

ΕΣ 08 – Επεξεργαστές Ψηφιακών Σημάτων



Αρχιτεκτονική Επεξεργαστών Ψ.Ε.Σ

- Εισαγωγή
- Βασικά Χαρακτηριστικά
- Κύρια Δομικά Στοιχεία
- Δομή Εντολών Assembly
- Σύνολο εντολών για τους επεξεργαστές Cxxx

Βιβλιογραφία Ενότητας

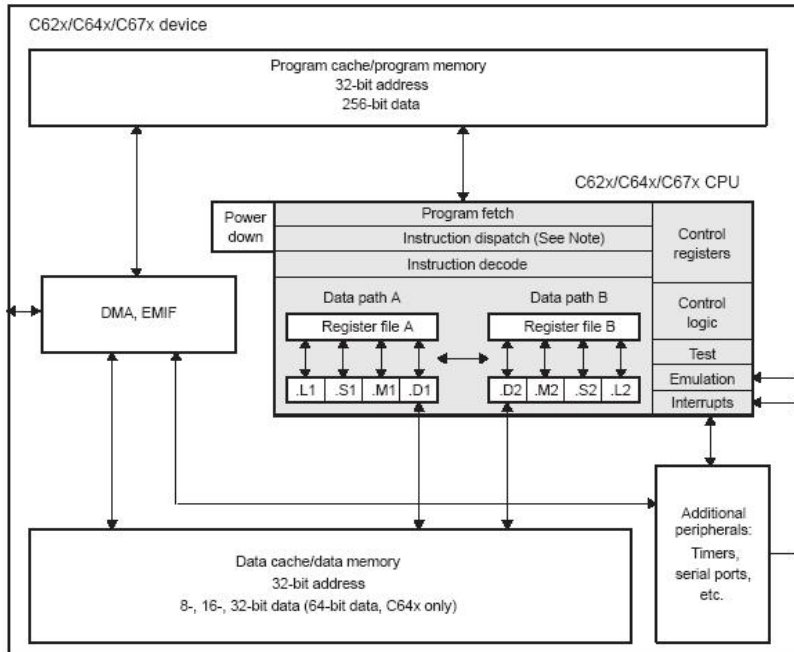


- ◇ Kehtarnavaz [2005]: Chapter 3
- ◇ Kuo [2005]: Chapters 1 & 4-5
- ◇ Lapsley [2002]: Chapter 4
- ◇ Hayes [2000]: Κεφάλαιο 8
- ◇ TMS320C6000 CPU and Instruction Set Reference Guide

★ Εισαγωγή

- Βασικά Χαρακτηριστικά
- Κύρια Δομικά Στοιχεία
- Δομή Εντολών Assembly
- Σύνολο εντολών για τους επεξεργαστές Cxxx

Εισαγωγή



Η γενική δομή της αρχιτεκτονικής Επεξεργαστών Ψ.Ε.Σ επιδεικνύεται στο διπλανό σχήμα (το σχήμα αφορά τους επεξεργαστές C62x, C64x, C67x αλλά είναι ενδεικτικό):

- › Η αρχιτεκτονική αυτή έχει σχεδιαστεί για βέλτιστη υλοποίηση της λειτουργίας FIR (Finite Impulse Response) filtering
- › Περιλαμβάνει δύο (ένα για εντολές, ένα για δεδομένα) ή τρεις (ένα για εντολές, δύο για δεδομένα) διαδρόμους δεδομένων (data buses) για εκτέλεση εντολών MAC (Multiply Accumulate) σε ένα κύκλο εντολής.

☑ Εισαγωγή

- ★ Βασικά Χαρακτηριστικά
- Κύρια Δομικά Στοιχεία
- Δομή Εντολών Assembly
- Σύνολο εντολών για τους επεξεργαστές Cxxx

Βασικά Χαρακτηριστικά Επεξεργαστών Ψ.Ε.Σ



- ◇ Οι περισσότεροι επεξεργαστές Ψ.Ε.Σ έχουν κάποια ιδιαίτερα χαρακτηριστικά που τους καθιστούν κατάλληλους για την διεκπεραίωση αριθμητικά απαιτητικών εφαρμογών. Τα χαρακτηριστικά αυτά είναι:
- ◇ Υλοποίηση λειτουργιών MAC (Multiply-Accumulate) σε ένα κύκλο εντολής. Πολλοί αλγόριθμοι Ψ.Ε.Σ έχουν τη δομή FIR φιλτραρίσματος δηλαδή υπολογίζουν ποσότητες $y(n)$ της μορφής:

$$y(n) = \sum_{i=0}^{L-1} b_i \cdot x(n-i) = [b_0 \quad b_1 \quad \dots \quad b_{L-1}] \cdot \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-L+1) \end{bmatrix}$$

Για την εκτέλεση MAC εντολών σε ένα κύκλο εντολής οι επεξεργαστές ακολουθούν την αρχιτεκτονική Harvard αντί την αρχιτεκτονική von Neumann που είναι περισσότερο διαδεδομένη στους επεξεργαστές γενικού σκοπού. Στην αρχιτεκτονική αυτή υπάρχει ξεχωριστή μνήμη για το πρόγραμμα και ξεχωριστή μνήμη για τα δεδομένα

- ☑ Εισαγωγή
- ★ Βασικά Χαρακτηριστικά
- ☐ Κύρια Δομικά Στοιχεία
- ☐ Δομή Εντολών Assembly
- ☐ Σύνολο εντολών για τους επεξεργαστές Cxxx

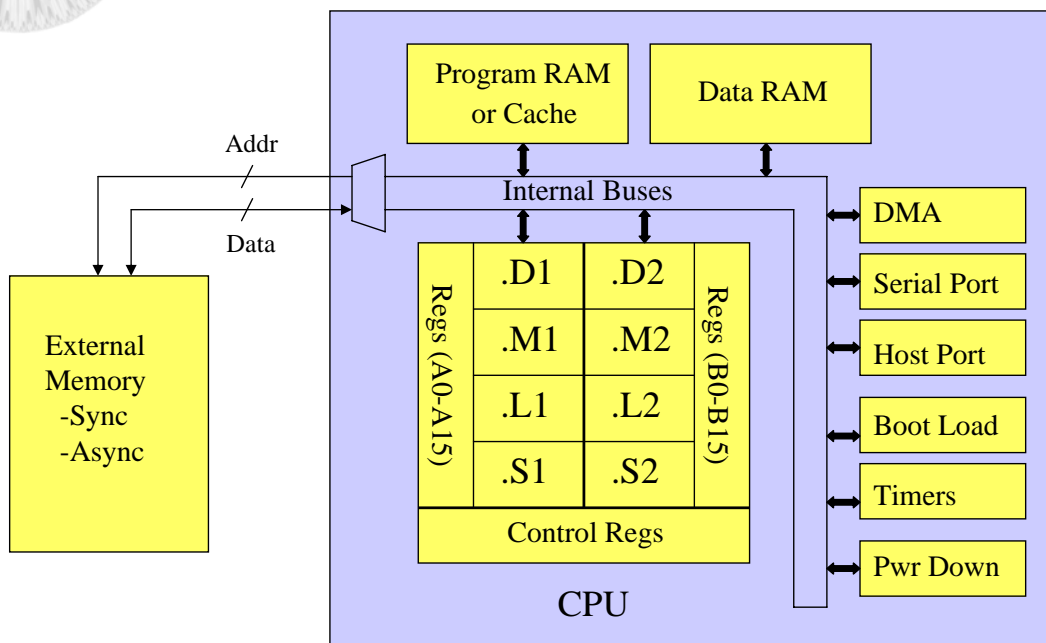
Βασικά Χαρακτηριστικά Επεξεργαστών Ψ.Ε.Σ (II)



- ◇ **Παράλληλισμός εκτέλεσης εντολών (pipelining)**. Εκτός από ξεχωριστή μνήμη για προγράμματα και δεδομένα οι επεξεργαστές Ψ.Ε.Σ υποστηρίζουν και ξεχωριστούς διαδρόμους δεδομένων για ανάκληση εντολών και δεδομένων. Με τον τρόπο αυτό υπάρχει δυνατότητα ταυτόχρονης πρόσβασης στη μνήμη για την λήψη των δύο ορισμάτων μιας αριθμητικής πράξης αλλά και την ανάκληση της επόμενης προς εκτέλεση εντολής.
- ◇ **Modulo-N πρόσβαση στη μνήμη**. Με τον τρόπο αυτό, για παράδειγμα, οι συντελεστές ενός FIR φίλτρου αποθηκεύονται σε διαδοχικά τμήματα μνήμης και έτσι δεν χρειάζεται να υπολογίζεται κάθε φορά η διεύθυνση του επόμενου συντελεστή. Απλά έχουμε ακολουθιακή πρόσβαση.
- ◇ **Ειδικός χειρισμός επαναληπτικών διαδικασιών (loops)**. Σε αρκετές περιπτώσεις υπάρχει υλοποίηση επαναληπτικών δομών με εξειδικευμένο υλικό.
- ◇ **Απευθείας πρόσβαση στη μνήμη και στις περιφερειακές συσκευές**. Δεν χρειάζεται η απασχόληση της CPU για πρόσβαση στη μνήμη και τις συσκευές αυτές.

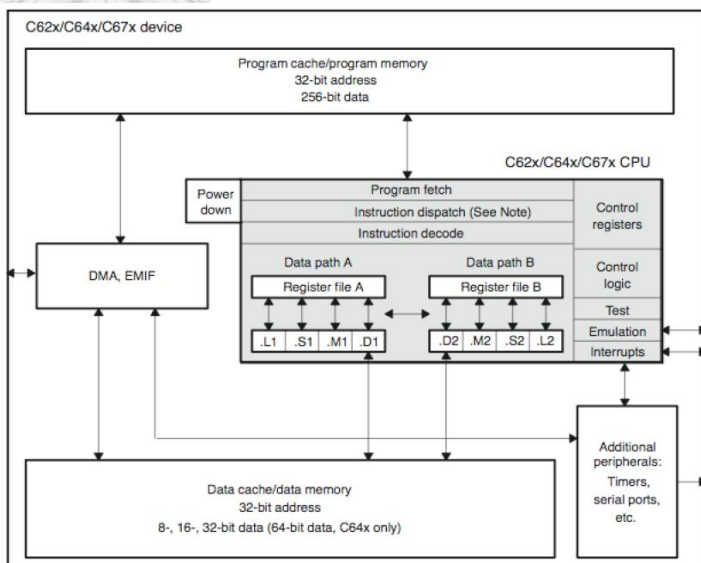
- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ★ Κύρια Δομικά Στοιχεία
- ☐ Δομή Εντολών Assembly
- ☐ Σύνολο εντολών για τους επεξεργαστές Cxxx

Κύρια Δομικά Στοιχεία



- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ★ Κύρια Δομικά Στοιχεία
- ☐ Δομή Εντολών Assembly
- ☐ Σύνολο εντολών για τους επεξεργαστές Cxxx

Κύρια Δομικά Στοιχεία (II)



Κάθε εντολή απαρτίζεται από 32 bits. Ο διάδρομος προγράμματος έχει εύρος 256 bits (32 bits διεύθυνση)

Επομένως μπορούν να μεταφερθούν και να εκτελεστούν 8 εντολές (σε συνεχόμενες διευθύνσεις μνήμης) σε ένα κύκλο ρολογιού (συχνότητα επεξεργαστή 225MHz, άρα διάρκεια κάθε κύκλου ρολογιού 4.8ns).

Υπάρχουν 8 ανεξάρτητες μονάδες σε κάθε επεξεργαστή της σειράς TMS320C6xxx:

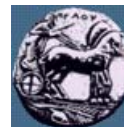
- 2 πολλαπλασιαστές
 - 6 ALUs (Αριθμητική και Λογική Μονάδα)
- Αποδοτικός κώδικας σημαίνει και οι 8 μονάδες να βρίσκονται διαρκώς σε εργασία.

Υπάρχουν 32 συνολικά καταχωρητές στις δύο ΚΜΕ (Κεντρική Μονάδα Επεξεργασίας - Data path A and data path B) 16 σε κάθε μια (A0-A15, B0-B15) με εύρος 32 bits.

Το μήκος λέξης δεδομένων είναι 32 bit (εξαιρέση αποτελεί η σειρά επεξεργαστών C64xx στην οποία έχουμε μήκος λέξης 64 bit) ενώ πρόσβαση στη μνήμη δεδομένων γίνεται μέσω διευθύνσεων από 32 bits (4GB συνολικός χώρος μνήμης).

- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ★ Κύρια Δομικά Στοιχεία
- ☐ Δομή Εντολών Assembly
- ☐ Σύνολο εντολών για τους επεξεργαστές Cxxx

Διάδρομοι Δεδομένων και Προγράμματος



- ◇ Οι διάδρομοι δεδομένων και προγράμματος υποδιαιρούνται σε επιμέρους διαδρόμους ως εξής:
- ◇ Διάδρομος δεδομένων
 - ◇ Διάδρομος διεύθυνσης ανάγνωσης δεδομένων (Data Read Address Bus), ο οποίος μεταφέρει τη διεύθυνση από την οποία θα διαβαστούν τα δεδομένα
 - ◇ Διάδρομος ανάγνωσης δεδομένων (Data Read Bus), ο οποίος μεταφέρει τα δεδομένα από τη μνήμη στη ΚΜΕ
 - ◇ Διάδρομος διεύθυνσης εγγραφής δεδομένων (Data Write Address Bus), ο οποίος μεταφέρει τη διεύθυνση στην οποία θα εγγραφούν τα δεδομένα
 - ◇ Διάδρομος εγγραφής δεδομένων (Data Write Address Bus), ο οποίος μεταφέρει τα δεδομένα που θα εγγραφούν στη μνήμη.
- ◇ Διάδρομος Προγράμματος
 - ◇ Διάδρομος διεύθυνσης ανάγνωσης προγράμματος (Program Address Bus), ο οποίος μεταφέρει τη διεύθυνση από την οποία θα διαβαστούν τα δεδομένα προγράμματος (π.χ εντολές)
 - ◇ Διάδρομος ανάγνωσης προγράμματος (Program Read Bus), ο οποίος μεταφέρει τα δεδομένα προγράμματος (π.χ εντολές) από τη μνήμη στη ΚΜΕ

- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ☑ Κύρια Δομικά Στοιχεία
- ★ Δομή Εντολών Assembly
- ☐ Σύνολο εντολών για τους επεξεργαστές Cxxx

Δομή εντολών ASSEMBLY



- ◇ Ένα πρόγραμμα σε κώδικα ASSEMBLY είναι ένα αρχείο κειμένου (ASCII text file) του οποίου κάθε γραμμή μπορεί να περιλαμβάνει έως και 7 στοιχεία:
 - ☐ Ετικέτα (Label)
 - ☐ Παράλληλες μπάρες (Parallel bars)
 - ☐ Συνθήκη (Condition)
 - ☐ Εντολή ή οδηγία (Instruction or Directive)
 - ☐ Μονάδα εκτέλεσης (Functional unit)
 - ☐ Ορίσματα (Operands)
 - ☐ Σχόλια (Comments).

```
label: parallel bars [condition] instruction unit operands ; comments
```

- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ☑ Κύρια Δομικά Στοιχεία
- ★ Δομή Εντολών Assembly
- ☐ Σύνολο εντολών για τους επεξεργαστές Cxxx

Ετικέτες



```
label: parallel bars [condition] instruction unit operands ; comments
```

- ◇ Μια ετικέτα καθορίζει μια γραμμή κώδικα και αναπαριστά μια μεταβλητή ή διεύθυνση μνήμης η οποία περιέχει είτε μια εντολή είτε δεδομένα

```
.global _sum
_sum:
ZERO .L1 A9          ;Sum register
MV .L1 B4,A2        ;initialize counter with passed argument

loop: LDH .D1 *A4++, A7 ;load value pointed by A4 into register A7
      SUB .L1 A2,1,A2  ;decrement counter
      B .S1 loop      ;branch back to loop
      NOP 3
      ADD .L1 A7,A9,A9 ;A9 += A7
      NOP 2

      MV .L1 A9,A4     ;move result into return register A4
      B .S2 B3        ;branch back to address stored in B3
      NOP 5
```

- ◇ Οι ετικέτες πρέπει να ικανοποιούν τις επόμενες συνθήκες:
 - ◇ Ο πρώτος χαρακτήρας μπορεί να είναι γράμμα ή κάτω παύλα (underscore _) ακολουθούμενη από γράμμα.
 - ◇ Ο πρώτος χαρακτήρας πρέπει να βρίσκεται στην πρώτη στήλη του αρχείου (text file).
 - ◇ Μπορούν να περιλαμβάνουν μέχρι 32 αλφαριθμητικούς χαρακτήρες.



- Εισαγωγή
- Βασικά Χαρακτηριστικά
- Κύρια Δομικά Στοιχεία
- Δομή Εντολών Assembly
- Σύνολο εντολών για τους επεξεργαστές Cxxx

Παράλληλες μπάρες

label: **parallel bars** [condition] instruction unit operands ; comments

- ◇ Μια εντολή η οποία εκτελείται σε παραλληλία με την προηγούμενή της δηλώνεται με παράλληλες μπάρες (||). Το σχετικό πεδίο παραμένει κενό αν η εν λόγω εντολή δεν εκτελείται παράλληλα με την προηγούμενή της

- ◇ Στο συγκεκριμένο παράδειγμα οι εντολές LDH και SUB εκτελούνται παράλληλα

```

.global _sum

_sum:
    ZERO    .L1 A9          ;Sum register
    MV      .L1 B4,A2       ;initialize counter with passed argument

loop:    LDH  .D1 *A4++, A7  ;load value pointed by A4 into register A7
    ||[A2] SUB .L1 A2,1,A2  ;decrement counter
    [A2] B   .S1 loop      ;branch back to loop
    NOP 3
    ADD .L1 A7,A9,A9       ;A9 += A7
    NOP 2

    MV      .L1 A9,A4       ;move result into return register A4
    B      .S2 B3          ;branch back to address stored in B3
    NOP 5
    
```



- Εισαγωγή
- Βασικά Χαρακτηριστικά
- Κύρια Δομικά Στοιχεία
- Δομή Εντολών Assembly
- Σύνολο εντολών για τους επεξεργαστές Cxxx

Συνθήκη

label: parallel bars **[condition]** instruction unit operands ; comments

- ◇ Πέντε καταχωρητές χρησιμοποιούνται για την εκτέλεση εντολών υπό συνθήκη: A1, A2, B0, B1, και B2.

- ◇ Αν δεν υπάρχει συνθήκη πριν από μια εντολή η εντολή εκτελείται πάντοτε.

Στο συγκεκριμένο παράδειγμα η εντολή branch (B) εκτελείται εφόσον ο καταχωρητής A2 είναι διάφορος του μηδέν

```

.global _sum

_sum:
    ZERO    .L1 A9          ;Sum register
    MV      .L1 B4,A2       ;initialize counter with passed argument

loop:    LDH  .D1 *A4++, A7  ;load value pointed by A4 into register A7
    |[A2] SUB .L1 A2,1,A2  ;decrement counter
    [A2] B   .S1 loop      ;branch back to loop
    NOP 3
    ADD .L1 A7,A9,A9       ;A9 += A7
    NOP 2

    MV      .L1 A9,A4       ;move result into return register A4
    B      .S2 B3          ;branch back to address stored in B3
    NOP 5
    
```

With this condition ...	The instruction executes if ...	With this condition ...	The instruction does not execute if ...
[A1]	A1 != 0	[A1]	A1 = 0
[!A1]	A1 = 0	[!A1]	A1 != 0

- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ☑ Κύρια Δομικά Στοιχεία
- ★ Δομή Εντολών Assembly
- ☐ Σύνολο εντολών για τους επεξεργαστές Cxxx

Εντολή ή οδηγία



label: parallel bars [condition] **instruction** unit operands ; comments

- ◇ Οι εντολές επεξεργαστή (processor mnemonics) είναι οι πραγματικές εντολές που εκτελούνται από το πρόγραμμα.

```

;FIRCSMfunc.asm ASM function called from C to implement FIR
;A4 = Samples address, B4 = coeff address, A6 = Filter order
;Delays organized as: x(n-(N-1)) ... x(n),coeff as h[0]...h[N-1]
.def      _fircasmfunc
_fircasmfunc:
;ASM function called from C
MV      A6,A1      ;setup loop count
MPY     A6,2,A6    ;since dly buffer data as byte
ZERO   A8         ;init A8 for accumulation
ADD     A6,B4,B4   ;since coeff buffer data as byte
SUB     B4,1,B4    ;B4=bottom coeff array h[N-1]
loop:   ;start of FIR loop
LDH     *A4++,A2   ;A2=x[n-(N-1)+i] i=0,1,...,N-1
LDH     *B4--,B2   ;B2=h[N-1-i] i=0,1,...,N-1
NOP     4
MPY     A2,B2,A6   ;A6=x[n-(N-1)+i]*h[N-1-i]
NOP
ADD     A6,A8,A8   ;accumulate in A8
LDH     *A4,A7     ;A7=x[(n-(N-1)+i+1]update delays
NOP     4          ;using data move "up"
STH     A7,*-A4[1] ;-->x[(n-(N-1)+i] update sample
SUB     A1,1,A1    ;decrement loop count
[A1] B  loop      ;branch to loop if count # 0
NOP     5

MV      A8,A4     ;result returned in A4
B       B3        ;return addr to calling routine
NOP     4

```

- ◇ Οι οδηγίες (assembler directives) είναι εντολές προς τον συμβολομεταφραστή (assembler) οι οποίες καθορίζουν τη διαδικασία μετατροπής σε κώδικα μηχανής και ορίζουν τις δομές δεδομένων (σταθερές και μεταβλητές) που χρησιμοποιούνται από το πρόγραμμα.

- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ☑ Κύρια Δομικά Στοιχεία
- ★ Δομή Εντολών Assembly
- ☐ Σύνολο εντολών για τους επεξεργαστές Cxxx

Εντολή ή οδηγία (II)



- ◇ Στον επόμενο πίνακα δίνονται μερικά παραδείγματα οδηγιών

Directives	Description
.sect "name"	Creates section of information (data or code)
.double value	Reserve two consecutive 32 bits (64 bits) in memory and fill with double-precision (64-bit) IEEE floating-point representation of specified value
.float value	Reserve 32 bits in memory and fill with single-precision (32-bit) IEEE floating-point representation of specified value
.int value	Reserve 32 bits in memory and fill with specified value
.long value	
.word value	
.short value	Reserve 16 bits in memory and fill with specified value
.half value	
.byte value	Reserve 8 bits in memory and fill with specified value

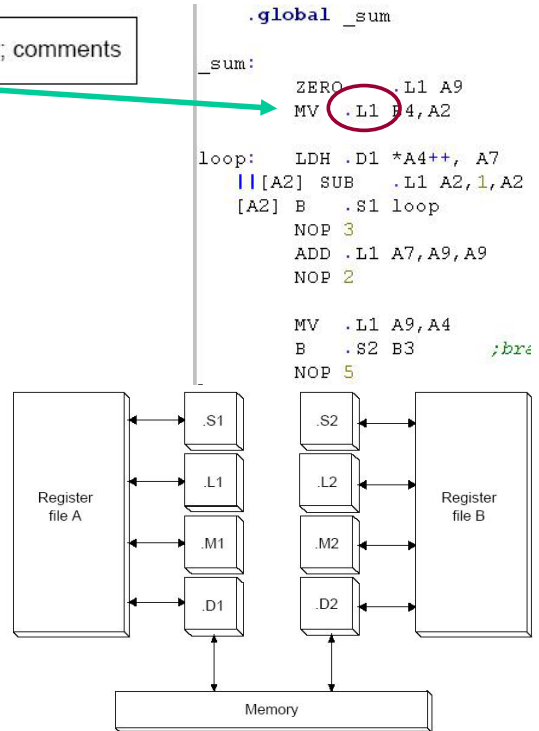


- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ☑ Κύρια Δομικά Στοιχεία
- ★ Δομή Εντολών Assembly
- ☐ Σύνολο εντολών για τους επεξεργαστές Cxxx

Μονάδα εκτέλεσης

label: parallel bars [condition] instruction **unit** operands ; comments

- ◇ Δύο ΚΜΕ (Data Paths), 1->A και 2->B, με τέσσερις μονάδες έκαστη:
 - ◇ Πολλαπλασιασμός (.M1, .M2)
 - ◇ Λογικές και Αριθμητικές πράξεις - ALU (.L1, .L2)
 - ◇ Διακλάδωση, χειρισμός bit και αριθμητικές πράξεις (.S1, .S2)
 - ◇ Ανάκληση και αποθήκευση δεδομένων και εντολών καθώς και αριθμητικές πράξεις (.D1, .D2)



- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ☑ Κύρια Δομικά Στοιχεία
- ★ Δομή Εντολών Assembly
- ☐ Σύνολο εντολών για τους επεξεργαστές Cxxx

Ορίσματα

- ◇ Τα ορίσματα των εντολών είναι θέσεις μνήμης ή καταχωρητές. Για τα ορίσματα ισχύουν τα παρακάτω:
 - ◇ Όλες οι εντολές απαιτούν ένα όρισμα προορισμού (destination operand)
 - ◇ Οι περισσότερες εντολές έχουν ένα ή δύο ορίσματα εισόδου
 - ◇ Το όρισμα προορισμού πρέπει να ανήκει στην ίδια ΚΜΕ με ένα τουλάχιστον από τα ορίσματα εισόδου
 - ◇ Ένα όρισμα εισόδου μπορεί να προέρχεται από διαφορετική ΚΜΕ στο πλαίσιο ενός μόνο Execution Packet (δηλαδή ενός συνόλου εντολών που εκτελούνται παράλληλα)

```

.global _sum
_sum:
    ZERO .L1 A9 ;Sum register
    MV .L1 B4, A2 ;initialize counter with passed argument
loop:
    LDH .D1 *A4++, A7 ;load value pointed by A4 into register A7
    |[A2] SUB .L1 A2, 1, A2 ;decrement counter
    [A2] B .S1 loop ;branch back to loop
    NOP 3
    ADD .L1 A7, A9, A9 ;A9 += A7
    NOP 2

    MV .L1 A9, A4 ;move result into return register A4
    B .S2 B3 ;branch back to address stored in B3
    NOP 5
    
```


- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ☑ Κύρια Δομικά Στοιχεία
- ★ Δομή Εντολών Assembly
- ☐ Σύνολο εντολών για τους επεξεργαστές Cxxx



Assembly συναρτήσεις καλούμενες από C

```
#include <stdio.h>

extern sum();

void main()
{
    int i,ret;
    short *point;

    point = (short *) 0x00000000;

    printf("BEGIN: Assembly sum\n");

    for(i=0;i<10;i++)
    {
        printf("[%d] %d\n",i, point[i]);
    }

    ret = sum(point,10);

    printf("Sum = %d\n",ret);

    printf("END\n");
}
```

```
.global _sum
_sum:
    ZERO .L1 A1,A2
    MV .L1 B4,A2
loop:
    LDH .D1 *A4++,A7
    NOP 4
    ADD .L1 A7,A7,A9
    [A2] SUB .L1 A2,1,A2
    [A2] B .S1 loop
    NOP 5
    MV .L1 A9,A4
    B .S2 B3
    NOP 5
```

- ◇ Τα ορίσματα των συναρτήσεων μεταφέρονται μέσω των καταχωρητών A4, B4, A6, B6, ... με τη σειρά που δίνεται
 - ◇ Το αποτέλεσμα της κλήσης της συνάρτησης επιστρέφεται μέσω του καταχωρητή A4.
 - ◇ Η διεύθυνση επιστροφής από τον κώδικα στον οποίο έχει γίνει η κλήση της συνάρτησης δίνεται στον καταχωρητή B3. Οπότε πρέπει να δίνεται μεγάλη προσοχή να μην χρησιμοποιείται για άλλο σκοπό ο B3 εντός της συνάρτησης ASSEMBLY
- Ονοματολογία (βλέπε προηγ. διαφάνεια):

Στον κώδικα C: **label** (η συνάρτηση δηλώνεται ως external)
 Στον κώδικα ASSEMBLY: **_label** (η συνάρτηση δηλώνεται ως global)
 Πρόσβαση σε global μεταβλητές εντός του κώδικα ASSEMBLY: **.ref _variablename**

- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ☑ Κύρια Δομικά Στοιχεία
- ☑ Δομή Εντολών Assembly
- ★ Σύνολο εντολών για τους επεξεργαστές Cxxx



Σύνολο εντολών για τους επεξεργαστές TMS320C6xxx

.L unit	.M unit	.S unit	.D unit
ABS2	AVG2	SHFL	ADD2
ADD2	AVGU4	SMPY2	ADDKPC
ADD4	BITC4	SSHVL	AND
AND	BITR	SSHVR	ANDN
ANDN	DEAL	XPND2	BDEC
MAX2	DOTP2	XPND4	XOR
MAXU4	DOTPN2		LDLDDW
MIN2	DOTPNRSU2		LDNDW
MINU4	DOTPNRUS2		LDNW
MVK	DOTPRSU2		MVK
OR	DOTPRUS2		OR
PACK2	DOTPSU4		STDW
PACKH2	DOTPUS4		STNDW
PACKH4	DOTPU4		STNDW
PACKHL2	GMPY4		STNWD
PACKL4	MPY2		STNWD
PACKLH2	MPYHI		SUB2
SHLMB	MPYIH		XOR
SHRMB	MPYHIR		
SUB2	MPYIHR		
SUB4	MPYLI		
SUBABS4	MPYIL		
SWAP2	MPYLIR		
SWAP4	MPYILR		
UNPKHU4	MPYSU4		
UNPKLU4	MPYUS4		
XOR	MPYU4		
	MVD		
	ROTL		

Στο διπλανό πίνακα φαίνονται οι εντολές (για αριθμητική σταθερής υποδιαστολής - fixed point instructions) ASSEMBLY που μπορούν να εκτελεστούν σε κάθε μονάδα (.L,.M,.S,.D) της σειράς επεξεργαστών TMS320C6xxx.

Μπορούμε να κατηγοριοποιήσουμε τις εντολές αυτές σε τρεις κατηγορίες:

- ◇ Αριθμητικές (Πρόσθεση / Αφαίρεση / Πολλαπλασιασμός)
 Παράδειγμα:
ADD .L1 A3,A7,A7 ; A3+A7->A7
- ◇ Ανάκλησης - Αποθήκευσης
 Παράδειγμα:
STW .D2 A1,*+A4[20] ; store A1->(A4+20)
- ◇ Διακλάδωσης - Μετακίνησης
 Παράδειγμα:
[A1] B .S2 Loop ; branch to Loop if A1≠0

- ☑ Εισαγωγή
- ☑ Βασικά Χαρακτηριστικά
- ☑ Κύρια Δομικά Στοιχεία
- ☑ Δομή Εντολών Assembly
- ★ Σύνολο εντολών για τους επεξεργαστές Cxxx

Floating Point Instructions



.L Unit	.M Unit	.S Unit	.D Unit
ADDDP	MPYDP	ABSDP	ADDAD
ADDSP	MPYI	ABSSP	LDDW
DPINT	MPYID	CMPEQDP	
DPSP	MPYSP	CMPEQSP	
DPTRUNC		CMPGTDP	
INTDP		CMPGTSP	
INTDPU		CMPLTDP	
INTSP		CMPLTSP	
INTSPU		RCPDP	
SPINT		RCPSP	
SPTRUNC		RSQRDP	
SUBDP		RSQRSP	
SUBSP		SPDP	

Στο διπλανό πίνακα φαίνονται οι εντολές (για αριθμητική κινητής υποδιαστολής – floating point instructions) ASSEMBLY που μπορούν να εκτελεστούν σε κάθε μονάδα (.L,.M,.S,.D) της σειράς επεξεργαστών TMS320C62xx και TMS320C67xx.

Οι εντολές αυτές είναι κυρίως αριθμητικές δεδομένου ότι δεν υπάρχει κάποια σημαντική διαφορά ανάμεσα στις εντολές σταθερής και κινητής υποδιαστολής όσον αφορά την αποθήκευση και τη μετακίνηση - διακλάδωση