Assembly Instructions to be executed, their Machine code in Instruction Memory and the Simulation and Memory from Questa Sim:

Case:1- GCD

The assembly code is as follows:-

```
li x8, 56
li x9, 84
gcd:
    beq x8, x9, stop
    blt x8, x9, less
    sub x8, x8, x9
    j gcd
less:
    sub x9, x9, x8
    j gcd
stop:
    sw x8, 0x08(x0)
    lw x10, 0x08(x0)
end:
    j end
```

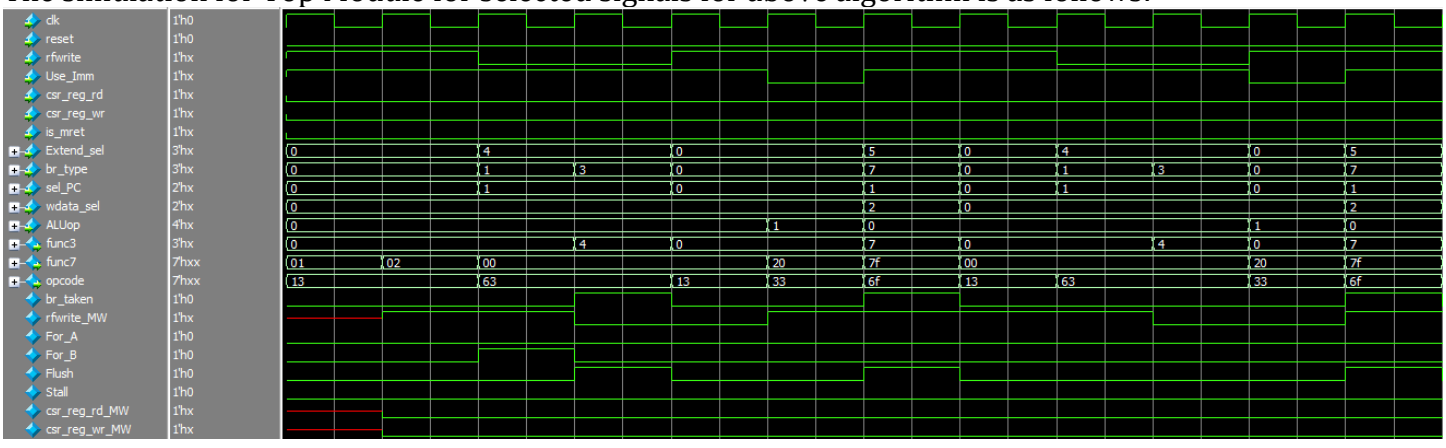
Listing 1. Assembly code to be executed

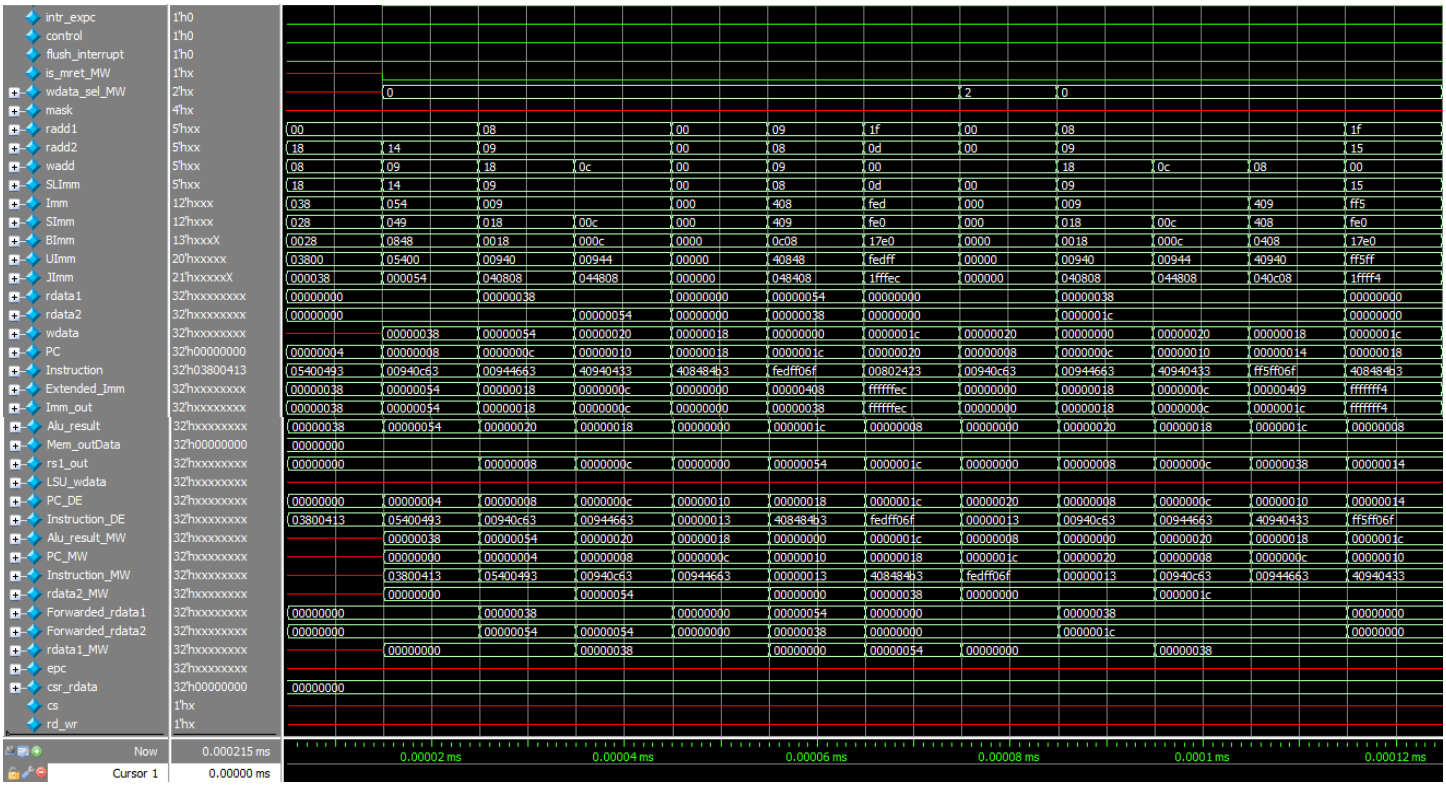
The generated machine code stored in instruction memory is as follows:-

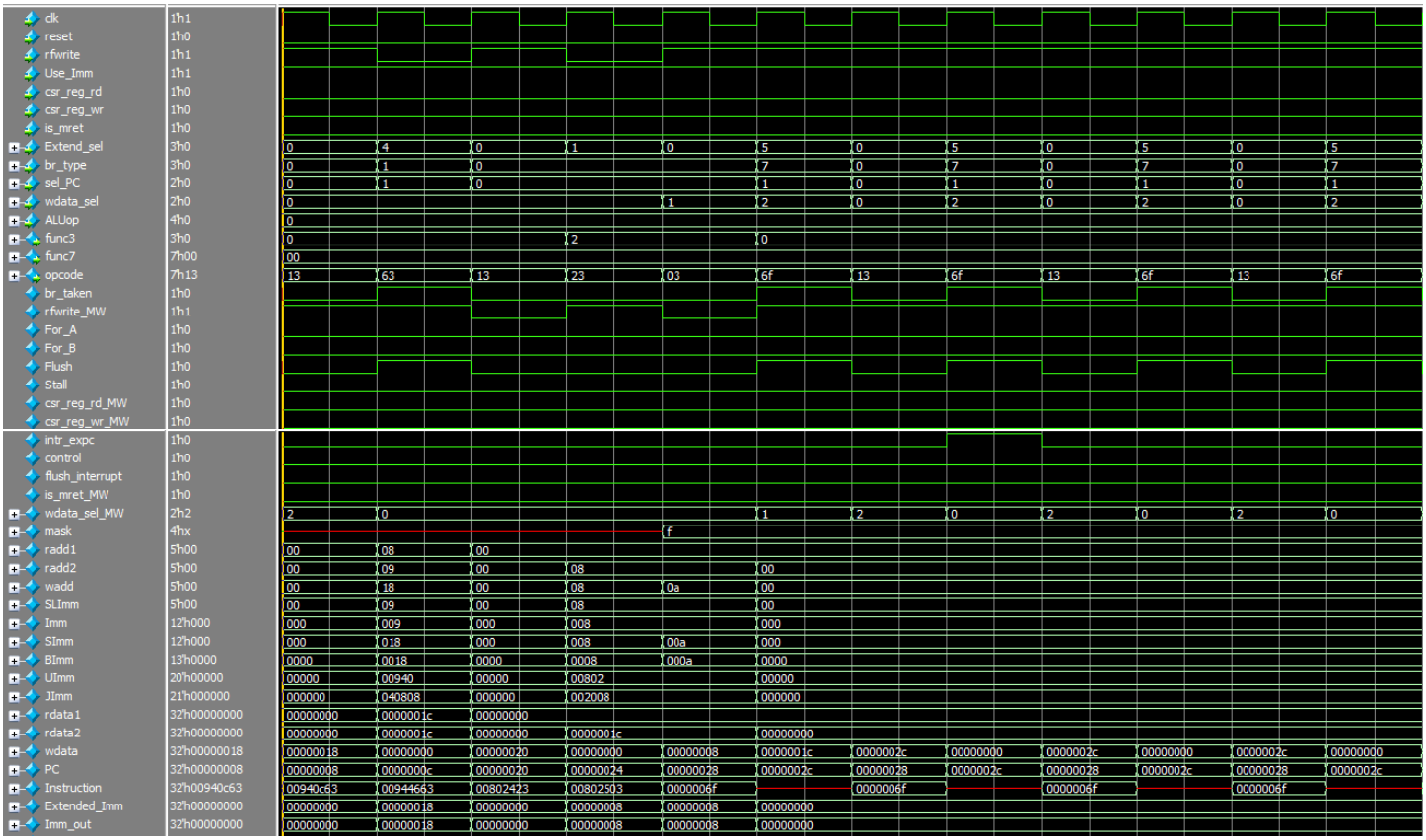
```
03800413
05400493
00940c63
00944663
40940433
ff5ff06f
408484b3
fedff06f
00802423
00802503
0000006f
```

Listing 2. Instruction memory

The simulation for Top Module for selected signals for above algorithm is as follows:-




intr_expc 1'h0
control 1'h0
flush_interrupt 1'h0
is_mret_MW 1'hx
wdata_sel_MW 2'hx
mask 4'hx
radd1 5'hxx
radd2 5'hxx
wadd 5'hxx
SLImm 5'hxx
Imm 12'hxxx
SImm 12'hxxx
BImm 13'hxxxX
UImm 20'hxxxxx
JImm 21'hxxxxxX
rdata1 32'hxxxxxxxx
rdata2 32'hxxxxxxxx
wdata 32'hxxxxxxxx
PC 32'h00000000
Instruction 32'h03800413
Extended_Imm 32'hxxxxxxxx
Imm_out 32'hxxxxxxxx
Alu_result 32'hxxxxxxxx
Mem_outData 32'h00000000
rs1_out 32'hxxxxxxxx
LSU_wdata 32'hxxxxxxxx
PC_DE 32'hxxxxxxxx
Instruction_DE 32'hxxxxxxxx
Alu_result_MW 32'hxxxxxxxx
PC_MW 32'hxxxxxxxx
Instruction_MW 32'hxxxxxxxx
rdata2_MW 32'hxxxxxxxx
Forwarded_rdata1 32'hxxxxxxxx
Forwarded_rdata2 32'hxxxxxxxx
rdata1_MW 32'hxxxxxxxx
epc 32'hxxxxxxxx
csr_rdata 32'h00000000
cs 1'hx
rd_wr 1'hx
Now 0.000215 ms
Cursor 1 0.00000 ms
0.00002 ms
0.00004 ms
0.00006 ms
0.00008 ms
0.0001 ms
0.00012 ms


clk 1'h1
reset 1'h0
rfwrite 1'h1
Use_Imm 1'h1
csr_reg_rd 1'h0
csr_reg_wr 1'h0
is_mret 1'h0
Extend_sel 3'h0
br_type 3'h0
sel_PC 2'h0
wdata_sel 2'h0
ALUop 4'h0
func3 3'h0
func7 7'h00
opcode 7'h13
br_taken 1'h0
rfwrite_MW 1'h1
For_A 1'h0
For_B 1'h0
Flush 1'h0
Stall 1'h0
csr_reg_rd_MW 1'h0
csr_reg_wr_MW 1'h0
intr_expc 1'h0
control 1'h0
flush_interrupt 1'h0
is_mret_MW 1'h0
wdata_sel_MW 2'h2
mask 4'hx
radd1 5'h00
radd2 5'h00
wadd 5'h00
SLImm 5'h00
Imm 12'h000
SImm 12'h000
BImm 13'h0000
UImm 20'h00000
JImm 21'h000000
rdata1 32'h00000000
rdata2 32'h00000000
wdata 32'h00000018
PC 32'h00000008
Instruction 32'h00940c63
Extended_Imm 32'h00000000
Imm_out 32'h00000000

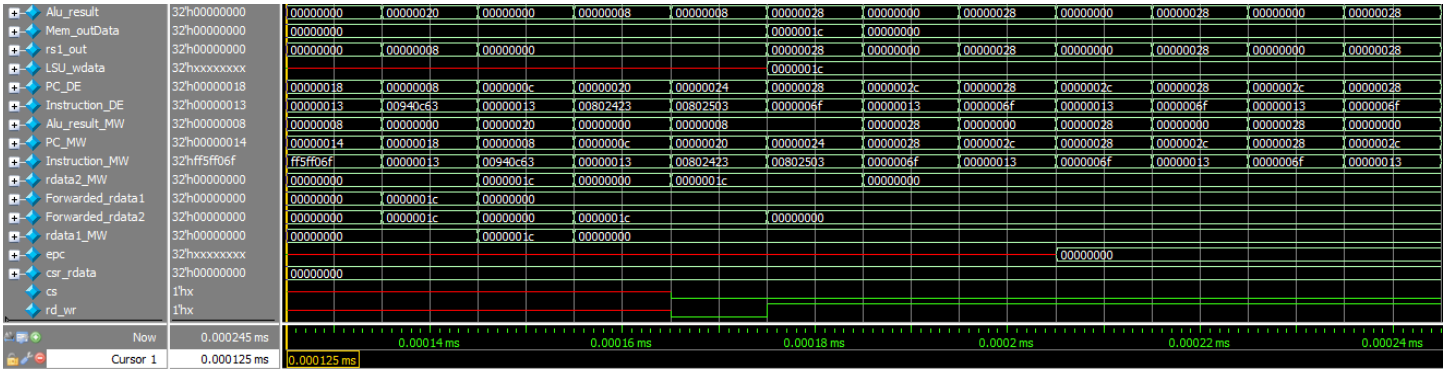


Figure 6. Top Module Simulation

The register memory after simulation is as follows:-

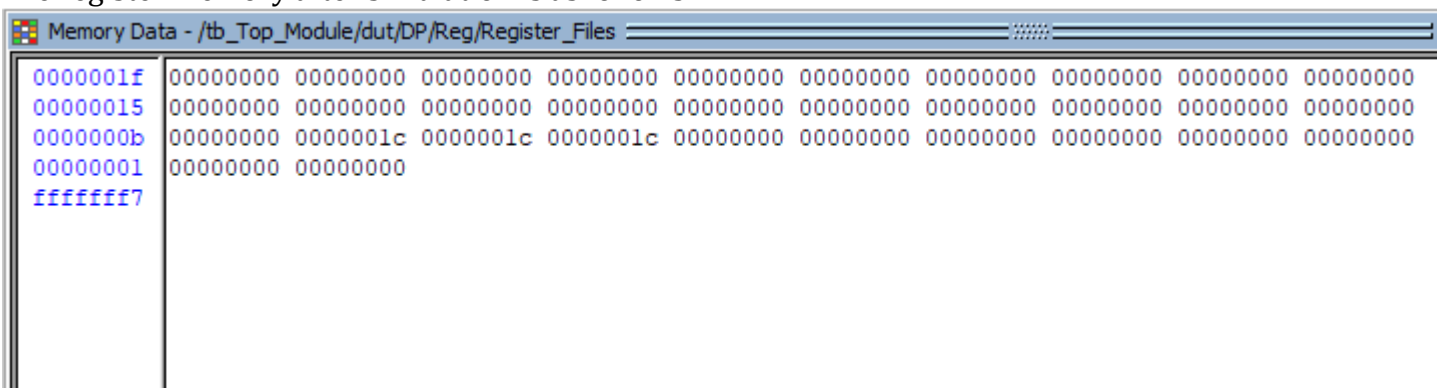


Figure 7. Register memory

The data memory after simulation is as follows:-

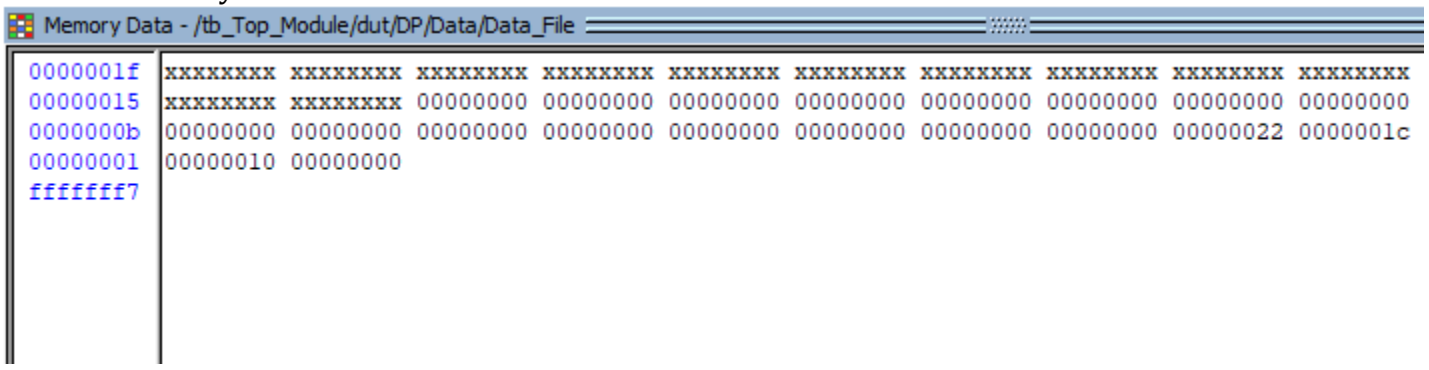


Figure 8. Data memory

The instruction memory after simulation is as follows:-

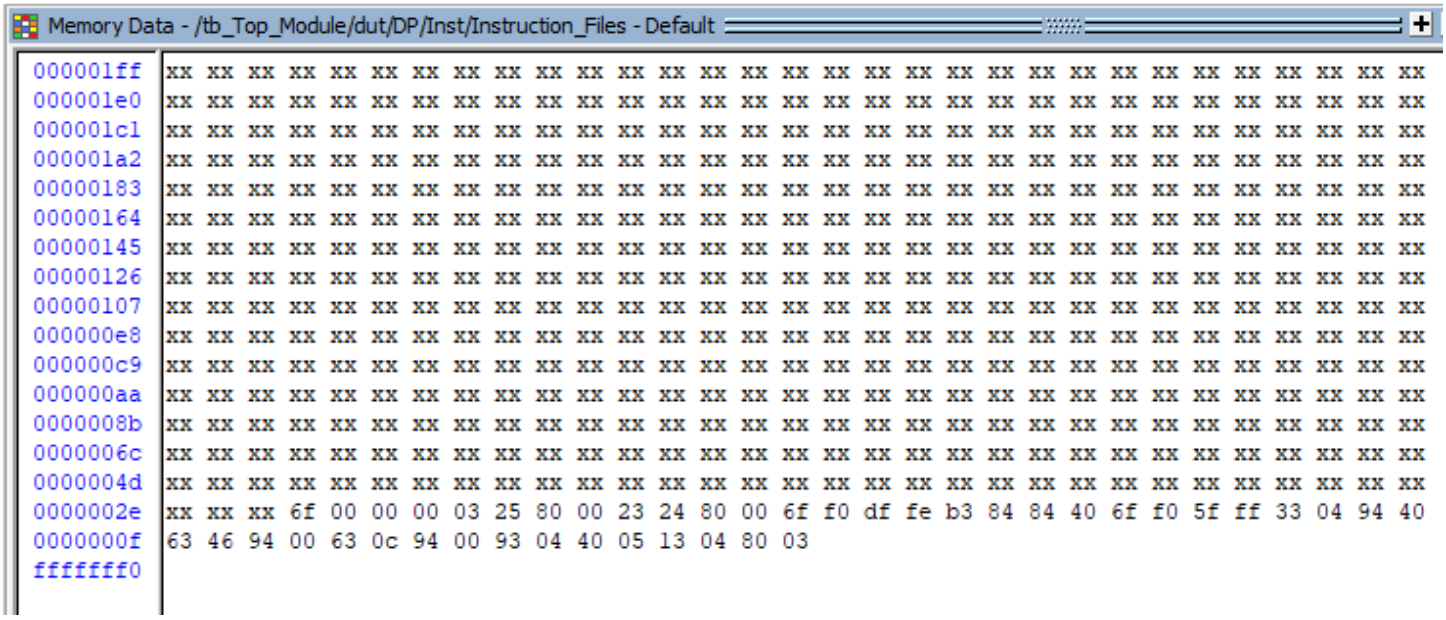


Figure 9. Instruction memory

Case:2- Stalling, Forwarding and Flushing

Here is the assembly code to be tested:-

```
    addi x7, x0, 3
    addi x8, x7, 25
    sw x8, 0x0c(x0)
    lw x9, 0x0c(x0)
    add x10, x9, x8
main:
    addi x1, x0, 2
    addi x2, x1, 3
    bne x2, x1, jump
back:
    sw x4, 0x08(x0)
    lw x5, 0x08(x0)
    sub x6, x5, x2
    bge x5, x1, stop
jump:
    sw x2, 0x04(x0)
    lw x3, 0x04(x0)
    add x4, x3, x2
    j back
stop:
    j stop
```

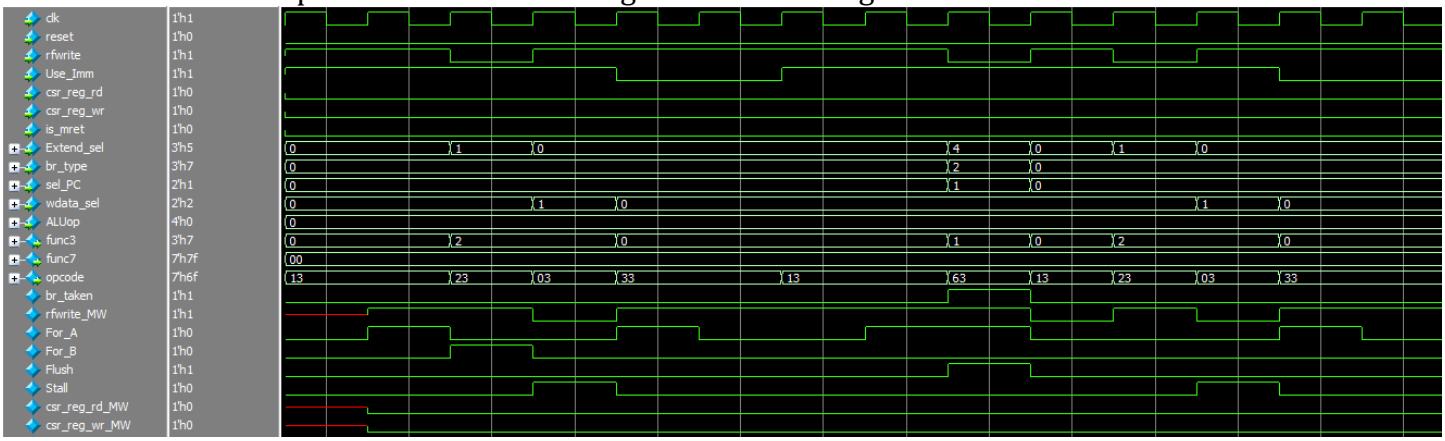
Listing 3. Assembly code to be executed

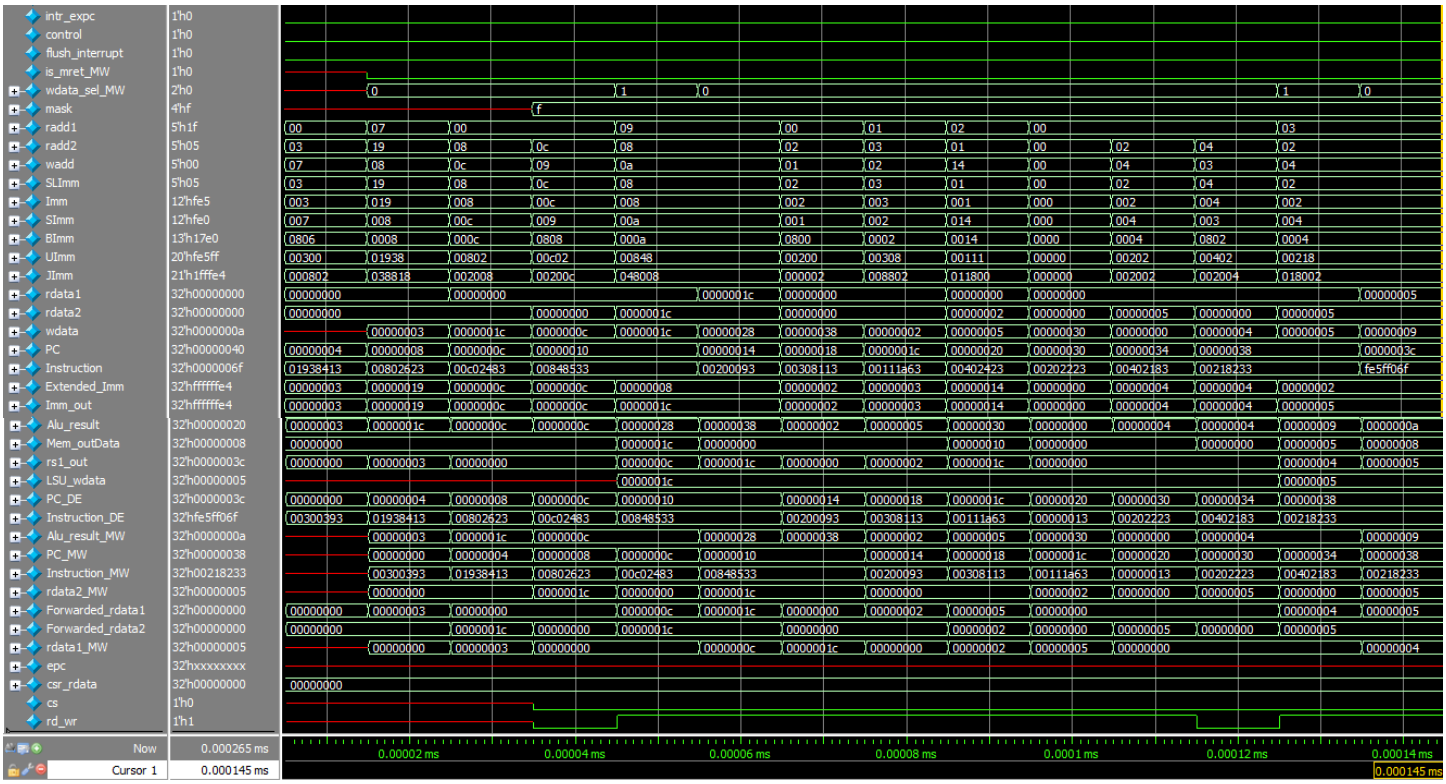
The generated machine code stored in instruction memory is as follows:-

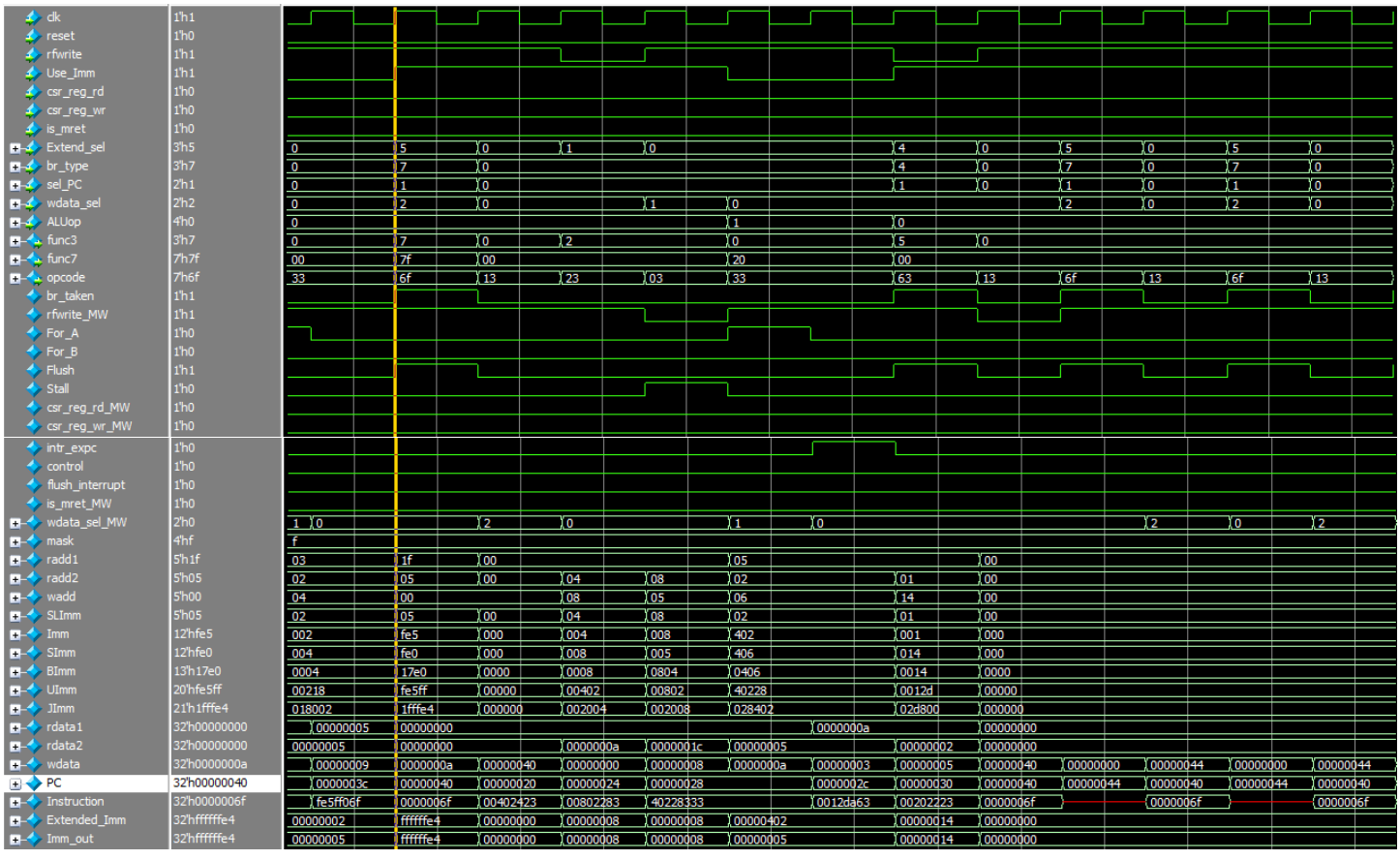
```
00300393
01938413
00802623
00c02483
00848533
00200093
00308113
00111a63
00402423
00802283
40228333
0012da63
00202223
00402183
00218233
fe5ff06f
0000006f
```

Listing 4. Instruction memory

The simulation for Top Module for selected signals for above algorithm is as follows:-




intr_expc 1'h0
control 1'h0
flush_interrupt 1'h0
is_mret_MW 1'h0
wdata_sel_MW 2'h0
mask 4'hf
radd1 5'h1f
radd2 5'h05
wadd 5'h00
SLImm 5'h05
Imm 12'hfe5
SImm 12'hfe0
BImm 13'h17e0
UImm 20'hfe5ff
JImm 21'h1fffe4
rdata1 32'h00000000
rdata2 32'h00000000
wdata 32'h0000000a
PC 32'h00000040
Instruction 32'h0000006f
Extended_Imm 32'hffffffe4
Imm_out 32'hffffffe4
Alu_result 32'h00000020
Mem_outData 32'h00000008
rs1_out 32'h0000003c
LSU_wdata 32'h00000005
PC_DE 32'h0000003c
Instruction_DE 32'hfe5ff06f
Alu_result_MW 32'h0000000a
PC_MW 32'h00000038
Instruction_MW 32'h00218233
rdata2_MW 32'h00000005
Forwarded_rdata1 32'h00000000
Forwarded_rdata2 32'h00000000
rdata1_MW 32'h00000005
epc 32'hxxxxxxxx
csr_rdata 32'h00000000
cs 1'h0
rd_wr 1'h1
Now 0.000265 ms
Cursor 1 0.000145 ms
0.00002 ms
0.00004 ms
0.00006 ms
0.00008 ms
0.0001 ms
0.00012 ms
0.00014 ms
0.000145 ms


clk 1'h1
reset 1'h0
rfwrite 1'h1
Use_Imm 1'h1
csr_reg_rd 1'h0
csr_reg_wr 1'h0
is_mret 1'h0
Extend_sel 3'h5
br_type 3'h7
sel_PC 2'h1
wdata_sel 2'h2
ALUop 4'h0
func3 3'h7
func7 7'h7f
opcode 7'h6f
br_taken 1'h1
rfwrite_MW 1'h1
For_A 1'h0
For_B 1'h0
Flush 1'h1
Stall 1'h0
csr_reg_rd_MW 1'h0
csr_reg_wr_MW 1'h0
intr_expc 1'h0
control 1'h0
flush_interrupt 1'h0
is_mret_MW 1'h0
wdata_sel_MW 2'h0
mask 4'hf
radd1 5'h1f
radd2 5'h05
wadd 5'h00
SLImm 5'h05
Imm 12'hfe5
SImm 12'hfe0
BImm 13'h17e0
UImm 20'hfe5ff
JImm 21'h1fffe4
rdata1 32'h00000000
rdata2 32'h00000000
wdata 32'h0000000a
PC 32'h00000040
Instruction 32'h0000006f
Extended_Imm 32'hffffffe4
Imm_out 32'hffffffe4

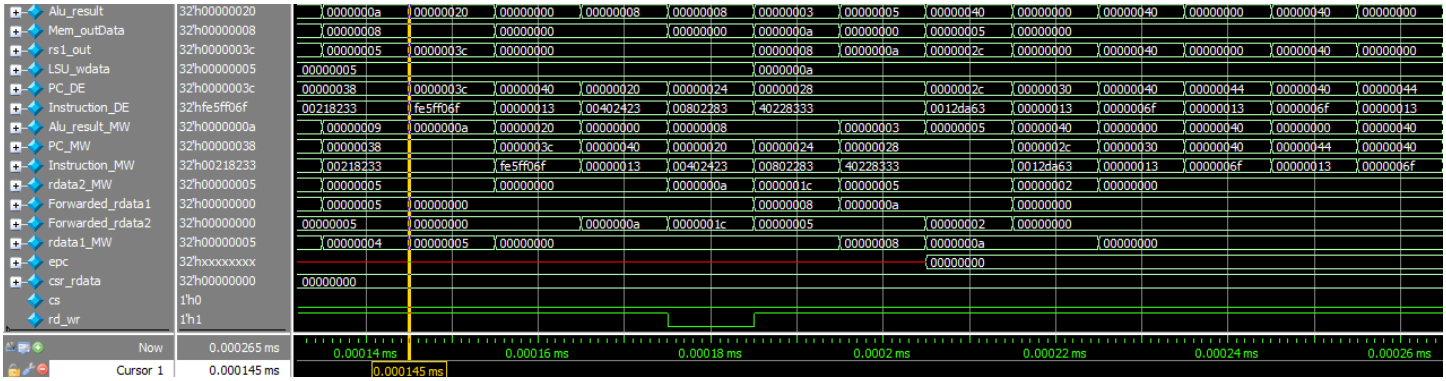


Figure 10. Top Module Simulation

The register memory after simulation is as follows:-

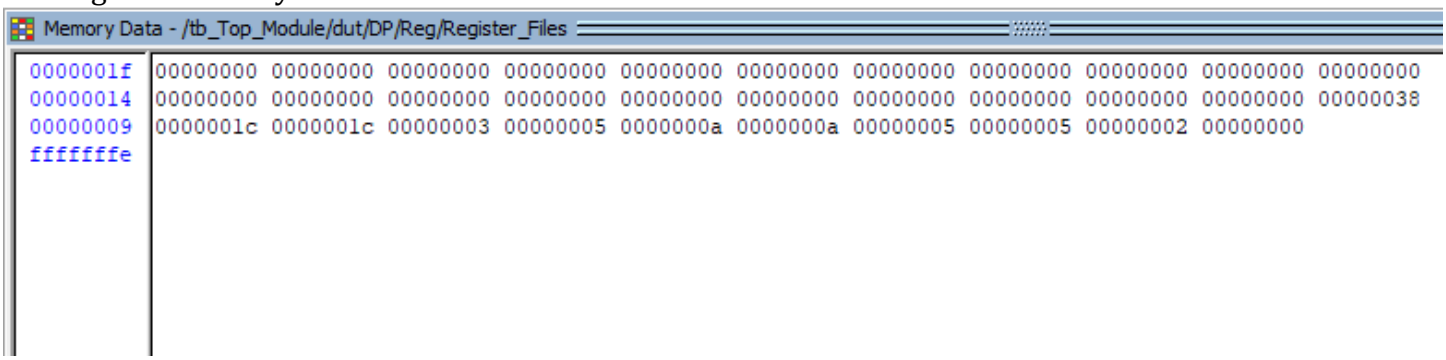


Figure 11. Register memory

The data memory after simulation is as follows:-

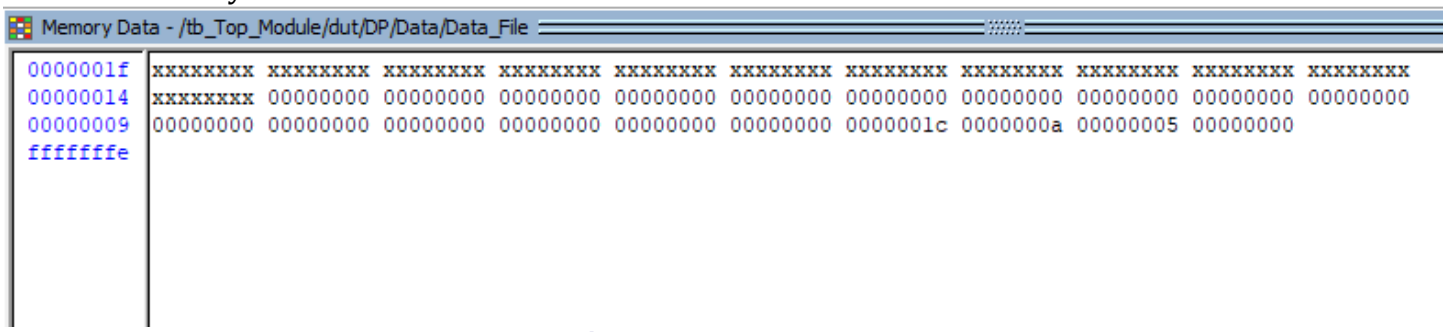


Figure 12. Data memory

The instruction memory after simulation is as follows:-

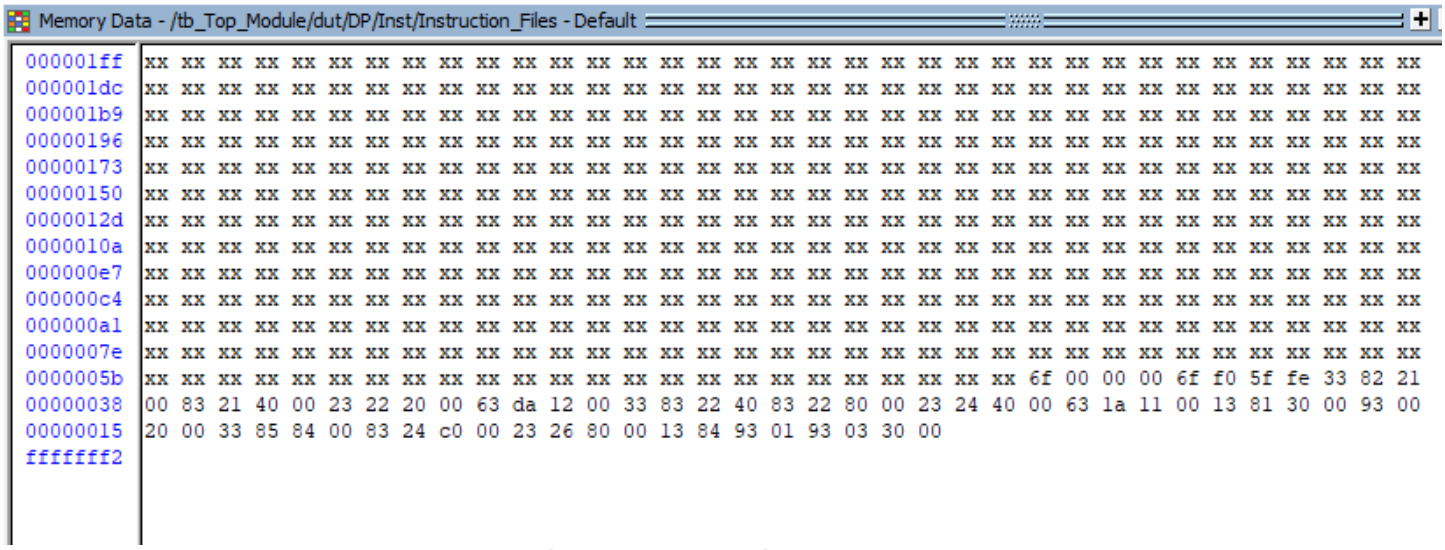


Figure 13. Instruction memory

Case:3- CSR Timer Interrupt Handler

Here is the assembly code to be tested:-

```
li x2, 8

li x3, 128
li x4, 28

csrrw x0, mstatus, x2
csrrw x0, mie, x3
csrrw x0, mtvec, x4

jal x12,  main
jal x13, handler

main:
    addi x6, x6, 1
    addi x7, x7, 2
stop:
    addi x9, x9, 1
    addi x10, x10, 2
    jal x14, stop

handler:
    csrrw x0,mie,x0
    xori x16, x16, 0xFFFFFFFF
    addi x17, x17, 10
    csrrw x0,mie, x3
    mret
    nop
    nop
    nop
```

Listing 5. Assembly code to be executed

The generated machine code stored in instruction memory is as follows:-

```
00800113
08000193
01c00213
30011073
30419073
30521073
0080066f
018006ef
00130313
00238393
00148493
00250513
ff9ff76f
30401073
fff84813
00a88893
30419073
30200073
00000013
00000013
00000013
```

Listing 6. Instruction memory

When interrupt occurs first time, register memory (handler executed) is as follows:-

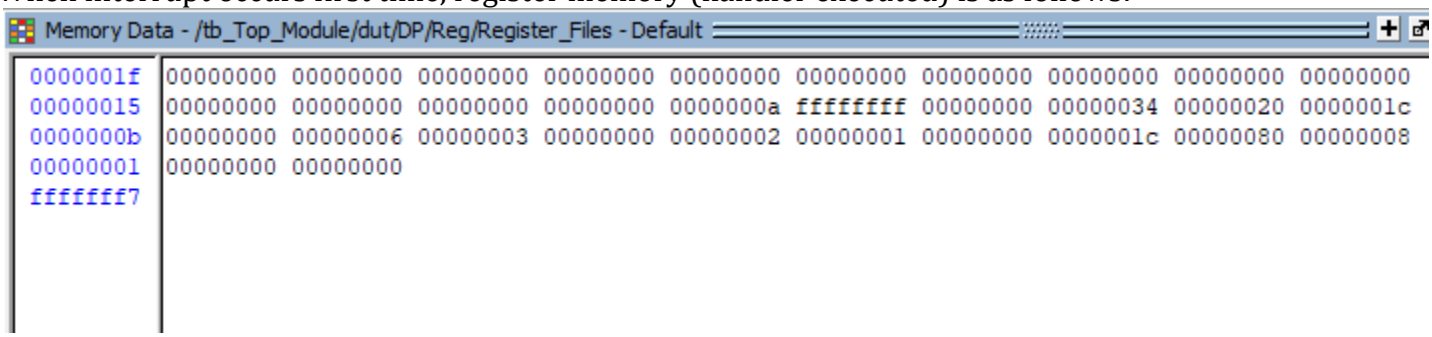


Figure 10. Register Memory

When interrupt occurs second time, register memory (handler executed) is as follows:-

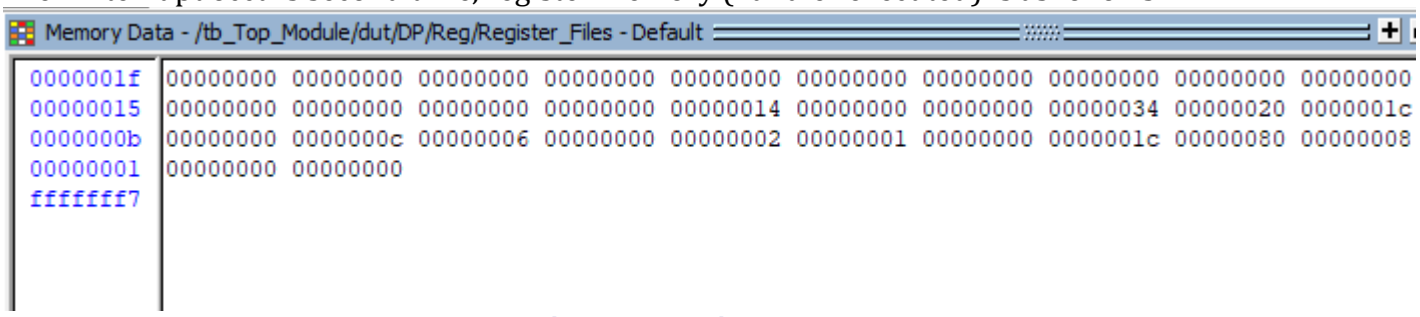
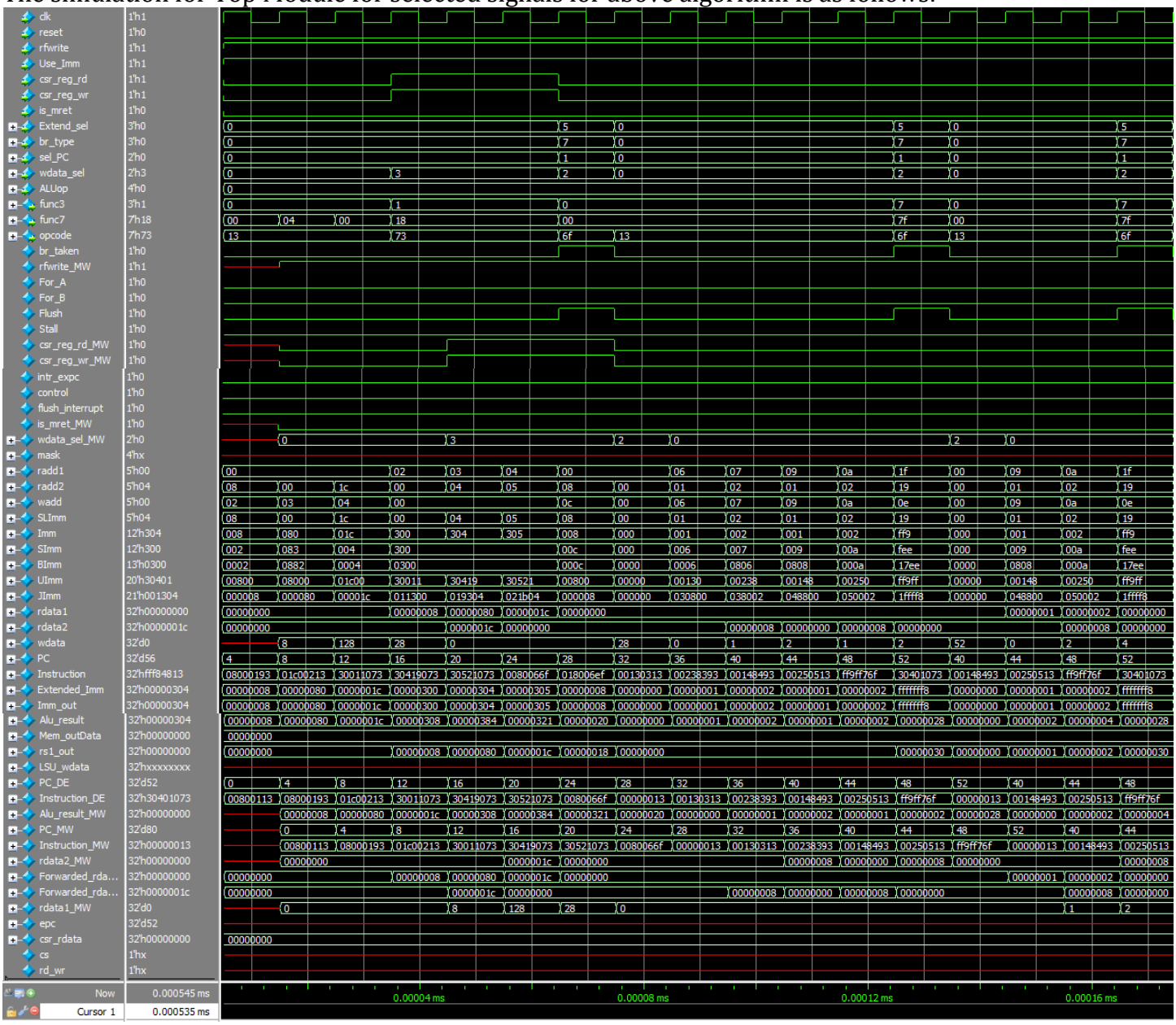
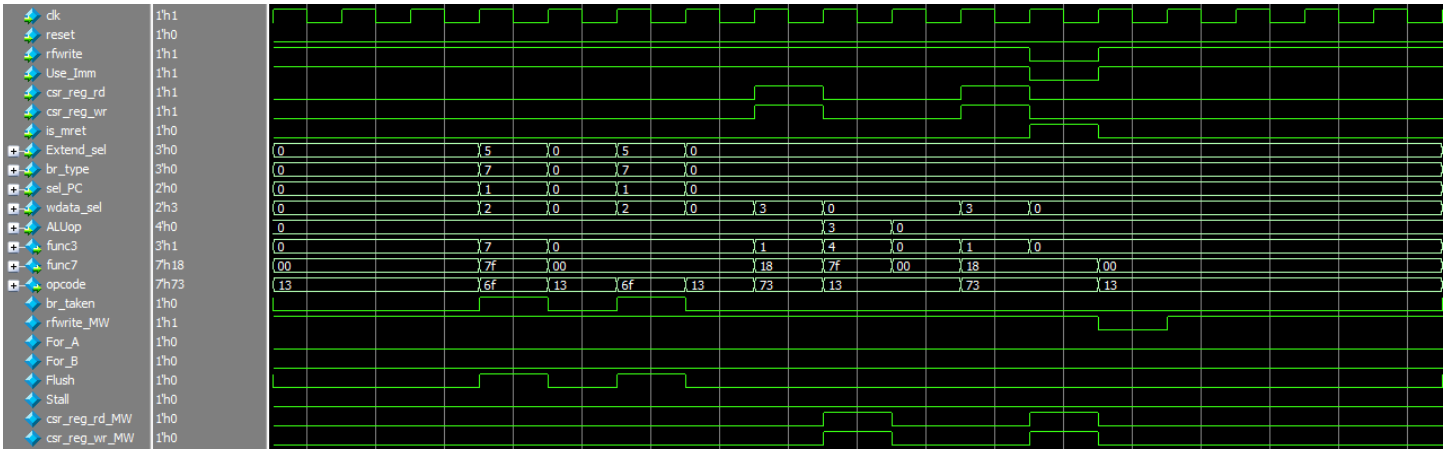
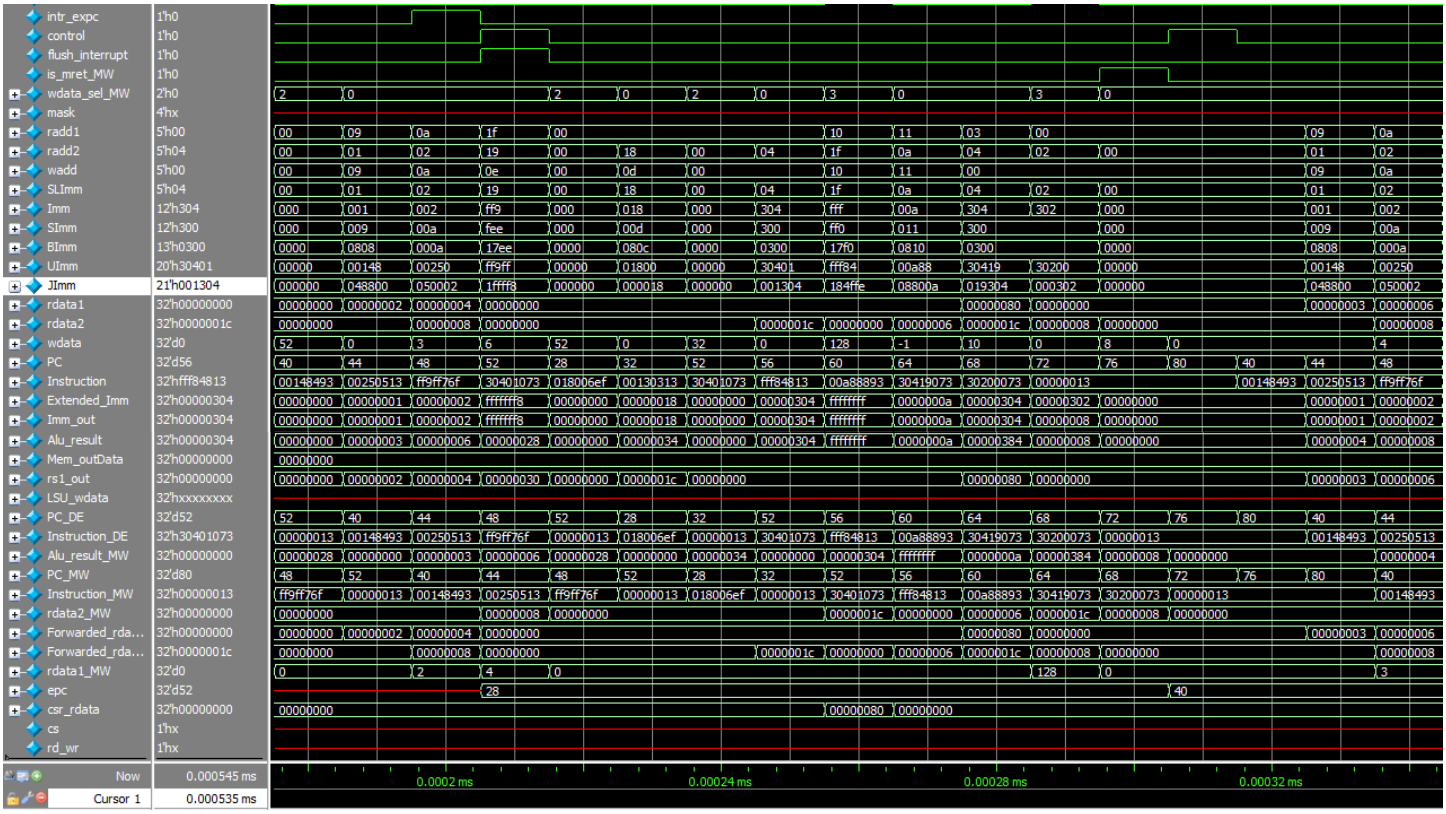


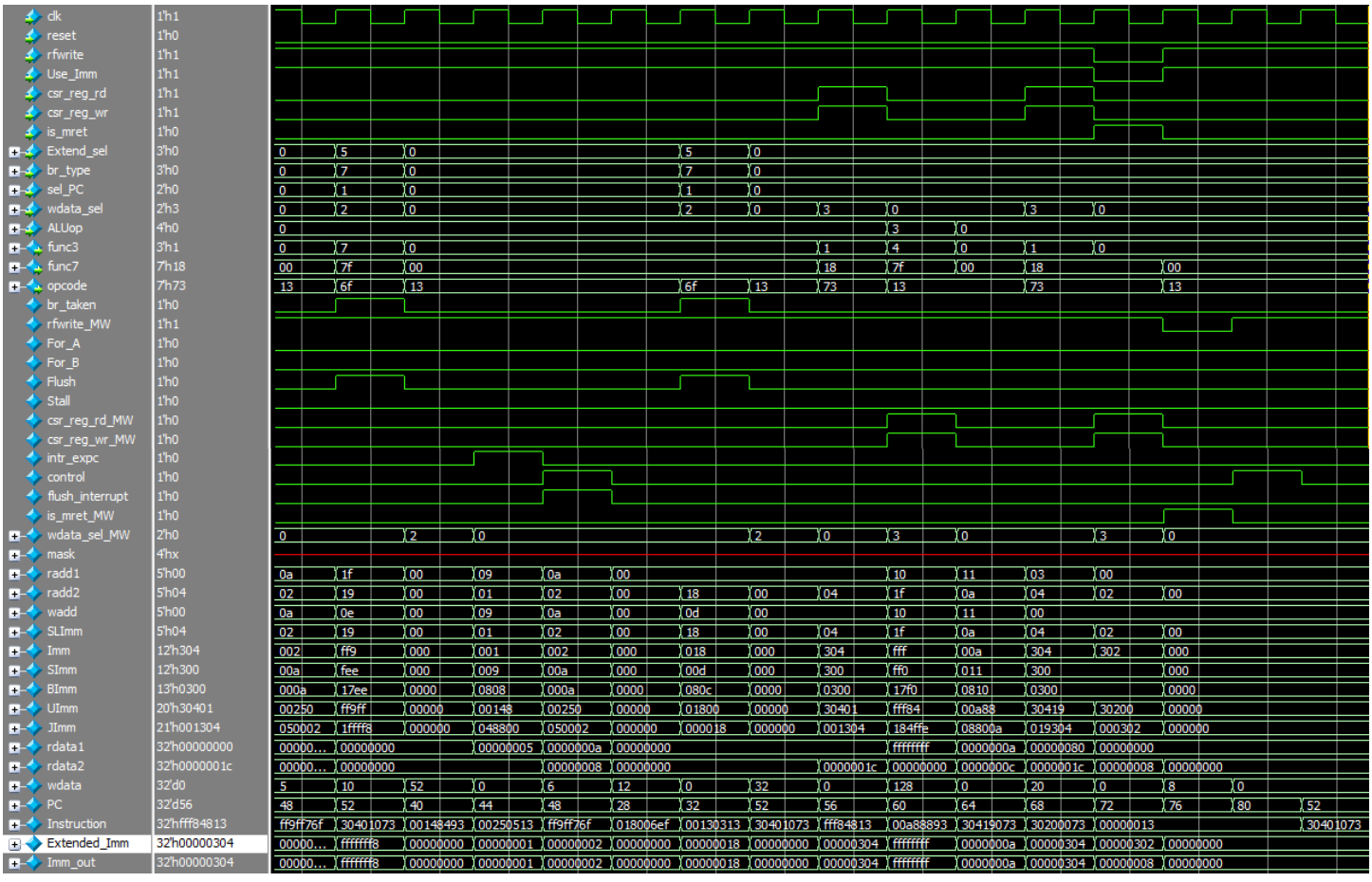
Figure 11. Register memory

The simulation for Top Module for selected signals for above algorithm is as follows:-






intr_expc 1'h0
control 1'h0
flush_interrupt 1'h0
is_mret_MW 1'h0
wdata_sel_MW 2'h0
mask 4'hx
radd1 5'h00
radd2 5'h04
wadd 5'h00
SLImm 5'h04
Imm 12'h304
SImm 12'h300
BImm 13'h0300
UImm 20'h30401
JImm 21'h001304
rdata1 32'h00000000
rdata2 32'h0000001c
wdata 32'd0
PC 32'd56
Instruction 32'hfff84813
Extended_Imm 32'h00000304
Imm_out 32'h00000304
Alu_result 32'h00000304
Mem_outData 32'h00000000
rs1_out 32'h00000000
LSU_wdata 32'hxxxxxxxx
PC_DE 32'd52
Instruction_DE 32'h30401073
Alu_result_MW 32'h00000000
PC_MW 32'd80
Instruction_MW 32'h00000013
rdata2_MW 32'h00000000
Forwarded_rda... 32'h00000000
Forwarded_rda... 32'h0000001c
rdata1_MW 32'd0
epc 32'd52
csr_rdata 32'h00000000
cs 1'hx
rd_wr 1'hx
Now 0.000545 ms
Cursor 1 0.000535 ms
0.0002 ms
0.00024 ms
0.00028 ms
0.00032 ms


clk 1'h1
reset 1'h0
rfwrite 1'h1
Use_Imm 1'h1
csr_reg_rd 1'h1
csr_reg_wr 1'h1
is_mret 1'h0
Extend_sel 3'h0
br_type 3'h0
sel_PC 2'h0
wdata_sel 2'h3
ALUop 4'h0
func3 3'h1
func7 7'h18
opcode 7'h73
br_taken 1'h0
rfwrite_MW 1'h1
For_A 1'h0
For_B 1'h0
Flush 1'h0
Stall 1'h0
csr_reg_rd_MW 1'h0
csr_reg_wr_MW 1'h0
intr_expc 1'h0
control 1'h0
flush_interrupt 1'h0
is_mret_MW 1'h0
wdata_sel_MW 2'h0
mask 4'hx
radd1 5'h00
radd2 5'h04
wadd 5'h00
SLImm 5'h04
Imm 12'h304
SImm 12'h300
BImm 13'h0300
UImm 20'h30401
JImm 21'h001304
rdata1 32'h00000000
rdata2 32'h0000001c
wdata 32'd0
PC 32'd56
Instruction 32'hfff84813
Extended_Imm 32'h00000304
Imm_out 32'h00000304

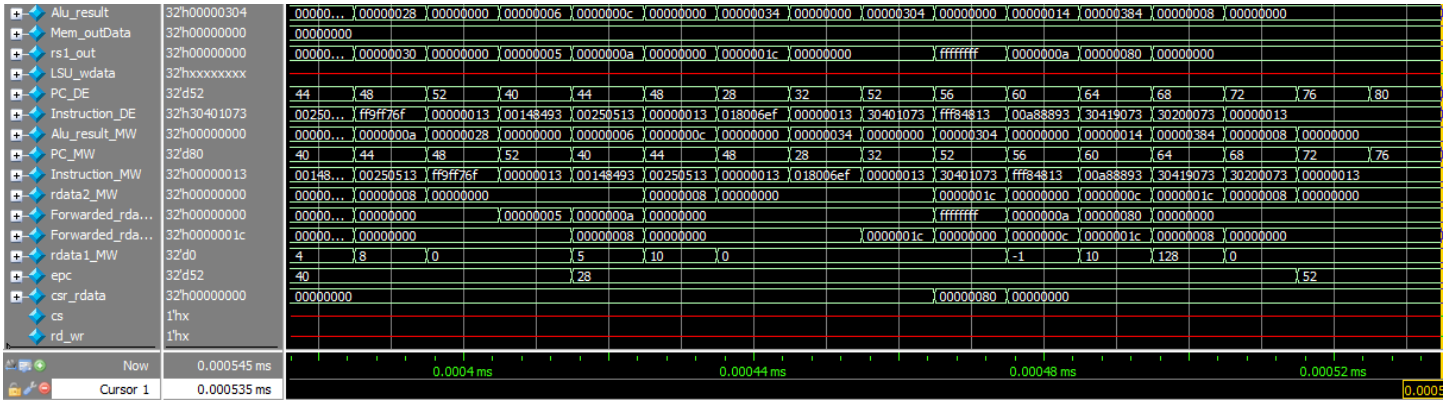


Figure 12. Top Module Simulation

The instruction memory after simulation is as follows:-

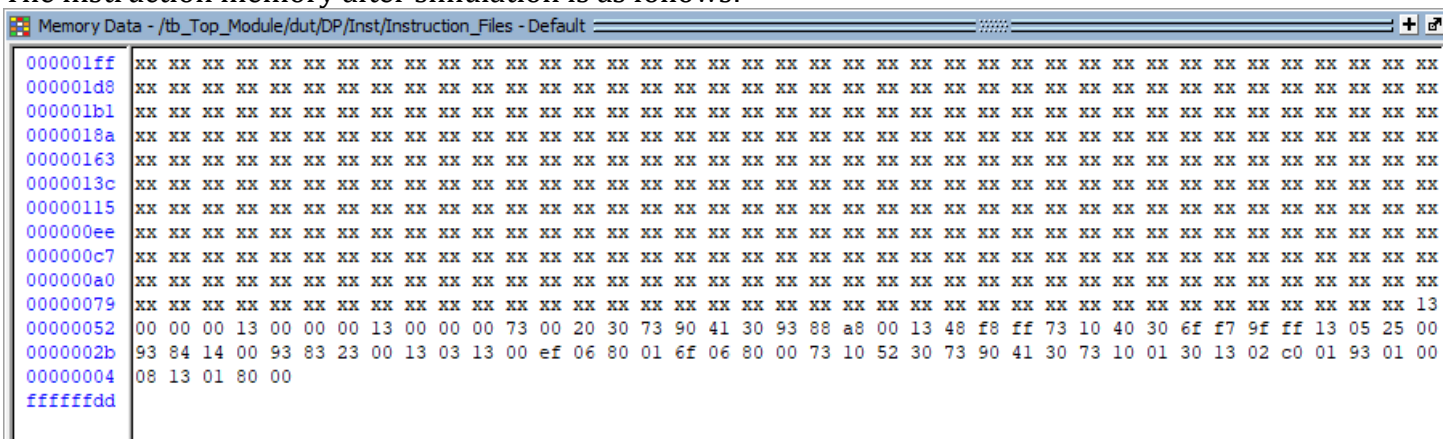


Figure 13. Instruction memory