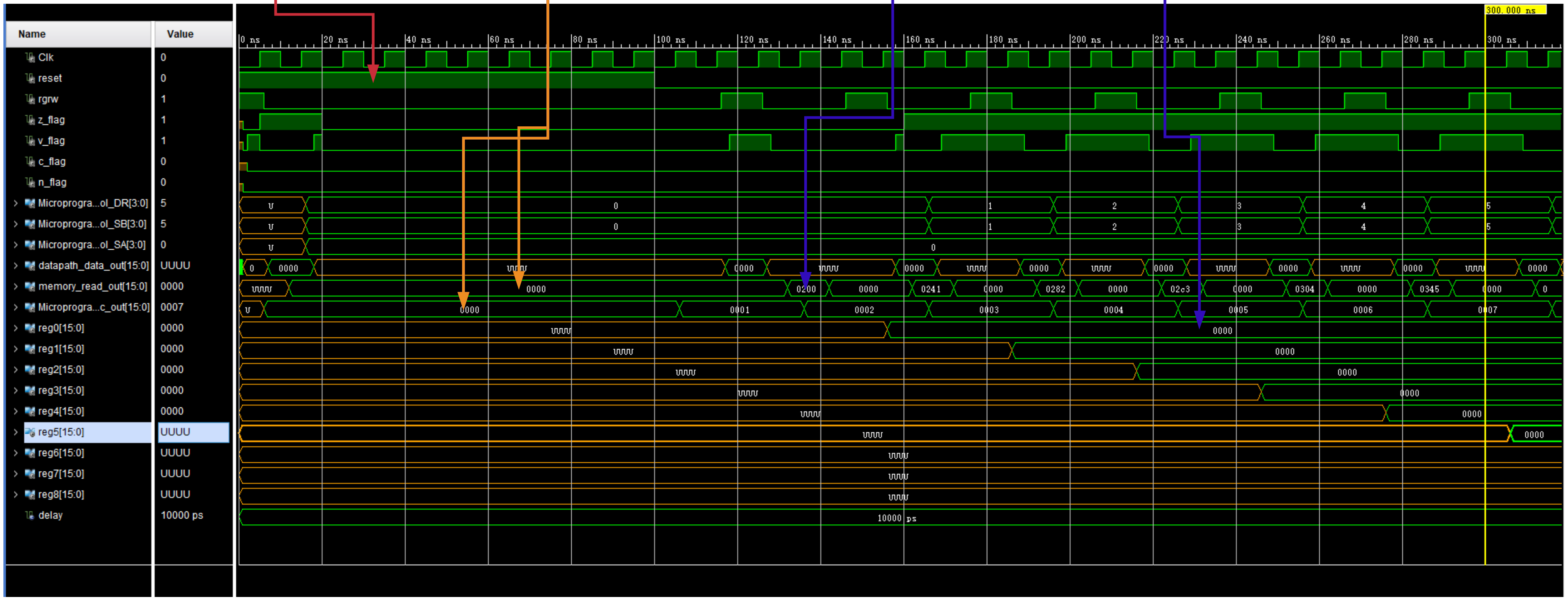
Instruction Processor Part 1: Reset

Reset

Memory Address: x"0000"
Instruction Opcode: x"0000"
Control Memory Address: x"00"

LD Rd, [Memory] for R0

LDR R0,=0



Name	Value
Clk	0
reset	0
rgrw	1
z_flag	1
v_flag	1
c_flag	0
n_flag	0
Micropogra...ol_DR[3:0]	5
Micropogra...ol_SB[3:0]	5
Micropogra...ol_SA[3:0]	0
datapath_data_out[15:0]	UUUU
memory_read_out[15:0]	0000
Micropogra...c_out[15:0]	0007
reg0[15:0]	0000
reg1[15:0]	0000
reg2[15:0]	0000
reg3[15:0]	0000
reg4[15:0]	0000
reg5[15:0]	UUUU
reg6[15:0]	UUUU
reg7[15:0]	UUUU
reg8[15:0]	UUUU
delay	10000 ps

Set up the instruction processor by doing reset to set the correct bit values in the control word, memory write, MD and etc.

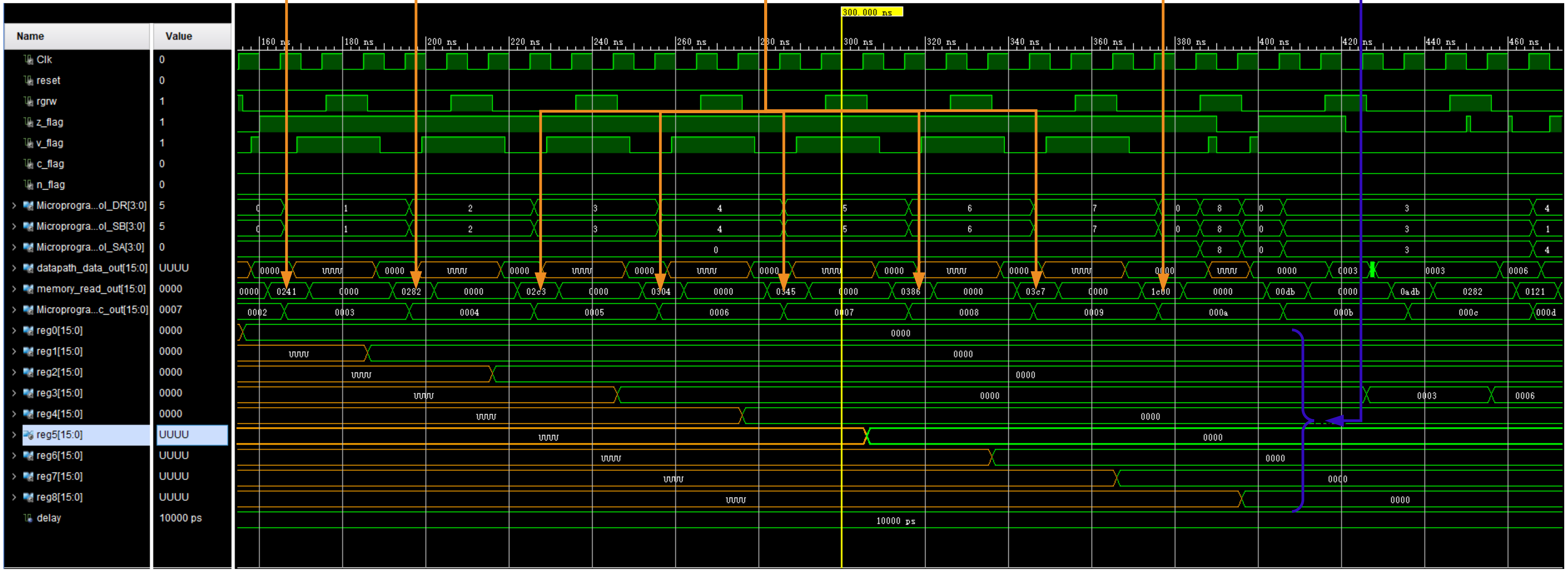
Then, do first instruction in memory load value 0 into all 9 registers.

Instruction Processor Part 2: LD

LD
Control Memory Code: 0xC02000C
Control Memory Address: 0x01

LD Rd,=0

LD Rd, [Memory] from R1-R8



Name	Value
Clk	0
reset	0
rgrw	1
z_flag	1
v_flag	1
c_flag	0
n_flag	0
Micropogra...ol_DR[3:0]	5
Micropogra...ol_SB[3:0]	5
Micropogra...ol_SA[3:0]	0
datapath_data_out[15:0]	UUUU
memory_read_out[15:0]	0000
Micropogra...c_out[15:0]	0007
reg0[15:0]	0000
reg1[15:0]	0000
reg2[15:0]	0000
reg3[15:0]	0000
reg4[15:0]	0000
reg5[15:0]	UUUU
reg6[15:0]	UUUU
reg7[15:0]	UUUU
reg8[15:0]	UUUU
delay	10000 ps

Do first instruction in memory load value from register 0 into all 9 registers. So, LD Rd, [R0]. As a result, all the registers ended up with value 0. Do a different LD for the temporary register, R since we need to set TD to enable it.

Instruction Processor Part 3: ADI & ADD

ADI
Control Memory Code: 0xC020224
Control Memory Address: 0x00
ADD
Control Memory Code: 0xC020024
Control Memory Address: 0x05

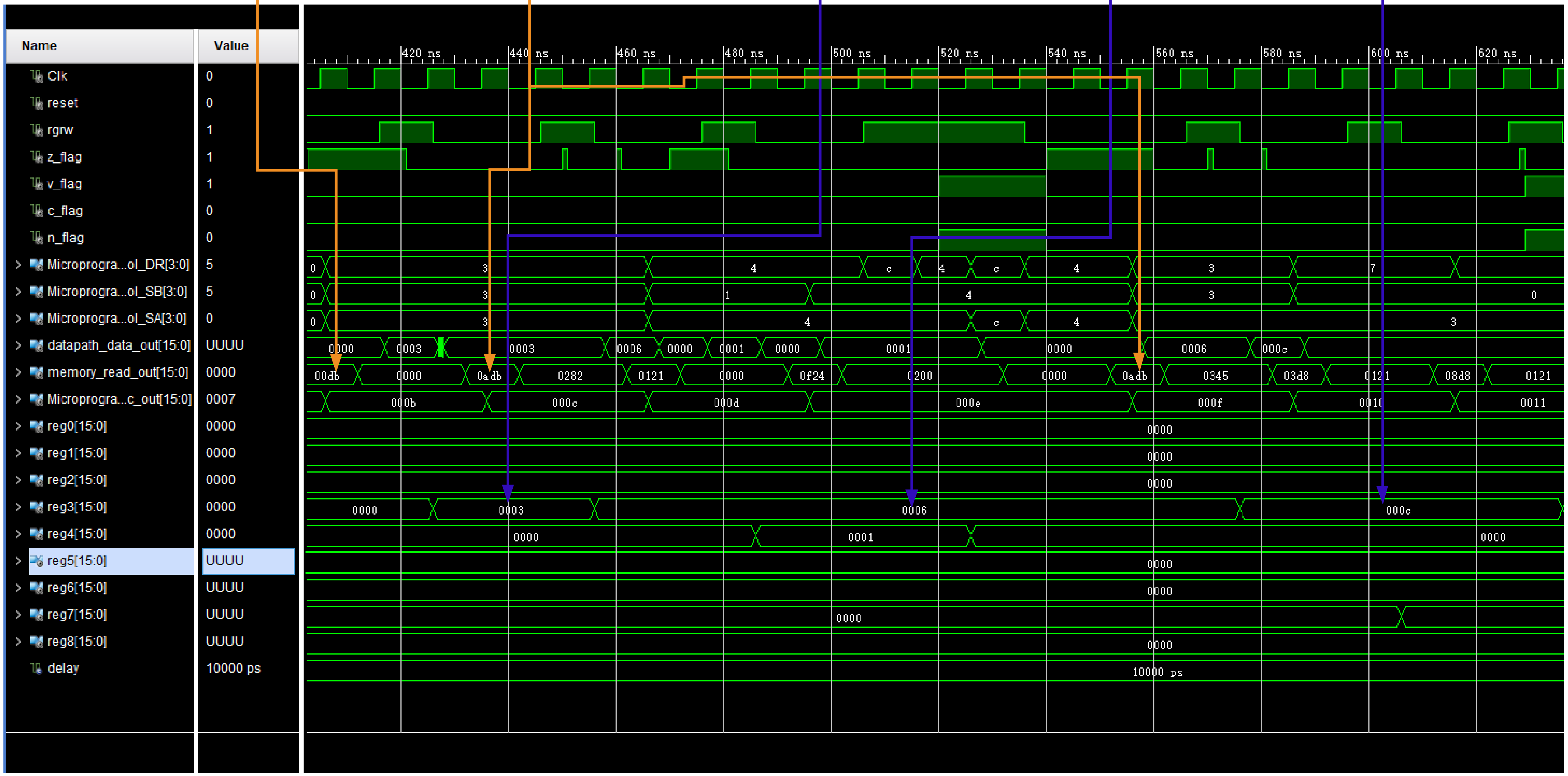
ADI R3, R3, #3

ADD R3, R3, R3

R3 = (0 + 3) = 3

R3 = (3 + 3) = 6

R3 = (6 + 6) = 12



Name	Value
Clk	0
reset	0
rgrw	1
z_flag	1
v_flag	1
c_flag	0
n_flag	0
Micropogra...ol_DR[3:0]	5
Micropogra...ol_SB[3:0]	5
Micropogra...ol_SA[3:0]	0
datapath_data_out[15:0]	UUUU
memory_read_out[15:0]	0000
Micropogra...c_out[15:0]	0007
reg0[15:0]	0000
reg1[15:0]	0000
reg2[15:0]	0000
reg3[15:0]	0000
reg4[15:0]	0000
reg5[15:0]	UUUU
reg6[15:0]	UUUU
reg7[15:0]	UUUU
reg8[15:0]	UUUU
delay	10000 ps

Do instruction ADI on R3 and the ADD instruction on R3 again. Initial result is 0. Final Result is 6.

Instruction Processor Part 4: LD, ADI then SR

ADI
Control Memory Code: 0xC020224
Control Memory Address: 0x00
SR
Control Memory Code: 0x87E1004
Control Memory Address: 0x07

LD
Control Memory Code: 0xC02000C
Control Memory Address: 0x01

ADI R4, R4, #1

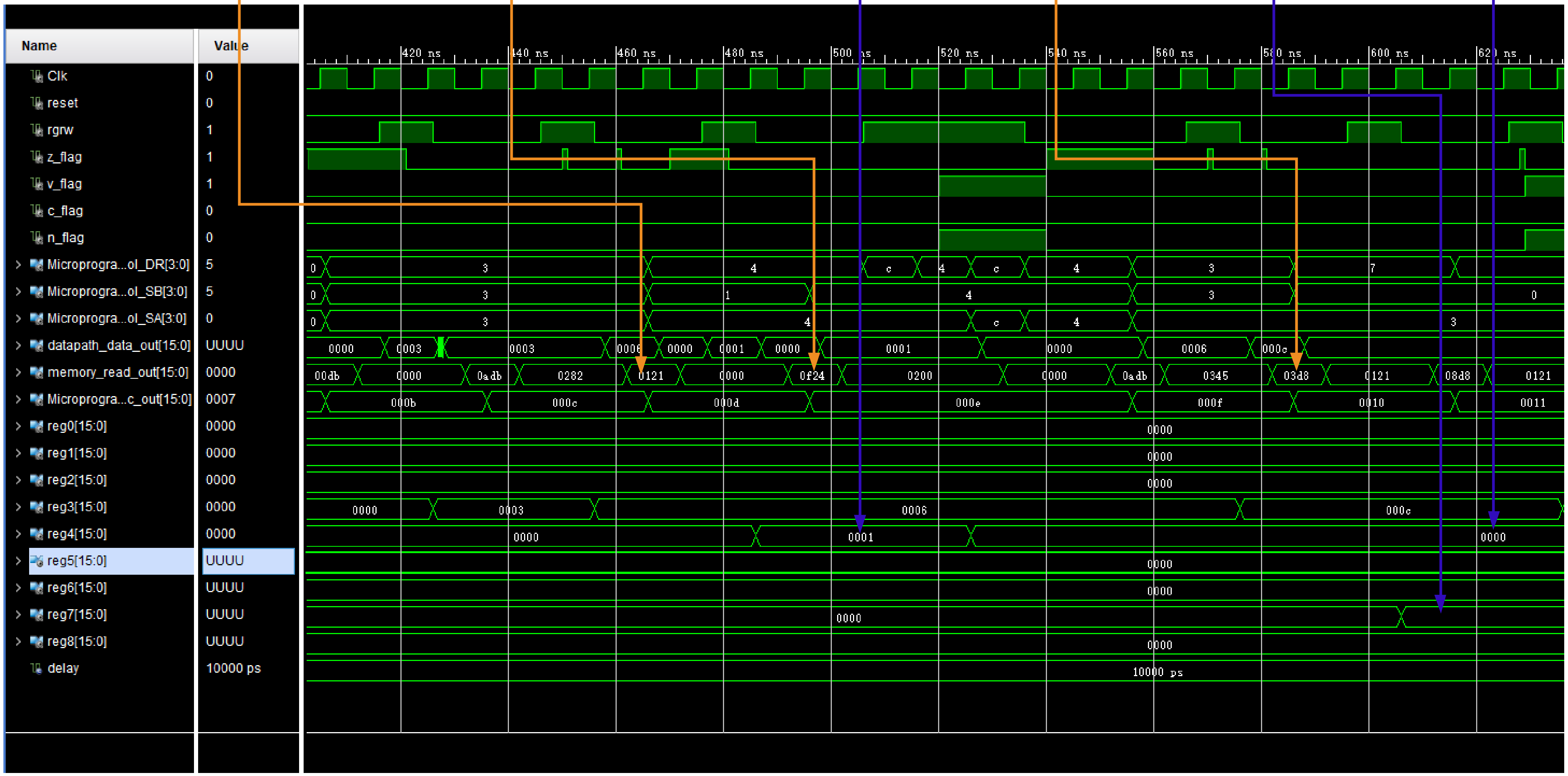
SR R4, R4, R4

R4 = (0 + 1) = 1

LD R7, Memory[R3]

R7 = Memory[12] = 0x0121

R4 = (1 >> 1) = 0x0



Name	Value
Clk	0
reset	0
rgrw	1
z_flag	1
v_flag	1
c_flag	0
n_flag	0
Micropogra...ol_DR[3:0]	5
Micropogra...ol_SB[3:0]	5
Micropogra...ol_SA[3:0]	0
datapath_data_out[15:0]	UUUU
memory_read_out[15:0]	0000
Micropogra...c_out[15:0]	0007
reg0[15:0]	0000
reg1[15:0]	0000
reg2[15:0]	0000
reg3[15:0]	0000
reg4[15:0]	0000
reg5[15:0]	UUUU
reg6[15:0]	UUUU
reg7[15:0]	UUUU
reg8[15:0]	UUUU
delay	10000 ps

Do instruction ADI on R4 and the SR instruction on R3 again. Initial result is 0, then 1, finally result is 0.

LD the memory value of index 12 (address value in R3) onto R7

Instruction Processor Part 5: NOT then BNZ and INC

NOT
Control Memory Code: 0xC0200E4
Control Memory Address: 0x04
INC
Control Memory Code: 0xC020014
Control Memory Address: 0x03

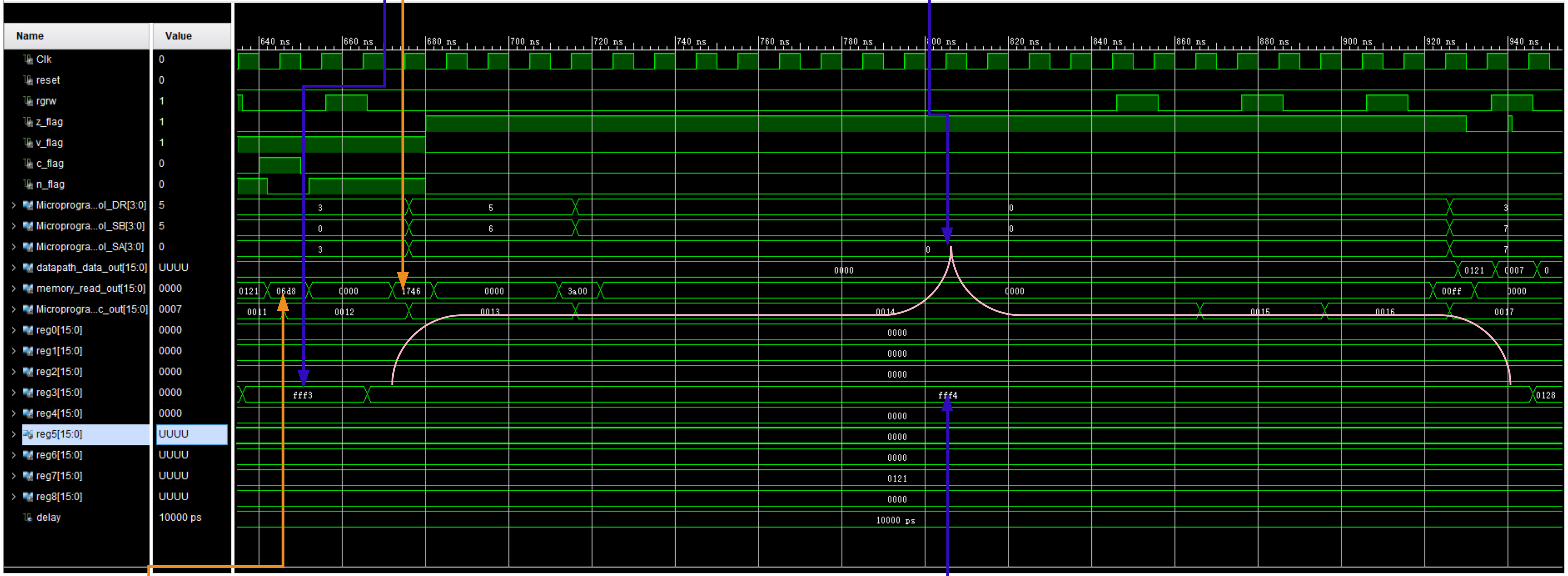
NOT R3, R3 <<in last screenshot>> memory read out is 0x08D8

R3 = (~12) = 0xFFF3

BNZ while

```
while(R3!=0) {
      (R3 = R3 + 1) ;
}
```

Will increment 12 times, until R3 becomes 0 and exit the loop, so 12 x (R3 = R3 + 1) = 0



Name	Value
Clk	0
reset	0
rgrw	1
z_flag	1
v_flag	1
c_flag	0
n_flag	0
Microprogra...ol_DR[3:0]	5
Microprogra...ol_SB[3:0]	5
Microprogra...ol_SA[3:0]	0
datapath_data_out[15:0]	UUUU
memory_read_out[15:0]	0000
Microprogra...c_out[15:0]	0007
reg0[15:0]	0000
reg1[15:0]	0000
reg2[15:0]	0000
reg3[15:0]	0000
reg4[15:0]	0000
reg5[15:0]	UUUU
reg6[15:0]	UUUU
reg7[15:0]	UUUU
reg8[15:0]	UUUU
delay	10000 ps

INC R3, R3

(R3 = R3 + 1) = 0xFFF4

Complement R3 , so it becomes negative. Then, we can perform BNZ (branch if not zero) and INC(increment)

Huge duration when BNZ is impelemented, because R3 is still not 0 and INC is being implemented until it reaches 0.

Instruction Processor Part 6: ADI & ADD to set N flag

ADI
Control Memory Code: 0xC020224
Control Memory Address: 0x00
NOT
Control Memory Code: 0xC0200E4
Control Memory Address: 0x04

ADD
Control Memory Code: 0xC020024
Control Memory Address: 0x05

ADI R1, R1, #5
ADI R2, R2, #6

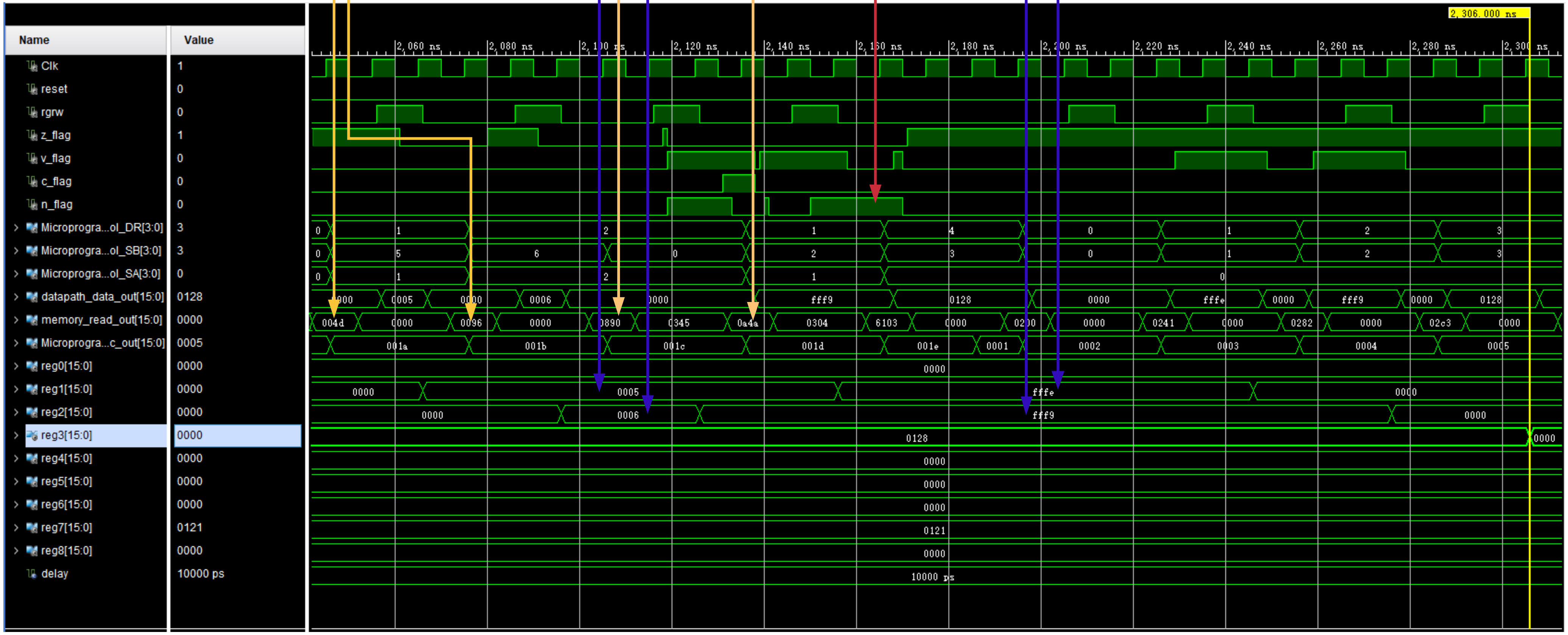
R1 = (0 + 5) = 5
R2 = (0 + 6) = 6

NOT R2, R2

R3 = (~12) = 0xFFF9

ADD R1, R1, R2

R1 = (5 + (-6)) = -1
<<set N flag>>



Name	Value
Clk	1
reset	0
rgrw	0
z_flag	1
v_flag	0
c_flag	0
n_flag	0
Microprogra...ol_DR[3:0]	3
Microprogra...ol_SB[3:0]	3
Microprogra...ol_SA[3:0]	0
datapath_data_out[15:0]	0128
memory_read_out[15:0]	0000
Microprogra...c_out[15:0]	0005
reg0[15:0]	0000
reg1[15:0]	0000
reg2[15:0]	0000
reg3[15:0]	0000
reg4[15:0]	0000
reg5[15:0]	0000
reg6[15:0]	0000
reg7[15:0]	0121
reg8[15:0]	0000
delay	10000 ps

Just doing addition to check for N flags. Add constant 5 and 6 to R1 and R2 , then complement R2 so that its is negative. Then add them together to get -1, N flag should be set.

Instruction Processor Part 7: Branch Unconditionally

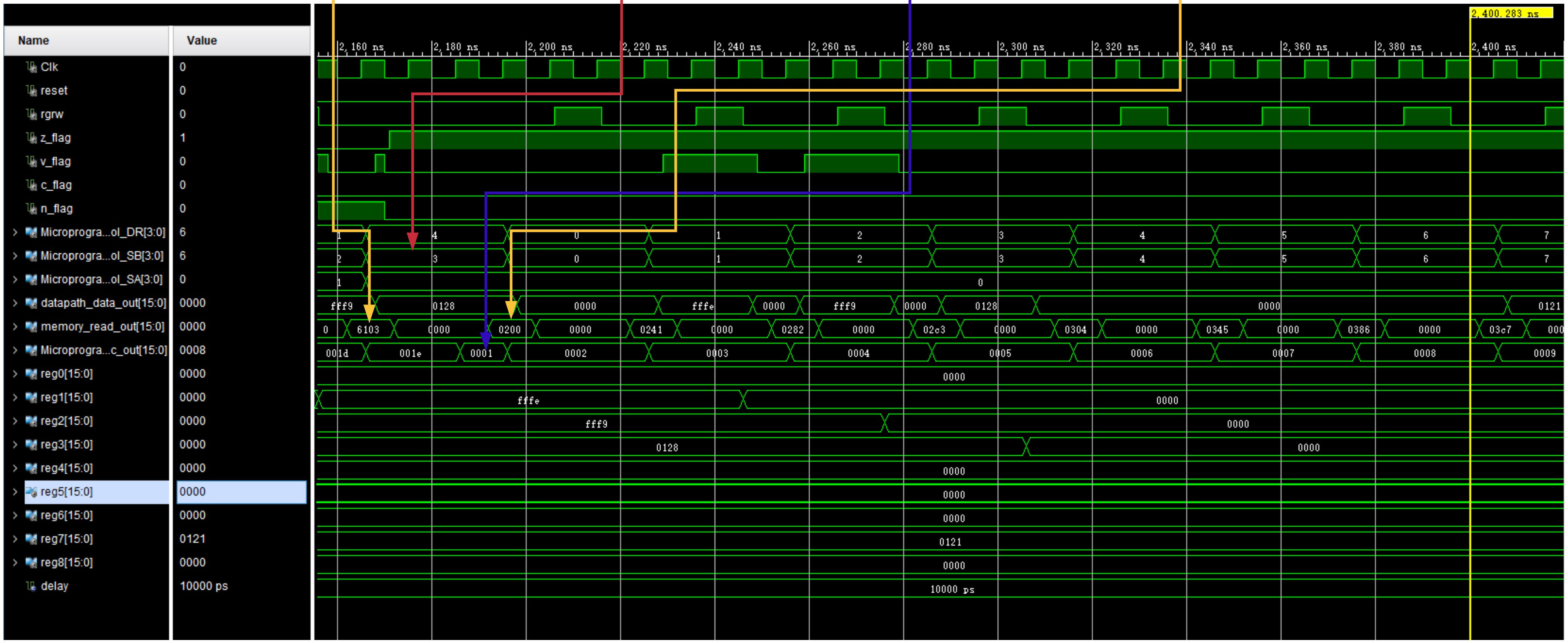
B
Control Memory Code: 0xC022000
Control Memory Address: 0x30

Branch unconditionally

SB=3; DR=4
PC extend = FFE3

PC should increment its counter, outputting a x"0001"

Do instruction in memory x"0001, which is x"0200"



Name	Value
Clk	0
reset	0
rgrw	0
z_flag	1
v_flag	0
c_flag	0
n_flag	0
Microprogra...ol_DR[3:0]	6
Microprogra...ol_SB[3:0]	6
Microprogra...ol_SA[3:0]	0
datapath_data_out[15:0]	0000
memory_read_out[15:0]	0000
Microprogra...c_out[15:0]	0008
reg0[15:0]	0000
reg1[15:0]	0000
reg2[15:0]	0000
reg3[15:0]	0000
reg4[15:0]	0000
reg5[15:0]	0000
reg6[15:0]	0000
reg7[15:0]	0121
reg8[15:0]	0000
delay	10000 ps

Branch unconditionally an address in memory and fecth it. In this case, branch back and redo instruction in x"0001", LD R0,=.0.