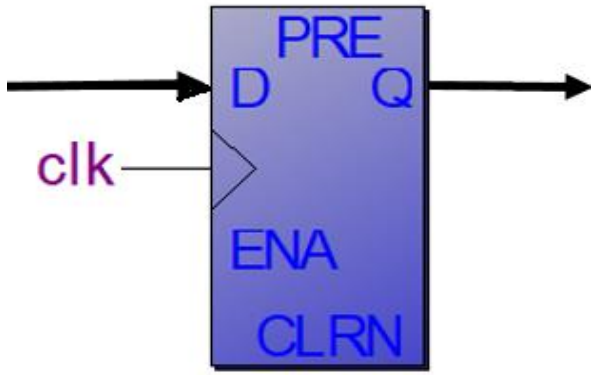
VHDL
Using both clk transitions

Synchronous building block



We can use only one from the next two clk's conditions per PROCESS:

- ✓ Positive edge trigger:

```
IF (clk'EVENT AND clk='1') THEN
    sequential_statements;
END IF;
```

```
IF (rising_edge(clk)) THEN
    sequential_statements;
END IF;
```

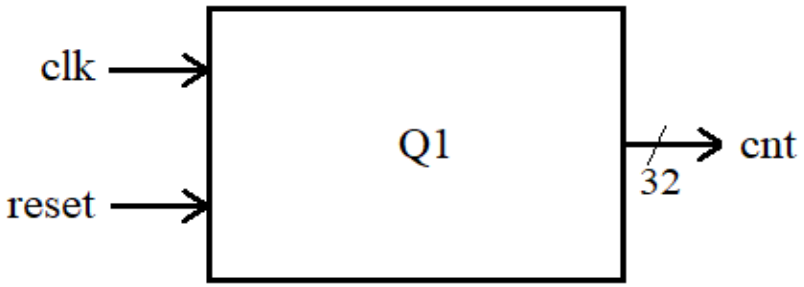
- ✓ Negative edge trigger:

```
IF (clk'EVENT AND clk='0') THEN
    sequential_statements;
END IF;
```

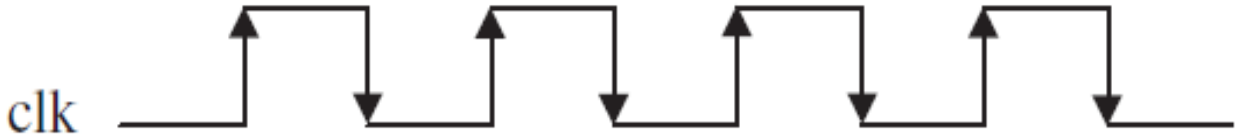
```
IF (falling_edge(clk)) THEN
    sequential_statements;
END IF;
```

Question:
How to design the given system ?

Design requirement: counter on both clk transitions.




```
entity clk_cnt is
   port(
        clk : in std_logic;
      reset : in std_logic;
        cnt : out std_logic_vector (32 downto 0));
end clk_cnt;
```



Wrong solution No.1

```
architecture arc of sol1 is
    signal counter : std_logic_vector (32 downto 0);
begin
    process(clk)
    begin
        if (reset='1') then
            counter <= (others => '0');
        elsif (clk'event and clk='1') then
            counter <= counter + 1;
        elsif (clk'event and clk='0') then
            counter <= counter + 1;
        end if;
    end process;
    cnt <= counter;
end arc;
```

The ModelSim simulation works well but the code is not synthesizable because:

- It contains signal assignments at both transitions of the reference clk signal (the target technology contains only single edge FF).

Wrong solution No.2

```
architecture arc of sol2 is
    signal counter : std_logic_vector (32 downto 0);
begin
    process(clk)
    begin
        if (reset='1') then
            counter <= (others => '0');
        elsif (clk'event) then
            counter <= counter + 1;
        end if;
   end process;
    cnt <= counter;
end arc;
```

The ModelSim simulation works well but the code is not synthesizable because :

The attribute EVENT must be related to a test condition (transition detector).

- if (clk'event and clk='1') – is correct , if (clk'event) then – is incorrect

Wrong solution No.3

```
architecture arc of sol3 is
    signal counter : std_logic_vector (32 downto 0);
begin
    process(clk)
    begin
        if (reset='1') then
            counter <= (others => '0');
        end if;
        counter <= counter + 1;
    end process;
    cnt <= counter;
end arc;
```

The ModelSim simulation doesn't work well because, if a signal appears in the sensitivity list but doesn't appear in any of the PROCESS's assignments the compiler ignores it (in this case only clk signal appears in the sensitivity list and it is unused so the compiler ignores the whole process).

Correct Solution - code

```
architecture arc of sol4 is
    signal          x,y : std_logic_vector (32 downto 0);
begin
    process(clk)
    begin
        if (reset='1') then
            x <= (others => '0');
        elsif (clk'event and clk='1') then
            x <= x + 1;
        end if;
    end process;

    process(clk)
    begin
        if (reset='1') then
            y <= (others => '0');
        elsif (clk'event and clk='0') then
            y <= y + 1;
        end if;
    end process;

    cnt <= x + y;
end arc;
```

Correct Solution - synthesis