Introduction to Digital Logic

- Exam 3 Review

Doug Shook

Topics

- High Level State Machines
  - Differences from FSMs
- Creating HLSMs
- Datapaths
  - Components
- RTL Design
  - Timing
  - VHDL

Topics

- Carry-lookahead adders
  - Contrast with carry-ripple
- Multipliers
- Register Files
- ALUs
- CPUs

Review Questions

- What are the differences between an HLSM and an FSM?
- What is a datapath? What components might it contain?
- Describe the differences between an HLSM circuit and an FSM circuit.

Review Questions

- What's the difference between a carry-lookahead adder and a carry-ripple adder?
- Describe how an array style multiplier works.
- What is a tri-state buffer?
- What is a port?
- How does a register file work?
- What components does an ALU contain?
  - A CPU?

Review Questions

- Create a HLSM for a queue. A queue has three inputs: dataIn, enqueue and dequeue. It has three outputs: empty, full, and dataOut. A queue behaves similarly to a stack, except a queue is First In First Out (FIFO). Your queue should be able to read (dequeue) and write (enqueue) at the same time.

- Construct a datapath for your HLSM

- Construct the FSM for your HLSM

- Write VHDL to implement your HLSM

Review Questions

Create an HLSM that looks for the minimum value present on the input A and counts the number of clock periods when this minimum value is present. It has two outputs, minVal and minCount. So for example, if the input sequence on A is 57, 85, 23, 34, 36, 23, 46, 23 then the sequences of values on the two two outputs will be 57, 57, 23, 23, 23, 23, 23, 23 and 1,1,1,1,1,2,2,3.

Construct the datapath/FSM.

Write VHDL to implement the circuit.

Review Questions

- Construct a register file circuit that has two write ports and one read port.

- Register files sometimes include a *bypass*. This bypass checks the address of the read port and compares it to the address of the write port. If they are the same, the input data is immediately propagated to the output of the read port, "bypassing" the register.
  - How is this helpful?
  - Update your register file circuit to include this.

Review Questions

- Construct an ALU circuit that implements the following operations:

$F_{2:0}$	Function
000	A AND B
001	A OR B
010	A + B
011	not used
100	A AND $\overline{B}$
101	A OR $\overline{B}$
110	A – B
111	SLT