## Six-variable Karnaugh Map: $f(\underbrace{uvw}_{8}\ \underbrace{xyz}_{8})$

8×8 Table : cell # for decimal notation

| uvw \ xyz | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
|---|---|---|---|---|---|---|---|---|
| 000 | 0 | 1 | 3 | 2 | 6 | 7 | 5 | 4 |
| 001 | 8 | 9 | 11 | 10 | 14 | 15 | 13 | 12 |
| 011 | 24 | 25 | 27 | 26 | 30 | 31 | 29 | 28 |
| 010 | 16 | 17 | 19 | 18 | 22 | 23 | 21 | 20 |
| 110 | 48 | 49 | 51 | 50 | 54 | 55 | 53 | 52 |
| 111 | 56 | 57 | 59 | 58 | 62 | 63 | 61 | 60 |
| 101 | 40 | 41 | 43 | 42 | 46 | 47 | 45 | 44 |
| 100 | 32 | 33 | 35 | 34 | 38 | 39 | 37 | 36 |

---

Find prime implicants : $\left\{ \begin{array}{l} \text{- Karnaugh map (visual)} \\ \text{- Quine-McCluskey (algorithmic)} \end{array} \right.$

(algorithmic) → computational purpose

# Quine - McCluskey Method:

Given a logic function: $f(w,x,y,z) = \sum m(1,3,4,5,7,8,15)$

(sum of products) Recall: decimal notation: 4 variables
$\rightarrow$ 16 entries: 0 (0000), 1 (0001), ... 15 (1111)

$f(w,x,y,z) = \bar{w}\bar{x}\bar{y}z + \bar{w}\bar{x}yz + \bar{w}x\bar{y}\bar{z} + \bar{w}x\bar{y}z$

$+ \bar{w}xyz + w\bar{x}\bar{y}\bar{z} + \cancel{wxyz}\; wxyz$

| | wxyz | f |
|---|---|---|
| 0 | 0000 | |
| 1 | 0001 | 1 |
| 2 | 0010 | |
| 3 | 0011 | 1 |
| 4 | 0100 | 1 |
| 5 | 0101 | 1 |
| 6 | 0110 | |
| 7 | 0111 | 1 |
| 8 | 1000 | 1 |
| 9 | 1001 | |
| 10 | 1010 | |
| 11 | 1011 | |
| 12 | 1100 | |
| 13 | 1101 | |
| 14 | 1110 | |
| 15 | 1111 | 1 |

| | Column #1 | | Column #2 | | Column #3 |
|---|---|---|---|---|---|
| 1 | $\bar{w}\bar{x}\bar{y}z$ ✓ | 13 | $\bar{w}\bar{x}z$ ✓ | 13,57 | $\bar{w}z$ |
| 3 | $\bar{w}\bar{x}yz$ ✓ | 15 | $\bar{w}\bar{y}z$ ✓ | 15,37 | |
| 4 | $\bar{w}x\bar{y}\bar{z}$ ✓ | 37 | $\bar{w}yz$ ✓ | | |
| 5 | $\bar{w}x\bar{y}z$ ✓ | 45 | $\bar{w}x\bar{y}$ | | |
| 7 | $\bar{w}xyz$ ✓ | 57 | $\bar{w}xz$ ✓ | | |
| 8 | $w\bar{x}\bar{y}\bar{z}$ | 715 | $xyz$ | | |
| 15 | $wxyz$ ✓ | | | | |

1) Use property: comparing two products:
$\boxed{AB + \bar{A}B = B}$ (eliminate the only different variable in the two adding products)

2) Put a check mark on non-prime implicants (since they imply a simpler implicant)

3) Repeat comparison: 13, 14, 15, ... | 57, 58, 515.
34, 35, 37, 38 ... | 78, 715
45, 47, 48, 415.

Note 8 is still a prime implicant of the function!

4) Repeat comparing process in column # 2

13 & 57 implies $\bar{w}z$, they are not prime implicants → add a check mark to both.

15 & 37 implies $\bar{w}z$ also, add check marks.

⇒ Prime implicants for $f(w,x,y,z)$:

$w\bar{x}\bar{y}\bar{z}$, $\bar{w}x\bar{y}$, $xyz$, $\bar{w}z$

---

Can make this process more systematic? use 01–

| | Column #1 | 01– | index (# of ones) |
|---|---|---|---|
| 1 | $\bar{w}\bar{x}\bar{y}z$ | 0001 | 1 |
| 3 | $\bar{w}\bar{x}yz$ | 0011 | 2 |
| 4 | $\bar{w}x\bar{y}\bar{z}$ | 0100 | 1 |
| 5 | $\bar{w}x\bar{y}z$ | 0101 | 2 |
| 7 | $\bar{w}xyz$ | 0111 | 3 |
| 8 | $w\bar{x}\bar{y}\bar{z}$ | 1000 | 1 |
| 15 | $wxyz$ | 1111 | 4 |

| | Column #1 | index (# of ones) |
|---|---|---|
| 1 | 0001 | 1 |
| 4 | 0100 | 1 |
| 8 | 1000 | 1 |
| 3 | 0011 | 2 |
| 5 | 0101 | 2 |
| 7 | 0111 | 3 |
| 15 | 1111 | 4 |

By using "01–" notation and grouping entries based on their index, we can reduce the number of needed comparisons: we only need to compare entries belonging to groups ~~whose~~ indices differ by one. No need to compare group of index 1 with group of index 3 or 4, etc. Only compare groups 1&2; 2&3; 3&4

| | Column #1 | (index) | |
|---|---|---|---|
| 1 | 0001 | (1) | ✓ |
| 4 | 0100 | (1) | ✓ |
| 8 | 1000 | (1) | |
| 3 | 0011 | (2) | ✓ |
| 5 | 0101 | (2) | ✓ |
| 7 | 0111 | (3) | ✓ |
| 15 | 1111 | (4) | ✓ |

| | Column #2 | (index) |
|---|---|---|
| 13 | 00-1 | (1) |
| 15 | 0-01 | (1) |
| 37 | 0-11 | (2) |
| 57 | 01-1 | (2) |
| 715 | -111 | (3) |
| 45 | 010- | (1) |

| | Column #2 | (index) | |
|---|---|---|---|
| 13 | 00-1 | (1) | ✓ |
| 15 | 0-01 | (1) | ✓ |
| 45 | 010- | (1) | |
| 37 | 0-11 | (2) | ✓ |
| 57 | 01-1 | (2) | ✓ |
| 715 | -111 | (3) | |

| | Column #3 | (index) |
|---|---|---|
| 13 & 57 | 0--1 | (1) |
| 15 & 37 | 0--1 | (1) |

| | Prime implicants | |
|---|---|---|
| 8 | 1000 | $(w\bar{x}\bar{y}\bar{z})$ |
| 4,5 | 010- | $(\bar{w}x\bar{y})$ |
| 7,15 | -111 | $(xyz)$ |
| 13 & 57 | 0--1 | $(\bar{w}z)$ |

## Another Example of Quine - McCluskey : an incomplete function

$$f(v,w,x,y,z) = \Sigma m(\underbrace{4,5,9,11,12,14,15,27,30}_{\text{nine ones}}) + dc(\underbrace{1,17,25,26,31}_{\text{five unknowns}})$$

Recall Karnaugh map: unknowns or — were considered as ones → same here!

Nine ones + five dashes → 14 entries or product terms to simplify: First we need to group them by index:

| | m | entry | index (# of ones) |
|---|---|---|---|
| nine ones | 4 | 00100 | 1 |
| | 5 | 00101 | 2 |
| | 9 | 01001 | 2 |
| | 11 | 01011 | 3 |
| | 12 | 01100 | 2 |
| | 14 | 01110 | 3 |
| | 15 | 01111 | 4 |
| | 27 | 11011 | 4 |
| | 30 | 11110 | 4 |
| five unknowns | 1 | 00001 | 1 |
| | 17 | 10001 | 2 |
| | 25 | 11001 | 3 |
| | 26 | 11010 | 3 |
| | 31 | 11111 | 5 |

| Column #1 | | Column #2 |
|---|---|---|
| 4 | 00100 | |
| 5 | 00101 | |
| 9 | 01001 | |
| 12 | 01100 | |
| 11 | 01011 | |
| 14 | 01110 | |
| 15 | 01111 | |
| 27 | 11011 | |
| 30 | 11110 | |

* compare b/w consecutive groups to use $A\bar{B} + AB = A$
* Column #1 needs to include unknown entries as well (see next page)

## Column #1 (index)

| m | | | |
|---|---|---|---|
| 1 | 00001 | ✓ | (1) |
| 4 | 00100 | ✓ | |
| 5 | 00101 | ✓ | (2) |
| 9 | 01001 | ✓ | |
| 12 | 01100 | ✓ | |
| 17 | 10001 | ✓ | |
| 11 | 01011 | ✓ | (3) |
| 14 | 01110 | ✓ | |
| 25 | 11001 | ✓ | |
| 26 | 11010 | | |
| 15 | 01111 | ✓ | (4) |
| 27 | 11011 | ✓ | |
| 30 | 11110 | ✓ | |
| 31 | 11111 | ✓ | (5) |

Column 1 → Column 2: all same except for one digit change, if yes mark those entries as not prime implicants.

## Column #2

| | | | |
|---|---|---|---|
| 1,5 | 00-01 | → Prime | (1) |
| 1,9 | 0-001 | ✓ | |
| 1,17 | -0001 | | |
| 4,5 | 0010- | → Prime | |
| 4,12 | 0-100 | → Prime | |
| 9,11 | 010-1 | ✓ | (2) |
| 9,25 | -1001 | ✓ | |
| 12,14 | 011-0 | → Prime | |
| 17,25 | 1-001 | ✓ | |
| 11,15 | 01-11 | ✓ | (3) |
| 11,27 | -1011 | ✓ | |
| 14,15 | 0111- | ✓ | |
| 14,30 | -1110 | ✓ | |
| 25,27 | 110-1 | ✓ | |
| 26,27 | 1101- | ✓ | |
| 26,30 | 11-10 | ✓ | |
| 15,31 | -1111 | ✓ | (4) |
| 27,31 | 11-11 | ✓ | |
| 30,31 | 1111- | ✓ | |

## Column #3

| | | | |
|---|---|---|---|
| 1,9,17,25 | --001 | → Prime | (1) |
| 9,11,25,27 | -10-1 | → Prime | (2) |
| 9,25,11,27 | -10-1 | | |
| 11,15,27,31 | -1-11 | → Prime | (3) |
| 11,27,15,31 | -1-11 | | |
| 14,15,30,31 | -111- | → Prime | |
| 14,30,15,31 | -111- | | |
| 26,27,30,31 | 11-1- | → Prime | |
| 26,30,27,31 | 11-1- | | |

## Column #4

No match

Our function has 9 prime implicants:

00-01, 0010-, 0-100, 011-0, --001, -10-1, -1-11,

$\bar v \bar w \bar y z$, $\bar v \bar w x \bar y$, $\bar v x \bar y \bar z$, $\bar v w x \bar z$, $\bar x \bar y z$, $w \bar x z$, $w y z$

-111-, 11-1-

$w x y$, $v w y$

# Quine-McKluskey Method:

Two functions:

ⓐ $f(w,x,y,z) = \sum m(1,3,4,5,7,8,15)$

4 prime implicants: $w\bar{x}\bar{y}\bar{z}$, $\bar{w}x\bar{y}$, $xyz$, $\bar{w}z$

ⓑ $f(v,w,x,y,z) = \sum m(4,5,9,11,12,14,15,27,30)$

$+ \; d_c(1,17,25,26,31)$

Nine prime implicants: $\bar{v}\bar{w}\bar{y}z$, $\bar{v}\bar{w}x\bar{y}$, $\bar{v}x\bar{y}\bar{z}$, $\bar{v}wx\bar{z}$, $\bar{x}\bar{y}z$, $w\bar{x}z$, $wyz$, $wxy$, $vwy$

We found all prime implicants for each of these functions, not all are essential. Next task:

1) Find minimum expressions → Build prime implicant Table!
2) Find lowest cost expression

$\boxed{\text{Prime Implicant Table}}$

ⓐ

| | | $m_1$ | $m_3$ | $m_4$ | $m_5$ | $m_7$ | $m_8$ | $m_{15}$ |
|---|---|---|---|---|---|---|---|---|
| 8 | A: $w\bar{x}\bar{y}\bar{z}$ | | | | | | X | |
| 4,5 | B: $\bar{w}x\bar{y}$ | | | X | X | | | |
| 7,15 | C: $xyz$ | | | | | X | | X |
| 1,3,5,7 | D: $\bar{w}z$ | X | X | | X | X | | |

→ Minterms in columns
→ Prime implicants in rows

• A prime implicant is essential if it is the only implicant for some min term

→ Essential prime implicants: $A(m_8)$, $B(m_4)$, $C(m_{15})$, $D(m_1 \text{ or } m_3)$

→ Min. expression: a combination of all essential prime implicants that covers all min terms (seven min terms in this example):

$f_{min} = A + B + C + D$ only one possibility to cover all min terms → we are done!

$f_{min}(w,x,y,z) = w\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y} + xyz + \bar{w}z$

b

## Prime implicant table

| | | | $m_4$ | $m_5$ | $m_9$ | $m_{11}$ | $m_{12}$ | $m_{14}$ | $m_{15}$ | $m_{27}$ | $m_{30}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1,9,17,25 | A | $\bar{x}\bar{y}z$ | | | X | | | | | | |
| 9,11,25,27 | B | $w\bar{x}z$ | | | X | X | | | | X | |
| 11,15,27,31 | C | $wyz$ | | | | X | | | X | X | |
| 14,15,30,31 | D | $wxy$ | | | | | | X | X | | X |
| 26,27,30,31 | E | $vwy$ | | | | | | | | X | X |
| 1,5 | F | $\bar{v}\bar{w}\bar{y}z$ | | X | | | | | | | |
| 4,5 | G | $\bar{v}\bar{w}x\bar{y}$ | X | X | | | | | | | |
| 4,12 | H | $\bar{v}x\bar{y}\bar{z}$ | X | | | | X | | | | |
| 12,14 | I | $\bar{v}wx\bar{z}$ | | | | | X | X | | | |

- Do not include unknown min terms in columns.

Essential prime implicants: the only prime implicant for some min term: ~~A ($m_9$)~~ none. We need to find different min. combinations that cover all min terms ($m_4$ $m_5$ $m_9$ $m_{11}$ $m_{12}$ $m_{14}$ $m_{15}$ $m_{27}$ $m_{30}$)

$f_I = B + D + G + H$

↓

For lower cost, look for prime implicant that covers most number of min terms: B covers 3 min terms, D covers 3 other min terms, (covered: $m_9, m_{11}, m_{14}, m_{15}, m_{27}, m_{30}$)

Need to cover now: $m_4, m_5, m_{12}$

G covers ($m_4, m_5$)

H or I covers ($m_{12}$)

$$f_{II} = B + D + G + I$$

Observation: in example (a) all prime implicants were used to cover the min terms. In example (b) only some prime implicants were needed to cover all min terms.

$$f_{III} = B + D + F + H$$

$$f_{IV} = B + C + E + G + I$$

...

Observation: $f_I, f_{II}, f_{IV}$ have lowest costs since we found them maximizing minterm coverage for each prime implicant:

$$f_I(v,w,x,y,z) = w\bar{x}z + wxy + \bar{v}\bar{w}x\bar{y} + \bar{v}x\bar{y}\bar{z}$$

$$f_{II}(v,w,x,y,z) = w\bar{x}z + wxy + \bar{v}\bar{w}x\bar{y} + \bar{v}wx\bar{z}$$

$$f_{III}(v,w,x,y,z) = w\bar{x}z + wxy + \bar{v}\bar{w}\bar{y}z + \bar{v}x\bar{y}\bar{z}$$

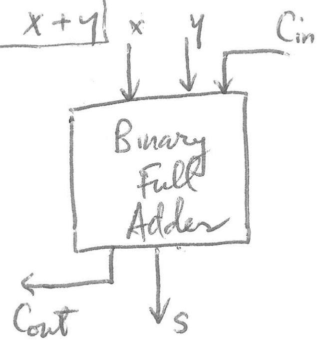
# Ch5 Logic Design with MSI Components & PLDs.

MSI : Medium-scale integrated circuits (10 - 100 gates)

PLD : Programmable Logic Devices.

## Binary Adders & Subtracters:

$\boxed{\text{Binary Full Adder: } x+y}$




( Carry-in: for example from a lower-place-digit addition )

(Carry-out, for next digit addition)

Truth Table:

| x | y | $C_{in}$ | 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 |

( 0+0+0 = 0 with nothing to carry out )

Now: Karnaugh Map → simplified function for logic gates.

$S$

| $x \backslash y\,C_{in}$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | $\boxed{1}$ | 0 | $\boxed{1}$ |
| 1 | $\boxed{1}$ | 0 | $\boxed{1}$ | 0 |

$C_{out}$

| $x \backslash y\,C_{in}$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 (A) | 0 |
| 1 | 0 | 1 (B) | 1 (A, B, C) | 1 (C) |

3 groups of 2 ones → 3 terms of 2 literals

Karnaugh map: largest blocks or groups of ones (8, 4, 2, 1)

$$S = \underbrace{\bar{x}\bar{y}C_{in} + \bar{x}y\bar{C}_{in}} + \underbrace{x\bar{y}\bar{C}_{in} + xyC_{in}}$$

(standard min term canonical; no simplification was possible)

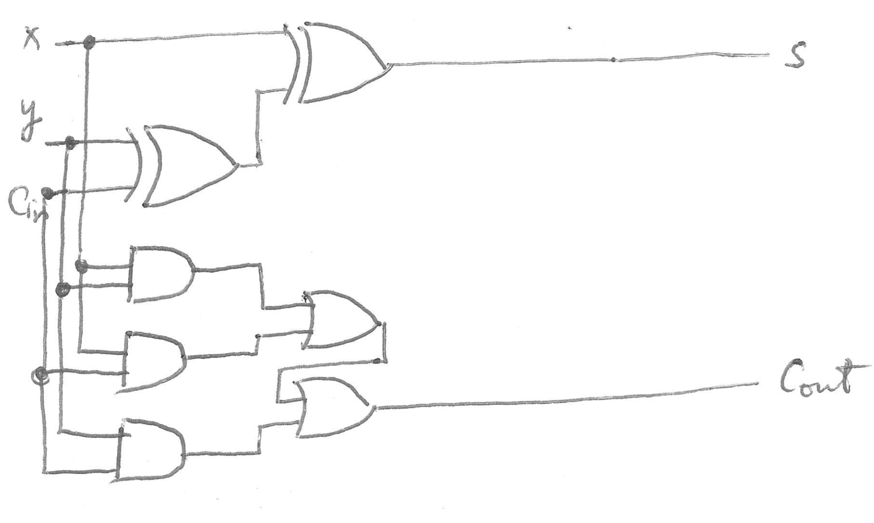
Some saving using XOR gates?
$$\begin{cases} ab' + a'b = a \oplus b \;\|\; a'b' + ab = \overline{(a \oplus b)} \\ a\bar{b} + \bar{a}b = a \oplus b \;\|\; \bar{a}\bar{b} + ab = \overline{a \oplus b} \end{cases}$$

$$S = \bar{x}(\underbrace{\bar{y}C_{in} + y\bar{C}_{in}}) + x(\overline{\bar{y}\bar{C}_{in} + yC_{in}})$$
$$= \bar{x}\cdot(y \oplus C_{in}) + x\cdot\overline{(y \oplus C_{in})}$$
$$= x \oplus (y \oplus C_{in})$$

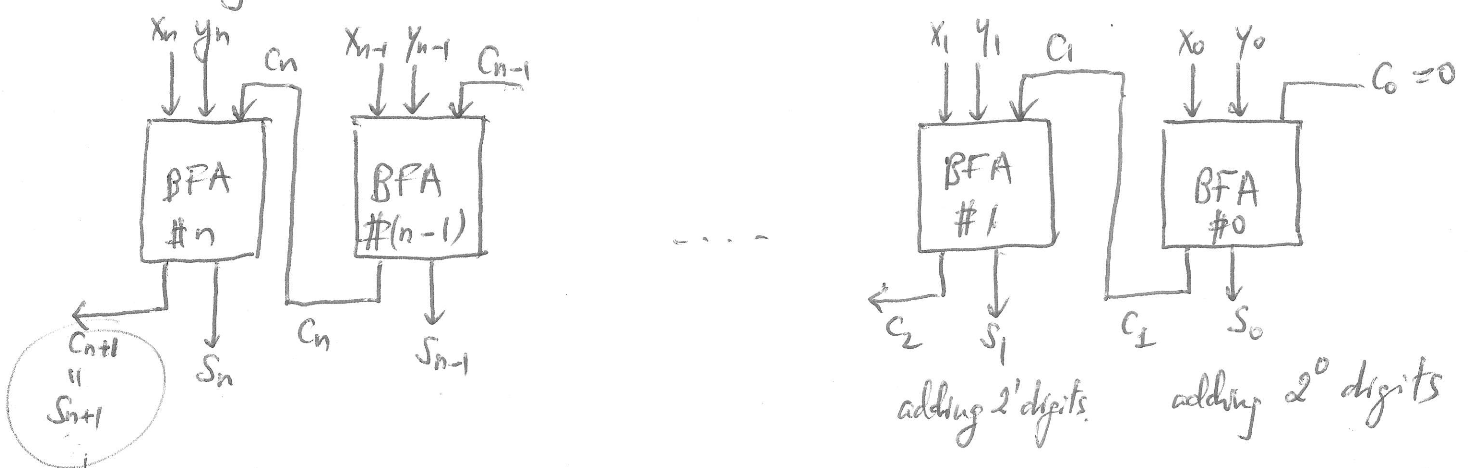
---

$$C_{out} = \overset{(A)}{yC_{in}} + \overset{(B)}{xC_{in}} + \overset{(C)}{xy}$$

Binary Full Adder: Logic diagram:



To add two n-digits binary numbers : combine n single-digit binary adders <u>in series</u>: first add the lowest digits ($C_{in} = 0$)




<u>Note:</u> when adding two n-digits binary numbers we will get in general: a (n+1) digit binary number:

$$\begin{array}{r} 10 \\ +\ 11 \\ \hline 101 \end{array}$$

the (n+1) digit is also the last carry-out!

So we have $S_{n+1} = C_{n+1}$

---

To subtract two n-digit binary numbers : can make BFS (Binary Full Subtracter) or add the 2's-complement of the second number to the first number : $N_1 - N_2 = N_1 + \overline{\overline{N_2}}$.

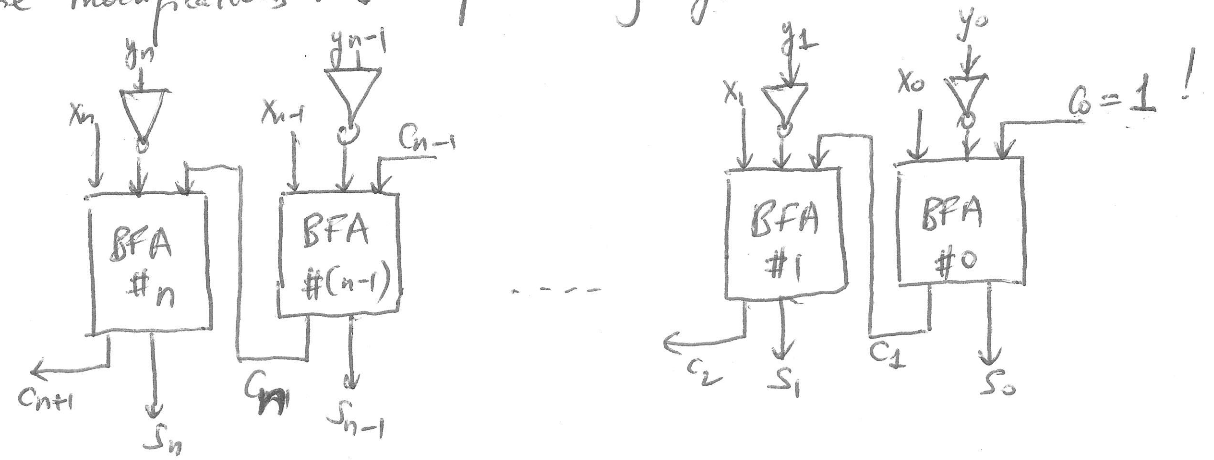
<u>Note:</u> a 2's complement of a binary is also a <u>1's complement</u> plus 1

{ 1's complement of 0 is 1
{ 1's complement of 1 is 0

<u>Note:</u> $\overline{N_2}$ is 1's complement of $N_2$

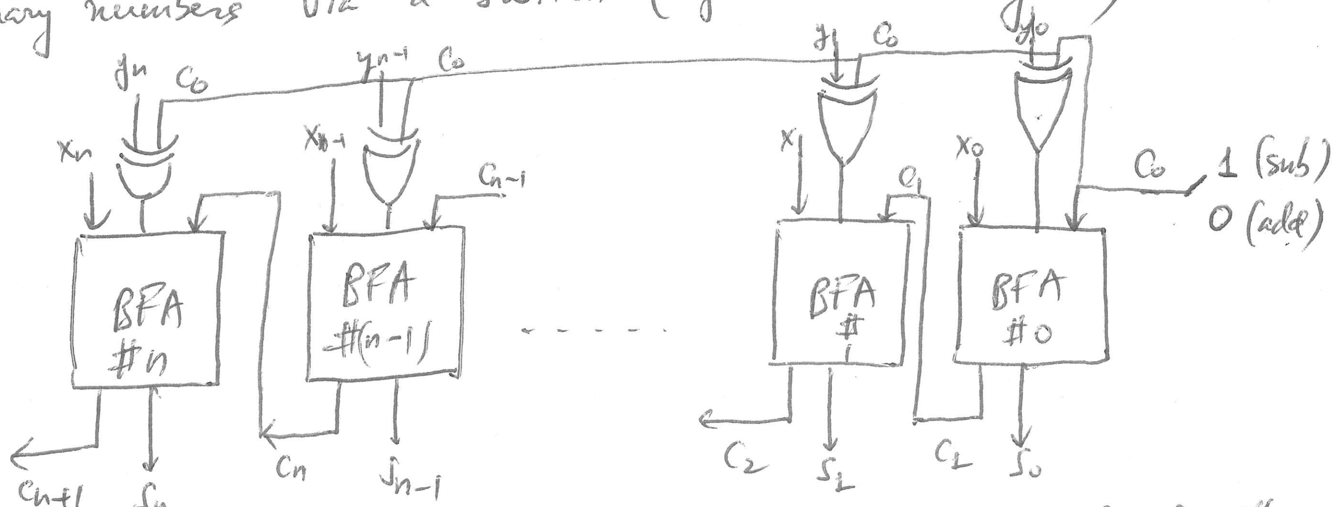
$\overline{\overline{N_2}}$ is 2's complement of $N_2$ (in base 2) ~ the 10's complement of $N_2$ in base 10, etc..

Using these properties we can perform a subtraction for two n-digits binary numbers using a similar combination of BFA's with some modifications: 1-complementing $y_0$ & add one by setting $C_0 = 1$




---

We can combine these into one combination that allows performing addition or subtraction b/w the same two n-digit binary numbers via a switch (of the 1st carry in)




Note: XOR Gate will produce $y_0$ if $C_0 = 0$ or $\overline{y_0}$ if $C_0 = 1$

example: $y_0 = 1$ a) if $C_0 = 0$ (add) → XOR gives 1
b) if $C_0 = 1$ (sub) → XOR gives 0 or the 1-complement of 1

Note: this works in sequence: need to wait until the carry-out is produce before the next add/sub → "ripple effect".