

probs # 4.1, 4.5, 4.6, 4.7

4.1 Consider the following instruction:

Instruction: AND Rd, Rn, Rm

Interpretation: $\text{Reg}[\text{Rd}] = \text{Reg}[\text{Rn}] \text{ AND } \text{Reg}[\text{Rm}]$

4.1.1 [5] <§4.3> What are the values of control signals generated by the control in Figure 4.10 for this instruction?

	Step 1 - and	Step 2 - into register
regWrite	0	1
ALUFC	0	0
ALUOP	0000	0000
memWrite	0	0
memRead	X	X
memtoReg	X	0

4.1.2 [5] <\$4.3> Which resources (blocks) perform a useful function for this instruction?

4.1.3 [10] <\$4.3> Which resources (blocks) produce no output for this instruction? Which resources produce output that is not used?

4.1.2) The register Bank and ALU were fundamental resources, as well as their corresponding Mux's for proper data flow.

4.1.3) Data memory has no output, b/c no mem's read out. The sign extend datapath is produced but not used

4.5 In this exercise, we examine in detail how an instruction is executed in a single-cycle datapath. Problems in this exercise refer to a clock cycle in which the processor fetches the following instruction word: 0xf8014062.

4.5.1 [5] <\$4.4> What are the outputs of the sign-extend and the "shift left 2" unit (near the top of [Figure 4.23](#)) for this instruction word?

4.5.2 [10] <\$4.4> What are the values of the ALU control unit's inputs for this instruction?

4.5.3 [5] <\$4.4> What is the new PC address after this instruction is executed? Highlight the path through which this value is determined.

$0x\ F8014062_{16}$

|||| 1000 0000 0001 0100 0000 0110 0010₂

Op code offset
→ STUR instruction

4.5.1) sign extend will extend offset to 64 bit

0100 0000 0000 0000 0000 0000 0001 0100
shift 2

0000 0000 0000 0006 0000 0000 0101 0000

4.5.2) ALU control inputs - 0010

to add data mem address to offset

4.5.3) The PC address by 4, b/c STLR is
a 4 byte instruction (as most all are)

4.5.4 [10] <\$4.4> For each mux, show the values of its inputs and outputs during the execution of this instruction. List values that are register outputs at Reg [Xn].

4.5.5 [10] <\$4.4> What are the input values for the ALU and the two add units?

4.5.6 [10] <\$4.4> What are the values of all inputs for the registers unit?

4.5.4)

- ALU Src Mux is set high so. the sign extend input is chosen over R_t. Output is fed into ALU.
- the Reg to Loc Mux inputs are 00001 and 00010. The high control setting outputs the latter.
- Mem2Reg sees DataMemory & ALU, yet output doesn't matter

- the branch mux has input PC+1, and 2 bit LSL sign extend inputs. Output is update PC.
- $R_n - R_3$

$$R_t - R_2$$

U.S.5) The ALU sees both R_n & the sign extended offset.

One adder sees current PC & U.

Another sees PC & sign extend's shift left 2 output.

U.S.6)

Read Register 1 sees R_n to be given to ALU
 (00011)

Read Register 2 sees R_t (00010)

Write Register sees R_t

4.6 Section 4.4 does not discuss I-type instructions like ADDI or ANDI.

4.6.1 [5] <\$4.4> What additional logic blocks, if any, are needed to add I-type instructions to the CPU shown in Figure 4.23? Add any necessary logic blocks to Figure 4.23 and explain their purpose.

4.6.2 [10] <\$4.4> List the values of the signals generated by the control unit for ADDI. Explain the reasoning for any “don’t care” control signals.

4.b.1) To include ADD we need to connect a register input & immediate constant to be inputted into ALU. The register is read in normal but we'll have to modify structure to get the immediate field of instruction into ALU. The sign extend can take instruction's immediate and extend to output into the ALU.

4.b.2) To achieve, aluSel must be high to read immediate into ALU. MemRead must be low and regWrite is high to make it write into input register. ALUOP is 11. Uncond & branch are low. Reg2loc doesn't matter b/c only 1 register in use. MemRead has no influence b/c output is Max'd out by Mem2Reg

4.7 Problems in this exercise assume that the logic blocks used to implement a processor's datapath have the following latencies:

I-Mem / D-Mem	Register File	Mux	ALU	Adder	Single gate	Register Read	Register Setup	Sign extend	Control
250 ps	150 ps	25 ps	200 ps	150 ps	5 ps	30 ps	20 ps	50 ps	50 ps

"Register read" is the time needed after the rising clock edge for the new register value to appear on the output. This value applies to the PC only. "Register setup" is the amount of time a register's data input must be stable before the rising edge of the clock. This value applies to both the PC and Register File.

4.7.1 [20] <\$4.4> Although the control unit as a whole requires 50 ps, it so happens that we can extract the correct value of the Reg2Loc control wire directly from the instruction. Thus, the value of this control wire is available at the same time as the instruction. Explain how we can extract this value directly from the instruction. Hints: Carefully examine the opcodes shown in [Figure 2.20](#). Also, remember that LSR and LSL do not use the Rm field. Finally, ignore STXR.

4.7.2 [5] <\$4.4> What is the latency of an R-type instruction (i.e., how long must the clock period be to ensure that this instruction works correctly)?

4.7.3 [10] <\$4.4> What is the latency of LDUR? (Check your answer carefully. Many students place extra muxes on the critical path.)

4.7.4 [10] <\$4.4> What is the latency of STUR? (Check your answer carefully. Many students place extra muxes on the critical path.)

4.7.5 [5] <\$4.4> What is the latency of CBZ?

4.7.6 [5] <\$4.4> What is the latency of B?

4.7.7 [5] <\$4.4> What is the latency of an I-type instruction?

$$\begin{aligned}4.7.2) \quad & 30 + 250 + 50 + 25 + 150 + 25 + 200 + 25 + 20 \\& = 775 \text{ ps}\end{aligned}$$

$$4.7.3) \quad 1050$$

$$4.7.4) \quad 800 \text{ ps}$$

$$4.7.5) \quad 725 \text{ ps}$$

$$4.7.6) \quad 525 \text{ ps}$$

$$4.7.7) \quad 750 \text{ ps}$$