UNIVERSITY OF CALIFORNIA, RIVERSIDE

Department of Computer Science and Engineering

Department of Electrical Engineering

CS/EE120A – Logic Design

Midterm 2

May 16, 2001

_____ / 20

Name: Solution Key **Student ID#:**____________________

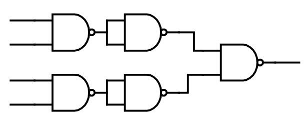
Please print legibly

Lab Section: 21 (TR 2-5):______ **22** (MW 6-9):_____

(Numbers in parenthesis denote total possible points for question.)

1. Construct a 4-input NAND gate circuit using only 2-input NAND gates. (4)

Answer

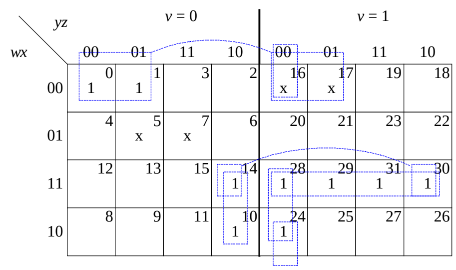


2. Use the K-map method to reduce the function $f(v,w,x,y,z)$ with the following 1-minterms and don’t care minterms:

   1-minterms: 0,1,10,14,24,28,29,30,31
   don’t cares: 5,7,16,17

   Draw the K-map. Derive (a) the PI list, (b) the EPI list, (c) the solution(s). (4)

Answer



(a) PI = w′x′y′, v′wyz′, wxyz′, vwx, vwy′z′, *vx′y′z′*
(b) EPI = *w′x′y′*, *v′wyz′*, *vwx*
(c) Solutions:
    (i) $f = w'x'y' + v'wyz' + vwx + vwy'z'$
    (ii) $f = w'x'y' + v'wyz' + vwx + vx'y'z'$

2 points for a) and b)

2 points for c)

3. Using the tabulation (Quine-McCluskey) method, derive the table(s) for the function $f(v,w,x,y,z) = \Sigma(0,1,10,14,16,17,24,28,29,30,31)$. (4)

Answer

Group ID	Subcube Minterms	Subcube Value					Subcube Covered
		v	*w*	*x*	*y*	*z*	
G_0	m_0	0	0	0	0	0	✓
G_1	m_1	0	0	0	0	1	✓
	m_{16}	1	0	0	0	0	✓
G_2	m_{10}	0	1	0	1	0	✓
	m_{17}	1	0	0	0	1	✓
	m_{24}	1	1	0	0	0	✓
G_3	m_{14}	0	1	1	1	0	✓
	m_{28}	1	1	1	0	0	✓
G_4	m_{29}	1	1	1	0	1	✓
	m_{30}	1	1	1	1	0	✓
G_5	m_{31}	1	1	1	1	1	✓

1 point for this table

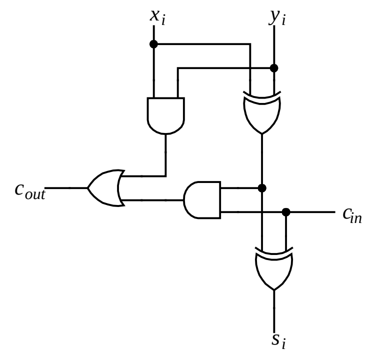
Group ID	Subcube Minterms	Subcube Value					Subcube Covered
		v	*w*	*x*	*y*	*z*	
G_0	$m_{0,1}$	0	0	0	0	-	✓
	$m_{0,16}$	-	0	0	0	0	✓
G_1	$m_{1,17}$	-	0	0	0	1	✓
	$m_{16,17}$	1	0	0	0	-	✓
	$m_{16,24}$	1	-	0	0	0	
G_2	$m_{10,14}$	0	1	-	1	0	
	$m_{24,28}$	1	1	-	0	0	
G_3	$m_{14,30}$	-	1	1	1	0	
	$m_{28,29}$	1	1	1	0	-	✓
	$m_{28,30}$	1	1	1	-	0	✓
G_4	$m_{29,31}$	1	1	1	-	1	✓
	$m_{30,31}$	1	1	1	1	-	✓

1 point for this table

Group ID	Subcube Minterms	Subcube Value					Subcube Covered
		v	*w*	*x*	*y*	*z*	
G_0	$m_{0,1,16,17}$	-	0	0	0	-	
G_3	$m_{28,29,30,31}$	1	1	1	-	-	

2 points for this table

4. Write the <u>structural</u> VHDL code for the full adder circuit below. You need to include both the entity and architecture. The components AND2, OR2, and XOR2 are already defined. (4)



Answer

```
library IEEE;
use IEEE.std_logic_1164.all;

entity Full_Adder is
        port (   xi:  in STD_LOGIC;
                 yi:  in STD_LOGIC;
                 cin: in STD_LOGIC;
                 si:  out STD_LOGIC;
                 cout out STD_LOGIC);
end Full_Adder;

architecture Full_Adder_structural of Full_Adder is
        signal XY, XXORY, XXORYCIN: STD_LOGIC;
        component AND2 port (I0,I1: in STD_LOGIC; O: out STD_LOGIC);
                             end component;
        component XOR2 port (I0,I1: in STD_LOGIC; O: out STD_LOGIC);
                             end component;
        component OR2 port (I0,I1: in STD_LOGIC; O: out STD_LOGIC);
                             end component;

begin
        U1: AND2 port map (xi, yi, XY);
        U2: XOR2 port map (xi, yi, XXORY);
        U3: XOR2 port map (XXORY, cin, si);
        U4: AND2 port map (XXORY, cin, XXORYCIN);
        U5: OR2  port map (XY, XXORYCIN, cout);
end Full_Adder_structural;
```

5. Write the <u>dataflow</u> VHDL code for the full adder circuit in question 4 above using concurrent signal assignment statements. You need to include both the entity and architecture. (4)

Answer

```
library IEEE;
use IEEE.std_logic_1164.all;

entity Full_Adder is
       port (   xi:  in STD_LOGIC;
                yi:  in STD_LOGIC;
                cin: in STD_LOGIC;
                si:  out STD_LOGIC;
                cout out STD_LOGIC);
end Full_Adder;

architecture Full_Adder_dataflow of Full_Adder is
       signal XY, XXORY, XXORYCIN: STD_LOGIC;

begin
       XY <= xi AND yi;
       XXORY <= xi XOR yi;
       si <= XXORY XOR cin;
       XXORYCIN <= XXORY AND cin;
       cout <= XY OR XXORYCIN;
end Full_Adder_dataflow;
```