

3 bit Asynchronous Up counter :

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

entity jk_ff is
Port ( j : in STD_LOGIC;
k : in STD_LOGIC;
clock : in STD_LOGIC;
reset : in STD_LOGIC;
q : out STD_LOGIC);
end jk_ff;

architecture Behavioral of jk_ff is
signal jk : std_logic_vector(1 downto 0) := "00";
signal qsig : std_logic := '0';
begin
jk <= j & k;
process(reset,clock)
begin
if (reset = '1')then
qsig <='0';
elsif (clock'event and clock = '1')then
case (jk) is
when "00" => qsig <= qsig;
when "01" => qsig <= '0';
when "10" => qsig <= '1';
when others => qsig <= not qsig;
end case;
end if;
end process;
q <= qsig;
end Behavioral;
```

--Step2. VHDL code for 3 bit Counter using structural modeling

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

entity counter_test is
Port ( clock : in STD_LOGIC;
reset : in STD_LOGIC;
count : out STD_LOGIC_VECTOR (2 downto 0));
end counter_test;

architecture structural_test of counter_test is

component jk_ff is
Port ( j : in STD_LOGIC;
```

```

k : in STD_LOGIC;
clock : in STD_LOGIC;
reset : in STD_LOGIC;
q : out STD_LOGIC;
end component;

signal temp:std_logic_vector(2 downto 0) := "000";

begin

label1 : jk_ffport map ( reset => reset, clock => clock, j => '1', k => '1', q => temp(2));
label2 : jk_ff port map (reset => reset,clock =>NOT temp(2), j => '1', k => '1', q =>
temp(1));
label3 : jk_ff port map ( reset => reset,clock => NOT temp(1), j => '1', k => '1',q =>
temp(0));
count(2) <= temp(0);
count(1) <= temp(1);
count(0) <= temp(2);

end structural_test;

```

--Step3. Test Bench Code

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY tb_test1 IS
END tb_test1;
ARCHITECTURE behavior OF tb_test1 IS
COMPONENT counter_test
PORT(
clock : IN std_logic;
reset : IN std_logic;
count : OUT std_logic_vector(2 downto 0)
);
END COMPONENT;
--Inputs
signal clock : std_logic := '0';
signal reset : std_logic := '0';
--Outputs
signal count : std_logic_vector(2 downto 0);
BEGIN
-- Instantiate the Unit Under Test (UUT)
uut: counter_test PORT MAP (
clock => clock,
reset => reset,
count => count
);

process

```

```
begin  
wait for 5ns;  
clock <= not clock;  
end process;
```

```
process  
begin  
reset <= '1';  
wait for 50ns;  
reset <= not reset;  
wait;  
end process;
```

```
END;
```

3 bit Asynchronous Down Counter :

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

entity jk_ff is
Port ( j : in STD_LOGIC;
      k : in STD_LOGIC;
      clock : in STD_LOGIC;
      reset : in STD_LOGIC;
      q : out STD_LOGIC);
end jk_ff;

architecture Behavioral of jk_ff is
signal jk : std_logic_vector(1 downto 0) := "00";
signal qsig : std_logic := '0';
begin
jk <= j & k;
process(reset,clock)
begin
if (reset = '1')then
qsig <= '0';
elsif (clock'event and clock = '1')then
case (jk) is
when "00" => qsig <= qsig;
when "01" => qsig <= '0';
when "10" => qsig <= '1';
when others => qsig <= not qsig;
end case;
end if;
end process;
q <= qsig;
end Behavioral;
```

--Step2. VHDL code for 3 bit Counter using structural modeling

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

entity counter_test is
Port ( clock : in STD_LOGIC;
      reset : in STD_LOGIC;
      count : out STD_LOGIC_VECTOR (2 downto 0));
end counter_test;

architecture structural_test of counter_test is
component jk_ff is
Port ( j : in STD_LOGIC;
      k : in STD_LOGIC;
      clock : in STD_LOGIC;
      reset : in STD_LOGIC;
      q : out STD_LOGIC);
end component;
```

```

signal temp:std_logic_vector(2 downto 0) := "000";
begin
label1 : jk_ff
port map ( reset => reset, clock => clock, j => '1', k => '1', q => temp(2));
label2 : jk_ff port map (reset => reset,clock => temp(2), j => '1', k => '1', q => temp(1));
label3 : jk_ff port map ( reset => reset,clock => temp(1), j => '1', k => '1',q => temp(0));
count(2) <= temp(0);
count(1) <= temp(1);
count(0) <= temp(2);
end structural_test;

```

--Step3. Test Bench Code

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY tb_test1 IS
END tb_test1;
ARCHITECTURE behavior OF tb_test1 IS
COMPONENT counter_test
PORT(
clock : IN std_logic;
reset : IN std_logic;
count : OUT std_logic_vector(2 downto 0)
);
END COMPONENT;
--Inputs
signal clock : std_logic := '0';
signal reset : std_logic := '0';
--Outputs
signal count : std_logic_vector(2 downto 0);
BEGIN
-- Instantiate the Unit Under Test (UUT)
uut: counter_test PORT MAP (
clock => clock,
reset => reset,
count => count
);

process
begin
wait for 5ns;
clock <= not clock;
end process;

process
begin
reset <= '1';
wait for 50ns;
reset <= not reset;
wait;
end process;

END;

```