Lecutre #6
May 19

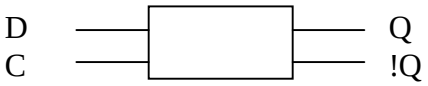
```
If s='1' then
        Elseif r='1' then
End if:
```

Behavioral descriptions

- we have to be sure to capture all of the behavior
- if both S and R are 1 the behavior is unknown
- we can capture this in VDHL:
  - if s= '1' and r = '1' then
    - q<= 'X';
    - z_b <= 'X';
  - elseif s= '1' then

```
…
elseif r='1' then
….
endif;
```

D latch"



- When C= 1, copy D into the flip flop (Q<=D)
- When C=0 no change
- A D flip flop
  - same except D can only change on the rising edge of C

```
entity pet_d is port (
        d, c: in std_logic;
        q,q_b out std_logic);
end pet_d;
architecture vehav of pet_d is
begin
        d_process: process (c)
        begin
                if rising_edge (c) then
```

```
                q<= d;
            end ifl
        end process;
        q_b <= not q;
        end behave;

        -       rising_edge is built into the IEEE std_logic library
    -           the statement
                q_b <= not q;
```

works like this:

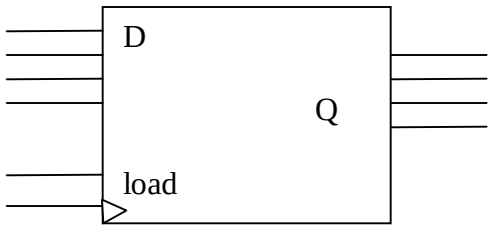
```
        process (q)
            q_b<= not q;
        end process;
```

A register in VHDL

- arrays: a group of several signals or variables
- generics: allow you to define parameters for each instance
  - we can use the same code for an 8, 16, 32, bit register

```
entity reg is
        generic(
            n: integer);
        port(
            D: in std_logic_vector( n -1 down to 0);
            Q: out std_logic vector ( n-1 down to 0);
            Clock load: in std_logic);
End reg;
```

If we instantiate with n=4, we get this:



```
Architecture behave of reg is
Begin
        Load_process: process (clock)
        Begin
                If rising_edge (clock)
                        And load = '1' then
                                Q<=D;
                End if;
        End process;
End beahv;
```

- we can also copy D to Q bit-by-bit:

```
        f… then
                        for index in n-1 down to 0
                                Q(index) <= D(index);
                        End loop;
                End if;
```