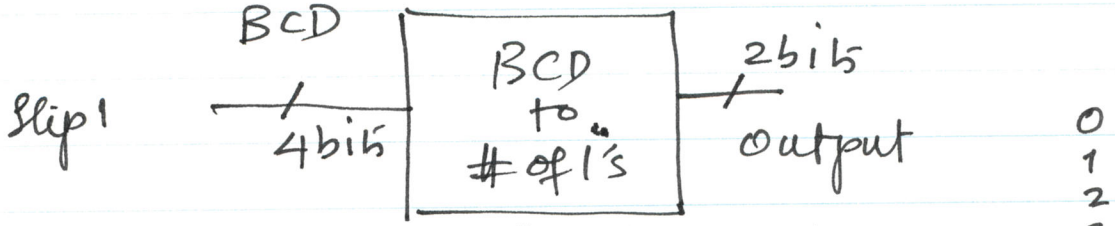
Sept 15, 2017

Problem solving using logic gates.

BCD → odd parity

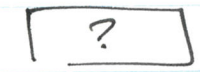
BCD → count the number 1's in the input.



Step 1

Step 2. TT BCD - (0-9) inputs

BCD	A	B	C	D	f_1	f_0
0	0	0	0	0	0	0
1	0	0	0	1	0	1
2	0	0	1	0	0	1
3	0	0	1	1	1	0
4	0	1	0	0	0	1
5	0	1	0	1	1	0
6	0	1	1	0	1	0
7	0	1	1	1	1	1
8	1	0	0	0	0	1
9	1	0	0	1	1	0



$f_0(A,B,C,D) = ? = \Sigma(1,2,4,7,8$

$f_1(A,B,C,D) = ? \Sigma(3,5,6,7,9$

⇓

Simplify the expression

⇓

Implement using gates

algebraic simplification

Sept 15, 2017

$f_0(A,B,C,D) = \Sigma(m_1, m_2, m_4, m_7, m_8)$

$f_1(A,B,C,D) = \Sigma(m_3, m_5, m_6, m_7, m_9)$

→ $A'B'C'D + A'B'CD' + \cancel{AB'C'D'}$ $A'BC'D' + A'BCD + AB'C'D'$

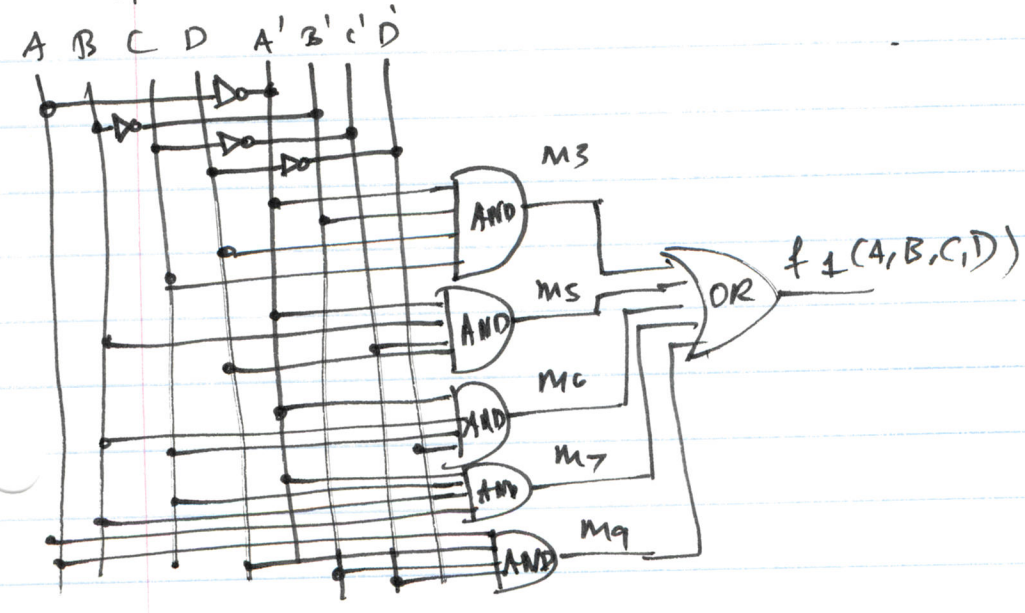
Sum of products:

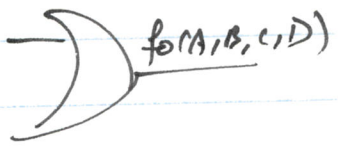
$f_0 = A'B'(C'D + CD') + A'B(C'D' + CD) + AB'C'D'$

$f_1(A,B,C,D)$: $\underbrace{A'B'CD}_{m_3} + \underbrace{\cancel{A'BC'D}}_{m_5}$ ($A'BC'D$) $+ \underbrace{A'BCD'}_{m_6} + \underbrace{A'BCD}_{m_7} + \underbrace{AB'C'D}_{m_9}$

$= B'CD(A'+A) + \cdots$

Bus





2017

Improvements. Sept 15 (3)

1. We can use don't care conditions in simplification

BCD 0 – 9 ↕ 6 combination
4 bits 0 – 15 ↕

2. Algebraic simplication
→ K-map
Karnaugh map

3. use only NAND gates for implementation

. Problem : $f_0(A,B,C,D) = \Sigma(1,2,4,7,8)$

$f_0(A,B,C,D) = \Sigma(3,5,6,7,9)$

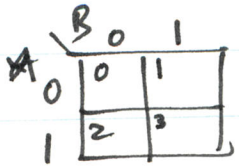
+ $d(A,B,C,D) = \Sigma(10,11,12,13,14,15)$

d = don't cares

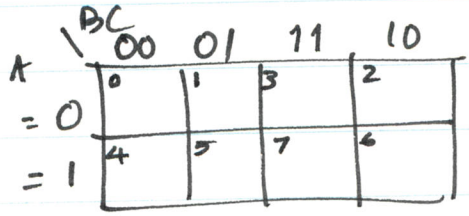
never going to happen in the input.

use these if you want to, for simplication.

2-variable K-map



3-variable K-map



K-map

Sept 15 (24)

Empty 4-variable map

AB \ CD	=00	=01	=11	=10
=00	0	1	3	2
=01	4	5	7	6
=11	12	13	15	14
=10	8	9	11	10

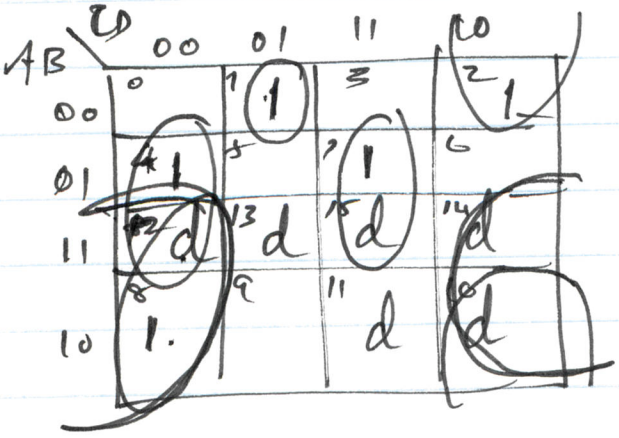
goal:

- to group the 1's into largest groups of rectangular adjacent 1's.
- cover all 1's with min. # of groups

use K-maps for simplification

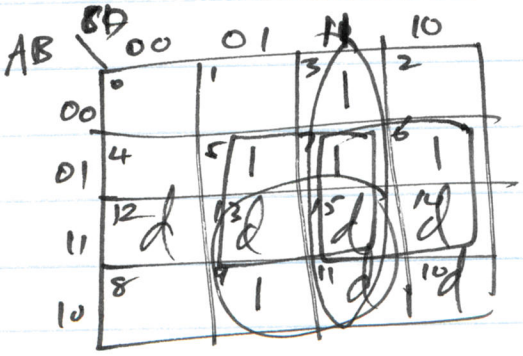
plot the fn:

f0:



group 1: BCD
group 2: B'CD'
group 3: AD'
wraparound:
group 4: BC'D'
group 5: A'B'C'D

f1:



f1 group 1: BC
f1 group 2: BD
f3 group 3: AD
f11 group 4: CD