ADDED COMPONENTS

1- Mux to select pc+4 or data to write a register.
2- Mux to select address from instruction or (memory or register).
3- Register file consist of 3 registers with 1 bit.
4- B_new signal.
5- Link signal.
6- Addr_from_mem signal.
7- Flag_sig signal.

Simulations

1- Sim_1

Srlv $2, $1, $0

Sw $2, 0($3)

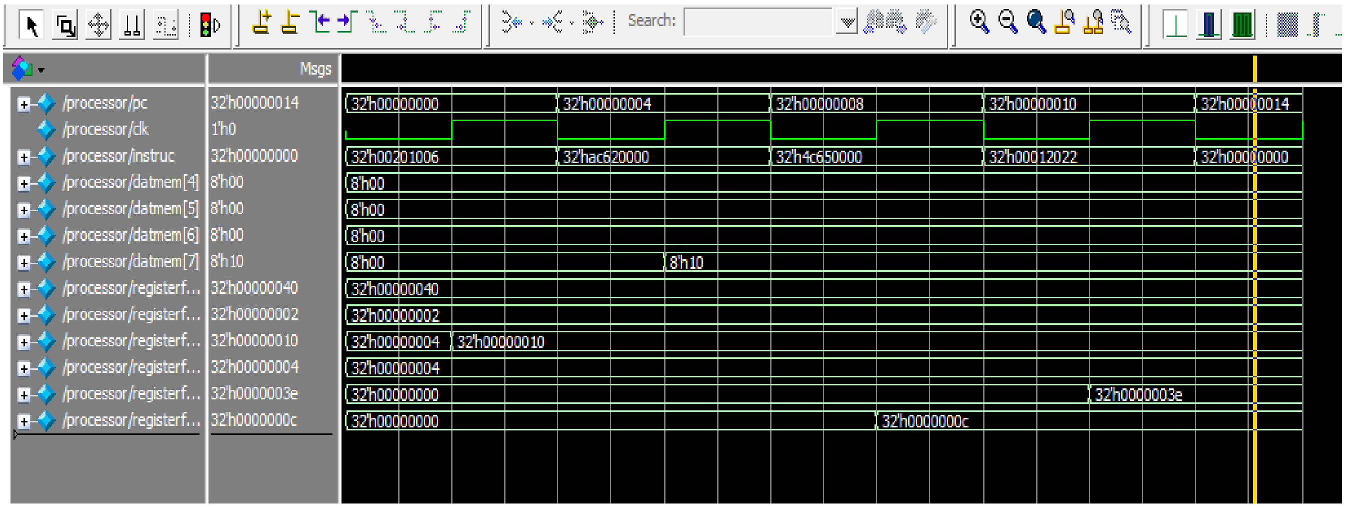
Jalm $5, 0($3)

Add $4, $0, $1

Sub $4, $0, $1

Initial;

$0 = 0x40 $1 = 0x2 $2 = 0x4 $3 = 0x4



	Msgs
/processor/pc	32'h00000014
/processor/clk	1'h0
/processor/instruc	32'h00000000
/processor/datmem[4]	8'h00
/processor/datmem[5]	8'h00
/processor/datmem[6]	8'h00
/processor/datmem[7]	8'h10
/processor/registerf...	32'h00000040
/processor/registerf...	32'h00000002
/processor/registerf...	32'h00000010
/processor/registerf...	32'h00000004
/processor/registerf...	32'h0000003e
/processor/registerf...	32'h0000000c

Srlv -> $2 = 0x10

Sw -> mem[4] = 0x10

Jalm -> branch to 0x10 , $5 = 0xc

Add -> X

Sub -> $4 = 0x3e

2- Sim_2

Sub $2, $1, $0

Bmn 0($3)

Or $2, $2, $0

Add $2, $1, $0

Bmn 4($3)

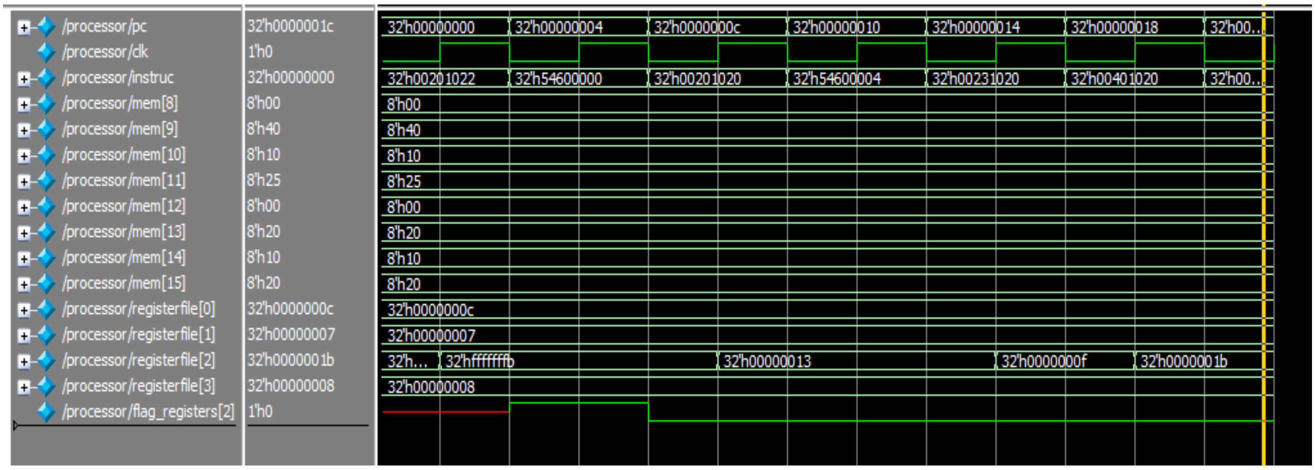
Add $2, $1, $3

Add $2, $2, $0

Initial;

$1 = 0x7 $0 = 0xc $3 = 0x8

mem[8] = 0xc mem[12] = 0x18


/processor/pc 32'h0000001c 32'h00000000 32'h00000004 32'h0000000c 32'h00000010 32'h00000014 32'h00000018 32'h00...
/processor/clk 1'h0
/processor/instruc 32'h00000000 32'h00201022 32'h54600000 32'h00201020 32'h54600004 32'h00231020 32'h00401020 32'h00...
/processor/mem[8] 8'h00 8'h00
/processor/mem[9] 8'h40 8'h40
/processor/mem[10] 8'h10 8'h10
/processor/mem[11] 8'h25 8'h25
/processor/mem[12] 8'h00 8'h00
/processor/mem[13] 8'h20 8'h20
/processor/mem[14] 8'h10 8'h10
/processor/mem[15] 8'h20 8'h20
/processor/registerfile[0] 32'h0000000c 32'h0000000c
/processor/registerfile[1] 32'h00000007 32'h00000007
/processor/registerfile[2] 32'h0000001b 32'h... 32'hfffffffb 32'h00000013 32'h0000000f 32'h0000001b
/processor/registerfile[3] 32'h00000008 32'h00000008
/processor/flag_registers[2] 1'h0

sub -> $2 = -5

bmn -> branch to 0xc (N = 1)

or -> X

add -> $2 = 0x13

bmn -> not branch (N = 0)

add -> $2 = 0xf

add -> $2 = 0x1b

3- Sim_3

add $2, $0, $1

brz $3

sw $3, 0($4)

add $2, $2, $1

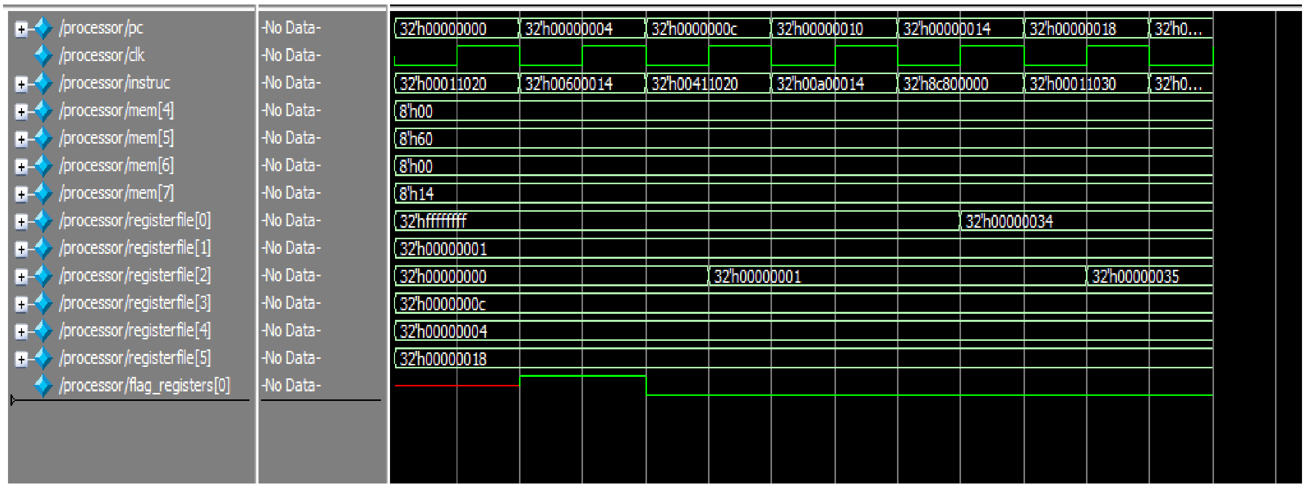
brz $5

lw $0, 0($4)

add $2, $0, $1

Initial;

$0 = -1 $1 = 0x1 $3 = 0xc $4 = 0x4 $5 = 0x18 mem[4] = 0x34



Add -> $2 = 0

Brz -> branch to 0xc

Sw -> X

Add -> $2 = 0x1

Brz -> not branch

Lw -> $0 = 0x34

Add -> $2 = 0x35

4- Sim_4

add $2, $0, $1

bz (target = 3)

sw $3, 0($4)

add $2, $2, $1

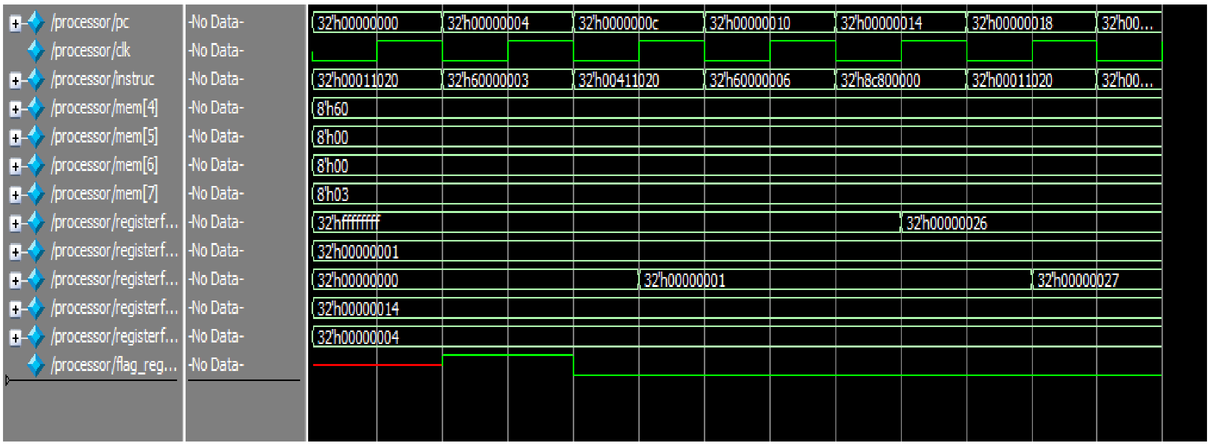
bz (target = 6)

lw $0, 0($4)

add $2, $0, $1

Initial;

$0 = -1 $1 = 0x1 $3 = 0x14 $4 = 0x4 mem[4] = 0x26



Add -> $2 = 0

Bz -> branch to 0xc

Sw -> X

Add -> $2 = 0x1

Bz -> not branch

Lw -> $0 = 0x26

Add -> $2 = 0x27

5- Sim_5

Sub $2, $1, $0

Balrn $3, $6

Or $2, $2, $0

Add $2, $1, $0

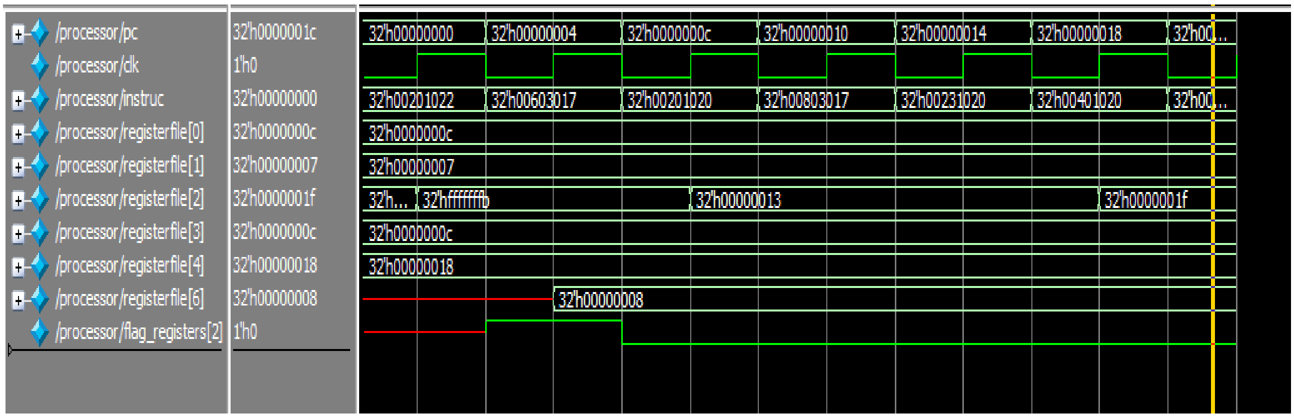
Balrn $4, $6

Add $2, $1, $3

Add $2, $2, $0

Initial;

$1 = 0x7 $0 = 0xc $3 = 0xc $4 = 0x18



sub -> $2 = -5

balrn -> branch to 0xc (N = 1)

or -> X

add -> $2 = 0x13

balrn -> not branch (N = 0)

add -> $2 = 0x13

add -> $2 = 0x1f