İSTANBUL TEKNİK ÜNİVERSİTESİ

1773

ERDEM ÇİL

040120157

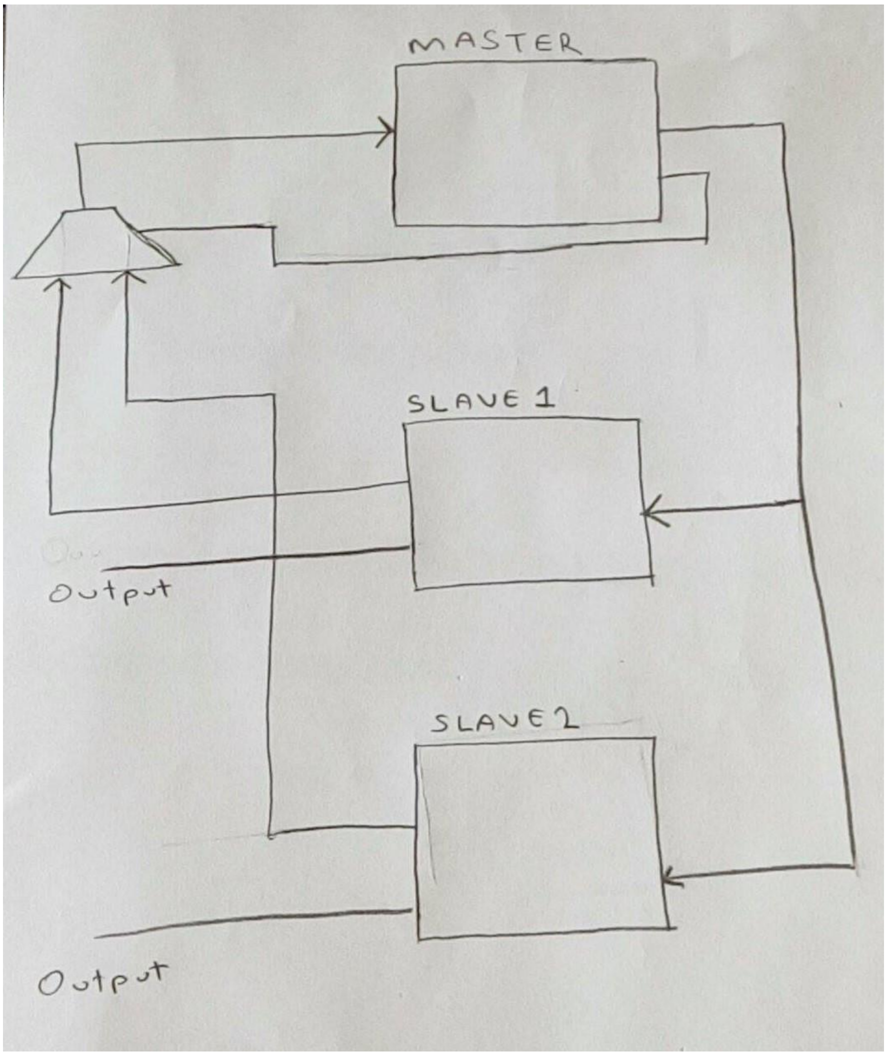
EHB 432E

MICROPROCESSOR SYSTEM DESIGN

PROJECT 2

PROJECT NUMBER: 12

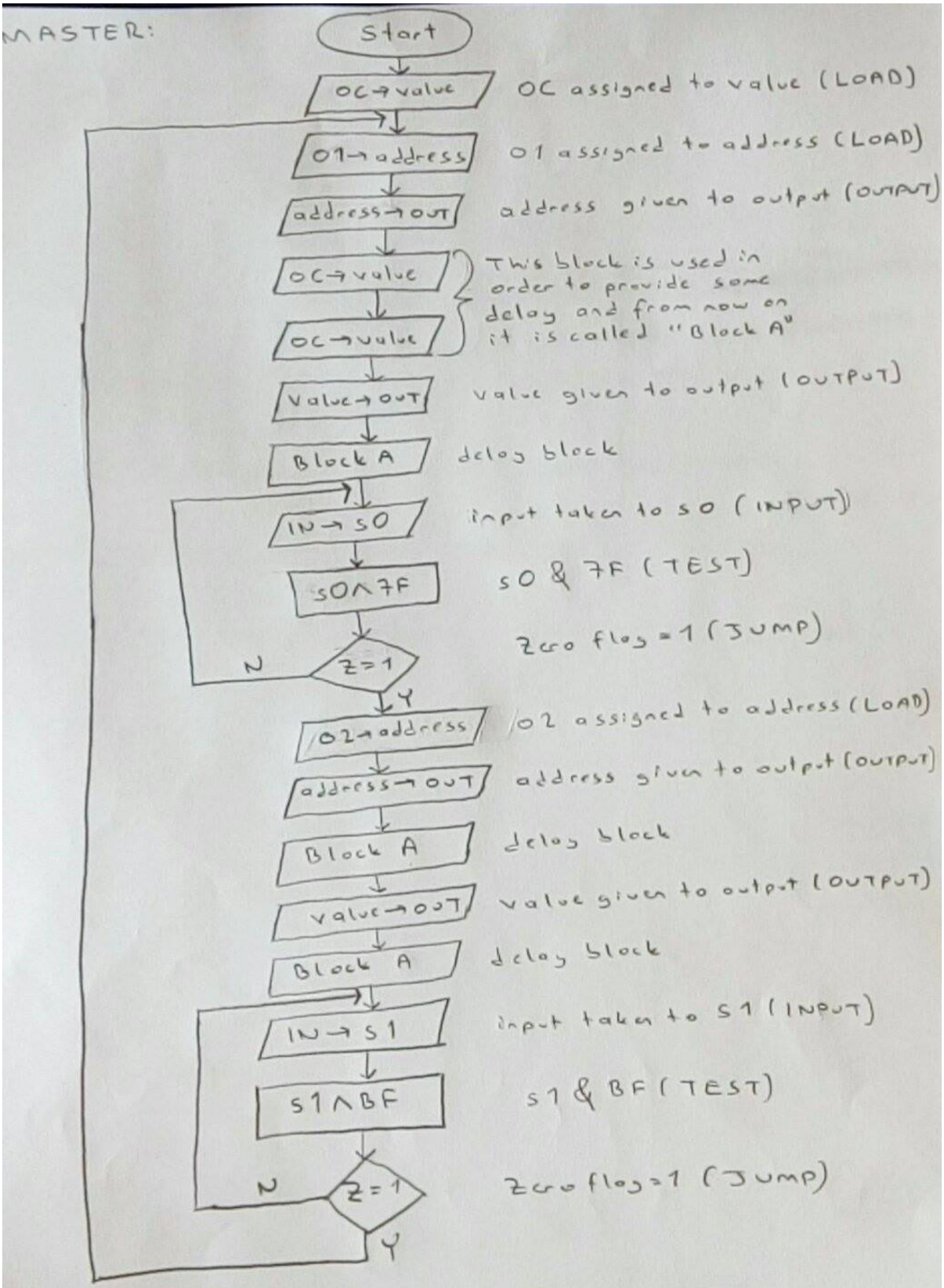
In this project, a system including one master picoblaze and two slave picoblazes has been implemented using ISE.



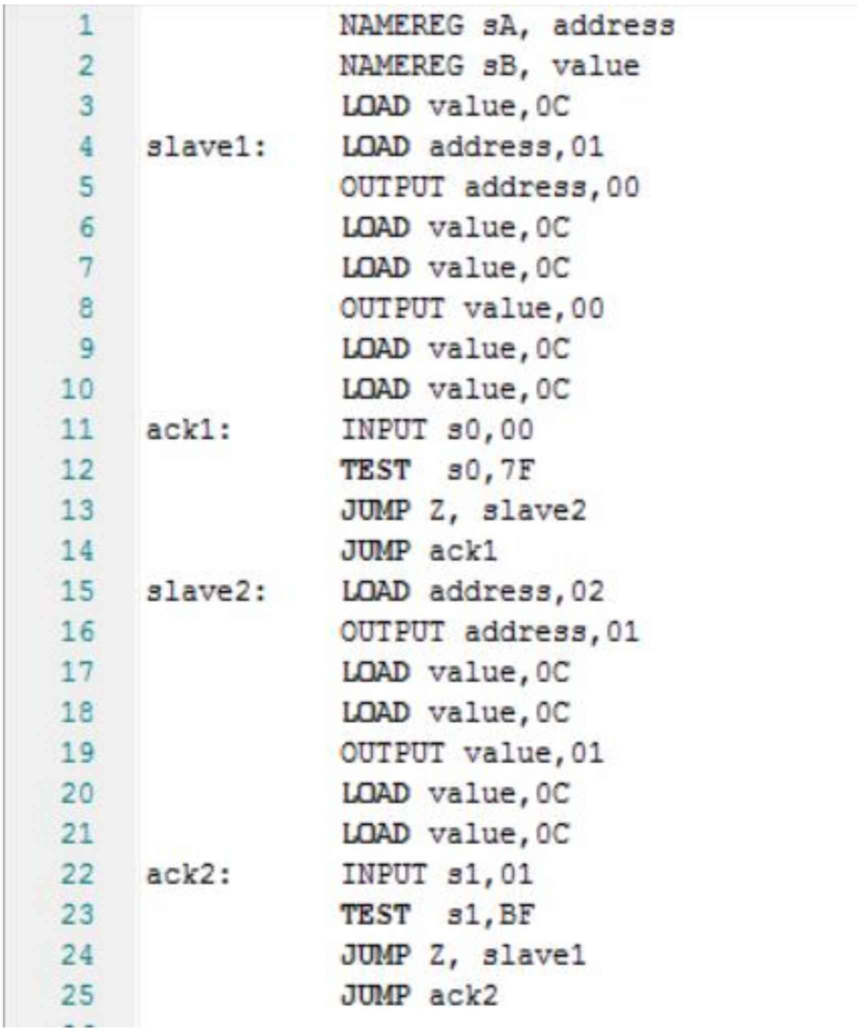
Block Diagram of the System

In the system, master picoblaze has “0Ch” as stored data. It sends first the address information then this stored data to slave picoblazes. If the address information is “01h”, then slave1 picoblaze assigns the data sent from master picoblaze to its output and sends an acknowledgement signal to master picoblaze in order that it understands that the data has been read and accordingly the address information must be changed. If the address information is “02h”, then slave2 picoblaze works similarly with slave1 picoblaze. The selection between the acknowledgement signals of two slave picoblazes is done using a multiplexer. The select input of this multiplexer is connected to the least significant bit of port_id output of the master picoblaze.

1. Master Picoblaze:


MASTER:
Start
0C→value
0C assigned to value (LOAD)
01→address
01 assigned to address (LOAD)
address→OUT
address given to output (OUTPUT)
0C→value
0C→value
This block is used in order to provide some delay and from now on it is called "Block A"
Value→OUT
value given to output (OUTPUT)
Block A
delay block
IN→s0
input taken to s0 (INPUT)
s0∧7F
s0 & 7F (TEST)
Z=1
Zero flag=1 (JUMP)
N
Y
02→address
02 assigned to address (LOAD)
address→OUT
address given to output (OUTPUT)
Block A
delay block
value→OUT
value given to output (OUTPUT)
Block A
delay block
IN→s1
input taken to s1 (INPUT)
s1∧BF
s1 & BF (TEST)
Z=1
Zero flag=1 (JUMP)
N
Y

Flowchart of Master Picoblaze

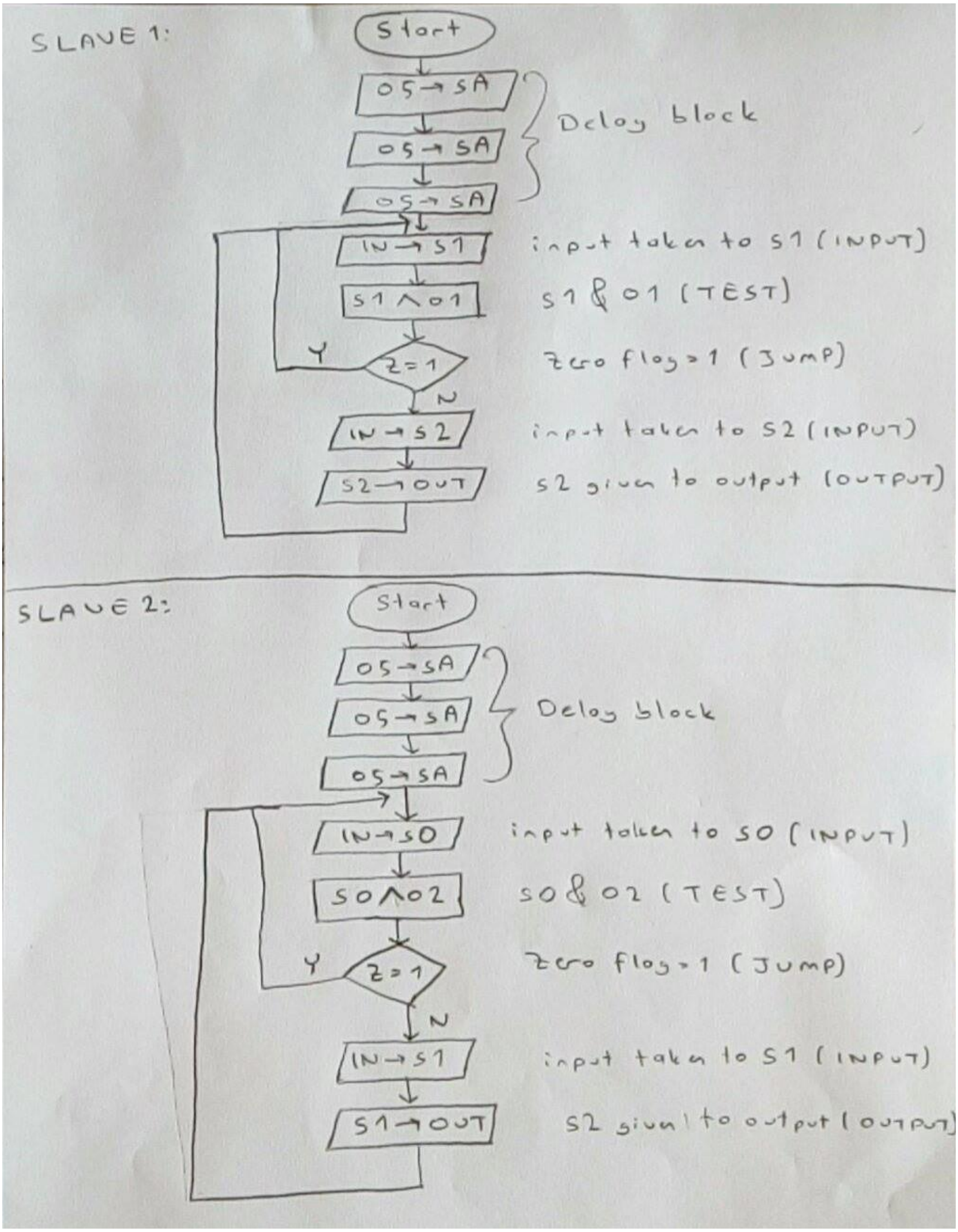


```
             NAMEREG sA, address
             NAMEREG sB, value
             LOAD value,0C
slave1:      LOAD address,01
             OUTPUT address,00
             LOAD value,0C
             LOAD value,0C
             OUTPUT value,00
             LOAD value,0C
             LOAD value,0C
ack1:        INPUT s0,00
             TEST  s0,7F
             JUMP Z, slave2
             JUMP ack1
slave2:      LOAD address,02
             OUTPUT address,01
             LOAD value,0C
             LOAD value,0C
             OUTPUT value,01
             LOAD value,0C
             LOAD value,0C
ack2:        INPUT s1,01
             TEST  s1,BF
             JUMP Z, slave1
             JUMP ack2
```

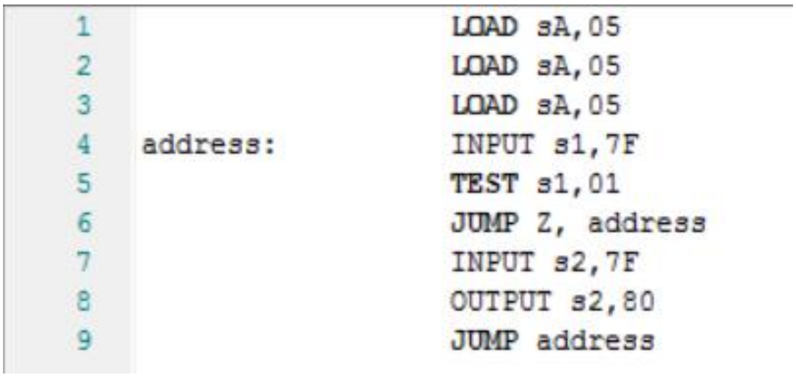
Assembly Code of Master Picoblaze

In the assembly code of master picoblaze, “LOAD value, 0C” command is used several times in order to provide some delay. This delay is necessary for the slave picoblazes. During these delay blocks, slave picoblazes take the address information and tests whether it is their address information. In the ack1 cycle, the input data sent by slave1 picoblaze is tested with the value “7Fh”. The acknowledgement signal sent by slave1 picoblaze is “80h” and the result of the test is only zero when this value is sent. In the ack2 cycle, “7Fh” is replaced with “BFh”, since the acknowledgement signal sent by slave2 picoblaze is “40h”.

2. Slave Picoblazes:

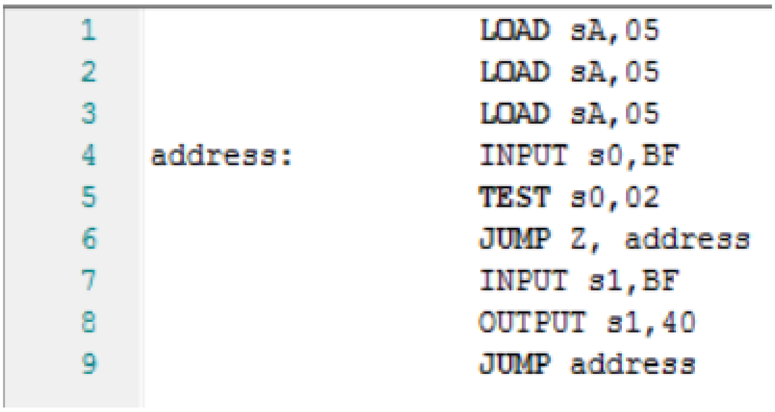


Flowcharts of Slave Picoblazes



```
                    LOAD sA,05
                    LOAD sA,05
                    LOAD sA,05
address:            INPUT s1,7F
                    TEST s1,01
                    JUMP Z, address
                    INPUT s2,7F
                    OUTPUT s2,80
                    JUMP address
```

Assembly Code of Slave1 Picoblaze

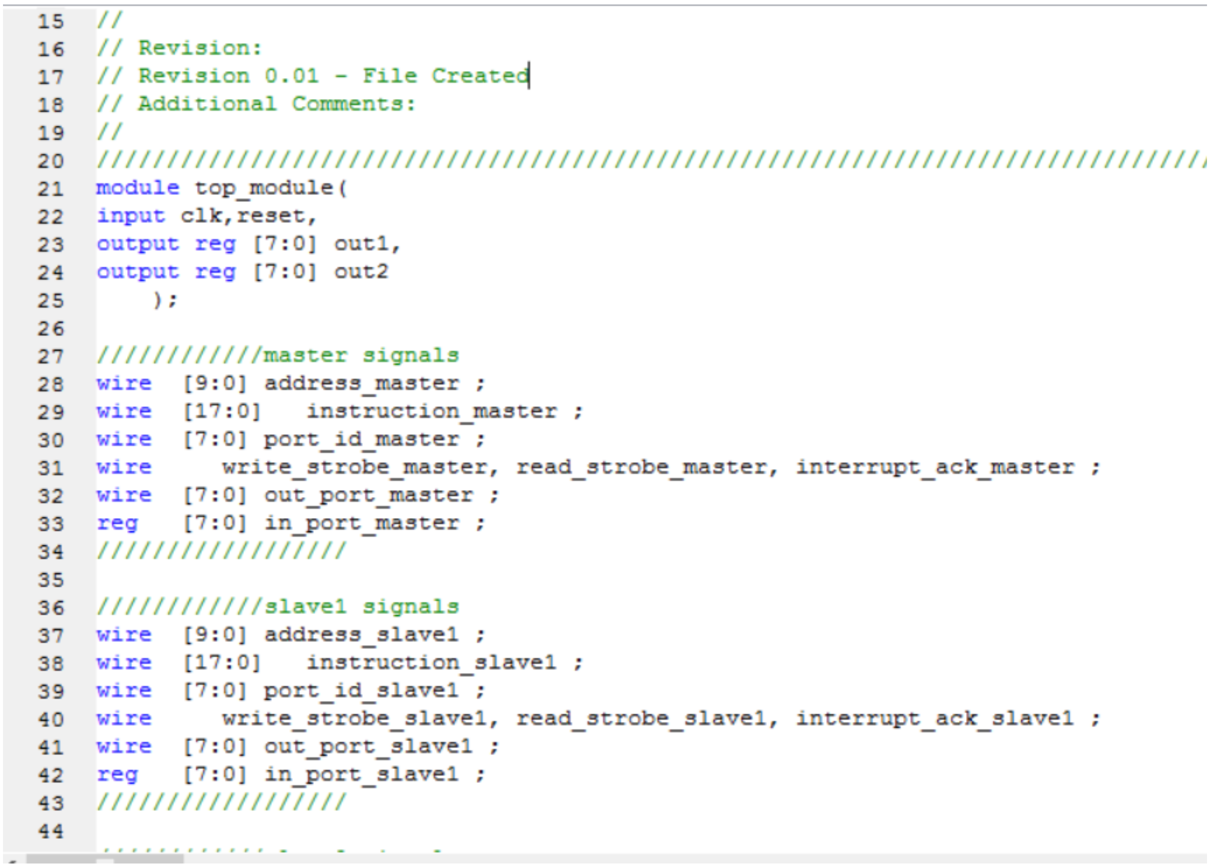


```
                    LOAD sA,05
                    LOAD sA,05
                    LOAD sA,05
address:            INPUT s0,BF
                    TEST s0,02
                    JUMP Z, address
                    INPUT s1,BF
                    OUTPUT s1,40
                    JUMP address
```

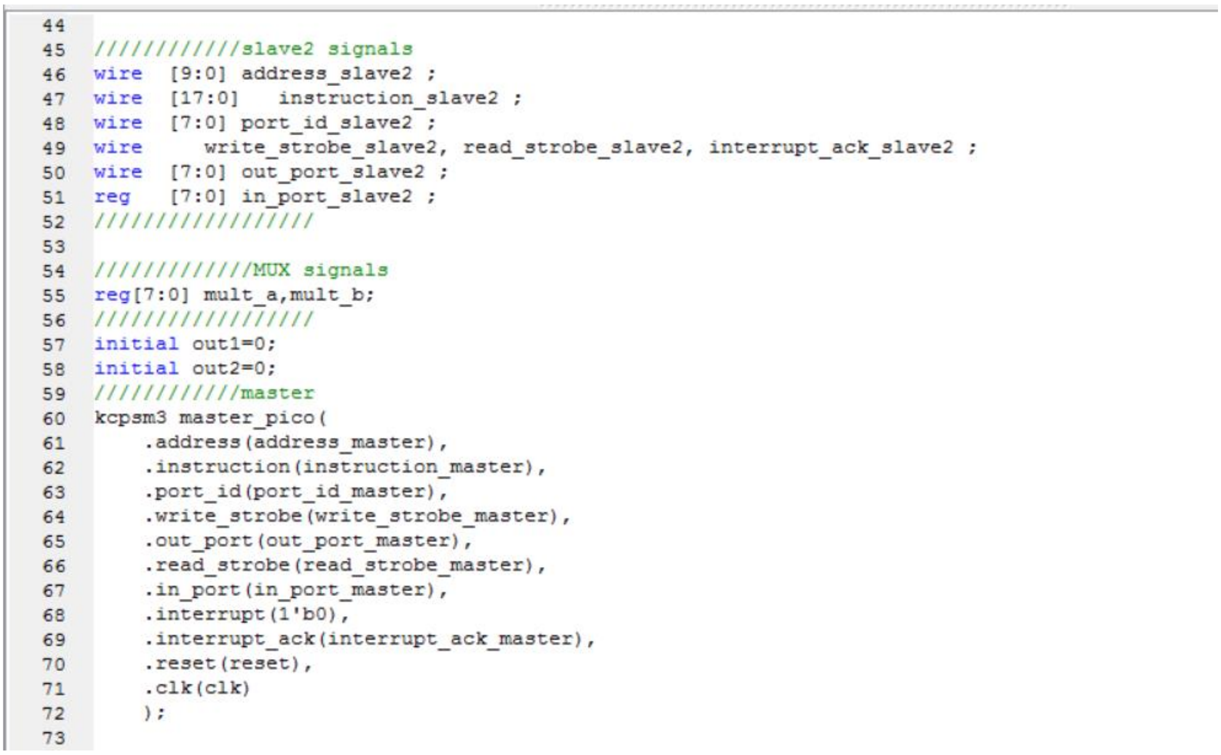
Assembly Code of Slave2 Picoblaze

In the assembly codes of slave picoblazes, “LOAD sA, 05” command is used several times in order to provide the required delay. During this delay, address information is sent by master picoblaze. In slave1 picoblaze, the address is tested with the value “01h”, whereas it is tested with the value “02h” in slave2 picoblaze. After testing the address information, the data read in the input is given to the output. The port_id outputs of slave picoblazes are connected to the in_port of master picoblaze via a multiplexer and the acknowledgement signal is sent form port_id output. Therefore, the value “80h” is sent in slave1 picoblaze and “40h” in slave2 picoblaze.

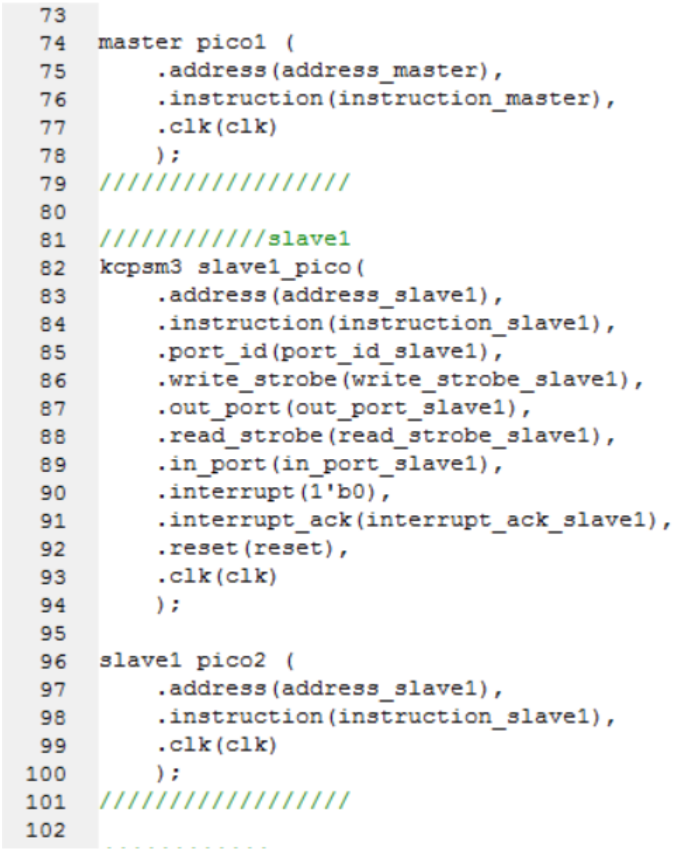
3. Verilog Code:



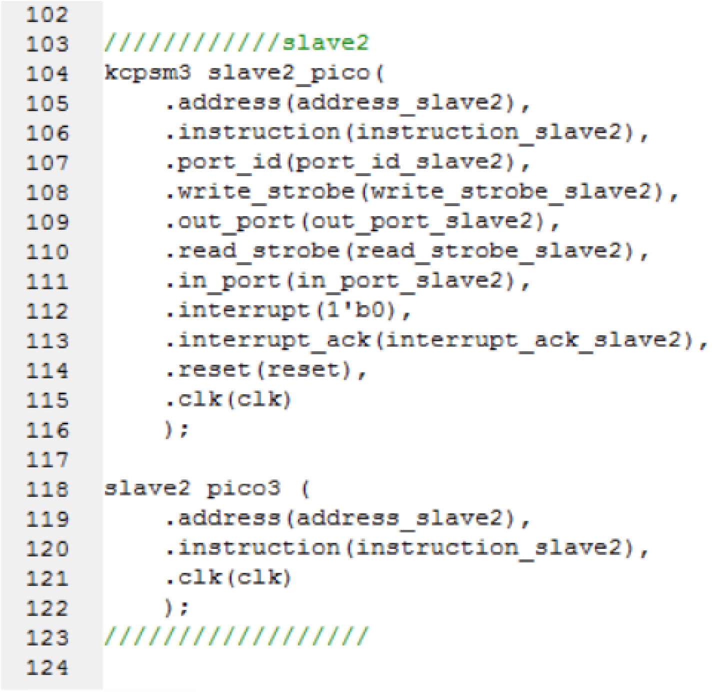
```
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////
module top_module(
input clk,reset,
output reg [7:0] out1,
output reg [7:0] out2
    );

////////////master signals
wire  [9:0] address_master ;
wire  [17:0]   instruction_master ;
wire  [7:0] port_id_master ;
wire     write_strobe_master, read_strobe_master, interrupt_ack_master ;
wire  [7:0] out_port_master ;
reg   [7:0] in_port_master ;
//////////////////

////////////slave1 signals
wire  [9:0] address_slave1 ;
wire  [17:0]   instruction_slave1 ;
wire  [7:0] port_id_slave1 ;
wire     write_strobe_slave1, read_strobe_slave1, interrupt_ack_slave1 ;
wire  [7:0] out_port_slave1 ;
reg   [7:0] in_port_slave1 ;
//////////////////

```



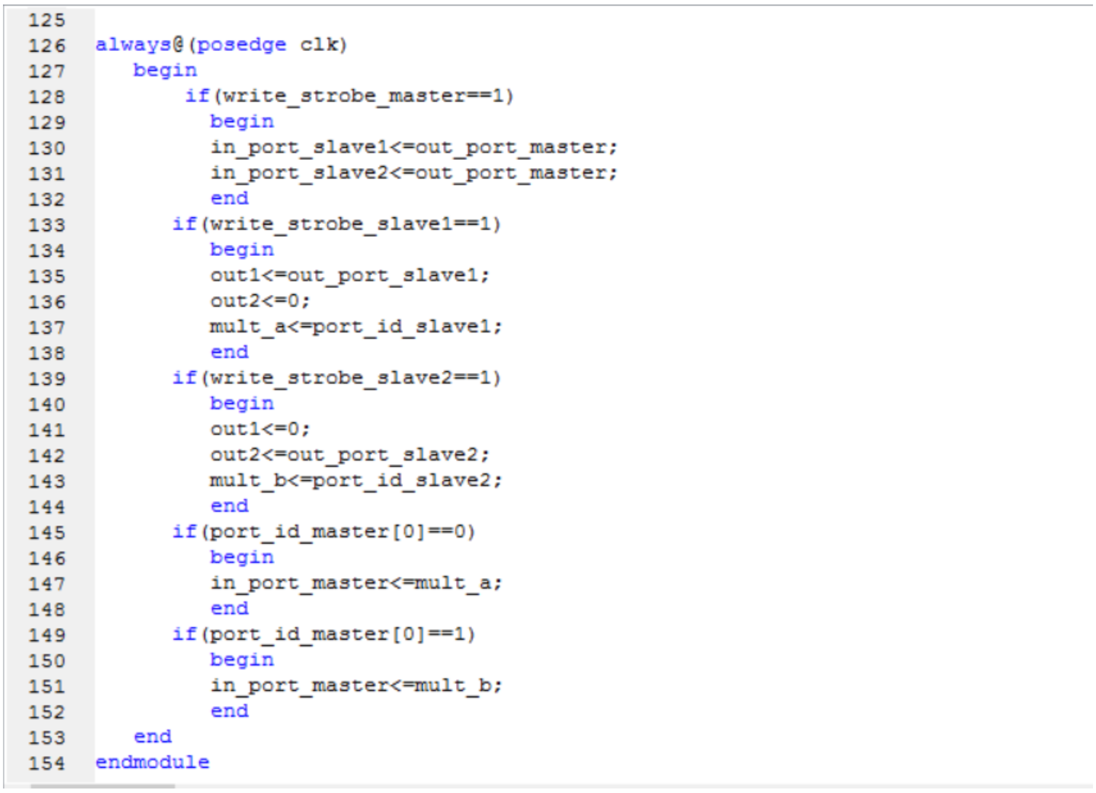
```

////////////slave2 signals
wire  [9:0] address_slave2 ;
wire  [17:0]   instruction_slave2 ;
wire  [7:0] port_id_slave2 ;
wire     write_strobe_slave2, read_strobe_slave2, interrupt_ack_slave2 ;
wire  [7:0] out_port_slave2 ;
reg   [7:0] in_port_slave2 ;
//////////////////

////////////MUX signals
reg[7:0] mult_a,mult_b;
//////////////////
initial out1=0;
initial out2=0;
///////////master
kcpsm3 master_pico(
    .address(address_master),
    .instruction(instruction_master),
    .port_id(port_id_master),
    .write_strobe(write_strobe_master),
    .out_port(out_port_master),
    .read_strobe(read_strobe_master),
    .in_port(in_port_master),
    .interrupt(1'b0),
    .interrupt_ack(interrupt_ack_master),
    .reset(reset),
    .clk(clk)
    );

```



```
master pico1  (
     .address(address_master),
     .instruction(instruction_master),
     .clk(clk)
     );
//////////////////

////////////slave1
kcpsm3 slave1_pico(
     .address(address_slave1),
     .instruction(instruction_slave1),
     .port_id(port_id_slave1),
     .write_strobe(write_strobe_slave1),
     .out_port(out_port_slave1),
     .read_strobe(read_strobe_slave1),
     .in_port(in_port_slave1),
     .interrupt(1'b0),
     .interrupt_ack(interrupt_ack_slave1),
     .reset(reset),
     .clk(clk)
     );

slave1 pico2  (
     .address(address_slave1),
     .instruction(instruction_slave1),
     .clk(clk)
     );
//////////////////

```

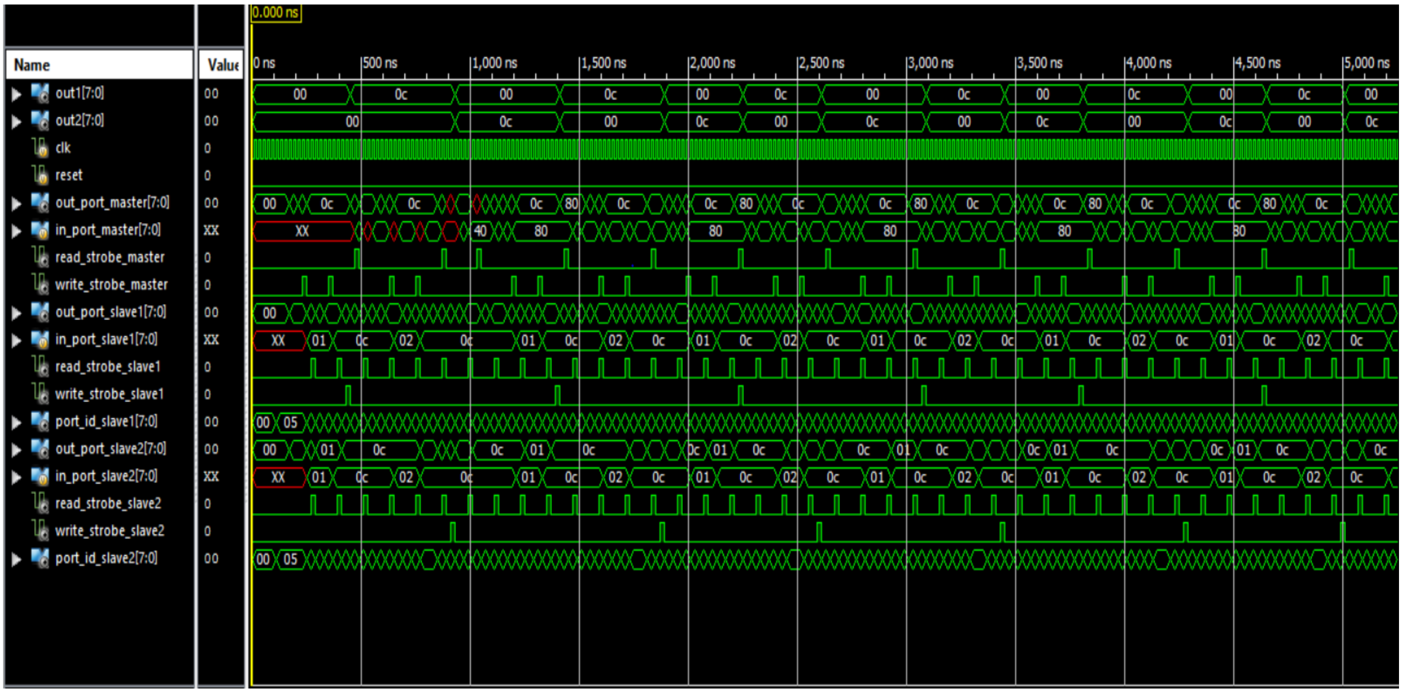


```

////////////slave2
kcpsm3 slave2_pico(
     .address(address_slave2),
     .instruction(instruction_slave2),
     .port_id(port_id_slave2),
     .write_strobe(write_strobe_slave2),
     .out_port(out_port_slave2),
     .read_strobe(read_strobe_slave2),
     .in_port(in_port_slave2),
     .interrupt(1'b0),
     .interrupt_ack(interrupt_ack_slave2),
     .reset(reset),
     .clk(clk)
     );

slave2 pico3  (
     .address(address_slave2),
     .instruction(instruction_slave2),
     .clk(clk)
     );
//////////////////

```



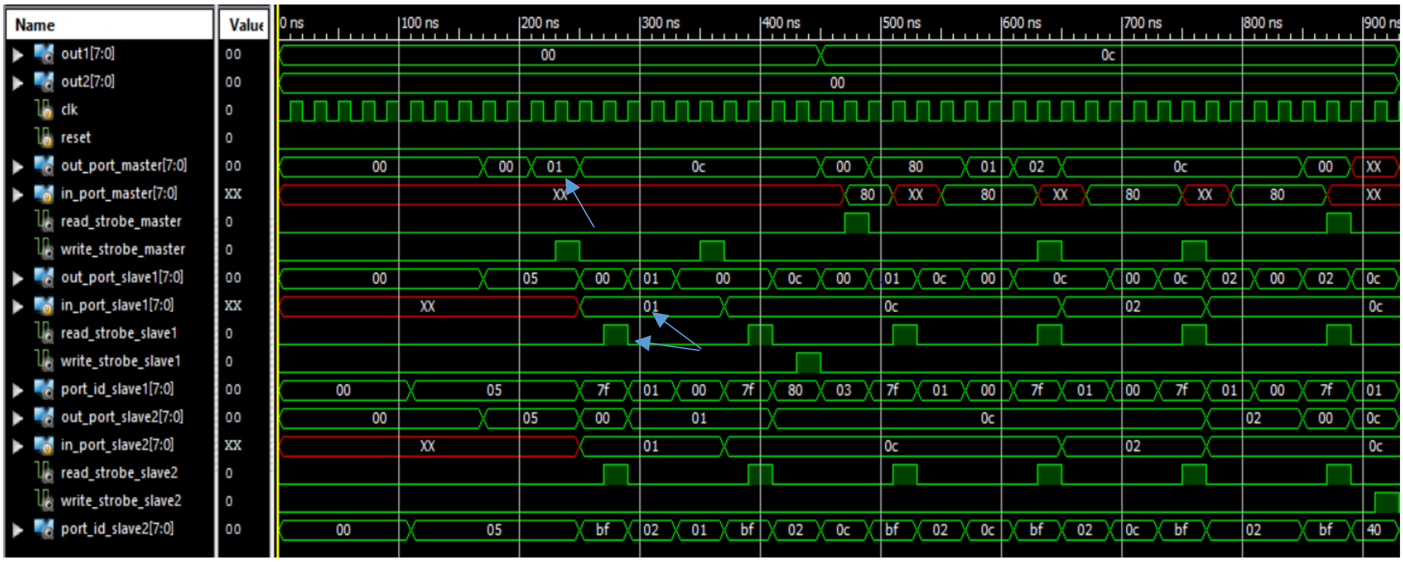
```

always@(posedge clk)
   begin
       if(write_strobe_master==1)
         begin
         in_port_slave1<=out_port_master;
         in_port_slave2<=out_port_master;
         end
      if(write_strobe_slave1==1)
         begin
         out1<=out_port_slave1;
         out2<=0;
         mult_a<=port_id_slave1;
         end
      if(write_strobe_slave2==1)
         begin
         out1<=0;
         out2<=out_port_slave2;
         mult_b<=port_id_slave2;
         end
      if(port_id_master[0]==0)
         begin
         in_port_master<=mult_a;
         end
      if(port_id_master[0]==1)
         begin
         in_port_master<=mult_b;
         end
   end
endmodule
```

4. Simulation:

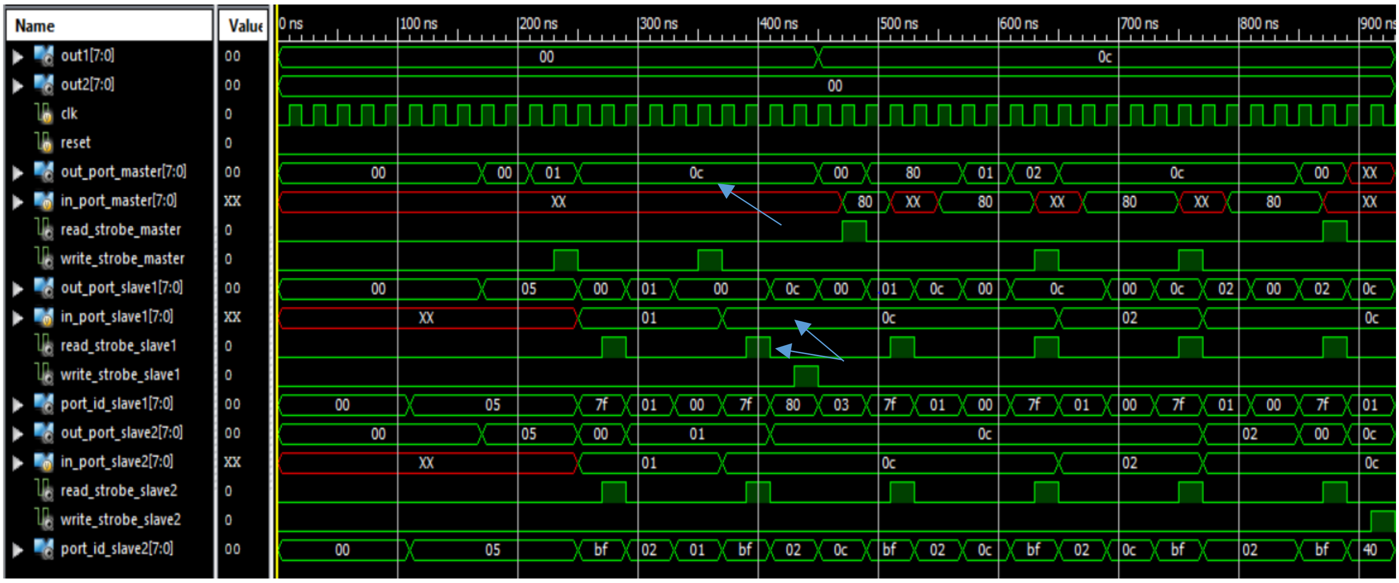


General Simulation Screen

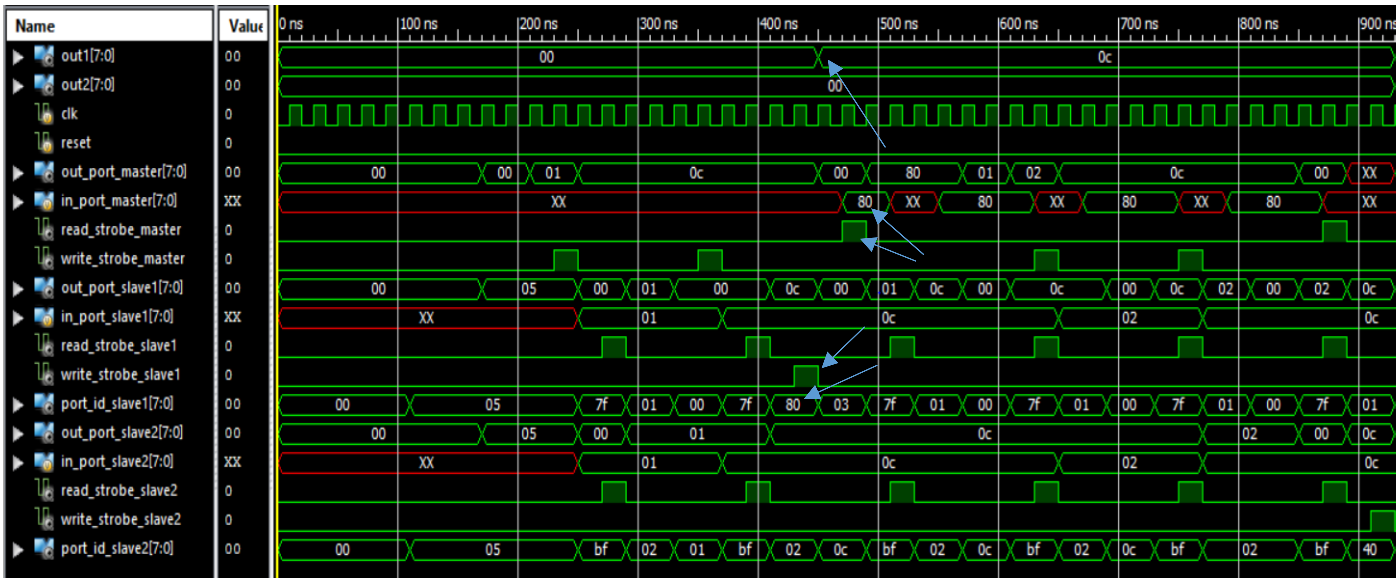
The information “0Ch” is sent to two outputs one by one.



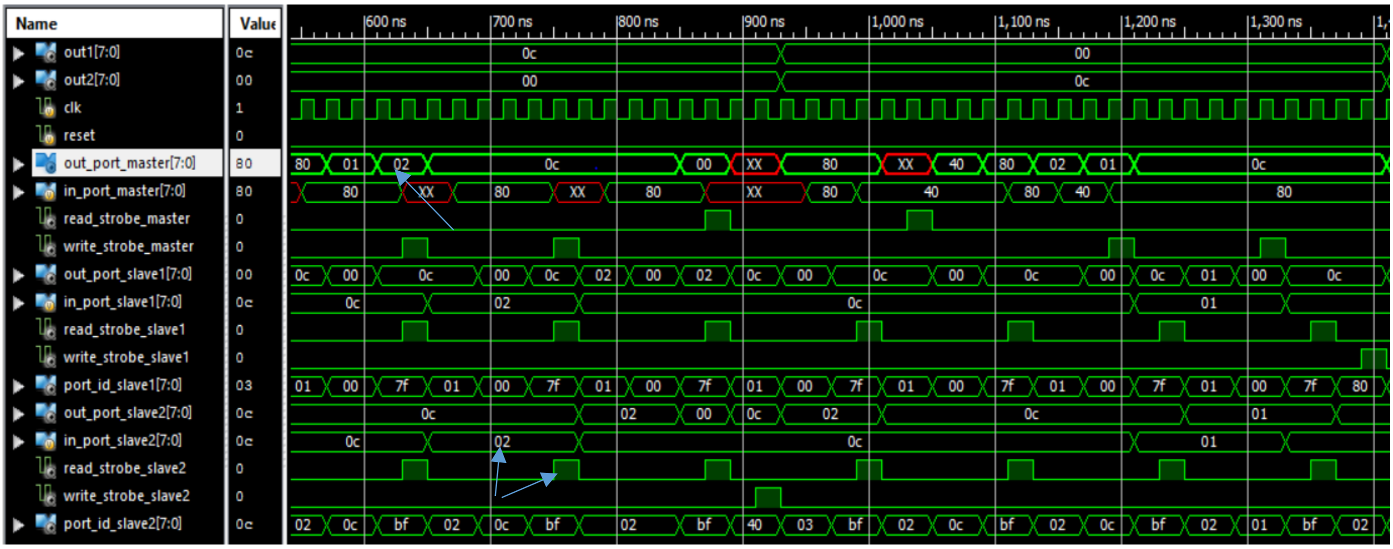
The address information “01h” is given to the output of master picoblaze, it is assigned to the input of slave1 picoblaze and read by slave1 picoblaze.



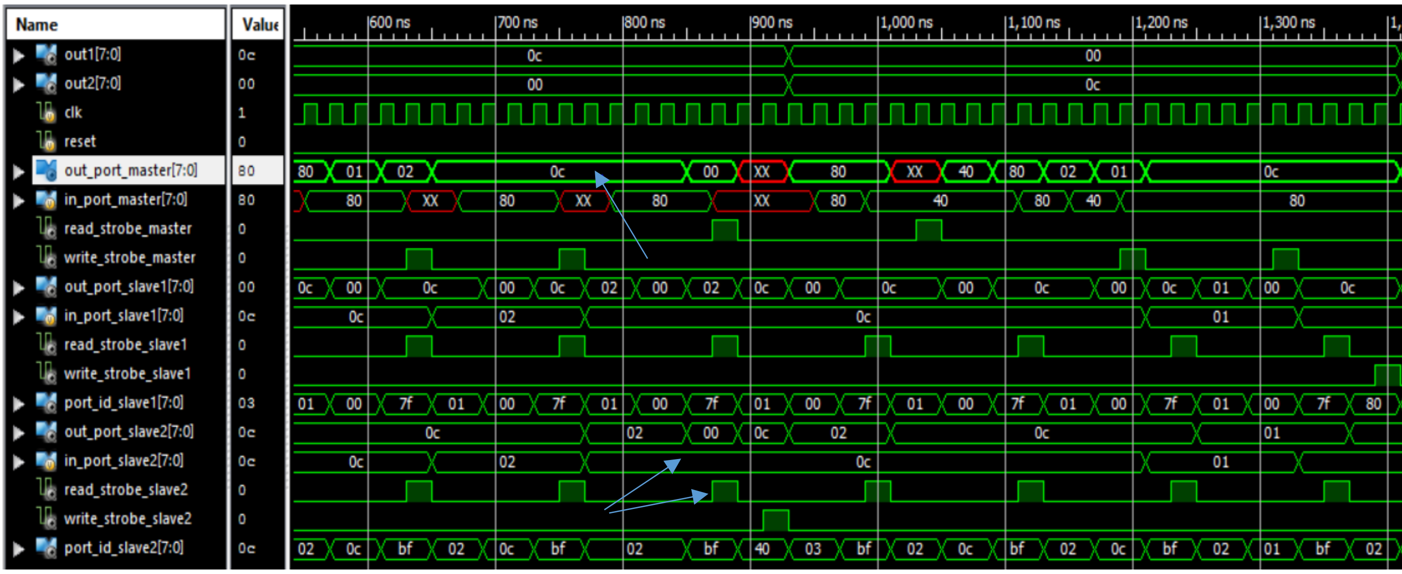
The stored data “0Ch” is given to the output of master picoblaze, it is assigned to the input of slave1 picoblaze and read by slave1 picoblaze.



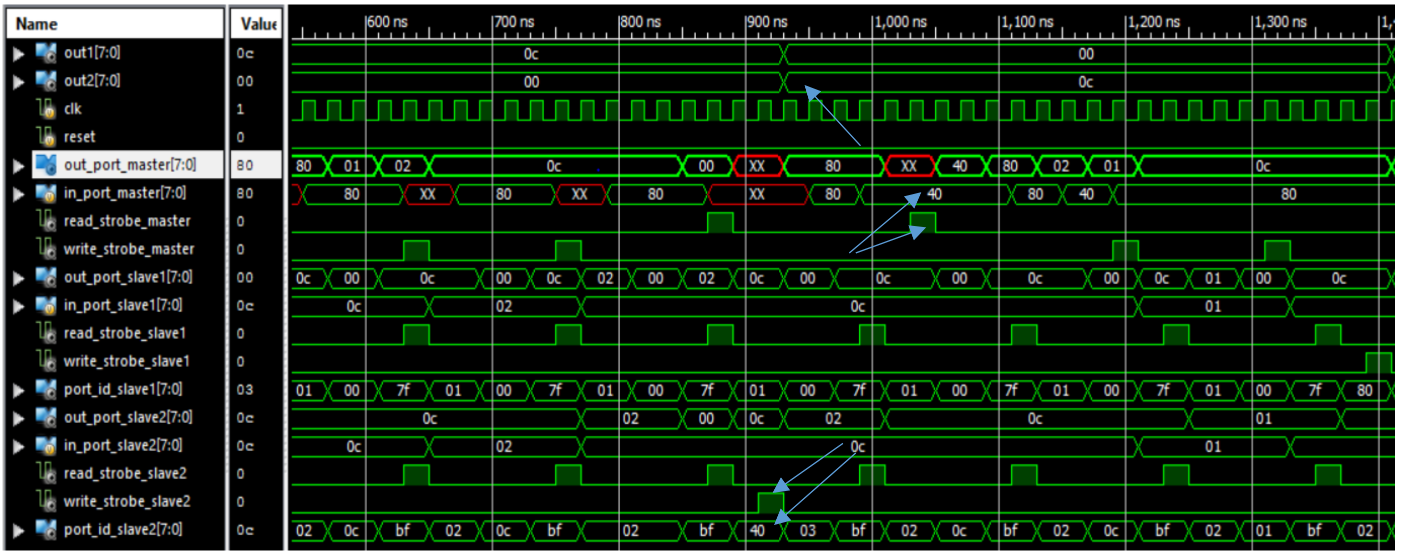
The stored data “0Ch” is given to the output of slave1 picoblaze and it is assigned to out1. Simultaneously, “80h” is given to the port_id_slave1 and it is assigned to the input of master picoblaze and read by master picoblaze.



The address information “02h” is given to the output of master picoblaze, it is assigned to the input of slave2 picoblaze and read by slave2 picoblaze.



The stored data “0Ch” is given to the output of master picoblaze, it is assigned to the input of slave2 picoblaze and read by slave2 picoblaze.



The stored data “0Ch” is given to the output of slave2 picoblaze and it is assigned to out2. Simultaneously, “40h” is given to the port_id_slave2 and it is assigned to the input of master picoblaze and read by master picoblaze.