OXCFU950 DATA SHEET

FEATURES

- USB/UART multi-function 16-bit PC Card device
- Host interface
  - 16-bit PC Card (PCMCIA revision 8.0) compliant
  - CompactFlash/CF+ (revision 3.0) compliant
- USB host controller
  - Supports USB 2.0/1.1 full-speed compliant (12 Mbps) & low speed (1.5 Mbps) devices
  - 8-Kbyte data buffer
  - Integrated OHCI-like controller supported by Oxford Semiconductor drivers
  - Supports control, bulk, interrupt & isochronous transfers
  - Supports USB PORT_OVER_CURRENT & PORT_POWER signals via MIO pins
- Integrated USB PHY transceivers
- Flexible OX16C95*x*-based UART
  - Supports 450 up to 950 (fully programmable)
  - Baud rates up to 12 Mbps
  - 128-byte deep transmit/receive FIFO
  - Software-compatible with standard 16C550
  - Readable FIFO levels
  - 9-bit data framing as well as 5,6,7 & 8
  - Detection of bad data in the receiver FIFO
  - Optional wakeup from low-power mode
  - Automated in-band flow control using programmable Xon/Xoff characters
  - Automated hardware flow control using CTS#/RTS# & DSR#/DTR#
  - Integral support for 4-byte I/O access for enhanced host-side data throughput
- 8-Kbitx8 EEPROM for CIS & default configuration. Programmable using Oxford Semiconductor utilities
- Power management for low-power operation
  - USB suspend & resume
  - UART sleep mode
- Single 12-MHz crystal oscillator for low power
- 4 multi-purpose I/O pins configurable as interrupts
- UART & MIO pins have variable I/O voltage levels from 1.8 V to 3.3 V nominal
- PC Card/CF+ I/Os are all 3.3 V with 5-V tolerance
- Single supply voltage range 2.85 V to 3.6 V
- Internal regulation for 1.8 V
- Supply current
  - Less than 4 mA idle
  - Less than 13 mA active
- Industrial temperature range –40°C to +85°C
- 64-pin QFN (9 mm x 9 mm) package

DESCRIPTION

The OXCFU950 is a low-cost, synchronous 16-bit PC Card (PCMCIA) or CompactFlash/CF+ to UART and USB host controller. It combines the high performance of Oxford Semiconductor OX16C95x UART technology with USB host capabilities. The inclusion of both a UART and USB make the OXCFU950 ideal for wireless data cards.

The OXCFU950 integrates an OHCI-based USB host controller, which supports a single USB port via the integrated transceiver. An 8-Kbyte memory provides an efficient interface for buffer and control structures between the USB controller and the device driver.

The attribute memory, UART and USB registers can be programmed via the internal EEPROM during power up or hard/soft reset, allowing different card manufacturers to customize information contained in the attribute memory or UART/USB registers, for example PC Card ID value.

The 3.3-V technology has been specified to operate as low as 2.85 V to allow an in-line regulator to be used for mixed 3-V/5-V applications. CF/PC Card pins are 5-V tolerant and the variable I/O voltage to the UART/MIO pins eases the connection to application chipsets.

A number of power-down modes are included to minimize power consumption, including sleep modes for when a device function is not being used.

The OXCFU950 contains a single-channel ultra-high performance OX16C95x UART offering data rates up to 12 Mbps and 128-byte deep transmitter and receiver FIFOs. Deep FIFOs reduce CPU overhead and allow utilization of higher data rates.

The OXCFU950 USB controller supports OHCI-compliant host stack software (available from Oxford Semiconductor).

The OXCFU950 UART is software-compatible with the widely-used industry-standard 16C550 type devices and compatibles, as well as other OX16C95x family devices.

OXCFU950 DS-0023—February 07
Part Nos. OXCFU950-QFAG

The addition of software resets enable the device to recover from unforeseen error conditions, allowing drivers to restart gracefully.
The OXCFU950 includes four user I/O pins to enhance external device control. These I/O pins can also be configured as interrupt inputs.

For simplicity, the host interface is referred to as the CompactFlash interface throughout this specification. Unless otherwise stated, this should be taken to include PCMCIA 16-bit Card operation as well as CompactFlash/CF+.

CONTENTS

REVISION HISTORY

REV	DATE	REASON FOR CHANGE / SUMMARY OF CHANGE
Dec 2005	7 Dec 2005	First publication
May 2006	5 May 2006	Features revision
Feb 2007		Updates to reset values for UART Interrupt Enable register (address of 0x15) & Soft Reset and Clock Enable register (address of 0x18); reset text correction for ICR REV register; clarification of use of theOXCFU950 with USB hub devices

1 BLOCK DIAGRAM

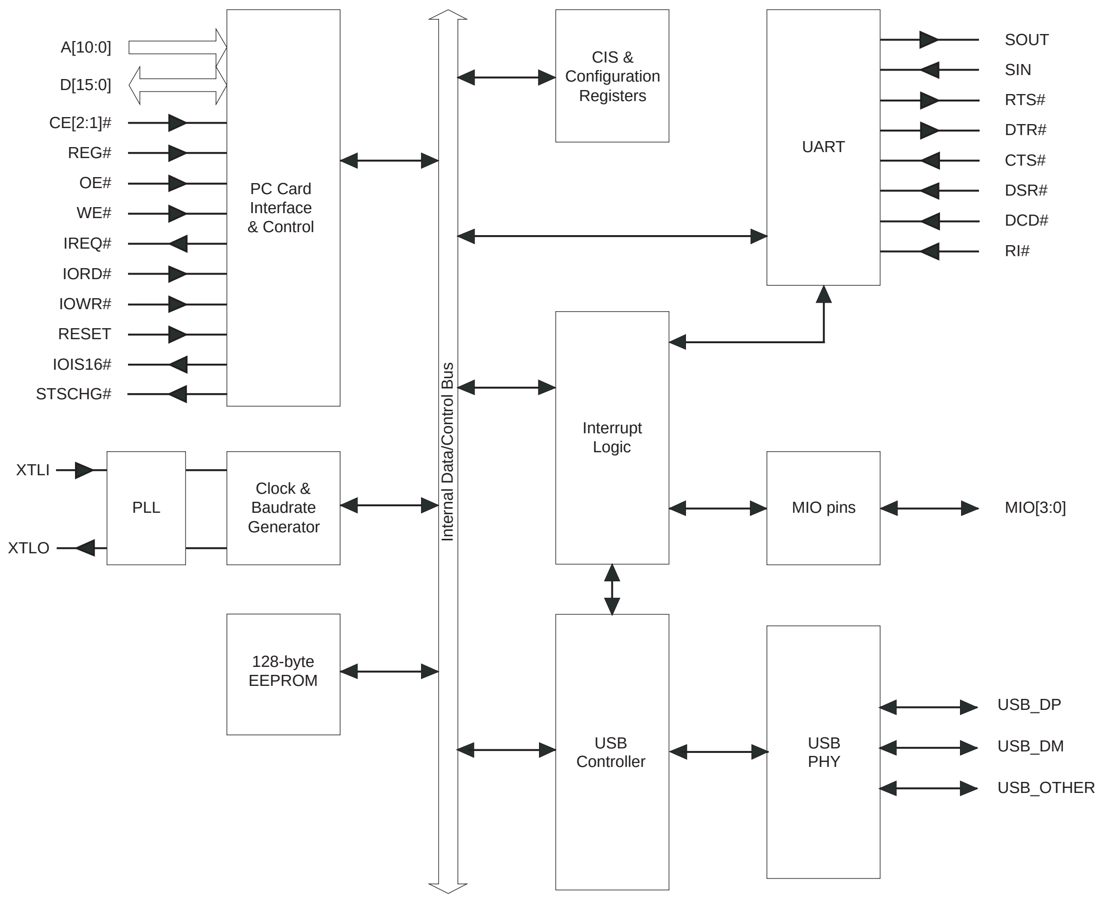


Figure 1: OXCFU950 Block Diagram

2 FUNCTIONAL OVERVIEW

2.1 Introduction

The OXCFU950 provides a multi-function CompactFlash interface to a USB 1.1 FS host controller and UART interface. Figure 1 shows the structure of the device. The OXCFU950 provides two functions: USB and UART; and three spaces: attribute memory, common memory and I/O space. It can also be configured, with software support, as a single-function CompactFlash card, supporting either USB-only or UART-only operation. Note: the hardware underpinning both functions is accessible via single-function mode, which would facilitate a single-function hybrid driver able to control both the USB and UART functions, if required.

2.2 Attribute Memory

The attribute memory space contains the card information structure (CIS) (see PCMCIA metaformat specification), and a set of function configuration registers (FCRs) (see PCMCIA electrical specification) for each function (USB and UART).

A CompactFlash host initially reads the CIS from the attribute memory to establish the features and requirements of the OXCFU950. The OXCFU950 provides a default hardwired version of the CIS and a user-programmable version of the CIS in EEPROM; see Section 5 for details. The CIS is read-only, but programming the internal EEPROM can change its contents. This allows the OXCFU950 to be used in many different applications and allows product differentiation.

Each function must be configured and enabled by the host by accessing the appropriate FCRs, which exist at a specific location in attribute memory. The CIS tells the host where the FCRs for each function can be found.

2.3 Common Memory

The USB host controller used in the OXCFU950 is based on the OHCI standard, which defines how the OS and USB host controller communicate. OHCI host controllers use an area of memory and a standard register set. Structures that describe packets that are to be transmitted or have been received are built up in memory, and are read or written by the USB host controller or operating system (OS).

Traditional OHCI host controllers use memory on the host system, which is not possible for a CompactFlash card, so the OXCFU950 provides 8 Kbytes of on-board RAM for this purpose, which can be accessed via the common memory space and is large enough to contain all the required structures.

2.4 I/O Space

The I/O space in the OXCFU950 contains three register sets that allow access to local configuration registers (LCRs), USB registers and UART registers. The registers become available when the functions are enabled (via the FCRs).

The LCRs provide access to the OXCFU950 non-USB and non-UART functions. They also provide a mechanism to start and stop clocks within the device and to generate a soft reset to the USB controller and UART controller.

The UART register set is compatible with the 950, but has been extended to allow higher performance with many host controllers.

The USB register set provides indirect 32-bit access to both an OHCI-compatible register set and the common memory space.

2.5 USB Host Controller

The USB host controller is based on an embedded OHCI model. It provides a single USB 2.0 full-speed (FS) port. A device driver is required on the host PC to access the port, because the OHCI registers must be accessed indirectly, due to limitations of the CompactFlash/CF+ and PCMCIA 16-bit PC Card standards.

It is possible to create a device driver that will allow a specific USB device to be accessed via the CompactFlash interface. This could, for example, be used in a product without a public USB interface, where a known USB peripheral is always present.

It is also possible to create a device driver that is integrated into the USB stack of the host operating system USB stack. This is ideal for a solution that includes a public USB interface, or for a solution that interfaces into a USB peripheral with existing drivers that could be used with the OS USB stack.

Note: The USB controller supports all OHCI endpoint types. However, support is not provided for USB hub devices.

2.6 UART Interface

The UART interface provides a 550- and 950-compatible register set, with all the usual 950 features. It has also been extended over the normal 950 register set to include a number of performance-enhancing features.

The UART is completely synchronous with a maximum baud rate of 12 Mbps. The UART clock can be switched off to save power, and it also includes an asynchronous wake-up mechanism.

2.7 EEPROM

The OXCFU950 contains an integral 8-Kbyte EEPROM, which is used to hold user-configurable CIS and LCR (local configuration register) data. The EEPROM can be configured using Oxford Semiconductor software utilities via a host machine or during board manufacture; see section 4.3. The functional use of the EEPROM is discussed in section 10.

2.8 Software Drivers

The OXCFU950 USB host controller is typically driven by the appropriate Oxford Semiconductor supplied driver. The UART interface can be driven either by an appropriate Oxford Semiconductor serial driver or by the default serial driver of the operating system. The advanced features of the UART interface are not available to default drivers. See section 11 for further information.

.

3 PIN INFORMATION

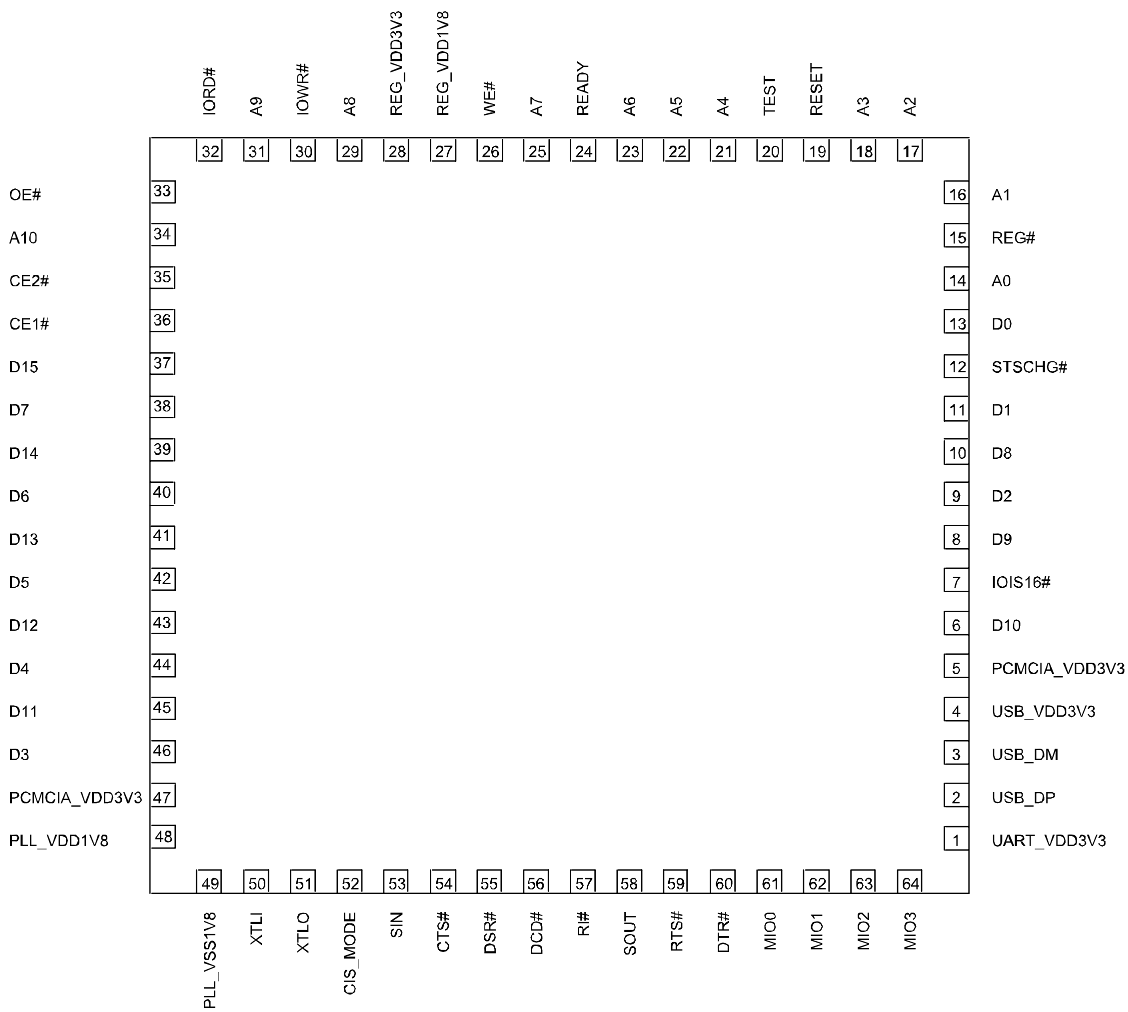


Figure 2: OXCFU950 Pin Information (QFN64)

4 PIN DESCRIPTIONS

Pin Number	Pad Type Note 1	Pin Name	Description
PC Card/Compact Flash Interface & Control Note 2			
14,16,17,18,21,22,23, 25,29,31, 34	I_C_33_5_N_	A[10:0]	PC Card/CompactFlash address bus, bits [10:0]
13,11,9,46,44,42,40,38 ,10,8,6,45,43,41,39,37	B_T_33_5_N_2_	D[15:0]	PC Card/CompactFlash data bidirectional bus.
15	I_C_33_5_N_	REG#	Register select and I/O enable.
35,36	I_C_33_5_N_	CE[2:1]#	Active-low card enable.
33	I_C_33_5_N_	OE#	Active-low output enable used to gate memory read data (attribute memory). Host must negate the OE# signal during write operations.
26	I_C_33_5_N_	WE#	Active-low write enable used for strobing memory write data (attribute memory).
32	I_C_33_5_N_	IORD#	Active-low I/O read enable.
30	I_C_33_5_N_	IOWR#	Active-low I/O write enable.
7	O_T_33_1	WP IOIS16#	Write protect (in memory-only mode). Data is 16-bit (in I/O & memory mode).
19	I_C_33_5_S_	RESET	PC Card/CompactFlash reset. **Note** : This pin requires an external reset pulse generator (RC network) to allow for the startup time for the internal oscillator and PLL.
24	O_T_33_2	READY IREQ#	Device ready (in memory-only mode). Active-low Interrupt request (in I/O & memory mode). Indicates to the host system that the PC Card/CF requires host software service. Note: interrupt can support level or pulsed types.
12	O_T_33_2	BVD1 STSCHG#	Battery voltage detect 1 (in memory-only mode). Not supported so held static. Active-low status-changed pin (in memory & I/O mode). Used to alert the host system that the card status has changed. In this case it means that a function's ready state has changed. The host should check this by considering each function's PRR.
5,28,47	P_33	PCMCIA_VDD3V3	Connected internally.
UART (MODEM) Function Note3			
58	O_T_33_1	SOUT IrDA_Out	UART serial data output. UART IrDA data output when MCR[6] set in enhanced mode.
53	I_C_33_5_N_	SIN IrDA_In	UART serial data input. UART IrDA data input when IrDA mode is enabled (see above).
56	I_C_33_5_N_	DCD#	Active-low modem data-carrier-detect input.
60	O_T_33_1	DTR# 485_En	Active-low modem data-terminal-ready output. If automated DTR# flow control is enabled, the DTR# pin is asserted & deasserted if the receiver FIFO reaches or falls below the programmed thresholds. In RS485 half-duplex mode, the DTR# pin may be programmed to reflect the state of the transmitter empty bit to control the direction of the RS485 transceiver buffer automatically (see register ACR[4:3]).
59	O_T_33_1	RTS#	Active–low modem request-to-send output. If automated RTS# flow control is enabled, the RTS# pin is de-asserted & reasserted when the receiver FIFO reaches or falls below the programmed thresholds.
54	I_C_33_5_N_	CTS#	Active-low modem clear-to-send input. If automated CTS# flow control is enabled, upon de-assertion of the CTS# pin the transmitter completes the current character & enters idle mode until the CTS# pin is reasserted. Note: flow control characters are transmitted regardless of the state of the CTS# pin.
55	I_C_33_5_N_	DSR#	Active-low modem data-set-ready input. If automated DSR# flow control

Pin Number	Pad Type[Note 1]	Pin Name	Description
			is enabled, upon de-assertion of the DSR# pin, the transmitter completes the current character & enters idle mode until the DSR# pin is reasserted. Note: flow control characters are transmitted regardless of the state of the DSR# pin.
57	I_C_33_5_N_	RI#	Active-low modem ring-indicator input.
1	P_33	UART_VDD3V3	Power supply to UART and MIO I/O interfaces. Voltage on these pins should be tied to 1.8 V to 3.3 V supply (depending on requirement).
Crystal Oscillator / PLL pins[Note4]			
51	A_18	XTLO	Crystal oscillator output.
50	A_18	XTLI	Crystal oscillator input. Frequency = 12 MHz. Note : a pull-down resistor of 500 KΩ is required on this pin.
49	P_18	PLL_VSS1V8	Ground (0 Volts) for PLL & oscillator cells. This pin should be tied to ground.
48	P_18	PLL_VDD1V8	Power supply for PLL & oscillator cells. This pin should be tied to 1.8 volts (i.e. tied to REG_VDD1V8 pin).
USB[Note 5]			
2	A_33	USB_DP	USB data plus.
3	A_33	USB_DM	USB data minus.
4	P_33	USB_VDD3V3	Power supply to USB I/O interface. This pin should be tied to 3.3 volts.
Multi-Purpose I/O[Note 3]			
64,63,62,61	B_T_33_5_N_1_	MIO[3:0]	Multipurpose IO pins. Note: if enabled, MIO[3:0] can be used as an interrupt inputs. MIO[3:2] can be configured to act as the USB power management control signals PORT_OVER_CURRENT & PORT_POWER
Miscellaneous Pins/Pads[Note2]			
52	I_C_33_5_N_U	CIS_MODE	Test mode select pin/default CIS select. (See Section 5.1)
20	I_C_33_5_N_D	TEST	Test pin. This pin should be tied to VSS for normal operation.
Additional Power and Ground			
65 (Thermal bonding pad)	P_00	VSS	Main digital ground pin. The VSS pin should be tied to ground. Note this is the thermal bonding pin underneath the 64-pin QFN package
27	P_18	REG_VDD1V8	Output supply from internal voltage regulator at 1.8 V. Can be used as power supply to PLL (i.e. pin #48).

Table 1: Pin Descriptions

Note 1 : Pad syntax description

Pad type	syntax
Digital Input pad	t_a_xy_h_i_p
Digital Output pad	t_a_xy_d
Digital Tristate output pad	t_a_xy_d
Digital Bidirectional pad	t_a_xy_h_i_d_p
Analogue pad	t_xy
Power Pad	t_xy

Table 2: Pad Type Syntax

symbol	description	values
t	Pad type	I—input pad O—output pad T—Tri-state output pad B—Bidirectional pad A—Analogue pad P—Power pad
a	Logic type	C—cmos T—TTL
xy	Operating voltage	33—3V3 18—1V8 00—0 V (i.e. VSS Ground)
h	Tolerant input voltage level	5—5 V N—same as operating voltage
i	Input type	N—normal S—Schmitt
d	Output drive strength	1—x1 drive strength 2—x2 drive strength 3—x3 drive strength
p	Pull up/down resistor	U—pull up resistor D—pull down resistor

Table 3: Pad Type Syntax Symbol Descriptions

Note 2 : PC Card/CompactFlash Power Domain Group
- This group of signals is in the PC Card/CompactFlash power domain group.
- Supply voltage range is specified from 2.85 V to 3.6 V.
- This is also the internally connected supply to the internal regulator. Hence pin #28 should be connected to the PC Card/CompactFlash supply source.

Note 3 : Modem Power Domain Group
- This group of I/O signals is in the MODEM power domain group.
- Supply voltage range is specified from 1.8 V to 3.6 V.

Note 4 : Analogue Power Domain Group
- This group of I/O signals is in the analogue power domain group (which supplies the oscillator and PLL).

Note 5 : USB Power Domain Group
- This group signals is in the USB power domain group.
- Supply voltage range is specified from 2.85 V to 3.6 V.

4.1 Power Supply Filtering

4.1.1 PLL Supply De-Coupling

The low frequency noise at the Vdd and Gnd supplies is automatically corrected by the PLL control loop. However, where noise frequency exceeds the loop bandwidth (60 to 100kHz) the noise is partially converted into clock jitter.
The conversion of supply noise into long-term clock jitter is important above 700 kHz. Oxford Semiconductor recommends you to filter Vdd in this range efficiently.
For full-speed USB applications, where the jitter must not exceed 0.5 ns peak for a time duration approximately 1.2 µs, Oxford Semiconductor recommends you to ensure that the noise level above 700kHz is lower than 7mV rms.

4.1.2 Internal Voltage Regulator Supply De-Coupling

Adequate output supply decoupling is mandatory to reduce ripple and avoid oscillations—use two capacitors in parallel on the REG_1V8 pin:

- One external 470-pF (or 1-nF) NPO capacitor must be connected between REG_VDD1V8 (pin #27) & VSS as close to the chip as possible
- One external 2.2-µF (or 3.3-µ F) X7R capacitor must be connected between REG_VDD1V8 (pin #27) & VSS as close to the chip as possible

Note : Z5u and Y5V capacitors must be avoided—the poor high-frequency behavior of these dielectrics makes them unsuitable for power supply decoupling.

Adequate input supply decoupling is mandatory to improve start-up stability and reduce source voltage drop_

- One external 100-pF NPO capacitor must be connected between REG_VDDV3V (pin #28) & VSS as close to the chip as possible
- One external 4.7-µ F X7R capacitor must be connected between REG_VDD3V3 (pin #28) & VSS as close to the chip as possible

4.2 Oscillator Specification

Code	Parameter	Condition	Min	Typ	Max	Unit
Ton	Startup Time	With crystal defined below			2	ms
Pon	Drive level				150	µW
ESR	Equivalent series resistance				80	Ohm
Cm	Motion capacitance		5		9	fF
Cshunt	Shunt capacitance				7	pF
Cload	Load Capacitance	Max external capacitors: 40 pF	15		20	pF

XTLI and XTLO shall not be used to drive other circuits.

The external capacitors on XTLI and XTLO must have the correct crystal load capacitance (max 40 pF). Additionally, a 500-KΩ pull-down resistor is required on XTLI.

4.3 I/O Pins for EEPROM Direct Access Programming Mode

The OXCFU950 allows a direct access mode to the internal EEPROM, for programming the EEPROM without using the PC Card Interface. This can be used by manufacturers to configure The EEPROM during board manufacture.

The following table shows the pins used for this EEPROM programming mode.

Pin Name	Direction	Pin Usage Description
TEST	Input	Test mode enable signal (Set to 'logic 1' – VDD3V3)
A6	Input	Static test signal (set to logic '1' – VDD3V3)
A5	Input	Static test signal (set to'logic '1' – VDD3V3)
A4	Input	Static test signal (set to logic '0' – VSS)
A8	Input	Serial data in (SDI)
A9	Input	Shift & load
READY	Output	Serial data out (SDO)
Z_IORD	Input	Test clock input
A7	Input	Write-enable signal to write in the memory (active-low)
Z_WE	Input	Reset signal
Z_IOWR	Input	EEPROM test-enable signal

Full details of the internal configuration of the EEPROM and details of programming utilities are available on request from Oxford Semiconductor.

5 CONFIGURATION & OPERATION

5.1 Mode Selection

The OXCFU950 has two modes of operation, as shown in the table below.

CIS_MODE pin	Operation
1	Normal mode
0	Generic mode

Table 4: Modes of Operation

5.1.1 Normal Mode

The chip usually operates in normal mode, in which the contents of the EEPROM are used to define the CIS and the action taken on a USB/UART reset. If no CIS is found in the EEPROM, the hard-wired CIS is used.

5.1.2 Generic Mode

Generic mode disables the EEPROM controller. It forces the OXCFU950 to use the hardwired CIS. Under normal circumstances it should not be used unless an image placed on the EEPROM provides an invalid CIS. In this event, it is unlikely that the OXCFU950 can be enumerated in normal mode, so the contents of the EEPROM can only be changed by using generic mode.

5.2 16-Bit PC Card & CompactFlash/CF+ Operation

PCMCIA 16-bit PC Card and CompactFlash/CF+ host systems allow for hot card insertion.

When a card is inserted into a host system, the host system configures it. The PCMCIA standard defines two card detect pins that allow the host to be notified when a card is inserted or removed.

By default the device powers up in either normal or generic mode, depending on the CIS_MODE pin.

The host system waits for the READY# signal to be active before reading the CIS in the attribute memory of the device. By reading this tuple information, the host system is able to identify the device type and the necessary resources requested by the device.

The host system then loads the device-driver software according to the tuple information and configures the I/O, memory and interrupt resources. After determining that the device is a memory and I/O type device, the host enables its I/O mode by writing to its configuration options register(s) in attribute memory space. Device drivers can then access the functions at the assigned addresses.

A set of LCRs is provided to control the device characteristics, such as interrupt handling, and report its internal functional status.

Two zones in the EEPROM can be used to redefine the reset values of all the registers in I/O space. One zone is executed when the USB function is reset; the other is executed when the UART is reset. In this way the USB and UART functions can be tailored to end-user requirements if the default values do not meet the specific requirements of the manufacturer.

It is good practice to make sure that the EEPROM is always used to provide reset values for the LCR registers. Because each function can access any of the LCR registers, using the EEPROM in this way gives ownership of each LCR to a specific function. This feature can also be used to pre-configure the USB or UART registers without modifying the device driver. This allows, for example, the enhanced features of the integrated UART to be in place prior to handing over to any generic device drivers. It is also possible to pre-configure and then lock four of the UART registers (CPR, TCR, DLL, DLM), so they cannot be modified by the device driver.

Device reprogramming can also be performed for the CIS area, for example to allow the manufacturer to modify the resources required or manufacturer ID values.

Note: a default set of tuples is provided for both multifunction and single-function solutions.

6 PC CARD & COMPACTFLASH/CF+ TARGET CONTROLLER

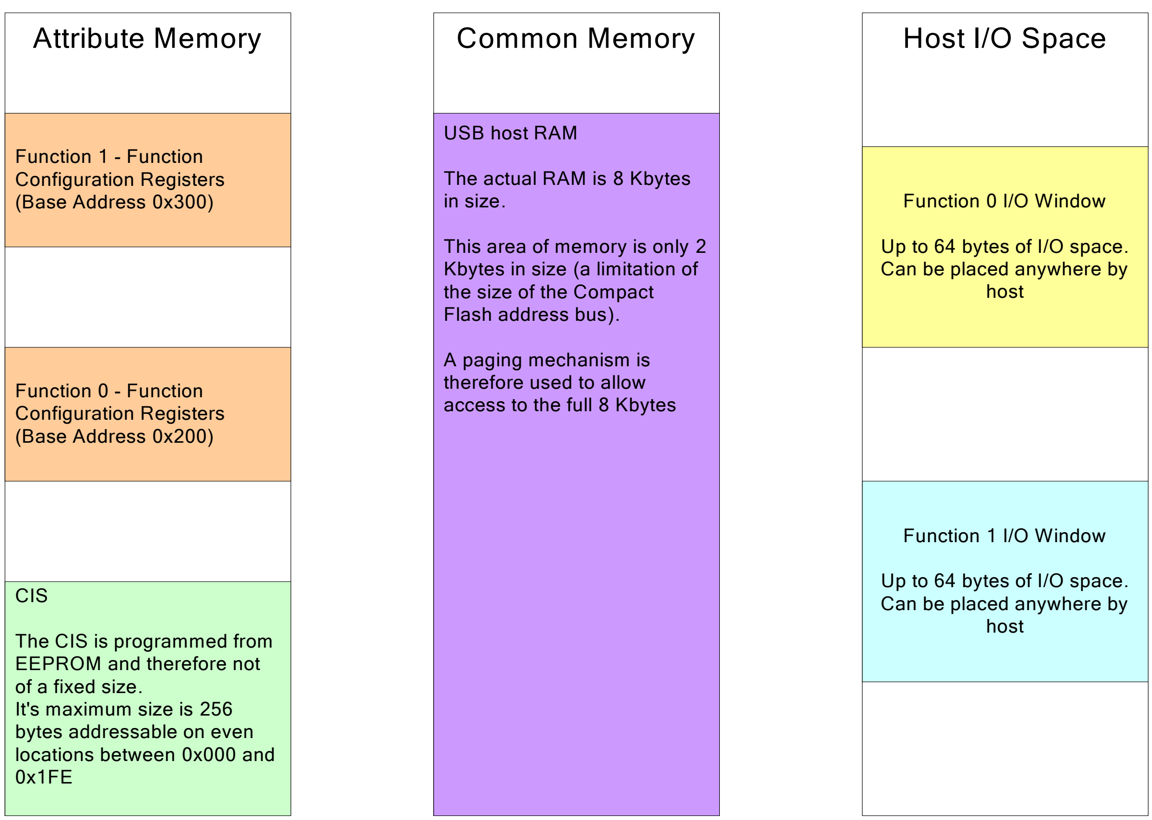
6.1 Operation

Note: See section 14 for timing waveforms.

The OXCFU950 responds to a number of different 16-bit PC Card/CompactFlash accesses (detailed below). Section 14 contains timing diagrams and information for each of these types of access:

- *Direct common memory read/writes*: These are required to gain access to the 8 Kbytes of RAM used by the OHCI USB host controller.
- *Direct attribute memory read/writes*: Access to direct attribute memory is required to allow the host access to both the CIS and FCRs (for each function). Valid data is 8 bits wide and on even bytes only, for direct attribute memory.
- *I/O read/writes*: I/O accesses are performed to access the USB host controller, UART and local configuration registers. Data width is restricted to 8 bits, as required by the standard UART function. While the device CIS information configures the card as a multifunction device, the host (or device driver/EEPROM controller) must set up a base address within I/O space for each function. As reads and writes are immediate, there is no requirement to hold the WAIT# signal in its active state, thus providing maximum speed access to I/O space.

6.2 OXCFU950 Memory Space Map



6.3 Function Access

6.3.1 I/O Access Windows

The OXCFU950 can support two functions. Each function has its own I/O access window and each window has a programmable base address and is up to 64 bytes long. The windows can therefore be located anywhere in the host I/O space, and there is no assumed relationship between their positions.

The base address for each window can be set either by using the I/O base address register (in attribute memory—see the PCMCIA electrical specification) or by using the alternative I/O base address register set in the I/O space. The register can be set up by the EEPROM on a reset.

6.4 I/O Space Register Map

Offset from function’s base address in I/O space	Function	Register
0x00 through to 0x07	UART	Standard UART registers
0x08	UART 32-bit FIFO access	
0x09		
0x0A		
0x0B		
0x0C	LCRs (EEPROM control)	EEPROM status & control register
0x0D	UART Extra Features	
0x0E		
0x0F		
0x10	LCRs	TEST register. I/O or memory mode function enable
0x11		Multipurpose I/O configuration register
0x12		IRQ test register
0x13		MIO interrupt enable register
0x14		USB interrupt enable register
0x15		UART interrupt enable register
0x16		Interrupt status register
0x17		Soft USB reset & clock enable register
0x18		Soft UART reset & clock enable register
0x19		Soft system reset & clock enable register
0x1A		EEPROM clock control register
0x1B		Alternative I/O base control register
0x1C		Alternative UART I/O base 0 register
0x1D		Alternative UART I/O base 1 register
0x1E		Alternative USB I/O base 0 register
0x1F		Alternative USB I/O base 1 register
0x20 through to 0x40	USB	USB host controller registers

6.5 I/O Space Registers

0x00 to 0x0B	UART
Notes	See Section 8 for details of the UART register map

0x0C	EEPROM Status and Control Register			
Notes	This 8-bit register provides bitmapped access to the control signals of the EEPROM. By manipulating these signals it is possible to program and read back the various locations of the EEPROM.			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7	e2_rdybsyn	-	R	X
6	e2_fn_wtn	-	RW	0
5	e2_fn_shandloadb	-	RW	0
4	e2_fn_sdi / e2_fn_sdo	-	RW	0
3	e2_fn_clock	-	RW	0
2	e2_fn_reset	-	RW	0
1	e2_fn_test	-	RW	0
0	e2_valid	-	R	X

0x0D to 0x0F	UART
Note	See Section 8 for details of the UART register map

0x10	Test			
Notes	This register provides 8 bits of read/write storage & can be used for test purposes.			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:0	Test register Bit 1—enables I/O memory mode for UART Bit 3—enables I/O memory mode for USB	W	R/W	10100101

0x11	Multi-Purpose I/O Configuration Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:6	MIO3 configuration register 00—MIO3 is a non-inverting input pin 01—MIO3 is an inverting input pin 10—MIO3 is an output pin driving '0' 11—MIO3 is an output pin driving '1'	W	R/W	00
5:4	MIO2 configuration register 00—MIO2 is a non-inverting input pin 01—MIO2 is an inverting input pin 10—MIO2 is an output pin driving '0' 11—MIO2 is an output pin driving '1'	W	R/W	00
3:2	MIO1 configuration register 00—MIO1 is a non-inverting input pin 01—MIO1 is an inverting input pin 10—MIO1 is an output pin driving '0' 11—MIO1 is an output pin driving '1'	W	R/W	00
1:0	MIO0 configuration register 00—MIO0 is a non-inverting input pin 01—MIO0 is an inverting input pin 10—MIO0 is an output pin driving '0' 11—MIO0 is an output pin driving '1'	W	R/W	00

0x12	IRQ Test register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:2	Reserved	-	R	000000
1	Function '1' IRQ (USB). When set forces USB IRQ to be generated.	W	R/W	0
0	Function '0' IRQ (UART). When set forces UART IRQ to be generated.	W	R/W	0

0x13	MIO Interrupt Enable Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:4	Reserved	-	R	0000
3	MIO[3] interrupt enable	W	R/W	0
2	MIO[2] interrupt enable	W	R/W	0
1	MIO[1] interrupt enable	W	R/W	0
0	MIO[0] interrupt enable	W	R/W	0

0x14	USB Interrupt Enable Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:2	Reserved	-	R	000000
1	USB fast IRQ enable	-	R/W	0
0	USB interrupt enable	W	R/W	0

0x15	UART Interrupt Enable Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:1	Reserved	-	R	0000000
0	UART host interrupt enable	W	R/W	1

0x16	Interrupt Status Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7	Reserved	-	R	0
6	Read: USB fast IRQ status Write: Setting this bit to '1' clears the USB fast IRQ.	-	R/W	0
5	This bit reflects the state of the USB interrupt	-	R	0
4	This bit reflects the state of the UART interrupt	-	R	0
3	MIO3. This bit reflects the state of the internal MIO[3]	-	R	X
2	MIO2. This bit reflects the state of the internal MIO[2]	-	R	X
1	MIO1. This bit reflects the state of the internal MIO[1]	-	R	X
0	MIO0. This bit reflects the state of the internal MIO[0]	-	R	X

0x17	Soft USB Reset & Clock Enable Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:6	Reserved	-	R	00
5	0—selects USB OVER_CURRENT signal is active-high input 1—selects USB OVER_CURRENT signal is active-low input	W	R/W	0
4	0—selects normal MIO3 functions 1—selects MIO3 as OVER_CURRENT (input) signal	W	R/W	0
3	0—selects USB PORT_POWER signal is active-high output 1—selects USB PORT_POWER signal is active-low output	W	R/W	0
2	0—selects normal MIO2 functions 1—selects MIO2 as USB PORT_POWER (output) signal	W	R/W	0
1	Active-high clock enable for USB host controller	W	R/W	0
0	Active-high soft reset for USB host controller	W	R/W	0

0x18	Soft UART Reset & Clock Enable Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:2	Reserved	-	R	000000
1	Active high clock enable for UART	W	R/W	1
0	Active high soft reset for UART	W	R/W	0

0x1A	EEPROM Clock Control Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:1	Reserved	-	R	0000000
0	EEPROM clock enable	W	R/W	1

0x1B	Alternate I/O Base Control Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:2	Reserved	-	R	000000
1	Select alternate I/O base address for USB	W	R/W	0
0	Select alternate I/O base address for UART	W	R/W	0

0x1C	Alternate UART I/O Base 0 Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:0	Alternate UART I/O base address bits 7:0	W	R/W	00000000

0x1D	Alternate UART I/O Base 1 Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:3	Alternate UART I/O base address bits 15:11	-	R	00000
2:0	Alternate UART I/O base address bits 10:8	W	R/W	000

0x1E	Alternate USB I/O Base 0 Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:0	Alternate USB I/O base address bits 7:0	W	R/W	00000000

0x1F	Alternate USB I/O Base 1 Register			
Bits	Description	Read/Write		Reset
		EEPROM	Host Access	
7:3	Alternate USB I/O base address bits 15:11	-	R	00000
2:0	Alternate USB I/O base address bits 10:8	W	R/W	000

0x20 to 0x40	USB Host Controller
Notes	See Section 7 for details of the USB host controller register map.

6.6 PC Card/CompactFlash Interrupt

6.6.1 Interrupt Generation

PCMCIA 16-bit PC Cards and CompactFlash/CF+ cards support pulse or level type interrupt signals to request interrupt service from the host system. The CIS of the card specifies whether pulse, level or both types of interrupt can be generated. When the host has read the CIS, it can set the LevlReq field in the Configuration Options Register (COR) for each function, to tell the card which type of interrupts should be generated.

The OXCFU950 can generate either type of interrupt. However, to reduce power consumption, the default CIS states that only level type interrupts can be generated. A custom CIS can be constructed to direct the card to generate pulse type interrupts if required, by using appropriate EEPROM values.

The OXCFU950 uses a clock divider circuit clocked at 12 MHz to generate pulse type interrupts. The generated pulse is guaranteed to be greater than the minimum length of 0.5 µs as defined in the PCMCIA standard.

6.6.2 Interrupt Sources

The OXCFU950 has six possible interrupt sources: the UART function, the USB host controller function and the four multi-purpose I/O (MIO) pins, which can be configured as interrupts using the MIO configuration register (MIO [3:0]) and the interrupt status and control register (ISR)

The OXCFU950 is a multi-function device with two functions, the UART function (function 0) and the USB host controller function (function 1). Interrupts generated by MIO [3:0] are mapped to UART function 0 so that any MIO enabled interrupt sets the Intr bit to ‘1’ in the UART CSR.

6.6.3 MIO Interrupts

To enable one or more MIO interrupts, the appropriate bit(s) in the MIO interrupt enable register (an LCR at offset 0x13) must be set to ‘1’ by the host system. Bit 0 corresponds to the MIO[0] interrupt enable, bit 1 corresponds to MIO[1] interrupt enable, bit 2 corresponds to MIO[2] interrupt enable and bit 3 corresponds to MIO[3] interrupt enable. When an MIO interrupt is asserted, the Intr field in the UART CSR is set to ‘1’

6.6.4 UART Interrupts

To enable the UART interrupt, set bit 0 (the UART interrupt enable bit in the UART interrupt enable register, which is an LCR register at offset 0x15) to ‘1’. When a UART interrupt is asserted, the Intr field within the UART CSR is set to ‘1’.

6.6.5 USB Interrupts

The OXCFU950 supports two forms of USB interrupt: the standard OHCI interrupt (standard IRQ) and a faster interrupt. Oxford Semiconductor USB host driver software uses the faster interrupt (fast IRQ) to improve system performance.,

USB Standard IRQ

To enable the USB interrupt, set bit 0 (the USB Host interrupt enable bit in the USB interrupt enable register, an LCR register at offset 0x14) to ‘1’. When a USB interrupt is asserted the Intr field within the USB CSR is set to ‘1’.

USB Fast IRQ

To enable the fast USB IRQ, set bit 1, the USB host fast enable bit in the USB interrupt enable register (an LCR register at offset 0x14) to ‘1’.

6.6.6 UART/USB Test Interrupts

There is a test facility for the host system to initiate UART and USB interrupts by writing to the UART/USB Interrupt Test Register at offset address 0x12. To force a function 0 UART interrupt, set bit 0 of the UART/USB Interrupt Test Register to ‘1’. To force a function 1 USB interrupt, set bit 1 of the UART/USB Interrupt Test Register to ‘1’.

6.6.7 Configuration & Status Register Interrupt Request/Acknowledge

Intr Field

The Intr field (bit 1 of the CSR) reports whether the function is requesting interrupt servicing and may be used to acknowledge that the host system is ready to process another interrupt request from the PC Card.

The function sets this field to ‘1’ to request interrupt service. ‘0’ signifies that it is not requesting interrupt service.

Writes to the Intr field in the CSR are ignored when the IntrAck field of both CSRs on the PC Card are reset to zero. If the host system writes a ‘0’ to this field in any CSR on the PC Card when the IntAck field of any CSR is set to ‘1’ and either function on the PC Card is requesting interrupt servicing, the PC Card must create an additional interrupt notification to the host system.

IntrAck Field

The interrupt acknowledge field (IntrAck, bit 0 of the CSR) changes the response characteristics of the Intr field. This field is used to enable a hardware/software protocol that permits the IREQ# signal to be shared by the two functions.

If the IntrAck bit for both function CSRs is set to ‘0’, writes to the Intr field are ignored. If the IntAck bit for either function is set to ‘1’:

- The host system must acknowledge it is ready to receive additional interrupts from the PC Card by writing ‘0’ to the Intr field of either function CSR after the host system has completed an entire interrupt processing cycle.
- The PC Card creates an additional interrupt notification to the host system when the host system writes a ‘0’ to the Intr bit to either function CSR and either function on the PC Card enabled for interrupt reporting and sharing is requesting interrupt service.

6.7 PC Card/CompactFlash Function Configuration Registers

Each function supplied by a 16-bit PC Card or CompactFlash/CF+ card must implement function configuration registers (FCRs). These registers allow the host to configure the function provided by the card, and are mapped into the attribute memory space at the location specified within the CONFIG tuple for that function in the CIS. The CONFIG tuple defines a base address for the FCRs and also shows which FCRs are supported by that function. The registers supported in the OXCFU950 are shown in the following table.

Offset from FCR base address	Attribute memory address (USB)	Attribute memory address (UART)	Register
0x00	0x300	0x200	Configuration options register
0x02	0x302	0x202	Configuration & status register
0x04	0x304	0x204	Pin replacement register
0x06	0x306	0x206	Socket & copy register
0x0A	0x30A	0x20A	I/O base address register 0
0x0C	0x30C	0x20C	I/O base address register 1

Table 5: Configuration Register Mapping

The definition of each configuration register is detailed below.

6.7.1 Configuration Options Register—COR (Offset 0x00)

The configuration options register is used to configure 16-bit PC Card and CompactFlash/CF+ cards that have programmable address decoders. When the card client driver has successfully parsed the CIS, it attempts to obtain system resources as requested by the CIS. On completion of this it assigns the resources to the card via the COR. The COR format and description is given in Table 6.

D7	D6	D5	D4	D3	D2	D1	D0
SRESET	LevIREQ	Function Configuration Index					

Field	Type	Description
SRESET	R/W	Software reset Setting this field to ‘1’ places the card in the reset state. This is equivalent to setting the RESET signal (on pin) except this SRESET field is not reset. Returning this field to ‘0’ leaves the card in the same un-configured, reset state as the card would be following a power-up and hard reset.
LevIREQ[1]	R/W	Level Mode IREQ# Setting this field to ‘1’ enables level type interrupts Setting this field to ‘0’ enables pulse type interrupts
Function Configuration Index	R/W	Configuration Index The host sets this field to the value of the Configuration Entry Number field of a Configuration Table Entry tuple as defined in the CIS. On setting the non-zero value in this field the function IO is enabled and IO accesses are allowed. When the field is set to zero (e.g. after a hard reset) the card will be configured to memory only mode and all IO accesses will be ignored by the card.

Table 6: Configuration Option Register

Note 1

The default tuples in the CIS tell the host that only level type interrupts are supported, to allow lowest power consumption. The OXCFU950 supports both level and pulse type interrupts, and if a particular manufacturer requires to use pulse type, or both, then the CIS can be modified using the internal EEPROM.

6.7.2 Configuration Status Register—CSR (Offset 0x02)

The configuration and status register is an optional register supported by the OXCFU950. The register allows additional control over function configuration and reports the configuration status.

D7	D6	D5	D4	D3	D2	D1	D0
Changed	SigChg	IOIs8	RFU	Audio	PwrDwn	Intr	IntrAck

Field	Type	Description
Changed	R	If a 16-bit PC Card or CompactFlash/CF+ card is using the I/O interface, the pin replacement register for the function is present & one or more of the state change signals in the pin replacement register are set to 1, or one of the Event bits in the extended status register is 1 and the corresponding Enable bit is 1, the function sets this field to 1. If a 16-bit PC Card or CompactFlash/CF+ card is not using the I/O interface or the pin replacement register for the function is not present, this field is undefined and should be ignored.
SigChg	R/W	This field serves as a gate for STSCHG# If a 16-bit PC Card or CompactFlash/CF+ card is using the I/O interface & both the Changed and SigChg fields are set to 1, the function asserts STSCHG#. If a 16-bit PC Card or CompactFlash/CF+ card is using the I/O interface and this field is reset to 0, the function does not assert STSCHG#. If a 16-bit PC Card or CompactFlash/CF+ card is not using the I/O interface or the pin replacement register for the function is not present, this field is undefined and should be ignored.
IOIs8	R/W	The host sets this field to 1 when it can provide I/O cycles only with an 8-bit D[7..0] data path. The card is guaranteed that accesses to the 16-bit registers will occur as two byte accesses rather than as a single 16-bit access.
RFU	-	Reserved. Must be 0.
Audio[1]	R/W	This bit is set to 1 to enable audio information on SPKR# when the card is configured. Not used: 0
PwdDwn[2]	R/W	When the host sets this field to 1, the function enters a power-down state, if such a state exists. Not used: 0 If a 16-bit PC Card or CompactFlash/CF+ card function does not have a power-down state, the function ignores this field.
Intr	R	Interrupt request/acknowledge – This field reports whether the function is requesting interrupt servicing and may be used to acknowledge the host system is ready to process another interrupt request from the 16-bit PC Card or CompactFlash/CF+ card. The function shall set this field to 1 when it is requesting interrupt service. The function sets this field to 0 when it is not requesting interrupt service. Writes to this field are ignored when the IntrAck field of all configuration and status registers on the 16-bit PC Card or CompactFlash/CF+ card are reset to 0.
IntrAck	R/W	Single function cards ignore this field on writes and always return 0.

Table 7: Configuration Status Register

Note 1
Audio is not supported on the device.

Note 2
The OXCFU950 does not support a specific power-down mode, because it is a low power device that features a number of sleep modes (see Section 6.7 for further details).

6.7.3 Pin Replacement Register—PRR (Offset 0x04)

The pin replacement register is implemented to provide information about READY, WP or the BVD[2..1] status when implementing the I/O interface.

D7	D6	D5	D4	D3	D2	D1	D0
CBVD1	CBVD2	CREADY	CWProt	RBVD1	RBVD2	RREADY	RWProt

Field	Description
CBVD1	This bit is set (1) when the corresponding bit, RBVD1, changes state. This bit may also be written by the host. Not used: 0
CBVD2	This bit is set (1) when the corresponding bit, RBVD2, changes state. This bit may also be written by the host. Not used: 0
CREADY	This bit is set (1) when the corresponding bit, RREADY, changes state. This bit may also be written by the host.
CWProt	This bit is set (1) when the corresponding bit, RWProt, changes state. This bit may also be written by the host. Not used: 0
RBVD1	When read, this bit represents the internal state of the Battery Voltage Detect circuits on the BVD1 pin. Not used: 1 When this bit is written as 1 the corresponding CBVD1 bit is also written. When this bit is written as 0, the CBVD1 bit is unaffected.
RBVD2	When read, this bit represents the internal state of the Battery Voltage Detect circuits on the BVD2 pin. Not used: 1 When this bit is written as 1 the corresponding CBVD2 bit is also written. When this bit is written as 0, the CBVD2 bit is unaffected.
RREADY	When read, this bit represents the internal state of the READY signal. This bit may also be used to determine the state of READY as that pin has been relocated for use as Interrupt Request on IO Cards. When this bit is written as 1 the corresponding changed bit is also written. When this bit is written as 0, the changed bit is unaffected.
RWProt	When read, this bit represents the state of the WP signal. This signal may also be used to determine the state of the Write Protect switch when pin 33 is being used for IOIS16#. Not used: 0 When this bit is written as 1 the corresponding changed bit is also written. When this bit is written as 0, the changed bit is unaffected.

Table 8: Pin Replacement Register

6.7.4 Socket & Copy Register—SCR (Offset 0x06)

This is an optional read/write register implemented by the OXCFU950, which the 16-bit PC Card or CompactFlash/CF+ card may use to distinguish between similar cards installed in a system. This register is always written by the system before writing the card Function Configuration Index field in the COR.

D7	D6	D5	D4	D3	D2	D1	D0
Reserved	Copy Number			Socket Number			

Field	Description
Reserved	This bit is reserved for future standardization. This bit must be set to zero (0) by software when the register is written.
Copy Number	16-bit PC Card and CompactFlash/CF+ cards that indicate in their CIS that they support more than one copy of identically configured cards, should have a copy number (0 to MAX twin cards, MAX = n-1) written back to the SCR. This field indicates to the card that it is the n'th copy of the card installed in the system, which is identically configured. The first card installed receives the value 0. This permits identical cards designed to share a common set of I/O ports while remaining uniquely identifiable and consecutively ordered.
Socket Number	This field indicates to the 16-bit PC Card and CompactFlash/CF+ card that it is located in the n'th socket. The first socket is numbered 0. This permits any cards designed to do so to share a common set of I/O ports while remaining uniquely identifiable.

Table 9: Socket and Copy Register

6.7.5 I/O Base Address Register—IOBA (Offset 0x0A & 0x0C)

This register holds the I/O base address that the host PC has assigned this function.

D7	D6	D5	D4	D3	D2	D1	D0
IO Base Address (7:0)							

Field	Description
IO Base Address (7:0)	Bits 7:0 of I/O base address for function.

Table 10: IOBA 0 (Offset 0x0A)

D7	D6	D5	D4	D3	D2	D1	D0
Reserved					Bits 10:8 of I/O base address		

Field	Description
IO Base Address (7:0)	Bits 7:0 of I/O base address for function.

Table 11: IOBA 1 (Offset 0x0C)

6.8 Card Information Structure

The card information structure (CIS) in the OXCFU950 is normally sourced from the integral EEPROM, and the OXCFU950 is usually shipped with a standard pre-programmed CIS image, which is documented in Section 6.8.2. When the EEPROM image is invalid, or when the CIS_Mode pin is asserted, the CIS defaults to a fixed default CIS image, which is documented in Section 6.8.1. Both the default CIS Image and the standard pre-programmed CIS image support multi-function operation.

6.8.1 Fixed Default CIS Image

Value (Hex)	Tuple Name	Description
Direct Attribute (Offset 0x0000)		
01 03 62 01 FF	CISTPL_DEVICE	Common Memory: 1) SRAM 2) 200 ns 3) 1x2048 bytes
17 05 12 00 D2 00 FF	CISTPL_DEVICE_A	Attribute Memory: 1) ROM 2) 200 ns 3) 1x512 bytes (CIS) 4) 1x512 bytes (Registers)
1C 04 03 62 01 FF	CISTPL_DEVICE_OC	Common Memory at 3.3 volts
1D 06 03 12 00 D2 00 FF	CISTPL_DEVICE_OA	Attribute Memory at 3.3 volts

Value (Hex)	Tuple Name	Description
06 0B 02 00 50 00 00 00 00 A0 00 00 00	CISTPL_LONGLINK_MFC	Two functions Function 0 at offset 0x00A0 Function 1 at offset 0x140
20 04 79 02 2A 95	CISTPL_MANFID	Manufacturer ID = 0x0279 Product ID = 0x952A
15 15 06 01 43 46 20 43 41 52 44 00 47 45 4E 45 52 49 43 00 00 00 FF	CISTPL_VERS_1	PC Card Standard 6.1 Manufacture Name = “CF CARD” Product Name = “GENERIC”
FF	CISTPL_END	
Direct Attribute (Offset 0x00A0)		
13 03 43 49 53	CISTPL_LINKTARGET	Contains ASCII text “CIS” so host knows that this is the start of a new valid chain
21 02 02 01	CISTPL_FUNCID	Serial Port
22 04 00 02 0F 7F	CISTPL_FUNCE	

Value (Hex)	Tuple Name	Description
1A 05 01 07 00 02 6F	CISTPL_CONFIG	Last CFTABLE_ENTRY has index of 7 Base address of FCRs is 0x0200
1B 0B C7 41 99 01 B5 1E 24 B0 FF FF 07	CISTPL_CFTABLE_ENTRY	Index 7 3.3 volts Any IRQ 16 bytes of I/O space.
FF	CISTPL_END	
Direct Attribute (Offset 0x0140)		
13 03 43 49 53	CISTPL_LINKTARGET	Contains ASCII text “CIS” so host knows that this is the start of a new valid chain
21 02 0B 00	CISTPL_FUNCID	OHCI USB Controller
22 05 02 00 02 0C 00	CISTPL_FUNCE	
1A 05 01 07 00 03 6F	CISTPL_CONFIG	Last CFTABLE_ENTRY has index of 7 Base address of FCRs is 0x0300
1B 0D C7 41 B9 01 B5 1E 26 B0 FF FF 08 00 07	CISTPl_CFTABLE_ENTRY	Index 7 3.3 volts Any IRQ 64 bytes of I/O space
FF	CISTPL_END	

6.8.2 Standard Pre-Programmed CIS Image (cfu950_multi_level_level_3v3_gold)

Value (Hex)	Tuple Name	Description
Direct Attribute (Offset 0x0000)		
01 03 62 01 FF	CISTPL_DEVICE	Common Memory: 4) SRAM 5) 200 ns 6) 1x2048 bytes
17 05 12 00 D2 00 FF	CISTPL_DEVICE_A	Attribute Memory: 5) ROM 6) 200 ns 7) 1x512 bytes (CIS) 8) 1x512 bytes (Registers)
1C 04 03 62 01 FF	CISTPL_DEVICE_OC	Common Memory at 3.3 volts
1D 06 03 12 00 D2 00 FF	CISTPL_DEVICE_OA	Attribute Memory at 3.3 volts
20 04 79 02 2B 95	CISTPL_MANFID	Manufacturer ID = 0x0279 Product ID = 0x952B
15 15 08 00 43 46 20 43 41 52 44 00 47 45 4E 45 52 49 43 00 00 00 FF	CISTPL_VERS_1	PC Card Standard 8.0 Manufacture Name = “CF CARD” Product Name = “GENERIC”

06 0B 02 00 50 00 00 00 00 A0 00 00 00	CISTPL_LONGLINK_MFC	Two functions. Function 0 at offset 0x00A0 Function 1 at offset 0x140
FF	CISTPL_END	
Direct Attribute (Offset 0x00A0)		
13 03 43 49 53	CISTPL_LINKTARGET	Contains ASCII text “CIS” so host knows that this is the start of a new valid chain
21 02 02 00	CISTPL_FUNCID	Serial Port
22 04 00 02 0F 7F	CISTPL_FUNCE	
1A 05 01 07 00 02 6F	CISTPL_CONFIG	Last CFTABLE_ENTRY has index of 7 Base address of FCRs is 0x0200
1B 0D C7 41 99 01 B5 1E A4 40 0F B0 FF FF 07	CISTPL_CFTABLE_ENTRY	Index 7 3.3 volts Any IRQ 16 bytes of I/O space
FF	CISTPL_END	
Direct Attribute (Offset 0x0140)		
13 03 43 49 53	CISTPL_LINKTARGET	Contains ASCII text “CIS” so host knows that this is the start of a new valid chain
21 02 0B 00	CISTPL_FUNCID	OHCI USB Controller

22 05 02 00 02 0C 00	CISTPL_FUNCE	
1A 05 01 07 00 03 6F	CISTPL_CONFIG	Last CFTABLE_ENTRY has index of 7 Base address of FCRs is 0x0300
1B 0F C7 41 B9 01 B5 1E A6 40 3F B0 FF FF 08 00 07	CISTPl_CFTABLE_ENTRY	Index 7 3.3 volts Any IRQ 64 bytes of I/O space
FF	CISTPL_END	

6.8.3 *Example USB Single Function CIS Image* (cfu950_usb_level_3v3_gold)

The table below shows an example CIS image suitable for single-function USB operation at 3.3 V.

Value (Hex)	Tuple Name	Description
Direct Attribute (Offset 0x0000)		
01 03 62 01 FF	CISTPL_DEVICE	Common Memory: 7) SRAM 8) 200 ns 9) 1x2048 bytes
17 05 12 00 D2 00 FF	CISTPL_DEVICE_A	Attribute Memory: 9) ROM 10) 200 ns 11) 1x512 bytes (CIS) 12) 1x512 bytes (Registers)
1C 04 03 62 01 FF	CISTPL_DEVICE_OC	Common Memory at 3.3 volts

1D 06 03 12 00 D2 00 FF	CISTPL_DEVICE_OA	Attribute Memory at 3.3 volts
20 04 79 02 2B 95	CISTPL_MANFID	Manufacturer ID = 0x0279 Product ID = 0x952B
21 02 FE 00	CISTPL_FUNCID	
15 15 08 00 43 46 20 43 41 52 44 00 47 45 4E 45 52 49 43 00 00 00 FF	CISTPL_VERS_1	PC Card Standard 8.0 Manufacture Name = “CF CARD” Product Name = “GENERIC”
1A 05 01 07 00 03 6F	CISTPL_CONFIG	Last CFTABLE_ENTRY has index of 7 Base address of FCRs is 0x0300

1B 0F C7 41 B9 01 B5 1E A6 40 3F B0 FF FF 08 00 07	CISTPl_CFTABLE_ENTRY	Index 7 3.3 volts Any IRQ 64 bytes of I/O space
FF	CISTPL_END	

6.8.4 Example UART Single Function CIS Image (cfu950_usb_pulse_5v_gold)

The table below shows an example CIS image suitable for single-function UART operation at 5 V.

Value (Hex)	Tuple Name	Description
Direct Attribute (Offset 0x0000)		
01 03 62 01 FF	CISTPL_DEVICE	Common Memory: 10) SRAM 11) 200 ns 12) 1x2048 bytes
17 05 12 00 D2 00 FF	CISTPL_DEVICE_A	Attribute Memory: 13) ROM 14) 200 ns 15) 1x512 bytes (CIS) 16) 1x512 bytes (Registers)
1C 04 03 62 01 FF	CISTPL_DEVICE_OC	Common Memory at 3.3 volts
1D 06 03 12 00 D2 00 FF	CISTPL_DEVICE_OA	Attribute Memory at 3.3 volts
20 04 79 02 2B 95	CISTPL_MANFID	Manufacturer ID = 0x0279 Product ID = 0x952B

15 15 08 00 43 46 20 43 41 52 44 00 47 45 4E 45 52 49 43 00 00 00 FF	CISTPL_VERS_1	PC Card Standard 8.0 Manufacture Name = "CF CARD" Product Name = "GENERIC"
21 02 02 00	CISTPL_FUNCID	Serial Port
22 04 00 02 0F 7F	CISTPL_FUNCE	
1A 05 01 07 00 02 6F	CISTPL_CONFIG	Last CFTABLE_ENTRY has index of 7 Base address of FCRs is 0x0200
1B 0C C7 41 99 01 55 A4 40 0F D0 FF FF 07	CISTPL_CFTABLE_ENTRY	Index 7 5.0 volts Any IRQ 16 bytes of I/O space Pulsed interrupts
FF	CISTPL_END	

7 USB HOST CONTROLLER

This datasheet assumes the reader has access to the public domain OHCI specification 1.0A, which is freely downloadable from the internet. It also assumes the reader has access to the USB Specification 2.0.
Both documents can be downloaded from http://www.usb.org/developers/docs/

The OXCFU950 includes an embedded OHCI USB Host controller. The OHCI specification defines a standard way for a host system (e.g. a PCMCIA 16-bit PC Card host, such as a desktop or Laptop PC) to control a USB host. The OHCI specification defines a number of 32-bit registers and a data structure that exists in memory, to which both the host system and the USB host controller have access.

The OXCFU950 CF interface does not support DMA-to-host memory, so an area of memory (the USB memory) is included on-chip and made available to the host system via the CompactFlash interface interface.

The register set is mainly used during the setup of the USB host controller, and is not used a great deal during normal operation. The memory is accessed by both the OHCI controller and the CompactFlash interface host throughout USB operation.

The OHCI specification assumes that both the memory and registers can be accessed via 32-bit atomic operations. This is not possible via the CompactFlash/CF+ or PCMCIA 16-bit PC Card interfaces, because only 8-bit and 16-bit operations are available. This could lead to problems, because the USB host controller assumes that the host PC uses 32-bit atomic operations to update the registers and the memory, when the host PC is making two 16-bit accesses or 4 8-bit accesses. Therefore, for a short time the 32-bit value consists of a number of bits of old data and a number of bits of new data—in effect, invalid data is potentially present in the USB registers.

To avoid this situation, an indirect access mechanism is used to allow the host PC to make atomic 32-bit accesses to the memory and USB register set. It allows the host PC to enter all details of the read or write operation, following which the device performs the operation in a single step.

However, USB memory is also mapped into common memory space to allow the host PC to perform DMAs into the common memory area when there is no risk of the USB host controller reading invalid data.

The indirect mechanism is also used to access the USB OHCI register set.

For details of the operation of OHCI compatible USB hosts, refer to the OHCI standard.

The USB host controller can be accessed via I/O space registers at offset 0x20 through 0x40, as shown below.

Offset from I/O base address (byte-based)	Name	Description
0x20	Register Address	Address of USB Register to be indirectly read/written. (Quadlet address, i.e. 1, 2, 3 etc, not 0, 4, 8…)
0x23	Register Control	7: if '1' then USB host controller is out of reset and ready for operation. 7: if '0' then USB host controller is in reset and not yet ready for operation. 6: reserved 5: reserved 4: reserved 3: reserved 2: reserved 1: if '1' then last write operation has been completed. 0: if '1' then it is safe to read contents of Register Read Data 0-3"
0x24	Register Write Data 0	Bits 7 down to 0 of 32-bit data to be written to the USB registers.
0x25	Register Write Data 1	Bits 15 down to 8 of 32-bit data to be written to the USB registers.
0x26	Register Write Data 2	Bits 23 down to 16 of 32-bit data to be written to the USB registers.
0x27	Register Write Data 3	Bits 31 down to 24 of 32-bit data to be written to the USB registers.
0x28	Register Read Data 0	Bits 7 down to 0 of 32-bit data to be read from the USB registers.
0x29	Register Read Data 1	Bits 15 down to 8 of 32-bit data to be read from the USB registers.
0x2A	Register Read Data 2	Bits 23 down to 16 of 32-bit data to be read from the USB registers.
0x2B	Register Read Data 3	Bits 31 down to 24 of 32-bit data to be read from the USB registers.
0x30	Memory Address 0	Address of memory location to be indirectly read/written (bits 7 down to 0).
0x31	Memory Address 1	Address of memory location to be indirectly read/written (bits 15 down to 8).
0x33	Memory Control	7: Reserved 6: Reserved 5: Reserved 4: Reserved 3: Reserved 2: Reserved 1: if '1' then last write operation has been completed. 0: if '1' then it is safe to read contents of Memory Read Data 0-3.
0x34	Memory Write Data 0	Bits 7 down to 0 of 32-bit data to be written to the USB memory.
0x35	Memory Write Data 1	Bits 15 down to 8 of 32-bit data to be written to the USB memory.
0x36	Memory Write Data 2	Bits 23 down to 16 of 32-bit data to be written to the USB memory.
0x37	Memory Write Data 3	Bits 31 down to 24 of 32-bit data to be written to the USB memory.
0x38	Memory Read Data 0	Bits 7 down to 0 of 32-bit data to be read from the USB memory.
0x39	Memory Read Data 1	Bits 15 down to 8 of 32-bit data to be read from the USB memory.
0x3A	Memory Read Data 2	Bits 23 down to 16 of 32-bit data to be read from the USB memory.
0x3B	Memory Read Data 3	Bits 31 down to 24 of 32-bit data to be read from the USB memory.
0x3C	Memory Page Select	1:0 = "11" Page 3 (locations 6144-8191) selected. 1:0 = "10" Page 2 (locations 4096-6143) selected. 1:0 = "01" Page 1 (locations 2048-4095) selected. 1:0 = "00" Page 0 (locations 0-2047) selected.
0x3D	USB Coverage Reg 1	Bits 7 to 1 are readable and, if set, give the following information: 7: Common memory read has occurred since last reset. 6: Common memory write has occurred since last reset. 5: IAR RAM read has occurred since last reset. 4: HCI read has occurred since last reset. 3: HCI write (with at least one byte lane disabled) has occurred since last reset. 2: IAR RAM Write has occurred since last reset. 1: HCI write (no byte lanes disabled) has occurred since last reset. Bit 0 must be set to enable the USB coverage registers.

0x3E	USB Coverage Reg 2	Bits 7 to 0 are readable and, if set, show that the memory controller has been interrupted by a common memory read, while in the following states: 7: HCI_READ_RESULT 6: HCI_READ 5: CM_WRITE 4: CM_WRITE_PAUSE 3: HCI_WRITE 2: HCI_WRITE_PAUSE 1: CM_READ 0: IDLE
0x3F	USB Coverage Reg 3	Bits 1 to 0 are readable and, if set, show that the memory controller has been interrupted by a common memory read, while in the following states: 1: IAR_READ_RESULT 0: IAR_READ

Table 25: USB I/O Space Description

7.1 Writing to an OHCI Register in the Embedded USB Host Controller

The host must use the following procedure to write to a USB register:

- Write the register offset into Register Address (0x20)
- Write bits 7:0 of the 32 bit data value into Register Write Data 0 (0x24).
- Write bits 15:8 of the 32 bit data value into Register Write Data 1 (0x25).
- Write bits 23:16 of the 32 bit data value into Register Write Data 2 (0x26).
- Write bits 31: 24 of the 32 bit data value into Register Write Data 3 (0x27).

The 32-bit data value in Register Write Data 0-3 is written to the USB core in a single atomic transaction when a write to Register Write Data 3 occurs.

On writing to Register Write Data 3, bit 1 of Register Control is cleared and set to 1 when the write is complete. This allows software to wait until the write has completed before continuing. This may be of use in checking that the USB host controller clock is running. Software is not required to check that this has happened.

7.2 Reading from an OHCI Register in the Embedded USB Host Controller

The host must use the following procedure to read from a USB register:

- Write the register offset into Register Address (0x20)

On writing to Register Address, bit 0 of Register Control is cleared and set to 1 when the read data is available for reading from Register Read Data 0-3.

- Poll Register Control (0x23) & wait until bit 0 is set
- Read bits 7:0 of the 32 bit data value from Register Read Data 0 (0x28).
- Read bits 15:8 of the 32 bit data value from Register Read Data 1 (0x29).
- Read bits 23:16 of the 32 bit data value from Register Read Data 2 (0x2A).
- Read bits 31:24 of the 32 bit data value from Register Read Data 3 (0x2B).

The data read from the USB core is returned in the Register Read Data registers. The read is initiated every time a byte is written to Register Address.

7.3 Writing to the USB Memory

The host must use the following procedure to write to the USB memory:

- Write the register offset into Memory Address (0x30 and 0x31)
- Write bits 7:0 of the 32 bit data value into Memory Write Data 0 (0x34).
- Write bits 15:8 of the 32 bit data value into Memory Write Data 1 (0x35).
- Write bits 23:16 of the 32 bit data value into Memory Write Data 2 (0x36).
- Write bits 31:24 of the 32 bit data value into Memory Write Data 3 (0x37).

The 32-bit data value in Memory Write Data 0-3 is written to the USB memory in a single atomic transaction when a write to Memory Write Data 3 occurs.

On writing to Memory Write Data 3, bit 1 of Memory Control is cleared and set to 1 when the write is complete, allowing software to wait until the write is complete before continuing. This may be of use in checking that the USB memory clock is running. Software is not required to check that this has happened.

7.4 Reading from USB Memory

The host must use the following procedure to read from USB memory:

- Write the memory address into Memory Address 0 and Memory Address 1.

  On writing to Memory Address 1, bit 0 of Memory Control is cleared and set to 1 when the read data is available for reading from Memory Read Data 0-3.

- Poll Memory Control (0x33) & wait until bit 0 is set.
- Read bits 7:0 of the 32 bit data value from Memory Read Data 0 (0x38).
- Read bits 15:8 of the 32 bit data value from Memory Read Data 1 (0x39).
- Read bits 23:16 of the 32 bit data value from Memory Read Data 2 (0x3A).
- Read bits 31:24 of the 32 bit data value from Memory Read Data 3 (0x3B).

The data read from USB memory is returned in Memory Read Data 0-3. The read is initiated every time Memory Address 1 is written.

7.5 Accessing USB Memory via Common Memory Access

Due to the limited number of address bits available in the compact flash form factor (11 address bits) a paging mechanism must be used if direct access to the USB Memory (via common memory space) is required. It should be used as follows.
To access a portion of the USB memory via the common memory space the host system should follow this procedure:

- Select the portion of USB memory via the Memory Page Select (0x40) register.
- Access the page of memory via common memory space (location 0x0000–0x07FF).

7.6 USB Host Controller Overview

The OXCFU950 USB module uses an OHCI-compliant USB host controller. This is augmented by a 2-Kbyte x 32-bit RAM to which both the OHCI controller and the host machine have access. The host access can be direct via common memory space or indirect via I/O space.

A USB RAM controller arbitrates between all three sources of RAM read and writes. The priority of the accesses is shown below (most important first):

1) CF host read—has to be highest priority because there are strict requirements of the CF bus to be met. A CF host read access interrupts and pauses any other accesses. All other accesses can be held off, and so do not interrupt a lower priority access.
2) OHCI read/write—the USB host controller should generally be given priority.
3) CF write access—because this can be queued until the OHCI read/write has been completed.
4) Indirect I/O access—lowest priority because this can easily be held off, and doing so should not adversely affect performance.

The RAM controller is responsible for this arbitration.

The HCI bus controller is responsible for communicating with the UHOSTC. Local bus indirect access can be generated either to the UHOSTC register set or to the RAM. The HCI bus controller must direct these accesses appropriately.

Note: The USB controller supports all OHCI endpoint types. However, support is not provided for USB hub devices.

8 UART INTERFACE

The internal UART in the OXCFU950 is based on that used in the well-proven OX16C95*x* UARTs, but with a number of significant enhancements. It is referred to throughout this document as the OXCFU950 UART core, or simply UART.

8.1 Mode Selection

The OXCFU950 UART is software-compatible with the 16C450, 16C550, 16C654 and 16C750 UARTs. The operation of the 950 depends on a number of modes which are referenced throughout this data sheet. The FIFO depth and compatibility modes are tabulated below:

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

Table 12: UART Mode Configuration
* Note that 950 mode configuration is identical to 650 configuration

8.1.1 450 Mode

After a hardware reset, bit 0 of the FiCR is cleared, making the 950 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 FiCR[0] is cleared, all other mode selection parameters are ignored.

8.1.2 550 Mode

After a hardware reset, writing a 1 to FiCR[0] increases the FIFO size to 16, providing compatibility with 16C550 devices.

8.1.3 750 Mode

Writing a 1 to FiCR[0] increases 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 FiCR[5]. Note that access to FiCR[5] is protected by LCR[7]; i.e., to set FiCR[5], software should first set LCR[7] to remove the guard temporarily. When FiCR[5] is set, 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

8.1.4 650 Mode

The 950 is compatible with the 16C654 when EFR[4] is set, i.e., the device is in enhanced mode. Because 650 software drivers usually put the device into enhanced mode, running 650 drivers on the 950 results in 650 compatibility with 128-deep FIFOs, as long as FiCR[0] is set. Note that the 650 emulation mode of the 950 provides 128-byte deep FIFOs whereas the standard 16C654 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

8.1.5 950 Mode

The additional features offered in 950 mode generally only apply when the UART is in enhanced mode (EFR[4]=1). Provided FiCR[0] is set, in enhanced mode the FIFO size is 128.

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 8.11.3). In addition to larger FIFOs and higher baud rates, the enhancements of the 950 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 & 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 (0x16C95011)
- Readable status for automatic in-band and out-of-band flow control
- Flexible *M N*/8 clock prescaler (see section 8.10.2)
- Fixed 48-MHz clock input from internal PLL (12-MHz crystal input multiplied by 4)
- Programmable sample clock to allow data rates up to 12 Mbps (see section 8.10.3)
- 9-bit data mode

The 950 trigger levels are enabled when ACR[5] is set (bits 4 to 7 of FiCR are ignored). Then arbitrary trigger levels can be defined in RTL, TTL, FCL and FCH registers (see section 8.11). 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 how to synthesize arbitrary baud rates. The default division value is 4 to provide backward compatibility with 16C654 devices.

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, the 950 can employ any over-sampling rate from 4 to 16 by programming the TCR register. The default value after a reset for this register is 0x00, which corresponds to a 16-cycle sampling clock. Writing 0x01, 0x02 or 0x03 also results 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. For example, to set the device to a 13 cycle sampling clock it would be necessary to write 0x0D to TCR. For further information see section 8.10.3.

The 950 also offers 9-bit data frames for multi-drop industrial applications.

8.2 Register Description Tables

The UART registers are accessible from the host machine I/O space. In order to support legacy software, selection of the registers is also dependent on the state of the LCR and 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 43.

<table>
<tr><th>Register Name</th><th>Address[0]</th><th>R/W</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td>THR [1]</td><td>0x00</td><td>W</td><td colspan="8">Data to be transmitted</td></tr>
<tr><td>RHR [1]</td><td>0x00</td><td>R</td><td colspan="8">Data received</td></tr>
<tr><td>IER [1,2]
650/950 Mode</td><td rowspan="2">0x01</td><td rowspan="2">R/W</td><td>CTS interrupt mask</td><td>RTS interrupt mask</td><td>Special Char. Detect</td><td rowspan="2">Unused</td><td rowspan="2">Modem interrupt mask</td><td rowspan="2">Rx Stat interrupt mask</td><td rowspan="2">THRE interrupt mask</td><td rowspan="2">RxRDY interrupt mask</td></tr>
<tr><td>550/750 Mode</td><td colspan="3">Unused</td></tr>
<tr><td>FiCR [3]
650 mode</td><td rowspan="3">0x02</td><td rowspan="3">W</td><td colspan="2">RHR Trigger Level</td><td colspan="2">THR Trigger Level</td><td rowspan="3">DMA Mode / Tx Trigger Enable</td><td rowspan="3">Flush THR</td><td rowspan="3">Flush RHR</td><td rowspan="3">Enable FIFO</td></tr>
<tr><td>750 mode</td><td colspan="2">RHR Trigger Level</td><td>FIFO Size</td><td>Unused</td></tr>
<tr><td>950 mode</td><td colspan="4">Unused</td></tr>
<tr><td>ISR [3]</td><td>0x02</td><td>R</td><td colspan="2">FIFOs enabled</td><td colspan="2">Interrupt priority (Enhanced mode)</td><td colspan="3">Interrupt priority (All modes)</td><td>Interrupt pending</td></tr>
<tr><td>LCR [4]</td><td>0x03</td><td>R/W</td><td>Divisor latch access</td><td>Tx break</td><td>Force parity</td><td>Odd / even parity</td><td>Parity enable</td><td>Number of stop bits</td><td colspan="2">Data length</td></tr>
<tr><td>MCR [3,4]
550/750 Mode</td><td rowspan="2">0x04</td><td rowspan="2">R/W</td><td colspan="2">Unused</td><td>CTS & RTS Flow Control</td><td rowspan="2">Internal Loop Back Enable</td><td rowspan="2">OUT2 (Int En)</td><td rowspan="2">OUT1</td><td rowspan="2">RTS</td><td rowspan="2">DTR</td></tr>
<tr><td>650/950 Mode</td><td>Baud prescale</td><td>IrDA mode</td><td>XON-Any</td></tr>
<tr><td>LSR [3,5]
Normal</td><td rowspan="2">0x05</td><td rowspan="2">R</td><td rowspan="2">Data Error</td><td rowspan="2">Tx Empty</td><td rowspan="2">THR Empty</td><td rowspan="2">Rx Break</td><td rowspan="2">Framing Error</td><td>Parity Error</td><td rowspan="2">Overrun Error</td><td rowspan="2">RxRDY</td></tr>
<tr><td>9-bit data mode</td><td>9th Rx data bit</td></tr>
<tr><td>MSR [3]</td><td>0x06</td><td>R</td><td>DCD</td><td>RI</td><td>DSR</td><td>CTS</td><td>Delta DCD</td><td>Trailing RI edge</td><td>Delta DSR</td><td>Delta CTS</td></tr>
<tr><td>SPR [3]
Normal</td><td rowspan="2">0x07</td><td rowspan="2">R/W</td><td colspan="8">Temporary data storage register and
Indexed control register offset value bits</td></tr>
<tr><td>9-bit data mode</td><td colspan="7">Unused</td><td>9th Tx data bit</td></tr>
<tr><td colspan="11">Additional Standard Registers – These registers require divisor latch access bit (LCR[7]) to be set to 1.</td></tr>
<tr><td>DLL</td><td>0x00</td><td>R/W</td><td colspan="8">Divisor latch bits [7:0] (Least significant byte)</td></tr>
<tr><td>DLM</td><td>0x01</td><td>R/W</td><td colspan="8">Divisor latch bits [15:8] (Most significant byte)</td></tr>
</table>

Table 13: 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 14: 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	'0'	Special Char Detect	DTR	RTS	Remote Tx Disabled	Tx Disabled
RFL [6,10]	011	R	Number of characters in the receiver FIFO							
TFL [3,6,10]	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 17)							

Table 15: 950 Specific Registers

Register Name	Address[0]	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
32-bit[11] FIFO Access registers	0x08[11] to 0x0B	W	On any write to these registers a byte will be written to the TX FIFO. Writing any of these locations is functionally the same as writing to THR (Address offset 0x00).							
		R	On any read of these registers a byte will be returned from the RX FIFO. Reading any of these locations is functionally the same as reading RHR (Address offset 0x00).							
	0x0C		This location is not assigned to the UART							
TFL[10]	0x0D	R	Returns the Tx FIFO fill level.							
RFL[10]	0x0E	R	Returns the Rx FIFO fill level.							
GDS[10]	0x0F	R								Good Data Status

Table 16: OXCFU950 Specific Registers

Register access notes:

Note 0: Offset from function base address in I/O space
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
Note 10: TFL, RFL and GDS are available at two locations, accessing via the OXCFU950 specific register location being the most simple
Note 11: The 32-bit FIFO Access registers can also be accessed at 0x00 through 0x03, when DBURST is set to 0x01. This is for systems where the UART must be mapped to an 8-byte address space.

Register Name	SPR Offset [12]	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	Additional Status Enable	ICR Read Enable	950 Trigger Level Enable	DTR definition and control		AutoDSR 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]			
	0x03	R/W	Unused							
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 (0x0B)							
CSR	0x0C	W	Writing 0x00 to this register will reset the UART							
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			Disable wake-up sensitivity	Δ DCD Wakeup disable	Trailing RI edge disable	Δ DSR Wakeup disable	Δ CTS Wakeup disable
RFC	0X0F	R	FiCR[7]	FiCR[6]	FiCR[5]	FiCR[4]	FiCR[3]	FiCR[2]	FiCR[1]	FiCR[0]
GDS[13]	0X10	R	Unused							Good Data Status
DMS	0x11	R/W	Force internal TxRdy inactive	Force internal RxRdy inactive	Unused				Internal TxRdy status (R)	Internal RxRdy status (R)
PIDX	0x12	R	Hardwired Port Index (0x00)							
	0x13	R/W	Unused							

Table 17: Indexed Control Register Set

Note 12: 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.

<u>Writing to ICR registers:</u>
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).

<u>Reading from ICR registers:</u>
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 $x1xxxxxx_2$ 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 $x0xxxxxx_2$ to ICR, thus enabling access to standard registers again.

Note 13: GDS available directly in OXCFU950 specific register space at 0x0F

Register Name	SPR Offset [11]	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Indexed Control Register Set – alternative baud rate control registers										
A_LATCH	0x30	R/W	To access addresses 0x31 to 0x35 in this alternative register set the value 0xEB must be written to this location.							
A_ENABLE	0x31	R/W					Enables A_TCR	Enables A_CPR	Enables A_DLM	Enables A_DLL
A_DLL	0x32	R/W	Alternative DLL register.							
A_DLM	0x33	R/W	Alternative DLM register.							
A_CPR	0x34	R/W	Alternative CPR register.							
A_TCR	0x35	R/W	Alternative TCR register.							
	0x36		Unused							
	0x37		Unused							
Indexed Control Register Set – Alternative 32-bit FIFO access registers enable										
DBURST	0x38		Unused							DBURST Enable

Table 18: OXCFU950 Specific Alternative UART Baud Rate Control Registers

Table 18 shows the set of OXCFU950 specific alternative UART baud rate control registers, which can be used instead of the standard versions of these registers. This facility prevents legacy drivers with knowledge of the 950 UART register set from making incorrect assumptions about the OXCFU950 crystal frequency and synthesizing incorrect baud rate values by writing to the standard DLL, DLM, CPR and TCR registers. It also shows the Alternative 32-bit FIFO Access Registers Enable.

8.3 Reset Configuration

8.3.1 Host Reset

After a hardware reset or soft reset (bit 7 of COR register), all writable registers are reset to 0x00, with the following exceptions:

1. DLL—reset to 0x01.
2. CPR—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

The reset state of output signals are tabulated below:

Signal	Reset state
SOUT	Inactive High
RTS#	Inactive High
DTR#	Inactive High

Table 19: Output Signal Reset State

8.3.2 Software Reset

An additional feature available in the 950 core 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. To reset the UART, write 0x00 to the CSR.

8.4 Transmitter & Receiver FIFOs

The transmitter and receiver holding registers (FIFOs), are referred to as THR and RHR respectively.

In normal operation, when the transmitter finishes transmitting a byte it removes the next data from the top of the THR and transmits it. If the THR is empty, it waits until data is written into it. If THR is empty and the last character transmission is complete (i.e. the transmitter shift register is empty) the transmitter is said to be idle. Similarly, when the receiver finishes receiving a byte, it transfers it to the bottom of the RHR. If the RHR is full, an overrun condition occurs (see section 8.5.3).

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.

FIFO size depends on the setting of the FiCR register. In byte mode, FIFOs only accept one byte at a time before indicating that they are full; this is compatible with the 16C450. 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 is indicated in LSR (see section 8.5.3). 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 8.11.2).

8.4.1 4-Byte Access to THR & RHR

As with the OX16C95*x* UARTs, the OXCFU950 THR and RHR registers can be accessed via the standard 550 compatible register space. See Table 13.

In the OXCFU950, the THR and RHR registers can also be accessed at four contiguous locations in the OXCFU950 specific register locations. See Table 16.

Note: the same RHR data is read irrespective of which of the four byte locations are read. Similarly, writing to any of the additional THR locations has the same effect as writing to any other—all four locations provide duplicate addresses to the same THR or RHR register.

This enables the host CPU to initiate a 32-bit access to the word address of the replicated THR or RHR locations. This is broken down into four separate byte-size accesses by the system PC Card or CF controller. Because each byte is individually accessed, the FIFO is shifted in the manner previously described.

The DBURST register is provided to support this feature in systems where only an 8-byte UART address space is available. Writing 1 to bit 0 of the DBURST register enables address locations 0x00 through 0x03 to be used in the same way as locations 0x08 through 0x0B.

Although the number of accesses made to the OXCFU950 register set is not changed, host CPU utilization is reduced and maximum data throughput across the PC Card or CF interface is increased.

8.4.2 FIFO Control Register—FiCR

FiCR[0]: Enable FIFO mode

logic 0 ⇒ Byte mode.
logic 1 ⇒ FIFO mode.

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

FiCR[1]: Flush RHR

logic 0 ⇒ No change.
logic 1 ⇒ Flushes the contents of the RHR

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

FiCR[2]: Flush THR

logic 0 ⇒ No change.
logic 1 ⇒ Flushes the contents of the THR in the same manner as FiCR[1] does for the RHR.

DMA Transfer Signaling:

FiCR[3]: DMA signaling mode / Tx trigger level enable

logic 0 ⇒ DMA mode 0.
logic 1 ⇒ DMA mode 1.

DMA signals are not bonded out in the OXCFU950, so this control only affects the transmitter trigger level in DMA mode 0.

FiCR[5:4]: THR trigger level

Generally in 450, 550, extended 550 and 950 modes these bits are unused (see section 8.1 for mode definition). In 650 mode they define the transmitter interrupt trigger levels and in 750 mode FiCR[5] increases the FIFO size.

450, 550 and extended 550 modes:

The transmitter interrupt trigger levels are set to 1 and FiCR[5:4] are ignored.

650 mode:

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

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

Table 20: Transmit Interrupt Trigger Levels

These levels only apply to enhanced mode and DMA mode 1 (FiCR[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, FiCR[4] is unused and FiCR[5] defines the FIFO depth as follows:

FiCR[5]=0 transmitter and receiver FIFO size is 16 bytes.
FiCR[5]=1 transmitter and receiver FIFO size is 128 bytes.

In non-enhanced mode FiCR[5] is only writable when LCR[7] is set. Note: in enhanced mode, the FIFO size is also increased to 128 bytes when FiCR[0] is set.

950 mode:

Setting ACR[5] to 1 enables arbitrary transmitter trigger level setting using the TTL register (see section 8.11.4), so FiCR[5:4] are ignored.

FiCR[7:6]: RHR trigger level

In 550, extended 550, 650 and 750 modes, the receiver FIFO trigger levels are defined using FiCR[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 8.9).

FiCR [7:6]	Mode					
	650 *FIFO Size 128*		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 21: Receiver Trigger Levels

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

In all cases, a receiver data interrupt is 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 8.11) hence ignoring FiCR[7:6].

8.5 Line Control & Status

8.5.1 False Start Bit Detection

On the falling edge of a start bit, the receiver waits for 1/2 bit and then resynchronizes the receiver's sampling clock to 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 proceeds 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.

When the first stop bit is sampled, the received data is transferred to the RHR and the receiver waits for a low transition on SIN signifying the next start bit.

The receiver continues to receive data even if the RHR is full or has been disabled (see section 8.11.3) in order to maintain framing synchronization. The only difference is that the received data is not transferred to the RHR.

8.5.2 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 22: 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 23: 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 24: 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.

The software driver must ensure that the break duration is longer than the character period for it to be recognized remotely as a break rather than data.

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

8.5.3 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 is 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 was 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 tries to re-synchronize 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: the SIN line is sampled at the bit rate). A zero character with associated break flag set is transferred to the RHR, and the receiver then waits until the SIN line returns high. The LSR[4] break flag is set when this data item gets to the top of the RHR and 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 is 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].

8.6 UART Interrupts

The serial interrupt on the OXCFU950 is routed to the OXCFU950 interrupt control, regardless of MCR[3].

8.6.1 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 & address bit interrupt.
logic 1 ⇒ Enable receiver status & 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]: Reserved[0]

IER[5]: Special character interrupt mask[1]
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 16C654-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 alternative sleep mode.
logic 1 ⇒ Enable alternative sleep mode whereby the internal clock of the channel is switched off.

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-hnhanced 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.

Note 0:
In the OX16C95*x* UARTs this bit is used for sleep mode. In the OXCFU950, sleep management is managed outside of the UART. See 8.6.4. The OXCFU950 UART core has all standard 950 power management features removed, so that a single clock domain solution is possible, so sleep mode with this device is not possible. Most OS support for power management in UART devices is based on waking the device up on a modem event such as a falling edge on ring indicator (RI). The UART core contains a single flip-flop that is not on the standard clock domain, allowing events like these to be signaled to the surrounding logic, even when the clock is not running.

Note 1:
In the OX16C95*x* UARTs 16C750-compatible mode this bit is used as an alternative sleep mode, whcih is not the case with the OXCFU950

8.6.2 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 25: 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.

8.6.3 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 950 mode trigger levels are enabled (ACR[5]=1) and the transmitter trigger level of zero is selected (TTL=0x00), a transmitter empty interrupt is only 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 8.11.8).

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.

8.6.4 UART Power Saving

UART power saving is facilitated by the OXCFU950 selectively disabling the clock supplied to the OXCFU950 UART core.

The wake-up signal can be sensitive to any of the following events

- Falling edge of RI
- Delta CTS
- Delta DSR
- Delta DCD

The sensitivity to each of these events is controlled by the modem disable mask register (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 wake-up operation due to individual modem lines.

To initiate UART power saving correctly:

- Ensure there are no outstanding interrupts
- Set MDM to enable the desired wake-up events (e.g. set to 0x0B for RI sensitivity only). This may be set up permanently elsewhere if different sensitivity for the modem status interrupt is not required under normal running operation
- Read the MSR
- Read the MSR again, to ensure all events have been cleared down
- Disable the UART clock to enable the wake-up route through to an interrupt (logic outside UART core)

To exit power saving:

- Re-enable the UART clock to disable the wake-up route to an interrupt (logic outside UART core)
- Perform a read access from the UART to ensure that enough clocks have occurred to update registered versions of the modem status lines. A suitable access with no side-effects would be to read SPR. If modem status is unimportant, the MSR could be read instead
- Read the MSR to clear the wake-up signal, subject to no other events occurring. Note: because the event occurred while the clock was turned off, a multiple change (glitch) on any of the modem lines may have triggered the wake-up, but the event may be missed and thus not recorded in the modem status register.

8.7 Modem Interface

8.7.1 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 8.11.3).

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 8.9.4).

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

OUT1# is not bonded out in the OXCFU950, but is used internally for loopback testing.

MCR[3]: OUT2
logic 0 ⇒ Force OUT2# output low when loopback mode is disabled.
logic 1 ⇒ Force OUT2# output high.

OUT2# is not bonded out in the OXCFU950, but is used internally for loopback testing.

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 modem outputs (DTR#, RTS#) are set inactive (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 MCR 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, received data is accepted as a valid Xon (see in-band flow control, section 8.9.3).

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 & transmitter data format.
logic 1 ⇒ Data is transmitted & 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 reset to 0. User writes to this flag only take effect in enhanced mode. See section 8.9.1.

8.7.2 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.

8.8 Other Standard Registers

8.8.1 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 See section 8.10 for full details.

8.8.2 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 17 and section 8.11.

8.9 Automatic Flow Control

Automatic in-band flow control, automatic out-of-band flow control and special character detection features can be used in enhanced mode and are software-compatible with the 16C654. Alternatively, 16C750-compatible automatic out-of-band flow control can be enabled 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.

8.9.1 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(s). When this occurs, the UART disables transmission as soon as any current character transmission is complete. The UART then compares the received data with the programmed Xon character(s). When a match occurs, the UART re-enables transmission (see section 8.11.6).

For automatic in-band flow control, EFR[4] 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, recognizing Xon2 as the XON character and Xoff2 as the XOFF character.
logic [10] ⇒ Single character in-band receive flow control enabled, recognizing Xon1 as the Xon character and Xoff1 and the Xoff character.
logic [11] ⇒ The behavior of the receive flow control depends 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 00.

EFR[3:2]: In-band transmit flow control mode

When in-band transmit flow control is enabled, Xon/Xoff characters are 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, EFR[4] 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[7:4], ISR[5:4], FiCR[5:4], MCR[7:5] and in-band flow control. Whenever this bit is cleared, the setting of other bits of EFR is ignored.
logic 1 ⇒ Enhanced mode. Enables the enhanced mode functions. These functions include enabling IER[7:4], FiCR[5:4], MCR[7:5]. 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[7:6], 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 is forced inactive high if the RFL reaches the upper flow control threshold. This is 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.

8.9.2 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] is set and a level 5 interrupt is asserted, if enabled by IER[5]. This flag is 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.

8.9.3 Automatic In-band Flow Control

When in-band receive flow control is enabled, the UART compares the received data with Xoff1 or Xoff2 characters to detect an Xoff condition. When this occurs, the UART disables transmission as soon as any current character transmission is complete. Status bits ISR[4] and ASR[0] are set. A level 5 interrupt occurs, if enabled by IER[5]. The UART then compares all received data with Xon1 or Xon2 characters to detect an Xon condition. When this occurs, the UART re-enables transmission and status bits ISR[4] and ASR[0] are cleared.

Any valid Xon/Xoff characters are not written into the RHR, apart from when special character detection is enabled and an Xoff2 character is received that is a valid Xoff. In this instance, the character is 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 is re-enabled. The received data is transferred to the RHR.

When in-band transmit flow control is enabled, the RFL is 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 is sent and ASR[1] is set. If ASR[1] is set and the RFL falls below the lower trigger level, Xon is sent and ASR[1] is cleared.

If transmit flow control is disabled after an Xoff has been sent, an Xon is sent automatically.

8.9.4 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. In non-enhanced mode, MCR[5] enables both RTS and CTS flow control. In enhanced mode, EFR[6] enables automatic RTS flow control and EFR[7] enables automatic CTS flow control. This allows software compatibility with both 16C654 and 16C750 drivers.

When automatic CTS flow control is enabled and the CTS# input becomes active, the UART disables 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 is forced inactive when the RFL reaches the upper trigger level and returns 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.

8.10 Baud Rate Generation

8.10.1 General Operation

The UART contains a programmable baud rate generator capable of taking the fixed 48-MHz clock input from the OXCFU950 internal PLL and dividing it by any 16-bit divisor number from 1 to 65535 written into the DLM (MSB) and DLL (LSB) registers. In addition, 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. A further feature is the Times Clock Register (TCR) which allows the sampling clock to be set to between 4 and 16.

These clock options allow for highly-flexible baud rate generation. 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 = (256 x DLM) + DLL
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 Section 8.10.3 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 (Hex) Divisor Word		Baud Rate (bps))
EA	60	50
6A	88	110
27	10	300
13	88	600
09	C4	1,200
04	E2	2,400
02	71	4,800
01	38	9,600
00	9C	19,200
00	68	28,800
00	4E	38,400
00	34	57,600
00	1A	115,200

Table 26: Standard PC COM Port Baud Rate Divisors

8.10.2 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 reset to 0 after a hardware reset but may be overwritten by software. Note: because 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 MCR[7] is set by software, the internal clock prescaler is enabled.

Upon a hardware reset, CPR defaults to 0x20 (division-by-4).

The flexible prescaler allows system designers to generate popular baud rates using clocks that are not integer multiples of the required rate. With the OXCFU950 fixed-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 1.8432 MHz.

Table 28 on the following page gives typical TCR, CPR and Divisor values, which can be used to synthesize a range of typical baud rates.

8.10.3 Times Clock Register—TCR

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

The 16C550 and other compatible devices such as 16C654 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 950 UART is designed 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 OXCFU950 has the facility to operate at baud-rates up to 12 Mbps. This is due to the fixed 48-MHz baud rate and the minimum TCR value of 4.

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 return 0000 if read.

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

Table 27: 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 are 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 operates in 16x mode.

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

Table 28 shows typical UART baud rates and the register settings which can be used to set them:

TCR Reg(Bin)	CPR Bits(7:3)	Bits(2:0)	Divisor DLM (Hex)	DLL(Hex)	Nominal Baud Rate (Bits per sec)
0100	00110	100	90	3B	50
0100	00110	100	41	8F	110
0100	00110	100	18	09	300
0100	00110	100	0C	04	600
0100	00110	100	06	02	1200
0100	00110	100	03	01	2400
0100	00110	100	01	80	4807
0100	00110	100	00	C0	9600
0100	00110	100	00	60	19,200
0100	00110	100	00	40	28800
0100	00110	100	00	30	38400
0100	00110	100	00	20	57600
0100	00110	100	00	10	115200
0100	00001	100	00	08	230400
0100	00001	000	00	08	1500000.00
0100	00001	000	00	04	3000000.00
0100	00001	000	00	02	6000000.00
0100	00001	000	00	01	12000000.00

Table 28: Typical Baud Rates and Register Settings to Achieve Them

8.10.4 Alternative Baud Rate Control Registers

Table 18 shows the set of OXCFU950 specific alternative UART baud rate control registers, which can be used instead of the standard versions of these registers. They give the facility to prevent legacy drivers with knowledge of the 950 UART register set from making incorrect assumptions about the OXCFU950 crystal frequency and synthesizing incorrect baud rate values by writing to the standard DLL, DLM, CPR and TCR registers.

Example

- Write 0xEB to UART Indirect register 0x30 to enable register access to the alternative register set.
- Write new values into A_CPR (0x34) and A_TCR (0x35) UART indirect registers.
- Write 0x0C into A_ENABLE (0x31) UART indirect register to enable the alternative CPR and TCR registers.
- Write 0x00 to UART Indirect register 0x30 to disable register access to the alternative register set.

8.11 Additional Features

8.11.1 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 0 to this bit to re-enable the transmitter if it was disabled by in-band flow control. Writing 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 0 to this bit to re-enable the remote transmitter (an XON is transmitted). Writing 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 whether 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 whether 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]: RESERVED

This bit is unused in the OXCFU950 and reads 0.

ASR[6]: FIFO size

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

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

Note: If FiCR[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.

8.11.2 FIFO Fill levels—TFL & RFL

The TFL and RFL registers can be accessed in both the 950-specific registers and the OXCFU950-specific registers; the latter is the most direct means of access.

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.

8.11.3 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 synchronization 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 are not detected.

Changes to this bit are only recognized 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 are only recognized 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

The DTR# pin is defined as follows:

logic [00] ⇒ DTR# is compatible with 16C450, 16C550, 16C654 and 16C750 (i.e. normal).

logic [01] ⇒ DTR# pin is used for out-of-band flow control. It is forced inactive high if the RFL reaches the upper flow control threshold. DTR# line is 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], 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 its idle marking state.

ACR[5]: 950 mode trigger levels enable

logic 0 ⇒ Interrupts and flow control trigger levels are as described in FiCR register and are compatible with 16C654/16C750 modes.

logic 1 ⇒ 950 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 LSR is readable.

logic 1 ⇒ The ICRs are readable.

Setting this bit maps 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.

8.11.4 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 FiCR 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.

8.11.5 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 FiCR 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 16C654 and 16C750 devices. It enables the system designer to optimize the interrupt performance hence minimizing 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.

8.11.6 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 optimizing 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 FiCR[7:6] as described in section 8.4.2. 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 FiCR[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 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 63 characters remain.

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].

8.11.7 Device Identification Registers

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

The 950 offers four bytes of device identification. The device identification 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 type and return 0x16, 0xC9 and 0x50 respectively. The REV register resides at offset 0x0B of ICR and identifies the revision of CFU950 UART core. This register returns 0x11 for the OXCFU950.

8.11.8 Nine-bit Mode Register—NMR

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

The 950 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 the 950 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, on 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 950 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 results 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: if an overrun or error interrupt actually occurs, 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, users can configure the OXCFU950 UART 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] enables 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.

8.11.9 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 UART power saving operation due to individual modem lines or the serial input line. Note: UART power saving is controlled from outside the UART. See 8.6.4.

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[4]: Disable wake-up sensitivity
logic 0 ⇒ Enables wake-ups based on the sensitivity settings of MDM[3:0]
logic 1 ⇒ Disables all wake-ups.
MDM[7:3]: Reserved
These bits must be set to 0000

8.11.10 Readable FiCR—RFC

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

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

8.11.11 Good-data status register—GDS

The GDS register is available directly in OXCFU950 specific register space at 0x0F. It is also accessible 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

8.11.12 DMA Status Register—DMS

The DMS register is located at offset 0x11 of the ICR. This register is unused in the OXCFU950 except for test purposes.

8.11.13 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 OXCFU950 this reads 0.

8.11.14 Alternative UART Baud Rate Control Registers: A_LATCH, A_ENABLE, A_DLL, A_DLM, A_CPR, A_TCR

Table 18 shows the set of OXCFU950-specific alternative UART baud rate control registers, which can be used instead of the standard versions of these registers. This facility prevents legacy drivers with knowledge of the 950 UART register set from making incorrect assumptions about the OXCFU950 crystal frequency and synthesize incorrect baud rate values by writing to the standard DLL, DLM, CPR and TCR registers. See 8.10.4 for further details.

8.11.15 Alternative 32-Bit FIFO Access Enable: DBURST

Table 18 shows the DBURST, the Alternative 32-bit FIFO Access Registers Enable. The 32-bit FIFO Access registers can be accessed at 0x00 through 0x03, when DBURST is set to 0x01. This is for systems where the UART must be mapped to an8-byte address space. See 8.4.1 for further details.

9 MULTI PURPOSE I/O (MIO) AND USB POWER MANAGEMENT MODULE

The OXCFU950 supports four multi-purpose I/Os (MIOs) for general applications which are fully configurable/controllable by software. Such applications may be used to control external devices via a serial interface, or act as a status monitor of external circuits, for example.

The MIO pins can be configured as inputs or outputs, and can cause a general purpose interrupt to be generated (individually generated for each MIO pin). In addition, in input mode, the active state can be set to logic '1' or logic '0' for additional flexibility.

Refer to the Multi-purpose I/O Configuration description (offset 0x11) and MIO interrupt Enable Register description (offset 0x13) for a description of the MIO pins configuration. See section 6.5 for further details.

In addition to the normal MIO operation, MIO[3:2] can be configured to be used as power management pins, with the use of an external circuit, to provide USB bus power generation (VBUS) and over-current detection as described in the USB Specification. Refer to Soft USB Reset and Clock Enable Register (offset 0x17) in section 6.5 for further details.

10 INTERNAL EEPROM SPECIFICATION

The OXCFU950 can be configured using its on-board EEPROM. If the EEPROM does not contain a valid image, the device remains in its default configuration after reset. Although this may be adequate for some purposes, normal functional operation requires at least a partial image in the EEPROM.

The EEPROM also allows accesses to the integrated UART and USB host controller, which can be useful for default setups.

The OXCFU950 device includes 512 bytes of internal EEPROM. This allows the device to be configured in the following ways:

1) The CIS can be modified to allow customers to differentiate their products.
2) The registers in I/O space can be written to allow the UART function to be setup in a specific way after a power on, hard or soft reset.
3) The registers in I/O space can be written to allow the USB function to be setup in a specific way after a power on, hard or soft reset.

A zone type system is used to tell the device which areas of the device are being programmed. If only one area is being programmed by the EEPROM, the programming time is reduced by only enabling that particular zone to be downloaded (encoded in header in EEPROM).

The EEPROM download occurs at three possible times during the PC Card operation:

- Following a power-up of card (after insertion of the card into a socket)
- Following a hard reset, from the RESET pin on the PC Card / CF interface.
- After a soft reset (either USB or UART) a partial reset of the card should take place as follows:
  - o Following a soft reset (via the UART COR soft reset bit) the UART FCR is reset, as is the UART. The UART register access zone should then be executed, if present.
  - o Following a soft reset of the UART (via the LCR) the UART is reset, and the UART register access zone should then be executed, if present. The UART FCR is not reset.
  - o Following a soft reset (via the USB COR soft reset bit) the USB FCR is reset, as is the USB host controller. The USB register access zone should then be executed, if present.
  - o Following a soft reset of the USB (via the LCR) the USB host controller is reset, and the USB register access zone should then be executed, if present. The USB FCR is not reset.

When one of these conditions occurs, the EEPROM controller detects whether a valid image is present on the EEPROM. If no valid image is present, the download is not performed and the default values (for attribute tuple data, etc) are selected.

During the download period the READY# line is kept high to hold off the host from interrogating the card. This is required because if normal interrogation was permitted during an EEPROM download, incorrect data may be read by the host causing error in configuration or even a graceful rejection of the card by the host.

The EEPROM can be programmed from the CF/PC Card Interface. When the EEPROM has been programmed, a PCMCIA reset or function reset must be performed to load in the new values and configure the CF950 to the required state.

If an invalid CIS is downloaded to the OXCFU950, the host may gracefully reject the card. This means that the card is no longer accessible via the host, and the EEPROM can not be reprogrammed. However, by using the CIS_MODE pin to select the default CIS, it is possible enumerate the card, and the EEPROM can be reprogrammed. To allow an EEPROM download to take place the CIS_MODE pin must be held HIGH. To stop the EEPROM download taking place the CIS_MODE pin will be held LOW.

The zones included for the EEPROM download are described below. First of all a header 16 value is read in. If it is a valid value then the EEPROM controller detects from this byte which zones are included in the download. It then reads in each zone of data and passes it to the relevant area in the OXCFU950 (i.e. CIS memory, function or local registers). When all the zones have been loaded into the relevant area, the download is complete.

10.1 Zone 0 : Zone Header

The zone header is the first value to be read and is at address 0 in the EEPROM. It has the following format:

Bits	Description
15:4	These bits should return 0xF87 to identify a valid program.
2	1—Zone 1 (CIS Configuration) data exists 0—Zone 1 does not exist
1	1—Zone 2 (USB Function Access) data exists 0—Zone 2 does not exist
0	1—Zone 3 (UART Function Access) data exists 0—Zone 3 does not exist

Table 26. Zone 0 format

The programming data for each zone follows the preceding zone if it exists. For example, a header value of 0xF877 indicates that all zones exist and they follow one another, while 0xF876 indicates that only zone 1 and zone 2 exist.

10.2 Zone 1 : CIS Configuration Zone

This zone allows the user to provide custom tuple information for the card information structure, overriding the default hard-coded tuple values found in the device. Downloading into this zone programs the internal RAM with the user's tuple data and automatically sets the source of the CIS to be this RAM.

Tuple data bytes are read until the specified number of type data-bytes have been collected in which case the EEPROM moves over to the next zone if it exists, or the EEPROM download terminates if no other zones are present.

The zone contains data to be downloaded into the attribute memory (RAM address 0 to 255, attribute memory addresses 0 to 510, even locations only).

The first WORD in this zone describes how many bytes of data are present to be downloaded. The next words contain the tuple data for the attribute memory.

Byte Number	Description	
	15 8	7 0
1st WORD	"00000000"	Number of tuple bytes in Attribute Memory (M) Note : Must be a value of multiple of 2
2nd WORD	Attribute Mem.Tuple Byte 1	Attribute Mem.Tuple Byte 0
3rd WORD	Attribute Mem.Tuple Byte 3	Attribute Mem.Tuple Byte 2
(N + 1) th WORD	Attribute Mem.Tuple Byte N-1	Attribute Mem.Tuple Byte N-2

Table 27. Zone 1 Format

The EEPROM controller outputs a flag to signify whether a new CIS has been downloaded from the EEPROM. This allows the memory controller to select between the default CIS or the new CIS from the EEPROM.

10.3 Zone 2 : USB Function Access zone.

This zone of the EEPROM contains information required to do any vendor-specific setup of the chip for USB operation. 8-bit values can be written to any valid location in I/O space. Therefore this zone can set up the UART, USB and LCR if required. This zone is executed if present on a hard reset, or a USB soft reset (via LCR or FCR). Care should be taken that the contents of zone 2 and zone 3 are compatible, and do not conflict with each other.

Bits	Description
15	0—no more configuration WORDs to follow in zone 2. Move to next available zone or end EEPROM program if no more zones are enabled in the header. 1—another configuration WORD follows for the local configuration registers
14:8	I/O space address.
7:0	8-bit value of the register to be programmed

Table 28. Zone 2 Format

10.4 Zone 3 : UART Function Access

This zone of the EEPROM contains information required for any vendor-specific setup of the chip for UART operation. 8-bit values can be written to any valid location in I/O space. Therefore this zone can set up the UART, USB and LCR if required. This zone is executed if present on a hard reset, or a UART soft reset (via LCR or FCR). Care should be taken that the contents of zone 2 and zone 3 are compatible, and do not conflict with each other.

Bits	Description
15	0—no more configuration WORDs to follow in Zone 2. Move to next available zone or end EEPROM program if no more zones are enabled in the header. 1—another configuration WORD follows for the local configuration registers
14:8	I/O space address.
7:0	8-bit value of the register to be programmed

Table 29. Zone 3 Format

11 INTRODUCTION TO THE OXCFU950 SOFTWARE DRIVERS

This section gives a brief overview of the organization and usage of software drivers for the OXCFU950 operating in the Microsoft Windows XP™ environment.

The following figure illustrates the driver stack that is created in Windows XP for the OXCFU950 drivers. Once the device is inserted, the PCMCIA bus driver (*pcmcia.sys*) drives Multi-Function driver (*mf.sys*), and it loads both the UART(*oxser.sys*) and USB (*oxcfu950.sys*) driver.

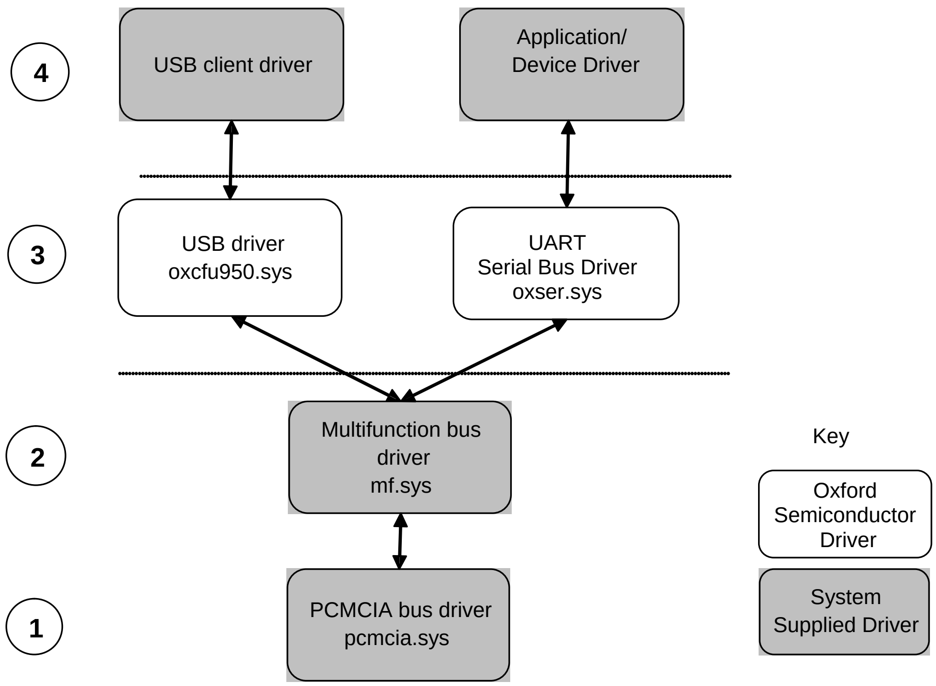


Figure 1.1 OXCFU950 Driver Stack on Windows XP

Starting from the bottom of the figure, the following describes each driver in the stack:

1. At the bottom of the driver stack tree is the PCMCIA driver *pcmcia.sys* which is a built-in driver of Windows XP. The PCMCIA driver, *pcmcia.sys* detects the OXCFU950 device when the card is inserted, and reports the device details described in attribute memory in the OXCFU950. Because the OXCFU950 complies with the PC Card multifunction standard, the Multi-Function driver, *mf.sys* is automatically loaded after PCMCIA 16-bit PC Card or CompactFlash/ CF+card is inserted.

2. The Multi-Function driver, *mf.sys* allows both functions to communicate with the hardware independently and defines a range of I/O resources that are exclusive to each function. Once the *mf.sys* is loaded by *pcmcia.sys*, *mf.sys* enumerates the functions such as Device 0 (which is the UART Port), and the Device1(which is the USB Controller) and thereby causes the Plug and Play (PnP) Manager to load individual function drivers. *mf.sys* is installed by the system supplied *inf* file, *mf.inf*.

3. The Multi-function driver, *mf.sys* enables the individual device drivers to access individual hardware resources without any interaction with each other. The OXCFU950 is driven by two individual drivers, The UART port is driven by the *oxser.sys* and the USB port is driven by the *oxcfu950.sys*.

   The UART driver oxser.sys has an interface compatible with built-in Microsoft COM port driver, serial.sys, and the USB driver *oxcfu950.sys* has an interface compatible with built-in Microsoft USB driver, *usbhub.sys*. The drivers at level 4 in the diagram can access the supplied Oxford Semiconductor driver using the same I/O request codes that are used for the built in Microsoft equivalents.

oxser.sys should be installed by *OXSER.INF*, and *oxcfu950.sys* should be installed by *OXCFU950.INF*. These INF files are provided in OXCFU950 driver package.

4. Each particular USB device is supported by a *USB client driver*. USB client device drivers for devices are layered directly above the *oxcfu950.sys*.

12 OPERATING CONDITIONS

Symbol	Parameter	Min.	Max.	Units
V_{DD}	DC supply voltage	-0.3	3.8	V
V_{IN}	DC input voltage	-0.3	V_{DD} + 0.3	V
I_{OUT}	DC output current		+/- 10	mA
T_{STG}	Storage temperature	-55	125	°C

Table 29: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Units
V_{DD}	DC supply voltage	2.85	3.6	V
T_O	Operating Temperature range	-40	85	°C

Table 30: Recommended Operating Conditions

13 DC ELECTRICAL CHARACTERISTICS

13.1 2.85 V to 3.6 V Operation

OXCFU950 IO Operation at 3.3 V, 2.5 V & 1.8 V					
	3.3 Volts				
Symbol	*Parameter*	*Conditions*	*Min*	*Max*	*Unit*
VDD3.3	3.3V I/Os		2.85	3.60	V
VIL	Low Level Input Voltage		-0.3	+0.8	V
VIH	High Level Input Voltage		2.0	VDD3.3 + 0.3 Note 1	V
VOL	Low Level Output Voltage	IOL = 2 mA		0.4	V
VOH	High Level Output Voltage	IOH= 2 mA	VDD3.3 - 0.4		V
I_{CC}	Operating supply current in normal mode			13.0	mA
I_{CC}	Operating supply current in sleep mode			4.0	mA
	2.5 Volts	UART & MIO Pin Groups Only			
Symbol	*Parameter*	*Conditions*	*Min*	*Max*	*Unit*
VDD2.5	2.5V I/Os		2.25	2.75	V
VIL	Low Level Input Voltage		-0.25	+0.7	V
VIH	High Level Input Voltage		1.7	VDD2.5 + 0.25 Note 1	V
VOL	Low Level Output Voltage	IOL = 1 mA		0.30	V
VOH	High Level Output Voltage	IOH= 1 mA	VDD2.5 - 0.30		V
	1.8 Volts	UART & MIO Pin Groups Only			
Symbol	*Parameter*	*Conditions*	*Min*	*Max*	*Unit*
VDD1.8	1.8V I/Os		1.65	1.95	V
VIL	Low Level Input Voltage		-0.18	0.35 * VDD1.8	V
VIH	High Level Input Voltage		0.65 * VDD1.8	VDD1.8 + 0.18 #1	V
VOL	Low Level Output Voltage	IOL = 0.35 mA		0.20	V
VOH	High Level Output Voltage	IOH= 0.35 mA	VDD1.8 - 0.20		V

Table 31: DC Electrical Characteristics

Note 1: These I/Os are 5V-tolerant

14 TIMING WAVEFORMS / AC CHARACTERISTICS

14.1 Memory Access

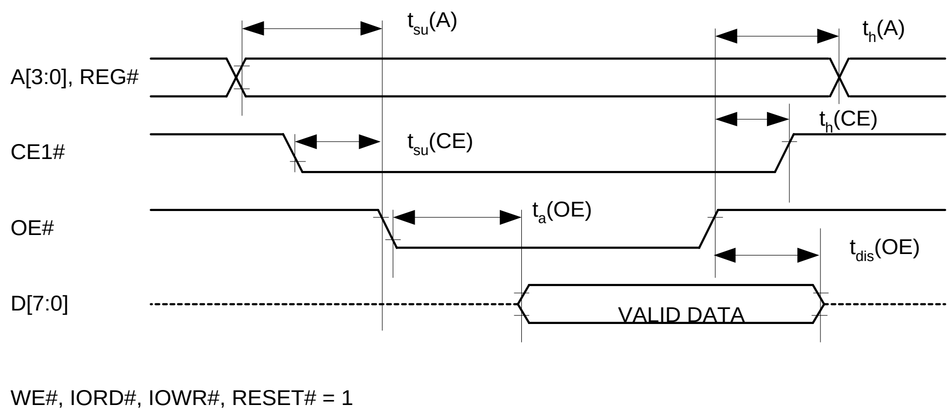


Figure 3: Attribute/Common Memory Read Timing

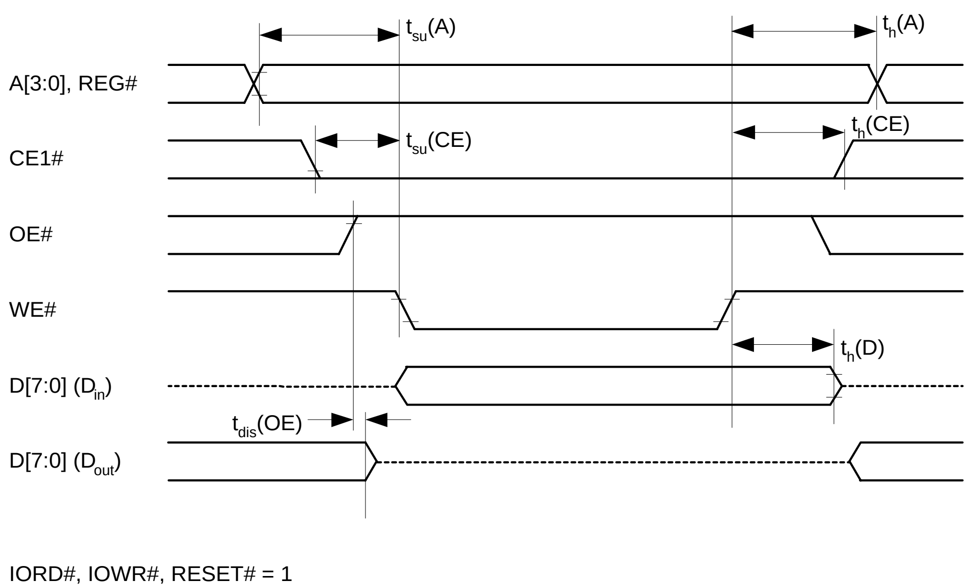
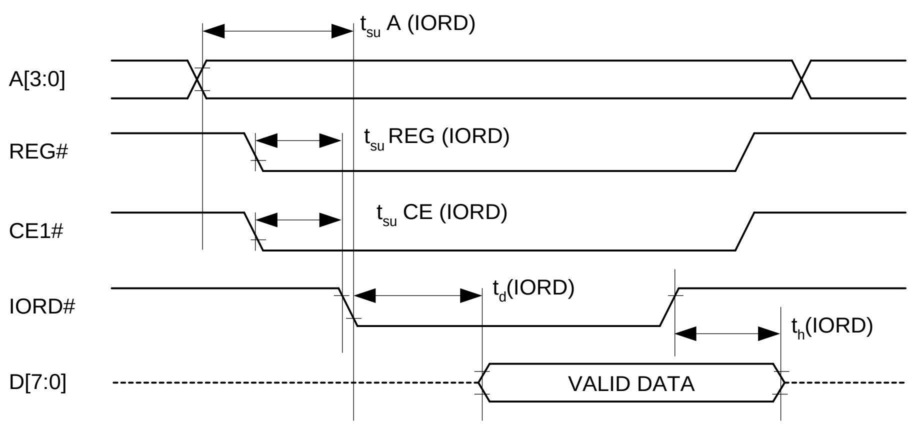


Figure 4: Attribute/Common Memory write timing

Symbol	Read Access		Write Access	
	Min (ns)	Max (ns)	Min (ns)	Max (ns)
t_{su}(A)	*TBD*	*TBD*	*TBD*	*TBD*
t_{su}(CE)	*TBD*	*TBD*	*TBD*	*TBD*
t_a(OE)	*TBD*	*TBD*		
t_{dis}(OE)	*TBD*	*TBD*	*TBD*	*TBD*
t_h(D)			*TBD*	*TBD*
t_h(A)	*TBD*	*TBD*	*TBD*	*TBD*
t_h(CE)	*TBD*	*TBD*	*TBD*	*TBD*

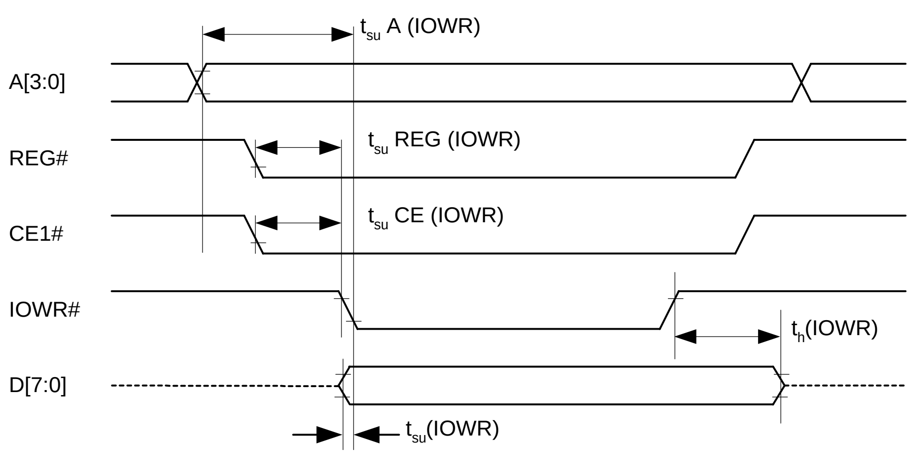
Table 32: Memory Access Timing Specification

14.2 I/O Access



OE#, WE#, IOWR#, RESET# = 1

Figure 5: I/O read timing



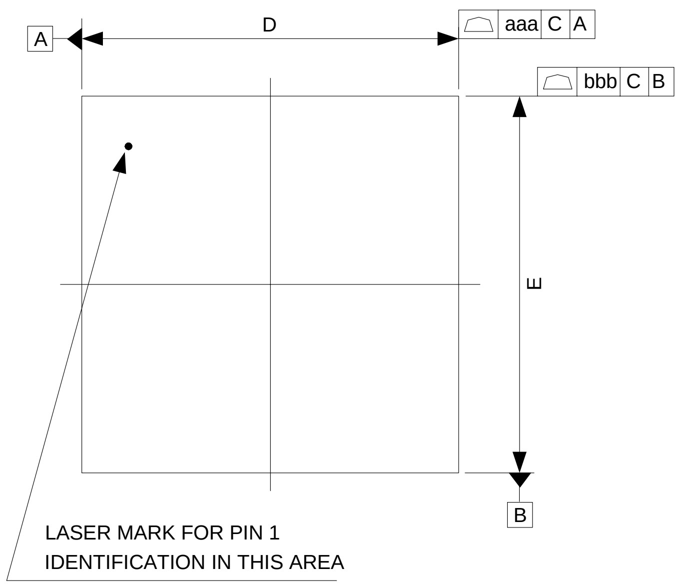
OE#, WE#, IORD#, RESET# = 1

Figure 6: I/O write timing

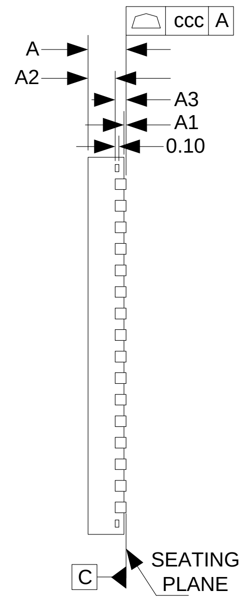
Symbol	Read Access		Write Access	
	Min (ns)	Max (ns)	Min (ns)	Max (ns)
t_{su} A (IORD)	*TBD*			
t_{su} REG (IORD)	*TBD*			
t_{su} CE (IORD)	*TBD*			
t_d(IORD)		*TBD*		
t_h(IORD)	*TBD*			
t_{su} A (IOWR)			*TBD*	
t_{su} REG (IOWR)			*TBD*	
t_{su} CE (IOWR)			*TBD*	
t_{su}(IOWR)			*TBD*	
t_h(IOWR)			*TBD*	

Table 33: I/O Access Timing Specification

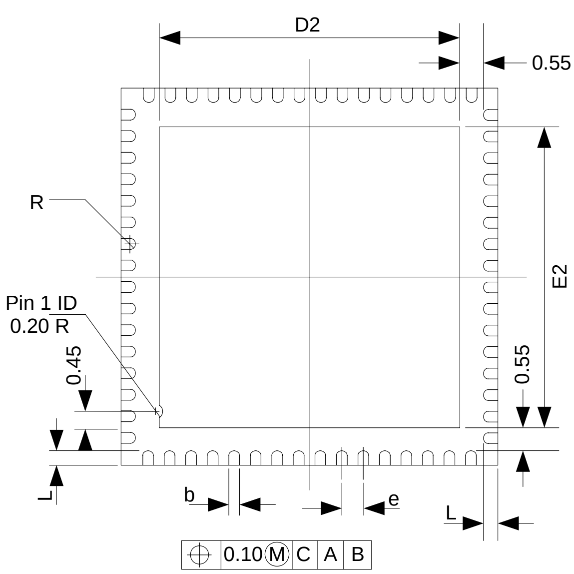
15 PACKAGE INFORMATION



TOP VIEW



SIDE VIEW



BOTTOM VIEW

15.1 Package Dimensions

Symbol	Millimeter			Inch		
	Min	Nom	Max	Min	Nom	Max
A	---	---	0.90	---	---	0.035
A1	---	---	0.05	---	---	0.001
A2	---	0.65	0.70	---	0.026	0.028
A3	0.20 REF.			0.008 REF.		
b	0.23	0.25	0.28	0.009	0.010	0.011
D	9.00 bsc			0.354 bsc		
D2	6.95	7.10	7.25	0.274	0.280	0.285
E	9.00 bsc			0.354 bsc		
E2	6.95	7.10	7.25	0.274	0.280	0.285
L	0.35	0.40	0.45	0.014	0.016	0.018
e	0.50 bsc			0.020 bsc		
R	0.125	---	---	0.005	---	---
Tolerances of Form and Position						
aaa	0.10			0.004		
bbb	0.10			0.004		
ccc	0.05			0.002		

16 ORDERING INFORMATION

OXCFU950-QFAG

- G: Green (RoHS-compliant)
- A: Revision
- QF: Package Type–QF

NOTES

This page has been intentionally left blank

CONTACT DETAILS

Oxford Semiconductor
1900 McCarthy Boulevard, Suite 210
Milpitas, CA 95035
USA

Sales e-mail: sales@oxsemi.com
Web site: http://www.oxsemi.com

DISCLAIMER