**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;

            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**

