实验四预习报告

201914020128 计科 1903　陈旭

- **SM**

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

entity sm is
    port(
        clk, Sm_en: in std_logic;
        z: out std_logic
    );
end sm;

architecture sm of sm is
    signal sm:std_logic:='0';
    begin
        process(clk, Sm_en)
        begin
            if(clk'event and clk='0') then
                if(Sm_en='1') then
                    z<=not sm;
                    sm<=not sm;
                else
                    z<=sm;
                end if;
            else
                sm<=sm;
            end if;
        end process;
    end architecture sm;
```

- **指令计数器(PC)**

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

entity program_counter is
```

```
    port(
        ldpc, inpc, clk:in std_logic;
        a:in std_logic_vector(7 downto 0);
        c:out std_logic_vector(7 downto 0)
    );
end program_counter;

architecture program_counter of program_counter is
    signal adress: std_logic_vector(7 downto 0):="00000000";
    begin
        process(ldpc, inpc, clk, a)
        begin
            if(clk'event and clk='0') then
                if inpc='1' and ldpc='0' then
                    adress<=adress+"00000001";
                elsif(inpc='0' and ldpc='1') then
                    adress<=a;
                else
                end if;
            else
            end if;
        end process;
        c<=adress;
    end architecture program_counter;
```

- **通用寄存器组**

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

entity general_registers is
    port(
        we, clk:in std_logic;
        raa, rwba:in std_logic_vector(1 downto 0);
        i:in std_logic_vector(7 downto 0);
        s, d:out std_logic_vector(7 downto 0)
    );
end general_registers;

architecture general_registers of general_registers is
    signal a:std_logic_vector(7 downto 0);
    signal b:std_logic_vector(7 downto 0);
    signal c:std_logic_vector(7 downto 0);
    signal temp:std_logic_vector(7 downto 0);
```

```
begin
    process(we, clk, raa, rwba, i)
    begin
        if(we='0') then
            if(clk'event and clk='0') then
                if(rwba="00") then
                    a<=i;
                elsif(rwba="01") then
                    b<=i;
                elsif(rwba="10") then
                    c<=i;
                else
                    c<=i;
                end if;
            else
            end if;
        if(raa="00") then
            s<=a;
        elsif(raa="01") then
            s<=b;
        elsif(raa="10") then
            s<=c;
        else
            s<=c;
        end if;
        if(rwba="00") then
            d<=a;
        elsif (rwba="01") then
            d<=b;
        elsif (rwba="10") then
            d<=c;
        else
            d<=c;
        end if;
    end process;
end general_registers;
```

- **RAM**

Parameter	Value	Type
LPM_ADDRESS_CONTRO	REGISTERED	String
LPM_FILE	./ram.mif	String
LPM_INDATA	REGISTERED	String
LPM_NUMWORDS	256	Unsigned Integer
LPM_OUTDATA	UNREGISTERE	String
LPM_WIDTH	8	Unsigned Integer
LPM_WIDTHAD	8	Unsigned Integer