DATA SHEET



SAA7392
Channel encoder/decoder CDR60

Preliminary specification
File under Integrated Circuits, IC01

2000 Mar 21

Philips
Semiconductors

PHILIPS

CONTENTS

1 FEATURES
2 GENERAL DESCRIPTION
3 QUICK REFERENCE DATA
4 ORDERING INFORMATION
5 BLOCK DIAGRAM
6 PINNING INFORMATION
6.1 Pinning
6.2 Pin description
7 FUNCTIONAL DESCRIPTION
7.1 Microprocessor interfaces
7.2 Register map
7.3 System clocks
7.4 HF analog front-end
7.5 Bit recovery
7.6 Decoder function
7.7 Subcode interface
7.8 Digital output
7.9 Serial output interface
7.10 Motor control
7.11 The serial in function
7.12 The subcode insert function
7.13 The data encoder block
7.14 Encode control block
7.15 The EFM modulator
7.16 The EFM clock generator
7.17 The Wobble processor
8 LIMITING VALUES
9 OPERATING CHARACTERISTICS
9.1 ADC and AGC parameters
10 APPLICATION INFORMATION
10.1 Write start control of encoder in CD-ROM mode
10.2 Write start control of encoder in Audio mode
10.3 Start-up of encode in flow-control operation
10.4 Start-up of encoder in synchronous stream mode
10.5 Operating CDR60 in CAV mode, flow control on input stream
10.6 Operating in CLV Mode, Flow Control on Input Stream
10.7 Operating in CLV Mode, Synchronous Stream Operation
11 PACKAGE OUTLINE
12 SOLDERING
12.1 Introduction to soldering surface mount packages
12.2 Reflow soldering
12.3 Wave soldering
12.4 Manual soldering
12.5 Suitability of surface mount IC packages for wave and reflow soldering methods
13 DEFINITIONS
14 LIFE SUPPORT APPLICATIONS
15 PURCHASE OF PHILIPS I^2C COMPONENTS

1 FEATURES

- Very high speed Compact Disc (CD) compatible decoding and encoding device
- On-chip Analog-to-Digital Converter (ADC) and Automatic Gain Control (AGC) for HF data capture
- Eight-to-Fourteen Modulation (EFM)
- Advanced motor control loop to allow CAV, CLV and pseudo-CLV playback
- Integrated FIFO for de-coupling of mechanism speed and application speed
- Versatile output interface allowing different I^2S-bus and Electronic Industries Association of Japan (EIAJ) formats
- Device is fully compatible with ELM, PLUM and Sanyo CD-ROM block decoders
- Quad-pass CIRC correction for CD mode (C1-C2-C1-C2)
- Subcode/header processing for CD format
- Frequency multiplier allows use of a 8 MHz crystal.

2 GENERAL DESCRIPTION

CDR60 is a channel encoder/decoder for CD/CD-R/CD-RW/CD Audio Recorder systems. It incorporates all logic and RAM required for the complete encoding and decoding processes.

There are two main datapaths through the CDR60 device. The decode datapath captures the incoming EFM data stream via the HF ADC and AGC functions.

The bit detector recovers the individual bits from the incoming signal, correcting asymmetry, performing noise filtering and equalisation, and recovering the channel bit clock using a digital PLL. The demodulator converts the EFM bits to byte-wide data symbols, before passing them onto the decoder for subcode extraction, de-interleaving and error correction. The decoded data is then made available via the multi-function serial output interface.

The encode datapath takes data symbols from the block encoder/decoder via the serial data and subcode input functions, encoding them via the encoder block. The encoded data stream is passed to the EFM modulator, which generates the required EFM signal, output as a digital bit stream. The encode process is controlled via the Wobble processor, encode control and EFM clock generator functions.

As well as these two data processing sections, three further blocks support overall device operation. The system clock generator provides all digital clocks required by the CDR60. The motor servo allows the CDR60 to control the spindle motor and is controlled by the microprocessor interface. This interface can be accessed either via a parallel (80C51) or a serial (I^2C-bus) interface.

3 QUICK REFERENCE DATA

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
V_{DDD}	supply voltage (core and pad ring)	3.0	3.3	3.6	V
V_{DDA}	supply voltage (analog)	3.0	3.3	3.6	V
V_{DDE}	supply voltage (output drivers)	3.0	3.3	3.6	V
I_{DD}	supply current	–	200	–	mA
f_{xtal}	crystal frequency	8	8.4672	33	MHz
T_{amb}	operating ambient temperature	0	–	70	°C
T_{stg}	storage temperature	–55	–	+125	°C

4 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
SAA7392HL	LQFP80	plastic low profile quad flat package; 80 leads; body 12 × 12 × 1.4 mm	SOT315-1

5 BLOCK DIAGRAM

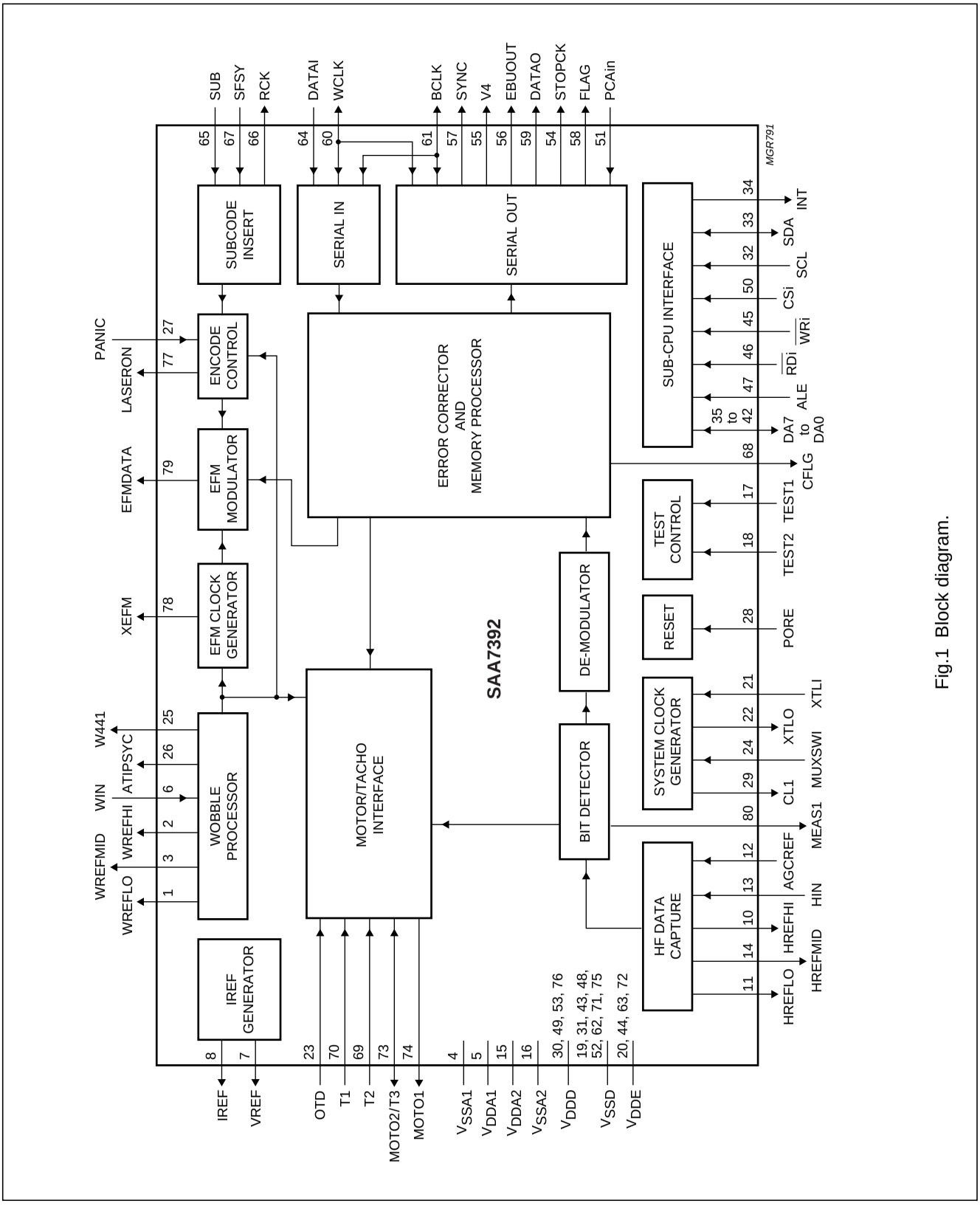


Fig.1 Block diagram.

6 PINNING INFORMATION

6.1 Pinning

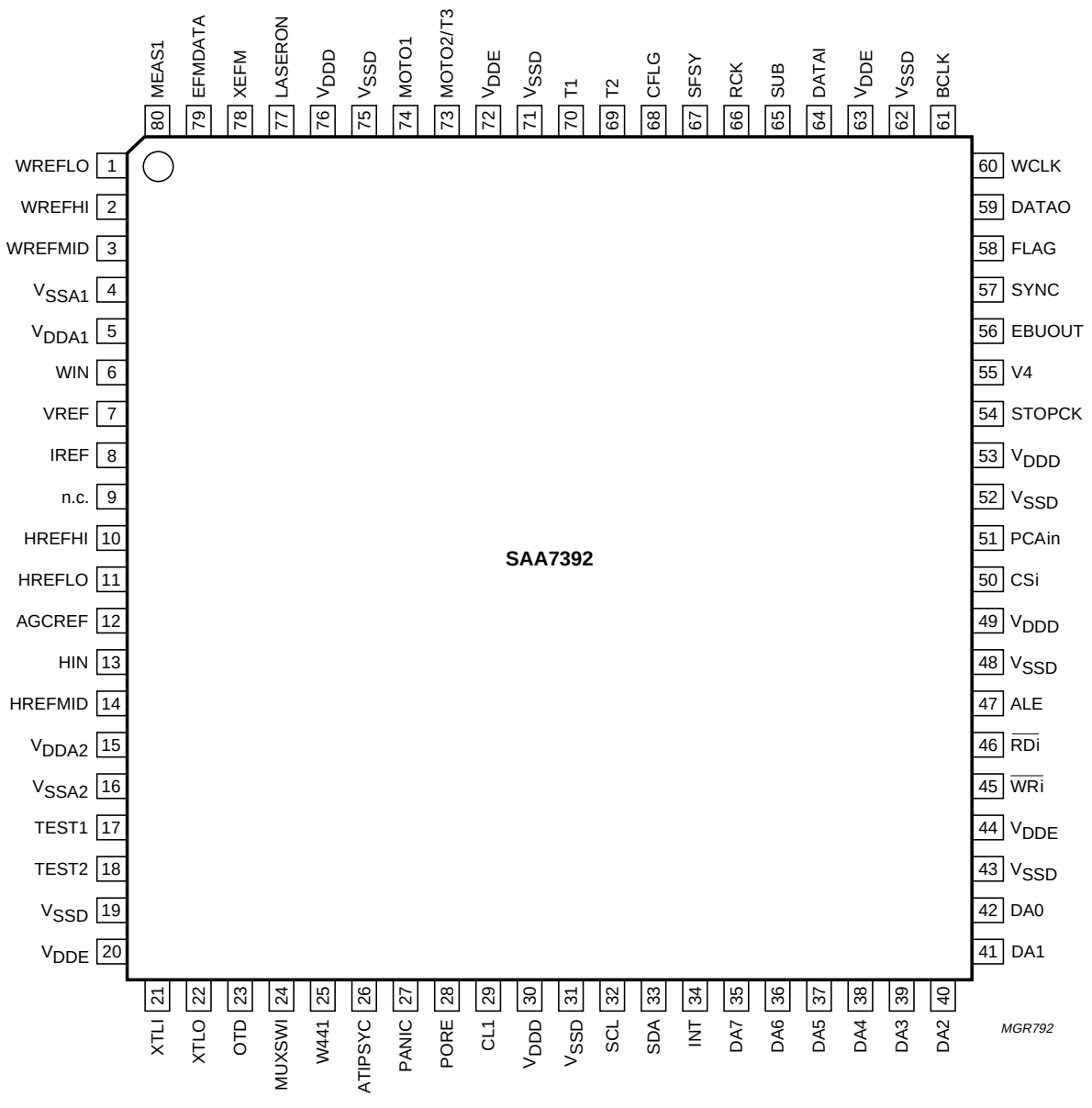


Fig.2 Pin configuration.

6.2 Pin description

Table 1 LQFP80 package; note 1

SYMBOL	PIN	TYPE	DESCRIPTION
WREFLO	1	O	wobble ADC analog reference voltage
WREFHI	2	O	wobble ADC analog reference voltage
WREFMID	3	O	wobble ADC analog reference voltage
V_{SSA1}	4	supply	analog ground
V_{DDA1}	5	supply	3 V analog supply voltage 1; note 2
WIN	6	I	wobble analog input
VREF	7	O	analog voltage reference
IREF	8	O	analog current reference
n.c.	9	–	not connected
HREFHI	10	O	HF ADC analog reference voltage
HREFLO	11	O	HF ADC analog reference voltage
AGCREF	12	I	AGC analog reference voltage
HIN	13	I	HF analog data input
HREFMID	14	O	HF ADC analog reference voltage
V_{DDA2}	15	supply	3 V analog supply voltage 2; note 2
V_{SSA2}	16	supply	analog ground
TEST1	17	I	test input 1
TEST2	18	I	test input 2
V_{SSD}	19, 43, 62, 71	supply	output driver ground
V_{DDE}	20	supply	output driver 3 V supply voltage
XTLI	21	I	crystal oscillator input
XTLO	22	O	crystal oscillator output
OTD	23	I	off track detect input
MUXSWI	24	I	clock multiplier enable
W441	25	O	wobble 44.1 kHz clock output
ATIPSYC	26	O	ATIPSync output
PANIC	27	I	laser low power (LLP)
PORE	28	I	power-on reset
CL1	29	O	divided clock output
V_{DDD}	30, 49, 53, 76	supply	core and pad ring 3 V supply voltage; note 2
V_{SSD}	31, 48, 52, 75	supply	core and pad ring ground
SCL	32	I	sub-CPU clock
SDA	33	I/O	bidirectional sub-CPU data
INT	34	O	sub-CPU interrupt
DA7	35	I/O	bidirectional sub-CPU parallel data bus
DA6	36	I/O	bidirectional sub-CPU parallel data bus
DA5	37	I/O	bidirectional sub-CPU parallel data bus
DA4	38	I/O	bidirectional sub-CPU parallel data bus

SYMBOL	PIN	TYPE	DESCRIPTION
DA3	39	I/O	bidirectional sub-CPU parallel data bus
DA2	40	I/O	bidirectional sub-CPU parallel data bus
DA1	41	I/O	bidirectional sub-CPU parallel data bus
DA0	42	I/O	bidirectional sub-CPU parallel data bus
V_{DDE}	44	supply	output driver 3 V supply voltage
$\overline{WRi}$	45	I	sub-CPU write enable; active LOW
$\overline{RDi}$	46	I	sub-CPU read enable; active LOW
ALE	47	I	sub-CPU address latch enable
CSi	50	I	sub-CPU chip select
PCAin	51	I	PCA input
STOPCK	54	O	stop clock output
V4	55	O	serial subcode output
EBUOUT	56	O	digital output
SYNC	57	O	I^2S sector sync output
FLAG	58	O	I^2S correction flag
DATAO	59	O	I^2S data output
WCLK	60	I/O	bidirectional I^2S word clock
BCLK	61	I/O	bidirectional I^2S bit clock
V_{DDE}	63	supply	output driver 3 V supply voltage
DATAI	64	I	I^2S data input
SUB	65	I	EIAJ subcode data
RCK	66	O	EIAJ subcode clock
SFSY	67	I	EIAJ subcode sync
CFLG	68	O	correction statistics; open-drain
T2	69	I	tacho control input 2
T1	70	I	tacho control input 1
V_{DDE}	72	supply	output driver 3 V supply voltage
MOTO2/T3	73	I/O	motor output 2/tacho input 3
MOTO1	74	O	motor control output 1
LASERON	77	O	laser write control
XEFM	78	O	EFM clock output
EFMDATA	79	O	EFM data output
MEAS1	80	O	front end telemetry; open-drain

Notes

1. No signal may be applied to this device when it is not powered.
2. The analog and digital supply pins (V_{DDA} and V_{DDD}) must be connected to the same external supply.

7 FUNCTIONAL DESCRIPTION

7.1 Microprocessor interfaces

The SAA7392 is programmed via two independent microprocessor interfaces:

- Serial I²C-bus
  - SDA = I²C-bus data
  - SCL = I²C-bus clock
  - I²C-bus write address = 3EH
  - I²C-bus read address = 3FH.
- Parallel 80C51 compatible
  - DA(7:0) = address/data bus
  - ALE = address latch enable; latches the address information on the bus
  - $\overline{WRi}$ = active LOW write signal; write to SAA7392
  - $\overline{RDi}$ = active LOW read signal; read from SAA7392
  - CSi = chip select signal; gates the $\overline{RDi}$ and $\overline{WRi}$ signals.

7.1.1 SERIAL I²C-BUS INTERFACE

Data is transferred over the interface in single bytes, via write data or read data commands.

The sequence for a write data command is as follows:

1. Send START condition
2. Send address 3EH (write)
3. Write register address byte
4. Write data byte
5. Send STOP condition.

The sequence for a read data command is as follows:

1. Send START condition
2. Send address 3EH (write)
3. Write status register address byte
4. Send STOP condition
5. Send address 3FH (read)
6. Read data byte
7. Send STOP condition.

7.1.2 PARALLEL INTERFACE

The parallel interface has a multiplexed address/data bus. Information can be written to or read from the SAA7392 using the protocols shown in Figs 3 and 4; specific timings are shown in Table 2. Note that only the lower six address bits are decoded; so writing to address 40H would have the same effect as writing to address 00H.

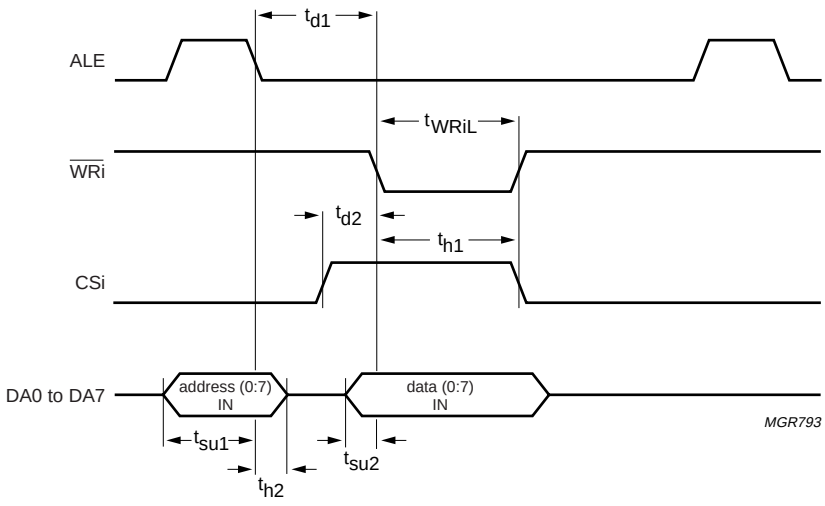


Fig.3 Microprocessor write protocol.

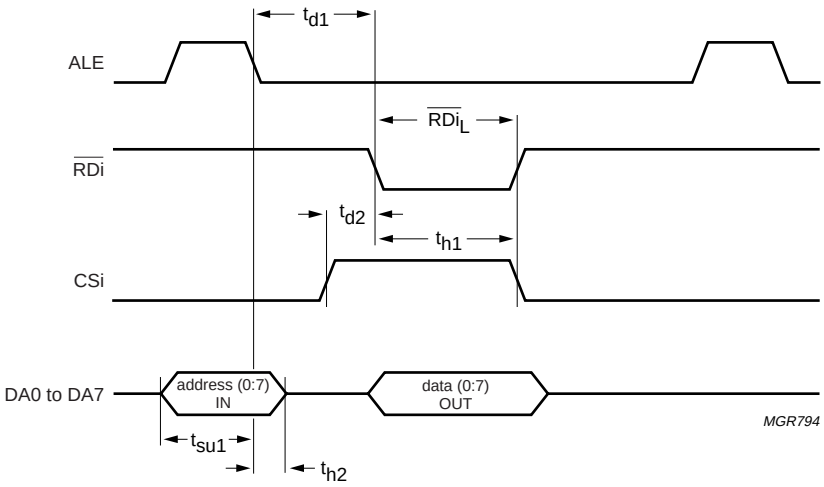


Fig.4 Microprocessor read protocol.

Table 2 Parallel interface timing

SYMBOL	DESCRIPTION	MIN.(1)	MAX.(1)	UNIT
t_{d1}	Delay ALE falling to $\overline{RDi}$/$\overline{WRi}$ falling.	17	–	ns
t_{d2}	Delay CSi rising to $\overline{RDi}$/$\overline{WRi}$ falling.	17	–	ns
t_{h1}	CSi hold time after $\overline{RDi}$/$\overline{WRi}$ falling.	$2T_{clk}$ + 17	–	ns
t_{su1}	Address setup time before ALE falling.	17	–	ns
t_{h2}	Address hold time after ALE falling.	17	–	ns
t_{su2}	Data setup time before $\overline{WRi}$ falling.	0	–	ns
t_{h3}	Data hold time after $\overline{WRi}$ falling.	$2T_{clk}$ + 17	–	ns
t_{WRiL}	$\overline{WRi}$ LOW time.	$1T_{clk}$ + 17	–	ns
t_{h4}	ALE LOW hold time after $\overline{WRi}$ LOW.	$3T_{clk}$ + 17	–	ns
t_{d3}	Delay data valid after $\overline{RDi}$ LOW.	–	$3T_{clk}$ + 17	ns
t_{d4}	Delay $\overline{RDi}$ HIGH to data out high-impedance.	–	17	ns
t_{RDiL}	$\overline{RDi}$ LOW time.	$3T_{clk}$ + 128	–	ns

Note

1. T_{clk} is the system clock period.

7.2 Register map

Table 3 Register map

ADDRESS (HEX)	REGISTER NAME	TYPE	FUNCTION	BLOCK RESPONSIBLE
00	PLL Lock Select Register (PLLLock)	Write	PLL lock select	bit detector
		Read	8-bit PLL frequency	bit detector
01	PLL Bandwidth Select Register (PLLSet)	Write	PLL bandwidth select	bit detector
		Read	8-bit asymmetry signal	bit detector
02	PLL Frequency Preset Register (PLLFreq)	Write	PLL frequency preset	bit detector
		Read	8-bit jitter signal	bit detector
03	PLL Equalizer Preset Register (PLLEqu)	Write	PLL equalizer preset	bit detector
		Read	Observe internal lock flags	bit detector
04	PLL Lock Aid2 Preset Register (PLLFMeas)	Write	PLL lock aid 2 preset	bit detector
05	I^2S Output Register 1 (Output1)	Write	I^2S output 1	serial out
06	I^2S Output Register 2 (Output2)	Write	I^2S output 2	serial out
07	I^2S Output Register 3 (Output3)	Write	I^2S output 3	serial out
08	Semaphore Register 1 (Sema1)	Write/Read	Inter-microprocessor communication	sub-CPU
09	Semaphore Register 2 (Sema2)	Write/Read	Inter-microprocessor communication	sub-CPU
0A	Semaphore Register 3 (Sema3)	Write/Read	Inter-microprocessor communication	sub-CPU
0B	Interrupt Enable Register (IntEn)	Write	Enable interrupts	sub-CPU
0B	Status Register (Status)	Read	Interrupt status	sub-CPU
0C	Motor Control Register 1 (Motor1)	Write	Frequency set-point	motor/tacho
		Read	8-bit slicer compensation value	bit detector
0D	Motor Mode Select Register 2 (Motor2)	Write	Motor coefficient preset	motor/tacho
		Read	Opening of eye pattern	bit detector
0E	Motor Control Register 3 (Motor3)	Write	Motor integrator preset	motor/tacho
		Read	Read back of motor frequency	motor/tacho
0F	Motor Control Register 4 (Motor4)	Write	Motor control	motor/tacho
10	Motor Control Register 5 (Motor5)	Read/Write	Motor integrator value	motor/tacho
11	Motor Control Register 6 (Motor6)	Read/Write	Motor integrator value	motor/tacho

ADDRESS (HEX)	REGISTER NAME	TYPE	FUNCTION	BLOCK RESPONSIBLE
12	Clock Preset Register (ClockPre)	Write	Clock control	clock generator
		Read	Status of Q-channel subcode	encoder/decoder
13	Decoder Mode Select Register (DecoMode)	Write	Decoder mode select	encoder/decoder
		Read	Q-channel subcode data	encoder/decoder
14	Subcode Read End Register (SubReadEnd)	Read	Subcode data read finished	encoder/decoder
15	Analog Settings Register 1 (AnaSet1)	Write	Analog control	analog
		Read	C1 frames in FIFO + offset	encoder/decoder
16	Viterbi Detector Settings Register (VitSet)	Write	Viterbi detector control	bit detector
17	Tacho Gain Setting Register (Tacho1)	Write	Tacho gain setting	motor/tacho
18	Tacho Trip Setting Register (Tacho2)	Write	Tacho trip setting	motor/tacho
19	Tacho Control Register (Tacho3)	Write	Tacho control settings	motor/tacho
1B	Soft Reset Register (SoftReset)	Write	Sub-block reset	sub-CPU
1D	Motor Control Register 7 (Motor7)	Write	Control coefficients select	motor/tacho
1E	Input Configuration Register (InputConfig)	Write	EBU clock frequency and input format	serial input
20	Status Register 2 (Status2)	Read/Write	Interrupt status	sub-CPU
21	Interrupt Enable Register 2 (IntEn2)	Write	Enable interrupts	sub-CPU
22	Subcode Preset Count Register (SubPresetCount)	Write	Preset count field	subcode insert
		Read	Current count field	subcode insert
23	Subcode Configuration Register 1 (SubConfig1)	Write	Subcode control	subcode insert
24	Subcode Configuration Register 2 (SubConfig2)	Read/Write	Subcode control	subcode insert
25	Subcode Start Data Register (SubStartData)	Write	Subcode control	subcode insert
26	Subcode Data Register (SubData)	Read/Write	Subcode data	subcode insert
27	Wobble Configuration Register 1 (WobbleConfig1)		Integrator and loop bandwidth	Wobble processor
			Window width ATIP syncs	Wobble processor
28	Wobble Configuration Register 2 (WobbleConfig2)	Write	Wobble PLL control	Wobble processor
29	ATIP Status Register (ATIPStatus)	Read	ATIP status	Wobble processor
2A	Wobble Frequency Register 1 (WobbleFreq1)	Read/Write	8 MSBs of PLL frequency	Wobble processor
2B	Wobble Frequency Register 2 (WobbleFreq2)	Read/Write	8 LSBs of PLL frequency	Wobble processor
2C	ATIP Data Register (ATIPData)	Read	ATIP data	Wobble processor

ADDRESS (HEX)	REGISTER NAME	TYPE	FUNCTION	BLOCK RESPONSIBLE
2D	ATIP Data End Register (ATIPDataEnd)	Read	Least significant byte ATIP data	Wobble processor
2E	Wobble Peak Status Register (WobbleStatus)	Read	Peak value of wobble signal	Wobble processor
30	Encode WriteOn Control Register (EncodeWContr)	Read/Write	Laser and data flow control	encode control
31	Encode Start Offset Register (EncodeStartOffset)	Write	Start WriteOn flags delay	encode control
32	Encode Stop Offset Register (EncodeStopOffset)	Write	Stop WriteOn flags delay	encode control
33	Encode Offset Register (EncodeXOffset)	Write	10-bit value for Xoffset	encode control
34	EFM Clock Configuration Register 1 (EFMClockConf1)	Write	EFM clock control	EFM clock generator
35	EFM Clock Configuration Register 2 (EFMClockConf2)	Write	EFM clock control	EFM clock generator
36	EFM Clock Configuration Register 3 (EFMClockConf3)	Write	EFM clock control	EFM clock generator
		Read	Integrator value	EFM clock generator
37	EFM PLL Frequency Register (EFMPLLFreq)	Read	EFM PLL frequency	EFM clock generator
37	EFM Clock Configuration Register 4 (EFMClockConf4)	Write	EFM clock control	EFM clock generator
38	ATIP Error Register (ATER)	Read	Counter for ATIP CRC errors	sub-CPU
39	C1 Block Error Register (C1BLER)	Read	Counter for C1 errors	sub-CPU
3A	C2 Block Error Register (C2BLER)	Read	Counter for C2 errors	sub-CPU
3C	EFM Preset Count Register (EFMPresetCount)	Write	EFM frame position for output	EFM modulator
3D	EFM Modulator Configuration Register (EFMModConfig)	Write	XEFM control and output data format	EFM modulator
3E	EFM Modulator Configuration Register 2 (EFMModConfig2)	Write	XEFM control and output data format	EFM modulator

7.2.1 Interrupt pin

The interrupt pin (INT) is the AND-OR-INVERT of the Status and Interrupt Enable Registers, i.e. INT will become active when corresponding bits are set at the same time in the Status and Interrupt Enable Registers.

7.2.2 The Semaphore Registers (Sema1, Sema2 and sema3)

The Semaphore Registers are intended for inter-microprocessor communications. For example, microcontroller 1 can write data to microcontroller 2 via Sema1 and microcontroller 2 can write data to microcontroller 1 via Sema2. The Status Register of the SAA7392 offers a mechanism so that both microcontrollers can see when new data has been written and when it has been read by looking at the contents of the Semaphore Registers. Version M3 of the CDR60 can be identified by writing and reading register Sema3. In version M3, bit 1 of Sema3 is always read as logic 0, whereas in other CDR60 versions this bit reads the same value as what was written to it before.

7.2.2.1 Semaphore Register 1 (Sema1)

Table 4 Semaphore Register 1 (address 08H) - READ/WRITE

7	6	5	4	3	2	1	0
Sema1.7	Sema1.6	Sema1.5	Sema1.4	Sema1.3	Sema1.2	Sema1.1	Sema1.0

7.2.2.2 Semaphore Register 2 (Sema2)

Table 5 Semaphore Register 2 (address 09H) - READ/WRITE

7	6	5	4	3	2	1	0
Sema2.7	Sema2.6	Sema2.5	Sema2.4	Sema2.3	Sema2.2	Sema2.1	Sema2.0

7.2.2.3 Semaphore Register 3 (Sema3)

Table 6 Semaphore Register 3 (address 0AH) - READ/WRITE

7	6	5	4	3	2	1	0
Sema3.7	Sema3.6	Sema3.5	Sema3.4	Sema3.3	Sema3.2	Sema3.1	Sema3.0

7.2.3 Status Register (Status)

Table 7 Status Register (address 0BH) - READ

7	6	5	4	3	2	1	0
Sema1	Sema2	Sema3	LockIn	HeaderVal	MotorOverflow	FIFOOv	–

Table 8 Description of Status bits

BIT	SYMBOL	DESCRIPTION
7	Sema1	If Sema1 = 1, change in register Sema1 has been detected. Reset if register Sema1 read.
6	Sema2	If Sema2 = 1, change in register Sema2 has been detected. Reset if register Sema2 read.
5	Sema3	If Sema3 = 1, change in register Sema3 has been detected. Reset if register Sema3 read.
4	LockIn	If LockIn = 1, then channel data PLL in lock (not latched).
3	HeaderVal	HeaderVal is set when new header/subcode is available; reset on reading SubReadEnd.
2	MotorOverflow	If MotorOverflow = 1, then a motor overflow is occurring (not latched).
1	FIFOOv	If FIFOOv = 1, then the FIFO has overflowed.
0	–	This bit is reserved.

7.2.4 Interrupt Enable Register (IntEn)

Table 9 Interrupt Enable Register (address 0BH) - WRITE

7	6	5	4	3	2	1	0
Sema1En	Sema2En	Sema3En	LockInEn	HeaderValen	MotorOverflowEn	FIFOOvEn	–

Table 10 Description of IntEn bits

BIT	SYMBOL	DESCRIPTION
7	Sema1En	If Sema1En = 1, then Semaphore Register 1 interrupt is enabled.
6	Sema2En	If Sema2En = 1, then Semaphore Register 2 interrupt is enabled.
5	Sema3En	If Sema3En = 1, then Semaphore Register 3 interrupt is enabled.
4	LockInEn	If LockinEn = 1, then channel data PLL in lock interrupt is enabled.
3	HeaderValEn	If HeaderValEn = 1, then new header/subcode available interrupt is enabled.
2	MotorOverflowEn	If MotorOverflowEn = 1, then motor overflow interrupt is enabled.
1	FIFOOvEn	If FIFOOvEn = 1, then FIFO overflow interrupt is enabled.
0	–	This bit is reserved.

7.2.5 Status Register 2 (Status2)

Table 11 Status Register 2 (address 20H) - READ/WRITE

7	6	5	4	3	2	1	0
BankSwitch	SyncError	DataNotValid	QSync	ATIPSync	LaserOn	LaserOff	XErrorLarge

Table 12 Description of Status2 bits

BIT	SYMBOL	DESCRIPTION
7	BankSwitch	When set a ‘Bank switch’ in the subcode insert block has occurred; reset when a logic 1 is written to this bit.
6	SyncError	When set synchronisation with PLUM on subcode transfer has failed; reset when a logic 1 is written to this bit.
5	DataNotValid	When set an under-run on subcode transfer with PLUM has occurred; reset when a logic 1 is written to this bit.
4	QSync	When set a Q-channel subcode sync has been written to disc; reset when a logic 1 is written to this bit.
3	ATIPSync	When set sync has been found in the ATIP channel; reset when a logic 1 is written to this bit.
2	LaserOn	When set a rising edge of the internal LaserOn signal has occurred; reset when a logic 1 is written to this bit.
1	LaserOff	When set a falling edge of the internal LaserOn signal has occurred; reset when a logic 1 is written to this bit.
0	XErrorLarge	When set the offset between QSync and ATIPSync is more than 2 EFM frames different from the programmed value.

7.2.6 Interrupt Enable Register 2 (IntEn2)

Table 13 Interrupt Enable Register 2 (address 21H) - WRITE

7	6	5	4	3	2	1	0
BankSwitch En	SyncErrorEn	DataNotValid En	QSyncEn	ATIPSyncEn	LaserOnEn	LaserOffEn	XErrorLarge En

Table 14 Description of IntEn2 bits

BIT	SYMBOL	DESCRIPTION
7	BankSwitch En	If BankSwitchEn = 1, then BankSwitch interrupt is enabled.
6	SyncErrorEn	If SyncErrorEn = 1, then SyncError interrupt is enabled.
5	DataNotVali dEn	If DataNotValidEn = 1, then DataNotValid interrupt is enabled.
4	QSyncEn	If QSyncEn = 1, then QSync interrupt is enabled.
3	ATIPSyncEn	If ATIPSyncEn = 1, then ATIPSync interrupt is enabled.
2	LaserOnEn	If LaserOnEn = 1, then LaserOn interrupt is enabled.
1	LaserOffEn	If LaserOffEn = 1, then LaserOff interrupt is enabled.
0	XErrorLarge En	If XerrorLarge = 1, then XErrorLarge interrupt is enabled.

7.2.7 Soft Reset Register (SoftReset)

Table 15 Soft Reset Register (address 1BH) - WRITE

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	SReset1

Table 16 Description of SoftReset bits

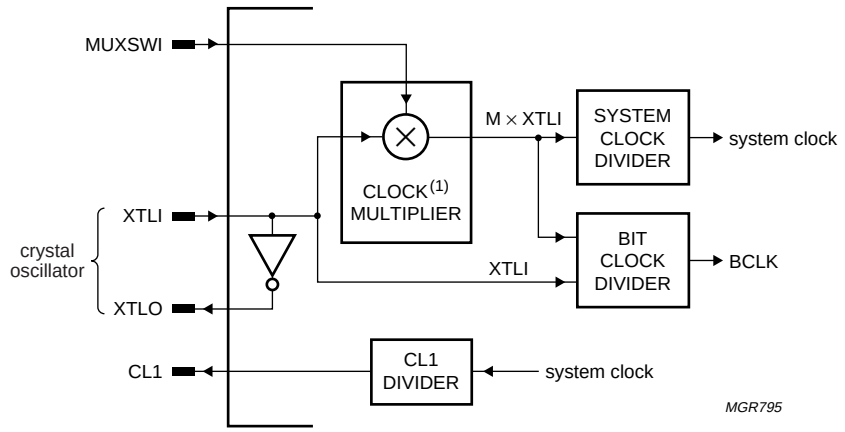
BIT	SYMBOL	DESCRIPTION
7 to 1	–	These 7 bits are reserved.
0	SReset1	When set, synchronisation with PLUM on subcode transfer has failed; reset when a logic 1 is written to this bit (Status2). This bit is an active HIGH reset to the following blocks: Encoder/decoder, EFM modulator, Encode control block, Serial input/output block and Encode subcode insert block. The clock control, EFM PLL, tacho, motor interface and wobble interface remain running. Soft reset will reset the following registers: EFMPresetCount, EFMModulateConfig, EFMModulateConfig2, EncodeXOffset, EncodeWriteControl, EncodeStartOffset, EncodeStopOffset, SubPresetCount, SubConfig1, Subconfig2, SubStartData, SubData, InputConfig, DecoMode, Output1, Output2 and Output3. A soft reset is mandatory in the following cases: 1. After programming the BCLK clock 2. When switching from encode to decode 3. When switching from decode to encode.

7.3 System clocks

The principle clocks used in the SAA7392 are derived from the crystal oscillator input pin XTLI (alternatively, an external clock can be connected to this pin). These clocks are the system clock (also used as the ADC clock) and the I^2S output bit clock (BCLK).

The system clock (f_{clk}) defines the maximum operational channel rate for the device. The maximum EFM channel clock is twice the system clock, for CD it is equivalent to system clock/(4.3×10^6) which is approximately $11.5 \times$ CDROM for a 25 MHz system clock.

The other clock in the system is the channel data clock, this is recovered by the front-end bit recovery PLL.



(1) M = 1 if MUXSWI is LOW; M = 8 if MUXSWI is HIGH.

Fig.5 System clock generator.

7.3.1 Clock Preset Register (ClockPre)

Table 17 Clock Preset Register (address 12H) - WRITE

7	6	5	4	3	2	1	0
CL1Div	GateBClk	Div.1	Div.0	Mux2	Div2.2	Div2.1	Div2.0

Table 18 Description of ClockPre bits

BIT	SYMBOL	DESCRIPTION
7	CL1Div	If CL1Div = 0, then CL1 output frequency is $\frac{1}{3}f_{clk}$. If CL1Div = 1, then CL1 output frequency is $\frac{1}{2}f_{clk}$.
6	GateBClk	If GateBClk = 0, then I^2S output bit clock gating is disabled. If GateBClk = 1, then I^2S output bit clock gating enabled, BCLK is output, clock is automatically stopped if FIFO underflows (this is known as Flow control mode).
5	Div.1	These 2 bits select the system clock frequency (f_{clk}); see Table 19. This frequency should be programmed for the expected disc channel rate (e.g. 4.33 MHz for $1 \times$ CD) within the following constraints: $\frac{\text{Channel rate}}{2} < f_{clk} < 4 \times \text{Channel rate}$ In this clock range, reliable bit detection is possible. All data found will be written to the FIFO. It is the responsibility of the user to select system clock values so that the FIFO performance is controlled.
4	Div.0	
3	Mux2	If Mux2 = 0, then N (bit clock divider pre-scaler) = 1. If Mux2 = 1, then N = M.
2 to 0	Div2<2:0>	These 3 bits select the BCLK frequency (f_{BCLK}); see Table 20. It is the responsibility of the user to select BCLK values so that the FIFO performance is controlled.

Table 19 Selection of system clock frequency

Div.1	Div.0	SYSTEM CLOCK FREQUENCY (f_{clk})
0	0	$M \times f_{XTLI}$
0	1	$0.5 \times M \times f_{XTLI}$
1	0	$0.25 \times M \times f_{XTLI}$
1	1	$0.125 \times M \times f_{XTLI}$

Table 20 Selection of BCLK frequency

Div2.1	Div2.1	Div2.0	BCLK FREQUENCY (f_{BCLK})
0	0	0	$N \times f_{XTLI}$
0	0	1	$N \times f_{XTLI}$
0	1	0	$\frac{1}{2}(N \times f_{XTLI})$
0	1	1	$\frac{1}{3}(N \times f_{XTLI})$
1	0	0	$\frac{1}{4}(N \times f_{XTLI})$
1	0	1	$\frac{1}{6}(N \times f_{XTLI})$
1	1	0	$\frac{1}{8}(N \times f_{XTLI})$
1	1	1	$\frac{1}{12}(N \times f_{XTLI})$

7.4 HF analog front-end

The HF ADC in the SAA7392 encodes the EFM high frequency signal from the disc light pen assembly. These signals are pre-processed, externally to the SAA7392, by either AEGER-2 or a DALAS equivalent. The dynamic range of the ADC is optimized by the inclusion of an AC coupled AGC function under digital control.

In order to make use of the whole digital front-end resolution, the output of the gain control amplifier should constantly deliver 1.4 $V_{(p-p)}$ output signal. The gain range of the ADC is approximately 14 dB, with 32 steps. The gain control for the variable gain amplifier is controlled by an on-chip digital gain control block (AGC). This block allows for both automatic and microprocessor gain control. The gain control block will detect ADC extreme conditions (00H or FFH outputs); on these values the gain control block will decrement the gain. If no extreme codes occur the gain is incremented.

7.4.1 Fixed gain

Control of the gain is as follows:

1. Writing XX1X XXXX to the Anaset1 register (address 15H) increases the AGC gain by 1.1 dB
2. Writing XX0X XXXX to the AnaSet1 register (address 15H) decreases the AGC gain by 1.1 dB
3. Instructions to increment/decrement gain are ignored when the AGC gain limits of –4/+12 dB are reached.

7.4.2 Automatic Gain Control (AGC)

The gain of the AGC cell is adjusted until the analog signal at the ADC input extends over the complete range of the ADC. Detection of this condition is in the digital domain where the maximum and minimum ADC codes are measured. The dynamics of the AGC system are as follows.

1. If the ADC output codes are not full scale (i.e. 000 0000 and 111 11111) the AGC gain is incremented in 1.1 dB steps with a time constant of 1000/n µs, where n is the over-speed factor i.e. n = 1 for basic audio CD.
2. When full scale is detected at the output of the ADC the AGC gain is fixed provided that full scale is maintained and clipping does not occur for greater than 20% of the time.
3. If clipping occurs for more than 20% of the time, then the AGC gain is reduced in 1.1 dB steps with a time constant of 60/n µs.

The ADC and AGC electrical characteristics are specified in Chapter 9.

7.4.3 Analog Settings Register 1 (AnaSet1)

Table 21 Analog Settings Register 1 (address 15H) - WRITE

7	6	5	4	3	2	1	0
GainControl	MaxGain	StepUp	StepDown	PowerDown	–	–	–

Table 22 Description of AnaSet1 bits

BIT	SYMBOL	DESCRIPTION
7	GainControl	If GainControl = 0, then gain control is in Hold mode. If GainControl = 1, then automatic gain control is on.
6	MaxGain	If MaxGain = 0, then there is no gain limit. If MaxGain = 1, then the maximum gain is 7.66 dB.
5	StepUp	If StepUp = 1, then step up gain by one LSB.
4	StepDown	If StepDown = 1, then step down gain by one LSB.
3	PowerDown	If PowerDown = 0, then analog blocks are powered up. If PowerDown = 1, then analog blocks are powered down.
2 to 0	–	These 3 bits are reserved and must be set to a logic 0s.

7.5 Bit recovery

The bit recovery block (shown in Fig.6) contains the slice level circuitry, a noise filter to limit the HF-EFM signal noise contribution, an adaptive slicer circuit and a digital PLL. These blocks can be controlled via the microprocessor.

The channel rate should always obey the following constraints:

- It should be less than 2 × the system clock
- It should be greater than 0.25 × the system clock.

In this clock range reliable bit clock detection is possible. All data found will be written to the FIFO. It is the responsibility of the user to select BCLK and system clock values so that the FIFO operation is controlled.

The digital noise filter runs on the PLL bit clock and limits the bandwidth of the incoming signal to 0.25 of the PLL bit clock frequency. The characteristics of the filter are:

- Passband: 0 to 0.22 f_b
- Stopband: 0.28 f_b to (f_{clk} – 0.28 f_b)
- Rejection: –28 dB.

The slice level determination circuit compensates the incoming signal asymmetry component. The bandwidth of this circuit is programmable via register PLLSet.

A programmable (one tap presetable, asymmetrical) equaliser is used in the bit detection circuit. The first and last tap settings are different. Possible tap values are settable via register PLLEqu.

The advanced detector has two extra detection circuits (adaptive slicer and run length 2 push-back) which are controlled via the VitSet register, that allow improved margin in the bit detector.

The adaptive slicer does a second stage slice operation; the bandwidth is higher than the first slicer. It can be turned on/off via the VitSet register.

If the advanced detector is switched on all run length 2 symbols are pushed back to run length 3. The circuit will determine the transition that was most likely to be in error, and shift the transition on that edge.

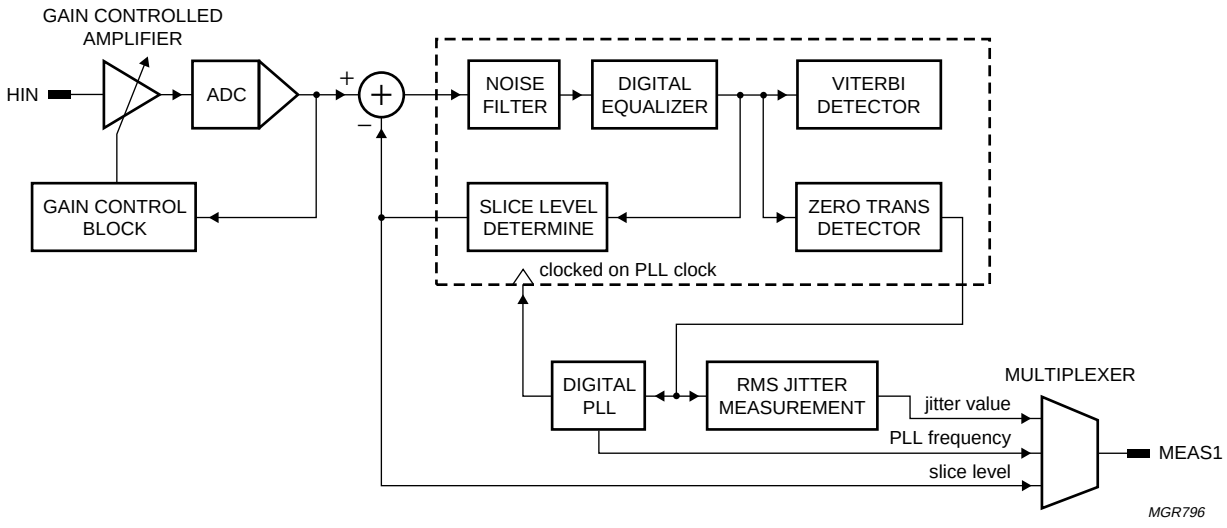


Fig.6 Block diagram of bit recovery block.

7.5.1 DIGITAL PLL

The digital PLL will recover the channel bit clock. As the capture range of the PLL itself is limited, lock detectors and 2 capture aids are present. In total three different PLL operation modes exist: In-lock, Inner-lock aid and Outer-lock aid.

The PLL behaviour during in-lock (the normal on-track situation) can be best explained in the frequency domain. The PLL operation is completely linear during in-lock situations. The open-loop response of the PLL is given in Fig.7. The three frequencies, f_0 (integrator cross-over frequency), f_1 (PLL bandwidth) and f_2 (low-pass bandwidth) are programmable via register PLLSet.

To extend the PLL capture range two lock aids are used:

- Inner lock aid: has a capture range of ±10% and will bring the PLL frequency to the lock point
- Outer lock range: has no limitation on capture range, and will bring the PLL within the range of the inner lock range.

Two outer lock aids can be used:

- Run length 3 deviation detector: this circuit is known to be sensitive to systematic over/under equalization; this over/under equalization can be counter-acted by writing a non-zero phase offset value to register PLLLock.
- Frequency measurement detector: this circuit regulates the PLL frequency so that the average number of EFM transitions is a fixed fraction of the PLL bit clock; the transition frequency is settable via register PLLFMeas.

Programmability/observability is built into the PLL. Its operation can be influenced in two ways:

- It is possible to select the state the PLL is in (in-lock, near-lock, outer-lock) via register PLLLock
- It is possible to preset the PLL frequency to a certain value via registers PLLEqu and PLLFreq.

The operation of the bit detector can be monitored by the microprocessor and via the MEAS1 pin. Four signals are available for measurement:

- PLL frequency signal: the most significant 8 bits are available via register PLLLock
- Asymmetry signal: the 8-bit signal in 2's complement form is available via register PLLSet
- Jitter signal: the most significant 8 bits are available via register PLLFreq. This gives an impression of the detection jitter after all processing is done.

  jitter<9:0> = average ((jitter individual transition)$^2 \times 8192$)

  To obtain the jitter in the bit clocks the jitter<9:0> value must be divided by 8192 and square routed. Note that the jitter<9:0> overestimates the jitter (by approximately rms jitter increase of 0.03 bit clock), because the quantization of the zero transitions is in 4 intervals.

  Note the jitter is measured before the bit detection and contains contributions due to various imperfections in the complete signal path; i.e. disc, preamplifier, ADC, limited bitwidths, PLL performance, internal filter noise, asymmetry compensation, equalizer.
- Internal lock flags: The internally generated inner-lock signal (f_lock_in), lock signal (lock_in) and flag that indicates when a run length 14 is detected (long_symbol) are available via register PLLEqu.

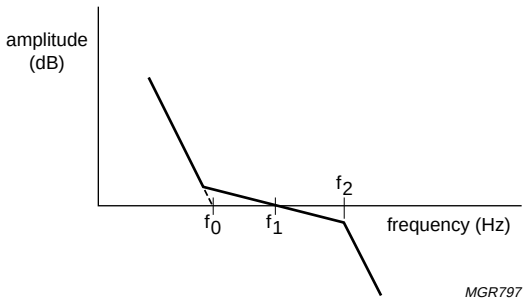


Fig.7 PLL bode diagram.

7.5.2 MEAS1 PIN

The MEAS1 pin carries the 3 measurement signals: jitter (sampled twice), PLL frequency, and asymmetry. Each frame consists of 64 bits (each 4 system clock periods long), beginning with a start bit, then data bits then pause bits (see Fig.8). The start bit is always preceded by 17 pause bits; and the intermediate start bits at locations 12, 24 and 36 guarantee that no other ‘1’ bit is preceded by 17 ‘0’ bits, making the start detection easy. The structure of the frame is described in Table 23 and shown in Fig.8.

Table 23 Frame structure

BIT	VALUE	FUNCTION
0	logic 1	start bit
1 to 10	jitter<9:0>	jitter word
11	logic 0	
12	logic 1	intermediate start bit
13 to 22	pllfreq<9:0>	PLL frequency word
23	logic 0	
24	logic 1	intermediate start bit
25 to 32	assym<7:0>	asymmetry word
33	logic 0	
34	logic 1	intermediate start bit
37 to 46	jitter<9:0>	second sample of jitter word
47 to 63	logic 0	pause

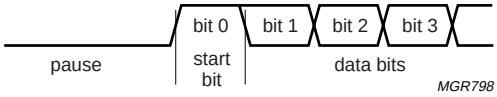


Fig.8 Format on MEAS1 pin.

7.5.3 PLL Lock Select Register (PLLLock)

The behaviour of this register is dependent upon whether its being read or written. The behaviour for the write operation is described in Tables 24 to 27. When read the 8 MSBs of the PLL frequency counter are returned; this is described in Tables 24 and 28.

Table 24 PLL Lock Select Register (address 00H) - WRITE/READ

7	6	5	4	3	2	1	0
LockOride	PhaOset.2	PhaOset.1	PhaOset.0	PLLForceL.3	PLLForceL.2	PLLForceL.1	PLLForceL.0
PLLFreq.7	PLLFreq.6	PLLFreq.5	PLLFreq.4	PLLFreq.3	PLLFreq.2	PLLFreq.1	PLLFreq.0

Table 25 Description of PLLLock bits for write operation

BIT	SYMBOL	DESCRIPTION
7	LockOride	When LockOride = 0, then automatic lock behaviour selected, PLLForceL<3:0> must be set to ‘0000’. When LockOride = 1, then PLL manual override, PLLForceL<3:0> must also be programmed.
6	PhaOset.2	These 3 bits are used to select the phase override settings; see Table 26.
5	PhaOset.1	
4	PhaOset.0	
3	PLLForceL.3	These 4 bits are used to select the PLL lock; see Table 27.
2	PLLForceL.2	
1	PLLForceL.1	
0	PLLForceL.0	

Table 26 Selection of phase override setting

PhaOset.2	PhaOset.1	PhaOset.0	PHASE OVERRIDE
0	0	0	reserved
0	0	1	$^{3}/_{8} \times$ PLL clock over-equalized T3
0	1	0	$^{2}/_{8} \times$ PLL clock over-equalized T3
0	1	1	$^{1}/_{8} \times$ PLL clock over-equalized T3
1	0	0	correct equalisation
1	0	1	$^{1}/_{8} \times$ PLL clock under-equalized T3
1	1	0	$^{2}/_{8} \times$ PLL clock under-equalized T3
1	1	1	$^{3}/_{8} \times$ PLL clock under-equalized T3

Table 27 Selection of PLL lock

PLLForceL.3	PLLForceL.2	PLLForceL.1	PLLForceL.0	PLL LOCK
0	0	0	0	automatic lock behaviour
0	0	0	1	force PLL in-lock
0	1	0	0	force PLL into outer-lock
0	1	1	0	force PLL into inner-lock
1	0	0	0	force PLL into Hold mode (PLL frequency can be forced using preset value in register PLLFreq)
X	X	X	X	all other combinations are reserved

Table 28 Description of PLLock bits for read operation

BIT	SYMBOL	DESCRIPTION
7 to 0	PLLFreq<7:0>	This register holds the 8 MSBs of the PLL frequency counter. The PLL frequency is calculated as shown below: $f_{PLL}(Hz) = \frac{(PLLFreq<7:0> \times ADC\ clock\ (Hz))}{128}$

7.5.4 PLL Bandwidth Select Register (PLLSet)

The function of this register is dependent upon whether its being read or written. The function for the write operation is described in Tables 29 to 34. Note the measurement conditions are: system clock = 2.15 MHz, bit clock = 4.3 MHz, bandwidth is proportional to the system clock.

When read this register returns the 8-bit PLL asymmetry value, see Table 29.

Table 29 PLL Bandwidth Select Register (address 01H) - WRITE/READ

7	6	5	4	3	2	1	0
SliceBW.1	SliceBW.0	IntegF0.1	IntegF0.0	PLLBWF1.1	PLLBWF1.0	LPBWF2.1	LPBWF2.0
PLLAsym.7	PLLAsym.6	PLLAsym.5	PLLAsym.4	PLLAsym.3	PLLAsym.2	PLLAsym.1	PLLAsym.0

Table 30 Description of PLLSet bits for write operation

BIT	SYMBOL	DESCRIPTION
7	SliceBW.1	These 2 bits select the Slicer bandwidth; see Table 31.
6	SliceBW.0	
5	IntegF0.1	These 2 bits select the integrator crossover frequency; see Table 32.
4	IntegF0.0	
3	PLLBWF1.1	These 2 bits select the PLL bandwidth; see Table 33.
2	PLLBWF1.0	
1	LPBWF2.1	These 2 bits select the low-pass bandwidth; see Table 34.
0	LPBWF2.0	

Table 31 Selection of Slicer bandwidth

SliceBW.1	SliceBW.0	SLICER BANDWIDTH
0	0	12 Hz
0	1	50 Hz
1	0	200 Hz
1	1	This value is reserved.

Table 32 Selection of integrator crossover frequency

IntegFO.1	IntegFO.0	INTEGRATOR CROSSOVER FREQUENCY
0	0	3780 Hz
0	1	1890 Hz
1	0	945 Hz
1	1	This value is reserved.

Table 33 Selection of PLL bandwidth

PLLBWF1.1	PLLBWF1.0	PLL BANDWIDTH
0	0	21000 Hz
0	1	10528 Hz
1	0	5264 Hz
1	1	2632 Hz

Table 34 Selection of low-pass bandwidth

LPBWF2.1	LPBWF2.0	LOW-PASS BANDWIDTH
0	0	42100 Hz
0	1	21000 Hz
1	0	10528 Hz
1	1	This value is reserved.

7.5.5 PLL Frequency Preset Register (PLLFreq)

The function of this register is dependent upon whether its being read or written. Tables 35 and 36 define the register function for the write operation. Tables 35 and 37 define the register function for the read operation.

Table 35 PLL Frequency Preset Register (address 02H) - WRITE/READ

7	6	5	4	3	2	1	0
PLLFreq.9	PLLFreq.8	PLLFreq.7	PLLFreq.6	PLLFreq.5	PLLFreq.4	PLLFreq.3	PLLFreq.2
JV.7	JV.6	JV.5	JV.4	JV.3	JV.2	JV.1	JV.0

Table 36 Description of PLLFreq bits for write operation

BIT	SYMBOL	DESCRIPTION
7 to 0	PLLFreq<9:2>	These are the 8 MSBs of the 10-bit code used to set the PLL frequency. The 2 LSBs reside in the PLLEqu register; these 2 bits should be written to first. The PLL frequency can be set using the following equation: $f_{PLL} = \left(\frac{PLLFreq<9:0>}{512} + 2^{-5}\right) \times f_{clk}$

Table 37 Description of PLLFreq bits for read operation

BIT	SYMBOL	DESCRIPTION
7 to 0	JV.7 to JV.0	**Jitter value.** These 8 bits determine the PLL clock jitter value (jitter values below 7% cannot be measured with this register). The absolute clock jitter value can be calculated as follows: PLL clock recovery jitter % $= \sqrt{\frac{JV<7:0> - 6.5}{2048}} \times 100$

7.5.6 PLL Equalizer Preset Register (PLLEqu)

The function of this register is dependent upon whether its being read or written. Tables 38, 39 and 40 define the register function for the write operation. Tables 38 and 41 define the register function for the read operation.

Table 38 PLL Equalizer Preset Register (address 03H) - WRITE/READ

7	6	5	4	3	2	1	0
PLLFreq.1	PLLFreq.0	Tap a1.2	Tap a1.1	Tap a1.0	Tap a2.2	Tap a2.1	Tap a2.0
–	–	–	–	–	LongSymb	FLock	InLock

Table 39 Description of PLLEqu bits for write operation

BIT	SYMBOL	DESCRIPTION
7	PLLFreq.1	These 2 bits are the 2 LSBs of the 10-bit PLL frequency code; see Section 7.5.5.
6	PLLFreq.0	
5	Tap a1.2	These 3 bits select the equalizer tap setting a1; see Table 40.
4	Tap a1.1	
3	Tap a1.0	
2	Tap a2.2	These 3 bits select the equalizer tap setting a2; see Table 40.
1	Tap a2.1	
0	Tap a2.0	

Table 40 Selection of equalizer tap settings: a1 and a2

Tap a1.2	Tap a1.1	Tap a1.0	a1 OR a2 EQUALIZER TAP SETTINGS
Tap a2.2	Tap a2.1	Tap a2.0	
0	0	0	0
0	0	1	–0.0625
0	1	0	–0.125
0	1	1	–0.1875
1	0	0	–0.25
1	0	1	–0.3125
X	X	X	All other settings are reserved.

Table 41 Description of PLLEqu bits for read operation

BIT	SYMBOL	DESCRIPTION
7 to 3	–	These 5 bits are reserved.
2	LongSymb	If LongSymb = 1, then a run length of 14 has been detected.
1	FLock	If FLock = 1, then PLL is in inner-lock range.
0	InLock	If Inlock = 1, then PLL is in lock.

7.5.7 PLL Lock Aid2 Preset Register (PLLFMeas)

The PLL setting point for the EFM counting locking strategy is controlled by setting the PLL frequency such that, there are, on average, a fixed number of EFM transitions per PLL clock period:

PLL Locking Frequency/EFM Transition Frequency = (EFM_Count + 32)/16 = 4.75 in a typical application.

This value (4.75) is dependent on disc and mechanical variations, improvements may be achieved by adjusting the value slightly.

Table 42 PLL Lock Aid2 Preset Register (address 04H) - WRITE

7	6	5	4	3	2	1	0
RL3_En	–	EFMns.5	EFMns.4	EFMns.3	EFMns.2	EFMns.1	EFMns.0

Table 43 Description of PLLFMeas bits

BIT	SYMBOL	DESCRIPTION
7	RL3_En	If RL3_En = 0, then EFM transition counting outer PLL lock strategy. If RL3_En = 1, then RL3 detection PLL outer lock strategy.
6	–	This bit is reserved.
5 to 0	EFMns<5:0>	These 6 bits select the EFM nominal setting. The default nominal setting should be set to ‘101110’.

7.5.8 Motor Control Register 2 (Motor2)

This is a dual-function register. When read Motor2 gives an indication of the EYE opening of the equalised HF.

Table 44 Eye Open Register (address 0DH) - READ

7	6	5	4	3	2	1	0
EOV.7	EOV.6	EOV.5	EOV.4	EOV.3	EOV.2	EOV.1	EOV.0

7.5.9 Viterbi Detector Setting Register (VitSet)

This register controls an advanced data slicer for improved bit detector performance.

- An adaptive slicer performs a second slice operation. This has a higher bandwidth than the first slicer.
- If switched on, the run length 2 push-back circuit pushes all run length two symbols to run length 3. The circuit will determine which transition was most likely in error and shift transition on that edge.

To avoid advanced detector hang-up, caused by a detection level that is too high and is not brought down, a Watchdog counter on the slicer level is installed.

The Watchdog counter is a counter that counts on the front-end PLL clock.

- If rl1 or a rl2 received: count + stepsize
- Elsif no transition: count + 1
- Elsif transition on a valid runlength: count – 8
- Elsif (count > maxcount): reset count.

Stepsize and maxcount can be set by writing to the VitSet register. On a reset of the counter the slice level is also reset.

Table 45 Viterbi Detector Setting Register (address 16H) - WRITE

7	6	5	4	3	2	1	0
AdSliceON	AdDetON	FEndAutoSON	RL2PB	WDog	MaxCnt	WDogCnt.1	WDogCnt.0

Table 46 Description of VitSet bits

BIT	SYMBOL	DESCRIPTION
7	AdSliceON	If AdSliceON = 0, then slicer reset (to logic 0). If AdSliceON = 1, then slicer active.
6	AdDetON	If AdDetON = 0, then advanced bit detector off. If AdDetON = 1, then advanced bit detector on.
5	FEndAutoSON	If FEndAutoSON = 0, then auto-scaling in front-end Hold mode. If FEndAutoSON = 1, then auto-scaling in front-end on.
4	RL2PB	If RL2PB = 0, then run length 2 push-back off. If RL2PB = 1, then run length 2 push-back on.
3	WDog	If WDog = 0, then slicer Watchdog is off. If WDog = 1, then slicer Watchdog is on.
2	MaxCnt	If MaxCnt = 0, then maxcount is 1024. If MaxCnt = 1, then maxcount is 2048.
1	WDogCnt.1	These 2 bits select the Watchdog count step; see Table 47.
0	WDogCnt.0	

Table 47 Selection of Watchdog count step

WDogCnt.1	WDogCnt.0	WATCHDOG COUNT STEP
0	0	32
0	1	64
1	0	128
1	1	256

7.5.10 Motor Control Register 1 (Motor1)

When read this register holds the 8-bit advanced slicer compensation value.

Table 48 Motor Control Register 1 (address 0CH) - READ

7	6	5	4	3	2	1	0
ASCV.7	ASCV.6	ASCV.5	ASCV.4	ASCV.3	ASCV.2	ASCV.1	ASCV.0

7.6 Decoder function

7.6.1 DEMODULATOR

The demodulator block includes sync extraction, interpolation and protection circuits, and converts the 14-bit EFM data and subcode words into 8-bit symbols.

Two counters are used to detect frame synchronisation. The coincidence counter detects the coincidence of successive syncs (i.e. 2 syncs are within 588 ± 1 EFM clock). The main counter partitions the EFM signal into 16 or 17-bit bytes; and is reset when a sync coincidence is found or the sync pulse is within ±6 EFM clock pulses. The sync coincidence signal generates the 'lock' signal which goes active HIGH when one sync coincidence is found, and goes inactive when no sync coincidence is found within 61 consecutive EFM frames. The frame sync detection circuit extracts the frame sync and will guard against mis-detection; up to 7 consecutive corrupted syncs will not disturb the sync detection.

After data demodulation the sector sync is extracted; a double lock counter is used. The main counter interpolates the sector syncs, and a coincidence counter resets the main counter.

7.6.2 ERROR CORRECTOR

The error corrector can correct up to 2 errors on the C1 level and up to 4 errors on the C2 level. The error corrector also contains a flag processor. Flags are assigned to symbols when the error corrector cannot ascertain if the symbols are definitely good. C1 generates output flags that are used by C2. The C2 output flags are output via the FLAG signal along with the I^2S, and can be used by the interpolator for concealment of uncorrectable errors for audio output.

- **Muting of data.** Data output via the serial interface and/or the EBU can be set to zero using register Output3.
- **Concealment of audio errors.** A simple 1 sample linear interpolator can be selected via register Output3. If selected the interpolator becomes active if a single sample is flagged as erroneous; left and right channels have independent interpolators.

7.6.2.1 CFLG pin

The error corrector outputs status information in serial format on the CFLG pin. Each frame consists of 11 bits (each 7 system clock periods long), beginning with a start bit, then data bits then pause bit (see Fig.9). The repetition rate of CFLG is not fixed; it depends on the disc speed and output interface speed. There is always at least one pause bit. The structure of the frame is shown in Table 49.

Table 49 Frame structure

BIT	VALUE	COMMENT
0	logic 1	start bit
1 to 3	cormode<2:0>	000 = C1 correction
		011 = C2 correction
		100 = corrector not active
		all other codes not used
4	corfail	failure flag set because correction impossible
5	flagfail	failure flag set because correction too risky
6 to 9	rootcount<3:0>	number of errors corrected, after Euclidean algorithm
10	logic 0	pause bit

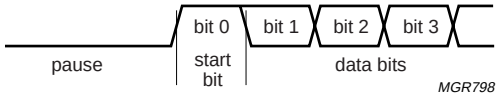


Fig.9 Format on CFLG pin.

7.6.3 Decoder Mode Select Register (DecoMode)

Table 50 Decoder Mode Select Register (address 13H) - WRITE

7	6	5	4	3	2	1	0
Mode.6	Mode.5	Mode.4	Mode.3	Mode.2	Mode.1	Mode.0	LWCon

Table 51 Description of DecoMode bits

BIT	SYMBOL	DESCRIPTION
7 to 1	Mode<6:0>	These 7 bits select the Decoder mode; see Table 52.
0	LWCon	When a logic 0, LaserOn and WriteOn2 signals operate normally. When a logic 1, LaserOn and WriteOn2 signals are reset.

Table 52 Selection of Decoder mode

MODE							DECODER MODE
6	5	4	3	2	1	0	
0	0	0	0	0	0	0	**Flush mode.** Deinterleaver table is emptied and all data is discarded.
0	1	0	0	0	0	0	**Normal play.** Uses the Quad-pass error correction mode used for disc speeds up to 75% of maximum defined by ADCCLK. Note the data integrity should be checked using the CRC as the FLAG pin is not fully defined in this mode.
0	1	0	1	0	0	0	**Fast play.** Used for audio play or fast CDROM mode (25 to 100% of maximum defined by ADCCLK), this reduces the error correction performance, but also halves the throughput time of error corrector (only dual pass error correction algorithm used).
0	1	1	0	0	0	0	**Hold mode.** Data into output FIFO is stopped (header/subheader decoding remains operative).
0	1	0	1	0	1	0	**Encode mode.**
X	X	X	X	X	X	X	All other combinations reserved.

7.6.4 Frame error status registers (C1BLER and C2BLER)

These two registers are non-buffered counters. Each time a C1 frame with errors is found the register C1BLER is incremented. In the same way C2BLER increments on all C2 frames with errors. When the value of either register reaches 255, it will hold. When read, the register value is reset.

7.6.4.1 C1 Block Error Register (C1BLER)

Table 53 C1 Block Error Register (address 39H) - READ

7	6	5	4	3	2	1	0
C1BLER.7	C1BLER.6	C1BLER.5	C1BLER.4	C1BLER.3	C1BLER.2	C1BLER.1	C1BLER.0

7.6.4.2 C2 Block Error Register (C2BLER)

Table 54 C2 Block Error Register (address 3AH) - READ

7	6	5	4	3	2	1	0
C2BLER.7	C2BLER.6	C2BLER.5	C2BLER.4	C2BLER.3	C2BLER.2	C2BLER.1	C2BLER.0

7.6.5 DATA FIFO

The decoder block can be viewed as a FIFO, demodulated data is written in while the output interface reads from it. The way in which the FIFO is filled depends on the decoder mode set via register DecoMode. The decoder modes that effect the filling of the FIFO are described in Sections 7.6.5.1 to 7.6.5.3.

Data is read out via the output interface and is only possible if enough data is present in the FIFO. The minimum amount of data is 110 C1 frames (C1 frame = 24 user bytes).

7.6.5.1 Flush mode

The FIFO content is thrown away; no read-out possible. It is necessary to flush the FIFO every time a context switch is made; when data written to the FIFO is not subsequent with data already present in the FIFO.

7.6.5.2 Hold mode

Writing to the FIFO is stopped; read-out possible if FIFO was filled. This mode is intended to avoid FIFO overflow by implementing a ‘stop write - jump back’ action in the microprocessor. When the microprocessor switches to hold mode the switch-over is synchronised internally with the next subcode block start, allowing the microprocessor to know exactly where the writing will stop or start. This mode is also intended to avoid overflow in the block decoder; on imminent overflow the microprocessor can switch to hold mode. When it jumps back one track and switches off hold mode on the same subcode address the SAA7392 will make a seamless link.

7.6.5.3 Play mode

The FIFO is filled; read-out possible when FIFO filling big enough.

7.6.5.4 Data FIFO monitoring

The state of the internal data FIFO may be monitored by reading the AnaSet1 register. This gives the number of C1 frames present in the FIFO including those frames in the outer corrector (i.e. 110 in CD fast and 220 in CD normal). One C1 frame equates to 24 user bytes in CD mode).

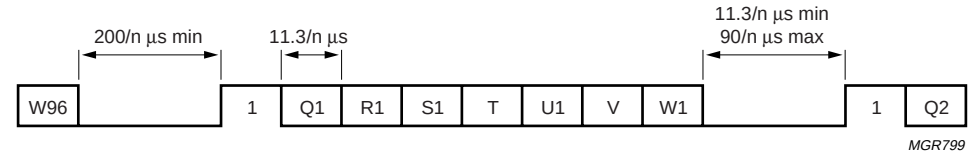
7.7 Subcode interface

Q-channel subcode data can be read using the ClockPre and DecoMode registers. The ClockPre register must be read first; this contains the status of the Q-channel subcode that may be read. Reading the ClockPre register with Ready = 1 will block the subcode interface for data read; no new subcode will overwrite the current subcode, and the microprocessor may retrieve as many bytes (up to 12) as required by issuing reads to register DecoMode.

After finishing the subcode read the microprocessor must release the interface by issuing a read to the dummy register SubReadEnd (no data can be read from it); this allows the SAA7392 to capture new subcode frames.

The serial Q to W subcode is output on the V4 pin as illustrated in Fig.10. The subcode sync word is formed by a pause of $^{200}/_{n}$ µs minimum; where n = the disc speed. Each subcode word starts with a logic 1 followed by 7 bits (Q to W); the bit time is 0.5 of the period of the WCLK signal, ($^{11.3}/_{n}$ µs). The gap between the words is between $^{11.3}/_{n}$ and $^{90}/_{n}$ µs. Note that the subcode data cannot be guaranteed at a rate higher than 0.5 × the maximum data rate programmed.

The subcode data is also available in the EBU output (EBUOUT).



n = disc speed

Fig.10 Subcode format and timing on V4 pin.

7.7.1 Clock Preset Register (ClockPre)

This is a dual-function register. When read the status of the Q-channel is returned.

Table 55 Clock Preset Register (address 12H) - READ

7	6	5	4	3	2	1	0
Ready	Busy	CRC OK	–	–	–	–	–

Table 56 Description of ClockPre bits

BIT	SYMBOL	DESCRIPTION
7	Ready	If Ready = 0, then buffer filling. If Ready = 1, then valid Q subcode frame available.
6	Busy	If Busy = 0, then buffer filling. If Busy = 1, then buffer held (set after reading this register with Ready = 1).
5	CRC OK	If CRC OK = 0, then CRC not checked or not OK. If CRC OK = 1, then CRC of Q-channel checks OK, only valid if Ready = 1.
4 to 0	–	These 5 bits are reserved.

7.7.2 Decoder Mode Select Register (DecoMode)

This is a dual-function register. When read this register holds the Q-channel subcode data.

Table 57 Decoder Mode Select Register (address 13H) - READ

7	6	5	4	3	2	1	0
SubD.7	SubD.6	SubD.5	SubD.4	SubD.3	SubD.2	SubD.1	SubD.0

7.7.3 Subcode Read End Register (SubReadEnd)

After finishing a subcode read, the microprocessor must release the interface to allow the SAA7392 to capture new subcode frames. This is done by issuing a read to SubReadEnd. No data can be read from this register; only the side effect is important.

Table 58 Subcode Read End Register (address 14H) - READ

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

7.8 Digital output

The AES/EBU signal is available on pin EBUOUT, according to the format defined by the *"IEC958 specification"*. This signal is only available in the CLV modes of the decoder (not in QCLV). Three different modes can be selected:

- EBU off
- EBU data all zero
- EBU data valid.

The EBU interface uses a clock derived from XTLI for timing generation. For correct operation of the EBU interface, BCLK must be non-gated and the selected EBU clock and BCLK must fulfil the following constraint:

$$\text{EBU clock} = \text{WCLK} \times 64$$

WCLK is BCLK divided by 16, 24 or 32 depending on the chosen output format.

7.8.1 Input Configuration Register (InputConfig)

Table 59 Input Configuration Register (address 1EH) - WRITE

7	6	5	4	3	2	1	0
EBUClkSelect	–	–	–	–	ScramOn	InputFmt.1	InputFmt.0

Table 60 Description of InputConfig bits

BIT	SYMBOL	DESCRIPTION
7	EBUClkSelect	If EBUClkSelect = 0, then the EBU clock frequency is $^1/_3 f_{XTLI}$. If EBUClkSelect = 1, then the EBU clock frequency is $^2/_3 f_{XTLI}$ (input MUXSWI must be a logic 1 for this setting).
6 to 3	–	These 4 bits are reserved.
2	ScramOn	See Section 7.11.1.
1	InputFmt1	
0	InputFmt0	

7.8.2 FORMAT

The digital audio output consists of 32-bit words (subframes) transmitted in biphasemark code (two transitions for a logic 1 and one transition for a logic 0). Words are transmitted in blocks of 384.

Table 61 Digital audio output subframe format

BIT	FIELD NAME	DESCRIPTION
0 to 3	sync	The sync word is formed by violation of the biphase rule and therefore does not contain any data. Its length is equivalent to 4 data bits. The 3 different sync patterns indicate the following situations: • Sync B: Start of a block (384 words), word contains left sample • Sync M: Word contains left sample (no block start) • Sync W: Word contains right sample.
4 to 7	auxiliary	These bits are not used; normally zero.
8 to 27	audio sample	Left and right samples are transmitted alternately. The first 4 bits not used (always zero); 2's compliment; LSB = bit 12 and MSB = bit 27.
28	validity flag	If validity flag = 1, audio samples are flagged if an error has been detected but was uncorrectable. This flag remains the same even if data is taken after concealment.
29	user data	Subcode bits Q until W from the subcode section are transmitted via the user data bit. This data is asynchronous with the block rate.
30	channel status	The channel status bit is the same for left and right words. Therefore, a block of 384 words contains 192 channel status bits. The category code is always CD. The bit assignment is shown in Table 62.
31	parity bit	Even parity for bits 4 to 30.

Table 62 Channel status bit assignment

BIT	FIELD NAME	DESCRIPTION
0 to 4	control	Bits 1 to 4 copied from register Output2. Bit 2 is logic 1 when copy permitted. Bit 3 is logic 1 when recording has pre-emphasis
5 to 7	reserved mode	always zero
8 to 15	category code	CD: bit 8 = logic 1, all other bits = logic 0
28 to 29	clock accuracy	set by register Output2: 10 = level I 00 = level II 01 = level III
16 to 27, 30 to 191	remaining	always zero

7.9 Serial output interface

The serial data output interface consists of three signals: WCLK (word select), BCLK (serial clock), DATAO (serial data). The polarity of WCLK and the data can be inverted. The FLAG signal is used to identify if there are errors in either the LSB or MSB of the 16-bit data word. The interface can be used as a master or slave interface (BCLK and WCLK are then inputs), selectable by register Output1.

The serial data interface can be switched into two modes: Philips I^2S and the EIAJ format (the protocol can use either 16, 24 or 32 BCLK clocks for each 16-bit output, selectable by Output1 register). The formats are shown in Figs 11 to 16.

The BCLK frequency can be selected by register ClockPre, or input externally. If the data out rate does not correspond with the disc speed and the bit clock data, the FIFO will either fill or empty. FIFO underflow (STOPCK output goes HIGH indicating absence of data) must be avoided. If BCLK is an input, it should be stopped and restarted again when STOPCK goes LOW. If BCLK is output, it is automatically stopped if BCLK gate enable is set (via register ClockPre); if it is not set then unpredictable operation will result.

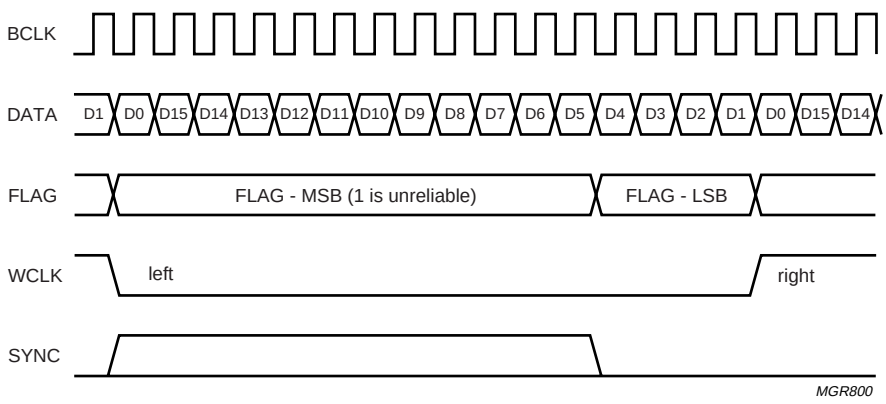


Fig.11 Format 1: 16 clocks/word I^2S format.

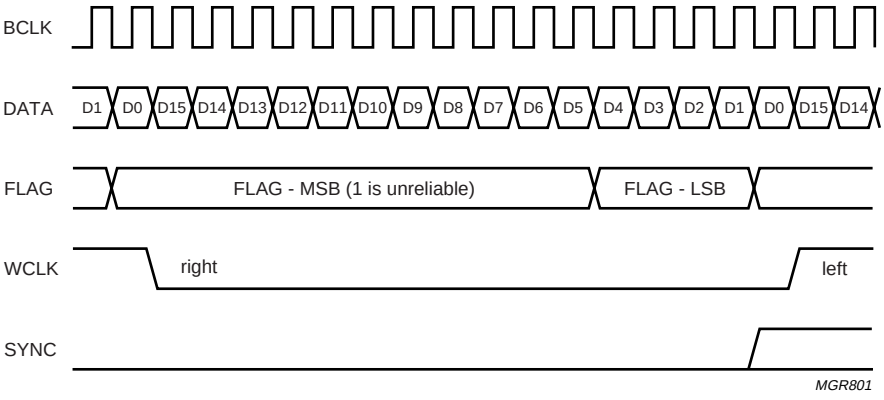


Fig.12 Format 2: 16 clocks/word ‘S’ format (WCLK inverted).

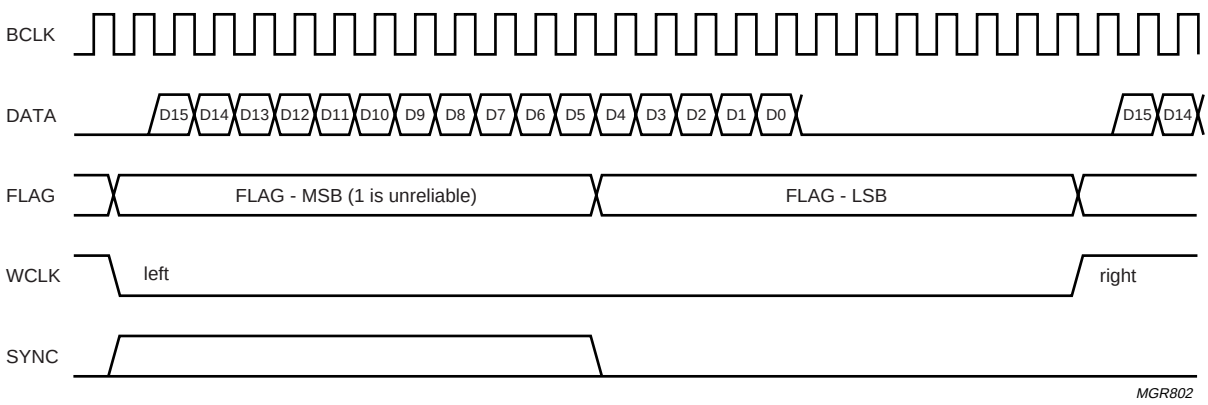


Fig.13 Format 3: 24 clocks/word I^2S format.

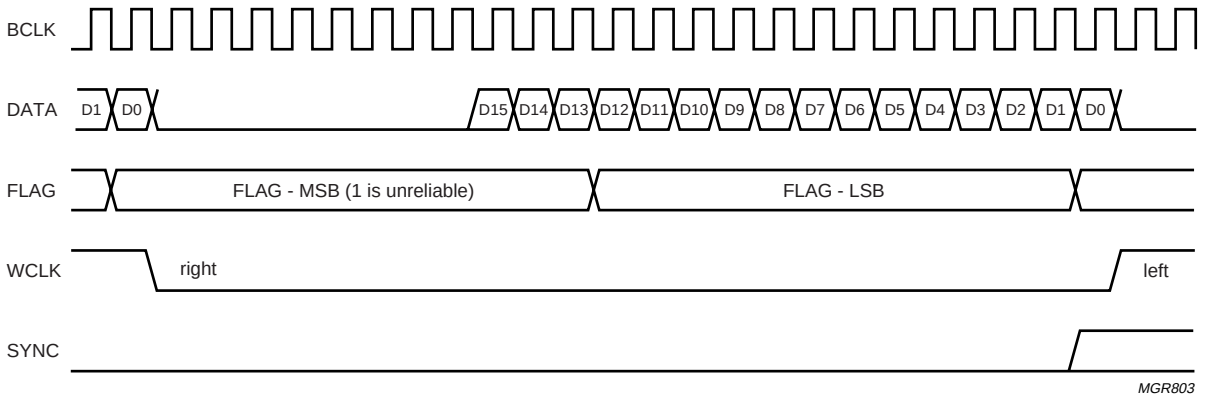


Fig.14 Format 4: 24 clocks/word ‘S’ format (WCLK inverted).

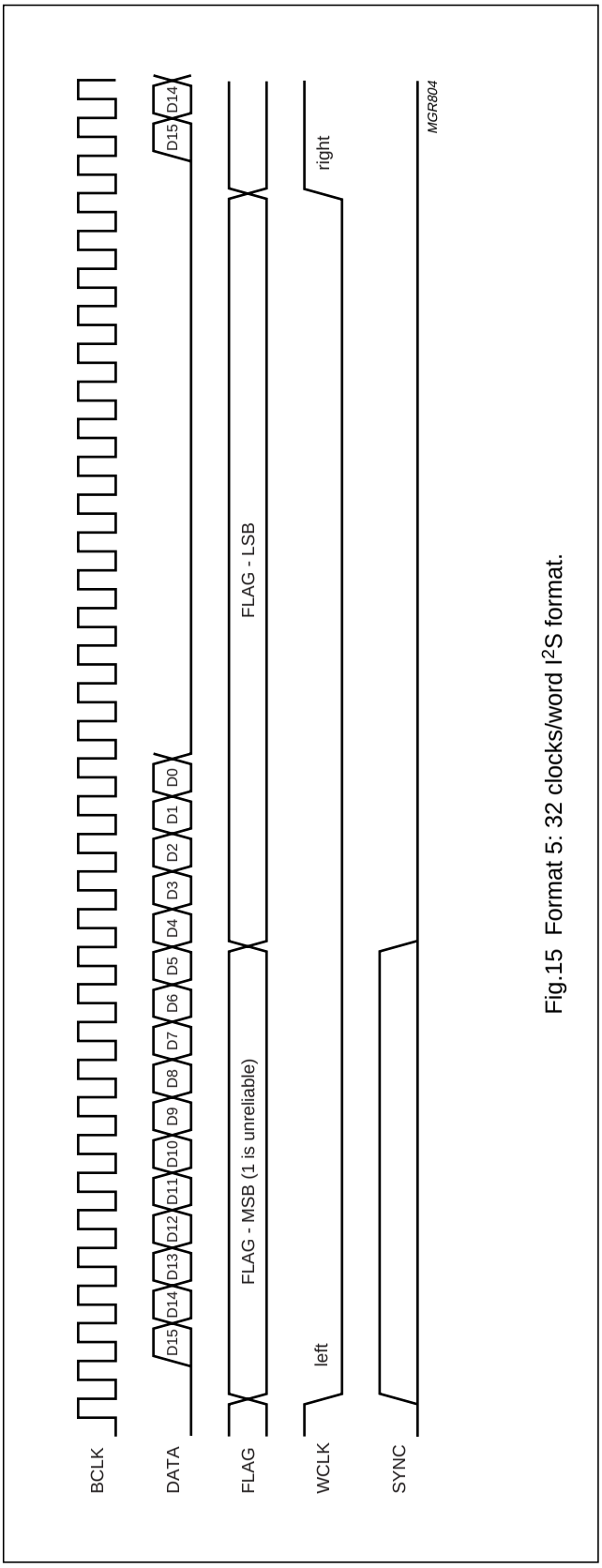


Fig.15 Format 5: 32 clocks/word I²S format.

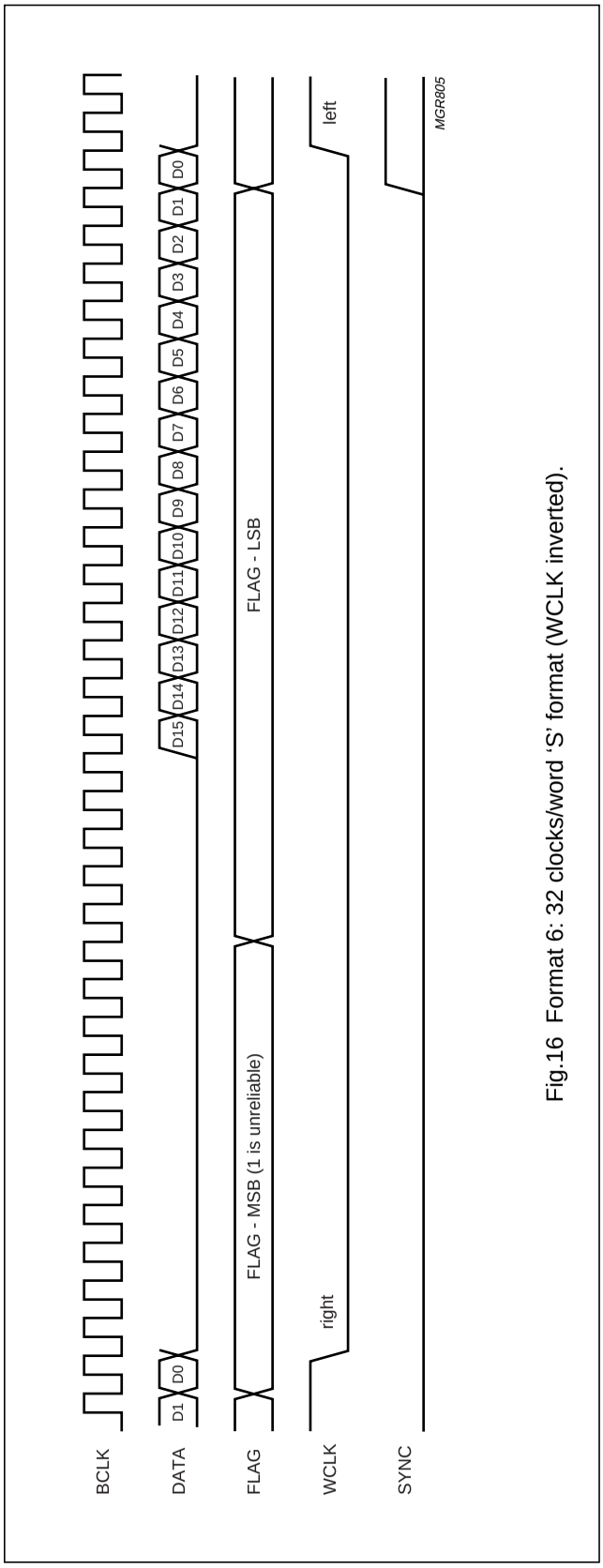


Fig.16 Format 6: 32 clocks/word 'S' format (WCLK inverted).

7.9.1 I²S Output Register (Output1)

Table 63 I²S Output Register 1 (address 05H) - WRITE

7	6	5	4	3	2	1	0
Format.2	Format.1	Format.0	WClkIO	BClkIO	–	–	–

Table 64 Description of Output1 bits

BIT	SYMBOL	DESCRIPTION
7	Format.2	These 3 bits select the format; see Table 65.
6	Format.1	
5	Format.0	
4	WClkIO	When WClkIO = 0, then WCLK is in Input mode. When WClkIO = 1, then WCLK is in Output mode.
3	BClkIO	When BClkIO = 0, then BCLK is in Input mode. When BClkIO = 1, then BCLK is in Output mode.
2 to 0	–	These 3 bits are reserved.

Table 65 Format selection

Format.2	Format.1	Format.0	FORMAT
0	0	0	Format 1
0	0	1	Format 2
0	1	0	Format 3
0	1	1	Format 4
1	0	0	Format 5
1	0	1	Format 6

7.9.2 I²S Output Register 2 (Output2)

Table 66 I²S Output Register 2 (address 06H) - WRITE

7	6	5	4	3	2	1	0
EBUValid	EBUOn	EBUCon.29	EBUCon.28	EBUCon.3	EBUCon.2	EBUCon.1	EBUCon.0

Table 67 Description of Output2 bits

BIT	SYMBOL	DESCRIPTION
7	EBUValid	If EBUValid = 0, then EBU/IEC958 output data zero. If EBUValid = 1, then EBU/IEC958 output data valid.
6	EBUOn	If EBUOn = 0, then EBU/IEC958 output is switched off. If EBUOn = 1, then EBU/IEC958 output switched on.
5	EBUCon.29	These 2 bits are copied to bits 29 and 28 of the IEC958 control channel (crystal accuracy); see Table 62.
4	EBUCon.28	
3 to 0	EBUCon<3:0>	These 4 bits are copied to bits 4 to 1 of the IEC958 control channel (data/audio, copy protect, de-emphasis); see Table 62.

7.9.3 I²S Output Register 3 (Output3)

Table 68 I²S Output Register 3 (address 07H) - WRITE

7	6	5	4	3	2	1	0
WClkHLeft	DescrmOn	InterpOn	–	FlagPin	KillDataOn	KillEBUOn	–

Table 69 Description of Output3 bits

BIT	SYMBOL	DESCRIPTION
7	WClkHLeft	When WClkHLeft = 0, then WCLK is HIGH on right byte of I²S/S format output; (use for I²S format modes). When WCLKHLeft = 1, then WCLK is HIGH on left byte of I²S/S format output; (use for S format modes).
6	DescrmOn	When DescrmOn = 1, then Descrambling function on. For CD-DA playback descrambling has to be switched off. For CD-ROM descrambling is required but it might also be possible that a descrambling function is available in the buffer manager (block decoder).
5	InterpOn	When InterpOn = 0, then audio data interpolation is switched off. When InterpOn = 1, then audio data interpolation switched on.
4	–	This bit is reserved.
3	FlagPin	When FlagPin = 0, then flag pin sends out EDC-OK signal on byte 2063 of the sector. When FlagPin = 1, then flag pin sends out reliability flag information.
2	KillDataOn	When KillDataOn = 0, then in normal mode. When KillDataOn = 1, then all data in serial output channel is set to zero.
1	KillEBUOn	When KillEBUOn = 0, then in normal mode. When KillEBUOn = 1, then all data in IEC958/EBU channel is set to zero
0	–	This bit is reserved.

7.10 Motor control

The spindle motor is controlled by a fully integrated digital servo sub-system within the SAA7392. Information from the data FIFO, data recovery PLL and tacho inputs may be used to calculate the motor control output signals.

The frequency set-point, the FIFO settings and coefficients G, K_i and K_f are all programmable using the seven motor control registers. The motor can be controlled in either Pulse Width Mode (PWM) or Pulse Density Mode (PDM).

The tacho control subsystem allows a range of different motor tacho systems to be used, the modes and prescalers are programmable via the tacho register set. (Note due to pin multiplexing, PWM motor control mode and CAV mode using tacho control are mutually exclusive).

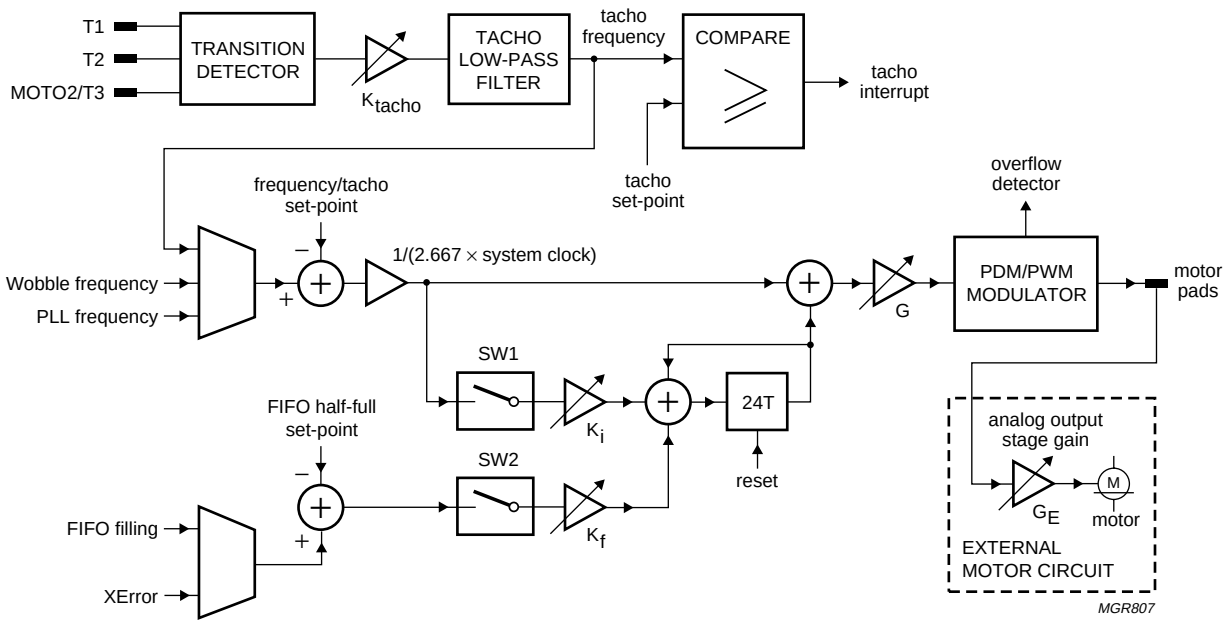


Fig.17 Motor servo block diagram.

7.10.1 Motor Control Register 1 (Motor1)

The frequency/tacho set-point (i.e. the target PLL or tacho frequency) is calculated as follows:

$$\text{Frequency set-point} = \left[1 - \left(\frac{\text{MFS<7:0>}}{256}\right)\right] \times 2.667 \times \text{ADC clock}$$

$$\text{PLL frequency} = \left(\frac{\text{PLLFreqR<7:0>}}{128}\right) \times \text{ADC clock}$$

Note that: (PLL frequency – frequency set-point) must be less than $1.33 \times$ ADC clock.

Table 70 Motor Control Register 1 (address 0CH) - WRITE

7	6	5	4	3	2	1	0
MFS.7	MFS.6	MFS.5	MFS.4	MFS.3	MFS.2	MFS.1	MFS.0

7.10.2 Motor Control Register 2 (Motor2)

Table 71 Motor Control Register 2 (address 0DH) - WRITE

7	6	5	4	3	2	1	0
G.2	G.1	G.0	Ki.1	Ki.0	Kf.2	Kf.1	Kf.0

Table 72 Description of Motor2 bits

BIT	SYMBOL	DESCRIPTION
7	G.2	These 3 bits select coefficient G; see Table 73.
6	G.1	
5	G.0	
4	Ki.1	These 2 bits select coefficient K_i; see Table 74. In order to set the integrator bandwidth low enough at high system clock speeds, an extra divider for K_i has been added. This is set by writing to register Motor7. The resulting $K_{i(tot)}$ is then the K_i set by Motor2 multiplied by the $K_{i'}$ set by Motor7.
3	Ki.0	
2	Kf.2	These 3 bits select coefficient K_f; see Table 75.
1	Kf.1	
0	Kf.0	

Table 73 Selection of coefficient G

G.2	G.1	G.0	COEFFICIENT G
0	0	0	8.36
0	0	1	10.4
0	1	0	16.7
0	1	1	20.9
1	0	0	33.4
1	0	1	41.8
1	1	0	66.9
1	1	1	83.6

Table 74 Selection of coefficient K_i

Ki.1	Ki.0	COEFFICIENT K_i
0	0	3.1×10^{-5}
0	1	6.1×10^{-5}
1	0	1.2×10^{-4}
1	1	2.4×10^{-4}

Table 75 Selection of the coefficient K_f

Kf.2	Kf.1	Kf.0	COEFFICIENT K_f
0	0	0	reserved
0	0	1	3.7×10^{-9}
0	1	0	7.5×10^{-9}
0	1	1	1.5×10^{-8}
1	0	0	3.0×10^{-8}
1	0	1	6.0×10^{-8}
1	1	0	1.2×10^{-7}
1	1	1	2.4×10^{-7}

7.10.3 Motor Control Register 7 (Motor7)

Table 76 Motor Control Register 7 (address 1DH) - WRITE

7	6	5	4	3	2	1	0
PhErSrc	–	Kf'.2	Kf'.1	Kf'.0	Ki'.2	Ki'.1	Ki'.0

Table 77 Description of Motor7 bits

BIT	SYMBOL	DESCRIPTION
7	PhErSrc	If PhErSrc = 0, then the phase error source is FIFOFil. If PhErSrc = 1, then the phase error source is XError.
6	–	This bit is reserved.
5	Kf'.2	These 3 bits select the value of the $K_{f'}$ coefficient; see Table 78. $K_{f'}$ operates by sampling the input. For example, for $K_{f'} = 1$, every sample of the input is passed through to a following integrator circuit, for a $K_{f'}$ of 0.5 every 2nd sample is passed through, for a $K_{f'}$ of 0.25 every 4th sample is passed through, and so on. For a DC input signal, $K_f \times K_{f'}$ should always give the same result. If however, the input is varying sufficiently quickly, the $K_f \times K_{f'}$ combinations with the same product will not always give the same result, especially for low values of $K_{f'}$, where the sampling in the extreme becomes 1 out of every 128 samples. (The input samples to the block that performs the $K_{f'}$ multiplication occur at a rate of 1 sample every 24 system clock periods.)
4	Kf'.1	
3	Kf'.0	
2	Ki'.2	These 3 bits select the value of the $K_{i'}$ coefficient; see Table 78.
1	Ki'.1	
0	Ki'.0	

Table 78 Selection of coefficient $K_{f'}$ and $K_{i'}$

Kf'.2	Kf'.1	Kf'.0	COEFFICIENTS $K_{f'}$ AND $K_{i'}$
Ki'.2	Ki'.1	Ki'.0	
0	0	0	1
0	0	1	0.5
0	1	0	0.25
0	1	1	0.125
1	0	0	0.0625
1	0	1	0.03125
1	1	0	0.015625
1	1	1	0.0078125

7.10.4 Motor Control Register 3 (Motor3)

The function of this register is dependent upon whether its being read or written.

When written, the value determines the FIFO set-point value in C1 frames (one C1 frame is 24 bytes). The FIFO set-point value should be set to a value greater than 110 frames for CD Fast mode and greater than 219 frames for CD Normal mode. The maximum value for the FIFO half-full value is 256. Normally, the FIFO set-point value should be set to the minimum value plus a margin for disc overspeed. The set-point value is calculated as shown below:

FIFO set-point value = HFSP<7:0> $\times$ 8.

When read, the motor speed can be determined from the recovered tacho frequency. Note this information is only valid when a Hall motor is being used in the application and the tacho subsystem is correctly configured. The reading is related to the motor frequency (relative to pulse rate on a single tacho input pin) as shown in Equation (1); the actual Tacho frequency can be calculated as shown in Equation (2). Negative frequencies will be measured if the motor is rotating in reverse.

$$\text{Tacho4} = \frac{[(6 \times \text{Tacho1}) \times \text{Motor frequency}]}{\text{Tacho sample rate}} \quad (1)$$

$$\text{Tacho frequency (Hz)} = \text{Tacho4} \times \frac{\text{ADC clock}}{128} \quad (2)$$

Table 79 Motor Control Register 3 (address 0EH) - WRITE/READ

7	6	5	4	3	2	1	0
HFSP.7	HFSP.6	HFSP.5	HFSP.4	HFSP.3	HFSP.2	HFSP.1	HFSP.0
TachoFreq.7	TachoFreq.6	TachoFreq.5	TachoFreq.4	TachoFreq.3	TachoFreq.2	TachoFreq.1	TachoFreq.0

Table 80 Description of Tacho4 bits for read operation

BIT	SYMBOL	DESCRIPTION
7 to 0	TachoFreq<7:0>	These 8 bits indicate the Tacho frequency (2's complement notation).

7.10.5 Motor Control Register 4 (Motor4)

Table 81 Motor Control Register (address 0FH) - WRITE

7	6	5	4	3	2	1	0
PWM_PDM	OVF_SW	SW1	SW2	MSCON.3	MSCON.2	MSCON.1	MSCON.0

Table 82 Description of Motor4 bits

BIT	SYMBOL	DESCRIPTION
7	PWM_PDM	If PWM_PDM = 0, then motor control in PWM mode. If PWM_PDM = 1, then motor control in PDM mode.
6	OVF_SW	If OVF_SW = 0, then SW1 and SW2 in normal operation. If OVF_SW = 1, then SW1 and SW2 will both open on overflow.
5	SW1	If SW1 = 0, then SW1 is open. If SW1 = 1, then SW1 is closed.
4	SW2	If SW2 = 0, then SW2 is open. If SW2 = 1, then SW2 is closed.
3	MSCON.3	These 4 bits are used to select the motor servo state; see Table 83.
2	MSCON.2	
1	MSCON.1	
0	MSCON.0	

Table 83 Selection of motor servo state

MSCON.3	MSCON.2	MSCON.1	MSCON.0	MOTOR SERVO STATE
0	0	0	0	Motor servo active.
0	0	0	1	Motor servo off (clears integrator).
0	0	1	0	Motor servo 3-state (motor output pin 3-states).
0	0	1	1	These 5 codes are reserved.
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	Motor start at 37% power.
1	0	0	1	Motor start at 50% power.
1	0	1	0	Motor start at 75% power.
1	0	1	1	Motor start at 100% power.
1	1	0	0	Motor stop at 37% power.
1	1	0	1	Motor stop at 50% power.
1	1	1	0	Motor stop at 75% power.
1	1	1	1	Motor stop at 100% power.

7.10.6 Motor Control Registers 5 and 6

These two registers hold the 16-bit motor integrator value. The motor integrator value can be read or updated using registers Motor5 and Motor6. Register Motor5 should be read or written first to prevent spurious results.

7.10.6.1 Motor Control Register 5 (Motor5)

Table 84 Motor Control Register 5 (address 10H) - READ/WRITE

7	6	5	4	3	2	1	0
MIV.7	MIV.6	MIV.5	MIV.4	MIV.3	MIV.2	MIV.1	MIV.0

7.10.6.2 Motor Control Register 6 (Motor6)

Table 85 Motor Control Register 6 (address 11H) - READ/WRITE

7	6	5	4	3	2	1	0
MIV.15	MIV.14	MIV.13	MIV.12	MIV.11	MIV.10	MIV.9	MIV.8

7.10.7 Tacho Gain Setting Register (Tacho1)

This register holds the 8-bit Tacho multiplier frequency value.

Table 86 Tacho Gain Setting Register (address 17H) - WRITE

7	6	5	4	3	2	1	0
KTacho.7	KTacho.6	KTacho.5	KTacho.4	KTacho.3	KTacho.2	KTacho.1	KTacho.0

7.10.8 Tacho Trip Setting Register (Tacho2)

This register holds the 8-bit Tacho interrupt trip frequency value.

Table 87 Tacho Trip Setting Register (address 18H) - WRITE

7	6	5	4	3	2	1	0
TIntF.7	TIntF.6	TIntF.5	TIntF.4	TIntF.3	TIntF.2	TIntF.1	TIntF.0

7.10.9 TACHO CONTROL REGISTER 3 (TACHO3)

Table 88 Tacho Control Register 3 (address 19H) - WRITE

7	6	5	4	3	2	1	0
SConS.1	SConS.0	TachoFRes	Moto2/T3	Fsam.1	Fsam.0	TachoIntLF	TachoMode

Table 89 Description of Tacho3 bits

BIT	SYMBOL	DESCRIPTION
7	SConS.1	These 2 bits select the motor servo frequency source; see Table 90.
6	SConS.0	
5	TachoFRes	If TachoFRes = 0, then the tacho filter is enabled (normal mode). If TachoFRes = 1, then the tacho filter is reset.
4	Moto2/T3	If Moto2/T3 = 0, then MOTO2 pin is an output. If Moto2/T3 = 1, then MOTO2 pin is tacho T3.
3	Fsam.1	These 2 bits select the tacho sample rate (f_s) as shown below: $f_s(Hz) = \frac{2^2 \times Fsam<1:0> \times system\ clock}{32768}$
2	Fsam.0	
1	TachoIntLF	If TachoIntLF = 0, then tacho interrupt is enabled on frequencies LOWER than set-point. If TachoIntLF = 1, then tacho interrupt is enabled on frequencies HIGHER than set-point.
0	TachoMode	If TachoMode = 0, tacho is in 3-pin mode. If TachoMode = 1, tacho is in 1-pin mode.

Table 90 Motor servo frequency source selection

SConS.1	SConS.0	MOTOR SERVO FREQUENCY SOURCE
0	0	Motor servo locks to PLL clock (recovered from channel EFM).
0	1	This value is reserved.
1	0	Motor servo runs on tacho frequency.
1	1	This value is reserved.

7.11 The serial in function

The serial in function takes serial data from the block decoder and passes the data on to the encoder. It supports the I^2S and Sony 3-wire serial interfaces (DATAI, BCLK and WCLK). The block is slave to the interface and therefore the BCLK and WCLK signals are generated externally and are the same signals as the serial data output interface. The selection of the input format follows the output interface setting.

7.11.1 Input Configuration Register (InputConfig)

Table 91 Input Configuration Register (address 1EH) - WRITE

7	6	5	4	3	2	1	0
EBUClkSelect	–	–	–	–	ScramOn	InputFmt.1	InputFmt.0

Table 92 Description of InputConfig bits

BIT	SYMBOL	DESCRIPTION
7	EBUClkSelect	See Section 7.8.1.
6 to 3	–	These 4 bits are reserved.
2	ScramOn	This bit enables/disables the scrambling function. If ScramOn = 1, then the scrambling function is enabled. Scrambling must be off for CD-DA and on for CD-ROM. However, if the buffer manager (block decoder) already scrambles the CD-ROM data, then the user must switch scrambling off in the CDR60.
1	InputFmt.1	These 2 bits select the input format to the encoder; see Table 93.
0	InputFmt.0	

Table 93 Selection of encoder input format

INPUTFMT.1	INPUTFMT.0	INPUT FORMAT
0	0	normal input data to encoder
0	1	all zero input data to encoder
1	X	random input data to encoder

7.12 The subcode insert function

The SAA7392 allows two modes of subcode insertion: Bypass mode and Auto-format mode.

- **Bypass mode.** In this mode the subcode is generated by a block decoder (such as PLUM) and transmitted to the SAA7392 via the proprietary Subcode Record Interface (SRI). P, Q and R to W channels are supported in this mode. However, the Q-channel CRC and the S0 and S1 bytes are generated by the SAA7392. The Q-channel CRC is calculated from the SRI data Q-bits and hence the 16 Q-channel CRC bits in the SRI data are discarded.
- **Auto-format mode.** In this mode, the SAA7392 will generate subcode data itself. However, only P and Q subcode channels are supported.

It is possible to begin subcode recording at any point in the subcode frame, for both internally and externally generated subcode, by loading the preset count field (bits PCF<6:0> in the SubPresetCount register) with the start position in the subcode frame and setting the init_frame bit to a logic 1.

It must be ensured that, if used, the external block encoder is ready to transmit subcode data before the frame initialisation is started. When the frame initialisation sequence is complete the init_frame bit is automatically cleared to a logic 0.

It should be noted that the frame initialisation sequence must be carried out at the beginning of every record session, even if starting from byte 0.

7.12.1 Subcode Preset Count Register (SubPresetCount)

The preset count field is loaded when this register is written. The current count field is returned when this register is read.

Table 94 Subcode Preset Count Register (address 22H) - WRITE

7	6	5	4	3	2	1	0
init_frame	PCF.6	PCF.5	PCF.4	PCF.3	PCF.2	PCF.1	PCF.0

Table 95 Subcode Preset Count Register (address 22H) - READ

7	6	5	4	3	2	1	0
init_frame	CCF.6	CCF.5	CCF.4	CCF.3	CCF.2	CCF.1	CCF.0

7.12.2 SUBCODE CONFIGURATION REGISTER 1 (SUBCONFIG1)

Table 96 Subcode Configuration Register 1 (address 23H) - WRITE

7	6	5	4	3	2	1	0
sri_on	p_toggle	p_one	time1	time2	copy_alter	zero_inc	time3

Table 97 Description of SubConfig1 bits

BIT	SYMBOL	DESCRIPTION
7	sri_on	If sri_on = 0, then subcode generated by CDR60 for P and Q-channels; R to W channels are all logic 0's. If sri_on = 1, then subcode data from SRI except Q-channel CRC and S0 and S1 bytes. SRI frame sync checking is enabled. SubConfig1 register bits 1 to 6 do not effect the SRI subcode data.
6	p_toggle	If p_toggle = 0, then P-channel takes value of the p_one bit. If p_toggle = 1, then P-channel is the value of bit 5 of the absolute frame byte in the current Q-code memory bank.
5	p_one	P-channel is the value of this bit. The P-channel takes on a new value of p_one at the start of a subcode frame.
4	time1	These 3 bits control the incrementing and decrementing of the Q-channel absolute and relative time fields; see Table 98.
3	time2	
0	time3	
2	copy_alter	If copy_alter = 0, then the copy bit will hold its current value. The copy bit value can be changed by writing to the Q-code memory control byte via the SubData register. If copy_alter = 1, then the copy bit in the subcode control byte will alternate every 4 subcode frames. This is accomplished by copying bit 2 of the current Q-code memory absolute frame byte to the copy bit of the Q-code memory control byte.
1	zero_inc	If zero_inc = 0, then all bits of the Q-code memory zero byte will hold. If zero_inc = 1, then bits 0 to 3 of the current Q-code memory zero byte are incremented modulo 10. Bits 4 to 7 are unchanged.

Table 98 Selection of Q-channel time fields

time3	time2	time1	Q-CHANNEL TIME FIELD
0	0	0	increment relative and absolute time
0	1	1	increment relative time and hold absolute time
1	0	0	hold relative time and increment absolute time
1	0	1	decrement relative time and increment absolute time
1	1	1	hold relative and absolute time
X	X	X	All other settings are reserved.

7.12.3 Subcode Configuration Register 2 (SubConfig2)

Table 99 Subcode Configuration Register 2 (address 24H) - READ/WRITE

7	6	5	4	3	2	1	0
BSwOn	InOBO	relcnt.2	relcnt.1	relcnt.0	curcnt.2	curcnt.1	curcnt.0

Table 100 Description of SubConfig2 bits

BIT	SYMBOL	DESCRIPTION
7	BSwOn	If BSwOn = 0, then no bank switching. If BSwOn = 1, then bank switch will occur when first subcode sync output by EFM modulator and curcnt<2:0> = 000
6	InOBO	If InOBO = 0, then original bank is used again. In this case no bank switching or interrupts occur and automatic bank update is still done on the original bank, even when the other bank is output. If InOBO = 1, then the other bank is automatically inserted once into Q-channel, this bit is automatically reset to logic 0.
5 to 3	relcnt<2:0>	Count value to load curcnt<2:0>.
2 to 0	curcnt<2:0>	Current value of subcode frame counter.

7.12.4 Subcode Start Data Register (SubStartData)

Table 101 Subcode Start Data Register (address 25H) - WRITE

7	6	5	4	3	2	1	0
NextSet	–	–	–	–	–	–	–

Table 102 Description of SubStartData bits

BIT	SYMBOL	DESCRIPTION
7	NextSet	If NextSet = 0, then access other bank. If NextSet = 1, then access current bank that is being used for Q-channel data.
6 to 0	–	These 7 bits are reserved.

7.12.5 Subcode Data Register (SubData)

Sub-CPU read/write from/to the SubData register causes a read or write to occur on the Q-code memory. Normally the sub-CPU would access all 10 bytes of a Q-code bank. Before such a block access the sub-CPU must issue a write to the SubStartWrite register.

Table 103 Subcode Data Register (address 26H) - READ/WRITE

7	6	5	4	3	2	1	0
QCD.7	QCD.6	QCD.5	QCD.4	QCD.3	QCD.2	QCD.1	QCD.0

7.12.6 GENERATING CD-TEXT

CD-Text is stored in the R to W channels of the disc. The CD-Text standard states that a CD-Text disc must at least contain CD-Text information in the lead-in, and it may contain CD-Text information in the program area.

CDR60 supports writing of the R to W subcode channels only in bypass mode which means that the whole subcode information has to be generated from an external device. It is not possible to use the automatic subcode formatting features like auto-incrementing or the two buffers which means that the application has to take care of the whole subcode formatting itself. However, if an application likes to write a CD-Text disc that contains CD-Text information only in the lead-in area, then the application can use the auto-format features for writing the program area and only use the bypass mode to write CD-Text information when writing the lead-in.

7.13 The data encoder block

This block generates all C1/C2 error correction data for the raw data.

7.14 Encode control block

This block controls the starting and stopping of the writing process. It does this based on the QSync signal it receives from the subcode insert block and the ATIPSync and W441 disc position information it receives from the Wobble processor.

7.14.1 ENCODE WRITEON CONTROL REGISTER (ENCODEWCON)

Table 104 Encode WriteOn Control Register (address 30H) - READ/WRITE

7	6	5	4	3	2	1	0
WriteOn1	WriteOn2	EFMDelOn	Count.4	Count.3	Count.2	Count.1	Count.0

Table 105 Description of EncodeWContr bits

BIT	SYMBOL	DESCRIPTION
7	WriteOn1	If WriteOn1 = 0, then laser is off or turn laser off. If WriteOn1 = 1, then laser is on or turn laser on.
6	WriteOn2	If WriteOn2 = 0, then data flow is off or turn data flow off. If WriteOn2 = 1, then data flow is on or turn data flow on.
5	EFMDelOn	If EFMDelOn = 0, then disable delay of WriteOn1. If EFMDelOn = 1, then enable delay of WriteOn1 with EncodeStartOffset/EncodeStopOffset of W441 pulses relative to ATIPSync.
4	Count.4	These 5 bits determine the delay generation of internal WriteOn flags by Count<4:0> ATIPSyncs.
3	Count.3	
2	Count.2	
1	Count.1	
0	Count.0	

7.14.2 Encode Start Offset Register (EncodeStartOffset)

Table 106 Encode Start Offset Register (address 31H) - WRITE

7	6	5	4	3	2	1	0
StartOffset.7	StartOffset.6	StartOffset.5	StartOffset.4	StartOffset.3	StartOffset.2	StartOffset.1	StartOffset.0

Table 107 Description of EncodeStartOffset bits

BIT	SYMBOL	DESCRIPTION
7 to 0	StartOffset<7:0>	If enabled, WriteOn flags are delayed by EncodeStartOffset pulses of W441 when starting the encode process.

7.14.3 Encode Stop Offset Register (EncodeStopOffset)

Table 108 Encode Stop Offset Register (address 32H) - WRITE

7	6	5	4	3	2	1	0
StopOffset.7	StopOffset.6	StopOffset.5	StopOffset.4	StopOffset.3	StopOffset.2	StopOffset.1	StopOffset.0

Table 109 Description of EncodeStopOffset bits

BIT	SYMBOL	DESCRIPTION
7 to 0	StopOffset<7:0>	If enabled, WriteOn flags are delayed by EncodeStopOffset pulses of W441 when stopping the encode process.

7.14.4 Encode XOffset Register (EncodeXOffset)

The 10-bit value for Xoffset must be written in two steps. EncodeXOffset can be written to in any order. XOffset<9:0> is a 2’s complement number which gives a range of –511 to 511.

Table 110 Encode XOffset Register (address 33H) - WRITE

7	6	5	4	3	2	1	0
0	0	XOffset.5	XOffset.4	XOffset.3	XOffset.2	XOffset.1	XOffset.0
0	1	–	–	XOffset.9	XOffset.8	XOffset.7	XOffset.6

Table 111 Description of EncodeXoffset bits

BIT	SYMBOL	DESCRIPTION
7 to 0	XOffset<9:0>	Offset applied to the phase error calculated between ATIPSync and Q-code sync.

7.15 The EFM modulator

This block takes data from the data encoder block and adds the subcode and synchronisation information. This data stream is modulated using EFM according to the Red Book standard. This data is output serially from the SAA7392, with the associated EFM clock signal.

7.15.1 EFM Preset Count Register (EFMPresetCount)

Table 112 EFM Preset Count Register (address 3CH) - WRITE

7	6	5	4	3	2	1	0
–	–	Count.5	Count.4	Count.3	Count.2	Count.1	Count.0

Table 113 Description of EFMPresetCount bits

BIT	SYMBOL	DESCRIPTION
7 and 6	–	These 2 bits are reserved.
5 to 0	Count<5:0>	This field can be used to program a position in an EFM frame for the modulator to start outputting data.

7.15.2 EFM Modulator Configuration Register (EFMModConfig)

Table 114 EFM Modulator Configuration Register (address 3DH) - WRITE

7	6	5	4	3	2	1	0
EFMClkSel	ModifiedEFM	EnableLWRT	LWRTOn	GateClkOn	DataSel.1	DataSel.0	PanicOn

Table 115 Description of EFMModConfig bits

BIT	SYMBOL	DESCRIPTION
7	EFMClkSel	If EFMClkSel = 0, then EFM clock generator outputs normal XEFM frequency clock. If EFMClkSel = 1, then EFM clock generator outputs double frequency XEFM clock.
6	ModifiedEFM	If ModifiedEFM = 0, then normal EFM output. If ModifiedEFM = 1, then all ‘0’ output symbols are increased in length by one bit.
5	EnableLWRT	If EnableLWRT = 0, then LaserOn is off. If EnableLWRT = 1, then LaserOn will be set when WriteOn1 is turned on.
4	LWRTOn	If LWRTOn = 0, then LaserOn determined by EnableLWRT bit. If LWRTOn = 1, then LaserOn output is enabled unconditionally.
3	GateClkOn	If GateClkOn = 0, then XEFM clock is on if WriteOn2 is set. If GateClkOn = 1, then XEFM clock is gated on unconditionally.
2	DataSel.1	These 2 bits along with the DataSel.2 bit in register EFMCON2 select the output data format; see Table 119.
1	DataSel.0	
0	PanicOn	If PanicOn = 1, the Panic input is enabled. LaserOn will be disabled immediately when the PANIC pin (pin 27) is logic 1.

7.15.3 EFM Modulator Configuration Register 2 (EFMModConfig2))

Table 116 EFM Modulator Configuration Register 2 (address 3EH) - WRITE

7	6	5	4	3	2	1	0
–	–	–	–	–	TIM2Mode	TIM2ClkOn	DataSel.2

Table 117 Description of EFMModConfig2 bits

BIT	SYMBOL	DESCRIPTION
7 to 3	–	These 5 bits are reserved.
2	TIM2Mode	These 2 bits control the XEFM output; see Table 118.
1	TIM2ClkOn	
0	DataSel.2	This bit along with the DataSel.1 and DataSel.0 bits in register EFMModConfig select the output data format; see Table 119.

Table 118 Control of the XEFM output

TIM2Mode	TIM2ClkOn	
0	X	XEFM is operated according to GateClkOn bit setting in the EFMCON register (address 3DH).
1	0	XEFM output is not influenced by LaserOn/WriteOn and is off.
1	1	XEFM is gated on.

Table 119 Selection of output data format

DataSel.2	DataSel.1	DataSel.0	OUTPUT DATA FORMAT
0	0	0	hold data output at zero
0	0	1	output normal data
0	1	0	output I3 pattern
0	1	1	output I11 pattern
1	0	1	output special OPC pattern 3-3-4-4-5-5-6-6-7-7
X	X	X	All other settings are reserved.

7.16 The EFM clock generator

The EFM clock generator will produce the recording clock based on one of three reference sources. There are five stages to the function. The first selects the reference source using a multiplexer. The next stage is a digital PLL to up-multiply the reference source. The source will determine the base frequency output by the PLL, whilst a position error signal is used to alter the frequency so that the position error tends to zero. Next, an analog PLL is used to up-multiply the output from the digital PLL. The output of the analog PLL is then multiplexed with an external source to allow the EFM clock generator to be bypassed completely. Finally, a programmable divider is used to enable the clock output, XEFM, to be doubled producing XEFM_2.

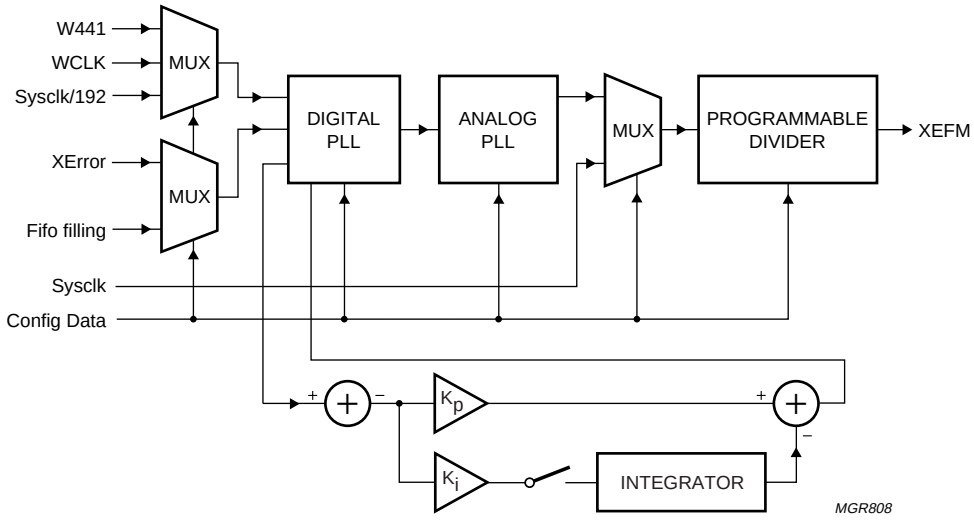


Fig.18 EFM PLL block diagram.

7.16.1 EFM Clock Configuration Register 1 (EFMClockConf1)

Table 120 EFM Clock Configuration Register 1 (address 34H) - WRITE

7	6	5	4	3	2	1	0
DPLLBW.2	DPLLBW.1	DPLLBW.0	Bypass	Div.2	Div.1	Div.0	–

Table 121 Description of EFMClockConf1 bits

BIT	SYMBOL	DESCRIPTION
7	DPLLBW.2	These 3 bits select the digital PLL bandwidth; see Table 122.
6	DPLLBW.1	
5	DPLLBW.0	
4	Bypass	When Bypass = 0, then bypass is off. When Bypass = 1, then PLL is bypassed and the XEFM clock is derived from the system clock.
3	Div.2	These 3 bits select the analog PLL output divisor; see Table 123.
2	Div.1	
1	Div.0	
0	–	This bit is reserved.

Table 122 Selection of PLL bandwidth

DPLLBW.2	DPLLBW.1	DPLLBW.0	PLL BANDWIDTH
0	0	0	Digital PLL bandwidth is 400 Hz.
0	0	1	Digital PLL bandwidth is 200 Hz.
0	1	0	Digital PLL bandwidth is 100 Hz.
0	1	1	Digital PLL bandwidth is 50 Hz.
1	0	0	Digital PLL bandwidth is 25 Hz.

Table 123 Selection of analog PLL output divisor

Div.2	Div.1	Div.0	PLL DIVISOR
0	0	0	Analog PLL output is divided by 1 (XEFM_2 output is not available).
0	0	1	Analog PLL output is divided by 2.
0	1	0	Analog PLL output is divided by 3 (XEFM_2 output is not available).
0	1	1	Analog PLL output is divided by 4.
1	0	0	Analog PLL output is divided by 6.
1	0	1	Analog PLL output is divided by 8.
1	1	0	Analog PLL output is divided by 12.
1	1	1	Analog PLL output is divided by 16.

7.16.2 EFM Clock Configuration Register 2 (EFMClockConf2)

Table 124 EFM Clock Configuration Register 2 (address 35H) - WRITE

7	6	5	4	3	2	1	0
FreqSrc.1	FreqSrc.0	PosSrc.1	PosSrc.0	DPLLMF.3	DPLLMF.2	DPLLMF.1	DPLLMF.0

Table 125 Description of EFMClockConf2 bits

BIT	SYMBOL	DESCRIPTION
7	FreqSrc.1	These 2 bits select the frequency source for the PLL; see Table 126.
6	FreqSrc.0	
5	PosSrc.1	These 2 bits select the position source; see Table 127.
4	PosSrc.0	
3	DPLLMF.3	These 4 bits select the multiplication factor for the PLL; see Table 128
2	DPLLMF.2	
1	DPLLMF.1	
0	DPLLMF.0	

Table 126 Selection of PLL frequency source

FreqSrc.1	FreqSrc.0	FREQUENCY SOURCE
0	0	PLL locks to W441. Use this signal as frequency source to lock EFM frequency to disc wobble.
0	1	PLL locks to WCLK. Use this signal if encoding must slave to the word clock of some external I^2S/EBU input.
1	0	PLL locks to sysclock/192
1	1	This setting is reserved.

Table 127 Selection of position source

PosSrc.1	PosSrc.0	POSITION SOURCE
0	0	No position error used.
1	0	Position source is XError<6:0>. Use this source if EFM clock is locked to ATIP carrier (W441).
1	1	Position source is FifoFill<6:0>. Use this source is EFM clock is locked to WCLK signal.
X	X	All other combinations are reserved.

Table 128 Digital PLL multiplication factor

DPLLMF.3	DPLLMF.2	DPLLMF.1	DPLLMF.0	DIGITAL PLL MULTIPLICATION FACTOR
0	0	0	0	294
0	0	0	1	196
0	0	1	0	147
0	1	0	0	98
0	1	1	0	73.5
1	0	0	0	49
1	0	1	0	36.75

7.16.3 EFM Clock Configuration Register 3 (EFMClockConf3)

This is a dual-function register, the specific function is controlled by the state of bit 7.

The integrator of the PI controller can be preset by writing to this register with the MSB set to logic 0. The value written is interpreted as a signed value. The current integrator value can be read back via this register.

To program the K_p and K_i values, bit 7 of the EFMClockConf3 register must be logic 1.

To determine the time constant of the I-branch, the sample frequency of the integrator must be known. This sample frequency is programmed via register EFMClockConf4 when the MSB is logic 1. The LSB is actually the switch to turn off the integrator path. The value in the integrator is not affected by programming this bit, so the integrator can be used as offset when Ki is logic 0.

Table 129 EFM Clock Configuration Register 3 (address 36H) - READ/WRITE

7	6	5	4	3	2	1	0
0	IntegVal.6	IntegVal.5	IntegVal.4	IntegVal.3	IntegVal.2	IntegVal.1	IntegVal.0
1	Kp.2	Kp.1	Kp.0	Ki.3	Ki.2	Ki.1	Ki.0

Table 130 Selection of coefficient K_p

Kp.2	Kp.1	Kp.0	COEFFICIENT K_p
0	0	0	4
0	0	1	2
0	1	0	1
0	1	1	0.5
1	0	0	0.25

Table 131 Selection of the coefficient K_i

Ki.3	Ki.2	Ki.1	Ki.0	COEFFICIENT K_i
0	0	0	0	0
0	0	0	1	1
0	0	1	1	0.5
0	1	0	1	0.25
0	1	1	1	0.125
1	0	0	1	0.0625
1	0	1	1	0.03125
1	1	0	1	0.015625
1	1	1	1	0.0078125

7.16.4 EFM Clock Configuration Register 4 (EFMClockConF4)

This is a dual-function register, the specific function is controlled by the state of bit 7.

The sample rate of the integrator may be programmed by writing to this register when the MSB is set to a logic 1. Higher sample rates let the integrator work faster. This is effectively the same as increasing the K_i value.

Good stability and critical damping of the integrator can be expected with a sample rate of $^1/_{98}$ EFM frames and $K_i = 2^{-7}$ at n = 2.

The position error set-point may be programmed by writing to this register when the MSB is set to a logic 0. A position error setpoint between −64 and 63 can be programmed using the remaining 7 bits.

On reset this register has the value −8.

Table 132 EFM Clock Configuration Register 4 (address 37H) - WRITE

7	6	5	4	3	2	1	0
1	–	–	–	–	–	Samplerate.1	Samplerate.0
0	PosErr.6	PosErr.5	PosErr.4	PosErr.3	PosErr.2	PosErr.1	PosErr.0

Table 133 Selection of the integrator sample rate

Samplerate.1	Samplerate.0	SAMPLE RATE
0	0	392
0	1	196
1	0	98
1	1	49

Table 134 EFMPLL settings information

INTENDED ENCODER SPEED	DIGITAL MULTIPLICATION FACTOR	ANALOG PLL MULTIPLICATION FACTOR	PROGRAMMABLE DIVIDER FACTOR	POSITION ERROR TIME CONSTANT
n = 1	294	4	12	2581 ms
n = 2	147	4	6	635 ms
n = 4	98	4	4	160 ms

7.17 The Wobble processor

The Wobble processor is a critical part of the recording process, and performs two main functions:

- To extract the ATIP data from the wobble signal
- To control the linear disc speed during recording.

The Wobble processor comprises four functions, a front-end ADC, a digital PLL, the ATIP bit detector and the ATIP data read interface.

7.17.1 The Wobble ADC function

This converts the ATIP signal (WIN) to the digital domain.

7.17.2 The Wobble PLL

The PLL has a PI type loop filter. The bandwidth of the loop and the integrator are programmable by the user. The user can also read and write the PLL frequency, the write mode is available to aid PLL lock-in. The block outputs a pulse signal at 44.1 kHz, which is phase modulated by the ATIP signal.

7.17.3 The ATIP bit detector

This function extracts the ATIP information bits from the ADC output and passes them to the ATIP read interface block. It also outputs the ATIPSync pulse. In order to protect the ATIPSyncs, a flywheel mechanism exists to interpolate ATIPSyncs if none are detected. This is a requirement to ensure that the recording process does not get corrupted.

7.17.4 The ATIP read interface

The ATIP data is read by the on-chip microprocessor. Data availability can be checked by polling the ATIPReady bit in the ATIP Status Register. The microprocessor must read both the ATIPData and ATIPDataEnd registers in order to complete the data read process correctly. If this does not happen, a status bit is set to warn the microcontroller.

7.17.5 Wobble Configuration Register 1 (WobbleConfig1)

This is a dual-function register, the specific function is determined by the state of bit 7. When bit 7 = 0, the function is as described in Tables 135 to 138. When bit 7 = 1, the function is as described in Tables 135 and 139.

Table 135 Wobble Configuration Register 1 (address 27H) - WRITE

7	6	5	4	3	2	1	0
0	0	PLLIntBW.2	PLLIntBW.1	PLLIntBW.0	LoopBW.2	LoopBW.1	LoopBW.0
1	ATIPhold	–	WinWidth.4	WinWidth.3	WindWidth.2	WindWidth.1	WindWidth.0

Table 136 Description of WobbleConfig1 bits, bit 7 = 0

BIT	SYMBOL	DESCRIPTION
5	PLLIntBW.2	These 3 bits select the integrator bandwidth; see Table 137.
4	PLLIntBW.1	
3	PLLIntBW.0	
2 to 0	LoopBW<2:0>	These 3 bits select the loop bandwidth; see Table 138. The PLL bandwidth is proportional to the system clock frequency and determines the following performance points: • Loop bandwidth should be approximately equal to bit rate to get good detector performance • Loop bandwidth depends on ATIP signal scaling. Current figures are valid for –6 dB scaling • PLL lock-in range is approximately equal to loop bandwidth.

Table 137 Selection of integrator bandwidth

PLLIntBW.2	PLLIntBW.1	PLLIntBW.0	INTEGRATOR BANDWIDTH
0	0	0	1.32 kHz for 4.26 MHz system clock
0	0	1	662 Hz for 4.26 MHz system clock
0	1	0	331 Hz for 4.26 MHz system clock
0	1	1	166 Hz for 4.26 MHz system clock
1	0	0	83 Hz for 4.26 MHz system clock
1	0	1	42 Hz for 4.26 MHz system clock
1	1	0	21 Hz for 4.26 MHz system clock
1	1	1	10 Hz for 4.26 MHz system clock

Table 138 Selection of loop bandwidth

LOOPBW.2	LOOPBW.1	LOOPBW.0	LOOP BANDWIDTH
0	0	0	10.6 kHz for 4.26 MHz system clock
0	0	1	5.3 kHz for 4.26 MHz system clock
0	1	0	2.65 kHz for 4.26 MHz system clock
0	1	1	1.32 kHz for 4.26 MHz system clock
1	0	0	662 Hz for 4.26 MHz system clock
1	0	1	331 Hz for 4.26 MHz system clock
1	1	0	166 Hz for 4.26 MHz system clock
1	1	1	83 Hz for 4.26 MHz system clock

Table 139 Description of WobbleConfig1 bits, bit 7 = 1

BIT	SYMBOL	DESCRIPTION
6	ATIPhold	When ATIPhold = 1, then Wobble PLL in Hold mode.
5	–	This bit is reserved.
4	WinWidth.4	Determines the half of the width of the window where ATIP syncs are being accepted for resynchronization. The unit of WindowWidth is ATIP clock pulses. As an ATIP frame consists of 42 ATIP clock pulses, the maximum value of WindowWidth that can be programmed is 21. The reset value is logic 0. The OutOfWindow state can be detected by reading the ATIPStatus register.
3	WinWidth.3	
2	WinWidth.2	
1	WinWidth.1	
0	WinWidth.0	

7.17.6 WOBBLE CONFIGURATION REGISTER 2 (WOBBLECONFIG2)

Table 140 Wobble Configuration Register 2 (address 28H) - WRITE

7	6	5	4	3	2	1	0
–	–	PLLLBW.2	PLLLBW.1	PLLLBW.0	PLLHBW.2	PLLHBW.1	PLLHBW.0

Table 141 Description of WobbleConfig2 bits

BIT	SYMBOL	DESCRIPTION
7	–	These 2 bits are reserved.
6	–	
5	PLLLBW.2	These 3 bits select the PLL low-pass bandwidth; see Table 142.
4	PLLLBW.1	
3	PLLLBW.0	
2	PLLHBW.2	These 3 bits select the PLL high-pass bandwidth; see Table 142.
1	PLLHBW.1	
0	PLLHBW.0	

Table 142 Selection of the PLL high/low-pass bandwidth

PLLLBW.2	PLLLBW.1	PLLLBW.0	HIGH/LOW-PASS BANDWIDTH
PLLHBW.2	PLLHBW.1	PLLHBW.0	
0	0	0	43 kHz for 4.26 MHz system clock
0	0	1	21.5 kHz for 4.26 MHz system clock
0	1	0	10.7 kHz for 4.26 MHz system clock
0	1	1	5.4 kHz for 4.26 MHz system clock
1	0	0	2.7 kHz for 4.26 MHz system clock
1	0	1	1.35 kHz for 4.26 MHz system clock
1	1	0	670 Hz for 4.26 MHz system clock
1	1	1	350 Hz for 4.26 MHz system clock

7.17.7 ATIP STATUS REGISTER (ATIPSTATUS)

Table 143 ATIP Status Register (address 29H) - READ

7	6	5	4	3	2	1	0
ATIPReady	Busy	CRCOK	–	–	–	SyncErr	OIS

Table 144 Description of ATIPStatus bits

BIT	SYMBOL	DESCRIPTION
7	ATIPReady	If ATIPReady = 0, then ATIP data is not ready. If ATIPReady = 1, then ATIP data is ready for read. Can also be an interrupt source.
6	Busy	If Busy = 0, then ATIP data is not being read. If Busy = 1, then sub-CPU is busy reading ATIP data.
5	CRCOK	If CRCOK = 0, then CRC check is not OK. If CRCOK = 1, then CRC check is OK.
4	–	These 3 bits are reserved.
3	–	
2	–	
1	SyncErr	If SyncErr = 1, then detected sync is not in-phase with the interpolated sync. Resync will be done on next detected sync.
0	OIS	If OIS = 1, then sync has been interpolated, no detected sync.

7.17.8 WOBBLE FREQUENCY REGISTER 1 (WOBBLEFREQ1)

Table 145 Wobble Frequency Register 1 (address 2AH) - WRITE/READ

7	6	5	4	3	2	1	0
PLLFreq.15	PLLFreq.14	PLLFreq.13	PLLFreq.12	PLLFreq.11	PLLFreq.10	PLLFreq.9	PLLFreq.8

Table 146 Description of WobbleFreq1 bits

BIT	SYMBOL	DESCRIPTION
7 to 0	PLLFreq<15:8>	These 8 bits form the most significant byte of the PLL frequency.

7.17.9 WOBBLE FREQUENCY REGISTER 2 (WOBBLEFREQ2)

Table 147 Wobble Frequency Register 2 (address 2BH) - READ/WRITE

7	6	5	4	3	2	1	0
PLLFreq.7	PLLFreq.6	PLLFreq.5	PLLFreq.4	PLLFreq.3	PLLFreq.2	PLLFreq.1	PLLFreq.0

Table 148 Description of WobbleFreq2 bits

BIT	SYMBOL	DESCRIPTION
7 to 0	PLLFreq<7:0>	These 8 bits form the least significant byte of the PLL frequency.

7.17.10 ATIP Data Register (ATIPData)

The upper 16 bits of the ATIP data can be obtained from this register by carrying out consecutive read operations.

Table 149 ATIP Data Register (address 2CH) - READ

7	6	5	4	3	2	1	0
ATIPData.23	ATIPData.22	ATIPData.21	ATIPData.20	ATIPData.19	ATIPData.18	ATIPData.17	ATIPData.16
ATIPData.15	ATIPData.14	ATIPData.13	ATIPData.12	ATIPData.11	ATIPData.10	ATIPData.9	ATIPData.8

Table 150 Description of ATIPData bits

BIT	SYMBOL	DESCRIPTION
7 to 0	ATIPData<23:16>	most significant byte of ATIP data
7 to 0	ATIPData<15:8>	penultimate byte of ATIP data

7.17.11 ATIP Data End Register (ATIPDataEnd)

Table 151 ATIP Data End Register (address 2DH) - READ

7	6	5	4	3	2	1	0
ATIPData.7	ATIPData.6	ATIPData.5	ATIPData.4	ATIPData.3	ATIPData.2	ATIPData.1	ATIPData.0

Table 152 Description of ATIPDataEnd bits

BIT	SYMBOL	DESCRIPTION
7 to 0	ATIPData<7:0>	least significant byte of ATIP data

7.17.12 Wobble Peak Status Register (WobbleStatus)

Table 153 Wobble Peak Status Register (address 2EH) - READ

7	6	5	4	3	2	1	0
WPPV.7	WPPV.6	WPPV.5	WPPV.4	WPPV.3	WPPV.2	WPPV.1	WPPV.0

Table 154 Description of WobbleStatus bits

BIT	SYMBOL	DESCRIPTION
7 to 0	WPPV<7:0>	Peak value of the bit signal recovered by the Wobble processor.

7.17.13 ATIP Error Register (ATER)

This register is an unbuffered counter and is incremented on every ATIP code CRC error. When the register reaches 255 it will hold and is reset by being read.

Table 155 ATIP Error Register (address 38H) - READ

7	6	5	4	3	2	1	0
ATER.7	ATER.6	ATER.5	ATER.4	ATER.3	ATER.2	ATER.1	ATER.0

8 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 60134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{DDE}	supply voltage - pad output drivers	notes 1 and 2	−0.5	+4.0	V
V_{DDD}	supply voltage - core/pad ring	notes 1 and 2	−0.5	+4.0	V
V_{DDA}	supply voltage - analog	notes 1 and 2	−0.5	+4.0	V
$V_{i(max)}$	maximum input voltage (any input)		−0.5	+5.0	V
V_O	output voltage (any output)		−0.5	note 3	V
ΔV_{DDD}	voltage difference between V_{DDA} and V_{DDD}		−	0.25	V
I_O	output current (continuous)		−	20	mA
I_{IK}	DC input diode current (continuous)		−	20	mA
T_{amb}	operating ambient temperature		0	+70	°C
T_{stg}	storage temperature		−55	+125	°C
V_{es1}	electrostatic handling	note 4	−1000	+1000	V
V_{es2}	electrostatic handling	note 5	−100	+100	V

Notes

1. All pad driver supply connections (V_{DDE}) and analog and digital core/pad ring supply connections (V_{DDA} and V_{DDD}) must be made externally to the same power supply.
2. All V_{SS} pins must be connected to the same external voltage.
3. (V_{DD} + 0.5) or 4.1 V depending on which one is lower.
4. Equivalent to discharging a 100 pF capacitor via a 1.5 kΩ series resistor with a rise time of 15 ns.
5. Equivalent to discharging a 200 pF capacitor via a 0.75 μH series inductor.

9 OPERATING CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Supply						
V_{DDD}	supply voltage (core/pad ring)		3.0	3.3	3.6	V
V_{DDA}	supply voltage (analog)		3.0	3.3	3.6	V
V_{DDE}	supply voltage (pad output drivers)		3.0	3.3	3.6	V
I_{DD}	supply current		–	200	–	mA
Digital inputs						
SCL (CMOS)						
V_{IL}	LOW-level input voltage		–	–	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	–	5.0	V
I_{LI}	input leakage current	V_i = 0 to V_{DD}	–10	–	+10	µA
C_i	input capacitance		–	–	10	pF
TEST1, TEST2, OTD, MUXSWI, PANIC, PORE, $\overline{WR}$I, $\overline{RD}$I, ALE, CSI, PCAIN, DATAI, SUB, SFSY, T2 AND T1						
V_{IL}	LOW-level input voltage		–	–	0.8	V
V_{IH}	HIGH-level input voltage		2.0	–	5.0	V
I_{LI}	input leakage current	V_i = 0 to V_{DD}	–10	–	+10	µA
C_i	input capacitance		–	–	10	pF
Digital outputs						
W441, ATPSYC, CL1, INT, STOPCK, V4, EBUOUT, SYNC, FLAG, DATAO, RCK, LASERON, XEFM, EFMDATA, CFLG AND MEAS1						
V_{OL}	LOW-level output voltage	I_{OL} = 4 mA	–	–	0.4	V
V_{OH}	HIGH-level output voltage	I_{OH} = –4 mA	$0.85V_{DD}$	–	–	V
C_L	load capacitance		–	–	20	pF
t_r	output rise time	0.4 to (V_{DD} – 0.4); C_L = 20 pF	–	–	20	ns
t_f	output fall time	(V_{DD} – 0.4) to 0.4; C_L = 20 pF	–	–	20	ns
MOTO1 (3-STATE)						
V_{OL}	LOW-level output voltage	I_{OL} = 4 mA	–	–	0.4	V
V_{OH}	HIGH-level output voltage	I_{OH} = –4 mA	$0.85V_{DD}$	–	–	V
C_L	load capacitance		–	–	20	pF
t_r	output rise time	0.4 to (V_{DD} – 0.4); C_L = 20 pF	–	–	20	ns
t_f	output fall time	(V_{DD} – 0.4) to 0.4; C_L = 20 pF	–	–	20	ns
I_{LI}	3-state leakage current	V_i = 0 to V_{DD}	–10	–	+10	µA
Digital I/O						
DA7 TO DA0, WCLK, BCLK AND MOTO2/T3						
V_{IL}	LOW-level input voltage		–	–	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	–	5.0	V

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
I_{LI}	3-state leakage current	V_i = 0 to V_{DD}	–10	–	+10	µA
C_i	input capacitance		–	–	10	pF
V_{OL}	LOW-level output voltage	I_{OL} = 4 mA	–	–	0.4	V
V_{OH}	HIGH-level output voltage	I_{OH} = –4 mA	$0.85V_{DD}$	–	–	V
C_L	load capacitance		–	–	20	pF
t_r	output rise time	0.4 to (V_{DD} – 0.4); C_L = 20 pF	–	–	20	ns
t_f	output fall time	(V_{DD} – 0.4) to 0.4; C_L = 20 pF	–	–	20	ns
SDA (I^2C-BUS)						
V_{IL}	LOW-level input voltage		–	–	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	–	5.0	V
I_{LI}	3-state leakage current	V_i = 0 to V_{DD}	–10	–	+10	µA
C_i	input capacitance		–	–	10	pF
V_{OL}	LOW-level output voltage	I_{OL} = 4 mA	0	–	0.4	V
I_{OL}	LOW-level output current		–	–	8	mA
C_L	load capacitance		–	–	20	pF
t_f	output fall time	C_L = 20 pF	–	–	250	ns
Crystal oscillator						
XTLI (INPUT)						
V_{IL}	LOW-level input voltage		–	–	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	–	5.0	V
I_{LI}	input leakage current		–10	–	+10	µA
C_i	input capacitance		–	–	–	pF
XTLO (OUTPUT)						
f_{xtal}	crystal frequency		4	8	–	MHz
g_m	mutual conductance	f = 100 kHz	–	–	–	mA/V
G_v	small signal voltage gain	$G_v = gm.R_O$	–	–	–	V/V
C_F	feedback capacitance		–	–	–	pF
C_o	output capacitance		–	–	–	pF

9.1 ADC and AGC parameters

Table 156 ADC parameters

PARAMETER	VALUE	COMMENTS
Resolution	8-bit	
Data format	Simple Binary	
Maximum sample frequency	70 MS/s	nominal rate 50 MS/s
SINAD	>26 dB	17.5 MHz signal – 12 dB down on ADC full scale, 70 MS/s sample rate
DNL	–1.0/+1.0 LSB	70 MS/s; 1 MHz input
INL	–2.0/+2.0 LSB	70 MS/s; 1 MHz input
Power dissipation	<100 mW	
Input range	1.47 – 2.91 V	
Voltage reference	2.9 V	top ladder reference
Bias current	100 μA	provided externally
Bandwidth	35 MHz	–3 dB (–0.5 effective bits)
Clock frequency	70 MHz	defines sample rate
Clock duty cycle	40 to 60%	
Clock jitter	±200 ps	for 6.5-bit accuracy, f = 17.5 MHz

Table 157 AGC parameters

PARAMETER	VALUE	COMMENTS
Output level	1.4 V(p-p)	
Gain range	–2.69 to +11.25 dB	approx. linear in dBs
Gain step	1.1 dB	
Bandwidth	35 MHz	–3 dB point
Phase linearity	0.6 ns	in bandwidth of 35 MHz
Resolution	5-bit	
SNR	50 dB	in bandwidth of 35 MHz
THD	<–35 dB	f = 17.5 MHz; $V_{out(p-p)}$ = 1.44 V; ADC load
Input range	0.39 to 1.96 V	peak-to-peak value
Input impedance	7.57 kΩ	nominal, at HIN
Offset voltage	<30 mV	

10 APPLICATION INFORMATION

10.1 Write start control of encoder in CD-ROM mode

In CD-ROM mode, the CDR60 is intended to operate in lock-to-disc mode during write. When writing is stopped by the Encode Control Block, the data flow through the device will also stop. All pointers will stop incrementing and the BCLK clock of the I^2S input interface will stop. As a result, the buffer manager will stop outputting data to the CDR60.

Write start will be given while the encoder is in this 'hot stand-by' state, with all pointers stopped and all clocks stopped. It is extremely important to start the write from a precisely defined initial state. Hence, a 'cold start' must be executed prior to the start of the write. This is as follows:

1. Start the CDR servo. Start the spindle motor servo in 'tacho' mode. Start focus and radial servo loops.
2. Program the Wobble processor to capture ATIP time code data from the disc. (Lock-in the wobble PLL).
3. Jump to a suitable time prior to the 'record start address'. (e.g. 10 frames.)
4. Switch motor servo to ATIP signal. Allow enough time for motor speed to stabilize before start of write. Use motor monitor and wobble monitor functions for this.
5. Lock EFM PLL to crystal clock.
6. Program encoder in 'CD, encode, flush mode'. Program the EFM modulator to 'WriteOn1, WriteOn2 off' mode. This is the reset state.
7. Reset the block encoder.
8. Program the encoder to 'CD encode, normal mode'. The encoder will fill its internal tables, but data flow will stop after a while, because no EFM data is output.
9. Program SubPresetCount and EFMPresetCount registers.
10. Initialize the subcode insert block.
11. Read ATIP time again.
12. Start encoding after suitable delay by writing to EncodeWriteControl. Turn on both WriteOn1 and WriteOn2.

10.2 Write start control of encoder in Audio mode

In this mode the data flow is fixed speed as it is not possible to stop the data stream from the ADC to the CDR60. Because of this, a 'hot start' is needed.

1. Start the CDR servo. Start the spindle motor servo in Tacho mode. Start focus and radial servo loops.
2. Program the Wobble processor to capture ATIP time code from the disc. (Lock-in the wobble PLL).
3. Jump to a suitable time prior to the 'record start address' (e.g. 10 frames).
4. Switch motor servo to ATIP signal. Allow enough time for motor speed to stabilize before start of write. Use motor monitor and wobble monitor functions for this.
5. Lock EFM PLL to I^2S bit clock and FIFO filling.
6. Program encoder in 'CD, encode, flush mode'. Program EFM modulator to WriteOn1 off and WriteOn2 on mode. Program input interface to digital silence.
7. Program the encoder to 'CD, encode, normal mode'. The encoder will start streaming digital zero data. The operation of PLL locked to I^2S bit clock and FIFO filling will regulate the FIFO filling toward a nominal value.
8. Program SubPresetCount and EFMPresetCount registers.
9. Initialize the subcode insert block.
10. Read ATIP time again.
11. Start encoding after suitable delay by writing to EncodeWriteControl. Turn on WriteOn1.

10.3 Start-up of encode in flow-control operation

1. Reset the encoder. Switch LaserOn off, switch WriteOn2 off and switch encoder to 'flush' mode.
2. Preset the device.
   a) Preset the serial input format
   b) Preset the subcode insertion block
   c) Preset the EFM modulator block
   d) Preset the EFM clock generator.
3. Let the disc run at the correct speed, let the EFM clock generator produce the correct clock.
4. At this stage, the device is in Standby mode. Everything is initialized, the encoder is still in 'reset' state and the I^2S input clock is stopped. The encoder cannot accept data because it is reset. Because flow control is enabled the clock is stopped.
5. Initialize the block encoder pointers to point to the correct data from where encoding should start.
6. Switch the encoder from 'flush' to 'encode play' mode. At this point the encoder starts reading data from the serial interface but stops again after a few milliseconds because the its FIFO gets filled as the data out to the EFM modulator is still blocked.

7. When the correct start address is found on the disc, the LaserOn and WriteOn2 signals are switched on at the same time. As a result data flow in the encoder starts while data is written to the disc.

10.4 Start-up of encoder in synchronous stream mode

In synchronous stream mode operation is different because the serial input stream cannot be gated off.

1. Reset the encoder. Switch LaserOn off, Switch WriteOn2 off and switch encoder to 'flush' mode.
2. Preset the device.
   a) Preset the serial input format
   b) Preset the subcode insertion block
   c) Preset the EFM modulator block
   d) Preset the EFM clock generator. The EFM clock generator must be programmed to slave to the incoming serial stream.
3. Let the disc run at the required speed. Make the motor servo slave to the W441 wobble clock.
4. Switch on WriteOn2. This signal must be switched on to allow data to leave the encoder when it enters it. (Important to avoid overflow.)
5. At this stage everything is initialized. The input stream is running but does not enter the encoder because it is still in the 'reset' state. The output stream is running too at the same speed as the input speed because the EFM clock generator is setup like this.
6. The encoder is switched from 'flush' to 'encode play' mode. At this point the data stream starts. Data from the serial in interface enters the encoder while data is read out from it at approximately the same speed via the EFM modulator. The FIFO will not overflow. The data is discarded and not written to the disc because LaserOn is still switch off.
7. When the correct start address is found on the disc LaserOn is switched on allowing data stream to the disc.

10.5 Operating CDR60 in CAV mode, flow control on input stream

For very high-speed operation it is possible to lock the EFM clock to the ATIP wobble carrier. To do this W441 must be selected as the clock source while the position error is taken from XError. The XError signal gives the time difference between the ATIPSync and the QSync. Programming the EFM clock generator like this with a multiplication factor of 98 will produce an EFM clock that is synchronous with the ATIP carrier. It can be used to write data in lock-to-disc mode.

10.6 Operating in CLV Mode, Flow Control on Input Stream

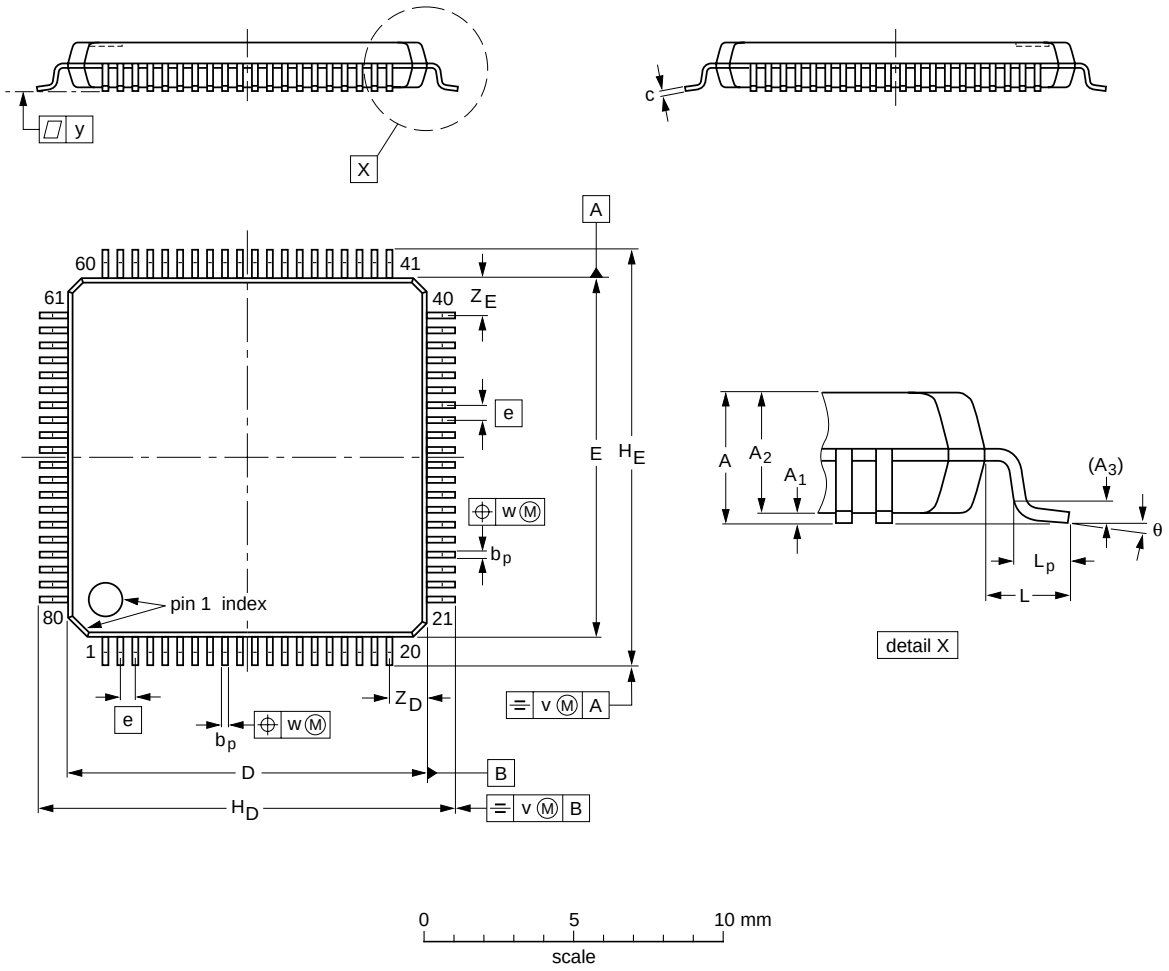
Simply bypass the EFM PLL and use the system clock as EFM source.

10.7 Operating in CLV Mode, Synchronous Stream Operation

When an audio stream is input to the CDR60 encoder the EFM clock must lock to the serial input WCLK clock signal. The position error is taken from the FIFOFill signal. This will produce an EFM clock that is 98 times the WCLK frequency.

11 PACKAGE OUTLINE

LQFP80: plastic low profile quad flat package; 80 leads; body 12 x 12 x 1.4 mm **SOT315-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	1.6	0.16 0.04	1.5 1.3	0.25	0.27 0.13	0.18 0.12	12.1 11.9	12.1 11.9	0.5	14.15 13.85	14.15 13.85	1.0	0.75 0.30	0.2	0.15	0.1	1.45 1.05	1.45 1.05	7^o 0^o

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT315-1	136E15	MS-026				~~99-12-27~~ 00-01-19

12 SOLDERING

12.1 Introduction to soldering surface mount packages

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *"Data Handbook IC26; Integrated Circuit Packages"* (document order number 9398 652 90011).

There is no soldering method that is ideal for all surface mount IC packages. Wave soldering is not always suitable for surface mount ICs, or for printed-circuit boards with high population densities. In these situations reflow soldering is often used.

12.2 Reflow soldering

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement.

Several methods exist for reflowing; for example, infrared/convection heating in a conveyor type oven. Throughput times (preheating, soldering and cooling) vary between 100 and 200 seconds depending on heating method.

Typical reflow peak temperatures range from 215 to 250 °C. The top-surface temperature of the packages should preferable be kept below 230 °C.

12.3 Wave soldering

Conventional single wave soldering is not recommended for surface mount devices (SMDs) or printed-circuit boards with a high component density, as solder bridging and non-wetting can present major problems.

To overcome these problems the double-wave soldering method was specifically developed.

If wave soldering is used the following conditions must be observed for optimal results:

- Use a double-wave soldering method comprising a turbulent wave with high upward pressure followed by a smooth laminar wave.
- For packages with leads on two sides and a pitch (e):
  - larger than or equal to 1.27 mm, the footprint longitudinal axis is **preferred** to be parallel to the transport direction of the printed-circuit board;
  - smaller than 1.27 mm, the footprint longitudinal axis **must** be parallel to the transport direction of the printed-circuit board.

  The footprint must incorporate solder thieves at the downstream end.
- For packages with leads on four sides, the footprint must be placed at a 45° angle to the transport direction of the printed-circuit board. The footprint must incorporate solder thieves downstream and at the side corners.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Typical dwell time is 4 seconds at 250 °C.
A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

12.4 Manual soldering

Fix the component by first soldering two diagonally-opposite end leads. Use a low voltage (24 V or less) soldering iron applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C.

When using a dedicated tool, all other leads can be soldered in one operation within 2 to 5 seconds between 270 and 320 °C.

12.5 Suitability of surface mount IC packages for wave and reflow soldering methods

PACKAGE	SOLDERING METHOD	
	WAVE	REFLOW[1]
BGA, LFBGA, SQFP, TFBGA	not suitable	suitable
HBCC, HLQFP, HSQFP, HSOP, HTQFP, HTSSOP, SMS	not suitable[2]	suitable
PLCC[3], SO, SOJ	suitable	suitable
LQFP, QFP, TQFP	not recommended[3][4]	suitable
SSOP, TSSOP, VSO	not recommended[5]	suitable

Notes

1. All surface mount (SMD) packages are moisture sensitive. Depending upon the moisture content, the maximum temperature (with respect to time) and body size of the package, there is a risk that internal or external package cracks may occur due to vaporization of the moisture in them (the so called popcorn effect). For details, refer to the Drypack information in the *"Data Handbook IC26; Integrated Circuit Packages; Section: Packing Methods"*.
2. These packages are not suitable for wave soldering as a solder joint between the printed-circuit board and heatsink (at bottom version) can not be achieved, and as solder may stick to the heatsink (on top version).
3. If wave soldering is considered, then the package must be placed at a 45° angle to the solder wave direction. The package footprint must incorporate solder thieves downstream and at the side corners.
4. Wave soldering is only suitable for LQFP, TQFP and QFP packages with a pitch (e) equal to or larger than 0.8 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.65 mm.
5. Wave soldering is only suitable for SSOP and TSSOP packages with a pitch (e) equal to or larger than 0.65 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.5 mm.

13 DEFINITIONS

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

14 LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

15 PURCHASE OF PHILIPS I²C COMPONENTS

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

NOTES

NOTES

Philips Semiconductors – a worldwide company

Argentina: see South America

Australia: 3 Figtree Drive, HOMEBUSH, NSW 2140,
Tel. +61 2 9704 8141, Fax. +61 2 9704 8139

Austria: Computerstr. 6, A-1101 WIEN, P.O. Box 213,
Tel. +43 1 60 101 1248, Fax. +43 1 60 101 1210

Belarus: Hotel Minsk Business Center, Bld. 3, r. 1211, Volodarski Str. 6,
220050 MINSK, Tel. +375 172 20 0733, Fax. +375 172 20 0773

Belgium: see The Netherlands

Brazil: see South America

Bulgaria: Philips Bulgaria Ltd., Energoproject, 15th floor,
51 James Bourchier Blvd., 1407 SOFIA,
Tel. +359 2 68 9211, Fax. +359 2 68 9102

Canada: PHILIPS SEMICONDUCTORS/COMPONENTS,
Tel. +1 800 234 7381, Fax. +1 800 943 0087

China/Hong Kong: 501 Hong Kong Industrial Technology Centre,
72 Tat Chee Avenue, Kowloon Tong, HONG KONG,
Tel. +852 2319 7888, Fax. +852 2319 7700

Colombia: see South America

Czech Republic: see Austria

Denmark: Sydhavnsgade 23, 1780 COPENHAGEN V,
Tel. +45 33 29 3333, Fax. +45 33 29 3905

Finland: Sinikalliontie 3, FIN-02630 ESPOO,
Tel. +358 9 615 800, Fax. +358 9 6158 0920

France: 51 Rue Carnot, BP317, 92156 SURESNES Cedex,
Tel. +33 1 4099 6161, Fax. +33 1 4099 6427

Germany: Hammerbrookstraße 69, D-20097 HAMBURG,
Tel. +49 40 2353 60, Fax. +49 40 2353 6300

Hungary: see Austria

India: Philips INDIA Ltd, Band Box Building, 2nd floor,
254-D, Dr. Annie Besant Road, Worli, MUMBAI 400 025,
Tel. +91 22 493 8541, Fax. +91 22 493 0966

Indonesia: PT Philips Development Corporation, Semiconductors Division,
Gedung Philips, Jl. Buncit Raya Kav.99-100, JAKARTA 12510,
Tel. +62 21 794 0040 ext. 2501, Fax. +62 21 794 0080

Ireland: Newstead, Clonskeagh, DUBLIN 14,
Tel. +353 1 7640 000, Fax. +353 1 7640 200

Israel: RAPAC Electronics, 7 Kehilat Saloniki St, PO Box 18053,
TEL AVIV 61180, Tel. +972 3 645 0444, Fax. +972 3 649 1007

Italy: PHILIPS SEMICONDUCTORS, Via Casati, 23 - 20052 MONZA (MI),
Tel. +39 039 203 6838, Fax +39 039 203 6800

Japan: Philips Bldg 13-37, Kohnan 2-chome, Minato-ku,
TOKYO 108-8507, Tel. +81 3 3740 5130, Fax. +81 3 3740 5057

Korea: Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL,
Tel. +82 2 709 1412, Fax. +82 2 709 1415

Malaysia: No. 76 Jalan Universiti, 46200 PETALING JAYA, SELANGOR,
Tel. +60 3 750 5214, Fax. +60 3 757 4880

Mexico: 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905,
Tel. +9-5 800 234 7381, Fax +9-5 800 943 0087

Middle East: see Italy

Netherlands: Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB,
Tel. +31 40 27 82785, Fax. +31 40 27 88399

New Zealand: 2 Wagener Place, C.P.O. Box 1041, AUCKLAND,
Tel. +64 9 849 4160, Fax. +64 9 849 7811

Norway: Box 1, Manglerud 0612, OSLO,
Tel. +47 22 74 8000, Fax. +47 22 74 8341

Pakistan: see Singapore

Philippines: Philips Semiconductors Philippines Inc.,
106 Valero St. Salcedo Village, P.O. Box 2108 MCC, MAKATI,
Metro MANILA, Tel. +63 2 816 6380, Fax. +63 2 817 3474

Poland: Al.Jerozolimskie 195 B, 02-222 WARSAW,
Tel. +48 22 5710 000, Fax. +48 22 5710 001

Portugal: see Spain

Romania: see Italy

Russia: Philips Russia, Ul. Usatcheva 35A, 119048 MOSCOW,
Tel. +7 095 755 6918, Fax. +7 095 755 6919

Singapore: Lorong 1, Toa Payoh, SINGAPORE 319762,
Tel. +65 350 2538, Fax. +65 251 6500

Slovakia: see Austria

Slovenia: see Italy

South Africa: S.A. PHILIPS Pty Ltd., 195-215 Main Road Martindale,
2092 JOHANNESBURG, P.O. Box 58088 Newville 2114,
Tel. +27 11 471 5401, Fax. +27 11 471 5398

South America: Al. Vicente Pinzon, 173, 6th floor,
04547-130 SÃO PAULO, SP, Brazil,
Tel. +55 11 821 2333, Fax. +55 11 821 2382

Spain: Balmes 22, 08007 BARCELONA,
Tel. +34 93 301 6312, Fax. +34 93 301 4107

Sweden: Kottbygatan 7, Akalla, S-16485 STOCKHOLM,
Tel. +46 8 5985 2000, Fax. +46 8 5985 2745

Switzerland: Allmendstrasse 140, CH-8027 ZÜRICH,
Tel. +41 1 488 2741 Fax. +41 1 488 3263

Taiwan: Philips Semiconductors, 6F, No. 96, Chien Kuo N. Rd., Sec. 1,
TAIPEI, Taiwan Tel. +886 2 2134 2886, Fax. +886 2 2134 2874

Thailand: PHILIPS ELECTRONICS (THAILAND) Ltd.,
209/2 Sanpavuth-Bangna Road Prakanong, BANGKOK 10260,
Tel. +66 2 745 4090, Fax. +66 2 398 0793

Turkey: Yukari Dudullu, Org. San. Blg., 2.Cad. Nr. 28 81260 Umraniye,
ISTANBUL, Tel. +90 216 522 1500, Fax. +90 216 522 1813

Ukraine: PHILIPS UKRAINE, 4 Patrice Lumumba str., Building B, Floor 7,
252042 KIEV, Tel. +380 44 264 2776, Fax. +380 44 268 0461

United Kingdom: Philips Semiconductors Ltd., 276 Bath Road, Hayes,
MIDDLESEX UB3 5BX, Tel. +44 208 730 5000, Fax. +44 208 754 8421

United States: 811 East Arques Avenue, SUNNYVALE, CA 94088-3409,
Tel. +1 800 234 7381, Fax. +1 800 943 0087

Uruguay: see South America

Vietnam: see Singapore

Yugoslavia: PHILIPS, Trg N. Pasica 5/v, 11000 BEOGRAD,
Tel. +381 11 3341 299, Fax.+381 11 3342 553

For all other countries apply to: Philips Semiconductors,
International Marketing & Sales Communications, Building BE-p, P.O. Box 218,
5600 MD EINDHOVEN, The Netherlands, Fax. +31 40 27 24825

Internet: http://www.semiconductors.philips.com

© Philips Electronics N.V. 2000 SCA 69

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Printed in The Netherlands 753503/01/pp76 Date of release: 2000 Mar 21 Document order number: 9397 750 04685

Let's make things better.

Philips Semiconductors

PHILIPS