Comp 331 – Introduction to Hardware

Counters
A Divide by 8 Counter using D-flipflops
Arbitrary Waveforms
3-Bit Counters
Mod 5 Counter
N-Bit Counter
Clock Divider: Modulo 10K Counter

State Diagram for 3-bit divide-by-8 counter

Truth Tables, Logic Equations ....

<table>
<thead>
<tr>
<th>State</th>
<th>q2 q1 q0</th>
<th>D2 D1 D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>s0</td>
<td>0 0 0 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>s1</td>
<td>0 0 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>s2</td>
<td>0 1 0 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>s3</td>
<td>0 1 1 0</td>
<td>1 1 0</td>
</tr>
<tr>
<td>s4</td>
<td>1 0 0 0</td>
<td>1 0 1</td>
</tr>
<tr>
<td>s5</td>
<td>1 0 1 0</td>
<td>1 1 1</td>
</tr>
<tr>
<td>s6</td>
<td>1 1 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>s7</td>
<td>1 1 1 0</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>

D0 <= not q0
D1 <= (not q0 and q1) or (not q1 and q0)
D2 <= (q0 and q1 and not q2) or (not q1 and q2) or (not q0 and q2)

Example 49: 3-Bit Counter

Entity declaration

library IEEE;
use IEEE.STD_LOGIC_1164.all;
-- use IEEE.STD_LOGIC_unsigned.all;
entity count3a is
port( clr : in STD_LOGIC;
     clk : in STD_LOGIC;
     q : out STD_LOGIC_VECTOR(2 downto 0));
end count3a;

architecture - implementation

architecture count3a of count3a is
signal D, qs: STD_LOGIC_VECTOR(2 downto 0);
begin
D(2) <= (not qs(2) and qs(1) and qs(0)) or (qs(2) and not qs(1)) or (qs(2) and not qs(0));
D(1) <= (not qs(1) and qs(0)) or (qs(1) and not qs(0));
D(0) <= not qs(0);
process(clk, clr) is three D flip-flops
begin
if clr = '1' then
qs <= "000";
elsif clk'event and clk = '1' then
qs <= D;
end if;
end process;
qu <= qs;
end count3a;
3-Bit Behavioral Counter in VHDL

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_unsigned.all;
architecture count3b of count3b is
signal count: STD_LOGIC_VECTOR(2 downto 0);
begin
process(clk, clr) begin
if clr = '1' then
    count <= "000";
elsif clk'event and clk = '1' then
    count <= count +1;
end if;
end process;
q <= count;
end count3b;
```

N-Bit Counter (Generic)

```vhdl
entity counter is
  generic(N : integer := 8);
  port(clr : in STD_LOGIC;
    clk : in STD_LOGIC;
    q : out STD_LOGIC_VECTOR(N-1 downto 0));
end counter;
```

Counter Variants

**Modulo-5 Counter**

```vhdl
begin
if clr = '1' then
    count <= "000";
elsif clk'event and clk = '1' then
    if count = "100" then -- limit
        count <= "000";
    else
        count <= count +1;
    end if;
end if;
end process;
```

**N-Bit Counter (Generic)**

```vhdl
architecture counter of counter is
signal count: STD_LOGIC_VECTOR(N-1 downto 0);
begin
process(clk, clr)
begin
if clr = '1' then
    count <= (others => '0');
elsif clk'event and clk = '1' then
    count <= count + 1;
end if;
end process;
q <= count;
end counter;
```

**Modulo-10K Counter**

```vhdl
architecture mod10kcnt of mod10kcnt is
signal count: STD_LOGIC_VECTOR(13 downto 0);
begin
process(clk, clr)
begin
if clr = '1' then
    count <= (others => '0');
elsif clk'event and clk = '1' then
    if conv_integer(count) = 9999 then
        count <= (others => '0');
    else
        count <= count +1;
    end if;
end if;
end process;
q <= count;
end counter;
```

from `Example 52`

**Clock Divider**

```vhdl
entity clkdiv is
  port(mclk : in STD_LOGIC;
    clr : in STD_LOGIC;
    clk25 : out STD_LOGIC;
    clk190 : out STD_LOGIC;
    clk3 : out STD_LOGIC);
end clkdiv;
```

**For**

```vhdl
architecture clkdiv of clkdiv is
signal q: STD_LOGIC_VECTOR(23 downto 0);
begin
process(mclk, clr)
begin
if clr = '1' then
    q <= X"000000";
elsif mclk'event and mclk = '1' then
    q <= q + 1;
end if;
end process;
clk25 <= q(0);    -- 25 MHz
clk190 <= q(17);  -- 190 Hz
clk3 <= q(23);    -- 3 Hz
end clkdiv;
```

**Exercise:** Implement Example 52: ___________

Block Diagram of Mod 10000 Counter

```
```

Note: Use clk3 as there is no clk48