

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

```

```

entity sign_add_sub is
    generic(
        n : integer := 4); -- number of input bits
    port(a, b: in std_logic_vector(n-1 downto 0);
        cin: in std_logic := '0';
        sum, sub: out std_logic_vector(n-1 downto 0);
        cout_sum, cout_sub: out std_logic);
end sign_add_sub;

```

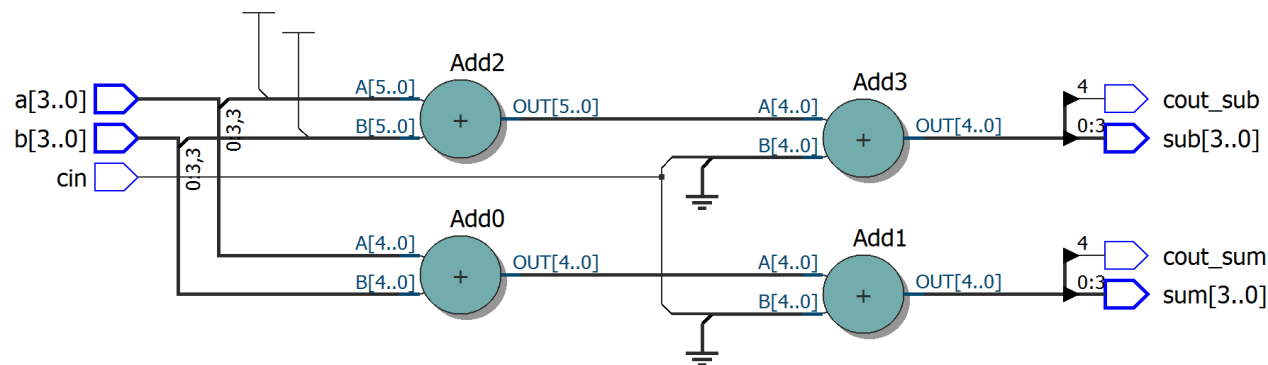
```

architecture add_sub of sign_add_sub is
    signal a_sig, b_sig: unsigned(n-1 downto 0);
    signal sum_sig, sub_sig: unsigned(n downto 0);
begin
    --convert to unsigned
    a_sig <= unsigned(a);
    b_sig <= unsigned(b);
    --add and subtract
    sum_sig <= (a_sig(n-1) & a_sig) + (b_sig(n-1) & b_sig) + ('0' & cin); -- add a mux and register to increment
    sub_sig <= (a_sig(n-1) & a_sig) - (b_sig(n-1) & b_sig) - ('0' & cin); -- same same to decrement

    sum <= std_logic_vector(sum_sig(n-1 downto 0));
    cout_sum <= std_logic(sum_sig(n));

    sub <= std_logic_vector(sub_sig(n-1 downto 0));
    cout_sub <= std_logic(sub_sig(n));
end add_sub;

```



```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

```

```

entity tbsign_add_sub is
end entity;

```

```

architecture beh of tbsign_add_sub is

```

```

    component sign_add_sub is
        generic(n : integer);
        port(a, b: in std_logic_vector(n-1 downto 0);
            cin: in std_logic := '0';
            sum, sub: out std_logic_vector(n-1 downto 0);
            cout_sum, cout_sub: out std_logic);
    end component;

```

```

    constant n : integer := 4;
    signal a, b: std_logic_vector(n-1 downto 0);
    signal cin: std_logic := '0';
    signal sum, sub: std_logic_vector(n-1 downto 0);
    signal cout_sum, cout_sub: std_logic;

```

```

    signal lfsr0 : std_logic_vector(0 to n-1) := (0 => '1', others => '0');
    signal lfsr1 : std_logic_vector(0 to n-1) := (0 => '1', others => '0');

```

```

    constant period : time := 50 ns;
    constant strobe : time := 45 ns;

```

```

begin

```

```

    p0 : process
    begin
        loop
            lfsr0 <= (lfsr0(n-2) xor lfsr0(n-1)) & lfsr0(0 to n-2);
            lfsr1 <= (lfsr1(n-3) xor lfsr1(n-1)) & lfsr1(0 to n-2);
            wait for period;
        end loop;
    end process;

```

```

    sign_add_sub0 : sign_add_sub
        generic map(n => 4)
        port map(a => lfsr0,
            b => lfsr1,
            cin => '1',
            sum => sum,
            sub => sub,
            cout_sub => cout_sub,
            cout_sum => cout_sum);

```

```

end beh;

```