ALL ABOUT FPGA

Menu

VHDL Code for Flipflop – D,JK,SR,T

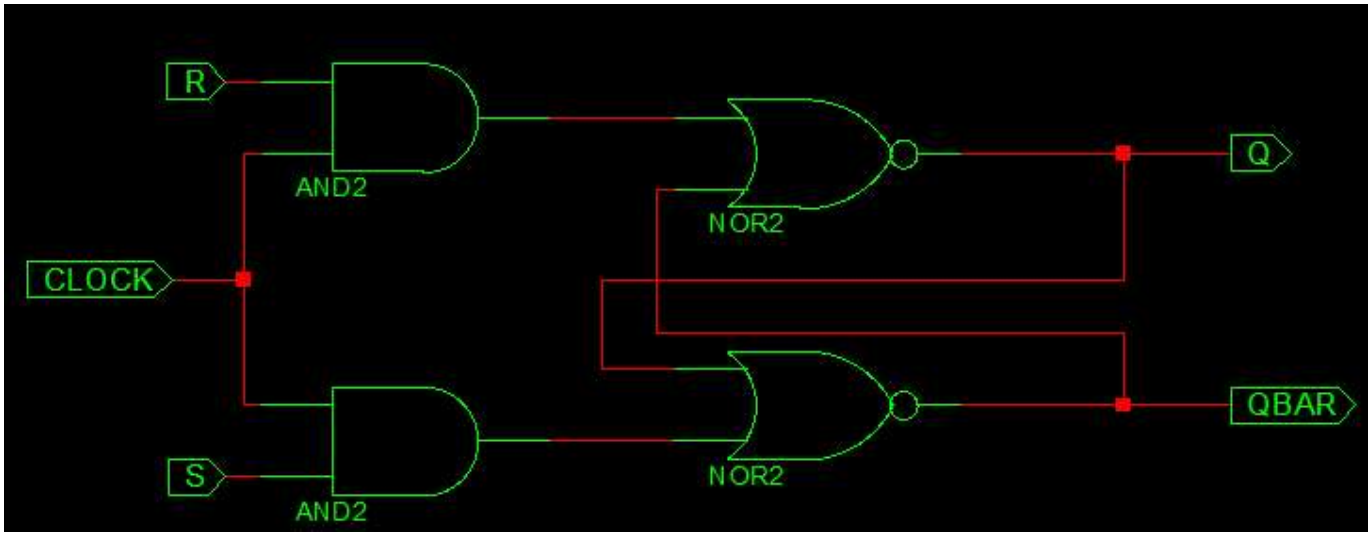
July 26, 2014 by Admin

All flip-flops can be divided into four basic types: SR, JK, D and T. They differ in the number of inputs and in the response invoked by different value of input signals.

Contents [hide]

SR FlipFlop

A flip-flop circuit can be constructed from two NAND gates or two NOR gates. These flip-flops are shown in Figure. Each flip-flop has two outputs, Q and Q', and two inputs, set and reset. This type of flip-flop is referred to as an SR flip-flop.



SR Flipflop truth table

Q	S	R	Q(T+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	UNKNOWN
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	UNKNOWN

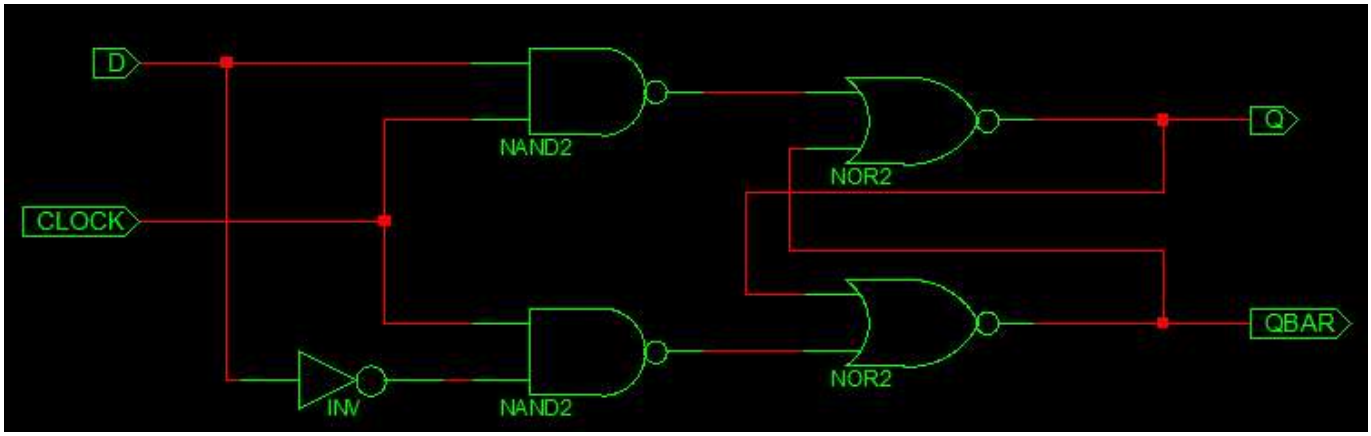
VHDL Code for SR FlipFlop

```
1  library ieee;
2  use ieee. std_logic_1164.all;
3  use ieee. std_logic_arith.all;
4  use ieee. std_logic_unsigned.all;
5
6  entity SR_FF is
7  PORT( S,R,CLOCK: in std_logic;
8  Q, QBAR: out std_logic);
9  end SR_FF;
10
11 Architecture behavioral of SR_FF is
12 begin
13 PROCESS(CLOCK)
14 variable tmp: std_logic;
```

```
15  begin
16  if(CLOCK='1' and CLOCK'EVENT) then
17  if(S='0' and R='0')then
18  tmp:=tmp;
19  elsif(S='1' and R='1')then
20  tmp:='Z';
21  elsif(S='0' and R='1')then
22  tmp:='0';
23  else
24  tmp:='1';
25  end if;
26  end if;
27  Q <= tmp;
28  QBAR <= not tmp;
29  end PROCESS;
30  end behavioral;
```

D FlipFlop

The D flip-flop shown in figure is a modification of the clocked SR flip-flop. The D input goes directly into the S input and the complement of the D input goes to the R input. The D input is sampled during the occurrence of a clock pulse. If it is 1, the flip-flop is switched to the set state (unless it was already set). If it is 0, the flip-flop switches to the clear state.



D Flipflop truth table

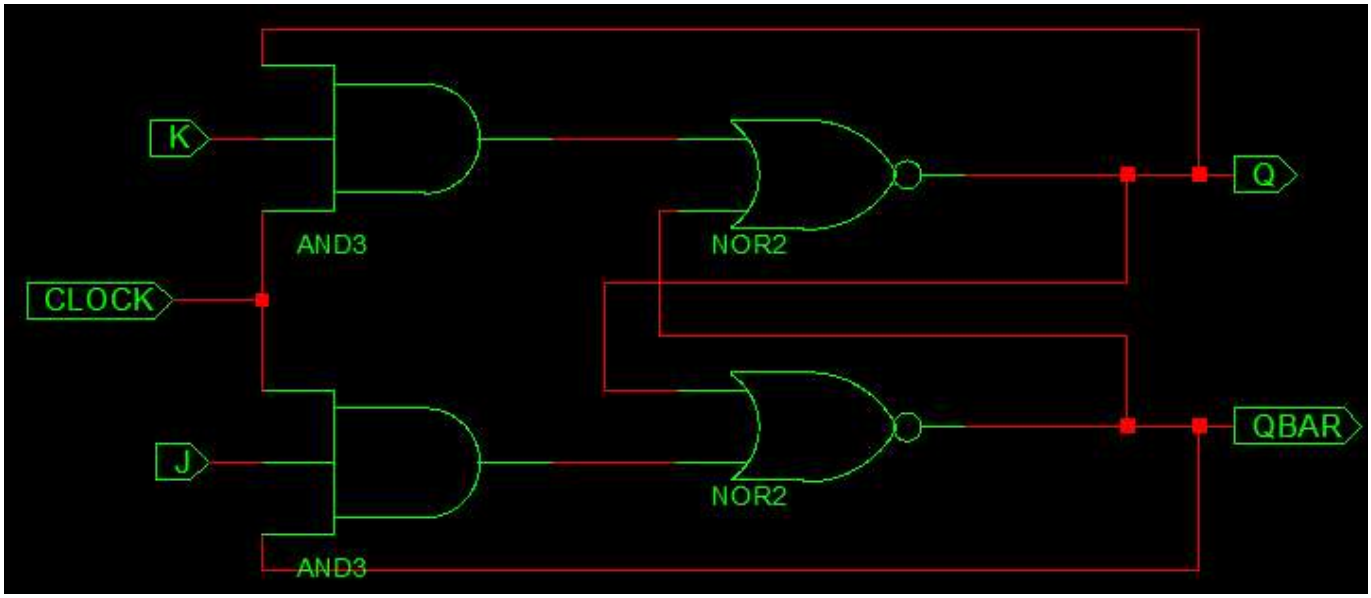
Q	D	Q(T+1)
0	0	0
0	1	1
1	0	0
1	1	1

VHDL Code for D FlipFlop

```
1  library ieee;
2  use ieee. std_logic_1164.all;
3  use ieee. std_logic_arith.all;
4  use ieee. std_logic_unsigned.all;
5
6  entity D_FF is
7  PORT( D,CLOCK: in std_logic;
8  Q: out std_logic);
9  end D_FF;
10
11 architecture behavioral of D_FF is
12 begin
13 process(CLOCK)
14 begin
15 if(CLOCK='1' and CLOCK'EVENT) then
16 Q <= D;
17 end if;
18 end process;
19 end behavioral;
```

JK FlipFlop

A JK flip-flop is a refinement of the SR flip-flop in that the indeterminate state of the SR type is defined in the JK type. Inputs J and K behave like inputs S and R to set and clear the flip-flop (note that in a JK flip-flop, the letter J is for set and the letter K is for clear).



JK Flipflop truth table

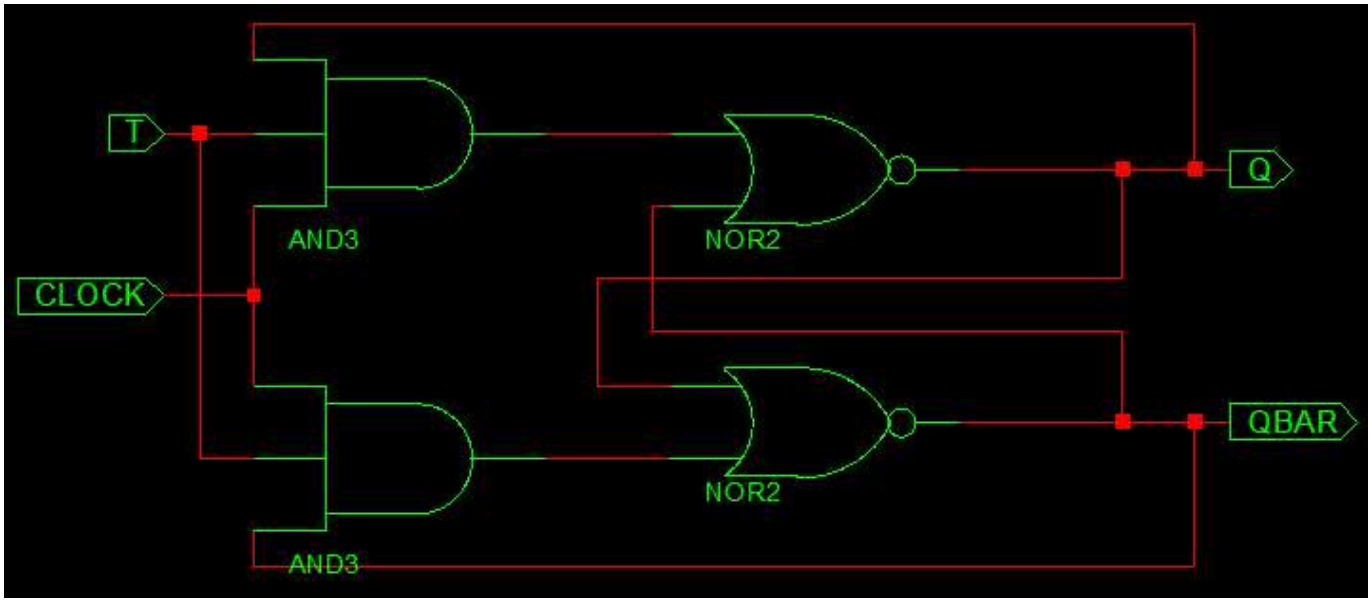
Q	J	K	Q(T+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

VHDL Code for JK FlipFlop

```
1  library ieee;
2  use ieee. std_logic_1164.all;
3  use ieee. std_logic_arith.all;
4  use ieee. std_logic_unsigned.all;
5
6  entity JK_FF is
7  PORT( J,K,CLOCK: in std_logic;
8  Q, QB: out std_logic);
9  end JK_FF;
10
11 Architecture behavioral of JK_FF is
12 begin
13 PROCESS(CLOCK)
14 variable TMP: std_logic;
15 begin
16 if(CLOCK='1' and CLOCK'EVENT) then
17 if(J='0' and K='0')then
18 TMP:=TMP;
19 elsif(J='1' and K='1')then
20 TMP:= not TMP;
21 elsif(J='0' and K='1')then
22 TMP:='0';
23 else
24 TMP:='1';
25 end if;
26 end if;
27 Q<=TMP;
28 Q <=not TMP;
29 end PROCESS;
30 end behavioral;
```

T FlipFlop

The T flip-flop is a single input version of the JK flip-flop. As shown in figure, the T flip-flop is obtained from the JK type if both inputs are tied together. The output of the T flip-flop “toggles” with each clock pulse.



T Flipflop truth table

Q	T	Q(T+1)
0	0	0
0	1	1
1	0	1
1	1	0

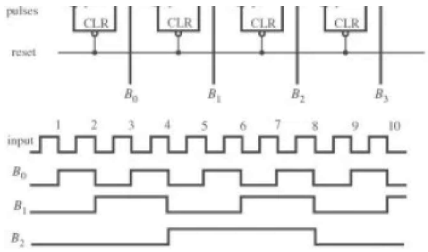
VHDL Code for T FlipFlop

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity T_FF is
5  port( T: in std_logic;
6  Clock: in std_logic;
7  Q: out std_logic);
8  end T_FF;
9
10 architecture Behavioral of T_FF is
11 signal tmp: std_logic;
12 begin
13 process (Clock)
14 begin
15 if Clock'event and Clock='1' then
16
```

```
17  if T='0' then
18  tmp <= tmp;
19  elsif T='1' then
20  tmp <= not (tmp);
21  end if;
22  end if;
23  end process;
24  Q <= tmp;
25  end Behavioral;
```

Download Post as PDF

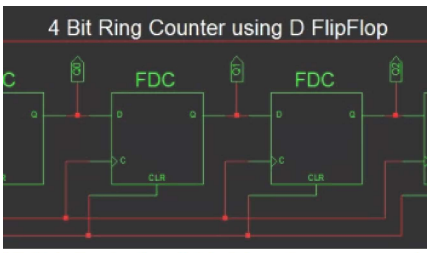
Related



VHDL Code for 4-Bit Binary Up Counter

February 13, 2014

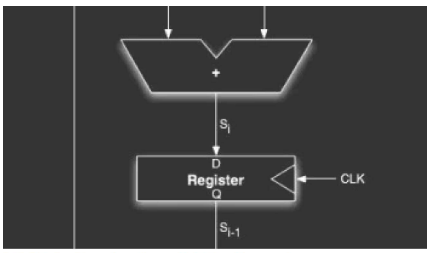
In "VHDL"



VHDL Code for 4-bit Ring Counter and Johnson Counter

May 22, 2016

In "VHDL"



VHDL Code for 4-Bit Aynchronous Accumulator

February 4, 2016

In "VHDL"