Embedded Systems

Ch 6A.
Serial Interface
Part A

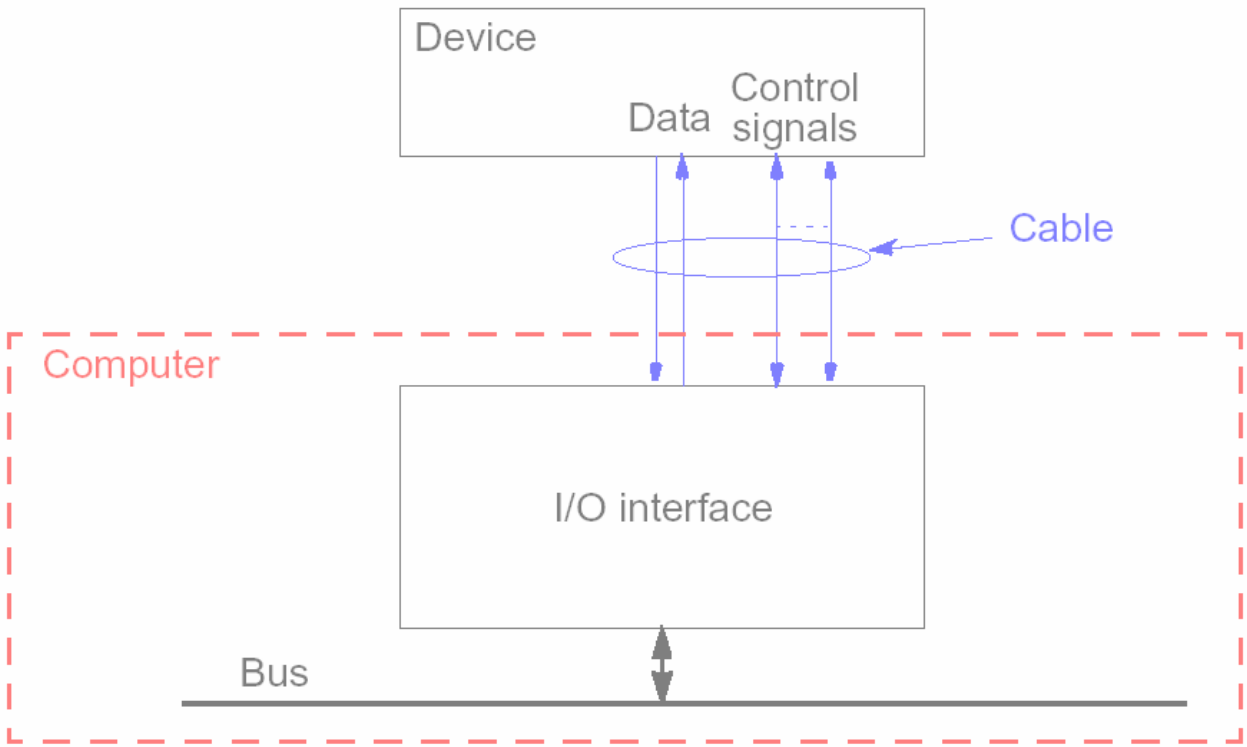
Byung Kook Kim
Dept of EECS
Korea Advanced Institute of Science and Technology

Overview

- 1. Introduction to Serial Ports
- 2. RS-232C
- 3. Serial Hardware
- 4. UARTs in Xscale

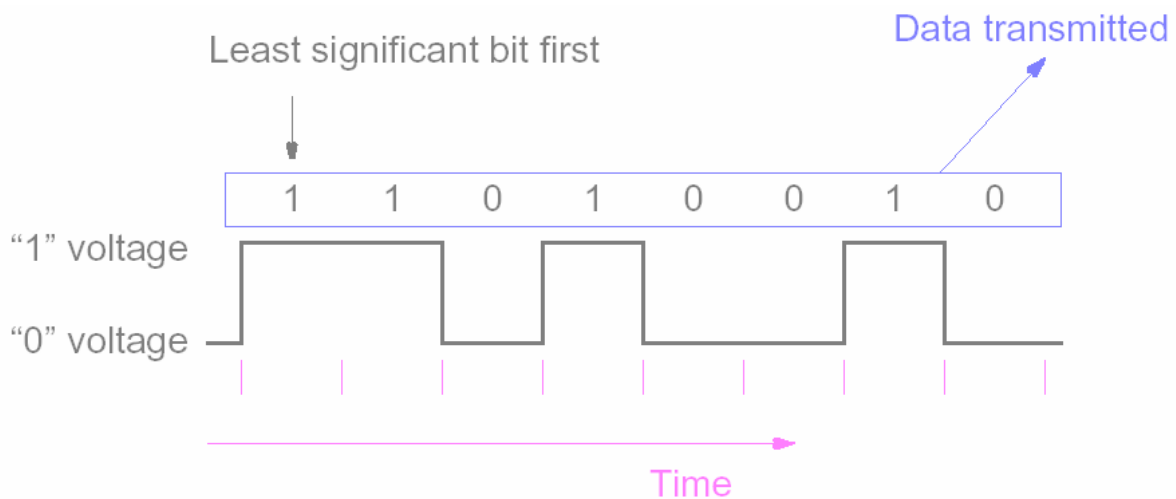
1. Introduction to Serial Interface

- **Serial Interface**
  - Data transmitted along one wire (for each direction). Bits of the data are sent one after the other. Less data wires in cable but slower.



Introduction to Serial Interface (II)

- **Serial Interfacing**
  - The Serial Port is harder to interface than the Parallel Port.
    - In most cases, any device you connect to the serial port will need the serial transmission converted back to parallel so that it can be used.
    - This can be done using a **UART** (Universal Asynchronous Receiver Transmitter).
    - On the software side of things, there are many more registers that you have to attend to than on a Standard Parallel Port (SPP).
  - Serial interface data transmission



Introduction to Serial Interface (III)

- Advantages of using serial data transfer rather than parallel
  - 1. Serial Cables can be longer than Parallel cables.
    - The serial port transmits a '1' as -3 to -25 volts and a '0' as +3 to +25 volts where as a parallel port transmits a '0' as 0v and a '1' as 5v. Therefore the serial port can have a maximum swing of 50V compared to the parallel port which has a maximum swing of 5 Volts. Therefore cable loss is not going to be as much of a problem for serial cables than they are for parallel.
  - 2. You don't need as many wires than parallel transmission.
    - If your device needs to be mounted a far distance away from the computer then 3 core cable (Null Modem Configuration) is going to be a lot cheaper that running 19 or 25 core cable. However you must take into account the cost of the interfacing at each end.

Introduction to Serial Interface (IV)

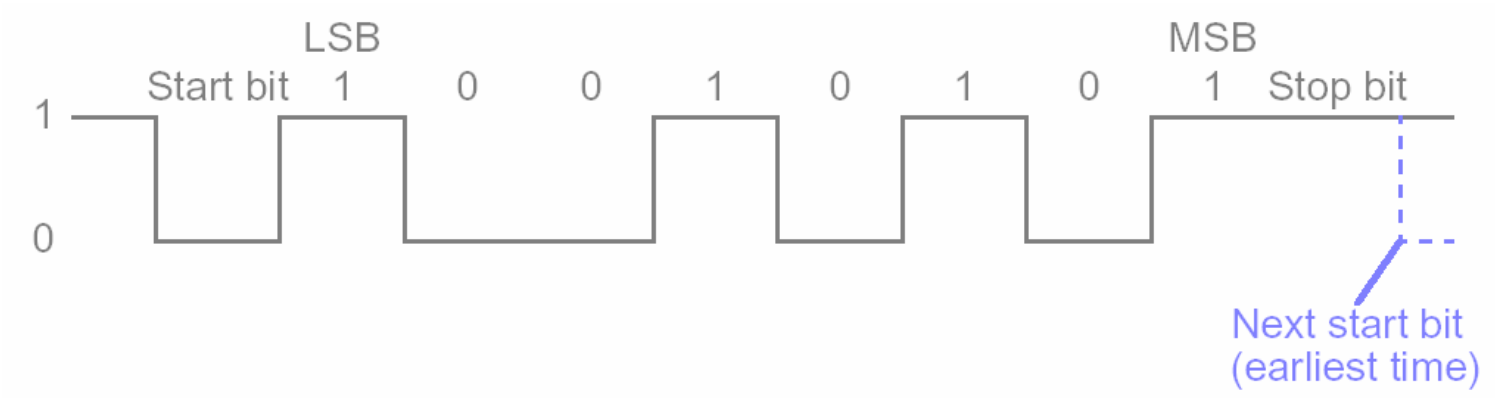
- Advantages (Cont'd)
  - 3. Infra Red devices have proven quite popular recently.
    - You may of seen many electronic diaries and palmtop computers which have infra red capabilities build in. However could you imagine transmitting 8 bits of data at the one time across the room and being able to (from the devices point of view) decipher which bits are which? Therefore serial transmission is used where one bit is sent at a time. IrDA-1 (The first infra red specifications) was capable of 115.2k baud and was interfaced into a UART. The pulse length however was cut down to 3/16th of a RS232 bit length to conserve power considering these devices are mainly used on diaries, laptops and palmtops.
  - 4. Microcontroller's have also proven to be quite popular recently.
    - Many of these have in built SCI (Serial Communications Interfaces) which can be used to talk to the outside world. Serial Communication reduces the pin count of these MPU's. Only two pins are commonly used, Transmit Data (TXD) and Receive Data (RXD) compared with at least 8 pins if you use a 8 bit Parallel method (You may also require a Strobe).

2. RS-232C

- **RS232/V24 Asynchronous Serial Interface**
  - Early (1970's) "asynchronous" serial interface used for slow printers.
    - No transmitted clock signal.
    - Character orientated.
    - Transmits in units of one fixed sized binary word (5-8 bits).
    - Transmission starts with a 0-level "start" bit, followed by the data, and terminating with one (or two) 1-level "stop" bits.
    - Multiple characters sent in this format.
    - Any time may elapse between end of one character and beginning of next.

RS-232C (II)

- RS-232/V24 Timing
  - 1, 1 ½, or 2 stop bits
  - 10 to 11 bits/1 character



RS-232C (III)

- **Hardware properties of Serial interface**
  - Two categories:
    - DCE (Data Communications Equipment): modem, TA adapter, plotter etc.
    - DTE (Data Terminal Equipment): Computer or Terminal.
  - The electrical specifications: the EIA (Electronics Industry Association) RS232C standard.
    - 1. A "Space" (logic 0) will be between +3 and +25 Volts.
    - 2. A "Mark" (Logic 1) will be between -3 and -25 Volts.
    - 3.The region between +3 and -3 volts is undefined.
    - 4. An open circuit voltage should never exceed 25 volts (In Reference to GND).
    - 5. A short circuit current should not exceed 500mA. The driver should be able to handle this without damage. (Take note of this one!)
  - The EIA RS232-C standard: maximum baud rate of 20,000 BPS!
    - A new standard, RS-232D has been recently released.
  - Two "sizes“
    - D-Type 25 pin male connector and
    - D-Type 9 pin male connector.

RS-232C (IV)

- Serial pinouts

D-type 25 pin	D-type 9 pin	Symbol	Name
2	3	TD	Transmit data
3	2	RD	Receive data
4	7	RTS	Request to Send
5	8	CTS	Clear to Send
6	6	DSR	Data Set Ready
7	5	SG	Signal Ground
8	1	CD	Carrier Detect
20	4	DTR	Data Terminal Ready
22	9	RI	Ring Indicator

RS-232C (V)

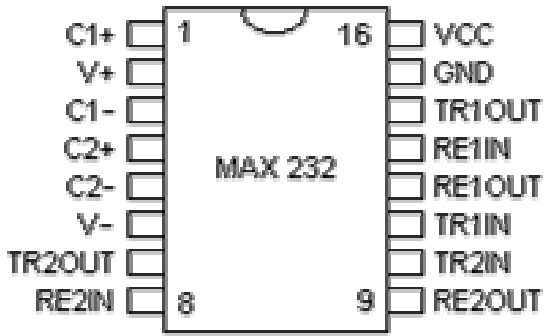
- Pin Functions
  - TD (Transmit Data)
    - Serial Data Output (TXD)
  - RD (Receive Data)
    - Serial Data Input (RXD)
  - CTS (Clear to Send)
    - This line indicates that the Modem is ready to exchange data.
  - DCD (Data Carrier Detect)
    - When the modem detects a "Carrier" from the modem at the other end of the phone line, this Line becomes active.
  - DSR (Data Set Ready)
    - This tells the UART that the modem is ready to establish a link.
  - DTR (Data Terminal Ready)
    - This is the opposite to DSR. This tells the Modem that the UART is ready to link.
  - RTS (Request To Send)
    - This line informs the Modem that the UART is ready to exchange data.
  - RI (Ring Indicator)
    - Goes active when modem detects a ringing signal from the PSTN.

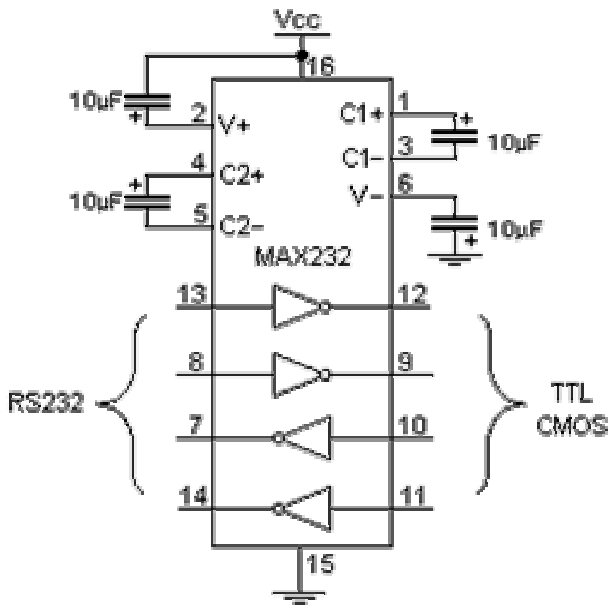
3. Serial Hardware

- **Types of UARTS (For PC's)**
  - 8250: First UART in this series. Contains no scratch register.
  - 8250A: This UART is faster than the 8250 on the bus side. Looks exactly the same to software than 16450.
  - 8250B: Very similar to that of the 8250 UART.
  - 16450: Used in AT's (Improved bus speed over 8250's).
    - Operates comfortably at 38.4KBPS. Still quite common today.
  - 16550: This was the first generation of buffered UART. It has a 16 byte buffer, however it doesn't work and is replaced with the 16550A.
  - 16550A: the most common UART use for high speed communications eg 14.4K & 28.8K Modems.
    - They made sure the FIFO buffers worked on this UART.
  - 16650: Very recent breed of UART.
    - Contains a 32 byte FIFO, programmable X-On / X-Off characters and supports power management.
  - 16750: Produced by Texas Instruments. Contains a 64 byte FIFO.

Serial Hardware (II)

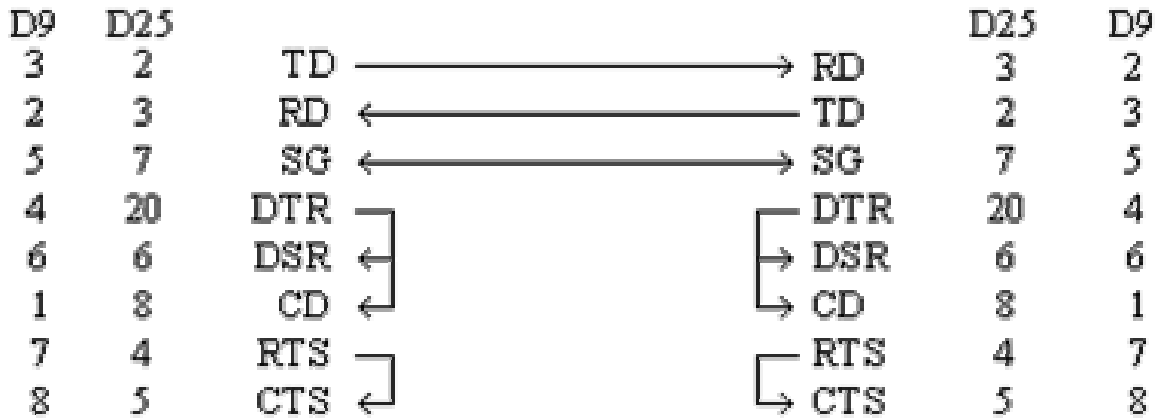
- **RS-232 Level Converters**
  - Transform the RS-232 levels back into 0 and 5 Volts.
  - The 1488 RS-232 Driver and the 1489 RS-232 Receiver.
    - Each package contains 4 inverters of the one type, either Drivers or Receivers.
    - The driver requires two supply rails, +7.5 to +15v and -7.5 to -15v.
    - Tthis may pose a problem in many instances where only a single supply of +5V is present.
  - Pinouts for the MAX-232 RS-232 Driver/Receiver ->
    - Includes a Charge Pump, which generates +10V and -10V from a single 5v supply.
    - Includes two receivers and two transmitters in the same package.





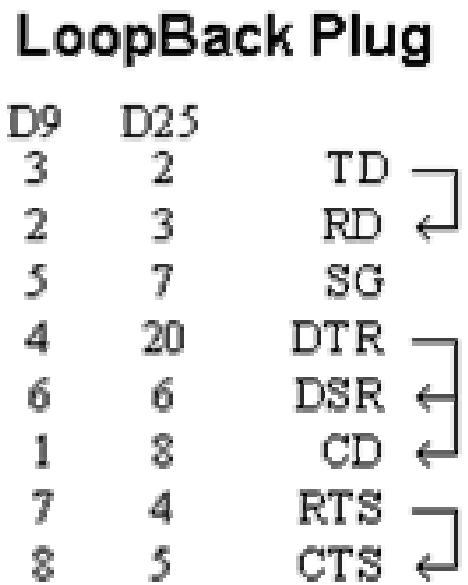
Serial Hardware (III)

- **Null Modems**
  - Used to connect two DTE's together.
  - A cheap way to network games or to transfer files between computers using Zmodem Protocol, Xmodem Protocol etc.
  - Null Modem Wiring Diagram



  - Only requires 3 wires (TD, RD & SG) to be wired: More cost effective
  - The aim is to make to computer think it is talking to a modem rather than another computer.

Serial Hardware (IV)



- **LoopBack Plug**
  - Loopback Plug Wiring Diagram ->
  - This loopback plug can come in extremely handy when writing Serial / RS232 Communications Programs.
    - It has the receive and transmit lines connected together, so that anything transmitted out of the Serial Port is immediately received by the same port.
    - If you connect this to a Serial Port an load a Terminal Program, anything you type will be immediately displayed on the screen.
  - *Note that this is not intended for use with Diagnostic Programs and thus will probably not work. For these programs you require a differently wired Loop Back plug which may vary from program to program.*

4. UARTs in Xscale

- UARTs in Xscale
  - Controlled via direct memory access (DMA) or programmed I/O.
  - The PXA255 processor has four UARTs:
    - Full Function UART (FFUART),
    - Bluetooth UART (BTUART),
    - Standard UART (STUART) and
    - Hardware UART (HWUART).
  - The UARTs use the same programming model.

UARTs in Xscale (II)

- **Feature List**
  - Functionally compatible with the 16550
  - Ability to add or delete standard asynchronous communications bits (start, stop, and parity) in the serial data
  - Independently controlled transmit, receive, line status, and data set interrupts
  - Programmable baud rate generator that allows the internal clock to be divided by 1 to (216–1) to generate an internal 16X clock
  - Modem control pins that allow flow control through software. Each UART has different modem control capability.
  - Fully programmable serial-interface:
    - — 5-, 6-, 7-, or 8-bit characters
    - — Even, odd, and no parity detection
    - — 1, 1.5, or 2 stop bit generation
    - — Baud rate generation up to 921 Kbps for the BTUART and HWUART. Up to 230 Kbps for other UARTs.

UARTs in Xscale (III)

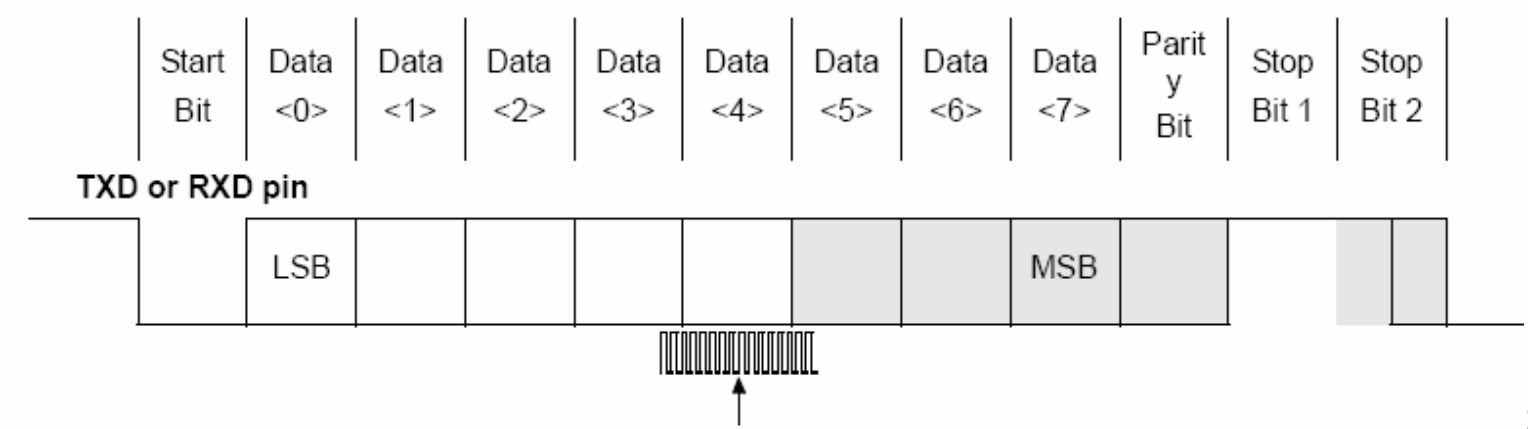
- Feature List (Cont'd)
  - 64-byte transmit FIFO
  - 64-byte receive FIFO
  - Complete status reporting capability
  - Ability to generate and detect line breaks
  - Internal diagnostic capabilities that include:
    - — Loopback controls for communications link fault isolation
    - — Break, parity, and framing error simulation
  - Fully prioritized interrupt system controls
  - Separate DMA requests for transmit and receive data services
  - Slow infrared asynchronous interface that conforms to the Infrared Data Association (IRDA) standard.

UARTs in Xscale (IV)

- Overview
  - **Full Function UART**
    - Supports modem control capability. The maximum tested baud rate on this UART is 230.4 kbps. The divisor programmed in the divisor latch registers must be equal to or greater than four.
  - **Bluetooth UART**
    - High speed UART that supports baud rates up to 921.6 kbps and can be connected to a Bluetooth module. It supports the functions in the feature list, but only supports two modem control pins (nCTS, nRTS).
  - **Standard UART**
    - Supports all functions in the feature list, but does not support modem control capability. The UART's maximum tested baud rate is 230.4 kbps. The divisors programmed in divisor latch registers must be equal to or greater than four.
  - **Hardware UART**
    - Contains a UART and a slow infrared transmit encoder and receive decoder that conforms to the IrDA Serial Infrared (SIR) Physical Layer Link Specification.

UARTs in Xscale (V)

- Operation
  - Receive data sample counter frequency is 16 times the value of the bit frequency.
    - The 16X clock is created by the baud rate generator.
    - Each bit is sampled three times in the middle.
  - Shaded bits are optional and can be programmed by software.
    - Each data frame is between seven and 12 bits long, depending on the size of the data programmed, whether parity is enabled, and the number of stop bits.



UARTs in Xscale (VI)

- Reset
  - The UARTs are disabled on reset.
  - To enable a UART, Software must program the GPIO registers, and then set IER[UUE].
  - When the UART is enabled, the receiver waits for a frame start bit and the transmitter sends data if it is available in the Transmit Holding Register.
  - Transmit data can be written to the Transmit Holding Register before the UART unit is enabled.
  - In FIFO mode, data is transmitted from the FIFO to the Transmit Holding Register before it goes to the pin.
  - When the UART unit is disabled, the transmitter or receiver finishes the current byte and stops transmitting or receiving more data.
  - Data in the FIFO is not cleared and transmission resumes when the UART is enabled.

UARTs in Xscale (VII)

- Internal registers
  - The state of the SLCR[DLAB] bit affects the selection of some UART registers.

UART Register Addresses (Base + offset)	DLAB Bit Value	Register Accessed
Base	0	Receive Buffer (read only)
Base	0	Transmit Buffer (write only)
Base + 0x04	0	Interrupt Enable (read/write)
Base + 0x08	X	Interrupt Identification (read only)
Base + 0x08	X	FIFO Control (write only)
Base + 0x0C	X	Line Control (read/write)
Base + 0x10	X	Modem Control (read/write)
Base + 0x14	X	Line Status (read only)
Base + 0x18	X	Modem Status (read only)
Base + 0x1C	X	Scratch Pad (read/write)
Base + 0x20	X	Infrared Selection (read/write)
Base	1	Divisor Latch Low (read/write)
Base + 0x04	1	Divisor Latch High (read/write)

UARTs in Xscale (VIII)

- Divisor Latch Register (DLH & DLL)
  - BaudRate = 14.7456 MHz / (16*Divisor)
- Line Control Register
  - Bit 7: DLAB — Divisor Latch Access Bit
  - Bit 6: SB — Set Break
  - Bit 5: STKYP — Sticky Parity
  - Bit 4: EPS — Even Parity Select
  - Bit 3: PEN — Parity Enable
  - Bit 2: STP — Stop bits
  - Bit1-0: WLS — Word length select
- Line Status Register
  - Bitas 31-8: Reserved
  - Bit 7: FIFOE — FIFO Error
  - Bit 6: TEMT — Transmitter Empty
  - Bit 5: TDRT — Transmit Data Request
  - Bit 4: BI — Break Interrupt
  - Bit 3: FE — Framing error
  - Bit 2: PE — Parity error
  - Bit 1: OE — Overrun error
  - Bit 0: DR — Data Ready

References

- Serial interface, RS232C, and Serial hardware
  - Search Internet
- UARTs in Xscale
  - PXA255 Developer's Manual, http://developer.intel.com