HW6.3. Datapath Design Part 2: Pipelined

Feel free to check out the guide that we have prepared to help you in this problem.

This question is supposed to be answered after Datapath Design Part 1: Single Cycle

We decide to improve the performance of our CPU by splitting it into 3 stages: Instruction Fetch (IF), Execution (X), and Memory Access (M). Note that this is different from the 5-stage pipeline shown in lecture. The pipelined datapath is shown below. **The blue lines indicate where we decide to make the split, meaning any necessary pipeline registers will be added to the datapath at these locations.**

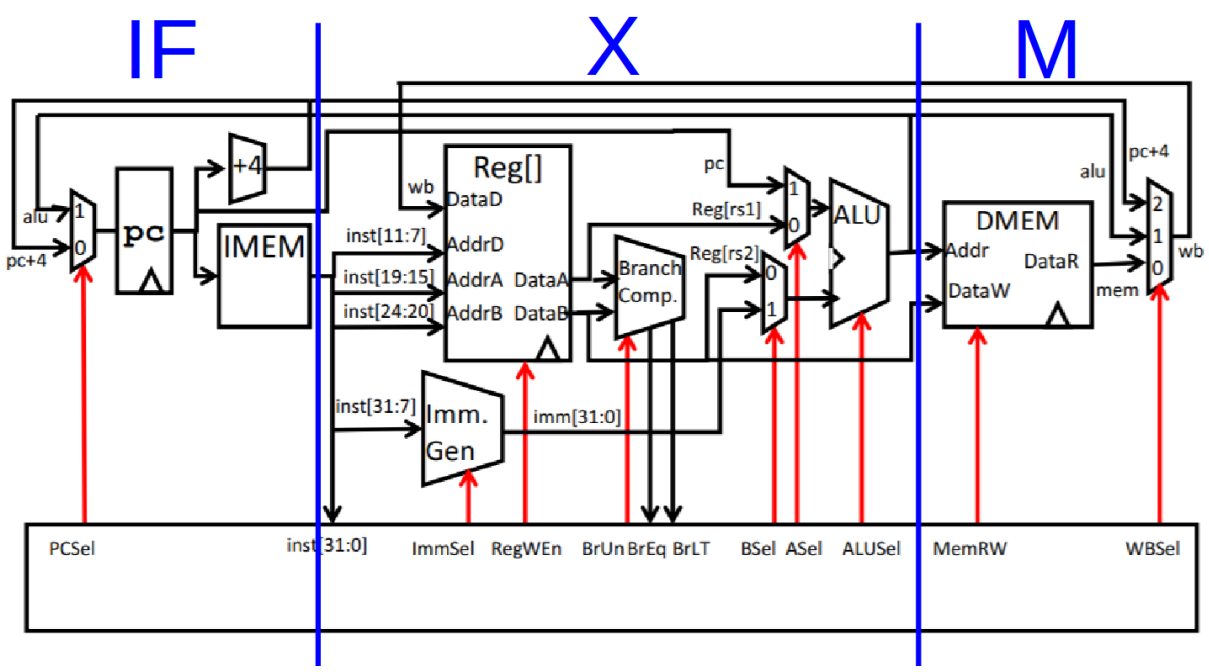
A more detailed description of each stage (Read the descriptions carefully):

IF: the PC is input into the instruction memory and the output is saved in a pipeline register.

X: the instruction is decoded, we read data from the regfile, perform some computation with the ALU, and save any necessary data in pipeline registers. **Note that writing back the ALU output to the PC mux is considered part of the X stage, but there are no pipeline registers separating the ALU output and the PC mux.**

M: data is read/written to memory, and data is potentially written back into the registers. **Note that writing back data from the WB mux output to the Reg File is considered part of the M stage, but there are no pipeline registers separating the WB mux output and the Reg File.**

For this problem, you may assume any control signal is properly handled for all pipeline stages. You can also assume that hazards will not occur.



Assume the following delays for components. Any component not listed is assumed to have a negligible delay. *These are the same numbers from Part 1: Single Cycle.*

Element	Reg clk-to-q	Reg setup	MUX	ALU	MemRead	MemWrite	RegFile Read	RegFile Setup	BranchComp
Parameter	$t_{clk-to-q}$	t_{setup}	t_{mux}	t_{alu}	$t_{mem-read}$	$t_{mem-write}$	$t_{rf-read}$	$t_{rf-setup}$	$t_{branch-comp}$
Delay (ps)	30	20	50	200	250	250	150	20	75

1.1: What instruction type takes the longest to execute given the pipeline stages (i.e. the instruction type that dictates the critical path of the pipelined datapath)?
Hint: Try to find the path with the longest combinational delay in the datapath (you can try to answer 1.2 first), then check which of the instructions listed below utilize that path. The Reference Card might be able to help when identifying the specific instruction traversing a specific path.

- (a) Branches: beq, bne, ...
- (b) jal
- (c) I-type ALU Instructions: addi, xori, ...
- (d) jalr ✅
- (e) lui
- (f) auipc
- (g) Store Instructions: sw, sbu, ...

Homework 6

Assessment overview

Total points: 50/100

Score: 50%

Question

Value: 20

History: 20

Awarded points: 20/20

Report an error in this question

Previous question

Next question

Attached files

No attached files

Attach a file

Attach text

- [ ] (h) R-type ALU Instructions: add, xor, ...
- [ ] (i) Load Instructions: lb, lw, ...

✓ 100%

1.2: For the instruction described in question , which components would be relevant to the critical path calculation?

- [x] (a) Pipeline registers between IF and X ✓
- [ ] (b) Pipeline registers between X and M
- [x] (c) PC-MUX ✓
- [x] (d) PC register ✓
- [ ] (e) IMEM
- [ ] (f) Imm-Gen
- [x] (g) RegFile read ✓
- [ ] (h) Branch Comparator
- [x] (i) A-MUX ✓
- [ ] (j) B-MUX
- [x] (k) ALU ✓
- [ ] (l) DMEM
- [ ] (m) WB-MUX
- [ ] (n) RegFile Write

Select all possible options that apply.

✓ 100%

1.3: What is the critical path delay (or the minimum operating clock period) of this circuit, in ps?

500 ✓ 100%

1.4: What is the fastest clock frequency that can be used with this datapath? Express your answer in GHz.
Take note of the time units

2.00 ✓ 100%

Recall that latency refers to how long any given instruction takes to finish, while throughput is defined as the number of operations finished per unit time.

1.5: What is the minimum latency of a single instruction, in cycles?
Hint: How many clock cycles does it take to finish 1 instruction?

3 ✓ 100%

1.6: What is the minimum latency of a single instruction, in picoseconds?
Hint: The clock period you calculated earlier is the number of seconds per clock cycle.

1500 ✓ 100%

1.7: What is the maximum throughput, in instructions/cycle?
Hint: Assuming the processor has been executing instructions for a while, how many instructions finish every clock cycle?

1.00 ✓ 100%

1.8: What is the maximum throughput, in instructions/second?
Hint: Take note of the time units, count the zeroes properly.

2.00e9 ✓ 100%

1.9: Based on the above and your answers in Part 1: Single Cycle, what can reasonably be concluded about the advantages of pipelining?

- [x] (a) We can increase the clock frequency ✓

(b) We can finish executing more instructions per cycle

(c) A single instruction passes through the datapath faster

(d) We can handle more than one instruction at a time ✅

Select all possible options that apply.

✓ 100%

Try a new variant

Correct answer

1.1: What instruction type takes the longest to execute given the pipeline stages (i.e. the instruction type that dictates the critical path of the pipelined datapath)?
Hint: Try to find the path with the longest combinational delay in the datapath (you can try to answer 1.2 first), then check which of the instructions listed below utilize that path. The Reference Card might be able to help when identifying the specific instruction traversing a specific path.

(d) jalr

1.2: For the instruction described in question , which components would be relevant to the critical path calculation?
(a) Pipeline registers between IF and X
(c) PC-MUX
(d) PC register
(g) RegFile read
(i) A-MUX
(k) ALU

1.3: What is the critical path delay (or the minimum operating clock period) of this circuit, in ps?
`500`

1.1-1.3: The critical path goes through the X stage, and ends at the PC register. This path is taken only by jalr instructions since it performs `PC = rs1 + imm` which requires a register read, an addition through the ALU, and a write to the PC register.

We start from the pipeline register between IF and X: clk-to-q (30 ps).

Then we read the register file: Reg Read (150 ps)

Then we go through the A-Mux (50 ps). Note that B-mux path is not relevant for the critical path since that is selecting the data from the immediate and not the register file. That is, the data going through B-mux will arrive faster (and therefore will not contribute to the critical path) than the data going through the A-mux.

Then we go through the ALU (200 ps)

Then we go through the PC mux (50 ps). Note that the path from the ALU to the PC mux does not go through a register. The pipeline stages only partition the data flow going from left (fetching instructions from the memory) to the right (ALU execution and memory accesses). Writeback paths (data from later stages, data memory or ALU, going back to the PC mux and Reg File writeback) are not pipelined.

And finally ending at the PC Register: setup time (20 ps)

The total is 500 ps. Note that branch instructions performs the comparison of rs1 and rs2 through the Branch Comp block, while PC+offset is being calculated through the ALU. These two paths are executed in parallel and does not add up. jalr instructions, on the other hand, needs to calculate rs1+imm which requires reading from the register file and then using the ALU, which adds up the delays.

1.4: What is the fastest clock frequency that can be used with this datapath? Express your answer in GHz.
Take note of the time units

`2.00`

1.4: Frequency = 1/period = 1/(500 ps) = 2 GHz

Recall that latency refers to how long any given instruction takes to finish, while throughput is defined as the number of operations finished per unit time.

1.5: What is the minimum latency of a single instruction, in cycles?
Hint: How many clock cycles does it take to finish 1 instruction?

3

1.6: What is the minimum latency of a single instruction, in picoseconds?
Hint: The clock period you calculated earlier is the number of seconds per clock cycle.

1500

1.7: What is the maximum throughput, in instructions/cycle?
Hint: Assuming the processor has been executing instructions for a while, how many instructions finish every clock cycle?

1.00

1.8: What is the maximum throughput, in instructions/second?
Hint: Take note of the time units, count the zeroes properly.

2.00e9

1.5 through Q1.8: Each instruction takes three cycles to go through the entire datapath, and one instruction finishes per cycle. Assuming we use our fastest clock cycle of 2 GHz, we get a latency of 1.5 ns and 2 billion ops/second, respectively. Note that it generally takes longer for a single instruction to get through a pipelined circuit, because the values get "stuck" at the end of the I and M stages while waiting for the X stage to finish.

1.9: Based on the above and your answers in Part 1: Single Cycle, what can reasonably be concluded about the advantages of pipelining?
(a) We can increase the clock frequency
(d) We can handle more than one instruction at a time

1.9: We generally can't finish more than one instruction per cycle, since we still can't have multiple instructions go through the same stage at the same time.

In general, instructions must pass through the length of the pipeline (aside: strictly speaking it is possible to commit (i.e, finish) specific instructions at earlier stages) the additional setup time constraints and clock-to-Q delays introduced by pipeline registers necessarily increase the total time it takes for a single instruction to pass through the pipeline. Additionally, the logic required to detect and resolve pipeline hazards only exacerbates this effect.

In a pipelined datapath, we can have roughly as many instructions in-flight as we have pipeline stages! So while any one instruction takes longer to execute, the latencies of consecutive instructions are overlapped. This is why the CPI of pipeline can still approach 1 even though it might take 5+ cycles to compute any single instruction in isolation.

Increasing frequency is one of the primary results of pipelining as the amount of logical work done per clock-cycle is being reduced. Since each stage of the processor does less work, we can meet timing constraints with a shorter period.

It's critical to note at this point that we pipeline processors to get both (3) and (4) at once -- not just one or the other. Taken together, the rate at which the processor can execute instructions (or throughput, in units: (instructions/second) improves. See the Iron Law of Processor Performance for more detail!

Submitted answer **correct: 100%**
Submitted at 2022-10-23 08:26:21 (PDT)

hide

1.1: What instruction type takes the longest to execute given the pipeline stages (i.e. the instruction type that dictates the critical path of the pipelined datapath)?
Hint: Try to find the path with the longest combinational delay in the datapath (you can try to answer 1.2 first), then check which of the instructions listed below utilize that path. The Reference Card might be able to help when identifying the specific instruction traversing a specific path.

(d) jalr ✓ 100%

1.2: For the instruction described in question , which components would be relevant to the critical path calculation?
(a) Pipeline registers between IF and X ✓
(c) PC-MUX ✓
(d) PC register ✓
(g) RegFile read ✓
(i) A-MUX ✓
(k) ALU ✓
✓ 100%

1.3: What is the critical path delay (or the minimum operating clock period) of this circuit, in ps?
`500` ✓ 100%

1.4: What is the fastest clock frequency that can be used with this datapath? Express your answer in GHz.
Take note of the time units

2 ? ✓ 100%

Recall that latency refers to how long any given instruction takes to finish, while throughput is defined as the number of operations finished per unit time.

1.5: What is the minimum latency of a single instruction, in cycles?
Hint: How many clock cycles does it take to finish 1 instruction?

3 ✓ 100%

1.6: What is the minimum latency of a single instruction, in picoseconds?
Hint: The clock period you calculated earlier is the number of seconds per clock cycle.

`1500` ✓ 100%

1.7: What is the maximum throughput, in instructions/cycle?
Hint: Assuming the processor has been executing instructions for a while, how many instructions finish every clock cycle?

1 ? ✓ 100%

1.8: What is the maximum throughput, in instructions/second?
Hint: Take note of the time units, count the zeroes properly.

`2000000000` ? ✓ 100%

1.9: Based on the above and your answers in Part 1: Single Cycle, what can reasonably be concluded about the advantages of pipelining?
(a) We can increase the clock frequency ✓
(d) We can handle more than one instruction at a time ✓
✓ 100%