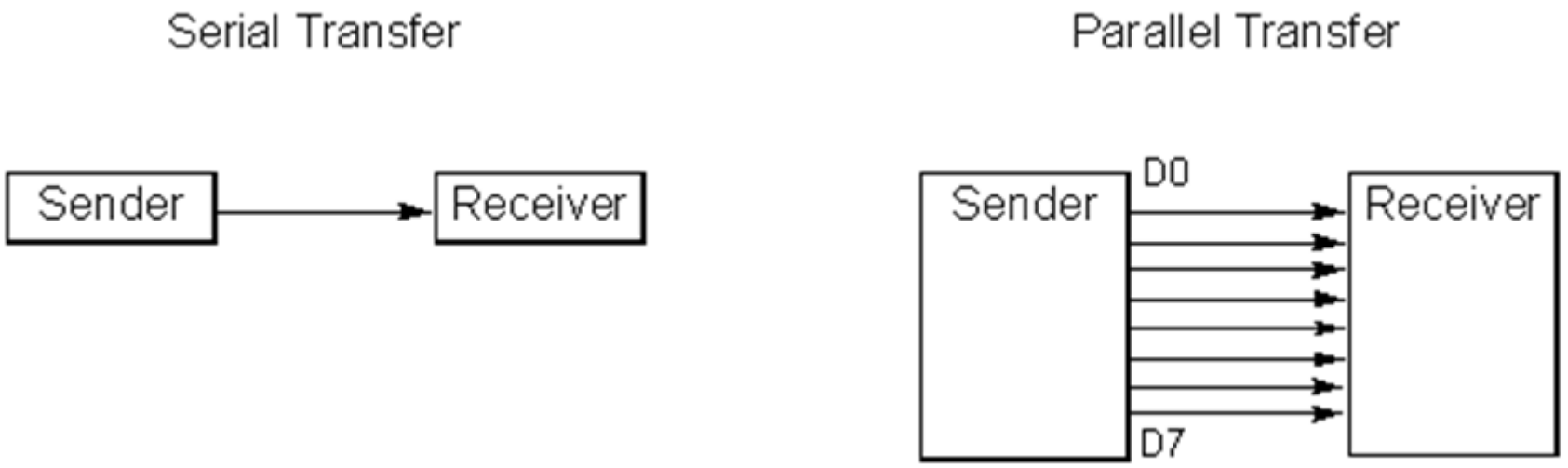
PIC18 Serial Port

Hsiao-Lung Chan
Dept Electrical Engineering
Chang Gung University, Taiwan
chanhl@mail.cgu.edu.tw

Objectives

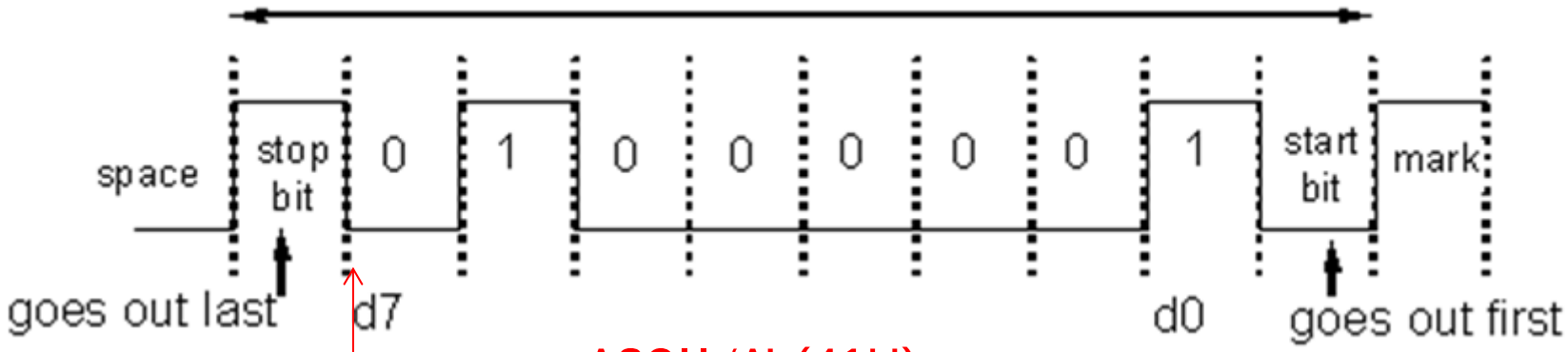
- Operation of serial port
- Transmit a series of bits
- Receive a series of bits

Serial vs. parallel data transfer



Data framing in asynchronous serial communication

- 8 ASCII bits and 1 stop bit



ASCII 'A' (41H)

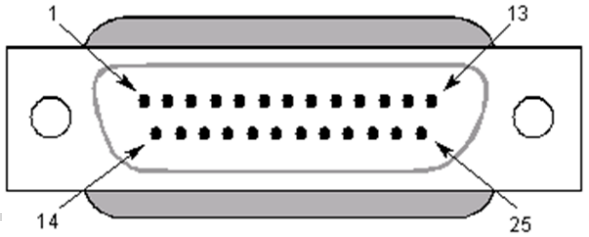
Parity bit
- odd: '1' is added for 'A'
- or even: '0' is added for 'A'

(Figure 10-3)

Data transfer rate (baud rate)

- Early IBM PC/XT was 100 ~ 9600 bps (bits per second)
- Pentium-based PC is as high as 56K bps
- Generally limited to 100K bps

RS232 connectors



DB-9

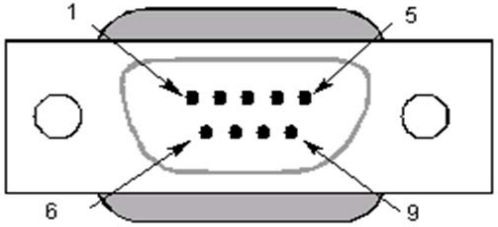


Table 10-2: IBM PC DB-9 Signals

Pin	Description
1	Data carrier detect (DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (DSR)
7	Request to send (RTS)
8	Clear to send (CTS)
9	Ring indicator (RI)

Table 10-1: RS232 Pins (DB-25)

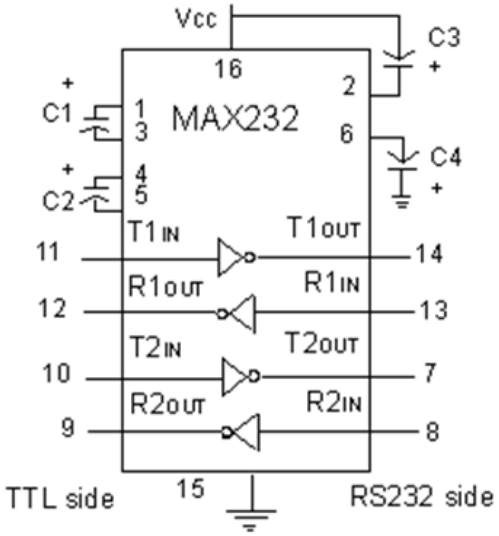
Pin	Description
1	Protective ground
2	Transmitted data (TxD)
3	Received data (RxD)
4	Request to send (RTS)
5	Clear to send (CTS)
6	Data set ready (DSR)
7	Signal ground (GND)
8	Data carrier detect (DCD)
9/10	Reserved for data testing
11	Unassigned
12	Secondary data carrier detect
13	Secondary clear to send
14	Secondary transmitted data
15	Transmit signal element timing
16	Secondary received data
17	Receive signal element timing
18	Unassigned
19	Secondary request to send
20	Data terminal ready (DTR)
21	Signal quality detector
22	Ring indicator
23	Data signal rate select
24	Transmit signal element timing
25	Unassigned

(Figures 10-4, 10-5)

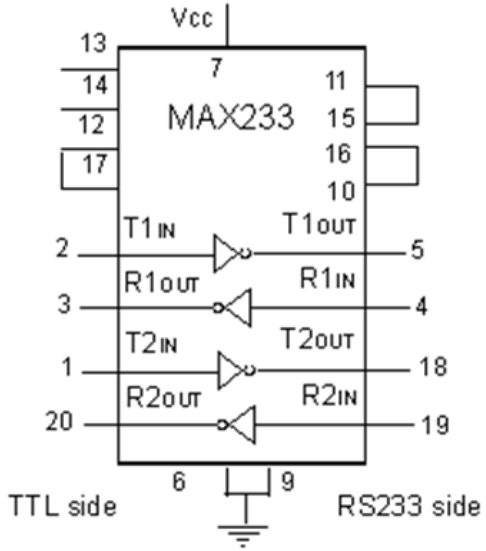
RS232 standard

- RS232 voltage levels
  - “1” : -3 ~ -25 V
  - “0” : 3 ~ 25 V
- Line driver (voltage converter)
  - Convert RS232 voltage levels to TTL voltage levels

MAX232

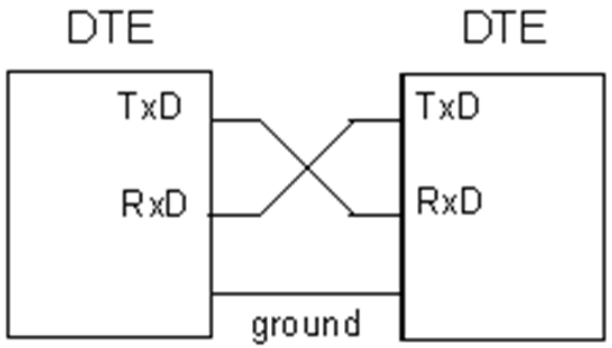


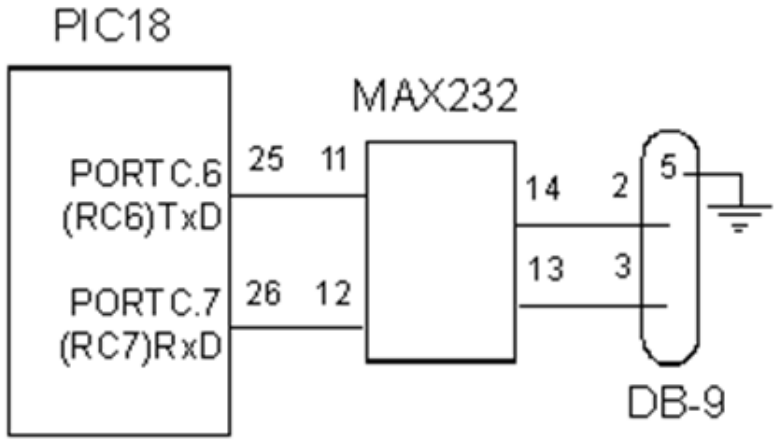
MAX233

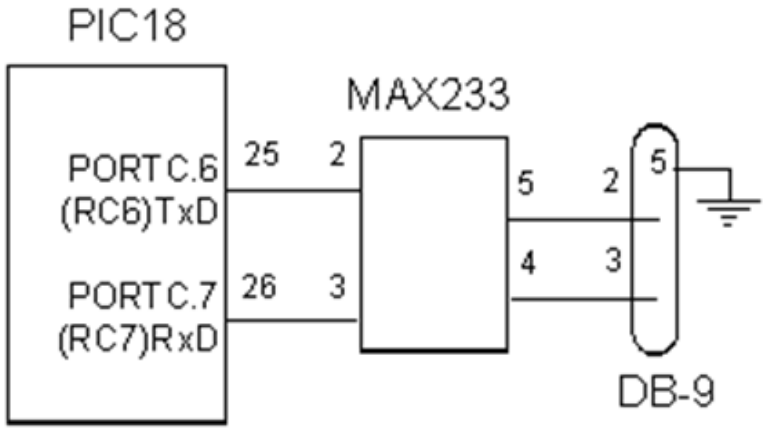


(Figures 10-7, 10-8)

PIC18 connection to RS232







40-Pin DIP Package PIC18

(Figures 10-7, 10-8)

PIC18 UART (universal asynchronous receiver/transmitter) programming

- SPBGR (serial port baud rate generator)
- TXREG (transfer register)
- RCREG (receiver register)
- TXSTA (transmit status and control register)
- RCSTA(receive status and control register)
- PIR1 (peripheral interrupt request register1)

SPBRG register and baud rate

- Desired baud rate = Fosc/(4x16x(Y+1))
  - Fosc/4 instruction cycle

- The value for SPBGR Y = Fosc/64/BaudRate -1

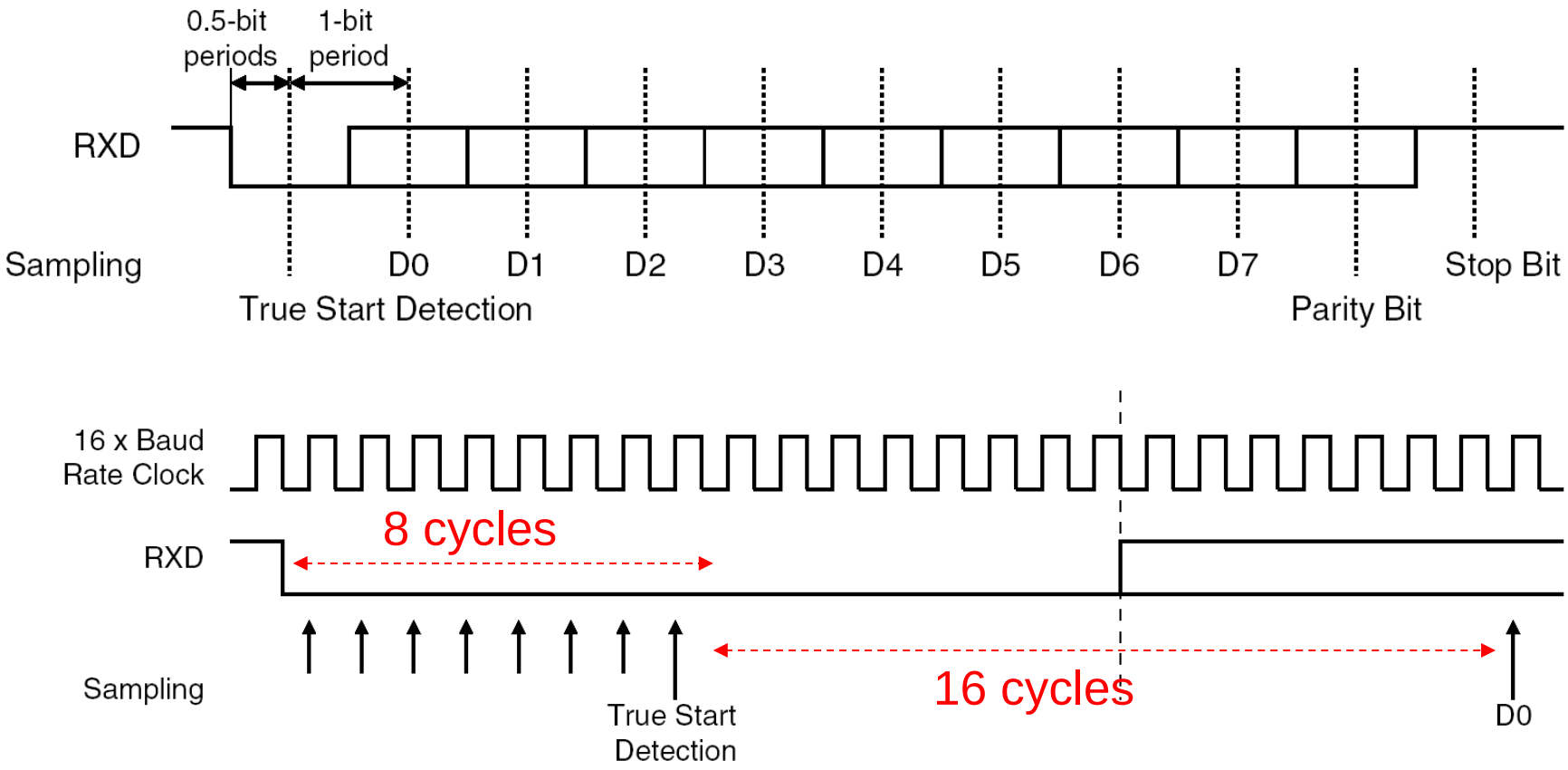
Table 10-4: SPBRG Values for Various Baud Rates (Fosc = 10 MHz, BRGH = 0)

Baud Rate	SPBRG (Decimal Value)	SPBRG (Hex Value)
38400	3	3
19200	7	7
9600	15	F
4800	32	20
2400	64	40
1200	129	81

Note: For Fosc = 10 MHz we have SPBRG = (156,250/BaudRate) – 1

Start bit detection in asynchronous Receiver

- Example: 8-bit, parity enabled 1 stop



Tim Wilmshurst, An Introduction to the Design of Small-Scale Embedded Systems, Palgrave, 2001.

TXSTA: Transmit status and control register

CSRC	TX9	TXEN	SYNC	0	BRGH	TRMT	TX9D

CSRC D7 Clock Source Select (not used in asynchronous mode, therefore D7 = 0.)

TX9 D6 9-bit Transmit Enable
1 = Select 9-bit transmission
0 = Select 8-bit transmission (We use this option, therefore D6 = 0.)

TXEN D5 Transmit Enable
1 = Transmit Enabled
0 = Transmit Disabled
We turn "on" and "off" this bit in order to start or stop data transfer.

SYNC D4 USART mode Select (We use asynchronous mode, therefore D4 = 0.)
1 = Synchronous
0 = Asynchronous

0 D3

BRGH D2 High Baud Rate Select
0 = Low Speed (Default)
1 = High Speed
We can double the baud rate with the same Fosc. See the end of this section for further discussion on this bit.

TRMT D1 Transmit Shift Register (TSR) Status
1 = TSR empty
0 = TSR full

TXD9 D0 9th bit of Transmit Data (Because we use the 8-bit option, we make D0 = 0)
Can be used as an address/data or a parity bit in some applications

(Figure 10-9)

RCSTA: Receive status and control register

SPEN	RX9	SREN	CREN	ADDE	FERR	OERR	RX9D

SPEN D7 Serial port enable bit
1 = Serial port enabled, which makes TX and RX pins as serial port pins
0 = Serial port disabled

RX9 D6 9-bit Receive enable bit
1 = Select 9-bit reception
0 = Select 8-bit reception (We use this option; therefore, D6 = 0.)

SREN D5 Single receive enable bit (not used in asynchronous mode D5 = 0)

CREN D4 Continuous receive enable bit
1 = Enable continuous Receive (in asynchronous mode)
0 = Disable continuous Receive (in asynchronous mode)

ADDEN D3 Address delete enable bit (Because used with the 9-bit data frame D3 = 0)

FERR D2 Framing error bit
1 = Framing error
0 = No Framing error

OERR D1 Overrun error bit
1 = Overrun error
0 = No overrun error

TXD9 D0 9th bit of Receive data (Because we use the 8-bit option, we make D0 = 0)
Can be used as an address/data or a parity bit in some applications.

(Figure 10-10)

PIR1 (interrupt control register 1)

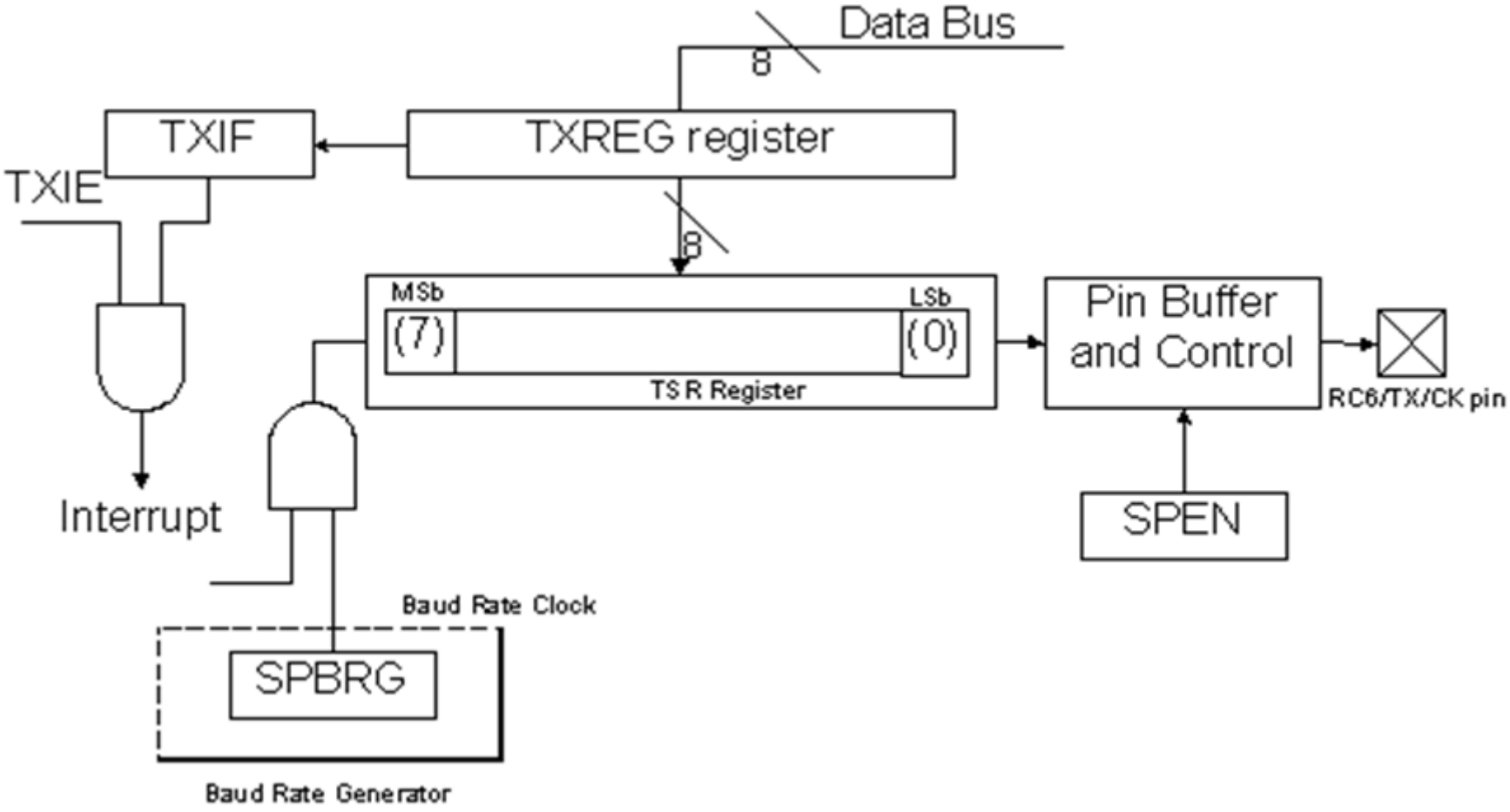
---	--	RCIF	TXIF	--	---	--	

RCIF Receive interrupt flag bit
1 = The UART has received a byte of data and it is sitting in the RCREG register (receive buffer), waiting to be picked up.
Upon reading the RCREG register, the RCIF is cleared to allow the next byte to be received.
0 = The RCREG is empty.

TXIF Transmit interrupt flag bit
0 = The TXREG register is full.
1 = The TXREG (transmit buffer) register is empty.

(Figure 10-11)

Simplified USART transmit block diagram



(Figure 10-12)

Quadrupling the baud rate

- Baud rate for BRGH = 0
  = Fosc/(4x**16**x(Y+1))
- Baud rate for BRGH = 1
  = Fosc/(4x**4**x(Y+1))

Table 10-7: SPBRG Values for Various Baud Rates (XTAL = 10 MHz)

	BRGH = 0	BRGH = 1
Baud Rate	**SPBRG (Decimal)**	**SPBRG (Decimal)**
57600	2	10
38400	3	15
19200	7	32
9,600	15	64
4,800	32	129
SPBRG = (156250/Baud rate) – 1		SPBRG = (625000/Baud rate) – 1

Examples for serial port programming

- Assembly programming
  - Transmit the message “YES” serial (Example 10-3)
  - Receive bytes of data serially (Example 10-4)
- C programming
  - Transmit the letter ‘G’ serial (Example 10-11)
  - Receive bytes of data serially (Example 10-13)

Reference

- M.A. Mazidi, R.D. Mckinlay, D Causey, PIC Microcontroller and Embedded Systems Using Assembly and C for PIC18, Pearson Education Inc., 2008.