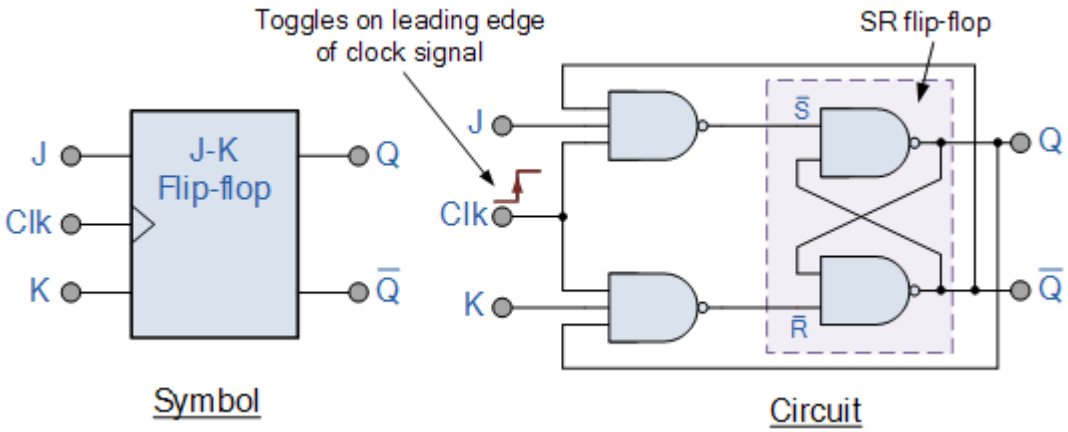
DAY 22

JK FLIPFLOP:

Both the S and the R inputs of the previous SR bistable have now been replaced by two inputs called the J and K inputs, respectively after its inventor Jack Kilby. Then this equates to: J = S and K = R.

The two 2-input AND gates of the gated SR bistable have now been replaced by two 3-input NAND gates with the third input of each gate connected to the outputs at Q and Q. This cross coupling of the SR flip-flop allows the previously invalid condition of S = "1" and R = "1" state to be used to produce a "toggle action" as the two inputs are now interlocked.

If the circuit is now "SET" the J input is inhibited by the "0" status of Q through the lower NAND gate. If the circuit is "RESET" the K input is inhibited by the "0" status of Q through the upper NAND gate. As Q and Q are always different we can use them to control the input. When both inputs J and K are equal to logic "1", the JK flip flop toggles as shown in the following truth table.



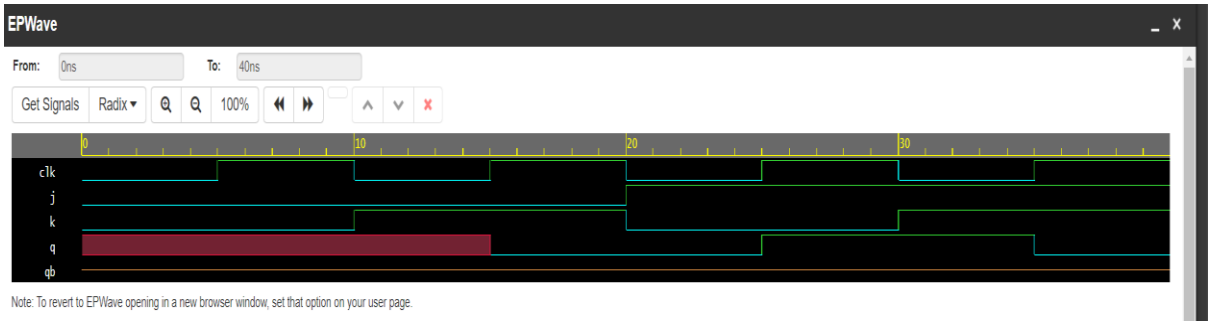
RTL CODE:

```
module JK_flipflop( input clk,j,k, output reg q, qb);
    always @(posedge clk) begin
      case({j, k})
            2'b00: q <= q;
            2'b01: q <= 1'b0;
            2'b10: q <= 1'b1;
            2'b11: q <= !q;
        endcase
    end
endmodule
```

TEST BENCH:

```
module testbench;
  reg clk,j,k;
  wire q,qb;
  JK_flipflop a1 (clk,j,k,q,qb);
  initial
    begin
      $dumpfile(".vcd");
      $dumpvars(1);
    end
  always
```

```
    begin
      clk = 0;
      #5 clk = 1;
      #5 clk = 0;
    end
  initial
    begin
      j = 0; k = 0;
      #10 j = 0; k = 1;
      #10 j = 1; k = 0;
      #10 j = 1; k = 1;
      #10 $finish;
    end
  always @(posedge clk) begin
    $display("Time %0t: j = %b, k = %b, q = %b, qb = %b", $time, j, k,
q, qb);
  end
endmodule
```


EPWave
From:
0ns
To:
40ns
Get Signals
Radix
100%
0
10
20
30
clk
j
k
q
qb
Note: To revert to EPWave opening in a new browser window, set that option on your user page.