CpE 213
Digital Systems Design

Lecture 26

Tuesday 11/25/2003

UMR UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.

Announcements

- HW 8 deadline has been postponed.
  - New due date: Tuesday Dec. 2.
- Project peer review has been posted.
  - Please keep the ratings in mind while working on the project.
- Project demos are due on Monday 12/1.
  - Signup sheet is on my door. The spokesperson for each group should sign up for a slot on the schedule.
  - The sheet is posted on Blackboard for your reference.
  - All group members must be present at demonstration.
  - Deadline for signup is today.

Outline

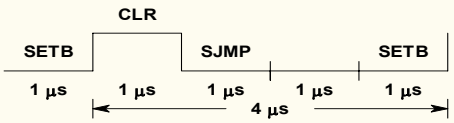
- Timers and Counters for the 8051
  - Final notes and summary
- Serial Communications
  - Introduction to serial I/O transfer
  - Introduction to asynchronous serial communication
  - Description of UART
  - RS-232 standard and signals
  - Interfacing the 8051 to an RS232 connector.

Some slides adapted from Mr. K. T. Ng

Short Timing Interval Using Software Loops

- Very short intervals can not be generated using timers because of the overhead needed to start and stop the timers.
- Such short timing intervals are usually generated using tight software loops.
- The following code segment generates a 250 KHz wave with 25% duty cycle on P1.0. Assume XTAL = 12 MHz.

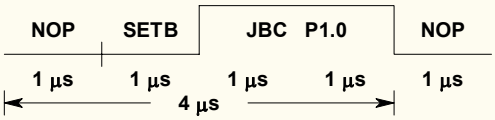
```
LOOP:    SETB  P1.0  ;1 machine cycle
         CLR   P1.0  ;1 machine cycle
         SJMP  LOOP  ;2 machine cycles
```



Short Timing Interval (Cont.)

- The following code segment generates a 250 KHz square wave on P1.0. Assume XTAL = 12 MHz.

```
LOOP:       NOP              ;1 machine cycle
            SETB  P1.0       ;1 machine cycle
            JBC   P1.0,LOOP  ;2 machine cycles
```



Summary of Techniques

Time Interval	Technique
No limit	Software loops
No limit	16-bit plus software loops
65536 machine cycles	16-bit timer
256 machine cycles	Auto-reload (8-bit)
<~ 10 machine cycles	Tight software tuning

Important Note

- See Lecture 25 handouts for two examples of programming timers in C.

Serial Communication with the 8051

Intro to Data Transfer

- Computers transfer data in two ways:
  - Parallel
    - hard disks, printers
    - data is sent one or more bytes at a time
    - 8 or more lines required
    - works well for short distances
  - Serial
    - mouse, PDA sync, keyboard
    - data is sent one bit at a time
    - only 1 line required
    - works well for long distances
    - good for wireless and remote access
- Serial transmission is preferred mostly for its low cost, ease of use and simplicity.

Serial Communication

- Serial communications means that information is transmitted from source to destination over a single pathway.
- Serial data transmission system is divided into 2 categories:
  - Synchronous transmission
    - transmits a block of data and a clock signal or known bit pattern that can be used for synchronization.
    - A block number and check sum are also transmitted.
    - Since digital data are transmitted as a stream of bits, synchronization is required between the source and destination of the data.

Serial Communication (Cont.)

- Asynchronous transmission
  - transmits one character at a time by adding start and stop bits to the character code.
  - Asynchronous here means “asynchronous at the byte level” i.e. there may be a variable-length gap between each byte. But within each byte, the bits are still synchronized; their durations are the same.
- The 8051 and the IBM-compatible PCs support only asynchronous serial communication.

Serial Transmission Modes

- The transmission mode is used to define the direction of signal flow between two linked devices.
- There are three transmission modes:
  - Simplex transmission means that data flows in one direction only.
  - Half-duplex transmission allows data to flow in both directions, but not at the same time.
  - Full-duplex transmission allows data to flow in both directions at the same time.

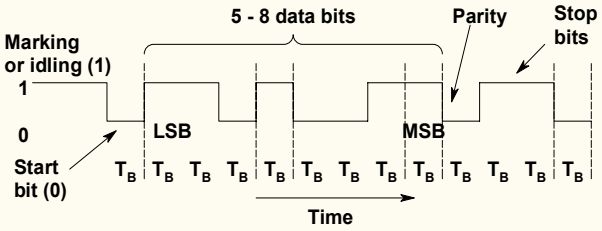
Serialization and De-serialization

- Serialization: converting bit-parallel bytes off a data bus to a serial bit stream for transmission over a single channel.
  - Uses shift registers
  - Uses the TRANSMITTER part of a special integrated circuit developed for asynchronous systems.
- De-serialization: converting a serial stream of bits to bit-parallel bytes for use within the computer data bus.
  - Uses shift registers
  - Uses the RECEIVER part of a special integrated circuit developed for asynchronous systems.

Operations of Asynchronous Transmission

- The line is normally high.
- A START bit (a low bit) is transmitted to synchronize the receiver to the byte.
- The data bits are sent, LSB first.
- An optional PARITY bit is appended. It is used to checked to see if any bit in the received byte was corrupted or not.
- Either 1 or 2 STOP bit(s) inform(s) the receiver of the end of the byte.
- This procedure happens for every byte.

Serial Data Signal



- Baud rate is the number of signal changes per second

$$\text{Baud rate} = \frac{1}{T_B}\ s^{-1}$$

- Standard baud rates are: 110 (the old teletype or TTY rate), 300 (early PC modems), 1200, 2400, 9600, 14.4K, 19.2K, 28.8K, 38.4K, and 57.6K

Baud Rate vs. Bits/Sec(bps)

- Baud rate is modem terminology
  - Defined as signaling speed.
- Rate of data transfer (bps) is the number of bits of information that are transmitted/received in a unit time (per second).
- In modems, a single signal change could correspond to transfer of several bits.
- The two are the same as far as the conductor wire is concerned.

Group Exercise

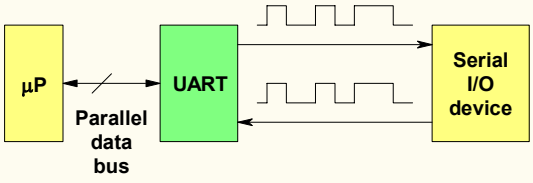
- Calculate the overhead (%) incurred by asynchronously transmitting one byte of data over a serial connection. Assume that we are using one stop bit and:
  a) a parity bit
  b) no parity bit
- The transfer of data using parallel lines is (faster,slower) but (more, less) expensive.
- True or false: sending data to a printer is duplex.
- True or false: we need two data lines for full duplex serial transmission.

Character Encoding

- The most common form of character encoding is the American Standard Code for Information Interchange (ASCII). This is a seven-bit code that allows for 128 characters.
- The codes that are less than 20H are not alpha-numeric characters, but control codes such as carriage return or backspace.
- The numbers 0 - 9 are encoded as 30H - 39H. The capital letters A - Z have codes 41H - 5AH, and the lower case letters a - z are 61H - 7FH.

Universal Asynchronous Receiver/Transmitter (UART)

- Asynchronous serial data communication uses a special IC (UART) operating in asynchronous mode to serialize and de-serialize data.
- The UART also controls the bit rate, generates the start and stop bits, and provides various control functions.



Functions of the UART

- During transmission the UART:
  - Converts data from parallel to serial.
  - Adds the proper parity bit.
  - Adds start and stop bits.
  - Clocks the data out at a proper bit rate.
  - Notifies the microprocessor each time a character has been sent so the microprocessor can make more data available.

Functions of the UART

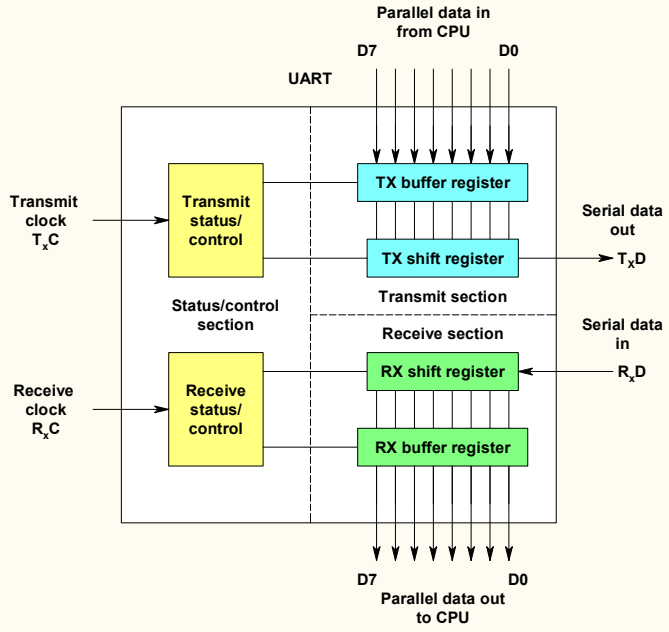
- During reception the UART:
  - Looks for the start bit to anticipate the next incoming data word.
  - Converts the incoming data from serial to parallel.
  - Checks parity.
  - Checks framing by looking for the stop bit.
  - Notifies the microprocessor each time a complete character (one byte) has been received.
- The start and stop bits are collectively referred to as framing bits. The absence of an expected stop bit is defined as a framing error.

The Three Modules of a UART

A UART consists of three main modules:

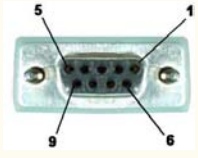
1. a transmit module, which converts data from the computer's parallel data format to the serial format used by the serial communication channel.
2. a receive module, which converts the serial data received from the communication channel to the parallel format needed by the computer's data bus.
3. a status/control module, which regulates and monitors the operation of the other two modules.

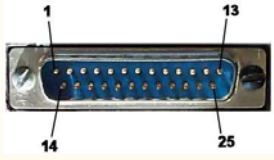
Block Diagram of the UART



RS-232C

- TTL logic signals are not what is transmitted between equipment such as modems, computers, and terminals.
- One standard that specifies the signals for asynchronous communication is the RS-232C standard.
- The RS-232C specifies 25 signal pins with a male DTE (Data Terminal Equipment) and a female DCE (Data Communications Equipment) connector. Most of these signals are not used in most applications, so a 9 pin subset is used (popularized by the IBM PC or compatible).
- Actually in our 8051 interfacing case, only three wires are used. They are the Transmit (TxD), Receive (RxD) and GND signals.





Important Asynchronous Signals in RS-232C

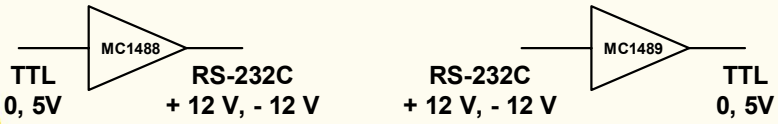
Description	PIN # (9-PIN)	From	Abbreviation
Data Leads			
Transmit Data	3	DTE	TD
Receive Data	2	DCE	RD
Power On Indicator Leads			
Data set ready	6	DCE	DSR
Data terminal ready	4	DTE	DTR
Leads that announce that an outside event has taken place			
Data carrier detect	1	DCE	CD
Ring Indicator	9	DCE	RI
Ready to send/receive handshake leads			
Request to send	7	DTE	RTS
Clear to send	8	DCE	CTS
Ground leads			
Signal ground	5		SG

RS-232C Voltage Specifications

- Consider voltages from point of view of TX and RX.
- Need to recall difference between levels on DATA and CONTROL lines.
  - For data
    - logic 1 is a -ve, logic 0 is +ve.
  - For controls
    - Asserted is +ve, de-asserted is -ve.
- A logic high or mark is between - 3V and - 15V under load.
- A logic low or space is between + 3V and + 15 V under load.
- Typically, +12V and -12 volts are used. The voltage swing is > the TTL 5 volts for noise immunity.

RS232 Voltage Level Convertors

- UARTs are generally used with RS-232 hardware that converts the 5V to -12V and the 0V to +12V and vice-versa.
- If -12 V and +12 V are available in the system, then an 1488 and 1489 are generally used as driver and receiver.



RS232 to TTL Converter with MAX-233

- Nowadays the MAX233 is a good solution because it generates its own -12 V and +12 V from 0 and 5 V, respectively, and because it integrates both driver and receiver.
- The MAX233 basically performs the same job as the MAX232 but eliminates the need for the capacitors.
- However, the MAX233 chip is much more expensive than the MAX232 and they are not pin compatible. You cannot take a MAX232 out of a board and directly replace it with a MAX233.

Interfacing MAX233 with the 8051