Introduction to process Architecture
SEQ & PIPE– Architecture
Team-2 Members:

Shaik Affan Adeeb
2022102054

V.V.S.R Chetan Krishna
2022102047

Sequential Architecture:

- We need to design a Y86-64 Processor using verilog by implementing the sequential design.
- Our Y86-64 processor is capable of running following instructions : halt, nop, rrmovq, vmovle, cmovl, cmove, cmovne, cmovge, cmovg, irmovq, rmmovq, mrmovq, addq, subq, andq, xorq, jmp, jle, jl, je, jne, jge, jg, call, ret, pushq, popq.
- Now let's discuss the various stages involved in our SEQ design.

Fetch Stage :

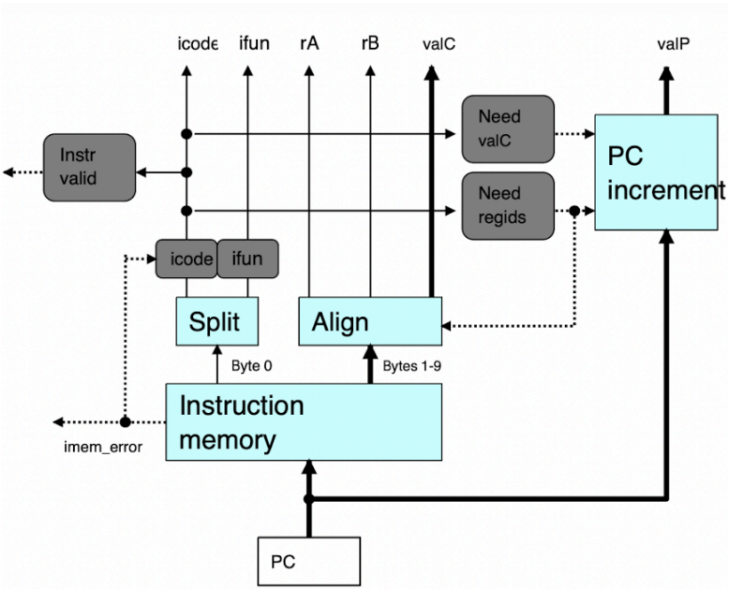
- The main purpose of the fetch stage is to fetch data from the instruction memory based on the Program Counter (PC) value and updating the PC to the next instruction's address.
- In the fetch stage, we typically take inputs on the negative edge of the clock, and after processing within the module, the outputs are ready on the positive edge of the clock. This process ensures proper synchronization and smooth operation.
- To do this, we split the instruction memory at the byte where the PC points, dividing it into two parts: the most significant 4 bits (MSB) and the least significant 4 bits (LSB). The MSB represents the icode, while the LSB represents the instruction function (Ifun).

1st Byte	**i_code (MSB 4bits)**	**i_fun (LSB 4bits)**
2nd Byte	**r_A (MSB 4bits)**	**r_B (LSB 4bits)**
3rd Byte	**Val C**	
4th Byte	**Val C**	
5th Byte	**Val C**	
6th Byte	**Val C**	
7th Byte	**Val C**	
8th Byte	**Val C**	
9th Byte	**Val C**	
10th Byte	**Val C**	

Our instruction set implementaion diagram. In our implementation we are using PC+8 bits instead of PC+1 Byte. So in report for simplicity we are using PC+1 or PC+x bytes everywhere instead of PC+8 bits.

- The fetch stage reads the bytes of an instruction from memory, using the program counter (PC) as the memory address. From the instruction it extracts the two 4-bit portions of the instruction specifier byte, referred to as icode (the instruction code) and ifun (the instruction function). It possibly fetches a register specifier byte, giving one or both of the register operand specifiers rA and rB. It also possibly fetches a 4-byte constant word valC. It computes valP to be the address of the instruction following the current one in sequential order. That is, valP equals the value of the PC plus the length of the fetched instruction.

- The inputs to the fetch module include the instruction memory itself and the PC value. From these inputs, we derive various outputs such as register encodings (`rA` and `rB`), `icode`, `Ifun`, the value to be stored in the Constant (`valC`), the next PC value (`valP`), instruction validity (`instr_valid`), and any errors that may occur during instruction memory access (`imem_error`).



block digram showing fetch operations

Brief expalanation of each instruction:

Halt :

- To implement this, the fetch stage checks the instruction memory at the PC address. If the icode is 0000(Ifun is anything), it sets the halt flag to 1 and increments the next PC value (valP) to PC+8 (In our case as we have taken it in bits so we do PC+8 bits). This stops the iterations as no changes occur in the execute, decode, or memory blocks due to the halt flag being set.

```
1        /// HALT
2        if(Ins_Code == 0 && Ins_fun == 0)begin
3            // Do not do anything
4            // to write something such that the code
                stops executing
5        end
```

Nop :

- After checking the PC value to locate the instruction, if the icode is 0001, fetch stage encounters a "nop" (no operation) instruction, with any Ifun value. Then we increment the PC value (valP) to PC+1(In our implementation PC+8 bits). No other operations occur, and the processor moves directly to the next iteration since the PC is updated after each instruction execution.

```
1        /// No operation nop
2        if(Ins_Code == 1 && Ins_fun == 0)begin
3            // skip this instruction
4            need_regids = 0;
5            need_Val_C = 0;
6            //PC_increment_bytes = 8 + need_regids * 8
                + need_Val_C * 64;
7            Val_P =
                memory_for_instructions[$unsigned(PC_adress)
                + 8 + need_regids * 8 + need_Val_C *
                64][63 : 0];
8        end
```

CMOVXX :

- For the cmovxx instruction, the fetch stage begins by looking at the instruction code (icode), which is 00010, and the function code (ifun), represented as xxxx(ranging from 0 to 6). Since the icode is 0010, the fetch stage knows it needs to fetch data from the instruction memory at addresses PC and PC+1(PC+8 bits in our implementation) to get both the icode, ifun and r_A, r_B encodings.

- After retrieving this data, it increments the Program Counter (PC) to PC+2 since it has processed instructions from addresses PC and PC+1.

```
1            /// conditional move cmovXX
2        if(Ins_Code == 2)begin
3            if(0 <= Ins_fun <= 6)begin
4                rA = instruction_set
                     [($unsigned(PC_adress)/8) + 1][7 :
                     4];
5                rB = instruction_set
                     [($unsigned(PC_adress)/8) + 1][3 :
                     0];
6                need_Val_C = 0;
7                // since Val_C not present and needed,
                     we do not fetch val_C.
8                need_regids = 1;
9                // since regids needed, we fetched
                     registers ra and rb
10               //PC_increment_bytes = 8 + need_regids
                     * 8 + need_Val_C * 64;
11               Val_P =
                     memory_for_instructions[$unsigned(PC_adress)
                     + 8 + need_regids * 8 + need_Val_C *
                     64][63:0];
12           end
13           if(Ins_fun >6)begin
14               $display("For␣cmovXX␣the␣function␣code␣
                     range␣between␣0␣to␣6");
15           end
16       end
```

IRMOVQ :

- As the name implies, the fetch stage immediately moves data from the constant value (valC) to the specified register (rA or rB).
- In decoding the instruction memory from PC to PC+9, the PC provides the instruction code (ICode), which is 0011, along with the function code (IFun). The next memory location, PC+1, provides values for rA and rB. Finally, the memory locations from PC+2 to PC+9 provide the constant value (valC).
- Since it fetches all 10 PC instructions, it increments the PC by 10 (PC+10) after processing.
- So, in brief: PC provides ICode and IFun, PC+1 provides rA and rB, and PC+2 to PC+9 provide valC. After processing, the next PC value (valP) is set to PC+10.

```
/// immediate Resigter move irmovq
        if(Ins_Code == 3)begin
            if(Ins_fun != 0)begin
                $display("ERROR: for irmovq : function
                    code is 0, but not entered 0");

            end
            if(Ins_fun == 0)begin
                // immediate register move.
                need_Val_C = 1;
                need_regids = 1;
rA = instruction_set[($unsigned(PC_adress)/8)+1][7:4];
rB = instruction_set[($unsigned(PC_adress)/8)+1][3:0];
Val_C = {instruction_set [($unsigned(PC_adress)/8)+2],
instruction_set [($unsigned(PC_adress)/8)+3],
instruction_set [($unsigned(PC_adress)/8)+4],
instruction_set [($unsigned(PC_adress)/8)+5],
instruction_set [($unsigned(PC_adress)/8)+6],
instruction_set [($unsigned(PC_adress)/8)+7],
instruction_set [($unsigned(PC_adress)/8)+8],
instruction_set [($unsigned(PC_adress)/8)+9]};
Val_P = memory_for_instructions [$unsigned(PC_adress) +
   8 + need_regids * 8 + need_Val_C * 64][63 : 0];

            end
        end
```

RMMOVQ :

- As the name implies, this stage moves data from registers to memory, based on the values of rA and rB.

- In decoding the instruction memory from PC to PC+9, the PC provides the instruction code (ICode), which is 0100, along with the function code (IFun). The next memory location, PC+1, provides values for rA and rB. The subsequent memory locations from PC+2 to PC+9 provide the constant value (valC).

- Since it processes all 10 PC instructions, it increments the PC by 10 (PC+10) after completion. So, in brief: PC provides ICode and IFun, PC+1 provides rA and rB, and PC+2 to PC+9 provide valC. After processing, the next PC value (valP) is set to PC+10.

```
if(Ins_Code == 4)begin
if(Ins_fun != 0)begin
$display("ERROR: for rmmovq : function code is 0, but
   not entered 0");
        end
if(Ins_fun == 0)begin
need_Val_C = 1;
need_regids = 1;
rA = instruction_set[($unsigned(PC_adress)/8) + 1][7:4];
rB = instruction_set[($unsigned(PC_adress)/8) + 1][3:0];
Val_C = same as of previous operation updation
Val_P = memory_for_instructions [$unsigned(PC_adress) +
   8 + need_regids * 8 + need_Val_C * 64][63 : 0];
        end
    end
```

MRMOVQ :

- As the name implies, this stage moves data from memory to registers, based on the values of rA and rB.

- When decoding the instruction memory from PC to PC+9, the PC provides the instruction code (ICode), which is 0100, along with the function code (IFun). At PC+1, the values for rA and rB are found, while the values from memory locations PC+2 to PC+9 provide the constant value (valC).

- After processing all 10 PC instructions, the PC is incremented by 10 (PC+10). So, to brief: PC provides ICode and IFun, PC+1 provides rA and rB, and PC+2 to PC+9 provide valC. After processing, the next PC value (valP) is set to PC+10.

```
/// memory to register move mrmovq
if(Ins_Code == 5)begin
if(Ins_fun == 0)begin
need_Val_C = 1;
need_regids = 1;
rA = instruction_set[($unsigned(PC_adress)/8) + 1][7:4];
rB = instruction_set[($unsigned(PC_adress)/8) + 1][3:0];
Val_C = same as of previous operation updation
Val_P = memory_for_instructions [$unsigned(PC_adress) +
   8 + need_regids * 8 + need_Val_C * 64][63 : 0];
        end
    end
```

FETCH FOR OPQ :

- As the name suggests, this instruction performs addition, subtraction, and logical operations (AND, XOR) on the register addresses rA and rB, depending on the function code (ifun).

- For the OPQ instruction, the instruction code (Icode) must be 0110, and the function code (ifun) determines the specific operation to execute.

- To fetch the instruction, it reads the memory locations from PC to PC+1, extracting the Icode and ifun from PC, and rA and rB from PC+1. Then, it increments the PC value by 2, setting valP to PC+2.

```
/// Arithematic memory operations: OPq
if(Ins_Code == 6)begin
    if(Ins_fun >3)begin
    $display("For␣OPq␣the␣i_fun␣code␣range␣is␣0␣to␣3");
    end
if(Ins_fun >=0 && Ins_fun <=3)begin
need_Val_C = 0;
need_regids = 1;
rA = instruction_set[($unsigned(PC_adress)/8) + 1][7:4];
rB = instruction_set[($unsigned(PC_adress)/8) + 1][3:0];
```

```
Val_P = memory_for_instructions [$unsigned(PC_adress) +
   8 + need_regids * 8 + need_Val_C * 64][63 : 0];
end
end
```

FETCH FOR JXX :

- This instruction, as the name implies, facilitates jumping from one instruction to another based on certain conditions.
- For the JXX instruction, the instruction code (Icode) must be 0111, while the function code (Ifun) doesn't affect the execution or PC update.
- To fetch this instruction, it reads memory locations from PC to PC+8. It extracts Icode and Ifun from PC, and valC from PC+1 to PC+8. Then, it increments the PC value by 9, setting valP to PC+9.
- So, in short: the instruction code must be 0111, Ifun doesn't matter, PC provides Icode, and valC comes from PC+1 to PC+8. After processing, the next PC value (valP) is set to PC+9.

```
        /// JUMP operations
if(Ins_Code == 7)begin
    if(Ins_fun > 6)begin
    $display("Error : for JUMP operations , the function
       codes between 0 and 6");
    end
if(Ins_fun>=0 && Ins_fun <=6)begin
need_regids = 0;
need_Val_C =  1;
Val_C = {instruction_set [($unsigned(PC_adress)/8) + 1],
instruction_set [($unsigned(PC_adress)/8) + 2],
instruction_set [($unsigned(PC_adress)/8) + 3],
instruction_set [($unsigned(PC_adress)/8) + 4],
instruction_set [($unsigned(PC_adress)/8) + 5],
instruction_set [($unsigned(PC_adress)/8) + 6],
instruction_set [($unsigned(PC_adress)/8) + 7],
instruction_set [($unsigned(PC_adress)/8) + 8]};
Val_P = memory_for_instructions [$unsigned(PC_adress) +
   8 + need_regids * 8 + need_Val_C * 64][63 : 0];
end
end
```

FETCH FOR CALL :

- This instruction, as indicated by its name, retrieves a value from the register pointed to by the stack within the storage of the Y86-64 architecture.
- For the Call instruction, the instruction code (icode) should be 1000, while the function code (ifun) can be any value (xxxx).
- During the fetch stage, instructions are fetched from the instruction memory ranging from PC to PC+8. PC provides the Icode and Ifun, and from PC to PC+8 provides the valC. Therefore, the PC needs to be incremented by 9, resulting in valP=PC+9.

```
        /// CALL operations : call
if(Ins_Code == 8)begin
    if(Ins_fun != 0)begin
$display("ERROR : for call the function code is 0");
    end
if(Ins_fun == 0)begin
need_regids = 0;
need_Val_C = 1;
Val_C = {instruction_set [($unsigned(PC_adress)/8) + 1],
instruction_set [($unsigned(PC_adress)/8) + 2],
instruction_set [($unsigned(PC_adress)/8) + 3],
instruction_set [($unsigned(PC_adress)/8) + 4],
instruction_set [($unsigned(PC_adress)/8) + 5],
instruction_set [($unsigned(PC_adress)/8) + 6],
instruction_set [($unsigned(PC_adress)/8) + 7],
instruction_set [($unsigned(PC_adress)/8) + 8]};
Val_P = memory_for_instructions [$unsigned(PC_adress) +
   8 + need_regids * 8 + need_Val_C * 64][63 : 0];
end
end
```

FETCH FOR RETURN :

- This instruction, as its name implies, returns the value pointed to by the stack. For the Return instruction, the instruction code (Icode) is 1001, and the function code (Ifun) is considered.
- During the fetch stage, only one instruction is fetched from PC, providing the Icode and Ifun. Therefore, the PC is incremented only once, resulting in valP=PC+1.

```
        /// Return operation : ret
        if(Ins_Code == 9)begin
            if(Ins_fun != 0)begin
                $display("Function code for return is
                    0");
            end

            if(Ins_fun == 0)begin
                //no need to fetch anything, we just
                    need to update pc to the next
                    instruction after the call

                need_regids = 0;
                need_Val_C = 0;
            end
        end
```

Pushq :

- This instruction, pushes data to a lower memory location, typically decremented by 8. Therefore, it's usually stored at the initial memory minus 8. For Push instruction, the instruction code (Icode) is 1010, and the function code (Ifun) is considered.
- During the fetch stage, it retrieves instructions from the instruction memory at PC and PC+1. PC provides the Icode (1010) and Ifun, while PC+1 provides the values for rA and rB. After this, the PC is incremented by 2, resulting in valP=PC+2.

```
        /// Pushq
    if(Ins_Code == 10)begin
        if(Ins_fun != 0)begin
            $display("Function code for pushq is 0");
            end
if(Ins_fun == 0)begin
need_regids = 1;
need_Val_C = 0;
rA = instruction_set[($unsigned(PC_adress)/8) + 1][7:4];
rB = instruction_set[($unsigned(PC_adress)/8) + 1][3:0];
Val_P = memory_for_instructions [$unsigned(PC_adress) +
   8 + need_regids * 8 + need_Val_C * 64][63 : 0];
            end
        end
```

Popq :

- This instruction, pops data from a higher memory location, typically incremented by 8. Therefore, it's usually stored at the initial memory plus 8. For the Pop instruction, the instruction code (Icode) is 1011, and the function code (Ifun) is considered.

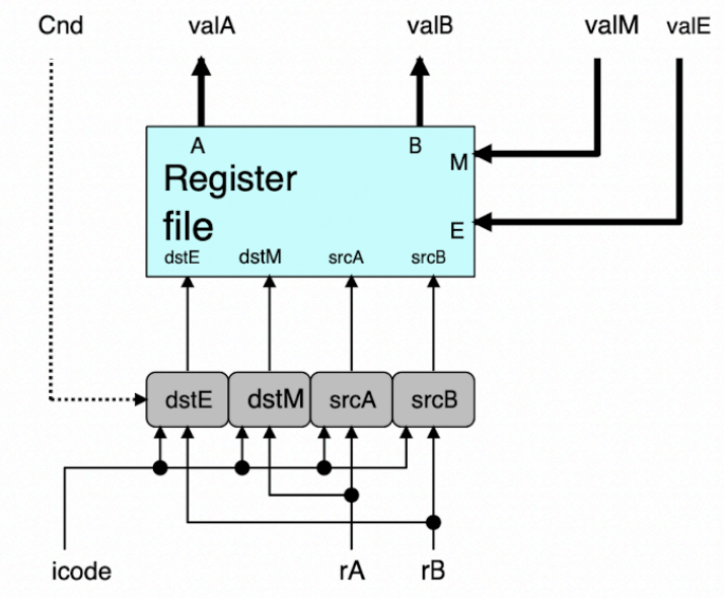
- During the fetch stage, it retrieves instructions from the instruction memory at PC and PC+1. PC provides the Icode (1011) and Ifun, while PC+1 provides the values for rA and rB. After this, the PC is incremented by 2, resulting in valP=PC+2.

```
 /// Popq
if(Ins_Code == 11)begin
    if(Ins_fun != 0)begin
    $display("Function code for popq is 0");
    end
if(Ins_fun == 0)begin
need_Val_C = 0;
need_regids = 1;
rA = instruction_set[($unsigned(PC_adress)/8) + 1][7:4];
rB = instruction_set[($unsigned(PC_adress)/8) + 1][3:0];
Val_P = memory_for_instructions [$unsigned(PC_adress) +
   8 + need_regids * 8 + need_Val_C * 64][63 : 0];
    end
end
```

.....

Decode and Write Back Stage :

- This block provides either the values of valA and valB, or just valA or just valB, depending on the case being used.



block digram showing decode operations

Brief expalanation of each instruction :

CMOVxx:

Decode: Only the value of A is needed, which will be fetched from the register memory rA.
Writeback: If the condition is true (1), the value of register memory rB is stored as valE.

IRMOVQ:

Decode: No decoding is required for IRMOVQ.
Writeback: The value of valE is assigned to the register memory rB.

MRMOVQ:

Decode: The value of valB is fetched from the register memory rB.
Writeback: The value of valM is stored in the register memory rB.

OPQ instruction:

Decode: Fetch valA and valB from the register memory rA and rB respectively.
Writeback: Store the value of valE in the register memory of rB.

CALL instruction:

Decode: Fetch valB from the stack pointer rsp.
Writeback: Update the stack pointer rsp with the value of valE.

RETURN :

Decode: Retrieve valA and valB from the register memory of rsp.
Writeback: Update the stack pointer rsp with the executed value valE.

PUSHQ:

Decode: Obtain valA from register memory rA and valB from rsp.
Writeback: Update the stack pointer register rsp with the value of valE.

POPQ :

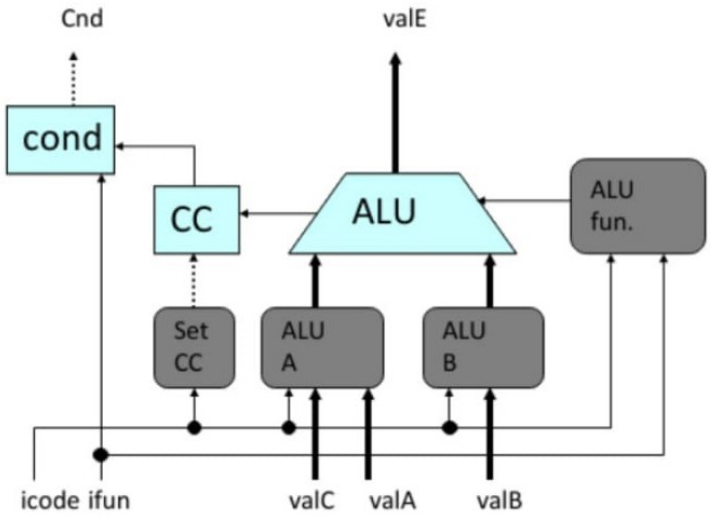
Decode: Fetch valA and valB from the register memory of rsp.
Writeback: Update the rsp with valE and R[rA] with valM.

- Other instructions won't require any additional decoding or writeback steps.
- Code for decode and write back stage is already uploaded.

. . . . . .

Execute Stage :

- During the execute stage, the main focus is on the operations of the Arithmetic Logic Unit (ALU). It performs operations based on the values of valA and valB obtained during the decode stage. The outputs typically include Zf (zero flag), Sf (sign flag), Of (overflow flag), and valE (result of the operation). These outputs are crucial for determining the status and outcome of the operation being executed.
- The Inputs to this module are clk,icode,ifun,valA,valB,valC and outputs are cnd signal,valE,cc.



block digram showing execute stage

Brief expalanation of each instruction :

- For the HALT instruction, no execution occurs.
- Similarly, for the NOP (No Operation) instruction, there are no executions as implied by the name.

CMOV :

- For the CMOVxx instruction, execution primarily involves evaluating the condition control using the provided flags (such as Sf, Zf, and Of) based on the Ifun parameter. This determines whether the operation specified by the instruction should be performed.

Instruction		Synonym	Move condition	Description
cmove	S, R	cmovz	ZF	Equal / zero
cmovne	S, R	cmovnz	~ZF	Not equal / not zero
cmovs	S, R		SF	Negative
cmovns	S, R		~SF	Nonnegative
cmovg	S, R	cmovnle	~(SF ^ OF) & ~ZF	Greater (signed >)
cmovge	S, R	cmovnl	~(SF ^ OF)	Greater or equal (signed >=)
cmovl	S, R	cmovnge	SF ^ OF	Less (signed <)
cmovle	S, R	cmovng	(SF ^ OF) | ZF	Less or equal (signed <=)
cmova	S, R	cmovnbe	~CF & ~ZF	Above (unsigned >)
cmovae	S, R	cmovnb	~CF	Above or equal (Unsigned >=)
cmovb	S, R	cmovnae	CF	Below (unsigned <)
cmovbe	S, R	cmovna	CF | ZF	below or equal (unsigned <=)

Based on the condition provided, cnd is set to 1 if the move condition specified in the above instruction is satisfied.

IRMOVQ :

- For the IRMOVQ instruction, the execution involves a single operation: setting valE to the immediate value valC $\implies$ valE = valC

RMMOVQ AND MRMOVQ :

- For both RMMOVQ and MRMOVQ instructions, the execution is similar since they involve moving data between registers and memory. In both cases, the execution calculates the memory address as valE = valC + valB.

EXECUTE FOR OPQ :

- For the OPQ instruction, various operations are performed on valA and valB based on the value of ifun. For example, if ifun is 0000, addition is performed; if ifun represents subtraction, multiplication, or any other operation, the corresponding operation is executed accordingly.

EXECUTE FOR JXX :

- In this stage, the execution primarily involves carrying out conditional jumps based on the flags provided, such as Sf, Zf, and Of, depending on the value of ifun.

Instruction		Synonym	Jump condition	Description
`jmp`	*Label*		1	Direct jump
`jmp`	**Operand*		1	Indirect jump
`je`	*Label*	`jz`	`ZF`	Equal / zero
`jne`	*Label*	`jnz`	`~ZF`	Not equal / not zero
`js`	*Label*		`SF`	Negative
`jns`	*Label*		`~SF`	Nonnegative
`jg`	*Label*	`jnle`	`~(SF ^ OF) & ~ZF`	Greater (signed >)
`jge`	*Label*	`jnl`	`~(SF ^ OF)`	Greater or equal (signed >=)
`jl`	*Label*	`jnge`	`SF ^ OF`	Less (signed <)
`jle`	*Label*	`jng`	`(SF ^ OF) | ZF`	Less or equal (signed <=)
`ja`	*Label*	`jnbe`	`~CF & ~ZF`	Above (unsigned >)
`jae`	*Label*	`jnb`	`~CF`	Above or equal (unsigned >=)
`jb`	*Label*	`jnae`	`CF`	Below (unsigned <)
`jbe`	*Label*	`jna`	`CF | ZF`	Below or equal (unsigned <=)

CALL :

- Execution involves simply decreasing valC by 8 and assigning it to valE.

RETURN :

- Execution involves simply increasing valC by 8 and assigning it to valE.

PUSH :

- Execution requires decrementing valB by 8, as valE is determined by subtracting 8 from valB (valE = -64'd8 +valB).

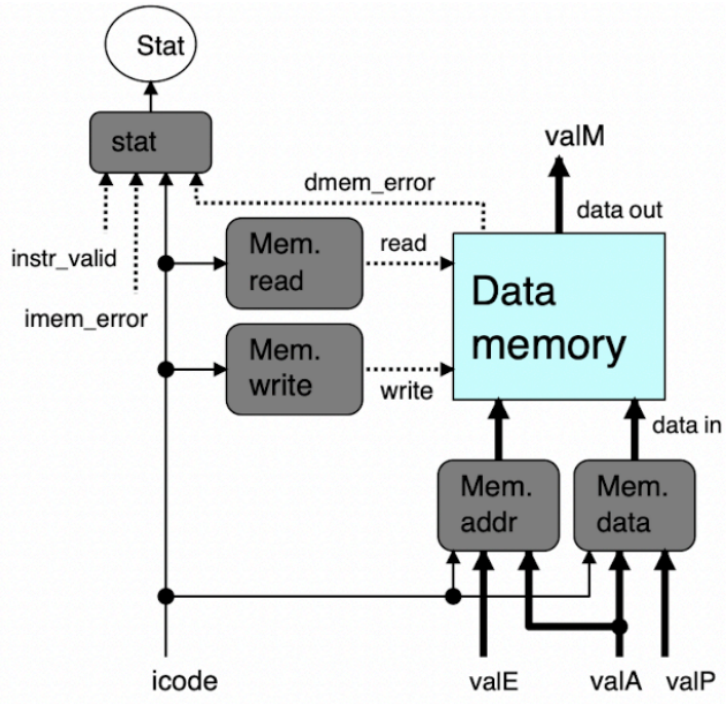
POP :

- Here execution invloves incrementing valB by 8, as valE is determined by adding 8 to valB. (valE = 64'd8 +valB).

. . . . . .

Memory Block :

- The memory module is responsible for storing bits of memory data and can be accessed or utilized whenever necessary. It requires an address to fetch or store data in memory. This module is primarily utilized in instructions such as rmmovq, mrmovq, pop, push, and others, where data needs to be transferred to or from memory.
- We determine the memory address using valE, and based on the instruction, we either write data to or read data from memory. The data memory primarily stores the processor's data.



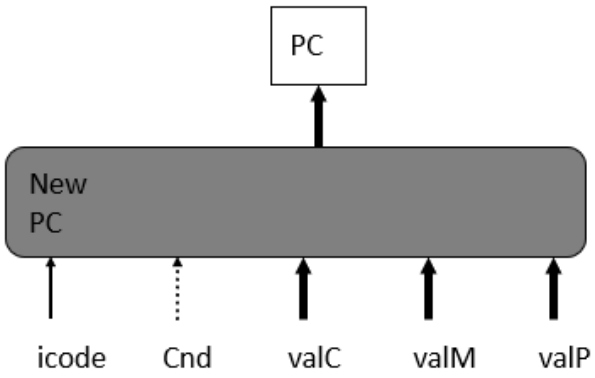
- In this module, the inputs are clk, icode, valA, valE, and valP. The outputs are valM and data_memory_error
- The primary operations performed in this stage are reading from memory or writing to memory, depending on the type of instruction being executed also checking instruction status, Select address, Select data.
- We've implemented a memory module, which is essentially an array of 64-bit registers used for storing, reading, and writing data.If an invalid memory address is provided as input, it will result in the generation of a data_memory_error

- **rmmovq :** M8[valE] ← valA (Writing value to memory)
- **popq :** valM ← M8[valA] (Read from old stack pointer)
- **Conditional moves:** Do nothing.
- **Jumps :** Do nothing.
- **Call :** M8[valE] ← valP (Write incremented PC to new value of stack pointer or Writing return value on stack.)
- **ret :** valM ← M8[valA](Read return address from old stack pointer)

. . . . . .

PC Update Block :

- In this module, the primary task is to update the Program Counter (PC) with the respective valP under normal conditions. However, for instructions like CMOVxx, Jxx, and some others, the PC update might involve using valM or valC instead of the typical valP.
- The inputs to this module are clk, icode, cnd, valC, valM, and valP. The module outputs a signal called new_PC_address, indicating the updated PC value. Firstly we are doing a mem_invalid_check == 0 and then instruction_invalid_check == 0 then updating PC.



	OPq rA, rB	
PC update	PC ← valP	Update PC
	rmmovq rA, D(rB)	
PC update	PC ← valP	Update PC
	popq rA	
PC update	PC ← valP	Update PC
	jXX Dest	
PC update	PC ← Cnd ? valC : valP	Update PC
	call Dest	
PC update	PC ← valC	Set PC to destination
	ret	
PC update	PC ← valM	Set PC to return address

PC updation in different operations

Now explaining each stage of SEQ briefly in pictorial form as mentioned in class :

	OPq rA, rB	
Fetch	icode:ifun ← M_1[PC] rA:rB ← M_1[PC+1] valP ← PC+2	Read instruction byte Read register byte Compute next PC
Decode	valA ← R[rA] valB ← R[rB]	Read operand A Read operand B
Execute	valE ← valB OP valA Set CC	Perform ALU operation Set condition code register
Memory		
Write back	R[rB] ← valE	Write back result
PC update	PC ← valP	Update PC

OPQ Operation

	rmmovq rA, D(rB)	
Fetch	icode:ifun ← M_1[PC] rA:rB ← M_1[PC+1] valC ← M_8[PC+2] valP ← PC+10	Read instruction byte Read register byte Read displacement D Compute next PC
Decode	valA ← R[rA] valB ← R[rB]	Read operand A Read operand B
Execute	valE ← valB + valC	Compute effective address
Memory	M_8[valE] ← valA	Write value to memory
Write back		
PC update	PC ← valP	Update PC

rmmovq Operation

	popq rA	
Fetch	icode:ifun ← M_1[PC] rA:rB ← M_1[PC+1] valP ← PC+2	Read instruction byte Read register byte Compute next PC
Decode	valA ← R[%rsp] valB ← R[%rsp]	Read stack pointer Read stack pointer
Execute	valE ← valB + 8	Increment stack pointer
Memory	valM ← M_8[valA]	Read from stack
Write back	R[%rsp] ← valE R[rA] ← valM	Update stack pointer Write back result
PC update	PC ← valP	Update PC

- Use ALU to increment stack pointer
- Must update two registers
  - Popped value
  - New stack pointer

popq Operation

	cmovXX rA, rB	
Fetch	icode:ifun ← M_1[PC] rA:rB ← M_1[PC+1] valP ← PC+2	Read instruction byte Read register byte Compute next PC
Decode	valA ← R[rA] valB ← 0	Read operand A
Execute	valE ← valB + valA If ! Cond(CC,ifun) rB ← 0xF	Pass valA through ALU (Disable register update)
Memory		
Write back	R[rB] ← valE	Write back result
PC update	PC ← valP	Update PC

- Read register rA and pass through ALU
- Cancel move by setting destination register to 0xF
  - If condition codes & move condition indicate no move

cmovxx Operation

	jXX Dest	
Fetch	icode:ifun ← M_1[PC] valC ← M_8[PC+1] valP ← PC+9	Read instruction byte Read destination address Fall through address
Decode		
Execute	Cnd ← Cond(CC,ifun)	Take branch?
Memory		
Write back		
PC update	PC ← Cnd ? valC : valP	Update PC

- Compute both addresses
- Choose based on setting of condition codes and branch condition

jump Operation

	`call` Dest	
Fetch	icode:ifun ← M_1[PC] valC ← M_8[PC+1] valP ← PC+9	Read instruction byte Read destination address Compute return point
Decode	valB ← R[`%rsp`]	Read stack pointer
Execute	valE ← valB + –8	Decrement stack pointer
Memory	M_8[valE] ← valP	Write return value on stack
Write back	R[`%rsp`] ← valE	Update stack pointer
PC update	PC ← valC	Set PC to destination

- Use ALU to decrement stack pointer
- Store incremented PC

CALL Operation

	ret	
Fetch	icode:ifun ← M_1[PC]	Read instruction byte
Decode	valA ← R[%rsp] valB ← R[%rsp]	Read operand stack pointer Read operand stack pointer
Execute	valE ← valB + 8	Increment stack pointer
Memory	valM ← M_8[valA]	Read return address
Write back	R[%rsp] ← valE	Update stack pointer
PC update	PC ← valM	Set PC to return address

- Use ALU to increment stack pointer
- Read return address from memory

RET Operation

Now showing inputs and outputs for SEQ:

AT LAST i have kept:

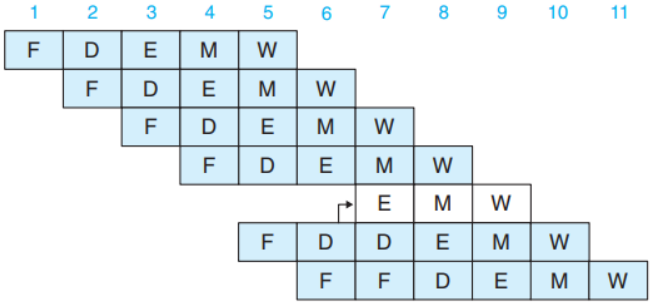
Pipeline Architecture :

- We've developed a 5-stage pipelined architecture for designing a processor compatible with the Y-86 instruction set.
- The Fetch, Decode, Execute, Memory, and Writeback stages are separated by pipeline registers. These registers update at each positive edge of the clock.
- Instructions enter the pipeline starting from the Fetch stage. Once an instruction completes execution, the next instruction in sequence is fed into the pipeline at the positive edge of the next clock cycle e i.e the first instruction which is in fetch stage will go to decode stage and the 2nd instruction will go into fetch stage. This process continues until all instructions have been executed.
- **Data hazards** occur when an instruction that needs a register as a source follows closely after an instruction that writes to the same register. This is a common issue that we want to avoid as it can slow down the pipeline.
- **Control hazards** arise when there's a misprediction of a conditional branch. For instance, our design may predict all branches as taken, leading to the execution of extra instructions unnecessarily. Additionally, getting the return address for the "ret" instruction might require executing three extra instructions in a naive pipeline design.
- Ensuring the pipeline works effectively also involves considering scenarios where multiple special cases happen simultaneously. This requires careful design and testing to handle such situations appropriately.

Avoiding Data Hazards using stalls :

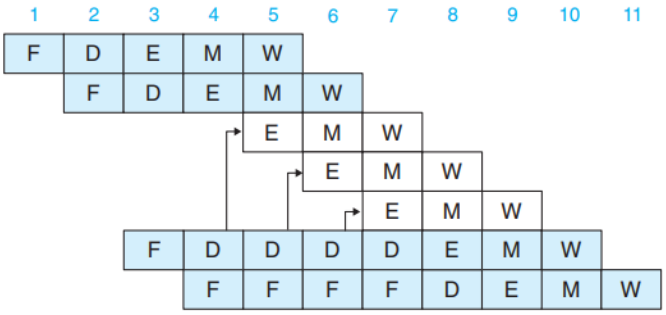
- In below execution, after decoding the addl instruction in cycle 6, the stall control logic detects a data hazard due to the pending write to register stage. It injects a bubble into execute stage and repeats the decoding of the addl instruction in cycle 7. In effect, the machine has dynamically inserted a nop instruction, giving a flow similar to that shown for prog1

```
# prog2
0x000: irmovl $10,%edx
0x006: irmovl  $3,%eax
0x00c: nop
0x00d: nop
       bubble
0x00e: addl %edx,%eax
0x010: halt
```



- In another execution below, after decoding the addl instruction in cycle 4, the stall control logic detects data hazards for both source registers. It injects a bubble into the execute stage and repeats the decoding of the addl instruction on cycle 5. It again detects hazards for both source registers, injects a bubble into the execute stage, and repeats the decoding of the addl instruction on cycle 6. Still, it detects a hazard for source register of the addl instruction on cycle 7. In effect, the machine has dynamically inserted three nop instructions,

```
# prog4
0x000: irmovl $10,%edx
0x006: irmovl  $3,%eax
       bubble
       bubble
       bubble
0x00c: addl %edx,%eax
0x00e: halt
```

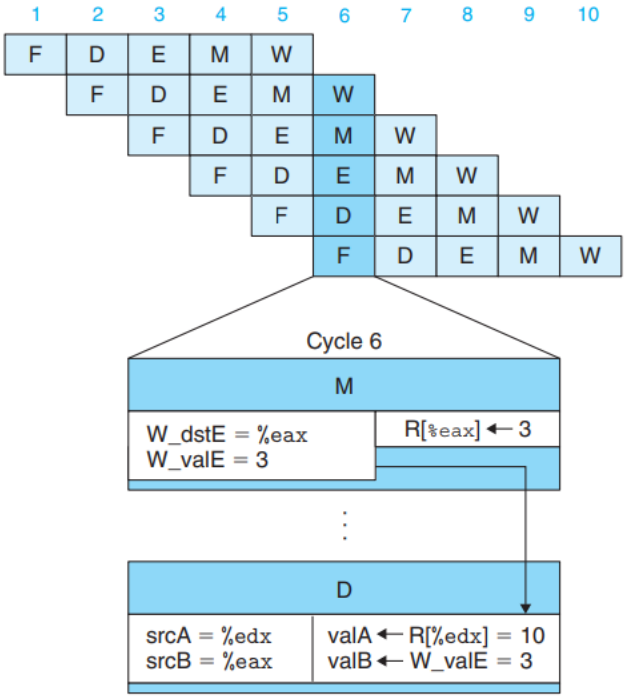


Avoiding Data Hazards by Forwarding :

- The technique of passing a result value directly from one pipeline stage to an earlier one is commonly known as data forwarding (or simply forwarding, and sometimes bypassing).

- In cycle 6, the decode stage logic detects the presence of a pending write to register eax in the write-back stage. It uses this value for source operand valB rather than the value read from the register file.

```
# prog2
0x000: irmovl $10,%edx
0x006: irmovl  $3,%eax
0x00c: nop
0x00d: nop
0x00e: addl %edx,%eax
0x010: halt
```



Pipeline registers :

Fetch register : The Fetch register stores the predicted value of the Program Counter (PC), denoted as F_pred_PC. It takes this predicted PC value and stores it as F_predPC, which serves as an input for the fetch block.

Decode registers : The Decode register stores the values of D_iCode, D_iFun, D_stat, D_rA, D_rB, D_valC, and D_valP. These values correspond to the signals icode, ifun, stat, rA, rB, valC, and valP of the instruction currently present in the decode stage. These values are essential for the functioning of the decode stage.

Execute register : The Execute register stores the values of E_stat, E_icode, E_ifun, E_valC, E_valA, E_valB, E_rA, E_rB, and E_valP. These values correspond to the signals stat, icode, ifun, valC, valA, valB, rA, rB, and valP of the instruction currently present in the execute stage. This register facilitates the passing of all outputs, including the values passed on from the previous stage register, to subsequent stages.

Memory register : The Memory register stores the values of M_stat, M_icode, M_Cnd, M_valE, M_valA, M_rB, M_rA, M_valP, M_valC, and

M_valB. These values represent the signals stat, icode, Cnd, valE, valA, rB, rA, valP, valC, and valB of the instruction currently in the memory stage.

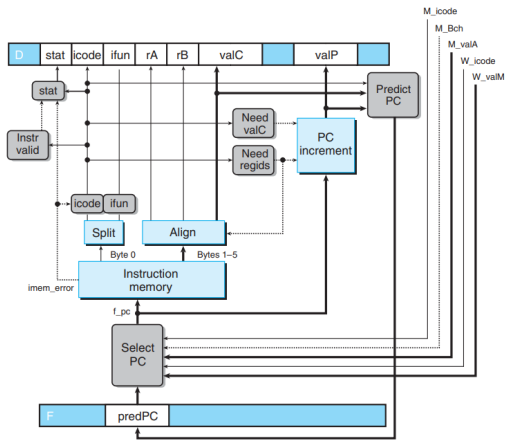
Writeback register : The Writeback register contains the values of W_Cnd, M_valE, M_valM, W_stat, W_iCode, W_valE,W_valM, W_rB, W_rA, W_valC, W_valA, W_valB, and W_valP. These values represent the signals of the instruction currently in the writeback stage.

- The signals stored in these registers are utilized by combinational logic circuits in their corresponding stages (fetch, decode, execute, memory, and writeback). This design ensures that each instruction can be executed independently, without relying on previous stages in the pipeline.

Combinational Logics :

Fetch :
The combinational logic in the Fetch stage is responsible for several tasks:

1. Selecting the current Program Counter (PC).
2. Reading the instruction from memory.
3. Calculating various signals including f_stat, f_icode, f_ifun, rA, rB, f_valC, f_valP, f_instr_Validity, f_imem_error, and f_HF.



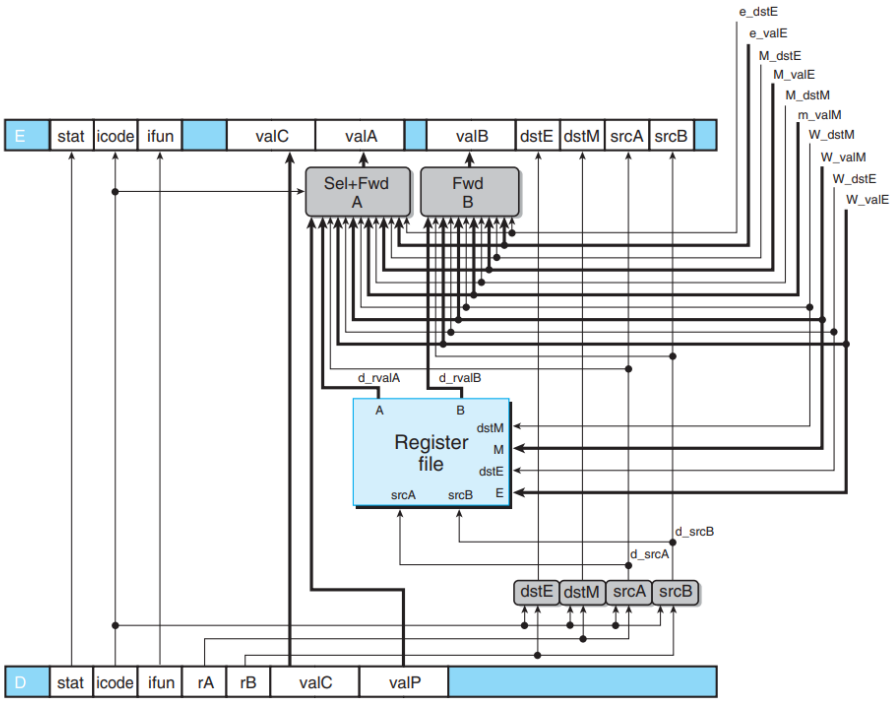
Within the one cycle time limit, the processor can only predict the address of the next instruction

Decode Writeback :

1. In the Decode stage, the logic extracts valA and valB from the inputs rA and rB. Additionally, this stage plays a critical role in data forwarding, crucial for handling data dependencies. It determines whether to obtain data from subsequent pipeline stages or directly from the register file, based on these dependencies.

2. Similarly, in the Writeback stage, the logic updates or writes back the result into the register, whether it's valE or valM.

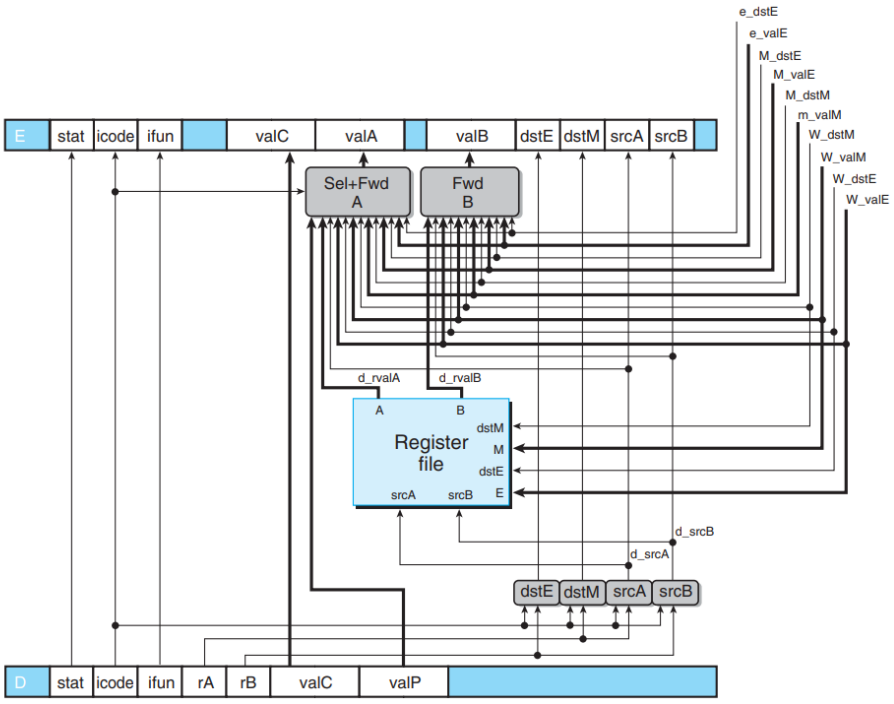
3. The logic blocks labeled "Instr valid," "Need regids," and "Need valC" are the same as for SEQ.



PIPE decode and write-back stage logic. No instruction requires both valP and the value read from register port A, and so these two can be merged to form the signal valA for later stages. The block labeled "Sel+Fwd A" performs this task and also implements the forwarding logic for source operand valA. The block labeled "Fwd B" implements the forwarding logic for source operand valB. The register write locations are specified by the dstE and dstM signals from the write-back stage rather than from the decode stage, since it is writing the results of the instruction currently in the write-back stage.

Execute :

- In the Execute stage, the logic computes e_valE and the condition signal e_Cnd using the Arithmetic Logic Unit (ALU). This block remains largely unchanged from the sequential implementation.



This part of the design is very similar to the logic in the SEQ implementation.

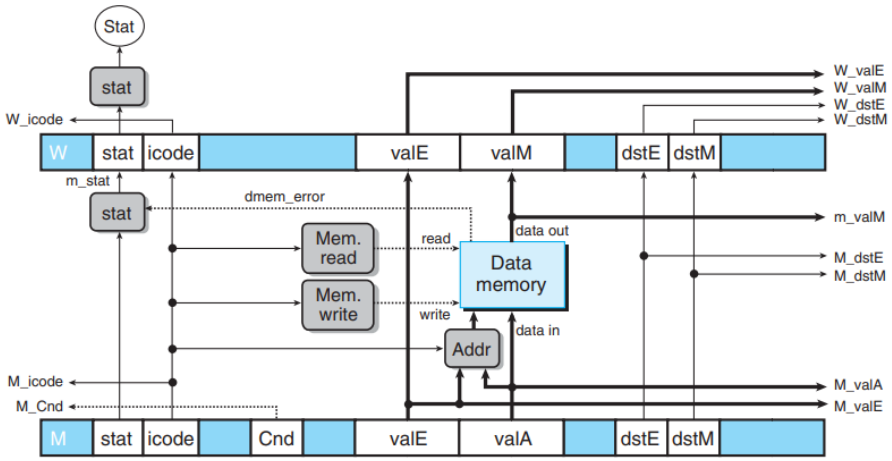
- The logic labeled "Set CC," which determines whether or not update the condition codes, has signals m_stat and W_stat as inputs. These signals are used to detect cases where an instruction causing an exception is passing through later pipeline stages, and therefore any updating of the condition codes should be suppressed.

Memory :

- In the Memory stage, the processor reads from or writes to the data memory. It calculates the signals valM and data_memory_error, obtaining its input from the memory register.
- The block labeled "Data" in SEQ is not present in PIPE. This block served to select between data sources valP (for call instructions) and

valA, but this selection is now performed by the block labeled "Sel+Fwd A" in the decode stage. Most other blocks in this stage are identical to their counterparts in SEQ, with an appropriate renaming of the signals.

- In below figure, we can also see that many of the values in pipeline registers and M and W are supplied to other parts of the circuit as part of the forwarding and pipeline control logic.



Many of the signals from pipeline registers M and W are passed down to earlier stages to provide write-back results, instruction addresses, and forwarded results.

. . . . . .

Pipeline Control Logic:

This logic must handle the following four control cases for which other mechanisms, such as data forwarding and branch prediction, do not suffice:

- **Processing ret:** The pipeline must stall until the ret instruction reaches the write-back stage.
- **Load/use hazards:** The pipeline must stall for one cycle between an instruction that reads a value from memory and an instruction that uses this value.
- **Mispredicted branches:** By the time the branch logic detects that a jump should not have been taken, several instructions at the branch

target will have started down the pipeline. These instructions must be removed from the pipeline.

- **Exceptions:** When an instruction causes an exception, we want to disable the updating of the programmer-visible state by later instructions and halt execution once the excepting instruction reaches the write-back stage.

Detection conditions for pipeline control logic :

- Four different conditions require altering the pipeline flow by either stalling the pipeline or canceling partially executed instructions.

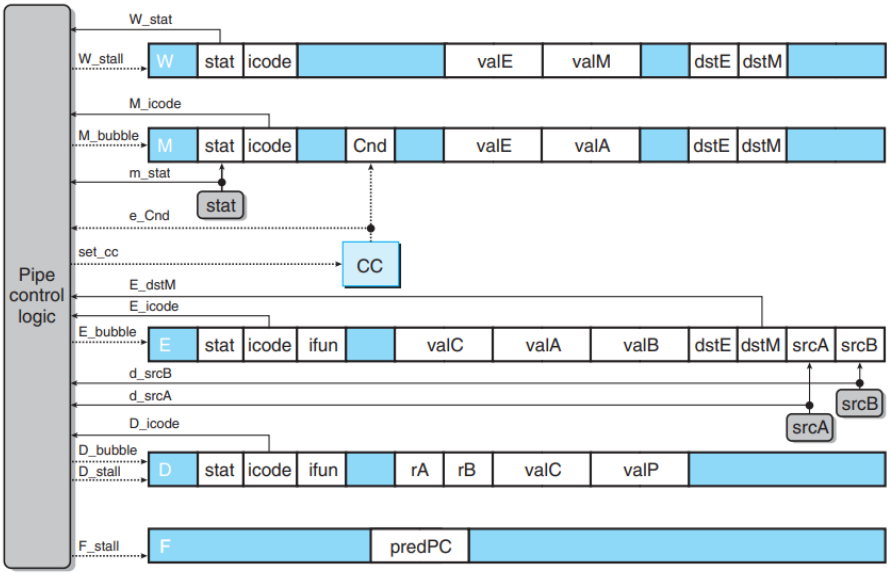
Condition	Trigger
Processing `ret`	IRET ∈ {D_icode, E_icode, M_icode}
Load/use hazard	E_icode ∈ {IMRMOVL, IPOPL} && E_dstM ∈ {d_srcA, d_srcB}
Mispredicted branch	E_icode = IJXX && !e_Cnd
Exception	m_stat ∈ {SADR, SINS, SHLT} || W_stat ∈ {SADR, SINS, SHLT}

Actions for pipeline control logic:

- The different conditions require altering the pipeline flow by either stalling the pipeline or by canceling partially executed instructions.

	Pipeline register				
Condition	F	D	E	M	W
Processing `ret`	stall	bubble	normal	normal	normal
Load/use hazard	stall	stall	bubble	normal	normal
Mispredicted branch	normal	bubble	bubble	normal	normal

- Based on signals from the pipeline registers and pipeline stages, the control logic generates stall and bubble control signals for the pipeline registers, and also determines whether the condition code registers should be updated.



This logic overrides the normal flow of instructions through the pipeline to handle special conditions such as procedure returns, mispredicted branches, load/use hazards, and program exceptions.

Now showing inputs and outputs :

Our input :

```
    instruction_set[0] = 8'b00010000; // 1 0  :  no operation

instruction_set[1] = 8'b01100011; //2 fn  :  OPq
instruction_set[2] = 8'b01100111; //rA rB

instruction_set[3] = 8'b00110000; //3 fn  : irmovq
instruction_set[4] = 8'b01100111; //rA rB
instruction_set[5] = 8'b00000000;
instruction_set[6] = 8'b00000000;
instruction_set[7] = 8'b00000000;
instruction_set[8] = 8'b00000000;
instruction_set[9] = 8'b00000000;
instruction_set[10] = 8'b00000000;
instruction_set[11] = 8'b00000000;
instruction_set[12] = 8'b00000111;
```

```
instruction_set[13] = 8'b01000000; //4 fn : rmmovq
instruction_set[14] = 8'b01100111; //rA rB
instruction_set[15] = 8'b00000000;
instruction_set[16] = 8'b00000000;
instruction_set[17] = 8'b00000000;
instruction_set[18] = 8'b00000000;
instruction_set[19] = 8'b00000000;
instruction_set[20] = 8'b00000000;
instruction_set[21] = 8'b00000000;
instruction_set[22] = 8'b00000011;

instruction_set[23] = 8'b01010000; //5 fn : mromvq
instruction_set[24] = 8'b10000111; //rA rB
instruction_set[25] = 8'b00000000;
instruction_set[26] = 8'b00000000;
instruction_set[27] = 8'b00000000;
instruction_set[28] = 8'b00000000;
instruction_set[29] = 8'b00000000;
instruction_set[30] = 8'b00000000;
instruction_set[31] = 8'b00000000;
instruction_set[32] = 8'b00000011;

instruction_set[33] = 8'b01100000; //6 fn : OPq
instruction_set[34] = 8'b01100111; //rA rB

instruction_set[35] = 8'b01110001; //7 fn : jXX
instruction_set[36] = 8'b00000000;
instruction_set[37] = 8'b00000000;
instruction_set[38] = 8'b00000000;
instruction_set[39] = 8'b00000000;
instruction_set[40] = 8'b00000000;
instruction_set[41] = 8'b00000000;
instruction_set[42] = 8'b00000000;
instruction_set[43] = 8'b00011000;

instruction_set[44] = 8'b10000000; //8 fn : Call
instruction_set[45] = 8'b00000000;
instruction_set[46] = 8'b00000000;
instruction_set[47] = 8'b00000000;
instruction_set[48] = 8'b00000000;
instruction_set[49] = 8'b00000000;
```

```
instruction_set[50] = 8'b00000000;
instruction_set[51] = 8'b00000001;
instruction_set[52] = 8'b10111000;

instruction_set[53] = 8'b00100110; // 2 fn : cmovXX fn = 1
instruction_set[54] = 8'b00101010; //rA rB

instruction_set[55] = 8'b10100000; // 10 fn : PUSHq
instruction_set[56] = 8'b10011111; // rA rB

instruction_set[57] = 8'b10110000; // 11 fn : POPq
instruction_set[58] = 8'b01111111; // rA rB

instruction_set[59] = 8'b10010000; //9 fn : ret

instruction_set[60]  = 8'b00000000; // 1  :  halt
```

Output :

```
[Running] SEQ.v
VCD info: dumpfile seq.vcd opened for output.
TIME INSTANT :                    0
clk : 0
Memory invalid check : x
PC Adress :                    x
Instrction code : xxxx
Instruction invalid Check : x
Function code : xxxx
Need for Val_C  : x
Need for registers : x
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                     x
register A : xxxx
register B : xxxx
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                      x
rcx :                      x
rdx :                      x
rbx :                      x
rsp :                      x
rbp :                      x
```

```
rsi :                    x
rdi :                    x
r8 :                    x
r9 :                    x
r10 :                    x
r11 :                    x
r12 :                    x
r13 :                    x
r14 :                    x
 ------------------------------------
Value_A :                        x
Value_B :                        x
 ------------------------------------
Value_E :                        x
Value_M :                        x
No. of valid instructions :                        0
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                      x
------STATUS FLAGS : -------
ALL OK FLAG  : x
Memory Adress flag : x
Halt Flag : x
Instruction code flag : x

TIME INSTANT :                       1
clk : 0
Memory invalid check : x
PC Adress :                    x
Instrction code : xxxx
Instruction invalid Check : x
Function code : xxxx
Need for Val_C  : x
Need for registers : x
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                    x
```

```
register A : xxxx
register B : xxxx
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                        x
rcx :                         x
rdx :                         x
rbx :                         x
rsp :                         x
rbp :                         x
rsi :                         x
rdi :                         x
r8 :                         x
r9 :                         x
r10 :                         x
r11 :                         x
r12 :                         x
r13 :                         x
r14 :                         x
 ---------------------------------
Value_A :                        x
Value_B :                        x
-------------------------------
Value_E :                        x
Value_M :                        x
No. of valid instructions :                      300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                      x
------STATUS FLAGS : -------
ALL OK FLAG   : x
Memory Adress flag : x
Halt Flag : x
Instruction code flag : x

TIME INSTANT :                     2
```

```
clk : 0
Memory invalid check : x
PC Adress :                    x
Instrction code : xxxx
Instruction invalid Check : x
Function code : xxxx
Need for Val_C  : x
Need for registers : x
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                  x
register A : xxxx
register B : xxxx
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                    1
rcx :                     2
rdx :                     4
rbx :                     8
rsp :                   256
rbp :                    32
rsi :                    64
rdi :                   128
r8 :                    256
r9 :                    512
r10 :                  1024
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
-----------------------------------
Value_A :                  x
Value_B :                  x
-----------------------------------
Value_E :                  x
Value_M :                  x
No. of valid instructions :              300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
```

```
instruction invalid adress : x
status of memory : x
new pc adress :                    x
------STATUS FLAGS : -------
ALL OK FLAG   : x
Memory Adress flag : x
Halt Flag : x
Instruction code flag : x

TIME INSTANT :                     9
clk : 0
Memory invalid check : x
PC Adress :                    0
Instrction code : xxxx
Instruction invalid Check : x
Function code : xxxx
Need for Val_C   : x
Need for registers : x
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                    x
register A : xxxx
register B : xxxx
PROGRAM REGISTER VALUES:
-------------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                      128
r8 :                      256
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
-------------------------------------
Value_A :                    x
```

```
Value_B :                    x
-------------------------------

Value_E :                    x
Value_M :                    x
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  x
------STATUS FLAGS : -------
ALL OK FLAG  : x
Memory Adress flag : x
Halt Flag : x
Instruction code flag : x

TIME INSTANT :                   10
clk : 1
Memory invalid check : 0
PC Adress :                   0
Instrction code : 0001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 0
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                    8
register A : xxxx
register B : xxxx
PROGRAM REGISTER VALUES:
---------------------------------
rax  :                     1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
```

```
rdi :                    128
r8 :                    256
r9 :                    512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
-------------------------------------
Value_A :                       x
Value_B :                       x
-------------------------------
Value_E :                       x
Value_M :                       x
No. of valid instructions :                   300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                 x
------STATUS FLAGS : -------
ALL OK FLAG  : x
Memory Adress flag : x
Halt Flag : x
Instruction code flag : x

TIME INSTANT :                  14
clk : 1
Memory invalid check : 0
PC Adress :                     0
Instrction code : 0001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 0
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                 8
register A : xxxx
```

```
register B : xxxx
PROGRAM REGISTER VALUES:
-------------------------------------
rax   :                    1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                    128
r8 :                     256
r9 :                     512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
 ------------------------------------
Value_A :                  x
Value_B :                  x
-----------------------------------
Value_E :                  x
Value_M :                  x
No. of valid instructions :                300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  x
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                  20
clk : 0
```

```
Memory invalid check : 0
PC Adress :                    0
Instrction code : 0001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 0
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                    8
register A : xxxx
register B : xxxx
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                      1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
rsi :                      64
rdi :                     128
r8 :                      256
r9 :                      512
r10 :                    1024
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
-----------------------------------
Value_A :                    x
Value_B :                    x
-----------------------------------
Value_E :                    x
Value_M :                    x
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
```

```
status of memory : x
new pc adress :                    x
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    23
clk : 0
Memory invalid check : 0
PC Adress :                    0
Instrction code : 0001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C   : 0
Need for registers : 0
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                    8
register A : xxxx
register B : xxxx
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                     1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
rsi :                      64
rdi :                     128
r8 :                      256
r9 :                      512
r10 :                    1024
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
 ----------------------------------
Value_A :                    x
Value_B :                    x
```

```
--------------------------------
Value_E :                    x
Value_M :                    x
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                8
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                  29
clk : 0
Memory invalid check : 0
PC Adress :                    8
Instrction code : 0001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 0
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                    8
register A : xxxx
register B : xxxx
PROGRAM REGISTER VALUES:
--------------------------------
rax  :                     1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                    128
```

```
r8 :                    256
r9 :                    512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
-------------------------------------
Value_A :                       x
Value_B :                       x
-------------------------------------
Value_E :                       x
Value_M :                       x
No. of valid instructions :                     300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                   8
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   30
clk : 1
Memory invalid check : 0
PC Adress :                     8
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0011
Need for Val_C  : 0
Need for registers : 1
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                  24
register A : 0110
register B : 0111
```

```
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                          1
rcx   :                          2
rdx   :                          4
rbx   :                          8
rsp   :                        256
rbp   :                         32
rsi   :                         64
rdi   :                        128
r8    :                       256
r9    :                       512
r10   :                      1024
r11   :                      2048
r12   :                      4096
r13   :                      8192
r14   :                     16384
-----------------------------------
Value_A :                        x
Value_B :                        x
-----------------------------------
Value_E :                        x
Value_M :                        x
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                    8
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    31
clk : 1
Memory invalid check : 0
```

```
PC Adress :                    8
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0011
Need for Val_C  : 0
Need for registers : 1
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                  24
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                    1
rcx :                     2
rdx :                     4
rbx :                     8
rsp :                   256
rbp :                    32
rsi :                    64
rdi :                   128
r8 :                    256
r9 :                    512
r10 :                  1024
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
 ----------------------------------
Value_A :                64
Value_B :               128
-------------------------------
Value_E :                 x
Value_M :                 x
No. of valid instructions :          300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
```

```
new pc adress :                    8
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

------in_xnor1 : 0--------
------in_xnor2 : 0--------
------in_xor11 : 0--------
------in_xor12 : 0--------
------in_and21 : x--------
------in_and22 : x--------
------out_and2 : x --------
  ----------SIGNED FLAG = 0---------------
----------OVERFLOW FLAG = x---------------
----------ZERO FLAG = 0-----------------
TIME INSTANT :                    38
clk : 1
Memory invalid check : 0
PC Adress :                    8
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0011
Need for Val_C   : 0
Need for registers : 1
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                   24
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                    1
rcx :                    2
rdx :                    4
rbx :                    8
rsp :                  256
rbp :                   32
rsi :                   64
rdi :                  128
r8 :                  256
```

```
r9 :                    512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
-----------------------------------
Value_A :                 64
Value_B :                128
-----------------------------------
Value_E :                192
Value_M :                  x
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  8
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                  40
clk : 0
Memory invalid check : 0
PC Adress :                   8
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0011
Need for Val_C  : 0
Need for registers : 1
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                 24
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
```

```
-------------------------------------
rax   :                          1
rcx :                            2
rdx :                            4
rbx :                            8
rsp :                          256
rbp :                           32
rsi :                           64
rdi :                          128
r8 :                           256
r9 :                           512
r10 :                         1024
r11 :                         2048
r12 :                         4096
r13 :                         8192
r14 :                        16384
-------------------------------------
Value_A :                       64
Value_B :                      128
-------------------------------------
Value_E :                      192
Value_M :                        x
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  8
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                  42
clk : 0
Memory invalid check : 0
PC Adress :                      8
```

```
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0011
Need for Val_C  : 0
Need for registers : 1
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                    24
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                      1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
rsi :                      64
rdi :                     192
r8 :                      256
r9 :                      512
r10 :                    1024
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
-----------------------------------
Value_A :                   64
Value_B :                  128
-----------------------------------
Value_E :                  192
Value_M :                    x
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  8
```

```
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    43
clk : 0
Memory invalid check : 0
PC Adress :                    8
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0011
Need for Val_C   : 0
Need for registers : 1
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                   24
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-------------------------------------
rax   :                    1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                    192
r8 :                     256
r9 :                     512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
 ------------------------------------
Value_A :                  64
Value_B :                 128
-------------------------------
Value_E :                 192
```

```
Value_M :                          x
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                     24
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                      49
clk : 0
Memory invalid check : 0
PC Adress :                       24
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0011
Need for Val_C   : 0
Need for registers : 1
Val_C =  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Val P =                        24
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
------------------------------------
rax   :                          1
rcx :                            2
rdx :                            4
rbx :                            8
rsp :                          256
rbp :                           32
rsi :                           64
rdi :                          192
r8 :                           256
r9 :                           512
```

```
r10 :                    1024
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
-------------------------------------

Value_A :                  64
Value_B :                 128
-------------------------------

Value_E :                 192
Value_M :                   x
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                 24
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                     50
clk : 1
Memory invalid check : 0
PC Adress :                   24
Instrction code : 0011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000000000
Val P =                  104
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-------------------------------------
```

```
rax   :                    1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                    192
r8 :                     256
r9 :                     512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
-------------------------------------
Value_A :                  x
Value_B :                  x
-------------------------------------
Value_E :                  x
Value_M :                  x
No. of valid instructions :                300
Condition satisfy check : 0
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  24
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   53
clk : 1
Memory invalid check : 0
PC Adress :                    24
Instrction code : 0011
```

```
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                    104
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                       1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                      192
r8 :                      256
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
 ----------------------------------
Value_A :                        x
Value_B :                        x
-----------------------------------
Value_E :                        7
Value_M :                        x
No. of valid instructions :                    300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  24
------STATUS FLAGS : -------
```

```
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    60
clk : 0
Memory invalid check : 0
PC Adress :                    24
Instrction code : 0011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C   : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                   104
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                     1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
rsi :                      64
rdi :                     192
r8 :                      256
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
-----------------------------------
Value_A :                      x
Value_B :                      x
-----------------------------------
Value_E :                      7
Value_M :                      x
```

```
No. of valid instructions :                    300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  24
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   62
clk : 0
Memory invalid check : 0
PC Adress :                    24
Instrction code : 0011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C   : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                   104
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                     1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
rsi :                      64
rdi :                       7
r8 :                     256
r9 :                     512
r10 :                   1024
```

```
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
-------------------------------------
Value_A :                        x
Value_B :                        x
-------------------------------
Value_E :                        7
Value_M :                        x
No. of valid instructions :                    300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                   24
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                       63
clk : 0
Memory invalid check : 0
PC Adress :                      24
Instrction code : 0011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                     104
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-------------------------------------
rax  :                       1
```

```
rcx :                    2
rdx :                    4
rbx :                    8
rsp :                  256
rbp :                   32
rsi :                   64
rdi :                    7
r8 :                   256
r9 :                   512
r10 :                 1024
r11 :                 2048
r12 :                 4096
r13 :                 8192
r14 :                16384
-------------------------------------
Value_A :                x
Value_B :                x
-------------------------------------
Value_E :                7
Value_M :                x
No. of valid instructions :                  300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :              104
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :               69
clk : 0
Memory invalid check : 0
PC Adress :                  104
Instrction code : 0011
Instruction invalid Check : 0
```

```
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                  104
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-------------------------------------
rax  :                    1
rcx :                     2
rdx :                     4
rbx :                     8
rsp :                   256
rbp :                    32
rsi :                    64
rdi :                     7
r8 :                    256
r9 :                    512
r10 :                  1024
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
 ------------------------------------
Value_A :                  x
Value_B :                  x
-----------------------------
Value_E :                  7
Value_M :                  x
No. of valid instructions :              300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                104
------STATUS FLAGS : -------
ALL OK FLAG  : 1
```

```
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    70
clk : 1
Memory invalid check : 0
PC Adress :                  104
Instrction code : 0100
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                  184
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                    1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                      7
r8 :                     256
r9 :                     512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
-----------------------------------
Value_A :                  x
Value_B :                  x
-----------------------------------
Value_E :                  x
Value_M :                  x
No. of valid instructions :              300
```

```
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                    104
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                     71
clk : 1
Memory invalid check : 0
PC Adress :                  104
Instrction code : 0100
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                    184
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                      1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
rsi :                      64
rdi :                       7
r8 :                     256
r9 :                     512
r10 :                   1024
r11 :                   2048
```

```
r12 :                    4096
r13 :                    8192
r14 :                   16384
-----------------------------------
Value_A :                   64
Value_B :                    7
-----------------------------------
Value_E :                    x
Value_M :                    x
No. of valid instructions :                  300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                 104
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                  78
clk : 1
Memory invalid check : 0
PC Adress :                  104
Instrction code : 0100
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                  184
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                     1
rcx :                     2
```

```
rdx :                    4
rbx :                    8
rsp :                  256
rbp :                   32
rsi :                   64
rdi :                    7
r8 :                   256
r9 :                   512
r10 :                 1024
r11 :                 2048
r12 :                 4096
r13 :                 8192
r14 :                16384
-------------------------------------
Value_A :                64
Value_B :                 7
-------------------------------
Value_E :                10
Value_M :                 x
No. of valid instructions :                300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                104
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

---------- Memory data value =                64----------
------ Memory Status : Data pushed from register to memory -------
TIME INSTANT :                79
clk : 1
Memory invalid check : 0
PC Adress :                  104
Instrction code : 0100
```

```
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000000000000
Val P =                    184
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-------------------------------------
rax  :                       1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                        7
r8 :                       256
r9 :                       512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
 -------------------------------------
Value_A :                    64
Value_B :                     7
-------------------------------------
Value_E :                    10
Value_M :                     x
No. of valid instructions :                  300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                  104
------STATUS FLAGS : -------
```

```
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    80
clk : 0
Memory invalid check : 0
PC Adress :                  104
Instrction code : 0100
Instruction invalid Check : 0
Function code : 0000
Need for Val_C   : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                   184
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                     1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
rsi :                      64
rdi :                       7
r8 :                     256
r9 :                     512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
-----------------------------------
Value_A :                  64
Value_B :                   7
-----------------------------------
Value_E :                  10
Value_M :                   x
```

```
No. of valid instructions :                    300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                 104
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                  83
clk : 0
Memory invalid check : 0
PC Adress :                  104
Instrction code : 0100
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                   184
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                     1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                      7
r8 :                     256
r9 :                     512
r10 :                    1024
```

```
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
-------------------------------------
Value_A :                  64
Value_B :                   7
-------------------------------
Value_E :                  10
Value_M :                   x
No. of valid instructions :                  300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                 184
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                  89
clk : 0
Memory invalid check : 0
PC Adress :                   184
Instrction code : 0100
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                  184
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-------------------------------------
rax  :                     1
```

```
rcx :                    2
rdx :                    4
rbx :                    8
rsp :                  256
rbp :                   32
rsi :                   64
rdi :                    7
r8 :                   256
r9 :                   512
r10 :                 1024
r11 :                 2048
r12 :                 4096
r13 :                 8192
r14 :                16384
-------------------------------------
Value_A :                64
Value_B :                 7
-------------------------------------
Value_E :                10
Value_M :                 x
No. of valid instructions :                    300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                184
-------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                 90
clk : 1
Memory invalid check : 0
PC Adress :                184
Instrction code : 0101
Instruction invalid Check : 0
```

```
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                  264
register A : 1000
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                    1
rcx :                     2
rdx :                     4
rbx :                     8
rsp :                   256
rbp :                    32
rsi :                    64
rdi :                     7
r8 :                    256
r9 :                    512
r10 :                  1024
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
-----------------------------------
Value_A :                 x
Value_B :                 x
-----------------------------------
Value_E :                 x
Value_M :                 x
No. of valid instructions :              300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :              184
------STATUS FLAGS : -------
ALL OK FLAG  : 1
```

```
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    91
clk : 1
Memory invalid check : 0
PC Adress :                   184
Instrction code : 0101
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                   264
register A : 1000
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                     1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
rsi :                      64
rdi :                       7
r8 :                     256
r9 :                     512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
-----------------------------------
Value_A :                     x
Value_B :                     7
-----------------------------------
Value_E :                     x
Value_M :                     x
No. of valid instructions :                 300
```

```
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                    184
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                     98
clk : 1
Memory invalid check : 0
PC Adress :                   184
Instrction code : 0101
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                    264
register A : 1000
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                      1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
rsi :                      64
rdi :                       7
r8 :                    256
r9 :                    512
r10 :                   1024
r11 :                   2048
```

```
r12 :                    4096
r13 :                    8192
r14 :                   16384
-----------------------------------
Value_A :                    x
Value_B :                    7
-----------------------------------
Value_E :                   10
Value_M :                    x
No. of valid instructions :                  300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                184
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

--------- Memory data value for mrmovq =
64---------
Valid data taken from memory
TIME INSTANT :                  99
clk : 1
Memory invalid check : 0
PC Adress :                 184
Instrction code : 0101
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                  264
register A : 1000
register B : 0111
PROGRAM REGISTER VALUES:
```

```
-------------------------------------
rax   :                          1
rcx :                            2
rdx :                            4
rbx :                            8
rsp :                          256
rbp :                           32
rsi :                           64
rdi :                            7
r8 :                           256
r9 :                           512
r10 :                         1024
r11 :                         2048
r12 :                         4096
r13 :                         8192
r14 :                        16384
-------------------------------------
Value_A :                        x
Value_B :                        7
-------------------------------------
Value_E :                       10
Value_M :                       64
No. of valid instructions :                    300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : 0
instruction invalid adress : 0
status of memory : 1
new pc adress :                  184
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   100
clk : 0
Memory invalid check : 0
PC Adress :                      184
```

```
Instrction code : 0101
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                  264
register A : 1000
register B : 0111
```
PROGRAM REGISTER VALUES:

```
rax  :                    1
rcx :                     2
rdx :                     4
rbx :                     8
rsp :                   256
rbp :                    32
rsi :                    64
rdi :                     7
r8 :                    256
r9 :                    512
r10 :                  1024
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
```

```
Value_A :                  x
Value_B :                  7
```

```
Value_E :                 10
Value_M :                 64
No. of valid instructions :                300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : 0
instruction invalid adress : 0
status of memory : 1
new pc adress :                184
```

```
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    102
clk : 0
Memory invalid check : 0
PC Adress :                    184
Instrction code : 0101
Instruction invalid Check : 0
Function code : 0000
Need for Val_C   : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                     264
register A : 1000
register B : 0111
PROGRAM REGISTER VALUES:
-------------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                        7
r8 :                        64
r9 :                       512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
 -------------------------------------
Value_A :                    x
Value_B :                    7
-------------------------------------
Value_E :                   10
```

```
Value_M :                    64
No. of valid instructions :                  300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : 0
instruction invalid adress : 0
status of memory : 1
new pc adress :                  184
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   103
clk : 0
Memory invalid check : 0
PC Adress :                  184
Instrction code : 0101
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                    264
register A : 1000
register B : 0111
PROGRAM REGISTER VALUES:
---------------------------------
rax  :                      1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                      7
r8 :                     64
r9 :                    512
```

```
r10 :                    1024
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
_____________________________________

Value_A :                    x
Value_B :                    7
_____________________________

Value_E :                   10
Value_M :                   64
No. of valid instructions :                  300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : 0
instruction invalid adress : 0
status of memory : 1
new pc adress :                 264
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                  109
clk : 0
Memory invalid check : 0
PC Adress :                  264
Instrction code : 0101
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000000000
Val P =                  264
register A : 1000
register B : 0111
PROGRAM REGISTER VALUES:
_____________________________________
```

```
rax   :                    1
rcx   :                    2
rdx   :                    4
rbx   :                    8
rsp   :                  256
rbp   :                   32
rsi   :                   64
rdi   :                    7
r8    :                  64
r9    :                 512
r10   :                1024
r11   :                2048
r12   :                4096
r13   :                8192
r14   :               16384
```

```
Value_A :                  x
Value_B :                  7
```

```
Value_E :                 10
Value_M :                 64
No. of valid instructions :                300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : 0
instruction invalid adress : 0
status of memory : 1
new pc adress :                  264
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   110
clk : 1
Memory invalid check : 0
PC Adress :                  264
Instrction code : 0110
```

```
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                    280
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-------------------------------
rax  :                      1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
rsi :                      64
rdi :                       7
r8 :                      64
r9 :                     512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
 -------------------------------
Value_A :                    x
Value_B :                    x
-------------------------------
Value_E :                    x
Value_M :                   64
No. of valid instructions :                  300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                 264
------STATUS FLAGS : -------
```

```
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    111
clk : 1
Memory invalid check : 0
PC Adress :                     264
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                     280
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                       1
rcx :                         2
rdx :                         4
rbx :                         8
rsp :                       256
rbp :                        32
rsi :                        64
rdi :                         7
r8 :                        64
r9 :                       512
r10 :                      1024
r11 :                      2048
r12 :                      4096
r13 :                      8192
r14 :                     16384
 ----------------------------------
Value_A :                     64
Value_B :                      7
-------------------------------
Value_E :                      x
Value_M :                     64
```

```
No. of valid instructions :                    300
Condition satisfy check : x
signed flag : 0
Overflow flag : x
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  264
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

------in_xnor1 : 0--------
------in_xnor2 : 0--------
------in_xor11 : 0--------
------in_xor12 : 0--------
------in_and21 : 1--------
------in_and22 : 0--------
------out_and2 : 0 -------
  ----------SIGNED FLAG = 0---------------
----------OVERFLOW FLAG = 0----------------
----------ZERO FLAG = 0-----------------
TIME INSTANT :                   118
clk : 1
Memory invalid check : 0
PC Adress :                  264
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                  280
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                    1
```

```
rcx :                    2
rdx :                    4
rbx :                    8
rsp :                  256
rbp :                   32
rsi :                   64
rdi :                    7
r8 :                   64
r9 :                  512
r10 :                 1024
r11 :                 2048
r12 :                 4096
r13 :                 8192
r14 :                16384
-------------------------------------
Value_A :                64
Value_B :                 7
-------------------------------------
Value_E :                71
Value_M :                64
No. of valid instructions :                    300
Condition satisfy check : x
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  264
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   120
clk : 0
Memory invalid check : 0
PC Adress :                  264
Instrction code : 0110
Instruction invalid Check : 0
```

```
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                  280
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                    1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                      7
r8 :                      64
r9 :                     512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
 ----------------------------------
Value_A :                  64
Value_B :                   7
-----------------------------
Value_E :                  71
Value_M :                  64
No. of valid instructions :                300
Condition satisfy check : x
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  264
------STATUS FLAGS : -------
ALL OK FLAG  : 1
```

```
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    122
clk : 0
Memory invalid check : 0
PC Adress :                     264
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                    280
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                       1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
-----------------------------------
Value_A :                    64
Value_B :                     7
-----------------------------------
Value_E :                    71
Value_M :                    64
No. of valid instructions :                  300
```

```
Condition satisfy check : x
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                    264
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    123
clk : 0
Memory invalid check : 0
PC Adress :                    264
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                    280
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                      1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                     71
r8 :                    64
r9 :                   512
r10 :                  1024
r11 :                  2048
```

```
r12 :                    4096
r13 :                    8192
r14 :                   16384
-------------------------------------
Value_A :                  64
Value_B :                   7
-------------------------------
Value_E :                  71
Value_M :                  64
No. of valid instructions :                    300
Condition satisfy check : x
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                 280
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                  129
clk : 0
Memory invalid check : 0
PC Adress :                  280
Instrction code : 0110
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000000000
Val P =                  280
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-------------------------------------
rax  :                     1
rcx :                    2
```

```
rdx :                    4
rbx :                    8
rsp :                  256
rbp :                   32
rsi :                   64
rdi :                   71
r8 :                   64
r9 :                  512
r10 :                 1024
r11 :                 2048
r12 :                 4096
r13 :                 8192
r14 :                16384
```

```
Value_A :                64
Value_B :                 7
```

```
Value_E :                71
Value_M :                64
No. of valid instructions :                    300
Condition satisfy check : x
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  280
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   130
clk : 1
Memory invalid check : 0
PC Adress :                   280
Instrction code : 0111
Instruction invalid Check : 0
Function code : 0001
```

```
Need for Val_C  : 1
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000011(
Val P =                    352
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
 ----------------------------------
Value_A :                      x
Value_B :                      x
-----------------------------------
Value_E :                      x
Value_M :                     64
No. of valid instructions :                    300
Condition satisfy check : x
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  280
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
```

```
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    133
clk : 1
Memory invalid check : 0
PC Adress :                    280
Instrction code : 0111
Instruction invalid Check : 0
Function code : 0001
Need for Val_C  : 1
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000011
Val P =                    352
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                      1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
rsi :                      64
rdi :                      71
r8 :                      64
r9 :                     512
r10 :                    1024
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
 ----------------------------------
Value_A :                    x
Value_B :                    x
-----------------------------------
Value_E :                    x
Value_M :                   64
No. of valid instructions :                  300
Condition satisfy check : 0
```

```
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  280
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   140
clk : 0
Memory invalid check : 0
PC Adress :                    280
Instrction code : 0111
Instruction invalid Check : 0
Function code : 0001
Need for Val_C  : 1
Need for registers : 0
Val_C =  0000000000000000000000000000000000000000000000000000000000000110
Val P =                    352
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                       1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
```

```
r13 :                    8192
r14 :                   16384
-------------------------------------
Value_A :                    x
Value_B :                    x
-------------------------------------
Value_E :                    x
Value_M :                   64
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                280
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                  143
clk : 0
Memory invalid check : 0
PC Adress :                  280
Instrction code : 0111
Instruction invalid Check : 0
Function code : 0001
Need for Val_C  : 1
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000011(
Val P =                  352
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-------------------------------------
rax  :                     1
rcx :                     2
rdx :                     4
```

```
rbx :                    8
rsp :                  256
rbp :                   32
rsi :                   64
rdi :                   71
r8 :                   64
r9 :                  512
r10 :                 1024
r11 :                 2048
r12 :                 4096
r13 :                 8192
r14 :                16384
-------------------------------------
Value_A :                 x
Value_B :                 x
-------------------------------------
Value_E :                 x
Value_M :                64
No. of valid instructions :                300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  352
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   149
clk : 0
Memory invalid check : 0
PC Adress :                  352
Instrction code : 0111
Instruction invalid Check : 0
Function code : 0001
Need for Val_C  : 1
```

```
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000011(
Val P =                    352
register A : 0110
register B : 0111
```
PROGRAM REGISTER VALUES:
```
------------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
 ----------------------------------
Value_A :                    x
Value_B :                    x
-----------------------------------
Value_E :                    x
Value_M :                   64
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                352
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
```

```
Instruction code flag : 0

TIME INSTANT :                    150
clk : 1
Memory invalid check : 0
PC Adress :                    352
Instrction code : 1000
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    424
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                        64
r9 :                       512
r10 :                      1024
r11 :                      2048
r12 :                      4096
r13 :                      8192
r14 :                     16384
-----------------------------------
Value_A :                      x
Value_B :                      x
-----------------------------
Value_E :                      x
Value_M :                     64
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
```

```
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                 352
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                  151
clk : 1
Memory invalid check : 0
PC Adress :                   352
Instrction code : 1000
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                   424
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------

rax  :                     1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                     71
r8 :                     64
r9 :                    512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
```

```
r14 :                    16384
_____________________________________
Value_A :                          x
Value_B :                        256
_____________________________________
Value_E :                          x
Value_M :                         64
No. of valid instructions :                        300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                      352
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                       158
clk : 1
Memory invalid check : 0
PC Adress :                          352
Instrction code : 1000
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000001101110
Val P =                        424
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
_____________________________________
rax  :                           1
rcx :                            2
rdx :                            4
rbx :                            8
```

```
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                     71
r8 :                     64
r9 :                    512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
-------------------------------------
Value_A :                   x
Value_B :                 256
-------------------------------------
Value_E :                 192
Value_M :                  64
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                352
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

------ Memory Status : Adress of next unstruction pushed to memory ------
TIME INSTANT :                 159
clk : 1
Memory invalid check : 0
PC Adress :                352
Instrction code : 1000
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
```

```
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    424
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
 ----------------------------------
Value_A :                     x
Value_B :                   256
-----------------------------------
Value_E :                   192
Value_M :                    64
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                352
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
```

```
Instruction code flag : 0

TIME INSTANT :                    160
clk : 0
Memory invalid check : 0
PC Adress :                    352
Instrction code : 1000
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    424
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
-----------------------------------
Value_A :                      x
Value_B :                    256
-----------------------------------
Value_E :                    192
Value_M :                     64
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
```

```
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                  352
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   162
clk : 0
Memory invalid check : 0
PC Adress :                    352
Instrction code : 1000
Instruction invalid Check : 0
Function code : 0000
Need for Val_C   : 1
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    424
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------

rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      192
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
```

```
r14 :                    16384
_____________________________________
Value_A :                          x
Value_B :                        256
_____________________________________
Value_E :                        192
Value_M :                         64
No. of valid instructions :                        300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                      352
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                       163
clk : 0
Memory invalid check : 0
PC Adress :                      352
Instrction code : 1000
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000001101110
Val P =                    424
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
_____________________________________
rax  :                           1
rcx :                            2
rdx :                            4
rbx :                            8
```

```
rsp :                    192
rbp :                     32
rsi :                     64
rdi :                     71
r8 :                     64
r9 :                    512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
-------------------------------------
Value_A :                   x
Value_B :                 256
-------------------------------------
Value_E :                 192
Value_M :                  64
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                  440
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   169
clk : 0
Memory invalid check : 0
PC Adress :                    440
Instrction code : 1000
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 1
Need for registers : 0
```

```
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    424
register A : 0110
register B : 0111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      192
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                        64
r9 :                       512
r10 :                      1024
r11 :                      2048
r12 :                      4096
r13 :                      8192
r14 :                     16384
 ----------------------------------
Value_A :                     x
Value_B :                   256
-----------------------------------
Value_E :                   192
Value_M :                    64
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                  440
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0
```

```
TIME INSTANT :                    170
clk : 1
Memory invalid check : 0
PC Adress :                   440
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                   456
register A : 1001
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                     1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     192
rbp :                      32
rsi :                      64
rdi :                      71
r8 :                      64
r9 :                     512
r10 :                    1024
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
-----------------------------------
Value_A :                     x
Value_B :                     x
-------------------------------
Value_E :                     x
Value_M :                    64
No. of valid instructions :                   300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
```

```
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                    440
------STATUS FLAGS : ------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    171
clk : 1
Memory invalid check : 0
PC Adress :                    440
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000001101110
Val P =                    456
register A : 1001
register B : 1111
PROGRAM REGISTER VALUES:
--------------------------------
rax  :                      1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    192
rbp :                     32
rsi :                     64
rdi :                     71
r8 :                    64
r9 :                   512
r10 :                  1024
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
```

```
--------------------------------------
Value_A :                    512
Value_B :                    192
--------------------------------------
Value_E :                      x
Value_M :                     64
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  440
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   178
clk : 1
Memory invalid check : 0
PC Adress :                    440
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000011011110
Val P =                  456
register A : 1001
register B : 1111
PROGRAM REGISTER VALUES:
--------------------------------------
rax  :                     1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    192
```

```
rbp :                    32
rsi :                    64
rdi :                    71
r8 :                    64
r9 :                   512
r10 :                  1024
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
-------------------------------------
Value_A :                512
Value_B :                192
-------------------------------------
Value_E :                128
Value_M :                 64
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                440
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

------- Memory Status : Adress of pushed data instruction pushed to memo
TIME INSTANT :                 179
clk : 1
Memory invalid check : 0
PC Adress :                  440
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
```

```
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    456
register A : 1001
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      192
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                        64
r9 :                       512
r10 :                      1024
r11 :                      2048
r12 :                      4096
r13 :                      8192
r14 :                     16384
-----------------------------------
Value_A :                    512
Value_B :                    192
-----------------------------------
Value_E :                    128
Value_M :                     64
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                  440
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0
```

```
TIME INSTANT :                    180
clk : 0
Memory invalid check : 0
PC Adress :                    440
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    456
register A : 1001
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                      1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     192
rbp :                      32
rsi :                      64
rdi :                      71
r8 :                      64
r9 :                     512
r10 :                    1024
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
-----------------------------------
Value_A :                  512
Value_B :                  192
-----------------------------------
Value_E :                  128
Value_M :                   64
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
```

```
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                    440
------STATUS FLAGS : ------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    182
clk : 0
Memory invalid check : 0
PC Adress :                    440
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000110111
Val P =                    456
register A : 1001
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                      1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    128
rbp :                     32
rsi :                     64
rdi :                     71
r8 :                    64
r9 :                   512
r10 :                  1024
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
```

```
-----------------------------------
Value_A :                    512
Value_B :                    192
-----------------------------------
Value_E :                    128
Value_M :                     64
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                  440
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   183
clk : 0
Memory invalid check : 0
PC Adress :                    440
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    456
register A : 1001
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                       1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      128
```

```
rbp :                    32
rsi :                    64
rdi :                    71
r8 :                    64
r9 :                   512
r10 :                  1024
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
-------------------------------------
Value_A :                512
Value_B :                192
-------------------------------------
Value_E :                128
Value_M :                 64
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                456
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                 189
clk : 0
Memory invalid check : 0
PC Adress :                  456
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
```

```
Val P =                    456
register A : 1001
register B : 1111
PROGRAM REGISTER VALUES:
-------------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      128
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
-------------------------------------
Value_A :                   512
Value_B :                   192
-------------------------------------
Value_E :                   128
Value_M :                    64
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : 1
new pc adress :                  456
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0
```

```
TIME INSTANT :                    190
clk : 1
Memory invalid check : 0
PC Adress :                   456
Instrction code : 1011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      128
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
-----------------------------------
Value_A :                     x
Value_B :                     x
-----------------------------------
Value_E :                     x
Value_M :                    64
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
```

```
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  456
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   191
clk : 1
Memory invalid check : 0
PC Adress :                    456
Instrction code : 1011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C   : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                      472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
-------------------------------------
rax   :                        1
rcx :                          2
rdx :                          4
rbx :                          8
rsp :                        128
rbp :                         32
rsi :                         64
rdi :                         71
r8 :                         64
r9 :                        512
r10 :                       1024
r11 :                       2048
r12 :                       4096
r13 :                       8192
r14 :                      16384
 -------------------------------------
```

```
Value_A :                    128
Value_B :                    128
-------------------------------
Value_E :                      x
Value_M :                     64
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  456
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   198
clk : 1
Memory invalid check : 0
PC Adress :                    456
Instrction code : 1011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000110111
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
--------------------------------
rax  :                       1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      128
rbp :                       32
```

```
rsi :                    64
rdi :                    71
r8 :                    64
r9 :                   512
r10 :                  1024
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
 ---------------------------------
Value_A :                  128
Value_B :                  128
---------------------------------
Value_E :                  192
Value_M :                   64
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  456
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

-------- Memory Status : Data extracted from the memory --------
------Valid data taken from memory------
TIME INSTANT :                  199
clk : 1
Memory invalid check : 0
PC Adress :                  456
Instrction code : 1011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
```

```
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                     1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     128
rbp :                      32
rsi :                      64
rdi :                      71
r8 :                      64
r9 :                     512
r10 :                    1024
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
 ----------------------------------
Value_A :                  128
Value_B :                  128
-----------------------------------
Value_E :                  192
Value_M :                  512
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : 0
status of memory : 1
new pc adress :                456
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0
```

```
TIME INSTANT :                    200
clk : 0
Memory invalid check : 0
PC Adress :                    456
Instrction code : 1011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000001101110
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      128
rbp :                       32
rsi :                       64
rdi :                       71
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
-----------------------------------
Value_A :                   128
Value_B :                   128
-----------------------------------
Value_E :                   192
Value_M :                   512
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
```

```
Zero Flag : 0
address memory error : 0
instruction invalid adress : 0
status of memory : 1
new pc adress :                    456
------STATUS FLAGS : ------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                    202
clk : 0
Memory invalid check : 0
PC Adress :                    456
Instrction code : 1011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000001101110
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
--------------------------------
rax  :                      1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    192
rbp :                     32
rsi :                     64
rdi :                    512
r8 :                     64
r9 :                    512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
```

```
--------------------------------------
Value_A :                    128
Value_B :                    128
--------------------------------------
Value_E :                    192
Value_M :                    512
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : 0
status of memory : 1
new pc adress :                456
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   203
clk : 0
Memory invalid check : 0
PC Adress :                  456
Instrction code : 1011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000110111
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
--------------------------------------
rax  :                      1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    192
```

```
rbp :                    32
rsi :                    64
rdi :                   512
r8 :                     64
r9 :                    512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
-------------------------------------
Value_A :                128
Value_B :                128
-------------------------------------
Value_E :                192
Value_M :                512
No. of valid instructions :                300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : 0
status of memory : 1
new pc adress :                472
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                 209
clk : 0
Memory invalid check : 0
PC Adress :                472
Instrction code : 1011
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
```

```
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
-------------------------------------
rax   :                     1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     192
rbp :                      32
rsi :                      64
rdi :                     512
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
-------------------------------------
Value_A :                   128
Value_B :                   128
-------------------------------------
Value_E :                   192
Value_M :                   512
No. of valid instructions :                 300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : 0
status of memory : 1
new pc adress :                472
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0
```

```
TIME INSTANT :                    210
clk : 1
Memory invalid check : 0
PC Adress :                   472
Instrction code : 1001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                     472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                       1
rcx :                         2
rdx :                         4
rbx :                         8
rsp :                       192
rbp :                        32
rsi :                        64
rdi :                       512
r8 :                         64
r9 :                        512
r10 :                       1024
r11 :                       2048
r12 :                       4096
r13 :                       8192
r14 :                      16384
-----------------------------------
Value_A :                     x
Value_B :                     x
-----------------------------------
Value_E :                     x
Value_M :                   512
No. of valid instructions :                   300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
```

```
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  472
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   211
clk : 1
Memory invalid check : 0
PC Adress :                    472
Instrction code : 1001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C   : 0
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000001101110
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
-------------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      192
rbp :                       32
rsi :                       64
rdi :                      512
r8 :                        64
r9 :                       512
r10 :                      1024
r11 :                      2048
r12 :                      4096
r13 :                      8192
r14 :                     16384
 -------------------------------------
```

```
Value_A :                    192
Value_B :                    192
-------------------------------
Value_E :                      x
Value_M :                    512
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  472
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   218
clk : 1
Memory invalid check : 0
PC Adress :                    472
Instrction code : 1001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 0
Val_C =  0000000000000000000000000000000000000000000000000000000000110111
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
--------------------------------
rax  :                       1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      192
rbp :                       32
```

```
rsi :                    64
rdi :                   512
r8 :                     64
r9 :                    512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
 ---------------------------------
Value_A :                192
Value_B :                192
---------------------------------
Value_E :                256
Value_M :                512
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                472
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

------- Memory Status : Data extracted from the memory --------
TIME INSTANT :                 219
clk : 1
Memory invalid check : 0
PC Adress :                  472
Instrction code : 1001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
```

```
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
-------------------------------------
rax   :                     1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     192
rbp :                      32
rsi :                      64
rdi :                     512
r8 :                       64
r9 :                      512
r10 :                     1024
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
-------------------------------------
Value_A :                   192
Value_B :                   192
-------------------------------------
Value_E :                   256
Value_M :                   424
No. of valid instructions :                 300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : x
new pc adress :                472
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0
```

```
TIME INSTANT :                    220
clk : 0
Memory invalid check : 0
PC Adress :                  472
Instrction code : 1001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      192
rbp :                       32
rsi :                       64
rdi :                      512
r8 :                        64
r9 :                       512
r10 :                      1024
r11 :                      2048
r12 :                      4096
r13 :                      8192
r14 :                     16384
-----------------------------------
Value_A :                    192
Value_B :                    192
-----------------------------------
Value_E :                    256
Value_M :                    424
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
```

```
address memory error : 0
instruction invalid adress : x
status of memory : x
new pc adress :                  472
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                     222
clk : 0
Memory invalid check : 0
PC Adress :                    472
Instrction code : 1001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C   : 0
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000001101111
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
-------------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                      512
r8 :                        64
r9 :                       512
r10 :                      1024
r11 :                      2048
r12 :                      4096
r13 :                      8192
r14 :                     16384
 -------------------------------------
```

```
Value_A :                    192
Value_B :                    192
-------------------------------
Value_E :                    256
Value_M :                    424
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : x
new pc adress :                  472
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   223
clk : 0
Memory invalid check : 0
PC Adress :                  472
Instrction code : 1001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 0
Val_C =  0000000000000000000000000000000000000000000000000000000000110111
Val P =                    472
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
--------------------------------
rax  :                      1
rcx :                       2
rdx :                       4
rbx :                       8
rsp :                     256
rbp :                      32
```

```
rsi :                    64
rdi :                   512
r8 :                     64
r9 :                    512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
 ---------------------------------
Value_A :                192
Value_B :                192
---------------------------------
Value_E :                256
Value_M :                424
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : x
new pc adress :                424
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                 229
clk : 0
Memory invalid check : 0
PC Adress :                424
Instrction code : 1001
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 0
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                 472
```

```
register A : 0111
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                          1
rcx :                            2
rdx :                            4
rbx :                            8
rsp :                          256
rbp :                           32
rsi :                           64
rdi :                          512
r8 :                            64
r9 :                           512
r10 :                          1024
r11 :                          2048
r12 :                          4096
r13 :                          8192
r14 :                         16384
 ----------------------------------
Value_A :                        192
Value_B :                        192
-------------------------------
Value_E :                        256
Value_M :                        424
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : x
new pc adress :                  424
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   230
```

```
clk : 1
Memory invalid check : 0
PC Adress :                    424
Instrction code : 0010
Instruction invalid Check : 0
Function code : 0110
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    440
register A : 0010
register B : 1010
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                       1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                      512
r8 :                        64
r9 :                       512
r10 :                      1024
r11 :                      2048
r12 :                      4096
r13 :                      8192
r14 :                     16384
-----------------------------------
Value_A :                      x
Value_B :                      x
-----------------------------------
Value_E :                      x
Value_M :                    424
No. of valid instructions :                    300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
```

```
instruction invalid adress : x
status of memory : x
new pc adress :                  424
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   231
clk : 1
Memory invalid check : 0
PC Adress :                  424
Instrction code : 0010
Instruction invalid Check : 0
Function code : 0110
Need for Val_C   : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                  440
register A : 0010
register B : 1010
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                    1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                    512
r8 :                      64
r9 :                     512
r10 :                    1024
r11 :                    2048
r12 :                    4096
r13 :                    8192
r14 :                   16384
-----------------------------------
Value_A :                    4
```

```
Value_B :                    0
--------------------------------
Value_E :                    x
Value_M :                  424
No. of valid instructions :                  300
Condition satisfy check : 0
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                424
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                 238
clk : 1
Memory invalid check : 0
PC Adress :                424
Instrction code : 0010
Instruction invalid Check : 0
Function code : 0110
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                  440
register A : 0010
register B : 1010
PROGRAM REGISTER VALUES:
--------------------------------
rax  :                     1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
```

```
rdi :                    512
r8 :                     64
r9 :                    512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
-------------------------------------
Value_A :                   4
Value_B :                   0
-------------------------------
Value_E :                   4
Value_M :                 424
No. of valid instructions :                  300
Condition satisfy check : 1
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                424
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                 240
clk : 0
Memory invalid check : 0
PC Adress :                424
Instrction code : 0010
Instruction invalid Check : 0
Function code : 0110
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000011011110
Val P =                  440
register A : 0010
```

```
register B : 1010
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                    1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
rdi :                    512
r8 :                      64
r9 :                     512
r10 :                   1024
r11 :                   2048
r12 :                   4096
r13 :                   8192
r14 :                  16384
 ----------------------------------
Value_A :                  4
Value_B :                  0
-----------------------------------
Value_E :                  4
Value_M :                424
No. of valid instructions :                300
Condition satisfy check : 1
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                424
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                 242
clk : 0
```

```
Memory invalid check : 0
PC Adress :                  424
Instrction code : 0010
Instruction invalid Check : 0
Function code : 0110
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                  440
register A : 0010
register B : 1010
PROGRAM REGISTER VALUES:
-----------------------------------
rax  :                    1
rcx :                     2
rdx :                     4
rbx :                     8
rsp :                   256
rbp :                    32
rsi :                    64
rdi :                   512
r8 :                     64
r9 :                    512
r10 :                     4
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
-----------------------------------
Value_A :                  4
Value_B :                  0
-----------------------------------
Value_E :                  4
Value_M :                424
No. of valid instructions :                300
Condition satisfy check : 1
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
```

```
status of memory : x
new pc adress :                    424
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                     243
clk : 0
Memory invalid check : 0
PC Adress :                   424
Instrction code : 0010
Instruction invalid Check : 0
Function code : 0110
Need for Val_C   : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    440
register A : 0010
register B : 1010
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                      512
r8 :                        64
r9 :                       512
r10 :                        4
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
 ----------------------------------
Value_A :                        4
Value_B :                        0
```

```
--------------------------------
Value_E :                    4
Value_M :                  424
No. of valid instructions :                  300
Condition satisfy check : 1
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  440
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   249
clk : 0
Memory invalid check : 0
PC Adress :                  440
Instrction code : 0010
Instruction invalid Check : 0
Function code : 0110
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000000110111
Val P =                  440
register A : 0010
register B : 1010
PROGRAM REGISTER VALUES:
--------------------------------
rax  :                    1
rcx :                    2
rdx :                    4
rbx :                    8
rsp :                  256
rbp :                   32
rsi :                   64
rdi :                  512
```

```
r8 :                    64
r9 :                   512
r10 :                    4
r11 :                 2048
r12 :                 4096
r13 :                 8192
r14 :                16384
-----------------------------------
Value_A :                 4
Value_B :                 0
-----------------------------------
Value_E :                 4
Value_M :               424
No. of valid instructions :                  300
Condition satisfy check : 1
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                440
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                 250
clk : 1
Memory invalid check : 0
PC Adress :                440
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                 456
register A : 1001
register B : 1111
```

PROGRAM REGISTER VALUES:

```
rax   :                    1
rcx   :                    2
rdx   :                    4
rbx   :                    8
rsp   :                  256
rbp   :                   32
rsi   :                   64
rdi   :                  512
r8    :                   64
r9    :                  512
r10   :                    4
r11   :                 2048
r12   :                 4096
r13   :                 8192
r14   :                16384
```

```
Value_A :                  x
Value_B :                  x
```

```
Value_E :                  x
Value_M :                424
No. of valid instructions :                300
Condition satisfy check : 1
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  440
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                   251
clk : 1
Memory invalid check : 0
```

```
PC Adress :                    440
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    456
register A : 1001
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                      512
r8 :                        64
r9 :                       512
r10 :                        4
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
 ----------------------------------
Value_A :                   512
Value_B :                   256
-------------------------------
Value_E :                     x
Value_M :                   424
No. of valid instructions :                    300
Condition satisfy check : 1
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
```

```
new pc adress :                    440
------STATUS FLAGS : -------
ALL OK FLAG   : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

TIME INSTANT :                     258
clk : 1
Memory invalid check : 0
PC Adress :                    440
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C   : 0
Need for registers : 1
Val_C =  00000000000000000000000000000000000000000000000000000000000110111
Val P =                    456
register A : 1001
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------
rax   :                      1
rcx :                        2
rdx :                        4
rbx :                        8
rsp :                      256
rbp :                       32
rsi :                       64
rdi :                      512
r8 :                        64
r9 :                       512
r10 :                        4
r11 :                     2048
r12 :                     4096
r13 :                     8192
r14 :                    16384
-----------------------------------
Value_A :                    512
Value_B :                    256
-----------------------------------
```

```
Value_E :                    192
Value_M :                    424
No. of valid instructions :                    300
Condition satisfy check : 1
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : x
instruction invalid adress : x
status of memory : x
new pc adress :                  440
------STATUS FLAGS : ------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0

SEQ.v:163: $finish called at 259 (1s)
------ Memory Status : Adress of pushed data instruction pushed to memo
TIME INSTANT :                 259
clk : 1
Memory invalid check : 0
PC Adress :                  440
Instrction code : 1010
Instruction invalid Check : 0
Function code : 0000
Need for Val_C  : 0
Need for registers : 1
Val_C =  0000000000000000000000000000000000000000000000000000000110111
Val P =                   456
register A : 1001
register B : 1111
PROGRAM REGISTER VALUES:
-----------------------------------

rax  :                     1
rcx :                      2
rdx :                      4
rbx :                      8
rsp :                    256
rbp :                     32
rsi :                     64
```

```
rdi :                    512
r8 :                     64
r9 :                    512
r10 :                     4
r11 :                  2048
r12 :                  4096
r13 :                  8192
r14 :                 16384
-------------------------------------
Value_A :                512
Value_B :                256
-------------------------------------
Value_E :                192
Value_M :                424
No. of valid instructions :                  300
Condition satisfy check : 1
signed flag : 0
Overflow flag : 0
Zero Flag : 0
address memory error : 0
instruction invalid adress : x
status of memory : x
new pc adress :                440
------STATUS FLAGS : -------
ALL OK FLAG  : 1
Memory Adress flag : 0
Halt Flag : 0
Instruction code flag : 0
```