OXFORD SEMICONDUCTOR

Data Sheet

OXUSB954
USB to Quad Serial Port Bridge

Features

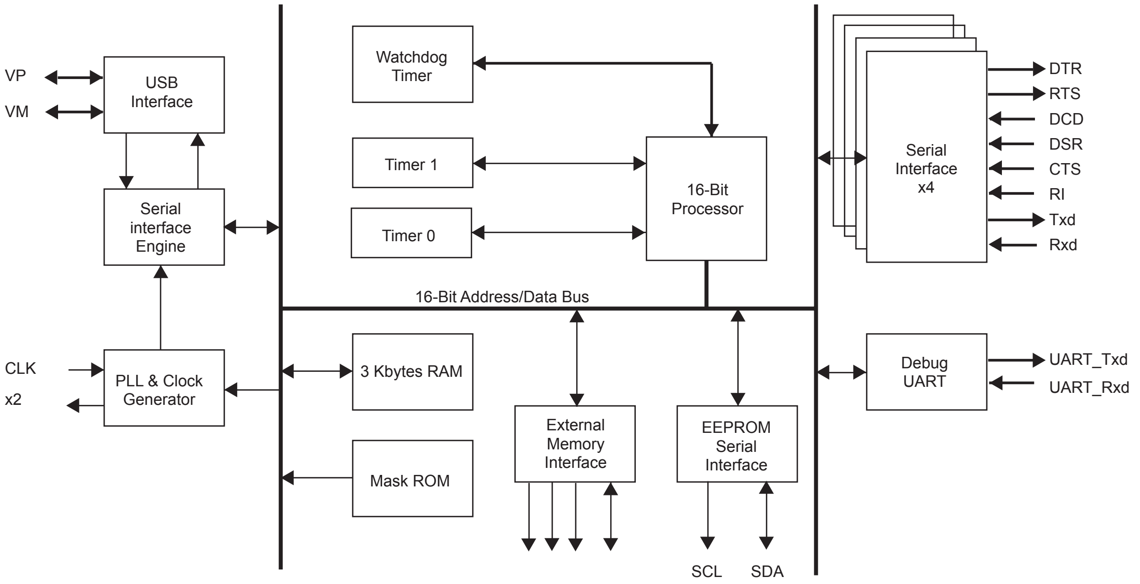
- USB 2.0 compatible at full speed; built-in 12-Mbps transceiver & SIE
- Backward-compatible with USB 1.1
- USB suspend/resume
- Serial port—high-speed transfers at up to 230 Kbps
- Additional UART for debug & code development
- Advanced 16-bit processor for USB transaction processing & control data processing
- 1.5 K×16 internal RAM buffer for fast communications
- 128-byte receive & transmit FIFO
- Configurable line control allows 8-bit words as well as 5, 6 & 7
- Optional odd, even or no parity & 1 or 2 stop bits
- Serial 2-wire interface to support EEPROM configuration
- Watchdog timer
- Utilizes low-cost external crystal circuitry
- Plug & play compatible
- USB host device drivers available (Windows® 98, Windows 2000, Windows XP)
- 3.3-V operation
- 100-pin LQFP

Overview

The Oxford Semiconductor OXUSB954 USB-to-quad serial port is the ideal bridge between USB port and up to four serial port peripherals. This intelligent device complies with USB2.0 at full speed, as well as standard serial port specifications. It delivers the advantages of USB, such as high-speed data transfers and plug-and-play capabilities, to peripherals with a serial port interface, making it ideal for connections to high-speed modems or ISDN terminal adapters. The combination of device and software renders the interface transparent to peripherals and requires no firmware changes, making it possible for serial peripherals to interface with USB with minimum modification. This feature is ideal for legacy applications.

Figure 1 shows the Oxford Semiconductor OXUSB954.

Figure 1 OXUSB954 Diagram



Functionality

The OXUSB954 integrated 16-bit processor has direct access to the RAM buffer, external memory, I/O interfaces, and all control and status registers. It runs at up to 5 MIPs. It serves as a micro-controller for USB peripherals, offering additional processing power that allows the design of intelligent peripherals that can process data prior to passing it to the host PC. This type of task optimization enhances system efficiency and improves overall performance, while the masked ROM instruction set promotes efficient code for algorithm and USB transaction processing. The processor supports up to 240 software interrupt vectors.

The processor provides the following address modes:

- Memory-to-memory
- Memory-to register
- Register-to-register
- Immediate-to-register
- Immediate-to-memory

Register, direct, immediate, indirect and indirect-indexed addressing modes are supported, plus an additional auto-increment mode, in which a register used as an address pointer is automatically incremented after each use, making repetitive operations more efficient.

The processor features program control, logical and integer arithmetic instructions. All instructions are sixteen bits wide, although some instructions require operands, which may occupy another one or two words. Several short immediate instructions are available, allowing operations with small constant operands to fit into a 16-bit instruction.

The processor divide/multiply function contains all the instructions of the base processor with additional integer divide and multiply instructions. A signed multiply instruction takes two 16-bit operands and returns a 32-bit result. A signed divide instruction divides a 32-bit operand by a 16-bit operand.

Programmable Timers

There are two built-in programmable timers that generate interrupts. Both timers decrement on every micro-second clock tick and generate an interrupt when the timer reaches zero. Similarly, a separate watchdog timer, that can also generate an interrupt on the OXUSB954, is also provided.

USB

Internal buffer memory in the USB controller is used to buffer data and USB packets. The memory is accessed by the 16-bit processor and the serial interface engine (SIE).

USB transactions are automatically routed to the memory buffer, using pointers and block sizes set up by the processor, which reads data from the interface, processes and packetizes it. If there is no activity on the USB for 3 ms, the USB enters a suspend state.

The USB controller transceiver with its differential driver can transmit and receive serial data at full speed (12 Mbps). While the transceiver driver is differential, the receiver section comprises a differential receiver and two single-ended receivers. Internally, the transceiver interfaces to the SIE; externally it connects to the USB PHY. The USB controller serial interface provides access to external EEPROMs and can support a variety of serial EEPROM formats.

Communication and data flow on the USB is implemented using uniquely-identifiable endpoints, which are the terminals of communication flow between a USB host and USB devices. The OXUSB954 supports 10 endpoints, numbered 0 to 9.

Serial Ports

The four UART serial ports can be individually configured for rates from 300 to 230.4 K baud, providing USB access to external serial devices.

An additional independent UART serial port is provided for code development and debugging, supporting transmit and receive data at rates from 7200 to 115.2 K baud. The UART timers are independent of the general-purpose timers.

When the receiver buffer transitions to full or the transmit buffer transitions to empty, a UART interrupt is generated.

Clocks

In the OXUSB954, PLL circuitry generates the internal 48-MHz clock, which is connected to CLK and X2. The circuitry is designed to allow the use of a low-cost 12-MHz crystal oscillator, supplied internally or from an application source connected directly to the CLK input pin. If the PLL is disabled, a 48-MHz crystal or clock can be used.

EEPROM Support

The OXUSB954 interfaces with a serial EEPROM device to program the OXUSB954.

An EEPROM is required for the correct functionality of the OXUSB954

External Memory

The 16-bit memory interface on the OXUSB954 can support a variety of external RAM and ROM. Each external memory space can be 8 or 16 bits wide, and can be programmed for up to seven wait states. The byte-addressable memory address range is divided into two banks that can each be assigned to internal or external ROM under register control.

External memory is required for the correct functionality of the OXUSB954.

Registers

This section documents the registers used to configure, use, and obtain status information about, the OXUSB954. Register locations and reset values are given for each register. Other locations in the register address space are reserved.

Processor Control

The processor control registers in the OXUSB954 configure the device to operate at different clock frequencies and select a power-saving mode that is used to suspend USB operation. In addition, the version control register located in the ROM BIOS can be read to discover the firmware version. Table 1 lists the processor control registers

Table 1 Processor Control Registers

Register	Address	Register Details
Config	0x0C006	page 6
Speed Control	0x0C008	page 6
Power Down Control	0x0C00A	page 5
Breakpoint	0x0C014	page 7
UART Control	0x0C0E0	page 7
Status	0x0C0E2	page 8
UART Transmit Data	0x0C0E4	page 8
UART Receive Data	0x0C0E4	page 8

PowerDownControl

Offset: 0x0C00A

Reset: 0x00

Read/write

7	6	5	4	3	2	1	0
		USB	GPIO	PUD[1]	PUD[0]	Suspend	Halt

In power-down mode, the peripherals are paused and the counters and timers stop incrementing. Specifying Suspend or Halt invokes power-down mode, although associated features differ as explained below.

USB
Enable restart on USB transition; results in device power up. Must be one of:
0—disable restart
1—enable restart

PUD[1:0][(1)]
Selects the power-up delay. Select one of the following:
00—0 ms
01—1 ms
10—8 ms
11—64 ms
The power-up delay is the time between the device powering-up and executing the application. A delay in the power-up procedure allows the clock to settle before the application commences

Suspend Invoke suspend mode; stops all device clocks to save power. Must be one of:
0—no action
1—suspend the device
Suspend mode terminates with a transition on either USB or the GPIO interrupt lines, if enabled. Device resumption is followed by the delay specified in PUD[1:0]

Halt Invoke halt mode; stops solely the processor clock. Halt mode terminates with any interrupt. Device resumption is followed by the delay specified in PUD[1:0]

Note:

[1] Oxford Semiconductor advises you not to set this field to anything other than 1 ms.

The processor can be configured to restart if there is activity on any of the four ring indicator pins.

SpeedControl

Offset: 0x0C008

Reset: 0x00

Read/write

7	6	5	4	3	2	1	0
				SPD[3]	SPD[2]	SPD[1]	SPD[0]

This register is used to select the operational speed of the OXUSB954. 48 MHz is selected after a power-up or reset.

SPD[3:0] Speed selector control; must be one of:
0000—48 MHz (default)
0001—24 MHz
0010—16 MHz
0011—12 MHz
0100—9.6 MHz
0101—8 MHz
0110—6.86 MHz
0111—6 MHz
1000—5.33 MHz
1001—4.80 MHz
1010—4.36 MHz
1011—4.00 MHz
1100—3.69 MHz
1101—3.42 MHz
1110—3.20 MHz
1111—3.00 MHz

Config

Offset: 0x0C006

Reset: 0x02

Read only

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ILC	C0	CD

ILC Custom logic selector; must be one of:
0—external custom logic
1—internal custom logic

C0[1] Clock source selector; must be one of:
0—PCLK=X1, RCLK=X1
1—PCLK=2/3 X1, RCLK=X1 (default)

CD Configuration disable; must be one of:
0—configuration by software allowed
1—configuration by software disallowed
This is a sticky bit used to lock the configuration by writing to it from within the boot PROM code

Note:

[1] X1 input pin must be 12 MHz when using the PLL.

Breakpoint

Offset: 0x0C014

Reset: 0x00

Read/write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A[15]	A[14]	A[13]	A[12]	A[11]	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]

This register holds the breakpoint address. Accessing this register generates an interrupt.

A[15:0] Breakpoint address

UARTControl

Offset: 0x0C0E0

Reset: 0x00

Read/write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											Div8	B[2]	B[1]	B[0]	

This register facilitates debugging over the UART serial debugging interface.

Div8 Pre-scaler trigger; used in conjunction with B[2:0]. Must be one of:
0—not used
1—used
If the pre-scaler is used, its effect is to divide the input clock by 8 to generate the UART clock

B[2:0] Baud rate selector; see Table 2 on page 7 for details

Table 2 Debug UART Baud Rates

B[2:0] Bit Setting	Baud Rate (Kbaud)	Baud Rate with Div8 Pre-Scaler (Kbaud)
000	115.2	14.4
001	57.6	7.2
010	38.4	4.8
011	28.8	3.6
100	19.2	2.4
101	14.4	1.8
110	9.6	1.2

Table 2 Debug UART Baud Rates

B[2:0] Bit Setting	Baud Rate (Kbaud)	Baud Rate with Div8 Pre-Scaler (Kbaud)
111	7.2	0.9

UARTStatus

Offset: 0x0C0E2

Reset: 0x00

Read only

7	6	5	4	3	2	1	0
						RxF	TxF

This register facilitates debugging over the UART serial debugging interface.

RxF Receive buffer full flag; must be one of:
0—not full
1—full

TxF Transmit buffer flag; must be one of:
0—not full
1—full

Note:

1 The device does not support error detection on receive

UARTTransmit

Offset: 0x0C0E4

Reset: 0x00

Write only

7	6	5	4	3	2	1	0
TR[7]	TR[6]	TR[5]	TR[4]	TR[3]	TR[2]	TR[1]	TR[0]

TR[7:0] UART data for transmitting

UARTReceive

Offset: 0x0C0E4

Reset: 0x00

Read only

7	6	5	4	3	2	1	0
RD[7]	RD[6]	RD[5]	RD[4]	RD[3]	RD[2]	RD[1]	RD[0]

RD[7:0] UART data received

UART Interface

On the OXUSB954 UART serial ports, individual baud rate selection is made in the appropriate line control register, which includes an optional pre-scaler. For each UART, buffer and control status are monitored in the Status register, while transmit and receive data is written or read from the UART transmit/receive data register. The UART timers are independent of the general-purpose timers.

When the receiver buffer transitions to full or the transmit buffer transitions to empty, a UART interrupt is generated on the rising edge. The interrupts are prioritized, with UART 1 having the highest priority. The QuadUARTInterrupt Status register identifies the highest-priority interrupt requiring service, adjusting this information as each interrupt is serviced until no interrupts are outstanding.

Table 3 lists the UART interface registers.

Table 3 UART Interface Registers

Register	Address	Register Details
UART 1 LineControl	0x0BF00	page 10
UART 1 Modem Control	0x0BF02	page 11
UART 1 Status	0x0BF04	page 12
UART 1 Transmit Data	0x0BF06	page 13
UART 1 Receive Data	0x0BF06	page 13
UART 2 LineControl	0x0BF10	
UART 2 Modem Control	0x0BF12	
UART 2 Status	0x0BF14	
UART 2 Transmit Data	0x0BF16	
UART 2 Receive Data	0x0BF16	
UART 3 LineControl	0x0BF20	
UART 3 Modem Control	0x0BF22	
UART 3 Status	0x0BF24	
UART 3 Transmit Data	0x0BF26	
UART 3 Receive Data	0x0BF26	
UART 4 LineControl	0x0BF30	
UART 4 Modem Control	0x0BF32	
UART 4 Status	0x0BF34	
UART 4 Transmit Data	0x0BF36	
UART 4 Receive Data	0x0BF36	
Quad UART Interrupt Status	0x0BF40	page 13
Quad UART Enable	0x0BF42	page 14

UART 1 LineControl

Offset: 0x0BF00

UART 2 LineControl

Offset: 0x0BF10

UART 3 LineControl

Offset: 0x0BF20

UART 4LineControl

Offset: 0x0BF30

Reset: 0x00

Read/write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				W[1]	W[0]	BK	SP	OE	PE	S	Div32	Div8	B[2]	B[1]	B[0]

W[1:0]
Word length; must be one of:
00—5 bits
01—6 bits
10—7 bits
11—8 bits

BK
Break trigger; must be one of:
0—resumes serial output
1—stops serial output
When a break is in force, serial output becomes inactive and remains so, regardless of transmitter activity, until BK is cleared

SP
Stick parity; must be one of:
0—no action
1—send parity as 0 if even, or 1 if odd

OE
Parity indicator; must be one of:
0—even
1—odd

PE
Parity enable; must be one of:
0—turn parity off
1—turn parity off

S
Number of stop bits; must be one of:
0—1 stop bit
1—2 stop bits

Div32
Pre-scaler trigger; used in conjunction with B[2:0]. Must be one of:
0—not used
1—used
If the pre-scaler is used, its effect is to divide the input clock by 32 to generate the UART clock

Div8
Pre-scaler trigger; used in conjunction with B[2:0]. Must be one of:
0—not used
1—used
If the pre-scaler is used, its effect is to divide the input clock by 8 to generate the UART clock

B[2:0]
Baud rate selector; see Table 4 on page 11 for details

Table 4 UART Baud Rates

B[2:0] Bit Setting	Baud Rate (Kbaud)	Baud Rate with Div32 Pre-Scaler (Kbaud)	Baud Rate with Div8 Pre-Scaler (Kbaud)
000	230.4	28.8	4.8
001	115.2	14.4	3.6
010	76.8	9.6	2.4
011	57.6	7.2	1.8
100	38.4	4.8	1.2
101	28.8	3.6	900 baud
110	19.2	2.4	600 baud
111	14.4	1.8	450 baud

UART 1 ModemControl

Offset: 0x0BF02

UART 2 ModemControl

Offset: 0x0BF12

UART 3 ModemControl

Offset: 0x0BF22

UART 4 ModemControl

Offset: 0x0BF32

Reset: 0x00

Read/write

7	6	5	4	3	2	1	0
						DTR	RTS

DTR	Data terminal ready output
RTS	Request to send output

UART 1 StatusRegister

Offset: 0x0BF04

UART 2 StatusRegister

Offset: 0x0BF14

UART 3 StatusRegister

Offset: 0x0BF24

UART 4 StatusRegister

Offset: 0x0BF34

Reset: 0x00

Read only

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						FE	BD	OV	E	RI	DCD	DSR	CTS	RxF	TxF

FE
Frame error indicator; must be one of:
0—no problems
1—frame error detected[1]

BD
Break detected indicator; must be one of:
0—no problems
1—break detected[1]

OV
Receiver overrun indicator; must be one of:
0—no problems
1—receiver overrun error

E
Data parity error indicator; must be one of:
0—no problems
1—data parity error

RI
Ring indicator; must be one of:
0—no ring
1—ring input detected

DCD
Data carrier detected indicator; must be one of:
0—no data carrier
1—data carrier input detected

DSR
Data set ready indicator; must be one of:
0—no data set ready
1—data set ready input detected

CTS
Clear to send indicator; must be one of:
0—clear to send
1—clear to send input detected

RxF
Receive buffer full flag; must be one of:
0—no problems
1—Rx buffer is full[1]

TxE
Transmit buffer full flag; must be one of:
0—no problems
1—Tx buffer is empty

Note

[1] RxF, BD and FE are cleared when the Rx data buffer is read.

UART 1 TransmitData

Offset: 0x0BF06

UART 2 TransmitRegister

Offset: 0x0BF16

UART 3 TransmitRegister

Offset: 0x0BF26

UART 4 TransmitRegister

Offset: 0x0BF36

Reset: 0x00

Write only

7	6	5	4	3	2	1	0
TR[7]	TR[6]	TR[5]	TR[4]	TR[3]	TR[2]	TR[1]	TR[0]

TR[7:0] UART data for transmitting

UART 1 ReceiveData

Offset: 0x0BF06

UART 2 ReceiveRegister

Offset: 0x0BF16

UART 3 ReceiveRegister

Offset: 0x0BF26

UART 4 ReceiveRegister

Offset: 0x0BF36

Reset: 0x00

Read only

7	6	5	4	3	2	1	0
RD[7]	RD[6]	RD[5]	RD[4]	RD[3]	RD[2]	RD[1]	RD[0]

RD[7:0] UART data received

QuadUARTInterrupt Status

Offset: 0x0BF40

Reset: 0x00

Read only

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											IS[3]	IS[2]	IS[1]	IS[0]	

IS[3:0] Quad UART interrupt status. See Table 5 on page 14 for details

Table 5 Quad UART Interrupt Status

IS[3:0]	Interrupt
0000	No interrupt pending
0001	Channel 1 Rx full
0010	Channel 2 Rx full
0011	Channel 3 Rx full
0100	Channel 4 Rx full
0101	Channel 1 Tx empty
0110	Channel 2 Tx empty
0111	Channel 3 Tx empty
1000	Channel 4 Tx empty

QuadUARTEnable

Offset: 0x0BF42

Reset: 0x00

Write only

7	6	5	4	3	2	1	0
I				E4	E3	E2	E1

I — Interrupt select; must be one of:
0—only issue an interrupt if IS[3:0] = 0000
1—always issue interrupts

E4, E3, E2, E1 — Quad UART channel enables; must be one of:
0—disable the channel
1—enable the channel
The quad UART channels should be disabled for external custom logic development

Programmable Timers

Table 6 lists the programmable timers.

Table 6 Programmable Timer Registers

Register	Address	Register Details
Watchdog Timer Count & Control Register	0x0C00C	page 15
Timer 0 Count Register	0x0C010	page 15
Timer 1 Count Register	0x0C012	page 15

WatchdogTimer

Offset: 0x0C00C

Reset: 0x00

Read only

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WT	TO[1]	TO[0]	EP	ENB	RC

WT[1][3] Watchdog timeout indicator; must be one of:
0—no timeout
1—watchdog timeout occurred

TO[1:0] Timeout count; must be one of:
00—1 ms
01—4 ms
10—16 ms
11—64 ms

EP Enable permanent watchdog timer; must be one of:
0—not permanently enabled
1—permanently enabled

ENB Enable watchdog timer; must be one of:
0—watchdog timer disabled
1—watchdog timer enabled

RC[2] Enable the watchdog reset count; must be one of:
0—watchdog reset count disabled
1—watchdog reset count enabled

Note:

1 WT is cleared following the next external reset.

2 RC must be set before a timeout occurs to prevent the watchdog triggering and the processor being reset.

3 The watchdog timer overflow causes an internal processor reset, after which WT can be read to determine whether a watchdog timeout occurred.

Timer0

Offset: 0x0C010

Reset: 0x00

Read/write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T[15]	T[14]	T[13]	T[12]	T[11]	T[10]	T[9]	T[8]	T[7]	T[6]	T[5]	T[4]	T[3]	T[2]	T[1]	T[0]

T[15:0] Timer count value

Timer1

Offset: 0x0C012

Reset: 0x00

Read/write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T[15]	T[14]	T[13]	T[12]	T[11]	T[10]	T[9]	T[8]	T[7]	T[6]	T[5]	T[4]	T[3]	T[2]	T[1]	T[0]

T[15:0] Timer count value

Interrupts

Table 7 lists the software interrupts on the OXUSB954

Table 7 OXUSB954 Software Interrupts (Sheet 1 of 2)

Number	Vector Address	Interrupt Type
0	0x0000	Timer0
1	0x0002	Timer1
2	0x0004	
3	0x0006	
4	0x0008	UART Tx (debug port)
5	0x000A	UART Rx (debug port)
6	0x000C	Reserved
7	0x000E	USB Reset
8	0x0010	USB SOF(1)
9	0x0012	USB endpoint0 no error
10	0x0014	USB endpoint0 error
11	0x0016	USB endpoint1 no error
12	0x0018	USB endpoint1 error
13	0x001A	USB endpoint2 no error
14	0x001C	USB endpoint2 error
15	0x001E	USB endpoint3 no error
16	0x0020	USB endpoint3 error
17	0x0022	USB endpoint4 no error
18	0x0024	USB endpoint4 error
19	0x0026	USB endpoint5 no error
20	0x0028	USB endpoint5 error
21	0x002A	USB endpoint6 no error
22	0x002C	USB endpoint6 error
23	0x002E	USB endpoint7 no error
24	0x0030	USB endpoint7 no error
25	0x0032	USB endpoint8 error
26	0x0034	USB endpoint8 no error
27	0x0036	USB endpoint9 error
28	0x0038	USB endpoint9 no error
29	0x003A	Quad UART interface/CU_IRQ0(2)
30	0x003C	Quad UART interface/CU_IRQ1(2)
31	0x003E	Quad UART interface/DMA_IRQ0(2)
32	0x0040	Quad UART interface/DMA_IRQ1(2)
33	0x0042	Ring 1 (if enabled)(3)
34	0x0044	Ring 2 (if enabled)(3)

Table 7 OXUSB954 Software Interrupts (Sheet 2 of 2)

Number	Vector Address	Interrupt Type
35	0x0046	Ring 3 (if enabled)[(3)]
36	0x0048	Ring 4 (if enabled)[(3)]

Notes:

1 An incoming SOF on the USB invokes an SOF interrupt.

2 These interrupts are provided for external circuitry development. They can only be used if the quad UART channels are disabled

3 These interrupts are available when the quad UART channels are enabled. Each UART has a GPIO pin dedicated for the ring indicator signal, and these inputs can be used as interrupts. They can also be used to activate the device when it is in suspend mode, by enabling the GPIO bit in PowerDownControl.

Table 8 lists the interrupt registers on the device.

Table 8 Interrupt Registers

Register	Address	Register Details
Interrupt Enable Register	0x0C00E	page 17

InterruptEnable

Offset: 0x0C00E

Reset: 0x00

Read/write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RI	QINT	USB		UART	GP	T1	T0

Setting the bits identified in this register enables interrupts from the specified elements of the device. This register must be initialized after power-up and resets.

RI	0—disable ring interrupt 1—enable ring interrupt
QINT	0—disable quad UARTs 1—enable quad UARTs
USB	0—disable USB interrupt 1—enable USB interrupt
UART	0—enable debug UART interrupt 1—disable debug UART interrupt
T1	0—disable timer1 interrupt 1—enable timer1 interrupt
T0	0—disable timer0 interrupt 1—enable timer0 interrupt

Memory Interface

For external memory, the address range C100–FFFF is divided into two banks that can each be assigned to internal or external ROM by setting bits in the Extended Memory Control Register.The address range C100–FFFF is divided into two banks that can each be assigned to internal or external ROM under register control. The memory space is byte-addressable and is divided as shown in Table 9.

Table 9 Memory Allocation

Function	Address
Internal Ram	0x0000 to 0x0BFF
External Ram	0x0C00 to 0xBFFF
Memory Mapped Registers	0xC000 to 0xC0FF
Bank 0 External/Internal ROM(1)(2)	0xC100 to 0xDFFF
Bank 1 External/Internal ROM(1)(3)	0xE000 to 0xFFFF

Note:

1 External or internal ROM is selected in the ExtendedMemory register.
2 Default is external ROM.
3 Default is internal ROM.

External RAM can be implemented as 1 × 8-bit SRAM, 2 × 8-bit SRAM or 1 × 16-bit SRAM; variations are selectable under register control.

The signals `nXRAMSEL`, `nXBHE` and `XA_0` are used differently depending on the SRAM configuration, as shown in Table 10.

Table 10 Signal Use for the Memory Interface

	2 - *n* × 8 RAM		1 - *n* × 8 ram	1 - *n* × 16
Data	XD[15:8]	XD[7:0]	XD[7:0	XD[15:0]
CS-	nXBHE	nXRAMSEL	nXRAMSEL	nXRAMSEL
BHE-				nXBHE
BLE-				XA_0

For pinout details, see “Pinout” on page 29. For SRAM timings, see “Timings” on page 26.

Table 11 lists the memory interface registers.

Table 11 Memory Interface Registers

Register	Address	Register Details
Memory Control Register	0x0C03E	page 19
Extended Memory Control Register	0x0C03A	page 19

MemoryControl

Offset: 0x0C03E

Reset: 0x02

Read/write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													RA	R0	DB

RA Wait state selector for internal RAM; must be one of:
0—0 wait states(default)
1—1 wait state

RO Wait state selector for internal ROM; must be one of:
0—0 wait states
1—1 wait state (default)

DB Enable debug mode; must be one of:
0—disable debug mode
1—enable debug mode[1]

Note:

[1] In debug mode, the internal address bus is echoed to the external address pins.

ExtendedMemory

Offset: 0x0C03A

Reset: 0x87FF

Read/write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	2CS	EB1	EB0	CRD	C[2]	C[1]	C[0]	ROW	RO[2]	RO[1]	RO[0]	RAW	RA[2]	RA[1]	RA[0]

2CS Chip selector; must be one of:
0—1 × external RAM
1—2 × external RAM (default)
For 2 × 8-bit RAM, also tie nXBHE to the upper 8 bits of the SRAM chip-select & nXRAMSEL to the lower 8 bits of the SRAM chip-select

EB1 Enable ROM bank 1; must be one of:
0—not enabled
1—enabled (default)
If ROM bank 1 is enabled, default is for external ROM

EB0 Enable ROM bank 0; must be one of:
0—not enabled (default)
1—enabled
If ROM bank 0 is enabled, default is for internal ROM

CRD Enable clean read; must be one of:
0—disabled (default)
1—enabled

C[2:0] Custom logic wait states in the range 2 to 7 (binary). Default 7

ROW	External ROM width selector; must be one of: 0—16 bits 1—8 bits (default)
RO[2:0]	External ROM wait states in the range 0 to 7 (binary). Default 7
RAW	External ROM width selector; must be one of: 0—16 bits 1—8 bits (default)
RA[2:0]	External RAM wait states in the range 0 to 7 (binary). Default 7

USB

The USB controller contains a number of registers. The first set of registers is for overall control and status functions, while the second group is dedicated to specific endpoint functions. Table 12 lists the USB registers.

Table 12 USB Registers

Register	Address	Register Details
Endpoint 0 Address Register to Endpoint 9 Address Register	0x00120, 0x00124, 0x00128, 0x0012C, 0x00130, 0x00134, 0x00138, 0x0013C, 0x00140, 0x00144	page 24
Endpoint 0 Count Register to Endpoint 9 Count Register	0x00122, 0x00126, 0x0012A, 0x0012E, 0x00132, 0x00136, 0x0013A, 0x0013E, 0x00142, 0x00146	page 24
Control & Status Register	0x0C080	page 21
Frame Number Register	0x0C082	page 21
USB Address Register	0x0C084	page 22
Command Done Register	0x0C086	page 22
Endpoint 0 Control & Status Register to Endpoint 9 Control & Status Register	0x0C090, 0x0C092, 0x0C094, 0x0C096, 0x0C098, 0x0C09A, 0x0C09C, 0x0C09E, 0x0C0A0, 0x0C0A2	page 24

Endpoints

Communication and data flow on the USB is implemented using uniquely-identifiable endpoints, which are the terminals of communication flow between a USB host and USB devices. The OXUSB954 supports 10 endpoints, numbered 0 to 9. Endpoint 0 is a control endpoint. It is the default pipe, which is used to initialize and manipulate the device. It also provides access to the device's configuration information, and supports control transfers. Other endpoints support interrupt transfers, bulk transfers, or isochronous transfers.

On endpoint 0, DIR is read-only; it indicates the direction of the last completed transaction. If the direction is incorrect, the firmware must handle the error. On other endpoints, DIR is written, and if the direction of the transfer does not match DIR, the transaction is treated as though the endpoint is not enabled.

At the end of any transfer to an armed and enabled endpoint with DIR set correctly, an interrupt occurs. Depending on whether an error occurred, the interrupt vectors to a different location. At the end of the transfer, the corresponding endpoint is disarmed and the toggle bit is advanced if no error occurs. If a packet is received with an incorrect toggle state, it is ignored, so that if the host misses an ACK and resends data, it is only seen once.

USB Control

USBControl&Status

Offset: 0x0C080

Reset: 0x00

Read/write

7	6	5	4	3	2	1	0
		URL	WK	UA	US	URA	UE

URL USB reset latch; must be one of:
0—no reset
1—USB received reset command
Write 1 to this bit to clear it

WK Wakeup trigger; must be one of:
0—no wakeup
1—send remote wakeup command to the USB

UA Activity indicator; must be one of:
0—no activity seen
1—USB activity seen
Write 1 to this bit to clear it

US USB start-of-file command indicator; must be one of:
0—no SOF command encountered
1—SOF command encountered
This bit is cleared when USBFrameNumber is read

URA USB reset active indicator; must be one of:
0—no reset currently active
1—reset command currently active

UE USB enable; must be one of:
0—USB disabled
1—USB enabled

USBFrameNumber

Offset: 0x0C082

Reset: 0x00

Read/write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					S[10]	S[9]	S[8]	S[7]	S[6]	S[5]	S[4]	S[3]	S[2]	S[1]	S[0]

S[10:0] SOF ID of the last SOF received

USBAddress

Offset: 0x0C084

Reset: 0x00

Read/write

7	6	5	4	3	2	1	0
0	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]

A[6:0] USB address of the device assigned by the host

USBCommandDone

Offset: 0x0C086

Reset: 0x00

Read/write

7	6	5	4	3	2	1	0
						M[1]	M[0]

M[1:0] Command done; see Table 13 on page 22 for details

Table 13 Command Done Settings

M1	M0	Function
0	0	Sets command done for In or Out status phase
0	1	Sets command done for Out status phase only
1	0	Sets command done for In status phase only
1	1	Undefined

USB Endpoints

EndpointControl&Status0 to EndpointControl&Status0

Offset: 0x0C090, 0x0C092, 0x0C094, 0x0C096, 0x0C098, 0x0C09A, 0x0C09C, 0x0C09E, 0x0C0A0, 0x0C0A2

Reset: 0x00

Read/write

Write

7	6	5	4	3	2	1	0
		Z	Stall	ISO	Dir	Enable	Arm

Read

15	14	13	12	11	10	9	8	7	6	5	4 0	3	2	1	
Done	Error	Setup									Stall	ISO	Dir	Enable	Arm

Read mode

Done Done indicator; must be one of:
0—not done
1—transaction completed
When the transaction is complete, Arm is set to 0

Error Error indicator; must be one of:
0—no error
1—error occurred on the last transaction for this endpoint

Setup	Setup packet indicator; must be one of: 0—no setup packet 1—setup packet received
Stall	Stall trigger; must be one of: 0—no stall sent 1—1—stall response sent in response to the next request on this endpoint
ISO	ISO enable; must be one of: 0—isochronous mode disabled 1—isochronous mode enabled
Dir	Direction indicator; must be one of: 0—receive from host 1—transmit to host
Enable	Enable endpoint; must be one of: 0—endpoint disabled 1—endpoint enabled
Arm	Endpoint armed indicator; must be one of: 0—not armed 1—armed

Write mode

Z	Zero-length packet indicator; must be one of: 0—non-zero packet 1—zero-length packet
Stall	Stall trigger; must be one of: 0—no stall response sent 1—stall response sent in response to the next request on this endpoint
ISO[2]	ISO enable; must be one of: 0—isochronous mode disabled 1—isochronous mode enabled
Dir[1]	Direction indicator; must be one of: 0—receive from host 1—transmit to host
Enable	Enable transfers to this endpoint; must be one of: 0—transactions ignored 1—transfers allowed If Arm is 0 & Enable is 1, the endpoint returns NAK to USB transmissions
ARM	Allow enabled transfers; must be one of: 0—transfer complete 1—allow enabled transfers

Note:

[1] Dir is read-only for endpoint 0.

[2] ISO is interpreted differently for endpoint 0. It allows independent stall responses for solely in or out transactions as follows: if ISO is set to 1, setting Stall to 0 stalls only in transactions and setting Stall to 1 stalls solely out transactions.

EndpointAddress0

Offset: 0x00120, 0x00124, 0x00128, 0x0012C, 0x00130, 0x00134, 0x00138, 0x0013C, 0x00140, 0x00144

Reset: 0x00

Read/write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A[15]	A[14]	A[13]	A[12]	A[11]	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]

These registers point to the memory buffer location for USB reads and writes to the endpoint.

A[15:0]	Address

EndpointCount0

Offset: 0x00122, 0x00126, 0x0012A, 0x0012E, 0x00132, 0x00136, 0x0013A, 0x0013E, 0x00142, 0x00146

Reset: 0x00

Read/write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C[15]	C[14]	C[13]	C[12]	C[11]	C[10]	C[9]	C[8]	C[7]	C[6]	C[5]	C[4]	C[3]	C[2]	C[1]	C[0]

These registers indicate the maximum packet size for the endpoint.

C[15:0]	Maximum packet size

Environmental Characteristics

Tables 14 to 15 detail the required operating conditions for the device and the DC electrical characteristics.

Table 14 Device Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit
Supply voltage	V_{DD}	–0.3 to +4.0	V
Input voltage	V_{IN} (normal)	–0.3 to V_{DD}+0.3	V
	V_{IN} (5-V tolerant)	–0.3 to +6.0	V
Storage temperature	TSTG	–55 to +125	°C
Operating temperature		0 to 70	°C

Table 15 DC Characteristics & Conditions[1]

Symbol	Parameter	Condition	Value			Unit
			Min	Typ	Max	
V_{DD}	Supply voltage		3.0	3.3	3.6	V
V_{IH}	Input high voltage		2.0	–	–	V
V_{IL}	Input low voltage		–	–	0.8	V
V+[2]	Input high voltage	Schmitt	–	1.8	2.3	V
V- [2]	Input low voltage	Schmitt	0.5	0.9	–	V
V_H[2]	Hysteresis voltage	Schmitt	0.4	–	–	V
I_{IH}	Input high current	V_{IN} = V_{DD}	–10	–	–10	µA
I_{IL}	Input low current	V_{IN} = V_{ss}	–10	–	–10	µA
V_{OH}	Output high voltage		2.4	–	–	V
V_{OL}	Output low voltage		–	–	0.4	V
I_{OZ}	3-state leakage current	V_{OH}=V_{SS}	–10	–	–10	µA
		V_{OL}=V_{DD}	–10	–	–10	µA

Note:

[1] V_{DD} @ 3.3 V ± 0.3 V

[2] For reset (nRESET is pin 66)

Power Consumption

Table 16 gives typical power consumption figures for the OXUSB954.

Table 16 OXUSB954 Power Consumption

	Suspend Mode	Idle Mode	Active Mode[1]
3.3 V	250 µA	22 mA	35 mA

Note:

[1] The active current depends on the baud rate and number of ports in use and could be higher or lower than the figure quoted.

Timings

The timings shown in Figures 2 to 4 show the behavior of the external SRAM in 8-bit and 16-bit width configurations, with one wait state selected.

Figure 2 8-Bit External SRAM

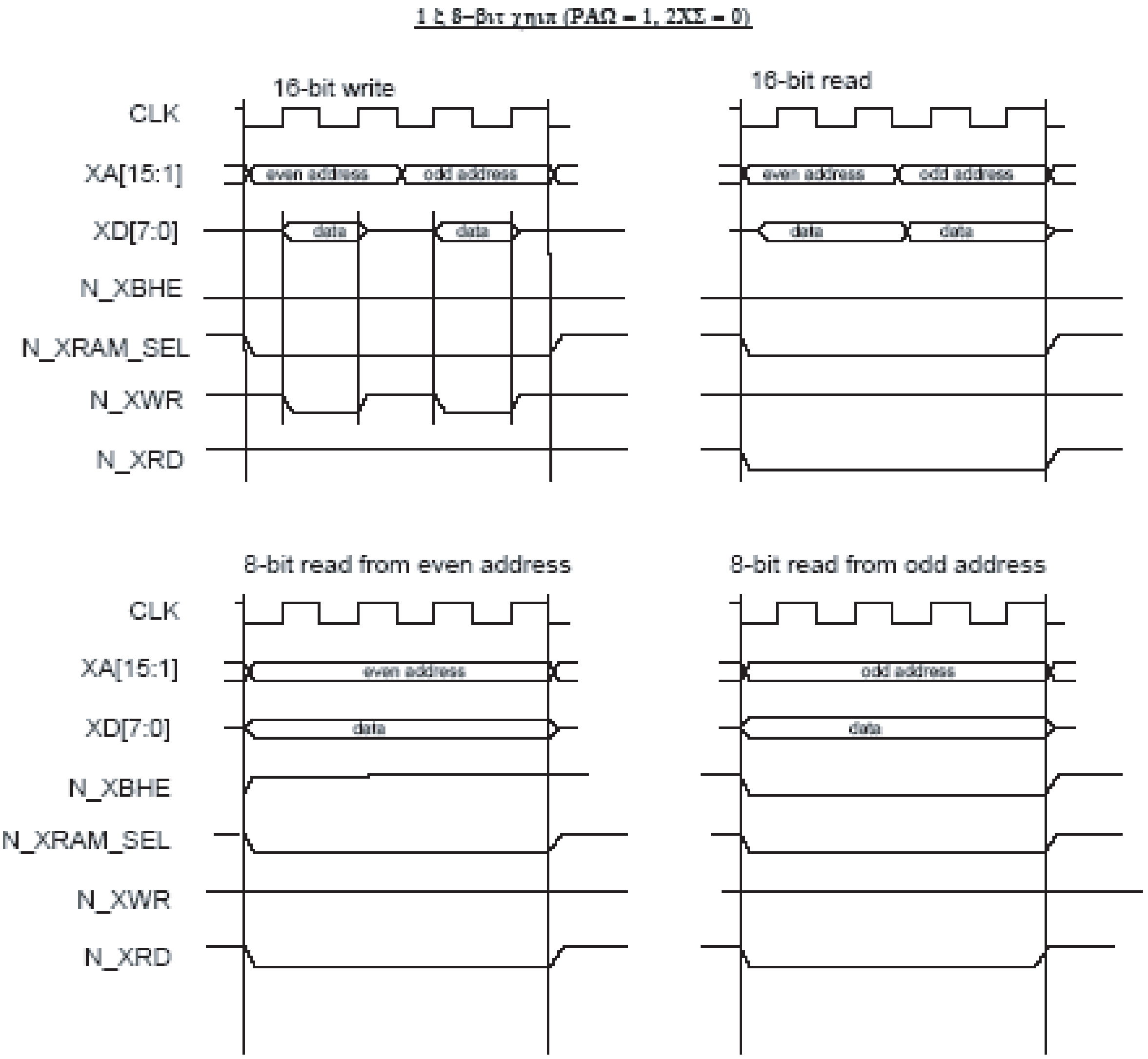


Figure 3 2 × 8-Bit External SRAM

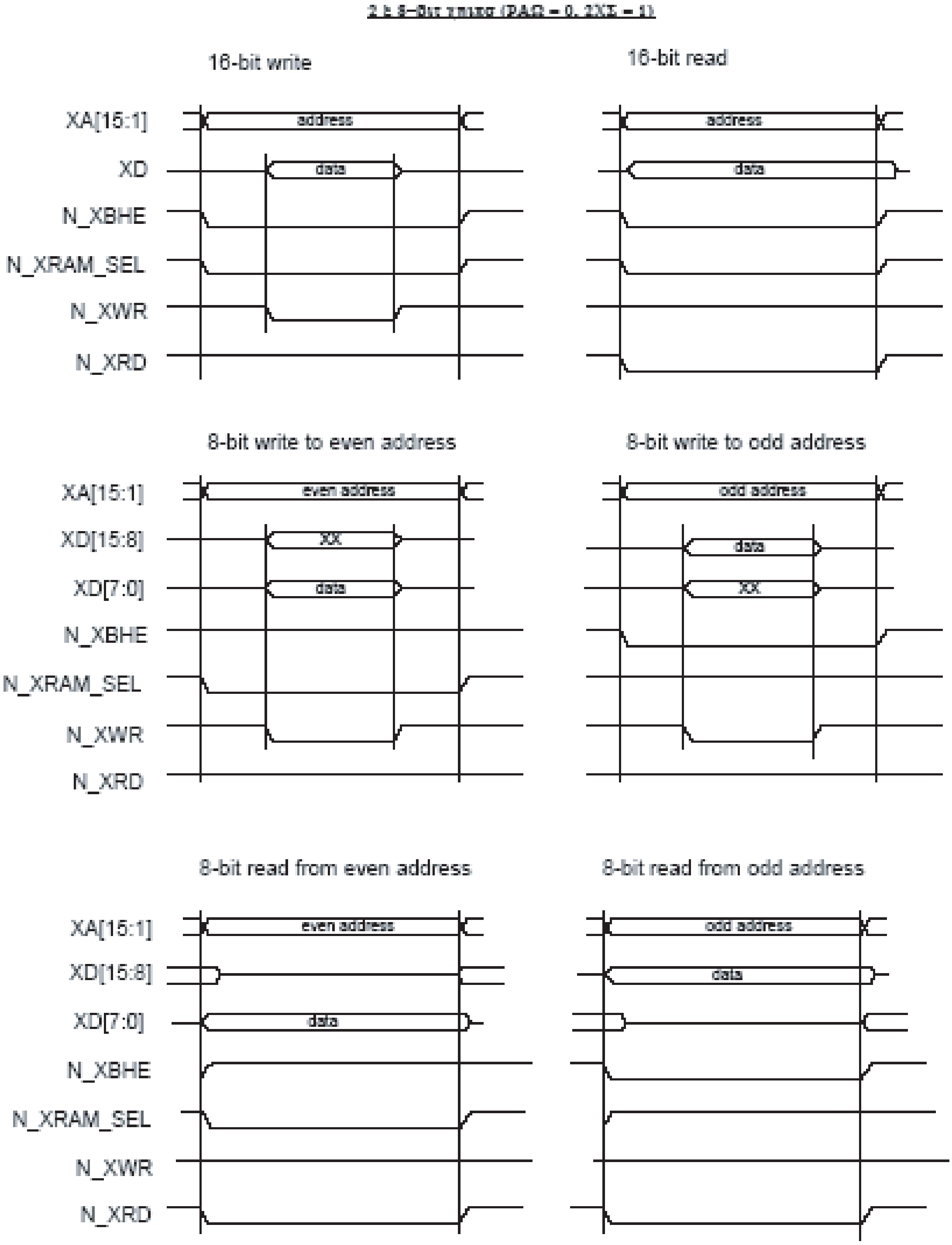
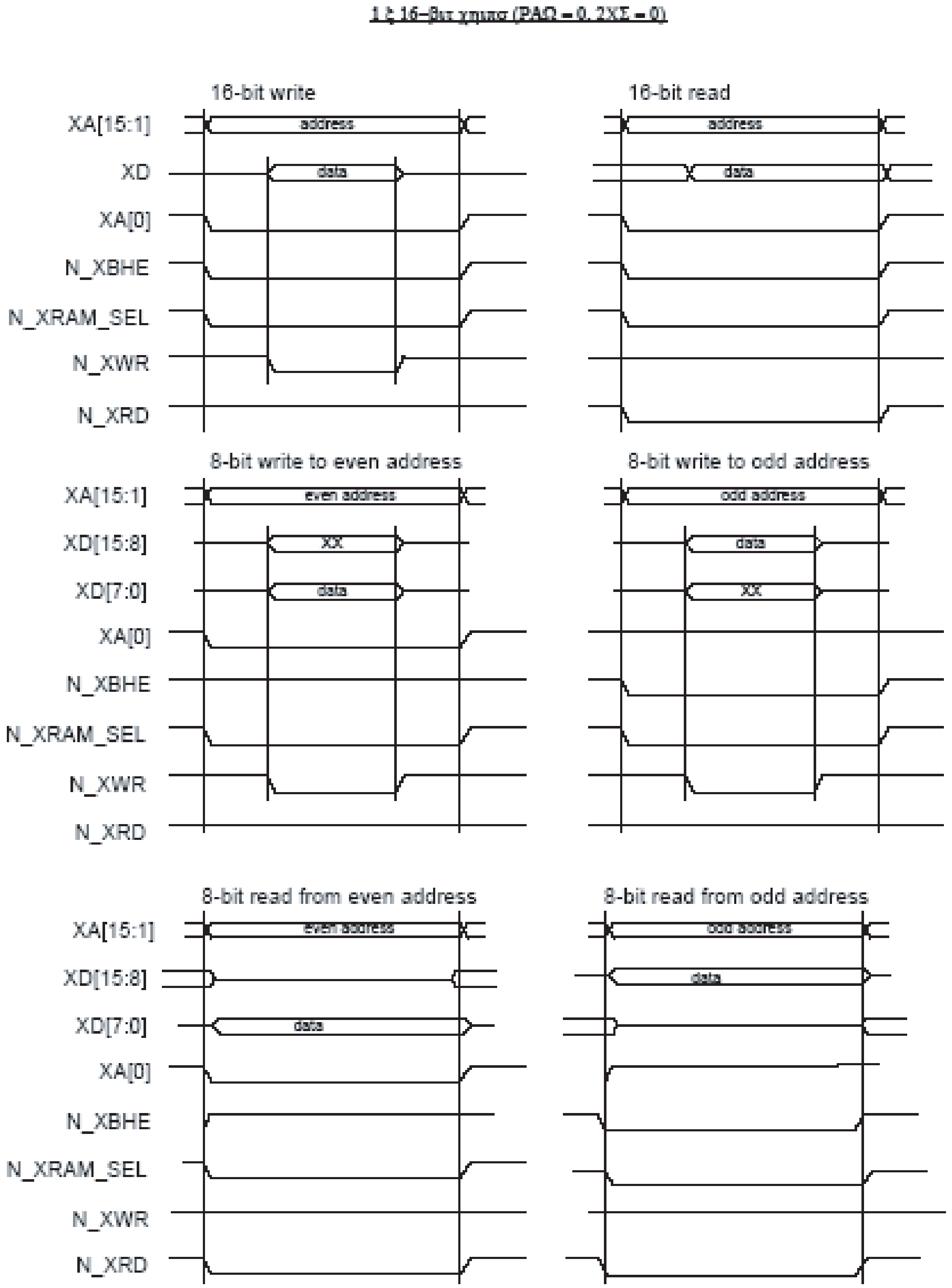


Figure 4 16-Bit External SRAM


1 ξ 16–βιτ χηιπσ (ΡΑΩ = 0, 2ΧΣ = 0)
16-bit write
16-bit read
XA[15:1]
XD
XA[0]
N_XBHE
N_XRAM_SEL
N_XWR
N_XRD
address
data
8-bit write to even address
8-bit write to odd address
XD[15:8]
XD[7:0]
even address
odd address
XX
8-bit read from even address
8-bit read from odd address

Pinout

Figure 5 shows the chip layout for the OXUSB954.

Figure 5 OXUSB954 Chip Layout

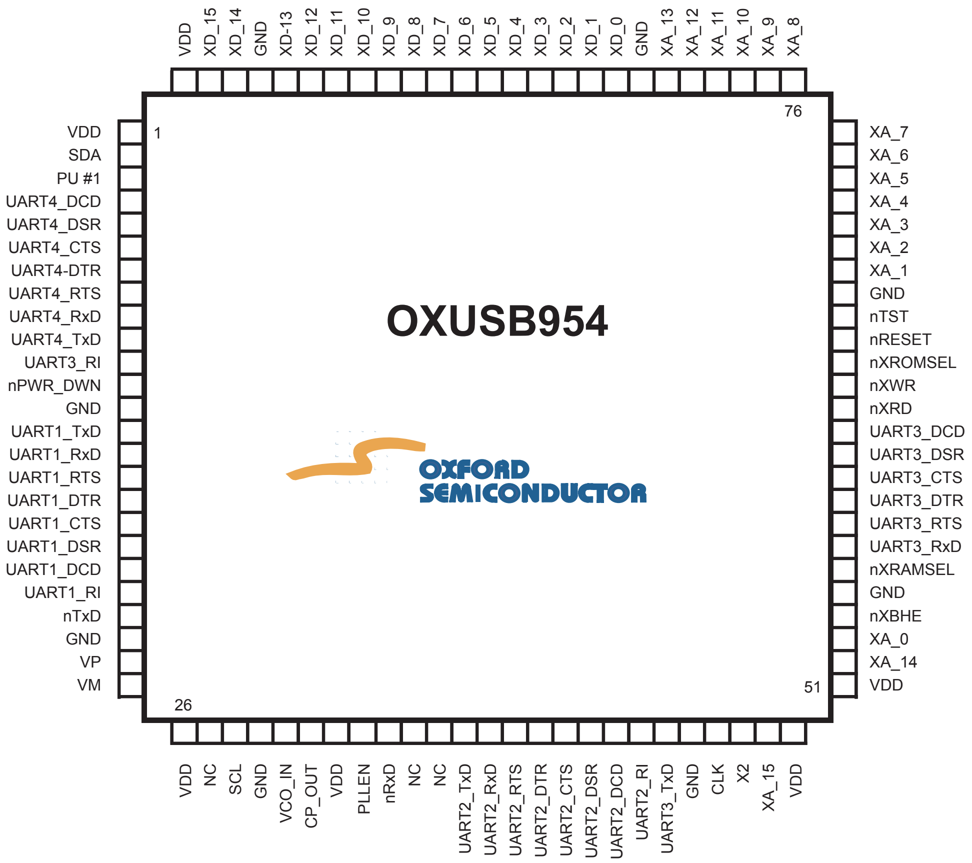


Table 17 gives the pinout details for the OXUSB954.

Table 17 OXUSB954 Pinout (Sheet 1 of 4)

Pin	I/O	Identifier	Description
1		VDD	VDD
2	In/out	SDA(1)	Serial EEPROM serial data. Connect to EEPROM/SDA
3	In	PU#1(1)	Pull up to USB +Pin for high speed
4	In	UART4_DCD(1)	Data carrier detect
5	In	UART4_DSR(1)	Data set ready
6	In	UART4_CTS(1)	Clear to send
7	Out	UART4_DTR(1)	Data terminal ready
8	Out	UART4_RTS(1)	Request to send
9	In	UART4_Rxd(1)	Receive data
10	Out	UART4_Txd(1)	Transmit data

Table 17 OXUSB954 Pinout (Sheet 2 of 4)

Pin	I/O	Identifier	Description
11	In	UART3_RI(1)	Ring indicate
12	Out	nPWR_DWN(1)	Active-low Power-down mode signal
13		GND	GND
14	Out	UART1_Txd(1)	Transmit data
15	In	UART1_Rxd(1)	Receive data
16	Out	UART1_RTS(1)	Request to send
17	Out	UART1_DTR(1)	Data terminal ready
18	In	UART1_CTS(1)	Clear to send
19	In	UART1_DSR(1)	Data Set Ready
20	In	UART1_DCD(1)	Data carrier detect
21	In	UART1_RI(1)	Ring indicate
22	Out	nTXD	Debug UART TxD
23		GND	USB GND
24	In/out	VP	USB + Pin
25	In/out	VM	USB - Pin
26		VDD	USB VDD
27	In	UART4_RI(1)	Ring indicate
28	Out	SCL(1)	Serial EEPROM clock. Connect to EEPROM/SCL
29		GND	GND
30	In	VCO_IN	PLL VCO in
31	Out	CP_OUT	PLL VCO out
32		VDD	VDD
33	In	PLLEN*	PLL enable
34	In	nRXD*	Debug UART RxD
35		N/C	No connection
36		N/C	No connection
37	Out	UART2_Txd(1)	Transmit data
38	In	UART2_Rxd(1)	Receive data
39	Out	UART2_RTS(1)	Request to send
40	Out	UART2_DTR(1)	Data terminal ready
41	In	UART2_CTS(1)	Clear to send
42	In	UART2_DSR(1)	Data set ready
43	In	UART2_DCD(1)	Data carrier detect
44	In	UART2_RI(1)	Ring indicate
45	Out	UART3_Txd(1)	Transmit data

Table 17 OXUSB954 Pinout (Sheet 3 of 4)

Pin	I/O	Identifier	Description
46		GND	GND
47	In	CLK	12-MHz clock/crystal input
48	Out	X2	12-MHz crystal output
49	Out	XA_15	External address pin
50		VDD	VDD
51		VDD	VDD
52	Out	XA_14	External address pin
53	Out	XA_0	External address pin
54	Out	nXBHE	External byte high enable (Active-low)
55		GND	GND
56	Out	nXRAMSEL	External RAM CS (active-low)
57	In	UART3_Rxd(1)	Receive data
58	Out	UART3_RTS(1)	Request to send
59	Out	UART3_DTR(1)	Data terminal ready
60	In	UART3_CTS(1)	Clear to send
61	In	UART3_DSR(1)	Data set ready
62	In	UART3_DCD(1)	Data carrier detect
63	Out	nXRD	External memory read (active-low)
64	Out	nXWR	External memory write (active-low)
65	Out	nXROMSEL	External ROM CS (active-low)
66	In	nNRESET	Reset pin
67	In	nNTST(1)	Test pin, *disconnect for normal operation*
68		GND	GND
69	Out	XA_1	External address pin
70	Out	XA_2	External address pin
71	Out	XA_3	External address pin
72	Out	XA_4	External address pin
73	Out	XA_5	External address pin
74	Out	XA_6	External address pin
75	Out	XA_7	External address pin
76	Out	XA_8	External address pin
77	Out	XA_9	External address pin
78	Out	XA_10	External address pin
79	Out	XA_11	External address pin
80	Out	XA_12	External address pin
81	Out	XA_13	External address pin
82		GND	GND
83	In/out	XD_0(1)	External Data Pin

Table 17 OXUSB954 Pinout (Sheet 4 of 4)

Pin	I/O	Identifier	Description
84	In/out	XD_1[1]	External data pin
85	In/out	XD_2[1]	External data pin
86	In/out	XD_3[1]	External Data Pin
87	In/out	XD_4[1]	External data pin
88	In/out	XD_5[1]	External data pin
89	In/out	XD_6[1]	External data pin
90	In/out	XD_7[1]	External data pin
91	In/out	XD_8[1]	External data pin
92	In/out	XD_9[1]	External data pin
93	In/out	XD_10[1]	External data pin
94	In/out	XD_11[1]	External data pin
95	In/out	XD_12[1]	External data pin
96	In/out	XD_13[1]	External data pin
97		GND	GND
98	In/out	XD_14[1]	External data pin
99	In/out	XD_15[1]	External data pin
100		VDD	VDD

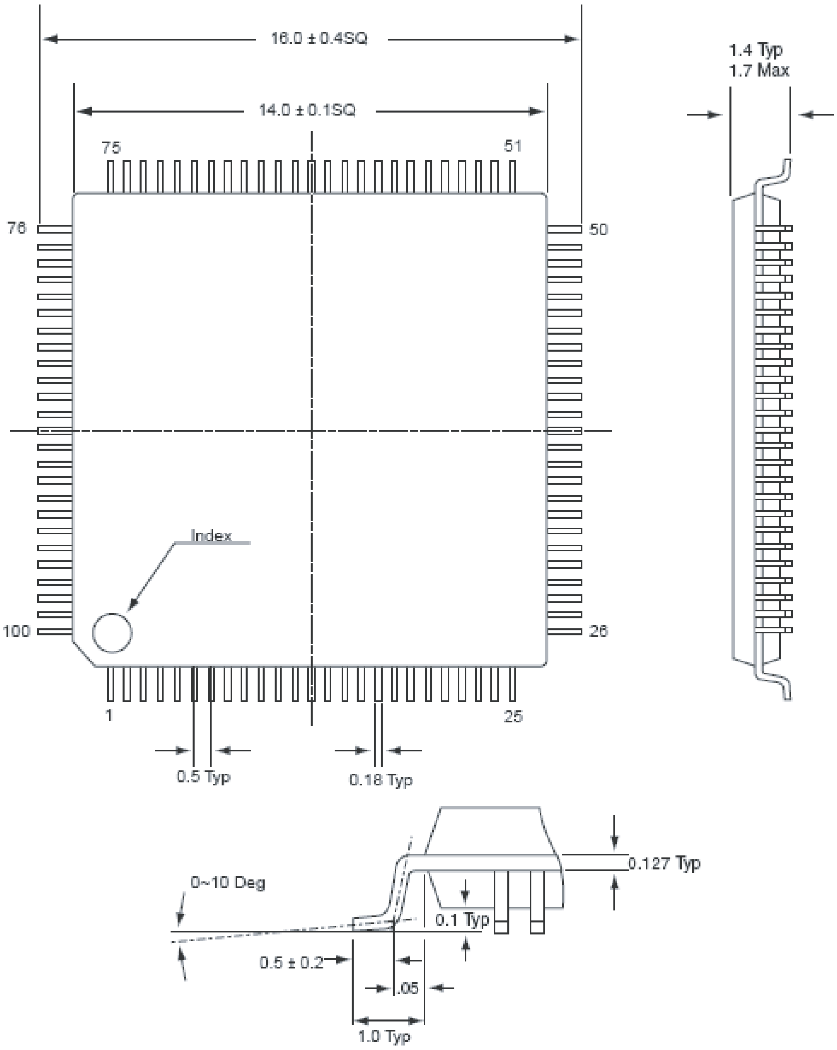
Note:

[1] 5-V tolerant.

Package Details

The OXUSB954 is supplied as a 100-pin LQFP, as shown in Figure 6.

Figure 6 OXUSB954 Package

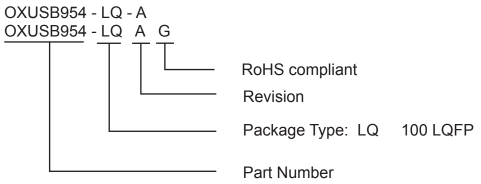


Ordering Information

The order codes for the Oxford Semiconductor OXUSB954 are as follows:

OXUSB954-LQ-A
OXUSB954-LQAG

The following conventions are used to identify Oxford Semiconductor products:



Contacting Oxford Semiconductor

Oxford Semiconductor contact details:

Oxford Semiconductor, Inc.
1768 McCandless Drive
Milpitas, CA 95035
USA

Website: http://www.oxsemi.com

Email: sales@oxsemi.com

Alternatively, you can contact your local representative.

Revision Information

Table 18 documents the revisions of this manual.

Table 18 Revision Information

Revision	Modification
December 2004	First publication
July 2005	Add operating temperature range; modify support level; add green order code
November 2005	Revision to USB compatibility; power consumption figures added
October 2006	Further clarification concerning EEPROM & external SRAM

Windows is a trademark of Microsoft, Inc., registered in the US and other countries.
All other trademarks are the property of their respective owners

The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Oxford Semiconductor, Inc. Oxford Semiconductor, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.