OXFORD SEMICONDUCTOR

# *OX16C950 rev B*
## *High Performance UART with 128 byte FIFOs*

## *FEATURES*

- Single full-duplex asynchronous channel
- 128-byte deep transmitter / receiver FIFO
- Fully software compatible with industry standard 16C550 type UARTs
- Pin compatible with TL16C550B/C, ST16C650 and TL16C750
- IBM PC/AT compatible
- Baud rates up to 15 Mbps in normal mode and 60Mbps in external 1x clock mode
- Readable FIFO levels
- Flexible clock prescaler from 1 to 31.875
- Isochronous mode using external 1x baud rate clock up to 60Mbps
- 9-bit data framing as well as 5,6,7 and 8
- Detection of bad data in the receiver FIFO
- Automated in-band flow control using programmable Xon/Xoff characters
- Transmitter and receiver can be disabled
- Automated out-of-band flow control using CTS# / RTS# and DSR# / DTR#
- Readable in-band and out-of-band flow control status
- Programmable special character detection
- Arbitrary trigger levels for receiver and transmitter FIFO interrupts and automatic in-band and out-of-band flow control
- Transmitter idle interrupt (shift register and FIFO both empty)
- Optional Infra-red (IrDA) receiver and transmitter operation
- RS-485 buffer enable signals
- Software channel reset
- Four byte device ID
- Sleep mode (low operating current)
- System clock up to 60 MHz (at 5V), 50 MHz at 3.3V
- 44 PLCC and 48 TQFP packages
- 5 volts operation (PLCC), 3.3/ 5V operation TQFP

## *REV B ENHANCEMENTS*

The OX16C950B is an enhanced, fully backward-compatible revision of the OX16C950 rev A. The chief enhancements are as follows –

- All known errata fixed
- Enhanced features first offered in OX16PCI954 added – these include controls for sleep-mode sensitivity, ability to read FCR and Good Data Status
- 3V operation possible with 48 pin TQFP
- Enhanced isochronous clocking options (optional inversions)
- Enhanced system clock selection options (use of CLKSEL as a clock input)
- Readable TxRdy, RxRdy status and forcing TxRdy or RxRdy inactive

Hereafter OX16C950 rev B is simply referred to as OX16C950.

**Oxford Semiconductor Ltd.**
**25 Milton Park, Abingdon, Oxon, OX14 4SH, UK**
**Tel: +44 (0)1235 824900 Fax: +44 (0)1235 821141**

**© Oxford Semiconductor 2001**
**OX16C950 rev B Datasheet R1.2 – May 2001**
**Part No. OX16C950-PCC60-B**

## *DESCRIPTION*

The OX16C950 is a single-channel ultra-high performance UART offering data rates up to 15Mbps and 128-deep transmitter and receiver FIFOs. Deep FIFOs reduce CPU overhead and allow utilisation of higher data rates.

It is software compatible with the widely used industry-standard 16C550 type devices and compatibles, as well as other OX16C95x family devices. It is pin-compatible with the TL16C550, ST16C650 devices.

In addition to increased performance and FIFO size, the OX16C950 also provides enhanced features including improved flow control. Automated software flow control using Xon/Xoff and automated hardware flow control using CTS#/RTS# and DSR#/DTR# prevent FIFO over-run. Flow control and interrupt thresholds are fully programmable and readable, enabling programmers to fine-tune the performance of their system. FIFO levels are readable to facilitate fast driver applications.

The addition of software reset enables recovery from unforeseen error condition allowing drivers to restart gracefully. The OX16C950 supports 9-bit data frames used in multi-drop industrial protocols. It also offers multiple external clock options for isochronous applications, e.g. ISDN, xDSL.

The OX16C950 is ideally suited to PC applications, such as high-speed COM port add-in cards which enable PC users to take advantage of the maximum performance of analogue modems or ISDN terminal adapters. It is also suitable for any equipment requiring high speed RS232/RS422/RS485 interfaces. Fabricated in 0.6μm process, OX16C950 also has a low operating current and sleep mode for battery powered applications.

**Oxford Semiconductor Ltd.**
**25 Milton Park, Abingdon, Oxon, OX14 4SH, UK**
**Tel: +44 (0)1235 824900 Fax: +44 (0)1235 821141**

**© Oxford Semiconductor 2001**

## CONTENTS


**FEATURES** ..... **1**
**REV B ENHANCEMENTS** ..... **1**
**DESCRIPTION** ..... **2**
**CONTENTS** ..... **3**
**1 PERFORMANCE COMPARISON** ..... **5**
**2 BLOCK DIAGRAM** ..... **6**
**3 PIN INFORMATION** ..... **7**
**4 PIN DESCRIPTIONS** ..... **8**
4.1 FURTHER PIN INFORMATION ..... 11
**5 MODE SELECTION** ..... **13**
5.1 450 MODE ..... 13
5.2 550 MODE ..... 13
5.3 EXTENDED 550 MODE ..... 13
5.4 750 MODE ..... 13
5.5 650 MODE ..... 13
5.6 950 MODE ..... 14
**6 REGISTER DESCRIPTION TABLES** ..... **15**
**7 RESET CONFIGURATION** ..... **19**
7.1 HARDWARE RESET ..... 19
7.2 SOFTWARE RESET ..... 19
**8 TRANSMITTER & RECEIVER FIFOS** ..... **20**
8.1 FIFO CONTROL REGISTER 'FCR' ..... 20
**9 LINE CONTROL & STATUS** ..... **21**
9.1 FALSE START BIT DETECTION ..... 21
9.2 LINE CONTROL REGISTER 'LCR' ..... 21
9.3 LINE STATUS REGISTER 'LSR' ..... 22
**10 INTERRUPTS & SLEEP MODE** ..... **23**
10.1 INTERRUPT ENABLE REGISTER 'IER' ..... 23
10.2 INTERRUPT STATUS REGISTER 'ISR' ..... 24
10.3 INTERRUPT DESCRIPTION ..... 24
10.4 SLEEP MODE ..... 25
**11 MODEM INTERFACE** ..... **25**
11.1 MODEM CONTROL REGISTER 'MCR' ..... 25
11.2 MODEM STATUS REGISTER 'MSR' ..... 26
**12 OTHER STANDARD REGISTERS** ..... **27**
12.1 DIVISOR LATCH REGISTERS 'DLL & DLM' ..... 27
12.2 SCRATCH PAD REGISTER 'SPR' ..... 27
**13 AUTOMATIC FLOW CONTROL** ..... **28**
13.1 ENHANCED FEATURES REGISTER 'EFR' ..... 28
13.2 SPECIAL CHARACTER DETECTION ..... 29
13.3 AUTOMATIC IN-BAND FLOW CONTROL ..... 29
13.4 AUTOMATIC OUT-OF-BAND FLOW CONTROL ..... 29
**14 BAUD RATE GENERATION** ..... **30**
14.1 GENERAL OPERATION ..... 30
14.2 CLOCK PRESCALER REGISTER 'CPR' ..... 31
14.3 TIMES CLOCK REGISTER 'TCR' ..... 31
14.4 INPUT CLOCK OPTIONS ..... 33
14.5 TTL CLOCK MODULE ..... 33

14.6 EXTERNAL 1X CLOCK MODE ..... 33
14.7 CRYSTAL OSCILLATOR CIRCUIT ..... 33
**15 ADDITIONAL FEATURES ..... 34**
15.1 ADDITIONAL STATUS REGISTER 'ASR' ..... 34
15.2 FIFO FILL LEVELS 'TFL & RFL' ..... 34
15.3 ADDITIONAL CONTROL REGISTER 'ACR' ..... 34
15.4 TRANSMITTER TRIGGER LEVEL 'TTL' ..... 36
15.5 RECEIVER INTERRUPT. TRIGGER LEVEL 'RTL' ..... 36
15.6 FLOW CONTROL LEVELS 'FCL & FCH' ..... 36
15.7 DEVICE IDENTIFICATION REGISTERS ..... 36
15.8 CLOCK SELECT REGISTER 'CKS' ..... 37
15.9 NINE-BIT MODE REGISTER 'NMR' ..... 37
15.10 MODEM DISABLE MASK 'MDM' ..... 38
15.11 READABLE FCR 'RFC' ..... 38
15.12 GOOD-DATA STATUS REGISTER 'GDS' ..... 39
15.13 DMA STATUS REGISTER 'DMS' ..... 39
15.14 PORT INDEX REGISTER 'PIX' ..... 39
15.15 CLOCK ALTERATION REGISTER 'CKA' ..... 39
**16 OPERATING CONDITIONS ..... 40**
**17 DC ELECTRICAL CHARACTERISTICS ..... 40**
17.1 5V OPERATION ..... 40
17.2 3V OPERATION ..... 41
**18 AC ELECTRICAL CHARACTERISTICS ..... 42**
18.1 5V OPERATION ..... 42
18.2 3V OPERATION ..... 43
**19 TIMING WAVEFORMS ..... 44**
**20 PACKAGE INFORMATION ..... 46**
**21 ORDERING INFORMATION ..... 47**
**NOTES ..... 48**
**CONTACT DETAILS ..... 49**

# 1 PERFORMANCE COMPARISON

| Feature | OX16C950 | 16C450 | 16C550 | 16C650 | 16C750 |
|---|---|---|---|---|---|
| External 1x baud rate clock | Yes | No | No | No | No |
| Max baud rate in normal mode | 15 Mbps | 115 kbps | 115 kbps | 1.5 Mbps | 1 Mbps |
| Max baud rate in 1x clock mode | 60 Mbps | n/a | n/a | n/a | n/a |
| FIFO depth | 128 | 1 | 16 | 32 | 64 |
| Sleep mode | Yes | No | No | Yes | Yes |
| Auto Xon/Xoff flow | Yes | No | No | Yes | No |
| Auto CTS#/RTS# flow | Yes | No | No | Yes | Yes |
| Auto DSR#/DTR# flow | Yes | No | No | No | No |
| No. of Rx interrupt thresholds | 127 | 1 | 4 | 4 | 4 |
| No. of Tx interrupt thresholds | 128 | 1 | 1 | 4 | 1 |
| No. of flow control thresholds | 128 | n/a | n/a | 4 | n/a |
| Transmitter empty interrupt | Yes | No | No | No | No |
| Readable status of flow control | Yes | n/a | No | No | No |
| Readable FIFO levels | Yes | n/a | No | No | No |
| Clock prescaler options | 248 | n/a | n/a | 2 | n/a |
| Rx/Tx disable | Yes | No | No | No | No |
| Software reset | Yes | No | No | No | No |
| Device ID | Yes | No | No | No | No |
| 9-bit data frames | Yes | No | No | No | No |
| RS485 buffer enable | Yes | No | No | No | No |
| Infra-red (IrDA) | Yes | No | No | Yes | No |

**Table 1 OX16C950 performance compared with 16C450, 16C550, 16C650 and 16C750 devices**

Improvements of the OX16C950 over previous generations of PC UART:

**Deeper FIFOs:**
OX16C950 offers 128-byte deep FIFOs for the transmitter and receiver.

**Higher data rates:**
Transmission and reception baud rates up to 15Mbps. A flexible clock prescaler offers division ratios of 1 to 31 7/8 in steps of 1/8 using a divide-by-"M N/8" circuitry. The flexible prescaler allows users to select from a wide variety of input clock frequencies as well as access to higher baud rates whilst maintaining compatibility with existing software drivers (see section 14.2).

**External clock options:**
The receiver can accept an external 1x clock on the DSR# input. The transmitter can accept a 1x clock on the RI# input and/or assert its own (Nx) clock on the DTR# output. In 1x mode, asynchronous data may be transmitted and received at speeds up to 60Mbps (see section 14.6).

**Automatic flow control:**
The UART automatically handles either or both in-band (software) flow control (transmitting and receiving Xon/Xoff characters) and out-of-band (hardware) flow control using the RTS#/CTS# or DSR#/DTR# modem control lines.

**Special character detection:**
The receiver can be programmed to generate an interrupt upon reception of a particular character value.

**Power-down:**
The device can be placed in 'sleep mode' to conserve power.

**Readable FIFO levels:**
Driver efficiency can be improved by using readable FIFO levels.

**Selectable trigger levels:**
The receiver FIFO threshold can be arbitrarily programmed. The transmitter FIFO threshold and thresholds for automatic flow control can be programmed to operate at a variety of trigger levels.

**Additional control:**
The transmitter and receiver can be independently disabled.

**Additional status:**
Software drivers are able to read the status of in-band and out-of-band automatic flow control, and distinguish between Xoff and special character received interrupts.

**Software reset:**
The software driver may reset the device to recover from unforeseen or unusual error conditions.

**Transmitter empty interrupt:**
The transmitter can generate an interrupt when the FIFO and shift register are both empty.

**RS485 buffer enable:**
The DTR# pin may be re-assigned as a buffer-enable signal for RS485 line driver in half-duplex mode (see ACR[4:3] in section 15.3).

**Device ID:**
Four bytes of device ID are available to identify the OX16C950 device to software drivers.

**Infra-red 'IrDA' interface:**
The UART contains an IrDA compliant modulator and demodulator.

**9-bit data framing:**
The OX16C950 may be configured to use in 9-bit character framing for multi-drop protocols where a tag ID (9th bit) differentiates address and data characters.

## 2 BLOCK DIAGRAM

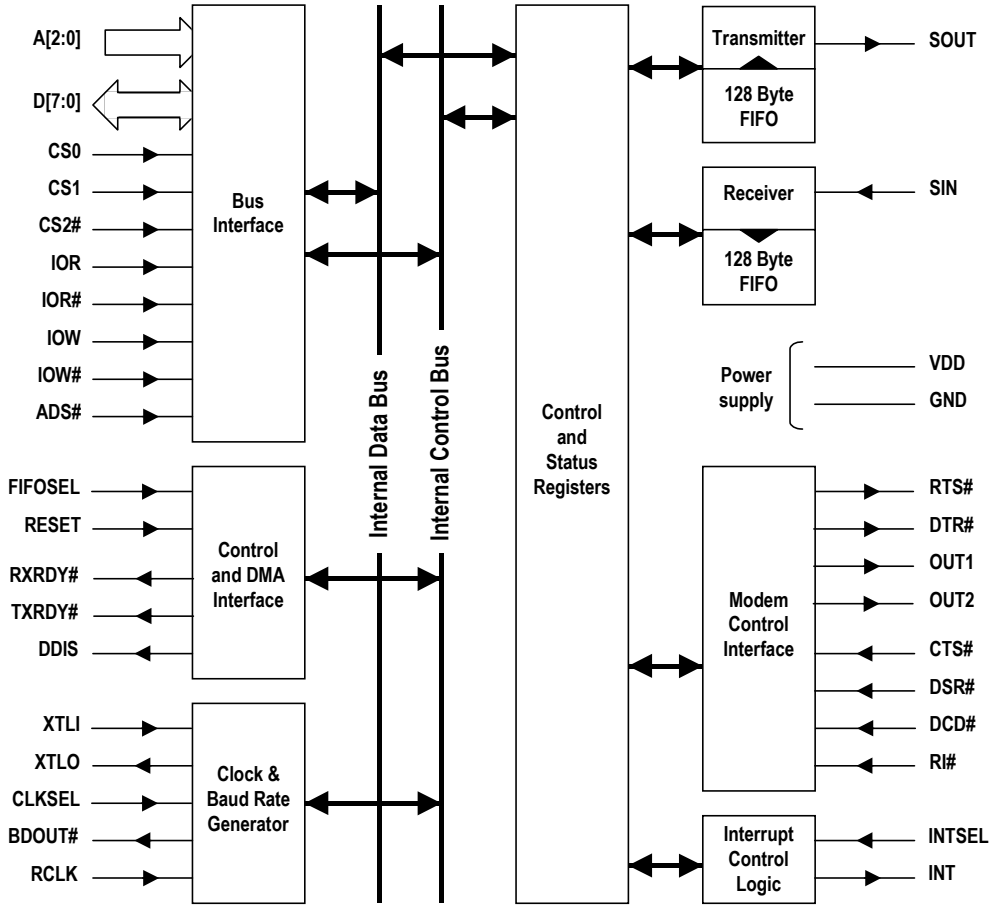
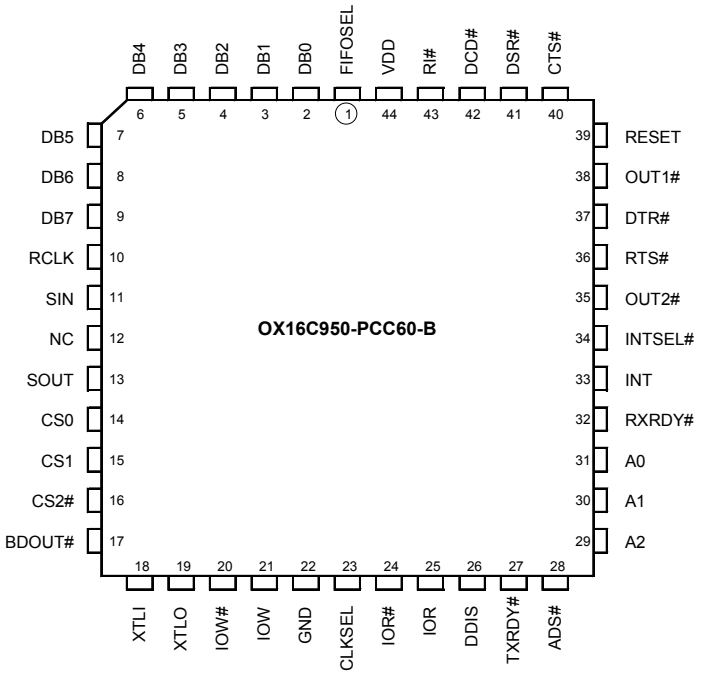


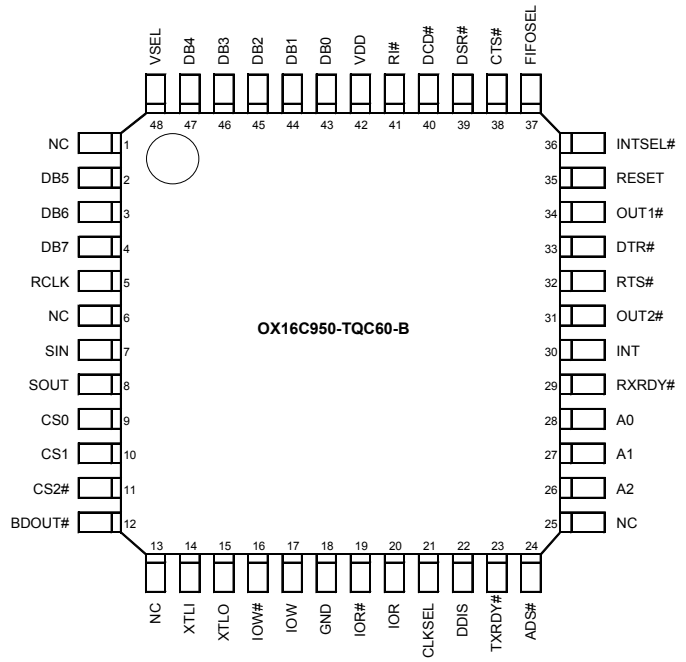

**Figure 1: OX16C950 Block Diagram**

# 3 PIN INFORMATION

## 44 Pin Plastic Leaded Chip Carrier




## 48 Pin Thin Quad Flat Pack



## 4 PIN DESCRIPTIONS

| PLCC | TQFP | Dir[1] | Name | Description |
|---|---|---|---|---|
| **Clock** | | | | |
| 18 | 14 | I | XTLI | Crystal oscillator input or external clock pin.<br>Maximum frequency 60 MHz @ 5V, 50 MHz @ 3.3V |
| 19 | 15 | O | XTLO | Crystal oscillator output. Not used when an alternative TTL level clock is applied to XTLI and can be left unconnected. |
| 23 | 21 | IU | CLKSEL | The state of this pin on power up configures the internal clock prescaler. This pin has an internal pull-up. When CLKSEL pin is high the pre-scalar is bypassed. Connect this pin to GND to enable the internal clock prescaler (see section 14.2). The complement of this pin is loaded in MCR[7] after a hardware reset.<br><br>This pin can also be used as an alternative external clock pin under software control (replacing XTLI and thus reducing noise/power due to XTLO) for embedded applications |
| **Processor Interface** | | | | |
| 39 | 35 | I | RESET | Active-high hardware reset. Hardware reset is described in section 7.1. This pin must be tied inactive when not in use. |
| 14, 15 | 9, 10 | I | CS0,CS1 | Active-high chip select. All chip select pins must be active for the device to be selected. |
| 16 | 11 | I | CS2# | Active-low chip select. |
| 29 -31 | 26 – 28 | I | A[2:0] | Address lines to select channel registers. |
| 28 | 24 | I | ADS# | Active-low address strobe. When ADS# signal is low, the address (A[2:0]) and chip select signal (CS0, CS1, CS2#) drive the internal logic, otherwise they are latched at the level they were when low-to-high transition of ADS# signal occurred. This pin is used when address and chip selects are not stable during read or write cycles. If this functionality is not required, this pin can be permanently tied to GND. |
| 9 - 2 | 4 – 2, 47 – 43 | I/O | DB[7:0] | Eight-bit 3-state data bus. |
| 26 | 22 | O | DDIS | Drive Disable. This pin goes active (high) when CPU is not reading from OX16C950. This signal can be used to disable an external transceiver. |
| 20 | 16 | I | IOW# | Active-low write strobe. When IOW# is used to write the chip, IOW should be tied low (inactive). |
| 21 | 17 | I | IOW | Active-high write strobe. When IOW is used to write the chip, IOW# should be tied high (inactive). |
| 24 | 19 | I | IOR# | Active-low read strobe. When IOR# is used to read from the chip, IOR should be tied low (inactive). |
| 25 | 20 | I | IOR | Active-high read strobe. When IOR is used to read from the chip, IOR# should be tied high (inactive). |

| PLCC | TQFP | Dir[1] | Name | Description |
|---|---|---|---|---|
| **Serial port pins** | | | | |
| 13 | 8 | O | SOUT | Transmitter serial data output. |
| | | O | IrDA_Out | This pin is re-defined to IrDA output when IrDA mode is enabled, i.e. MCR[6] set in Enhanced mode. |
| 36 | 32 | O | RTS# | Active-low Request-To-Send output. Whenever the automated RTS# flow control is enabled, the RTS# pin is de-asserted and re-asserted if the receiver FIFO reaches or falls below a pair of programmed flow control thresholds, respectively. This pin's state is controlled by bit 1 of the MCR. RTS may also be used as a general-purpose output. |
| 37 | 33 | O | DTR# | Active-low modem Data-Terminal-Ready output. Whenever the automated DTR# flow control is enabled, the DTR# pin is asserted and de-asserted if the receiver FIFO reaches or falls below a pair of programmed flow control thresholds, respectively. The state is set by bit 0 of the MCR. DTR may also be used as a general purpose output. |
| | | O | 485_EN | In RS485 half-duplex mode, the DTR# pin may be programmed to reflect the state of the transmitter empty bit (or it's inverse) to automatically control the direction of the RS485 transceiver buffer (see ACR[4:3]). |
| | | O | Tx_Clk_Out | Transmitter 1x (or baud rate generator output) clock. For isochronous applications, the 1x (or Nx) transmitter clock may be asserted on the DTR# pin (see CKS[5:4]). |
| 11 | 7 | I | SIN | Receiver serial data input. |
| | | I | IrDA_In | This pin is re-defined to IrDA input when IrDA mode is enabled, i.e. MCR[6] set in Enhanced mode. |
| 40 | 38 | I | CTS# | Active-low Clear-To-Send input. Whenever the automated CTS# flow control is enabled and the CTS# pin is de-asserted, the transmitter will complete the current character and enter the idle mode until the CTS# pin is re-asserted. However, flow control characters are transmitted regardless of the state of the CTS# pin. The state of this pin is reflected in bit 4 of the MSR. It can also be used as a general-purpose input. |
| 41 | 39 | I | DSR# | Active-low modem Data-Set-Ready input. Whenever the automated DSR# flow control is enabled and the DSR# pin is de-asserted, the transmitter will complete the current character and enter the idle mode until the DSR# pin is re-asserted. However, flow control characters are transmitted regardless of the state of the DSR# pin. The state of this pin is reflected in bit 5 of the MSR. It can also be used as a genera- purpose input. |
| | | I | Rx_Clk_In | External receiver clock for isochronous applications. The Rx_Clk_In is selected when CKS[1:0] = '01'. |
| 42 | 40 | I | DCD# | Active-low modem Data-Carrier-Detect input. The state of this pin is reflected in bit 7 of the MSR. It can also be used as a general-purpose input |
| 43 | 41 | I | RI# | Active-low modem Ring-Indicator input. The state of this pin is reflected in bit 6 of the MSR. It can also be used as a general-purpose input. RI can be configured as tx and rx for a 1x clock in isochronous operation. |
| | | I | Tx_Clk_In | External transmitter clock. This clock can be used by the transmitter (and by the receiver indirectly) when CKS[6]='1'. |
| 17 | 12 | O | BDOUT# | Baud out. BDOUT# is a Nx (usually 16x, see TCR) clock signal for the transmitter. It is the output of the baud generator module. The receiver can use this clock by connecting BDOUT# to the RCLK pin or setting CKS[1:0] to '10' where BDOUT# will be connected to RCLK internally. In this case setting CKS[2] to '1' will disable the BDOUT# pin to conserve power. |
| 10 | 5 | I | RCLK | Receiver clock. RCLK is the Nx (usually 16x, see TCR) baud rate clock for the receiver. |

| PLCC | TQFP | Dir[1] | Name | Description |
|---|---|---|---|---|
| **Interrupt & DMA Pins** | | | | |
| 33 | 30 | O | INT | The serial channel has a three-state interrupt output. This signal goes active (high) when an interrupt condition occurs. The three-state logic is controlled by INTSEL# and MCR[3] as described below. |
| 27 | 23 | O | TXRDY# | Signal for DMA transfer of transmitter data. There are two modes of DMA signalling described in section 8.1. |
| 32 | 29 | O | RXRDY# | Signal for DMA transfer of received data. There are two modes of DMA signalling described in section 8.1. |
| 34 | 36 | IU | INTSEL# | Active-low interrupt select. This pin has an internal pull-up resistor. When INTSEL# is high or unconnected, the INT pin is enabled and MCR[3] is ignored. When INTSEL# is low, the tri-state control of INT is controlled by MCR[3]. In this case INT is enabled when MCR[3] is set and is high-impedance when MCR[3] is low.<br>This pin is used to save the external three-state buffer for the interrupt pin. When using this facility, the INT output should be pulled down to GND using a 1KΩ resistor. |
| **Miscellaneous Pins** | | | | |
| 38 | 34 | O | OUT1# | This user defined output pin reflects the complement of MCR[2]. It is inactive (high) after a hardware reset or during loopback mode. |
| 35 | 31 | O | OUT2# | This user defined output pin reflects the complement of MCR[3]. It is inactive (high) after a hardware reset or during loopback mode |
| 1 | 37 | ID | FIFOSEL | FIFO select. This pin has an internal pull-down. For backward compatibility with 16C550, 16C650 and 16C750 devices the FIFO depth is 16 when FIFOSEL is low or left open. The FIFO size is 128 when FIFOSEL is high. The unlatched state of this pin is readable by software. The FIFO size may be set to 128 by writing a 1 in FCR[5] when LCR[7] is set or by putting the device into Enhanced mode, thus overriding the state of the FIFOSEL pin. This pin is unconnected in 16C550 and 16C750 devices. |
| - | 48 | ID | VSEL | Voltage selector. This pin is used to control the voltage thresholds on all input pins. When low (or unconnected), 5V biased TTL thresholds are used. When high, 3V biased TTL thresholds are used. Generally should be tied high when the OX16C950 is being powered off 3 Volts, and low (or unconnected) when powered off 5 Volts. If tied high under 5V operation, CMOS compatible input thresholds are obtained.<br><br>As this pin is not accessible in the PLCC, the PLCC is unsuitable for 3V applications. |
| 12 | 1, 13, 25, 6 | | NC | These pins are not connected. |
| **Power and Ground** | | | | |
| 22 | 18 | | GND | Ground (0 Volts). The GND pin should be tied to ground. |
| 44 | 42 | | VDD | Power supply. The VDD pin should be tied to 5 Volts or 3.3 Volts |

**Table 2: Pin Descriptions**

**Note 1: Direction key:**
I Input
IU Input with pull-up
ID Input with pull-down
O Output
I/O Bi-directional
Note: Attention should be given to high frequency decoupling of power and ground pins due to the high frequency internal switching that occurs under normal operation

## 4.1 Further Pin Information

| Pin | Description | Action when used | Action when not used |
|---|---|---|---|
| | | *Bus Interface Pins* | |
| CS0 | Chip Select | Connect to active high chip select generation logic | Tie high – All chip selects must be active in order to access the device |
| CS1 | Chip Select | Connect to active high chip select generation logic | Tie high – All chip selects must be active in order to access the device |
| CS2# | Chip Select | Connect to active low chip select generation logic | Tie low – All chip selects must be active in order to access the device |
| IOR | Additional I/O Read Control | Connect to processors active high I/O read line (and tie IOR# high) | Tie low (IOR# will be used to control I/O read operations) |
| IOW | Additional I/O Write Control | Connect to processors active high I/O write line (and tie IOW# high) | Tie low (IOW# will be used to control I/O read operations) |
| | | *Control Pins* | |
| INTSEL# | Interrupt Control Mode | Tie low to allow software enable/disable of the interrupt pin. | Leave unconnected (Pulled high internally to leave the interrupt pin permanently enabled). |
| | | *DMA Pins* | |
| RXRDY# | DMA Control signal output | Connect direct to DMA control circuitry | Leave unconnected |
| TXRDY# | DMA Control signal output | Connect direct to DMA control circuitry | Leave unconnected |
| | | *Clock Related Pins* | |
| BDOUT# | Baud rate generator output | Connect direct to the RCLK pin in order to run the receiver with the same clock as the transmitter | Leave unconnected |
| RCLK | Receiver clock input | Connect directly to a suitable receiver clock source (Usually the BDOUT# pin) | n/a |
| XTLI | Crystal circuit input | Connect to suitable clock input | n/a |
| XTLO | Crystal circuit output | Connect to crystal oscillator circuit | Leave unconnected |
| | | *Miscellaneous Pins* | |
| DDIS | Driver Disable output | Connect to active high bus transceiver drive disable (goes high when device is not being read from) | Leave unconnected |
| ADS# | Address Strobe In | Connect direct to external control circuitry (Low-High transition on this pin latches CS0-2 and A0-2) | Tie low |
| OUT1# | User defined output | Connect direct to TTL input of external circuit to control | Leave unconnected |
| OUT2# | User defined output | Connect direct to TTL input of external circuit to control | Leave unconnected |
| | | *Common Channel Pins* | |
| SOUT | Serial data output | Connect to a suitable line driver | Leave unconnected<br>(Serial data can not be transmitted) |
| SIN | Serial data input | Connect to a suitable line receiver | Leave unconnected<br>(Serial data can not be received) |
| RTS# | Request-To-Send Modem signal output | Connect to a suitable line driver | Leave unconnected |
| CTS# | Clear-To-Send Modem signal input | Connect to a suitable line receiver | Tie high |
| DTR# | Data-Terminal-Ready Modem signal output | Connect to a suitable line driver | Leave unconnected |
| DSR# | Data-Set-Ready Modem signal input | Connect to a suitable line receiver | Tie high |

| Pin | Description | Action when used | Action when not used |
|---|---|---|---|
| DCD# | Data-Carrier-Detect<br>Modem signal input | Connect to a suitable line receiver | Tie high |
| RI# | Ring-Indicator<br>Modem signal input | Connect to a suitable line receiver | Tie high |
| INT | Interrupt Output | Connect to an available processor interrupt line | Leave unconnected<br>(Interrupts can not be used) |

# 5 MODE SELECTION

The OX16C950 device is a single channel device software compatible with the 16C450, 16C550, 16C654 and 16C750 UARTs. The operation of the OX16C950 depends on a number of mode settings. These modes are referred to throughout this data sheet. The FIFO depth and compatibility modes are tabulated below:

| UART Mode | FIFO size | FCR[0] | Enhanced mode (EFR[4]=1) | FCR[5] (guarded with LCR[7] = 1) | FIFOSEL pin |
|---|---|---|---|---|---|
| 450 | 1 | 0 | X | X | X |
| 550 | 16 | 1 | 0 | 0 | 0 |
| Extended 550 | 128 | 1 | 0 | X | 1 |
| 650 | 128 | 1 | 1 | X | X |
| 750 | 128 | 1 | 0 | 1 | 0 |
| 950* | 128 | 1 | 1 | X | X |

**Table 3: UART Mode Configuration**
* Note that 950 mode configuration is identical to 650 configuration

## 5.1 450 Mode

After a hardware reset bit 0 of the FIFO Control Register ('FCR') is cleared, hence OX16C950 is compatible with the 16C450. The transmitter and receiver FIFOs (referred to as the 'Transmit Holding Register' and 'Receiver Holding Register' respectively) have a depth of one. This is referred to as 'Byte mode'. When FCR[0] is cleared, all other mode selection parameters are ignored.

## 5.2 550 Mode

Connect FIFOSEL to GND or leave it unconnected. After a hardware reset, writing a 1 to FCR[0] will increase the FIFO size to 16, providing compatibility with 16C550 devices. Since this pin is a no-connect in 16C550 devices, replacing a 16C550 with OX16C950 would result in a 550 compatible device with 16 byte deep FIFOs.

## 5.3 Extended 550 Mode

Connect FIFOSEL to VDD. Writing a 1 to FCR[0] will now increase the FIFO size to 128, thus providing a 550 device with 128 deep FIFOs.

## 5.4 750 Mode

For compatibility with 16C750, leave FIFOSEL unconnected.

Writing a 1 to FCR[0] will increase the FIFO size to 16. In a similar fashion to 16C750, the FIFO size can be further increased to 128 by writing a 1 to FCR[5]. Note that access to FCR[5] is protected by LCR[7]. I.e., to set FCR[5], software should first set LCR[7] to temporarily remove the guard. Once FCR[5] is set, the software should clear LCR[7] for normal operation.

The 16C750 additional features over the 16C550 are available as long as the UART is not put into Enhanced mode (i.e. EFR[4] should be '0'). These features are:

1. Deeper FIFOs
2. Automatic RTS/CTS out-of-band flow control
3. Sleep mode

## 5.5 650 Mode

The OX16C950 is compatible with the 16C650 when EFR[4] is set, i.e. the device is in Enhanced mode. As 650 software drivers usually put the device into Enhanced mode, running 650 drivers on the OX16C950 device will result in 650 compatibility with 128 deep FIFOs, as long as FCR[0] is set. This is regardless of the state of the FIFOSEL pin or package option. Note that the 650 emulation mode of the OX16C950 provides 128 byte deep FIFOs whereas the standard 16C650 has only 32 byte FIFOs.

650 mode has the same enhancements as the 16C750 over the 16C550, but these are enabled using different registers.

There are also additional enhancements over those of the 16C750 in this mode, these are:

1. Automatic in-band flow control
2. Special character detection
3. Infra-red "IrDA-format" transmit and receive mode
4. Transmit trigger levels
5. Optional clock prescaler

## 5.6 950 Mode

The additional features offered in OX16C950 (950 mode) generally only apply when the UART is in Enhanced mode (EFR[4]='1'). Provided FCR[0] is set, in Enhanced mode the FIFO size is 128 regardless of the state of FIFOSEL.

Note that 950 mode configuration is identical to that of 650 mode, however additional 950 specific features are enabled using the Additional Control Register 'ACR' (see section 15.3). In addition to larger FIFOs and higher baud rates, the enhancements of the 16C950 over the 16C654 are:

- Selectable arbitrary trigger levels for the receiver and transmitter FIFO interrupts
- Improved automatic flow control using selectable arbitrary thresholds
- DSR#/DTR# automatic flow control
- Transmitter and receiver can be optionally disabled
- Software reset of device
- Readable FIFO fill levels
- Optional generation of an RS-485 buffer enable signal
- Four-byte device identification (0x16C95003)
- Readable status for automatic in-band and out-of-band flow control
- External 1x clock modes (see section 14.4)
- Flexible "M N/8" clock prescaler (see section 14.2)
- Programmable sample clock to allow data rates up to 15 Mbps (see section 14.3)
- 9-bit data mode

The 950 trigger levels are enabled when ACR[5] is set (bits 4 to 7 of FCR are ignored). Then arbitrary trigger levels can be defined in RTL, TTL, FCL and FCH registers (see section 15). The Additional Status Register ('ASR') offers flow control status for the local and remote transmitters. FIFO levels are readable using RFL and TFL registers.

The UART has a flexible prescaler capable of dividing the system clock by any value between 1 and 31.875 in steps of 0.125. It divides the system clock by an arbitrary value in "M N/8" format, where M and N are 5 and 3-bit binary numbers programmed in CPR[7:3] and CPR[2:0] respectively. This arrangement offers a great deal of flexibility when choosing an input clock frequency to synthesize arbitrary baud rates. The default division value is 4 to provide backward compatibility with 16C650 devices.

The user may apply an external 1x (or Nx) clock for the transmitter and receiver to the RI# and DSR# pin respectively. The transmitter clock may be asserted on the DTR# pin. The external clock options are selected through the CKS register (offset 0x02 of ICR).

It is also possible to define the over-sampling rate used by the transmitter and receiver clocks. The 16C450/16C550 and compatible devices employ 16 times over-sampling, i.e. There are 16 clock cycles per bit. However, OX16C950 can employ any over-sampling rate from 4 to 16 by programming the TCR register. This allows the data rates to be increased to 460.8 Kbps using a 1.8432MHz clock, or 15 Mbps using a 60 MHz clock. The default value after a reset for this register is 0x00, which corresponds to a 16 cycle sampling clock. Writing 0x01, 0x02 or 0x03 will also result in a 16 cycle sampling clock. To program the value to any value from 4 to 15 it is necessary to write this value into TCR i.e. to set the device to a 13 cycle sampling clock it would be necessary to write 0x0D to TCR. For further information see sections 14.3.

The OX16C950 also offers 9-bit data frames for multi-drop industrial applications.

## 6 REGISTER DESCRIPTION TABLES

The three address lines select the various registers in the UART. Since there are more than 8 registers, selection of the registers is also dependent on the state of the Line Control Register 'LCR' and Additional Control Register 'ACR':

1. LCR[7]=1 enables the divider latch registers DLL and DLM.
2. LCR specifies the data format used for both transmitter and receiver. Writing 0xBF (an unused format) to LCR enables access to the 650 compatible register set. Writing this value will set LCR[7] but leaves LCR[6:0] unchanged. Therefore, the data format of the transmitter and receiver data is not affected. Write the desired LCR value to exit from this selection.
3. ACR[7]=1 enables access to the 950 specific registers.
4. ACR[6]=1 enables access to the Indexed Control Register set (ICR) registers as described on page 17.

| Register Name | Address | R/W | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| THR [1] | 000 | W | Data to be transmitted | | | | | | | |
| RHR [1] | 000 | R | Data received | | | | | | | |
| IER [1,2] 650/950 Mode | 001 | R/W | CTS interrupt mask | RTS interrupt mask | Special Char. Detect | Sleep mode | Modem interrupt mask | Rx Stat interrupt mask | THRE interrupt mask | RxRDY interrupt mask |
| 550/750 Mode | | | Unused | | Alternate sleep mode | | | | | |
| FCR [3] 650 mode | 010 | W | RHR Trigger Level | | THR Trigger Level | | DMA Mode / Tx Trigger Enable | Flush THR | Flush RHR | Enable FIFO |
| 750 mode | | | RHR Trigger Level | | FIFO Size | Unused | | | | |
| 950 mode | | | Unused | | | | | | | |
| ISR [3] | 010 | R | FIFOs enabled | | Interrupt priority (Enhanced mode) | | Interrupt priority (All modes) | | | Interrupt pending |
| LCR [4] | 011 | R/W | Divisor latch access | Tx break | Force parity | Odd / even parity | Parity enable | Number of stop bits | Data length | |
| MCR [3,4] 550/750 Mode | 100 | R/W | Unused | | CTS & RTS Flow Control | Internal Loop Back Enable | OUT2 (Int En) | OUT1 | RTS | DTR |
| 650/950 Mode | | | Baud prescale | IrDA mode | XON-Any | | | | | |
| LSR [3,5] Normal | 101 | R | Data Error | Tx Empty | THR Empty | Rx Break | Framing Error | Parity Error | Overrun Error | RxRDY |
| 9-bit data mode | | | | | | | | 9th Rx data bit | | |
| MSR [3] | 110 | R | DCD | RI | DSR | CTS | Delta DCD | Trailing RI edge | Delta DSR | Delta CTS |
| SPR [3] Normal | 111 | R/W | Temporary data storage register and Indexed control register offset value bits | | | | | | | |
| 9-bit data mode | | | Unused | | | | | | | 9th Tx data bit |
| Additional Standard Registers – These registers require divisor latch access bit (LCR[7]) to be set to 1. | | | | | | | | | | |
| DLL | 000 | R/W | Divisor latch bits [7:0] (Least significant byte) | | | | | | | |
| DLM | 001 | R/W | Divisor latch bits [15:8] (Most significant byte) | | | | | | | |

**Table 4: Standard 550 Compatible Registers**

| Register Name | Address | R/W | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| To access these registers LCR must be set to 0xBF | | | | | | | | | | |
| EFR | 010 | R/W | CTS flow control | RTS Flow control | Special char detect | Enhanced mode | In-band flow control mode | | | |
| XON1 | 100 | R/W | XON Character 1 | | | | | | | |
| 9-bit mode | | | Special character 1 | | | | | | | |
| XON2 | 101 | R/W | XON Character 2 | | | | | | | |
| 9-bit mode | | | Special Character 2 | | | | | | | |
| XOFF1 | 110 | R/W | XOFF Character 1 | | | | | | | |
| 9-bit mode | | | Special character 3 | | | | | | | |
| XOFF2 | 111 | R/W | XOFF Character 2 | | | | | | | |
| 9-bit mode | | | Special character 4 | | | | | | | |

**Table 5: 650 Compatible Registers**

| Register Name | Address | R/W | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| ASR [1,6,7] | 001 | R/W [7] | Tx Idle | FIFO size | FIFO-SEL | Special Char Detect | DTR | RTS | Remote Tx Disabled | Tx Disabled |
| RFL [6] | 011 | R | Number of characters in the receiver FIFO | | | | | | | |
| TFL [3,6] | 100 | R | Number of characters in the transmitter FIFO | | | | | | | |
| ICR [3,8,9] | 101 | R/W | Data read/written depends on the value written to the SPR prior to the access of this register (see Table 7) | | | | | | | |

**Table 6: 950 Specific Registers**

*Register access notes:*

Note 1: Requires LCR[7] = 0
Note 2: Requires ACR[7] = 0
Note 3: Requires that last value written to LCR was not 0xBF
Note 4: To read this register ACR[7] must be = 0
Note 5: To read this register ACR[6] must be = 0
Note 6: Requires ACR[7] = 1
Note 7: Only bits 0 and 1 of this register can be written
Note 8: To read this register ACR[6] must be = 1
Note 9: This register acts as a window through which to read and write registers in the Indexed Control Register set

| Register Name | SPR Offset [10] | R/W | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Indexed Control Register Set | | | | | | | | | | |
| ACR | 0x00 | R/W | Addit-ional Status Enable | ICR Read Enable | 950 Trigger Level Enable | DTR definition and control | | Auto DSR Flow Control Enable | Tx Disable | Rx Disable |
| CPR | 0x01 | R/W | 5 Bit "integer" part of clock prescaler | | | | | 3 Bit "fractional" part of clock prescaler | | |
| TCR | 0x02 | R/W | Unused | | | | 4 Bit N-times clock selection bits [3:0] | | | |
| CKS | 0x03 | R/W | Tx 1x Mode | Tx CLK Select | BDOUT on DTR | DTR 1x Tx CLK | Rx 1x Mode | Disable BDOUT | Receiver Clock Sel[1:0] | |
| TTL | 0x04 | R/W | Unused | Transmitter Interrupt Trigger Level (0-127) | | | | | | |
| RTL | 0x05 | R/W | Unused | Receiver Interrupt Trigger Level (1-127) | | | | | | |
| FCL | 0x06 | R/W | Unused | Automatic Flow Control Lower Trigger Level (0-127) | | | | | | |
| FCH | 0x07 | R/W | Unused | Automatic Flow Control Higher Trigger level (1-127) | | | | | | |
| ID1 | 0x08 | R | Hardwired ID byte 1 (0x16) | | | | | | | |
| ID2 | 0x09 | R | Hardwired ID byte 1 (0xC9) | | | | | | | |
| ID3 | 0x0A | R | Hardwired ID byte 1 (0x50) | | | | | | | |
| REV | 0x0B | R | Hardwired revision byte (0x03) | | | | | | | |
| CSR | 0x0C | W | Writing 0x00 to this register will reset the UART (Except the CKS and CKA registers) | | | | | | | |
| NMR | 0x0D | R/W | Unused | | 9th Bit SChar 4 | 9th Bit Schar 3 | 9th Bit SChar 2 | 9th Bit SChar 1 | 9th-bit Int. En. | 9 Bit Enable |
| MDM | 0x0E | R/W | Unused | | | | Δ DCD Wakeup disable | Trailing RI edge disable | Δ DSR Wakeup disable | Δ CTS Wakeup disable |
| RFC | 0X0F | R | FCR[7] | FCR[6] | FCR[5] | FCR[4] | FCR[3] | FCR[2] | FCR[1] | FCR[0] |
| GDS | 0X10 | R | Unused | | | | | | | Good Data Status |
| DMS | 0x11 | R/W | Force TxRdy inactive | Force RxRdy inactive | Unused | | | | TxRdy status ( R ) | RxRdy status ( R ) |
| PIDX | 0x12 | R | Hardwired Port Index ( 0x00 ) | | | | | | | |
| CKA | 0x13 | R/W | Unused | | | Output sys-clk on txrdy | Use CLKSEL pin for sys-clk | Invert DTR signal | Invert internal tx clock | Invert internal rx clock |

**Table 7: Indexed Control Register Set**

Note 10: The SPR offset column indicates the value that must be written into SPR prior to reading / writing any of the indexed control registers via ICR. Offset values not listed in the table are reserved for future use and must not be used.

To read or write to any of the Indexed Control Registers use the following procedure.

Writing to ICR registers:
Ensure that the last value written to LCR was not 0xBF (reserved for 650 compatible register access value).
Write the desired offset to SPR (address $111_2$).

Write the desired value to ICR (address $101_2$).

Reading from ICR registers:
Ensure that the last value written to LCR was not 0xBF (see above).
Write 0x00 offset to SPR to select ACR.
Set bit 6 of ACR (ICR read enable) by writing $x1xxxxx_2$ to address $101_2$. Ensure that other bits in ACR are not changed.
(Software drivers should keep a copy of the contents of the ACR elsewhere since reading ICR involves overwriting ACR!)
Write the desired offset to SPR (address $111_2$).
Read the desired value from ICR (address $101_2$).
Write 0x00 offset to SPR to select ACR.
Clear bit 6 of ACR bye writing $x0xxxxx_2$ to ICR, thus enabling access to standard registers again.

# 7 RESET CONFIGURATION

## 7.1 Hardware Reset

After a hardware reset, all writable registers are reset to 0x00, with the following exceptions:

1. DLL which is reset to 0x01.
2. MCR[7] is reset to the complement of the CLKSEL input pin value (see section 11.1).
3. CPR is reset to 0x20.

The state of read-only registers following a hardware reset is as follows:

RHR[7:0]: Indeterminate
RFL[6:0]: $0000000_2$
TFL[6:0]: $0000000_2$
LSR[7:0]: 0x60 signifying that both the transmitter and the transmitter FIFO are empty
MSR[3:0]: $0000_2$
MSR[7:4]: Dependent on modem input lines DCD, RI, DSR and CTS respectively
ISR[7:0]: 0x01, i.e. no interrupts are pending
ASR[7:0]: $1xx00000_2$
RFC[7:0]: $00000000_2$
GDS[7:0]: $00000001_2$
DMS[7:0]: $00000010_2$
CKA[7:0]: $00000000_2$

The reset state of output signals for are tabulated below:

| Signal | Reset state |
|---|---|
| SOUT | Inactive High |
| RTS# | Inactive High |
| DTR# | Inactive High |
| INT | Inactive low when INTSEL# pin is high or floating, otherwise high-impedance |
| RXRDY# | Inactive High |
| TXRDY# | Active low (THR is able to receive data). |

**Table 8: Output Signal Reset State**

## 7.2 Software Reset

An additional feature available in the OX16C950 device is software resetting of the serial channel. The software reset is available using the CSR register. Software reset has the same effect as a hardware reset except it does not reset the clock source selections (i.e. CKS register and CKA register). To reset the UART write 0x00 to the Channel Software Reset register 'CSR'.

# 8 TRANSMITTER & RECEIVER FIFOS

Both the transmitter and receiver have associated holding registers (FIFOs), referred to as the transmitter holding register (THR) and receiver holding register (RHR) respectively.

In normal operation, when the transmitter finishes transmitting a byte it will remove the next data from the top of the THR and proceed to transmit it. If the THR is empty, it will wait until data is written into it. If THR is empty and the last character being transmitted has been completed (i.e. the transmitter shift register is empty) the transmitter is said to be idle. Similarly, when the receiver finishes receiving a byte, it will transfer it to the bottom of the RHR. If the RHR is full, an overrun condition will occur (see section 9.3).

Data is written into the bottom of the THR queue and read from the top of the RHR queue completely asynchronously to the operation of the transmitter and receiver.

The size of the FIFOs is dependent on the setting of the FCR register. When in Byte mode, these FIFOs only accept one byte at a time before indicating that they are full; this is compatible with the 16C450. When in a FIFO mode, the size of the FIFOs is either 16 (compatible with the 16C550) or 128.

Data written to the THR when it is full is lost. Data read from the RHR when it is empty is invalid. The empty or full status of the FIFOs are indicated in the Line Status Register 'LSR' (see section 9.3). Interrupts can be generated or DMA signals can be used to transfer data to/from the FIFOs. The number of items in each FIFO may also be read back from the transmitter FIFO level (TFL) and receiver FIFO level (RFL) registers (see section 15.2).

## 8.1 FIFO Control Register 'FCR'

**FCR[0]: Enable FIFO mode**
logic 0 ⇒ Byte mode.
logic 1 ⇒ FIFO mode.

This bit should be enabled before setting the FIFO trigger levels.

**FCR[1]: Flush RHR**
logic 0 ⇒ No change.
logic 1 ⇒ Flushes the contents of the RHR

This is only operative when already in a FIFO mode. The RHR is automatically flushed whenever changing between Byte mode and a FIFO mode. This bit will return to zero after clearing the FIFOs.

**FCR[2]: Flush THR**
logic 0 ⇒ No change.
logic 1 ⇒ Flushes the contents of the THR, in the same manner as FCR[1] does for the RHR.

*DMA Transfer Signalling:*

**FCR[3]: DMA signalling mode / Tx trigger level enable**
logic 0 ⇒ DMA mode '0'.
logic 1 ⇒ DMA mode '1'.

**Note: In DMA mode 0, the transmitter trigger level is ALWAYS set to 1, thus ignoring FCR[5:4] and TTL.**

DMA Control signals can be generated using the TXRDY# and RXRDY# pins. Their operation is defined as follows:

The TXRDY# pin has no hysteresis and is simply activated using a comparison operation. When the UART is in DMA mode 0 (or in Byte mode), the TXRDY# output pin is active (low) whenever THR is empty, otherwise it is inactive. When in DMA mode 1, the TXRDY# pin is inactive (high) when the THR is full, otherwise it is active, signifying that there is room in the transmit FIFO.

The RXRDY# pin can operate with hysteresis. In DMA mode 0 (or in Byte mode), RXRDY# is only active (low) when RHR contains data. When in DMA mode 1 however, the operation is as follows:

1. RXRDY# is set active when RFL has reached the receiver interrupt trigger level or a time-out event has occurred (see section 10.3) It remains active as long as RHR is not empty.
2. RXRDY# is set inactive when RHR is empty. It remains in this state until condition 1 occurs.

**FCR[5:4]: THR trigger level**
Generally in 450, 550, extended 550 and 950 modes these bits are unused (see section 5 for mode definition). In 650 mode they define the transmitter interrupt trigger levels and in 750 mode FCR[5] increases the FIFO size.

*450, 550 and extended 550 modes:*
The transmitter interrupt trigger levels are set to 1 and FCR[5:4] are ignored.

*650 mode:*
In 650 mode the transmitter interrupt trigger levels are set to the following values:

| FCR[5:4] | Transmit Interrupt Trigger level |
|---|---|
| 00 | 16 |
| 01 | 32 |
| 10 | 64 |
| 11 | 112 |

**Table 9: Transmit Interrupt Trigger Levels**

These levels only apply when in Enhanced mode and in DMA mode 1 (FCR[3] = 1), otherwise the trigger level is set to 1. A transmitter empty interrupt will be generated (if enabled) if the TFL falls below the trigger level.

*750 Mode:*
In 750 compatible non-Enhanced (EFR[4]=0) mode, transmitter trigger level is set to 1, FCR[4] is unused and FCR[5] defines the FIFO depth as follows:

FCR[5]=0 Transmitter and receiver FIFO size is 16 bytes.
FCR[5]=1 Transmitter and receiver FIFO size is 128 bytes.

In non-Enhanced mode and when FIFOSEL pin is low, FCR[5] is only writable when LCR[7] is set. Note that in Enhanced mode, the FIFO size is also increased to 128 bytes when FCR[0] is set.

*950 mode:*
Setting ACR[5]=1 enables arbitrary transmitter trigger level setting using the TTL register (see section 15.4), hence FCR[5:4] are ignored.

**FCR[7:6]: RHR trigger level**
In 550, extended 550, 650 and 750 modes, the receiver FIFO trigger levels are defined using FCR[7:6]. The interrupt trigger level and upper flow control trigger level where appropriate are defined by L2 in the table below. L1 defines the lower flow control trigger level where applicable. Separate upper and lower flow control trigger levels introduce a hysteresis element in in-band and out-of-band flow control (see section 13).

| FCR [7:6] | Mode | | | | | |
|---|---|---|---|---|---|---|
| | 650 *FIFO Size 128* | | Ext. 550 / 750 *FIFO Size 128* | | 550 *FIFO Size 16* | |
| | L1 | L2 | L1 | L2 | L1 | L2 |
| 00 | 1 | 16 | 1 | 1 | n/a | 1 |
| 01 | 16 | 32 | 1 | 32 | n/a | 4 |
| 10 | 32 | 112 | 1 | 64 | n/a | 8 |
| 11 | 112 | 120 | 1 | 112 | n/a | 14 |

**Table 10: Receiver Trigger Levels**

In Byte mode (450 mode) the trigger levels are all set to 1.

In all cases, a receiver data interrupt will be generated (if enabled) if the Receiver FIFO Level ('RFL') reaches the upper trigger level L2.

950 Mode:
When 950 trigger levels are enabled (ACR[5]=1), more flexible trigger levels can be set by writing to the TTL, RTL, FCL and FCH (see section 15) hence ignoring FCR[7:6].

# 9 LINE CONTROL & STATUS

## 9.1 False Start Bit Detection

On the falling edge of a start bit, the receiver will wait for 1/2 bit and re-synchronise the receiver's sampling clock onto the centre of the start bit. The start bit is valid if the SIN line is still low at this mid-bit sample and the receiver will proceed to read in a data character. Verifying the start bit prevents the receiver from assembling a false data character due to a low going noise spike on the SIN input.

Once the first stop bit has been sampled, the received data is transferred to the RHR and the receiver will then wait for a low transition on SIN signifying the next start bit.

The receiver will continue receiving data even if the RHR is full or the receiver has been disabled (see section 15.3) in order to maintain framing synchronisation. The only difference is that the received data does not get transferred to the RHR.

## 9.2 Line Control Register 'LCR'

The LCR specifies the data format that is common to both transmitter and receiver. Writing 0xBF to LCR enables access to the EFR, XON1, XOFF1, XON2 and XOFF2, DLL and DLM registers. This value (0xBF) corresponds to an unused data format. Writing the value 0xBF to LCR will set LCR[7] but leaves LCR[6:0] unchanged. Therefore, the data format of the transmitter and receiver data is not affected. Write the desired LCR value to exit from this selection.

**LCR[1:0]: Data length**
LCR[1:0] Determines the data length of serial characters. Note however, that these values are ignored in 9-bit data framing mode, i.e. when NMR[0] is set.

| LCR[1:0] | Data length |
|---|---|
| 00 | 5 bits |
| 01 | 6 bits |
| 10 | 7 bits |
| 11 | 8 bits |

**Table 11: LCR Data Length Configuration**

**LCR[2]: Number of stop bits**
LCR[2] defines the number of stop bits per serial character.

| LCR[2] | Data length | No. stop bits |
|---|---|---|
| 0 | 5,6,7,8 | 1 |
| 1 | 5 | 1.5 |
| 1 | 6,7,8 | 2 |

**Table 12: LCR Stop Bit Number Configuration**

**LCR[5:3]: Parity type**
The selected parity type will be generated during transmission and checked by the receiver, which may produce a parity error as a result. In 9-bit mode parity is disabled and LCR[5:3] is ignored.

| LCR[5:3] | Parity type |
|---|---|
| xx0 | No parity bit |
| 001 | Odd parity bit |
| 011 | Even parity bit |
| 101 | Parity bit forced to 1 |
| 111 | Parity bit forced to 0 |

**Table 13: LCR Parity Configuration**

**LCR[6]: Transmission break**
logic 0 ⇒ Break transmission disabled.
logic 1 ⇒ Forces the transmitter data output SOUT low to alert the communication terminal, or send zeros in IrDA mode.

It is the responsibility of the software driver to ensure that the break duration is longer than the character period for it to be recognised remotely as a break rather than data.

**LCR[7]: Divisor latch enable**
logic 0 ⇒ Access to DLL and DLM registers disabled.
logic 1 ⇒ Access to DLL and DLM registers enabled.

## *9.3 Line Status Register 'LSR'*

This register provides the status of data transfer to CPU.

**LSR[0]: RHR data available**
logic 0 ⇒ RHR is empty: no data available
logic 1 ⇒ RHR is not empty: data is available to be read.

**LSR[1]: RHR overrun error**
logic 0 ⇒ No overrun error.
logic 1 ⇒ Data was received when the RHR was full. An overrun error has occurred. The error is flagged when the data would normally have been transferred to the RHR.

**LSR[2]: Received data parity error**
logic 0 ⇒ No parity error in normal mode or 9th bit of received data is '0' in 9-bit mode.
logic 1 ⇒ Data has been received that did not have correct parity in normal mode or 9th bit of received data is '1' in 9-bit mode.

The flag will be set when the data item in error is at the top of the RHR and cleared following a read of the LSR. In 9-bit mode LSR[2] is no longer a flag and corresponds to the 9th bit of the received data in RHR.

**LSR[3]: Received data framing error**
logic 0 ⇒ No framing error.
logic 1 ⇒ Data has been received with an invalid stop bit.

This status bit is set and cleared in the same manner as LSR[2]. When a framing error occurs, the UART will try to re-synchronise by assuming that the error was due to sampling the start bit of the next data item.

**LSR[4]: Received break error**
logic 0 ⇒ No receiver break error.
logic 1 ⇒ The receiver received a break.

A break condition occurs when the SIN line goes low (normally signifying a start bit) and stays low throughout the start, data, parity and first stop bit. (Note that the SIN line is sampled at the bit rate). One zero character with associated break flag set will be transferred to the RHR and the receiver will then wait until the SIN line returns high. The LSR[4] break flag will be set when this data item gets to the top of the RHR and it is cleared following a read of the LSR.

**LSR[5]: THR empty**
logic 0 ⇒ Transmitter FIFO (THR) is not empty.
logic 1 ⇒ Transmitter FIFO (THR) is empty.

**LSR[6]: Transmitter and THR empty**
logic 0 ⇒ The transmitter is not idle
logic 1 ⇒ THR is empty and the transmitter has completed the character in shift register and is in idle mode. (I.e. set whenever the transmitter shift register and the THR are both empty.)

**LSR[7]: Receiver data error**
logic 0 ⇒ Either there are no receiver data errors in the FIFO or it was cleared by an earlier read of LSR.
logic 1 ⇒ At least one parity error, framing error or break indication in the FIFO.

In 450 mode LSR[7] is permanently cleared, otherwise this bit will be set when an erroneous character is transferred from the receiver to the RHR. It is cleared when the LSR is read. **Note that in 16C550 this bit is only cleared when all of the erroneous data are removed from the FIFO**. In 9-bit data framing mode parity is permanently disabled, so this bit is not affected by LSR[2].

# 10 INTERRUPTS & SLEEP MODE

The serial channel interrupts are asserted on the INT pin. When INTSEL# is high or unconnected, the INT pin is forcing logic and MCR[3] is ignored. When INTSEL# is low, the tri-state control of INT is controlled by MCR[3]. In this case the INT pin is forcing when MCR[3] is set. It is in high-impedance state when MCR[3] is cleared.

## 10.1 Interrupt Enable Register 'IER'

Serial channel interrupts are enabled using the Interrupt Enable Register ('IER').

**IER[0]: Receiver data available interrupt mask**
logic 0 ⇒Disable the receiver ready interrupt.
logic 1 ⇒Enable the receiver ready interrupt.

**IER[1]: Transmitter empty interrupt mask**
logic 0 ⇒Disable the transmitter empty interrupt.
logic 1 ⇒Enable the transmitter empty interrupt.

**IER[2]: Receiver status interrupt**
*Normal mode:*
logic 0 ⇒ Disable the receiver status interrupt.
logic 1 ⇒ Enable the receiver status interrupt.

*9-bit data mode:*
logic 0 ⇒ Disable receiver status and address bit interrupt.
logic 1 ⇒ Enable receiver status and address bit interrupt.

In 9-bit mode (i.e. when NMR[0] is set) reception of a character with the address-bit (9th bit) set can generate a level 1 interrupt if IER[2] is set.

**IER[3]: Modem status interrupt mask**
logic 0 ⇒ Disable the modem status interrupt.
logic 1 ⇒ Enable the modem status interrupt.

**IER[4]: Sleep mode**
logic 0 ⇒ Disable sleep mode.
logic 1 ⇒ Enable sleep mode whereby the internal clock of the channel is switched off.

Sleep mode is described in section 10.4.

**IER[5]: Special character interrupt mask or alternate sleep mode**
*9-bit data framing mode:*
logic 0 ⇒ Disable the special character receive interrupt.
logic 1 ⇒ Enable the special character receive interrupt.

In 9-bit data mode, The receiver can detect up to four special characters programmed in Special Character 1 to 4. When IER[5] is set, a level 5 interrupt is asserted when a match is detected.

*650/950 modes (non-9-bit data framing):*
logic 0 ⇒ Disable the special character receive interrupt.
logic 1 ⇒ Enable the special character receive interrupt.

In 16C650 compatible mode when the device is in Enhanced mode (EFR[4]=1), this bit enables the detection of special characters. It enables both the detection of XOFF characters (when in-band flow control is enabled via EFR[3:0]) and the detection of the XOFF2 special character (when enabled via EFR[5]).

*750 mode (non-9-bit data framing):*
logic 0 ⇒ Disable alternate sleep mode.
logic 1 ⇒ Enable alternate sleep mode whereby the internal clock of the channel is switched off.

In 16C750 compatible mode (i.e. non-Enhanced mode), this bit is used an alternate sleep mode and has the same effect as IER[4]. (See section 10.4)

**IER[6]: RTS interrupt mask**
logic 0 ⇒ Disable the RTS interrupt.
logic 1 ⇒ Enable the RTS interrupt.

This enable is only operative in Enhanced mode (EFR[4]=1). In non-Enhanced mode, RTS interrupt is permanently enabled.

**IER[7]: CTS interrupt mask**
logic 0 ⇒ Disable the CTS interrupt.
logic 1 ⇒ Enable the CTS interrupt.

This enable is only operative in Enhanced mode (EFR[4]=1). In non-Enhanced mode, CTS interrupt is permanently enabled.

## 10.2 Interrupt Status Register 'ISR'

The source of the highest priority interrupt pending is indicated by the contents of the Interrupt Status Register 'ISR'. There are nine sources of interrupt at six levels of priority (1 is the highest) as tabulated below:

| Level | Interrupt source | ISR[5:0] *see note 3* |
|---|---|---|
| - | No interrupt pending [1] | 000001 |
| 1 | Receiver status error **or** Address-bit detected in 9-bit mode | 000110 |
| 2a | Receiver data available | 000100 |
| 2b | Receiver time-out | 001100 |
| 3 | Transmitter THR empty | 000010 |
| 4 | Modem status change | 000000 |
| 5 [2] | In-band flow control XOFF **or** Special character (XOFF2) **or** Special character 1, 2, 3 or 4 **or** bit 9 set in 9-bit mode | 010000 |
| 6 [2] | CTS or RTS change of state | 100000 |

**Table 14: Interrupt Status Identification Codes**

Note1: ISR[0] indicates whether any interrupts are pending.
Note2: Interrupts of priority levels 5 and 6 cannot occur unless the UART is in Enhanced mode.
Note3: ISR[5] is only used in 650 & 950 modes. In 750 mode, it is '0' when FIFO size is 16 and '1' when FIFO size is 128. In all other modes it is permanently set to '0'.

## 10.3 Interrupt Description

*Level 1:*
**Receiver status error interrupt (ISR[5:0]='000110'):**
*Normal (non-9-bit) mode:*
This interrupt is active whenever any of LSR[1], LSR[2], LSR[3] or LSR[4] are set. These flags are cleared following a read of the LSR. This interrupt is masked with IER[2].

*9-bit mode:*
This interrupt is active whenever any of LSR[1], LSR[2], LSR[3] or LSR[4] are set. The receiver error interrupt due to LSR[1], LSR[3] and LSR[4] is masked with IER[3]. The 'address-bit' received interrupt is masked with NMR[1]. The software driver can differentiate between receiver status error and received address-bit (9th data bit) interrupt by examining LSR[1] and LSR[7]. In 9-bit mode LSR[7] is only set when LSR[3] or LSR[4] is set and it is not affected by LSR[2] (i.e. 9th data bit).

*Level 2a:*
**Receiver data available interrupt (ISR[5:0]='000100'):**
This interrupt is active whenever the receiver FIFO level is above the interrupt trigger level.

*Level 2b:*
**Receiver time-out interrupt (ISR[5:0]='001100'):**
A receiver time-out event, which may cause an interrupt, will occur when all of the following conditions are true:

- The UART is in a FIFO mode
- There is data in the RHR.
- There has been no read of the RHR for a period of time greater than the time-out period.
- There has been no new data received and written into the RHR for a period of time greater than the time-out period. The time-out period is four times the character period (including start and stop bits) measured from the centre of the first stop bit of the last data item received.

Reading the first data item in RHR clears this interrupt.

*Level 3:*
**Transmitter empty interrupt (ISR[5:0]='000010'):**
This interrupt is set when the transmit FIFO level falls below the trigger level. It is cleared on an ISR read of a level 3 interrupt or by writing more data to the THR so that the trigger level is exceeded. Note that when 16C950 mode trigger levels are enabled (ACR[5]=1) and the transmitter trigger level of zero is selected (TTL=0x00), a transmitter empty interrupt will only be asserted when both the transmitter FIFO and transmitter shift register are empty and the SOUT line has returned to idle marking state.

*Level 4:*
**Modem change interrupt (ISR[5:0]='000000'):**
This interrupt is set by a modem change flag (MSR[0], MSR[1], MSR[2] or MSR[3]) becoming active due to changes in the input modem lines. This interrupt is cleared following a read of the MSR.

*Level 5:*
**Receiver in-band flow control (XOFF) detect interrupt, Receiver special character (XOFF2) detect interrupt, Receiver special character 1, 2, 3 or 4 interrupt or 9th Bit set interrupt in 9-bit mode (ISR[5:0]='010000'):**
A level 5 interrupt can only occur in Enhanced-mode when any of the following conditions are met:

- A valid XOFF character is received while in-band flow control is enabled.
- A received character matches XOFF2 while special character detection is enabled.
- A received character matches special character 1, 2, 3 or 4 in 9-bit mode (see section 15.9).

It is cleared on an ISR read of a level 5 interrupt.

*Level 6:*
**CTS or RTS changed interrupt (ISR[5:0]='100000'):**
This interrupt is set whenever either of the CTS# or RTS# pins changes state from low to high. It is cleared on an ISR read of a level 6 interrupt.

### 10.4 Sleep Mode

For a channel to go into sleep mode, all of the following conditions must be met:

- Sleep mode enabled (IER[4]=1 in 650/950 modes, or IER[5]=1 in 750 mode):
- The transmitter is idle, i.e. the transmitter shift register and FIFO are both empty.
- SIN is high.
- The receiver is idle.
- The receiver FIFO is empty (LSR[0]=0).
- The UART is not in loopback mode (MCR[4]=0).
- Changes on modem input lines have been acknowledged (i.e. MSR[3:0]=0000).
- No interrupts are pending.

A read of IER[4] (or IER[5] if a 1 was written to that bit instead) shows whether the power-down request was successful. The UART will fully retain its programmed state whilst in power-down mode.
The channel will automatically exit power-down mode when any of the conditions 1 to 7 becomes false. It may be woken manually by clearing IER[4] (or IER[5] if the alternate sleep mode is enabled).

**Sleep mode operation is not available in IrDA mode.**

## 11 MODEM INTERFACE

### 11.1 Modem Control Register 'MCR'

**MCR[0]: DTR**
logic 0 ⇒ Force DTR# output to inactive (high).
logic 1 ⇒ Force DTR# output to active (low).

Note that DTR# can be used for automatic out-of-band flow control when enabled using ACR[4:3] (see section 15.3).

**MCR[1]: RTS**
logic 0 ⇒ Force RTS# output to inactive (high).
logic 1 ⇒ Force RTS# output to active (low).

Note that RTS# can be used for automatic out-of-band flow control when enabled using EFR[6] (see section 13.4).

**MCR[2]: OUT1**
logic 0 ⇒ Force OUT1# output low when loopback mode is disabled.
logic 1 ⇒ Force OUT1# output high.

**MCR[3]: OUT2/External interrupt enable**
logic 0 ⇒ Force OUT2# output low when loopback mode is disabled. If INTSEL# is low the external interrupt is in high-impedance state when MCR[3] is cleared. If INTSEL# is high MCR[3] does not affect the interrupt.
logic 1 ⇒ Force OUT2# output high. If INTSEL# is low the external interrupt is enabled and operating in normal active (forcing) mode when MCR[3] is high. If INTSEL# is high MCR[3] does not affect the interrupt.

**MCR[4]: Loopback mode**
logic 0 ⇒ Normal operating mode.
logic 1 ⇒ Enable local loop-back mode (diagnostics).

In local loop-back mode, the transmitter output (SOUT) and the four modem outputs (DTR#, RTS#, OUT1# and OUT2#) are set in-active (high), and the receiver inputs SIN, CTS#, DSR#, DCD#, and RI# are all disabled. Internally the transmitter output is connected to the receiver input and DTR#, RTS#, OUT1# and OUT2# are connected to modem status inputs DSR#, CTS#, RI# and DCD# respectively.

In this mode, the receiver and transmitter interrupts are fully operational. The modem control interrupts are also operational, but the interrupt sources are now the lower four bits of the Modem Control Register instead of the four modem status inputs. The interrupts are still controlled by the IER.

**MCR[5]: Enable XON-Any in Enhanced mode or enable out-of-band flow control in non-Enhanced mode**

*650/950 modes (Enhanced mode):*
logic 0 ⇒ XON-Any is disabled.
logic 1 ⇒ XON-Any is enabled.

In enhanced mode (EFR[4]=1), this bit enables the Xon-Any operation. When Xon-Any is enabled, any received data will be accepted as a valid XON (see in-band flow control, section 13.3).

*750 mode (Non-Enhanced mode):*
logic 0 ⇒ CTS/RTS flow control disabled.
logic 1 ⇒ CTS/RTS flow control enabled.

In non-enhanced mode, this bit enables the CTS/RTS out-of-band flow control.

**MCR[6]: IrDA mode**
logic 0 ⇒ Standard serial receiver and transmitter data format.
logic 1 ⇒ Data will be transmitted and received in IrDA format.

This function is only available in Enhanced mode. It requires a 16x clock to function correctly.

**MCR[7]: Baud rate prescaler select**
logic 0 ⇒ Normal (divide by 1) baud rate generator prescaler selected.
logic 1 ⇒ Divide-by-"M N/8" baud rate generator prescaler selected.

Where M & N are programmed in CPR (ICR offset 0x01). After a hardware reset, CPR defaults to 0x20 (divide-by-4) and MCR[7] is loaded with the complement of the CLKSEL pin. User writes to this flag will only take effect in enhanced mode. See section 13.1.

## 11.2 Modem Status Register 'MSR'

**MSR[0]: Delta CTS#**
Indicates that the CTS# input has changed since the last time the MSR was read.

**MSR[1]: Delta DSR#**
Indicates that the DSR# input has changed since the last time the MSR was read.

**MSR[2]: Trailing edge RI#**
Indicates that the RI# input has changed from low to high since the last time the MSR was read.

**MSR[3]: Delta DCD#**
Indicates that the DCD# input has changed since the last time the MSR was read.

**MSR[4]: CTS**
This bit is the complement of the CTS# input. It is equivalent to RTS (MCR[1]) during internal loop-back mode.

**MSR[5]: DSR**
This bit is the complement of the DSR# input. It is equivalent to DTR (MCR[0]) during internal loop-back mode.

**MSR[6]: RI**
This bit is the complement of the RI# input. In internal loop-back mode it is equivalent to the internal OUT1.

**MSR[7]: DCD**
This bit is the complement of the DCD# input. In internal loop-back mode it is equivalent to the internal OUT2.

# 12 OTHER STANDARD REGISTERS

## 12.1 Divisor Latch Registers 'DLL & DLM'

The divisor latch registers are used to program the baud rate divisor. This is a value between 1 and 65535 by which the input clock is divided by in order to generate serial baud rates. After a hardware reset, the baud rate used by the transmitter and receiver is given by:

$$Baudrate = \frac{InputClock}{16 * Divisor}$$

Where divisor is given by DLL + ( 256 x DLM ). More flexible baud rate generation options are also available. See section 14 for full details.

## 12.2 Scratch Pad Register 'SPR'

The scratch pad register does not affect operation of the rest of the UART in any way and can be used for temporary data storage. The register may also be used to define an offset value to access the registers in the Indexed Control Register set. For more information on Indexed Control registers see Table 7 and section 15.

# 13 AUTOMATIC FLOW CONTROL

Automatic in-band flow control, automatic out-of-band flow control and special character detection features can be used when in Enhanced mode and are software compatible with the 16C654. Alternatively, 16C750 compatible automatic out-of-band flow control can be enabled when in non-Enhanced mode. In 950 mode, in-band and out-of-band flow controls are compatible with 16C654, with the addition of fully programmable flow control thresholds.

## 13.1 Enhanced Features Register 'EFR'

Writing 0xBF to LCR enables access to the EFR and other Enhanced mode registers. This value corresponds to an unused data format. Writing 0xBF to LCR will set LCR[7] but leaves LCR[6:0] unchanged. Therefore, the data format of the transmitter and receiver data is not affected. Write the desired LCR value to exit from this selection.

Note: In-band transmit and receive flow control is disabled in 9-bit mode.

**EFR[1:0]: In-band receive flow control mode**
When in-band receive flow control is enabled, the UART compares the received data with the programmed XOFF character. When this occurs, the UART will disable transmission as soon as any current character transmission is complete. The UART then compares the received data with the programmed XON character. When a match occurs, the UART will re-enable transmission (see section 15.6).

For automatic in-band flow control, bit 4 of EFR must be set. The combinations of software receive flow control can be selected by programming EFR[1:0] as follows:

logic [00] ⇒ In-band receive flow control is disabled.
logic [01] ⇒ Single character in-band receive flow control enabled, recognising XON2 as the XON character and XOFF2 as the XOFF character.
logic [10] ⇒ Single character in-band receive flow control enabled, recognising XON1 as the XON character and XOFF1 and the XOFF character.
logic [11] ⇒ The behavior of the receive flow control is dependent on the configuration of EFR[3:2]. single character in-band receive flow control is enabled, accepting both XON1 and XON2 as valid XON characters and both XOFF1 and XOFF2 as valid XOFF characters when EFR[3:2] = "01" or "10". EFR[1:0] should not be set to "11" when EFR[3:2] is either "00".

**EFR[3:2]: In-band transmit flow control mode**
When in-band transmit flow control is enabled, an XON/XOFF character is inserted into the data stream whenever the RFL passes the upper trigger level and falls below the lower trigger level respectively.

For automatic in-band flow control, bit 4 of EFR must be set. The combinations of software transmit flow control can then be selected by programming EFR[3:2] as follows:

logic [00] ⇒ In-band transmit flow control is disabled.
logic [01] ⇒ Single character in-band transmit flow control enabled, using XON2 as the XON character and XOFF2 as the XOFF character.
logic [10] ⇒ Single character in-band transmit flow control enabled, using XON1 as the XON character and XOFF1 as the XOFF character.
Logic[11] ⇒ The value EFR[3:2] = "11" is reserved for future use and should not be used

**EFR[4]: Enhanced mode**
logic 0 ⇒ Non-Enhanced mode. Disables IER bits 4-7, ISR bits 4-5, FCR bits 4-5, MCR bits 5-7 and in-band flow control. Whenever this bit is cleared, the setting of other bits of EFR are ignored.
logic 1 ⇒ Enhanced mode. Enables the Enhanced Mode functions. These functions include enabling IER bits 4-7, FCR bits 4-5, MCR bits 5-7. For in-band flow control the software driver must set this bit first. If this bit is set, out-of-band flow control is configured with EFR bits 6-7, otherwise out-of-band flow control is compatible with 16C750.

**EFR[5]: Enable special character detection**
logic 0 ⇒ Special character detection is disabled.
logic 1 ⇒ While in Enhanced mode (EFR[4]=1), the UART compares the incoming receiver data with the XOFF2 value. Upon a correct match, the received data will be transferred to the RHR and a level 5 interrupt (XOFF or special character) will be asserted if level 5 interrupts are enabled (IER[5] set to 1).

**EFR[6]: Enable automatic RTS flow control.**

logic 0 ⇒ RTS flow control is disabled (default).

logic 1 ⇒ RTS flow control is enabled in Enhanced mode (i.e. EFR[4] = 1), where the RTS# pin will be forced inactive high if the RFL reaches the upper flow control threshold. This will be released when the RFL drops below the lower threshold. The 650 and 950 software drivers should use this bit to enable RTS flow control. The 750 compatible driver uses MCR[5] to enable RTS flow control.

**EFR[7]: Enable automatic CTS flow control.**

logic 0 ⇒ CTS flow control is disabled (default).

logic 1 ⇒ CTS flow control is enabled in Enhanced mode (i.e. EFR[4] = 1), where the data transmission is prevented whenever the CTS# pin is held inactive high. The 650 and 950 software drivers should use this bit to enable CTS flow control. The 750 compatible driver uses MCR[5] to enable CTS flow control.

## 13.2 Special Character Detection

In Enhanced mode (EFR[4]=1), when special character detection is enabled (EFR[5]=1) and the receiver matches received data with XOFF2, the 'received special character' flag ASR[4] will be set and a level 5 interrupt is asserted, (if enabled by IER[5]). This flag will be cleared following a read of ASR. The received status (i.e. parity and framing) of special characters does not have to be valid for these characters to be accepted as valid matches.

## 13.3 Automatic In-band Flow Control

When in-band receive flow control is enabled, the UART will compare the received data with XOFF1 or XOFF2 characters to detect an XOFF condition. When this occurs, the UART will disable transmission as soon as any current character transmission is complete. Status bits ISR[4] and ASR[0] will be set. A level 5 interrupt will occur if enabled by IER[5]. The UART will then compare all received data with XON1 or XON2 characters to detect an XON condition. When this occurs, the UART will re-enable transmission and status bits ISR[4] and ASR[0] will be cleared.

Any valid XON/XOFF characters will not be written into the RHR. An exception to this rule occurs if special character detection is enabled and an XOFF2 character is received that is a valid XOFF. In this instance, the character will be written into the RHR.

The received status (i.e. parity and framing) of XON/XOFF characters does not have to be valid for these characters to be accepted as valid matches.

When the 'XON Any' flag (MCR[5]) is set, any received character is accepted as a valid XON condition and the transmitter will be re-enabled. The received data will be transferred to the RHR.

When in-band transmit flow control is enabled, the RFL will be sampled whenever the transmitter is idle (briefly, between characters, or when the THR is empty) and an XON/XOFF character may be inserted into the data stream if needed. Initially, remote transmissions are enabled and hence ASR[1] is clear. If ASR[1] is clear and the RFL has passed the upper trigger level (i.e. is above the trigger level), XOFF will be sent and ASR[1] will be set. If ASR[1] is set and the RFL falls below the lower trigger level, XON will be sent and ASR[1] will be cleared.

If transmit flow control is disabled after an XOFF has been sent, an XON will be sent automatically.

## 13.4 Automatic Out-of-band Flow Control

Automatic RTS/CTS flow control is selected by different means, depending on whether the UART is in Enhanced or non-Enhanced mode. When in non-Enhanced mode, MCR[5] enables both RTS and CTS flow control. When in Enhanced mode, EFR[6] enables automatic RTS flow control and EFR[7] enables automatic CTS flow control. This allows software compatibility with both 16C650 and 16C750 drivers.

When automatic CTS flow control is enabled and the CTS# input becomes active, the UART will disable transmission as soon as any current character transmission is complete. Transmission is resumed whenever the CTS# input becomes inactive.

When automatic RTS flow control is enabled, the RTS# pin will be forced inactive when the RFL reaches the upper trigger level and will return to active when the RFL falls below the lower trigger level. The automatic RTS# flow control is ANDed with MCR[1] and hence is only operational when MCR[1]=1. This allows the software driver to override the automatic flow control and disable the remote transmitter regardless by setting MCR[1]=0 at any time.

Automatic DTR/DSR flow control behaves in the same manner as RTS/CTS flow control but is enabled by ACR[3:2], regardless of whether or not the UART is in Enhanced mode.

# 14 BAUD RATE GENERATION

## 14.1 General Operation

The UART contains a programmable baud rate generator that is capable of taking any clock input from DC to 60MHz (at 5V) and dividing it by any 16-bit divisor number from 1 to 65535 written into the DLM (MSB) and DLL (LSB) registers. In addition to this, a clock prescaler register is provided which can further divide the clock by values in the range 1.0 to 31.875 in steps of 0.125. Also, a further feature is the Times Clock Register 'TCR' which allows the sampling clock to be set to any value between 4 and 16.

These clock options allow for highly flexible baud rate generation capabilities from almost any input clock frequency (up to 60MHz). The actual transmitter and receiver baud rate is calculated as follows:

$$BaudRate = \frac{InputClock}{SC * Divisor * prescaler}$$

*Where*:

SC = Sample clock values defined in TCR[3:0]
Divisor = DLL + ( 256 x DLM )
Prescaler = 1 when MCR[7] = '0' else:
= M + ( N / 8 ) where:
M = CPR[7:3] (Integer part – 1 to 31)
N = CPR[2:0] (Fractional part – 0.000 to 0.875 )

See next section for a discussion of the clock prescaler and times clock register.

After a hardware reset, the prescaler is bypassed (set to 1) and TCR is set to 0x00 (i.e. SC = 16). Assuming this default configuration, the following table gives the divisors required to be programmed into the DLL and DLM registers in order to obtain various standard baud rates:

| DLM:DLL Divisor Word | Baud Rate (bits per second) |
|---|---|
| 0x0900 | 50 |
| 0x0300 | 110 |
| 0x0180 | 300 |
| 0x00C0 | 600 |
| 0x0060 | 1,200 |
| 0x0030 | 2,400 |
| 0x0018 | 4,800 |
| 0x000C | 9,600 |
| 0x0006 | 19,200 |
| 0x0004 | 28,800 |
| 0x0003 | 38,400 |
| 0x0002 | 57,600 |
| 0x0001 | 115,200 |

**Table 15: Standard PC COM Port Baud Rate Divisors (assuming a 1.8432MHz crystal)**

## 14.2 Clock Prescaler Register 'CPR'

The CPR register is located at offset 0x01 of the ICR

The prescaler divides the system clock by any value in the range of 1 to "31 7/8" in steps of 1/8. The divisor takes the form "M + N/8", where M is the 5 bit value defined in CPR[7:3] and N is the 3 bit value defined in CPR[2:0].

The prescaler is by-passed and a prescaler value of '1' is selected by default when MCR[7] = 0.

MCR[7] is set to the complement of CLKSEL pin after a hardware reset but may be overwritten by software. Note however that since access to MCR[7] is restricted to Enhanced mode only, EFR[4] should first be set and then MCR[7] set or cleared as required.

If CLKSEL is connected to ground or MCR[7] is set by software, the internal clock prescaler is enabled.

Upon a hardware reset, CPR defaults to 0x20 (division-by-4). Compatibility with existing 16C550 baud rate divisors is maintained using either a 1.8432MHz clock with CLKSEL pin connected to VDD, or a 7.372MHz clock with CLKSEL connected to GND. In the latter case, clearing MCR[7] would bypass the prescaler and hence quadruple all selected baud rates (providing a maximum of 460.8kbps as opposed to 115.2kbps)

For higher baud rates use a higher frequency clock, e.g. 14.7456MHz, 18.432MHz, 32MHz, 40MHz or 60.0MHz. The flexible prescaler allows system designers to generate popular baud rates using clocks that are not integer multiples of the required rate. When using a non-standard clock frequency, compatibility with existing 16C550 software drivers may be maintained with a minor software patch to program the on-board prescaler to divide the high frequency clock down to 1.8432MHz.

Table 17 on the following page gives the prescaler values required to operate the UARTs at compatible baud rates with various different crystal frequencies. Also given is the maximum available baud rates in TCR = 16 and TCR = 4 modes with CPR = 1.

## 14.3 Times Clock Register 'TCR'

The TCR register is located at offset 0x02 of the ICR

The 16C550 and other compatible devices such as 16C650 and 16C750 use a 16 times (16x) over-sampling channel clock. The 16x over-sampling clock means that the channel clock runs at 16 times the selected serial bit rate. It limits the highest baud rate to 1/16 of the system clock when using a divisor latch value of unity. However, the 16C950 UART is designed in a manner to enable it to accept other multiplications of the bit rate clock. It can use values from 4x to 16x clock as programmed in the TCR as long as the clock (oscillator) frequency error, stability and jitter are within reasonable parameters. Upon hardware reset the TCR is reset to 0x00 which means that a 16x clock will be used, for compatibility with the 16C550 and compatibles.

The maximum baud-rates available for various system clock frequencies at all of the allowable values of TCR are indicated in Table 18 on the following page. These are the values in bits-per-second (bps) that are obtained if the divisor latch = 0x01 and the Prescaler is set to 1.

The OX16C950 has the facility to operate at baud-rates up to 15 Mbps at 5V.

The table below indicates how the value in the register corresponds to the number of clock cycles per bit. TCR[3:0] is used to program the clock. TCR[7:4] are unused and will return "0000" if read.

| TCR[3:0] | Clock cycles per bit |
|---|---|
| 0000 to 0011 | 16 |
| 0100 to 1111 | 4-15 |

**Table 16: TCR Sample Clock Configuration**

The use of TCR does not require the device to be in 650 or 950 mode although only drivers that have been written to take advantage of the 950 mode features will be able to access this register. Writing 0x01 to the TCR will not switch the device into 1x isochronous mode, this is explained in the following section. (TCR has no effect in isochronous mode). If 0x01, 0x10 or 0x11 is written to TCR the device will operate in 16x mode.

Reading TCR will always return the last value that was written to it irrespective of mode of operation.

| Clock Frequency (MHz) | CPR value | Effective crystal frequency | Error from 1.8432MHz (%) | Max. Baud rate with CPR = 1, TCR = 16 | Max. Baud rate with CPR = 1, TCR = 4 |
|---|---|---|---|---|---|
| 1.8432 | 0x08 (1.000) | 1.8432 | 0.00 | 115,200 | 460,800 |
| 7.3728 | 0x20(4.000) | 1.8432 | 0.00 | 460,800 | 1,843,200 |
| 14.7456 | 0x80 (8.000) | 1.8432 | 0.00 | 921,600 | 3,686,400 |
| 18.432 | 0x50 (10.000) | 1.8432 | 0.00 | 1,152,000 | 4,608,000 |
| 32.000 | 0x8B(17.375) | 1.8417 | 0.08 | 2,000,000 | 8,000,000 |
| 33.000 | 0x8F (17.875) | 1.8462 | 0.16 | 2,062,500 | 8,250,000 |
| 40.000 | 0xAE (21.750) | 1.8391 | 0.22 | 2,500,000 | 10,000,000 |
| 50.000 | 0xD9 (27.125) | 1.8433 | 0.01 | 3,125,000 | 12,500,000 |
| 60.000* | 0xFF (31.875) | 1.8824 | 2.13 | 3,750,000 | 15,000,000 |

**Table 17: Example clock options and their assosiacted maximum baud rates**

| Sampling Clock | TCR Value | System Clock (MHz) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1.8432 | 7.372 | 14.7456 | 18.432 | 32 | 40 | 50 | 60 |
| 16 | 0x00 | 115,200 | 460,750 | 921,600 | 1.152M | 2.00M | 2.50M | 3.125M | 3.75M |
| 15 | 0x0F | 122,880 | 491,467 | 983,040 | 1,228,800 | 2,133,333 | 2,666,667 | 3,333,333 | 4.00M |
| 14 | 0x0E | 131,657 | 526,571 | 1,053,257 | 1,316,571 | 2,285,714 | 2,857,143 | 3,571,429 | 4,285,714 |
| 13 | 0x0D | 141,785 | 567,077 | 1,134,277 | 1,417,846 | 2,461,538 | 3,076,923 | 3,846,154 | 4,615,384 |
| 12 | 0x0C | 153,600 | 614,333 | 1,228,800 | 1,536,000 | 2,666,667 | 3,333,333 | 4,166,667 | 5.00M |
| 11 | 0x0B | 167,564 | 670,182 | 1,340,509 | 1,675,636 | 2,909,091 | 3,636,364 | 4,545,455 | 5,454545 |
| 10 | 0x0A | 184,320 | 737,200 | 1,474,560 | 1,843,200 | 3.20M | 4.00M | 5.00M | 6.00M |
| 9 | 0x09 | 204,800 | 819,111 | 1,638,400 | 2,048,000 | 3,555,556 | 4,444,444 | 5,555,556 | 6,666,667 |
| 8 | 0x08 | 230,400 | 921,500 | 1,843,200 | 2,304,000 | 4.00M | 5.00M | 6.25M | 7.50M |
| 7 | 0x07 | 263,314 | 1,053,143 | 2,106,514 | 2,633,143 | 4,571,429 | 5,714,286 | 7,142,857 | 8,571428 |
| 6 | 0x06 | 307,200 | 1,228,667 | 2,457,600 | 3,072,000 | 5,333,333 | 6,666,667 | 8,333,333 | 10.00M |
| 5 | 0x05 | 368,640 | 1,474,400 | 2,949,120 | 3,686,400 | 6.40M | 8.00M | 10.00M | 12.00M |
| 4 | 0x04 | 460,800 | 1,843,000 | 3,686,400 | 4,608,000 | 8.00M | 10.00M | 12.50M | 15.00M |

**Table 18: Maximum Baud Rates Available at all 'TCR' Sampling Clock Values**

## 14.4 Input Clock Options

A system clock must be applied to XTLI pin on the device (or CLKSEL if selected by software). The speed of this clock determines the maximum baud rate at which the device can receive and transmit serial data. This maximum is equal to one sixteenth of the frequency of the system clock (Increasing to one quarter of this value if TCR=4 is used).

The industry standard system clock for PC COM ports is 1.8432 MHz, limiting the maximum baud rate to 115.2 Kbps. The OX16C95x UARTs support system clocks up to 50MHz (60MHz for the OX16C950 at 5V) and its flexible baud rate generation hardware means that almost any frequency can be optionally scaled down for compatibility with standard devices.

Designers have the option of using either TTL clock modules or crystal oscillator circuits for system clock input, with minimal additional components. The following two sections describe how each can be connected.

## 14.5 TTL Clock Module

Using a TTL module for the system clock simply requires the module to be supplied with +5v power and GND connections. The clock output can then be connected directly to XTLI. XTLO should be left unconnected.

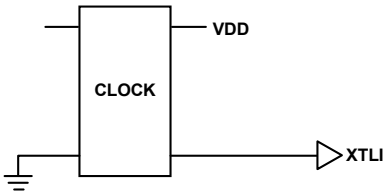



**Figure 2: TTL Clock Module Connectivity**

## 14.6 External 1x Clock Mode

The transmitter and receiver can accept an external clock applied to the RI# and DSR# pins respectively. The clock options are selected using the clock select register (CKS - see section 15.8). The transmitter and receiver may be configured to operate in 1x (Isochronous) mode by setting CKS[7] and CKS[3], respectively. In Isochronous mode, transmitter or receiver will use the 1x clock (usually but not necessarily an external source) where asynchronous framing is maintained using start, parity and stop-bits. However serial transmission and reception is synchronised to the 1x clock. In this mode asynchronous data may be transmitted at baud rates up to 60Mbps. The local 1x clock source can be asserted on the DTR# pin.

Note that line drivers need to be capable of transmission at data rates twice the system clock used (as one cycle of the system clock corresponds to 1 bit of serial data). Also note that enabling modem interrupts is illegal in isochronous mode, as the clock signal will cause a continuous change to the modem status (unless masked in MDM register, see section 15.10).

## 14.7 Crystal Oscillator Circuit

The OX16C950 may be clocked by a crystal connected to XTLI and XTLO or directly from a clock source connected to the XTLI pin (or CLKSEL if selected by software). The circuit required to use the on-chip oscillator is shown opposite.

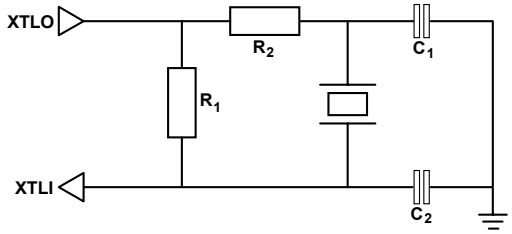



**Figure 3: Crystal Oscillator Circuit**

| Frequency Range (MHz) | C1 (pF) | C2 (pF) | R1 (Ω) | R2 (Ω) |
|---|---|---|---|---|
| 1.8432 – 8 | 68 | 22 | 220K | 470R |
| 8-60 | 33-68 | 33 – 68 | 220K-2M2 | 470R |

**Table 19: Component Values**

Note: For better stability use a smaller value of $R_1$. Increase $R_1$ to reduce power consumption.
The total capacitive load (C1 in series with C2) should be that specified by the crystal manufacturer (nominally 16pF)

# 15 ADDITIONAL FEATURES

## 15.1 Additional Status Register 'ASR'

**ASR[0]: Transmitter disabled**

logic 0 ⇒ The transmitter is not disabled by in-band flow control.

logic 1 ⇒ The receiver has detected an XOFF, and has disabled the transmitter.

This bit is cleared after a hardware reset or channel software reset. The software driver may write a 0 to this bit to re-enable the transmitter if it was disabled by in-band flow control. Writing a 1 to this bit has no effect.

**ASR[1]: Remote transmitter disabled**

logic 0 ⇒ The remote transmitter is not disabled by in-band flow control.

logic 1 ⇒ The transmitter has sent an XOFF character, to disable the remote transmitter. (Cleared when a subsequent XON is sent).

This bit is cleared after a hardware reset or channel software reset. The software driver may write a 0 to this bit to re-enable the remote transmitter (an XON is transmitted). Writing a 1 to this bit has no effect.

**Note**: The remaining bits (ASR[7:2]) of this register are read only

**ASR[2]: RTS**

This is the complement of the actual state of the RTS# pin when the device is not in loopback mode. The driver software can determine if the remote transmitter is disabled by RTS# out-of-band flow control by reading this bit. In loopback mode this bit reflects the flow control status rather than the pin's actual state.

**ASR[3]: DTR**

This is the complement of the actual state of the DTR# pin when the device is not in loopback mode. The driver software can determine if the remote transmitter is disabled by DTR# out-of-band flow control by reading this bit. In loopback mode this bit reflects the flow control status rather than the pin's actual state.

**ASR[4]: Special character detected**

logic 0 ⇒ No special character has been detected.

logic 1 ⇒ A special character has been received and is stored in the RHR.

This can be used to determine whether a level 5 interrupt was caused by receiving a special character rather than an XOFF. The flag is cleared following the read of the ASR.

**ASR[5]: FIFOSEL**

This bit reflects the unlatched state of the FIFOSEL pin.

**ASR[6]: FIFO size**

logic 0 ⇒ FIFOs are 16 deep if FCR[0] = 1.

logic 1 ⇒ FIFOs are 128 deep if FCR[0] = 1.

Note: If FCR[0] = 0, the FIFOs are 1 deep.

**ASR[7]: Transmitter Idle**

logic 0 ⇒ Transmitter is transmitting.

logic 1 ⇒ Transmitter is idle.

This bit reflects the state of the internal transmitter. It is set when both the transmitter FIFO and shift register are empty.

## 15.2 FIFO Fill levels 'TFL & RFL'

The number of characters stored in the THR and RHR can be determined by reading the TFL and RFL registers respectively. As the UART clock is asynchronous with respect to the processor, it is possible for the levels to change during a read of these FIFO levels. It is therefore recommended that the levels are read twice and compared to check that the values obtained are valid. The values should be interpreted as follows:

1. The number of characters in the THR is no greater than the value read back from TFL.
2. The number of characters in the RHR is no less than the value read back from RFL.

## 15.3 Additional Control Register 'ACR'

The ACR register is located at offset 0x00 of the ICR

**ACR[0]: Receiver disable**

logic 0 ⇒ The receiver is enabled, receiving data and storing it in the RHR.

logic 1 ⇒ The receiver is disabled. The receiver continues to operate as normal to maintain the framing synchronisation with the receive data stream but received data is not stored into the RHR. In-band flow control characters continue to be detected and acted upon. Special characters will not be detected.

Changes to this bit will only be recognised following the completion of any data reception pending.

**ACR[1]: Transmitter disable**

logic 0 ⇒ The transmitter is enabled, transmitting any data in the THR.

logic 1 ⇒ The transmitter is disabled. Any data in the THR is not transmitted but is held. However, in-band flow control characters may still be transmitted.

Changes to this bit will only be recognised following the completion of any data transmission pending.

**ACR[2]: Enable automatic DSR flow control**

logic 0 ⇒ Normal. The state of the DSR# line does not affect the flow control.

logic 1 ⇒ Data transmission is prevented whenever the DSR# pin is held inactive high.

This bit provides another automatic out-of-band flow control facility using the DSR# line.

**ACR[4:3]: DTR# line configuration**

When bits 4 or 5 of CKS (offset 0x03 of ICR) are set, the transmitter 1x clock or the output of the baud rate generator (Nx clock) are asserted on the DTR# pin, otherwise the DTR# pin is defined as follows:

logic [00] ⇒ DTR# is compatible with 16C450, 16C550, 16C650 and 16C750 (i.e. normal).

logic [01] ⇒ DTR# pin is used for out-of-band flow control. It will be forced inactive high if the Receiver FIFO Level ('RFL') reaches the upper flow control threshold. DTR# line will be re-activated when the RFL drops below the lower threshold (see FCL & FCH).

logic [10] ⇒ DTR# pin is configured to drive the active low enable pin of an external RS485 buffer. In this configuration the DTR# pin will be forced low whenever the transmitter is not empty (LSR[6]=0), otherwise DTR# pin is high.

logic [11] ⇒ DTR# pin is configured to drive the active-high enable pin of an external RS485 buffer. In this configuration, the DTR# pin will be forced high whenever the transmitter is not empty (LSR[6]=0), otherwise DTR# pin is low.

If the user sets ACR[4], then the DTR# line is controlled by the status of the transmitter empty bit of LCR. When ACR[4] is set, ACR[3] is used to select active high or active low enable signals. In half-duplex systems using RS485 protocol, this facility enables the DTR# line to directly control the enable signal of external 3-state line driver buffers. When the transmitter is empty the DTR# would go inactive once the SOUT line returns to it's idle marking state.

**ACR[5]: 950 mode trigger levels enable**

logic 0 ⇒ Interrupts and flow control trigger levels are as described in FCR register and are compatible with 16C650/16C750 modes.

logic 1 ⇒ 16C950 specific enhanced interrupt and flow control trigger levels defined by RTL, TTL, FCL and FCH are enabled.

**ACR[6]: ICR read enable**

logic 0 ⇒ The Line Status Register is readable.

logic 1 ⇒ The Indexed Control Registers are readable.

Setting this bit will map the ICR set to the LSR location for reads. During normal operation this bit should be cleared.

**ACR[7]: Additional status enable**

logic 0 ⇒ Access to the ASR, TFL and RFL registers is disabled.

logic 1 ⇒ Access to the ASR, TFL and RFL registers is enabled.

When ACR[7] is set, the MCR and LCR registers are no longer readable but remain writable, and the TFL and RFL registers replace them in the memory map for read operations. The IER register is replaced by the ASR register for all operations. The software driver may leave this bit set during normal operation, since MCR, LCR and IER do not generally need to be read.

### 15.4 Transmitter Trigger Level 'TTL'

The TTL register is located at offset 0x04 of the ICR

Whenever 950 trigger levels are enabled (ACR[5]=1), bits 4 and 5 of FCR are ignored and an alternative arbitrary transmitter interrupt trigger level can be defined in the TTL register. This 7-bit value provides a fully programmable transmitter interrupt trigger facility. In 950 mode, a priority level 3 interrupt occurs indicating that the transmitter buffer requires more characters when the interrupt is not masked (IER[1]=1) and the transmitter FIFO level falls below the value stored in the TTL register. The value 0 (0x00) has a special meaning. In 950 mode when the user writes 0x00 to the TTL register, a level 3 interrupt only occurs when the FIFO and the transmitter shift register are both empty and the SOUT line is in the idle marking state. This feature is particularly useful to report back the empty state of the transmitter after its FIFO has been flushed away.

### 15.5 Receiver Interrupt. Trigger Level 'RTL'

The RTL register is located at offset 0x05 of the ICR

Whenever 950 trigger levels are enabled (ACR[5]=1), bits 6 and 7 of FCR are ignored and an alternative arbitrary receiver interrupt trigger level can be defined in the RTL register. This 7-bit value provides a fully programmable receiver interrupt trigger facility as opposed to the limited trigger levels available in 16C650 and 16C750 devices. It enables the system designer to optimise the interrupt performance hence minimising the interrupt overhead.

In 950 mode, a priority level 2 interrupt occurs indicating that the receiver data is available when the interrupt is not masked (IER[0]=1) and the receiver FIFO level reaches the value stored in this register.

### 15.6 Flow Control Levels 'FCL & FCH'

The FCL and FCH registers are located at offsets 0x06 and 0x07 of the ICR respectively

Enhanced software flow control using XON/XOFF and hardware flow control using RTS#/CTS# and DTR#/DSR# are available when 950 mode trigger levels are enabled (ACR[5]=1). Improved flow control threshold levels are offered using Flow Control Lower trigger level ('FCL') and Flow Control Higher trigger level ('FCH') registers to provide a greater degree of flexibility when optimising the flow control performance. Generally, these facilities are only available in Enhanced mode.

In 650 mode, in-band flow control is enabled using the EFR register. An XOFF character is transmitted when the receiver FIFO exceeds the upper trigger level defined by FCR[7:6] as described in section 8.1. An XON is then sent when the FIFO is read down to the lower fill level. The flow control is enabled and the appropriate mode selected using EFR[3:0].

In 950 mode, the flow control thresholds defined by FCR[7:6] are ignored. In this mode threshold levels are programmed using FCL and FCH. When in-band flow control is enabled (defined by EFR[3:0]) and the receiver FIFO level ('RFL') reaches the value programmed in the FCH register, an XOFF is transmitted to stop the flow of serial data . The flow is resumed when the receiver FIFO fill level falls to below the value programmed in the FCL register, at which point an XON character is sent. The FCL value of 0x00 is illegal.

For example if FCL and FCH contain 64 and 100 respectively, XOFF is transmitted when the receiver FIFO contains 100 characters, and XON is transmitted when sufficient characters are read from the receiver FIFO such that there are 63 characters remaining.

CTS/RTS and DSR/DTR out-of-band flow control use the same trigger levels as in-band flow control. When out-of-band flow control is enabled, RTS# (or DTR#) line is de-asserted when the receiver FIFO level reaches the upper limit defined in the FCH and is re-asserted when the receiver FIFO is drained below the lower limit defined in FCL. When 950 trigger levels are enabled (ACR[5]=1), the CTS# flow control functions as in 650 mode and is configured by EFR[7]. However, when EFR[6] is set, RTS# is automatically de-asserted when RFL reaches FCH and re-asserted when RFL drops below FCL.

DSR# flow control is configured with ACR[2]. DTR# flow control is configured with ACR[4:3].

### 15.7 Device Identification Registers

The identification registers is located at offsets 0x08 to 0x0B of the ICR

The OX16C950 offers four bytes of device identification. The device ID registers may be read using offset values 0x08 to 0x0B of the Indexed Control Register. Registers ID1, ID2 and ID3 identify the device as an OX16C950 and return 0x16, 0xC9 and 0x50 respectively. The REV register resides at offset 0x0B of ICR and identifies the revision of 950 core. This register returns 0x03 for revision B of the OX16C950.

## 15.8 Clock Select Register 'CKS'

The CKS register is located at offset 0x03 of the ICR

This register is cleared to 0x00 after a hardware reset to maintain compatibility with 16C550, but is unaffected by software reset. This allows the user to select a clock source and then reset the channel to work-around any timing glitches.

**CKS[1:0]: Receiver Clock Source Selector**

logic [00] ⇒ The RCLK pin is selected for the receiver clock (550 compatible mode).

logic [01] ⇒ The DSR# pin is selected for the receiver clock.

logic [10] ⇒ The output of baud rate generator (internal BDOUT#) is selected for the receiver clock.

logic [11] ⇒ The transmitter clock is selected for the receiver. This allows RI# to be used for both transmitter and receiver.

**CKS[2]: Disable BDOUT# pin**

logic 0 ⇒ The BDOUT# pin is enabled and connected to the output of the internal baud rate generator which is a Nx clock used by the UART. In 16C550 compatibility mode, the baud rate generator produces a 16x clock (See TCR, section 14.3).

logic 1 ⇒ The BDOUT# pin is disabled and set permanently low.

**CKS[3]: Receiver 1x clock mode selector**

logic 0 ⇒ The receiver is in Nx clock mode as defined in the TCR register. After a hardware reset the receiver operates in 16x clock mode, i.e. 16C550 compatibility.

logic 1 ⇒ The receiver is in isochronous 1x clock mode.

**CKS[5:4]: Transmitter 1x clock or baud rate generator output (BDOUT) on DTR# pin**

logic [00] ⇒ The function of the DTR# pin is defined by the setting of ACR[4:3].

logic [01] ⇒ The transmitter 1x clock (bit rate clock) is asserted on the DTR# pin and the setting of ACR[4:3] is ignored.

logic [10] ⇒ The output of baud rate generator (Nx clock) is asserted on the DTR# pin and the setting of ACR[4:3] is ignored.

logic [11] ⇒ Reserved.

**CKS[6]: Transmitter clock source selector**

logic 0 ⇒ The transmitter clock source is the output of the baud rate generator (550 compatibility).

logic 1 ⇒ The transmitter uses an external clock applied to the RI# pin.

**CKS[7]: Transmitter 1x clock mode selector**

logic 0 ⇒ The transmitter is in Nx clock mode as defined in the TCR register. After a hardware reset the transmitter operates in 16x clock mode, i.e. 16C550 compatibility.

logic 1 ⇒ The transmitter is in isochronous 1x clock mode.

## 15.9 Nine-bit Mode Register 'NMR'

The NMR register is located at offset 0x0D of the ICR

The OX16C950 offers 9-bit data framing for industrial multi-drop applications. 9-bit mode is enabled by setting bit 0 of the Nine-bit Mode Register (NMR). In 9-bit mode the data length setting in LCR[1:0] is ignored. Furthermore as parity is permanently disabled, the setting of LCR[5:3] is also ignored.

The receiver stores the 9th bit of the received data in LSR[2] (where parity error is stored in normal mode). Note that OX16C950 provides a 128-deep FIFO for LSR[3:1]. The transmitter FIFO is 9-bit wide and 128 deep. The user should write the 9th (MSB) data bit in SPR[0] first and then write the other 8 bits to THR.

As parity mode is disabled, LSR[7] is set whenever there is an overrun, framing error or received break condition. It is unaffected by the contents of LSR[2] (Now the received 9th data bit).

In 9-bit mode, in-band flow control is disabled regardless of the setting of EFR[3:0] and the XON1/XON2/XOFF1 and XOFF2 registers are used for special character detection.

*Interrupts in 9-Bit Mode:*

While IER[2] is set, upon receiving a character with status error, a level 1 interrupt is asserted when the character and the associated status are transferred to the FIFO.

The OX16C950 can assert an optional interrupt if a received character has its 9th bit set. As multi-drop systems often use the 9th bit as an address bit, the receiver is able to generate an interrupt upon receiving an address character. This feature is enabled by setting NMR[2]. This will result in a level 1 interrupt being asserted when the address character is transferred to the receiver FIFO.

In this case, as long as there are no errors pending, i.e. LSR[1], LSR[3], and LSR[4] are clear, '0' can be read back from LSR[7] and LSR[1], thus differentiating between an 'address' interrupt and receiver error or overrun interrupt in 9-bit mode. Note however that should an overrun or error interrupt actually occur, an address character may also reside in the FIFO. In this case, the software driver should examine the contents of the receiver FIFO as well as process the error.

The above facility produces an interrupt for recognizing any 'address' characters. Alternatively, the user can configure OX16C950 to match the receiver data stream with up to four programmable 9-bit characters and assert a level 5 interrupt after detecting a match. The interrupt occurs when the character is transferred to the FIFO (See below).

**NMR[0]: 9-bit mode enable**

logic 0 ⇒ 9-bit mode is disabled.

logic 1 ⇒ 9-bit mode is enabled.

**NMR[1]: Enable interrupt when 9th bit is set**

logic 0 ⇒ Receiver interrupt for detection of an 'address' character (i.e. 9th bit set) is disabled.

logic 1 ⇒ Receiver interrupt for detection of an 'address' character (i.e. 9th bit set) is enabled and a level 1 interrupt is asserted.

*Special Character Detection*

While the UART is in both 9-bit mode and Enhanced mode, setting IER[5] will enable detection of up to four 'address' characters. The least significant eight bits of these four programmable characters are stored in special characters 1 to 4 (XON1, XON2, XOFF1 and XOFF2 in 650 mode) registers and the 9th bit of these characters are programmed in NMR[5] to NMR[2] respectively.

**NMR[2]: Bit 9 of Special Character 1**
**NMR[3]: Bit 9 of Special Character 2**
**NMR[4]: Bit 9 of Special Character 3**
**NMR[5]: Bit 9 of Special Character 4**

**NMR[7:6]: Reserved**

Bits 6 and 7 of NMR are always cleared and reserved for future use.

## 15.10 Modem Disable Mask 'MDM'

The MDM register is located at offset 0x0E of the ICR

This register is cleared after a hardware reset to maintain compatibility with 16C550. It allows the user to mask interrupts and control sleep operation due to individual modem lines or the serial input line.

**MDM[0]: Disable delta CTS**

logic 0 ⇒ Delta CTS is enabled. It can generate a level 4 interrupt when enabled by IER[3]. Delta CTS can wake up the UART when it is asleep under auto-sleep operation.

logic 1 ⇒ Delta CTS is disabled. It can not generate an interrupt or wake up the UART.

**MDM[1]: Disable delta DSR**

logic 0 ⇒ Delta DSR is enabled. It can generate a level 4 interrupt when enabled by IER[3]. Delta DSR can wake up the UART when it is asleep under auto-sleep operation.

logic 1 ⇒ Delta DSR is disabled. In can not generate an interrupt or wake up the UART.

**MDM[2]: Disable Trailing edge RI**

logic 0 ⇒ Trailing edge RI is enabled. It can generate a level 4 interrupt when enabled by IER[3]. Trailing edge RI can wake up the UART when it is asleep under auto-sleep operation.

logic 1 ⇒ Trailing edge RI is disabled. In can not generate an interrupt or wake up the UART.

**MDM[3]: Disable delta DCD**

logic 0 ⇒ Delta DCD is enabled. It can generate a level 4 interrupt when enabled by IER[3]. Delta DCD can wake up the UART when it is asleep under auto-sleep operation.

logic 1 ⇒ Delta DCD is disabled. In can not generate an interrupt or wake up the UART.

**MDM[7:4]: Reserved**

These bits must be set to '0000'

## 15.11 Readable FCR 'RFC'

The RFC register is located at offset 0x0F of the ICR

This read-only register returns the current state of the FCR register (Note that FCR is write-only). This register is included for diagnostic purposes.

## 15.12 Good-data status register 'GDS'

The GDS register is located at offset 0x10 of the ICR
Good data status is set when the following conditions are true:

- ISR reads level0 (no interrupt), level2 or 2a (receiver data) or level3 (THR empty) interrupt.
- LSR[7] is clear i.e. no parity error, framing error or break in the FIFO.
- LSR[1] is clear i.e. no overrun error has occurred.

**GDS[0]: Good Data Status**

**GDS[7:1]: Reserved**

## 15.13 DMA Status Register 'DMS'

The DMS register is located at offset 0x11 of the ICR. This allows the TXRDY# and RXRDY# lines to be permanently deasserted, and the current internal status to be monitored. This mainly has applications for testing.
**DMS[0]: RxRdy Status**
Read Only: set when RxRdy is asserted (pin driven low).
**DMS[1]: TxRdy Status**
Read Only: set when TxRdy is asserted (pin driven low).
**DMS[5:2] Reserved**
**DMS[6]: Force RxRdy Inactive**
logic 0 ⇒ RxRdy# acts normally
logic 1 ⇒ RxRdy# is permanently inactive (high) regardless of FIFO thresholds
**DMA[7]: Force TxRdy Inactive**
logic 0 ⇒ TxRdy# acts normally
logic 1 ⇒ TxRdy# is permanently inactive (high) regardless of FIFO thresholds.

## 15.14 Port Index Register 'PIX'

The PIX register is located at offset 0x12 of the ICR. This read-only register gives the UART index. For a single channel device such as the OX16C950 this reads '0'.

## 15.15 Clock Alteration Register 'CKA'

The CKA register is located at offset 0x13 of the ICR. This register adds additional clock control mainly for isochronous and embedded applications. The register is effectively an enhancement to the CKS register.
This register is cleared to 0x00 after a hardware reset to maintain compatibility with 16C550, but is unaffected by software reset. This allows the user to select a clock mode and then reset the channel to work-around any timing glitches.

# 16 OPERATING CONDITIONS

| Symbol | Parameter | Min. | Max. | Units |
|---|---|---|---|---|
| $V_{DD}$ | DC supply voltage | -0.3 | 7.0 | V |
| $V_{IN}$ | DC input voltage | -0.3 | $V_{DD}$ + 0.3 | V |
| $I_{IN}$ | DC input current | | +/- 10 | mA |
| $T_{STG}$ | Storage temperature | -40 | 125 | °C |

**Table 20: Absolute Maximum Ratings**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $V_{DD}$ | DC supply voltage | 3 | 5.25 | V |
| $T_O$ | Operating Temperature range | 0 | 70 | °C |

**Table 21: Recommended Operating Conditions**

N.B. Under 3V operation some inputs are 5V tolerant (I/O signals, CPU interface, and UART signals).

# 17 DC ELECTRICAL CHARACTERISTICS

## 17.1 5V Operation

.

| Symbol | Parameter | Condition | Min. | Max. | Units |
|---|---|---|---|---|---|
| $V_{DD}$ | Supply voltage | Commercial | 4.75 | 5.25 | V |
| $V_{IH}$ | Input high voltage | TTL Interface Note1<br>TTL Schmitt trigger | 2.0<br>2.4 | | V |
| $V_{IL}$ | Input low voltage | TTL Interface Note 1<br>TTL Schmitt trigger | | 0.8<br>0.6 | V |
| $C_{IL}$ | Capacitance of input buffers | | | 5.0 | pF |
| $C_{OL}$ | Capacitance of output buffers | | | 10.0 | pF |
| $I_{IH}$ | Input high leakage current | $V_{in} = V_{DD}$ | -10 | 10 | µA |
| $I_{IL}$ | Input low leakage current | $V_{in} = V_{SS}$ | -10 | 10 | µA |
| $V_{OH}$ | Output high voltage | $I_{OH}$ = 1 µA | $V_{DD}$ – 0.05 | | V |
| $V_{OH}$ | Output high voltage | $I_{OH}$ = 4 mA Note2 | 2.4 | | V |
| $V_{OL}$ | Output low voltage | $I_{OL}$ = 1 µA | | 0.05 | V |
| $V_{OL}$ | Output low voltage | $I_{OL}$ = 4 mA Note2 | | 0.4 | V |
| $I_{OZ}$ | 3-state output leakage current | | -10 | 10 | µA |
| $I_{ST}$ | Static current | $V_{in} = V_{DD}$ or $V_{SS}$ | 45 | 100 | µA |
| $I_{CC}$ | Operating supply current in normal mode Note3 | $f_{CK}$ = 1.8432 MHz<br>$f_{CK}$ = 7.372 MHz<br>$f_{CK}$ = 60.00 MHz | 0.40<br>1.50<br>10.0 | 2.0<br>5.0<br>30.0 | mA |
| | Operating supply current in sleep mode Note3 | $f_{CK}$ = 1.8432 MHz<br>$f_{CK}$ = 7.372 MHz<br>$f_{CK}$ = 60.00 MHz | 0.35<br>1.25<br>3.5 | 0.5<br>2.0<br>5.0 | |

**Table 22: DC Electrical Characteristics**

Note 1: All input buffers are TTL with the exception of RESET which is a Schmitt trigger buffer.
Note 2: $I_{OH}$ and $I_{OL}$ are 12 mA for DB[7:0] and 4 mA for all other outputs.
Note 3: For further details on operating current please refer to the'OX16C95x Test Document'.

## 17.2 3V Operation

These figures assume the TQFP package, with VSEL applied for 3V operation.
I/O signals, CPU interface and UART signals are 5V tolerant.

| Symbol | Parameter | Condition | Min. | Max. | Units |
|---|---|---|---|---|---|
| $V_{DD}$ | Supply voltage | Commercial | 3.0 | 3.45 | V |
| $V_{IH}$ | Input high voltage | TTL Interface Note1<br>TTL Schmitt trigger | 0.7 $V_{DD}$<br>TBD | $V_{DD}$ + 0.5<br>TBD | V |
| $V_{IL}$ | Input low voltage | TTL Interface Note 1<br>TTL Schmitt trigger | -0.5<br>TBD | 0.2 $V_{DD}$<br>TBD | V |
| $C_{IL}$ | Capacitance of input buffers | | | 5.0 | pF |
| $C_{OL}$ | Capacitance of output buffers | | | 10.0 | pF |
| $I_{IH}$ | Input high leakage current | $V_{in} = V_{DD}$ | -1 | 1 | μA |
| $I_{IL}$ | Input low leakage current | $V_{in} = V_{SS}$ | -1 | 1 | μA |
| $V_{OH}$ | Output high voltage | $I_{OH}$ = 1 μA | $V_{DD}$ – 0.05 | | V |
| $V_{OH}$ | Output high voltage | $I_{OH}$ = 4 mA Note2 | 2.4 | | V |
| $V_{OL}$ | Output low voltage | $I_{OL}$ = 1 μA | | 0.05 | V |
| $V_{OL}$ | Output low voltage | $I_{OL}$ = 4 mA Note2 | | 0.4 | V |
| $I_{OZ}$ | 3-state output leakage current | | -1 | 1 | μA |
| $I_{ST}$ | Static current | $V_{in} = V_{DD}$ or $V_{SS}$ | 45 | 100 | μA |
| $I_{CC}$ | Operating supply current in normal mode Note3 | $f_{CK}$ = 1.8432 MHz<br>$f_{CK}$ = 7.372 MHz<br>$f_{CK}$ = 60.00 MHz | TBD | TBD | mA |
| | Operating supply current in sleep mode Note3 | $f_{CK}$ = 1.8432 MHz<br>$f_{CK}$ = 7.372 MHz<br>$f_{CK}$ = 60.00 MHz | TBD | TBD | |

**Table 23: DC Electrical Characteristics**

Note 1: All input buffers are TTL with the exception of RESET which is a Schmitt trigger buffer.
Note 2: $I_{OH}$ and $I_{OL}$ are 6 mA for DB[7:0] and 2 mA for all other outputs.
Note 3: For further details on operating current please refer to the'OX16C95x Test Document'.

# 18 AC ELECTRICAL CHARACTERISTICS

## 18.1 5V Operation

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_{sa}$ | Address set-up time to IOR# or IOW# falling<br>Address set-up time to IOR or IOW rising | 0 | | ns |
| $t_{ha}$ | Address hold time after IOR# or IOW# rising Note1<br>Address hold time after IOR or IOW falling Note1 | 0 | | ns |
| $t_{sc}$ | Chip select set-up time to IOR# or IOW# falling<br>Chip select set-up time to IOR or IOW rising | 0 | | ns |
| $t_{hc}$ | Chip select hold time after IOR# or IOW# rising Note1<br>Chip select hold time after IOR or IOW falling Note1 | 0 | | ns |
| $t_{r1}$ | Pulse duration of IOR# or IOR | 25 | | ns |
| $t_{r2}$ | Delay between IOR# rising and IOR#/IOW# falling<br>Delay between IOR falling and IOR/IOW rising | 38 | | |
| $t_{acc}$ | Access time; Data valid after IOR# falling or IOR rising | | 20 | ns |
| $t_{df}$ | Data bus floating after IOR# rising or IOR rising | | 10 | ns |
| $t_{w1}$ | Pulse duration of IOW# or IOW | 25 | | ns |
| $t_{w2}$ | Delay between IOW# rising and IOR# /IOW# falling<br>Delay between IOW falling and IOR/IOW rising | 38 | | ns |
| $t_{sd}$ | Data set-up time to IOW# rising or IOW falling | 0 | | ns |
| $t_{hd}$ | Data hold time after IOW# rising or IOW falling | 3 | | ns |
| $t_{sac}$ | Address and chip select set-up time to ADS# rising Note2 | 0 | | ns |
| $t_{hac}$ | Address and chip select hold time after ADS# rising Note2 | 2 | | ns |
| $t_{a1}$ | Pulse duration of ADS# Note2 | 2 | | ns |
| $t_{had}$ | IOR#/IOW# rising or IOR/IOW falling to ADS# falling Note3 | | | ns |
| $t_{irs}$ | SIN set-up time to Isochronous input clock 'Rx_Clk_In rising Note4 | 1 | | ns |
| $t_{irh}$ | SIN hold time after Isochronous input clock 'Rx_Clk_In' rising Note4 | 3 | | ns |
| $t_{its}$ | SOUT valid after Isochronous output clock 'Tx_Clk_Out' falling Note4 | 0 | 4 | ns |

**Table 24: AC Electrical Characteristics**

Note 1: $t_{ha}$ and $t_{hc}$ timing constrains only apply to non-multiplexed arrangement where ADS# is permanently tied low.
Note 2: ADS# signal may be tied low if address is stable during read or write cycles.
Note 3: $t_{had}$, $t_{a1}$ and $t_{sac}$ timing constrains only apply to multiplexed arrangement where ADS# is used.
Note 4: In Isochronous mode, transmitter data is available after the falling edge of the x1 clock and the receiver data is sampled using the rising edge of the x1 clock. The system designer is should ensure that mark-to-space ratio of the x1 clock is such that the required set-up and hold timing constraint are met. One way of achieving this is to choose a crystal frequency which is twice the required data rate and then divide the clock by two using the on-board prescaler. In this case the mark-to-space ratio is 50/50 for the purpose of set-up and hold calculations.

## 18.2 3V Operation

***N.B. Maximum frequency of operation is downgraded under 3V operation to 50 MHz.***

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_{sa}$ | Address set-up time to IOR# or IOW# falling<br>Address set-up time to IOR or IOW rising | 0 | | ns |
| $t_{ha}$ | Address hold time after IOR# or IOW# rising Note1<br>Address hold time after IOR or IOW falling Note1 | 0 | | ns |
| $t_{sc}$ | Chip select set-up time to IOR# or IOW# falling<br>Chip select set-up time to IOR or IOW rising | 0 | | ns |
| $t_{hc}$ | Chip select hold time after IOR# or IOW# rising Note1<br>Chip select hold time after IOR or IOW falling Note1 | 0 | | ns |
| $t_{r1}$ | Pulse duration of IOR# or IOR | 35 | | ns |
| $t_{r2}$ | Delay between IOR# rising and IOR#/IOW# falling<br>Delay between IOR falling and IOR/IOW rising | 45 | | |
| $t_{acc}$ | Access time; Data valid after IOR# falling or IOR rising | | 28 | ns |
| $t_{df}$ | Data bus floating after IOR# rising or IOR rising | | 12 | ns |
| $t_{w1}$ | Pulse duration of IOW# or IOW | 35 | | ns |
| $t_{w2}$ | Delay between IOW# rising and IOR# /IOW# falling<br>Delay between IOW falling and IOR/IOW rising | 45 | | ns |
| $t_{sd}$ | Data set-up time to IOW# rising or IOW falling | 0 | | ns |
| $t_{hd}$ | Data hold time after IOW# rising or IOW falling | 4 | | ns |
| $t_{sac}$ | Address and chip select set-up time to ADS# rising Note2 | 0 | | ns |
| $t_{hac}$ | Address and chip select hold time after ADS# rising Note2 | 2 | | ns |
| $t_{a1}$ | Pulse duration of ADS# Note2 | 3 | | ns |
| $t_{had}$ | IOR#/IOW# rising or IOR/IOW falling to ADS# falling Note3 | 45 | | ns |
| $t_{irs}$ | SIN set-up time to Isochronous input clock 'Rx_Clk_In rising Note4 | 2 | | ns |
| $t_{irh}$ | SIN hold time after Isochronous input clock 'Rx_Clk_In' rising Note4 | 4 | | ns |
| $t_{its}$ | SOUT valid after Isochronous output clock 'Tx_Clk_Out' falling Note4 | 0 | 6 | ns |

**Table 25: AC Electrical Characteristics**

Note 1: $t_{ha}$ and $t_{hc}$ timing constrains only apply to non-multiplexed arrangement where ADS# is permanently tied low.
Note 2: ADS# signal may be tied low if address is stable during read or write cycles.
Note 3: $t_{had}$, $t_{a1}$ and $t_{sac}$ timing constrains only apply to multiplexed arrangement where ADS# is used.
Note 4: In Isochronous mode, transmitter data is available after the falling edge of the x1 clock and the receiver data is sampled using the rising edge of the x1 clock. The system designer is should ensure that mark-to-space ratio of the x1 clock is such that the required set-up and hold timing constraint are met. One way of achieving this is to choose a crystal frequency which is twice the required data rate and then divide the clock by two using the on-board prescaler. In this case the mark-to-space ratio is 50/50 for the purpose of set-up and hold calculations.

## 19 TIMING WAVEFORMS

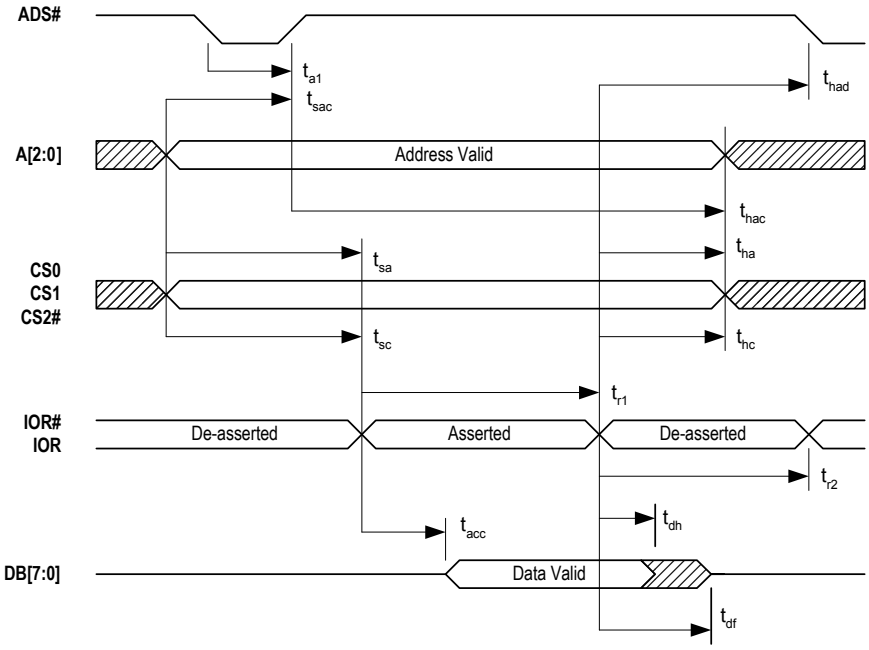



**Figure 4: Read Cycle Timing**

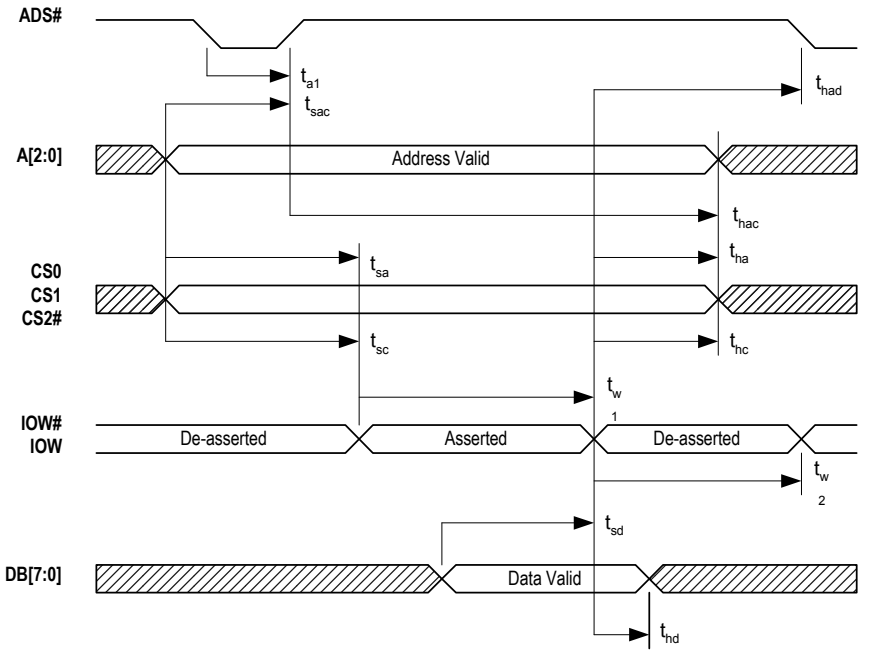



**Figure 5: Write Cycle Timing**

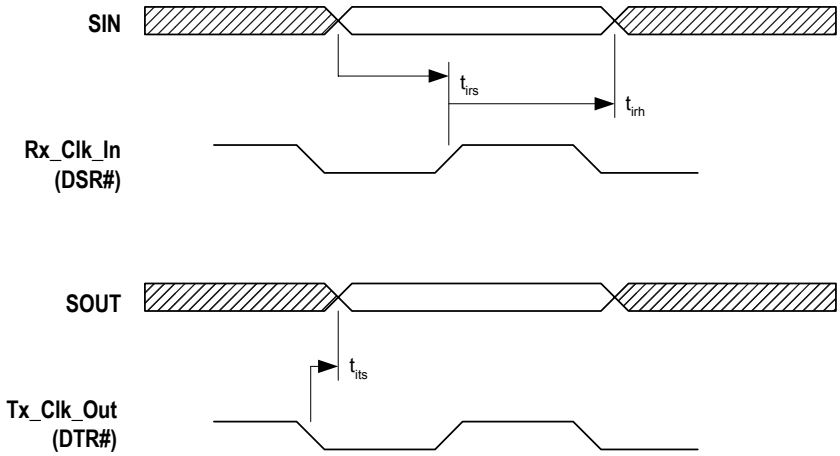



**Figure 6: Isochronous Mode Timing**

## 20 PACKAGE INFORMATION

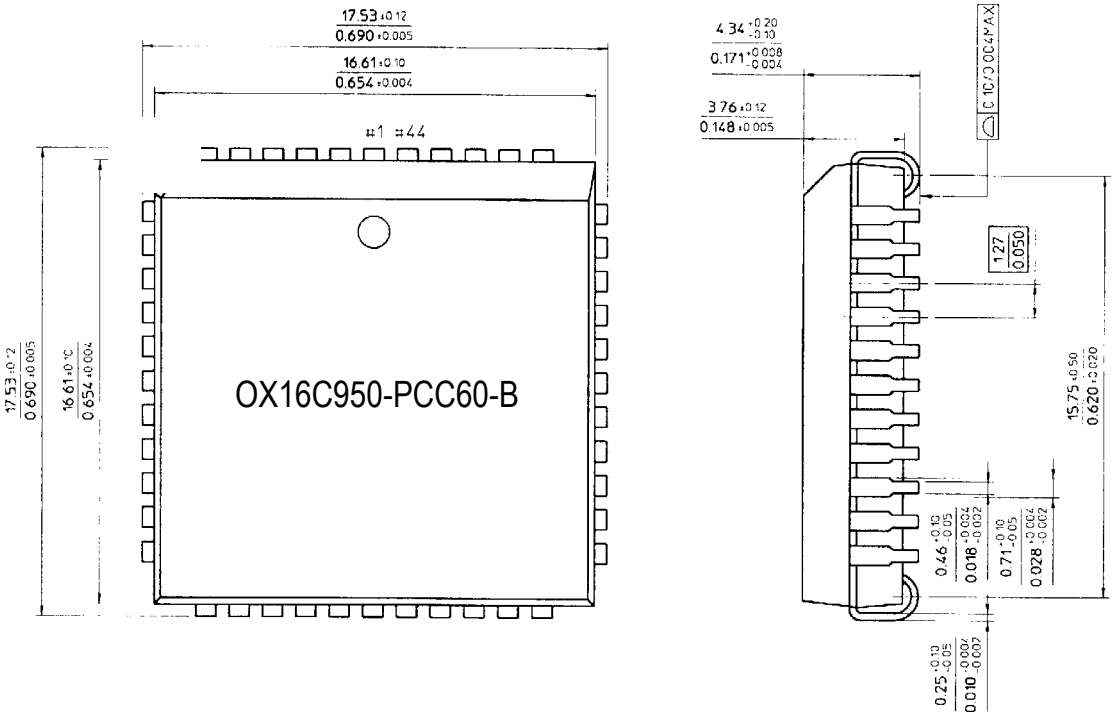



**Figure 7: 44 Pin Plastic Leaded Chip Carrier**

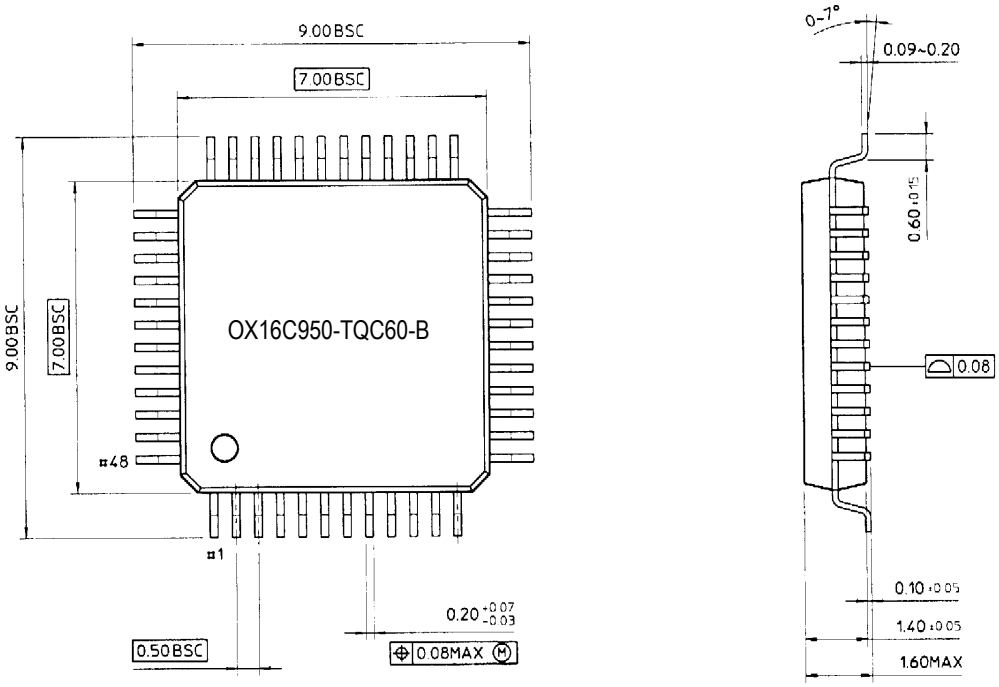



**Figure 8: 48 Pin Thin Quad Flat Pack (48 TQFP)**

## 21 ORDERING INFORMATION

**OX16C950-PCC60-B**

- Revision
- Operating Conditions -Commercial
- Package Type - 44 PLCC

**OX16C950-TQC60-B**

- Revision
- Package Material - Plastic
- Package Type - 48 TQFP

## NOTES

This page has intentionally been left blank.

## *CONTACT DETAILS*

**Oxford Semiconductor Ltd.**
25 Milton Park
Abingdon
Oxfordshire
OX14 4SH
United Kingdom

*Telephone:* +44 (0)1235 824900
*Fax:* +44 (0)1235 821141
*Sales e-mail:* sales@oxsemi.com
*Web site:* http://www.oxsemi.com

**©Copyright Oxford Semiconductor Ltd 1998-2001**
Oxford Semiconductor Ltd believes the information contained in this document to be accurate and reliable. However, it is subject to change without notice. No responsibility is assumed by Oxford Semiconductor for its use, nor for infringement of patents or other rights of third parties. No part of this publication may be reproduced, or transmitted in any form or by any means without the prior consent of Oxford Semiconductor Ltd. Oxford Semiconductor's terms and conditions of sale apply at all times.