Desain Rangkaian kombinasional

Rangkaian Kombinasional Vs Sequential

- Rangkaian kombinasional adalah rangkaian yang outputnya hanya tergantung pada input ”pada saat itu”
- rangkaian sekuensial outputnya tergantung pada input saat itu dan input sebelumnya
  - dapat dikatakan rangkaian yang bekerja berdasarkan urutan waktu. Ciri rangkaian logika sekuensial yang utama adalah adanya jalur umpan balik (feedback) di dalam rangkaiannya.

Steps

1. Determine required number of inputs and outputs from the specifications.
2. Derive the truth table for each of the outputs based on their relationships to the input.
3. Simplify the boolean expression for each output. Use Karnaugh Maps or Boolean algebra.
4. Draw a logic diagram that represents the simplified Boolean expression. Verify the design by analysing or simulating the circuit.

Contoh

Bank Alarm System

A bank wants to install an alarm system with movement sensors. The bank have 3 sensors (A,B,C).

To prevent false alarms produced by a single sensor activation, the alarm will be triggered only when at least two sensors activate simultaneously.

Step 1

- Jumlah input: 3 (A, B, C), A sebagai MSB
- Jumlah output : 1 (X)
- Perilaku: X akan bernilai 1 jika minimal 2 input bernilai 1

Tabel Kebenaran

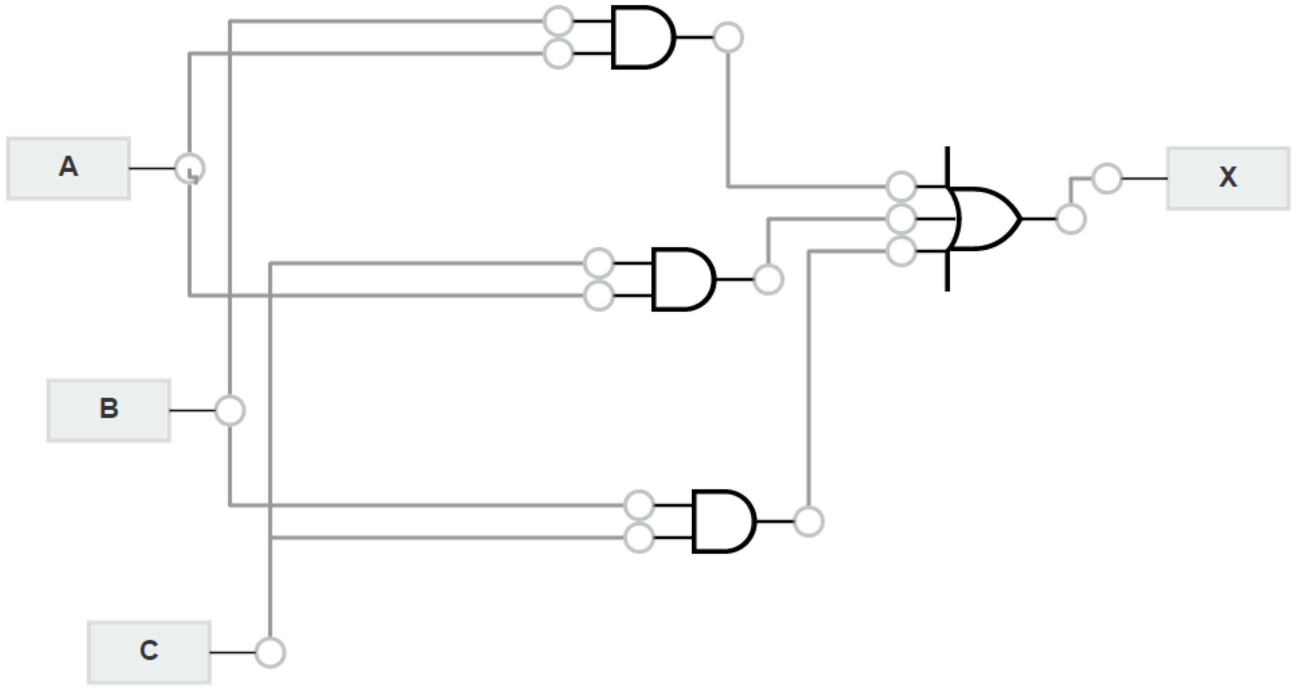
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

KMap

	-B-C	~BC	BC	B~C
-A	0	0	1	0
A	0	1	1	1

X=BC+AB+AC

Rangkaian



Tugas

Odd Numbers

Design a circuit that has a 3-bit binary input B2,B1,B0 (where B2 is MSB and B0 is LSB) and a single output (Z) specified as follows:

- Z = 0, even numbers
- Z = 1, odd numbers 1, 3, 5, 7

Car Safety Buzzer

Turn On the B(uzzer) whenever the D(oor) is Open OR when the K(ey) is in the Ignition AND the S(eat belt) is NOT Buckled.

0 : Seat Belt is NOT Buckled
1 : Seat Belt is Buckled
0 : Key is NOT in the Ignition
1 : Key is in the Ignition
0 : Door is NOT Open
1 : Door is Open
0 : Buzzer is OFF
1 : Buzzer is ON

Prime Numbers

Design a circuit that has a 3-bit binary input B2,B1,B0 (where B2 is MSB and B0 is LSB) and a single output (Z) specified as follows:

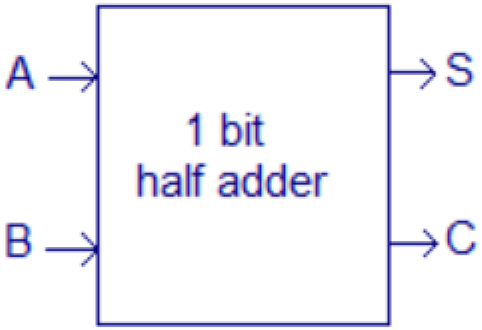
- Z = 0, non prime number
- Z = 1, prime numbers 2, 3, 5, 7

Half Adder

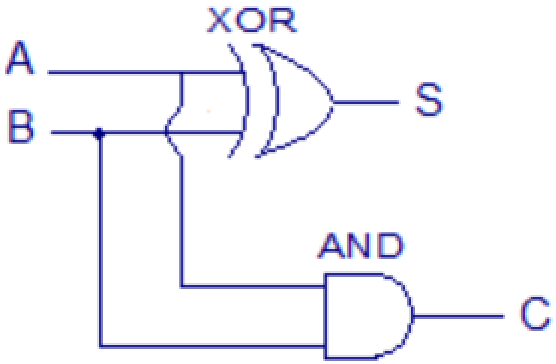
- Half adder is a combinational arithmetic circuit that adds two numbers and produces a sum bit (S) and carry bit (C) as the output
- Half adder is the simplest of all adder circuit, but it has a major disadvantage.
  - The half adder can add only two input bits (A and B) and has nothing to do with the carry if there is any in the input
  - So if the input to a half adder have a carry, then it will be neglected it and adds only the A and B bits. That means the binary addition process is not complete and that’s why it is called a half adder

Inputs		Outputs	
A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Truth table



Schematic



Realization

Truth table, schematic and realization of half adder

Full adder

- Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.

Inputs			Outputs	
A	B	C – IN	Sum	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

Logical Expression for SUM:

$= A' B' C\text{-}IN + A' B C\text{-}IN' + A B' C\text{-}IN' + A B C\text{-}IN$

$= C\text{-}IN (A' B' + A B) + C\text{-}IN' (A' B + A B')$

$= C\text{-}IN \text{ XOR } (A \text{ XOR } B)$

$= (1,2,4,7)$

Logical Expression for C-OUT:

$= A' B C\text{-}IN + A B' C\text{-}IN + A B C\text{-}IN' + A B C\text{-}IN$

$= A B + B C\text{-}IN + A C\text{-}IN$

$= (3,5,6,7)$

Another form in which C-OUT can be implemented:

$= A B + A C\text{-}IN + B C\text{-}IN (A + A')$

$= A B C\text{-}IN + A B + A C\text{-}IN + A' B C\text{-}IN$

$= A B (1 + C\text{-}IN) + A C\text{-}IN + A' B C\text{-}IN$

$= A B + A C\text{-}IN + A' B C\text{-}IN$

$= A B + A C\text{-}IN (B + B') + A' B C\text{-}IN$

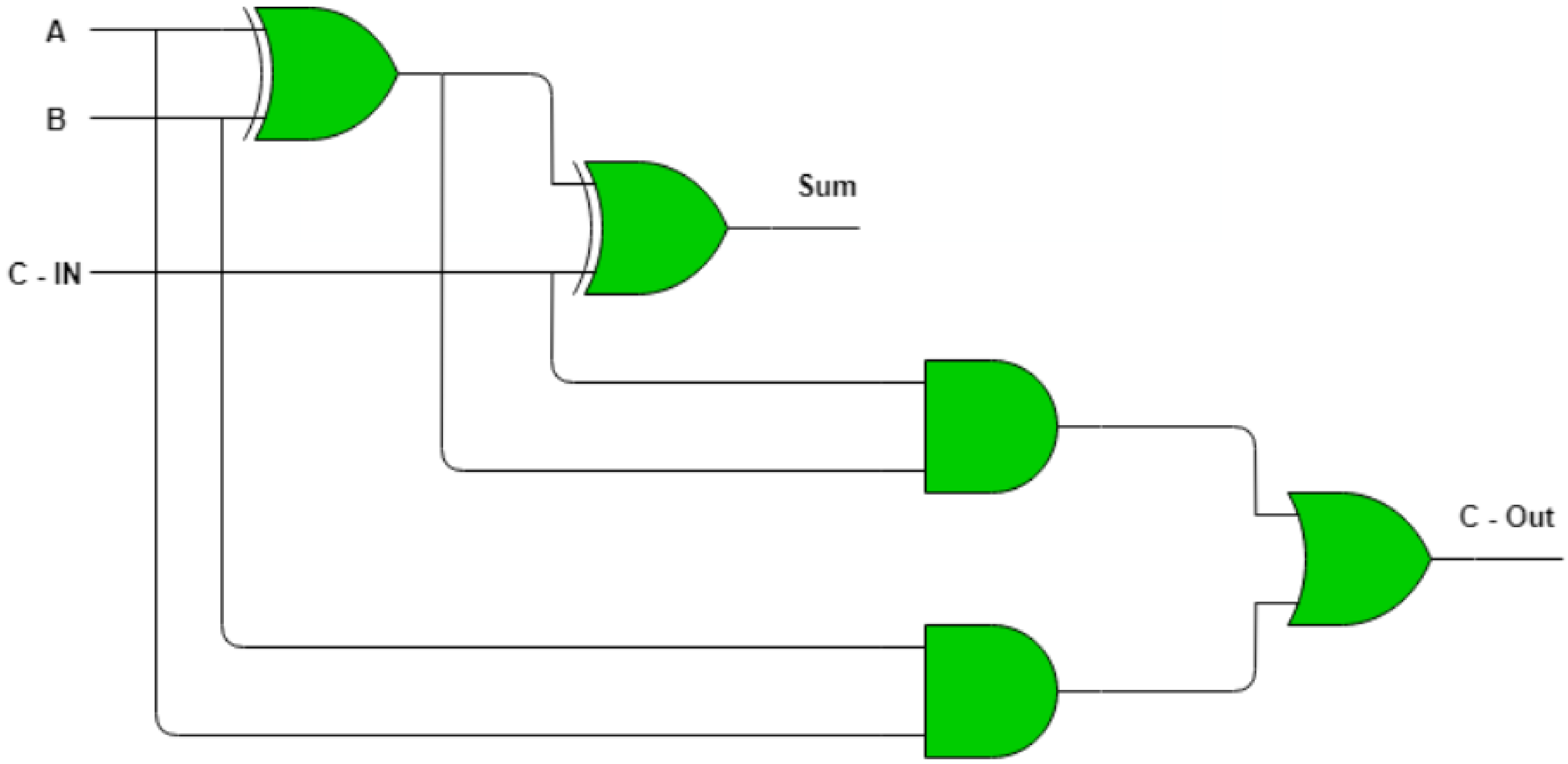
$= A B C\text{-}IN + A B + A B' C\text{-}IN + A' B C\text{-}IN$

$= A B (C\text{-}IN + 1) + A B' C\text{-}IN + A' B C\text{-}IN$

$= A B + A B' C\text{-}IN + A' B C\text{-}IN$

$= AB + C\text{-}IN (A' B + A B')$

Therefore $COUT = AB + C\text{-}IN (A \text{ EX} - \text{OR } B)$



Full Adder logic circuit.