Nov. 28, 2010

COMPUTER ENGINEERING DEPARTMENT

COE 561

Digital System Design and Synthesis

Major Exam I

(Open Book Exam)

First Semester (101)

Time: 1:00-3:30 PM

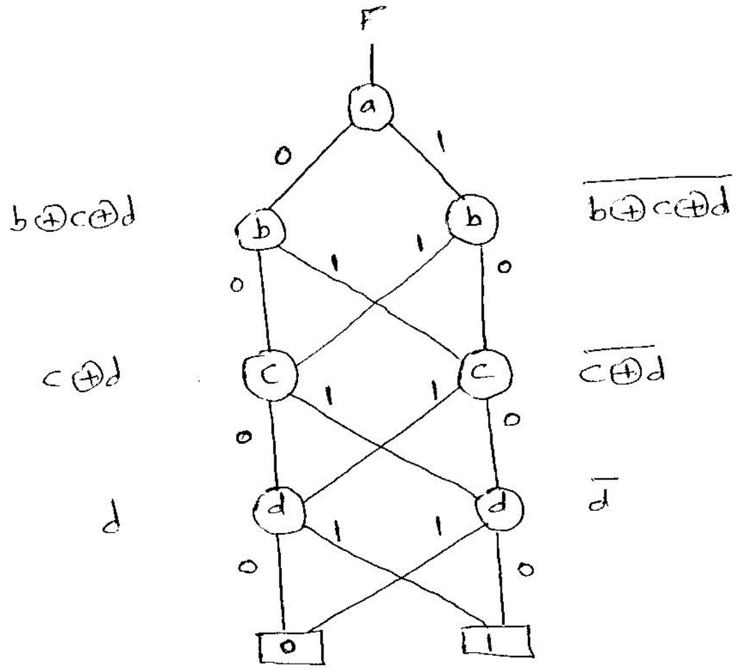
Student Name : _KEY____

Student ID.   : ____

Question	Max Points	Score
Q1	15	
Q2	15	
Q3	10	
Q4	20	
Q5	20	
Q6	20	
Total	100	

[15 Points]

(Q1) Draw the ROBDD for the function F=a⊕b⊕c⊕d, with the variable ordering {a, b, c, d}. How can we easily obtain the ROBDD for $\bar{F}$ from the ROBDD for F? Don't draw the ROBDD for $\bar{F}$, just explain.



The ROBDD for $\bar{F}$ can be easily obtained by just interchanging the 0 & 1 leaf vertices.

[15 Points]

(Q2) Write an algorithm, called **ROBDD**, that receives a function F and a variable ordering and constructs and ROBDD for the input function. Explain clearly the terminal cases and the structure of the tables you will use in your algorithm.

```
ROBDD(F){
    If (terminal case)
        return (r = trivial result)
    else {
        if (computed table has entry (F, r) )
            return (r from computed table)
        else {
            x is top variable of F
            t = ROBDD(F_x)
            e = ROBDD(F_x')
            if ( t == e) return (t)
            r = find_or_add_unique_table (x, t, e)
            Update computed table with (F, r)
            return (r)
        }
    }
}
```

Terminal cases are when F is a single literal i.e. x or x’ or 0 or 1.
The unique table contains a key for a vertex of an ROBDD where the key is a triple of variable, identifiers of right and left children.
The computed table stores a function and its identifier in the form (F, r) to improve the performance of the algorithm.

[10 Points]

(Q3) Consider the function $F(A,B,C,D) = AB + A\overline{C} + A\overline{D} + \overline{C}D + \overline{A}\overline{C} + \overline{A}\overline{B} + \overline{A}D + \overline{A}B\overline{D}$. Using recursive paradigm, determine if the function F is **tautology** or not. You need to choose the right variable for expansion to minimize computations.

Since the cover is negative unate with respect to C, it is sufficient to show that F_C is Tautology, since $F_{\overline{C}} \supseteq F_C$.

Thus, $F_C = AB + A\overline{D} + \overline{A}\,\overline{B} + \overline{A}D + \overline{A}B\overline{D}$

$$= A[B + \overline{D}] + \overline{A}[\overline{B} + D + B\overline{D}]$$

Since F_{CA} is unate $\Rightarrow$ Not Tautology.

Thus, F is not Tautology.

[20 Points]

(Q4) Consider the two Boolean functions F_1 and F_2 given below:

$$F_1(A,B) = A \oplus B$$

$$F_2(C,D) = C \oplus D$$

Draw the **ITE DAG** for the function $F_1 \oplus F_2$ using the variable order {A, B, C, D}. Show all the details of your solution using ITE procedure including the resulting unique table and computed table.

$f \oplus g = ITE(A \oplus B, \overline{C \oplus D}, C \oplus D)$

- $x = A$

$t = ITE(\overline{B}, \overline{C \oplus D}, C \oplus D)$

- $x = B$

$t = ITE(0, \overline{C \oplus D}, C \oplus D)$

- $x = C$

$t = ITE(0, D, \overline{D}) = \overline{D}$ (trivial case)

we assign id = 3 $\Rightarrow t = 3$

$e = ITE(0, \overline{D}, D) = D$ (trivial case)

we assign id = 4 $\Rightarrow e = 4$

since $t \neq e$, we add the entry (C, 3, 4)

in the unique table with id = 5

$\Rightarrow t = 5$

we add an entry in the computed table with

$\{(0, \overline{C \oplus D}, C \oplus D), 5\}$.

$e = ITE(1, \overline{C \oplus D}, C \oplus D)$

- $x = C$

$t = ITE(1, D, \overline{D}) = D$ (trivial case)

$\Rightarrow t = 4$

$e = ITE(1, \overline{D}, D) = \overline{D}$ (trivial case)

$\Rightarrow e = 3$

since $t \neq e$, we add the entry (C, 4, 3)

in the unique table with id = 6

$\Rightarrow e = 6$

we add an entry in the computed table with
$\{(1, \overline{C \oplus D}, C \oplus D), 6\}$.

Since $t \neq e$, we add an entry $(B, 5, 6)$
in the unique table with $id = 7$.

$\Rightarrow t = 7$

we add an entry in the computed table with
$\{(\overline{B}, \overline{C \oplus D}, C \oplus D), 7\}$.

$e = ITE(B, \overline{C \oplus D}, C \oplus D)$

- $x = B$

$t = (1, \overline{C \oplus D}, C \oplus D) = 6$ from computed table

$e = (0, \overline{C \oplus D}, C \oplus D) = 5$ from computed table

Since $t \neq e$, we add an entry $(B, 6, 5)$ in
the unique table with $id = 8$.

$\Rightarrow e = 8$

we add an entry in the computed table with
$\{(B, \overline{C \oplus D}, C \oplus D), 8\}$.

Since $t \neq e$, we add the entry $(A, 7, 8)$ to the
unique table with $id = 9$.

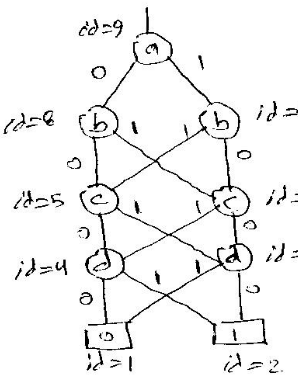
Unique Table:

id	var	H	L
3	D	1	2
4	D	2	1
5	C	3	4
6	C	4	3
7	B	5	6
8	B	6	5
9	A	7	8

Computed Table:

f	g	h	id
0	$\overline{C \oplus D}$	$C \oplus D$	5
1	$\overline{C \oplus D}$	$C \oplus D$	6
$\overline{B}$	$\overline{C \oplus D}$	$C \oplus D$	7
B	$\overline{C \oplus D}$	$C \oplus D$	8

ITE DAG:



[20 Points]

(Q5) Consider the function $F(A,B,C,D) = \overline{A}\overline{C} + \overline{A}B + \overline{C}D + A\overline{D} + \overline{B}C$

(i) Compute the **complement** of the function using the recursive complementation procedure outlined in section 7.3.4.

(ii) **C**ompute all the **prime implicants** of the function using the method outlined in section 7.3.4.

(i)

$$F = \overline{A}\left[\overline{C} + B + \overline{C}D + \overline{B}C\right] + A\left[\overline{D} + \overline{C}D + \overline{B}C\right]$$

$$= \overline{A}\left[\overline{C}\,[1] + C\,[\underbrace{B+\overline{B}}_{=1}]\right] + A\left[\overline{C}\,[\underbrace{\overline{D}+D}_{=1}] + C\,[\overline{D}+\overline{B}]\right]$$

$$= \overline{A}\left[\overline{C}\,[1] + C\,[1]\right] + A\left[\overline{C}\,[1] + C\left[\overline{D}\,[1] + D\,[\overline{B}]\right]\right]$$

$$\Rightarrow \overline{F} = \overline{A}\left[\overline{C}\,[0] + C\,[0]\right] + A\left[\overline{C}\,[0] + C\left[\overline{D}\,[0] + D\,[B]\right]\right]$$

$$= ABCD$$

(ii) from part (i), we have

$$F = \bar{A}\,[\,\bar{C}\,[1] + C\,[1]\,] + A\,[\,\bar{C}\,[1] + C\,[\bar{D}+\bar{B}]\,]$$

prime implicants of $f_{\bar{A}\bar{C}} = 1$

prime implicants of $f_{\bar{A}C} = 1$

$\Rightarrow$ prime implicants of $f_{\bar{A}} = \text{scc}\,\{\bar{C}, C, 1\} = 1$

prime implicants of $f_{A\bar{C}} = 1$

prime implicants of $f_{AC} = \{\bar{B}, \bar{D}\}$

$\Rightarrow$ prime implicants of $f_A = \text{scc}\,\{\bar{C}, C\bar{B}, C\bar{D}, \bar{B}, \bar{D}\}$

$= \{\bar{C}, \bar{B}, \bar{D}\}$

$\Rightarrow$ prime implicants of $f = \text{scc}\,\{\bar{A}, A\bar{C}, A\bar{B}, A\bar{D}, \bar{C}, \bar{B}, \bar{D}\}$

$= \{\bar{A}, \bar{B}, \bar{C}, \bar{D}\}$

[20 Points]

(Q6) Consider the following given matrix representing a covering problem:

$$
\begin{array}{c}
 \\ r1 \\ r2 \\ r3 \\ r4 \\ r5 \\ r6 \\ r7
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} c1 & c2 & c3 & c4 & c5 & c6 & c7 & c8 \end{array} \\
\begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

Find a **minimum cover** using **EXACT_COVER** procedure. Show all the details of the algorithm. Assume the following order in branching selection when needed: C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8. Propose two ideas that can be employed to make the **EXACT_COVER** procedure execute efficiently in general.

There are no essential columns and no row dominance or column dominance. Thus, we select c1 and call exact_cover with $x = (1,0,0,0,0,0,0,0)$ and $b = (1,1,1,1,1,1,1,1)$ and the matrix:

	c2	c3	c4	c5	c6	c7	c8
r4	1	0	0	1	0	1	0
r5	1	0	1	0	0	0	1
r6	1	1	0	0	1	0	0

Since c2 dominates all other columns, they get removed and c2 becomes essential and is selected. Since the matrix has no rows, then $x = (1,1,0,0,0,0,0,0)$ and $b = (1,1,0,0,0,0,0,0)$

Next, exact_cover is called with c1 not selected with $x = (0,0,0,0,0,0,0,0)$ and $b = (1,1,0,0,0,0,0,0)$ and the matrix :

	c2	c3	c4	c5	c6	c7	c8
r1	0	1	0	0	0	0	1
r2	0	0	1	0	0	1	0
r3	0	0	0	1	1	0	0
r4	1	0	0	1	0	1	0
r5	1	0	1	0	0	0	1
r6	1	1	0	0	1	0	0
r7	1	0	0	0	0	0	0

c2 is essential and is selected and we obtain the reduced matrix :

	c3	c4	c5	c6	c7	c8
r1	1	0	0	0	0	1
r2	0	1	0	0	1	0
r3	0	0	1	1	0	0

c3 dominates c8 $\Rightarrow$ c8 is removed
c4 dominates c7 $\Rightarrow$ c7 is removed
c5 dominates c6 $\Rightarrow$ c6 is removed

Thus, c3, c4, and c5 become essential and are selected.

Since the matrix has no rows, then $x = (0,1,1,1,1,0,0,0)$. Since $|x| > |b|$, the final returned solution is $(1,1,0,0,0,0,0,0)$.

Two approaches to mak exact_cover efficient :

1. Start with best solution based on a heuristic algorithm for solving the covering problem.
2. Select the branching column with the largest number of 1's in the matrix.