

Course 1dt301: Computer Technology
1 Lab Assignment 1, 2023
Daniel Pearson, Yasaman Naghiloo

Task 1:

17 dec (or 11 hex, represented by 0x11) is stored in register x2 upon the assembly and execution of the provided code.

Task 2:

11010010100 0000000010000000 00010
MOVZ - 128 2

MOVZ x2,#128

11010010100 0000000011100111 00100
MOVZ - move= 231 rd=4

MOVZ x4,#231

11001011000 00010 000000 00100 00101
SUB x=2 - x=4 x=5

SUB x5,x4,x2

D360 0CA5

11010011011 00000 000011 00101 00101
LSL rm=0 shamt=3 rn=5 rd=5

LSL x5,x5,#3

Task 3:

LEGv8 Simulator

Execution Mode: Single Cycle Assemble Execute Instruction Help

1	MOVZ	x1, #5
2	LSL	x2, x1, #2
3	MOVZ	x3, #11
4	LSL	x4, x3, #4
5	MOVZ	x5, #25
6	ADD	x6, x2, x4
7	ADDI	x0, x6, #25
8		

PC	0x40001c	Hex	Dec	Z	0	N	0	C	0	V	0
X0	221	Hex	Dec	X16		0x0				Hex	Dec
X1	5	Hex	Dec	X17		0x0				Hex	Dec
X2	20	Hex	Dec	X18		0x0				Hex	Dec
X3	11	Hex	Dec	X19		0x0				Hex	Dec
X4	176	Hex	Dec	X20		0x0				Hex	Dec
X5	25	Hex	Dec	X21		0x0				Hex	Dec
X6	196	Hex	Dec	X22		0x0				Hex	Dec
X7	0x0	Hex	Dec	X23		0x0				Hex	Dec
X8	0x0	Hex	Dec	X24		0x0				Hex	Dec
X9	0x0	Hex	Dec	X25		0x0				Hex	Dec
X10	0x0	Hex	Dec	X26		0x0				Hex	Dec
X11	0x0	Hex	Dec	X27		0x0				Hex	Dec
X12	0x0	Hex	Dec	SP		0x7ffffffc				Hex	Dec
X13	0x0	Hex	Dec	FP		0x0				Hex	Dec
X14	0x0	Hex	Dec	LR		0x0				Hex	Dec
X15	0x0	Hex	Dec	XZR		0x0				Hex	Dec

The diagram illustrates the internal architecture of the LEGv8 processor. It shows the flow of data and control signals between various components:

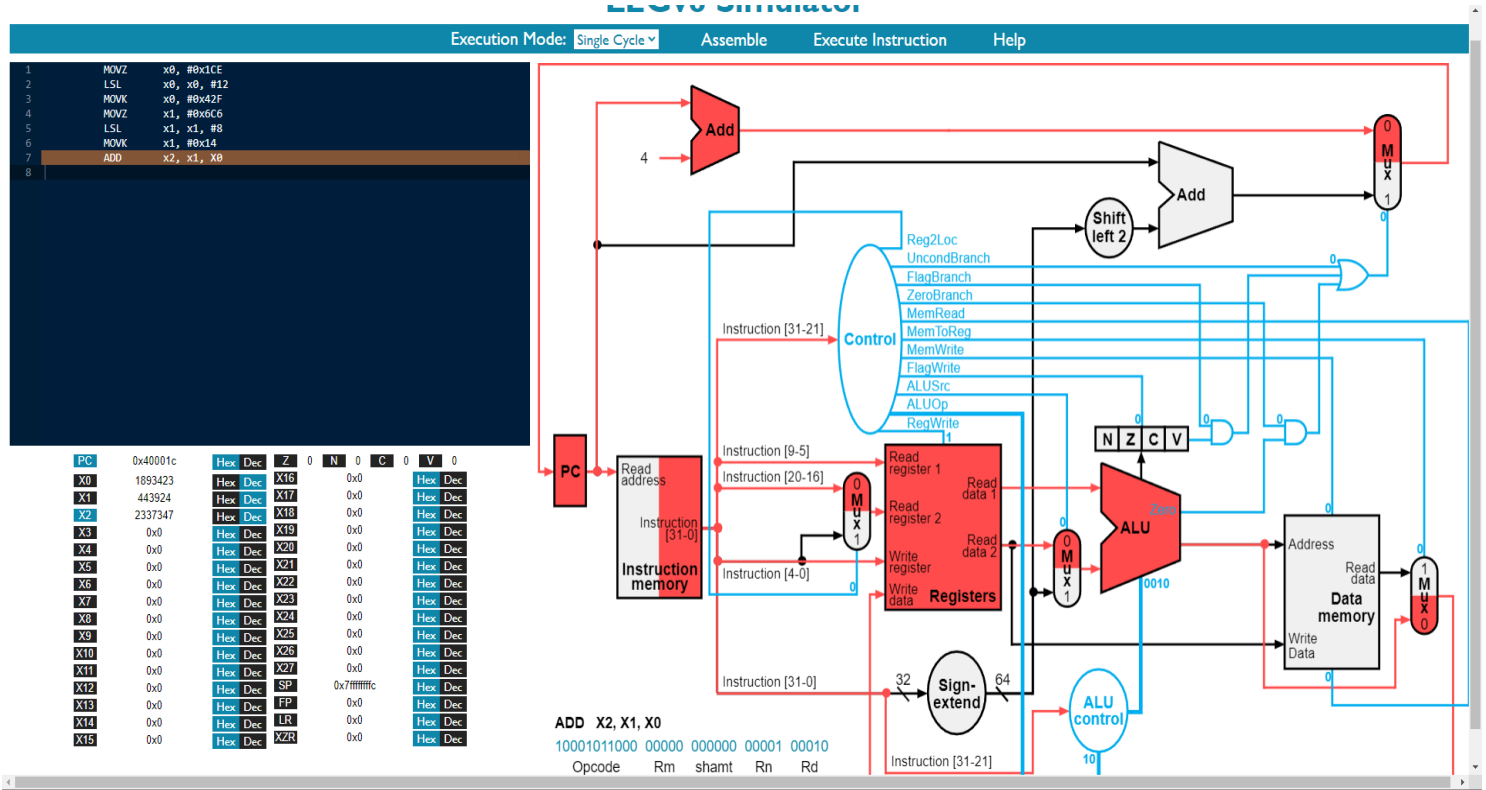
- PC (Program Counter):** Holds the current instruction address (0x40001c).
- Instruction Memory:** Provides instructions to the processor based on the PC value.
- Control Unit:** Decodes the instruction and generates control signals for other components.
- Registers:** A set of 32 registers (X0-X31) used for storing data. The diagram shows registers X0, X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18, X19, X20, X21, X22, X23, X24, X25, X26, X27, SP, FP, LR, and XZR.
- ALU (Arithmetic Logic Unit):** Performs arithmetic and logical operations on data from registers or memory. It includes an ALU control unit and a 64-bit ALU.
- Data Memory:** Stores data and is accessed via the ALU and registers.
- Shifters:** Perform shift operations on data from registers.
- Flags:** Status flags (N, Z, C, V) that are updated based on the ALU results.

ADDI X0, X6, #25
 1001000100 000000011001 00110 00000
 Opcode ALU_immediate Rn Rd

Printed Code, resulting in 221dec stored in register x0:

```
MOVZ    x1, #5
LSL     x2, x1, #2
MOVZ    x3, #11
LSL     x4, x3, #4
ADD     x6, x2, x4
ADDI    x0, x6, #25
```

Task 4:



Printed Code, resulting in 2337347dec stored in register x2:

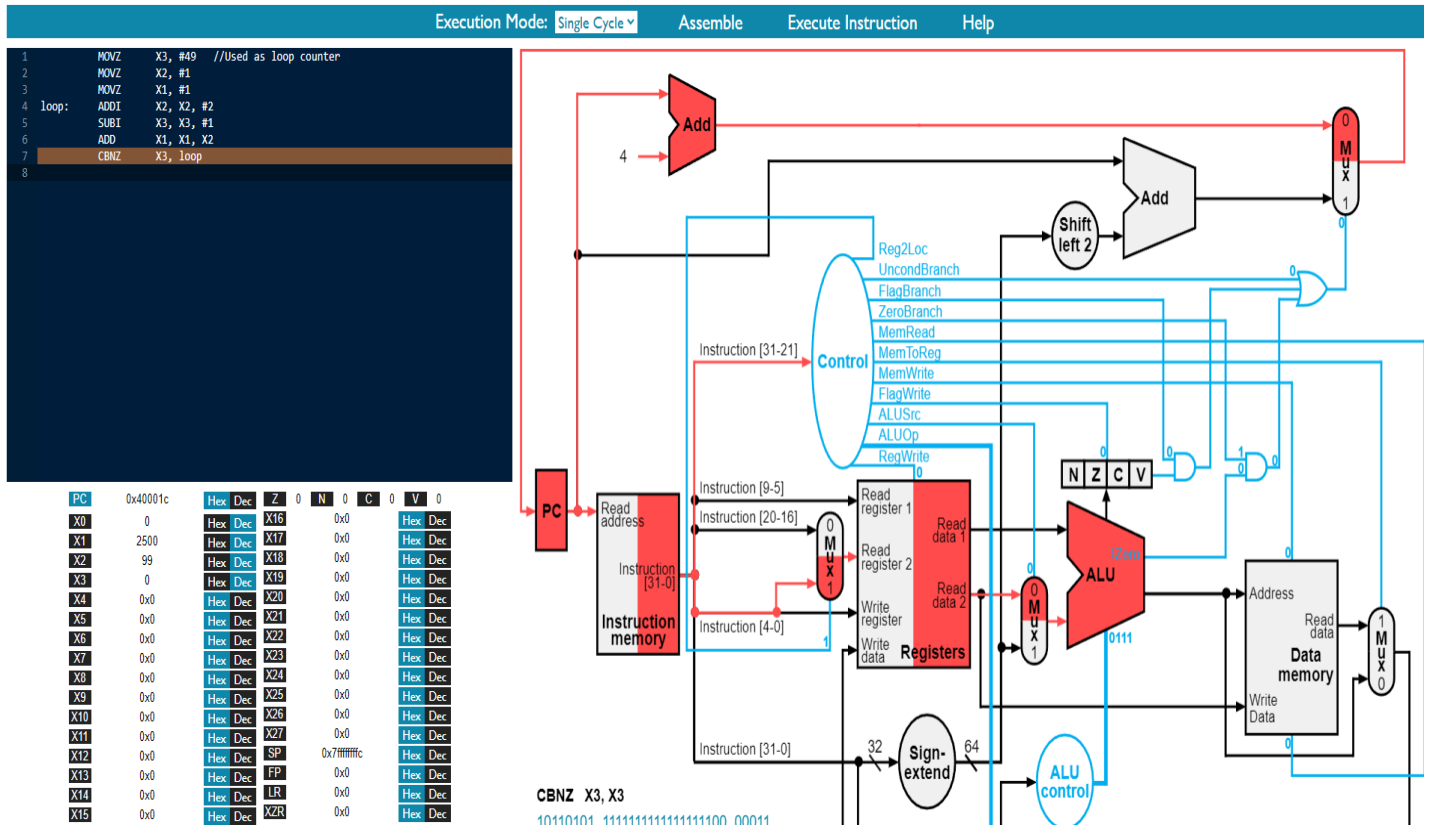
```

MOVZ    x0, #0x1CE
LSL     x0, x0, #12
MOVK    x0, #0x42F
MOVZ    x1, #0x6C6
LSL     x1, x1, #8
MOVK    x1, #0x14
ADD     x2, x1, x0

```

Task 5:

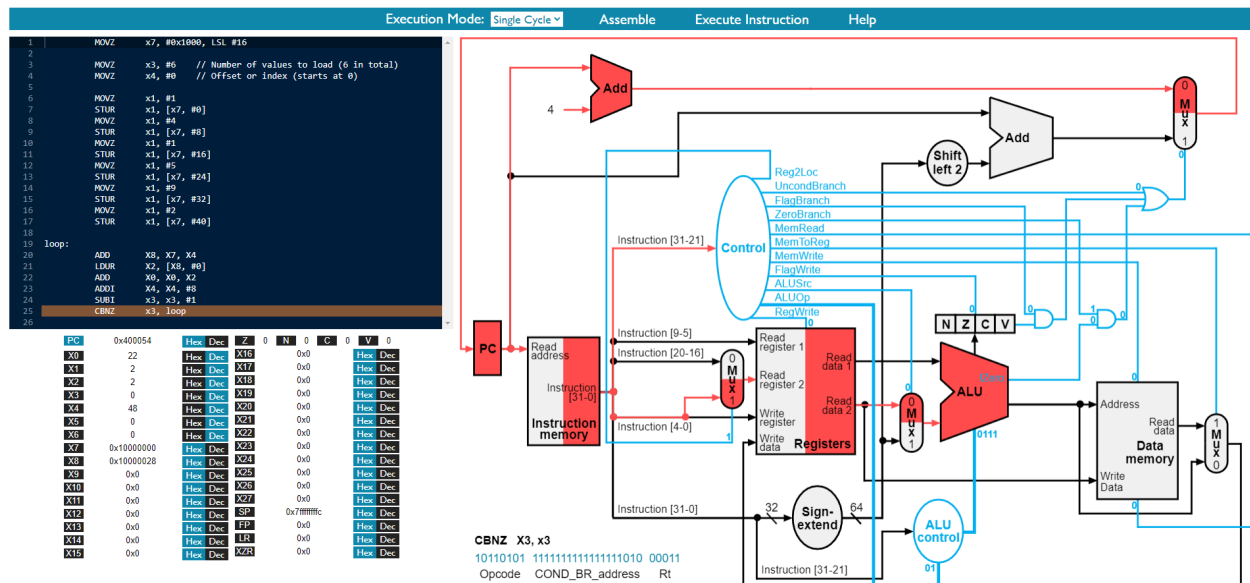
LEGv8 Simulator



Printed Code, resulting in 2500dec stored in register x1:

```
MOVZ X3, #49 // loop counter
MOVZ X2, #1
MOVZ X1, #1
loop: ADDI X2, X2, #2
      SUBI X3, X3, #1
      ADD  X1, X1, X2
      CBNZ X3, loop
```

Task6:



Printed code, resulting in 22dec stored in register x0.

```
MOVZ x7, #0x1000, LSL #16
MOVZ x3, #6 // Number of values to load (6 in total)
MOVZ x4, #0 // Offset or index (starts at 0)
```

```
MOVZ x1, #1
STUR x1, [x7, #0]
MOVZ x1, #4
STUR x1, [x7, #8]
MOVZ x1, #1
STUR x1, [x7, #16]
MOVZ x1, #5
STUR x1, [x7, #24]
MOVZ x1, #9
STUR x1, [x7, #32]
MOVZ x1, #2
STUR x1, [x7, #40]
```

loop:

```
ADD X8, X7, X4
LDUR X2, [X8, #0]
ADD X0, X0, X2
ADDI X4, X4, #8
SUBI x3, x3, #1
CBNZ x3, loop
```

