Q1- Consider the following instruction mix:

R-type	I-type (non-lw)	Load	Store	Branch	Jump
24%	28%	25%	10%	11%	2%

a- What fraction of all instructions use data memory?
b- What fraction of all instructions use instruction memory?
c- What fraction of all instructions use the sign extend?

Q1 Solution

R-type	I-type (non-lw)	Load	Store	Branch	Jump
24%	28%	25%	10%	11%	2%

a- What fraction of all instructions use data memory?

25 + 10 = 35%. Only Load and Store use Data memory.

b- What fraction of all instructions use instruction memory?

100% Every instruction must be fetched from instruction memory before it can be executed.

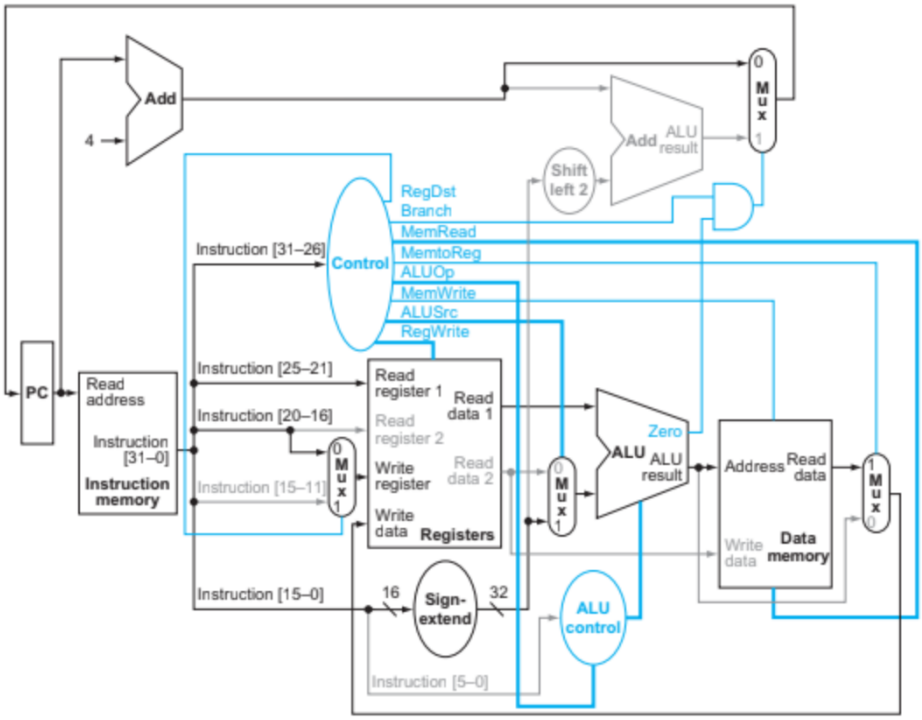
c- What fraction of all instructions use the sign extend?

28 + 25 + 10 + 11 = 74%. R-type and Jump instructions do not use the Sign extender.

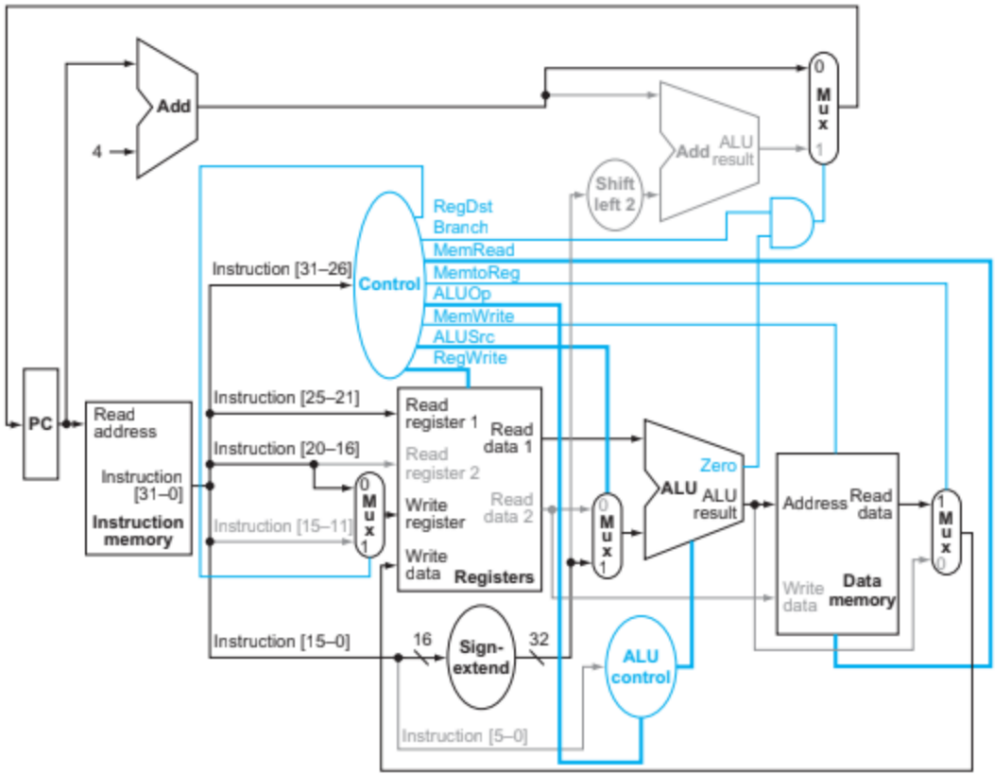
Q2

a- What additional logic blocks, if any, are needed to add I-type instructions to the CPU shown? Add any necessary logic blocks to the figure and explain their purpose.

b- List the values of the signals generated by the control unit for `addi`. Explain the reasoning for any "don't care" control signals.



Q2 Solution



a- What additional logic blocks, if any, are needed to add I-type instructions to the CPU shown? Add any necessary logic blocks to the figure and explain their purpose.

No additional logic blocks are needed.

b- List the values of the signals generated by the control unit for `addi`. Explain the reasoning for any "don't care" control signals.

```
Branch: 0
MemRead: 0
MemToReg: 0
ALUop: 10
MemWrite: 0
ALUsrc: 1
RegWrite: 1
```

Q3- Examine the difficulty of adding a proposed `lwi, rd, rs1, rs2` ("Load With Increment") instruction to MIPS.

Interpretation: `Reg[rd] = Mem[Reg[rs1]+Reg[rs2]]`

a- Which new functional blocks (if any) do we need for this instruction?
b- Which existing functional blocks (if any) require modification?
c- Which new data paths (if any) do we need for this instruction?
d- Which new signals do we need (if any) from the control unit to support this instruction?

Q3 Solution:

a- Which new functional blocks (if any) do we need for this instruction?

No new functional blocks are needed.

b- Which existing functional blocks (if any) require modification?

Only the control unit needs modification.

c- Which new data paths (if any) do we need for this instruction?

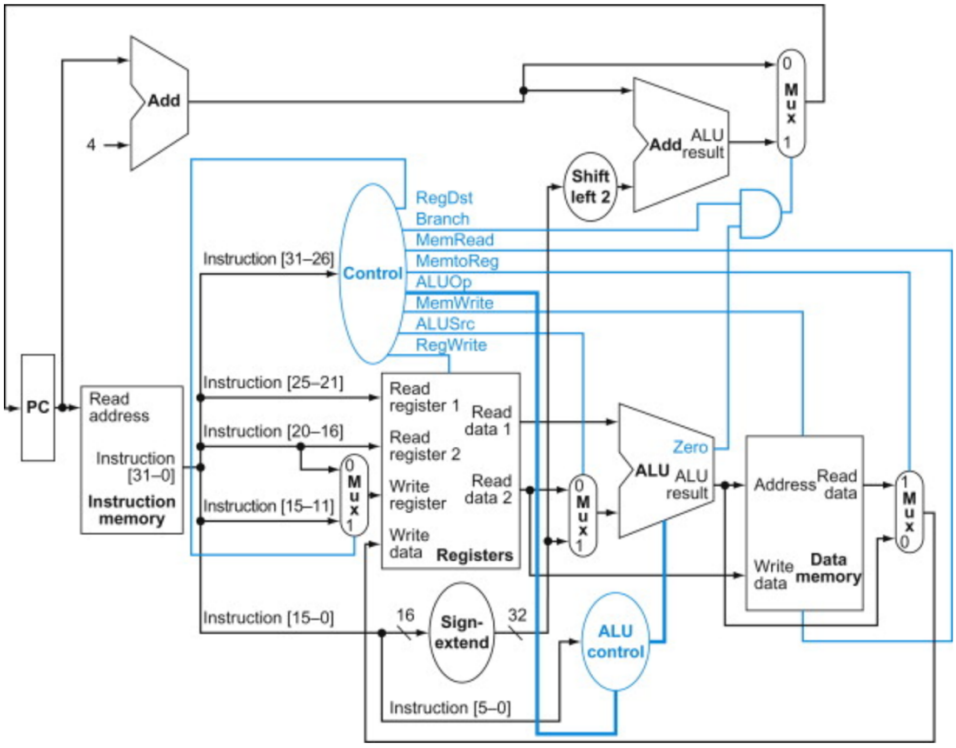
No new data paths are needed.

d- Which new signals do we need (if any) from the control unit to support this instruction?

No new signals are needed.

Q4- 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
250ps	150ps	25ps	200ps	150ps	5ps	30ps	20ps	50ps	50ps

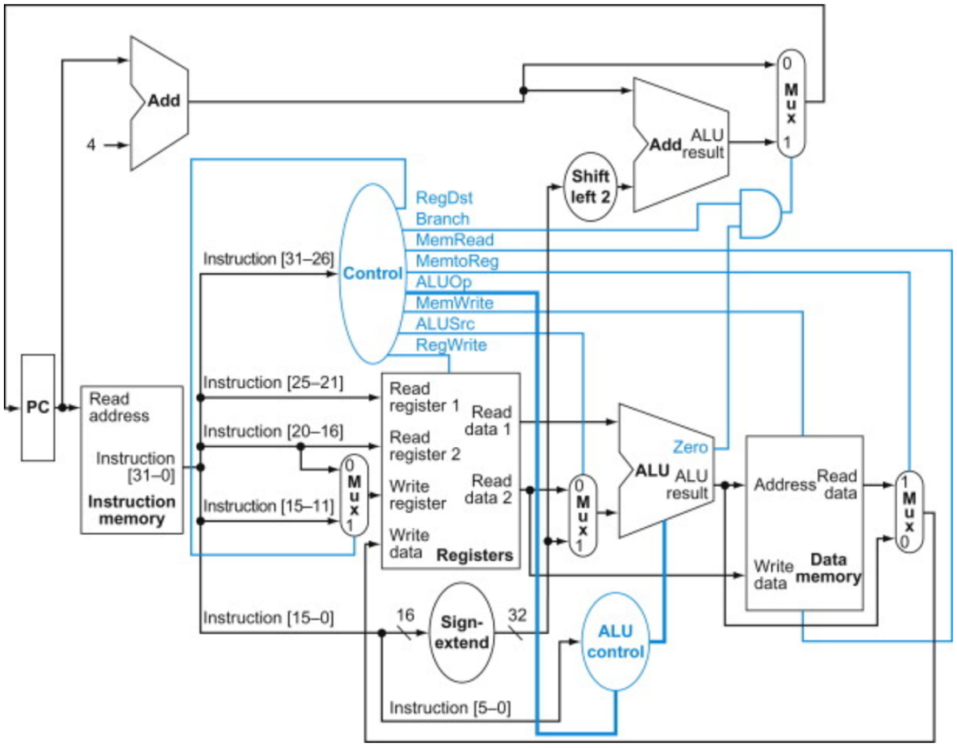


"Register read" is the time needed after the rising clock edge for the new register value to appear on the output. This value applies to the PC only. "Register setup" is the amount of time a register's data input must be stable before the rising edge of the clock. This value applies to both the PC and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

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)?
b- What is the latency of `lw`? (Check your answer carefully. Many students place extra muxes on the critical path.)
c- What is the latency of `sw`? (Check your answer carefully. Many students place extra muxes on the critical path.)
d- What is the latency of `beq`?
e- What is the latency of an arithmetic, logical, or shift I-type (non-load) instruction?
f- What is the minimum clock period for this CPU?

Q4 Solution

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



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)?

`R-type`: 30 +250 + 150 + 25 + 200 + 25 + 20 = 700ps

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

`lw`: 30 + 250 + 150 + 25 + 200 + 250 + 25 + 20 = 950ps

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

`sw`: 30 + 250 + 150 + 200 + 25 + 250 = 905

d- What is the latency of `beq`?

`beq`: 30 + 250 + 150 + 25 + 200 + 5 + 25 + 20 = 705

e- What is the latency of an arithmetic, logical, or shift I-type (non-load) instruction?

`I-type`: 30 + 250 + 150 + 25 + 200 + 25 + 20 = 700ps

f- What is the minimum clock period for this CPU?

950ps