ECE154A — Discussion 06

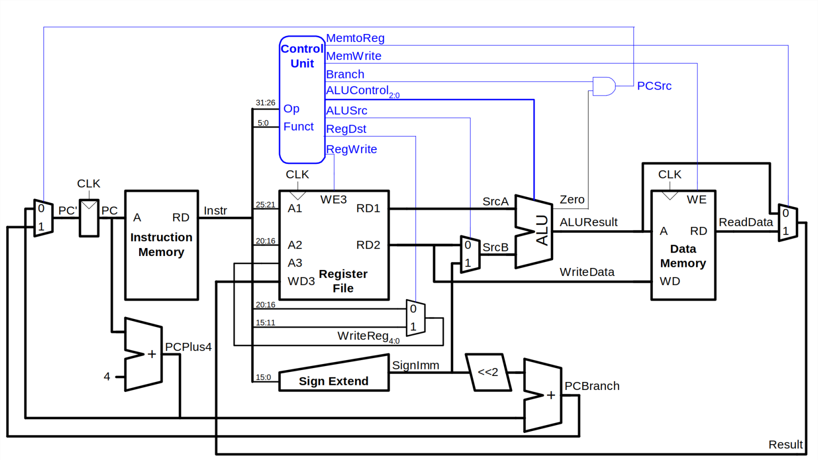
George Higgins Hutchinson

November 5, 2021

Keep your eyes open for. . .

- PSet 3: due Friday, November 5
- Lab 3: due Friday, November 5
- PSet 4: due Friday, November 12
- Lab 4: due Monday, November 15

MIPS single-cycle, no absolute addressing



Example Program

Assume all registers initialize zeroed-out. What data flow is necessary to correctly run each of these?

```
00   addi $2 $0 10
04   sll $2 $2 2
08   lw $3 0($2)
0c   bne $3 $0 +4
10   sub $4 $3 $2
14   add $4 $4 $4
```

Memory in your processor (Lab4)

```
// Instruction memory (already implemented)
module imem(input    [5:0]  a,
            output   [31:0] rd);

  reg [31:0] RAM[63:0];

  initial
    begin
      $readmemh(”memfile.dat”,RAM);
      // initialize memory with test program.
    end

  assign rd = RAM[a]; // word aligned
endmodule
```

Memory in your processor (Lab4)

When can you read from this? What can you write to it?

When do you need to read/write from the data memory? The regfile?

Control-Datapath interface (Lab4)

Note: the names that I think are helpful don't totally match H&H.

signal	value
RegFileWriteSource	
MemWriteEnable	
Branch	
Jump	
ALUSelect[2:0]	
ALUSrcBSel	
RegDstSel	
RegFileWriteEnable	

Fill out this table for a few different types of instructions, then try to generalize.

Control-Datapath interface (for addi)

Semantic answers here; but remember that they're representing numbers on wires.

signal	value
RegFileWriteSource	ALU
MemWriteEnable	false
Branch	false
Jump	false
ALUSelect[2:0]	add
ALUSrcBSel	imm
RegDstSel	instr[20:16]
RegFileWriteEnable	true