Week 4: Single cycle and pipelined CPUs

This is a preview of the published version of the quiz

Started: Feb 6 at 4:47pm

Quiz Instructions

This quiz covers single cycle CPU design and performance and pipelining.

See https://jlpteaching.github.io/comparch/modules/processor architecture/single-cycle/ (https://jlpteaching.github.io/comparch/modules/processor%20architecture/single-cycle/)

See https://jlpteaching.github.io/comparch/modules/processor architecture/pipelined/ (https://jlpteaching.github.io/comparch/modules/processor%20architecture/pipelined/)

Question 1 — 1 pts

The PC is used to access memory

- ◯ memory
- ◯ writeback
- ◯ fetch
- ◯ decode
- ◯ execute

Question 2 — 1 pts

Bits from the instruction are used to determine the size of the access to memory

- ◯ memory
- ◯ decode
- ◯ writeback

- ◯ fetch
- ◯ execute

Question 3 — 1 pts

The instruction is read from memory

- ◯ memory
- ◯ decode
- ◯ execute
- ◯ fetch
- ◯ writeback

Question 4 — 1 pts

The data is read from the register file

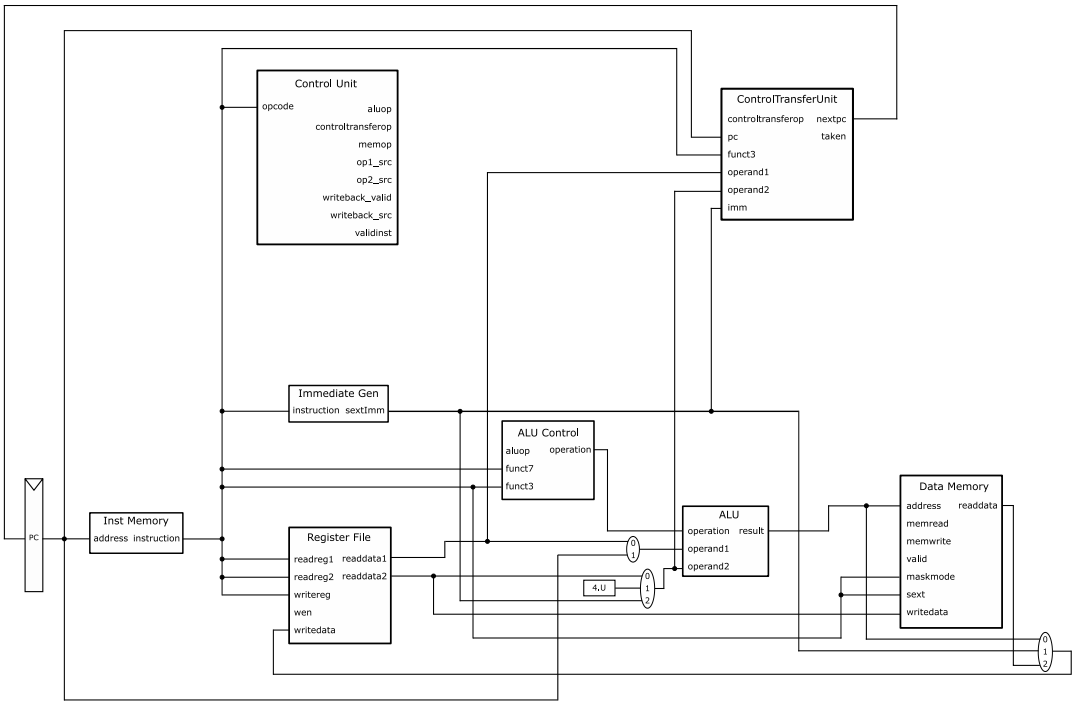
- ◯ writeback
- ◯ execute
- ◯ memory
- ◯ decode
- ◯ fetch

Question 5 — 1 pts

The ALU performs an operation as specified by the instruction

- ◯ decode
- ◯ memory
- ◯ execute
- ◯ writeback
- ◯ fetch

Use the following image for the next questions.



Single cycle DINO CPU

Mux numbers are from left to right.

Question 6 — 1 pts

For the following instruction, choose the correct value for each mux as shown in the figure above. If it doesn't matter, default to 0.

Mux numbers are from left to right.

```
jal x3, x8
```

Mux 1: [Select] Mux 2: [Select] Mux 3: [Select]

Question 7

1 pts

For the following instruction, choose the correct value for each mux as shown in the figure above. If it doesn't matter, default to 0.

Mux numbers are from left to right.

```
sll x5, x12, x2
```

Mux 1: [Select] Mux 2: [Select] Mux 3: [Select]

Question 8

1 pts

For the following instruction, choose the correct value for each mux as shown in the figure above. If it doesn't matter, default to 0.

Mux numbers are from left to right.

```
add x3, x7, x12
```

Mux 1: [Select] Mux 2: [Select] Mux 3: [Select]

Question 9

1 pts

[Select] is shared by all instructions.

[Select] selects subsets of components to use for each instruction

Question 10

1 pts

"A state element that contains a set of locations that can be read/written by supplying a location number and that is usually accessed multiple time for each instruction" describes which of the following data path elements?

- ◯ Muxes
- ◯ PC
- ◯ ALU
- ◯ data memory
- ◯ Immediate generator
- ◯ Register file
- ◯ control unit
- ◯ Instruction memory

Question 11

2 pts

There is a processor design that combines some steps. It has three stages: fetch & decode, execute, and memory & writeback. Use the following information to

answer the question. Note: This may be different from other questions.

Fetch & Decode	Execute	Memory & Writeback
150ps	210ps	230ps

If this is a single-cycle processor design, what is the cycle time of this processor in ps?

Question 12

2 pts

There is a processor design that combines some steps. It has three stages: fetch & decode, execute, and memory & writeback. Use the following information to answer the question. Note: This may be different from other questions.

Fetch & Decode	Execute	Memory & Writeback
150ps	150ps	280ps

What is the CPI for this processor?

CPI single-cycle = 1

Pipeline CPU:

Instruction latency: 5cycle

CPI ~ 1 (ideal) → > 1 other case

IPC ~ 1 (ideal)

Question 13

2 pts

There is a processor design that combines some steps. It has three stages: fetch & decode, execute, and memory & writeback. Use the following information to answer the question. Note: This may be different from other questions.

< 1 (other

Fetch & Decode	Execute	Memory & Writeback
180ps	240ps	300ps

How long does it take to execute an application with 6 billion instructions on this single cycle processor (in seconds)?

Question 14

1 pts

How long does it take to complete a single load of laundry (in minutes)?

Pre-wash	Washing	Drying	Folding/hanging
60 min	50 min	50 min	60 min

Question 15

1 pts

What is the limiting "stage" for this laundry system? (Can have multiple answers)?

Pre-wash	Washing	Drying	Folding/hanging
50 min	50 min	50 min	60 min

- [ ] Washing
- [ ] Pre-wash
- [ ] Drying
- [ ] Hanging/folding

Question 16

1 pts

What is the cycle time for the pipelined laundry (in minutes)? I.e., how frequently will a load be finished?

Pre-wash	Washing	Drying	Folding/hanging
40 min	40 min	60 min	60 min

Question 17 1 pts

What is the throughput for the pipelined laundry? I.e., how many loads can you complete **per hour**?

Pre-wash	Washing	Drying	Folding/hanging
40 min	50 min	50 min	50 min

Question 18 1 pts

What is the speedup pipelining compared to not pipelining? Assume you only care about the steady state (i.e., no need to consider warmup/cooldown time).

Pre-wash	Washing	Drying	Folding/hanging
50 min	50 min	50 min	60 min

Question 19 1 pts

How long does it take to execute a single instruction? (in ps)

Fetch	Decode	Execute	Memory	Writeback
400 ps	400 ps	200 ps	500 ps	300 ps

Question 20 — 1 pts

What is the limiting stage for this pipeline? (Can have multiple answers)?

Fetch	Decode	Execute	Memory	Writeback
500 ps	300 ps	300 ps	400 ps	200 ps

- [ ] Memory
- [ ] Decode
- [ ] Writeback
- [ ] Fetch
- [ ] Execute

Question 21 — 1 pts

What is the cycle time for the pipelined processor?

Fetch	Decode	Execute	Memory	Writeback
400 ps	400 ps	200 ps	600 ps	200 ps

Question 22 — 1 pts

What is the throughput for the pipeline? I.e., how many instructions can you complete **per second**?

Fetch	Decode	Execute	Memory	Writeback
400 ps	500 ps	200 ps	400 ps	300 ps

Question 23 — 1 pts

What is the speedup pipelining compared to not pipelining? Assume you only care about the steady state (i.e., no need to consider warmup/cooldown time).

Fetch	Decode	Execute	Memory	Writeback
500 ps	300 ps	300 ps	500 ps	300 ps

Question 24 — 2 pts

For the following program mark the data dependencies. Put a check when the register listed after the : depends on one of previous (older) instructions `xori s2, t2, 787`

```
sub s2, t3, t4
sra s6, s2, s1
ori s4, s6, 38
```

- [x] sra s6, s2, s1: s2
- [ ] sra s6, s2, s1: s1
- [ ] xori s2, t2, 787: s2
- [ ] sub s2, t3, t4: t3
- [ ] sub s2, t3, t4: s2
- [ ] ori s4, s6, 38: s4
- [ ] sra s6, s2, s1: s6
- [ ] xori s2, t2, 787: t2
- [x] ori s4, s6, 38: s6
- [ ] sub s2, t3, t4: t4

Question 25

2 pts

For the following program mark the data dependencies. Put a check when the register listed after the : depends on one of previous (older) instructions `ori s2, s9, -807`

```
xor s2, s1, s4
```

```
add s8, s2, a5
```

```
ori a7, s8, -893
```

- [ ] xor s2, s1, s4: s4
- [ ] add s8, s2, a5: a5
- [ ] add s8, s2, a5: s8
- [ ] xor s2, s1, s4: s2
- [ ] ori a7, s8, -893: a7
- [ ] ori s2, s9, -807: s2
- [ ] ori a7, s8, -893: s8
- [ ] add s8, s2, a5: s2
- [ ] ori s2, s9, -807: s9
- [ ] xor s2, s1, s4: s1

Question 26

1 pts

For the following program mark the data dependencies. Put a check when the register listed after the : depends on a previous (older) instruction.

```
sub a7, s0, a0
```

```
lw s8, (-944)a7
```

```
lw t2, (112)s8
```

- [ ] lw s8, (-944)a7: s8
- [ ] lw t2, (112)s8: t2
- [ ] sub a7, s0, a0: a7
- [ ] sub a7, s0, a0: a0
- [ ] lw t2, (112)s8: s8
- [ ] sub a7, s0, a0: s0
- [ ] lw s8, (-944)a7: a7

Question 27

1 pts

For the following program mark the data dependencies. Put a check when the register listed after the : depends on a previous (older) instruction.

```
sub t1, t3, a5
```

```
sw a7, (-1188)t1
```

```
sw t1, (1768)a7
```

- [ ] sub t1, t3, a5: t1
- [ ] sw t1, (1768)a7: a7
- [ ] sw a7, (-1188)t1: a7
- [ ] sw t1, (1768)a7: t1
- [ ] sub t1, t3, a5: a5
- [ ] sw a7, (-1188)t1: t1
- [ ] sub t1, t3, a5: t3

Question 28

2 pts

Which of the following instructions will cause later instructions to stall assuming there's no branch predictor?

`beq a7, a1, -1676`

`jal a6, 120`

`ori s0, a6, -1263`

`bne s5, s2, 1618`

`sub t5, t6, t5`

`sra a1, t3, t2`

`auipc a7, 470`

- [x] `bne s5, s2, 1618`
- [x] `jal a6, 120`
- [ ] `auipc a7, 470`
- [ ] `sub t5, t6, t5`
- [ ] `ori s0, a6, -1263`
- [ ] `sra a1, t3, t2`
- [x] `beq a7, a1, -1676`

Question 29

2 pts

Which of the following instructions will cause later instructions to stall assuming there's no branch predictor?

`ori s6, s2, 1808`

`blt s10, t0, -1708`

`ori a0, t5, -767`

`sw t3, -896(a0)`

`sra t4, a5, s4`

`bge t1, a0, 633`

`addi t3, a4, 1226`

- [ ] `blt s10, t0, -1708`

- [ ] `bge t1, a0, 633`
- [ ] `sw t3, -896(a0)`
- [ ] `ori s6, s2, 1808`
- [ ] `sra t4, a5, s4`
- [ ] `addi t3, a4, 1226`
- [ ] `ori a0, t5, -767`

Question 30

1 pts

In which stage do you *know* you need to stall for a control hazard?

- ◯ Execute
- ◯ Memory
- ◯ Decode
- ◯ Writeback
- ◯ Fetch

Question 31

1 pts

When predicting a branch, you need to predict which two things:

- [ ] which stage it is in
- [ ] if it is an exception
- [ ] number of cycles to stall
- [ ] the target address
- [ ] forward or backward
- [ ] taken or not taken