



Xilinx Synthesis Technology (XST) User Guide
Chapter 2: HDL Coding Techniques

Counters

XST is able to recognize counters with the following controls signals:

- Asynchronous Set/Clear
- Synchronous Set/Clear
- Asynchronous/Synchronous Load (signal and/or constant)
- Clock Enable
- Modes (Up, Down, Up/Down)
- Mixture of all mentioned above possibilities

HDL coding styles for the following control signals are equivalent to the ones described in the ["Registers" section](#) of this chapter:

- Clock
- Asynchronous Set/Clear
- Synchronous Set/Clear
- Clock Enable

Moreover, XST supports unsigned as well as signed counters.

Log File

The XST log file reports the type and size of recognized counters during the macro recognition step:

```
...
Synthesizing Unit <counter>.
  Extracting 4-bit up counter for signal <tmp>.
  Summary:
    inferred 1 Counter(s).
Unit <counter> synthesized.
...
=====
HDL Synthesis Report

Macro Statistics
# Counters : 1
  4-bit up counter: 1
=====
...
```

4-bit Unsigned Up Counter with Asynchronous Clear

IO Pins	Description
C	Positive-Edge Clock
CLR	Asynchronous Clear (active High)
Q[3:0]	Data Output

VHDL Code

Following is VHDL code for a 4-bit unsigned Up counter with asynchronous clear.

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

entity counter is
    port(C, CLR : in std_logic;
          Q : out std_logic_vector(3 downto 0));
end counter;
architecture archi of counter is
    signal tmp: std_logic_vector(3 downto 0);
    begin
        process (C, CLR)
        begin
            if (CLR='1') then
                tmp <= "0000";
            elsif (C'event and C='1') then
                tmp <= tmp + 1;
            end if;
        end process;
        Q <= tmp;
    end archi;
```

Verilog Code

Following is the Verilog code for a 4-bit unsigned Up counter with asynchronous clear.

```
module counter (C, CLR, Q);
input C, CLR;
output [3:0] Q;
reg [3:0] tmp;

    always @(posedge C or posedge CLR)
    begin
        if (CLR)
            tmp = 4'b0000;
        else
            tmp = tmp + 1'b1;
        end
        assign Q = tmp;
    endmodule
```

4-bit Unsigned Down Counter with Synchronous Set

IO Pins	Description
C	Positive-Edge Clock
S	Synchronous Set (active High)
Q[3:0]	Data Output

VHDL Code

Following is the VHDL code for a 4-bit unsigned Down counter with synchronous set.

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

entity counter is
    port(C, S : in std_logic;
         Q : out std_logic_vector(3 downto 0));
end counter;
architecture archi of counter is
    signal tmp: std_logic_vector(3 downto 0);
begin
    process (C)
    begin
        if (C'event and C='1') then
            if (S='1') then
                tmp <= "1111";
            else
                tmp <= tmp - 1;
            end if;
        end if;
    end process;
    Q <= tmp;
end archi;
```

Verilog Code

Following is the Verilog code for a 4-bit unsigned Down counter with synchronous set.

```
module counter (C, S, Q);
input C, S;
output [3:0] Q;
reg [3:0] tmp;

    always @(posedge C)
    begin
        if (S)
            tmp = 4'b1111;
        else
            tmp = tmp - 1'b1;
        end
        assign Q = tmp;
    endmodule
```

4-bit Unsigned Up Counter with Asynchronous Load

from Primary Input

IO Pins	Description
C	Positive-Edge Clock
ALOAD	Asynchronous Load (active High)
D[3:0]	Data Input
Q[3:0]	Data Output

VHDL Code

Following is the VHDL code for a 4-bit unsigned Up Counter with asynchronous load from primary input.

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

entity counter is
    port(C, ALOAD : in std_logic;
          D : in std_logic_vector(3 downto 0);
          Q : out std_logic_vector(3 downto 0));
end counter;
architecture archi of counter is
    signal tmp: std_logic_vector(3 downto 0);
    begin
        process (C, ALOAD, D)
        begin
            if (ALOAD='1') then
                tmp <= D;
            elsif (C'event and C='1') then
                tmp <= tmp + 1;
            end if;
        end process;
        Q <= tmp;
    end archi;
```

Verilog Code

Following is the Verilog code for a 4-bit unsigned Up Counter with asynchronous load from primary input.

```
module counter (C, ALOAD, D, Q);
input C, ALOAD;
input [3:0] D;
output [3:0] Q;
reg [3:0] tmp;

always @(posedge C or posedge ALOAD)
begin
    if (ALOAD)
        tmp = D;
    else
        tmp = tmp + 1'b1;
end
```

```

        end
        assign Q = tmp;
    endmodule

```

4-bit Unsigned Up Counter with Synchronous Load with a Constant

IO Pins	Description
C	Positive-Edge Clock
SLOAD	Synchronous Load (active High)
Q[3:0]	Data Output

VHDL Code

Following is the VHDL code for a 4-bit unsigned Up Counter with synchronous load with a constant.

```

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

entity counter is
    port(C, SLOAD : in  std_logic;
          Q : out std_logic_vector(3 downto 0));
end counter;
architecture archi of counter is
    signal tmp: std_logic_vector(3 downto 0);
    begin
        process (C)
        begin
            if (C'event and C='1') then
                if (SLOAD='1') then
                    tmp <= "1010";
                else
                    tmp <= tmp + 1;
                end if;
            end if;
        end process;
        Q <= tmp;
    end archi;

```

Verilog Code

Following is the Verilog code for a 4-bit unsigned Up Counter with synchronous load with a constant.

```

module counter (C, SLOAD, Q);
input C, SLOAD;
output [3:0] Q;
reg [3:0] tmp;

    always @(posedge C)
        begin

```

```

        if (SLOAD)
            tmp = 4'b1010;
        else
            tmp = tmp + 1'b1;
        end
        assign Q = tmp;
    endmodule

```

4-bit Unsigned Up Counter with Asynchronous Clear and Clock Enable

IO Pins	Description
C	Positive-Edge Clock
CLR	Asynchronous Clear (active High)
CE	Clock Enable
Q[3:0]	Data Output

VHDL Code

Following is the VHDL code for a 4-bit unsigned Up counter with asynchronous clear and clock enable.

```

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

entity counter is
    port(C, CLR, CE : in std_logic;
         Q : out std_logic_vector(3 downto 0));
end counter;
architecture archi of counter is
    signal tmp: std_logic_vector(3 downto 0);
    begin
        process (C, CLR)
        begin
            if (CLR='1') then
                tmp <= "0000";
            elsif (C'event and C='1') then
                if (CE='1') then
                    tmp <= tmp + 1;
                end if;
            end if;
        end process;
        Q <= tmp;
    end archi;

```

Verilog Code

Following is the Verilog code for a 4-bit unsigned Up counter with asynchronous clear and clock enable.

```

module counter (C, CLR, CE, Q);
input C, CLR, CE;
output [3:0] Q;
reg      [3:0] tmp;

```

```

always @(posedge C or posedge CLR)
begin
    if (CLR)
        tmp = 4'b0000;
    else
        if (CE)
            tmp = tmp + 1'b1;
        end
    assign Q = tmp;
endmodule

```

4-bit Unsigned Up/Down counter with Asynchronous Clear

IO Pins	Description
C	Positive-Edge Clock
CLR	Asynchronous Clear (active High)
up_down	up/down count mode selector
Q[3:0]	Data Output

VHDL Code

Following is the VHDL code for a 4-bit unsigned Up/Down counter with asynchronous clear.

```

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

entity counter is
    port(C, CLR, up_down : in std_logic;
         Q : out std_logic_vector(3 downto 0));
end counter;

architecture archi of counter is
    signal tmp: std_logic_vector(3 downto 0);
begin
    process (C, CLR)
    begin
        if (CLR='1') then
            tmp <= "0000";
        elsif (C'event and C='1') then
            if (up_down='1') then
                tmp <= tmp + 1;
            else
                tmp <= tmp - 1;
            end if;
        end if;
    end process;
    Q <= tmp;
end archi;

```

Verilog Code

Following is the Verilog code for a 4-bit unsigned Up/Down counter with asynchronous clear.

```
module counter (C, CLR, up_down, Q);
input C, CLR, up_down;
output [3:0] Q;
reg      [3:0] tmp;

    always @(posedge C or posedge CLR)
    begin
        if (CLR)
            tmp = 4'b0000;
        else
            if (up_down)
                tmp = tmp + 1'b1;
            else
                tmp = tmp - 1'b1;
        end
        assign Q = tmp;
    endmodule
```

4-bit Signed Up Counter with Asynchronous Reset

IO Pins	Description
C	Positive-Edge Clock
CLR	Asynchronous Clear (active High)
Q[3:0]	Data Output

VHDL Code

Following is the VHDL code for a 4-bit signed Up counter with asynchronous reset.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;

entity counter is
    port(C, CLR : in  std_logic;
         Q : out std_logic_vector(3 downto 0));
end counter;
architecture archi of counter is
    signal tmp: std_logic_vector(3 downto 0);
    begin
        process (C, CLR)
        begin
            if (CLR='1') then
                tmp <= "0000";
            elsif (C'event and C='1')    then
                tmp <= tmp + 1;
            end if;
        end process;
        Q <= tmp;
    end archi;
```

Verilog Code

There is no equivalent Verilog code.

No constraints are available.