Maseeh College of Engineering and Computer Science

PORTLAND STATE UNIVERSITY

Project Report

ECE_586_WINTER2023

RISC-V Instruction Set Architecture Simulator

By:

Adithya Rajesh
Sanket Patil
Viraj Pashte
Zoheb Mohammed Anis Mir

Portland State
UNIVERSITY

Introduction

RISC-V is an open-source, modular Instruction Set Architecture (ISA) that is gaining significant attention in the computing industry. It is designed to be simple, elegant, and extensible, allowing for customization and specialization for different applications. The RISC-V 32I base integer instruction set architecture is the most fundamental version of the RISC-V ISA, providing a basic set of instructions for implementing a general-purpose computer. This instruction set includes 32 registers, basic arithmetic, logical, and memory instructions, and control transfer instructions. The simplicity and flexibility of the RISC-V 32I make it an ideal choice for embedded systems, microcontrollers, and other low-power devices.

In addition to the base integer instruction set architecture (32I), RISC-V also includes M and F type instructions. The M extension adds integer multiplication and division operations, while the F extension adds support for single-precision floating-point operations. These extensions are optional and can be added to the base integer instruction set as needed for specific applications. The M and F type instructions expand the capabilities of RISC-V, making it suitable for a wider range of applications, including scientific computing, graphics processing, and signal processing. The modularity of the RISC-V ISA allows for custom instruction sets to be designed and added, which can provide significant performance gains for specific tasks.

Portland State UNIVERSITY

Specifications:

Base ISA:

RV32I

31 27 26 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type
imm[11:0]		rs1	funct3	rd	opcode	I-type
imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	S-type
imm[12|10:5]	rs2	rs1	funct3	imm[4:1|11]	opcode	B-type
imm[31:12]				rd	opcode	U-type
imm[20|10:1|11|19:12]				rd	opcode	J-type

RV32I Base Instruction Set

imm[31:12]				rd	0110111	LUI
imm[31:12]				rd	0010111	AUIPC
imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[11:0]		rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	BLTU
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	BGEU
imm[11:0]		rs1	000	rd	0000011	LB
imm[11:0]		rs1	001	rd	0000011	LH
imm[11:0]		rs1	010	rd	0000011	LW
imm[11:0]		rs1	100	rd	0000011	LBU
imm[11:0]		rs1	101	rd	0000011	LHU
imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	SB
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	SH
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	SW
imm[11:0]		rs1	000	rd	0010011	ADDI
imm[11:0]		rs1	010	rd	0010011	SLTI
imm[11:0]		rs1	011	rd	0010011	SLTIU
imm[11:0]		rs1	100	rd	0010011	XORI
imm[11:0]		rs1	110	rd	0010011	ORI
imm[11:0]		rs1	111	rd	0010011	ANDI
0000000	shamt	rs1	001	rd	0010011	SLLI
0000000	shamt	rs1	101	rd	0010011	SRLI
0100000	shamt	rs1	101	rd	0010011	SRAI
0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND
fm pred	succ	rs1	000	rd	0001111	FENCE
000000000000		00000	000	00000	1110011	ECALL
000000000001		00000	000	00000	1110011	EBREAK

Portland State
UNIVERSITY

RV32M

RV32M Standard Extension

0000001	rs2	rs1	000	rd	0110011	MUL
0000001	rs2	rs1	001	rd	0110011	MULH
0000001	rs2	rs1	010	rd	0110011	MULHSU
0000001	rs2	rs1	011	rd	0110011	MULHU
0000001	rs2	rs1	100	rd	0110011	DIV
0000001	rs2	rs1	101	rd	0110011	DIVU
0000001	rs2	rs1	110	rd	0110011	REM
0000001	rs2	rs1	111	rd	0110011	REMU

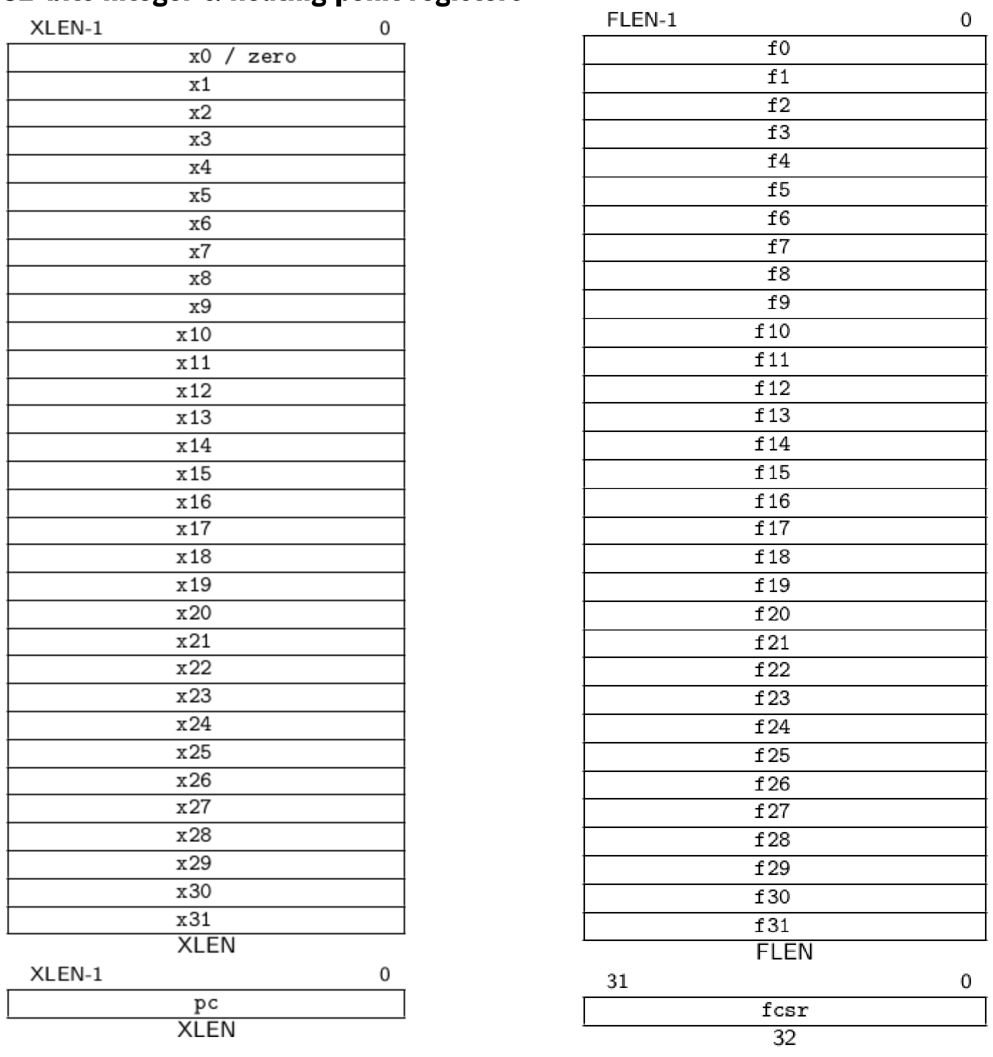
RV32F

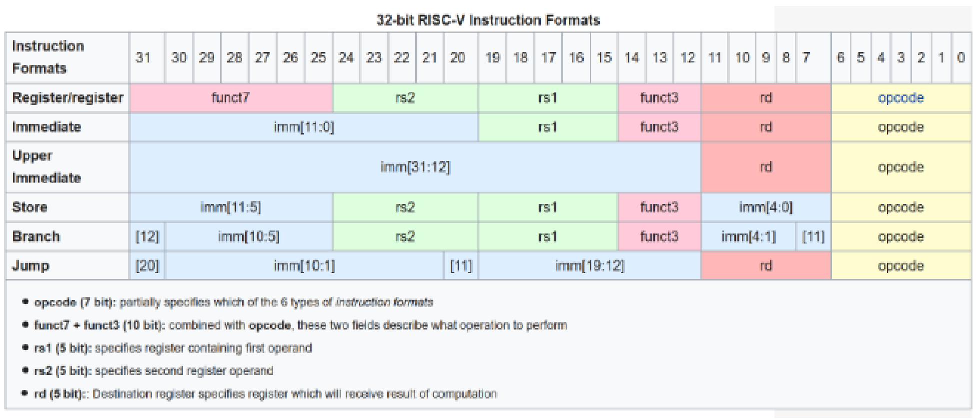
31 27	26 25	24 20	19 15	14 12	11 7	6 0	
funct7		rs2	rs1	funct3	rd	opcode	R-type
rs3	funct2	rs2	rs1	funct3	rd	opcode	R4-type
imm[11:0]			rs1	funct3	rd	opcode	I-type
imm[11:5]		rs2	rs1	funct3	imm[4:0]	opcode	S-type

RV32F Standard Extension

imm[11:0]			rs1	010	rd	0000111	FLW
imm[11:5]		rs2	rs1	010	imm[4:0]	0100111	FSW
rs3	00	rs2	rs1	rm	rd	1000011	FMADD.S
rs3	00	rs2	rs1	rm	rd	1000111	FMSUB.S
rs3	00	rs2	rs1	rm	rd	1001011	FNMSUB.S
rs3	00	rs2	rs1	rm	rd	1001111	FNMADD.S
0000000		rs2	rs1	rm	rd	1010011	FADD.S
0000100		rs2	rs1	rm	rd	1010011	FSUB.S
0001000		rs2	rs1	rm	rd	1010011	FMUL.S
0001100		rs2	rs1	rm	rd	1010011	FDIV.S
0101100		00000	rs1	rm	rd	1010011	FSQRT.S
0010000		rs2	rs1	000	rd	1010011	FSGNJ.S
0010000		rs2	rs1	001	rd	1010011	FSGNJN.S
0010000		rs2	rs1	010	rd	1010011	FSGNJX.S
0010100		rs2	rs1	000	rd	1010011	FMIN.S
0010100		rs2	rs1	001	rd	1010011	FMAX.S
1100000		00000	rs1	rm	rd	1010011	FCVT.W.S
1100000		00001	rs1	rm	rd	1010011	FCVT.WU.S
1110000		00000	rs1	000	rd	1010011	FMV.X.W
1010000		rs2	rs1	010	rd	1010011	FEQ.S
1010000		rs2	rs1	001	rd	1010011	FLT.S
1010000		rs2	rs1	000	rd	1010011	FLE.S
1110000		00000	rs1	001	rd	1010011	FCLASS.S
1101000		00000	rs1	rm	rd	1010011	FCVT.S.W
1101000		00001	rs1	rm	rd	1010011	FCVT.S.WU
1111000		00000	rs1	000	rd	1010011	FMV.W.X

32-bits integer & floating point registers





32-bit RISC-V Instruction Formats

Instruction Formats	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register/register	funct7							rs2					rs1					funct3			rd					opcode						
Immediate	imm[11:0]												rs1					funct3			rd					opcode						
Upper Immediate	imm[31:12]																				rd					opcode						
Store	imm[11:5]							rs2					rs1					funct3			imm[4:0]					opcode						
Branch	[12]	imm[10:5]						rs2					rs1					funct3			imm[4:1]				[11]	opcode						
Jump	[20]	imm[10:1]										[11]	imm[19:12]								rd					opcode						

- **opcode (7 bit):** partially specifies which of the 6 types *of instruction formats*
- **funct7 + funct3 (10 bit):** combined with **opcode**, these two fields describe what operation to perform
- **rs1 (5 bit):** specifies register containing first operand
- **rs2 (5 bit):** specifies second register operand
- **rd (5 bit):**: Destination register specifies register which will receive result of computation

Portland State UNIVERSITY

Modes:

- Normal Mode:

  It is the PC and hexadecimal value of each instruction as it's fetched along with the contents (in hexadecimal) of each register after the instruction's execution.

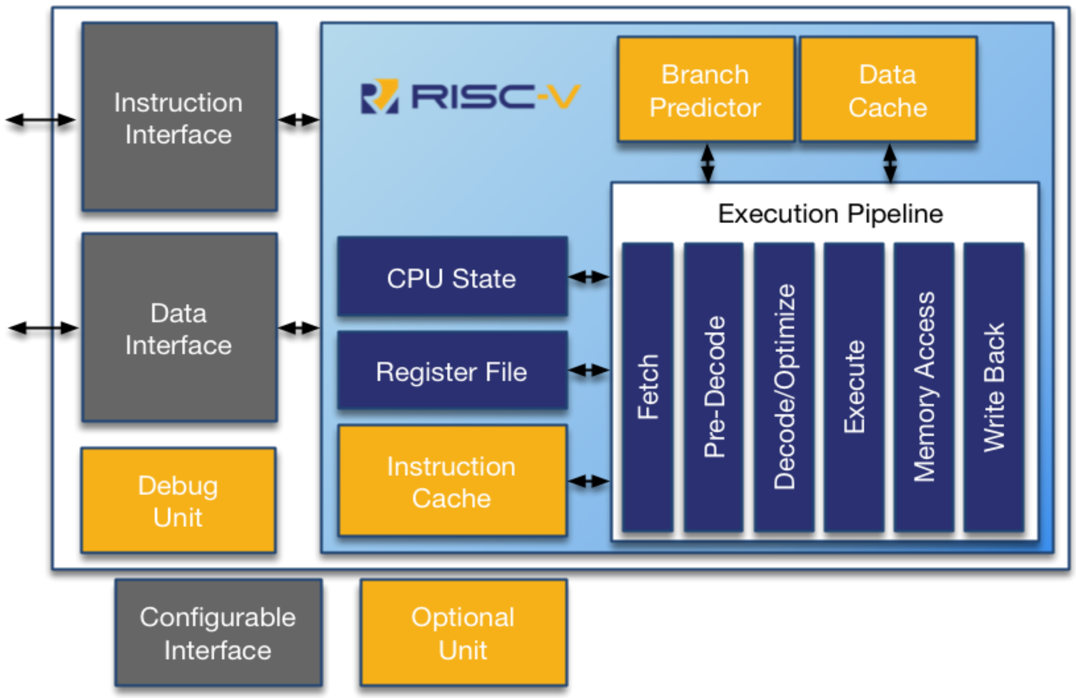
- Silent Mode:

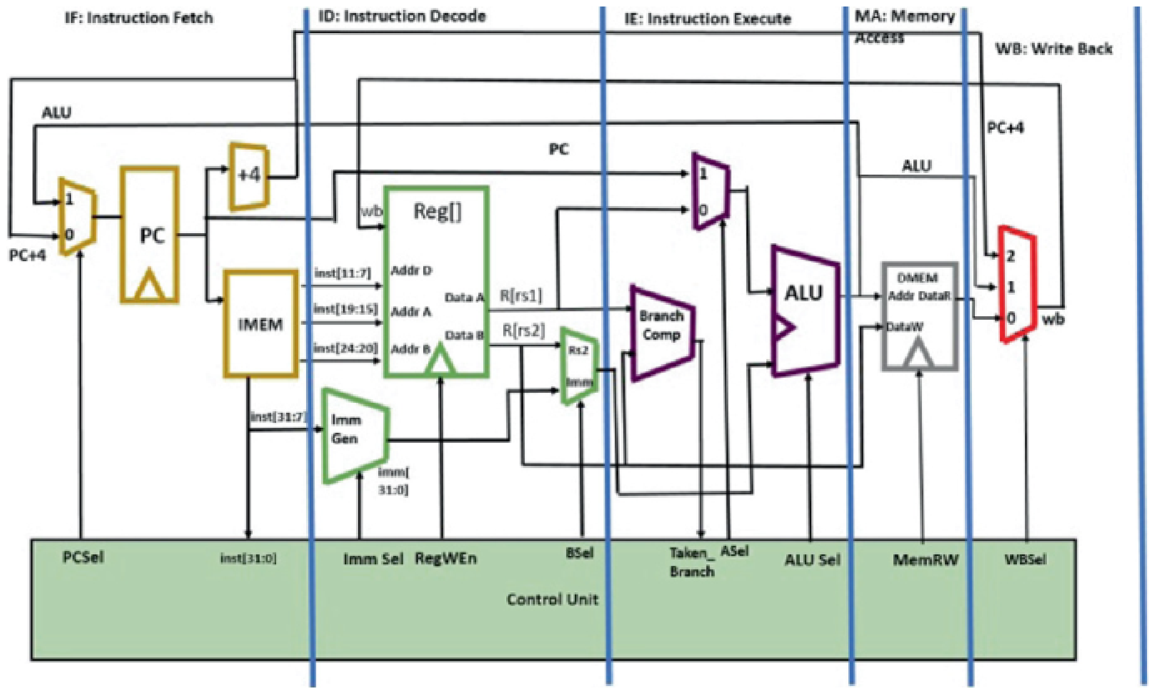
  It prints the PC of the final instruction and hexadecimal value of each register by the end of the simulation.

- Debug Mode:

  Provides the ability to single step and print contents of register, memory, display current instruction and move forward one by one.

Design:


Instruction Interface
Data Interface
Debug Unit
RISC-V
Branch Predictor
Data Cache
Execution Pipeline
CPU State
Register File
Instruction Cache
Fetch
Pre-Decode
Decode/Optimize
Execute
Memory Access
Write Back
Configurable Interface
Optional Unit


IF: Instruction Fetch
ID: Instruction Decode
IE: Instruction Execute
MA: Memory Access
WB: Write Back
ALU
PC+4
PC
+4
IMEM
wb
Reg[]
inst[11:7]
inst[19:15]
inst[24:20]
Addr D
Addr A
Addr B
Data A
Data B
R[rs1]
R[rs2]
inst[31:7]
Imm Gen
imm[31:0]
Rs2
Imm
Branch Comp
ALU
DMEM
Addr DataR
DataW
wb
PCSel
inst[31:0]
Imm Sel
RegWEn
BSel
Taken_Branch
ASel
ALU Sel
MemRW
WBSel
Control Unit

Portland State
UNIVERSITY

<u>Github repository link:</u>

https://github.com/adrajesh/ece526_finalproj_g2.git

<u>Execution commands:</u>

1)
   ```
   rvgcc -S -fpic -march=rv32g -mabi=ilp32 prog.c
   ```
   (here we use the rv32g to indicate general. This will run the M and F instructions and compile the assembly using correct instructions. If you want to run only the I instructions, use rv32i)

2)
   ```
   rvas -ahld prog.s -o prog.o
   ```

3)
   ```
   rvobjdump -d prog.o | grep -o
   ^[[:blank:]]*[[:xdigit:]]*:[[:blank:]][[:xdigit:]]*' > prog.mem
   ```

4)
   ```
   g++ riscv_simul.cpp -o out
   ```

5)
   ```
   ./out prog.mem
   ```

Test Plan

1. AddSubAndOrXor.mem

The above memory image file tests the following instructions

- ADD
- SUB
- AND
- OR
- XOR

For the ADD instruction we check the result when we add

- 2 positive integers
- 2 negative integers
- 1 positive integer and 1 negative integer

For the SUB instruction we check the result when we subtract

- 2 positive integers
- 2 negative integers
- 1 positive integer and 1 negative integer

For AND,OR and XOR - since it is a bitwise operator, we just check the result for various values.

2. SLL_SRL_SRA_SLT_SLTU.mem

The above memory image file tests the following instructions

- SLL
- SRL
- SRA
- SLT
- SLTU

For the SLL instruction we check the shift left for a

- positive integer (6 < 1 is 12)
- negative integer (-6 < 1 is -12)

For the SRL instruction we check the logical shift right for a

- Positive integer (6 > 1 is 3)
- Negative integer (-6 > 1 is 7FFFFFFD)
  as logical shift does not preserve sign

For the SRA instruction we check the arithmetic shift right for a

- Positive integer (6 > 1 is 3)
- Negative integer (-6 > 1 is FFFFFFFD)
  as arithmetic shift preserves sign

For the SLT instruction we check if rs1 < rs2 for

- 2 positive integers
- 2 negative integers
- 1 positive integer and 1 negative integer

For the SLTU instruction we check if rs1 < rs2 for

- 2 positive integers
- 2 negative integers (here even though we load -ve integers, would be treated as +ve)
- 1 positive integer and 1 negative integer (negative number would be treated as +ve)

3. **SW_LW.mem**

The above memory image file tests the following instructions

- SW
- LW

We check if SW stores the contents of registers to the memory location specified.
We also check if LW retrieves the correct contents from memory
If we give an unaligned reference to memory during LW, the program execution stops

4. **SH_LH_LHU.mem**

The above memory image file tests the following instructions

- SH
- LH
- LHU

We check if SH stores the contents of registers to the memory location specified.
We also check if LH and LHU retrieve the correct contents from memory.
In case of LH, it is sign extended.
In case of LHU, it is zero extended.
If we give an unaligned reference to memory during LH or LHU, the program execution stops

5. **SB_LB_LBU.mem**

The above memory image file tests the following instructions

- SB
- LB
- LBU

We check if SB stores the contents of registers to the memory location specified.
We also check if LB and LBU retrieve the correct contents from memory.
In the case of LB, it is sign extended.
In the case of LBU, it is zero extended.

6. **ADDI_ANDI_ORI_XORI.mem**

The above memory image file tests the following instructions

- ADDI
- ANDI
- ORI
- XOR

For all the above instructions we check with result with immediates which is

- Positive integer
- Negative integer
- Extreme values in the range

7. **SLLI_SRLI_SRAI_SLTI_SLTIU.mem**

The above memory image file tests the following instructions

- SLLI
- SRLI
- SRAI
- SLTI
- SLTIU

For the SLLI,SRLI and SRAI instruction we check the shift left for a

- positive integer
- negative integer

For SLTI and STLIU instructions we check the result when comparing

- 2 positive integers
- 2 negative integers
- 1 positive and 1 negative integer
- same integers

In case of SLTIU, even though we load -ve integers, it would be treated as positive.

8. **JALR.mem**

The above memory image file tests the JALR and AUIPC instruction. We create a nested function and make the main function call function1, which in turn calls function2 and then we check if it executes both the functions properly and returns the execution back to the main.

9. **loop.mem**

A simple for loop to iterate through and check if it jumps to the correct location every time. Once the loop is done executing, it finishes the rest and stops program execution.

10. <u>BEQ.mem</u>

This memory image file tests the BEQ instruction. We load 2 registers a4 and a5 with 2 values and check the branching as below

- Branch taken when 2 values are equal (positive integers)
- Branch taken when 2 values are equal (negative integers)
- Branch not taken when 2 values are not equal

11. <u>BNE.mem</u>

This memory image file tests the BNE instruction. We load 2 registers a4 and a5 with 2 values and check the branching as below

- Branch taken when 2 values are not equal (positive integers)
- Branch taken when 2 values are not equal (negative integers)
- Branch not taken when 2 values are equal

12. <u>BLT.mem</u>

This memory image file tests the BLT instruction. We load 2 registers a4 and a5 with 2 values and check the branching as below

- Branch taken when a4 < a5
- Branch not taken when a4 > a5
- Branch not taken when a4 == a5

13. <u>BGE.mem</u>

This memory image file tests the BGE instruction. We load 2 registers a4 and a5 with 2 values and check the branching as below

- Branch taken when a4 > a5
- Branch not taken when a4 < a5
- Branch taken when a4 == a5

14. <u>BLTU.mem</u>

This memory image file tests the BLTU instruction. We load 2 registers a4 and a5 with 2 values and chek the branching as below

- Branch taken when a4<a5 (for same and different MSB)
- Branch not taken when a4>a5 (for same and different MSB)
- Branch not taken when a4 == a5

Portland State University

15. BGEU.mem

This memory image file tests the BGEU instruction. We load 2 registers a4 and a5 with 2 values and check the branching as below

- Branch taken when a4 > a5 (for same and different MSB)
- Branch not taken when a4 < a5 (for same and different MSB)
- Branch taken when a4 == a5

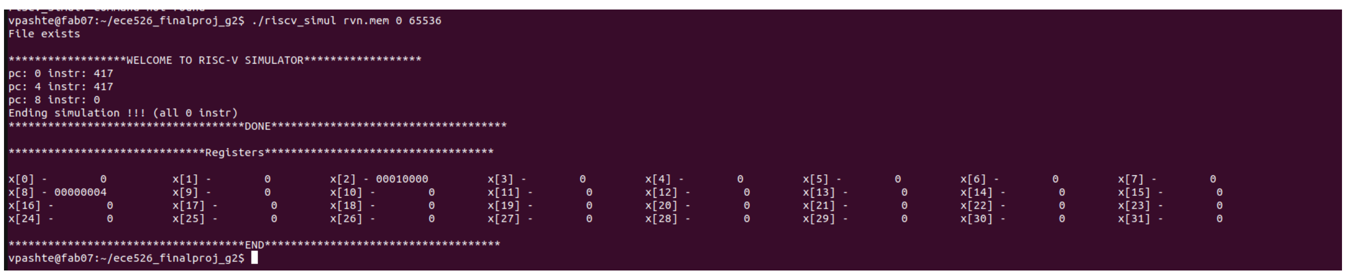
This also tests the LUI instruction as we try to load the max value possible and check if it stores properly in the destination registers

Instr	**Opcode values in hex (decimal)**	**Testing (Y/N)**	**Test file**
LUI	37 (55)	Y	BGEU.mem
AUIPC	17 (23)	Y	JALR.mem
JAL	6F (111)	Y	loop.mem
JALR	67 (103)	Y	JALR.mem
BEQ	63 (99)	Y	BEQ.mem
BNE	63 (99)	Y	BNE.mem
BLT	63 (99)	Y	BLT.mem
BGE	63 (99)	Y	BGE.mem
BLTU	63 (99)	Y	BLTU.mem
BGEU	63 (99)	Y	BGEU.mem
LB	3 (3)	Y	SB_LB_LBU.mem
LH	3 (3)	Y	SH_LH_LHU.mem
LW	3 (3)	Y	SW_LW.mem
LBU	3 (3)	Y	SB_LB_LBU.mem
LHU	3 (3)	Y	SH_LH_LHU.mem

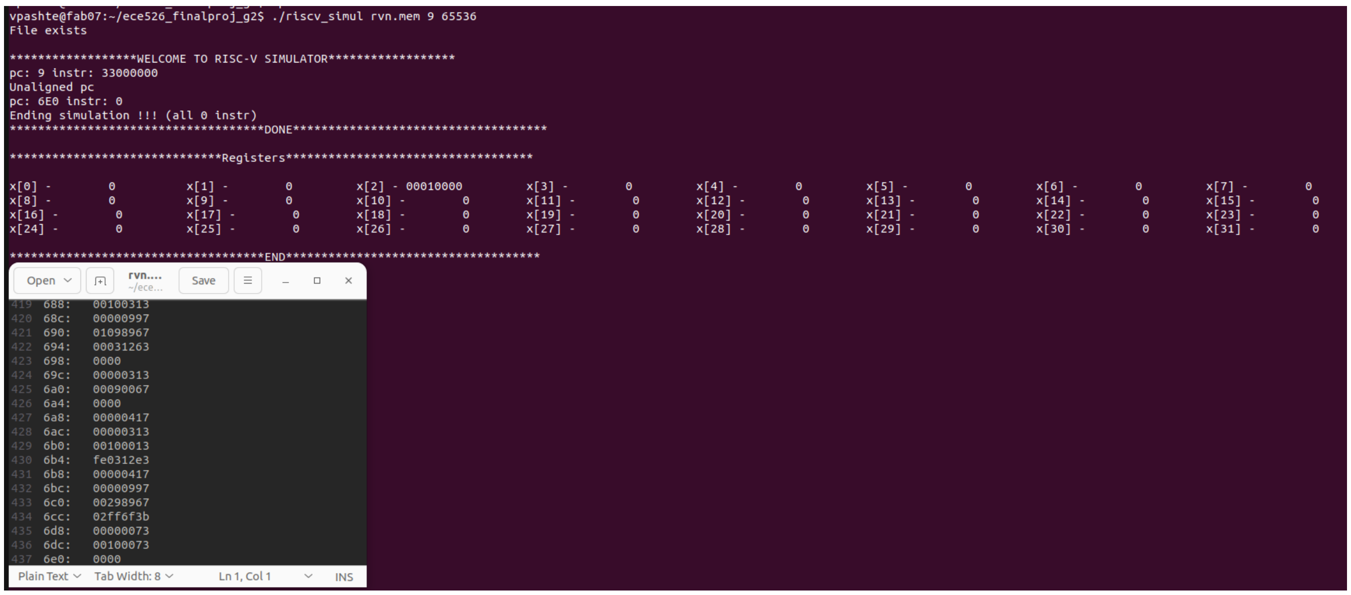
SB	23 (35)	Y	SB_LB_LBU.mem
SH	23 (35)	Y	SH_LH_LHU.mem
SW	23 (35)	Y	SW_LW.mem
ADDI	13 (19)	Y	ADDI_ANDI_ORI_XORI.mem
SLTI	13 (19)	Y	SLLI_SRLI_SRAI_SLTI_SLTIU.mem
SLTIU	13 (19)	Y	SLLI_SRLI_SRAI_SLTI_SLTIU.mem
XORI	13 (19)	Y	ADDI_ANDI_ORI_XORI.mem
ORI	13 (19)	Y	ADDI_ANDI_ORI_XORI.mem
ANDI	13 (19)	Y	ADDI_ANDI_ORI_XORI.mem
SLLI	13 (19)	Y	SLLI_SRLI_SRAI_SLTI_SLTIU.mem
SRLI	13 (19)	Y	SLLI_SRLI_SRAI_SLTI_SLTIU.mem
SRAI	13 (19)	Y	SLLI_SRLI_SRAI_SLTI_SLTIU.mem
ADD	33 (51)	Y	AddSubAndOrXor.mem
SUB	33 (51)	Y	AddSubAndOrXor.mem
SLL	33 (51)	Y	SLL_SRL_SRA_SLT_SLTU.mem
SLT	33 (51)	Y	SLL_SRL_SRA_SLT_SLTU.mem
SLTU	33 (51)	Y	SLL_SRL_SRA_SLT_SLTU.mem
XOR	33 (51)	Y	AddSubAndOrXor.mem
SRL	33 (51)	Y	SLL_SRL_SRA_SLT_SLTU.mem
SRA	33 (51)	Y	SLL_SRL_SRA_SLT_SLTU.mem
OR	33 (51)	Y	AddSubAndOrXor.mem
AND	33 (51)	Y	AddSubAndOrXor.mem
ECALL	73 (115)		

Output:

1. Silent Mode



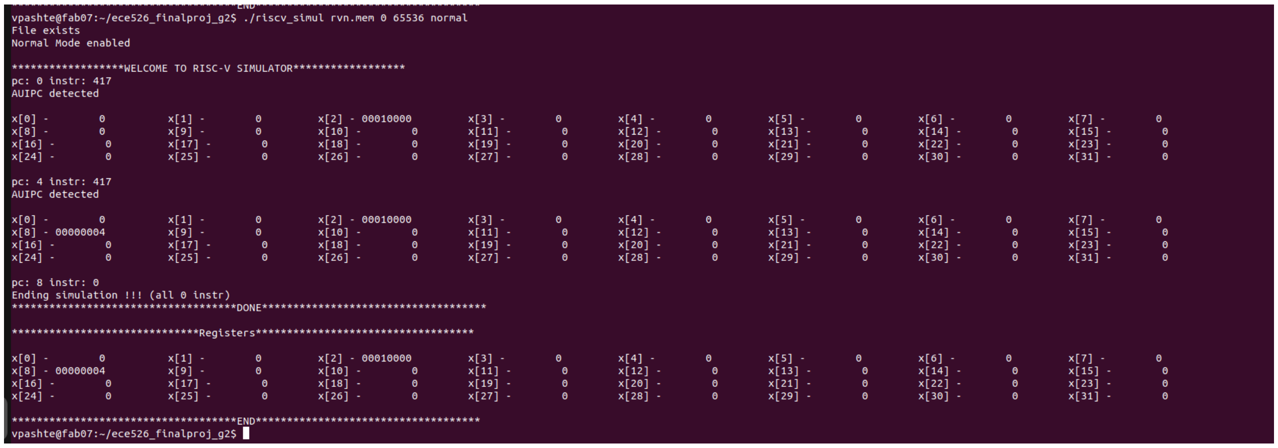
```
vpashte@fab07:~/ece526_finalproj_g2$ ./riscv_simul rvn.mem 0 65536
File exists

*****************WELCOME TO RISC-V SIMULATOR*****************
pc: 0 instr: 417
pc: 4 instr: 417
pc: 8 instr: 0
Ending simulation !!! (all 0 instr)
************************************DONE************************************

******************************Registers**********************************

x[0] -        0          x[1] -        0        x[2] - 00010000        x[3] -        0        x[4] -        0        x[5] -        0        x[6] -        0        x[7] -        0
x[8] - 00000004          x[9] -        0        x[10] -        0       x[11] -        0       x[12] -        0       x[13] -        0       x[14] -        0       x[15] -        0
x[16] -        0         x[17] -        0       x[18] -        0       x[19] -        0       x[20] -        0       x[21] -        0       x[22] -        0       x[23] -        0
x[24] -        0         x[25] -        0       x[26] -        0       x[27] -        0       x[28] -        0       x[29] -        0       x[30] -        0       x[31] -        0

************************************END************************************
vpashte@fab07:~/ece526_finalproj_g2$ 
```



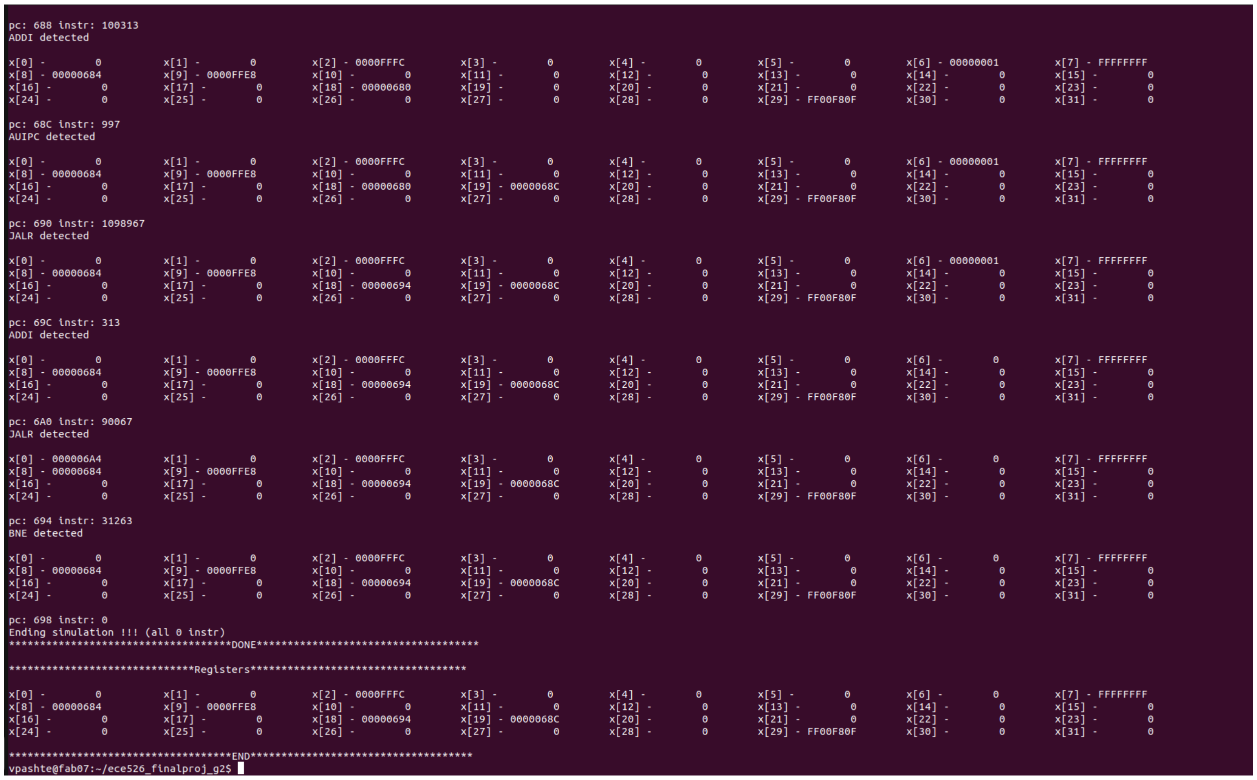
```
vpashte@fab07:~/ece526_finalproj_g2$ ./riscv_simul rvn.mem 9 65536
File exists

*****************WELCOME TO RISC-V SIMULATOR*****************
pc: 9 instr: 33000000
Unaligned pc
pc: 6E0 instr: 0
Ending simulation !!! (all 0 instr)
************************************DONE************************************

******************************Registers**********************************

x[0] -        0          x[1] -        0        x[2] - 00010000        x[3] -        0        x[4] -        0        x[5] -        0        x[6] -        0        x[7] -        0
x[8] -        0          x[9] -        0        x[10] -        0       x[11] -        0       x[12] -        0       x[13] -        0       x[14] -        0       x[15] -        0
x[16] -        0         x[17] -        0       x[18] -        0       x[19] -        0       x[20] -        0       x[21] -        0       x[22] -        0       x[23] -        0
x[24] -        0         x[25] -        0       x[26] -        0       x[27] -        0       x[28] -        0       x[29] -        0       x[30] -        0       x[31] -        0

************************************END************************************
```

```
688:   00100313
68c:   00000997
690:   01098967
694:   00031263
698:   0000
69c:   00000313
6a0:   00090067
6a4:   0000
6a8:   00000417
6ac:   00000313
6b0:   00100013
6b4:   fe0312e3
6b8:   00000417
6bc:   00000997
6c0:   00298967
6cc:   02ff6f3b
6d8:   00000073
6dc:   00100073
6e0:   0000
```

2. Normal Mode

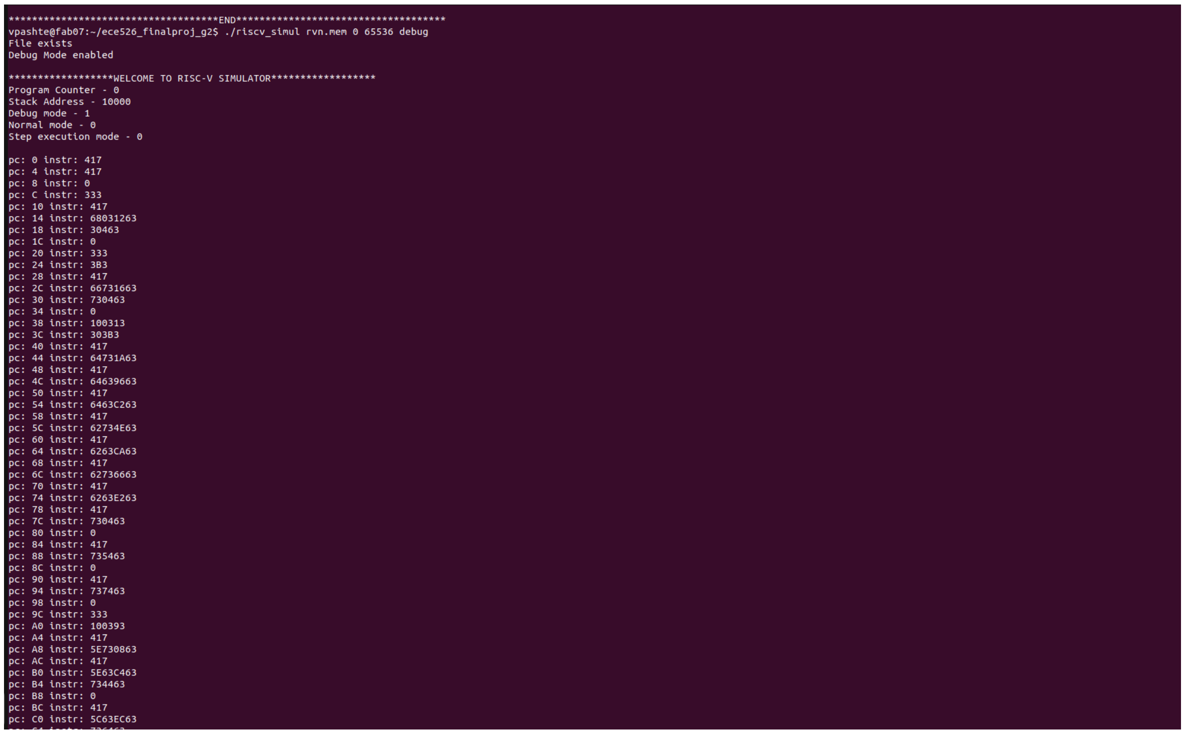


```
************************************END************************************
vpashte@fab07:~/ece526_finalproj_g2$ ./riscv_simul rvn.mem 0 65536 normal
File exists
Normal Mode enabled

*****************WELCOME TO RISC-V SIMULATOR*****************
pc: 0 instr: 417
AUIPC detected

x[0] -        0     x[1] -        0     x[2] - 00010000     x[3] -        0     x[4] -        0     x[5] -        0     x[6] -        0     x[7] -        0
x[8] -        0     x[9] -        0     x[10] -        0    x[11] -        0    x[12] -        0    x[13] -        0    x[14] -        0    x[15] -        0
x[16] -        0    x[17] -        0    x[18] -        0    x[19] -        0    x[20] -        0    x[21] -        0    x[22] -        0    x[23] -        0
x[24] -        0    x[25] -        0    x[26] -        0    x[27] -        0    x[28] -        0    x[29] -        0    x[30] -        0    x[31] -        0

pc: 4 instr: 417
AUIPC detected

x[0] -        0     x[1] -        0     x[2] - 00010000     x[3] -        0     x[4] -        0     x[5] -        0     x[6] -        0     x[7] -        0
x[8] - 00000004     x[9] -        0     x[10] -        0    x[11] -        0    x[12] -        0    x[13] -        0    x[14] -        0    x[15] -        0
x[16] -        0    x[17] -        0    x[18] -        0    x[19] -        0    x[20] -        0    x[21] -        0    x[22] -        0    x[23] -        0
x[24] -        0    x[25] -        0    x[26] -        0    x[27] -        0    x[28] -        0    x[29] -        0    x[30] -        0    x[31] -        0

pc: 8 instr: 0
Ending simulation !!! (all 0 instr)
************************************DONE************************************

*****************************Registers*********************************

x[0] -        0     x[1] -        0     x[2] - 00010000     x[3] -        0     x[4] -        0     x[5] -        0     x[6] -        0     x[7] -        0
x[8] - 00000004     x[9] -        0     x[10] -        0    x[11] -        0    x[12] -        0    x[13] -        0    x[14] -        0    x[15] -        0
x[16] -        0    x[17] -        0    x[18] -        0    x[19] -        0    x[20] -        0    x[21] -        0    x[22] -        0    x[23] -        0
x[24] -        0    x[25] -        0    x[26] -        0    x[27] -        0    x[28] -        0    x[29] -        0    x[30] -        0    x[31] -        0

************************************END************************************
vpashte@fab07:~/ece526_finalproj_g2$
```

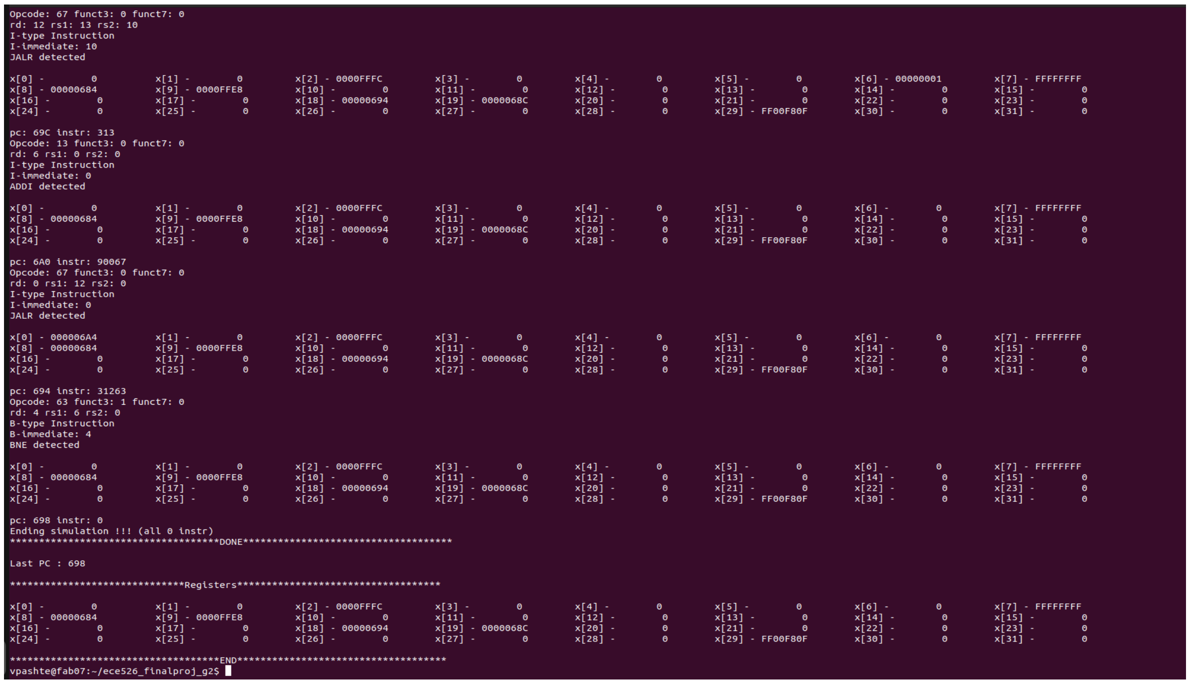


```
pc: 688 instr: 100313
ADDI detected

x[0] -        0     x[1] -        0     x[2] - 0000FFFC     x[3] -        0     x[4] -        0     x[5] -        0     x[6] - 00000001     x[7] - FFFFFFFF
x[8] - 00000684     x[9] - 0000FFE8     x[10] -        0    x[11] -        0    x[12] -        0    x[13] -        0    x[14] -        0    x[15] -        0
x[16] -        0    x[17] -        0    x[18] - 00000680    x[19] -        0    x[20] -        0    x[21] -        0    x[22] -        0    x[23] -        0
x[24] -        0    x[25] -        0    x[26] -        0    x[27] -        0    x[28] -        0    x[29] - FF00F80F    x[30] -        0    x[31] -        0

pc: 68C instr: 997
AUIPC detected

x[0] -        0     x[1] -        0     x[2] - 0000FFFC     x[3] -        0     x[4] -        0     x[5] -        0     x[6] - 00000001     x[7] - FFFFFFFF
x[8] - 00000684     x[9] - 0000FFE8     x[10] -        0    x[11] -        0    x[12] -        0    x[13] -        0    x[14] -        0    x[15] -        0
x[16] -        0    x[17] -        0    x[18] - 00000680    x[19] - 0000068C    x[20] -        0    x[21] -        0    x[22] -        0    x[23] -        0
x[24] -        0    x[25] -        0    x[26] -        0    x[27] -        0    x[28] -        0    x[29] - FF00F80F    x[30] -        0    x[31] -        0

pc: 690 instr: 1098967
JALR detected

x[0] -        0     x[1] -        0     x[2] - 0000FFFC     x[3] -        0     x[4] -        0     x[5] -        0     x[6] - 00000001     x[7] - FFFFFFFF
x[8] - 00000684     x[9] - 0000FFE8     x[10] -        0    x[11] -        0    x[12] -        0    x[13] -        0    x[14] -        0    x[15] -        0
x[16] -        0    x[17] -        0    x[18] - 00000694    x[19] - 0000068C    x[20] -        0    x[21] -        0    x[22] -        0    x[23] -        0
x[24] -        0    x[25] -        0    x[26] -        0    x[27] -        0    x[28] -        0    x[29] - FF00F80F    x[30] -        0    x[31] -        0

pc: 69C instr: 313
ADDI detected

x[0] -        0     x[1] -        0     x[2] - 0000FFFC     x[3] -        0     x[4] -        0     x[5] -        0     x[6] -        0     x[7] - FFFFFFFF
x[8] - 00000684     x[9] - 0000FFE8     x[10] -        0    x[11] -        0    x[12] -        0    x[13] -        0    x[14] -        0    x[15] -        0
x[16] -        0    x[17] -        0    x[18] - 00000694    x[19] - 0000068C    x[20] -        0    x[21] -        0    x[22] -        0    x[23] -        0
x[24] -        0    x[25] -        0    x[26] -        0    x[27] -        0    x[28] -        0    x[29] - FF00F80F    x[30] -        0    x[31] -        0

pc: 6A0 instr: 90067
JALR detected

x[0] - 000006A4     x[1] -        0     x[2] - 0000FFFC     x[3] -        0     x[4] -        0     x[5] -        0     x[6] -        0     x[7] - FFFFFFFF
x[8] - 00000684     x[9] - 0000FFE8     x[10] -        0    x[11] -        0    x[12] -        0    x[13] -        0    x[14] -        0    x[15] -        0
x[16] -        0    x[17] -        0    x[18] - 00000694    x[19] - 0000068C    x[20] -        0    x[21] -        0    x[22] -        0    x[23] -        0
x[24] -        0    x[25] -        0    x[26] -        0    x[27] -        0    x[28] -        0    x[29] - FF00F80F    x[30] -        0    x[31] -        0

pc: 694 instr: 31263
BNE detected

x[0] -        0     x[1] -        0     x[2] - 0000FFFC     x[3] -        0     x[4] -        0     x[5] -        0     x[6] -        0     x[7] - FFFFFFFF
x[8] - 00000684     x[9] - 0000FFE8     x[10] -        0    x[11] -        0    x[12] -        0    x[13] -        0    x[14] -        0    x[15] -        0
x[16] -        0    x[17] -        0    x[18] - 00000694    x[19] - 0000068C    x[20] -        0    x[21] -        0    x[22] -        0    x[23] -        0
x[24] -        0    x[25] -        0    x[26] -        0    x[27] -        0    x[28] -        0    x[29] - FF00F80F    x[30] -        0    x[31] -        0

pc: 698 instr: 0
Ending simulation !!! (all 0 instr)
************************************DONE************************************

*****************************Registers*********************************

x[0] -        0     x[1] -        0     x[2] - 0000FFFC     x[3] -        0     x[4] -        0     x[5] -        0     x[6] -        0     x[7] - FFFFFFFF
x[8] - 00000684     x[9] - 0000FFE8     x[10] -        0    x[11] -        0    x[12] -        0    x[13] -        0    x[14] -        0    x[15] -        0
x[16] -        0    x[17] -        0    x[18] - 00000694    x[19] - 0000068C    x[20] -        0    x[21] -        0    x[22] -        0    x[23] -        0
x[24] -        0    x[25] -        0    x[26] -        0    x[27] -        0    x[28] -        0    x[29] - FF00F80F    x[30] -        0    x[31] -        0

************************************END************************************
vpashte@fab07:~/ece526_finalproj_g2$
```

3. Debug Mode

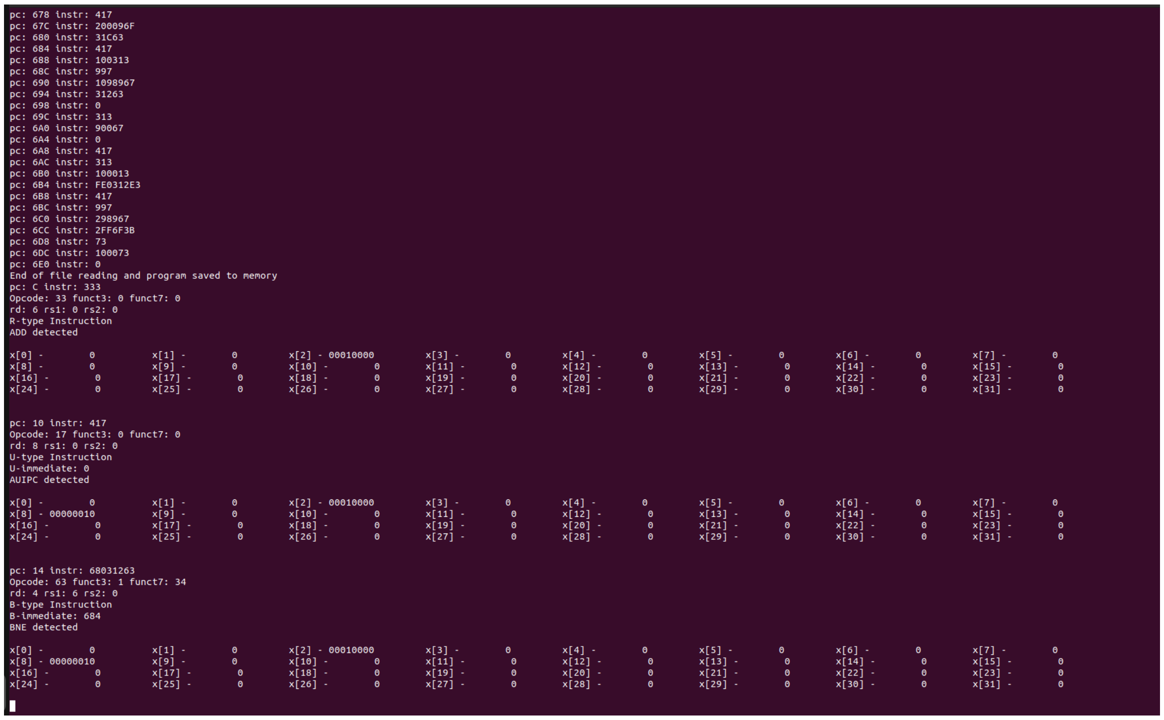


```
************************************END************************************
vpashte@fab07:~/ece526_finalproj_g2$ ./riscv_simul rvn.mem 0 65536 debug
File exists
Debug Mode enabled

*****************WELCOME TO RISC-V SIMULATOR*****************
Program Counter - 0
Stack Address - 10000
Debug mode - 1
Normal mode - 0
Step execution mode - 0

pc: 0 instr: 417
pc: 4 instr: 417
pc: 8 instr: 0
pc: C instr: 333
pc: 10 instr: 417
pc: 14 instr: 68031263
pc: 18 instr: 30463
pc: 1C instr: 0
pc: 20 instr: 333
pc: 24 instr: 3B3
pc: 28 instr: 417
pc: 2C instr: 66731663
pc: 30 instr: 730463
pc: 34 instr: 0
pc: 38 instr: 100313
pc: 3C instr: 303B3
pc: 40 instr: 417
pc: 44 instr: 64731A63
pc: 48 instr: 417
pc: 4C instr: 64639663
pc: 50 instr: 417
pc: 54 instr: 6463C263
pc: 58 instr: 417
pc: 5C instr: 62734E63
pc: 60 instr: 417
pc: 64 instr: 6263CA63
pc: 68 instr: 417
pc: 6C instr: 62736663
pc: 70 instr: 417
pc: 74 instr: 6263E263
pc: 78 instr: 417
pc: 7C instr: 730463
pc: 80 instr: 0
pc: 84 instr: 417
pc: 88 instr: 735463
pc: 8C instr: 0
pc: 90 instr: 417
pc: 94 instr: 737463
pc: 98 instr: 0
pc: 9C instr: 333
pc: A0 instr: 100393
pc: A4 instr: 417
pc: A8 instr: 5E730863
pc: AC instr: 417
pc: B0 instr: 5E63C463
pc: B4 instr: 734463
pc: B8 instr: 0
pc: BC instr: 417
pc: C0 instr: 5C63EC63
```

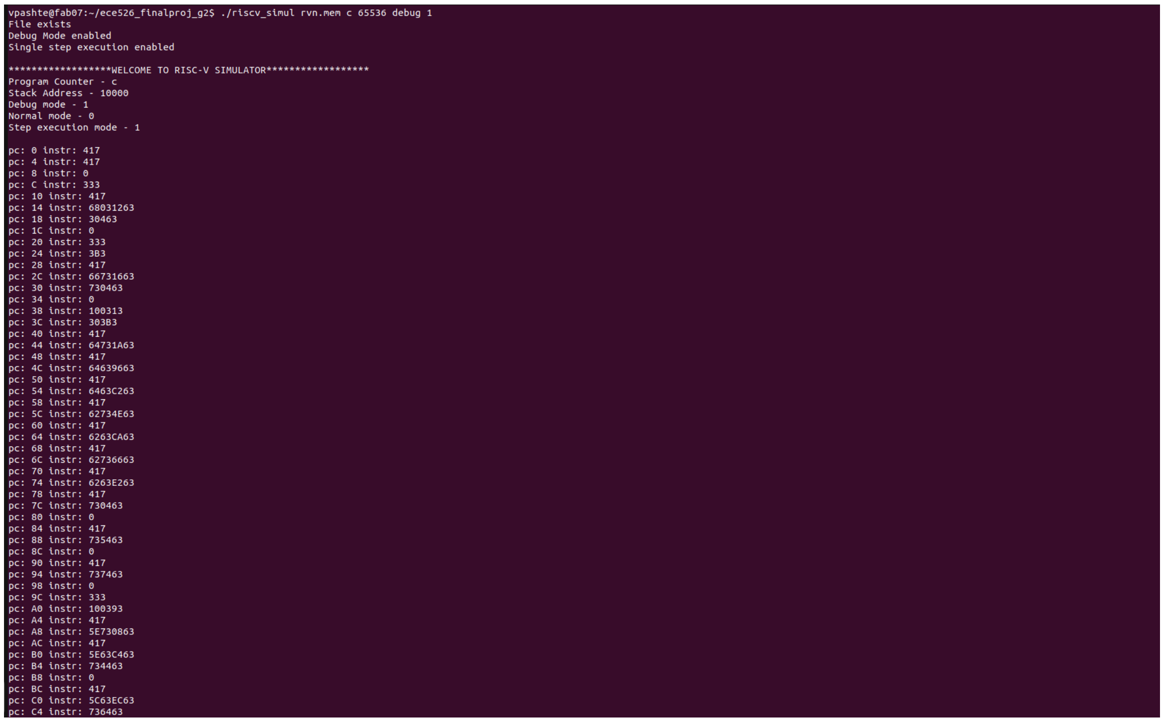


```
Opcode: 67 funct3: 0 funct7: 0
rd: 12 rs1: 13 rs2: 10
I-type Instruction
I-immediate: 10
JALR detected

x[0] -        0         x[1] -        0         x[2] - 0000FFFC         x[3] -        0         x[4] -        0         x[5] -        0         x[6] - 00000001         x[7] - FFFFFFFF
x[8] - 00000684         x[9] - 0000FFE8         x[10] -        0        x[11] -        0        x[12] -        0        x[13] -        0        x[14] -        0        x[15] -        0
x[16] -        0        x[17] -        0        x[18] - 00000694        x[19] - 0000068C        x[20] -        0        x[21] -        0        x[22] -        0        x[23] -        0
x[24] -        0        x[25] -        0        x[26] -        0        x[27] -        0        x[28] -        0        x[29] - FF00F80F        x[30] -        0        x[31] -        0

pc: 69C instr: 313
Opcode: 13 funct3: 0 funct7: 0
rd: 6 rs1: 0 rs2: 0
I-type Instruction
I-immediate: 0
ADDI detected

x[0] -        0         x[1] -        0         x[2] - 0000FFFC         x[3] -        0         x[4] -        0         x[5] -        0         x[6] -        0         x[7] - FFFFFFFF
x[8] - 00000684         x[9] - 0000FFE8         x[10] -        0        x[11] -        0        x[12] -        0        x[13] -        0        x[14] -        0        x[15] -        0
x[16] -        0        x[17] -        0        x[18] - 00000694        x[19] - 0000068C        x[20] -        0        x[21] -        0        x[22] -        0        x[23] -        0
x[24] -        0        x[25] -        0        x[26] -        0        x[27] -        0        x[28] -        0        x[29] - FF00F80F        x[30] -        0        x[31] -        0

pc: 6A0 instr: 90067
Opcode: 67 funct3: 0 funct7: 0
rd: 0 rs1: 12 rs2: 0
I-type Instruction
I-immediate: 0
JALR detected

x[0] - 000006A4         x[1] -        0         x[2] - 0000FFFC         x[3] -        0         x[4] -        0         x[5] -        0         x[6] -        0         x[7] - FFFFFFFF
x[8] - 00000684         x[9] - 0000FFE8         x[10] -        0        x[11] -        0        x[12] -        0        x[13] -        0        x[14] -        0        x[15] -        0
x[16] -        0        x[17] -        0        x[18] - 00000694        x[19] - 0000068C        x[20] -        0        x[21] -        0        x[22] -        0        x[23] -        0
x[24] -        0        x[25] -        0        x[26] -        0        x[27] -        0        x[28] -        0        x[29] - FF00F80F        x[30] -        0        x[31] -        0

pc: 694 instr: 31263
Opcode: 63 funct3: 1 funct7: 0
rd: 4 rs1: 6 rs2: 0
B-type Instruction
B-immediate: 4
BNE detected

x[0] -        0         x[1] -        0         x[2] - 0000FFFC         x[3] -        0         x[4] -        0         x[5] -        0         x[6] -        0         x[7] - FFFFFFFF
x[8] - 00000684         x[9] - 0000FFE8         x[10] -        0        x[11] -        0        x[12] -        0        x[13] -        0        x[14] -        0        x[15] -        0
x[16] -        0        x[17] -        0        x[18] - 00000694        x[19] - 0000068C        x[20] -        0        x[21] -        0        x[22] -        0        x[23] -        0
x[24] -        0        x[25] -        0        x[26] -        0        x[27] -        0        x[28] -        0        x[29] - FF00F80F        x[30] -        0        x[31] -        0

pc: 698 instr: 0
Ending simulation !!! (all 0 instr)
***********************************DONE***********************************

Last PC : 698

*****************************Registers*********************************

x[0] -        0         x[1] -        0         x[2] - 0000FFFC         x[3] -        0         x[4] -        0         x[5] -        0         x[6] -        0         x[7] - FFFFFFFF
x[8] - 00000684         x[9] - 0000FFE8         x[10] -        0        x[11] -        0        x[12] -        0        x[13] -        0        x[14] -        0        x[15] -        0
x[16] -        0        x[17] -        0        x[18] - 00000694        x[19] - 0000068C        x[20] -        0        x[21] -        0        x[22] -        0        x[23] -        0
x[24] -        0        x[25] -        0        x[26] -        0        x[27] -        0        x[28] -        0        x[29] - FF00F80F        x[30] -        0        x[31] -        0

************************************END************************************
vpashte@fab07:~/ece526_finalproj_g2$
```

4. Step execution



```
pc: 678 instr: 417
pc: 67C instr: 200096F
pc: 680 instr: 31C63
pc: 684 instr: 417
pc: 688 instr: 100313
pc: 68C instr: 997
pc: 690 instr: 1098967
pc: 694 instr: 31263
pc: 698 instr: 0
pc: 69C instr: 313
pc: 6A0 instr: 90067
pc: 6A4 instr: 0
pc: 6A8 instr: 417
pc: 6AC instr: 313
pc: 6B0 instr: 100013
pc: 6B4 instr: FE0312E3
pc: 6B8 instr: 417
pc: 6BC instr: 997
pc: 6C0 instr: 298967
pc: 6CC instr: 2FF6F3B
pc: 6D8 instr: 73
pc: 6DC instr: 100073
pc: 6E0 instr: 0
End of file reading and program saved to memory
pc: C instr: 333
Opcode: 33 funct3: 0 funct7: 0
rd: 6 rs1: 0 rs2: 0
R-type Instruction
ADD detected

x[0] -        0          x[1] -        0          x[2] - 00010000          x[3] -        0          x[4] -        0          x[5] -        0          x[6] -        0          x[7] -        0
x[8] -        0          x[9] -        0          x[10] -        0         x[11] -        0         x[12] -        0         x[13] -        0         x[14] -        0         x[15] -        0
x[16] -        0         x[17] -        0         x[18] -        0         x[19] -        0         x[20] -        0         x[21] -        0         x[22] -        0         x[23] -        0
x[24] -        0         x[25] -        0         x[26] -        0         x[27] -        0         x[28] -        0         x[29] -        0         x[30] -        0         x[31] -        0


pc: 10 instr: 417
Opcode: 17 funct3: 0 funct7: 0
rd: 8 rs1: 0 rs2: 0
U-type Instruction
U-immediate: 0
AUIPC detected

x[0] -        0          x[1] -        0          x[2] - 00010000          x[3] -        0          x[4] -        0          x[5] -        0          x[6] -        0          x[7] -        0
x[8] - 00000010          x[9] -        0          x[10] -        0         x[11] -        0         x[12] -        0         x[13] -        0         x[14] -        0         x[15] -        0
x[16] -        0         x[17] -        0         x[18] -        0         x[19] -        0         x[20] -        0         x[21] -        0         x[22] -        0         x[23] -        0
x[24] -        0         x[25] -        0         x[26] -        0         x[27] -        0         x[28] -        0         x[29] -        0         x[30] -        0         x[31] -        0


pc: 14 instr: 68031263
Opcode: 63 funct3: 1 funct7: 34
rd: 4 rs1: 6 rs2: 0
B-type Instruction
B-immediate: 684
BNE detected

x[0] -        0          x[1] -        0          x[2] - 00010000          x[3] -        0          x[4] -        0          x[5] -        0          x[6] -        0          x[7] -        0
x[8] - 00000010          x[9] -        0          x[10] -        0         x[11] -        0         x[12] -        0         x[13] -        0         x[14] -        0         x[15] -        0
x[16] -        0         x[17] -        0         x[18] -        0         x[19] -        0         x[20] -        0         x[21] -        0         x[22] -        0         x[23] -        0
x[24] -        0         x[25] -        0         x[26] -        0         x[27] -        0         x[28] -        0         x[29] -        0         x[30] -        0         x[31] -        0

█
```



```
vpashte@fab07:~/ece526_finalproj_g2$ ./riscv_simul rvn.mem c 65536 debug 1
File exists
Debug Mode enabled
Single step execution enabled

*****************WELCOME TO RISC-V SIMULATOR******************
Program Counter - c
Stack Address - 10000
Debug mode - 1
Normal mode - 0
Step execution mode - 1

pc: 0 instr: 417
pc: 4 instr: 417
pc: 8 instr: 0
pc: C instr: 333
pc: 10 instr: 417
pc: 14 instr: 68031263
pc: 18 instr: 30463
pc: 1C instr: 0
pc: 20 instr: 333
pc: 24 instr: 3B3
pc: 28 instr: 417
pc: 2C instr: 66731663
pc: 30 instr: 730463
pc: 34 instr: 0
pc: 38 instr: 100313
pc: 3C instr: 303B3
pc: 40 instr: 417
pc: 44 instr: 64731A63
pc: 48 instr: 417
pc: 4C instr: 64639663
pc: 50 instr: 417
pc: 54 instr: 6463C263
pc: 58 instr: 417
pc: 5C instr: 62734E63
pc: 60 instr: 417
pc: 64 instr: 6263CA63
pc: 68 instr: 417
pc: 6C instr: 62736663
pc: 70 instr: 417
pc: 74 instr: 6263E263
pc: 78 instr: 417
pc: 7C instr: 730463
pc: 80 instr: 0
pc: 84 instr: 417
pc: 88 instr: 735463
pc: 8C instr: 0
pc: 90 instr: 417
pc: 94 instr: 737463
pc: 98 instr: 0
pc: 9C instr: 333
pc: A0 instr: 100393
pc: A4 instr: 417
pc: A8 instr: 5E730863
pc: AC instr: 417
pc: B0 instr: 5E63C463
pc: B4 instr: 734463
pc: B8 instr: 0
pc: BC instr: 417
pc: C0 instr: 5C63EC63
pc: C4 instr: 736463
```

Portland State
UNIVERSITY

END