J\_JR\_ASSEMBLY.txt

LI A5,4 # Load 4 into A5

J 20 # Increase PC by 20 bytes

LI A7,20 # Load 20 into A7 (This should not happen)

LI RA,12 # Load 12 into RA

JR RA # Overwrite PC with RA

LI A1,10 # Load 10 into A1 (This should not happen)

J -12 # Decrease PC by 12 bytes

ADDI A6,A7,1 # Add A6 = A7 + 1

Overwritten registers:

RA = X1 = 12

SP = X2 = 1024

A5 = X15 = 4

A6 = X16 = 1

LI\_SP\_ADD\_ASSEMBLY.txt

LI A0,22166 # Load 22166 = 0b101 0110 1001 0110 into A0.

ADDI SP,SP,-16 # Add 16 bytes of memory to SP.

SW A0,0(SP) # Save 22166 into the first 4 bytes of SP.

LB A2,1(SP) # Get the second byte in SP (0b1010110 = 86) and save it in A2.

ADDI SP,SP,16 # Free SP memory.

Overwritten registers:

SP = X2 = 1024

A0 = X10 = 22166

A2 = X12 = 86

MV\_X0\_ASSEMBLY.txt

LI T0,10 # Load 10 into S0

MV X0,T0 # MV 10 into X0

MV A1,T0 # MV 10 into A1

Overwritten registers:

A1 = X11 = 10

T3 = X28 = 10

NEG\_NOT\_ASSEMBLY.txt

LI T5,256 # Load 256 into T5.

NEG A7,T5 # Negate 256 (-256) and save it in A7.

NOT A6,T55 # Ones complement of 256 (~256).

LI T5,1 # Overwrite T5 with a 1

ADD A5,A6,T5 # A5 = ~N + 1 = -N

Overwritten registers:

A7 = X17 = -256 = 4294967040 = 11111111111111111111111100000000

A6 = X16 = ~256 = 4294967039 = 11111111111111111111111011111111

A5 = X15 = ~256 + 1 = -256 = 4294967040

T5 = X28 = 1