

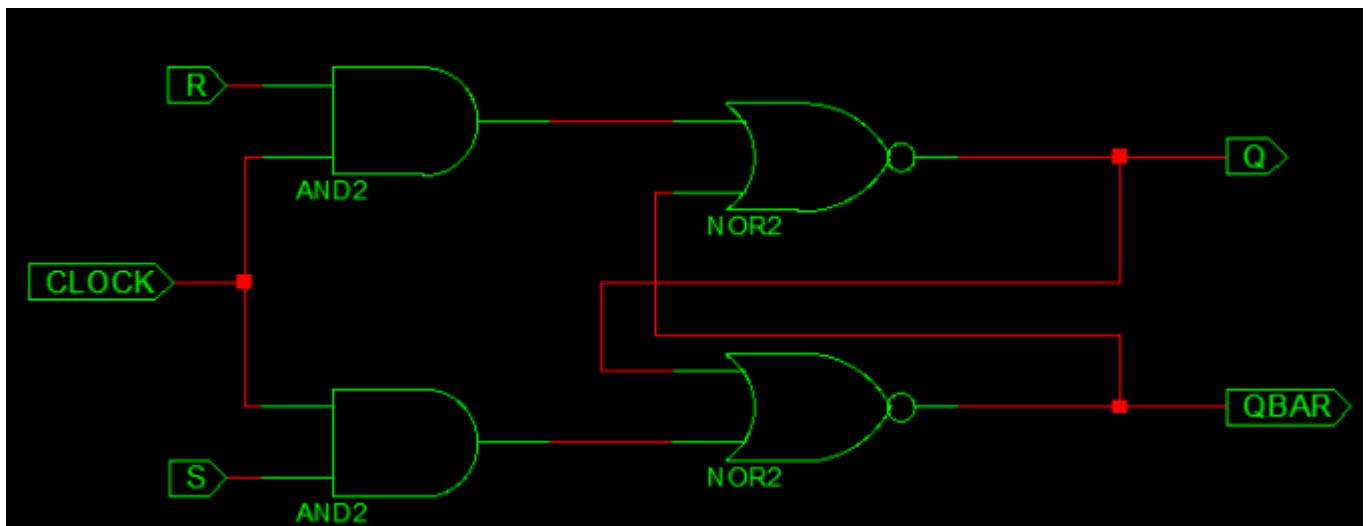
All flip-flops can be divided into four basic types: SR, JK, D and T. They differ in the number of inputs and in the response invoked by different value of input signals.

Contents

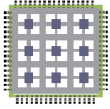
- 1 SR FlipFlop
- 2 VHDL Code for SR FlipFlop
- 3 D FlipFlop
- 4 VHDL Code for D FlipFlop
- 5 JK FlipFlop
- 6 VHDL Code for JK FlipFlop
- 7 T FlipFlop
- 8 VHDL Code for T FlipFlop

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



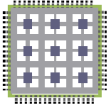
Q	S	R	Q(T+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	UNKNOWN
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	UNKNOWN

VHDL Code for SR FlipFlop

```
library ieee;
use ieee. std_logic_1164.all;
use ieee. std_logic_arith.all;
use ieee. std_logic_unsigned.all;

entity SR_FF is
PORT( S,R,CLOCK: in std_logic;
Q, QBAR: out std_logic);
end SR_FF;

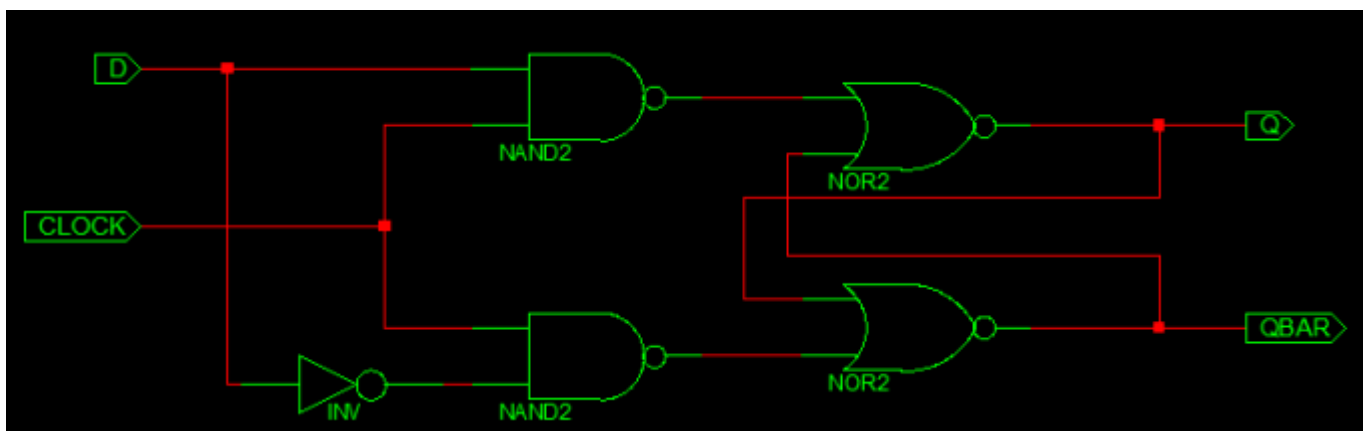
Architecture behavioral of SR_FF is
begin
PROCESS(CLOCK)
variable tmp: std_logic;
begin
if(CLOCK='1' and CLOCK'EVENT) then
if(S='0' and R='0')then
```



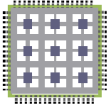
```
tmp:=tmp;
elsif(S='1' and R='1')then
tmp:='Z';
elsif(S='0' and R='1')then
tmp:='0';
else
tmp:='1';
end if;
end if;
Q <= tmp;
QBAR <= not tmp;
end PROCESS;
end behavioral;
```

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



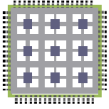
Q	D	Q(T+1)
0	0	0
0	1	1
1	0	0
1	1	1

VHDL Code for D FlipFlop

```
library ieee;
use ieee. std_logic_1164.all;
use ieee. std_logic_arith.all;
use ieee. std_logic_unsigned.all;

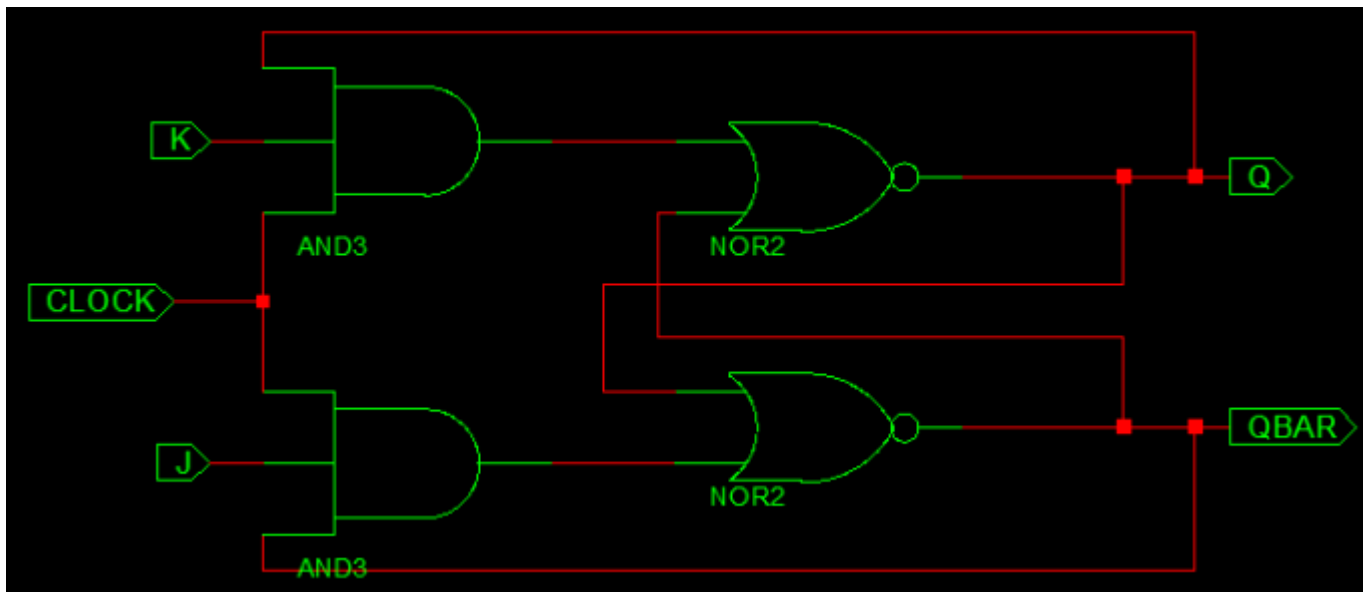
entity D_FF is
PORT( D,CLOCK: in std_logic;
Q: out std_logic);
end D_FF;

architecture behavioral of D_FF is
begin
process(CLOCK)
begin
if(CLOCK='1' and CLOCK'EVENT) then
Q <= D;
end if;
end process;
end behavioral;
```



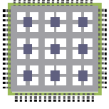
JK FlipFlop

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

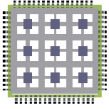


VHDL Code for JK FlipFlop

```
library ieee;
use ieee. std_logic_1164.all;
use ieee. std_logic_arith.all;
use ieee. std_logic_unsigned.all;

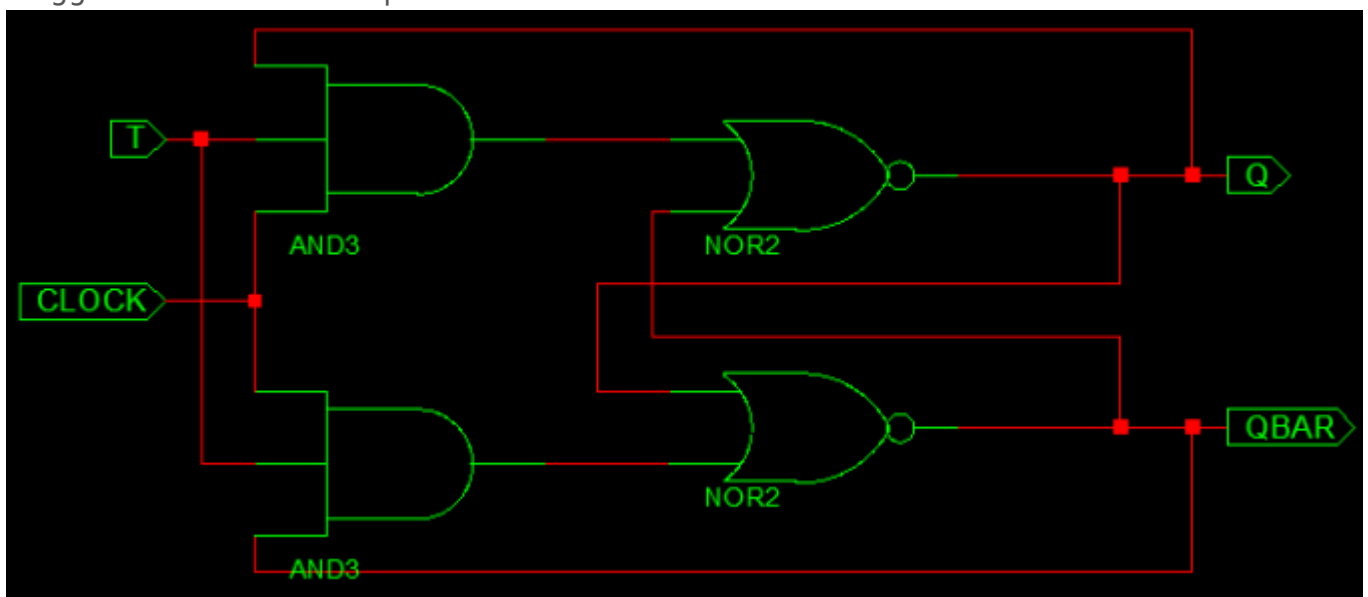
entity JK_FF is
PORT( J,K,CLOCK: in std_logic;
Q, QB: out std_logic);
end JK_FF;

Architecture behavioral of JK_FF is
begin
PROCESS(CLOCK)
variable TMP: std_logic;
begin
if(CLOCK='1' and CLOCK'EVENT) then
if(J='0' and K='0')then
TMP:=TMP;
elsif(J='1' and K='1')then
TMP:= not TMP;
elsif(J='0' and K='1')then
TMP:='0';
else
TMP:='1';
end if;
end if;
Q<=TMP;
Q <=not TMP;
end PROCESS;
end behavioral;
```

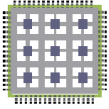


T FlipFlop

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



VHDL Code for T FlipFlop

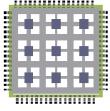
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity T_FF is
port( T: in std_logic;
Clock: in std_logic;
Q: out std_logic);
end T_FF;

architecture Behavioral of T_FF is
signal tmp: std_logic;
begin
```

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

```
process (Clock)
begin
if Clock'event and Clock='1' then
if T='0' then
tmp <= tmp;
elsif T='1' then
tmp <= not (tmp);
end if;
end if;
end process;
Q <= tmp;
```

```
end Behavioral;
```