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

entity count is
  port (
    D           : in  std_logic_vector(3 downto 0);
    Q           : out std_logic_vector(3 downto 0);
    count, load : in  std_logic;
    clock       : in  std_logic;
    carry       : out std_logic := '0');
end count;

architecture behav of count is
  signal Q_int : std_logic_vector(0 to 3);
  signal here : std_logic := '0';
    -- added for testing
begin
  count_process: process (clock)
  begin
    if rising_edge(clock) and load='1' then
      Q_int <= D;
      carry <= '0';
```

```
    elsif rising_edge(clock) and count=’1’ then
      -- for testing
      here <= ’1’, ’0’ after 5 ns;

      if Q_int = "1111" then
        carry <= ’1’;
      else
        carry <= ’0’;
      end if;

      Q_int <= Q_int + "0001";
    end if;
  end process;

  -- Update Q whenever Q_int changes:
  Q_update: process (Q_int)
  begin
    Q <= Q_int after 3 ns;
  end process Q_update;

end behav;
```