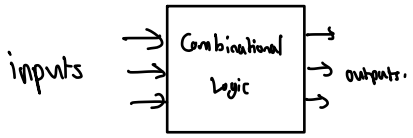
# Combinational Circuits.

- 2 classes of logic circuits

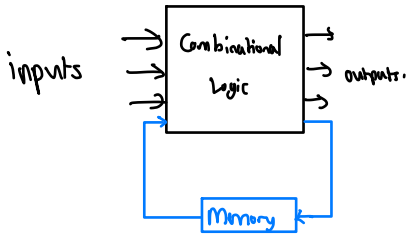
1. Combinational

   - each output depends entirely on the immediate (present) inputs.

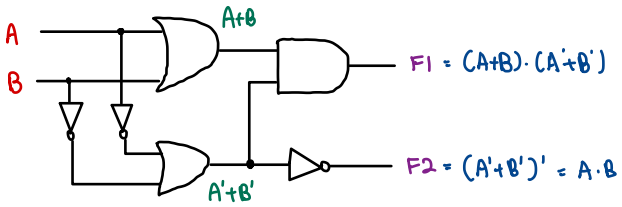



2. Sequential

   - each output depends on both present inputs and state.




- Circuit analysis




Steps:

1. Label the inputs and outputs.
2. Obtain the functions of intermediate points and the outputs.
3. Draw the truth table.

| A | B | (A+B) | (A'+B') | F1 (Sum) | F2 (Carry) |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |

4. Deduce the functionality of the circuit

⇒ Half adder

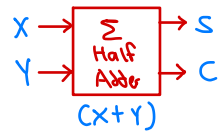
- Circuit Design.
  - Gate-level design method (w/ logic gates) → available as integrated circuit chips.
  - Block-level design method (w/ functional blocks) → available as integrated circuit chips.
  - Main Objective: reduce cost / no. of gates / no. of IC,
    increase speed,
    design simplicity (re-usuable blocks)

- eg. Design Half Adder.

  Design procedure:

  1. State problem: Build a Half Adder
  2. Determine the inputs and outputs of the circuit.

     eg.

     } block diagram.

  3. Draw the truth table

| X | Y | C | S |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

  4. Obtain simplified Boolean functions

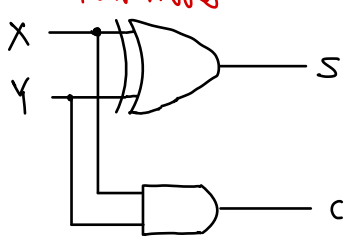
     eg. $C = X \cdot Y$

     $S = X' \cdot Y + X \cdot Y'$

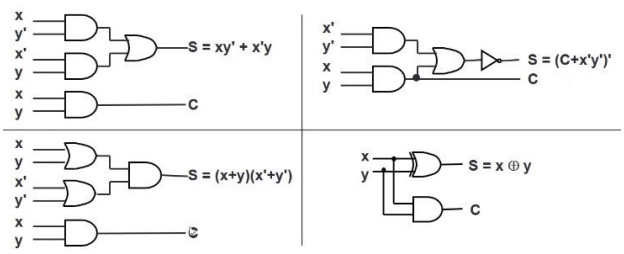
     $= X \oplus Y$

  5. Draw the logic

     Half Adder.

     other ways:

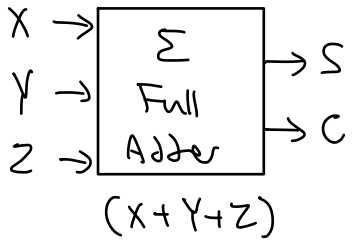
     
     

eg. Design Full Adder

- Half adder adds up only 2 bits
- To add 2 binary numbers, we need to add 3 bits (including the carry).
- i.e.

| | | | | |
|---|---|---|---|---|
| | 1 | 1 | 1 | 0 carry |
| | 0 | 0 | 1 | 1 X |
| + | 0 | 1 | 1 | 1 Y |
| | 1 | 0 | 1 | 0 S |

- Need Full Adder (can be made from 2 half adders)




- Truth Table.

| X | Y | Z | C | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Note:
Z - carry in (to the current position)
C - carry out (to the next position)

- K-map

C

| X\YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |

S

| X\YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

- Simplified SOP form:

$$C = X \cdot Y + X \cdot Z + Y \cdot Z$$

$$S = X' \cdot Y' \cdot Z + X' \cdot Y \cdot Z' + X \cdot Y' \cdot Z' + X \cdot Y \cdot Z$$

↓ Simplify w NOR

$$C = X\cdot Y + X\cdot Z + Y\cdot Z$$
$$= X\cdot Y + (X+Y)\cdot Z$$
$$= X\cdot Y + ((X\oplus Y) + X\cdot Y)\cdot Z$$
$$= X\cdot Y + (X\oplus Y)\cdot Z + X\cdot Y\cdot Z$$
$$= X\cdot Y + (X\oplus Y)\cdot Z$$

$X + Y = (X\oplus Y) + X\cdot Y$

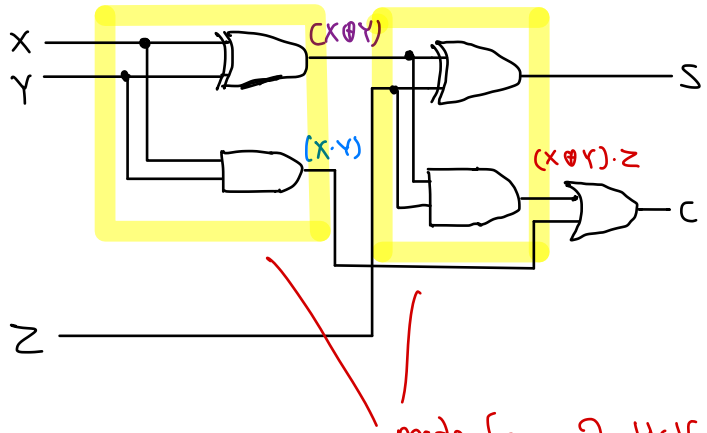
$X \oplus Y = X'\cdot Y + X\cdot Y'$

$(X\oplus Y) + X\cdot Y = X'\cdot Y + X\cdot Y' + X\cdot Y$
$= X'\cdot Y + X\cdot Y' + X\cdot Y + X\cdot Y$
$= (X'+X)\cdot Y + X\cdot(Y'+Y)$
$= Y + X.$

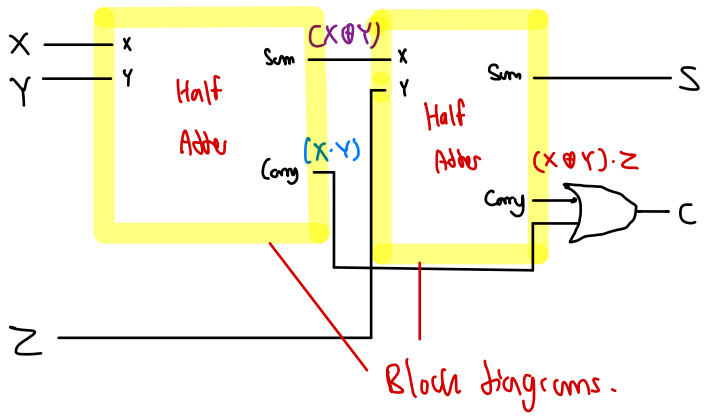
$$S = X'\cdot Y'\cdot Z + X'\cdot Y\cdot Z' + X\cdot Y'\cdot Z' + X\cdot Y\cdot Z$$
$$= X'\cdot(Y'\cdot Z + Y\cdot Z') + X\cdot(Y'\cdot Z' + Y\cdot Z)$$
$$= X'\cdot(Y\oplus Z) + X\cdot(Y\oplus Z)'$$
$$= X\oplus(Y\oplus Z)$$

- Circuit Implementation

$$C = X\cdot Y + (X\oplus Y)\cdot Z$$
$$S = X\oplus(Y\oplus Z)$$
$$= (X\oplus Y)\oplus Z$$




made from 2 Half-Adders (+ an OR gate)




Block diagrams.

- Code converter
  - takes an input code, translates to its equivalent output code.
  - i.e.

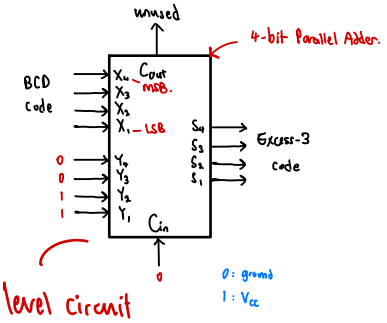

  - eg. BCD to excess-3 code converter

Binary Coded Decimal

Input: BCD code

Output: Excess-3 code.

| Digit | BCD code | Excess-3 code |
|---|---|---|
| 0 | 0000 | 0011 |
| 1 | 0001 | 0100 |
| 2 | 0010 | 0101 |
| 3 | 0011 | 0110 |
| 4 | 0100 | 0111 |
| 5 | 0101 | 1000 |
| 6 | 0110 | 1001 |
| 7 | 0111 | 1010 |
| 8 | 1000 | 1011 |
| 9 | 1001 | 1100 |




∴ Block level circuit

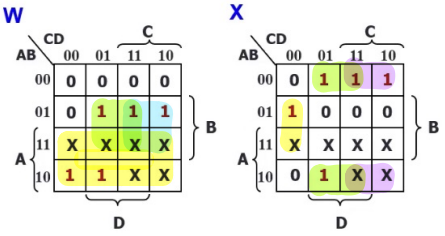
∴ 4 variable truth table

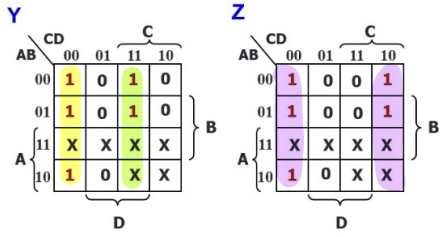
output = input + 3.

| | BCD | | | | Excess-3 | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | W | X | Y | Z |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 10 | 1 | 0 | 1 | 0 | X | X | X | X |
| 11 | 1 | 0 | 1 | 1 | X | X | X | X |
| 12 | 1 | 1 | 0 | 0 | X | X | X | X |
| 13 | 1 | 1 | 0 | 1 | X | X | X | X |
| 14 | 1 | 1 | 1 | 0 | X | X | X | X |
| 15 | 1 | 1 | 1 | 1 | X | X | X | X |

invalid since decimal is from 0-9.

⇒ K-maps




∴ Simplified boolean expression:

$$\therefore W = A + B \cdot C + B \cdot D$$

$$X = B' \cdot C + B' \cdot D + B \cdot C' \cdot D'$$

$$Y = C \cdot D + C' \cdot D$$

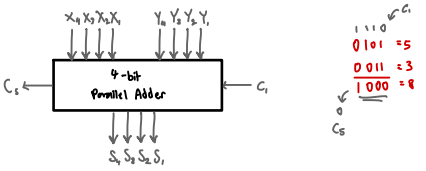
$$Z = D'$$

- Block-Level Design
  - More complex circuits can also be built using block-level method.
  - In general, block-level design method (as opposed to gate-level design) relies on algorithms or formulae of the circuit, which are obtained by decomposing the main problem to sub-problems recursively (until small enough to be directly solved by blocks of circuits)
  - eg. Using 4-bit parallel adders as building blocks, we can create the following:
    1. BCD-to-Excess-3 code converter.
    2. 16-bit Parallel Adder.

---

4-bit parallel adder

- add 2 4-bit numbers together and carry-in, to produce a 5-bit result.







- 5-bit result is sufficient because the largest result is:

$$1111_2 + 1111_2 + 1_2 = 11111_2.$$

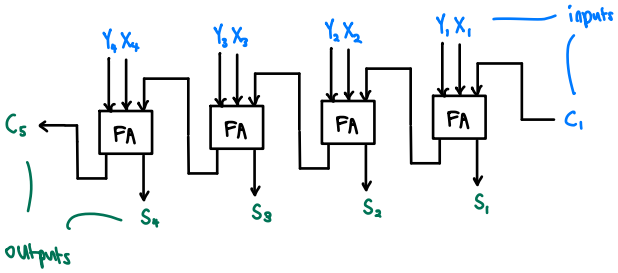
- Gate-level design should not be used here!
  - ⇒ truth table for 9 inputs is too big: $2^9 = 512$ rows!
- Alternative design possible
  - Addition formula for pair of bits (w carry in)

$$C_{i+1} S_i = X_i + Y_i + C_i$$

has the same function as a full adder:

$$C_{i+1} = X_i \cdot Y_i + (X_i \oplus Y_i) \cdot C_i$$
$$S_i = X_i \oplus Y_i \oplus C_i$$

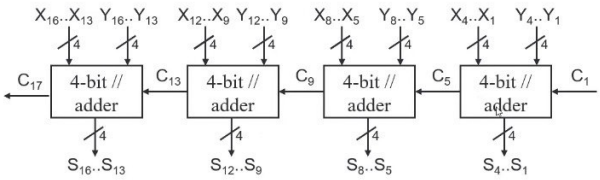
  - ∴ each full adder handles one column of addition:




  - Carry is propagated by cascading the carry from one full adder to the next.
  - Parallel Adder because the inputs are presented simultaneously (in parallel) (also called the Ripple-Carry Adder)

# 16-bit Parallel Adder?

- Larger parallel adders can be built from smaller ones
- eg.



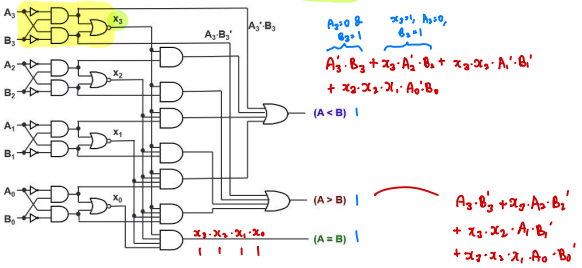

---

- Magnitude Comparator
  - Compares 2 unsigned values A and B, checks if $A>B$, $A=B$, $A<B$.
  - An n-bit magnitude comparator using classical method requires $2^{2n}$ rows in truth table.
  - Exploit regularity in design
    - Compare 2 4-bit unsigned values

$A(a_3a_2a_1a_0)$ and $B(b_3b_2b_1b_0)$

- If $a_3>b_3$, then $A>B$.
  If $a_3<b_3$, then $A<B$.
  If $a_3=b_3$, then if $a_2>b_2$ ...

Let $A = A_3A_2A_1A_0$, $B = B_3B_2B_1B_0$; $x_i = A_i \cdot B_i + A_i' \cdot B_i'$ (1 if $A_i=B_i$, 0 if $A_i \neq B_i$)

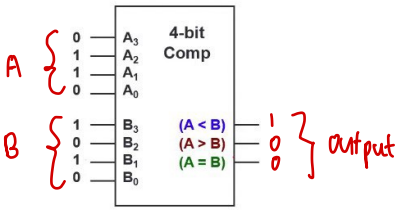



$A_3' \cdot B_3 + x_3 \cdot A_2' \cdot B_2 + x_3 \cdot x_2 \cdot A_1' \cdot B_1' + x_3 \cdot x_2 \cdot x_1 \cdot A_0' \cdot B_0$ ($A_3=0$ & $B_3=1$; $x_3=1$, $A_2=0$, $B_2=1$)

$A_3 \cdot B_3' + x_3 \cdot A_2 \cdot B_2' + x_3 \cdot x_2 \cdot A_1 \cdot B_1' + x_3 \cdot x_2 \cdot x_1 \cdot A_0 \cdot B_0'$

...

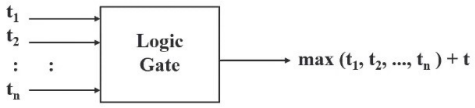
Block diagram of 4-bit magnitude comparator



# Circuit Delays

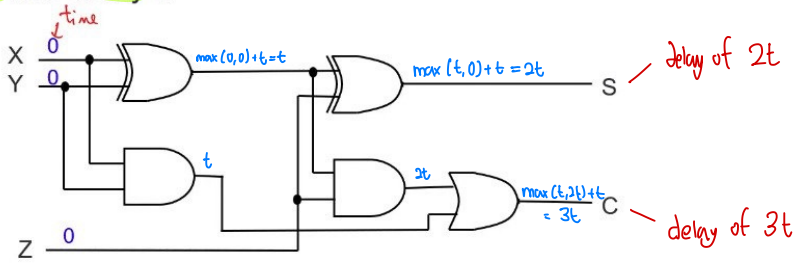
- each logic gate have a certain delay.
- given a logic gate with delay $t$, if inputs are stable at times $t_1, t_2, \ldots, t_n$, then the earliest time in which the output will be stable is:

$$\max(t_1, t_2, \ldots, t_n) + t.$$

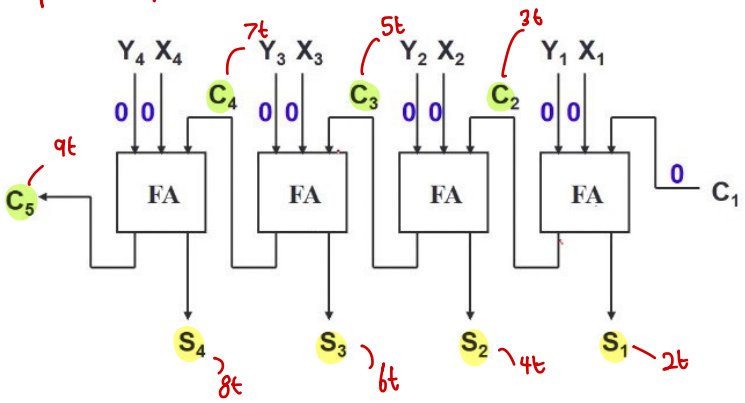



Delay of all outputs of a combinational circuit, repeat the above rule for all gates.

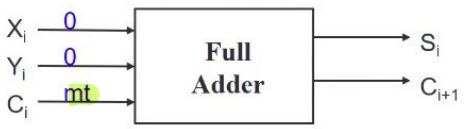
As a simple example, consider the full adder circuit where all inputs are available at time 0. Assume each gate has delay t.




More complex example:

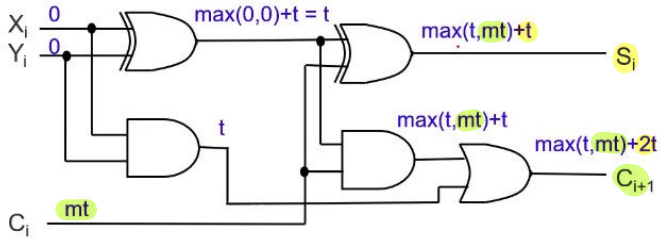



- Analyse the delay for the repeated block.




where $X_i$, $Y_i$ are stable at 0t, while $C_i$ is assumed to be stable at mt.

- Performing the delay calculation:



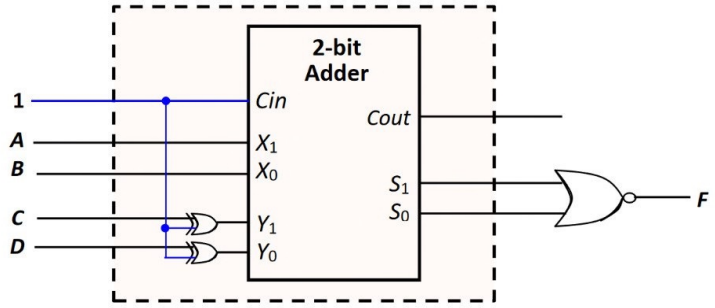

In general, an n-bit ripple-carry parallel adder will experience the following delay times:

$$S_n = (2n-1)2+2)t$$

$$C_{n+1} = (2n-1)2+3)t$$

Propagation delay of ripple-carry parallel adder is proportional to the no. of bits it handles.

The circuit is a 2-bit adder-cum-substractor. If Cin is set to 0, it performs $X + Y$. If Cin is 1, it performs $X - Y$.
Since $F$=1 when $AB = CD$, or $AB - CD = 0$, the solution is shown below. (Other answers possible).




Half-adder ≡ or