February 6, 2026

* Truth tables

- An exhaustive list of all possible input values and their corresponding output values.
  - Same output for a particular input.

* Example

a_1	a_0	$f(a_1, a_0)$
0	0	0
0	1	0
1	0	0
1	1	1

input outputs

* Example: fn. that adds a 1 to an unsigned number (3-bit)

a_2	a_1	a_0	b_3	b_2	b_1	b_0
0	0	0	0	0	0	1
0	0	1	0	0	1	0
0	1	0	0	0	1	1
0	1	1	0	1	0	0
1	0	0	0	1	0	1
1	0	1	0	1	1	0
1	1	0	0	1	1	1
1	1	1	1	0	0	0

input output

indicates an overflow.

$b_0 = a_0'$

$b_1 = a_1 \oplus a_0$

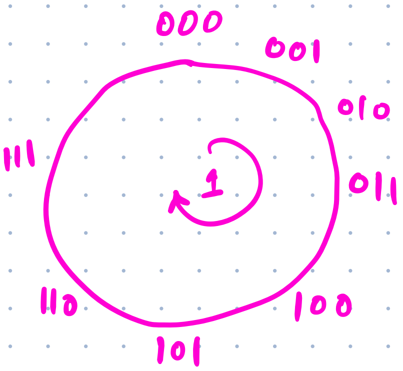
XOR

$b_2 =$

$b_3 = a_2 \cdot a_1 \cdot a_0$

* Truth Table for 1's compliment (3-bit)

a_2	a_1	a_0	b_2	b_1	b_0	b_2	b_1	b_0
0	0	0	1	1	1	0	0	0
0	0	1	1	1	0	1	1	1
0	1	0	1	0	1	1	1	0
0	1	1	1	0	0	1	0	1
1	0	0	0	1	1	1	0	0
1	0	1	0	1	0	0	1	1
1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	0	0	1
input			1's compliment			2's compliment		



* Boolean Algebra

- variables (true or false) — (1) over true, (0) over false
- operators
- axioms : $ab = ba$, $a(b+c) = ab + ac$, $(ab)' = a' + b'$

* Operators :

NOT, AND, OR, NAND, NOR, XOR, ...

* Boolean Algebric Expression's

$$f(x,y,z) = xy + xz + yz$$

x	y	z	xy	xz	yz	$xy+xz+yz$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	0	0	0
1	0	1	0	1	0	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1
input			intermediate steps			output

* Example: $f(x,y,z) = xy' + xz' + yz$ ← To do

if and only if

(i) A Boolean expression is satisfiable iff at least one combination of the input values makes the expression true.

x	y	xy'	$xx'yy'$	$(x+y')x'y$
0	0	0	0	0
0	1	0	0	0
$\boxed{1 \quad 0}$		$\boxed{1}$	0	0
1	1	0	0	0

satisfiable expression

(ii) Unsatisfiable

(iii) Universally valid Boolean expression.

x	y	$x'y + x'y' + x$
0	0	1
0	1	1
1	0	1
1	1	1

universally valid

(iv) Two Boolean expressions are logically equivalent iff for each combination of the variable values these produce the same output.

x	y	$x'y'$	$(x+y)'$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

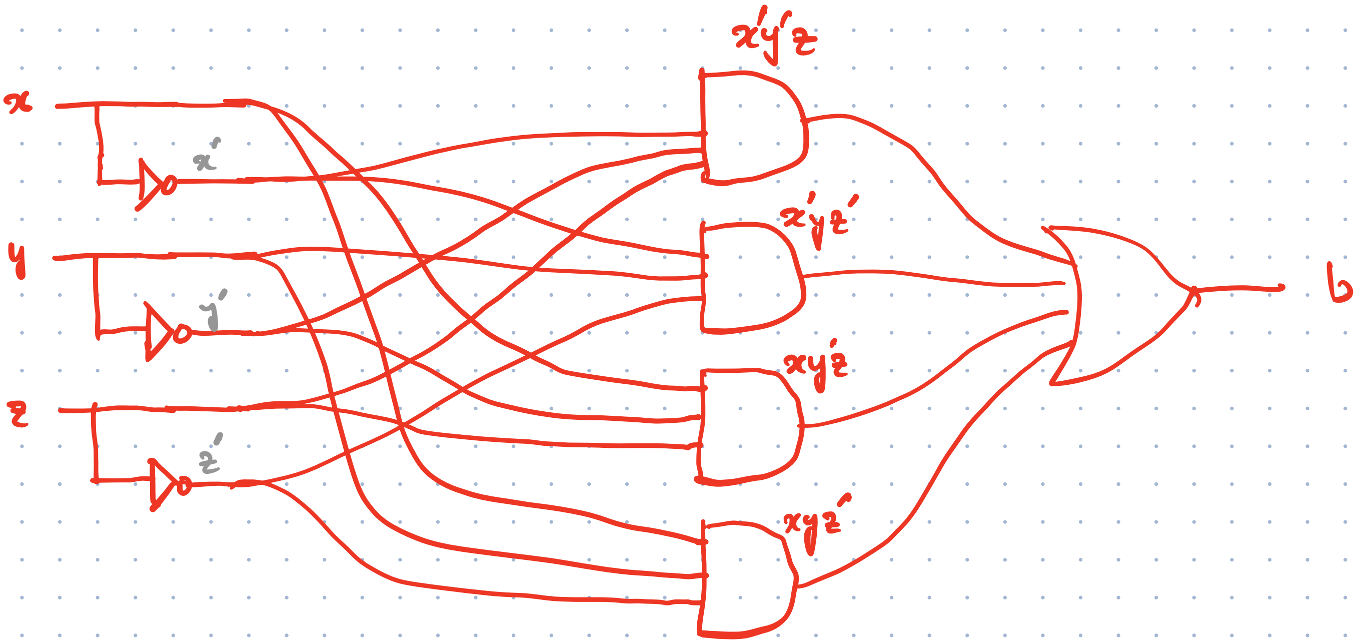
logically equivalent.

* Sum of Product form (SOP)

x	y	z	b	
0	0	0	0	
0	0	1	1	$x'y'z$
0	1	0	1	$x'yz'$
0	1	1	0	
1	0	0	0	
1	0	1	1	$xy'z$
1	1	0	1	xyz'
1	1	1	0	

$$b = \underbrace{x'y'z + x'yz' + xy'z + xyz'}_{\text{SOP}}$$

$$b = x'y'z + x'yz' + xy'z + xyz'$$



(i) Easy to create SOP from a truth table

(ii) Easy to create circuit diagram using AND, OR, NOT

(iii) ↓ Inefficient, $\underbrace{\text{may not use the fewest number of gates.}}_{\text{impacts latency/speed \& power}}$

* Circuit that takes in a 3-bit signed integer in 2's compliment form and detects whether its a negative number.

a_2	a_1	a_0	b	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	1	$a_2 a_1' a_0'$
1	0	1	1	$a_2 a_1' a_0$
1	1	0	1	$a_2 a_1 a_0'$
1	1	1	1	$a_2 a_1 a_0$

input

2's compliment -ve or +ve

SOP

$$b = a_2 a_1' a_0 + a_2 a_1' a_0 + a_2 a_1 a_0' + a_2 a_1 a_0$$

// logically equivalent

$$b = a_2$$

much simpler.

Exercise: Multiply $\underbrace{000011}_{\text{6-bits}}$ with $\underbrace{000101}_{\text{6-bits}}$.

$$000011 = 2^1 + 2^0$$

$$\underset{(5)}{\{000101\}} \times (2^1 + 2^0) = \underset{\overset{\leftarrow 1}{2^1}}{\underline{001010}} + \underset{2^0}{\underline{000101}}$$

$$= 001111$$

Booth's Method

$Q = 000101$

$M = 000011$

$-M = 111101$

$A = 000000$

$Q_{-1} = 0$

	A	Q	Q_{-1}
	0 0 0 0 0 0	0 0 0 1 0 1	0
	1 1 1 1 0 1	0 0 0 1 0 1	0
S1 →	1 1 1 1 1 0	1 0 0 0 1 0	1
	0 0 0 0 0 1	1 0 0 0 1 0	1
S2 →	0 0 0 0 0 0	1 1 0 0 0 1	0
	1 1 1 1 0 1	1 1 0 0 0 1	0
S3 →	1 1 1 1 1 0	1 1 1 0 0 0	1
	0 0 0 0 0 1	1 1 1 0 0 0	1
S4 →	0 0 0 0 0 0	1 1 1 1 0 0	0
S5 →	0 0 0 0 0 0	0 1 1 1 1 0	0
S_6 →	0 0 0 0 0 0	0 0 1 1 1 1	0

↓

15

A 0 0 0 0 0 0

-M 1 1 1 1 0 1

A-M 1 1 1 1 0 1

①1 1 1 1

A 1 1 1 1 1 0

M 0 0 0 0 1 1

A+M 0 0 0 0 0 1