Lecture 10: Quine-McCluskey 2-Level Minimization Algorithm

CSE 370, Autumn 2007
Benjamin Ylvisaker

Where We Are

- Last lecture: ROMs, PLAs and PALs, oh my
- This lecture: Quine-McCluskey Minimization
- Next lecture: Multi-Level Logic
- Homework 3 due today; homework 4 out
- In the midst of lab 3

5+ Variable K-Maps are a Pain

- Tedious to draw big K-maps
- Harder to find prime implicants in larger K-maps
- Relatively hard to automate directly

Quine-McCluskey to the Rescue

- Finds the exact same minimum 2-level implementations as K-map minimization
- Table-based method
- Easier to automate

Example Function

- F(A,B,C,D) = Σm(4,5,6,8,9,10,13) + Σd(0,7,15)

Step 1

- Translate all minterms and don't care terms to binary
- F(A,B,C,D) = Σm(4,5,6,8,9,10,13) + Σd(0,7,15)
- 0100, 0101, 0110, 1000, 1001, 1010, 1101, 0000, 0111, 1111

Step 2

- Sort according to number of 1's in the binary representation
- 0100, 0101, 0110, 1000, 1001, 1010, 1101, 0000, 0111, 1111
- 0000, 0100, 1000, 0101, 0110, 1001, 1010, 1101, 0111, 1111

Step 3

- Put terms in a table, grouped by number of 1's
- 0000, 0100, 1000, 0101, 0110, 1001, 1010, 1101, 0111, 1111
- Column 1

```
0000

0100
1000

0101
0110
1001
1010

0111
1101

1111
```


Step 4a

- Compare all pairs of terms from adjacent groups
  - If they match perfectly except for one 0-1 pair, place a term in the next column with a "-" in that bit position
- Column 1 Column 2

```
0000      0-00
          -000

0100
1000      010-
          01-0
0101      100-
0110      10-0
1001
1010      01-1
          -101
0111      011-
1101      1-01

1111      -111
          11-1
```

Step 4b

- If a term in column 1 can be combined with any other term in column 1 to make a term in column 2, place a check next to it

- Column 1 / Column 2

```
0000√    0-00
         -000
0100√
1000√    010-
         01-0
0101√    100-
0110√    10-0
1001√
1010√    01-1
         -101
0111√    011-
1101√    1-01

1111√    -111
         11-1
```


Step 4a (again)

- Column 1 / Column 2 / Column 3

```
0000√    0-00     01--
         -000
0100√             -1-1
1000√    010-
         01-0
0101√    100-
0110√    10-0
1001√
1010√    01-1
         -101
0111√    011-
1101√    1-01

1111√    -111
         11-1
```


Step 4b (again)

- Column 1 / Column 2 / Column 3

```
0000√    0-00     01--
         -000
0100√             -1-1
1000√    010-√
         01-0√
0101√    100-
0110√    10-0
1001√
1010√    01-1√
         -101√
0111√    011-√
1101√    1-01

1111√    -111√
         11-1√
```

Step 4c

- Place a * next to and term in column 2 that cannot be combined with another term in column 2 to make a term in column 3

- Column 1 Column 2 Column 3

```
0000√   0-00*    01--
        -000*
0100√            -1-1
1000√   010-√
        01-0√
0101√   100-*
0110√   10-0*
1001√
1010√   01-1√
        -101√
0111√   011-√
1101√   1-01*

1111√   -111√
        11-1√
```


Step 4a (again)

- Column 1 Column 2 Column 3 Column 4

```
0000√   0-00*    01--
        -000*
0100√            -1-1
1000√   010-√
        01-0√
0101√   100-*
0110√   10-0*
1001√
1010√   01-1√
        -101√
0111√   011-√
1101√   1-01*

1111√   -111√
        11-1√
```


Step 4b (again)

- Column 1 Column 2 Column 3 Column 4

```
0000√   0-00*    01--
        -000*
0100√            -1-1
1000√   010-√
        01-0√
0101√   100-*
0110√   10-0*
1001√
1010√   01-1√
        -101√
0111√   011-√
1101√   1-01*

1111√   -111√
        11-1√
```

Step 4c (again)

- Column 1 Column 2 Column 3 Column 4

```
0000√    0-00*    01--*
         -000*
0100√             -1-1*
1000√    010-√
         01-0√
0101√    100-*
0110√    10-0*
1001√
1010√    01-1√
         -101√
0111√    011-√
1101√    1-01*

1111√    -111√
         11-1√
```

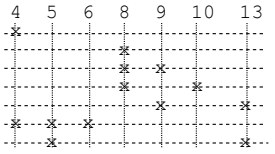

Step 5

- List all of the *-ed terms; these are the prime implicants
  - 0-00, -000, 100-, 10-0, 1-01, 01--, -1-1
- Column 1 Column 2 Column 3 Column 4

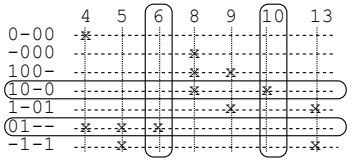
```
0000√    0-00*    01--*
         -000*
0100√             -1-1*
1000√    010-√
         01-0√
0101√    100-*
0110√    10-0*
1001√
1010√    01-1√
         -101√
0111√    011-√
1101√    1-01*

1111√    -111√
         11-1√
```


Step 6

- Build a prime implicant table, with the prime implicants along the left, the minterms (but not the don't cares) along the top, and an x in every cell where the prime implicant covers the minterm
-

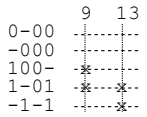


Step 7a

- Find columns (minterms) that only have and x in one row (implicant); "include" these implicants in the function implementation
- 

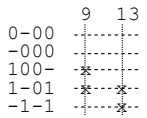
- {10-0, 01--}

Step 7b

- Remove the included implicants from the table, and any minterms that they cover
- 

- {10-0, 01--}

Step 8

- Heuristically include more implicants until all minterms are covered
- 

- {10-0, 01--, 1-01}

Step 9

- Translate included implicants into a minimized sum-of-products form
- {10-0, 01--, 1-01}
- F(A,B,C,D) = A¬B¬D + ¬AB + A¬CD

Now You Try

- F(A,B,C,D) = Σm(0,3,6,8,9,11,15) + Σd(1,2,4,12)
- Column 1

```
0000

0001
0010
0100
1000

0011
0110
1001
1100

1011

1111
```


Thank You for Your Attention

- Finish lab 3; start reading lab 4
- Start looking at homework 4
- Continue reading the book