ECEN 4856

Lecture 5 Board I/O

Material to cover

- VHDL Reference Items
- Board I/O

VHDL – IF statement

The general form of an IF statement.

```
IF expression THEN
      statement ;
      {statement ;}
ELSIF expression THEN
      statement ;
      {statement ;}
ELSE
      statement ;
      {statement ;}
END IF ;
```

Example:

```
IF Sel = '0'THEN
        f <= x1 ;
ELSE
        f <= x2 ;
END IF ;
```

VHDL – CASE statement

The general form of an CASE statement.

```
CASE expression IS
      WHEN constant value >
       statement ;
       {statement ;}
      WHEN constant value >
       statement ;
       {statement ;}
      WHEN OTHERS >
       statement ;
       {statement ;}
END CASE ;
```

Example:

```
CASE Sel IS
     WHEN '0'=>
          f <= x1 ;
     WHEN OTHERS =>
          f <= x2 ;
END CASE ;
```

VHDL – FOR loop

The general forms of FOR-LOOP

```
[loop label:]
FOR variable name IN range LOOP
      statement ;
      {statement ;}
END LOOP [loop label] ;
```

Example:

```
process (A)
begin
        Z <= "0000";
        for I in 0 to 3 loop
              if (A = I) then
                    Z(I) <= '1';
              end if;
        end loop;
end process;
```

VHDL – WHILE loop

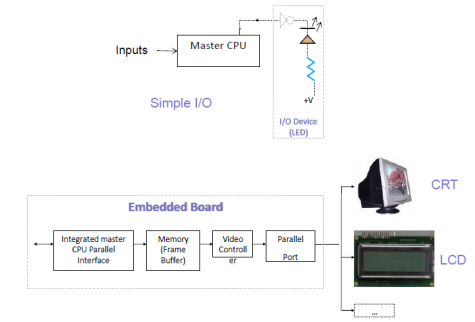
The general forms of WHILE-LOOP

```
[loop label:]
WHILE boolean expression LOOP
     statement ;
     {statement ;}
END LOOP [loop label] ;
```

Example:

```
process (A)
      variable I : integer range 0 to 4;
begin
      Z <= "0000";
      I := 0;
      while (I <= 3) loop
         if (A = I) then
            Z(I) <= '1';
         end if;
         I := I + 1;
      end loop;
end process;
```

Interfaces

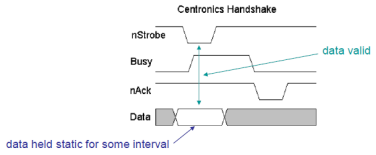


Common Implementations of Computer Interfaces

- Parallel port (8 bits per shot)
- Serial (RS-232, RS-485, USB)
  - usually asynchronous
- GPIB (IEEE-488) parallel
  - General Purpose Interface (or Instrument) Bus
  - originally HPIB; Hewlett Packard
- DAQ card (data acquisition)
  - like national instruments A/D, D/A, digital I/O
- CAMAC
  - Computer Automated Measurement And Control
- Ethernet
  - Widely implemented LAN protocol

Parallel Port

- Primarily a printer port on the PC (obsolete)
  - goes by name LPTx: line printer
  - usually LPT1
- 8 data bits
  - with strobe to signal valid data
  - can be fast (1 Mbit/sec)
- Other control and status bits for (printer) communication
- Some CNC Machines use this port for motor control

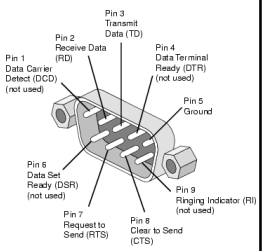


Parallel Port - Pin out

Pin No (D-Type 25)	Pin No (Centronics)	SPP Signal	Direction In/out	Register	Hardware Inverted
1	1	nStrobe	In/Out	Control	Yes
2	2	Data 0	Out	Data	
3	3	Data 1	Out	Data	
4	4	Data 2	Out	Data	
5	5	Data 3	Out	Data	
6	6	Data 4	Out	Data	
7	7	Data 5	Out	Data	
8	8	Data 6	Out	Data	
9	9	Data 7	Out	Data	
10	10	nAck	In	Status	
11	11	Busy	In	Status	Yes
12	12	Paper-Out PaperEnd	In	Status	
13	13	Select	In	Status	
14	14	nAuto-Linefeed	In/Out	Control	Yes
15	32	nError / nFault	In	Status	
16	31	nInitialize	In/Out	Control	
17	36	nSelect-Printer nSelect-In	In/Out	Control	Yes
18 - 25	19-30	Ground	Gnd		

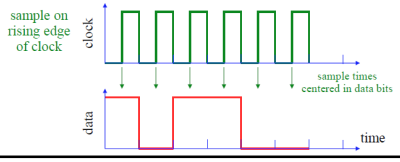
The Serial Port

- Most PCs have a DB9 male plug for RS-232 serial asynchronous communications
  - often COM1 on a PC
- In most cases, it is sufficient to use a 2- or 3-wire connection
  - ground (pin 5) and either or both receive and transmit (pins 2 and 3)
- Other controls available, but seldom used
- Data transmitted one bit at a time, with protocols establishing how one represents data
- Slow-ish (most common is 9600 bits/sec)



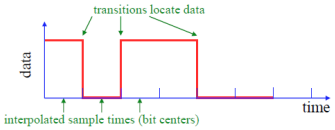
Serial Port cont.

- With separate clock and data, the transmitter *gives* the receiver timing on one signal, and data on another
- Requires two signals (clock and data): can be expensive
- Data values are arbitrary (no restrictions)
- Used by local interfaces: V.35, (synchronous) EIA-232, HSSI, etc.
- As distance and/or speed increase, **clock/data skew** destroys timing



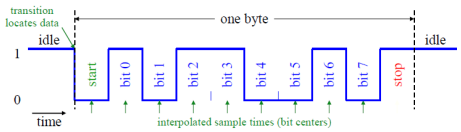
Serial-Timing

- Most long-distance, high speed, or cheap signaling is **self timed**: it has no separate clock; the receiver recovers timing from the signal itself
- Receiver knows the *nominal* data rate, but requires **transitions** in the signal to locate the bits, and interpolate to the sample points
- Two General Methods:
  - Asynchronous: data sent in short blocks called **frames**
  - Synchronous: continuous stream of bits
- Receiver *tracks* the timing continuously, to stay in synch
- Tracking requires sufficient **transition density** throughout the data stream
- Used in all DSLs, DS1 (T1), DS3, SONET, all Ethernets, etc.

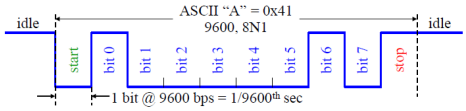


Serial - Asynchronous

- Asynchronous
  - technical term meaning "whenever I feel like it"
- Start bit is always 0. Stop bit is always 1.
- The line "idles" between bytes in the "1" state.
- This guarantees a 1 to 0 transition at the start of every byte
- After the leading edge of the start bit, if you know the data rate, you can find all the bits in the byte



Asynchronous cont.



- 4 asynchronous communication parameters must agree:
  - **Data rate**: Speed at which bits are sent, in bits per seconds (bps)
  - **Number of data bits**: data bits in each byte; usually 8 (old stuff often used 7)
  - **Parity:** An error detecting method: None, Even, Odd, Mark, Space
  - **Stop bits:** number of stop bits on each byte; usually 1. {Rarely 2 or (more rarely) 1.5: just a minimum wait time: can be indefinite }

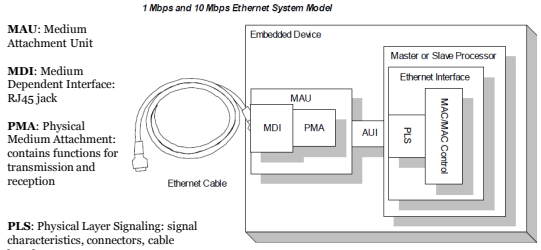
The Serial Port

- RS-232 is an electrical (physical) specification for communication
  - idle, or "mark" state is logic 1;
    - -5 to -15 V (usually about -12 V) on transmit
    - -3 to -25 V on receive
  - "space" state is logic 0;
    - +5 to +15 V (usually ~12 V) on transmit
    - +3 to +25 V on receive
  - the dead zone is from -3 V to +3 V (indeterminate state)
- Usually used in asynchronous mode
  - so idles at -12; start jumps to +12; stop bit at -12
  - since each packet is framed by start/stop bits, you are guaranteed a transition at start
  - parity (if used) works as follows:
    - even parity guarantees an even number of ones
    - odd parity guarantees an odd number of ones in the train

Other Serial Devices - Ethernet

- Ethernet:
  - IEEE Standard 802: Has many subparts for different components (ie 802.11 for wireless)
  - Communicates in data packets trnasporting frames –blocks of data sent and delivered individually
  - Cabling can be twisted-pairs or fiber (co-ax sometimes)
  - Speeds from 1Mb/s to 100Gb/s

Ethernet Components



MAU: Medium Attachment Unit

MDI: Medium Dependent Interface: RJ45 jack

PMA: Physical Medium Attachment: contains functions for transmission and reception

PLS: Physical Layer Signaling: signal characteristics, connectors, cable length

MAC: Medium Access Control: initiates the transmission of data

Other Serial - USB

- Universal Serial Bus (USB)
  - Communicates on **pipes** (logic channel / connection between the host and device)
    - 2 types of pipes
      - Message: bi-directional, used for short messages
      - Stream: unidirectional transferring data in isochronous or bulk transfers
  - Different connectors on each side A & B
  - Speeds of 1.5Mbits/s (USB 1) up to 10Gbits/s (USB 3.1)
  - Host device controls the transfers

References and Links

- http://www.altera.com/devices/fpga/new-to-fpgas/resource-center.html