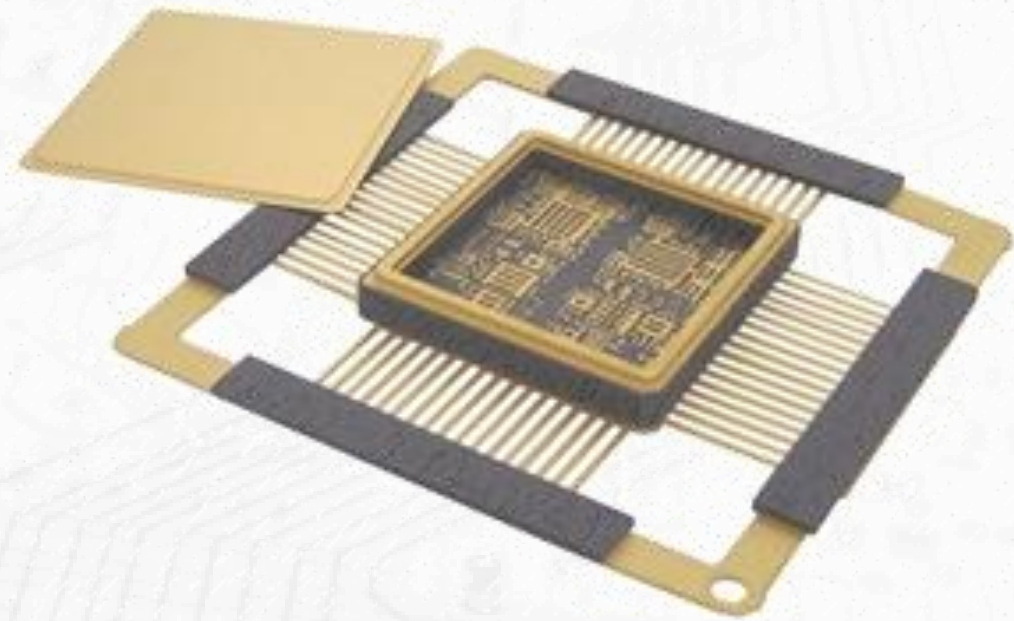




Μάθημα 3^ο

Ανασκόπηση Μικροελεγκτών

Ενσωματωμένα Συστήματα

Διδάσκον: Γρηγόριος Δουμένης

Διδακτικό Υλικό: Γρηγόριος Δουμένης, Νικόλαος Γιαννακέας, Ιωάννης Μασκλαβάνος

Αριθμητικά συστήματα

- Η βάση ενός αριθμού είναι ένας **συμβολισμός** μέσω του οποίου τα σύμβολα εκλαμβάνουν **διαφορετικές αριθμητικές τιμές**
- Ανάλογα με την βάση του αριθμητικού συστήματος το «11» σημαίνει:
 - Έντεκα για αριθμητικό σύστημα με βάση δέκα
 - Τρία για αριθμητικό σύστημα με βάση το δύο
 - Δεκαεπτά για αριθμητικό σύστημα με βάση το δεκαέξι
 - ...
- Η τιμή ενός αριθμού στο αριθμητικό σύστημα με βάση του δέκα αναπαρίσταται ως εξής:

$$1234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$$

Αριθμητικά συστήματα

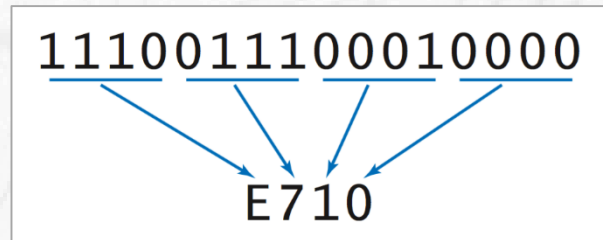
Δεκαεξαδικό

- Γιατί Δεκαεξαδικό αριθμητικό σύστημα?
 - Η βασική δυσκολία που ανακύπτει από το δυαδικό σύστημα απορρέει από το γεγονός ότι **το δυαδικό σύστημα είναι «φλύαρο»**
 - Για να αναπαρασταθεί ο αριθμός 202_{10} στο αριθμητικό σύστημα με βάση το 2 (δυαδικός) απαιτούνται 8 δυαδικά ψηφία
 - Ενώ το δεκαδικό σύστημα τον αναπαριστά μόνο με 3 δεκαδικά ψηφία και άρα περισσότερο συμπτυγμένα
- Το δεκαεξαδικό σύστημα αίρει την «φλυαρία» του δυαδικού και προσφέρει δύο χρήσιμα χαρακτηριστικά
 - Οι αριθμοί αναπαρίστανται ιδιαίτερα συμπτυγμένοι
 - Η μετατροπή από το δεκαεξαδικό αριθμητικό σύστημα στο δυαδικό είναι ιδιαίτερα εύκολη και υλοποιήσιμη

Αριθμητικά συστήματα

Δεκαεξαδικό

- Χαρακτηριστικά Δεκαεξαδικού Αριθμητικού Συστήματος
Σετ Συμβόλων {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
- Η μετατροπή της αναπαράστασης σε δεκαεξαδικό αριθμητικό σύστημα από δυαδικό αριθμητικό σύστημα είναι **άμεση**
- Παράδειγμα μετατροπής Bin2Hex (Binary to Hexadimal):



- Παράδειγμα μετατροπής Hex2Bin: 0x0ABCD

0000 1010 1011 1100 1101

Μετατροπές στα αριθμητικά συστήματα

Το σύστημα με
το οποίο είμαστε
εξοικειωμένοι

Δεκαδικό

Δυαδικό

Γλώσσα μηχανής

Μετατροπές
από Σύστημα
σε σύστημα

Δεκαεξαδικό

Το σύστημα
διευθυνσιοδότηση
μνήμης

2αδικό $\Rightarrow$ 10αδικό

2αδικό $\Rightarrow$ 16αδικό

10αδικό $\Rightarrow$ 2αδικό

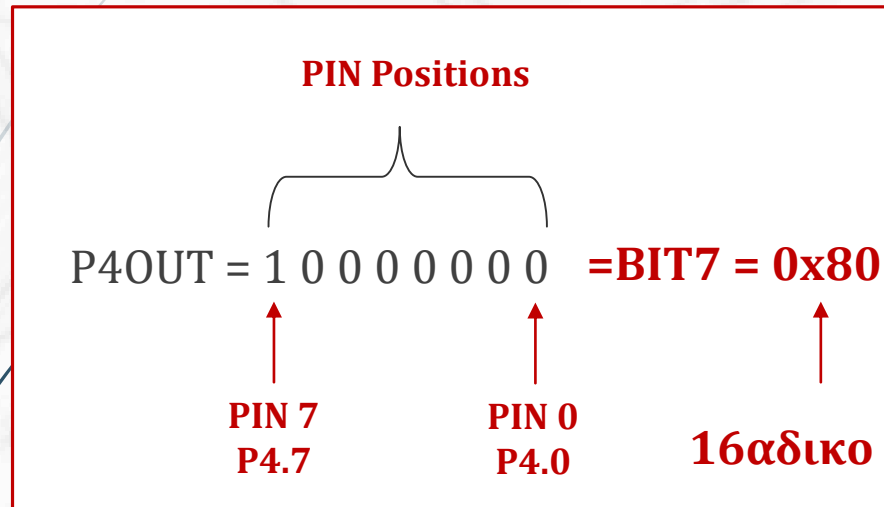
10αδικό $\Rightarrow$ 16αδικό

16αδικό $\Rightarrow$ 2αδικό

16αδικό $\Rightarrow$ 10αδικό

BITWISE SYSTEM

C Programming



P4OUT &= BIT1



Logical AND

P4OUT = P4OUT & BIT1 = 1 0 0 0 0 0 1 0

~BIT0 = 1 1 1 1 1 1 1 0

P4OUT ^= BIT1



Logical XOR

P4OUT = P4OUT ^ BIT1 = 1 0 0 0 0 0 1 0

P4OUT = P4OUT ^ BIT1 = 1 0 0 0 0 0 1 0

	1 st Loop	2 nd Loop	
	1 0 0 0 0 0 0 0	1 0 0 0 0 0 1 0	
XOR	0 0 0 0 0 0 1 0	0 0 0 0 0 0 1 0	1 st Loop
	1 0 0 0 0 0 1 0	1 0 0 0 0 0 0 0	2 nd Loop

Bitwise System vs. DriverLib

```
#include <msp430.h>
```

```
int main(void) {  
    WDTCTL = WDTPW | WDTHOLD;    // Stop watchdog timer  
    P4DIR |= 0x80;                // Set P4.7 to output direction  
  
    for(;;) {  
        volatile unsigned int i; // volatile to prevent optimization  
  
        P4OUT ^= 0x80;           // Toggle P4.7 using exclusive-OR  
  
        i = 60000;               // SW Delay(~1sec)  
        do i--;  
        while(i != 0);  
    }  
  
    return 0;  
}
```

```
#include <msp430.h>
```

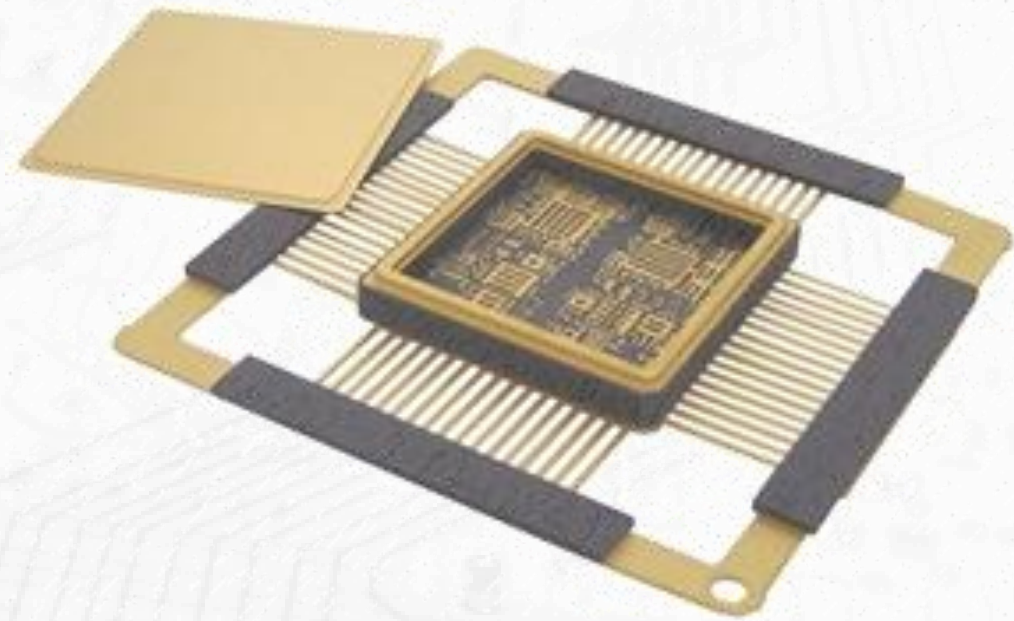
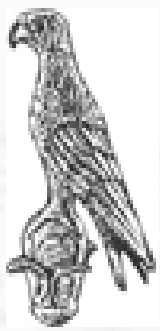
```
void main(void) {  
    WDTCTL = WDTPW | WDTHOLD  
    P4DIR |= BIT7;           //P4.7 (Grn) -> Output  
  
    P4OUT &= ~BIT7;          //Green LED -> OFF  
  
    while(1) {  
        volatile unsigned long i;  
  
        P4OUT ^= BIT7;  
  
        for(i = 0; i < 20000; i++); //delay  
    }  
}
```

```
include "driverlib.h"
```

```
#define RED_LED    GPIO_PORT_P1,GPIO_PIN0  
#define DELAY_CYCLES 1045000  
//----- main -----  
void main(void)  
{  
    // setup  
    WDT_A_hold(WDT_A_BASE); // stop watchdog timer  
  
    GPIO_setAsOutputPin(RED_LED); // set red LED to output  
  
    for(;;){ // loop  
        GPIO_setOutputHighOnPin(RED_LED); // turn red LED on  
        _delay_cycles(DELAY_CYCLES);      // delay a while  
        GPIO_setOutputLowOnPin(RED_LED);  // turn red LED off  
        _delay_cycles(DELAY_CYCLES);      // delay a while  
    }  
}
```

Bitwise manipulation

DriverLib



Ανασκόπηση Αρχιτεκτονικής Η/Υ

Συστήματα Μικροεπεξεργαστών

Διδάσκων: Γρηγόριος Δουμένης

Διδακτικό Υλικό: Νικόλαος Γιαννακέας, Γρηγόριος Δουμένης

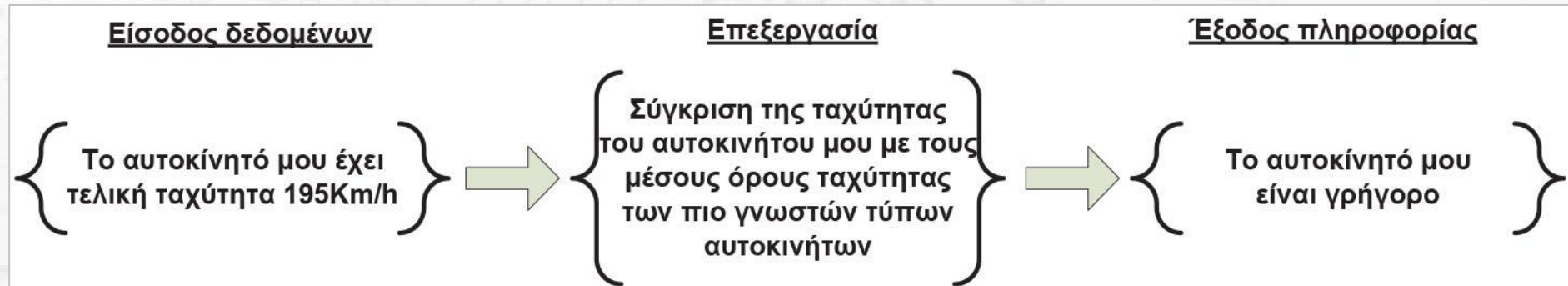
Ανασκόπηση Αρχιτεκτονικής Υπολογιστών

1. Δομή Υπολογιστή
2. Επίπεδα Συστήματος
3. Δομή μικροεπεξεργαστή
 1. Καταχωρητές Γενικής Χρήσης
 2. Καταχωρητές ειδικής Χρήσης
 3. Αριθμητική λογική μονάδα (ALU)
 4. Μνήμη Cache
 5. Κεντρική Μνήμη
4. Δομή μικροελεγκτή

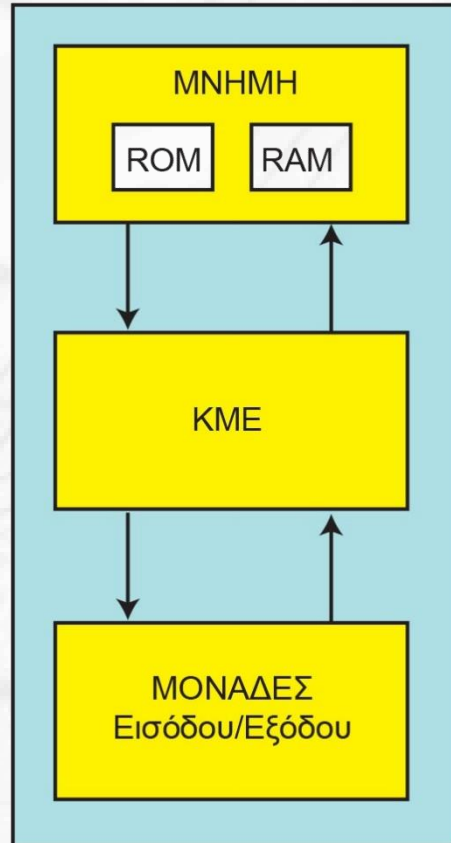
Φιλοσοφία λειτουργίας υπολογιστικού συστήματος



Παράδειγμα επεξεργασίας



Δομή Υπολογιστή

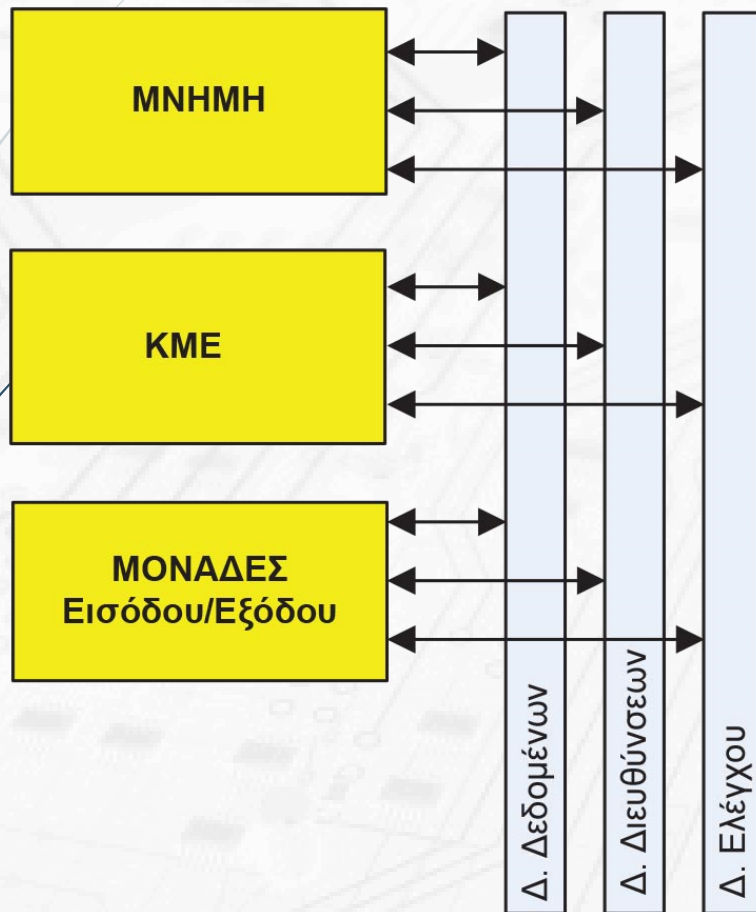


Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ): Η ΚΜΕ υλοποιείται σε ένα ολοκληρωμένο ψηφιακό κύκλωμα (IC) που ονομάζεται **μικροεπεξεργαστής**. Η αποστολή του μικροεπεξεργαστή είναι η εκτέλεση των εντολών που απαρτίζουν το πρόγραμμα.

Μνήμη: Η μνήμη φιλοξενεί τα προγράμματα (εντολές) καθώς και τα δεδομένα που αυτά διαχειρίζονται.

Μονάδες Εισόδου/Εξόδου: Οι μονάδες αυτές εξασφαλίζουν τόσο την αλληλεπίδραση του υπολογιστή με τον «έξω κόσμο» (π.χ. πληκτρολόγιο, οθόνη, εκτυπωτής), όσο και με άλλα βασικά συστατικά όπως τα μέσα αποθήκευσης (π.χ. σκληρός δίσκος), κλπ.

Σύστημα διαύλων



Δίαυλος Δεδομένων (Data Bus)
μεταφορά των δεδομένων (π.χ. μεταξύ μικροεπεξεργαστή και μνήμης)

Δίαυλος Διευθύνσεων (Address Bus)
ενεργοποίηση της τοποθεσίας που θα μεταφερθούν τα δεδομένα

Δίαυλος Ελέγχου (Control bus)
συντονισμός μονάδων συστήματος

Βασικές υπομονάδες ΚΜΕ

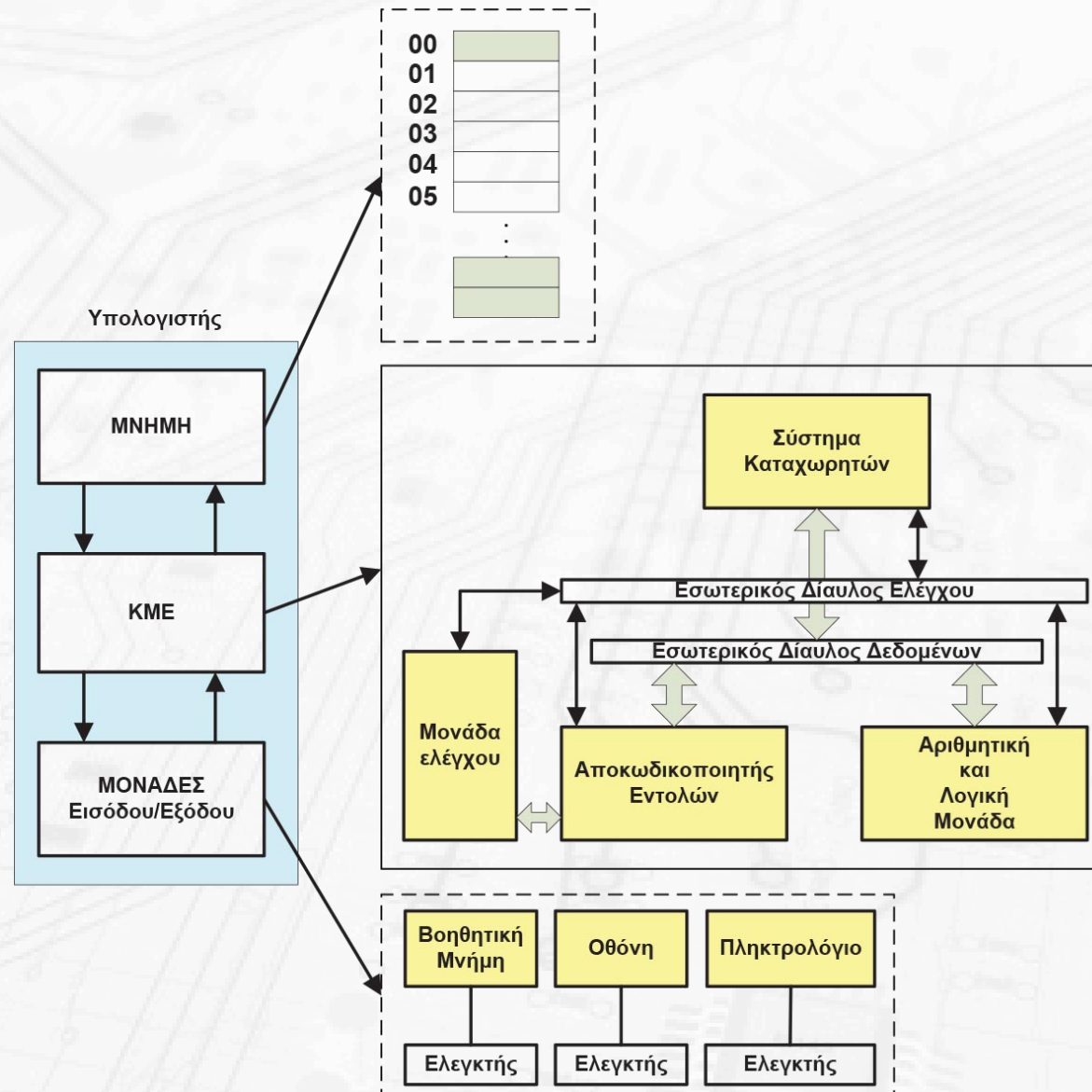


Αριθμητική και Λογική Μονάδα (ΑΛΜ-ALU-Arithmetic & Logic Unit) = εκτέλεση αριθμητικών και λογικών πράξεων

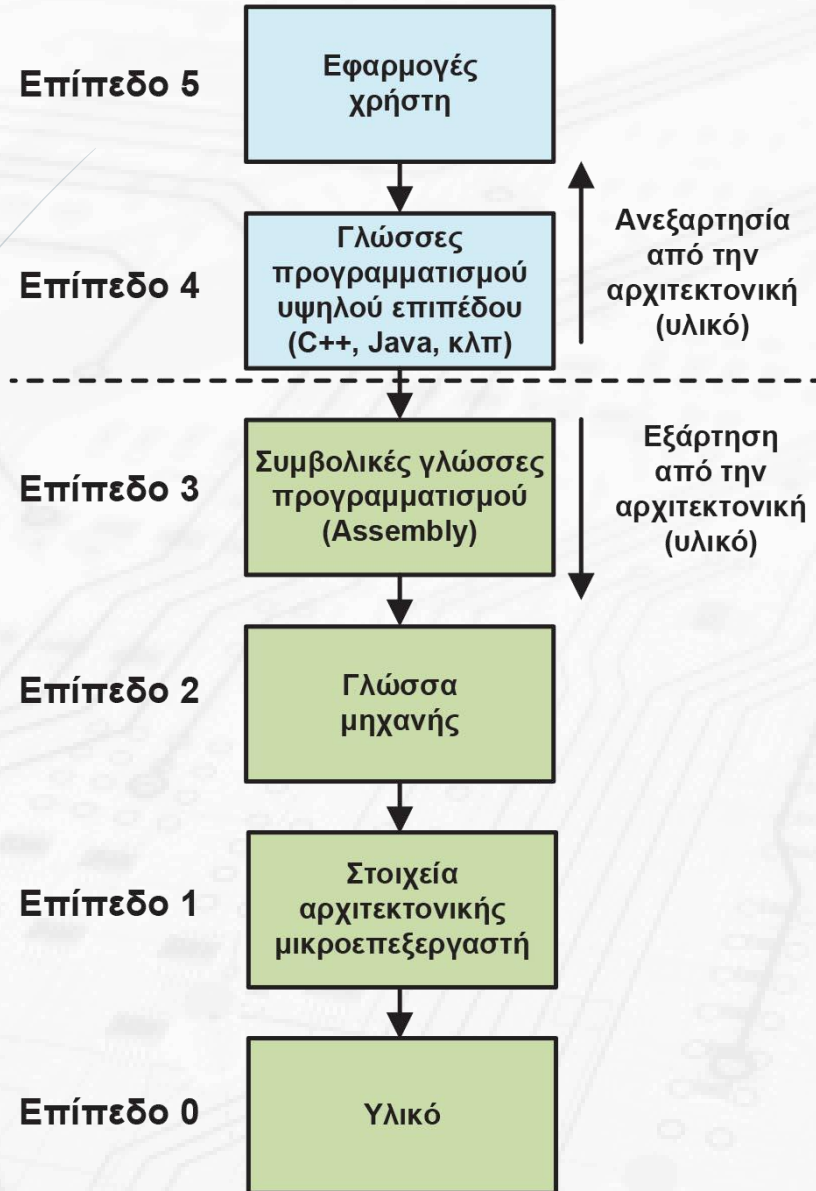
Μονάδα ελέγχου = «συντονισμός» εσωτερικών λειτουργιών μικροεπεξεργαστή

Καταχωρητές = συγκεκριμένες περιοχές αποθήκευσης που χρησιμοποιούνται για τη διαχείριση των δεδομένων που λαμβάνει χώρα στο εσωτερικό της ΚΜΕ

Βασικά χαρακτηριστικά υπολογιστικού συστήματος



Λογισμικό & επίπεδα συστήματος



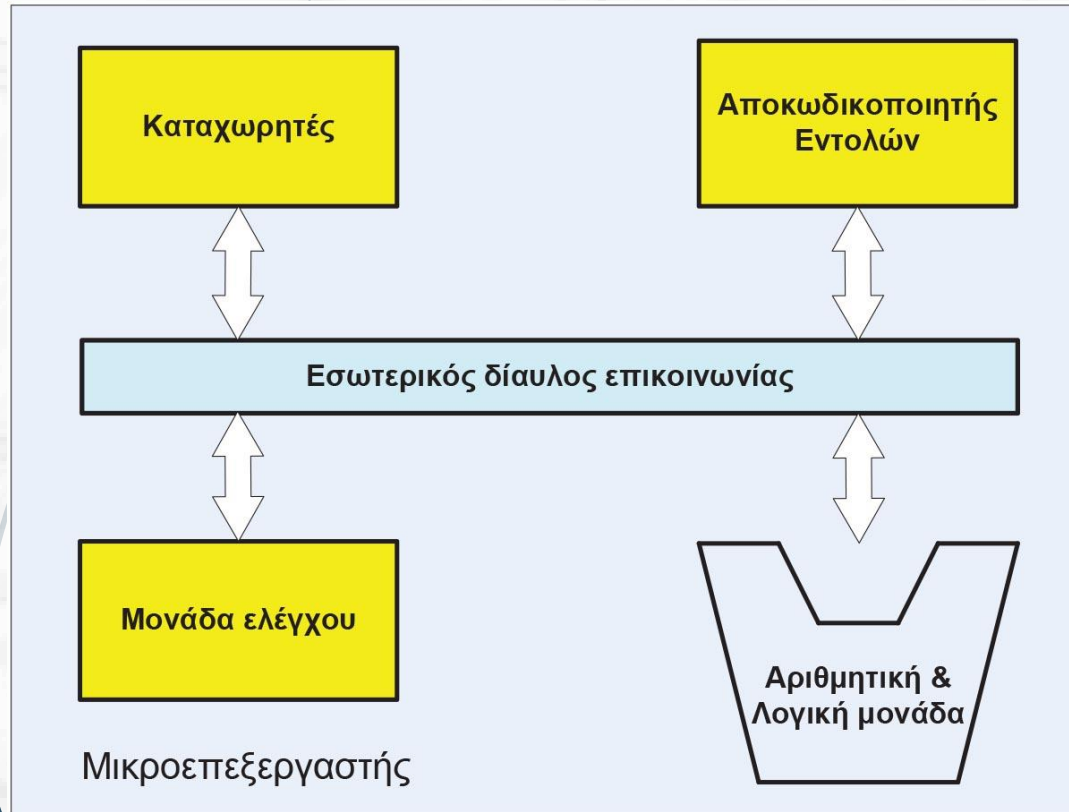
Υλικό (Hardware)

- Περιλαμβάνει ολοκληρωμένα κυκλώματα και ηλεκτρονικές διατάξεις
- Καθιστά το υπολογιστικό σύστημα λειτουργικό

Λογισμικό (Software)

- Δίνει «ζωή» στο υλικό
- Υποστηρίζει τη λειτουργία του συστήματος και τις εφαρμογές των χρηστών
- Αξιοποιεί το υλικό

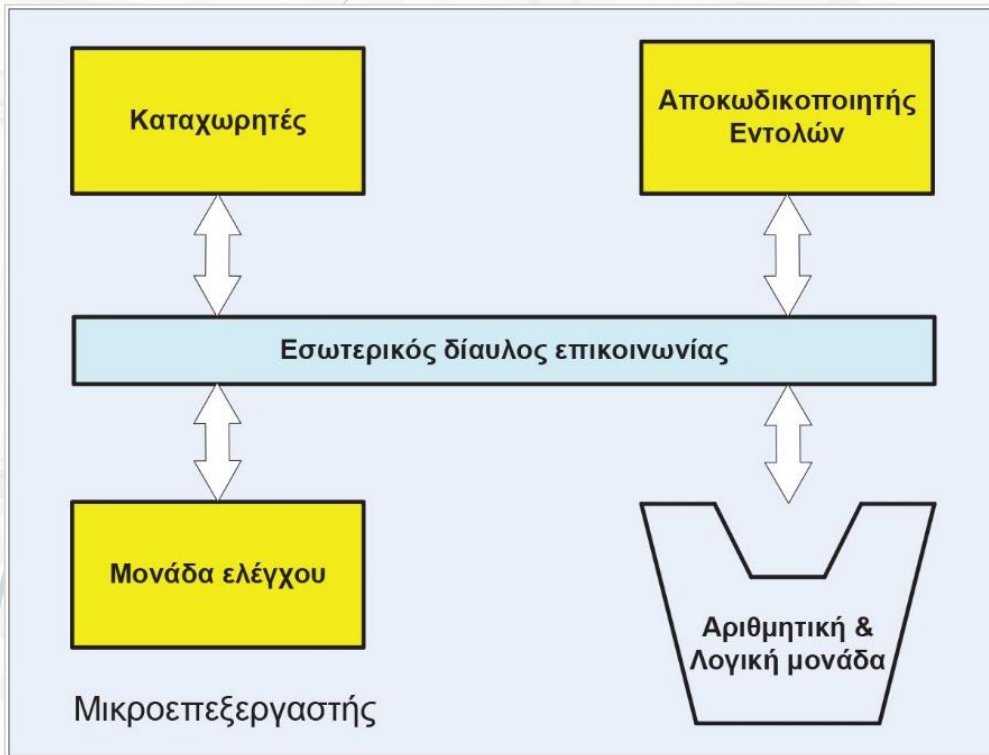
Γενική δομή μικροεπεξεργαστή



Καταχωρητές

- ✓ Συγκεκριμένες περιοχές προσωρινής αποθήκευσης
- ✓ Συγκεκριμένο όνομα και χωρητικότητα
- ✓ Πλήθος και χωρητικότητα ανάλογα με την αρχιτεκτονική του μικροεπεξεργαστή
- ✓ Τυπική χωρητικότητα 8/16/32/64/128 bit
- ✓ Φιλοξενούν δεδομένα που μεταφέρονται από τη μνήμη στον μικροεπεξεργαστή και το αντίστροφο
- ✓ Χρησιμοποιούνται υποχρεωτικά για τη μεταφορά δεδομένων
- ✓ Γενικής και ειδικής χρήσης

Γενική δομή μικροεπεξεργαστή- II



Μονάδα ελέγχου

- ✓ Συντονισμός ενεργειών των μονάδων του μικροεπεξεργαστή προκειμένου να είναι δυνατή η εκτέλεση των εντολών
- ✓ Παραγωγή κατάλληλων σημάτων «ενεργοποίησης» (μέσω συστήματος εσωτερικών διαύλων)

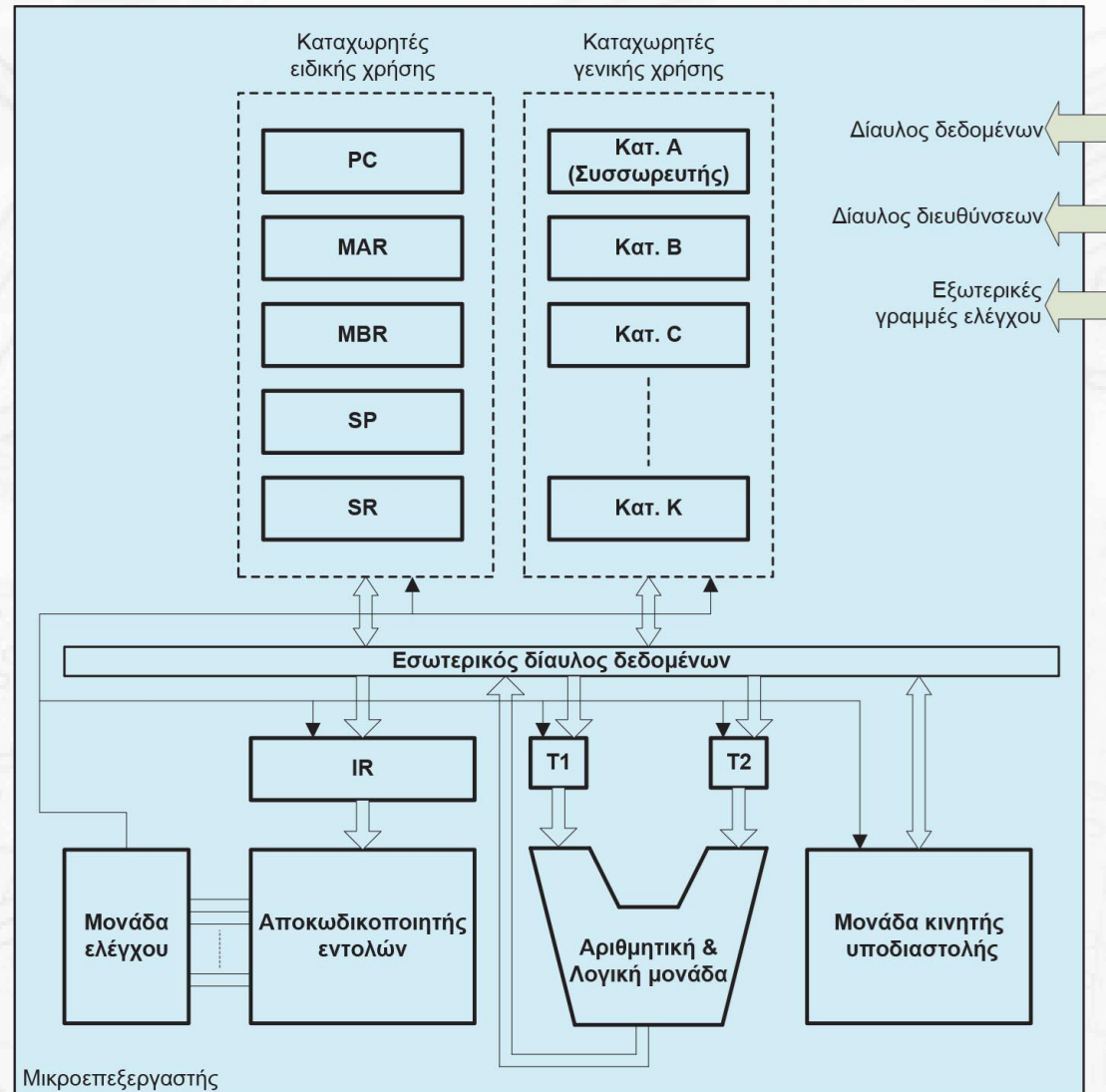
Αποκωδικοποιητής εντολών

- ✓ «Υποδέχεται» μια εντολή όταν αυτή «έρθει» από τη μνήμη, ώστε στη συνέχεια να γνωρίζει ο μικροεπεξεργαστής τι ακριβώς θα πρέπει να κάνει αναφορικά με την εκτέλεσή της

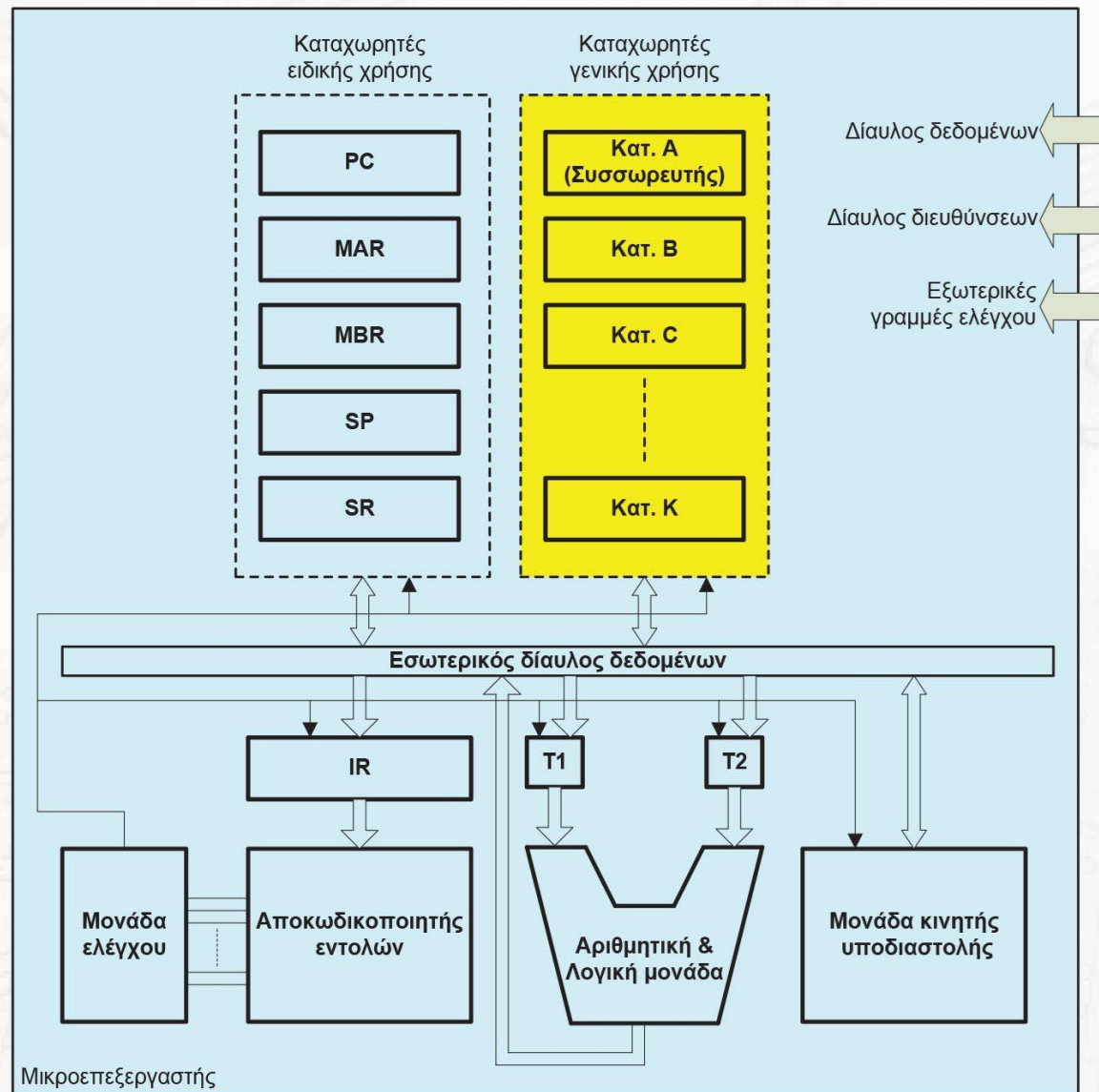
Αριθμητική & Λογική μονάδα

- ✓ Εκτέλεση αριθμητικών και λογικών πράξεων για τις εντολές του προγράμματος ή τις εσωτερικές λειτουργίες του μικροεπεξεργαστή

Περισσότερα για το εσωτερικό του μικροεπεξεργαστή

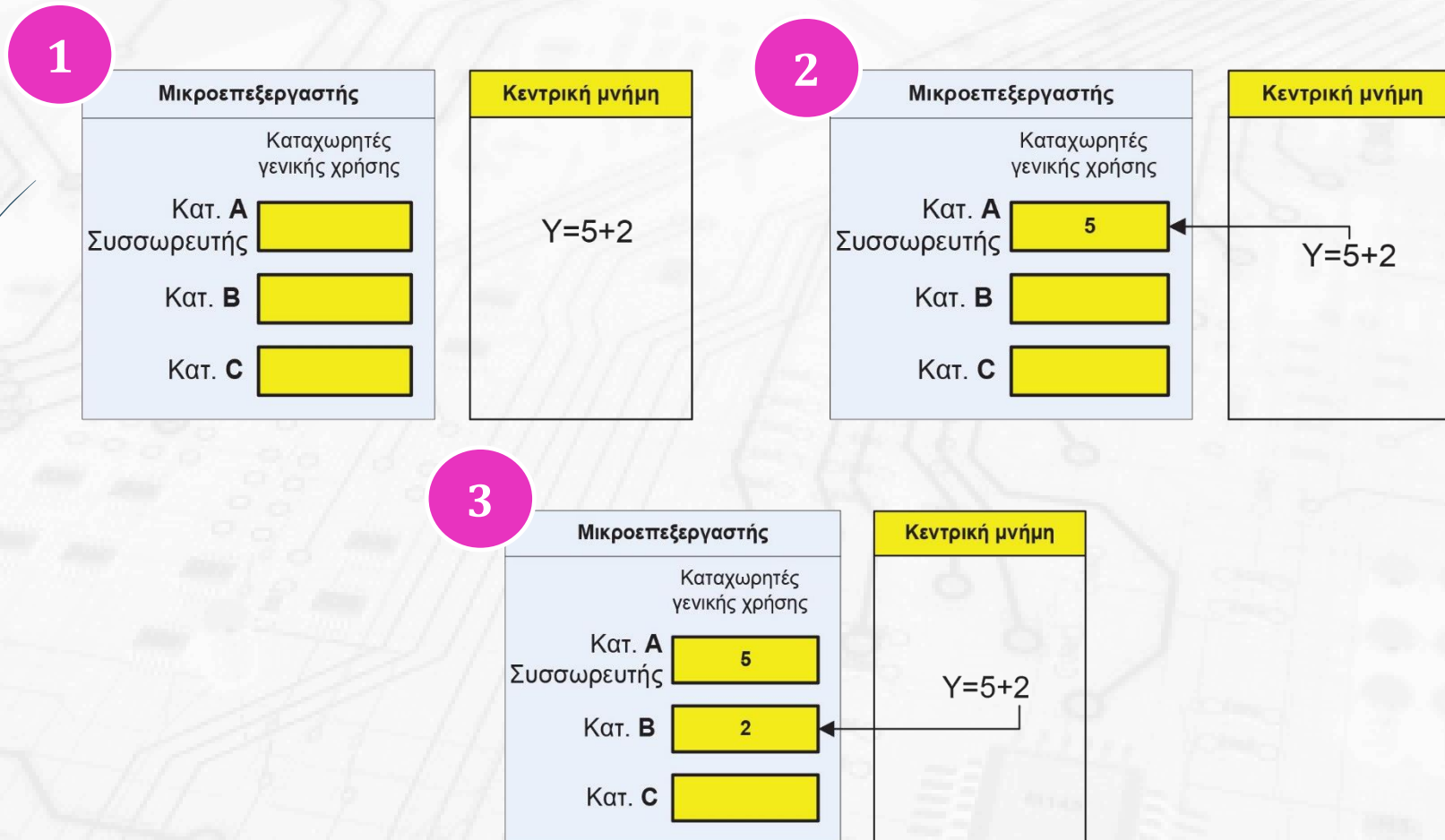


Καταχωρητές γενικής χρήσης

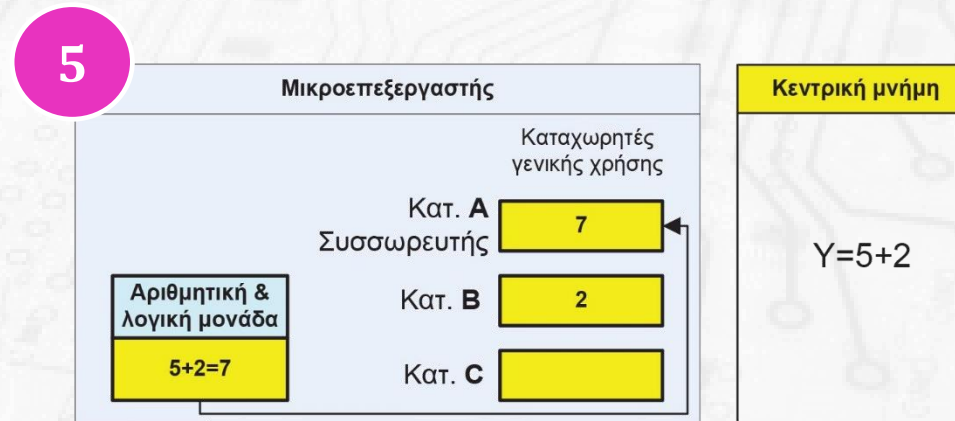
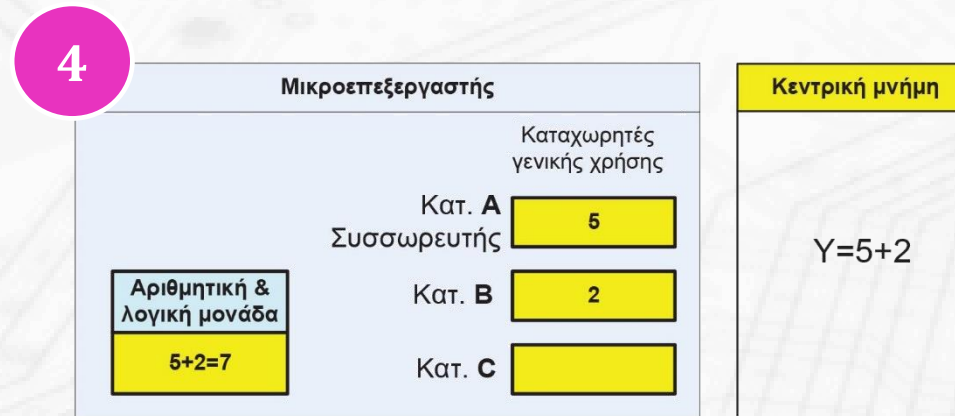


Ένα γενικό παράδειγμα αριθμητικής πράξης με την εμπλοκή καταχωρητών⁽¹⁾

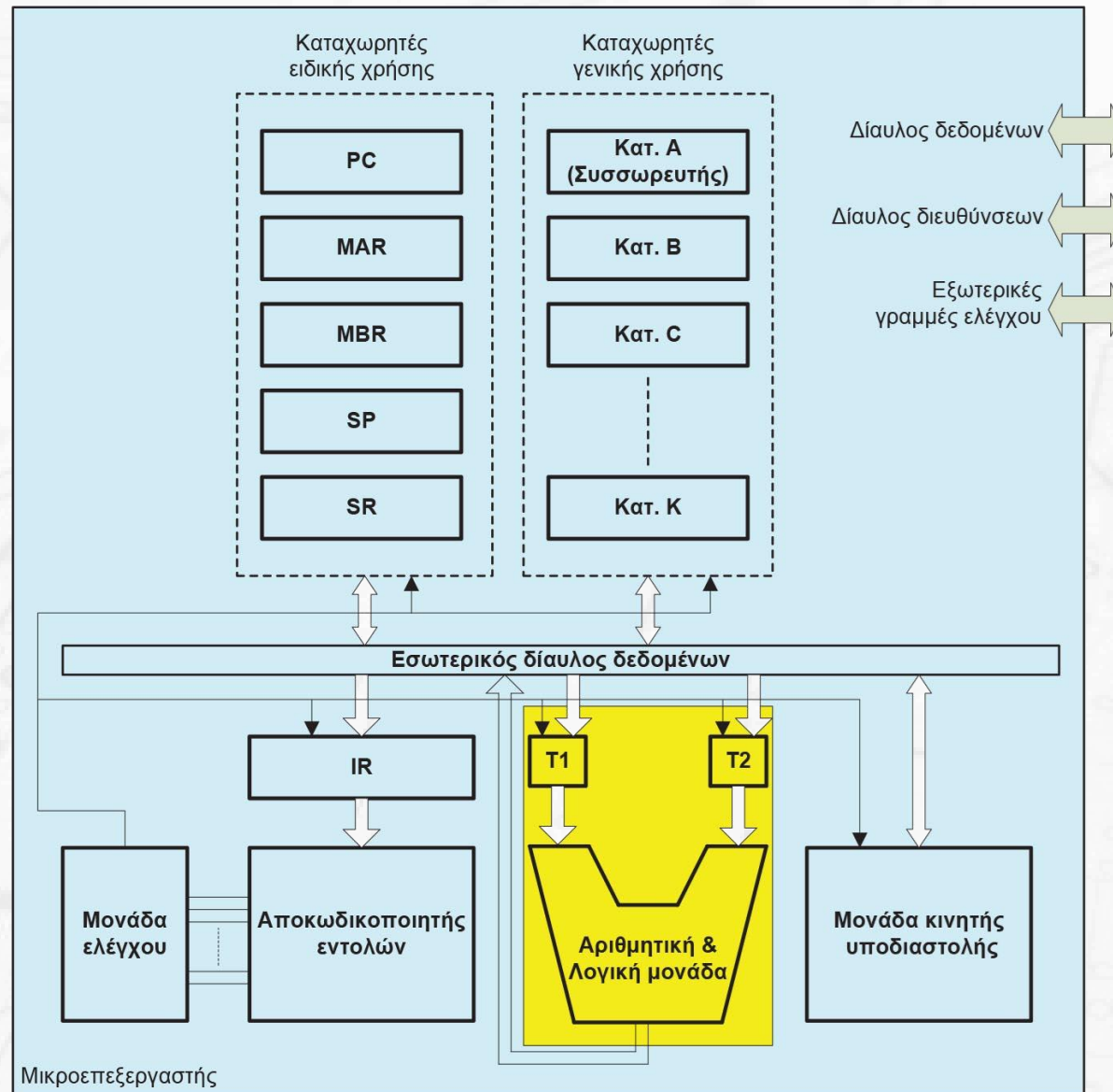
Ο προγραμματιστής έχει δημιουργήσει την εντολή $Y=5+2$ (πρόσθεση των αριθμών 5 και 2)



Ένα γενικό παράδειγμα αριθμητικής πράξης με την εμπλοκή καταχωρητών⁽²⁾



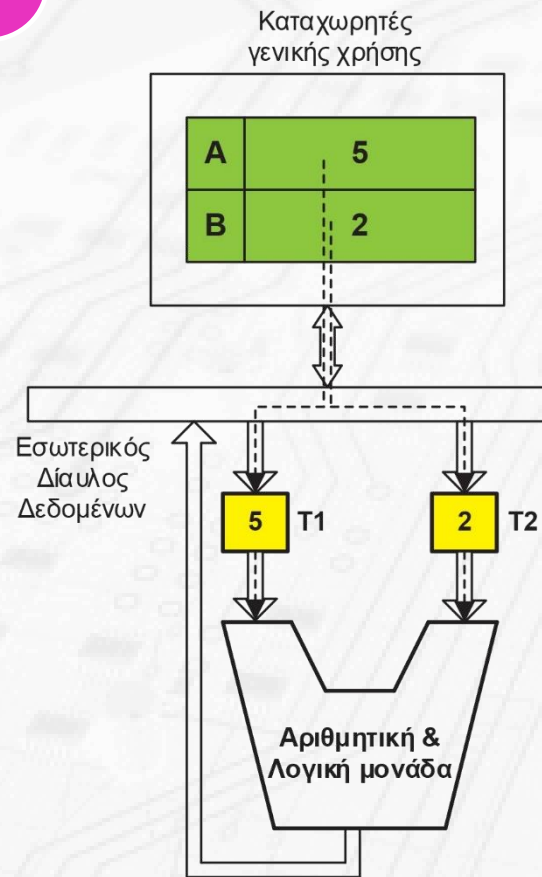
Αριθμητική και λογική μονάδα



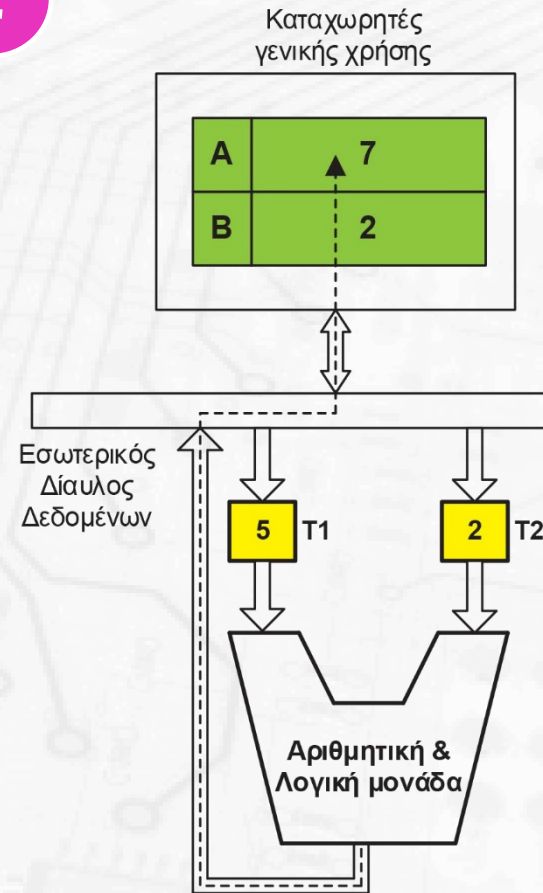
Αριθμητική και λογική μονάδα

Φιλοσοφία λειτουργίας για την πρόσθεση 5+2

1

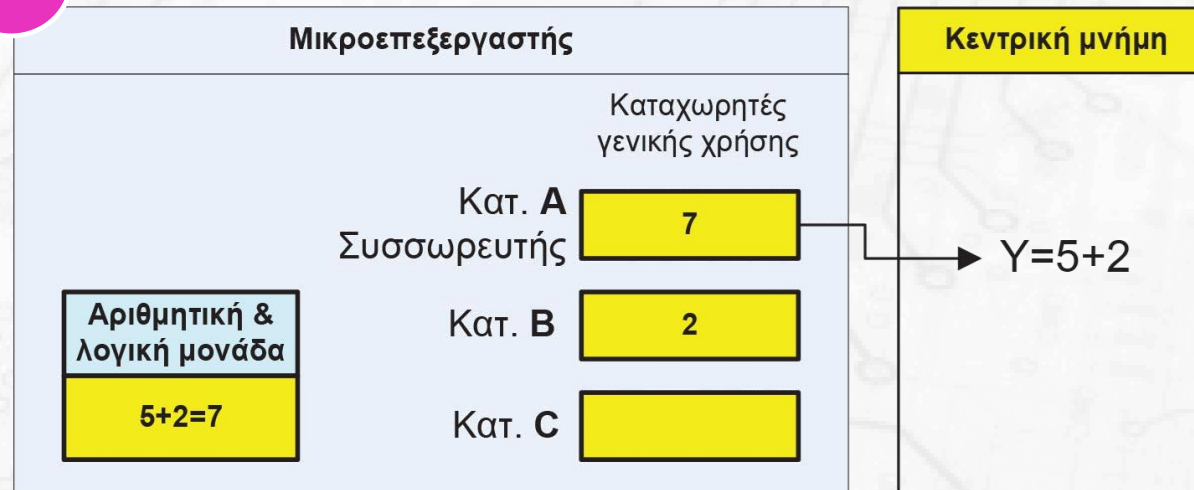


2

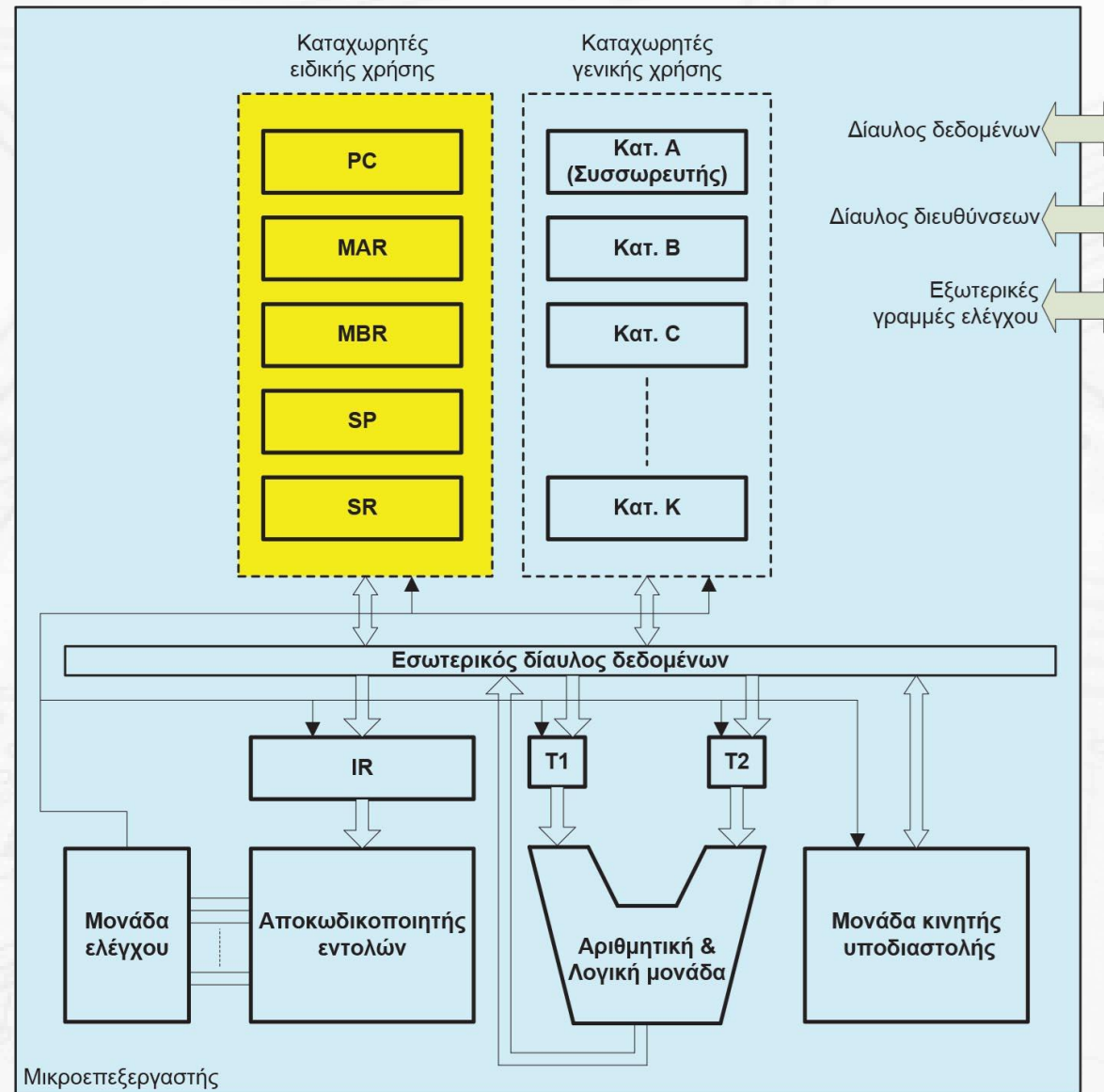


Ένα γενικό παράδειγμα αριθμητικής πράξης με την εμπλοκή καταχωρητών⁽³⁾

6



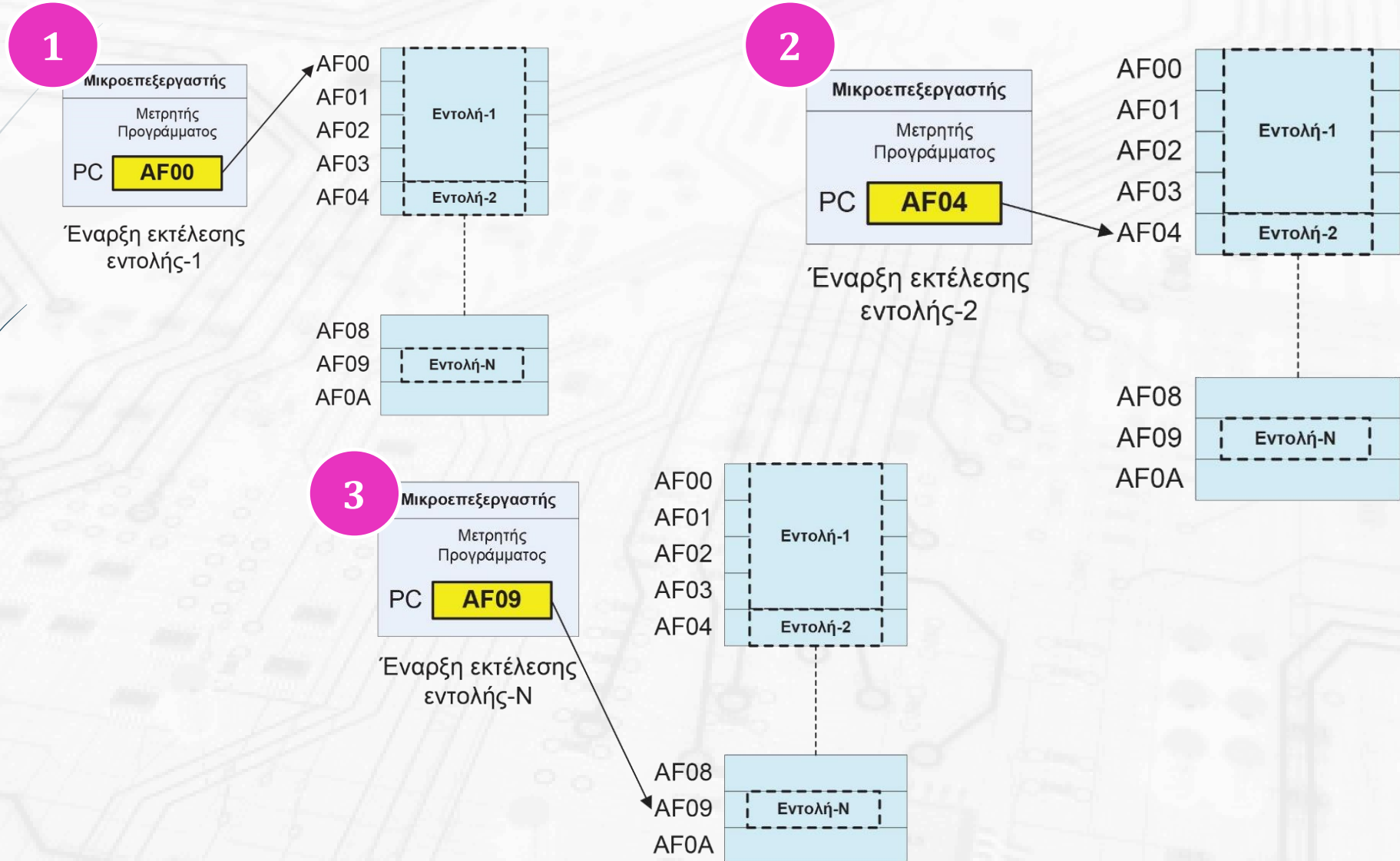
Καταχωρητές ειδικής χρήσης



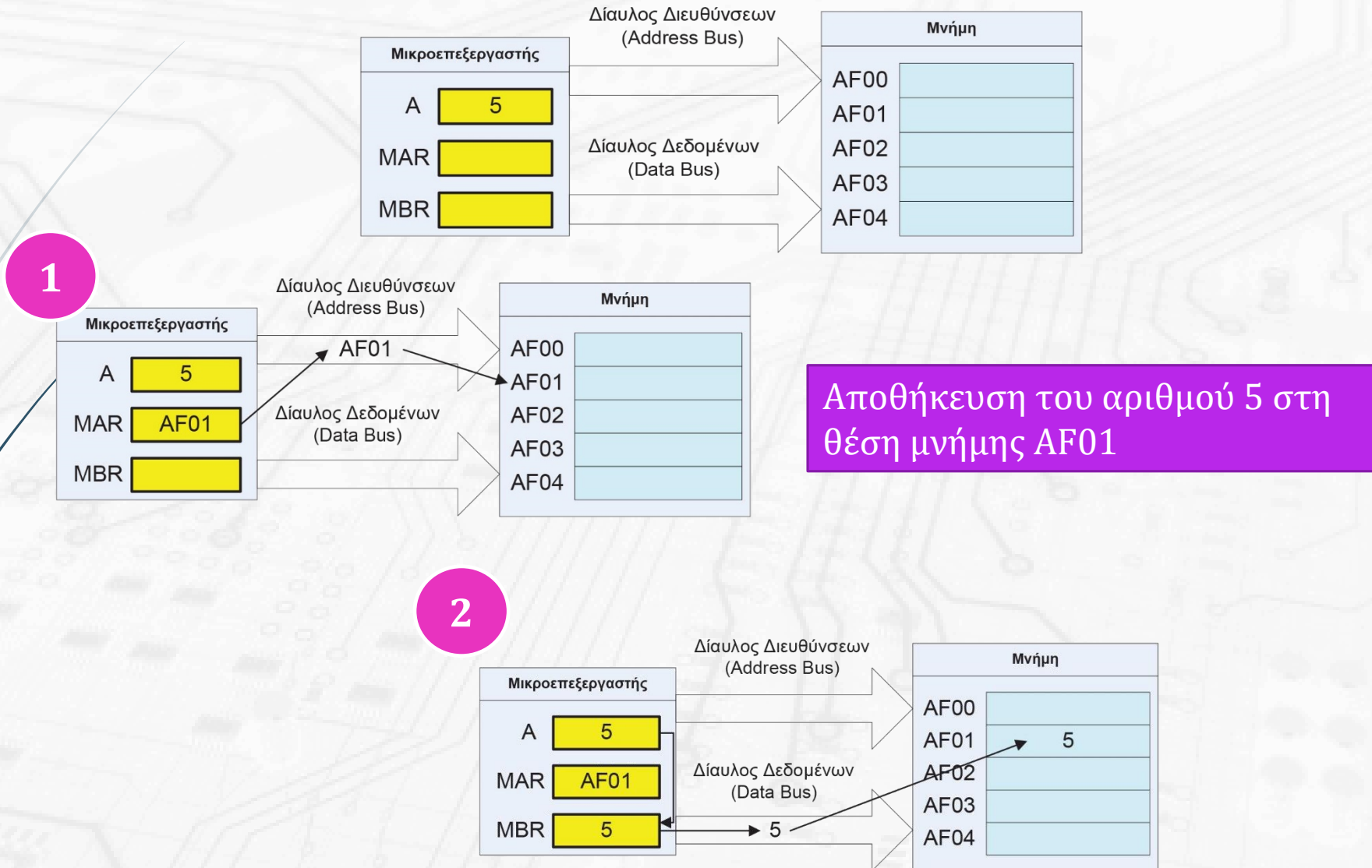
PC – Program Counter (ΜΠ – Μετρητής Προγράμματος)

- ✓ Δείχνει τη διεύθυνση της επόμενης προς εκτέλεση εντολής
- ✓ Οι εντολές του προγράμματος βρίσκονται αποθηκευμένες στη μνήμη
- ✓ Εκτέλεση εντολής = προϋποθέτει την ανάκλησή της από τη μνήμη
- ✓ Ανάκληση = ο μικροεπεξεργαστής πρέπει να γνωρίζει σε ποια θέση μνήμης (διεύθυνση) ξεκινά η εντολή (μια εντολή μπορεί να καταλαμβάνει περισσότερες από μια θέσεις μνήμης)

Γενικό παράδειγμα εκτέλεσης εντολών (χρήση του ΜΠ)



MAR – Memory Access Register (ΚΔΜ – Καταχωρητής Διευθύνσεων Μνήμης) & MBR – Memory Buffer Register (ΠΚΔ – Προσωρινός Καταχωρητής Δεδομένων)



SP – Stack Pointer (ΔΣ – Δείκτης Σωρού)

Δείχνει πάντα την κορυφή του σωρού

- ✓ Σωρός = τρόπος οργάνωσης δεδομένων
- ✓ Υλοποίηση = στη μνήμη
- ✓ Πρόσβαση στο σωρό = στην κορυφή του
- ✓ Προώθηση – Ανάκτηση = Αυτό που προωθείται τελευταίο ανακτάται πρώτο
- ✓ Απλός τρόπος διαχείρισης



Παράδειγμα χρήσης σωρού

Βήμα 1: Ο σωρός είναι κενός

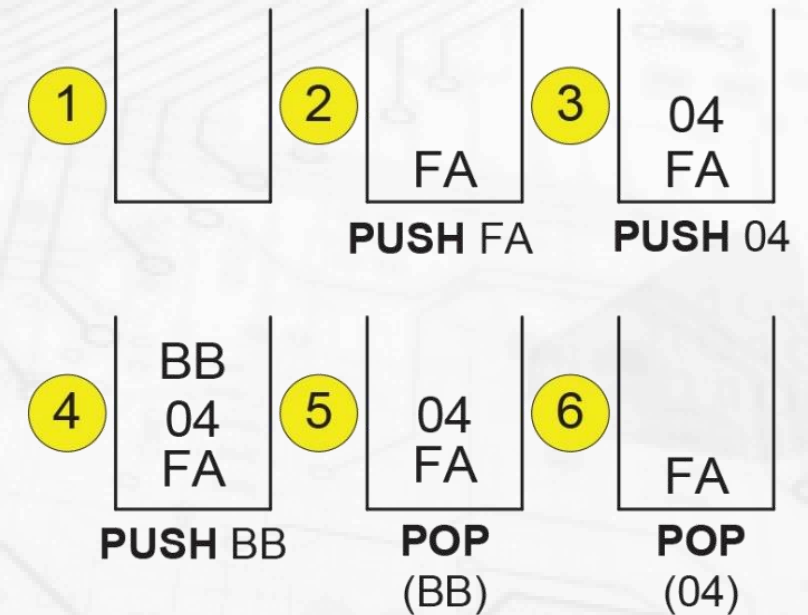
Βήμα 2: Αποθήκευση (ώθηση) του αριθμού FA στην κορυφή του σωρού (διαδικασία PUSH)

Βήμα 3: Αποθήκευση (ώθηση) του αριθμού 04 στην κορυφή του σωρού (διαδικασία PUSH)

Βήμα 4: Αποθήκευση (ώθηση) του αριθμού BB στην κορυφή του σωρού (διαδικασία PUSH)

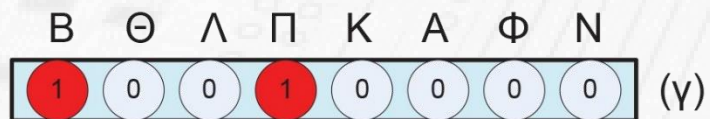
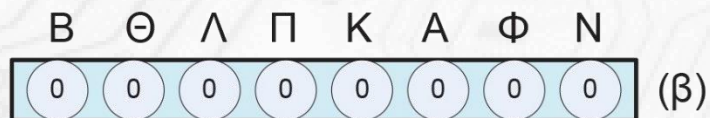
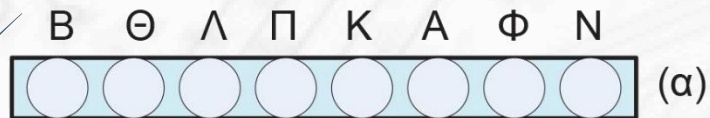
Βήμα 5: Ανάκτηση (απώθηση) του αριθμού από την κορυφή του σωρού (αριθμός BB, διαδικασία POP)

Βήμα 6: Ανάκτηση (απώθηση) του αριθμού από την κορυφή του σωρού (αριθμός 04, διαδικασία POP)



SR – Status Register (ΚΚ – Καταχωρητής Κατάστασης)

- ✓ Αποτελείται από bit που λειτουργούν ανεξάρτητα μεταξύ τους
- ✓ Τα bit δείχνουν ανά πάσα στιγμή την «κατάσταση» του μικροεπεξεργαστή
- ✓ Πληροφορίες για κρατούμενο, υπερχείλιση, πρόσημο, κλπ



(α) Ενδεικτικά λαμπάκια

(β) Δεν υπάρχει κάποια ειδοποίηση

(γ) Υπάρχουν δύο ειδοποιήσεις (B, Π)

- ✓ Ας φανταστούμε τα ενδεικτικά λαμπάκια του αυτοκινήτου
- ✓ Κάθε ένα από αυτά λειτουργεί ανεξάρτητα και μας δίνει διαφορετική πληροφορία
- ✓ Στο διπλανό σχήμα, οι ενδείξεις είναι:

B: Βενζίνη (χαμηλή στάθμη),

Θ: Θερμοκρασία (υψηλή)

Λ: Λάδια (χαμηλή στάθμη)

Π: Πόρτες (ανοιχτές)

Κ: Καυσαέρια (κακή ποιότητα)

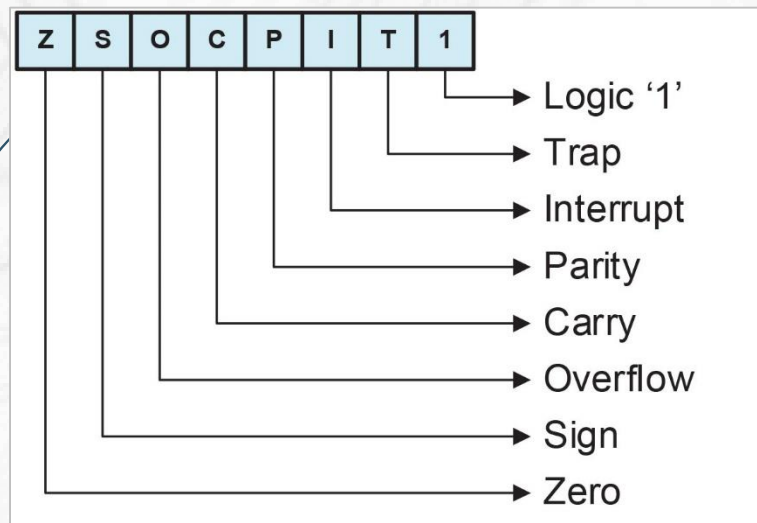
Α: Αερόσακος (δυσλειτουργία)

Φ: Φρένα (χαμηλή πίεση)

Ν: Νερό (χαμηλή στάθμη)

SR – Status Register (ΚΚ – Καταχωρητής Κατάστασης)

Συνηθισμένη δομή καταχωρητή κατάστασης στους μικροεπεξεργαστές



Zero: έχει προκύψει μηδέν από μια αριθμητική πράξη (π.χ. μείωση του περιεχομένου ενός καταχωρητή)

Sign: όταν το πρώτο bit ενός αριθμού γίνει 1 (σε προσημασμένους αριθμούς)

Overflow: όταν ένας αριθμός δεν χωρά να αποθηκευτεί στη θέση προορισμού (π.χ. και σε υπερχείλιση αριθμητικής με συμπλήρωμα ως προς 2)

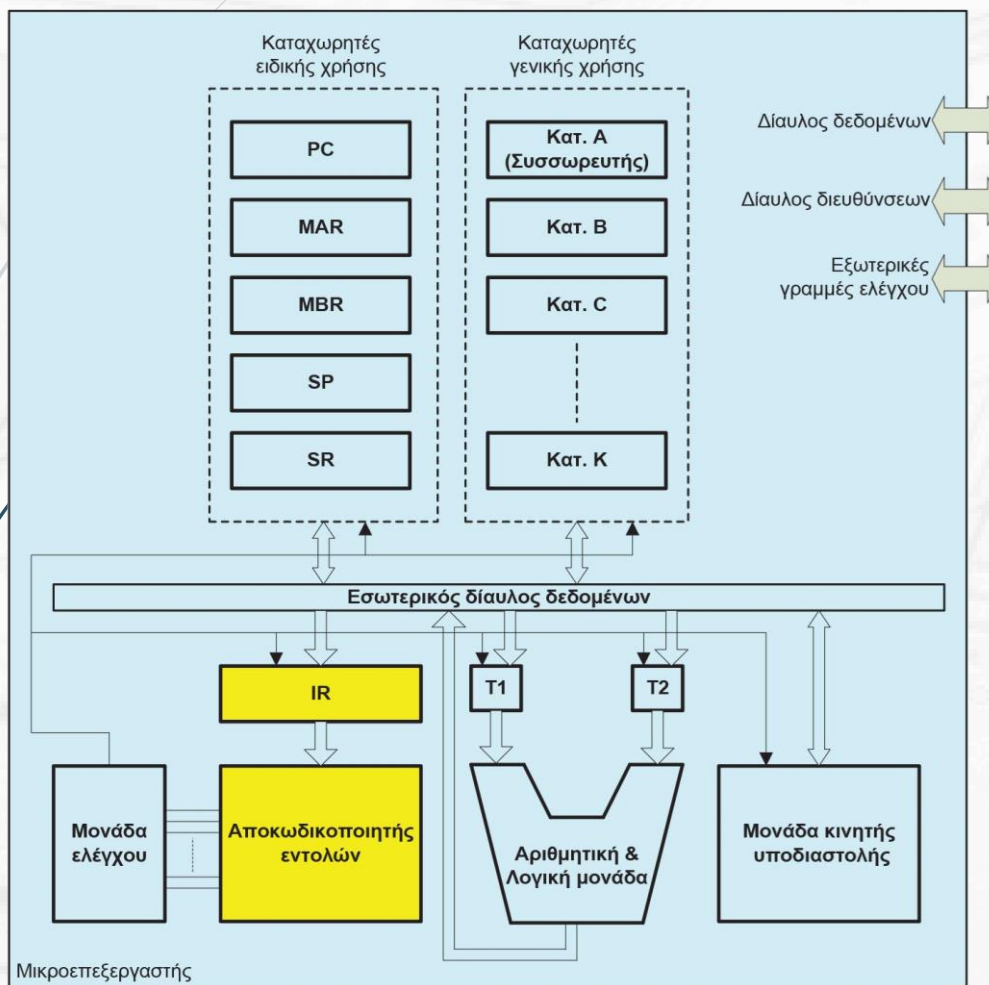
Carry: έχει προκύψει κρατούμενο από πράξη

Parity: ισοτιμία των bit ενός αριθμού (π.χ. περιττό πλήθος ψηφίων με την τιμή 1)

Interrupt: επιτρέπονται ή όχι οι διακοπές

Trap: χρήση για βήμα προς βήμα αποσφαλμάτωση **1:** λογικό '1' ως τιμή αναφοράς

Αποκωδικοποιητής εντολών



- ✓ Όταν πρόκειται να εκτελεστεί μια εντολή, μεταφέρεται ο αντίστοιχος κώδικας (σε ορισμένες περιπτώσεις μαζί με τα ορίσματα από την πρώτη «μεταφορά») από τη μνήμη στον **Καταχωρητή Εντολών** ΚΕ (IR-Instruction Register)
- ✓ Ο **αποκωδικοποιητής** δέχεται ως είσοδο το περιεχόμενο του ΚΕ και «δίνει οδηγίες» για την εκτέλεση της εντολής

Μονάδα ελέγχου

- ✓ Συντονίζει με τα κατάλληλα σήματα τη λειτουργία όλων των μονάδων του μικροεπεξεργαστή ώστε να είναι δυνατή η εκτέλεση των εντολών.
- ✓ Ο συντονισμός αυτός εστιάζεται κυρίως στον έλεγχο της ροής δεδομένων μεταξύ των μονάδων του μικροεπεξεργαστή μέσω του εσωτερικού διαύλου δεδομένων
- ✓ Ελέγχει το χρονισμό των μονάδων με την αξιοποίηση του ρολογιού
- ✓ Δέχεται πληροφορίες από τον αποκωδικοποιητή

Μονάδα κινητής υποδιαστολής

- ✓ Παλαιοί μικροεπεξεργαστές = αριθμητικές πράξεις μέσω λογισμικού, αργότερα από ξεχωριστό ολοκληρωμένο κύκλωμα (συνεπεξεργαστής)
- ✓ Σύγχρονοι μικροεπεξεργαστές = την περιλαμβάνουν στο εσωτερικό τους, διαθέτει δικές της εντολές και καταχωρητές, ενώ απαιτεί πιο «ειδικό» προγραμματισμό

Κεντρική μνήμη

Θέσεις και διευθύνσεις

Διεύθυνση	Περιεχόμενο
00	
01	
02	
03	
04	
05	
06	
07	
08	
09	

Υποθετικό μοντέλο 10 θέσεων

N το πλήθος bit για την αποτύπωση κάθε διεύθυνσης (M=πλήθος θέσεων μνήμης)

$$2^N \geq M$$

Για μεγάλο M χρησιμοποιούμε τον επόμενο τύπο (Το N στρογγυλοποιημένο προς τα πάνω)

$$N = \lceil \log_2 M \rceil$$

Γενικά χαρακτηριστικά της κεντρικής μνήμης



**Ανάγνωση του περιεχομένου
της θέσης AF00**

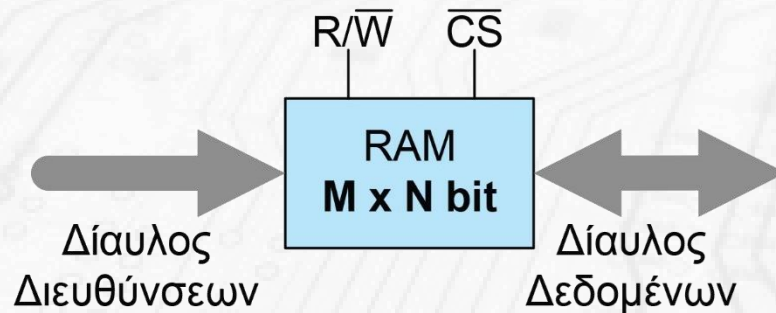
Γενικά χαρακτηριστικά της κεντρικής μνήμης

Λειτουργίες κυκλώματος μνήμης (μέσω σημάτων)

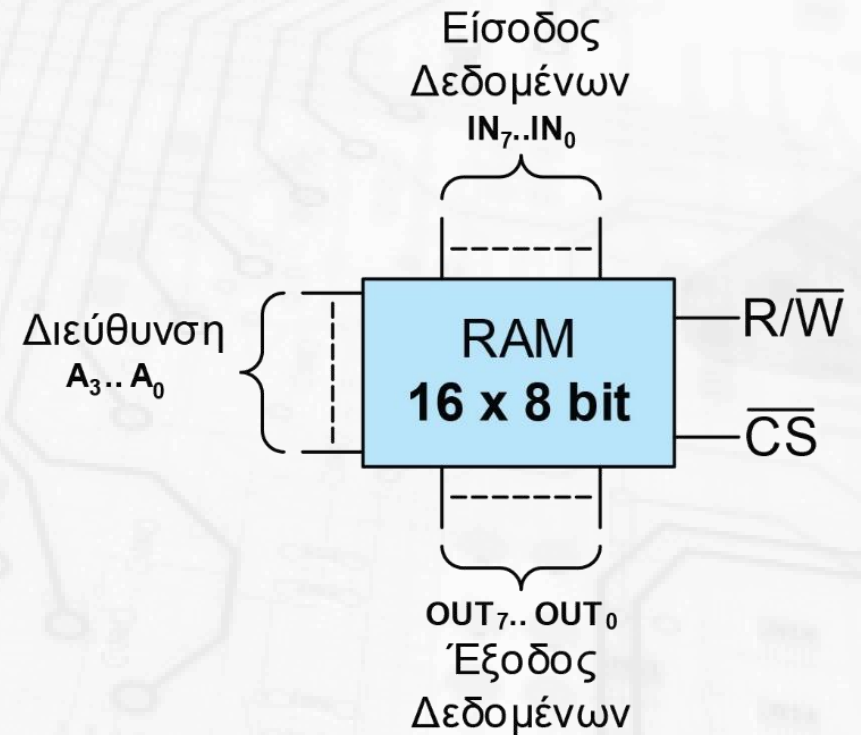
- ✓ Επιλογή διεύθυνσης μνήμης για εγγραφή / ανάγνωση
- ✓ Επιλογή λειτουργίας (ανάγνωση / εγγραφή)
- ✓ Τροφοδοσία με δεδομένα προς αποθήκευση
- ✓ Δέσμευση ή απελευθέρωση μνήμης

Γενικά χαρακτηριστικά της κεντρικής μνήμης

Γενική μορφή στοιχείου μνήμης RAM



Στοιχείο μνήμης RAM 16x8 bit



Χαρτογράφηση μνήμης

Έστω ότι ένα σύστημα διαθέτει μνήμη που αποτελείται από δύο περιοχές RAM μεγέθους 8Kbyte η κάθε μία. Γνωρίζοντας ότι 1Kbyte=1024byte μπορούμε να βρούμε τις οριακές διευθύνσεις των αντίστοιχων περιοχών

Οριακές
διευθύνσεις

Περιοχή

0

8Kbyte

$(8 \times 1024) - 1 = 8191$

8192

8Kbyte

$(16 \times 1024) - 1 = 16383$

Περιοχή

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

8Kbyte

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

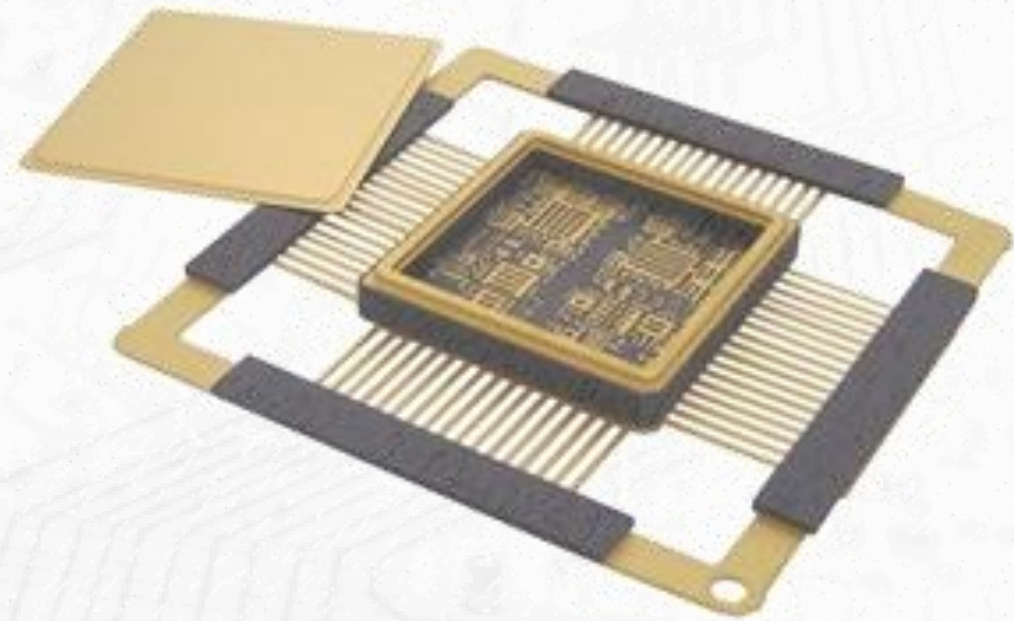
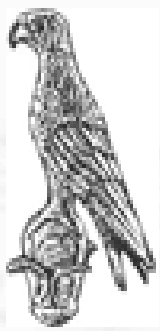
8Kbyte

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1



Το περισσότερο σημαντικό bit (A13) θα χρησιμοποιηθεί για την επιλογή περιοχής μνήμης

	Περιοχή
0 0 0 0 0 0 0 0 0 0 0 0 0 0	8Kbyte
0 1 1 1 1 1 1 1 1 1 1 1 1 1	
1 0 0 0 0 0 0 0 0 0 0 0 0 0	8Kbyte
1 1 1 1 1 1 1 1 1 1 1 1 1 1	



Εισαγωγή στους μικροελεγκτές (παράδειγμα οικογένεια MSP430)

Συστήματα Μικροεπεξεργαστών

Διδάσκων: Γρηγόριος Δουμένης

Διδακτικό Υλικό: Νικόλαος Γιαννακέας, Γρηγόριος Δουμένης

Μικροελεγκτές/Μικροεπεξεργαστές

Ορισμός

Οι μικροελεγκτές είναι **μικροεπεξεργαστές υψηλής ολοκλήρωσης** οι οποίοι έχουν σχεδιαστεί ειδικά για την χρήση τους σε **ενσωματωμένα συστήματα**. Ένας τυπικός μικροελεγκτής περιλαμβάνει:

- Ολοκληρωμένο κύκλωμα CPU
- Μνήμες (RAM + ROM / EPROM / EEPROM)
- Συσκευές Εισόδου Εξόδου
 - Μετατροπείς Αναλογικού σε Ψηφιακό (Analog-to-digital / ADC)
 - Μετατροπείς Ψηφιακού σε Αναλογικό σε DAC (Digital-to-analog) converters
 - Πομποδέκτες UARTs (Universal Asynchronous Receiver Transmitter)
 - Κυκλώματα I²C (Inter-Integrated Circuit)



Μικροελεγκτές/Μικροεπεξεργαστές

Συστατικά

Απαραίτητα:

- Μικροεπεξεργαστής/ Μικροελεγκτής
- Μνήμη (On-chip ή Off chip)
- Communication path with the interacting environment

Προαιρετικά:

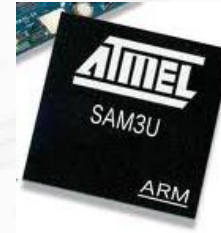
- Μέσα αποθήκευσης (Μόνιμη μνήμη)
- Διεπαφή Χρήστη
- Αισθητήρες
- Μετατροπής (A-D and D-A)
- Ενεργοποιητές (Actuators)

- Integration: Able to implement a whole design onto a single chip.
- Cost: Are usually low-cost devices (a few \$ each);
- Clock frequency: Compared with other devices (microprocessors and DSPs), MCUs use a low clock frequency:
 - MCUs today run up to 100 MHz/100 MIPS (Million Instructions Per Second).
- Power consumption: Low power (battery operation);
- Bits: 4 bits (older devices) to 32 bits devices;
- Memory: Limited available memory, usually less than 1 MByte;
- Input / Output (I/O): Low to high (8 to 150) pin-out count.

Lots of manufacturers ship ARM products



Actel[®]
POWER MATTERS



ATMEL[®]



 **CYPRESS**
PERFORM



DUST
NETWORKS



ember



NXP



ST



 **TEXAS**
INSTRUMENTS

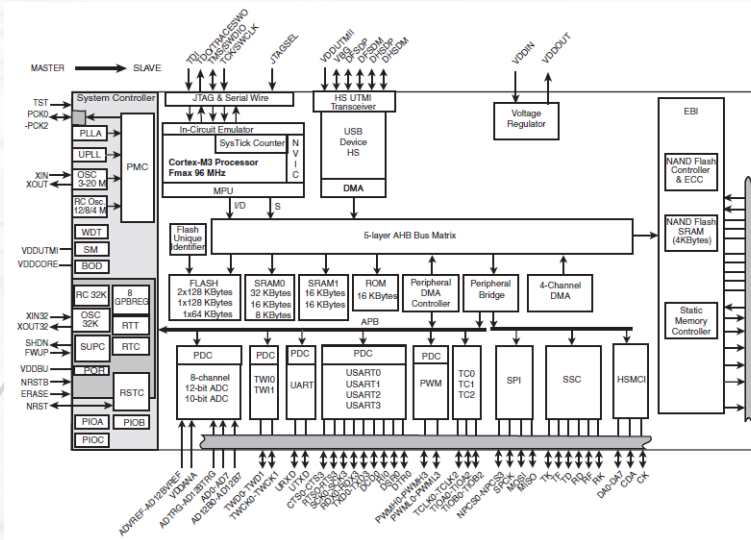
Γνωστοί μικροελεγκτές

Εταιρία	Διεύθυνση στο internet	Ενδεικτικό μοντέλο μικροελεγκτή
Texas Instruments	http://www.ti.com/	MSP430F1x
Microchip	http://www.microchip.com/	PIC16F84A
Zilog	http://www.zilog.com/	S3F80P5
Freescale	http://www.freescale.com/	HC08EY
Atmel	http://www.atmel.com/	AVR ATMEGA328
Parallax	http://www.parallax.com/	Basic Stamp
Maxim	http://www.maximintegrated.com/	MAXQ611
Intel	-	8051 (δεν παράγεται πλέον από την Intel)



What differentiates these
products from one another?

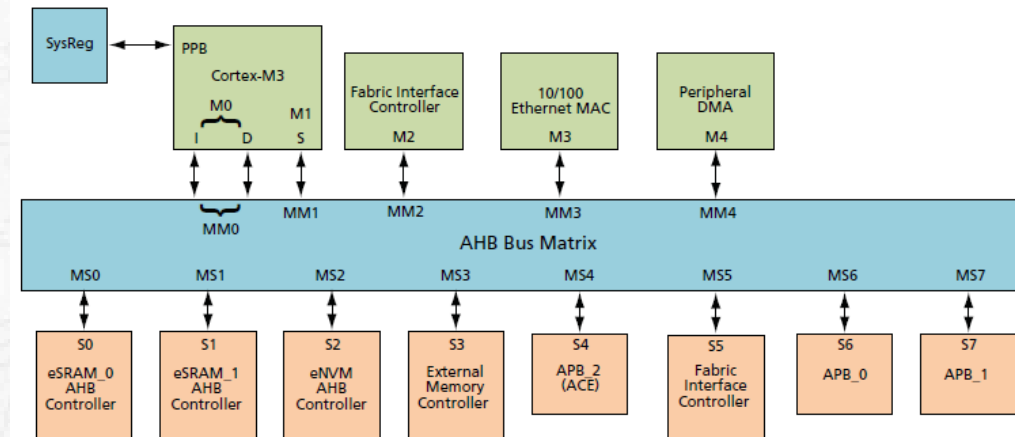
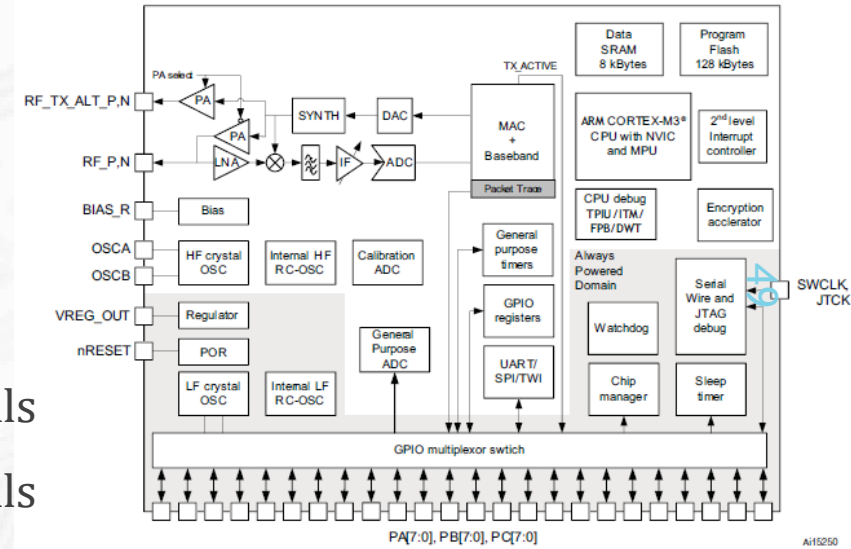
The difference is...



Peripherals

Peripherals

Peripherals



Γνωστοί μικροελεγκτές

Microcontrollers (MCU)				Application (MPU)		
MSP430	C2000	Tiva C	Hercules	Sitara	DSP	Keystone
16-bit Ultra Low Power & Cost	32-bit Real-time	32-bit All-around MCU	32-bit Safety	32-bit Linux Android	16/32-bit All-around DSP	32-bit Massive Performance
MSP430 ULP RISC MCU	• Real-time C28x MCU • ARM M3+C28	ARM Cortex-M4F	ARM Cortex-M3 Cortex-R4	ARM Cortex-A8 Cortex-A9	DSP C5000 C6000	• C66 + C66 • A15 + C66 • A8 + C64 • ARM9 + C674
• Low Pwr Mode ▪ 250nA (RTC) ▪ 770nA (LCD) • Analog I/F • USB and RF	• Motor Control • Digital Power • Precision Timers/PWM	• 32-bit Float • Nested Vector IntCtrl (NVIC) • Ethernet (MAC+PHY)	• Lock step Dual-core R4 • ECC Memory • SIL3 Certified	• \$5 Linux CPU • 3D Graphics • PRU-ICSS industrial subsys	• C5000 Low Power DSP • 32-bit fix/float C6000 DSP	• Fix or Float • Up to 12 cores 4 A15 + 8 C66x • DSP MMAC's: 352,000
TI-RTOS	TI-RTOS (k)	TI-RTOS	3rd Party (only)	Linux, Android, TI-RTOS Kernel	C5x: DSP/BIOS C6x: TI-RTOS (k)	Linux TI-RTOS (k)
Flash: 512K FRAM: 128K	512K Flash	1MB Flash	256K to 3M Flash	L1: 32K x 2 L2: 256K	L1: 32K x 2 L2: 256K	L1: 32K x 2 L2: 1M + 4M
25 MHz	300 MHz	120 MHz	220 MHz	1.35 GHz	800 MHz	1.4 GHz
\$0.25 to \$9.00	\$1.85 to \$20.00	\$1.00 to \$8.00	\$5.00 to \$30.00	\$5.00 to \$25.00	\$2.00 to \$25.00	\$30.00 to \$225.00

Μικροελεγκτής MSP430

Utility Metering

Electricity Meters
Gas Meters
Flow Meters
Smart Meters



Wireless Applications

Remote Sensors
Communication
Controllers
RFID



Consumer Electronics

Portable Electronics
Remote Controls
Personal Care
PC peripherals



Thousands of applications are enabled by MSP430 MCUs

Differentiation is possible with MSP430 MCU's Ultra-Low Power performance, high analog & digital peripheral integration, and easy-to-use tool chain.

Energy Harvesting

Renewable Energy
Battery-less devices
Solar, thermal, vibration, etc



Portable Medical

Blood Glucose Meters
Thermometers
Heart-Rate Monitors
Implantable Devices



Sensors & Security

Smoke Detector
Motion Detector
Vibration Detector
Smart Sensors



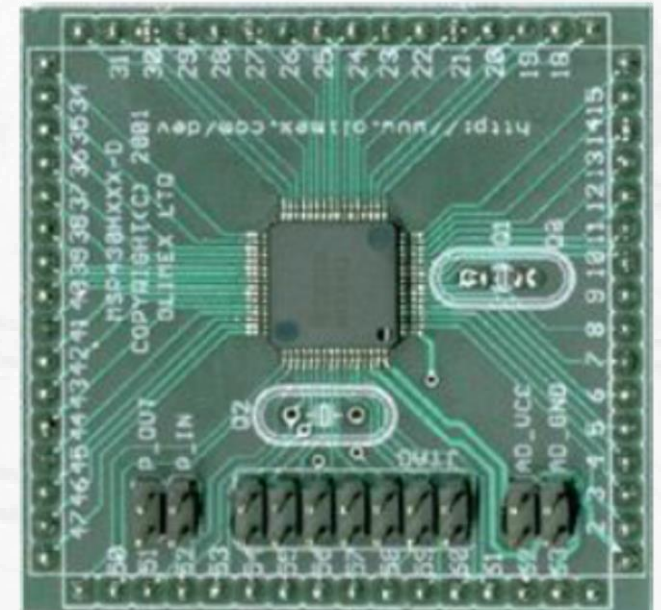
Personal Health & Fitness

Sports Watches
Pedometers
Calorimeters
Dive watches



Μικροελεγκτής MSP430

- **Αρχιτεκτονική εξαιρετικά χαμηλής κατανάλωσης (Ultralow-power architecture)**
 - 0.1- μ A RAM retention, 0.8- μ A real-time clock mode
 - 250- μ A / MIPS active
- **16-bit RISC CPU**
 - 27 εντολές (instructions) και 7 τρόπους διευθυνσιοδότησης (addressing modes)
- **Αρχιτεκτονική von-Neumann με ένα χώρο διευθύνσεων που διαμοιράζονται:**
 - Καταχωρητές ειδικού σκοπού (special function registers - SFRs)
 - Περιφερειακά (peripherals)
 - Μνήμη RAM (128 bytes $\rightarrow$ 2k)
 - Μνήμη Flash / ROM (1k $\rightarrow$ 60k)
- **In-system programmable Flash**
- **Μετατροπείς**
 - 12-bit or 10-bit ADC, 12-bit dual-DAC
- **Διεπαφή (USART Peripheral Interface)**
- **Είσοδοι/ Έξοδοι (Digital I/O ports)**
 - 2, 3 ή 6 πόρτες 8-bit (I/O ports)



Γνωστοί μικροελεγκτές

Series	Ultra-Low Power				FRxx FRAM	Low Power + Performance	Security + Communications	
Part Number	L09x Low Voltage	G2x/I2x	F1x	F2x/F4x		F5x/6x	RF-430	CC430
Max speed (MHz)	4	16	16	16	24	25	4	20
NVM (max KB)	0	56	120	120	128	512	ROM Fixed Function	32
SRAM (max KB)	2	4	10	8	2	67	4	4
GPIO	11	4–32	10–48	14–80	17–40	29–90	Up to 8	30–44
Comparator	•	•	•	•	•	•	•	•
Timer	•	•	•	•	•	•	•	•
ADC	•	•	•	•	•	•	On select	•
DAC	•		•	•		•		
UART		•	•	•	•	•	•	•
I ² C		•	•	•	•	•	•	•
SPI		•	•	•	•	•	•	•
Capacitive touch		•			•			
Multiplier		•	•	•	•	•		•
DMA			•	•	•	•		•
Op amps			•	•				
LCD				•	•	•		•
RTC				•	•	•		•
PMM					•	•	•	•
1.8-V I/O						•		
CRC					•	•	•	•
High-resolution timer						•		
USB						•		
Hardware encryption (AES)					•	•		•
FRAM					•		On select	
RF							13.56 MHz (ISO 15693 or ISO 14443B interface)	Sub-1GHz

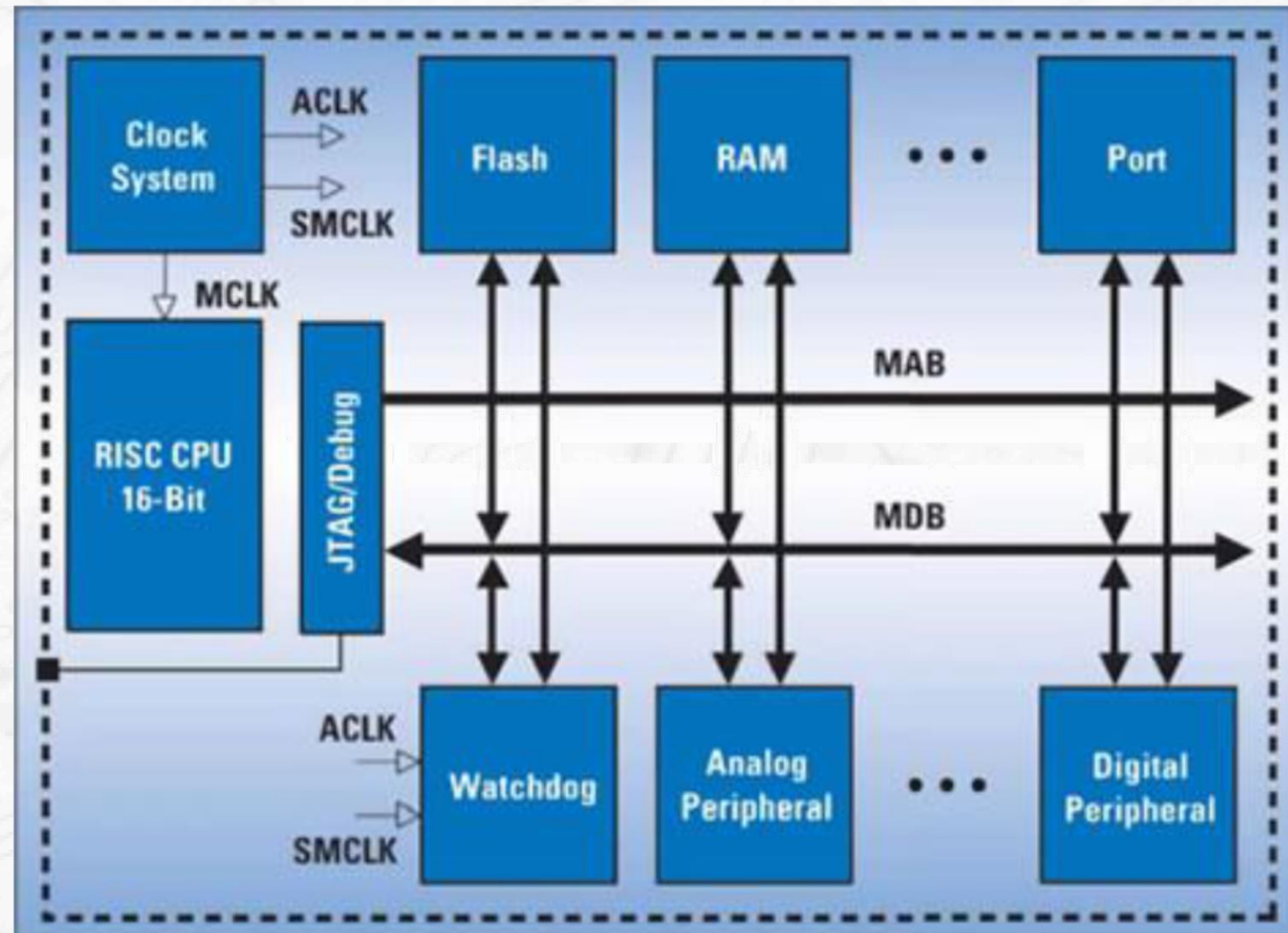
Low Power +
Performance

Ultra Low
Power

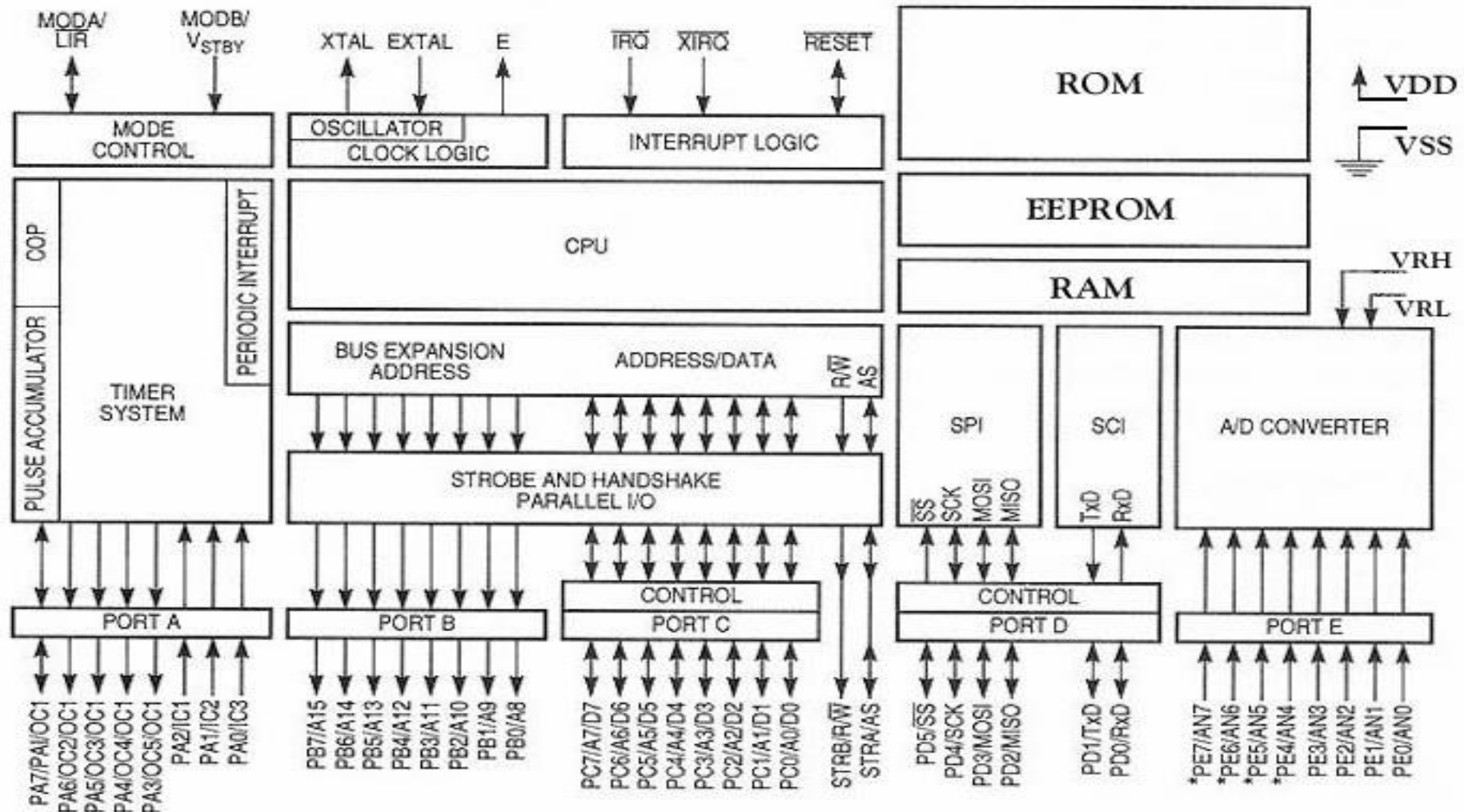
Security +
Comm

Αναπτυξιακό Σύστημα MSP430

Αρχιτεκτονική



Μικροελεγκτες



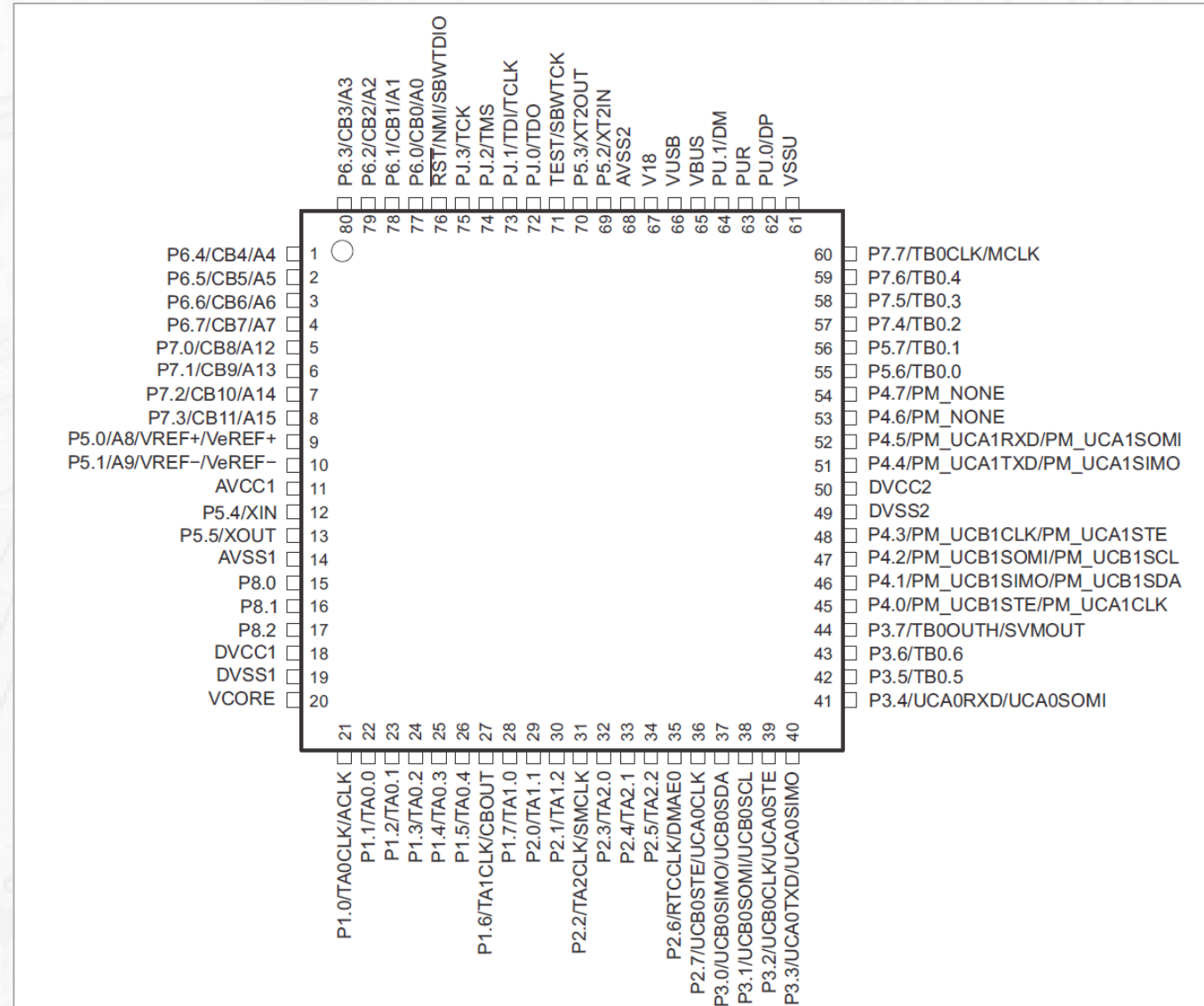
* NOT BONDED ON 48-PIN VERSION.

Αναπτυξιακό Σύστημα MSP430

Φύλλα δεδομένων

Φυλλάδια δεδομένων του MSP430:

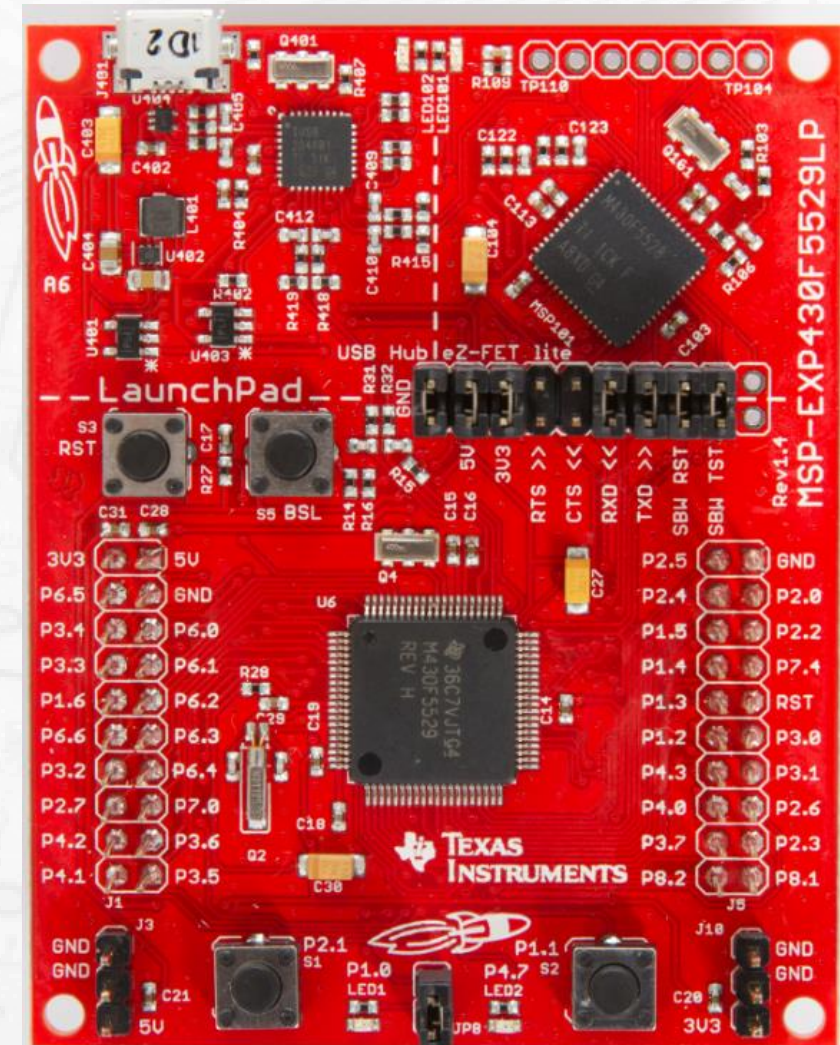
- Η διάταξη έχει ένα μεγάλο αριθμό περιφερειακών
- Κάθε ακροδέκτης εισόδου/εξόδου συνήθως έχει πάνω από μία λειτουργία;
- Περιλαμβάνει πίνακα περιγραφής της λειτουργίας κάθε ακροδέκτη
- Παράδειγμα, Αριθμός ακροδέκτη 2 = P6.5/A5;
- Ψηφιακή θύρα 6 εισόδου/εξόδου bit3;
- •3η αναλογική είσοδος.



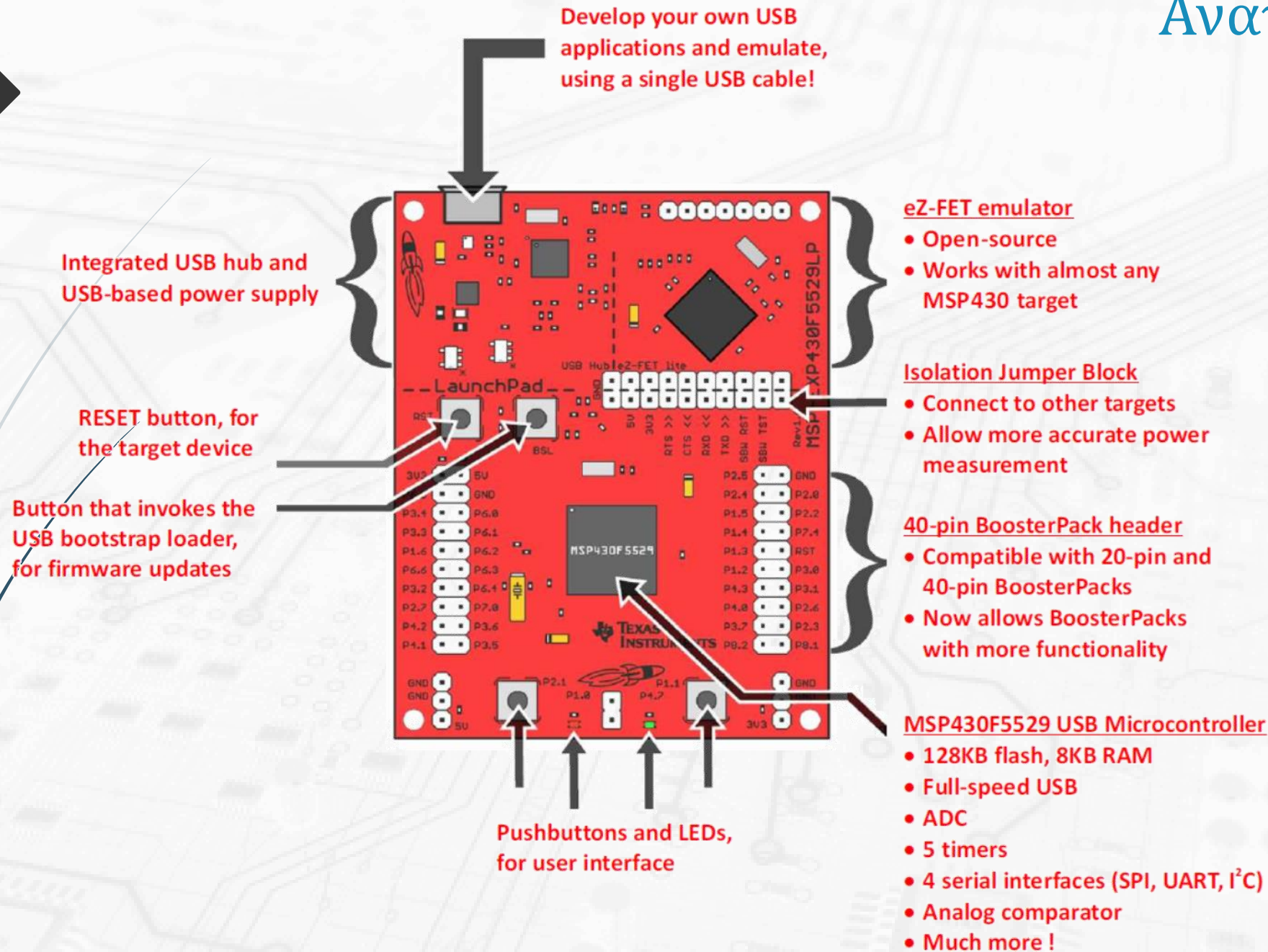
Αναπτυξιακό Σύστημα MSP430

MSP430

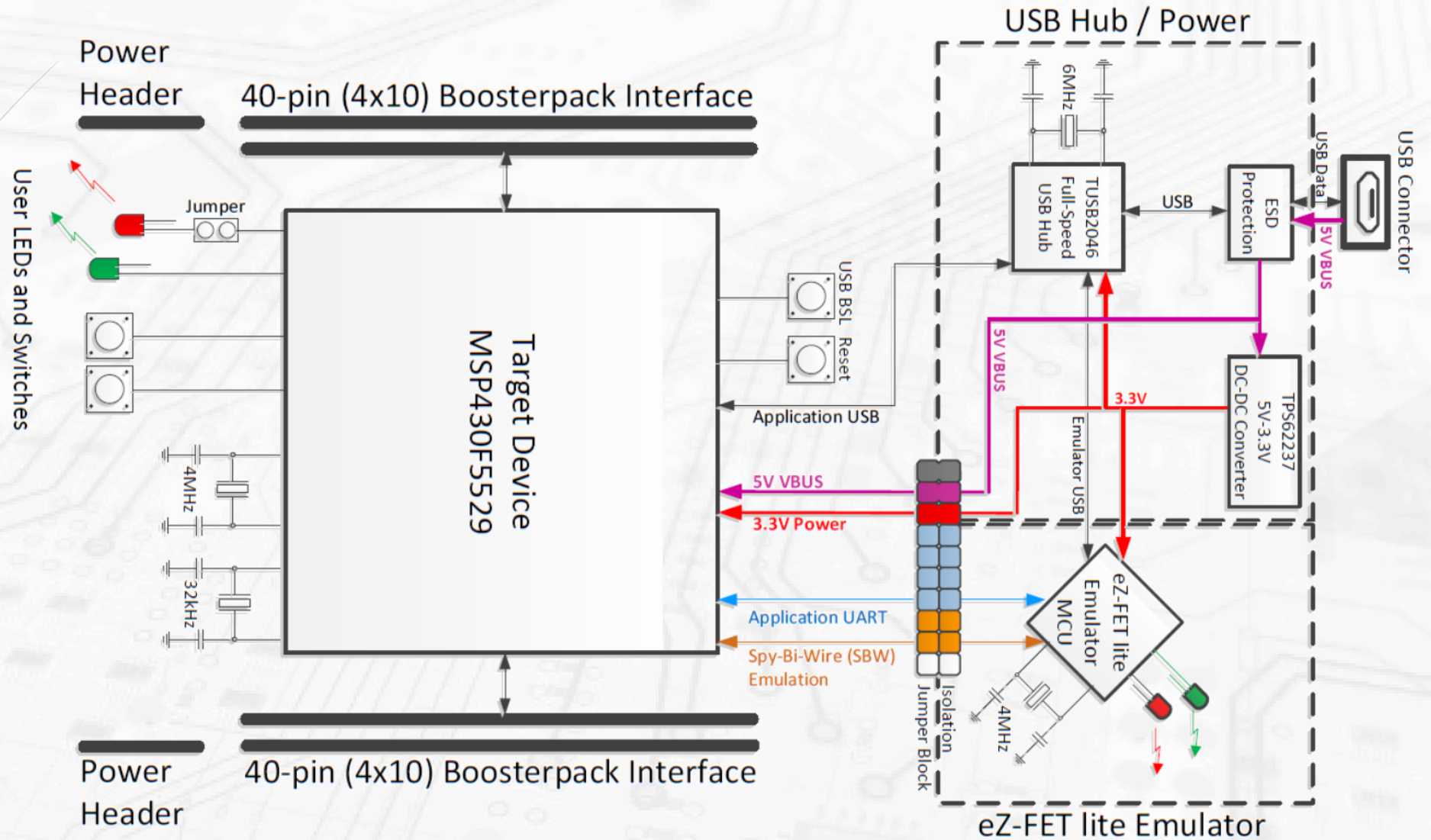
- Ένα αναπτυξιακό Σύστημα με σύνδεση USB
- Χαμηλού κόστους και εύχρηστο
- Είναι ιδανικό για την εξοικείωση αρχάριων χρηστών με μικρολεγκτές
- On-board εξομοιωτής με δυνατότητα απασφαλμάτωσης
- Παρέχει κουμπιά και LED για την εύκολη διεπαφή με τον χρήστη



Αναπτυξιακό Σύστημα MSP430



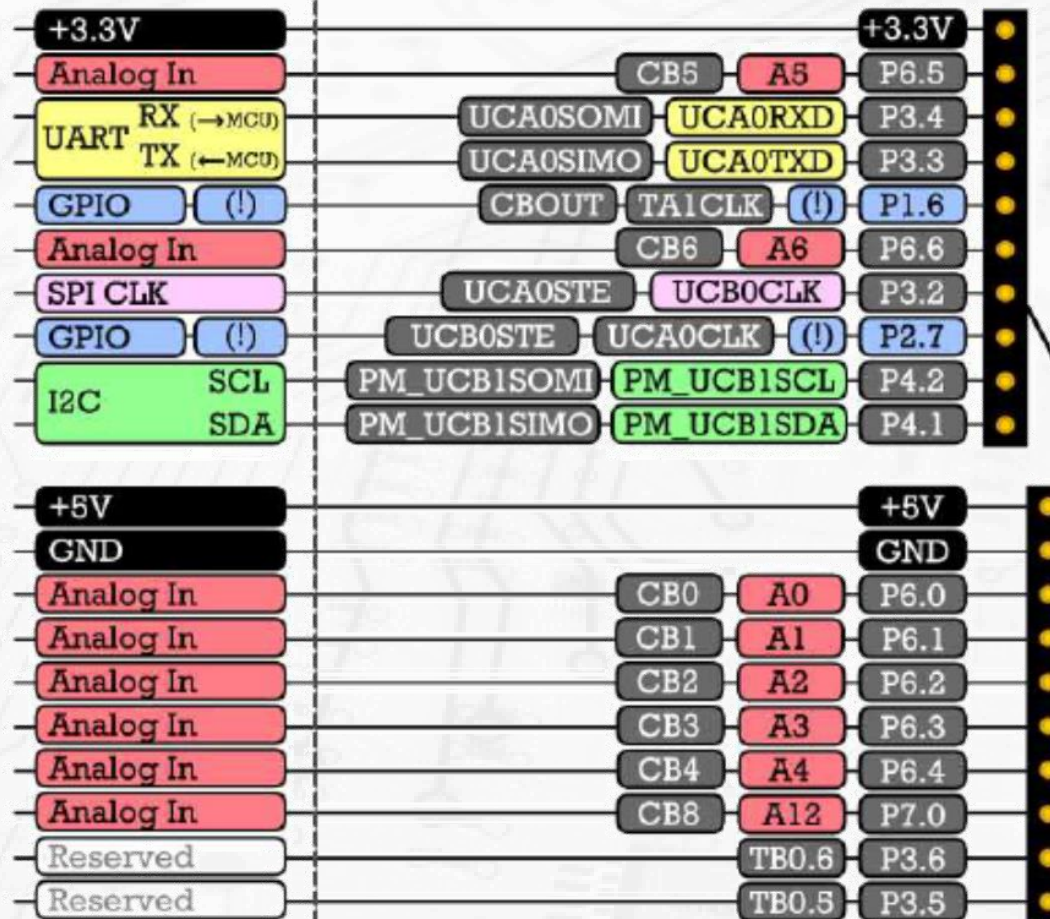
Αναπτυξιακό Σύστημα MSP430



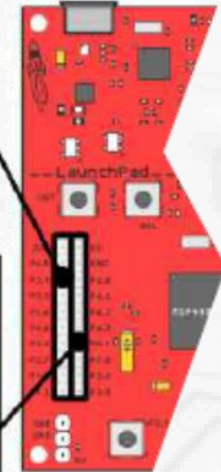
Αναπτυξιακό Σύστημα MSP430

BoosterPack Pinout Standard

Software-Configurable MSP430F5529 Pin Functions



F5529 LaunchPad

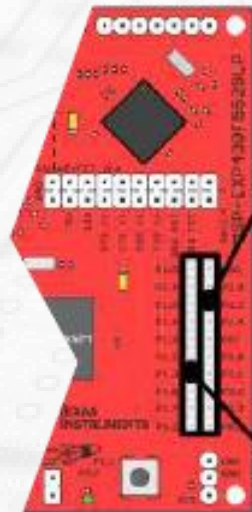


- Power
- Analog
- SPI
- I2C
- General I/O
- Unused function

(!) Denotes an interrupt-capable I/O

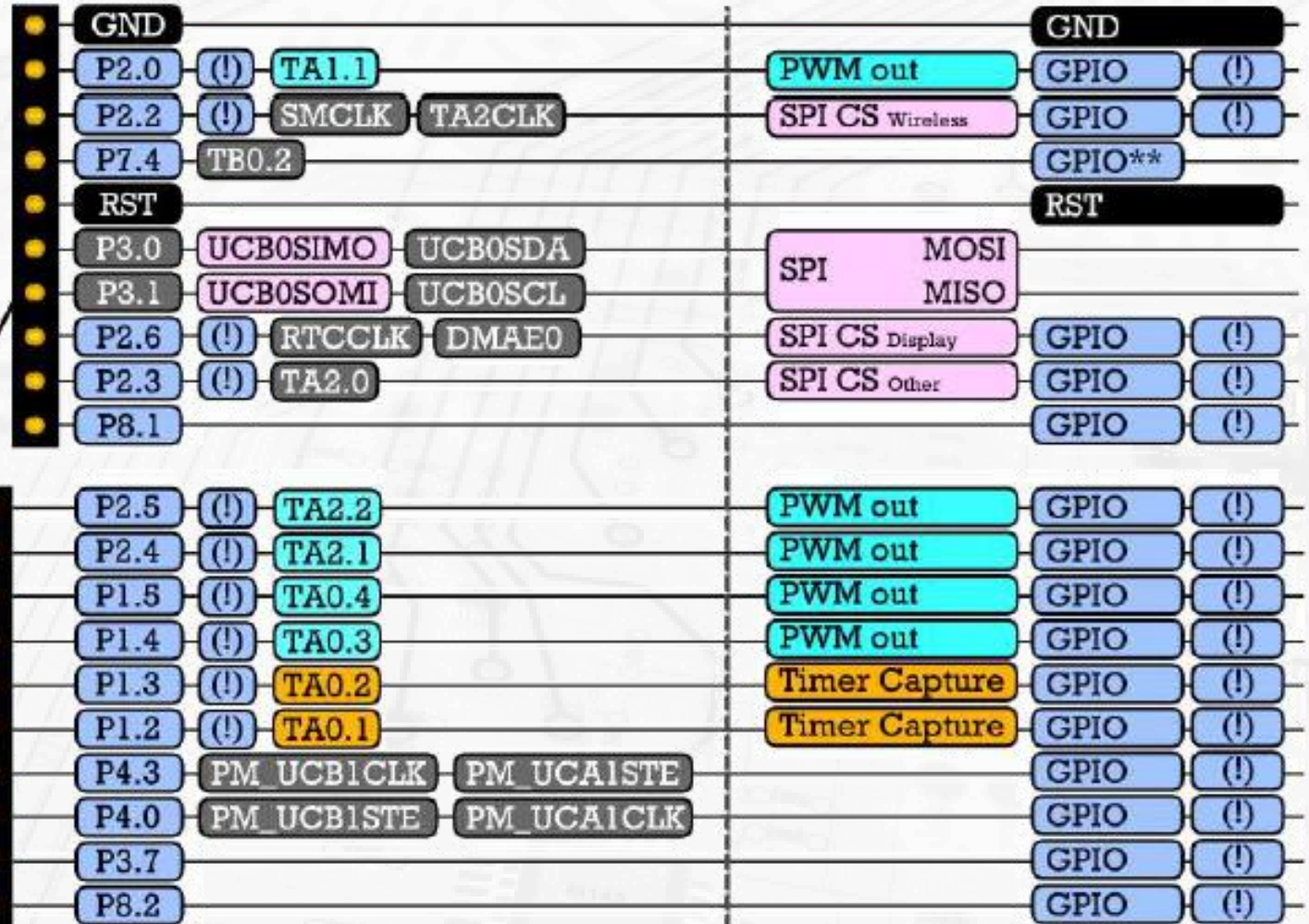
Αναπτυξιακό Σύστημα MSP430

F5529 LaunchPad



Software-Configurable
MSP430F5529 Pin Functions

BoosterPack
Pinout Standard



- Power
- Analog
- SPI
- I2C
- General I/O
- Unused function

Αναπτυξιακό Σύστημα MSP430

Φύλλα δεδομένων

Καταχωρητής Ελέγχου FLL_CTL1

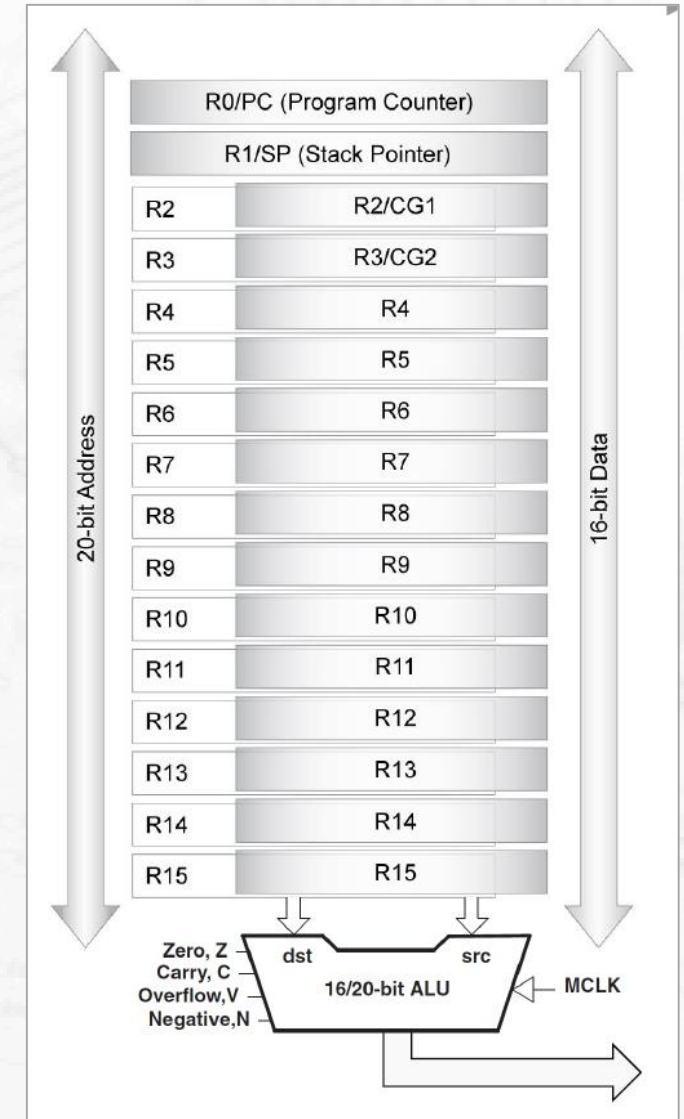
7	6	5	4	3	2	1	0
-	SMCLKOFF	XT2OFF	SELMx		SELS	FLL_DIVx	

Bit	Description	
6	SMCLKOFF	Disable the submain clock signal (SMCLK): SMCLKOFF = 0 ⇒ SMCLK active SMCLKOFF = 1 ⇒ SMCLK inactive
5	XT2OFF	Disable the second crystal oscillator (XT2): XT2OFF = 0 ⇒ XT2 active XT2OFF = 1 ⇒ XT2 inactive
4-3	SELMx	Select the master clock (MCLK) source: SELM1 SELM0 = 0 0 ⇒ DCO SELM1 SELM0 = 0 1 ⇒ DCO SELM1 SELM0 = 1 0 ⇒ XT2 SELM1 SELM0 = 1 1 ⇒ LFXT1
2	SELS	Select the submain clock (SMCLK) source: SELS = 0 ⇒ DCO SELS = 1 ⇒ XT2
1-0	FLL_DIVx	Select the auxiliary clock (ACLK) signal divider: FLL_DIV_0 = 0 0 ⇒ Divider factor: /1 FLL_DIV_1 = 0 1 ⇒ Divider factor: /2 FLL_DIV_2 = 1 0 ⇒ Divider factor: /4 FLL_DIV_3 = 1 1 ⇒ Divider factor: /8

Αναπτυξιακό Σύστημα MSP430

Αρχιτεκτονική

- **Ενεργειακά Αποδοτικός (Efficient) - Ultra-low power CPU**
- Φιλικό μεταγλωττιστή C/C++
- Αρχιτεκτονική RISC
 - 51 Εντολές
 - 7 τρόπους διευθυνσιοδότησης
 - Γεννήτρια Σταθερών (Constant Generator)
- Πράξεις καταχωρητών ενός κύκλου ρολογιού
- Επεξεργασία BIT, byte, λέξεων
- 1 MB ενιαίος χάρτης μνήμης (memory map)
 - Χωρίς σελιδοποίηση (paging)
- Τρόποι Διευθυνσιοδότηση
 - Χωρίς σελιδοποίηση (20-bit)
 - Ταχύτατη εκτέλεση
- 100% συμβατός κώδικας με τις προηγούμενες εκδόσεις



Αναπτυξιακό Σύστημα MSP430

Αρχιτεκτονική-Μνήμη

MEMORY MAP

In-System Prog (ISP)

- Write using: User program, JTAG, BSL
- Byte, word, long-word
- Erase one (or all) segments at a time

Main Flash

- 512 byte segments
- Start address moves according to RAM

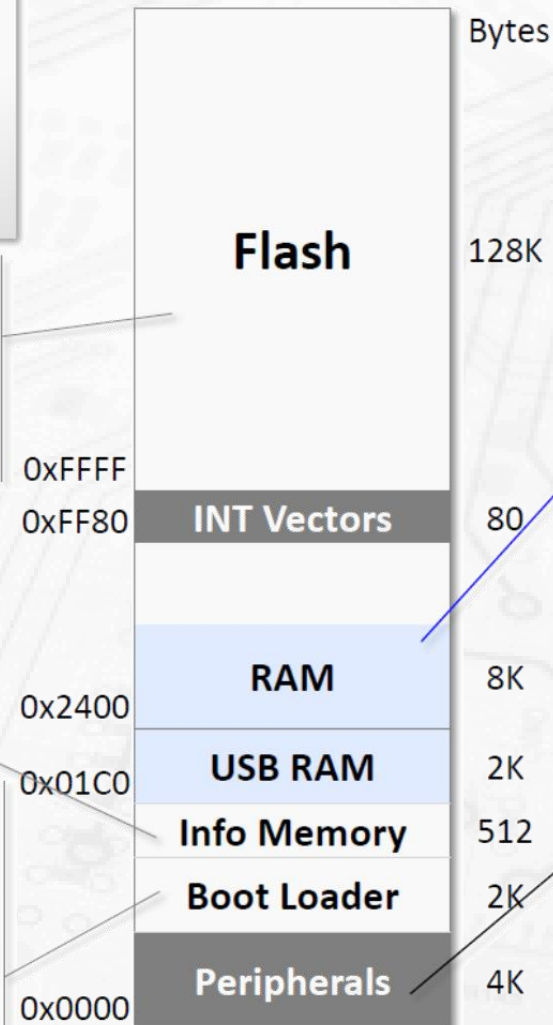
Info Memory

- Use for your own calibration data, etc.
- 4 segments (A-D)
- 128 byte segments

Boot Loader (BSL)

- Program Flash/RAM with serial (slau319)
- Password protected
- 512 byte segments

'F5529 Memory Map



MSP430 Memory

- Unified memory map (program or data)
- Absolutely no paging

RAM

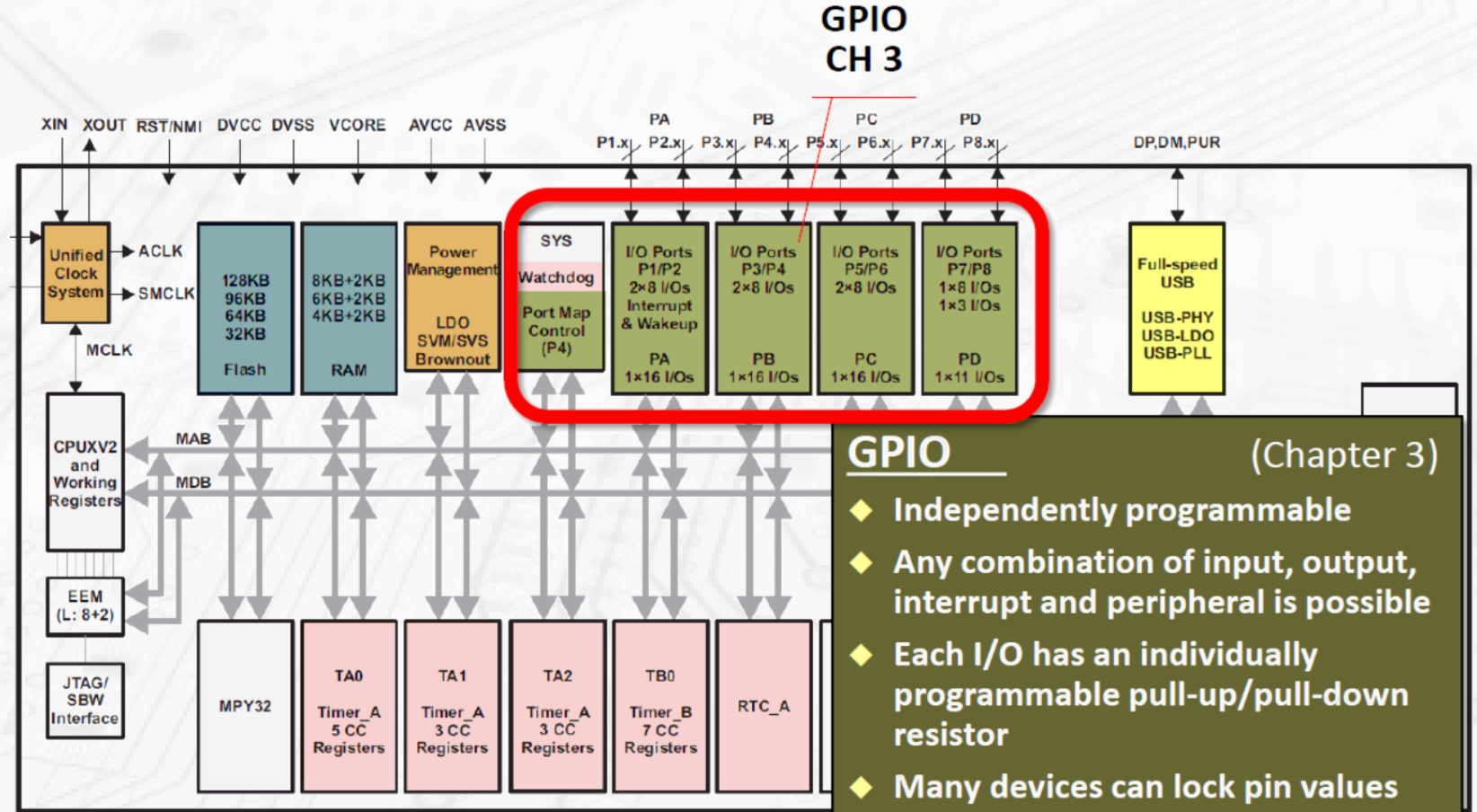
- Always a config. block
- If enabled, USB port uses first 2K
- RAM segments can be powered down

Device Descriptors (TLV)

- Factory calibration data, periph support,...
- Found in peripherals (at 0x1A00)

Αναπτυξιακό Σύστημα MSP430

Αρχιτεκτονική-GPIOs



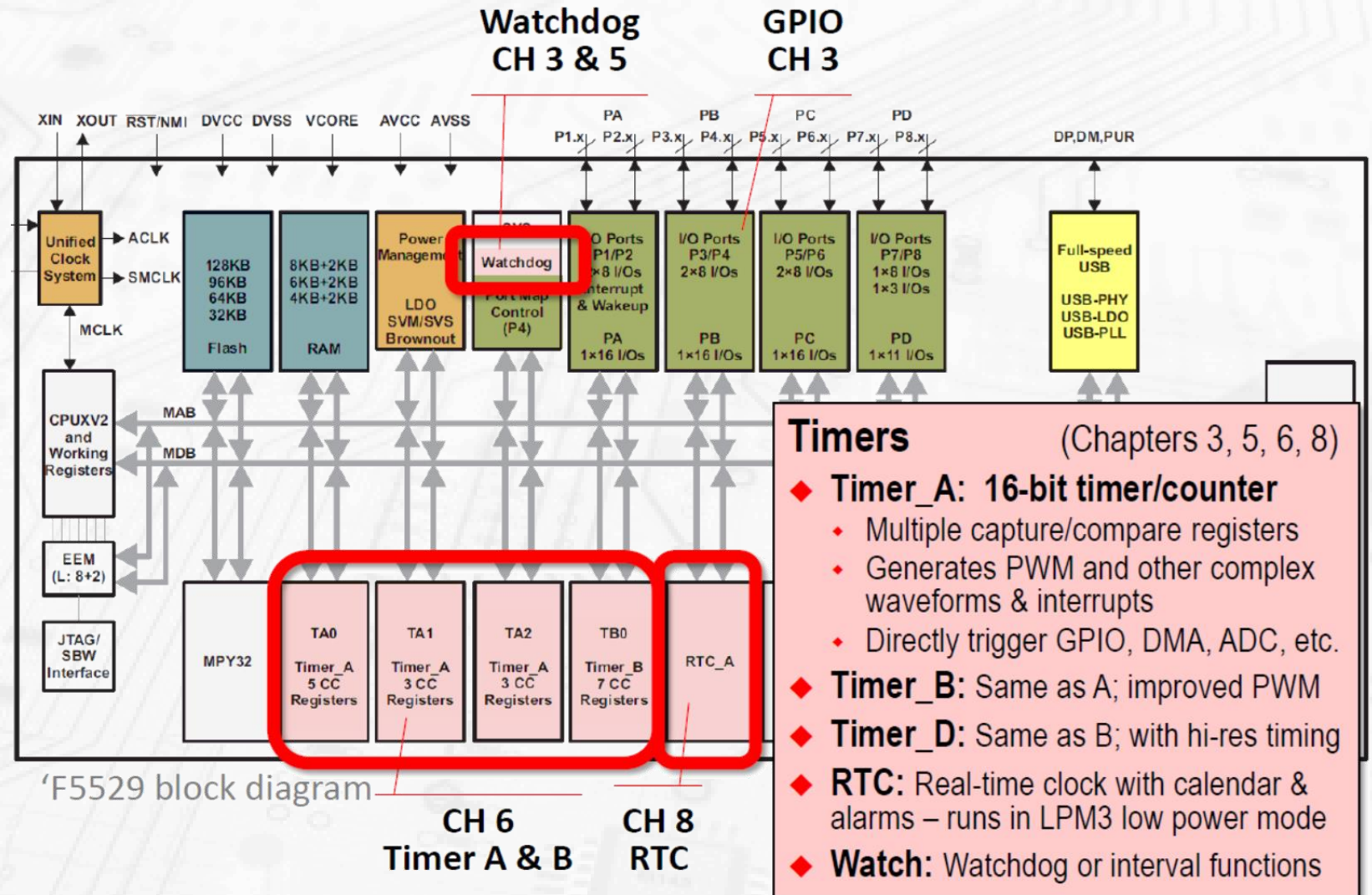
'F5529 block diagram

GPIO (Chapter 3)

- ◆ Independently programmable
- ◆ Any combination of input, output, interrupt and peripheral is possible
- ◆ Each I/O has an individually programmable pull-up/pull-down resistor
- ◆ Many devices can lock pin values during low-power modes
- ◆ Some devices support touch-sense capability built into the pins

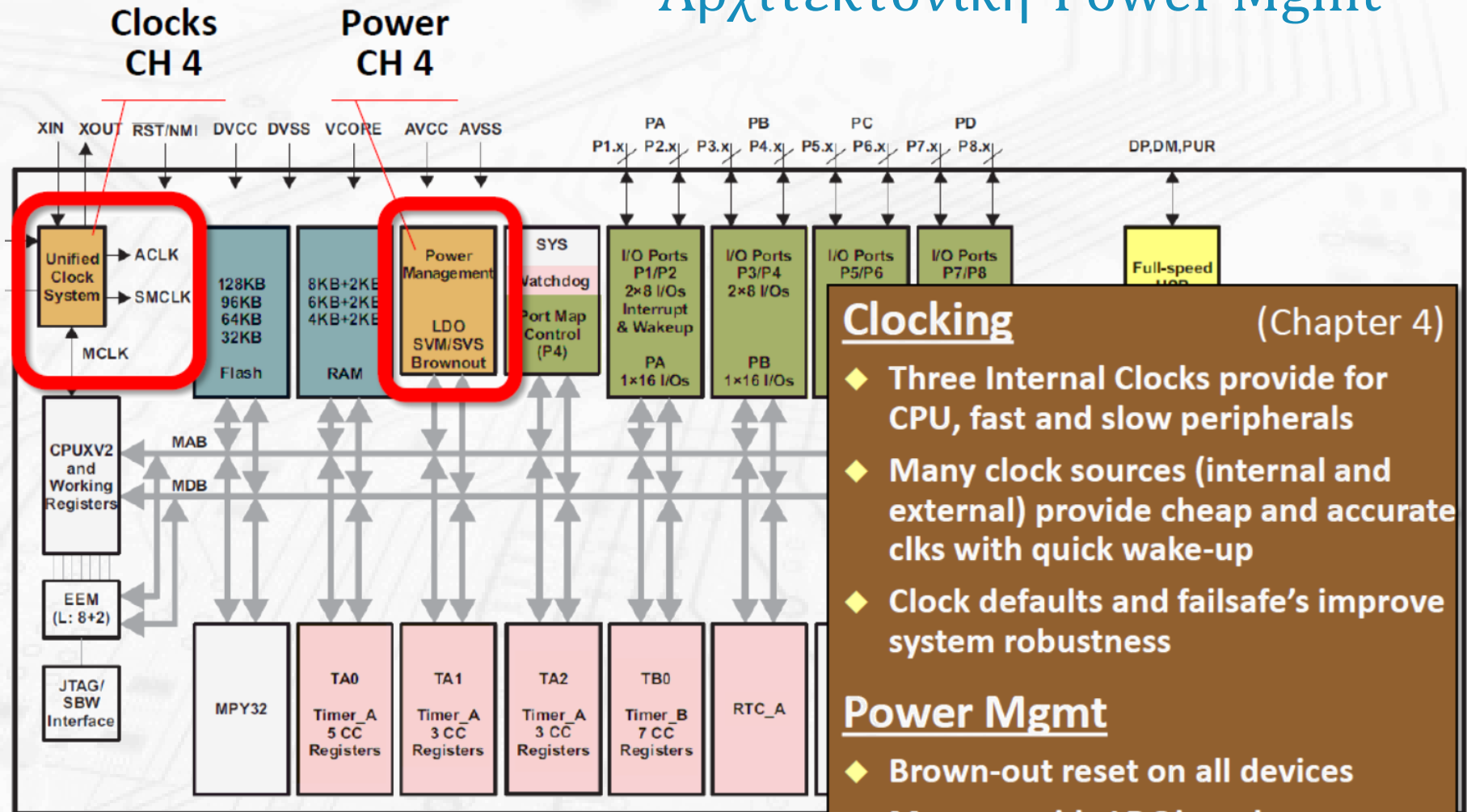
Αναπτυξιακό Σύστημα MSP430

Αρχιτεκτονική-Timers



Αναπτυξιακό Σύστημα MSP430

Αρχιτεκτονική-Power Mgmt



'F5529 block diagram

Clocking

(Chapter 4)

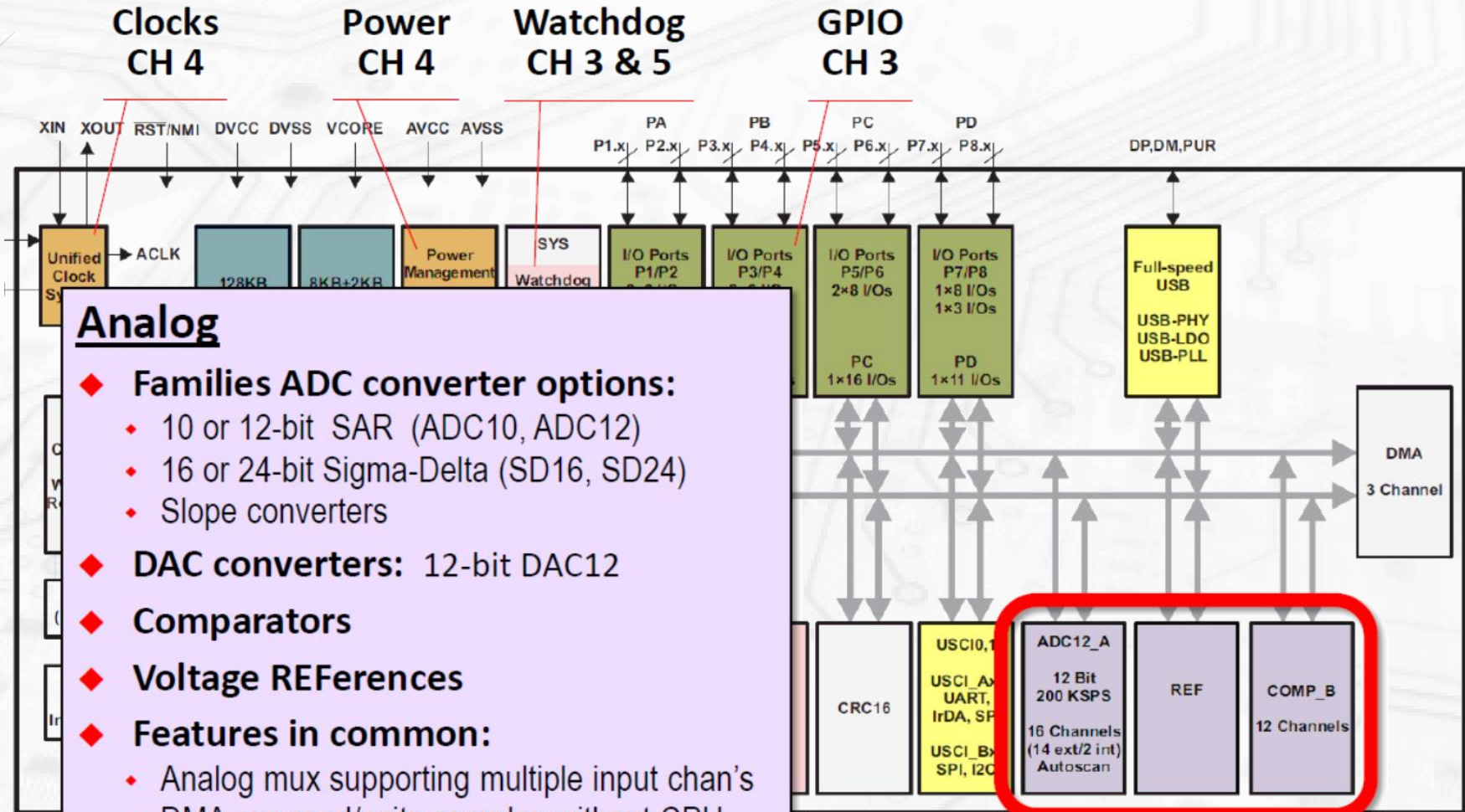
- ◆ Three Internal Clocks provide for CPU, fast and slow peripherals
- ◆ Many clock sources (internal and external) provide cheap and accurate clks with quick wake-up
- ◆ Clock defaults and failsafe's improve system robustness

Power Mgmt

- ◆ Brown-out reset on all devices
- ◆ Many provide LDO's and power supervisors
- ◆ On-chip power gating drives ULP

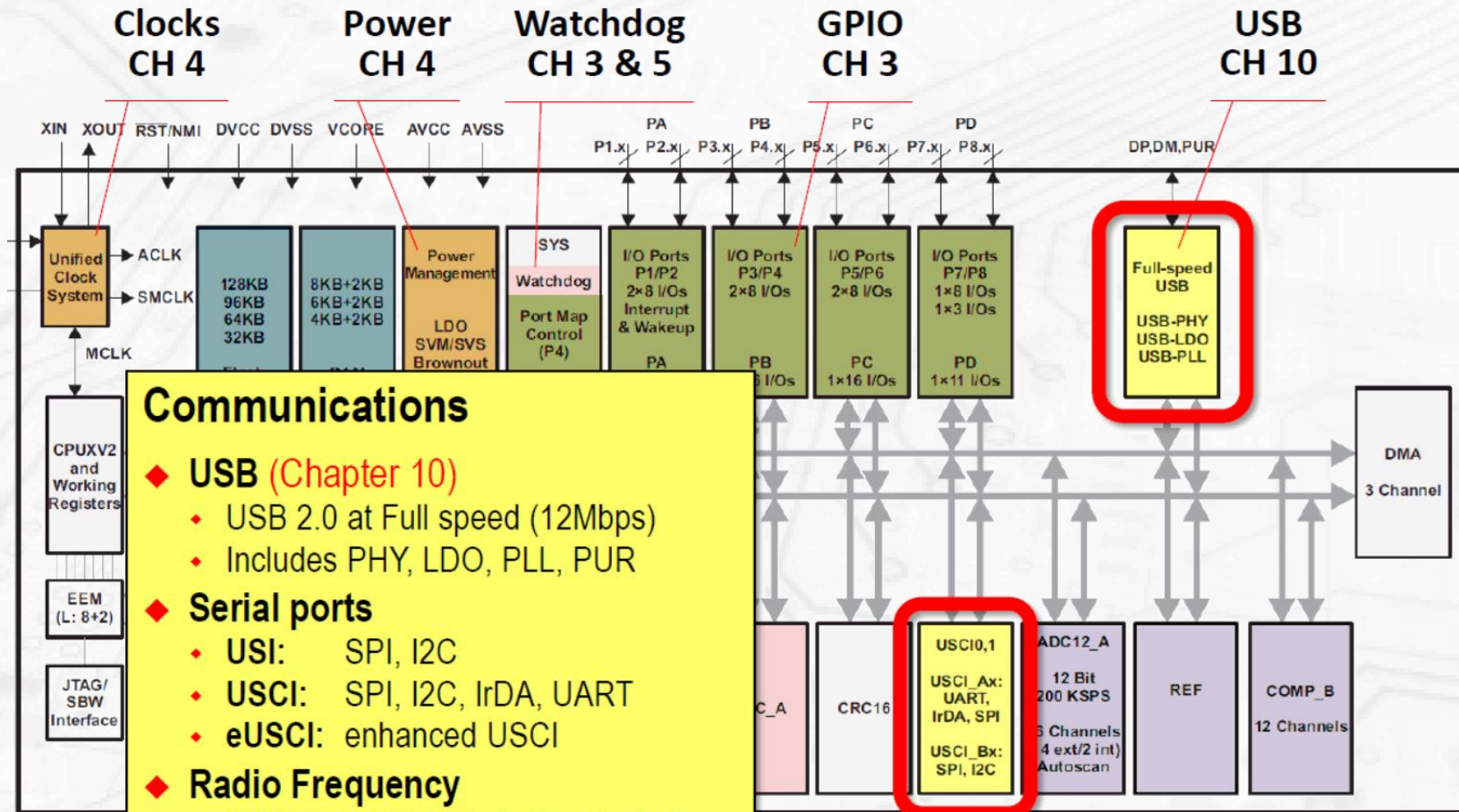
Αναπτυξιακό Σύστημα MSP430

Αρχιτεκτονική-Αναλογικό



Αναπτυξιακό Σύστημα MSP430

Αρχιτεκτονική-Επικοινωνία



Communications

◆ USB (Chapter 10)

- ◆ USB 2.0 at Full speed (12Mbps)
- ◆ Includes PHY, LDO, PLL, PUR

◆ Serial ports

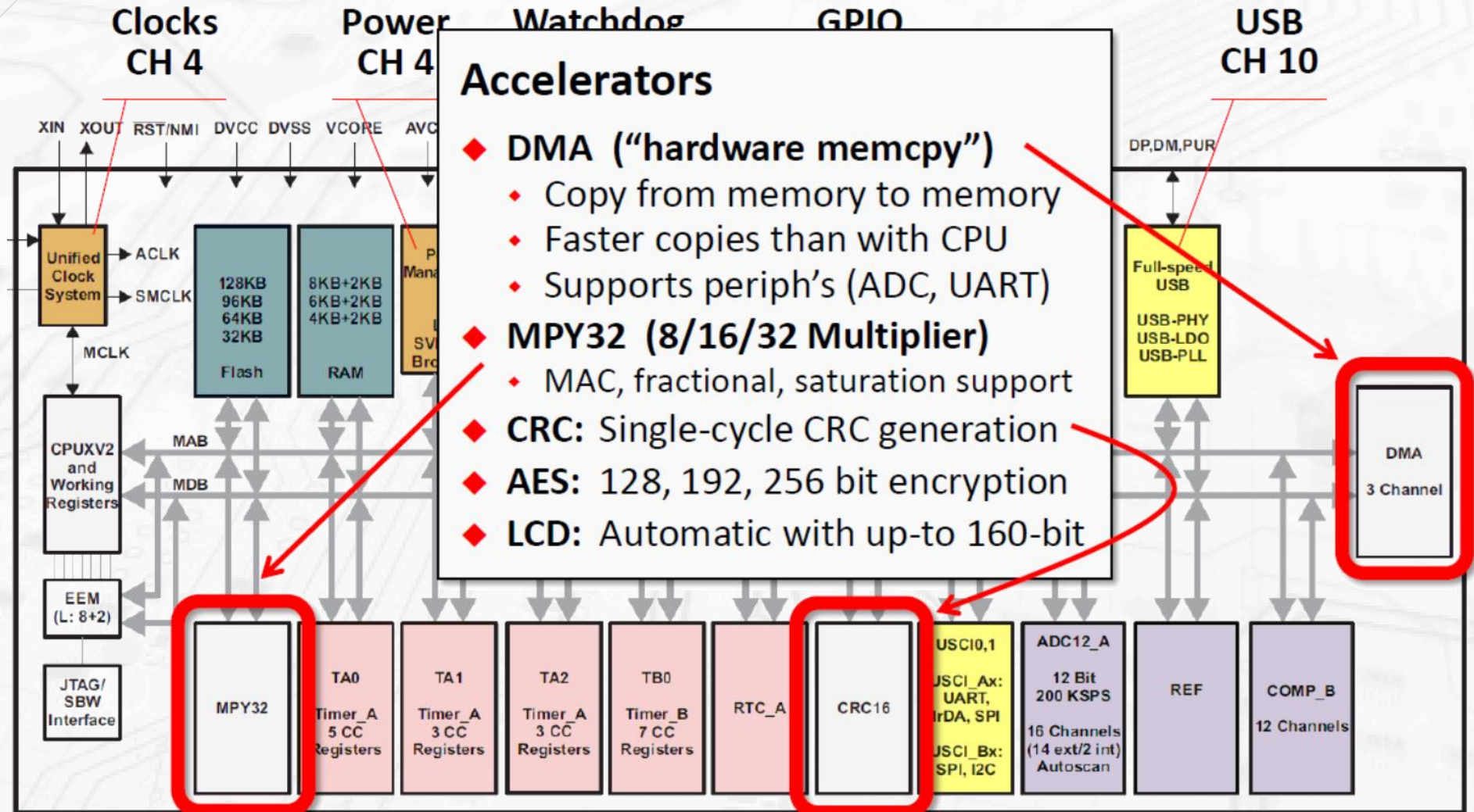
- ◆ USI: SPI, I2C
- ◆ USCI: SPI, I2C, IrDA, UART
- ◆ eUSCI: enhanced USCI

◆ Radio Frequency

- ◆ CC430 and RF430 devices include Sub-1GHz or NFC radios
- ◆ WiFi, BLE, ANT, Bluetooth & Sub1GHz communications via TI SimpleLink

Αναπτυξιακό Σύστημα MSP430

Αρχιτεκτονική-Επιταχυντές



'F5529 block diagram

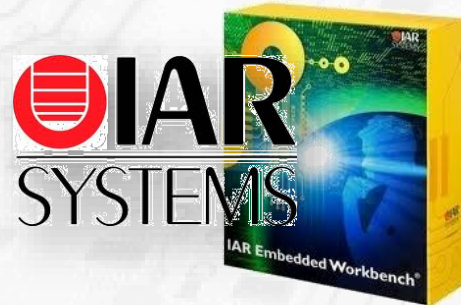
Code Composer Studio IDEs

Free Integrated Development Environments (IDE) available



Code Composer Studio

- Eclipse-based IDE (Compiler, debugger, linker, etc) for all TI embedded processors
- Unrestricted version available for \$495
- Free versions are available!
 - Free 16kB code-limited version available for download
 - Free, full-featured, 120-day trial version available



IAR Embedded Workbench

- Strong third-party IDE offering with project management tools and editor. Includes config files for all MSP430 devices.
- Free versions are available!
 - Free 4/8/16kB code-limited Kickstart version available for download
 - Free, full-featured, 30-day trial version available

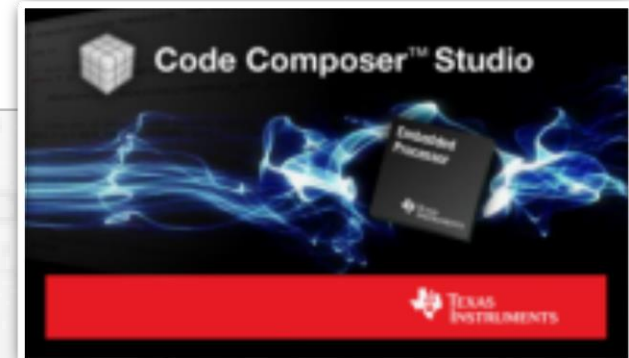


MSPGCC

- Free, Open source, GCC tool chain for MSP430
- includes the GNU C compiler (GCC), the assembler and linker (binutils), the debugger (GDB)
- Tools can be used on Windows, Linux, BSD and most other flavors of Unix.
- Learn more @ <http://mspgcc.sourceforge.net/>

Code Composer Studio

- ◆ **Integrated development environment for TI embedded processors**
 - Includes debugger, compiler, editor, simulator, OS...
 - The IDE is built on the Eclipse open source software framework
 - Extended by TI to support device capabilities
- ◆ **CCSv5 is based on “off the shelf” Eclipse (version 3.7 in CCS 5.1)**
 - Future CCS versions will use **unmodified** versions of Eclipse
 - TI contributes changes directly to the open source community
 - Drop in Eclipse plug-ins from other vendors or take TI tools and drop them into an existing Eclipse environment
 - Users can take advantage of all the latest improvements in Eclipse
- ◆ **Integrate additional tools**
 - OS application development tools (Linux, Android...)
 - Code analysis, source control...
- ◆ **Linux support soon**



Code Composer Studio

Κοινή χρήση

◆ Creating New Projects

- Very simple to create a new project for a device using a template

◆ Build options

- Many users have difficulty using the build options dialog and find it overwhelming
- Updates to options are delivered via compiler releases and not dependent on CCS updates

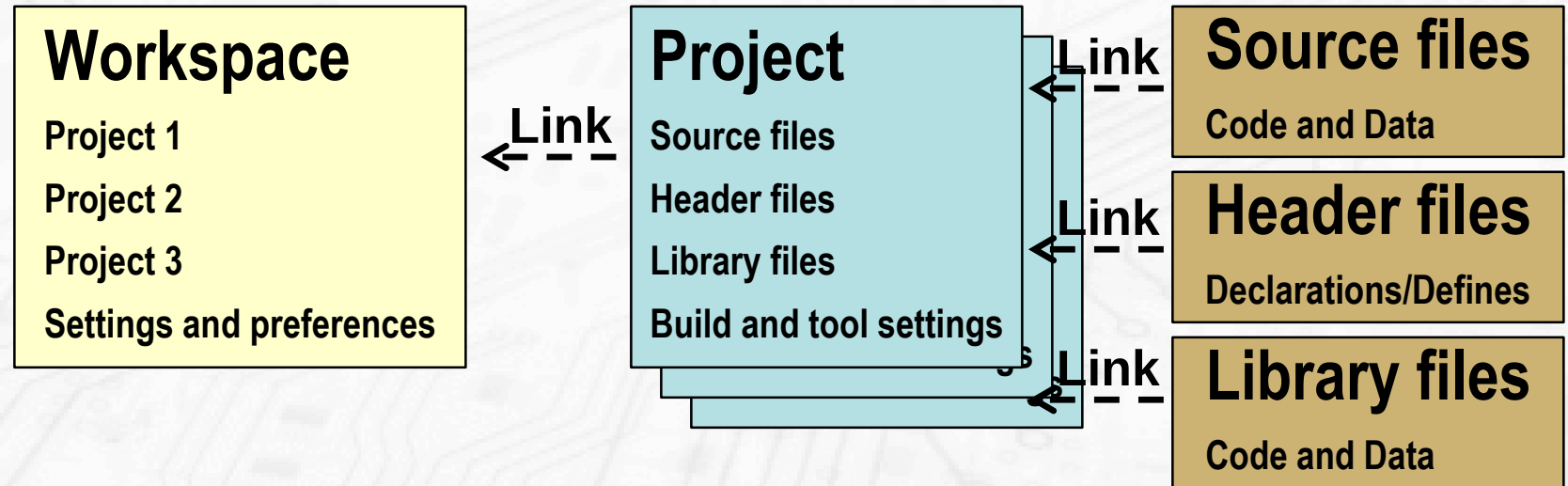
◆ Sharing projects

- Easy for users to share projects, including working with version control (portable projects)
- Setting up linked resources has been simplified



Code Composer Studio

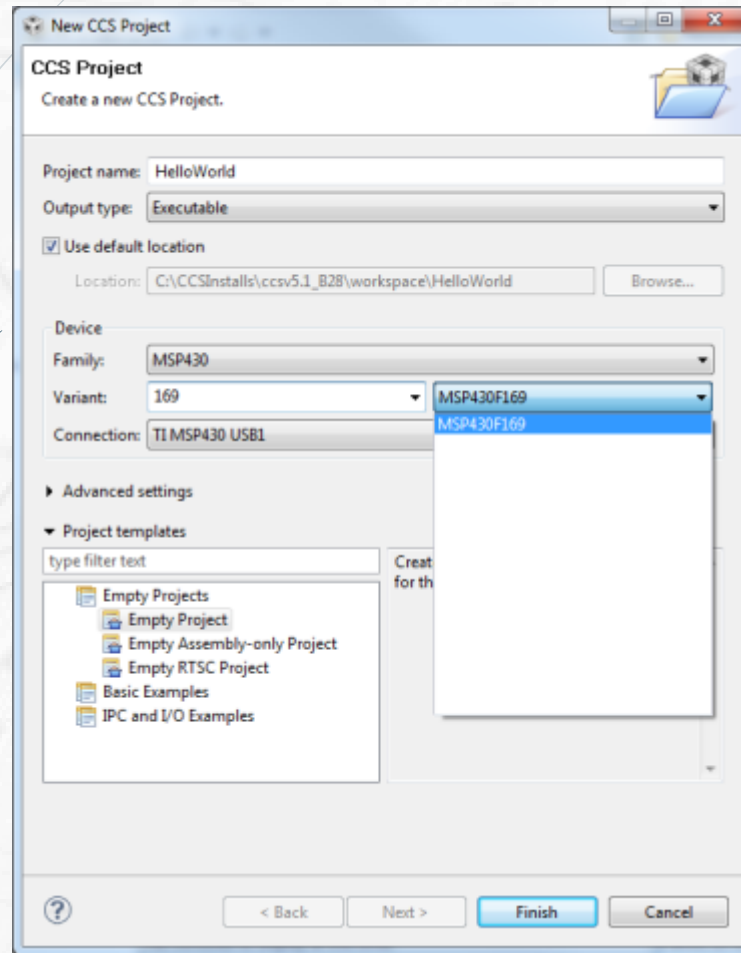
Projects



A workspace contains your settings and preferences, as well as links to your projects. Deleting projects from the workspace deletes the links, not the files

A project contains your build and tool settings, as well as links to your input files. Deleting files from the workspace deletes the links, not the files

Code Composer Studio Projects



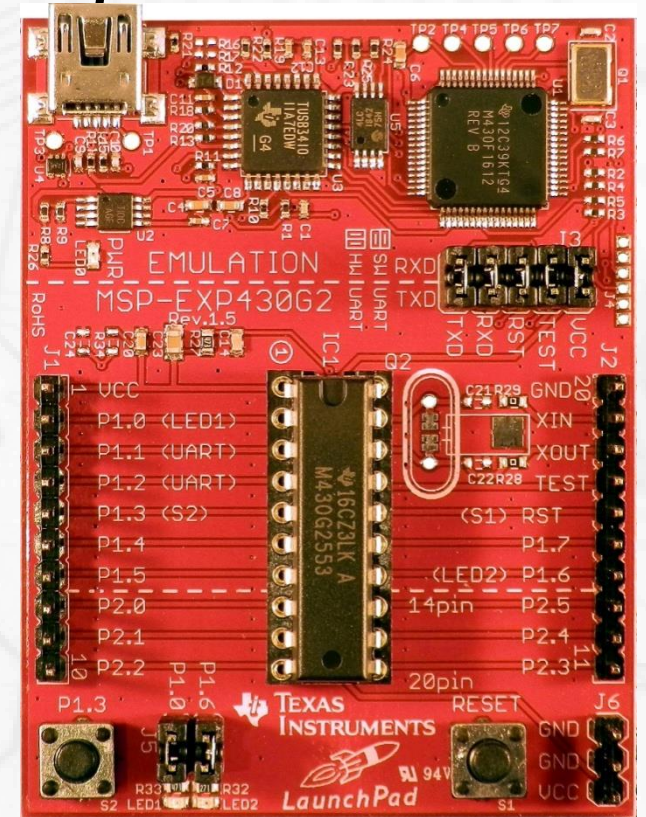
- ◆ **Single page wizard for majority of users**
 - Next button will show up if a template requires additional settings
- ◆ **Debugger setup included**
 - If a specific device is selected, then user can also choose their connection, ccxml file will be created
- ◆ **Simple by default**
 - Compiler version, endianness... are under advanced settings



Code Composer Studio Projects



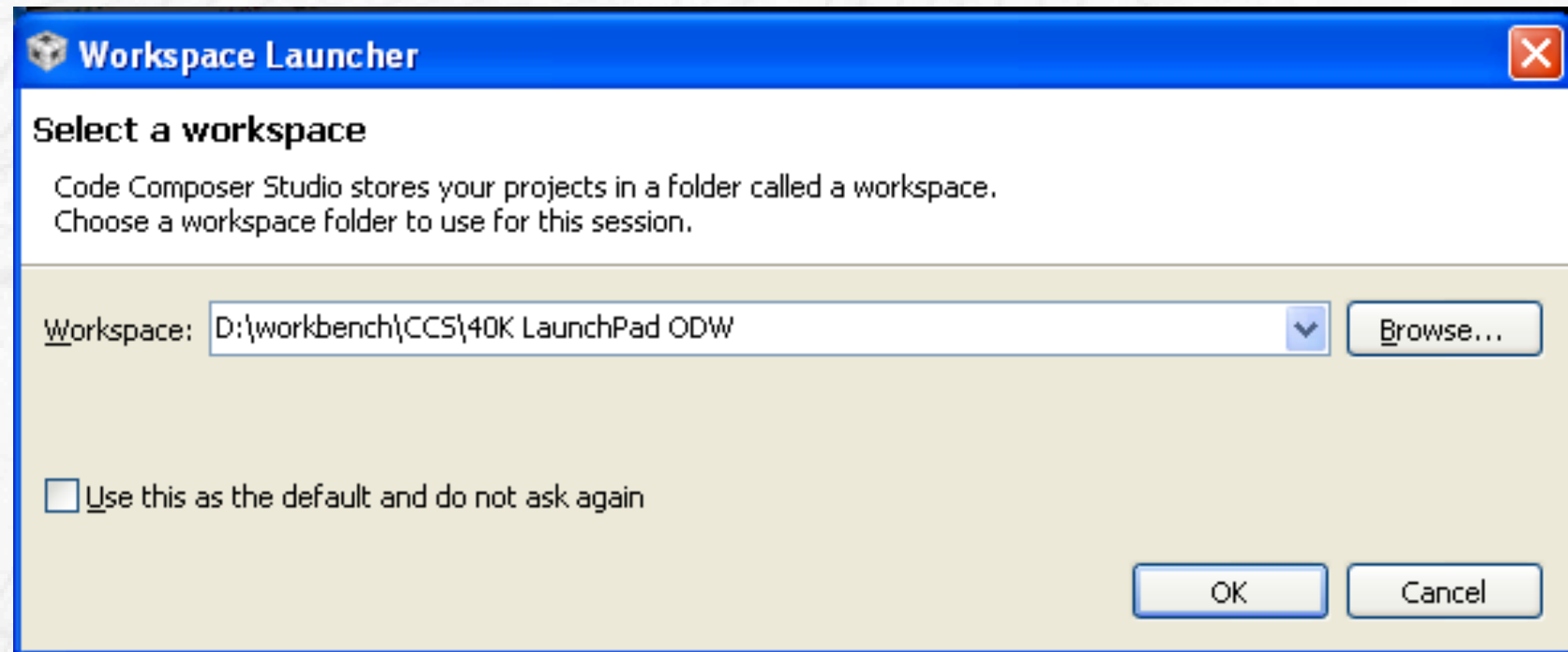
- Create a new workspace
- Create Lab1 Project
- Add in temperature sense demo
- Compile it and run



Code Composer Studio Projects

Step 1: Create CCS workspace

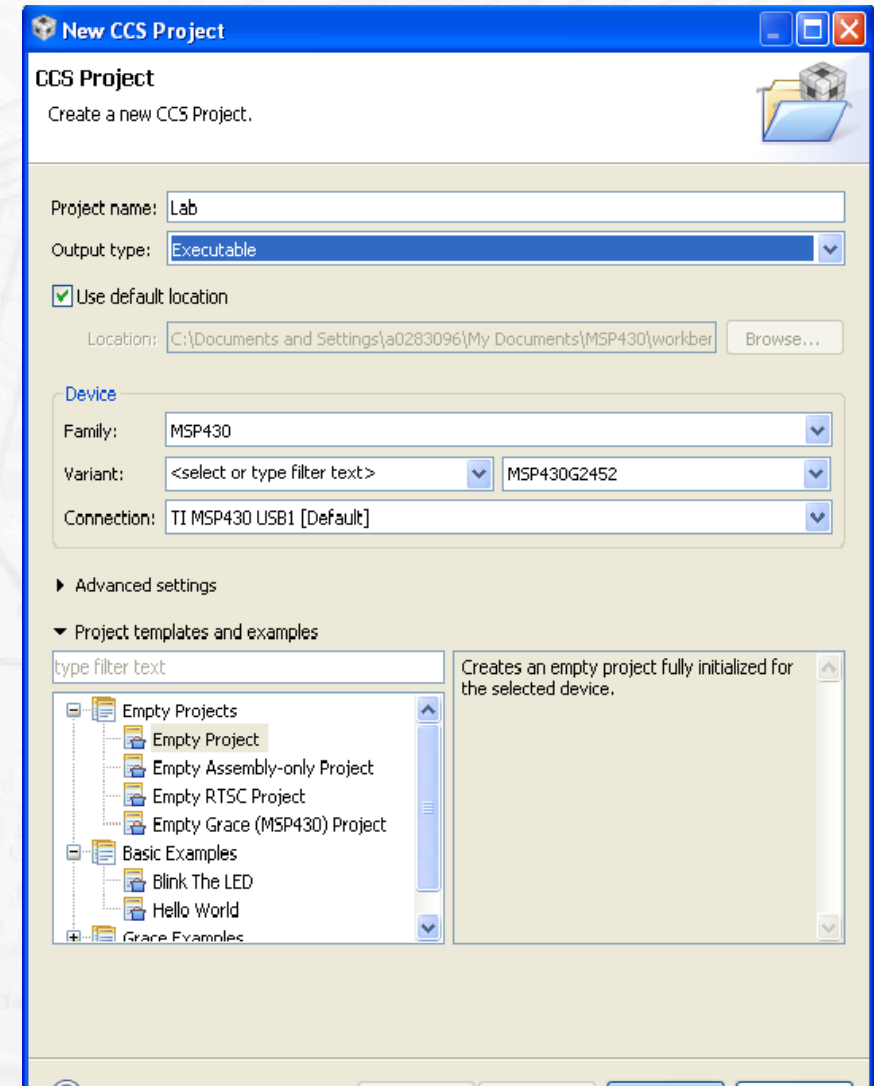
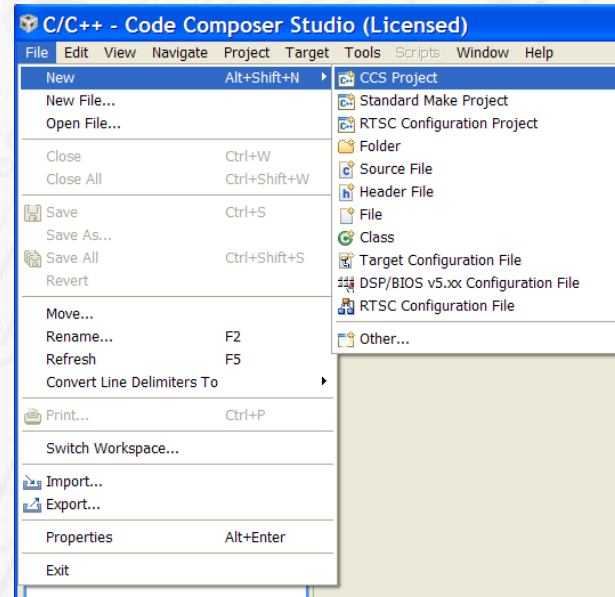
- Put the Lab files onto your desktop
- Launch CCS v5 Core Edition
- Select a “Workspace” location



Code Composer Studio Projects

Step 2: Create CCS project

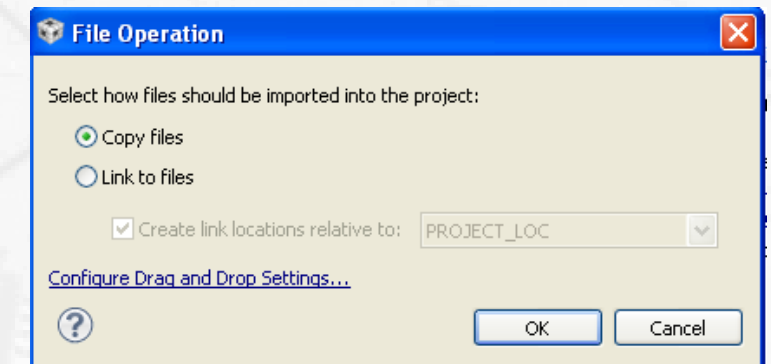
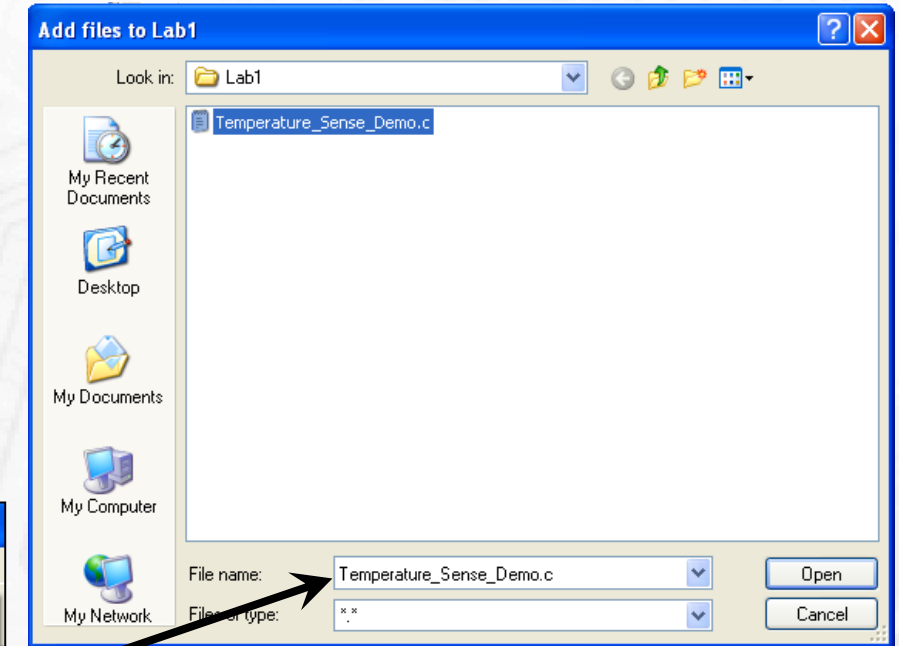
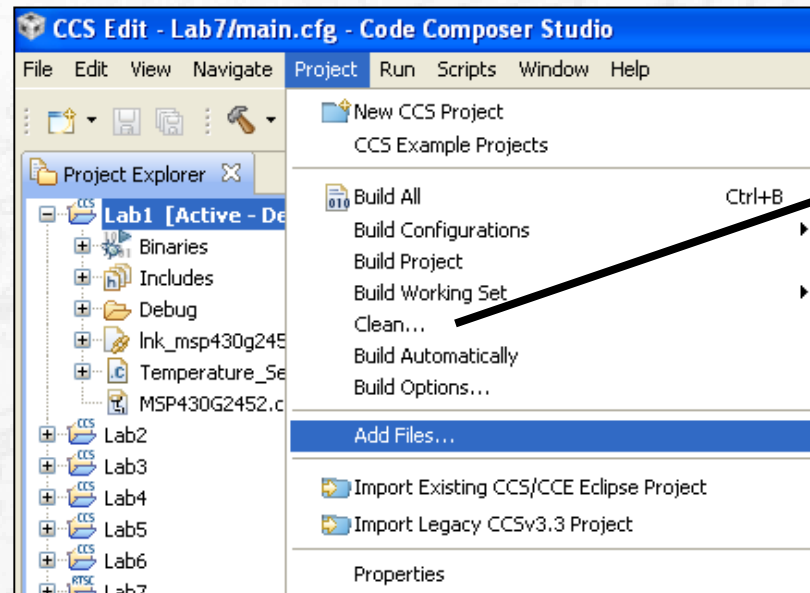
- File > New > CCS Project
- Project Name: **Lab1**
- Device>Family: **MSP430**
- Variant: **MSP430F5529**
- Project templates and examples
- : **Empty Project**



Code Composer Studio Projects

Step 3: Add a File to the CSS Project

- Project > Add Files
- Navigate to **Lab source folder**
- And select :
Temperature_Sense_Demo.c



Code Composer Studio Environment

1-click project Debug

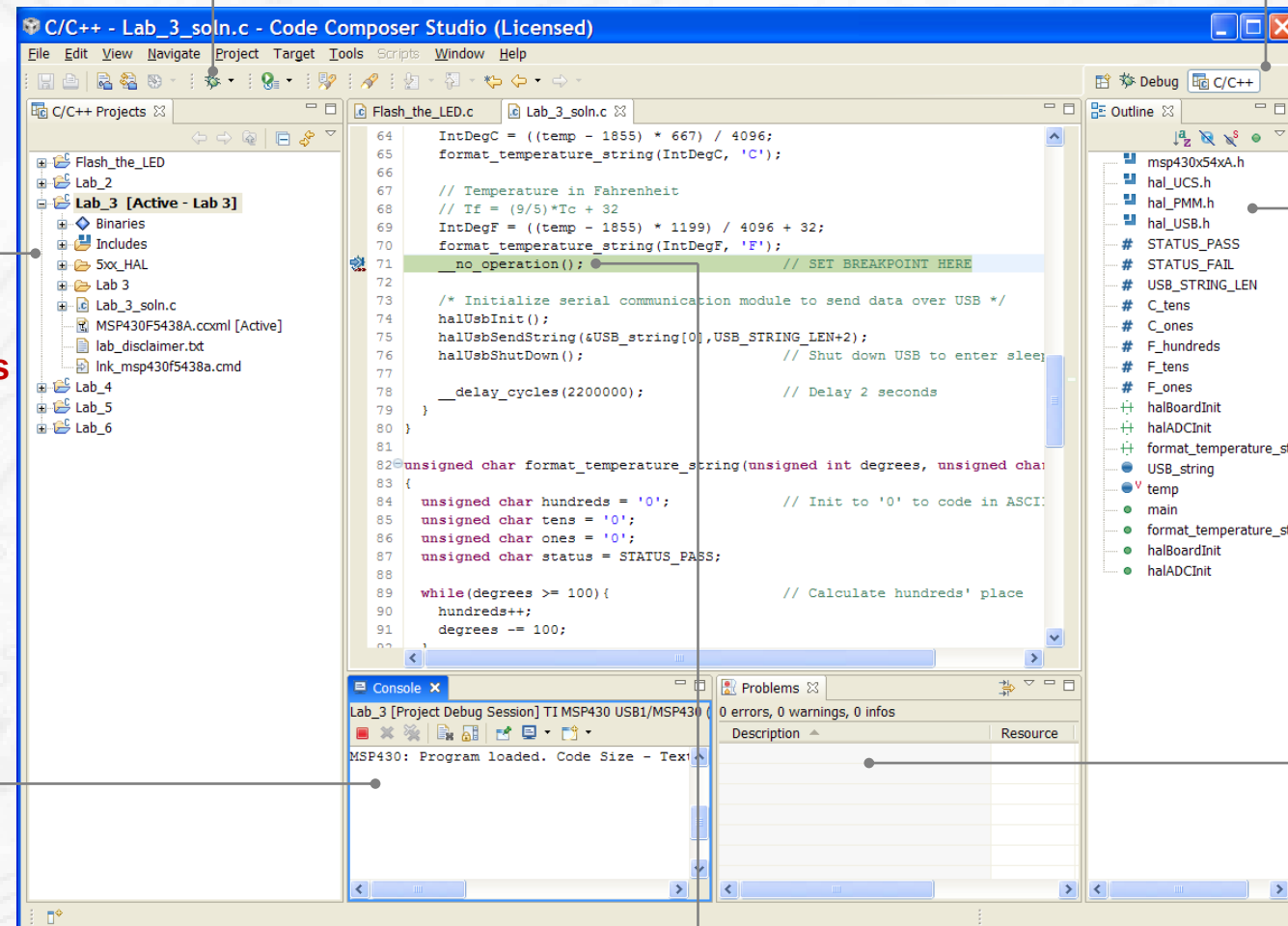
Project View
• List of all Projects

Project Outline
• Shortcut to project parts

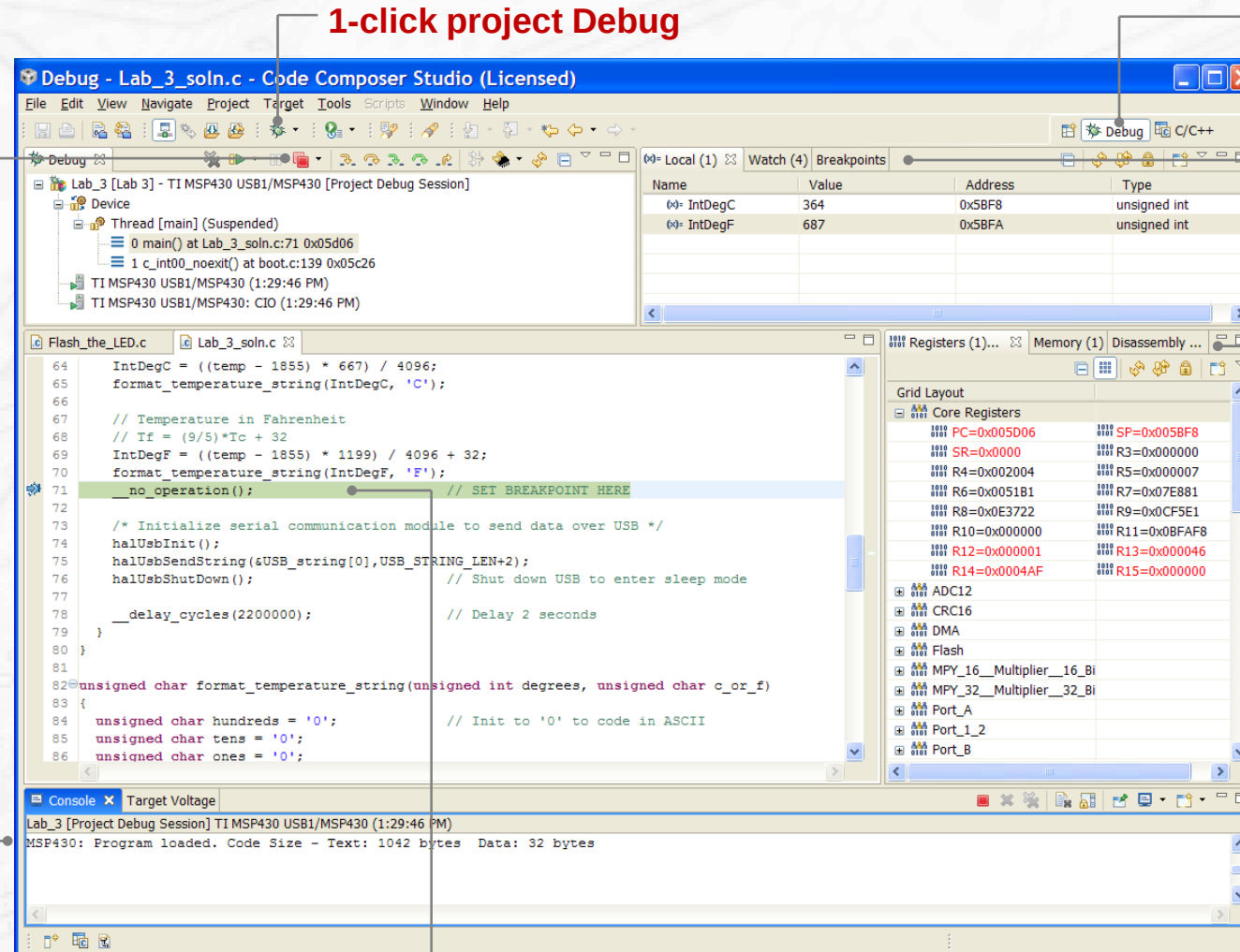
Console
• Build Information

Code Window
• Real-time breakpoints, Syntax highlighting

Problems View
• Information, Warnings, Errors



Code Composer Studio Environment



Target control

- Start
- Stop
- Halt
- Stepping
- Stack Trace

Program Size Info

Code Window

- Real-time breakpoints, Syntax highlighting

Independent Debug
and C/C++ Project
Perspectives

Highly configurable
window layout

- User preferences
- Plugin support

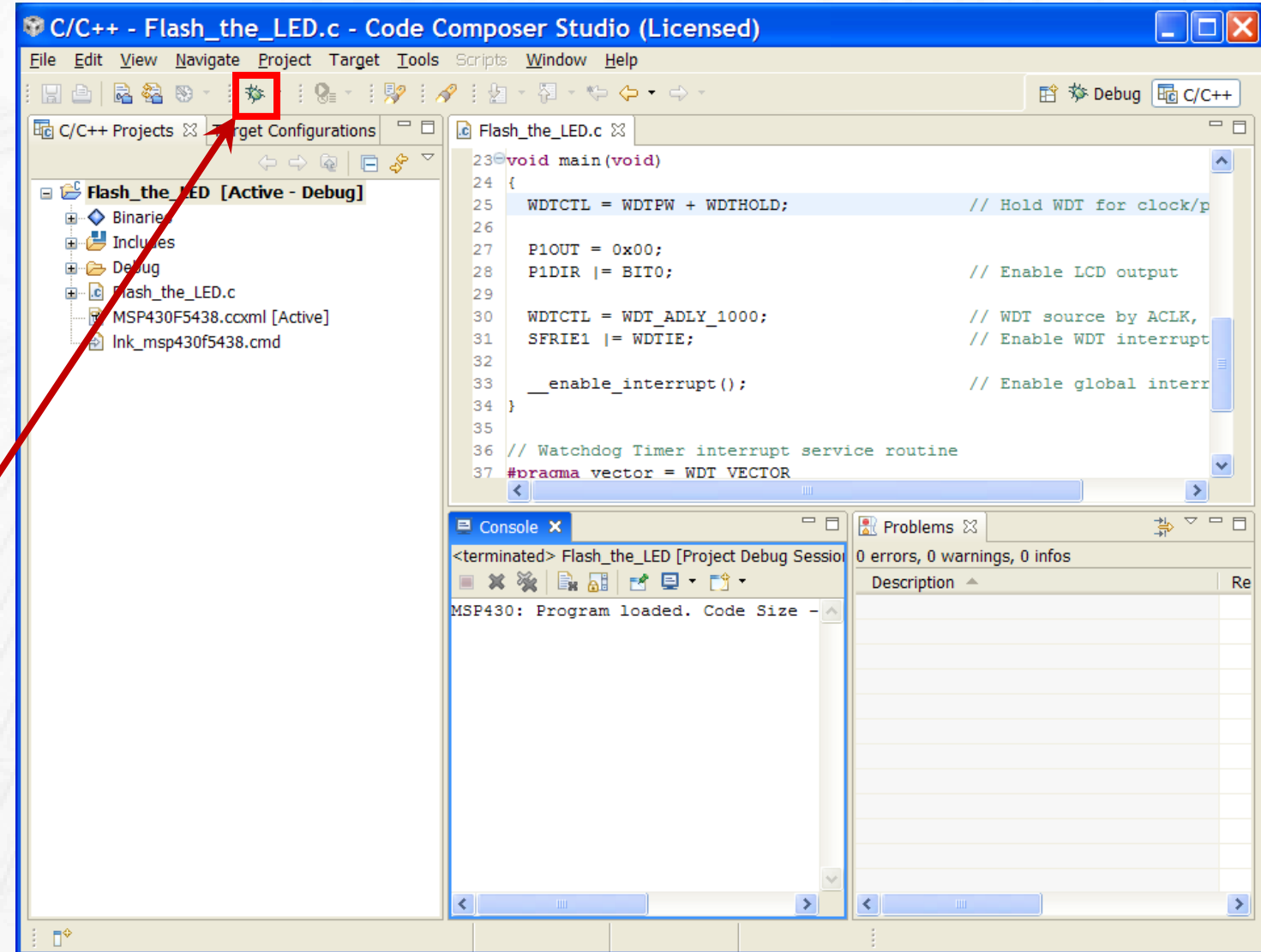
Real-time, in-system
MSP430 information

- Register access
- Flash, RAM, Info
segment access
- Disassembly view

Code Composer Studio Projects

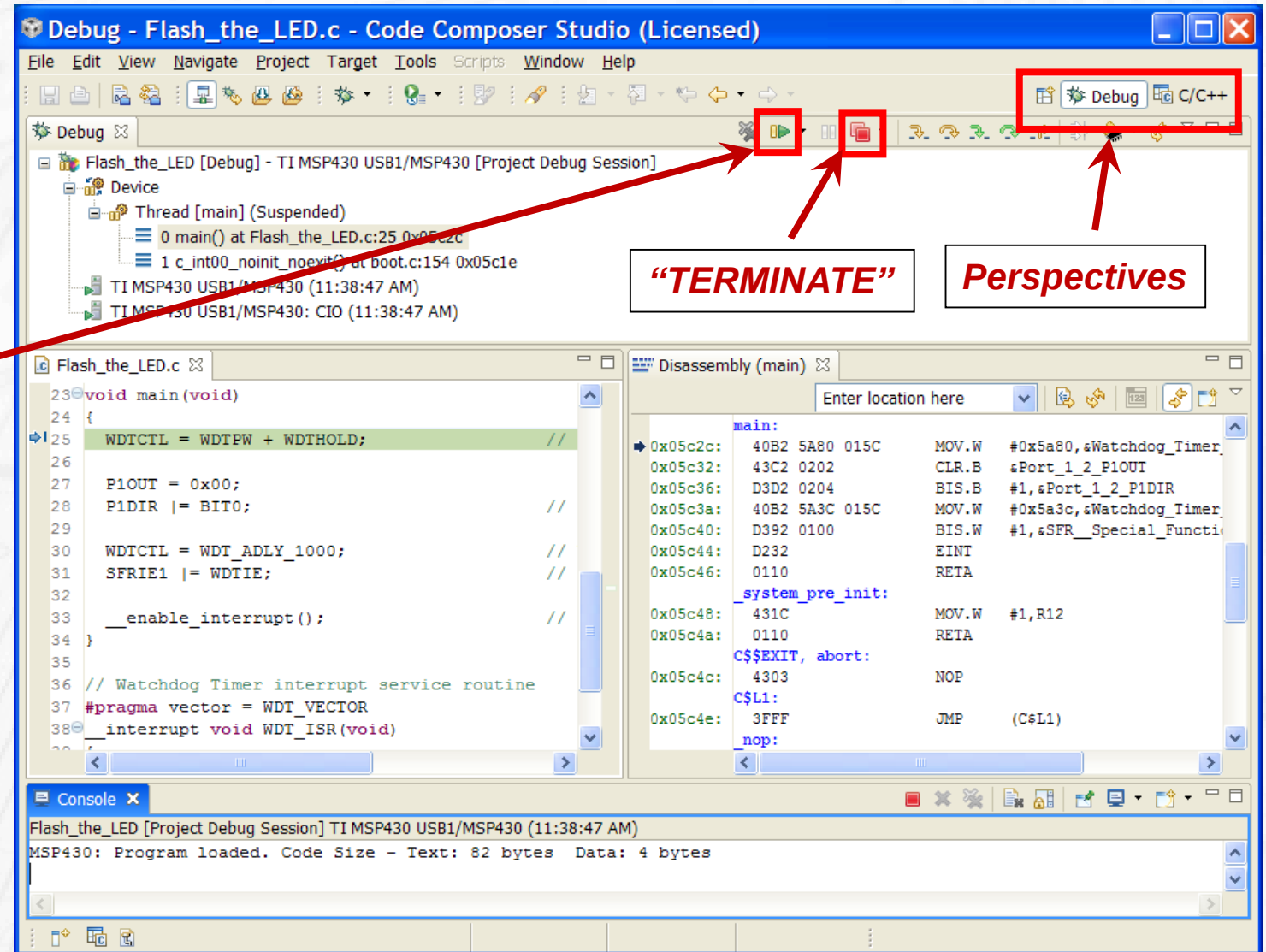
Step 4: Build and Debug a CSS Project

*Click the “BUG” to build the
code & launch the debugger*



Code Composer Studio Projects

Step 5: Run, Terminate a CSS Project



Code Composer Studio

Projects

HELLO WORLD!

```
WDTCTL = WDTPW + WDTHOLD;           // Stop watchdog timer
volatile unsigned int i=0;           // volatile to prevent optimization
P1DIR |= 0x01;                       // Set P1.0 to output direction

for (;;)
{
    P1OUT ^= 0x01;                    // Toggle P1.0 using exclusive-OR - Red Blinks
    i = 10000;                        // SW Delay
    do i--;
    while (i != 0);
}
```