EEC18

Oct. 19

__Implicants__ - any valid K-map group

(P.I.)
~~Prime~~ __Prime Implicant__ - max-size implicant

(E.P.I.)
__Essential Prime Implicant__ - P.I. that is essential for a min solution

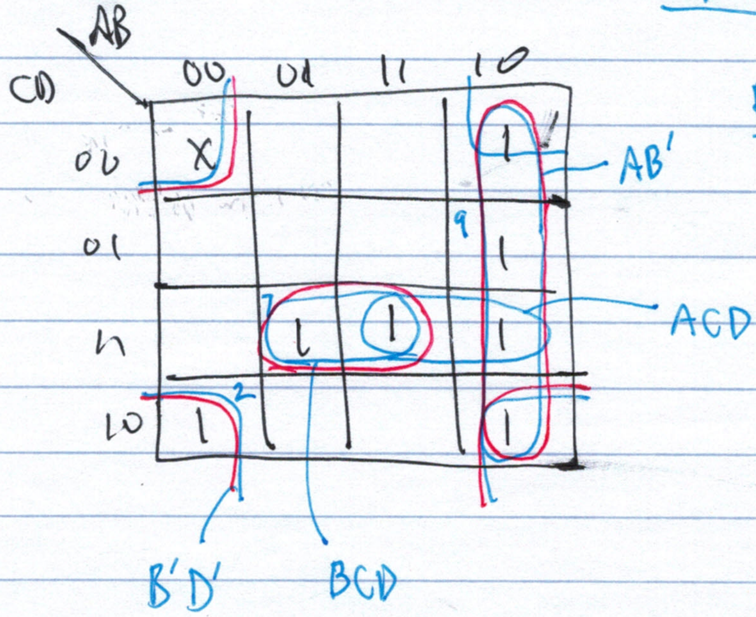
__Minimum Solution:__

always contains all EPI's
may " some P.I.'s (not EPI)
never " any plain Implicants (not P.I. or EPIs)

To find min soln:

1) Include all EPIs
2) If necessary, add P.Is until all minterms are covered. Cover X's only if helpful.

__Implicants__: all groups



__PI's:__ AB', ACD, BCD, B'D'

__EPIs:__ AB', BCD, B'D'

$m_9 \rightarrow AB'$

$m_7 \rightarrow BCD$

$m_2 \rightarrow B'D'$

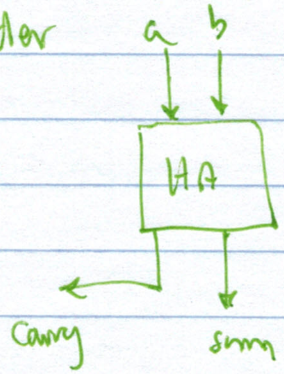
To find EPIs:

1) Choose a minterm (1 on K-map) and look at all P.I.s that cover it
2) If only 1 P.I. → that one is essential
   If > 1 P.I. → learn nothing
3) Repeat for all minterms

Adders (Unit 4)

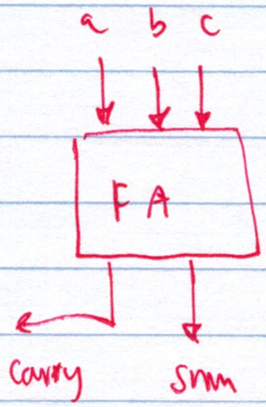
```
        1     - carry
  0 1 0 1     - in
+ 0 1 0 1
---------
        0     - sum
```

Half Adder



a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

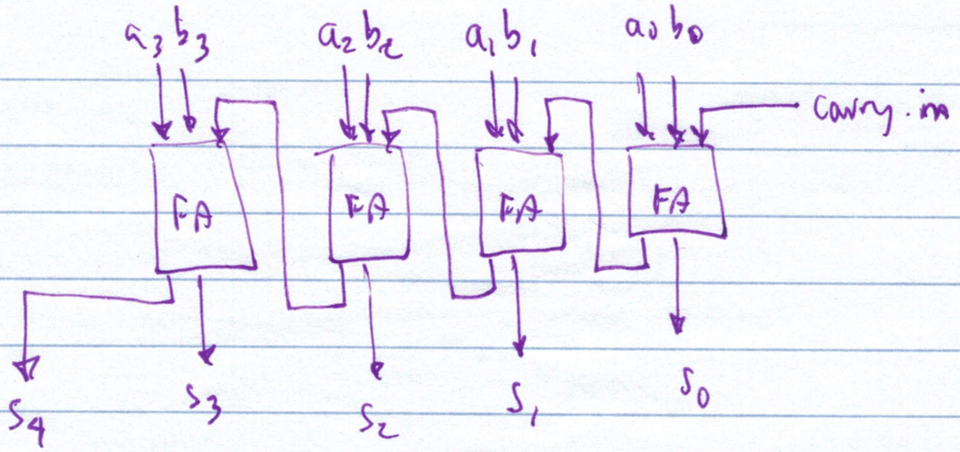
Full Adder



a	b	c	car	sum
0	0	0	0	0
0	0	1	0	1
1	1	1	1	1

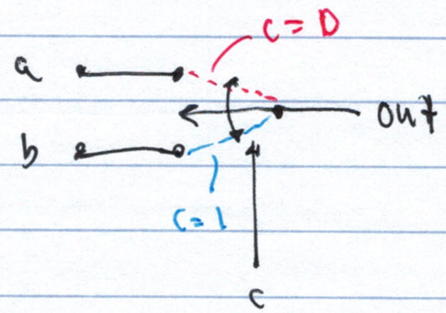
$sum = a \oplus b \oplus c$

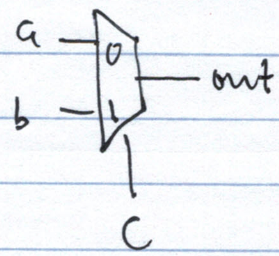
$car = ab + bc + ac$



Ripple - Carry Adder
- simplest
- slowest

Mux

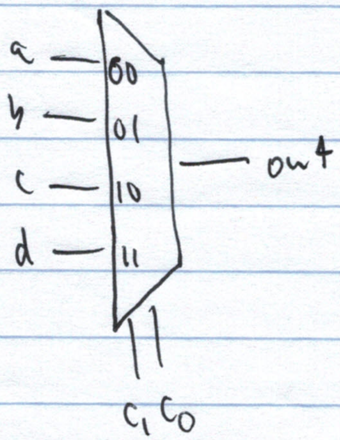




2:1 mux

c	out
0	a
1	b

a	b	c	out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



4:1 mux

2^N inputs N select bits

Quine-McCluskey (Unit 6)

- Works with many inputs
- Easy to program

1) Find all P.Is
2) ~~E~~ Find all E.P.Is
3) Build min solution

$$\overset{001}{} \; \overset{010}{} \; \overset{101}{} \; \overset{110}{} \; \overset{111}{}$$

$$Z = \sum m(1, 2, 5, 6, 7)$$

$$= A'B'C + A'BC' + AB'C + ABC' + ABC$$

AB \ C	0	1
00		1
01	1	
11	1	1
10		1

	minterms	col 1	col 2
group 0 1's			
group 1 1's	1	001 ✓	−01
	2	010 ✓	−10
group 2 1's	5	101 ✓	1−1
	6	110 ✓	11−
group 3 1's	7	111 ✓	

All P.I's

Unchecked terms are P.I's

List groups by # of 1's in minterm

Compare minterm pairs in adjacent groups

Copy term pairs that differ in one variable to next column and check "✓" matched terms

Repeat until no more matches

	Prime Implicants		Minterms 1	2	5	6	7	
EPI	(1,5)	$B'C$	X		X			
EPI	(2,6)	BC'		X		X		
	(5,7)	AC			X		X	Choose
	(6,7)	AB				X	X	Choose

Min solution $= B'C + BC' + AC$

$= B'C + BC' + AB$

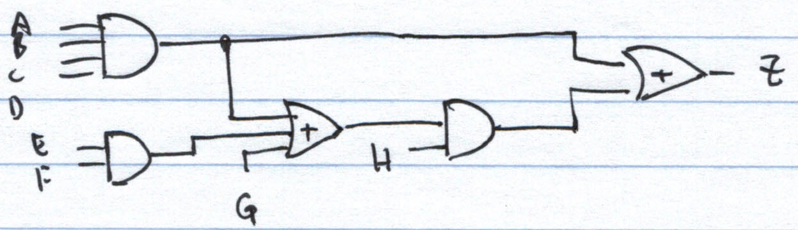
Multi-level Circuits

Level of a circuit = max. # of gates in series between an input and output

SOP : 2-level : AND - OR

POS : 2-level : OR - AND

Ex: $Z = (ABCD + EF + G) \cdot H + ABCD$



4-level

Why do levels matter?

- # of gates
- fanin (# of inputs to a gate)
- fanout (# of outputs of a gate)
- delay

NANDs and NORs

Minterms and maxterms → AND, OR, NOT are "functionally complete"

AND? can't make an inverter → Not functionally complete

$A \cdot A$

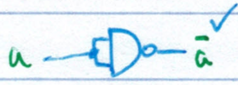
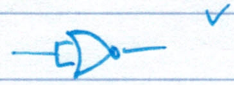
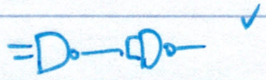
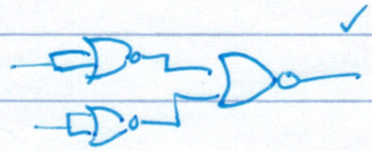
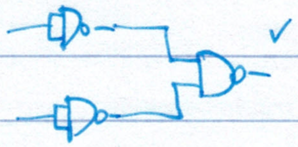
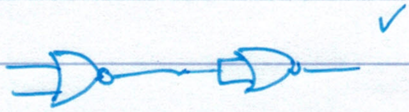
OR? can't make an inverter → " " "

$A + A$

INV? 

can't make AND, OR → Not func. complete

a & a = a

NAND?	NAND	NOR
NOT		
AND		
OR		

NAND is func. complete!

NOR " " "