```
entity ALU_4bit is

Port (A: in STD_LOGIC_VECTOR (3 downto 0);

B: in STD_LOGIC_VECTOR (3 downto 0); F: in STD_LOGIC_VECTOR (2 downto 0);

Y:out STD_LOGIC_VECTOR (3 downto 0);

C_B: out STD_LOGIC

); end ALU_4bit;

architecture ALU_4bit_arch of ALU_4bit is signal result:STD_LOGIC_VECTOR(4 downto 0):="00000";

begin

process(A,B,F)

begin

CASE F IS

CB

X

X

X

X
```

X

X

```
when "000" => result <= '0' & (A AND B);

when "001" => result <= '0' & (A NAND B);

when "010" => result <= '0' & (A OR B);

when "011" => result <= '0' & (A XOR B);
```

rry

```
when "100" => result <= '0' & (A XNOR B);
```

row

```
when "101" => result <= '0' & (A NOR B);

when "110" => result <= ('0' & A)+('0' & B);
when others =>

if A<B then result <= '0' & (NOT B); result <= result+1; result <= ('0' & A) + result; result <= (NOT result) +1; result <= (NOT(('0' & A) + ('0' &(NOT B)) + 1))+1; else result <= ('0' & A)-('0' & B); end if;

end CASE;

end process;
```

```
Y <= result(3 downto 0);

C_B <= result(4);

end ALU_4bit_arch;

tbbb

COMPONENT ALU_4bit

PORT(

A: IN std_logic_vector(3 downto 0);

B: IN std_logic_vector(3 downto 0);

F: IN std_logic_vector(2 downto 0);

Y: OUT std_logic_vector(3 downto 0); C_B: OUT std_logic

);

END COMPONENT;
--Inputs

signal A: std_logic_vector(3 downto 0) := "0010"; signal B: std_logic_vector(3 downto 0) := "1111";

signal F: std_logic_vector(2 downto 0) := (others => '1');

--Outputs
```

```
signal Y: std_logic_vector(3 downto 0);

signal C_B: std_logic;

- No clocks detected in port list. Replace <clock> below with

- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: ALU_4bit PORT MAP (

A => A,

B => B,

F => F,

Y => Y,

C_B => C_B

);

-- Stimulus process

stim_proc_F: process begin F <= F + 1; wait for 25 ns;
end process;
END;
```