Microprocessor Design

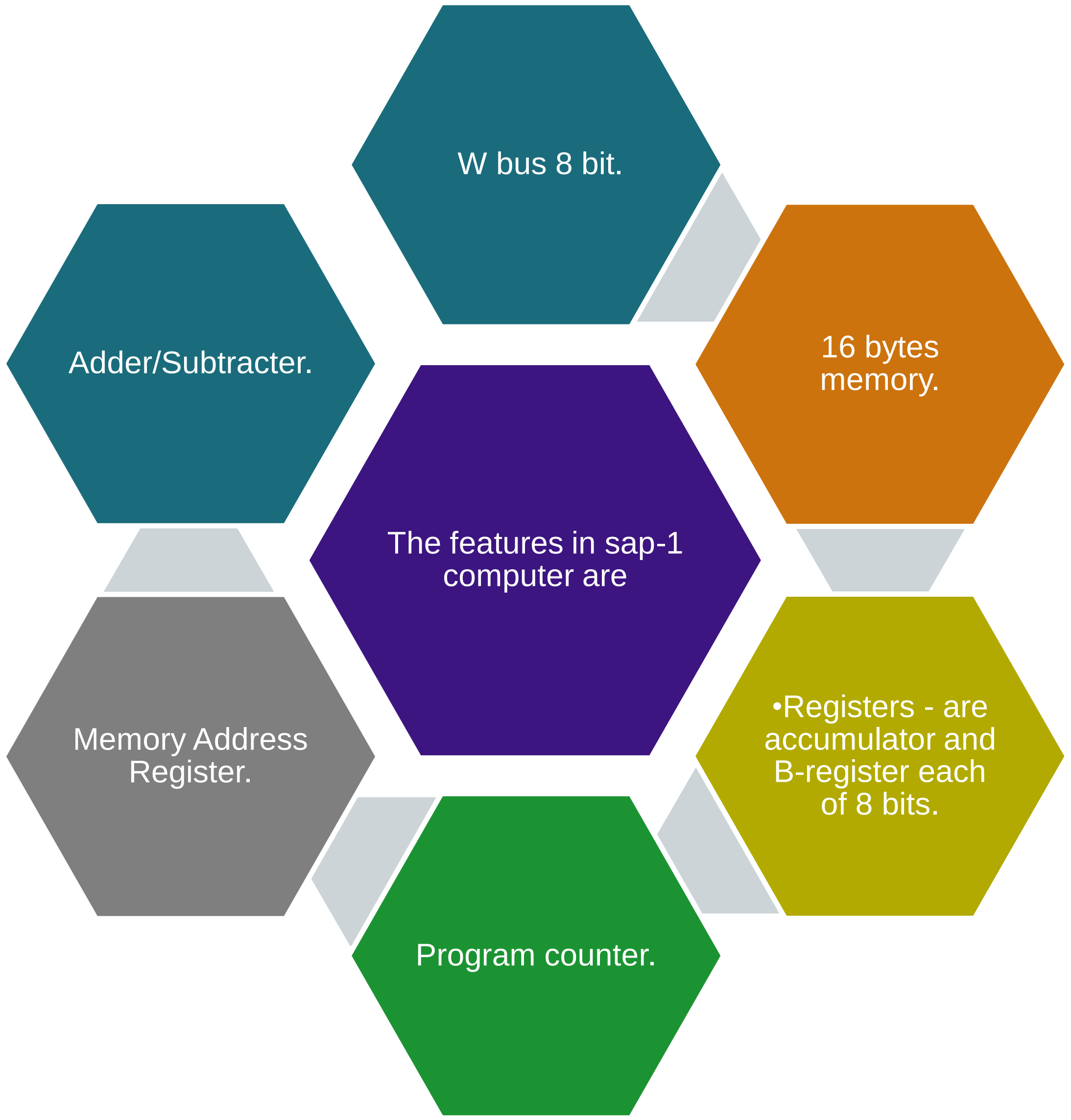
Prepared By

Abdelrahman Khaled EL-sayed

Supervised by
Dr. Eslam Tawfik

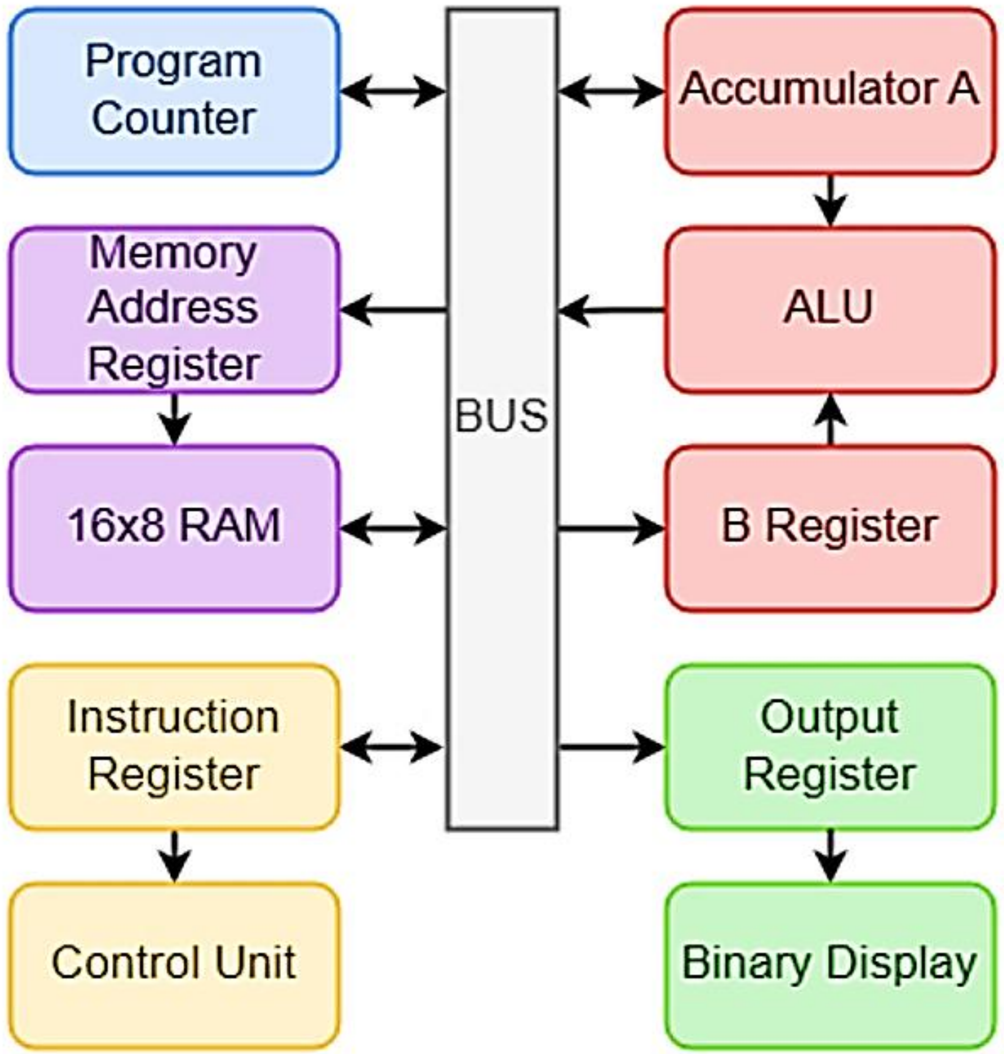
Introduction

Simple as Possible (SAP) computers in general were designed to introduce beginners to some of the crucial ideas behind computer operations. SAP computers are classified into stages, each stage more evolved and considering more advanced concepts in computer architecture than the previous.


W bus 8 bit.
16 bytes memory.
•Registers - are accumulator and B-register each of 8 bits.
Program counter.
Memory Address Register.
Adder/Subtracter.
The features in sap-1 computer are

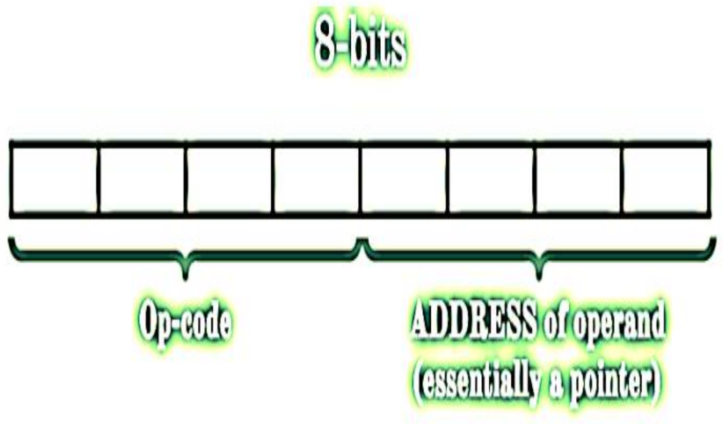
Architecture

All register outputs to the 8 bit W-bus are three states this allows orderly transfer of data.

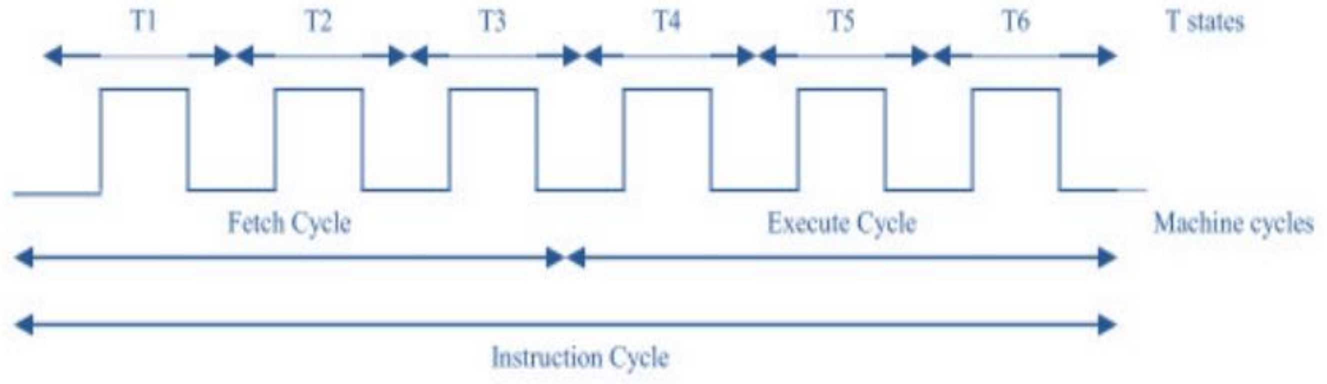


Instruction set

The instruction format of SAP-1 Computer is (XXXX) (XXXX) The first four bits make the op-code while the last four bits make the operand (address).



SAP1 has six T-states (three fetch and three execute cycles). All instructions require all the six T-states for execution. The unused T- state is marked as No Operation (NOP) cycle



The SAP-1 has four operations that it can perform:

- [0000] LDA X
  - Load the value at memory location X into Accumulator.
- [0001] ADD X
  - Load the value at memory location X into register b and Add the value at memory location X to Accumulator and store the sum in Accumulator.
- [0010] SUB X
  - Load the value at memory location X into register b and sub the value at memory location X to Accumulator and store the sum in Accumulator.
- [1110] OUT
  - Load accumulator data into output register

[1111] HLT

- Halt execution of the program.

FETCH CYCLES

SAP-1 instruction cycle:
3 clock cycles to fetch and decode
3 clock cycles to execute.

The first three states are:

1. address
2. increment
3. memory

Stage 1

- Put the PC onto the bus.
- Load that value into the MAR.

Stage 2

- Increment the PC.

Stage 3

- Put whatever is in memory at the MAR address onto the bus.
- Load it into the IR.

EXECUTE CYCLE

LDA

- Stage 4
  - Put the instruction operand onto the bus
- Load that value into the MAR
- Stage 5
  - Put whatever is in memory at the MAR address onto the bus
  - Load that value into accumulator
- Stage 6
  - NOP

ADD

- Stage 4
  - Put the instruction operand onto the bus.
  - Load that value into the MAR.
- Stage 5
  - Put whatever is in memory at the MAR address onto the bus.
  - Load that value into Register B.
- Stage 6
  - Put the value in the adder onto the bus.
  - Load that value into accumulator.

SUB

- Stage 4
  - Put the instruction operand onto the bus.
  - Load that value into the MAR.
- Stage 5
  - Put whatever is in memory at the MAR address onto the bus.
  - Load that value into Register B.
- Stage 6
  - Do subtraction rather than addition.
  - Put the value in the adder onto the bus.
  - Load that value into accumulator.

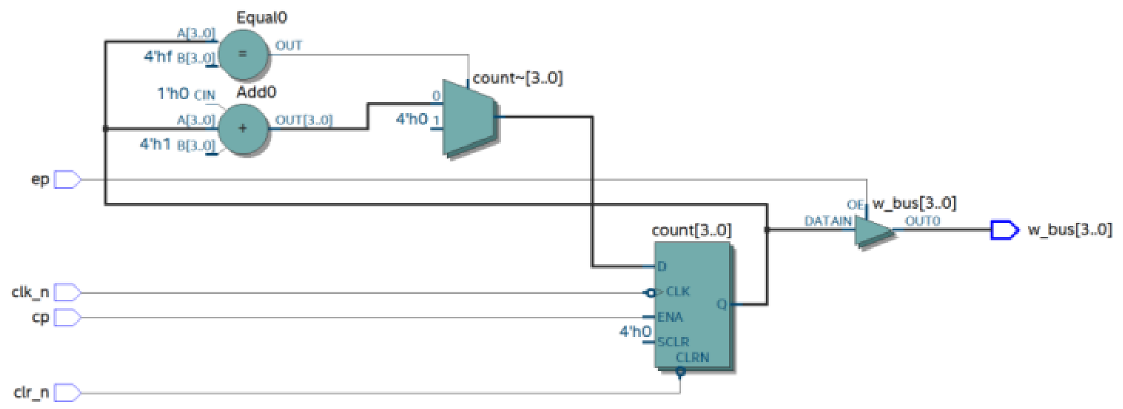
CENTER OF NANOELECTRONICS & DEVICES

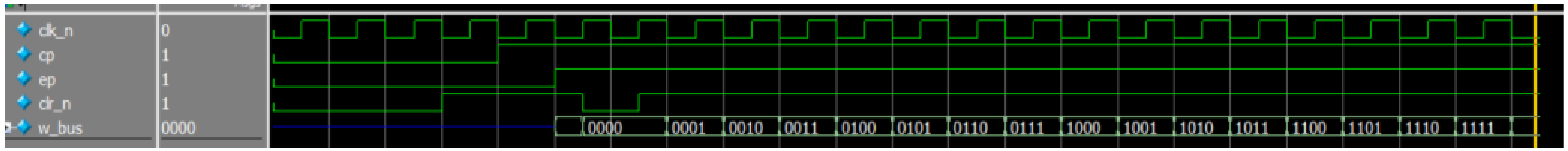
EXECUTE CYCLE

OUT	HLT
• Stage 4 ▪ put the accumulator in bus . ▪ load the bus value in output register ○ Stage 5 ▪ NOP ○ Stage 6 ○ NOP	• Stage 4 ▪ Halt the clock (hlt) ○ Stage 5 ▪ NOP ○ Stage 6 ○ NOP

program counter

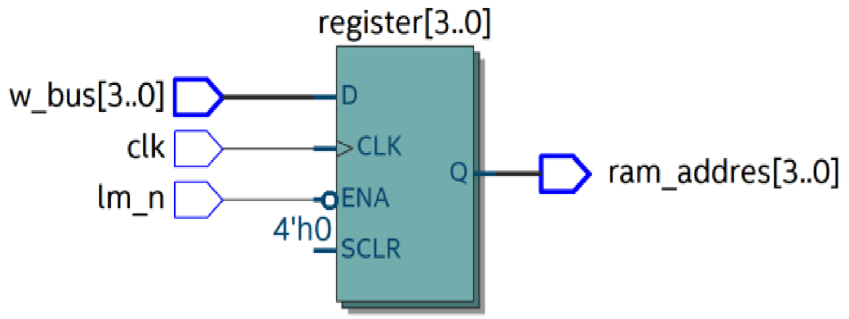
The program is stored at the beginning of the memory with the first instruction at binary address 0000, the second instruction at address 0001, the third at address 0010, and so on. sometimes called a pointer; it points to an address in memory where something important is being stored.

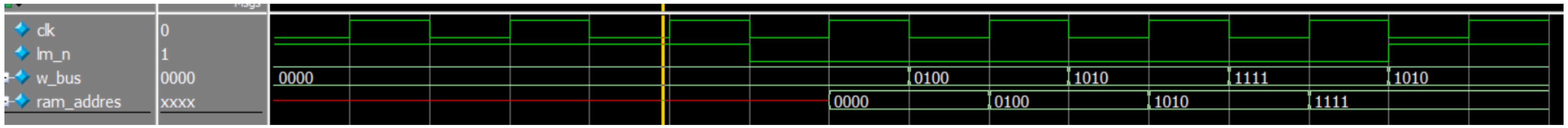




memory address register

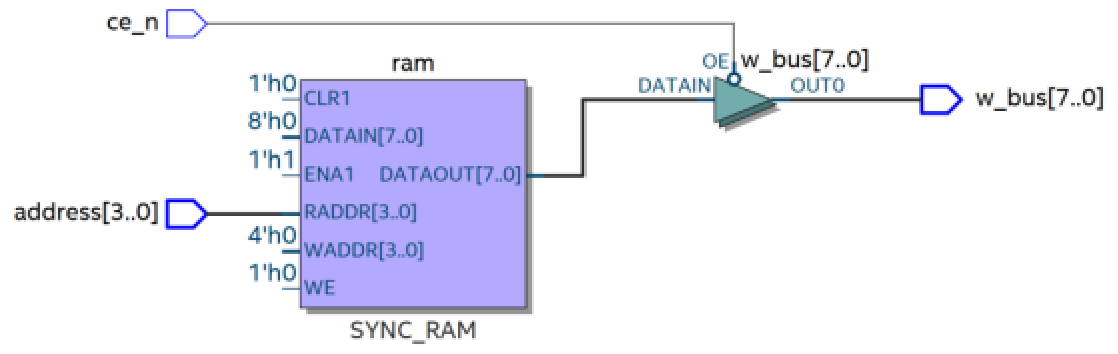
The MAR stores the 4-bit address of data or instruction which are placed in memory. When the SAP-1 is running, the 4-bit address is gotten from the Program Counter through the W-bus and then stored. This stored address is sent to the RAM where data or instructions are read from

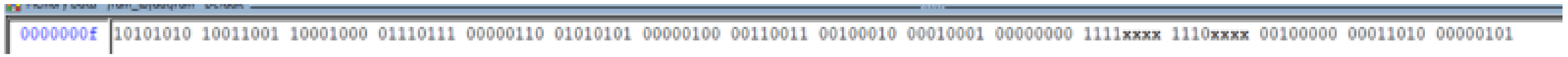




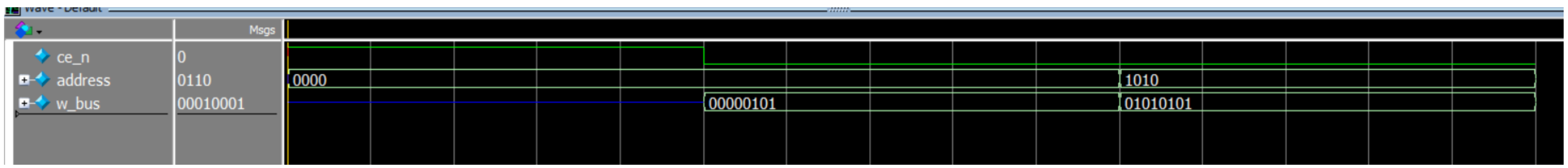
RAM

The SAP-1 makes use of a 16 x 8 RAM (16 memory locations each storing 8 bits of data). During a computer run, the RAM receives its 4-bit address from the MAR and read operation is performed. In this way the instruction or data word stored in the RAM is placed on the W bus for use in some other part of the computer.



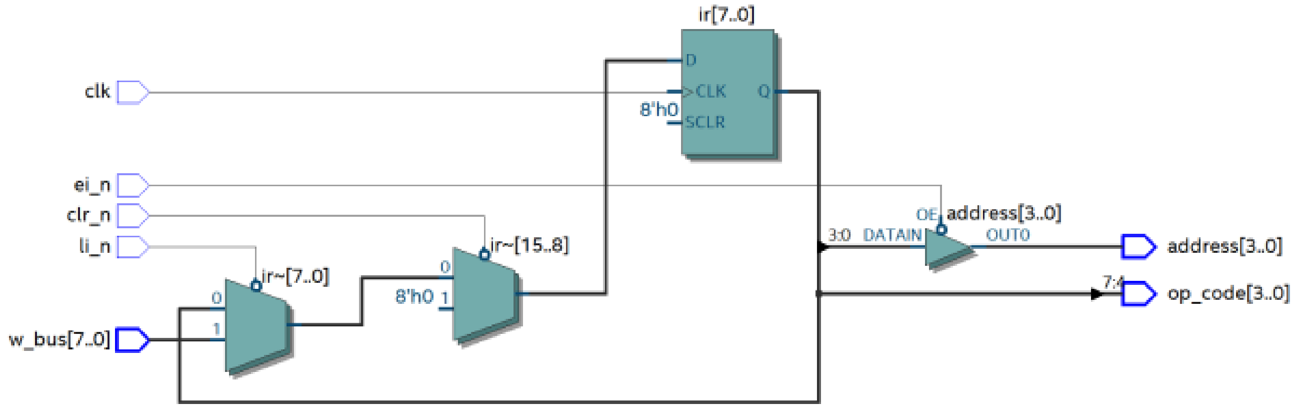


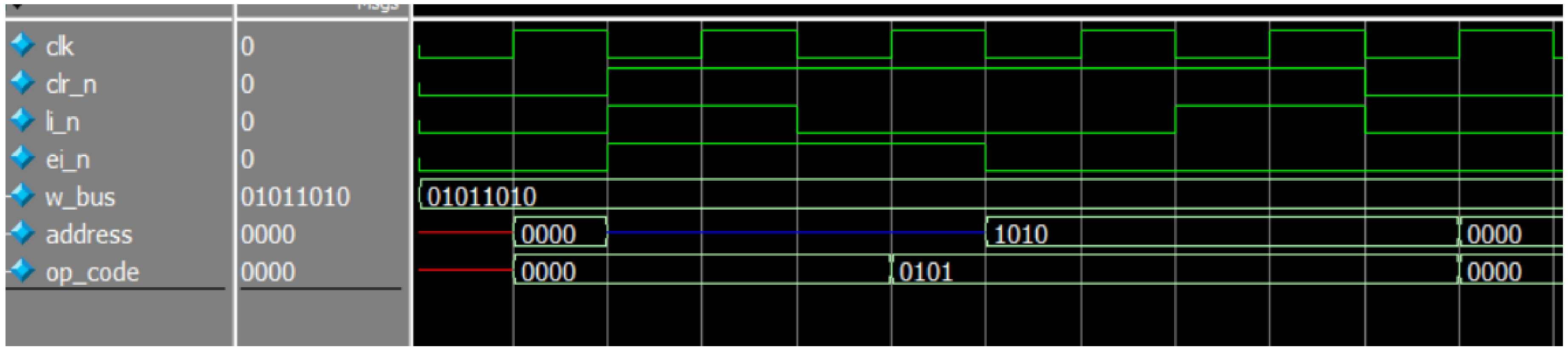
```
0000000f 10101010 10011001 10001000 01110111 00000110 01010101 00000100 00110011 00100010 00010001 00000000 1111xxxx 1110xxxx 00100000 00011010 00000101
```



instruction register

The instruction register is part of the control unit. The contents of the instruction register are split into two nibbles. The upper nibble is a two-state output that goes directly to the block labeled "Controller sequencer"

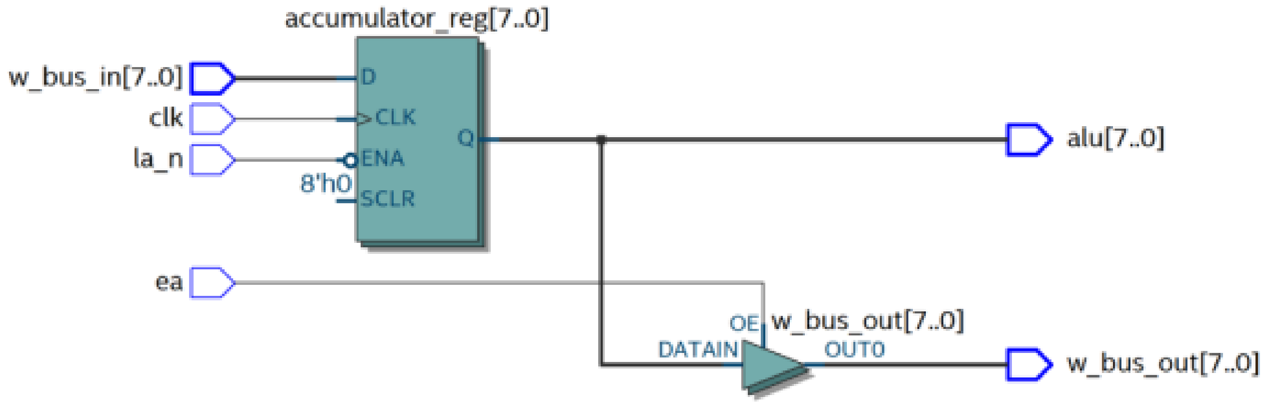


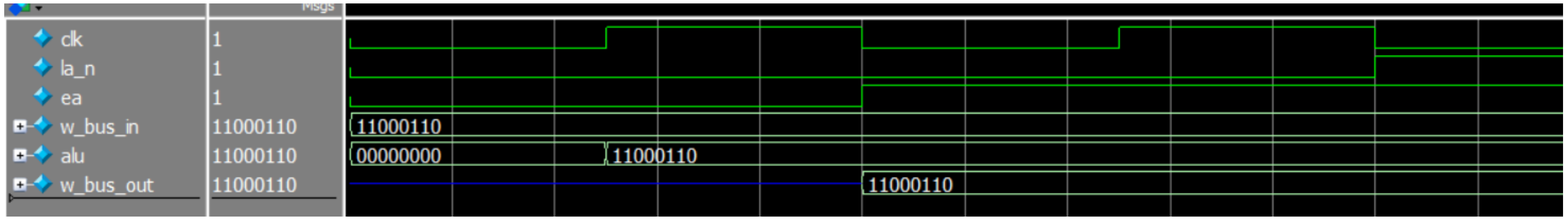


```
clk        0         
clr_n      0
li_n       0
ei_n       0
w_bus      01011010  01011010
address    0000      0000  1010  0000
op_code    0000      0000  0101  0000
```

Accumulator

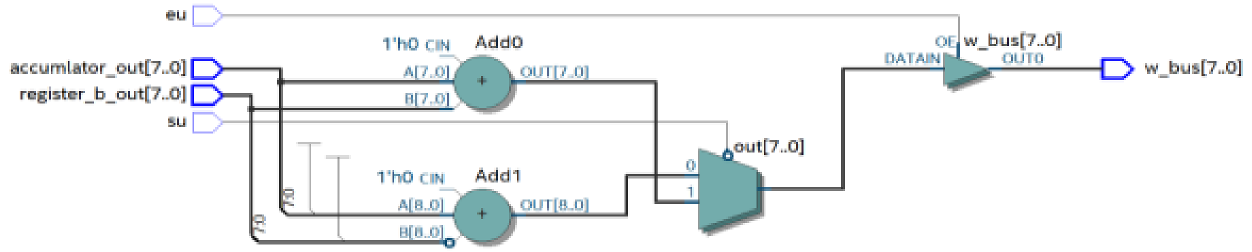
The accumulator is an 8-bit buffer register that stores intermediate answers during a computer run. The accumulator has two outputs. The two-state output goes directly to the adder-subtractor and the three-state output goes to the bus.

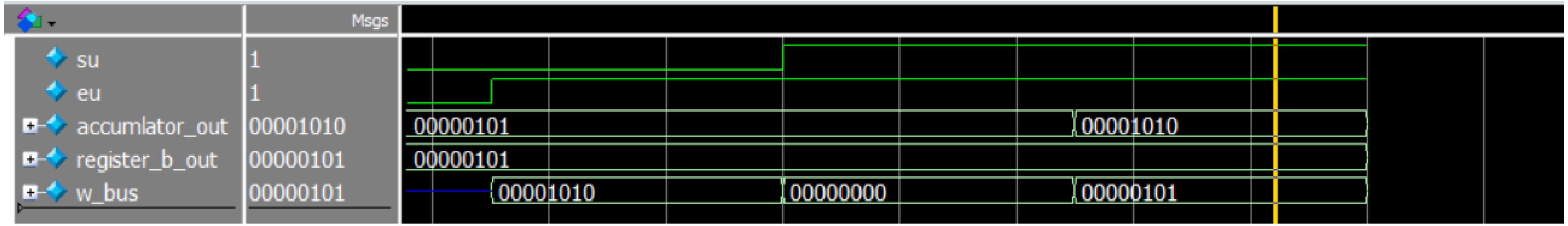




Adder / subtractor

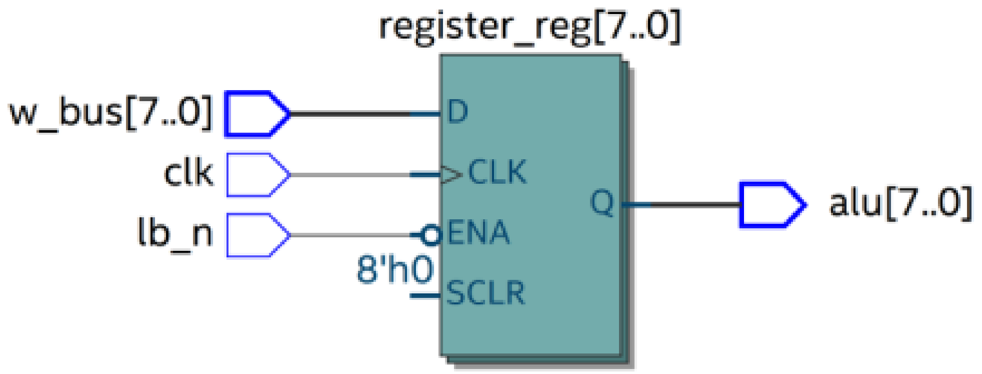
The adder-subtractor asynchronously adds to or subtracts a value from the the accumulator depending on the value of Su. It makes use of 2’s complement to achieve this When Su is low the output of the adder-subtractor is the sum of the values in the accumulator .

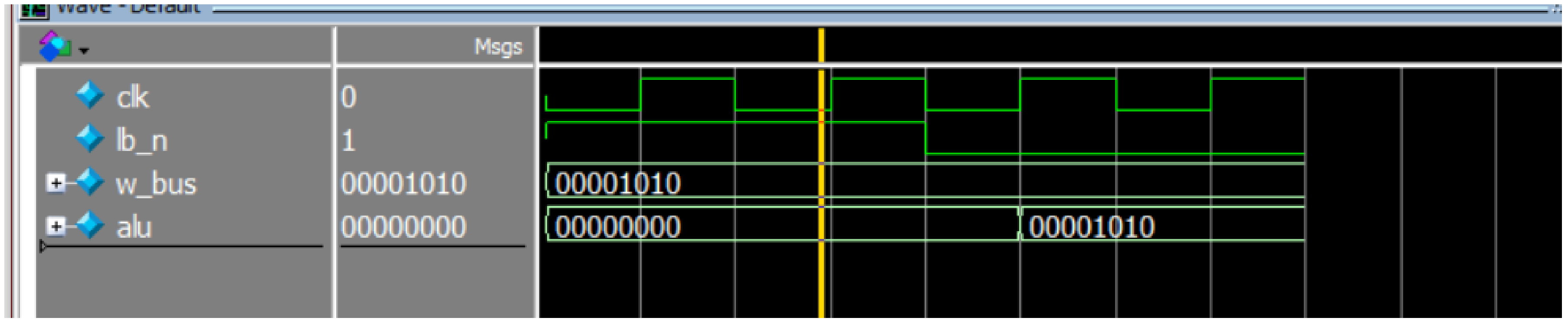



```
                  Msgs
su                1          
eu                1
accumlator_out    00001010    00000101                                    00001010
register_b_out    00000101    00000101
w_bus             00000101        00001010          00000000              00000101
```

register b

The B-register is a buffer register used in performing arithmetic operations. It supplies the number to be added or subtracted from the contents of the accumulator to the adder/subtractor.

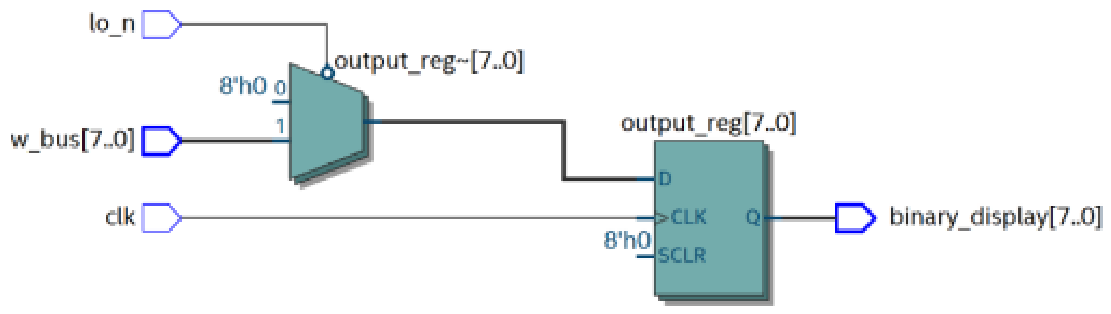


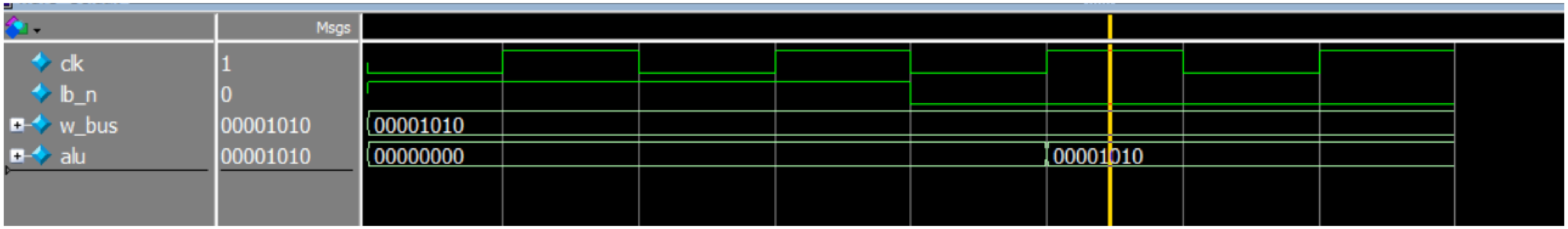


```
clk     0
lb_n    1
w_bus   00001010    00001010
alu     00000000    00000000    00001010
```

output register

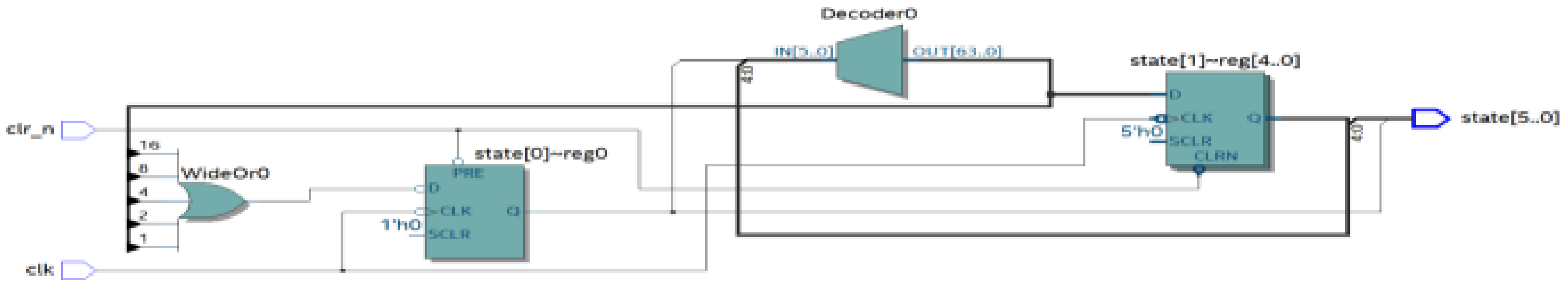
The output register gets and stores the value stored in the accumulator usually after the performance of an arithmetic operation. The answer that is stored in the accumulator is loaded into the output register through the W bus. This is done in the next positive clock edge when Ea is high and Lo is low. The processed data can now be displayed to the outside world.

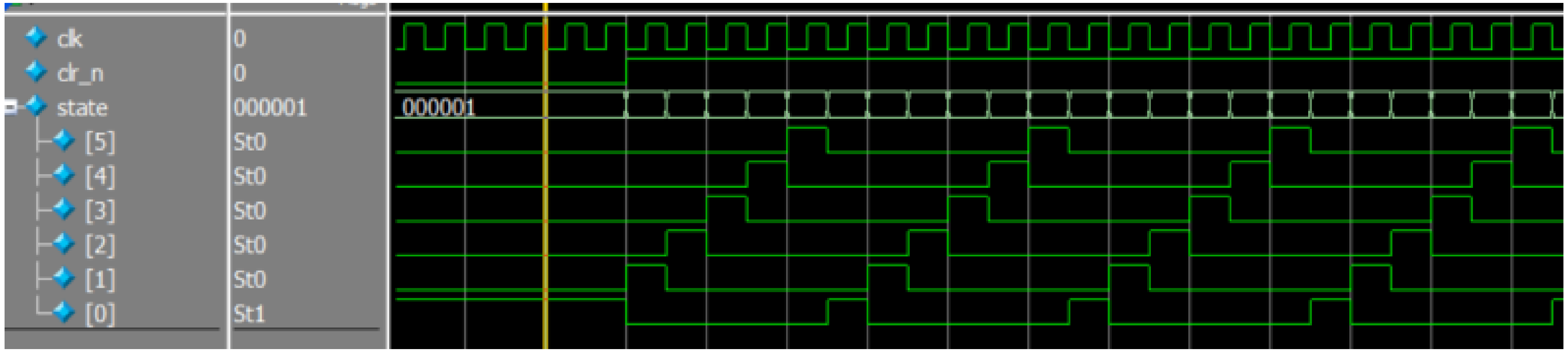




Ring counter

. In controller we must use ring counter because the sap use six clock cycles and counter will use to determine the state





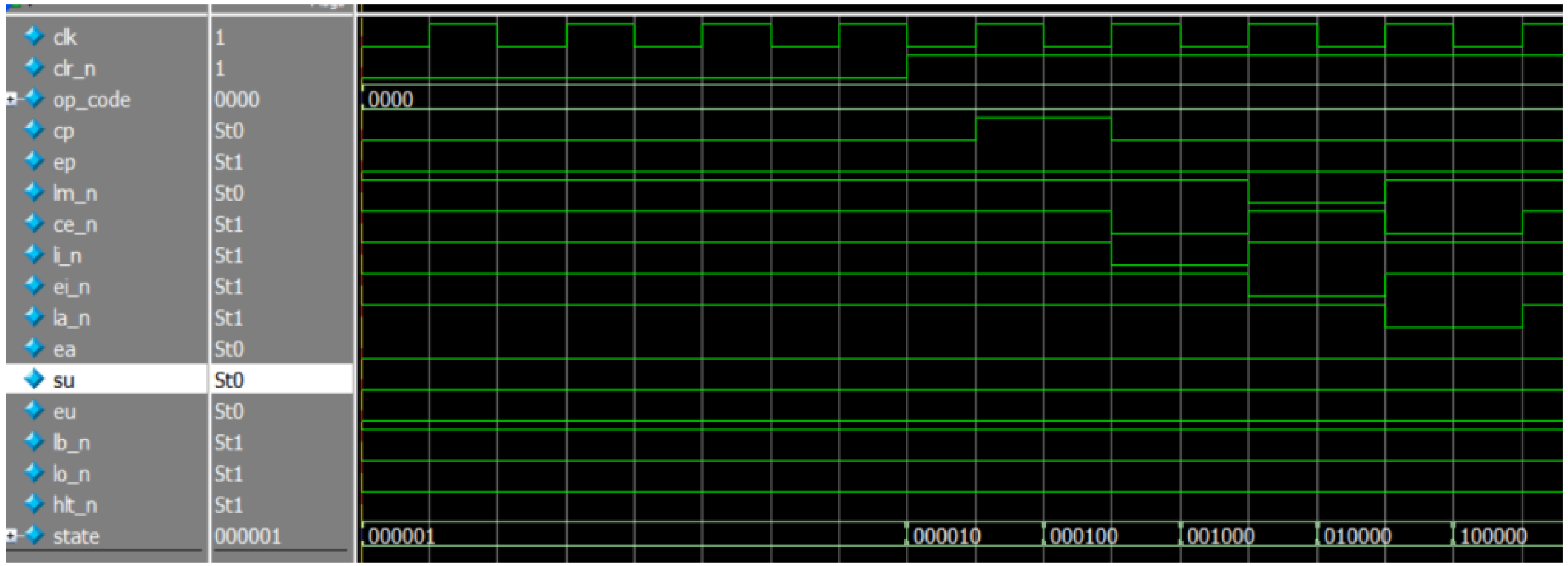
controller

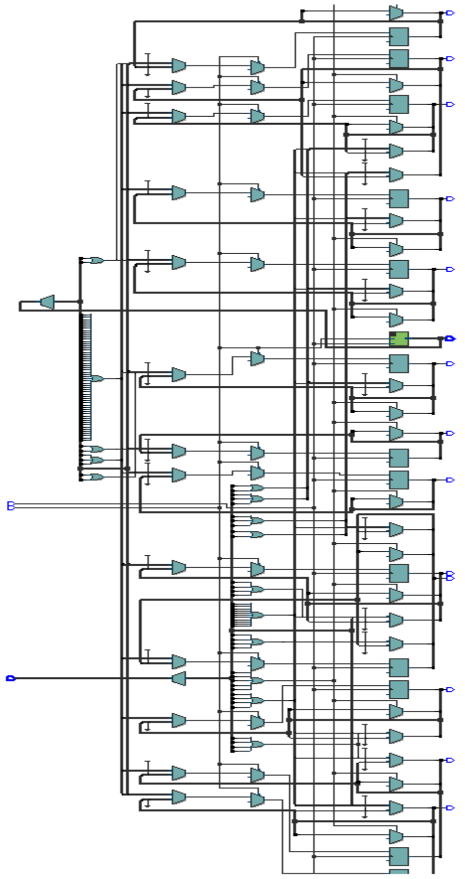
The controller-sequencer sends out signals that control the computer and makes sure things happen only when they are supposed to. The 12 bit output signals from controller sequencer is called the control word which determines how the registers will react to the next positive clock edge. The controller control other block by FSM.

Those control signals are:

- Cp = Increment PC
- Ep = Enable PC ouput to WBUS (1 = Enable)
- lm_n = load in memory address register(0 = load)
- ce_n = Enable output of RAM data to WBUS (0 = enable)
- Li_n = Load Instruction Register from WBUS (0 = load)
- Ei_n = Enable ouput of address from Instruction Register to lower 4 bits of WBUS
- La_n = Load data into the accumulator from WBUS (0 = load)
- Ea = put ouput of accumulator to WBUS (1 = enable)
- Su = ALU operation (0 = Add, 1 = Subtract)
- Eu = Enable output of ALU to WBUS (1 = enable)
- Lb_n = Load data into Register B from WBUS (0 = load)
- Lo_n = Load data into Output Register (0 = load)
- enio = Enable output from Input Register (1 = enable)
- HLT_n = Inverse Halt operation. (0 = HALT)

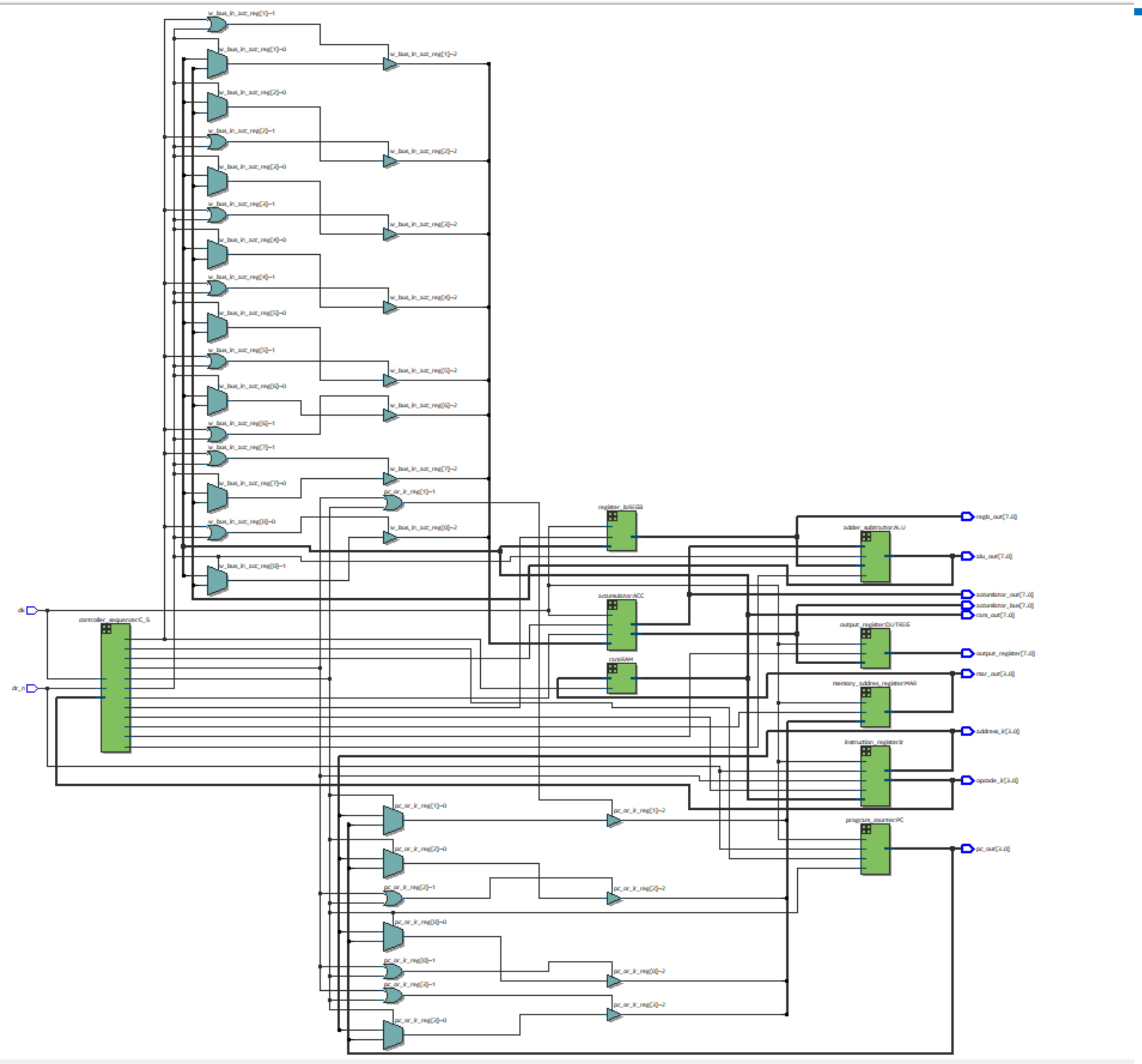
Controller RTL and simulation



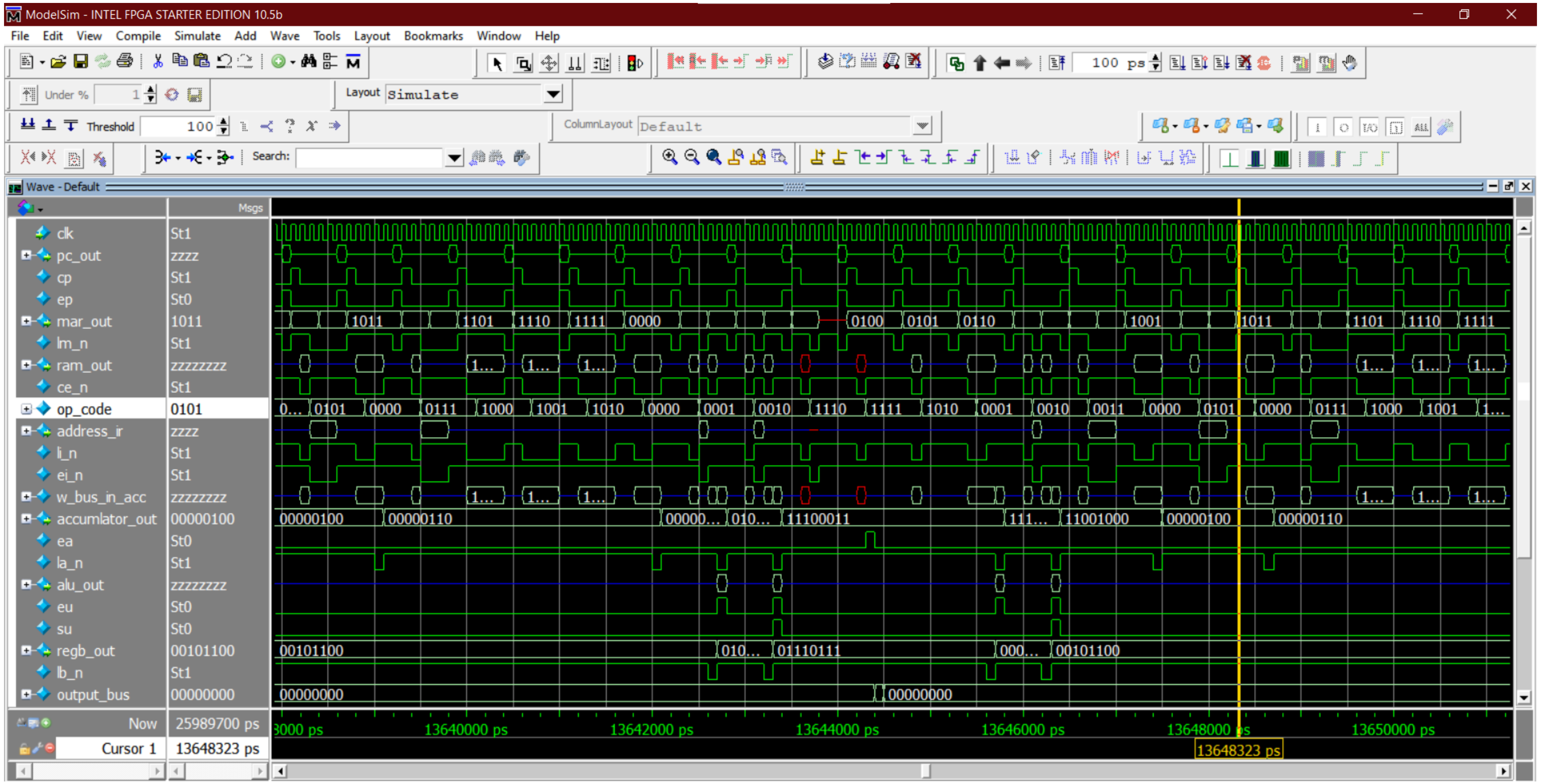


results

The figure shows the RTL netlist results for the sap1



simulation results


ModelSim - INTEL FPGA STARTER EDITION 10.5b
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Wave - Default

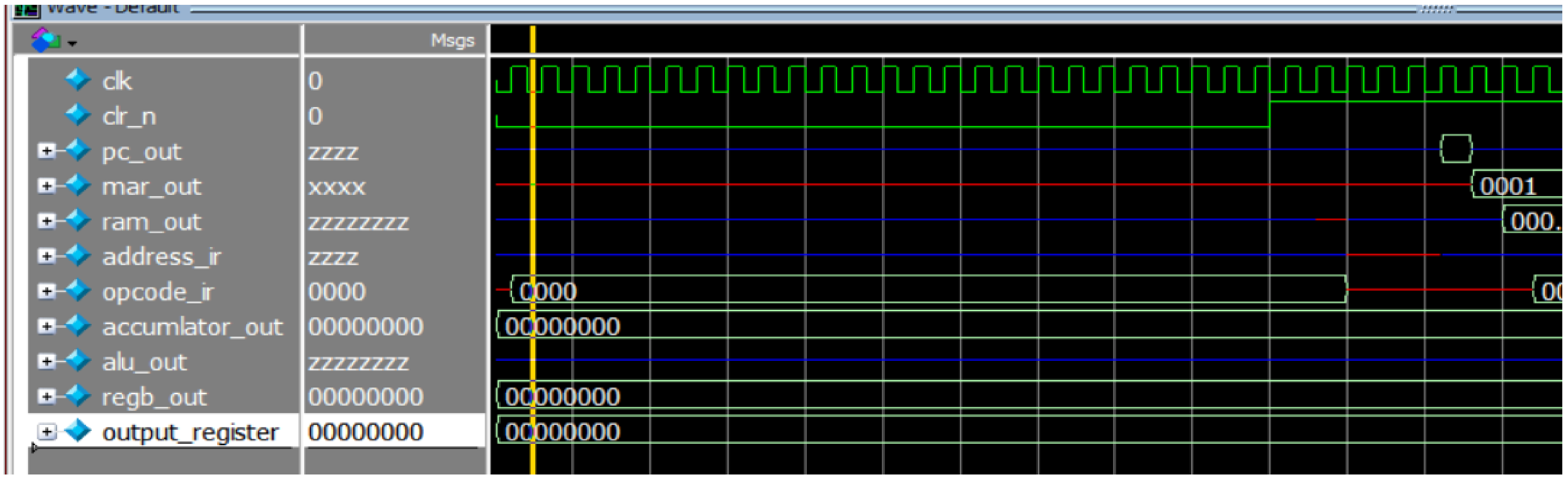
Layout Simulate
ColumnLayout Default
Threshold 100
Under % 1
100 ps

Signal	Msgs
clk	St1
pc_out	zzzz
cp	St1
ep	St0
mar_out	1011
lm_n	St1
ram_out	zzzzzzzz
ce_n	St1
op_code	0101
address_ir	zzzz
li_n	St1
ei_n	St1
w_bus_in_acc	zzzzzzzz
accumulator_out	00000100
ea	St0
la_n	St1
alu_out	zzzzzzzz
eu	St0
su	St0
regb_out	00101100
lb_n	St1
output_bus	00000000

Now 25989700 ps
Cursor 1 13648323 ps

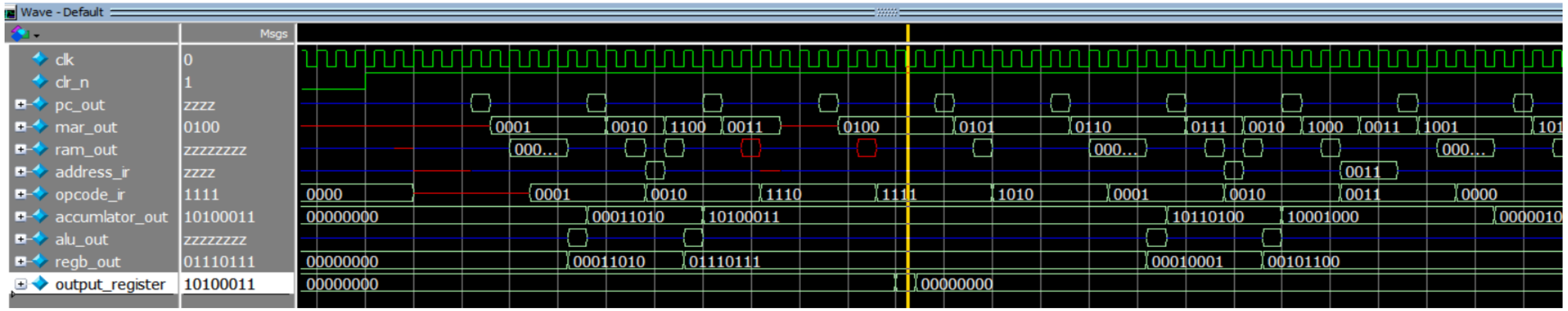
simulation results

The figure below shows the RTL simulation results for the sap1 when clear is low.



	Msgs
clk	0
clr_n	0
pc_out	zzzz
mar_out	xxxx
ram_out	zzzzzzzz
address_ir	zzzz
opcode_ir	0000
accumlator_out	00000000
alu_out	zzzzzzzz
regb_out	00000000
output_register	00000000

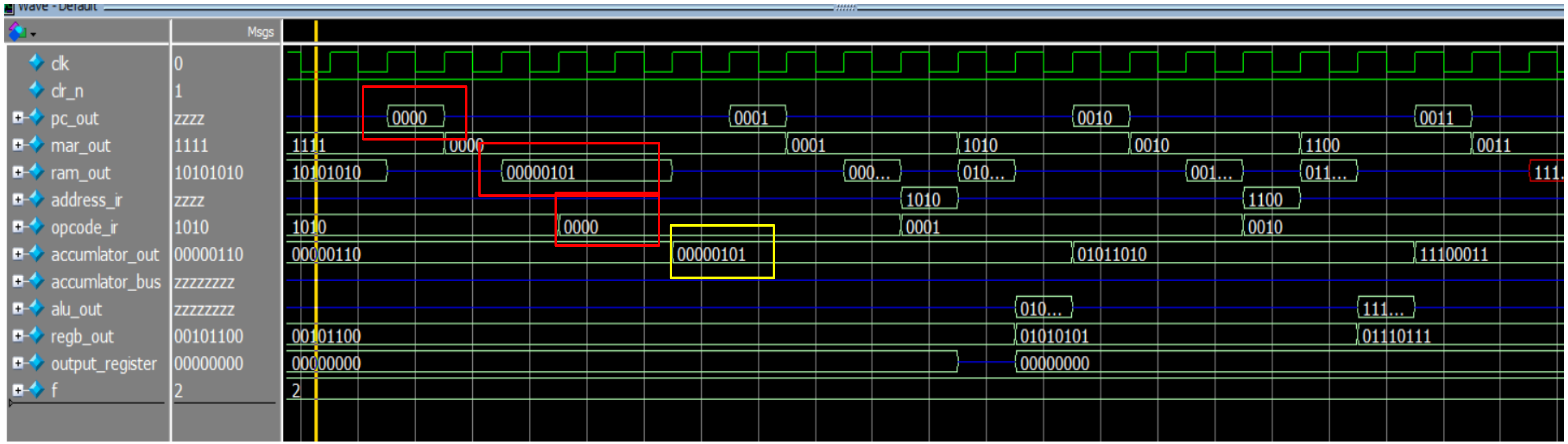
And when clear is high. Output is 10100011 which is in accumulator and appear at opcode 1110 .



	Msgs
clk	0
clr_n	1
pc_out	zzzz
mar_out	0100
ram_out	zzzzzzzz
address_ir	zzzz
opcode_ir	1111
accumlator_out	10100011
alu_out	zzzzzzzz
regb_out	01110111
output_register	10100011

simulation results

When pc =0000 which pointer to the first location in ram and the out of ram is 0000101, The instruction register divide it to op code=0000 which indicate to load the data from ram to accumulator.

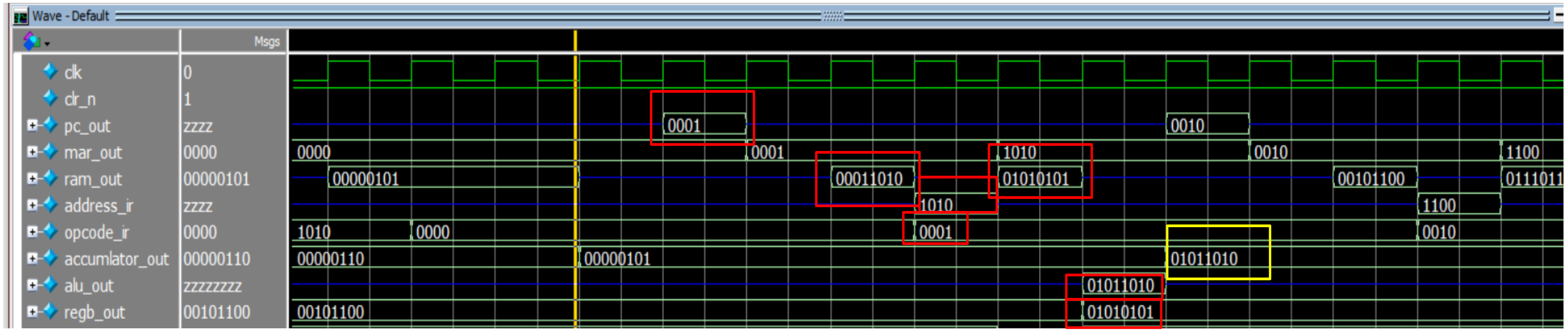

Wave - Default

	Msgs
clk	0
clr_n	1
pc_out	zzzz
mar_out	1111
ram_out	10101010
address_ir	zzzz
opcode_ir	1010
accumlator_out	00000110
accumlator_bus	zzzzzzzz
alu_out	zzzzzzzz
regb_out	00101100
output_register	00000000
f	2

simulation results

When pc =0001 which pointer to the second location in ram and the out of ram is 00011010, The instruction register divide it to op code=0001 which indicate to add.

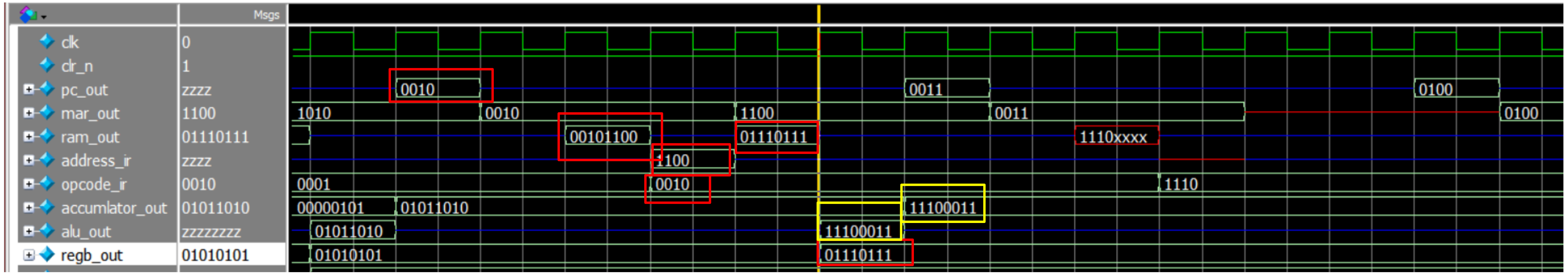
So the data at location 1010 load to reg b and added with accumulator data in alu and then store output in accumulator



simulation results

When pc =0010 which pointer to the third location in ram and the out of ram is 0010_1100, The instruction register divide it to op code=0010 which indicate to sub.

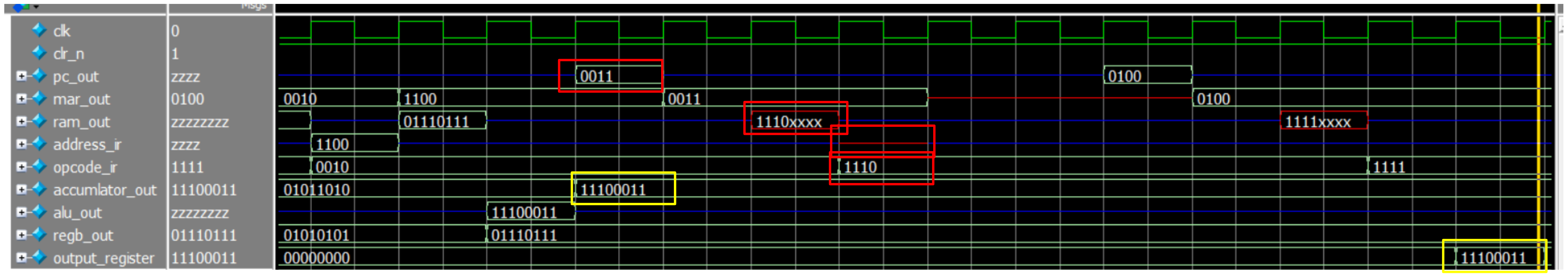
So the data at location 1100 load to reg b and subtract with accumulator data in alu and then store output in accumulator



simulation results

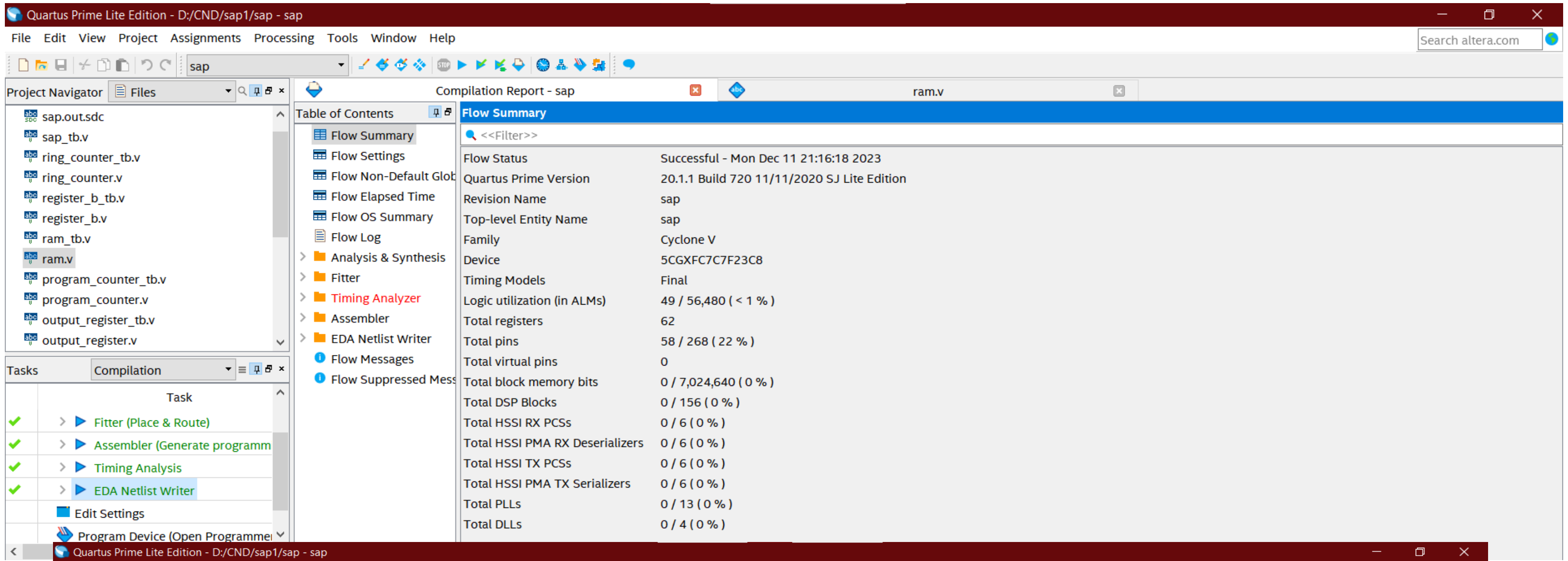
When pc =0011 which pointer to the fourth location in ram and the out of ram is 1110_xxxx, The instruction register divide it to op code=1110 which indicate to output.

So the data at accumulator will be go to output register.



simulation results

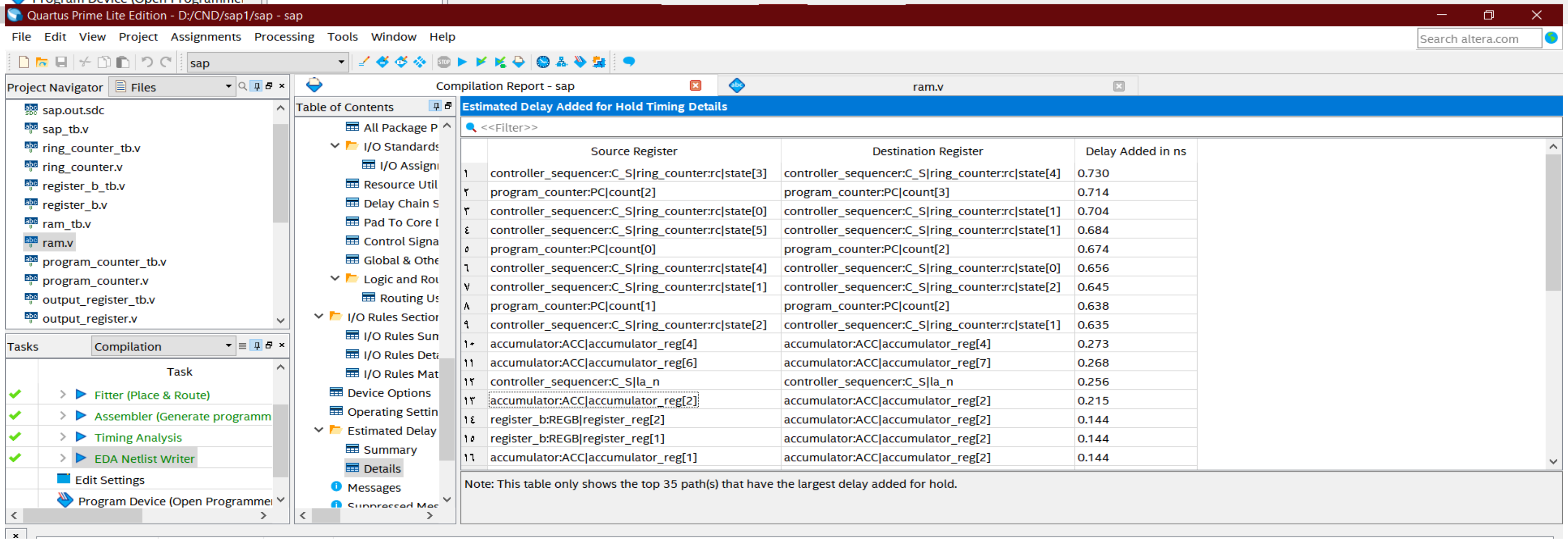
RTL Flow Summary results



Flow Summary

Flow Status	Successful - Mon Dec 11 21:16:18 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	sap
Top-level Entity Name	sap
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	49 / 56,480 (< 1 %)
Total registers	62
Total pins	58 / 268 (22 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

delay due to register



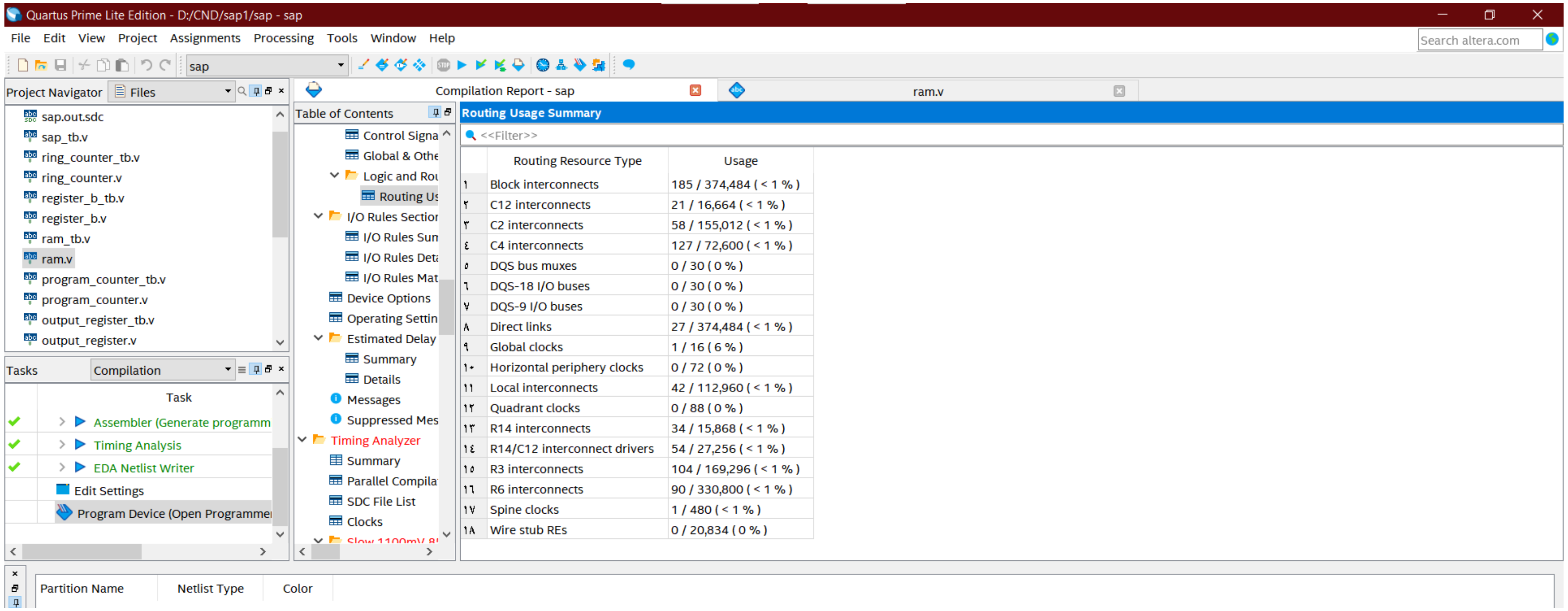
Estimated Delay Added for Hold Timing Details

	Source Register	Destination Register	Delay Added in ns
١	controller_sequencer:C_S|ring_counter:rc|state[3]	controller_sequencer:C_S|ring_counter:rc|state[4]	0.730
٢	program_counter:PC|count[2]	program_counter:PC|count[3]	0.714
٣	controller_sequencer:C_S|ring_counter:rc|state[0]	controller_sequencer:C_S|ring_counter:rc|state[1]	0.704
٤	controller_sequencer:C_S|ring_counter:rc|state[5]	controller_sequencer:C_S|ring_counter:rc|state[1]	0.684
٥	program_counter:PC|count[0]	program_counter:PC|count[2]	0.674
٦	controller_sequencer:C_S|ring_counter:rc|state[4]	controller_sequencer:C_S|ring_counter:rc|state[0]	0.656
٧	controller_sequencer:C_S|ring_counter:rc|state[1]	controller_sequencer:C_S|ring_counter:rc|state[2]	0.645
٨	program_counter:PC|count[1]	program_counter:PC|count[2]	0.638
٩	controller_sequencer:C_S|ring_counter:rc|state[2]	controller_sequencer:C_S|ring_counter:rc|state[1]	0.635
١٠	accumulator:ACC|accumulator_reg[4]	accumulator:ACC|accumulator_reg[4]	0.273
١١	accumulator:ACC|accumulator_reg[6]	accumulator:ACC|accumulator_reg[7]	0.268
١٢	controller_sequencer:C_S|la_n	controller_sequencer:C_S|la_n	0.256
١٣	accumulator:ACC|accumulator_reg[2]	accumulator:ACC|accumulator_reg[2]	0.215
١٤	register_b:REGB|register_reg[2]	accumulator:ACC|accumulator_reg[2]	0.144
١٥	register_b:REGB|register_reg[1]	accumulator:ACC|accumulator_reg[2]	0.144
١٦	accumulator:ACC|accumulator_reg[1]	accumulator:ACC|accumulator_reg[2]	0.144

Note: This table only shows the top 35 path(s) that have the largest delay added for hold.

simulation results

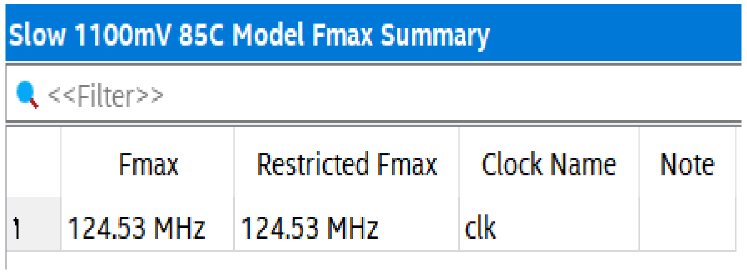
Routing summary



	Routing Resource Type	Usage
١	Block interconnects	185 / 374,484 (< 1 %)
٢	C12 interconnects	21 / 16,664 (< 1 %)
٣	C2 interconnects	58 / 155,012 (< 1 %)
٤	C4 interconnects	127 / 72,600 (< 1 %)
٥	DQS bus muxes	0 / 30 (0 %)
٦	DQS-18 I/O buses	0 / 30 (0 %)
٧	DQS-9 I/O buses	0 / 30 (0 %)
٨	Direct links	27 / 374,484 (< 1 %)
٩	Global clocks	1 / 16 (6 %)
١٠	Horizontal periphery clocks	0 / 72 (0 %)
١١	Local interconnects	42 / 112,960 (< 1 %)
١٢	Quadrant clocks	0 / 88 (0 %)
١٣	R14 interconnects	34 / 15,868 (< 1 %)
١٤	R14/C12 interconnect drivers	54 / 27,256 (< 1 %)
١٥	R3 interconnects	104 / 169,296 (< 1 %)
١٦	R6 interconnects	90 / 330,800 (< 1 %)
١٧	Spine clocks	1 / 480 (< 1 %)
١٨	Wire stub REs	0 / 20,834 (0 %)

simulation results

The maximum frequency to work in slow 1100mv 85c model show in the next figure:

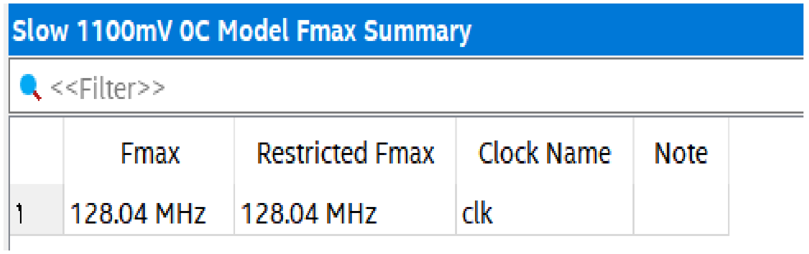


Slow 1100mV 85C Model Fmax Summary

<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	124.53 MHz	124.53 MHz	clk	

The maximum frequency to work in slow 1100mv 0c model show in the next figure:



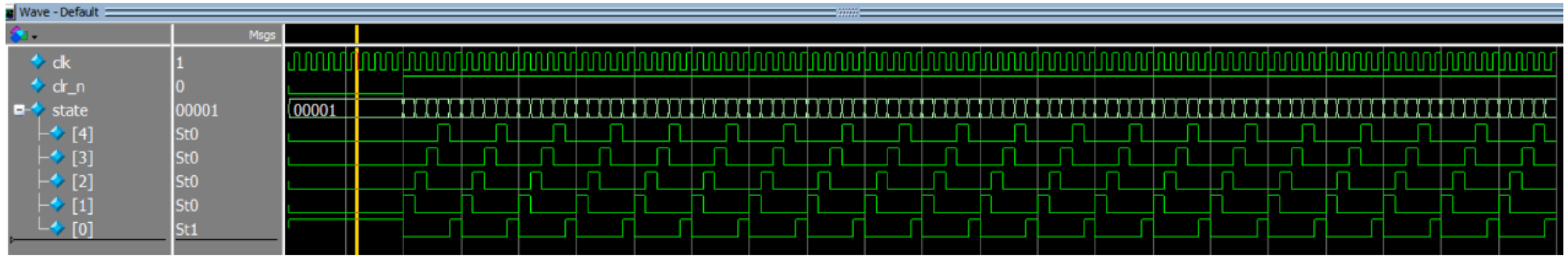
Slow 1100mV 0C Model Fmax Summary

<<Filter>>

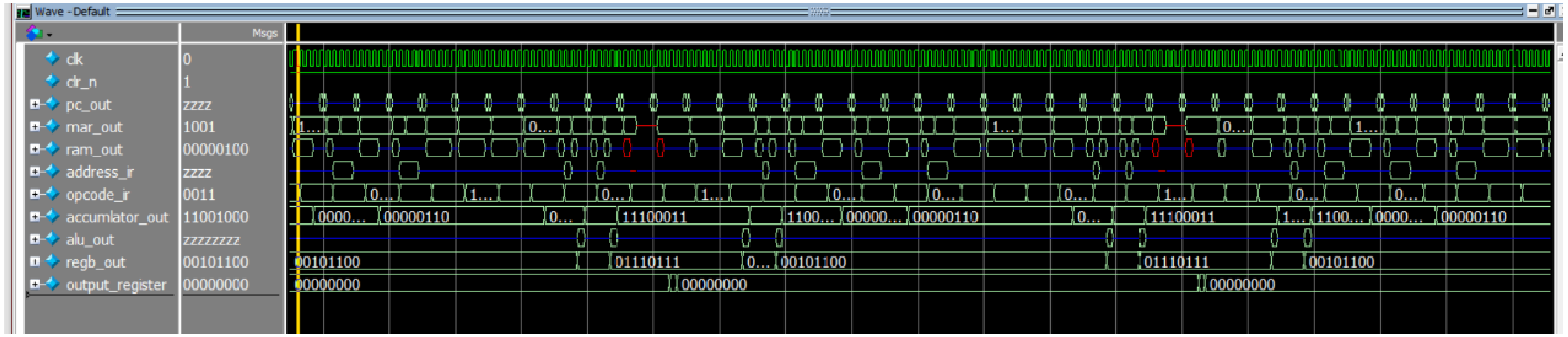
	Fmax	Restricted Fmax	Clock Name	Note
1	128.04 MHz	128.04 MHz	clk	

design improvement

I work to make the design work only in five clock cycle instead of six clocks so the ring counter will be as in next figure

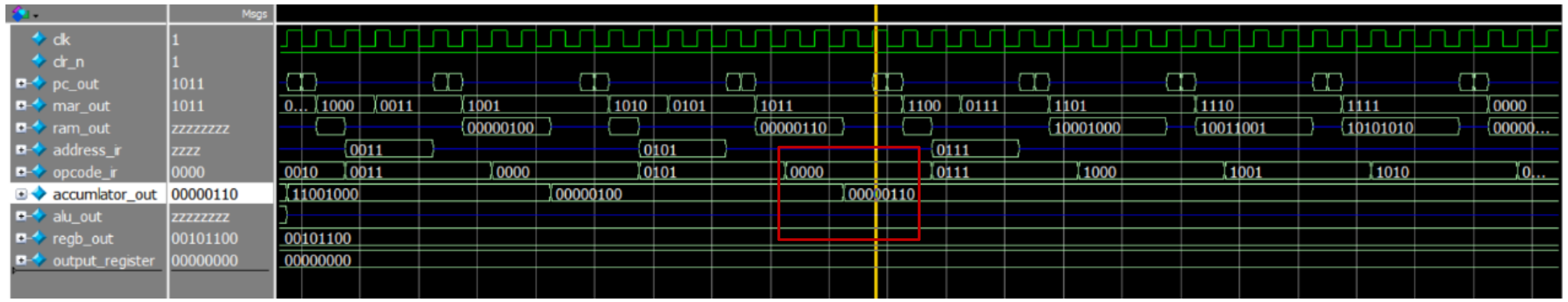


The figure below shows the RTL simulation results for the sap1 .



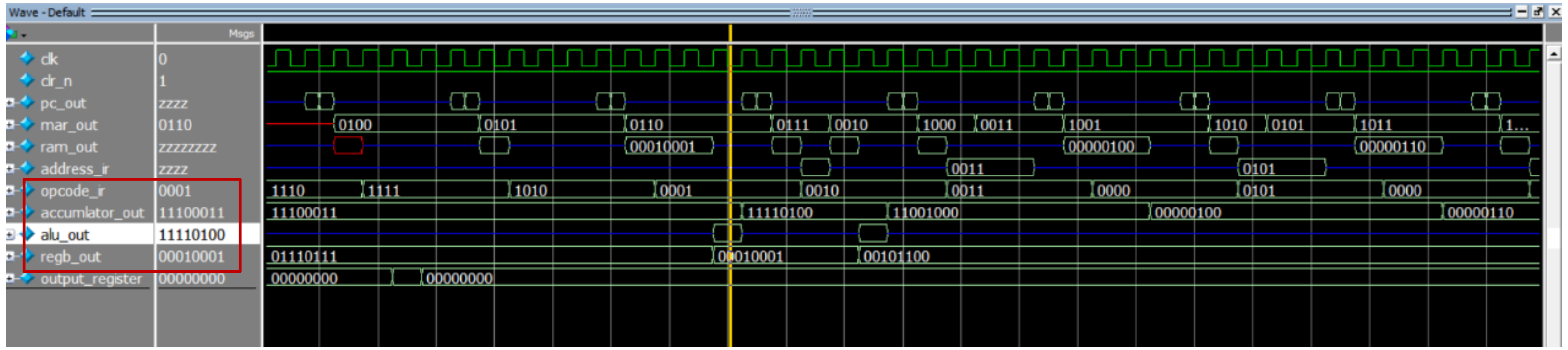
design improvement

When pc =0000 which pointer to the first location in ram and the out of ram is 00000110, The instruction register divide it to op code=0000 which indicate to load the data from ram to accumulator like previous simulation.



```
clk              1
clr_n            1
pc_out           1011
mar_out          1011
ram_out          zzzzzzzz
address_ir       zzzz
opcode_ir        0000
accumulator_out  00000110
alu_out          zzzzzzzz
regb_out         00101100
output_register  00000000
```

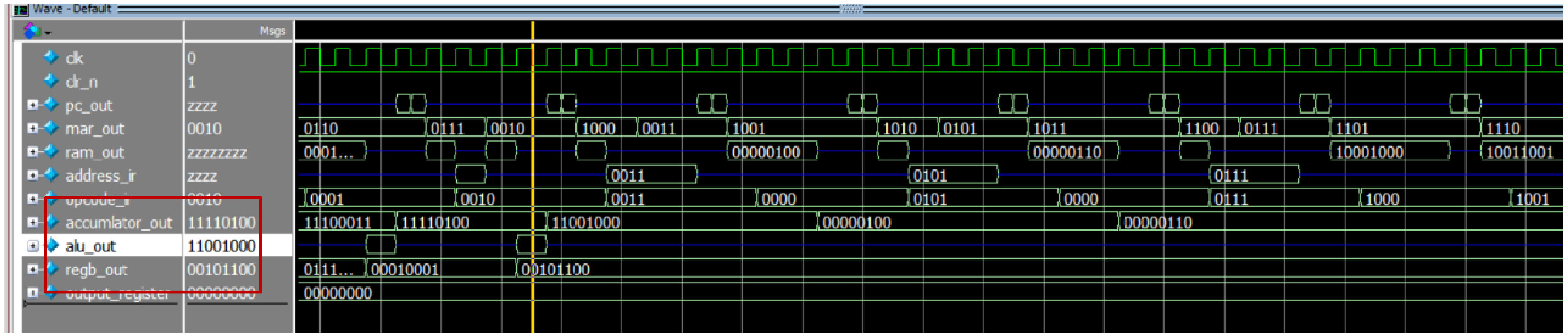
When pc =0001 which pointer to the second location in ram and the out of ram is 00010001, The instruction register divide it to op code=0001 which indicate to add.



```
clk              0
clr_n            1
pc_out           zzzz
mar_out          0110
ram_out          zzzzzzzz
address_ir       zzzz
opcode_ir        0001
accumulator_out  11100011
alu_out          11110100
regb_out         00010001
output_register  00000000
```

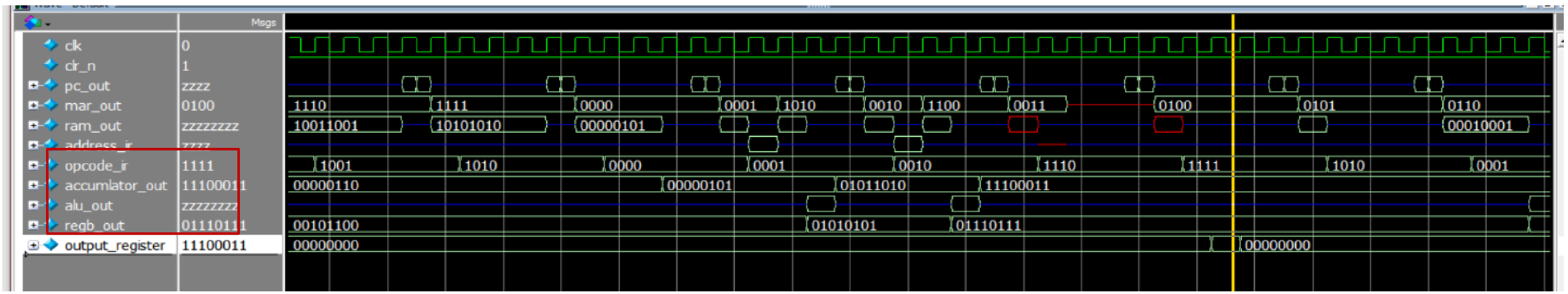
design improvement

And when op code= 0010 which indicate to sub.



```
Wave - Default
                    Msgs
clk                 0
clr_n               1
pc_out              zzzz
mar_out             0010
ram_out             zzzzzzzz
address_ir          zzzz
opcode_ir           0010
accumulator_out     11110100
alu_out             11001000
regb_out            00101100
output_register     00000000
```

When op code =1110 to output data



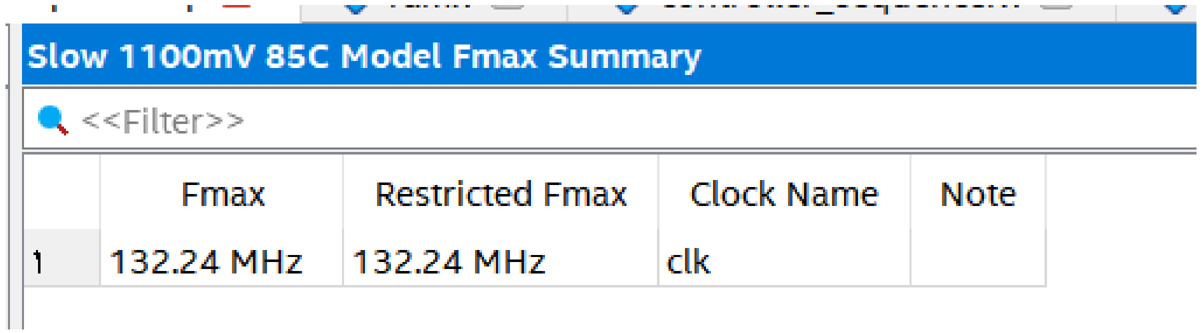
```
                    Msgs
clk                 0
clr_n               1
pc_out              zzzz
mar_out             0100
ram_out             zzzzzzzz
address_ir          zzzz
opcode_ir           1111
accumulator_out     11100011
alu_out             zzzzzzzz
regb_out            01110111
output_register     11100011
```

design improvement

At six clock cycles the maximum frequency in technique slow 1100mv 85c =124 MHZ And in technique slow 1100mv 0c =128 MHZ.

After making enhancement:

At five clock cycles the maximum frequency in technique slow 1100mv 85c =132 MHZ

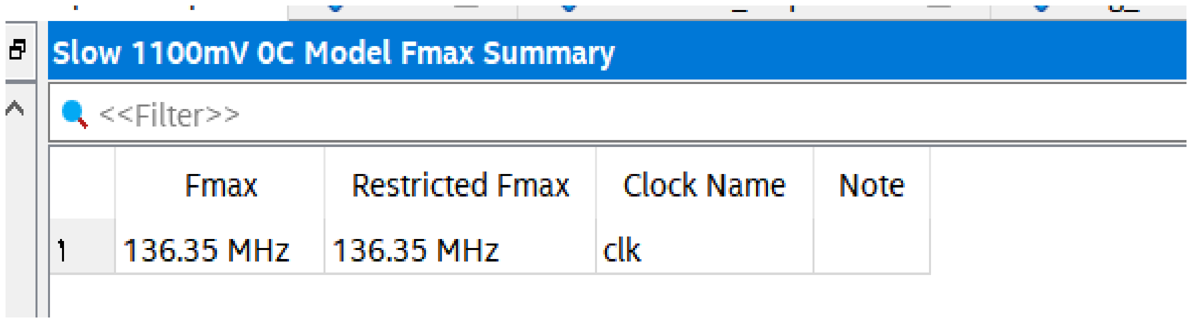


Slow 1100mV 85C Model Fmax Summary

<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	132.24 MHz	132.24 MHz	clk	

And in technique slow 1100mv 0c =136 MHZ



Slow 1100mV 0C Model Fmax Summary

<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	136.35 MHz	136.35 MHz	clk	

Thank You