Name: Duc Anh Nguyen

Course: EE126

Date: 10.24.2022

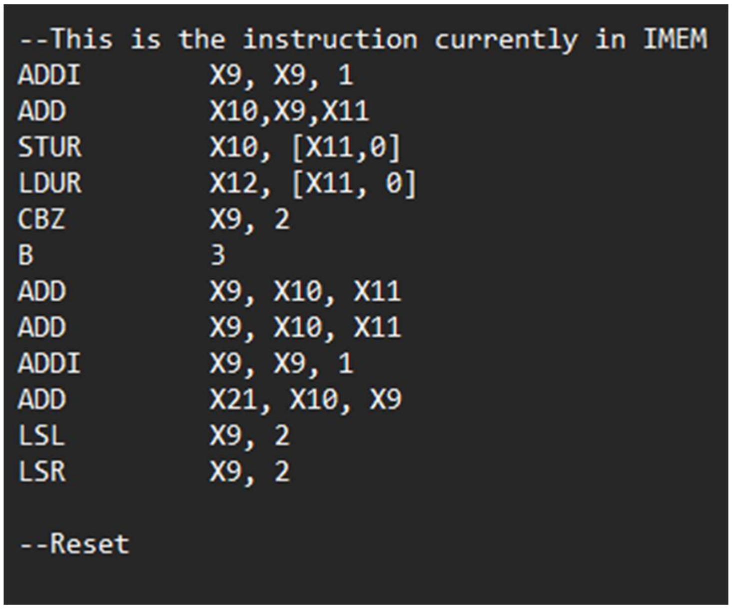
Lab Report 3

I. Introduction:

- The goal of this lab is to implement the previous components from the earlier lab to form a single-cycle CPU that runs LEGv8 Instructions set.

II. Procedure:

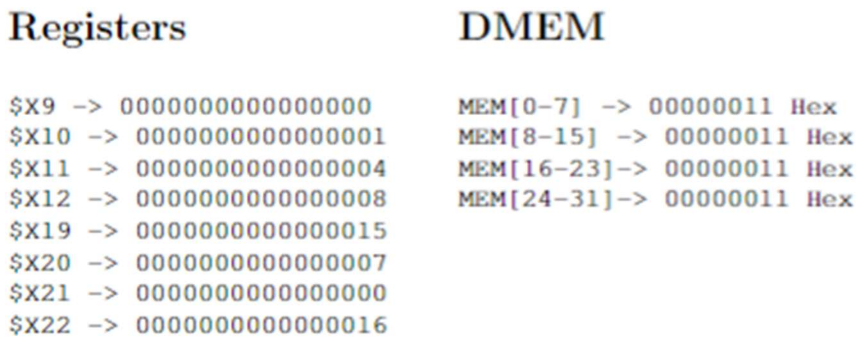
- The final test of the program consists of these instructions and contents of the registers and memory are as follows
- Instruction in IMEM:



```
--This is the instruction currently in IMEM
ADDI        X9, X9, 1
ADD         X10,X9,X11
STUR        X10, [X11,0]
LDUR        X12, [X11, 0]
CBZ         X9, 2
B           3
ADD         X9, X10, X11
ADD         X9, X10, X11
ADDI        X9, X9, 1
ADD         X21, X10, X9
LSL         X9, 2
LSR         X9, 2

--Reset
```

- Contents of Memory and Register:

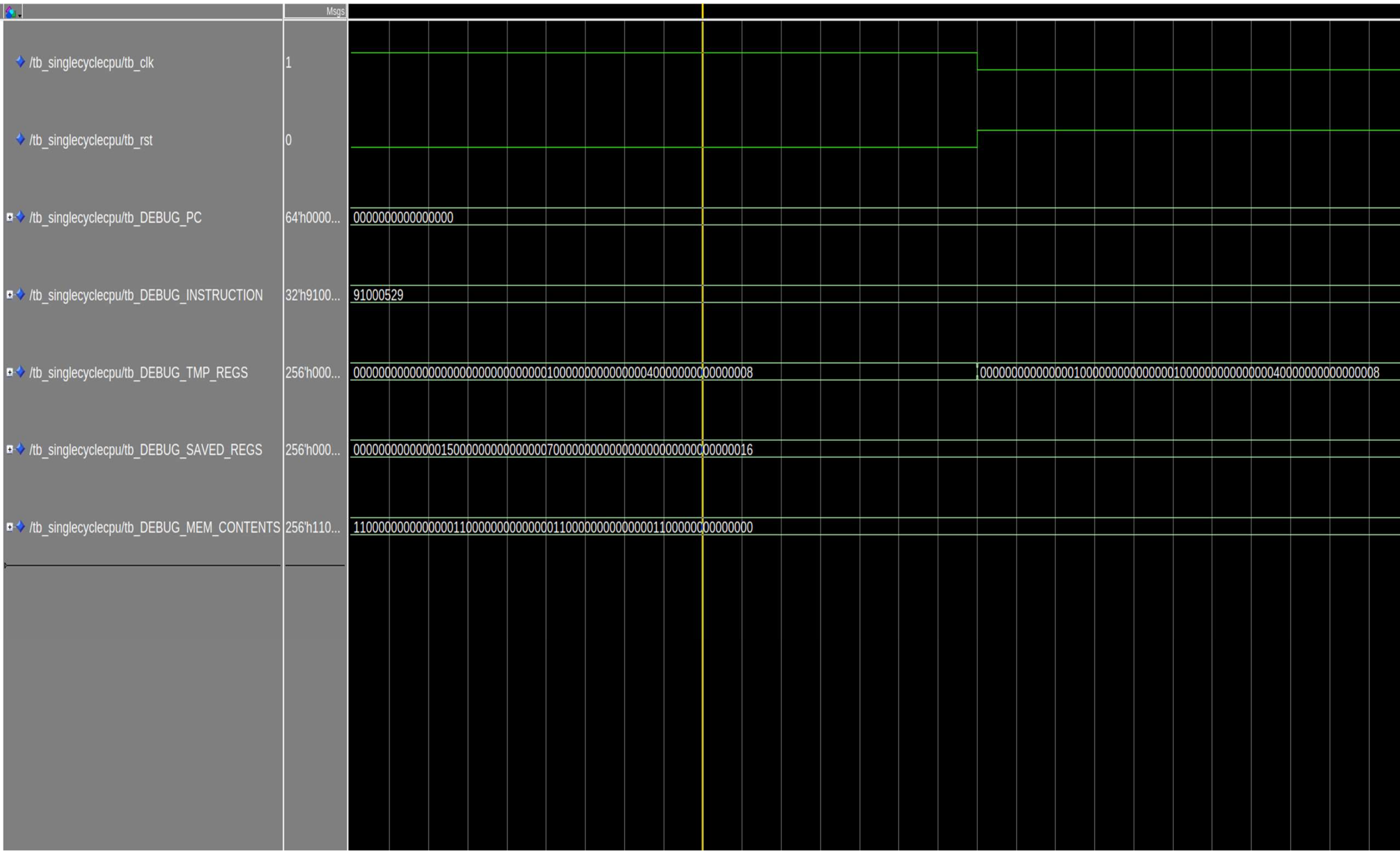


```
Registers                    DMEM

$X9 -> 0000000000000000      MEM[0-7] -> 00000011 Hex
$X10 -> 0000000000000001     MEM[8-15] -> 00000011 Hex
$X11 -> 0000000000000004     MEM[16-23]-> 00000011 Hex
$X12 -> 0000000000000008     MEM[24-31]-> 00000011 Hex
$X19 -> 0000000000000015
$X20 -> 0000000000000007
$X21 -> 0000000000000000
$X22 -> 0000000000000016
```

- Signal of Waveform Notation
  - DEBUG_PC: Program Counter
  - DEBUG_INSTRUCTIONS: Instruction that currently being executed
  - DEBUG_TMP_REGS: X9 & X10 & X11 & X12
  - DEBUG_SAVE_REZGS: X19 & X20 & X21 & X22
  - DEBUG_MEM_CONTENTS: mem (0) & … & mem (31)
- Each Registers index are 16 bytes
- Each Memory index are 2 bytes

Cycle 1:

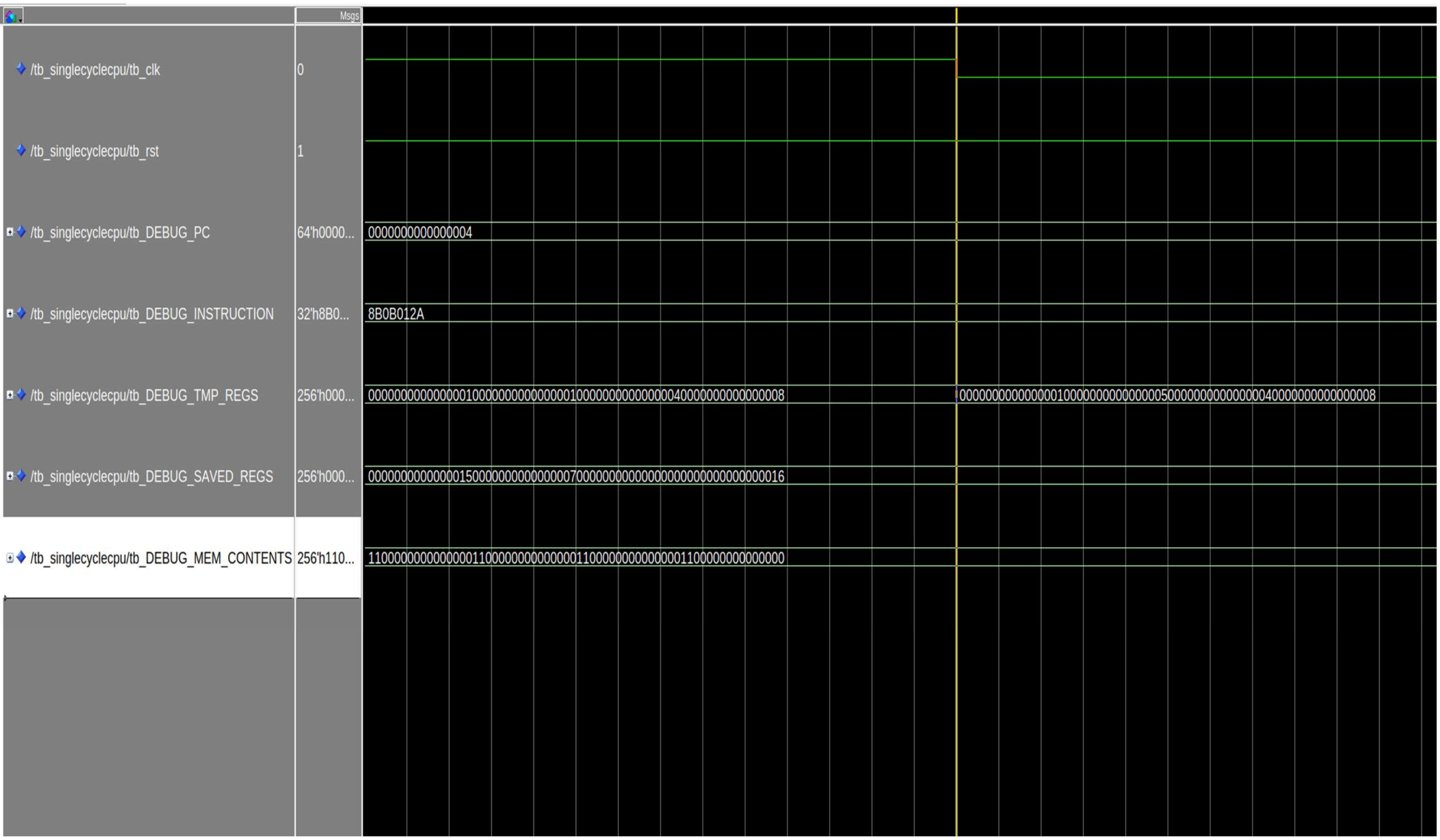


```
                                        Msgs
/tb_singlecyclecpu/tb_clk               1
/tb_singlecyclecpu/tb_rst               0
/tb_singlecyclecpu/tb_DEBUG_PC          64'h0000...   0000000000000000
/tb_singlecyclecpu/tb_DEBUG_INSTRUCTION 32'h9100...   91000529
/tb_singlecyclecpu/tb_DEBUG_TMP_REGS    256'h000...   0000000000000000000000000000000100000000000000040000000000000008   0000000000000001000000000000000100000000000000040000000000000008
/tb_singlecyclecpu/tb_DEBUG_SAVED_REGS  256'h000...   0000000000000015000000000000000700000000000000000000000000000016
/tb_singlecyclecpu/tb_DEBUG_MEM_CONTENTS 256'h110...  1100000000000000110000000000000011000000000000001100000000000000
```

ADDI X9, X9, 1

- Program Counter starts at 0x0000000000000000
- It fetches the 32-bit instruction in hexadecimal
- Register X9 changes from 0x000 to 0x001 in the DEBUG_TMP_REGS, which the original content is 0x000.
- Nothing changes in other registers or memory contents

Cycle 2:

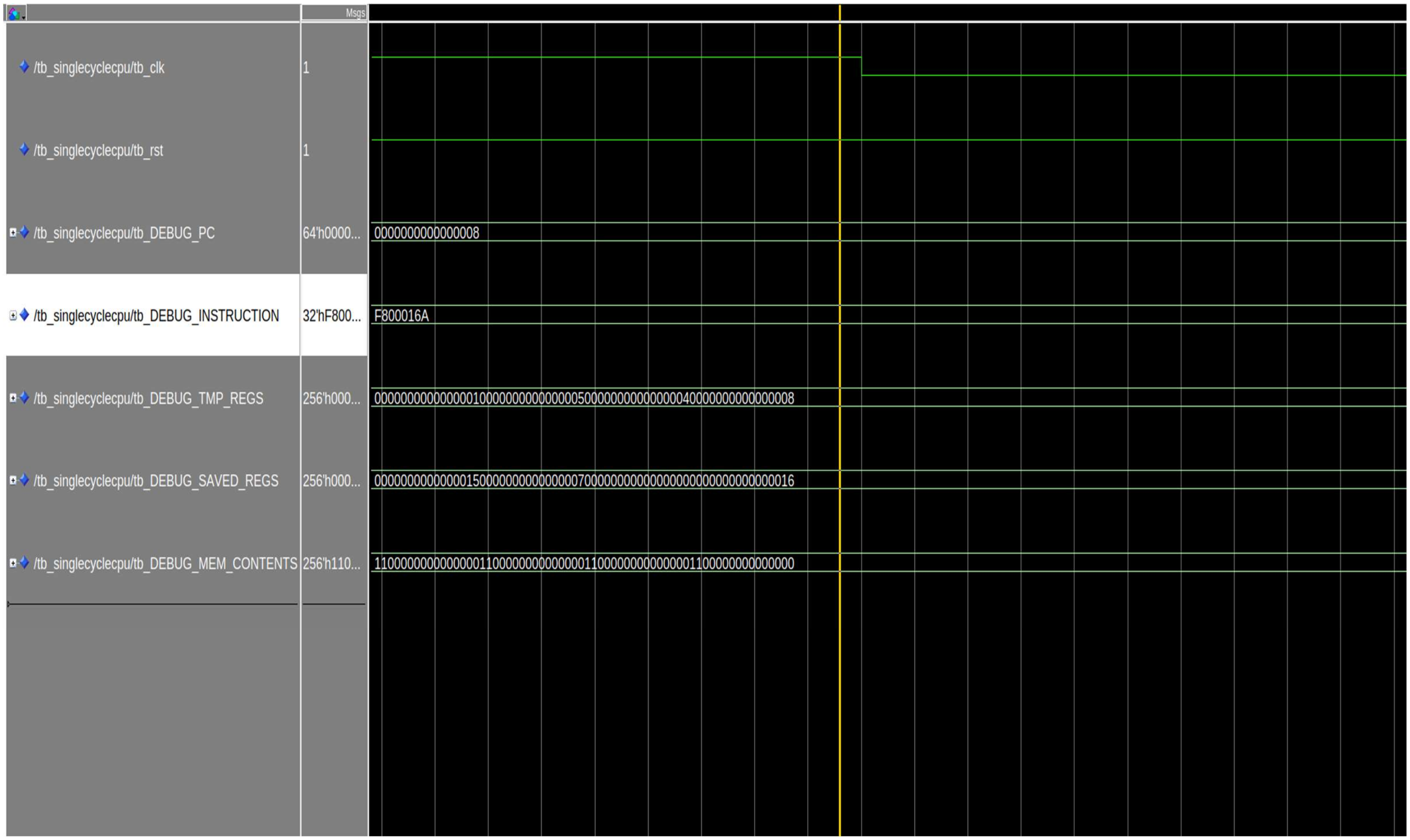


```
Msgs
/tb_singlecyclecpu/tb_clk                 0
/tb_singlecyclecpu/tb_rst                 1
/tb_singlecyclecpu/tb_DEBUG_PC            64'h0000...   0000000000000004
/tb_singlecyclecpu/tb_DEBUG_INSTRUCTION   32'h8B0...    8B0B012A
/tb_singlecyclecpu/tb_DEBUG_TMP_REGS      256'h000...   0000000000000001000000000000000100000000000000040000000000000008   0000000000000001000000000000000500000000000000040000000000000008
/tb_singlecyclecpu/tb_DEBUG_SAVED_REGS    256'h000...   0000000000000015000000000000000700000000000000000000000000000016
/tb_singlecyclecpu/tb_DEBUG_MEM_CONTENTS  256'h110...   1100000000000000110000000000000011000000000000001100000000000000
```

ADD X10, X9, X11

- Program Counter increment to 0x0000000000000004
- It fetches the 32-bit instruction in hexadecimal
- Register X10 changes from 0x001 to 0x005 in the DEBUG_TMP_REGS because the addition of X11 (0x004) and X9 (0x001) is 0x005.
- Nothing changes in other registers or memory contents

Cycle 3:

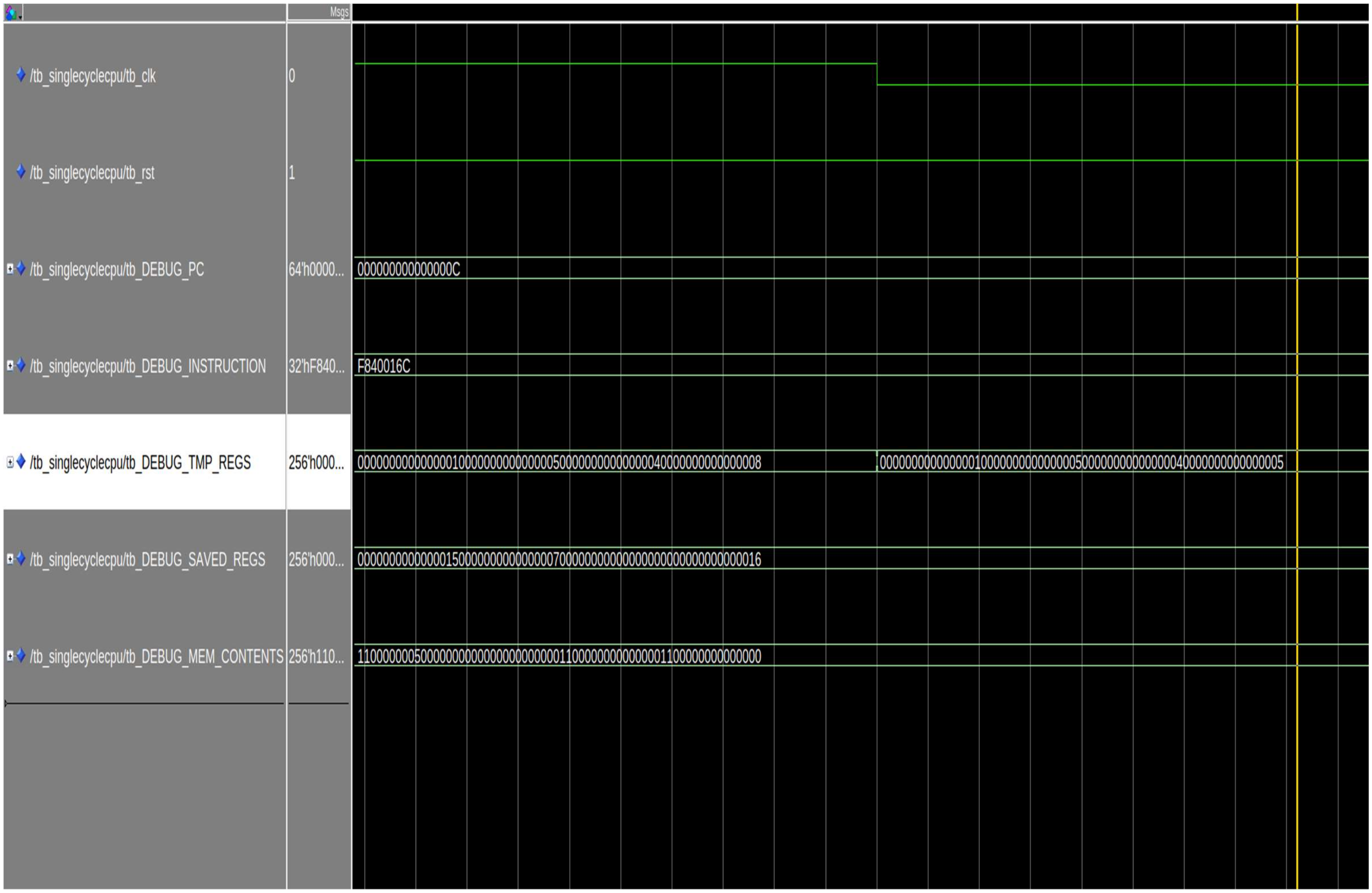


```
/tb_singlecyclecpu/tb_clk                 1
/tb_singlecyclecpu/tb_rst                 1
/tb_singlecyclecpu/tb_DEBUG_PC            64'h0000...   0000000000000008
/tb_singlecyclecpu/tb_DEBUG_INSTRUCTION   32'hF800...   F800016A
/tb_singlecyclecpu/tb_DEBUG_TMP_REGS      256'h000...   0000000000000001000000000000000500000000000000040000000000000008
/tb_singlecyclecpu/tb_DEBUG_SAVED_REGS    256'h000...   0000000000000015000000000000000700000000000000000000000000000016
/tb_singlecyclecpu/tb_DEBUG_MEM_CONTENTS  256'h110...   1100000000000000110000000000000011000000000000001100000000000000
```

STUR X10, [X11, 0]

- Program Counter increment to 0x0000000000000008
- It fetches the 32-bit instruction in hexadecimal
- The CPU is storing the value of X10, 0x005, to memory contents mem(4) because X11 is 0x004
- Nothing appears in the waveform because the data is going to be ready in the next clock cycle

Cycle 4:

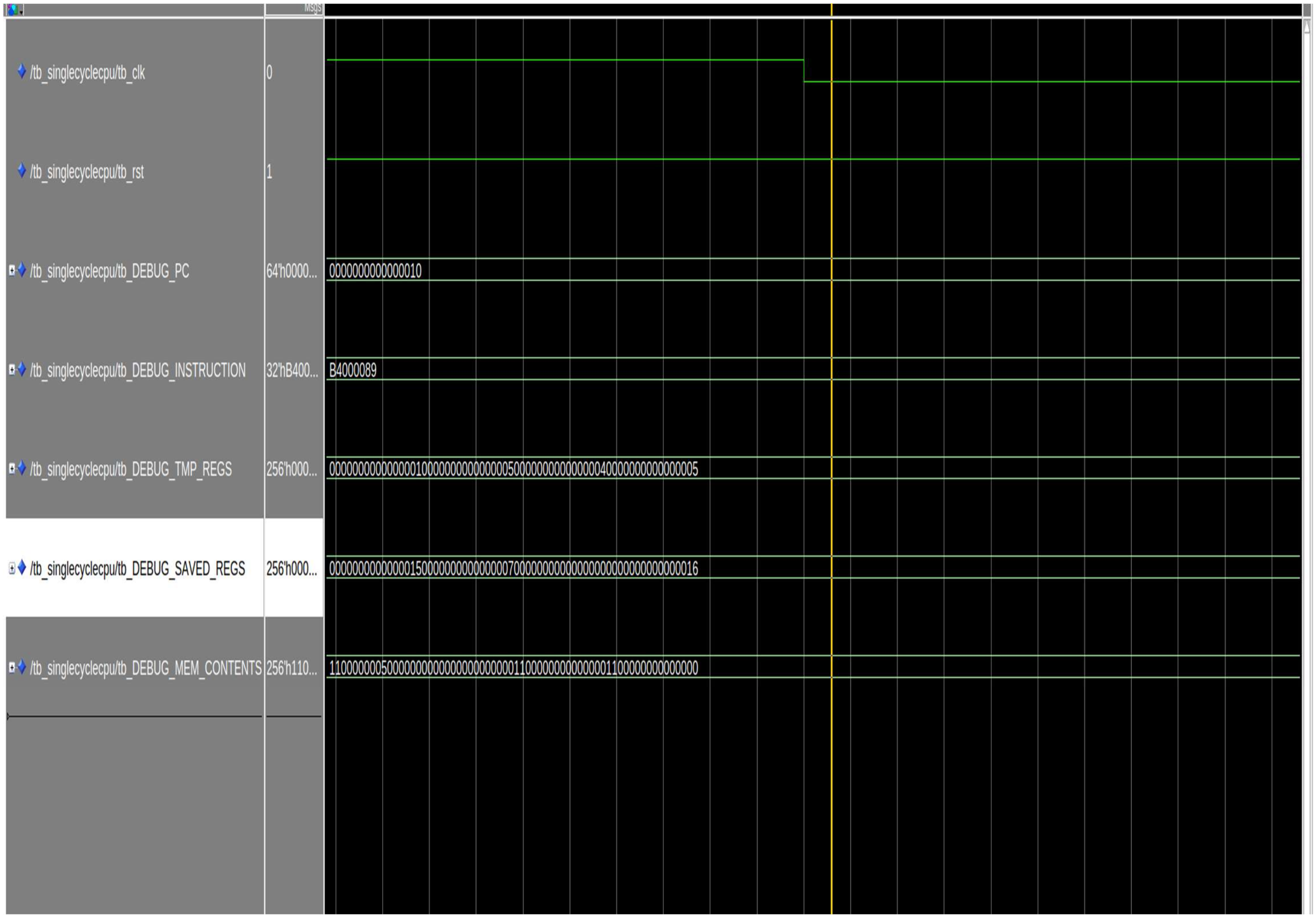


```
/tb_singlecyclecpu/tb_clk                 0
/tb_singlecyclecpu/tb_rst                 1
/tb_singlecyclecpu/tb_DEBUG_PC            64'h0000...   000000000000000C
/tb_singlecyclecpu/tb_DEBUG_INSTRUCTION   32'hF840...   F840016C
/tb_singlecyclecpu/tb_DEBUG_TMP_REGS      256'h000...   0000000000000001000000000000000500000000000000040000000000000008   0000000000000001000000000000000500000000000000040000000000000005
/tb_singlecyclecpu/tb_DEBUG_SAVED_REGS    256'h000...   0000000000000015000000000000000700000000000000000000000000000016
/tb_singlecyclecpu/tb_DEBUG_MEM_CONTENTS  256'h110...   1100000005000000000000000000000011000000000000001100000000000000
```

LDUR X12, [X11, 0]

- Program Counter increment to 0x000000000000000C
- It fetches the 32-bit instruction in hexadecimal
- In the rising edge of the clock, we can see that the contents of DEBUG_MEM_CONTENTS has changes to 0x005 in mem(4)
- Moreover, in the falling edge of the clock, the CPU writes into the register X12 the contents of mem(4) which is 0x005

Cycle 5:

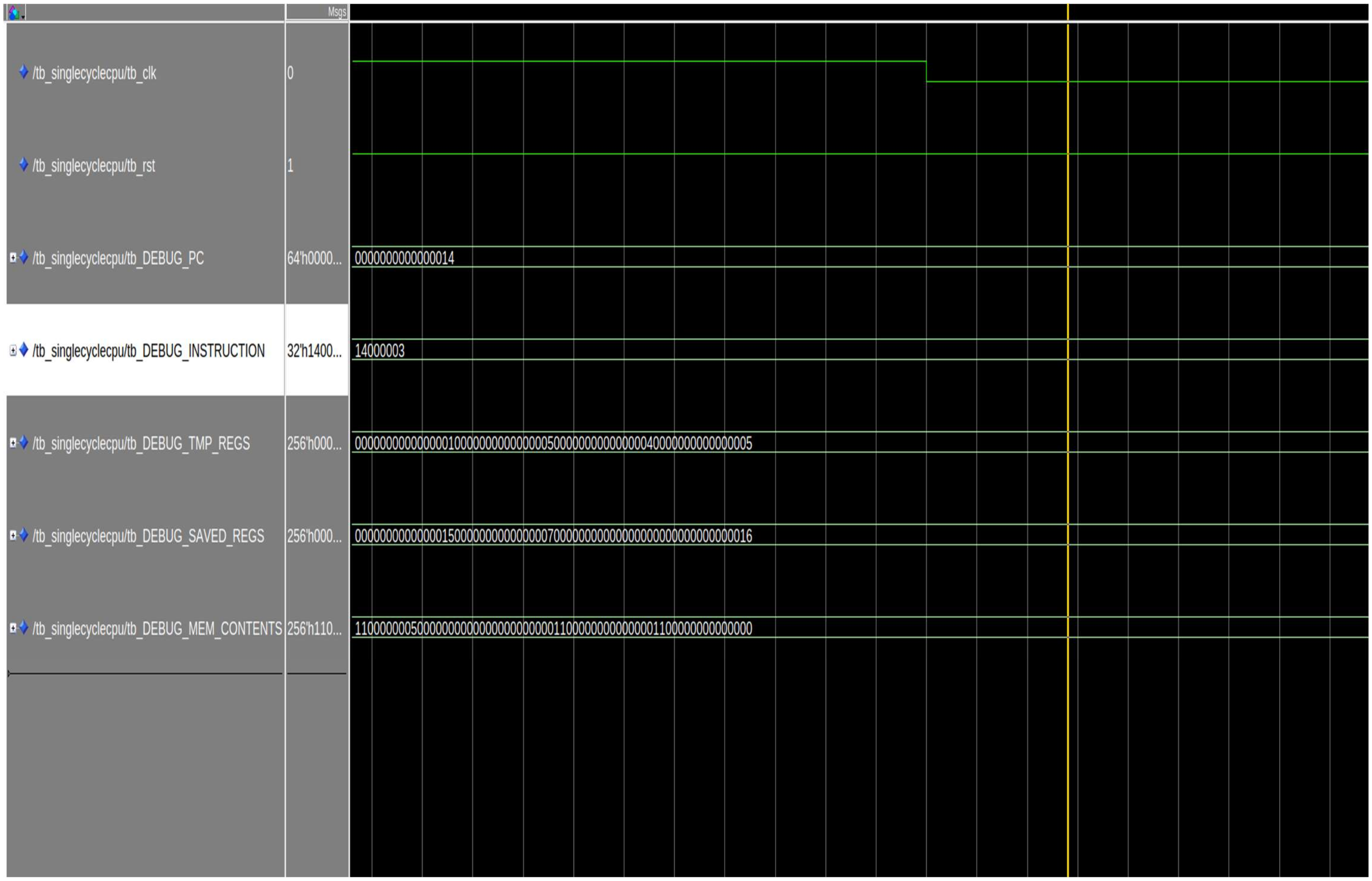


```
/tb_singlecyclecpu/tb_clk                 0
/tb_singlecyclecpu/tb_rst                 1
/tb_singlecyclecpu/tb_DEBUG_PC            64'h0000...   0000000000000010
/tb_singlecyclecpu/tb_DEBUG_INSTRUCTION   32'hB400...   B4000089
/tb_singlecyclecpu/tb_DEBUG_TMP_REGS      256'h000...   0000000000000001000000000000000500000000000000040000000000000005
/tb_singlecyclecpu/tb_DEBUG_SAVED_REGS    256'h000...   0000000000000015000000000000000700000000000000000000000000000016
/tb_singlecyclecpu/tb_DEBUG_MEM_CONTENTS  256'h110...   1100000005000000000000000000000011000000000000001100000000000000
```

CBZ X9, 2

- Program Counter increment to 0x0000000000000010
- It fetches the 32-bit instruction in hexadecimal
- Nothing happen here because the register X9 is not 0, so it will not jump

Cycle 6:

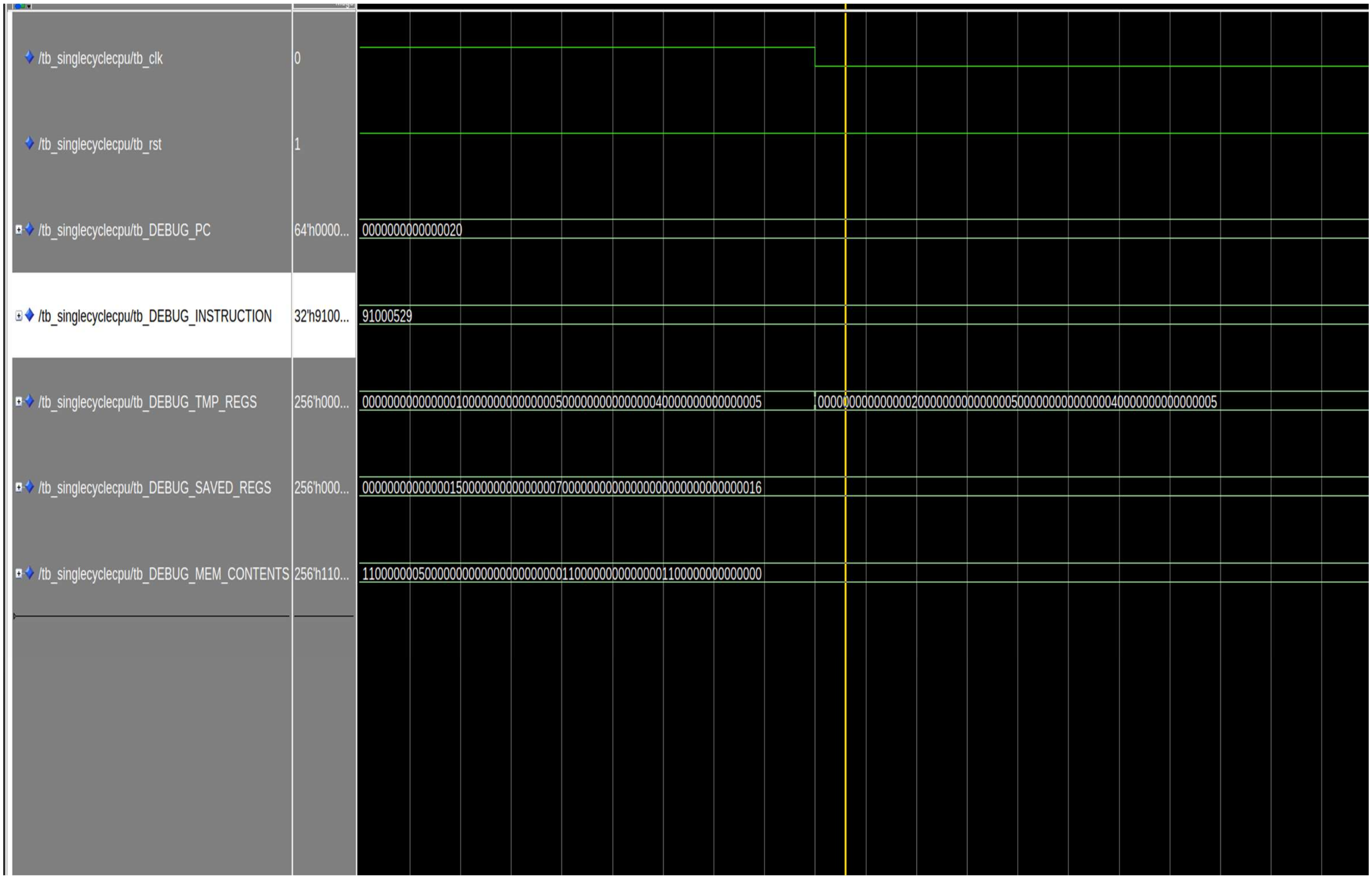


```
/tb_singlecyclecpu/tb_clk                  0
/tb_singlecyclecpu/tb_rst                  1
/tb_singlecyclecpu/tb_DEBUG_PC             64'h0000...   0000000000000014
/tb_singlecyclecpu/tb_DEBUG_INSTRUCTION    32'h1400...   14000003
/tb_singlecyclecpu/tb_DEBUG_TMP_REGS       256'h000...   0000000000000001000000000000000500000000000000040000000000000005
/tb_singlecyclecpu/tb_DEBUG_SAVED_REGS     256'h000...   0000000000000015000000000000000700000000000000000000000000000016
/tb_singlecyclecpu/tb_DEBUG_MEM_CONTENTS   256'h110...   1100000005000000000000000000000011000000000000001100000000000000
```

B 3

- Program Counter increment to 0x0000000000000014
- It fetches the 32-bit instruction in hexadecimal
- The program counter will jump to the offset + program counter location in the next clock cycle after executing this instruction

Cycle 7:

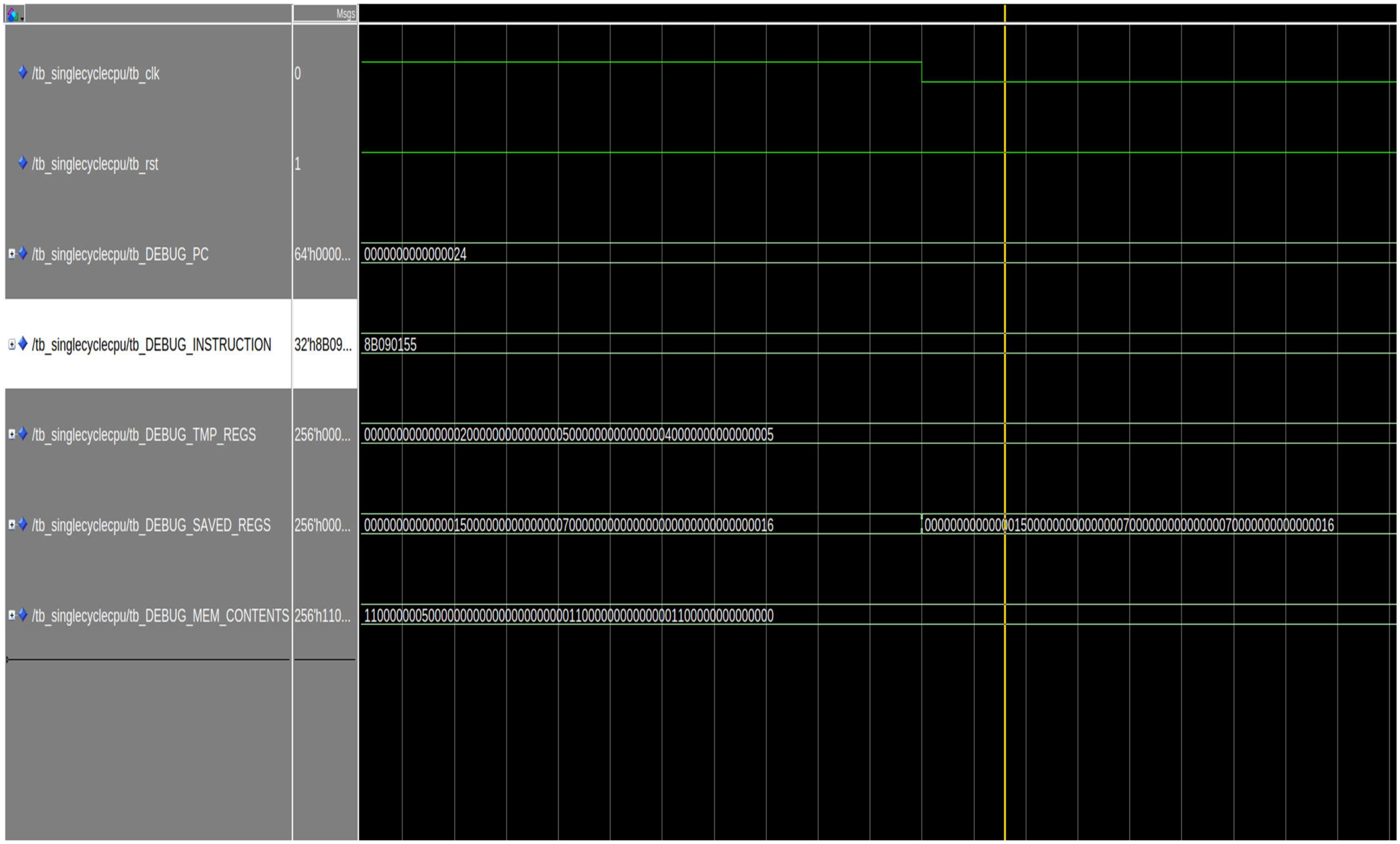


```
/tb_singlecyclecpu/tb_clk                  0
/tb_singlecyclecpu/tb_rst                  1
/tb_singlecyclecpu/tb_DEBUG_PC             64'h0000...   0000000000000020
/tb_singlecyclecpu/tb_DEBUG_INSTRUCTION    32'h9100...   91000529
/tb_singlecyclecpu/tb_DEBUG_TMP_REGS       256'h000...   0000000000000001000000000000000500000000000000040000000000000005   0000000000000002000000000000000500000000000000040000000000000005
/tb_singlecyclecpu/tb_DEBUG_SAVED_REGS     256'h000...   0000000000000015000000000000000700000000000000000000000000000016
/tb_singlecyclecpu/tb_DEBUG_MEM_CONTENTS   256'h110...   1100000005000000000000000000000011000000000000001100000000000000
```

ADD X9, X9, 1

- Program Counter branch to 0x0000000000000020 instead of incrementing by 4 to 0x000000000000014
- It fetches the 32-bit instruction in hexadecimal
- The registers X9 got updated into 0x002 because of X9 = X9 + 1, in which the previous contents of X9 is 0x001

Cycle 8:

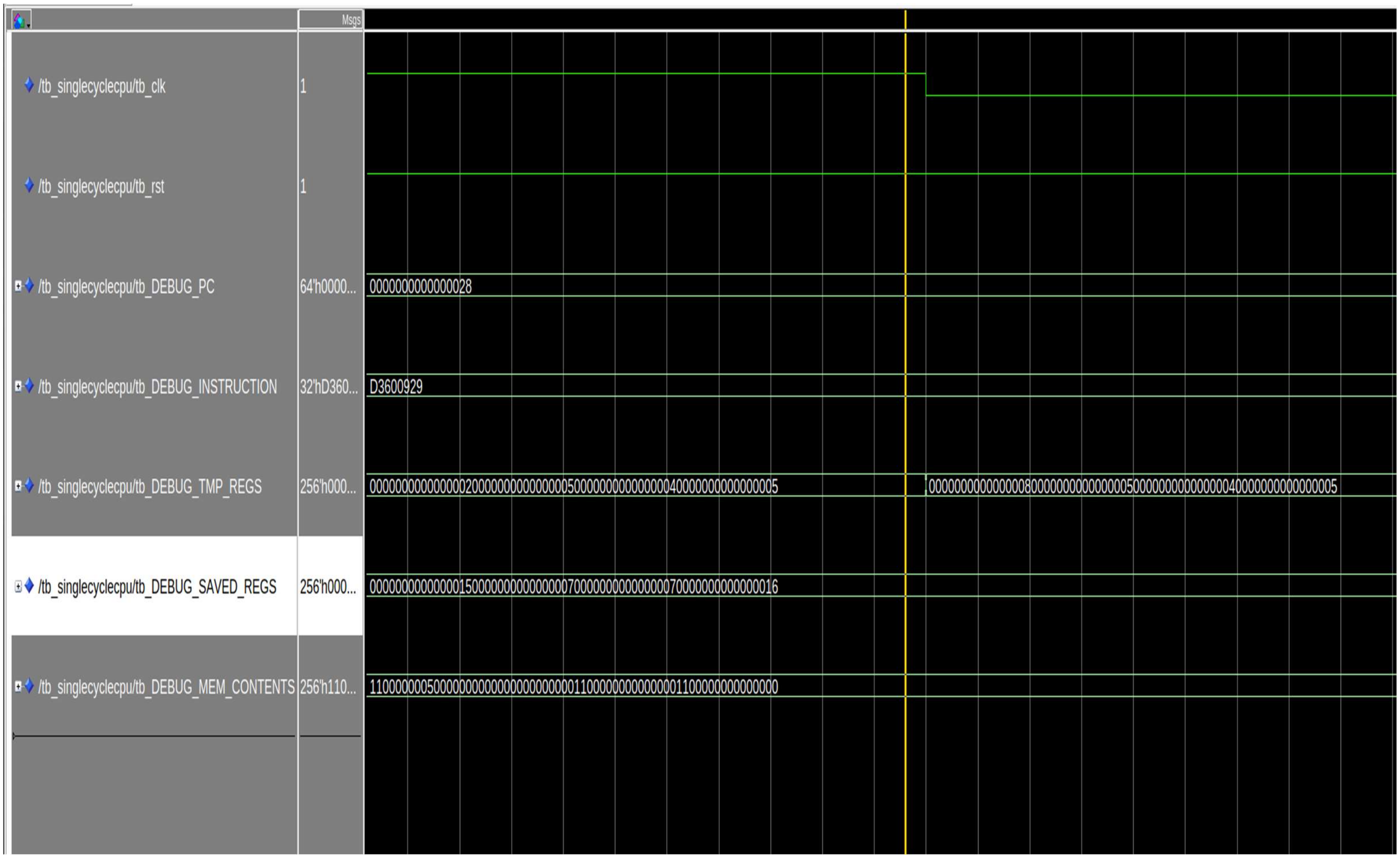


```
                                          Msgs
/tb_singlecyclecpu/tb_clk                 0
/tb_singlecyclecpu/tb_rst                 1
/tb_singlecyclecpu/tb_DEBUG_PC            64'h0000...   0000000000000024
/tb_singlecyclecpu/tb_DEBUG_INSTRUCTION   32'h8B09...   8B090155
/tb_singlecyclecpu/tb_DEBUG_TMP_REGS      256'h000...   0000000000000002000000000000000500000000000000040000000000000005
/tb_singlecyclecpu/tb_DEBUG_SAVED_REGS    256'h000...   0000000000000015000000000000000700000000000000000000000000000016   0000000000000015000000000000000700000000000000070000000000000016
/tb_singlecyclecpu/tb_DEBUG_MEM_CONTENTS  256'h110...   1100000005000000000000000000000000110000000000000011000000000000
```

ADD X21, X10, X9

- Program Counter increments to 0x0000000000000024
- It fetches the 32-bit instruction in hexadecimal
- The summation of X10, 0x005, and X9, 0x002, is 0x007, which is being stored in X21

Cycle 9:

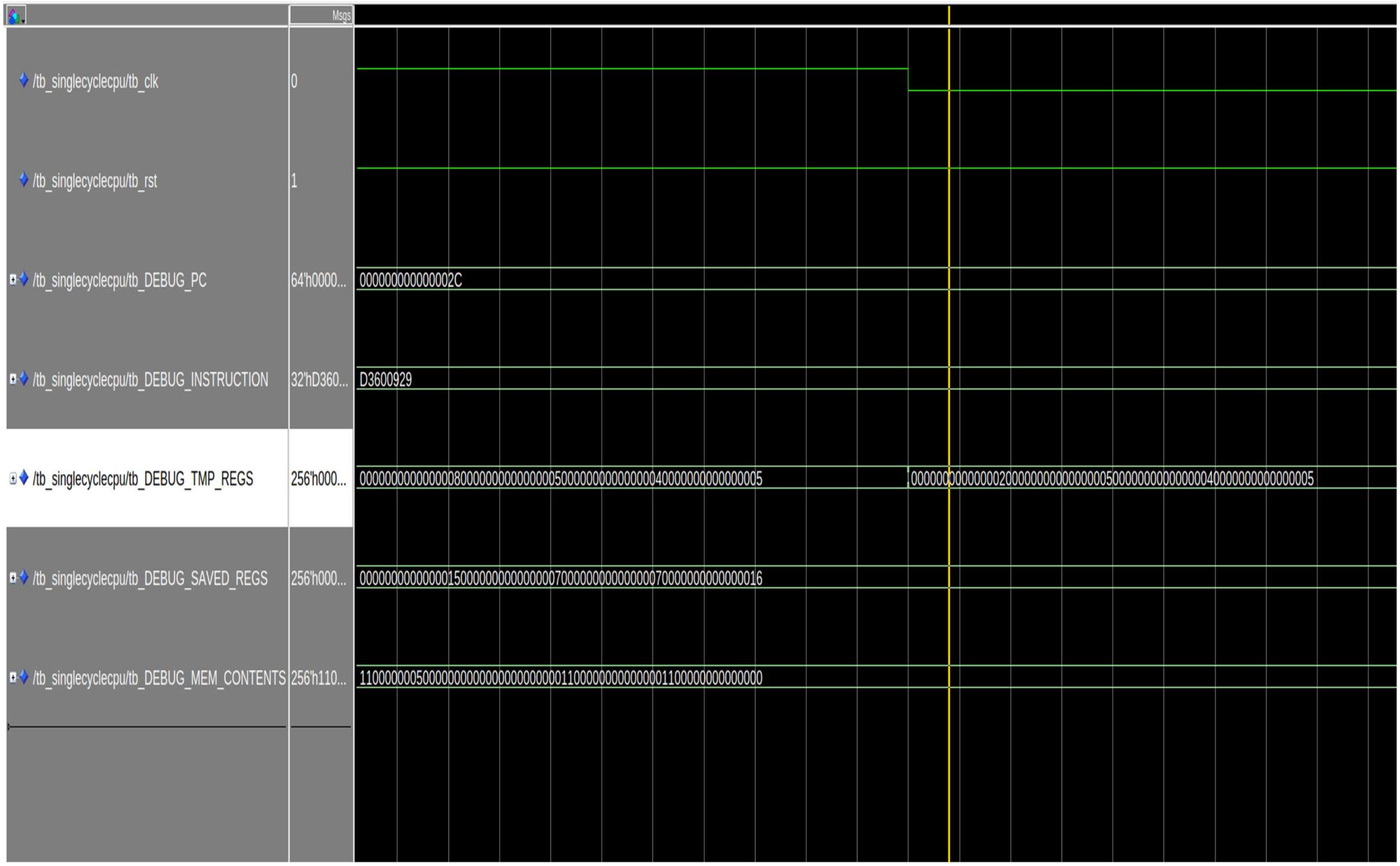


```
                                            Msgs
/tb_singlecyclecpu/tb_clk                   1
/tb_singlecyclecpu/tb_rst                   1
/tb_singlecyclecpu/tb_DEBUG_PC              64'h0000...   0000000000000028
/tb_singlecyclecpu/tb_DEBUG_INSTRUCTION     32'hD360...   D3600929
/tb_singlecyclecpu/tb_DEBUG_TMP_REGS        256'h000...   0000000000000002000000000000000500000000000000040000000000000005   0000000000000008000000000000000500000000000000040000000000000005
/tb_singlecyclecpu/tb_DEBUG_SAVED_REGS      256'h000...   0000000000000015000000000000000700000000000000070000000000000016
/tb_singlecyclecpu/tb_DEBUG_MEM_CONTENTS    256'h110...   1100000005000000000000000000000011000000000000001100000000000000
```

LSL X9, X9, 2

- Program Counter increments to 0x0000000000000028
- It fetches the 32-bit instruction in hexadecimal
- X9 being shift left by 2, which turns 0x002 into 0x008, and the value is stored in X9

Cycle 10:

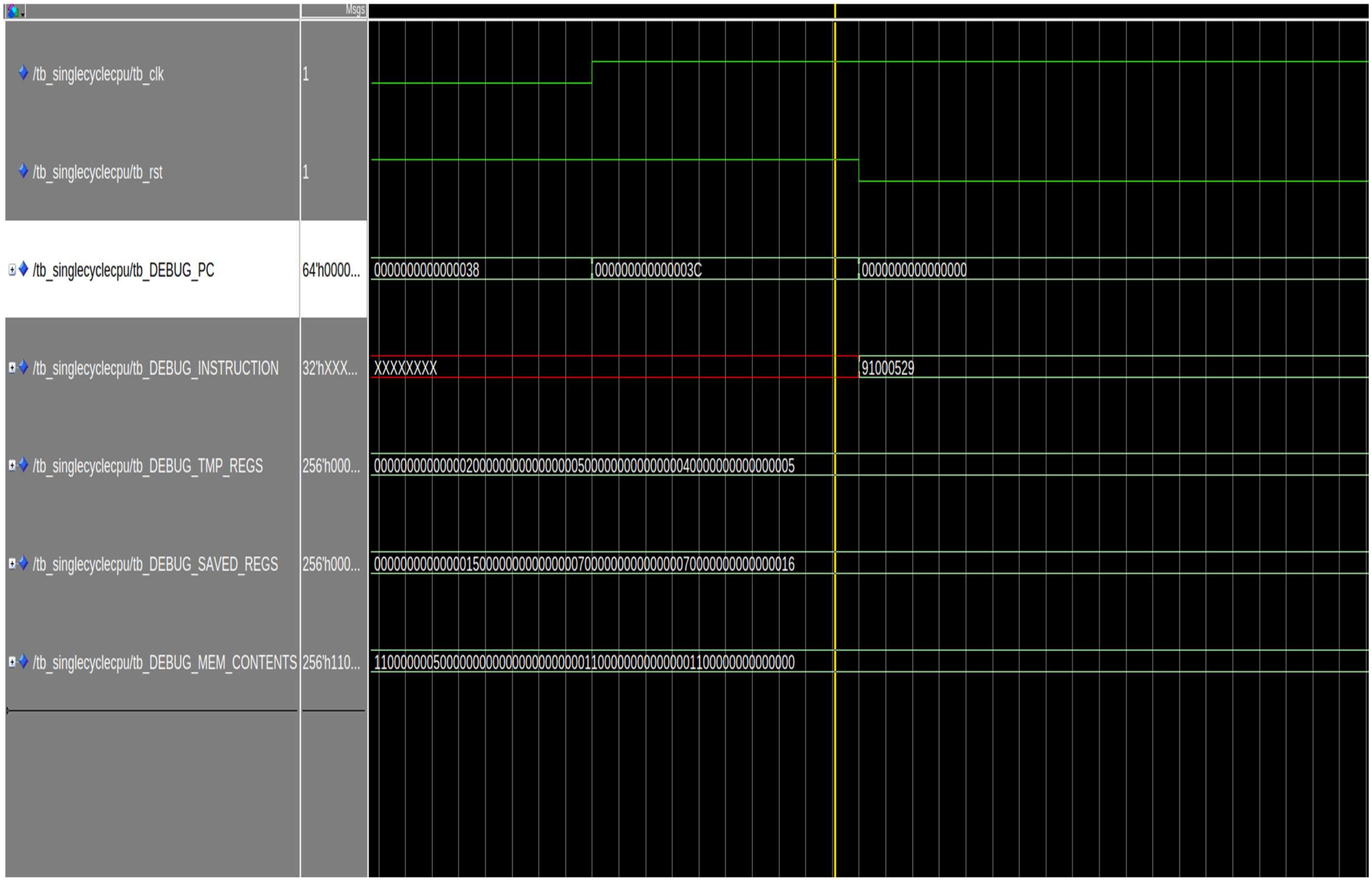


```
                                      Msgs
/tb_singlecyclecpu/tb_clk             0
/tb_singlecyclecpu/tb_rst             1
/tb_singlecyclecpu/tb_DEBUG_PC        64'h0000...  000000000000002C
/tb_singlecyclecpu/tb_DEBUG_INSTRUCTION 32'hD360... D3600929
/tb_singlecyclecpu/tb_DEBUG_TMP_REGS  256'h000...  0000000000000008000000000000000500000000000000040000000000000005   0000000000000002000000000000000500000000000000040000000000000005
/tb_singlecyclecpu/tb_DEBUG_SAVED_REGS 256'h000... 0000000000000015000000000000000700000000000000070000000000000016
/tb_singlecyclecpu/tb_DEBUG_MEM_CONTENTS 256'h110... 1100000005000000000000000000000011000000000000001100000000000000
```

LSR X9, X9 , 2

- Program Counter increments to 0x000000000000002C
- It fetches the 32-bit instruction in hexadecimal
- X9 being shift right by 2, which turns 0x008 into 0x002, and the value is stored in X9

Reset Cycle:



Reset:

- After finishing the instruction sets, I assert the reset signal and turn the program counter back to 0x0000000000000000.