Study Session 3

Brandon Thompson

November 18, 2019

1 Single cycle

`lw $t3, 4($t0)`
`lw` instruction reads from memory, writes into register.
Read from `4($t0)` write to `$t3` register file.

1.1 State Elements

PC
Instruction Memeory
Register File
Data memory

1.2 Fetch

1. PC State element.
2. Instruction memory state element.
3. connect PC output to A of instruction memory.

1.3 Decode

1. Register File

2. Output of instruction memory goes to A1(`$t0`), A3(`$t3`), and sign extension for offset (`4`).

1.4 Execution

Add RD1 and sign extension with ALU.

ALU output is passed to data memory state element.

Data memory output is passed to register file WD3.

Enable register writing (WE3 = 1).

increment PC by 4.
Decode and Execution overlap.

2 Multi-cycle

Implement `lw $t3, 4($t0)`

2.1 State Elements

PC
Instruction / Data memory
Register File

2.2 Fetch

1. PC output fed into 2 input multiplexer.
2. Output of multiplexer passed to memory (address)
3. RD from memory passed to register.

2.3 Decode

1. From register to A1 of register file (`$t0`), sign-extension (`4`), and A3 (`$t3`).

2.4 Execution

1. output of Register File saved into register
2. Multiplexer with PC and output of register.
3. Multiplexer output passed into ALU.
4. Another multiplexer chooses between sign-extended value (`4`) and PC counter (4).
5. Save into register
6. Pass from register into PC multiplexer.
7. RD from data memory passed into writing register.
8. WE3 set to 1.
9. pass from writing register into WD3.