CHAD2..0: IOM-channel number (7 - 0)

*Note: The contents of the receive status register is not changed.*

### 2.2.7.7 Serial Port Configuration

The SACCO supports different serial port configuration, enabling the use of the circuit in

– point-to-point configurations
– point-to-multi-point configurations
– multi master configurations

#### Point-to-Point Configuration

The SACCO transmits frames without collision detection/resolution.
(CCR1:SC1, SC0: 00)

Additionally the input CxD can be used as a "clear to send" strobe. Transmission is inhibited by a "1" on the CxD-input. If "CxD" becomes "1" during the transmission of a frame, the frame is aborted and IDLE is transmitted. The CxD-pin is evaluated with the falling edge of HDC.
When the "clear to send" function is not needed, CxD must be tied to $V_{SS}$.

#### Bus Configuration

The SACCO can perform a bus access procedure and collision detection. As a result, any number of HDLC-controllers can be assigned to one physical channel, where they perform statistical multiplexing.

Collisions are detected by automatic comparison of each transmitted bit with the bit received via the CxD input. For this purpose a logical AND of the bits transmitted by parallel controllers is formed and connected to the input CxD. This may be implemented most simply by defining the output line to be open drain. Consequently the logical AND of the outputs is formed by simply tying them together ("wired or"). The result is returned to the CxD-input of all parallel circuits.

When a mismatch between a transmitted bit and the bit on CxD is detected, the SACCO-stops sending further data and IDLE is transmitted. As soon as it detects the transmit bus to be idle again, the controller automatically attempts to re-transmit its frame. By definition, the bus is assumed idle when x consecutive ones are detected in the transmit channel. Normally x is equal to 8.

An automatic priority adjustment is implemented in the multi master mode. Thus, when a complete frame is successfully transmitted, x is increased to 10, and its value is restored to 8 when 10 '1's are detected on the bus (CxD). Furthermore, transmission of new frames may be started by the controller after the $10^{th}$ '1'.

This multi master, deterministic priority management ensures an equal right of access of every HDLC-controller to the transmission medium, thereby avoiding blocking situations.

Compared to the Version 1.2 the Version 1.3 provides new features:

Push-pull operation may be selected in bus configuration (up to Version 1.2 only open drain):

- When active TXDA / TXDB outputs serial data in push-pull-mode.
- When inactive (interframe or inactive timeslots) TXDA / TXDB outputs '1'.

*Note: When bus configuration with direct connection of multiple ELIC's is used open drain option is still recommended.*
*The push-pull option with bus configuration can only be used if an external tri-state buffer is placed between TXDA / TXDB and the bus.*
*Due to the delay of TSCA / TSCB in this mode (see description of bits SOC(0:1) in register CCR2 (**chapter 4.7.9**)) these signals cannot directly be used to enable this buffer.*

**Timing Mode**

When the multi master configuration has been selected, the SACCO provides two timing modes, differing in the period between sending data and evaluating the transmitted data for collision detection.

– Timing mode 1 (CCR1:SC1, SC0 = 01)
  Data is output with the rising edge of the transmit clock via TxD and evaluated 1/2 clock period later with the falling clock edge at the CxD pin.
– Timing mode 2 (CCR1:SC1, SC0 = 11)
  Data is output with the falling clock edge and evaluated with the next falling clock edge. Thus a complete clock period is available during data output and their evaluation.

### 2.2.7.8 Test Mode

To provide support for fast and efficient testing, the SACCO can be operated in the test mode by setting the TLP-bit in the MODE-register.

The serial input and output pins (TxD, RxD) are connected generating a local loop back. As a result, the user can perform a self-test of the SACCO. Transmit lines TXDA/B are also active in this case, receive inputs RXDA/B are deactivated.

### 2.2.8 D-Channel Arbiter

The D-channel arbiter facilitates the simultaneous serving of multiple D-channels with one HDLC-controller (SACCO-A) allowing a full duplex signaling protocol (e.g. LAPD). It builds the interface between the serial input/output of SACCO-channel A and the time slot oriented D-channels on the EPIC-1 IOM-2 interface.

The SACCO-operation mode "transparent mode 0" has to be selected when using the arbiter.

It is only possible to operate the D-channel arbiter with framing control modes 3, 6 and 7, (refer to register EPIC-1.CMD2:FC(2:0)).

The arbiter consists of three sub blocks:

- Arbiter state machine (ASM): selects one subscriber for upstream D-channel assignment
- Control channel master (CCM): issues the "D-channel available" information from the arbiter in the control channel
- Transmit channel selector (TCHS): selects one or a group of subscribers for D-channel assignment

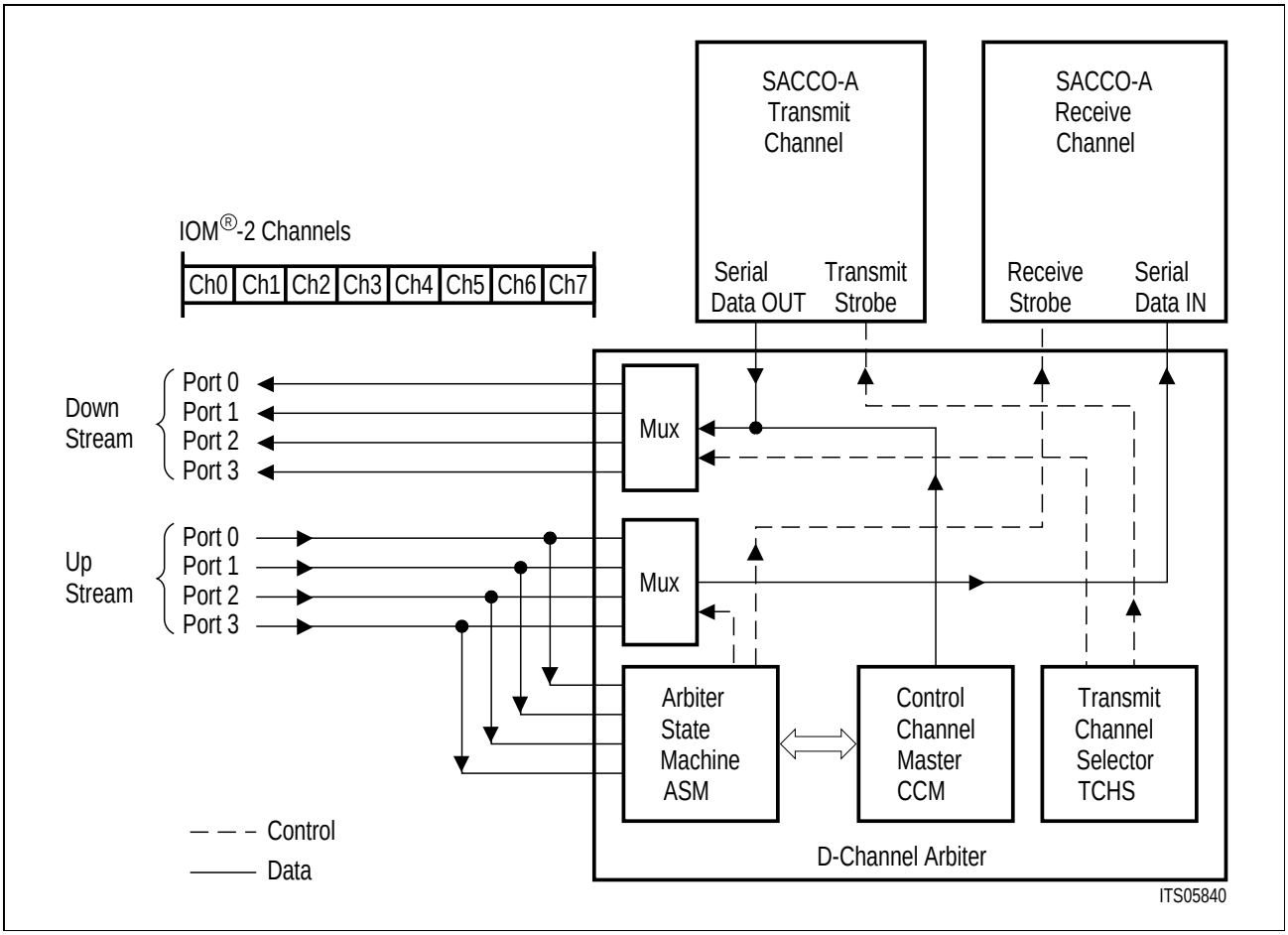



**Figure 43**
**D-Channel Arbiter**

#### 2.2.8.1 Upstream Direction

In upstream direction the arbiter assigns the receive channel of SACCO-A to one subscriber terminal.

It uses an unidirectional control channel to indicate the terminals whether their D-channels are available or blocked. The control channel is implemented using different existing channel structures to close the transmission path between the line card HDLC-controller and the HDLC-controller in the subscriber terminal. On the line card, the control channel is either integrated in the C/I-channel or transmitted in the MR-bit depending on a programming of bit AMO:CCHH (OCTAT-P -> C/I channel, IBC -> MR-bit), see also **chapter 1.6.1.2**.

**Arbiter State Machine**

The D-channel assignment is performed by the arbiter state machine (ASM), implementing the following functionality.

(0) After reset or when SACCO-A clock mode is not 3 the ASM is in the state **"suspended"**. The user can initialize the arbiter and select the appropriate SACCO clock mode (mode 3).

(1) When the receiver of SACCO-A is reset and clock mode 3 is selected the ASM enters the state **"full selection"**. In this state all D-channels enabled in the D-channel enable registers (DCE) are monitored.

(2) Upon the detection of the first ’0’ the ASM enters the state **"expect frame"**. When simultaneously ’0’s are detected on different IOM-2 channels, the lowest channels number is selected. Channel and port address of the related subscriber are latched in arbiter state register (ASTATE), the receive strobe for SACCO-A is generated and the DCE-values are latched into a set of slave registers (DCES). Additionally a suspend counter is loaded with the value stored in register SCV. The counter is decremented after every received byte (4 IOM-frames).

(3) When the counter underflows before the state "expect frame" was left, the corresponding D-channel is considered to produce permanent bit errors (typical pattern: …111011101011…). The ASM emits an interrupt, disables the receive strobe and enters the state **"suspended"** again. The user can determine the affected channel by reading register ASTATE. In order to reactivate the ASM the user has to reset the SACCO-A receiver.

(4) When seven consecutive '1's are detected in the state "expect frame" before the suspend counter underflows the ASM changes to the state **"limited selection"**. The previously detected '0' is considered a single bit error (typical pattern: …11111101111111111…). The receive strobe is turned off and the DCES-bit related to the corresponding D-channel is reset, i.e. the subscriber is temporarily excluded of the priority list.

(5) When SACCO-A indicates the recognition of a frame (frame indication after receiving 3 bytes incl. the flag) before the suspend counter underflows the ASM enters the state **"receive frame"**.

(6) The ASM-state changes from "receive frame" to **"limited selection"** when SACCO-A indicates "end of frame". The receive strobe is turned off and the DCES-bit related to the corresponding D-channel is reset. The ASM again monitors the D-channels but limited to the group enabled in the slave registers DCES "anded" with DCE. The "and" function guarantees, that the user controlled disabling of a subscriber has immediate effect.

(7) When the ASM detects a ’0’ on the serial input line it enters the state **"expect frame"**. Channel and port address of the related subscriber are latched in the arbiter state register (ASTATE), the receive strobe for SACCO-A is generated and the suspend counter is loaded with the value stored in register SCV. The counter is decremented after every received byte. When simultaneously ’0’s are detected on different IOM-2 channels, the lowest channel is selected.

(8) When the ASM does not detect any ’0’ on the remaining serial input lines during n IOM-frames (n is programmed in the register AMO) it re-enters the state **"full selection"**. The list of monitored D-channels is then increased to the group selected in the user programmable DCE-registers. **In order to avoid arbiter locking n has to be greater than the value described in chapter 2.2.8.3 or must be set to 0 (see chapter 4.8.1 Arbiter Mode Register).**

(9) If n is set to 0, then the state “limited selection” is skipped.

The described combination of DCE and DCES implements a priority scheme guaranteeing that (almost) simultaneous requesting subscribers are served sequentially before one is selected a second time.

The current ASM-state is accessible in ASTATE7:5.

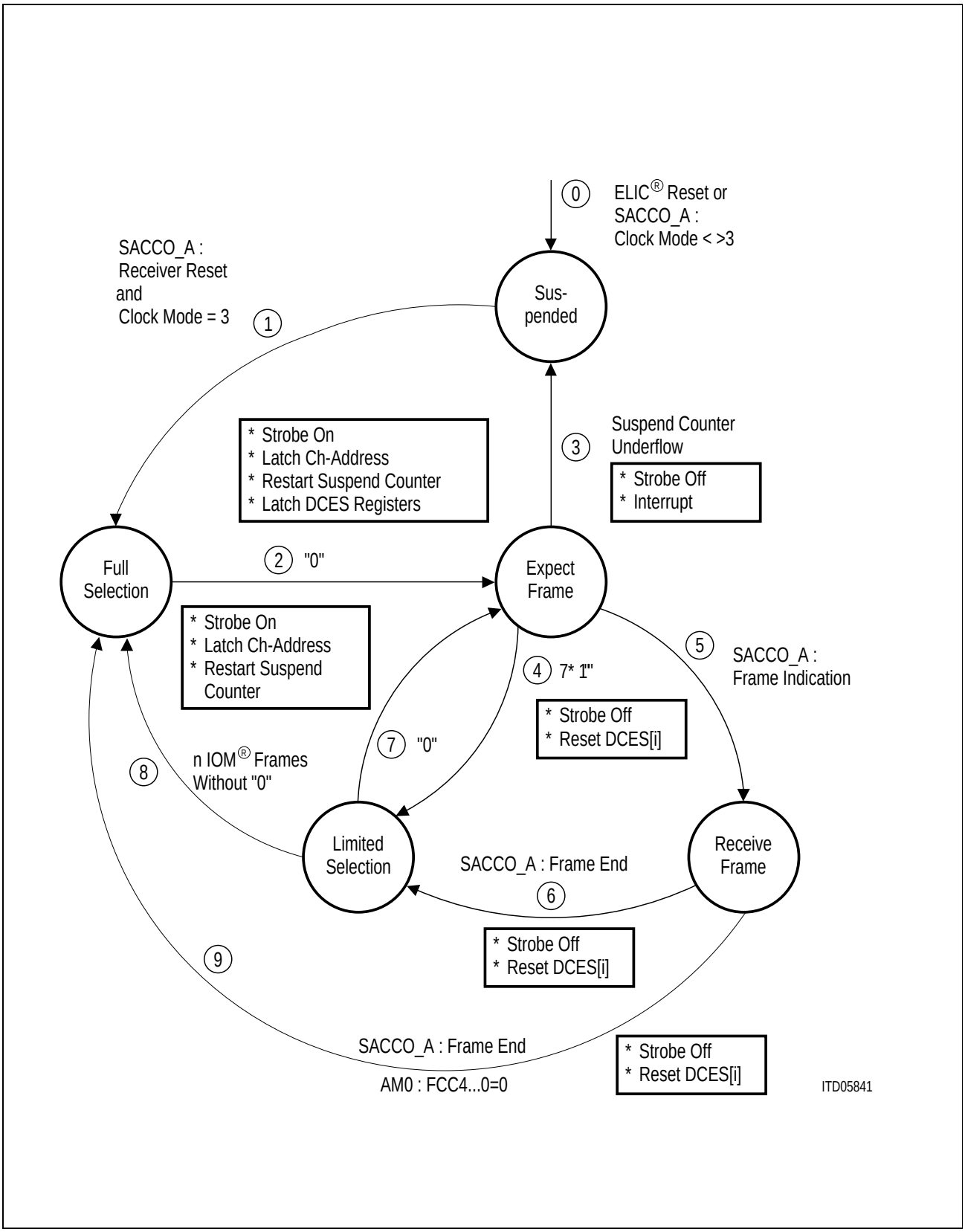



**Figure 44**
**Arbiter State Machine (ASM)**

### Control Channel Master

The control channel master (CCM) issues the "D-channel available" information in the control channel as shown in **table 13**. If a D-channel is not enabled by the arbiter, the control channel passes the status, stored in the EPIC-1 control memory (C/I, MR). For correct operation of the arbiter this status bit has to contain the "blocked" information for all D-channels under control of the arbiter.

If the ASM is in the state **"suspended"** the arbiter functionality depends on the status of the Control Channel Master:
The CCM is enabled if AMO:CCHM = ’1’. All subscribers will be sent the "available/ blocked" information (C/I or MR) as programmed in the control memory. However, the control memory should be programmed as "blocked".

The CCM is disabled if AMO:CCHM = ’0’. All in the DCE-registers enabled subscribers (DCE = ’1’) will be sent the information "available" (which has a higher priority than the "blocked" information from EPIC-1).

If the ASM is in the state **"full selection"** all D-channels are marked to be available which are enabled in the user programmable DCE-registers. When the user reprograms a DCE-register this has an immediate effect, i.e. a currently transmitting subscriber can be forced to abort its message.

If the ASM is in the state **"limited selection"** the subscribers which are currently enabled in DCE and DCES get the information "available"; they can access the D-channel. The DCE/DCES **and**ing is performed in order to allow an immediate disabling of individual subscribers.

In the state **"expect frame"** and **"receive frame"** all channels except one (addressed by ASTATE4:0) have blocked D-channels. The disabling of the currently addressed D-channel in DCE has an immediate effect; the transmitter (HDLC-controller in the subscriber terminal) is forced to abort the current frame.

Depending on the programming of AMO:CCHH the available/blocked information is coded in the C/I-channel or in the MR-bit.

**Table 13**
**Control Channel Implementation**

| CCHH | Control via | Available | Blocked |
|---|---|---|---|
| 1 | MR | 1 | 0 |
| 0 | C/I | x0xx | x1xx |

The CCHM is activated independently of the SACCO-clock mode by programming AMO:CCHM. Even when the ASM is disabled (clock mode not 3) the CCHM can be activated. In this case the content of the DCE-registers defines which D-channels are enabled.

When a D-channel is enabled in the DCE-register and available, the control channel master takes priority over the C/I- (MR) values stored in the EPIC-1 control memory and writes out either MR = 1 or C/I = x0xx. When a D-channel is enabled but blocked, the control channel master simply passes the C/I- (MR) values which are stored in the EPIC-1 control memory. These values should have been programmed as MR = 0 or C/I = x1xxx.

When a D-channel is disabled in the DCE-register the control channel master simply passes the C/I- (MR) values which are stored in the EPIC-1 control memory. This gives the user the possibility to exclude a D-channel from the arbitration but still decide whether the excluded channel is available or blocked.

Overview of different conditions for control channel handling/information sent to subscribers:

| **Clock Mode** | **3** | | X | | X | |
|---|---|---|---|---|---|---|
| **ASM State** | Not suspended | | Suspended | | X | |
| **CCHM** | '1' = enabled | | '1' = enabled | | **'0' = disabled** | |
| **Subscriber in DCEs** | Enabled | Disabled | Enabled | Disabled | Enabled | Disabled |
| **Information sent to Subscribers = "available" or "blocked"** | According to the D-channel **Arbiter** State (CCM) | Content of the EPIC-1 Control Memory- (C/I or MR) | Content of the EPIC-1 Control Memory- (C/I or MR) | Content of the EPIC-1 Control Memory- (C/I or MR) | **Available!** | Content of the EPIC-1 Control Memory- (C/I or MR) |

### 2.2.8.2 Downstream Direction

In downstream direction no channel arbitration is necessary because the sequentiality of the transmitted frames is guaranteed.
In order to define IOM-channel and port number to be used for a transmission, the transmit channel selector (TCHS) provides a transmit address register (XDC) which the user has to write before a transmit command (XTF) is executed. Depending on the programming of the XDC-register the frame is transmitted in the specified D-channel or send as broadcast message to the broadcast group defined in the registers BCG1-4.
Due to the continuous frame transmission feature of the SACCO, the full 16-kbit/s bandwidth of the D-channel can be utilized, even when addressing different subscribers.

*Note: The broadcast group must not be changed during the transmission of a frame*

### 2.2.8.3 Control Channel Delay

Depending on the selected system configuration different delays between the activation of the control channel and the corresponding D-channel response occur.

**Table 14**
**Control Channel Delay Examples**

| System Configuration | Circuit Chain | Number of Frames (= 125 µs) | | | |
|---|---|---|---|---|---|
| | | Blocked → Available | | Available → Blocked | |
| | | min. | max. | min. | max. |
| $U_{PN}$ line card - - $U_{PN}$ phone | ELIC + OCTAT-P + ISAC-P TE | 4 | 8 | 4 | 8 |
| $U_{PN}$ line card - - $S_0$ adapter - - $S_0$ phone | ELIC + OCTAT-P + ISAC-P TE + SBCX + ISAC-S | 9 | 13 | 5 | 9 |
| $U_{PN}$ line card - - $U_{PN}$ adapter - - $U_{PN}$ phone | ELIC + OCTAT-P + ISAC-P TE + ISAC-P TE + ISAC-P TE | 9 | 13 | 9 | 13 |
| $S_0$ line card - $S_0$ phone | ELIC + QUAT-S + ISAC-S TE | 4 | 8 | 4 | 8 |

**Beware of Arbiter Locking!**

In the state "limited selection", the D-channel arbiter sends the "blocked" information to the terminal from which the last HDLC-frame was received. Since the "blocked" information reaches the terminal with several IOM-frames delay $t_{CCDD}$ (e.g. after $5 \times 125$ µs) the terminal may already have started sending a second HDLC-frame. On reception of the "blocked" information the terminal immediately aborts this frame.

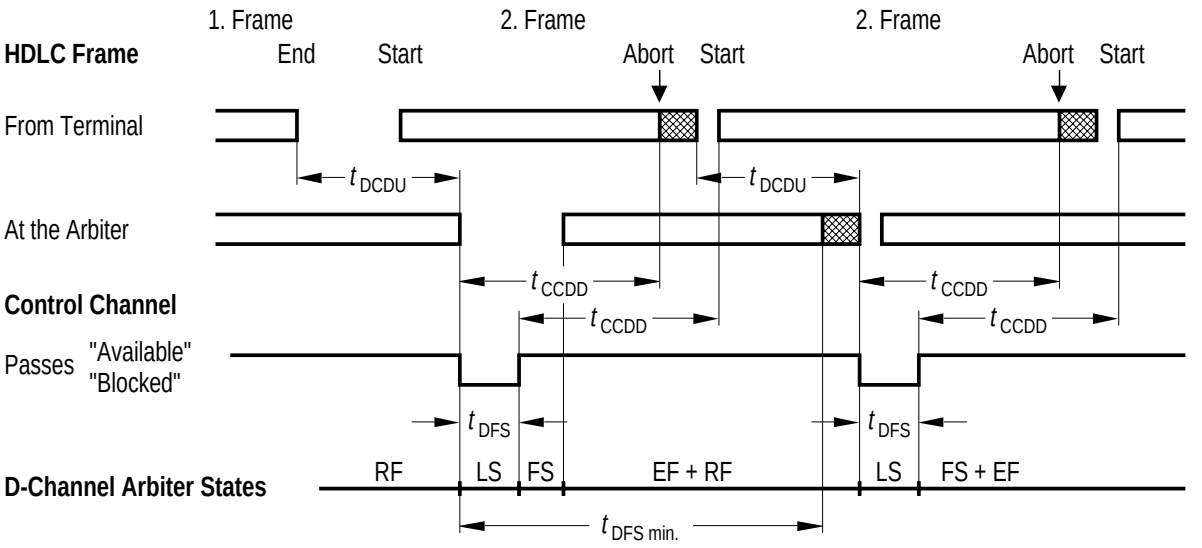
Since the abort sequence of the second frame reaches the ELIC with several frames delay $t_{DCDU}$, the full selection counter value must be set so that the D-channel arbiter re-enters the state "full selection" only after the abort sequence of the second frame has reached the ELIC.

If the D-channel arbiter re-enters the "full selection" state (in which it again sends an "available" information to the terminal) before the abort sequence has reached the ELIC, it would mistake a ’0’ of the second frame as the start of a new frame. When the delayed abort sequence arrives at the ELIC, the D-channel arbiter would then switch back to the state "limited selection" and re-block the terminal. Thus the D-channel arbiter would toggle between sending "available" and "blocked" information to the terminal, forever aborting the terminal‘s frame. The arbiter would have locked.

In order to avoid such a locking situation the time $t_{DFS}$ min. (value in the AMO-register) has to be greater then the maximum delay $t_{CCDD}$ (for the case "available" → "blocked") plus the delay $t_{DCDU}$.

For the QUAT-S a value of 0 is recommended for the suspend counter (register SCV). For the OCTAT-P it is recommended to program SCV=1 in the case of 2 terminals
SCV=0 if one terminal is used.

See the following diagram:




FS = Full Selection
LS = Limited Selection
EF = Expect Frame
RF = Receive Frame

$t_{DFS}$ = Delay for Switch to "Full Selection" (Value in AMO)
$t_{DFS\ min.}$ = Min. Delay for not Locking Condition I
$t_{DCDU}$ = D-Channel Delay Upstream
$t_{CCDD}$ = Control Channel Delay Downstream

Note: If the full selection counter value (AMO : FCC4...0) is not changed from its reset value $00_H$ , then the D-channel arbiter (ASM) skips the state "Limited Selection".

ITD05842

**Figure 45**

### 2.2.8.4 D-Channel Arbiter Co-operating with QUAT-S Circuits

When D-channel multiplexing is used on a $S_0$-bus line card, only the transmit channel selector of the arbiter is used.

The arbiter state machine can be disabled because the QUAT-S offers a self arbitration mechanism between several $S_0$-buses. This feature is implemented by building a wired OR connection between the different E-channels. As a result, the arbitration function does not add additional delays. This means that the priority management on the $S_0$-bus (two classes) still may be used, allowing the mixture of signaling and packet data.

Nevertheless, it still can make sense to use the ELIC arbiter in this configuration. The advantage of using the arbiter is, that if one terminal fails the others will not be blocked.

# 3 Operational Description

The ELIC, designed as a flexible line-card controller, has the following main applications:

– Digital line cards, with the CFI typically configured as IOM-2, IOM-1 (MUX) or SLD.
– Analog line cards, with the CFI typically configured as IOM-2 or SLD.
– Key systems, where the ELIC's ability to mix CFI-configurations is utilized.

To operate the ELIC the user must be familiar with the device's microprocessor interface, interrupt structure and reset logic. Also, the operation of the ELIC's component parts should be understood.

The devices major components are the EPIC-1, the SACCO-A and SACCO-B, and the D-channel arbiter. While EPIC-1, SACCO-A and SACCO-B may all be operated independently of each other, the D-channel arbiter can be used to interface the SACCO-A to the CFI of the EPIC-1. This mode of operation may be considered to utilize the ELIC most extensively. The initialization example, with which this operational description closes, will therefore set the ELIC to operate in this manner.

## 3.1 Microprocessor Interface Operation

The ELIC is programmed via an 8-bit parallel interface that can be selected to be

(1) Motorola type, with control signals $\overline{DS}$, R or $\overline{W}$, and $\overline{CSS}$ or $\overline{CSE}$.
(2) Siemens / Intel non-multiplexed bus type, with control signals $\overline{WR}$, $\overline{RD}$, and $\overline{CSS}$ or $\overline{CSE}$.
(3) Siemens / Intel multiplexed address/data bus type, with control signals ALE, $\overline{WR}$, $\overline{RD}$, and $\overline{CSS}$ or $\overline{CSE}$.

The selection is performed via pin ALE as follows:

ALE tied to $V_{DD} \Rightarrow$ (1)
ALE tied to $V_{SS} \Rightarrow$ (2)
Edge on ALE $\Rightarrow$ (3)

The occurrence of an edge on ALE, either positive or negative, at any time during the operation immediately selects interface type (3). A return to one of the other interface types is only possible by issuing a hardware reset.

With an active $\overline{CSS}$, the addressing selects the FIFOs and registers of the SACCO-A or SACCO-B. With an active $\overline{CSE}$, the addressing selects the memories and/or registers of the

– top level interrupt,
– EPIC-1,
– D-channel arbiter,
– parallel ports, or
– watchdog timer.

When using the Siemens / Intel multiplexed interface, the ELIC can be addressed

– either with even addresses only (i.e. AD0 always 0), which allows data always to be transferred in the low data byte,
– or with even and odd addresses, so that the address range does not extend past $7F_H$.

The selection is performed with the EMOD.DMXAD-bit as follows

DMXAD = 1 ⇒ even addresses only,
DMXAD = 0 ⇒ reduced address range (same addresses as in DEMUX mode).

As a feature of interest to those wishing to use only the EPIC-1 component of the ELIC, note that in the non-multiplexed mode the OMDR.RBS-bit and the A4-address pin are internally ORed. In non-multiplexed mode, it is thus possible to tie the A4-address pin low, and to address the EPIC-1 using the OMDR.RBS-bit and pins A3 … A0.

*Note: It is recommended to tie unused input pins to a defined voltage level.*

### 3.2 Interrupt Structure and Logic

The ELIC-signals events that the µP should know about immediately by emitting an interrupt request on the $\overline{INT}$-line. To indicate the detailed cause of the request a tree of interrupt status registers is provided.

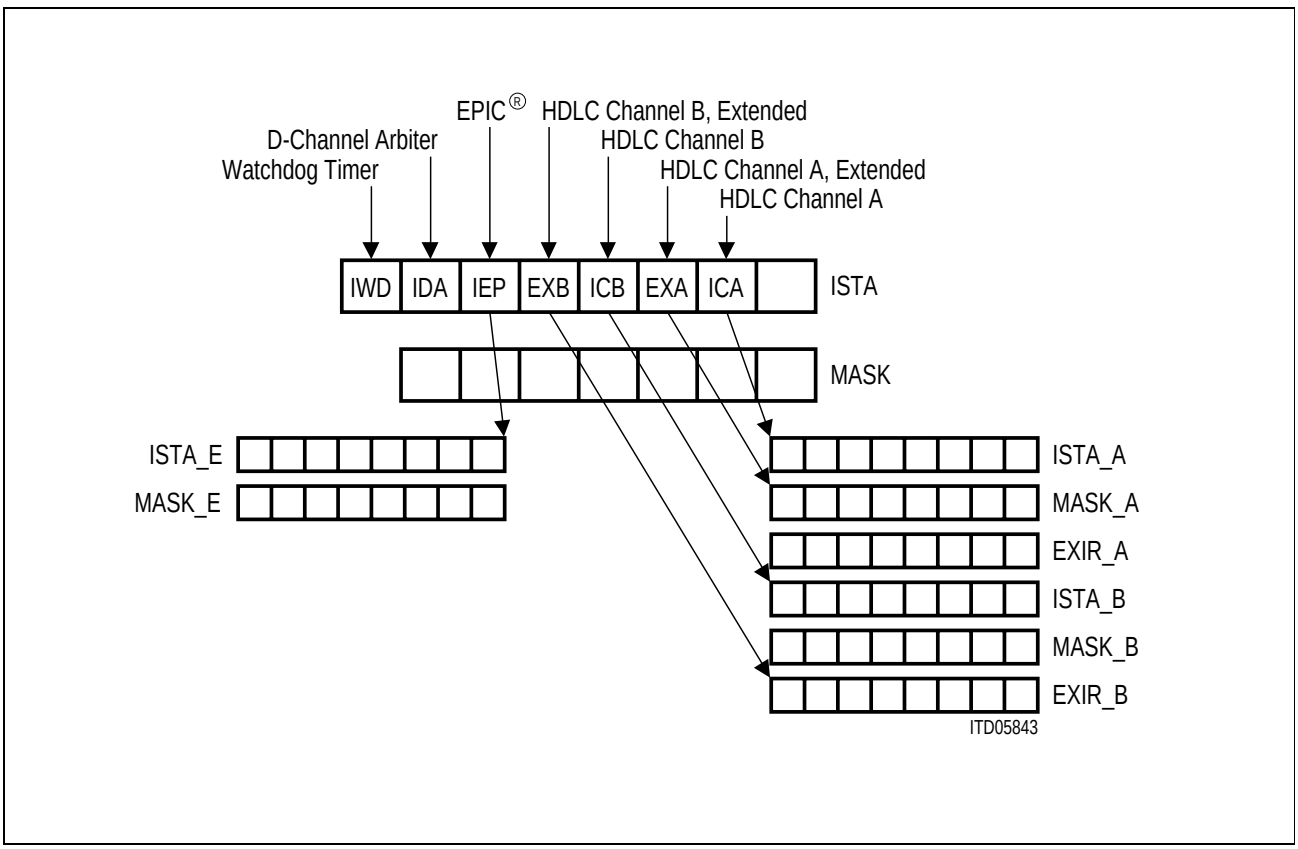



**Figure 46**
**ELIC® Interrupt Structure**

When serving an ELIC-interrupt, the user first reads the top level interrupt status register (ISTA). This register flags which subblock has generated the request. If a subblock can issue different interrupt types a local ISTA/EXIR exists.

A read of the top level ISTA-register resets bits IWD and IDA. The other bits are reset when reading the corresponding local ISTA- or EXIR-registers.

The $\overline{\text{INT}}$-output is level active. It stays active until all interrupt sources have been serviced. If a new status bit is set while an interrupt is being serviced, the $\overline{\text{INT}}$ stays active. However, for the duration of a write access to the MASK-register the $\overline{\text{INT}}$-line is deactivated. When using an edge-triggered interrupt controller, it is thus recommended to rewrite the MASK-register at the end of any interrupt service routine.

### Masking Interrupts

The watchdog timer interrupt can not be masked. Setting the MASK.IDA-bit masks the ISTA.IDA-interrupt: a D-channel arbiter interrupt will then neither activate the $\overline{\text{INT}}$-line nor be indicated in the ISTA-register. Setting the MASK.IEP/EXB/ICB/EXA or ICA-bits only masks the $\overline{\text{INT}}$-line; that is, with a set top level MASK bit these EPIC-1 and SACCO interrupts are indicated in the ISTA-register but they will not activate the $\overline{\text{INT}}$-line.

For the ISTA_E, ISTA_A and ISTA_B registers local masking is also provided. Every interrupt source indicated in these registers can be selectively masked by setting the respective bit of the local MASK-register. Such locally masked interrupts will not be indicated in the local or the top ISTA-register, nor will they activate the $\overline{\text{INT}}$-line.

Locally masked interrupts are internally stored. Thus, resetting the local mask will release the interrupt to be indicated in the local interrupt register, flagged in the top level ISTA-register, and to activate the $\overline{\text{INT}}$-line.

## 3.3 Clocking

To operate properly, the ELIC always requires a PDC-clock.

To synchronize the PCM-side, the ELIC should normally also be provided with a PFS-strobe. In most applications, the DCL and FSC will be output signals of the ELIC, derived from the PDC via prescalers.

If the required CFI-data rate cannot be derived from the PDC, DCL and FSC can also be programmed as input signals. This is achieved by setting the EPIC-1 CMD1:CSS-bit. Frequency and phase of DCL and FSC may then be chosen almost independently of the frequency and phase of PDC and PFS. However, the CFI-clock source **must** still be synchronous to the PCM-interface clock source; i.e. the clock source for the CFI-interface and the clock source for the PCM-interface must be derived from the same master clock.

**Chapter 5.2.2** provides further details on clocking.

### 3.4 Reset

After power-up the ELIC is locked in the "resetting" state. Neither read not write accesses are possible while the ELIC is resetting.

There are two ways to release the ELIC into the operational/programmable state:

a) With an active PDC, 8 PFS-cycles release the ELIC from the "resetting state".
b) With an active PDC, a RESEX-pulse of at least 4 PDC-clock periods also releases the ELIC from the "resetting" state.

On being released from the "resetting" state, the ELIC has completed a reset. Its registers and FIFOs now hold the reset values described in **chapter 4.1**, and can be read from and written to normally.

**Chapter 2.2.4** provides a functional description of the reset logic.

### 3.5 EPIC®-1 Operation

The EPIC-1 component of the ELIC is principally an intelligent switch of PCM-data between two serial interfaces, the system interface (PCM-interface) and the configurable interface (CFI). Up to 128 channels per direction can be switched dynamically between the CFI and the PCM-interfaces. The EPIC-1 performs non-blocking space and time switching for these channels which may have a bandwidth of 16, 32 or 64 kbit/s.

Both interfaces can be programmed to operate at different data rates of up to 8192 kbit/s. The PCM-interface consists of up to four duplex ports with a tristate control signal for each output line. The configurable interface can be selected to provide either four duplex ports or 8 bi-directional (I/O) ports.

The configurable interface incorporates a control block (layer-1 buffer) which allows the µP to gain access to the control channels of an IOM- (ISDN-Oriented Modular) or SLD- (Subscriber Line Data) interface. The EPIC-1 can handle the layer-1 functions buffering the C/I and monitor channels for IOM compatible devices and the feature control and signaling channels for SLD compatible devices. One major application of the EPIC-1 is therefore as line card controller on digital and analog line cards. The layer-1 and codec devices are connected to the CFI, which is then configured to operate as, IOM-2, SLD or multiplexed IOM-1 interface.

The configurable interface of the EPIC-1 can also be configured as plain PCM-interface i.e. without IOM- or SLD-frame structure. Since it’s possible to operate the two serial interfaces at different data rates, the EPIC-1 can then be used to adapt two different PCM- systems.

The EPIC-1 can handle up to 32 ISDN-subscribers with their 2B + D channel structure or up to 64 analog subscribers with their 1B channel structure in IOM-configuration. In SLD- configuration up to 16 analog subscribers can be accommodated.

The system interface is used for the connection to a PCM-back plane. On a typical digital line card, the EPIC-1 switches the ISDN B-channels and, if required, also the D-channels

to the PCM-back plane. Due to its capability to dynamically switch the 16-kbit/s D-channel, the EPIC-1 is one of the fundamental building blocks for networks with either central, decentral or mixed signaling and packet data handling architecture.

### 3.5.1 PCM-Interface

The serial PCM-interface provides up to four duplex ports consisting each of a data transmit (TxD#), a data receive (RxD#) and a tristate control ($\overline{\text{TSC}}$#) line. The transmit direction is also referred to as the upstream direction, whereas the receive direction is referred to as the downstream direction.

Data is transmitted and received at normal TTL / CMOS-levels, the output drivers being of the tristate type. Unassigned time slots may be either be tristated, or programmed to transmit a defined idle value. The selection of the states "high impedance" and "idle value" can be performed with a two bit resolution. This tristate capability allows several devices to be connected together for concentrator functions. If the output driver capability of the EPIC-1 should prove to be insufficient for a specific application, an external driver controlled by the $\overline{\text{TSC}}$# can be connected.

The **PCM-standby function** makes it possible to switch all PCM-output lines to high impedance with a single command. Internally, the device still works normally. Only the output drivers are switched off.

The number of time slots per 8-kHz frame is programmable in a wide range (from 4 to 128). In other words, the **PCM-data rate can range between 256 kbit/s up to 8192 kbit/s**. Since the overall switching capacity is limited to 128 time slots per direction, the number of PCM-ports also depends on the required number of time slots: in case of 32 time slots per frame (2048 kbit/s) for example, four highways are available, in case of 128 time slots per frame (8192 kbit/s), only one highway is available.

The partitioning between number of ports and number of bits per frame is defined by the **PCM-mode.** There are four PCM-modes.

The timing characteristics at the PCM-interface (data rate, bit shift, etc.) can be varied in a wide range, but they are the same for each of the four PCM-ports, i.e. if a data rate of 2048 kbit/s is selected, all four ports run at this data rate of 2048 kbit/s.

The PCM-interface has to be clocked with a **PCM-Data Clock (PDC)** signal having a frequency equal to or twice the selected PCM-data rate. In **single clock rate** operation, a frame consisting of 32 time slots, for example, requires a PDC of 2048 kHz. In **double clock rate** operation, however, the same frame structure would require a PDC of 4096 kHz.

For the synchronization of the time slot structure to an external PCM-system, a **PCM-Framing Signal (PFS)** must be applied. The EPIC-1 evaluates the rising PFS edge to reset the internal time slot counters. In order to adapt the PFS-timing to different timing requirements, the EPIC-1 can latch the PFS-signal with either the rising or the falling PDC- edge. The PFS-signal defines the position of the first bit of the internal PCM-frame.

The actual position of the external upstream and downstream PCM-frames with respect to the framing signal PFS can still be adjusted using the **PCM-offset function** of the EPIC-1. The offset can then be programmed such that PFS marks any bit number of the external frame.

Furthermore it is possible to select either the rising or falling PDC-clock edge for transmitting and sampling the PCM-data.

Usually, the repetition rate of the applied framing pulse PFS is identical to the frame period (125 µs). If this is the case, the **loss of synchronism indication function** can be used to supervise the clock and framing signals for missing or additional clock cycles. The EPIC-1 checks the PFS-period internally against the duration expected from the programmed data rate. If, for example, double clock operation with 32 time slots per frame is programmed, the EPIC-1 expects 512 clock periods within one PFS-period. The synchronous state is reached after the EPIC-1 has detected two consecutive correct frames. The synchronous state is lost if one bad clock cycle is found. The synchronization status (gained or lost) can be read from an internal register and each status change generates an interrupt.

### 3.5.2 Configurable Interface

The serial configurable interface (CFI) can be operated either in duplex modes or in a bi-directional mode.

In **duplex modes** the EPIC-1 provides up to four ports consisting each of a data output (DD#) and a data input (DU#) line. The output pins are called "Data Downstream" pins and the input pins are called "Data Upstream" pins. These modes are especially suited to realize a standard serial PCM-interface (PCM-highway) or to implement an IOM (ISDN-Oriented Modular) interface. The IOM-interface generated by the EPIC-1 offers all the functionality like C/I- and monitor channel handling required for operating all kinds of IOM compatible layer-1 and codec devices.

In **bi-directional mode** the EPIC-1 provides eight bi-directional ports (SIP). Each time slot at any of these ports can individually be programmed as input or output. This mode is mainly intended to realize an SLD-interface (Serial Line Data). In case of an SLD-interface the frame consists of eight time slots where the first four time slots serve as outputs (downstream direction) and the last four serve as inputs (upstream direction). The SLD-interface generated by the EPIC-1 offers signaling and feature control channel handling.

Data is transmitted and received at normal TTL/CMOS-levels at the CFI. **Tristate or open-drain output drivers** can be selected. In case of open-drain drivers, external pull-up resistors are required. Unassigned output time slots may be switched to high impedance or be programmed to transmit a defined idle value. The selection between the states "high impedance" or "idle value" can be performed on a per time slot basis.

The **CFI-standby function** switches all CFI-output lines to high impedance with a single command. Internally the device still works normally, only the output drivers are switched off.

The number of time slots per 8-kHz frame is programmable from 2 to 128. In other words, the **CFI-data rate can range between 128 kbit/s up to 8192 kbit/s**. Since the overall switching capacity is limited to 128 time slots per direction, the number of CFI- ports also depends on the required number of time slots: in case of 32 time slots per frame (2048 kbit/s) for example, four highways are available, in case of 128 time slots per frame (8192 kbit/s), only one highway is available. Usually, the number of bits per 8-kHz frame is an integer multiple of the number of time slots per frame (1 time slot = 8 bits).

The timing characteristics at the CFI (data rate, bit shift, etc.) can be varied in a wide range, but they are the same for each of the four CFI-ports, i.e. if a data rate of 2048 kbit/s is selected, all four ports run at this data rate of 2048 kbit/s. It is thus not possible to have one port used in IOM-2 line card mode (2048 kbit/s) while another port is used in IOM-2 terminal mode (768 kbit/s)!

The clock and framing signals necessary to operate the configurable interface may be derived either from the clock and framing signals of the PCM-interface (PDC and PFS pins), or may be fed in directly via the DCL- and FSC-pins.

In the first case, the CFI-data rate is obtained by internally dividing down the PCM-clock signal PDC. Several prescaler factors are available to obtain the most commonly used data rates. A CFI reference clock (CRCL) is generated out of the PDC-clock. The PCM-framing signal PFS is used to synchronize the CFI-frame structure. Additionally, the EPIC-1 generates clock and framing signals as outputs to operate the connected subscriber circuits such as layer-1 and codec filter devices. The generated data clock DCL has a frequency equal to or twice the CFI-data rate. The generated framing signal FSC can be chosen from a great variety of types to suit the different applications: IOM-2, multiplexed IOM-1, SLD, etc.

Note that if PFS is selected as the framing signal source, the FSC-signal is an output with a fixed timing relationship with respect to the CFI-data lines. The relationship between FSC and the CFI-frame depends only on the selected FSC-output wave form (CMD2- register). The CFI-offset function shifts both the frame and the FSC-output signal with respect to the PFS-signal.

In the second case, the CFI-data rate is derived from the DCL-clock, which is now used as an input signal. The DCL-clock may also first be divided down by internal prescalers before it serves as the CFI reference clock CRCL and before defining the CFI-data rate. The framing signal FSC is used to synchronize the CFI-frame structure.

### 3.5.3 Switching Functions

The major tasks of the EPIC-1 part is to dynamically switch PCM-data between the serial PCM-interface, the serial configurable interface (CFI) and the parallel µP-interface. All possible switching paths are shown in **figure 47**.

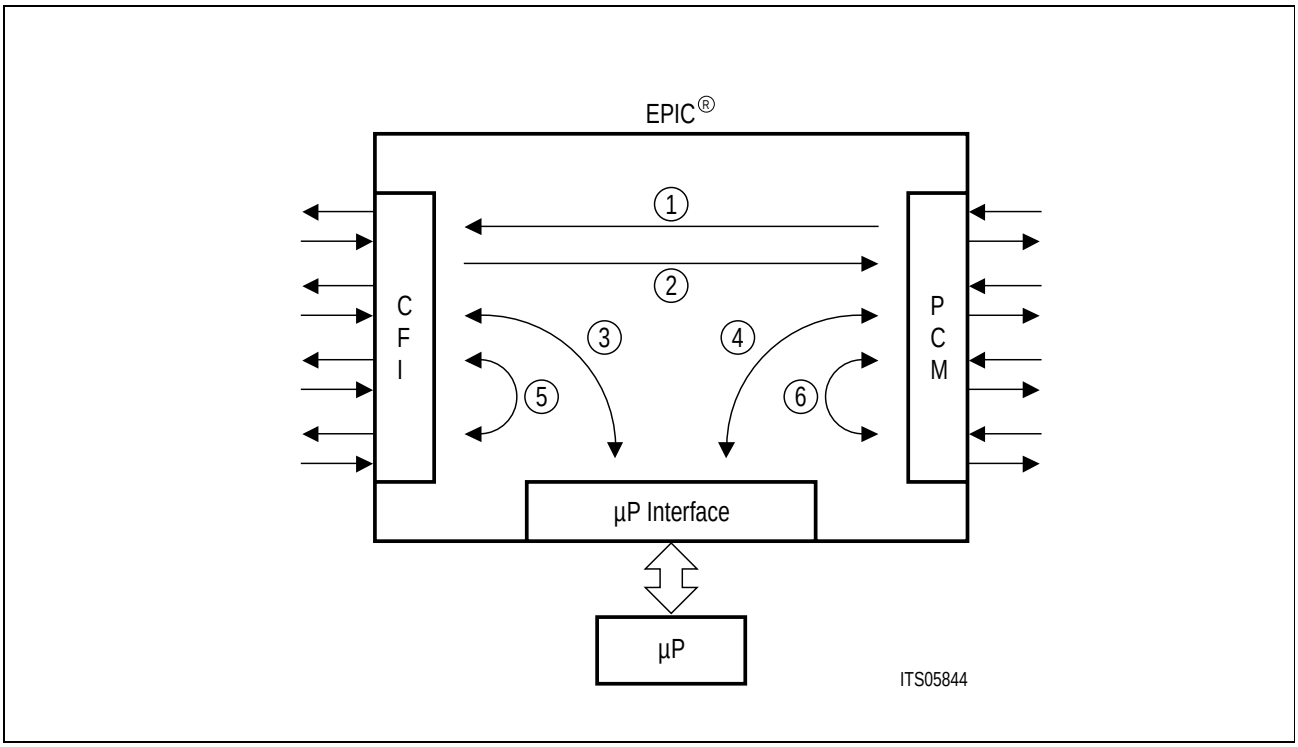



**Figure 47**
**Switching Paths Inside the EPIC®-1**

Note that the time slot selections in upstream direction are completely independent of the time slot selections in downstream direction.

### CFI - PCM Time Slot Assignment

Switching paths 1 and 2 of **figure 47** can be realized for a total number of 128 channels per path, i.e. 128 time slots in upstream and 128 time slots in downstream direction. To establish a connection, the µP writes the addresses of the involved CFI and PCM time slots to the control memory. The actual transfer is then carried out frame by frame without further µP-intervention.

The switching paths 5 and 6 can be realized by programming time slot assignments in the control memory. The total number for such loops is limited to the number of available time slots at the respective opposite interface, i.e. looping back a time slot from CFI to CFI requires a spare upstream PCM time slot and looping back a time slot from PCM to PCM requires a spare downstream and upstream CFI time slot.

Time slot switching is always carried out on 8-bit time slots, the actual position and number of transferred bits can however be limited to 4-bit or 2-bit sub time slots within these 8-bit time slots. On the CFI-side, only one sub time slot per 8-bit time slot can be switched, whereas on the PCM-interface up to 4 independent sub time slots can be switched.

Examples are given in **chapter 5.3**.

### Sub Time Slot Switching

Sub time slot positions at the PCM-interface can be selected at random, i.e. each single PCM time slot-may contain any mixture of 2- and 4-bit sub time slots. A PCM time slot may also contain more than one sub time slot. On the CFI however, two restrictions must be observed:

– Each CFI time slot may contain one and one only sub time slot.
– The sub-slot position for a given bandwidth within the time slot is fixed on a per port basis.

For more detailed information on sub-channel switching please refer to **chapter 5.4.2**.

### µP-Transfer

Switching paths 3 and 4 of **figure 47** can be realized for all available time slots. Path 3 can be implemented by defining the corresponding CFI time slots as "µP-channels" or as "pre-processed channels".

Each single time slot can individually be declared as **"µP-channel"**. If this is the case, the µP can write a static 8-bit value to a downstream time slot which is then transmitted repeatedly in each frame until a new value is loaded. In upstream direction, the µP can read the received 8-bit value whenever required, no interrupts being generated.

The **"pre-processed channel"** option must always be applied to two consecutive time slots. The first of these time slots must have an even time slot number. If two time- slots are declared as "pre-processed channels", the first one can be accessed by the monitor/ feature control handler, which gives access to the frame via a 16-byte FIFO. Although this function is mainly intended for IOM- or SLD-applications, it could also be used to transmit or receive a "burst" of data to or from a 64-kbit/s channel. The second pre-processed time slot, the odd one, is also accessed by the µP. In downstream direction a 4-, 6- or 8-bit static value can be transmitted. In upstream direction the received 8-bit value can be read. Additionally, a change detection mechanism will generate an interrupt upon a change in any of the selected 4, 6 or 8 bits.

Pre-processed channels are usually programmed after Control Memory (CM) reset during device initialization. Resetting the CM sets all CFI time slots to unassigned channels (CM code '0000'). Of course, pre-processed channels can also be initialized or re-initialized in the operational phase of the device.

To program a pair of pre-processed channels the correct code for the selected handling scheme must be written to the CM. **Figure 48** gives an overview of the available pre-processing codes and their application. For further detail, please refer to **chapter 5.5** of the EPIC-1 Application Manual.

*Note: To operate the D-channel arbiter, an IOM-2 configuration with central-, or decentral D-channel handling should be programmed. With the D-channel arbiter enabled, D-channel bits are handled by the SACCO-A.*

| DD Application | Even Control Memory Address MAAR = 0.....0<br>Code Field MACR = 0111... / Data Field MADR = ..... | Odd Control Memory Address MAAR = 0.....1<br>Code Field MACR = 0111... / Data Field MADR = ..... | Output at the Configurable Interface<br>Downstream Preprocessed Channels<br>Even Time-Slot / Odd Time-Slot |
|---|---|---|---|
| Decentral D Channel Handling | 1 0 0 0 \| 1 1 \| C/I \| 1 1 | 1 0 1 1 \| X X X X X X X X | m m m m m m m m \| - - C/I m m<br>Monitor Channel / Control Channel |
| Central D Channel Handling | 1 0 1 0 \| 1 1 \| C/I \| 1 1 | PCM Code for a 2 Bit Sub. Time-Slot \| Pointer to a PCM Time-Slot | m m m m m m m m \| D D C/I m m<br>Monitor Channel / Control Channel |
| 6 Bit Signaling (e.g. analog IOM®) | 1 0 1 0 \| SIG \| 1 1 | 1 0 1 1 \| X X X X X X X X | m m m m m m m m \| SIG m m<br>Monitor Channel / Control Channel |
| 8 Bit Signaling (e.g. SLD) | 1 0 1 0 \| SIG | 1 0 1 1 \| X X X X X X X X | m m m m m m m m \| SIG<br>Feature Control Channel / Signaling Channel |
| SACCO_A D Channel Handling | 1 0 1 0 \| 1 1 \| C/I \| MR 1<br>When using handshaking, set MR = 1 | 1 0 1 1 \| X X X X X X X X | m m m m m m m m \| D D C/I m m<br>Monitor Channel / Control Channel |

ITD05845

| DU Application | Even Control Memory Address MAAR = 1.....1<br>Code Field MACR = 0111... / Data Field MADR = ..... | Odd Control Memory Address MAAR = 1.....1<br>Code Field MACR = 011... / Data Field MADR = ..... | Input from the Configurable Interface<br>Upstream Preprocessed Channels<br>Even Time-Slot / Odd Time-Slot |
|---|---|---|---|
| Decentral D Channel Handling | 1 0 0 0 \| 1 1 \| C/I \| 1 1 | 0 0 0 0 \| X X X X X X X X | m m m m m m m m \| - - C/I m m<br>Monitor Channel / Control Channel |
| Central D Channel Handling | 1 0 0 0 \| 1 1 \| C/I \| 1 1 | PCM Code for a 2 Bit Sub. Time-Slot \| Pointer to a PCM Time-Slot | m m m m m m m m \| D D C/I m m<br>Monitor Channel / Control Channel |
| 6 Bit Signaling (e.g. analog IOM®) | 1 0 1 0 \| SIG Actual Value \| X X | 1 0 1 0 \| SIG Stable Value \| X X | m m m m m m m m \| SIG m m<br>Monitor Channel / Control Channel |
| 8 Bit Signaling (e.g. SLD) | 1 0 1 1 \| SIG Actual Value | 1 0 1 1 \| SIG Stable Value | m m m m m m m m \| SIG<br>Feature Control Channel / Signaling Channel |

ITD05846

m : Monitor channel bits, these bits are treated by the monitor/feature control handler

\- : Inactive sub. time-slot, in downstream direction these bits are tristated (OMDR : COS = 0) or set to logical 1 (OMDR :COS = 1)

C/I : Command/Indication channel, these bits are exchanged between the CFI in/output and the CM data field. A change of the C/I bits in upstream direction causes an interrupt (ISTA : SFI). The address of the change is stored in the CIFIFO

D : D channel, these D channel bit switched to and from the PCM interface, or handled by the SACCO_A, it the D channel arbiter is enabled.

SIG actual value / stable value : Signaling Channel, these bits are exchanged between the CFI in/output and tne CM data field. The SIG value which was present in the last frame is stored as the actual value in the even address CM location. The stable value is updated if a valid change in the actual value has been detected according to the last look algorithm. A change of the SIG stable value in upstream direction causes an interrupt (ISTA : CFI). The address of the change is stored in the CIFIFO.

**Figure 48**
**Pre-processed Channel Codes**

### Synchronous Transfer

For two channels, all switching paths of **figure 47** can also be realized using Synchronous Transfer. The working principle is that the µP specifies an input time slot (source) and an output time slot (destination). Both source and destination time slots can be selected independently from each other at either the PCM- or CFI-interfaces. In each frame, the EPIC-1 first transfers the serial data from the source time slot to an internal data register from where it can be read and if required overwritten or modified by the µP. This data is then fed forward to the destination time slot.

**Chapter 5.7** provides examples of such transfers.

## 3.5.4 Special Functions

### Hardware Timer

The EPIC-1 provides a hardware timer which continuously interrupts the µP after programmable time periods. The timer period can be selected in the range of 250 µs up to 32 ms in multiples of 250 µs. Beside the interrupt generation, the timer can also be used to determine the last look period for 6 and 8-bit signaling channels on IOM-2 and SLD-interfaces and for the generation of an FSC-multiframe signal (see **chapter 5.8.1**).

### Power and Clock Supply Supervision

The + 5 V power supply line and the clock lines are continuously checked by the EPIC-1 for spikes that may disturb its proper operation. If such an inappropriate clocking or power failure occurs, the µP is requested to reinitialize the device.

## 3.6 SACCO-A/B

**Chapter 2.2.8** provides a detailed functional SACCO-description. This operational section will therefore concentrate on outlining how to run these HDLC-controllers.

With the SACCO initialized as outlined in **chapter 3.8.3**, it is ready to transmit and receive data. Data transfer is mainly controlled by commands from the CPU to the SACCO via the CMDR-register, and by interrupt indications from SACCO to CPU. Additional status information, which need not trigger an interrupt, is available in the STAR-register.

## 3.6.1 Data Transmission in Interrupt Mode

In transmit direction $2 \times 32$-byte FIFO-buffers (transmit pools) are provided for each channel. After checking the XFIFO-status by polling the Transmit FIFO Write Enable bit (XFW in STAR-register) or after a Transmit Pool Ready (XPR) interrupt, up to 32 bytes may be entered by the CPU to the XFIFO.

The transmission of a frame can then be started issuing a XTF/XPD or XDD command via the CMDR-register. If prepared data is sent, an end of message indication

(CMDR:XME) must also be set. If transparent or direct data is sent, CMDR:XME may but need not be set. If CMDR:XME is not set, the SACCO will repeatedly request for the next data block by means of a XPR-interrupt as soon as the CPU accessible part of the XFIFO is available. This process will be repeated until the CPU indicates the end of message per command, after which frame transmission is ended by appending the CRC and closing flag sequence.

If no more data is available in the XFIFO prior to the arrival of XME, the transmission of the frame is terminated with an abort sequence and the CPU is notified per interrupt (EXIR:XDU). The frame may also be aborted per software (CMDR:XRES).

**Figure 49** outlines the data transmission sequence from the CPU's point of view:

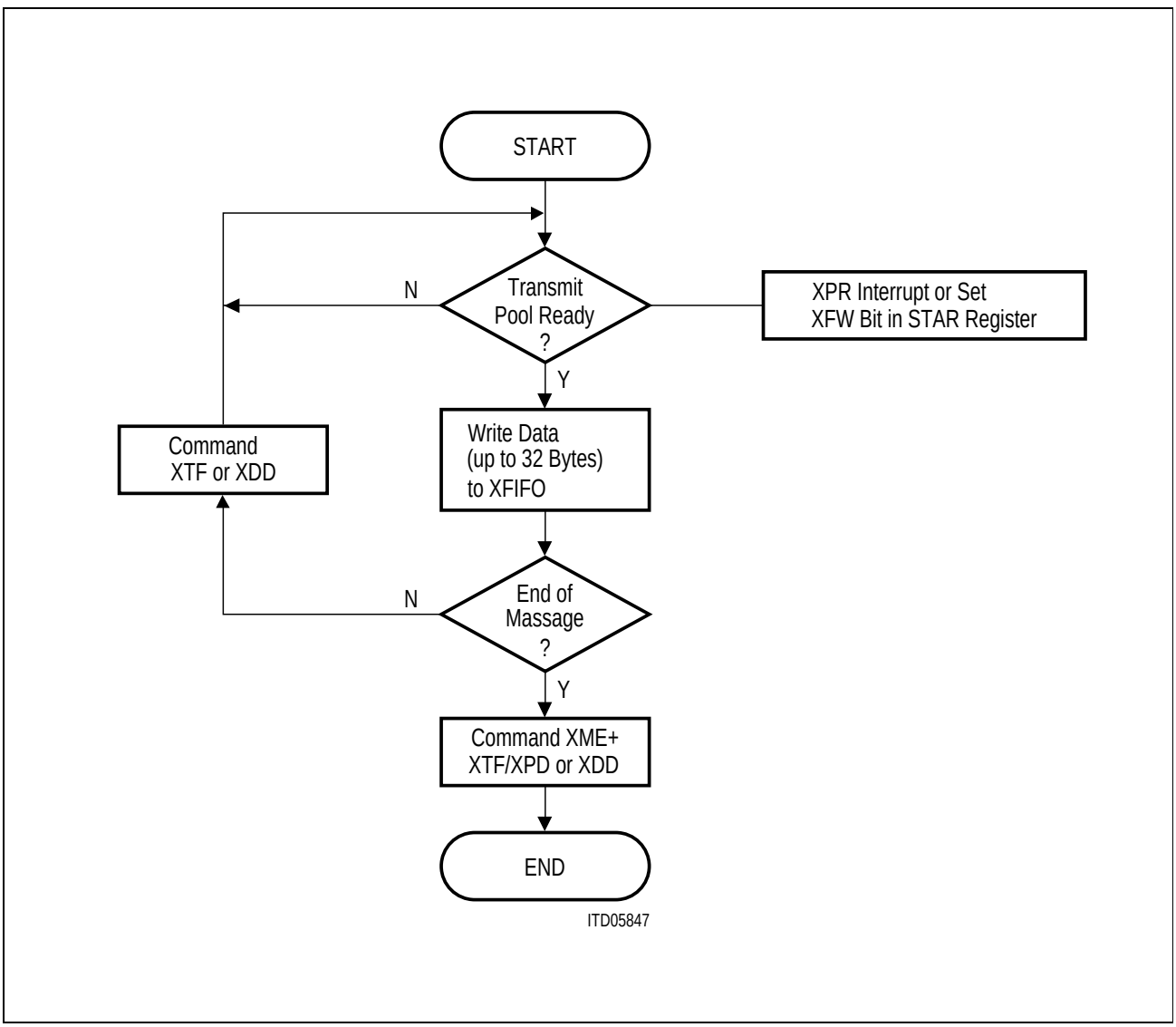



**Figure 49**
**Interrupt Driven Transmission Sequence (flow diagram)**

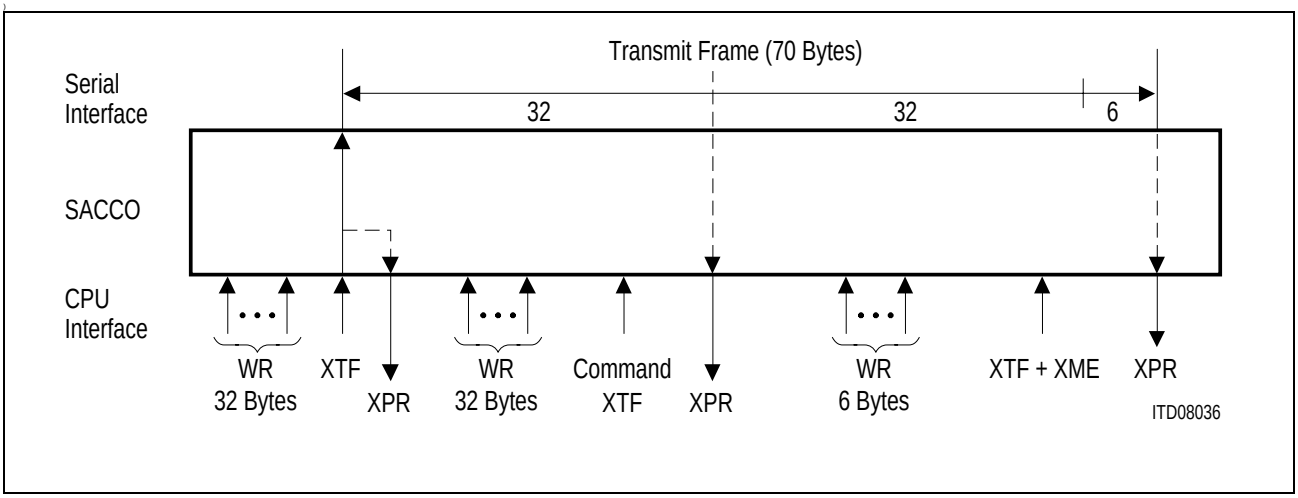



**Figure 50**
**Interrupt Driven Transmission Sequence Example**

### 3.6.2 Data Transmission in DMA-Mode

Prior to data transmission, the length of the frame to be transmitted must be programmed via the Transmit Byte Count Registers (XBCH, XBCL). The resulting byte count equals the programmed value plus one byte. Since 12 bits are provided via XBCH, XBCL (XBC11 … XBC0) a frame length between 1 and 4096 bytes can be selected.

Having written the Transmit Byte Counter Registers, data transmission can be initiated by command XTF/XPD or XDD. The SACCO will then autonomously request the correct amount of write bus cycles by activating the DRQT-line. Depending on the programmed frame length, block data transfers of n × 32-bytes + remainder are requested every time the 32 byte transmit pool is accessible to the DMA-controller.

The following figure gives an example of a DMA driven transmission sequence with a frame length of 70 bytes, i.e. programmed transmit byte count (XCNT) equal 69 bytes.

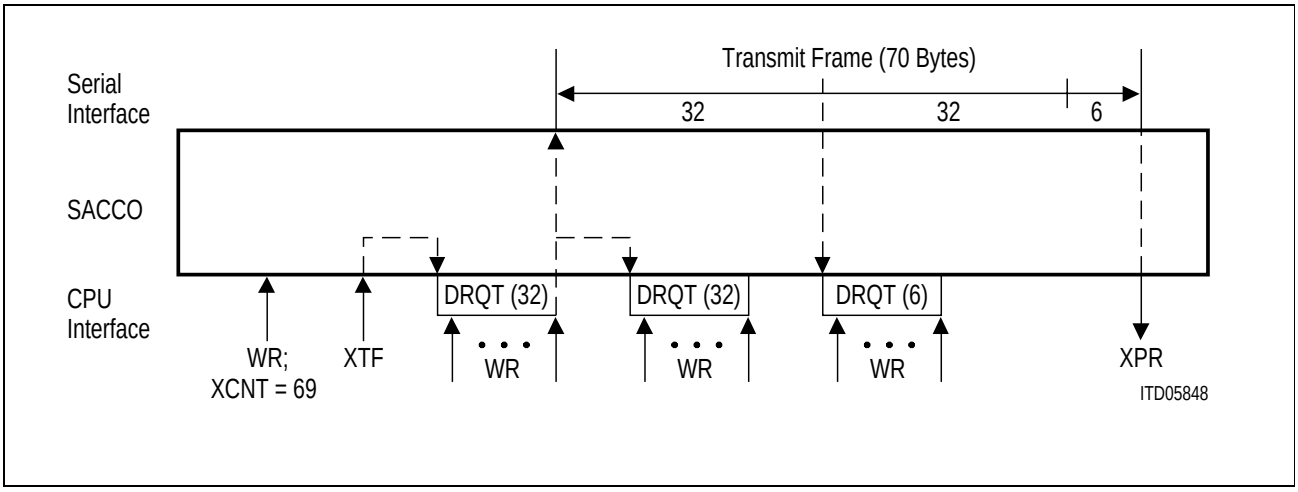



**Figure 51**
**DMA Driven Transmission Example**

### 3.6.3 Data Reception in Interrupt Mode

In receive direction 2 × 32-byte FIFO-buffers (receive pools) are also provided for each channel. There are two different interrupt indications concerned with the reception of data:

– A RPF (Receive Pool Full) interrupt indicates that a 32-byte block of data can be read from the RFIFO with the received message not yet complete.
– A RME (Receive Message End) interrupt indicates that the reception of one message is completed, i.e. either
  – one message with less than 32 bytes, or the
  – last part of a message with more than 32 bytes
  is stored in the CPU accessible part of the RFIFO.

The CPU must handle the RPF-interrupt before additional 32 bytes are received via the serial interface, as failure to do so causes a RDO (Receive Data Overflow).

Status information about the received frame is appended to the frame in the RFIFO. This status information follows the format of the RSTA-register, unless using the SACCO-A in clock mode 3. The CPU can read the length of the received message (including the appended Receive Status byte) from the RBCH- and RBCL-registers.

**After the received data has been read from the RFIFO, this must be explicitly acknowledged by the CPU issuing a RMC- (Receive Message Complete) command!**

The following figure gives an example of an interrupt controlled reception sequence, supposing that a long frame (66 bytes) followed by a short frame (6 bytes) are received.

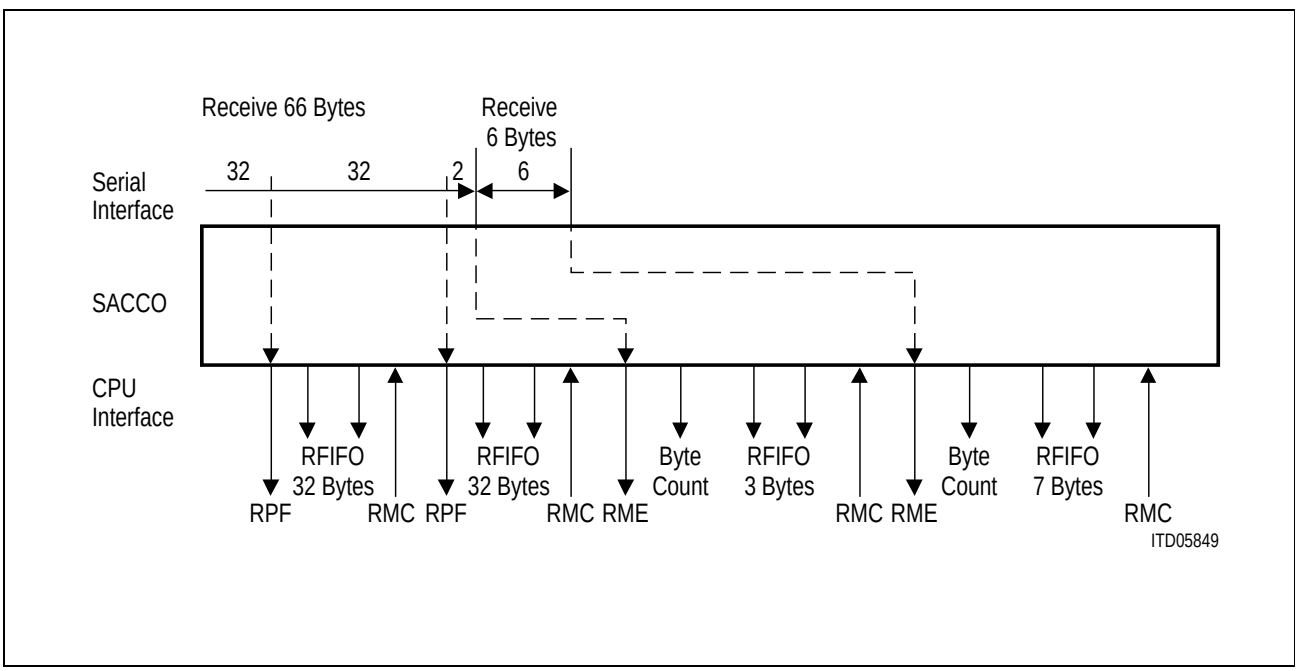



**Figure 52**
**Interrupt Driven Reception Example**

### 3.6.4 Data Reception in DMA-Mode

If the RFIFO contains 32 bytes, the SACCO autonomously requests a block DMA-transfer by activating the DRQR-line. This forces the DMA-controller to continuously perform bus cycles until 32 bytes are transferred from the SACCO to the system memory.

If the RFIFO contains less than 32 bytes (one short frame or the last part of a long frame) the SACCO requests a block data transfer depending on the contents of the RFIFO according to the following table:

| RFIFO Contents (in bytes) | DMA Request (in bytes) |
|---|---|
| 1, 2, 3 | 4 |
| 4, 5, 6, 7 | 8 |
| 8 - 15 | 16 |
| 16 - 32 | 32 |

After the DMA-controller has been set up for the reception of the next frame, the CPU must issue a RMC-command to acknowledge the completion of the receive frame processing. Prior to the reception of this RMC, the SACCO will not initiate further DMA-cycles by activating the DRQR-line.

The following figure gives an example of a DMA controlled reception sequence supposing that a long frame (66 bytes) followed by a short frame (6 byte) are received.

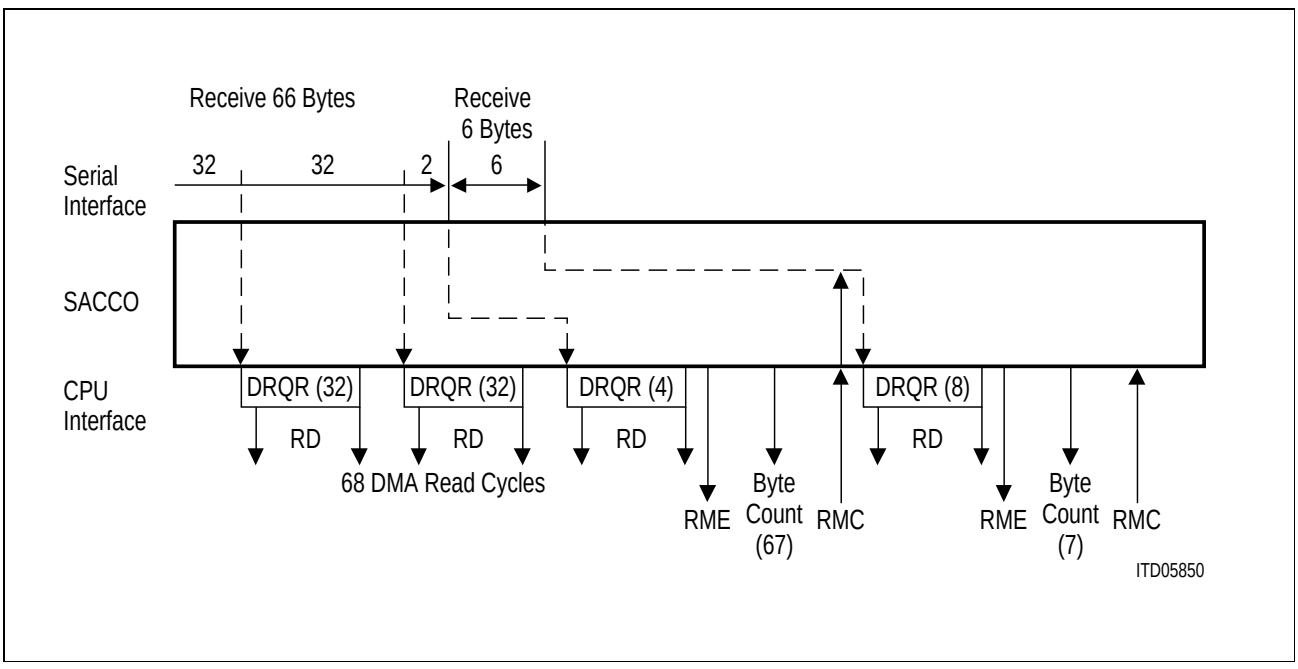



**Figure 53**
**DMA-Driven Reception Example**

## 3.7 D-Channel Arbiter

The D-channel arbiter links the SACCO-A to the CFI of the EPIC-1. EPIC-1 and SACCO-A should therefore be initialized before setting up the D-channel arbiter, as demonstrated in **chapter 3.8**.

In downstream direction, the D-channel arbiter distributes data from the SACCO-A to the selected subscribers. In upstream direction, the D-channel arbiter ensures that the SACCO-A receives data from only a single correspondent at a time. Given proper initialization, the operation of the D-channel arbiter is largely transparent. The user of the ELIC can thus concentrate on operating the SACCO-A as described in **chapters 2.2.8 and 3.6**.

For the D-channel arbiter to operate as desired, the SACCO-A must be set clock mode 3 and inter frame timefill set to all ’1’s. It is also recommended that the SACCO-A **not** be set into auto-mode when communicating with downstream subscribers. The EPIC-1’s CFI should be configured to follow the line card IOM-2 protocol, i.e.:

– CFI mode 0
– 2-Mbit/s data rate (usually with a double rate clock)
– 256 bits per frame and port (8 subscribers per port)
– 16-kbit/s D-channels positioned as bits 7,6 of time slots $(n \times 4) - 1$ for n = 1 … 8

### 3.7.1 SACCO-A Transmission

Sending data from the SACCO-A to downstream subscribers is handled by the transmit channel master of the D-channel arbiter. The downstream Control Memory (CM) Code for subscribers who may be sent data by the SACCO-A must be set to '1010'$_B$ for the even time slot and to '1011'$_B$ for the odd time slot. The CM-data of the even time slot should be programmed to "11 C/I-code 11". For example, a CM-data entry of '11000011' would set the C/I-code to '0000'. Refer to **figure 48**.

If data is to be sent to a single subscriber (no broadcasting), this subscriber must be selected in the XDC-register. Whenever the subscribers D-channel is to be output at the ELIC’s CFI, the transmit channel master provides a 2-bit transmit strobe to the SACCO-A. Every frame, 2 data bits are thus strobed from the SACCO-A into the subscriber’s D-channel, when the SACCO-A has been commanded to send data. As the subscribers D-channel recurs every 125 µs, the data is transmitted from the SACCO-A to the subscriber at a rate of 16 kbit/s. If the SACCO-A has no data to send, it sends its inter frame timefill ('1's) to the subscriber when strobed by the transmit channel master.

With the XDC.BCT bit set (broadcasting), the BCG-registers are used to select the subscribers to whom the SACCO’s data is to be sent. The SACCO’s output is first copied to an internal buffer. From this buffer, the data is strobed, 2 bits at a time, to all selected subscribers. When the SACCO-A has no data to send, its inter frame timefill ('1's) is copied to the buffer and strobed into the D-channels of the selected subscribers.

### 3.7.2 SACCO-A Reception

Subscribers who are to participate in the D-channel arbitration for the SACCO-A must send ’all 1s’ as inter frame timefill of their D-channels. Flags or idle codes other than ’all 1s’ are not permitted as inter frame timefill. For any participating subscriber, the "blocked" code must be programmed into the downstream Control Memory (CM). Also, the subscriber’s D-channel must be enabled in the DCE-register.

In the full selection state, the D-channel arbiter overwrites the downstream "blocked" code of enabled subscribers with the "available" code. On the upstream CFI-input lines, the D-channel arbiter monitors all D-channels enabled in the DCE-registers.

When the D-channel arbiter detects a '0' on any monitored D-channel it assumes this to be the start of an opening flag. It therefore strobes the D-channel data of this subscriber to the SACCO-A and starts the Suspend Counter. For this selected subscriber, the D-channel arbiter continues to overwrite the downstream "blocked" code with the "available" code. However, all other enabled subscribers are now passed the "blocked" code from the downstream CM.

If the SACCO-A does indeed receive an HDLC-frame - complete or aborted - from the selected subscriber, the Suspend Counter is reset. While the SACCO-A receives data from the selected subscriber, the "blocked" code stops all other subscribers from sending data to the SACCO-A. After the SACCO-A has received a closing flag or abort sequence for the subscribers frame, the D-channel arbiter stops strobing the subscriber’s data to the SACCO-A and enters the limited selection state.

If, after the initial '0', the SACCO-A does not receive an HDLC-frame - complete or aborted - from the selected subscriber, it does not reset the Suspend Counter. Eventually, the Suspend Counter under flows, setting off the ISTA.IDA-interrupt. The subscriber who sent the erroneous '0' can then be identified in the ASTATE-register. Any subscriber who frequently sends erroneous '0's should be disabled from the DCE, and the cause of the error investigated. After the ISTA.IDA-interrupt, the SACCO-A receiver must be reset to resume operation in the full selection state.

The limited selection state is identical to the full selection state, except that the subscriber who last sent data to the SACCO-A is excluded from the arbitration. This prevents any single subscriber from constantly keeping the SACCO-A busy. The "blocked" code of the CM is passed to the excluded subscriber, while the D-channel arbiter sends all other enabled subscribers the "available" code. All enabled subscribers - except the one excluded - are monitored for the starting '0' of an opening flag. How long the exclusion lasts can be programmed in the AMO-register. If none of the monitored subscribers has started sending data during this time, the D-channel Arbiter re-enters the full selection state.

## 3.8 Initialization Procedure

For proper initialization of the ELIC the following procedure is recommended:

### 3.8.1 Hardware Reset

A reset pulse can be applied at the RESEX-pin for at least 4 PDC-clock cycles. The reset pulse sets all registers to their reset values (cf. **chapter 4.1**).
Note that in this state DCL and FSC do not deliver any clock signals.

### 3.8.2 EPIC®-1 Initialization

As the EPIC-1 forms the core of the ELIC, it should usually be programmed first.

#### 3.8.2.1 Register Initialization

The PCM- and CFI-configuration registers (PMOD, PBNR, …, CMD1, CMD2, …) should be programmed to the values required for the application. The correct setting of the PCM- and CFI-registers is important in order to obtain a reference clock (RCL) which is consistent with the externally applied clock signals.
The state of the operation mode (OMDR:OMS1..0 bits) does not matter for this programming step.

PMOD = PCM-mode, timing characteristics, etc.
PBNR = Number of bits per PCM-frame
POFD = PCM-offset downstream
POFU = PCM-offset upstream
PCSR = PCM-timing
CMD1 = CFI-mode, timing characteristics, etc.
CMD2 = CFI-timing
CBNR = Number of bits per CFI-frame
CTAR = CFI-offset (time slots)
CBSR = CFI-offset (bits)
CSCR = CFI-sub channel positions

#### 3.8.2.2 Control Memory Reset

Since the hardware reset does not affect the EPIC-1 memories (Control and Data Memories), it is mandatory to perform a "software reset" of the CM. The CM-code ’0000’ (unassigned channel) should be written to each location of the CM. The data written to the CM-data field is then don’t care, e.g. $FF_H$.

OMDR:OMS1..0 must be to $'00'_B$ for this procedure (reset value).

MADR = $FF_H$
MACR = $70_H$
Wait for EPIC.STAR:MAC = 0

The resetting of the complete CM takes 256 RCL-clock cycles. During this time, the EPIC.STAR:MAC-bit is set to logical 1.

#### 3.8.2.3 Initialization of Pre-processed Channels

After the CM-reset, all CFI time slots are unassigned. If the CFI is used as a plain PCM-interface, i.e. containing only switched channels (B-channels), the initialization steps below are not required. The initialization of pre-processed channels applies only to IOM- or SLD-applications.

An IOM- or SLD-"channel" consists of four consecutive time slots. The first two time slots, the B-channels need not be initialized since they are already set to unassigned channels by the CM-reset command. Later, in the application phase of the software, the B-channels can be dynamically switched according to system requirements. The last two time slots of such an IOM- or SLD-channel, the pre-processed channels must be initialized for the desired functionality. There are five options that can be selected:

**Table 16**
**Pre-processed Channel Options at the CFI**

| Even CFI Time Slot | Odd CFI Time Slot | Main Application |
|---|---|---|
| Monitor/feature control channel | 4-bit C/I-channel, D-channel handled by SACCO-A and D-ch. arbiter | IOM-1 or IOM-2 digital subscriber |
| Monitor/feature control channel | 4-bit C/I-channel, D-channel not switched (decentral D-ch. handling) | IOM-1 or IOM-2 digital subscriber |
| Monitor/feature control channel | 4-bit C/I-channel, D-channel switched (central D-ch. handling) | IOM-1 or IOM-2 digital subscriber |
| Monitor/feature control channel | 6-bit SIG-channel | IOM-2, analog subscriber |
| Monitor/feature control channel | 8-bit SIG/channel | SLD, analog subscriber |

Also refer to **figure 49**.

**Example**

In CFI-mode 0 all four CFI-ports shall be initialized as IOM-2 ports with a 4-bit C/I-field and D-channel handling by the SACCO-A.

CFI time slots 0, 1, 4, 5, 8, 9 … 28, 29 of each port are B-channels and need not to be initialized.

CFI time slots 2, 3, 6, 7, 10, 11, …, 30, 31 of each port are pre-processed channels and need to be initialized:

**CFI-port 0, time slot 2 (even), downstream**

MADR = $FF_H$ ; the C/I-value ’1111’ will be transmitted upon CFI-activation
MAAR = $08_H$ ; addresses ts 2 down
MACR = $7A_H$ ; CM-code ’1010’
Wait for STAR:MAC = 0

**CFI-port 0, time slot 3 (odd), downstream**

MADR = $FF_H$ ; don’t care
MAAR = $09_H$ ; addresses ts 3 down
MACR = $7B_H$ ; CM-code '1011'
Wait for STAR:MAC = 0

**CFI-port 0, time slot 2 (even), upstream**

MADR = $FF_H$ ; the C/I-value '1111' is expected upon CFI-activation
MAAR = $88_H$ ; address ts 2 up
MACR = $78_H$ ; CM-code '1000'
Wait for STAR:MAC = 0

**CFI-port 0, time slot 3 (odd), upstream**

MADR = $FF_H$ ; don’t care
MAAR = $89_H$ ; address ts 3 up
MACR = $70_H$ ; CM-code '0000'
Wait for STAR:MAC = 0

Repeat the above programming steps for the remaining CFI-ports and time slots.

This procedure can be speeded up by selecting the CM-initialization mode (OMDR:OMS1..0 = 10). If this selection is made, the access time to a single memory location is reduced to 2.5 RCL-cycles. The complete initialization time for 32 IOM-2 channels is then reduced to $128 \times 0.61\ \mu s = 78\ \mu s$

#### 3.8.2.4 Initialization of the Upstream Data Memory (DM) Tristate Field

For each PCM time slot the tristate field defines whether the contents of the DM-data field are to be transmitted (low impedance), or whether the PCM time slot shall be set to high impedance. The contents of the tristate field is not modified by a hardware reset. In order to have all PCM time slots set to high impedance upon the activation of the PCM-interface, each location of the tristate field must be loaded with the value ’0000’. For this purpose, the "tristate reset" command can be used:

OMDR = $C0_H$ ; OMS1..0 = 11, normal mode
MADR = $0\mathbf{0}_H$ ; code field value $'0000'_B$
MACR = $68_H$ ; MOC-code to initialize all tristate locations ($1101_B$)
Wait for STAR:MAC = 0

The initialization of the complete tristate field takes 1035 RCL-cycles.

*Note: It is also possible to program the value ’1111’ to the tristate field in order to have all time slots switched to low impedance upon the activation of the PCM-interface.*

*Note: While OMDR:PSB = 0, all PCM-output drivers are set to high impedance, regardless of the values written to the tristate field.*

### 3.8.3 SACCO-Initialization

To initialize the SACCO, the CPU has to write a minimum set of registers. Depending on the operating mode and on the features required, an optional set of register must also be initialized.

As the first register to be initialized, the MODE-register defines operating and address mode. If data reception shall be performed, the receiver must be activated by setting the RAC-bit. Depending on the mode selected, the following registers must also be defined:

**Table 17**
**Mode Dependent Register Set-up**

| | 1 Byte Address | 2 Byte Address |
|---|---|---|
| Transparent mode 1 | | RAH1<br>RAH2 |
| Non-auto mode | RAH1 = $00_H$<br>RAH2 = $00_H$<br>RAL1<br>RAL2 | RAH1<br>RAH2<br>RAL1<br>RAL2 |
| Auto-mode | XAD1<br>XAD2<br>RAH1 = $00_H$<br>RAH2<br>RAL1<br>RAL2 | XAD1<br>XAD2<br>RAH1<br>RAH2<br>RAL1<br>RAL2 |

The second minimum register to be initialized is the CCR2. In combination with the CCR1, the CCR2 defines the configuration of the serial port. It also allows enabling the RFS-interrupt.

If bus configuration is selected, the external serial bus must be connected to the C×D-pin for collision detection. In point-to-point configuration, the C×D-pin must be tied to ground if no "clear to send" function is provided via a modem.

Depending on the features desired, the following registers may also require initializing before powering up the SACCO:

**Table 18**
**Feature Dependent Register Set-up**

| Feature | Register(s) |
|---|---|
| Clock mode 2 | TSAR, TSAC, XCCR, RCCR |
| Masking selected interrupts | MASK |
| DMA controlled data transfer | XBCH |
| Check on receive length | RLCR |

The CCR1 is the final minimum register that has to be programmed to initialize the SACCO. In addition to defining the serial port configuration, the CCR1 sets the clock mode and allows the CPU to power-up or power-down the SACCO.

In power-down mode all internal clocks are disabled, and no interrupts are forwarded to the CPU. This state can be used as standby mode for reduced power consumption.

Switching between power-up or power-down mode has no effect on the contents of the register, i.e. the internal state remains stored.

After power-up of the SACCO, the CPU should bring the transmitter and receiver to a defined state by issuing a XRES (transmitter reset) and RHR (receiver reset) command via the CMDR-register. The SACCO will then be ready to transmit and receive data.

The CPU controls the data transfer phase mainly by commands to the SACCO via the CMDR-register, and by interrupt indications from the SACCO to the CPU. Status information that does not trigger an interrupt is constantly available in the STAR-register.

### 3.8.4 Initialization of D-Channel Arbiter

The D-channel arbiter links the SACCO-A to the CFI of the EPIC-1 part of the ELIC. Thus the EPIC-1 and SACCO-parts of the ELIC should be initialized before initializing the D-channel arbiter.

For subscribers wishing to communicate with the SACCO-A, the correct pre-processed channel code must have been programmed in the EPIC-1's control memory. In downstream direction, this code is CMC = 1010 for the even time slot and CMC = 1011 for the odd time slot. In upstream direction, any pre-processed channel code is also valid for arbiter operation. This is shown in **figure 48** of **chapter 3.5.3**. For an example refer to **chapter 3.8.2.3**.

If the MR-bit is used to block downstream subscribers, the blocking code MR = '0'$_B$ can be written as MADR = '11xxxx01'$_B$ when initializing the even downstream time slot. The 'x' stand for the C/I-code. This also is shown in **figure 48**.

If the C/I-code is used to block downstream subscribers, such subscribers must be activated with the C/I-code '1100'$_B$, not '1000'$_B$.

The SACCO-A must be initialized to clock mode 3 to communicate with downstream subscribers. In clock mode 3, the SACCO-A receives its input and transmit its output via the D-channel arbiter. If the CCR2.T×DE-bit is set, the SACCO-A's output is transmitted at the T×DA-pin in addition to being transmitted via the D-channel arbiter.

Once EPIC-1 and SACCO-A have been correctly initialized, writing the subscriber's address into the XDC-register allows the SACCO-A to send the subscriber data. By setting the XDC.BCT-bit and programming the BCG-registers, the SACCO-A can transmit its data to several subscribers.

To strobe upstream data from the CFI-interface to the SACCO-A's receiver, the AMO-register must be programmed for the desired functionality. Subscribers who are to be allowed to send data must be enabled via the DCE-registers. If a subscriber tries to send data during the initialization of the upstream D-channel arbiter, a ISTA.IDA-interrupt may occur. This interrupt can be cleared by resetting the SACCO-A receiver.

*Note: The EPIC-1 and SACCO-A must be initialized correctly before the D-channel arbiter can operate properly. Particular care must be given to programming the EPIC-1's Control Memory (CM) with the required CM-Codes (CMCs).*

*Note: The upstream and downstream D-channel arbiter initializations are independent of each other.*

### 3.8.5 Activation of the PCM- and CFI-Interfaces

With EPIC-1, SACCO-A and D-channel arbiter all configured to the system requirements, the PCM- and CFI-interface can be switched to the operational mode.

The OMDR:OMS1..0 bits must be set (if this has not already be done) to the normal operation mode (OMS1..0 = 11). When doing this, the PCM-framing interrupt (ISTA:PFI) will be enabled. If the applied clock and framing signals are in accordance with the values programmed to the PCM-registers, the PFI-interrupt will be generated (if not masked). When reading the status register, the STAR:PSS-bit will be set to logical 1.

To enable the PCM-output drivers set OMDR:PSB = 1. The CFI-interface is activated by programming OMDR:CSB = 1. This enables the output clock and framing signals (DCL and FSC), if these have been programmed as outputs. It also enables the CFI-output drivers. The output driver type can be selected between "open drain" and "tristate" with the OMDR:COS-bit.

**Example:** Activation of the EPIC-1 part of the ELIC for a typical IOM-2 application:

OMDR = $EE_H$; Normal operation mode (OMS1..0 = 11)
PCM-interface active (PSB = 1)
PCM-test loop disabled (PTL = 0)
CFI-output drivers: open drain (COS = 1)
Monitor handshake protocol selected (MFPS = 1)
CFI active (CSB = 1)
Access to EPIC-1 registers via address pins A4..A0 (RBS = 0)

### 3.8.6 Initialization Example

In this sample initialization the ELIC is set up to handle a digital IOM-2 subscriber. The interfaces of the ELIC are shown below:

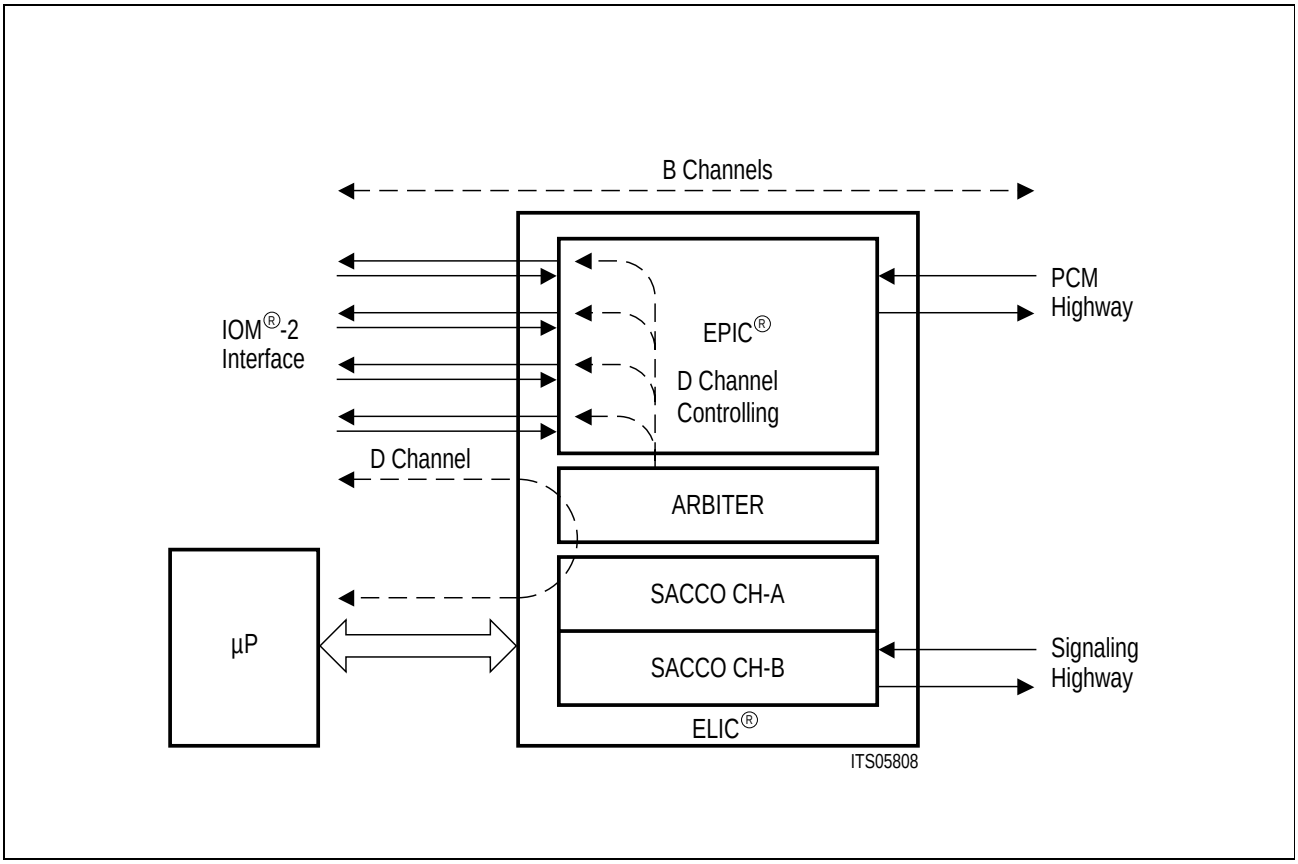



**Figure 54**
**ELIC® Interfaces for Initialization Example**

The subscriber uses the ELIC's CFI-port 0, channel 0 (time slots 0 - 3). The subscriber's upstream $B_1$-channel is to be switched to PCM-port 0, time slot 5. The subscriber's upstream $B_2$-channel is to be looped back to the subscriber on the downstream $B_1$-channel. The subscriber's downstream $B_2$-channel is to be switched from PCM-port 0, time slot 1. The subscriber's HDLC-data is exchanged via the D-channel with the SACCO-A. Monitor and C/I-channels are to be handled via the ELIC.

The SACCO-B communicates via a dedicated signaling highway with a non-PBC group controller. A 4-MHz clock is input as PDC and HDCB.

Port 1 of the ELIC is to be used as active low output. Thus the port should be linked to pull-up resistors.

Write PCON1 = $FF_H$
Write PORT1 = $FF_H$

### 3.8.6.1 EPIC®-1 Initialization Example

Configure PCM-side of ELIC:

| | | |
|---|---|---|
| Write | PMOD = $44_H$ | PCM-mode 1, single clock rate, PFS evaluated with falling edge of PDC, R×D0 = logical input port 0 |
| Write | PBNR = $FF_H$ | 512 bits per PCM-frame |
| Write | POFD = $F0_H$ | the internal PFS marks downstream bit 6, ts 0 (second bit of frame) |
| Write | POFU = $18_H$ | the internal PFS marks upstream bit 6, ts 0 (second bit of frame) |
| Write | PCSR = $45_H$ | no clock shift; PCM-data sampled with falling, transmitted with rising PDC |

Configure CFI-side of ELIC:

| | | |
|---|---|---|
| Write | CMD1 = $20_H$ | PDC and PFS used as clock and framing source for the CFI; CRCL = PDC; CFI-mode 0 |
| Write | CMD2 = $D0_H$ | FSC shaped for IOM-2 interface; DCL = 2 × data rate; CFI-data received with falling, transmitted with rising CRCL |
| Write | CBNR = $FF_H$ | 256 bits per CFI-frame |
| Write | CTAR = $02_H$ | PFS is to mark CFI time slot 0 |
| Write | CBSR = $20_H$ | PFS is to mark bit 7 of CFI time slot 0; no shift of CFI-upstream data relative to CFI-downstream data |
| Write | CSCR = $00_H$ | 2-bit channels located in position 7, 6 on all CFI-ports |

Reset EPIC-1 Control Memory (CM) to $FF_H$:

| | | |
|---|---|---|
| Write | MADR = $FF_H$ | |
| Write | MACR = $70_H$ | |

Initialize EPIC-1 CM:

| | | |
|---|---|---|
| Write | OMDR = $80_H$ | set EPIC-1 from CM-reset mode into CM-initialization mode |

The subscriber's upstream $B_1$-channel is switched to PCM-port 0, time slot 5

| | | |
|---|---|---|
| Write | MADR = $89_H$ | connection to PCM-port 0, time slot 5 |
| Write | MAAR = $80_H$ | from upstream CFI-port 0, time slot 0 |
| Write | MACR = $71_H$ | write CM-data adressed by MAAR with content of MADR; write CM-code addressed by MAAR with '0001'$_B$ (code for a simple 64-kbit/s connection) |
| Read | STAR | Wait for STAR:MAC = 0 |

The subscriber's upstream $B_2$-channel is internally looped via PCM-port 1, time slot 1

| | | |
|---|---|---|
| Write | MADR = $85_H$ | loop to PCM-port 1, time slot 1 |
| Write | MAAR = $81_H$ | from upstream CFI-port 0, time slot 1 |
| Write | MACR = $71_H$ | write CM-data addressed by MAAR with content of MADR; write CM-code addressed by MAAR with '0001'$_B$ (code for a simple 64 kbit/s connection) |
| Read | STAR | Wait for STAR:MAC = 0 |

The subscriber's upstream time slots 2 and 3 are initialized as monitor and C/I-channels with decentral D-channel handling

| | | |
|---|---|---|
| Write | MADR = $FF_H$ | received C/I-code to be compared to '1111'$_B$ |
| Write | MAAR = $88_H$ | from upstream CFI-port 0, time slot 2 |
| Write | MACR = $78_H$ | write CM-data addressed by MAAR with content of MADR; write CM-code addressed by MAAR with '1000'$_B$ (even address code for decentral monitor and C/I-channels) |
| Read | STAR | Wait for STAR:MAC = 0 |
| Write | MAAR = $89_H$ | from upstream CFI-port 0, time slot 3 |
| Write | MACR = $70_H$ | write CM-code addressed by MAAR with '0000'$_B$ (odd address code for decentral monitor and C/I-channels) |
| Read | STAR | Wait for STAR:MAC = 0 |

The subscriber's downstream $B_1$-channel is internally looped via PCM-port 1, time slot 1

| | | |
|---|---|---|
| Write | MADR = $85_H$ | internal loop from PCM-port 1, time slot 1 |
| Write | MAAR = $00_H$ | to downstream CFI-port 0, time slot 0 |
| Write | MACR = $71_H$ | write CM-data addressed by MAAR with content of MADR; write CM-code addressed by MAAR with '0001'$_B$ (code for a simple 64-kbit/s connection) |
| Read | STAR | Wait for STAR:MAC = 0 |

The subscriber's downstream $B_2$-channel is switched from PCM-port 0, time slot 1

| | | |
|---|---|---|
| Write | MADR = $01_H$ | connection from PCM-port 0, time slot 1 |
| Write | MAAR = $01_H$ | to downstream CFI-port 0, time slot 1 |
| Write | MACR = $71_H$ | write CM-data addressed by MAAR with content of MADR; write CM-code addressed by MAAR with '0001'$_B$ (code for a simple 64-kbit/s connection) |
| Read | STAR | Wait for STAR:MAC = 0 |

The subscriber's downstream time slots 2 and 3 are initialized as monitor and C/I-channels with D-channel handling by the SACCO-A

| | | |
|---|---|---|
| Write | MADR = $FF_H$ | C/I-code to be transmitted = '1111'$_B$ |
| | (MADR = $F3_H$ | D-channel blocking code '1100'$_B$ to be transmitted.) |
| Write | MAAR = $08_H$ | to downstream CFI-port 0, time slot 2 |
| Write | MACR = $7A_H$ | write CM-data addressed by MAAR with content of MADR; write CM-code addressed by MAAR with '1010'$_B$ (even address code for monitor and C/I-channels with D-channel handling by SACCO-A) |
| Read | STAR | Wait for STAR:MAC = 0 |
| Write | MAAR = $09_H$ | from upstream CFI-port 0, time slot 3 |
| Write | MACR = $7B_H$ | write CM-code addressed by MAAR with '1011'$_B$ (odd address code for monitor and C/I-channels with D-channel handling by SACCO-A) |
| Read | STAR | Wait for STAR:MAC = 0 |

Set EPIC-1 to normal mode

| | | |
|---|---|---|
| Write | OMDR = $C0_H$ | set EPIC-1 to CM-normal mode; Interrupt line will go active |
| Read | ISTA = $20_H$ | EPIC-1 interrupt |
| Read | ISTA_E = $08_H$ | PFI-interrupt: PCM-synchronisity status has changed |
| Read | STAR_E = $25_H$ | ELIC is synchronized to PCM-interface; MFIFO ready |

Reset tristate field of Data Memory (DM)

| | | |
|---|---|---|
| Write | MADR = $00_H$ | all bits of time slot set to high impedance |
| Write | MACR = $68_H$ | write MADR to all locations of PCM-tristate field |
| Read | STAR | Wait for STAR:MAC = 0 |

### 3.8.6.2 SACCO-A Initialization Example

Configure the SACCO-A for communication with downstream subscribers

| | | |
|---|---|---|
| Write | MODE = $A8_H$ | set SACCO-B to transparent mode 1; switch receiver active |
| Write | RAH1 = $00_H$ | response SAPI1: Signaling data |
| Write | RAH2 = $40_H$ | response SAPI 2: Packet-switched data |
| (Write | CCR2 = $00_H$) | reset value: T×DA pin disabled; standard data sampling; RFS-interrupt disabled |
| Write | CCR1 = $87_H$ | power-up SACCO-A in point to point configuration and clock mode 3 with double rate clock; inter frame timefill = all ’1’s |

Reset the SACCO-A’s FIFOs

| | | |
|---|---|---|
| Write | CMDR = $C1_H$ | reset CPU accessible and CPU inaccessible part of RFIFO, and reset XFIFO; the interrupt line will go active |
| Read | ISTA = $02_H$ | interrupt of SACCO-A |
| Read | ISTA_A = $10_H$ | transmit pool ready |

### 3.8.6.3 D-Channel Arbiter Initialization Example

Enable D-channel transmission to CFI-port 0, channel 0

| | | |
|---|---|---|
| (Write | XDC = $00_H$) | reset value: broadcasting disabled; transmit to channel 0 of port 0 |

Enable D-channel reception on CFI-port 0, channel 0

| | | |
|---|---|---|
| Write | AMO = $F9_H$ | start with maximum selection delay; suspend counter active; control of D-channel to take place via C/I-bit; control channel master enabled |
| Write | DCE0 = $01_H$ | enable CFI-port 0, channel 0 for data reception |

### 3.8.6.4 PCM- and CFI-Interface Activation Example

Write OMDR = $EE_H$ see **chapter 3.8.5**.

Enable upstream PCM-port 0, time slot 5

| | | |
|---|---|---|
| Write | MADR = $0F_H$ | set all bits of time slot to low impedance |
| Write | MAAR = $89_H$ | PCM-port 0, time slot 5 |
| Write | MACR = $60_H$ | write only single tristate control position |
| Read | STAR | Wait for STAR:MAC = 0 |

### 3.8.6.5 SACCO-B Initialization Example

Configure the SACCO-B as secondary station for an upstream (non-PBC) group controller

| | | |
|---|---|---|
| Write | MODE = $48_H$ | set SACCO-B to 8-bit non-auto mode; switch receiver active |
| Write | RAH1 = $00_H$ | the high-byte comparison registers should be set to $00_H$ when using non-auto mode |
| Write | RAH2 = $00_H$ | |
| Write | RAL1 = $89_H$ | 8-bit address of SACCO-B |
| Write | RAL2 = $FF_H$ | 8-bit group address (broadcast by group controller) |
| Write | CCR2 = $08_H$ | TxDB pin enabled; standard data sampling; RFS-interrupt disabled |
| Write | CCR1 = $98_H$ | power-up SACCO-B in point-to-point configuration and clock mode 0 with single rate clock;<br>inter frame timefill = flags;<br>TxDB is push-pull output |

Reset the SACCO-B's FIFOs

| | | |
|---|---|---|
| Write | CMDR = $C1_H$ | reset CPU accessible and CPU inaccessible part of RFIFO, and reset XFIFO; the interrupt line will go active |
| Read | ISTA = $08_H$ | interrupt of SACCO-B |
| Read | ISTA_B = $10_H$ | transmit pool ready |

# 4 Detailed Register Description

## 4.1 Register Address Arrangement

### Interrupt Top Level

| Group | Reg Name | Chip Select | Access | Address mux | Address demux | Reset Value | Comment | Refer to page |
|---|---|---|---|---|---|---|---|---|
| Interrupt | ISTA | $\overline{CSE}$ | RD | $82_H$ | $41_H$ | $00_H$ | interrupt status reg. | 124 |
| top level | MASK | $\overline{CSE}$ | WR | $82_H$ | $41_H$ | $00_H$ | mask reg. | 125 |

### Parallel Ports

| Group | Reg Name | Chip Select | Access | Address mux | Address demux | Reset Value | Comment | Refer to page |
|---|---|---|---|---|---|---|---|---|
| PORT0 | PORT0 | $\overline{CSE}$ | RD | $84_H$ | $42_H$ | $xx_H$ | port 0 data | 126 |
| PORT1 | PORT1 | $\overline{CSE}$ | RD/WR | $86_H$ | $43_H$ | $Fx_H$ | port 1 data | 126 |
| | PCON1 | $\overline{CSE}$ | WR | $88_H$ | $44_H$ | $F0_H$ | port 1 configuration reg. | 127 |

### Watchdog Timer

| Group | Reg Name | Chip Select | Access | Address mux | Address demux | Reset Value | Comment | Refer to page |
|---|---|---|---|---|---|---|---|---|
| Watchdog timer | WTC | $\overline{CSE}$ | RD/WR | $80_H$ | $40_H$ | $1F_H$ | watchdog timer control reg. | 127 |

### ELIC® Mode Register

| Group | Reg Name | Chip Select | Access | Address mux | Address demux | Reset Value | Comment | Refer to page |
|---|---|---|---|---|---|---|---|---|
| ELIC Mode | EMOD | $\overline{CSE}$ | RD/WR | $7E_H$ | $3F_H$ | $XF_H$ | ELIC mode version number | 128 |

## EPIC®-1

| Group | Reg Name | Chip Select | Access | Address mux | Address demux | Reset Value | Comment | Refer to page |
|---|---|---|---|---|---|---|---|---|
| EPIC-1-PCM interface | PMOD | $\overline{CSE}$ | RD/WR | $20_H$ | $10_H$ | $00_H$ | PCM-mode reg. | 130 |
| | PBNR | $\overline{CSE}$ | RD/WR | $22_H$ | $11_H$ | $FF_H$ | PCM-bit number reg. | 132 |
| | POFD | $\overline{CSE}$ | RD/WR | $24_H$ | $12_H$ | $00_H$ | PCM-offset downstream reg. | 132 |
| | POFU | $\overline{CSE}$ | RD/WR | $26_H$ | $13_H$ | $00_H$ | PCM-offset upstream reg. | 133 |
| | PCSR | $\overline{CSE}$ | RD/WR | $28_H$ | $14_H$ | $00_H$ | PCM-clock shift reg. | 134 |
| | PICM | $\overline{CSE}$ | RD | $2A_H$ | $15_H$ | $xx_H$ | PCM-input comparison mismatch reg. | 135 |
| EPIC-1 CFI | CMD1 | $\overline{CSE}$ | RD/WR | $2C_H$ | $16_H$ | $00_H$ | CFI-mode reg. 1 | 136 |
| | CMD2 | $\overline{CSE}$ | RD/WR | $2E_H$ | $17_H$ | $00_H$ | CFI-mode reg. 2 | 138 |
| | CBNR | $\overline{CSE}$ | RD/WR | $30_H$ | $18_H$ | $FF_H$ | CFI-bit number reg. | 141 |
| | CTAR | $\overline{CSE}$ | RD/WR | $32_H$ | $19_H$ | $00_H$ | CFI time slot adjustment reg. | 141 |
| | CBSR | $\overline{CSE}$ | RD/WR | $34_H$ | $1A_H$ | $00_H$ | CFI-bit shift reg. | 142 |
| | CSCR | $\overline{CSE}$ | RD/WR | $36_H$ | $1B_H$ | $00_H$ | CFI-subchannel reg. | 143 |
| EPIC-1 memory access | MACR | $\overline{CSE}$ | RD/WR | $00_H$ | $00_H$ | $xx_H$ | memory access control reg. | 144 |
| | MAAR | $\overline{CSE}$ | RD/WR | $02_H$ | $01_H$ | $xx_H$ | memory access address reg. | 147 |
| | MADR | $\overline{CSE}$ | RD/WR | $04_H$ | $02_H$ | $xx_H$ | memory access data reg. | 148 |

**EPIC®-1** (cont'd)

| Group | Reg Name | Chip Select | Access | Address mux | Address demux | Reset Value | Comment | Refer to page |
|---|---|---|---|---|---|---|---|---|
| EPIC-1 synchro-nous transfer | STDA | $\overline{CSE}$ | RD/WR | $06_H$ | $03_H$ | $xx_H$ | synchron transfer data reg. A | 148 |
| | STDB | $\overline{CSE}$ | RD/WR | $08_H$ | $04_H$ | $xx_H$ | synchron transfer data reg. B | 149 |
| | SARA | $\overline{CSE}$ | RD/WR | $0A_H$ | $05_H$ | $xx_H$ | synchron transfer receive address reg. A | 149 |
| | SARB | $\overline{CSE}$ | RD/WR | $0C_H$ | $06_H$ | $xx_H$ | synchron transfer receive address reg. B | 150 |
| | SAXA | $\overline{CSE}$ | RD/WR | $0E_H$ | $07_H$ | $xx_H$ | synchron transfer transmit address reg. A | 150 |
| | SAXB | $\overline{CSE}$ | RD/WR | $10_H$ | $08_H$ | $xx_H$ | synchron transfer transmit address reg. B | 151 |
| | STCR | $\overline{CSE}$ | RD/WR | $12_H$ | $09_H$ | 00xxxx xx | synchron transfer control reg. | 151 |
| EPIC-1 monitor/ feature control | MFAIR | $\overline{CSE}$ | RD | $14_H$ | $0A_H$ | 0xxxxx xx | MF-channel active indication reg. | 152 |
| | MFSAR | $\overline{CSE}$ | WR | $14_H$ | $0A_H$ | $00_H$ | MF-channel subscriber address reg. | 153 |
| | MFFIFO | $\overline{CSE}$ | RD/WR | $16_H$ | $0B_H$ | $xx_H$ | MF-channel FIFO | 153 |
| EPIC-1 status/ control | CIFIFO | $\overline{CSE}$ | RD | $18_H$ | $0C_H$ | $00_H$ | signaling channel FIFO | 154 |
| | TIMR | $\overline{CSE}$ | WR | $18_H$ | $0C_H$ | $00_H$ | timer reg. | 154 |
| | STAR_E | $\overline{CSE}$ | RD | $1A_H$ | $0D_H$ | $05_H$ | status register EPIC | 155 |
| | CMDR_E | $\overline{CSE}$ | WR | $1A_H$ | $0D_H$ | $00_H$ | command reg. EPIC | 156 |
| | ISTA_E | $\overline{CSE}$ | RD | $1C_H$ | $0E_H$ | $00_H$ | interrupt status EPIC-1 | 158 |
| | MASK_E | $\overline{CSE}$ | WR | $1C_H$ | $0E_H$ | $00_H$ | mask register EPIC-1 | 159 |
| | OMDR | $\overline{CSE}$ | RD/WR | $1E_H$ $3E_H$ | $0F_H$ | $00_H$ | operation mode reg. | 160 |
| | VNSR | $\overline{CSE}$ | RD/WR | $3A_H$ | $1D_H$ | $01_H$ | version number status register | 162 |

## SACCO

| Group | Reg Name | Chip Select | Access | Address mux | Address demux | Reset Value | Comment | Refer to page |
|---|---|---|---|---|---|---|---|---|
| SACCO-FIFO | RFIFO | $\overline{CSS}$ | RD | $00_H$ $80_H$ : : $3E_H$ $BE_H$ | $00_H$ $40_H$ : : $1F_H$ $5F_H$ | $xx_H$ : $xx_H$ | receive FIFO | 163 |
| | XFIFO | $\overline{CSS}$ | WR | $00_H$ $80_H$ : : $3E_H$ $BE_H$ | $00_H$ $40_H$ : : $1F_H$ $5F_H$ | $xx_H$ : $xx_H$ | transmit FIFO | 164 |
| SACCO-status/ control | ISTA_A/B | $\overline{CSS}$ | RD | $40_H$ $C0_H$ | $20_H$ $60_H$ | $00_H$ | interrupt status reg. channel A/B | 165 |
| | MASK_A/B | $\overline{CSS}$ | WR | $40_H$ $C0_H$ | $20_H$ $60_H$ | $00_H$ | mask reg. channel A/B | 166 |
| | EXIR_A/B | $\overline{CSS}$ | RD | $48_H$ $C8_H$ | $24_H$ $64_H$ | $00_H$ | extended interrupt channel A/B | 166 |
| | CMDR | $\overline{CSS}$ | WR | $42_H$ $C2_H$ | $21_H$ $61_H$ | $00_H$ | command reg. | 168 |
| | MODE | $\overline{CSS}$ | RD/WR | $44_H$ $C4_H$ | $22_H$ $62_H$ | $00_H$ | mode reg. | 170 |
| | CCR1 | $\overline{CSS}$ | RD/WR | $5E_H$ $DE_H$ | $2F_H$ $6F_H$ | $00_H$ | channel configuration reg. 1 | 171 |
| | CCR2 | $\overline{CSS}$ | RD/WR | $58_H$ $D8_H$ | $2C_H$ $6C_H$ | $00_H$ | channel configuration reg. 2 | 173 |
| | RLCR | $\overline{CSS}$ | WR | $5C_H$ $DC_H$ | $2E_H$ $6E_H$ | $xx_H$ | receive frame length check | 174 |
| | STAR | $\overline{CSS}$ | RD | $42_H$ $C2_H$ | $21_H$ $61_H$ | $48_H$ | status reg. | 175 |
| | RSTA | $\overline{CSS}$ | RD | $4E_H$ $CE_H$ | $27_H$ $67_H$ | $xx_H$ | receive status reg. | 176 |
| | RHCR | $\overline{CSS}$ | WR | $52_H$ $D2_H$ | $29_H$ $69_H$ | $xx_H$ | receive HDLC-control byte | 178 |
| SACCO-transmit addr. | XAD1 | $\overline{CSS}$ | WR | $48_H$ $C8_H$ | $24_H$ $64_H$ | $xx_H$ | transmit address 1 | 178 |
| | XAD2 | $\overline{CSS}$ | WR | $4A_H$ $CA_H$ | $25_H$ $65_H$ | $xx_H$ | transmit address 2 | 179 |
| SACCO-address recognition | RAL1 | $\overline{CSS}$ | RD/WR | $50_H$ $D0_H$ | $28_H$ $68_H$ | $xx_H$ | receive address low 1 | 179 |
| | RAL2 | $\overline{CSS}$ | WR | $52_H$ $D2_H$ | $29_H$ $69_H$ | $xx_H$ | receive address low 2 | 180 |
| | RAH1 | $\overline{CSS}$ | WR | $4C_H$ $CC_H$ | $26_H$ $66_H$ | $xx_H$ | receive address high 1 | 180 |
| | RAH2 | $\overline{CSS}$ | WR | $4E_H$ $CE_H$ | $27_H$ $67_H$ | $xx_H$ | receive address high 2 | 181 |

## SACCO (cont'd)

| Group | Reg Name | Chip Select | Access | Address mux | Address demux | Reset Value | Comment | Refer to page |
|---|---|---|---|---|---|---|---|---|
| SACCO-DMA-support | RBCL | $\overline{CSS}$ | RD | $4A_H$ $CA_H$ | $25_H$ $65_H$ | $00_H$ | receive byte count low | 181 |
| | RBCH | $\overline{CSS}$ | RD | $5A_H$ $DA_H$ | $2D_H$ $6D_H$ | 000xxx xx | receive byte count high | 182 |
| | XBCL | $\overline{CSS}$ | WR | $54_H$ $D4_H$ | $2A_H$ $6A_H$ | $xx_H$ | transmit byte count low | 182 |
| | XBCH | $\overline{CSS}$ | WR | $5A_H$ $DA_H$ | $2D_H$ $6D_H$ | 000xxx xx | transmit byte count high | 183 |
| SACCO-time slot assignment | TSAX | $\overline{CSS}$ | WR | $60_H$ $E0_H$ | $30_H$ $70_H$ | $xx_H$ | time slot assign-ment transmit | 183 |
| | TSAR | $\overline{CSS}$ | WR | $60_H$ $E0_H$ | $30_H$ $70_H$ | $xx_H$ | time slot assignment receiver | 184 |
| | XCCR | $\overline{CSS}$ | WR | $62_H$ $E2_H$ | $31_H$ $71_H$ | $00_H$ | transmit channel capacity | 184 |
| | RCCR | $\overline{CSS}$ | WR | $66_H$ $E6_H$ | $33_H$ $73_H$ | $00_H$ | receive channel capacity | 185 |
| SACCO version | VSTR | $\overline{CSS}$ | WR | $5C_H$ | $2E_H$ | $80_H$ | version status register | 185 |

## Arbiter

| Group | Reg Name | Chip Select | Access | Address mux | Address demux | Reset Value | Comment | Refer to page |
|---|---|---|---|---|---|---|---|---|
| Arbiter control | AMO | $\overline{CSE}$ | RD/WR | $C0_H$ | $60_H$ | $00_H$ | arbiter mode register | 186 |
| | ASTATE | $\overline{CSE}$ | RD | $C2_H$ | $61_H$ | $xx_H$ | arbiter state register | 187 |
| | SCV | $\overline{CSE}$ | RD/WR | $C4_H$ | $62_H$ | $00_H$ | suspend counter value register | 187 |
| D-Channel enabling upstream | DCE0 | $\overline{CSE}$ | RD/WR | $C6_H$ | $63_H$ | $00_H$ | D-channel enable reg. 0 | 188 |
| | DCE1 | $\overline{CSE}$ | RD/WR | $C84_H$ | $64_H$ | $00_H$ | D-channel enable reg. 1 | 188 |
| | DCE2 | $\overline{CSE}$ | RD/WR | $CA_H$ | $65_H$ | $00_H$ | D-channel enable reg. 2 | 188 |
| | DCE3 | $\overline{CSE}$ | RD/WR | $CC_H$ | $66_H$ | $00_H$ | D-channel enable reg. 3 | 189 |
| D-Channel selecting downstream | XDC | $\overline{CSE}$ | RD/WR | $CE_H$ | $66_H$ | $00_H$ | transmit D-channel address register | 189 |
| | BCG0 | $\overline{CSE}$ | RD/WR | $D0_H$ | $68_H$ | $00_H$ | broadcast group reg. 0 | 190 |
| | BCG1 | $\overline{CSE}$ | RD/WR | $D2_H$ | $69_H$ | $00_H$ | broadcast group reg. 1 | 190 |
| | BCG2 | $\overline{CSE}$ | RD/WR | $D4_H$ | $6A_H$ | $00_H$ | broadcast group reg. 2 | 190 |
| | BCG3 | $\overline{CSE}$ | RD/WR | $D6_H$ | $6B_H$ | $00_H$ | broadcast group reg. 3 | 190 |

## 4.2 Interrupt Top Level

### 4.2.1 Interrupt Status Register (ISTA)

Access in demultiplexed µP-interface mode: read address: $41_H$
Access in multiplexed µP-interface mode: read address: $82_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| IWD | IDA | IEP | EXB | ICB | EXA | ICA | 0 |

**IWD** Interrupt Watchdog Timer.

The watchdog timer is expired and an external reset (RESIN) was generated. The software failed to program the bits WTC1 and WTC2 in the correct sequence.

**IDA** Interrupt D-channel Arbiter.

The suspend counter expired while the arbiter was in the state "expect frame". The affected D-channel can be determined by reading register ASTATE.

**IEP** Interrupt EPIC-1,

detailed information is indicated in register ISTA_E.

**EXB** Extended interrupt SACCO-B,

detailed information is indicated in register EXIR_B.

**ICB** Interrupt SACCO-B,

detailed information is indicated in register ISTA_B.

**EXA** Extended interrupt SACCO-A,

detailed information is indicated in register EXIR_A.

**ICA** Interrupt SACCO-A,

detailed information is indicated in register ISTA_A.

IWD and IDA are reset when reading ISTA. The other bits are reset when reading the corresponding local ISTA- or EXIR-register.

### 4.2.2 Mask Register (MASK)

Access in demultiplexed µP-interface mode: write address: $41_H$
Access in multiplexed µP-interface mode: write address: $82_H$
Reset value: $00_H$ (all interrupts enabled)

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | IDA | IEP | EXB | ICB | EXA | ICA | 0 |

**IDA** enables(0)/disables(1) the D-Channel Arbiter interrupt

**IEP** enables(0)/disables(1) the EPIC-1 Interrupts

**EXB** enables(0)/disables(1) the SACCO-B Extended interrupts

**ICB** enables(0)/disables(1) the SACCO-B Interrupts

**EXA** enables(0)/disables(1) the SACCO-A Extended interrupts

**ICA** enables(0)/disables(1) the SACCO-A Interrupts

Each interrupt source/group can be selectively masked by setting the respective bit in the MASK-register (bit position corresponding to the ISTA-register). A masked IDA-interrupt is not indicated when reading ISTA. Instead it remains internally stored and will be indicated after the respective MASK-bit is reset. The watchdog timer interrupts is not maskable.

Even with a set MASK-bit EPIC-1 and SACCO-interrupts are indicated but no interrupt signal is generated.

When writing the MASK-register while an interrupt is indicated, $\overline{INT}$ is temporarily set into the inactive state.

## 4.3 Parallel Ports

### 4.3.1 PORT0 Data Register (PORT0)

Demultiplexed address mode: read address: $42_H$
Access in multiplexed µP-interface mode: read address: $84_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| P0D7 | P0D6 | P0D5 | P0D4 | P0D3 | P0D2 | P0D1 | P0D0 |

**P0D7..0** PORT0 data 7…0.
Data sampled on the related pin with the falling RD-edge.

*Note: Port 0 is only available when the multiplexed Siemens/Intel type bus mode is used (ALE is switching).*

### 4.3.2 PORT1 Data Register (PORT1)

Access in demultiplexed µP-interface mode: read/write address: $43_H$
Access in multiplexed µP-interface mode: read/write address: $86_H$
Reset value: $Fx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | P1D3 | P1D2 | P1D1 | P1D0 |

**P1D3..0** PORT1 data 3…0.
Write operation:
Data is output on the related pin, assuming the pin is configured in PCON1 as an output. The data is activated with the falling $\overline{WR}$ edge. It is stable until another write access to PORT1 is executed or PCON1 is reprogrammed. All outputs have push-pull characteristic.

Read operation:
Data is sampled on the related pin with the falling $\overline{RD}$-edge, assuming the pin is configured in PCON1 as an input.

*Note: In order to avoid an undefined behavior of pins P1(3:0) when reprogramming PCON1-values from input to output, it is recommended to use external pull-up/ pull-down devices at pins P1(3:0).*

### 4.3.3 Port1 Configuration Register (PCON1)

Access in demultiplexed µP-interface mode: read/write address: $44_H$
Access in multiplexed µP-interface mode: read address: $88_H$
Reset value: $F0_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | P1C3 | P1C2 | P1C1 | P1C0 |

**P1C3..0** PORT1 Configuration 3…0.
0…port1, pin # is configured as input.
1…port1, pin # is configured as output with push-pull characteristic.

## 4.4 Watchdog Timer

### 4.4.1 Watchdog Control Register (WTC)

Access in demultiplexed µP-interface mode: read/write address: $40_H$
Access in multiplexed µP-interface mode: read address: $80_H$
Reset value: $1F_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| WTC1 | WTC2 | SWT | 1 | 1 | 1 | 1 | 1 |

**SWT** Start Watchdog Timer.
When set, the watchdog timer is started. The only way to disable it, is a ELIC-reset (power-up or RESEX).

**WTC1..2** Watchdog Timer Control.
Once the watchdog timer has been started WTC1, WTC2 have to be written once every 1024 PFS-cycles in the following sequence in order to prevent the watchdog expiring.

| | WTC1 | WTC2 |
|---|---|---|
| 1) | 1 | 0 |
| 2) | 0 | 1 |

The minimum required interval between the two write accesses is 2 PDC-periods.

### 4.5 ELIC® Mode Register / Version Number Register (EMOD)

Access in demultiplexed µP-interface mode: read/write address: $3F_H$
Access in multiplexed µP-interface mode: read/write address: $7E_H$
Reset value: $XF_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| VN3 | VN2 | VN1 | VN0 | 1 | 1 | ECMD2 | DMXAD |

**VN(3:0)** ELIC-Version Number according to the following table:

| VN (3:0) | ELIC-Version |
|---|---|
| 1111 | V 1.1 |
| 1110 | V 1.2 |
| 1101 | V 1.3 |

**ECMD2** ELIC CFI-Mode Bit 2.
If set to ’0’, the CFI-mode 0 with a 2.048-MBit/s data rate can be used with a 2.048-MHz PDC-input clock.

This mode requires further restrictions of the current ELIC-specification:

1) EPIC-1 PMOD:PCR must be set to ’1’.

   *Note: Although the PCM clock PDC is set to double clock rate by this bit, the data rate must always be equal to the clock rate.*

2) EPIC-1 CMD2:COC must be programmed to ’0’, i.e. it is not possible to output a DCL-clock with a frequency of twice the CFI-data rate.

3) EPIC-1 CMD1:CSS must be programmed to ’0’, i.e. it is not possible to select DCL as clock and FSC as framing signal source for the configurable interface.

4) The timing of the PCM-interface is expanded:

| Parameter | Symbol | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|
| | | min. | max. | | |
| Clock period | $T_{CP}$ | 480 | – | ns | EMOD:ECMD2 = '0' |
| Clock period low | $T_{CPL}$ | 200 | – | ns | |
| Clock period high | $T_{CPH}$ | 200 | – | ns | |

5) PCSR:DRE has to be set to ‘1’.
PCSR:URE has to be set to ‘1’.
When provided with a 2 MHz PDC, the ELIC internally generates a 4 MHz clock.
Since the clock shift capabilities (provided by register bits PCSR:DRCS and PCSR:ADSR0) apply to the internal 4 MHz clock, the frame can thus be shifted with a resolution of a half bit.

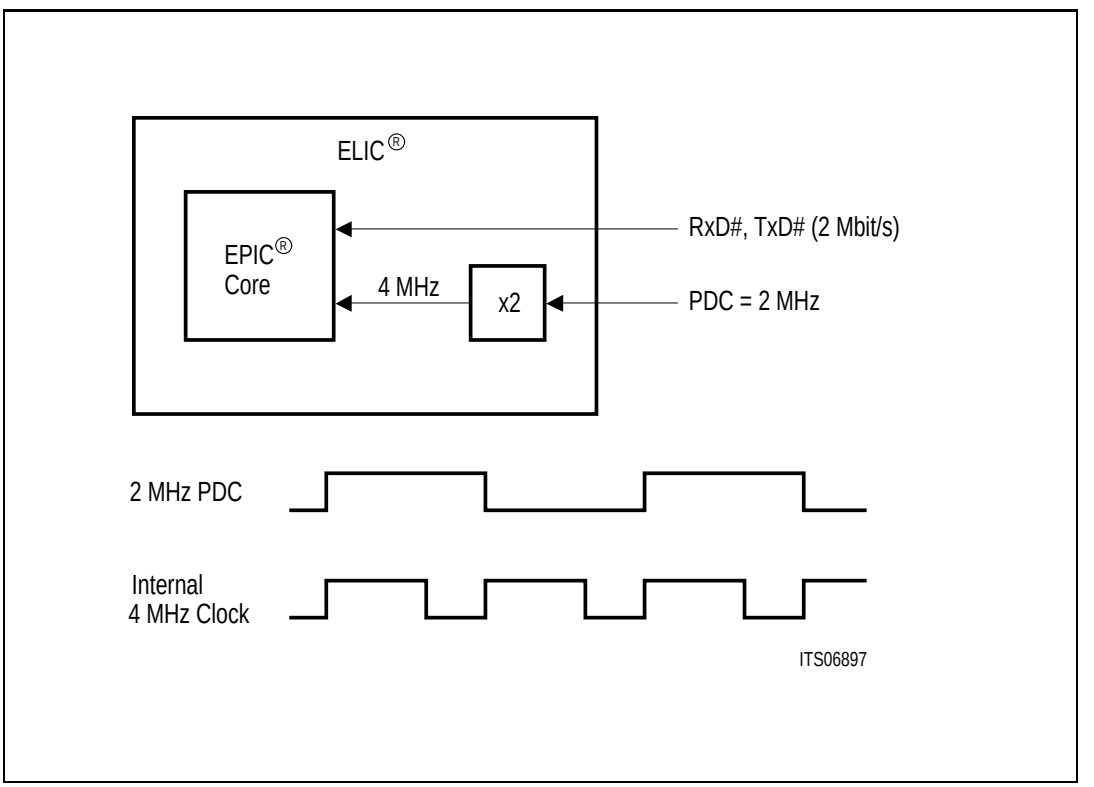



**Figure 55**
**Timing Relation Between Internal and External Clock**

6) PMOD:PSM has to be set to ’1’.
The frame signal PFS must always be sampled with the rising edge of PDC. The set-up and hold times of PFS are still valid respected to external PDC.

**DMXAD** Demultiplexed Address.
If set to '0' the **de**multiplexed addresses are also valid in the multiplexed µP-interface mode.

## 4.6 EPIC®-1

### 4.6.1 PCM-Mode Register (PMOD)

Access in demultiplexed µP-interface mode: read/write address: $10_H$
Access in multiplexed µP-interface mode: read/write address: $20_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| PMD1 | PMD0 | PCR | PSM | AIS1 | AIS0 | AIC1 | AIC0 |

*Note: If EMOD:ECMD2 is set to '0' some restrictions apply to the setting of register PMOD (see **chapter 4.5**).*

**PMD1..0** PCM-Mode. Defines the actual number of PCM-ports, the data rate range and the data rate stepping.

| PMD1..0 | PCM-Mode | Port Count | Data Rate [kbBt/s] | | Data Rate Stepping [kBit/s] |
|---|---|---|---|---|---|
| | | | min. | max. | |
| 00 | 0 | 4 | 256 | 2048 | 256 |
| 01 | 1 | 2 | 512 | 4096 | 512 |
| 10 | 2 | 1 | 1024 | 8192 | 1024 |
| 11 | 3 | 2 | 512 | 4096 | 512 |

The actual selection of physical pins is described below (AIS1/0).

**PCR** PCM-Clock Rate.
0… single clock rate, data rate is identical with the clock frequency supplied on pin PDC.
1… double clock rate, data rate is half the clock frequency supplied on pin PDC.

*Note: Only single clock rate is allowed in PCM-mode 2!*

**PSM** PCM Synchronization Mode.
A rising edge on PFS synchronizes the PCM-frame. PFS is not evaluated directly but is sampled with PDC.
0… the external PFS is evaluated with the falling edge of PDC. The internal PFS (internal frame start) occurs with the next rising edge of PDC.
1… the external PFS is evaluated with the rising edge of PDC. The internal PFS (internal frame start) occurs with this rising edge of PDC.

**AIS1..0** Alternative Input Selection.
These bits determine the relationship between the physical pins and the logical port numbers. The logical port numbers are used when programming the switching functions.

*Note: In PCM-mode 0 these bits may not be set!*

| PCM | Port 0 | | | Port 1 | | | Port 2 | | | Port 3 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Mode** | **RxD0** | **TxD0** | **$\overline{\text{TSC0}}$** | **RxD1** | **TxD1** | **$\overline{\text{TSC1}}$** | **RxD2** | **TxD2** | **$\overline{\text{TSC2}}$** | **RxD3** | **TxD3** | **$\overline{\text{TSC3}}$** |
| 0 | IN0 | OUT0 | val0 | IN1 | OUT1 | val1 | IN2 | OUT2 | val2 | IN3 | OUT3 | val3 |
| 1 | IN0 (AIS0=1) | OUT0 | val0 | IN0 (AIS0=0) | tristate | AIS0 | IN1 (AIS1=1) | OUT1 | val1 | IN1 (AIS1=0) | tristate | AIS1 |
| 2 | not active | OUT | val | not active | tristate | AIS0 | IN (AIS1=1) | undef. | undef. | IN (AIS1=0) | tristate | AIS1 |
| 3 | IN0 (AIS0=1) | OUT0 | val0 | IN0 (AIS0=0) | $\overline{\text{OUT0}}$ | val0 | In1 (AIS1=1) | OUT1 | val1 | IN1 (AIS1=0) | $\overline{\text{OUT1}}$ | val1 |

**AIC1** Alternate Input Comparison 1.
0…input comparison of port 2 and 3 is disabled
1…the inputs of port 2 and 3 are compared

**AIC0** Alternate Input Comparison 0.
0…input comparison of port 0 and 1 is disabled
1…the inputs of port 0 and 1 are compared

*Note: The comparison function is operational in all PCM-modes; however, a redundant PCM-line which can be switched over to by means of the PMOD: AIS-bits is only available in PCM-modes 1, 2 and 3.*

### 4.6.2 Bit Number per PCM-Frame (PBNR)

Access in demultiplexed µP-interface mode: read/write address: $11_H$
Access in multiplexed µP-interface mode: read/write address: $22_H$
Reset value: $FF_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| BNF7 | BNF6 | BNF5 | BNF4 | BNF3 | BNF2 | BNF1 | BNF0 |

**BNF7..0** Bit Number per PCM Frame.
PCM-mode 0: BNF7..0 = number of bits – 1
PCM-mode 1: BNF7..0 = (number of bits – 2) / 2
PCM-mode 2: BNF7..0 = (number of bits – 4) / 4
PCM-mode 3: BNF7..0 = (number of bits – 2) / 2

The value programmed in PBNR is also used to check the PFS-period.

### 4.6.3 PCM-Offset Downstream Register (POFD)

Access in demultiplexed µP-interface mode: read/write address: $12_H$
Access in multiplexed µP-interface mode: read/write address: $24_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| OFD9 | OFD8 | OFD7 | OFD6 | OFD5 | OFD4 | OFD3 | OFD2 |

**OFD9..2** Offset Downstream bit 9…2.
These bits together with PCSR:OFD1**..**0 determine the offset of the PCM-frame in downstream direction. The following formulas apply for calculating the required register value. BND is the bit number in downstream direction marked by the rising internal PFS-edge. BPF denotes the actual number of bits constituting a frame.

PCM-mode 0: OFD9..2 = $mod_{BPF}$ (BND – 17 + BPF)
PCSR:OFD1..0 = 0
PCM-mode 1,3: PFD9..1 = $mod_{BPF}$ (BND – 33 + BPF)
PCSR: PFD0 = 0
PCM-mode 2: OFD9..0 = $mod_{BPF}$ (BND – 65 + BPF)

### 4.6.4 PCM-Offset Upstream Register (POFU)

Access in demultiplexed µP-interface mode: read/write address: $13_H$
Access in multiplexed µP-interface mode: read/write address: $26_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| OFU9 | OFU8 | OFU7 | OFU6 | OFU5 | OFU4 | OFU3 | OFU2 |

**OFU9..2** Offset Upstream bit 9…2.
These bits together with PCSR:OFU1**..**0 determine the offset of the PCM-frame in upstream direction. The following formulas apply for calculating the required register value. BNU is the bit number in upstream direction marked by the rising internal PFS-edge.

PCM-mode 0: OFU9..2 = $mode_{BPF}$ (BNU + 23)
PCSR:OFU1..00 = 0

PCM-mode 1,3: OFU9..1 = $mod_{BPF}$ (BNU + 47)
PCSR:OFU0 = 0

PCM-mode 2: OFU9..0 = $mod_{BPF}$ (BNU + 95)

### 4.6.5 PCM-Clock Shift Register (PCSR)

Access in demultiplexed µP-interface mode: read/write address: $14_H$
Access in multiplexed µP-interface mode: read/write address: $28_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| DRCS | OFD1 | OFD0 | DRE | ADSRO | OFU1 | OFU0 | URE |

**DRCS** Double Rate Clock Shift.
0…the PCM-input and output data are not delayed
1…the PCM-input and output data are delayed by one PDC-clock cycle

**OFD1..0** Offset Downstream bits 1…0, see POFD-register.

**DRE** Downstream Rising Edge.
0…the PCM-data is sampled with the falling edge of PDC
1…the PCM-data is sampled with the rising edge of PDC

**ADSRO** Add Shift Register on Output.
0…the PCM-output data are not delayed
1…the PCM-output data are delayed by one PDC-clock cycle

*Note: If both DRCS and ADSRO are set to logical 1, the PCM-output data are delayed by two PDC-clock cycles.*
*DRCS and ADSRO were added to the standard EPIC-1 PCSR register implemented in the PEB 2055 up to and including version A3.*

**OFU1..0** Offset Upstream bits 1…0, **see POFU-register**.

**URE** Upstream Rising Edge.
0…the PCM-data is transmitted with the falling edge of PDC
1…the PCM-data is transmitted with the rising edge of PDC

*Note: If EMOD:ECMD2 is set to '0' some restrictions apply to the setting of PCSR (see **chapter 4.5**).*

### 4.6.6 PCM-Input Comparison Mismatch (PICM)

Access in demultiplexed µP-interface mode: read/write address: $15_H$
Access in multiplexed µP-interface mode: read/write address: $2A_H$
Reset value: $xx_H$

bit 7 bit 0

| IPN | TSN6 | TSN5 | TSN4 | TSN3 | TSN2 | TSN1 | TSN0 |
|---|---|---|---|---|---|---|---|

**IPN** Input Pair Number.
This bit denotes the pair of ports, where a bit mismatch occurred.
0…mismatch between ports 0 and 1
1…mismatch between ports 2 and 3

**TSN6..0** Time slot Number.
When a bit mismatch occurred these bits identify the affected bit position.

| PCM-Mode | Time Slot Identification | Bit Identification |
|---|---|---|
| 0 | TSN6…TSN2 + 2 | TSN1,0: 00 bits 6,7<br>01 bits 4,5<br>10 bits 2,3<br>11 bits 0,1 |
| 1, 3 | TSN6…TSN1 + 4 | TSN0: 0 bits 4…7<br>1 bits 0…3 |
| 2 | TSN6…TSN0 + 8 | |

### 4.6.7 Configurable Interface Mode Register 1 (CMD1)

Access in demultiplexed µP-interface mode: read/write address: $16_H$
Access in multiplexed µP-interface mode: read/write address: $2C_H$
Reset value: $00_H$

bit 7 bit 0

| CSS | CSM | CSP1 | CSP0 | CMD1 | CMD0 | CIS1 | CIS0 |
|---|---|---|---|---|---|---|---|

**CSS** Clock Source Selection.
0…PDC and PFS are used as clock and framing source for the CFI. Clock and framing signals derived from these sources are output on DCL and FSC.
1…DCL and FSC are selected as clock and framing source for the CFI.
If EMOD:ECMD2 is set to ’0’, then CSS has to be set to ’0’ (see **chapter 4.5**).

**CSM** CFI-Synchronization Mode.
The rising FSC-edge synchronizes the CFI-frame.
0…FSC is evaluated with every falling edge of DCL.
1…FSC is evaluated with every rising edge of DCL.

*Note: If CSS = 0 is selected, CSM and PMOD:PSM must be programmed identical.*

**CSP1..0** Clock Source Prescaler 1,0.
The clock source frequency is divided according to the following table to obtain the CFI-reference clock CRCL.

| CSP1,0 | Prescaler Divisor |
|---|---|
| 00 | 2 |
| 01 | 1.5 |
| 10 | 1 |
| 11 | not allowed |

**CMD1..0** CFI-Mode1,0.
Defines the actual number and configuration of the CFI-ports.

| CMD1..0 | CFI-Mode | Number of Logical Ports | CFI-Data Rate [kBit/s] | | Min. Required CFI-Data Rate [kBit/s] Relative to PCM-Data Rate | Necessary Reference Clock (RCL) | DCL-Output Frequencies CMD1:CSS0 = 0 |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | | | |
| 00 | 0 | 4 DU (0..3) | 128 | 2048 | 32N/3 | 2xDR | DR, 2xDR[1)] |
| 01 | 1 | 2 DU (0..1) | 128 | 4096 | 64N/3 | DR | DR |
| 10 | 2 | 1 DU | 128 | 8192 | 64N/3 | 0.5xDR | DR |
| 11 | 3 | 8 bit (0..7) | 128 | 1024 | 16N/3 | 4xDR | DR, 2xDR |

[1)] In CFI-mode 0 data rate of 2.048 kBit/s can be used with a 2.048-kBit/s PDC-input clock, if EMOD:ECMD2 = '0'. Refer to **chapter 4.5 ELIC-Mode Register (EMOD)**.

where N = number of time slots in a PCM-frame

**CIS1..0** CFI Alternative Input Selection.
In CFI-mode 1 and 2 CIS1**..**0 controls the assignment between logical and physical receive pins. In CFI-mode 0 and 3 CIS1,0 should be set to 0.

| CFI-Mode | Port 0 | | Port 1 | | Port 2 | | Port 3 | |
|---|---|---|---|---|---|---|---|---|
| | DU0 | DD0 | DU1 | DD1 | DU2 | DD2 | DU3 | DD3 |
| 0 | IN0 | OUT0 | IN1 | OUT1 | IN2 | OUT2 | IN3 | OUT3 |
| 1 | IN0 CIS0 = 0 | OUT0 | IN1 CIS1 = 0 | OUT1 | IN0 CIS0 = 1 | tristate | IN1 CIS1 = 1 | tristate |
| 2 | IN CIS0 = 0 | OUT | not active | tristate | IN CIS0 = 1 | tristate | not active | tristate |
| 3 | I/O4 | I/O0 | I/O5 | I/O1 | I/O6 | I/O2 | I/O7 | I/O3 |

### 4.6.8 Configurable Interface Mode Register 2 (CMD2)

Access in demultiplexed µP-interface mode: read/write address: $17_H$
Access in multiplexed µP-interface mode: read/write address: $2E_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| FC2 | FC1 | FC0 | COC | CXF | CRR | CBN9 | CBN8 |

**FC2..0** Framing output Control.
Given that CMD1:CSS = 0, these bits determine the position of the FSC-pulse relative to the CFI-frame, as well as the type of FSC-pulse generated. The position and width of the FSC-signal with respect to the CFI-frame can be found in the following two **figures 56** and **57**.

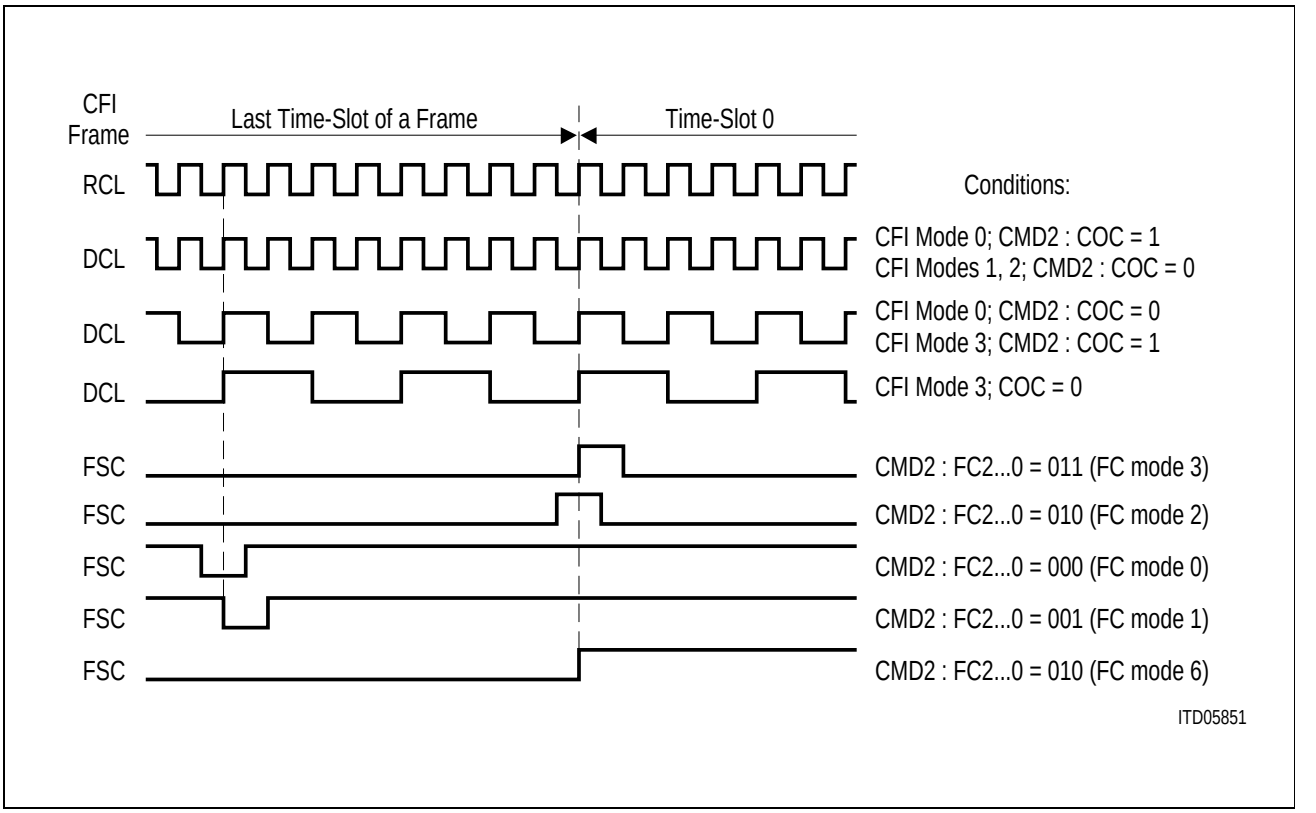



**Figure 56**
**Position of the FSC-Signal for FC-Modes 0, 1, 2, 3 and 6**

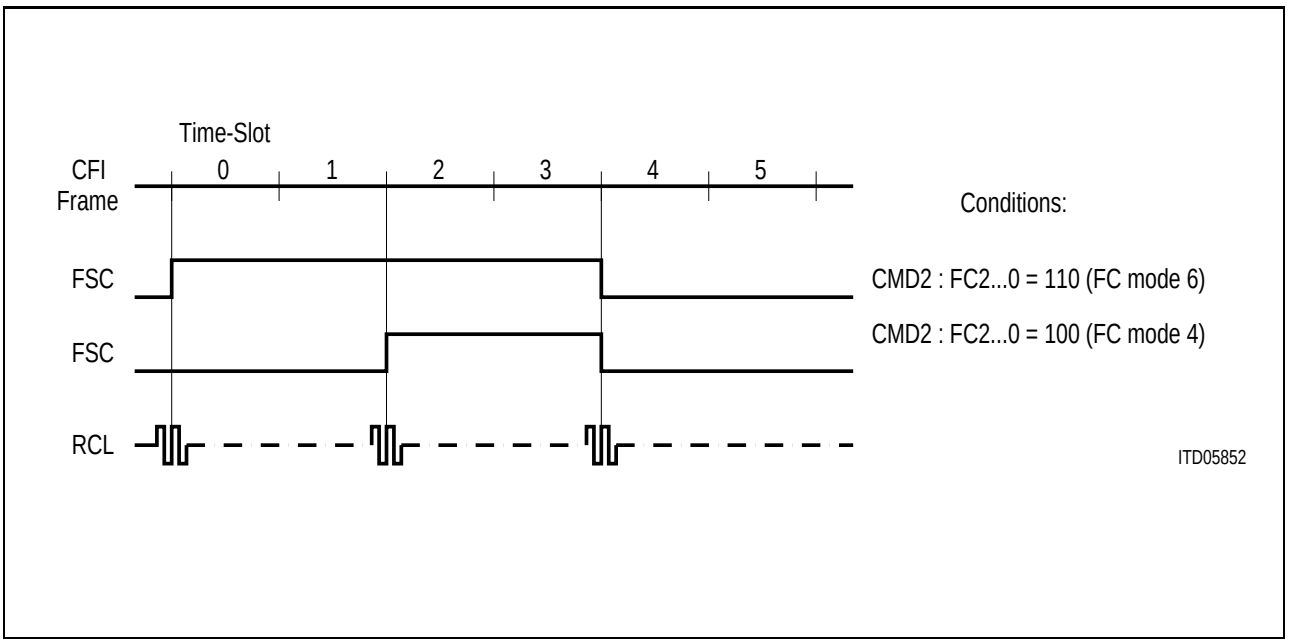



**Figure 57**
**Position of the FSC-Signal for FC-Modes 4 and 6**

*Note: The D-channel arbiter can only be operated with framing control modes 3, 6 and 7.*

Application examples:

| FC2 | FC1 | FC0 | FC-Mode | Main Applications |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | IOM-1 multiplexed (burst) mode |
| 0 | 0 | 1 | 1 | general purpose |
| 0 | 1 | 0 | 2 | general purpose |
| 0 | 1 | 1 | 3 | general purpose |
| 1 | 0 | 0 | 4 | 2 ISAC-S per SLD-port |
| 1 | 0 | 1 | 5 | reserved |
| 1 | 1 | 0 | 6 | IOM-2 or SLD-modes |
| 1 | 1 | 1 | 7 | software timed multiplexed applications |

For further details on the framing output control please refer to **chapter 5.2.2.3**.

**COC** CFI-Output Clock rate.
0…the frequency of DCL is identical to the CFI-data rate (all CFI-modes),
1…the frequency of DCL is twice the CFI-data rate (CFI-modes 0 and 3 only!)

*Note: Applies only if CMD1:CSS = 0.*
*If EMOD:ECMD2 is set to '0' then CMD2:COC must be set to '0' (see **chapter 4.5**).*

**CXF** CFI-Transmit on Falling edge.
0…the data is transmitted with the rising CRCL edge,
1…the data is transmitted with the falling CRCL edge.

**CRR** CFI-Receive on Rising edge.
0…the data is received with the falling CRCL edge,
1…the data is received with the rising CRCL edge.

*Note: CRR must be set to 0 in CFI-mode 3.*

**CBN9..8** CFI-Bit Number 9..8
these bits, together with the CBNR:CBN7..0, hold the number of bits per CFI-frame.

### 4.6.9 Configurable Interface Bit Number Register (CBNR)

Access in demultiplexed µP-interface mode: read/write address: $18_H$
Access in multiplexed µP-interface mode: read/write address: $30_H$
Reset value: $FF_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| CBN7 | CBN6 | CBN5 | CBN4 | CBN3 | CBN2 | CBN1 | CBN0 |

**CBN7..0** CFI-Bit Number 7..0.
The number of bits that constitute a CFI-frame must be programmed to CMD2, CBNR:CBN9..0 as indicated below.

CBN9..0 = number of bits – 1

For a 8-kHz frame structure, the number of bits per frame can be derived from the data rate by division with 8000.

### 4.6.10 Configurable Interface Time Slot Adjustment Register (CTAR)

Access in demultiplexed µP-interface mode: read/write address: $19_H$
Access in multiplexed µP-interface mode: read/write address: $32_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | TSN6 | TSN5 | TSN4 | TSN3 | TSN2 | TSN1 | TSN0 |

**TSN6..0** Time Slot Number.
The CFI-framing signal (PFS if CMD1:CSS = 0 or FSC if CMD1:CSS = 1) marks the CFI time slot called TSN according to the following formula:

TSN6**..**0 = TSN + 2

E.g.: If the framing signal is to mark time slot 0 (bit 7), CTAR must be set to $02_H$ (CBSR to $20_H$).

*Note: If CMD1:CSS = 0, the CFI-frame will be shifted - together with the FSC-output signal - with respect to PFS. The position of the CFI-frame relative to the FSC-output signal is not affected by these settings, but is instead determined by CMD2:FC2**..**0. If CMD1:CSS = 1, the CFI-frame will be shifted with respect to the FSC-input signal.*

### 4.6.11 Configurable Interface Bit Shift Register (CBSR)

Access in demultiplexed µP-interface mode: read/write address: $1A_H$
Access in multiplexed µP-interface mode: read/write address: $34_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | CDS2 | CDS1 | CDS0 | CUS3 | CUS2 | CUS1 | CUS0 |

**CDS2..0** CFI-Downstream bit Shift 2..0.
From the zero offset bit position (CBSR = $20_H$) the CFI-frame (downstream and upstream) can be shifted by up to 6 bits to the left (within the time slot number TSN programmed in CTAR) and by up to 2 bits to the right (within the previous time slot TSN – 1) by programming the CBSR:CDS2..0 bits:

| CBSR:CDS2..0 | Time Slot No. | Bit No. |
|---|---|---|
| 000 | TSN – 1 | 1 |
| 001 | TSN – 1 | 0 |
| 010 | TSN | 7 |
| 011 | TSN | 6 |
| 100 | TSN | 5 |
| 101 | TSN | 4 |
| 110 | TSN | 3 |
| 111 | TSN | 2 |

The bit shift programmed to CBSR:CDS2..0 affects both the upstream and downstream frame position in the same way.

**CUS3..2** CFI-Upstream bit Shift 3..0.
These bits shift the upstream CFI-frame relative to the downstream frame by up to 15 bits. For CUS3..0 = 0000, the upstream frame is aligned with the downstream frame (no bit shift).

### 4.6.12 Configurable Interface Subchannel Register (CSCR)

Access in demultiplexed µP-interface mode: read/write address: $1B_H$
Access in multiplexed µP-interface mode: read/write address: $36_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| SC31 | SC30 | SC21 | SC20 | SC11 | SC10 | SC01 | SC00 |

**SC#1..#0** CFI-Subchannel Control for logical port #.
The subchannel control bits SC#1..SC#0 specify separately for each logical port the bit positions to be exchanged with the data memory (DM) when a connection with a channel bandwidth as defined by the CM-code has been established:

| **SC#1** | **SC#0** | **Bit Positions for CFI Subchannels having a Bandwidth of** | | |
|---|---|---|---|---|
| | | 64 kBit/s | 32 kBit/s | 16 kBit/s |
| 0 | 0 | 7..0 | 7..4 | 7..6 |
| 0 | 1 | 7..0 | 3..0 | 5..4 |
| 1 | 0 | 7..0 | 7..4 | 3..2 |
| 1 | 1 | 7..0 | 3..0 | 1..0 |

*Note: In CFI-mode 1: SC21 = SC01; SC20 = SC00; SC31 = SC11; SC30 = SC10*
*In CFI-mode 2: SC31 = SC21 = SC11 = SC01; SC30 = SC20 = SC10 = SC00*
*In CFI-mode 3: SC0x-control ports 0 and 4; SC1x-control ports 1 and 5; SC2x-control ports 2 and 6; SC3x-control ports 3 and 7*

### 4.6.13 Memory Access Control Register (MACR)

Access in demultiplexed µP-interface mode: read/write address: $00_H$
Access in multiplexed µP-interface mode: read/write address: $00_H$
Reset value: $xx_H$

bit 7 bit 0

| RWS | MOC3 | MOC2 | MOC1 | MOC0<br>CMC3 | CMC2 | CMC1 | CMC0 |
|---|---|---|---|---|---|---|---|

With the MACR the µP selects the type of memory (CM or DM), the type of field (data or code) and the access mode (read or write) of the register access. When writing to the control memory code field, MACR also contain the 4 bit code (CMC3..0) defining the function of the addressed CFI time slot.

**RWS** Read/Write Select.
0…write operation on control or data memories
1…read operation on control or data memories

**MOC3..0** Memory Operation Code.

**CMC3..0** Control Memory Code.
These bits determine the type and destination of the memory operation as shown below.

*Note: Prior to a new access to any memory location (i.e. writing to MACR) the STAR:MAC bit must be polled for '0'.*

**1. Writing data to the upstream DM-data field (e.g. PCM-idle code). Reading data from the upstream or downstream DM-data field.**

MACR:

| RWS | MOC3 | MOC2 | MOC1 | MOC0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

**MOC3..0** defines the bandwidth and the position of the subchannel as shown below:

| MOC3..0 | Transferred Bits | Channel Bandwidth |
|---|---|---|
| 0000 | – | – |
| 0001 | bits 7..0 | 64 kBit/s |
| 0011 | bits 7..4 | 32 kBit/s |
| 0010 | bits 3..0 | 32 kBit/s |
| 0111 | bits 7..6 | 16 kBit/s |
| 0110 | bits 5..4 | 16 kBit/s |
| 0101 | bits 3..2 | 16 kBit/s |
| 0100 | bits 1..0 | 16 kBit/s |

*Note: When reading a DM-data field location, all 8 bits are read regardless of the bandwidth selected by the MOC-bits.*

**2. Writing to the upstream DM-code (tristate) field.**
**Control-reading the upstream DM-code (tristate).**

MACR:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| RWS | MOC3 | MOC2 | MOC1 | MOC0 | 0 | 0 | 0 |

MOC = 1100 Read/write tristate info from/to single PCM time slot

MOC = 1101 Write tristate info to all PCM time slots

*Note: The tristate field is exchanged with the 4 least significant bits (LSBs) of the MADR.*

**3. Writing data to the upstream or downstream CM-data field (e.g. signaling code).**
**Reading data from the upstream or downstream CM-data field.**

MACR:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| RWS | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

**4. Writing data to the upstream or downstream CM-data field and code field (e.g. switching a CFI to/from PCM-connection).**

MACR:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | CMC3 | CMC2 | CMC1 | CMC0 |

The 4-bit code field of the control memory (CM) defines the functionality of a CFI time slot and thus the meaning of the corresponding data field. This 4-bit code, written to the MACR:CMC3..0 bit positions, will be transferred to the CM-code field. The 8-bit MADR value is at the same time transferred to the CM-data field. There are codes for switching applications, pre-processed applications and for direct µP-access applications, as shown below:

**a) Switching Applications**

CMC = 0000 Unassigned channel (e.g. cancelling an assigned channel)
CMC = 0001 Bandwidth 64 kBit/s PCM time slot bits transferred: 7..0
CMC = 0010 Bandwidth 32 kBit/s PCM time slot bits transferred: 3..0
CMC = 0011 Bandwidth 32 kBit/s PCM time slot bits transferred: 7..4
CMC = 0100 Bandwidth 16 kBit/s PCM time slot bits transferred: 1..0
CMC = 0101 Bandwidth 16 kBit/s PCM time slot bits transferred: 3..2
CMC = 0110 Bandwidth 16 kBit/s PCM time slot bits transferred: 5..4
CMC = 0111 Bandwidth 16 kBit/s PCM time slot bits transferred: 7..6

*Note: The corresponding CFI time slot bits to be transferred are chosen in the CSCR-register.*

**b) Pre-processed Applications**

Downstream:

| Application | Even CM Address | Odd CM Address |
|---|---|---|
| Decentral D-channel handling | CMC = 1000 | CMC = 1011 |
| Central D-channel handling | CMC = 1010 | CMC = PCM-code for a 2-bit subtime slot |
| 6-bit Signaling (e.g. analog IOM) | CMC = 1010 | CMC = 1011 |
| 8-bit Signaling (e.g. SLD) | CMC = 1010 | CMC = 1011 |
| D-Channel handling by SACCO-A with ELIC-arbiter | CMC = 1010 | CMC = 1011 |

Upstream:

| Application | Even CM Address | Odd CM Address |
|---|---|---|
| Decentral D-channel handling | CMC = 1000 | CMC = 0000 |
| Central D-channel handling | CMC = 1000 | CMC = PCM-code for a 2-bit subtime slot |
| 6-bit Signaling (e.g. analog IOM) | CMC = 1010 | CMC = 1010 |
| 8-bit Signaling (e.g. SLD) | CMC = 1011 | CMC = 1011 |
| All code combinations are also valid for ELIC-arbiter operation. | | |

**c) µP-access Applications**

MACR:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

Setting CMC = 1001, initializes the corresponding CFI time slot to be accessed by the µP. Concurrently, the datum in MADR is written (as 8-bit CFI-idle code) to the CM-data field. The content of the CM-data field is directly exchanged with the corresponding time slot.

Note that once the CM-code field has been initialized, the CM-data field can be written and read as described in **chapter 3**.

**5. Control-reading the upstream or downstream CM-code.**

MACR:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

The CM-code can then be read out of the 4 LSBs of the MADR-register.

### 4.6.14 Memory Access Address Register (MAAR)

Access in demultiplexed µP-interface mode: read/write address: $01_H$
Access in multiplexed µP-interface mode: read/write address: $02_H$
Reset value: $xx_H$

bit 7 bit 0

| U/$\overline{D}$ | MA6 | MA5 | MA4 | MA3 | MA2 | MA1 | MA0 |
|---|---|---|---|---|---|---|---|

The Memory Access Address Register MAAR specifies the address of the memory access. This address encodes a CFI time slot for control memory (CM) and a PCM time slot for data memory (DM) accesses. Bit 7 of MAAR (U/$\overline{D}$-bit) selects between upstream and downstream memory blocks. Bits MA6..0 encode the CFI- or PCM-port and time slot number as in the following tables:

**Table 19**
**Time Slot Encoding for Data Memory Accesses**

| Data Memory Address | | |
|---|---|---|
| PCM-mode 0 | bit U/$\overline{D}$<br>bits MA6..MA3, MA0<br>bits MA2..MA1 | direction selection<br>time slot selection<br>logical PCM-port number |
| PCM-mode 1,3 | bit U/$\overline{D}$<br>bits MA6..MA3, MA1, MA0<br>bit MA2 | direction selection<br>time slot selection<br>logical PCM-port number |
| PCM-mode 2 | bit U/$\overline{D}$<br>bits MA6..MA0 | direction selection<br>time slot selection |

**Table 20**
**Time Slot Encoding for Control Memory Accesses**

| Control Memory Address | | |
|---|---|---|
| CFI-mode 0 | bit U/$\overline{D}$<br>bits MA6..MA3, MA0<br>bits MA2..MA1 | direction selection<br>time slot selection<br>logical CFI-port number |
| CFI-mode 1 | bit U/$\overline{D}$<br>bits MA6..MA3, MA2, MA0<br>bit MA1 | direction selection<br>time slot selection<br>logical CFI-port number |
| CFI-mode 2 | bit U/$\overline{D}$<br>bits MA6..MA0 | direction selection<br>time slot selection |
| CFI-mode 3 | bit U/$\overline{D}$<br>bits MA6..MA4, MA0<br>bits MA3..MA1 | direction selection<br>time slot selection<br>logical CFI-port number |

### 4.6.15 Memory Access Data Register (MADR)

Access in demultiplexed µP-interface mode: read/write address: $02_H$
Access in multiplexed µP-interface mode: read/write address: $04_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| MD7 | MD6 | MD5 | MD4 | MD3 | MD2 | MD1 | MD0 |

The Memory Access Data Register MADR contains the data to be transferred from or to a memory location. The meaning and the structure of this data depends on the kind of memory being accessed.

### 4.6.16 Synchronous Transfer Data Register (STDA)

Access in demultiplexed µP-interface mode: read/write address: $03_H$
Access in multiplexed µP-interface mode: read/write address: $06_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| MTDA7 | MTDA6 | MTDA5 | MTDA4 | MTDA3 | MTDA2 | MTDA1 | MTDA0 |

The STDA-register buffers the data transferred over the synchronous transfer channel A.

MTDA7 to MTDA0 hold the bits 7 to 0 of the respective time slot. MTDA7 (MSB) is the bit transmitted/received first, MTDA0 (LSB) the bit transmitted/received last over the serial interface.

### 4.6.17 Synchronous Transfer Data Register B (STDB)

Access in demultiplexed µP-interface mode: read/write address: $04_H$
Access in multiplexed µP-interface mode: read/write address: $08_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| MTDB7 | MTDB6 | MTDB5 | MTDB4 | MTDB3 | MTDB2 | MTDB1 | MTDB0 |

The STDA-register buffers the data transferred over the synchronous transfer channel A.

MTDA7 to MTDA0 hold the bits 7 to 0 of the respective time slot. MTDA7 (MSB) is the bit transmitted/received first, MTDA0 (LSB) the bit transmitted/received last over the serial interface.

### 4.6.18 Synchronous Transfer Receive Address Register A (SARA)

Access in demultiplexed µP-interface mode: read/write address: $05_H$
Access in multiplexed µP-interface mode: read/write address: $0A_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| ISRA | MTRA6 | MTRA5 | MTRA4 | MTRA3 | MTRA2 | MTRA1 | MTRA0 |

The SARA-register specifies for synchronous transfer channel A from which input interface, port and time slot the serial data is extracted. This data can then be read from the STDA-register.

**ISRA** Interface Select Receive for channel A.

0… selects the PCM-interface as the input interface for synchronous channel A.

1… selects the CFI-interface as the input interface for synchronous channel A.

**MTRA6..0** µP-Transfer Receive Address for channel A; selects the port and time slot number at the interface selected by ISRA according to **tables 16** and **17**: MTRA6..0 = MA6..0.

### 4.6.19 Synchronous Transfer Receive Address Register B (SARB)

Access in demultiplexed µP-interface mode: read/write address: $06_H$
Access in multiplexed µP-interface mode: read/write address: $0C_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| ISRB | MTRB6 | MTRB5 | MTRB4 | MTRB3 | MTRB2 | MTRB1 | MTRB0 |

The SARB-register specifies for synchronous transfer channel B from which input interface, port and time slot the serial data is extracted. This data can then be read from the STDB register.

**ISRB** Interface Select Receive for channel B.
0… selects the PCM-interface as the input interface for synchronous channel B.
1… selects the CFI-interface as the input interface for synchronous channel B.

**MTRB6..0** µP-Transfer Receive Address for channel B; selects the port and time slot number at the interface selected by ISRB according to **tables 16** and **17**: MTRB6..0 = MA6..0.

### 4.6.20 Synchronous Transfer Transmit Address Register A (SAXA)

Access in demultiplexed µP-interface mode: read/write address: $07_H$
Access in multiplexed µP-interface mode: read/write address: $0E_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| ISXA | MTXA6 | MTXA5 | MTXA4 | MTXA3 | MTXA2 | MTXA1 | MTXA0 |

The SAXA-register specifies for synchronous transfer channel A to which output interface, port and time slot the serial data contained in the STDA-register is sent.

**ISXA** Interface Select Transmit for channel A.
0… selects the PCM-interface as the output interface for synchronous channel A.
1… selects the CFI-interface as the output interface for synchronous channel A.

**MTXA6..0** µP-Transfer Transmit Address for channel A; selects the port and time slot number at the interface selected by ISXA according to **tables 16** and **17**: MTXA6..0 = MA6..0.

### 4.6.21 Synchronous Transfer Transmit Address Register B (SAXB)

Access in demultiplexed µP-interface mode: read/write address: $08_H$
Access in multiplexed µP-interface mode: read/write address: $10_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| ISXB | MTXB6 | MTXB5 | MTXB4 | MTXB3 | MTXB2 | MTXB1 | MTXB0 |

The SAXB-register specifies for synchronous transfer channel B to which output interface, port and time slot the serial data contained in the STDB-register is sent.

**ISXB** Interface Select Transmit for channel B.
0… selects the PCM-interface as the output interface for synchronous channel B.
1… selects the CFI-interface as the output interface for synchronous channel B.

**MTXB6..0** µP-Transfer Transmit Address for channel B; selects the port and time slot number at the interface selected by ISXB according to **tables 16** and **17**: MTXB6..0 = MA6..0.

### 4.6.22 Synchronous Transfer Control Register (STCR)

Access in demultiplexed µP-interface mode: read/write address: $09_H$
Access in multiplexed µP-interface mode: read/write address: $12_H$
Reset value: $00xxxxxx_B$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| TBE | TAE | CTB2 | CTB1 | CTB0 | CTA2 | CTA1 | CTA0 |

The STCR-register bits are used to enable or disable the synchronous transfer utility and to determine the sub time slot bandwidth and position if a PCM-interface time slot is involved.

**TAE, TBE** Transfer Channel A (B) Enable.
1… enables the µP transfer of the corresponding channel.
0… disables the µP transfer of the corresponding channel.

**CTA2..0** Channel Type A (B); these bits determine the bandwidth of the channel and the position of the relevant bits in the time slot acoording to the table below.

**CTB2..0** Note that if a CFI time slot is selected as receive or transmit time slot of the synchronous transfer, the 64-kBit/s bandwidth must be selected (CT#2..CT#0 = 001).

| CT#2 | CT#1 | CT#0 | Bandwidth | Transferred Bits |
|---|---|---|---|---|
| 0 | 0 | 0 | not allowed | – |
| 0 | 0 | 1 | 64 kBit/s | bits 7..0 |
| 0 | 1 | 0 | 32 kBit/s | bits 3..0 |
| 0 | 1 | 1 | 32 kBit/s | bits 7..4 |
| 1 | 0 | 0 | 16 kBit/s | bits 1..0 |
| 1 | 0 | 1 | 16 kBit/s | bits 3..2 |
| 1 | 1 | 0 | 16 kBit/s | bits 5..4 |
| 1 | 1 | 1 | 16 kBit/s | bits 7..6 |

### 4.6.23 MF-Channel Active Indication Register (MFAIR)

Access in demultiplexed µP-interface mode: read/write address: $0A_H$
Access in multiplexed µP-interface mode: read/write address: $14_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | SO | SAD5 | SAD4 | SAD3 | SAD2 | SAD1 | SAD0 |

This register is only used in IOM-2 applications (active handshake protocol) in order to identify active monitor channels when the "Search for active monitor channels" command (CMDR:MFSO) has been executed.

**SO** MF Channel Search On.
0…the search is completed.
1…the EPIC-1 is still busy looking for an active channel.

**SAD5..0** Subscriber Address 5..0; after an ISTA:MAC-interrupt these bits point to the port and time slot where an active channel has been found. The coding is identical to MFSAR:SAD5..SAD0.

### 4.6.24 MF-Channel Subscriber Address Register (MFSAR)

Access in demultiplexed µP-interface mode: read/write address: $0A_H$
Access in multiplexed µP-interface mode: read/write address: $14_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| MFTC1 | MFTC0 | SAD5 | SAD4 | SAD3 | SAD2 | SAD1 | SAD0 |

The exchange of monitor data normally takes place with only one subscriber circuit at a time. This register serves to point the MF-handler to that particular CFI time slot.

**MFTC1..0** MF Channel Transfer Control 1..0; these bits, in addition to CMDR:MFT1,0 and OMDR:MFPS control the MF-channel transfer as indicated in **table 21**.

**SAD5..0** Subscriber address 5..0; these bits define the addressed subscriber. The CFI time slot encoding is similar to the one used for Control Memory accesses using the MAAR-register (**tables 19** and **20**):

CFI time slot encoding of MFSAR derived from MAAR:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| MAAR: | MA7 | MA6 | MA5 | MA4 | MA3 | MA2 | MA1 | MA0 |
| | | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | |
| MFSAR: | MFTC1 | MFTC0 | SAD5 | SAD4 | SAD3 | SAD2 | SAD1 | SAD0 |

MAAR:MA7 selects between upstream and downstream CM-blocks. This information is not required since the transfer direction is defined by CMDR (transmit or receive).

MAAR:MA0 selects between even and odd time slots. This information is also not required since MF-channels are always located on even time slots.

### 4.6.25 Monitor/Feature Control Channel FIFO (MFFIFO)

Access in demultiplexed µP-interface mode: read/write address: $0B_H$
Access in multiplexed µP-interface mode: read/write address: $16_H$
Reset value: empty

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| MFD7 | MFD6 | MFD5 | MFD4 | MFD3 | MFD2 | MFD1 | MFD0 |

The 16-byte bi-directional MFFIFO provides intermediate storage for data bytes to be transmitted or received over the monitor or feature control channel.

**MFD7..0** MF Data bits 7..0; MFD7 (MSB) is the first bit to be sent over the serial CFI, MFD0 (LSB) the last.

*Note: The byte n + 1 of an n-byte transmit message in monitor channel is not defined.*

### 4.6.26 Signaling FIFO (CIFIFO)

Access in demultiplexed µP-interface mode: read address: $0C_H$
Access in multiplexed µP-interface mode: read address: $18_H$
Reset value: $0xxxxxxx_B$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| SBV | SAD6 | SAD5 | SAD4 | SAD3 | SAD2 | SAD1 | SAD0 |

The 9 byte deep CIFIFO stores the addresses of CFI time slots in which a C/I- and/or a SIG-value change has taken place. This address information can then be used to read the actual C/I- or SIG-value from the control memory.

**SBV** Signaling Byte Valid.

0… the SAD6..0 bits are invalid.

1… the SAD6..0 bits indicate a valid subscriber address. The polarity of SBV is chosen such that the whole 8 bits of the CIFIFO can be copied to the MAAR register in order to read the upstream C/I- or SIG-value from the control memory.

**SAD6..0** Subscriber Address bits 6..0; The CM-address which corresponds to the CFI time slot where a C/I- or SIG-value change has taken place is encoded in these bits. For C/I-channels SAD6..0 point to an even CM-address (C/I-value), for SIG-channels SAD6..0 point to an odd CM-address (stable SIG-value).

### 4.6.27 Timer Register (TIMR)

Access in demultiplexed µP-interface mode: write address: $0C_H$
Access in multiplexed µP-interface mode: write address: $18_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| SSR | TVAL6 | TVAL5 | TVAL4 | TVAL3 | TVAL2 | TVAL2 | TVAL0 |

The EPIC-1 timer can be used for 3 different purposes: timer interrupt generation (ISTA:TIG), FSC multiframe generation (CMD2:FC2..0 = 111) and last look period generation.

**SSR** Signaling Sampling Rate.
0… the last look period is defined by TVAL6..0.
1… the last look period is fixed to 125 µs.

**TVAL6..0** Timer Value bits 6..0; the timer period, equal to (1+TVAL6..0) × 250 µs, is programmed here. It can thus be adjusted within the range of 250 µs up to 32 ms.

The timer is started as soon as CMDR:ST is set to 1 and stopped by writing the TIMR-register or by selecting OMDR:OMS0 = 0.

### 4.6.28 Status Register EPIC®-1 (STAR_E)

Access in demultiplexed µP-interface mode: read address: $0D_H$
Access in multiplexed µP-interface mode: read address: $1A_H$
Reset value: $05_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| MAC | TAC | PSS | MFTO | MFAB | MFAE | MFRW | MFFE |

The status register STAR displays the current state of certain events within the EPIC-1. The STAR register bits do not generate interrupts and are not modified by reading STAR.

**MAC** Memory Access
0…no memory access is in operation.
1… a memory access is in operation. Hence, the memory access registers may not be used.

*Note: MAC is also set and reset during synchronous transfers.*

**TAC** Timer Active
0… the timer is stopped.
1… the timer is running.

**PSS** PCM-Synchronization Status.
1… the PCM-interface is synchronized.
0… the PCM-interface is not synchronized. There is a mismatch between the PBNR-value and the applied clock and framing signals (PDC/PFS) or OMDR:OMS0 = 0.

**MFTO** MF-Channel Transfer in Operation.
0… no MF-channel transfer is in operation.
1… an MF-channel transfer is in operation.

**MFAB** MF-Channel Transfer Aborted.
0… the remote receiver did not abort a handshake message transfer.
1… the remote receiver aborted a handshake message transfer.

**MFAE** MFFIFO-Access Enable.
0… the MFFIFO may not be accessed.
1… the MFFIFO may be either read or written to.

**MFRW** MFFIFO Read/Write.
0… the MFFIFO is ready to be written to.
1… the MFFIFO may be read.

**MFFE** MFFIFO Empty
0… the MFFIFO is not empty.
1… the MFFIFO is empty.

### 4.6.29 Command Register EPIC®-1 (CMDR_E)

Access in demultiplexed µP-interface mode: write address: $0D_H$
Access in multiplexed µP-interface mode: write address: $1A_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | ST | TIG | CFR | MFT1 | MFT0 | MFSO | MFFR |

Writing a logical 1 to a CMDR-register bit starts the respective operation.

**ST** Start Timer.
0… not action. If the timer shall be stopped, the TIMR-register must simply be written with a random value.
1… starts the timer to run cyclically from 0 to the value programmed in TIMR:TVAL6..0.

**TIG** Timer Interrupt Generation.
0… setting the TIG-bit to logical 0 together with the CMDR:ST-bit set to logical 1 disables the interrupt generation.
1… setting the TIG-bit to logical 1 together with CMDR:ST-bit set to logical 1 causes the EPIC-1 to generate a periodic interrupt (ISTA:TIN) each time the timer expires.

**CFR** CIFIFO Reset.
0… no action.
1… resets the signaling FIFO within 2 RCL-periods, i.e. all entries and the ISTA:SFI-bit are cleared.

**MFT1..0** MF-channel Transfer Control Bits 1,0; these bits start the monitor transfer enabling the contents of the MFFIFO to be exchanged with the subscriber circuits as specified in MFSAR. The function of some commands depends furthermore on the selected protocol (OMDR:MFPS). **Table 21** summarizes all available MF-commands.

**MFSO** MF-channel Search On.

0… no action.

1… the EPIC-1 starts to search for active MF-channels. Active channels are characterized by an active MX-bit (logical 0) sent by the remote transmitter. If such a channel is found, the corresponding address is stored in MFAIR and an ISTA:MAC-interrupt is generated. The search is stopped when an active MF-channel has been found or when OMDR:OMS0 is set to 0.

**MFFR** MFFIFO Reset.

0… no action

1… resets the MFFIFO and all operations associated with the MF-handler (except for the search function) within 2 RCL-periods. The MFFIFO is set into the state "MFFIFO empty", write access enabled and any monitor data transfer currently in process will be aborted.

**Table 21**
**Summary of MF-Channel Commands**

| Transfer Mode | CMDR: MFT1,MFT0 | MFSAR | Protocol Selection | Application |
|---|---|---|---|---|
| Inactive | 00 | xxxxxxxx | HS, no HS | idle state |
| Transmit | 01 | 00 SAD5..0 | HS, no HS | IOM-2, IOM-1, SLD |
| Transmit broadcast | 01 | 01xxxxxx | HS, no HS | IOM-2, IOM-1, SLD |
| Test operation | 01 | 10------ | HS, no HS | IOM-2, IOM-1, SLD |
| Transmit continuous | 11 | 00 SAD5..0 | HS | IOM-2 |
| Transmit + receive same time slot | | | | |
| Any # of bytes | 10 | 00 SAD5..0 | HS | IOM-2 |
| 1 byte expected | 10 | 00 SAD5..0 | no HS | IOM-1 |
| 2 bytes expected | 10 | 01 SAD5..0 | no HS | (IOM-1) |
| 8 bytes expected | 10 | 10 SAD5..0 | no HS | (IOM-1) |
| 16 bytes expected | 10 | 11 SAD5..0 | no HS | (IOM-1) |
| Transmit + receive same line | | | | |
| 1 byte expected | 11 | 00 SAD5..0 | no HS | SLD |
| 2 bytes expected | 11 | 01 SAD5..0 | no HS | SLD |
| 8 bytes expected | 11 | 10 SAD5..0 | no HS | SLD |
| 16 bytes expected | 11 | 11 SAD5..0 | no HS | SLD |

HS: handshake facility enabled (OMDR:MFPS = 1)

no HS: handshake facility disable (OMDR:MFPS = 0)

### 4.6.30 Interrupt Status Register EPIC®-1 (ISTA_E)

Access in demultiplexed µP-interface mode: read address: $0E_H$
Access in multiplexed µP-interface mode: read address: $1C_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| TIN | SFI | MFFI | MAC | PFI | PIM | SIN | SOV |

The ISTA-register should be read after an interrupt in order to determine the interrupt source.

**TIN** Timer interrupt; a timer interrupt previously requested with CMDR:ST,TIG = 1 has occurred. The TIN-bit is reset by reading ISTA. It should be noted that the interrupt generation is periodic, i.e. unless stopped by writing to TIMR, the ISTA:TIN will be generated each time the timer expires.

**SFI** Signaling FIFO-Interrupt; this interrupt is generated if there is at least one valid entry in the CIFIFO indicating a change in a C/I- or SIG-channel. Reading ISTA does not clear the SFI-bit. Instead SFI is cleared if the CIFIFO is empty which can be accomplished by reading all valid entries of the CIFIFO or by resetting the CIFIFO by setting CMDR:CFR to 1.

**MFFI** MFFIFO-Interrupt; the last MF-channel command (issued by CMDR:MFT1,MFT0) has been executed and the EPIC-1 is ready to accept the next command. Additional information can be read from STAR:MFTO…MFFE. MFFI is reset by reading ISTA.

**MAC** Monitor channel Active interrupt; the EPIC-1 has found an active monitor channel. A new search can be started by reissuing the CMDR:MFSO-command. MAC is reset by reading ISTA.

**PFI** PCM-Framing Interrupt; the STAR:PSS-bit has changed its polarity. To determine whether the PCM-interface is synchronized or not, STAR must be read. The PFI-bit is reset by reading ISTA.

**PIM** PCM-Input Mismatch; this interrupt is generated immediately after the comparison logic has detected a mismatch between a pair of PCM-input lines. The exact reason for the interrupt can be determined by reading the PICM-register. Reading ISTA clears the PIM-bit. A new PIM-interrupt can only be generated after the PICM-register has been read.

**SIN** Synchronous transfer Interrupt; The SIN-interrupt is enabled if at least one synchronous transfer channel (A and/or B) is enabled via the STCR:TAE, TBE-bits. The SIN-interrupt is generated when the access window for the µP opens. After the occurrence of the SIN-interrupt the µP can read and/or write the synchronous transfer data registers (STDA, STDB). The SIN-bit is reset by reading ISTA.

**SOV** Synchronous transfer Overflow; The SOV-interrupt is generated if the µP fails to access the data registers (STDA, STDB) within the access window. The SOV-bit is reset by reading ISTA.

### 4.6.31 Mask Register EPIC®-1 (MASK_E)

Access in demultiplexed µP-interface mode: write address: $0E_H$
Access in multiplexed µP-interface mode: write address: $1C_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| TIN | SFI | MFFI | MAC | PFI | PIM | SIN | SOV |

A logical 1 disables the corresponding interrupt as described in the ISTA-register.

A masked interrupt is stored internally and reported in ISTA immediately if the mask is released. However, an SFI-interrupt is also reported in ISTA if masked. In this case no interrupt is generated. When writing register MASK_E while ISTA_E indicates a non masked interrupt $\overline{INT}$ is temporarily set into the inactive state.

### 4.6.32 Operation Mode Register (OMDR)

Access in demultiplexed µP-interface mode: read/write address: $0F_H$
Access in multiplexed µP-interface mode: read/write address: $1E_H/3E_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| OMS1 | OMS0 | PSB | PTL | COS | MFPS | CSB | RBS |

**OMS1..01** Operational Mode Selection; these bits determine the operation mode of the EPIC-1 is working in according to the following table:

| OMS1..0 | Function |
|---|---|
| 00 | The **CM-reset mode** is used to reset all locations of the control memory code and data fields with a single command within only 256 RCL-cycles. A typical application is resetting the CM with the command MACR = $70_H$ which writes the contents of MADR ($xx_H$) to all data field locations and the code '0000' (unassigned channel) to all code field locations. A CM-reset should be made after each hardware reset. In the CM-reset mode the EPIC-1 does not operate normally i.e. the CFI- and PCM-interfaces are not operational. |
| 10 | The **CM-initialization mode** allows fast programming of the control memory since each memory access takes a maximum of only 2.5 RCL-cycles compared to the 9.5 RCL-cycles in the normal mode. Accesses are performed on individual addresses specified by MAAR. The initialization of control/signaling channels in IOM- or SLD- applications can for example be carried out in this mode. In the CM- initialization mode the EPIC-1 does also not work normally. |
| 11 | In the **normal operation mode** the CFI- and PCM-interfaces are operational. Memory accesses performed on single addresses (specified by MAAR) take 9.5 RCL-cycles. An initialization of the complete data memory tristate field takes 1035 RCL-cycles. |
| 01 | In **test mode** the EPIC-1 sustains normal operation. However memory accesses are no longer performed on a specific address defined by MAAR, but on all locations of the selected memory, the contents of MAAR (including the U/D-bit!) being ignored. A test mode access takes 2057 RCL-cycles. |

**PSB** PCM-Standby.

0…the PCM-interface output pins TxD0..3 are set to high impedance and those $\overline{\text{TSC}}$-pins that are actually used as tristate control signals are set to logical 1 (inactive).

1…the PCM-output pins transmit the contents of the upstream data memory or may be set to high impedance via the data memory tristate field.

**PTL** PCM-Test Loop.

0…the PCM-test loop is disabled.

1…the PCM-test loop is enabled, i.e. the physical transmit pins TxD# are internally connected to the corresponding physical receive pins RxD#, such that data transmitted over TxD# are internally looped back to RxD# and data externally received over RxD# are ignored. The TxD# pins still output the contents of the upstream data memory according to the setting of the tristate field (only modes 0 and 1; mode 1 with AIS-bit set).

**COS** CFI-Output driver Selection.

0…the CFI-output drivers are tristate drivers.

1…the CFI-output drivers are open drain drivers.

**MFPS** Monitor/Feature control channel Protocol Selection.

0…handshake facility disabled (SLD and IOM-1 applications)

1…handshake facility enabled (IOM-2 applications)

**CSB** CFI-Standby.

0…the CFI-interface output pins DD0..3, DU0..3, DCL and FSC are set to high impedance.

1…the CFI-output pins are active.

**RBS** Register Bank Selection. Used in demultiplexed data/address modes only.

The RBS-bit is internally ORed with the A4 address pin. The EPIC-1 registers can therefore be accessed using two different methods:

1) If RBS is always set to logical 0, the registers can be accessed using all 5 address pins A4..A0.

2) If A4 is externally set to logical 0 during EPIC-1 accesses, the RBS-bit has to be set to

   0…to access the registers used during device initialization

   1…to access the registers used during device operation.

### 4.6.33 Version Number Status Register (VNSR)

Access in demultiplexed µP-interface mode: write address: $1D_H$
Access in multiplexed µP-interface mode: write address: $3A_H$
Reset value: $0x_H$

bit 7 bit 0

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| IR | 0 | 0 | SWRX | VN3 | VN2 | VN1 | VN0 |

The VNSR-register bits do not generate interrupts and are not modified by reading VNSR. The IR and VN3..0 bits are read only bits, the SWRX-bit is a write only bit.

**IR** Initialization Request; this bit is set to logical 1 after an inappropriate clocking or after a power failure. It is reset to logical 0 after a control memory reset command: OMDR:OMS1..0 = 00, MACR = 7X.

**SWRX** Software Reset External.

When set, the pin RESIN is activated. RESIN is reset with the next EPIC-1 interrupt, i.e. the EPIC-1 timer may be used to generate a RESIN-pulse without generating an internal ELIC-reset.

**VN3..0** Version status Number; these bits display the EPIC-1 chip version as follows

| VN3..0 | Chip Versions |
|---|---|
| 0001 | V1.2 |

## 4.7 SACCO

### 4.7.1 Receive FIFO (RFIFO)

Access in demultiplexed µP-interface mode: read address (Ch-A/Ch-B): $00_H..1F_H/40_H..5F_H$
Access in multiplexed µP-interface mode: read address: (Ch-A/Ch-B): $00_H..3E_H/80_H..BE_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 |

**RD7..0** Receive Data 7…0, data byte received on the serial interface.

**Interrupt controlled data transfer** (interrupt mode, selected if DMA-bit in register XBCH is reset).

Up to 32 bytes of received data can be read from the RFIFO following an RPF or an RME interrupt.

RPF-interrupt: exactly 32 bytes to be read.

RME-interrupt: the number of bytes can be determined reading the registers RBCL, RBCH.

**DMA controlled data transfer** (DMA-mode, selected if DMA-bit in register XBCH is set).

If the RFIFO contains 32 bytes, the SACCO autonomously requests a block data transfer by activating the DRQRA/B-line as long as the 31st read cycle is finished. This forces the DMA-controller to continuously perform bus cycles until 32 bytes are transferred from the SACCO to the system memory (DMA-controller mode: demand transfer, level triggered).

If the RFIFO contains less than 32 bytes (one short frame or the last bytes of a long frame) the SACCO requests a block data transfer depending on the contents of the RFIFO according to the following table:

| RFIFO Contents (bytes) | DMA Transfers (bytes) |
|---|---|
| (1), 2, 3 | 4 |
| 4 - 7 | 8 |
| 8 - 15 | 16 |
| 16 - 32 | 32 |

Additionally an RME-interrupt is issued after the last byte has been transferred. As a result, the DMA-controller may transfer more bytes as actually valid in the current received frame. The valid byte count must therefore be determined reading the registers RBCH, RBCL following the RME-interrupt.

The corresponding DRQRA/B pin remains "high" as long as the RFIFO requires data transfers. It is deactivated upon the rising edge of the 31st DMA-transfer or, if n < 32 or n is the remainder of a long frame, upon the falling edge of the last DMA-transfer.

If n $\geq$ 32 and the DMA-controller does not perform the 32nd DMA-cycle, the DRQRA/B-line will go high again as soon as $\overline{CSS}$ goes high, thus indicating further bytes to fetch.

### 4.7.2 Transmit FIFO (XFIFO)

Access in demultiplexed µP-interface mode: write address (Ch-A/Ch-B): $00_H..1F_H/40_H..5F_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $00_H..3E_H/B0_H..BE_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| TD7 | TD6 | TD5 | TD4 | TD3 | TD2 | TD1 | TD0 |

**TD7..0** Transmit Data 7…0, data byte to be transmitted on the serial interface.

**Interrupt controlled data transfer** (interrupt mode, selected if DMA-bit in register XBCH is reset).

Up to 32 bytes of transmit data can be written to the XFIFO following an XPR-interrupt.

**DMA controlled data transfer** (DMA-mode, selected if DMA-bit in register XBCH is set).

Prior to any data transfer, the actual byte count of the frame to be transmitted must be written to the registers XBCH, XBCL:

1 byte: XBCL = 0

n bytes: XBCL = n – 1

If a data transfer is then initiated via the CMDR-register (commands XPD/XTF or XDD), the SACCO autonomously requests the correct amount of block data transfers (n $\times$ 32 + remainder, n = 0,1, …).

The corresponding DRQTA/B pin remains "high" as long as the XFIFO requires data transfers. It is deactivated upon the rising edge of $\overline{WR}$ in the DMA-transfer 31 or n – 1 respectively. The DMA-controller must take care to perform the last DMA-transfer. If it is missing, the DRQTA/B-line will go active again when $\overline{CSS}$ is raised.

### 4.7.3 Interrupt Status Register (ISTA_A/B)

Access in demultiplexed
µP-interface mode: read address: (Ch-A/Ch-B): $20_H/60_H$
Access in multiplexed
µP-interface mode: read address: (Ch-A/Ch-B): $40_H/C0_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RME | RPF | 0 | XPR | 0 | 0 | 0 | 0 |

**RME** Receive Message End.
A message of up to 32 bytes or the last part of a message greater then 32 bytes has been received and is now available in the RFIFO. The message is complete! The actual message length can be determined by reading the registers RBCL, RBCH. RME is not generated when an extended HDLC-frame is recognized in auto-mode (EHC interrupt).
In DMA-mode a RME-interrupt is generated after the DMA-transfer has been finished correctly, indicating that the processor should read the registers RBCH/RBCL to determine the correct message length.

**RPF** Receive Pool Full.
A data block of 32 bytes is stored in the RFIFO. The message is not yet completed!

*Note: This interrupt is only generated in interrupt mode (not in DMA-mode).*

**XPR** Transmit Pool Ready.
A data block of up to 32 bytes can be written to the XFIFO.

### 4.7.4 Mask Register (MASK_A/B)

Access in demultiplexed
µP-interface mode: write address: (Ch-A/Ch-B): $20_H/60_H$
Access in multiplexed
µP-interface mode: write address: (Ch-A/Ch-B): $40_H/C0_H$
Reset value: $00_H$ (all interrupts enabled)

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RME | RPF | 0 | XPR | 0 | 0 | 0 | 0 |

**RME** enables(0)/disables(1) the Receive Message End interrupt.

**RPF** enables(0)/disables(1) the Receive Pool Full interrupts.

**XPR** enables(0)/disables(1) the Transmit Pool Ready interrupt.

Each interrupt source can be selectively masked by setting the respective bit in the MASK_A/B-register (bit position corresponding to the ISTA_A/B-register). Masked interrupts are internally stored but not indicated when reading ISTA_A/B and also not flagged into the top level ISTA. After releasing the respective MASK_A/B-bit they will be indicated again in ISTA_A/B and in the top level ISTA.

When writing register MASK_A/B while ISTA_A/B indicates a non masked interrupt the $\overline{INT}$-pin is temporarily set into the inactive state. In this case the interrupt remains indicated in the ISTA_A/B until these registers are read.

### 4.7.5 Extended Interrupt Register (EXIR_A/B)

Access in demultiplexed
µP-interface mode: read address: (Ch-A/Ch-B): $24_H/64_H$
Access in multiplexed
µP-interface mode: read address: (Ch-A/Ch-B): $48_H/C8_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| XMR | XDU/EXE | EHC | RFO | 0 | RFS | 0 | 0 |

**XMR** Transmit Message Repeat.

The transmission of a frame has to be repeated because:

– A frame consisting of more then 32 bytes is polled a second time in auto-mode.

– Collision has occurred after sending the 32nd data byte of a message in a bus configuration.

– CTS (transmission enable) has been withdrawn after sending the 32nd data byte of a message in point-to-point configuration.

**XDU/EXE** Transmission Data Underrun/Extended transmission End.
The actual frame has been aborted with IDLE, because the XFIFO holds no further data, but the frame is not yet complete according to registers XBCH/ XBCL.
In extended transparent mode, this bit indicates the transmission end condition.

*Note: It is not possible to transmit frames when a XMR- or XDU-interrupt is indicated.*

**EHC** Extended HDLC-frame.
The SACCO has received a frame in auto-mode which is neither a RR- nor an I-frame. The control byte is stored temporarily in the RHCR-register but not in the RFIFO.

**RFO** Receive Frame Overflow.
A frame could not be stored due to the occupied RFIFO (i.e. whole frame has been lost). This interrupt can be used for statistical purposes and indicates, that the CPU does not respond quickly enough to an incoming RPF- or RME-interrupt.

**RFS** Receive Frame Start.
This is an early receiver interrupt activated after the start of a valid frame has been detected, i.e. after a valid address check in operation modes providing address recognition, otherwise after the opening flag (transparent mode 0), delayed by two bytes.
After a RFS-interrupt the contents of

- RHCR
- RAL1
- RSTA bit3-0

are valid and can by read by the CPU.

The RFS-interrupt is maskable by programming bit CCR2:RIE.

### 4.7.6 Command Register (CMDR)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $21_H/61_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $42_H/C2_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RMC | RHR | AREP/ XREP | 0 | XPD/ XTF | XDD | XME | XRES |

*Note: The maximum time between writing to the CMDR-register and the execution of the command is 2.5 HDC-clock cycles. Therefore, if the CPU operates with a very high clock speed in comparison to the SACCO-clock, it is recommended that the bit STAR:CEC is checked before writing to the CMDR-register to avoid loosing of commands.*

**RMC** Receive Message Complete.
A '1' confirms, that the actual frame or data block has been fetched following a RPF- or RME-interrupt, thus the occupied space in the RFIFO can be released.

*Note: In DMA-mode this command is only issued once after a RME-interrupt. The SACCO does not generate further DMA requests prior to the reception of this command.*

**RHR** Reset HDLC-Receiver.
A '1' deletes all data in the RFIFO and in the HDLC-receiver.

**AREP/ XREP** Auto Repeat/Transmission Repeat.

- Auto-mode: AREP
  The frame (max. length 32 byte) stored in XFIFO can be polled repeatedly by the opposite station until the frame is acknowledged.
- Extended transparent mode 0,1: XREP
  Together with XTF- and XME-set (CMDR = $2A_H$) the SACCO repeatedly transmits the contents of the XFIFO (1…32 bytes) fully transparent without HDLC-framing, i.e. without flag, CRC-insertion, bit stuffing.
  The cyclical transmission continues until the command (CMDR:XRES) is executed or the bit XREP is reset. The inter frame timefill pattern is issued afterwards.
  When resetting XREP, data transmission is stopped after the next XFIFO-cycle is completed, the XRES-command terminates data transmission immediately.

*Note: MODE:CFT must be set to '0' when using cyclic transmission.*

**XPD/XTF** Transmit Prepared Data/Transmit Transparent Frame.

- Auto-mode: XPD
  Prepares the transmission of an I-frame ("prepared data") in auto-mode. The actual transmission starts, when the SACCO receives an I-frame with poll-bit set and AxH as the first data byte (PBC-command "transmit prepared data"). Upon the reception of a different poll frame a response is generated automatically (RR-poll ⇒ RR-response, I-poll with first byte not AxH ⇒ I-response).
- Non-auto-mode, transparent mode 0,1: XTF
  The transmission of the XFIFO contents is started, an opening flag sequence is automatically added.
- Extended transparent mode 0,1: XTF
  The transmission of the XFIFO contents is started, no opening flag sequence is added.

**XDD** Transmit Direct Data (auto-mode only!).
Prepares the transmission of an I-frame ("direct data") in auto-mode. The actual transmission starts, when the SACCO receives a RR-frame with poll-bit set. Upon the reception of an I-frame with poll-bit set, an I-response is issued.

**XME** Transmit Message End (interrupt mode only).
A ’1’ indicate that the data block written last to the XFIFO completes the actual frame. The SACCO can terminate the transmission operation properly by appending the CRC and the closing flag sequence to the data. XME is used only in combination with XPD/XTF or XDD.

*Note: When using the DMA-mode XME must not be used.*

**XRES** Transmit Reset.
The contents of the XFIFO is deleted and IDLE is transmitted. This command can be used by the CPU to abort a frame currently in transmission. After setting XRES a XPR-interrupt is generated in every case.

### 4.7.7 Mode Register (MODE)

Access in demultiplexed µP-interface mode: read/write address: (Ch-A/Ch-B): $22_H/62_H$
Access in multiplexed µP-interface mode: read/write address: (Ch-A/Ch-B): $44_H/C4_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| MDS1 | MDS0 | ADM | CFT | RAC | 0 | 0 | TLP |

**MDS1..0** Mode Select.
The operating mode of the HDLC-controller is selected.
00…auto-mode
01…non-auto-mode
10…transparent mode (D-channel arbiter)
11…extended transparent mode

**ADM** Address Mode.

The meaning of this bit varies depending on the selected operating mode:

- Auto-mode / non-auto mode
  Defines the length of the HDLC-address field.
  0…8-bit address field,
  1…16-bit address field.
- Transparent mode
  0…no address recognition: transparent mode 0 (D-channel arbiter)
  1…high byte address recognition: transparent mode 1
- Extended transparent mode
  0…receive data in RAL1: extended transparent mode 0
  1…receive data in RFIFO and RAL1: extended transparent mode 1

*Note: In extended transparent mode 0 and 1 the bit MODE:RAC must be reset to enable fully transparent reception.*

**CFT** Continuous Frame Transmission.

1…When CFT is set the XPR-interrupt is generated immediately after the CPU accessible part of XFIFO is copied into the transmitter section.

0…Otherwise the XPR-interrupt is delayed until the transmission is completed (D-channel arbiter).

**RAC** Receiver Active.
Via RAC the HDLC-receiver can be activated/deactivated.
0…HDLC-receiver inactive
1…HDLC-receiver active
In extended transparent mode 0 and 1 RAC must be reset (HDLC-receiver disabled) to enable fully transparent reception.

**TLP** Test Loop.
When set input and output of the HDLC-channel are internally connected.
(transmitter channel A - receiver channel A
transmitter channel B - receiver channel B)
TXDA/B are active, RXDA/B are disabled.

### 4.7.8 Channel Configuration Register 1 (CCR1)

Access in demultiplexed µP-interface mode: read/write address: (Ch-A/Ch-B): $2F_H/6F_H$
Access in multiplexed µP-interface mode: read/write address: (Ch-A/Ch-B): $5E_H/DE_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| PU | SC1 | SC0 | ODS | ITF | CM2 | CM1 | CM0 |

**PU** Power-Down Mode.
0…power-down (standby), the internal clock is switched off.
Nevertheless, register read/write access is possible.
1…power-up (active).

**SC1..0** Serial Port Configuration
00…point to point configuration,
01…bus configuration, timing mode 1, data is output with the rising edge of the data clock on pin TxDA/B and evaluated 1/2 clock period later with the falling clock edge at pin CxDA/B
11…bus configuration, timing mode 2, data is output with the falling edge of the data clock and evaluated with the next falling clock edge. Thus one complete clock period is available between data output and evaluation.

**ODS** Output Driver Select.
Defines the function of the transmit data pin (TxDA/B).
0…TxDA/B-pin open drain output
1…TxDA/B-pin push-pull output

**Up to Version 1.2 when selecting a bus configuration only the open drain option must be selected.**

Compared to the Version 1.2 the **Version 1.3** provides new features:

Push-pull operation may be selected in bus configuration (up to Version 1.2 only open drain):

- When active TXDA / TXDB outputs serial data in push-pull-mode
- When inactive (interframe or inactive timeslots) TXDA / TXDB outputs ’1’

*Note: When bus configuration with direct connection of multiple ELIC’s is used open drain option is still recommended.*
*The push-pull option with bus configuration can only be used if an external tri-state buffer is placed between TXDA / TXDB and the bus.*
*Due to the delay of $\overline{TSCA}$ / $\overline{TSCB}$ in this mode (see description of bits SOC(0:1) in register CCR2 (**chapter 4.7.9**)) these signals cannot directly be used to enable this buffer.*

ITF Inter frame Time Fill.

Determines the "no data to send" state of the transmit data pin (TxDA/B).

0… continuous IDLE-sequences are output ('11111111' bit pattern). In a bus configuration (CCR1:SC0 = 1) ITF is implicitly set to '0' (continuous '1's are transmitted).

1… continuous FLAG-sequences are output ('01111110' bit pattern). In a bus configuration (CCR1:SC0 = 1) ITF is implicitly set to '0' (continuous '1's are transmitted).

*Note: ITF has to be set 0 if clock mode 3 is used.*

**CM2** Clock rate.
0…single rate data clock
1…double rate data clock

**CM1..0** Clock Mode.
Determines the mode in which the data clock is forwarded toward the receiver/transmitter.

00…clock mode 0: external data clock, permanently enabled.
01…clock mode 1: external data clock, gated by an enable strobe forwarded via pin HFS.
10…clock mode 2: external data clock, programmable time slot assignment, frame synchronization pulse forwarded via pin HFS.
11…clock mode 3: **internal data clock derived from the CFI**, gated by an internally generated enable strobe.

*Note: Clock mode 3 is only applicable for **SACCO-A** in combination with the D-channel arbiter.*

### 4.7.9 Channel Configuration Register 2 (CCR2)

Access in demultiplexed µP-interface mode: read/write address: (Ch-A/Ch-B): $2C_H/6C_H$
Access in multiplexed µP-interface mode: read/write address: (Ch-A/Ch-B): $58_H/D8_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| SOC1 | SOC0 | XCS0 | RCS0 | **TXDE** | **RDS** | RIE | 0 |

**SOC1, SOC0** The function of the TSCA/B-pin can be defined programming SOC1,SOC0.

- Bus configuration:
  - 00…the TSCA/B output is activated only during the transmission of a frame delayed by one clock period. **When transmission was stopped due to a collision TSCA/B remains inactive.**
  - 10…the TSCA/B-output is always high (disabled).
  - 11…the TSCA/B-output indicates the reception of a data frame (active low).
- Point-to-point configuration:
  - 0x…the TSCA/B-output is activated during the transmission of a frame.
  - 1x…the TSCA/B-output is activated during the transmission of a frame and of inter frame timefill.

**XCS0,** Transmit/receive Clock Shift, bit 0 (only clock mode 2).

**RCS0** Together with the bits XCS2, XCS1 (RCS2, RCS1) in TSAX (TSAR) the clock shift relative to the frame synchronization signal of the transmit (receive) time slot can be adjusted. A clock shift of 0…7 bits is programmable (clock mode 2 only!).

*Note: In the clock modes 0,1 and 3 XCS0 and RCS0 has to be set to '0'.*

**TXDE** Transmit Data Enable.
0…the pin TxDA/B is disabled (in the state high impedance).
1…the pin TxDA/B is enabled. Depending on the programming of bit CCR1:ODS it has a push pull or open drain characteristic.

**RDS** Receive Data Sampling.
0 : serial data on RXDA/B is sampled at the falling edge of HDCA/B.
1 : serial data on RXDA/B is sampled at the rising edge of HDCA/B.

*Note: With RDS = 1 the sampling edge is shifted 1/2 clock phase forward. The data is internally still processed with the falling edge.*

**RIE** Receive frame start Enable.
When set, the RFS-interrupt in register EXIR_A/B is enabled.

### 4.7.10 Receive Length Check Register (RLCR)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $2E_H/6E_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $5C_H/DC_H$
Reset value: $0xxxxxxx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RC | RL6 | RL5 | RL4 | RL3 | RL2 | RL1 | RL0 |

**RC** Receive Check enable.
A ’1’ enables, a ’0’ disables the receive frame length feature.

**RL6..0** Receive Length.
The maximum receive length after which data reception is suspended can be programmed in RL6**..**0. The maximum allowed receive frame length is (RL + 1) $\times$ 32 bytes. A frame exceeding this length is treated as if it was aborted by the opposite station (RME-interrupt, RAB-bit set (VFR in clock mode 3)).
In this case the receive byte count (RBCH, RBCL) is greater than the programmed receive length.

### 4.7.11 Status Register (STAR)

Access in demultiplexed µP-interface mode: read address: (Ch-A/Ch-B): $21_H/61_H$
Access in multiplexed µP-interface mode: read address: (Ch-A/Ch-B): $42_H/C2_H$
Reset value: $48_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| XDOV | XFW | AREP/ XREP | RFR | RLI | CEC | XAC | AFI |

**XDOV** Transmit Data Overflow.
A '1' indicates, that more than 32 bytes have been written into the XFIFO.

**XFW** XFIFO Write enable.
A '1' indicates, that data can be written into the XFIFO.

*Note: XFW is only valid when CEC = 0.*

**AREP/** Auto Repeat/Transmission Repeat.

**XREP** Read back value of the corresponding command bit CMDR:AREP/XREP.

**RFR** RFIFO Read enable.
A '1' indicates, that valid data is in the RFIFO and read access is enabled.
RFR is set with the RME- or RPF-interrupt and reset when executing the RMC-command.

**RLI** Receiver Line Inactive.
Neither flags as inter frame time fill nor frames are received via the receive line.

*Note: Significant in point-to-point configurations!*

**CEC** Command Execution.
When '0' no command is currently executed, the CMDR-register can be written to.
When '1' a command (written previously to CMDR) is currently executed, no further command must temporarily be written to the CMDR-register.

**XAC** Transmitter Active.
A '1' indicates, that the transmitter is currently active.
In bus mode the transmitter is considered active also when it waits for bus access.

**AFI** Additional Frame Indication.
A '1' indicates, that one or more completely received frames or the last part of a frame are in the CPU inaccessible part of the RFIFO.
In combination with the bit STAR:RFR multiple frames can be read out of the RFIFO without interrupt control.

### 4.7.12 Receive Status Register (RSTA)

Access in demultiplexed µP-interface mode: read address: (Ch-A/Ch-B): $27_H/67_H$
Access in multiplexed µP-interface mode: read address: (Ch-A/Ch-B): $4E_H/CE_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| VFR | RDO | CRC | RAB | HA1 | HA0 | C/R | LA |

RSTA always displays the momentary state of the receiver. Because this state can differ from the last entry in the FIFO it is reasonable to always use the status bytes in the FIFO.

**VFR** Valid Frame.

Indicates whether the received frame is valid (’1’) or not (’0’ invalid).

A frame is invalid when

– its length is not an integer multiple of 8 bits (n × 8 bits), e.g. 25 bit,

– its is to short, depending on the selected operation mode:

| | |
|---|---|
| auto-mode/non-auto mode (2-byte address field): | 4 bytes |
| auto-mode/non-auto mode (1-byte address field): | 3 bytes |
| transparent mode 1: | 3 bytes |
| transparent mode 0: | 2 bytes |

– a frame was aborted (note: VFR can also be set when a frame was aborted)

*Note: Shorter frames are not reported.*

**RDO** Receive Data Overflow.
A '1' indicates, that a RFIFO-overflow has occurred within the actual frame.

**CRC** CRC-Compare Check.
0: CRC check failed, received frame contains errors.
1: CRC check o.k., received frame is error free.

**RAB** Receive message Aborted.
When '1' the received frame was aborted from the transmitting station. According to the HDLC-protocol, this frame must be discarded by the CPU.

**HA1..0** High byte Address compare.
In operating modes which provide high byte address recognition, the SACCO compares the high byte of a 2-byte address with the contents of two individual programmable registers (RAH1, RAH2) and the fixed values FEH and FCH (group address). Depending on the result of the comparison, the following bit combinations are possible:

10…RAH1 has been recognized.
00…RAH2 has been recognized.
01…group address has been recognized.

*Note: If RAH1, RAH2 contain the identical value, the combination 00 will be omitted. HA1..0 is significant only in 2-byte address modes.*

**C/R** Command/Response; significant only, if 2-byte address mode has been selected. Value of the C/R bit (bit of high address byte) in the received frame.

**LA** Low byte Address compare.
The low byte address of a 2-byte address field or the single address byte of a 1-byte address field is compared with two programmable registers (RAL1, RAL2). Depending on the result of the comparison LA is set.
0…RAL2 has been recognized,
1…RAL1 has been recognized.
In non-auto mode, according to the X.25 LAP B-protocol, RAL1/RAL2 may be programmed to differ between COMMAND/RESPONSE frames.

*Note: The receive status byte is duplicated into the RFIFO (clock mode 0-2) following the last byte of the corresponding frame. In clock mode 3 a modified receive status byte is copied into RFIFO containing IOM-port and channel address of the received frame. Please refer to **chapter 2.2.7.6** the RFIFO in clock mode 3.*

### 4.7.13 Receive HDLC-Control Register (RHCR)

Access in demultiplexed µP-interface mode: read address: (Ch-A/Ch-B): $29_H/69_H$
Access in multiplexed µP-interface mode: read address: (Ch-A/Ch-B): $52_H/D2_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RHCR7 | RHCR6 | RHCR5 | RHCR4 | RHCR3 | RHCR2 | RHCR1 | RHCR0 |

**RHCR7..0** Receive HDLC-Control Register.

The contents of the RHCR depends on the selected operating mode.

- Auto-mode (1- or 2-byte address field):
  I-frame — compressed control field (bit 7-4: bit 7-4 of PBC-command, bit 3-0: bit 3-0 of HDLC-control field)
  else — HDLC-control field

*Note: RR-frames and I-frames with the first byte = AxH (PBCcommand "transmit prepared data") are handled automatically and are not transferred to the CPU (no interrupt is issued).*

- Non-auto mode (1-byte address field): 2nd byte after flag
- Non-auto mode (2-byte address field): 3rd byte after flag
- Transparent mode 1: 3nd byte after flag
- Transparent mode 0: 2nd byte after flag

*Note: The value in RHCR corresponds to the last received frame.*

### 4.7.14 Transmit Address Byte 1 (XAD1)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $24_H/64_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $48_H/C8_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| XAD17 | XAD16 | XAD15 | XAD14 | XAD13 | XAD12 | XAD11 | XAD10 |

**XAD17..10** Transmit Address byte 1.
The value stored in XAD1 is included automatically as the address byte (high address byte in case of 2-byte address field) of all frames transmitted in auto mode.
Using a 2 byte address field, XAD11 and XAD10 have to be set to ’0’.

### 4.7.15 Transmit Address Byte 2 (XAD2)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $25_H/65_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $4A_H/CA_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| XAD27 | XAD26 | XAD25 | XAD24 | XAD23 | XAD22 | XAD21 | XAD20 |

**XAD27..20** Transmit Address byte 2.
The value stored in XAD2 is included automatically as the low address byte of all frames transmitted in auto-mode (2-byte address field only).

### 4.7.16 Receive Address Byte Low Register 1 (RAL1)

Access in demultiplexed µP-interface mode: read/write address: (Ch-A/Ch-B): $28_H/68_H$
Access in multiplexed µP-interface mode: read/write address: (Ch-A/Ch-B): $50_H/D0_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RAL17 | RAL16 | RAL15 | RAL14 | RAL13 | RAL12 | RAL11 | RAL10 |

**RAL17..10** Receive Address byte Low register 1.
The general function (read/write) and the meaning or contents of this register depends on the selected operating mode:

- Auto-mode, non-auto mode (address recognition) - write only:
  compare value 1, address recognition (low byte in case of 2-byte address field).
- Transparent mode 1 (high byte address recognition) - read only:
  RAL1 contains the byte following the high byte of the address in the received frame (i.e. the second byte after the opening flag).
- Transparent mode 0 (no address recognition) - read only:
  contains the first byte after the opening flag (first byte of the received frame).
- Extended transparent mode 0,1 - read only:
  RAL1 contains the actual data byte currently assembled at the RxD-pin by passing the HDLC-receiver (fully transparent reception without HDLC-framing).

*Note: In auto-mode and non-auto mode the read back of the programmed value is inverted.*

### 4.7.17 Receive Address Byte Low Register 2 (RAL2)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $29_H/69_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $52_H/D2_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RAL27 | RAL26 | RAL25 | RAL24 | RAL23 | RAL22 | RAL21 | RAL20 |

**RAL27..20** Receive Address byte Low register 1.

- Auto-mode, non-auto mode (address recognition):
  compare value 2, address recognition (low byte in case of 2-byte address field).

*Note: Normally used for broadcast address.*

### 4.7.18 Receive Address Byte High Register 1 (RAH1)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $26_H/66_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $4C_H/CC_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RAH17 | RAH16 | RAH15 | RAH14 | RAH13 | RAH12 | 0 | 0 |

**RAL17..12** Receiver Address byte High register 1.

- Auto-mode, non-auto mode transparent mode 1, (2-byte address field). Compare value 1, high byte address recognition.

*Note: When a 1-byte address field is used in non-auto or auto-mode, RAH1 must be set to $00_H$.*

### 4.7.19 Receive Address Byte High Register 2 (RAH2)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $27_H/67_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $4E_H/CE_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RAH27 | RAH26 | RAH25 | RAH24 | RAH23 | RAH22 | 0 | 0 |

**RAL27..22** Receiver Address byte High register 2.

- Auto-mode, non-auto mode transparent mode 1, (2-byte address field). Compare value 2, high byte address recognition.

*Note: When a 1-byte address field is used in non-auto or auto-mode, RAH2 must be set to $00_H$.*

### 4.7.20 Receive Byte Count Low (RBCL)

Access in demultiplexed µP-interface mode: read address: (Ch-A/Ch-B): $25_H/65_H$
Access in multiplexed µP-interface mode: read address: (Ch-A/Ch-B): $4A_H/CA_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RBC7 | RBC6 | RBC5 | RBC4 | RBC3 | RBC2 | RBC1 | RBC0 |

**RBC7..0** Receive Byte Count.
Together with RBCH (bits RBC11 - RBC8), the length of the actual received frame (0…4095 bytes) can be determined. These registers must be read by the CPU following a RME interrupt.

### 4.7.21 Receive Byte Count High (RBCH)

Access in demultiplexed µP-interface mode: read address: (Ch-A/Ch-B): $2D_H/6D_H$
Access in multiplexed µP-interface mode: read address: (Ch-A/Ch-B): $5A_H/DA_H$
Reset value: $000xxxxx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| DMA | 0 | 0 | OV | RBC11 | RBC10 | RBC9 | RBC8 |

**DMA** DMA-mode status indication.
Read back value representing the DMA-bit programmed in register XBCH.

**OV** Counter Overflow.
A ’1’ indicates that more than 4095 bytes were received.
The received frame exceeded the byte count in RBC11…RBC0.

**RBC11..8** Receive Byte Count high.
Together with RBCL (bits RBC7…RBC0) the length of the received frame can be determined.

### 4.7.22 Transmit Byte Count Low (XBCL)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $2A_H/6A_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $54_H/D4_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| XBC7 | XBC6 | XBC5 | XBC4 | XBC3 | XBC2 | XBC1 | XBC0 |

**XBC7..0** Together with XBCH (bits XBC11…XBC8) this register is used in DMA-mode to program the length of the next frame to be transmitted (1…4096 bytes). The number of transmitted bytes is XBC + 1.
Consequently the SACCO can request the correct number of DMA-cycles after a XDD/XTF- or XDD-command.

### 4.7.23 Transmit Byte Count High (XBCH)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $2D_H/6D_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $5A_H/DA_H$
Reset value: 0000xxxx

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| DMA | 0 | 0 | XC | XBC11 | XBC10 | XBC9 | XBC8 |

**DMA** DMA-mode.
Selects the data transfer mode between the SACCO FIFOs and the system memory:
0…interrupt controlled data transfer (interrupt mode).
1…DMA controlled data transfer (DMA-mode).

**XC** Transmit Continuously.
When XC is set the SACCO continuously requests for transmit data ignoring the transmit byte count programmed in register XBCH and XBCL.

*Note: Only valid in DMA-mode.*

**XBC11..8** Transmit Byte Count high.
Together with XBC7…XBC0 the length of the next frame to be transmitted in DMA-mode is determined (1…4096 bytes).

### 4.7.24 Time Slot Assignment Register Transmit (TSAX)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $30_H/70_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $60_H/E0_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| TSNX5 | TSNX4 | TSNX3 | TSNX2 | TSNX1 | TSNX0 | XCS2 | XCS1 |

**TSNX5..0** Time Slot Number Transmit.
Selects one of up to 64 time slots ($00_H$ - $3F_H$) in which data is transmitted in clock mode 2. The number of bits per time slot is programmable in register XCCR.

**XCS2..1** Transmit Clock Shift bit2-1.
Together with XCS0 in register CCR2 the transmit clock shift can be adjusted in clock mode 2.

### 4.7.25 Time Slot Assignment Register Receive (TSAR)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $31_H/71_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $62_H/E2_H$
Reset value: $xx_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| TSNR5 | TSNR4 | TSNR3 | TSNR2 | TSNR1 | TSNR0 | RCS2 | RCS1 |

**TSNR5..0** Time Slot Number Receive.
Selects one of up to 64 time slots ($00_H$ - $3F_H$) in which data is received in clock mode 2. The number of bits per time slot is programmable in register RCCR.

**RCS2..1** Receive Clock Shift bit2-1.
Together with RCS0 in register CCR2 the transmit clock shift can be adjusted in clock mode 2.

### 4.7.26 Transmit Channel Capacity Register (XCCR)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $32_H/72_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $64_H/E4_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| XBC7 | XBC6 | XBC5 | XBC4 | XBC3 | XBC2 | XBC1 | XBC0 |

**XBC7..0** Transmit Bit Count.
Defines the number of bits to be transmitted in a time slot in clock mode 2 (number of bits per time slot = XBC + 1 (1…256 bits/time slot)).

*Note: In extended transparent mode the width of the time slot has to be n × 8 bits.*

### 4.7.27 Receive Channel Capacity Register (RCCR)

Access in demultiplexed µP-interface mode: write address: (Ch-A/Ch-B): $33_H/73_H$
Access in multiplexed µP-interface mode: write address: (Ch-A/Ch-B): $66_H/E6_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| RBC7 | RBC6 | RBC5 | RBC4 | RBC3 | RBC2 | RBC1 | RBC0 |

**RBC7..0** Receive Bit Count.
Defines the number of bits to be received in a time slot in clock mode 2.
Number of bits per time slot = RBC + 1 (1…256 bits/time slot).

*Note: In extended transparent mode the width of the time slot has to be n × 8 bits.*

### 4.7.28 Version Status Register (VSTR)

Access in demultiplexed µP-interface mode: read address: (Ch-A/Ch-B): $2E_H$
Access in multiplexed µP-interface mode: read address: (Ch-A/Ch-B): $5C_H$
Reset value: $80_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | VN3 | VN2 | VN1 | VN0 |

**VN3..0** SACCO Version Number.
$80_H$…version A1.
$81_H$...version A2 (ELIC V1.3).

## 4.8 D-Channel Arbiter

### 4.8.1 Arbiter Mode Register (AMO)

Access in demultiplexed µP-interface mode: read/write address: $60_H$
Access in multiplexed µP-interface mode: read/write address: $C0_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| FCC4 | FCC3 | FCC2 | FCC1 | FCC0 | SCA | CCHH | CCHM |

**FCC4..0** Full selection Counter.
The value (FCC4..0 + 1) defines the number of IOM-frames before the arbiter state machine changes from the state "limited selection" to the state "full selection", if the ASM does not detect any ’0’ on the remaining serial input lines (D-channels).
E.g. max. delay = 9 frames ⇒ AMO:FCC4..0 = 01000.

*Note: To avoid arbiter locking, either*
*a) the state limited selection can be skipped by setting FCC4..0 = $00_H$, or*
*b) the FCC4..0 value must be greater than the value described in **chapter 2.2.8.3**.*

**SCA** Suspend Counter Activation.
0…the suspend counter controls the arbiter state machine.
1…the suspend counter is disabled (e.g. for control by µP).

**CCHH** Control Channel Handling.
The control channel takes place:
0…in the C/I channel
1…in the MR bit (Monitor channel receive bit)

**CCHM** Control Channel Master activation.

0…disables the control channel master.
When disabled, all channels enabled in the DCE0-3 registers are sent the "available" information even when the SACCO-A is currently not available.

1…enables the control channel master.
During reception of D-channel data from a channel which has been enabled in the DCE0-3 registers all other enabled channels are sent the "blocked" information from the Control Memory (CM).

*Note: The D-channel arbiter can only be operated with framing control modes 3, 6 and 7.*

### 4.8.2 Arbiter State Register (ASTATE)

Access in demultiplexed µP-interface mode: read address: $61_H$
Access in multiplexed µP-interface mode: read address: $C2_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| AS2 | AS1 | AS0 | PAD1 | PAD0 | CHAD2 | CHAD1 | CHAD0 |

**AS2..0** Arbiter (receive channel selector) State:
000 : suspended
100 : full selection
011 : limited selection
001 : expect frame
010 : receive frame

**PAD1..0** Port Address.
The related frame was received on IOM-port PAD1..0

**CHAD2..0** Channel Address.
The related frame was received in IOM-channel CHAD2..0.

### 4.8.3 Suspend Counter Value Register (SCV)

Access in demultiplexed µP-interface mode: read/write address: $62_H$
Access in multiplexed µP-interface mode: read/write address: $C4_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| SCV7 | SCV6 | SCV5 | SCV4 | SCV3 | SCV2 | SCV1 | SCV0 |

**SCV7..0** Suspend Counter Value.
The value (SCV7..0 + 1) $\times$ 32 defines the number of D-bits which are analyzed in the state "expect frame" before the arbiter enters the state suspended state and an interrupt is issued.
Min.: 32 $\times$ D-bits (16 frames), max: 8192 D-bits.

### 4.8.4 D-Channel Enable Register IOM-Port 0 (DCE0)

Access in demultiplexed µP-interface mode: read/write address: $63_H$
Access in multiplexed µP-interface mode: read/write address: $C6_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| DCE07 | DCE06 | DCE05 | DCE04 | DCE03 | DCE02 | DCE01 | DCE00 |

### 4.8.5 D-Channel Enable Register IOM-Port 1 (DCE1)

Access in demultiplexed µP-interface mode: read/write address: $64_H$
Access in multiplexed µP-interface mode: read/write address: $C8_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| DCE17 | DCE16 | DCE15 | DCE14 | DCE13 | DCE12 | DCE11 | DCE10 |

### 4.8.6 D-Channel Enable Register IOM-Port 2 (DCE2)

Access in demultiplexed µP-interface mode: read/write address: $65_H$
Access in multiplexed µP-interface mode: read/write address: $CA_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| DCE27 | DCE26 | DCE25 | DCE24 | DCE23 | DCE22 | DCE21 | DCE20 |

### 4.8.7 D-Channel Enable Register IOM-Port 3 (DCE3)

Access in demultiplexed µP-interface mode: read/write address: $66_H$
Access in multiplexed µP-interface mode: read/write address: $CC_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| DCE37 | DCE36 | DCE35 | DCE34 | DCE33 | DCE32 | DCE31 | DCE30 |

**DCEn7..0** D-Channel Enable bits channel 7-0, IOM-port n.

0…D-channel i on IOM-port n is disabled for data reception. The control channel of a disabled D-channel is not manipulated by the control channel master. It passes the value stored in the EPIC-1 control memory (C/I or MR must = "blocked"). The disabling of a D-channel has an immediate effect also when the channel is active. In this case the transmitter (HDLC-controller in the subscriber terminal) is forced to abort the current frame.

1…D-channel i on IOM-port n is enabled for data reception.
The control channel of an enabled D-channel is manipulated
a) by the control channel master, if AMO:CCHM = 1,
b) directly via DCE, if AMO:CCHM = 0.

### 4.8.8 Transmit D-Channel Address Register (XDC)

Access in demultiplexed µP-interface mode: read/write address: $67_H$
Access in multiplexed µP-interface mode: read/write address: $CE_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | BCT | PAD1 | PAD0 | CHAD2 | CHAD1 | CHAD0 |

**BCT** Broadcast Transmission, BCT = 1 enables broadcast transmission. The transmitted frame is send to all channels enabled in the registers BCG0-3.

**PAD1..0** Port address, defines the transmit IOM-port when BCT = 0.

**CHAD2..0** Channel Address, defines the transmit IOM-channel when BCT = 0.

### 4.8.9 Broadcast Group IOM-port 0 (BCG0)

Access in demultiplexed µP-interface mode: read/write address: $68_H$
Access in multiplexed mP-interface mode: read/write address: $D0_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| BCE07 | BCE06 | BCE05 | BCE04 | BCE03 | BCE02 | BCE01 | BCE00 |

### 4.8.10 Broadcast Group IOM-port 1 (BCG1)

Access in demultiplexed µP-interface mode: read/write address: $69_H$
Access in multiplexed µP-interface mode: read/write address: $D2_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| BCE17 | BCE16 | BCE15 | BCE14 | BCE13 | BCE12 | BCE11 | BCE10 |

### 4.8.11 Broadcast Group IOM-port 2 (BCG2)

Access in demultiplexed µP-interface mode: read/write address: $6A_H$
Access in multiplexed µP-interface mode: read/write address: $D4_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| BCE27 | BCE26 | BCE25 | BCE24 | BCE23 | BCE22 | BCE21 | BCE20 |

### 4.8.12 Broadcast Group IOM-port 3 (BCG3)

Access in demultiplexed µP-interface mode: read/write address: $6B_H$
Access in multiplexed µP-interface mode: read/write address: $D6_H$
Reset value: $00_H$

| bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| BCE37 | BCE36 | BCE35 | BCE34 | BCE33 | BCE32 | BCE31 | BCE30 |

**BCEn7..0** Broadcast Enable bit channel 7-0, IOM-port n.

BCEni: 0… D-channel i, IOM-port n is disabled for broadcast transmission.

1… D-channel i, IOM-port n is enabled for broadcast transmission.

# 5 Application Hints

## 5.1 Introduction

### 5.1.1 IOM® and SLD Functions

**IOM® (ISDN Oriented Modular) Interface**

The IOM-2 standard defines an industry standard serial bus for interconnecting telecommunications ICs. The standard covers line card, NT1, and terminal architectures for ISDN, DECT and analog loop applications. The IOM-2 standard is a derivative of the IOM-1 interface formerly designed by Siemens to interconnect layer-1 and layer-2 devices within ISDN terminals and on digital line cards.

The **IOM®-1 interface** provides a symmetrical full-duplex communication link, containing user data, control/programming, and status channels for 1 ISDN subscriber, i.e. it provides capacity for 2 B channels at 64 kBit/s and 1 D channel at 16 kBit/s. The IOM-1 channel consists of four 8 bit timeslots which are serially transferred within an 8 kHz frame. The first 2 timeslots carry the B1 and B2 channels, the third timeslot carries an 8 bit monitor channel and the fourth timeslot carries the 2 bit D channel, a 4 bit Command/Indication (C/I) channel plus 2 additional control bits (T and E bits). The monitor channel serves to exchange control and status information in a message oriented fashion of one byte per message. The C/I channel carries real-time status information between the line transceiver and the layer-2 device or the line card controller. Status information transmitted over the C/I channel is “static” in the sense that the 4 bit word is repeatedly transmitted, every frame, as long as the status condition that it indicates is valid. The T bit is used by some U layer-1 devices as a transparent channel. The E bit is used in conjunction with the monitor channel to indicate the transfer of a monitor byte to the slave device. The various channels are time-multiplexed over a four wire serial interface. The data transfer rate at the IOM-1 interface is 256 kBit/s, the data is clocked with a double rate clock of 512 kHz (DCL) and the frame is synchronized by an 8 kHz framing signal (FSC).

Because the IOM-1 interface structure can handle only 1 ISDN channel, which is too little for line card applications, the **multiplexed IOM®-1 bus** was developed. It multiplexes 8 individual IOM-1 channels into the 8 kHz frame. The data transfer rate is now increased to 2048 kBit/s, the data is clocked with a double rate clock of 4096 kHz (DCL) and the frame is synchronized with an 8 kHz framing signal (FSC). The bit timing and FSC position differs slightly from the 256 kBit/s IOM-1 interface. The IOM channel structure however is identical to the non-multiplexed IOM-1 case.

The **IOM®-2 bus standard** is an enhancement of both the IOM-1 and multiplexed IOM-1 standards. Both the line card and terminal portions of the IOM-2 standard utilize the same basic frame and clocking structure, but differ in the number and usage of the individual channels. Data is clocked by a data clock (DCL) that operates at twice the data

rate. Frames are delimited by an 8 kHz frame synchronization signal (FSC). The bit timing and FSC position is identical to the non-multiplexed IOM-1 case.

**The line card version of the IOM®-2** provides a connection path between line transceivers (ISDN) or codecs (analog), and the line card controller, the EPIC or ELIC; the line card controller provides the connection to the switch backbone. The IOM-2 bus time-multiplexes data, control, and status information for up to 8 ISDN transceivers or up to 16 codec/filters over a single full-duplex interface.

**Figure 58** shows the IOM-2 frame structure for the line card. It consists of 8 individual and independent IOM channels, each having a structure similar to the IOM-1 channel structure. The main difference compared to IOM-1 is the more powerful monitor channel performance. Monitor messages of unlimited length can now be transferred at a variable speed, controlled by a handshake procedure using the MR and MX bits. The C/I channel can have a width of 4 bits for ISDN applications or of 6 bits for analog signaling applications.

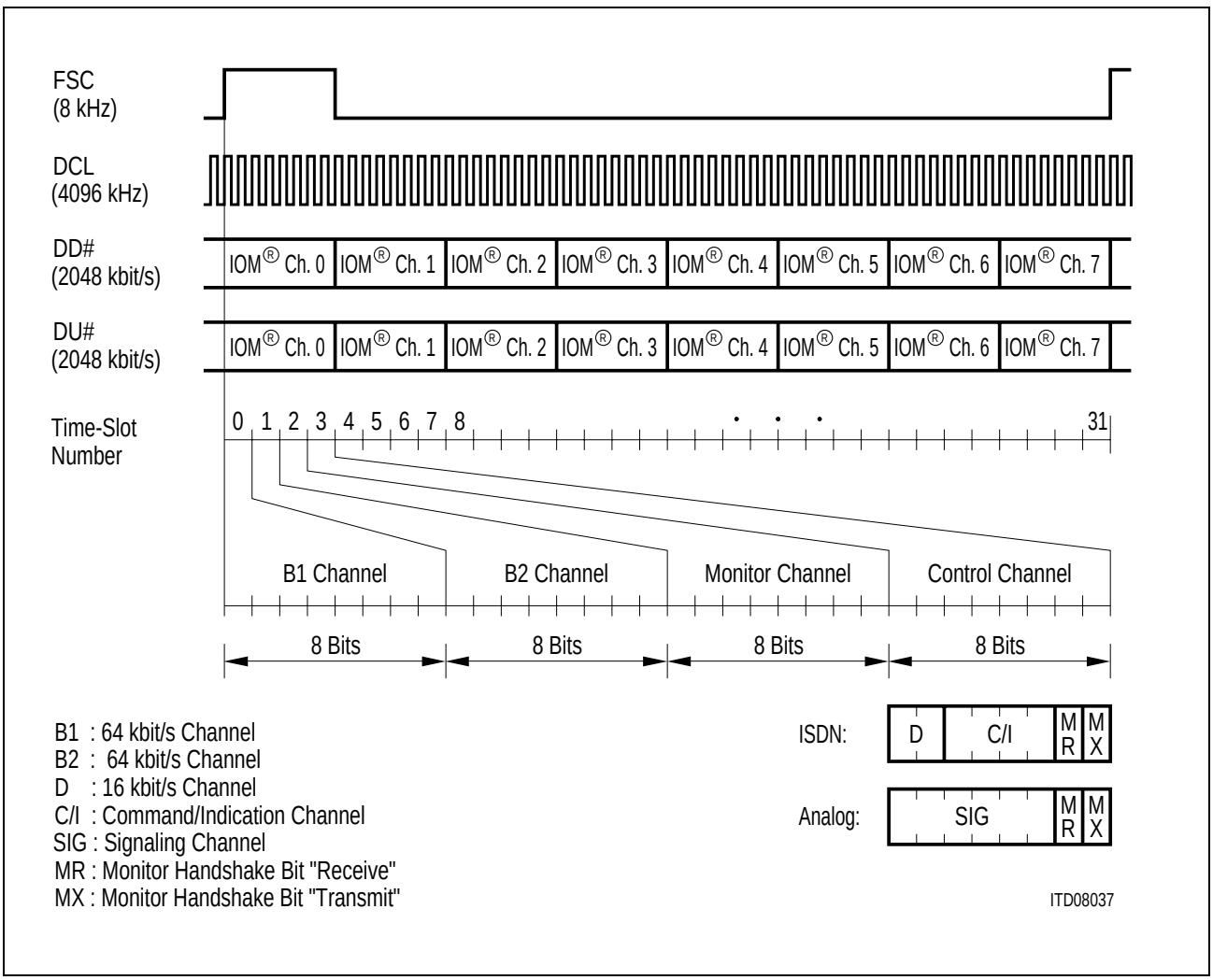



**Figure 58**
**IOM®-2 Frame Structure for Line Card Applications**

The **terminal version of the IOM®-2** is a variation of the line card bus, designed for ISDN terminal and NT1 applications. It consists of three IOM channels, each containing four 8 bit timeslots. The resultant data transfer rate is therefore 768 kBit/s and the data is clocked with a 1536 kHz double rate clock (DCL). The IOM channel structure is similar to the line card case. The first channel is dedicated for controlling the layer-1 transceiver (monitor and C/I channels) and passing the user data (B and D channels) to the layer-1 transceiver. The second and third channels are used for communication between a controlling device and devices other than the layer-1 transceiver, or for transferring user data between data processing devices (IC channels). The C/I channel of the third IOM channel is used for TIC bus applications (D and C/I channel arbitration). The TIC bus allows multiple layer-2 devices to individually gain access to the D and C/I channels located in the first IOM channel.

Finally, for NT1 applications, it is also possible to operate the IOM-2 interface at a data rate of 256 kBit/s (1 IOM channel). This is sufficient for the simple back to back connection of layer-1 transceivers in Network Terminator (NT) and Repeater (RP) applications.

The following table summarizes the different operation modes and applications of the IOM-1 and IOM-2 standards (TE = Terminal Equipment, NT = Network Terminator, LT = Line Terminator):

**Table 22**
**Overview of IOM® Applications and Data Rates**

| Mode | Applications | Data Rate / Clock Rate |
|---|---|---|
| IOM-1 | TE, NT, LT | 256 kBit/s / 512 kHz |
| Multiplexed IOM-1 | LT | 2048 kBit/s / 4096 kHz |
| IOM-2 | LT | 2048 kBit/s / 4096 kHz |
| IOM-2 | TE, NT | 768 kBit/s / 1536 kHz |
| IOM-2 | NT | 256 kBit/s / 512 kHz |

The main application of the ELIC is on digital and analog line cards. The ELIC is therefore primarily designed to support the line card modes (2048 kBit/s) of the IOM-2 standard. It can however be programmed to support all the above mentioned IOM data rates and C/I and monitor processing schemes. However, it must be assured that the desired PCM to IOM data rate ratio is feasible (refer to **chapter 5.2.2.3**).

### SLD (Subscriber Line Data) Interface

The SLD bus is used by the ELIC to interface with the subscriber line devices. A Serial Interface Port (SIP) is used for the transfer of all digital voice and data, feature control and signaling information between the individual subscriber line devices, the PCM highways and the control backplane. The SLD approach provides a common interface for one analog or digital component per line. The ELIC switches the PCM data transparently switched onto the PCM highways.

There are three wires connecting each subscriber line device and the ELIC: two common clock signals shared among all devices, and a unique bidirectional data wire for each of the eight SIP ports. The direction signal (FSC) is an 8 kHz clock output from the ELIC (master) that serves as a frame synch to the subscriber line devices (slave) as well as a transfer indicator. The data is transferred at a 512 kHz data rate, clocked by the subscriber clock (DCL). When FSC is high (first half of the 125 µs SLD frame), four bytes of digital data are transmitted on the SLD bus from the ELIC to the slave (downstream direction). During the second half of the frame when FSC is low, four bytes of data are transferred from the slave back to the ELIC (upstream direction).

Channel B1 and B2 are 64 kBit/s channels reserved for voice and data to be routed to and from the PCM highways. The third and seventh byte are used to transmit and receive control information for programming the slave devices (feature control channel). The last byte in each direction is reserved for signaling data.

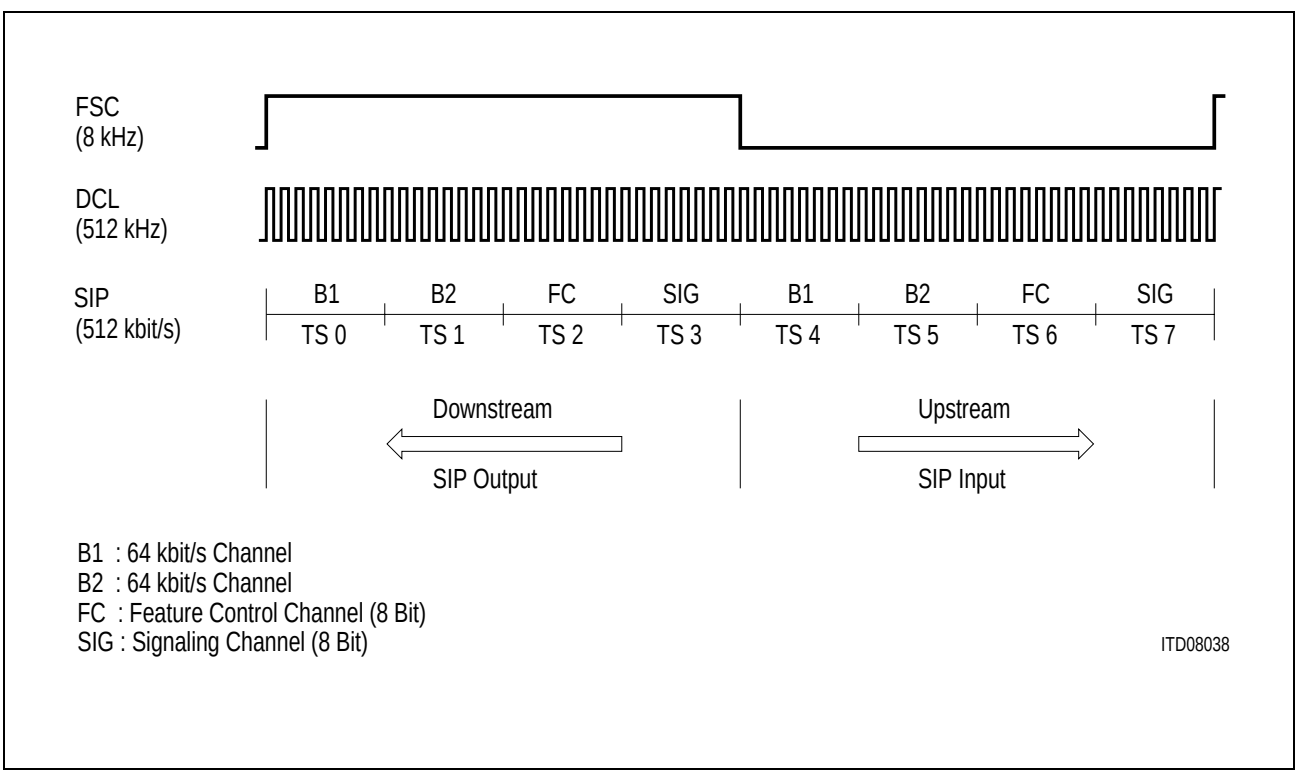



**Figure 59**
**SLD Frame Structure**

In contrast to other Siemens telecom devices, the ELIC does not provide an ‘IOM mode’ or an ‘SLD mode’ that can be selected by programming a single ‘mode bit’. Instead, the ELIC provides a configurable interface (CFI) that can be configured for a great variety of interfaces, including IOM-1, multiplexed IOM-1, IOM-2 and SLD interfaces.

**The Characteristics of the Different IOM® and SLD Interfaces can be Divided into Two Groups**

- Timing characteristics and
- Handling of special channels (C/I or signaling channel, monitor or feature control channel)

The timing characteristics (data rate, clock rate, bit timing, etc. … ) are programmed in the CFI registers (see **chapter 5.2.2.2**). The CFI data rate, for example, can be selected between 128 kBit/s and 8192 kBit/s. This covers the standard IOM and SLD data rates of 256, 512, 768 and 2048 kBit/s.

The special channels are initialized on a per timeslot basis in the control memory (CM). This programming on a per timeslot basis allows a dedicated usage of each CFI port and timeslot: an application that requires only two IOM-2 compatible layer-1 transceivers will also only occupy 8 CFI timeslots (2 IOM channels) for that purpose. The remaining 24 timeslots can then be used for general switching applications or for the connection of non IOM-2 compatible devices that require a special µP handling.

**The Special Channels can be Divided into Two Groups**

- Monitor/Feature Control channels and
- Control/Signaling channels

The **Monitor/Feature Control handler** can be adjusted to operate according to the

– IOM-1 protocol (up to 1 byte, no handshake), the
– IOM-2 protocol (any number of bytes, handshake using the MR and MX bits) and to the
– SLD protocol (up to 16 bytes in subsequent frames without handshake)

The Monitor/Feature Control handler is a dedicated unit that communicates only with one IOM or SLD channel at a time. An address register selects one out of 64 possible MF channels. A 16 byte bidirectional FIFO (MFFIFO) provides intermediate storage for the data to be sent or received. The message transfer over the MF channel is always half-duplex, i.e. data can either be sent at a time or received at a time. It should be noted that if the IOM-2 protocol is selected, the actual message length i.e. the number of bytes to be sent or received is unlimited and is not restricted by the MFFIFO size!

If non handshake protocols (IOM-1 and SLD) are used, the ELIC must always be the master of the MF communication. Example: the ELIC programs and reads back the coefficients of a SICOFI (PEB 2060) device.

If the handshake protocol is used (IOM-2), a balanced MF communication is also possible: since the MF handler cannot be pointed to all IOM-2 channels at the same time, the ELIC has implemented a search function that looks for active monitor transmit handshake (MX) bits on all upstream IOM-2 channels. If an active channel is found, the address is stored and an interrupt is generated. The MF handler can then be pointed to that particular channel and the message transfer can take place.
Example: the ELIC reads an EOC message out of an IECQ (PEB 2091) device.

The **Control/Signaling handler** can be adjusted to handle the following types of channels:

- 4 bit C/I channel (IOM-1 and digital IOM-2)
- 6 bit C/I or Signaling channel (analog IOM-2)
- 8 bit Signaling channel (SLD)

In **downstream direction**, the µP can write the 4, 6 or 8 bit C/I or Signaling value to be transmitted directly to the CFI timeslot i.e. to the control memory. This value will then be transmitted repeatedly in each frame until a new value is loaded.

If the 4 bit C/I channel option is selected, the two D channel bits can either be tristated by the ELIC (decentral D channel handling scheme) or they can be switched transparently from any 2 bit sub-timeslot position at the PCM interface (central D channel handling scheme).

In **upstream direction**, the µP can read the received 4, 6, or 8 bit C/I or Signaling value directly from the CFI timeslot i.e. from the control memory. In addition the Control/ Signaling handler checks all received C/I and Signaling channels for changes. Upon a change:

– an interrupt is generated,
– the address of the involved CFI timeslot is stored in a 9 byte FIFO (CIFIFO) and
– the new value is stored in the control memory.

The CIFIFO serves to buffer the address information in order to increase the µP latency time.

The change detection mechanism is based on a single last look procedure for 4 bit C/I channels and on a double last look procedure for 6 and 8 bit C/I or Signaling channels. The single last look period is fixed to 125 µs, whereas the double last look period is programmable from 125 µs to 32 ms. The last look period is programmed using the ELIC timer.

With the single last look procedure, each C/I value change immediately leads to a valid change and thus to an interrupt.

With the double last look procedure, a C/I or Signaling value change must be detected two times at the sampling points of the last look interval before a valid change is recognized and an interrupt is generated.

If the 4 bit C/I channel option is selected, the two D channel bits can either be ignored by the ELIC (decentral D channel handling scheme) or they can be switched transparently to any 2 bit sub-timeslot position at the PCM interface (central D channel handling scheme).

## 5.2 Configuration of Interfaces

### 5.2.1 PCM Interface Configuration

#### 5.2.1.1 PCM Interface Signals

The PCM interface signals are summarized in **table 23.**

**Table 23**
**Signals at the PCM Interface**

| Pin No. | Symbol | I: Input O: Output | Function |
|---|---|---|---|
| 63<br>65<br>67<br>69 | TxD0<br>TxD1<br>TxD2<br>TxD3 | O<br>O<br>O<br>O | Transmit PCM interface data: serial data is sent at standard TTL or CMOS levels (tristate drivers). These pins can be set to high impedance with a 2 bit resolution. |
| 62<br>64<br>66<br>68 | $\overline{\text{TSC0}}$<br>$\overline{\text{TSC1}}$<br>$\overline{\text{TSC2}}$<br>$\overline{\text{TSC3}}$ | O<br>O<br>O<br>O | Tristate control signals for the PCM transmit lines. These signals are low when the corresponding TxD# outputs are valid. |
| 61<br>60<br>59<br>58 | RxD0<br>RxD1<br>RxD2<br>RxD3 | I<br>I<br>I<br>I | Receive PCM interface data: serial data is received at standard TTL or CMOS levels. |
| 70 | PFS | I | PCM interface frame synchronization signal. |
| 71 | PDC | I | PCM interface data clock, single or double rate. |

#### 5.2.1.2 PCM Interface Registers

The characteristics at the PCM interface (timing, modes of operation, etc. … ) are programmed in the 4 PCM interface registers and in the Operation Mode Register OMDR. The function of each bit is described in **chapter 5.2.1.3**. For addresses, refer to **chapter 4.1**.

**PCM Mode Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| PMOD | PMD1 | PMD0 | PCR | PSM | AIS1 | AIS0 | AIC1 | AIC0 |

**PCM Bit Number Register** read/write reset value: $FF_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| PBNR | BNF7 | BNF6 | BNF5 | BNF4 | BNF3 | BNF2 | BNF1 | BNF0 |

**PCM Offset Downstream Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| POFD | OFD9 | OFD8 | OFD7 | OFD6 | OFD5 | OFD4 | OFD3 | OFD2 |

**PCM Offset Upstream Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| POFU | OFU9 | OFU8 | OFU7 | OFU6 | OFU5 | OFU4 | OFU3 | OFU2 |

**PCM Clock Shift Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| PCSR | DRCS | OFD1 | OFD0 | DRE | ADSRO | OFU1 | OFU0 | URE |

**Operation Mode Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| OMDR | OMS1 | OMS0 | **PSB** | **PTL** | COS | MFPS | CSB | RBS |

### 5.2.1.3 PCM Interface Characteristics

In the following the PCM interface characteristics that can be programmed in the PCM interface registers are explained in more detail.

#### PCM Mode PMOD: PMD1, PMD0

The PCM mode primarily defines the actual number of PCM highways that can be used for switching purposes (logical ports). 1, 2, or 4 logical PCM ports can be selected. Since

the channel capacity of the ELIC is constant (128 channels per direction), the PCM mode also influences the maximum possible data rate. In each PCM mode a minimum data rate as well as a minimum data rate stepping are specified.

It should also be noticed that there are some restrictions concerning the PCM to CFI data rate ratio which may affect some applications. These restrictions are described in **chapter 5.2.2.3**.

The table below summarizes the specific characteristics of each PCM mode (DR = PCM data rate):

**Table 24**
**Operation Modes at the PCM Interface**

| PMD1 | PMD0 | PCM Mode | Number (Label) of Logical Ports | Data Rate [kBit/s] | | Data Rate Stepping [kBit/s] | PDC Frequency (Clock Rate) |
|---|---|---|---|---|---|---|---|
| | | | | min. | max. | | |
| 0 | 0 | 0 | 4 (0 … 3) | 256 | 2048 | 256 | DR, 2 × DR |
| 0 | 1 | 1 | 2 (0 … 1) | 512 | 4096 | 512 | DR, 2 × DR |
| 1 | 0 | 2 | 1 | 1024 | 8192 | 1024 | DR |
| 1 | 1 | 3 | 2 (0 … 1) | 512 | 4096 | 512 | DR, 2 × DR |

*Note: The label is used to specify a PCM port (logical port) when programming a switching function. It should not be confused with the physical port number which refers to actual hardware pins. The relationship between logical and physical port numbers is given in **table 30** and is illustrated in **figure 64**.*

### PCM Clock Rate PMOD:PCR

The PCM interface is clocked via the PDC pin. If PCR is set to logical 0, the PDC frequency must be identical to the selected data rate (single clock operation). If PCR is set to logical 1, the PDC frequency must be twice the selected data rate (double clock operation). Note that in PCM mode 2, only single clock rate operation is allowed.

In PCM mode 0 for example, PCR can be set to 1 to operate at up to four 2048 kBit/s PCM highways with a PCM clock of 4096 kHz.

### PCM Bit Number PBNR:BNF7 … BNF0

The PCM data rate is determined by the clock frequency applied to the PDC pin and the clock rate selected by PMOD:PCR. The number of bits which constitute a PCM frame can be derived from this data rate by dividing by 8000 (8 kHz frame structure).

If the PCM interface is for example operated at 2048 kBit/s, the frame would consist of 256 bits or 32 timeslots.

*Note: There is a mode dependent restriction on the possible number of bits per frame BPF:*

**Table 25**

| PCM Mode | Possible Values for BNF |
|---|---|
| 0 | BPF must be modulo 32 |
| 1, 3 | BPF must be modulo 64 |
| 2 | BPF must be modulo 128 |

Also refer to **table 25**.

This number of bits must be programmed to PBNR:BNF7 … 0 as indicated in **table 26**.

**Table 26**
**Formulas to Calculate the PBNR Value**

| PCM Mode | PBNR:BNF7 … 0(Hex) |
|---|---|
| 0 | BPF7 … 0 = BPF – 1 |
| 1, 3 | BPF7 … 0 = (BPF – 2)/2 |
| 2 | BPF7 … 0 = (BPF – 4)/4 |

The externally applied frame synchronization pulse PFS resets the internal PCM timeslot and bit counters. The value programmed to PBNR is internally used to reset the PCM timeslot and bit counters so that these counters always count modulo the actual number of bits per frame even in the absence of the external PFS pulse. Additionally, the PFS period is internally checked against the PBNR value. The result of this comparison is displayed in the PCM Synchronization Status bit (STAR:PSS). Also, refer to **chapter 5.8.3**.

**Examples**

In PCM mode 0 a PCM frame consisting of 32 timeslots would require a setting of PBNR $= 32 \times 8 - 1 = 255_D = FF_H$.

In PCM mode 1 a PCM frame consisting of 24 timeslots would require a setting of PBNR $= (24 \times 8 - 2)/2 = 95_D = 5F_H$.

In PCM mode 2 a PCM frame consisting of 64 timeslots would require a setting of PBNR $= (64 \times 8 - 4)/4 = 127_D = 7F_H$.

### PCM Synchronization Mode PMOD:PSM

The PCM interface is synchronized via the PFS signal. A transition from low to high of PFS synchronizes the PCM frame. It should be noted that the rising PFS edge does not directly synchronize the frame, it is instead first internally sampled with the PDC clock:

If PSM is set to logical 0, the PFS signal is sampled with the falling clock edge of PDC, if it is set to logical 1, the PFS signal is sampled with the rising clock edge of PDC.

PSM should be selected such that the PDC signal detects stable low and high levels of the PFS signal, meeting the set-up ($T_{FS}$) and hold ($T_{FH}$) times with respect to the programmed PDC clock edge.

In other words, if for example the rising PFS edge has some jitter with respect to the rising PDC edge, the falling PDC edge should be taken for the evaluation.

The high phase of the PFS pulse may be of arbitrary length, however it must be assured that it is sampled low at least once before the next framing pulse.

The relationship between the PFS signal and the beginning of the PCM frame is given in **figure 60** and **figure 61**.

### PCM Bit Timing and Bit Shift POFD, POFU, PCSR

The position of the PCM frame can be shifted relative to the framing source PFS in increments of bits by programming the PCM offset bits OFD9 … 0, OFU9 … 0, DRCS, ADSRO in the POFD, POFU and PCSR. This shifting can be performed separately for up- and downstream directions and by up to a whole frame. Additionally, the polarity of the PDC clock edge used for transmitting and sampling the data can be selected with the URE and DRE bits in the PCSR register.

The timeslot structure on the PCM interface is synchronized with the externally applied PFS pulse. The rising edge of PFS, after it has been sampled by the PDC signal, marks the first bit of the PCM frame. This first bit is referenced to as the BND (Bit Number Downstream) of the downstream and the BNU (Bit Number Upstream) of the upstream frame.

If PCSR:URE is set to 1, data is transmitted with the rising edge of PDC, if URE is set to 0, data is transmitted with the next following falling edge of PDC.

If PCSR:DRE is set to 0, data is sampled with the falling edge of PDC, if DRE is set to 1, data is sampled with the next following rising edge of PDC.

The relationship between the PFS, PDC signals and the PCM bit stream on RxD# and TxD# is illustrated in **figure 60** and **figure 61**.

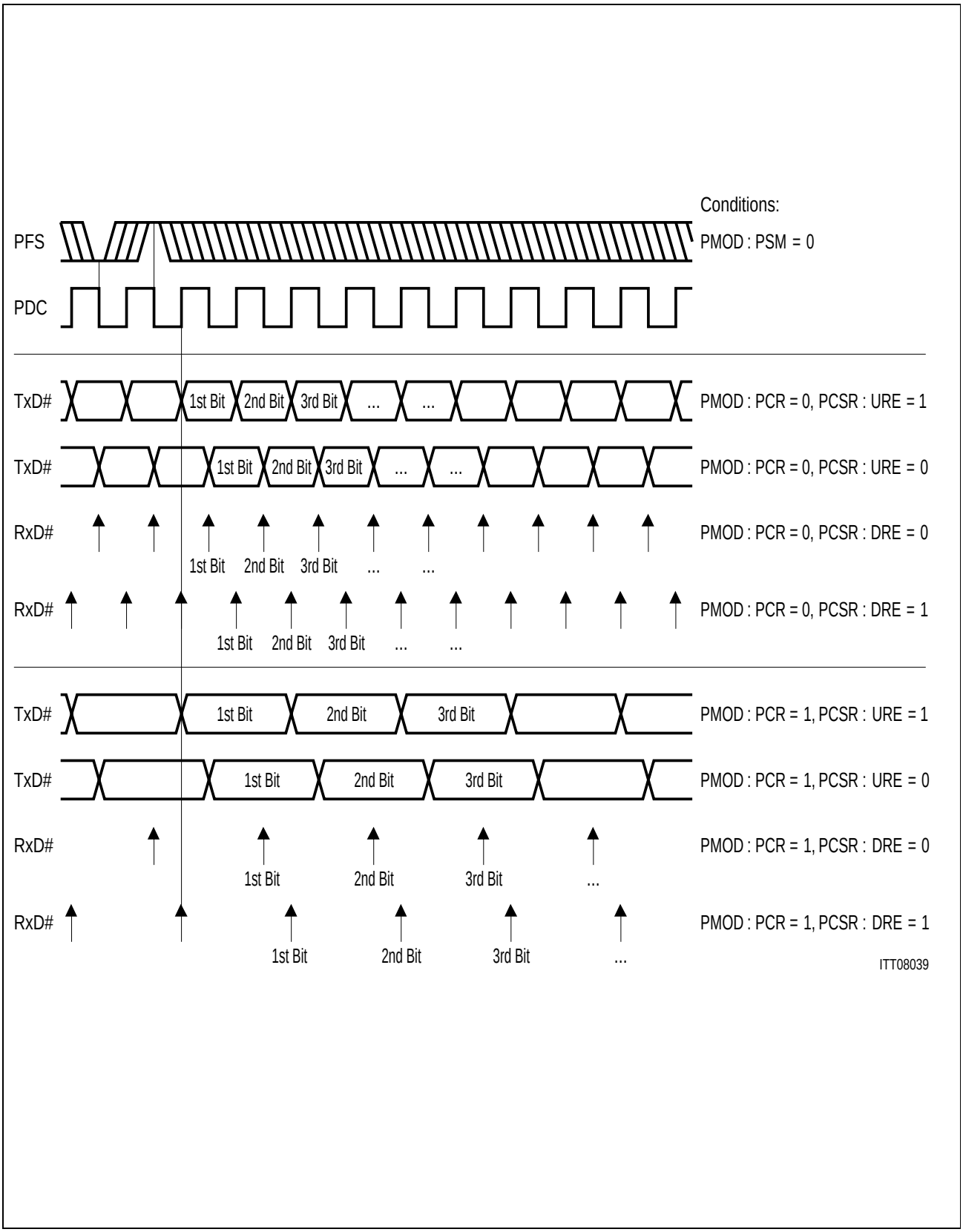



**Figure 60**
**PCM Interface Framing Offset for PMOD:PSM = 0**

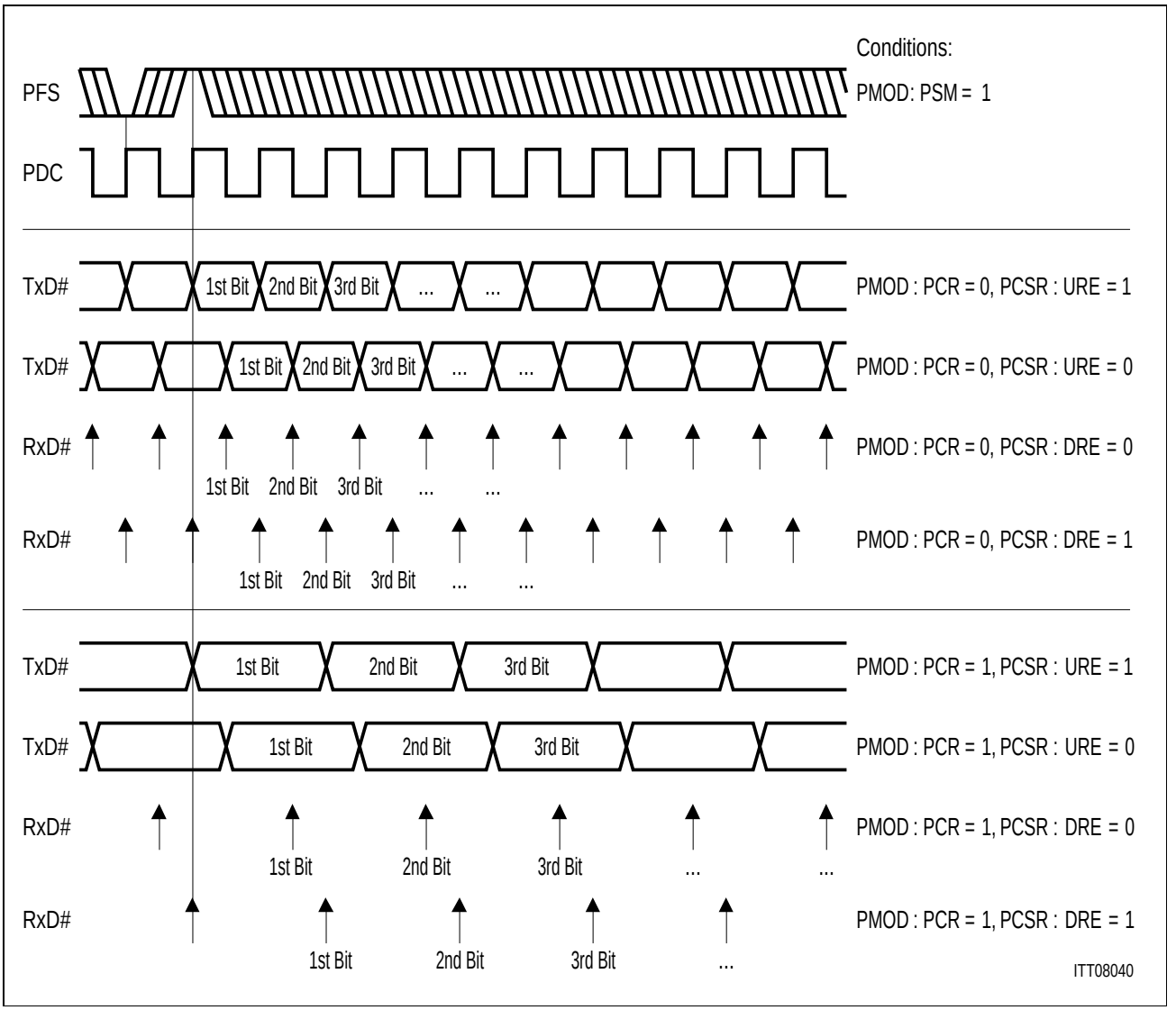



**Figure 61**
**PCM Interface Framing for PMOD:PSM = 1**

The formulas given in **table 27** and **table 28** apply for calculating the values to be programmed to the offset registers (OFD, OFU) given the desired bit number (BND, BNU) to be marked. BPF denotes the actual number of bits constituting a frame.

**Table 27**
**Formulas to Calculate the PCM Frame Offset Downstream (RxD#)**

| PCM Mode | Offset Downstream, POFD, PCSR | Remarks |
|---|---|---|
| 0 | OFD9 … 2 = $(BND - 17 + BPF)_{mod\ BPF}$ | PCSR:OFD1 … 0 = 0 |
| 1, 3 | OFD9 … 1 = $(BND - 33 + BPF)_{mod\ BPF}$ | PCSR:OFD0 = 0 |
| 2 | OFD9 … 0 = $(BND - 65 + BPF)_{mod\ BPF}$ | – |

**Table 28**
**Formulas to Calculate the PCM Frame Offset Upstream (TxD#)**

| PCM Mode | Offset Upstream, POFU, PCSR | Remarks |
|---|---|---|
| 0 | OFU9 … 2 = $(BNU + 23)_{mod\ BPF}$ | PCSR:OFU1 … 0 = 0 |
| 1, 3 | OFU9 … 1 = $(BNU + 47)_{mod\ BPF}$ | PCSR:OFU0 = 0 |
| 2 | OFU9 … 0 = $(BNU + 95)_{mod\ BPF}$ | – |

**Examples**

1) In PCM mode 0, with a frame consisting of 32 timeslots, the following timing relationship between the framing signal and the data signals is required:

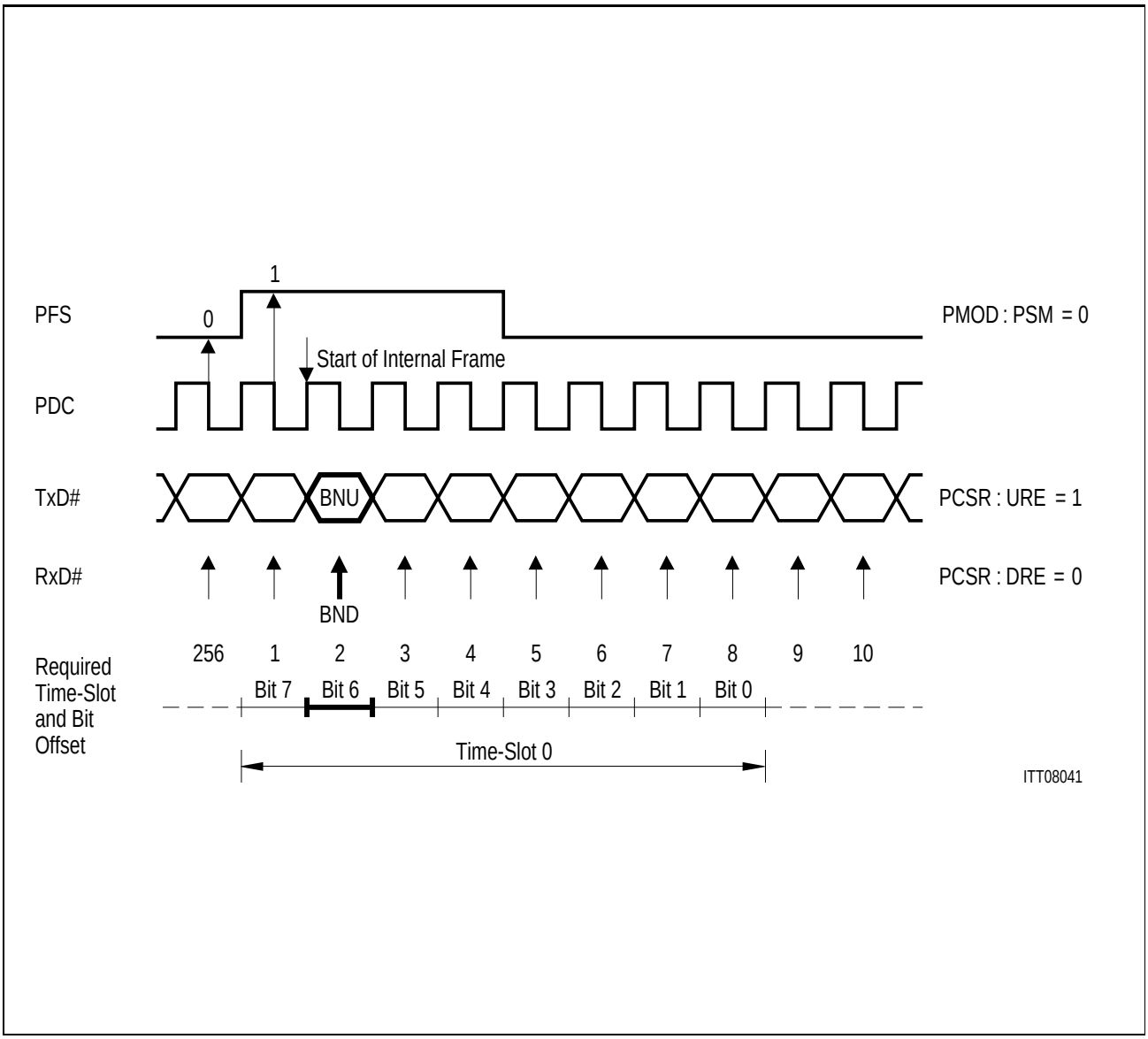



**Figure 62**
**Timing PCM Frame Offset for Example 1**

The PCM interface shall be clocked with a PDC having the same frequency as the data rate i.e. 2048 kHz. Since the rising edge of PFS occurs at the same time as the rising edge of PDC, it is recommended to select the falling PDC edge for sampling the PFS signal (PMOD:PSM0 = 0). In this case the 1st bit of internal framing structure (according to **figure 62**) will represent timeslot 0, bit 6 (2nd bit) of the external frame (according to **figure 60**). The values to be programmed to the POFD, POFD and PCSR can now be determined as follows:

With BND = BNU = 2 and BPF = 256:

$POFD = OFD9 \ldots 2 = (BND - 17 + BPF)_{mod\ BPF} = (2 - 17 + 256)_{mod\ 256} = 241_D = F1_H$
$POFU = OFU9 \ldots 2 = (BNU + 23)_{mod\ BPF} = (2 + 23)_{mod\ 256} = 25_D = 19_H$

With URE = 1 and DRE = 0:

$PCSR = 01_H$

2) In PCM mode 1, with a frame consisting of 48 timeslots, the following timing relationship between the framing signal and the data signals is required:

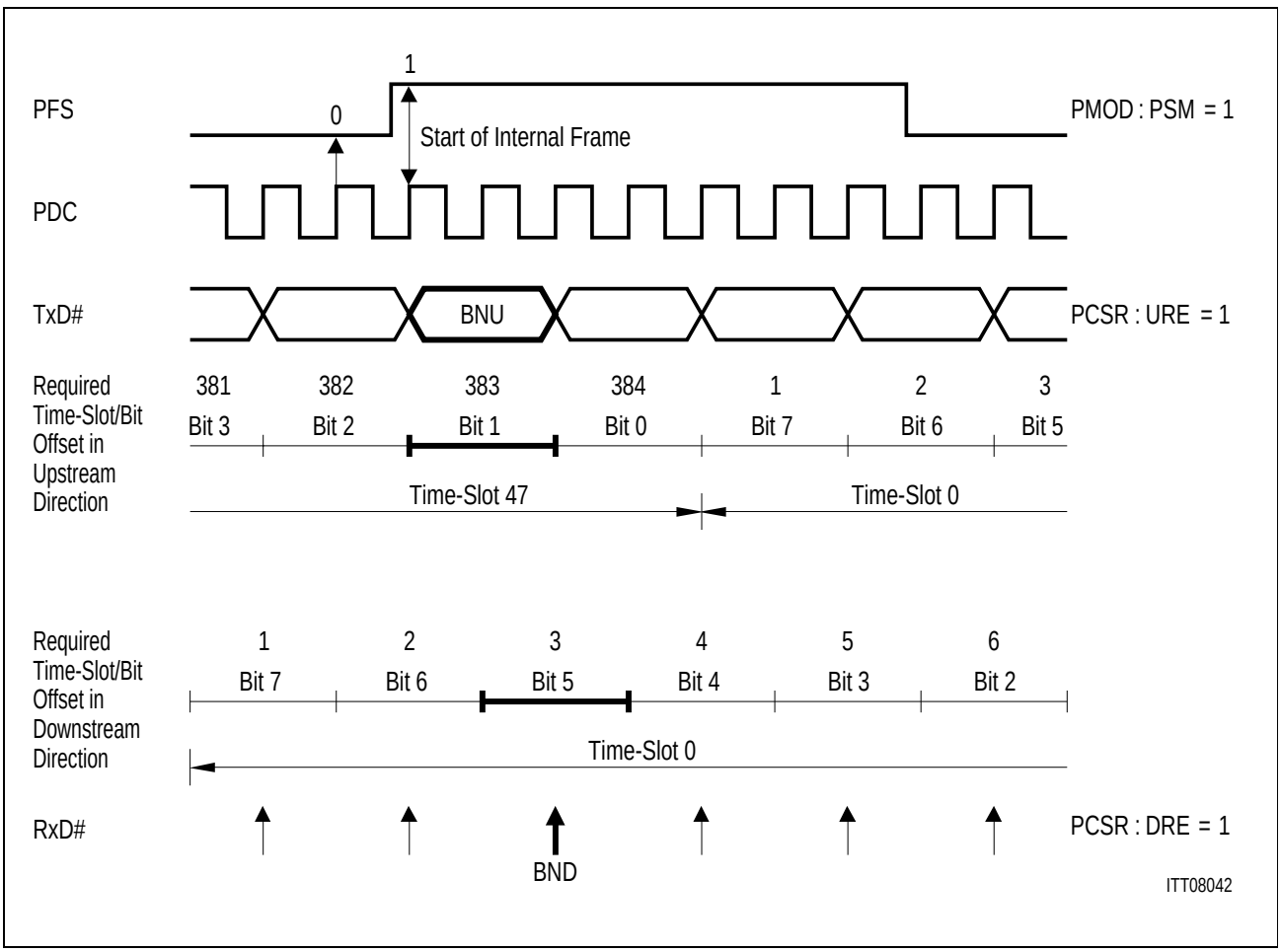



**Figure 63**
**Timing for PCM Frame Offset of Example 2**

The PCM interface shall be clocked with a PDC having twice the frequency of the data rate i.e. 6144 kHz. Since the rising edge of PFS occurs a little bit before the rising edge of PDC i.e. the set-up and hold times with respect to the rising PDC are met, it is possible to select the rising PDC edge for sampling the PFS signal (PMOD:PSM = 1). In this case the 1st bit of the internal framing structure (according to **figure 63**) will represent timeslot 47, bit 1 (383rd bit) in upstream and timeslot 0, bit 5 (3rd bit) in downstream direction of the external frame (according to **figure 61**). The values to be programmed to the POFD, POFD and PCSR can now be determined as follows:

With BND = 3, BNU = 383 and BPF = 384:

$OFD9 \ldots 1 = (BND - 33 + BPF)_{mod\ BPF} = (3 - 33 + 384)_{mod\ 384} = 354_D = 1\ 0110\ 0010_B$
$OFU9 \ldots 1 = (BNU + 47)_{mod\ BPF} = (383 + 47)_{mod\ 384} = 46_D = 0001\ 0111\ 0_B$

$POFD = 1011\ 0001_B = B1_H$,
$POFU = 1000\ 1111_B = 17_H$

With URE = 1 and DRE = 1:

$PCSR = 0001\ 0001_B = 11_H$,

### PCM Receive Line Selection PMOD:AIS1 … AIS0

The PCM transmit line of a given logical port (as it is used for programming the switching function) is always assigned to a dedicated physical transmit pin, e.g. in PCM mode 1, pin TxD2 carries the PCM data of logical port 1.

In receive direction however, an assignment between logical and physical ports can be made in PCM modes 1 and 2. This selection is programmed via the Alternative Input Selection bits 1 and 0 (AIS1, AIS0) in the PMOD register.

In PCM mode 0, AIS1 and AIS0 should both be set to 0.

In PCM mode 1, AIS0 selects between receive lines RxD0 and RxD1 for logical port 0 and AIS1 between the receive lines RxD2 and RxD3 for logical port 1.

In PCM mode 2, AIS1 selects between the receive lines RxD2 and RxD3, the setting of AIS0 is don’t care.

In PCM mode 3, AIS0 selects between receive lines RxD0 and RxD1 for logical port 0 and AIS1 between the receive lines RxD2 and RxD3 for logical port 1.

The state of the AIS# bits is furthermore put out via the $\overline{TSC\#}$ pins and can thus be used to control external circuits (drivers, relays … ).

**Table 29** shows the function taken on by each of the PCM interface pins, depending on the PCM mode and the values programmed to AIS1 and AIS0.

**Table 29**
**PCM Pin Configuration**

| PCM Mode | Port 0 | | | Port 1 | | | Port 2 | | | Port 3 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **RxD0** | **TxD0** | **$\overline{\text{TSC0}}$** | **RxD1** | **TxD1** | **$\overline{\text{TSC1}}$** | **RxD2** | **TxD2** | **$\overline{\text{TSC2}}$** | **RxD3** | **TxD3** | **$\overline{\text{TSC3}}$** |
| 0 | IN0 | OUT0 | $\overline{\text{TSC0}}$ | IN1 | OUT1 | $\overline{\text{TSC1}}$ | IN2 | OUT2 | $\overline{\text{TSC2}}$ | IN3 | OUT3 | $\overline{\text{TSC3}}$ |
| 1 | IN0 for AIS0=1 | OUT0 | $\overline{\text{TSC0}}$ | IN0 for AIS0=0 | high Z | AIS0 | IN1 for AIS1=1 | OUT1 | $\overline{\text{TSC1}}$ | IN1 for AIS1=0 | high Z | AIS1 |
| 2 | – | OUT | $\overline{\text{TSC}}$ | – | high Z | AIS0 | IN for AIS1=1 | undef. | undef. | IN for AIS1=0 | high Z | AIS1 |
| 3 | IN0 for AIS0=1 | OUT0 | $\overline{\text{TSC0}}$ | IN0 for AIS0=0 | $\overline{\text{OUT0}}$ | AIS0 | IN1 for AIS0=1 | OUT1 | $\overline{\text{TSC1}}$ | IN1 for AIS1=0 | $\overline{\text{OUT1}}$ | AIS1 |

**Figure 64** shows the correlation between physical and logical PCM ports for PCM modes 0, 1, 2, 3:

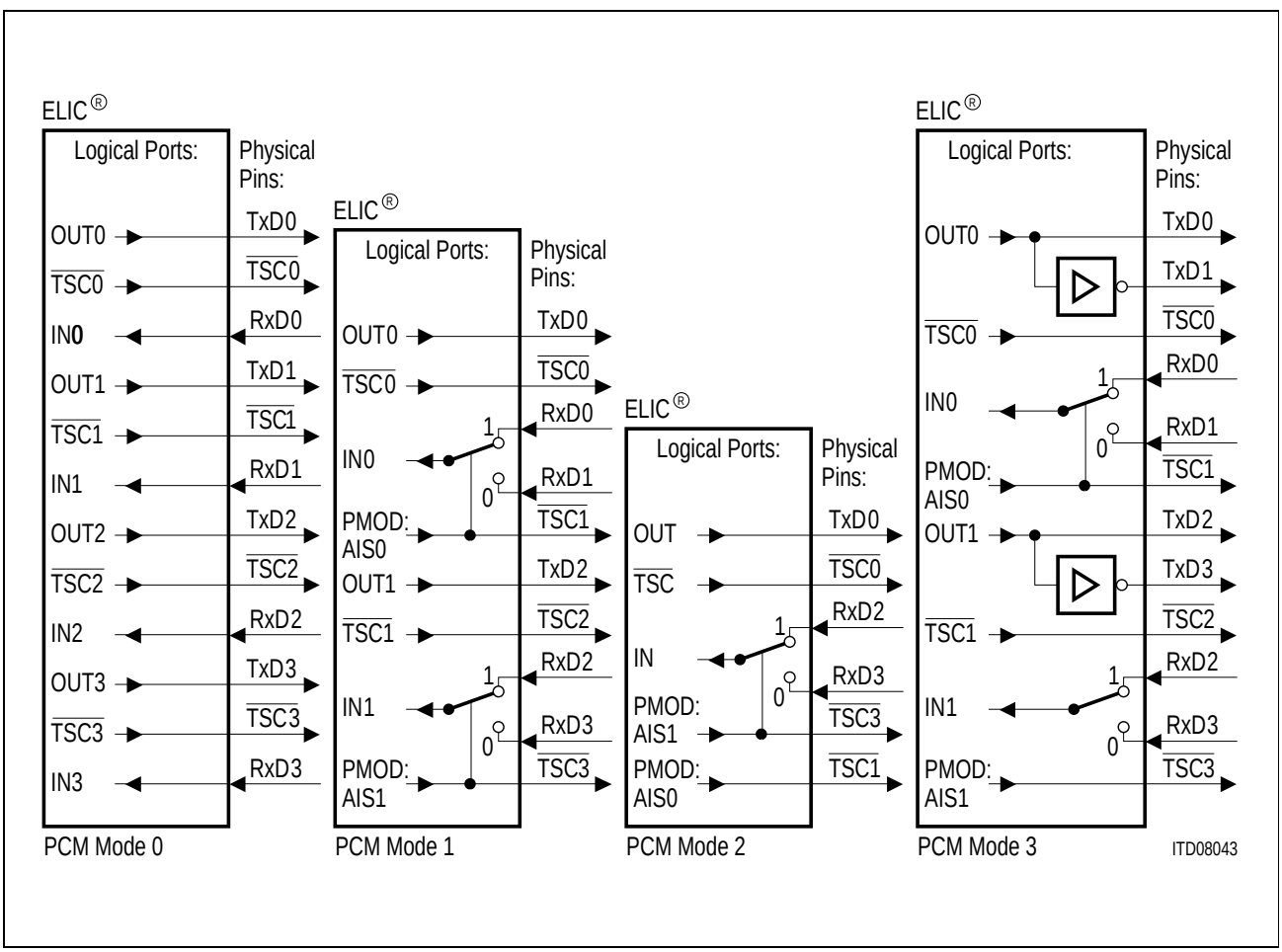



**Figure 64**
**Correlation between Physical and Logical PCM Ports**

### PCM Input Comparison PMOD:AIC1 … AIC0

If the PCM input comparison is enabled, the ELIC checks the contents of two PCM receive lines (physical ports) against each other for mismatches. (Also refer to **chapter 5.8.2**).

The comparison function is operational in all PCM modes, a redundant PCM line which can be switched over to by means of the PMOD:AIS bits is of course only available in PCM modes 1, 2 and 3.

AIC0 set to logical 1 enables the comparison function between RxD0 and RxD1.
AIC1 set to logical 1 enables the comparison function between RxD2 and RxD3.
AIC1, AIC0 set to logical 0 disables the respective comparison function.

### PCM Standby Mode OMDR:PSB

In standby mode (OMDR:PSB = 0), the PCM interface output pins TxD0 … 3 are set to high impedance and those ($\overline{\text{TSC\#}}$) pins which are actually used as tristate control signals are set to logical 1 (inactive).

Note that the internal operation of the ELIC is not affected in standby mode, i.e. the received PCM data is still written into the downstream data memory and may still be processed by the ELIC (switched to the CFI or to the µP, compared with other input line, etc.)

In operational mode (OMDR:PSB = 1), the PCM output pins transmit the contents of the upstream data memory data field or may be set to high impedance via the data memory tristate field (refer to **chapter 5.3.3.2**).

### PCM Test Loop OMDR:PTL

The PCM test loop function can be used for diagnostic purposes if desired. If however a ‘simple’ CFI to CFI connection (CFI → PCM → CFI loop) shall be established, it is recommended to program the PCM loop in the control memory (refer to **chapter 5.4.3.1**).

If OMDR:PTL is set to logical 1, the test loop is enabled i.e. the physical transmit pins TxD# are internally connected to the corresponding physical receive pins RxD#, such that data transmitted over TxD# are internally looped back to RxD# and data externally received over RxD# are ignored. The TxD# pins still output the contents of the upstream data memory according to the setting of the tristate field.

Note that this loop back function can only work if the upstream and downstream bit shifts match and if the port assignment (PMOD:AIS1 … 0) is such that a logical transmitter is looped back to a logical receiver (e.g. the PTL loop cannot work in PCM mode 2!).

For normal operation OMDR:PTL should be set to logical 0 (test loop disabled).

**Figure 65** illustrates the effect of the PTL bit:

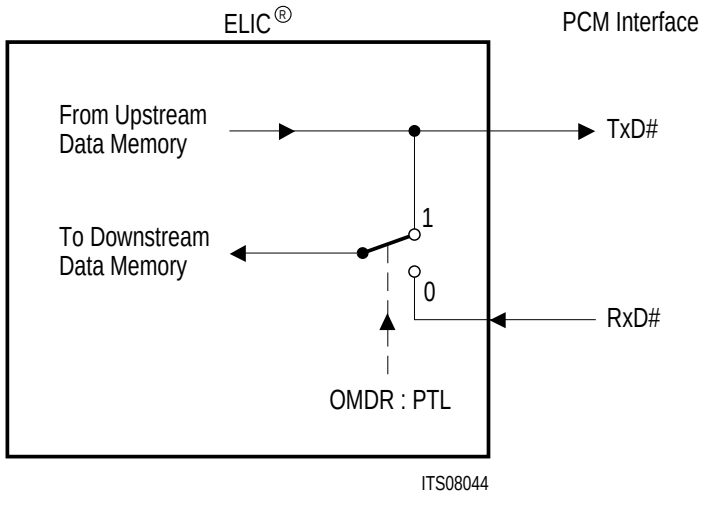



**Figure 65**
**Effect of the OMDR:PTL Bit**

### 5.2.2 Configurable Interface Configuration

#### 5.2.2.1 CFI Interface Signals

The configurable interface signals are summarized in the table below:

**Table 30**
**Signals at the Configurable Interface**

| Pin No. | Symbol | I: Input O: Output | Function |
|---|---|---|---|
| 34<br>35<br>36<br>37 | DD0/SIP0<br>DD1/SIP1<br>DD2/SIP2<br>DD3/SIP3 | O/IO<br>O/IO<br>O/IO<br>O/IO | Data downstream outputs in CFI modes 0, 1 and 2 (PCM and IOM applications).<br>Bidirectional serial interface ports in CFI mode 3 (SLD application).<br>Tristate or open drain output drivers selectable (OMDR:COS). |
| 29<br>30<br>32<br>33 | DU0/SIP4<br>DU1/SIP5<br>DU2/SIP6<br>DU3/SIP7 | I/IO<br>I/IO<br>I/IO<br>I/IO | Data upstream inputs in CFI modes 0, 1 and 2 (PCM and IOM applications).<br>Bidirectional serial interface ports in CFI mode 3 (SLD application).<br>Tristate or open drain output drivers for SIP lines selectable (OMDR:COS). |
| 27 | FSC | I or O | Frame synchronization input (CMD1:CSS = 1) or output (CMD1:CSS = 0). |
| 28 | DLC | I or O | Data clock input (CMD1:CSS = 1) or output (CMD1:CSS = 0). |

#### 5.2.2.2 CFI Registers

The characteristics at the configurable interface (timing, modes of operation, etc. … ) are programmed in the 5 CFI interface registers and the Operation Mode Register OMDR. The function of each bit is described in **chapter 5.2.2.3**. For addresses refer to **chapter 4.1**.

**CFI Mode Register 1** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| CMD1 | CCS | CSM | CSP1 | CSP0 | CMD1 | CMD0 | CIS1 | CIS0 |

**CFI Mode Register 2** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| CMD2 | FC2 | FC1 | FC0 | COC | CXF | CRR | CBN9 | CBN8 |

**CFI Bit Number Register** read/write reset value: $FF_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| CBNR | CBN7 | CBN6 | CBN5 | CBN4 | CBN3 | CBN2 | CBN1 | CBN0 |

**CFI Timeslot Adjustment Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| CTAR | 0 | TSN6 | TSN5 | TSN4 | TSN3 | TSN2 | TSN1 | TSN0 |

**CFI Bit Shift Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| CBSR | 0 | CDS2 | CDS1 | CDS0 | CUS3 | CUS2 | CUS1 | CUS0 |

**CFI Bit Subchannel Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| CSCR | SC31 | SC30 | SC21 | SC20 | SC11 | SC10 | SC01 | SC00 |

**Operation Mode Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| OMDR | OMS1 | OMS0 | PSB | PTL | **COS** | MFPS | **CSB** | RBS |

### 5.2.2.3 CFI Characteristics

In the following the configurable interface characteristics that can be programmed in the CFI registers are explained in more detail.

#### CFI Mode CMD1:CMD1, CMD0

The CFI mode primarily defines the actual number of CFI ports that can be used for switching purposes (logical ports). 1, 2 or 4 duplex or 8 bidirectional logical CFI ports can be selected. Since the channel capacity of the ELIC is constant (128 channels/direction), the CFI mode also influences the maximum possible data rate.

In each CFI mode a reference clock (RCL) of a specific frequency is required. This clock may be derived from the PCM clock signal PDC (CMD1:CSS = 0) or from the DCL signal (CMD1:CSS = 1). Also refer to **figure 66** and **figure 67**.

**Table 31** states the specific characteristics of each CFI mode.

(DR = CFI data rate, N = number of 8 bit timeslots in PCM frame, du = duplex port, bi = bidirectional port).

**Table 31**
**Modes at the Configurable Interface**

| CMD1 | CMD0 | CFI Mode | Number (Label) of Logical Ports | CFI Data Rate [kBit/s] | | Min. Required CFI DR [kBit/s] relative to PCM Data Rate | Necessary Reference Clock (RCL) | DCL Output Frequencies CMD1: CSS = 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | min. | max. | | | |
| 1 | 1 | 3 | 8 bi (0 … 7) | 128 | 1024 | 16N/3 | 4 × DR | DR, 2 × DR |
| 0 | 0 | 0 | 4 du (0 … 3) | 128 | 2048 | 32N/3 | 2 × DR | DR, 2 × DR |
| 0 | 1 | 1 | 2 du (0 … 1) | 128 | 4096 | 64N/3 | DR | DR |
| 1 | 0 | 2 | 1 du | 128 | 8192 | 64N/3 | 0.5 × DR | DR |

*Note: The label is used to specify a CFI port when programming a switching function. It should not be confused with the physical port number which refers to actual hardware pins. The relationship between logical and physical port numbers is given in **table 35** and is illustrated in **figure 82**.*

### Important Note

It should be noticed that there are some restrictions concerning the PCM to CFI data rate ratio. If the CFI data rate is chosen higher than the PCM data rate, no restrictions apply. If however the CFI data rate is lower than the PCM data rate, a minimum CFI date rate relative to the PCM data rate must be maintained (refer also to examples below).

Another important restriction is, that the number of bits per CFI frame must always be modulo 16.

### Examples

If the PCM frame consists of 32 timeslots (2048 kBit/s), the minimum possible CFI data rate in CFI mode 0 is $(32 \times 32)/3$ = 341.3 kBit/s or if rounded to an integer number of timeslots 344 kBit/s. It is thus not possible to have an IOM-1 interface with 256 kBit/s together with a 2048 kBit/s PCM interface in CFI mode 0. If instead the PCM frame consists of 24 timeslots (1536 kBit/s), the IOM-1 data rate of 256 kBit/s is feasible since $(24 \times 32)/3$ = 256 kBit/s.

### CFI Clock and Framing Signal Source CMD1:CSS

The PCM interface is always clocked and synchronized by the PDC and PFS input signals. The configurable interface however can be clocked and synchronized either by signals internally derived from PDC and PFS or it can be clocked and synchronized by the externally applied DCL and FSC input signals.

If **PDC** and **PFS** are selected as **clock and framing signal source** (CMD1:CSS = 0), the CFI reference clock CRCL is obtained out of PDC after division by 1, 1.5 or 2 according to the prescaler selection (CMD:CSP1 … 0). The CFI frame structure is synchronized by the PFS input signal. The ELIC generates DCL and FSC as output signals which may be specified by CMD2:COC (DCL clock rate) and CMD2:FC2 … 0 (FSC pulse form). This mode should be selected whenever the required CFI data rate can be obtained out of the PCM clock source using the internal prescalers. An overview of the different possibilities to generate the PCM and CFI data and clock rates for CMD1:CSS = 0 is given in **figure 66**.

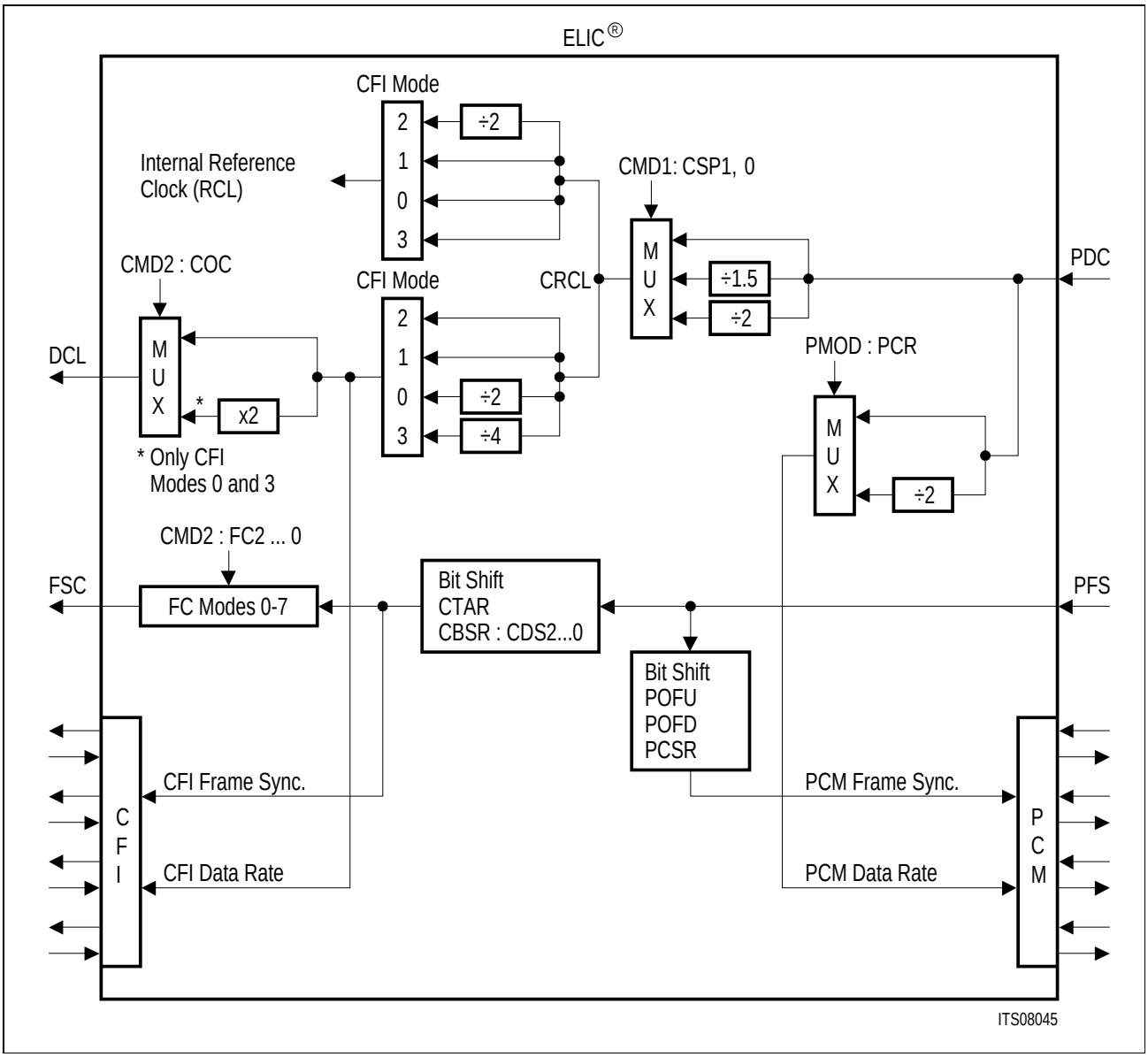



**Figure 66**
**ELIC® Clock Sources for the CFI and PCM Interfaces if CMD1:CSS = 0**

If **DCL** and **FSC** are selected as **clock and framing signal source** (CMD1:CSS = 1), the CFI reference clock CRCL is obtained out of the DCL input signal after division by 1, 1.5 or 2 according to the prescaler selection (CMD1:CSP1 … 0). The CFI frame structure is synchronized by the FSC input signal. Note that although the frequency and phase of DCL and FSC may be chosen almost independently with respect to the frequency and phase of PDC and PFS, the CFI clock source must still be synchronous to the PCM interface clock source i.e. the two clock sources must always be derived from one master clock. This mode must be selected if it is impossible to derive the required CFI data rate from the PCM clock source. An overview of the different possibilities to generate the PCM and CFI data and clock rates for CMD1:CSS = 1 is given in **figure 67**.

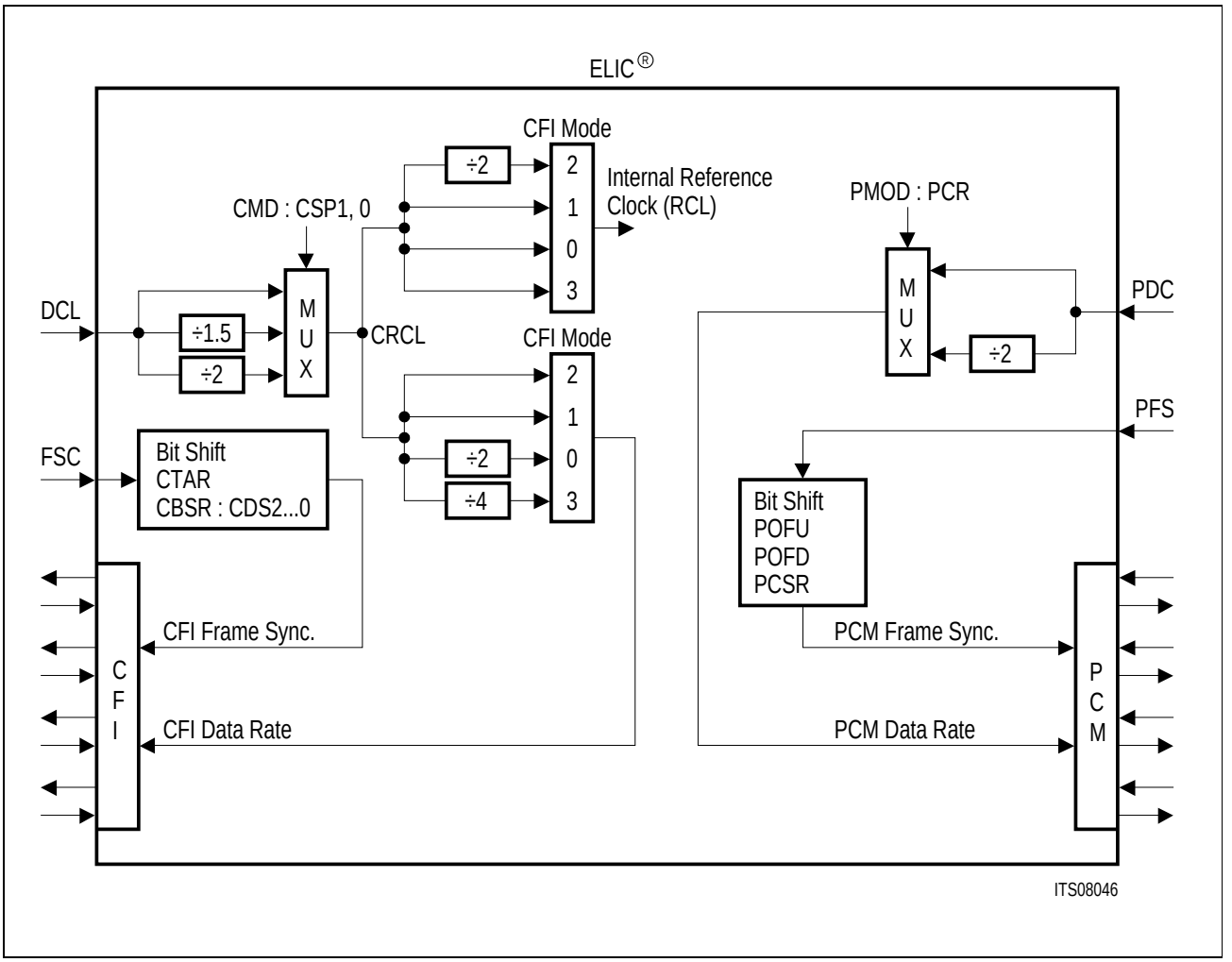



**Figure 67**
**ELIC® Clock Sources for the CFI and PCM Interfaces if CMD1:CSS = 1**

### CFI Clock Source Prescaler CMD1:CSP1 … 0

The CFI clock source PDC (CMD1:CSS = 0) or DCL (CMD1:CSS = 1) can be divided by a factor of 1, 1.5 or 2 in order to obtain the CFI reference clock CRCL (see **table 32**).

Note that in CFI mode 2, the frequency of RCL is only half the CFI data rate.

**Table 32**
**Prescaler Divisors**

| CSP1 | CSP0 | Prescaler Divisor |
|---|---|---|
| 0 | 0 | 2 |
| 0 | 1 | 1.5 |
| 1 | 0 | 1 |
| 1 | 1 | not allowed |

**Figure 68** shows the relationship between the DCL input and the generated RCL for the different prescaler divisors in case CMD1:CSS = 1:

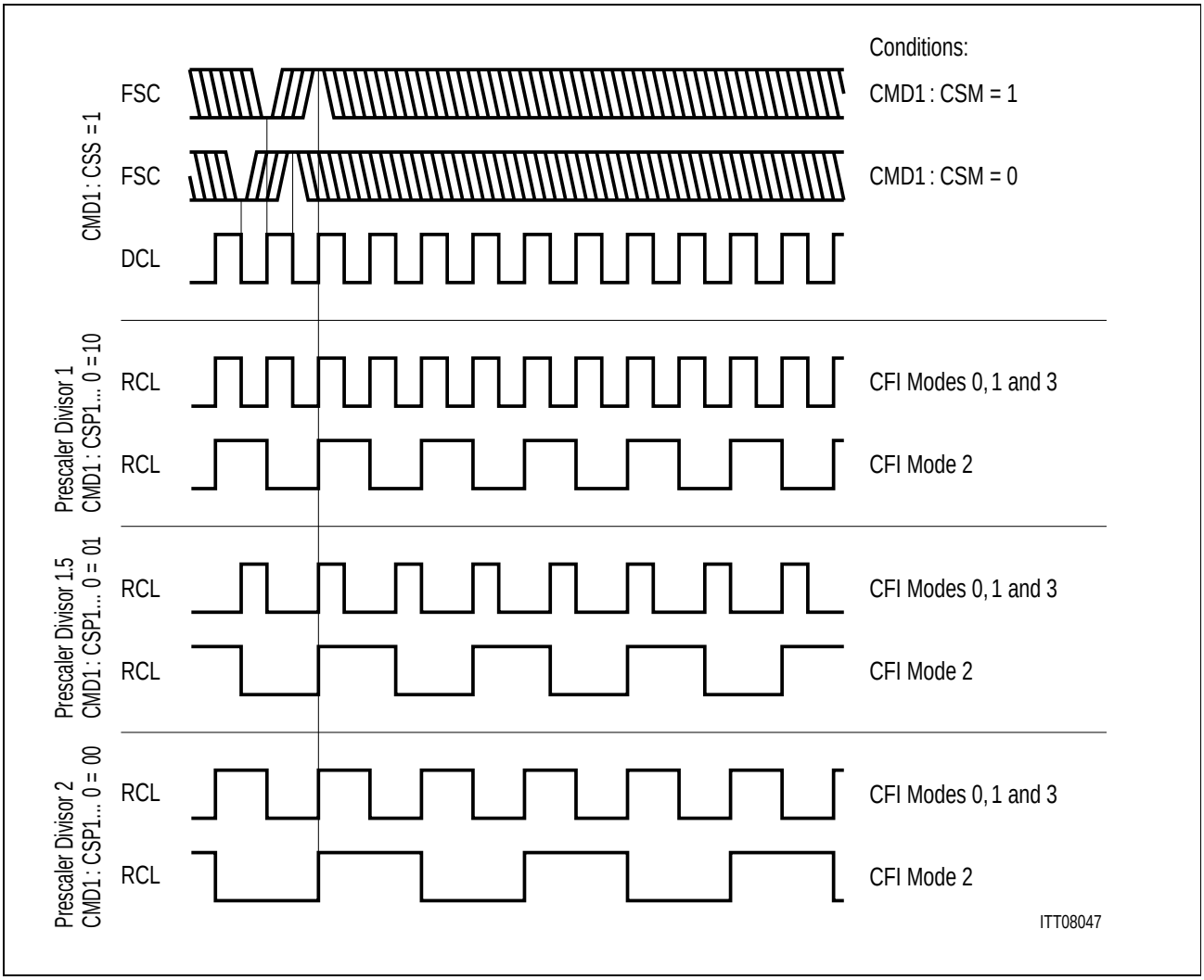



**Figure 68**
**Clock Signal Timing for the Different Prescaler Divisors if CMD1:CSS = 1**

### CFI Clock Output Rate CMD2:COC

This feature applies only if the configurable interface is clocked and synchronized via the PCM interface clock and framing signals (PDC, PFS), i.e. if CMD1:CSS = 0.

In this case the ELIC delivers an output clock signal at pin DCL with a frequency identical to or double the selected CFI data rate:

For CMD2:COC = 0, the frequency of DCL is identical to the CFI data rate (all CFI modes)

For CMD2:COC = 1, the frequency of DCL is twice the CFI data rate (CFI modes 0 and 3 only!)

**Figure 69** shows the relationship between the PFS, PDC, RCL and DCL signals in the different CFI modes.

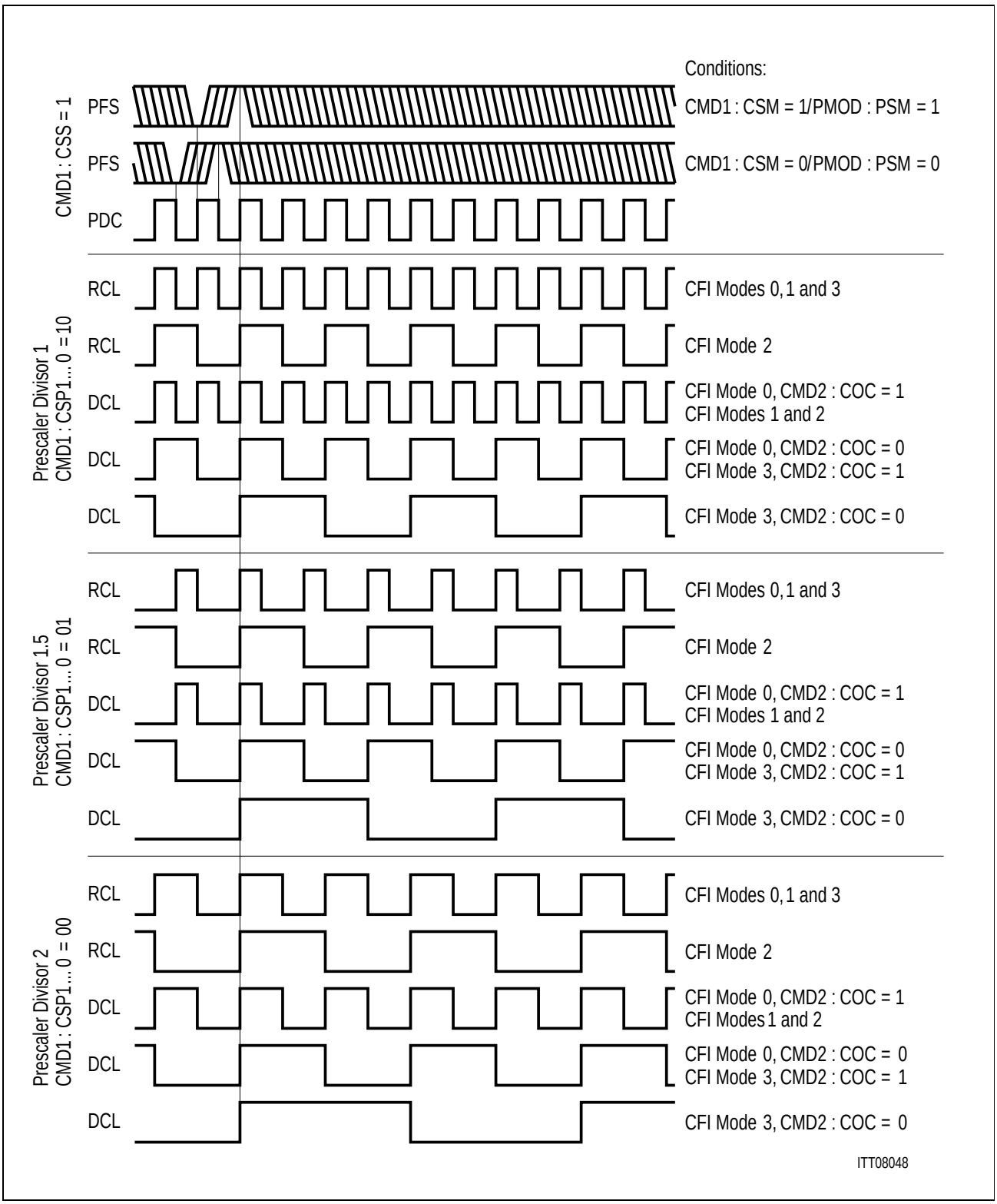



**Figure 69**
**Clock Signal Timing for the Different Prescaler Divisors if CMD1:CSS = 0**

### CFI Framing Signal Output Control CMD2:FC2 … 0

This feature applies only if the configurable interface is clocked and synchronized via the PCM interface clock and framing signals (PDC, PFS), i.e. if CMD1:CSS = 0.

In this case the ELIC delivers an output framing signal at pin FSC with a programmable pulse width and position.

Note that the up- and downstream CFI frame position relative to the FSC output is not affected by the setting of the CTAR and CBSR:CDS2 … 0 register bits.

**Table 33** summarizes the 7 possible FSC Control (FC) modes:

**Table 33**
**Applications of the Different Framing Control Modes**

| FC2 | FC1 | FC0 | FC Mode | Main Applications | Notes |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | IOM-1 multiplexed (burst) mode | SBC, IBC, IEC-T |
| 0 | 0 | 1 | 1 | General purpose | |
| 0 | 1 | 0 | 2 | General purpose | |
| 0 | 1 | 1 | 3 | General purpose | |
| 1 | 0 | 0 | 4 | Special SLD application | 2 ISAC-S per SLD port |
| 1 | 0 | 1 | 5 | reserved | |
| 1 | 1 | 0 | 6 | IOM-2, IOM-1 or SLD modes | Standard IOM-2 setting; no Superframes generated |
| 1 | 1 | 1 | 7 | Software timed multiplexed IOM-2 applications | Standard IOM-2 setting; Superframes generated |

**Figure 70** and **figure 71** show the position of the FSC pulse relative to the CFI frame:

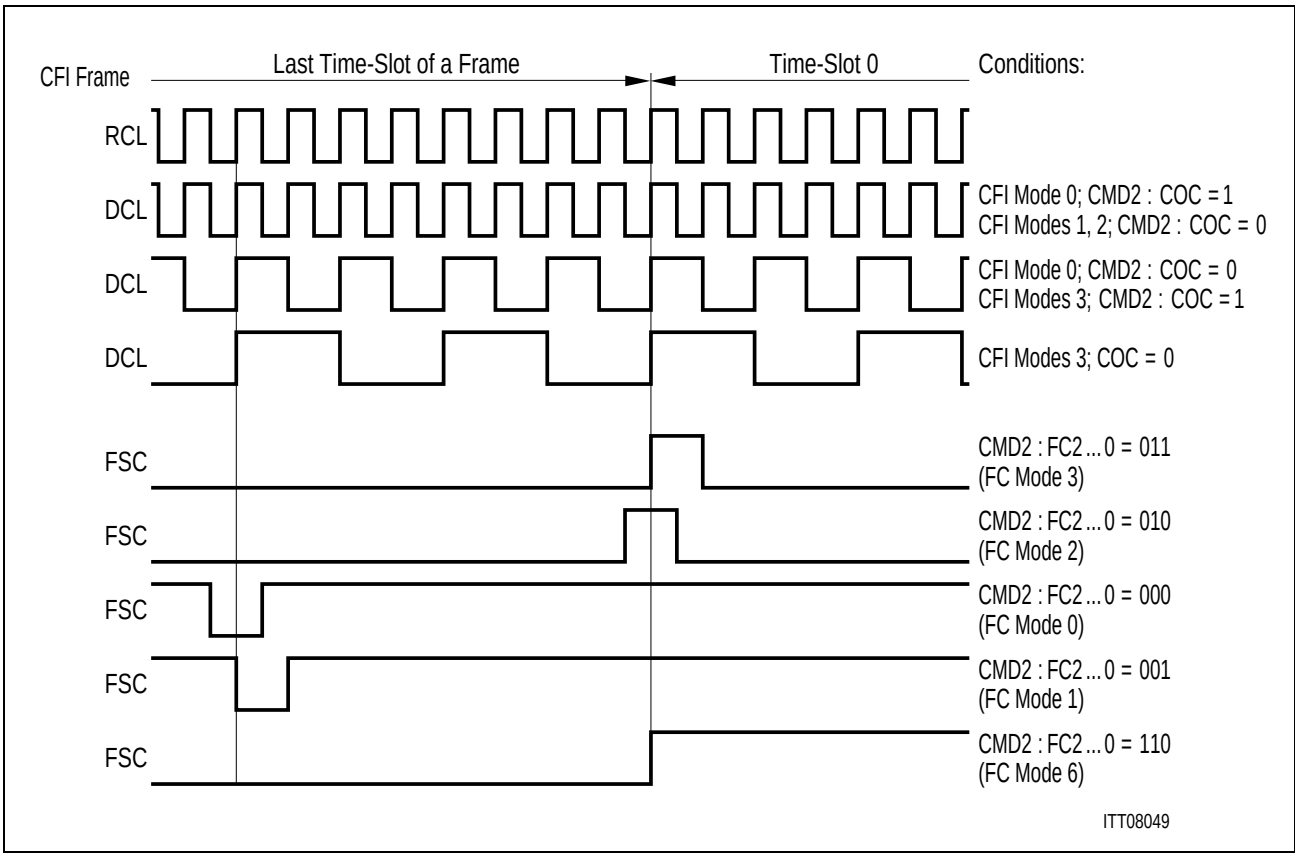



**Figure 70**
**Position of the FSC Signal for FC Modes 0, 1, 2, 3 and 6**

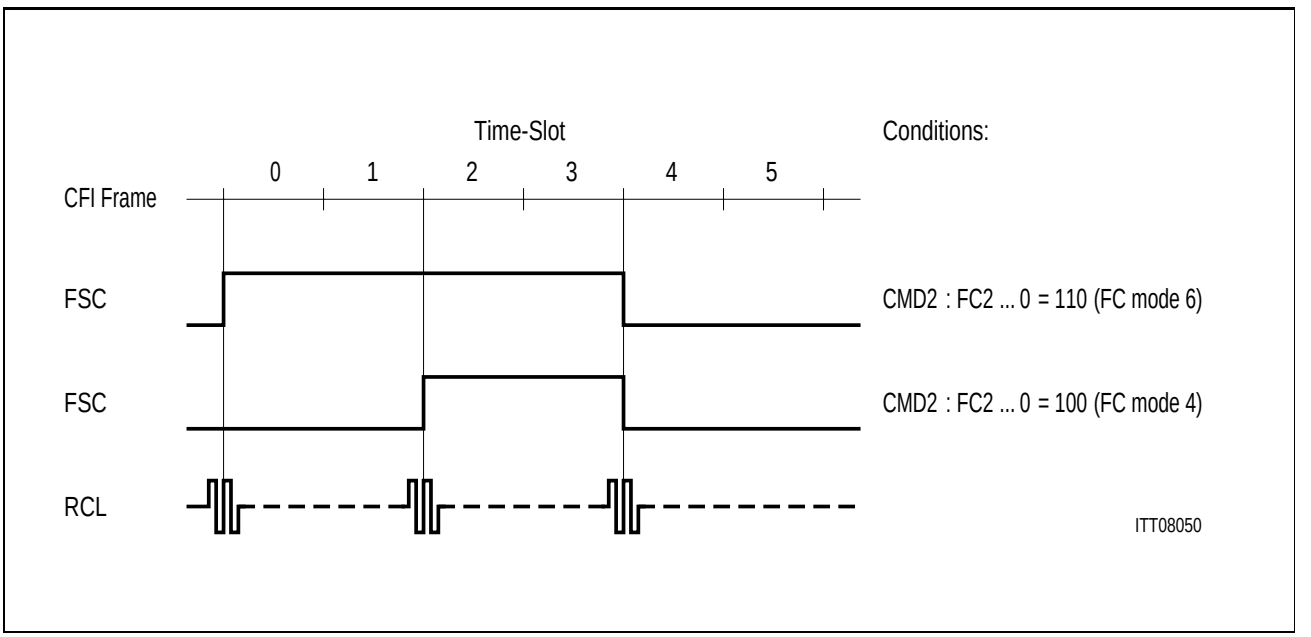



**Figure 71**
**Position of the FSC Signal for FC Modes 4 and 6**

## Application Examples of the Different FC Modes

### FC Mode 0

FC mode 0 applies for IOM-1 multiplexed mode applications, i.e. for IOM-1 interfaces with 2048 kBit/s data rate. Accommodated layer-1 devices: SBC (PEB 2080), IBC (PEB 2095), IEC-T (PEB 20901/20902), …

In IOM-1 mux. mode, the frame is synchronized with a negative pulse with a duration of one DCL period which marks bit number 251. The bits are transmitted with the falling clock edge and received with the rising clock edge.

Required register setting: CMD1 = $0XXX0000_B$, CMD2 = $1C_H$, CBNR = $FF_H$, CTAR = $XX_H$, CBSR = $X0_H$.

**Figure 72** shows the relationship between FSC, DCL, DD# and DU#:

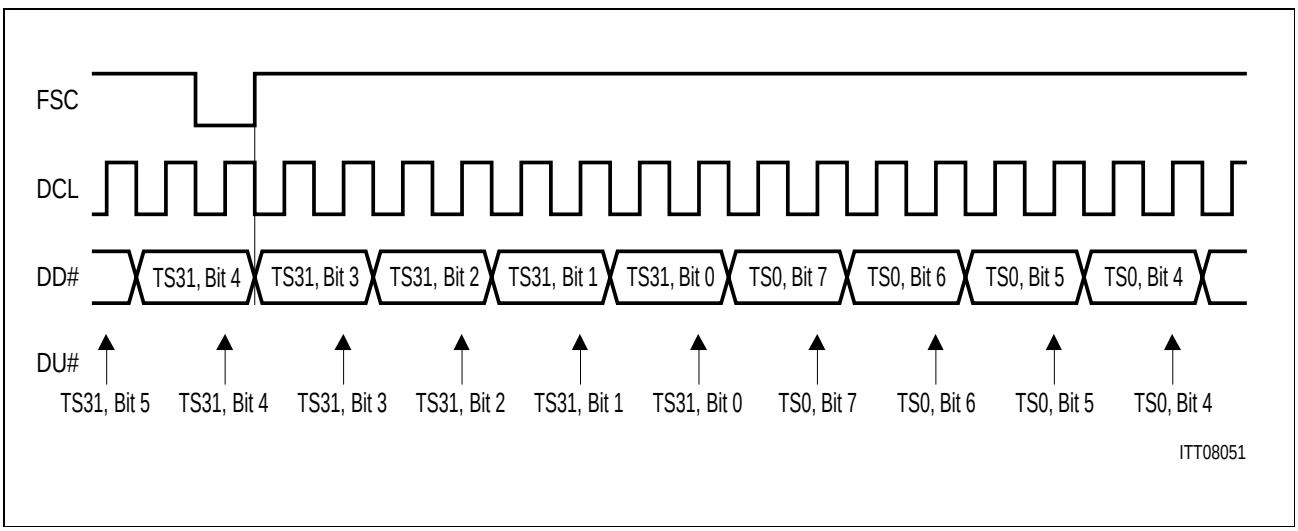



**Figure 72**
**Multiplexed IOM®-1 Interface Signals**

### FC Mode 1

FC mode 1 is similar to FC mode 0. The FSC pulse is shifted by half a RCL period to the right compared to FC mode 0. It can be used for general purposes.

### FC Mode 2

FC mode 2 is similar to FC mode 3. The FSC pulse is shifted by half a RCL period to the left compared to FC mode 3. It can be used for general purposes.

### FC Mode 3

FC mode 3 can be used for IOM-2 applications, but it should be noted that some IOM-2 layer-1 transceivers will interpret an FSC pulse of only one DCL period as a superframe marker (e.g. SBCX PEB 2081, IEC-Q PEB 2091, … ), and it is not allowed to provide a superframe marker in every frame. For these applications it is recommended to use either FC mode 6 or FC mode 7.

### FC Mode 4

FC mode 4 applies for special SLD applications like 2 ISAC-S devices connected to one SIP line. Usually each SIP line carries the two 64 kBit/s B channels followed by a feature control and a signaling channel. The feature control and signaling channels however are not required for all applications. This is, for example, the case if a digital subscriber circuit (S- or U- layer-1 transceiver) is connected via an ISDN Communication Controller (ICC PEB 2070) to the ELIC. The task of the ICC is to handle the D-channel and to switch the B1 and B2 channels from the SLD to the IOM-1 interface. The capacity of such an SLD line card can be doubled if the unused timeslots for the feature control and signaling channels are also used as 64 kBit/s B channels. This is possible if the additionally connected ICC (or ISAC-S) is synchronized with an FSC that is delayed by 2 timeslots i.e. the rising FSC edge is at the beginning of timeslot 2 instead of 0. The CFI timeslots 2, 3, 6 and 7 can then be programmed as normal B channels within the ELIC instead of being programmed as feature control and signaling channels.

### FC Mode 6

This is the most often used type of FSC signal, because it covers the standard IOM-1, IOM-2 and SLD applications. The rising edge of FSC marks timeslot 0, bit 7 of the CFI frame.

The pulse width is 32 bits or 4 timeslots, i.e. the FSC is symmetrical (duty cycle 1:1) if the CFI frame consists of 8 timeslots (SLD), and the FSC is high during the first IOM-2 channel if the CFI frame consists of 32 timeslots (IOM-2).

Required register setting for IOM-2:

CMD1 = $0XXX0000_B$, CMD2 = $D0_H$, CBNR = $FF_H$, CTAR = $XX_H$, CBSR = $X0_H$.

**Figure 73** shows the relationship between FSC, DCL, DD# and DU#:

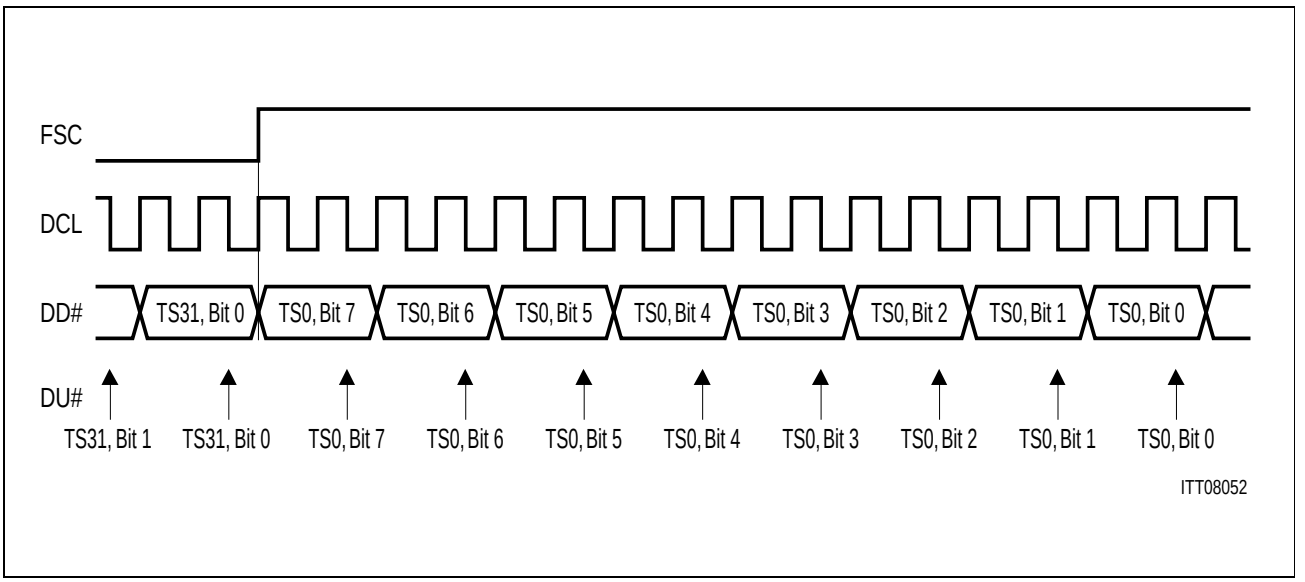



**Figure 73**
**IOM®-2 Interface Signals**

Required register setting for SLD:

CMD1 = $0XXX1100_B$, CMD2 = $D0_H$, CBNR = $1F_H$, CTAR = $XX_H$, CBSR = $X0_H$.

**Figure 74** shows the relationship between FSC, DCL and SIP#:

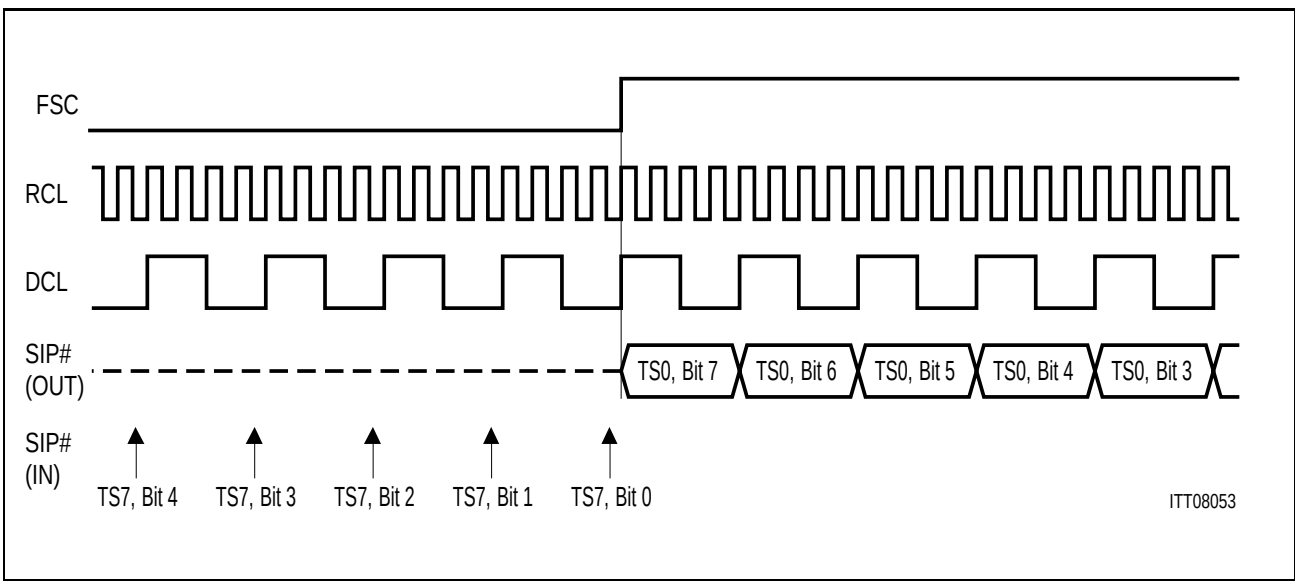



**Figure 74**
**SLD Interface Signals**

### FC Mode 7

FC mode 7 is intended for IOM-2 line cards to synchronize the multiframe structure among several S- or $U_k$-interface transceivers. The layer-1 multiframe is reset by an FSC pulse having a width of at most, one DCL period. Between the multiframe reset pulses, FSC pulses with a width of at least two DCL periods must be applied. Devices which support this option are for example the OCTAT-P (PEB 2096-H), QUAT-S (PEB 2084-H), SBCX (PEB 2081), and the IEC-Q (PEB 2091).

FC mode 7 is a combination of FC modes 3 and 6. The timer register TIMR must be loaded with the required multiframe period (e.g. 5 ms for the S-interface or 12 ms for the $U_k$-interface). When the timer is started with CMDR:ST, a cyclic multiplexing process is started: whenever the timer expires, the frame signal has the pulse shape of FC mode 3 during one frame. For all the other frames the FSC signal has the pulse form of FC mode 6.

After setting the CMDR:ST bit, the inverted value of TVAL is loaded to the timer and the timer is incremented as soon as time slot 3 is passed (i.e. the FSC high phase is passed which lasts for 4 TSs in FC mode 6) and then every 250 µs.

When the timer expires (timer value = 0), an interrupt is generated immediately and the next FSC pulse has the shape of FC mode 3.

**Figure 75** illustrates this behavior for a timer value of TVAL6 … 0 = 0000001.

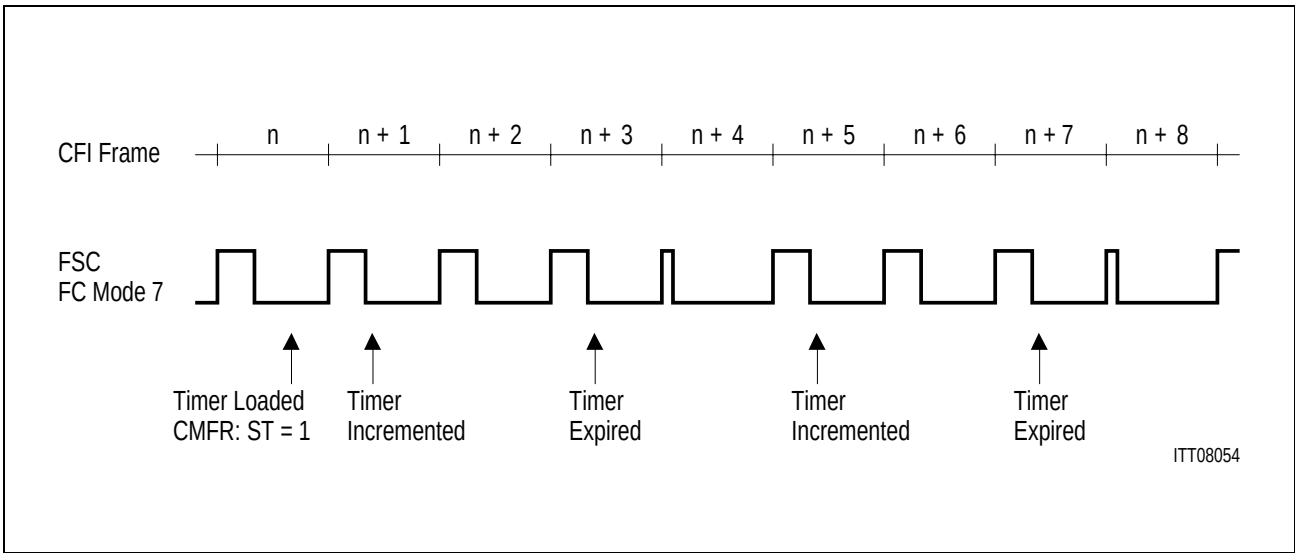



**Figure 75**
**FSC Signal in FC Mode 7**

*Note: If the timer is stopped, the generated pulse form is the one of FC mode 6.*

Timer value examples:

Required timer value for 5 ms period: TIMR:TVAL6 … 0 = $010011_B$, e.g. TIMR = $13_H$
Required timer value for 12 ms period: TIMR:TVAL6 … 0 = $101111_B$, e.g TIMR = $2F_H$

### CFI Bit Number CMD2, CBNR:CBN9 … CBN0

The CFI data rate is determined by the reference clock RCL and the CFI mode selected by CMD1:CMD1 … 0. The number of bits which constitute a CFI frame can be derived from this data rate by division of 8000 (8 kHz frame structure). If the CFI interface is for example operated at 2048 kBit/s, the frame would consists of 256 bits or 32 timeslots.

This number of bits must be programmed to CMD2,CBNR:CBN9 … 0 as indicated below. Note that the formula is valid for all CFI modes:

CBN9 … 0 = number of bits – 1

### Examples

A CFI frame consisting of 64 timeslots would require a setting of
CBN9 … 0 = $64 \times 8 - 1 = 511D = 01\ 1111\ 1111_B$

A CFI frame consisting of 48 timeslots would require a setting of
CBN9 … 0 = $48 \times 8 - 1 = 383D = 01\ 0111\ 1111_B$

### CFI Synchronization Mode CMD1:CSM

The CFI interface can either be synchronized via the PFS pin (CMD1:CSS = 0), or via the FSC pin (CMD1:CSS = 1). A transition from low to high of either PFS or FSC synchronizes the CFI frame. The PFS (FSC) signal is internally sampled with the PDC (DCL) clock:

If CSM is set to logical 0, the PFS/FSC signal is sampled with the falling clock edge of PDC/DCL, if set to logical 1, the PFS/FSC signal is sampled with the rising clock edge of PDC/DCL.

If CMD1:CSS is set to logical 0 (CFI clocks are internally derived from the PCM clocks), then CMD1:CSM should be equal to PMOD:PSM.

If CMD1:CSS is set to logical 1 (CFI clock signals are inputs), then CMD1:CSM should be selected such that stable low and high phases of the FSC signal can be detected, meeting the set-up ($T_{FS}$) and hold ($T_{FH}$) times with respect to the programmed DCL clock edge.

The high phase of the PFS/FSC pulse may be of arbitrary length, however it must be assured that it is sampled low at least once before the next framing pulse.

The relationship between the framing and clock signals (PFS, FSC, PDC, DCL and RCL) for the different modes of operation is illustrated in **figures 68** and **69**.

*Note: In case DCL and FSC are selected as inputs (CMD1:CSS = 1), FSC must always be synchronized with the positive edge of DCL (CMD1:CSM = 1). Otherwise, an IOM-2 compatible timing cannot be installed by means of a bit shift (When the negative edge is used for synchronization the internal frame start is delayed by one DCL clock. In double rate mode a bit shift of half a bit cannot be adjusted). Anyway, if the rising edges of DCL and FSC do not meet the frame setup time $T_{FS}$,*

*additional hardware must delay the frame signal to enable a synchronization with the positive edge of DCL.* ***Figure 76*** *gives a suggestion of how to adapt the external timing.*

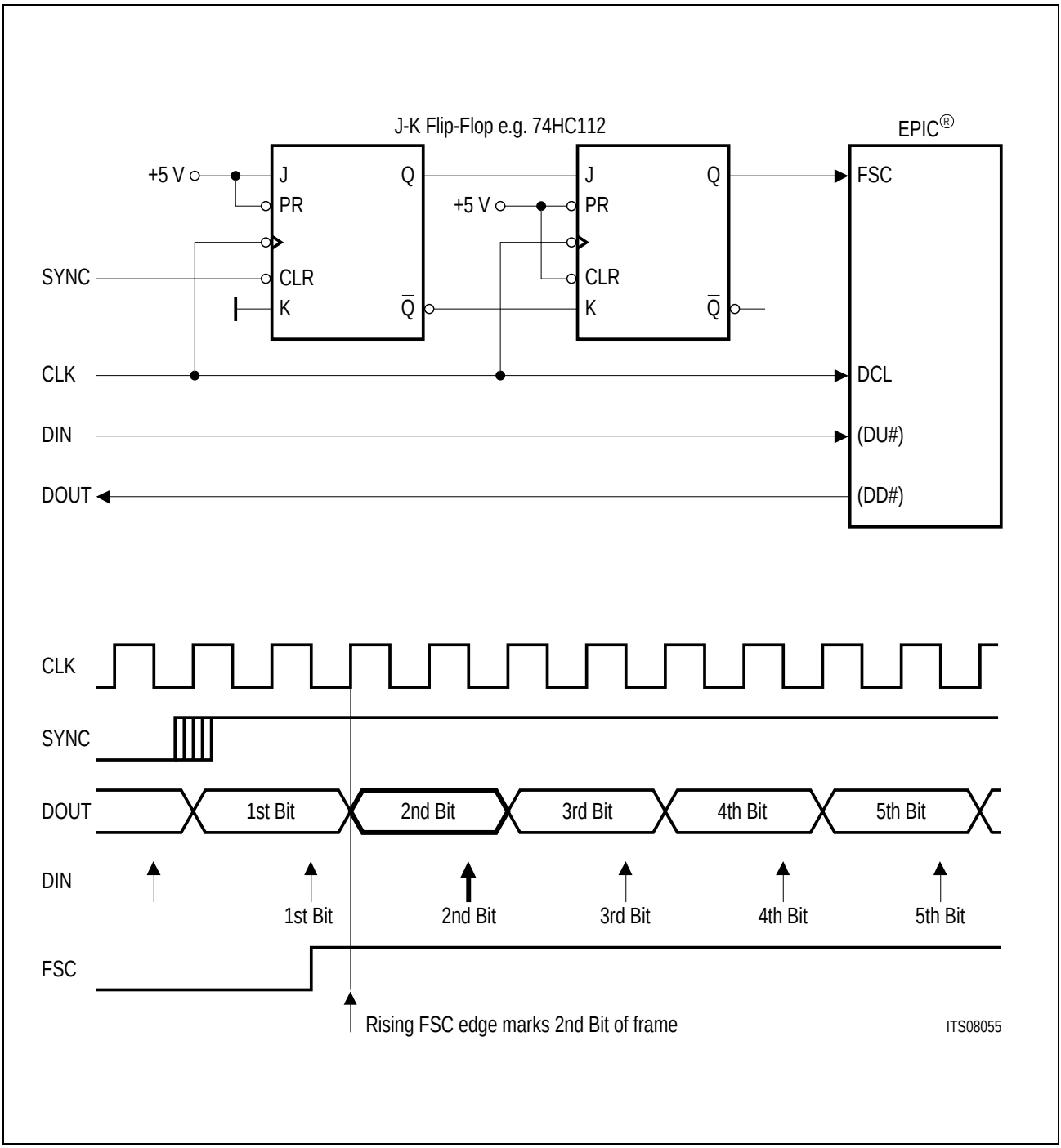



**Figure 76**
**Circuit for Delaying the Framing Signal at the CFI Interface**

### CFI Bit Timing and Bit Shift CMD2, CTAR, CBSR

The position of the CFI frame can be shifted relative to the CFI frame synchronization pulse using the CFI Timeslot Adjustment Register CTAR and the CFI Bit Shift Register CBSR. This shifting can be performed simultaneously for up- and downstream directions with a one bit resolution by up to a whole frame. The upstream frame can additionally be shifted relative to the downstream frame by up to 15 bits. Furthermore, the polarity of the clock edge (CRCL) used for transmitting and sampling the data can be programmed in the CMD2 register.

Since the frame synchronization source of the configurable interface is either PFS (for CMD1:CSS = 0) or FSC (for CMD1:CSS = 1), the bit shift also refers to either the PFS or the FSC framing signal.

*Note: If **PFS/PDC** is selected as CFI **sync/clock source**, the timeslot and bit shift values programmed to CTAR and CBSR:CDS2 … 0 affect both the CFI data lines and the CFI output framing signal FSC. The CFI frame together with the FSC signal can thus be shifted with respect to the PCM frame (PFS). The position of the CFI frame relative to the FSC output signal is not affected by these settings but is instead determined by the FSC framing control mode programmed to CMD2:FC2 … 0. The upstream CFI frame can, however, still be shifted relative to the downstream CFI frame with the CBSR:CUS3 … 0 bits.*

If **FSC/DCL** is selected as CFI **sync/clock source**, the timeslot and bit shift functions affect the CFI frame with respect to the FSC framing input signal. In this case, the CFI frame start can be selected completely independently from the PCM frame start, it must only be assured that a phase relationship once established between the CFI and PCM frames is maintained all the time.

### CFI Timeslot Adjustment and Bit Shift

If CBSR = $20_H$, the CFI framing signal (PFS if CMD1:CSS = 0 or FSC if CMD1:CSS = 1) marks bit 7 of the CFI timeslot called **TSN** according to the following formula:

**CTAR:TSN6 … 0 = TSN + 2**

e.g. CTAR must be set to $02_H$ if the framing signal should mark timeslot 0, bit 7 (TS = 0). See examples.

Note that the value of TSN may not exceed the actual number of timeslots per CFI frame:

TSN = [ – 2; I – 3], I = total number of timeslots per CFI frame

From the zero offset bit position (CBSR = $20_H$) the CFI frame (downstream and upstream) can be shifted by up to 5 bits to the left (within the timeslot number TSN programmed in CTAR) and by up to 2 bits to the right (within the previous timeslot N – 1) by programming the CBSR:CDS2 … 0 bits:

**Table 34**
**CFI Shift with Respect to the Frame Synchronization Signal**

| CBSR:CDS2 … 0 | Timeslot # | Marked Bit # | Bit Shift |
|---|---|---|---|
| 000 | TSN – 1 | 1 | 2 bits to the right |
| 001 | TSN – 1 | 0 | 1 bit to the right |
| 010 | TSN | 7 | no bit shift |
| 011 | TSN | 6 | 1 bit to the left |
| 100 | TSN | 5 | 2 bits to the left |
| 101 | TSN | 4 | 3 bits to the left |
| 110 | TSN | 3 | 4 bits to the left |
| 111 | TSN | 2 | 5 bits to the left |

The bit shift programmed to CBSR:CDS2 … 0 affects both the upstream and downstream frame position in the same way.

If CBSR:CUS3 … 0 = 0000, the upstream frame is aligned to the downstream frame.

With CBSR:CUS3 … 0 = 0001 to 1111, the upstream CFI frame can be shifted relative to the downstream frame by up to 15 bits to the left as indicated in **figure 77**.

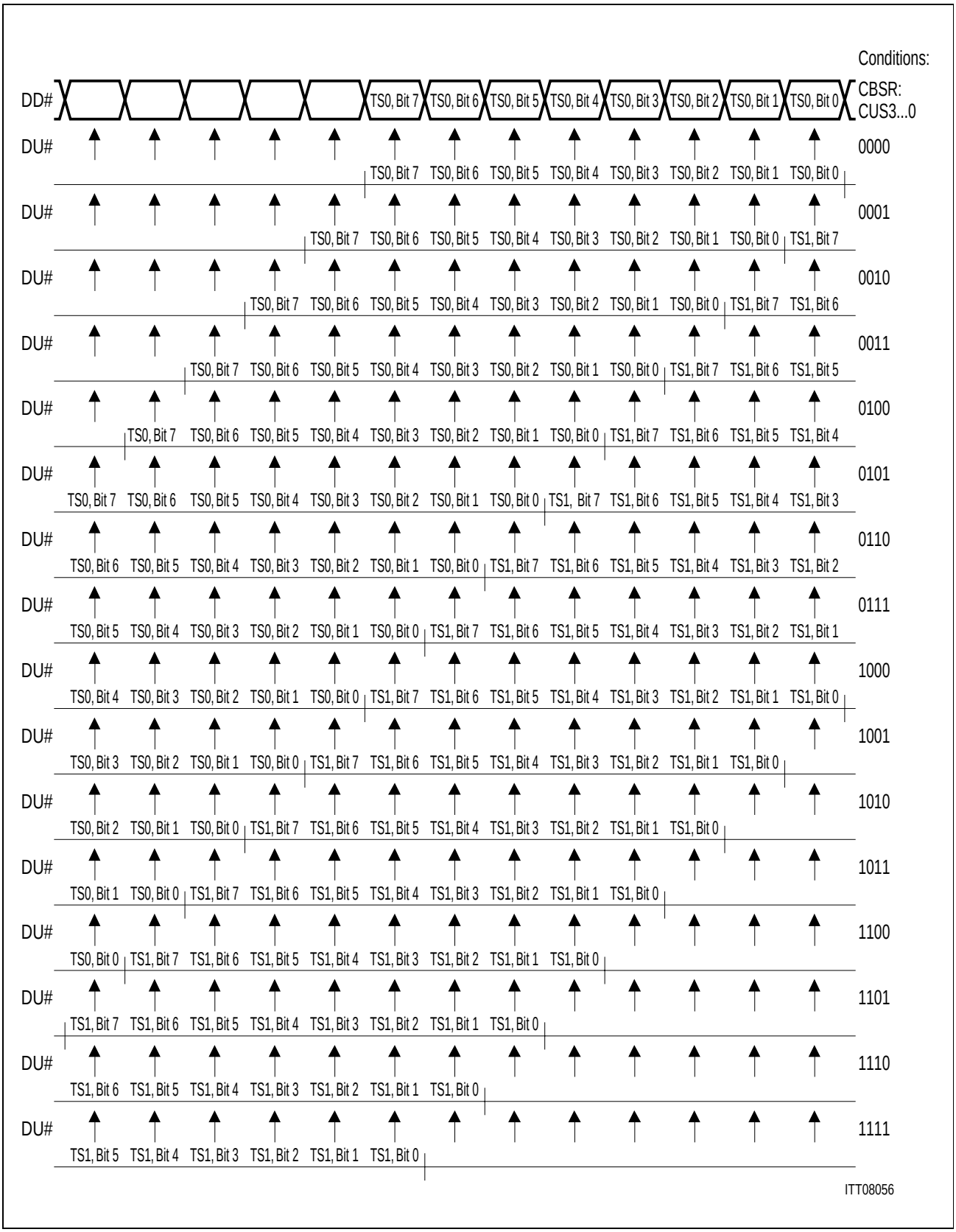



**Figure 77**
**CFI Upstream Bit Shifting**

### CFI Bit Timing

In CFI modes 0, 1 and 2, the rising or falling **CRCL** clock edge can be selected for transmitting and sampling the data.

In CFI mode 3, the rising or falling **CRCL** clock edge can be selected for transmitting the data, the sampling of data however must always be done with the falling edge of CRCL (CRR = 0).

If CMD2:CXF = 0 (CFI Transmit on Falling edge), the data is transmitted with the rising CRCL edge, if CXF = 1, the data is transmitted with the next following falling edge of CRCL.

If CMD2:CRR = 0 (CFI Receive on Rising edge), the data is sampled with the falling CRCL edge, if CRR = 1, the data is sampled with the next following rising edge of CRCL.

The relationship between the framing and clock signals and the CFI bit stream on DD# and DU# for CTAR = $02_H$ and CBSR = $20_H$ are illustrated in **figure 78** and **figure 79**.

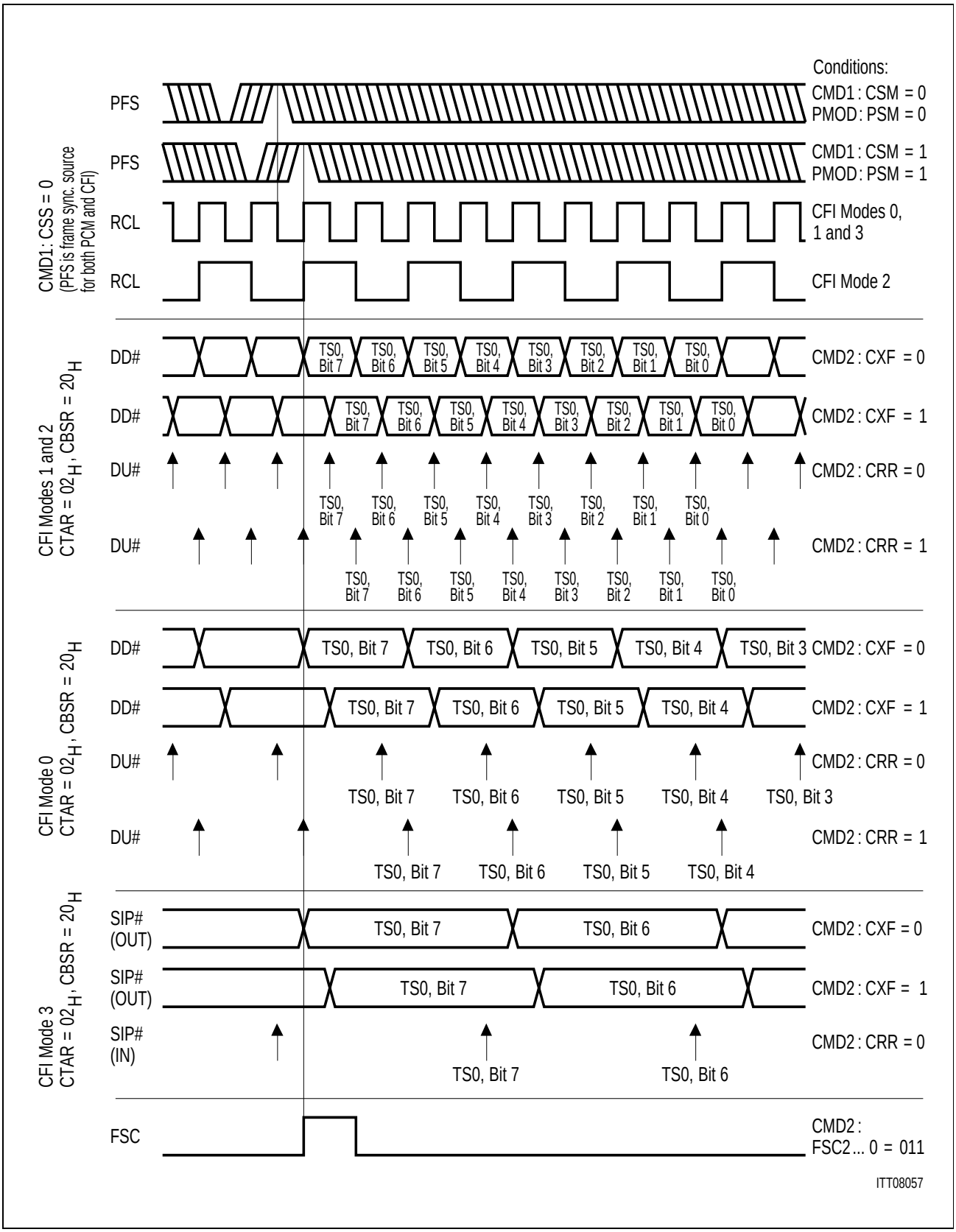



**Figure 78**
**CFI Bit Timing with Respect to the Framing Signal PFS (CMD1:CSS = 0)**

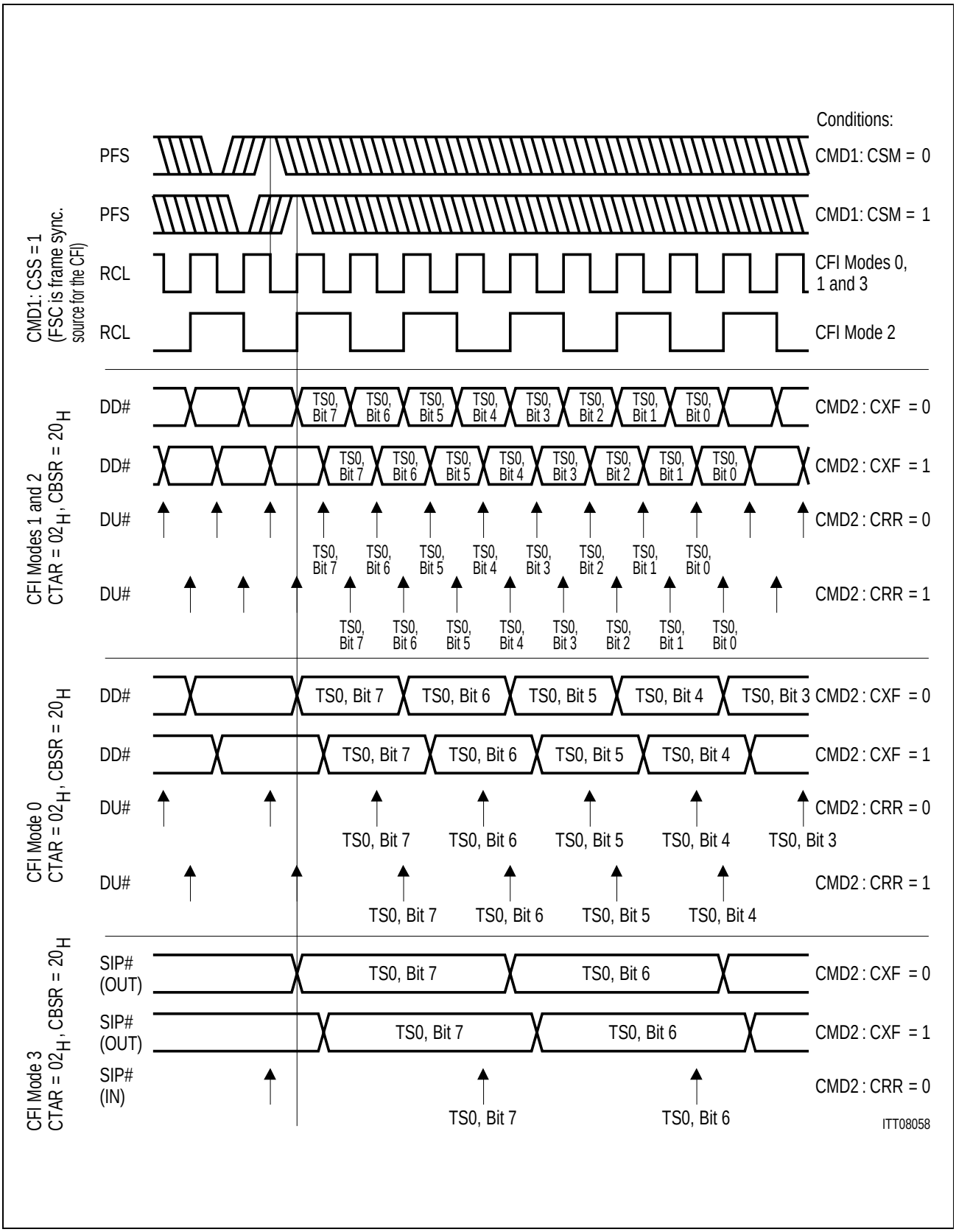



**Figure 79**
**CFI Bit Timing with Respect to the Framing Signal FSC (CMD1:CSS = 1)**

### Examples

1) In CFI mode 0, with a frame consisting of 32 timeslots, the following timing relationship between the framing signal source PFS and the data signals is required:

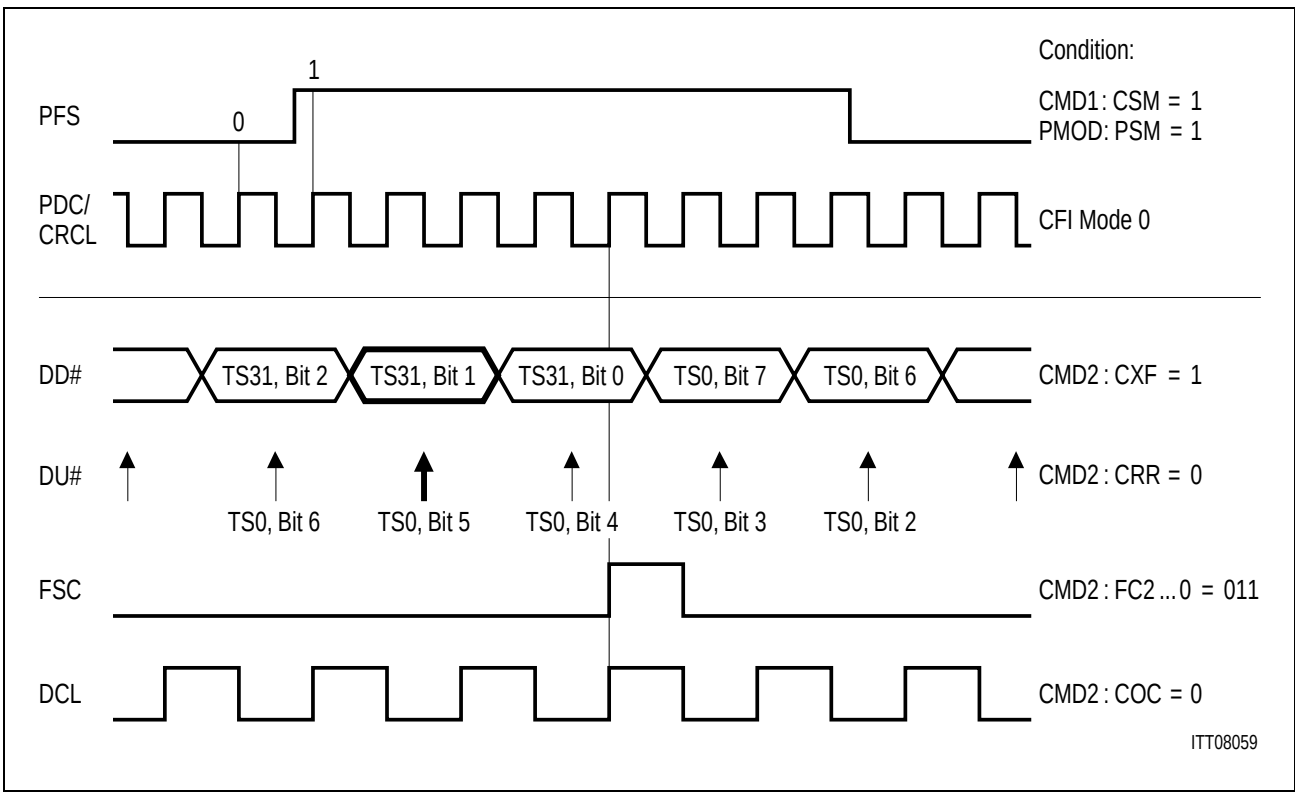



**Figure 80**
**Timing Signals for CFI Bit Shift Example 1**

The framing signal source PFS shall mark CFI timeslot 31, bit 1 in downstream direction and CFI timeslot 0, bit 5 in upstream direction. The data shall be transmitted and sampled with the falling CRCL edge. The timing of the FSC and DCL output signals shall be as shown in **figure 80**. The PFS signal is sampled with the rising PDC edge.

The following CFI register values result:

Since PFS marks the downstream bit 1, the CBSR:CDS bits must be set to ‘000’, according to **table 34**.

If the CBSR:CDS bits are set to ‘000’, PFS marks the timeslot TSN – 1, according to **table 34**.

PFS shall mark CFI timeslot 31, i.e. TSN – 1 = 31, or
$TSN = 31 + 1 = (32)_{mod\ 32} = 0$

From this it follows that:

$CTAR:TSN6 \ldots 0 = TSN + 2 = 0 + 2 = 2_D = 0000010_B$; i.e. $CTAR = 02_H$

The upstream CFI frame shall be shifted by 4 bits to the left (TS31, bit 1 + 4 bits yields in TS0, bit 5).

The CBSR:CUS bits must therefore be set to ‘0100’, according to **figure 77**.

The complete value for CBSR is: CBSR = $04_H$

Finally, the CMD2 register bits must be set to

FC2 … 0 = 011, COC = 0, CXF = 1, CRR = 0, CBN9 … 8 = 00, i.e. CMD2 = $68_H$

2) In CFI mode 0, with a frame consisting of 32 timeslots, the following timing relationship between the framing signal source FSC and the data signals is required:

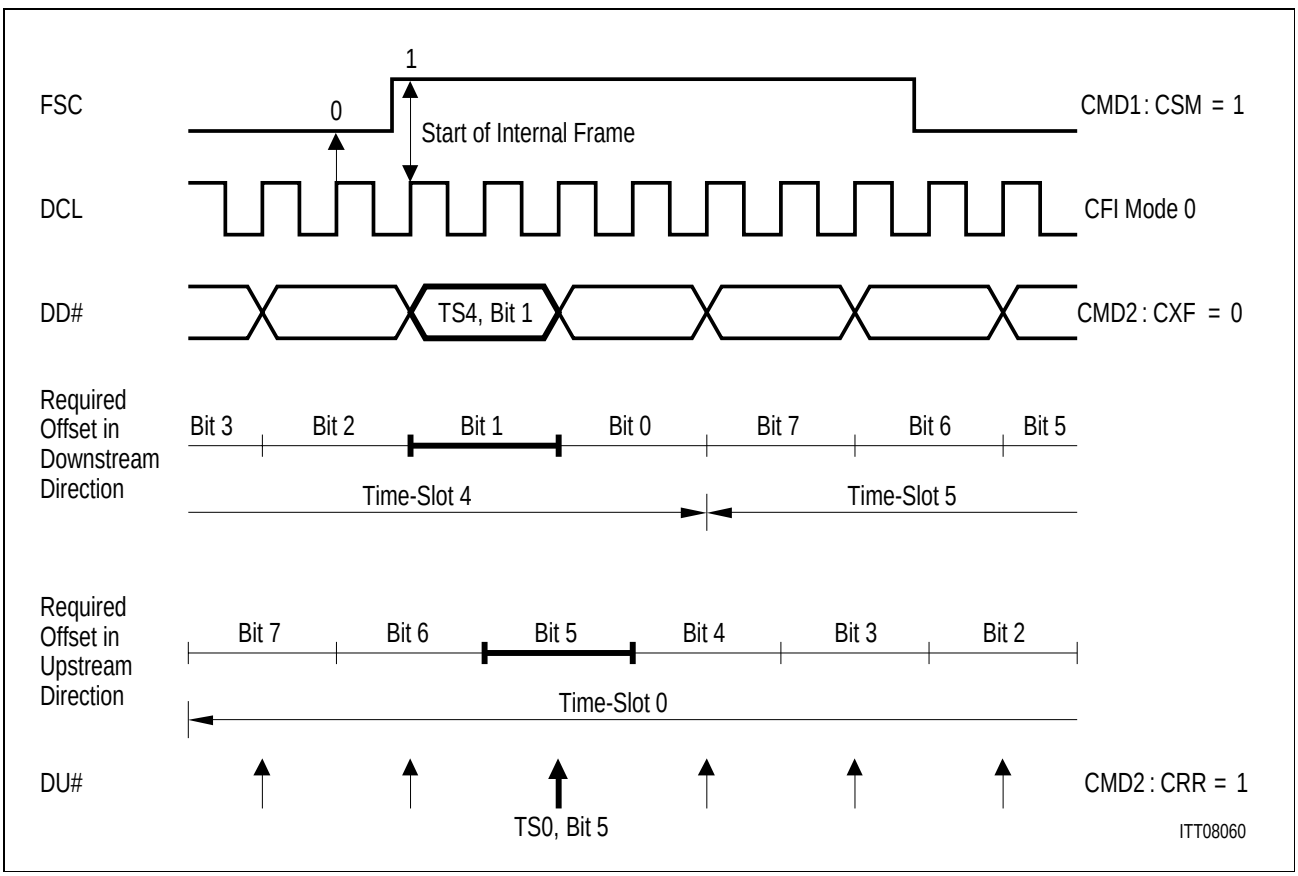



**Figure 81**
**Timing Signals for CFI Bit Shift Example 2**

The framing signal source FSC shall mark CFI timeslot 4, bit 1 in downstream direction and CFI timeslot 0, bit 5 in upstream direction. The data shall be transmitted with the rising CRCL edge and sampled with the rising CRCL edge. The FSC signal shall be sampled with the rising DCL edge.

The following CFI register values result:

Since FSC marks the downstream bit 1, the CBSR:CDS bits must be set to ‘000’, according to **table 34**.

If the CBSR:CDS bits are set to ‘000’, FSC marks the timeslot TSN – 1, according to **table 34**.

FSC shall mark CFI timeslot 4, i.e. TSN – 1 = 4, or TSN = 4 + 1 = 5

From this it follows that:

CTAR:TSN6 … 0 = TSN + 2 = 5 + 2 = $7_D$ = $0000111_B$; i.e. CTAR = $07_H$

The upstream CFI frame shall be shifted by 28 bits to the right (ts 4, bit 1 - 28 bits yields in TS0, bit 5)

Since it is not possible to shift the upstream frame with respect to the downstream frame by more than 15 bits when using the CBSR:CUS bits, the following trick must be used:

The CBSR:CUS bits are set to '0100' to shift the frame by 4 bits to the left. The remaining shift to the right of 28 + 4 = 32 bits (equivalent to 4 timeslots) can now be performed by renumbering the upstream CFI timeslots in the software. This results in an offset of 4 timeslots when addressing a CFI timeslot via the Control Memory (CM):

If CFI timeslot N shall be switched (N refers to the external timeslot numbering), the CM must be written with the CFI address $(N + 4)_{mod\ 32}$.

If for example the upstream CFI timeslot 0 of port 0 shall be switched to a PCM timeslot, the CM address $88_H$ (CFI p 0, TS4) must be used.

The complete value for CBSR is: CBSR = $04_H$

Finally the CMD2 register bits must be set to

FC2 … 0 = XXX, COC = X, CXF = 0, CRR = 1, CBN9 … 8 = 00, i.e.: CMD2 = $04_H$

**CFI Receive Line Selection CMD1:CIS1 … CIS0**

The CFI transmit line of a given logical port (as it is used for programming the switching function) is always assigned to a dedicated physical transmit pin, e.g. in CFI mode 1, pin DD1 carries the CFI data of logical port 1.

In receive direction however, an assignment between logical and physical ports can be made in CFI modes 1 and 2. This selection is programmed via the alternative input selection bits 1 and 0 (CIS1, CIS0) in the CMD1 register.

In CFI mode 0 and 3, CIS1 and CIS0 should both be set to 0.

In CFI mode 1, CIS0 selects between receive lines DU0 and DU2 for logical port 0 and CIS1 between the receive lines DU1 and DU3 for logical port 1.

In CFI mode 2, CIS0 selects between the receive lines DU0 and DU2, CIS1 should be set to 0.

**Table 35** shows the function taken over by each of the CFI interface pins, depending on the CFI mode and the values programmed to CIS1 and CIS0.

**Table 35**
**CFI Pin Configuration**

| CFI Mode | Port 0 | | Port 1 | | Port 2 | | Port 3 | |
|---|---|---|---|---|---|---|---|---|
| | DU0 | DD0 | DU1 | DD1 | DU2 | DD2 | DU3 | DD3 |
| 0 | IN0 | OUT0 | IN1 | OUT1 | IN2 | OUT2 | IN3 | OUT3 |
| 1 | IN0<br>CIS0 = 0 | OUT0 | IN1<br>CIS1 = 0 | OUT1 | IN0<br>CIS0 = 1 | high Z | IN1<br>CIS1 = 1 | high Z |
| 2 | IN<br>CIS0 = 0 | OUT | – | high Z | IN<br>CIS0 = 1 | high Z | – | high Z |
| 3 | I/O4 | I/O0 | I/O5 | I/O1 | I/O6 | I/O2 | I/O7 | I/O3 |

**Figure 82** shows the correlation between physical and logical CFI ports in CFI modes 0, 1, 2 and 3:

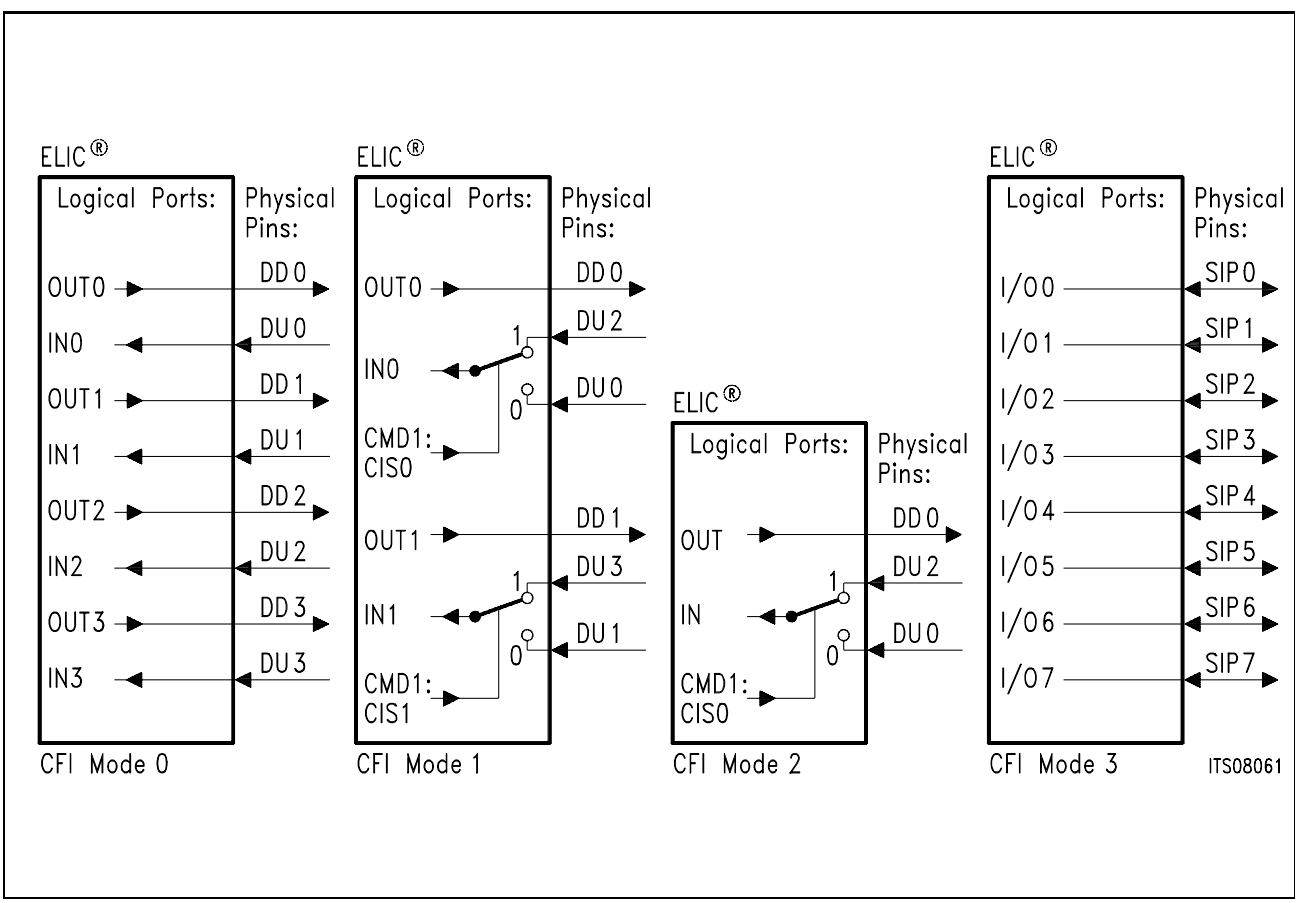



**Figure 82**
**Correlation Between Physical and Logical CFI Ports**

### CFI Sub-Timeslot Position CSCR

If a timeslot assignment is programmed in the control memory (CM), the used control memory code defines the channel bandwidth and the subchannel position at the PCM interface (refer to **chapter 5.4.2**). The subchannel position at the configurable interface however is defined on a per port basis in the **C**onfigurable interface **S**ub**C**hannel **R**egister CSCR.

The subchannel control bits SC#1 … SC#0 specify separately for each logical port the bit positions to be exchanged with the data memory (DM) when a connection with a channel bandwidth as defined by the CM code has been established:

**Table 36**
**Subchannel Positions at the CFI**

| SC#1 | SC#0 | Bit Positions for CFI Subchannels Having a Bandwidth of | | |
|---|---|---|---|---|
| | | 64 kBit/s | 32 kBit/s | 16 kBit/s |
| 0 | 0 | 7 … 0 | 7 … 4 | 7 … 6 |
| 0 | 1 | 7 … 0 | 3 … 0 | 5 … 4 |
| 1 | 0 | 7 … 0 | 7 … 4 | 3 … 2 |
| 1 | 1 | 7 … 0 | 3 … 0 | 1 … 0 |

**Table 37** shows the effect of the different subchannel control bits SC#1 … SC#0 on the CFI ports in each CFI mode:

**Table 37**
**Correlation between the Subchannel Cntrol Bits and the CFI Ports**

| SC#1 | SC#0 | CFI Mode | | | |
|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 |
| SC01 | SC00 | port 0 | port 0 | port | ports 0 and 4 |
| SC11 | SC10 | port 1 | port 1 | see note | ports 1 and 5 |
| SC21 | SC20 | port 2 | see note | see note | ports 2 and 6 |
| SC31 | SC30 | port 3 | see note | see note | ports 3 and 7 |

*Note: In CFI mode 1:SC21 = SC01; SC20 = SC00; SC31 = SC11; SC30 = SC10*
*In CFI mode 2:SC31 = SC21 = SC11 = SC01; SC30 = SC20 = SC10 = SC00*

If for example at CFI port 1 a 16 kBit/s channel shall be switched to (or from) a CFI bit position 5 … 4 from (or to) any 2 bit sub-timeslot position at the PCM interface, a CM code defining a channel bandwidth of 16 kBit/s and defining the subchannel position at the PCM interface must be written to the CM code field of the involved 8 bit CFI timeslot (i.e. 0111, 0110, 0101 or 0100). In order to insert (or extract) bit positions 5 … 4 of the selected 8 bit CFI timeslot, SC11 … SC10 have to be set to 01. Once fixed to this value, all timeslot connections programmed on CFI port 1 are performed on bits 7 … 0 for

64 kBit/s channels, bits 3 … 0 for 32 kBit/s channels and bits 5 … 4 for 16 kBit/s channels.

Since for each CFI timeslot there is only one control memory location, only one subchannel may be mapped to each CFI timeslot. The remaining bits of such a partly unused CFI timeslot are inactive e.g. set to high impedance if OMDR:COS = 0.

Note that if an odd numbered CFI timeslot is initialized as an IOM channel with switched D channel, SC#1 … SC#0 must be set to '00' because the D channel is located at bits 7 … 6. In this case the remaining bits can still be used for C/I and monitor channel applications (refer to **chapter 5.5**).

For more detailed information on subchannel switching refer to **chapter 5.4.2**.

### CFI Standby Mode OMDR:CSB

In standby mode (OMDR:CSB = 0), the CFI output ports are set to high impedance and the clock signals DCL and FSC, if programmed as outputs (CMD1:CSS = 0), are switched off.

Note that the internal operation of the ELIC is not affected in standby mode, i.e. the received CFI data is still read in and may still be processed by the ELIC (switched to PCM or µP, etc.)

In operational mode (OMDR:CSB = 1), the CFI output pins take over the function programmed in the control memory and DCL and FSC deliver clock and framing output signals (if CMD1:CSS = 0) as programmed in CMD1 and CMD2.

### CFI Output Driver Selection OMDR:COS

The output drivers at the configurable interface (DD# or I/**O**#) can be programmed as open drain or tristate drivers.

If programmed as open drain drivers (OMDR:COS = 1), external pull-up resistors (connected to $V_{DD}$) are required in order to pull the output line to a high level if a logical 1 is being transmitted. For unassigned channels (e.g. control memory code '0000') the ELIC transmits a logical 1. The maximum output current at a low voltage level of 0.45 V is 7 mA, pull-up resistors down to 680 Ω can thus be used.

If programmed as tristate drivers (OMDR:COS = 0), logical 0s and 1s are transmitted with push-pull output drivers, whereas unassigned channels are set to high impedance.

## 5.3 Data and Control Memories

### 5.3.1 Memory Structure

The ELIC memory is composed of the **Control Memory (CM)** and the **Data Memory (DM)**. Their structure is shown in **figure 83**.

The **control memory** refers to the Configurable Interface (CFI) such that for each CFI timeslot and for each direction (upstream and downstream) there is a 4 bit code field and an 8 bit data field location.

The code field defines the function of the corresponding CFI timeslot. A timeslot, may for example, be transparently switched through to the PCM interface (switched channel) or it may serve as monitor, feature control, command/indication or signaling channel in an IOM or SLD application (preprocessed channel) or it may be directly switched to the µP interface (µP channel).

The use of the data field depends on the function defined by the code field. If a CFI timeslot is defined as a switched channel, the data field is interpreted as a pointer to the data memory and defines therefore to which PCM timeslot the connection shall be made. For preprocessed channels, the data field serves as a buffer for the command/indication or signaling value. If a µP channel is programmed, the data field content is directly exchanged with the CFI timeslot.

The **data memory** refers to the PCM interface such that for each upstream timeslot there is a 4 bit code field and an 8 bit data field location, whereas for each downstream timeslot there is only an 8 bit data field location.

The data field locations buffer the PCM data transmitted and received over the PCM interface. The code field (tristate field) defines whether the upstream data field contents should be transmitted in the associated PCM timeslot or whether the timeslot should be switched to high impedance.

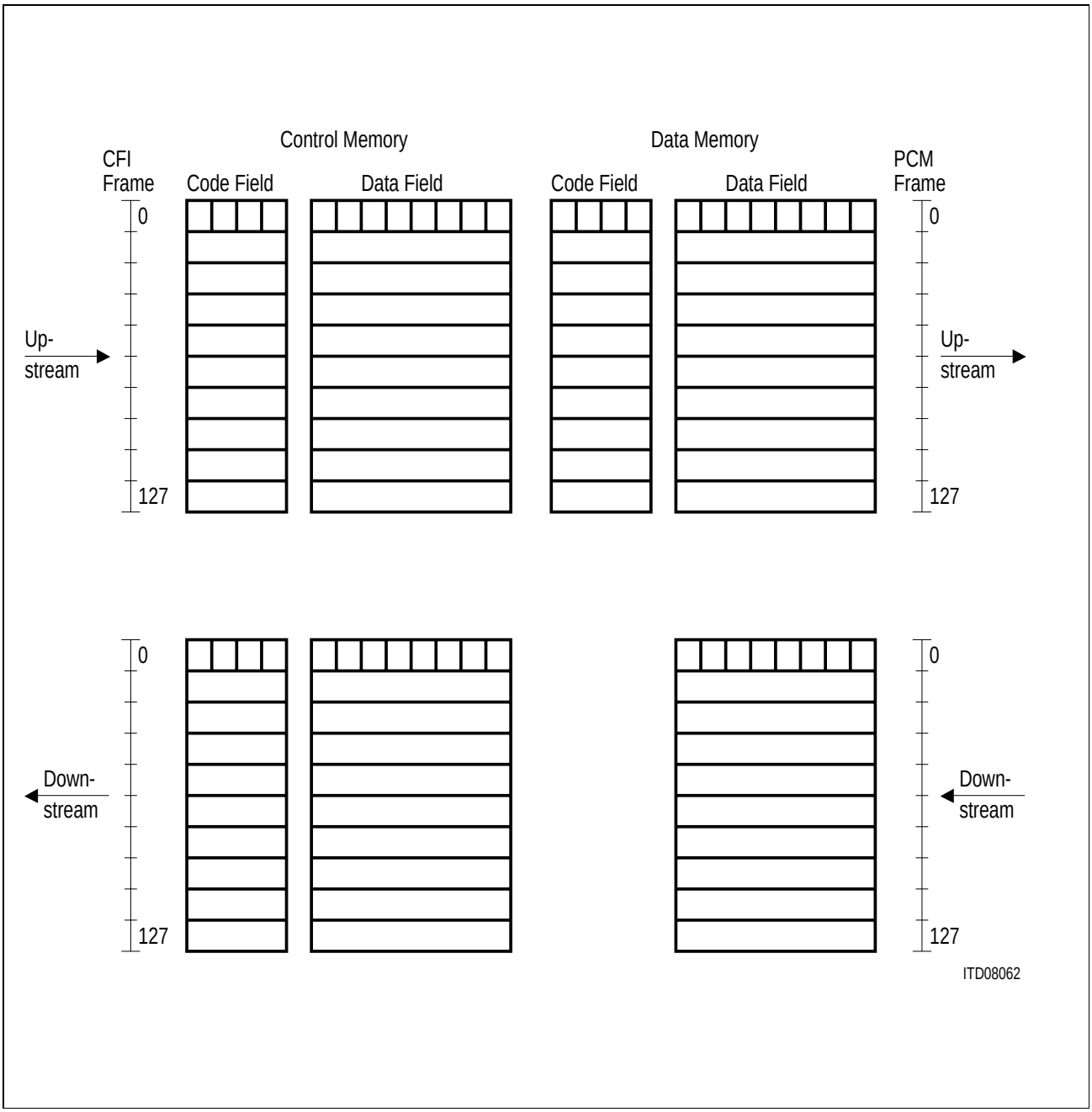



**Figure 83**
**ELIC® Memory Structure**

### 5.3.2 Indirect Register Access

The control and data memories must be accessed by the µP in order to initialize the CFI and PCM interfaces for the required functionality, to program timeslot assignments, to access the control/signaling channels (IOM/SLD), etc.

This access is performed through indirect addressing using the **memory access registers MADR, MAAR, and MACR**.

**Memory Access Data Register** read/write reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| MADR | MD7 | MD6 | MD5 | MD4 | MD3 | MD2 | MD1 | MD0 |

The **M**emory **A**ccess **D**ata **R**egister **MADR** contains the data to be transferred from or to a memory location. The meaning and the structure of this data depends on the kind of memory being accessed. If, for example, MADR contains a pointer to a PCM timeslot, the data must be encoded according to **figure 84**. If it contains a 4 bit C/I code the structure would for example be '11 C/I 11'. For accesses to 4 bit code fields only the 4 least significant bits of MADR are relevant.

**Memory Access Address Register** read/write reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| MAAR | U/D | MA6 | MA5 | MA4 | MA3 | MA2 | MA1 | MA0 |

The **M**emory **A**ccess **A**ddress **R**egister **MAAR** specifies the address of the memory access. This address encodes a CFI timeslot for control memory and a PCM timeslot for data memory accesses. Bit 7 of MAAR (U/D bit) selects between upstream and downstream memory blocks.

Bits MA6 … 0 encode the CFI or PCM port and timeslot number according to **figure 84**.

**Memory Access Control Register** read/write reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| MACR | RWS | MOC3 | MOC2 | MOC1 | MOC3/ CMC3 | CMC2 | CMC1 | CMC0 |

The **M**emory **A**ccess **C**ontrol **R**egister **MACR** selects the type of memory (control or data memory), the type of field (data or code field) and the access mode (read or write) of the register access. When writing to the control memory code field, MACR also contains the 4 bit code (CMC3 … 0) defining the function of the addressed CFI timeslot.

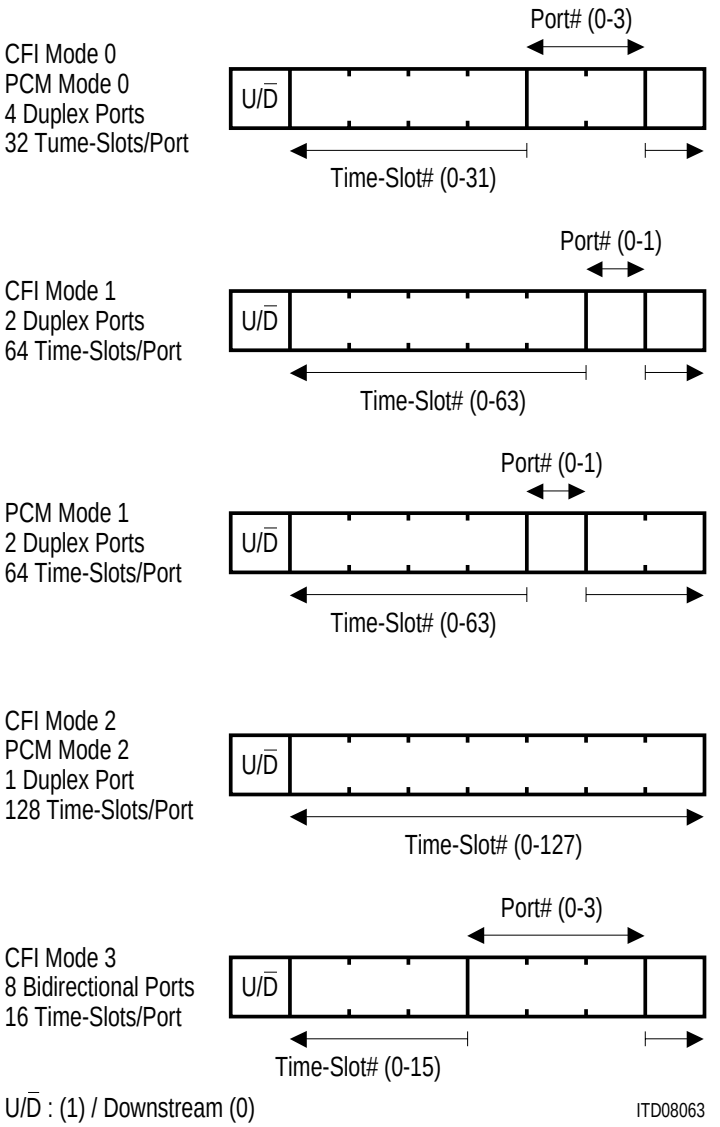



**Figure 84**
**Timeslot Encoding for the Different PCM and CFI Modes**

### Memory Access Time

Writing to MACR starts a memory write or read operation which takes a certain time. During this time no further memory accesses may be performed i.e. the MADR, MAAR, and MACR registers may not be written. The STAR:MAC bit indicates whether a memory operation is still in progress (MAC = 1) or already completed (MAC = 0) and should therefore be interrogated before each access.

Since memory operations must be synchronized to the ELIC internal bus which is clocked by the reference clock (RCL), the time required for an indirect register access can be given as a multiple of RCL clock cycles. A 'normal' access to a single memory location, for example, takes a maximum of 9.5 RCL cycles which is approximately 2.4 µs assuming a 4 MHz clock (e.g. CFI configured as standard IOM-2 interface).

### Memory Access Modes

Access to memory locations is furthermore influenced by the operation mode set via the Operation Mode Register OMDR. There are 4 modes which can be selected with the OMDR:OMS1, OMS0 bits:

**Operation Mode Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| OMDR | **OMS1** | **OMS0** | PSB | PTL | COS | MFPS | CSB | RBS |

- The **CM reset mode** (OMS1 … 0 = 00) is used to reset all locations of the control memory code and data fields with a single command within only 256 RCL cycles. A typical application is resetting the CM with the command MACR = $70_H$ which writes the contents of MADR ($XX_H$) to all data field locations and the code '0000' (unassigned channel) to all code field locations. A CM reset should be made after each hardware reset. In the CM reset mode the ELIC does not operate normally i.e. the CFI and PCM interfaces are not operational.
- The **CM initialization mode** (OMS1 … 0 = 10) allows fast programming of the Control Memory since each memory access takes a maximum of only 2.5 RCL cycles compared to the 9.5 RCL cycles in the normal mode. Accesses are performed on individual addresses specified by MAAR. The initialization of control/signaling channels in IOM or SLD applications can, for example, be carried out in this mode (see **chapter 5.5.1**). In the CM initialization mode the ELIC does also not work normally.
- In the **normal operation mode** (OMS1 … 0 = 11) the CFI and PCM interfaces are operational. Memory accesses performed on single addresses (specified by MAAR) take 9.5 RCL cycles. An initialization of the complete data memory tristate field takes 1035 RCL cycles.

– In **test mode** (OMS1 … 0 = 01) the ELIC sustains normal operation. However memory accesses are no longer performed on a specific address defined by MAAR, but on all locations of the selected memory, the contents of MAAR (including the U/D bit!) being ignored. This function can for example be used to program a PCM idle code to all PCM ports and timeslots with a single command.

### 5.3.3 Memory Access Commands

The memory access commands can be divided into the following four categories:

– Access to the Data Memory Data Field: µP access to PCM frame
– Access to the Data Memory Code Field: PCM tristate control
– Access to the Control Memory Data Field: timeslot assignment, µP access to CFI frame
– Access to the Control Memory Code Field: set-up of CFI timeslot functionality

In the following chapters, these commands are explained in more detail.

#### 5.3.3.1 Access to the Data Memory Data Field

The data memory (DM) data field buffers the PCM data transmitted (upstream block) and received (downstream block) via the PCM interface. Normally this data is switched transparently from or to the CFI and there is no need to access it from the µP interface. For some applications however it is useful to have a direct µP access to the PCM frame.

When an upstream PCM timeslot (or even sub-timeslot) is not switched from the CFI (unassigned channel), it is possible to write a fixed value to the corresponding DM data field location. This value will then be transmitted repeatedly in each PCM frame without further µP interaction (PCM idle code). If instead a continuous pattern should be sent, the write access can additionally be synchronized to the frame by means of synchronous transfer interrupts (see **chapter 5.7**).

Writing to an upstream DM data field location can also be restricted to a 2 or 4 bit sub-timeslot. It is thus possible to have certain sub-timeslots of the same 8 bit timeslot switched from the CFI with the other sub-timeslots containing a PCM idle code. This restriction is made via the Memory Operation Code (refer to **table 38**).

For test purposes the upstream DM data field contents can also be read back.

The downstream DM data field cannot be written to, it can only be read. Reading such a location reflects the PCM data contained in the received PCM frame regardless of a connection to the CFI having been established or not. The µP can thus determine the contents of received PCM timeslots simply by reading the corresponding downstream DM locations. This reading can, if required, also be synchronized to the frame by means of synchronous transfer interrupts.

**The Procedure for Writing to the Upstream DM Data Field is**

W:MADR = value to be transmitted in the PCM (sub)timeslot

W:MAAR = address of the desired (upstream)[1)] PCM timeslot encoded according to **figure 84**

| | bit 7 | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| W:MACR = | 0 | MOC3 | MOC2 | MOC1 | MOC0 | 0 | 0 | 0 |

MOC3 … 0 defines the bandwidth and the position of the subchannel according to **table 38**.

**Table 38**
**Memory Operation Codes for Accesses to the DM Data Field**

| MOC3 … 0 | Transferred Bits | Channel Bandwidth |
|---|---|---|
| 0000 | – | – |
| 0001 | bits 7 … 0 | 64 kBit/s |
| 0011 | bits 7 … 4 | 32 kBit/s |
| 0010 | bits 3 … 0 | 32 kBit/s |
| 0111 | bits 7 … 6 | 16 kBit/s |
| 0110 | bits 5 … 4 | 16 kBit/s |
| 0101 | bits 3 … 2 | 16 kBit/s |
| 0100 | bits 1 … 0 | 16 kBit/s |

**The Procedure for Reading the DM Data Field is**

W:MAAR = address of the desired PCM timeslot encoded according to **figure 84**
W:MACR = 1000 $0000_B$ = $80_H$[2)]
wait for STAR:MAC = 0
R:MADR = value

**Figure 85** illustrates the access to the Data Memory Data Field.

1 The U/D bit of MAAR will implicitly be set to 1.
2 When reading a DM data field location, all 8 bits are read regardless of the bandwidth selected by the MOC bits.

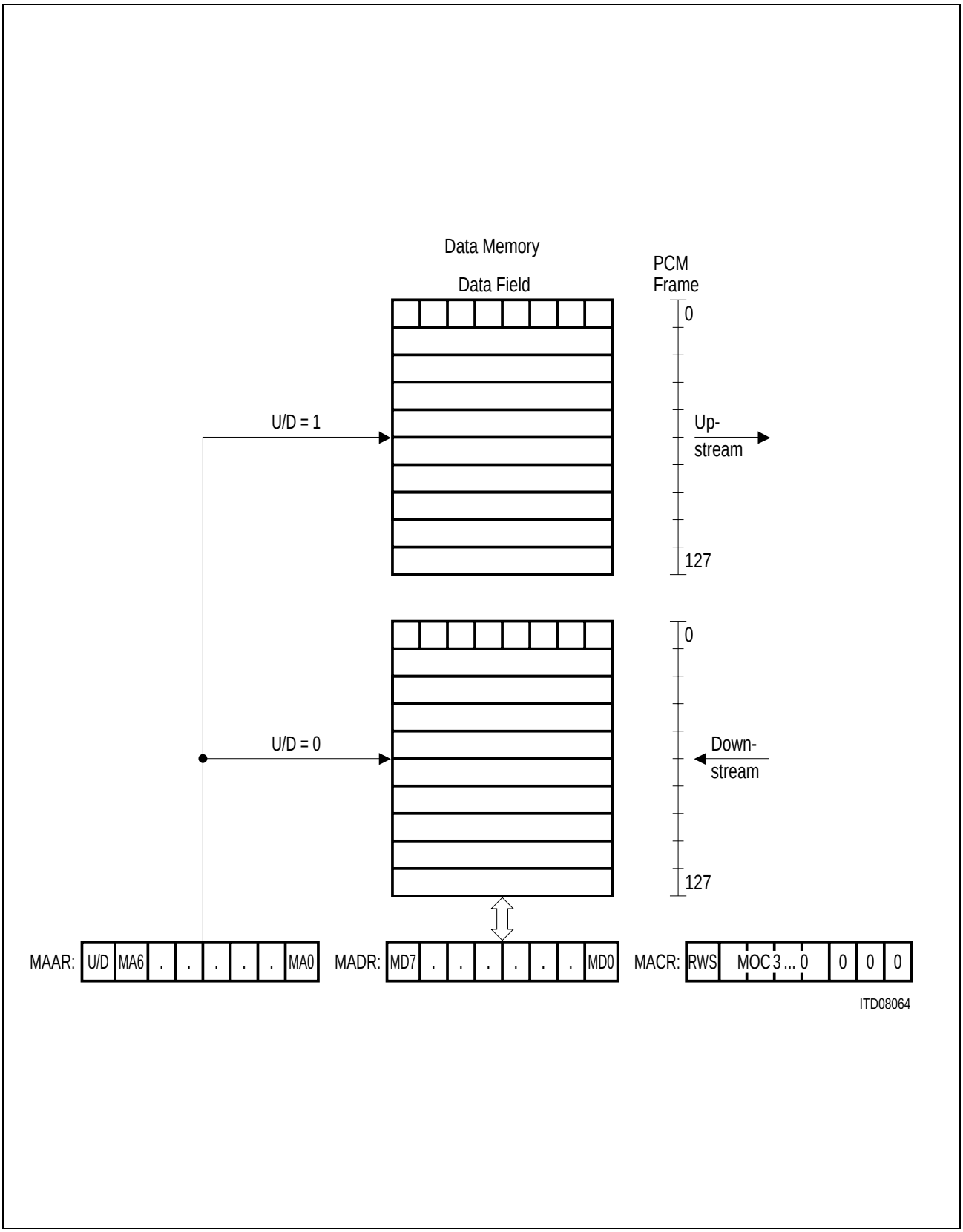



**Figure 85**
**Access to the Data Memory Data Field**

### Examples

In PCM mode 0 the idle code ‘1010 $0101_B$’ shall be transmitted in timeslot 16 of port 0:

W:MADR = $1010\ 0101_B$ ; idle code
W:MAAR = $1100\ 0000_B$ ; address of upstream PCM timeslot 16 of port 0 according to **figure 84**
W:MACR = $0000\ 1000_B$ : write access, MOC code '0001'

The idle code can, of course, only be transmitted on the TxD# line if the corresponding tristate bits are enabled (refer to **chapter 5.3.3.2**):

W:MADR = $XXXX\ 1111_B$ ; all 8 bits of addressed timeslot to low impedance
W:MAAR = $1100\ 0000_B$ ; address of upstream PCM timeslot 16 of port 0 according to **figure 84**
W:MACR = $0110\ 0000_B$ : write access, MOC code ‘1100’

For test purposes the idle code can also be read back:

W:MAAR = $1100\ 0000_B$ ; address of upstream PCM timeslot 16 of port 0 according to **figure 84**
W:MACR = $10XX\ X000_B$ : read access, MOC code '0XXX'
wait for STAR:MAC = 0
R:MADR = $1010\ 0101_B$ ; idle code

In PCM mode 2 the idle pattern ‘0110’ shall be transmitted in bit positions 3 … 0 of timeslot 63, bits 7 … 4 shall be tristated:

W:MADR = $XXXX\ 0110_B$ ; idle code
W:MAAR = $1011\ 1111_B$ ; address of upstream PCM timeslot 63 according to **figure 84**
W:MACR = $0001\ 0000_B$ ; write access, MOC code ‘0010’

Programming of the desired tristate functions:

W:MADR = $XXXX\ 0011_B$ ; bits 7 … 4 to high impedance, bits 3 … 0 to low impedance
W:MAAR = $1011\ 1111_B$ ; address of upstream PCM timeslot 63 according to **figure 84**
W:MACR = $0110\ 0000_B$ ; write access, MOC code ‘1100’

### 5.3.3.2 Access to the Data Memory Code (Tristate) Field

The data memory code field exists only for the upstream DM block and is also called the PCM tristate field. Each (sub)timeslot of each PCM transmit port can be individually tristated via these code field locations.

If a (sub)timeslot is set to low impedance, the contents of the corresponding DM data field location is transmitted with a push-pull driver onto the transmit port TxD# and the tristate control line $\overline{\text{TSC\#}}$ is pulled low for the duration of that (sub)timeslot.

If a (sub)timeslot is set to high impedance, the transmit port TxD# will be tristated and the $\overline{\text{TSC\#}}$ line is pulled high for the duration of that (sub)timeslot.

There are 4 code bits for selecting the tristate function of each 8 bit timeslot i.e. 1 control bit for each 16 kBit/s (2 bits) sub-timeslot. If a control bit is set to 1, the corresponding sub-timeslot is set to low impedance, if it is set to 0 the sub-timeslot is tristated.

**Figure 86** illustrates this behavior.

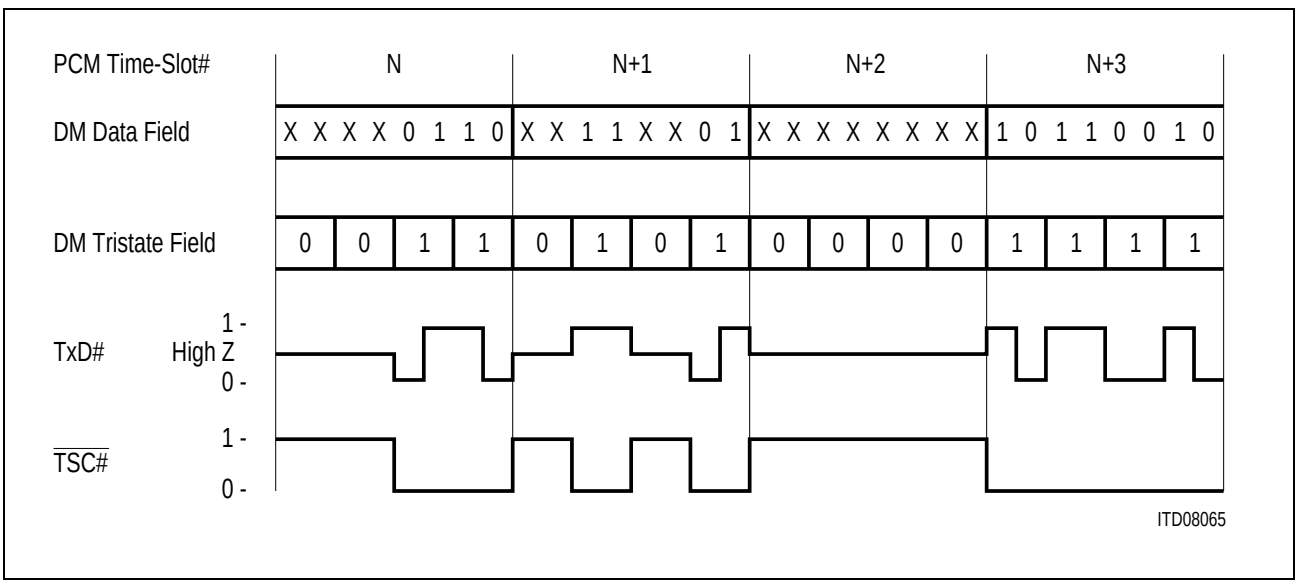



**Figure 86**
**Tristate Control at the PCM Interface**

The tristate field can be written to and, for test purposes, also be read back.

There are two commands (Memory Operation Codes) for accessing the tristate field:

With the **“Single Channel Tristate Control”** command (MOC3 … 0 = 1100) the tristate field of a single PCM timeslot can be written to and also read back. The 4 least significant bits of MADR are exchanged with the code field of the timeslot selected by the MAAR register.

With the **“Tristate Control Reset”** command (MOC3 … 0 = 1101) the tristate field of all PCM timeslots can be written to with a single command. The 4 bits of MADR are then copied to all code field locations regardless of the address programmed to MAAR. Such a complete access to the DM tristate field takes 1035 RCL cycles.

The MADR bits MD7 … MD0 control the PCM timeslot bit positions 7 … 0 in the following way:

MD7 … MD4 are not used (don't care);
MD3 … MD0 select between the states high impedance (MD# = 0) or low impedance (MD# = 1)

| Timeslot Bit Position: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| MADR Bits: | MD3 | | MD2 | | MD1 | | MD0 | |

**The Procedure for Writing to a Single PCM Tristate Field is**

W:MADR = X X X X MD3 MD2 MD1 $MD0_B$
W:MAAR = address of the desired (upstream)[1)] PCM timeslot according to **figure 84**
W:MACR = 0110 $000_B = 60_H$

**The Procedure for Reading Back a (Single) PCM Tristate Field Location is**

W:MAAR = address of the desired (upstream)[1)] PCM timeslot according to **figure 84**
W:MACR = $E0_H$
wait for STAR:MAC = 0
R:MADR = X X X X MD3 MD2 MD1 $MD0_B$

**The Procedure for Writing to all PCM Tristate Field Positions is**

W:MADR = X X X X MD3 MD2 MD1 $MD0_B$
W:MACR = 0110 $1000_B = 68_H$

1 The U/D bit of MAAR will implicitly be set to 1.

**Figure 87** illustrates the access to the tristate field:

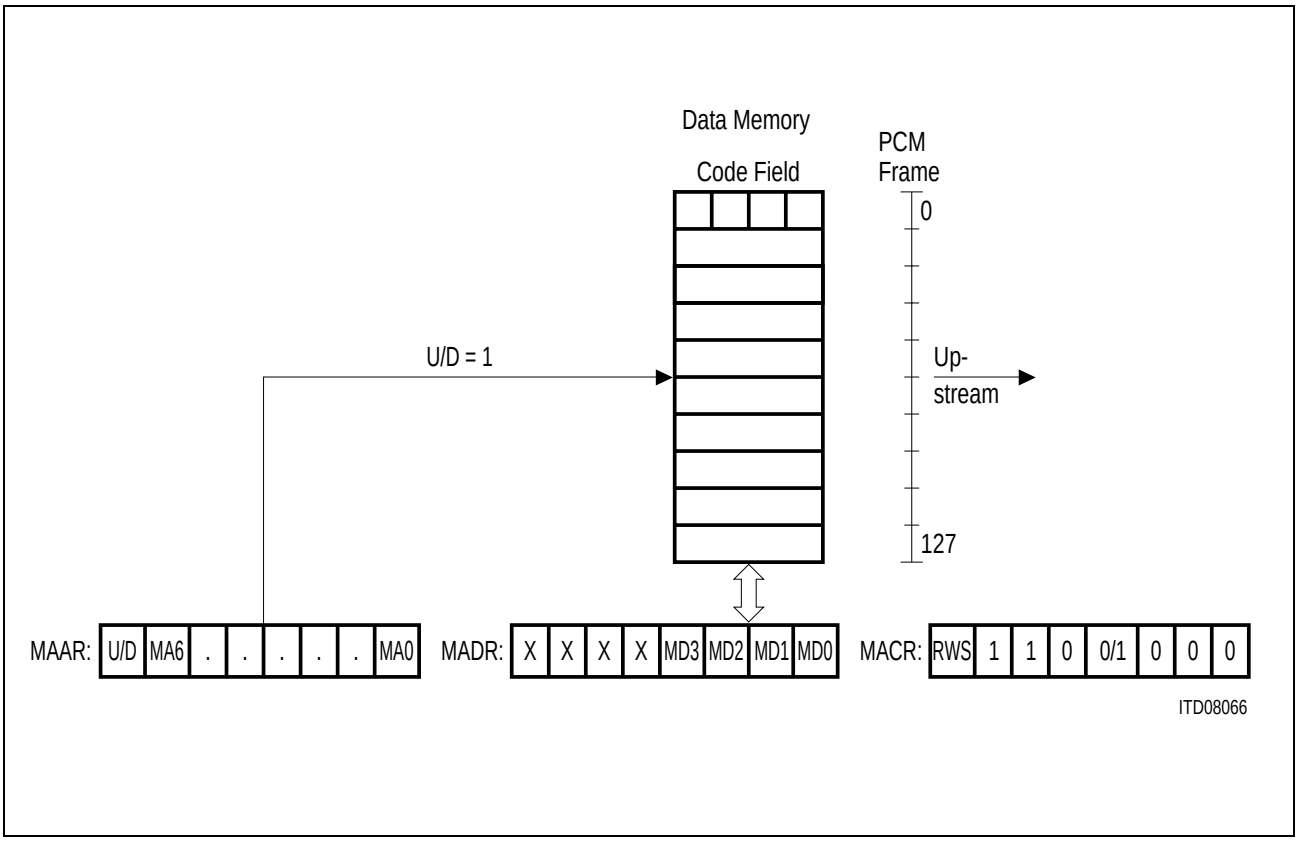



**Figure 87**
**Access to the Data Memory Code (Tristate) Field**

### Examples

All PCM timeslots shall be set to high impedance (disabled):

W:MADR = $00_H$ ; all bits to high impedance
W:MACR = $68_H$ ; write access with MOC = 1101

All PCM timeslots shall be set to low impedance (enabled):

W:MADR = $FF_H$ ; all bits to low impedance
W:MACR = $68_H$ ; write access with MOC = 1101

In PCM mode 1, bits 7 … 6 and 1 … 0 of PCM port 1, timeslot 10 shall be set to low impedance, bits 5 … 2 to high impedance:

W:MADR = $0000\ 1001_B$ ; bits 7 … 6 and 1 … 0 to low impedance, bits 5 … 2 to high impedance
W:MAAR = $1010\ 1010_B$ ; address of upstream PCM port 1, timeslot 10 according to **figure 84**
W:MACR = $0110\ 0000_B$ ; write access with MOC = 1100

For test purposes this setting shall be read back:

W:MAAR = 1010 $1010_B$ ; address of upstream PCM port 1, timeslot 10 according to **figure 84**
W:MACR = 1110 $0000_B$ = $E0_H$; read access with MOC = 1100
wait for STAR:MAC = 0
R:MADR = XXXX $1001_B$ ; read back of MD3 … 0

### 5.3.3.3 Access to the Control Memory Data Field

Writing to or reading the control memory (CM) data field may serve different purposes depending on the function given to the corresponding CFI timeslot which is defined by the 4 bit code field value:

**Table 39**
**CFI Timeslot Applications**

| CFI Timeslot Application | Meaning of CM Data Field |
|---|---|
| Switched channel | Pointer to PCM interface |
| Preprocessed channel | C/I or SIG value |
| µP channel | CFI idle code |

There are two types of commands which give access to the CM data field:

The memory operation code MACR:MOC = 111X is used for writing to the CM data field and code field simultaneously. The MADR content is transferred to the data field while the MACR:CMC3 … 0 bits are transferred to the code field. This command is explained in more detail in **chapter 5.3.3.4**.

The memory operation code MACR:MOC = 1001 is used for reading or writing to the CM data field. Since the CM code field is not affected, this command makes only sense if the related CFI timeslot has already the desired functionality.

**The Procedure for Writing to the CM Data Field (using the MOC = 1001 command) is**

W:MADR = value
W:MADR = CFI timeslot address according **figure 84**
R:MADR = 0100 $1000_B$ = $48_H$

**The Procedure for Reading the CM Data Field is**

W:MAAR = CFI timeslot address according **figure 84**
W:MACR = 1100 $1000_B$ = $C8_H$
wait for STAR:MAC = 0
R:MADR = value

**Figure 88** illustrates this behavior.

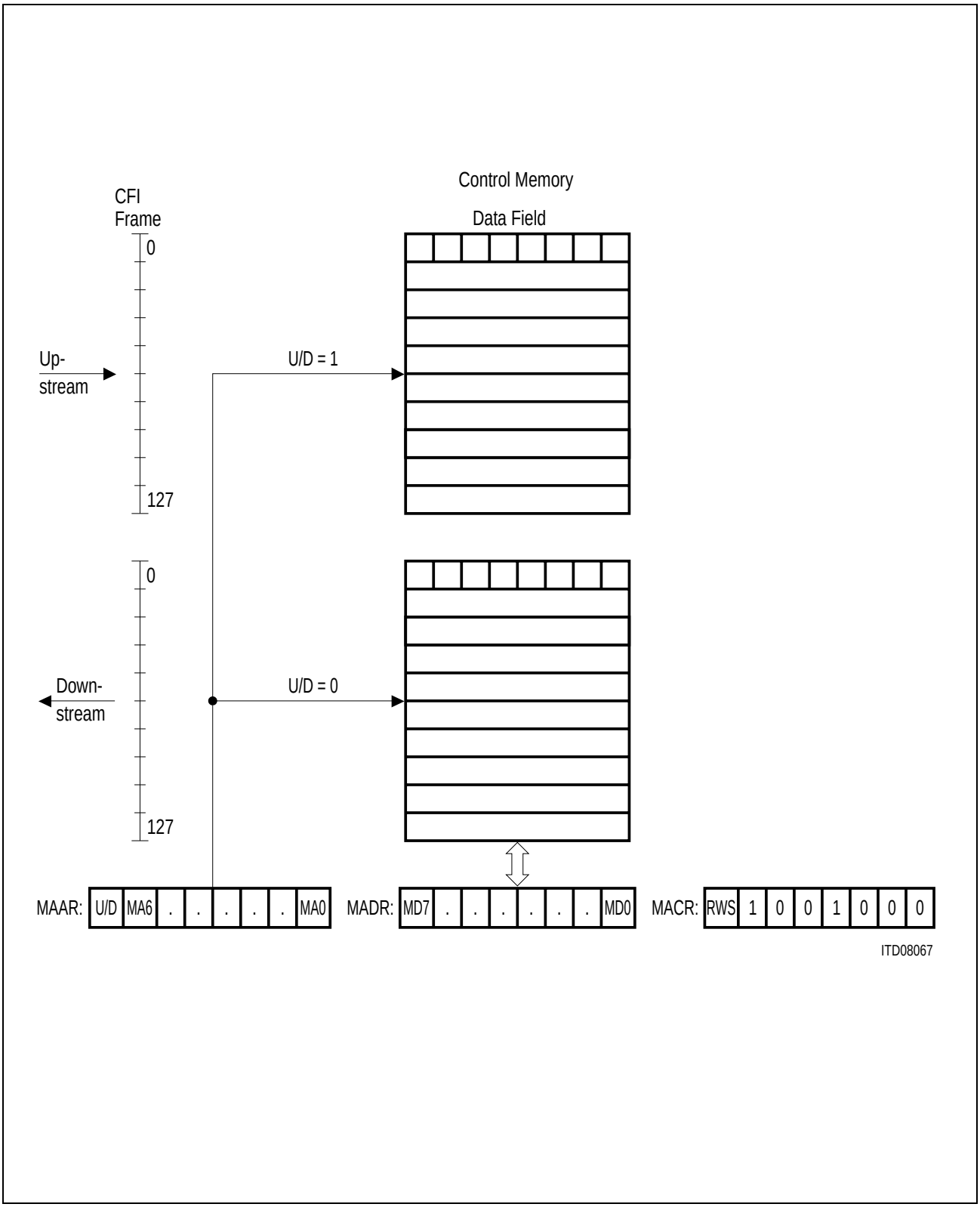



**Figure 88**
**Access to the Control Memory Data Field**

**Examples**

In CFI mode 2, CFI timeslot 123 has been initialized as a switched channel. The CM data field value therefore represents a pointer to the PCM interface.
In a first step, the involved upstream and downstream PCM timeslots shall be determined:

W:MAAR = $1111\ 1011_B$ ; address of upstream CFI timeslot 123
W:MACR = $1100\ 1000_B$ ; read back command
wait for STAR:MAC = 0
R:MADR = value ; encoded according **figure 84**

W:MAAR = $0111\ 1011_B$ ; address of downstream CFI timeslot 123
W:MACR = $1100\ 1000_B$ ; read back command
wait for STAR:MAC = 0
R:MADR = value ; encoded according **figure 84**

In the next step a new timeslot assignment (to PCM port 1, timeslot 34, PCM mode 1) shall be made for the upstream connection:

W:MADR = $1100\ 0110_B$ ; upstream PCM timeslot 34, port 1
W:MAAR = $1111\ 1011_B$ ; address of upstream CFI timeslot 123
W:MACR = $0100\ 1000_B$ ; write command

### 5.3.3.4 Access to the Control Memory Code Field

The 4 bit code field of the control memory (CM) defines the functionality of a CFI timeslot and thus the meaning of the corresponding data field.

There are codes for switching applications, preprocessed applications and for direct µP access applications (see **table 40**).

This 4 bit code, written to the MACR:CMC3 … 0 bit positions, will be transferred to the CM code field by selecting MACR:MOC = 111X. The 8 bit MADR value is at the same time transferred to the CM data field.

**The Procedure for Writing to the CM Code and Data Fields with a Single Command is**

W:MADR = value for data field
W:MAAR = CFI timeslot address encoded according to **figure 84**

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| W:MACR = | 0 | 1 | 1 | 1 | CMC3 | CMC2 | CMC1 | CMC0 |

CMC3 … 0 CM code, refer to **table 40**

**Figure 89** illustrates this behavior.

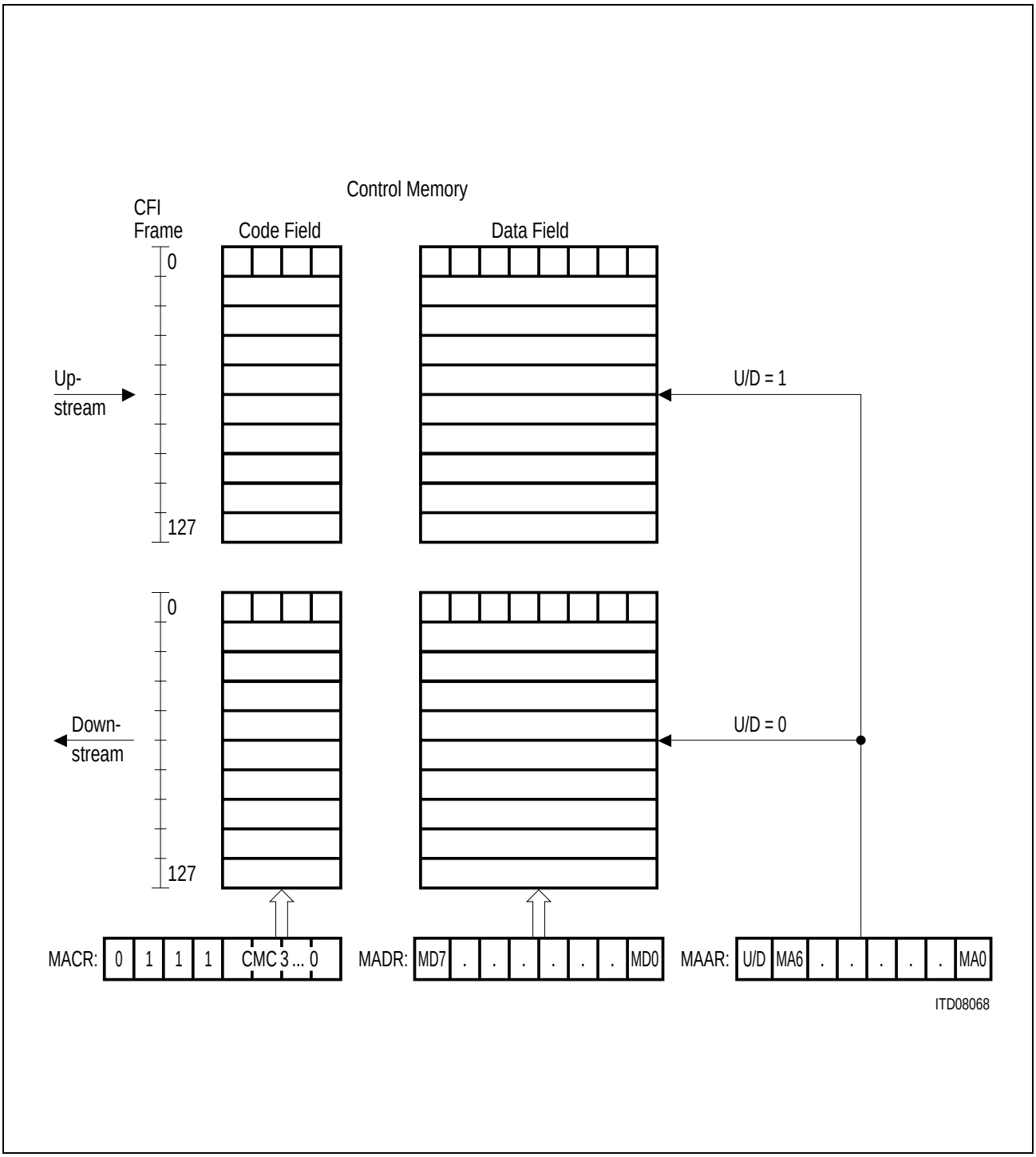



**Figure 89**
**Write Access to the Control Memory Data and Code Fields**

For reading back the CM code field, the command MACR:MOC = 111X is also used, the value of CMC3 … 0 being don't care. The code field value can then be read from the lower 4 bits of MADR.

**The Procedure for Reading the CM Code is**

W:MAAR = CFI timeslot address encoded according to **figure 84**
W:MACR = 1111 $XXXX_B$
wait for STAR:MAC = 0

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| R:MADR = | X | X | X | X | CMC3 | CMC2 | CMC1 | CMC0 |

CMC3 … 0: CM code, refer to **table 40**

**Figure 90** illustrates this behavior.

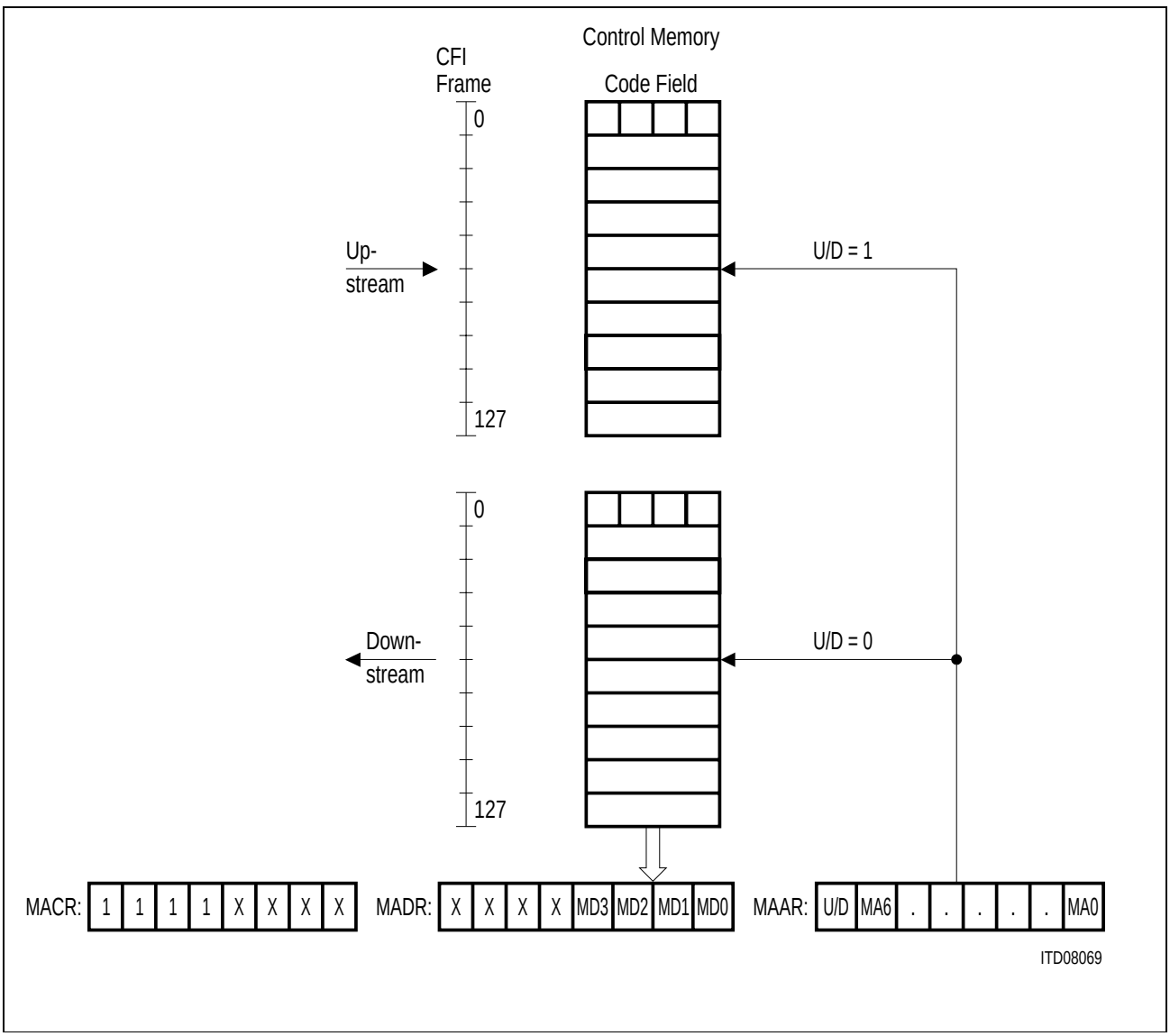



**Figure 90**
**Read Access to the Control Memory Code Field**

**Table 40** shows all available Control Memory codes.

**Table 40**
**Control Memory Codes**

| Application | CMC3 … 0 | Transferred Bits | Channel Bandwidth |
|---|---|---|---|
| Disable connection | 0000 | – | unassigned |
| Switched 8 bit channel | 0001 | bits 7 … 0 | 64 kBit/s |
| Switched 4 bit channel | 0011 | bits 7 … 4 | 32 kBit/s |
| Switched 4 bit channel | 0010 | bits 3 … 0 | 32 kBit/s |
| Switched 2 bit channel | 0111 | bits 7 … 6 | 16 kBit/s |
| Switched 2 bit channel | 0110 | bits 5 … 4 | 16 kBit/s |
| Switched 2 bit channel | 0101 | bits 3 … 2 | 16 kBit/s |
| Switched 2 bit channel | 0100 | bits 1 … 0 | 16 kBit/s |
| Preprocessed channel<br>Preprocessed channel<br>Preprocessed channel | 1000<br>1010<br>1011 | refer to **chapter 5.5** | |
| µP channel | 1001 | refer to **chapter 5.6** and **chapter 5.7** | |

**Examples**

In CFI mode 2, CFI timeslot 123 shall be initialized as a switched channel. The CM data field value therefore represents a pointer to the PCM interface.

In a first step, a timeslot assignment to PCM port 1, timeslot 34 (PCM mode 1) shall be made for a 64 kBit/s upstream connection:

W:MADR = $1100\ 0110_B$ ; upstream PCM timeslot 34, port 1
W:MAAR = $1111\ 1011_B$ ; address of upstream CFI timeslot 123
W:MACR = $0111\ 0001_B$ ; write data + code field command, code ‘0001’

In a next step, the bandwidth of the previously made connection shall be verified:

W:MAAR = $1111\ 1011_B$ ; address of upstream CFI timeslot 123
W:MACR = $1111\ 0000_B$ ; read back code field command
wait for STAR:MAC = 0
R:MADR = $XXXX\ 0001_B$ ; the code ‘0001’ (64 kBit/s channel) is read back

### Tristate Behavior at the Configurable Interface

The downstream control memory code field, together with the CSCR and OMDR registers also defines the state of the output driver at the downstream CFI ports. Unassigned channels (code ‘0000’) are set to the inactive state. Subchannels (codes ‘0010’ to ‘0111’) are only active during the sub-timeslot position specified in CSCR. The OMDR:COS bit selects between tristate outputs and open drain outputs:

**Table 41**
**Tristate/Open Drain Output Characteristics at the CFI**

| Logical State | Tristate Outputs | Open Drain Outputs |
|---|---|---|
| Logical 0 | Low voltage level | Low voltage level |
| Logical 1 | High voltage level | Not driven[1)] |
| Inactive | High impedance | Not driven[1)] |

1) An external pull-up resistor is required to establish a high voltage level.

**Figure 91** illustrates this behavior in case of tristate outputs:

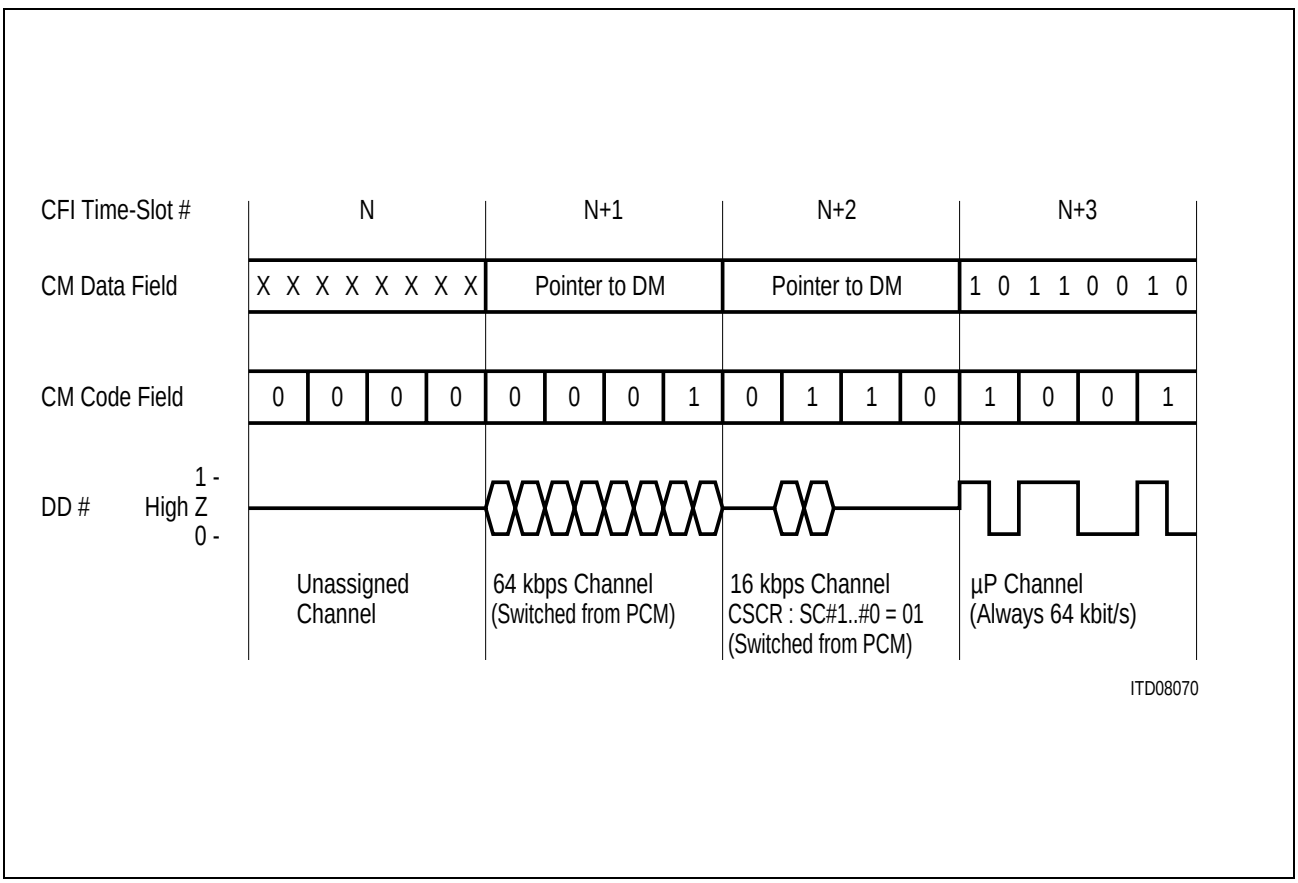



**Figure 91**
**Tristate Behavior at the CFI**

## Summary of Memory Operations

**Table 42**
**Summary of Control and Data Memory Commands**

| Application | MADR | MAAR | MACR (Hex) |
|---|---|---|---|
| Writing a PCM idle value to the upstream DM data field The MACR value specifies the bandwidth and bit position at the PCM interface | 8 bit, 4 bit or 2 bit idle value to be transmitted at the PCM interface | Address of the (upstream) PCM port and timeslot | $08_H$ (bits 7 … 0)<br>$18_H$ (bits 7 … 4)<br>$10_H$ (bits 3 … 0)<br>$38_H$ (bits 7 … 6)<br>$30_H$ (bits 5 … 4)<br>$28_H$ (bits 3 … 2)<br>$20_H$ (bits 1 … 0) |
| Reading the up- or downstream DM data field | 8 bit value transmitted at the upstream or 8 bit value received at the downstream PCM interface | Address of the PCM port and timeslot | $88_H$ |
| Writing to a single tristate field location | Tristate information contained in the 4 LSBs:<br>0 = tristated,<br>1 = active | Address of the (upstream) PCM port and timeslot | $60_H$ |
| Writing to all tristate field locations | Tristate information contained in the 4 LSBs:<br>0 = tristated,<br>1 = active | Don’t care | $68_H$ |
| Reading a single tristate field location | Tristate information contained in the 4 LSBs | Address of the (upstream) PCM port and timeslot | $E0_H$ |
| Writing to the CM data field | 8 bit value (C/I value, pointer to PCM interface, etc.) | Address of the CFI port and timeslot | $48_H$ |

**Table 42**
**Summary of Control and Data Memory Commands** (cont'd)

| Application | MADR | MAAR | MACR (Hex) |
|---|---|---|---|
| Reading the CM data field | 8 bit value (C/I value, pointer to PCM interface, etc.) | Address of the CFI port and timeslot | $C8_H$ |
| Reading the CM code field | 4 bit code contained in the 4 LSBs | Address of the CFI port and timeslot | $F0_H$ |
| Writing a switching code to the CM<br><br>The MACR value specifies the bandwidth and bit position at the PCM interface | Pointer to DM: PCM port and timeslot | Address of the CFI port and timeslot | $70_H$ (unassigned)<br>$71_H$ (bits 7 … 0)<br>$73_H$ (bits 7 … 4)<br>$72_H$ (bits 3 … 0)<br>$77_H$ (bits 7 … 6)<br>$76_H$ (bits 5 … 4)<br>$75_H$ (bits 3 … 2)<br>$74_H$ (bits 1 … 0) |
| Writing the "µP channel" code to the CM | 8 bit idle value | Address of the CFI port and timeslot | $79_H$ |
| Writing a "preprocessed channel" code to the CM | refer to **figure 104** | refer to **figure 104** | refer to **figure 104** |

## 5.4 Switched Channels

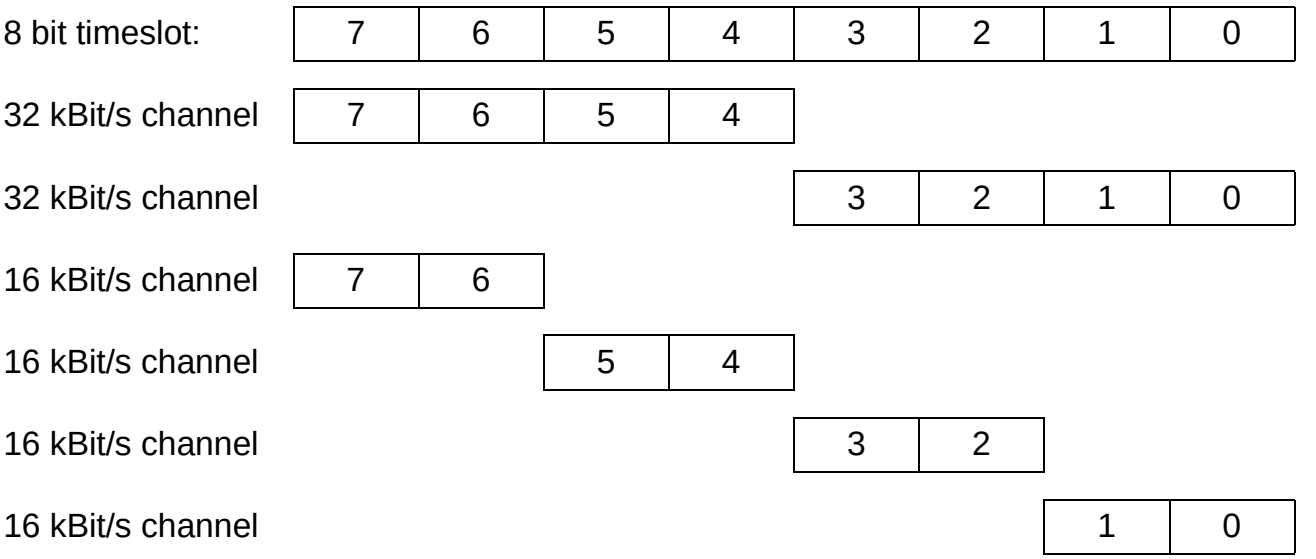
This chapter treats the switching functions between the CFI and PCM interfaces which are programmed exclusively in the control memory. The switching functions of channels which involve the µP interface or which are programmed in the synchronous transfer registers are treated in **chapter 5.6** and **chapter 5.7**.

The ELIC is a non-blocking space and time switch for 128 channels per direction. Switching is performed between the configurable (CFI) and the PCM interfaces. Both interfaces provide up to 128 timeslots which can be split up into either 4 ports with up to 32 timeslots, 2 ports with up to 64 timeslots or 1 port with up to 128 timeslots. In all of these cases each port consists of a separate transmit and receive line (duplex ports). On the CFI side a bidirectional mode is also provided (CFI mode 3) which offers 8 ports with up to 16 timeslots per port. In this case each timeslot of each port can individually be programmed to be either input or output.

The timeslot numbering always ranges from 0 to N – 1 (N = number of timeslots/frame), and each timeslot always consists of 8 contiguous bits. The bandwidth of a timeslot is therefore always 64 kBit/s.
The ELIC can switch single timeslots (64 kBit/s channels), double timeslots (128 kBit/s channels) and also 2 bit and 4 bit wide sub-timeslots (16 and 32 kBit/s channels). The bits in a timeslot are numbered 7 through 0. On the serial interfaces (PCM and CFI), bit 7 is the first bit to be transmitted or received, bit 0 the last. If the µP has access to the serial data, bit 7 represents the MSB (D7) and bit 0 the LSB (D0) on the µP bus.

The switching of 128 kBit/s channels implies that two consecutive timeslots starting with an even timeslot number are used, e.g. PCM timeslots 22 and 23 can be switched as a single 16 bit wide timeslot to CFI timeslots 4 and 5. Under these conditions it is guaranteed that the involved timeslots are submitted to the same frame delay (also refer to **chapter 5.4.4**).

The switching of channels with a data rate of 16 and 32 kBit/s is possible for the following sub-timeslot positions within an 8 bit timeslot:




### 5.4.1 CFI - PCM Timeslot Assignment

All timeslot assignments are programmed in the control memory (CM). Each line (address) of the CM refers to one CFI timeslot. The MAAR register, which is used to address the CM, therefore specifies the CFI port and timeslot to be switched. The data field of the CM contains a pointer which points to a location in the data memory (DM). The data memory contains the actual PCM data to be switched. The MADR register contains the data to be copied to the CM data field. Since this data is interpreted as a pointer to the DM, the MADR contents therefore specifies the PCM port and timeslot to be switched. The 4 bit CM code field must finally contain a value to declare the corresponding CFI timeslot as a switched channel (codes with a leading 0). This code must be written at least once to the CM using the MACR register.

Since the CFI - PCM timeslot assignment is programmed at the CFI side, it is possible to switch a single downstream PCM timeslot to several downstream CFI timeslots. It is, however, not possible to switch a single upstream CFI timeslot to several upstream PCM timeslots.

If several upstream 64 kBit/s CFI timeslots are assigned to the same upstream 64 kBit/s PCM timeslot, only the data of one CFI timeslot will be actually be switched since each upstream connection will simply overwrite the DM data field. This switching mode can therefore only effectively be used if the upstream switching is performed on different sub-timeslot locations within the same PCM timeslot (refer to **chapter 5.4.2**).

The following sequences can be used to program, verify, and cancel a CFI - PCM timeslot connection:

### Programming of a 64 kBit/s CFI - PCM Timeslot Connection

– in case the CM code field has not yet been initialized with a switching code:

W:MADR = PCM port and timeslot encoded according to **figure 84**
W:MAAR = CFI port and timeslot encoded according to **figure 84**
W:MACR = $0111\ 0001_B = 71_H$

– in case the CM code field has already been initialized with a switching code:

W:MADR = PCM port and timeslot encoded according to **figure 84**
W:MAAR = CFI port and timeslot encoded according to **figure 84**
W:MACR = $0100\ 1000_B = 48_H$

### Enabling the PCM Output Driver for a 64 kBit/s Timeslot

W:MADR = $XXXX\ 1111_B = XF_H$
W:MAAR = PCM port and timeslot encoded according to **figure 84**
W:MACR = $0110\ 0000_B = 60_H$

### Reading Back a Timeslot Assignment of a Given CFI Timeslot

– reading back the PCM timeslot involved:

W:MAAR = CFI port and timeslot encoded according to **figure 84**
W:MACR = $1100\ 1000_B = C8_H$
wait for STAR:MAC = 0
R:MADR = PCM port and timeslot encoded according to **figure 84**

– reading back the involved bandwidth and PCM sub-timeslot position:

W:MAAR = CFI port and timeslot encoded according to **figure 84**
W:MACR = $1111\ 0000_B = F0_H$
wait for STAR:MAC = 0
R:MADR = XXXX code; 4 bit bandwidth code encoded according to **table 40**

### Cancelling of a Programmed CFI - PCM Timeslot Connection

W:MADR = don’t care
W:MAAR = CFI port and timeslot encoded according to **figure 84**
W:MACR = $0111\ 0000_B = 70_H$; code ‘0000’ (unassigned channel)

### Disabling the PCM Output Driver

W:MADR = $XXXX\ 0000_B = X0_H$
W:MAAR = PCM port and timeslot encoded according to **figure 84**
W:MACR = $0110\ 0000_B = 60_H$

### Examples

In PCM mode 1 and CFI mode 3 the following connections shall be programmed:

Upstream: CFI port 5, timeslot 7, bits 7 … 0 to PCM port 0, timeslot 12, bits 7 … 0

W:MADR = $1001\ 1000_B$ ; PCM timeslot encoding according to **figure 84**
W:MAAR = $1011\ 1011_B$ ; CFI timeslot encoding according to **figure 84**
W:MACR = $0111\ 0001_B$ ; CM code for switching a 64 kBit/s channel (code ‘0001’)

Downstream: CFI port 4, timeslot 2, bits 7 … 0 from PCM port 1, timeslot 3, bits 7 … 0

W:MADR = $0000\ 0111_B$ ; PCM timeslot encoding according to **figure 84**
W:MAAR = $0001\ 1000_B$ ; CFI timeslot encoding according to **figure 84**
W:MACR = $0111\ 0001_B$ ; CM code for switching a 64 kBit/s channel (0001)

The following sequence sets transmit timeslot 12 of PCM port 0 to low impedance:

W:MADR = $0000\ 1111_B$ ; all bits to low Z
W:MAAR = $1001\ 1011_B$ ; PCM timeslot encoding according to **figure 84**
W:MACR = $0110\ 0000_B$ ; MOC code ‘1100’ to access the tristate field

After these three programming steps, the ELIC memories will have the following contents:

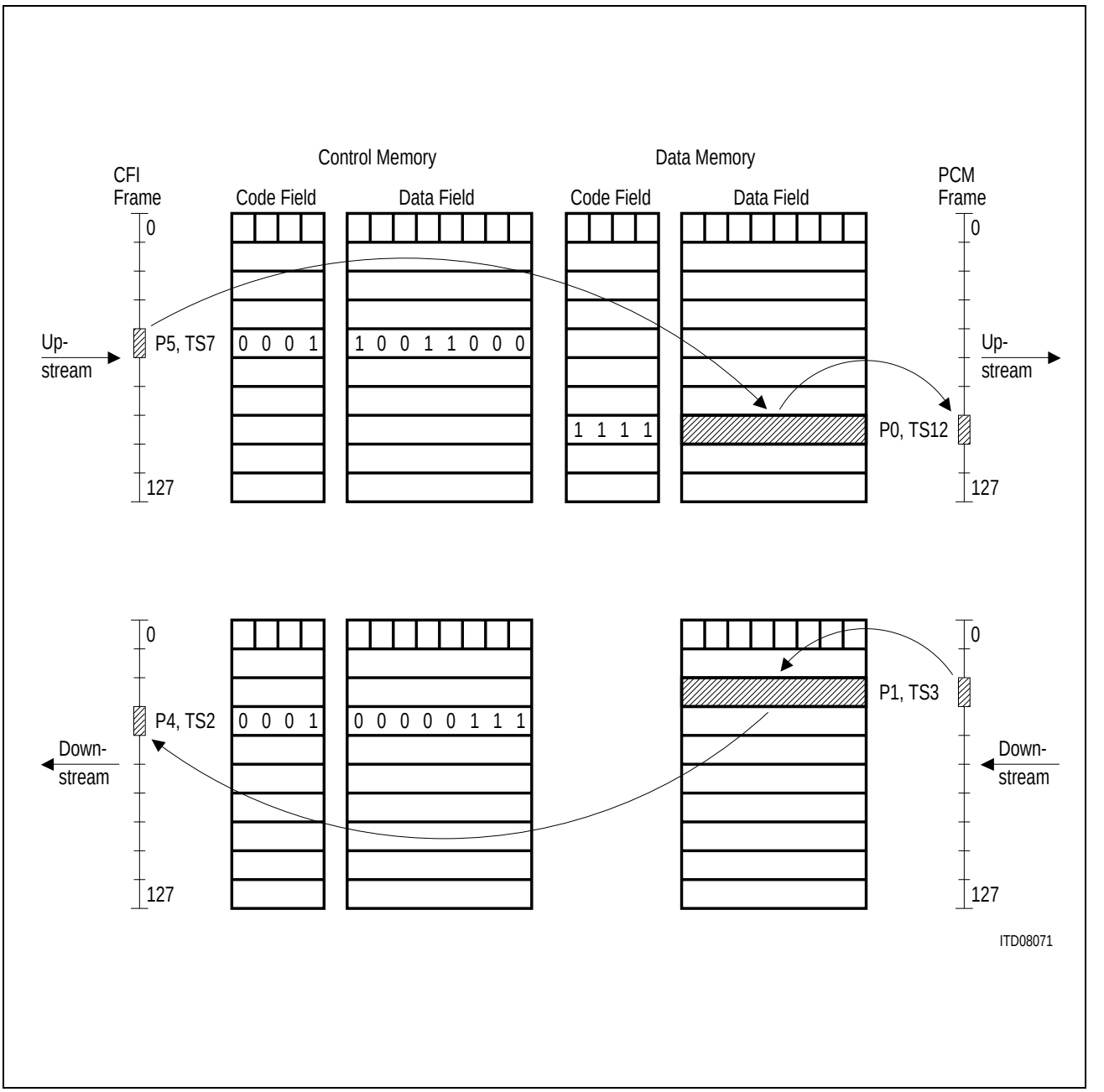



**Figure 92**
**Memory Content of the ELIC® for a CFI - PCM Timeslot Connection**

### 5.4.2 Subchannel Switching

The switching of subchannels is programmed by first specifying the timeslot (which is always 8 bits wide) to be switched, then by restricting the actual switching operation to the desired bandwidth and sub-timeslot position. The switching function for an (8 bit) CFI timeslot is programmed in the control memory (CM) by writing a pointer that points to an (8 bit) PCM timeslot to the corresponding data field location. The MADR register contains the pointer (PCM timeslot) and the MAAR register is used to specify the CFI timeslot.

The ‘8 bit’ connection can now be restricted to the desired 4 or 2 bit connection by selecting an appropriate control memory code. The code is programmed via MACR:CMC3 … 0. These subchannel codes perform two functions: they specify the bandwidth (actual number of bits to be switched) and the location of the sub-timeslot within the selected (8 bit) PCM timeslot. The location of the sub-timeslot within the selected (8 bit) CFI timeslot is predefined by the setting of the CSCR register. Each CFI port can be set to a different sub-timeslot mode. In each mode a certain relationship exists between programmed bandwidth (which can still be individually selected for each CFI timeslot) and the occupied bit positions within the timeslot (which is fixed for each CFI port by the CSCR register).

It should be noted that only one sub-timeslot can exist within a given CFI timeslot. On the PCM side however each timeslot may be split up into $2 \times 4$ bits, $4 \times 2$ bits or any mixture of these.

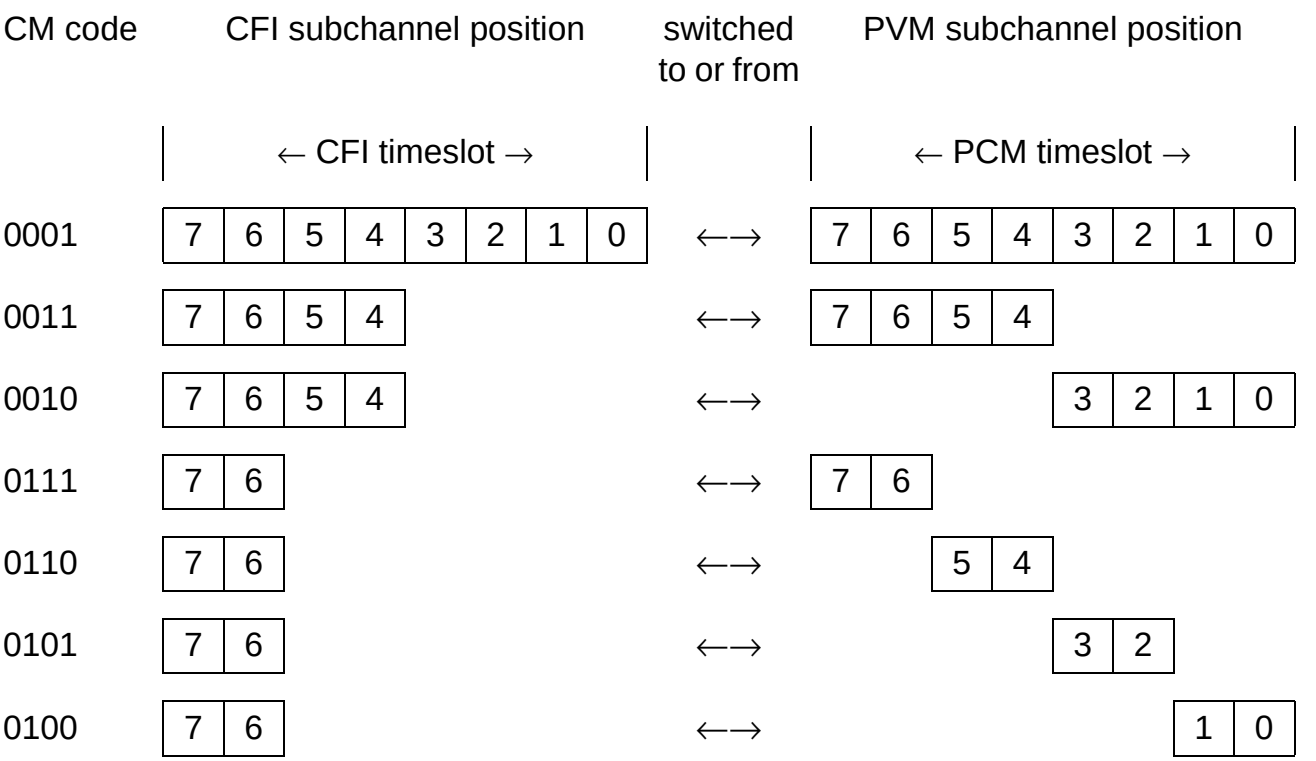
The CSCR register has the following format:

**CFI Subchannel Register** read/write reset value: $00_H$

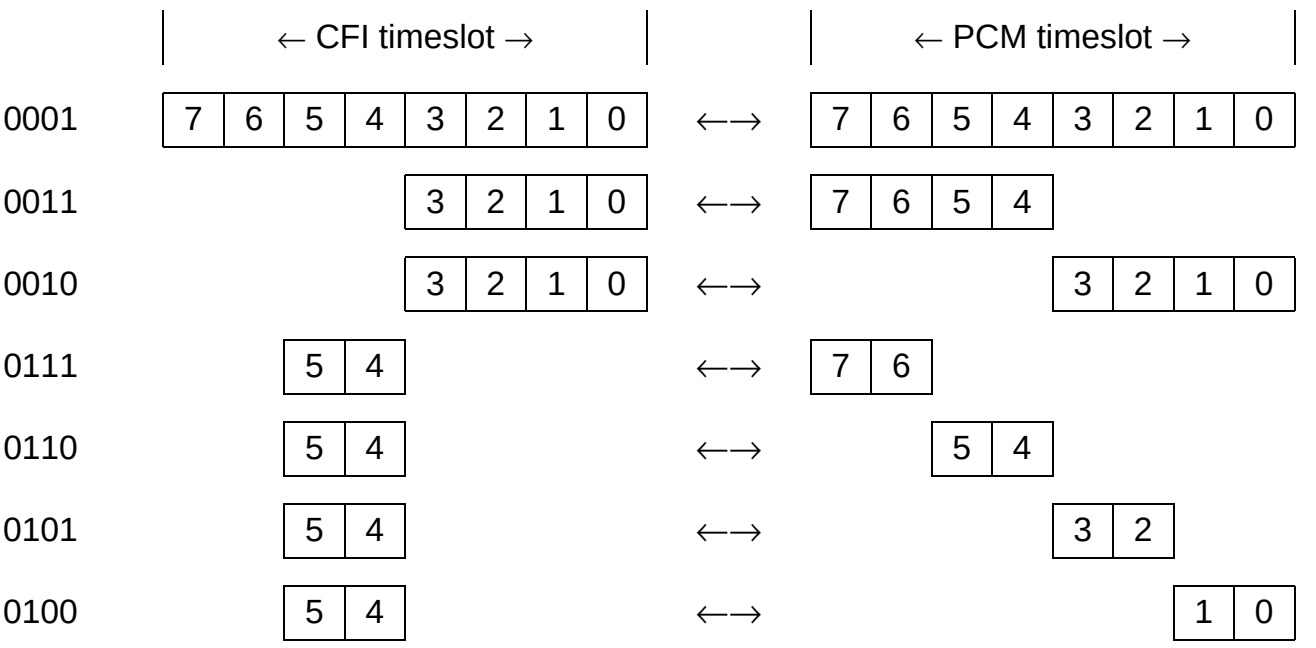
| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| CSCR | SC31 | SC30 | SC21 | SC20 | SC11 | SC10 | SC01 | SC00 |

Below, all possible combinations of subchannel switching between the CFI and PCM interfaces are shown:

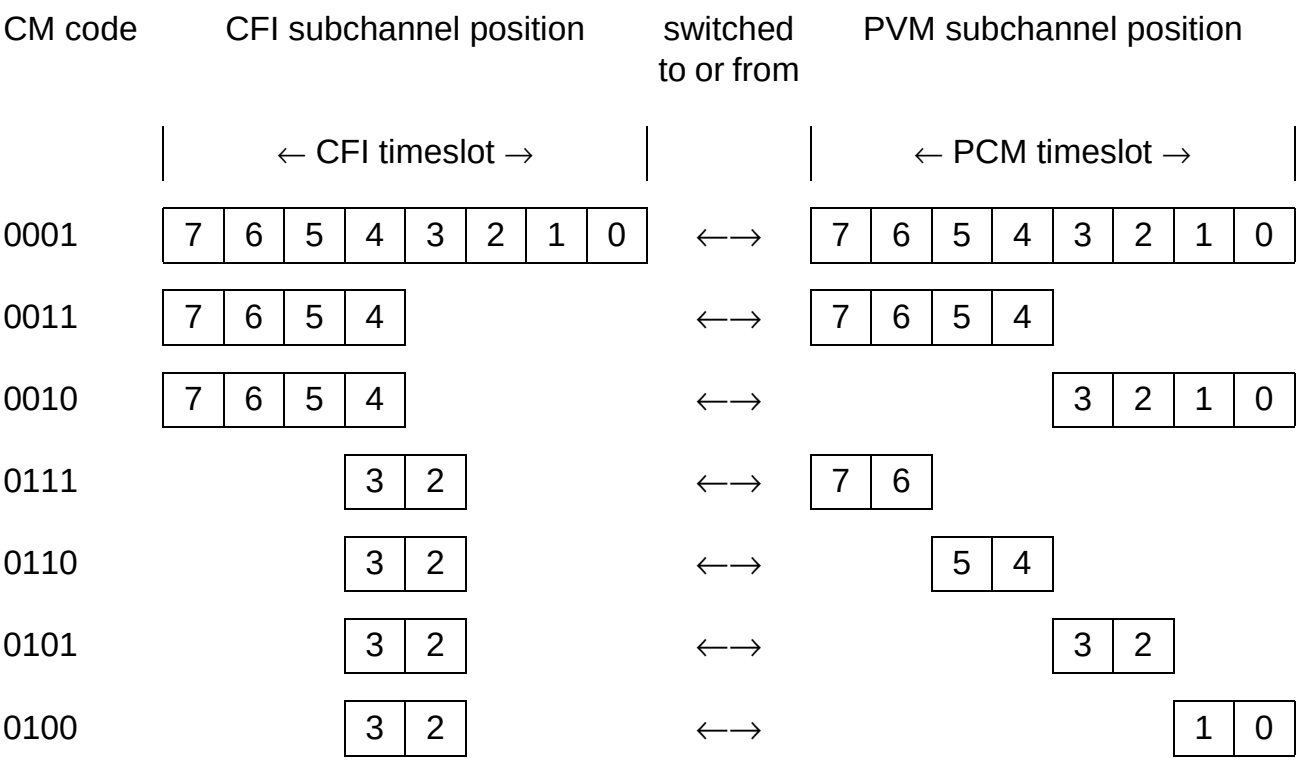
Subchannel selection SC#1 … SC#0 = 00:



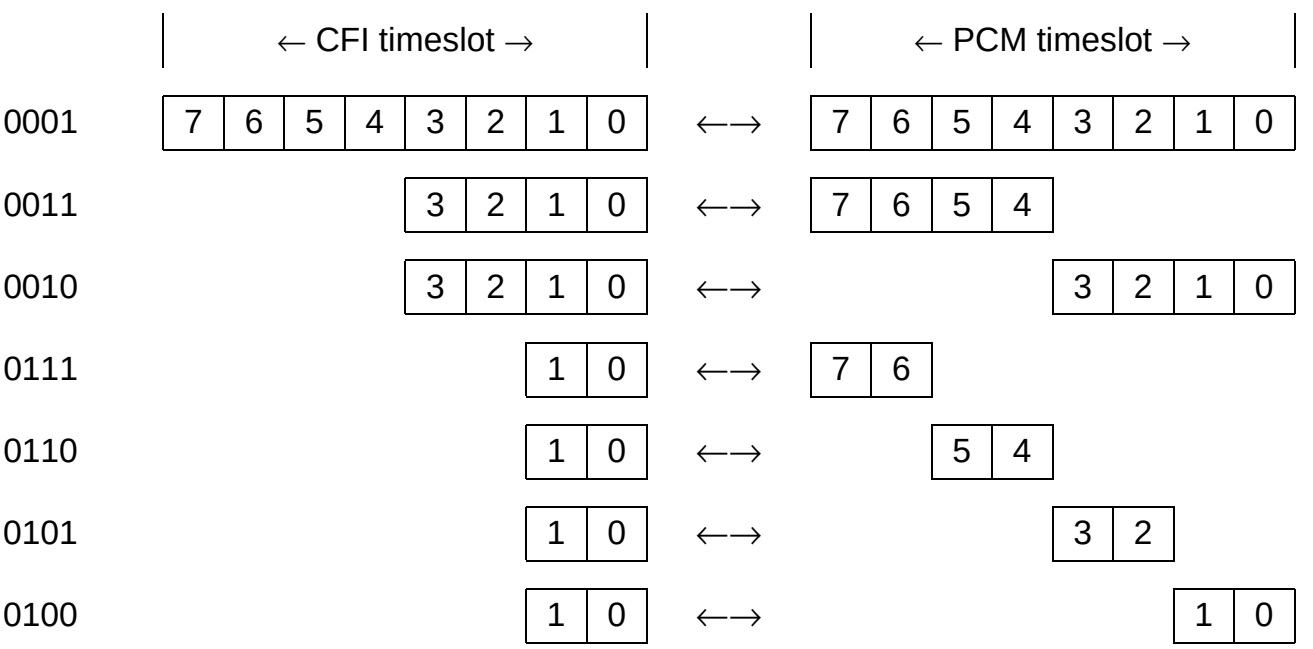

Subchannel selection SC#1 … SC#0 = 01:



Subchannel selection SC#1 … SC#0 = 10:




Subchannel selection SC#1 … SC#0 = 11:



### Examples

In PCM mode 0 and CFI mode 0 the following connections shall be programmed:

Upstream: CFI port 0, timeslot 3, bits 1 … 0 to PCM port 0, timeslot 4, bits 1 … 0

| | | | |
|---|---|---|---|
| W:MADR | = | $1001\ 0000_B$ | PCM timeslot encoding, the subchannel position is defined by MACR:CMC3 … 0 = 0100 |
| W:MAAR | = | $1000\ 1001_B$ | CFI timeslot encoding, the subchannel position is defined by CSCR:SC01 … 00 = 11 |
| W:MACR | = | $0111\ 0100_B$ | CM code for switching a 16 kBit/s/bits 1 … 0 channel (0100) |

Upstream: CFI port 3, timeslot 7, bits 3 … 2 to PCM port 0, timeslot 4, bits 5 … 4

| | | | |
|---|---|---|---|
| W:MADR | = | $1001\ 0000_B$ | PCM timeslot encoding, the subchannel position is defined by MACR:CMC3 … 0 = 0110 |
| W:MAAR | = | $1001\ 1111_B$ | CFI timeslot encoding, the subchannel position is defined by CSCR:SC31 … 30 = 10 |
| W:MACR | = | $0111\ 0110_B$ | CM code for switching a 16 kBit/s, bits 3 … 2 channel (0110) |

The following sequence sets transmit timeslot 4 of PCM port 0 bits 5 … 4 and 1 … 0 to low impedance and bits 7 … 6 and 3 … 2 to high impedance:

| | | | |
|---|---|---|---|
| W:MADR | = | $0000\ 0101_B$ | bits 5, 4, 1, 0 to low Z and bits 7, 6, 3, 2 to high Z |
| W:MAAR | = | $1001\ 0000_B$ | PCM timeslot encoding |
| W:MACR | = | $0110\ 0000_B$ | MOC code to access the tristate field |

Downstream: CFI port 2, timeslot 7, bits 3 … 0 from PCM port 1, timeslot 3, bits 7 … 4

| | | | |
|---|---|---|---|
| W:MADR | = | $0000\ 1011_B$ | PCM timeslot encoding, the subchannel position is defined by MACR:CMC3 … 0 = 0011 |
| W:MAAR | = | $0001\ 1101_B$ | CFI timeslot encoding, the subchannel position is defined by CSCR:SC21 … 20 = 01 |
| W:MACR | = | $0111\ 0011_B$ | CM code for switching a 32 kBit/s/bits 7 … 4 channel (0011) |

Downstream: CFI port 2, timeslot 10, bits 5 … 4 from PCM port 0, timeslot 4, bits 7 … 6

W:MADR = $0001\ 0000_B$ PCM timeslot encoding, the subchannel position is defined by MACR:CMC3 … 0 = 0111

W:MAAR = $0010\ 1100_B$ CFI timeslot encoding, the subchannel position is defined by CSCR:SC21 … 20 = 01

W:MACR = $0111\ 0111_B$ CM code for switching a 16 kBit/s/bits 7 … 6 channel (0111)

Finally the CSCR register has to be programmed to define the subchannel positions at the CFI:

W:CSCR = $1001\ XX11_B$ port 0: bits 1 … 0 or 3 … 0; port 1: not used in this example;
port 2: bits 5 … 4 or 3 … 0; port 3: bits 3 … 2 or 7 … 4

After these three programming steps, the ELIC memories will have the following content:

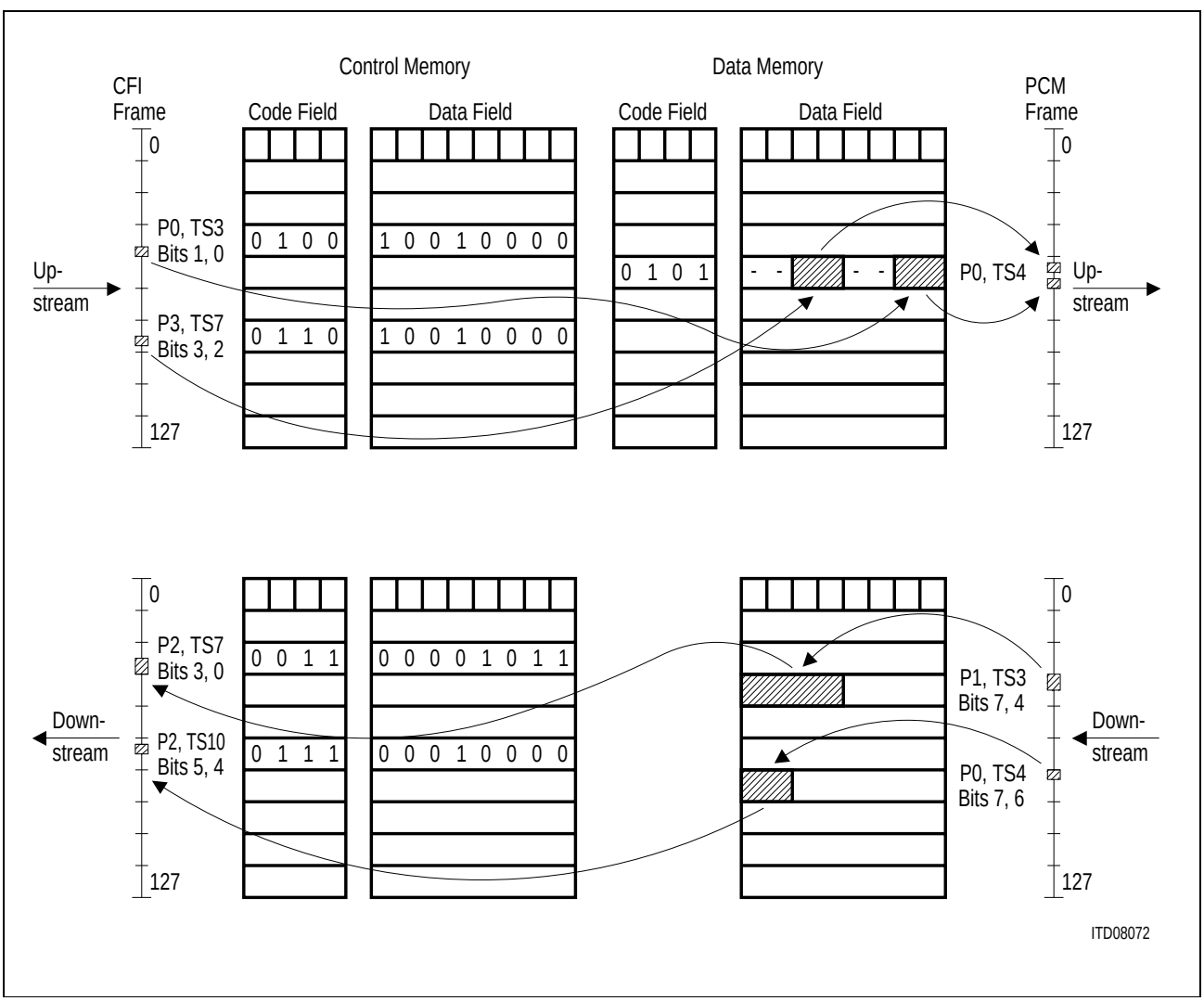



**Figure 93**
**Memory Content in Case of CFI - PCM Subchannel Connections**

### 5.4.3 Loops

Loops between timeslots (or even sub-timeslots) of the CFI (CFI $\rightarrow$ CFI) or the PCM interface (PCM $\rightarrow$ PCM) can easily be programmed in the control memory. It is thus possible to establish individual loops for individual timeslots on both interfaces without making external connections. These loops can serve for test purposes only or for real switching applications within the system. It should be noted that such a loop connection is always carried out over the opposite interface i.e. looping back a CFI timeslot to another CFI timeslot occupies a spare upstream PCM timeslot and looping back a PCM timeslot to another PCM timeslot occupies a spare downstream and upstream CFI timeslot. The required timeslot on the opposite interface can however be switched to high impedance in order not to disturb the external line.

#### 5.4.3.1 CFI - CFI Loops

For looping back a timeslot of a CFI input port to a CFI output port, two connections must be programmed:

A first connection switches the upstream CFI timeslot to a spare PCM timeslot. This connection is programmed like a normal CFI to PCM link, i.e the MADR contains the encoding for the upstream PCM timeslot (U/D = 1) which is written to the upstream CM (MAAR contains the encoding for the upstream CFI timeslot (U/D = 1)). If the data should also be transmitted at TxD#, the tristate field of that PCM timeslot can be set to low impedance (transparent loop). If TxD# should be disabled, the tristate field of that PCM timeslot can be set to high impedance (non-transparent loop).

The second connection switches the “upstream” PCM timeslot (contents of the upstream data memory) back to the downstream CFI timeslot. This connection is programmed by using exactly the same MADR value as has been used for the first connection, i.e. the encoding for the spare upstream PCM timeslot (with U/D = 1). This MADR value is written to the downstream CM (MAAR contains the encoding for the downstream CFI timeslot (U/D = 0).

The following example illustrates the necessary programming steps for establishing CFI to CFI loops.

**Example**

In PCM mode 0 and CFI mode 0 the following non-transparent CFI to CFI loop via PCM port 0, timeslot 0 shall be programmed:

Upstream: CFI port 2, timeslot 4, bits 7 … 0 to PCM port 0, timeslot 0, bits 7 … 0

| | | | |
|---|---|---|---|
| W:MADR | = | $1000\ 0000_B$ | PCM timeslot encoding (pointer to upstream DM) |
| W:MAAR | = | $1001\ 0100_B$ | CFI timeslot encoding (address of upstream CM) |
| W:MACR | = | $0111\ 0001_B$ | CM code for switching a 64 kBit/s/bits 7 … 0 channel (0001) |

Downstream: CFI port 1, timeslot 7, bits 7 … 0 from PCM port 0, timeslot 0, bits 7 … 0

| | | | |
|---|---|---|---|
| W:MADR | = | $1000\ 0000_B$ | PCM timeslot encoding (pointer to **upstream** DM) |
| W:MAAR | = | $0001\ 1011_B$ | CFI timeslot encoding (address of downstream CM) |
| W:MACR | = | $0111\ 0001_B$ | CM code for switching a 64 kBit/s/bits 7 … 0 channel (0001) |

The following sequence sets transmit timeslot 0 of PCM port 0 to high impedance:

| | | | |
|---|---|---|---|
| W:MADR | = | $0000\ 0000_B$ | all bits to high Z |
| W:MAAR | = | $1000\ 0000_B$ | PCM timeslot encoding |
| W:MACR | = | $0110\ 0000_B$ | MOC code to access the tristate field |

After these three programming steps, the ELIC memories will have the following contents:

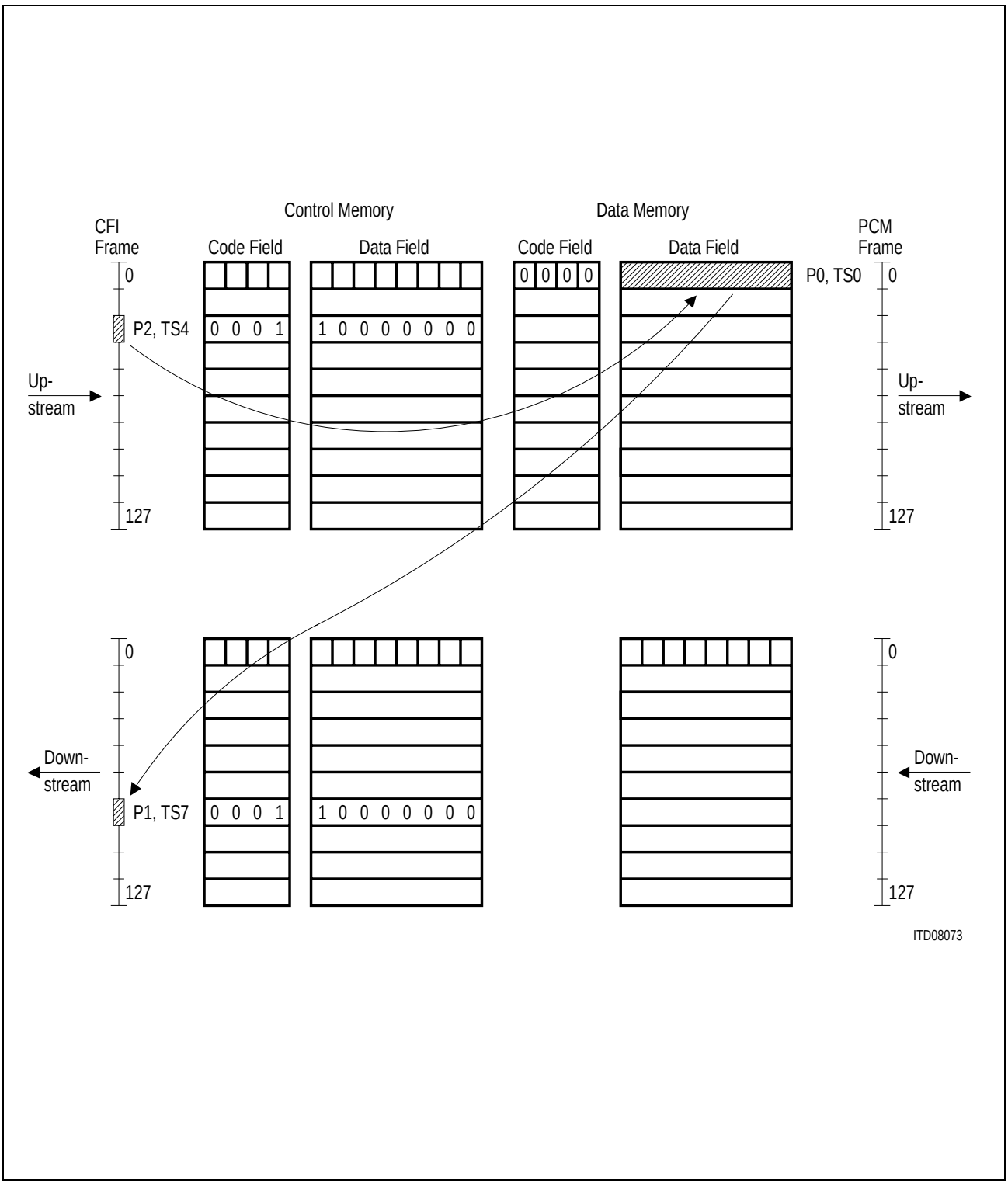



**Figure 94**
**Memory Content in Case of a CFI → CFI Loop**

### 5.4.3.2 PCM - PCM Loops

For looping back a timeslot of a PCM input port to a PCM output port, two connections must be programmed:

The first connection switches the downstream PCM timeslot to a spare CFI timeslot. This connection is programmed like a normal PCM to CFI link, i.e the MADR contains the encoding for the downstream PCM timeslot (U/D = 0) which is written to the downstream CM (MAAR contains the encoding for the downstream CFI timeslot (U/D = 0)). If the data should also be transmitted at DD# (transparent loop), the programming is performed with MACR:CMC3 … 0 = 0001 … 0111, the actual code depending on the required bandwidth. If DD# should be disabled (non-transparent loop), the programming is performed with MACR:CMC3 … 0 = 0000, the code for unassigned channels.

The second connection switches the serial CFI timeslot data back to the upstream PCM timeslot. This connection is programmed by writing the encoded PCM timeslot via MADR to the upstream CM. This “upstream” pointer must however have the MSB set to 0 (U/D = 0). This MADR value is written to the same spare CFI timeslot as the PCM timeslot had been switched to in the first step. Only that now the upstream CM is accessed (MAAR addresses the upstream CFI timeslot (U/D = 1)).

In contrast to the CFI $\rightarrow$ PCM $\rightarrow$ CFI loop, which is internally realized by extracting the CFI data out of the upstream data memory (see **chapter 5.4.3.1**), the PCM $\rightarrow$ CFI $\rightarrow$ PCM loop is realized differently:

The downstream PCM $\rightarrow$ CFI connection switches the PCM data to the internal downstream serial CFI output. From this internal output, the data is switched to the upstream serial CFI input if the control memory of the corresponding upstream CFI timeslot contains a pointer with a leading 0 (U/D = 0). However, this pointer (with U/D = 0) still points to the upstream data memory, i.e to an upstream PCM timeslot.

The following example illustrates the necessary programming steps for establishing PCM to PCM loops:

**Example**

In PCM mode 1 and CFI mode 0 the following non-transparent PCM to PCM loop via CFI port 1, timeslot 4 shall be programmed:

Downstream: CFI port 1, timeslot 4, bits 7 … 0 from PCM port 0, timeslot 13, bits 7 … 0

W:MADR = $0001\ 1001_B$ PCM timeslot encoding (pointer to downstream DM)
W:MAAR = $0001\ 0010_B$ CFI timeslot encoding (address of downstream CM)
W:MACR = $0111\ 0000_B$ CM code for unassigned channel (0000)

Upstream: CFI port 1, timeslot 4, bits 7 … 0 to PCM port 0, timeslot 5, bits 7 … 0

W:MADR = $0000\ 1001_B$ PCM timeslot encoding (pointer to 'upstream' DM, loop switch (MSB = 0) activated)
W:MAAR = $1001\ 0010_B$ CFI timeslot encoding (address of upstream CM)
W:MACR = $0111\ 0001_B$ CM code for switching a 64 kBit/s/bits 7 … 0 channel (0001)

The following sequence sets transmit timeslot 5 of PCM port 0 to low impedance:

W:MADR = $0000\ 1111_B$ all bits to low Z
W:MAAR = $1000\ 1001_B$ PCM timeslot encoding
W:MACR = $0110\ 0000_B$ MOC code to access the tristate field

After these three programming steps, the ELIC memories will have the following contents:

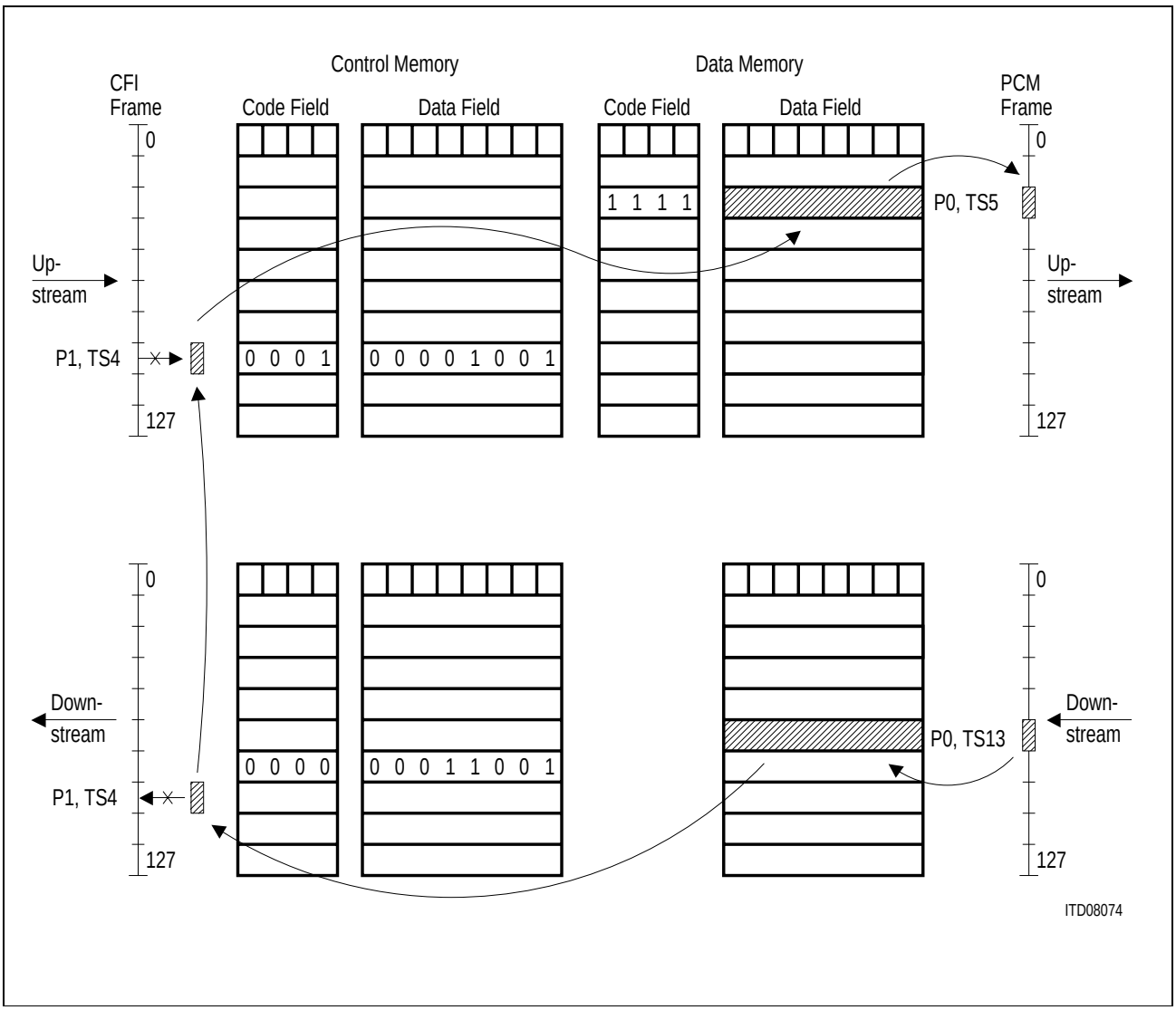



**Figure 95**
**Memory Content in Case of a PCM → PCM Loop**

### 5.4.4 Switching Delays

When a channel is switched from an input time slot (e.g. from the PCM interface) to an output time slot (e.g. to the CFI), it is sometimes useful to know the frame delay introduced by this connection. This is of prime importance for example if channels having a bandwidth of n × 64 kBit/s (e.g. H0 channels: 6 × 64 = 384 kBit/s) shall be switched by the ELIC. If all 6 time slots of an H0 channel are not submitted to the same frame delay, time slot integrity is no longer maintained.

Since the ELIC has only a one frame buffer, the switching delay depends mainly on the location of the output time slot with respect to the input time slot. If there is 'enough' time between the two locations, the ELIC switches the input data to the output data within the same frame (see **figure 96 a)**). If the time between the two locations is too small or if the output time slot is later in time than the input time slot, the data received in frame N will only be transmitted in frame N + 1 or even N + 2 (see **figure 96 b)**) and **figure 96 c)**).

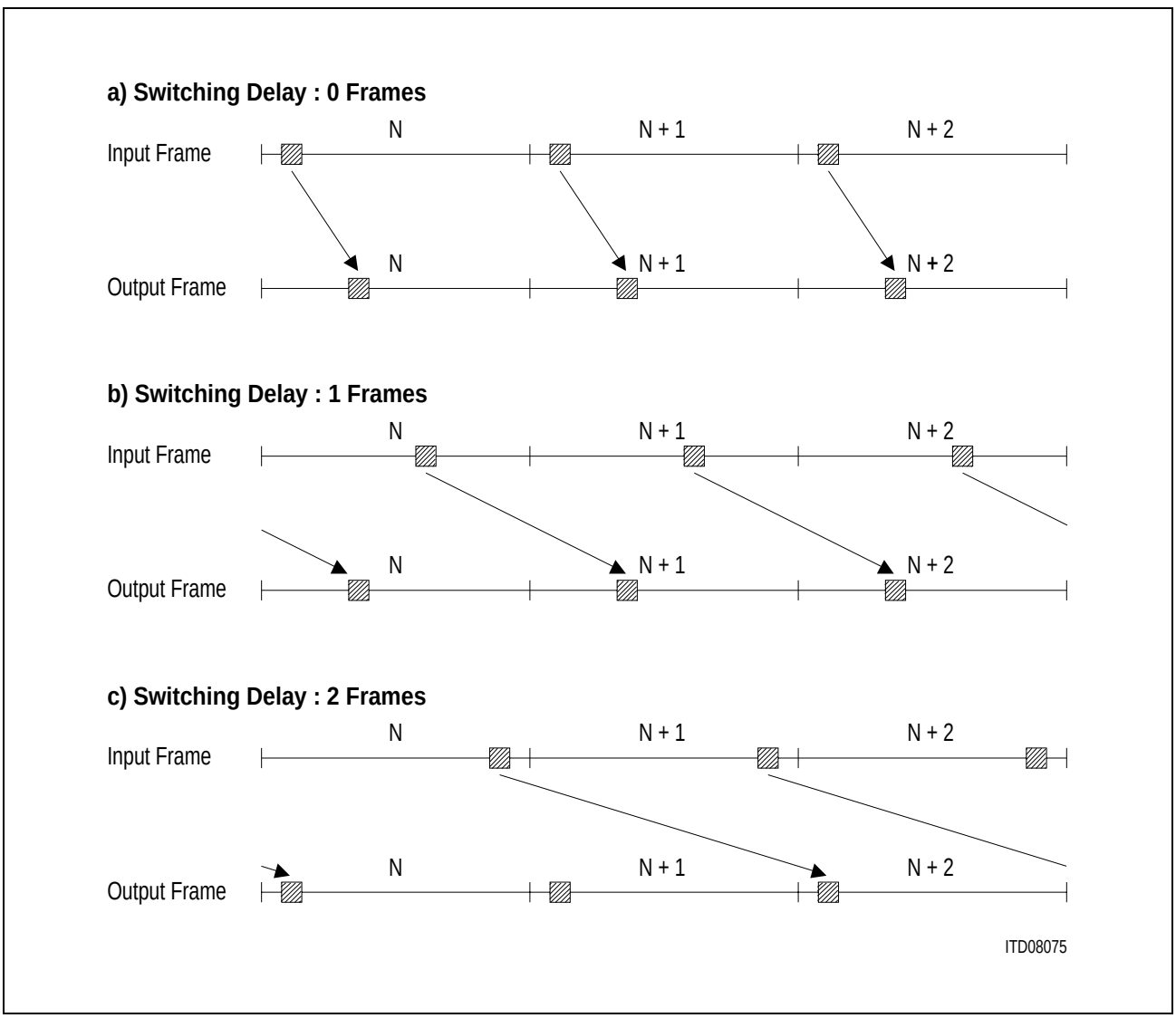



**Figure 96**

The exact respective time slot positions where the delay skips from 0 frames to 1 frame and from 1 frame to 2 frames can be determined when having a closer look at the internal read and write cycles to the Data Memory.

The next two figures show the internal timing characteristics for the access to the data memory (DM) of the ELIC. For simplicity, only the case where the PCM and CFI frames both start simultaneously at position ‘time slot 0, bit 7’ is shown. Also, only the cases with 2, 4 and 8 $\times$ 1024 kBit/s data rates are shown. All other cases (different frame offsets and different data rates) can, however, be deduced by taking into account the respective frame positions, and, eventually, by taking into account a different RCL frequency.

#### 5.4.4.1 Internal Procedures at the Serial Interfaces

The data is received and transmitted at the PCM and configurable interfaces in a serial format. Before being written to the DM, the data is converted into parallel format. The vertical arrows indicate the position in time where the incoming time slot data is written to the data memory. The writing to the DM is only possible during certain time intervals which are also indicated in the figures. For outgoing time slots, the data is first read in parallel format from the DM. This also is only possible during certain read cycles as indicated in the figures (vertical arrows). Before the time slot data is sent out, it must first be converted into serial format.

The data contained in a time slot can be switched from an incoming time slot position to an outgoing time slot position within the same frame (0 frame delay) if the reading from the DM occurs after the writing to the DM. If the reading occurs before the writing, the data from the previous frame is taken, i.e. the frame delay is one frame.

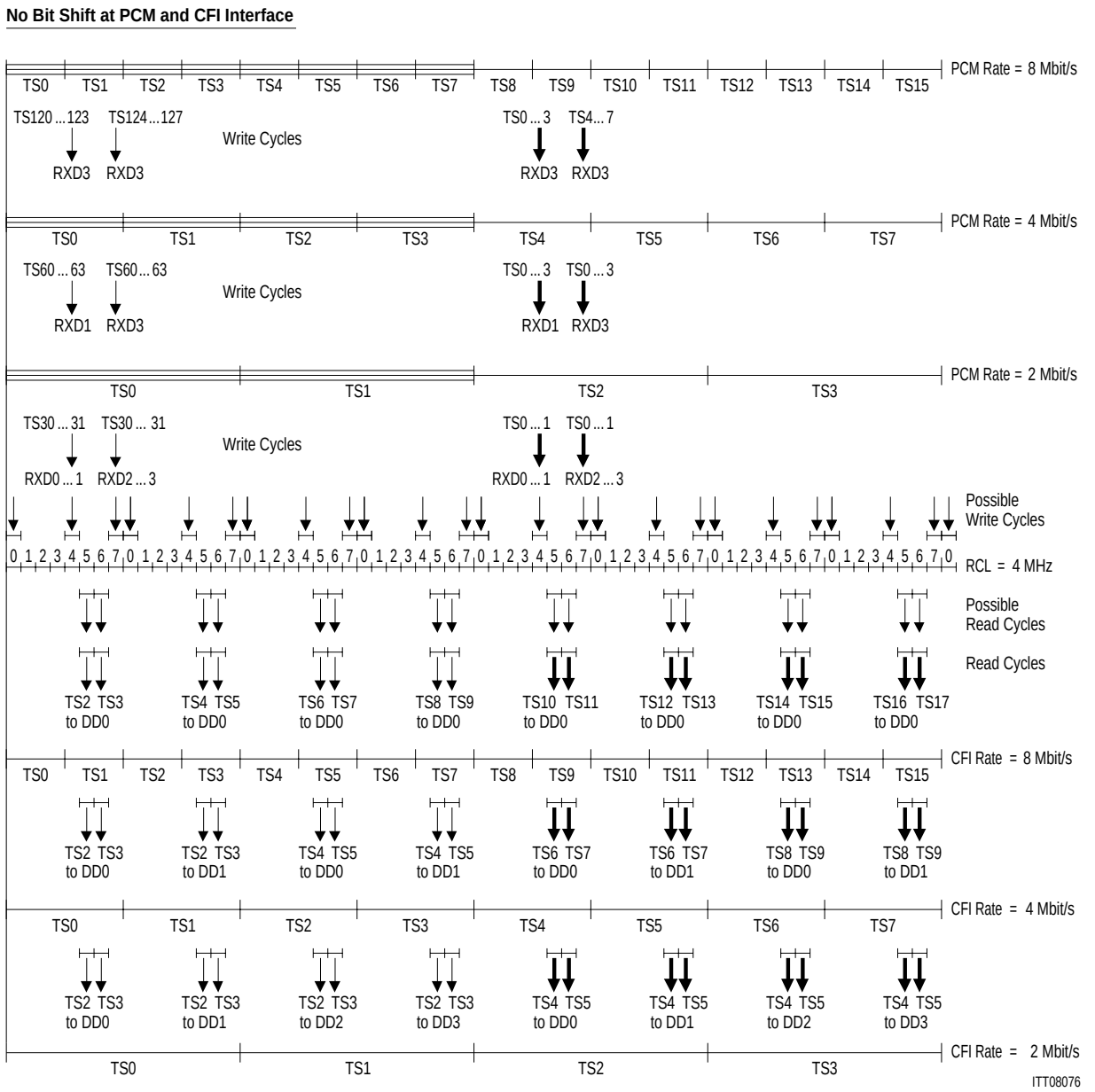



**Figure 97**
**Internal Timing Data Downstream**

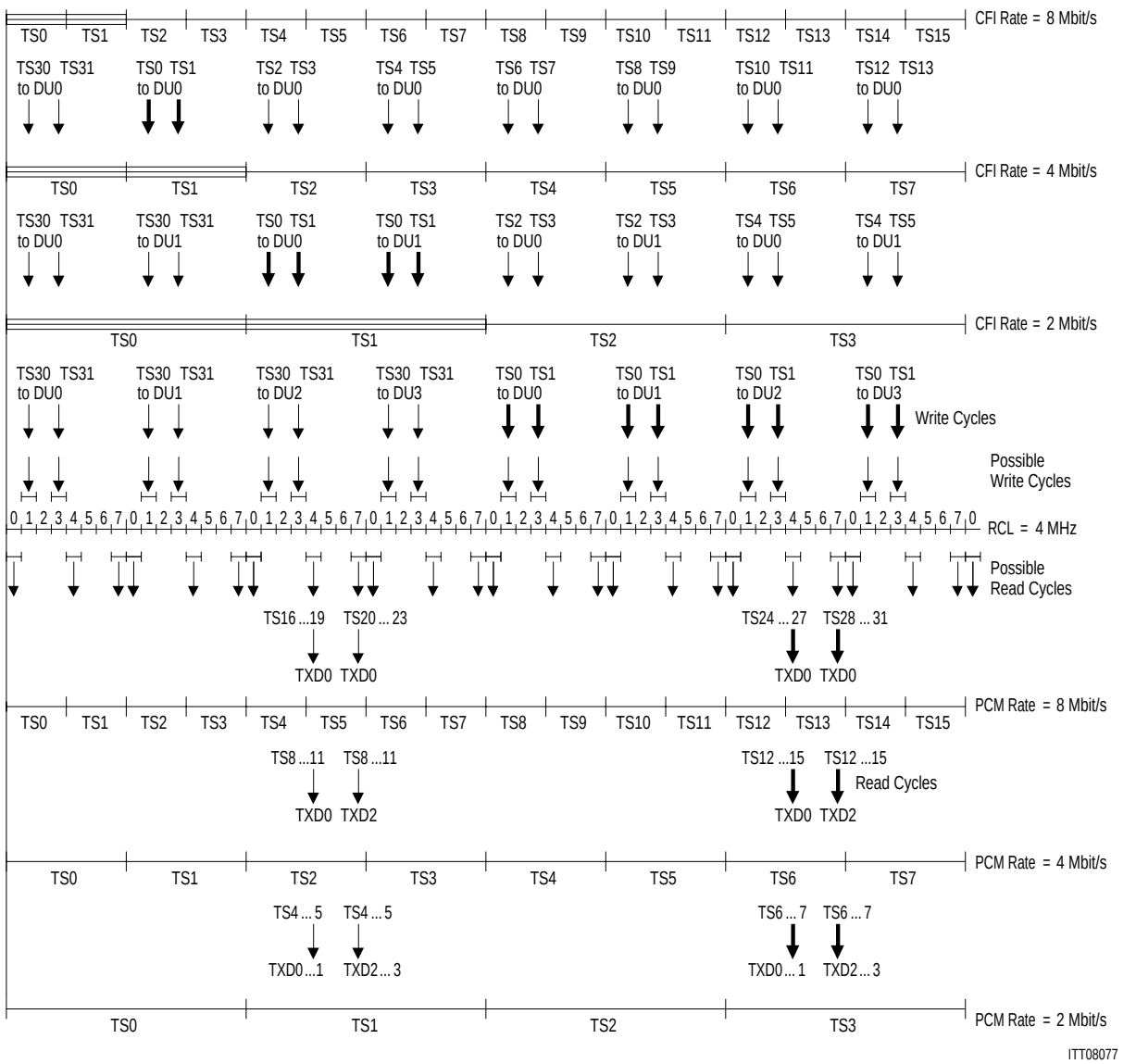



**Figure 98**
**Internal Timing Data Upstream**

#### 5.4.4.2 How to Determine the Delay

In order to determine the switching delay for a certain configuration, the following rules have to be applied with respect to the timing diagram:

**Data Downstream**

– At the PCM interface the incoming data (data downstream) is written to the RAM after the beginning of:

time slot: $2 \times n$ for mode 0
time slot: $4 \times n$ for mode 1
time slot: $8 \times n$ for mode 2

*Note: n is an integer number.*

The point of time to write the data to the RAM is RCL period 0, 4, 7 for the PCM interface Due to internal delays, the RCL period at the beginning of time slot $2 \times n$ (for mode 0), $4 \times n$ (for mode 1), $8 \times n$ for mode 2) is not a valid write cycle.

– At the CFI interface the data, that is to be transmitted on:

TS $2 \times n + 4$ ... $2 \times n + 5$ (CFI mode 0)
TS $2 \times n + 6$ ... $2 \times n + 7$ (CFI mode 1)
TS $2 \times n + 10$ ... $2 \times n + 11$ (CFI mode 2)

is read out of the RAM as soon as time slot:

$2 \times n + 1$ (for mode 0)
$2 \times n + 3$ (for mode 1)
$2 \times n + 7$ (for mode 2) **is transmitted**

*Note: n is an integer number; the time slot number can't exceed the max. number of TS.*

The point of time to read the data from the RAM is RCL period 5 and 6 for the CFI interface.

The data is read out of the RAM in several steps in the following order:

CFI mode 0: - even TS for DD0, odd TS for DD0,
even TS for DD0, odd TS for DD1,
even TS for DD0, odd TS for DD2,
even TS for DD0, odd TS for DD3

CFI mode 1: - even TS for DD0, odd TS for DD0,
even TS for DD0, odd TS for DD1

CFI mode 2: - even TS for DD0, odd TS for DD0

**Data Upstream**

– At the CFI interface the incoming data (data upstream) is written to the RAM starting with DU0 at the beginning of:

time slot: $2 \times n$ for CFI mode 0
time slot: $2 \times n$ for CFI mode 1
time slot: $2 \times n$ for CFI mode 2

*Note: n is an integer number; the time slot number can't exceed the max. number of TS.*

The point of time to write the data to the RAM is RCL period 1 and 3 for the CFI interface

– At the PCM interface the data, that is to be transmitted on

TS $2 \times n + 4$ ... TS $2 \times n + 5$ (for PCM mode 0)
TS $4 \times n + 8$ ... TS $4 \times n + 11$ (for PCM mode 1)
TS $8 \times n + 16$ ... TS $8 \times n + 23$ (for PCM mode 2)

is read out of the RAM as soon as time slot:

$2 \times n$ (for PCM mode 0)
$4 \times n + 1$ (for PCM mode 1)
$8 \times n + 3$ (for PCM mode 2) is transmitted

*Note: n is an integer number; the time slot number can't exceed the max. number of TS.*

The point of time to read the data from the RAM, is RCL period 0, 4, 7 for the PCM interface

Due to internal delays, the RCL period at the beginning of time slot $2 \times n + 1$ (for PCM 0), $4 \times n + 2$ (for PCM mode1), $8 \times n + 4$ for PCM mode 2) is no valid write cycle.

The data is read out of the RAM in two steps:

PCM mode 0: in a block of 2 TS for TXD0 … 1 then for TXD2 … 3
PCM mode 1: in a block of 4 TS for TXD0 then for TXD2
PCM mode 2: in halfs of a 8 TS blocks for TXD0 (first half) then for TXD0 (second half)

**Considering a Bit Shift**

A bit shift will also influence switching delays.

If the PCM frame is shifted relative to the frame signal, proceed as indicated below:

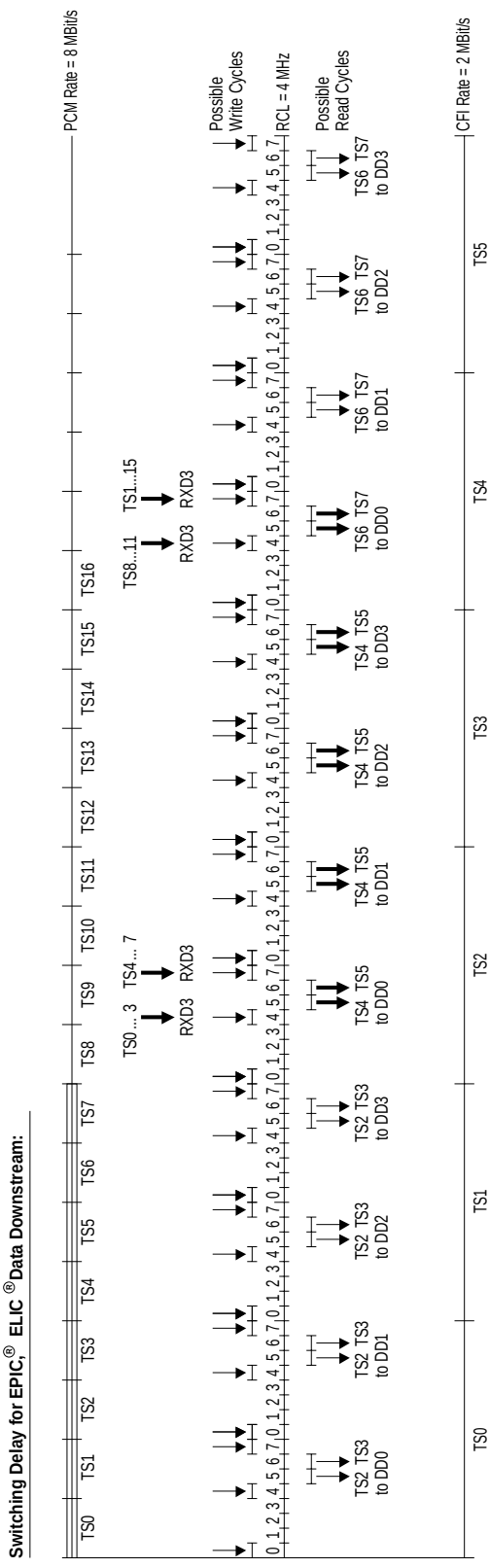
Shift only the PCM part of the figure (‘PCM line’ with the time slot numbers), relative to the rest of the figure, to the left.

If the CFI frame is shifted relative to the framing signal, then the CFI part, including the figure of the RCL, and all read and write cycle points are shifted left relative to the PCM part. If CBSR:CDS = 000 or 001, then the frame CFI part is shifted to the right.
The figure so produced should be processed as previously described.

*Note: If a bit shift has been installed while the PCM interface is already in the synchronous state, the following procedure has to be applied:*
*1.) Unsynchronize the PCM interface by writing an invalid number to register PBNR*
*2.) Resynchronize the PCM interface by writing the correct number to PBNR*

### 5.4.4.3 Example: Switching of Wide Band ISDN Channels with the ELIC®

The ELIC shall switch 6 B-channels of a digital subscriber to an 8 MBit/s PCM highway guaranteeing frame integrity. The system uses the IOM-2 interface to adapt to a multiple S-interface. No bit shift has to be applied. The tables below will help to determine the combination of input/output ports and time slots, that meet the requirements.




| Ports | PCM Input TS | Switched to CFI Output Time-Slots | | |
|---|---|---|---|---|
| RXD3 Switched to DD0 | 0...3 | 4...31 | 0...3 | - |
| | 4...7 | 6...31 | 0...5 | - |
| | 8...11 | 6...31 | 0...5 | - |
| | 12...15 | 8...31 | 0...7 | - |
| | 16...19 | 8...31 | 0...7 | - |
| | . | . | . | - |
| | . | . | . | - |
| | 96...99 | 28...31 | 0...27 | - |
| | 100...103 | 30...31 | 0...29 | - |
| | 104...107 | 30...31 | 0...29 | - |
| | . | . | . | - |
| | 120...123 | - | 2...31 | 0...1 |
| | 124...127 | - | 4...31 | 0...3 |
| RXD3 Switched to DD1...3 | 0...7 | 4...31 | 0...3 | - |
| | 8...15 | 6...31 | 0...5 | - |
| | 16...23 | 8...31 | 0...7 | - |
| | . | . | . | - |
| | . | . | . | - |
| | 96...103 | 28...31 | 0...27 | - |
| | 104...111 | 30...31 | 0...29 | - |
| | 112...119 | - | 0...31 | - |
| | 120...127 | - | 2...31 | 0...1 |
| Delay in [Frames] | | 0 | 1 | 2 |

**Configuration:** PCM Mode 2,
CFI Mode 0

PDC = 8 MHz, DCL 4 MHz
PCM Data Rate = 8 Mbit/s
CFI Data Rate = 2 Mbit/s
RCL = 4 MHz

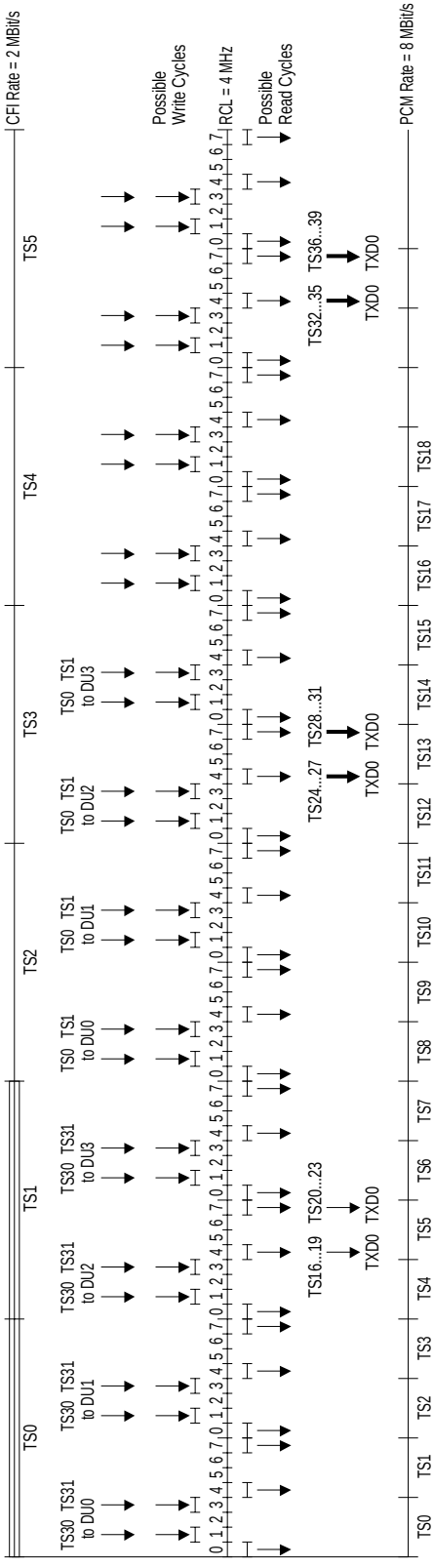
CTAR = 02
CBSR = 20
POFD = FO H
POFU = 18 H
PCSR = 00 H

No Bit Shift at PCM and CFI Interface

ITD08078

**Figure 99**

**Switching Delay for EPIC, ® ELIC® Data Upstream:**




| Ports | CFI Input TS | Switched to PCM Output Time-Slots | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DU3<br>Switched to TXD0 | 0...1<br>2...3<br>4...5<br>.<br>.<br>22...23<br>24...25<br>26...27<br>28...29<br>30...31 | 32...127<br>40...127<br>48...127<br>.<br>.<br>120...127<br>-<br>-<br>-<br>- | 0...31<br>0...39<br>0...47<br>.<br>.<br>0...119<br>0...127<br>8...127<br>16...127<br>24...127 | -<br>-<br>-<br>-<br>-<br>-<br>-<br>0...7<br>0...15<br>0...23 | | | | | | | | | | | | | | | | | | | | | | |
| DU0...2<br>Switched to TXD0 | 0...1<br>2...3<br>4...5<br>.<br>.<br>24...25<br>26...27<br>28...29<br>30...31 | 24...127<br>32...127<br>40...127<br>.<br>.<br>120...127<br>-<br>-<br>- | 0...23<br>0...31<br>0...39<br>.<br>.<br>0...119<br>0...127<br>8...127<br>16...127 | -<br>-<br>-<br>-<br>-<br>-<br>-<br>0...7<br>0...15 | | | | | | | | | | | | | | | | | | | | | | |
| Delay in [Frames] | | 0 | | 1 | | | | | | | | | | | | | | | | | | | | | | 2 |

**Figure 100**

## 5.5 Preprocessed Channels

The configurable interface (CFI) is at first sight a timeslot oriented serial interface similar to the PCM interface: a CFI frame contains a number of timeslots which can be switched to the PCM interface. But in addition to the switching functions, the CFI timeslots can also individually be configured as preprocessed channels. In this case, the contents of a CFI timeslot are directly, or after an eventual preprocessing, exchanged with the µP interface. The main application is the realization of IOM (ISDN Oriented Modular) and SLD (Subscriber Line Data) interfaces for the connection of subscriber circuits such as layer-1 transceivers (ISDN line cards) or codec filter devices (analog line cards). Also refer to **chapter 5.1.1**.

The preprocessing functions can be divided into 2 categories:

### Monitor/Feature Control (MF) Channels

The monitor channel in IOM and the feature control channel in SLD applications are handled by the MF handler. This MF handler consists of a 16 byte bidirectional FIFO providing intermediate storage for the messages to be transmitted or received. Internal microprograms can be executed in order to control the communication with the connected subscriber circuit according to the IOM or SLD protocol. The exchange of individual data is carried out with only one channel at a time. The MF handler must therefore be pointed to that particular subscriber address (CFI timeslot).

### Control/Signaling (CS) Channels

The access to the Command/Indication (C/I) channel of an IOM and to the signaling (SIG) channel of an SLD interface is realized by reading or writing to the corresponding control memory (CM) locations. In upstream direction, a change detection logic supervises the received C/I or SIG values on all CS channels and reports all changes via interrupt to the µP.

The MF and CS channel functions are inseparably linked to each other such that an MF channel must always be followed by a CS channel in the next following CFI timeslot. An MF channel must furthermore, be located on an even CFI timeslot, the associated CS channel must consequentially be always located on the following odd timeslot.

### 5.5.1 Initialization of Preprocessed Channels

The initialization of preprocessed channels is usually performed after the CM reset sequence during device initalization. Resetting the CM sets all CFI timeslots to unassigned channels (CM code ‘0000’). The initialization of preprocessed channels consists of writing appropriate CM codes to those CFI timeslots that should later be handled by the CS or MF handler.

The initialization or re-initialization of preprocessed channels can of course also be carried out during the operational phase of the device.

If the CFI shall be operated as a standard IOM-2 interface, for example, the CFI frame consists of 32 timeslots, numbered from 0 to 31 (see **figure 58**).

The B channels occupy timeslots 0 and 1 (IOM channel 0), 4 and 5 (IOM channel 1), 8 and 9 (IOM channel 2), and so on. The B channels are normally switched to the PCM interface and are programmed only if the actual switching function is required.

The monitor, D and C/I channels occupy timeslots 2 and 3 (IOM channel 0), 6 and 7 (IOM channel 1), 10 and 11 (IOM channel 2), and so on. These timeslots must be initialized in both upstream and downstream directions for the desired functionality. In order to speed up this initialization, the ELIC can be set into the CM initialization mode as described in **chapter 5.3.2**.

There are several options available to cover the different applications like switched D channel, 6 bit signaling, etc. It should be noted that each pair of timeslots can individually be set for a specific application and that the up- and downstream directions can also be set differently, if required.

#### D-Channel Handling Scheme by SACCO-A and D-Channel Arbiter

This option applies for IOM-2 channels where the even timeslot consists of an 8 bit monitor channel and the odd timeslot of a 2 bit D channel followed by a 4 bit C/I channel followed by the 2 monitor handshake bits MR and MX.

The monitor channel is handled by the MF handler according to the selection of handshake or non-handshake protocol. If the handshake option is selected (IOM-2), the MF handler controls the MR and MX bits according to the IOM-2 specification. If the no handshake option is selected (IOM-1), the MF handler sets both MR and MX bits to logical 1; the MR and MX bit positions can then, if required, be accessed together with the 4 bit C/I field via the even control memory address.

The information of the D-bits are passed to the arbiter in upstream direction where a decision is made whether the demanding D-channel is allowed to use the SACCO-A HDLC controller.

In downstream direction the SACCO-A sends D-channel information on a previously selected IOM-channel.

The 4 bit C/I channel can be accessed by the µP for controlling layer-1 devices, or by the ELIC arbiter to transmit the available/blocked information to the requesting HDLC controller.

In upstream direction each change in the C/I value is reported by interrupt to the µP and the CFI time slot address is stored in the CIFIFO (refer to **chapter 5.5.2**). A C/I change is detected if the value of the current CFI frame is different from the value of the previous frame i.e. after, at most, 125 µs.

To initialize two consecutive CFI timeslots for the arbiter D-Channel handling scheme, the CM codes as given in **table 43** must be used.

**Table 43**
**Control Memory Codes and Data for the Arbiter D-Channel Handling Cheme**

| CM Address | CM Code | CM Data |
|---|---|---|
| Even timeslot downstream | 1010 | 11 C/I $11_B$ |
| Odd timeslot downstream | 1011 | $XXXXXXXX_B$ |
| Even timeslot upstream | 1000 | XX C/I $XX_B$ |
| Odd timeslot upstream | 0000 | $XXXXXXXX_B$ |

### Decentral D-Channel Handling Scheme

This option applies for IOM channels where the even timeslot consists of an 8 bit monitor channel and the odd timeslot of a 2 bit D-Channel followed by a 4 bit C/I channel followed by the 2 monitor handshake bits MR and MX.

The monitor channel is handled by the MF handler according to the selection of handshake or non-handshake protocol. If the handshake option is selected (IOM-2), the MF handler controls the MR and MX bits according to the IOM-2 specification. If the no handshake option is selected (IOM-1), the MF handler sets both MR and MX bits to logical 1; the MR and MX bit positions can then, if required, be accessed together with the 4 bit C/I field via the even control memory address.

The D-Channel is not processed at all, i.e. the input in upstream direction is ignored and the output in downstream direction is set to high impedance. External D-Channel controllers, e.g. 2 × IDECs PEB 2075, can then be connected to each IOM interface in order to realize **decentral D-Channel** processing.

The 4 bit C/I channel can be accessed by the µP for controlling layer-1 devices. In upstream direction each change in the C/I value is reported by interrupt to the µP and the CFI timeslot address is stored in the CIFIFO (refer to **chapter 5.5.2**). A C/I change is detected if the value of the current CFI frame is different from the value of the previous frame i.e. after at most 125 µs.

To initialize two consecutive CFI timeslots for the decentral D-Channel handling scheme, the CM codes as given in **table 44** must be used.

**Table 44**
**Control Memory Codes and Data for the Decentral D-Channel Handling Scheme**

| CM Address | CM Code | CM Data |
|---|---|---|
| Even timeslot downstream | 1000 | 11 C/I $11_B$ |
| Odd timeslot downstream | 1011 | $XXXXXXXX_B$ |
| Even timeslot upstream | 1000 | XX C/I $XX_B$ |
| Odd timeslot upstream | 0000 | $XXXXXXXX_B$ |

Application hint: If the D-Channel is idle and if it is required to transmit a 2 bit idle code in the D-Channel (e.g. during the layer-1 activation or for testing purposes), the 6 bit signaling handling scheme can be selected for the downstream direction. The 2 D bits together with the 4 C/I bits can then be written to via the even control memory address. If the high impedance state is needed again, the decentral D-Channel scheme has to be selected again.

**Example**

In CFI mode 0, timeslots 2 and 3 of port 3 are to be initialized for decentral D-Channel handling:

W:MADR = $1100\ 0011_B$ ; C/I value ‘0000’
W:MAAR = $0000\ 1110_B$ ; downstream even TS, port 3 timeslot 2
W:MACR = $0111\ 1000_B$ ; write CM code + data fields, CM code ‘1000’

W:MADR = $XXXX\ XXXX_B$ ; don’t care
W:MAAR = $0000\ 1111_B$ ; downstream odd TS, port 3 timeslot 3
W:MACR = $0111\ 1011_B$ ; write CM code + data fields, CM code ‘1011’

W:MADR = $1111\ 1111_B$ ; expected C/I value ‘1111’
W:MAAR = $1000\ 1110_B$ ; upstream even TS, port 3 timeslot 2
W:MACR = $0111\ 1000_B$ ; write CM code + data fields, CM code ‘1000’

W:MADR = $XXXX\ XXXX_B$ ; don’t care
W:MAAR = $1000\ 1111_B$ ; upstream odd TS, port 3 timeslot 3
W:MACR = $0111\ 0000_B$ ; write CM code + data fields, CM code ‘0000’

After these programming steps, the control memory will have the following content:

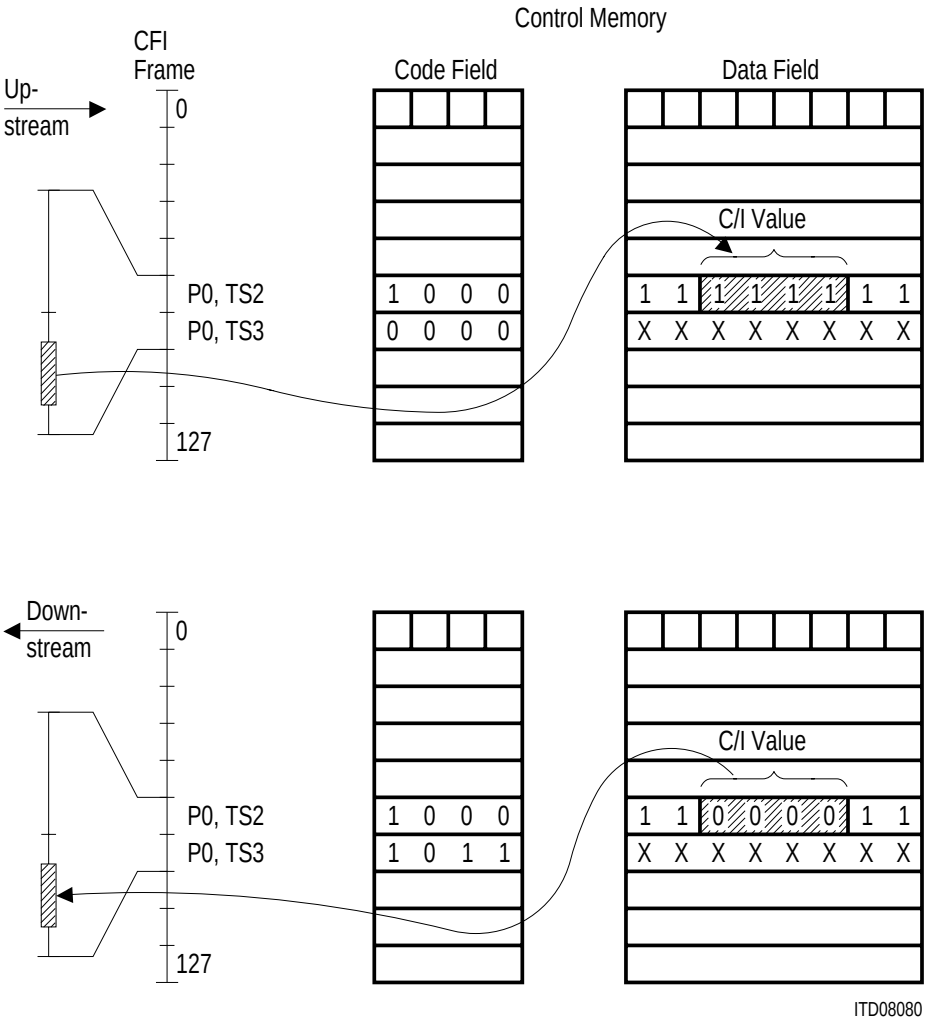



**Figure 101**
**Control Memory Contents for Decentral D-Channel Handling**

### Central D-Channel Handling Scheme

This option applies for IOM channels where the even timeslot consists of an 8 bit monitor channel and the odd timeslot of a 2 bit D-Channel followed by a 4 bit C/I channel followed by the 2 monitor handshake bits MR and MX.

The monitor channel is handled by the MF handler according to the selected protocol, handshake or non-handshake. If the handshake option is selected (IOM-2), the MF handler controls the MR and MX bits according to the IOM-2 specification. If the non-handshake option is selected (IOM-1), the MF handler sets both MR and MX bits to

logical 1; the MR and MX bit positions can then, if required, be accessed together with the 4 bit C/I field via the even control memory address.

The D-Channel can be switched as a 16 kbps channel to and from the PCM interface in order to be handled by a **centralized D-Channel** processing unit.

The 4 bit C/I channel can be accessed by the µP for controlling layer-1 devices. In the upstream direction each change in the C/I value is reported by interrupt to the µP and the CFI timeslot address is stored in the CIFIFO (refer to **chapter 5.5.2**). A C/I change is detected if the value of the current CFI frame is different from the value of the previous frame i.e. after at most 125 µs.

To initialize two consecutive CFI timeslots for the decentral D-Channel handling scheme, the CM codes as given in **table 45** must be used.

**Table 45**
**Control Memory Codes and Data for the Central D-Channel Handling Scheme**

| CM Address | CM Code | CM Data |
|---|---|---|
| Even timeslot downstream | 1010 | 11 C/I $11_B$ |
| Odd timeslot downstream | Switch. code | Pointer to PCM TS |
| Even timeslot upstream | 1000 | XX C/I $XX_B$ |
| Odd timeslot upstream | Switch. code | Pointer to PCM TS |

The switching codes specify the PCM sub-timeslot positions of the 16 kBit/s transfer. Note that the 2 D bits are always located on bits 7 … 6 of a CFI timeslot, the CSCR:SC#1, SC#0 bits must therefore be set to 00 (see **chapter 5.4.2**).

**Table 46**
**Control Memory Codes for the Switching a 16 kBit/s CFI Channel to or from the PCM Interface**

| Transferred Channel PCM Bit Positions | Downstream CM Codes | Upstream CM Codes |
|---|---|---|
| Unassigned channel | 1011[1)] | 0000 |
| 16 kBit/s/ bits 7 … 6 | 0111 | 0111 |
| 16 kBit/s/ bits 5 … 4 | 0110 | 0110 |
| 16 kBit/s/ bits 3 … 2 | 0101 | 0101 |
| 16 kBit/s/ bits 1 … 0 | 0100 | 0100 |

1) This code sets the D bits to high impedance

Application hints: 1) If the D channel is idle and if it is required to transmit a 2 bit idle code in the D channel (e.g. during the layer-1 activation or for testing purposes), the 6 bit signaling handling scheme can be selected for the downstream direction. The 2 D bits together with the 4 C/I bits can then be written to via the even control memory address. If the high impedance state is needed again, the decentral D channel scheme has to be selected again.

2) The central D channel scheme has primarily been designed to switch the 16 kBit/s D channel to the PCM interface and to process the C/I channel by the local µP. For some applications however, it is advantageous to switch the 2 D bits together with the 4 C/I bits transparently to and from the PCM interface. The monitor channel shall, however, still be handled by the internal MF handler. This function might be useful if two layer-1 transceivers, operated in “Repeater Mode”, shall be connected via a PCM link. For these applications, the odd control memory address is written with the 64 kBit/s switching code ‘0001’, the CM data field pointing to the desired PCM timeslot. Since also the MR and MX bits are being switched, these must be carefully considered: in upstream direction the two least significant bits of the PCM timeslot can be set to high impedance via the tristate field; in downstream direction the two least significant bits of the PCM timeslot must be received at a logical 1 level since these bits will be logical ANDed at the CFI with the downstream MR and MX bits generated by the MF handler.

**Example**

In CFI and PCM modes 0, CFI timeslots 10 and 11 of port 1 shall be initialized for central D channel handling, the downstream D channel shall be switched from PCM port 0, TS5, bits 5 … 4 and the upstream D channel shall be switched to PCM port 2, TS8, bits 3 … 2:

```
W:MADR  =  1100 0011B     ; C/I value ‘0000’
W:MAAR  =  0010 1010B     ; downstream even TS, port 1 timeslot 10
W:MACR  =  0111 1010B     ; write CM code + data fields, CM code ‘1010’

W:MADR  =  0001 0001B     ; pointer to PCM port 0, TS5
W:MAAR  =  0010 1011B     ; downstream odd TS, port 1 timeslot 11
W:MACR  =  0111 0110B     ; write CM code + data fields, CM code ‘0110’

W:MADR  =  1111 1111B     ; expected C/I value ‘1111’
W:MAAR  =  1010 1010B     ; upstream even TS, port 1 timeslot 10
W:MACR  =  0111 1000B     ; write CM code + data fields, CM code ‘1000’
```

W:MADR = $1010\ 0100_B$ ; pointer to PCM port 2, TS8
W:MAAR = $1010\ 1011_B$ ; upstream odd TS, port 1 timeslot 11
W:MACR = $0111\ 0101_B$ ; write CM code + data fields, CM code '0101'
W:MADR = $0000\ 0010_B$ ; set bits 3 … 2 to low Z and rest of timeslot to high Z
W:MAAR = $1010\ 0100_B$ ; pointer to PCM port 2, TS8
W:MACR = $0110\ 0000_B$ ; write DM CF, single channel tristate command

After these programming steps, the ELIC memory will have the following contents:

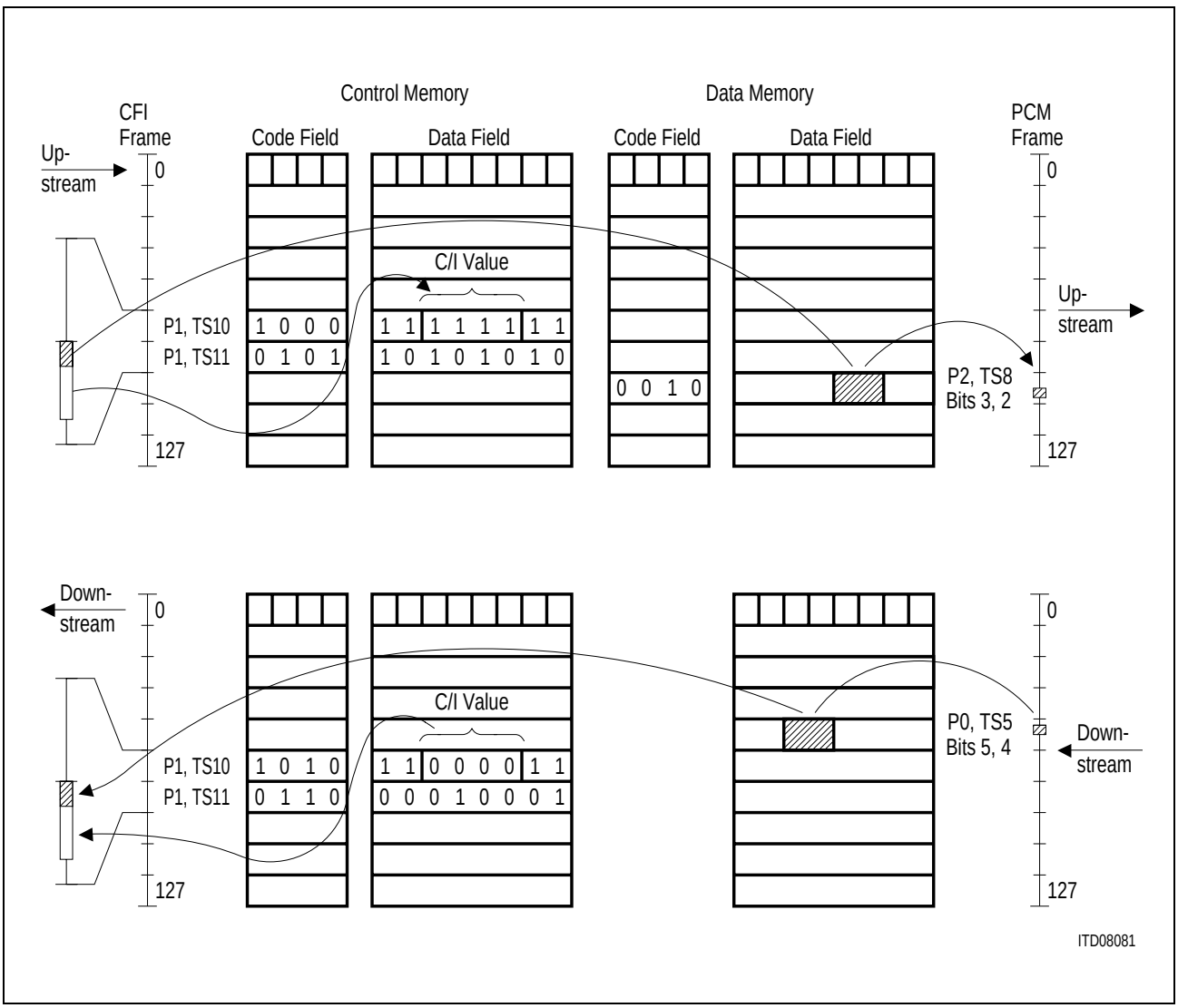



**Figure 102**
**Control Memory Contents for Central D-Channel Handling**

### 6-Bit Signaling Channel Scheme

This option is intended for IOM channels where the even timeslot consists of an 8 bit monitor channel and the odd timeslot of a 6 bit signaling channel followed by the 2 monitor handshake bits MR and MX.

The monitor channel is handled by the MF handler according to the selected protocol, handshake or non-handshake. If the handshake option is selected (IOM-2), the MF handler controls the MR and MX bits according to the IOM-2 specification. If the non-handshake option is selected (IOM-1), the MF handler sets both MR and MX bits to logical 1; the MR and MX bit positions can then, if required, be accessed together with the 6 bit SIG field via the even control memory address.

The 6 bit SIG channel can be accessed by the µP for controlling codec filter devices. In upstream direction each valid change in the SIG value is reported by interrupt to the µP and the CFI timeslot address is stored in the CIFIFO (refer to **chapter 5.5.1**). The change detection mechanism consists of a double last look logic with a programmable period.

To initialize two consecutive CFI timeslots for the 6 bit signaling channel scheme, the CM codes as given in **table 47** must be used:

**Table 47**
**Control Memory Codes and Data for the 6 Bit Signaling Channel Handling Scheme**

| CM Address | CM Code | CM Data |
|---|---|---|
| Even timeslot downstream | 1010 | SIG $11_B$ |
| Odd timeslot downstream | 1011 | $XXXXXXXX_B$ |
| Even timeslot upstream | 1010 | actual value $XX_B$ |
| Odd timeslot upstream | 1010 | stable value $XX_B$ |

Application hint: For some applications it is useful to switch the 6 SIG bits transparently to and from the PCM interface. The monitor channel shall, however, still be handled by the internal MF handler. For this purpose, a slightly modified central D channel scheme can be used. This mode, which has primarily been designed to switch the 16 kBit/s D channel to the PCM interface, can be modified as follows: the odd control memory address is written with the 64 kBit/s switching code ‘0001’, the CM data field pointing to the desired PCM timeslot. Since the MR and MX bits are being switched, these must be carefully considered: in upstream direction the two least significant bits of the PCM timeslot can be set to high impedance via the tristate field; in downstream direction the two least significant bits of the PCM timeslot must be received at a logical 1 level since these bits will be logical ANDed at the CFI with the downstream MR and MX bits generated by the MF handler.

**Example**

In CFI mode 0, timeslots 2 and 3 of port 0 shall be initialized for 6 bit signaling channel handling:

| | | | |
|---|---|---|---|
| W:MADR | = | $0100\ 0111_B$ | ; SIG value ‘010001’ |
| W:MAAR | = | $0000\ 1000_B$ | ; downstream even TS, port 0 timeslot 2 |
| W:MACR | = | $0111\ 1010_B$ | ; write CM code + data fields, CM code ‘1010’ |
| W:MADR | = | $XXXX\ XXXX_B$ | ; don’t care |
| W:MAAR | = | $0000\ 1001_B$ | ; downstream odd TS, port 0 timeslot 3 |
| W:MACR | = | $0111\ 1011_B$ | ; write CM code + data fields, CM code ‘1011’ |
| W:MADR | = | $1101\ 1111_B$ | ; expected SIG value ‘110111’ |
| W:MAAR | = | $1000\ 1000_B$ | ; upstream even TS, port 0 timeslot 2 |
| W:MACR | = | $0111\ 1010_B$ | ; write CM code + data fields, CM code ‘1010’ |
| W:MADR | = | $1101\ 1111_B$ | ; expected SIG value ‘110111’ |
| W:MAAR | = | $1000\ 1001_B$ | ; upstream odd TS, port 0 timeslot 3 |
| W:MACR | = | $0111\ 1010_B$ | ; write CM code + data fields, CM code ‘1010’ |

After these programming steps, the ELIC memory will have the following contents:

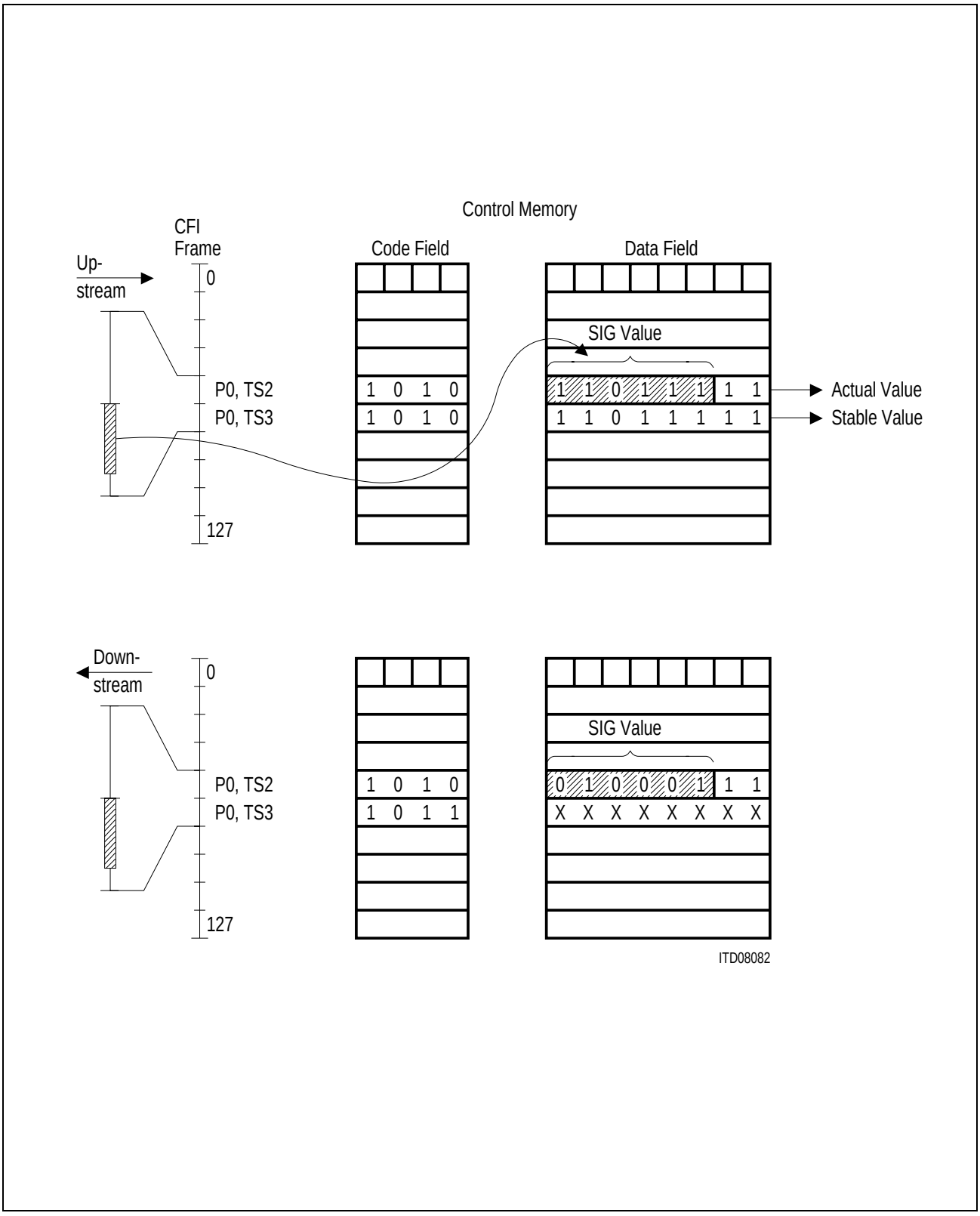



**Figure 103**
**Control Memory Contents for 6-Bit Signaling Channel Handling**

### 8-Bit Signaling Scheme

This option is intended for SLD channels where the even timeslot consists of an 8 bit feature control channel and the odd timeslot of an 8 bit signaling channel.

The feature control channel is handled by the MF handler according to the selected protocol, handshake or non-handshake. Note that only the non-handshake mode makes sense in SLD applications.

The 8 bit SIG channel can be accessed by the µP for controlling codec filter devices. In upstream direction each valid change in the SIG value is reported by interrupt to the µP and the CFI timeslot address is stored in the CIFIFO (refer to **chapter 5.5.2**). The change detection mechanism consists of a double last look logic with a programmable period.

To initialize two consecutive CFI timeslots for the 8 bit signaling channel scheme, the CM codes as given in **table 48** must be used:

**Table 48**
**Control Memory Codes and Data for the 8-Bit Signaling**

| CM Address | CM Code | CM Data |
|---|---|---|
| Even timeslot downstream | 1010 | $SIG_B$ |
| Odd timeslot downstream | 1011 | $XXXXXXXX_B$ |
| Even timeslot upstream | 1011 | actual $value_B$ |
| Odd timeslot upstream | 1011 | stable $value_B$ |

### Example

In CFI mode 3, downstream timeslots 2 and 3 and upstream timeslots 6 and 7 of port 0 shall be initalized for 8 bit signaling channel handling:

```
W:MADR  =  0100 0101B     ; SIG value '0100 0101'
W:MAAR  =  0001 0000B     ; downstream even TS, port 0 timeslot 2
W:MACR  =  0111 1011B     ; write CM code + data fields, CM code '1011'

W:MADR  =  XXXX XXXXB     ; don't care
W:MAAR  =  0001 0001B     ; downstream odd TS, port 0 timeslot 3
W:MACR  =  0111 1011B     ; write CM code + data fields, CM code '1011'

W:MADR  =  1101 0110B     ; expected SIG value '1101 0110'
W:MAAR  =  1011 0000B     ; upstream even TS, port 0 timeslot 6
W:MACR  =  0111 1011B     ; write CM code + data fields, CM code '1011'

W:MADR  =  1101 0110B     ; expected SIG value '1101 0110'
W:MAAR  =  1011 0001B     ; upstream odd TS, port 0 timeslot 7
W:MACR  =  0111 1011B     ; write CM code + data fields, CM code '1011'
```

## Summary of “Preprocessed Channel” Codes

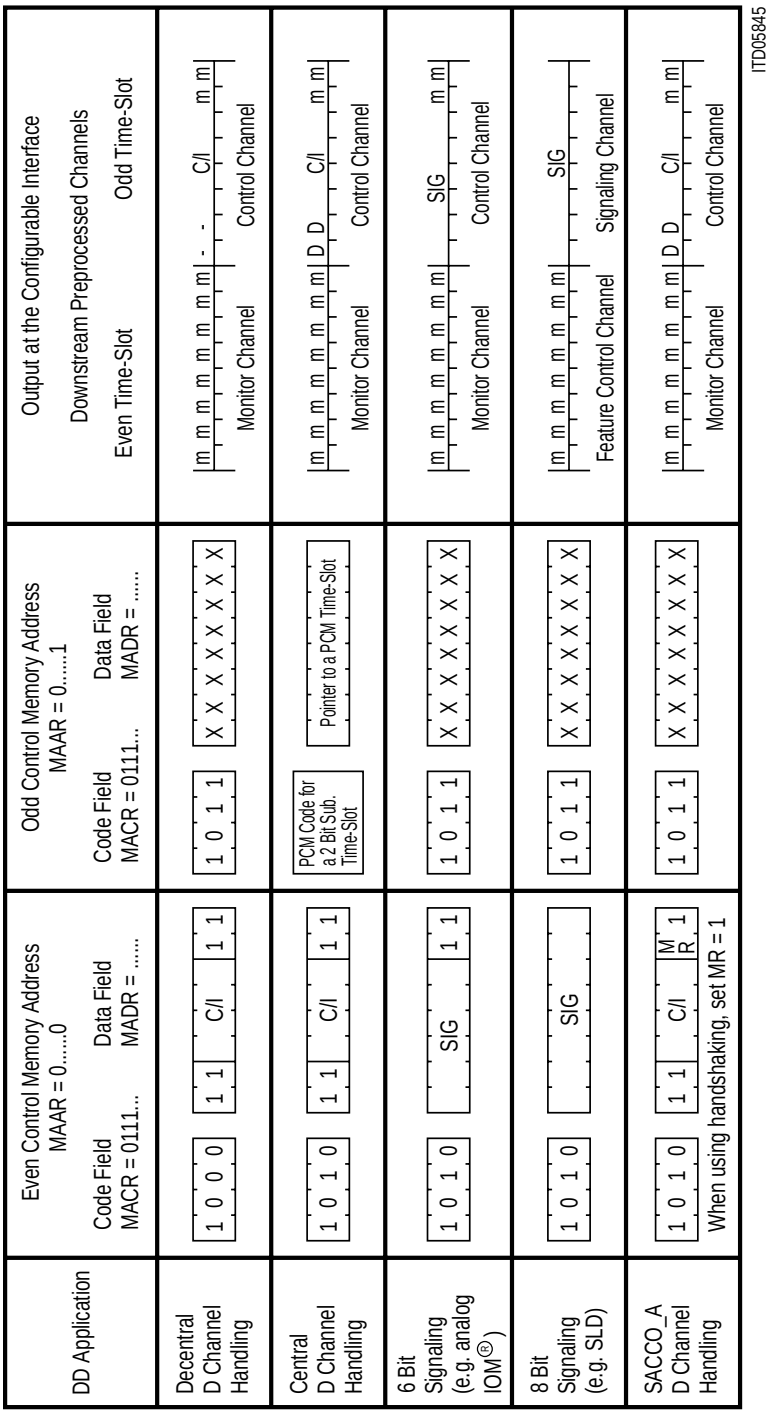



**Figure 104 a**
**“Preprocessed Channel” Codes**

m : Monitor channel bits, these bits are treated by the monitor/feature control handler

- : Inactive sub. time-slot, in downstream direction these bits are tristated (OMDR : COS = 0) or set to logical 1 (OMDR :COS = 1)

C/I : Command/Indication channel, these bits are exchanged between the CFI in/output and the CM data field. A change of the C/I bits in upstream direction causes an interrupt (ISTA : SFI). The address of the change is stored in the CIFIFO

D : D channel, these D channel bit switched to and from the PCM interface, or handled by the SACCO_A, it the D channel arbiter is enabled.

SIG actual value stable value : Signaling Channel, these bits are exchanged between the CFI in/output and tne CM data field. The SIG value which was present in the last frame is stored as the actual value in the even address CM location. The stable value is updated if a valid change in the actual value has been detected according to the last look algorithm. A change of the SIG stable value in upstream direction causes an interrupt (ISTA : CFI). The address of the change is stored in the CIFIFO.

**Figure 104 b**
**“Preprocessed Channel” Codes**

### 5.5.2 Control/Signaling (CS) Handler

If the configurable interface (CFI) of the ELIC is operated as IOM or SLD interface, it is necessary to communicate with the connected subscriber circuits such as layer-1 transceivers (ISDN line cards) or codec filter devices (analog line cards) over the Command/Indication (C/I) or the signaling (SIG) channel. In order to simplify this task the ELIC has implemented the **Control/Signaling Handler** (CS Handler).

In downstream direction, the 4, 6 or 8 bit C/I or SIG value can simply be written to the Control Memory data field which will then be repeatedly transmitted in every frame to the subscriber circuit until a new value is loaded.

Note that the downstream C/I or SIG value must always be written to the **even** CM address in order to be transmitted in the subsequent **odd** CFI timeslot!

In upstream direction a change detection mechanism is active to search for changes in the received C/I or SIG values. Upon a change, the address of the involved subscriber is stored in a 9 byte deep FIFO (CIFIFO) and an interrupt (ISTA:SFI) is generated. The µP can then first determine the CM address by reading the FIFO before reading the new C/I or SIG value out of the Control Memory. The address FIFO serves to increase the latency time for the µP to react to SFI interrupts. If several C/I or SIG changes occur before the µP executes the SFI interrupt handling routine, the addresses of the first 9 changes are stored in the CIFIFO and the corresponding C/I or SIG values are stored in the control memory (CM). If more than 9 changes occur before the µP reads the CIFIFO, these additional changes are no longer updated in the control memory. This is to prevent any loss of change information. These additional changes remain pending at the serial interface. As soon as the µP reads the CIFIFO, and thus, empties locations of the FIFO, these pending changes are sequentially written to the CM and the corresponding addresses to the FIFO. It is thus ensured that no change information is lost even if, for example, all 32 subscribers simultaneously generate a change in their C/I or SIG channel!

CFI timeslots which should be processed by the CS handler must first be initialized as MF/CS channels with appropriate codes in the Control Memory code field (refer to **chapter 5.5.1**).

#### 5.5.2.1 Registers used in Conjunction with the CS Handler

In detail, the following register bits are used in conjunction with the CS handler:

**Signaling FIFO** read reset value: $0XXXXXXX_B$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| CIFIFO | SBV | SAD6 | SAD5 | SAD4 | SAD3 | SAD2 | SAD1 | SAD0 |

The 9 byte deep CIFIFO stores the addresses of CFI timeslots in which a C/I and/or a SIG value change has taken place. This address information can then be used to read the actual C/I or SIG value from the Control Memory.

SBV: Signaling Byte Valid; if SBV = 1, the SAD6 … 0 bits indicate a valid subscriber address. The polarity of SBV is chosen such that the whole 8 bits of the CIFIFO can be copied to the MAAR register in order to read the upstream C/I or SIG value from the Control Memory.

SAD6 … 0: Subscriber Address bits 6 … 0; The CM address which corresponds to the CFI timeslot where a C/I or SIG value change has taken place is encoded in these bits. For C/I channels SAD6 … 0 point to an even CM address (C/I value), for SIG channels SAD6 … 0 point to an odd CM address (stable SIG value).

**Timer Register** write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| TIMR | SSR | TVAL6 | TVAL5 | TVAL4 | TVAL3 | TVAL2 | TVAL1 | TVAL0 |

The ELIC timer can be used for 3 different purposes: timer interrupt generation (ISTA:TIG), FSC multiframe generation (CMD2:FC2 … 0 = 111), and last look period generation.

In case of last look period generation, the following functions are provided:

SSR: Signaling Sampling Rate; If SSR = 1, the last look period is fixed to 125 µs, i.e. the timer is not used at all for the last look logic. The value programmed to TVAL has then no influence on the last look period. The timer can then still be used for timer interrupt generation, and/or FSC multiframe generation, with a period as defined by TVAL6 … 0. If SSR = 0, the last look period is defined by TVAL6 … 0. Note that if the timer is used, it must also be started with CMDR:ST = 1.

TVAL6 … 0: Timer Value bits 6 … 0; the timer period, equal to (1 + TVAL6 … 0) × 250 µs, is programmed here. It can thus be adjusted within the range of 250 µs up to 32 ms.

The timer is started as soon as CMDR:ST is set to 1 and stopped by writing the TIMR register or by selecting OMDR:OMS0 = 0.

If the timer is used to generate the last look period, it can still be used for timer interrupt generation and/or FSC multiframe generation if it is acceptable that all three applications use the same timer value.

**Command Register EPIC®** write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| CMDR_E | 0 | **ST** | TIG | **CFR** | MFT1 | MFT0 | MFSO | MFFR |

Writing a logical 1 to a CMDR_E register bit starts the respective operation.

The signaling handler uses two command bits:

ST: Start Timer; must be set to 1 if the last look period is defined by TIMR:TVAL6 … 0, i.e. if TIMR:SSR = 0. Note that if TIMR:SSR = 1, the timer need not be started.

CFR: CIFIFO Reset; setting CFR to logical 1 resets the signaling FIFO within 2 RCL periods, i.e. all entries and the ISTA:SFI bit are cleared.

**Status Register EPIC®** read reset value: $05_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| STAR_E | MAC | **TAC** | PSS | MFTO | MFAB | MFAE | MFRW | MFFE |

The status register STAR_E displays the current state of certain events within the ELIC. The STAR_E register bits do not generate interrupts and are not modified by reading STAR_E.

The following bit is indirectly used by the signaling handler:

TAC: Timer Active; the timer is running if TAC is set to logical 1, the timer is not running if TAC is set to logical 0.

Note that the timer is only necessary for signaling channels (not C/I) and when using a last look period greater or equal to 250 µs.

**Interrupt Status Register EPIC®** read reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| ISTA_E | TIN | **SFI** | MFFI | MAC | PFI | PIM | SIN | SOV |

The ISTA_E register should be read after an interrupt in order to determine the interrupt source.

In connection with the signaling handler one maskable (MASK_E) interrupt bit is provided by the ELIC in the ISTA_E register:

SFI: Signaling FIFO Interrupt; This bit is set to logical 1 if there is at least one valid entry in the CIFIFO indicating a change in a C/I or SIG channel. Reading ISTA_E does not clear the SFI bit. Instead SFI is cleared (logical 0) if the CIFIFO is empty which can be accomplished by reading all valid entries of the CIFIFO or by resetting the CIFIFO by setting CMDR:CFR to 1.

Note that the MASK_E:SFI bit only disables the interrupt pin ($\overline{INT}$); the ISTA_E:SFI bit will still be set to logical 1.

### 5.5.2.2 Access to Downstream C/I and SIG Channels

If two consecutive downstream CFI timeslots, starting with an even timeslot number, are programmed as MF and CS channels, the µP can write a 4, 6 or 8 bit wide C/I or SIG value to the even addressed downstream CM data field. This value will then be transmitted repeatedly in the odd CFI timeslot until a new value is loaded.

This value, first written into MADR, can be transferred to the CM data field using the memory operation codes MACR:MOC = 111X or MACR:MOC = 1001 (refer to **chapter 5.3.3.3**).

The code MACR:MOC = 111X applies if the code field has not yet been initialized with a CS channel code. Writing to MACR with MACR:RWS = 0 will then copy the CS channel code written to MACR:CMC3 … CMC0 to the CM code field and the value written to MADR to the CM data field. The CM address (CFI timeslot) is specified by MAAR according to **figure 84**.

The code MACR:MOC = 1001 applies if the code field has already been properly initialized with a CS channel code. In this case only the MADR content will be copied to the CM data field addressed by MAAR.

The value written to MADR should have the following format:

4 bit C/I value: MADR = 1 1 _ _ _ _ 1 $1_B$
6 bit SIG value: MADR = _ _ _ _ _ _ 1 $1_B$
8 bit SIG value: MADR = _ _ _ _ _ _ _ _ B

**Examples**

In CFI mode 0 the downstream timeslots 6 and 7 of port 2 shall be initialized as MF and CS channels, 6 bit signaling scheme. The initialization value shall be ‘010101’:

W:MADR = $0101\ 0111_B$ ; SIG value ‘010101’
W:MAAR = $0001\ 1100_B$ ; downstream, port 2, timeslot 6
W:MACR = $0111\ 1010_B$ ; write CM code + data fields, CM code ‘1010’

W:MADR = $XXXX\ XXXX_B$ ; don’t care
W:MAAR = $0001\ 1101_B$ ; downstream, port 2, timeslot 7
W:MACR = $0111\ 1011_B$ ; write CM code + data fields, CM code ‘1011’

The above programming sequence can for example be performed during the initialization phase of the ELIC. Once the CFI timeslots have been loaded with the appropriate codes ('1010' in timeslot 6 and ‘1011’ in timeslot 7), an access to the downstream SIG channel (timeslot 7) can be accomplished simply by writing a new value to the address of timeslot 6:

W:MADR = $1100\ 1111_B$ ; new SIG value ‘110011’
W:MAAR = $0001\ 1100_B$ ; downstream, port 2, timeslot 6
W:MACR = $0100\ 1000_B$ ; write CM DF, MOC = 1001

### 5.5.2.3 Access to the Upstream C/I and SIG Channels

If two consecutive upstream CFI timeslots, starting with an even timeslot number, are programmed as MF and CS channels, the µP can read the received 4, 6 or 8 bit C/I or SIG values simply by reading the upstream CM data field.

Two cases can be distinguished:

When a 4 bit Command/Indication handling scheme is selected, the C/I value received in the odd CFI timeslot can be read from the even CM address. This value is sampled in each frame (every 125 µs). Each change is furthermore indicated by an ISTA_E:SFI interrupt and the address of the corresponding even CM location is stored in the CIFIFO. Since the MSB of the CIFIFO is set to 1 for a valid entry (SBV = 1), the value read from the CIFIFO can directly be copied to MAAR in order to read the upstream CM data field which also requires an MSB set to 1 (U/D = 1).

When a 6 or 8 bit signaling scheme is selected, the received SIG value is sampled at intervals of 125 µs or (TVAL + 1) $\times$ 250 µs and stored as the “actual value” at the even CM address. The µP can access the actual value simply by reading this even CM data field location. Additionally, a ‘stable value’, based on the double last look algorithm is generated: in order to assure that erroneous bit changes at the sampling time point do

not initiate a definite change, the values of two consecutive sampling points are compared with the current old stable value. The stable value is then only updated if both new values are identical and differ from the old stored value. The stable value can be read from the odd CM data field location. Each change in the stable value is furthermore indicated by an ISTA_E:SFI interrupt and the address of the corresponding odd CM location is stored in the CIFIFO. Since the MSB of the CIFIFO is set to 1 for a valid entry (SBV = 1), the value read from the CIFIFO can directly be copied to MAAR in order to read the upstream CM data field, which also requires an MSB set to 1 (U/D = 1).

*Note: The sampling interval is selected in the TIMR register (refer to **chapter 5.5.2.1**). If the sampling interval is set to 125 µs (TIMR:SSR = 1), it is not necessary to start the timer to operate the change detection logic. If, however, the last look period is determined by TIMR:TVAL6 … 0 (TIMR:SSR = 0) it is required to start the timer (CMDR:ST = 1) to operate the change detection logic and to generate SFI interrupts.*

### Examples

In CFI mode 0 the upstream timeslots 6 and 7 of port 2 shall be initialized as MF and CS channels, 6 bit signaling scheme, the expected value from the codec after power up shall be ‘011101’:

```
W:MADR  =  0111 0111B   ; expected actual value ‘011101’
W:MAAR  =  1001 1100B   ; upstream, port 2, timeslot 6
W:MACR  =  0111 1010B   ; write CM code + data fields, CM code ‘1010’
W:MADR  =  0111 0111B   ; expected stable value ‘011101’
W:MAAR  =  1001 1101B   ; upstream, port 2, timeslot 7
W:MACR  =  0111 1010B   ; write CM code + data fields, CM code ‘1010’
```

The above programming sequence can for example be performed during the initialization phase of the ELIC. At this stage the CFI is not operational (OMDR = $80_H$), i.e. the values received at the CFI are ignored.

If the expected value ‘011101’ is actually received upon activation of the CFI (e.g. OMDR = EEH), no interrupt will be generated at this moment. But the change detection is now enabled and each valid change in the received SIG value (e.g. new value ‘001100’) will generate an interrupt, with the address being stored in the CIFIFO. The reaction of the µP to such an event would then look like this:

R:ISTA_E = $0100\ 0000_B$ ; SFI interrupt
R:CIFIFO = $1001\ 1101_B$ ; address of upstream, port 2, timeslot 7
W:MAAR = $1001\ 1101_B$ ; copy the address from CIFIFO to MAAR
W:MACR = $1100\ 1000_B$ ; read back command for CM DF, MOC = 1001
wait for STAR_E:MAC = 0
R:MADR = $0011\ 00XX_B$ ; read new SIG value (e.g. 001100)
wait for further ISTA_E:SFI interrupts

### 5.5.3 Monitor/Feature Control (MF) Handler

If the configurable interface CFI of the ELIC is configured as IOM or SLD interface, it is necessary to communicate with the connected subscriber circuits such as layer-1 transceivers (ISDN line cards) or codec filter devices (analog line cards) over the monitor channel (IOM) or feature control channel (SLD). In order to simplify this task the ELIC has implemented the **Monitor/Feature Control (MF) Handler** which autonomously controls and supervises the data transfer via these channels.

The communication protocol used in an MF channel is interface and subscriber circuit specific.

Three cases can be distinguished:

#### IOM®-2 Interface Protocol

In this case the monitor channel protocol is a handshake procedure used for high speed information exchange between the ELIC and other devices such as the IEC-Q (PEB 2091), SBCX (PEB 2081) or SICOFI2 (PEB 2260).

The monitor channel operates on an asynchronous basis.While data transfers on the IOM-2 interface take place synchronized to the IOM frame, the flow of data is controlled by a handshake procedure based on the monitor channel receive (MR) and the monitor channel transmit (MX) bits located at the end of the fourth timeslot of the respective IOM-2 channel.

For the transmission of a data byte for example, the data is placed onto the downstream monitor channel and the MX bit is activated. This byte will then be transmitted repeatedly once per 8 kHz frame until the receiver acknowledges the transfer via the upstream MR bit.

A detailed description of the IOM-2 monitor channel operation can be found in the ‘IOM-2 Interface Reference Guide’.

### IOM®-1 Interface Protocol

In this case the monitor channel protocol is a non handshake procedure which can be used to exchange one byte of information at a time between the ELIC and a layer-1 device such as the IBC (PEB 2095) or the IEC-T (PEB 2090).

Data bytes to be transmitted are sent once in the downstream monitor channel. Since the monitor channel is idle (FF) when no data is being transmitted, the receiving device accepts only valid data bytes which are different from FF. If a message shall be sent back to the ELIC, this must occur in the frame following the frame of reception.

### SLD Interface Protocol

The transfer of control information over the feature control channel of an SLD interface e.g. for programming the coefficients to a SICOFI (PEB 2060) device is also performed without a handshake procedure. Data is transmitted and received synchronous to the 8 kHz frame at a speed of one data byte per frame.

The MF handler of the ELIC supports all three kinds of protocols. A bidirectional 16 byte FIFO, the MFFIFO, serves as data buffer for outgoing and incoming MF messages in all protocol modes. This implies that the MF communication is always performed on a half-duplex basis.

Differentiation between IOM-2 and IOM-1/SLD modes is made via the MF Protocol Selection bit **MFPS** in the Operation Mode Register **OMDR**.

Since the IOM-1 and SLD protocols are very similar, they are treated by the ELIC in exactly the same way i.e. without handshake protocol. The only processing difference concerns the involved upstream timeslot when receiving data:

When configured as IOM interface (CFI modes 0, 1 or 2), the CFI ports consist of separate upstream (DU) and downstream (DD) lines. In this case MF data is transmitted on DD and received on DU of the **same** CFI **timeslot.**

When configured as SLD interface (CFI mode 3), the CFI ports consist of bidirectional lines (SIP). The first four timeslots of the frame are used as downstream timeslots and the last four as upstream timeslots. In this case the MF data is transmitted in the downstream feature control timeslot and received on the **same** CFI **line** but four timeslots later in the upstream feature control timeslot.

CFI timeslots which should be processed by the MF handler must first be initialized as MF/CS channels with appropriate codes in the Control Memory Code Field (refer to **chapter 5.5.1**).

Except for broadcast operation, communication over the MF channel is only possible with one subscriber circuit at a time. The MF handler must therefore be pointed to that particular timeslot via the address register **MFSAR**.

Normally MF channel transfers are initiated by the ELIC (master). The subscriber circuits (slaves) will only send back monitor messages upon a request from the master device.

In IOM-2 applications, however, (active handshake protocol), it is also possible that a slave device requests a data transfer e.g. when an IEC-Q device has received an EOC message over the U interface.

For these applications the ELIC has implemented a search mechanism that looks for active handshake bits. When such a monitor channel is found, the µP is interrupted (ISTA_E:MAC) and the address of the involved MF channel is stored in a register (**MFAIR**). The MF handler can then be pointed to that channel by copying the contents of MFAIR to MFSAR and the actual message transfer can take place.

### 5.5.3.1 Registers used in Conjunction with the MF Handler

In detail, the following registers are involved when performing MF channel transfers:

**Operation Mode Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| OMDR: | OMS1 | OMS0 | PSB | PTL | COS | **MFPS** | CSB | RBS |

MFPS: MF channel Protocol Selection;
MFPS = 0: Handshake facility disabled; to be used for SLD and IOM-1 applications.
MFPS = 1: Handshake facility enabled; to be used for IOM-2 applications.

**Monitor/Feature Control Channel FIFO** read/write reset value: empty

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| MFFIFO: | MFD7 | MFD6 | MFD5 | MFD4 | MFD3 | MFD2 | MFD1 | MFD0 |

The 16 byte bidirectional MFFIFO provides intermediate storage for data bytes to be transmitted or received over the monitor or feature control channel.

*Note: The data transfer over an MF channel is half-duplex i.e. if a 'transmit + receive' command is issued, the transmit section of the transfer must first be completed before the receive section starts.*

MFD7 … 0: MF Data bits 7 … 0; MFD7 (MSB) is the first bit to be sent over the serial CFI, MFD0 (LSB) the last.

**MF Channel Subscriber Address Register** write reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| MFSAR: | MFTC1 | MFTC0 | SAD5 | SAD4 | SAD3 | SAD2 | SAD1 | SAD0 |

The exchange of monitor data normally takes place with only one subscriber circuit at a time. This register serves to point the MF handler to that particular CFI timeslot.

MFTC1 … 0: MF Channel Transfer Control 1 … 0; these bits, in addition to CMDR:MFT1,0 and OMDR:MFPS control the MF channel transfer as indicated in **table 49**.

SAD5 … 0: Subscriber address 5 … 0; these bits define the addressed subscriber. The CFI timeslot encoding is similar to the one used for Control Memory accesses using the MAAR register (see **figure 84**).

CFI timeslot encoding of MFSAR derived from MAAR:

| MAAR: | MA7 | **MA6** | **MA5** | **MA4** | **MA3** | **MA2** | **MA1** | MA0 |
|---|---|---|---|---|---|---|---|---|
| | | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | |
| MFSAR: | MFTC1 | MFTC0 | **SAD5** | **SAD4** | **SAD3** | **SAD2** | **SAD1** | **SAD0** |

MAAR:MA7 selects between upstream and downstream CM blocks. This information is not required since the transfer direction is defined by CMDR (transmit or receive).

MAAR:MA0 selects between even and odd timeslots. This information is also not required since MF channels are always located on even timeslots.

**Example**

In CFI mode 0, IOM channel 5 (timeslot 16 … 19) of port 2 shall be addressed for a transmit monitor transfer:

MFSAR = 0010 $0110_B$; the monitor channel occupies timeslot 18 ($10010_B$) of port 2 ($10_B$)

**MF Channel Active Indication Register** read reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| MFAIR: | 0 | SO | SAD5 | SAD4 | SAD3 | SAD2 | SAD1 | SAD0 |

This register is only used in IOM-2 applications (active handshake protocol) in order to identify active monitor channels when the ‘Search for active monitor channels’ command (CMDR:MFSO) has been executed.

SO: MF Channel Search On; this bit indicates whether the ELIC is still busy looking for an active channel (1) or not (0).

SAD5 … 0: Subscriber Address 5 … 0; after an ISTA:MAC interrupt these bits point to the port and timeslot where an active channel has been found. The coding is identical to MFSAR:SAD5 … SAD0. The contents of MFAIR can directly be copied to MFSAR in order to point the MF handler to the channel which requests a monitor receive operation.

**Command Register EPIC®** read reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| CMDR_E | 0 | ST | TIG | CFR | **MFT1** | **MFT0** | **MFSO** | **MFFR** |

Writing to CMDR starts the respective monitor channel operation.

MFT1 … 0: MF Channel Transfer Control Bits 1, 0; these bits start the monitor transfer enabling the contents of the MFFIFO to be exchanged with the subscriber circuits as specified in MFSAR. The function of some commands depends furthermore on the selected protocol (OMDR:MFPS). **Table 49** summarizes all available MF commands.

MFSO: MF Channel Search On; if set to 1, the ELIC starts to search for active MF channels. Active channels are characterized by an active MX bit (logical 0) sent by the remote transmitter. If such a channel is found, the corresponding address is stored in MFAIR and an ISTA_E:MAC

interrupt is generated. The search is stopped when an active MF channel has been found or when OMDR:OMS0 is set to 0.

MFFR: MFFIFO Reset; setting this bit resets the MFFIFO and all operations associated with the MF handler (except for the search function) within 2 RCL periods. The MFFIFO is set into the state 'MFFIFO empty, write access enabled' and any monitor data transfer currently in process will be aborted. MFFR should be set when all data bytes have been read from the MFFIFO after a monitor receive operation.

**Table 49**
**Monitor/Feature Control Channel Commands**

| Transfer Mode | CMDR: MFT, MFT0 | MFSAR | Protocol Selection | Application |
|---|---|---|---|---|
| Inactive | 00 | XXXXXXXX | HS, no HS[1)] | idle state |
| Transmit | 01 | 00 SAD5 … 0 | HS, no HS[1)] | IOM-2, IOM-1, SLD |
| Transmit Broadcast | 01 | 01XXXXXX | HS, no HS[1)] | IOM-2, IOM-1, SLD |
| Test Operation | 01 | 10 - - - - - - | HS, no HS[1)] | IOM-2, IOM-1, SLD |
| Transmit Continuous | 11 | 00 SAD5 … 0 | HS[2)] | IOM-2 |
| Transmit + Receive Same Timeslot | | | | |
| Any # of Bytes | 10 | 00 SAD5 … 0 | HS[2)] | IOM-2 |
| 1 byte expected | 10 | 00 SAD5 … 0 | no HS[1)] | IOM-1 |
| 2 bytes expected | 10 | 01 SAD5 … 0 | no HS[1)] | (IOM-1) |
| 8 bytes expected | 10 | 10 SAD5 … 0 | no HS[1)] | (IOM-1) |
| 16 bytes expected | 10 | 11 SAD5 … 0 | no HS[1)] | (IOM-1) |
| Transmit + Receive Same Line | | | | |
| 1 byte expected | 11 | 00 SAD5 … 0 | no HS[1)] | SLD |
| 2 bytes expected | 11 | 01 SAD5 … 0 | no HS[1)] | SLD |
| 8 bytes expected | 11 | 10 SAD5 … 0 | no HS[1)] | SLD |
| 16 bytes expected | 11 | 11 SAD5 … 0 | no HS[1)] | SLD |

1) Handshake facility disabled (OMDR:MFPS = 0)

2) Handshake facility enabled (OMDR:MFPS = 1)

**Status Register EPIC®** read reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| STAR_E | MAC | TAC | PSS | **MFTO** | **MFAB** | **MFAE** | **MFRW** | **MFFE** |

The status register STAR_E displays the current state of the MFFIFO and of the monitor transfer operation. It should be interrogated after an ISTA_E:MFFI interrupt and prior to accessing the MFFIFO.

The STAR_E register bits do not generate interrupts and are not modified by reading STAR_E.

MFTO: MF Channel Transfer in Operation; an MF channel transfer is in operation (1) or not (0).

MFAB: MF Channel Transfer Aborted; a logical 1 indicates that the remote receiver aborted a handshaked message transfer.

MFAE: MFFIFO Access Enable; the MFFIFO may be either read or written to (1) or it may not be accessed (0).

MFRW: MFFIFO Read/Write; if MFAE is set to logical 1 the MFFIFO may be read (1) or is ready to be written to (0).

MFFE: MFFIFO Empty; the MFFIFO is empty (1) or not empty (1).

**Interrupt Status Register EPIC®** read reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| ISTA_E | TIN | SFI | **MFFI** | **MAC** | PFI | PIM | SIN | SOV |

The ISTA register should be read after an interrupt in order to determine the interrupt source. In connection with the monitor handler two maskable (MASK_E) interrupt bits are provided by the ELIC:

MFFI: MFFIFO interrupt; if this bit is set to 1, the last MF channel command (issued by CMDR:MFT1, MFT0) has been executed and the ELIC is ready to accept the next command. Additional information can be read from STAR_E:MFTO … MFFE. MFFI is reset by reading ISTA_E.

MAC: Monitor Channel Active Interrupt; this bit set to 1 indicates that the ELIC has found an active monitor channel. A new search can be started by reissuing the CMDR:MFSO command. MAC is reset by reading ISTA_E.

### 5.5.3.2 Description of the MF Channel Commands

#### Transmit Command

The transmit command can be used for sending MF data to a single subscriber circuit when no answer is expected. It is applicable for both handshake and non handshake protocols. The message (up to 16 bytes) can be written to the MFFIFO after interrogation of the STAR_E register. After writing of the MF channel address to MFSAR the transfer can be started using the transmit command (CMDR_E = $04_H$). The contents of the MFFIFO will then be transmitted byte by byte to the subscriber circuit.

If the handshake facility is disabled (IOM-1/SLD), the data is sent at a speed of one byte per frame.

If the handshake facility is enabled (IOM-2), each data byte must be acknowledged by the subscriber circuit before the next one is sent. The transfer speed depends therefore on the reaction time of the subscriber circuit. The ELIC can transmit a message at a maximum speed of one byte per two frames.

In order to avoid blocking the software when a subscriber circuit fails to acknowledge a message, a software time out, which resets the monitor transfer (CMDR_E = $01_H$) should be implemented.

If the remote partner aborts the reception of an arriving message i.e. if the ELIC detects an inactive MR bit during at least two consecutive frames, the transmit operation will be stopped, the ISTA_E:MFFI interrupt will be generated and the STAR_E:MFAB bit will be set to 1. The CMDR_E:MFFR bit should then be set to clear the MFAB bit before the next transfer.

When all data bytes of the MFFIFO have been sent (and eventually acknowledged) the ELIC generates an ISTA_E:MFFI interrupt indicating the end of the transfer. The MF handler may then be pointed to another subscriber address for another monitor transfer.

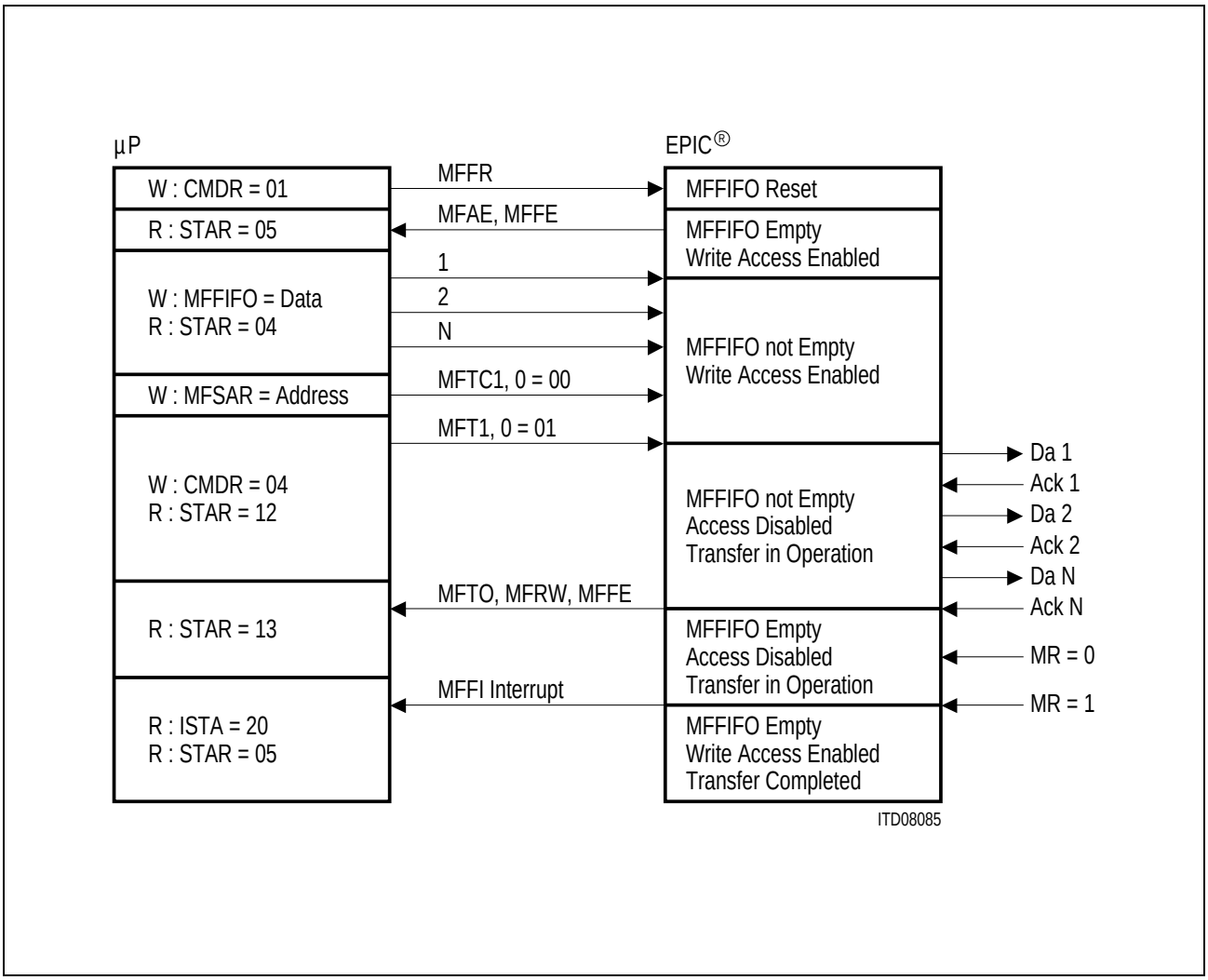



**Figure 105**
**Flow Diagram “Transmit Command”**

### Transmit Continuous Command

The transmit continuous command can be used in IOM-2 applications only (active handshake protocol) to send monitor messages longer than 16 bytes to a single subscriber circuit.

When this command is given, the ELIC transmits the contents of the MFFIFO as with the normal transmit command but does not conclude the transfer by setting MX inactive when the MFFIFO is empty. Instead, the µP is interrupted (ISTA_E:MFFI) and requested to write a new block of data into the MFFIFO. This block may then again be transmitted using the transmit continuous command or, if it is the last block of the long message, it may be transmitted using the normal transmit command (CMDR_E:MFT1, MFT0 = 01). If an answer is expected from the subscriber circuit, the last block may also be terminated using the transmit + receive command (CMDR_E:MFT1, MFT0 = 10). Each message block may be of arbitrary length (1 to 16 bytes).

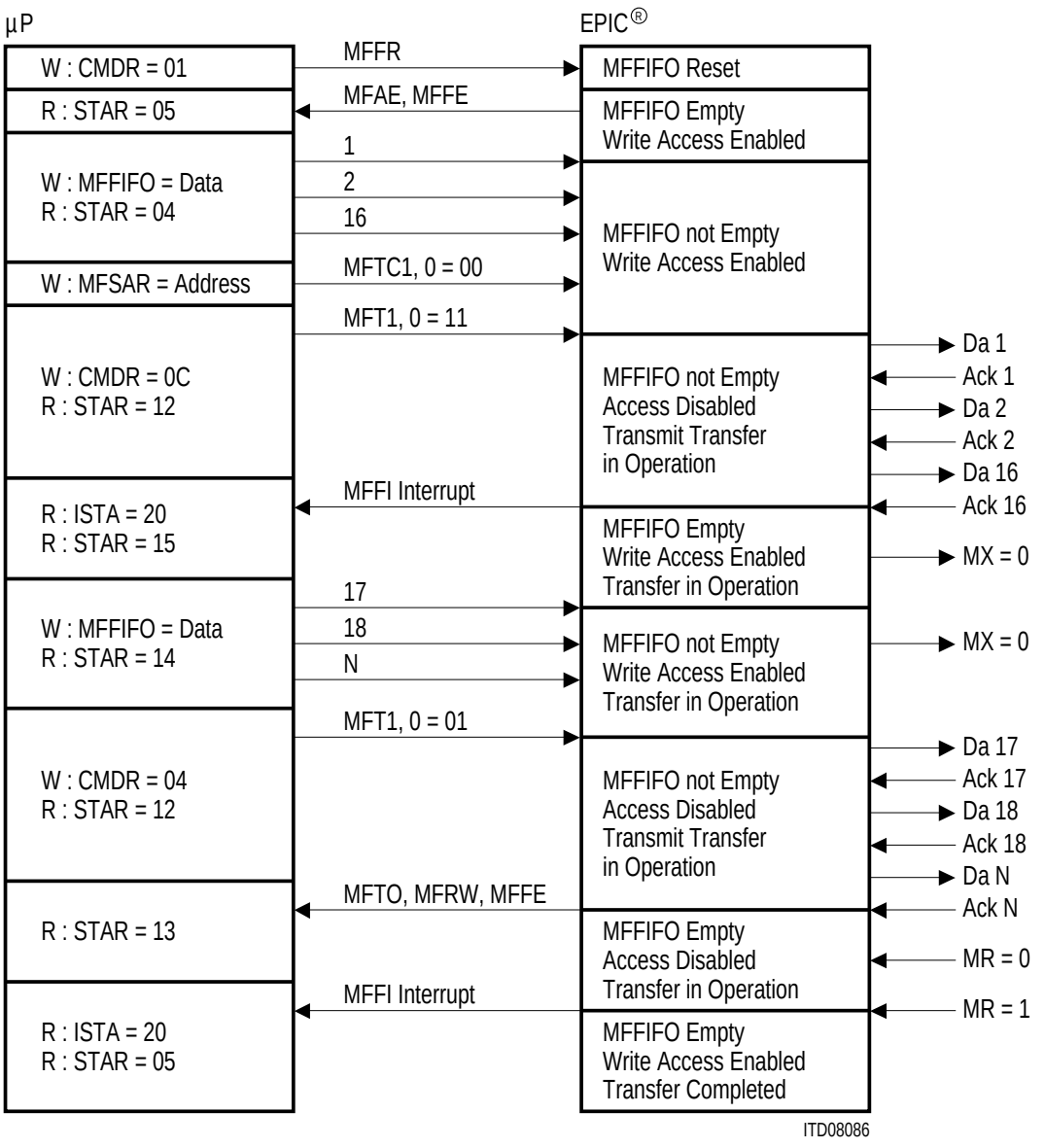



**Figure 106**
**Flow Diagram “Transmit Continuous Command”**

### Transmit + Receive Same Timeslot Command

The transmit + receive same timeslot command can be used to send a message to a subscriber circuit, which will respond with an answer, e.g. reading back the coefficients of a SICOFI device. After first transmitting the contents of the MFFIFO (as with the normal transmit command), the MFFIFO is ready to accept an incoming message which can then be read by the µP when the transfer is completed.

This command can also be used to perform a receive only operation: if a message shall be received without transmission (e.g. after an active monitor channel has been found) the transmit + receive command is issued with an empty MFFIFO.

The command is applicable for both handshake and non-handshake protocols. Since the transfer operation is performed on the same timeslot, its use is intended for IOM applications:

– IOM-2, handshake facility enabled:

The contents of the MFFIFO is sent to the subscriber circuit subject to the IOM-2 protocol i.e each byte must be acknowledged before the next one is sent. When the MFFIFO is empty, the ELIC starts to receive the incoming data bytes, each byte being autonomously acknowledged by the ELIC. Up to 16 bytes may be stored in the MFFIFO. When the end of message is detected (MX bit inactive during two consecutive frames), the transfer is considered terminated and an ISTA_E:MFFI interrupt is generated. The µP can then fetch the message from the MFFIFO. In order to determine the length of the arrived message, the STAR_E:MFFE bit (MFFIFO Empty) should be evaluated before each read access to the MFFIFO. After all bytes have been read, the MFFIFO must be reset with the CMDR_E:MFFR command in order to enable new monitor transfer operations.

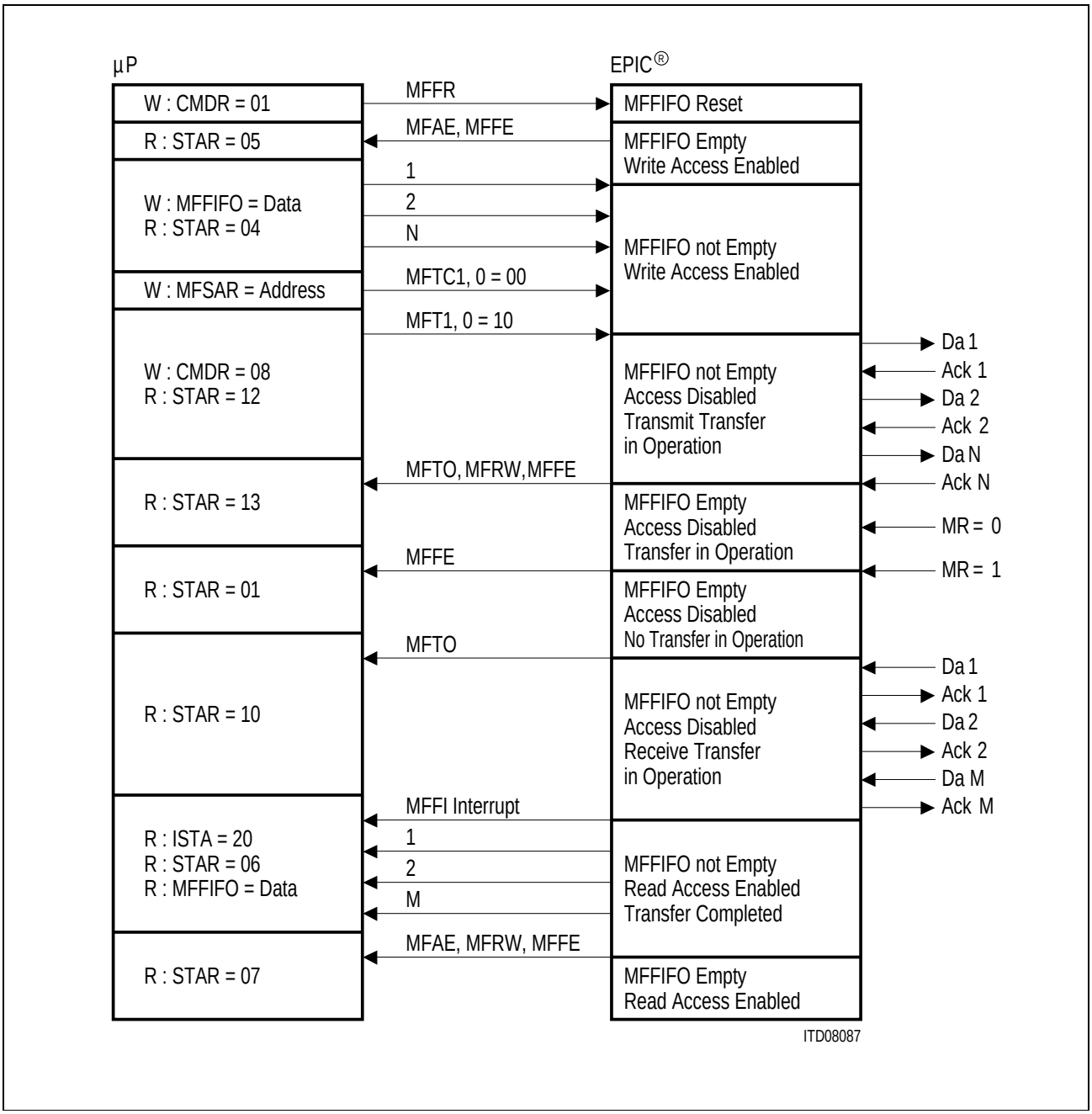



**Figure 107**
**Flow Diagram “Transmit + Receive Same Timeslot Command”**

The reception of monitor messages may also (if required) be aborted at any time simply by setting the CMDR_E:MFFR bit while the receive transfer is still in operation.

If more than 16 bytes shall be received, the following procedure can be adopted:

The first 16 data bytes received will be stored in the MFFIFO and acknowledged to the remote partner. The presence of a 17th byte on the receive line will lead to an ISTA_E:MFFI interrupt. While the transfer is still in operation (STAR_E = $16_H$), with the 17th byte still left unacknowledged, the µP can read the first 16 bytes out of the MFFIFO.

When this is done (STAR_E = $17_H$), the µP issues again (with an empty MFFIFO) the transmit + receive command (CMDR_E = $08_H$) and the ELIC is again ready to receive and acknowledge further monitor bytes.

– IOM-1, handshake facility disabled

The contents of the MFFIFO are sent to the subscriber circuit at a speed of 1 byte per frame. When the last byte has been transmitted, the ELIC stores the received monitor bytes of the next subsequent frames into the MFFIFO. The receive transfer is completed and an ISTA_E:MFFI interrupt is generated after either 1, 2, 8, or 16 frames. The actual number of stored bytes can be selected with MFSAR:MFTC1,MFTC0.

### Transmit + Receive Same Line Command

This command is similar to the Transmit + Receive same timeslot command i.e. it can be used to send a message to a subscriber circuit which will respond with an answer. Its use is, however, intended for SLD applications: CFI mode 3, 8 timeslots/frame, handshake facility disabled.

The transmit operation is performed in the downstream timeslot specified in MFSAR while the receive operation is performed on the same SIP line, but four timeslots later in the upstream timeslot.

### Transmit Broadcast Command

The Transmit Broadcast Command can be used for sending a monitor/feature control message to all subscriber circuits simultaneously. It is applicable for both handshake and non handshake protocols. The procedure is similar to the normal transmit command with the exception that the contents of the MFFIFO is transmitted on all downstream MF timeslots (defined by the CM code field). If the handshake protocols is active (IOM-2) the data bytes are transmitted at a speed of one byte per three frames and the arriving acknowledgments are ignored.

### Test Operation Command

When executing the Test Operation Command, a message written to the MFFIFO will not be transmitted to the subscriber circuit but may instantaneously be read back. All interrupts (ISTA_E) and status (STAR_E) bits will be generated in the same manner as for a normal transmit + receive transfers. It is applicable for both handshake and non-handshake protocols.

### Search For Active Monitor Channels Command

In IOM-2 applications the monitor channel is sometimes used for low speed data transfers over the S and Q channels of an S interface or over the EOC channel of a U (2B1Q) interface. The layer-1 transceivers (SBCX PEB 2081, IEC-Q PEB 2091) may then, upon reception of a new message, start a monitor channel communication with the ELIC.

For those applications where a slave device initiates an MF channel transfer, the ELIC has implemented the “Search For Active Monitor Channels Command”.

The active handshake protocol (OMDR:MFPS = 1) must be selected for this function.

When the “MF Search On” command (CMDR:MFSO = 1) is executed, the ELIC searches for active handshake bits (MX) on all upstream monitor channels. As soon as an active channel is found, an ISTA_E:MAC interrupt is generated, the search is stopped, and the address of this channel is stored in MFAIR. The µP can then copy the value of MFAIR to MFSAR in order to point the MF handler to that particular channel. With an empty MFFIFO the transmit + receive same timeslot command can be executed to initiate the reception of the monitor message. The ELIC will then autonomously acknowledge each received byte and report the end of the transfer by an ISTA_E:MFFI interrupt. The µP can read the message from the MFFIFO and, if required, execute a new MF Search command.

*Note: The search should only be started when no receive transfer is in operation, otherwise each received byte will lead to the ISTA_E:MAC interrupt.*

Once started, the search for active monitor channels can only be stopped when such a channel has been found or when the Control Memory is reset or initialized (OMDR:OMS0 = 0).

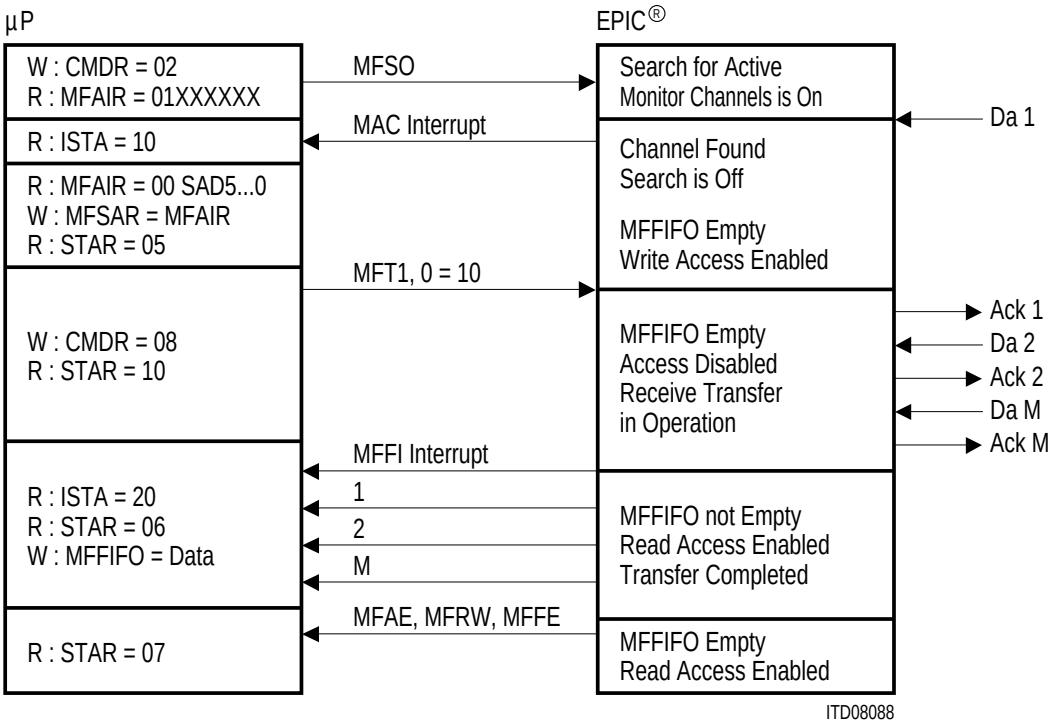



**Figure 108**
**Flow Diagram “Search For Active Monitor Channels Command”**

## 5.6 µP Channels

If a CFI timeslot shall be accessed by the µP instead of being switched to the PCM interface, this channel can be configured as a µP channel. This is achieved by writing the code ‘1001’ to the CM code field. In this case the content of the corresponding CFI timeslot is directly exchanged with the CM data field. **Figure 109** and **figure 110** illustrate the use of the Control Memory (CM) data and code fields for such applications.

If a CFI timeslot is initialized as µP channel, the function taken on by the CM data field can be compared to the function taken on by the Data Memory (DM) data field at the PCM interface, i.e. it buffers the PCM data received or to be transmitted at the serial interface. In contrast to the PCM interface, where PCM idle channels can be programmed on a 2 bit sub-timeslot basis, the CFI only allows µP access for full 8 bit timeslots.

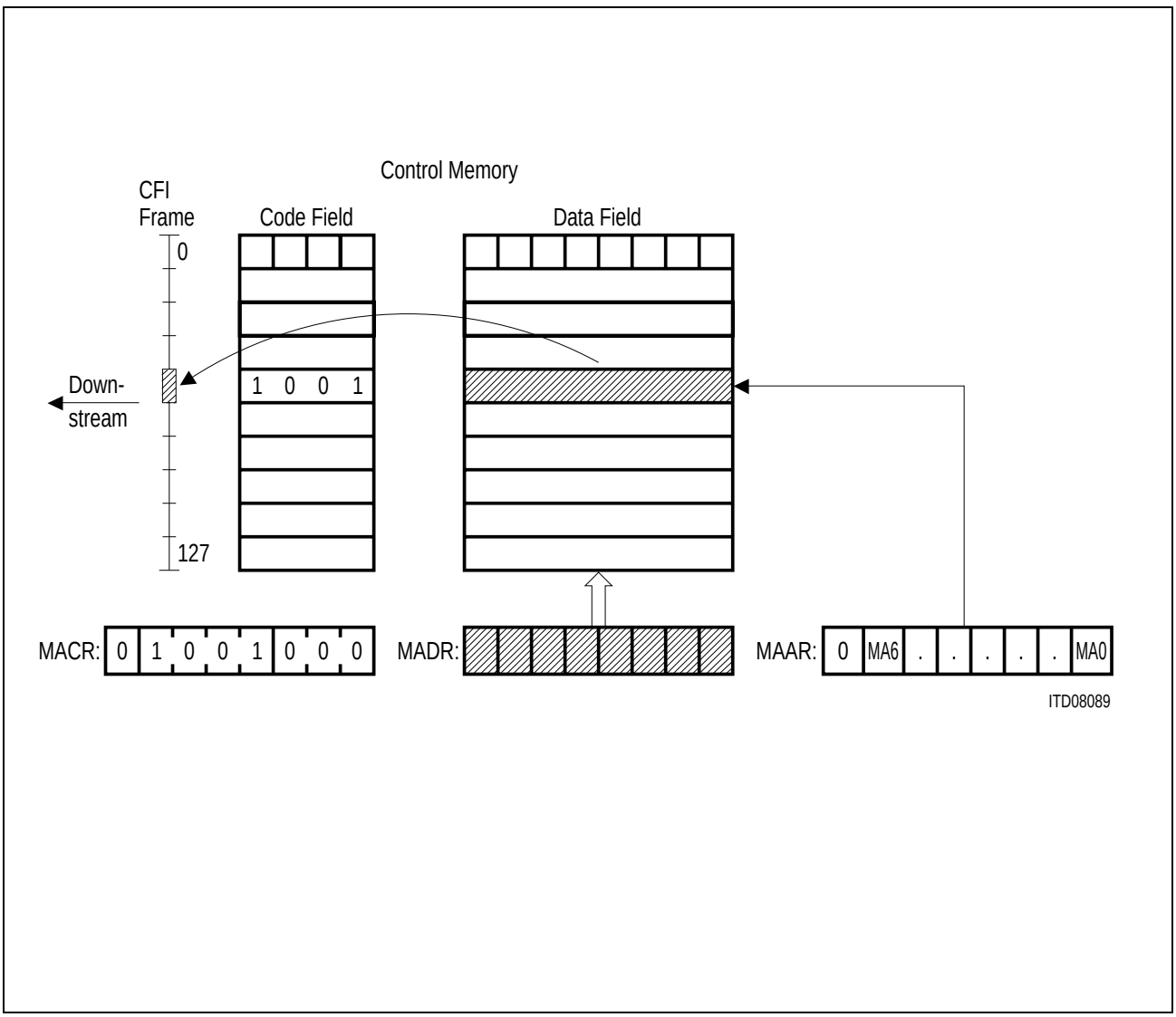



**Figure 109**
**µP Access to the Downstream CFI Frame**

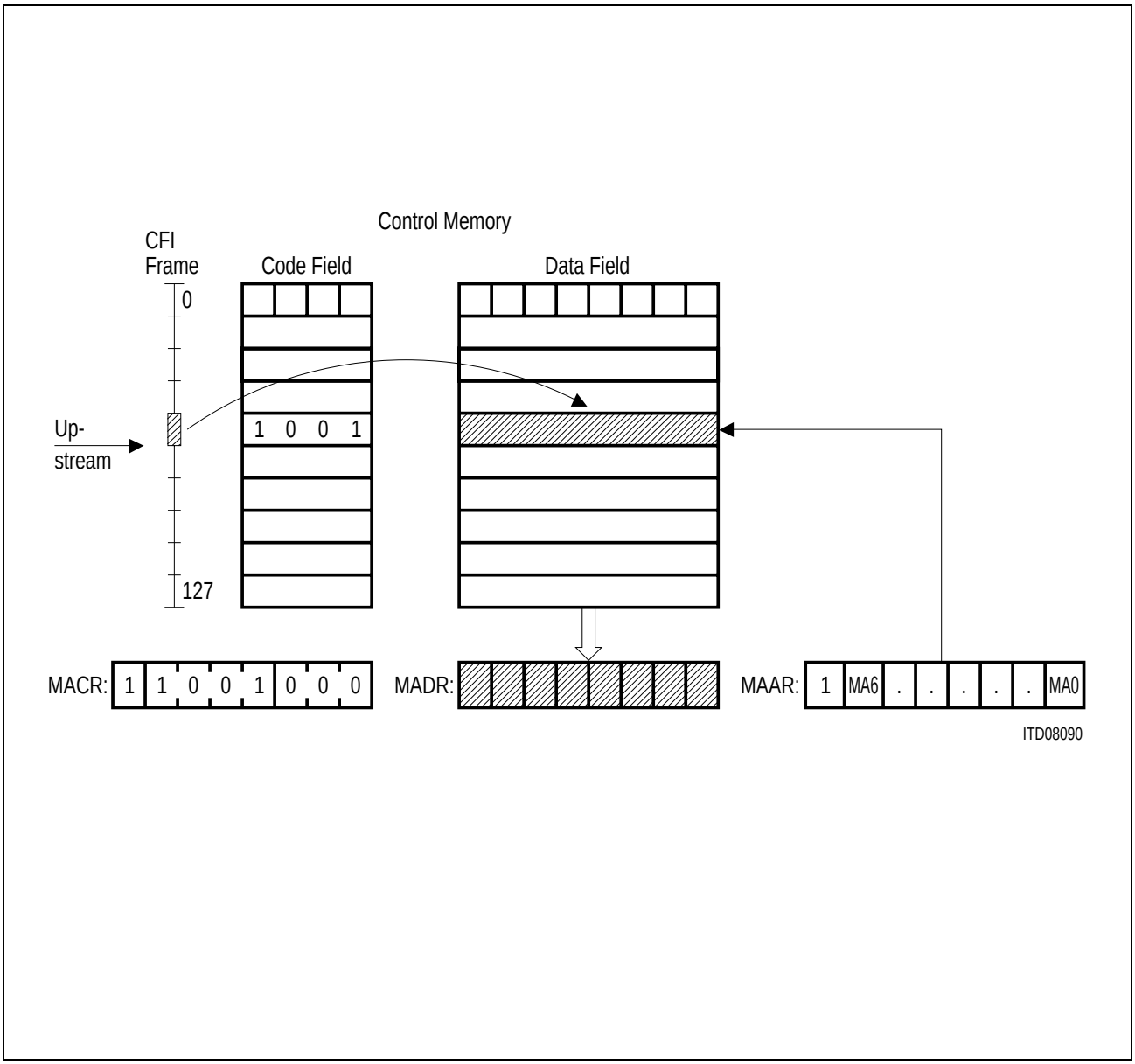



**Figure 110**
**µP Access to the Upstream CFI Frame**

The value written to the downstream CM data field location is transmitted repeatedly in every frame (CFI idle value) during the corresponding downstream CFI timeslot until a new value is loaded or the 'µP channel' function is disabled. There are no interrupts generated.

The upstream CM data field can be read at any time. The CM data field is updated in every frame. The last value read represents the value received. There are no interrupts generated.

For frame-synchronous exchange of data between the µP and the CFI, the synchronous transfer utility must be used (refer to **chapter 5.7**). Since this utility realizes the data

exchange between the STDA (STDB) register and the CM data field, it is also necessary to initialize the corresponding CFI timeslots as µP channels.

The following sequences can be used to program, verify, and cancel a CFI µP channel:

**Writing a Downstream CFI Idle Value**

– in case the CM code field has not yet been initialized with the 'µP channel' code:

W:MADR = CFI idle value to be transmitted
W:MAAR = downstream CFI port and timeslot encoded according to **figure 84**
W:MACR = $0111\ 1001_B = 79_H$ ; CM code '1001' (µP transfer)

– in case the CM code field has already been initialized with the 'µP channel' code:

W:MADR = CFI idle value to be transmitted
W:MAAR = downstream CFI port and timeslot encoded according to **figure 84**
W:MACR = $0100\ 1000_B = 48_H$; MOC code '1001' (CM data field access)

**Reading an Upstream CFI idle Value**

– Initializing an upstream CFI timeslot as a µP channel:

W:MADR = don't care
W:MAAR = upstream CFI port and timeslot encoded according to **figure 84**
W:MACR = $0111\ 1001_B = 79_H$; CM code '1001' (µP transfer)

– Reading the upstream CFI idle value:

W:MAAR = upstream CFI port and timeslot encoded according to **figure 84**
W:MACR = $1100\ 1000_B = C8_H$; MOC code '1001' (CM data field access)
wait for STAR:MAC = 0
R:MADR = received CFI idle value

**Reading Back the Idle Value Transmitted at a Downstream CFI µP Channel:**

W:MAAR = downstream CFI port and timeslot encoded according to **figure 84**
W:MACR = $1100\ 1000_B = C8_H$; MOC code '1001' (CM data field access)
wait for STAR:MAC = 0
R:MADR = transmitted CFI idle value

**Reading Back the CFI Functionality of a given CFI Timeslot:**

W:MAAR = CFI port and timeslot encoded according to **figure 84**
W:MACR = $1111\ 0000_B = F0_H$; MOC code '111X' (CM code field access)
wait for STAR:MAC = 0
R:MADR = $XXXX\ code_B$; if code = 1001, the CFI timeslot is a 'µP channel'

**Cancelling of a Programmed CFI µP Channel:**

W:MADR = don't care
W:MAAR = CFI port and timeslot encoded according to **figure 84**
W:MACR = $0111\ 0000_B = 70_H$; code '0000' (unassigned channel)

**Examples**

In CFI mode 1 the following µP channels shall be realized:

Upstream: CFI port 1, timeslot 7:

W:MADR = $1111\ 1111_B$ ; don't care
W:MAAR = $1000\ 1111_B$ ; CFI timeslot encoding according to **figure 84**
W:MACR = $0111\ 1001_B$ ; CM code for a µP channel (code '1001')

Downstream: CFI port 0, timeslot 2, the value '0000 0111' shall be transmitted:

W:MADR = $0000\ 0111_B$ ; CFI idle value '0000 0111'
W:MAAR = $0000\ 0100_B$ ; CFI timeslot encoding according to **figure 84**
W:MACR = $0111\ 1001_B$ ; CM code for a µP channel (code '1001')

The next sequence will read the currently received value at DU1, TS7:

W:MAAR = $1000\ 1111_B$ ; upstream CFI port and timeslot
W:MACR = $1100\ 1000_B = C8_H$; read back command
wait for STAR:MAC = 0
R:MADR = value ; received CFI idle value

## 5.7 Synchronous Transfer Utility

The synchronous transfer utility allows the synchronous exchange of information between the PCM interface, the configurable interface, and the µP interface for two independent channels (A and B). The µP can thus monitor, insert, or manipulate the data synchronously to the frame repetition rate. The synchronous transfer is controlled by the synchronous transfer registers.

The information is buffered in the synchronous transfer data register STDA (STDB). It is copied to STDA (STDB) from a data or control memory location pointed to by the content of the synchronous receive register SARA (SARB) and copied from the STDA (STDB) to a data or control memory location pointed to by the content of the synchronous transfer transmit register SAXA (SAXB).
The SAXA (SAXB) and SARA (SARB) registers identify the interface (PCM or CFI) as well as the timeslot and port numbers of the involved channels according to **figure 84**. Control bits in the synchronous transfer control register STCR allow restricting the synchronous transfer to one of the possible sub-timeslots and enables or disables the synchronous transfer utility.

For example, it is possible to read information via the downstream data memory from the PCM interface input to the STDA (STDB) register and to transmit it from this register back via the upstream data memory to the PCM interface output, thus establishing a PCM - PCM loop. Similarly the synchronous transfer facility may be used to loop back configurable interface channels or to establish connections between the CFI and PCM interfaces. While the information is stored in the data register STDA (STDB), it may be read and or modified by the µP.

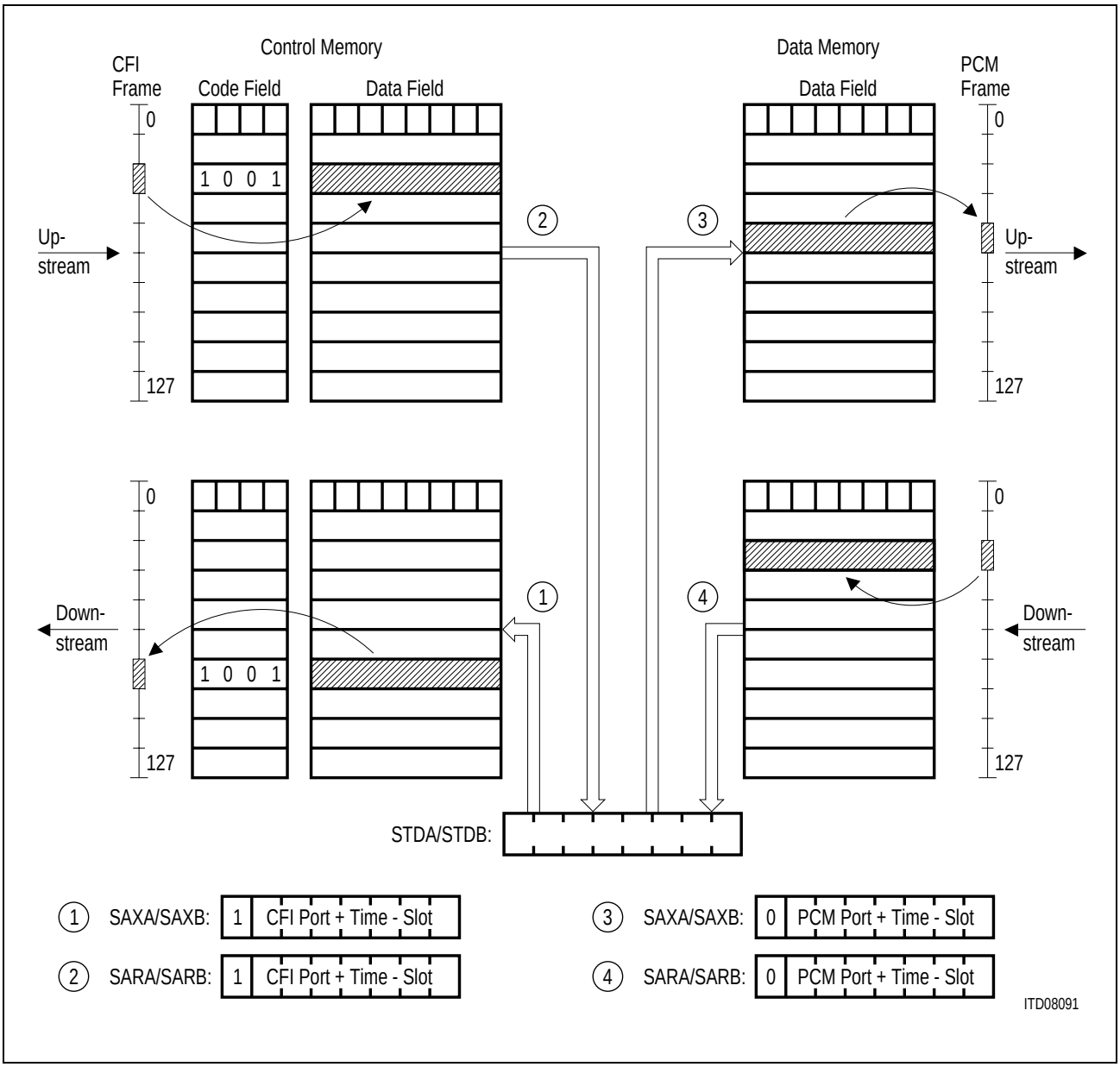



**Figure 111**
**Access to PCM and CFI Data Using the Synchronous Transfer Utility**

In upstream transmit direction (PCM interface output), it is necessary to assure that no other data memory access writes to the same location in the upstream DM block. Hence an upstream connection involving the same PCM port and timeslot as the synchronous transfer may not be programmed.

An idle code previously written to the data or control memory for the upstream or downstream directions is overwritten.

At the PCM interface it is possible to restrict the synchronous exchange with the data registers STDA (STDB) to a 2 or 4 bit sub-timeslot position. The working principle is similar to the subchannel switching described in **chapter 5.4.2**.

If the CFI is selected as source/destination of the synchronous transfer, the contents of the data register STDA (STDB) are exchanged with the control memory data field. It is therefore necessary to initialize the corresponding control memory code field as ‘µP channel’ (code ‘1001’). Also refer to **chapter 5.6**.

Since the µP channel set-up at the CFI only allows a channel bandwidth of 64 kBit/s, the synchronous transfer utility also allows only 64 kBit/s channels at the CFI.

The ELIC generates interrupts guiding through the synchronous transfer. Upon the ISTA_E:SIN interrupt the data registers STDA (STDB) may be accessed for some time. If the data register of an active channel has not been accessed at the end of this time interval the ISTA:SOV interrupt is generated, before the ELIC performs the transfer to the selected memory locations. If the µP fails to overwrite the data register with a new value, the value previously received from the timeslot pointed to by SARA (SARB) will be transmitted. The ISTA_E:SIN and SOV interrupts are generated periodically at fixed time points within the frame regardless of the actual positions of the involved timeslots. The repetition cycle of the synchronous transfer is identical to a frame length (125 µs). The access window is closed for at most, 16 RCL periods per active channel + 1 RCL period, leaving a very long access time.

This behavior is also shown in **figure 112**:

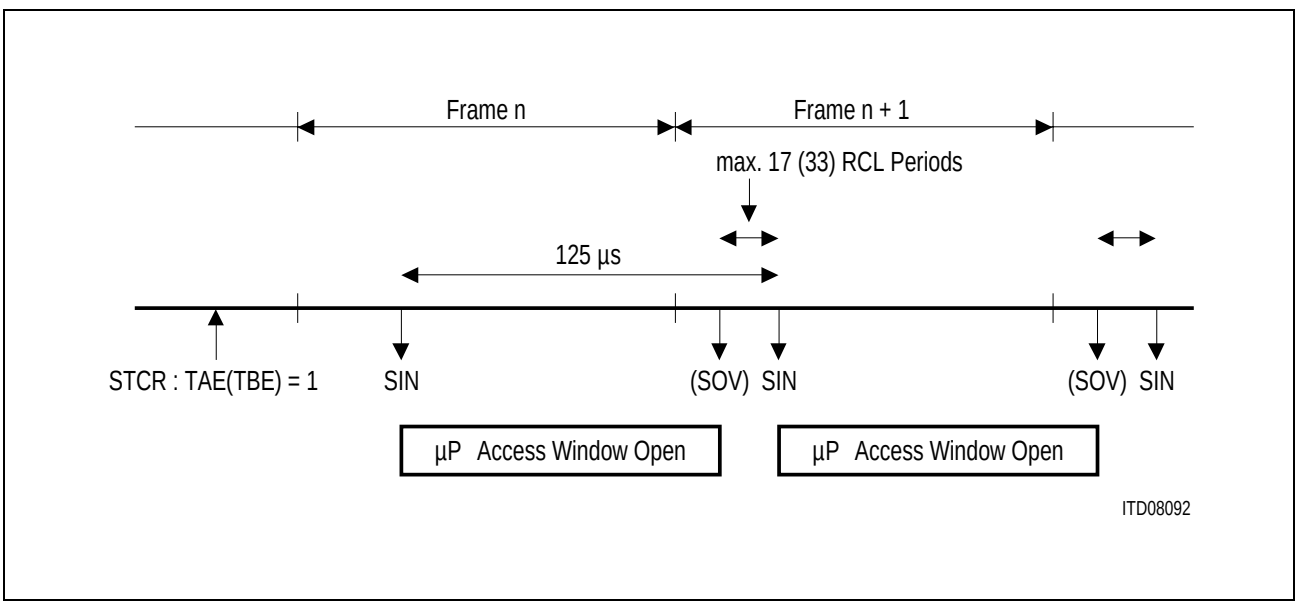



**Figure 112**
**Synchronous Transfer Flow Diagram**

### Example

In a typical IOM-2 application, the RCL frequency is 4096 kHz, i.e. an RCL period lasts 244 ns. The IOM-2 frame duration is 125 µs. If one synchronous channel is enabled, the access window is open for 121 µs and closed for 4 µs. If both synchronous channels are enabled, the access window is open for 117 µs and closed for 8 µs.

### 5.7.1 Registers Used in Conjunction with the Synchronous Transfer Utility

**Synchronous Transfer Data Register A** read/write reset value: undefined

| | bit 7 | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|
| STDA: | MTDA7 | MTDA6 | MTDA4 | MTDA3 | MTDA2 | MTDA1 | MTDA0 |

The STDA register buffers the data transferred over the synchronous transfer channel A.

MTDA7 to MTDA0 hold the bits 7 to 0 of the respective timeslot. MTDA7 (MSB) is the bit transmitted/received first, and MTDA0 (LSB) the bit transmitted/received last over the serial interface.

**Synchronous Transfer Receive Address Register B** read/write reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| STDB: | MTDB7 | MTDB6 | MTDB5 | MTDB4 | MTDB3 | MTDB2 | MTDB1 | MTDB0 |

The STDB register buffers the data transferred over the synchronous transfer channel B.

MTDB7 to MTDB0 hold the bits 7 to 0 of the respective timeslot. MTDB7 (MSB) is the bit transmitted/received first, MTDB0 (LSB) the bit transmitted/received last over the serial interface.

**Synchronous Transfer Receive Address Register A** read/write reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| SARA: | ISRA | MTRA6 | MTRA5 | MTRA4 | MTRA3 | MTRA2 | MTRA1 | MTRA0 |

The SARA register specifies for synchronous transfer channel A from which input interface, port, and timeslot the serial data is extracted. This data can then be read from the STDA register.

ISRA: Interface Select Receive for channel A; selects the PCM interface (ISRA = 0) or the CFI (ISRA = 1) as the input interface for synchronous channel A.

MTRA6 … 0: µP Transfer Receive Address for channel A; selects the port and timeslot number at the interface selected by ISRA according to **figure 84**: MTRA6 … 0 = MA6 … 0.

**Synchronous Transfer Receive Address Register B** read/write reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| SARB: | ISRB | MTRB6 | MTRB5 | MTRB4 | MTRB3 | MTRB2 | MTRB1 | MTRB0 |

The SARB register specifies for synchronous transfer channel B from which input interface, port, and timeslot the serial data is extracted. This data can then be read from the STDB register.

ISRB: Interface Select Receive for channel B; selects the PCM interface (ISRB = 0) or the CFI (ISRB = 1) as the input interface for synchronous channel B.

MTRB6 … 0: µP Transfer Receive Address for channel B; selects the port and timeslot number at the interface selected by ISRB according to **figure 84**: MTRB6 … 0 = MA6 … 0.

**Synchronous Transfer Receive Address Register A** read/write reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| SAXA: | ISXA | MTXA6 | MTXA5 | MTXA4 | MTXA3 | MTXA2 | MTXA1 | MTXA0 |

The SAXA register specifies for synchronous transfer channel A to which output interface, port, and timeslot the serial data contained in the STDA register is sent.

ISXA: Interface Select Transmit for channel A; selects the PCM interface (ISXA = 0) or the CFI (ISXA = 1) as the output interface for synchronous channel A.

MTXA6 … 0: µP Transfer Transmit Address for channel A; selects the port and timeslot number at the interface selected by ISXA according to **figure 84**: MTXA6 … 0 = MA6 … 0.

**Synchronous Transfer Transmit Address Register B** read/write reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| SAXB: | ISXB | MTXB6 | MTXB5 | MTXB4 | MTXB3 | MTXB2 | MTXB1 | MTXB0 |

The SAXB register specifies for synchronous transfer channel B to which output interface, port, and timeslot the serial data contained in the STDB register is sent.

ISXB: Interface Select Transmit for channel B; selects the PCM interface (ISXB = 0) or the CFI (ISXB = 1) as the output interface for synchronous channel B.

MTXB6 … 0: µP Transfer Transmit Address for channel B; selects the port and timeslot number at the interface selected by ISXB according to **figure 84**: MTXB6 … 0 = MA6 … 0.

**Synchronous Transfer Control Register STCR** read/write reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| STCR: | TBE | TAE | CTB2 | CTB1 | CTB0 | CTA2 | CTA1 | CTA0 |

The STCR register bits are used to enable or disable the synchronous transfer utility and to determine the sub-timeslot bandwidth and position if a PCM interface timeslot is involved.

TAE, TBE: Transfer Channel A (B) Enable; A logical 1 enables the µP transfer, a logical 0 disables the transfer of the corresponding channel.

CTA2 … 0:
CTB2 … 0: Channel Type A (B); these bits determine the bandwidth of the channel and the position of the relevant bits in the timeslot according to **tabel 50**. Note that if a CFI timeslot is selected as receive or transmit timeslot of the synchronous transfer, the 64 kBit/s bandwidth must be selected (CT#2 … CT#0 = 001).

**Table 50**
**Synchronous Transfer Channel Type**

| CT#2 | CT#1 | CT#0 | Bandwidth | Transferred Bits |
|---|---|---|---|---|
| 0 | 0 | 0 | not allowed | – |
| 0 | 0 | 1 | 64 kBit/s | bits 7 … 0 |
| 0 | 1 | 0 | 32 kBit/s | bits 3 … 0 |
| 0 | 1 | 1 | 32 kBit/s | bits 7 … 4 |
| 1 | 0 | 0 | 16 kBit/s | bits 1 … 0 |
| 1 | 0 | 1 | 16 kBit/s | bits 3 … 2 |
| 1 | 1 | 0 | 16 kBit/s | bits 5 … 4 |
| 1 | 1 | 1 | 16 kBit/s | bits 7 … 6 |

**Interrupt Status Register EPIC®** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| ISTA_E: | TIN | SFI | MFFI | MAC | PFI | PIM | **SIN** | **SOV** |

The ISTA register should be read after an interrupt in order to determine the interrupt source. Two maskable (MASK_E) interrupts are provided in connection with the synchronous transfer utility:

SIN: Synchronous Transfer Interrupt; The SIN interrupt is enabled if at least one synchronous transfer channel (A and/or B) is enabled via the STCR:TAE, TBE bits. The SIN interrupt is generated when the access window for the µP opens. After the occurrence of the SIN interrupt (logical 1) the µP can read and/or write the synchronous transfer data registers (STDA, STDB). The window where the µP can access the data registers is open for the duration of one frame (125 µs) minus 17 RCL cycles if only one synchronous channel is enabled and it is open for one frame minus 33 RCL cycles if both A and B channels are enabled. The SIN bit is reset by reading ISTA_E.

SOV: Synchronous Transfer Overflow; The SOV interrupt is generated (logical 1) if the µP fails to access the data registers (STDA, STDB) within the access window. The SOV bit is reset by reading ISTA_E.

**Examples**

1) In PCM mode 0, the synchronous transfer utility (channel A) shall be used to loop bits 7 … 6 of downstream PCM port 1, timeslot 5 back to bits 7 … 6 of upstream PCM port 2, timeslot 9. Since no µP access to the data is required the ISTA_E:SIN and SOV bits are both masked:

   W:MASK = $03_H$ ; SIN = SOV = 1
   W:SARA = $13_H$ ; ISRA = 0, port 1, TS5
   W:SAXA = $25_H$ ; ISXA = 0, port 2, TS9
   W:STCR = $47_H$ ; TAE = 1, CTA2 … 0 = 111 (bits 7 … 6)

2) In PCM mode 0 and CFI mode 0, the µP shall have access to both the downstream and upstream CFI port 0, timeslot 1 via the synchronous transfer channel B:

   W:SARB = $81_H$ ; ISRB = 1, port 0, TS1
   W:SAXB = $81_H$ ; ISXB = 1, port 0, TS1
   W:STCR = $88_H$ ; TBE = 1, CTA2 … 0 = 001 (bits 7 … 0)

   Wait for interrupt:

   R:ISTA = $02_H$ ; SIN = 1
   R:SADB = upstream CFI data
   W:SADB = downstream CFI data

Wait for next SIN interrupt and transfer further data bytes … .

## 5.8 Supervision Functions

### 5.8.1 Hardware Timer

**Hardware Timer**

The ELIC provides a programmable hardware timer which can be used for three purposes:

– General purpose timer for continuously interrupting the µP at programmable time intervals.
– Timer to define the last look period for signaling channels at the CFI (see **chapter 5.5.1**).
– Timer to define the FSC multiframe generation at the CFI (CMD2:FC2 … 0 = 111, see **chapter 5.2.2.3**).

Normally in a system only one of these functions is required and therefore active at a time. However, it is also possible to have any combination of these functions active, if it is acceptable that all three applications use the same timer value.

The timer period can be selected from 250 µs up to 32 ms in increments of 250 µs.

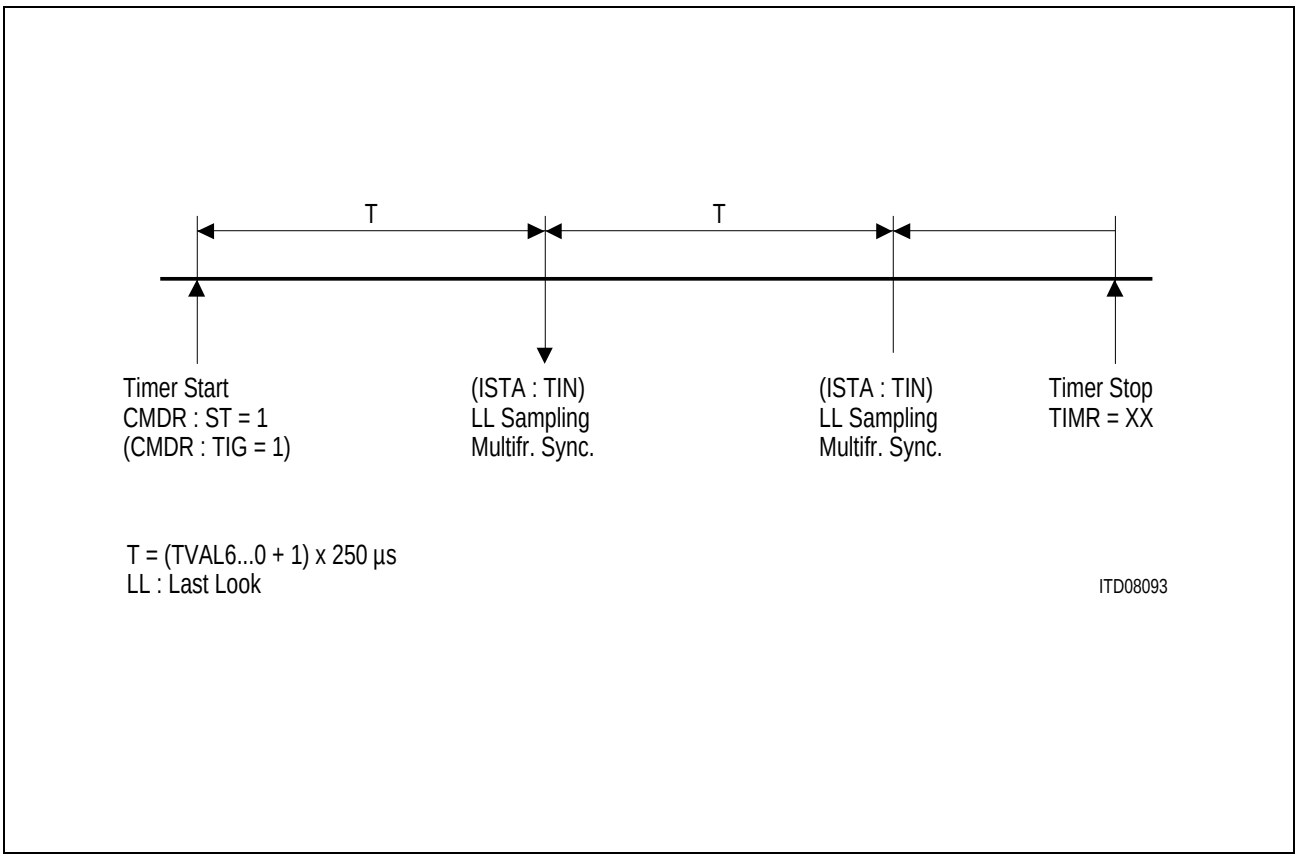



**Figure 113**
**Timer Applications**

The following register bits are used in conjunction with the hardware timer:

**Timer Register** write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| TIMR: | SSR | TVAL6 | TVAL5 | TVAL4 | TVAL3 | TVAL2 | TVAL1 | TVAL0 |

Writing to the TIMR register stops the timer operation!

SSR: Signaling Channel Sample Rate; this bit actually does not affect the timer operation. It is used to select between a fixed last look period for signaling channels of 125 µs (SSR = 1), which is independent of the timer operation and a signaling sample rate that is defined by the timer period (SSR = 0).

TVAL6 … 0: Timer Value; The timer period is programmed here in increments of 250 µs:
Timer period = (TVAL6 … 0 + 1) × 250 µs

**Command Register EPIC®** write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| CMDR_E | 0 | **ST** | **TIG** | CFR | MFT1 | MFT0 | MFSO | MFR |

ST: Start Timer; setting this bit to logical 1 starts the timer to run cyclically from 0 to the value programmed in TIMR:TVAL6 … 0. Setting this bit to logical 0 does not affect the timer operation. If the timer shall be stopped, the TIMR register must simply be written with a random value.

TIG: Timer Interrupt Generation; setting this bit together with CMDR_E:ST to logical 1 causes the ELIC to generate a periodic interrupt (ISTA_E:TIN) each time the timer expires. Setting the TIG bit to logical 0 together with the CMDR:ST bit set to logical 1 disables the interrupt generation. It should be noted that this bit only controls the ISTA_E:TIN interrupt generation and need not be set for the ISTA_E:SFI interrupt generation.

**Interrupt Status Register EPIC®** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| ISTA_E: | **TIN** | SFI | MFFI | MAC | PFI | PIM | SIN | SOV |

The ISTA register should be read after an interrupt in order to determine the interrupt source. In connection with the hardware timer one maskable (MASK_E) interrupt bit is provided by the ELIC:

TIN: Timer Interrupt; if this bit is set to logical 1, a timer interrupt previously requested with CMDR_E:ST,TIG = 1 has occurred. The TIN bit is reset by reading ISTA_E. It should be noted that the interrupt generation is periodic, i.e. unless stopped by writing to TIMR, the ISTA_E:TIN will be generated each time the timer expires.

**Status Register EPIC®** read reset value: $05_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| STAR_E: | MAC | **TAC** | PSS | MFTO | MFAB | MFAE | MFRW | MFFE |

The STAR_E register bits do not generate interrupts and are not modified by reading STAR_E.

TAC: Timer Active; While the timer is running (CMDR:ST=1) the TAC bit is set to logical 1. The TAC bit is reset to logical 0 after the timer has been stopped (W:TIMR = XX).

### 5.8.2 PCM Input Comparison

To simplify the realization of redundant PCM transmission lines, the ELIC can be programmed to compare the contents of certain pairs of its PCM input lines. If a pair of lines carry the same information (normal case), nothing happens. If however the two lines differ in at least one bit (error case), the ELIC generates an ISTA_E:PIM interrupt and indicates in the PICM register the pair of input lines and the timeslot number that caused that mismatch.

The comparison function is carried out between the pairs of physical PCM input lines RxD0/RxD1 and RxD2/RxD3. It can be activated in all PCM modes, including PCM mode 0. However, a redundant PCM input line that can be switched over to by means of the PMOD:AIS1 … 0 bits is of course only available in PCM modes 1 and 2.

The following register bits are used in conjunction with the PCM input comparison function:

**PCM Mode Register** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| PMOD: | PMD1 | PMD0 | PCR | PSM | AIS1 | AIS0 | **AIC1** | **AIC0** |

AIC1 … 0: Alternative Input Comparison 1 and 0.
AIC0 set to logical 1 enables the comparison function between RxD0 and RxD1.
AIC1 set to logical 1 enables the comparison function between RxD2 and RxD3.
AIC1, AIC0 set to logical 0 disables the respective comparison function.
In PCM mode 2, AIC0 must be set to logical 0.

**Interrupt Status Register EPIC®** read reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| ISTA_E: | TIN | SFI | MFFI | MAC | PFI | **PIM** | SIN | SOV |

The ISTA_E register should be read after an interrupt in order to determine the interrupt source. In connection with the PCM comparison function one maskable (MASK_E) interrupt bit is provided by the ELIC:

PIM: PCM Input Mismatch; this bit is set to logical 1 immediately after the comparison logic has detected a mismatch between a pair of PCM input lines. The exact reason for the interrupt can be determined by reading the PICM register. Reading ISTA_E clears the PIM bit. A new PIM interrupt can only be generated after the PICM register has been read.

**PCM Input Comparison Mismatch** read reset value: undefined

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| PICM: | IPN | TSN6 | TSN5 | TSN4 | TSN3 | TSN2 | TSN1 | TSN0 |

The contents of the PICM register is only valid after an ISTA_E:PIM interrupt!

The PICM register must be read after an ISTA_E:PIM interrupt in order to enable a new PIM interrupt generation.

IPN: Input Pair Number; this bit indicates the pair of input lines where a mismatch occurred. A logical 0 indicates a mismatch between lines RxD0 and RxD1, a logical 1 between lines RxD2 and RxD3.

TSN6 … 0: Timeslot Number 6 … 0; these bits specify the timeslot number and the bit positions that generated the ISTA_E:PIM interrupt according to the table below. TPF denotes the number of timeslots per PCM frame

**Table 51**
**Identification of the Timeslot and Bit Number in Case of a Mismatch**

| PCM Mode | Timeslot Identification | Bit Identification |
|---|---|---|
| 2 | $[TSN6 \ldots 0 + 8]_{mod\ TPF}$ | |
| 1, 3 | $[TSN6 \ldots 1 + 4]_{mod\ TPF}$ | TSN0 = 1 : bits 3 … 0<br>TSN0 = 0 : bits 7 … 4 |
| 0 | $[TSN6 \ldots 2 + 2]_{mod\ TPF}$ | TSN1 … 0 = 11 : bits 1 … 0<br>TSN1 … 0 = 10 : bits 3 … 2<br>TSN1 … 0 = 01 : bits 5 … 4<br>TSN1 … 0 = 00 : bits 7 … 6 |

**Example**

In PCM mode 1, the logical PCM port 0 is connected to two physical PCM transmission links. The comparison function for RxD0/RxD1 is enabled via PMOD:AIC0 = 1. Suddenly a bit error occurs at one of the receive lines in timeslot 13, bit 2. The µP would then get the following information from the ELIC:

Interrupt!

R: ISTA = $04_H$ ; PIM interrupt
R: PICM = $13_H$ ; IPN = 0, TSN6 … 1 = 9, TSN0 = 1

In order to determine the line actually at fault (RxD0 or RxD1) the system must send a known pattern in one of the timeslots and compare the actually received value with that known pattern.

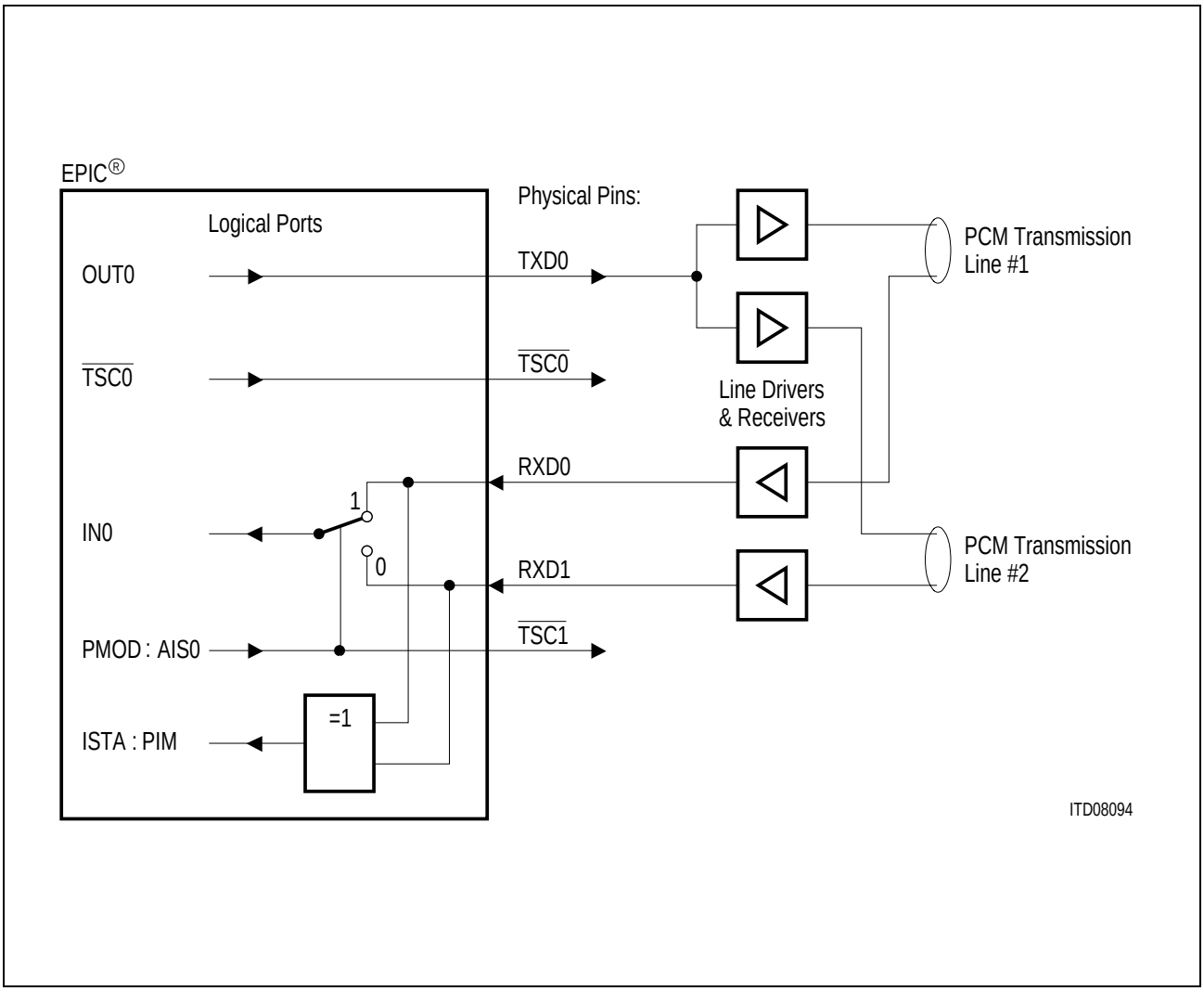



**Figure 114**
**Connection of Redundant PCM Transmission Lines to the ELIC®**

### 5.8.3 PCM Framing Supervision

Usually the repetition rate of the applied framing pulse PFS is identical to the frame period (125 µs). If this is the case, the 'loss of synchronism indication function' can be used to supervise the clock and framing signals for missing or additional clock cycles. The ELIC internally checks the PFS period against the duration expected from the programmed clock rate. The clock rate corresponds to the frequency applied to the PDC pin. The number of clock cycles received within one PFS period is compared with the values programmed to PBNR (number of bits per frame) and PMOD:PCR (single/double clock rate operation). If for example single clock rate operation with 24 timeslots per frame is programmed, the ELIC expects 192 clock cycles within one PFS period. The synchronous state is reached after the ELIC has detected two consecutive correct frames. The synchronous state is lost if one erroneous frame is found. The

synchronization status (gained or lost) can be read from the STAR_E register (PSS bit) and each status change generates an interrupt (ISTA_E:PFI).

It should be noted that the framing supervision function is optional, i.e. it is also allowed to apply a PFS signal having a period of several frame periods e.g. 4 kHz, 2 kHz, … . The STAR_E:PSS bit will then be at logical 0 all the time, which does however not affect the proper operation of the ELIC.

The following register bits are used in conjunction with the PCM framing supervision:

**Interrupt Status Register EPIC®** read/write reset value: $00_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| ISTA_E: | TIN | SFI | MFFI | MAC | **PFI** | PIM | SIN | SOV |

The ISTA_E register should be read after an interrupt in order to determine the interrupt source. In connection with the PCM framing control one maskable (MASK_E) interrupt bit is provided by the ELIC:

PFI: PCM Framing Interrupt; if this bit is set to logical 1, the STAR_E:PSS bit has changed its polarity. To determine whether the PCM interface is synchronized or not, STAR_E must be read. The PFI bit is reset by reading ISTA_E.

**Status Register EPIC®** read reset value: $05_H$

| | bit 7 | | | | | | | bit 0 |
|---|---|---|---|---|---|---|---|---|
| STAR_E: | MAC | TAC | **PSS** | MFTO | MFAB | MFAE | MFRW | MFFE |

The STAR_E register bits do not generate interrupts and are not modified by reading STAR_E. However, each change of the PSS bit ($0 \rightarrow 1$ and $1 \rightarrow 0$) causes an ISTA_E:PFI interrupt.

PSS: PCM Synchronization Status; while the PCM interface is synchronized, the PSS bit is set to logical 1. The PSS bit is reset to logical 0 if there is a mismatch between the PBNR value and the applied clock and framing signals (PDC/PFS) or if OMDR:OMS0 = 0.

### 5.8.4 Power and Clock Supply Supervision/Chip Version

#### Power and Clock Supply Supervision

The + 5 V power supply line ($V_{DD}$) and the reference clock (RCL) are continuously checked by the ELIC for spikes that may disturb the proper operation of the ELIC. If such an inappropriate clocking or power failure occurs, data in the internal memories may be lost, and a reinitialization of the ELIC is necessary. An Initialization Request status bit (VNSR:IR) can be interrogated periodically by the µP to determine the current status of the device.

In normal chip operation, the IR bit should never be set, not even after power on or when the clock signals are switched on and off. The IR bit will only be set if spikes (< 10 ns) are detected on the clock and power lines which may affect the data transfer on the ELIC internal buses.

## 5.9 Applications

### 5.9.1 Analog IOM®-2 Line Card with SICOFI®-4 as Codec/Filter Device

The line card consists of an ELIC (PEB 20550) device which handles the monitor and the signaling channels of up to 16 SICOFI-4 (PEB 2465) devices. Since each SICOFI-4 supports four analog lines, up to 64 analog subscriber lines (t/r lines) can be accommodated.

**Figure 115** shows the interconnection of the ELIC, and the SICOFI-4 devices via the IOM-2 interface:

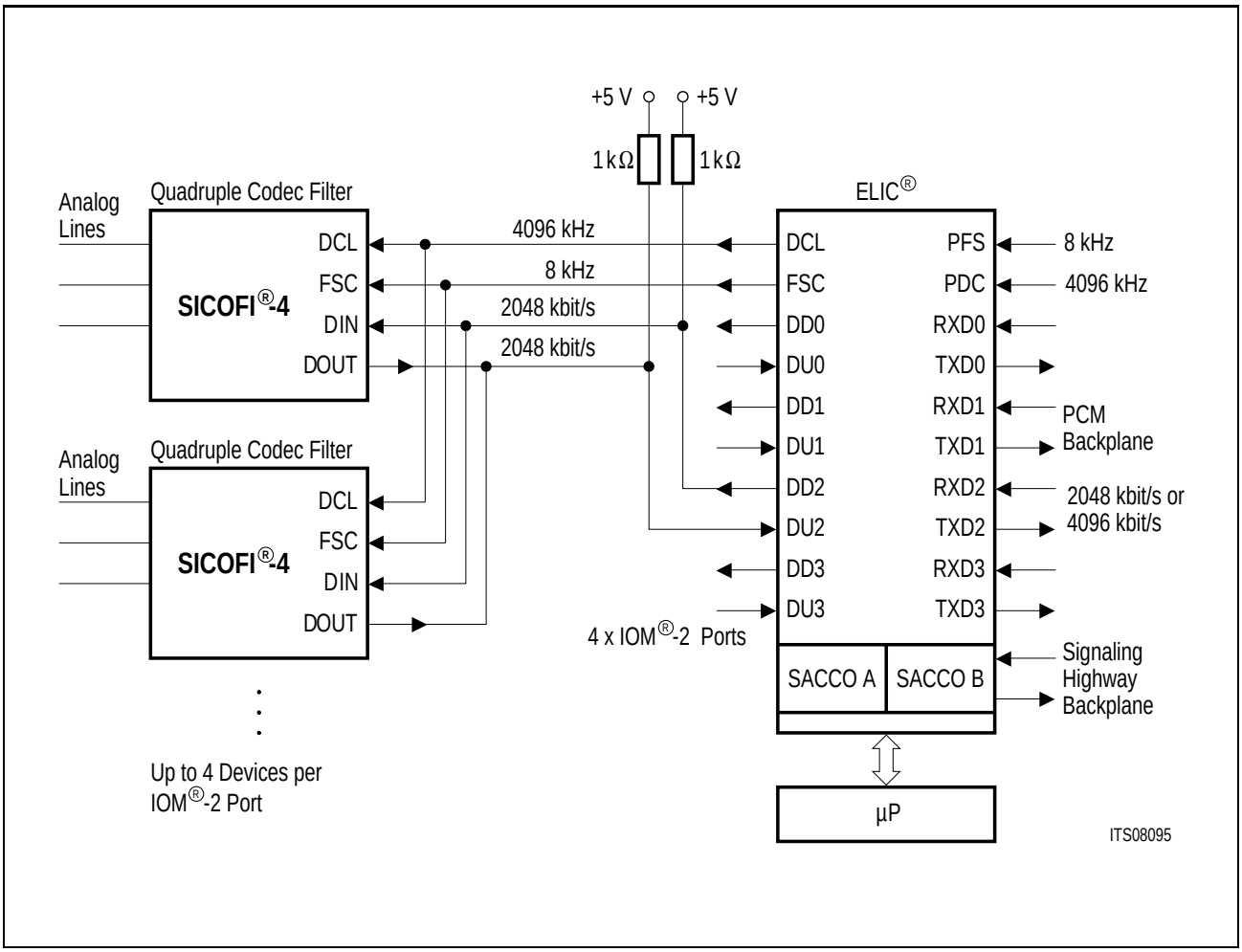



**Figure 115**
**Analog Line Card with SICOFI®-4 Devices Using the IOM®-2 Interface**

A typical timing example for the connection of the line card to a 2048 kBit/s PCM backplane is shown in **figure 116**. It should be noted that the PCM interface must be clocked with a 4096 kHz clock even if the PCM interface operates at only 2048 kBit/s. This is to obtain a DCL output frequency of 4096 kHz, which is required for the IOM-2 timing.

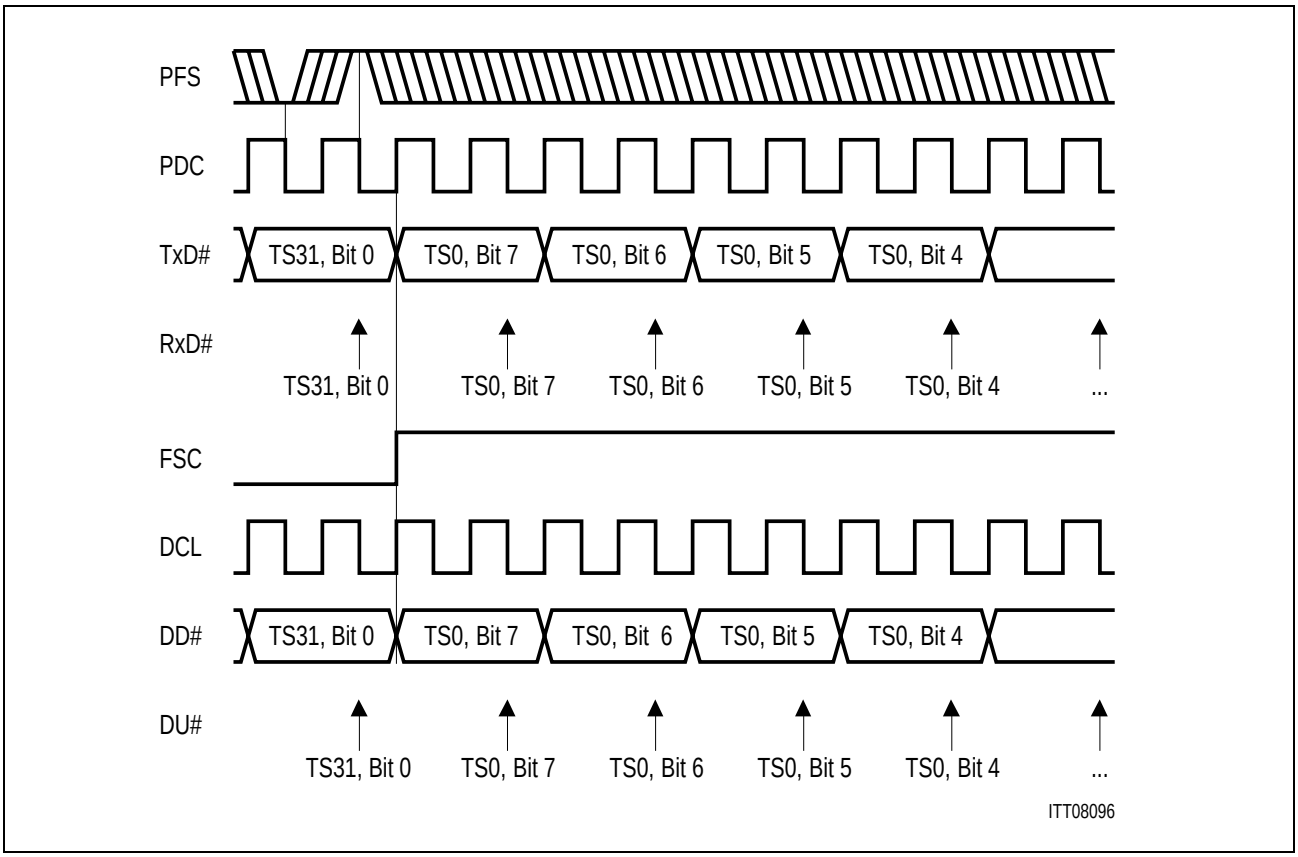



**Figure 116**
**Typical IOM®-2 Line Card Timing**

Based on these PCM and CFI timing requirements, the following ELIC initialization values for the PCM and CFI registers are recommended:

**ELIC®**

| | | | | | |
|---|---|---|---|---|---|
| PMOD | = | $0010\ 0000_B$ | = | $20_H$ | PCM mode 0, double rate clock, PFS evaluated with falling clock edge, PCM comparison disabled |
| PBNR | = | $1111\ 1111_B$ | = | $FF_H$ | 256 bits (32 ts) per PCM frame |
| POFD | = | $1111\ 0000_B$ | = | $F0_H$ | PFS marks downstream PCM TS0, bit 7 |
| POFU | = | $0001\ 1000_B$ | = | $18_H$ | PFS marks upstream PCM TS0, bit 7 |
| PCSR | = | $0000\ 0001_B$ | = | $01_H$ | PCM data received with falling, transmitted with rising clock edge |
| CMD1 | = | $0010\ 0000_B$ | = | $20_H$ | PDC/PFS clock source, PFS evaluated with falling clock edge, prescaler = 1, CFI mode 0 |
| CMD2 | = | $1101\ 0000_B$ | = | $D0_H$ | FC mode 6, double rate clock, CFI data transmitted with rising, received with falling clock edge |
| CBNR | = | $1111\ 1111_B$ | = | $FF_H$ | 256 bits (32 ts) per CFI frame |

| | | | | | |
|---|---|---|---|---|---|
| CTAR | = | $0000\ 0010_B$ | = | $02_H$ | PFS marks downstream CFI TS0 |
| CBSR | = | $0010\ 0000_B$ | = | $20_H$ | PFS marks downstream CFI bit 7, upstream bits not shifted |
| CSCR | = | $0000\ 0000_B$ | = | $00_H$ | 64, 32, 16 kBit/s channels located on CFI TS bits 7 … 0, 7 … 4, 7 … 6 |

Each SICOFI-4 device must be assigned to its individual IOM-2 channels by pin-strapping. The SICOFI-4 coefficients (filter characteristics, gain, … ) as well as other operation parameters, are programmed via the ELIC over the IOM-2 monitor channel.

**Example**

Initializing 4 consecutive CFI timeslots as an analog IOM-2 channel.

Timeslots 0, 1, 2, and 3 of CFI port 2 shall represent the IOM channel 0 of port 2. Timeslots 4, 5, 6, and 7 of CFI port 2 shall represent the IOM channel 1 of port 2. This requires the SICOFI-4 to be pin-strapped to that slot by connecting pin TSS0 and pin TSS1 to 0 V.

Timeslots 4 and 5 represent the two B channels that may for example be switched to the PCM interface. Timeslots 6 and 7 represent the monitor and signaling (SIG) channels and must be initialized in the ELIC control memory (CM):

| | | | |
|---|---|---|---|
| W: MADR | = | $FF_H$ | ; 6 bit signaling value to be transmitted in timeslot 7 |
| W:MAAR | = | $1C_H$ | ; CFI address of downstream IOM port 2, timeslot 6 |
| W: MACR | = | $7A_H$ | ; writing CM with code ‘1010’ |
| W: MADR | = | $FF_H$ | ; value don’t care, e.g. FF |
| W: MAAR | = | $1D_H$ | ; CFI address of downstream IOM port 2, timeslot 7 |
| W: MACR | = | $7B_H$ | ; writing CM with code ‘1011’ |
| W: MADR | = | $FF_H$ | ; 6 bit signaling value expected upon initialization in timeslot 7 |
| W: MAAR | = | $9C_H$ | ; CFI address of upstream IOM port 2, timeslot 6 |
| W: MACR | = | $7A_H$ | ; writing CM with code ‘1010’ |
| W: MADR | = | $FF_H$ | ; 6 bit signaling value expected upon initialization in timeslot 7 |
| W: MAAR | = | $9D_H$ | ; CFI address of upstream IOM port 2, timeslot 7 |
| W: MACR | = | $7A_H$ | ; writing CM with code ‘1010’ |

The above steps have to be repeated for all timeslots that shall be handled by the monitor or signaling handler of the ELIC (i.e. TS2 and TS3, TS10 and TS11, TS14 and TS15).

Example for programming the CODEC corresponding to TS6 of the SICOFI-4:

W: OMDR = $EE_H$ ; activation of ELIC with active handshake protocol
W: MFSAR = $0E_H$ ; monitor address for port 2, timeslot 6
W: CMDR = $01_H$ ; MFFIFO reset
R: STAR = $25_H$ ; MFFIFO write access enabled
W: MFFIFO = $81_H$ ; SICOFI-4 monitor address
W: MFFIFO = $14_H$($94_H$) ; SICOFI-4 channel A (B) data
W: MFFIFO = $00_H$ ; SICOFI-4 data
W: MFFIFO = $00_H$ ; SICOFI-4 data
W: MFFIFO = $00_H$ ; SICOFI-4 data
W: MFFIFO = $00_H$ ; SICOFI-4 data
W: CMDR = $00_H$ ; transmit command
Wait for interrupt!
R: ISTA = $20_H$; MFFI interrupt
R: STAR = $25_H$; transfer completed, MFFIFO write access enabled
Reading back data from SICOFI-4:
W: MFSAR = $0E_H$ ; monitor address for port 2, timeslot 6
W: CMDR = $01_H$ ; MFFIFO reset
R: STAR = $25_H$ ; MFFIFO write access enabled
W: MFFIFO = $81_H$ ; SICOFI-4 monitor address
W: MFFIFO = $65_H$($E5_H$) ; SICOFI-4 channel A (B) data, read back request
W: CMDR = $08_H$ ; transmit and receive command
Wait for interrupt!
R: ISTA = $20_H$ ; MFFI interrupt
R: STAR = $26_H$ ; transfer completed, MFFIFO not empty, read access enabled
R: MFFIFO = $81_H$ ; SICOFI-4 monitor address
R: MFFIFO = $00_H$ ; SICOFI-4 data
R: MFFIFO = $00_H$ ; SICOFI-4 data
R: MFFIFO = $00_H$ ; SICOFI-4 data
R: MFFIFO = $00_H$ ; SICOFI-4 data
R: STAR = $27_H$ ; transfer completed, MFFIFO empty, read access enabled

### 5.9.2 IOM®-2 Trunk Line Applications

Trunk lines connect the PBX to the central office (CO) network. **Figure 117** gives an overview of the different access possibilities to the central office.

One possibility is to use analog a/b lines. This is the most uncomplicated way since no clock recovery from the CO is required, i.e. the PBX operates with a free running crystal oscillator. The t/r access to the CO can easily be realized with one or several SICOFI-2 or SICOFI-4 codec/filter devices, which allow the connection of two or four analog lines per chip.

If an access to the ISDN world is desired, two options are possible:

For small PBXs, with only few external lines, one or several Basic Rate ISDN (BRI) connections are best suited. Each BRI connection provides a capacity of two B channels of 64 kBit/s and one D channel of 16 kBit/s. The BRI connection is usually performed via the T interface to the Network Terminator 1 (NT1). The T interface is physically identical to the S interface, all Siemens $S_0$ interface devices (QUAT-S, SBCX, ISAC-S, SBC) can be used for that purpose. A PBX can also be connected directly via the $U_k$- interface to the CO. In this case an IEC-Q device (2B1Q encoding) or an IEC-T (4B3T encoding) can be used as layer-1 device.

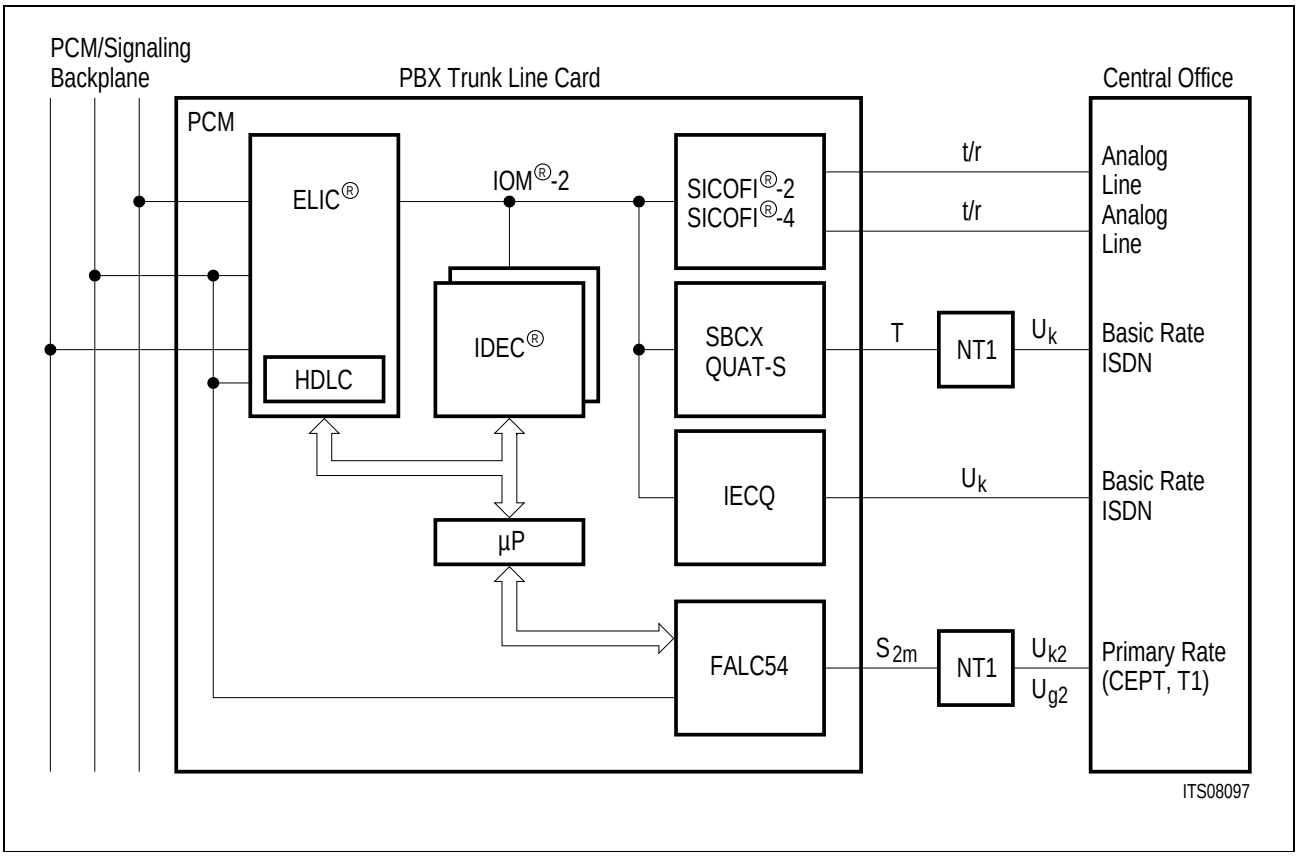



**Figure 117**
**Overview of Trunk Line Applications**

For large PBXs, with many external lines, one or several Primary Rate ISDN (PRI) connections are more advantageous. If the European CEPT standard is used, each PRI connection provides 30 B channels of 64 kBit/s each and one D channel of 64 kBit/s. The FALC54 can be used to implement the Primary Rate $S_{2m}$ interface according to the CEPT (2048 kBit/s) or the T1 (1544 kBit/s) standards. For both standards a common backplane data rate of 2048 or 4096 kBit/s can be selected to simplify the connection to the PBX internal PCM highway, which usually consists of 32 or 64 timeslots.

Digital trunk lines require a clock recovery from the received data stream such that the PBX clock system is locked up with the CO clock system. The examples given in the following chapters show how to deal with these points.

#### 5.9.2.1 PBX With Multiple ISDN Trunk Lines

In a trunk unit special attention must be given to the clock synchronization. The PBX clock generator must deliver a stable free running clock as long as no external calls are active. When an external call is established, the CO must be taken as reference to synchronize the local PBX clock system.

The Siemens $S_0$-layer-1 transceivers SBC, SBCX, QUAT-S and ISAC-S are prepared for this kinds of applications: In the LT-T (Line Termination at the T-reference point) mode, they deliver a clock signal that is synchronous to the incoming S-frame. This clock signal can be taken to synchronize the PCM clocks of the ELIC by means of a XTAL controlled PLL circuit. Since the ELIC generates the IOM-2 clocks for the connected layer-1 and layer-2 devices, the loop is closed. If several layer-1 devices are operated in LT-T mode, only 1 device may be selected to deliver the reference clock. The PABX software must determine an active line by evaluating the C/I indications of the layer-1 devices in order to select an appropriate clock source for the PLL. If several external lines are active, any of these lines can be taken, since the CO lines are synchronous among each other.

The layer-1 devices have a built-in frame buffer that compensates the phase offset that may persist between the IOM-2 frame and the $S_0$-frame. This buffer is ‘elastic’, such that a frame wander and jitter between the IOM-2 and the S-frame can be tolerated up to a certain extent. The maximum ‘wander’ value is device specific. For the SBCX, for example, 50µs of frame deviation are internally compensated. If this value is exceeded, a frame slip occurs that is reported to the µP by a ‘slip’ indication in the C/I code. If a frame slip occurs, the data of an S-frame may be lost or transferred twice. The slip indications can be evaluated for statistical purposes. However, in a final design with optimized PLL tracking, slips should not occur during normal operation of the PBX.

Since the $S_0$ interface allows bus configurations for terminals (TEs), and since it is physically possible to connect a PBX trunk line together with other PBX trunk lines, or with normal ISDN terminals, to a common S-bus, the trunk lines must also follow the D-channel access procedure specified for ISDN terminals. This D-channel access procedure is implemented in the QUAT-S, ISAC-S and SBCX devices and can optionally

be set. If not required, the D-channel can also be sent transparently. If the QUAT-S is used together with the IDEC as layer-2 controller, the IDEC must be informed about the availability of the D-channel at the T-interface. The QUAT-S provides an enable signal at pin DRDY that carries this information during the D-channel timeslot. This signal can be connected to the collision data input (CDR) of the IDEC to enable or disable HDLC transmission. The IDEC must then be programmed to the ‘slave mode’ in order to evaluate the CDR pin.

**Figure 118** illustrates a complete PBX trunk card, where the ELIC controls up to 8 QUAT-S devices connected to up to 4 IOM-2 ports. On each IOM-2 port 2 IDECs take care of the D-channel processing. The CDR input lines of the IDECs are connected with the DRDY output pins of the QUAT-S. This is to stop the HDLC controllers in case of a D-channel collision on the T-bus. The QUAT-S devices must be programmed via the monitor channel to deliver appropriate Stop/Go information at pin DRDY. The 1536 kHz reference clock outputs (pin CLK1) of the QUAT-Ss are fed via a multiplexer to the PBX clock generator. The µP controls the multiplexer as required by the state of the lines.

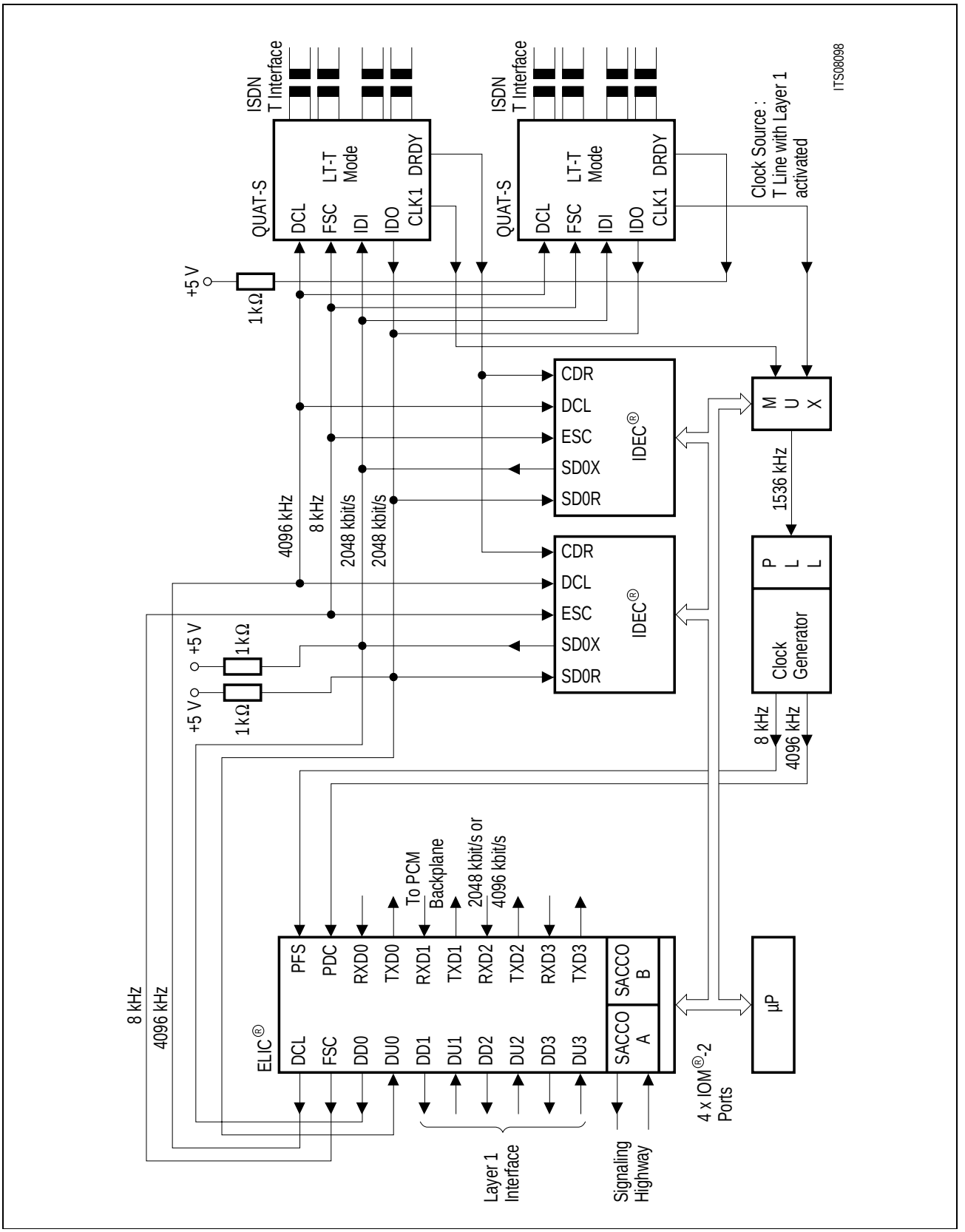



**Figure 118**
**PBX Trunk Card for Multiple Basic Rate Trunk Lines Using the QUAT-S**

Initialization values for the IDEC that controls the lower 4 channels of the IOM-2 interface:

**IDEC®**

| | | | | |
|---|---|---|---|---|
| CCR | = 1000 0010B | = | $82_H$ | IOM-2 mode, IOM ch. 0 - 3, double clock rate, 256 bits/frame |
| A_MODE | = 0000 1100B | = | $0C_H$ | uncond. trans., 16 kBit/s ch., channel and receiver active |
| A_TSR | = 0000 1100B | = | $0C_H$ | ch. A timeslot position: D channel of IOM ch. 0 |
| B_MODE | = 0000 1100B | = | $0C_H$ | uncond. trans., 16 kBit/s ch., channel and receiver active |
| B_TSR | = 0001 1100B | = | $1C_H$ | ch. B timeslot position: D channel of IOM ch. 1 |
| C_MODE | = 0000 1100B | = | $0C_H$ | uncond. trans., 16 kBit/s ch., channel and receiver active |
| C_TSR | = 0010 1100B | = | $2C_H$ | ch. C timeslot position: D channel of IOM ch. 2 |
| D_MODE | = 0000 1100B | = | $0C_H$ | uncond. trans., 16 kBit/s ch., channel and receiver active |
| D_TSR | = 0011 1100B | = | $3C_H$ | ch. D timeslot position: D channel of IOM ch. 3 |

Initialization values for the IDEC that controls the upper 4 channels of the IOM-2 interface:

**IDEC®**

| | | | | |
|---|---|---|---|---|
| CCR | = 1000 0010B | = | $A2_H$ | IOM-2 mode, IOM ch. 4-7, double clock rate, 256 bits/frame |
| A_MODE | = 0000 1100B | = | $0C_H$ | uncond. trans., 16 kBit/s ch., channel and receiver active |
| A_TSR | = 0100 1100B | = | $4C_H$ | ch. A timeslot position: D channel of IOM ch. 4 |
| B_MODE | = 0000 1100B | = | $0C_H$ | uncond. trans., 16 kBit/s ch., channel and receiver active |
| B_TSR | = 0101 1100B | = | $5C_H$ | ch. B timeslot position: D channel of IOM ch. 5 |
| C_MODE | = 0000 1100B | = | $0C_H$ | uncond. trans., 16 kBit/s ch., channel and receiver active |
| C_TSR | = 0110 1100B | = | $6C_H$ | ch. C timeslot position: D channel of IOM ch. 6 |
| D_MODE | = 0000 1100B | = | $0C_H$ | uncond. trans., 16 kBit/s ch., channel and receiver active |
| D_TSR | = 0111 1100B | = | $7C_H$ | ch. D timeslot position: D channel of IOM ch. 7 |

The ELIC initialization is the same as for the IOM-2 application described previously in this chapter.

If the D-channel access procedure is programmed, the IDEC MODE registers must additionally be programmed accordingly i.e. for each channel MODE = $2C_H$ (instead of $0C_H$).

**Example**

In a first step, the QUAT-S in IOM port 0, ch. 0 … 3 is programmed via the IOM-2 monitor handler to the LT-T mode:

| | | | |
|---|---|---|---|
| W:OMDR | = | $EE_H$ | ; activation ELIC with handshake protocol enabled |
| W:MFSAR | = | $04_H$ | ; monitor address of IOM port 0, channel 0 |
| W:CMDR_E | = | $01_H$ | ; reset MFFIFO |
| R:STAR_E | = | $25_H$ | ; MFFIFO write access enabled |
| W:MFFIFO | = | $81_H$ | ; select QUAT-S Configuration Register |
| W:MFFIFO | = | $41_H$ | ; set LT-T mode, output CLK1 |
| W:CMDR_E | = | $04_H$ | ; transmit MFFIFO content |
| R:ISTA_E | = | $20_H$ | ; MFFI interrupt |
| W:MFSAR | = | $0C_H$ | ; monitor address of IOM port 0, channel 1 |
| W:CMDR_E | = | $01_H$ | ; reset MFFIFO |
| R:STAR_E | = | $25_H$ | ; MFFIFO write access enabled |
| W:MFFIFO | = | $81_H$ | ; select QUAT-S Configuration Register |
| W:MFFIFO | = | $01_H$ | ; set LT-T mode |
| W:CMDR_E | = | $04_H$ | ; transmit MFFIFO content |
| R:ISTA_E | = | $20_H$ | ; MFFI interrupt |
| W:MFSAR | = | $14_H$ | ; monitor address of IOM port 0, channel 2 |
| W:CMDR_E | = | $01_H$ | ; reset MFFIFO |
| R:STAR_E | = | $25_H$ | ; MFFIFO write access enabled |
| W:MFFIFO | = | $81_H$ | ; select QUAT-S Configuration Register |
| W:MFFIFO | = | $01_H$ | ; set LT-T mode |
| W:CMDR_E | = | $04_H$ | ; transmit MFFIFO content |
| R:ISTA_E | = | $20_H$ | ; MFFI interrupt |
| W:MFSAR | = | $1C_H$ | ; monitor address of IOM port 0, channel 3 |
| W:CMDR_E | = | $01_H$ | ; reset MFFIFO |
| R:STAR_E | = | $25_H$ | ; MFFIFO write access enabled |
| W:MFFIFO | = | $81_H$ | ; select QUAT-S Configuration Register |
| W:MFFIFO | = | $01_H$ | ; set LT-T mode |
| W:CMDR_E | = | $04_H$ | ; transmit MFFIFO content |
| R:ISTA_E | = | $20_H$ | ; MFFI interrupt |

#### 5.9.2.2 Small PBX

**Figure 118** shows a realization example of a small PBX. If the total number of lines (internal or external) is smaller than the capacity of the ELIC (32 × (2 × B + D) or 64 × B), the PCM interface of the ELIC need not to be connected to a switching network since all the B (and D) channel switching can be done inside the ELIC. In this special case, it is sufficient to apply only a PCM clock to the ELIC, the PCM frame synchronization signal (8 kHz) can be omitted. The IOM-2 clock and framing signals DCL and FSC are still generated correctly by the ELIC. The STAR:PSS bit should then not be evaluated: it stays at logical 0 all the time.

The PBX shown in the **figure 118** offers 8 analog (t/r) subscriber lines, realized with two quadruple codec/filter devices SICOFI-4 (PEB 2465) and one digital, so subscriber interface realized with the SBCX (PEB2081).

The **figure 119** also shows a digital trunk line (external line) which is realized with a $U_k$-layer-1 device, IEC-Q (PEB 2091), operated in NT-PABX mode. The PBX can therefore be connected directly to the $U_k$ interface coming from the CO. The NT-PABX mode of the $U_k$- layer-1 devices is similar to the LT-T mode of the S layer-1 devices: in both cases the layer-1 device delivers a reference clock which is synchronous to the received S or $U_k$- frame and that can be used to synchronize the local PBX clock generator. Any phase differences between the local IOM-2 frame and the received S or $U_k$- frame are compensated for in an elastic buffer inside the layer-1 devices.

Signaling control for the $S_0$ subscriber interface is performed by the ELIC SACCO-A HDLC controller.

Since the digital trunk line also needs a D channel handler, the ELIC SACCO-B HDLC controller is assigned to that IOM-2 channel.

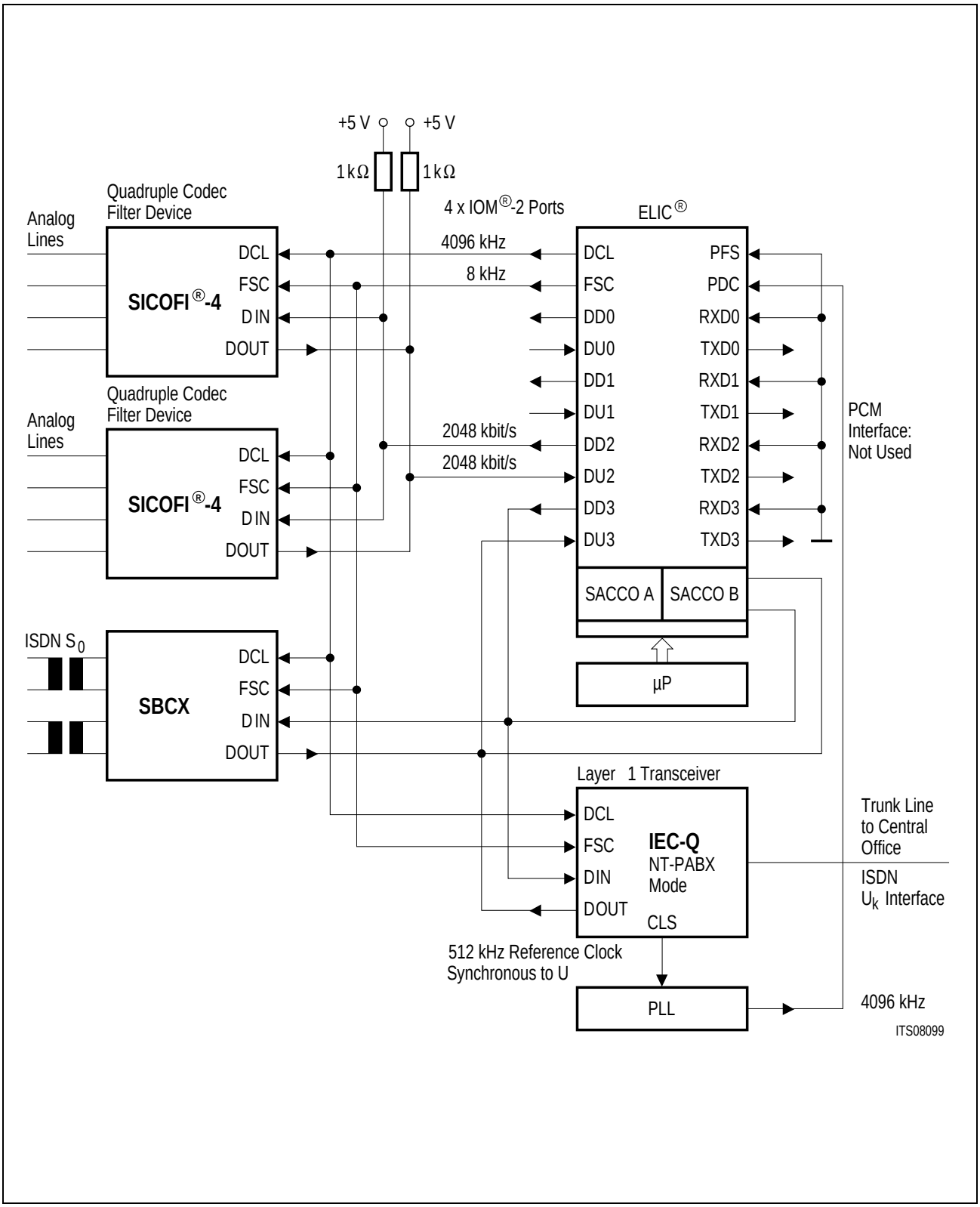



**Figure 119**
**Small PBX with SICOFI®-4, SBCX and IEC-Q**

### 5.9.3 Miscellaneous

#### 5.9.3.1 Interfacing the ELIC® to a MUSAC™

The PCM interface of the ELIC can easily be connected to the Multipoint Switching and Conferencing circuit MUSAC (PEB 2245) when using the set-up and PCM timing as shown in **figure 120**. This configuration can then for example be used in a PBX to implement conferencing functions for up to 21 simultaneous conferences.

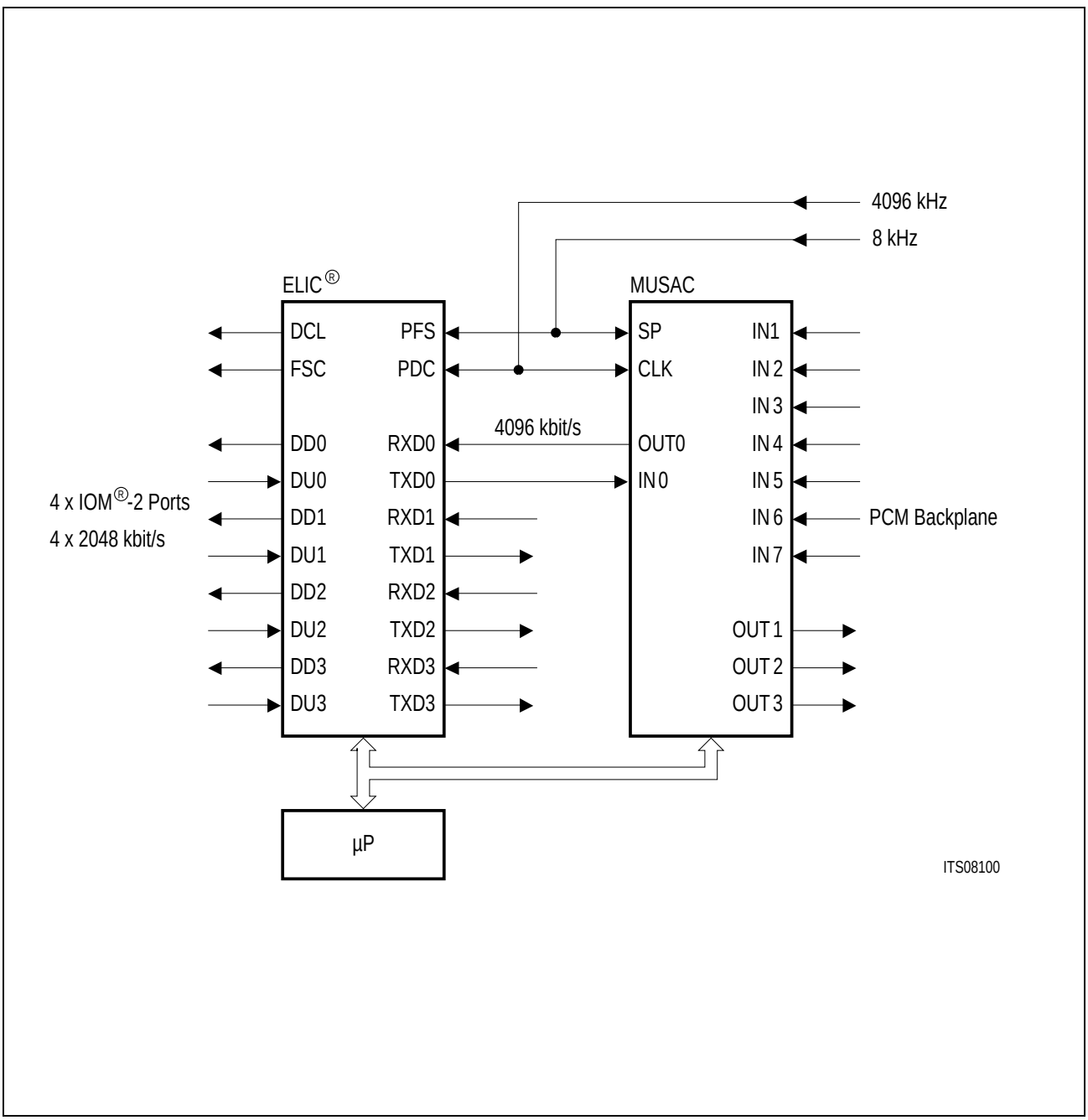



**Figure 120**
**Interconnection Example ELIC® - MUSAC™**

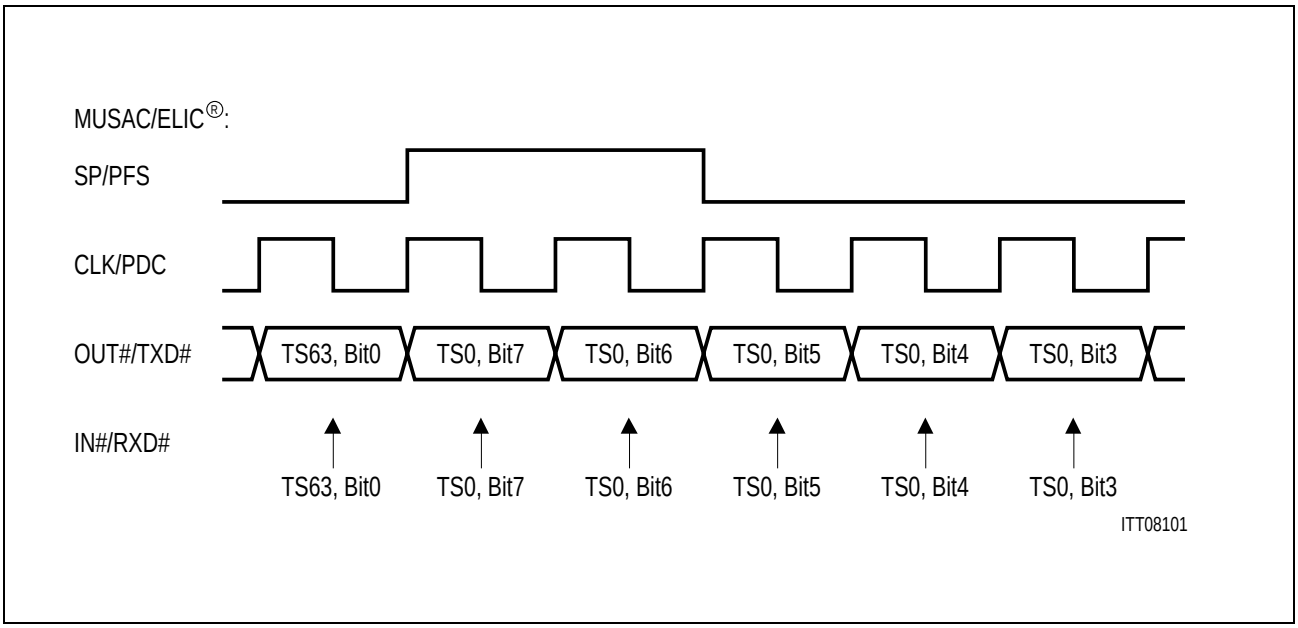



**Figure 121**
**Timing Example to Interconnect the ELIC® and the MUSAC™ on an IOM®-2 Line Card**

The following values must be programmed to the PCM and CFI registers of the ELIC and to the MOD and CFR registers of the MUSAC to obtain the desired PCM and IOM-2 timing:

**ELIC®**

| | | | | |
|---|---|---|---|---|
| PMOD | = 0100 0100$_B$ | = | 44$_H$ | PCM mode 1, single rate clock, PFS evaluated with falling clock edge, input selection RxD0 and RxD3, PCM comparison disabled |
| PBNR | = 1111 1111$_B$ | = | FF$_H$ | 512 bits (64 ts) per PCM frame |
| POFD | = 1111 0000$_B$ | = | F0$_H$ | PFS marks downstream PCM TS0, bit 6 |
| POFU | = 0001 1000$_B$ | = | 18$_H$ | PFS marks upstream PCM TS0, bit 6 |
| PCSR | = 0100 0101$_B$ | = | 45$_H$ | PCM data received with falling, transmitted with rising clock edge |
| CMD1 | = 0010 0000$_B$ | = | 20$_H$ | PDC/PFS clock source, PFS evaluated with falling clock edge, prescaler = 1, CFI mode 0 |
| CMD2 | = 1101 0000$_B$ | = | D0$_H$ | FC mode 6, double rate clock, CFI data transmitted with rising, received with falling clock edge |
| CBNR | = 1111 1111$_B$ | = | FF$_H$ | 256 bits (32 ts) per CFI frame |

| | | | | |
|---|---|---|---|---|
| CTAR | = 0000 0010$_B$ | = | 02$_H$ | PFS marks downstream CFI TS0 |
| CBSR | = 0010 0000$_B$ | = | 20$_H$ | PFS marks downstream CFI bit 6, upstream bits not shifted |
| CSCR | = 0000 0000$_B$ | = | 00$_H$ | 64, 32, 16 kBit/s channels located on CFI bits 7 … 0, 7 … 4, 7 … 6 |

**MUSAC™**

| | | | | |
|---|---|---|---|---|
| MOD | = 0100 0100$_B$ | = | 03$_H$ | input mode 8 × 4M, output mode 4 × 4M |
| CFR | = 1111 1111$_B$ | = | DE$_H$ | 4.096 MHz device clock, conferencing mode, A-law, even bits inverted |

### 5.9.3.2 Space and Time Switch for 16 kBit/s Channels

The ELIC is optimized for the space and time switching of 64 kBit/s channels (8 bit timeslots). The switching of 32 and 16 kBit/s subchannels is also supported, but these channels can only be freely selected at the PCM interface. At the CFI, only one subchannel per 8 bit timeslot can be switched (see **chapter 5.4.2**). Usually, this is sufficient because on the IOM-2 interface, only one 16 kBit/s D channel per timeslot needs to be switched. Up to four D channels may then be combined into a single 8 bit PCM timeslot.

If a completely flexible space and time switch for contiguous 16 kBit/s channels is required, the following method can be used:

The four CFI ports are connected in parallel as shown in **figure 122**. Each CFI port is programmed via the CFI subchannel register (CSCR) to handle a different 2 bit sub-timeslot position. With this configuration, any mixture of 16, 32 and 64 kBit/s channels may be switched between the CFI and the PCM interfaces. Up to 128 16 kBit/s channels per direction can be handled by the ELIC. The switching software must select the CFI port number according to the required CFI subchannel position for each CFI - PCM connection. The PCM subchannel position is selected via the control memory (CM) code field (see **table 40**). For 32 and 64 kBit/s connections, only one CFI port of a given timeslot may be programmed in order to avoid collisions on the CFI ‘bus’.

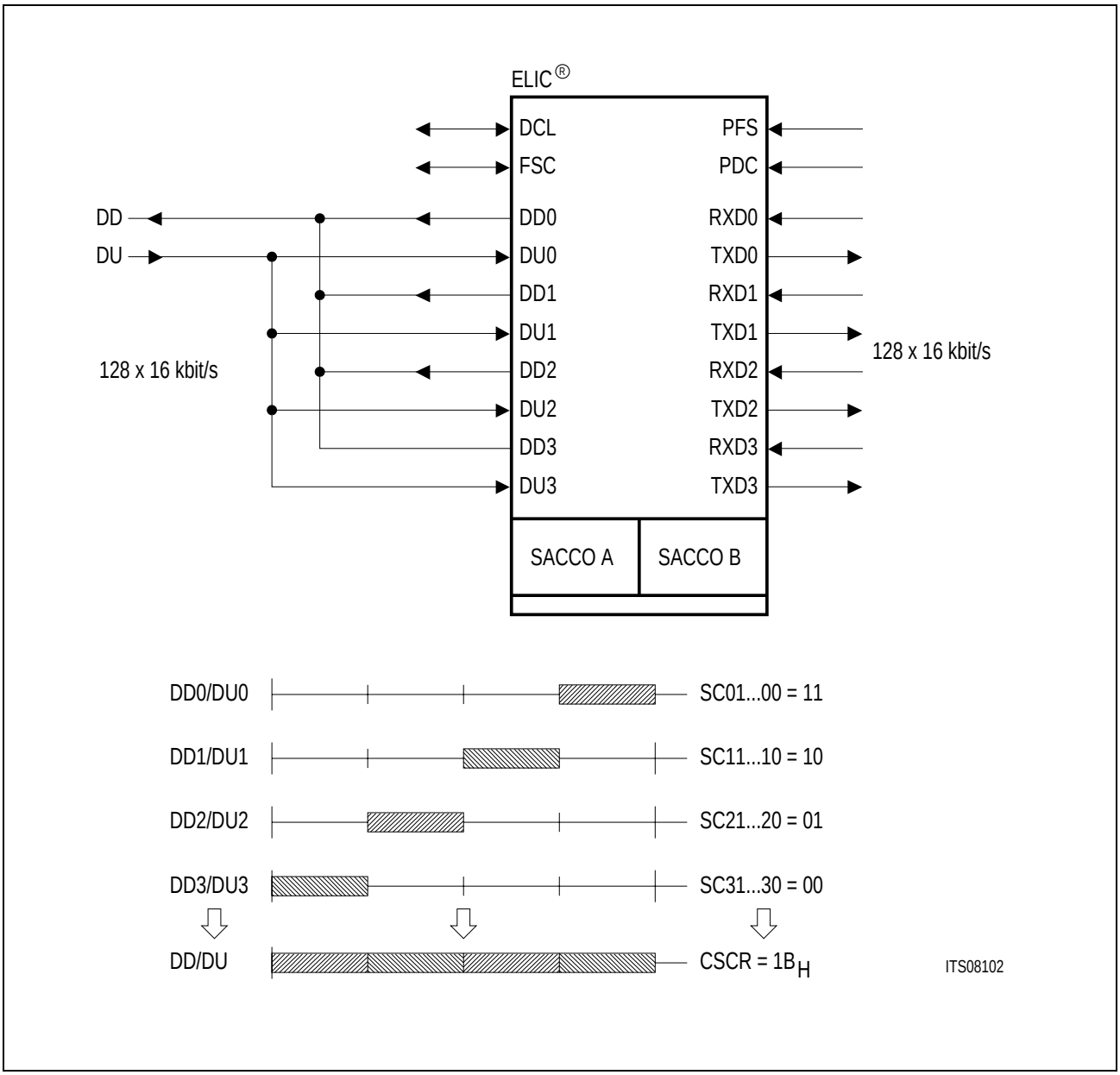



**Figure 122**
**Non-blocking Space and Time Switch for 16 kBit/s Channels**

# 6 Application Notes

## 6.1 Example of ELIC® Operation in a Digital PBX

### 6.1.1 Introduction

In an ISDN system the digital line card connects subscribers to the PCM highway as well as to each other. The optimum controlling device for such a line card is the Extended Line Card Interface Controller (ELIC) PEB 20550. Integrating the PCM interface controller EPIC, two independent HDLC controllers SACCO-A and SACCO-B, and a D-channel arbiter onto a single chip, the ELIC offers a high degree of specialisation, while maintaining wide-ranging flexibility.

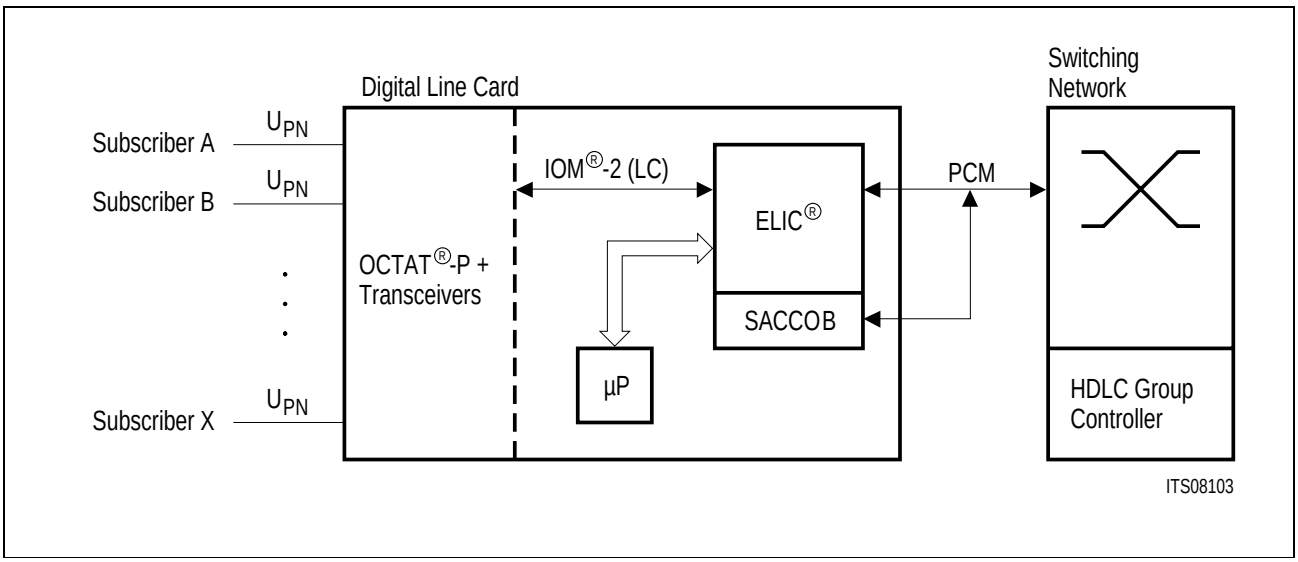



**Figure 123**
**Digital PBX**

This Application Note demonstrates operation of the ELIC as the key device of the digital PBX shown in **figure 123**, by establishing a connection from subscriber A to subscriber B. In this application the Configurable Interface (CFI) of the EPIC is set to IOM-2 (LC) protocol, and thus the D-channel arbiter is used to link the SACCO-A to the CFI.
In particular this Application Note will show:

- how to initialize the ELIC CFI and PCM interface
- how to initialize SACCO-A and D-channel Arbiter to handle D-channel data
- how to initialize the SACCO-B to communicate via the PCM highway
- how to initialize the ELIC Control Memory to handle, monitor, control, and B-channels
- how to use the ELIC C/I channel to control the OCTAT-P
- how to use the SACCO-A for D-channel communication with subscribers
- how to switch a telephone connection between subscribers A and B.

As a starting position this Application Note assumes that ELIC and OCTAT-P have been reset.

### 6.1.2 Basic Initialization

This part of the Application Note modifies the initialization example of the ELIC Technical Manual to interface two digital IOM-2 subscribers to a 2 Mbit PCM 30 switching network. As shown below, the ELIC is configured to accept a 4 MHz clock as PDC and HDCB input and to output a CFI clock and frame sync.

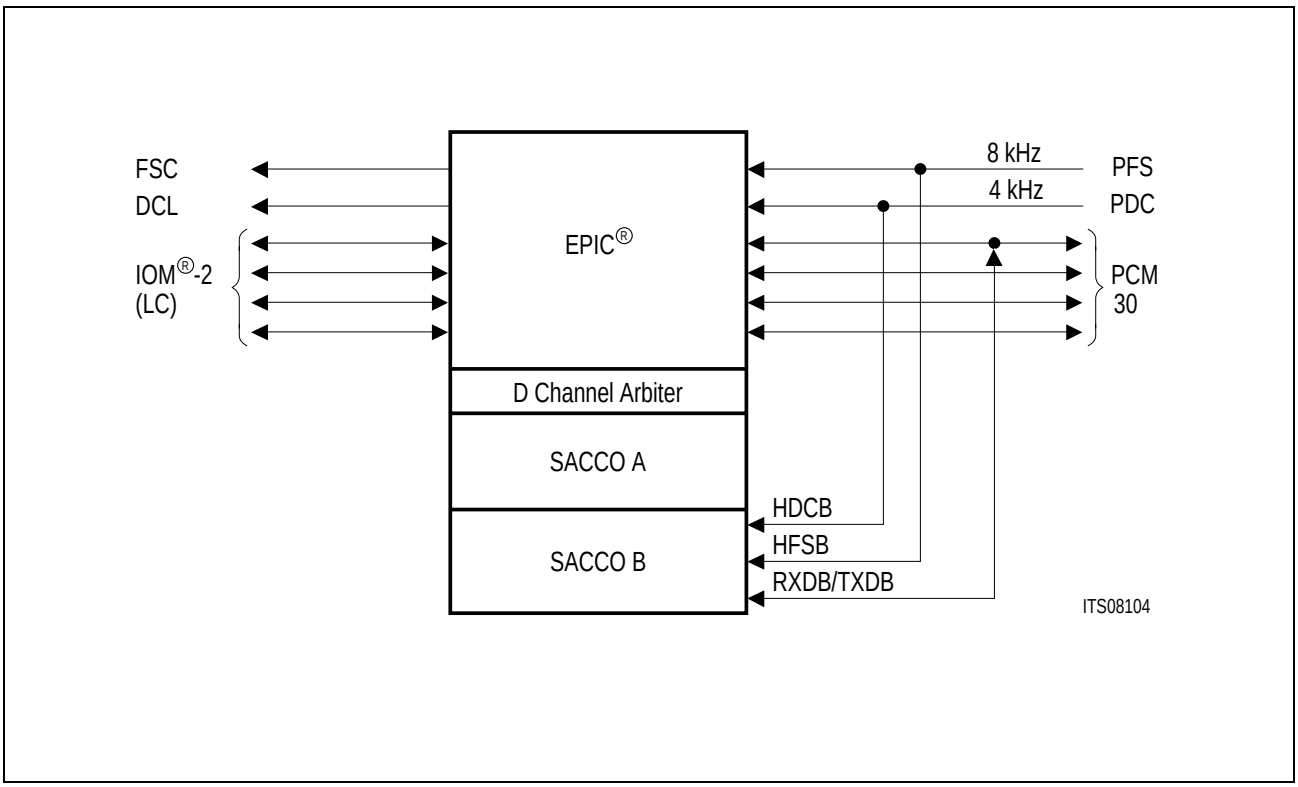



**Figure 124**
**Principal ELIC® Interfaces**

#### 6.1.2.1 EPIC® Interface Initialization

Configuration of the PCM interface: Set-up to meet the requirements of the switching network (as for W&G PCM 4 measurement device)

| | | |
|---|---|---|
| Write | PMOD = $20_H$ | PCM mode 0, double rate clock (4 MHz, 2 Mbit/s), PFS evaluated with falling edge of PDC |
| Write | PBNR = $FF_H$ | 256 bits per PCM frame |
| Write | POFD = $F1_H$ | with BPF = 256, the internal PFS marks downstream bit number (BND) 2; for detail, refer to the **Application Hints, figure 62** |
| Write | POFU = $19_H$ | the internal PFS marks upstream bit number (BNU) 2; for detail, refer to the **Application Hints, figure 62** |
| Write | PCSR = $01_H$ | no clock shift; PCM data sampled with falling, transmitted with rising PDC |

Configuration of the CFI side: Set-up for IOM-2 subscribers

| | | |
|---|---|---|
| Write | CMD1 = $20_H$ | PDC and PFS used as clock and framing source for the CFI; CRCL = PDC; prescaler divisor = 1; CFI mode 0 |
| Write | CMD2 = $D0_H$ | FSC shaped for IOM 2 interface; DCL = 2 x data rate; CFI data received with falling, transmitted with rising CRCL |
| Write | CBNR = $FF_H$ | 256 bits per CFI frame |
| Write | CTAR = $02_H$ | PFS is to mark CFI time slot 0 |
| Write | CBSR = $20_H$ | PFS is to mark bit 7 of CFI time slot 0; no shift of CFI upstream data relative to CFI downstream data |
| Write | CSCR = $00_H$ | 2 bit channels located in positions 7, 6 on all CFI ports |

#### 6.1.2.2 SACCO-A Initialization

Initialization of SACCO-A for operation with D-channel Arbiter:

| | | |
|---|---|---|
| Write | MODE = $98_H$ | transparent mode 0; continuous frame transmission switched ON; HDLC receiver active; test loop disabled |
| Write | CCR1 = $87_H$ | power up; point-to-point configuration. IDLE sequences as interframe output; double rate data clock; clock mode 3 |

Reset SACCO-A:

| | | |
|---|---|---|
| Write | CMDR = $C1_H$ | Receive Message Complete (RMC); Reset HDLC Receiver (RHR); Transmitter Reset (XRES) |
| Read | ISTA_A= $10_H$ | transmit pool ready |

#### 6.1.2.3 Basic D-Channel Arbiter Initialization

| | | | |
|---|---|---|---|
| Write | AMO | = $69_H$ | full selection counter set to general worst case delay of 14 frames; suspend counter active; arbiter control via C/I channel; control channel activated |

#### 6.1.2.4 SACCO-B Initialization

Initialization of SACCO-B for communication, via the PCM highway, with the (non-PBC) group controller:

| | | | |
|---|---|---|---|
| Write | MODE | = $48_H$ | 8 bit non-auto mode; continuous frame transmission OFF; HDLC receiver active; test loop disabled |
| Write | TSAX | = $0B_H$ | assign transmit time slot 3: set XCS bits to shift output window to time slot 1, set TSNX1 bit to delay the output window by 2 time slots |
| Write | TSAR | = $0B_H$ | assign receive time slot 3: set RCS bits to shift input window to time slot 1, set TSNR1 bit to delay the input window by 2 time slots |
| Write | XCCR | = $07_H$ | 8 bits transmitted per output window |
| Write | RCCR | = $07_H$ | 8 bits received per input window |
| Write | RAL1 | = $09_H$ | receive address |
| Write | RAL2 | = $FE_H$ | broadcast receive address |
| Write | CCR2 | = $38_H$ | set XCS and RCS bits (see TSAX, TSAR); enable TxDB pin |
| Write | CCR1 | = $9E_H$ | power up; point-to-point configuration; push-pull output; FLAGs as interframe time fill; double-rate data clock; clock mode 2 |

Reset SACCO-B:

| | | | |
|---|---|---|---|
| Write | CMDR | = $C1_H$ | RMC; RHR; XRES |
| Read | ISTA_B | = $10_H$ | transmit pool ready |

### 6.1.3 ELIC® CM and OCTAT-P Initialization

This part of the Application Note completes the initialization of the ELIC, and thus of the line card, by setting the Control Memory (CM) of the ELIC to handle the monitor and control time slots of two digital IOM-2 subscribers. As shown below, the OCTAT-P is set into the deactivated (idle) state:

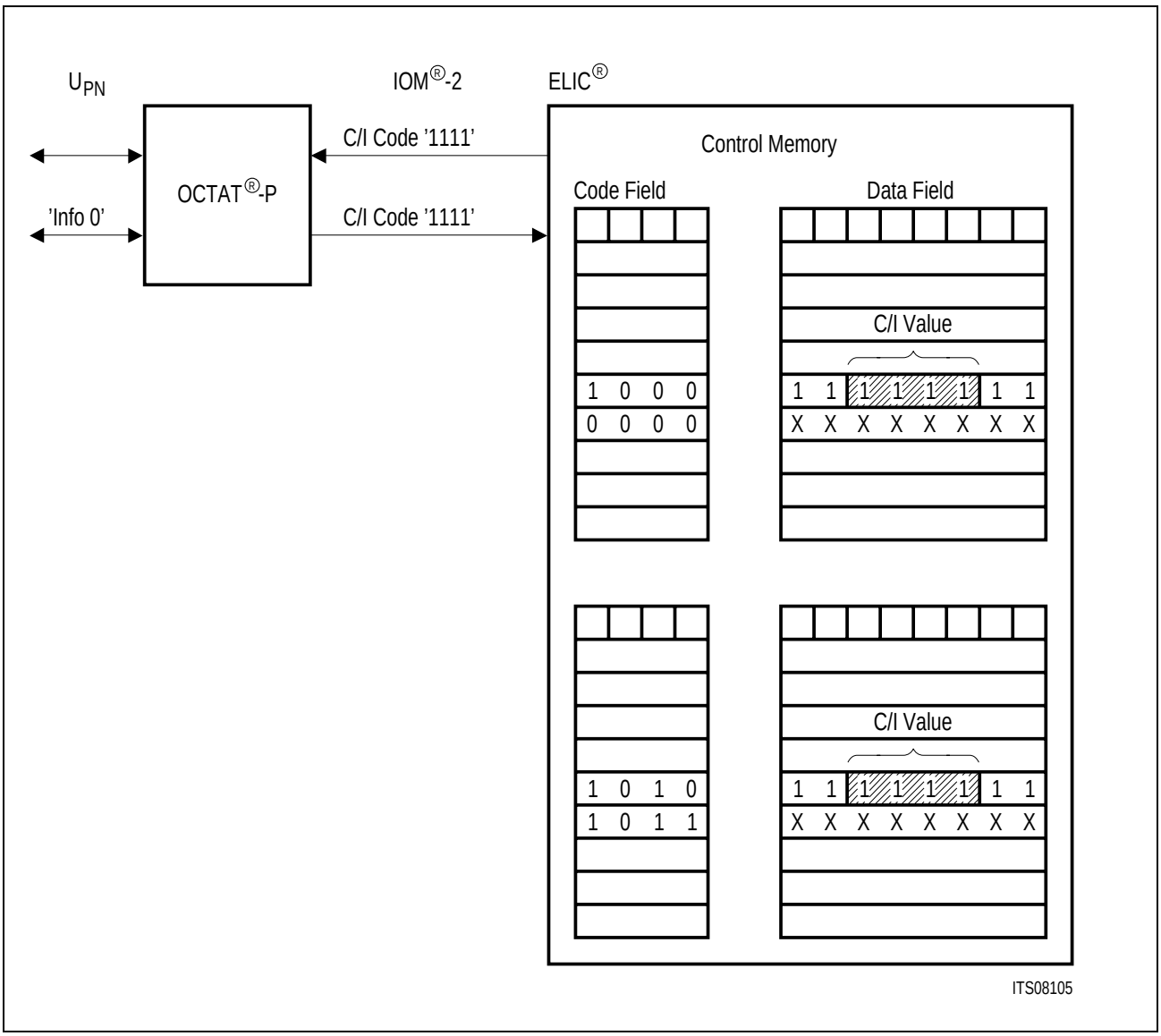



**Figure 125**
**Idle State of Line Card with ELIC® and OCTAT-P**

Note that the CM of the ELIC is not reset with a physical ELIC reset. Thus, the first step in initializing the CM is to set all addresses to ‘unassigned channel’. Next, the monitor and control time slot **pairs** are programmed for both IOM-2 channels. For a detailed description of CM handling please refer to **chapter 5.3** of the Application Hints. With the CM initialized, the CFI and PCM interfaces can be activated. Finally, the OCTAT-P is set into the deactivated (idle) state by the appropriate C/I code.

#### 6.1.3.1 Resetting the CM

| | | |
|---|---|---|
| (Write) | OMDR = $00_H$ | reset value: CM reset mode |
| Write | MADR = $FF_H$ | all CM data positions are set to $FF_H$ |
| Write | MACR = $70_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code to ‘unassigned channel’ |

#### 6.1.3.2 Initializing the CM

| | | |
|---|---|---|
| Write | OMDR = $80_H$ | set EPIC to CM init mode |

*Note: When writing to Memory Access registers the STAR_E:MFTO bit must be logical ‘0’.*

Downstream: initializing monitor and control time slot pair for IOM-2 channel 0

| | | |
|---|---|---|
| Write | MAAR = $08_H$ | downstream CM address: port 0, TS2 |
| Write | MADR = $C3_H$ | write ‘0000’ as C/I code (deactivate request) |
| Write | MACR = $7A_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for the downstream **even** address of a SACCO-A application |
| Write | MAAR = $09_H$ | downstream CM address: port 0, TS3 |
| Write | MACR = $7B_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for the downstream **odd** address of a SACCO-A application |

Downstream: initializing monitor and control time slot pair for IOM-2 channel 1

| | | |
|---|---|---|
| Write | MAAR = $18_H$ | downstream CM address: port 0, TS6 |
| Write | MADR = $C3_H$ | write ‘0000’ as C/I code (deactivate request) |
| Write | MACR = $7A_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for the downstream **even** address of a SACCO-A application |
| Write | MAAR = $19_H$ | downstream CM address: port 0, TS3 |
| Write | MACR = $7B_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for the downstream **odd** address of a SACCO-A application |

Upstream: initializing monitor and control time slot pair for IOM-2 channel 0

| | | |
|---|---|---|
| Write | MAAR = $88_H$ | upstream CM address: port 0, TS2 |
| Write | MADR = $FF_H$ | ‘1111’ expected as C/I code (deactivate indication) |
| Write | MACR = $78_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for the upstream even address of a decentral application |
| Write | MAAR = $89_H$ | upstream CM address: port 0, TS3 |
| Write | MACR = $70_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for the upstream **odd** address of a decentral application |

Upstream: initializing monitor and control time slot pair for IOM-2 channel 1

| | | | |
|---|---|---|---|
| Write | MAAR | = $98_H$ | upstream CM address: port 0, TS6 |
| Write | MADR | = $FF_H$ | '1111' expected as C/I code (deactivate indication) |
| Write | MACR | = $78_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for the upstream **even** address of a decentral application |
| Write | MAAR | = $99_H$ | upstream CM address: port 0, TS7 |
| Write | MACR | = $70_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for the upstream **odd** address of a decentral application |

#### 6.1.3.3 CFI Activation

| | | | |
|---|---|---|---|
| Write | OMDR | = $CE_H$ | change to normal op mode; set CFI outputs to open drain and activate them; enable monitor handshake |
| Read | ISTA | = $48_H$ | Interrupts: spurious C/I change due to CFI start-up; PCM sync change |
| Write | CMDR | = $10_H$ | reset C/I FIFO (ignore spurious C/I change) |
| Read | STAR | = $25_H$ | all in order and synchronized |

#### 6.1.3.4 PCM Interface Activation

| | | | |
|---|---|---|---|
| Write | MADR | = $F0_H$ | upper nibble: don't care<br>lower nibble: all bits set to high impedance |
| Write | MACR | = $68_H$ | write MADR to all tristate memory locations |
| Write | OMDR | = $EE_H$ | activate the PCM interface |

#### 6.1.3.5 Deactivating the OCTAT-P

To change the signalling for IOM-2 channel 0

| | | | |
|---|---|---|---|
| Write | MAAR | = $08_H$ | downstream CM address: port 0, TS2 |
| Write | MADR | = $FF_H$ | write '1111' as C/I code (deactivate confirmation) |
| Write | MACR | = $48_H$ | write to the CM data field |

To change the signalling for IOM-2 channel 1

| | | | |
|---|---|---|---|
| Write | MAAR | = $18_H$ | downstream CM address: port 0, TS6 |
| Write | MACR | = $48_H$ | write to the CM data field |

**The line card is now completely initialized.**

### 6.1.4 Line Activation by Subscriber A

When subscriber A takes the receiver off the hook, the layer-1 device (e.g. the ISAC-P TE) of terminal A activates the physical layer between itself and the line card.

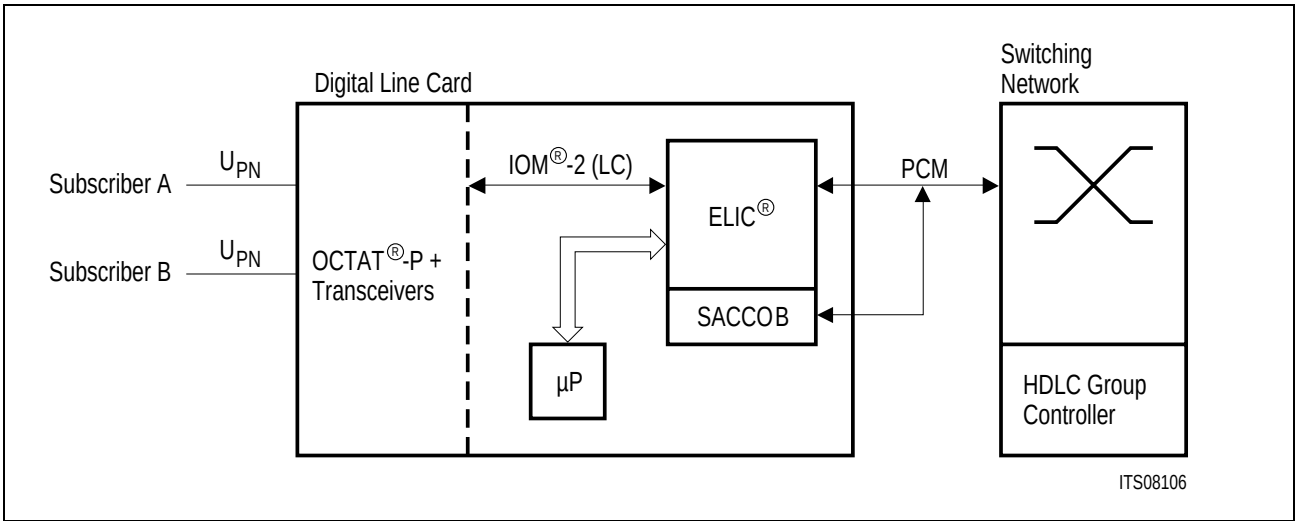



**Figure 126**
**Call-up by Subscriber A**

The ELIC reports this activation by signalling an interrupt. To allow the D-channel arbiter to monitor subscriber A, the line activation must be confirmed and the D-channel arbiter of the ELIC enabled for the subscriber. With D-channel communications between the subscriber and the ELIC established, the ETSI / E-DSS1 protocol can be followed to build up layers 2 and 3.

#### 6.1.4.1 Handling of C/I Interrupt

When subscriber A hooks OFF, the ELIC signals an interrupt:

Read ISTA_E= $40_H$ at least one valid entry in C/I FIFO
Read CIFIFO = $88_H$ change in C/I value at port 0, time slot 2 (IOM-2 channel 0)

If the µP responds slowly, the ELIC continues signalling an interrupt:

Read CIFIFO = $88_H$ second change in C/I value at port 0, time slot 2

If the µP responds slowly, the ELIC continues signalling an interrupt:

Read CIFIFO = $88_H$ third change in C/I value at port 0, time slot 2

Interrupt no longer signalled by the ELIC: when subscriber A hooks OFF, the ELIC reports three changes of the C/I code from the OCTAT-P.

Reading the current upstream C/I value of IOM-2 channel 0:

Write MAAR = $88_H$ upstream CM address where C/I value changed
Write MACR = $C8_H$ read data from the CM data field
Read MADR = $F3_H$ upstream C/I code ‘1100’ (active indication) → the subscriber of IOM-2 channel 0 has activated his line

#### 6.1.4.2 Confirmation of Line Activation

| | | | |
|---|---|---|---|
| Write | MADR | = $F3_H$ | set downstream C/I code to '1100' (active, but with the **arbitration control bit** set to 'blocked') |
| Write | MAAR | = $08_H$ | downstream CM address: port 0, TS2 |
| Write | MACR | = $48_H$ | write to the CM data field |

#### 6.1.4.3 Enabling the Arbiter

| | | | |
|---|---|---|---|
| Write | DCE0 | = $01_H$ | enable the D-channel arbiter to monitor the D-channel of IOM-2 port 0, channel 0 |

The D-channel arbiter is now resetting the downstream **arbitration control bit** of IOM-2 port 0, channel 0 to 'available' (C/I code '1000') → the ELIC is now ready to receive D-channel data from subscriber A.

#### 6.1.4.4 Build-up of Layer 2

With layer-1 communications established, terminal A initiates layer-2 build-up with an UI-frame; on receiving this UI-frame, the ELIC will signal an interrupt:

| | | | |
|---|---|---|---|
| Read | ISTA | = $02_H$ | interrupt at SACCO-A |
| Read | ISTA_A | = $80_H$ | Receive Message End (RME) interrupt |
| Read | RBCL | = $09_H$ | 9 byte in RFIFO: 8 bytes received + RSTA byte |
| Read | RFIFO | = | **UI-frame: ID Request**<br>RSTA byte= $80_H$ → frame received from IOM-2 port 0, channel 0 OK |
| Write | CMDR | = $80_H$ | reset CPU accessible portion of RFIFO |

Assigning an ID to the terminal:

| | | | |
|---|---|---|---|
| Write | XDC | = $00_H$ | direct SACCO-A transmission to IOM-2 port 0, channel 0 |
| Write | XFIFO | = | **UI-frame: ID Assignment** (8 bytes) |
| Write | CMDR | = $0A_H$ | transmit transparent frame; transmit message end |
| Read | ISTA_A | = $10_H$ | Transmit Pool Ready (XPR) |

Next, the terminal indicates that it wishes to use extended asynchronous-balanced mode. This message, as well as further D-channel communication between terminal and ELIC, is shown in abbreviated form:

| | |
|---|---|
| Received at RFIFO of SACCO-A from terminal A: | **U-frame: SABME** |
| Sent from XFIFO of SACCO-A to terminal A: | **Unnumbered Acknowledge** |

#### 6.1.4.5 Build-up of Layer 3

With layer-2 communications set up, terminal A initiates layer-3 build-up as follows:

| | |
|---|---|
| Received at RFIFO of SACCO-A from terminal A: | **I-frame: Set-up (with service indicator)** |
| Sent from XFIFO of SACCO-A to terminal A: | **I-frame: Set-up acknowledge** |
| Received at RFIFO of SACCO-A from terminal A: | **Receiver Ready (as acknowledgement)** |

### 6.1.5 Dialling the Desired Link

With layer-3 communication established, subscriber A can dial the desired link. Every digit dialled is transmitted individually from the terminal via the D-channel to the SACCO-A. If terminal A desires a link with a subscriber on another line card, the number dialled is passed via the SACCO-B to the HDLC group controller. In the current example, however, terminal A wishes to communicate with terminal B. The desired connection can thus be looped within the ELIC.

#### 6.1.5.1 Reception of Dialled Numbers at SACCO-A

| | |
|---|---|
| Received at RFIFO of SACCO-A from terminal A: | **I-frame: 1st digit dialled** |
| Sent from XFIFO of SACCO-A to terminal A: | **Receiver Ready (as acknowledgement)** |

In most PBXs, a special digit is used for outside calls. In this example, the first digit did not specify an outside call. Thus, the number of digits to follow is fixed (i.e. 3 more digits). The µP on the line card will collect these digits before deciding about passing them to the group controller.

| | |
|---|---|
| Received at RFIFO of SACCO-A from terminal A: | **I-frame: 2nd digit dialled** |
| Sent from XFIFO of SACCO-A to terminal A: | **Receiver Ready (as acknowledgement)** |
| Received at RFIFO of SACCO-A from terminal A: | **I-frame: 3rd digit dialled** |
| Sent from XFIFO of SACCO-A to terminal A: | **Receiver Ready (as acknowledgement)** |
| Received at RFIFO of SACCO-A from terminal A: | **I-frame: 4th digit dialled** |
| Sent from XFIFO of SACCO-A to terminal A: | **Receiver Ready (as acknowledgement)** |

#### 6.1.5.2 Preparing to Loop Data from Terminal A to Terminal B

With all digits received, the µP on the line card recognizes that the desired connection can be switched by looping B-channels within the ELIC. To prepare this loop between terminal A and terminal B the upstream B-channels are first switched to spare PCM time slots. As the tristate field of these PCM time slots is not changed from its initialization value (high impedance), the data is not switched to the PCM lines.

upstream: B1 time slot of IOM-2 channel 0 to PCM port 0, time slot 1

| | | |
|---|---|---|
| Write | MADR = $81_H$ | upstream connection: (to) PCM port 0, TS1 |
| Write | MAAR = $80_H$ | upstream CM address: (from) CFI port 0, time slot 0 |
| Write | MACR = $71_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for 64 kbit/s (8 bit) switching |

upstream: B1 time slot of IOM-2 channel 1 to PCM port 0, time slot 2

| | | |
|---|---|---|
| Write | MADR = $88_H$ | upstream connection: (to) PCM port 0, TS2 |
| Write | MAAR = $90_H$ | upstream CM address: (from) CFI port 0, time slot 4 |
| Write | MACR = $71_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for 64 kbit/s (8 bit) switching |

### 6.1.6 Calling up Subscriber B

Part 4 of this Application Note described the preparations for looping data from subscriber A to subscriber B. Up to this moment, however, subscriber B does not yet know that subscriber A wishes to communicate with him. In this part of the Application Note terminal B is alerted.

#### 6.1.6.1 Activating the Line to Subscriber B

First the ELIC activates that part of the OCTAT-P that connects to terminal B. The OCTAT-P activates and synchronizes the layer-1 device on terminal B before confirming the activation to the ELIC.

ELIC initiates activation of subscriber B's $U_{PN}$ line:

| | | |
|---|---|---|
| Write | MADR = $F3_H$ | set downstream C/I code to '1100' (active, blocked) |
| Write | MAAR = $18_H$ | downstream CM address: CFI port 0, time slot 6 |
| Write | MACR = $48_H$ | write data to the CM data field |

As the line becomes active, the ELIC will signal an interrupt:

| | | |
|---|---|---|
| Read | ISTA_E= $40_H$ | at least one valid entry in C/I FIFO |
| Read | CIFIFO = $98_H$ | change in C/I value at port 0, time slot 6 (IOM-2 channel 1) |

If the µP responds slowly, the ELIC will continue signalling an interrupt:

| | | |
|---|---|---|
| Read | CIFIFO = $98_H$ | second change in C/I value at port 0, time slot 6 |

If the µP responds slowly, the ELIC will continue signalling an interrupt:

| | | |
|---|---|---|
| Read | CIFIFO = $98_H$ | third change in C/I value at port 0, time slot 6 |

Interrupt no longer signalled by the ELIC: during activation of the line to subscriber B, the ELIC reports three changes of the C/I code from the OCTAT-P.

Reading the current upstream C/I value of IOM-2 channel 0:

| | | |
|---|---|---|
| Write | MAAR = $98_H$ | upstream CM address where C/I value changed |
| Write | MACR = $C8_H$ | read data from the CM data field |
| Read | MADR = $F3_H$ | upstream C/I code ‘1100’ (active indication) → confirmation that the line of IOM-2 channel 1 is active |

#### 6.1.6.2 Enabling the Arbiter

With the layer-1 link to terminal B established, the D-channel arbiter is set to monitor subscriber B in addition to subscriber A:

| | | |
|---|---|---|
| Write | DCE0 = $03_H$ | enable the D-channel arbiter to monitor the D-channels of IOM-2 port 0, channels 1 and 0 |

The D-channel arbiter is now resetting the downstream **arbitration control bit** of IOM-2 port 0, channels 0 and 1 to ‘available’ (C/I code ‘1000’) → the ELIC is now ready to receive D-channel data from subscribers A and B.

#### 6.1.6.3 Build-up of Layer 2

The ELIC now being ready to communicate to terminal B via the D-channel, layer-2 communication can be built up according to the ETSI / E-DSS1 protocol. First, the ELIC sends an UI-frame to the terminal being called up; as in **chapter 6.1.4.4**, the D-channel communication that follows is shown in abbreviated form:

Assigning an ID to the terminal:

| | | |
|---|---|---|
| Write | XDC = $01_H$ | direct SACCO-A transmission to IOM-2 port 0, channel 1 |
| Write | XFIFO = | **UI frame: Set_up** (with service indicator) |
| Write | CMDR = $0A_H$ | transmit transparent frame; transmit message end |
| Read | ISTA_A= $10_H$ | Transmit Pool Ready (XPR) |

| | |
|---|---|
| Received at RFIFO of SACCO-A from terminal B: | **UI-frame: ID_Request** |
| Sent from XFIFO of SACCO-A to terminal B: | **UI-frame: ID_Assigned** |
| Received at RFIFO of SACCO-A from terminal B: | **U-frame: SABME** |
| Sent from XFIFO of SACCO-A to terminal B: | **Unnumbered Acknowledge** |

#### 6.1.6.4 Build-up of Layer 3

With layer-2 communications set up, terminal B initiates layer-3 build-up as follows:

Received at RFIFO of SACCO-A from terminal B: **I-frame: Alerting**
Sent from XFIFO of SACCO-A to terminal B: **Receiver Ready (as acknowledgement)**
→ **Phone B Rings !**

Subscriber A is now informed that subscriber B is being alerted:

Write XDC = $00_H$ direct SACCO-A transmission to IOM-2 port 0, channel 0
Sent from XFIFO of SACCO-A to terminal A: **I-frame: Alerting**
Received at RFIFO of SACCO-A from terminal A: **Receiver Ready (as acknowledgement)**

Terminal A will now provide subscriber A with a tone that signals to subscriber A that subscriber B is being alerted.

### 6.1.7 Completing the Call

When subscriber B answers the call, terminal indicates ‘hook-off’ to the ELIC via a D-channel message. The data loop that was prepared in **chapter 6.1.5.2** can now be closed. Finally, the hook-off information is acknowledged to terminal B and passed to terminal A. This indicates to the terminals that their subscribers can start communication.

#### 6.1.7.1 Receiving the Hook-off Information at the ELIC®

When subscriber B answers the call, the ELIC will signal a RME interrupt:

Received at RFIFO of SACCO-A from terminal B: **I-frame: Connected**

Write XDC = $01_H$ direct SACCO-A transmission to IOM-2 port 0, channel 1
Sent from XFIFO of SACCO-A to terminal B: **Receiver Ready**

### 6.1.7.2 Closing the Data Loop

With both subscribers ready at their terminals, the data loop between them can be closed:

downstream: PCM port 0, time slot 2 to the B1 time slot of IOM-2 channel 0

| | | |
|---|---|---|
| Write | MADR = $08_H$ | downstream connection: (from) PCM port 0, TS2 |
| Write | MAAR = $00_H$ | downstream CM address: (to) CFI port 0, time slot 0 |
| Write | MACR = $71_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for 64 kbit/s (8 bit) switching |

downstream: PCM port 0, time slot 1 to the B1 time slot of IOM-2 channel 1

| | | |
|---|---|---|
| Write | MADR = $01_H$ | downstream connection: (from) PCM port 0, TS1 |
| Write | MAAR = $10_H$ | downstream CM address: (to) CFI port 0, time slot 4 |
| Write | MACR = $71_H$ | upper nibble: write CM data and code<br>lower nibble: set CM code for 64 kbit/s (8 bit) switching |

### 6.1.7.3 Giving both Terminals the ‘Go-Ahead’ to Transceive Data

Finally, the ELIC informs both terminals that the connection has been made. This acts as the go-ahead to pass their subscriber’s data, via the ELIC, to the other subscriber:

| | |
|---|---|
| Sent from XFIFO of SACCO-A to terminal B: | **I-frame: Connect Acknowledgement** |
| Received at RFIFO of SACCO-A from terminal B: | **Receiver Ready (as acknowledgement)** |

Write XDC = $00_H$ direct SACCO-A transmission to IOM-2 port 0, channel 0

| | |
|---|---|
| Sent from XFIFO of SACCO-A to terminal A: | **I-frame: Connected** |
| Received at RFIFO of SACCO-A from terminal A: | **Receiver Ready (as acknowledgement)** |

**The connection has been established, and subscribers A and B can now communicate.**

## 6.2 D-Channel Delay Due to Arbitration

When using D-channel arbitration, the ELIC notifies subscribers when the SACCO-A is available to receive D-channel data. As several subscribers may start sending data concurrently, the ELIC arbitrates among the subscribers: one subscriber is permitted to continue sending data, while the other subscribers are blocked until the SACCO-A has completed reception of the data of the selected subscriber.

When blocked, subscribers have to wait until the SACCO-A becomes available for accepting their data. How long subscribers have to wait depends on the number of subscribers who wish to send data, as well as on the average number of bytes that subscribers wish to send in their HDLC frame.

The subsequent investigation details the delay parameters and gives average delay times for typical $S_0$ and $U_{PN}$ applications.

### Theoretical Derivation

Using the ‘limited selection’ state, the ELIC effectively arbitrates among subscribers according to a token ring protocol. Let ‘m – 1’ be the number of subscribers already wishing to send data to the ELIC when an ‘m-th’ subscriber enters the arbitration. The length of time that the subscriber has to wait will then depend on the position of the token in the ring.

At best, the subscriber enters the arbitration just as it is passed the token. That is, the subscriber starts to send his data just as the ELIC becomes ready to listen to it. Regardless of the number of competing subscribers, the subscriber will then encounter no delay. The minimum delay time is thus always 0 milliseconds.

At worst, the subscriber enters the arbitration just after the token has passed. The subscriber will then have to wait for all ‘m – 1’ competing subscribers to send their data before his own data is accepted.

Due to the linearity of token-bus statistics, the average subscriber will thus have to wait for (m – 1)/2 subscribers to send their data before his own data is accepted by the ELIC.

Now, to determine the total time a subscriber will have to wait, let ‘x’ be the average length of the HDLC frame (including address and control bytes) that subscribers wish to send. Including the opening and the closing flag as well as the CRC word, the HDLC message thus extends over x + 4 HDLC bytes.

Additionally, the HDLC protocol inserts a ‘0’ bit after 5 consecutive ‘1’ bits. Of course, the precise number of inserted ‘0’ bits will depend on the content of the HDLC frame. A conservative estimate will allow for one ‘0’ bit inserted into every second HDLC byte. With every second HDLC byte extended by $^1/_8$ byte, $^1/_{16}$ of a byte must be added to every HDLC byte. Similarily, for the two-byte CRC sum $^1/_8$ of a byte is added. Including inserted ‘0’ bits, an HDLC message thus extends over $^{17}/_{16}x + {}^{33}/_8$ byte.

2 HDLC bits are sent in every IOM-2 frame. The average message thus requires

$4 \times ({}^{17}/_{16}x + {}^{33}/_{8})$, or $({}^{17}/_{4}x + {}^{33}/_{2})$ IOM-2 frames for transmission.

Since every IOM-2 frame takes 125 µs, the formula for the average waiting time is:

$t = 125\ \mu s \times {}^{1}/_{2}(m-1)[({}^{17}/_{4}x + {}^{33}/_{2}) + y]$

$= (m-1)[265.625\ \mu s \times x + 62.5\ \mu s \times (16.5 + y)]$

In this formula, the parameter ‘y’ allows for the time it takes subscribers to respond to the availablilty of the SACCO-A. As exemplified by the basic rate application that follows, the ‘y’ parameter depends on application specific features such as double-last-look logic, collision detection and interface delays.

### $S_0$ Application Example

The following figure shows a typical $S_0$ application of the ELIC:

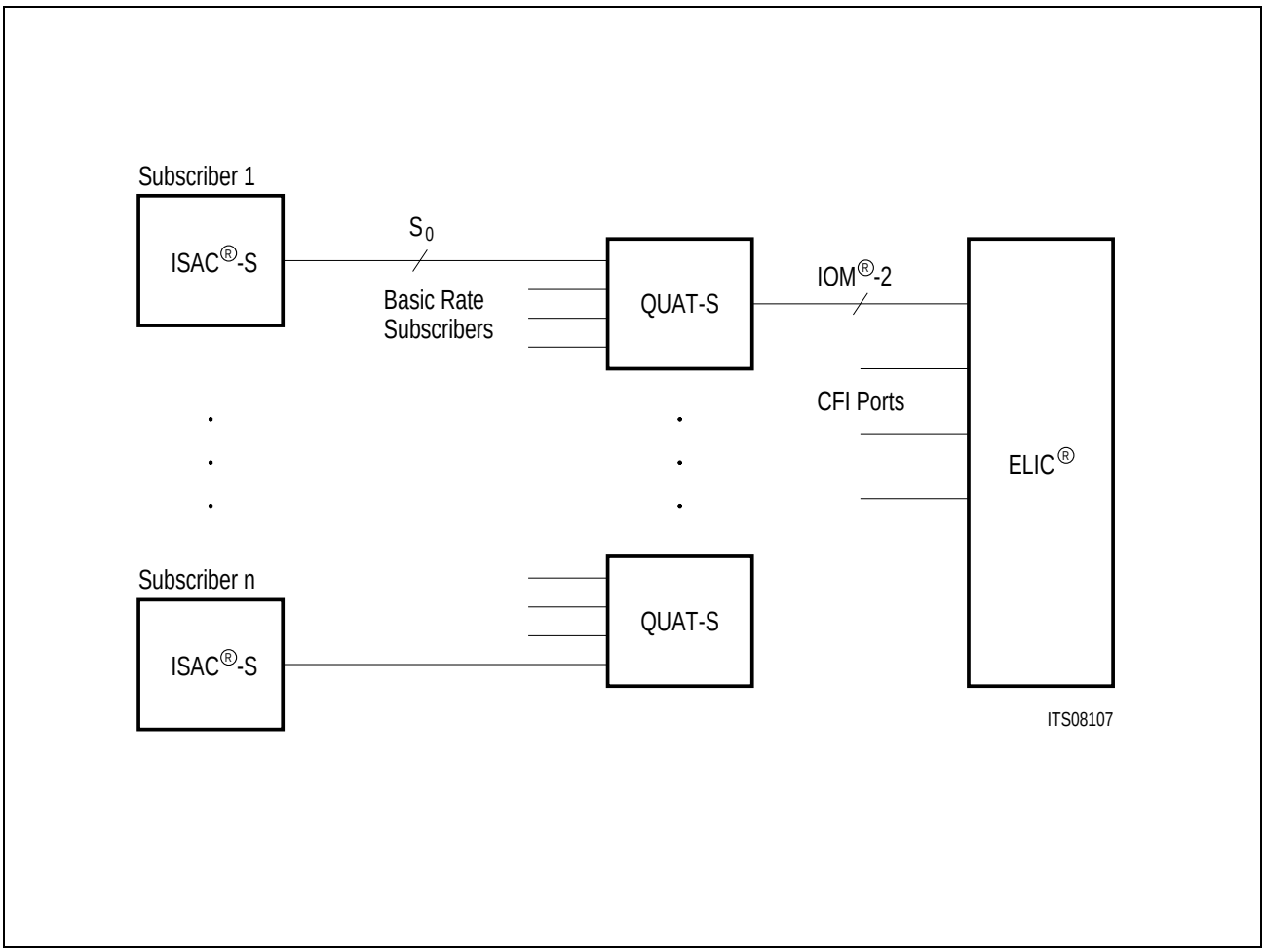



**Figure 127**
**Basic Rate Application of ELIC®**

To derive the response delay ‘y’ of this architecture, note that the double-last-look logic of the QUAT-S delays recognition of the ‘available’ C/I code by one frame. Another

(max.) 6 frames delay are due to interface delays and to the ISAC-S waiting for 8 non-inverted E-bits prior to sending data. This is shown in **figure 128** below:

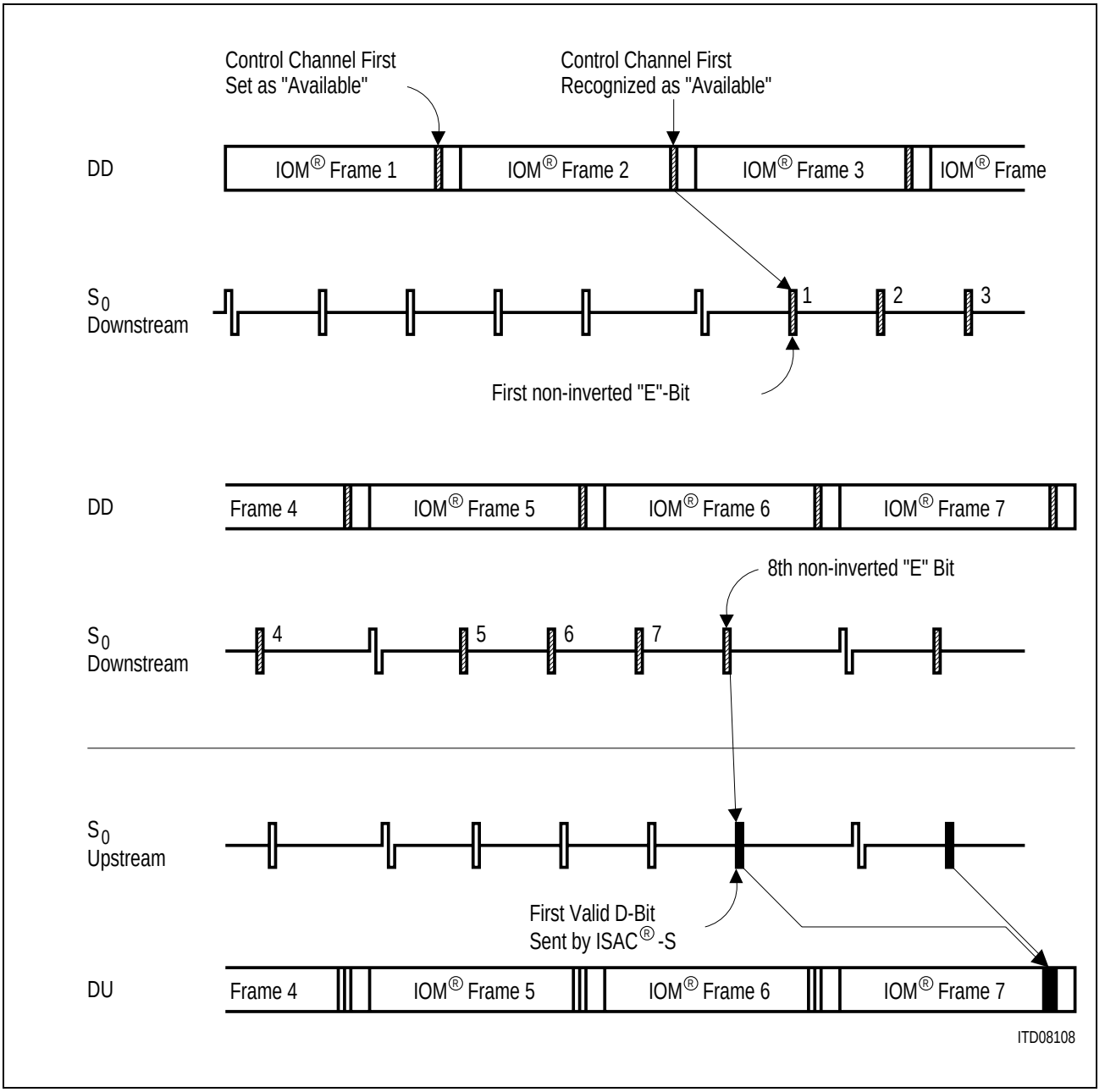



**Figure 128**
**Response Delay of typical $S_0$ Applications according to Figure 127**

The system-specific response delay of the $S_0$ architecture shown in **figure 127** is thus $y = 1 + 6 = 7$.

The delay time experienced by the average $S_0$ subscriber is then:

$t = 125\ \mu s \times {}^{1}/_{2}(m - 1)[({}^{17}/_{4}x + {}^{33}/_{2}) + 7]$
$= (m - 1)[265.625\ \mu s \times x + 1468.75\ \mu s]$

The following **table 52** gives these delay times for a variety of competing subscribers and HDLC frame lengths:

**Table 52**
**Average Delay Times for the $S_0$ Application of Figure 127 (in ms)**

| Average Length of HDLC Frames (in Byte) | Number of Subscribers | | | | | | |
|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **5** | **10** | **20** | **32** |
| 5 | 0.00 | 2.80 | 5.59 | 11.19 | 25.17 | 53.14 | 86.70 |
| 10 | 0.00 | 4.13 | 8.25 | 16.50 | 37.13 | 78.38 | 127.88 |
| 20 | 0.00 | 6.78 | 13.56 | 27.13 | 61.03 | 128.84 | 210.22 |
| 40 | 0.00 | 12.09 | 24.19 | 48.38 | 108.84 | 229.78 | 374.91 |
| 80 | 0.00 | 22.72 | 45.44 | 90.88 | 204.47 | 431.66 | 704.28 |
| 200 | 0.00 | 54.59 | 109.19 | 218.38 | 491.34 | 1037.28 | 1692.41 |
| 500 | 0.00 | 134.28 | 268.56 | 537.13 | 1208.53 | 2551.34 | 4162.72 |
| 1000 | 0.00 | 267.09 | 534.19 | 1068.38 | 2403.84 | 5074.78 | 8279.91 |
| 2000 | 0.00 | 532.72 | 1065.44 | 2130.88 | 4794.47 | 10121.7 | 16514.3 |

As has been shown theoretically, the maximum (worst case) delay time is twice the average delay time for any set of subscribers and frame lengths, whereas the minimum delay time is always 0 µs. The following example shows how to interpret **table 52**.

**Example:**

For the system of **figure 127**,
let n = 32 (total number of enabled subscribers),
let the average HDLC frame length = 20 Byte.

Assume that, during peak traffic times (e.g 10 a.m.), an average of 10 subscribers wishes to send data to the SACCO-A concurrently. Then the average delay for access to the SACCO-A will be 61.03 ms. The worst case delay will be 122.06 ms, whereas the minimum delay will be 0 ms.

Assume that, during times of little demand (e.g 10 p.m.), an average of 2 subscribers wishes to send data to the SACCO-A concurrently. Then the average delay for access to the SACCO-A will be 6.78 ms. The worst case delay will be 13.56 ms, whereas the minimum delay will be 0 ms.

Average waiting times in the example system will thus vary between 61.03 ms and 6.78 ms.

## $U_{PN}$ Application Example

The following **figure 129** shows a typical $U_{PN}$ application of the ELIC:

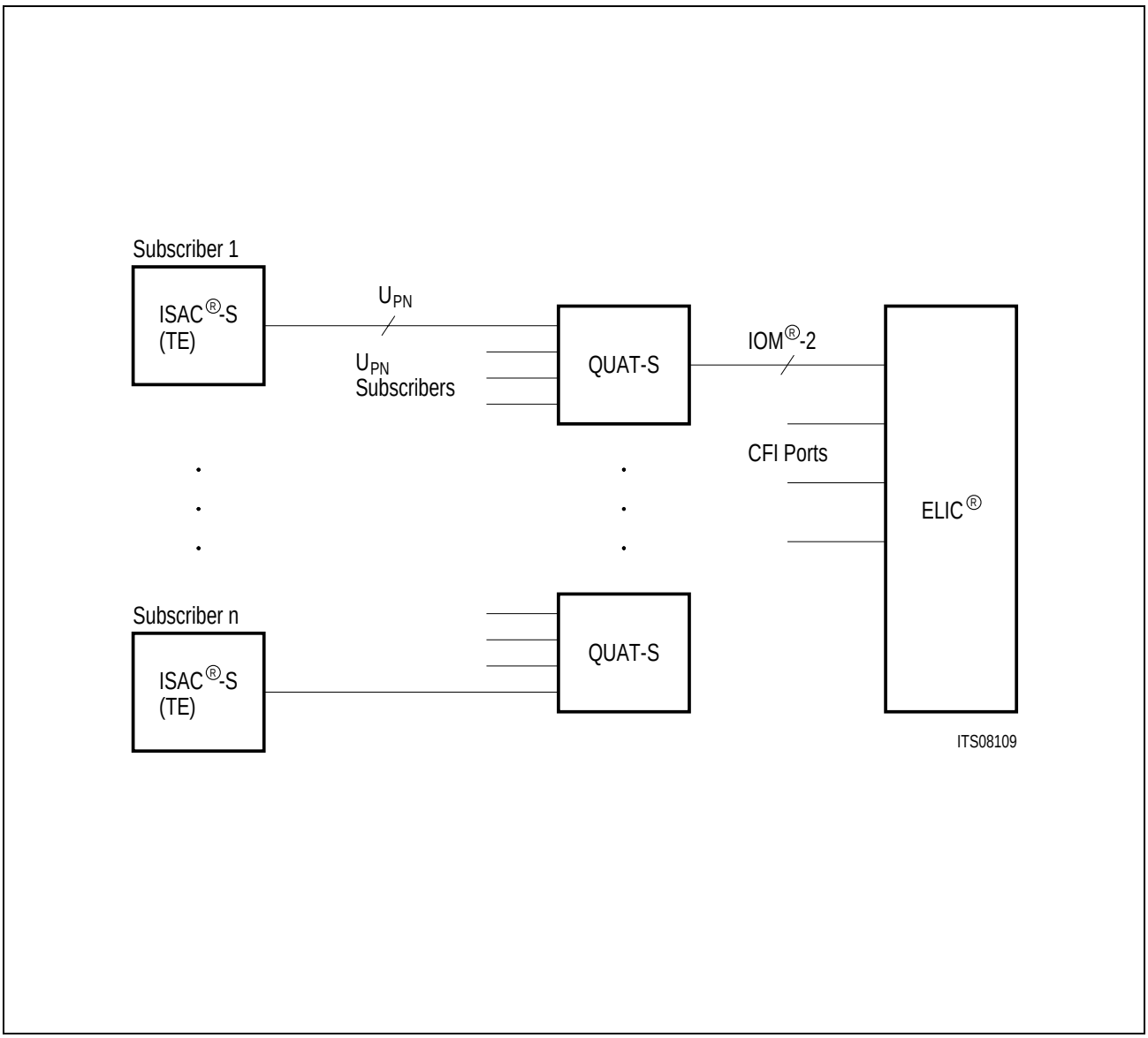



**Figure 129**
**$U_{PN}$ Application of ELIC®**

From **chapter 2.2.8.3**. of the ELIC Technical Manual, the average response delay of this $U_{PN}$ application is found to be y = 6 frames. Thus

$t = (m - 1)[265.625\ \mu s \times x + 1406.25\ \mu s]$,

and the average waiting time is

**Table 53**
**Average Delay Times for the $U_{PN}$ Application of Figure 129 (in ms)**

| Average Length of HDLC Frames (in Byte) | Number of Subscribers | | | | | | |
|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **5** | **10** | **20** | **32** |
| 5 | 0.00 | 2.73 | 5.47 | 10.94 | 24.61 | 51.95 | 84.77 |
| 10 | 0.00 | 4.06 | 8.13 | 16.25 | 36.56 | 77.19 | 125.94 |
| 20 | 0.00 | 6.72 | 13.44 | 26.88 | 60.47 | 127.66 | 208.28 |
| 40 | 0.00 | 12.03 | 24.06 | 48.13 | 108.28 | 228.59 | 372.97 |
| 80 | 0.00 | 22.66 | 45.31 | 90.63 | 203.91 | 430.47 | 702.34 |
| 200 | 0.00 | 54.53 | 109.06 | 218.13 | 490.78 | 1036.09 | 1690.47 |
| 500 | 0.00 | 134.22 | 268.44 | 536.88 | 1207.97 | 2550.16 | 4160.78 |
| 1000 | 0.00 | 267.03 | 534.06 | 1068.13 | 2403.28 | 5073.59 | 8277.97 |
| 2000 | 0.00 | 532.66 | 1065.31 | 2130.63 | 4793.91 | 10120.5 | 16512.3 |

In closing, note that the only element of the delay time formula to change between varying systems is the system-specific response delay ‘y’. This ‘y’-parameter, however, affects the actual average delay time ‘t’ only slightly. The delay times of **table 52** thus differ little from those of **table 52**. Indeed, unless their response delays differ drastically from those above, most architectures will find their average delay times well approximated by **tables 52 and 52**.

### 6.3 Behaviour of the SACCO-A when a RFIFO Overflow Occurs

When using the ELIC SACCO-A HDLC controller in conjunction with the D-channel arbiter there might be a critical situation when a RFIFO overflow occurs.

The situation can be managed by software solution.

The following text describes this situation and advices how to manage it. Please refer also to **page 82f**.

**Precondition for the Critical Situation:**

The ELIC D-channel arbiter is activated to serve all D-channels (register AMO:CCHM = 1, refer to **chapter 4.8.1**), that have been enabled in the DCE3..0 registers (refer to **page 188f**).

The ELIC SACCO-A receives a frame of one of the enabled D-channels, although there is not enough space for the whole frame in its RFIFO. A previously generated RME interrupt has not yet been served.

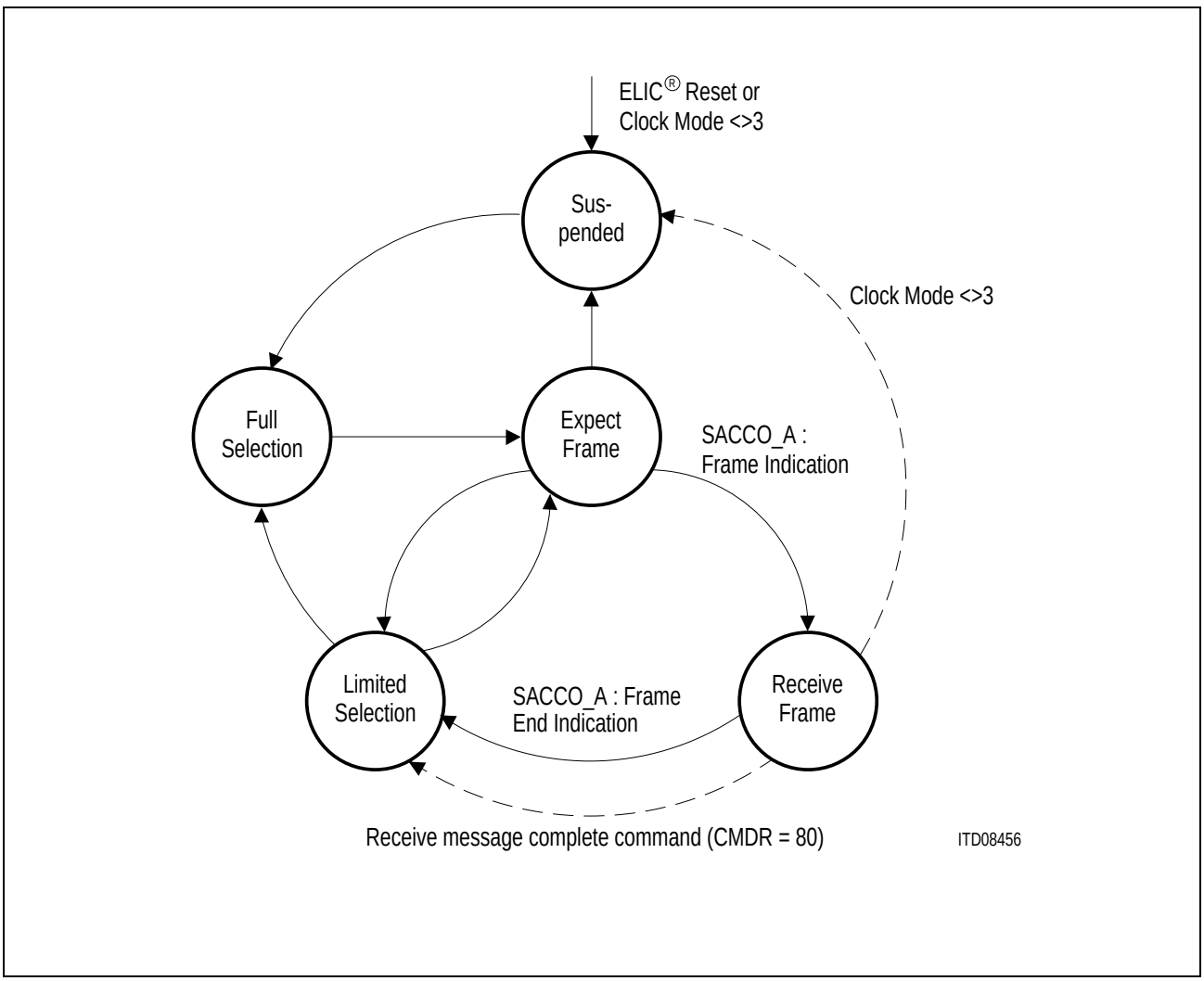



**Figure 130**

**Problem Description:**

Assuming the arbiter is in the state FULL SELECTION, it will change to the state EXPECT FRAME and then RECEIVE FRAME, as soon as the opening flag of the HDLC frame (frame indication) is detected.

The SACCO_A will then start to copy the received data to the RFIFO. Usually (if there is enough space for the whole frame + 1 byte) the arbiter would abandon this state at frame end and enter the state LIMITED SELECTION.

In the case, where there is not enough space for the whole frame + 1 byte, the arbiter will not get the frame end indication and the D-channel arbiter stays in the state RECEIVE FRAME.

**Explanation:**

The reason for this behaviour is, that the frame end indication is send to the arbiter as soon as the receive status byte (RSTA) is written to the FIFO. So if there is no space for the RSTA byte in the RFIFO, the arbiter will not receive a frame end indication.

**Resulting Behaviour:**

Sucessive frames will not be rejected (no blocking information is being send), but lead to a Receive Frame Overflow interrupt of the ELIC.

This behaviour makes the sending HDLC controller believe, it can continue in sending new frames, whereas all other channels still get the blocked information.

**How to Manage the Situation:**

In order to stop the HDLC controller from transmitting and give the other HDLC controllers a chance to be arbitrated, the arbiter state RECEIVE FRAME must be abandoned, as soon as the ELIC indicates this situation (e.g. Receive Frame Overflow interrupt).

This can be achieved by 2 possibilities:

1. Switching the clock mode (CCR1:CM1..0) unequal 3. The result is a change to the arbiter state SUSPENDED
2. Sending a Receive Message Complete command (CMDR = 80) to the SACCO_A. This generates internally a Frame End and the arbiter changes to the state LIMITED SELECTION.

*Note: If the command RESET HDLC receiver (CMDR:RHR = 1) is performed, the arbiter is not reset and stays in the state RECEIVE FRAME until a new frame has been sent, or the clock mode is changed, as described before.*

# 7 Electrical Characteristics

## Absolute Maximum Ratings

| Parameter | Symbol | Limit Values | | Unit |
|---|---|---|---|---|
| | | min. | max. | |
| Ambient temperature under bias: PEB<br>PEF | $T_A$<br>$T_A$ | 0<br>– 40 | 70<br>85 | °C<br>°C |
| Storage temperature | $T_{stg}$ | – 65 | 125 | °C |
| Voltage on any pin with respect to ground | $V_S$ | – 0.4 | $V_{DD}$ + 0.4 | V |
| Maximum voltage on any pin | $V_{max}$ | | 6 | V |

*Note: Stresses above those listed here may cause permanent damage to the device. Exposure to absolute maximum ratings conditions for extended periods may affect device reliability.*
*Maximum ratings are absolute ratings; exceeding only one of these values may cause irreversible damage to the integrated circuit.*

## Characteristics

PEB: $T_A$ = 0 to 70 °C; $V_{DD}$ = 5 V ± 5 %, $V_{SS}$ = 0 V
PEF: $T_A$ = – 40 to 85 °C; $V_{DD}$ = 5 V ± 5 %, $V_{SS}$ = 0 V

| Parameter | Symbol | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|
| | | min. | max. | | |
| L-input voltage | $V_{IL}$ | – 0.4 | 0.8 | V | |
| H-input voltage | $V_{IH}$ | 2.0 | $V_{DD}$ + 0.4 | | |
| L-output voltage | $V_{OL}$ | | 0.45 | V | $I_{OL}$ = 7 mA (pins TxDA, TxDB, TxD0-3, DU0-3, DD0-3)<br>$I_{OL}$ = 2 mA (all other) |
| H-output voltage | $V_{OH}$ | 2.4 | | V | $I_{OH}$ = – 400 µA |
| H-output voltage | $V_{OH}$ | $V_{DD}$ – 0.5 | | V | $I_{OH}$ = – 100 µA |

**Characteristics** (cont'd)
PEB: $T_A$ = 0 to 70 °C; $V_{DD}$ = 5 V ± 5 %, $V_{SS}$ = 0 V
PEF: $T_A$ = – 40 to 85 °C; $V_{DD}$ = 5 V ± 5 %, $V_{SS}$ = 0 V

| Parameter | | Symbol | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|---|
| | | | min. | max. | | |
| Power supply current | operational | $I_{CC}$ | | 15 | mA | $V_{DD}$ = 5 V, PDC = 8 MHz, HDCA/B = 4 MHz |
| | power down | | | 3 | mA | input at 0 V/$V_{DD}$, no output loads |
| Input leakage current | | $I_{LI}$ | | 1 | µA | 0 V < $V_{IN}$ < $V_{DD}$ to 0 V |
| Output leakage current | | $I_{LO}$ | | | | 0 V < $V_{OUT}$ < $V_{DD}$ to 0 V |

*Note: The listed characteristics are ensured over the operating range of the integrated circuit. Typical characteristics specify mean values expected over the production spread. If not otherwise specified, typical characteristics apply at $T_A$ = 25°C and the given supply voltage.*

### Capacitances

$T_A$ = 25 °C; $V_{DD}$ = 5 V ± 5 %, $V_{SS}$ = 0 V, $f_C$ = 1 MHz, unmeasured pins returned to $V_{SS}$.

| Parameter | Symbol | Limit Values | | Unit |
|---|---|---|---|---|
| | | min. | max. | |
| Input capacitance, $f_C$ = 1 MHz | $C_{IN}$ | 5 | 10 | pF |
| Output capacitance | $C_{OUT}$ | 8 | 15 | pF |
| I/O | $C_{I/O}$ | 10 | 20 | pF |

### AC-Characteristics

Ambient temperature under bias range, $V_{DD}$ = 5 V ± 5 %.

Inputs are driven to 2.4 V for a logical ’1’ and to 0.4 V for a logical ’0’.
Timing measurements are made at 2.0 V for a logical ’1’ and at 0.8 V for a logical ’0’.
The AC-testing input/output wave forms are shown below.

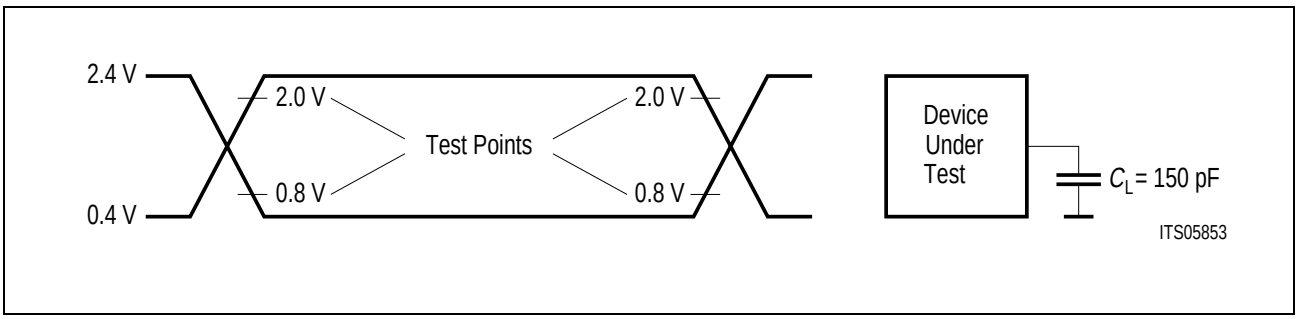



**Figure 131**
**I/O-Wave Form for AC-Test**

## Bus Interface Timing

| Parameter | Symbol | Limit Values | | Unit |
|---|---|---|---|---|
| | | min. | max. | |
| IR or $\overline{W}$ set-up to $\overline{DS}$ | $t_{DSD}$ | 0 | | ns |
| $\overline{RD}$-pulse width | $t_{RR}$ | 80 | | ns |
| $\overline{RD}$-control interval | $t_{RI}$ | 40 | | ns |
| Port data set-up time to $\overline{RD}$x$\overline{CS}$ | $t_{PR}$ | 30 | | ns |
| Port data hold time from $\overline{RD}$x$\overline{CS}$ | $t_{RP}$ | 30 | | ns |
| Data output delay from $\overline{RD}$ | $t_{RD}$ | | 80 | ns |
| Data float delay from $\overline{RD}$ | $t_{DF}$ | | 25 | ns |
| DMA-request delay | $t_{DRH}$ | | 65 | ns |
| $\overline{WR}$-pulse width | $t_{WW}$ | 45 | | ns |
| $\overline{WR}$-control interval | $t_{WI}$ | 40 | | ns |
| Data set-up time to $\overline{WR}$x$\overline{CS}$, $\overline{DS}$x$\overline{CS}$ | $t_{DW}$ | 30 | | ns |
| Data hold time from $\overline{WR}$x$\overline{CS}$, $\overline{DS}$x$\overline{CS}$ | $t_{WD}$ | 15 | | ns |
| Port data delay from $\overline{WR}$x$\overline{CS}$ | $t_{WP}$ | | 100 | ns |
| ALE-pulse width | $t_{AA}$ | 30 | | ns |
| Address set-up time to ALE | $t_{AL}$ | 10 | | ns |
| Address hold time from ALE | $t_{LA}$ | 15 | | ns |
| ALE set-up time to $\overline{WR}$, $\overline{RD}$ | $t_{ALS}$ | 8 | | ns |
| Address set-up time to $\overline{WR}$, $\overline{RD}$ | $t_{AS}$ | 10 | | ns |
| Address hold time from $\overline{WR}$ | $t_{AH}$ | 0 | | ns |

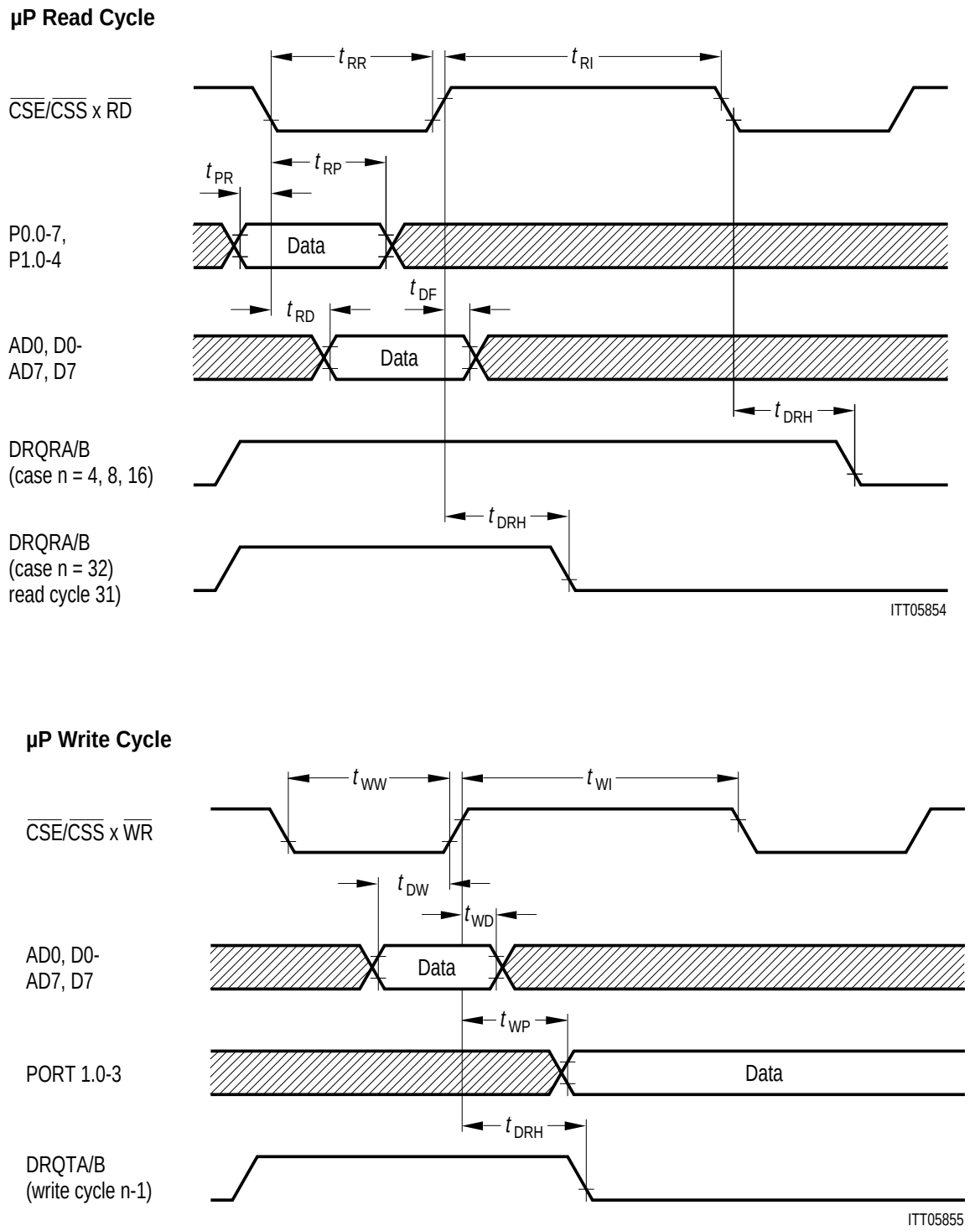



**Figure 132 a**
**Siemens/Intel Bus Mode**

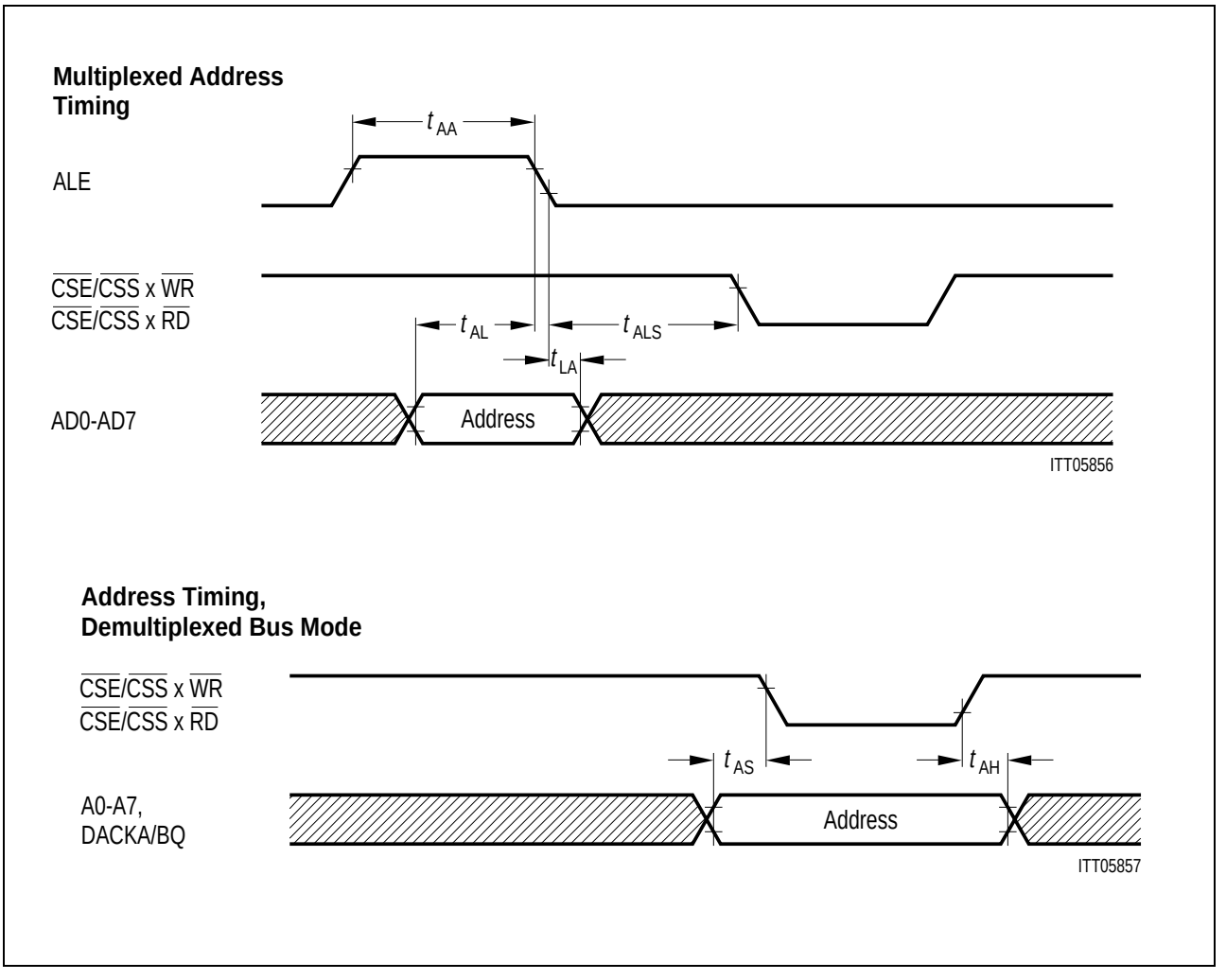
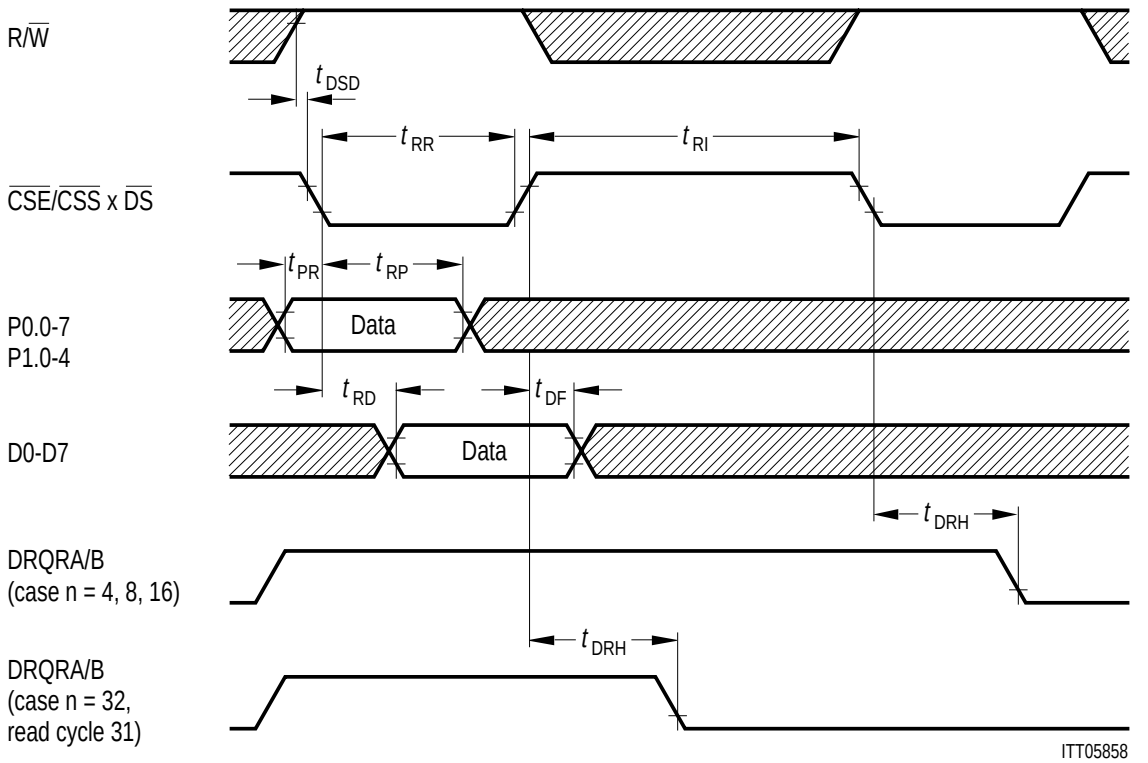



**Figure 132 b**
**Siemens/Intel Bus Mode**




**µP Write Cycle**

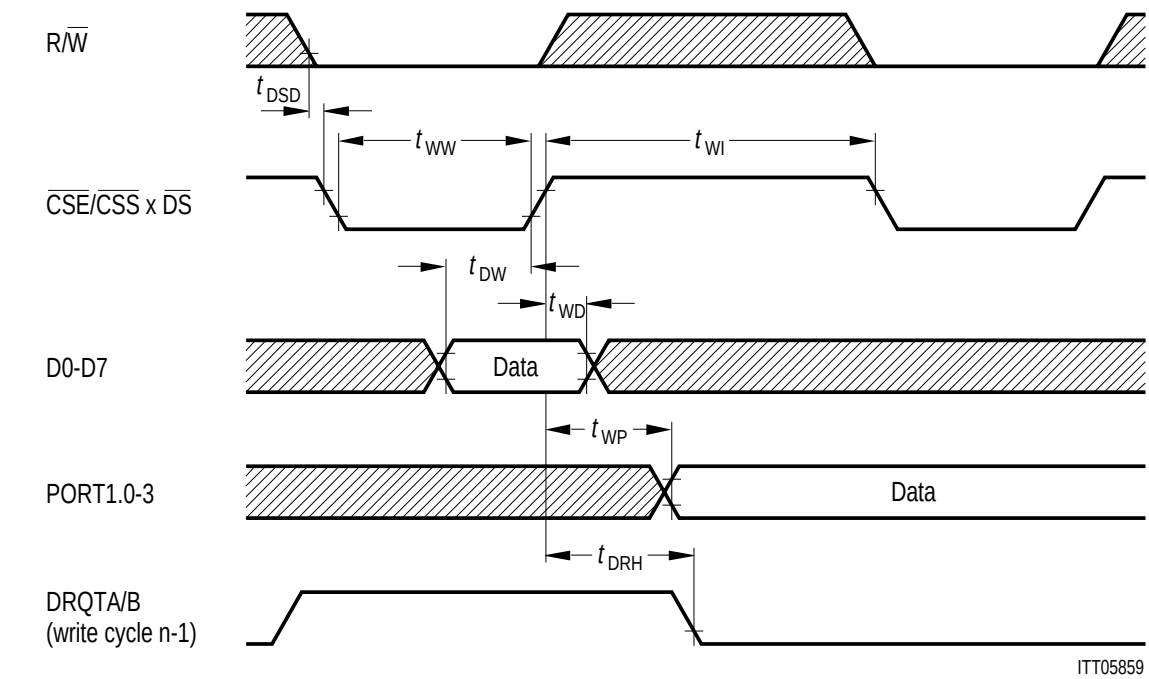



**Figure 133 a**
**Motorola Bus Mode**

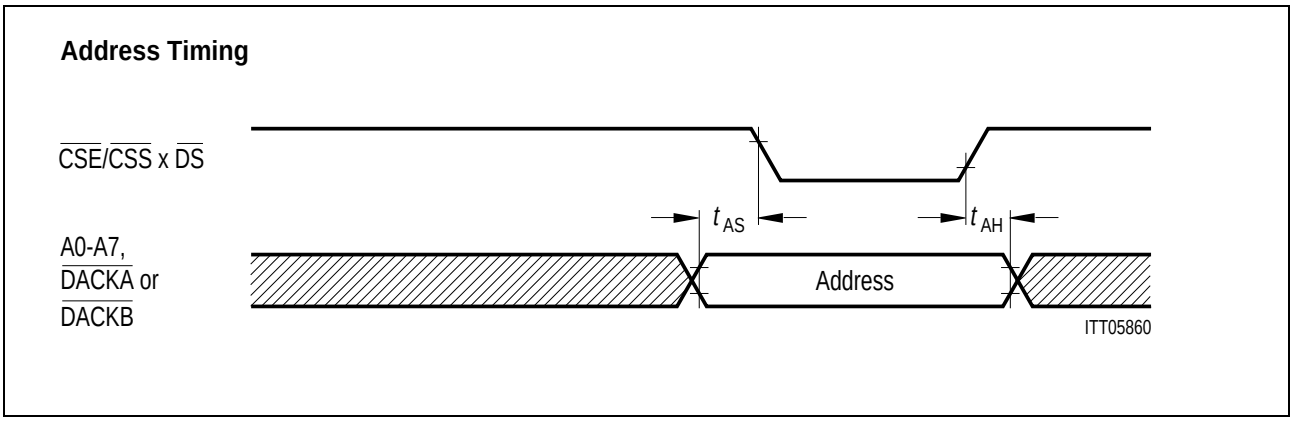



**Figure 133 b**
**Motorola Bus Mode**

## Interrupt Timing

| Parameter | Symbol | Limit Values | | Unit |
|---|---|---|---|---|
| | | min. | max. | |
| Interrupt activation delay | $t_{ID}$ | | 100 | ns |
| Interrupt inactivation delay | $t_{IID}$ | | 120 | ns |

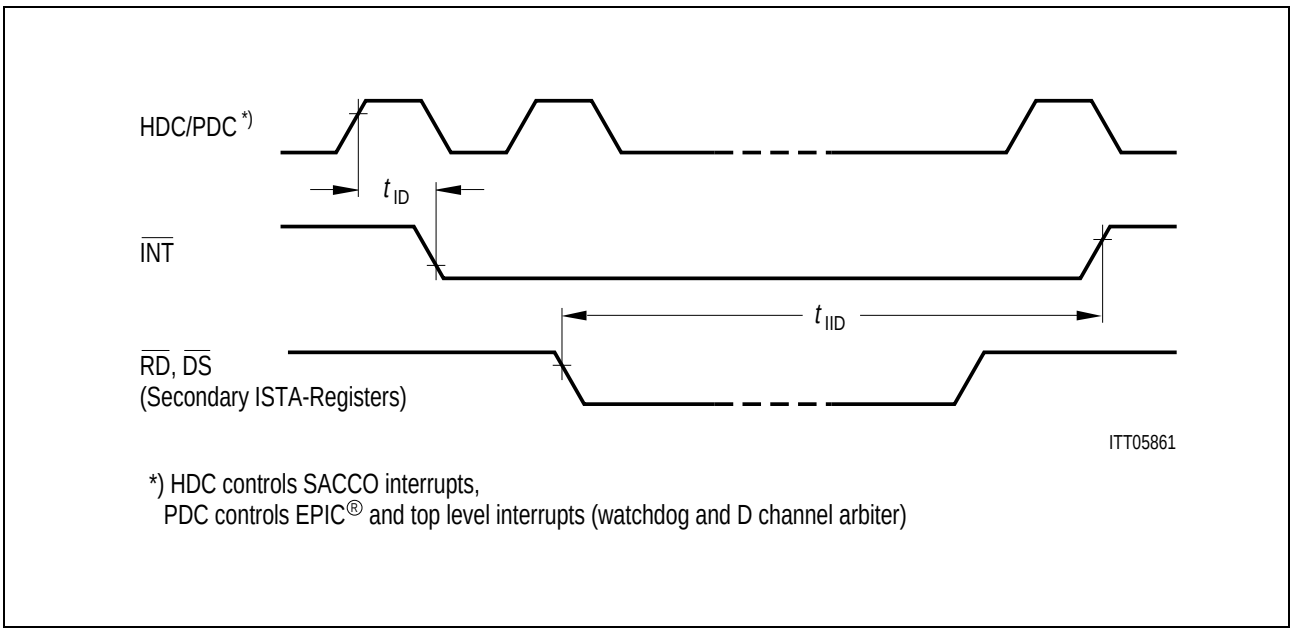



**Figure 134**
**Interrupt Timing**

## Reset Timing

| Parameter | Symbol | Limit Values | | Unit |
|---|---|---|---|---|
| | | min. | max. | |
| RESEX-spike pulse width | $t_{RESP}$ | | 5 | ns |
| RESEX-pulse | $t_{REPW}$ | 1200 | | ns |
| RESIN-activation delay | $t_{RAD}$ | | 50 | ns |
| RESIN-deactivation delay | $t_{RDD}$ | | 50 | ns |

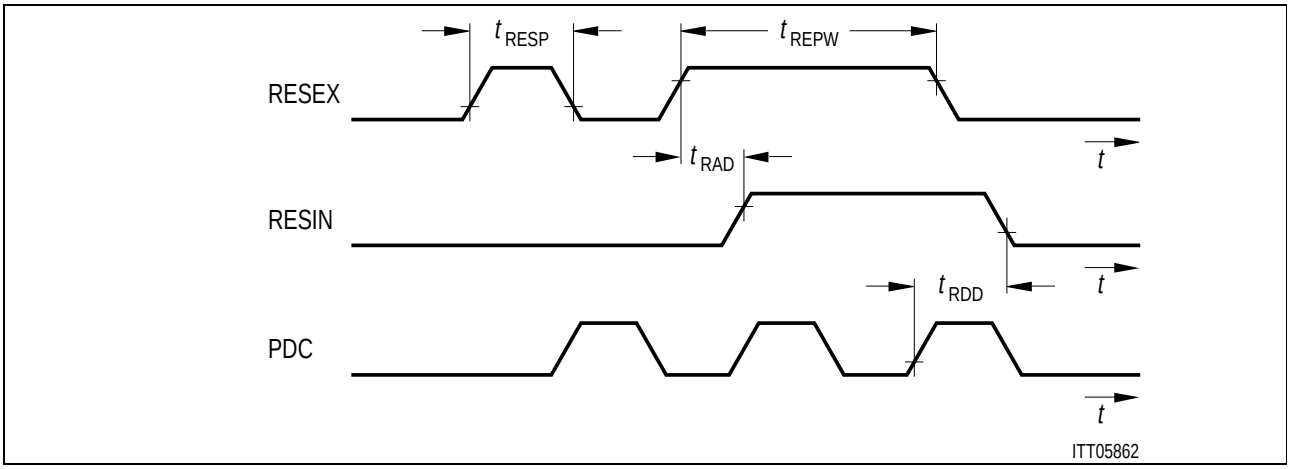



**Figure 135**
**RESEX/RESIN**

## Power-up Reset Timing

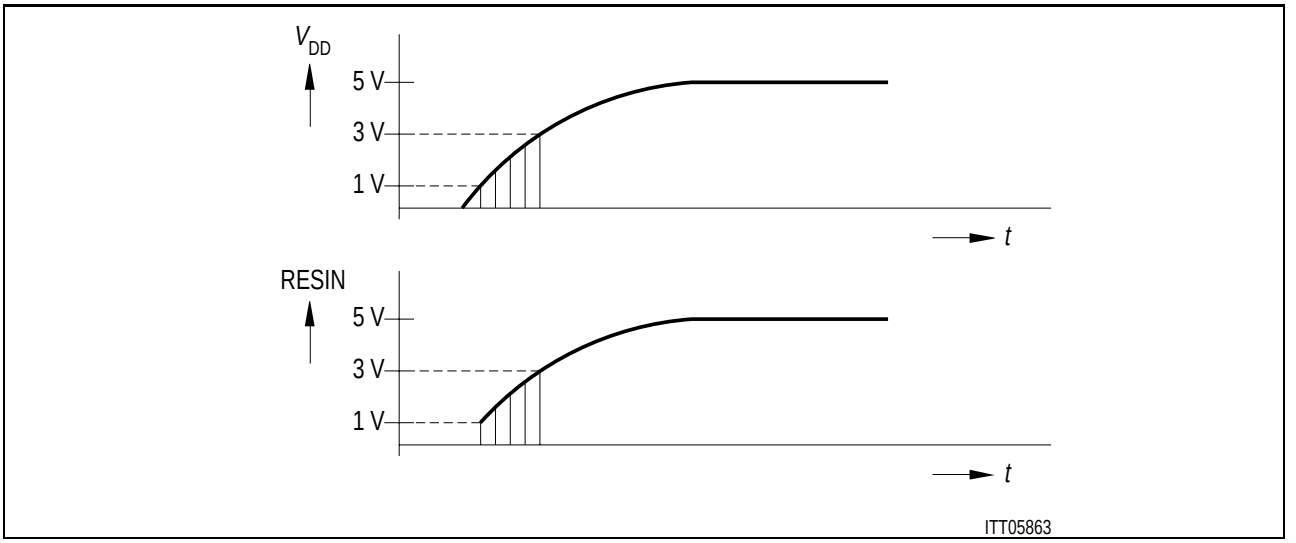



**Figure 136**
**Power-Up Reset Behaviour**

Power-up reset is generated if $V_{DD}$ raises from less than 1 V to more than 3 V.

## Boundary Scan Timing

| Parameter | Symbol | Limit Values | | Unit |
|---|---|---|---|---|
| | | min. | max. | |
| Test clock period | $t_{TCP}$ | 160 | | ns |
| Test clock period low | $t_{TCPL}$ | 80 | | ns |
| Test clock period high | $t_{TCPH}$ | 80 | | ns |
| TMS-set-up time to TCK | $t_{MSS}$ | 30 | | ns |
| TMS-hold time from TCK | $t_{MSH}$ | 30 | | ns |
| TDI-set-up time to TCK | $t_{DIS}$ | 30 | | ns |
| TDI-hold time from TCK | $t_{DIH}$ | 30 | | ns |
| TDO-valid delay from TCK | $t_{DOD}$ | | 60 | ns |

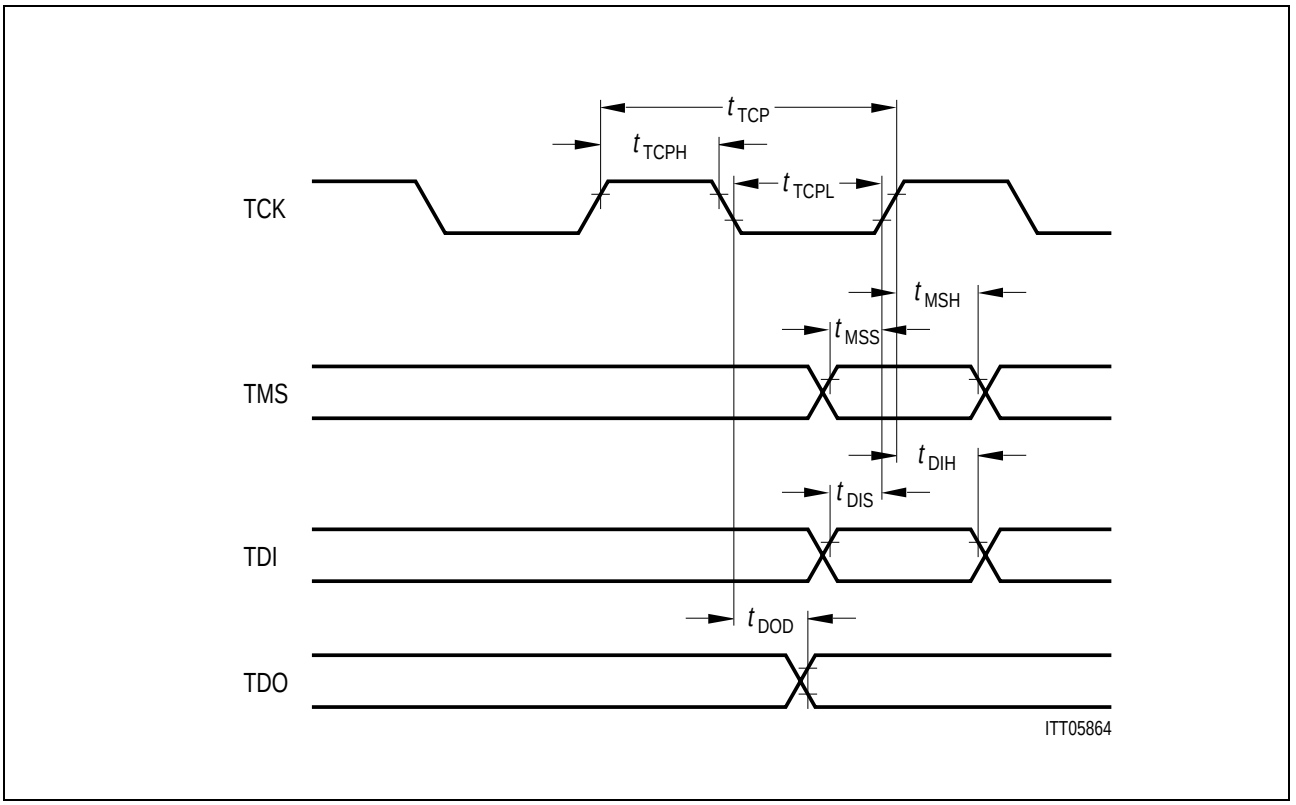



**Figure 137**
**Boundary Scan Timing**

## Serial Interface Timing

| Parameter | Symbol | Limit Values | | Unit |
|---|---|---|---|---|
| | | min. | max. | |
| Receive data set-up | $t_{RDS}$ | 20 | | ns |
| Receive data hold | $t_{RDH}$ | 10 | | ns |
| Collision data set-up | $t_{CDS}$ | 5 | | ns |
| Collision data hole | $t_{CDH}$ | 30 | | ns |
| Transmit data delay | $t_{XDD}$ | 20 | 68 | ns |
| Tristate control delay | $t_{RTD}$ | 20 | 85 | ns |
| Clock period | $t_{CP}$ | 240 | | ns |
| Clock period low | $t_{CPL}$ | 90 | | ns |
| Clock period high | $t_{CPH}$ | 90 | | |
| Strobe set-up time to clock | $t_{XSS}$ | 80 | $t_{CP}$ – 30 | ns |
| Strobe set-up time to clock (ext. transp. mode) | $t_{XSX}$ | 30 | $t_{CP}$ – 30 | ns |
| Strobe hold time from clock | $t_{XSH}$ | 30 | | ns |
| Transmit data delay from strobe | $t_{SDD}$ | | 90 | |
| Transmit data high impedance from clock | $t_{XCZ}$ | | 65 | ns |
| Transmit data high impedance from strobe | $t_{XSZ}$ | | 50 | ns |
| Sync pulse set-up time to clock | $t_{SS}$ | 30 | $t_{CP}$ – 30 | |
| Sync pulse width | $t_{SW}$ | 40 | | ns |

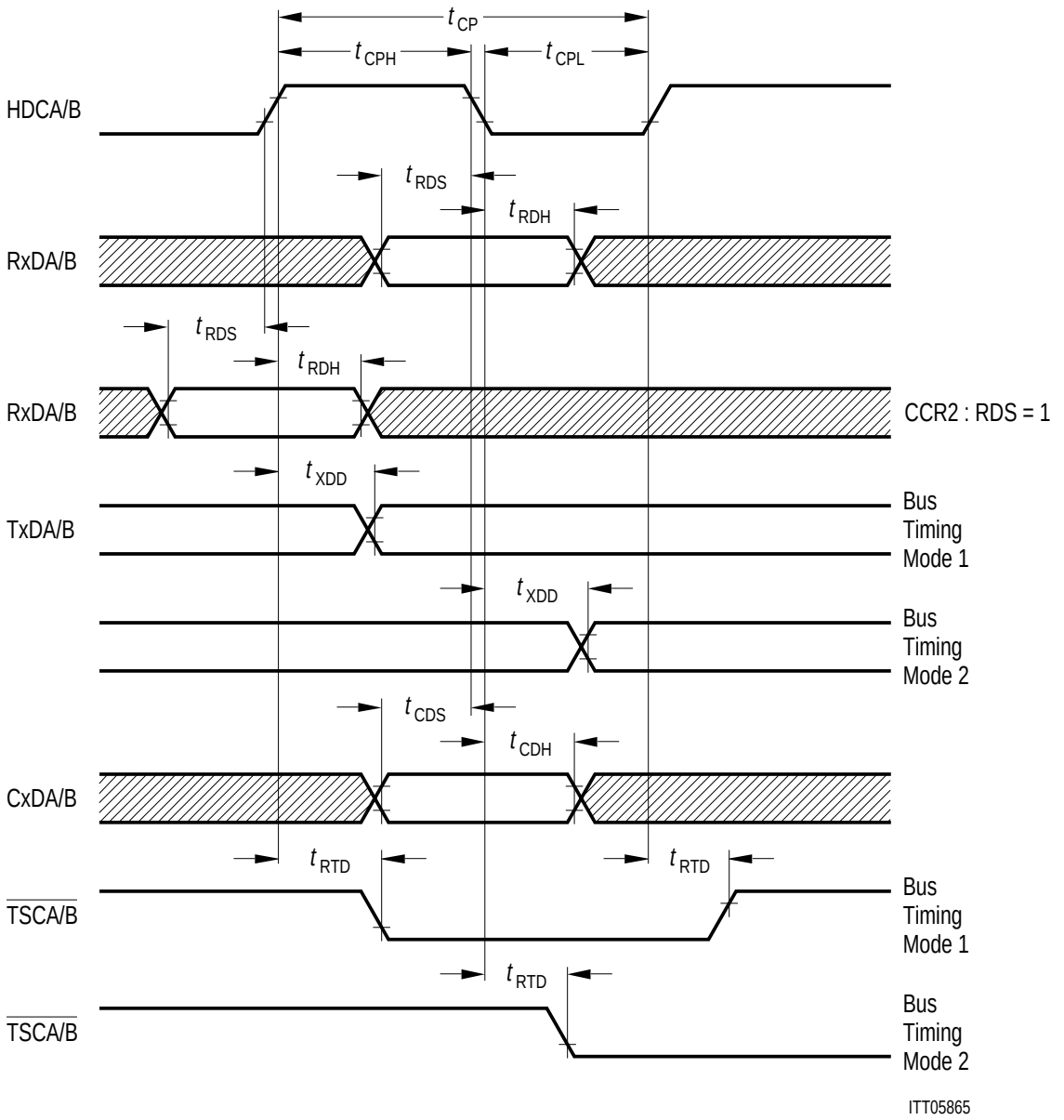



**Figure 138**
**Serial Interface Timing**

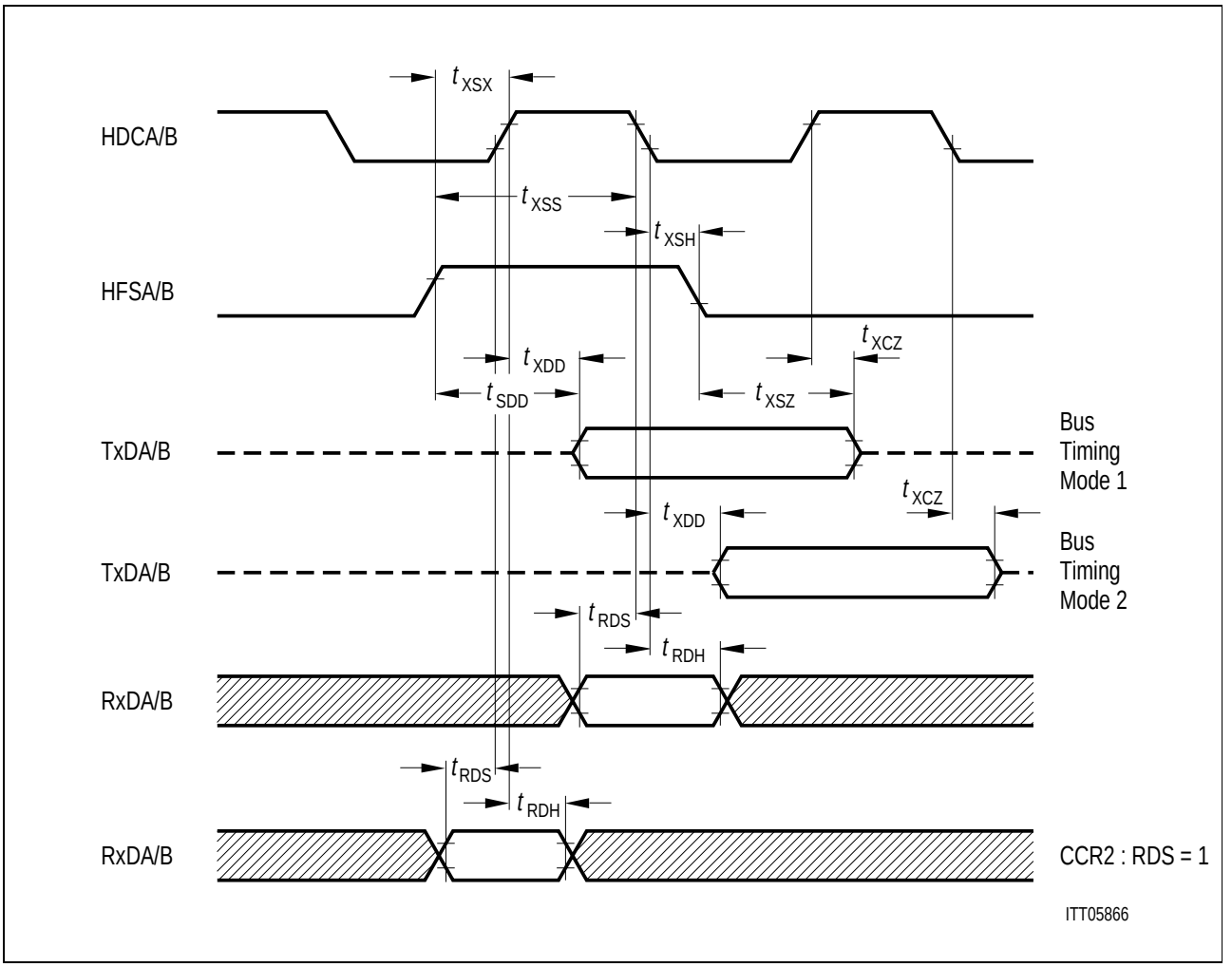



**Figure 139**
**Serial Interface Strobe Timing (clock mode 1)**

*Note: With RDS = 1 the sampling edge is shifted 1/2 clock phase forward. The data is internally still processed with the falling edge. Therefore the strobe timing is still relative to the next falling edge in that case.*

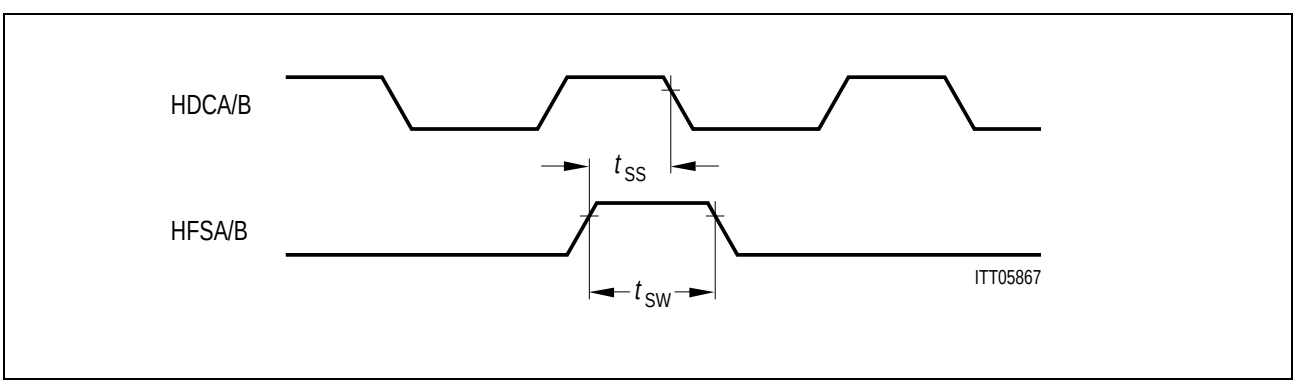



**Figure 140**
**Serial Interface Synchronization Timing (clock mode 2)**

## PCM and Configurable Interface Timing

| Parameter | Symbol | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|
| | | min. | max. | | |
| Clock period | $t_{CP}$ | 240 | | ns | clock frequency ≤ 4096 kHz |
| Clock period low | $t_{CPL}$ | 80 | | ns | |
| Clock period high | $t_{CPH}$ | 100 | | ns | |
| Clock period | $t_{CP}$ | 120 | | ns | clock frequency > 4096 kHz |
| Clock period low | $t_{CPL}$ | 50 | | ns | |
| Clock period high | $t_{CPH}$ | 50 | | ns | |
| Frame set-up time to clock | $t_{FS}$ | 25 | | ns | |
| Frame hold time from clock | $t_{FH}$ | 50 | | ns | |
| Data clock delay | $t_{DCD}$ | | 125 | | |
| Serial data input set-up time | $t_S$ | 7 | | | PCM-input data frequency > 4096 kbit/s |
| Serial data hold time from | $t_H$ | 35 | | ns | |
| Serial data input set-up time | $t_S$ | 15 | | ns | PCM-input data frequency ≤ 4096 kbit/s |
| Serial data hold time from | $t_H$ | 55 | | ns | |
| Serial data input set-up time | $t_S$ | 20 | | | CFI-input data frequency > 4096 kbit/s |
| Serial dta hold time from | $t_H$ | 50 | | | |
| Serial data input set-up time | $t_S$ | 0 | | ns | CFI-input data frequency ≤ 4096 kbit/s |
| Serial data hold time from | $t_H$ | 75 | | ns | |
| PCM-serial data output delay | $t_D$ | | 55 | ns | |
| Tristate control delay | $t_T$ | | 60 | | |
| CFI-serial data output delay | $t_{CDF}$ | | 65 | | falling clock edge |
| CFI-serial data output delay | $t_{CDR}$ | – | 90 | ns | rising clock edge |

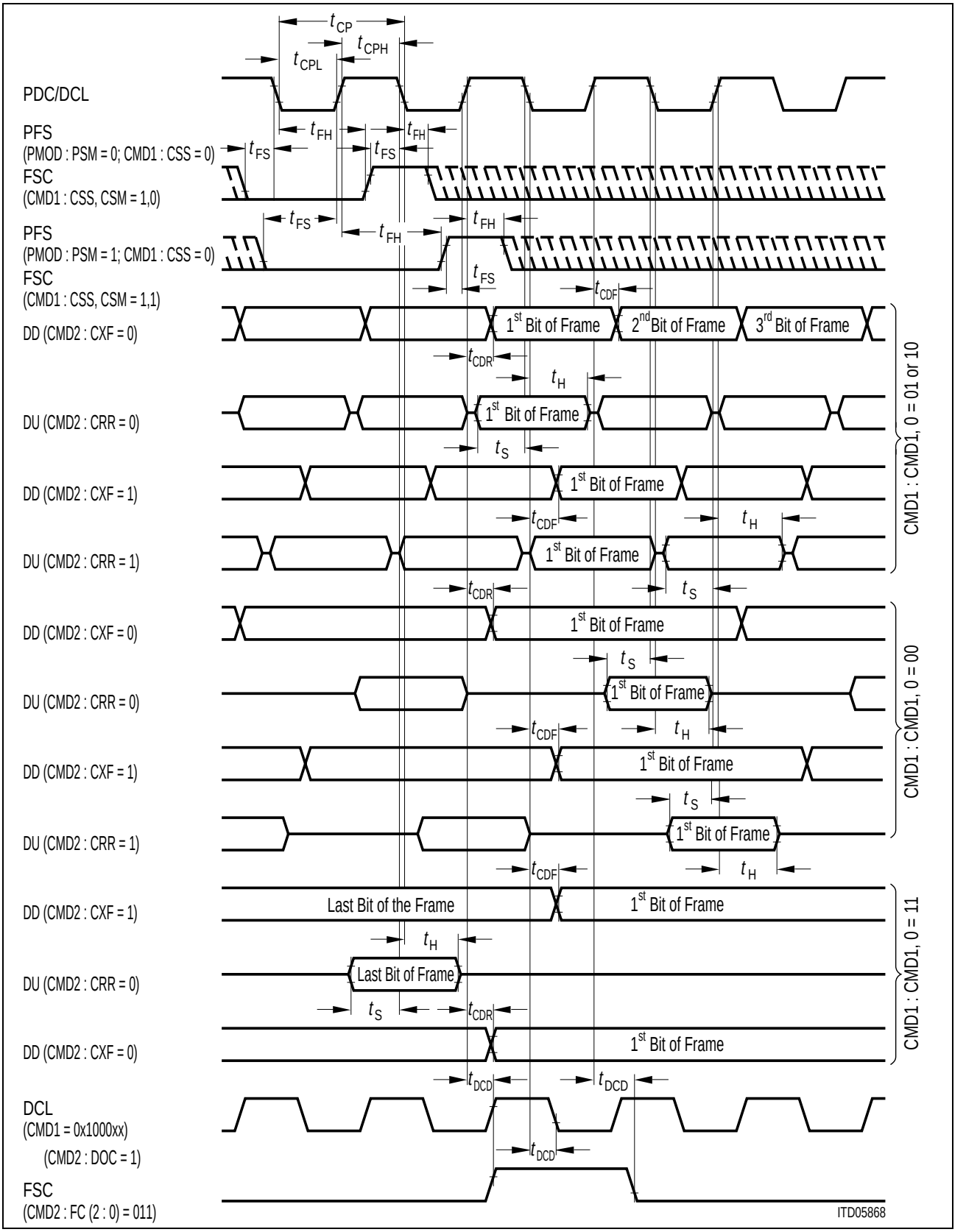



**Figure 141**
**Configurable Interface Timing, CMD:CSP1,0 = 10 (prescalor divisor = 1)**

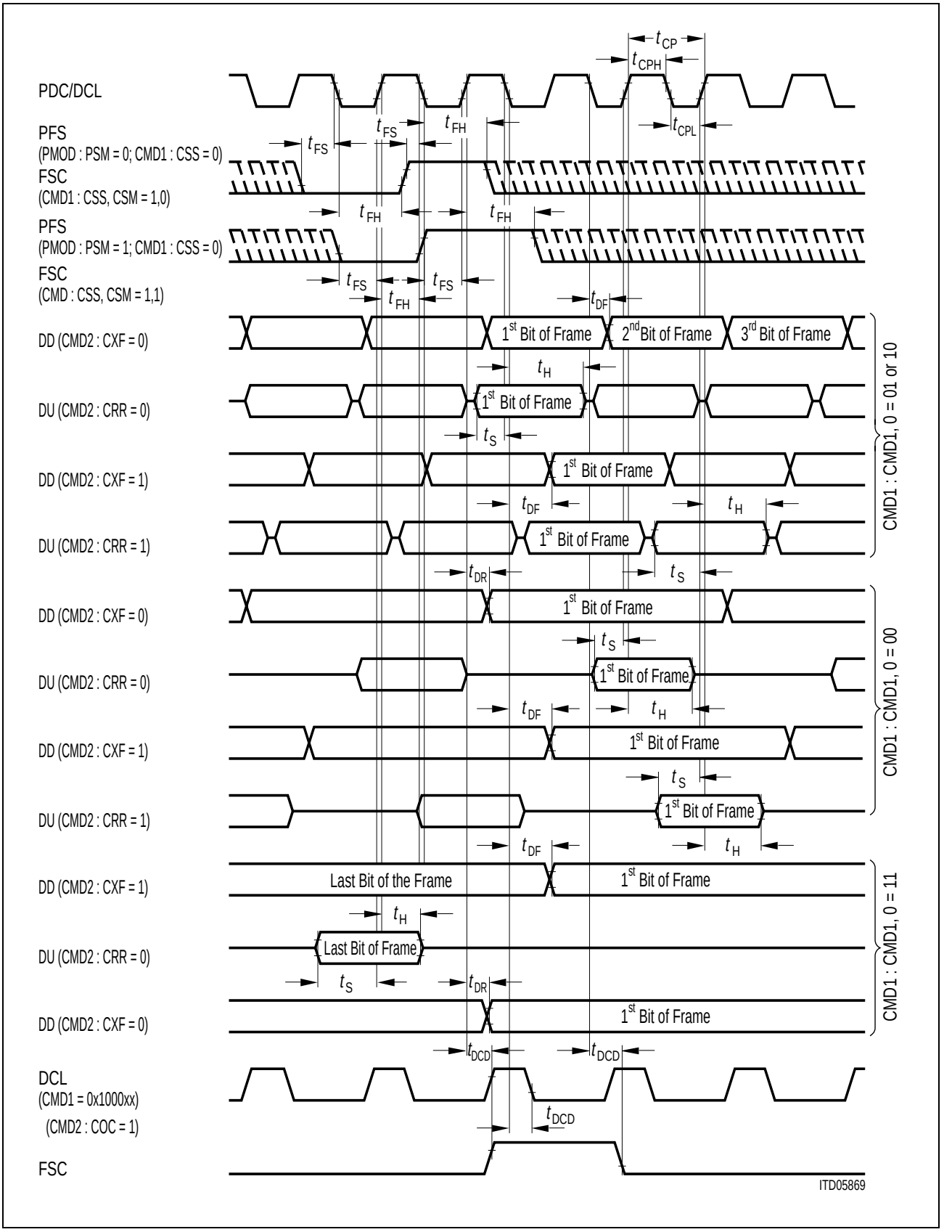



**Figure 142**
**Configurable Interface Timing, CMD:CSP1,0 = 01 (prescalor divisor = 1,5)**

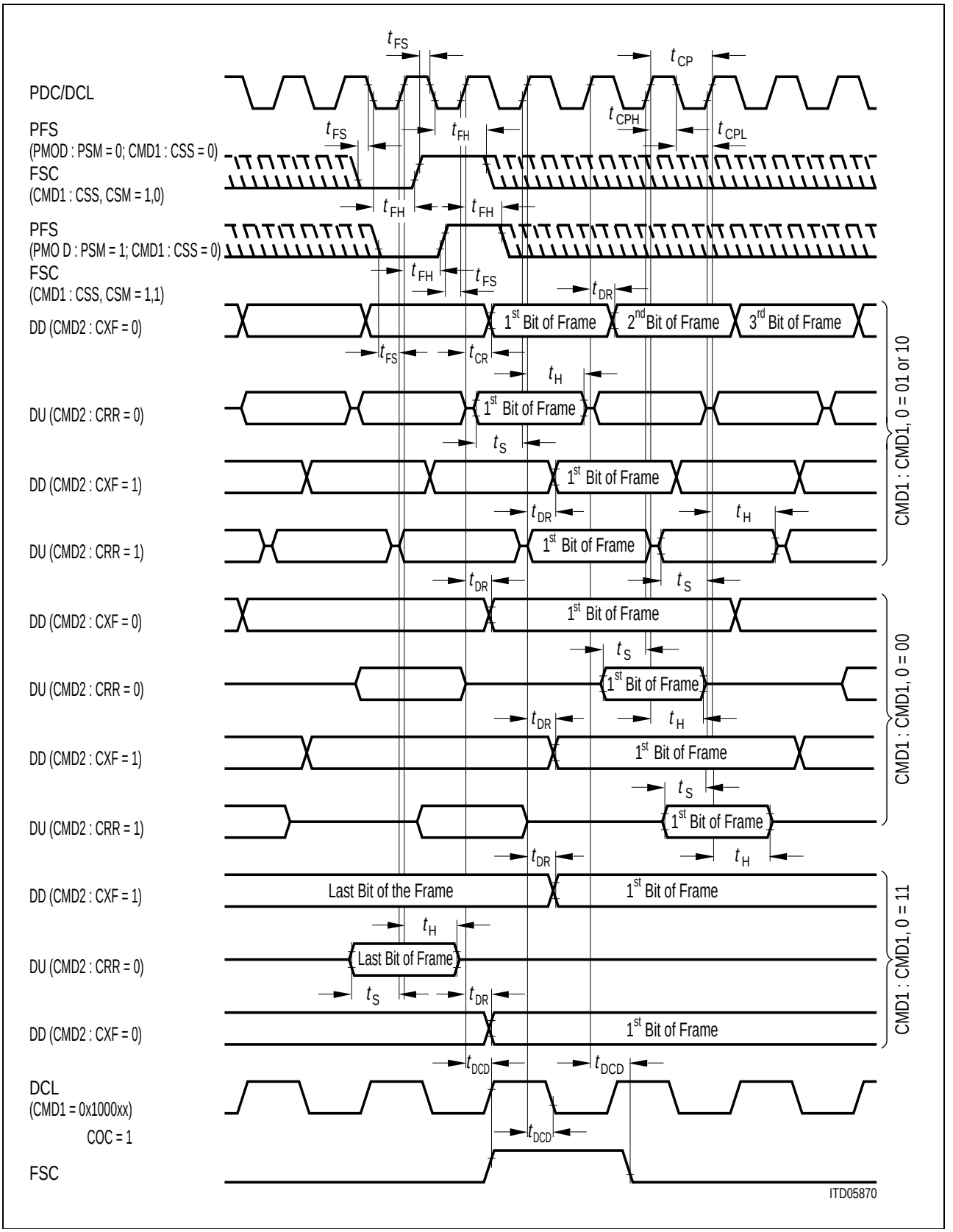



**Figure 143**
**Configurable Interface Timing, CMD:CSP1,0 = 00 (prescalor divisor = 2)**

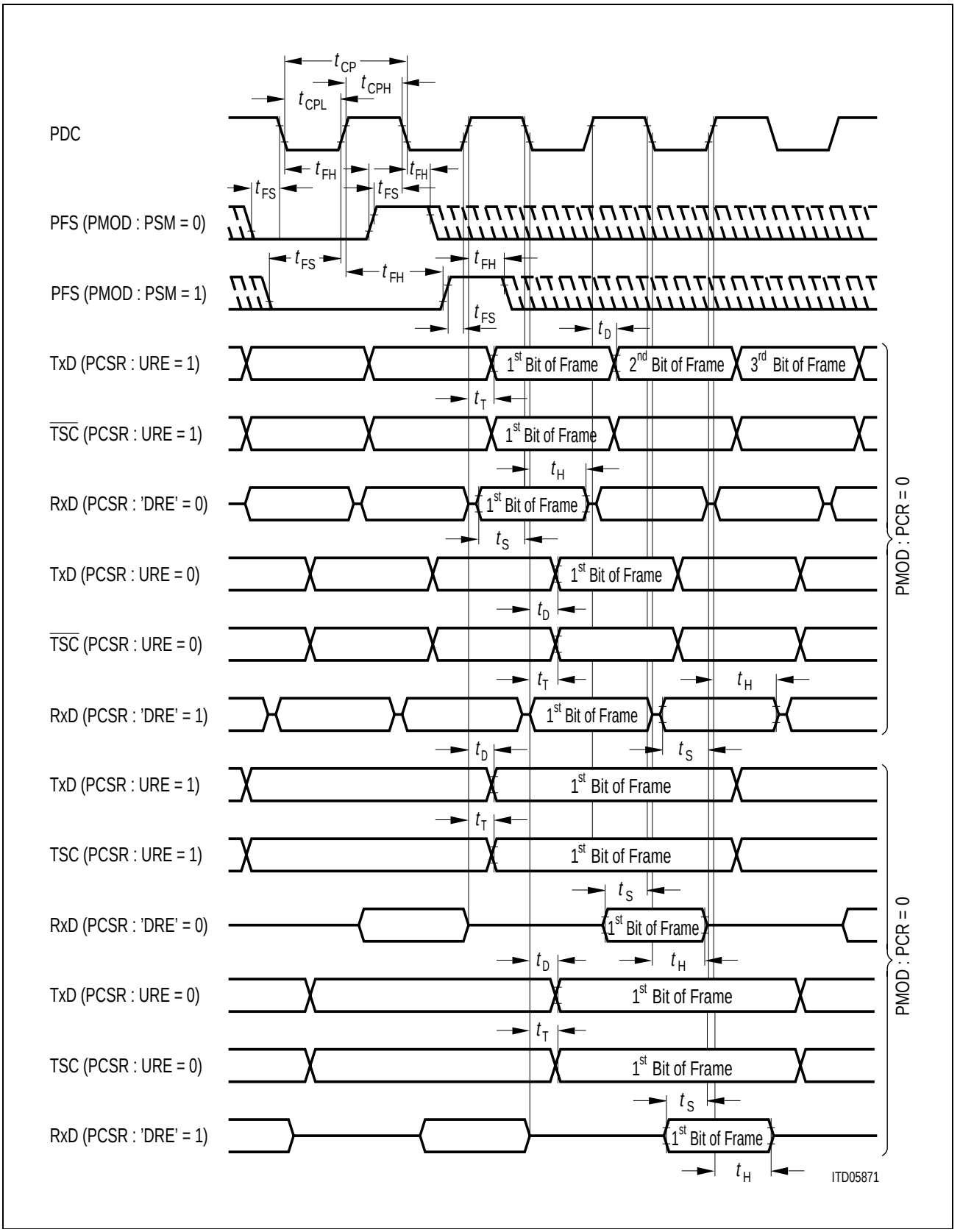
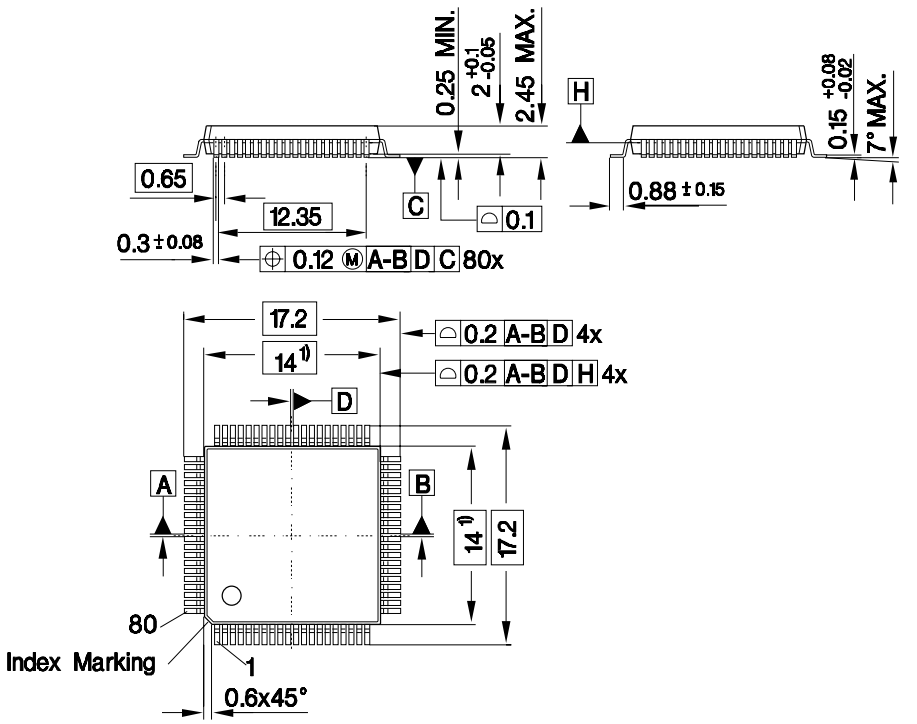



**Figure 144**
**PCM-Interface Timing**

## 8 Package Outlines

**P-MQFP-80-1**
(Plastic Metric Quad Flat Package)




1) Does not include plastic or metal protrusion of 0.25 max. per side

**Sorts of Packing**
Package outlines for tubes, trays etc. are contained in our Data Book "Package Information"

SMD = Surface Mounted Device

Dimensions in mm

# 9 Appendix

## 9.1 Differences between EPIC®-1 (PEB 2055) and the ELIC®-EPIC®

- In demultiplexed address mode the ELIC registers can (in comparison to the EPIC-1) additionally be addressed by A4..A0 (see register OMDR:RBS).
  If A4 is connected to ground, the registers are addressed by A3..A0 and RBS. This is compatible to the EPIC-1.
- The demultiplexed addresses can also be used in multiplexed mode (see register EMOD:DMXAD).
- The ELIC-EPIC has 4 PCM modes. PCM mode 3 is similar to PCM mode 1 unlike in PCM mode 1 the pins TXD1, TXD3 are not tristated, but drive the inverted values of TXD0, TXD2.
- The error in the double last look logic of the EPIC-1 up to Version A3 (6 bit C/I-channel change in time slot 1 and 3 will not be recognized; see EPIC errata sheet 02.95) was corrected in ELIC-EPIC.
- In preprocessed applications the combination of MACR:CMC3..0 = 1010 (for the even CMC address field) is used in downstream direction for the D-channel handling of SACCO_A with the arbiter.
- In a double clock rate configuration (clock frequency is twice the data rate), the ELIC PCM input and output data can be shifted additionally by one PDC clock (see register bits PCSR:DRCS and PCSR:ADSRO). If these two bits are not set, the ELIC-EPIC is compatible to the EPIC-1.
  This feature guarantees the capability to adapt to a PCM data stream also in double clock rate mode (unlike the EPIC-1 up to Version A3), when the negative PDC edge is used to synchronize PFS.
- The ELIC is able to generate a 2 Mbit/s (PCM and CFI) datastream out of a 2 MHz PCM clock also in CFI mode 0. See register EMOD:ECMD2.

## 9.2 Working Sheets

The following pages contain some working sheets to facilitate the programming of the EPIC-1. For several tasks (i.e. initialization, time slot switching, ...) the corresponding registers are summarized in a way the programmer gets a quick overview on the registers he has to use.

### 9.2.1 Register Summary for EPIC® Initialization

**PCM Interface**

**PMOD** PCM Mode Register RW, $20_H$ ($0_H$ + RBS = 1), reset-val. = 00

| PMD | PCR | PSM | AIS | AIC |
|---|---|---|---|---|

PMD0..1 = PCM Mode, 00 = 0, 01 = 1, 10 = 2
PCR = PCM Clock Rate:
0 = equal to PCM data rate
1 = double PCM data rate (not for mode 2)
PSM = PCM Synchron Mode:
0 = frame synchr. with falling edge,
1 = rising edge of PDC
AIS0..1 = Alternative Input Section: (PCM mode dependent)
Mode 0: AIS = 0
Mode 1: AIS0 = 0: RXD1 = IN0, AIS0 = 1: RXD0 = IN0
AIS1 = 0: RXD3 = IN1, AIS1 = 1: RXD2 = IN1
Mode 2: AIS0 = 0
AIS1 = 0: RXD3 = IN, AIS = 1: RXD2 = IN
AIC0..1 = Alternative Input Comparison: (PCM mode dependent)
Mode 0, 1: AIC0 = 0: no comparison, AIC0 = 1: RXD0 == RXD1
AIC1 = 0: no comparison, AIC1 = 1: RXD2 == RXD3
Mode 2: AIC0 = 0:
AIC1 = 0: no comparison, AIC1 = 1: RXD2 == RXD3

**PBNR** PCM Bit Number Register RW, $22_H$ ($1_H$ + RBS = 1), reset-val. = FF

| BNR |
|---|

BNR0..7 = Bit Number per Frame (mode dependent)
Mode 0: BNR = number of bits – 1
Mode 1: BNR = (number of bits)/2 – 1
Mode 2: BNR = (number of bits)/4 – 1

**Figure 145 a**
**EPIC® Initialization Register Summary (working sheet)**

**POFD** PCM Offset Downstream Register RW, $24_H$ ($2_H$ + RBS = 1), reset-val. = 0

| OFD9..2 |
|---|

OFD2..9 = Offset Downstream (see PCSR for OFD0..1)
Mode 0: (BND – 17 + BPF) mod BPF --> OFD2..9
Mode 1: (BND – 33 + BPF) mod BPF --> OFD1..9
Mode 2: (BND – 65 + BPF) mod BPF --> OFD0..9
BND = number of bits + 1 that the downstream frame start is left shifted relative to the frame sync
BPF = number of bits per frame
Unused bits must be set to 0 !

**POFU** PCM Offset Upstream Register RW, $26_H$ ($3_H$ + RBS = 1), reset-val. = 0

| OFU9..2 |
|---|

OFU2..9 = Offset Upstream (see PCSR for OFU0..1)
Mode 0: (BND + 23 + BPF) mod BPF --> OFU2..9
Mode 1: (BND + 47 + BPF) mod BPF --> OFU1..9
Mode 2: (BND + 95 + BPF) mod BPF --> OFU0..9
BND = number of bits + 1 that the upstream frame is left shifted relative to the frame start
BPF = number of bits per frame
Unused bits must be set to 0 !

**PCSR** PCM Clock Shift Register RW, $28_H$ ($4_H$ + RBS = 1), reset-val. = 0

| 0 | OFD1..0 | DRE | 0 | OFU1..0 | URE |
|---|---|---|---|---|---|

OFD0..1 = Offset Downstream (see POFD)
DRE = Downstream Rising Edge,
0 = receive data on falling edge,
1 = receive data on rising edge
OFU0..1 = Offset Upstream (see POFU)
URE = Upstream Rising Edge,
0 = send data on falling edge,
1 = send data on rising edge

**Figure 145 b**
**EPIC® Initialization Register Summary (working sheet)**

**CFI Interface**

**CMD1** CFI Mode Register 1 RW, $2C_H$ ($6_H$ + RBS = 1), reset-val. = 00

| | | | | |
|---|---|---|---|---|
| CSS | CSM | CSP1..0 | CMD1..0 | CIS1..0 |

CSS = Clock Source Select,
0 = PDC/PFS used for CFI,
1= DCL/FSC are inputs
CSM = CFI Synchronization Mode:
1 = frame syncr. with rising edge,
0 = falling edge of DCL
if CSS = 0 ==> CMD1:CSM = PMOD:PSM !
CSP0..1 = Clock Source Prescaler: 00 = 1/2, 01 = 1/1.5, 10 = 1/1
CMD0..1 = CFI Mode: 00 = 0, 01 = 1, 10 = 2, 11 = 3
CIS0..1 = CFI Alternative Input Section
Mode 0, 3: CIS0..1 = 0
Mode 1, 2: CIS0: 0 = IN0 = DU0, 1 = IN0 = DU2
Mode 1: CIS1: 0 = IN1 = DU1, 1 = IN1 = DU3

**CMD2** CFI Mode Register 2 RW, $2E_H$ ($7_H$ + RBS = 1), reset-val. = 00

| | | | | |
|---|---|---|---|---|
| FC2..0 | COC | CXF | CRR | CBN9..8 |

**For IOM®-2 CMD2 can be set to $D0_H$**
FC0..2 = Framing Signal Output Control (CMD1:CSS = 0)
= 010 suitable for PBC, = 011 for IOM-2, = 110 IOM-2 and SLD
COC = Clock Output Control (CMD1:CSS = 0)
= 0 DCL = data rate,
= 1 DCL 2 × data rate (only mode 0 and 3 !)
CXF = CFI Transmit on Falling Edge: 0 = send on rising edge, 1 = send on falling DCL edge
CRR = CFI Receive on Rising Edge: 0 = receive on falling edge, 1 = send on rising DCL edge
CBN8..9 = CFI Bit Number (see CBNR)

**CBNR** CFI Bit Number Register RW, $30_H$ ($8_H$ + RBS = 1), reset-val. = FF

| |
|---|
| CBN |

CBN0..7 = CFI Bit Number per Frame – 1 (see CMD2:CBN8..9)

**Figure 145 c**
**EPIC® Initialization Register Summary (working sheet)**

**CTAR** CFI Time Slot Adjustment Register RW, $32_H$ ($9_H$ + RBS = 1), reset-val. = 00

| | |
|---|---|
| 0 | TSN |

TSN0..6 = (number of time slots + 2) the DU and DD frame is left shifted relative to frame start (see also CBSR)

**CBSR** CFI Bit Shift Register RW, $34_H$ ($A_H$ + RBS = 1), reset-val. = 00

| | | |
|---|---|---|
| 0 | CDS2..0 | CUS3..0 |

CDS2..0: CFI Downstream/Upstream Bit Shift
Shift DU and DD frame:
000 = 2 bits right
001 = 1 bit right
010 = 6 bits left
011 = 5 bits left
100 = 4 bits left
101 = 3 bits left
110 = 2 bits left
111 = 1 bit left
Relative to PFS (if CMD1:CSS = 0)
Relative to FSC (if CMD1:CSS = 1)

**CSCR** CFI Subchannel Register RW, $36_H$ ($A_H$ + RBS = 1), reset-val. = 00

| | | | |
|---|---|---|---|
| CS3 | CS2 | CS1 | CS0 |

SC3 0..1 control port 3 (+ port 7 for CFI mode 3 (SLD))
SC2 0..1 control port 2 (+ port 6 for CFI mode 3 (SLD))
SC1 0..1 control port 1 (+ port 5 for CFI mode 3 (SLD))
SC0 0..1 control port 0 (+ port 4 for CFI mode 3 (SLD))
for 64 kBit/s channel: 00/01/10/11 = bits 7..0
for 32 kBit/s channel: 00/10 = bits 7..4,
01/11 = bits 3..0

**Figure 145 d**
**EPIC® Initialization Register Summary (working sheet)**

### 9.2.2 Switching of PCM Time Slots to the CFI Interface (data downstream)

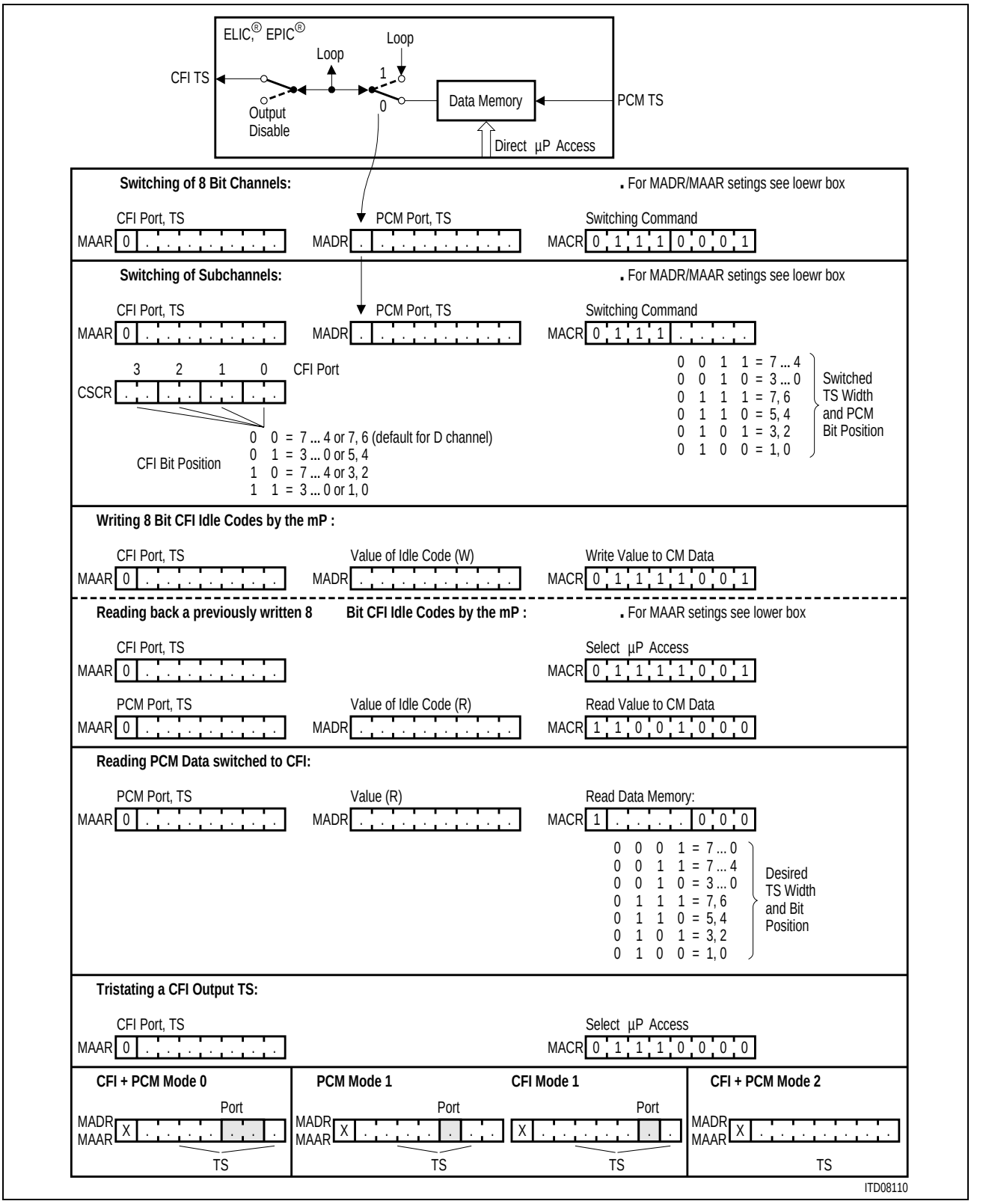



**Figure 146**
**Switching of PCM Time Slots to the CFI Interface (working sheet)**

### 9.2.3 Switching of CFI Time Slots to the PCM Interface (data upstream)

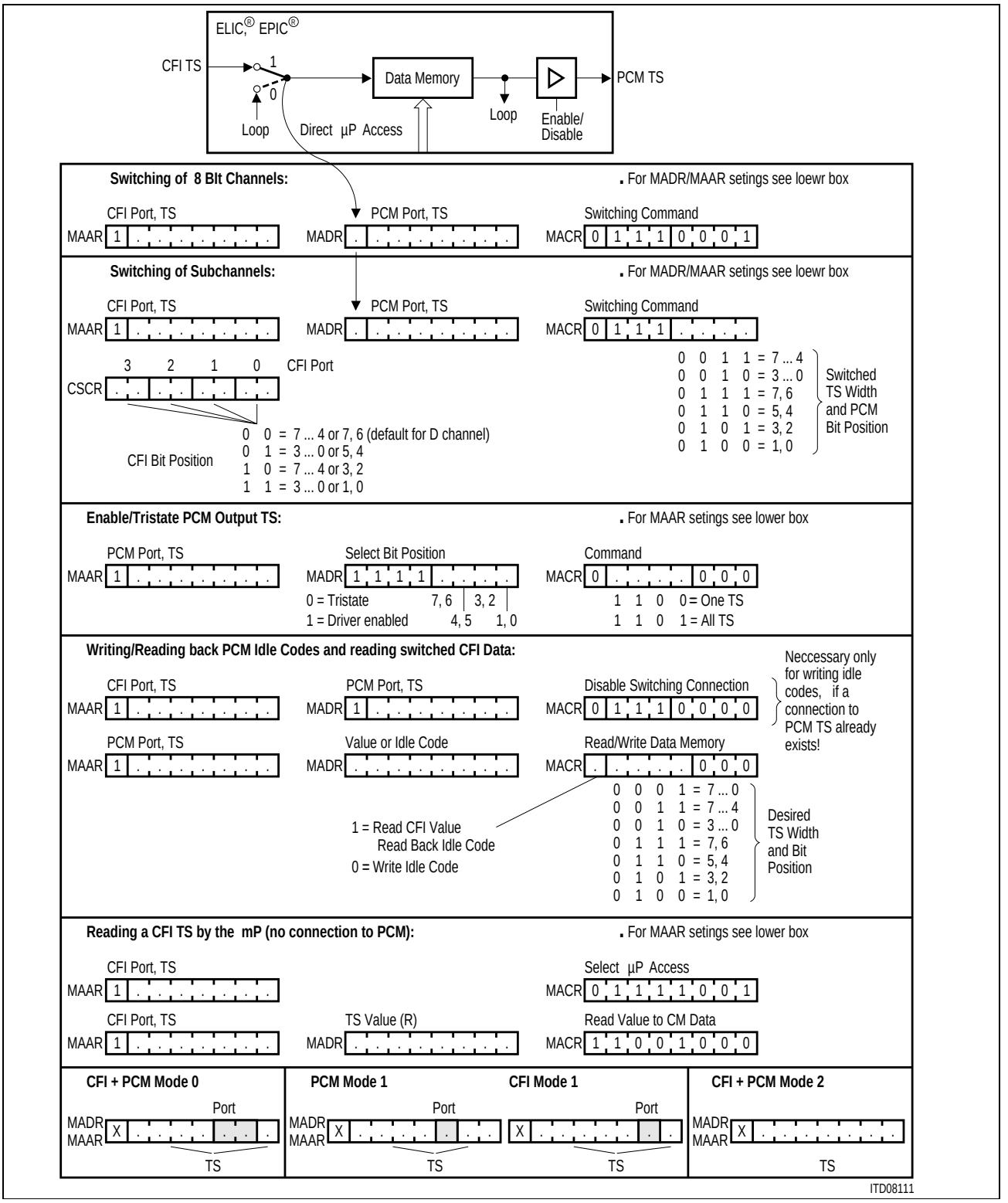



**Figure 147**
**Switching of CFI Time Slots to the PCM Interface (working sheet)**

### 9.2.4 Preparing EPIC®s C/I Channels

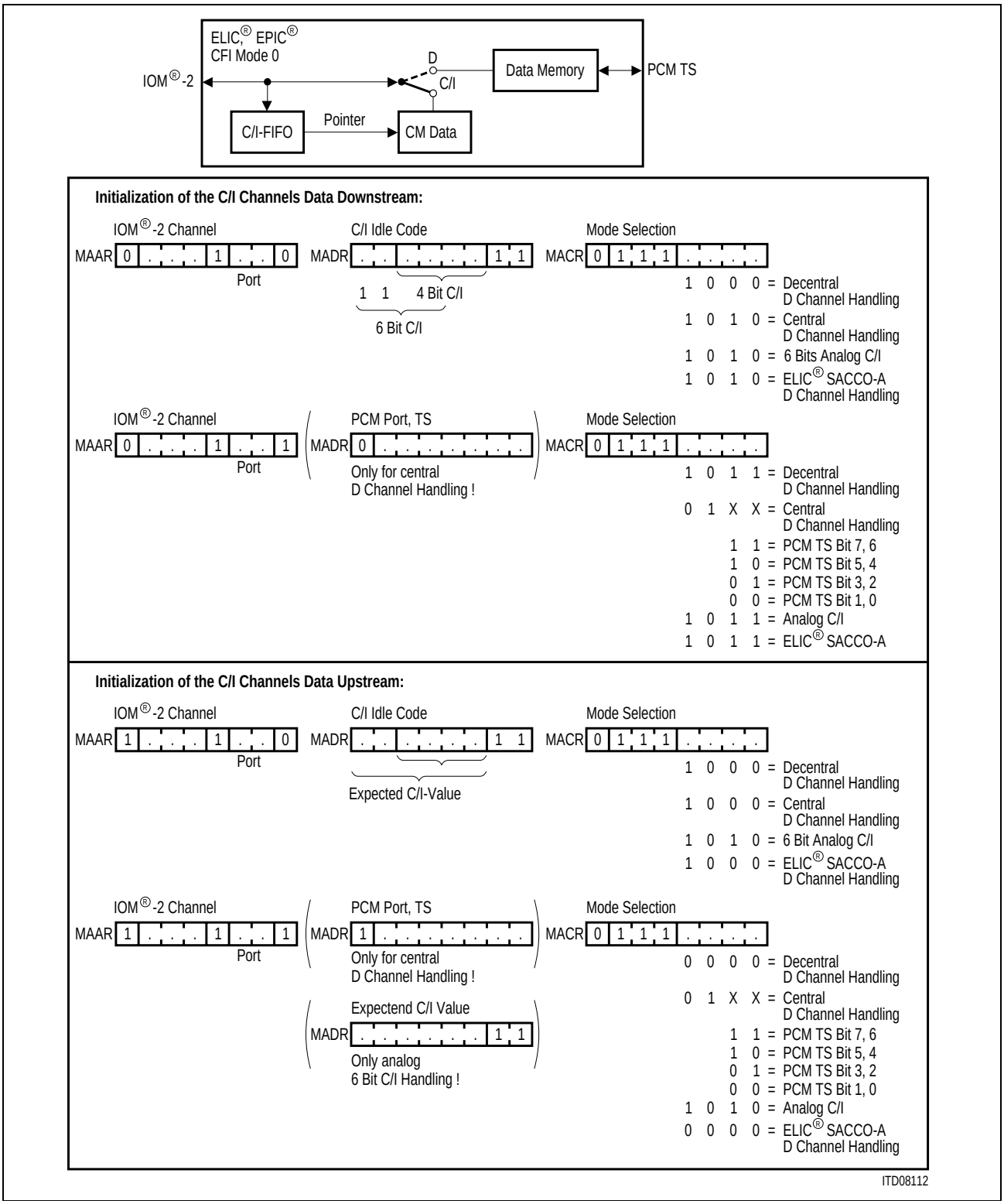



**Figure 148**
**Preparing EPIC®s C/I Channels (working sheet)**

### 9.2.5 Receiving and Transmitting IOM®-2 C/I-Codes

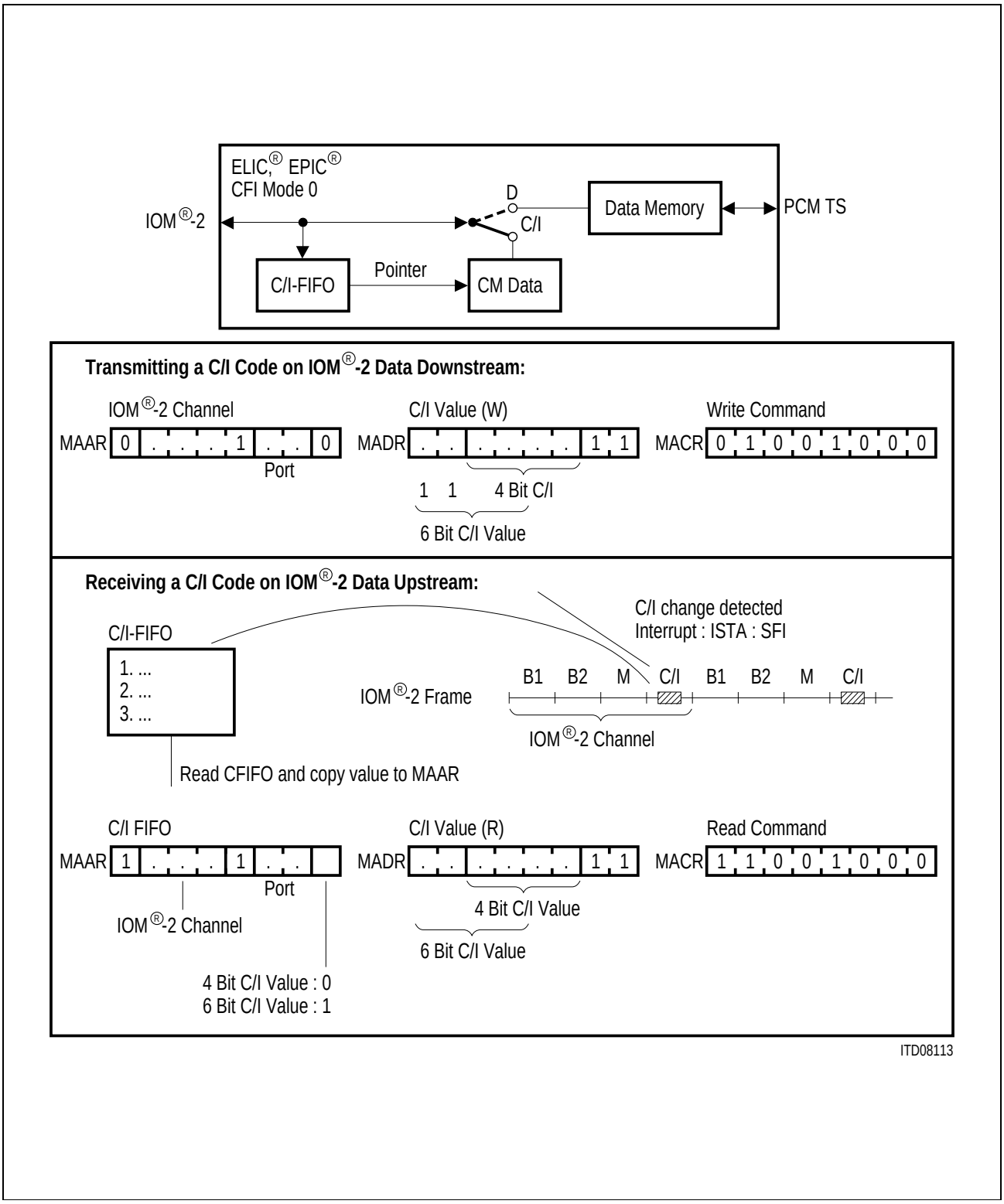



**Figure 149**
**Receiving and Transmitting IOM®-2 C/I-Codes (working sheet)**

## 9.3 Development Tools

The SIPB 5000 system can be used as a platform for all development steps. In a later stage it is of course necessary to make a cost optimized design. For this, a subset of the board design can be used. All the wiring diagrams are shipped with the board to speed up this process.

Siemens offers a very convenient menu driven testing and debugging software. The package that is delivered with the user board, allows a direct access to the chip registers using symbolic names. Subsequent access may be written to a file and run as a track file. Example track files are delivered in the package and will be a great help to the user.

### 9.3.1 SIPB 5000 Mainboard

| Description | Part Number | Ordering Code |
|---|---|---|
| SIPB Mainboard | SIPB 5000 | Q67100-H8647 |

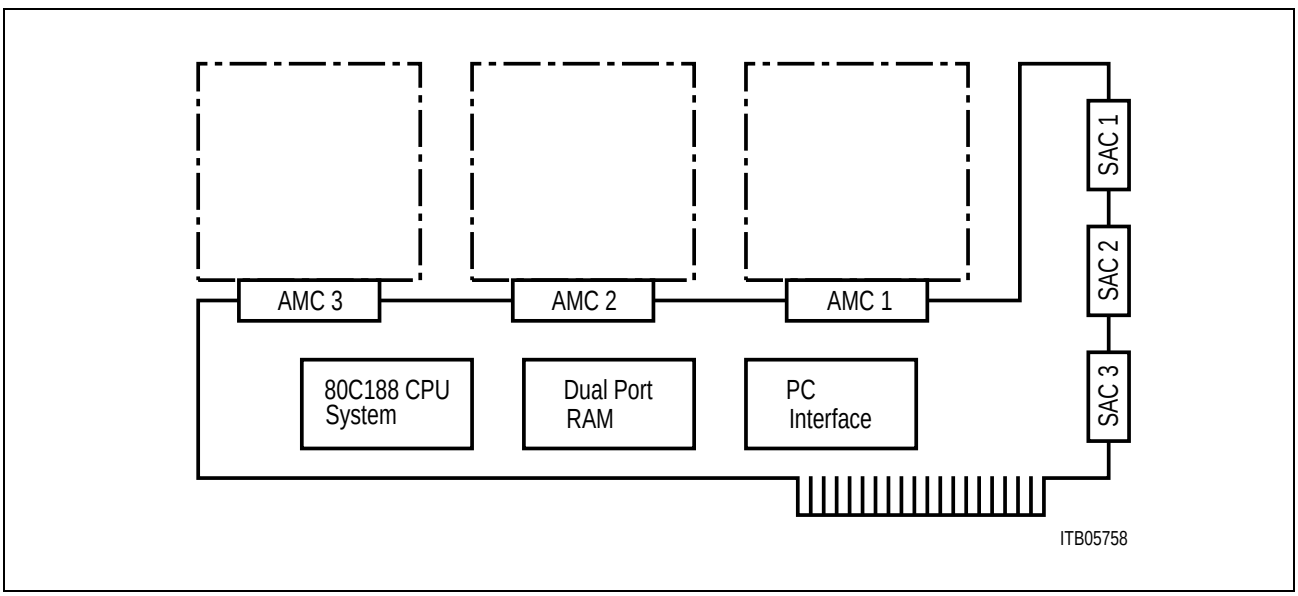



**Figure 150**

The SIPB 5000 Mainboard is the general backbone of the SIPB 5XXX user board system. It is designed as a standard PC interface card, and it contains basically a 80C188 CPU system with 7 interfaces. The interface to the PC is realized both as a Dual Port Ram and as an additional DMA interface. Up to three daughter modules (see dotted blocks) can be added to the Mainboard. They typically carry the components under evaluation. The interfaces which are accessible from the back side of the PC have a connection to the daughter modules as well. This is to allow access to the components under evaluation while the complete board system is hidden inside the PC.

### 9.3.2 SIPB 5122 IOM®-2 Line Card Module (ELIC®)

| Description | Part Number | Ordering Code |
|---|---|---|
| IOM-2 Line Card Module (ELIC) | SIPB 5122 | Q67100-H6397 |

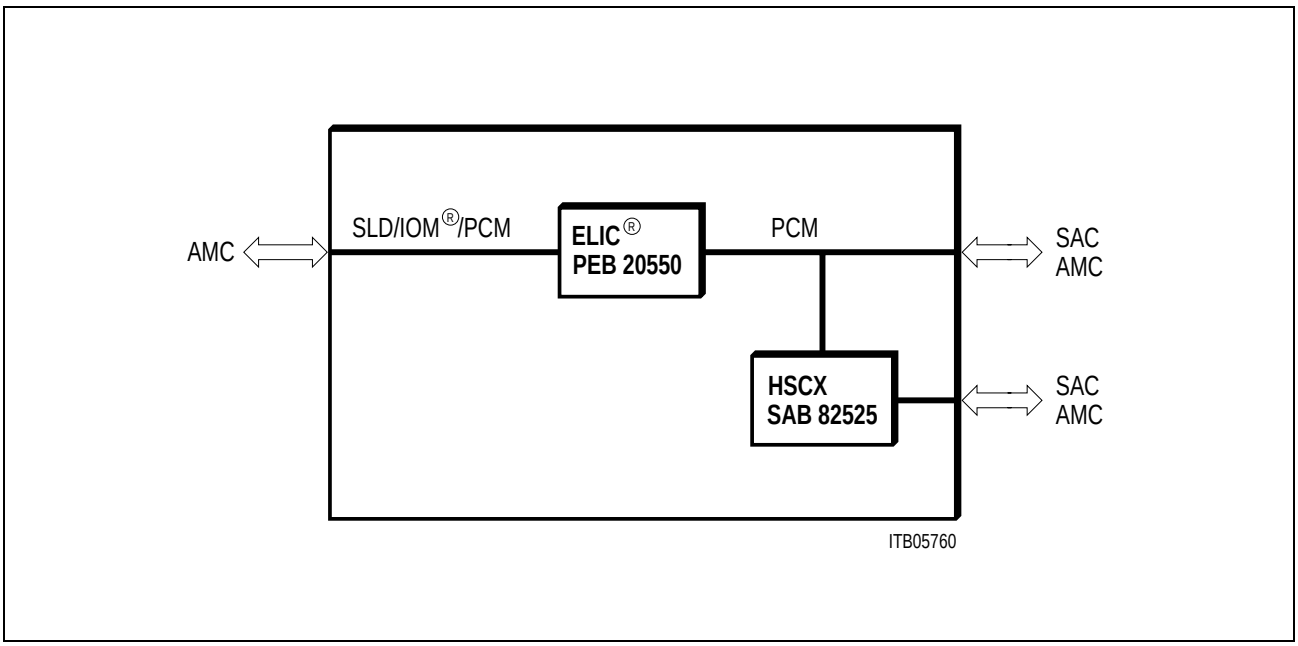



**Figure 151**

The Line Card Module SIBP 5122 is designed to be used with the ISDN User Board SIPB 5000. It serves as an evaluation tool for various line card architectures using the Extended Line Card Interface Controller ELIC PEB 20550.

Possible applications are e.g.:

– Centralized / decentralized D-channel handling of signaling and packet data
– Emulation of a PABX with primary rate module SIPB 7200
– Emulation of a small PABX using two line cards
– Emulation of a digital or analog line card using appropriate layer-1 and/ or CODEC filter modules

# 10 Lists

## 10.1 Glossary

| | |
|---|---|
| ARCOFI® | Audio ringing codec filter |
| BPF | Bits per PCM frame |
| CFI | Configurable interface |
| CM | Control memory |
| CO | Central office |
| DCL | Data clock |
| ELIC® | Extended line interface controller |
| EPIC® | Extended PCM interface controller |
| ETSI | European telecommunication standards institute |
| FIFO | First-in first-out (memory) |
| FSC | Frame synchronisation clock |
| HDCB | HDLC data clock channel B |
| HDLC | High-level data link control |
| IC | Integrated circuit |
| ID | Identifier |
| IOM® | ISDN oriented modular |
| ISAC®-P | ISDN subscriber access controller on U-interface |
| OCTAT®-P | Octal transceiver for $U_{PN}$-interfaces |
| PBC | Peripheral bus controller |
| PBX | Private branch exchange |
| PCM | Pulse code modulation |
| PDC | PCM interface data clock |
| PFS | PCM interface frame synchronisation |
| RHR | Reset HDLC receiver |
| RMC | Receive message complete |
| RME | Receive message end |
| SABME | Set asynchronous balanced mode extended |
| SACCO | Special applications communications controller |
| TE | Terminal equipment |
| UI | Unnumbered information frame |
| $U_{PN}$ | U-interface in private network (PBX) |