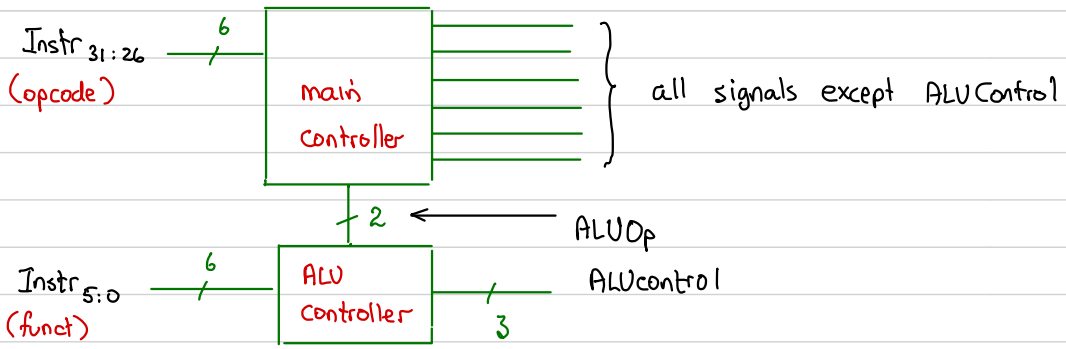
The control unit (continued)



ALUop rules: 00 – ALU to add (lw/sw)
01 – subtract (beq)
10 – use "funct" field

Control-signal examples:

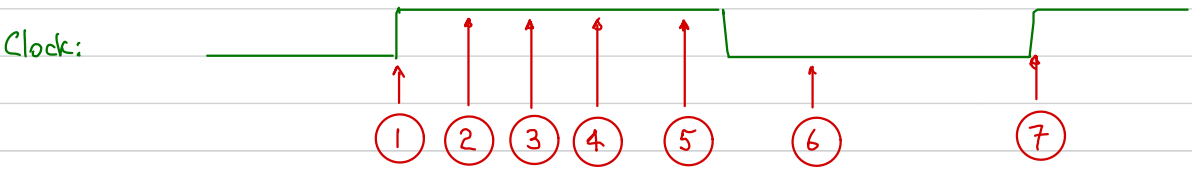
Signal		Instruction				
		SLT		LW		BEQ
MemtoReg	0	(use ALUResult)	1	(use D-mem)	x	(not using)
MemWrite	0		0		0	
Branch	0		0		1	
ALUSrc	0	(use GPR)	1	(use offset)	0	(use GPR)
RegDst	1	(use $Instr_{15:11}$)	0	(use $Instr_{20:16}$)	x	
RegWrite	1	(write answer)	1		0	(save nothing)
ALUOp	10	(use "funct")	00	(add)	01	(force subtract)
ALUcontrol	111	(slt)	010	(add)	110	(subtract)

Timing in a single-cycle processor

Example: Timing for:

lw $s1, 0x1234 ($s0)

– instruction address 0x0040_00a8



(1) Active edge – do all updating required for previous instruction
- PC becomes 0x0400_00a8
- R-type, lw : R-file update occurs
- sw : D-mem update occurs

(2) 32-bit lw instruction read from I-mem. PC's adder calculates 0x0040_00ac (stored at time 7)

(3) Main control unit senses lw from $Instr_{31:26}$; produces control signals

(4) R-file gives contents of $s0 and $s1 at RD1 and RD2 ($s1 not used). ALU Decoder produces 010 for addition.

(5) ALUResult becomes $s0 + 0x0000_1234.

(6) D-mem produces contents at address ($s0 + 0x0000_1234).

(7) Active edge of the clock – update.
- Contents of $s1 in R-File changed to D-mem word read in step 6.
- PC changes to 0x0040_00ac.

Detailed timing analysis

Assumptions:
- Reading the R-file ($t_{RF\text{-}read}$) much slower than sign-extender and mux.
- also much slower than control unit

The critical path for the lw instruction

- Once the PC read, the critical path is through 5 units

$$T_5 = (t_{mem}) + (t_{RF\text{-}read}) + (t_{ALU}) + (t_{mem}) + (t_{mux})$$

I-mem → R-file → ALU → D-mem → MUX

- Must include the read time of the PC (t_{pcq-pc}) and the setup time of the R-file ($t_{RF-setup}$)

For reliable operation:

$$T_c \geq t_{pcq-pc} + T_5 + t_{RF-setup}$$

(T_c: clock period)

Single-cycle processor is slow. Idea:

- Do fewer operations per cycle, allow operations to occur simultaneously
- Speed up the clock!

Pipelined processor design

For the single-cycle processor:

- Instructions are dealt with one at a time.
  - This is not true in a modern processor

Doing the laundry (analogy)

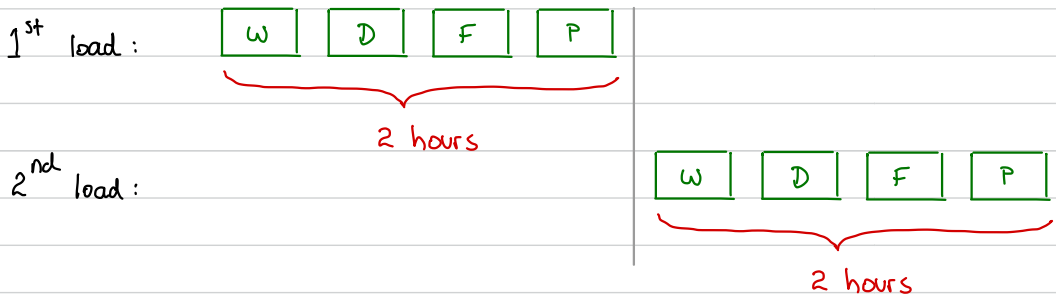
You have many loads of laundry with these resources:

- one washing machine
- one dryer
- one "folding unit" (you)
- one "putting away" unit (a friend)

Each unit takes 30 minutes for one load.
Suppose there are 4 loads.

Method 1: Sequentially

W = wash F = fold
D = dry P = put away



In sequence : 1 → 2 → 3 → 4

8 hours

Note resources are idle 75 % of the time !