8250/14450/16550 UART INTERNAL ARCHITECTURE

There are 10 Internal Regs

1) Rx Buffer Reg: Rx reg. (OFFSET 0; DLAB=0)

Tx Buffer Reg (Holding Reg): Tx Reg.

2) Interrupt Enable Reg (R/W; OFFSET 1)

D0: RDA
D1: TBE
D2: Rx error or Break. Framing, Overrun, and parity error.
D3: RS 232 Input. Interrupt occurs when any RS 232 input changes state.
D4-d7: 0000

2) Interrupt Identification (OFFSET 2; R only)

Used to identify the exact source of interrupt. D3 indicates that pending Tx or Rx interrupt was generated by timeout.

D3 D2 D1 D0	Priority	Interrupt ID
0001	None	None
0110	0 (HI)	Highest priority. serialization error (framing,, parity, overrun) or Break
0100	1	RDA/FIFO Rx Trigger level reached (like RDA for FIFO)
1100	1	FIFO Timeout. (No char removed from or input to Rx FIFO during last character times & there is at least 1 char during this time.)
0010	2	TBE
0000	3 (LO)	RS 232 Input (CTS, DSR, RI, DCD)

4) FIFO Control Register (OFFSET 2; W only)

D0: FIFO Enable. Enables operation of both FIFOs. A zero clears all bytes in FIFOs.
D1: Rx FIFO Reset. Clears all bytes from Rx FIFO, but NOT the Rx shift reg.
D2: Tx FIFO Reset. Clears all bytes from Tx FIFO, but NOT the Tx shift reg.
D3: DMA Mode Select. Supports DMA Operations
D4, D5: Reserved
D7 D6: Receive Trigger Level.
00= level 1;
01= level 4;
10= level 8;
11= level 16

5) Line Control Register (OFFSET 3; R/W)

Used to specify SDU (Serial Data Unit) Format

D1, D0: # OF DATA BITS
00=5
01=6
10=7
11=8

D2: # OF STOP BITS. 0= 1 Stop bit; 1= 2 stop bits.
1 ½ stop bits are automatically selected if 5 data bits are used.

D5, D5, D3: PARITY
000 = NO Parity
001 = Odd
011 = Even
101 = Mark. Parity fixed to 1 independent of data
111 = Space. Parity fixed to 0 independent of data

Note: What errors can mark and space parity detect?

D6: BREAK. 0 = Break OFF; 1 = Break ON. A 1 on this bit forces Tx (i.e. SOUT pin) to a logic 0 state. The Transmitter remains in this state until a 0 is written to this bit. What is the use of this?

D7: DLAB. This bit is actually used as MSB of Address. Why? There are 10 internal regs but only three ext. address lines to UART.
For eg. When DLAB=1, Offset 0 and 1 are used for LSB and MSB of Baud Rate Divisor Latch respectively.

6) Modem Control Register (OFFSET 4; R/W)

D0: Asserts DTS
D1: Asserts RTS
D3, D2 = GPO2, GPO1 are User Definable Outputs. Forces OUT2 and OUT1 to a specific state. In the current PCs Bit D3 must be set to Enable Interrupts. Ie. It serves as Master Interrupt Enable. Why?

D4: Local Loopback. Forces the following.

a) Tx output (SOUT) = 1 (Marking)
b) Rx Input (SIN0 is disconnected.
c) The output of Tx shift reg is connected to input of Rx shift reg.
d) RS232 pins are directly connected as follows.

/CTS to /RTS
/DSR to /DTR
/DCD to /GPO2
/RI to /GPO1

What really is the use of Loopback in practice?

7) Line Status Register (OFFSET 5; R/W)

D0: RDA. I.e. receive Data Available.
D1: Overrun Error
D2: Parity Error
D3: Framing Error
D4: BREAK detect. This is set if Rx detects a spacing lasting longer than one SDU.

D5: Transmit Buffer Empty (TBE)

D6: Tx Empty (TXE): Think of this bit as "all bytes sent" flag. Before termination Xmission, consult this bit to avoid leaving one in the pipe.

8) Modem Status (OFFSET 6; R/W)

Bits 0…3 report change of state since the last read
Bits 4…7 report absolute state eg. 1 = H. The bits take on special meaning during loopback testing.

D0: Delta CTS
D1: Delta DSR
D2: Delta RI
D3: Delta DCD
D4: CTS
D5: DSR
D6: RI
D7: DCD

Note: During loopback testing (D4 of MCR) these bits report current state of RS232 outputs as follows
D4: CTS = RTS
D5: DSR = DTS
D6: RI = GPO2
D7: DCD = GPO1

9) Baud Rate Divisor Latch (Offset 0; DLAB=1)

0/8: LSB of BRDL
1/9: MSB of BRDL

When DLAB = 1 these move from TxD and RxD to Baud Rate Divisor Latch.

The divisor for a specific baud rate is derived from the formula

Divisor = (Ref Clk Frequency) / (16 X Desired Baud Rate)

For eg. With a clk of 1.8432 MHz, baud Rate 28.8K needs 0004 divisor.

In general, baud rate = Ref Clk Freq/16 * Divisor
= 1.8432 X 10^ 6 /divisor
= 115200/divisor

INTERRUPT ISSUES

This UART issues interrupts but does not supply a vector. Presumable, the vector must be supplied by 8259 PIC

When UART issues interrupts, all equal and low priority interrupts are locked out

FIFO INTERRUPT MODE OF OPERATIUON

When FIFO & Rx Interrupts are enabled, Rx interrupts occur as follows:

1. RDA interrupts is issued to MPU when FIFO has reached its programmed trigger level; it is cleared as soon as it drops below triggered program level

2. Interrupts ID ref RDA indication also occurs when FIFO trigger level is reached

3. As in polled mode, data rdy bit (line status reg bit 0) is set as soon as a single char is transferred from shift reg to RTXC FIFP. It is cleared when FIFO is empty.

When FIFO & Rx Interrupts are enabled, Rx FIFO Timeout interrupts occur as follows:

A FIFO T.O. interrupts occur under conditions of

- >= 1 Char is in FIFO
- Most recent char was received longer than 4 char times ago.
- Most recent CPOU read of the FIFO occurred longer that 4 char times ago

TRANSMIT INTERRUPT OPERATION

1) The Tx holding reg interrupt occurs when the Tx FIFO is empty.
2) A single spurious interrupt is often generated immediately after Tx interrupts are enabled.

PROGRAMMING ASPECTS OF 8250/16550 UART

- Used in current IBM PCs as COMx Ports
- The Base address and IRQ channels are

COMx	BASE ADDRESS	IRQ CHANNELS
COM1	3f8h	IRQ4
COM2	2F8H	IRQ3
COM3	3E8H	IRQ4
COM4	2E8H	IRQ3

Programming EXAMPLES By DIRECT I/O

EQUATES

```
LSR    EQU   3FDH
TXRX EQU   3F8H
RDA    EQU   01H
TBE    EQU   20H
```

EXAMPLE 1: POLLING PROGRAM TO READ FROM COM1 PORT

```
LOOP: IN      AL,     LSR     ; read LSR
              TEST    AL,     RDA    ;test RDA
              JZ      LOOP    ; loop if not RDA
              IN      AL,     TXRX ; read data byte
```

EXAMPLE 2: POLLING PROGRAM TO write to COM1 PORT

```
LOOP: IN      AL,     LSR     ; read LSR
              TEST    AL,     TBE    ;test TBE
              JZ      LOOP    ; loop if not RDA
              MOV     AL,     'a'
              OUT     TXRX,AL      ; WRITE data byte to port
```

Note: Difference between TEST and AND is that with TEST the AL value does not change.

BIOS/DOS FUNCTION CALLS FOR SERIAL PORTS

DOS (limited) INT 21h

Function (in AH)	Operation
03h	Read char
04h	Output char
3fh	Read file/device. *Can be used to write a string rather than a ch*ar.
40h	Write file/device. *Can be used to write a string rather than a char.*

BIOS INT 14h

Function (in AH)	operation	Input condition	Output condition
00	Initialize	AL = parameter; DX= port # (0=COM1, 1=COM2, etc)	AH = port status AL = modem status
01	Write char	AL = char; DX= port #	AH b7 =0 if successful, =1 if not; AH b0..b6 = status if successful; AL = char
02	Read Char	DX = port#	AH b7 =0 if successful, =1 if not; AH b0..b6 = status if successful; AL = char read
03	Read Port Status	DX = port#	AH = port status AL = modem status

Note 1: Parameter Byte in AL is defined as
B7b6b5 : baud rate; 000 = 110; … ; 111 = 9600
B4b3: parity (01=odd, 11=even; x0=none)
B2: stop bits (0=1, 1=2)
B1b0: word length (10=7bits, 11=8bits)

Note 2: The Port Status returned in AH is defined as follows

B7: Timed out
B6: Tx shift reg empty
B5; Tx holding reg empty
B4; Break detected
B3: Framing Error detected
B2: parity Error detected
B1; Overrun error detected
B0: Rx Data Ready

Note 3: Modem status returned in AL

B7: Rx signal line detect (CD)
B6: RI
B5: DSR
B4: CTS
B3: change in CD
B2: trailing edge RI
B1: change in DSR
B0: change in CTS

Programming EXAMPLES By SOFTWARE INTERRUPTS

```
; Equates
COM1PORT  EQU   0
COM2PORT  EQU   1
TBE       EQU   20H
RDA       EQU   01H

;EXAMPLE 1: PROGRAM TO write to COM1 PORT

;Check TBE Status First
LOOPW:    MOV   DX,    COM1PORT
          MOV   AH,    03H
          INT   14H
          TEST  AH,    TBE
          JZ    LOOPW
;Now Write
          MOV   DX,    COM1PORT
          MOV   AH,    01H
          MOV   AL,    'a'
          INT 14H


;EXAMPLE 2: PROGRAM TO READ FROM COM1 PORT


;Check RDA Status First
LOOPR:    MOV   DX,    COM1PORT
          MOV   AH,    03H
          INT   14H
          TEST  AH,    RDA
          JZ    LOOPR
;Now Read
          MOV   DX,    COM1PORT
          MOV   AH,    02H
          INT 14H
```

C Programming EXAMPLES By DIRECT I/O AND POLLING

```
#define     COM1BASE  0x3F8
#define     COM2BASE  0x2F8
#define     LSR       COM1BASE + 5
#define     TBE       0x20
#define     RDA       0x01

// Assume Loopback Mode
main()
{
char    charW;
char    charR;

charW = 'a'

//to write, first check see if TBE

while ((inp(LSR) & TBE) != TBE);
outp(COM1BASE, charW);

//to read, first check to see if RDA

while ((inp(LSR) & RDA) != RDA);
charR = inp(COM1BASE);
}
```