

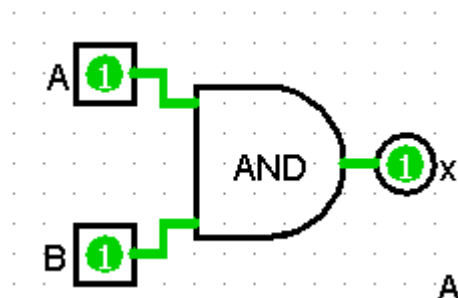
Computer Organization
LAB PRACTICALS
Harsh M. Panchal (FSMCA-I)

Practical performed under guidance of
Prof. Smita Mahajan

[1] To study and analyze logic gates using Logisim Simulator.

AND Gate

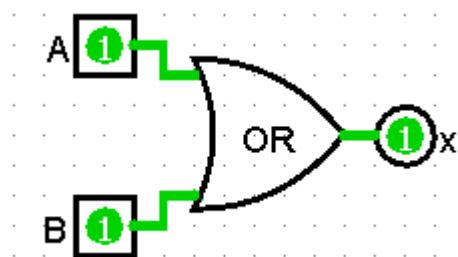
The AND gate is an electronic circuit that gives a **high** output (1) only if **all** its inputs are high. A dot(.) is used to show the AND operation. i.e. $A \cdot B$. This dot is sometimes omitted i.e. AB .



A	B	x
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

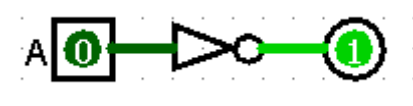
The OR gate is an electronic circuit that gives a high output (1) if **one or more** of its inputs are high. A plus (+) is used to show the OR operation. i.e. $A+B$



A	B	x
0	0	0
0	1	1
1	0	1
1	1	1

NOT or Inverter Gate

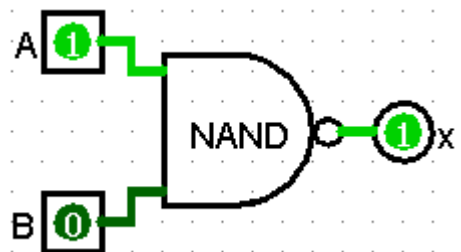
The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an inverter. If the input variable is A, the inverted output is known as NOT A. This is also shown as A' .



A	x
0	1
1	0

NAND Gate

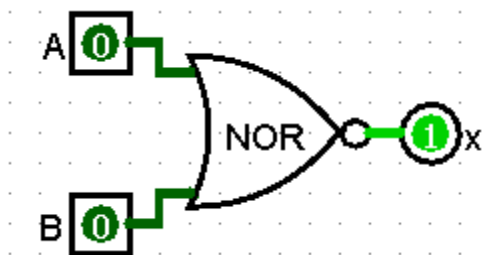
This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion. Output of NAND GATE is inverse of AND GATE.



A	B	x
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate

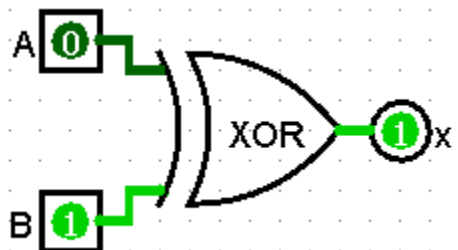
This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The small circle represents inversion. Output of NOR gate is inverse of OR gate.



A	B	x
0	0	1
0	1	0
1	0	0
1	1	0

XOR GATE

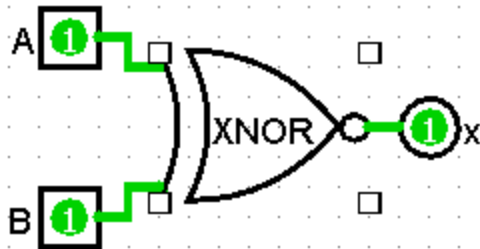
The 'Exclusive-OR' gate is a circuit which will give a high output **if either, but not both**, of its two inputs are high. An encircled plus sign (\oplus) is used to show the EOR operation.



A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

XNOR GATE

The '**Exclusive-NOR**' gate circuit does the opposite to the EOR gate. It will give a low output if **either, but not both**, of its two inputs are high. The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion.



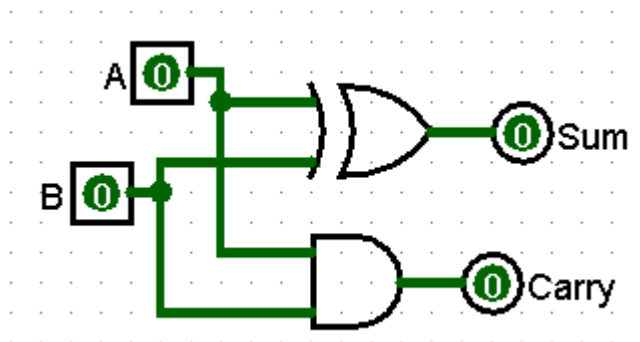
A	B	x
0	0	1
0	1	0
1	0	0
1	1	1

[2]

a. To study, design and simulate Half-ADDER Circuit.

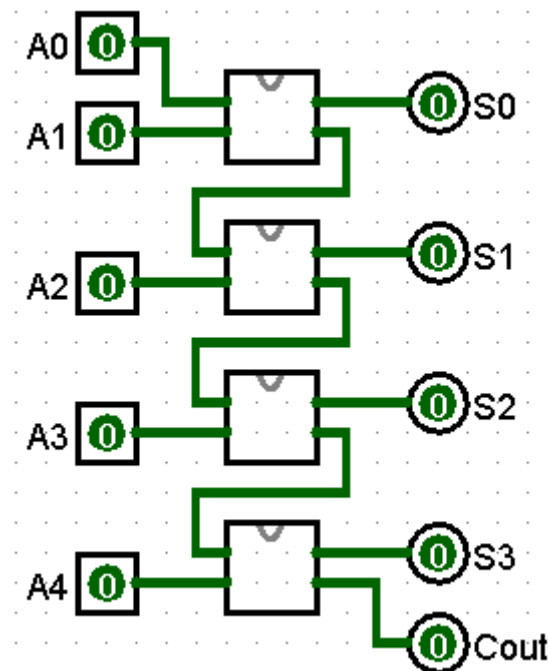
Half adder

A half adder has two inputs, generally labeled A and B, and two outputs, the sum S and carry C. S is the two-bit XOR of A and B, and C is the AND of A and B. Essentially the output of a half adder is the sum of two one-bit numbers, with C being the most significant of these two outputs.



A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

b. To design 4-bit magnitude adder using Half - ADDER circuit

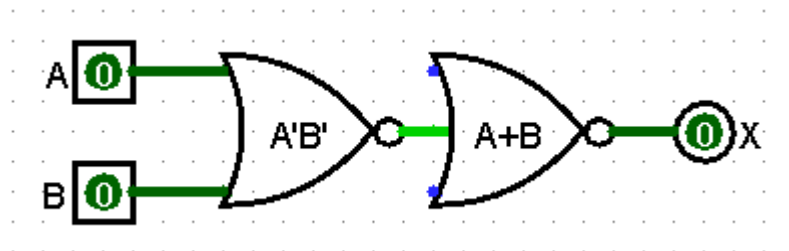


[3] To study and implementation of Universal gates

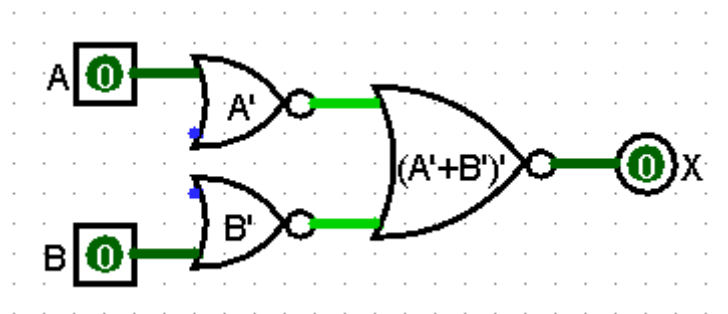
A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates. In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families.

Implementation of all gates using NOR:

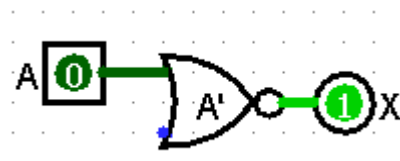
OR



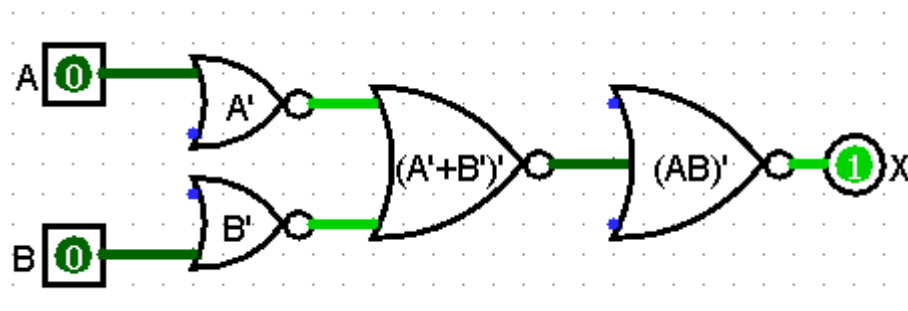
AND



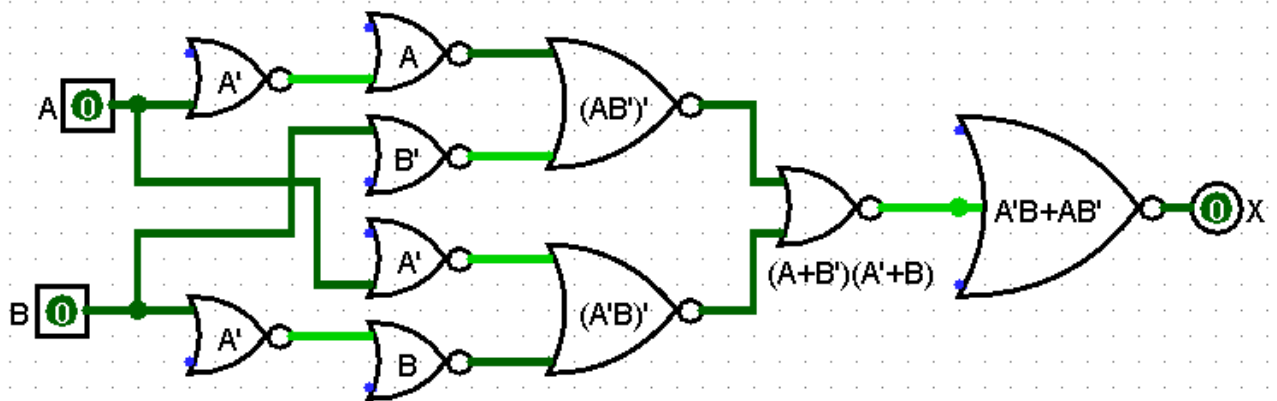
NOT



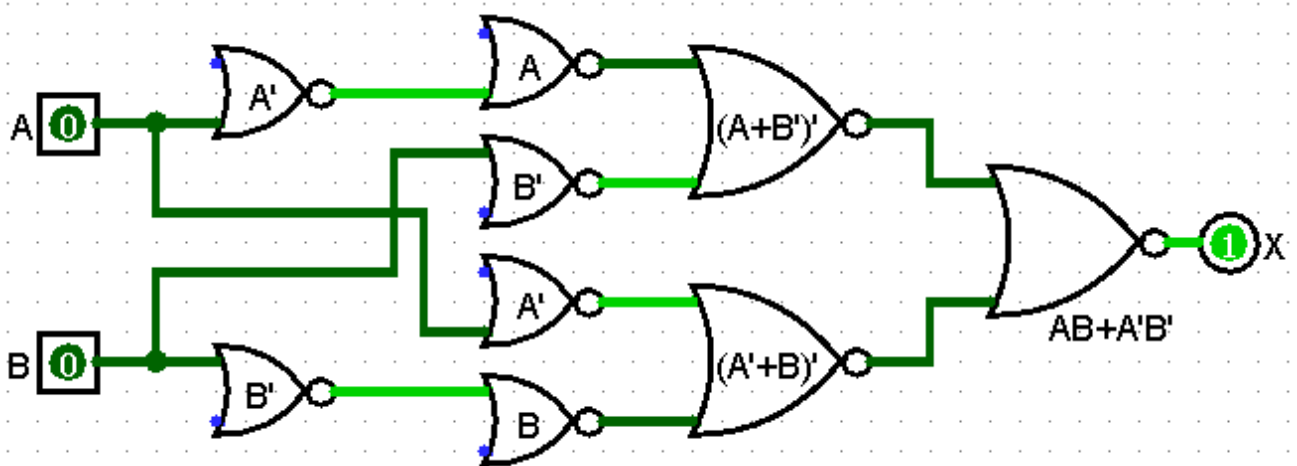
NAND



XOR

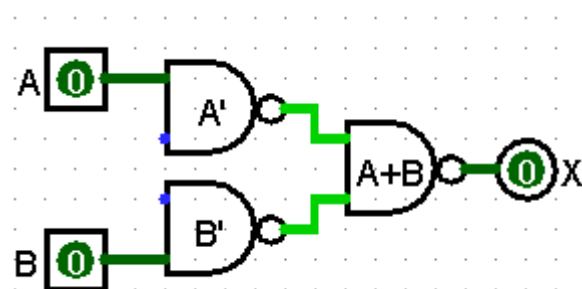


XNOR

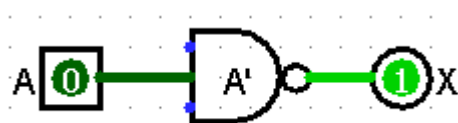


Implementation of all gates using NAND:

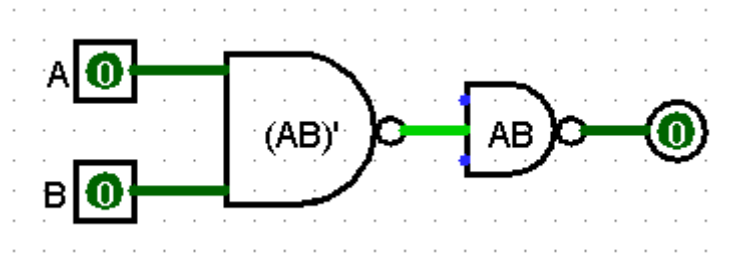
OR



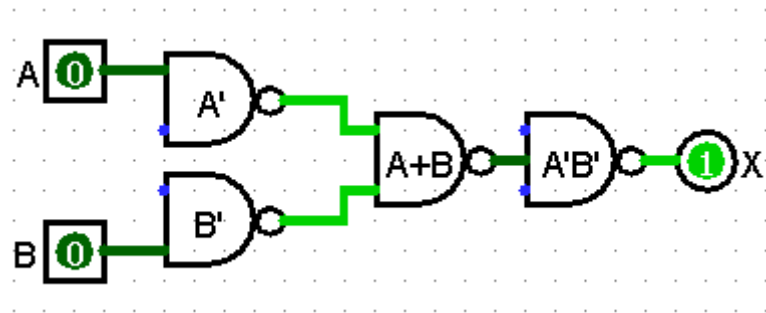
NOT



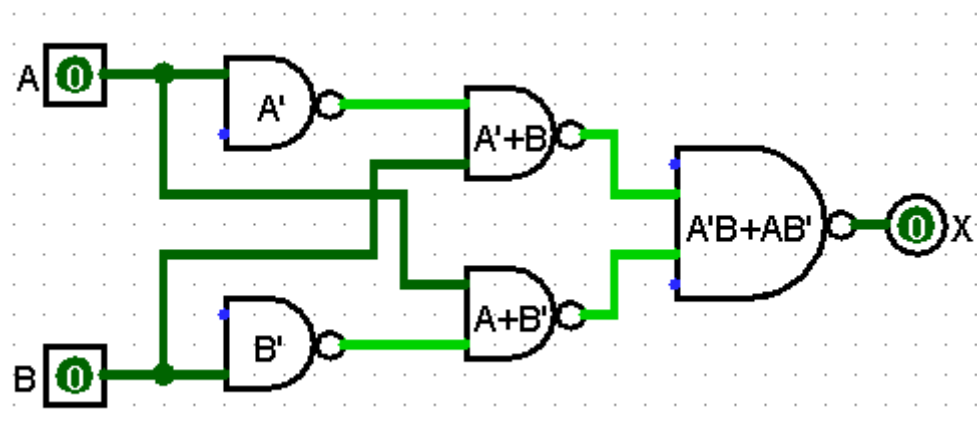
AND



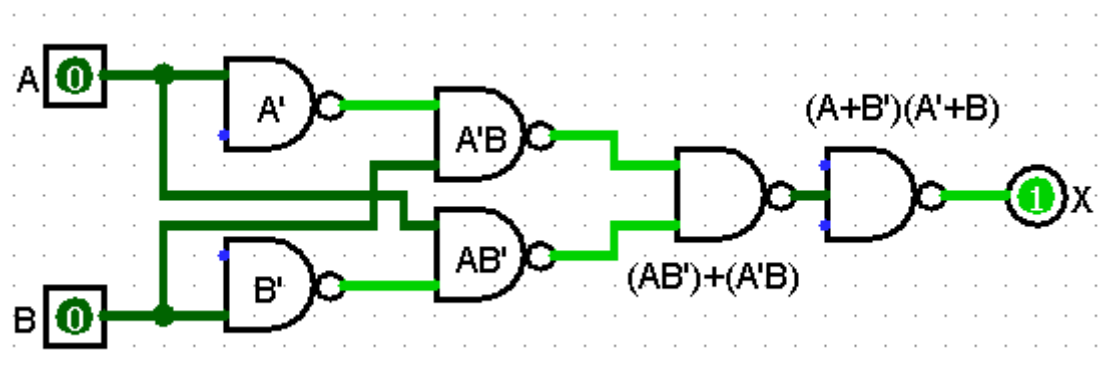
NOR



XOR

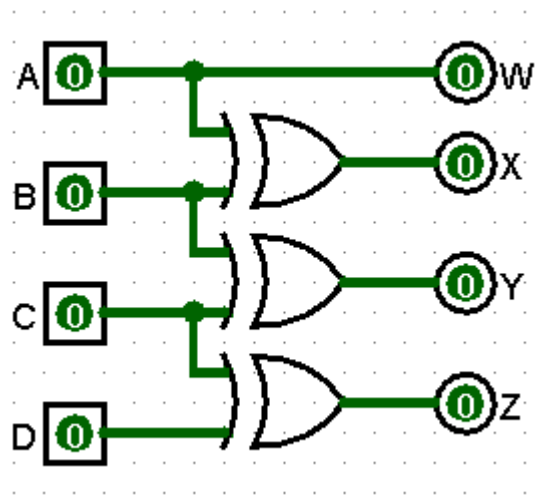


XNOR



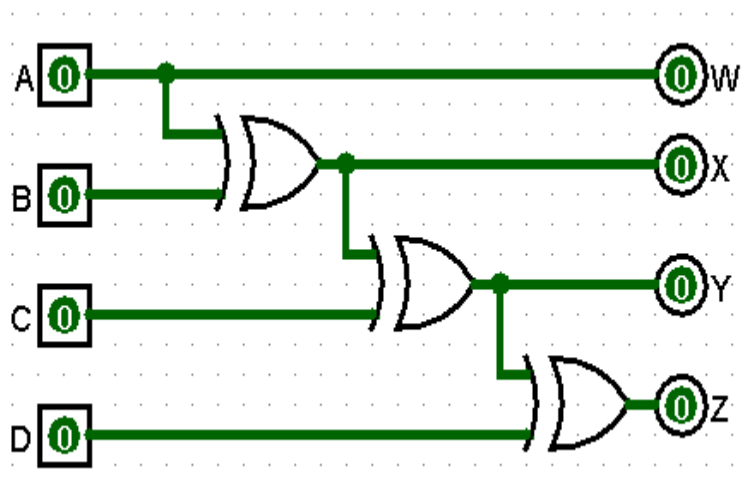
[5] Implement the following conversions

1. Convert Binary code to Gray code



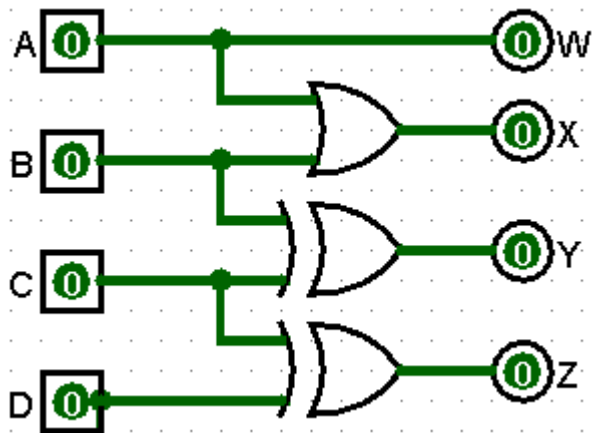
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

2. Convert Gray code to Binary code



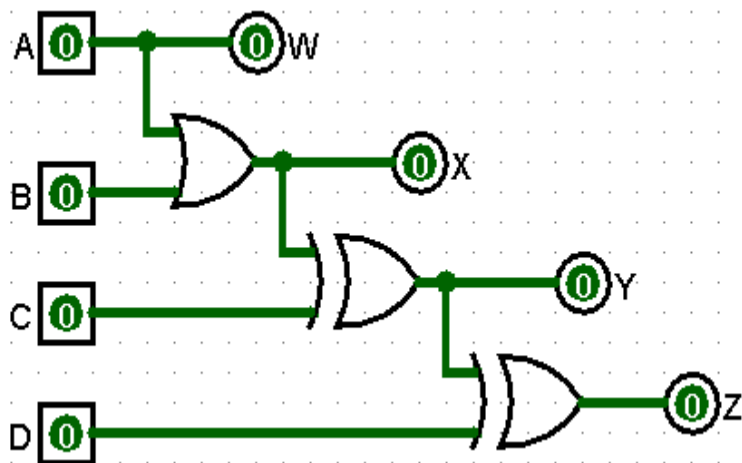
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

3. Convert BCD to Gray code



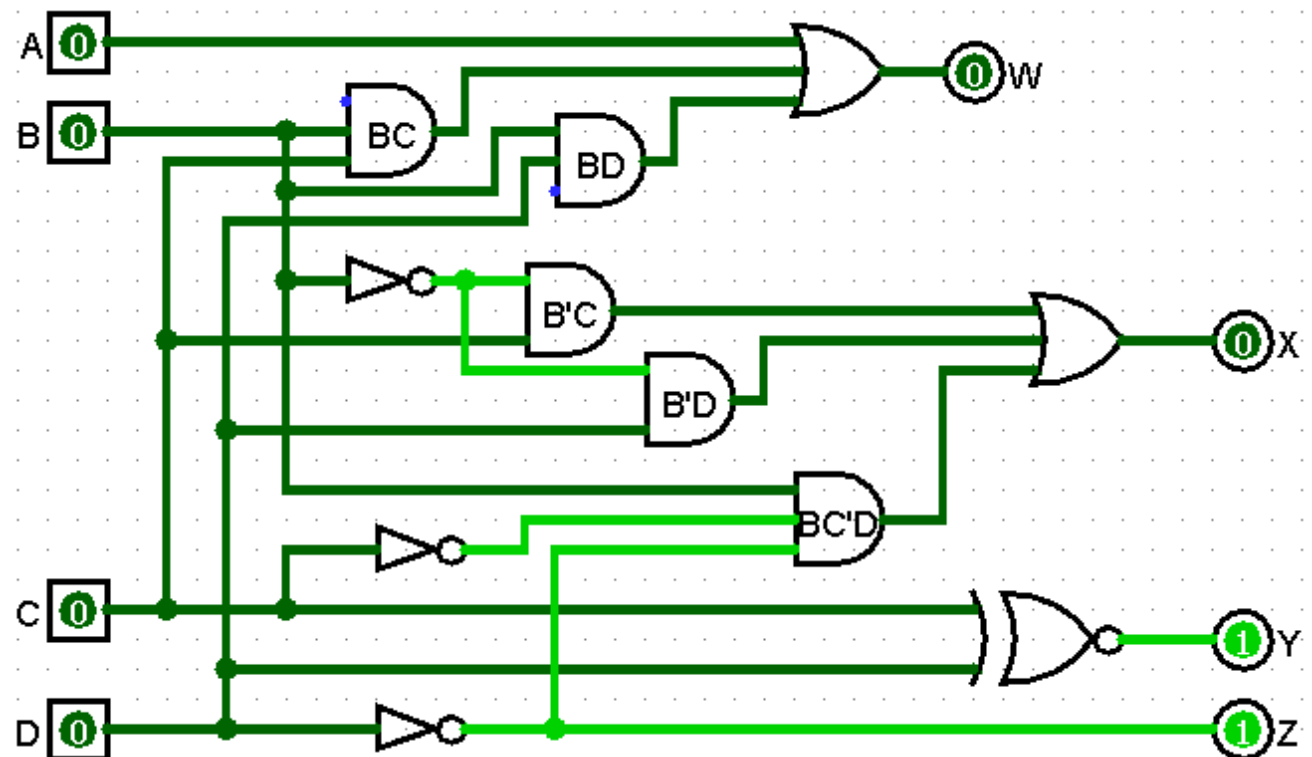
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	1	0
1	1	0	1	1	1	1	1
1	1	1	0	1	1	0	1
1	1	1	1	1	1	0	0

4. Convert Gray code to BCD



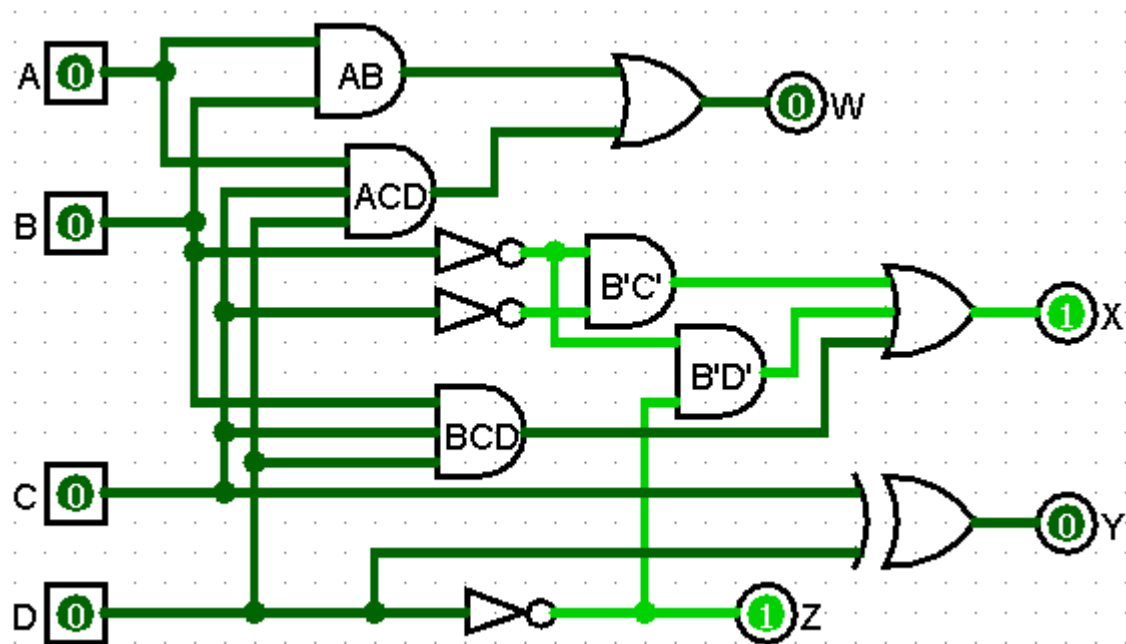
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	1	1	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	0	0
1	1	1	1	1	1	0	1

5. Convert BCD to Excess-3 code



A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	1
1	1	1	1	1	0	1	0

6. Convert Excess-3 to BCD code



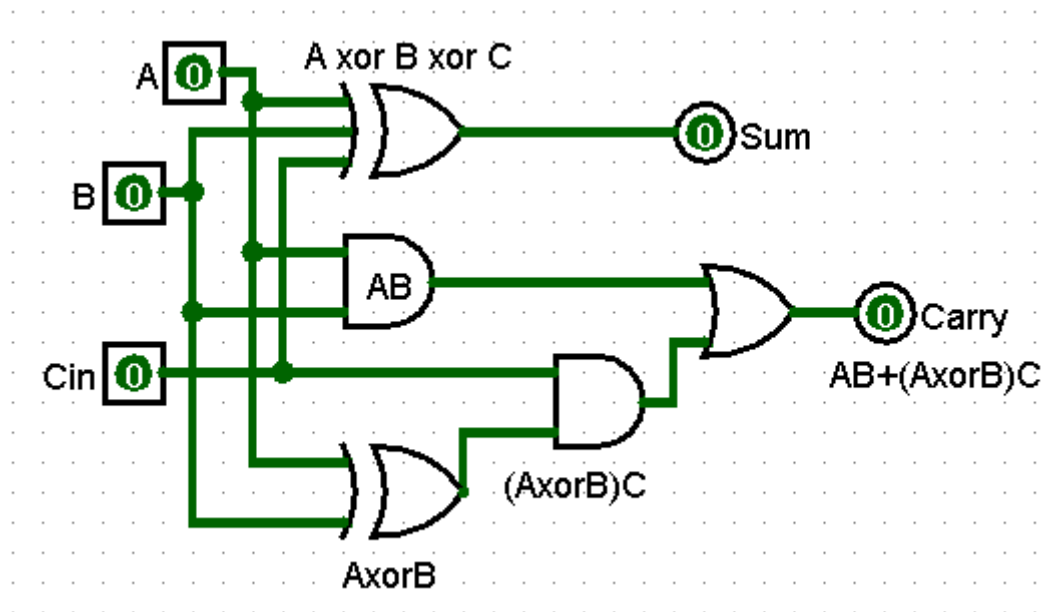
A	B	C	D	W	X	Y	Z
0	0	0	0	0	1	0	1
0	0	0	1	0	1	1	0
0	0	1	0	0	1	1	1
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1
1	1	0	1	1	0	1	0
1	1	1	0	1	0	1	1
1	1	1	1	1	1	0	0

[6]

- i. To study, design and simulate a FULL-ADDER circuit.

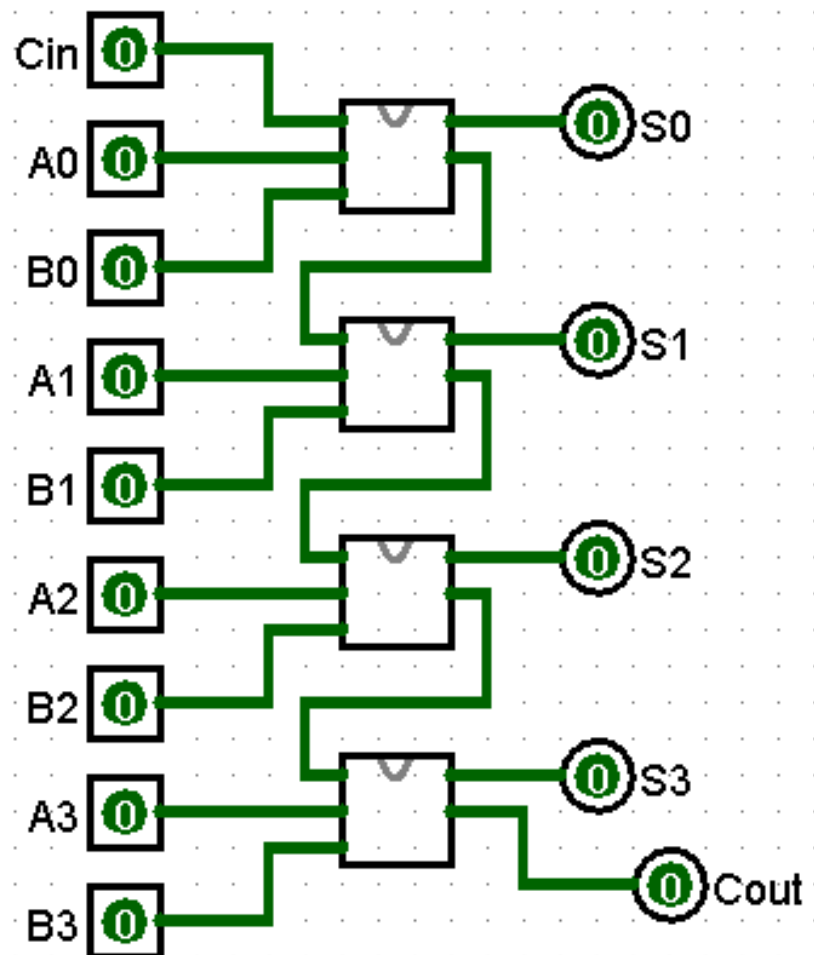
Full Adder

A full adder is a circuit that has two AND gates, two EX-OR gates, and one OR gate. The full adder adds three binary digits. Among all the three, one is the carry that we obtain from the previous addition as C-IN, and the two are inputs A and B.



A	B	Cin	Sum	Carry
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	0	1

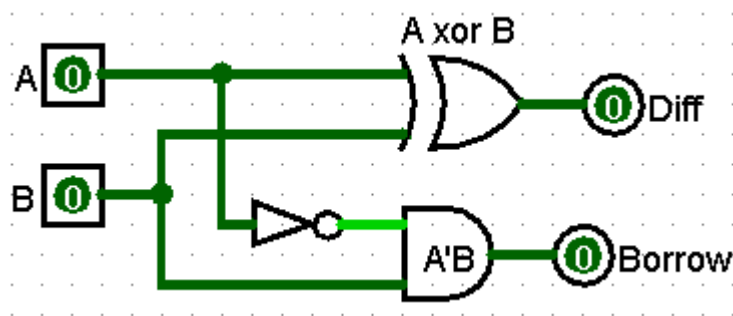
ii. To design 4-bit magnitude adder using FULL - ADDER circuit.



[7] To study, design and simulate Half Subtractor and Full Subtractor.

- **Half Subtractor**

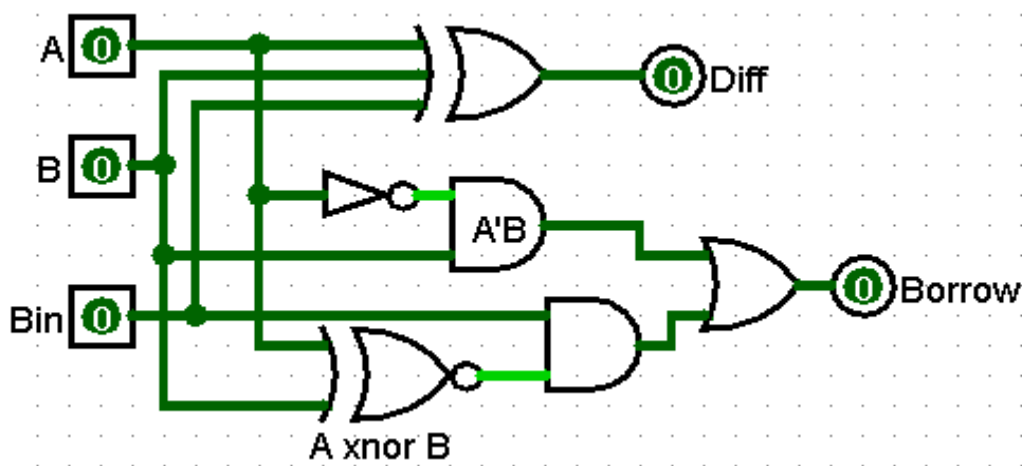
A half-subtractor is a combinational logic circuit that have two inputs and two outputs (i.e. difference and borrow). The half subtractor produces the difference between the two binary bits at the input and also produces a borrow output (if any).



A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

- **Full Subtractor**

A full-subtractor is a combinational circuit that has three inputs A, B, bin and two outputs d and b. Where, A is the minuend, B is subtrahend, bin is borrow produced by the previous stage, d is the difference output and b is the borrow output.



A	B	Bin	Diff	Borrow
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	0	1

[8] Perform addition of two 8 bit numbers using 8085

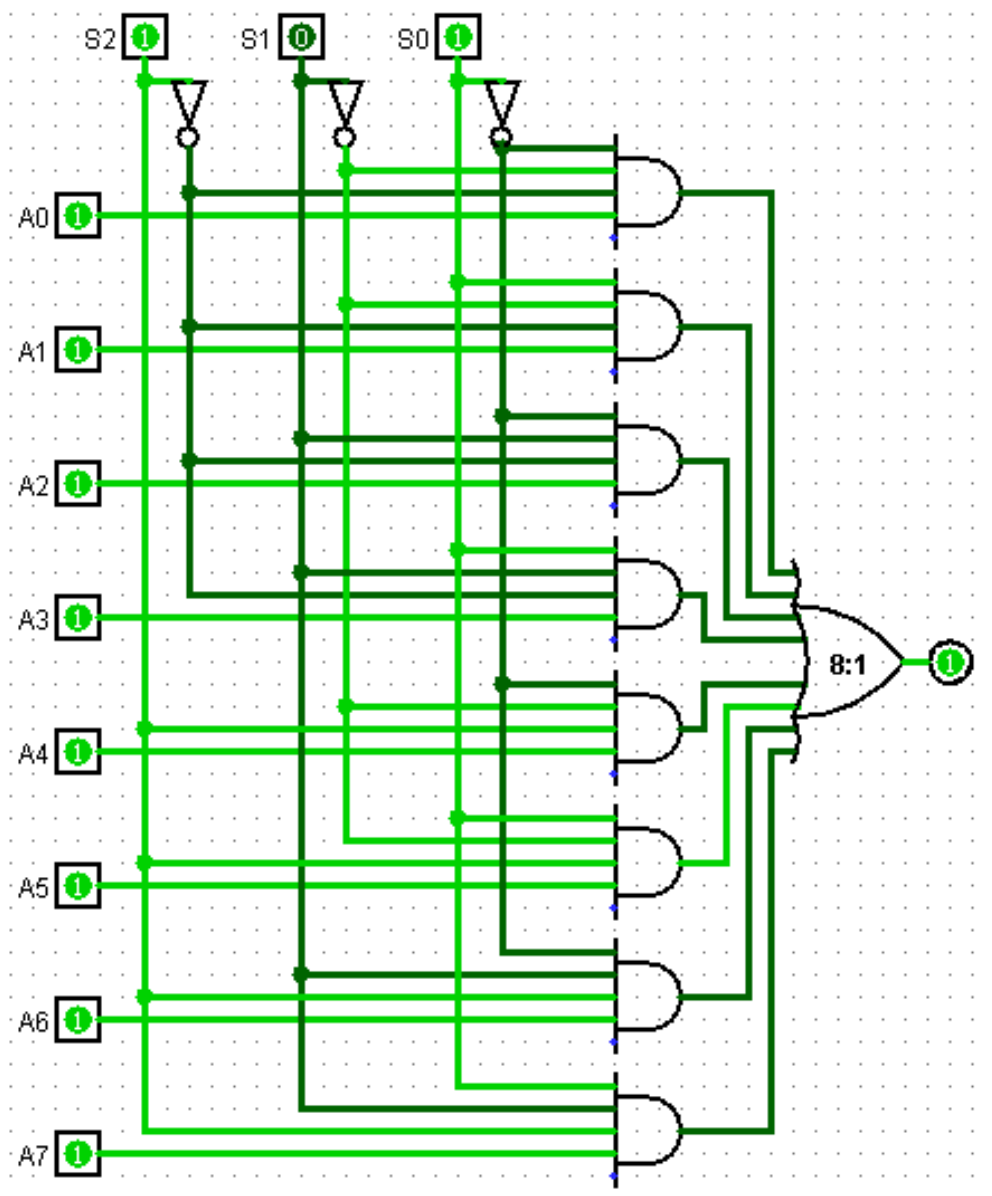
Assembler Disassembler

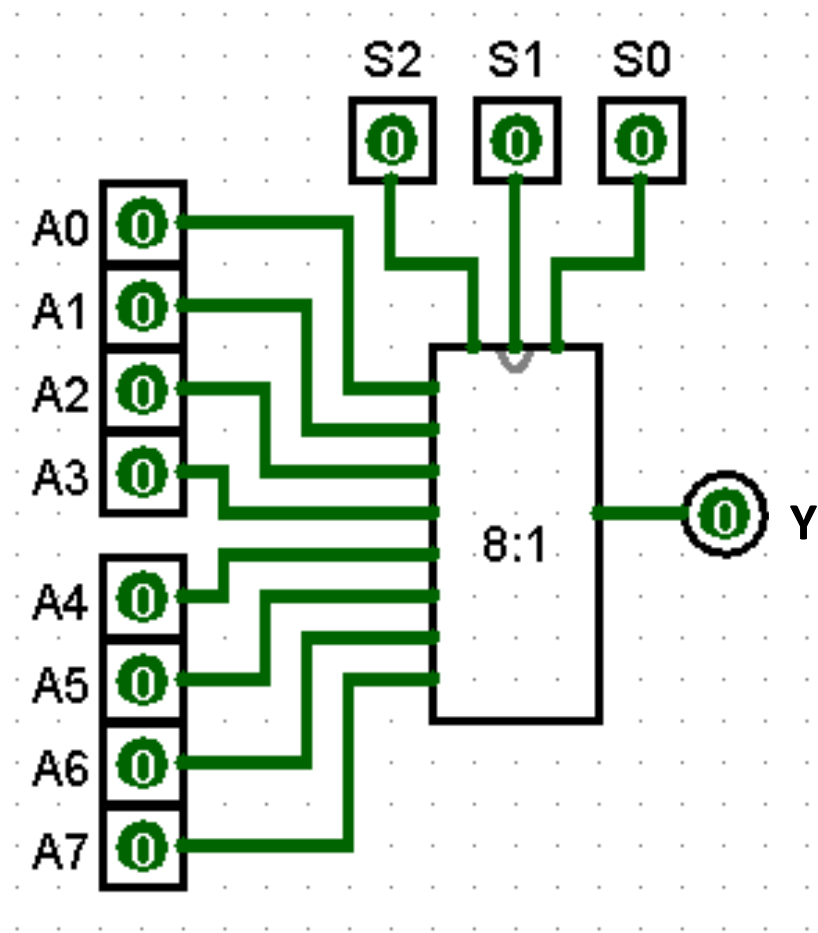
```

MVI B,3H
MVI A,5H
ADD B
          
```

Register	Value	7	6	5	4	3	2	1	0
Accumulator	08	0	0	0	0	1	0	0	0
Register B	03	0	0	0	0	0	0	1	1
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	06	0	0	0	0	0	1	1	0

[9] Study and Design 8 X 1 Multiplexer.





(8:1 MUX)

INPUTS			Output
S ₂	S ₁	S ₀	Y
0	0	0	A ₀
0	0	1	A ₁
0	1	0	A ₂
0	1	1	A ₃
1	0	0	A ₄
1	0	1	A ₅
1	1	0	A ₆
1	1	1	A ₇