

After changing the Q_INT variable into signal...

```
begin
  process (CLK)
    -- internal variable Q_INT
    variable Q_INT: STD_LOG_VECT(2 downto 0);
  begin
    if CLK'event and CLK = '1' then
      if CLR = '1' then
        Q_INT := "000";
      else
        Q_INT := Q_INT+1;
      end if;
    end if;
    Q <= Q_INT;
  end process;
```

```
-- internal signal Q_INT
signal Q_INT: STD_LOG_VECT(2 downto 0);
begin
  process (CLK)
  begin
    if CLK'event and CLK = '1' then
      if CLR = '1' then
        Q_INT <= "000";
      else
        Q_INT <= Q_INT+1;
      end if;
    end if;
    Q <= Q_INT;
  end process;
```

... output changes on the falling CLK Edge... ☹

Changing the active edge detection to falling...?

```
-- internal signal Q_INT
signal Q_INT: STD_LOG_VECT(2 downto 0);
begin
process (CLK)
begin
  if CLK'event and CLK = '1' then
    if CLR = '1' then
      Q_INT <= "000";
    else
      Q_INT <= Q_INT+1;
    end if;
  end if;
  Q <= Q_INT;
end process;
```

```
-- internal signal Q_INT
signal Q_INT: STD_LOG_VECT(2 downto 0);
begin
process (CLK)
begin
  if CLK'event and CLK = '0' then
    if CLR = '1' then
      Q_INT <= "000";
    else
      Q_INT <= Q_INT+1;
    end if;
  end if;
  Q <= Q_INT;
end process;
```

**Bad because it masks the intentions of code-writer.
The code is not self-documented anymore.**

Moving Q assignment into CLK edge detection condition...?

```

begin
  process (CLK)
    -- internal variable Q_INT
    variable Q_INT: STD_LOG_VECT(2 downto 0);
  begin
    if CLK'event and CLK = '1' then
      if CLR = '1' then
        Q_INT := "000";
      else
        Q_INT := Q_INT+1;
      end if;
    end if;
    Q <= Q_INT;
  end process;

```

```

-- internal signal Q_INT
signal Q_INT: STD_LOG_VECT(2 downto 0);
begin
  process (CLK)
  begin
    if CLK'event and CLK = '1' then
      if CLR = '1' then
        Q_INT <= "000";
      else
        Q_INT <= Q_INT+1;
      end if;
      Q <= Q_INT;
    end if;
  end process;

```

Bad because the changes are now delayed by full clock cycle.

Extending of the sensitivity list ...?

```
begin
  process (CLK)
    -- internal variable Q_INT
    variable Q_INT: STD_LOG_VECT(2 downto 0);
  begin
    if CLK'event and CLK = '1' then
      if CLR = '1' then
        Q_INT := "000";
      else
        Q_INT := Q_INT+1;
      end if;
    end if;
    Q <= Q_INT;
  end process;
```

```
-- internal signal Q_INT
signal Q_INT: STD_LOG_VECT(2 downto 0);
begin
  process (CLK, Q_INT)
  begin
    if CLK'event and CLK = '1' then
      if CLR = '1' then
        Q_INT <= "000";
      else
        Q_INT <= Q_INT+1;
      end if;
    end if;
    Q <= Q_INT;
  end process;
```

Better, but what if there are many Q-like outputs?

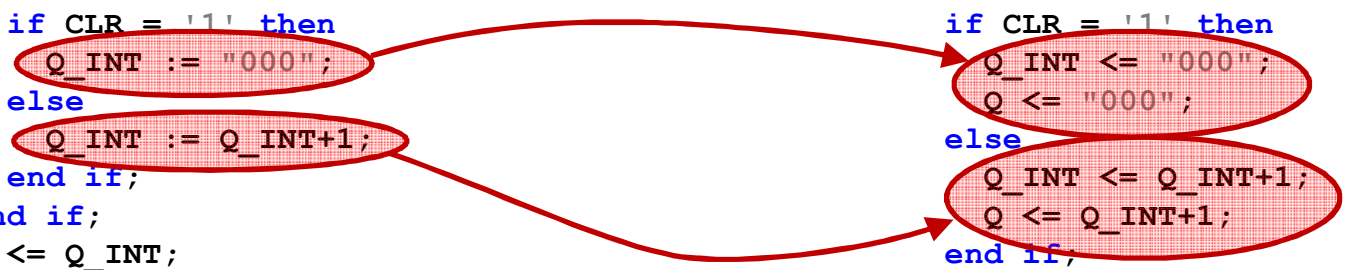
```
process (CLK, Q_INT, D_INT, L_INT, R_INT, ...)
```

Adding mirror assignments... ?

```

begin
  process (CLK)
    -- internal variable Q_INT
    variable Q_INT: STD_LOG_VECT(2 downto 0);
  begin
    if CLK'event and CLK = '1' then
      if CLR = '1' then
        Q_INT := "000";
      else
        Q_INT := Q_INT+1;
      end if;
    end if;
    Q <= Q_INT;
  end process;

```



```

-- internal signal Q_INT
signal Q_INT: STD_LOG_VECT(2 downto 0);
begin
  process (CLK)
  begin
    if CLK'event and CLK = '1' then
      if CLR = '1' then
        Q_INT <= "000";
        Q <= "000";
      else
        Q_INT <= Q_INT+1;
        Q <= Q_INT+1;
      end if;
    end if;
    Q <= Q_INT;
  end process;

```

... formally correct and synthesable, but too complicated.

Changing the Q assignment from procedural to concurrent ...?

```

begin
  process (CLK)
    -- internal variable Q_INT
    variable Q_INT: STD_LOG_VECT(2 downto 0);
  begin
    if CLK'event and CLK = '1' then
      if CLR = '1' then
        Q_INT := "000";
      else
        Q_INT := Q_INT+1;
      end if;
    end if;
    Q <= Q_INT;
  end process;

```

```

-- internal signal Q_INT
signal Q_INT: STD_LOG_VECT(2 downto 0);
begin
  process (CLK)
  begin
    if CLK'event and CLK = '1' then
      if CLR = '1' then
        Q_INT <= "000";
      else
        Q_INT <= Q_INT+1;
      end if;
    end if;
  end process;
  Q <= Q_INT;

```

YES! This is the right solution!