# UNIVERSITY OF CALIFORNIA, RIVERSIDE

**Department of Computer Science and Engineering**
**Department of Electrical Engineering**
**CS/EE120A –Logic Design**
**Midterm 2**
**May 14, 2003**

| |
|---|
| 20 |

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

***Please print legibly***

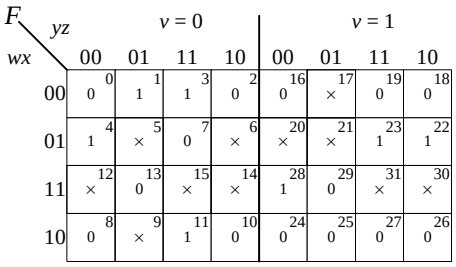
---

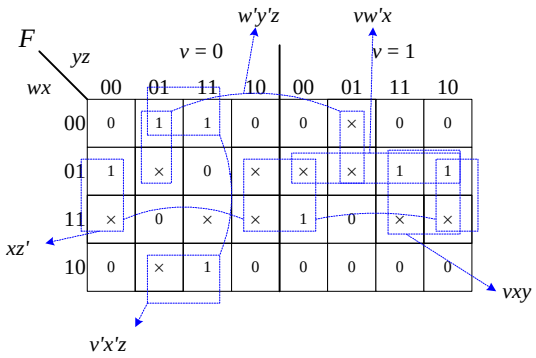
(Numbers in parenthesis denote total possible points for question.)

1. Given the following 1-minterms and "don't cares" for a 5-variable ($v$,$w$,$x$,$y$,$z$) function, list all the prime implicants and essential prime implicants. Use K-map to derive all minimized standard form solutions.

   1-minterms: $m_1$, $m_3$, $m_4$, $m_{11}$, $m_{22}$, $m_{23}$, $m_{28}$
   don't care minterms: $m_5$, $m_6$, $m_9$, $m_{12}$, $m_{14}$, $m_{15}$, $m_{17}$, $m_{20}$, $m_{21}$, $m_{30}$, $m_{31}$ (4)

### *Answer*




PIs: $xz'$, $w'y'z$, $vw'x$, $vxy$, $v'x'z$

| PI | Minterms covered | | | | | | |
|---|---|---|---|---|---|---|---|
| | $m_1$ | $m_3$ | $m_4$ | $m_{11}$ | $m_{22}$ | $m_{23}$ | $m_{28}$ |
| $xz'$ | | | ✓ | | ✓ | | ✓ |
| $w'y'z$ | ✓ | | | | | | |
| $vw'x$ | | | | | ✓ | ✓ | |
| $vxy$ | | | | | ✓ | ✓ | |
| $v'x'z$ | ✓ | ✓ | | ✓ | | | |

EPIs: $xz'$, $v'x'z$

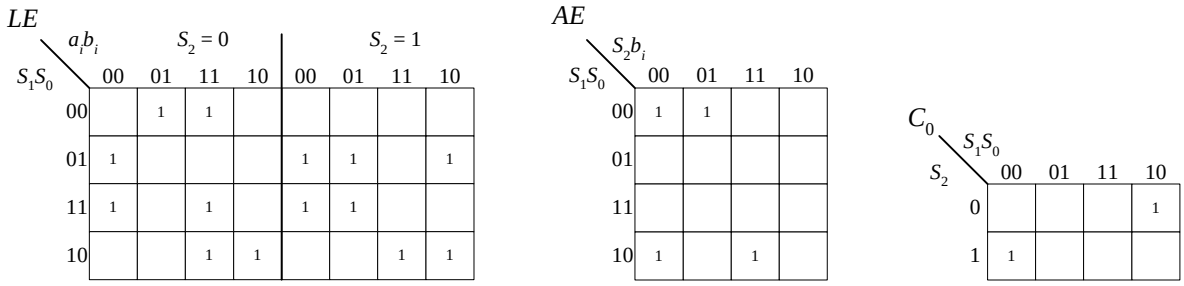
Solutions:

$F_1 = xz' + v'x'z + vxy$

and

$F_2 = xz' + v'x'z + vw'x$

2. Given the following K-maps for the LE, AE, and $C_0$ of an ALU, determine the ALU operations assigned to each of the select line combinations. Show how each operation is derived. (4)




***Answer***

| $S_2$ | $S_1$ | $S_0$ | *LE* | *AE* | $C_0$ | ALU Operation |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | $b_i$ | 1 | 0 | $B - 1$ |
| 0 | 0 | 1 | $a_i$ nor $b_i$ | 0 | 0 | *A* nor *B* |
| 0 | 1 | 0 | $a_i$ | $b_i'$ | 1 | $A - B$ |
| 0 | 1 | 1 | $a_i$ xnor $b_i$ | 0 | 0 | *A* xnor *B* |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | $a_i$ nand $b_i$ | 0 | 0 | *A* nand *B* |
| 1 | 1 | 0 | $a_i$ | $b_i$ | 0 | $A + B$ |
| 1 | 1 | 1 | $a_i'$ | 0 | 0 | $A'$ |

3. Write the complete VHDL code at the dataflow level for the AE circuit as given in question 2 above. (4)

***Answer***

```
LIBRARY ieee;
USE IEEE.std_logic_1164.all;

ENTITY AE IS PORT (
       S2, S1, S0: IN std_logic;    -- select lines
       Bi: IN std_logic;
       Yi: OUT std_logic);
END AE;

ARCHITECTURE Dataflow OF AE IS
BEGIN
       Yi <= ((NOT S2) AND (NOT S1) AND (NOT S0)) OR
                ((NOT S2) AND (S1) AND (NOT S0) AND (NOT Bi)) OR
                ((S2) AND (S1) AND (NOT S0) AND (Bi));

END Dataflow;
```

4. Design a circuit that inputs a four (4) bit number. The circuit outputs a 1 if the input number has an even number of zeros. Otherwise, it outputs a 0. The circuit must be as small as possible. Draw the final circuit. (4)

***Answer***

| F: wx \ yz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | | 1 | |
| 01 | | 1 | | 1 |
| 11 | 1 | | 1 | |
| 10 | | 1 | | 1 |

$$
\begin{aligned}
F &= w'x'y'z' + w'x'yz + w'xy'z + w'xyz' + wx'y'z + wx'yz' + wxy'z' + wxyz \\
&= w'x'(y'z' + yz) + wx(y'z' + yz) + w'x(y'z + yz') + wx'(y'z + yz') \\
&= w'x'(y \oplus z)' + wx(y \oplus z)' + w'x(y \oplus z) + wx'(y \oplus z) \\
&= (w'x' + wx)(y \oplus z)' + (w'x + wx')(y \oplus z) \\
&= (w \oplus x)'(y \oplus z)' + (w \oplus x)(y \oplus z) \\
&= (w \oplus x) \odot (y \oplus z) \\
&= (w \odot x) \odot (y \odot z)
\end{aligned}
$$

$(w \oplus x) \odot (y \oplus z) = 1 \Rightarrow (w \oplus x) = (y \oplus z)$.
If $(w \oplus x) = (y \oplus z)$ then certainly $(w \oplus x)' = (y \oplus z)'$
which is equal to $(w \odot x) = (y \odot z)$.
Hence $(w \oplus x) \odot (y \oplus z) = (w \odot x) \odot (y \odot z)$.

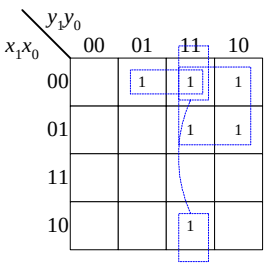
5. Design a subtractor circuit where both operands are two (2) bits wide. Design the circuit as one complete unit and not like what we did in class. That is, not as a single bit slice and then daisy chain the two bit slices together. The two 2-bit operands are interpreted as positive numbers. (This is equivalent to a 3-bit number where the leading bit is always a 0.) The result of the subtraction should be interpreted as a two's complement number and should be three bits wide. Derive the truth table, and the minimized standard form equations. You do not have to draw the circuit. (4)

***Answer***

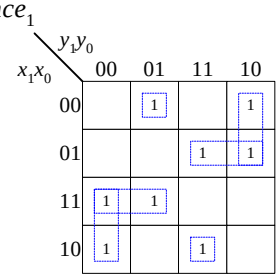
*Difference* = $x - y$

| $x_1$ | $x_0$ | $y_1$ | $y_0$ | *borrow* | *difference*$_1$ | *difference*$_0$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

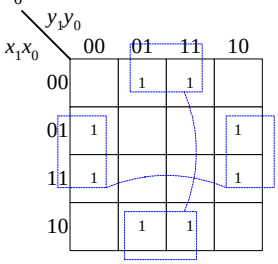
*borrow*




*difference*$_1$




*difference*$_0$




*borrow* = $x_1'y_1 + x_1'x_0'y_0 + x_0'y_1y_0$

*difference*$_1$ = $x_1x_0y_1' + x_1y_1'y_0' + x_1'x_0y_1 + x_1'y_1y_0' + x_1'x_0'y_1'y_0 + x_1x_0'y_1y_0$

*difference*$_0$ = $x_0'y_0 + x_0y_0'$