ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KỸ THUẬT MÁY TÍNH

BÁO CÁO LAB 04 & 05

MÔN THIẾT KỸ THUẬT THIẾT KẾ KIỂM TRA

HỌ VÀ TÊN: **Vòng Chí Cường – 21521910**

LỚP: **CE 409.O21**

GIẢNG VIÊN HƯỚNG DẪN

Phạm Thanh Hùng

TP. HỒ CHÍ MINH – Tháng 04 năm 2024

Mục Lục

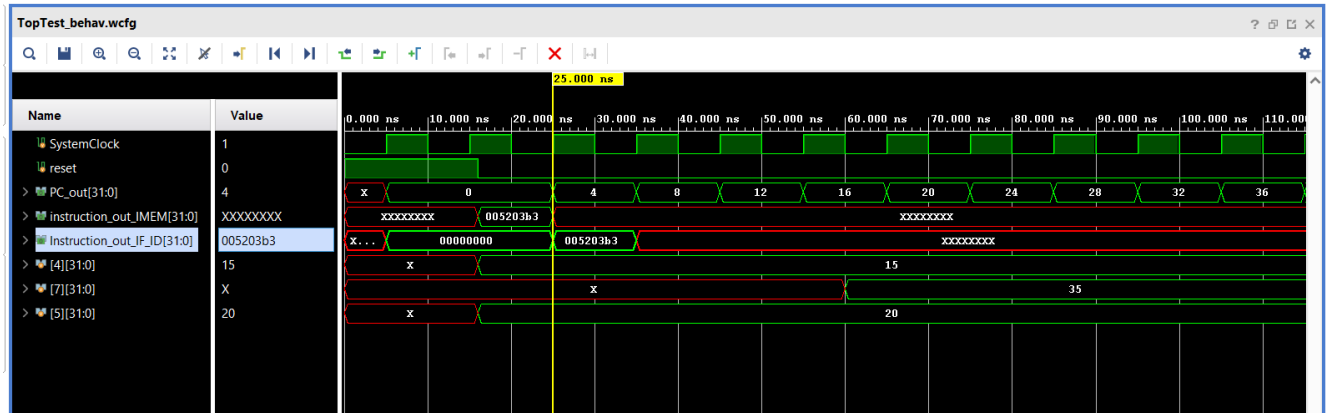
1. Test case

1.1. Nhóm lệnh R-type

1.1.1. Lệnh add

```
// Test case 01 : add
// add x7, x4, x5
// Đưa 15 vào x4, 20 vào x5
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.registerFile.Registers[5] = 20;
cpu.datapath.instructionMemory.Imemory[0] = 32'h005203b3;
```

Kết quả chạy mô phỏng :

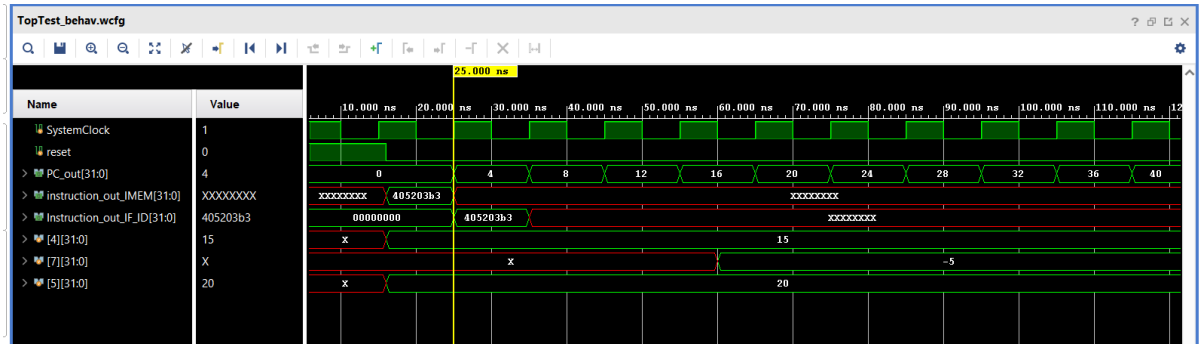


➔ Kể từ chu kỳ đầu tiên, trải qua 5 chu kỳ lệnh, Register file ghi vào x7 với giá trị là 35. Cpu hoạt động đúng như dự kiến.

1.1.2. Lệnh sub

```
// Test case 02 : sub
// sub x7, x4, x5
// Đưa 15 vào x4, 20 vào x5
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.registerFile.Registers[5] = 20;
cpu.datapath.instructionMemory.Imemory[0] = 32'h405203b3;
```

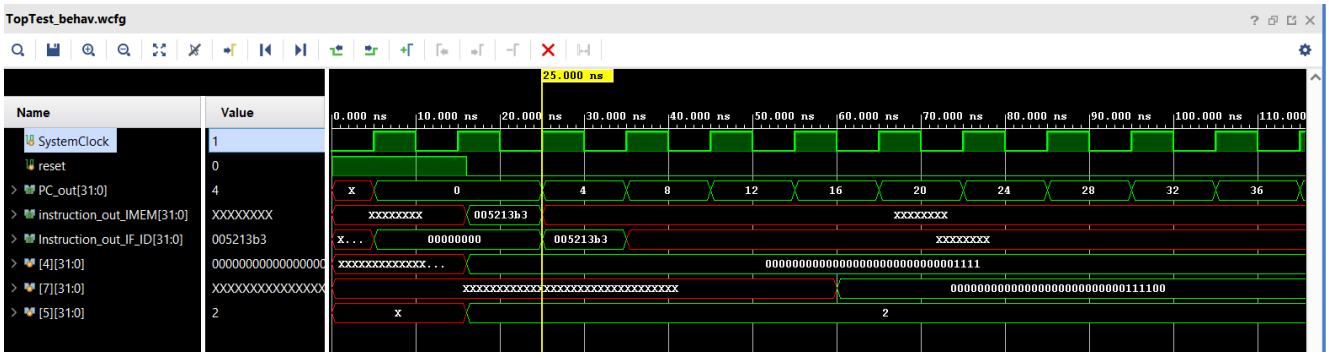
Kết quả chạy mô phỏng:



➔ Cpu hoạt động đúng như dự kiến, x7 = -5.

1.1.3. Lệnh sll

```
// Test case 03 : sll
// sll x7, x4, x5
// Đưa 15 vào x4, 2 vào x5
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.registerFile.Registers[5] = 20;
cpu.datapath.instructionMemory.Imemory[0] = 32'h005213b3;
```



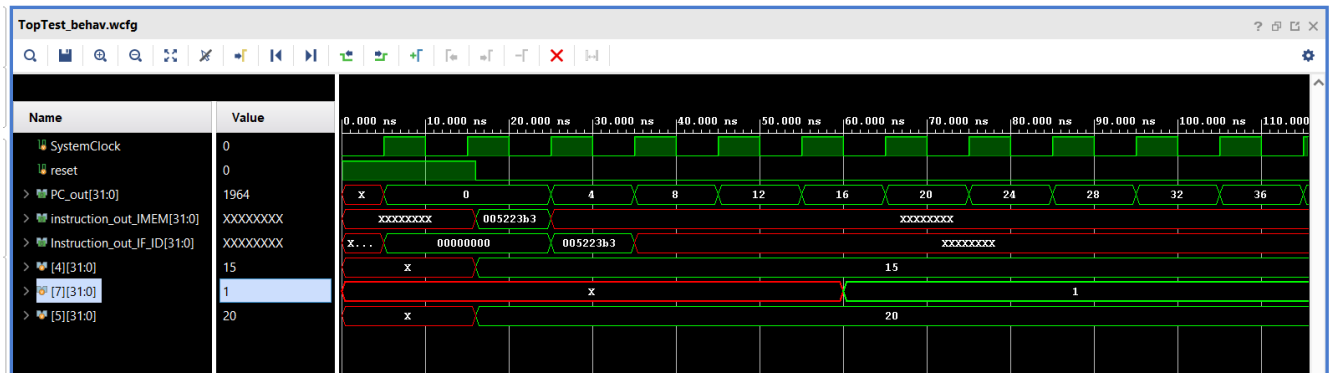
➔ Cpu hoạt động đúng như dự kiến, x7 = 60

1.1.4. Lệnh slt

TH1: x4 < x5

```
// Test case 04 : slt
// TH1 : x4 < x5
// slt x7, x4, x5
// Đưa 15 vào x4, 20 vào x5
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.registerFile.Registers[5] = 20;
cpu.datapath.instructionMemory.Imemory[0] = 32'h005223b3;
```

Kết quả chạy mô phỏng :

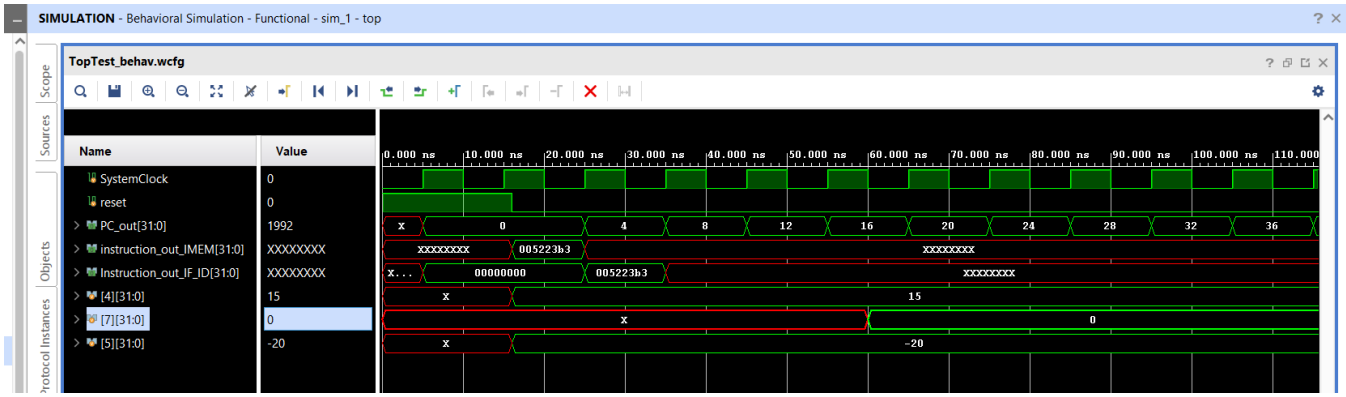


➔ Cpu hoạt động đúng như dự kiến, x7 = 1

TH2 : x4 > x5

```
// Test case 04 : slt
// Th2 : x4 > x5
// slt x7, x4, x5
// Đưa 15 vào x4, -20 vào x5
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.registerFile.Registers[5] = -20;
cpu.datapath.instructionMemory.Imemory[0] = 32'h005223b3;
```

Kết quả chạy mô phỏng :

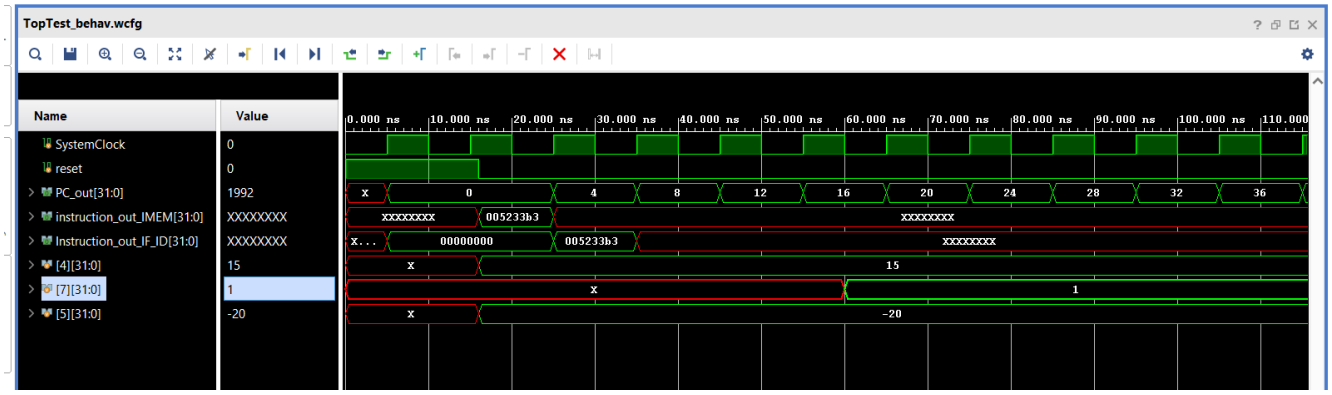


➔ Cpu hoạt động đúng như dự kiến, x7 = 0

1.1.5. Lệnh sltu

```
// Test case 05 : sltu
// sltu x7, x4, x5
// Đưa 15 vào x4, -20 vào x5
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.registerFile.Registers[5] = -20;
cpu.datapath.instructionMemory.Imemory[0] = 32'h005233b3;
```

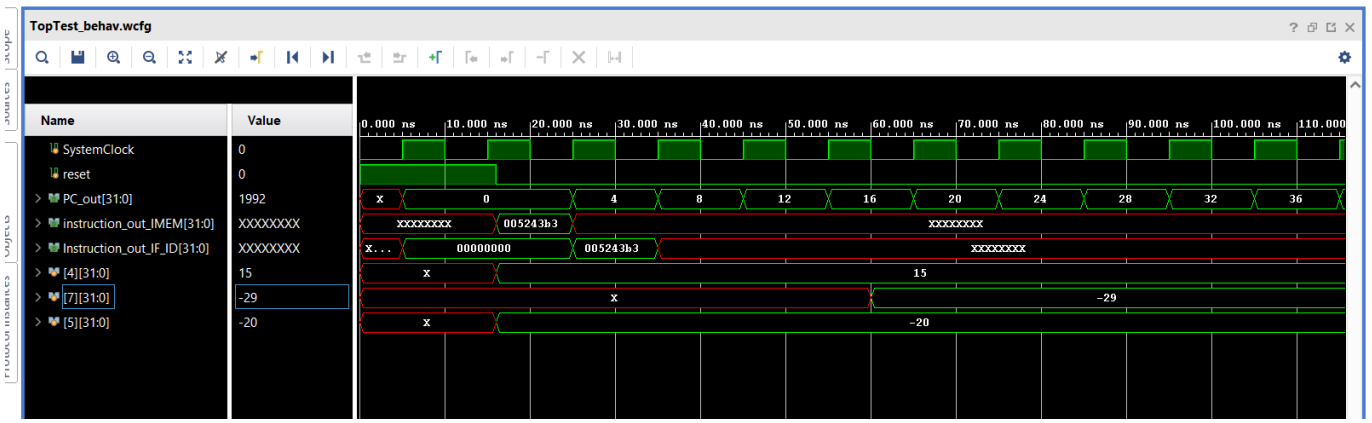
Kết quả chạy mô phỏng :



➔ Ta có x4 > x5 (15 > -20) nhưng dùng lệnh sltu nên x4 < x5, từ đó x7 = 1. Cpu hoạt động đúng như dự kiến.

1.1.6. Lệnh xor

```
// Test case 06 : xor
// xor x7, x4, x5
// Đưa 15 vào x4, -20 vào x5
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.registerFile.Registers[5] = -20;
cpu.datapath.instructionMemory.Imemory[0] = 32'h005243b3;
```

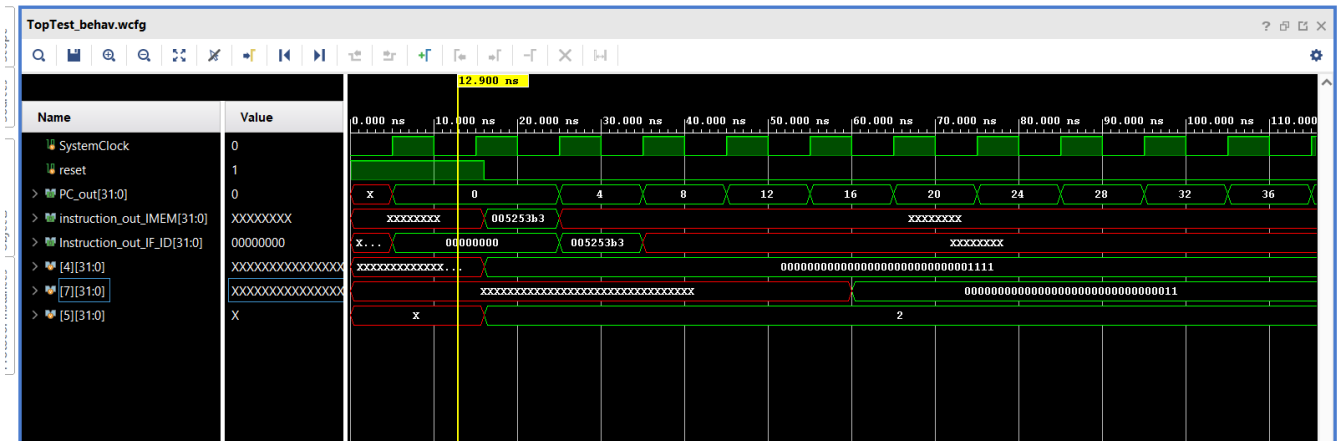


➔ Cpu hoạt động đúng như dự kiến, x7 = -29

1.1.7. Lệnh srl

```
// Test case 07 : srl
// srl x7, x4, x5
// Đưa 15 vào x4, 2 vào x5
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.registerFile.Registers[5] = 2;
cpu.datapath.instructionMemory.Imemory[0] = 32'h005253b3;
```

Kết quả chạy mô phỏng :

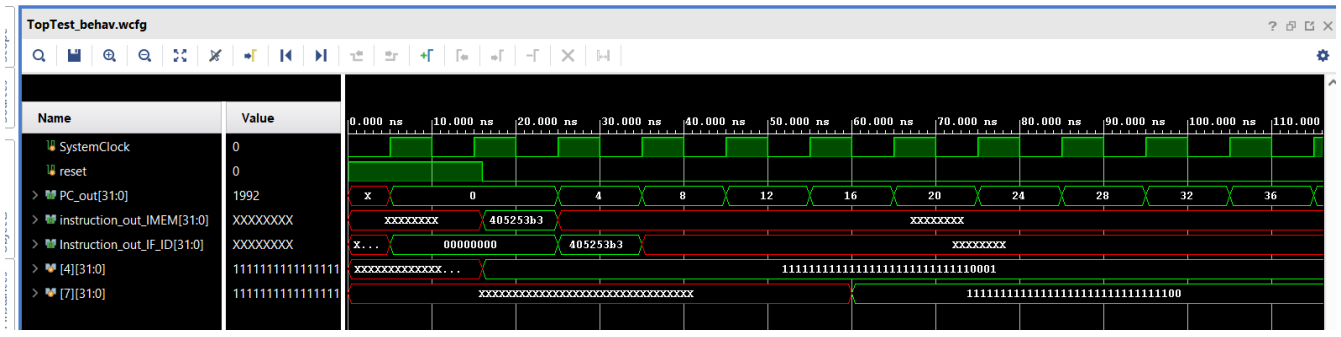


➔ Cpu hoạt động đúng như dự kiến, x7 = 3

1.1.8. Lệnh sra

```
// Test case 08 : sra
// sra x7, x4, x5
// Đưa -15 vào x4, 2 vào x5
cpu.datapath.registerFile.Registers[4] = -15;
cpu.datapath.registerFile.Registers[5] = 2;
cpu.datapath.instructionMemory.Imemory[0] = 32'h405253b3;
```

Kết quả chạy mô phỏng :



➔ Cpu hoạt động đúng như dự kiến.

1.1.9. Lệnh or

```
```

```
// Test case 09 : or
// or x7, x4, x5
// Đưa 15 vào x4, 27 vào x5
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.registerFile.Registers[5] = 27;
cpu.datapath.instructionMemory.Imemory[0] = 32'h005263b3;
```

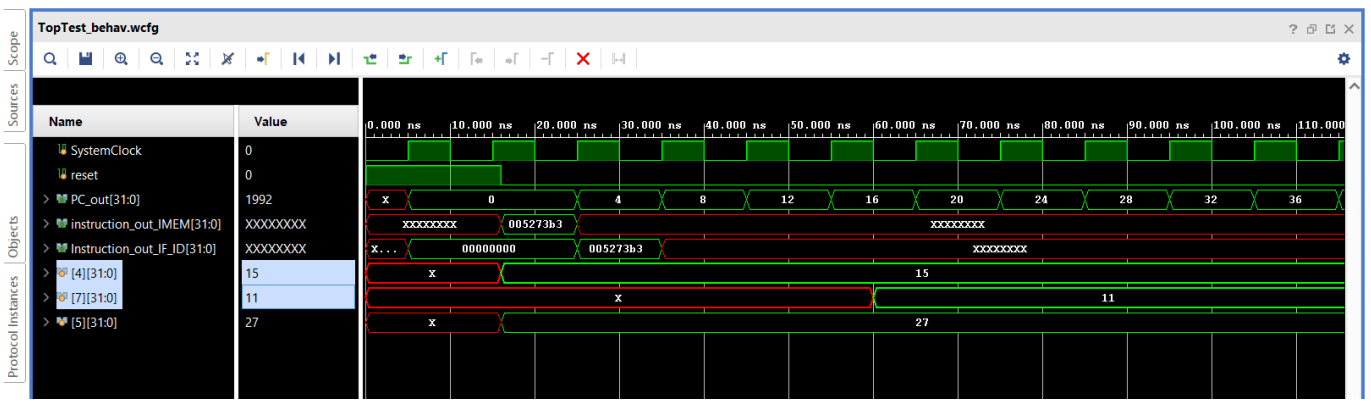
Kết quả chạy mô phỏng :

➔ Cpu hoạt động đúng như dự kiến, x7 = 31.

1.1.10.Lệnh and

```
// Test case 10 : and
// and x7, x4, x5
// Đưa 15 vào x4, 27 vào x5
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.registerFile.Registers[5] = 27;
cpu.datapath.instructionMemory.Imemory[0] = 32'h005273b3;
```

Kết quả chạy mô phỏng :



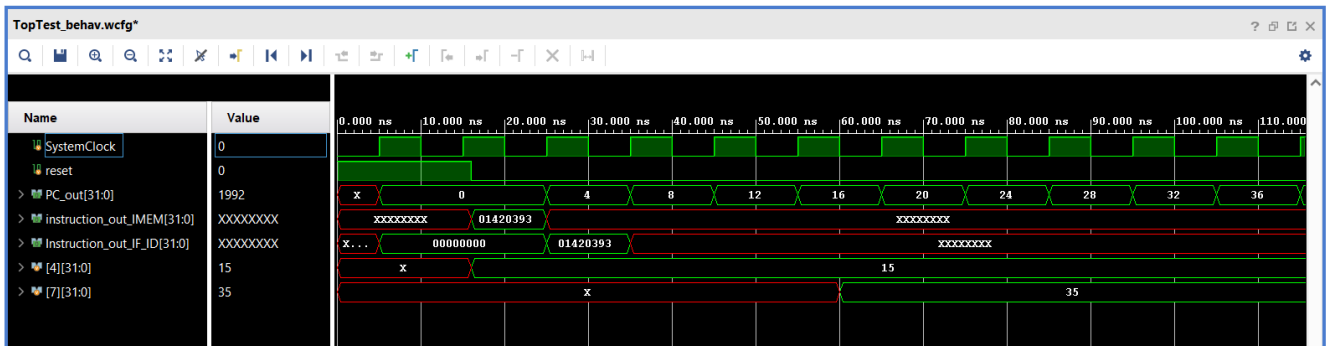
➔ Cpu hoạt động đúng như dự kiến, x7 = 11

1.2. Nhóm lệnh I-type

1.2.1. Lệnh addi

```
// Test case 11 : addi
// addi x7, x4, 20
// Đưa 15 vào x4
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.instructionMemory.Imemory[0] = 32'h01420393;
```

Kết quả chạy mô phỏng :



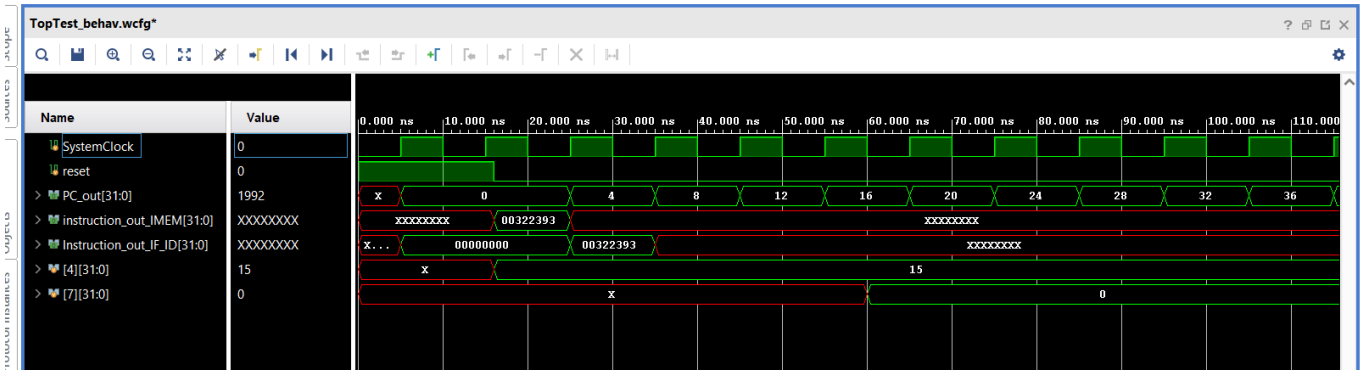
➔ Cpu hoạt động đúng như dự kiến, x7 = 35

1.2.2. Lệnh slti

TH1 : x4 > Immediate

```
// Test case 12 : slti
// TH1 : x4 > 3
// slti x7, x4, 3
// Đưa 15 vào x4
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.instructionMemory.Imemory[0] = 32'h00322393;
```

Kết quả chạy mô phỏng :

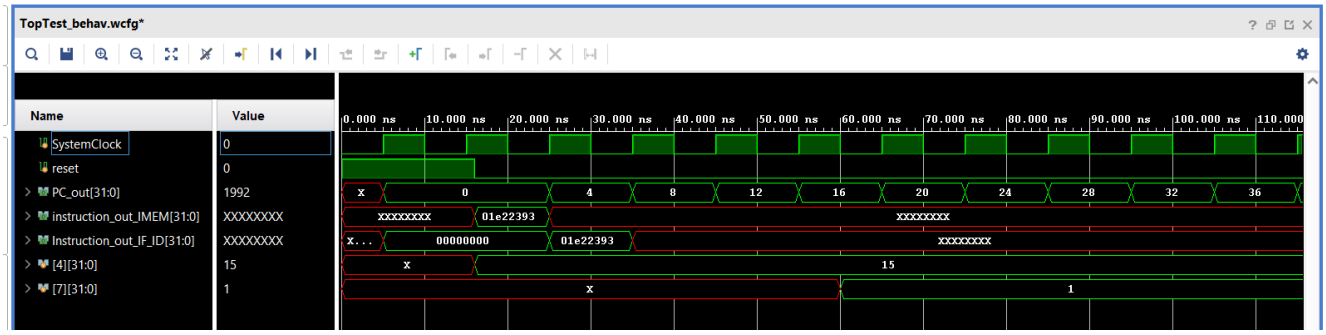


➔ Vì x4 = 15 > 3, nên x7 = 0. Cpu hoạt động đúng như dự kiến.

TH2 : x4 < Immediate

```
// Test case 12 : slti
// TH2 : x4 < 30
// slti x7, x4, 30
// Đưa 15 vào x4
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.instructionMemory.Imemory[0] = 32'h01e22393;
```

Kết quả chạy mô phỏng :

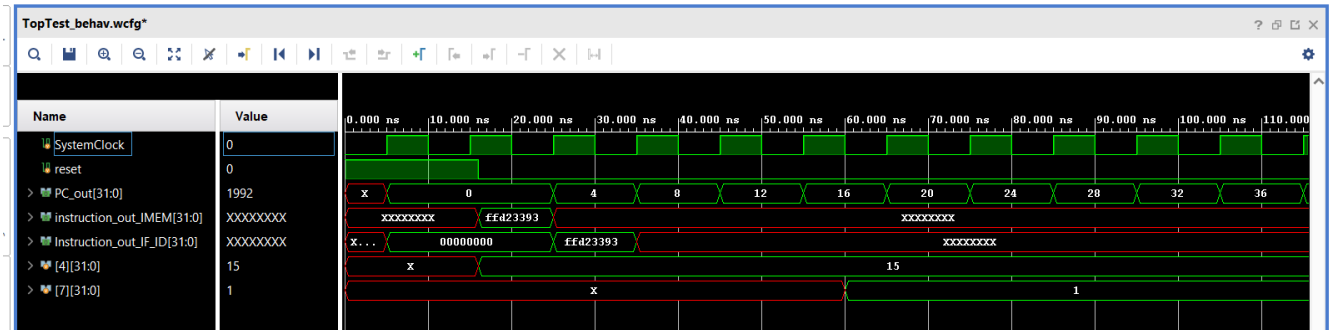


➔ Vì x4 = 15 < 30, nên x7 = 1. Cpu hoạt động đúng như dự kiến.

1.2.3. Lệnh sltiu

```
// Test case 13 : sltiu
// sltiu x7, x4, -3
// Đưa 15 vào x4
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.instructionMemory.Imemory[0] = 32'hffd23393;
```

Kết quả chạy mô phỏng :



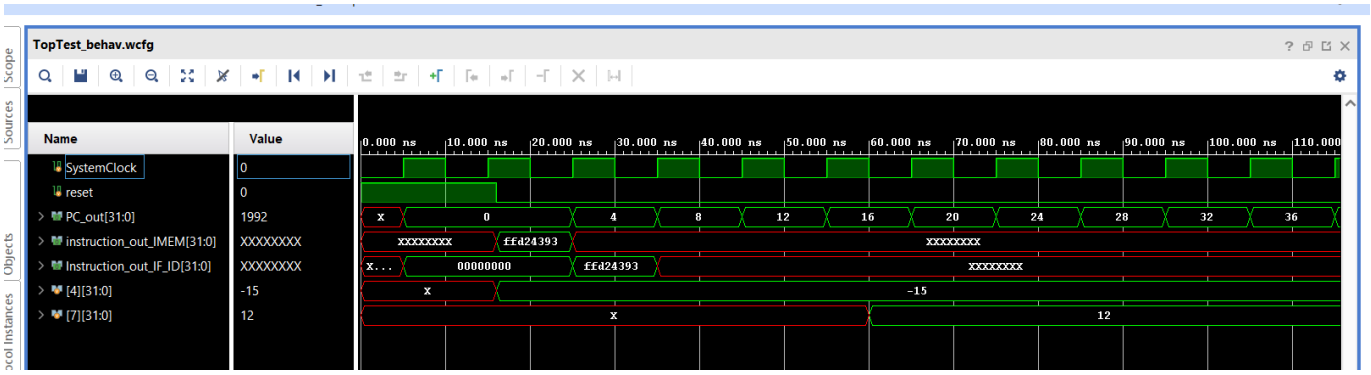
➔ Ta có : x4 = 15 > -3, nhưng vì dùng lệnh sltiu nên không xét bit dấu, x7 = 1. Cpu hoạt động đúng như dự kiến.

1.2.4. Lệnh xori

```
// Test case 14 : xori
```

```
// xori x7, x4, -3
// Đưa -15 vào x4
cpu.datapath.registerFile.Registers[4] = -15;
cpu.datapath.instructionMemory.Imemory[0] = 32'hffd24393;
```

Kết quả chạy mô phỏng :

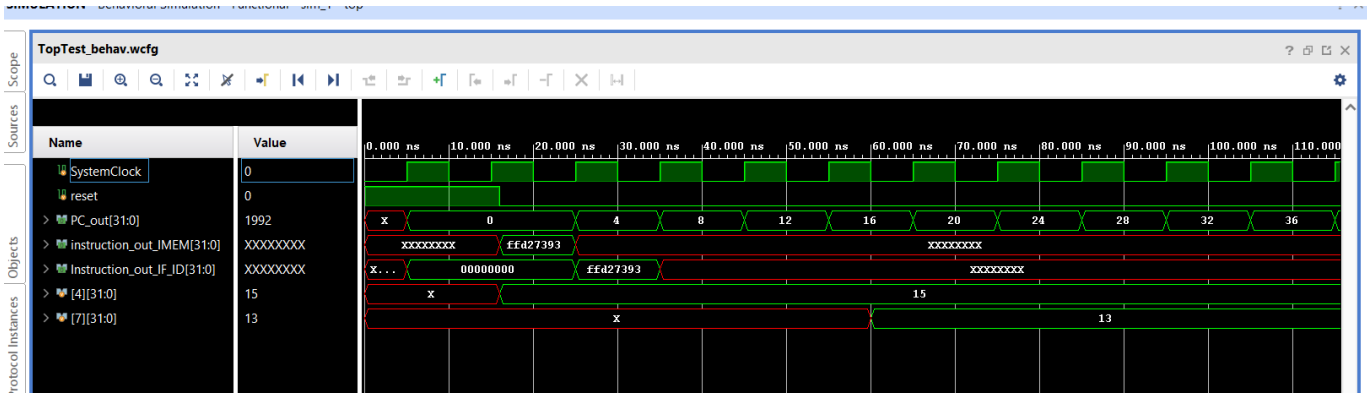


➔ Ta có : x4 = -15 xor (-3), x7 = 12. Cpu hoạt động đúng như dự kiến.

1.2.5. Lệnh andi

```
// Test case 15 : andi
// andi x7, x4, -3
// Đưa 15 vào x4
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.instructionMemory.Imemory[0] = 32'hffd27393;
```

Kết quả chạy mô phỏng :

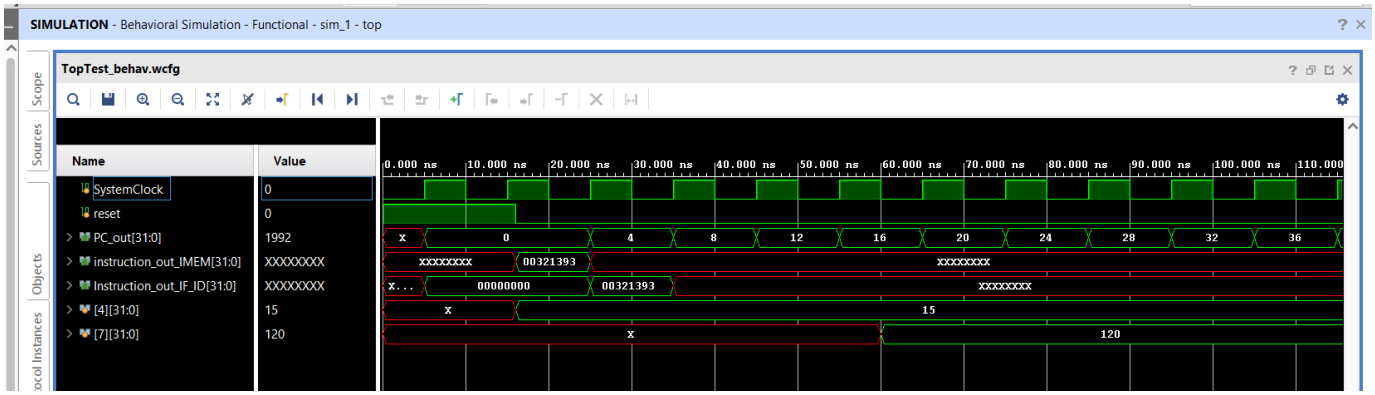


➔ Ta có : x4 = 15 and (-3), x7 = 13. Cpu hoạt động đúng như dự kiến.

1.2.6. Lệnh slli

```
// Test case 16 : slli
// slli x7, x4, 3
// Đưa 15 vào x4
cpu.datapath.registerFile.Registers[4] = 15;
cpu.datapath.instructionMemory.Imemory[0] = 32'h00321393;
```

Kết quả chạy mô phỏng :

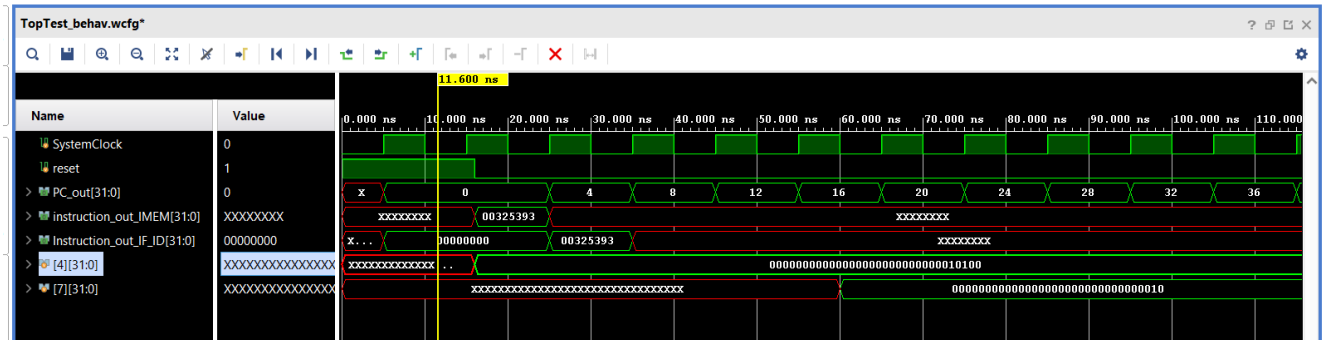


➔ Ta có : x7 = 120. Cpu hoạt động đúng như dự kiến.

1.2.7. Lệnh srli

```
// Test case 17 : srli
// srli x7, x4, 3
// Đưa 20 vào x4
cpu.datapath.registerFile.Registers[4] = 20;
cpu.datapath.instructionMemory.Imemory[0] = 32'h00325393;
```

Kết quả chạy mô phỏng :

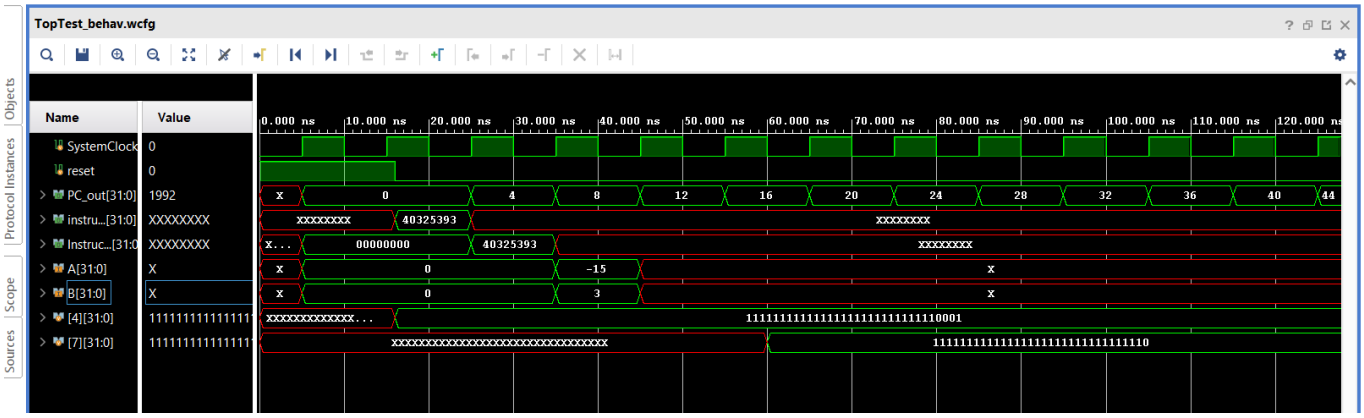


➔ Ta có : x7 = 2. Cpu hoạt động đúng như dự kiến.

1.2.8. Lệnh srai

```
// Test case 18 : srai
// srai x7, x4, 3
// Đưa 15 vào x4
cpu.datapath.registerFile.Registers[4] = -20;
cpu.datapath.instructionMemory.Imemory[0] = 32'h40325393;
```

Kết quả chạy mô phỏng :



➔ Cpu hoạt động đúng như dự kiến.

1.3. Nhóm lệnh B-type

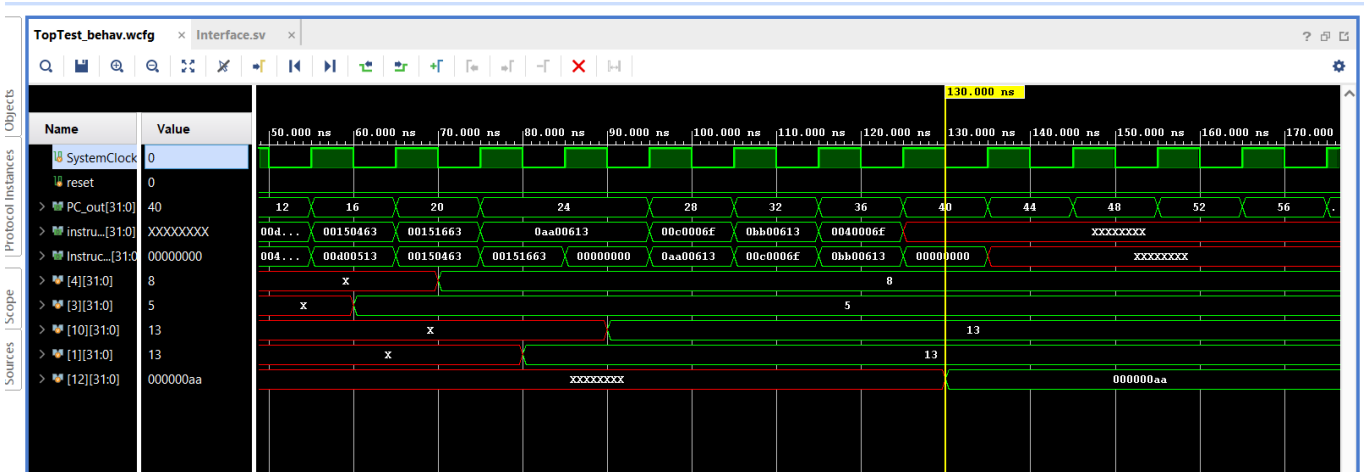
1.3.1. Lệnh beq, bne

TH beq: x1 = x10 = 13

```
/* Test case 19 : beq, bne
addi x3, x0 ,5
addi x4, x0, 8
add x1, x3, x4
addi x10, x0, 13
beq x10, x1, correct
bne x10, x1, fail
correct:
    addi x12, zero, 0xaa
    j end
fail:
    addi x12, zero, 0xbb
    j end
end
*/
```

```
cpu.datapath.instructionMemory.Imemory[0] = 32'h00500193;
cpu.datapath.instructionMemory.Imemory[1] = 32'h00800213;
cpu.datapath.instructionMemory.Imemory[2] = 32'h004180b3;
cpu.datapath.instructionMemory.Imemory[3] = 32'h00d00513;
cpu.datapath.instructionMemory.Imemory[4] = 32'h00150463;
cpu.datapath.instructionMemory.Imemory[5] = 32'h00151663;
cpu.datapath.instructionMemory.Imemory[6] = 32'h0aa00613;
cpu.datapath.instructionMemory.Imemory[7] = 32'h00c0006f;
cpu.datapath.instructionMemory.Imemory[8] = 32'h0bb00613;
cpu.datapath.instructionMemory.Imemory[9] = 32'h0040006f;
```

Kết quả chạy mô phỏng :



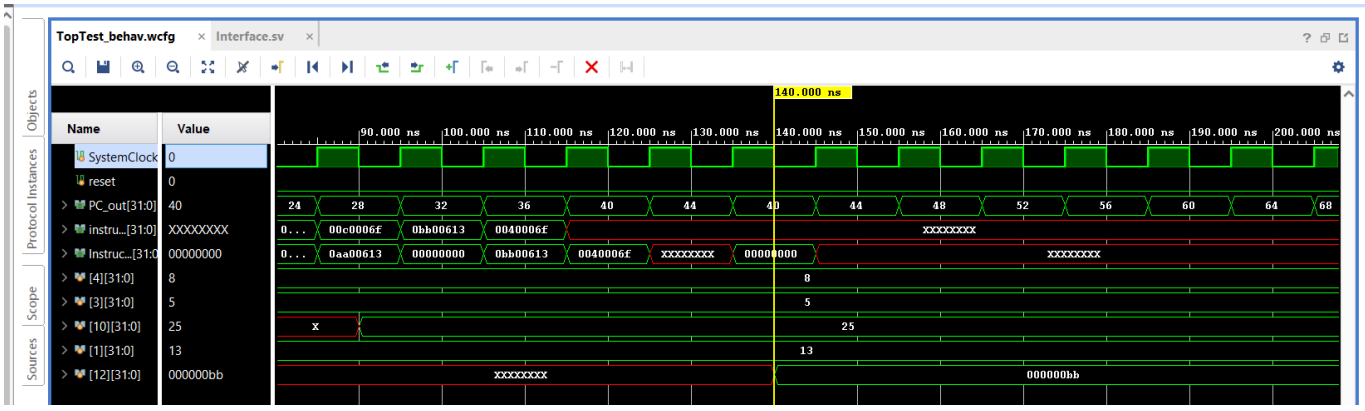
➔ Ta có : x12 = 0xaa và x10 = x1 = 13. Cpu hoạt động đúng như dự kiến.

TH bne : x1 = 13, x10 = 25

```
/* Test case 19 : beq, bne
addi x3, x0 ,5
addi x4, x0, 8
add x1, x3, x4
```

```
addi x10, x0, 25
beq x10, x1, correct
bne x10, x1, fail
correct:
    addi x12, zero, 0xaa
    j end
fail:
    addi x12, zero, 0xbb
    j end
end:
*/
cpu.datapath.instructionMemory.Imemory[0] = 32'h00500193;
cpu.datapath.instructionMemory.Imemory[1] = 32'h00800213;
cpu.datapath.instructionMemory.Imemory[2] = 32'h004180b3;
cpu.datapath.instructionMemory.Imemory[3] = 32'h01900513;
cpu.datapath.instructionMemory.Imemory[4] = 32'h00150463;
cpu.datapath.instructionMemory.Imemory[5] = 32'h00151663;
cpu.datapath.instructionMemory.Imemory[6] = 32'h0aa00613;
cpu.datapath.instructionMemory.Imemory[7] = 32'h00c0006f;
cpu.datapath.instructionMemory.Imemory[8] = 32'h0bb00613;
cpu.datapath.instructionMemory.Imemory[9] = 32'h0040006f;
```

Kết quả chạy mô phỏng :



➔ Ta có : x12 = 0xbb ,x10 = 25, x1 = 13. Cpu hoạt động đúng như dự kiến.

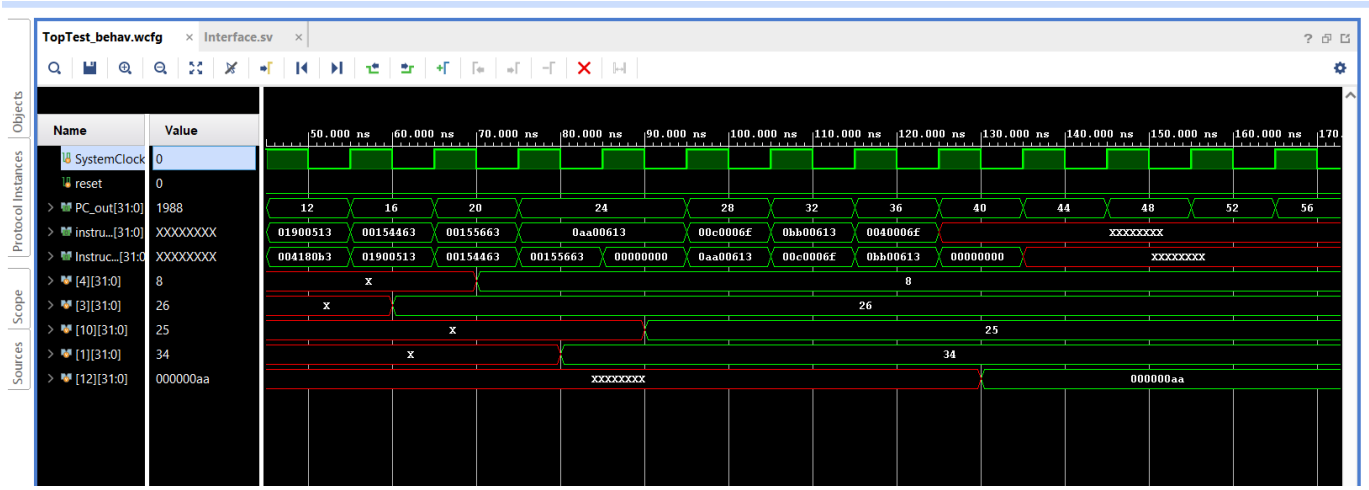
1.3.2. Lệnh blt, bge

TH blt : x10 = 25, x1 = 34

```
/* Test case 20 : blt
addi x3, x0 ,26
addi x4, x0, 8
add x1, x3, x4
addi x10, x0, 25
blt x10, x1, correct
bge x10, x1, fail
correct:
   addi x12, zero, 0xaa
   j end
fail:
   addi x12, zero, 0xbb
   j end
end:
*/
cpu.datapath.instructionMemory.Imemory[0] = 32'h00500193;
cpu.datapath.instructionMemory.Imemory[1] = 32'h00800213;
cpu.datapath.instructionMemory.Imemory[2] = 32'h004180b3;
cpu.datapath.instructionMemory.Imemory[3] = 32'h01900513;
cpu.datapath.instructionMemory.Imemory[4] = 32'h00154463;
cpu.datapath.instructionMemory.Imemory[5] = 32'h00155663;
cpu.datapath.instructionMemory.Imemory[6] = 32'h0aa00613;
cpu.datapath.instructionMemory.Imemory[7] = 32'h00c0006f;
cpu.datapath.instructionMemory.Imemory[8] = 32'h0bb00613;
```

```
cpu.datapath.instructionMemory.Imemory[9] = 32'h0040006f;
```

Kết quả chạy mô phỏng :



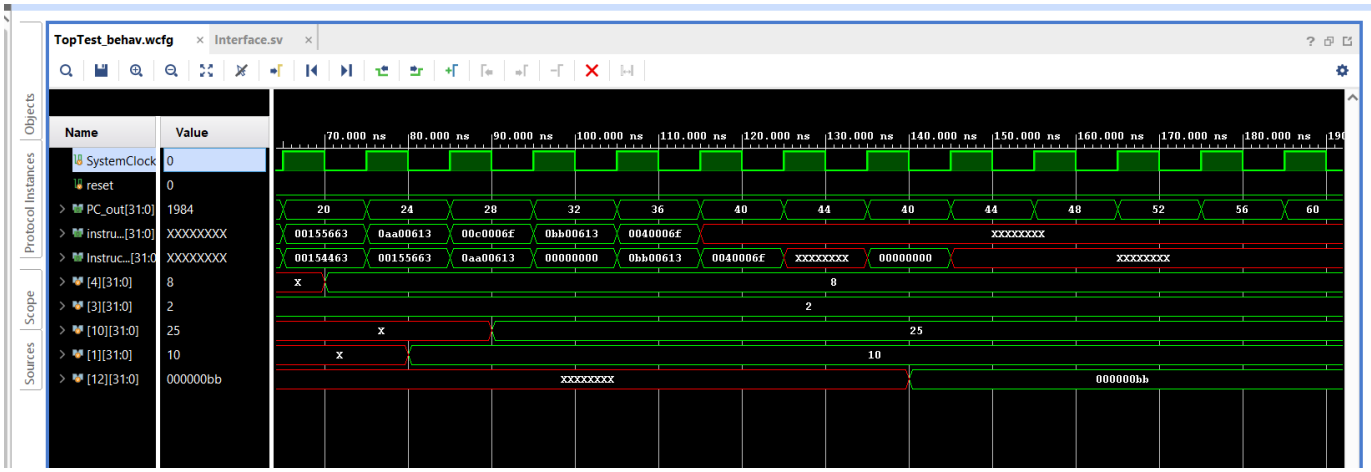
➔ Ta có : x12 = 0xaa, x10 = 25, x1 = 34. Cpu hoạt động đúng như dự kiến.

TH bge : x10 = 25, x1 = 10

```
/* Test case 20 : blt
addi x3, x0 ,2
addi x4, x0, 8
add x1, x3, x4
addi x10, x0, 25
blt x10, x1, correct
bge x10, x1, fail
correct:
    addi x12, zero, 0xaa
    j end
fail:
    addi x12, zero, 0xbb
    j end
```

```
end:
*/
cpu.datapath.instructionMemory.Imemory[0] = 32'h00200193;

cpu.datapath.instructionMemory.Imemory[1] = 32'h00800213;
cpu.datapath.instructionMemory.Imemory[2] = 32'h004180b3;
cpu.datapath.instructionMemory.Imemory[3] = 32'h01900513;
cpu.datapath.instructionMemory.Imemory[4] = 32'h00154463;
cpu.datapath.instructionMemory.Imemory[5] = 32'h00155663;
cpu.datapath.instructionMemory.Imemory[6] = 32'h0aa00613;
cpu.datapath.instructionMemory.Imemory[7] = 32'h00c0006f;
cpu.datapath.instructionMemory.Imemory[8] = 32'h0bb00613;
cpu.datapath.instructionMemory.Imemory[9] = 32'h0040006f;
```

Kết quả chạy mô phỏng :



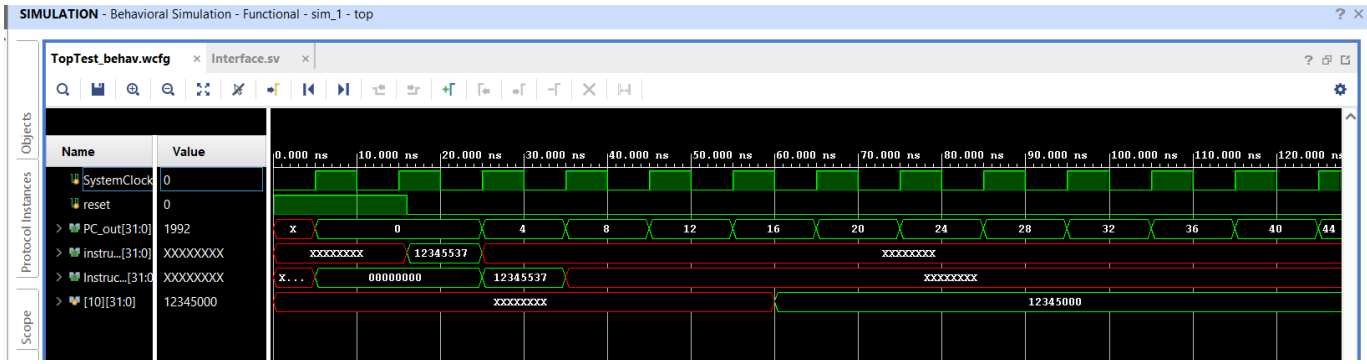
➔ Ta có : x12 = 0xbb, x10 = 25, x1 = 10. Cpu hoạt động đúng như dự kiến.

1.4. Nhóm lệnh lui, aupic

1.4.1. Lệnh lui

```
/* Test case 21 : lui
lui x10, 0x12345
*/
cpu.datapath.instructionMemory.Imemory[0] = 32'h12345537;
```

Kết quả chạy mô phỏng:

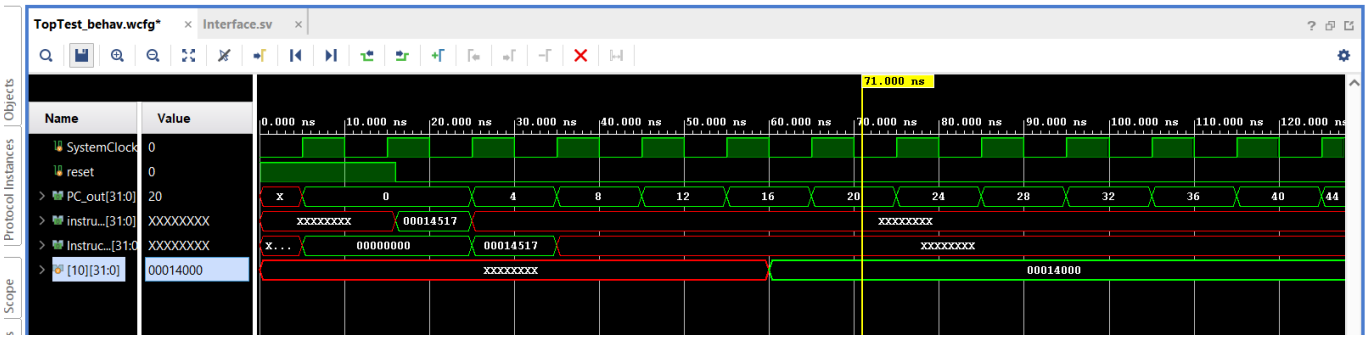


➔ Ta có : x10 = 0x12345000. Cpu hoạt động đúng như dự kiến.

1.4.2. Lệnh auipc

```
/* Test case 22 : auipc
auipc x10, 20
*/
cpu.datapath.instructionMemory.Imemory[0] = 32'h00014517;
```

Kết quả chạy mô phỏng:



➔ Ta có : x10 = 0x00014000. Cpu hoạt động đúng như dự kiến.

1.5. Nhóm lệnh j – type, và lw, sw

```
// Test case 22:
/* C++ :
    int mul(int a){
        return a*2;
    }

    int main(){
        int sum = 0;
        for (int i = 0; i < 10; i++){
            sum += mul(i);
            sum -= 1;
        }
        return 0;
    }

    Assembly code :

    j main
    mul:
        addi    sp,sp,-32
        sw      ra,28(sp)
```

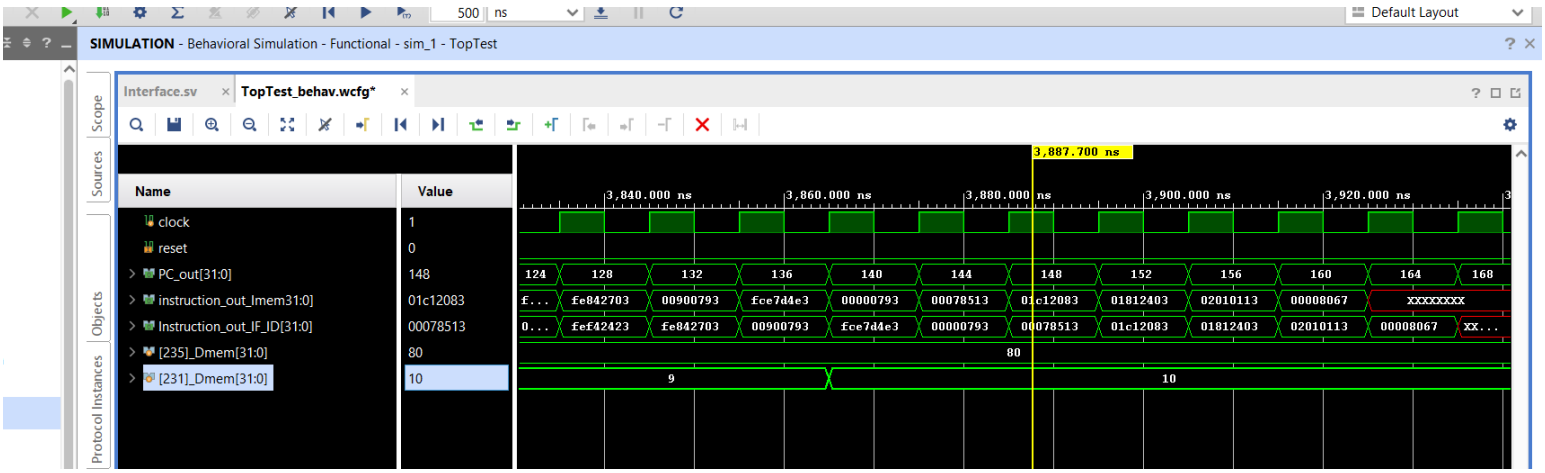
```
    sw      s0,24(sp)
    addi    s0,sp,32
    sw      a0,-20(s0)
    lw      a5,-20(s0)
    slli    a5,a5,1
    mv      a0,a5
    lw      ra,28(sp)
    lw      s0,24(sp)
    addi    sp,sp,32
    jr      ra

main:
    addi    sp,sp,-32
    sw      ra,28(sp)
    sw      s0,24(sp)
    addi    s0,sp,32
    sw      zero,-20(s0)
    sw      zero,-24(s0)
    j       L4

L5:
    lw      a0,-24(s0)
    call    mul
    mv      a4,a0
    lw      a5,-20(s0)
    add     a5,a5,a4
    sw      a5,-20(s0)
    lw      a5,-20(s0)
    addi    a5,a5,-1
    sw      a5,-20(s0)
    lw      a5,-24(s0)
```

```
        addi    a5,a5,1
        sw      a5,-24(s0)

    L4:
        lw      a4,-24(s0)
        li      a5,9
        ble     a4,a5,L5
        li      a5,0
        mv      a0,a5
        lw      ra,28(sp)
        lw      s0,24(sp)
        addi    sp,sp,32
        jr      ra

*/
```

- Trong đó sp là thanh ghi x2 = 255, x0 = 0.

5-bit Encoding (rx)	3-bit Compressed Encoding (rx')	Register	ABI Name	Description	Saved by Calle-
0	-	x0	zero	hardwired zero	-
1	-	x1	ra	return address	-R
2	-	x2	sp	stack pointer	-E
3	-	x3	gp	global pointer	-
4	-	x4	tp	thread pointer	-
5	-	x5	t0	temporary register 0	-R
6	-	x6	t1	temporary register 1	-R
7	-	x7	t2	temporary register 2	-R
8	0	x8	s0 / fp	saved register 0 / frame pointer	-E
9	1	x9	s1	saved register 1	-E
10	2	x10	a0	function argument 0 / return value 0	-R
11	3	x11	a1	function argument 1 / return value 1	-R
12	4	x12	a2	function argument 2	-R
13	5	x13	a3	function argument 3	-R
14	6	x14	a4	function argument 4	-R
15	7	x15	a5	function argument 5	-R
16	-	x16	a6	function argument 6	-R
17	-	x17	a7	function argument 7	-R
18	-	x18	s2	saved register 2	-E
19	-	x19	s3	saved register 3	-E
20	-	x20	s4	saved register 4	-E

- Kết quả chạy mô phỏng:



➔ Ta có :

+ Dmem[235] = 80 (lưu giá trị biến sum).

+ Dmem[231] = 10 (lưu giá trị biến i).

➔ Cpu hoạt động đúng như dự kiến.

--- Hết ---