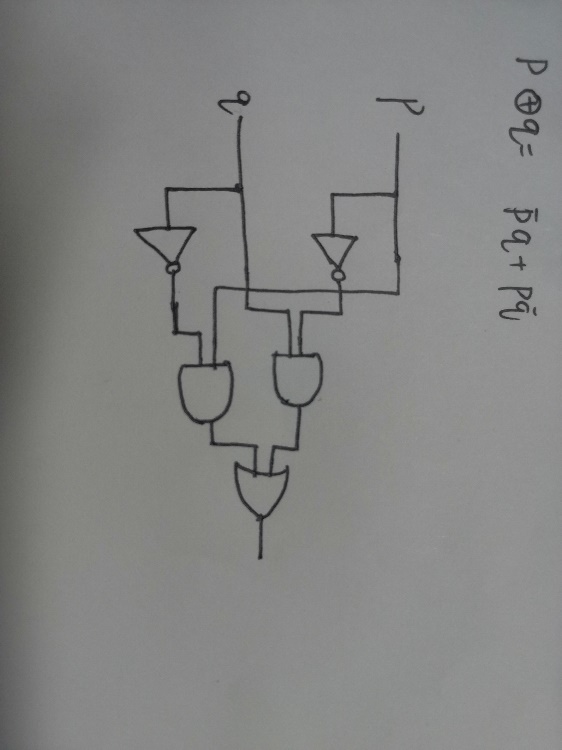
**实验一预习报告**

**201914020128 计科1903 陈旭**

* **异或门的原理图与VHDL程序**

1. **异或门的原理图**

****

1. **异或门的VHDL程序**

**library ieee;**

**use ieee.std\_logic\_1164.all;**

**entity twoin\_xor is**

**port(**

**a,b: in std\_logic;**

**c: out std\_logic**

**);**

**end entity twoin\_xor;**

**architecture twoin\_xor of twoin\_xor is**

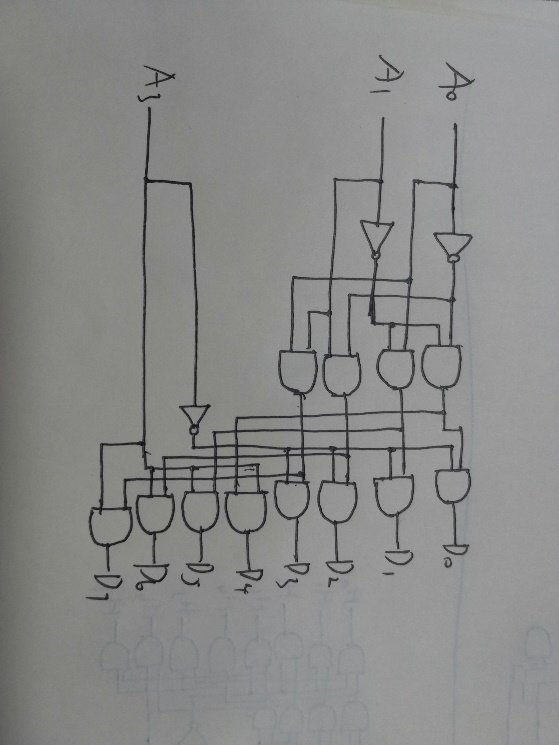
**begin**

**c <= a xor b;**

**end architecture twoin\_xor;**

* **3-8译码器的原理图与VHDL程序**

1. **3-8译码器的原理图**

****

1. **3-8译码器的VHDL程序**

**library ieee;**

**use ieee.std\_logic\_1164.all;**

**entity three\_eightdecoder is**

**port(**

**a, b, c: in std\_logic;**

**x: out std\_logic\_vector(7 downto 0)**

**);**

**end entity three\_eightdecoder;**

**architecture three\_eightdecoder of three\_eightdecoder is**

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

**begin**

**s <= a&b&c;**

**with s select**

**x <= "00000001" when "000",**

**"00000010" when "001",**

**"00000100" when "010",**

**"00001000" when "011",**

**"00010000" when "100",**

**"00100000" when "101",**

**"01000000" when "110",**

**"10000000" when "111";**

**end architecture three\_eightdecoder;**

* **模型机指令译码器的VHDL程序**

**library ieee;**

**use ieee.std\_logic\_1164.all;**

**entity cmddecoder is**

**port(**

**cmdar\_code: in std\_logic\_vector(7 downto 0);**

**ison: in std\_logic;**

**mov1, mov2, mov3, add, sub, or1, not1, rsr, rsl, jmp, jz, jc, in1, out1, nop, halt: out std\_logic**

**);**

**end entity cmddecoder;**

**architecture cmddecoder of cmddecoder is**

**signal cmdcode: std\_logic\_vector(3 downto 0);**

**signal r1code,r2code: std\_logic\_vector(1 downto 0);**

**begin**

**cmdcode <= cmdar\_code(3 downto 0);**

**r1code <= cmdar\_code(5 downto 4);**

**r2code <= cmdar\_code(7 downto 6);**

**process**

**begin**

**mov1 <= '0';**

**mov2 <= '0';**

**mov3 <= '0';**

**add <= '0';**

**sub <= '0';**

**or1 <= '0';**

**not1 <= '0';**

**rsr <= '0';**

**rsl <= '0';**

**jmp <= '0';**

**jz <= '0';**

**jc <= '0';**

**in1 <= '0';**

**out1 <= '0';**

**nop <= '0';**

**halt <= '0';**

**if ison='1' then**

**if cmdcode="1111" then**

**if r1code="11" then**

**mov2 <= '1';**

**elsif r2code="11" then**

**mov3 <= '1';**

**else**

**mov1 <= '1';**

**end if;**

**elsif cmdcode="1001" then**

**add <= '1';**

**elsif cmdcode="0110" then**

**sub <= '1';**

**elsif cmdcode="1011" then**

**or1 <= '1';**

**elsif cmdcode="0101" then**

**not1 <= '1';**

**elsif cmdcode="1010" then**

**if r2code="00" then**

**rsr <= '1';**

**elsif r2code="11" then**

**rsl <= '1';**

**end if;**

**elsif cmdcode="0011" then**

**if r2code="00" then**

**jmp <= '1';**

**elsif r2code="01" then**

**jz <= '1';**

**elsif r2code="10" then**

**jc <= '1';**

**end if;**

**elsif cmdcode="0010" then**

**in1 <= '1';**

**elsif cmdcode="0100" then**

**out1 <= '1';**

**elsif cmdcode="0111" then**

**nop <= '1';**

**elsif cmdcode="1000" then**

**halt <= '1';**

**end if;**

**end if;**

**end process;**

**end architecture cmddecoder;**