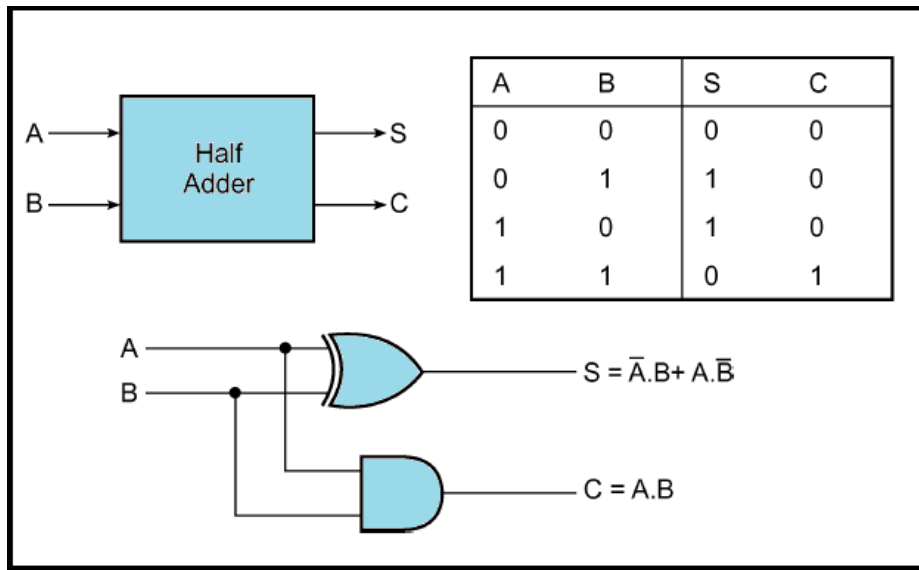


HALF ADDER

DEFINITION:

The half adder is able to add two single binary digits and provide the output plus a carry value. It has two inputs, called A and B, and two outputs S (sum) and C (carry). The common representation uses a XOR logic gate and an AND logic gate.



EXPLANATION:

A half adder is used to add two single-digit binary numbers and results into a two-digit output. It is named as such because putting two half adders together with the use of an OR gate results in a full adder. In other words, it only does half the work of a full adder.

The adder works by combining the operations of basic logic gates, with the simplest form using only a XOR and an AND gate. This can also be converted into a circuit that only has AND, OR and NOT gates. This is especially useful since these three simpler logic gate ICs (integrated circuits) are more common and available than the XOR IC, though this might result in a bigger circuit since three different chips are used instead of just one.

ADVANTAGES:

- Simple design, the basic building block to understanding 1-bit addition
- Just with an inverter, it can be converted to the half subtractor.

DISADVANTAGES:

- Half adders have no scope of adding the carry bit resulting from the addition of previous bits.
- The real-time scenarios involve adding the multiple numbers of bits which cannot be accomplished using half adder.
- It is not suitable for cascading for multi-bit additions.
- To get rid of this problem, a full adder is required which adds three 1 bit.
- It does not incorporate the previous carry for addition.

APPLICATIONS:

The applications of this basic adder are as follows

- To perform additions on binary bits the Arithmetic and Logic Unit present in the computer prefers this adder circuit.
- The combination of half adder circuits leads to the formation of the Full Adder circuit.
- These logic circuits are preferred in the design of calculators.
- To calculate the addresses and tables these circuits are preferred.
- Instead of only addition, these circuits are capable of handling various applications in digital circuits. Further, this becomes

RTL CODE:

DATA FLOW REPRESENTATION

```
module half_adder(s,c,a,b);  
    output s,c;  
    input a,b;  
    assign s=a^b; // data flow expression for sum  
    assign c=a&b; // data flow expression for carry  
endmodule
```

STRUCTURAL REPRESENTATION:

```
module half_adder(s,c,a,b);  
    output s,c;  
    input a,b;  
    xor g1(s,a,b); //xor gate for sum  
    and g1(c,a,b); // AND gate for carry  
endmodule
```

BEHAVIORAL REPRESENTATION

```
module half_adder(s,c,a,b);  
    output s,c;  
    input a,b;  
    always @ (*)begin  
        s=a^b; // xor operation for sum  
        c=a&b; // and operation for carry  
    end  
endmodule
```

TEST BENCH

```
module test_best;  
    reg a,b;  
    wire s,c;  
    // instantating design module  
  
    half_adder a1(s,c,a,b);  
    initial  
    begin  
        $dumpfile("dump.vcd");
```

```

    $dumpvars(1);
end
initial
begin
    a=0;b=0;
end
initial
begin
    #10 a=1;b=0;
    #10 a=0;b=1;
    #10 a=1;b=1;
end
initial
begin
    $monitor($time,"a=%b,b=%b,c=%b,s=%d",a,b,s,c);
    #60 $finish();
end
endmodule

```

