

CSE 460
Spring 2006 : Lockwood

Homework #2: Due Thursday, February 16, 2006
at start of class (2:30pm)

Name:	
-------	--

1. For the following function:

$$f(v, w, x, y, z) = \sum m(1, 4, 6, 9, 14, 17, 22, 27, 28) + d(12, 15, 20, 30, 31)$$

- (a) Use the tabular method to find the prime implicants of f .
- (b) Show the prime implicant chart for f . Identify the essential prime implicants.
- (c) Give a minimal sum of products expression for f .

2. Using any computer programming language that you desire, write a program which can generate an array of minterms from an expression of n -variables function .

- Your program should allow for functions with as many as 32 variables.
Assume that n is specified on the first line of the input file.
It should *not* be coded as a constant.
- Assume that $f(x_0, x_2, \dots, x_{n-1})$ is specified with terms that are:
true (1), false (0), or unspecified (-).
- For example
 - A 5-variable function, $f(x_0, x_1, \dots, x_4) = x_0x_1'x_4$ would be specified as:
5
10--1
 - Your program should generate a list of minterms like:
10001
10011
10101
10111
- Hint. If you happen to program the program in C++,
you *might* find the following helpful.

```
int n; // Number of variables in function

class sop_term {
private:
    unsigned int value; // up to 32-bits: 0=false, 1=true
    unsigned int mask; // up to 32-bits: 0=Care, 1=DontCare
};
```

- Submit your original source code.

- Submit the output of your program for the following 25-variable function:
25
11--00--1010-00001010---0
done

3. Extend the program so that it can generate minterms from a list of terms like the ones above. Assume that each additional term appears on new line. Assume that the end of the expression is denoted with an empty line.

- For example,

- For the function, $f(x_0, x_1, x_2, x_3, x_4) = x_0x_1'x_4 + x_1x_2x_3$, the given input of:

```
5
10--1
-111-
```

- Should produce output from your program of:

```
10001
10011
10101
10111
01110
01111
11110
11111
```

- Use your program to generate results for a 25-variable function that has the following input:

- 25
11--00--1010-00001010---0
01111111100000000--1-1111
done

4. Consider a function called `Is_Equal` that compares two SOP terms.

- The function should return true if and only if all of the bits in the term have identical mask bits (care or don't care) and each of the value bits that are in the care set match.

- For Example, the following functions return:

```
Is_Equal(010-,010-)=1
```

```
Is_Equal(--10,--10)=1
```

```
Is_Equal(010-,111-)=0
```

```
Is_Equal(010-,01-0)=0
```

(a) Attach your source code with your solution on a separate page.

- Assume that the value and mask bits are stored in a data structure like:

```
unsigned int value; // up to 32-bits: 0=false, 1=true
```

```
unsigned int mask; // up to 32-bits: 0=Care, 1=DontCare
```

(b) Use your code to determine determine the return value for the following inputs:

```
SOP1.value=1001, SOP1.mask=0100
```

```
SOP2.value=1101, SOP2.mask=0100
```

(c) Use your code to determine the return value for the following inputs:

```
SOP1.value=1001, SOP1.mask=0100
```

```
SOP2.value=1101, SOP2.mask=0100
```

5. Consider a function called `One_Difference` that compares two SOP terms.

- For terms that have an identical dontcare masks AND have values that we care about but differ by only one term: `One_Difference` should return a bitmask that specifies the bit that differs
- For terms with different dontcare masks or for terms that differ in unmasked value bits by more than one value: `One_Difference` should return zero.

- For Example, the following function calls return:

```
One_Difference(010-,110-)=1000
```

```
One_Difference(-010,-000)=0010
```

```
One_Difference(010-,111-)=0000
```

```
One_Difference(010-,01-0)=0000
```

(a) Attach source code with your solution on a separate page.

(b) Use your function to determine the return value for the following input:

```
SOP1.value=1001, SOP1.mask=0100
```

```
SOP2.value=1010, SOP2.mask=0100
```

(c) Use your function to determine the return value for the following input:

```
SOP1.value=1001, SOP1.mask=1000
```

```
SOP2.value=1101, SOP2.mask=0100
```