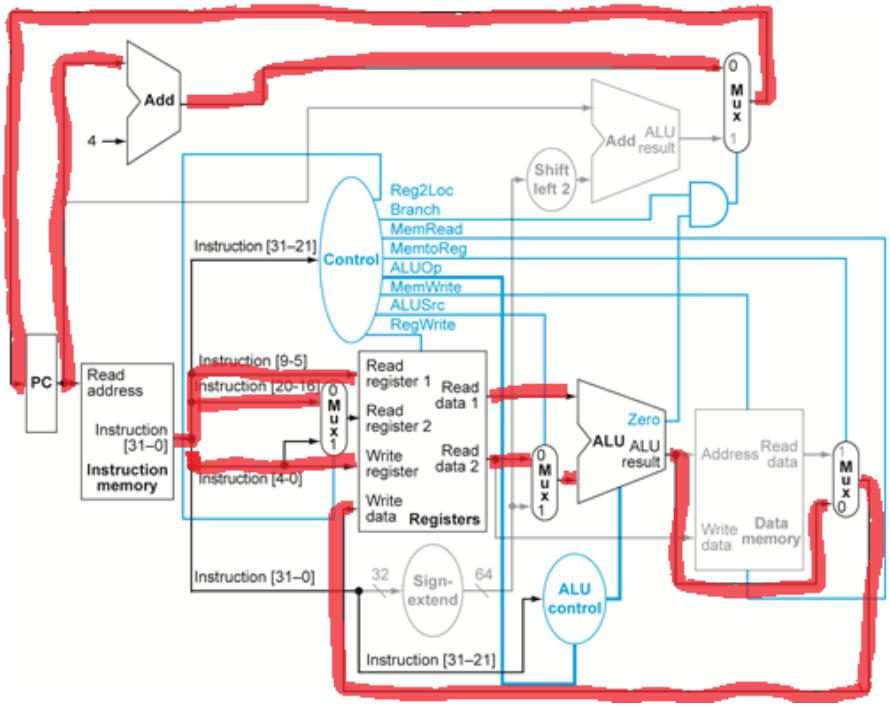
Homework #3

Due Jan. 20

5 points each

1. Consider the following instruction:
   Instruction: ORR Rd, Rn, Rm
   Interpretation: Reg[Rd] = Reg[Rn] ORR Reg[Rm]
   (a) Show the dataflow for this instruction using dash lines on the below figure for LEGv8 datapath.

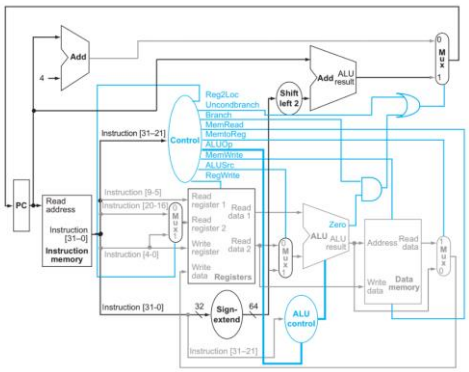


* had to do in paint -- kept breaking *

(b) What are the values of control signals generated by the control unit for this instruction?

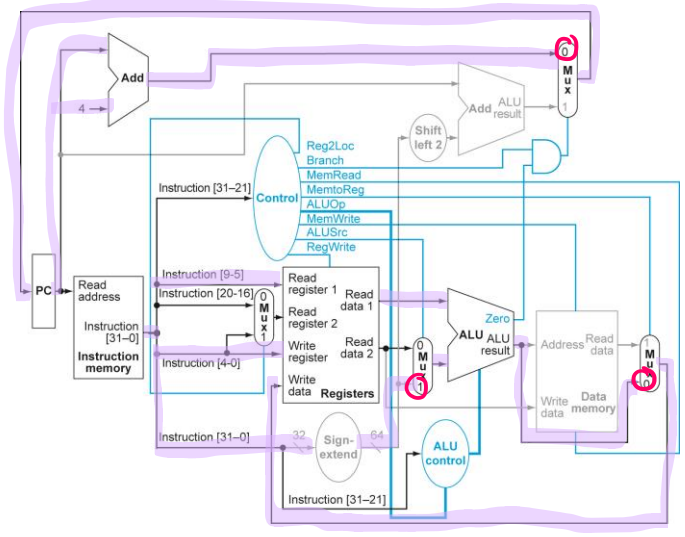
Reg2Loc	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	ALUOp1	ALUOp0
0	0	0	1	0	0	0	1	0

2. We do not discuss the datapath for I-type instructions like ADDI or ANDI.
   (a) What additional logic blocks, if any, are needed to add I-type instructions to the CPU shown in the following figure (Figure 4.23 in the textbook)? Add any necessary logic blocks to it and explain their purpose.



No additional blocks are needed because we have a sign extender (can extend an intermediate value and pass it to our other components for any arithmetic operation)

(b) Show the dataflow for this instruction using dash lines on the following figure



(c) List the values of the signals generated by the control unit for ADDI. Explain the reasoning for any "don't care" control signals.

Reg2Loc	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	ALUOp1	ALUOp0
X	1	0	1	0	0	0	1	0

↓ Reg2Loc is a don't care because the Mux ignores the "read data 2" output

3. Consider the following instruction mix:

R-type	I-Type	LDUR	STUR	CBZ	B
24%	28%	25%	10%	11%	2%

(a) What fraction of all instructions use data memory?

LDUR+STUR = 25% + 10% = 35%

(b) What fraction of all instructions use instruction memory?

Every instruction = 100%

(c) What fraction of all instructions use the sign extend?

Everything but R-type = 28% + 25% + 10% + 11% + 2% = 76%

4. When silicon chips are fabricated, defects in materials (e.g., silicon) and manufacturing errors can result in defective circuits. A very common defect is for one signal wire to get "broken" and always register a logical 1. This is often called a "stuck-at-1" fault.

(a) Which instructions fail to operate correctly if the MemToReg wire is stuck at 1?

LDUR is broken

(b) Which instructions fail to operate correctly if the ALUSrc wire is stuck at 1?

LDUR, STUR, I-type is broken

(c) Which instructions fail to operate correctly if the Reg2Loc wire is stuck at 1?

CBZ, CBNZ, STUR is broken

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 STUR X5, [X6, #12].

(a) What are the outputs of the sign-extend and the "shift left 2" unit (near the top of datapath) for this instruction word?

64-bit extension

STUR X5, [X6, #12]

D-Format

11111000000 000001100 00 00110 00101

offset

0000 0000 0000 0000 0000 0000 0000 1100

b→h = 0xC

shift left 2

0000 0000 0000 0000 0000 0000...0011 0000

↑ assume 64bit :)

(b) What are the values of the ALU control unit's inputs for this instruction?

ALUop = 010

(c) What is the new PC address after this instruction is executed?
Highlight the path through which this value is determined.

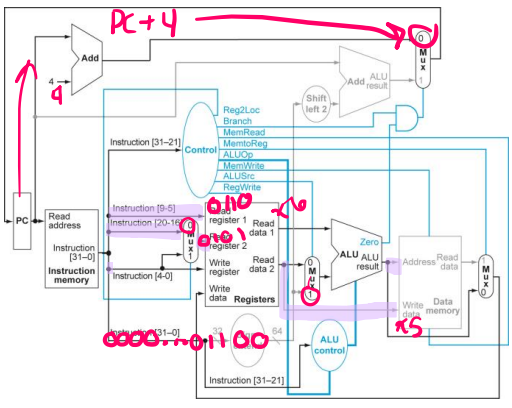
PC = PC + 4

(d) 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].

(e) What are the input values for the ALU and the two add units?

PC+4 & X6+12

(f) What are the values of all inputs for the registers unit?



STUR X5, [X6, #12]

6. 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

x2:10

"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's write(update) operation. "Shift left 2" takes 2 single gates' time.

(a) 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)?

30 + 250 + 50 + 25 + 150 + 25 + 200 + 25 + 20 = 775 ps

(b) What is the latency of LDUR? (Check your answer carefully. Many students place extra muxes on the critical path.)

30 + 250 + 50 + 150 + 200 + 250 + 25 + 20 = 975 ps

(c) What is the latency of STUR? (Check your answer carefully. Many students place extra muxes on the critical path.)

30 + 250 + 50 + 150 + 200 + 250 = 930 ps

(d) What is the latency of B?

30 + 250 + 50 + 10 + 150 + 25 + 20 = 435 ps

(e)What is the minimum clock period for this CPU?

1030 ps

7. In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:

IF	ID	EX	MEM	WB
250 ps	350 ps	150 ps	300 ps	200 ps

Also, assume that instructions executed by the processor are broken down as follows:

ALU	Branch/Jump	LDUR	STUR
45%	20%	20%	15%

(a) What is the clock cycle time in a pipelined and non-pipelined processor?

Pipelined = 350

non-pipelined = 1250

(b) What is the total latency of an LDUR instruction in a pipelined and non-pipelined processor?

Pipelined = 350 · 5 = 1750

non-pipelined = 1250

take 2nd highest II

(c) If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?

ID Stage, reduced clock cycle time = 300

(d) Assuming there are no stalls or hazards, what is the utilization of the data memory?

20% + 15% = 35%

(e) Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?

45% + 20% = 60%

8. Consider the LEGv8 code below. Assume that X1 is initialized to 11 and X2 is initialized to 22.

```
ADDI X1, X2, #5
ADD X3, X1, X2
ADDI X4, X1, #15
ADD X5, X1, X1
```

(a) What would the final values of registers X3 and X4 be if the above instructions are executed in a pipeline processor that does not handle data hazards (i.e., does not stall the pipeline or use data forwarding for data hazards)?

X3:
11 + 22 = 33

X4:
11 + 15 = 26

(b) What would the final values of register X5 be if the above instructions are executed in a pipeline processor that does not handle data hazards (i.e., does not stall the pipeline or use data forwarding for data hazards)? Assume the register file is written at the beginning of the cycle and read at the end of a cycle. Therefore, an ID stage will return the results of a WB state occurring during the same cycle. See Section 4.7 and Figure 4.51 for details.

22 + 5 = 27
27 + 27 = 54

(c) Suppose you executed the code on a version of the pipeline from Section 4.5 that handles data hazards by simply stalling the pipeline (i.e. inserting NOP instructions where necessary). Show the pipeline timing diagram below when the code is executed.

ADDI X1, X2, #5	IF	ID	EX	MEM	WB															
ADD X3, X1, X2		IF	S	S	ID	EX	ME	WB												
ADDI X4, X1, #15					IF	ID	EX	ME	WB											
ADD X5, X1, X1						IF	ID	EX	ME	WB										

(d) Suppose you executed the code below on a pipeline from Section 4.5 that uses data forwarding for handling data hazards. Show the pipeline timing diagram below:

ADDI X1, X2, #5	IF	ID	EX	MEM	WB															
ADD X3, X1, X2		IF	ID	EX	ME	WB														
ADDI X4, X1, #15			IF	ID	EX	ME	WB													
ADD X5, X3, X2				IF	ID	EX	ME	WB												

9. Consider the following loop.

```
LOOP: LDUR X10, [X1, #0]
      LDUR X11, [X1, #8]
      ADD X12, X10, X11
      STUR X12, [X1, #-8]
      SUBI X1, X1, #16
```

CBNZ X12, LOOP

(a) Assume that data and control hazards are handled by simply stalling the pipeline (i.e. inserting NOP instructions where necessary). Show the pipeline timing diagram of the code execution.

Cycle number:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LOOP: LDUR X10, [X1, #0]	IF	ID	EX	ME	WB															
LDUR X11, [X1, #8]		IF	ID	EX	ME	WB														
ADD X12, X10, X11			IF	S	S	ID	EX	ME	WB											
STUR X12, [X1, #-8]						IF	S	S	ID	EX	ME	WB								
SUBI X1, X1, #16									IF	ID	EX	ME	WB							
CBNZ X12, LOOP										IF	ID	EX	ME	WB						
2nd iteration: LDUR X10, [X1, #0]											S	S	IF	ID	EX	ME	WB			

update instruction till WB

(b) Can you reorder the code to reduce the number of stalls? If yes, show the reordered code.

Cycle number :

```
Loop : LDUR X10, [X1, #0]
       LDUR X11, [X1, #8]
       SUBI X1, X1, #16
       ADD  X12, X10, X11
       STUR X12, [X1, #8]
       CBNZ X12, LOOP
```

2nd iteration : LDUR X10, [X1, #0]

(c) Show the pipeline timing diagram of the code execution with data forwarding and assume that the branch is handled by predicting it **as taken** and the branch target address is calculated at the ID stage.

Cycle number:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LOOP: LDUR X10, [X1, #0]	IF	ID	EX	ME	WB															
LDUR X11, [X1, #8]		IF	ID	EX	ME	WB														
ADD X12, X10, X11			IF	ID	S	EX	ME	WB												
STUR X12, [X1, #-8]				IF	S	ID	EX	ME	WB											
SUBI X1, X1, #16						IF	ID	EX	ME	WB										
CBNZ X12, LOOP							IF	ID	EX	ME	WB									
2nd iteration: LDUR X10, [X1, #0]								S	IF	ID	EX	ME	WB							

(d) What is the speedup for the execution of (c) over (a)?

14/11 = 1.27 times faster

10. Compare the performance of a single-cycle datapath machine, a multi-cycle datapath machine and an ideal 5-stage pipeline machine. Assume that the single-cycle machine has a clock rate of 200MHz, and the multiple-cycle and pipelined machine have a clock rate of 1GHz. The CPI's of

load, store, ALU, branch/jumps in the multi-cycle machine are 5, 4, 4, 3, respectively. The program has the following instruction mix:

- Load: 30 percent
- Store: 10 percent
- ALU/R-format instr.: 40 percent
- Conditional Branch: 10 percent
- Jump instr.: Remaining instructions

Calculate the performance (CPU time) of the 3 machines.

Multi-cycle:

$$CPI = 0.3 \cdot 5 + 0.1 \cdot 4 + 0.4 \cdot 4 + 0.1 \cdot 3$$
$$= 1.5 + 0.4 + 1.6 + 0.3$$
$$= 3.8$$

$$CPU\ time = IC \cdot CPI / CR$$
$$= IC \cdot \frac{3.8}{1 \cdot 10^9}$$
$$= 3.8 \cdot IC \cdot 10^{-9}$$

Single-Cycle

$$CPU\ time = IC \cdot CPI / CR$$
$$= IC \cdot \frac{0.95}{200 \cdot 10^6}$$
$$= .00475 \cdot IC \cdot 10^{-6}$$
$$= 4.75 \cdot IC \cdot 10^{-9}$$

Pipelined

$$CPU\ time = IC \cdot CPI / CR$$
$$= \frac{0.95}{1 \cdot 10^9}$$
$$= 0.95 \cdot IC \cdot 10^{-9}$$