Digital Logic

CHAPTER 04

Lecture 18

Combinational Logic

Overview of previous lecture

- What is a Binary addition.
- What is Binary Adder

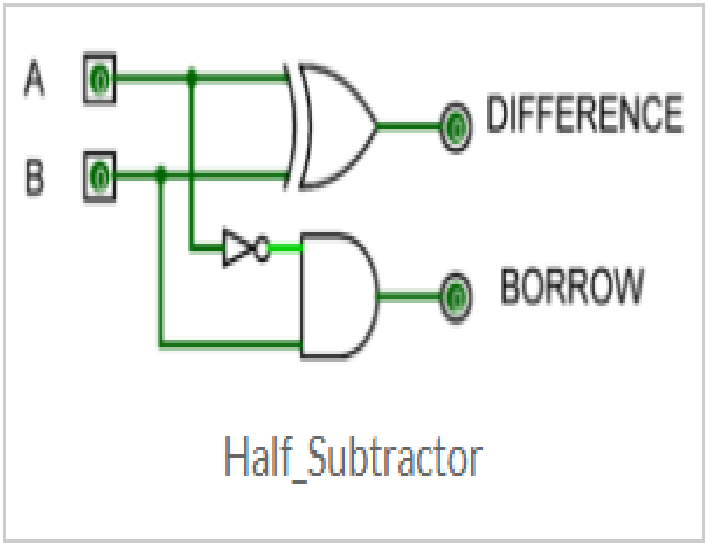
Half Subtractor

Half subtractor is a combination circuit with two inputs and two outputs (difference and borrow). It produces the difference between the two binary bits at the input and also produces an output (Borrow) to indicate if a 1 has been borrowed. In the subtraction (A-B), A is called as Minuend bit and B is called as Subtrahend bit.

Truth Table

Inputs		Output	
A	B	(A – B)	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

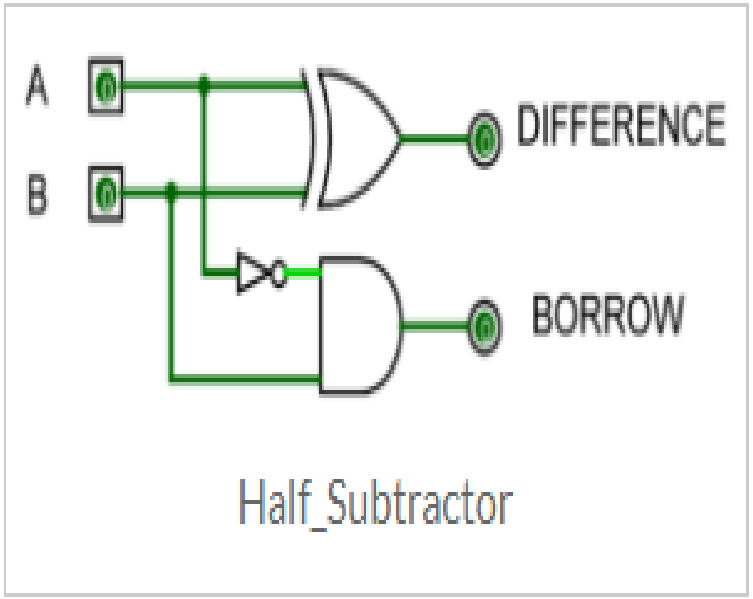
Circuit Diagram

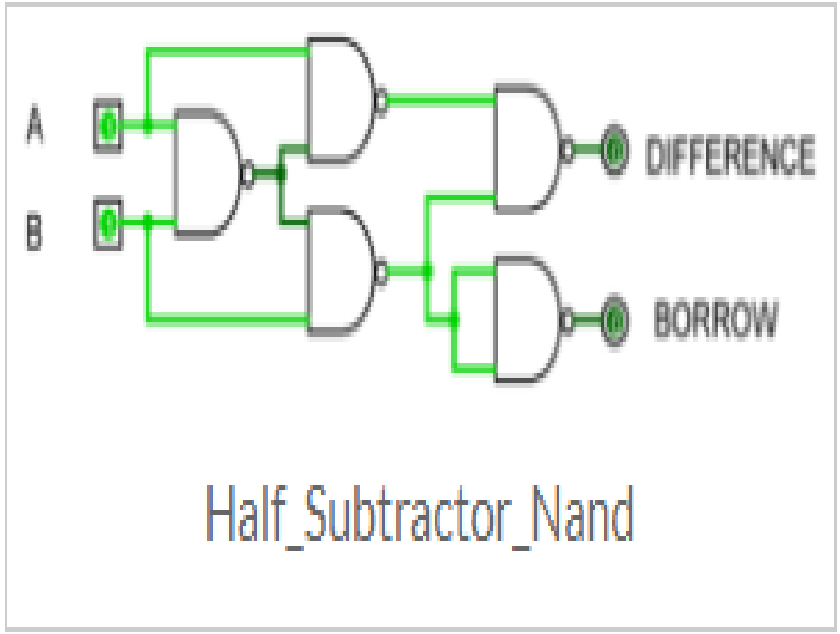


Implement HALF SUBTRACTOR using NAND GATES

$$D = A \oplus B$$

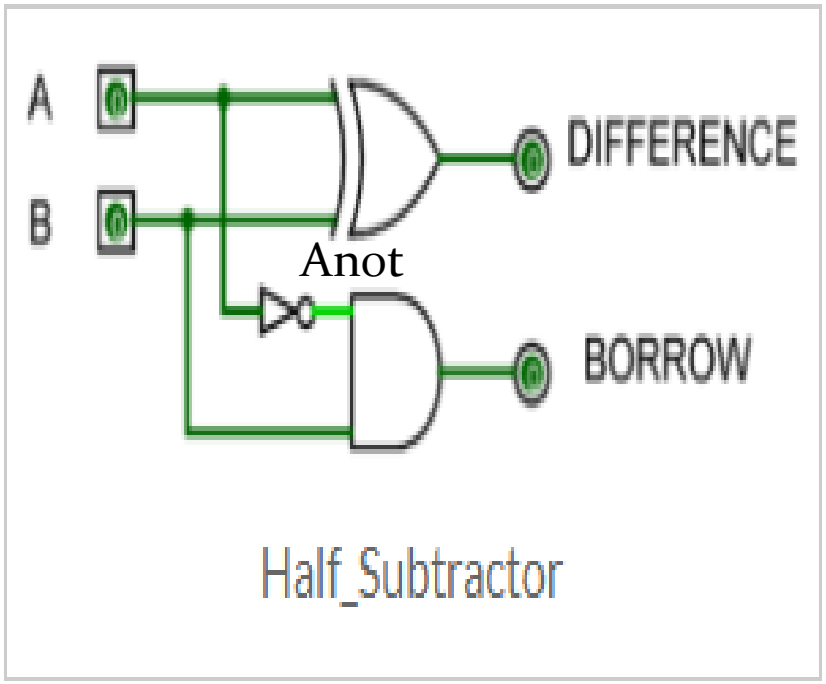
$$B = \overline{A}B$$





HDL for Half Subtractor

```
module  half_subtractor
(output  Difference, Borrow,
input  A, B
);
wire  Anot;
xor  (Difference, A, B);
not  (Anot, A);
and  (Borrow, Anot, B);
endmodule
```



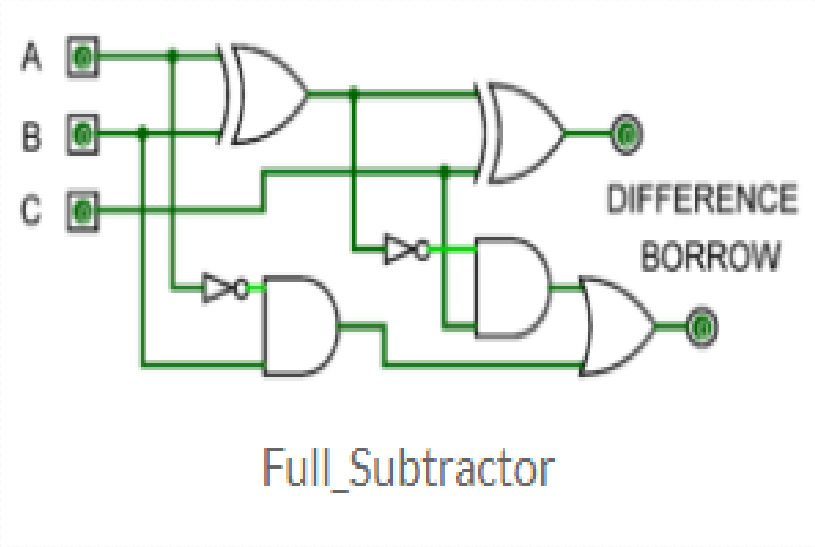
Full Subtractors

- The disadvantage of a half subtractor is overcome by full subtractor. The full subtractor is a combinational circuit with three inputs A,B,C and two output D and C'. A is the 'minuend', B is 'subtrahend', C is the 'borrow' produced by the previous stage, D is the difference output and C' is the borrow output.

Truth Table

Inputs			Output	
A	B	C	(A-B-C)	C'
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

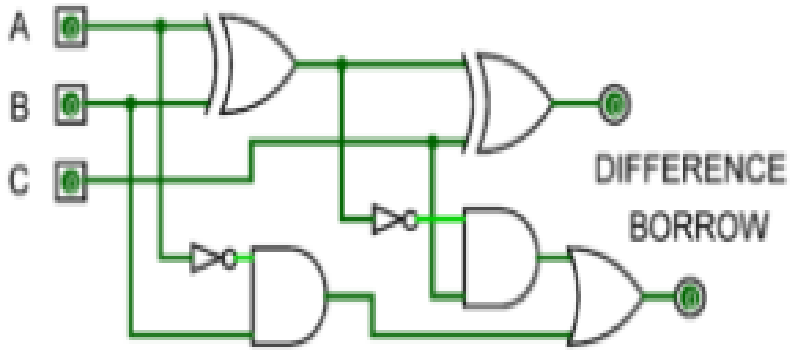
Circuit Diagram



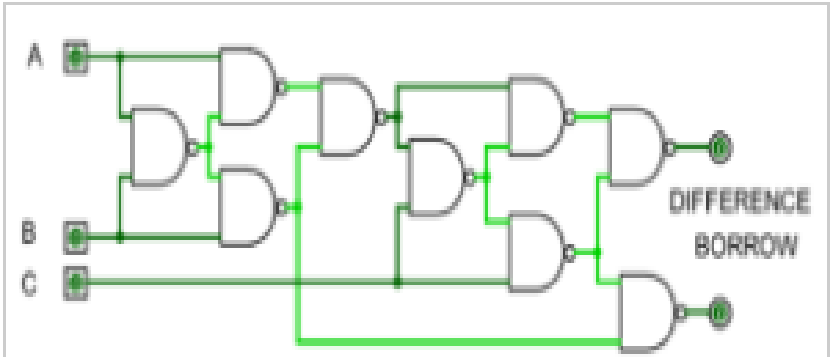
Realize Full SUBTRACTOR using NAND gates only

$$D = A \oplus B \oplus C$$

$$B = \bar{A}B + BC + \bar{A}C$$



Full_Subtractor



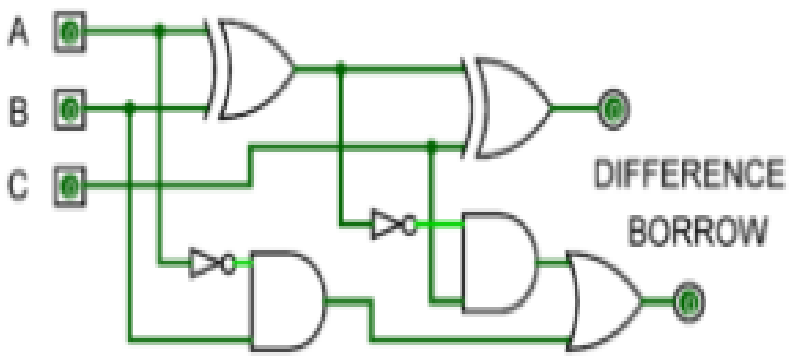
Full_Subtractor_Nand

HDL for Full Subtractor

$$D = A \oplus B \oplus C$$

$$B = \bar{A}B + BC + \bar{A}C$$

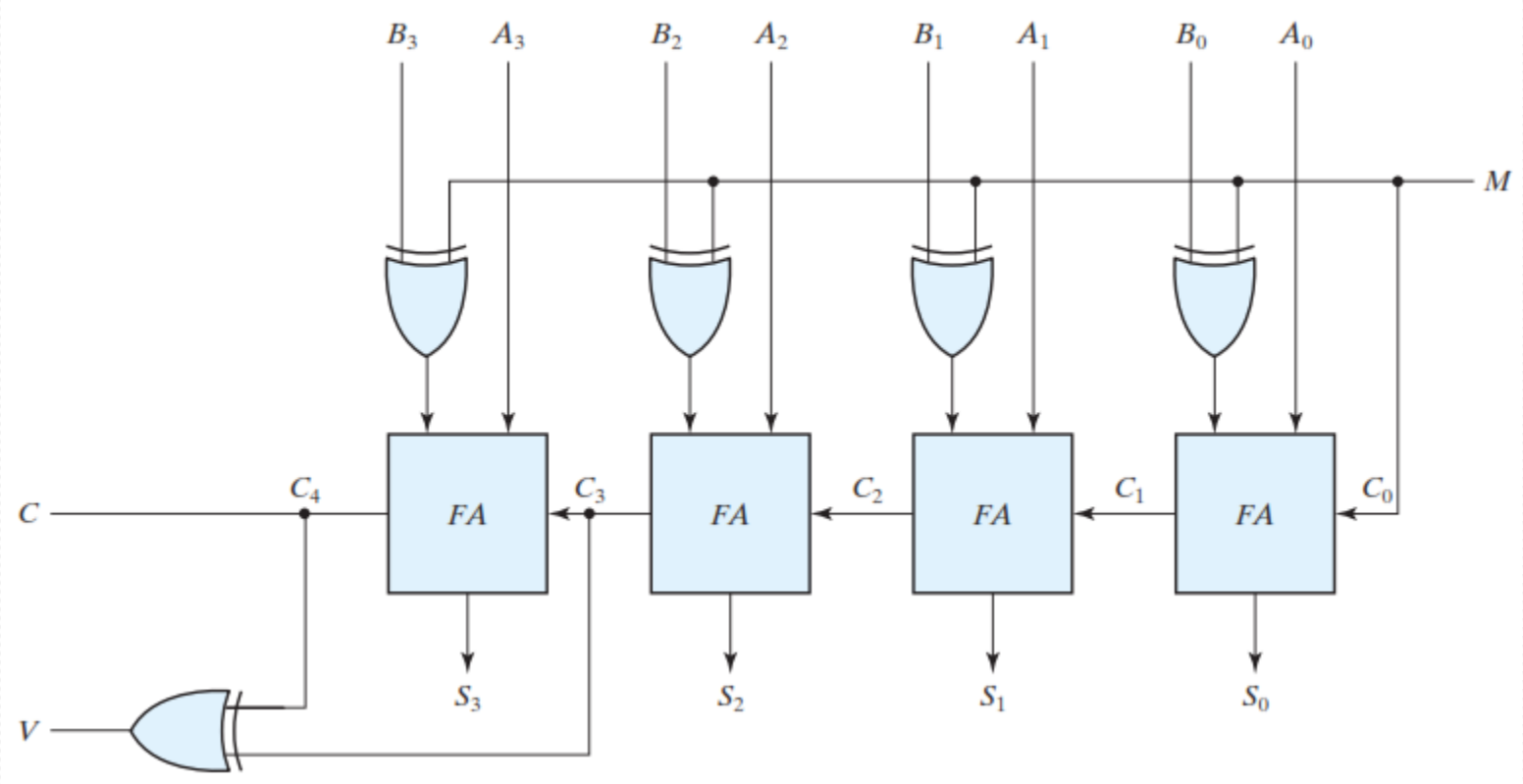
```
// Verilog model Full Subtractor circuit
module  Full-subtractor-df (difference,
borrow, A, B, C);
output difference, borrow;
input A, B, C;
assign  difference = ((!A) && (!B) && (C))
|| ((!A) && (B) && (!C)) || ((A) && (!B) &&
(!C)) || (A &&  B  &&  C);
assign borrow = ((!A) && B) || (B && C) ||
((!A) &&  C );
endmodule
```



Full_Subtractor

Binary Subtractor

The subtraction of unsigned binary numbers can be done most conveniently by means of complements. Remember that the subtraction $A - B$ *can* be done by taking the 2's complement of B *and adding it to* A.



Four-bit adder–subtractor (with overflow detection)

- The circuit for subtracting A - B consists of an adder with inverters placed between each data input B and the corresponding input of the full adder. The input carry C_0 must be equal to 1 when subtraction is performed. The operation thus performed becomes A, plus the 1's complement of B , plus 1. This is equal to A plus the 2's complement of B .

- The addition and subtraction operations can be combined into one circuit with one common binary adder by including an exclusive-OR gate with each full adder. A four-bit adder–subtractor circuit as shown above.

- The mode input M controls the operation. When M = 0 , the circuit is an adder, and when M = 1, the circuit becomes a subtractor.

Overflow

- When two numbers with n digits each are added and the sum is a number occupying n + 1 digits, we say that an overflow occurred.

- The detection of an overflow after the addition of two binary numbers depends on whether the numbers are considered to be signed or unsigned. When two unsigned numbers are added, an overflow is detected from the end carry out of the most significant position. In the case of signed numbers, two details are important:

- The leftmost bit always represents the sign, and negative numbers are in 2's-complement form. When two signed numbers are added, the sign bit is treated as part of the number and the end carry does not indicate an overflow.

- An overflow cannot occur after an addition if one number is positive and the other is negative, since adding a positive number to a negative number produces a result whose magnitude is smaller than the larger of the two original numbers.

- An overflow may occur if the two numbers added are both positive or both negative.

- An overflow condition can be detected by observing the carry into the sign bit position and the carry out of the sign bit position. If these two carries are not equal, an overflow has occurred.

carries:	0	1	carries:	1	0
+70	0	1000110	−70	1	0111010
+80	0	1010000	−80	1	0110000
+150	1	0010110	−150	0	1101010

- This is indicated in the examples in which the two carries are explicitly shown. If the two carries are applied to an exclusive-OR gate, an overflow is detected when the output of the gate is equal to 1.

- The binary adder–subtractor circuit with outputs *C and V is shown in above Fig. If the* two binary numbers are considered to be unsigned, then the *C bit detects a carry after* addition or a borrow after subtraction. If the numbers are considered to be signed, then the *V bit detects an overflow.*

THANK YOU