

§ 3.3. Unitățile funcționale

- pe fiecare cale există un set de câte 4 unități funcționale (.L1, .S1, .M1, .D1 pentru calea de date A, iar .L2, .S2, .M2, .D2 pentru calea de date B);
- fiecare unitate funcțională are propriul său port de scriere de 32 de biți în cadrul setului de registre de uz general și două porturi de citire de 32 de biți pentru operanzii sursă *src1* și *scr2*;
- 4 unități (.L1, .L2, .S1 și .S2) au un port suplimentar de 8 biți pentru scrierea cuvintelor de 40 de biți, precum și o intrare de 8 biți pentru citirea cuvintelor de 40 de biți;
- deoarece au propriul lor port de scriere de 32 de biți toate unitățile pot fi utilizate în paralel în fiecare ciclu mașină;
- deoarece fiecare multiplicator poate furniza un rezultat de maximum 64 de biți, un port de scriere suplimentar a fost adăugat de la fiecare multiplicator la setul de registre.

- Unitățile funcționale .L și .S realizează o serie funcții aritmetice, logice și de salt cu rezultate disponibile în fiecare ciclu de tact. Funcțiile aritmetice și logice ale CPU permit o operație de 32 de biți, 2 operații de 16 biți sau 4 operații de 8 biți.
- Unitățile funcționale .M realizează toate operațiile de multiplicare. Fiecare dintre aceste unități poate realiza 2 multiplicări 16x16-biți sau 4 multiplicări 8x8-biți în fiecare ciclu de tact. De asemenea, unitățile .M pot realiza operații de multiplicare 16x32-biți, 2 multiplicări 16x16-biți împreună cu operații de adunare/scădere, 4 multiplicări 8x8 biți împreună cu operații de adunare. În plus, unitățile .M includ numărări de biți, rotații, multiplicări pentru grupuri Galois, hardware de deplasare bidirecțională a variabilelor.

§ 3.4. Seturile de registre pentru căile încrucișate

- Fiecare unitate funcțională citește direct din sau scrie direct în seturile de registre aferente căilor din care fac parte (.L1, .S1, .D1, .M1 scriu în setul de registre A, iar L2, .S2, .D2, .M2 scriu în setul de registre B). Seturile de registre sunt conectate la seturile de registre aferente unităților funcționale situate în partea opusă (în cealaltă cale) via căile încrucișate 1X și 2X. Aceste căi încrucișate permit unităților funcționale dintr-o cale să acceseze operandul de 32 de biți aflat în setul de registre din cealaltă cale. Astfel, calea încrucișată 1X permite unităților funcționale din calea A să citească sursa lor din setul de registre B, iar calea încrucișată 2X permite unităților funcționale din calea B să citească sursa lor din setul de registre A.

- Intrările *src2* ale unităților funcționale *.M1*, *.M2*, *.S1*, *.S2*, *.D1* și *.D2* sunt selectate folosind calea încrucișată și setul de registre de pe aceeași parte. În cazul unităților funcționale *.L1* și *.L2*, ambele intrări *src1* și *src2* sunt selectate folosind calea încrucișată și setul de registre de pe aceeași parte.
- Deoarece există două căi încrucișate se poate citi în fiecare ciclu de tact o sursă din fiecare set de registre aferent celeilalte căi sau două surse de pe căi diferite. De asemenea, două unități de pe aceeași cale pot citi simultan aceeași sursă de pe cealaltă cale via calea încrucișată.
- În cazul în care o instrucțiune necesită citirea conținutului unui registru via calea încrucișată care a fost încărcat în ciclu de tact anterior procesorul introduce în mod automat o întârziere. Ca urmare, nu este necesară introducerea de instrucțiuni NOP.

§ 3.5. Căile de memorare, încărcare și salvare

- Procesorul utilizează pentru încărcare și salvare cuvinte duble (de 32 de biți).
- Sunt 4 căi de 32 de biți pentru încărcarea datei din memorie în setul de registre. Pentru calea A, LD1a este calea de încărcare pentru cei mai puțin semnificativi 32 de biți (LSB), iar LD1b este calea de încărcare pentru cei mai semnificativi 32 de biți (MSB). Pentru calea B, LD2a este calea de încărcare pentru cei mai puțin semnificativi 32 de biți , iar LD2b este calea de încărcare pentru cei mai semnificativi 32 de biți.

- De asemenea, există, 4 căi de 32 de biți pentru salvarea valorilor din fiecare set de registre în memorie. Pentru calea A, ST1a este calea de scriere pentru cei mai puțin semnificativi 32 de biți, iar ST1b este calea de scriere pentru cei mai semnificativi 32 de biți. Pentru calea B, ST2a este calea de scriere pentru cei mai puțin semnificativi 32 de biți, iar ST2b este calea de scriere pentru cei mai semnificativi 32 de biți.
- Trebuie remarcat faptul că anumite porturi pentru operanzi lungi sau cuvinte duble sunt împărțite între diferite unități funcționale. De aceea, apare o constrângere care specifică care operație (cu operanzi lungi sau cu cuvinte duble) poate fi executată în același pachet.

§ 3.6. Căile adreselor de date

- În ambele căi de date căile adreselor de date DA1 și DA2 sunt conectate fiecare la unitățile funcționale D1 și D2. Aceasta permite adreselor de date generate în orice cale să acceseze data în sau din orice registru.
- Resursele DA1 și DA2 și căile de date asociate sunt specificate prin T1, respectiv T2. T1 constă în calea de adrese DA1 și căile de date LD1 și ST1. LD1 este alcătuit din LD1a și LD1b pentru a asigura o încărcare de 64 de biți; ST1 este alcătuit din ST1a și ST1b pentru a asigura o încărcare de 64 de biți. Similar T2 constă în calea de adrese DA2 și căile de date LD2 și ST2. LD2 este alcătuit din LD2a și LD2b pentru a asigura o încărcare de 64 de biți; ST2 este alcătuit din ST2a și ST2b pentru a asigura o încărcare de 64 de biți.

- T1 și T2 apar scrise în câmpurile aferente unităților funcționale pentru încărcarea și salvarea instrucțiunilor.

Exemplu: LDW .D1T2 [A0], B1

- instrucțiunea de încărcare utilizează unitatea funcțională .D1 pentru generarea adresei, dar este utilizată resursa căi LD2 din DA2 (specificată prin T2) pentru încărcarea datei în setul de registre B.

§ 3.7. Grupul Galois

- Sistemele de comunicații numerice moderne utilizează, în general, scheme de codare a corecției erorilor în scopul îmbunătățirii performanței sistemului în cazul unor canale de comunicații imperfecte. Cea mai răspândită schemă este codul Reed-Solomon, datorită robusteții sale față de erorile burst și a implementării facile.
- Procesorul conține un hardware de multiplicare pentru grupuri Galois care este utilizat pentru funcțiile de codare/decodare Reed-Solomon.

Aritmetica modulo 2

+	0	1
0	0	1
1	1	0

-	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

- Adunarea (scăderea) echivalente cu operația **XOR**.
- Înmulțirea echivalentă cu operația **AND**.

Aritmetica modulo 5

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

-	0	1	2	3	4
0	0	4	3	2	1
1	1	0	4	3	2
2	2	1	0	4	3
3	3	2	1	0	4
4	4	3	2	1	0

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

- Grupurile definite în acest fel se numesc **grupuri finite** sau **grupuri Galois** – GF(2), GF(5).
- Grupurile Galois sunt definite numai pentru o aritmetică modulo un număr prim.
- Grupurile Galois se pot forma cu elemente care sunt vectori în locul numerelor întregi dacă se utilizează polinoame.
- Sistemele de codare Reed-Solomon sunt grupuri Galois GF(2^m), care utilizează aritmetica binară. În acest caz adunarea se realizează utilizând operația XOR efectuată asupra elementelor vectorului, iar înmulțirea utilizând o combinație de operații AND și XOR.

Exemplu: GF(2³)

Addition

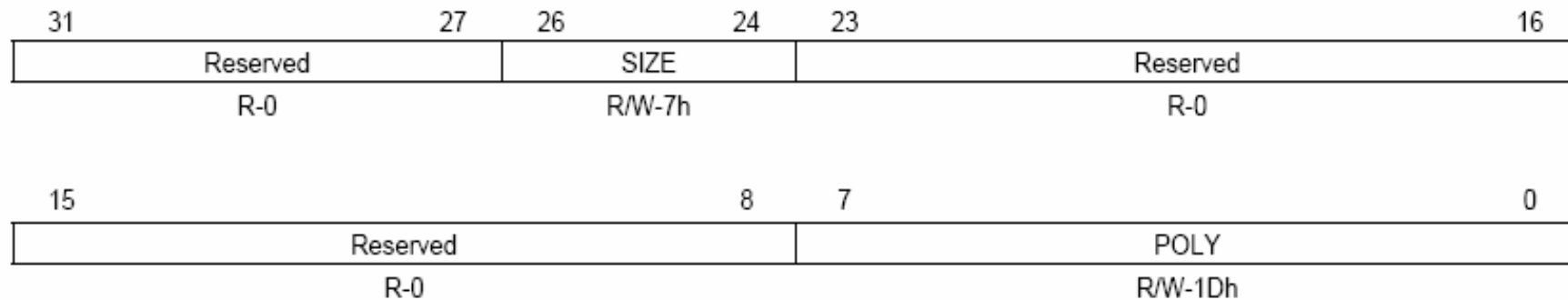
+	000	001	010	011	100	101	110	111
000	000	001	010	011	100	101	110	111
001	001	000	011	010	101	100	111	110
010	010	011	000	001	110	111	100	101
011	011	010	001	000	111	110	101	100
100	100	101	110	111	000	001	010	011
101	101	100	111	110	001	000	011	010
110	110	111	100	101	010	011	000	001
111	111	110	101	100	011	010	001	000

Multiplication

×	000	001	010	011	100	101	110	111
000	000	000	000	000	000	000	000	000
001	000	001	010	011	100	101	110	111
010	000	010	100	110	011	001	111	101
011	000	011	110	101	111	100	001	010
100	000	100	011	111	110	010	101	001
101	000	101	001	100	010	111	011	110
110	000	110	111	001	101	011	010	100
111	000	111	101	010	001	110	100	011

- Canalul de eroare este modelat ca un vector de biți, având valoarea 1 în fiecare poziție a vectorului în care apare eroare și valoarea 0 dacă nu apare eroare. În acest mod este determinat vectorul de eroare, care este scăzut din mesajul recepționat pentru determinarea cuvântului de cod corect.
- Multiplicatorul hardware pentru grupuri Galois se numește **GMPY4**. Instrucțiunea GMPY4 realizează 4 operații paralele pentru pachete de date de 8 biți în cadrul unității funcționale .M. Multiplicatorul pentru grupurile Galois poate fi programat pentru realizarea tuturor înmulțirilor aferente grupurilor $GF(2^m)$, unde $m = 1, 2, \dots, 8$, utilizând, în acest scop un generator de polinoame. Dimensiunea grupului și generatorul de polinoame este controlat prin registrul pentru generarea funcțiilor polinomiale pentru grupurile Galois – GFPGFR (Galois Field Polynomial Generator Function Register).

Figure 2-6. Galois Field Polynomial Generator Function Register (GFPGFR)



LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 2-10. Galois Field Polynomial Generator Function Register (GFPGFR) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
26-24	SIZE	0-7h	Field size.
23-8	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
7-0	POLY	0-FFh	Polynomial generator.

După inițializarea procesorului m (field size) = 7h, iar polinomul (polynomial generator) este 1Dh.