Microprocessors and Microcontrollers (EE-231)

Lecture-14

Main Objectives

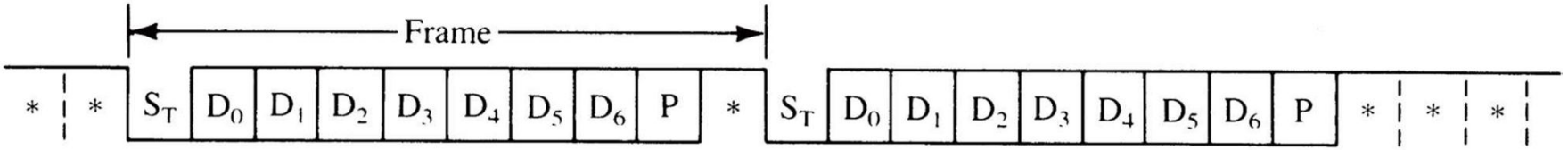
- 16550 Communication Device
- ADC and DAC interfacing

16550 Programmable Communications Interface

- National Semiconductor Corp's PC16550D is a programmable communications interface designed to connect to virtually any type of serial interface.
- 16550 is a universal asynchronous receiver/transmitter (UART) fully compatible with Intel microprocessors.
- 16550 operates at 0–1.5 M baud.
  - baud rate is bps (bits transferred per second) including start, stop, data, and parity
- 16550 also includes a programmable baud rate generator and separate FIFOs for input and output data to ease the load on the microprocessor.
- Each FIFO contains 16 bytes of storage.
- The most common communications interface found in the PC and many modems.

Asynchronous Serial Data

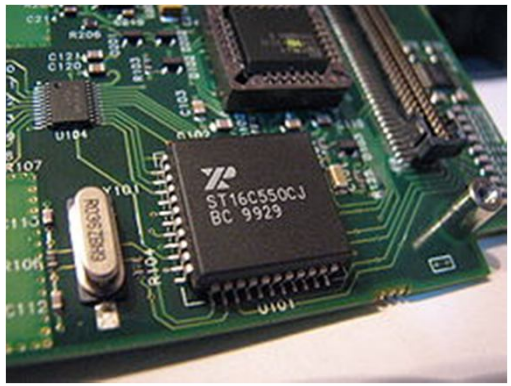
- Asynchronous serial data are transmitted and received without a clock and in frames as shown in figure.
  - each frame contains a start bit, seven data bits, parity, and one stop bit
    - Most Internet services use 10 bits, but normally do not use parity.
    - instead, eight data bits are transferred, replacing parity with a data bit



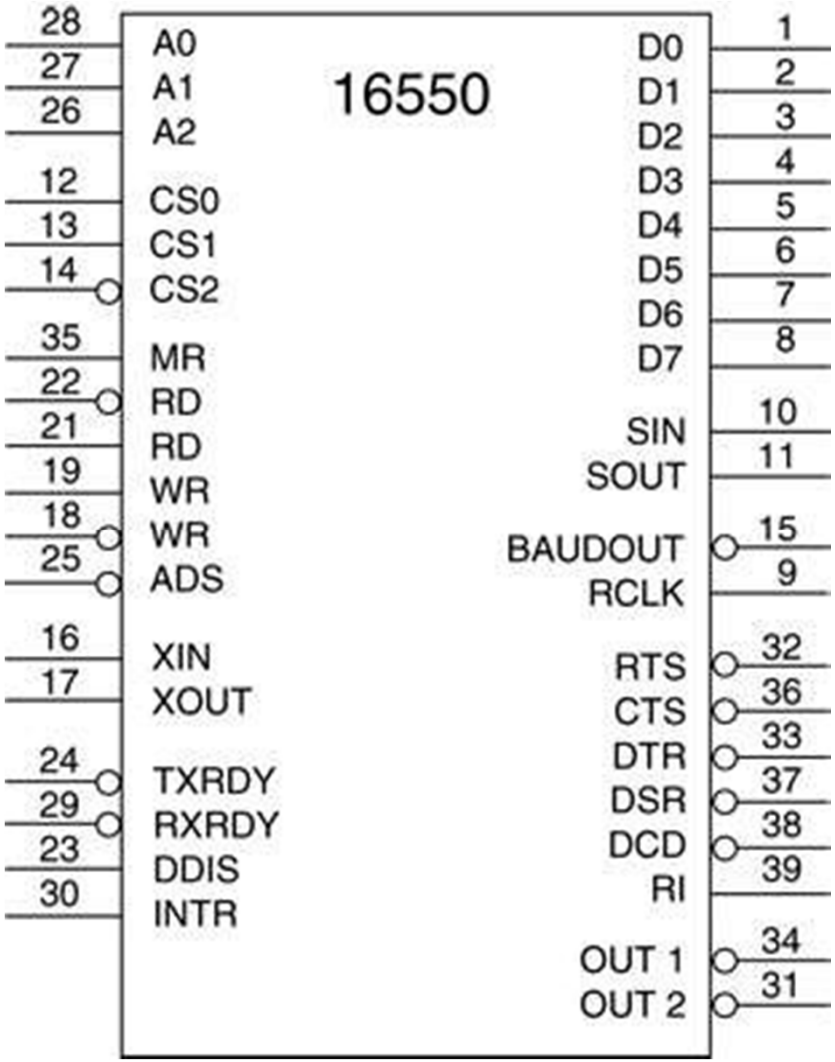
* = STOP
ST = START

16550 Functional Description

- The device is available as a 40-pin DIP (**dual in-line package**) or as a 44-pin PLCC (**plastic leadless chip carrier**).
- Two completely separate sections are responsible for data communications.
  - the receiver and the transmitter
- Because each sections is independent, 16550 is able to function in simplex, half-duplex, or full-duplex modes.



The pin-out of the 16550 UART.



16550 Functional Description

- A major feature of the 16550 is its internal receiver and transmitter FIFO (first-in, first-out) memories.
- Because each is 16 bytes deep, the UART requires attention from the processor only after receiving 16 bytes of data.
  - also holds 16 bytes before the processor must wait for the transmitter
- The FIFO makes this UART ideal when interfacing to high-speed systems because less time is required to service it.

Modes of Communication

- A **simplex** system is one in which the transmitter or receiver is used by itself.
  - such as in an FM (**frequency modulation**) radio station
- A **half-duplex** system is a CB (**citizens band**) radio.
  - transmit and receive, but not at the same time
- A **full-duplex** system allows transmission and reception in both directions simultaneously.
  - the telephone is a full-duplex system

16550 Functional Description

- The 16550 can control a **modem** (**modulator/demodulator**), a device that converts TTL serial data into audio tones that can pass through the telephone system.
- Six pins on 16650 are for modem control: DSR (**data set ready**), DTR (**data terminal ready**), CTS (**clear-to-send**), RTS (**request-to-send**), RI (**ring indicator**), and DCD (**data carrier detect**).
- The modem is referred to as the data set and the 16550 is referred to as the data terminal.

16550 Pin Functions

A_0, A_1, A_2

- The **address inputs** are used to select an internal register for programming and also data transfer.

A_2	A_1	A_0	*Function*
0	0	0	Receiver buffer (read) and transmitter holding (write)
0	0	1	Interrupt enable
0	1	0	Interrupt identification (read) and FIFO control (write)
0	1	1	Line control
1	0	0	Modem control
1	0	1	Line status
1	1	0	Modem status
1	1	1	Scratch

16550 Pin Functions

- **ADS**
- The **address strobe** input is used to latch the address lines and chip select lines.
- If not needed (as in the Intel system), connect this pin to ground.
- The ADS pin is designed for use with Motorola microprocessors.
- **BAUDOUT**
- The **baud out** pin is where the clock signal generated by the baud rate generator from the transmitter section is made available.
- It is most often connected to the RCLK input to generate a receiver clock that is equal to the transmitter clock.
- **CS0,CS1,CS2**
- The **chip select** inputs must all be active to enable the 16550 UART.

16550 Pin Functions

- **CTS**
- The **clear-to-send** (if low) indicates that the modem or data set is ready to exchange information.
- This pin is often used in a half-duplex system to turn the line around.
- **D0-D7**
- The **data bus** pins are connected to the microprocessor data bus.
- **DCD**
- **Data carrier detect** input is used by the modem to signal the 16550 that a carrier is present.
- **DTR**
- **Data terminal ready** is an output that indicates that the data terminal (16550) is ready to function.

16550 Pin Functions

- **INTR**
- **Interrupt request** is an output to the microprocessor used to request an interrupt (INTR=1) when the 16550 has a receiver error, it has received data, and the transmitter is empty.
- **DDIS**
- The disable driver output becomes logic 0 to indicate the microprocessor is reading data from the UART.
- DDIS can be used to change the direction of data flow through a buffer.
- **DSR**
- Data set ready is an input to the 16550, indicating that the modem or data set is ready to operate.

16550 Pin Functions

- **MR**
- Master reset initializes the 16550 and should be connected to the system RESET signal.
- **OUT1,OUT2**
- **User-defined output pins** that can provide signals to a modem or any other device as needed in a system.
- **RCLK**
- **Receiver clock** is the clock input to the receiver section of the UART.
- **RD,RD**
- **Read inputs** (either may be used) cause data to be read from the register specified by the address inputs to the UART.

16550 Pin Functions

- **RI**
- **Ring indicator** input is placed at logic 0 by the modem to indicate the phone is ringing.
- **RTS**
- Request-to-send is a signal to the modem indicating that the UART wishes to send data.
- **SIN, SOUT**
- These are the **serial data pins**. SIN accepts serial data and SOUT transmits serial data.
- **RXRDY**
- **Receiver ready** is a signal used to transfer received data via DMA techniques.

16550 Pin Functions

- **TXRDY**
- **Transmitter ready** is a signal used to transfer transmitter data via DMA.
- **WR,WR**
- Write (either may be used) connects to the microprocessor write signal to transfer commands and data to the 16550.
- **XIN,XOUT**
- These are the main **clock** connections.
- A crystal is connected across these pins to form a crystal oscillator, or XIN is connected to an external timing source.

Programming the 16550

- Programming is a two-part process & includes the initialization dialog and operational dialog.

- In the PC, which uses the 16550 or its programming equivalent, I/O port addresses are decoded at 3F8H - 3FFH for COM port 0 and 2F8H - 2FFH for COM port 2.

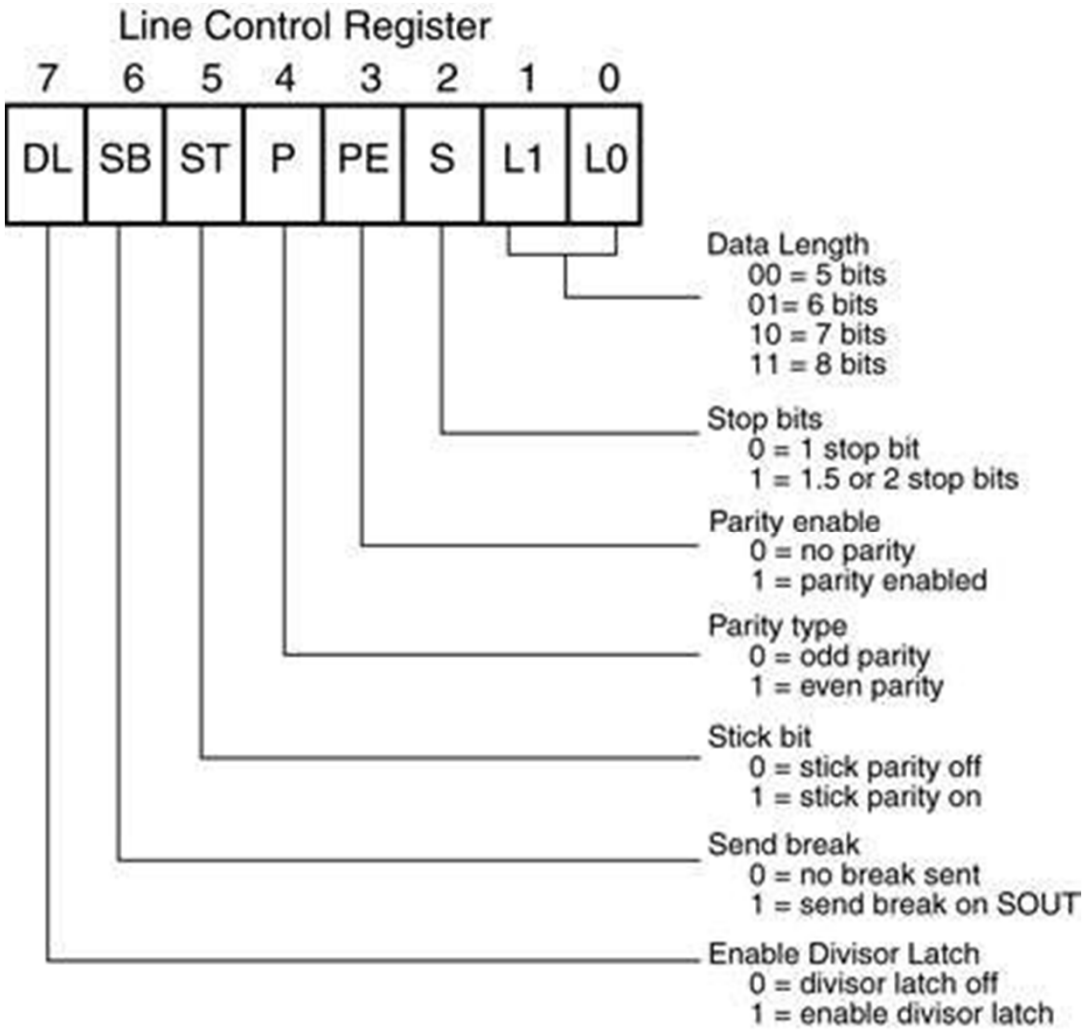
Initializing the 16550

- Initialization dialog after a hardware or software reset, consists of two parts:
  - programming the line control register
  - programming the baud rate generator
- The line control register selects the number of data bits, stop bits, and parity (whether even or odd, or if parity is sent as a 1 or a 0)
- Baud rate generator is programmed with a divisor that determines the baud rate of the transmitter section.

Initializing the 16550

- The line control register is programmed by outputting information to port 011 (A_2, A_1, A_0).
- The rightmost two bits of the line control register select the number of transmitted data bits (5, 6, 7, or 8).
- Number of stop bits is selected by S in the line control register.
  - if S = 0, one stop bit is used
  - if S = 1, 1.5 stop bits are used for five data bits,
    & two stop bits with six, seven, or eight data bits

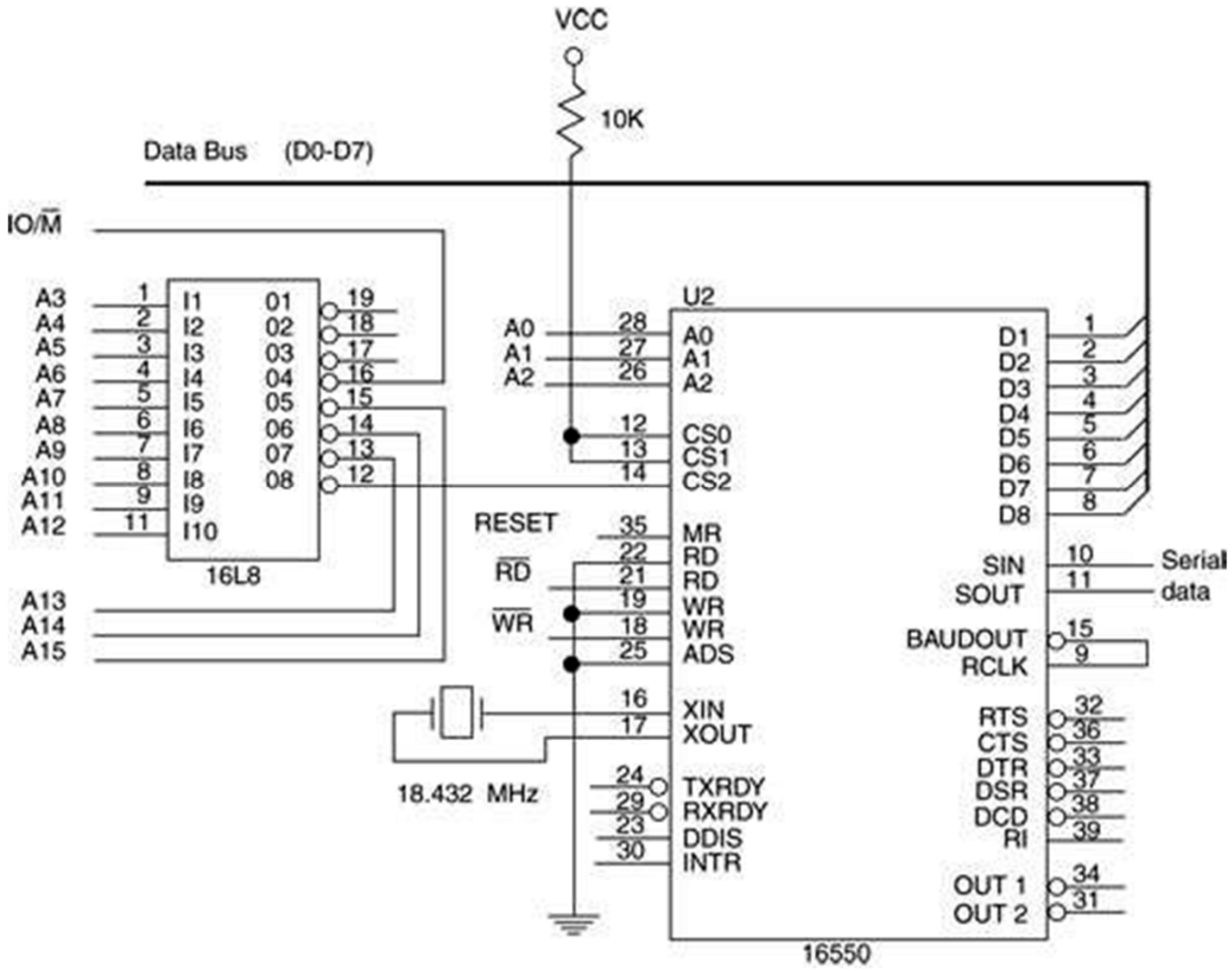
The contents of the 16550 line control register.



Programming the Baud Rate

- The baud rate generator is programmed at I/O addresses 000 and 001 (A_2, A_1, A_0).
- Port 000 is used to hold the least significant part of the 16-bit divisor and port 001 is used to hold the most significant part.
  - value used for the divisor depends on the external clock or crystal frequency

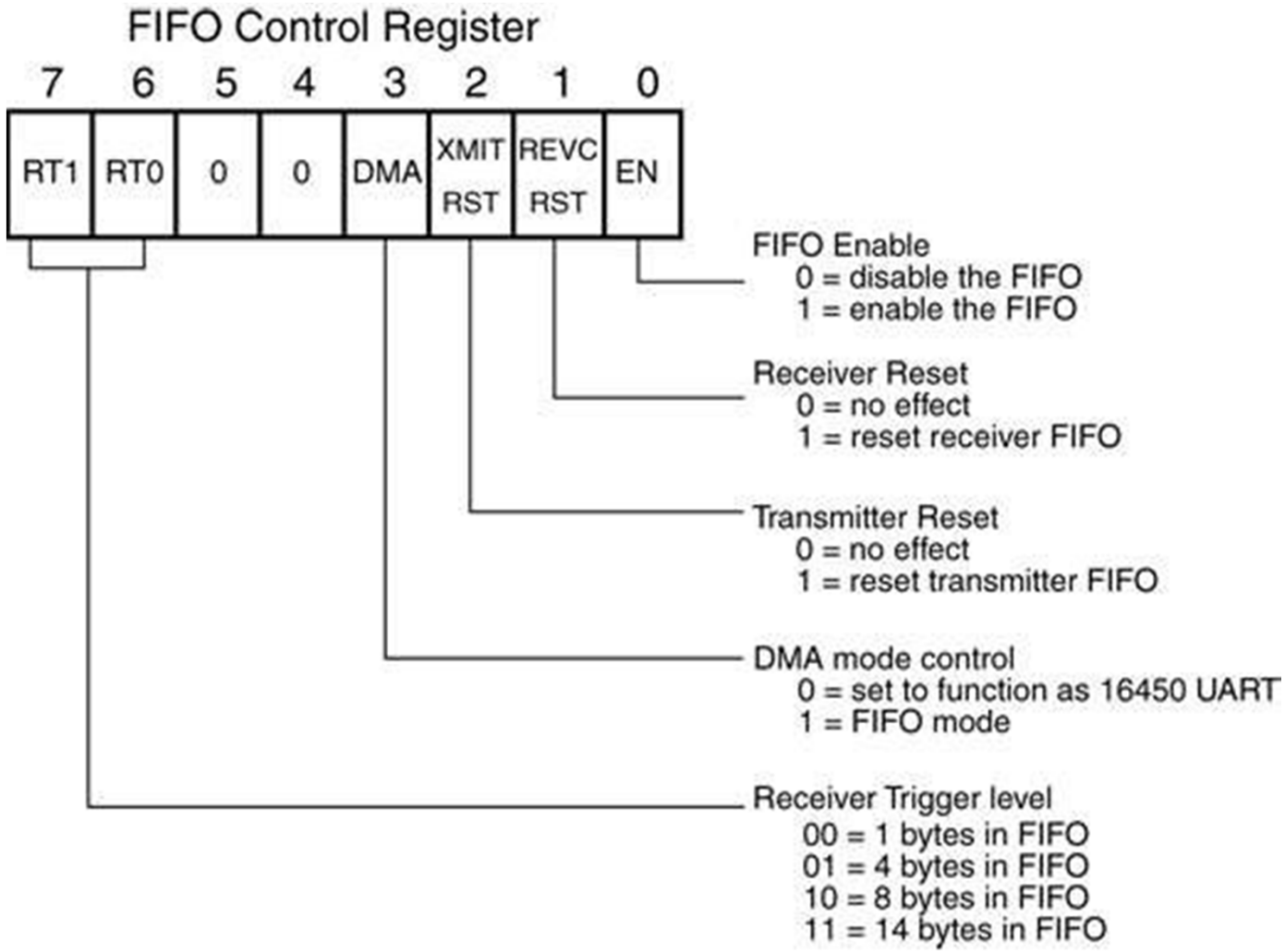
The 16550 interfaced to the 8088 microprocessor at ports 00F0H–00F7H.



Programming FIFO

- After the line control register and baud rate divisor are programmed into the 16550, it is still not ready to function.
- The FIFO control register must still be programmed.
  - at port F2H in the circuit of previous figure.
- Next Figure illustrates the FIFO control register for the 16550.
  - the register enables the transmitter & receiver (bit 0=1), clears the transmitter & receiver FIFOs
  - it also provides control for the 16550 interrupts

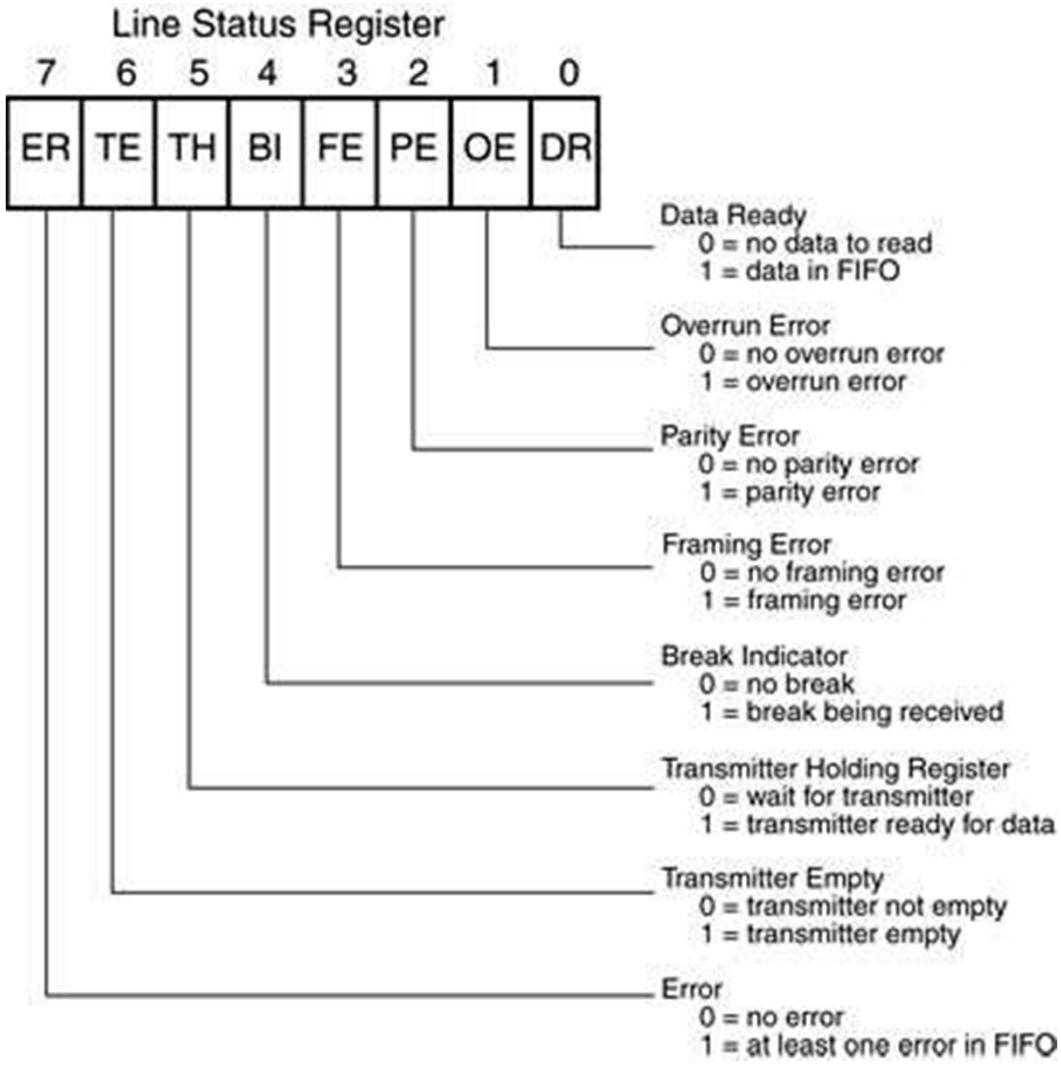
The FIFO control register of the 16550 UART.



Sending and Receiving Serial Data

- The line status register contains information about error conditions and the state of the transmitter and receiver.
- This register is tested before a byte is transmitted or can be received.
- To transmit we check TH bit.
- To read received information from the 16550, test the DR bit of the line status register.
- Upon the reception of data, the procedure tests for errors.
- if an error is detected, the procedure returns with AL equal to an ASCII ‘?’
- if no error has occurred, the procedure returns with AL equal to the received character

The contents of the line status register of the 16550 UART.

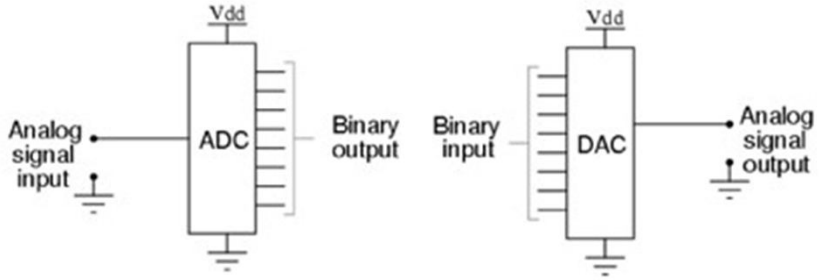


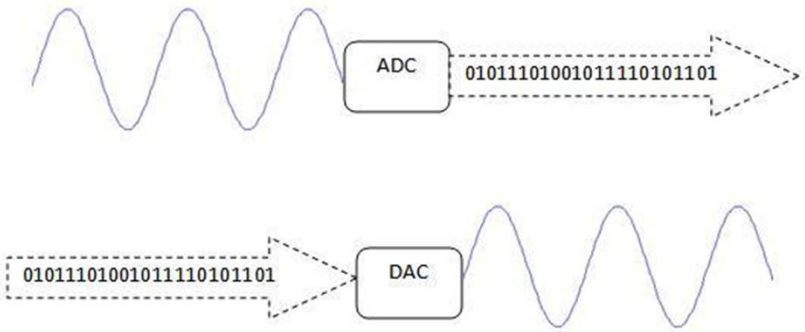
UART Errors

- Errors detected by 16550 are:
  i) Parity ii) Framing and iii) Overrun errors
- These errors should not occur during normal operation.
- A **parity error** indicates the received data contain the wrong parity.
  - if a parity error occurs, it indicates noise was encountered during reception
- A **framing error** indicates the start and stop bits are not in their proper places.
  - occurs if the receiver is receiving data at an incorrect baud rate
- An **overrun error** indicates data have overrun the internal receiver FIFO buffer.
  - occurs only if the software fails to read the data from the UART before the receiver FIFO is full

Analog-to-digital (ADC) & Digital-to-analog (DAC) Converters

- These devices are used to interface the microprocessor to the analog world.
- Many events monitored and controlled by the microprocessor are analog events.
- These range from monitoring all forms of events, even speech, to controlling motors and like devices.





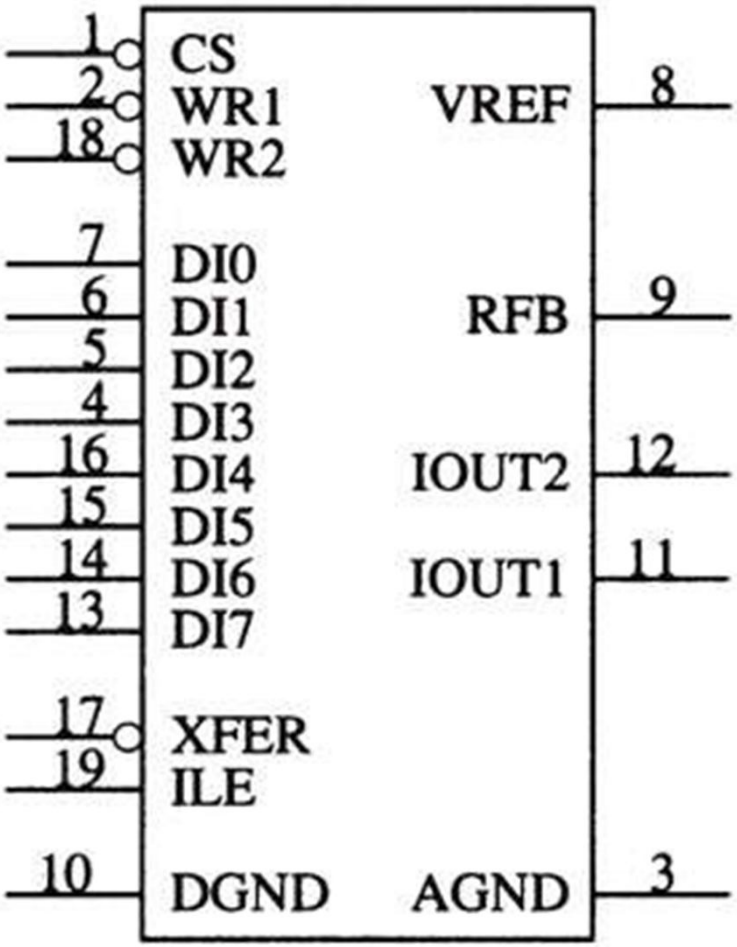
The DAC0830 Digital-to-Analog Converter

- A fairly common and low-cost digital-to-analog converter is the DAC0830.
  - a product of National Semiconductor Corp
- An 8-bit converter that transforms an 8-bit binary number into an analog voltage.
- Other converters are available that convert from 10-, 12-, or 16-bit binary numbers into analog voltages.
- The number of voltage steps generated by the converter is equal to the number of binary input combinations.
- an 8-bit converter generates 256 voltage levels
- a 10-bit converter generates 1024 levels
- The DAC0830 is a medium-speed converter that transforms a digital input to an analog output in approximately 1.0 µs.

The DAC0830 Digital-to-Analog Converter

- The device has eight data bus connections for the application of the digital input code.
- Analog outputs labeled IOUT1 & IOUT2 are inputs to an external operational amplifier.
- Because this is an 8-bit converter, its output step voltage is defined as $-V_{REF}$ (reference voltage), divided by 255.
  - the step voltage is often called the resolution of the converter

The pin-out of the DAC0830 digital-to-analog converter.

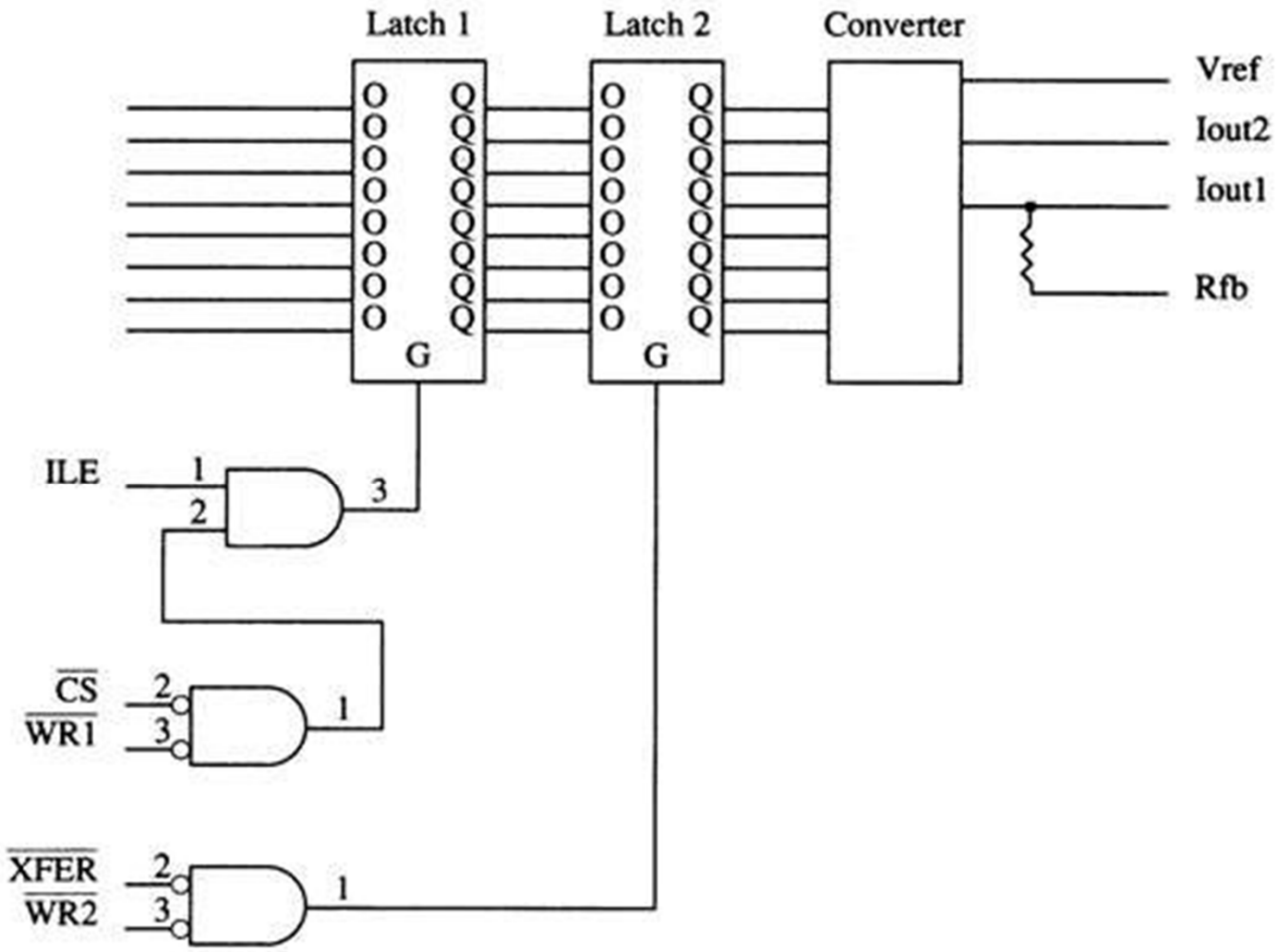


DAC0830

Internal Structure of the DAC0830

- This device contains two internal registers.
  - the first is a holding register
  - the second connects to the R–2R internal ladder converter
- The two latches allow one byte to be held while another is converted.
- The first latch is often disabled and the second for entering data into the converter.

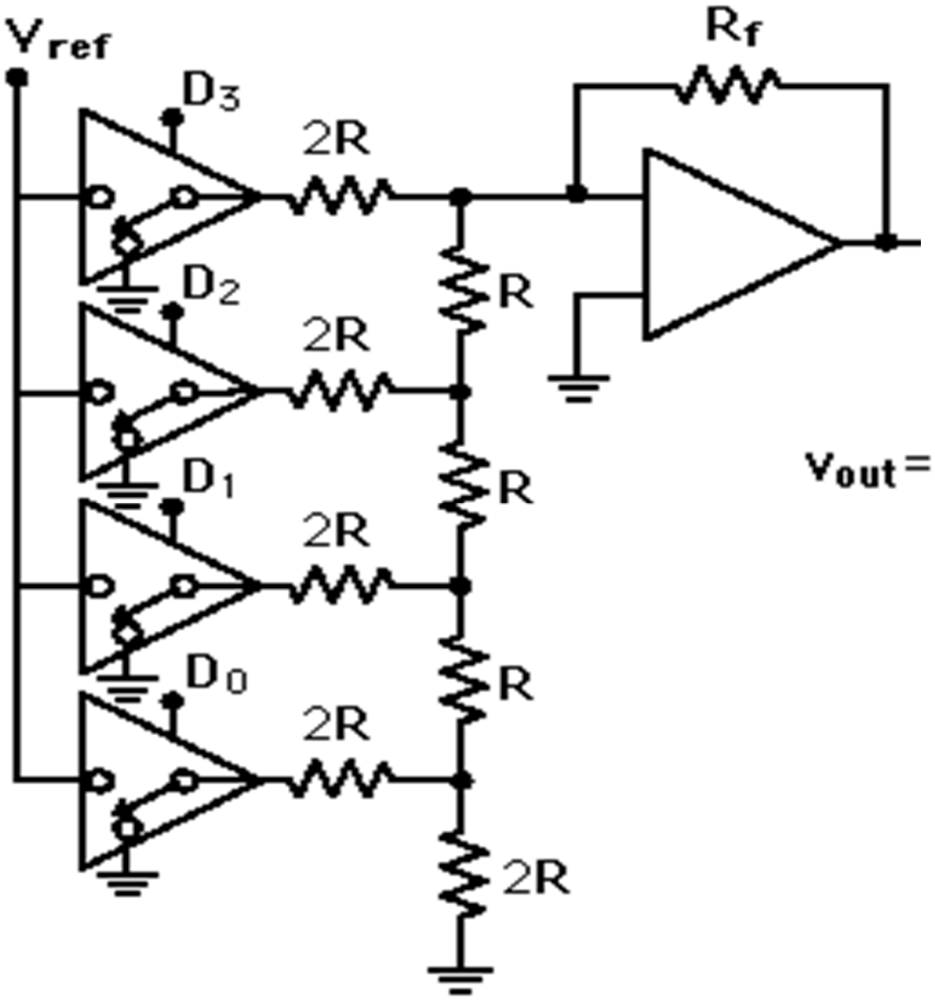
The internal structure of the DAC0830.



Internal Structure of the DAC0830

- Both latches within the DAC0830 are transparent latches.
  - when G input is logic 1, data pass through
  - when G input becomes logic 0, data are latched
- The output of the R–2R ladder within the converter appears at IOUT1 and IOUT2.
- These outputs are designed to be applied to an operational amplifier such as a 741 or similar device.

Internal Structure of the DAC0830



$$V_{out} = \frac{R_f}{R} V_{ref} \left[\frac{D_0}{16} + \frac{D_1}{8} + \frac{D_2}{4} + \frac{D_3}{2} \right]$$