

11. UNITĂȚI DE EXECUȚIE ȘI DE COMANDĂ INTEGRATE

11.1. UNITĂȚI DE EXECUȚIE INTEGRATE

Unitățile de execuție se prezintă sub forma unor circuite integrate pe scara medie/largă. În unele cazuri ele sunt structurate în tranșe de biți astfel încât, prin concatenare și prin utilizarea unor circuite adiționale, se pot construi sisteme de prelucrare pentru date organizate pe: 4, 8, 16, 24, 32, 48 sau 64 biți.

Din punct de vedere structural, unitățile de execuție sunt constituite dintr-o unitate aritmetică-logică, logica de generare a indicatorilor de condiții, un ansamblu de registre generale, unul sau mai multe circuite de selecție, un circuit de deplasare, și din porturi de acces pentru intrarea/ieșirea datelor.

Semnalele de comandă se aplică la intrările special prevăzute, fie codificate pe câmpuri, fie complet decodificate.

Sistemul funcționează sincron, cu un ceas a cărui frecvență de tact este condiționată de întârzierile semnalelor în unitatea de execuție (propagarea transportului, timpul de basculare a bistabilelor dintr-o stare în alta etc).

Unitățile de execuție integrate s-au comercializat în asociație cu unitățile de comandă corespunzătoare și cu o serie de circuite adiționale, formând ceea ce se chiamă microprocesoare pe tranșe de biți, microprocesoare "bit-slice", microprocesoare "multi-chip" etc.

Printre cele mai răspândite familii de microprocesoare "bit-slice" s-au aflat și cele produse de compania Advanced Micro Devices, sub numele de AMD 2900.

În cele ce urmează va fi prezentată unitatea de execuție AMD 2901 sub aspect structural-funcțional. De asemenea, se vor prezenta modalitățile de realizare a unor unități de execuție pentru prelucrarea datelor pe 16 biți, unitățile de comandă corespunzătoare, cât și elemente privind microprogramarea acestora.

UNITATEA DE EXECUȚIE AMD 2901

Unitatea de execuție AMD 2901 este organizată pe tranșe de 4 biți/circuit și este prevăzută cu elementele necesare cuplării în cascadă. Semnalele de comandă se aplică sub forma unor vectori binari la terminalele circuitului, fiind, de regulă, preluate sub controlul unui circuit micro-secvențiator integrat (AMD 2909, 2911) de la o memorie cu conținut permanent. Această soluție se folosește în cazul unităților de comandă microprogramate. Uneori, pentru a mări viteza de operare, în locul unității de comandă microprogramate se utilizează automate convenționale cu porți, bistabile și registre.

Unitatea de execuție AMD 2901 dispune de următoarele resurse hardware (figura 11.1):

- un ansamblu de 16 registre generale de câte 4 biți, organizate sub forma unei memorii (RAM) biport, cu două intrări de adrese, o intrare de date și două ieșiri de date;
- o unitate aritmetică-logică, cu transport anticipat, capabilă să efectueze 3 operații aritmetice binare și 5 operații logice și să genereze, atât indicatorii de condiții: depășire (OVR), zero ($F=0$), semn (F_3), transport (C_{n+4}), cât și condițiile de propagare (\bar{P}) și generare (\bar{G}) ale transportului, la nivelul întregului circuit;
- un selector de date (selector surse UAL) pentru cele două intrări ale unității aritmetice-logice, care pot reprezenta combinații între ieșirile memoriei biport (A,B), o intrare externă de date (D), constanta "zero" și ieșirea unui registru suplimentar-extensie (Q);
- un selector de ieșire din circuit, care furnizează prin intermediul unor tamponare TS, fie datele de la ieșirea A, a memoriei biport, fie datele de la ieșirea UAL;
- un registru auxiliar-extensie (Q), care poate fi încărcat, fie cu datele de la ieșirea UAL, fie cu propriul sau conținut deplasat stânga/dreapta prin intermediul unei rețele logice de deplasare-multiplexor QSHIFT;
- o rețea de deplasare-multiplexor RAMSHIFT, plasată pe intrarea B a memoriei biport RAM.

Indicatorii de condiții trebuie stocați într-un registru după fiecare operație efectuată de UAL.

Conținuturile celor 16 cuvinte din RAM pot fi citite la ieșirile A și B, pe baza adreselor A și B, fiind stocate în două circuite tampon/latch. Stocarea are loc pe frontul crescător al semnalului de ceas. După tranziția negativă a semnalului de ceas, în RAM, în locația specificată de adresa B, se poate înscrie o nouă valoare. Astfel, în cadrul unei perioade de ceas se efectuează o operație de citire și o operație de scriere în RAM. Datele forțate în RAM sunt preluate direct sau deplasate dreapta/stânga, prin intermediul multiplexorului RAMSHIFT, de la ieșirea F a UAL. Unitatea aritmetică-logică poate efectua, sub controlul

semnalelor $I_{3:5}$, trei operații aritmetice binare și cinci operații logice, asupra operanzilor aplicați la intrările R și S.

Circuitul a fost realizat în diverse variante tehnologice: AMD 2901, 2901A, 2901B, 2901C, cu sporuri de viteză de 25%, 50% și respectiv 75%, față de 2901. În figura de mai jos se prezintă schema bloc a circuitului AMD 2901.

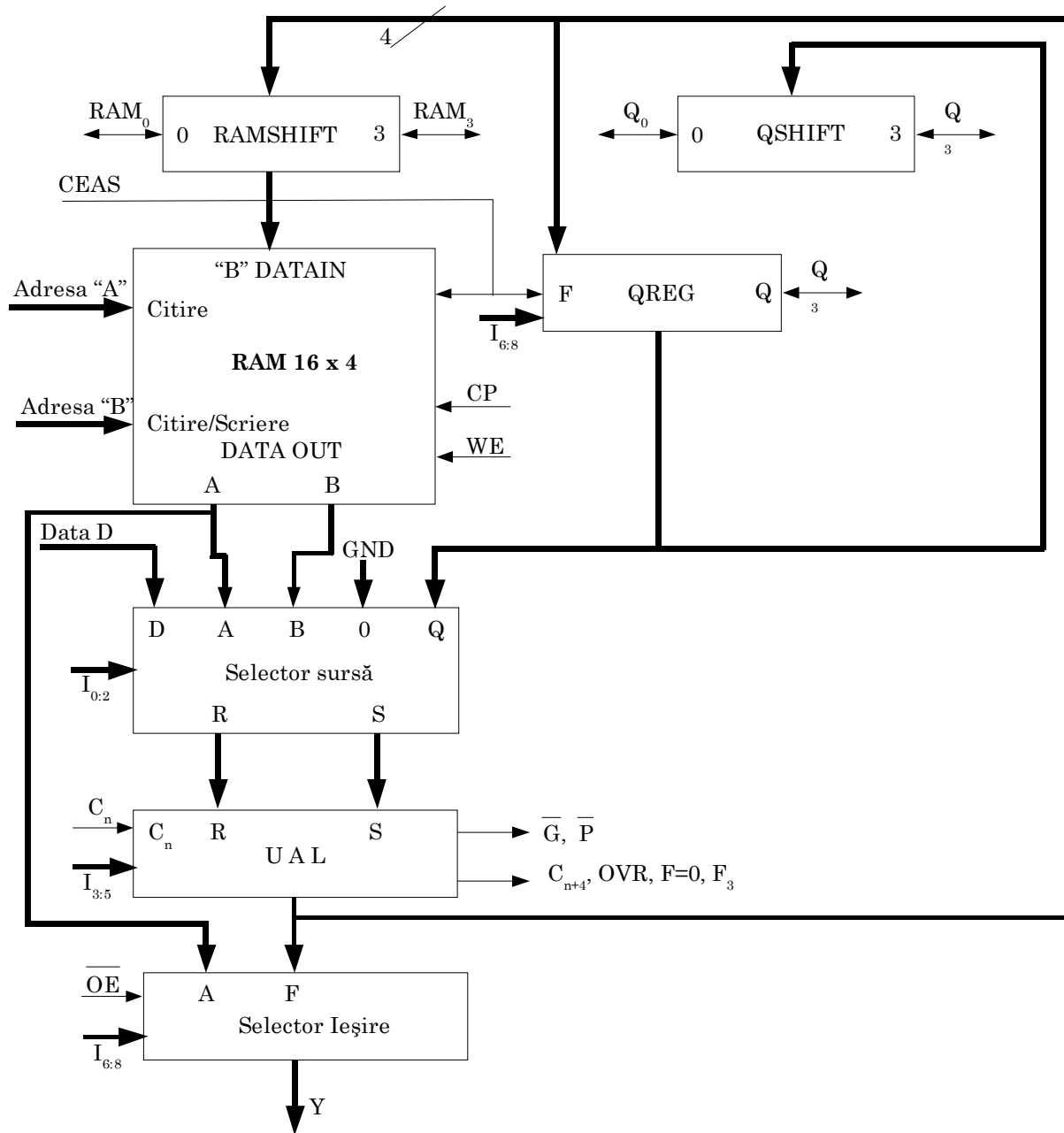


Figura 11.1. Unitatea de execuție AMD 2901

Intrările R și S ale UAL sunt preluate de la ieșirile unui circuit selector (Selector sursă UAL), care sub controlul semnalelor $I_{0,2}$, din intrările D (Date externe), A, B, 0 și Q, poate furniza combinațiile AB; AD;

AQ; A_0 ; DQ; D_0 ; Q_0 ; BD; BQ; B_0 . În tabelul 11.1 se prezintă selecția surselor de operanzi pentru UAL.

Tabelul 11.1 Selecția surselor de operanzi pentru UAL

Mnemonica	Microcod			Codul octal	Sursa	
	I_2	I_1	I_0		R	S
AQ	0	0	0	0	A	Q
AB	0	0	1	1	A	B
ZQ	0	1	0	2	0	Q
ZB	0	1	1	3	0	B
ZA	1	0	0	4	0	A
DA	1	0	1	5	D	A
DQ	1	1	0	6	D	Q
DZ	1	1	1	7	D	Z

Intrarea directă D este folosită pentru a introduce date în circuit, în RAM și în registrul Q. Registrul Q este utilizat în operațiile de înmulțire și împărțire, ca extensie sau ca acumulator/registru în unele aplicații.

Microcodul: $I_{8:0}$ (Structura vectorului de comandă).

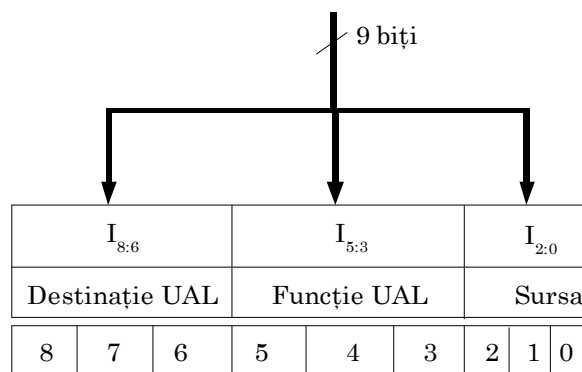


Figura 11.2. Structura vectorului de comandă

Operațiile efectuate de UAL sunt prezentate în tabelul 11.2:

Tabelul 11.2 Operațiile efectuate de UAL

Mnemonica	Microcod			Cod octal	Funcția UAL	Simbol
	I ₅	I ₄	I ₃			
ADD	0	0	0	0	R plus S	$R + S$
SUBR	0	0	1	1	S minus R	$S - R$
SUBS	0	1	0	2	R minus S	$R - S$
OR	0	1	1	3	R or S	$R \cup S$
AND	1	0	0	4	R and S	$R \cap S$
NOTRS	1	0	1	5	\bar{R} and S	$\bar{R} \cap S$
EXOR	1	1	0	6	R exor S	$R \oplus S$
EXNOR	1	1	1	7	R exnor S	$R \equiv S$

Pe baza informațiilor din tabelele 11.1 și 11.2 se pot genera funcțiile aritmetice și logice realizate de către unitatea de execuție AMD 2901.

Tabelul 11.3 Funcțiile aritmetice realizate de unitatea de execuție AMD2901

Microcod octal		C _n = 0 (Low)		C _n = 1 (High)	
I _{5:3}	I _{2:0}	Grup	Funcție	Grup	Funcție
0	0	ADD	A + Q	ADD plus 1	A + Q + 1
0	1		A + B		A + B + 1
0	5		D + A		D + A + 1
0	6		D + Q		D + Q + 1
0	2	Pass	Q	Increment	Q + 1
0	3		B		B + 1
0	4		Q		A + 1
0	7		D		D + 1
1	2	Decrement	Q - 1	Pass	Q
1	3		B - 1		B
1	4		A - 1		A
2	7		D - 1		D
2	2	Complementul față de 1	-Q - 1	Complementul față de 2	-Q
2	3		-B - 1		-B
2	4		-A - 1		-A
1	7		-D - 1		-D
1	0	Scădere în complementul față de 1	Q - A - 1	Scădere în complementul față de 2	Q - A
1	1		B - A - 1		B - A
1	5		A - D - 1		A - D
1	6		Q - D - 1		Q - D

2	0		A - Q - 1		A - Q
2	1		A - B - 1		A - B
2	5		D - A - 1		D - A
2	6		D - A - 1		D - Q

Tabelul 11.4 Funcțiile logice realizate de unitatea de execuție AMD2901

Microcod octal		Grup	Funcție
$I_{5:3}$	$I_{2:0}$		
4	0	AND	$A \cap Q$
4	1		$A \cap B$
4	5		$D \cap A$
4	6		$D \cap Q$
3	0	OR	$A \cup Q$
3	1		$A \cup B$
3	5		$D \cup A$
3	6		$D \cup Q$
6	0	EXOR	$A \oplus Q$
6	1		$A \oplus B$
6	5		$D \oplus A$
6	6		$D \oplus Q$
7	0	EXNOR	$A \equiv Q$
7	1		$A \equiv B$
7	5		$D \equiv A$
7	6		$D \equiv Q$
7	2	INVERT	\bar{Q}
7	3		\bar{B}
7	4		\bar{A}
7	7		\bar{D}
6	2	PASS	Q
6	3		B
6	4		A
6	7		D
3	2	PASS	Q
3	3		B
3	4		A
3	7		D
4	2	"ZERO"	0
4	3		0
4	4		0
4	7		0
5	0	MASK	$\bar{A} \cap Q$
5	1		$\bar{A} \cap B$
5	5		$\bar{D} \cap A$
5	6		$\bar{D} \cap Q$

Câmpul "Destinație UAL" ($I_{8:6}$), din vectorul de comandă $I_{8:0}$, are o semnificație complexă prin faptul că specifică:

- operarea RAM (deplasare/încărcare/inactiv);
- operarea registrului Q (deplasare/încărcare/inactiv);
- conexiunile la RAMSHIFT (RAM_0 și RAM_3);
- conexiunile la QSHIFT (Q_0 și Q_3);
- ieșirea Y.

Terminalele RAM_0/RAM_3 și Q_0/Q_3 , din punct de vedere electric, sunt terminale de intrare/ieșire, controlul lor fiind asigurat de câmpul $I_{8:6}$, din microcod. În tabelul de mai jos se prezintă semnificațiile câmpului $I_{8:6}$, în condițiile în care:

- B reprezintă locația din RAM, selectată pe baza adresei B;
- UP înseamnă deplasarea către MSB;
- DOWN specifică deplasarea către LSB.

Tabelul 11.5 Comandă destinație UAL

Mnemonica	Microcod			Cod	Funcție RAM		Funcția Q		Ieșire	RAMSHIFT		QSHIFT	
	I_8	I_7	I_6		Depl	Încărc	Depl	Încărc		RAM_0	RAM_3	Q_0	Q_3
QREG	0	0	0	0	*	Nu	*	$Q \leftarrow F$	F	*	*	*	*
NOP	0	0	1	1	*	Nu	*	Nu	F	*	*	*	*
RAMA	0	1	0	2	Nu	$B \leftarrow F$	*	Nu	A	*	*	*	*
RAMF	0	1	1	3	Nu	$B \leftarrow F$	*	Nu	F	*	*	*	*
RAMQD	1	0	0	4	dow n	$B \leftarrow F/2$	dow n	$Q \leftarrow 2Q$	F	F_0	IN_3	Q_0	IN_3
RAMD	1	0	1	5	dow n	$B \leftarrow F/2$	*	Nu	F	F_0	IN_3	Q_0	*
RAMQU	1	1	0	6	Up	$B \leftarrow 2F$	Up	$Q \leftarrow 2Q$	F	IN_0	F_3	IN_0	Q_3
RAMU	1	1	1	7	Up	$B \leftarrow 2F$	*	Nu	F	IN_0	F_3	*	Q_3

*) Din punct de vedere electric terminalul asigură o intrare TTL, conectată intern cu o ieșire TS aflată în starea de mare impedanță.

Intrarea B în RAM este furnizată de un multiplexor RAMSHIFT. Acesta, sub controlul microcodului $I_{8:6}$, selectează intrarea directă sau deplasată stângă/dreapta, cu inserția unor date specifice prin terminalele RAM_3/RAM_0 . După cum se va vedea ulterior, datele pentru terminalele RAM_3/RAM_0 sunt selectate cu ajutorul altor biți (MUX_1, MUX_0) din microcodul de comandă și cu un hardware adițional, constituit din două multiplexoare AMD 25LS-253, cu câte 8 intrări și 2 ieșiri. Multiplexoarele AMD 25LS253, amintite mai sus, soluționează și problemele circuitului QSHIFT, care asigură una din intrările registrului Q.

Ieșirea Y este realizată cu un circuit TS. Sub controlul semnalului \overline{OE} , ieșirea Y se conectează la o magistrală de date.

Intrarea de ceas în circuitul AMD 2901 controlează resursele RAM, Q și tamponarele-latch A și B, plasate pe ieșirile RAM. Când semnalul de ceas este pe nivel ridicat tamponarele A și B sunt activate, preluând datele din locațiile citite din RAM, pentru a le furniza la ieșire, ca surse UAL. Pe durata nivelului coborât al semnalului de ceas tamponarele A și B stochează datele încărcate anterior. În cazul în care se activează semnalul \overline{WE} , pe nivelul coborât al semnalului de ceas se poate stoca în RAM noua dată plasată la intrarea B, prin intermediul circuitului RAMSHIFT.

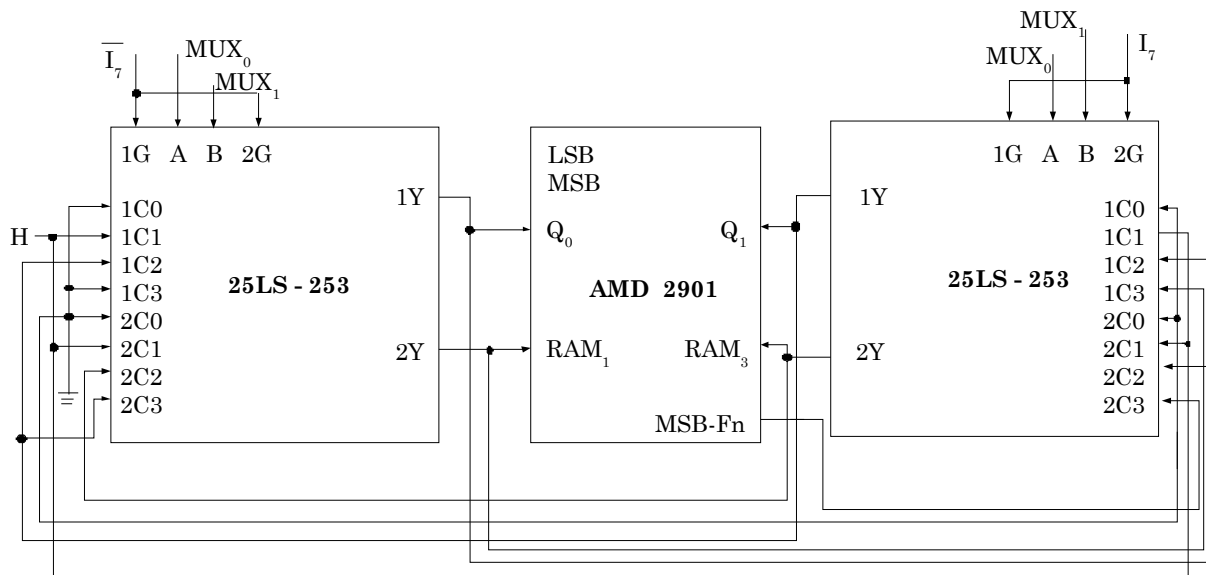


Figura 11.3. Schema bloc cu multiplexoare AMD 25LS253.

În continuare se va prezenta schema bloc parțială a unei unități de execuție pe 16 biți, constituită din patru circuite AMD 2901 și un circuit cu transport anticipat AMD 2902

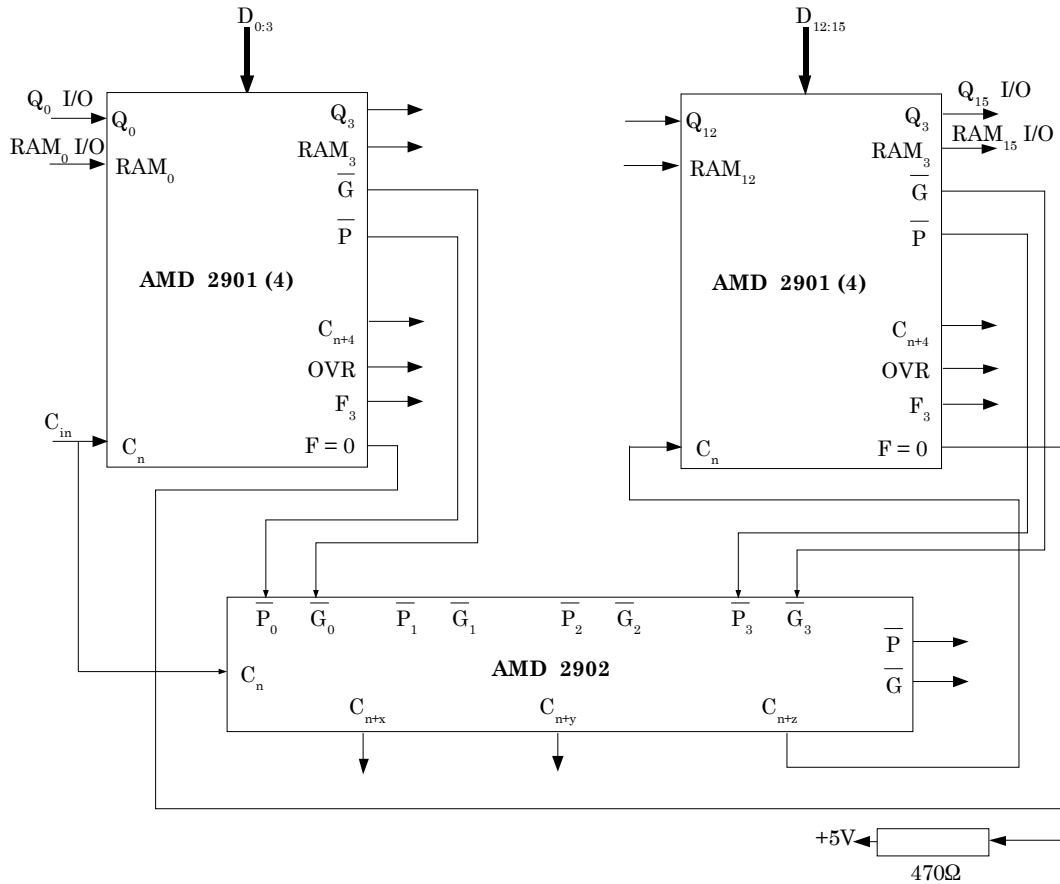


Figura 11.4. Schema bloc parțială a unei unități de execuție pe 16 biți.

Unitatea de execuție poate utiliza și multiplexoare 4:1 obișnuite, în locul celor produse de către AMD, 25LS-253. În schema din figura 11.5 se arată modul în care sunt interconectate logic cele 4 multiplexoare.

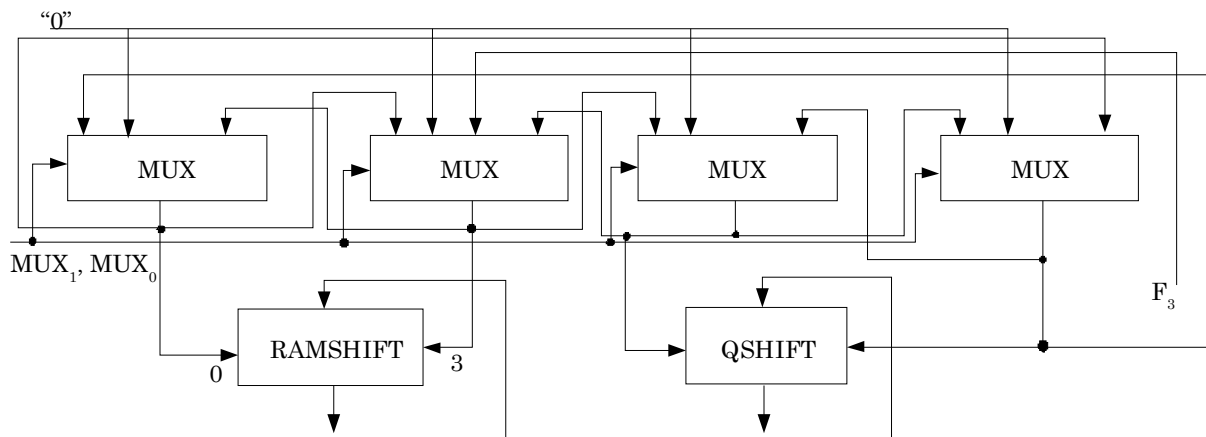


Figura 11.5. Schema bloc a unității de execuție cu multiplexoare 4:1 obișnuite.

Plecând de la elementele prezentate anterior se poate construi formatul cuvântului de comandă, al microinstrucțiunii, pentru controlul unității de execuție.

În scop didactic se va considera data de prelucrat conținută în microinstrucțiune, într-un câmp D. În cazul unei unități de prelucrare pe 4 biți, câmpul D va avea, de asemenea, 4 biți. Pe măsură ce crește lungimea cuvântului prelucrat, va crește și dimensiunea câmpului D, din microinstrucțiune. În condițiile plasării unor multiplexoare pe intrările de deplasare ale circuitelor RAMSHIFT și QSHIFT, registrele R și Q se pot concatena în vederea efectuării unor deplasări sau rotiri duble, conform tabelului 11.6 (originea de indexare este la stânga):

Tabelul 11.6 Deplasări efectuate prin concatenarea regsitrelor R și Q

MUX ₁	MUX ₀	TIP	DREAPTA (JOS)	STÂNGA (SUS)
0	0	Zero	$RAM_3 \leftarrow 0;$ $Q_3 \leftarrow 0$	$RAM_0 \leftarrow 0;$ $Q_0 \leftarrow 0$
0	1	Rotire	$RAM_3 \leftarrow RAM_0;$ $Q_3 \leftarrow Q_0$	$RAM_0 \leftarrow RAM_3;$ $Q_0 \leftarrow Q_3$
1	0	Rotire cu lungime dublă	$RAM_3 \leftarrow Q_0;$ $Q_3 \leftarrow RAM_0$	$RAM_0 \leftarrow Q_3;$ $Q_0 \leftarrow RAM_3$
1	1	Deplasare dublă aritmetică	$RAM_3 \leftarrow F_3;$ $Q_3 \leftarrow RAM_0$	$RAM_0 \leftarrow Q_3;$ $Q_0 \leftarrow 0$

Cuvântul (microinstrucțiunea) de comandă al unității de execuție va avea structura următoare:

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	I ₈	I ₇	I ₆	M	I ₂	I ₁	I ₀	C _n	I ₅	I ₄	I ₃	A ₃	A ₂	A ₁	A ₀	B ₃	B ₂	B ₁	B ₀	D ₃	D ₂	D ₁	D ₀
U	Selecție			U	Selecție				Operație			Adresa "A"				Adresa "B"				Data "D"			
X	destinație			X	sursă				UAL														

Se poate observa că, microinstrucțiunea care controlează unitatea de execuție posedă 24 de biți.

Biții 0, .. ,3 au apărut ca urmare a înglobării datei în microinstrucțiune (microinstrucțiune cu operand imediat). Câmpul D se va extinde în incremenți de 4 biți, odată cu extinderea lungimii cuvântului prelucrat de către unitatea de execuție. Câmpul C_n este asociat cu transportul în rangul cel mai puțin semnificativ al unității de execuție, fiind la latitudinea celui care scrie microprogramul.

11.2 UNITĂȚI DE COMANDĂ INTEGRATE

SECVENȚIATORUL DE MICROPROGRAM AMD 2909

Secvențiatorul de microprogram AMD 2909 face parte din familia de circuite AMD2900 și constituie elementul de bază în jurul căruia este organizată unitatea de comandă microprogramată.

După cum s-a aratat în paragraful anterior, atât unitatea de execuție AMD 2901, cât și circuitele asociate acestuia sunt controlate cu ajutorul unui vector de comandă constituit din câmpuri de biți, cu funcții bine precizate. Un asemenea vector controlează operațiile aritmetice și logice elementare, care se pot efectua pe durata unei perioade de ceas. Vectorul semnalelor de comandă poate fi generat cu ajutorul unui circuit secvențial, în cazul unităților de comandă convenționale, sau extras dintr-o memorie cu conținut permanent, în cazul unităților de comandă microprogramate.

Operațiile cu caracter complex, printre care se numără și instrucțiunile calculatoarelor convenționale, se implementează sub controlul unor secvențe de vectori de comandă. În sistemele microprogramate aceste secvențe sunt stocate într-o memorie de control sau memorie de microprograme.

Un cuvânt al memoriei de control reprezintă un vector de comandă sau o microinstrucțiune.

Secvențiatorul de microprogram, structurat pe tranșe de câte 4 biți, este destinat generării adreselor de microinstrucțiuni și asigurării înlănțuirii acestora.

Secvențiatorul, prezentat mai jos, are ca resurse hardware:

- Contorul de MicroProgram (CMP);
- Registrul de Ramificare (RR);
- Indicatorul de Stivă (IS);
- STiVa (STV) cu capacitate de 4 cuvinte \times 4 biți;
- INCrementatorul (INC);
- MULTipleXorul (MUX) $4 \times (4 : 1)$;
- Tampon cu trei Stari (TS) și alte circuite logice combinaționale auxiliare.

Schema bloc a unității de comandă va conține, pe lângă microsecvențiator, și elementele necesare realizării unei unități de comandă:

- Memoria de Microprogram (MM);
- Registrul MicroInstrucțiunii (RMI);
- un Comutator de Adrese (CA);
- o memorie de tip PROM.

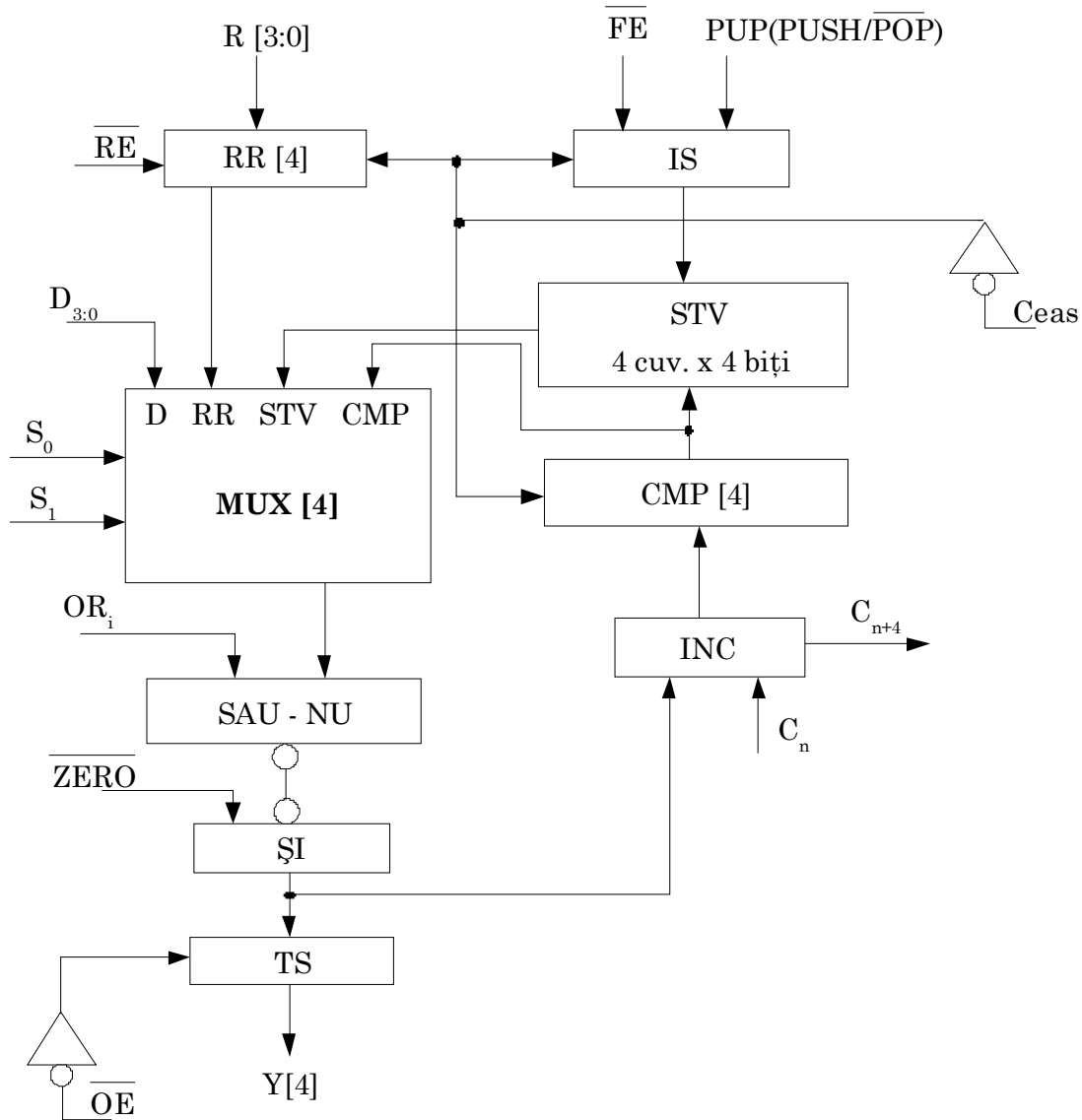


Figura 11.6. Schema bloc a unității de comandă

Prin conectarea în cascada a trei asemenea circuite se asigură o adresă de microinstrucțiune de 12 biți, ceea ce permite accesul la o memorie de comandă cu capacitatea de 4096 cuvinte.

Secvențiatorul furnizează o adresa $Y[4]$ de microinstrucțiune, care reprezintă ieșirea $X[4]$ a multiplexorului, prelucrată logic. Multiplexorul are ca intrări selectabile cu ajutorul semnalelor S_1, S_0 :

- adresa externă D;
- conținutul Registrului de Ramificare (RR);
- conținutul vârfului stivei (STV)
- conținutul Contorului de MicroProgram.

Ieșirea Y este furnizată prin intermediul unui circuit tampon TS, controlat de semnalul \overline{OE} .

Din punct de vedere logic, $Y[4]$ se exprimă astfel: $Y[4] = \overline{ZERO} \cap (Y[4] \cup OR[4])$ unde $OR[4]$ este un vector logic aplicat din exterior, adesea având toți biții egali cu 1. \overline{ZERO} este, de asemenea, un vector logic, care poate face ca $Y[4]$ să aibe toate rangurile egale cu 0. În acest mod se pot forța valori particulare pentru adresa microinstrucțiunii următoare sau se pot efectua transferuri relative la adresa curentă.

Secvențierea microinstrucțiunilor este realizată prin controlul circuitului 2909 cu ajutorul semnalelor preluate din câmpurile de control ale microinstrucțiunii și de la condițiile externe.

Operarea microsecvențiatorului este descrisă cu ajutorul următoarelor tabelelor 11.7.

Tabelul 11.7 Operarea microsecvențiatorului

Selecție MUX

S_1	S_0	Ieșire
0	0	CMP
0	1	RR
1	0	STV
1	1	D

Controlul ieșirii

OR_i	\overline{ZERO}	\overline{OE}	Y_i
*	*	1	Z (TS)
*	0	0	0
1	1	0	1
0	1	0	IES MUX

Operarea stivei

\overline{FE}	PUP	Operație
1	*	Inactivă
0	1	$IS \leftarrow INC(IS); PUSH(CMP)$
0	0	$POP(STV); DCR(IS)$

În funcție de semnalele de control și de condițiile externe, microsecvențiatorul poate realiza următoarele operații de control, la nivelul microprogramului:

- *Transfer necondiționat la adresa X* – mnemonica J X;
- *Chemare de microsubrutina* – mnemonica CALL X;
- *Revenire din microsubrutina* – mnemonica RETURN;
- *Trecerea la microinstrucțiunea următoare din secvența* – mnemonica CONTINUE.

Transferul necondiționat la adresa X se realizează prin aplicarea vectorului binar X la intrarea D a multiplexorului și a următorului vector binar la intrările de control ale microsecvențiatorului:

$S_1 S_0, \overline{FE}, PUP, OR[4], \overline{ZERO} = 1\ 1, 1, *, 0000, 1$

Chemarea unei microsubrutine, de la adresa X, este asigurată prin forțarea adresei X în registrul RR și prin aplicarea următorului vector binar la intrările de control ale microsecvențiatorului:

$S_1 S_0, \overline{FE}, PUP, OR[4], \overline{ZERO} = 0\ 1, 0, 1, 0000, 1$

Revenirea dintr-o microsubrutină este implementată folosind un vector de control al microsecvențiatorului cu următoarea structură:

$S_1 S_0, \overline{FE}, PUP, OR[4], \overline{ZERO} = 1\ 0, 0, 0, 0000, 1$

Trecerea la următoarea microinstrucțiune din secvență impune ca adresa Y să corespundă conținutului Contorului de Microprogram, ceea ce se realizează cu ajutorul următorului vector de comandă:

$S_1 S_0, \overline{FE}, PUP, OR[4], \overline{ZERO} = 0\ 0, 1, *, 0000, 1$

Conform celor arătate mai sus, microsecvențiatorul efectuează următoarele operații privitoare la generarea adresei Y a microinstrucțiunii următoare și la modificarea conținuturilor Contorului de MicroProgram și Stivei:

$$Y = \left(((CMP!RR!STV!D) * DCD_{0:3}(S_1, S_0) \cup OR_{0:3})! 4 \uparrow 0 \right) * (\overline{ZERO}, ZERO)$$

$$CMP \leftarrow \left((INC(CMP!RR!STV!D) * DCD_{0:3}(S_1, S_0) \cup OR_{0:3})! 4 \uparrow 0 \right) * (\overline{ZERO}, ZERO)$$

$$STV \leftarrow \left((STV_{1:3}! STV_0)! (CMP!CMP! STV_{0:2})! STV!STV \right) * DCD_{0:3}(\overline{FE}, PUP)$$

Conectarea în cascadă a circuitelor microsecvențiatorului permite explorarea unui spațiu mai mare în memoria de microprogram. În figura 11.7 se prezintă interconectarea în cascadă a trei microsecvențioare AMD 2909.

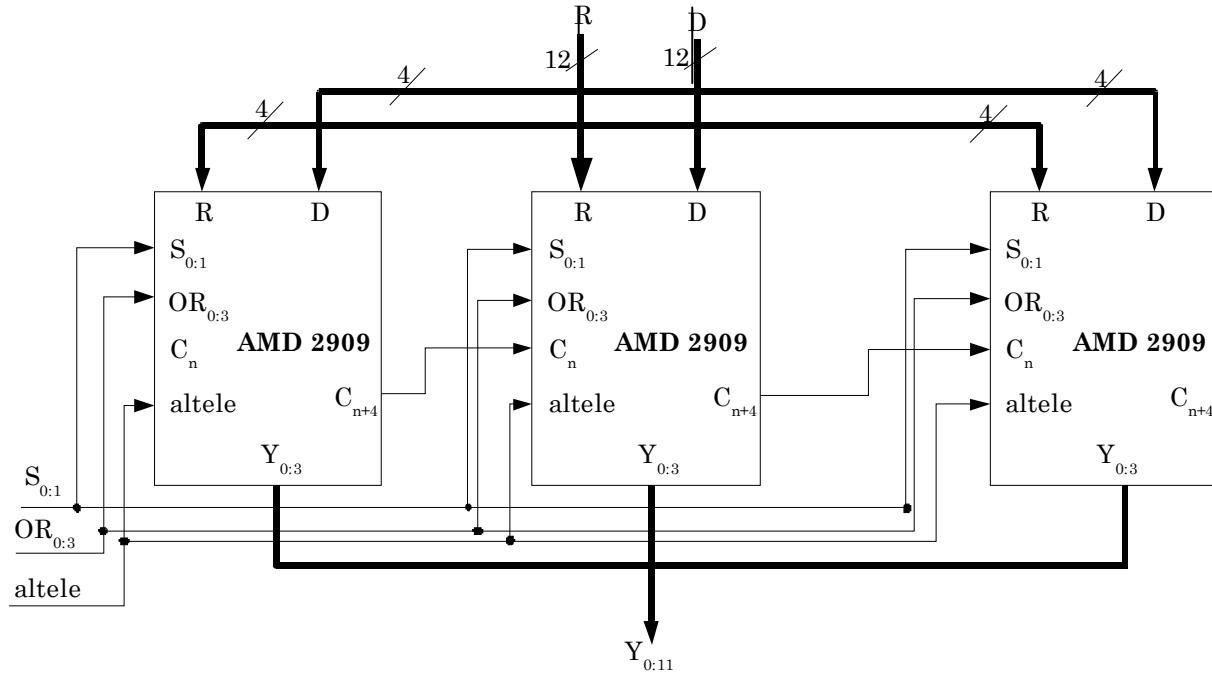


Figura 11.7. Interconectarea în cascadă a trei microsecvențioare AMD 2909.

După cum s-a mai arătat, pentru realizarea unei unități de comandă, microsecvențiatorul trebuie prevăzut cu o serie de circuite auxiliare, care sunt prezentate în figura 11.8.

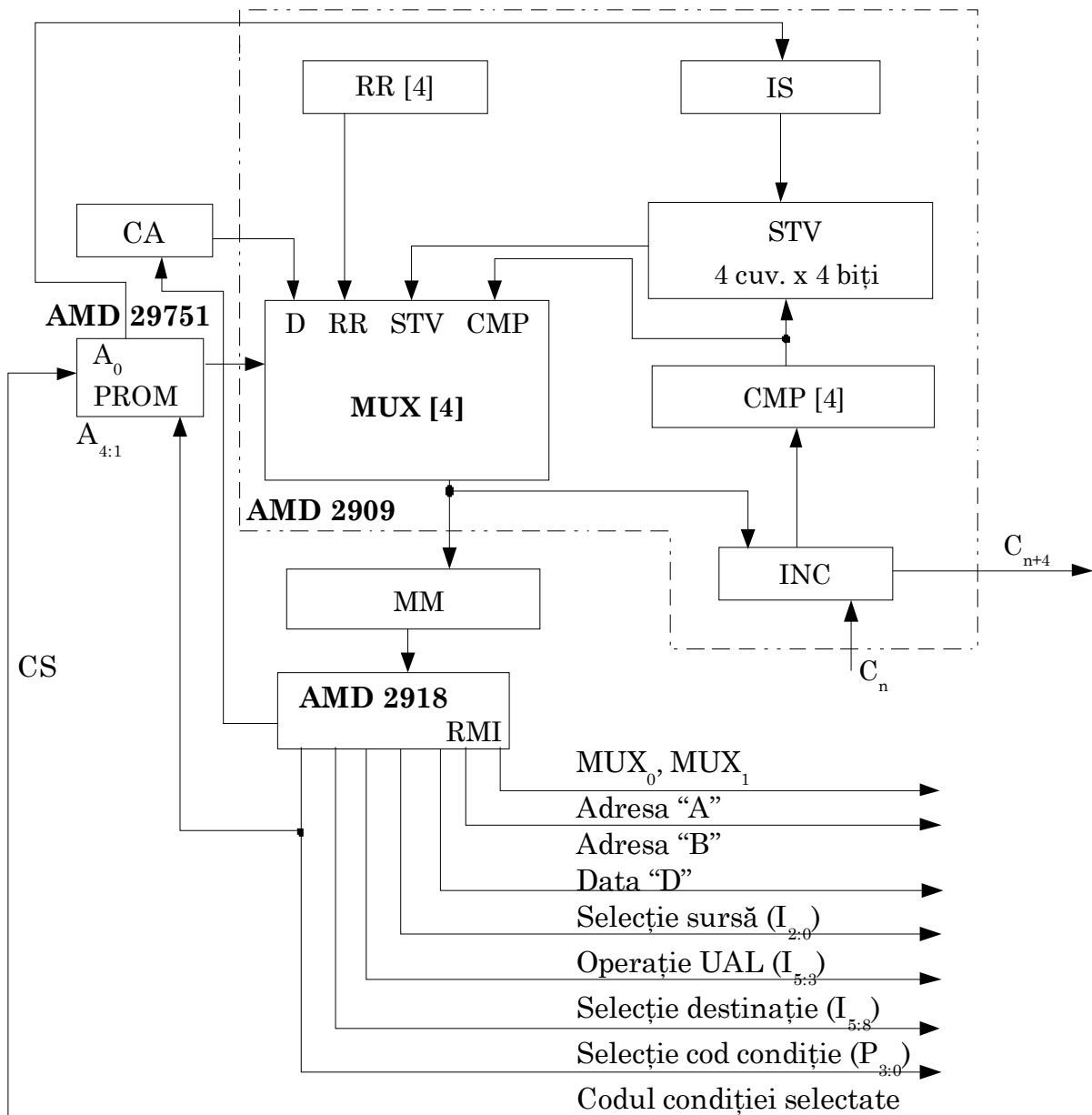


Figura 11.8. Microsecvențiatorul cu circuite auxiliare.

Între acestea un rol deosebit îl joacă memoria PROM 29751, care furnizează semnalele de comandă pentru microsecvențiator. Conținutul acestei memorii PROM este citit pe baza unei adrese obținute dintr-un câmp $P_{0:3}$, al registrului microinstrucțiunii RMI. Biții $P_{0:3}$ ai microinstrucțiunii se transformă în biții de adresă $A_{1:4}$, ai memoriei PROM, în timp ce bitul A_0 este furnizat de ieșirea unui multiplexor la intrările căruia se aplică indicatorii de condiții, cât și alte semnale. Intrările de date ale multiplexorului au fost stabilite pe durata execuției microinstrucțiunii precedente. Intrările de selecție ale multiplexorului

sunt asigurate tot de biții $P_{0:3}$, furnizați de microinstrucțiunea curentă. În figura 11.9 se prezintă o schema bloc a unității de comandă, cu indicarea elementelor discutate anterior.

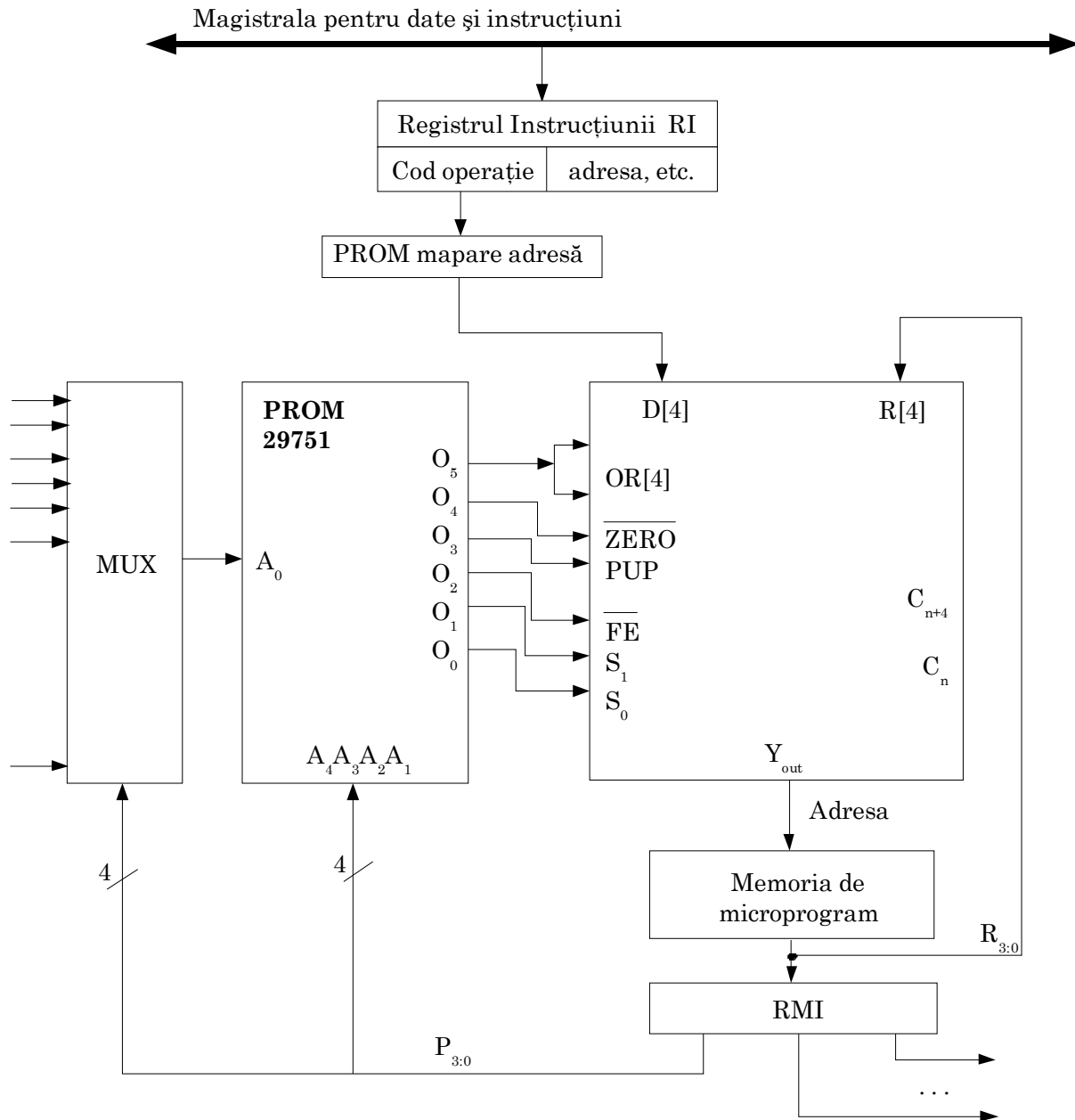


Figura 11.9. Schema bloc a unității de comandă, cu memorie PROM 29751.

Operarea unității de comandă microprogramată este stabilită prin conținutul memoriei PROM, care furnizează la ieșire semnalele $O_{5:0}$, care reprezintă vectorul de comandă al microsecvențiatorului: $OR[4]$, \overline{ZERO} , PUP , \overline{FE} , S_1 , S_0 . În locațiile memoriei PROM, adresate cu ajutorul câmpului $P_{0:3}$, sunt stocați

vectori de comandă, care impun unității de comandă microprogramate efectuarea operațiilor prezentate în tabelul 11.8.

Tabelul 11.8 Operarea unității de comandă

$P_{3:0}$	Operație	Mnemonică
Cod hexa		
0	Ramificare la adresa R dacă $F \neq 0$	JRNZF
1	Ramificare necondiționată la adresa R	JR
2	CONTINUĂ	CONT
3	Ramificare la adresa D	JD
4	Ramificare la subrutina cu adresa R dacă $F \neq 0$	JSRNZF
5	Ramificare la subrutina cu adresa R	JSR
6	Revenire din subrutină	RS
7	Ramificare la adresa conținută în vârful stivei (fără POP)	JSTV
8	Terminare de ciclu și POP dacă $F = 0$	TCPOZF
9	PUSH și CONTINUĂ	PUCONT
A	POP și CONTINUĂ	POCONT
B	Terminare de ciclu și POP dacă $C_{n+4} = 1$	TCPOC
C	Ramificare la adresa R dacă $F = 0$	JRZF
D	Ramificare la adresa R dacă $F_3 = 1$	JRF3
E	Ramificare la adresa R dacă $OVR = 1$	JROVR
F	Ramificare la adresa R dacă $C_{n+4} = 1$	JRC

Se poate observa, în ultima schema bloc, faptul că intrarea R a microsecvențiatorului este preluată direct de la ieșirea memoriei de comandă, fără a mai fi stocat în prealabil în registrul microinstrucțiunii RMI. În acest mod adresa de ramificare se încarcă direct în registrul R al microsecvențiatorului.

Din cele prezentate rezultă că operarea microsecvențiatorului este controlată de două câmpuri ale microinstrucțiunii stocate în Memoria de comandă. Este vorba de câmpul $P_{3:0}$, care stabilește operația, conform tabelului de mai sus, și de câmpul $R_{3:0}$, al adresei de ramificare.

Prin concatenarea mai multor microsecvențiatoare câmpul P rămâne constant ca lungime, în timp ce câmpul R se modifică, în incremenți de câte 4 biți, pentru a furniza spații de adresare mai mari în memoria de microprogram.

Structura unui cuvânt – microinstrucțiune, destinat controlului unui microsecvențiator AMD 2909, este formată din două câmpuri de câte patru biți:

- câmpul P, care specifică modul de obținere a microinstrucțiunii următoare, și
- câmpul R, care definește adresa de salt, dacă este cazul.

31	30	29	28	27	26	25	24
R ₃	R ₂	R ₁	R ₀	P ₃	P ₂	P ₁	P ₀
Adresa de salt				Obținerea μI următoare			

Microprogramarea unui procesor “bit-slice”, constituit cu ajutorul circuitelor AMD 2901 și AMD 2909, în vederea prelucrării datelor pe 4 biți.

11.3 MICROPROGRAMARE

Exemple. Secvențe de microinstrucțiuni pentru controlul operațiilor în unitatea AMD2901

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Observații
MUX ₁	l ₈	l ₇	l ₆	MUX ₂	l ₂	l ₁	l ₀	C _n	l ₅	l ₄	l ₃	A ₃	A ₂	A ₁	A ₀	B ₃	B ₂	B ₁	B ₀	D ₃	D ₂	D ₁	D ₀	Descrierea microinstruc- țiunilor individuale la nivel de mnemonice
	Selecție destinație				Selecție sursă				Operație UAL			Adresa "A"				Adresa "B"				Data "D"				
	0	1	1		1	1	1		0	1	1					0	0	0	0	0	0	1	0	LOAD R ₀ , 02H
	0	0	1		0	1	1		0	1	1					0	0	0	0					READ R ₀
0	0	1	1	1	1	1	1		0	1	1					0	0	0	1	0	1	0	0	LOAD R ₁ , 04H
0	1	1	1	1	0	1	1		0	1	1					0	0	0	1					ROTUP R ₁
0	1	0	1	1	0	1	1		0	1	1					0	0	0	1					ROTDWN R ₁
	0	0	0		0	1	0		1	0	0													CLR Q
1	1	1	0	0	0	1	1		0	1	1					0	0	0	1					DBLROTUP R ₁ , Q
1	1	0	0	0	0	1	1		0	1	1					0	0	0	1					DBLROTDWN R ₁ , Q
	0	0	1		0	1	1		0	1	1													NOP
	0	1	1		0	1	1		1	0	0					0	0	0	1					CLR R ₁
	0	1	1		0	1	1	1	0	0	0					0	0	0	1					INCR R ₁
	0	1	1		0	1	1	0	0	0	1					0	0	0	1					DCR R ₁
	0	1	1		1	1	1		0	1	1					0	0	1	1	0	1	0	1	LOAD R ₃ , 05H
	0	1	1		0	0	1	0	0	0	0	0	0	1	1	0	0	0	1					ADD R ₁ , R ₃

Secvențe de microinstrucțiuni pentru controlul operațiilor în Microsecvențiatorul AMD2909

Adresa de memorie, în memoria de control	31	30	29	28	27	26	25	24	. . .	Observații	
	R ₃	R ₂	R ₁	R ₀	P ₃	P ₂	P ₁	P ₀		. . .	Descrierea microinstrucțiunilor individuale la nivel de mnemonice
	Adresa de salt				Obținerea de microinstrucțiuni următoare						
0	1	0	0	1	0	0	0	1		JR 09H	
...	
3	0	0	0	0	0	0	0	1		JR 00H	
...	
6	1	0	1	0	0	0	0	1		JR 0AH	
...	
9	0	1	1	0	0	0	0	1		JR 06H	
A					0	0	1	0		CONT	
B					0	0	1	0		CONT	
C					0	0	1	0		CONT	
D					0	0	1	0		CONT	
E					0	0	1	0		CONT	
F	0	0	1	1	0	0	0	1		JR 03H	

Secvența: 0, 9, 6, A, B, C, D, E, F, 3, 0, 9, 6, ... etc.

Secvențe de microinstrucțiuni pentru controlul operațiilor în Microsecvențiatorul AMD2909

Adresa de memorie, în memoria de control	31	30	29	28	27	26	25	24	...	Observații
	R ₃	R ₂	R ₁	R ₀	P ₃	P ₂	P ₁	P ₀		Descrierea
	Adresa de salt				Obținerea de microinstrucțiuni următoare				...	microinstrucțiunilor individuale la nivel de mnemonice
0					0	0	1	0		CONT
1					0	0	1	0		CONT
2					1	0	0	1		PUCONT
3					0	0	1	0		CONT
4					0	0	1	0		CONT
5					0	0	1	0		CONT
6					0	0	1	0		CONT
7					0	0	1	0		CONT
8					0	0	1	0		CONT
9					0	0	1	0		CONT
C					0	0	1	0		CONT
D					0	0	1	0		CONT
E					0	1	1	0		JSTV

Ciclare în microprogram. Secvența: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, 3, 4, 5, ... etc.

Secvențe de microinstrucțiuni pentru controlul operațiilor în Microsecvențiatorul AMD2909

Adresa de memorie, în memoria de control	31	30	29	28	27	26	25	24	...	Observații
	R ₃	R ₂	R ₁	R ₀	P ₃	P ₂	P ₁	P ₀		Descrierea
	Adresa de salt				Obținerea de microinstrucțiuni următoare				...	microinstrucțiunilor individuale la nivel de mnemonice
0					0	0	1	0		CONT
1					0	0	1	0		CONT
2					0	0	1	0		CONT
3	1	1	0	0	0	1	0	1		JSR 0CH
4					0	0	1	0		CONT
5					0	0	1	0		CONT
6	0	0	0	0	0	0	0	1		JR 00H
7
8
9
C					0	0	1	0		CONT
D					0	0	1	0		CONT
E					0	1	1	0		RS

Salt la subrutină. Secvența: 0, 1, 2, 3, C, D, E, 4, 5, 6, 0, 1, 2, 3, C, D, ... etc.

Microsubrutine îmbricate

μP Principal	μSUB 0	μSUB 3	μSUB 6	μSUB 9	μSUB C
D CONT	0 JSR 0CH	3 JSR 0CH	6 JSR 0CH	9 JSR 0CH	C RS
E JSR 00H	1 JSR 06H	4 JSR 09H	7 JSR 03H	A RS	
F JR 0DH	2 RS	5 RS	8 RS		

Spațiul de adresare în memoria de comandă este următorul:

D, E, 0, 1, 6, D, 7, 3, D, 4, 9, A, 5, 8, 2, F, D etc

Microprogramul corespunzător este dat mai jos:

Secvențe de microinstrucțiuni pentru controlul operațiilor în Microsecvențiatorul AMD2909

Adresa de memorie, în memoria de control	31	30	29	28	27	26	25	24	. . .	Observații
	R ₃	R ₂	R ₁	R ₀	P ₃	P ₂	P ₁	P ₀		Descrierea
	Adresa de salt				Obținerea de microinstrucțiuni următoare				. . .	microinstrucțiunilor individuale la nivel de mnemonice
0	1	1	0	0	0	1	0	1		JSR 0CH
1	0	1	1	0	0	1	0	1		JSR 06H
2					0	1	1	0		RS
3	1	1	0	0	0	1	0	1		JSR 0CH
4	1	0	0	1	0	1	0	1		JSR 09H
5					0	1	1	0		RS
6	1	1	0	0	0	1	0	1		JSR 0CH
7	0	0	1	1	0	1	0	1		JSR 03H
8					0	1	1	0		RS
9					0	0	1	0		CONT
A					0	1	1	0		RS
C					0	0	1	0		RS
D					0	1	0	1		CONT
E	0	0	0	0	0	1	0	1		JSR 00H
F	1	1	0	1						JSR 0DH

Microinstrucțiuni îmbricate. Secvențierea adreselor de microinstrucțiuni: D, E, 0, 1, 6, D, 7, 3, D, 4, 9, A, 5, 8, 2, F, D etc.

Adr μ l	31...28	27...24	23	22...20	19	18...16	15	14...12	11...8	7...4	3...0	Observații
	R _{3:0}	P _{3:0}	M ₁	I _{8:6}	M ₀	I _{2:0}	C _n	I _{5:3}	"A"	"B"	"D"	Descrierea microinstrucțiunilor
0		2		3		3	1	0		0		CONT INC R ₀
1	15	15		1								JRC 15 NOP
2	0	1		1								JR 0 NOP
15		2		1								CONT NOP
INCREMENTARE ȘI TESTARE												
0		2		3		7		3		0	0	CONT LOAD R ₀ , 0
1	14	1		3		7		3		15	15	JR 14 LOAD R ₁₅ , 15
14		2		3		3		3		0		CONT READ R ₀
15	14	1		3		3		3		15		JR 14 READ R ₁₅
ÎNCĂRCARE ȘI CITIRE / AFIȘARE												

Calculul numărului de unități conținute în cuvintele V_0 , V_1 și V_2 , stocate în trei registre R_0 , R_1 și R_2 din RAM

- 0 LOAD R₀, V₀
- 1 LOAD R₁, V₁
- 2 LOAD R₂, V₂
- 3 LOAD R₄, V₄ /V₄ conține o valoare corespunzătoare numărului de biți explorați la nivel de cuvânt
- 4 CLEAR R₃
- 5 R₀ \cap D = 0001
- 6 R₀ \leftarrow R₀/2; JSRNZF 14
- 7 R₁ \cap D; D=0001
- 8 R₁ \leftarrow R₁/2; JSRNZF 14
- 9 R₂ \cap D; D=0001
- 10 R₂ \leftarrow R₂/2; JSRNZF 14
- 11 DCR R₄
- 12 JRZF 5
- 13 NOP; JR 15
- 14 RS; INC R₃
- 15 READ R₃; JR 15

Adr μ l	31...28	27...24	23	22...20	19	18...16	15	14...12	11...8	7...4	3...0	Observații
	R _{3:0}	P _{3:0}	M ₁	I _{8:6}	M ₀	I _{2:0}	C _n	I _{5:3}	"A"	"B"	"D"	Descrierea microinstrucțiunilor
0		2		3		7		3		0	V ₀	CONT LOAD R ₀ , V ₀
1		2		3		7		3		1	V ₁	CONT LOAD R ₁ , V ₁
2		2		3		7		3		2	V ₂	CONT LOAD R ₂ , V ₂
3		2		3		7		3		4	4	CONT LOAD R ₄ , 4
4		2		3		3		4		3		CONT LOAD R ₄ , 0
5		2		1		5		4	0	0	1	CONT R ₀ \cap MSK(001)
6	14	4		5		3		3		0		JSNZF 14 R ₀ \leftarrow R ₀ /2
7		2		1		5		4	1	1	1	CONT R ₀ \cap MSK(001)
8	14	4		5		3		3		1		JSNZF 14 R ₁ \leftarrow R ₁ /2
9		2		1		5		4	2	2	1	CONT R ₂ \cap MSK(001)
10	14	4		5		3		3		2		JSNZF 14 R ₂ \leftarrow R ₂ /2
11		2		3		3	0	1		4		CONT DCR R ₄
12	5	0		1								JRNZF 5 NOP
13	15	1		1								JR 15 NOP
14		6		3		3	1	0				RS INC R ₃
15	15	1		1		3		3				JR 15 READ R ₁₅

Implementarea unui contor pe 16 biți

Contorul este realizat prin concatenarea registrelor de câte 4 biți: R₃, R₂, R₁, R₀. Se presupune că microprogramul începe de la adresa 7.

- 7 R₀ \leftarrow R₀ + 1
- 8 JRNZF 7
- 9 R₁ \leftarrow R₁ + 1
- 10 JRNZF 7
- 11 R₂ \leftarrow R₂ + 1
- 12 JRNZF 7
- 13 R₃ \leftarrow R₃ + 1
- 14 JRNZF 7
- 15 JR 7

Adr μ l	31...28	27...24	23	22...20	19	18...16	15	14...12	11...8	7...4	3...0	Observații
	R _{3:0}	P _{3:0}	M ₁	I _{8:6}	M ₀	I _{2:0}	C _n	I _{5:3}	"A"	"B"	"D"	Descrierea microinstrucțiunilor
7		2		3		3	1			0		CONT R ₀ ← R ₀ + 1
8	7	0		1								JRNZF 7 NOP
9		2		3		3	1			1		CONT R ₁ ← R ₁ + 1
10	7	0		1								JRNZF 7 NOP
11		2		3		3	1			2		CONT R ₂ ← R ₂ + 1
12	7	0		1								JRNZF 7 NOP
13		2		3		3	1			3		CONT R ₃ ← R ₃ + 1
14	7	0		1								JR 7 NOP

Evaluarea întârzierilor în propagarea semnalelor

Unitatea de execuție

1	R μ l	Ceas → ieșire	15 ns
2	2901B	A, B → /G ₀ , /P ₀	50 ns
3	2902	/G ₀ , /P ₀ → C _{n+z}	10 ns
4	2901B	C _n → C _{n+4} , OVR, Y	29 ns
5	RCond, MUX	Intr → Ies	21 ns
6	2901B	RAM ₃ setup	16 ns
Total			141 ns

Unitatea de comandă

1	R μ l	Ceas → ieșire	15 ns
6	MUX	Selecție → ieșire	20 ns
7	2910	Condiții → ieșire	45 ns
8	PROM	Timpul de acces	55 ns
1	R μ l	Setup	05 ns
Total			140 ns

Evaluarea perioadei ceasului

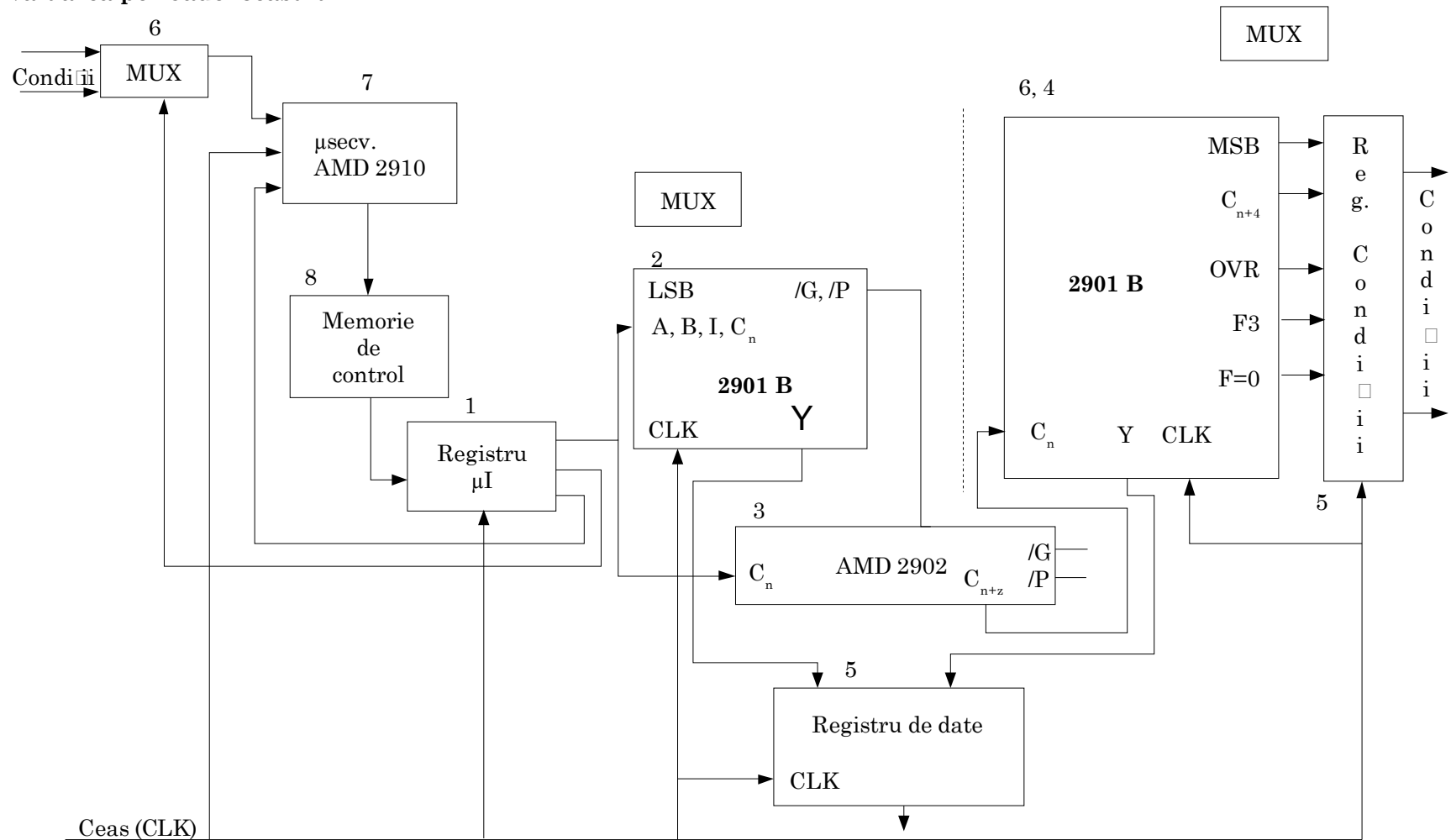


Figura 11.10. Evaluarea perioadei ceasului