EE-222: Microprocessor Systems

AVR Serial Communication

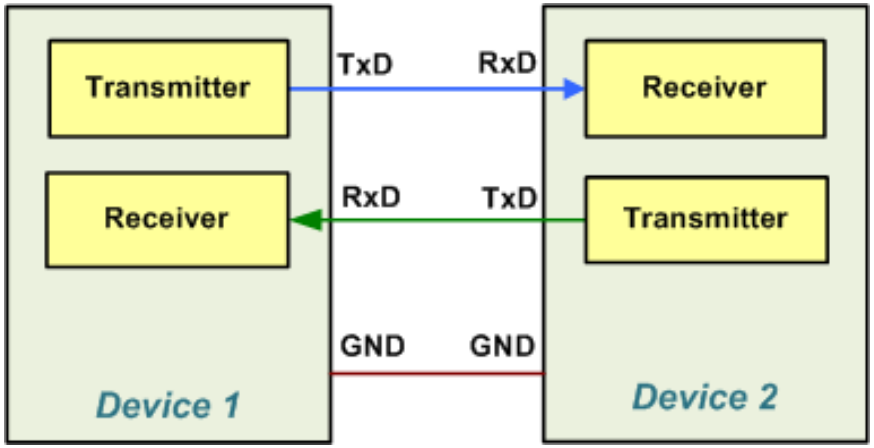
Instructor: Dr. Arbab Latif

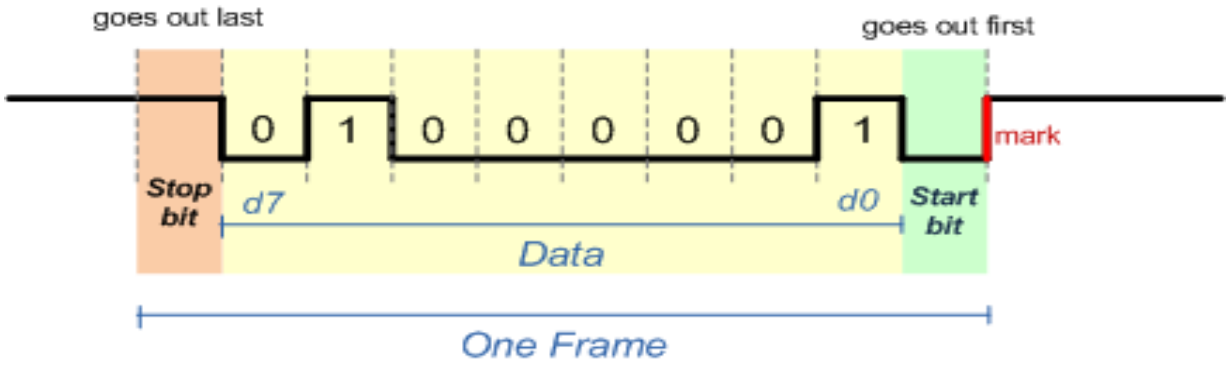
NUST

SCHOOL OF ELECTRICAL ENGINEERING &
COMPUTER SCIENCE (SEECS)

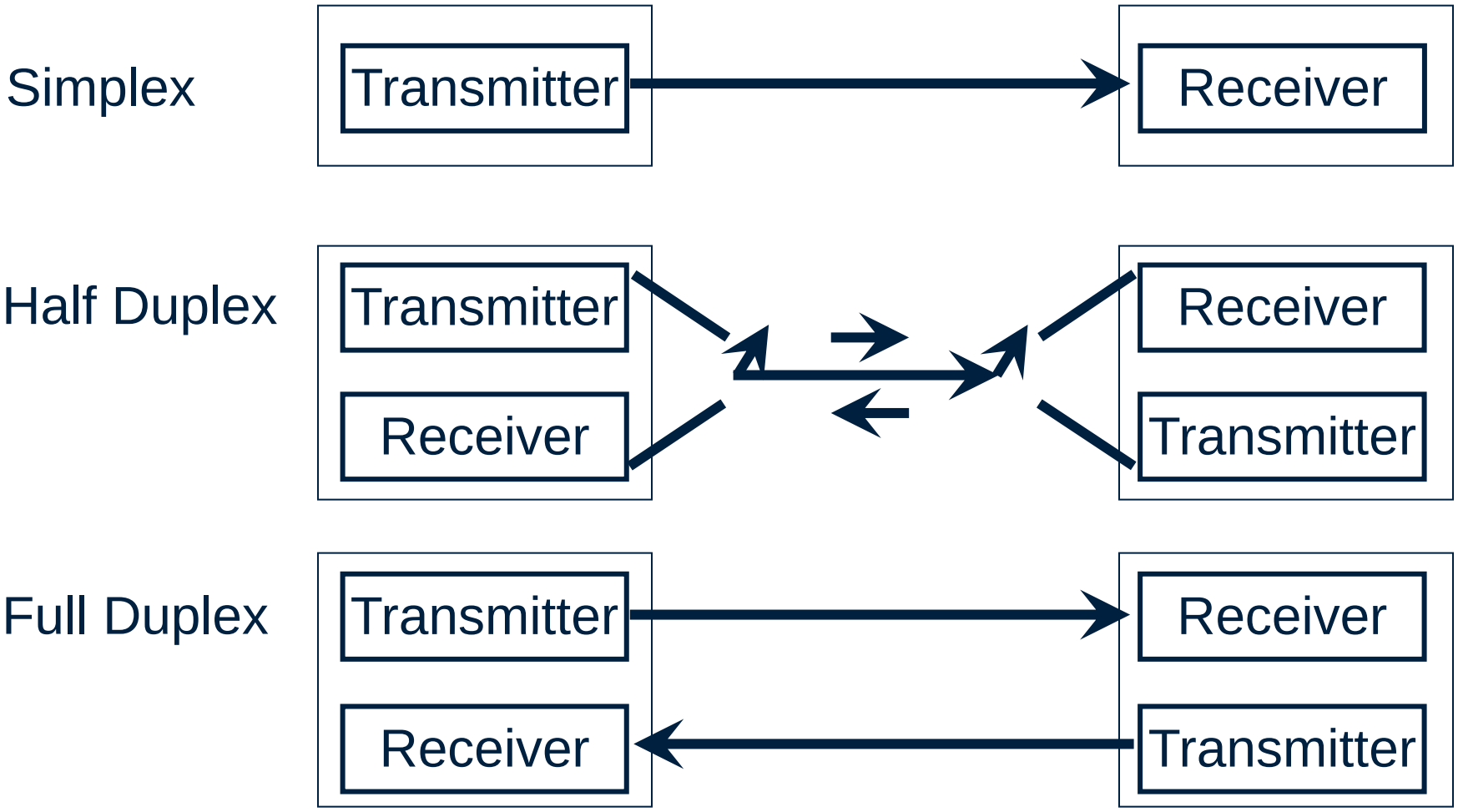
Review: Last Lecture

- Basics of Serial Communication:
  - Parallel vs. Serial
  - Direction: Simplex, Half duplex, Full duplex
  - Synchronization: Synchronous vs. Asynchronous
  - Framing
  - UART Protocol
  - Line Drivers



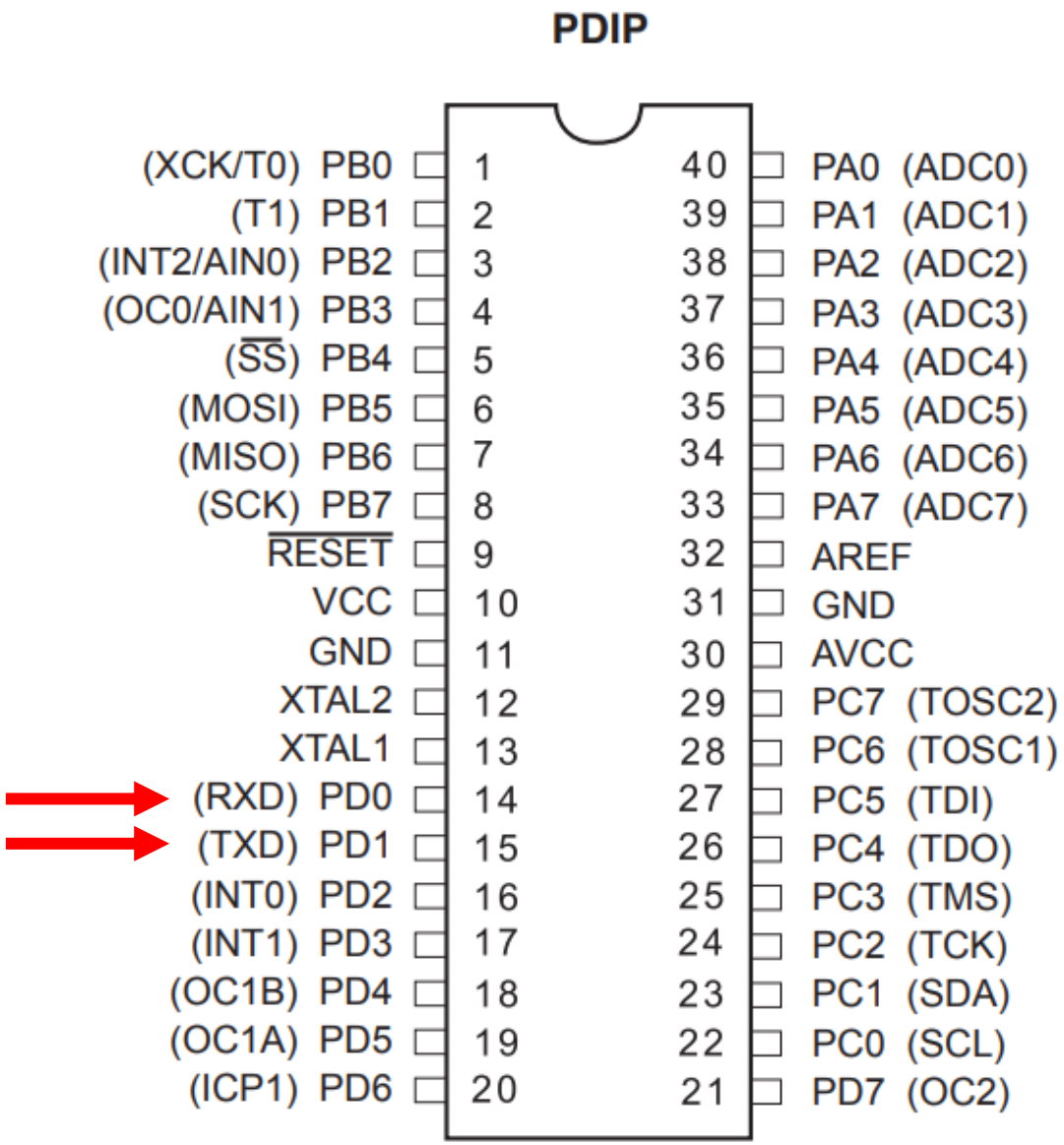


Simplex, Half-, and Full-Duplex Transfers



AVR Connection to RS232

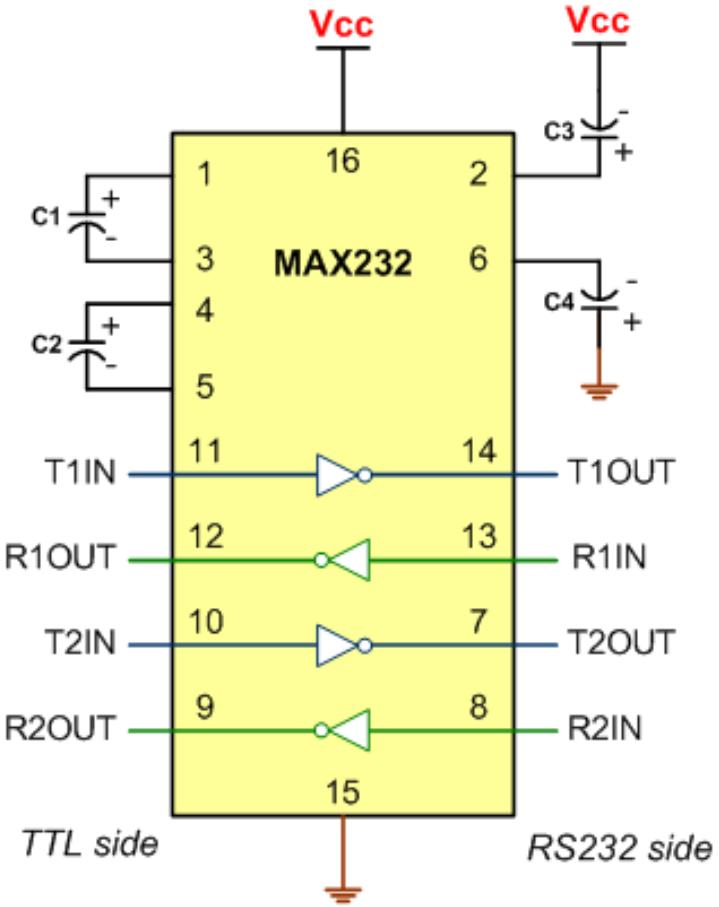
Review: Pin-out ATmega16A



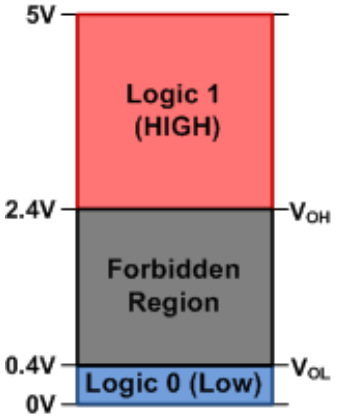
TxD and RxD pins in the AVR

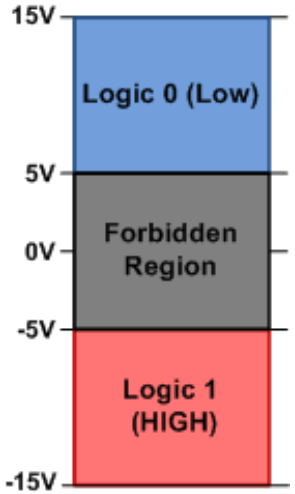
- Many of the pins of the RS232 connector are used for handshaking signals.
  - However, they are not supported by the AVR USART chips.
- In AVR, the data is received from or transmitted to
  - RxD: received data (PD0)
  - TxD: transmitted data (PD1)
- TxD and RxD of the AVR are TTL compatible.
- The AVR requires a line driver to make them RS232 compatible.
  - One such line driver is the MAX232 chip.

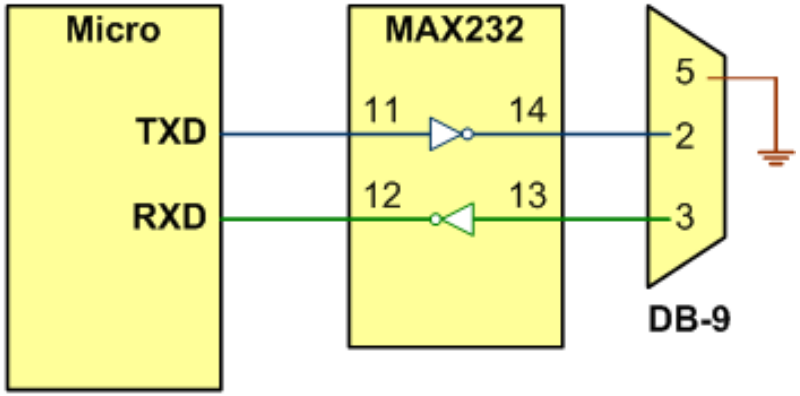
MAX232



(a) Inside MAX232



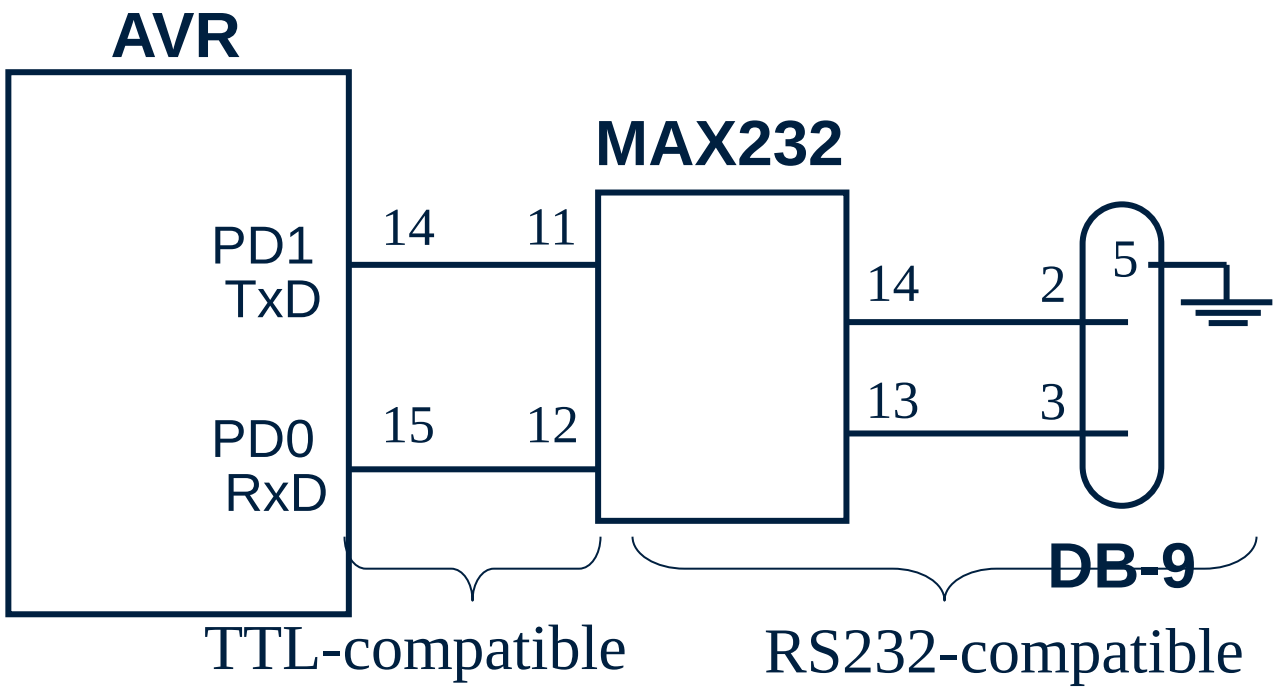




(b) MAX232 Connection to the Microcontroller / Microprocessor

MAX232

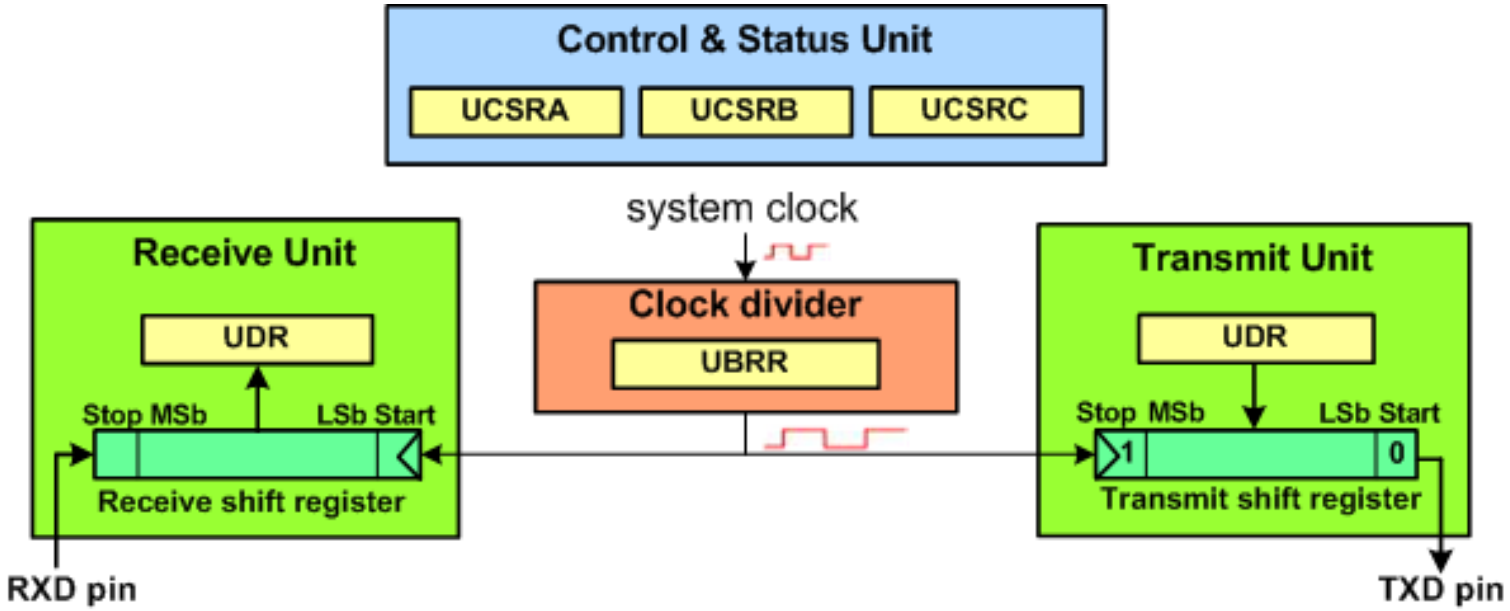
- MAX232 chip converts from RS232 voltage levels to TTL voltage levels, and vice versa:
  - MAX232 uses a +5V power source which is the same as the source voltage for the AVR.



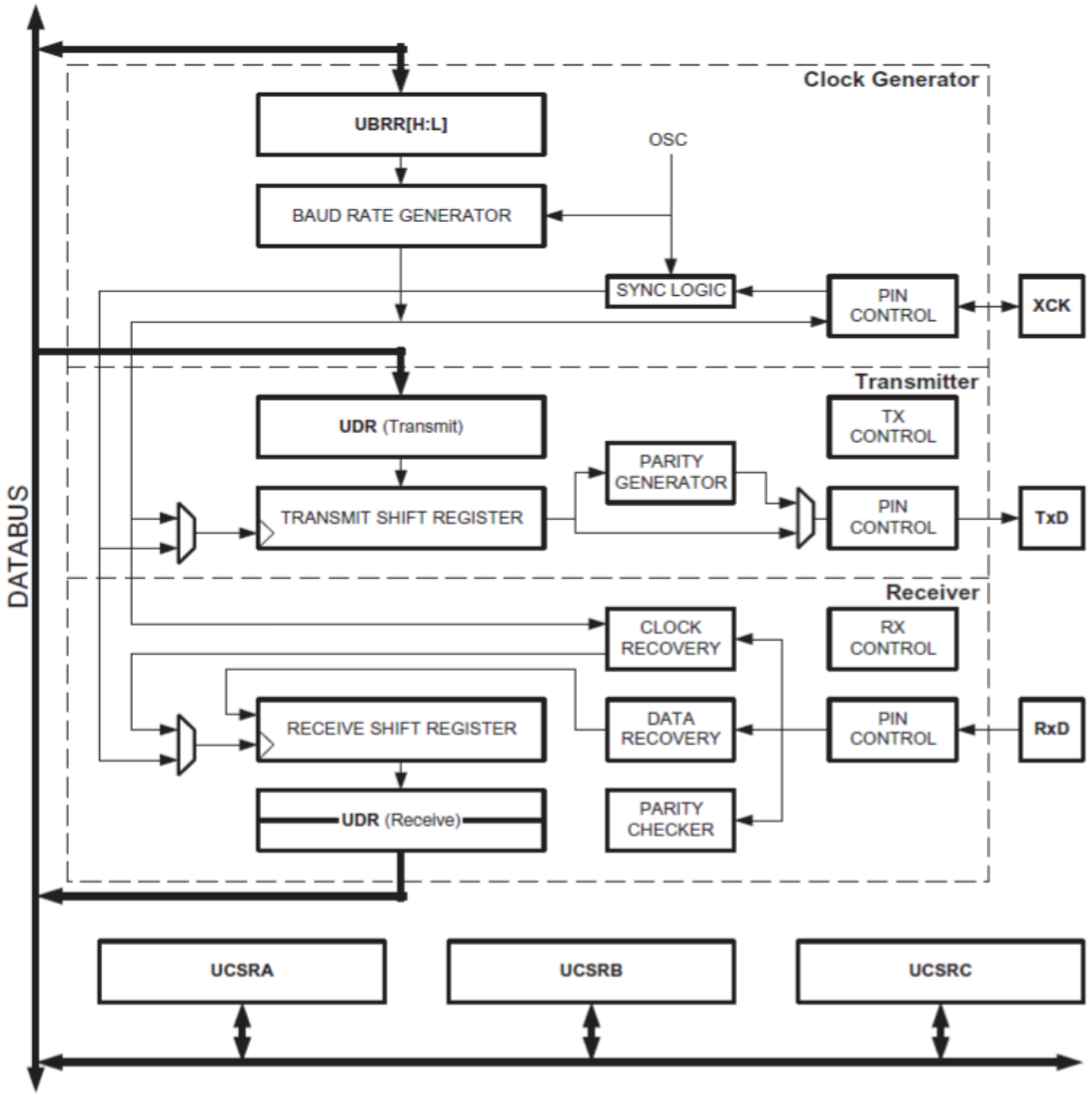
Atmega16A USART

UART in AVR

- Control registers: (initialize speed, data size, parity, etc)
  - UBRR, UCSRA, UCSRB, and UCSRC
- Send/receive register
  - UDR
- Status register
  - UCSRA

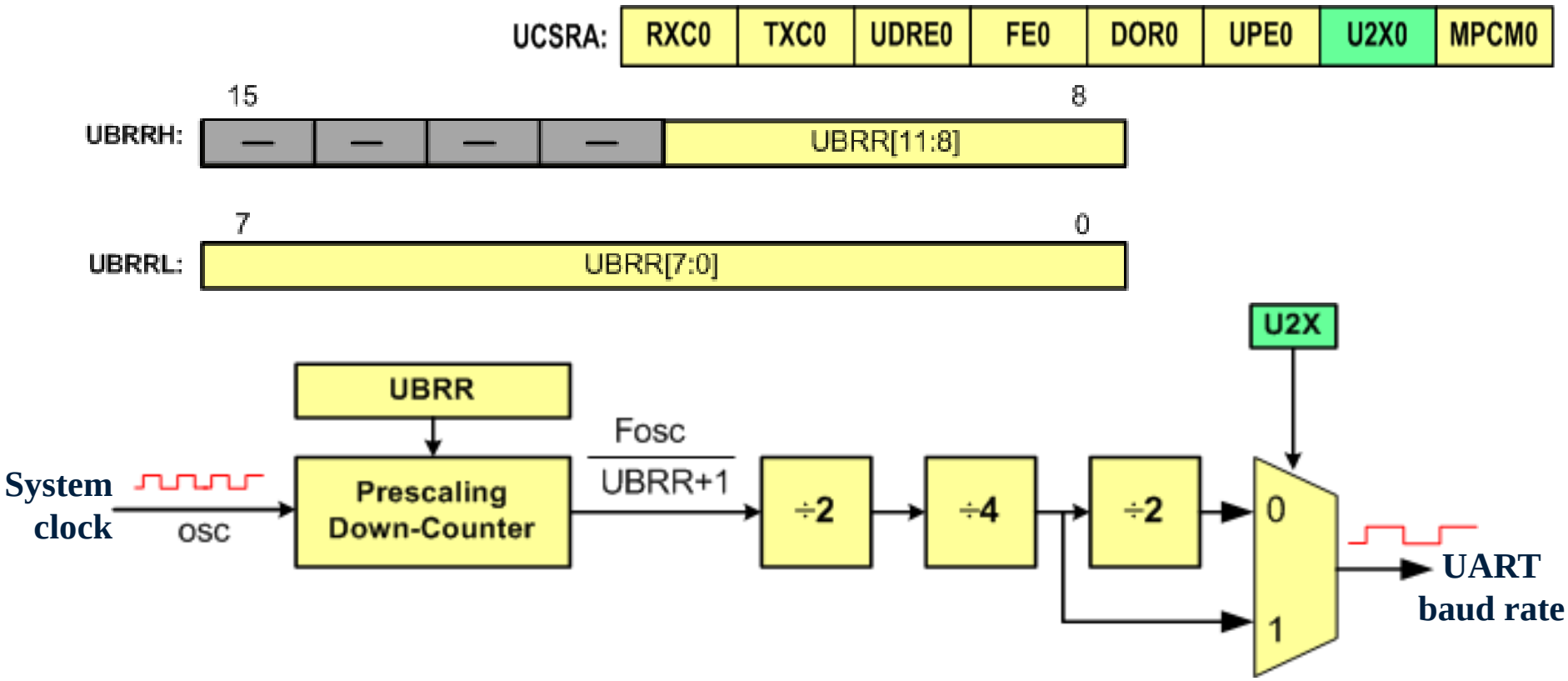


ATmega16A USART



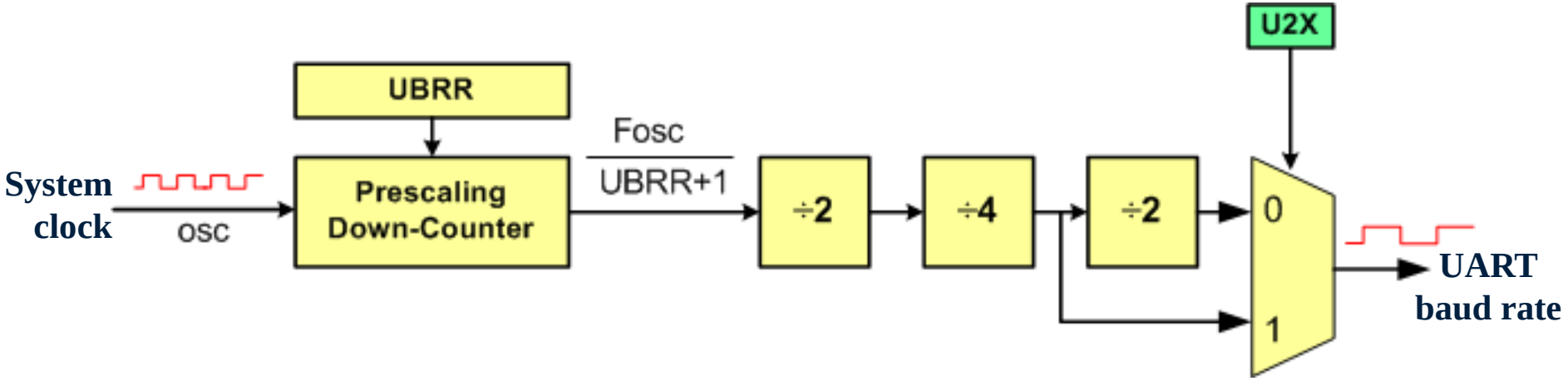
UBRR for Baud Rate

- UBRR (USART Baud Rate Register)



UBRR Register

- UBRR (USART Baud Rate Register)



- For a given crystal frequency (Fosc), the value loaded into the UBRR decides the baud rate:

$$\text{Baud rate} = \frac{\text{Fosc}}{(\text{UBRR}+1) \times 16}$$

Standard Baud Rates

- PC supports several baud rates.
  - You can use Netterm, terminal.exe, stty, ptty to send/receive data.

110 bps
150
300
600
1200
2400
4800
9600 (default)
19200

Note: Baud rates supported by 486/Pentium IBM PC BIOS.

Example: Find the UBRR value for 9600bps.

- Solution:

$$\text{Baud rate} = \frac{\text{Fosc}}{(\text{UBRR}+1) \times 16} \Rightarrow 9600 = \frac{16\ \text{MHz}}{(\text{UBRR}+1) \times 16}$$

$$\Rightarrow (\text{UBRR}+1) = \frac{1\text{MHz}}{9600} = 104.166 \Rightarrow \text{UBRR} = 103$$

Calculating Baud Rate Register Setting

Table 19-1. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

BAUD	Baud rate (in bits per second, bps)
f_{osc}	System Oscillator clock frequency
UBRR	Contents of the UBRRH and UBRRL Registers, (0 - 4095)

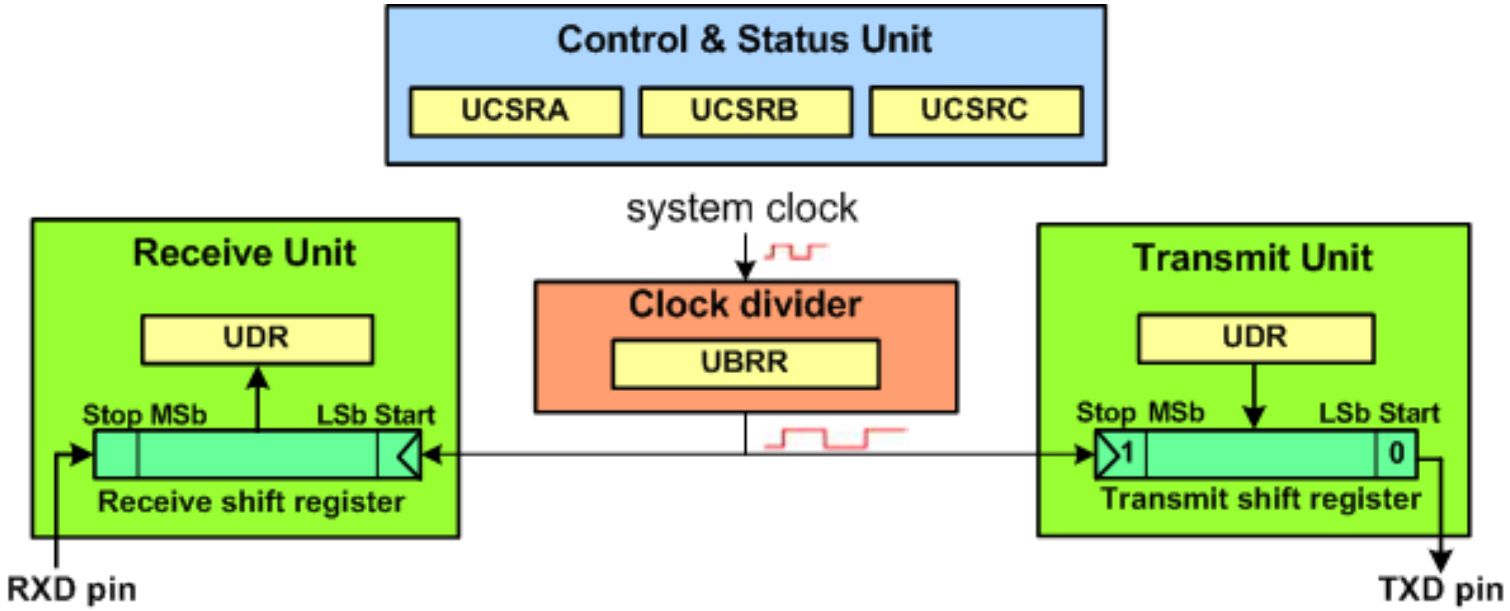
Example: AVR Baud Rate

Table 11-4: UBRR Values for Various Baud Rates (Fosc = 8 MHz, U2X = 0)

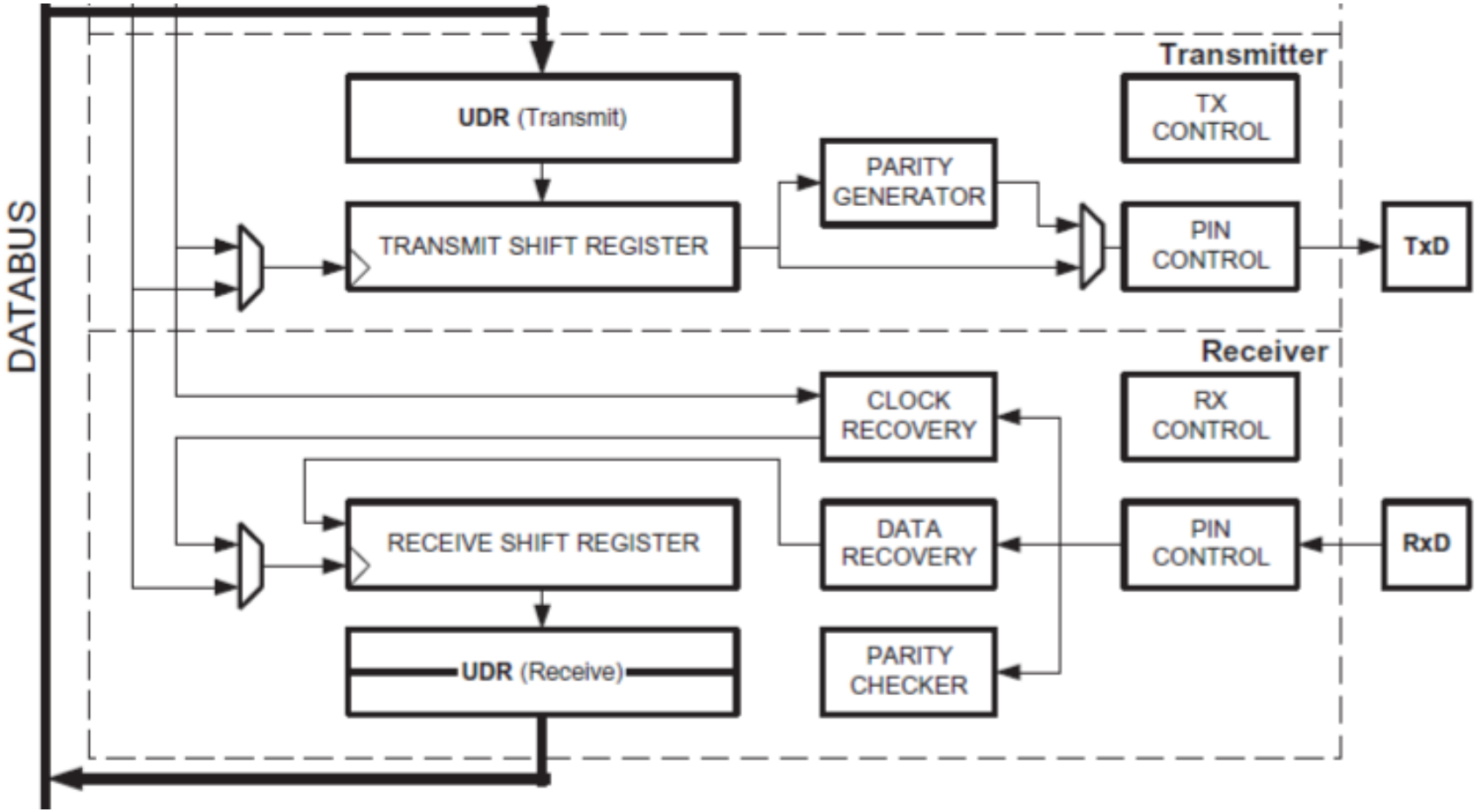
Baud Rate	UBRR (Decimal Value)	UBRR (Hex Value)
38400	12	C
19200	25	19
9600	51	33
4800	103	67
2400	207	CF
1200	415	19F

UART in AVR

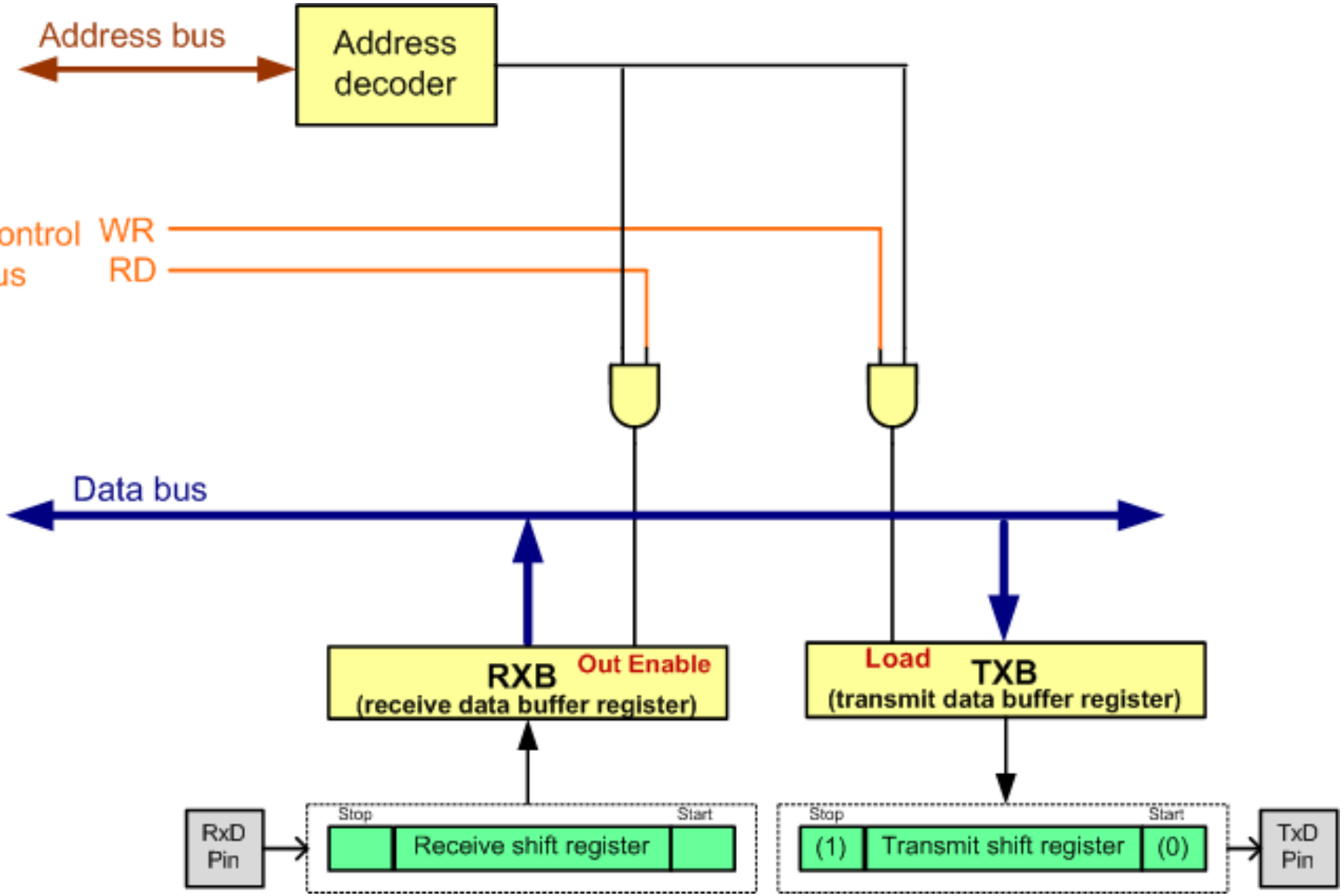
- Control registers: (initialize speed, data size, parity, etc)
  - UBRR, UCSRA, UCSRB, and UCSRC
- Send/receive register
  - UDR
- Status register
  - UCSRA



UDR: USART Data Register

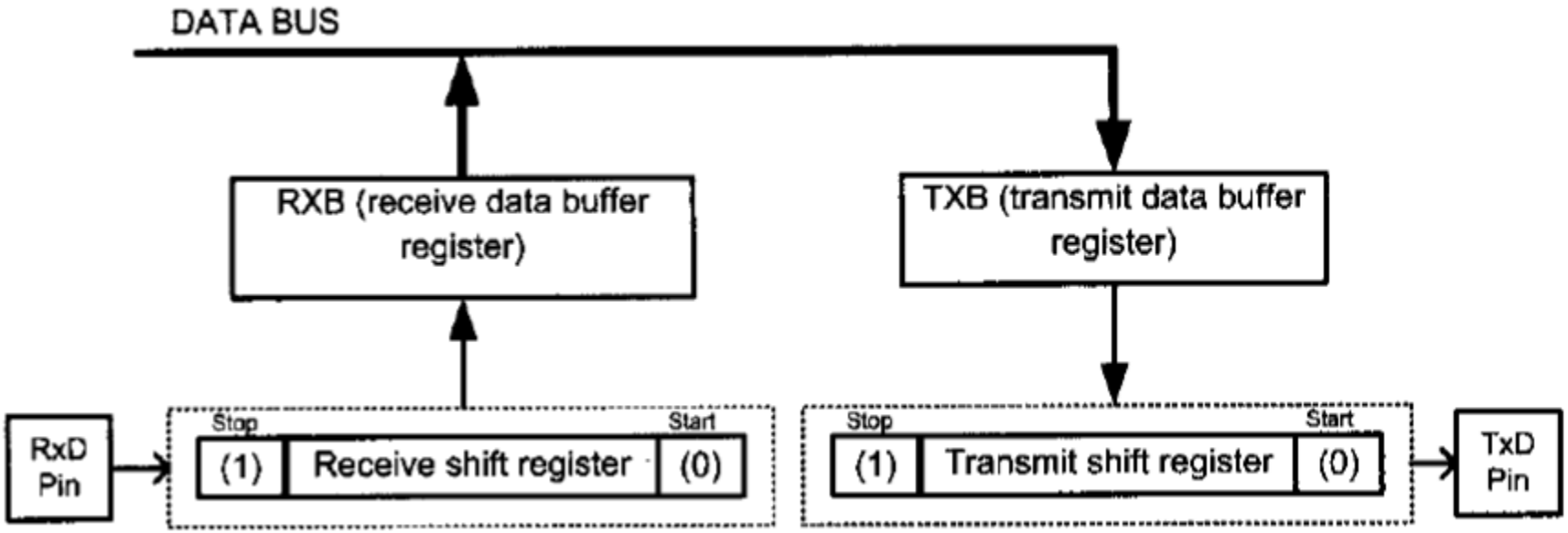


UDR


Address bus
Address decoder
Control bus
WR
RD
Data bus
RXB
Out Enable
(receive data buffer register)
Load
TXB
(transmit data buffer register)
RxD Pin
Stop
Start
Receive shift register
Stop
Start
(1)
Transmit shift register
(0)
TxD Pin

Trasnmit Buffer Register and Receive Register Buffer – (1)

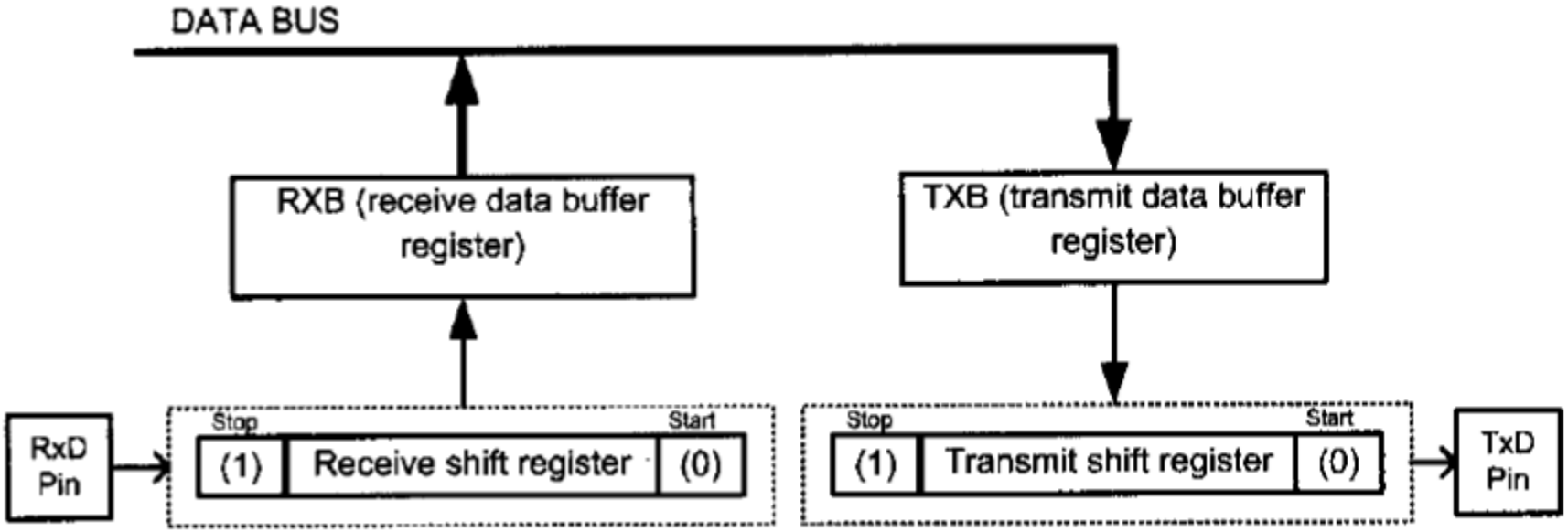
Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



Trasnmit Buffer Register and Receive Register Buffer – (2)

- When you write data to UDR, it will be transferred to the Transmist Data Buffer Register.

- When you read data from UDR, it will return the contents of the Receive Buffer Register.



USART Configuration: UCSR

- USART Control Status Registers (UCSR)
  - There are three UCSR
    - UCSRA
    - UCSRB
    - UCSRC

- Please consult Atmega16a Hardware Manual or your course book to examine the contents of these registers before continuing to read next slides.

UCSRA

UCSRA:	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0

- RXC0 (Bit 7): USART Receive Complete 0
  - This flag bit is set when there are new data in the receive buffer that are not read yet. It is cleared when the receive buffer is empty. It also can be used to generate a receive complete interrupt.
- TXC0 (Bit 6): USART Transmit Complete 0
  - This flag bit is set when the entire frame in the transmit shift register has been transmitted and there are no new data available in the transmit data buffer register (TXB). It can be cleared by writing a one to its bit location. Also it is automatically cleared when a transmit complete interrupt is executed. It can be used to generate a transmit complete interrupt.
- UDRE0 (Bit 5): USART Data Register Empty 0
  - This flag is set when the transmit data buffer is empty and it is ready to receive new data. If this bit is cleared you should not write to UDR0 because it overrides your last data. The UDRE0 flag can generate a data register empty interrupt.
- FE0 (Bit 4): Frame Error 0
  - This bit is set if a frame error has occurred in receiving the next character in the receive buffer. A frame error is detected when the first stop bit of the next character in the receive buffer is zero.
- DOR0 (Bit 3): Data OverRun 0
  - This bit is set if a data overrun is detected. A data overrun occurs when the receive data buffer and receive shift register are full, and a new start bit is detected.
- PE0 (Bit 2): Parity Error 0
  - This bit is set if parity checking was enabled (UPM1 = 1) and the next character in the receive buffer had a parity error when received.
- U2X0 (Bit 1): Double the USART Transmission Speed 0
- MPCM0 (Bit 0): Multi-processor Communication Mode 0

UCSRB

UCSRB:	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80

- RXCIE0 (Bit 7): Receive Complete Interrupt Enable
  - To enable the interrupt on the RXC0 flag in UCSR0A you should set this bit to one.
- TXCIE0 (Bit 6): Transmit Complete Interrupt Enable
  - To enable the interrupt on the TXC0 flag in UCSR0A you should set this bit to one.
- UDRIE0 (Bit 5): USART Data Register Empty Interrupt Enable
  - To enable the interrupt on the UDRE0 flag in UCSR0A you should set this bit to one.
- RXEN0 (Bit 4): Receive Enable
  - To enable the USART receiver you should set this bit to one.
- TXEN0 (Bit 3): Transmit Enable
  - To enable the USART transmitter you should set this bit to one.
- UCSZ02 (Bit 2): Character Size
  - This bit combined with the UCSZ1:0 bits in UCSRC sets the number of data bits (character size) in a frame.
- RXB80 (Bit 1): Receive data bit 8
  - This is the ninth data bit of the received character when using serial frames with nine data bits.
- TXB80 (Bit 0): Transmit data bit 8
  - This is the ninth data bit of the transmitted character when using serial frames with nine data bits.

UCSRC

UCSRC:	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0

- UMSEL01:00 (Bits 7:6): USART Mode Select
  - These bits select the operation mode of the USART:
    - 00 = Asynchronous USART operation
    - 01 = Synchronous USART operation
    - 10 = Reserved
    - 11 = Master SPI (MSPIM)

- UPM01:00 (Bit 5:4): Parity Mode
  - These bits disable or enable and set the type of parity generation and check.
    - 00 = Disabled
    - 01 = Reserved
    - 10 = Even Parity
    - 11 = Odd Parity

- USBS0 (Bit 3): Stop Bit Select
  - This bit selects the number of stop bits to be transmitted.
    - 0 = 1 bit
    - 1 = 2 bits

UCSZ02	UCSZ01	UCSZ00	Char. size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	1	1	9-bit

- UCSZ01:00 (Bit 2:1): Character Size
  - These bits combined with the UCSZ02 bit in UCSR0B set the character size in a frame.

- UCPOL0 (Bit 0): Clock Polarity
  - This bit is used for synchronous mode.

Recommended Reading

- The AVR Microcontroller and Embedded Systems: Using Assembly and C by Mazidi et al., Prentice Hall
  - Chapter 11 -> 11.2 and 11.3

THANK YOU