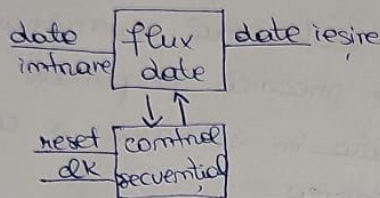


## Limbaaje de descriere hardware

08.01.2025

### Curs 11



Semnale bidirectionale

si buffer three state

Sunt situati in care modulele digitale sunt conectate la magistrale comune de date.

În acest caz trebuie creată o logică prin care porturile comune trebuie deconectate de la magistralele de date în vederea eliberării acestora pentru alte module care doresc să se conecteze.

Conectarea se face prin portul de tip three state. Porturi care trec în imalta impedanță atunci când sunt deconectate.



ex/ Să se descrie un numărator pe 8 biți care prezintă următoarele porturi:  
 se - când e inactiv portul de ieșire data-out va fi trecut în alta ipedanță.  
 em - când e activ se realizează numărarea  
 load - când e activ se încarcă pe portul de intrare data-in valoarea pe care începe numărarea.

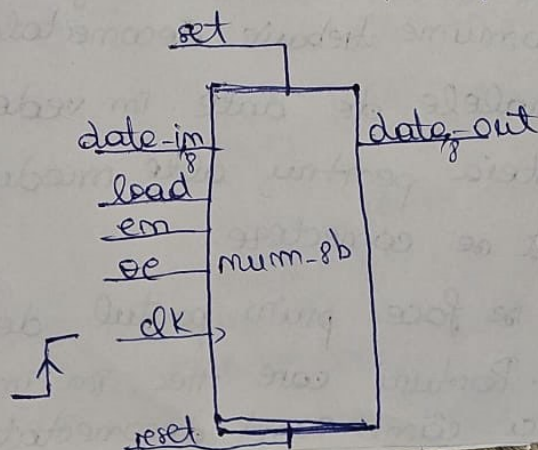
set - se trece portul de ieșire în valoarea FFF.

reset - în val. 00H

clk - semnalul de ceas după care se face sincronizarea.

data-in - port de intrare pe 8 biți pe care se plasează val. inițială de la care pornește numărarea.

data-out - val. rezultată în urma numărării, 8 biți.



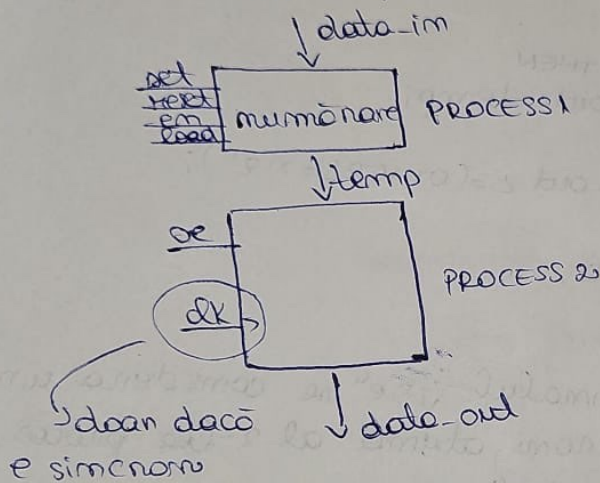
ce, reset - asincroniz



```

LIBRARY IEEE
USE IEEE.std_logic_1164.all;
ENTITY mumm_8b IS
    PORT (data_in: std_logic_vector(7 DOWNTO 0);
          set, load, em, oe, clk, reset: IN std_logic;
          data_out: OUT std_logic_vector(7 DOWNTO 0));
END mumm_8b;

```



ARCHITECTURE dsen OF mumm\_8b IS  
 temp: std\_logic\_vector(7 DOWNTO 0) := x"00";  
 BEGIN  
 PROCESS (clk, reset)  
 VARIABLE tm: std\_logic\_vector(7 DOWNTO 0) := x"00";  
 BEGIN  
 IF reset = '1' THEN  
 tm := x"00";  
 ELSE IF clk event AND clk = '1' THEN  
 IF set = '1' THEN  
 tm := x"FF";  
 ELSE IF load = '1' THEN  
 tm := data\_in;  
 END IF;  
 END IF;  
 END IF;  
 END PROCESS;  
 data\_out <= tm;  
END dsen;



```

ELSE IF em = '1' THEN
    tm := tm + 1;
END IF;
temp := tm;
END PROCESS;
PROCESS (oe, temp)
BEGIN
    IF oe = '1' THEN
        data_out := temp;
    ELSE
        data_out := (OTHERS => '2');
    END IF;
END PROCESS;
END dscr;

```

Dacă semnalul „oe” se consideră un semnal sincron, atunci al 2-lea proces va fi scris sub forma:

```

PROCESS (clk)
BEGIN
    IF (clk event AND clk = '1') THEN
        IF oe = '1' THEN
            data_out := temp;
        ELSE
            data_out := (OTHERS => '2');
        END IF;
    END IF;
END PROCESS;

```