Bus Systems & Address Decoding

- Address Decoding
- Memory-Mapped I/O
- Multiplexed Buses

How does the system bus work?

What devices are we reading/writing?

How does each device know it is being accessed?
Memory-mapped Bus  
- All components known non-overlapping address space
- Memory-mapped I/O
  - I/O is controlled through memory-mapped ports

Memory-mapped IO  
- IO is controlled through memory-mapped locations

How would we build the hardware required for PORT T?
**Memory-mapped I/O**
- All components known non-overlapping address space
- Memory-mapped I/O
  - I/O is controlled through memory-mapped locations

**Latches**
- Load input data value, D, into latch whenever the latch enable, LE, is high
- Retains latched value when latch enable is low
- Will output the latched value on the output, Q, whenever the output enable, OE', is active (low)
- Will output high impedance when output enable is not active (high)
- The output of this latch uses a tri-state buffer

**Memory-mapped I/O - Output Port**
- First Step: Design output port for PT0
  - Address for PT0 is 0x0240
  -Latch data written into latch when ECLK is high
  -Output of latch will be output to PT0

**Load data when address on bus is for port**
**Output should be enabled**

**Design logic for latch enable**

**Data Input**
**Data Output**

**Memory-mapped I/O - Output Port**
- Latch enable should be asserted if the ADDR is 0x0240, RD/WR is low, and ECLK is high
Memory-mapped IO - Input Port
- First Step: Design input port for PT0
  - Address for PT0 is $0240
  - Read data from latched input value when ECLK is high
  - Input of latch will be enabled if latch will be output to PT0

Always load data (or we could load data when ECLK is low)
Output should be enabled when address on bus is for port

Design logic for output enable

Memory-mapped IO - Input/Output Port
- Combine to input port and output port designs

We can’t read from the port if we are always outputting a value!
Memory-mapped I/O - Input/Output Port
- Combine to input port and output port designs

Address Decoding
(Addressing Scheme)
- ADDR[15]...
- ADDR[0]

Input Latch
LE
DQ
OE
RD/WR

Output Latch
LE
DQ
OE
RD/WR

Multiplexed Bus
- Time-multiplexed bus implementation uses a single set of bus lines for both data and address

Multiplexed Bus
ECLK
RD/WR
ADDRA[15:0]
DATAA[15:0]

Output ADDR when ECLK is Low
Read/Write DATA when ECLK is high

Multiplexed Bus
ECLK
RD/WR
ADDRA[15:0]
DATAA[15:0]
Memory-mapped IO – Input/Output Port

Address Decoding

DATA[0]

DDRT[0]

How can we modify this for a multiplexed bus?