

Information Technology Institute

Digital IC Design

Presented by:

Mohamed Adel Melad Abd_Elhameed Youssef

Subject: Lab_2 verilog

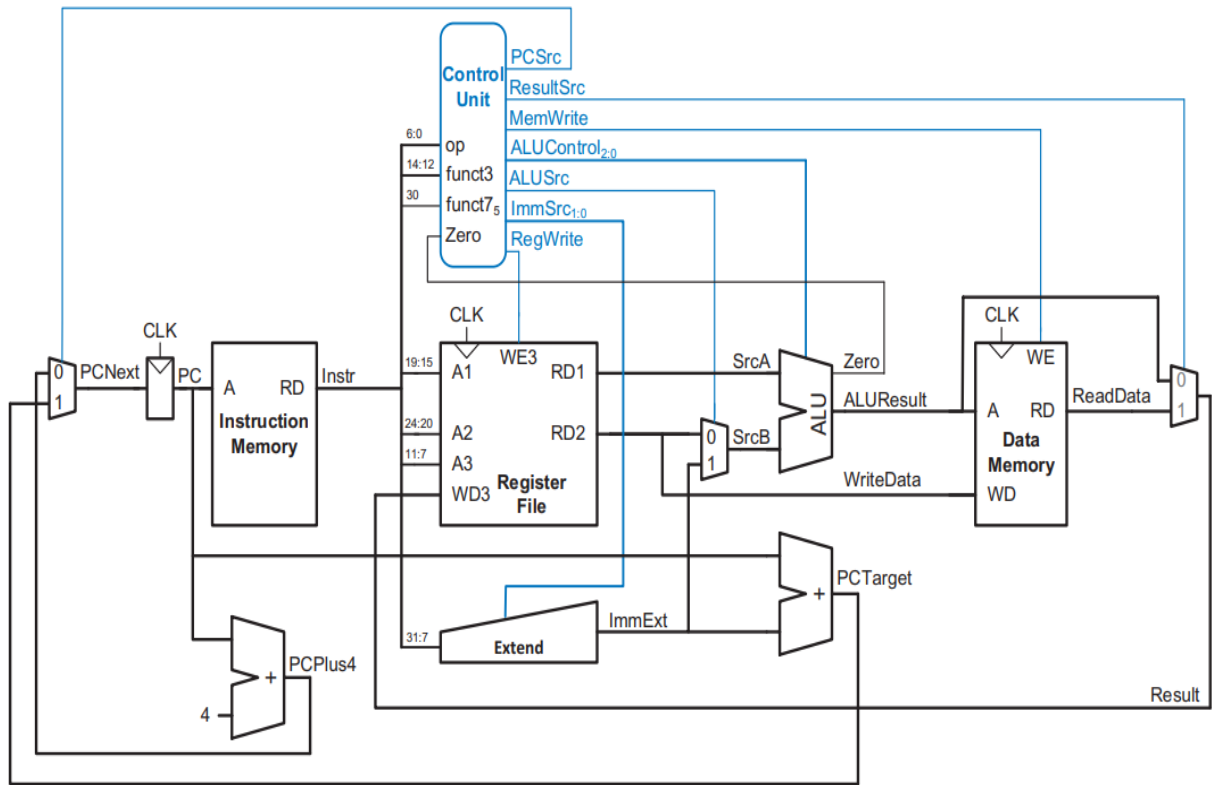
"RISC_V single cycle Processor"

Under supervising of:

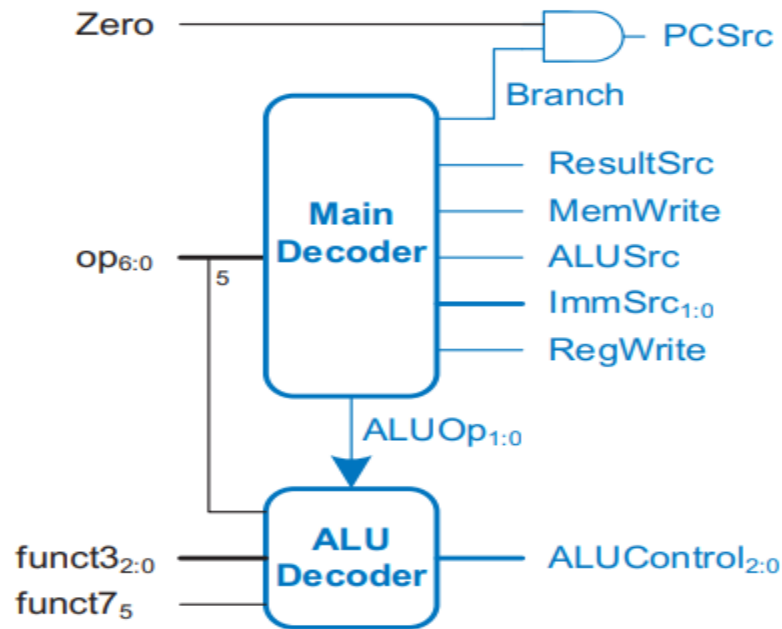
Eng. Rawan Naguib

●RISC_V Single Cycle Processor Description :

» The single-cycle datapath, adding one piece at a time to the state elements from RISC_V single cycle system Figure . The new connections are emphasized in black (or blue, for new control signals), whereas the hardware that has already been studied is shown in gray. The example instruction being executed is shown at the bottom of each figure. The program counter contains the address of the instruction to execute. The first step is to read this instruction from instruction memory. RISC_V single cycle system Figure shows that the PC is simply connected to the address input of the instruction memory. The instruction memory reads out, or fetches, the 32-bit instruction, labeled Instr. In our sample program from RISC_V single cycle system Figure, PC is 0x1000. (Note that this is a 32-bit processor, so PC is really 0x00001000, but we omit leading zeros to avoid cluttering the figure.)



=====RISC_V single cycle system Figure=====



=====RISC_V single cycle control unit=====

●what I need in testbench :

```
# riscvtest.s
# Sarah.Harris@unlv.edu
# David Harris@hmc.edu
# 27 Oct 2020
#
# Test the RISC-V processor:
#  add, sub, and, or, slt, addi, lw, sw, beq, jal
# If successful, it should write the value 25 to address 100
#
```

	RISC-V Assembly	Description	Address	Machine Code
main:	addi x2, x0, 5	# x2 = 5	0	00500113
	addi x3, x0, 12	# x3 = 12	4	00C00193
	addi x7, x3, -9	# x7 = (12 - 9) = 3	8	FF718393
	or x4, x7, x2	# x4 = (3 OR 5) = 7	C	0023E233
	and x5, x3, x4	# x5 = (12 AND 7) = 4	10	0041F2B3
	add x5, x5, x4	# x5 = 4 + 7 = 11	14	004282B3
	beq x5, x7, end	# shouldn't be taken	18	02728863
	slt x4, x3, x4	# x4 = (12 < 7) = 0	1C	0041A233
	beq x4, x0, around	# should be taken	20	00020463
	addi x5, x0, 0	# shouldn't execute	24	00000293
around:	slt x4, x7, x2	# x4 = (3 < 5) = 1	28	0023A233
	add x7, x4, x5	# x7 = (1 + 11) = 12	2C	005203B3
	sub x7, x7, x2	# x7 = (12 - 5) = 7	30	402383B3
	sw x7, 84(x3)	# [96] = 7	34	0471AA23
	lw x2, 96(x0)	# x2 = [96] = 7	38	06002103
	add x9, x2, x5	# x9 = (7 + 11) = 18	3C	005104B3
	jal x3, end	# jump to end, x3 = 0x44	40	008001EF
	addi x2, x0, 1	# shouldn't execute	44	00100113
end:	add x2, x2, x9	# x2 = (7 + 18) = 25	48	00910133
	sw x2, 0x20(x3)	# [100] = 25	4C	0221A023
done:	beq x2, x2, done	# infinite loop	50	00210063

●Simulation Result :

