```
library IEEE;
use IEEE.std_logic_1164.all;
use     ieee.std_logic_signed.all;

entity alu is

 port(s1,s2:in std_logic_vector(15 downto 0);
        controls:in std_logic_vector(1 downto 0);
        c_flag:out std_logic;
      result:out std_logic_vector(15 downto 0));
end alu;

architecture behav_alu of alu is
signal results:std_logic_vector(15 downto 0);
begin

 process(controls)
  begin
   case controls is
    when "00" =>
        results<=s1;
    when "01" =>
        results<=s1+s2;
        when "10"=>
                results<=s1 and s2;
        when others =>
                results<= not(s1);
   end case;
        if(controls="01") then
                c_flag<=(s1(15) and s2(15)) or ((s1(15) xor s2(15)) and (not results(15)));
        else
                c_flag<='0';
        end if;
  end process;
        result<=results;
end architecture behav_alu;
```

```
library ieee;
use ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;

entity control is
port(   irout:in std_logic_vector(15 downto 0);
                clk,clear,n,z,c:in std_logic;
                selmux1,selmux2,drin,marin,mdrin,irin,mdrmuxs,pcin,pccontrol,cset,rw:out
std_logic;
                alucontrol,selbus,sexts:out std_logic_vector(1 downto 0);
                sels1,sels2,seldr:out std_logic_vector(2 downto 0));
end entity;

architecture rtl of control is
        type states is (start,fetch0,fetch1,fetch2,exec0,exec1,exec2,exec3,exec4);
        signal state:states;

begin
process(state,n,z,c,clk,clear)
begin
if (clear='1') then
state<=start;
else
        if(clk'event and clk='0')then

                case state is
                        when start =>
                                state<=fetch0;
                                selmux1<='0';
                                selmux2<='0';
                                drin<='0';
                                marin<='0';
                                mdrin<='0';
                                irin<='0';
                                mdrmuxs<='0';
                                pcin<='0';
                                pccontrol<='0';
                                cset<='0';
                                rw<='0';
                                alucontrol<="00";
                                selbus<="00";
                                sexts<="00";
                                sels1<="000";
                                sels2<="000";
                                seldr<="000";
                        when fetch0 =>
                                selmux1<='0';
                                selmux2<='0';
                                drin<='0';
                                mdrin<='0';
                                irin<='0';
                                mdrmuxs<='0';
                                cset<='0';
                                rw<='0';
                                alucontrol<="00";
                                sexts<="00";
                                sels1<="000";
                                sels2<="000";
                                seldr<="000";

                                selbus<="00";
                                marin<='1';
                pcin<='1';
                pccontrol<='0';
                state<=fetch1;
            when fetch1 =>
                marin<='0';
                pcin<='0';

                rw<='0';
                mdrin<='1';
                mdrmuxs<='0';
                state<=fetch2;
            when fetch2 =>
                mdrin<='0';

                irin<='1';
                selbus<="10";
                state<=exec0;
            when exec0 =>
                irin<='0';
                case irout(15 downto 12) is
                    when "0000"=>
                        if ((irout(11)=n) and (irout(10)=z) and
(irout(9)=c))then
                            selmux1<='1';
                            selmux2<='1';
                            selbus<="01";
                            alucontrol<="01";
                            pcin<='1';
                            pccontrol<='1';
                            sexts<="10";
                        end if;
                        state<=fetch0;
                    when "0001"=>
                        case irout(5) is
                            when '1'=>
                                selmux1<='0';
                                selmux2<='1';
                                sels1<=irout(8 downto 6);
                                seldr<=irout(11 downto 9);
                                sexts<="00";
                                alucontrol<="01";       --add
                                selbus<="01";
                                drin<='1';
                                cset<='1';
                            when others=>
                                selmux1<='0';
                                selmux2<='0';
                                sels1<=irout(8 downto 6);
                                seldr<=irout(11 downto 9);
                                sels2<=irout(2 downto 0);
                                alucontrol<="01";       --add
                                selbus<="01";
                                drin<='1';
                                cset<='1';
                        end case;
                        state<=fetch0;
                    when "0010"=>
                            selmux1<='1';
                            selmux2<='1';
                            sexts<="10";
                            alucontrol<="01";
                            selbus<="01";
```

```
                            marin<='1';
                            state<=exec1;
                    when "0011"=>
                            selmux1<='1';
                            selmux2<='1';
                            sexts<="10";
                            alucontrol<="01";
                            selbus<="01";
                            marin<='1';
                            state<=exec1;
                    when "0100"=>
                            selmux1<='1';
                            alucontrol<="00";
                            selbus<="01";
                            seldr<="111";
                            drin<='1';
                            state<=exec1;
                    when "0101"=>
                        case irout(5) is
                            when '1'=>
                                selmux1<='0';
                                selmux2<='1';
                                sels1<=irout(8 downto 6);
                                seldr<=irout(11 downto 9);
                                sexts<="00";
                                alucontrol<="10";       --and
                                selbus<="01";
                                drin<='1';
                                cset<='1';
                            when others=>
                                selmux1<='0';
                                selmux2<='0';
                                sels1<=irout(8 downto 6);
                                seldr<=irout(11 downto 9);
                                sels2<=irout(2 downto 0);
                                alucontrol<="10";       --and
                                selbus<="01";
                                drin<='1';
                                cset<='1';
                        end case;
                        state<=fetch0;
                    when "0110"=>
                            sels1<=irout(8 downto 6);
                            selmux1<='0';
                            selmux2<='1';
                            sexts<="01";
                            alucontrol<="01";
                            selbus<="01";
                            marin<='1';
                            state<=exec1;
                    when "0111"=>
                            sels1<=irout(8 downto 6);
                            selmux1<='0';
                            selmux2<='1';
                            sexts<="01";
                            alucontrol<="01";
                            selbus<="01";
                            marin<='1';
                            state<=exec1;
                    when "1001"=>
                            sels1<=irout(8 downto 6);
                            selmux1<='0';
```

```
                                                        alucontrol<="11";
                                                        selbus<="01";
                                                        seldr<=irout(11 downto 9);
                                                        drin<='1';
                                                        cset<='1';
                                                        state<=fetch0;
                                        when "1010"=>
                                                        selmux1<='1';
                                                        selmux2<='1';
                                                        sexts<="10";
                                                        alucontrol<="01";
                                                        selbus<="01";
                                                        marin<='1';
                                                        state<=exec1;
                                        when "1011"=>
                                                        selmux1<='1';
                                                        selmux2<='1';
                                                        sexts<="10";
                                                        alucontrol<="01";
                                                        selbus<="01";
                                                        marin<='1';
                                                        state<=exec1;
                                        when "1100"=>
                                                        sels1<=irout(8 downto 6);
                                                        selmux1<='0';
                                                        alucontrol<="00";
                                                        selbus<="01";
                                                        pcin<='1';
                                                        pccontrol<='1';
                                                        state<=fetch0;
                                        when "1101"=>
                                                        sels1<="111";
                                                        selmux1<='0';
                                                        alucontrol<="00";
                                                        selbus<="01";
                                                        pcin<='1';
                                                        pccontrol<='1';
                                                        state<=fetch0;
                                        when others=>
                                                        selmux1<='1';
                                                        selmux2<='1';
                                                        sexts<="10";
                                                        alucontrol<="01";
                                                        selbus<="01";
                                                        cset<='1';
                                                        seldr<=irout(11 downto 9);
                                                        drin<='1';
                                                        state<=fetch0;
                                end case;
                        when exec1 =>
                                case irout(15 downto 12) is
                                        when "0010"=>
                                                        selmux1<='0';
                                                        selmux2<='0';
                                                        sexts<="00";
                                                        alucontrol<="00";
                                                        selbus<="00";
                                                        marin<='0';


                                                        rw<='0';
                                                        mdrin<='1';
```

```
                                                        mdrmuxs<='0';
                                                        state<=exec2;
                                        when "0011"=>
                                                        selmux2<='0';
                                                        sexts<="00";
                                                        marin<='0';

                                                        mdrin<='1';
                                                        mdrmuxs<='1';
                                                        selmux1<='0';
                                                        sels1<=irout(11 downto 9);
                                                        alucontrol<="00";
                                                        selbus<="01";
                                                        state<=exec2;
                                        when "0100"=>
                                                        seldr<="000";
                                                        drin<='0';

                                                        case irout(11) is
                                                                when '1'=>
                                                                        selmux1<='1';
                                                                        selmux2<='1';
                                                                        sexts<="11";
                                                                        alucontrol<="01";
                                                                        selbus<="01";
                                                                        pcin<='1';
                                                                        pccontrol<='1';
                                                                when others =>
                                                                        sels1<=irout(8 downto 6);
                                                                        selmux1<='0';
                                                                        alucontrol<="00";
                                                                        selbus<="01";
                                                                        pcin<='1';
                                                                        pccontrol<='1';
                                                        end case;
                                                        state<=fetch0;
                                        when "0110"=>
                                                        sels1<="000";
                                                        selmux2<='0';
                                                        sexts<="00";
                                                        alucontrol<="00";
                                                        selbus<="00";
                                                        marin<='0';


                                                        rw<='0';
                                                        mdrin<='1';
                                                        mdrmuxs<='0';
                                                        state<=exec2;
                                        when "0111"=>
                                                        selmux2<='0';
                                                        sexts<="00";
                                                        marin<='0';


                                                        mdrin<='1';
                                                        mdrmuxs<='1';
                                                        selmux1<='0';
                                                        sels1<=irout(11 downto 9);
                                                        alucontrol<="00";
                                                        selbus<="01";
                                                        state<=exec2;
                                        when "1010"=>
                                                        selmux1<='0';
                                                        selmux2<='0';
                                                        sexts<="00";
                                                        alucontrol<="00";
                                                        selbus<="00";
                                                        marin<='0';


                                                        rw<='0';
                                                        mdrin<='1';
                                                        mdrmuxs<='0';
                                                        state<=exec2;
                                        when "1011"=>
                                                        selmux1<='0';
                                                        selmux2<='0';
                                                        sexts<="00";
                                                        alucontrol<="00";
                                                        selbus<="00";
                                                        marin<='0';


                                                        rw<='0';
                                                        mdrin<='1';
                                                        mdrmuxs<='0';
                                                        state<=exec2;
                                        when others=>
                                                        state<=fetch0;
                                end case;
                        when exec2 =>
                                case irout(15 downto 12) is
                                        when "0010"=>
                                                        mdrin<='0';
                                                        seldr<=irout(11 downto 9);
                                                        drin<='1';
                                                        selbus<="10";
                                                        cset<='1';
                                                        state<=fetch0;
                                        when "0011"=>
                                                        mdrin<='0';
                                                        mdrmuxs<='0';
                                                        selmux1<='0';
                                                        sels1<="000";
                                                        alucontrol<="00";
                                                        selbus<="00";


                                                        rw<='1';
                                                        state<=fetch0;
                                        when "0110"=>
                                                        mdrin<='0';


                                                        seldr<=irout(11 downto 9);
                                                        drin<='1';
                                                        selbus<="10";
                                                        cset<='1';
                                                        state<=fetch0;
                                        when "0111"=>
                                                        mdrin<='0';
                                                        mdrmuxs<='0';
                                                        selmux1<='0';
```

```
                                                        sels1<="000";
                                                        alucontrol<="00";
                                                        selbus<="00";


                                                        rw<='1';
                                                        state<=fetch0;
                                        when "1010"=>
                                                        mdrin<='0';


                                                        selbus<="10";
                                                        marin<='1';
                                                        state<=exec3;
                                        when "1011"=>
                                                        mdrin<='0';


                                                        selbus<="10";
                                                        marin<='1';
                                                        state<=exec3;
                                        when others=>
                                                        state<=fetch0;
                                end case;
                        when exec3 =>
                                case irout(15 downto 12) is
                                        when "1010"=>
                                                        selbus<="00";
                                                        marin<='0';


                                                        rw<='0';
                                                        mdrin<='1';
                                                        mdrmuxs<='0';
                                                        state<=exec4;
                                        when "1011"=>
                                                        marin<='0';


                                                        mdrin<='1';
                                                        mdrmuxs<='1';
                                                        selmux1<='0';
                                                        sels1<=irout(11 downto 9);
                                                        alucontrol<="00";
                                                        selbus<="01";
                                                        state<=exec4;
                                        when others=>
                                                        state<=fetch0;
                                end case;
                        when others =>
                                case irout(15 downto 12) is
                                        when "1010"=>
                                                        mdrin<='0';


                                                        seldr<=irout(11 downto 9);
                                                        drin<='1';
                                                        selbus<="10";
                                                        cset<='1';
                                                        state<=fetch0;
                                        when "1011"=>
                                                        mdrin<='0';
                                                        mdrmuxs<='0';
                                                        selmux1<='0';
                                                        sels1<="000";
                                                        alucontrol<="00";
                                                        selbus<="00";


                                                        rw<='1';
                                                        state<=fetch0;
                                        when others=>
                                                        state<=fetch0;
                                end case;
                end case;
end if;
end if;

end process;
end architecture rtl;
```

```
library ieee;
use ieee.std_logic_1164.all;

entity drdec is

        port(   inen:in std_logic;
                        seldr:in std_logic_vector(2 downto 0);
                        drin:out std_logic_vector(7 downto 0));
end entity drdec;

architecture rtl of drdec is
begin
process (seldr)
begin
if(inen='1') then
        case seldr is
                when "000" => drin <="00000001";
                when "001" => drin <="00000010";
                when "010" => drin <="00000100";
                when "011" => drin <="00001000";
                when "100" => drin <="00010000";
                when "101" => drin <="00100000";
                when "110" => drin <="01000000";
                when others => drin <="10000000";
        end case;
else
        drin<="00000000";
end if;
end process;
end architecture rtl;
```

```
library ieee;
use ieee.std_logic_1164.all;

entity ir is

	port(	irin:in std_logic_vector(15 downto 0);
			clear,clk,irins:in std_logic;
			irout:out std_logic_vector(15 downto 0));
end entity ir;

architecture rtl of ir is

begin
process(clear,clk,irin)
	begin
	if (clear ='1') then
		irout<="0000000000000000";
	else
		if(clk'event and clk='1') then
			if (irins='1') then
			irout <= irin;
			end if;
		end if;
	end if;
end process;
end architecture rtl;
```

```
library IEEE;
use IEEE.std_logic_1164.all;

entity lc3 is
	port (clk,clear:in std_logic;
			busstate:out std_logic_vector(15 downto 0));
end entity;

architecture rtl of lc3 is
signal
s1,s2,re,pcout,sextout,irout,marout,mdrin,mdrout,memout,busout,rfs1,rfs2:std_logic_vector(15
downto 0);
signal alucontrols,selbus,sexts:std_logic_vector(1 downto 0);
signal
selmux1,selmux2,drins,marins,mdrins,irins,mdrmuxs,pcins,pccontrols,csets,rw,c,nout,zout,cout:std_l
ogic;
signal sels1s,sels2s,seldrs:std_logic_vector(2 downto 0);

component alu
 port(s1,s2:in std_logic_vector(15 downto 0);
	controls:in std_logic_vector(1 downto 0);
	c_flag:out std_logic;
      result:out std_logic_vector(15 downto 0));
end component;

component control
port(	irout:in std_logic_vector(15 downto 0);
		clk,clear,n,z,c:in std_logic;
		selmux1,selmux2,drin,marin,mdrin,irin,mdrmuxs,pcin,pccontrol,cset,rw:out
std_logic;
		alucontrol,selbus,sexts:out std_logic_vector(1 downto 0);
		sels1,sels2,seldr:out std_logic_vector(2 downto 0));
end component;

component ir
	port(	irin:in std_logic_vector(15 downto 0);
			clear,clk,irins:in std_logic;
			irout:out std_logic_vector(15 downto 0));
end component;

component mar

	port(	marin:in std_logic_vector(15 downto 0);
			clear,clk,marins:in std_logic;
			marout:out std_logic_vector(15 downto 0));
end component;

component mdr
	port(	mdrin:in std_logic_vector(15 downto 0);
			clear,clk,mdrins:in std_logic;
			mdrout:out std_logic_vector(15 downto 0));
end component;

component mem
port(	address,wdata:in std_logic_vector(15 downto 0);
		rw:in	std_logic;
		rdata:out std_logic_vector(15 downto 0));
end component;


component muxbus
port(pcout,aluout,mdrout:in std_logic_vector(15 downto 0);
        sels:in std_logic_vector(1 downto 0);
        busout:out std_logic_vector(15 downto 0));
end component;

component muxs
port(in1,in2:in std_logic_vector(15 downto 0);
        sels:in std_logic;
        outdata:out std_logic_vector(15 downto 0));
end component;

component nzc
port(   data:in std_logic_vector(15 downto 0);
                        c_flag,clear,clk,inen:in std_logic;
                        n,z,c:out std_logic);
end component;

component pc
        port(   pcload:in std_logic_vector(15 downto 0);
                        pccontrol,clear,clk,pcin:in std_logic;
                        pcout:buffer std_logic_vector(15 downto 0));
end component;

component regf
        port(   regin:in std_logic_vector(15 downto 0);
                        clear,clk,drin:in std_logic;
                        seldr,sels1,sels2:in std_logic_vector(2 downto 0);
                        s1out,s2out:out std_logic_vector(15 downto 0));
end component;

component sext
        port(   irout:in std_logic_vector(10 downto 0);
                        selir:in std_logic_vector(1 downto 0);
                        sext:out std_logic_vector(15 downto 0));
end component;

begin
u0: alu port map(s1,s2,alucontrols,c,re);
u1: control port
map(irout,clk,clear,nout,zout,cout,selmux1,selmux2,drins,marins,mdrins,irins,mdrmuxs,pcins,pccontr
ols,csets,rw,alucontrols,selbus,sexts,sels1s,sels2s,seldrs);
u2: ir port map(busout,clear,clk,irins,irout);
u3: mar port map(busout,clear,clk,marins,marout);
u4: mdr port map(mdrin,clear,clk,mdrins,mdrout);
u5: mem port map(marout,mdrout,rw,memout);
u6: muxbus port map(pcout,re,mdrout,selbus,busout);
u7: muxs port map (pcout,rfs1,selmux1,s1);
u8: muxs port map (sextout,rfs2,selmux2,s2);
u9: muxs port map (busout,memout,mdrmuxs,mdrin);
u10:nzc port map (busout,c,clear,clk,csets,nout,zout,cout);
u11:pc port map(busout,pccontrols,clear,clk,pcins,pcout);
u12:regf port map(busout,clear,clk,drins,seldrs,sels1s,sels2s,rfs1,rfs2);
u13:sext port map(irout(10 downto 0),sexts,sextout);
busstate<=busout;
end architecture;
```

```
library ieee;
use ieee.std_logic_1164.all;

entity mar is

        port(   marin:in std_logic_vector(15 downto 0);
                        clear,clk,marins:in std_logic;
                        marout:out std_logic_vector(15 downto 0));
end entity mar;

architecture rtl of mar is

begin
process(clear,clk,marin)
        begin
        if (clear ='1') then
                marout<="0000000000000000";
        else
                if(clk'event and clk='1') then
                        if (marins='1') then
                        marout <= marin;
                        end if;
                end if;
        end if;
end process;
end architecture rtl;
```

```
library ieee;
use ieee.std_logic_1164.all;

entity mdr is

        port(   mdrin:in std_logic_vector(15 downto 0);
                        clear,clk,mdrins:in std_logic;
                        mdrout:out std_logic_vector(15 downto 0));
end entity mdr;

architecture rtl of mdr is

begin
process(clear,clk,mdrin)
        begin
        if (clear ='1') then
                mdrout<="0000000000000000";
        else
                if(clk'event and clk='1') then
                        if (mdrins='1') then
                        mdrout <= mdrin;
                        end if;
                end if;
        end if;
end process;
end architecture rtl;
```

```
library IEEE;
use IEEE.std_logic_1164.all;

entity mem is
port(   address,wdata:in std_logic_vector(15 downto 0);
                rw:in   std_logic;
                rdata:out std_logic_vector(15 downto 0));
end entity;

architecture rtl of mem is
begin

process (address,wdata,rw)
variable a :std_logic_vector(15 downto 0);
begin
        case rw is
                when '1'=>      --write
                        if(address="0000000000100110") then
                                a:=wdata;
                        end if;
                when others=>
                                case address is
                                        when "0000000000000000"=>
                                                                rdata<="0100000000011100"; --ld
                                        when "0000000000000010"=>
                                                                rdata<="1010001000011010"; --ldi
                                        when "0000000000000100"=>
                                                                rdata<="0110010000000010";--ldr
                                        when "0000000000000110"=>
                                                                rdata<="0001100000000001"; --add
                                        when "0000000000001000"=>
                                                                rdata<="0001101000100110";--add
                                        when "0000000000001010"=>
                                                                rdata<="0101110000000001";--and
                                        when "0000000000001100"=>
                                                                rdata<="0101111000101101";--and
                                        when "0000000000001110"=>
                                                                rdata<="1001101000111111";--not
                                        when "0000000000010000"=>
                                                                rdata<="0011111000010100";--st
                                        when "0000000000010010"=>
                                                                rdata<="0111110000000110";--str
                                        when "0000000000010100"=>
                                                                rdata<="1011100000001110";--sti
                                        when "0000000000010110"=>
                                                                rdata<="1110111000110100";--lea
                                        when "0000000000011000"=>
                                                                rdata<="0000000000101100";--br

                                        when "0000000000011110"=>
                                                                rdata<="0000000000100000";
                                        when "0000000000100000"=>
                                                                rdata<="0000000000010000";
                                        when "0000000000100010"=>
                                                                rdata<="0000000001001100";
                                        when "0000000000100100"=>
                                                                rdata<="0000000000100110";
                                        when "0000000001000110"=>
                                                                rdata<="0100100000000000";
                                        when "0000000001001000"=>
                                                                rdata<="0100000111000000";
                                        when "0000000001001010"=>
```

```
                                                                rdata<="1100000010000000";--jmp
                                                when "0000000001001100"=>
                                                                rdata<="1101000111000000";--ret
                                                when others=>
                                                                rdata<="0000000000000000";
                                        end case;
        end case;
end process;
end architecture;
```

```
library IEEE;
use IEEE.std_logic_1164.all;

entity muxbus is

port(pcout,aluout,mdrout:in std_logic_vector(15 downto 0);
        sels:in std_logic_vector(1 downto 0);
        busout:out std_logic_vector(15 downto 0));
end entity muxbus;

architecture rtl of muxbus is
begin

process (sels)
begin

case sels is
when "00"=>
        busout<=pcout;
when "01"=>
        busout<=aluout;
when others =>
        busout<=mdrout;
end case;
end process;
end architecture rtl;
```

```
library IEEE;
use IEEE.std_logic_1164.all;

entity muxs is

port(in1,in2:in std_logic_vector(15 downto 0);
        sels:in std_logic;
        outdata:out std_logic_vector(15 downto 0));
end entity muxs;

architecture rtl of muxs is
begin

process (sels)
begin

case sels is
when '1'=>
        outdata<=in1;
when others =>
        outdata<=in2;
end case;
end process;
end architecture rtl;
```

```
library IEEE;
use IEEE.std_logic_1164.all;

entity nzc is
        port(   data:in std_logic_vector(15 downto 0);
                        c_flag,clear,clk,inen:in std_logic;
                        n,z,c:out std_logic);
end entity nzc;

architecture rtl of nzc is

begin
process(clear,clk,inen)
        begin
        if (clear ='1') then
                n<='0';
                z<='0';
                c<='0';
        else
                if(clk'event and clk='1') then
                        if (inen='1') then
                        n<=data(15);
                        z<=not(data(0)or data(1)or data(2)or data(3)or data(4)or data(5)or
data(6)or data(7)or data(8)or data(9)or data(10)or data(11)or data(12)or data(13)or data(14)or
data(15));
                        c<=c_flag;
                        end if;
                end if;
        end if;
end process;
end architecture rtl;
```

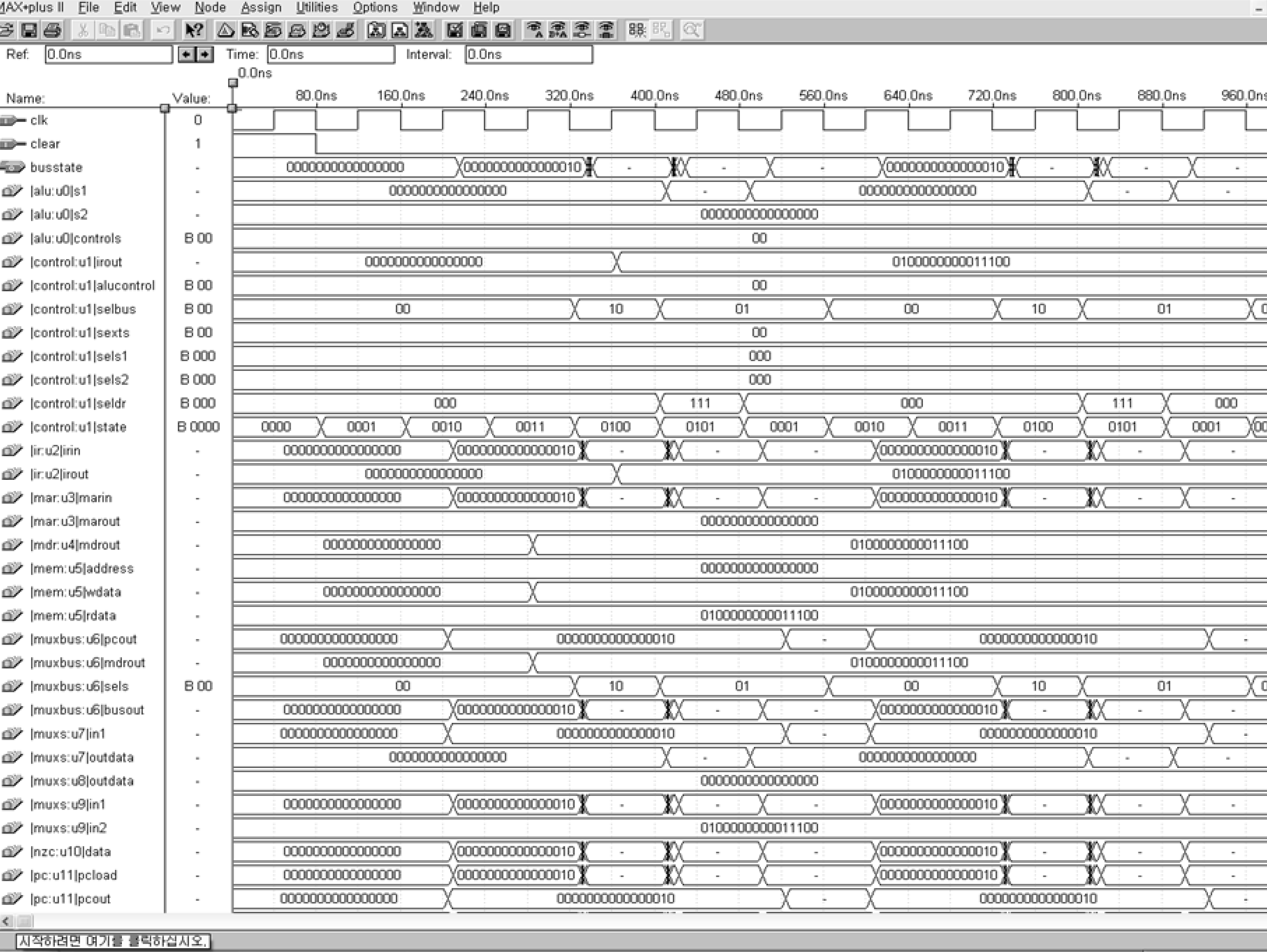
```
library IEEE;
use IEEE.std_logic_1164.all;
use     ieee.std_logic_signed.all;

entity pc is
        port(   pcload:in std_logic_vector(15 downto 0);
                        pccontrol,clear,clk,pcin:in std_logic;
                        pcout:buffer std_logic_vector(15 downto 0));
end entity pc;

architecture rtl of pc is
begin

process(clear,clk,pcin,pccontrol)
        begin
        if (clear ='1') then
                pcout<="0000000000000000";
        else
                if(clk'event and clk='1') then
                        if (pcin='1') then
                                if(pccontrol='1')then
                                        pcout<=pcload;
                                else
                                        pcout<=pcout+"10";
                                end if;
                        end if;
                end if;
        end if;
end process;
end architecture rtl;
```

```
library ieee;
use ieee.std_logic_1164.all;

entity reg is

        port(   regin:in std_logic_vector(15 downto 0);
                        clear,clk,drin:in std_logic;
                        out0:out std_logic_vector(15 downto 0));
end entity reg;

architecture rtl of reg is

begin
process(clear,clk,drin)
        begin
        if (clear ='1') then
                out0<="0000000000000000";
        else
                if(clk'event and clk='1') then
                        if (drin='1') then
                        out0 <= regin;
                        end if;
                end if;
        end if;
end process;
end architecture rtl;
```

```
library ieee;
use ieee.std_logic_1164.all;

entity regf is

        port(   regin:in std_logic_vector(15 downto 0);
                        clear,clk,drin:in std_logic;
                        seldr,sels1,sels2:in std_logic_vector(2 downto 0);
                        s1out,s2out:out std_logic_vector(15 downto 0));
end entity regf;

architecture rtl of regf is
signal inr,outr0,outr1,outr2,outr3,outr4,outr5,outr6,outr7:std_logic_vector(15 downto 0);
signal dregin:std_logic_vector(7 downto 0);
component reg
        port(   regin:in std_logic_vector(15 downto 0);
                        clear,clk,drin:in std_logic;
                        out0:out std_logic_vector(15 downto 0));
end component;

component drdec
        port(   inen:in std_logic;
                        seldr:in std_logic_vector(2 downto 0);
                        drin:out std_logic_vector(7 downto 0));
end component;

begin
u1: drdec port map(drin,seldr,dregin);
u2:     reg port map(regin,clear,clk,dregin(0),outr0);
u3:     reg port map(regin,clear,clk,dregin(1),outr1);
u4:     reg port map(regin,clear,clk,dregin(2),outr2);
u5:     reg port map(regin,clear,clk,dregin(3),outr3);
u6:     reg port map(regin,clear,clk,dregin(4),outr4);
u7:     reg port map(regin,clear,clk,dregin(5),outr5);
u8:     reg port map(regin,clear,clk,dregin(6),outr6);
u9:     reg port map(regin,clear,clk,dregin(7),outr7);
process(sels1)
begin

        CASE sels1 IS
                WHEN "000" => s1out<=outr0;
                WHEN "001" => s1out<=outr1;
                WHEN "010" => s1out<=outr2;
                WHEN "011" => s1out<=outr3;
                WHEN "100" => s1out<=outr4;
                WHEN "101" => s1out<=outr5;
                WHEN "110" => s1out<=outr6;
                WHEN OTHERS => s1out<=outr7;
        END CASE;
end process;
process(sels2)
begin

        CASE sels2 IS
                WHEN "000" => s2out<=outr0;
                WHEN "001" => s2out<=outr1;
                WHEN "010" => s2out<=outr2;
                WHEN "011" => s2out<=outr3;
                WHEN "100" => s2out<=outr4;
                WHEN "101" => s2out<=outr5;
                WHEN "110" => s2out<=outr6;
                WHEN OTHERS => s2out<=outr7;
```

```
        END CASE;
end process;
end architecture rtl;
```

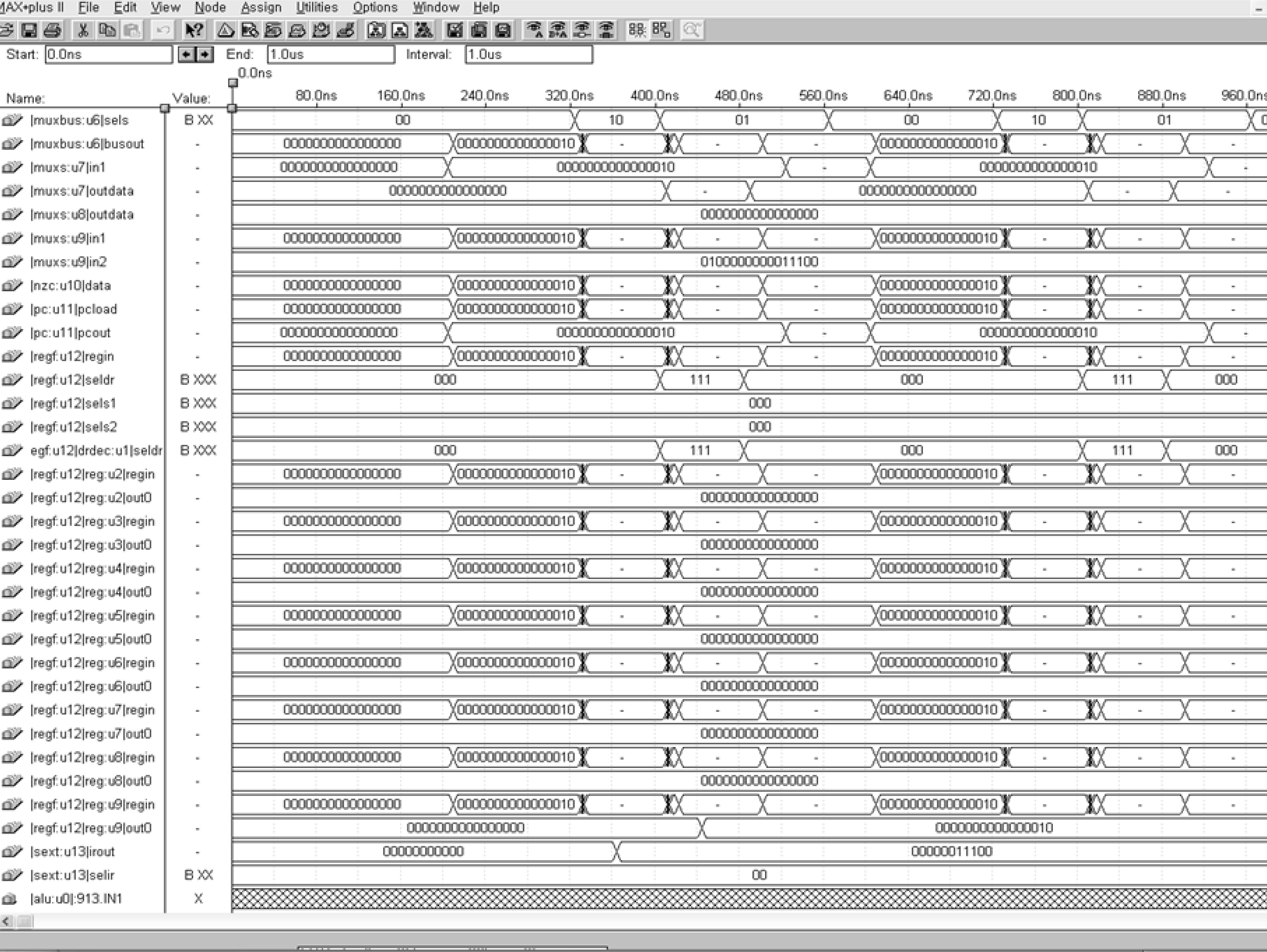
```
library ieee;
use ieee.std_logic_1164.all;

entity sext is

	port(	irout:in std_logic_vector(10 downto 0);
			selir:in std_logic_vector(1 downto 0);
			sext:out std_logic_vector(15 downto 0));
end entity sext;

architecture rtl of sext is
begin

process(selir)
begin

case selir is
	when "00"=>
		case irout(4) is
			when '1'=>
				sext<="11111111111"&irout(4 downto 0);
			when others =>
				sext<="00000000000"&irout(4 downto 0);
		end case;
	when "01"=>
		case irout(5) is
			when '1'=>
				sext<="1111111111"&irout(5 downto 0);
			when others =>
				sext<="0000000000"&irout(5 downto 0);
		end case;
	when "10"=>
		case irout(8) is
			when '1'=>
				sext<="1111111"&irout(8 downto 0);
			when others =>
				sext<="0000000"&irout(8 downto 0);
		end case;
	when others=>
		case irout(10) is
			when '1'=>
				sext<="11111"&irout(10 downto 0);
			when others =>
				sext<="00000"&irout(10 downto 0);
		end case;
end case;
end process;
end architecture rtl;
```



Ref: 0.0ns Time: 0.0ns Interval: 0.0ns

0.0ns 80.0ns 160.0ns 240.0ns 320.0ns 400.0ns 480.0ns 560.0ns 640.0ns 720.0ns 800.0ns 880.0ns 960.0ns

Name:	Value:	
clk	0	
clear	1	
busstate	-	0000000000000000, 0000000000000010, -, -, -, 0000000000000010, -, -, -
|alu:u0|s1	-	0000000000000000, -, 0000000000000000, -, -
|alu:u0|s2	-	0000000000000000
|alu:u0|controls	B 00	00
|control:u1|irout	-	0000000000000000, 0100000000011100
|control:u1|alucontrol	B 00	00
|control:u1|selbus	B 00	00, 10, 01, 00, 10, 01
|control:u1|sexts	B 00	00
|control:u1|sels1	B 000	000
|control:u1|sels2	B 000	000
|control:u1|seldr	B 000	000, 111, 000, 111, 000
|control:u1|state	B 0000	0000, 0001, 0010, 0011, 0100, 0101, 0001, 0010, 0011, 0100, 0101, 0001
|ir:u2|irin	-	0000000000000000, 0000000000000010, -, -, -, 0000000000000010, -, -, -
|ir:u2|irout	-	0000000000000000, 0100000000011100
|mar:u3|marin	-	0000000000000000, 0000000000000010, -, -, -, 0000000000000010, -, -, -
|mar:u3|marout	-	0000000000000000
|mdr:u4|mdrout	-	0000000000000000, 0100000000011100
|mem:u5|address	-	0000000000000000
|mem:u5|wdata	-	0000000000000000, 0100000000011100
|mem:u5|rdata	-	0100000000011100
|muxbus:u6|pcout	-	0000000000000000, 0000000000000010, -, 0000000000000010, -
|muxbus:u6|mdrout	-	0000000000000000, 0100000000011100
|muxbus:u6|sels	B 00	00, 10, 01, 00, 10, 01
|muxbus:u6|busout	-	0000000000000000, 0000000000000010, -, -, -, 0000000000000010, -, -, -
|muxs:u7|in1	-	0000000000000000, 0000000000000010, -, 0000000000000010, -
|muxs:u7|outdata	-	0000000000000000, -, 0000000000000000, -, -
|muxs:u8|outdata	-	0000000000000000
|muxs:u9|in1	-	0000000000000000, 0000000000000010, -, -, -, 0000000000000010, -, -, -
|muxs:u9|in2	-	0100000000011100
|nzc:u10|data	-	0000000000000000, 0000000000000010, -, -, -, 0000000000000010, -, -, -
|pc:u11|pcload	-	0000000000000000, 0000000000000010, -, -, -, 0000000000000010, -, -, -
|pc:u11|pcout	-	0000000000000000, 0000000000000010, -, 0000000000000010, -



```
Start: 0.0ns        End: 1.0us        Interval: 1.0us
                    0.0ns
Name:                     Value:   80.0ns 160.0ns 240.0ns 320.0ns 400.0ns 480.0ns 560.0ns 640.0ns 720.0ns 800.0ns 880.0ns 960.0ns

|muxbus:u6|sels           B XX     00 | 10 | 01 | 00 | 10 | 01 | 0
|muxbus:u6|busout         -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|muxs:u7|in1              -        0000000000000000 | 0000000000000010 | - | 0000000000000010 | -
|muxs:u7|outdata          -        0000000000000000 | - | 0000000000000000 | - | -
|muxs:u8|outdata          -        0000000000000000
|muxs:u9|in1              -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|muxs:u9|in2              -        0100000000011100
|nzc:u10|data             -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|pc:u11|pcload            -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|pc:u11|pcout             -        0000000000000000 | 0000000000000010 | - | 0000000000000010 | -
|regf:u12|regin           -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|regf:u12|seldr           B XXX    000 | 111 | 000 | 111 | 000
|regf:u12|sels1           B XXX    000
|regf:u12|sels2           B XXX    000
egf:u12|drdec:u1|seldr    B XXX    000 | 111 | 000 | 111 | 000
|regf:u12|reg:u2|regin    -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|regf:u12|reg:u2|out0     -        0000000000000000
|regf:u12|reg:u3|regin    -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|regf:u12|reg:u3|out0     -        0000000000000000
|regf:u12|reg:u4|regin    -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|regf:u12|reg:u4|out0     -        0000000000000000
|regf:u12|reg:u5|regin    -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|regf:u12|reg:u5|out0     -        0000000000000000
|regf:u12|reg:u6|regin    -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|regf:u12|reg:u6|out0     -        0000000000000000
|regf:u12|reg:u7|regin    -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|regf:u12|reg:u7|out0     -        0000000000000000
|regf:u12|reg:u8|regin    -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|regf:u12|reg:u8|out0     -        0000000000000000
|regf:u12|reg:u9|regin    -        0000000000000000 | 0000000000000010 | - | - | - | 0000000000000010 | - | - | -
|regf:u12|reg:u9|out0     -        0000000000000000 | 0000000000000010
|sext:u13|irout           -        00000000000 | 00000011100
|sext:u13|selir           B XX     00
|alu:u0|:913.IN1          X
```