sit.vhd

```
1  library ieee;
2  use ieee.std_logic_1164.all, ieee.std_logic_unsigned.all;
3  entity sit is
4  port(en,clk : in std_logic;
5       cin:in std_logic_vector(7 downto 0);
6       Z: out std_logic_vector(7 downto 0));
7  end sit;
8  architecture behave of sit is
9  signal tem: std_logic_vector(7 downto 0);
10 begin
11     process(clk,en,cin)
12         begin
13         if(en='1') then
14             if(clk'event and clk='0') then
15                 tem <= not tem;
16             end if;
17         end if;
18         Z <=tem;
19         end process;
20 end;
```

pcjoq.vhd

```
1  library ieee;
2  use ieee.std_logic_1164.all, ieee.std_logic_unsigned.all;
3
4  entity pcjoq is
5  port(ld, inc, clk : in std_logic;
6       a : in std_logic_vector(7 downto 0);
7       c : out std_logic_vector(7 downto 0));
8  end pcjoq;
9
10 architecture behave of pcjoq is
11 signal tem : std_logic_vector(7 downto 0);
12 begin
13     process(ld, inc, clk)
14     begin
15         if (clk'event and clk = '0') then
16             if inc = '1' then
17                 if tem = "11111111" then
18                     tem <= "00000000";
19                 else
20                     tem <= tem + 1;
21                 end if;
22             elsif ld = '1' then
23                 tem <= A;
24             end if;
25         end if;
26     end process;
27     c <= tem;
28 end;
```

tyjoq.vhd

```
1  library ieee;
2  use ieee.std_logic_1164.all, ieee.std_logic_unsigned.all;
3
4  entity tyjoq is
5  port(we,clk : in std_logic;
6       ra,wa : in std_logic_vector(1 downto 0);
7       i : in std_logic_vector(7 downto 0);
8       ao,bo : out std_logic_vector(7 downto 0));
9  end tyjoq;
10
11 architecture behave of tyjoq is
12 signal a,b,c: std_logic_vector(7 downto 0);
13 begin
14     process(we, clk)
15     begin
16         if (we = '1') then
17             if ra="00" then ao<=a;
18             elsif ra="01" then ao<=b;
19             elsif ra="10" then ao<=c;
20             else ao<="ZZZZZZZZ";
21             end if;
22             if wa="00" then bo<=a;
23             elsif wa="01" then bo<=b;
24             elsif wa="10" then bo<=c;
25             else bo<="ZZZZZZZZ";
26             end if;
27         elsif (clk'event and clk='0') then
28             if wa="00" then a<=i;
29             elsif wa="01" then b<=i;
30             elsif wa="10" then c<=i;
31             end if;
32         end if;
33     end process;
34 end;
```

lpm_ram_10.bdf

Parameter	Value	Type
LPM_ADDRESS_CONTROL	REGISTERED	String
LPM_FILE	/ram rw.mif	String
LPM_INDATA	REGISTERED	String
LPM_NUMWORDS	256	Unsigned Integer
LPM_OUTDATA	UNREGISTERED	String
LPM_WIDTH	8	Unsigned Integer
LPM_WIDTHAD	8	Unsigned Integer