**实验二预习报告**

**201914020128 计科1903 陈旭**

* **8重3-1多路复用器**

library ieee;

use ieee.std\_logic\_1164.all;

entity mux\_8\_3\_1 is

    port(

        inputs0: in std\_logic\_vector(7 downto 0);

        inputs1: in std\_logic\_vector(7 downto 0);

        inputs2: in std\_logic\_vector(7 downto 0);

        selinputs: in std\_logic\_vector(1 downto 0);

        outputs: out std\_logic\_vector(7 downto 0)

    );

end mux\_8\_3\_1;

architecture mux\_8\_3\_1 of mux\_8\_3\_1 is

    begin

        outputs<=inputs0 when selinputs="00" else

        inputs1 when selinputs="01" else

        inputs2 when selinputs="10"else

        "00000000";

end mux\_8\_3\_1;

* **参数化8\_1多路复用器**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity generic1 is

    generic (n:integer:=3);

    port(

        genein: in std\_logic\_vector(2\*\*n-1 downto 0);

        genesel: in std\_logic\_vector(n-1 downto 0);

        geneout: out std\_logic

    );

    end generic1;

architecture generic1 of generic1 is

    begin

        geneout<=genein(conv\_integer(genesel));

    end generic1;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity generic\_mux\_8\_1 is

    port(

        inputs: in std\_logic\_vector(7 downto 0);

        output: out std\_logic;

        selects: in std\_logic\_vector(2 downto 0)

    );

end generic\_mux\_8\_1;

architecture generic\_mux\_8\_1 of generic\_mux\_8\_1 is

    component generic1 is

        generic (n:integer:=3);

        port(

            genein: in std\_logic\_vector(2\*\*n-1 downto 0);

            genesel: in std\_logic\_vector(n-1 downto 0);

            geneout: out std\_logic

        );

    end component;

    begin

        g1:generic1 generic map(n=>3) port map(genein=>inputs, geneout=>output, genesel=>selects);

    end generic\_mux\_8\_1;

* **4位行波进位加法器**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity rca\_4 is

    port(

        x, y: in std\_logic\_vector(3 downto 0);

        cin: in std\_logic;

        s: out std\_logic\_vector(3 downto 0);

        cout: out std\_logic

    );

end rca\_4;

architecture rca\_4 of rca\_4 is

    signal sum: std\_logic\_vector(4 downto 0);

    begin

        sum<=('0'&x)+('0'&y)+("0000"&cin);

        cout <= sum(4);

        s<=sum(3 downto 0);

    end rca\_4;

* **4位先行进位加法器**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity aca\_4 is

    port(

        x, y: in std\_logic\_vector(3 downto 0);

        cin: in std\_logic;

        sout: out std\_logic\_vector(4 downto 0)

    );

    end aca\_4;

architecture aca\_4 of aca\_4 is

    signal cout: std\_logic\_vector(2 downto 0);

    begin

        cout(0)<=(x(0) and y(0))or(cin and (x(0) or y(0)));

        cout(1)<=(x(1) and y(1))or(cout(0) and (x(1) or y(1)));

        cout(2)<=(x(2) and y(2))or(cout(1) and (x(2) or y(2)));

        sout(4)<=(x(3) and y(3))or(cout(2) and (x(3) or y(3)));

        sout(0)<=x(0) xor y(0) xor cin;

        sout(1)<=x(1) xor y(1) xor cout(0);

        sout(2)<=x(2) xor y(2) xor cout(1);

        sout(3)<=x(3) xor y(3) xor cout(2);

    end aca\_4;