DDCO

UNIT-1

CLASS NOTES

feedback/corrections: vibha@pesu.pes.edu

Vibha Masti

introduction

Course instructor: Dr. Reetinder Sidhu, reetindersidhu@pes.edu

- knowledge of hardware essential for efficient software

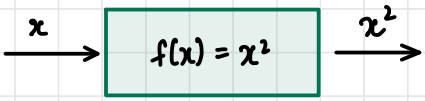
Moore's Law

- ~30 years ago, supercomputers needed liquid nitrogen
- Every ~18 months, no. of transistors per chip area doubles
- Transistor speed 2x, power consumption ½x.
- Moore's Law is now slowing down
- Therefore, to improve performance, deeper knowledge of hardware needed for good software
- Targeted hardware accelerators developed
- Google has TensorFlow & Tensor Processing Unit accelerators

BOOLEAN FUNCTIONS

Mathematical Functions

- Example: parabola
- Domain & range set of real numbers
- Can be specified on Cartesian plane
- Specify function as a box

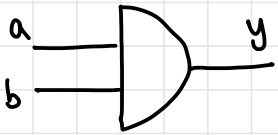


Boolean Functions

- Example: AND function / gate
- Domain & range: {0,1}
- Specify function using truth table

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

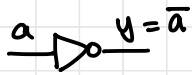
- Specify as logic gate (black box)

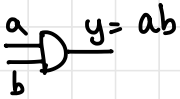


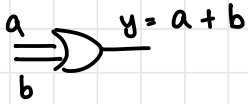
TODO: redraw

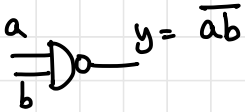
BASIC FUNCTIONS / LOGIC GATES

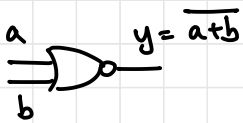
1) BUFFER 

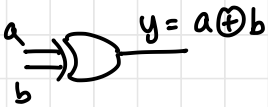
2) NOT 

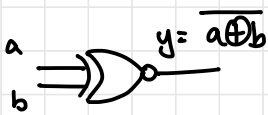
3) AND 

4) OR 

5) NAND 

6) NOR 

7) XOR 

8) XNOR 

- Number of rows for a n-input TT = 2^n
- How many different 1-input boolean functions can exist? (2^{2^1})

Layers of Abstraction

- Logic minimisation: truth table
- Logic design level: component in logic circuit
- CMOS VLSI design: transistor level circuit diagram with digital switches (on or off)
- VLSI layout level (silicon, metal)
- VLSI fabrication level – chip (microprocessor)

Boolean Algebra & Identities

Logic circuits

- Multiple logic gates
- O/P of one gate connected to I/P of another
- Represent Boolean functions
- Look at Perfect Induction

Boolean Formulas

- Boolean constants (0 and 1) are Boolean formulas
- Boolean variables (like x) are Boolean formulas
- If P & Q and Boolean formulas,

Precedence
1. $\overline{P}$
2. $P \cdot Q$
3. $P + Q$

] Boolean formulas

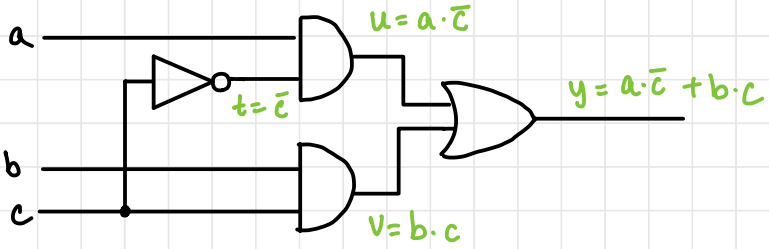
Q: How is $((a \cdot \overline{c}) + (b \cdot c))$ a Boolean formula?

1. a, b & c are Boolean formulas
2. $\overline{c}$ is a Boolean formula
3. $a \cdot \overline{c}$ and $b \cdot c$ are Boolean formulas
4. $a \cdot \overline{c} + b \cdot c$ is a Boolean formula

Boolean Formulas & Logic Circuits

1. Constants & variables are inputs to logic gates
2. $\cdot$ operator means AND
3. $+$ operator means OR
4. $\overline{\ }$ operator means NOT

Q: Convert $((a \cdot \bar{c}) + (b \cdot c))$ to a logic circuit



COMBINATIONAL LOGIC CIRCUITS

- Circuits that can be represented by Boolean formulas
- Unlike sequential logic circuits (with feedback)

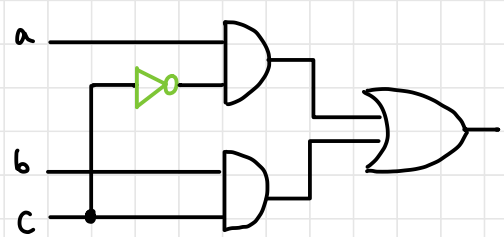
* HDL (hardware descriptive language) not in this course.

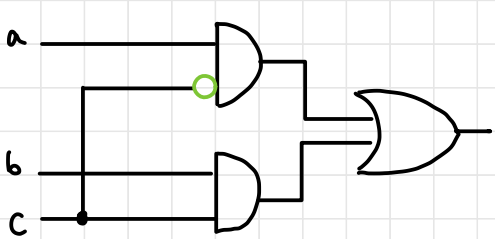
BOOLEAN ALGEBRA & IDENTITIES

- Boolean functions can be represented as truth tables, combinational logic circuits and Boolean formulas
- Multiple Boolean functions & CLCs for Boolean formula

Note-Bubbles

- NOT gates can be replaced with a bubble





Boolean Algebra

1. Set: {0, 1}
2. Operations: AND, OR, NOT
3. Identity element: (0 – OR, 1 – AND)
4. Laws: commutative, associative, distributive

Boolean Identities

1. Commutative

$$a \cdot b = b \cdot a$$
$$a + b = b + a$$

↕ dual

2. Associative

$$(a \cdot b) \cdot c = a \cdot (b \cdot c)$$
$$(a + b) + c = a + (b + c)$$

↕ dual

3. Distributive

$$a \cdot (b + c) = a \cdot b + a \cdot c$$
$$a + (b \cdot c) = (a + b) \cdot (a + c)$$

↕ dual

4. De Morgan's

$$\overline{(a + b)} = \bar{a} \cdot \bar{b}$$
$$\overline{(a \cdot b)} = \bar{a} + \bar{b}$$

↕ dual

5. Idempotency

$$a \cdot a = a$$
$$a + a = a$$

6. Identity

$$a \cdot 1 = a$$
$$a + 0 = a$$

7. Boundedness

$$a \cdot 0 = 0$$
$$a + 1 = 1$$

8. Complement

$$a \cdot \bar{a} = 0$$
$$a + \bar{a} = 1$$

9. Absorption

$$a + a \cdot b = a$$
$$a \cdot (a + b) = a$$

10. Involution

$$\bar{\bar{a}} = a$$

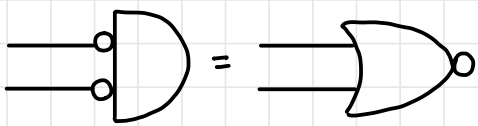
11. Useful Identity

$$a + \bar{a} \cdot b = a + b = (a + \bar{a}) \cdot (a + b) = a + b$$
$$a \cdot (\bar{a} + b) = a \cdot b = a \cdot \bar{a} + a \cdot b = a \cdot b$$

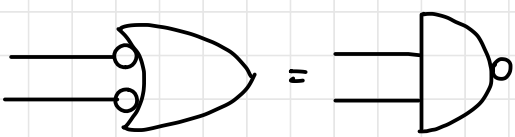
MULTIPLE INPUTS

- Due to associativity, $a+b+c$ & $a \cdot b \cdot c$ require no brackets and the order does not matter.
- Using DeMorgan's Laws

$$\bar{a} \cdot \bar{b} = \overline{a+b}$$



$$\bar{a} + \bar{b} = \overline{a \cdot b}$$



logic minimization & K-MAPS

- construct Boolean formula / CLC using truth table
- using smallest two-level Sum of Products formula

TERMINOLOGY

- **Literal** — Bool variable / complement (eg: $\bar{c}$)
- **Product** — AND of two or more literals (eg: $ab\bar{c}$)
- **Minterm** — product involving all inputs to Boolean function (such that minterm product = 1)

a	b	c	minterm	name
0	0	0	$\overline{a}\,\overline{b}\,\overline{c}$	m_0
0	0	1	$\overline{a}\,\overline{b}c$	m_1
0	1	0	$\overline{a}b\,\overline{c}$	m_2
0	1	1	$\overline{a}bc$	m_3
1	0	0	$a\,\overline{b}\,\overline{c}$	m_4
1	0	1	$a\,\overline{b}c$	m_5
1	1	0	$ab\,\overline{c}$	m_6
1	1	1	abc	m_7

lowercase m

names of the minterms

- sum — OR of two or more literals (eg: $a+\overline{b}+c$)
- maxterm — sum involving all inputs to Boolean function (such that maxterm sum = 0)

a	b	c	maxterm	name
0	0	0	$a+b+c$	M_0
0	0	1	$a+b+\overline{c}$	M_1
0	1	0	$a+\overline{b}+c$	M_2
0	1	1	$a+\overline{b}+\overline{c}$	M_3
1	0	0	$\overline{a}+b+c$	M_4
1	0	1	$\overline{a}+b+\overline{c}$	M_5
1	1	0	$\overline{a}+\overline{b}+c$	M_6
1	1	1	$\overline{a}+\overline{b}+\overline{c}$	M_7

uppercase M

names of maxterms

Sum of Products

- sum of all minterms corresponding to a 1 output
- eg:

a	b	c	y	minterm	name
0	0	0	0	$\bar{a}\bar{b}\bar{c}$	m_0
0	0	1	0	$\bar{a}\bar{b}c$	m_1
0	1	0	0	$\bar{a}b\bar{c}$	m_2
0	1	1	1	$\bar{a}bc$	m_3
1	0	0	1	$a\bar{b}\bar{c}$	m_4
1	0	1	0	$a\bar{b}c$	m_5
1	1	0	1	$ab\bar{c}$	m_6
1	1	1	1	abc	m_7

- SOP form Boolean formula:

$$y = \bar{a}bc + a\bar{b}\bar{c} + ab\bar{c} + abc$$

$$y = m_3 + m_4 + m_6 + m_7$$

- sigma (Σ) notation

$$y = \Sigma(m_3, m_4, m_6, m_7)$$

$$y = \Sigma(3,4,6,7)$$

- Boolean function f

$$y = f(a,b,c) = \Sigma(m_3, m_4, m_6, m_7)$$

- can use 4 3-input AND gates and 1 4-input OR gate to construct CLC

Product of sums

- product of all maxterms corresponding to a 0 output
- eg:

a	b	c	y	maxterm	name
0	0	0	0	$a+b+c$	M_0
0	0	1	0	$a+b+\bar{c}$	M_1
0	1	0	0	$a+\bar{b}+c$	M_2
0	1	1	1	$a+\bar{b}+\bar{c}$	M_3
1	0	0	1	$\bar{a}+b+c$	M_4
1	0	1	0	$\bar{a}+b+\bar{c}$	M_5
1	1	0	1	$\bar{a}+\bar{b}+c$	M_6
1	1	1	1	$\bar{a}+\bar{b}+\bar{c}$	M_7

- POS from Boolean formula

$$y = (a+b+c)\cdot(a+b+\bar{c})\cdot(a+\bar{b}+c)\cdot(\bar{a}+b+\bar{c})$$

$$y = M_0 M_1 M_2 M_5$$

- Pi (Π) notation

$$y = \Pi(M_0, M_1, M_2, M_5)$$

$$y = \Pi(0,1,2,5)$$

- Boolean function f

$$y = f(a,b,c) = \Pi(0,1,2,5)$$

Logic Minimisation using Identities

- Consider $y = \bar{a}bc + a\bar{b}\bar{c} + ab\bar{c} + abc$
- Minimise using identities

$$y = \bar{a}bc + a\bar{b}\bar{c} + ab\bar{c} + abc$$
$$= bc(a+\bar{a}) + a\bar{c}(b+\bar{b})$$
$$= bc + a\bar{c}$$

Q: Prove that $ab + bc + a\bar{c} = bc + a\bar{c}$

$$ab + bc + a\bar{c}$$
$$= ab(c+\bar{c}) + bc + a\bar{c}$$
$$= abc + ab\bar{c} + bc + a\bar{c}$$
$$= bc(a+1) + a\bar{c}(b+1)$$
$$= bc + a\bar{c}$$

$= ab + \bar{a}bc + a\bar{b}\bar{c}$

$= a(b + \bar{b}\bar{c}) + \bar{a}bc$

$= a(b+\bar{c}) + \bar{a}bc$

$= ab + a\bar{c} + \bar{a}bc$

$= b(a+\bar{a}c) + a\bar{c}$

$= b(a+c) + a\bar{c}$

$= bc + a\bar{c} + ab$

Q: Minimise $(a+b+c)(a+b+\bar{c})(a+\bar{b}+c)(\bar{a}+b+\bar{c})$

$$[(a+b) + (a+b)\bar{c} + (a+b)c](a+\bar{b}+c)(\bar{a}+b+\bar{c})$$
$$= (a+b)(a\bar{a} + ab + a\bar{c} + \bar{b}\bar{a} + \bar{b}b + \bar{b}\bar{c} + c\bar{a} + cb + 0)$$
$$= (a+b)(ab + a\bar{c} + \bar{a}\bar{b} + \bar{b}\bar{c} + c(\bar{a} + bc))$$
$$= ab + a\bar{c} + a\bar{b}\bar{c} + abc + ab + ab\bar{c} + \bar{a}bc + bc$$
$$= ab(1+c) + a\bar{c}(1+\bar{b}) + ab\bar{c} + bc(1+\bar{a})$$
$$= ab + a\bar{c} + bc + ab\bar{c}$$
$$= ab + a\bar{c} + bc$$
$$= abc + ab\bar{c} + a\bar{c} + bc$$
$$= bc + a\bar{c}$$

K-Maps

- Named after Maurice Karnaugh
- Minterms that differ in one literal must be adjacent

IMPLICANTS

- K-Map area composed of squares containing 1s
- Area is square/rectangular (wraparound allowed)
- No. of squares in area is power of 2
- Each implicant → product of literals

Prime Implicants

- Implicant with largest no. of squares obeying the rules

Essential prime Implicant

- Prime implicant containing square not in any other prime implicant

- K map method: includes all prime implicants such that each 1 square is covered and formula is minimal.

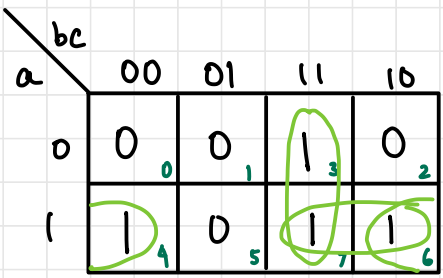
- For this TT

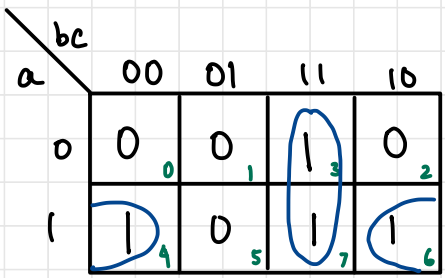
a	b	c	y	minterm	
0	0	0	0	$\bar{a}\bar{b}\bar{c}$	0
0	0	1	0	$\bar{a}\bar{b}c$	1
0	1	0	0	$\bar{a}b\bar{c}$	2
0	1	1	1	$\bar{a}bc$	3
1	0	0	1	$a\bar{b}\bar{c}$	4
1	0	1	0	$a\bar{b}c$	5
1	1	0	1	$ab\bar{c}$	6
1	1	1	1	abc	7

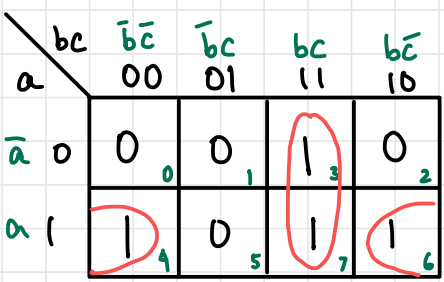
a \ bc	00	01	11	10
0	0 $\bar{a}\bar{b}\bar{c}$ 0	1 $\bar{a}\bar{b}c$ 0	3 $\bar{a}bc$ 1	2 $\bar{a}b\bar{c}$ 0
1	4 $a\bar{b}\bar{c}$ 1	5 $a\bar{b}c$ 0	7 abc 1	6 $ab\bar{c}$ 1

Example 1

1. Prime implicants — green
2. Essential — blue
3. Required — red



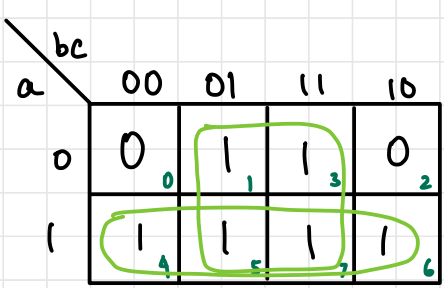


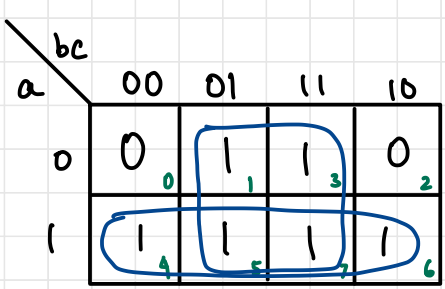


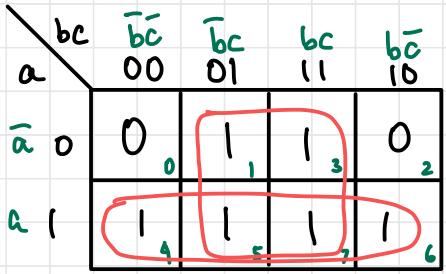
(3 & 4 only covered by these 2)

SOP: $a\bar{c} + bc$ (unchanging terms)

Example 2



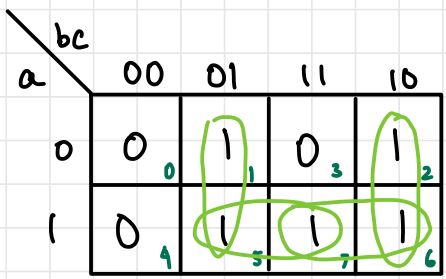


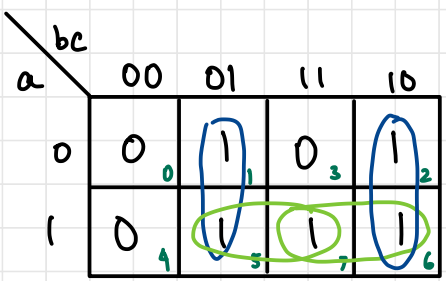


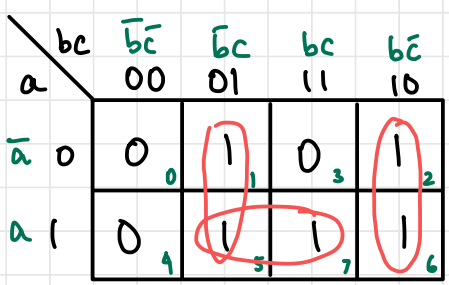
(1,3 & 4,6)

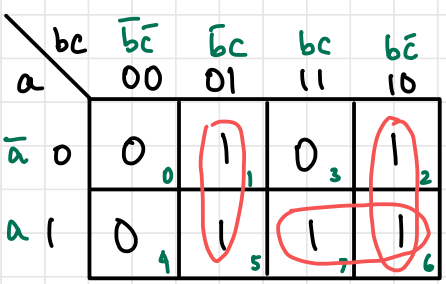
SOP: $c + a = f(a,b,c)$

Example 3







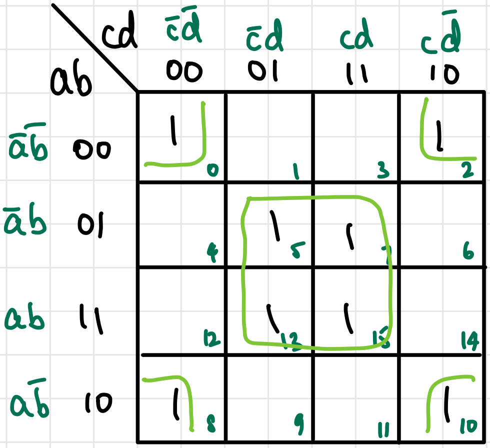


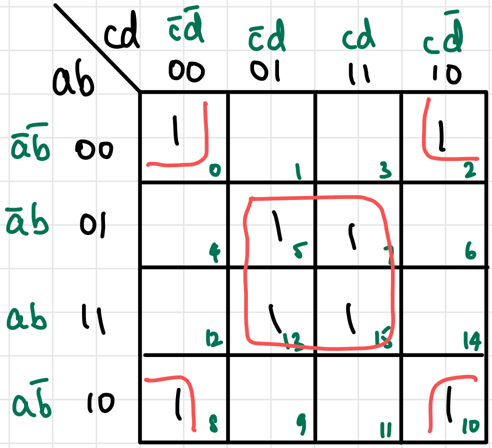
SOP = $\bar{b}c + b\bar{c} + ca$

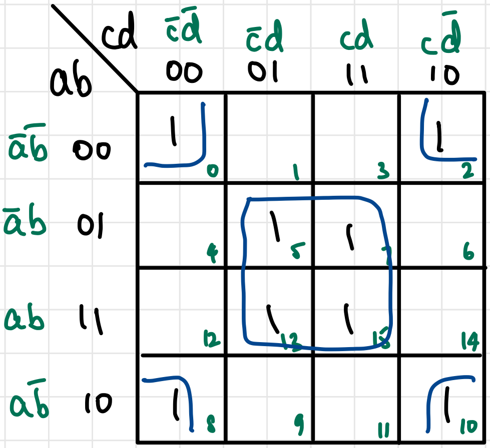
SOP = $\bar{b}c + b\bar{c} + ba$

(no unique solution)

Example 4



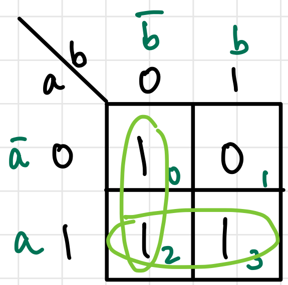




SOP = $db + \bar{d}\bar{b}$

Example 5

a	b	y
0	0	1
0	1	0
1	0	1
1	1	1



SOP = $\bar{b} + a$

DON'T CARES

- If we know for a fact that certain input combinations cannot occur in a logic gate, we place a don't care
- The logic circuit will never receive that input combination, so we don't care if it's a 0 or a 1

Example 6

green - required (?=X)

a \ bc	00	01	11	10
0	1 (0)	1 (1)	0 (3)	0 (2)
1	1 (4)	1 (5)	X (7)	1 (6)

Do for both 0 and 1

a \ bc	$\bar{b}\bar{c}$ 00	$\bar{b}c$ 01	bc 11	$b\bar{c}$ 10
$\bar{a}$ 0	1 (0)	1 (1)	0 (3)	0 (2)
a 1	1 (4)	1 (5)	0 (7)	1 (6)

$SOP = y = \bar{b} + \bar{c}a$

a \ bc	$\bar{b}\bar{c}$ 00	$\bar{b}c$ 01	bc 11	$b\bar{c}$ 10
$\bar{a}$ 0	1 (0)	1 (1)	0 (3)	0 (2)
a 1	1 (4)	1 (5)	1 (7)	1 (6)

$SOP = y = \bar{b} + a$

- Whichever results in a smaller Boolean formula is used

Example 7

$$f(a,b,c,d) = \Sigma(m_3, m_4, m_5, m_7, m_9, m_{13}, m_{14}, m_{15})$$

ab \ cd	$\bar{c}\bar{d}$ 00	$\bar{c}d$ 01	cd 11	$c\bar{d}$ 10
$\bar{a}\bar{b}$ 00	0 (0)	0 (1)	1 (3)	0 (2)
$\bar{a}b$ 01	1 (4)	1 (5)	1 (7)	0 (6)
ab 11	0 (12)	1 (13)	1 (15)	1 (14)
$a\bar{b}$ 10	0 (8)	1 (9)	0 (11)	0 (10)

$$y = \bar{c}\bar{a}b + cd\bar{a} + abc + \bar{c}da$$

Example 8

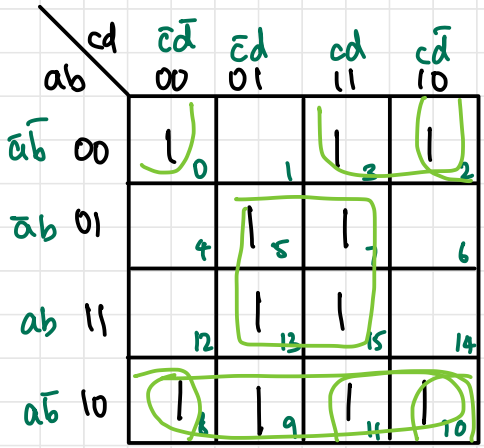
$$f(a,b,c,d) = \Sigma(0,1,2,6,8,9,10)$$

ab \ cd	$\bar{c}\bar{d}$ 00	$\bar{c}d$ 01	cd 11	$c\bar{d}$ 10
$\bar{a}\bar{b}$ 00	1 (0)	1 (1)	0 (3)	1 (2)
$\bar{a}b$ 01	0 (4)	0 (5)	0 (7)	1 (6)
ab 11	0 (12)	0 (13)	0 (15)	0 (14)
$a\bar{b}$ 10	1 (8)	1 (9)	0 (11)	1 (10)

$$y = \bar{c}\bar{b} + \bar{a}c\bar{d} + \bar{b}cd$$

Example 9

$f(a,b,c,d) = \Sigma(0,2,3,5,7,8,9,10,11,13,15)$



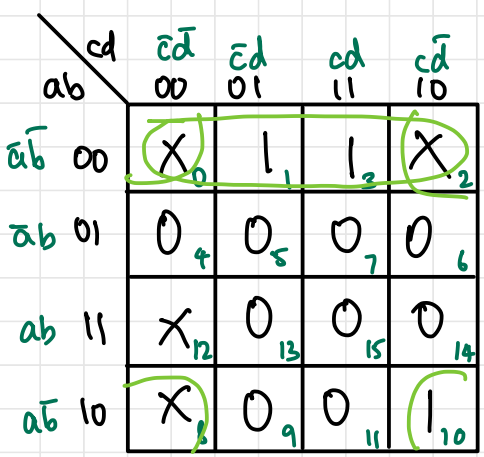
$y = bd + \bar{b}c + a\bar{b} + \bar{d}\bar{b}$

not the only solution

Example 10

don't care

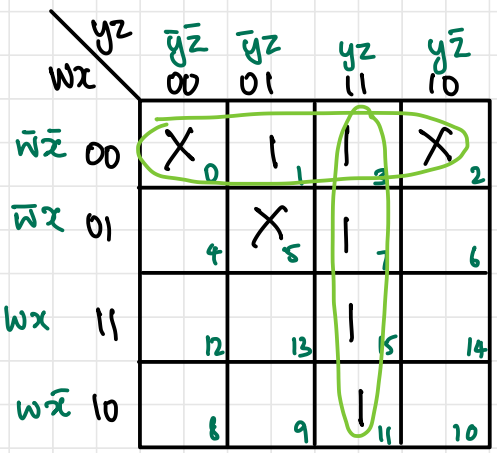
$f(a,b,c,d) = \Sigma(1,3,10) + \Sigma_d(0,2,8,12)$



$y = \bar{d}\bar{b} + \bar{a}\bar{b}$

Example 11

$$f(w,x,y,z) = \Sigma(1,3,7,11,15) + \Sigma_d(0,2,5)$$



SOP = $\bar{w}\bar{z} + yz$

BINARY NUMBERS

1) Unsigned
2) Signed
] representation

UNSIGNED

5 - 101
4 - 100

$$m = \sum_{i=0}^{n-1} x_i \times 2^i$$

n = no. of digits
x_i = digit

SIGNED

MSB indicates sign

MSB 1 ⟶ -ve
MSB 0 ⟶ +ve

Represent Positive No.s

+5 ⟶ 0101
+6 ⟶ 0110

Represent Negative No.s

(three ways)

1. Simple Sign Magnitude

+5 ⟶ 0101
-5 ⟶ 1101

2. 1's complement

$+5 \longrightarrow 0101$

1's complement $\longrightarrow 1010$ ↘ invert all 0's and 1's

$-5 \longrightarrow 1010$

problem: +0 and -0 are different

$+0 \longrightarrow 0000$
$-0 \longrightarrow 1111$

3. 2's Complement

- we do not use sign magnitude or 1's complement forms
- 1's comp + 1

$+5 \longrightarrow 0101$
1's comp $\longrightarrow 1010$
$-5 \longrightarrow 1011 \longrightarrow$ 2's complement

- advantage: 0 is universal

$+0 \longrightarrow 0000$
1's comp $\longrightarrow 1111$
$-0 \longrightarrow 0000$ (overflow)

- 2's complement of 2's complement is the original no.
- used in comps

- Eg:

$+5 \longrightarrow 0101$

$\downarrow$ 2's complement

$-5 \longrightarrow 1011$

$\downarrow$ 2's complement

$+5 \longrightarrow 0101$

Intuition:

$-5 \longrightarrow 1011$ ② $+1$

$-4 \longrightarrow 1100$

$\therefore$ can add 2's comp to +ve no. for subtraction

- for an n-bit signed no, 2^{n-1} no.s for 0 & +ve, 2^{n-1} for -ve

0000	0	1000	-8
0001	1	1001	-7
0010	2	1010	-6
0011	3	1011	-5
0100	4	1100	-4
0101	5	1101	-3
0110	6	1110	-2
0111	7	1111	-1

~~adding and subtracting~~ binary numbers

ADDITION

- binary addition

$5+3$

$$\begin{array}{r} {\scriptstyle 1\,1\,1\,} \\ 0101 \\ +\ 0011 \\ \hline 1000 \end{array} \quad = 8$$

SUBTRACTION

- to do $a-b$ do $a + (-b)$ or $a +$ 2's complement of b
- note: signed vs unsigned addition
- $3-5 = -5+3$

$$\begin{array}{r} {\scriptstyle \,1\,0\,1\,1} \\ +\ 0011 \\ \hline 1110 \end{array} \longrightarrow \text{find 2's comp}$$

$0001 + 1 = 0010$

$\therefore$ res $= -2$

Example 12

7 11 or -5

- Add 0111 and 1011 deal with overflow
  - ↳ signed
  - ↳ unsigned

unsigned

```
  1 11
  0111
+ 1011
-------
 10010  = 2
```

overflow is

signed

```
  1 11
  0111
+ 1011
-------
 10010  = 2
```

no overflow?

There are 2 4-bit no.s whose 2's complement does not reverse the sign

1. 0000 ⟶ 0

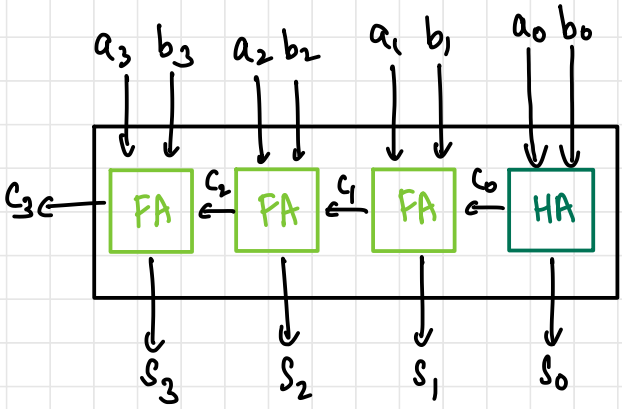
   2's comp: 1111 + 1 = 10000 (overflow)

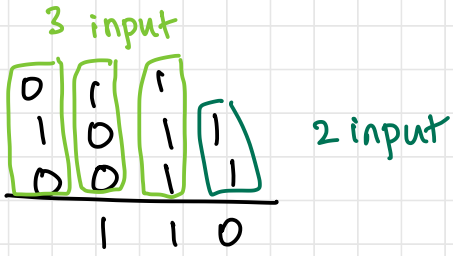
2. 1000 ⟶ -8

   2's comp: 0111 + 1 ⟶ 1000 ⟶ -8

LOGIC CIRCUITS FOR ADDITION

structure of logic circuits





- LSB: 2-input adder — half adder (HA)
- all other bits: 3-input adder — full adder (FA)

HALF ADDER

Truth Table

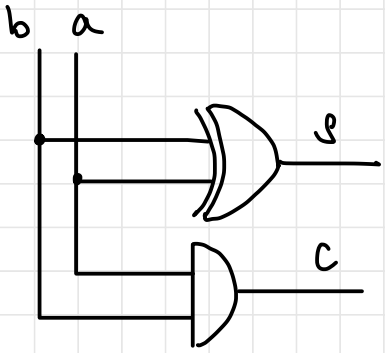
inputs		outputs	
a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- SOP formulas (minterms)

$s = \bar{a}b + a\bar{b}$ (XOR function)

$s = a \oplus b$

$c = ab$ (AND function)



FULL ADDER

a	b	c	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- SOP formulas

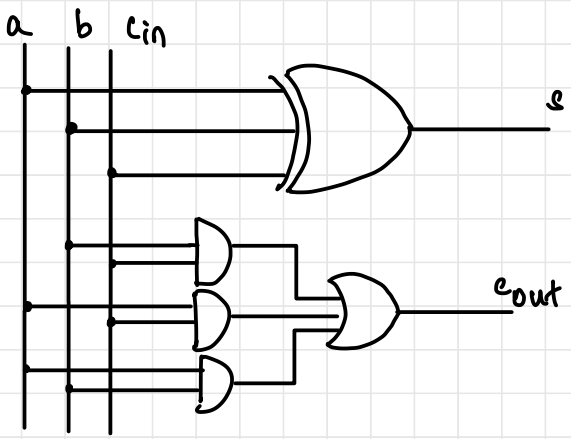
$$S = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$$

$$S = a \oplus b \oplus c$$

$$C_{out} = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc$$

$$= bc(a+\bar{a}) + ac(b+\bar{b}) + ab(c+\bar{c})$$

$$C_{out} = ab + bc + ac$$



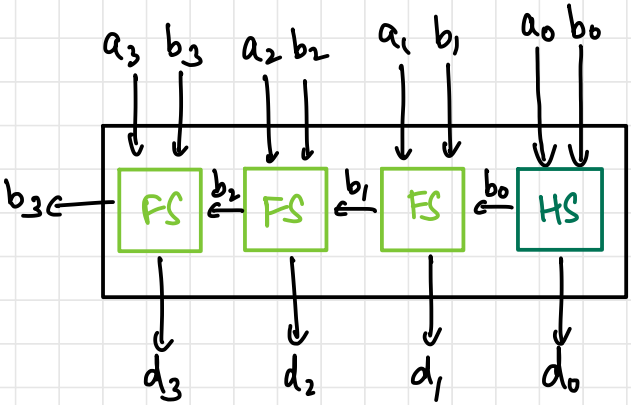
Example 13 – HW

Perform binary subtraction (not addition) and draw logic circuits

$$\begin{array}{r} {\scriptstyle 0\ \ 1} \\ 1\ 0\ 0\ 1 \\ -\ 0\ 0\ 1\ 1 \\ \hline 0\ 1\ 1\ 0 \end{array}$$

$$\begin{array}{r} 9 \\ -\ 3 \\ \hline 6 \end{array}$$

$$\begin{array}{r} {\scriptstyle 1\ \ 1} \\ 1\ 0\ 0\ 1 \\ -\ 0\ 0\ 1\ 1 \\ \hline 0\ 1\ 1\ 0 \end{array}$$



$6 - 3 = 3$

$$\begin{array}{r} {\scriptstyle 1\ \ 1} \\ 0\ 1\ 1\ 0 \\ -\ 0\ 0\ 1\ 1 \\ \hline 0\ 0\ 1\ 1 \end{array}$$

$5 - 2 = 3$

$$\begin{array}{r} {\scriptstyle 1} \\ 0\ 1\ 0\ 1 \\ -\ 0\ 0\ 1\ 0 \\ \hline 0\ 0\ 1\ 1 \end{array}$$

$$a - b_{in} - b$$

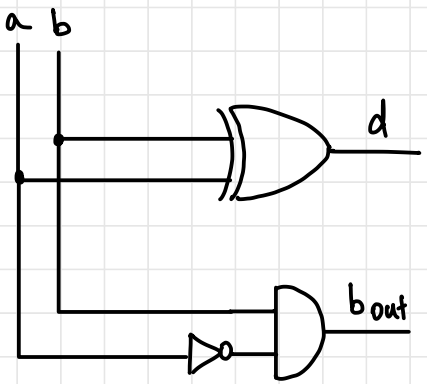
HALF SUBTRACTOR

a	b	d	b_{out}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

SOP Formulas

$$d = \bar{a}b + a\bar{b} = a \oplus b$$

$$b_{out} = \bar{a}b$$



FULL SUBTRACTOR

a	b	c=bin	d	b_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$a-b-b_{in}$ ⟶
$= -1\ (+2)$
$= 1$

$a-b-b_{in}$ ⟶
$= -1\ (+2)$
$= 1$

$a - b_{in} - b$

if $a < b + b_{in}$

2 is added to current digit

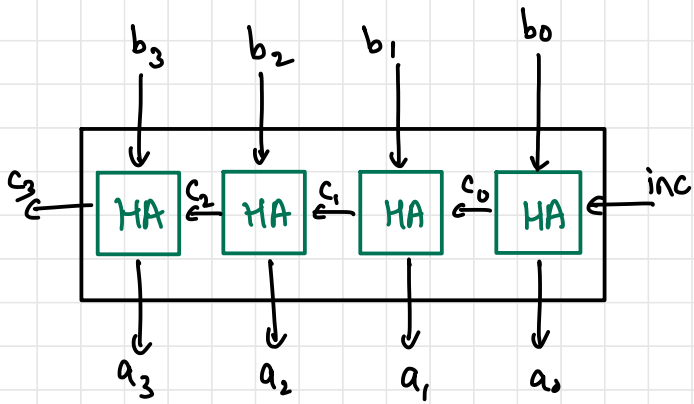
$$d = a \oplus b \oplus c$$

$$b_{out} = \bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{a}bc + abc$$
$$= bc(a+\bar{a}) + \bar{a}b(c+\bar{c}) + \bar{a}c(b+\bar{b})$$

$$b_{out} = bc + \bar{a}b + \bar{a}c$$

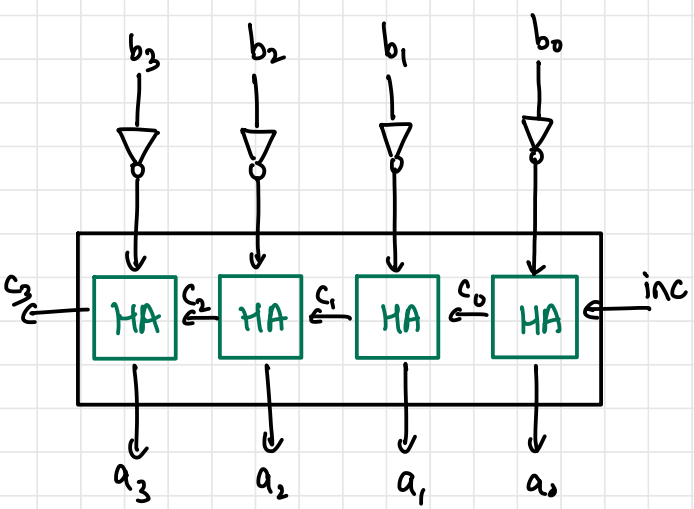
INCREMENT LOGIC CIRCUIT

- b + 1

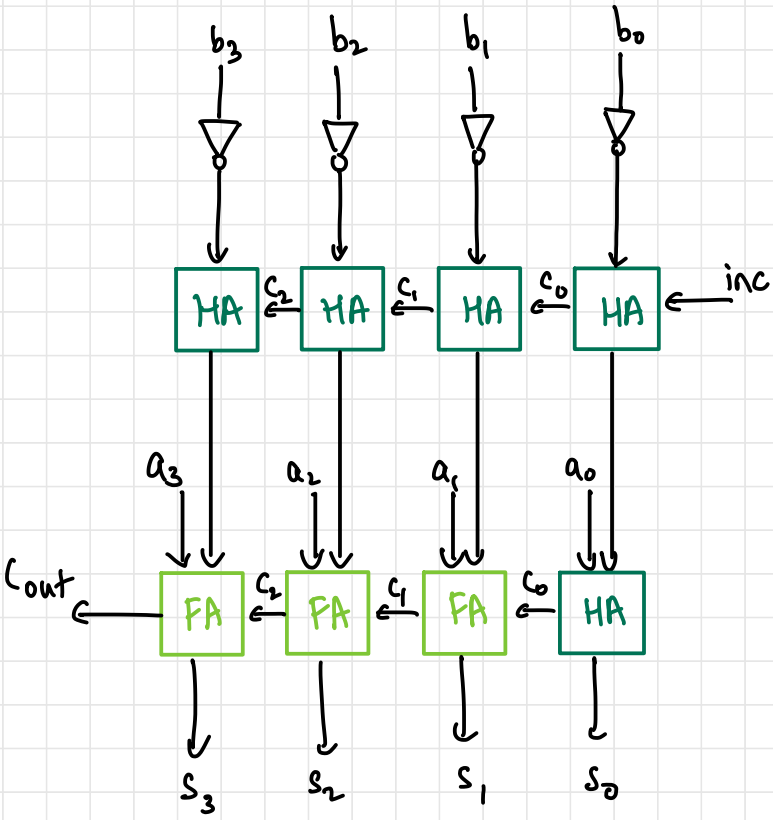


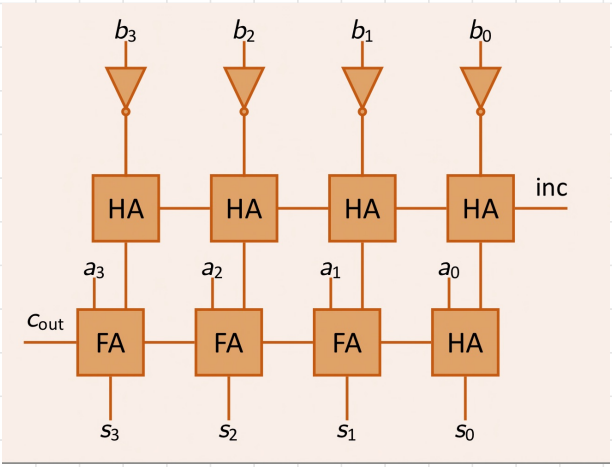
TWO'S COMPLEMENT

- b inverted +1



TWO'S COMPLEMENT SUBTRACTOR

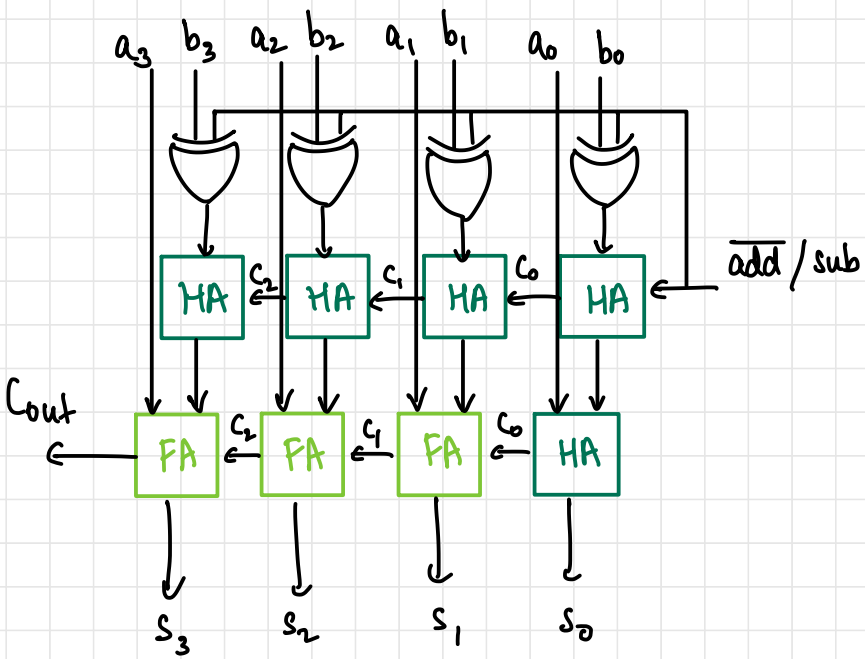




TWO'S COMPLEMENT ADDER / SUBTRACTOR

XOR as Controlled Inverter

inv	a	y
0	0	0
0	1	1
1	0	1
1	1	0



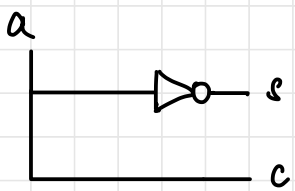
INCREMENTOR WITHOUT INC INPUT

first incrementor

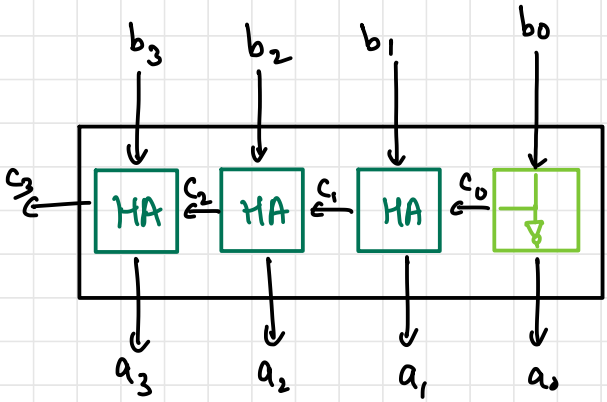
a	s	c
0	1	0
1	0	1

$s = \bar{a}$

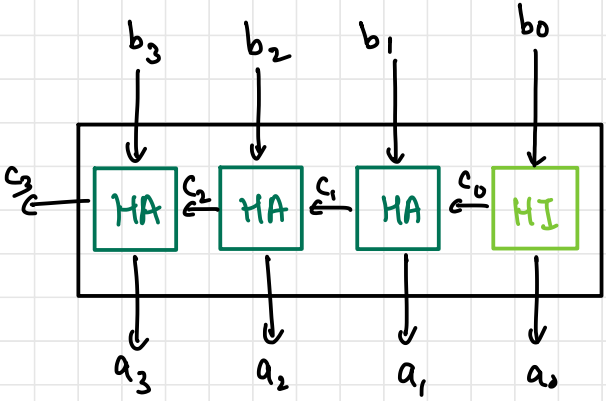
$c = a$



half incrementor
HA



OR



DECREMENTOR

- $b - 1 = b + (-1)$

TWO'S COMPLEMENT DECREMENTOR

- $-1 = (1111)$

```
  1 1
  0 1 1 0   6
+ 1 1 1 1  +(-1)
-----------
1 0 1 0 1   5
```

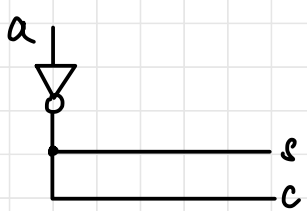
no overflow

```
    1
  0 1 1 0   6
- 0 0 0 1  -(1)
-----------
  0 1 0 1   5
```

Half 2's Complement Decrementor

a	s	c
0	1	1
1	0	0

$$s = \bar{a}$$
$$c = \bar{a}$$



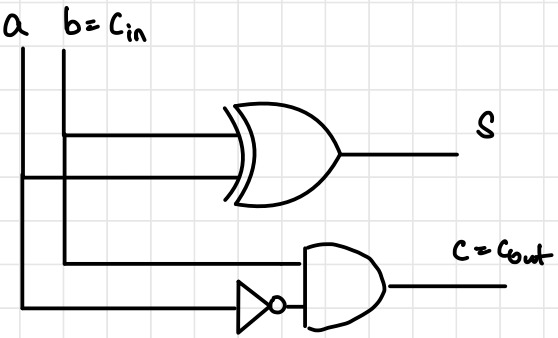
Full 2's Complement Decrementor

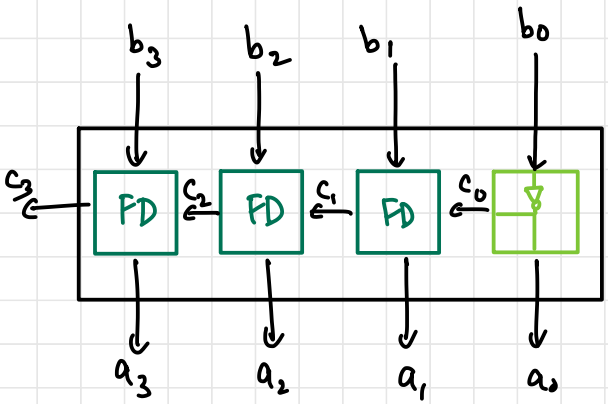
a	$C_{in}=b$	S	C_{out}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$S = \bar{a}b + a\bar{b}$ (XOR)

$C = \bar{a}b$

Full decrementor FD

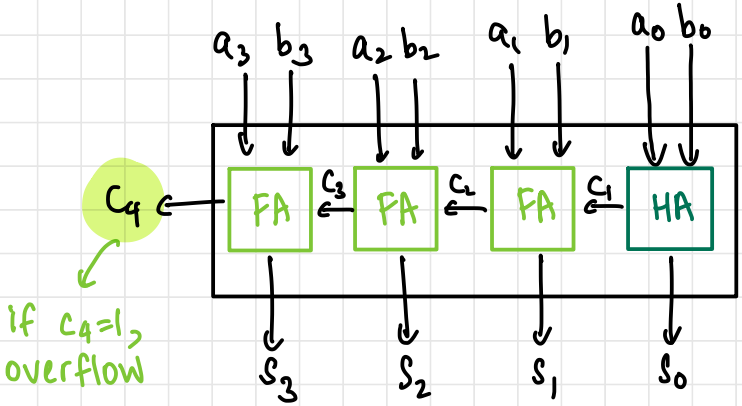




overflow

- Logic circuits have fixed bit widths
- Eg: 64 bit processor
- if result of an operation does not fit within the bit width

SIMPLE ADDITION



TWO'S COMPLEMENT ADDITION

- three cases

Case I – no overflow

one +ve, one -ve

a +ve, b -ve

case i

$|a| > |b|$

result : +ve

case ii

$|b| > |a|$

result : -ve

case iii

$|a| = |b|$

result : 0

Case II – overflow occurs

both +ve

↳ MSB is 1 (C_{msb} is 1) ← should not happen in 2's comp +ve

↳ C_{msb} is 1 and C_{msb+1} is 0

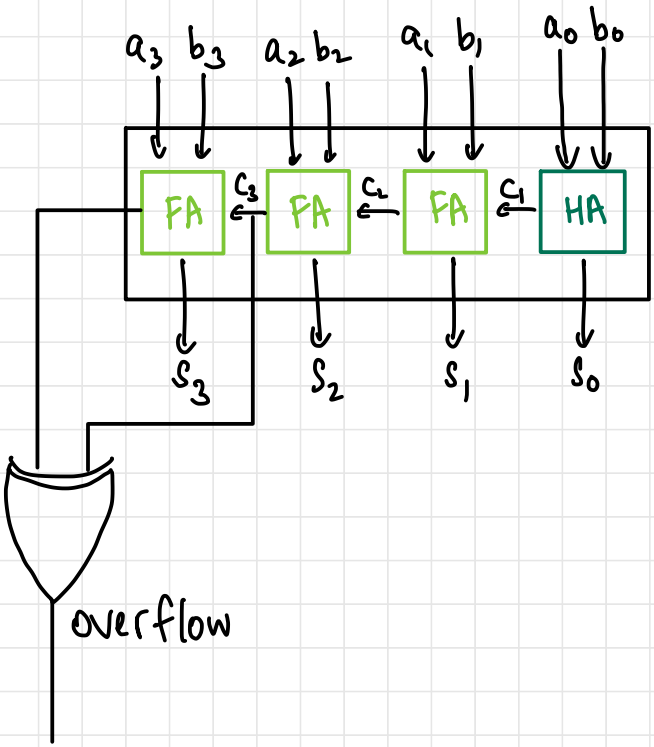
Case III – overflow occurs

both -ve

↳ msb is 0 (C_{msb} is 0)

↳ C_{msb} is 0 and C_{msb+1} is 1

$$\text{overflow} = C_{msb} \oplus C_{msb+1}$$



Question

Design combinational circuit for 3 I/P & 1 O/P. The O/P is 1 when the binary value of the input is less than 3. The O/P is 0 otherwise

a	b	c	y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

$$y = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{c}$$

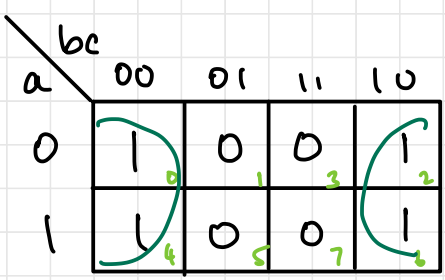
$$y = \bar{a}\bar{b} + \bar{a}b\bar{c} + \bar{a}\bar{b}\bar{c}$$
$$= \bar{a}\bar{b} + \bar{a}\bar{c}$$

$$y = \bar{a}\bar{b} + \bar{a}\bar{c}$$

Question

Design combinational circuit for 3 I/P & 1 O/P. The O/P is 1 when the binary value of the input is even. The O/P is 0 otherwise

a	b	c	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



$$y = \bar{c}$$

Question

Design combinational circuit for 3 I/P x, y, z and 3 O/Ps a, b, c. When I/P is 0, 1, 2 or 3, O/P is one greater than I/P. When I/P is 4, 5, 7 or 7, O/P is 2 less than I/P.

x	y	z	a	b	c
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	0	1

(a)

x \ yz	$\bar{y}\bar{z}$ 00	$\bar{y}z$ 01	yz 11	$y\bar{z}$ 10
$\bar{x}$ 0	0 (0)	0 (1)	1 (3)	0 (2)
x 1	0 (4)	0 (5)	1 (7)	1 (6)

$a = yz + xy$

(b)

x \ yz	$\bar{y}\bar{z}$ 00	$\bar{y}z$ 01	yz 11	$y\bar{z}$ 10
$\bar{x}$ 0	0 (0)	1 (1)	0 (3)	1 (2)
x 1	1 (4)	1 (5)	0 (7)	0 (6)

$b = \bar{x}y\bar{z} + \bar{y}z + x\bar{y}$

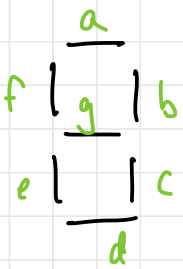
(b)

x \ yz	$\bar{y}\bar{z}$ 00	$\bar{y}z$ 01	yz 11	$y\bar{z}$ 10
$\bar{x}$ 0	1 (0)	0 (1)	0 (3)	1 (2)
x 1	0 (4)	1 (5)	1 (7)	0 (6)

$c = xz + \bar{x}\bar{z}$

Question

Logic circuit for 7-segment display



a	b	c	d	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9