Single Cycle CPU Design

1. Here we have a single cycle CPU diagram. Answer the following questions:

a. Name each component. (3 points)

b. Name each datapath stage and explain its functionality. (5 points)

Stage	Functionality
Instruction fetch	Send an address to the instruction memory. Read the instruction(IMEM[PC])
Decode/Register Read	Decode the instruction, use the register address to access the register group, and read the op number.
Execute	ALU Execute different operations and branch comparison.
Memory	This stage process load, store, and br inst. Read or write from memory. send the br address into PC
Register Write	Write the result into register[]

c. Provide data inputs and control signals to the next PC logic. (3 points)

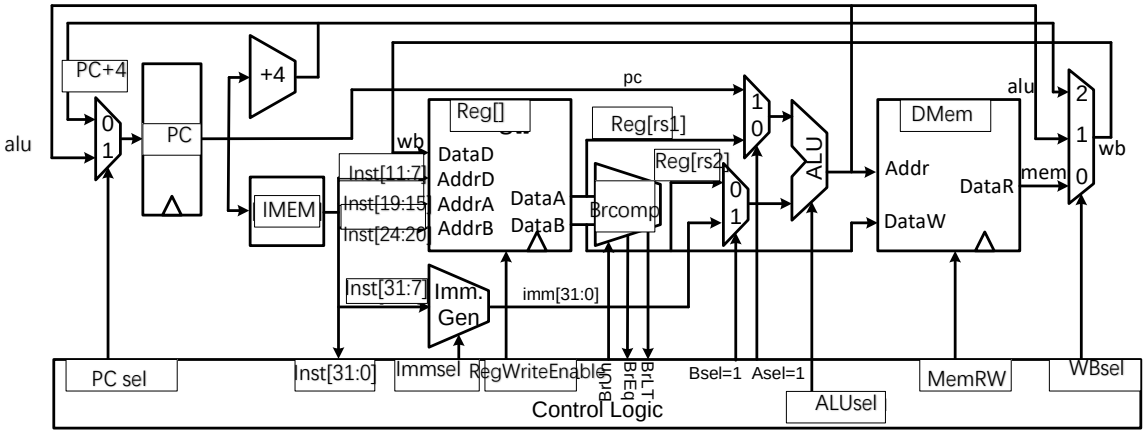


Fig. 1 Single Cycle RISC-V Datapath

Single Cycle CPU Control Logic

4. Fill out the values for the control signals from the previous CPU diagram. (8 points)

Instrs.	Control Signals									
	BrEq	PCSel	ImmSel	BrUn	ASel	BSel	ALUSel	MemRW	RegWEn	WBSel
addi	X	Pc+4	1	X	Reg	imm	Add	Read	1	ALU
and	X	Pc+4	0	X	Reg	reg	And	Read	1	ALU
lw	X	Pc+4	I	X	Reg	Imm	Add	read	1	MEM
sw	X	Pc+4	I	X	Reg	Imm	Add	Write	0	X
beq	1	Alu	0	X	Pc	Imm	Add（怎	X	0	X

							么比较的？跳转是add)			
jal	X	Alu	0	X	Pc	Imm	add	Read	0	PC+4
jalr	X	Alu	0	X	Reg	imm	add	Read	0	PC+4
Lui(load upper immediate.)	X	Pc+4	1	1	Reg	Imm	Pass	Read	1	ALU

(Put an X if the signal doesn't matter)

Clocking Methodology

- The input signal to each state element must stabilize before each rising edge.
- Critical path: Longest delay path between state elements in the circuit.
- If we place registers in the critical path, we can shorten the period by reducing the amount of logic between registers.

Single Cycle CPU Performance Analysis

5. The delays of circuit elements are given as follows:

Stage	IF	ID	EX	MEM	WB
Delay	200	100	200	300	100

a. Mark the stages the following instructions use and calculate the time to execute. (8 points)

Instruction	IF	ID	EX	MEM	WB	Total
Delay	200	100	200	300	100	-
addi	X	X	X		X	600
and	X	X	X		X	600
lw	X	X	X	X	X	900
sw	X	X	X	X		800
beq	X	X	X			500
jal	X	X	X		X	600
jalr	X	X	X		X	600
lui	X	X	X			500

b. Which instruction(s) exercises the critical path? (3 points)
load word , it needs to read from the memory and write to the register.

c. What is the fastest you could clock this single-cycle datapath? (3 points)
1/(900ps) = 1/(900*10^-12s) = 1.11*10^9 Hz = 1.11GHz

d. Why is a single cycle CPU inefficient? (3 points)
The instructions don't parallelize. Every instruction has the same clock cycle time, so we have to clock to the slowest instruction. This instruction usually lw because it uses many components : instruction store, read register [], ALU, dataMEM, write back

register[].

e. How can you improve its performance? (3 points)
use pipeline CPU, put registers between two stages so that we can make full use of each component in CPU.

6. A single cycle RISCV datapath is illustrated in Fig. 1. Now we have the time information for each component tabulated in the Table below.

Element	Register clk-to-q	Register Setup	MUX	ALU	Mem Read	Mem Write	RegFile Read	RegFile Setup
Parameter	$t_{clk\text{-}to\text{-}q}$	t_{setup}	t_{mux}	t_{ALU}	$t_{MEMread}$	$t_{MEMwrite}$	t_{RFread}	$t_{RFsetup}$
Delay(ps)	30	20	25	250	400	500	200	50

a. What's the clock time and frequency of a single cycle CPU (ignore the branch comp and imm.)? (10 points)
25+30+20+250+400+500+200+50=1475ps
1/1475ps = 1*10^12/1475=677966100 = 677MHz
b. If the branch comp, imm., and the control need 50 ps, 100 ps, and 250 ps, respectively. Which instructions below needs longest processing time? Please explain. (12 points)

`addi, jalr, lui, lw, sw`

lw

because it needs mem read and mem write， it costs much time.

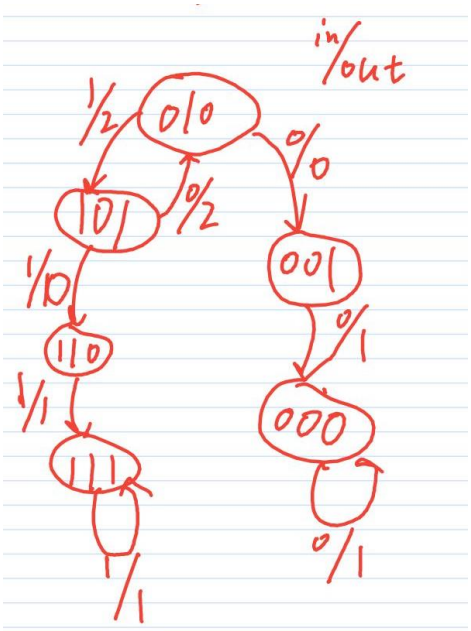
Digital Circuit Design

10. Create an XOR gate using only NAND gates. (6 points)

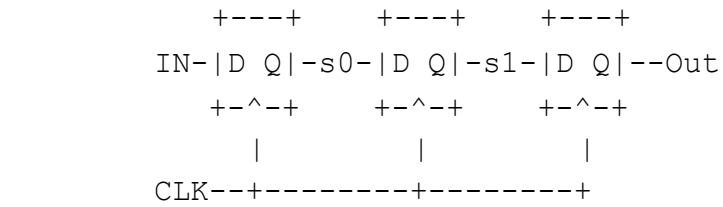
或非门 xor，不同出 1 相同出 0， y = (a'b'+ab)' = (a'b')'(ab)' = (a+b)(a'+b') = a'b +b'a =((a'b + b'a)')'加两个非 =((a')' (b'a)')'

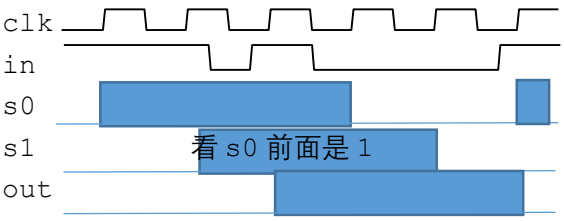
11. How many different two-input logic gates can there be? How many n-input logic gates? (6 points)
Each input can be either 0 or 1, so we have 2^n input, output can be either 0 or 1, so we have 2^(n+1) n-input logic gates.

12. Draw an FSM for outputting a 1 whenever we have three repeating bits as the most recent bits, a 2 whenever we have 010 or 101, and a 0 otherwise. You may not need all states. (12 points)



13. Fill out the timing diagram for the circuit below: (9 points)





14. Fill out the timing diagram for the circuit below: (6 points)