5. CONVENȚII DE PROIECTARE

5.1 INTRODUCERE

După cum s-a arătat într-un capitol anterior, un sistem numeric poate fi partiționat în:

- secțiunea de date (unitatea de execuție),
- secțiunea de comandă (unitatea de comandă).

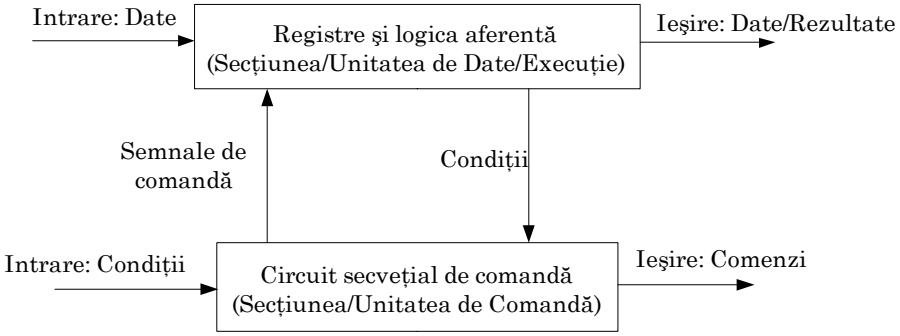


Figura 5.1. Partiționarea unui sistem numeric

Unitatea de execuție asigură prelucrarea datelor, reprezentate sub forma unor vectori binari, în cadrul transferului acestora între registrele sursă şi registrele destinație. Transferul se efectuează prin intermediul unor rețele/circuite logice combinaționale.

5.2 TRANSFERURILE ÎNTRE REGISTRE

Întrucât prelucrarea datelor are la bază transferul între registre, proiectarea unui sistem numeric, în vederea execuției unui algoritm, constă într-o planificare, ce va defini fiecare transfer, prin specificarea ordinii/temporizării (timing) în care aceste transferuri vor avea loc.

În continuare se prezintă mecanismele hardware de implementare a transferurilor între registre şi notațiile ad hoc corespunzătoare folosite. Registrele sunt notate cu majuscule având, de regulă, o semnificație mnemotehnica. Pe scurt, se vor folosi "mnemonice".

Transferul conținutului unui registru sursă într-un registru destinație, fără a afecta conținutul sursei, se notează astfel: $AC = RD$; Dacă registrul AC are patru biți, anularea/forțarea în unu a tuturor bistabililor se poate nota după cum urmează: $AC = 4'b\,0000$; $AC = 4'b\,1111$; Implementarea

registrelor se bazează pe bistabile JK şi D, cu intrări de sincronizare de ceas, CLK, şi cu intrări de tip $\overline{PRESET}$ / $\overline{CLR}$.

Transferurile sincrone au loc sub controlul semnalului de ceas pe fronturile anterior, pentru bistabilele de tip D, sau pe frontul posterior, pentru bistabilele master-slave, de tip JK. Transferurile implementate prin semnale de comandă aplicate pe intrările $\overline{PRESET}$ / $\overline{CLR}$ au, în general, caracter asincron. Mai jos se vor da, pentru cazul unui bistabil D controlat pe frontul anterior al semnalului de ceas, reprezentarea simbolică şi diagrama temporală de operare. Se observă că datele la intrarea D sunt stabile înaintea apariţiei frontului crescător al semnalului de ceas, când are loc forţarea datei în bistabil. Ieşirea va lua valoarea corespunzătoare intrării cu o anumită întârziere în raport cu frontul crescător al semnalului de ceas.

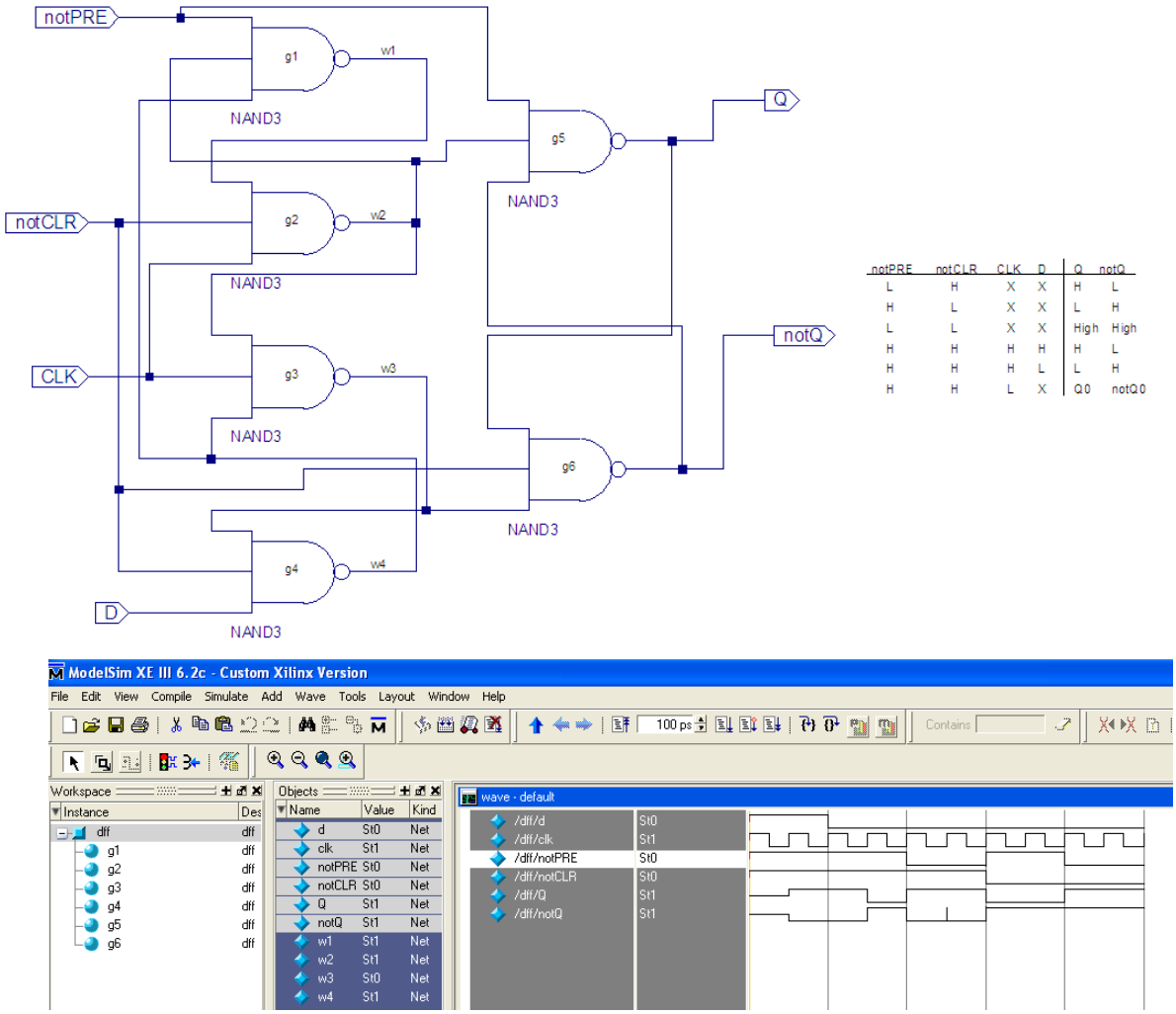


Figura 5.2. Bistabil D comandat pe frontal anterior al semnalului de ceas.

Codul Verilog structural pentru descrierea bistabilului D este prezentat mai jos:

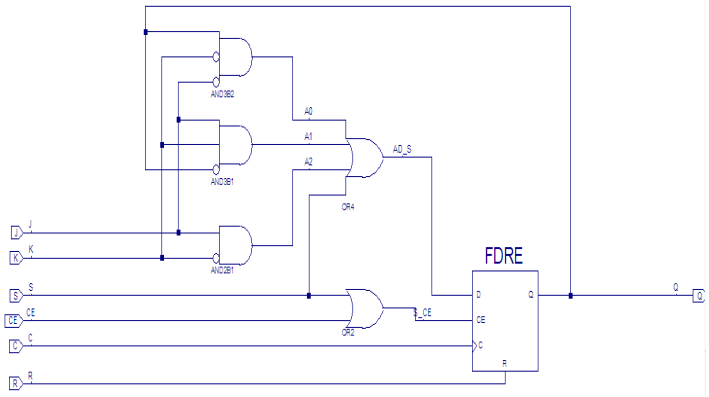
module dff(d, clk, notPRE, notCLR, Q, notQ);

input d, clk, notPRE, notCLR;
output Q, notQ;
wire w1, w2, w3, w4;

nand g1(w1, notPRE, w4, w2);
nand g2(w2, w1, notCLR, clk);
nand g3(w3, w2, clk, w4);
nand g4(w4, w3, notCLR, d);
nand g5(Q, notPRE, w2, notQ);
nand g6(notQ, Q, notCLR, w3);

endmodule

În cazul bistabilelor de tip master-slave forțarea datelor are loc pe frontul anterior al semnalului de ceas, iar apariția lor la ieşire se constată pe frontul posterior al ceasului. Mai jos se prezintă simbolul şi diagrama temporală pentru operarea bistabilului JK.



R	S	CE	J	K	C	Q
1	X	X	X	X	↑	0
0	1	X	X	X	↑	1
0	0	0	X	X	X	No Chg
0	0	1	0	0	X	No Chg
0	0	1	0	1	↑	0
0	0	1	1	1	↑	Toggle
0	0	1	1	0	↑	1

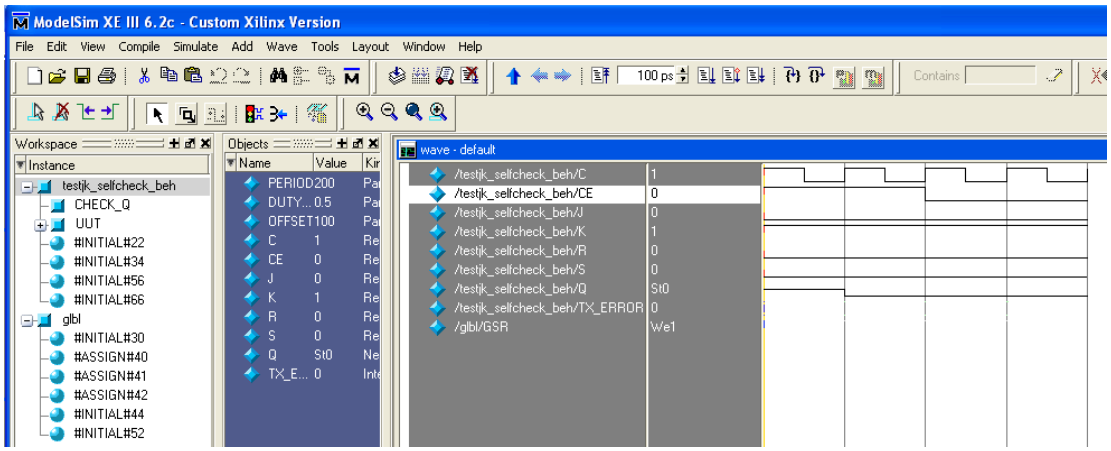


Figura 5.3. Bistabilul JK.

Codul Verilog este următorul:

```
always @(posedge C) begin
        if (R)
                Q <= 0;
        else if (S)
                Q <= 1;
        else if (CE)  begin
                if (!J) begin
                        if (K)
                                Q <= 0;
                end
                else  begin
                        if (!K)
                        Q <= 1;
                else
                        Q <= !Q;
                end
        end
end
```

În continuare se vor considera elemente bistabile de tip master-slave, atât pentru secțiunea de date, cât şi pentru secțiunea de comandă.

Un circuit secvențial de comandă furnizează, pentru secțiunea de date, diverse semnale de comandă, sincrone cu ceasul sistemului, cu perioade egale cu durata unei perioade de ceas sau cu multipli ai acesteia. Acestea sunt semnalele de comandă de tip nivel.

Semnalele de Comandă de tip Nivel (SCN) vor avea un sufix numeric *i* ce va specifica numărul ieşirii de comandă a automatului (SCN_i). Semnalele SCN_i pot fi strobate/eşantionate cu semnalul curent de ceas, pentru a forma Semnale de Comandă de tip Impuls (SCI_i), cu durata activă corespunzătoare semnalului curent de ceas.

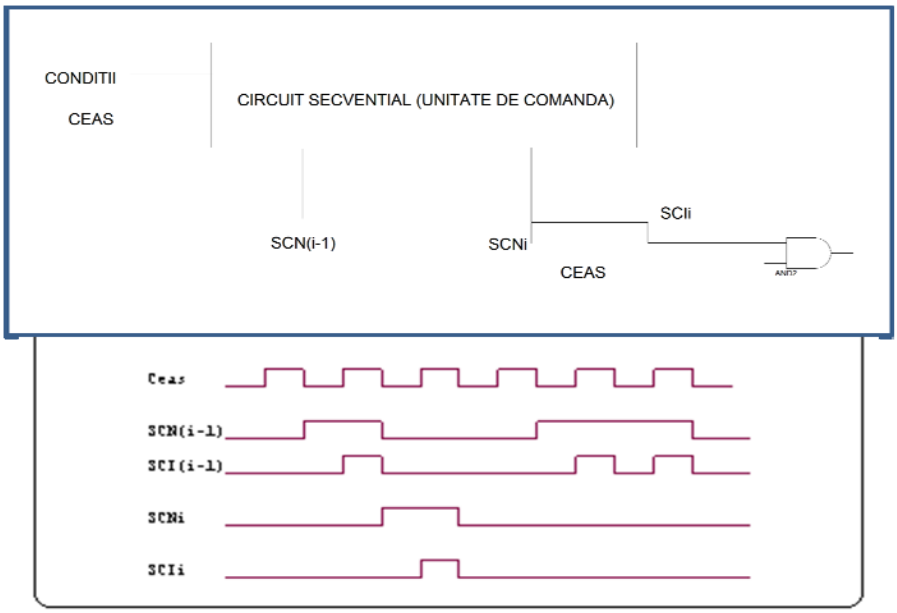


Figura 5.4. Circuit secvențial de comandă.

Să presupunem o situație în care în etapa *i* (pasul *i* al implementării/execuției) algoritmului trebuie să se efectueze transferul conținutului registrului de doi biți $RA[2]$, numerotați de la 0 la 1 ($RA_{0:1}$), în registrul destinație $RB[2]$ ($RB_{0:1}$):

i. $RB[0:1] = RA[0:1];$

Considerând, în continuare, că se vor folosi numai bistabile master-slave JK sau D (în unele cazuri se face presupunerea că şi bistabilele de tip D operează în maniera master-slave), schemele hardware pentru implementarea acestui transfer vor fi următoarele:

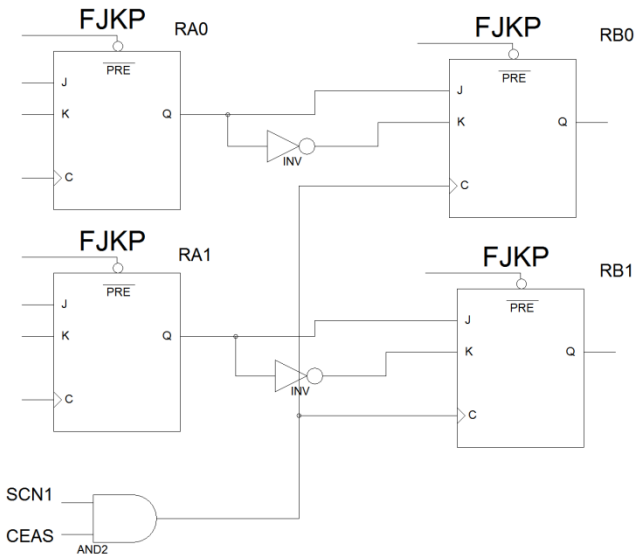


Figura 5.5. Transfer realizat cu bistabile de tip JK.

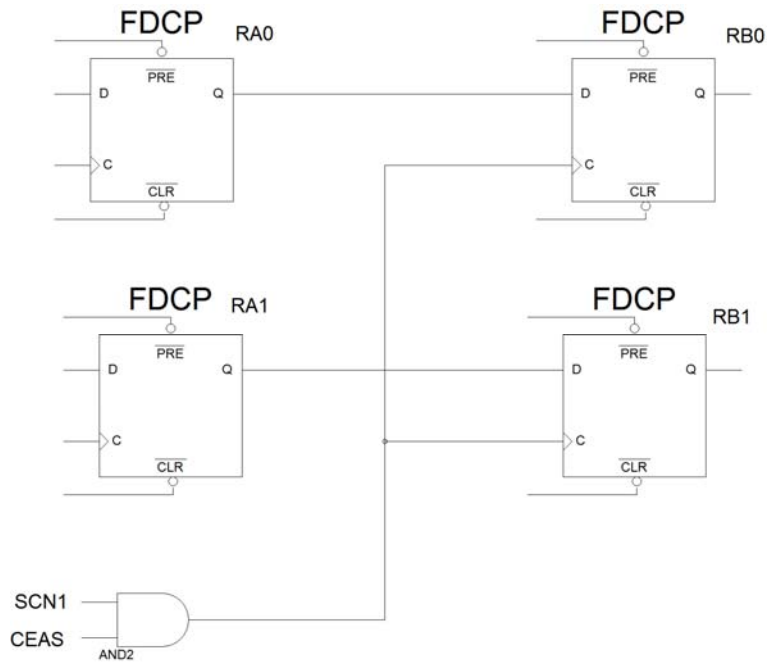


Figura 5.6. Transfer realizat cu bistabile de tip D.

Diagrama de timp care arată modul în care noua informație se transferă în RB este dată mai jos:

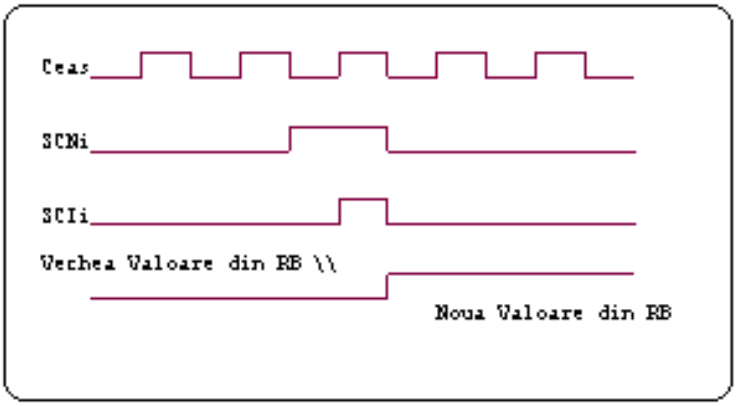


Figura 5.7. Diagramele de timp ale semnalelor implicate în transfer.

Adesea, într-un registru, trebuie să se transfere, în momente diferite, vectori diferiți, de la diverse surse. Fie registrele sursă: RA, RB şi registrul destinație RC. Fie, ca exemplu, următoarele transferuri pentru care se cere implementarea, la momentele 1, 2 :

1. $RC = RA$;
2. $RC = RB$;

sub controlul semnalelor SCN_1, SCN_2.

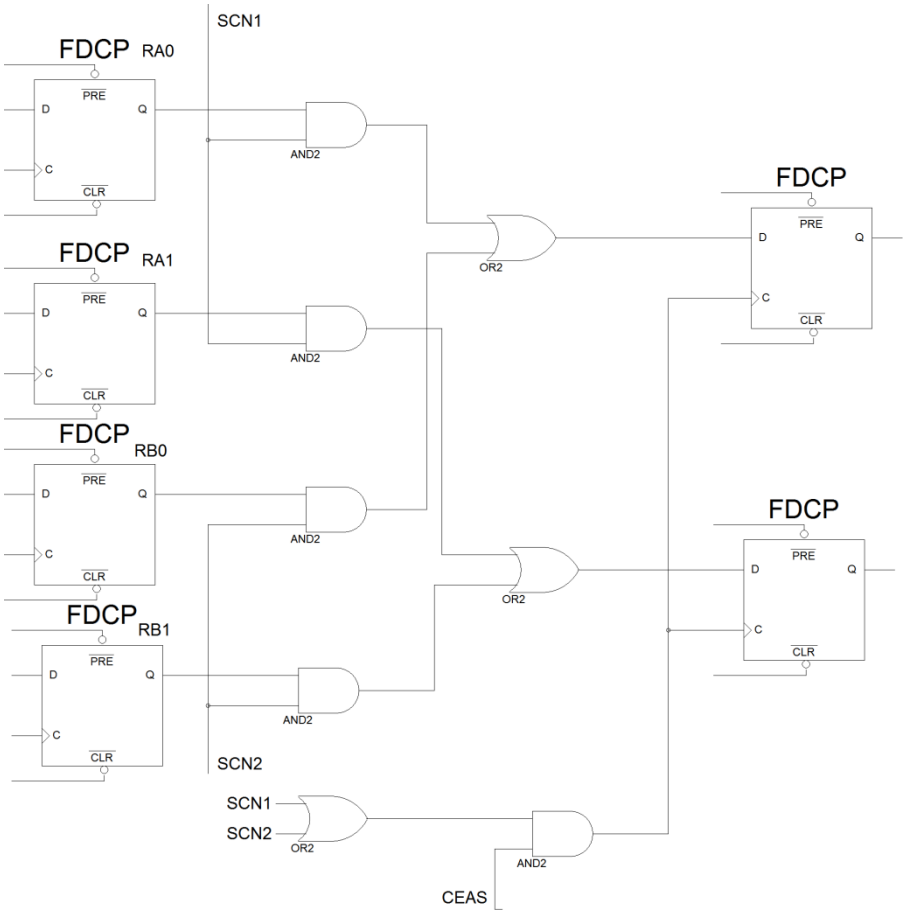


Figura 5.8. Transfer între registre prin intermediul unei rețele care simulează o magistrală.

Un alt exemplu de transfer între registre în care se cere implementarea hardware este analizat în cele ce urmează:

1. $RA = 2'b11;$ // $RA\ se\ incarca\ cu\ numarul\ binar\ 11$
2. $RA = 2'b00;$ // $RA\ se\ incarca\ cu\ numarul\ binar\ 00$
3. $RA = RB;$

Pentru a stabili expresia funcției logice combinaționale, care reprezintă intrarea D_i a bistabilului destinație RA_i, se utilizează diagrama Karnaugh, cu variabila RB_i introdusă în diagramă:

SCN_3 \ SCN_1, SCN_2	00	01	11	10
0	*	*	*	1
1	RB_i	*	*	*

Funcția de excitație D_i este dată de expresia: $D_i = SCN_1 \cup RB_i * SCN_3$

Aceasta va duce la următoarea implementare:

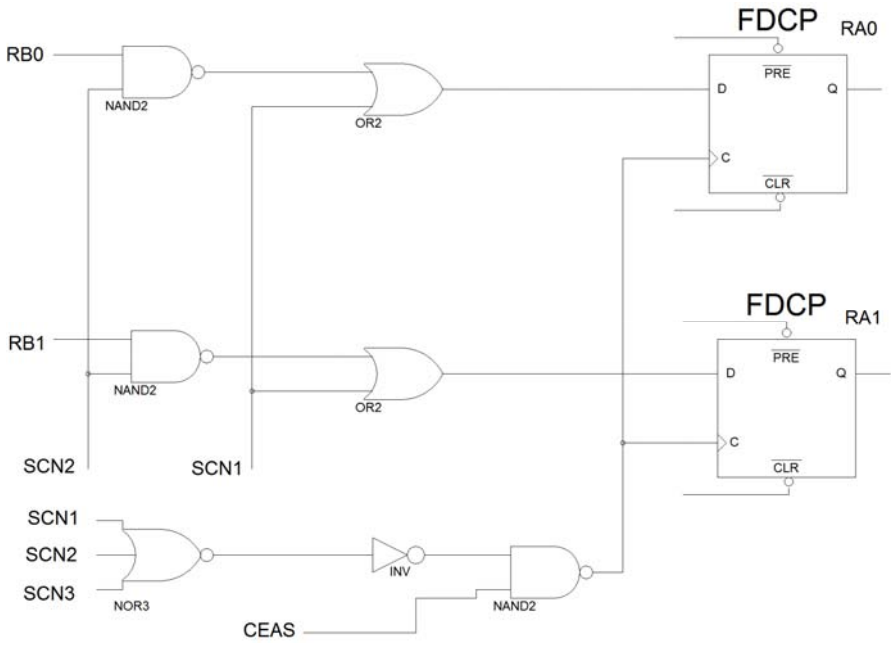


Figura 5.9. Implemenatrea secvenței de transferări

În cazul în care se urmăreşte implementarea operației elementare:

j. $RA[i] = RB[i] \ \ \& \ RC[i]$;

se poate folosi schema de mai jos:

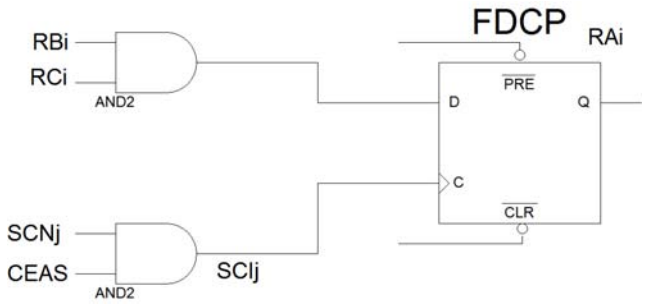


Figura 5.10. Transfer printr-o rețea logică combinațională.

Reprezentarea componentelor constructive logice combinaționale şi secvențiale (SSI/MSI).

În realizarea practică a unui sistem numeric de complexitate mică/medie (în afara cazurilor când se folosesc circuite programabile FPGA/FPLD) sunt utilizate, atât componente logice combinaționale, cât şi

secvențiale standard, integrate pe scară simplă şi medie. Mai jos se prezintă o modalitate de tratare unitară a acestor componente.

5.3 CONEXIUNI PRIN MAGISTRALE

În numeroase cazuri implementarea în hardware a unui algoritm de prelucrare a datelor impune, la diverse etape ale execuției sale, conectarea unor registre-sursă la anumite registre-destinație. Conexiunile directe oferă posibilitatea unui înalt grad de paralelism al transferurilor de date. În schimb sunt neeconomice sub aspectul consumului de circuite logice.

Pentru exemplificare se vor considera, pe de o parte două registre-sursă: RA şi RB, iar pe de altă parte doua registre-destinație: RC şi RD, între care se realizează următoarele transferuri:

1. $RC = RA;$
2. $RD = RB;$
3. $RC = RB;$
4. $RD = RA;$
5. $RC = RA; \quad RD = RB;$
6. $RC = RB; \quad RD = RA;$

Aceste transferuri se vor realiza sub controlul semnalelor de comandă de tip nivel: $SCN_1, SCN_2, SCN_3, SCN_4, SCN_5, SCN_6$ şi al semnalelor de comandă de tip impuls: $SCI_1, SCI_2, SCI_3, SCI_4, SCI_5, SCI_6$.

O ilustrare a implementării transferurilor de mai sus este prezentată în continuare, considerând, pentru simplificare, registre de câte un bit.

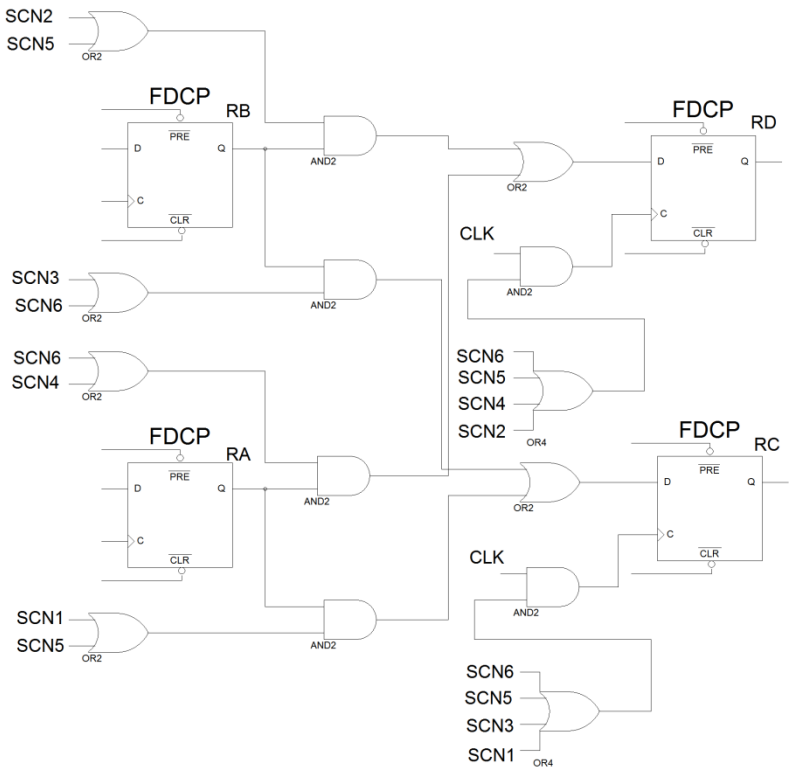


Figura 5.11. Conexiuni între registre sursă şi registre destinație folosind o magistrală.

Considerând ca sunt *s* registre-sursă, având n biți, şi d registre-destinație, de câte n biți se poate stabili o funcție de cost, pe baza numărului de porți ŞI, SAU şi a numărului de biți al fiecărui registru. Astfel:

- numărul de porți ŞI = $s * d * n$,
- numărul de porți SAU = $d * n$

Rezultă o funcție de cost $\approx n * d * (s+1)$

Pentru a reduce costul conexiunilor între surse şi destinații se utilizează magistralele. Acestea reprezintă, de regulă, trasee metalice/fire pe circuite imprimate sau la nivelul structurilor integrate pe siliciu. Modelul conexiunilor, prin intermediul magistralei, este ilustrat mai jos.

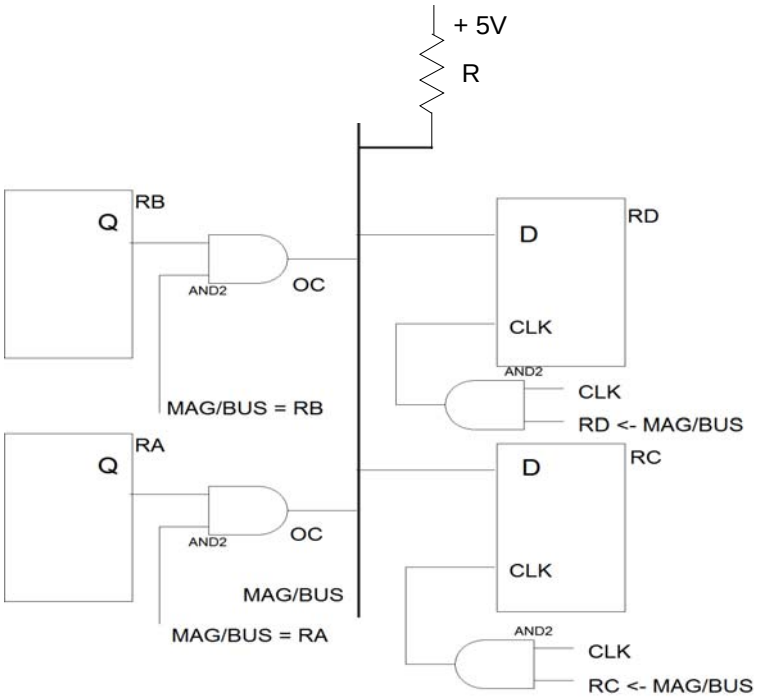


Figura 5.12. Magistrală ”open collector”.

Se poate observa că porțile ŞI, care conectează ieşirile registrelor RA şi RB, la magistrală sunt de tipul “colector deschis”- OC . Conexiunile Registrelor RA şi RB, la magistrala MAG/BUS se realizează sub controlul unor semnale de comandă de tip nivel, specificate în desen prin expresiile: $MAG/BUS = RA$ şi $MAG/BUS = RB$. Forțarea conținutului magistralei MAG/BUS în registrele RC şi RD are loc sub controlul semnalelor de comandă de tip impuls, specificate în desen prin expresiile: $RC = MAG/BUS;$ şi $RD = MAG/BUS;$.

Magistrala joacă rolul unui circuit SAU , conform figurii de mai jos.

Transferurile nu se mai pot efectua în paralel, de la surse la destinații. Ele vor avea un caracter secvențial. Astfel, transferul $RC = RA;$ se va efectua prin operațiile:

1. $MAG/BUS = RA;$;
2. $RC = MAG/BUS;$,

unde simbolul “=” reprezintă o conexiune, pe durata unui semnal de comandă de tip nivel.

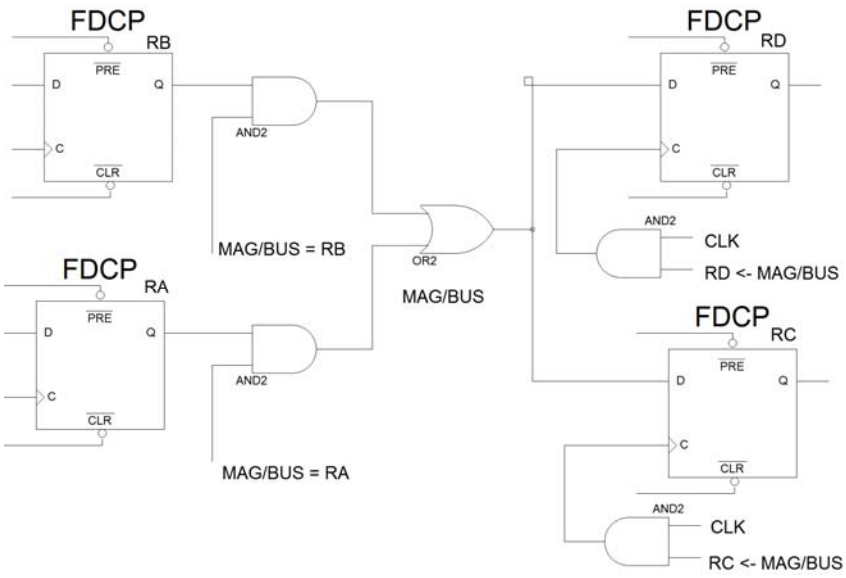


Figura 5.13. Exemplu de transfer prin magistrala MAG/BUS.

Funcția de cost va fi dată de numărul porților ŞI, al porților SAU, cât şi de numărul de biți ai vectorilor transferați:

- numărul de porți ŞI = $s * n$,
- numărul de porți SAU = n

Rezultă o funcție de cost $\approx n * (s-1)$

De cele mai multe ori conexiunile registrelor-sursă la magistrală se realizează prin circuite “Tri-state”, care înlocuiesc circuitele de tip “OC”.