CSE 331 HW 4 REPORT

EMRE SEZER
1901042640

Inner Modules of mini_mips:

mini_mips_instruction_memory : includes registers with instructions. These instructions are read from the file named “instruction_memory.mem”. Returns the 16-bit instruction which program counter points.

instruction_parser: Parses the 16 bit instruction to it’s properties (opcode, rs, rt, rd, func, imm).

control_unit: Produces control inputs (reg_des, alu_src, mem_to_reg, reg_write, mem_read, mem_write, branch, alu_op) for inner components of the mini MIPS processor.

alu_control: Takes alu_op and func inputs and returns alu_ctr output which tells ALU to execute which operation.

sign_extender: Takes imm which is produced at instruction_parser and produces it’s 32 bit sign-extended version.

alu32: This module is simply modified version of alu32 from HW3. Executes operations depending on input alu_ctr.

mini_mips_memory: Includes 2D registers named data_memory. These registers are loaded with data from file named “data_memory.mem”. If mem_read input is 1, read_data is equal to memory content which address points at the data_memory. If mem_write is 1, memory content of the specific register’s is set to write_data. This specific register is decided due to address. In the end, content of data_memory is written to “data_memory.mem”.

mini_mips_registers: Includes 2D registers named registers. These registers are loaded with data from file named “registers.mem”.Content of read_data_1 is set to value of read_address_1’th register. Content of read_data_2 is set to value of read_address_2’th register. Content of write_address’th register is set to value of write_data. Zero register is always set to 0. In the end, content of registers is written to “registers.mem”.

program_counter_handler: It simply handles PC. If branch input which is produced at the control_unit and 0’th bit of alu_result is 1 returns sign_extended imm which is produced at the sign_extender. Else returns 1. This part is for computing PC correctly depending on branch. If there is no branch instruction we need to add 1 to the PC. But, if there is a branch instruction is executed we can’t simply add 1 to the PC. We need to add the sign extended version of the imm field at the instruction to the PC.

mux16x1_32: 16x1 mux for 32 bit numbers.

equality_checker: Checks whether 2 32 bit numbers are equal to each other. If they are returns 1. Else, returns 0.

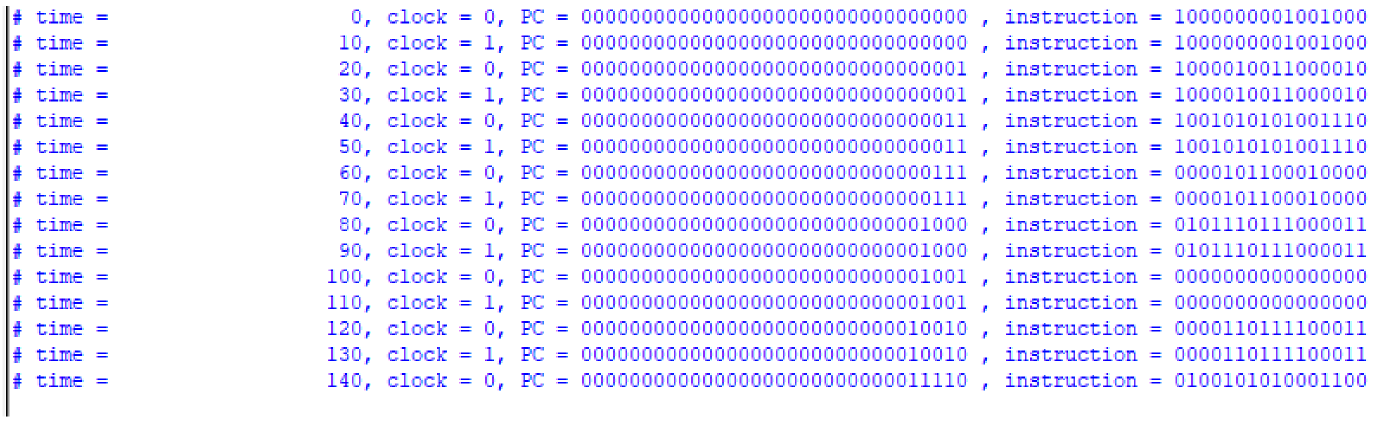
I used “registers.mem” file for storing register contents, “data_memory.mem” file for storingmemory contents and “instruction_memory.mem” file for storing instructions.

$0 register is always equal to 0. It can’t be changed. I calculated expressions for control inputs and coded on Verilog according to them

Inside alu_32 folder there are modules of the ALU. I updated the ALU from the HW3 and used it. Inside the testbenches folder, there are testbench modules.

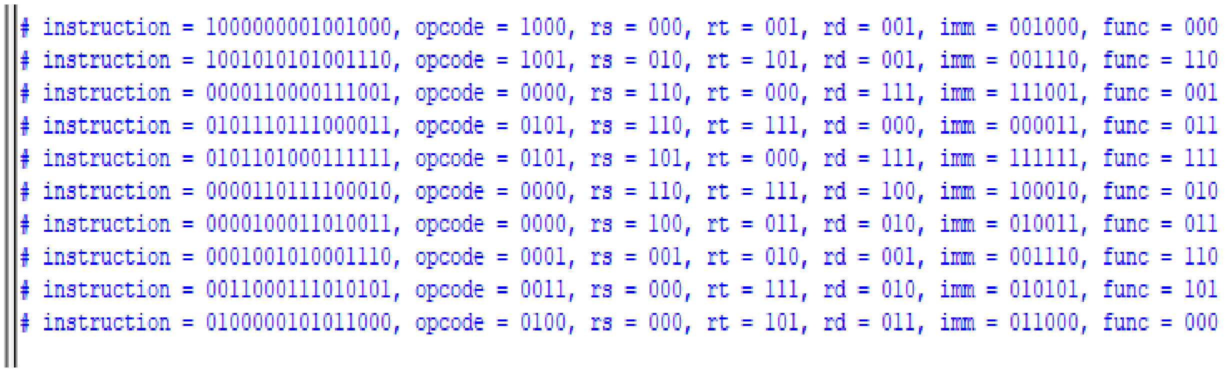
Screenshots:

Instruction Memory:



```
# time =                    0, clock = 0, PC = 00000000000000000000000000000000 , instruction = 1000000001001000
# time =                   10, clock = 1, PC = 00000000000000000000000000000000 , instruction = 1000000001001000
# time =                   20, clock = 0, PC = 00000000000000000000000000000001 , instruction = 1000010011000010
# time =                   30, clock = 1, PC = 00000000000000000000000000000001 , instruction = 1000010011000010
# time =                   40, clock = 0, PC = 00000000000000000000000000000011 , instruction = 1001010101001110
# time =                   50, clock = 1, PC = 00000000000000000000000000000011 , instruction = 1001010101001110
# time =                   60, clock = 0, PC = 00000000000000000000000000000111 , instruction = 0000101100010000
# time =                   70, clock = 1, PC = 00000000000000000000000000000111 , instruction = 0000101100010000
# time =                   80, clock = 0, PC = 00000000000000000000000000001000 , instruction = 0101110111000011
# time =                   90, clock = 1, PC = 00000000000000000000000000001000 , instruction = 0101110111000011
# time =                  100, clock = 0, PC = 00000000000000000000000000001001 , instruction = 0000000000000000
# time =                  110, clock = 1, PC = 00000000000000000000000000001001 , instruction = 0000000000000000
# time =                  120, clock = 0, PC = 00000000000000000000000000010010 , instruction = 0000110111100011
# time =                  130, clock = 1, PC = 00000000000000000000000000010010 , instruction = 0000110111100011
# time =                  140, clock = 0, PC = 00000000000000000000000000011110 , instruction = 0100101010001100
```

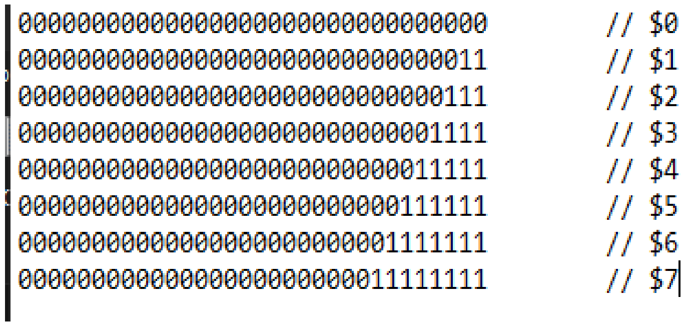
Instruction Parser:



```
# instruction = 1000000001001000, opcode = 1000, rs = 000, rt = 001, rd = 001, imm = 001000, func = 000
# instruction = 1001010101001110, opcode = 1001, rs = 010, rt = 101, rd = 001, imm = 001110, func = 110
# instruction = 0000110000111001, opcode = 0000, rs = 110, rt = 000, rd = 111, imm = 111001, func = 001
# instruction = 0101110111000011, opcode = 0101, rs = 110, rt = 111, rd = 000, imm = 000011, func = 011
# instruction = 0101101000111111, opcode = 0101, rs = 101, rt = 000, rd = 111, imm = 111111, func = 111
# instruction = 0000110111100010, opcode = 0000, rs = 110, rt = 111, rd = 100, imm = 100010, func = 010
# instruction = 0000100011010011, opcode = 0000, rs = 100, rt = 011, rd = 010, imm = 010011, func = 011
# instruction = 0001001010001110, opcode = 0001, rs = 001, rt = 010, rd = 001, imm = 001110, func = 110
# instruction = 0011000111010101, opcode = 0011, rs = 000, rt = 111, rd = 010, imm = 010101, func = 101
# instruction = 0100000101011000, opcode = 0100, rs = 000, rt = 101, rd = 011, imm = 011000, func = 000
```

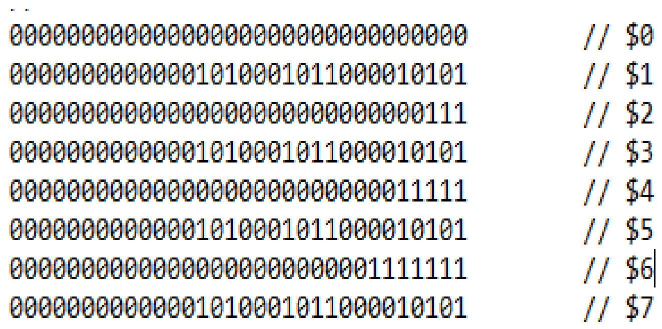
Registers:

Before running "registers.mem" file:

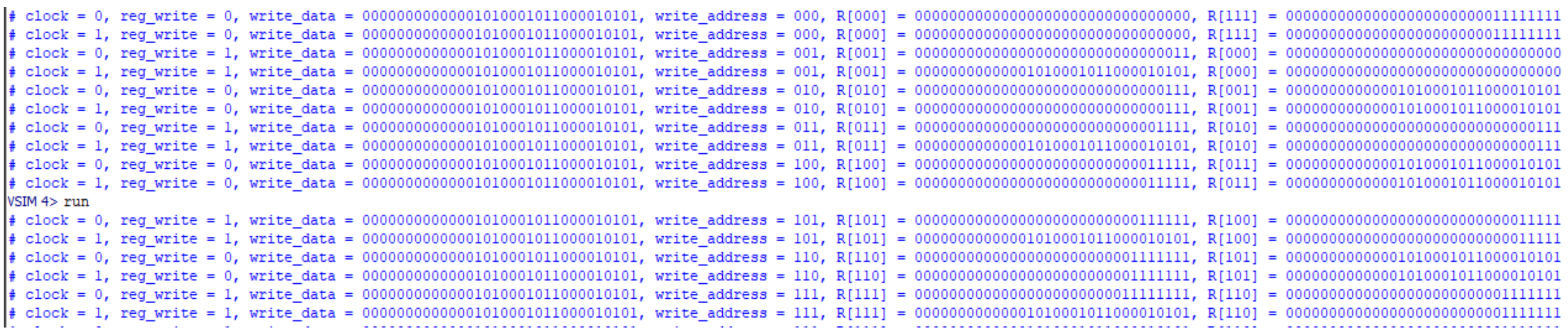


```
00000000000000000000000000000000        // $0
00000000000000000000000000000011        // $1
00000000000000000000000000000111        // $2
00000000000000000000000000001111        // $3
00000000000000000000000000011111        // $4
00000000000000000000000000111111        // $5
00000000000000000000000001111111        // $6
00000000000000000000000011111111        // $7
```

After running "registers.mem" file:

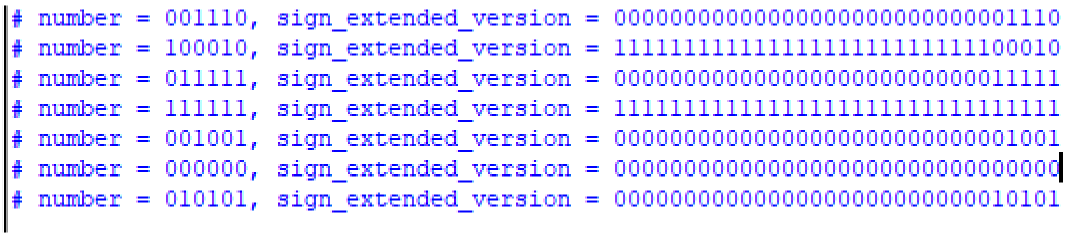


```
00000000000000000000000000000000        // $0
00000000000001010001011000010101        // $1
00000000000000000000000000000111        // $2
00000000000001010001011000010101        // $3
00000000000000000000000000011111        // $4
00000000000001010001011000010101        // $5
00000000000000000000000001111111        // $6
00000000000001010001011000010101        // $7
```



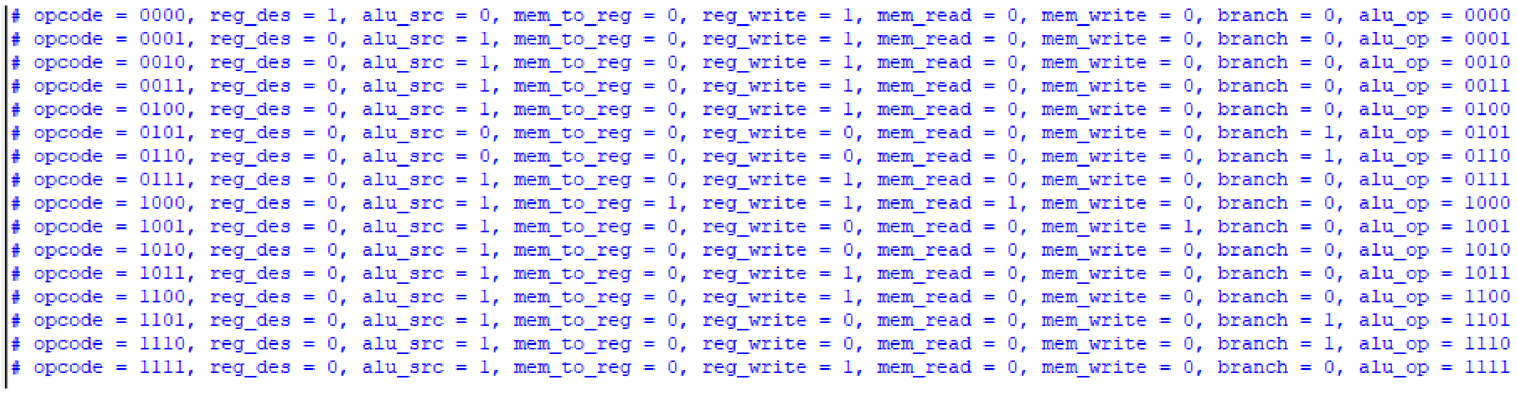
```
# clock = 0, reg_write = 0, write_data = 00000000000001010001011000010101, write_address = 000, R[000] = 00000000000000000000000000000000, R[111] = 00000000000000000000000011111111
# clock = 1, reg_write = 0, write_data = 00000000000001010001011000010101, write_address = 000, R[000] = 00000000000000000000000000000000, R[111] = 00000000000000000000000011111111
# clock = 0, reg_write = 1, write_data = 00000000000001010001011000010101, write_address = 001, R[001] = 00000000000000000000000000000011, R[000] = 00000000000000000000000000000000
# clock = 1, reg_write = 1, write_data = 00000000000001010001011000010101, write_address = 001, R[001] = 00000000000001010001011000010101, R[000] = 00000000000000000000000000000000
# clock = 0, reg_write = 0, write_data = 00000000000001010001011000010101, write_address = 010, R[010] = 00000000000000000000000000000111, R[001] = 00000000000001010001011000010101
# clock = 1, reg_write = 0, write_data = 00000000000001010001011000010101, write_address = 010, R[010] = 00000000000000000000000000000111, R[001] = 00000000000001010001011000010101
# clock = 0, reg_write = 1, write_data = 00000000000001010001011000010101, write_address = 011, R[011] = 00000000000000000000000000001111, R[010] = 00000000000000000000000000000111
# clock = 1, reg_write = 1, write_data = 00000000000001010001011000010101, write_address = 011, R[011] = 00000000000001010001011000010101, R[010] = 00000000000000000000000000000111
# clock = 0, reg_write = 0, write_data = 00000000000001010001011000010101, write_address = 100, R[100] = 00000000000000000000000000011111, R[011] = 00000000000001010001011000010101
# clock = 1, reg_write = 0, write_data = 00000000000001010001011000010101, write_address = 100, R[100] = 00000000000000000000000000011111, R[011] = 00000000000001010001011000010101
VSIM 4> run
# clock = 0, reg_write = 1, write_data = 00000000000001010001011000010101, write_address = 101, R[101] = 00000000000000000000000000111111, R[100] = 00000000000000000000000000011111
# clock = 1, reg_write = 1, write_data = 00000000000001010001011000010101, write_address = 101, R[101] = 00000000000001010001011000010101, R[100] = 00000000000000000000000000011111
# clock = 0, reg_write = 0, write_data = 00000000000001010001011000010101, write_address = 110, R[110] = 00000000000000000000000001111111, R[101] = 00000000000001010001011000010101
# clock = 1, reg_write = 0, write_data = 00000000000001010001011000010101, write_address = 110, R[110] = 00000000000000000000000001111111, R[101] = 00000000000001010001011000010101
# clock = 0, reg_write = 1, write_data = 00000000000001010001011000010101, write_address = 111, R[111] = 00000000000000000000000011111111, R[110] = 00000000000000000000000001111111
# clock = 1, reg_write = 1, write_data = 00000000000001010001011000010101, write_address = 111, R[111] = 00000000000001010001011000010101, R[110] = 00000000000000000000000001111111
```

Sign Extender:



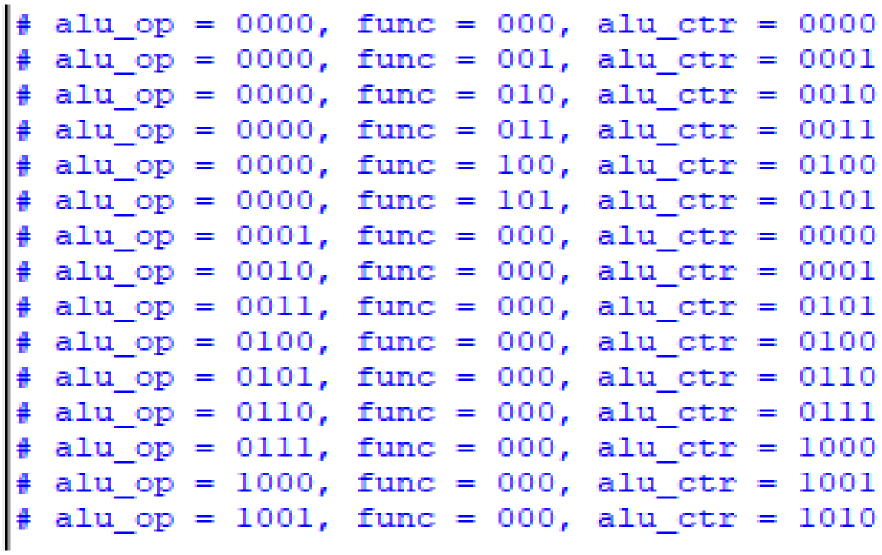
```
# number = 001110, sign_extended_version = 00000000000000000000000000001110
# number = 100010, sign_extended_version = 11111111111111111111111111100010
# number = 011111, sign_extended_version = 00000000000000000000000000011111
# number = 111111, sign_extended_version = 11111111111111111111111111111111
# number = 001001, sign_extended_version = 00000000000000000000000000001001
# number = 000000, sign_extended_version = 00000000000000000000000000000000
# number = 010101, sign_extended_version = 00000000000000000000000000010101
```

Control Unit:



```
# opcode = 0000, reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000
# opcode = 0001, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0001
# opcode = 0010, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0010
# opcode = 0011, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0011
# opcode = 0100, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0100
# opcode = 0101, reg_des = 0, alu_src = 0, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 0101
# opcode = 0110, reg_des = 0, alu_src = 0, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 0110
# opcode = 0111, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0111
# opcode = 1000, reg_des = 0, alu_src = 1, mem_to_reg = 1, reg_write = 1, mem_read = 1, mem_write = 0, branch = 0, alu_op = 1000
# opcode = 1001, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 1, branch = 0, alu_op = 1001
# opcode = 1010, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 0, alu_op = 1010
# opcode = 1011, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 1011
# opcode = 1100, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 1100
# opcode = 1101, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 1101
# opcode = 1110, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 1110
# opcode = 1111, reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 1111
```

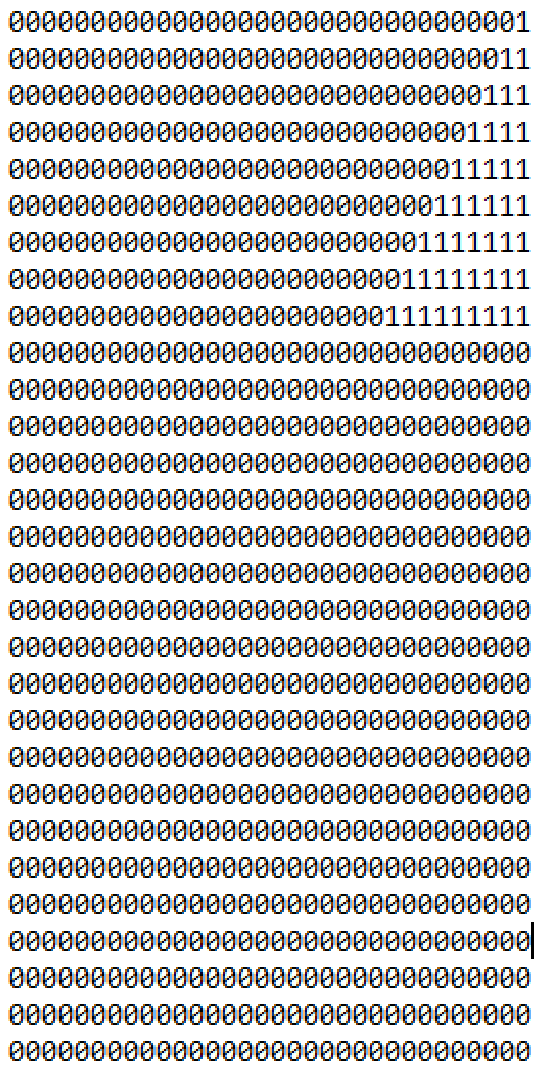
ALU Control:



```
# alu_op = 0000, func = 000, alu_ctr = 0000
# alu_op = 0000, func = 001, alu_ctr = 0001
# alu_op = 0000, func = 010, alu_ctr = 0010
# alu_op = 0000, func = 011, alu_ctr = 0011
# alu_op = 0000, func = 100, alu_ctr = 0100
# alu_op = 0000, func = 101, alu_ctr = 0101
# alu_op = 0001, func = 000, alu_ctr = 0000
# alu_op = 0010, func = 000, alu_ctr = 0001
# alu_op = 0011, func = 000, alu_ctr = 0101
# alu_op = 0100, func = 000, alu_ctr = 0100
# alu_op = 0101, func = 000, alu_ctr = 0110
# alu_op = 0110, func = 000, alu_ctr = 0111
# alu_op = 0111, func = 000, alu_ctr = 1000
# alu_op = 1000, func = 000, alu_ctr = 1001
# alu_op = 1001, func = 000, alu_ctr = 1010
```

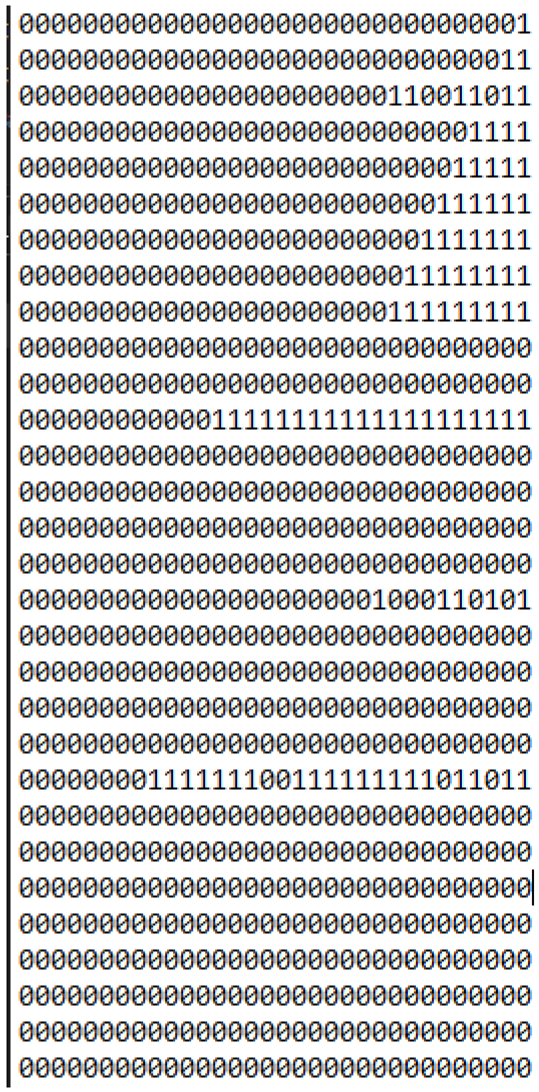
Memory:

Before running “data_memory.mem” file:

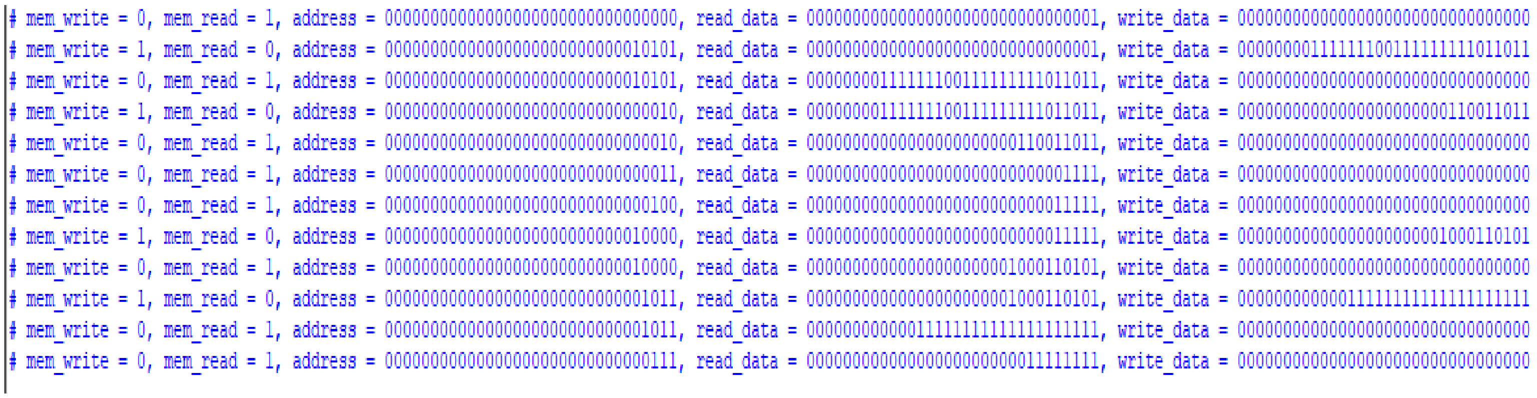


```
00000000000000000000000000000001
00000000000000000000000000000011
00000000000000000000000000000111
00000000000000000000000000001111
00000000000000000000000000011111
00000000000000000000000000111111
00000000000000000000000001111111
00000000000000000000000011111111
00000000000000000000000111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

After running “data_memory.mem” file:

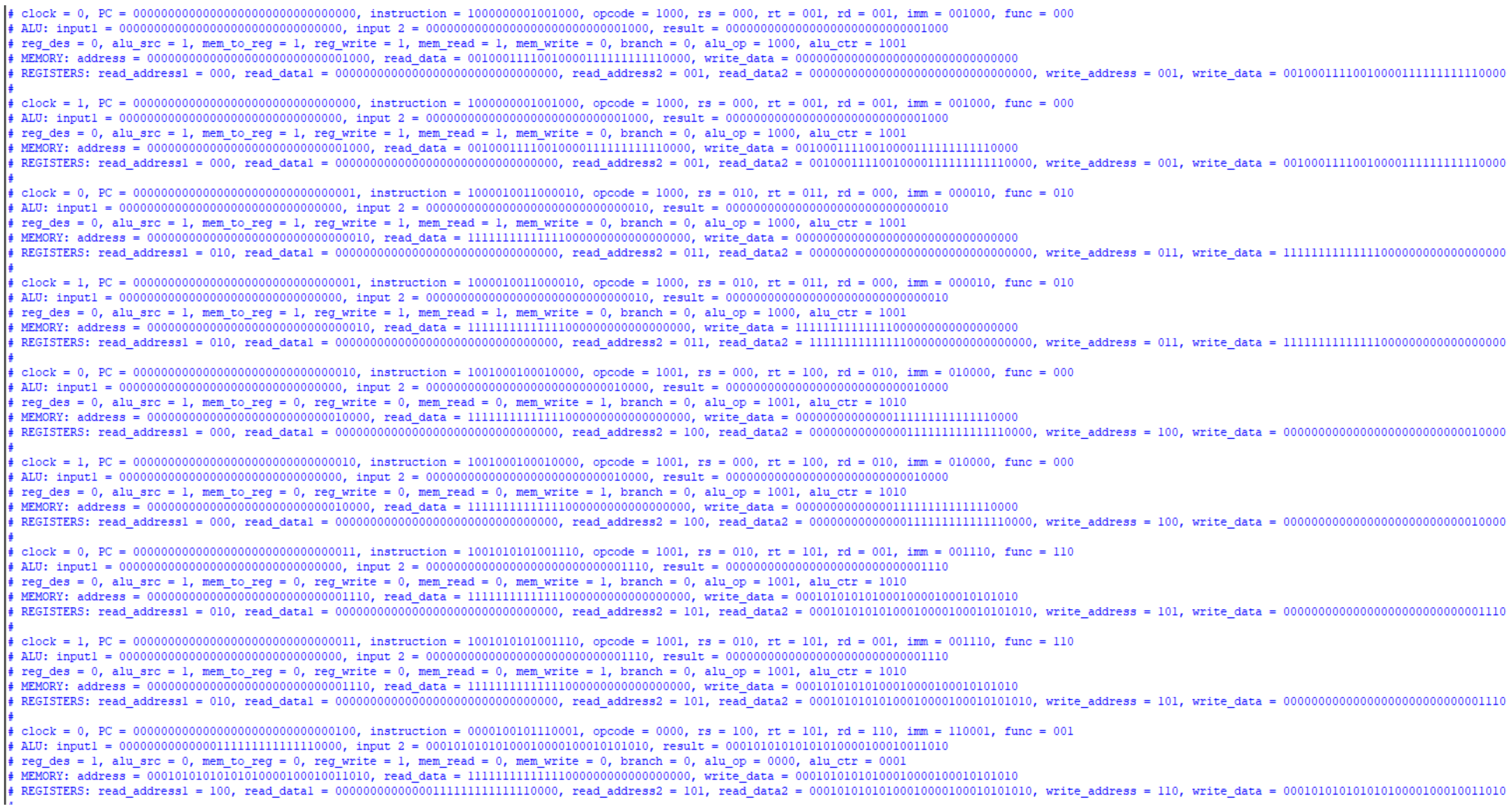


```
00000000000000000000000000000001
00000000000000000000000000000011
00000000000000000000000110011011
00000000000000000000000000001111
00000000000000000000000000011111
00000000000000000000000000111111
00000000000000000000000001111111
00000000000000000000000011111111
00000000000000000000000111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000011111111111111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000001000110101
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000111111100111111111011011
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

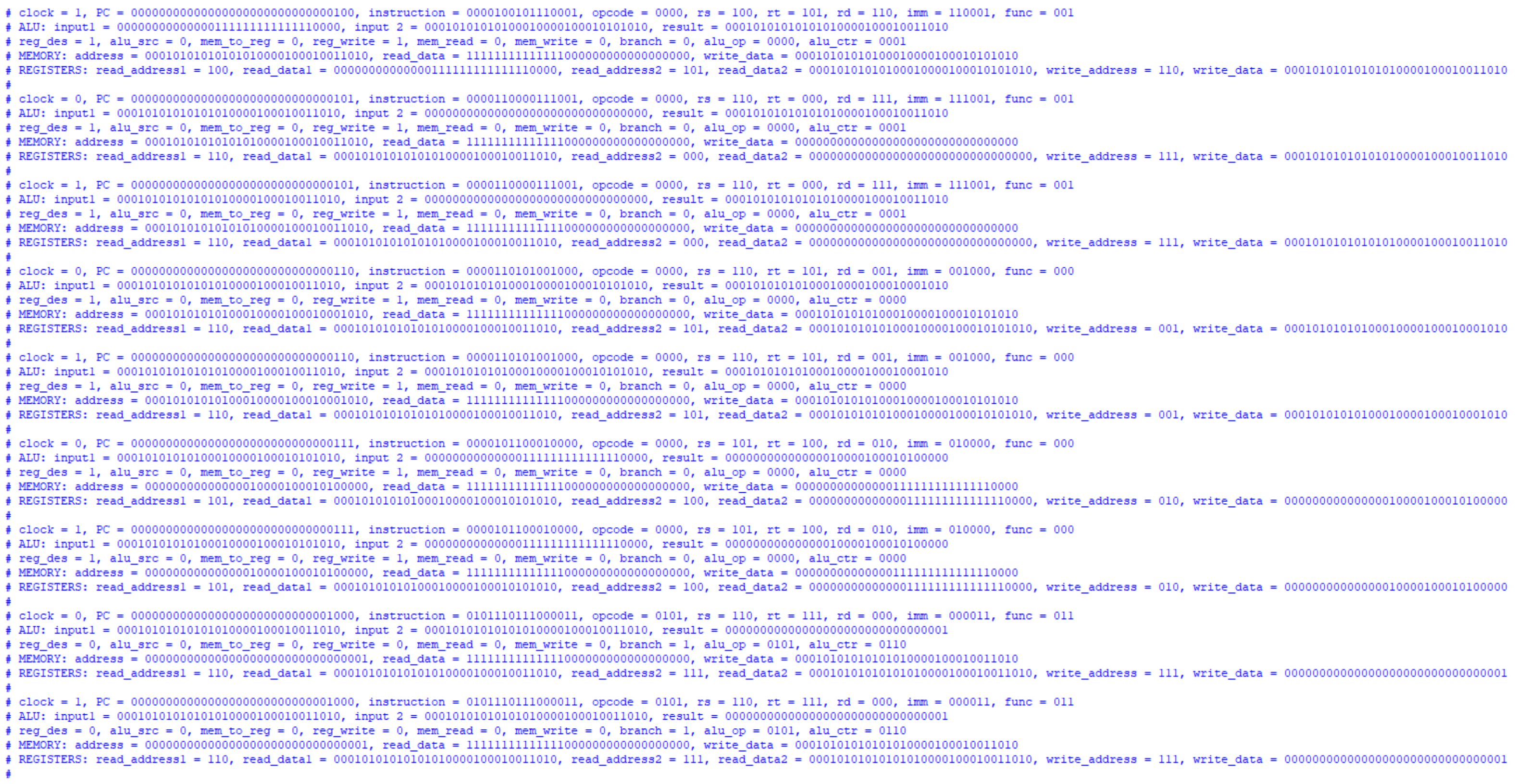


```
# mem_write = 0, mem_read = 1, address = 00000000000000000000000000000000, read_data = 00000000000000000000000000000001, write_data = 00000000000000000000000000000000
# mem_write = 1, mem_read = 0, address = 00000000000000000000000000010101, read_data = 00000000000000000000000000000001, write_data = 00000000111111100111111111011011
# mem_write = 0, mem_read = 1, address = 00000000000000000000000000010101, read_data = 00000000111111100111111111011011, write_data = 00000000000000000000000000000000
# mem_write = 1, mem_read = 0, address = 00000000000000000000000000000010, read_data = 00000000111111100111111111011011, write_data = 00000000000000000000000110011011
# mem_write = 0, mem_read = 1, address = 00000000000000000000000000000010, read_data = 00000000000000000000000110011011, write_data = 00000000000000000000000000000000
# mem_write = 0, mem_read = 1, address = 00000000000000000000000000000011, read_data = 00000000000000000000000000001111, write_data = 00000000000000000000000000000000
# mem_write = 0, mem_read = 1, address = 00000000000000000000000000000100, read_data = 00000000000000000000000000011111, write_data = 00000000000000000000000000000000
# mem_write = 1, mem_read = 0, address = 00000000000000000000000000010000, read_data = 00000000000000000000000000011111, write_data = 00000000000000000000001000110101
# mem_write = 0, mem_read = 1, address = 00000000000000000000000000010000, read_data = 00000000000000000000001000110101, write_data = 00000000000000000000000000000000
# mem_write = 1, mem_read = 0, address = 00000000000000000000000000001011, read_data = 00000000000000000000001000110101, write_data = 00000000000011111111111111111111
# mem_write = 0, mem_read = 1, address = 00000000000000000000000000001011, read_data = 00000000000011111111111111111111, write_data = 00000000000000000000000000000000
# mem_write = 0, mem_read = 1, address = 00000000000000000000000000000111, read_data = 00000000000000000000000011111111, write_data = 00000000000000000000000000000000
```

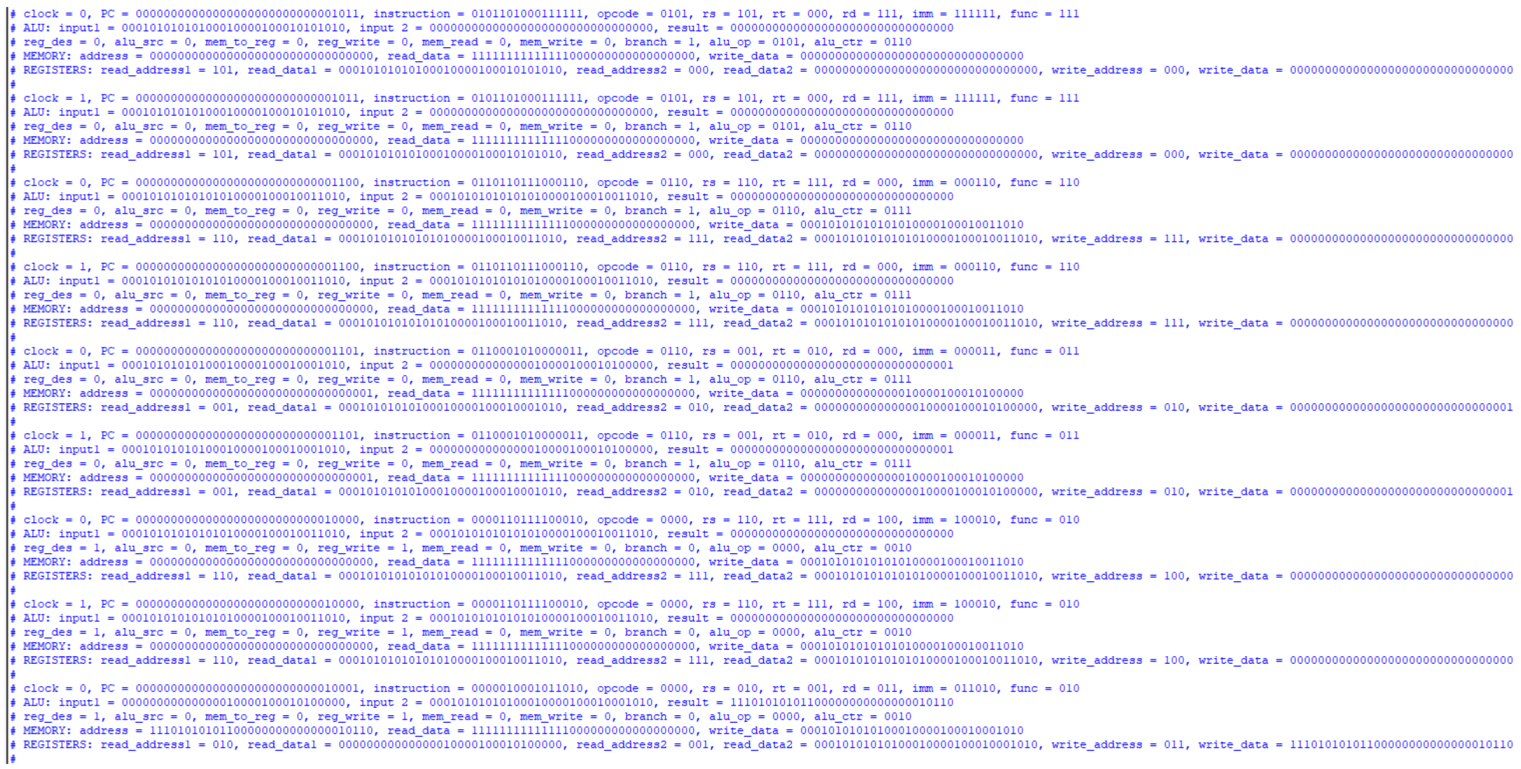
MINI MIPS:



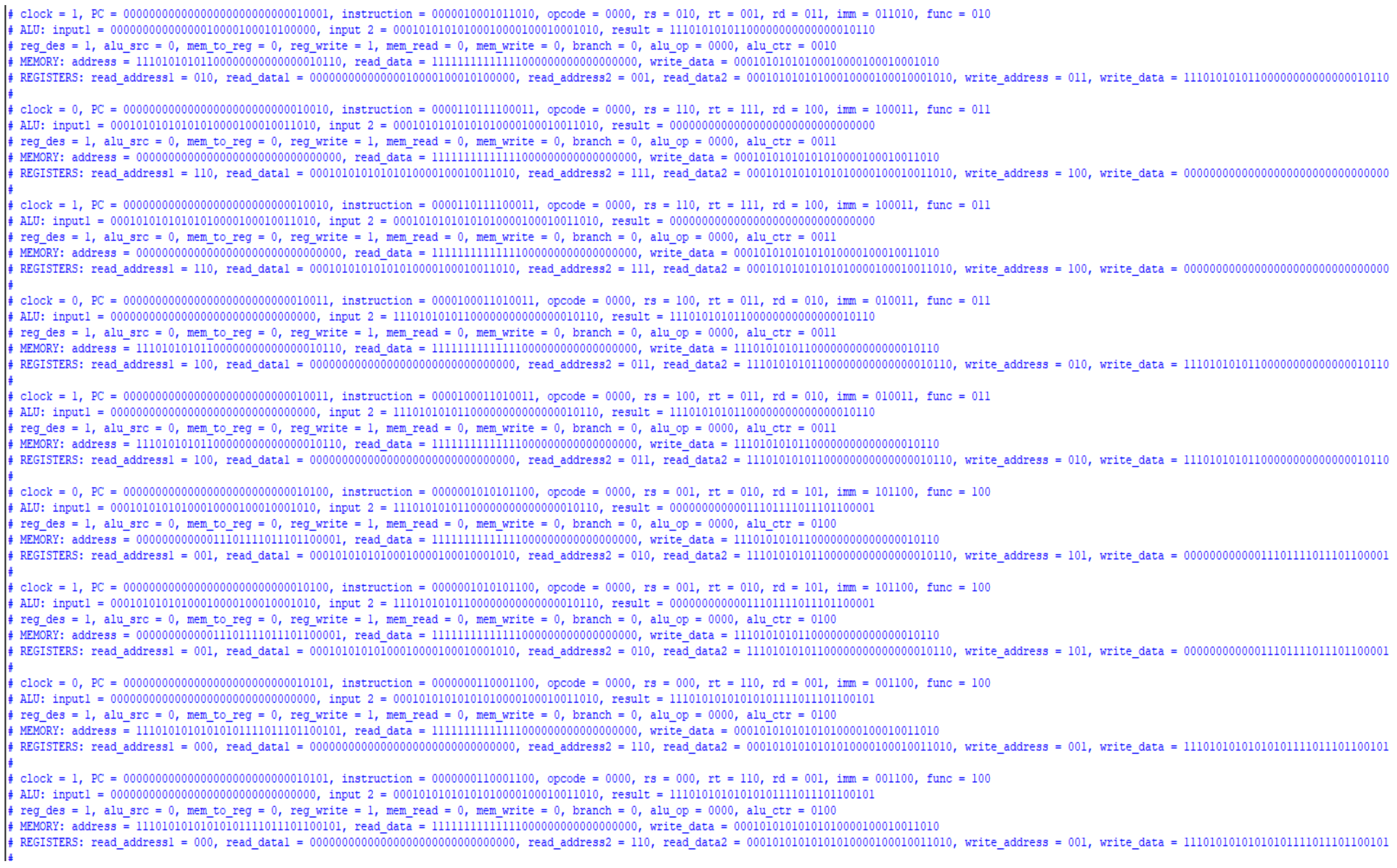
```
# clock = 0, PC = 00000000000000000000000000000000, instruction = 1000000001001000, opcode = 1000, rs = 000, rt = 001, rd = 001, imm = 001000, func = 000
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000001000, result = 00000000000000000000000000001000
# reg_des = 0, alu_src = 1, mem_to_reg = 1, reg_write = 1, mem_read = 1, mem_write = 0, branch = 0, alu_op = 1000, alu_ctr = 1001
# MEMORY: address = 00000000000000000000000000001000, read_data = 00100011110010000111111111110000, write_data = 00000000000000000000000000000000
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 001, read_data2 = 00000000000000000000000000000000, write_address = 001, write_data = 00100011110010000111111111110000
#
# clock = 1, PC = 00000000000000000000000000000000, instruction = 1000000001001000, opcode = 1000, rs = 000, rt = 001, rd = 001, imm = 001000, func = 000
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000001000, result = 00000000000000000000000000001000
# reg_des = 0, alu_src = 1, mem_to_reg = 1, reg_write = 1, mem_read = 1, mem_write = 0, branch = 0, alu_op = 1000, alu_ctr = 1001
# MEMORY: address = 00000000000000000000000000001000, read_data = 00100011110010000111111111110000, write_data = 00100011110010000111111111110000
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 001, read_data2 = 00100011110010000111111111110000, write_address = 001, write_data = 00100011110010000111111111110000
#
# clock = 0, PC = 00000000000000000000000000000001, instruction = 1000010011000010, opcode = 1000, rs = 010, rt = 011, rd = 000, imm = 000010, func = 010
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000000010, result = 00000000000000000000000000000010
# reg_des = 0, alu_src = 1, mem_to_reg = 1, reg_write = 1, mem_read = 1, mem_write = 0, branch = 0, alu_op = 1000, alu_ctr = 1001
# MEMORY: address = 00000000000000000000000000000010, read_data = 11111111111111000000000000000000, write_data = 00000000000000000000000000000000
# REGISTERS: read_address1 = 010, read_data1 = 00000000000000000000000000000000, read_address2 = 011, read_data2 = 00000000000000000000000000000000, write_address = 011, write_data = 11111111111111000000000000000000
#
# clock = 1, PC = 00000000000000000000000000000001, instruction = 1000010011000010, opcode = 1000, rs = 010, rt = 011, rd = 000, imm = 000010, func = 010
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000000010, result = 00000000000000000000000000000010
# reg_des = 0, alu_src = 1, mem_to_reg = 1, reg_write = 1, mem_read = 1, mem_write = 0, branch = 0, alu_op = 1000, alu_ctr = 1001
# MEMORY: address = 00000000000000000000000000000010, read_data = 11111111111111000000000000000000, write_data = 11111111111111000000000000000000
# REGISTERS: read_address1 = 010, read_data1 = 00000000000000000000000000000000, read_address2 = 011, read_data2 = 11111111111111000000000000000000, write_address = 011, write_data = 11111111111111000000000000000000
#
# clock = 0, PC = 00000000000000000000000000000010, instruction = 1001000100010000, opcode = 1001, rs = 000, rt = 100, rd = 010, imm = 010000, func = 000
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000010000, result = 00000000000000000000000000010000
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 1, branch = 0, alu_op = 1001, alu_ctr = 1010
# MEMORY: address = 00000000000000000000000000010000, read_data = 11111111111111000000000000000000, write_data = 00000000000000111111111111110000
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 100, read_data2 = 00000000000000111111111111110000, write_address = 100, write_data = 00000000000000000000000000010000
#
# clock = 1, PC = 00000000000000000000000000000010, instruction = 1001000100010000, opcode = 1001, rs = 000, rt = 100, rd = 010, imm = 010000, func = 000
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000010000, result = 00000000000000000000000000010000
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 1, branch = 0, alu_op = 1001, alu_ctr = 1010
# MEMORY: address = 00000000000000000000000000010000, read_data = 11111111111111000000000000000000, write_data = 00000000000000111111111111110000
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 100, read_data2 = 00000000000000111111111111110000, write_address = 100, write_data = 00000000000000000000000000010000
#
# clock = 0, PC = 00000000000000000000000000000011, instruction = 1001010101001110, opcode = 1001, rs = 010, rt = 101, rd = 001, imm = 001110, func = 110
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000001110, result = 00000000000000000000000000001110
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 1, branch = 0, alu_op = 1001, alu_ctr = 1010
# MEMORY: address = 00000000000000000000000000001110, read_data = 11111111111111000000000000000000, write_data = 00010101010100010000100010101010
# REGISTERS: read_address1 = 010, read_data1 = 00000000000000000000000000000000, read_address2 = 101, read_data2 = 00010101010100010000100010101010, write_address = 101, write_data = 00000000000000000000000000001110
#
# clock = 1, PC = 00000000000000000000000000000011, instruction = 1001010101001110, opcode = 1001, rs = 010, rt = 101, rd = 001, imm = 001110, func = 110
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000001110, result = 00000000000000000000000000001110
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 1, branch = 0, alu_op = 1001, alu_ctr = 1010
# MEMORY: address = 00000000000000000000000000001110, read_data = 11111111111111000000000000000000, write_data = 00010101010100010000100010101010
# REGISTERS: read_address1 = 010, read_data1 = 00000000000000000000000000000000, read_address2 = 101, read_data2 = 00010101010100010000100010101010, write_address = 101, write_data = 00000000000000000000000000001110
#
# clock = 0, PC = 00000000000000000000000000000100, instruction = 0000100101110001, opcode = 0000, rs = 100, rt = 101, rd = 110, imm = 110001, func = 001
# ALU: input1 = 00000000000000111111111111110000, input 2 = 00010101010100010000100010101010, result = 00010101010101010000100010011010
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0001
# MEMORY: address = 00010101010101010000100010011010, read_data = 11111111111111000000000000000000, write_data = 00010101010100010000100010101010
# REGISTERS: read_address1 = 100, read_data1 = 00000000000000111111111111110000, read_address2 = 101, read_data2 = 00010101010100010000100010101010, write_address = 110, write_data = 00010101010101010000100010011010
```



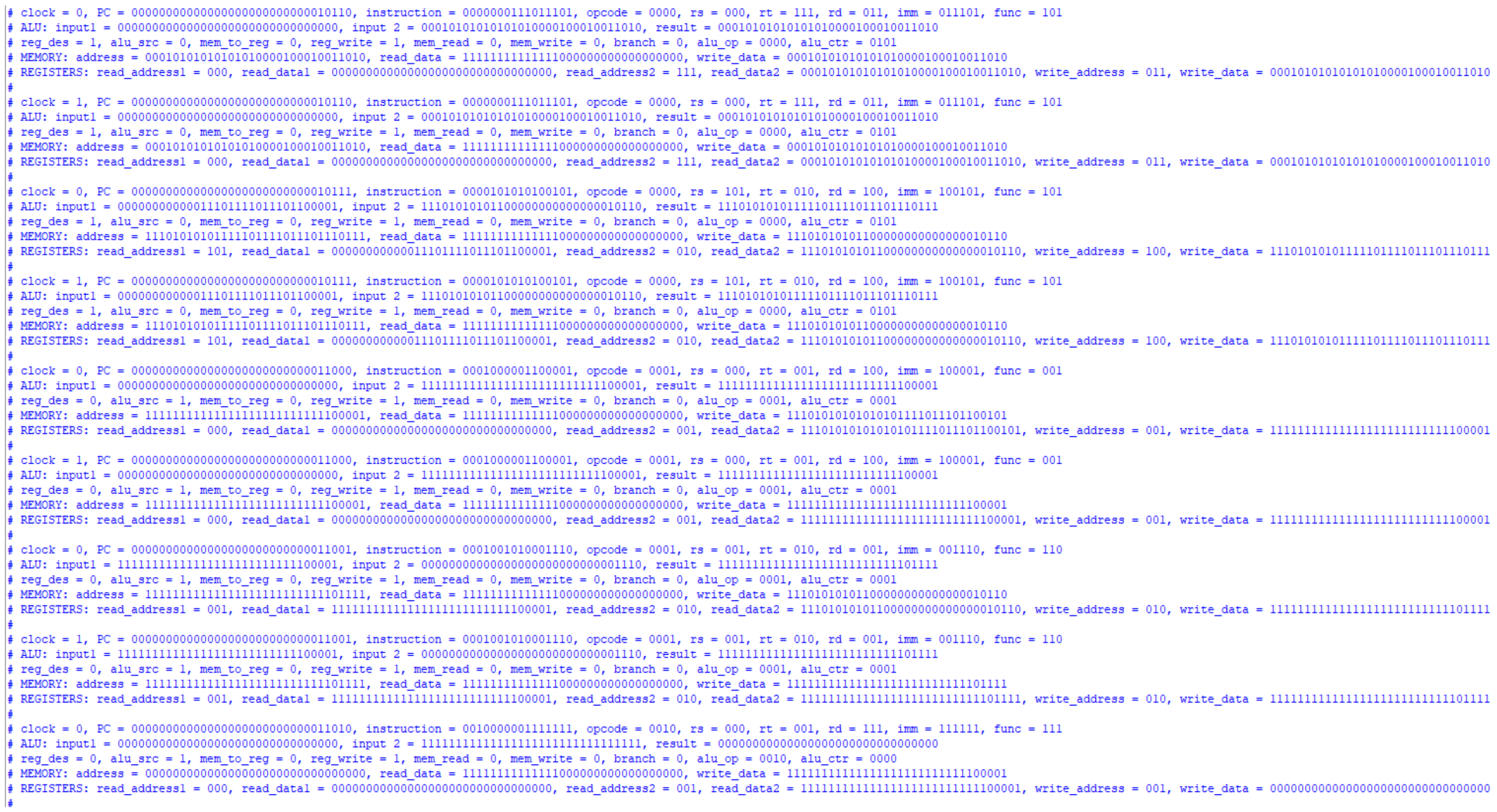
```
# clock = 1, PC = 00000000000000000000000000000100, instruction = 0000100101110001, opcode = 0000, rs = 100, rt = 101, rd = 110, imm = 110001, func = 001
# ALU: input1 = 00000000000000111111111111110000, input 2 = 00010101010100010000100010101010, result = 00010101010101010000100010011010
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0001
# MEMORY: address = 00010101010101010000100010011010, read_data = 11111111111111000000000000000000, write_data = 00010101010100010000100010101010
# REGISTERS: read_address1 = 100, read_data1 = 00000000000000111111111111110000, read_address2 = 101, read_data2 = 00010101010100010000100010101010, write_address = 110, write_data = 00010101010101010000100010011010
#
# clock = 0, PC = 00000000000000000000000000000101, instruction = 0000110000111001, opcode = 0000, rs = 110, rt = 000, rd = 111, imm = 111001, func = 001
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00000000000000000000000000000000, result = 00010101010101010000100010011010
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0001
# MEMORY: address = 00010101010101010000100010011010, read_data = 11111111111111000000000000000000, write_data = 00000000000000000000000000000000
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 000, read_data2 = 00000000000000000000000000000000, write_address = 111, write_data = 00010101010101010000100010011010
#
# clock = 1, PC = 00000000000000000000000000000101, instruction = 0000110000111001, opcode = 0000, rs = 110, rt = 000, rd = 111, imm = 111001, func = 001
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00000000000000000000000000000000, result = 00010101010101010000100010011010
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0001
# MEMORY: address = 00010101010101010000100010011010, read_data = 11111111111111000000000000000000, write_data = 00000000000000000000000000000000
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 000, read_data2 = 00000000000000000000000000000000, write_address = 111, write_data = 00010101010101010000100010011010
#
# clock = 0, PC = 00000000000000000000000000000110, instruction = 0000110101001000, opcode = 0000, rs = 110, rt = 101, rd = 001, imm = 001000, func = 000
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00010101010100010000100010101010, result = 00010101010100010000100010001010
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0000
# MEMORY: address = 00010101010100010000100010001010, read_data = 11111111111111000000000000000000, write_data = 00010101010100010000100010101010
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 101, read_data2 = 00010101010100010000100010101010, write_address = 001, write_data = 00010101010100010000100010001010
#
# clock = 1, PC = 00000000000000000000000000000110, instruction = 0000110101001000, opcode = 0000, rs = 110, rt = 101, rd = 001, imm = 001000, func = 000
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00010101010100010000100010101010, result = 00010101010100010000100010001010
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0000
# MEMORY: address = 00010101010100010000100010001010, read_data = 11111111111111000000000000000000, write_data = 00010101010100010000100010101010
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 101, read_data2 = 00010101010100010000100010101010, write_address = 001, write_data = 00010101010100010000100010001010
#
# clock = 0, PC = 00000000000000000000000000000111, instruction = 0000101100010000, opcode = 0000, rs = 101, rt = 100, rd = 010, imm = 010000, func = 000
# ALU: input1 = 00010101010100010000100010101010, input 2 = 00000000000000111111111111110000, result = 00000000000000010000100010100000
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0000
# MEMORY: address = 00000000000000010000100010100000, read_data = 11111111111111000000000000000000, write_data = 00000000000000111111111111110000
# REGISTERS: read_address1 = 101, read_data1 = 00010101010100010000100010101010, read_address2 = 100, read_data2 = 00000000000000111111111111110000, write_address = 010, write_data = 00000000000000010000100010100000
#
# clock = 1, PC = 00000000000000000000000000000111, instruction = 0000101100010000, opcode = 0000, rs = 101, rt = 100, rd = 010, imm = 010000, func = 000
# ALU: input1 = 00010101010100010000100010101010, input 2 = 00000000000000111111111111110000, result = 00000000000000010000100010100000
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0000
# MEMORY: address = 00000000000000010000100010100000, read_data = 11111111111111000000000000000000, write_data = 00000000000000111111111111110000
# REGISTERS: read_address1 = 101, read_data1 = 00010101010100010000100010101010, read_address2 = 100, read_data2 = 00000000000000111111111111110000, write_address = 010, write_data = 00000000000000010000100010100000
#
# clock = 0, PC = 00000000000000000000000000001000, instruction = 0101110111000011, opcode = 0101, rs = 110, rt = 111, rd = 000, imm = 000011, func = 011
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00010101010101010000100010011010, result = 00000000000000000000000000000001
# reg_des = 0, alu_src = 0, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 0101, alu_ctr = 0110
# MEMORY: address = 00000000000000000000000000000001, read_data = 11111111111111000000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 111, read_data2 = 00010101010101010000100010011010, write_address = 111, write_data = 00000000000000000000000000000001
#
# clock = 1, PC = 00000000000000000000000000001000, instruction = 0101110111000011, opcode = 0101, rs = 110, rt = 111, rd = 000, imm = 000011, func = 011
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00010101010101010000100010011010, result = 00000000000000000000000000000001
# reg_des = 0, alu_src = 0, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 0101, alu_ctr = 0110
# MEMORY: address = 00000000000000000000000000000001, read_data = 11111111111111000000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 111, read_data2 = 00010101010101010000100010011010, write_address = 111, write_data = 00000000000000000000000000000001
#
```



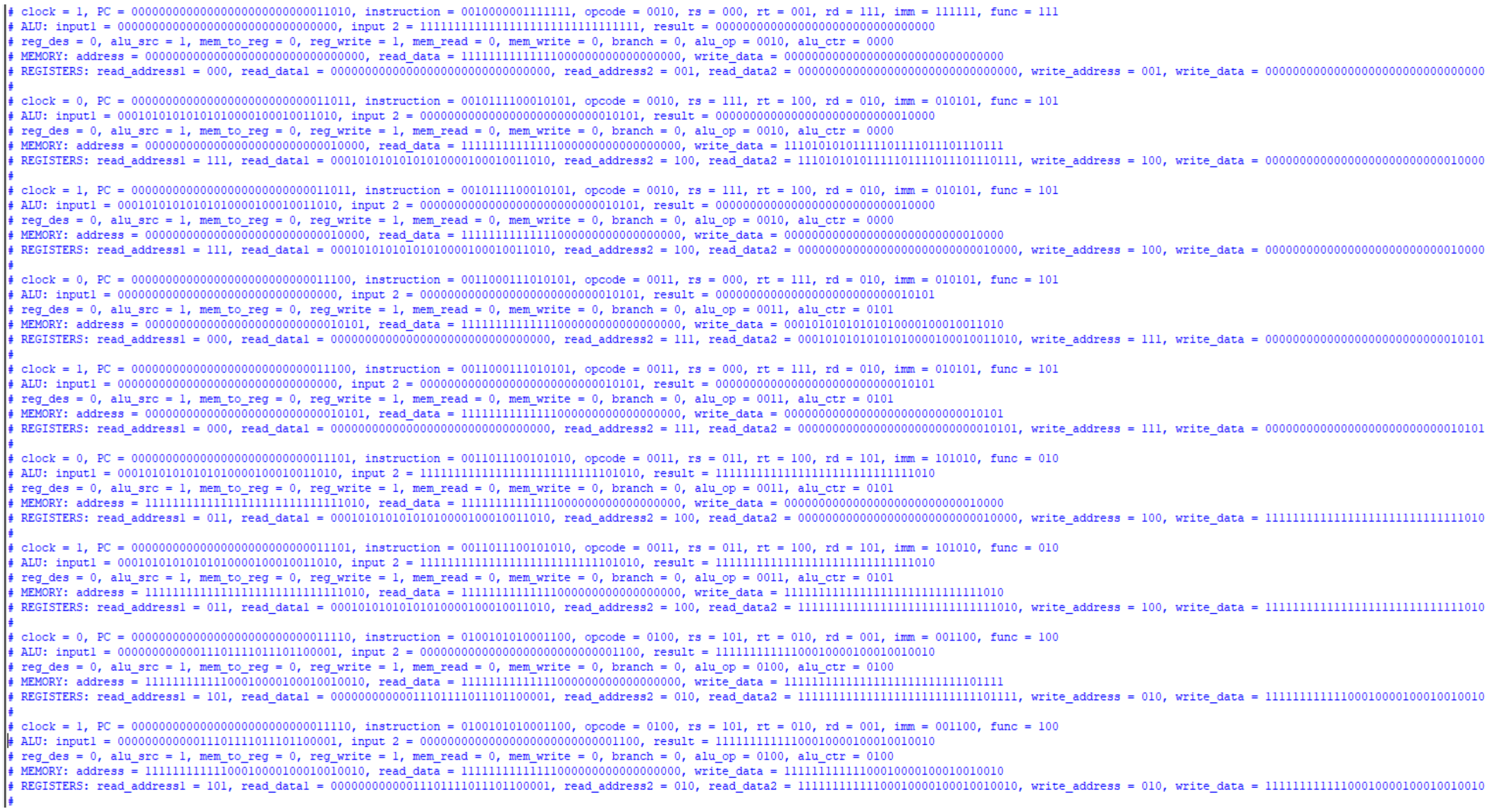
```
# clock = 0, PC = 00000000000000000000000000001011, instruction = 0101101000111111, opcode = 0101, rs = 101, rt = 000, rd = 111, imm = 111111, func = 111
# ALU: input1 = 00010101010100010000100010101010, input 2 = 00000000000000000000000000000000, result = 00000000000000000000000000000000
# reg_des = 0, alu_src = 0, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 0101, alu_ctr = 0110
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111000000000000000000, write_data = 00000000000000000000000000000000
# REGISTERS: read_address1 = 101, read_data1 = 00010101010100010000100010101010, read_address2 = 000, read_data2 = 00000000000000000000000000000000, write_address = 000, write_data = 00000000000000000000000000000000
#
# clock = 1, PC = 00000000000000000000000000001011, instruction = 0101101000111111, opcode = 0101, rs = 101, rt = 000, rd = 111, imm = 111111, func = 111
# ALU: input1 = 00010101010100010000100010101010, input 2 = 00000000000000000000000000000000, result = 00000000000000000000000000000000
# reg_des = 0, alu_src = 0, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 0101, alu_ctr = 0110
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111000000000000000000, write_data = 00000000000000000000000000000000
# REGISTERS: read_address1 = 101, read_data1 = 00010101010100010000100010101010, read_address2 = 000, read_data2 = 00000000000000000000000000000000, write_address = 000, write_data = 00000000000000000000000000000000
#
# clock = 0, PC = 00000000000000000000000000001100, instruction = 0110110111000110, opcode = 0110, rs = 110, rt = 111, rd = 000, imm = 000110, func = 110
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00010101010101010000100010011010, result = 00000000000000000000000000000000
# reg_des = 0, alu_src = 0, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 0110, alu_ctr = 0111
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111000000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 111, read_data2 = 00010101010101010000100010011010, write_address = 111, write_data = 00000000000000000000000000000000
#
# clock = 1, PC = 00000000000000000000000000001100, instruction = 0110110111000110, opcode = 0110, rs = 110, rt = 111, rd = 000, imm = 000110, func = 110
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00010101010101010000100010011010, result = 00000000000000000000000000000000
# reg_des = 0, alu_src = 0, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 0110, alu_ctr = 0111
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111000000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 111, read_data2 = 00010101010101010000100010011010, write_address = 111, write_data = 00000000000000000000000000000000
#
# clock = 0, PC = 00000000000000000000000000001101, instruction = 0110001010000011, opcode = 0110, rs = 001, rt = 010, rd = 000, imm = 000011, func = 011
# ALU: input1 = 00010101010100010000100010001010, input 2 = 00000000000000010000100010100000, result = 00000000000000000000000000000001
# reg_des = 0, alu_src = 0, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 0110, alu_ctr = 0111
# MEMORY: address = 00000000000000000000000000000001, read_data = 11111111111111000000000000000000, write_data = 00000000000000010000100010100000
# REGISTERS: read_address1 = 001, read_data1 = 00010101010100010000100010001010, read_address2 = 010, read_data2 = 00000000000000010000100010100000, write_address = 010, write_data = 00000000000000000000000000000001
#
# clock = 1, PC = 00000000000000000000000000001101, instruction = 0110001010000011, opcode = 0110, rs = 001, rt = 010, rd = 000, imm = 000011, func = 011
# ALU: input1 = 00010101010100010000100010001010, input 2 = 00000000000000010000100010100000, result = 00000000000000000000000000000001
# reg_des = 0, alu_src = 0, mem_to_reg = 0, reg_write = 0, mem_read = 0, mem_write = 0, branch = 1, alu_op = 0110, alu_ctr = 0111
# MEMORY: address = 00000000000000000000000000000001, read_data = 11111111111111000000000000000000, write_data = 00000000000000010000100010100000
# REGISTERS: read_address1 = 001, read_data1 = 00010101010100010000100010001010, read_address2 = 010, read_data2 = 00000000000000010000100010100000, write_address = 010, write_data = 00000000000000000000000000000001
#
# clock = 0, PC = 00000000000000000000000000010000, instruction = 0000110111100010, opcode = 0000, rs = 110, rt = 111, rd = 100, imm = 100010, func = 010
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00010101010101010000100010011010, result = 00000000000000000000000000000000
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0010
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111000000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 111, read_data2 = 00010101010101010000100010011010, write_address = 100, write_data = 00000000000000000000000000000000
#
# clock = 1, PC = 00000000000000000000000000010000, instruction = 0000110111100010, opcode = 0000, rs = 110, rt = 111, rd = 100, imm = 100010, func = 010
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00010101010101010000100010011010, result = 00000000000000000000000000000000
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0010
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111000000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 111, read_data2 = 00010101010101010000100010011010, write_address = 100, write_data = 00000000000000000000000000000000
#
# clock = 0, PC = 00000000000000000000000000010001, instruction = 0000010001011010, opcode = 0000, rs = 010, rt = 001, rd = 011, imm = 011010, func = 010
# ALU: input1 = 00000000000000010000100010100000, input 2 = 00010101010100010000100010001010, result = 11101010101100000000000000010110
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0010
# MEMORY: address = 11101010101100000000000000010110, read_data = 11111111111111000000000000000000, write_data = 00010101010100010000100010001010
# REGISTERS: read_address1 = 010, read_data1 = 00000000000000010000100010100000, read_address2 = 001, read_data2 = 00010101010100010000100010001010, write_address = 011, write_data = 11101010101100000000000000010110
#
```



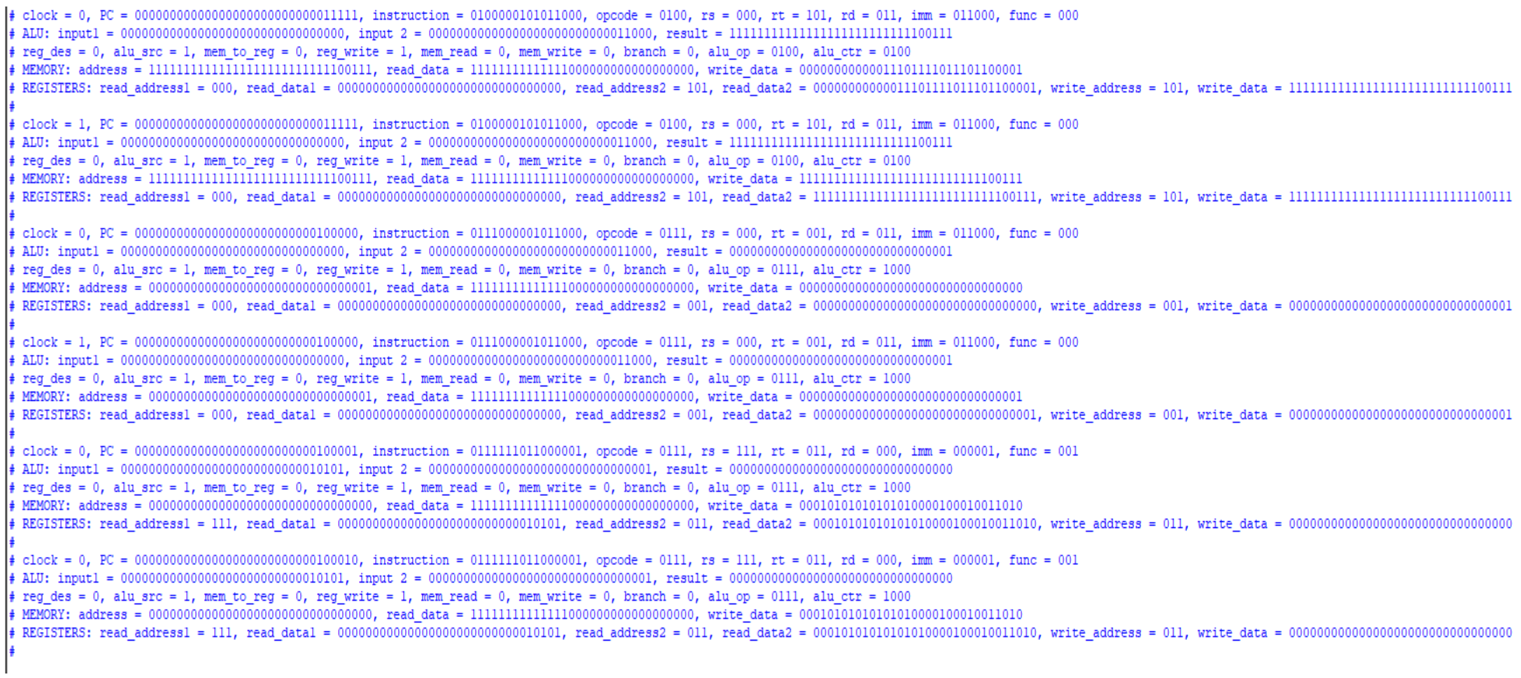
```
# clock = 1, PC = 00000000000000000000000000010001, instruction = 0000010001011010, opcode = 0000, rs = 010, rt = 001, rd = 011, imm = 011010, func = 010
# ALU: input1 = 00000000000000010000100010100000, input 2 = 00010101010100010000100010001010, result = 11101010101100000000000000010110
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0010
# MEMORY: address = 11101010101100000000000000010110, read_data = 11111111111111000000000000000000, write_data = 00010101010100010000100010001010
# REGISTERS: read_address1 = 010, read_data1 = 00000000000000010000100010100000, read_address2 = 001, read_data2 = 00010101010100010000100010001010, write_address = 011, write_data = 11101010101100000000000000010110
#
# clock = 0, PC = 00000000000000000000000000010010, instruction = 0000110111100011, opcode = 0000, rs = 110, rt = 111, rd = 100, imm = 100011, func = 011
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00010101010101010000100010011010, result = 00000000000000000000000000000000
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0011
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111000000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 111, read_data2 = 00010101010101010000100010011010, write_address = 100, write_data = 00000000000000000000000000000000
#
# clock = 1, PC = 00000000000000000000000000010010, instruction = 0000110111100011, opcode = 0000, rs = 110, rt = 111, rd = 100, imm = 100011, func = 011
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00010101010101010000100010011010, result = 00000000000000000000000000000000
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0011
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111000000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 110, read_data1 = 00010101010101010000100010011010, read_address2 = 111, read_data2 = 00010101010101010000100010011010, write_address = 100, write_data = 00000000000000000000000000000000
#
# clock = 0, PC = 00000000000000000000000000010011, instruction = 0000100011010011, opcode = 0000, rs = 100, rt = 011, rd = 010, imm = 010011, func = 011
# ALU: input1 = 00000000000000000000000000000000, input 2 = 11101010101100000000000000010110, result = 11101010101100000000000000010110
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0011
# MEMORY: address = 11101010101100000000000000010110, read_data = 11111111111111000000000000000000, write_data = 11101010101100000000000000010110
# REGISTERS: read_address1 = 100, read_data1 = 00000000000000000000000000000000, read_address2 = 011, read_data2 = 11101010101100000000000000010110, write_address = 010, write_data = 11101010101100000000000000010110
#
# clock = 1, PC = 00000000000000000000000000010011, instruction = 0000100011010011, opcode = 0000, rs = 100, rt = 011, rd = 010, imm = 010011, func = 011
# ALU: input1 = 00000000000000000000000000000000, input 2 = 11101010101100000000000000010110, result = 11101010101100000000000000010110
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0011
# MEMORY: address = 11101010101100000000000000010110, read_data = 11111111111111000000000000000000, write_data = 11101010101100000000000000010110
# REGISTERS: read_address1 = 100, read_data1 = 00000000000000000000000000000000, read_address2 = 011, read_data2 = 11101010101100000000000000010110, write_address = 010, write_data = 11101010101100000000000000010110
#
# clock = 0, PC = 00000000000000000000000000010100, instruction = 0000001010101100, opcode = 0000, rs = 001, rt = 010, rd = 101, imm = 101100, func = 100
# ALU: input1 = 00010101010100010000100010001010, input 2 = 11101010101100000000000000010110, result = 00000000000011101111011101100001
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0100
# MEMORY: address = 00000000000011101111011101100001, read_data = 11111111111111000000000000000000, write_data = 11101010101100000000000000010110
# REGISTERS: read_address1 = 001, read_data1 = 00010101010100010000100010001010, read_address2 = 010, read_data2 = 11101010101100000000000000010110, write_address = 101, write_data = 00000000000011101111011101100001
#
# clock = 1, PC = 00000000000000000000000000010100, instruction = 0000001010101100, opcode = 0000, rs = 001, rt = 010, rd = 101, imm = 101100, func = 100
# ALU: input1 = 00010101010100010000100010001010, input 2 = 11101010101100000000000000010110, result = 00000000000011101111011101100001
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0100
# MEMORY: address = 00000000000011101111011101100001, read_data = 11111111111111000000000000000000, write_data = 11101010101100000000000000010110
# REGISTERS: read_address1 = 001, read_data1 = 00010101010100010000100010001010, read_address2 = 010, read_data2 = 11101010101100000000000000010110, write_address = 101, write_data = 00000000000011101111011101100001
#
# clock = 0, PC = 00000000000000000000000000010101, instruction = 0000000110001100, opcode = 0000, rs = 000, rt = 110, rd = 001, imm = 001100, func = 100
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00010101010101010000100010011010, result = 11101010101010101111011101100101
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0100
# MEMORY: address = 11101010101010101111011101100101, read_data = 11111111111111000000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 110, read_data2 = 00010101010101010000100010011010, write_address = 001, write_data = 11101010101010101111011101100101
#
# clock = 1, PC = 00000000000000000000000000010101, instruction = 0000000110001100, opcode = 0000, rs = 000, rt = 110, rd = 001, imm = 001100, func = 100
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00010101010101010000100010011010, result = 11101010101010101111011101100101
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0100
# MEMORY: address = 11101010101010101111011101100101, read_data = 11111111111111000000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 110, read_data2 = 00010101010101010000100010011010, write_address = 001, write_data = 11101010101010101111011101100101
#
```



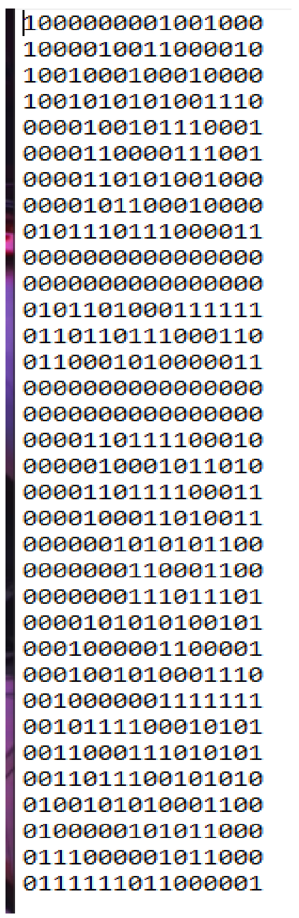
```
# clock = 0, PC = 00000000000000000000000000010110, instruction = 0000000111011101, opcode = 0000, rs = 000, rt = 111, rd = 011, imm = 011101, func = 101
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00010101010101010000100010011010, result = 00010101010101010000100010011010
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0101
# MEMORY: address = 00010101010101010000100010011010, read_data = 11111111111111110000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 111, read_data2 = 00010101010101010000100010011010, write_address = 011, write_data = 00010101010101010000100010011010
#
# clock = 1, PC = 00000000000000000000000000010110, instruction = 0000000111011101, opcode = 0000, rs = 000, rt = 111, rd = 011, imm = 011101, func = 101
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00010101010101010000100010011010, result = 00010101010101010000100010011010
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0101
# MEMORY: address = 00010101010101010000100010011010, read_data = 11111111111111110000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 111, read_data2 = 00010101010101010000100010011010, write_address = 011, write_data = 00010101010101010000100010011010
#
# clock = 0, PC = 00000000000000000000000000010111, instruction = 0000101010100101, opcode = 0000, rs = 101, rt = 010, rd = 100, imm = 100101, func = 101
# ALU: input1 = 00000000000011101111011101100001, input 2 = 11101010101100000000000000010110, result = 11101010101111101111011101110111
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0101
# MEMORY: address = 11101010101111101111011101110111, read_data = 11111111111111110000000000000000, write_data = 11101010101100000000000000010110
# REGISTERS: read_address1 = 101, read_data1 = 00000000000011101111011101100001, read_address2 = 010, read_data2 = 11101010101100000000000000010110, write_address = 100, write_data = 11101010101111101111011101110111
#
# clock = 1, PC = 00000000000000000000000000010111, instruction = 0000101010100101, opcode = 0000, rs = 101, rt = 010, rd = 100, imm = 100101, func = 101
# ALU: input1 = 00000000000011101111011101100001, input 2 = 11101010101100000000000000010110, result = 11101010101111101111011101110111
# reg_des = 1, alu_src = 0, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0000, alu_ctr = 0101
# MEMORY: address = 11101010101111101111011101110111, read_data = 11111111111111110000000000000000, write_data = 11101010101100000000000000010110
# REGISTERS: read_address1 = 101, read_data1 = 00000000000011101111011101100001, read_address2 = 010, read_data2 = 11101010101100000000000000010110, write_address = 100, write_data = 11101010101111101111011101110111
#
# clock = 0, PC = 00000000000000000000000000011000, instruction = 0001000001100001, opcode = 0001, rs = 000, rt = 001, rd = 100, imm = 100001, func = 001
# ALU: input1 = 00000000000000000000000000000000, input 2 = 11111111111111111111111111100001, result = 11111111111111111111111111100001
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0001, alu_ctr = 0001
# MEMORY: address = 11111111111111111111111111100001, read_data = 11111111111111110000000000000000, write_data = 11101010101010101111011101100101
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 001, read_data2 = 11101010101010101111011101100101, write_address = 001, write_data = 11111111111111111111111111100001
#
# clock = 1, PC = 00000000000000000000000000011000, instruction = 0001000001100001, opcode = 0001, rs = 000, rt = 001, rd = 100, imm = 100001, func = 001
# ALU: input1 = 00000000000000000000000000000000, input 2 = 11111111111111111111111111100001, result = 11111111111111111111111111100001
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0001, alu_ctr = 0001
# MEMORY: address = 11111111111111111111111111100001, read_data = 11111111111111110000000000000000, write_data = 11111111111111111111111111100001
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 001, read_data2 = 11111111111111111111111111100001, write_address = 001, write_data = 11111111111111111111111111100001
#
# clock = 0, PC = 00000000000000000000000000011001, instruction = 0001001010001110, opcode = 0001, rs = 001, rt = 010, rd = 001, imm = 001110, func = 110
# ALU: input1 = 11111111111111111111111111100001, input 2 = 00000000000000000000000000001110, result = 11111111111111111111111111101111
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0001, alu_ctr = 0001
# MEMORY: address = 11111111111111111111111111101111, read_data = 11111111111111110000000000000000, write_data = 11101010101100000000000000010110
# REGISTERS: read_address1 = 001, read_data1 = 11111111111111111111111111100001, read_address2 = 010, read_data2 = 11101010101100000000000000010110, write_address = 010, write_data = 11111111111111111111111111101111
#
# clock = 1, PC = 00000000000000000000000000011001, instruction = 0001001010001110, opcode = 0001, rs = 001, rt = 010, rd = 001, imm = 001110, func = 110
# ALU: input1 = 11111111111111111111111111100001, input 2 = 00000000000000000000000000001110, result = 11111111111111111111111111101111
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0001, alu_ctr = 0001
# MEMORY: address = 11111111111111111111111111101111, read_data = 11111111111111110000000000000000, write_data = 11111111111111111111111111101111
# REGISTERS: read_address1 = 001, read_data1 = 11111111111111111111111111100001, read_address2 = 010, read_data2 = 11111111111111111111111111101111, write_address = 010, write_data = 11111111111111111111111111101111
#
# clock = 0, PC = 00000000000000000000000000011010, instruction = 0010000001111111, opcode = 0010, rs = 000, rt = 001, rd = 111, imm = 111111, func = 111
# ALU: input1 = 00000000000000000000000000000000, input 2 = 11111111111111111111111111111111, result = 00000000000000000000000000000000
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0010, alu_ctr = 0000
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111110000000000000000, write_data = 11111111111111111111111111100001
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 001, read_data2 = 11111111111111111111111111100001, write_address = 001, write_data = 00000000000000000000000000000000
#
```



```
# clock = 1, PC = 00000000000000000000000000011010, instruction = 0010000001111111, opcode = 0010, rs = 000, rt = 001, rd = 111, imm = 111111, func = 111
# ALU: input1 = 00000000000000000000000000000000, input 2 = 11111111111111111111111111111111, result = 00000000000000000000000000000000
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0010, alu_ctr = 0000
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111110000000000000000, write_data = 00000000000000000000000000000000
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 001, read_data2 = 00000000000000000000000000000000, write_address = 001, write_data = 00000000000000000000000000000000
#
# clock = 0, PC = 00000000000000000000000000011011, instruction = 0010111100010101, opcode = 0010, rs = 111, rt = 100, rd = 010, imm = 010101, func = 101
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00000000000000000000000000010101, result = 00000000000000000000000000010000
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0010, alu_ctr = 0000
# MEMORY: address = 00000000000000000000000000010000, read_data = 11111111111111110000000000000000, write_data = 11101010101111101111011101110111
# REGISTERS: read_address1 = 111, read_data1 = 00010101010101010000100010011010, read_address2 = 100, read_data2 = 11101010101111101111011101110111, write_address = 100, write_data = 00000000000000000000000000010000
#
# clock = 1, PC = 00000000000000000000000000011011, instruction = 0010111100010101, opcode = 0010, rs = 111, rt = 100, rd = 010, imm = 010101, func = 101
# ALU: input1 = 00010101010101010000100010011010, input 2 = 00000000000000000000000000010101, result = 00000000000000000000000000010000
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0010, alu_ctr = 0000
# MEMORY: address = 00000000000000000000000000010000, read_data = 11111111111111110000000000000000, write_data = 00000000000000000000000000010000
# REGISTERS: read_address1 = 111, read_data1 = 00010101010101010000100010011010, read_address2 = 100, read_data2 = 00000000000000000000000000010000, write_address = 100, write_data = 00000000000000000000000000010000
#
# clock = 0, PC = 00000000000000000000000000011100, instruction = 0011000111010101, opcode = 0011, rs = 000, rt = 111, rd = 010, imm = 010101, func = 101
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000010101, result = 00000000000000000000000000010101
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0011, alu_ctr = 0101
# MEMORY: address = 00000000000000000000000000010101, read_data = 11111111111111110000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 111, read_data2 = 00010101010101010000100010011010, write_address = 111, write_data = 00000000000000000000000000010101
#
# clock = 1, PC = 00000000000000000000000000011100, instruction = 0011000111010101, opcode = 0011, rs = 000, rt = 111, rd = 010, imm = 010101, func = 101
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000010101, result = 00000000000000000000000000010101
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0011, alu_ctr = 0101
# MEMORY: address = 00000000000000000000000000010101, read_data = 11111111111111110000000000000000, write_data = 00000000000000000000000000010101
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 111, read_data2 = 00000000000000000000000000010101, write_address = 111, write_data = 00000000000000000000000000010101
#
# clock = 0, PC = 00000000000000000000000000011101, instruction = 0011011100101010, opcode = 0011, rs = 011, rt = 100, rd = 101, imm = 101010, func = 010
# ALU: input1 = 00010101010101010000100010011010, input 2 = 11111111111111111111111111101010, result = 11111111111111111111111111111010
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0011, alu_ctr = 0101
# MEMORY: address = 11111111111111111111111111111010, read_data = 11111111111111110000000000000000, write_data = 00000000000000000000000000010000
# REGISTERS: read_address1 = 011, read_data1 = 00010101010101010000100010011010, read_address2 = 100, read_data2 = 00000000000000000000000000010000, write_address = 100, write_data = 11111111111111111111111111111010
#
# clock = 1, PC = 00000000000000000000000000011101, instruction = 0011011100101010, opcode = 0011, rs = 011, rt = 100, rd = 101, imm = 101010, func = 010
# ALU: input1 = 00010101010101010000100010011010, input 2 = 11111111111111111111111111101010, result = 11111111111111111111111111111010
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0011, alu_ctr = 0101
# MEMORY: address = 11111111111111111111111111111010, read_data = 11111111111111110000000000000000, write_data = 11111111111111111111111111111010
# REGISTERS: read_address1 = 011, read_data1 = 00010101010101010000100010011010, read_address2 = 100, read_data2 = 11111111111111111111111111111010, write_address = 100, write_data = 11111111111111111111111111111010
#
# clock = 0, PC = 00000000000000000000000000011110, instruction = 0100101010001100, opcode = 0100, rs = 101, rt = 010, rd = 001, imm = 001100, func = 100
# ALU: input1 = 00000000000011101111011101100001, input 2 = 00000000000000000000000000001100, result = 11111111111100010000100010010010
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0100, alu_ctr = 0100
# MEMORY: address = 11111111111100010000100010010010, read_data = 11111111111111110000000000000000, write_data = 11111111111111111111111111101111
# REGISTERS: read_address1 = 101, read_data1 = 00000000000011101111011101100001, read_address2 = 010, read_data2 = 11111111111111111111111111101111, write_address = 010, write_data = 11111111111100010000100010010010
#
# clock = 1, PC = 00000000000000000000000000011110, instruction = 0100101010001100, opcode = 0100, rs = 101, rt = 010, rd = 001, imm = 001100, func = 100
# ALU: input1 = 00000000000011101111011101100001, input 2 = 00000000000000000000000000001100, result = 11111111111100010000100010010010
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0100, alu_ctr = 0100
# MEMORY: address = 11111111111100010000100010010010, read_data = 11111111111111110000000000000000, write_data = 11111111111100010000100010010010
# REGISTERS: read_address1 = 101, read_data1 = 00000000000011101111011101100001, read_address2 = 010, read_data2 = 11111111111100010000100010010010, write_address = 010, write_data = 11111111111100010000100010010010
#
```



```
# clock = 0, PC = 00000000000000000000000000011111, instruction = 0100000101011000, opcode = 0100, rs = 000, rt = 101, rd = 011, imm = 011000, func = 000
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000011000, result = 11111111111111111111111111100111
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0100, alu_ctr = 0100
# MEMORY: address = 11111111111111111111111111100111, read_data = 11111111111111100000000000000000, write_data = 00000000000011101111011101100001
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 101, read_data2 = 00000000000011101111011101100001, write_address = 101, write_data = 11111111111111111111111111100111
#
# clock = 1, PC = 00000000000000000000000000011111, instruction = 0100000101011000, opcode = 0100, rs = 000, rt = 101, rd = 011, imm = 011000, func = 000
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000011000, result = 11111111111111111111111111100111
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0100, alu_ctr = 0100
# MEMORY: address = 11111111111111111111111111100111, read_data = 11111111111111100000000000000000, write_data = 11111111111111111111111111100111
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 101, read_data2 = 11111111111111111111111111100111, write_address = 101, write_data = 11111111111111111111111111100111
#
# clock = 0, PC = 00000000000000000000000000100000, instruction = 0111000001011000, opcode = 0111, rs = 000, rt = 001, rd = 011, imm = 011000, func = 000
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000011000, result = 00000000000000000000000000000001
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0111, alu_ctr = 1000
# MEMORY: address = 00000000000000000000000000000001, read_data = 11111111111111100000000000000000, write_data = 00000000000000000000000000000000
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 001, read_data2 = 00000000000000000000000000000000, write_address = 001, write_data = 00000000000000000000000000000001
#
# clock = 1, PC = 00000000000000000000000000100000, instruction = 0111000001011000, opcode = 0111, rs = 000, rt = 001, rd = 011, imm = 011000, func = 000
# ALU: input1 = 00000000000000000000000000000000, input 2 = 00000000000000000000000000011000, result = 00000000000000000000000000000001
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0111, alu_ctr = 1000
# MEMORY: address = 00000000000000000000000000000001, read_data = 11111111111111100000000000000000, write_data = 00000000000000000000000000000001
# REGISTERS: read_address1 = 000, read_data1 = 00000000000000000000000000000000, read_address2 = 001, read_data2 = 00000000000000000000000000000001, write_address = 001, write_data = 00000000000000000000000000000001
#
# clock = 0, PC = 00000000000000000000000000100001, instruction = 0111111011000001, opcode = 0111, rs = 111, rt = 011, rd = 000, imm = 000001, func = 001
# ALU: input1 = 00000000000000000000000000010101, input 2 = 00000000000000000000000000000001, result = 00000000000000000000000000000000
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0111, alu_ctr = 1000
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111100000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 111, read_data1 = 00000000000000000000000000010101, read_address2 = 011, read_data2 = 00010101010101010000100010011010, write_address = 011, write_data = 00000000000000000000000000000000
#
# clock = 0, PC = 00000000000000000000000000100010, instruction = 0111111011000001, opcode = 0111, rs = 111, rt = 011, rd = 000, imm = 000001, func = 001
# ALU: input1 = 00000000000000000000000000010101, input 2 = 00000000000000000000000000000001, result = 00000000000000000000000000000000
# reg_des = 0, alu_src = 1, mem_to_reg = 0, reg_write = 1, mem_read = 0, mem_write = 0, branch = 0, alu_op = 0111, alu_ctr = 1000
# MEMORY: address = 00000000000000000000000000000000, read_data = 11111111111111100000000000000000, write_data = 00010101010101010000100010011010
# REGISTERS: read_address1 = 111, read_data1 = 00000000000000000000000000010101, read_address2 = 011, read_data2 = 00010101010101010000100010011010, write_address = 011, write_data = 00000000000000000000000000000000
#
```



```
1000000001001000
1000010011000010
1001000100010000
1001010101001110
0000100101110001
0000110000111001
0000110101001000
0000101100010000
0101110111000011
0000000000000000
0000000000000000
0101101000111111
0110110111000110
0110001010000011
0000000000000000
0000000000000000
0000110111100010
0000010001011010
0000110111100011
0000100011010011
0000001010101100
0000000110001100
0000000111011101
0000101010100101
0001000001100001
0001001010001110
0010000001111111
0010111100010101
0011000111010101
0011011100101010
0100101010001100
0100000101011000
0111000001011000
0111111011000001
```

Instruction Memory File:

Instruction Test Order:

LW, SW, ADD, AND, BEQ, BNE, SUB, XOR, NOR, OR, ADDI, ANDI, ORI, NORI, SLTI (Each of them tested 2 times)

(Between branch instructions, there are NOP instructions)

I added “registers.mem”, “instruction_memory.mem” and “data_memory.mem” files to the project. You can backup these files and test my homework. So, you can see the changes in the “registers.mem” and “data_memory.mem” files.