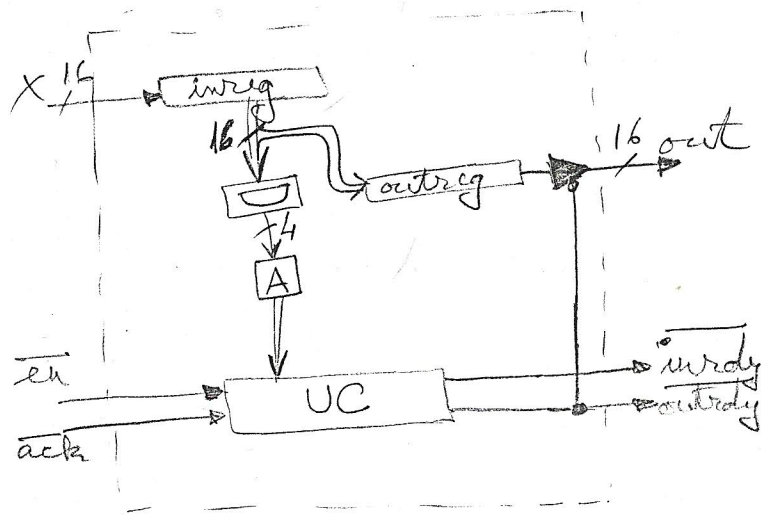
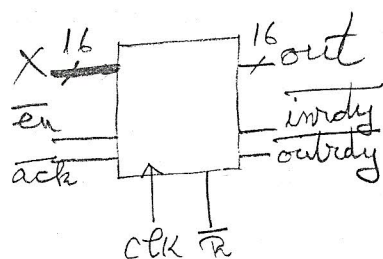


Se consideră un dispozitiv numeric pe 16 biți.
Acesta elimină cuvintele pt. care $X_{0:3} \wedge X_{4:7} \wedge X_{8:11} \wedge X'_{12:15} = 0000 (*)$.



Descriere

- În momentul în care este pregătit să primească un nou cuvânt, dispozitivul ridică linia "inrdy".
- Indiferent de valoarea intrării X , va aștepta activarea semnalului "en".
- După activare, X se va încărca în dispozitiv.
- După încărcare, se verifică relația (*) prin trecerea bitilor printr-un CLC și depunerea rezultatului în registrul A . Totodată, se transferă spre ieșire conținutul lui "inreg".
- UC verifică numărul de biți din A și ia decizia de a trimite sau nu conținutul lui "outreg" pe magistrală.
- Dacă acesta trebuie trimis spre magistrală, se activează semnalul "outrdy" și se așteaptă activarea lui "ack" pt a trece la un nou cuvânt.

MODULE disp

12

MEMORY: $\text{inreg}[12]; \text{outreg}[12]; A[4]$

INPUTS: $X[16]; \text{en}; \text{ack}$

OUTPUTS: $Z[16]; \text{inrdy}; \text{outrdy}$

1. $\text{inrdy} = 0;$

$\rightarrow (\text{en}, \overline{\text{en}}) / (1, 2)$

2. $\text{inreg} \leftarrow X$

3. $A \leftarrow \text{inreg}_{0:3} \wedge \text{inreg}_{4:7} \wedge \text{inreg}_{8:11} \wedge \text{inreg}_{12:15};$

$\text{outreg} \leftarrow \text{inreg}$

4. $\rightarrow (\sqrt{A}, \sqrt{A}) / (1, 5)$

5. $\text{outrdy} = 0;$

$\rightarrow (\text{ack}, \overline{\text{ack}}) / (5, 1)$

END SEQUENCE

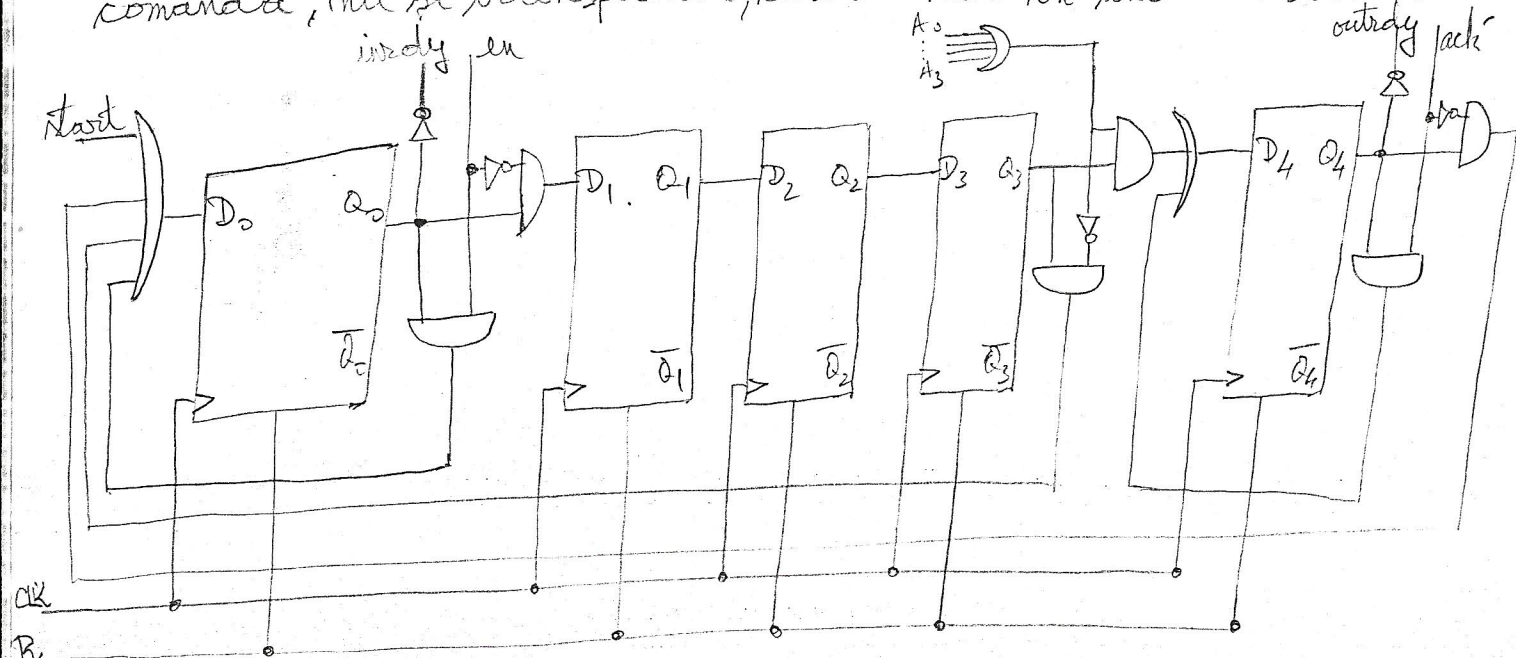
$Z = \text{outreg}$

END

Implementarea cu bistabili a unității de comandă.

Obs: 1. Fiecărei stări îi va corespunde un bistabil.

2. Fiind vorba despre UC, în schemă apar doar semnalele de comandă, nu și transferurile, care au loc în unitatea de execuție.



Instructione ROL (rotire la stânga)

13

Apelare: ROL reg.

Pas 1. Codificarea instrucțiunii

$RI_0 = 0 \rightarrow$ cu calcul de adresă efectiv

$RI_1 = 0 \rightarrow$ cu o singură adresă efectivă

$RI_2 = 0 \rightarrow$ operand imediat

$RI_3 = 1 \rightarrow$ instrucțiune operațională

$RI_{4:6} = 111 \rightarrow$ instrucțiunea se plasează pe poziția 7 în sub-grup

Pas 2. Alegerea semnificației și valorilor celorlalți biți din codul instrucțiunii

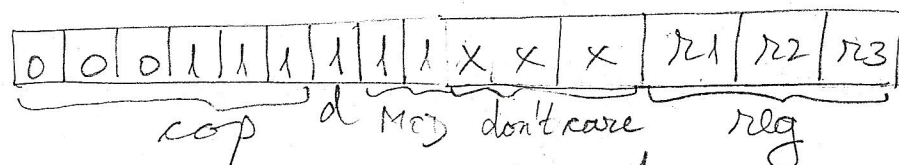
$RI_7 = 0 \rightarrow$ vom depune rezultatul în R/M

$RI_{8:9} = 11 \rightarrow$ destinația este chiar registrul

$RI_{13:15} = 000$ sau 001 sau $010 \rightarrow$ registrul R A sau R B sau R C

$RI_{10:12} = xxx \rightarrow$ don't care

Pas 3. Codul mașinii



Pas 4. Se determină modificările necesare a fi făcute în codul APL al unității de citire-interpretare a instrucțiunilor

Se urmărește organigrama din figura 5.19:

- La pasul 5 din organigramă se separă, în funcție de valorile $[RI_8:RI_9]$, instrucțiunile în două; pt. că am ales $RI_{8:9} = 11$, vom sări la pasul 26, direct în faza de execuție

- Dacă ar fi cazul, tot în acest pas se face și adaptarea codului pt. calculul adresii efective

Pas 5. Adaptarea fazei de execuție a instrucțiunilor

- La pasul 26 se testează RI_1 și se trece la 27

- La pasul 27 se testează RI_3 și se trece la 54

- La pasul 54 se transformă conținutul în reg. temp. T1

- Pasul 55 se modifică pt. a introduce noua instrucțiune:

$$55. T1 \leftarrow (ADD(T1; 0; 1) ! ADD(T1, 0FFFFH; 0) ! ADD(T1; 0; 1) ! T1 ! (T1_{1:5}, 0) ! (0, T1_{0:14}) ! (T1_{0:14}, T1_{1:5}) ! (T1_{1:15}, T1_0)) * DCD_{0:7}(RI_{4:6})$$

- Pasul 56 transformă rezultatul înapoi în registrul.