CENG 336
INT. TO EMBEDDED SYSTEMS DEVELOPMENT

Spring 2006
Recitation 09

OUTLINE

- USART Communication on PIC
- Introduction to PIC18 Series

Introduction

PIC16F877 has two serial I/O modules:

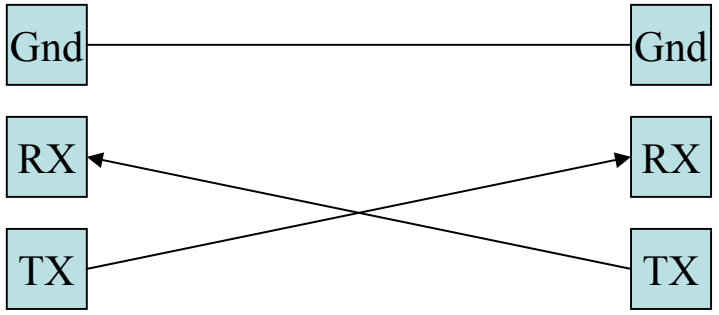
- The Universal Synchronous Asynchronous Receiver Transmitter (USART) module
  - **Asynchronous (full duplex)**
  - Synchronous - Master (half duplex)
  - Synchronous - Slave (half duplex)
- Master Synchronous Serial Port (MSSP)
  - SPI mode
  - I2C mode

USART

- Universal Synchronous and Asynchronous serial Receiver and Transmitter
- A standard I/O device that provides conversions between serial and parallel data

Asynchronous Serial Communication

- For asynchronous communication, only three lines are needed
  - Common ground
  - Transmit data (TX)
  - Receive data (RX)



- Full-duplex
  - Note that the transmit line on one end is the receive line on the other end

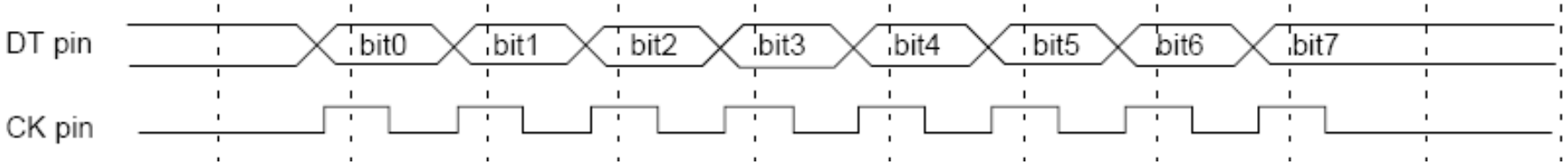
Asynchronous Serial Communication – Cont.

- The transmit line **idle** signal is 1 (**mark**)
- The transmitter signals the **start** of a **frame** by asserting a 0 (**space**) on TX
- At regular intervals (the baud rate), the transmitter asserts the bits of data, least significant first
- One or two **stop** bits (1) terminate the frame

Synchronous Serial Communication

- For synchronous communication, only three lines are needed
  - Common ground
  - Data (DT)
  - Clock(CK)

- Half-duplex
  - Transmission and reception do not occur at the same time.



USART Asynchronous Mode

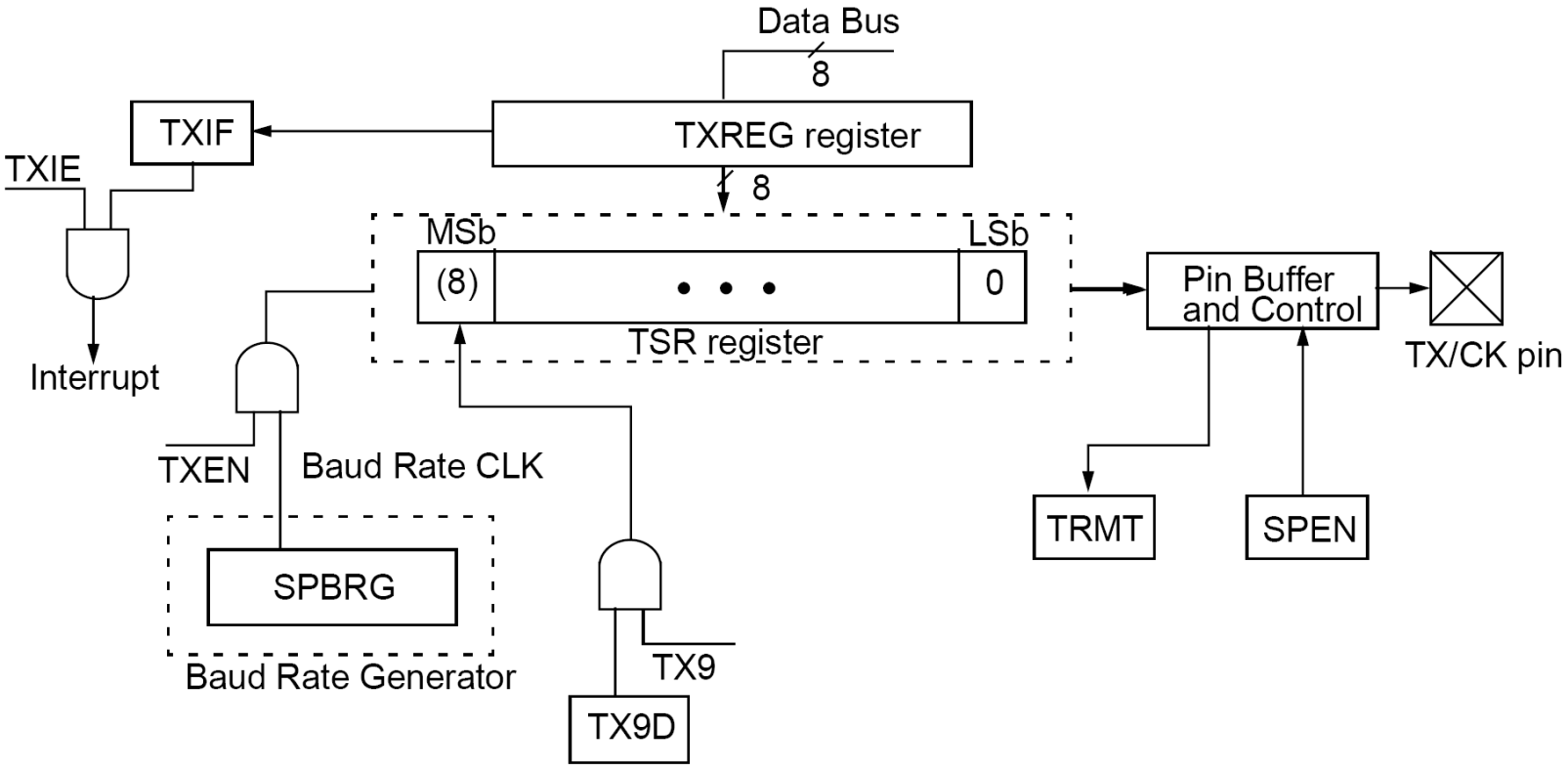
- In this mode, the USART uses standard non-return-to-zero (NRZ) format (one start bit, eight or nine data bits and one stop bit).

- The USART transmits and receives the LSb first.

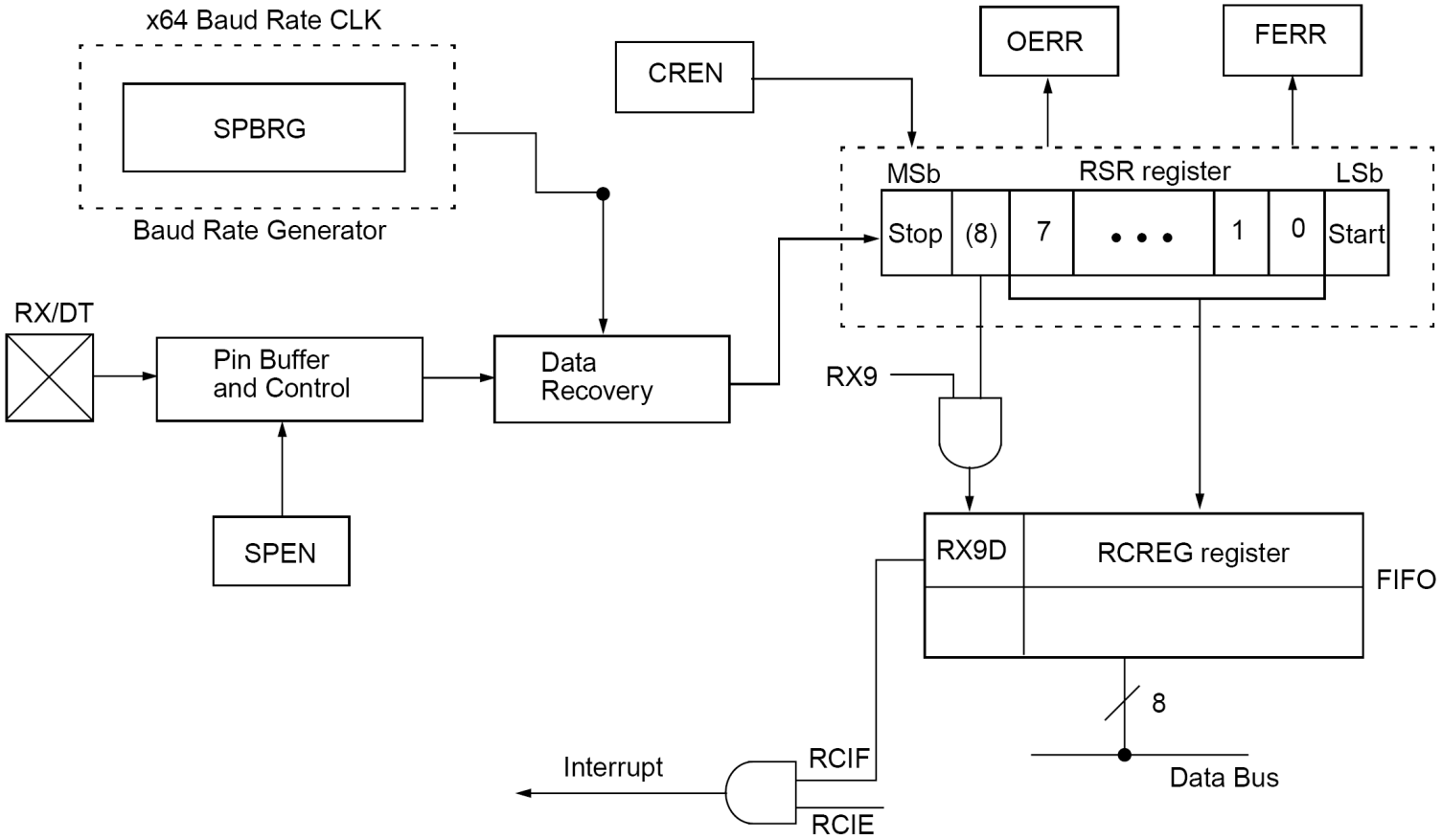
- Parity is not supported by the hardware, but can be implemented in software (stored as the ninth data bit).

- Asynchronous mode is stopped during SLEEP.

Transmitter Block Diagram



Receiver Block Diagram



TXSTA Register

REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC:** Clock Source Select bit

Asynchronous mode:
Don't care

Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled

Note: SREN/CREN overrides TXEN in SYNC mode.

bit 4 **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **Unimplemented:** Read as '0'

bit 2 **BRGH**: High Baud Rate Select bit

Asynchronous mode:
1 = High speed
0 = Low speed

Synchronous mode:
Unused in this mode

bit 1 **TRMT**: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data, can be parity bit

RCSTA Register

REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

bit 7 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)
0 = Serial port disabled

bit 6 **RX9**: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN**: Single Receive Enable bit
Asynchronous mode:
Don't care
Synchronous mode - master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode - slave:
Don't care

bit 4 **CREN**: Continuous Receive Enable bit
Asynchronous mode:
1 = Enables continuous receive
0 = Disables continuous receive
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received, and ninth bit can be used as parity bit

bit 2 **FERR**: Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR**: Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** 9th bit of Received Data (can be parity bit, but must be calculated by user firmware)

USART Baud Rate Generator

- SPBRG register controls the period
- In asynchronous mode BRGH (TXSTA<2>) also controls the baud rate. In synchronous mode bit BRGH is ignored.

Table 18-5: Baud Rates for Asynchronous Mode (BRGH = 1)

BAUD RATE (Kbps)	Fosc = 20 MHz			16 MHz			1
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAU
9.6	9.615	+0.16	129	9.615	+0.16	103	9.615
19.2	19.230	+0.16	64	19.230	+0.16	51	18.939
38.4	37.878	-1.36	32	38.461	+0.16	25	39.062
57.6	56.818	-1.36	21	58.823	+2.12	16	56.818
115.2	113.636	-1.36	10	111.111	-3.55	8	125
250	250	0	4	250	0	3	NA
625	625	0	1	NA	-	-	625
1250	1250	0	0	NA	-	-	NA

BAUD RATE (Kbps)	Fosc = 5.0688 MHz			4 MHz			3.579545 MHz		SPI
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	va (dec
9.6	9.6	0	32	NA	-	-	9.727	+1.32	22
19.2	18.645	-2.94	16	1.202	+0.17	207	18.643	-2.90	11
38.4	39.6	+3.12	7	2.403	+0.13	103	37.286	-2.90	5
57.6	52.8	-8.33	5	9.615	+0.16	25	55.930	-2.90	3
115.2	105.6	-8.33	2	19.231	+0.16	12	111.860	-2.90	1
250	NA	-	-	NA	-	-	223.721	-10.51	0
625	NA	-	-	NA	-	-	NA	-	-
1250	NA	-	-	NA	-	-	NA	-	-

The SPEN bit (RCSTA<7>), and the TRIS bits, have to be set in order to configure the RC6/TX/CK and RC7/RX/DT pins for the USART.

USART Asynchronous Transmission

Steps to follow when setting up a Asynchronous Transmission:
1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set the BRGH bit.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are desired, then set the TXIE, GIE and PEIE bits.
4. If 9-bit transmission is desired, then set the TX9 bit.
5. Enable the transmission by setting the TXEN bit, which will also set the TXIF bit.
6. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
7. Load data to the TXREG register (starts transmission).

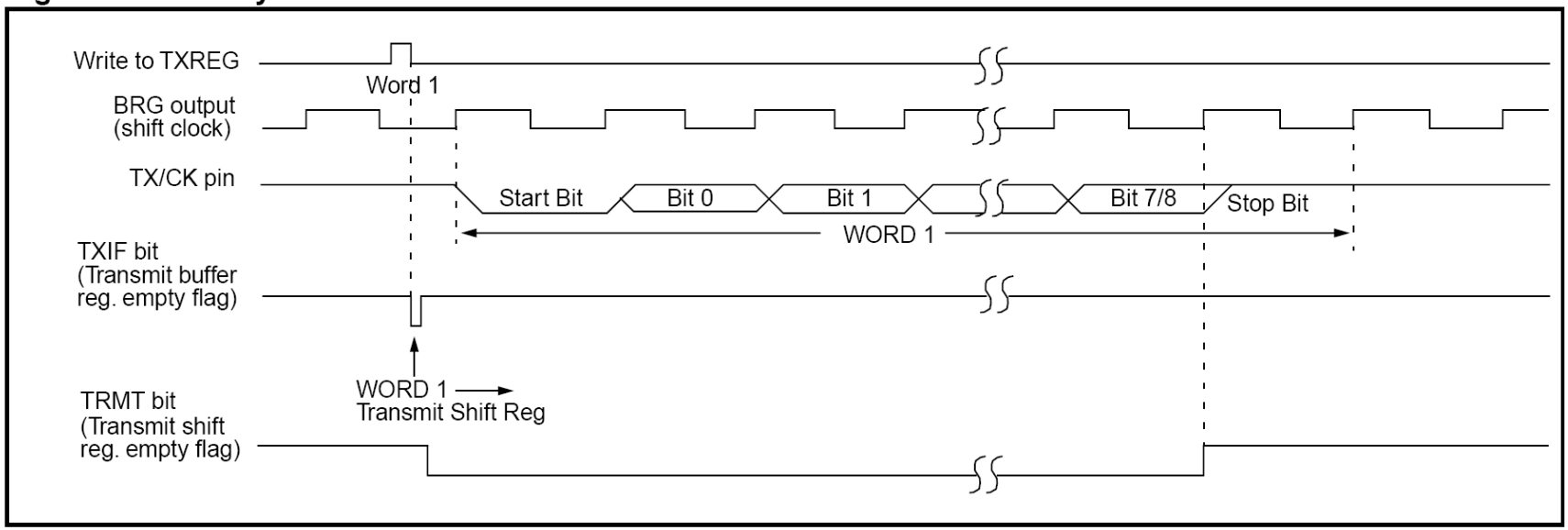
USART Asynchronous Reception

Steps to follow when setting up an Asynchronous Reception:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH.
2. Enable the asynchronous serial port by clearing the SYNC bit, and setting the SPEN bit.
3. If interrupts are desired, then set the RCIE, GIE and PEIE bits.
4. If 9-bit reception is desired, then set the RX9 bit.
5. Enable the reception by setting the CREN bit.
6. The RCIF flag bit will be set when reception is complete and an interrupt will be generated if the RCIE bit was set.
7. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing the CREN bit.

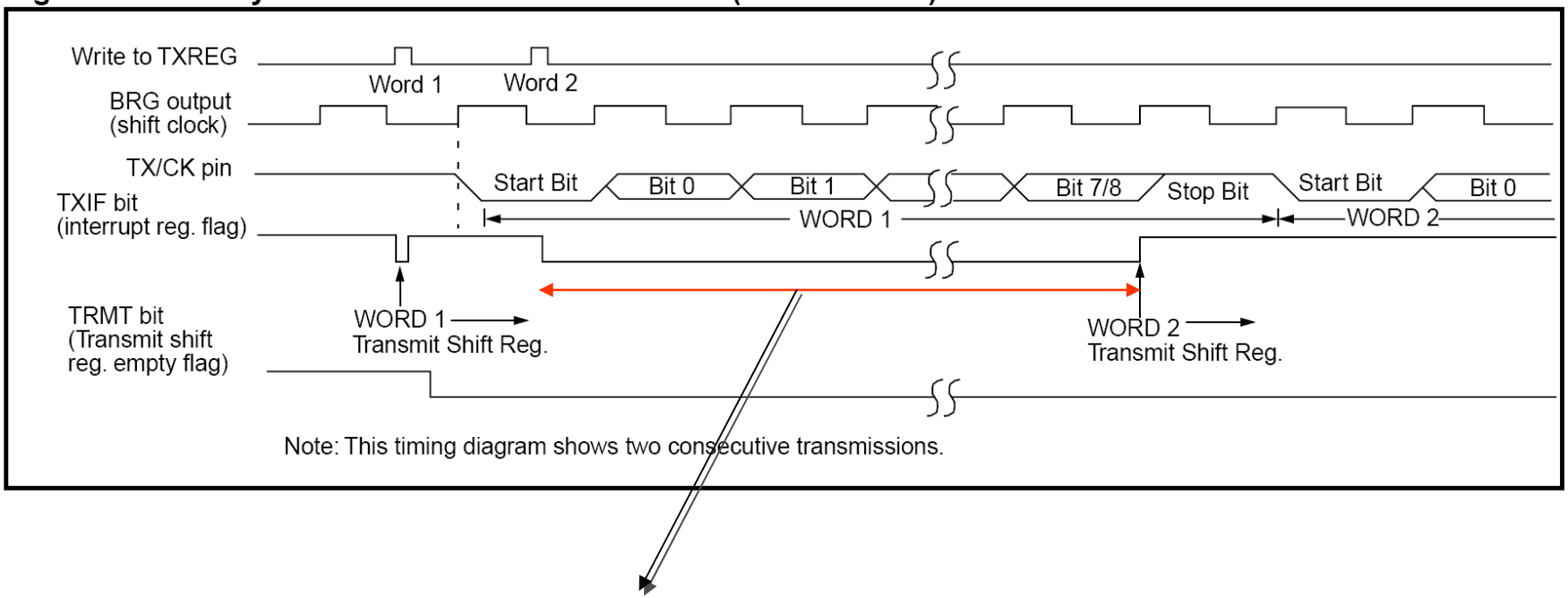
Timing Diagrams [1/3]

Figure 18-2: Asynchronous Master Transmission



Timing Diagrams [2/3]

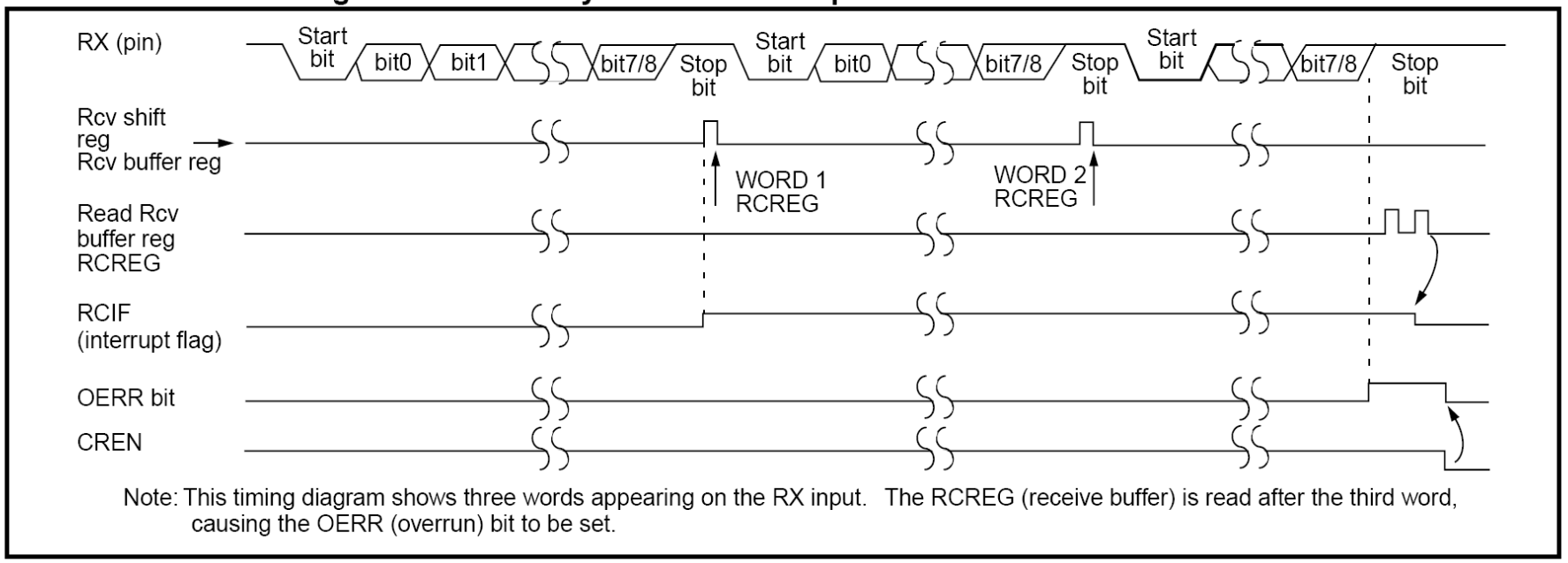
Figure 18-3: Asynchronous Master Transmission (Back to Back)



A third write to TXREG overwrites Word2

Timing Diagrams [3/3]

Figure 18-5: Asynchronous Reception



Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

Introduction to PIC18 Series

Features:

- Increased flash (16k words for the PIC18F4520)
- Increased RAM (1536 bytes for the PIC18F4520)
- Extended configuration bits
- Much deeper call stack (31 levels deep)
- The call stack may be read and written
- Two levels of interrupt priority
- Enhanced memory management (can acess the SFRs without switching banks; but not the GPRs)

Introduction to PIC18 Series

Features:

- New instructions
- 8x8 hardware multiplier
- Indexed addressing mode (PLUSW)
- Extending the FSR registers to 12 bits, allowing them to linearly address the entire data address space
- C has overtaken assembly
- We will use MPLAB C18 Student Edition.