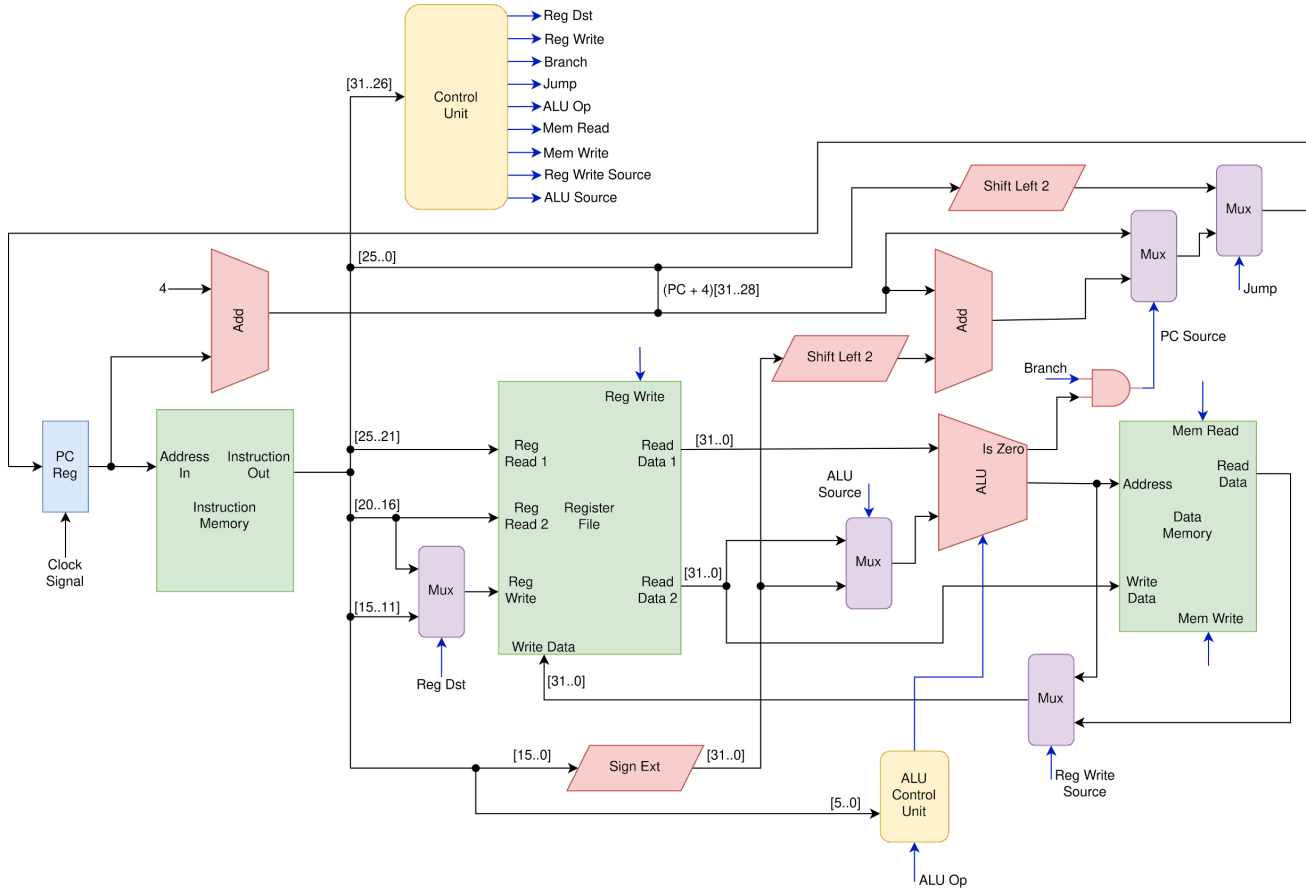


CSE490/590, Spring 2023, Homework 2

- Highlight the flow of data in the MIPS data path for the instructions:
 - add \$s1, \$s2, \$s3
 - lw \$t3, 4(\$t4)



CSE490/590, Spring 2023, Homework 2

2. Consider the following sequence of simplified instructions:

Instruction Address:	Instruction:
100	ADD
104	BEQ + 200
108	ADD
112	ADD
308	ADD

Assume the following:

- There are 5 pipeline stages — IF (Instruction Fetch), ID (Instruction Decode), EX (Execution), MEM (Memory Access), WB (Write Back).
- The second instruction, “BEQ +200”, takes the branch and jumps to the instruction at the address 308.
- A branch instruction can determine the next PC only at the EX stage.
- The CPU always speculates that it will execute the instruction at $(PC + 4)$ next.
- If a branch or jump needs to be taken, all instructions in the pipeline are killed.

Complete the time table below with the correct stages at each cycle for all instructions. Use *** for pipeline bubbles.

[illegible]

CSE490/590, Spring 2023, Homework 2

3. In the following instruction sequence for a MIPS 5-stage pipelined datapath, list the data hazards:

lw \$s2, 0(\$s1)

lw \$s1, 40(\$s6)

sub \$s6, \$s1, \$s2

add \$s6, \$s2, \$s2

or \$s3, \$s6, \$zero

sw \$s6, 50(\$s1)

CSE490/590, Spring 2023, Homework 2

4. Consider the following instruction sequence.

add r5,r2,r1

lw r3,4(r5)

lw r2,0(r2)

or r3,r5,r3

sw r3,0(r5)

a) Show the pipeline diagram after inserting NOPs to overcome data dependencies

Instruction:	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14

b) Show the pipeline diagram after inserting Data Forwarding Unit to overcome data dependencies

Instruction:	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14

c) Compare performance of (a) vs (b)

CSE490/590, Spring 2023, Homework 2

5. Assume the time for stages is

- 100ps for register read or write
- 200ps for other stages

Compare pipelined datapath with single-cycle datapath for the following instruction sequence:

lw \$1,100(\$0)

lw \$2,200(\$0)

lw \$3,300(\$0)

The following table provides how much time is spent in each stage by a specific instruction:

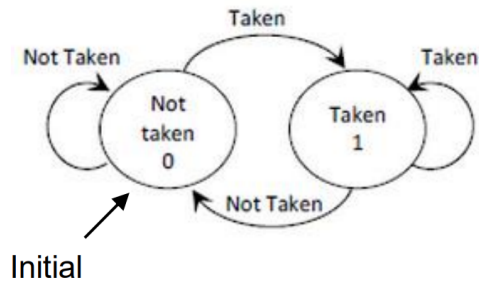
Instr	Instr fetch	Register read	ALU op	Memory access	Register write	Total time
lw	200ps	100 ps	200ps	200ps	100 ps	800ps
sw	200ps	100 ps	200ps	200ps		700ps
R-format	200ps	100 ps	200ps		100 ps	600ps
beq	200ps	100 ps	200ps			500ps

CSE490/590, Spring 2023, Homework 2

6. Consider the following instructions. Assume that the initial values for R1, R2, **R3, R4 and R5** are all 0:

```
loop:  ADDI R2, R1, -2
       BNE R5, R2, target1
       ADDI R3, R3, 1
target1: ADDI R1, R1, 1
       ADDI R4, R1, -3
       BNE R5, R4, loop
```

Assume that we have a 1-bit branch predictor that stores the result of the last branch and makes the prediction based on the result. Show the results of all predictions throughout the execution. (Use T/N to represent Taken/ Not Taken)



Branch	Predictor (T/N)	Actual Result (T/N)
1st BNE (target1)	N	T