

1 Sequential vs. Pipelined

- 1.1 Assume you have a RISC-V processor that has the following execution times:

IF	ID	EX	MEM	WB
100 ps	150 ps	300 ps	400 ps	250 ps

- (a) In an unpipelined processor, what is the maximum clock rate possible?
- (b) In a pipelined processor, what is the maximum clock rate possible?
- (c) Suppose we add some hardware that shortens the ALU processing time to 250 ps. Does this change the maximum clock rate in an unpipelined processor? In a pipelined processor?
- 1.2 Fill in the timing diagram for the following code snippet assuming a pipelined processor (the first row has been filled in for you):

Time	1	2	3	4	5	6	7	8	9
sw									
addi									
add									
lw									

```
sw    s1, 0(s0)
addi  t0, t1, 8
add   s2, s0, s1
lw    t2, 0(t3)
```

How many cycles did the pipelined processor take? How many would an unpipelined processor take?

```

Loop: lw      t1, 0(t2)
      srl     t1, t1, 16
      sw      t1, 0(t2)
      addi    t2, t2, -4
      sub     t4, t3, t2
      bne     t4, 0, Loop
End:  sll     x0, x0, x0

```

Assume a standard 5 stage pipeline with no forwarding. Register file writes happen before reads, in the same clock cycle. Comparator logic begins at the end of the decode stage. We do not have a branch delay slot. Fill in the corresponding pipeline stages (F, D, E, M, W) at the appropriate times in the table below.

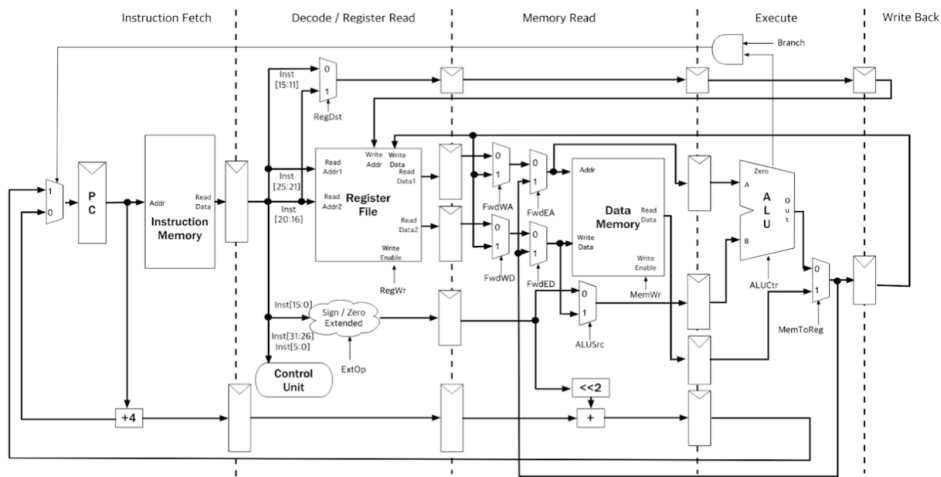
If the instruction requires a stall, write the stage again in the table. For example, if an instruction starts at cycle 2 but needs two stalls for the execute stage, then you would write “F D E E M W” for cycles 2 - 8.

[illegible]

- 2.2 How many cycles does this loop take to fully execute (from the first lw to the End label)?
- 2.3 Now assume the pipeline has 1 delayed branch slot and standard forwarding hardware. Also, reordering of instructions is allowed to minimize stalls. Write out the reordered sequence of instructions that achieves a minimal number of stalls needed.
- 2.4 How many cycles does this loop take to fully execute (from the lw to the End label)?

3 Pipelining Hazards

We construct a different five-stage pipelined CPU by swapping the execute and the memory read stages. Note that there is now only indirect addressing for the load word and store word.



- 3.1 Assume that this pipeline resolves control hazards by pipeline stalls. How many cycles is it stalled on a control hazard?
- 3.2 Should the pipeline stall for data hazards from load instructions? Give an example, fill in the corresponding pipeline stages, and explain your idea briefly in one or two sentences.

[illegible]