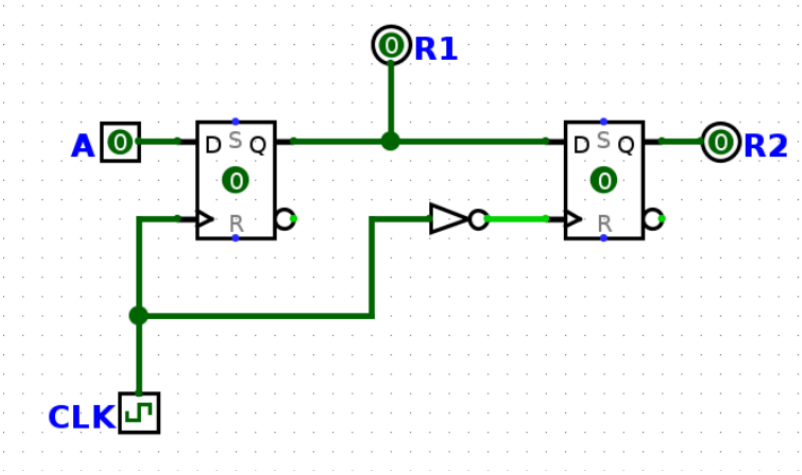
CS61C Spring 2025

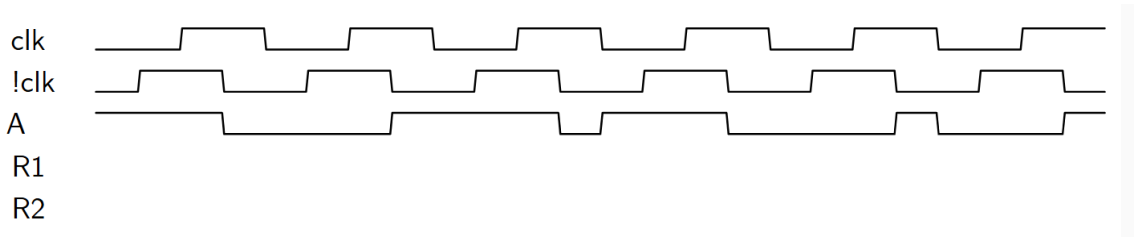
SDS, RISC-V Datapath

Discussion 8

1 SDS Intro

1.1 Fill out the timing diagram. The clock period (rising edge to rising edge) is 8ps. For every register, clk-to-q delay is 2ps, setup time is 4ps, and hold time is 2ps. NOT gates have a 2ps propagation delay, which is already accounted for in the !clk signal given.

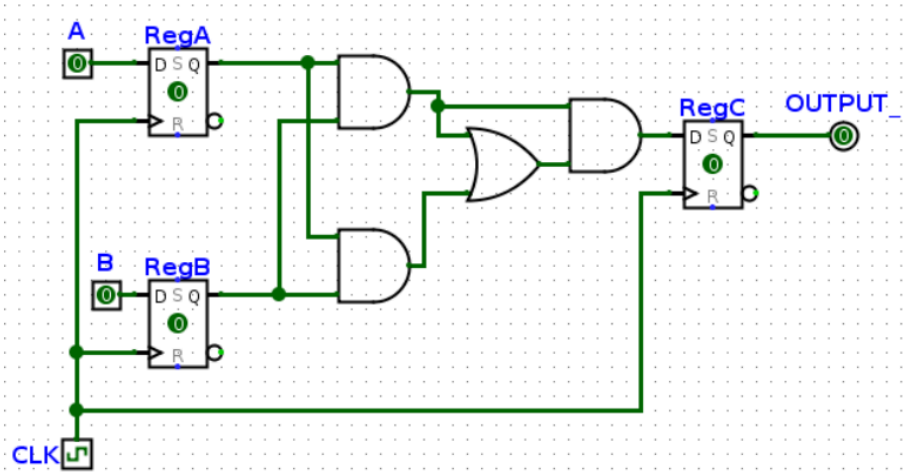




1.2 In the circuit below:

- RegA and RegB have setup, hold, and clk-to-q times of 4ns,
- All logic gates have a delay of 5ns
- RegC has a setup time of 6ns.

What is the maximum allowable hold time for RegC? What is the minimum acceptable clock cycle time for this circuit, and clock frequency does it correspond to?



2 Single-Cycle CPU

For this worksheet, we will be working with the single-cycle CPU datapath provided on the last page.

2.1 List all possible values that each control signal may take on for the single cycle datapath, then briefly describe what each value means for each signal.

(a) PCSel

(b) RegWEn

(c) ImmSel

(d) BrEq

(e) BrLt

(f) ALUSel

(g) MemRW

(h) WBSel

2.2 Fill out the following table with the control signals for each instruction based on the datapath on the last page.

- If the value of the signal does not affect the execution of an instruction, use the * (don't care) symbol to indicate this.
- If the value of the signal does affect the execution, but can be different depending on the program, list all possible values (for example, for a signal that may output 0 and 1, write 0/1).
- For ALUSel, write the ALU operation (`add`, `or`, `sll`, ...)

The first row has been filled out for you.

	BrEq	BrLT	PCSel	Imm-Sel	BrUn	ASel	BSel	ALUSel	MemRW	Reg-WEn	WB-Sel
add	*	*	0 (PC + 4)	*	*	0 (Reg)	0 (Reg)	add	0	1	1 (ALU)
ori											
lw											
sw											
beq											
jal											
blt											

3 Timing the Datapath

Clocking review:

- A **state element** is an element connected to the clock (denoted by a triangle at the bottom). The **input signal** to each state element must stabilize before each **rising edge**.
- The **critical path** is the longest delay path between state elements in the circuit. The circuit cannot be clocked faster than this, since anything faster would mean that the correct value is not guaranteed to reach the state element in the alloted time. If we place registers in the critical path, we can shorten the period by **reducing the amount of logic between registers**.

For this exercise, the times for each circuit element is given as follows:

Clk-to-Q	RegFile Read	PC/RegFile Setup	Mux	Adder
5ns	35ns	20ns	15ns	20ns

ALU	Branch Comp	Imm Gen	MEM Read	DMEM Setup
100ns	50ns	45ns	300ns	200ns

3.1 Mark an X for the datapath stages used by each instruction

	IF	ID	EX	MEM	WB
`add`					
`ori`					
`lw`					
`sw`					
`beq`					
`jal`					

3.2 How long does it take to execute each instruction? Ignore the length of a clock cycle based off of the critical path, and assume that the setup times to the RegFile and the PC are the same.

(a) jal

(b) lw

(c) sw

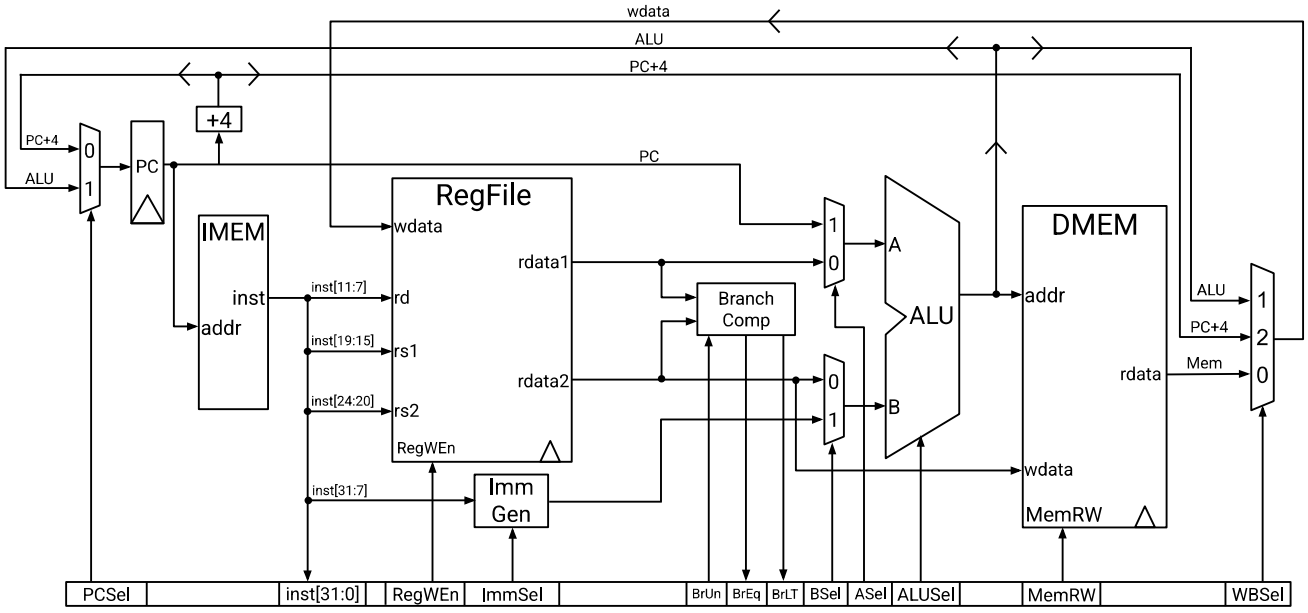
3.3 Which instruction(s) are responsible for the critical path?

3.4 What is the highest clock frequency for this single cycle datapath?

3.5 Why is the single-cycle datapath inefficient?

3.6 How can you improve its performance? What is the purpose of pipelining?

Single-Cycle Datapath Diagram



5-Stage Datapath Diagram