ALU

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_arith.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;
entity BBBB is
   Port ( a, b, Cin, s : in  STD_LOGIC_vector(3 downto 0);
          y : out  STD_LOGIC_vector(3 downto 0));
end BBBB;
architecture Behavioral of BBBB is
begin
with s(3 downto 0) select
y <= a + b + Cin when "0000",
a - b when "0001",
a + '1' when "0010",
a - '1' when "0011",
a + b when "0100",
b - '1' when "0101",
b when "0110",
b + '1' when "0111",
not a when "1000",
a and b when "1001",
a or b when "1010",
a xor b when "1011",
a nand b when "1100",
a nor b when "1101",
a xnor b when "1110",
not b when others;
end Behavioral;
```

JK

```
Library ieee;
Use ieee.std_logic_1164.all;
Entity FF is
Port (J, K, reset, clk: in std_logic;
q :out std_logic);
End FF;
Architecture Behavior of FF is
Signal qa: std_logic;
begin
process (clk, reset)
Begin
if (clk’event and clk=‘1’) then
if reset=‘1’ then
qa<=‘0’;
elsif (j=‘0’ and k=‘0’) then
qa<=qa;
elsif (j=‘0’ and k=‘1’) then
qa<=‘0’;
elsif (j=‘1’ and k=‘0’) then
qa<=‘1’;
else
qa<= not qa;
end if;
end if;
end process;
q<=qa;
end Behavior;
```

UP_DOWN

```
Library IEEE;
use ieee.STD_LOGIC_1164.ALL;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity UPDOWN is
port (clk, reset, up_down: in std_logic;
 q:out std_logic_vector(3 downto 0));
end UPDOWN;
architecture Behavioral of UPDOWN is
signal qa: std_logic_vector (3 downto 0);
begin
 process (clk, reset)
begin
 if (clk'event and clk='1') then
 if reset='1' then
 qa<="0000";
elsif (up_down='1') then
 qa<=qa+'1';
 else
qa<=qa-'1';
 end if;
 end if;
end process;
q<=qa;
end Behavioral;
```

SHIFT

```
Library ieee;
use ieee.std_logic_1164.all;
entity shift is
Port (clk, reset, din, dir: in std_logic;
q:out std_logic_vector(3 downto 0));
end shift;
architecture NNNN of shift is
signal qa: std_logic_vector (3 downto 0);
begin
Process (clk, reset)
begin
if reset=’1’ then
qa<=”0000”;
elsif (clk ’event and clk=’1’) then
if dir=’1’ then
qa (3) <=din;
qa (2) <=qa (3);
qa (1) <=qa (2);
qa (0) <= qa (1);
else
qa (3) <=qa (2);
qa (2) <=qa (1);
qa (1) <=qa (0);
qa (0) <= din;
end if;
end if;
end process;
q<=qa;
end NNNN;
```