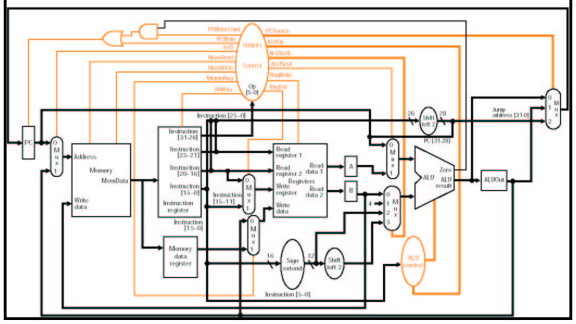
More Multicycle and Microprogramming

3-24-2003

Opening Discussion

- What did we talk about last class?
- Have you seen anything interesting in the news?
- For the minute essay I asked you how we might improve on the multicycle approach and many of you gave the answer I was looking for, better parallelism. We like to have everything we etch into silicon being used every clock cycle. Some current research looks at turning off parts that aren't in use for given periods of time. This is to reduce energy consumption, not boost speed.

Complete Multicycle Datapath



Multicycle Control Signals

- We now have 10 single bit signals and 3 2-bit signals.

RegDest	Write goes to rt or rd
RegWrite	Is value written to reg?
ALUSrcA	ALU Op1 from A or PC
MemRead	Mem Data to mem reg
MemWrite	Is value written in mem?
MemtoReg	Reg file input from mem or ALUOut?
IorD	PC or ALU for mem addr?
IRWrite	Is IR written?
PCWrite	Is PC written?
PCWriteCond	Write PC if ALU Zero on

2-bit Control Signals

ALUOp	00	Add
	01	Subtract
	10	Funct field determines op
ALUSrcB	00	Comes from B
	01	Is 4
	10	Sign extended 16-bits of IR
	11	Sign extended 16-bits of IR left shifted 2
PCSource	00	ALU output (PC+4)
	01	ALUOut (branch) sent to PC
	10	Jump target

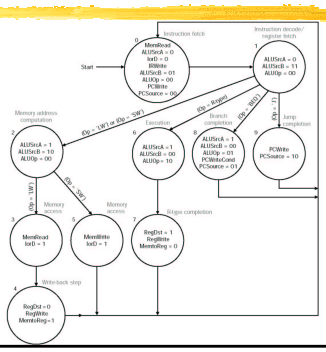
Tracing Execution Steps

- Last time we looked at the steps that we could use in our multicycle implementation. We aren't going to take class time to trace through this explicitly in the diagram and see what happens in every step. You should do that on your own to help you see how the multicycle implementation really works and what the controller must do. It's section 5.4 in the text.

Finite State Machines

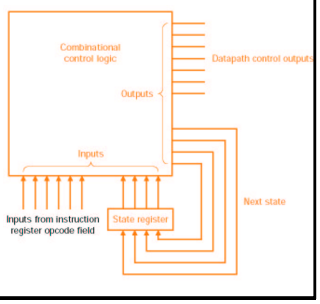
- One way of describing the control of our chip is with a finite state machine or FSM. These are used extensively in theoretical CS as they are machines that can produce and recognize regular expressions.
- They are extremely simple. They have states (nodes) that are connected to one another by transition edges. The machine is in one state at a time and moves to new states across the edges.

Complete Controller FSM



Abstract FSM Controller

- Moore vs. Mealy machines.
- Do outputs depend on inputs or just current state?



Microprogramming

- A problem with the use of FSMs is that when the system gets more complex (like the full MIPS with 100+ instructions) they can involve hundreds or thousands of states. This makes a diagram notation very difficult to understand.
- To get around this we turn to our programming background and the use of microinstructions to specify what control signals should be specified.

Minute Essay

- Assume you have an FSM that writes on states (put letters in the states). Draw the FSM that can generate the language $(10)^*$.
- Remember to turn in assignment #4 to me today. Paper submissions are fine for this one if you want. If you submit text e-mail please make sure your name is in the text of every e-mail.