

2012/10/03

1. Δίνονται οι αριθμοί \$3B και \$CC. Πώς ερμηνεύει το άθροισμά τους ο AVR όταν οι αριθμοί θεωρούνται μόνο θετικοί ή προσημανσμένοι ( 2's complement);

Πώς γίνεται αντιληπτή η υπερχείλιση;

\$3B → 0b 0011 1011 → 59

\$CC → 0b 1100 1100 → 0011 0011 → 0011 0100 → 52<sub>10</sub>

+ 0b1 0000 0111

Όπως αναφέρεται ο 2<sup>ος</sup> αριθμός είναι στη μορφή συμπληρώματος ως προς 2. Έτσι είναι αρνητικός και μάλιστα στο δεκαδικό θα έχω την πράξη 59-52 = 7. Πράγματι επαληθεύεται.

Η υπερχείλιση ελέγχεται μέσα από τις σημαίες C, V, (S)

2. Ποια είναι η χρησιμότητα του WDT;

Ο WDT κάνει μια αντίστροφη μέτρηση και όποιο πρόγραμμα και αν τρέχει πρέπει να τον επαναφέρει πριν αυτός μηδενιστεί. Αν δε γίνει επαναφορά και μηδενίσει ο WDT τότε θεωρείται πως το πρόγραμμα έχει κολλήσει και γίνεται επανεκκίνηση.

3. Ποια είναι η χρήση στον AVR της μνήμης FLASH και της EEPROM; Σε τι διαφέρουν;

Στην EEPROM έχω δεδομένα που δε θέλω να χαθούν με την απώλεια ρεύματος.

4. Περιγράψτε στάδια εκτέλεσης (κύκλους μηχανής) της εντολής icall του AVR.

5. Ποιους τύπους δεδομένων χειρίζεται το υλικό του AVR (χωρίς χρήση ειδικού λογισμικού);

Ακέραιους προσημασμένους και μη.

*1. Ποια θεωρούνται τα κύρια χαρακτηριστικά του AVR τα οποία δίνουν τη δυνατότητα βελτίωσης του χρόνου εκτέλεσης ενός προγράμματος;*

Ο AVR χρησιμοποιεί για τη βελτίωση του χρόνου εκτέλεσης των προγραμμάτων, pipelining ή αλλιώς διοχέτευση. Δηλαδή επιτρέπονται περισσότερα από ένα token και άρα περισσότερα ενεργά βήματα κάθε φορά.

Έστω  $n$  επίπεδα με χρόνο  $t$  το καθένα. Κάθε εντολή θέλει  $n \cdot t$  χρόνο για να εκτελεστεί. Ωστόσο ο μέσος αριθμός ολοκλήρωσης των εντολών είναι  $1/t$  εντολές ανα μονάδα χρόνου. Έτσι έχουμε πετύχει  $n$  φορές μεγαλύτερη ταχύτητα στη μηχανή μας (με την προϋπόθεση ότι υπάρχουν πάντοτε  $n$  tokens στο διάγραμμα καταστάσεων).

Άλλες τεχνικές είναι η παραλληλοποίηση και η βελτίωση του CLK.

*2. Αναφέρατε τρόπους υλοποίησης ψηφιακών κυκλωμάτων.*

- i. Χρήση ενός ψηφιακού κυκλώματος ειδικού σκοπού.
- ii. Χρήση ψηφιακών με PLD (programmable logic Device).
- iii. Υλοποίηση βασισμένη σε ASIC

*3. Πως αναγνωρίζει τους προσημασμένους αριθμούς ο AVR;*

Δεν τους αναγνωρίζει, έγκειται στον προγραμματιστή να καθορίσει πως ερμηνεύει τα περιεχόμενα των καταχωρητών (αν έχεις πχ 1 κλασματικό και 7 ακέραιο κλπ).

*4. Περιγράψτε στάδια εκτέλεσης της εντολής (1278): rcall K, του AVR.*

$Stack \leftarrow PC + 1$

$PC \leftarrow 1278 - 37 + 1$

$SP \leftarrow SP - 2$

*5. Ποια είναι η χρησιμότητα της δεικτοδότησης σε ένα πρόγραμμα;*

Μπορούμε να χρησιμοποιήσουμε τους δείκτες X, Y, Z για να προσπελάσουμε τις θέσεις μνήμης της SRAM ενώ αντίστοιχα μπορούμε να κάνουμε την ίδια δουλειά στην Program Memory μόνο με τον Z. Γενικά προσφέρουν ευελιξία.

*1. Σε ένα μικροεπεξεργαστή με μονάδα διοχέτευσης (pipeline) σε ποιες περιπτώσεις εμφανίζεται αναβολή στην εκτέλεση της εντολής (hazard);*

Κίνδυνοι δομής	Υλικό ανίκανο να υποστηρίξει συνδιασμό εντολών- Σύγκρουση για χρήση κοινών πόρων.
Κίνδυνοι δεδομένων	Όταν μια εντολή εξαρτάται από μια άλλη, η οποία όμως δεν έχει ακόμη τερματιστεί.
Κίνδυνοι ελέγχου	Είσοδο στο pipeline εντολών με ή χωρίς συνθήκη.

*2. Στη μνήμη FLASH του AVR τι είδους πληροφορία αποθηκεύεται;*

Στην μνήμη FLASH που αποτελεί την Program Memory αποθηκεύονται κατά βάση οι εντολές του προγράμματος, αλλά και δεδομένα που τυχόν θέλουμε να αποθηκεύσουμε.

*3. Να αναλυθεί σε κύκλους μηχανής η εντολή ADC R3, R6.*

Όταν έχω προσπέλαση μνήμης στον 1<sup>ο</sup> κύκλο υπολογίζω την διεύθυνση

A φορτώνονται οι τιμές των τελεστών της (π.χ. GPR) - Register Operands Fetch

B η ALU εκτελεί την εντολή - ALU operation execute

C το αποτέλεσμα γράφεται στον προορισμό του - Result write back

Φαντάζομαι θέλει να πεις ότι η εντολή εκτελείται κατά τη διάρκεια 2 περιόδων του clock (όχι καθ' όλη, χρονικά παίρνει 1, λόγω αρχιτεκτονικής Harvard και pipeline δύο σταδίων). Στο 1ο κύκλο πηγαίνει την εντολή στον καταχωρητή εντολών και στον δεύτερο:  $R3 \leftarrow R6 + R3 + C$

*4. Ποιες είναι οι βασικές κατηγορίες εντολών σε ένα τυπικό μικροεπεξεργαστή;*

- i. Αριθμητικών και λογικών πράξεων
- ii. Ελέγχου ροής προγράμματος
- iii. Εντολές μεταφοράς δεδομένων
- iv. Έλεγχος και διαχείριση σε επίπεδο bit

*5. Δίνονται οι αριθμοί \$37 και \$A9. Πως ερμηνεύει ο AVR το άθροισμά τους όταν οι αριθμοί θεωρούνται μόνο θετικοί ή με πρόσημο σε μορφή συμπληρώματος του 2; Σε ποια περίπτωση το αποτέλεσμα είναι σωστό; Με ποιο μηχανισμό αναγνωρίζεται η ορθότητα του αποτελέσματος;*

\$37 → 0b 0011 0111

\$A9 → 0b 1010 1001

1. Ποια είναι τα πλεονεκτήματα της αρχιτεκτονικής Harvard;

Η αρχιτεκτονική Harvard είναι μια αρχιτεκτονική υπολογιστών με ξεχωριστούς διαύλους επικοινωνίας και αποθήκευσης σημάτων για τις εντολές και τα δεδομένα. Στην αρχιτεκτονική Von Neumann η CPU μπορεί είτε να διαβάσει μια εντολή ή να διαβάσει αποθηκευμένα δεδομένα από/προς τη μνήμη. ΔΕΝ μπορούν να γίνουν και τα δυο ταυτόχρονα αφού χρησιμοποιούν το ίδιο bus system.

Η αρχιτεκτονική Von Neumann έχει ξεχωριστό χώρο διεύθυνσεων για κώδικα και δεδομένα, προσφέροντας έτσι ευελιξία.

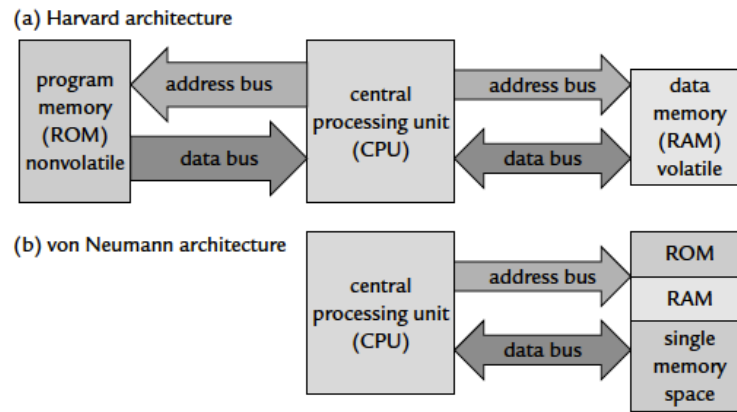


Figure 1.3: Harvard and von Neumann architectures for memory.

(von Neumann = Princeton)

2. Αναλύστε σε κύκλους μηχανής την εντολή SBC r3,r4.

3. Ποια η διαφορά ανάμεσα στις εντολές CALL και RCALL;

CALL 16 bits δίνουν πρόσβαση σε 64 Kbytes

RCALL δίνει πρόσβαση σε  $PC \pm 2 K$

CALL  $pc'$  όπου  $pc'$  η διεύθυνση που θέλω να πάω

RCALL  $pc' \leftarrow pc + k + 1$

4. Αναφέρατε τρόπους ελάττωσης του χρόνου εκτέλεσης μιας εντολής.

1. Απλές εντολές
2. Απλοί τρόποι προσπέλασης
3. Σταθερές μικρού μεγέθους
4. Μικρός αριθμός διαφορετικών τύπων δεδομένων
5. Συχνή εμφάνιση κλήσης διαδικασιών
6. Διοχέτευση (pipeline)
7. Παραλληλοποίηση
8. Πολλοί καταχωρητές

5. Ποια είναι τα όρια του πεδίου τιμών ενός προσημασμένου αριθμού ο οποίος αποτελείται από 4 bit για το ακέραιο τμήμα και από 4 bit για το κλασματικό.

1. Αναφέρατε τους τύπους υπολογιστικών μοντέλων κατά Flynn.

SISD : Σειριακοί υπολογιστές (μία εντολή → ένα δεδομένο)

Single Instruction Single Data

MISD : πολλοί επεξεργαστές (πολλές εντολές → ένας τελεστής) (ανέφικτο)

Multiple Instruction Single Data

SIMD : πολλοί επεξεργαστές (μία εντολή → πολλά δεδομένα)

Single Instruction Multiple Data

MIMD : πολλοί επεξεργαστές (πολλές εντολές → πολλά δεδομένα)

Multiple Instruction Multiple Data

2. Ποια είναι τα επιμέρους τμήματα μνήμης του AVR και πως χρησιμοποιούνται;

3. Ποιοι είναι οι κύκλοι μηχανής μιας εντολής του AVR και πως εκτελούνται;

4. Σε ποιες περιπτώσεις χρησιμοποιείται η στοίβα;

5. Με ποιο τρόπο αναγνωρίζει κλασματικούς αριθμούς ο AVR;

Έλα ντε.. εγώ νόμιζα οτι δεν τους αναγνωρίζει καν και είναι στη φιλοσοφία του προγραμματιστή πως θα χωρίσει σε ακέραια και κλασματικά μέρη και πόσα bit θα δώσει για το καθένα...

2009/02/24

*1. Αναφέρατε περιπτώσεις χρησιμοποίησης της στοίβας.*

Γενικά χρησιμοποιείται από την CPU για αποθήκευση των λεγόμενων *διευθύνσεων επιστροφής* με σκοπό να συνεχιστεί η εκτέλεση του προγράμματος που είχε διακοπεί από κλήση υπορουτίνας ή κάποιας ρουτίνας διακοπής.

(RCALL, CALL)

*2. Αναλύστε σε κύκλους μηχανής την εντολή : 1278 rcall -37*

$Stack \leftarrow PC + 1$

$PC \leftarrow 1278 - 37 + 1$

$SP \leftarrow SP - 2$

*3. Ποια είναι τα κύρια χαρακτηριστικά των μηχανών τύπου RISC;*

Η αρχιτεκτονική RISC (Reduced Instruction Set Computer) είναι μια αρχιτεκτονική σχεδίασης CPU που βασίζεται στο γεγονός πως απλοποιημένες και μικρές εντολές οδηγούν σε μεγαλύτερη απόδοση της CPU αφού κάθε εντολή απαιτεί λιγότερο χρόνο για να εκτελεστεί. Η αντίπαλη αρχιτεκτονική είναι η CISC

- Ομοιόμορφη «κατασκευή» των εντολών opcode στην ίδια θέση για κάθε εντολή
- Πανομοιότυποι καταχωρητές γενικής χρήσης που μπορούν να χρησιμοποιηθούν για διαφορετικές εργασίες
- Απλοποιημένος υπολογισμός διευθύνσεων, - περίπλοκες διευθύνσεις υπολογίζονται με αλληλουχία αριθμητικών και λογικών πράξεων
- Περιορισμένοι τύποι δεδομένων στο υλικό

*4. Τι αποθηκεύεται στη μνήμη προγράμματος του AVR;*

- i. Εντολές που πρέπει να εκτελεστούν
- ii. Δεδομένα αν το επιθυμώ

*5. Σε ποιες περιπτώσεις χρησιμοποιείται η μέθοδος προσπέλασης: Έμμεση με μετατόπιση;*

Όταν δε θέλω να επηρεάσω την τιμή του δείκτη μου (X, Y, Z). Η εντολή στην οποία αναφερόμαστε είναι η LDD.

2008/02/15

1. Περιγράψτε τη φυσική και λογική διάταξη της SRAM του AVR.
2. Σε ποιες περιπτώσεις χρησιμοποιείται ο WDT;
3. Ποιοι τύποι εντολών πρόσθεσης υλοποιούνται στον AVR και σε ποιες περιπτώσεις χρησιμοποιούνται.
4. Πως υλοποιείται και πότε χρησιμοποιείται ο τρόπος διευθυνσιοδότησης : Άμεση Προσπέλαση μνήμης προγράμματος (Direct Program Memory);
5. Από ποια επιμέρους τμήματα κώδικα συντάσσεται ένα πρόγραμμα assembly

2007/08/28

1. Ποιες σημαίες του SR (I, T, H, S, V, N, Z, C) θα επηρεαστούν με την εκτέλεση της εντολής ADC; Επηρεάζονται οι Z, C, N, V, S, H

2. Ποιο είναι το συμπλήρωμα του δύο του αριθμού 00000001;

$0b\ 0000\ 0001 \xrightarrow{1's\ complement} 0b\ 1111\ 1110 \xrightarrow{2's\ complement} 0b\ 1111\ 1111$

3. Με ποιες εντολές θα γίνει μεταφορά των δύο bit μικρότερης σημασίας του καταχωρητή R15 στο καταχωρητή R16, χωρίς όμως να αλλάξουν τιμή τα υπόλοιπα bit του R16;

4. Τι είναι το μοντέλο προγραμματισμού ενός μικροεπεξεργαστή; Ποια είναι τα πιο συνηθισμένα; Ποιο μοντέλο χρησιμοποιεί ο AVR;

Είναι ο τρόπος με τον οποίο αποθηκεύονται και διαχειρίζονται τα δεδομένα στη CPU. Υπάρχουν

- i. Μοντέλο στοίβας
- ii. Μοντέλο συσσωρευτή
- iii. Μοντέλο καταχωρητή-μνήμης
- iv. Μοντέλο πολλών καταχωρητών

5. Πότε χρησιμοποιείται η δεικτοδοτημένη με μετά-αύξηση προσπέλαση;

Όταν έχω για παράδειγμα έναν πίνακα και θέλω να προσπελάσω όλα τα κελιά του στη σειρά

\*LD R<sub>d</sub>, Y+

Όταν η προσπέλαση γίνεται μέσα σε label προγράμματος που πραγματοποιεί ένα loop και μετά τη συνθήκη ελέγχου και επιστροφής χρειάζεται να ελεγχθεί η επόμενη θέση μνήμης



2006/09/21

1. Πως χρησιμοποιείται η σχετική προσπέλαση μνήμης προγράμματος; Ποια είναι η χρησιμότητά της;
2. Ποια είναι η διαφορά LSR και ASR; Πότε χρησιμοποιούνται;
3. Τι λειτουργία εκτελούν οι ψευδοεντολές;

Οι ψευδοεντολές είναι εντολές προς τον ίδιο τον συμβολομεταφραστή όπως .equ .cseg .db κλπ

4. Με τη χρήση χρονικού διαγράμματος περιγράψτε την διαδικασία ανάκλησης –αποκωδικοποίησης και εκτέλεσης εντολών για τον AVR.
5. Αναφέρατε τις εναλλακτικές λύσεις για την υλοποίηση ενός ψηφιακού συστήματος.

1. Περιγράψτε τη φυσική και λογική διάταξη της SRAM του AVR

Απ:σελ.77-Παράγραφος 3.9,σχήμα 3.32,σχήμα 3.33,παράγραφος 3.9.1,σχήμα 3.34?

2. Σε ποιές περιπτώσεις χρησιμοποιείται ο WDT?

Απ:σελ.91 παράγραφος 3.13?

Hardware1 [slide29](#)

Χρησιμοποιείται για να ελέγξει τα κρσαρισματα του software. Επίσης πολλές φορές λέγεται COP ( Computer Operates Properly) χρησιμοποιείται για να ελέγξει την ομαλή εκτέλεση ενός προγράμματος. Η βασική ιδέα είναι ότι όταν είναι ενεργοποιημένος μετράει αντίστροφα. Είναι ευθύνη του προγράμματος να μην τον αφήσει να μηδενιστεί (kick the dog). Αν μηδενιστεί εκινούνται οι διαδικασίες της επανεκκίνησης.

3. Ποιοί τύποι εντολών πρόσθεσης υλοποιούνται στον AVR και σε ποιές περιπτώσεις χρησιμοποιούνται?

Απ:Γράφουμε όλες τις ADD?

**add** για μια απλή πρόσθεση μεταξύ δυο register

**adc** συνήθως για να προσθέσουμε τα high bytes 2 16μπιτων αριθμών

**addiw** Adds an immediate value (0 - 63) to a register pair and places the result in the register pair. This instruction operates on the upper four register pairs, and is well suited for operations on the pointer registers.

4. Πώς υλοποιείται και πότε χρησιμοποιείται ο τρόπος διευθυνσιοδότησης:Άμεση προσπέλαση μνήμης προγράμματος(Direct Program Memory)?

Απ?Κάτι μου κάνει για σελ.95-96 αλλά.....

LDS R1,\$100 ; πάρε το περιεχόμενο της διεύθυνσης \$100 της SRAM και βάλτο στην R1

STS \$100,R1 ; αποθήκευσε το περιεχόμενο της R1 στη θέση \$100 της SRAM

δηλαδή χρησιμοποιείται για προσπέλαση στην SRAM

δες και [slide 24](#)

5. Από ποια επιμέρους τμήματα κώδικα συντάσσεται ένα πρόγραμμα assembly?

Software slides [iset.pdf](#) slide 54

- |       |                          |                                   |
|-------|--------------------------|-----------------------------------|
| i.    | Header Comments          | title author date and description |
| ii.   | Definition Includes      | eg "m128def.inc"                  |
| iii.  | Register Renaming        | .def temp = r0;                   |
| iv.   | Constant Declaration     | .equ addr = \$2000                |
| v.    | Interrupt Vectors        |                                   |
| vi.   | Initialization Code      |                                   |
| vii.  | Main Code                |                                   |
| viii. | Subroutines              |                                   |
| ix.   | ISRs                     | interrupt subroutines             |
| x.    | Data                     | eg .db "hello"                    |
| xi.   | Additional Code Includes |                                   |

6. Αναφέρατε περιπτώσεις χρησιμοποίησης της στοίβας

Απ: σελ.44-->[3]Ένα τμήμα εσωτ. μνήμης.....επιστροφής?

### 7. Ποιά τα κύρια χαρακτηριστικά των μηχανών τύπου RISC?

Απ:σελ.39-40 Απαρίθμηση κύριων χαρακτηριστικών?

### 8. Τι αποθηκεύεται στη μνήμη προγράμματος του AVR?

IRQ (διανυσματα διακοπων)

Οι εντολές

Πίνακες (δλδ τα .db )

Boot Program

### 9. Σε ποιές περιπτώσεις χρησιμοποιείται η μέθοδος προσπέλασης: Έμμεση με μετατόπιση?

LDD  $Rd, Z+qRd \leftarrow (Z+q)$

Z: Unchanged, q: Displacement ( $0 \leq d \leq 31, 0 \leq q \leq 63$ )

(μπορει να ναι και Y αλλα OXI το X)

Το χρησιμοποιούμε όταν έχουμε να κάνουμε με pointers. Το Z (ή Y) μετατοπίζεται κάποιες θέσεις, παίρνουμε το στοιχείο που θέλουμε, αλλά η τιμή του παραμένει η ίδια με την αρχική.

LDI YH, 1 ; Δείχνει το Y στην \$100

LDI YL, 0

Somewhere later in the program I'd like to write to cell 2 above SRAM\_START:

STD Y+2, MyPreferredRegister

The corresponding instruction for reading from SRAM with an offset

LDD MyPreferredRegister, Y+2

### 10. Αρχιτεκτονική μικροελεγκτών

a) Με βάση τον συνολικό αριθμό εντολών

- RISC
- CISC

b) Με βάση τον τρόπο που πραγματοποιείται η πρόσβαση σε μνήμη προγράμματος και μνήμη δεδομένων

- Princeton/ Von Neumann
- Harvard

c) Με βάση τον τρόπο αποθηκευσης και διαχείρισης των δεδομένων από τη CPU

- Μοντέλο στοίβας (stack)
- Μοντέλο συσσωρευτή (accumulator)
- Μοντέλο καταχωρητής-μνήμη (register-memory)
- Μοντέλο πολλών καταχωρητών (register-register / load-store)