
I would like to use one late token for lab 4.

1: Compilation Order and Design Decisions

a Compilation Order

```
0 and2.vhd
1 xorgate.vhd
2 half.adder.vhd
3 full.adder.vhd
4 adder64.vhd
5 add.vhd
6 mux5.vhd
7 mux64.vhd
8 alu.vhd
9 alucontrol.vhd
10 and2.vhd
11 cpucontrol.vhd
12 shiftleft2.vhd
13 signextend.vhd
14 pc.vhd
15 reg_IFID.vhd
16 reg_IDEX.vhd
17 reg_EXMEM.vhd
18 reg_MEMWB.vhd
19 registers_p1.vhd
20 dmem.vhd
21 imem_p1.vhd
22 pipelinedcpu0.vhd
23 pipelinedcpu0_tb.vhd
```

b Design Decisions

CBNZ implementation

I create a new cpu control signal, NOTZERO, for CBNZ implementation. When the NOTZERO is '0', it will pass the value of signal 'Zero' to the and gate. And when the NOTZERO is '1', it will pass the value of '!Zero' to the and gate. I use (NOTZERO XOR ALU_Zero) to implement this logic.

Pipeline Registers

For all pipeline registers(IF/ID, ID/EX, EX/MEM, MEM/WB), I initialize their output signals to all zero. The input signals are assigned to output signals at the rising edge of 'clk'. The output signals are set to zero when 'rst' is '1'.

2: Cycle One

a Debug signals

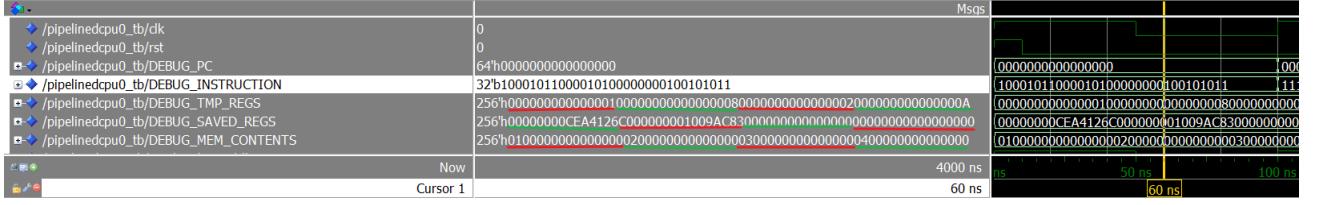


Figure 1: Debug Signals of Cycle one

In cycle one, we fetch instruction at PC address of x00, the instruction is 1000 1011 0000 1010 0000 0001 0010 1011.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x02, \$X12 = x0A.

Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = x0, \$X22 = x0. DMEM are set to DMEM(0x0) = x1, DMEM(0x8) = x2, DMEM(0x16) = x3, DMEM(0x24) = x4.

b IF/ID Register

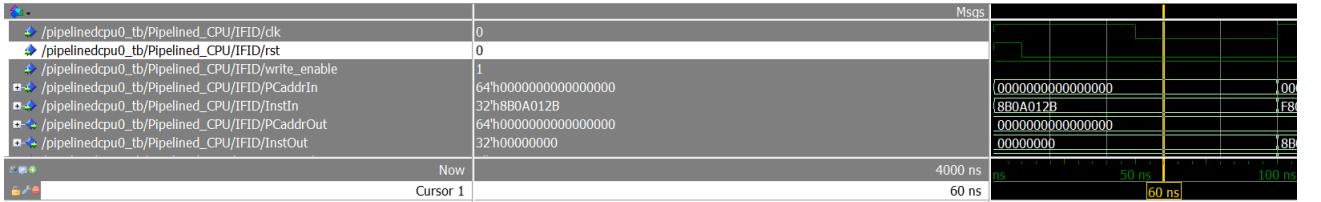


Figure 2: IF/ID Signals of Cycle one

The InstOut and PCaddrOut are both initialized to 0. The value of PCaddrIn and InstIn would be assigned to them correspondingly at the next rising edge.

Since IF/ID register do not have output signals, all the output signals of other pipeline registers are initialized to 0.

3: Cycle Two

a Debug signals

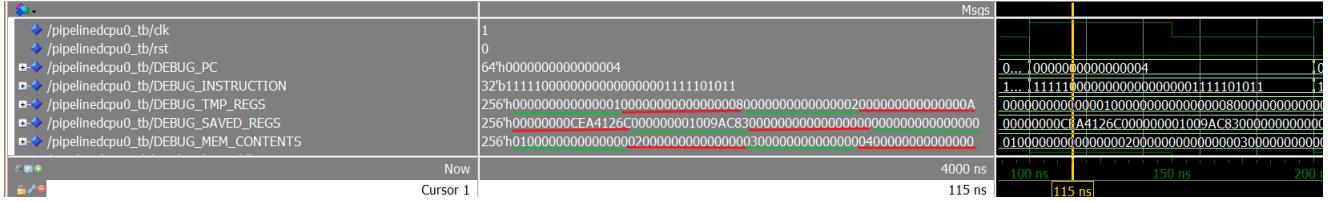


Figure 3: Debug Signals of Cycle two

In cycle two, we fetch instructions at PC address of x04, the instruction is 1111 1000 0000 0000 0000 0011 1110 1011.

Registers and Data memories stay the same as cycle one.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x02, \$X12 = x0A.

Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = x0, \$X22 = x0. DMEM are set to DMEM(0x0) = x1, DMEM(0x8) = x2, DMEM(0x16) = x3, DMEM(0x24) = x4.

b IF/ID Register

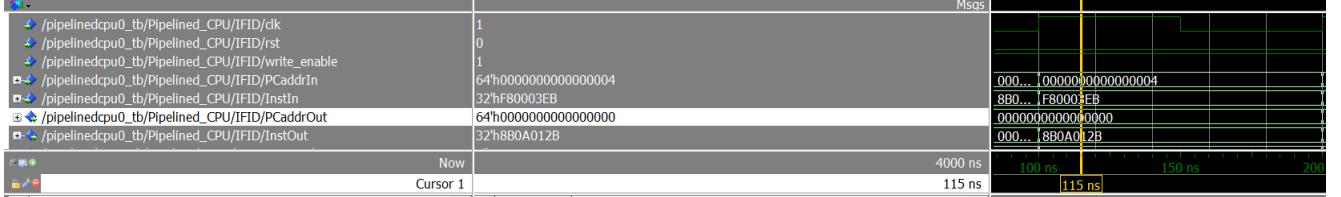


Figure 4: IF/ID Signals of Cycle two



Figure 5: PCSrc Signal of Cycle two

At 100 ns, a rising edge of 'clk' signal, the PC address is changed from x00 to x04. The output signals of IF/ID register are also assigned with the input signals of cycle one.

PCaddrOut = x0(value of the previous input) \leftarrow x0(initialization), InstOut = x8B0A012B \leftarrow x0.

c ID/EX Register

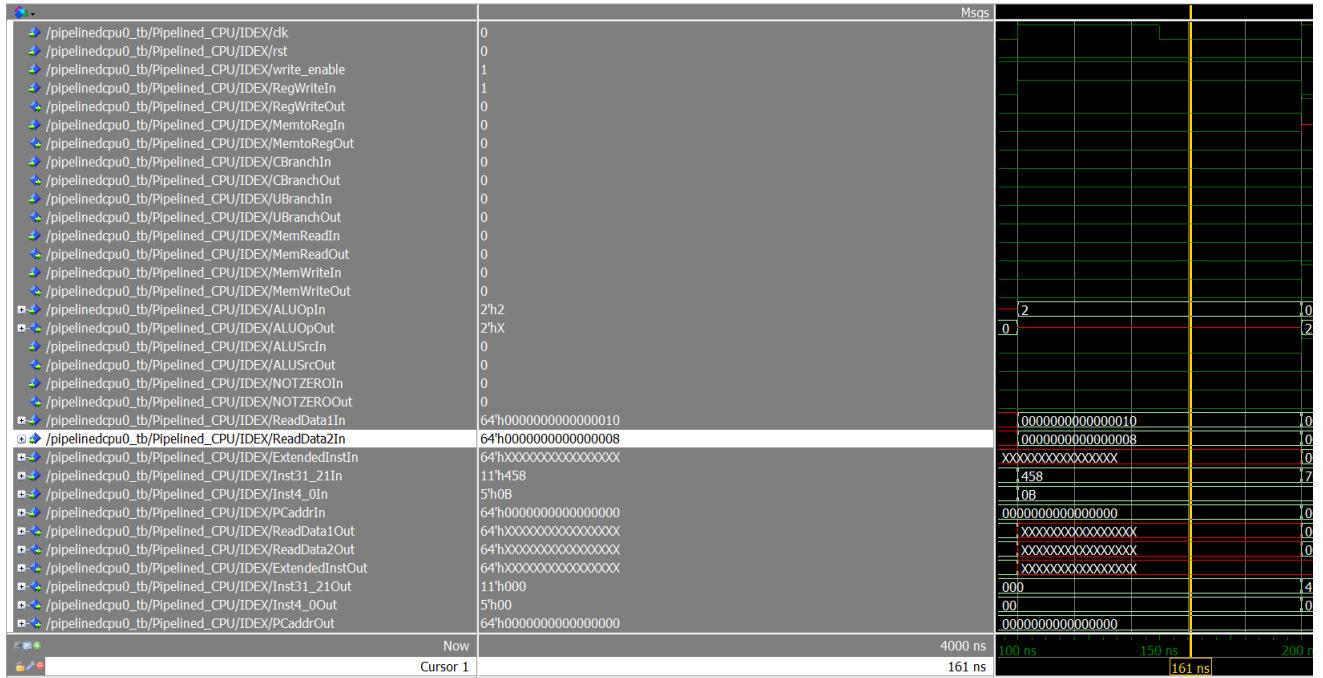


Figure 6: ID/EX Signals of Cycle two

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction one(ADD X11, X9, X10) decoding. ReadData1In = \$X9 = x10, ReadData2In = \$X10 = x08, ExtendedInstIn is unknown since this is a R-type instruction, PCaddrIn = x0.

All the output signals are not decided, and would be assigned the same value as the corresponding input signals at the rising edge of next cycle.

4: Cycle Three

a Debug signals

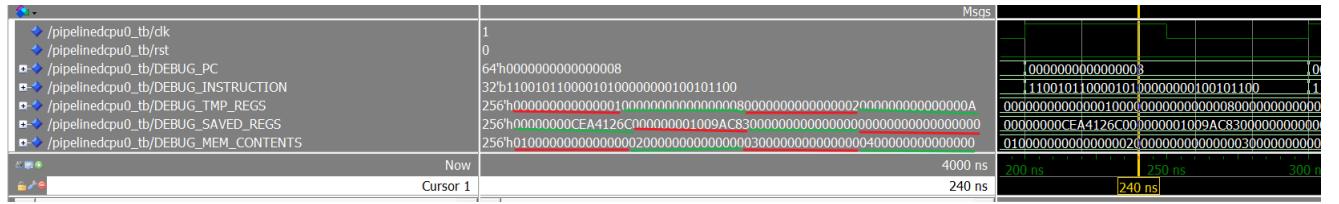


Figure 7: Debug Signals of Cycle three

In cycle three, we fetch instruction at PC address of x08, the instruction is 1100 1011 0000 1010 0000 0001 0010 1100.

Registers and Data memories stay the same as cycle two.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x02, \$X12 = x0A.

Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = x0, \$X22 = x0. DMEM are set to DMEM(0x0) = x1, DMEM(0x8) = x2, DMEM(0x16) = x3, DMEM(0x24) = x4.

b IF/ID Register

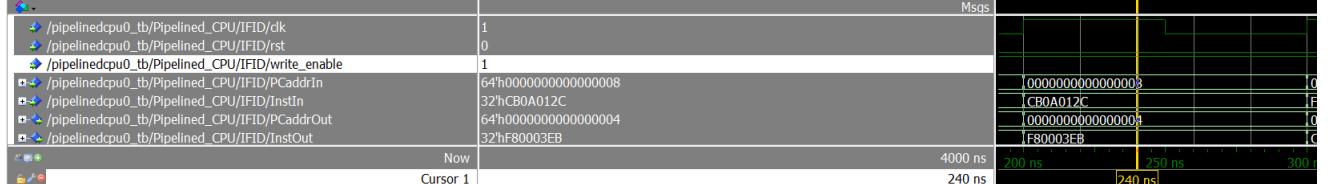


Figure 8: IF/ID Signals of Cycle three



Figure 9: PCSrc Signal of Cycle three

At 200 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x04 to x08. The output signals of IF/ID register are also assigned with the input signals of cycle two.

PCaddrOut = x4 \leftarrow x0, InstOut = xF80003EB \leftarrow x8B0A012B.

c ID/EX Register

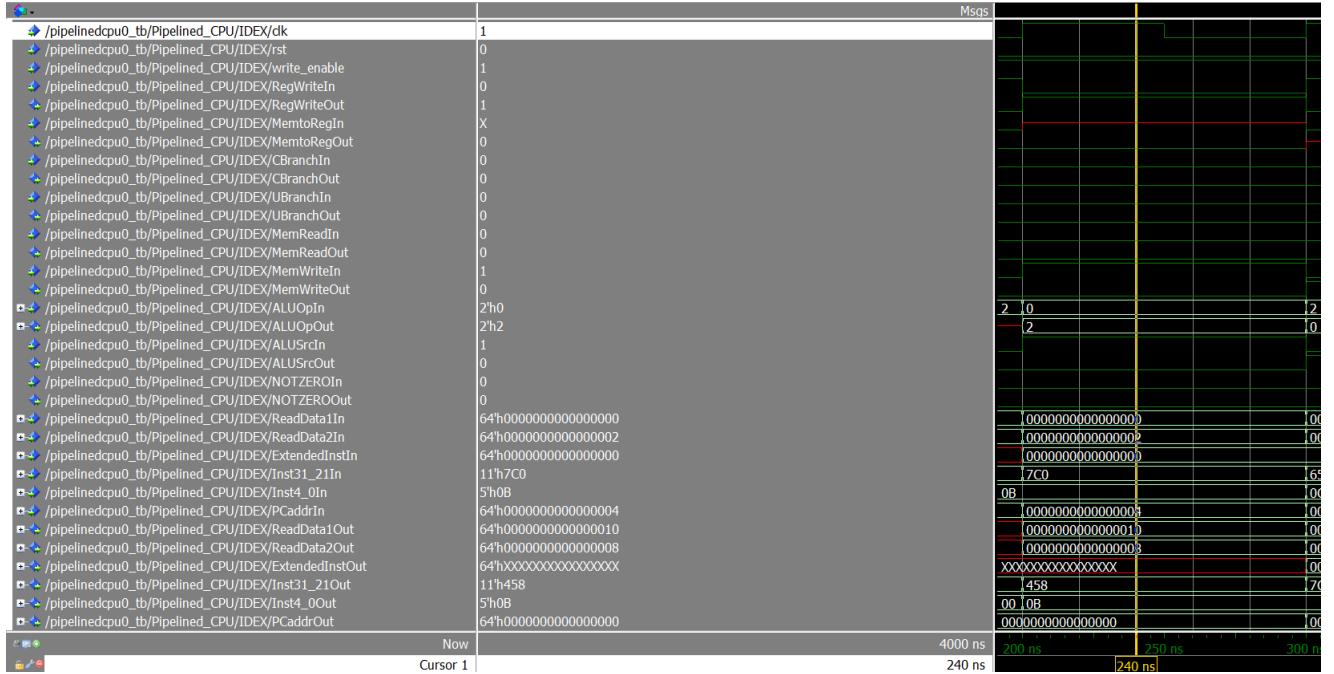


Figure 10: ID/EX Signals of Cycle three

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction two(STUR X11, [XZR, 0]) decoding. ReadData1In = \$X0 = 0, ReadData2In = \$X10 = x08, ExtendedInstIn = x0, PCaddrIn = x4.

All the output signals are set according to instruction one, and they would be assigned according to instruction two at the rising edge of next cycle.

d EX/MEM Register

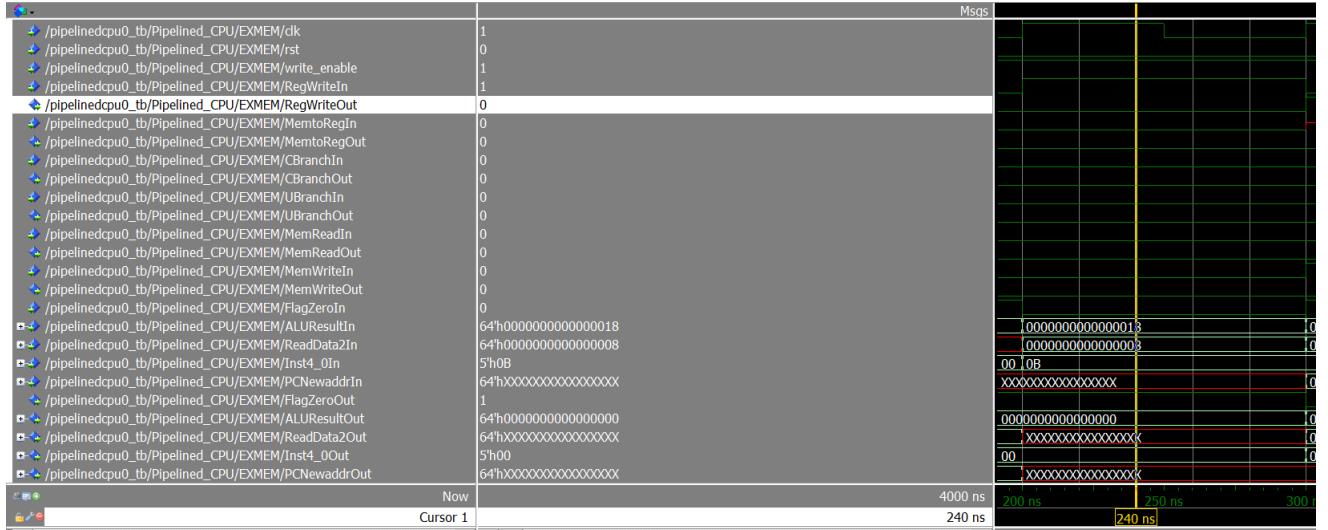


Figure 11: EX/MEM Signals of Cycle three

EX/MEM register gets CPU control signals from ID/EX register, all the input control signals are set according to the ID/EX register.

The other input signals are set according to the execution of instruction one(ADD X11, X9, X10). ALUResultIn = \$X19 + \$X10 = x18, FlagZeroIn = 0, ReadData2In = \$X11 = x02, PCNewaddrIn is don't care here since it's a R-type instruction.

All the output signals are not decided here, and they would be assigned according to the input signals here at the rising edge of next cycle.

5: Cycle Four

a Debug signals

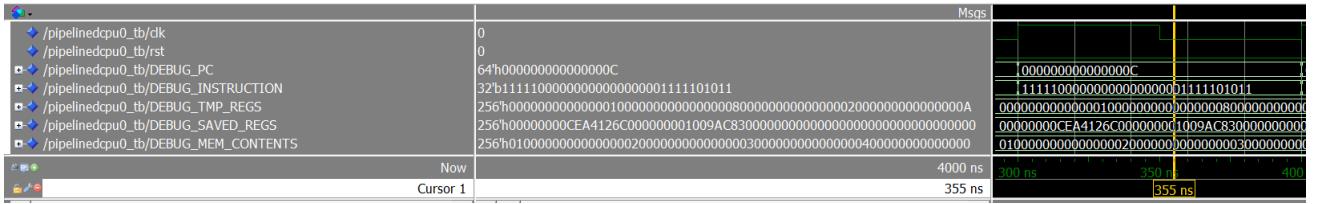


Figure 12: Debug Signals of Cycle four

In cycle four, we fetch instruction at PC address of x0C, the instruction is 1111 1000 0000 0000 0000 0011 1110 1011.

Registers and Data memories stay the same as cycle three.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x02, \$X12 = x0A.

Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = x0, \$X22 = x0. DMEM are set to DMEM(0x0) = x1, DMEM(0x8) = x2, DMEM(0x16) = x3, DMEM(0x24) = x4.

b IF/ID Register

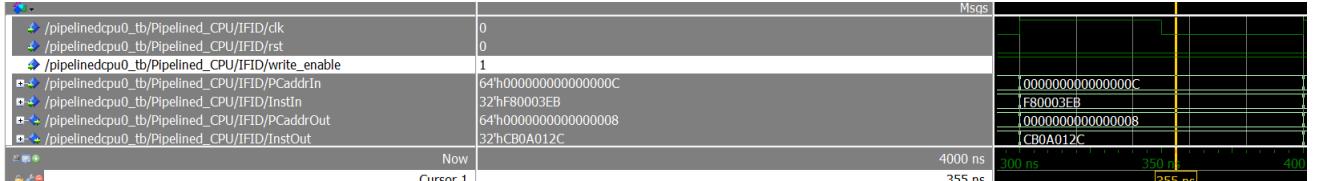


Figure 13: IF/ID Signals of Cycle four



Figure 14: PCSrc Signal of Cycle four

At 300 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x08 to x0C. The output signals of IF/ID register are assigned with the input signals of cycle three.

PCaddrOut = x8 \leftarrow x4, InstOut = xCB0A012C \leftarrow xF80003EB.

c ID/EX Register

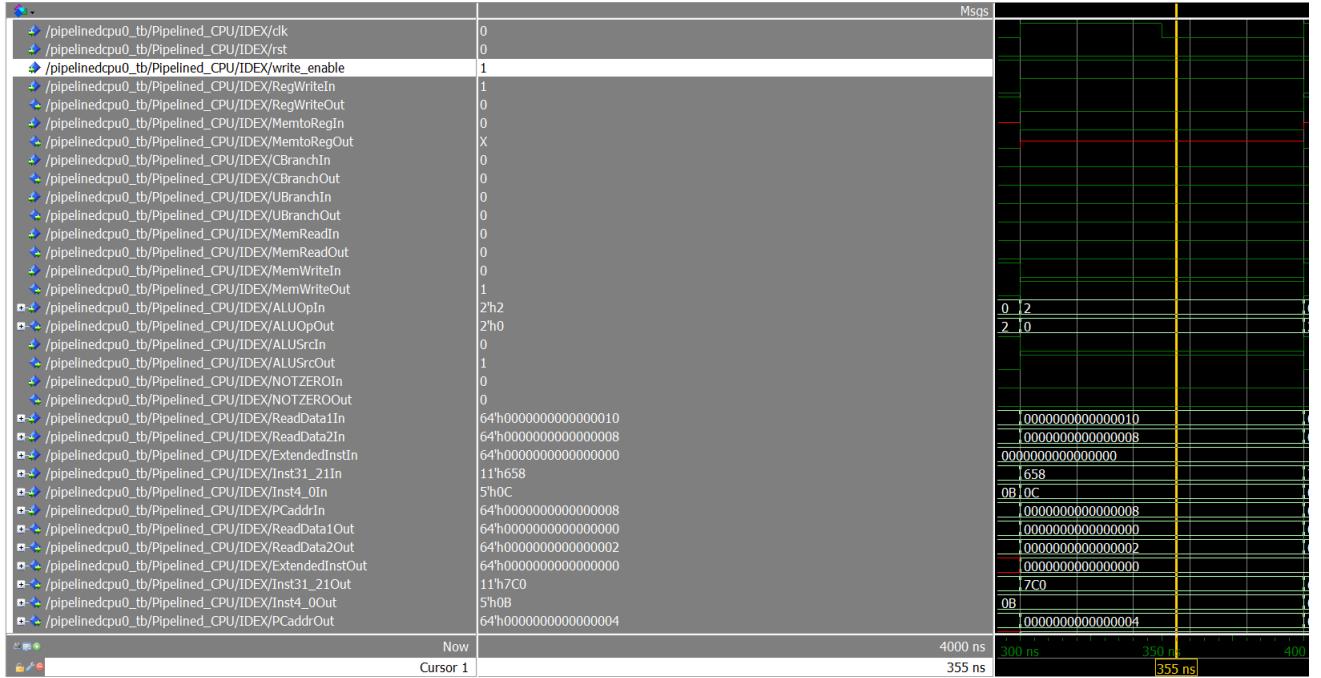


Figure 15: ID/EX Signals of Cycle four

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction three(SUB X12, X9, X10) decoding. ReadData1In = \$X9 = x10, ReadData2In = \$X10 = x08, ExtendedInstIn = x0(don't care here), PCAddrIn = x8.

All the output signals are set according to instruction two, and they would be assigned according to instruction three at the rising edge of next cycle.

d EX/MEM Register

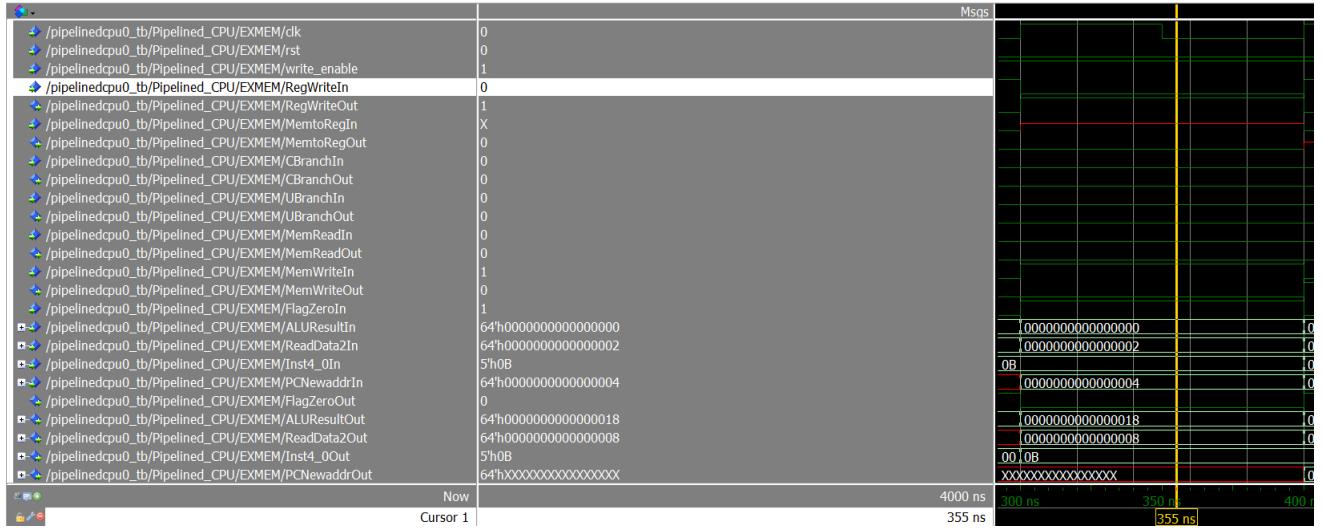


Figure 16: EX/MEM Signals of Cycle four

EX/MEM register gets CPU control signals from ID/EX register, all the input control signals are set according to the ID/EX register.

The other input signals are set according to the execution of instruction two(STUR X11, [XZR, 0]). ALUResultIn = \$XZR + 0 = x0, FlagZeroIn = 1, ReadData2In = \$X11 = x02, PCNewaddrIn is don't care here since it's a D-type instruction.

All the output signals are set according to instruction one, and they would be assigned according to instruction two at the rising edge of next cycle.

e MEM/WB Register

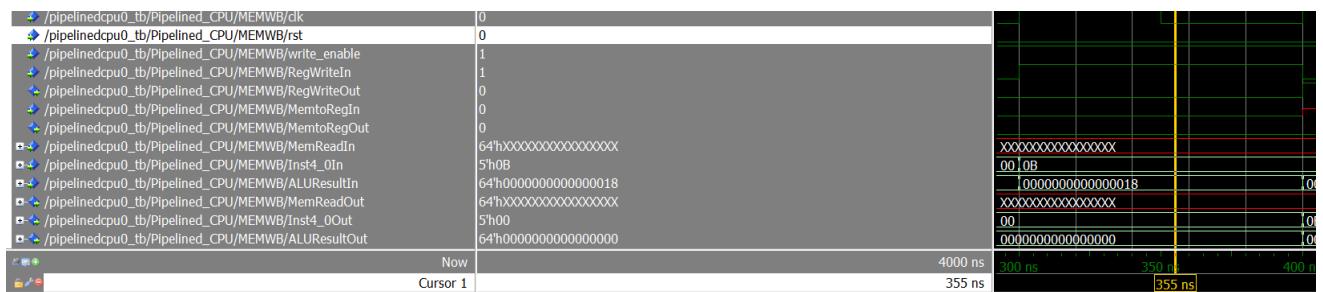


Figure 17: MEM/WB Signals of Cycle four

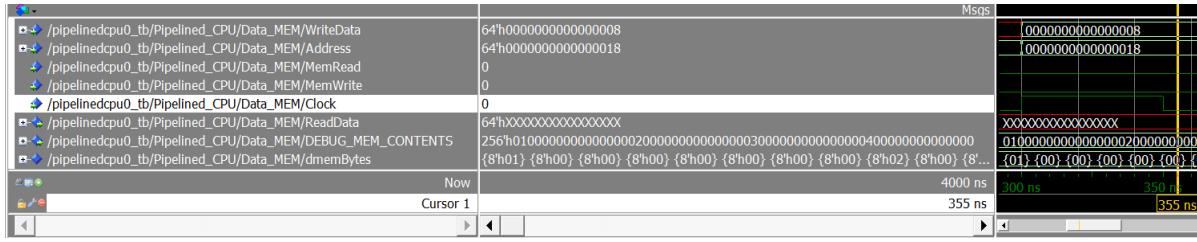


Figure 18: Data memory Signals of Cycle four

MEM/WB register gets CPU control signals from EX/MEM register, all the input control signals are set according to the EX/MEM register.

The other input signals are set according to the memory operation of instruction one(ADD X11, X9, X10). MemRead = 0 and MemWrite = 0, data memory did nothing here. The ALUResultIn = x18, came from EX/MEM.

All the output signals are not decided, and they would be assigned according to instruction one at the rising edge of next cycle.

6: Cycle Five

a Debug signals

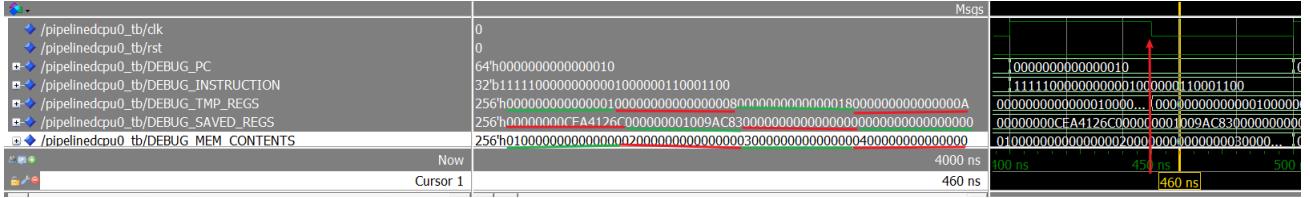


Figure 19: Debug Signals of Cycle five

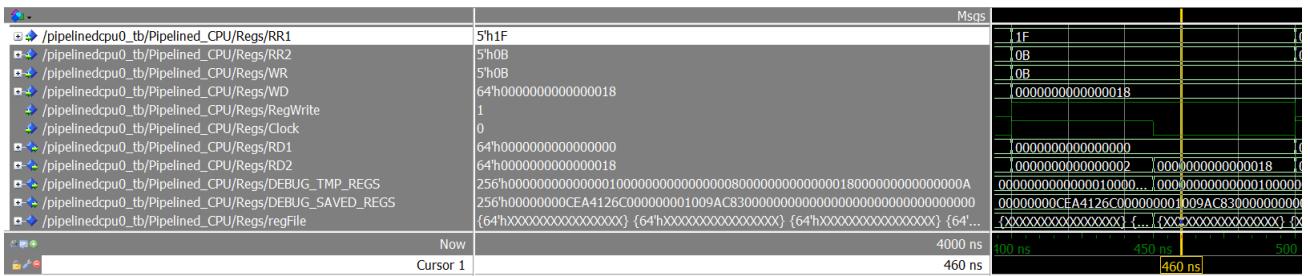


Figure 20: Debug Signals of Cycle five

In cycle five, we fetch instruction at PC address of x10, the instruction is 1111 1000 0000 0000 1000 0001 1000 1100.

Temp registers are changed at 450 ns, a falling edge of 'clk'. $\&X11 = x18 \leftarrow x02$, according to instruction one($\$X11 = \$X9 + \$X10 = x10 + x8$). Saved registers and DMEM are the same as the cycle four.

Saved registers are set to $\$X19 = xCEA4126C$, $\$X20 = x1009AC83$, $\$X21 = x0$, $\$X22 = x0$. DMEM are set to DMEM(0x0) = x1, DMEM(0x8) = x2, DMEM(0x16) = x3, DMEM(0x24) = x4.

This instruction is correctly operated.

b IF/ID Register

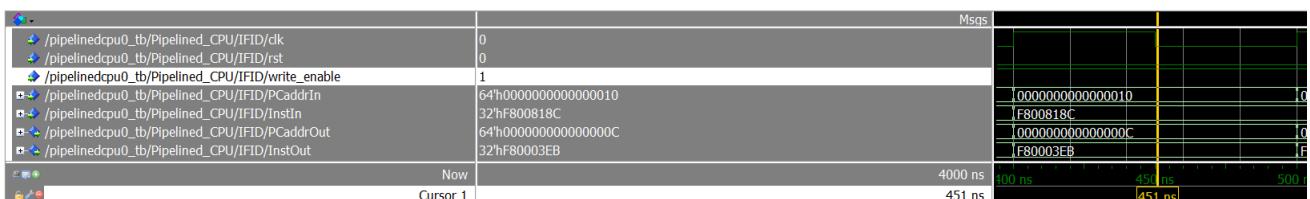


Figure 21: IF/ID Signals of Cycle five



Figure 22: PCSrc Signal of Cycle five

At 400 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x0C to x10. The output signals of IF/ID register are assigned with the input signals of cycle four.

PCaddrOut = xC \leftarrow x8, InstOut = xF80003EB \leftarrow xCB0A012C.

c ID/EX Register

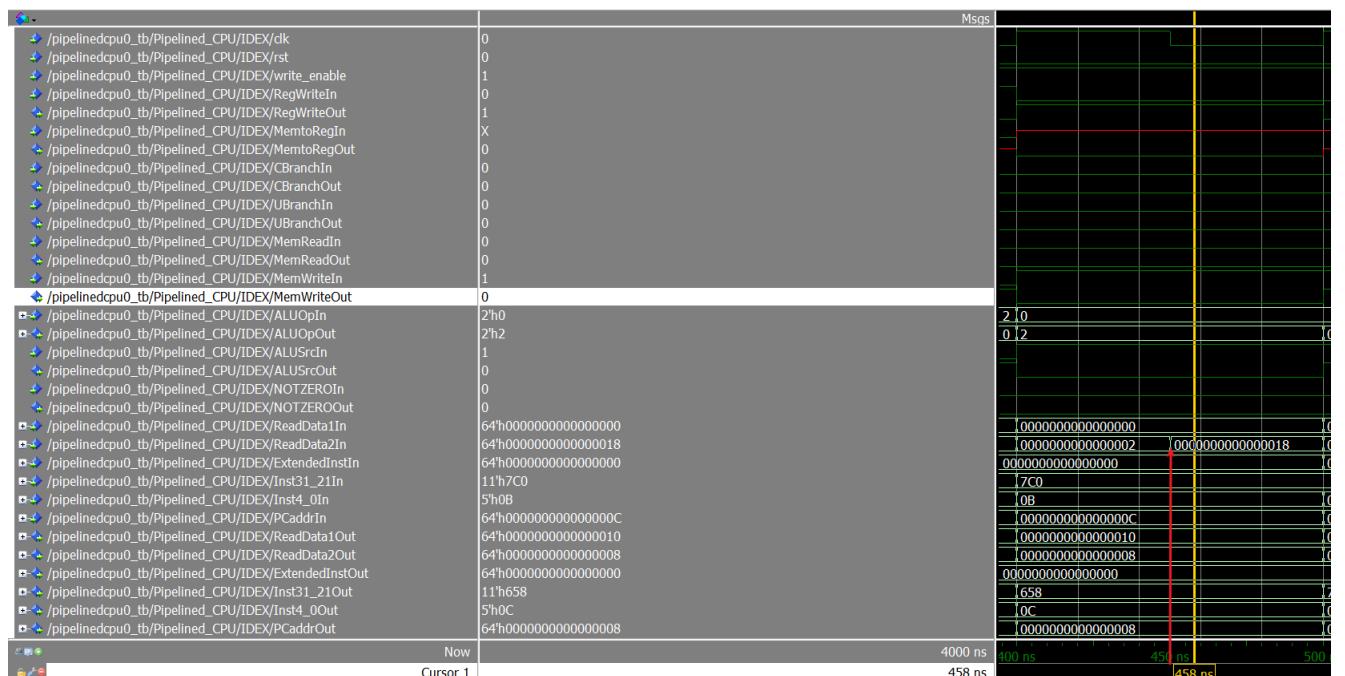


Figure 23: ID/EX Signals of Cycle five

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction four(STUR X11, [XRZ, 0]) decoding. ReadData1In = \$XRZ = x0, ReadData2In = \$X11 = x18 \leftarrow x2, reading data from registers is happening all the time, and the value of X11 is changed at 450 ns, ReadData2In changes correspondingly. ExtendedInstIn = x0(don't care here), PCaddrIn = xC.

All the output signals are set according to instruction three, and they would be assigned according to instruction four at the rising edge of next cycle.

d EX/MEM Register

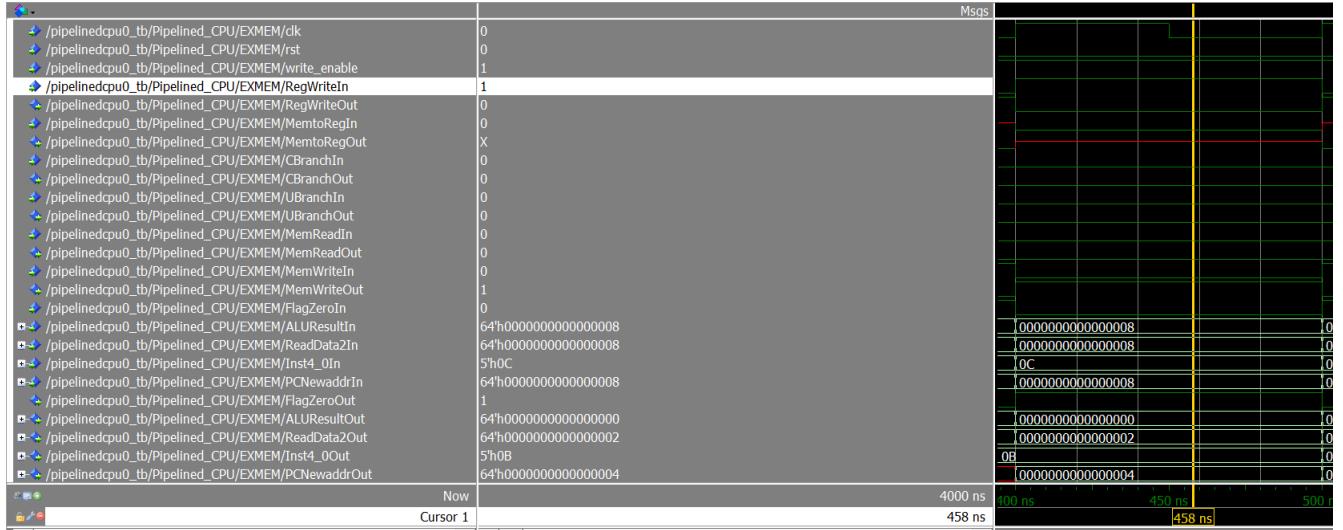


Figure 24: EX/MEM Signals of Cycle five

EX/MEM register gets CPU control signals from ID/EX register, all the input control signals are set according to the ID/EX register.

The other input signals are set according to the execution of instruction three(SUB X12, X9, X10). ALUResultIn = \$X9 - \$X10 = x10 - x8 = x8, FlagZeroIn = 0, ReadData2In = \$X10 = x08, PCNewaddrIn is don't care here since it's a R-type instruction.

All the output signals are set according to instruction two, and they would be assigned according to instruction three at the rising edge of next cycle.

e MEM/WB Register

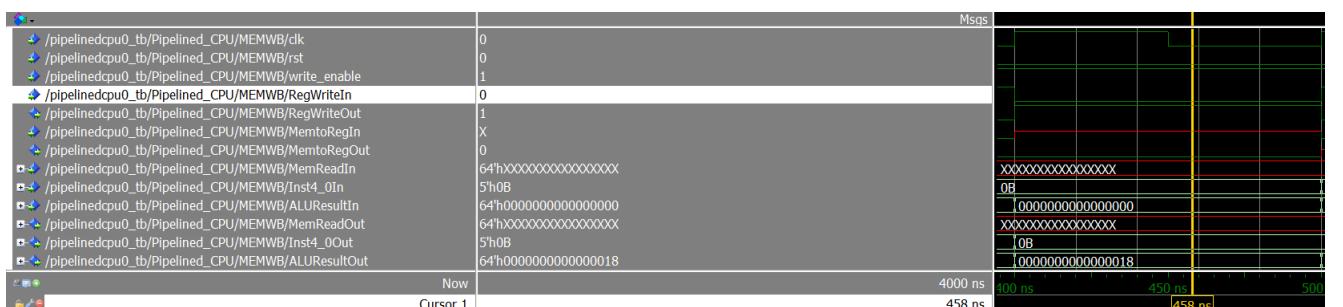


Figure 25: MEM/WB Signals of Cycle five

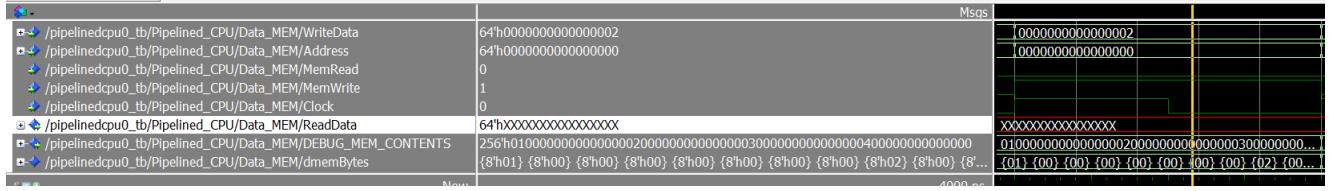


Figure 26: Data memory Signals of Cycle five

MEM/WB register gets CPU control signals from EX/MEM register, all the input control signals are set according to the EX/MEM register.

The other input signals are set according to the memory operation of instruction two(STUR X11, X9, X10). MemRead = 0 and MemWrite = 1, data memory will store WriteData = x02 into DMEM[Address=x0] at the next rising edge of 'clk'. The ALUResultIn = x0, came from EX/MEM.

All the output signals are set according to instruction one, and they would be assigned according to instruction two at the rising edge of next cycle.

7: Cycle Six

a Debug signals

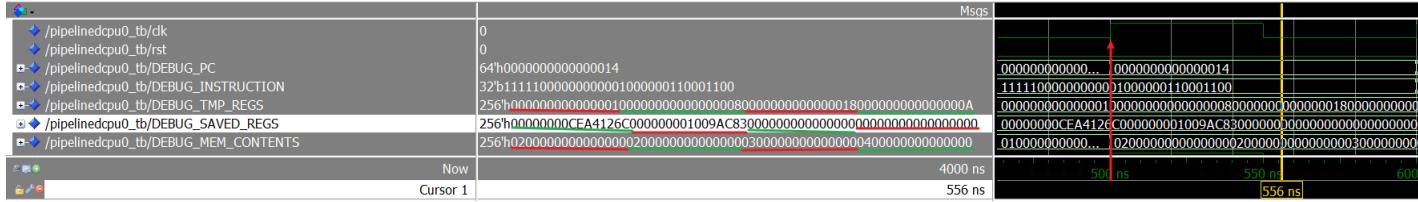


Figure 27: Debug Signals of Cycle six

In cycle six, we fetch instruction at PC address of x14, the instruction is 1111 1000 0000 0000 1000 0001 1000 1100.

Saved registers and temp registers are the same as the cycle five.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x18, \$X12 = x0A.

Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = x0, \$X22 = x0.

DMEM are set to DMEM(0x0) = x2 at 500 ns, a rising edge of 'clk'. DMEM(0x8) = x2, DMEM(0x16) = x3, DMEM(0x24) = x4.

This instruction has data hazard, we are supposed to write x18 into DMEM(0x0).

b IF/ID Register

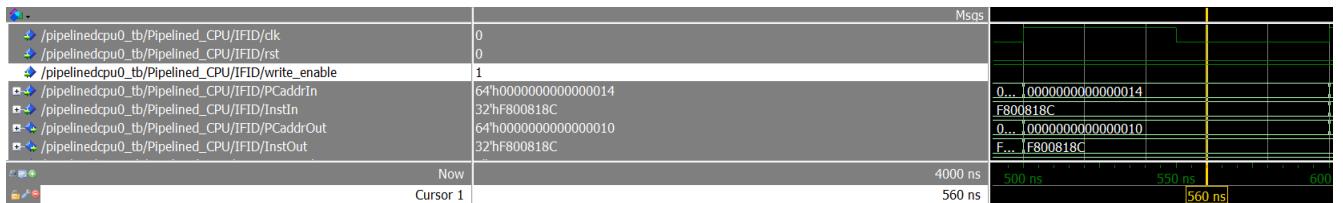


Figure 28: IF/ID Signals of Cycle six



Figure 29: PCSrc Signal of Cycle six

At 500 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x10 to x14. The output signals of IF/ID register are assigned with the input signals of cycle five.

PCAddrOut = x10 \leftarrow x0C, InstOut =xF800818C \leftarrow xF80003EB.

c ID/EX Register

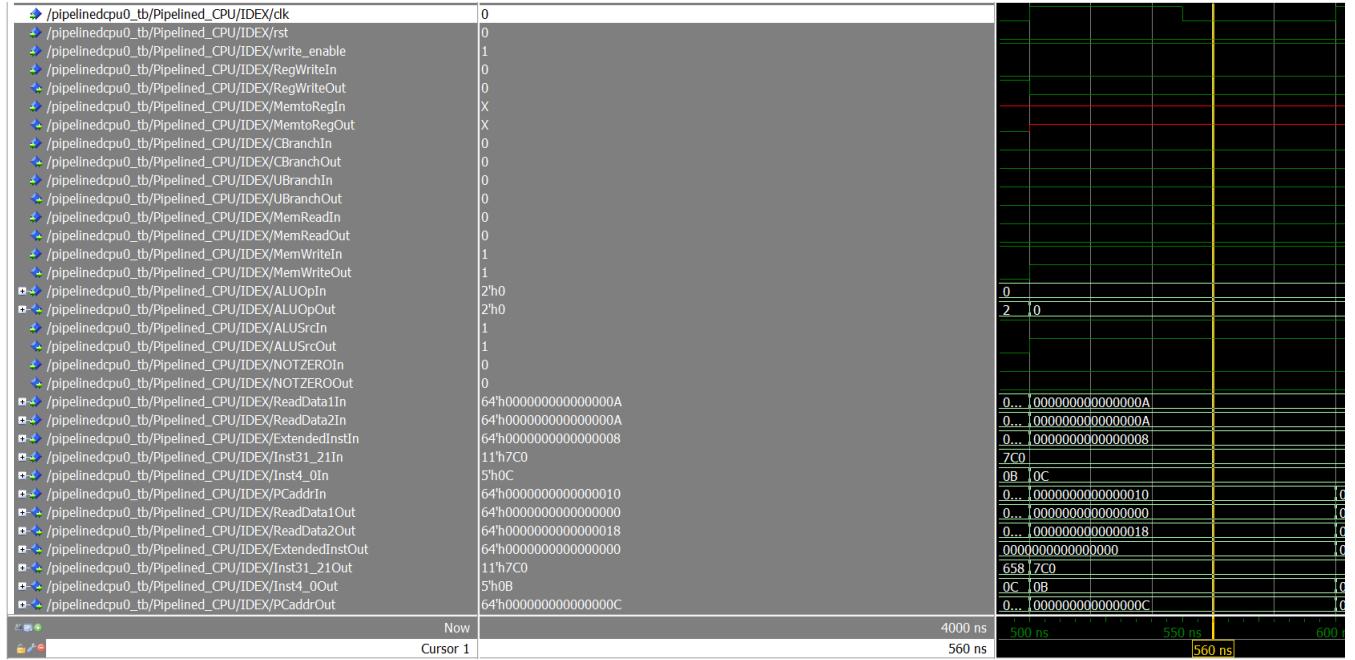


Figure 30: ID/EX Signals of Cycle six

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction five(STUR X12, [X12,8]) decoding. ReadData1In = \$X12 = xA, ReadData2In = \$X12 = xA, ExtendedInstIn = x8, PCaddrIn = x10.

All the output signals are set according to instruction four, and they would be assigned according to instruction five at the rising edge of next cycle.

d EX/MEM Register

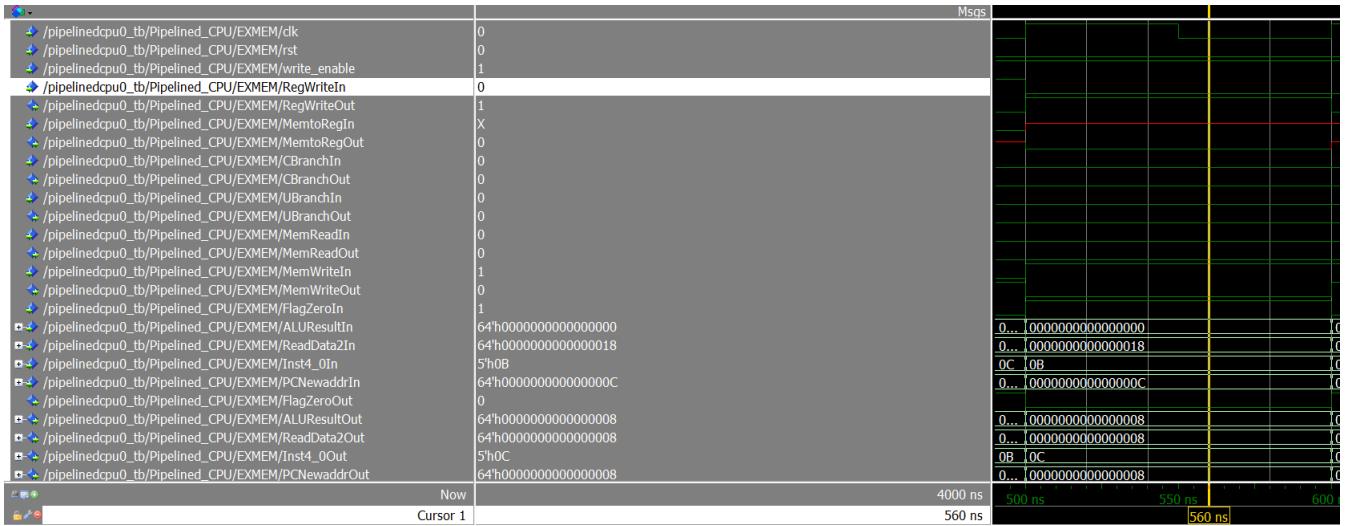


Figure 31: EX/MEM Signals of Cycle six

EX/MEM register gets CPU control signals from ID/EX register, all the input control signals are set according to the ID/EX register.

The other input signals are set according to the execution of instruction four(STUR X11, [XZR,0]). ALUResultIn = \$XZR + x0 = x0 + x0 = x0, FlagZeroIn = 1, ReadData2In = \$X11 = x18, PCNewaddrIn is don't care here since it's a D-type instruction.

All the output signals are set according to instruction three, and they would be assigned according to instruction four at the rising edge of next cycle.

e MEM/WB Register

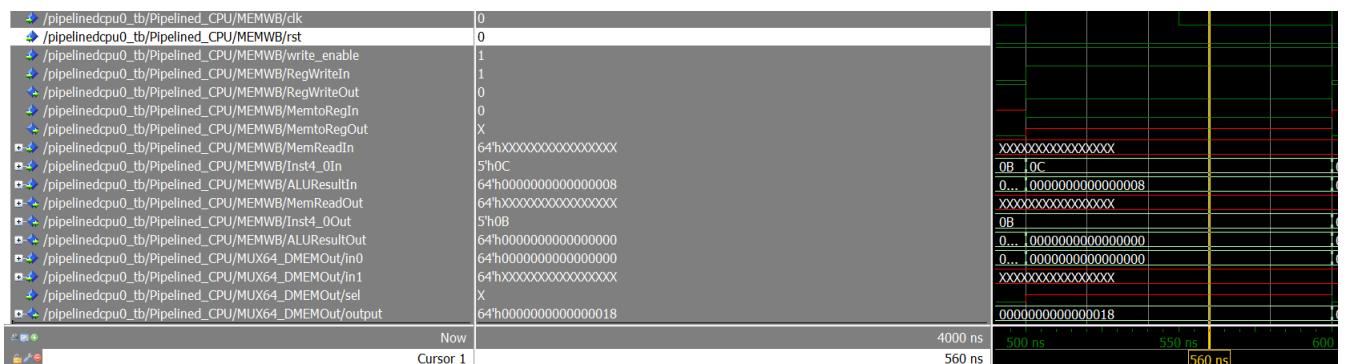


Figure 32: MEM/WB Signals of Cycle six

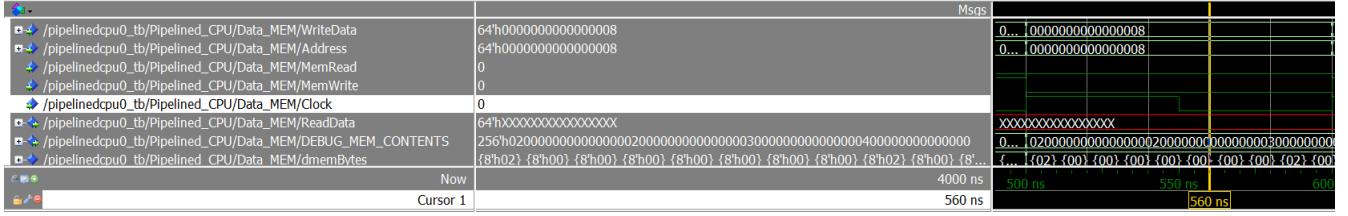


Figure 33: Data memory Signals of Cycle six

MEM/WB register gets CPU control signals from EX/MEM register, all the input control signals are set according to the EX/MEM register.

The other input signals are set according to the memory operation of instruction three(SUB X12, X9, X10). MemRead = 0 and MemWrite = 0, data memory will do nothing at the next rising edge of 'clk'. The ALUResultIn = x8, came from EX/MEM.

All the output signals are set according to instruction two, and they would be assigned according to instruction three at the rising edge of next cycle.

8: Cycle Seven

a Debug signals

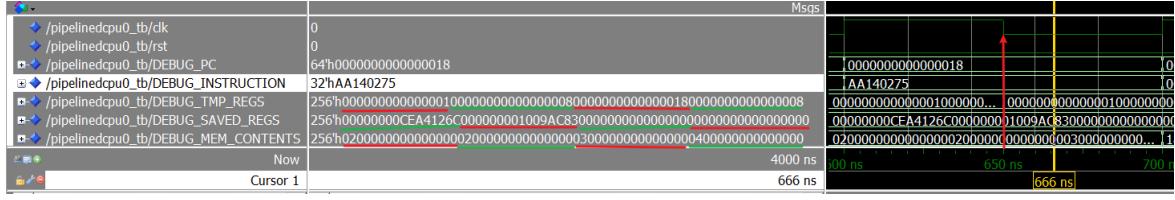


Figure 34: Debug Signals of Cycle seven

In cycle seven, we fetch instruction at PC address of x18, the instruction is 1010 1010 0001 0100 0000 0010 0111 0101.

Temp registers are changed at 650 ns, a falling edge of 'clk'. $\&X12 = x8 \leftarrow xA$, according to instruction three($X12 = X9 - X10 = x10 - x8 = x8$). Saved registers and DMEM are the same as the cycle six.

Saved registers are set to $X19 = xCEA4126C$, $X20 = x1009AC83$, $X21 = x0$, $X22 = x0$. DMEM are set to DMEM(0x0) = x2, DMEM(0x8) = x2, DMEM(0x16) = x3, DMEM(0x24) = x4.

This instruction is correctly operated.

b IF/ID Register



Figure 35: IF/ID Signals of Cycle seven



Figure 36: PCSrc Signal of Cycle seven

At 600 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x14 to x18. The output signals of IF/ID register are assigned with the input signals of cycle six.

PCaddrOut = x14 \leftarrow x10, InstOut = xF800818C \leftarrow xF800818C(two same instructions in a row).

c ID/EX Register

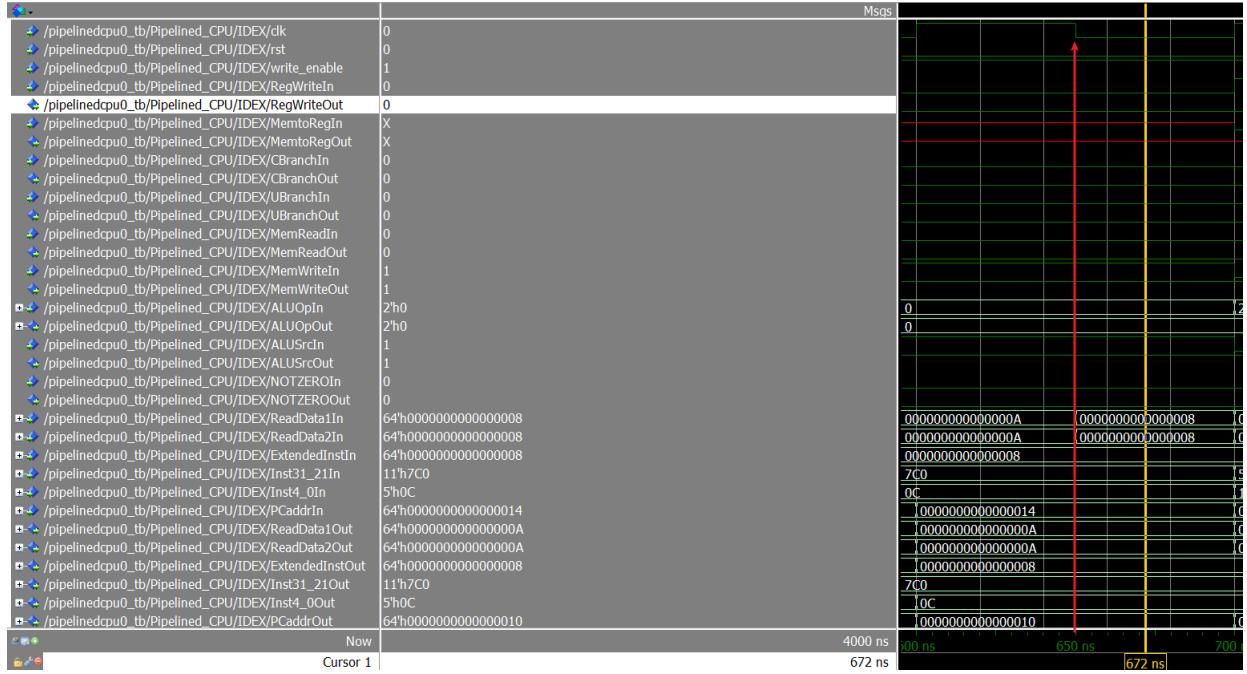


Figure 37: ID/EX Signals of Cycle seven

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction six(STUR X12, [X12,8]) decoding. ReadData1In = \$X12 = x8 ← xA, ReadData2In = \$X12 = x8 ← xA. ReadData change due to value changing in registers at 650 ns, a falling edge of 'clk'. ExtendedInstIn = x8, PCaddrIn = x14.

All the output signals are set according to instruction five, and they would be assigned according to instruction six at the rising edge of next cycle.

d EX/MEM Register

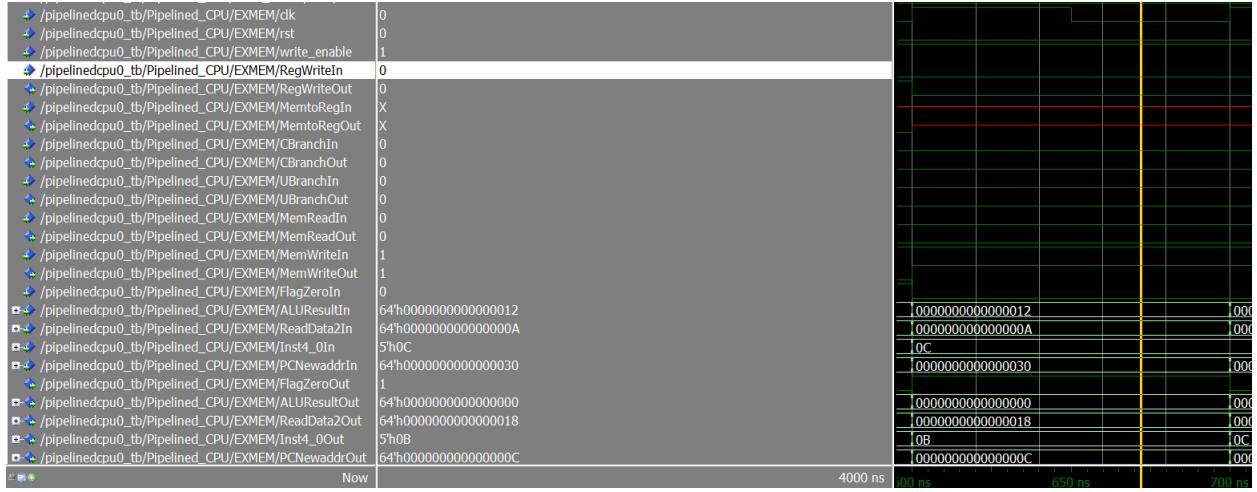


Figure 38: EX/MEM Signals of Cycle seven

EX/MEM register gets CPU control signals from ID/EX register, all the input control signals are set according to the ID/EX register.

The other input signals are set according to the execution of instruction five(STUR X12, [X12,8]). ALUResultIn = \$X12 + x8 = xA + x8 = x12. The ALUResult here is not what we expected, the value of \$X12 is changed from xA to x8 at 650 ns, and we calculate with xA here, so the instruction five has data hazard. FlagZeroIn = 0, ReadData2In = \$X12 = xA, PCNewaddrIn is don't care here since it's a D-type instruction.

All the output signals are set according to instruction four, and they would be assigned according to instruction five at the rising edge of next cycle.

e MEM/WB Register

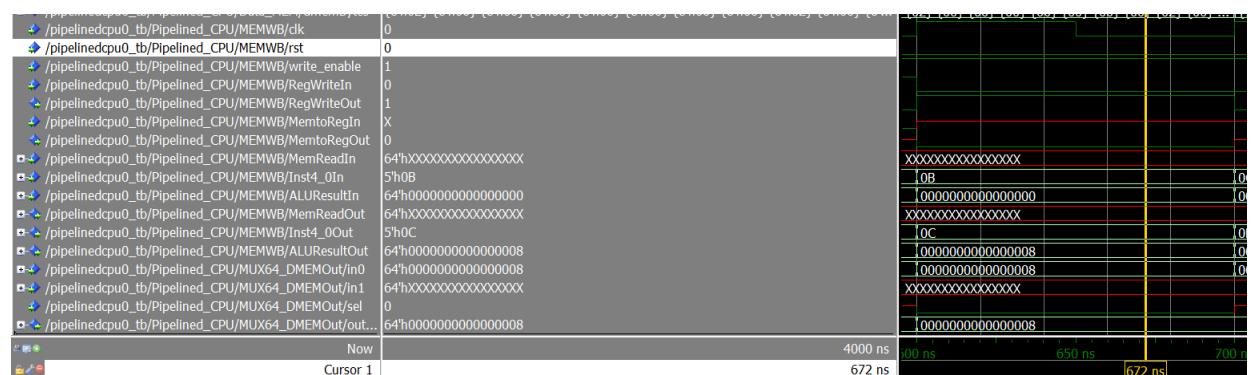


Figure 39: MEM/WB Signals of Cycle seven

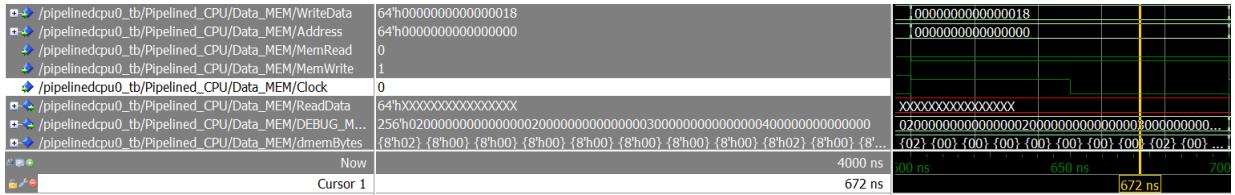


Figure 40: Data memory Signals of Cycle seven

MEM/WB register gets CPU control signals from EX/MEM register, all the input control signals are set according to the EX/MEM register.

The other input signals are set according to the memory operation of instruction four(STUR X11, [XZR,0]). MemRead = 0 and MemWrite = 1, data memory will store WriteData = \$X11 = x18 into DMEM[Address=x0] at the next rising edge of 'clk'. The ALUResultIn = x0, came from EX/MEM.

All the output signals are set according to instruction three, and they would be assigned according to instruction four at the rising edge of next cycle.

9: Cycle Eight

a Debug signals

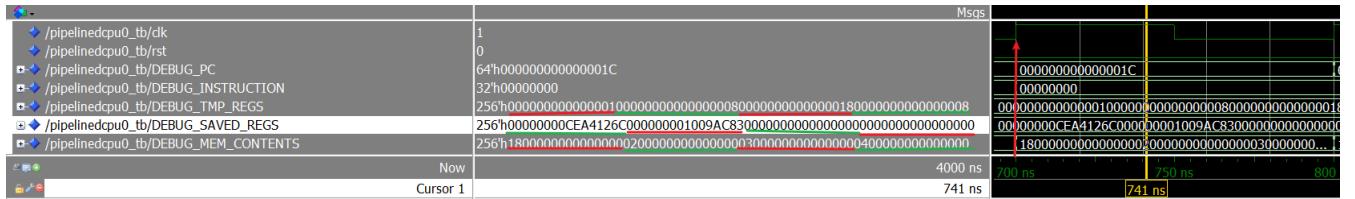


Figure 41: Debug Signals of Cycle eight

In cycle eight, we fetch instruction at PC address of x1C, the instruction is NOP.

Saved registers and temp registers are the same as the cycle seven.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x18, \$X12 = x08.

Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = x0, \$X22 = x0.

DMEM are set to DMEM(0x0) = x18 at 700 ns due to instruction four(STUR X11, [XZR,0]), a rising edge of 'clk'. DMEM(0x8) = x2, DMEM(0x16) = x3, DMEM(0x24) = x4.

This instruction is correctly operated.

b IF/ID Register

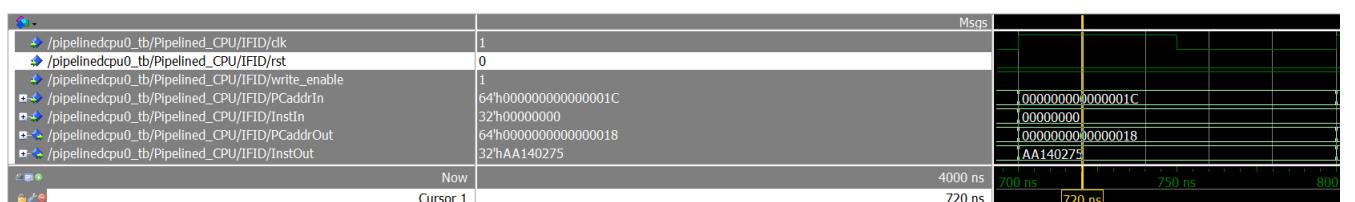


Figure 42: IF/ID Signals of Cycle eight



Figure 43: PCSrc Signal of Cycle eight

At 700 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x18 to x1C. The output signals of IF/ID register are assigned with the input signals of cycle seven.

PCaddrOut = x18 \leftarrow x14, InstOut = xAA140275 \leftarrow xF800818C.

c ID/EX Register

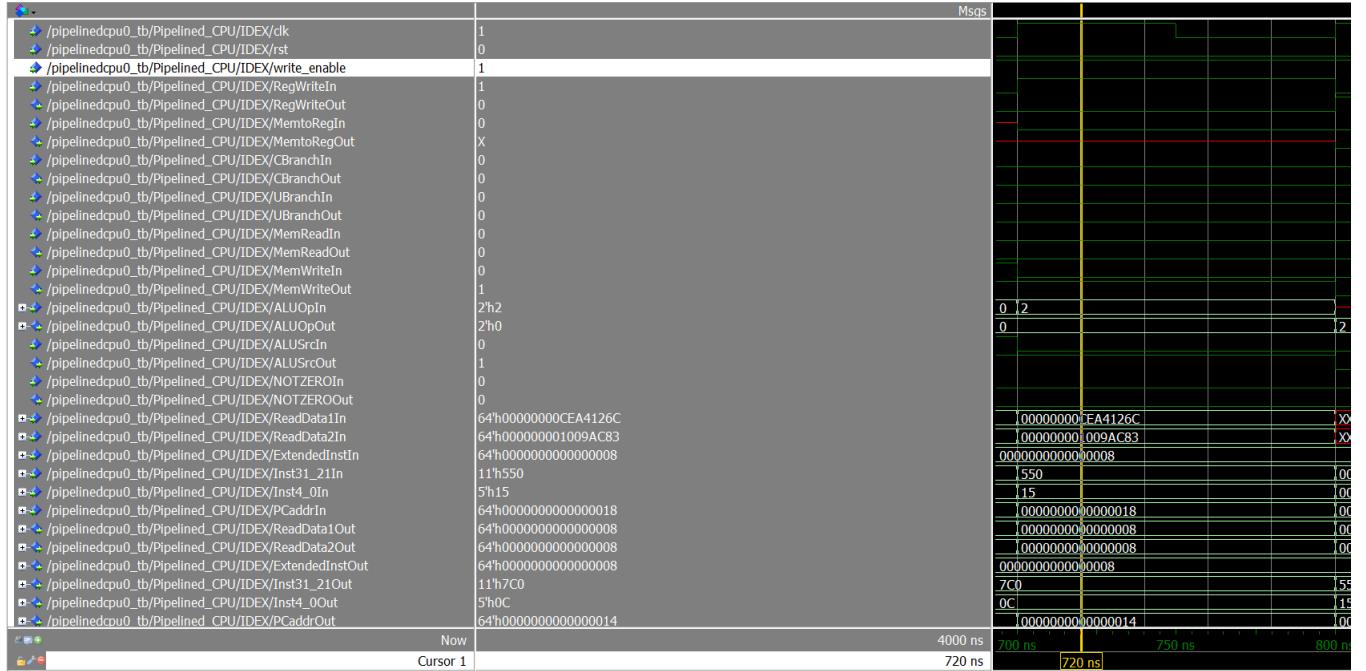


Figure 44: ID/EX Signals of Cycle eight

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction seven(ORR X21, X19, X20) decoding. ReadData1In = \$X19 = xCEA4126C, ReadData2In = \$X20 = x1009AC83, ExtendedInstIn = x8, PCAddrIn = x18.

All the output signals are set according to instruction six, and they would be assigned according to instruction seven at the rising edge of next cycle.

d EX/MEM Register

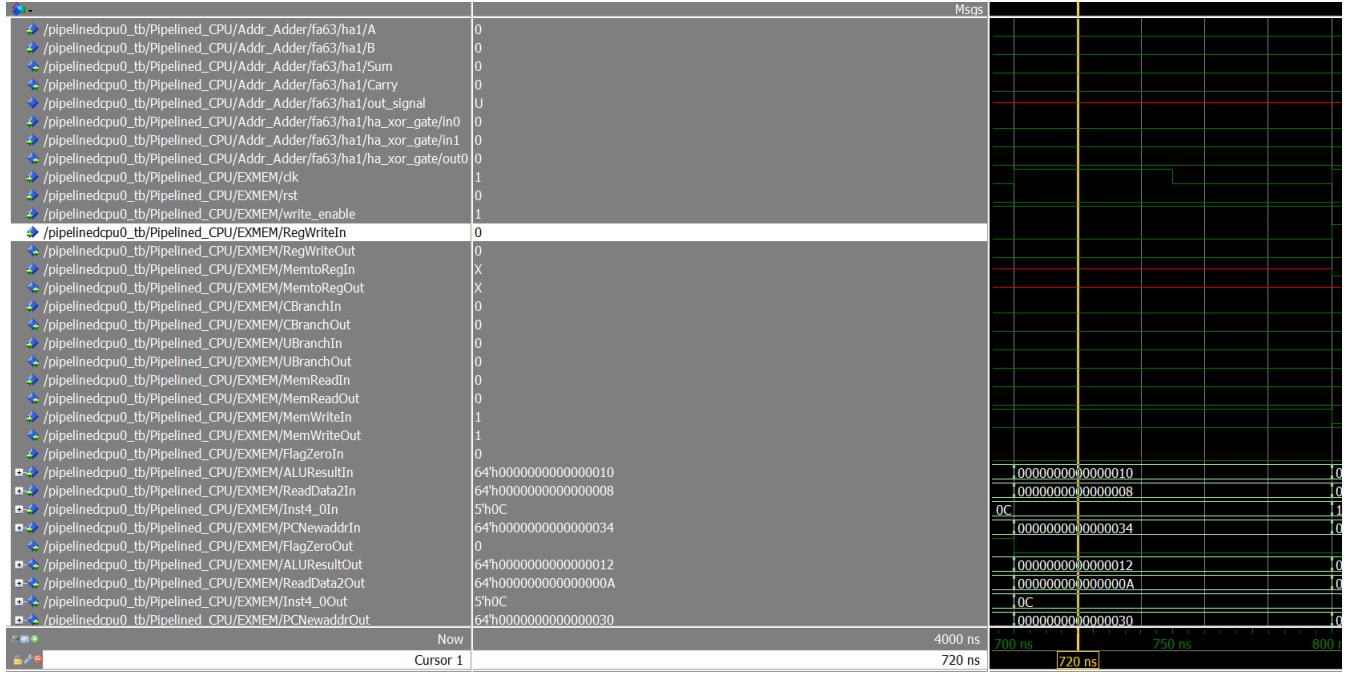


Figure 45: EX/MEM Signals of Cycle eight

EX/MEM register gets CPU control signals from ID/EX register, all the input control signals are set according to the ID/EX register.

The other input signals are set according to the execution of instruction six(STUR X12, [X12,8]). ALUResultIn = \$X12 + x8 = x8 + x8 = x10. FlagZeroIn = 0, ReadData2In = \$X12 = x8, PCNewaddrIn is don't care here since it's a D-type instruction.

All the output signals are set according to instruction five, and they would be assigned according to instruction six at the rising edge of next cycle.

e MEM/WB Register

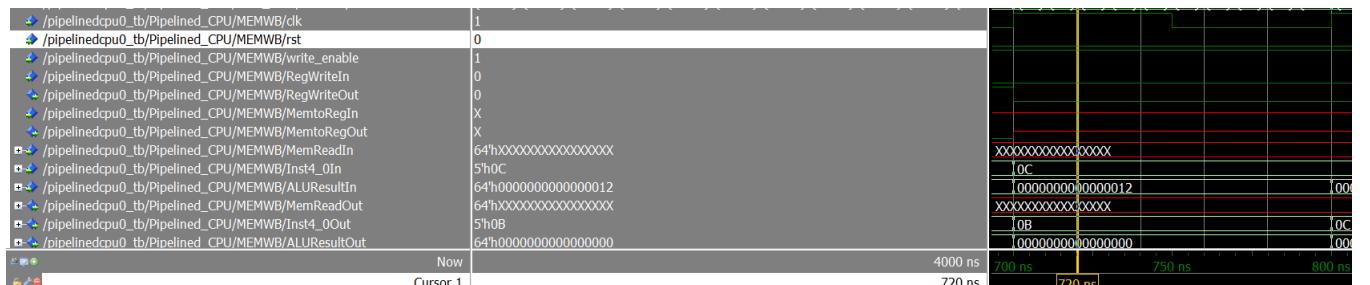


Figure 46: MEM/WB Signals of Cycle eight



Figure 47: Data memory Signals of Cycle eight

MEM/WB register gets CPU control signals from EX/MEM register, all the input control signals are set according to the EX/MEM register.

The other input signals are set according to the memory operation of instruction five(STUR X12, [X12,8]). MemRead = 0 and MemWrite = 1, data memory will store WriteData = \$X12 = xA into DMEM[Address=x12] at the next rising edge of 'clk'. The ALUResultIn = x12, came from EX/MEM.

All the output signals are set according to instruction four, and they would be assigned according to instruction five at the rising edge of next cycle.

10: Cycle Nine

a Debug signals

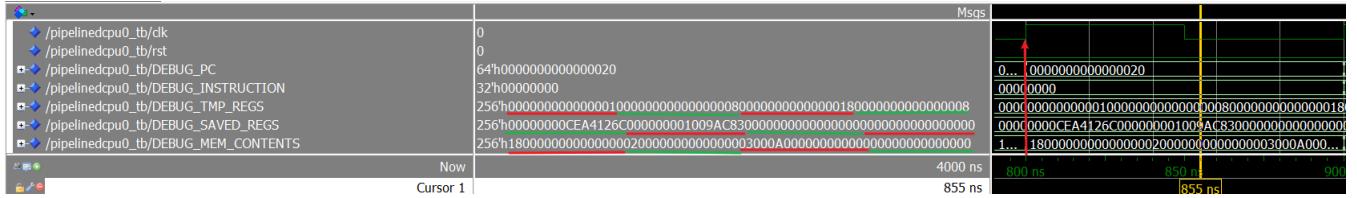


Figure 48: Debug Signals of Cycle nine

In cycle nine, we fetch instruction at PC address of x1C, the instruction is NOP. Saved registers and temp registers are the same as the cycle eight. Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x18, \$X12 = x08. Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = x0, \$X22 = x0. DMEM are set to DMEM[18-25] = x0A at 800 ns due to instruction five(STUR X12, [X12,8]), a rising edge of 'clk'. DMEM(0x0) = x18, DMEM(0x8) = x2, DMEM(0x16) = x3. This instruction has data hazard, we are supposed to use \$X12 = x8(the result of instruction three) in this instruction, however the value has not updated into registers yet, and we use a wrong value of \$X12 in instruction five.

b IF/ID Register

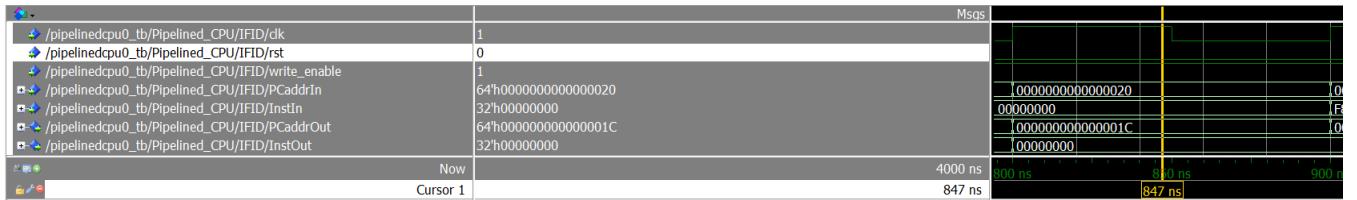


Figure 49: IF/ID Signals of Cycle nine



Figure 50: PCSrc Signal of Cycle nine

At 800 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x1C to x20. The output signals of IF/ID register are assigned with the input signals of cycle eight.
 $\text{PCaddrOut} = \text{x1C} \leftarrow \text{x18}$, $\text{InstOut} = \text{x00000000} \leftarrow \text{xAA140275}$.

c ID/EX Register

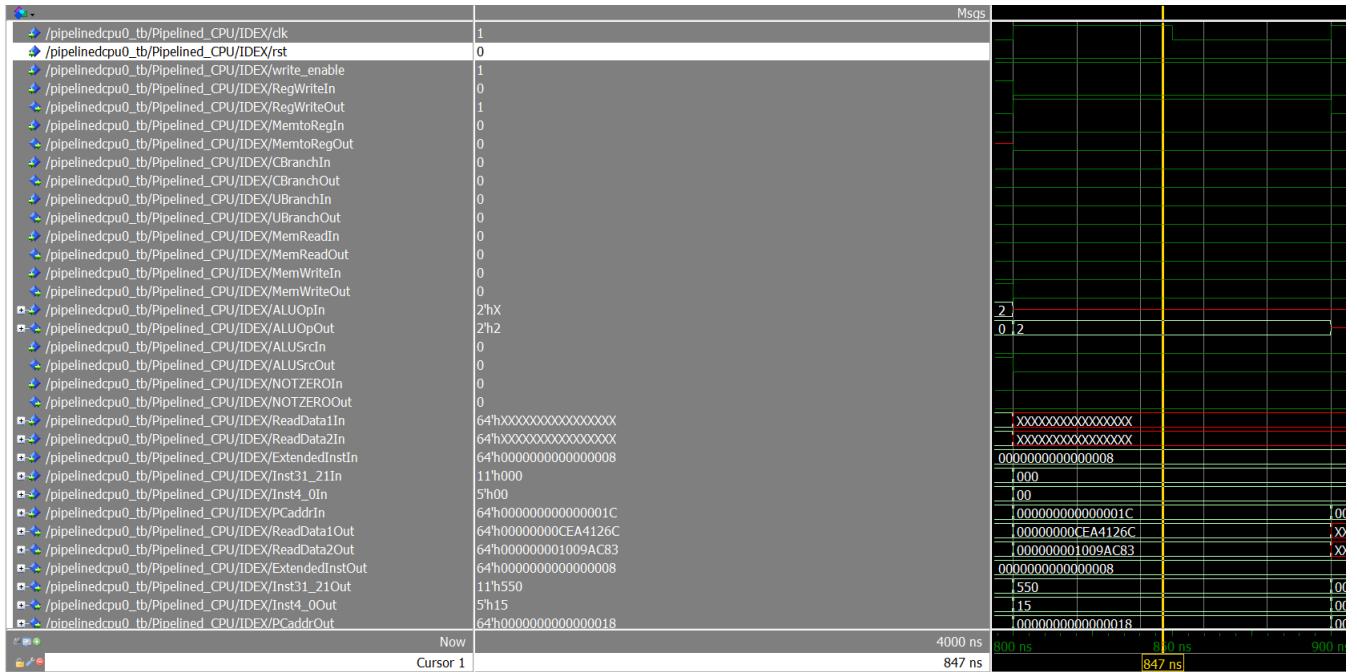


Figure 51: ID/EX Signals of Cycle nine

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction eight(NOP) decoding.

All the output signals are set according to instruction seven, and they would be assigned according to instruction eight at the rising edge of next cycle.

d EX/MEM Register

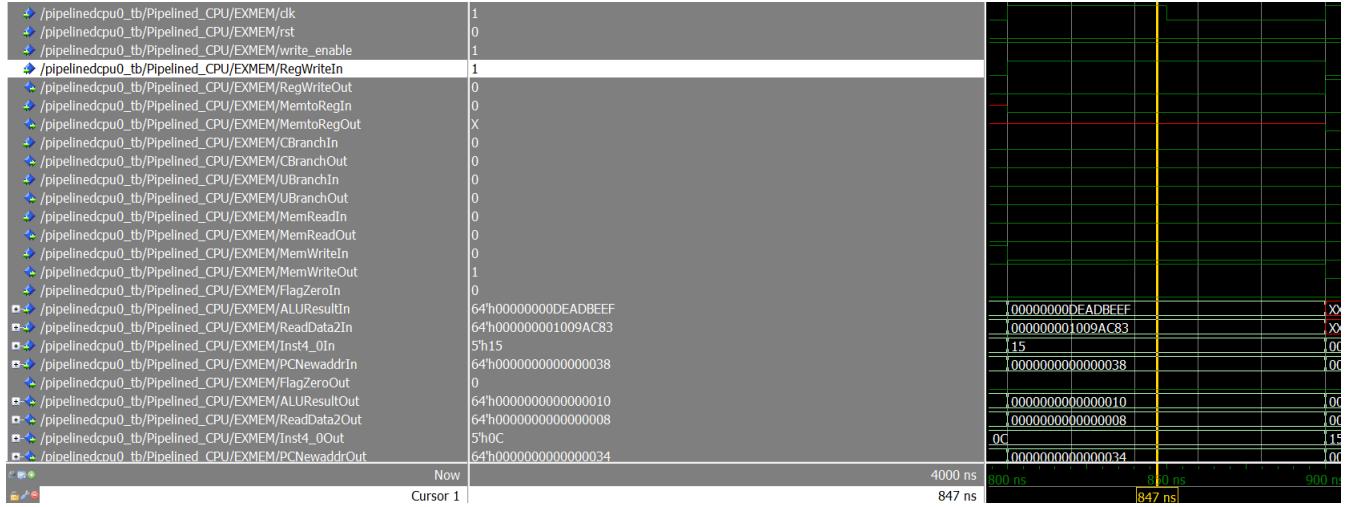


Figure 52: EX/MEM Signals of Cycle nine

EX/MEM register gets CPU control signals from ID/EX register, all the input control signals are set according to the ID/EX register.

The other input signals are set according to the execution of instruction seven(ORR X21, X19, X20). ALUResultIn = \$X19 OR \$X20 = xCEA4126C OR x1009AC83 = xDEADBEEF. FlagZeroIn = 0, ReadData2In = \$X20 = x1009AC83, PCNewaddrIn is dont care here since it's a R-type instruction.

All the output signals are set according to instruction six, and they would be assigned according to instruction seven at the rising edge of next cycle.

e MEM/WB Register

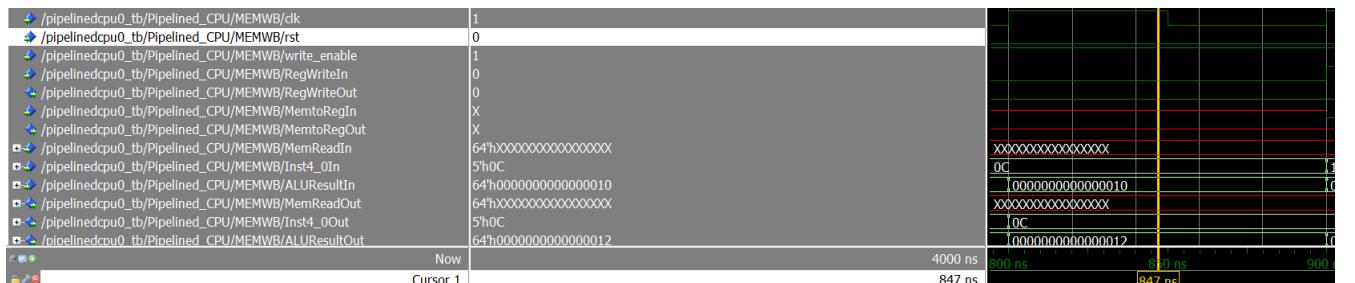


Figure 53: MEM/WB Signals of Cycle nine

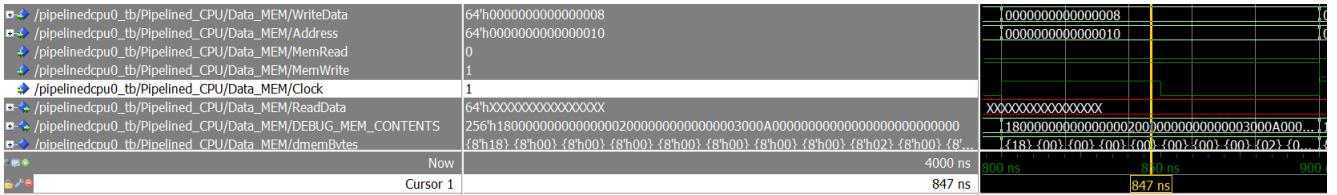


Figure 54: Data memory Signals of Cycle nine

MEM/WB register gets CPU control signals from EX/MEM register, all the input control signals are set according to the EX/MEM register.

The other input signals are set according to the memory operation of instruction six(STUR X12, [X12,8]). MemRead = 0 and MemWrite = 1, data memory will store WriteData = \$X12 = x8 into DMEM[Address=x10] at the next rising edge of 'clk'. The ALUResultIn = x10, came from EX/MEM.

All the output signals are set according to instruction five, and they would be assigned according to instruction six at the rising edge of next cycle.

11: Cycle Ten

a Debug signals

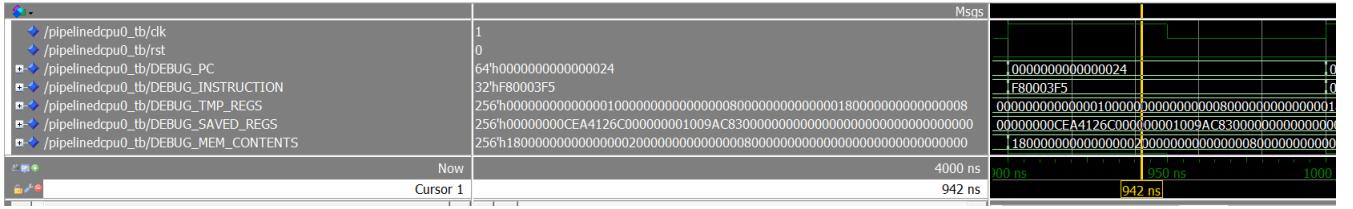


Figure 55: Debug Signals of Cycle ten

In cycle ten, we fetch instruction at PC address of x24, the instruction is 1111 1000 0000 0000 0000 0011 1111 0101.

Saved registers and temp registers are the same as the cycle nine.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x18, \$X12 = x08.

Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = x0, \$X22 = x0. Data memory DMEM[16-23] is changed from x03000A0000000000 to x0800000000000000 due to instruction six(STUR X12, [X12,8]). DMEM(0x0) = x18, DMEM(0x8) = x2, DMEM(0x24) = x0.

This instruction is correct, the expected value of \$X12 has been written back into register when we execute this instruction.

b IF/ID Register

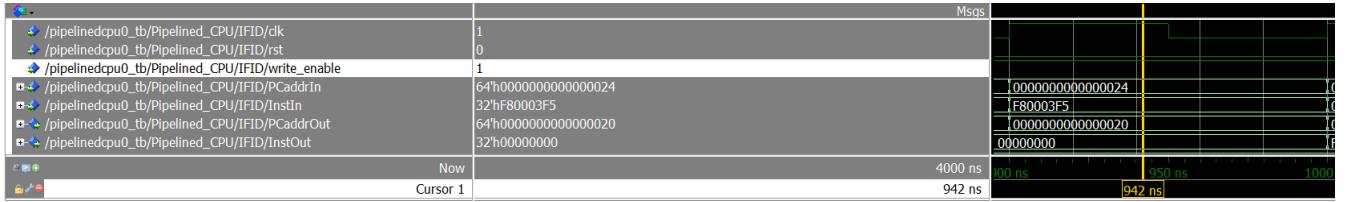


Figure 56: IF/ID Signals of Cycle ten



Figure 57: PCSrc Signal of Cycle ten

At 900 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x20 to x24. The output signals of IF/ID register are assigned with the input signals of cycle nine.

PCaddrOut = x20 \leftarrow x1C, InstOut = x00000000 \leftarrow x00000000(2 NOPs in a row).

c ID/EX Register

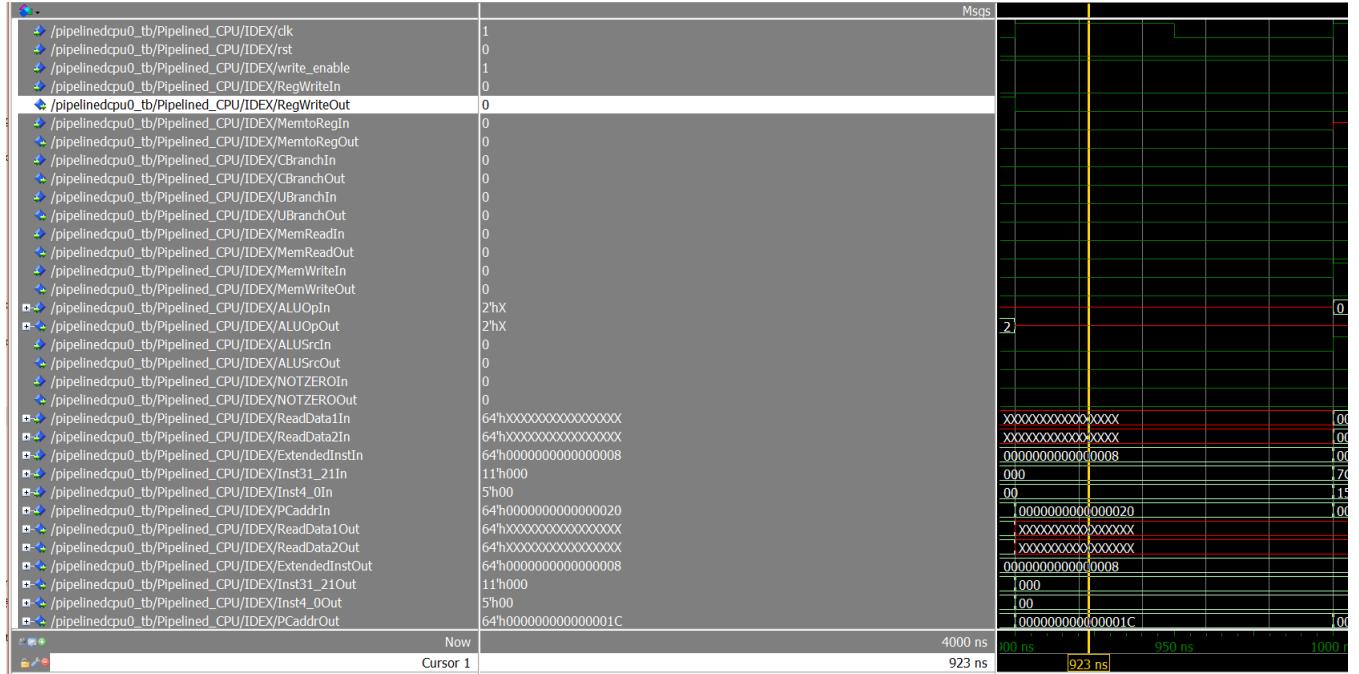


Figure 58: ID/EX Signals of Cycle ten

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction nine(NOP) decoding.

All the output signals are set according to instruction eight, and they would be assigned according to instruction nine at the rising edge of next cycle.

MEM/WB register gets CPU control signals from EX/MEM register, all the input control signals are set according to the EX/MEM register.

The other input signals are set according to the memory operation of instruction seven(ORR X21, X19, X20). MemRead = 0 and MemWrite = 0, data memory will do nothing at the next rising edge of 'clk'. The ALUResultIn = xDEADBEEF, came from EX/MEM.

All the output signals are set according to instruction six, and they would be assigned according to instruction seven at the rising edge of next cycle.

12: Cycle Eleven

a Debug signals

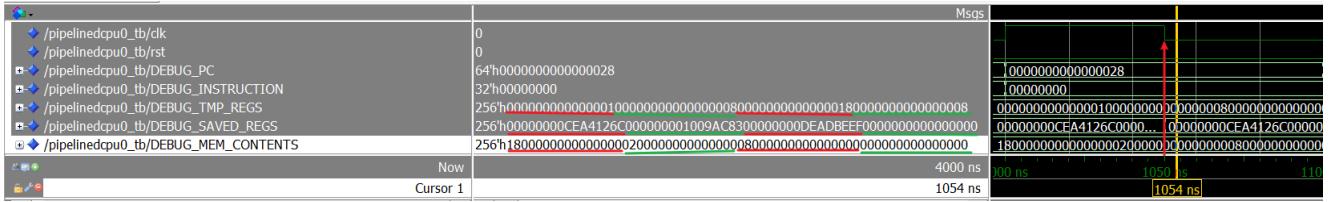


Figure 62: Debug Signals of Cycle eleven

In cycle eleven, we fetch instruction at PC address of x28, the instruction is NOP.

Temp registers and data memory are the same as the cycle ten.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x18, \$X12 = x08.

DMEM(0x0) = x18, DMEM(0x8) = x2, DMEM(0x16) = x8, DMEM(0x24) = x0.

Saved registers \$X21 are set to xDEADBEEF at 1050 ns, a falling edge of 'clk', according to instruction seven(ORR X21, X19, X20). \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X22 = x0.

This instruction is correct.

b IF/ID Register

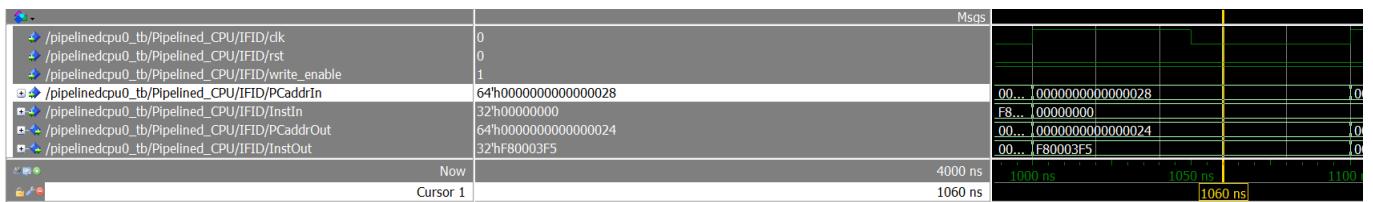


Figure 63: IF/ID Signals of Cycle eleven



Figure 64: PCSrc Signal of Cycle eleven

At 1000 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x24 to x28. The output signals of IF/ID register are assigned with the input signals of cycle ten.

PCaddrOut = x24 \leftarrow x20, InstOut = xF80003F5 \leftarrow x00000000.

c ID/EX Register

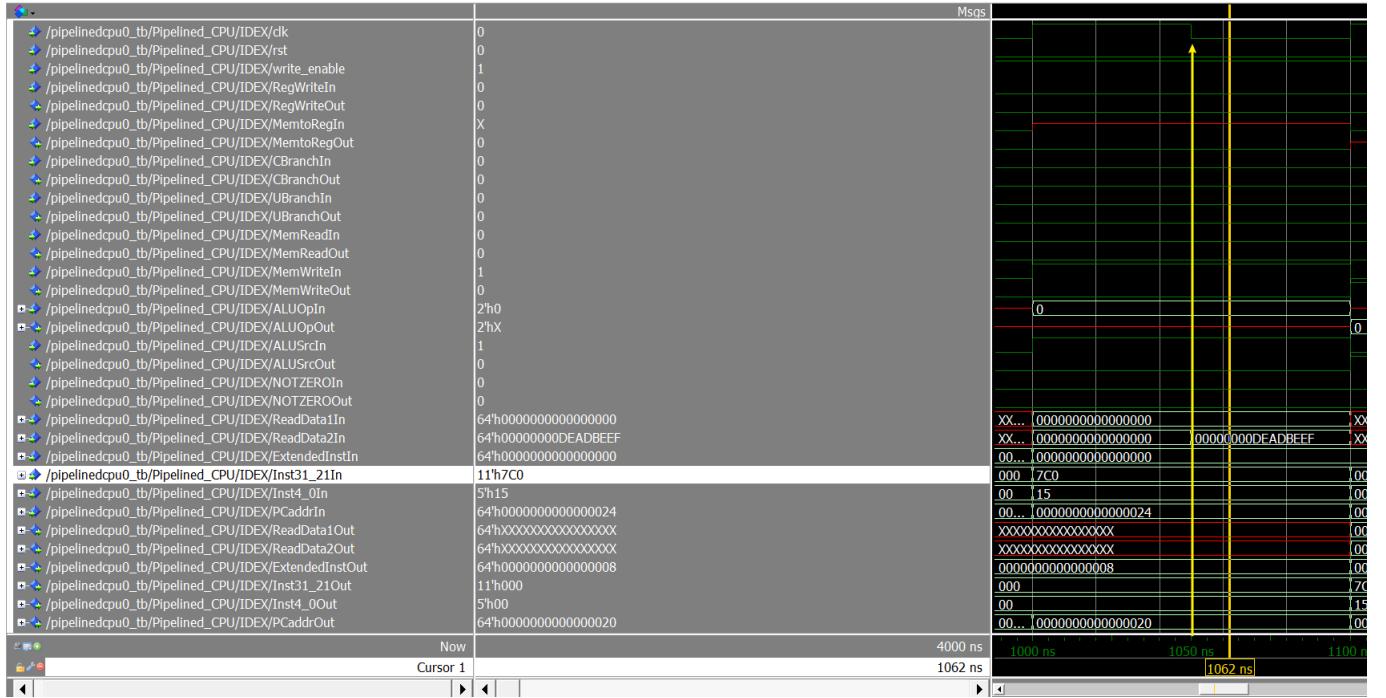


Figure 65: ID/EX Signals of Cycle eleven

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction ten(STUR X21, [XZR,0]) decoding. ReadData1In = \$XZR = x00000000, ReadData2In = \$X21 = xDEADBEEF, ExtendedInstIn = x0, PCaddrIn = x24.

All the output signals are set according to instruction nine, and they would be assigned according to instruction ten at the rising edge of next cycle.

d EX/MEM Register

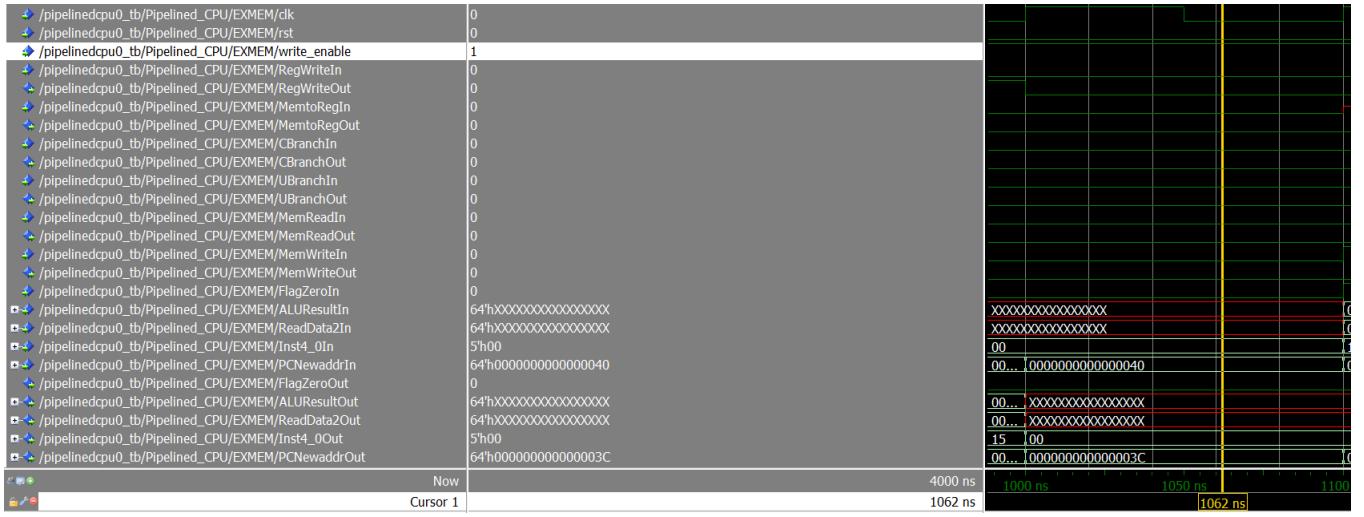


Figure 66: EX/MEM Signals of Cycle eleven

EX/MEM register gets CPU control signals from ID/EX register, all the input control signals are set according to the ID/EX register.

The other input signals are set according to the excution of instruction nine(NOP).

All the output signals are set according to instruction eight, and they would be assigned according to instruction nine at the rising edge of next cycle.

e MEM/WB Register



Figure 67: MEM/WB Signals of Cycle eleven

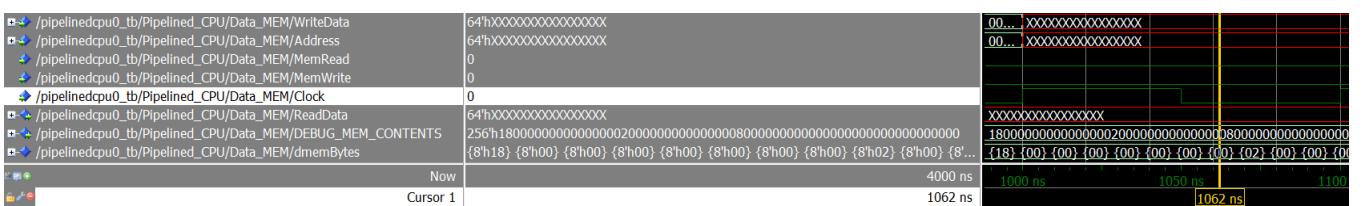


Figure 68: Data memory Signals of Cycle eleven

MEM/WB register gets CPU control signals from EX/MEM register, all the input control signals are set according to the EX/MEM register.

The other input signals are set according to the memory operation of instruction eight(NOP). All the output signals are set according to instruction seven, and they would be assigned according to instruction eight at the rising edge of next cycle.

13: Cycle Twelve

a Debug signals

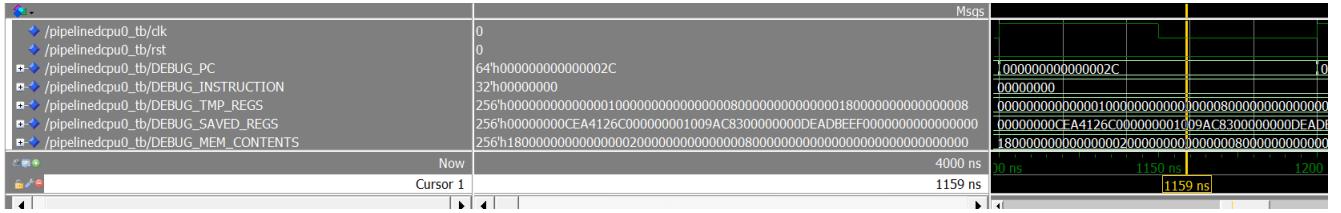


Figure 69: Debug Signals of Cycle twelve

In cycle twelve, we fetch instruction at PC address at x2C, the instruction is NOP. Temp registers, saved registers and data memory are the same as the cycle eleven. Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x18, \$X12 = x08. Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = xDEADBEEF, \$X22 = x0. DMEM(0x0) = x18, DMEM(0x8) = x2, DMEM(0x16) = x8, DMEM(0x24) = x0.

b IF/ID Register



Figure 70: IF/ID Signals of Cycle twelve



Figure 71: PCSrc Signal of Cycle twelve

At 1100 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x28 to x2C. The output signals of IF/ID register are assigned with the input signals of cycle eleven.

PCaddrOut = x28 \leftarrow x24, InstOut = x00000000 \leftarrow xF80003F5.

c ID/EX Register

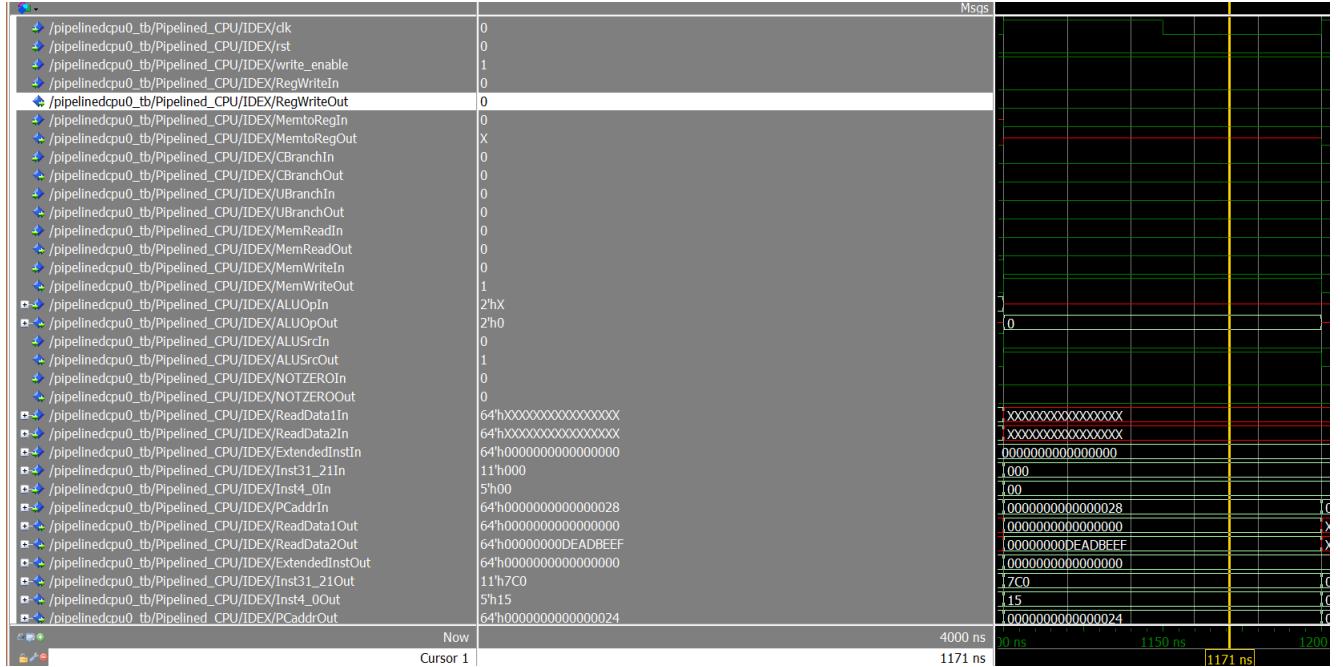


Figure 72: ID/EX Signals of Cycle twelve

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction eleven(NOP) decoding.

All the output signals are set according to instruction ten, and they would be assigned according to instruction eleven at the rising edge of next cycle.

d EX/MEM Register

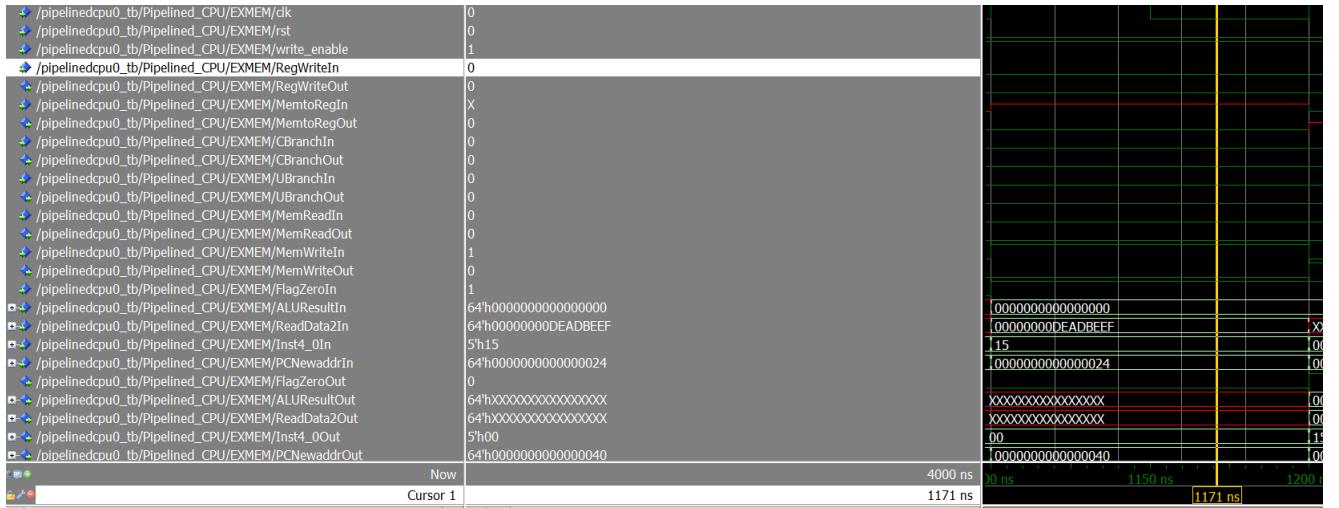


Figure 73: EX/MEM Signals of Cycle twelve

EX/MEM register gets CPU control signals from ID/EX register, all the input control signals are set according to the ID/EX register.

The other input signals are set according to the execution of instruction ten(STUR X21, [XZR,0]). ALUResultIn = \$XZR + x0 = x0 + x0 = x0, ReadData2In = \$X21 = xDEADBEEF, PCNewaddrIn is don't care here since it's a D-type instruction.

All the output signals are set according to instruction nine, and they would be assigned according to instruction ten at the rising edge of next cycle.

e MEM/WB Register



Figure 74: MEM/WB Signals of Cycle twelve

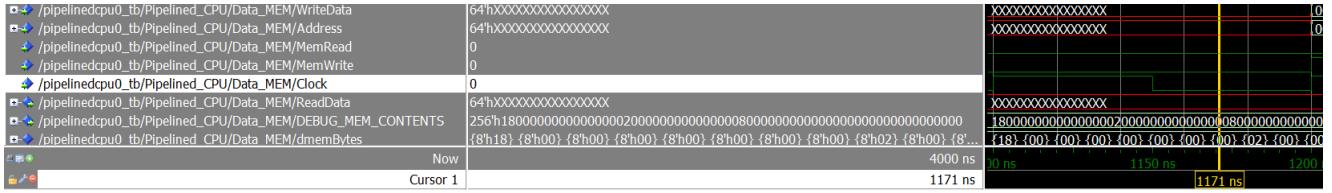


Figure 75: Data memory Signals of Cycle twelve

MEM/WB register gets CPU control signals from EX/MEM register, all the input control signals are set according to the EX/MEM register.

The other input signals are set according to the memory operation of instruction nine(NOP). All the output signals are set according to instruction eight, and they would be assigned according to instruction nine at the rising edge of next cycle.

14: Cycle Thirteen

a Debug signals

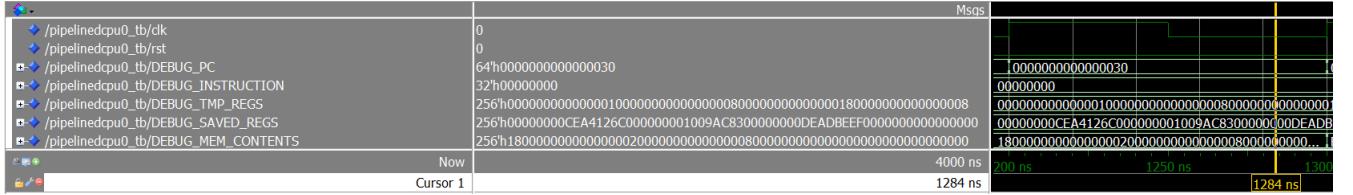


Figure 76: Debug Signals of Cycle thirteen

In cycle thirteen, we fetch instruction at PC address of x30, the instruction is NOP.

Temp registers, saved registers and data memory are the same as the cycle twelve.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x18, \$X12 = x08.

Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = xDEADBEEF, \$X22 = x0.

DMEM(0x0) = x18, DMEM(0x8) = x2, DMEM(0x16) = x8, DMEM(0x24) = x0.

b IF/ID Register

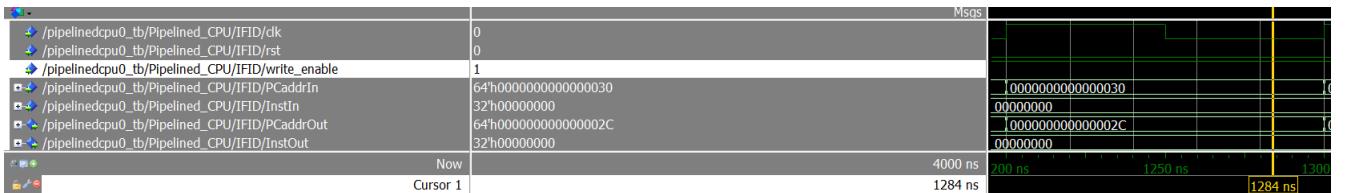


Figure 77: IF/ID Signals of Cycle thirteen



Figure 78: PCSrc Signal of Cycle thirteen

At 1200 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x2C to x30. The output signals of IF/ID register are assigned with the input signals of cycle twelve.

PCaddrOut = x2C \leftarrow x28, InstOut = x00000000 \leftarrow x00000000.

c ID/EX Register

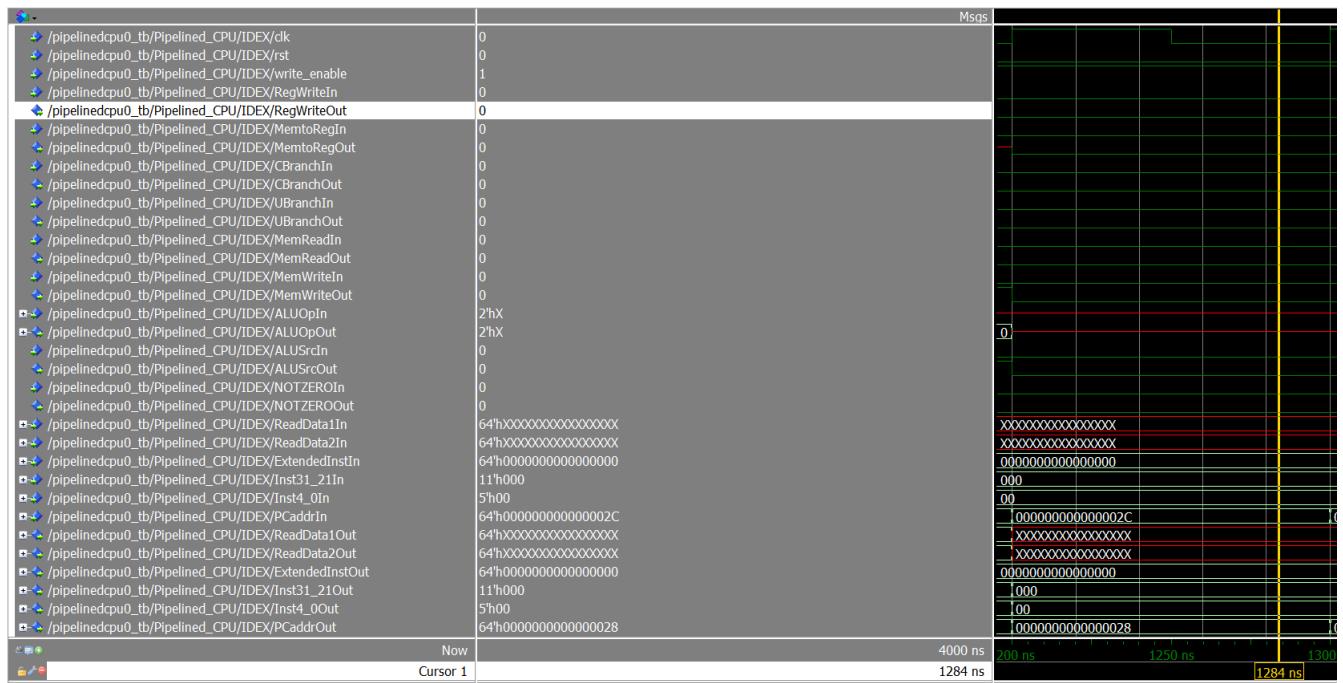


Figure 79: ID/EX Signals of Cycle thirteen

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction twelve(NOP) decoding.

All the output signals are set according to instruction eleven, and they would be assigned according to instruction twelve at the rising edge of next cycle.

d EX/MEM Register

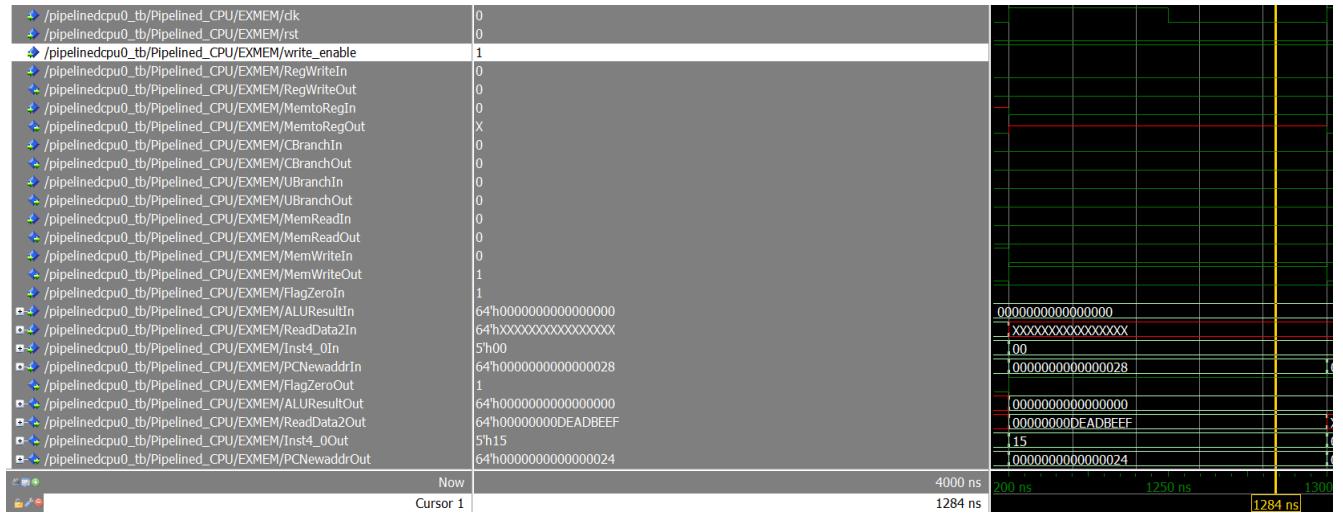


Figure 80: EX/MEM Signals of Cycle thirteen

EX/MEM register gets CPU control signals from ID/EX register, all the input control signals are set according to the ID/EX register.

The other input signals are set according to the execution of instruction eleven(NOP).

All the output signals are set according to instruction ten, and they would be assigned according to instruction eleven at the rising edge of next cycle.

e MEM/WB Register

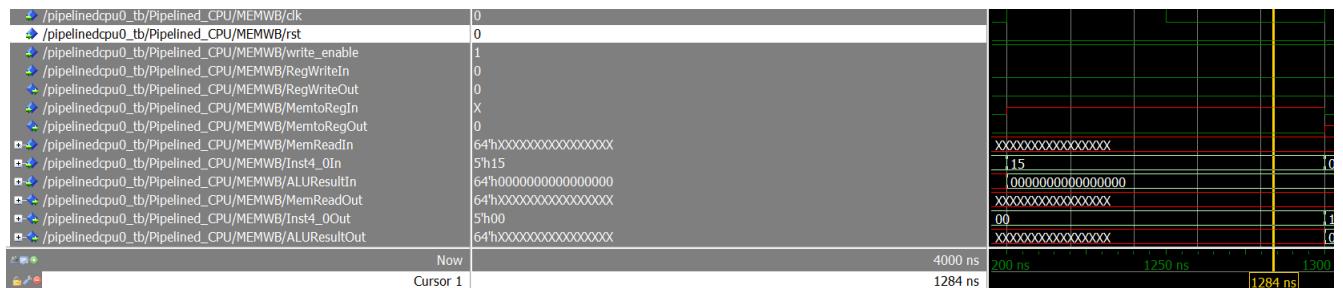


Figure 81: MEM/WB Signals of Cycle thirteen

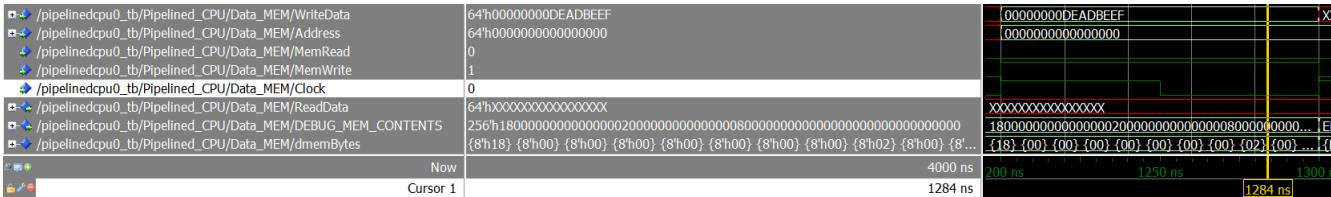


Figure 82: Data memory Signals of Cycle thirteen

MEM/WB register gets CPU control signals from EX/MEM register, all the input control signals are set according to the EX/MEM register.

The other input signals are set according to the memory operation of instruction ten(STUR X21, [XZR,0]). MemRead = 0 and MemWrite = 1, data memory will write WriteData = xDEADBEEF into DMEM[0-7] at the next rising edge of 'clk'. The ALUResultIn = x0, came from EX/MEM.

All the output signals are set according to instruction nine, and they would be assigned according to instruction ten at the rising edge of next cycle.

15: Cycle Fourteen

a Debug signals

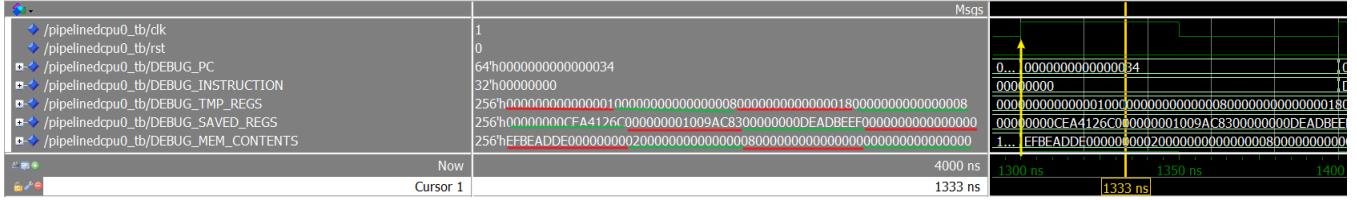


Figure 83: Debug Signals of Cycle fourteen

In cycle thirteen, we fetch instruction at PC address of x34, the instruction is NOP.
Temp registers, saved registers are the same as the cycle thirteen.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x18, \$X12 = x08.

Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = xDEADBEEF, \$X22 = x0.

DMEM[0-7] is set to xDEADBEEF at 1300 ns, a rising edge of 'clk', according to instruction ten(STUR X21, [XZR,0]). DMEM(0x8) = x2, DMEM(0x16) = x8, DMEM(0x24) = x0.

This instruction works correctly because we have 2 NOP instructions before this instruction, and when we execute instruction ten the expected value of \$X21 has been written into registers.

b IF/ID Register

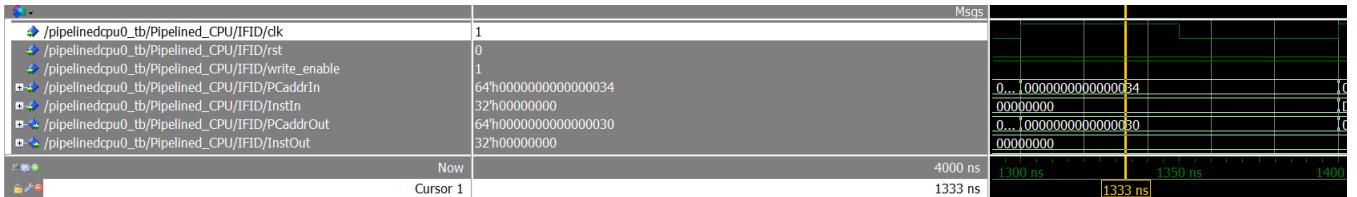


Figure 84: IF/ID Signals of Cycle fourteen



Figure 85: PCSrc Signal of Cycle fourteen

At 1300 ns, a rising edge of 'clk' signal, the PCaddrIn is changed from x30 to x34. The output signals of IF/ID register are assigned with the input signals of cycle thirteen.

PCaddrOut = x30 \leftarrow x2C, InstOut = x00000000 \leftarrow x00000000.

c ID/EX Register

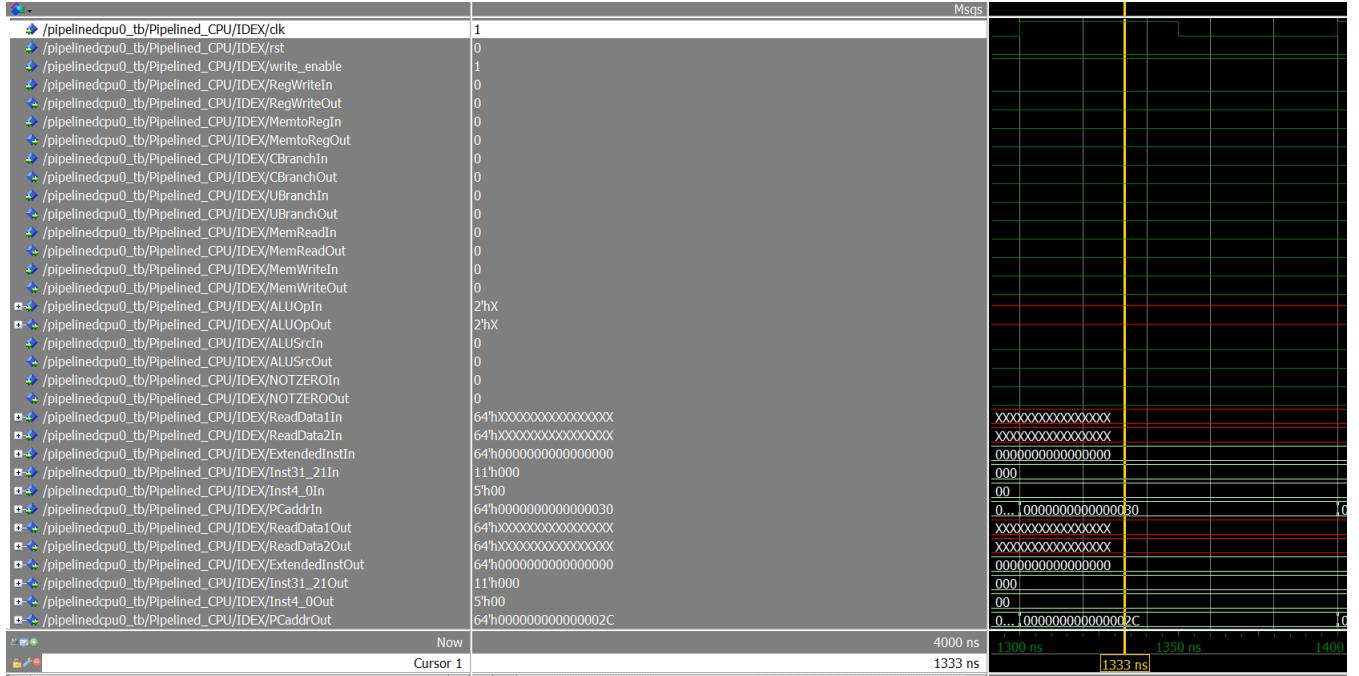


Figure 86: ID/EX Signals of Cycle fourteen

ID/EX register gets CPU control signals from CPUControl, all the input control signals are set according to the CPUControl.

The other input signals are set according to instruction thirteen(NOP) decoding.

All the output signals are set according to instruction twelve, and they would be assigned according to instruction thirteen at the rising edge of next cycle.

d EX/MEM Register

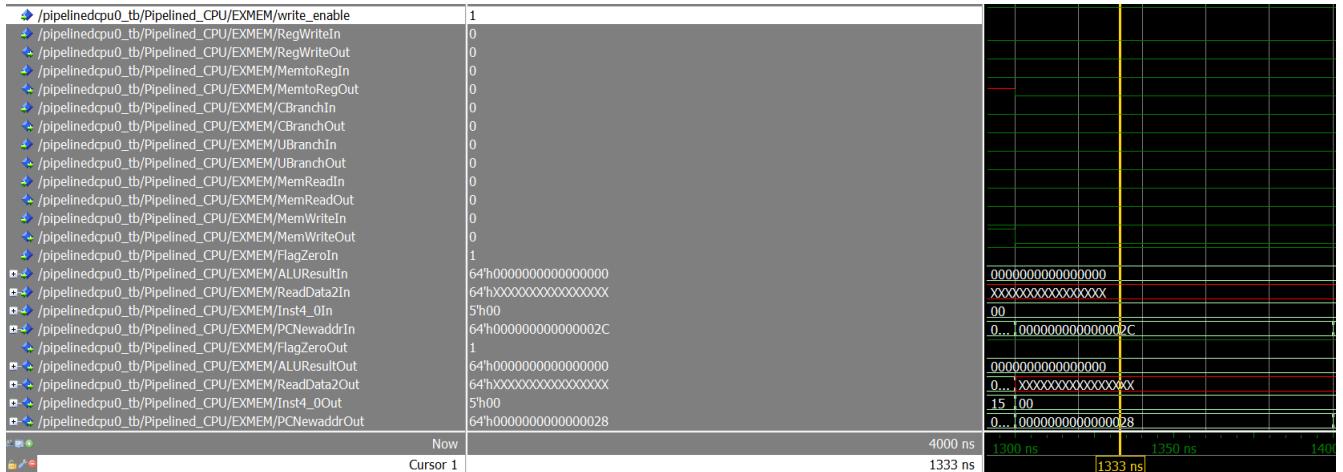


Figure 87: EX/MEM Signals of Cycle fourteen

EX/MEM register gets CPU control signals from ID/EX register, all the input control signals are set according to the ID/EX register.

The other input signals are set according to the execution of instruction twelve(NOP).

All the output signals are set according to instruction eleven, and they would be assigned according to instruction twelve at the rising edge of next cycle.

e MEM/WB Register

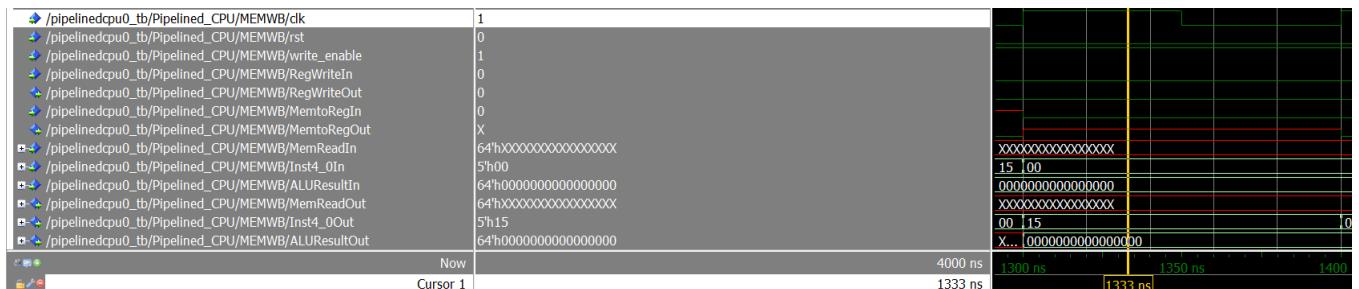


Figure 88: MEM/WB Signals of Cycle fourteen

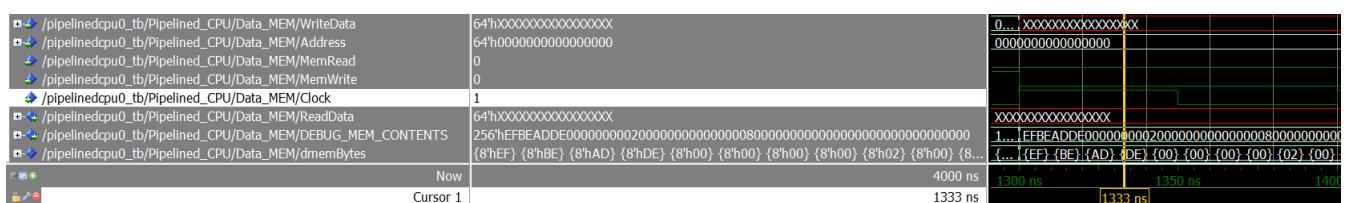


Figure 89: Data memory Signals of Cycle fourteen

MEM/WB register gets CPU control signals from EX/MEM register, all the input control signals are set according to the EX/MEM register.

The other input signals are set according to the memory operation of instruction eleven(NOP).

All the output signals are set according to instruction ten, and they would be assigned according to instruction eleven at the rising edge of next cycle.

16: Cycle Fifteen Nineteen

a Debug signals

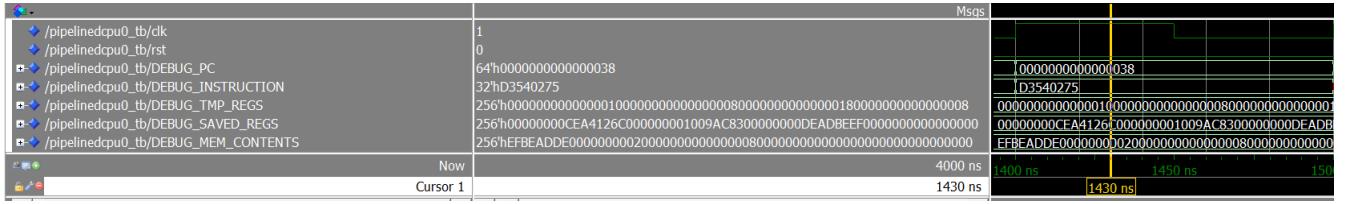


Figure 90: Debug Signals of Cycle fifteen

In cycle thirteen, we fetch instruction at PC address of x38, the instruction is 1101 0011 0101 0100 0000 0010 0111 0101(LSR X21, X19, #0 //Shift X19 right 0 bit and save into X21).

Temp registers, saved registers and data memory are the same as the cycle fourteen.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x18, \$X12 = x08.

Saved registers are set to \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X21 = xDEADBEEF, \$X22 = x0.

DMEM[0-7] = xDEADBEEF. DMEM(0x8) = x2, DMEM(0x16) = x8, DMEM(0x24) = x0.

After 4 cycles, the result of LSR operation is wrote into registers.

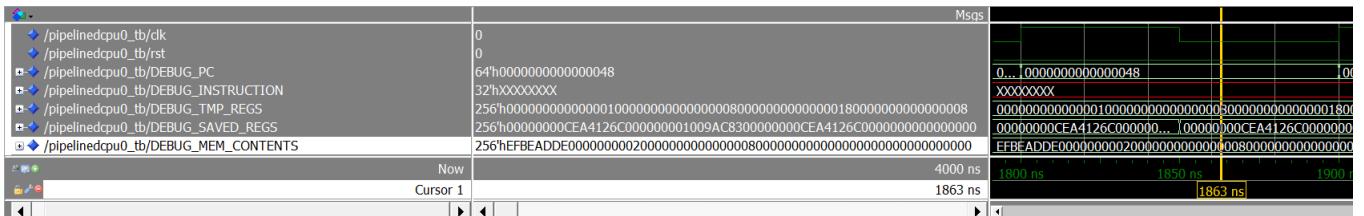


Figure 91: Debug Signals of Cycle nineteen

In cycle nineteen Temp registers and data memory are the same as the cycle fifteen.

Temp registers are set to \$X9 = x10, \$X10 = x08, \$X11 = x18, \$X12 = x08.

DMEM[0-7] = xDEADBEEF. DMEM(0x8) = x2, DMEM(0x16) = x8, DMEM(0x24) = x0.

Saved registers \$X21 are set to xCEA4126C at 1850 ns, a falling edge of 'clk', according to instruction fifteen(LSR X21, X19, #0), \$X19 = xCEA4126C, \$X20 = x1009AC83, \$X22 = x0.

This instruction is correct, because it's the last instruction.