Half Adder in Digital Logic

- A half adder is a digital logic circuit that performs binary addition of two single-bit binary numbers.
- It has two inputs, A and B, and two outputs, SUM and CARRY.
- The SUM output is the least significant bit (LSB) of the result, while the CARRY output is the most significant bit (MSB) of the result, indicating whether there was a carry-over from the addition of the two inputs

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$\overline{A}B + A\overline{B}$

Logical Expression:

For Sum:

B \ A	0	1
0	0	1
1	1	0

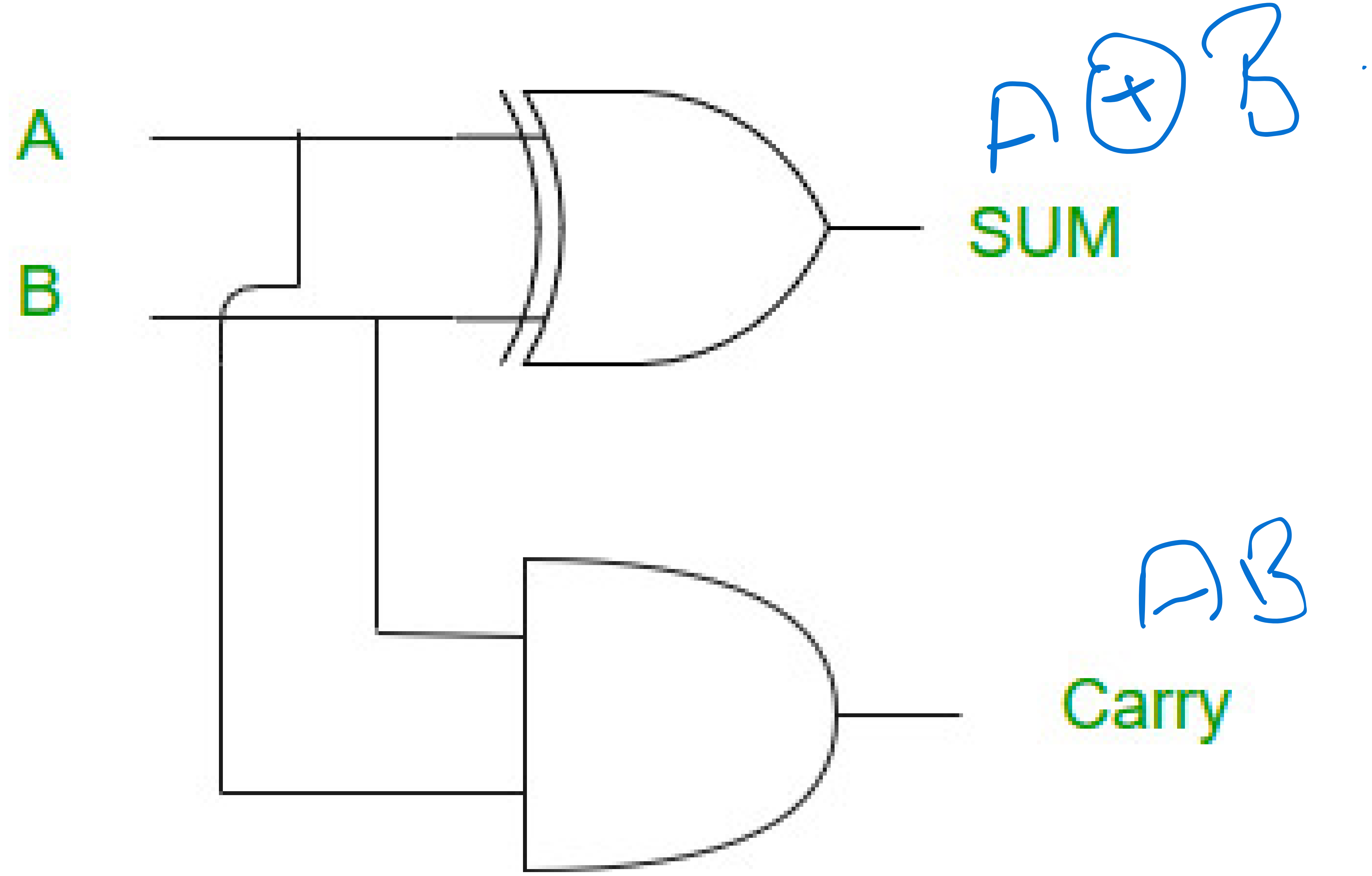
Sum = A XOR B

For Carry:

B \ A	0	1
0	0	0
1	0	$\boxed{1}$

Carry = A AND B $= AB$

Implementation:



Advantages of Half Adder in Digital Logic :

1.Simplicity: A half viper is a straightforward circuit that requires a couple of fundamental parts like XOR AND entryways. It is not difficult to carry out and can be utilized in numerous advanced frameworks.

2.Speed: The half viper works at an extremely rapid, making it reasonable for use in fast computerized circuits.

Disadvantages of Half Adder in Digital Logic :

1.Limited Usefulness: The half viper can add two single-piece numbers and produce a total and a convey bit. It can’t perform expansion of multi-bit numbers, which requires the utilization of additional intricate circuits like full adders.

2.Lack of Convey Info: The half snake doesn’t have a convey input, which restricts its value in more mind boggling expansion tasks. A convey input is important to perform expansion

Application of Half Adder in Digital Logic:

1.Arithmetic circuits: Half adders are utilized in number-crunching circuits to add double numbers. At the point when different half adders are associated in a chain, they can add multi-bit double numbers.

2.Data handling: Half adders are utilized in information handling applications like computerized signal handling, information encryption, and blunder adjustment.

3.Address unraveling: In memory tending to, half adders are utilized in address deciphering circuits to produce the location of a particular memory area.

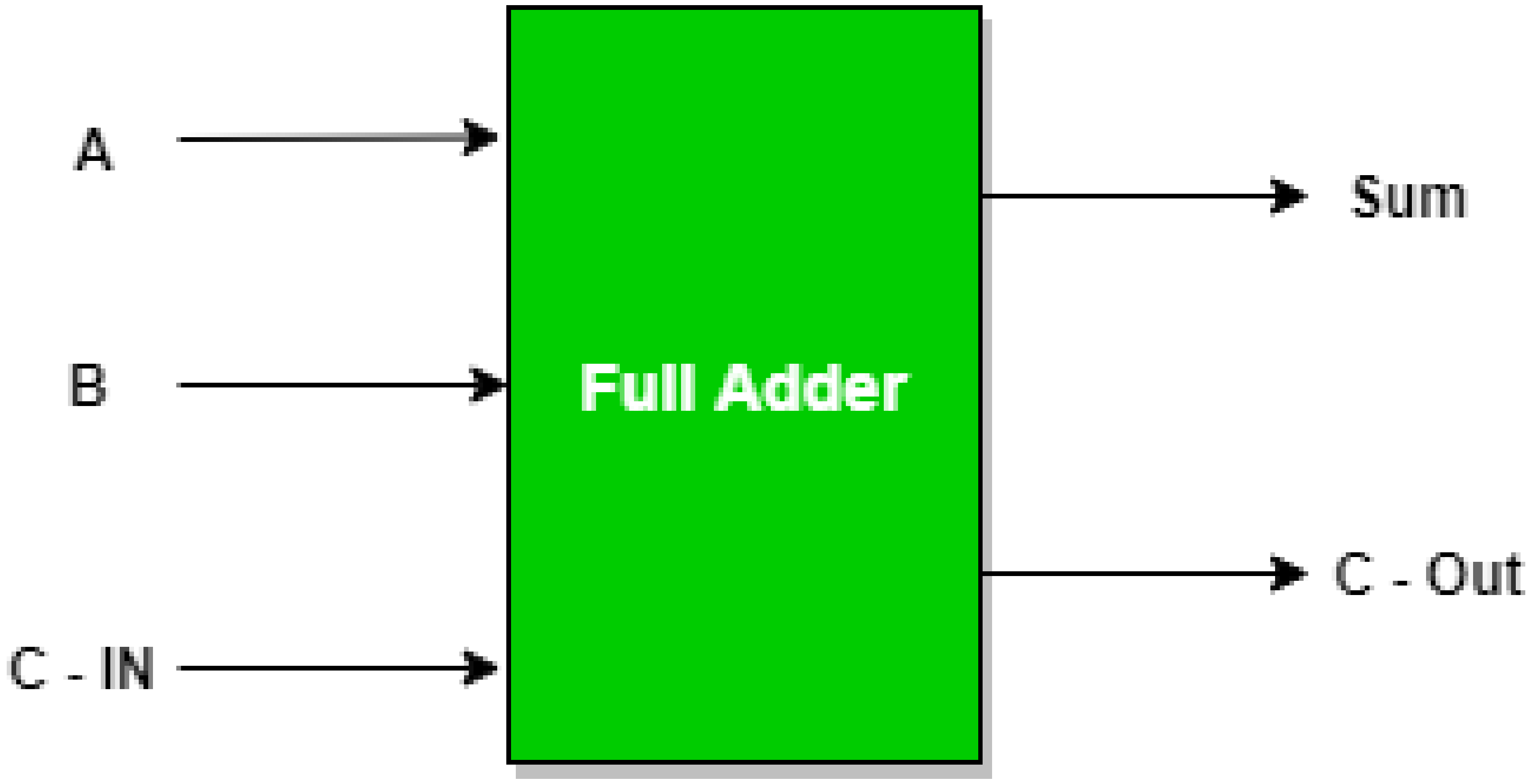
4.Encoder and decoder circuits: Half adders are utilized in encoder and decoder circuits fo computerized correspondence frameworks.

5.Multiplexers and demultiplexers: Half adders are utilized in multiplexers and demultiplexers to choose and course information.

Full Adder

Full Adder is the adder that 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.



Carry

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

For Sum (S)

A \ BC_{in}	00	01	11	10
0	0 (0)	1 (1)	0 (3)	1 (2)
1	1 (4)	0 (5)	1 (7)	0 (6)

$$S = A'B'C_{in} + A'BC'_{in} + AB'C'_{in} + ABC_{in}$$

For Output Carry (C_{out})

A \ BC_{in}	00	01	11	10
0	0 (0)	0 (1)	1 (3)	0 (2)
1	0 (4)	1 (5)	1 (7)	1 (6)

$$C_{out} = AB + AC_{in} + BC_{in}$$

Figure 2 - K Map for Full Adder

tutorialspoint
SIMPLY EASY LEARNING

Characteristic Equations of Full Adder

$$Sum, S = A \oplus B \oplus Cin = A'B'Cin + A'BC'in + AB'C'in + ABCin$$

$$Carry, C = AB + ACin + BCin$$

Applications of Full Adder

Full adders are used in ALUs (arithmetic logic units) of CPUs of computers.

Full adders are used in calculators.

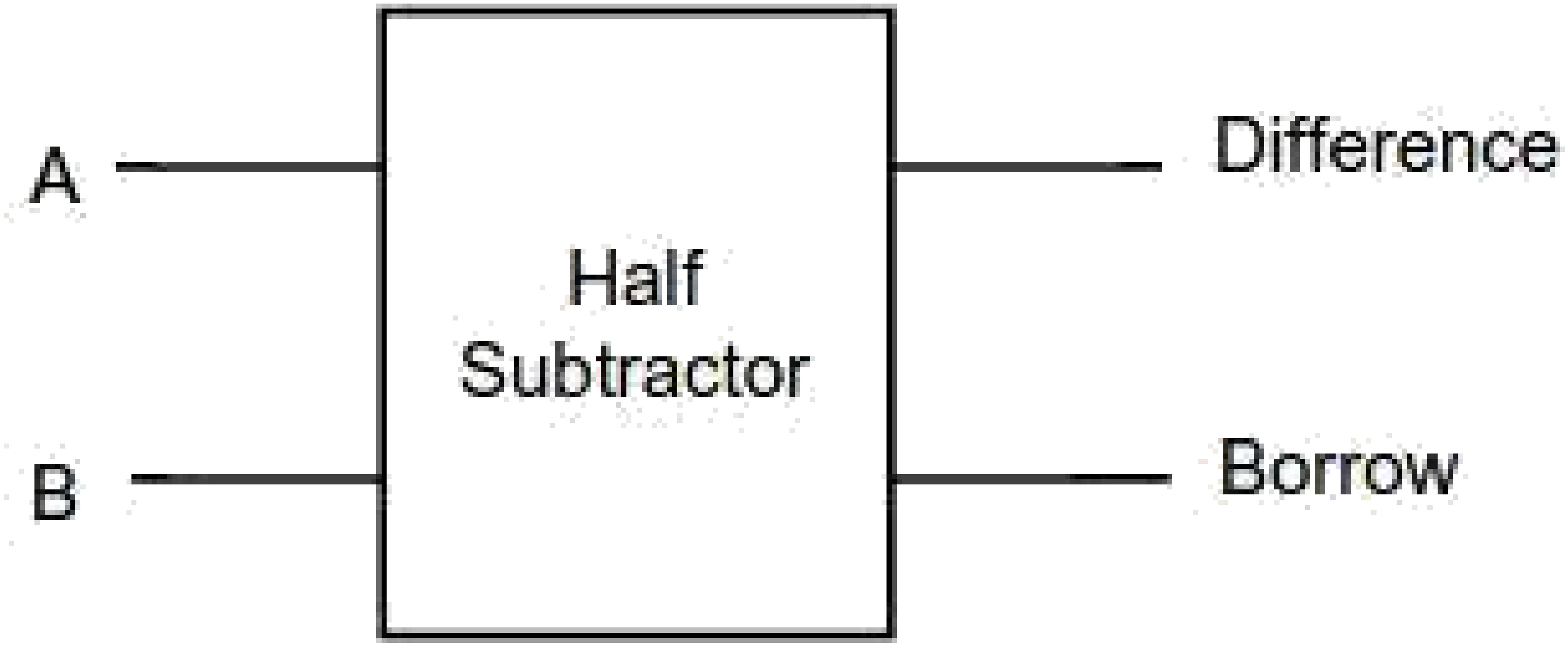
Full adders also help in carrying out multiplication of binary numbers.

Full adders are also used to realize critic digital circuits like multiplexers.

Full adders are used to generate memory addresses.

Half Subtractor

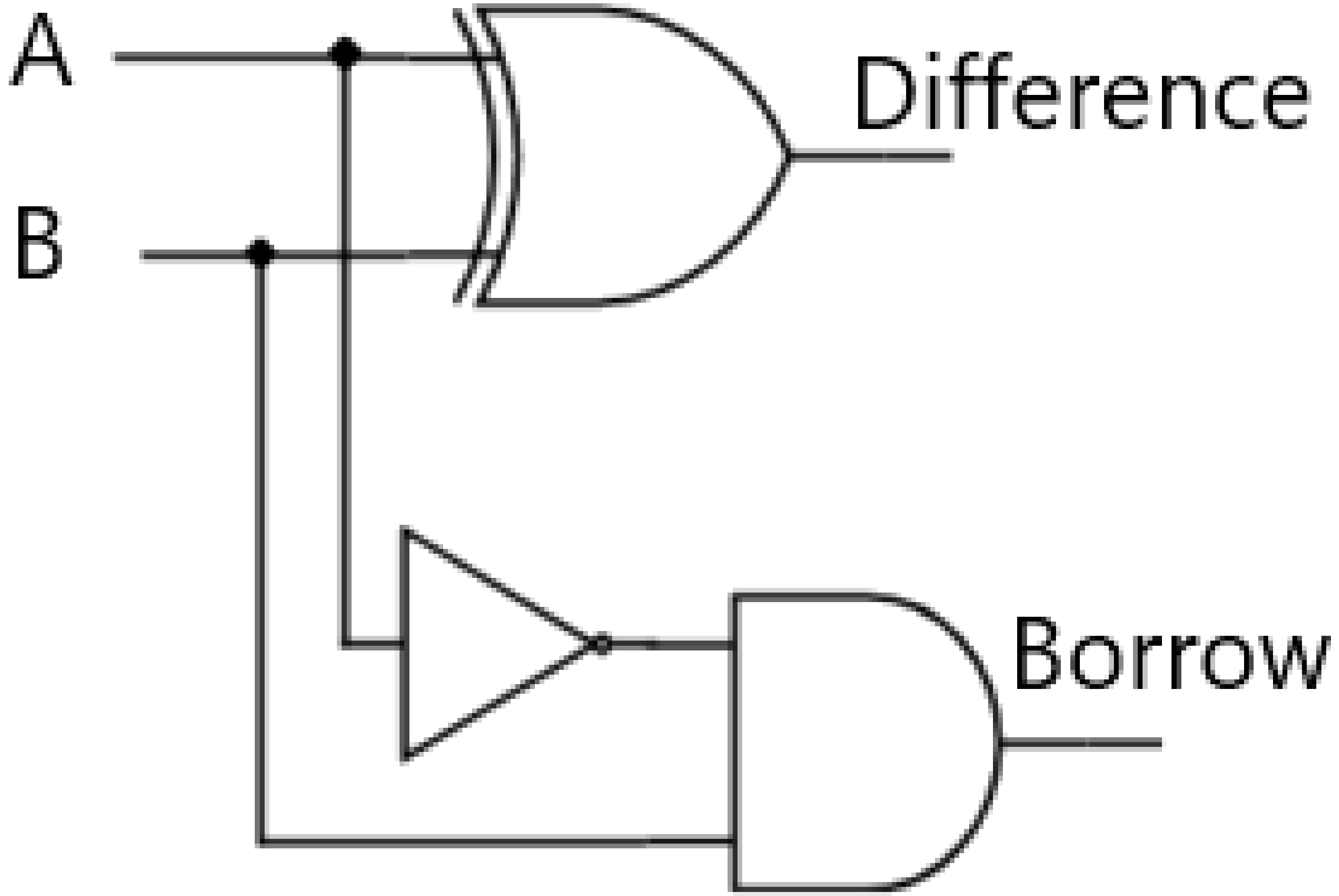
The half-subtractor is a combinational circuit which is used to perform subtraction of two bits.
It has two inputs, A (minuend) and B (subtrahend) and two outputs Difference and Borrow.



Inputs		Outputs	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$\text{Difference} = A \oplus B$$

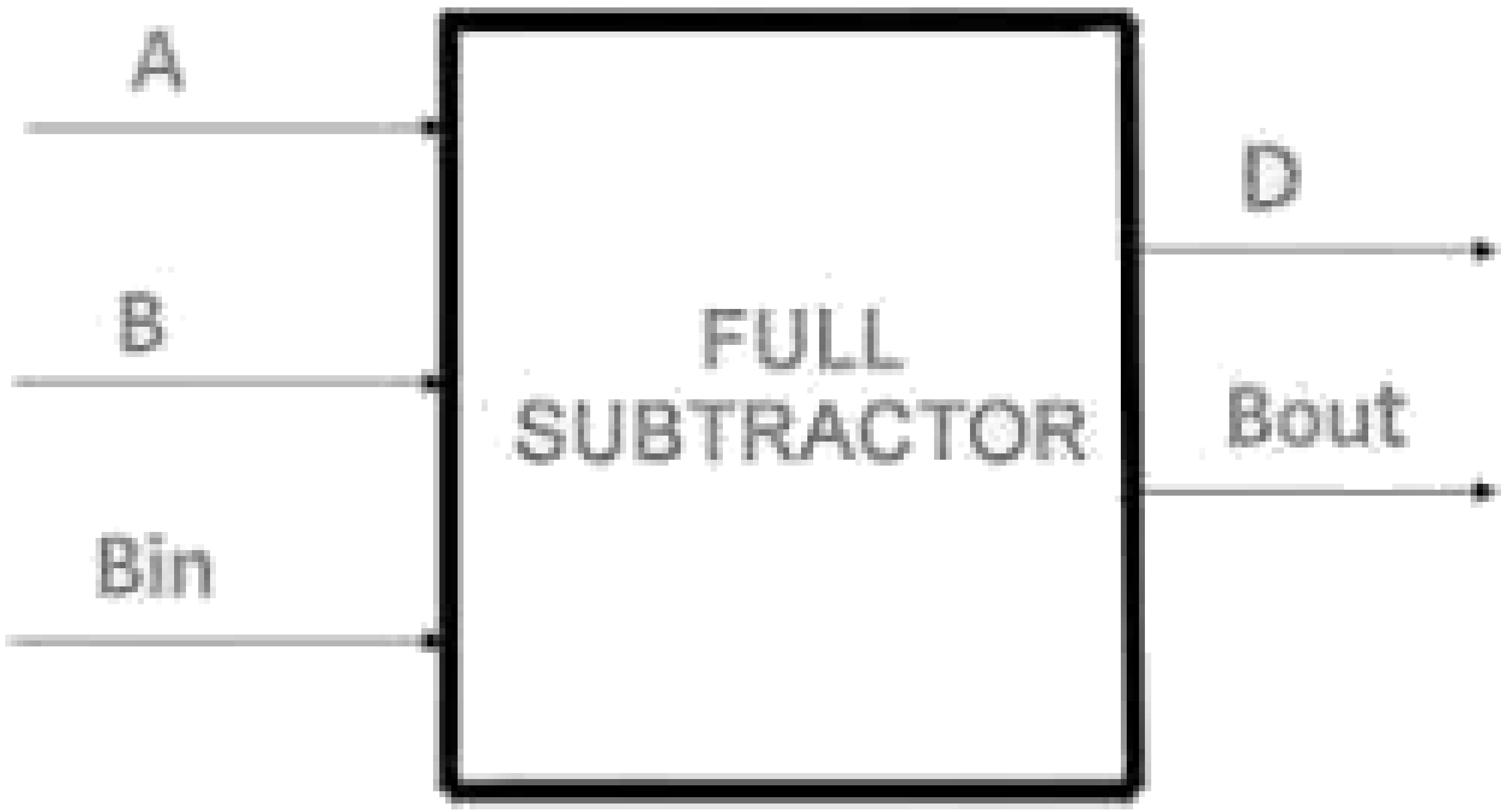
$$\text{Borrow} = A' B$$


A
B
Difference
Borrow

Full Subtractor

A full subtractor is a combinational circuit that performs subtraction **involving three bits,** namely A (minuend), B (subtrahend), and Bin (borrow-in) .

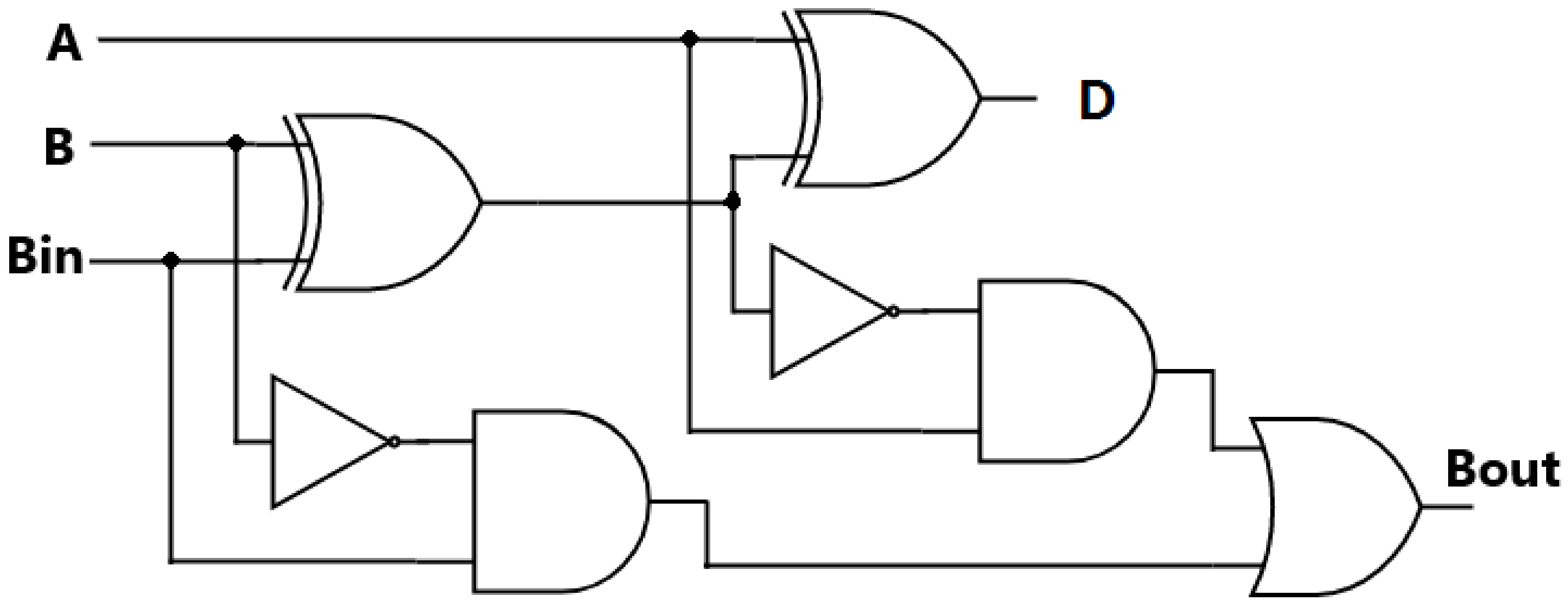
It accepts three inputs: A (minuend), B (subtrahend) and a Bin (borrow bit) and it produces two outputs: D (difference) and Bout (borrow out).


A
B
Bin
FULL
SUBTRACTOR
D
Bout

A	B	B_{in}	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

$$D = A \oplus B \oplus Bin$$

$$Bout = A'\,Bin + A'\,B + B\,Bin$$


A
B
Bin
D
Bout

Programmable Logic Device?

- integrated circuit that contains an array of logic elements and interconnections that can be programmed by the user to implement a desired logic function.
- The logic elements are usually simple combinational or sequential circuits, such as AND, OR, NOT, and XOR gates, or registers.
- The interconnections are usually programmable switches or multiplexers that can connect the inputs and outputs of the logic elements in different ways.

- The programming of a PLD can be done using a hardware description language (HDL), such as Verilog or VHDL, or a graphical user interface (GUI) software tool.
- The programming code or file is then downloaded to the PLD using a special device programmer or a standard interface, such as JTAG or USB.
- The programming code or file determines how the logic elements and interconnections are configured to perform the desired logic functio

Types of Programmable Logic Devices

- **Programmable Read-Only Memory (PROM):**
- **Programmable Logic Array (PLA):**
- **Programmable Array Logic (PAL):**
- **Generic Array Logic (GAL):** This is a device that consists of a programmable array of AND gates and a fixed array of OR gates that can implement any sum-of-products logic expression.
- The AND array is programmed by using electrically erasable programmable read-only memory (EEPROM) cells in the rows that connect the inputs to the AND gates.
- The OR array is fixed and has a predefined number of outputs. The advantage of GALs over PALs is that they can be erased and reprogrammed electrically.

Applications and Advantages of Programmable Logic Devices

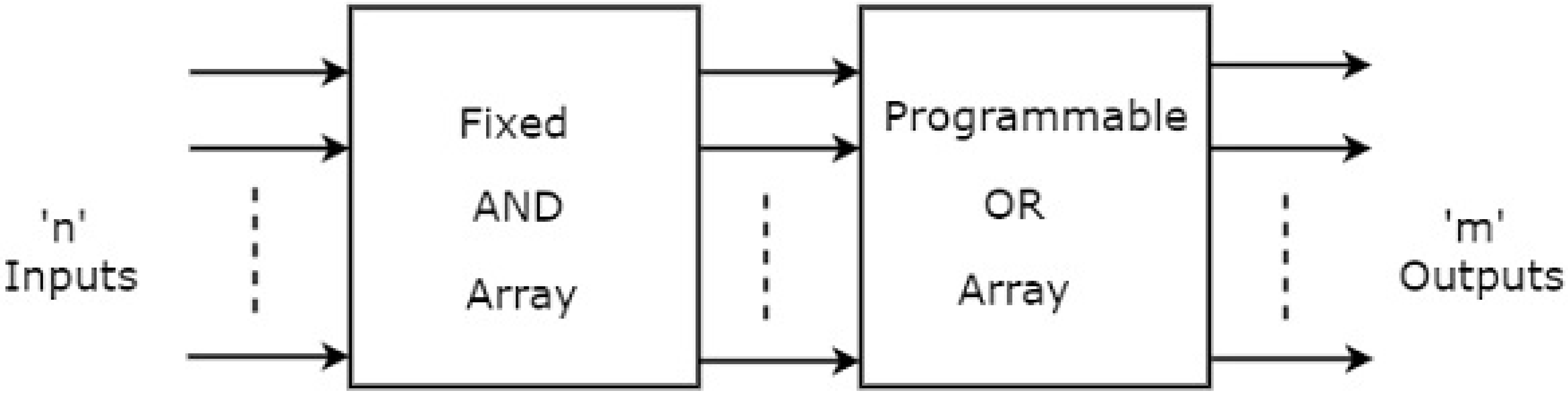
- **Digital Systems Design:** PLDs are used to design digital systems such as microprocessors, microcontrollers, digital signal processors, embedded systems, computer peripherals, communication devices, etc.
- PLDs offer flexibility in designing complex logic functions with fewer hardware components and wiring than discrete logic devices.
- **Rapid Prototyping**: PLDs are used to prototype digital systems quickly and easily without requiring expensive fabrication processes or long development cycles.
- PLDs allow the testing and debugging of digital systems on-the-fly by modifying their programming code without affecting their physical structure.

- **Performance:** PLDs offer high performance in terms of speed, power consumption, reliability, and scalability than fixed logic devices. PLDs enable parallel processing by allowing multiple operations to be performed simultaneously using different parts of the device.

- **Programmable Read Only Memory PROM**

Read Only Memory ROM is a memory device, which stores the binary information permanently. That means, we can’t change that stored information by any means later.

If the ROM has programmable feature, then it is called as Programmable ROM PROM The user has the flexibility to program the binary information electrically once by using PROM programmer.

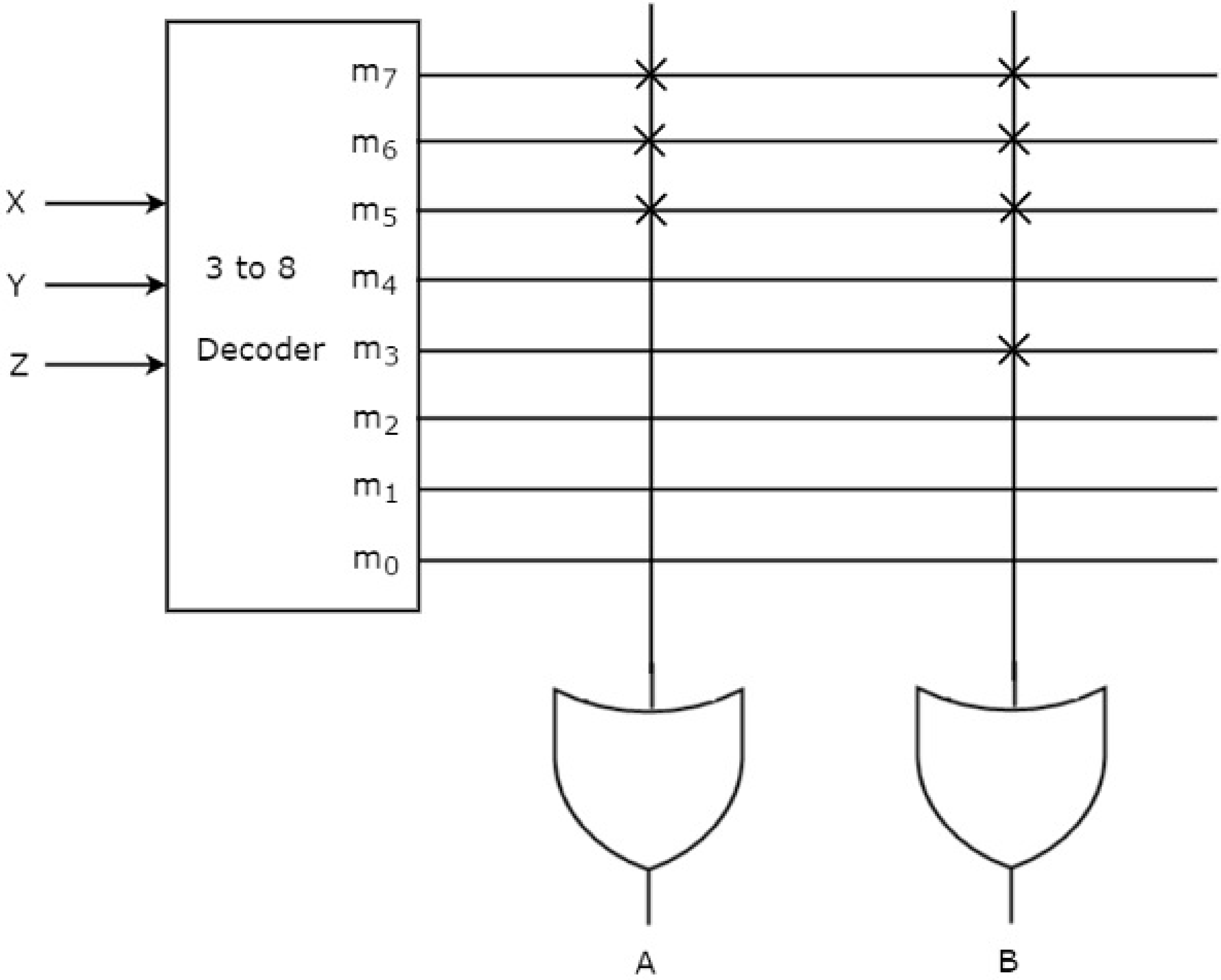


'n'
Inputs
Fixed
AND
Array
Programmable
OR
Array
'm'
Outputs

- Here, the inputs of **AND gates are not of programmable type.**
- So, we have to generate 2n product terms by using 2 ^n AND gates having n inputs each. We can implement these product terms by using nx2^n decoder. So, this decoder generates ‘n’ min terms.

- Here, the **inputs of OR gates are programmable.** That means, we can program any number of required product terms, since all the outputs of AND gates are applied as inputs to each OR gate. Therefore, the outputs of PROM will be in the form of sum of min terms.

Let us implement the following Boolean functions using PROM.



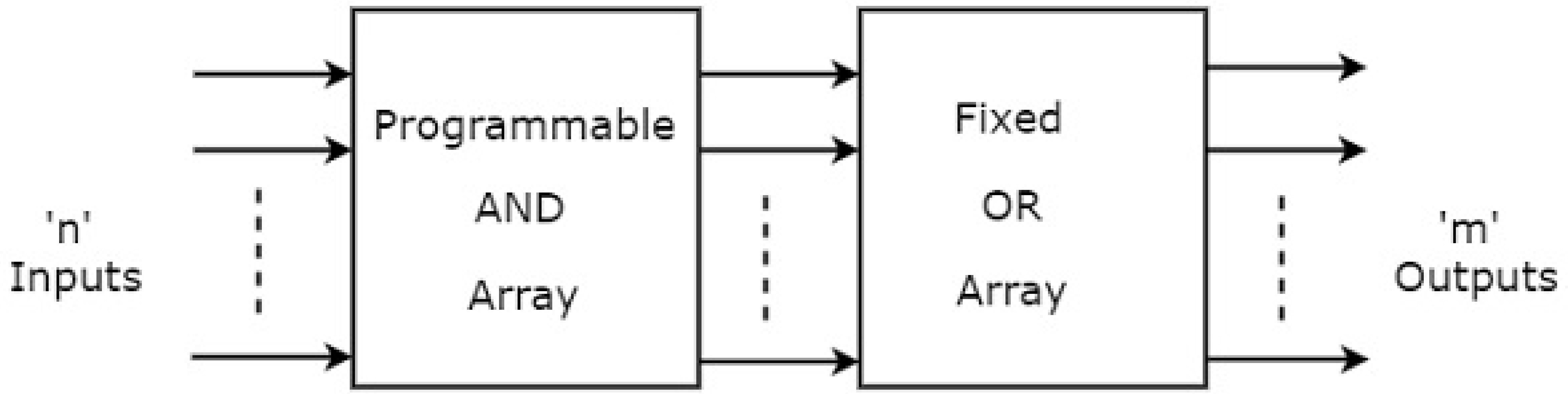
A(X,Y,Z)=∑m(5,6,7)

B(X,Y,Z)=∑m(3,5,6,7)

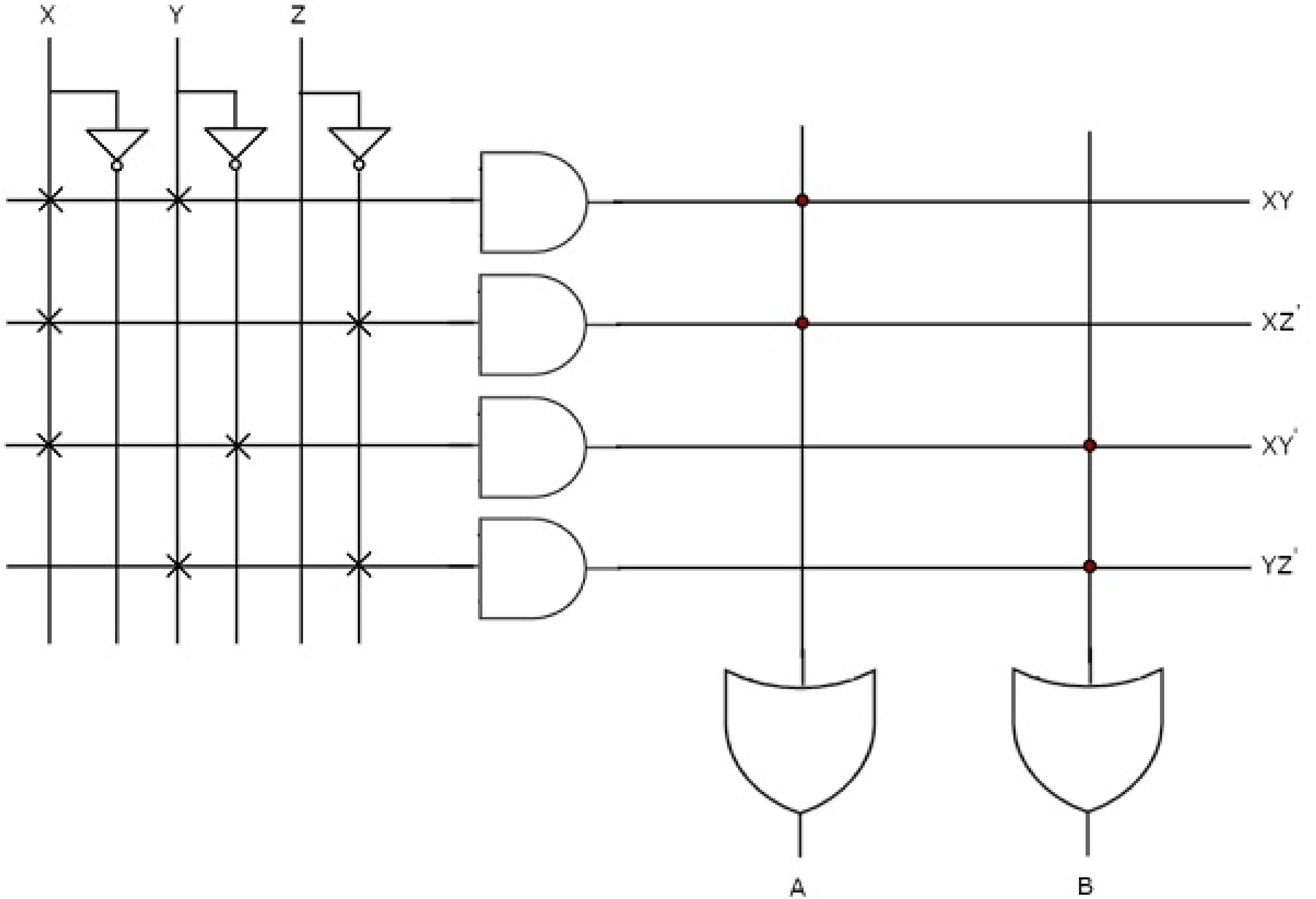
Programmable Array Logic PAL

PAL is a programmable logic device that has **Programmable AND array & fixed OR array.**

The advantage of PAL is that **we can generate only the required product terms of Boolean function** instead of generating all the min terms by using programmable AND gates.


'n'
Inputs
Programmable
AND
Array
Fixed
OR
Array
'm'
Outputs

Let us implement the following Boolean functions using PAL.



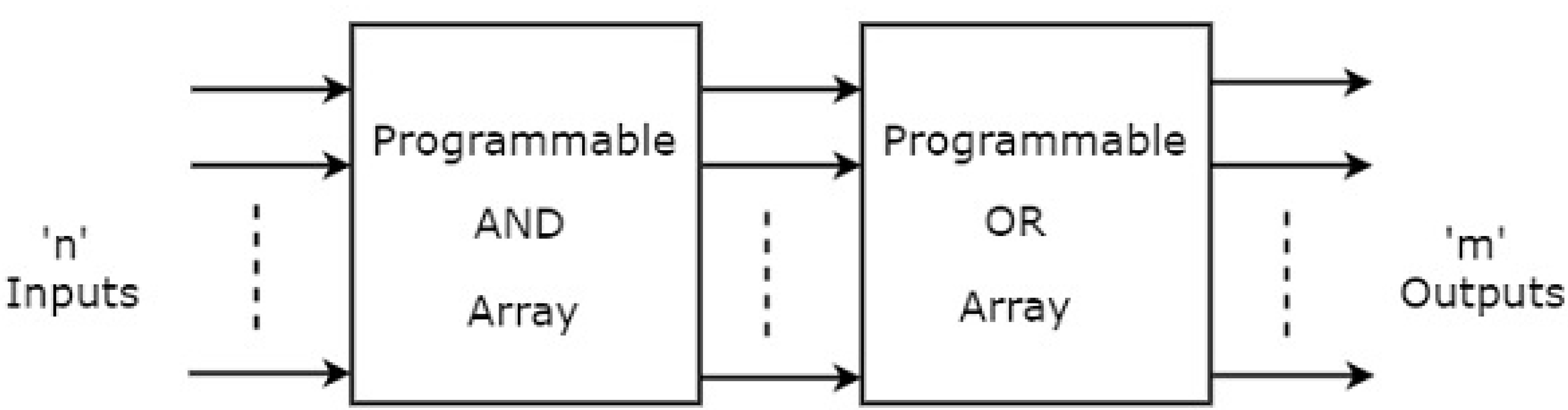
A=XY+XZ′

A=XY′+YZ′

The programmable AND gates have the access of both normal and complemented inputs of variables. In the above figure, the inputs
X, X′
, Y, Y′
, Z & Z′
, are available at the inputs of each AND gate.
So, program only the required literals in order to generate one product term by each AND gate. **The symbol ‘X’ is used for programmable connections.**
Here, the inputs of OR gates are of fixed type. So, the necessary product terms are connected to inputs of each OR gate. So that the OR gates produce the respective Boolean functions. **The symbol ‘.’ is used for fixed connections.**

Programmable Logic Array PLA

PLA is a programmable logic device that has both Programmable AND array & Programmable OR array. Hence, it is the most flexible PLD.



Let us implement the following Boolean functions using PLA.

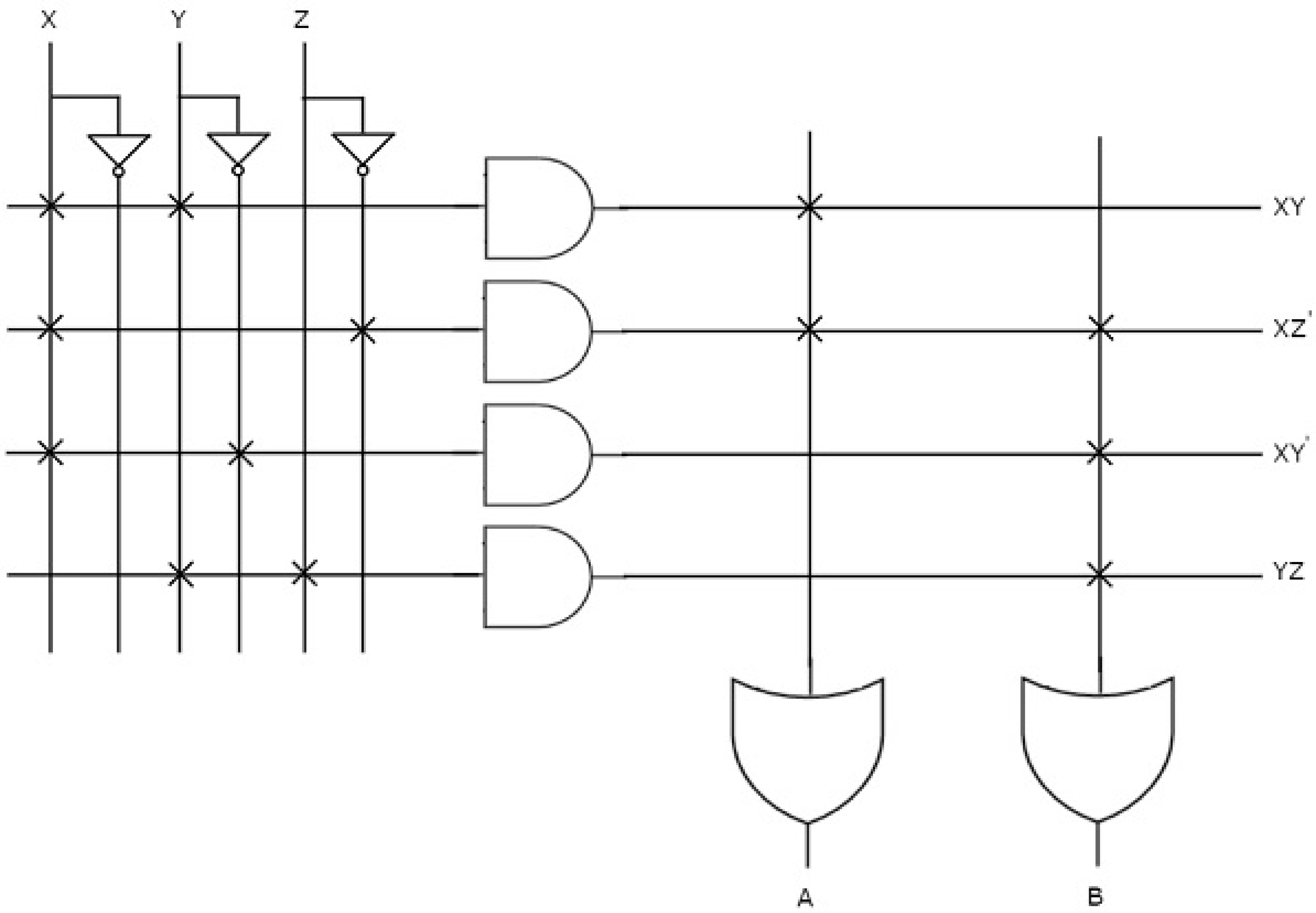
$A=XY+XZ'$

$B=XY'+YZ+XZ'$

The given two functions are in sum of products form. The number of product terms present in the given Boolean functions A & B are two and three respectively. One product term, $Z'X$

is common in each function.

So, we require four programmable AND gates & two programmable OR gates for producing those two functions


X
Y
Z
XY
XZ'
XY'
YZ
A
B