**Test Cases:**

* Testing (addi, slli, blt, sw):

addi x5, x5, 8

addi x6, x6, 10

slli x5, x5, 1

blt x6, x5, L2

andi x5, x5, 2

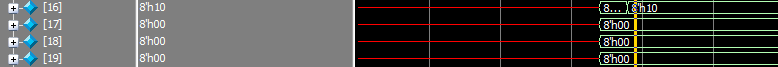
L2:

Sw x6, 0(x5)

Sw x5, 0(x5)

***the code above should skip the anding instruction. Therefore the final values in x6 and x5 should be 16,10 respectively.***

Memory bytes:



Registers:



* Testing ( slli, bne):

addi x5, x5, 8

addi x6, x6, 10

slli x5, x5, 1

bne x6, x5, L2

andi x5, x5, 2

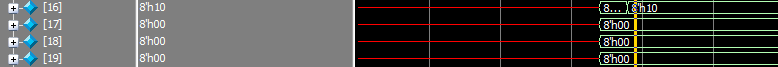
L2:

Sw x6, 0(x5)

Sw x5, 0(x5)

***the code above should skip the anding instruction. Therefore the final values in x6 and x5 should be 16,10 respectively.***

Memory Bytes:



Registers:



* Testing (srai, ble):

addi x5, x5, 8

addi x6, x6, 10

srai x6, x6, 1

ble x6, x5, L2

ori x5, x5, 2

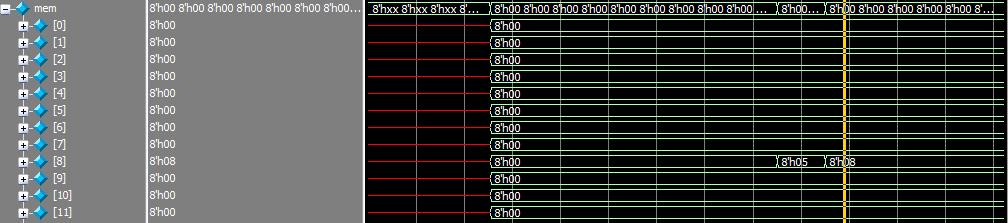
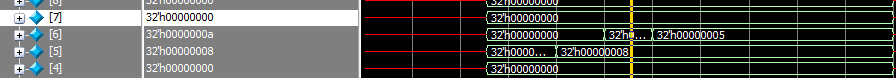
L2:

Sw x6, 0(x5)

Sw x5, 0(x5)

***the code above should skip the oring instruction. Therefore the final values in x6 and x5 should be 5, 8 respectively. These values will be strored in memory location 8( the value stored in x5***

Memory bytes:

Registers

* Testing (slli):

Addi x5, x5, -9

Slli x5,x5,2

***Testing logical lefts shifting directly.***

Registers:



* Testing (slri):

Addi x5, x5, -9

Slri x5,x5,2

***Testing logical right shifting directly.***

Registers:

