ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ Τ.Ε.Ι. ΑΘΗΝΑΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

Μάθημα: Εισαγωγή Στα Συστήματα Μικροελεγκτών **Εισηγητής: Δρ. Κουλούρας Γρηγόριος**	**Ακαδημαϊκό Έτος 2012-13** **Εξάμηνο Χειμερινό** **Α΄ Εξεταστική Περίοδος** Σημειώσεις : κλειστές Διάρκεια εξέτασης: 2 ώρες Ημ. εξέτασης: 18/02/2013

Θέμα 1^{ο} (3.0 μονάδες**):**

α) Τι γνωρίζετε για την οργάνωση της μνήμης προγράμματος «τύπου Flash» του μικροελεγκτή ATmega16;

β) Σε πόσα και ποιά τμήματα χωρίζεται η μνήμη δεδομένων «τύπου SRAM» του μικροελεγκτή ATmega16; Με ποιές εντολές γράφουμε και διαβάζουμε στο κάθε τμήμα μνήμης;

Θέμα 2^{ο} (2.0 μονάδες**):**

Γράψτε πρόγραμμα σε συμβολική γλώσσα «Assembly» για τον μικροελεγκτή ATmega16, που να αθροίζει μια φορά, τα περιεχόμενα των θέσεων μνήμης $013B και $013B και να καταχωρεί το άθροισμα στην διεύθυνση $013C. Στη συνέχεια να μηδενίζει τα περιεχόμενα των διευθύνσεων $013B και $013C.

Θέμα 3^{ο} (2.5 μονάδες**):**

Γράψτε πρόγραμμα σε συμβολική γλώσσα «Assembly» για τον μικροελεγκτή ATmega16, που να απαριθμεί συνεχώς, τα περιττά Bytes σε ένα μπλοκ μνήμης SRAM μεγέθους 12 Bytes που περιέχει μη προσημασμένους αριθμούς 8-bit. Το μπλοκ μνήμης αρχίζει από την διεύθυνση $0060. Στο τέλος του ελέγχου το πλήθος των περιττών Bytes που ανιχνεύτηκαν στο συγκεκριμένο διάστημα μνήμης θα απεικονίζονται στα 8 LEDs της πόρτας Α, που είναι συνδεδεμένα με θετική λογική.

Θέμα 4^{ο} (2.5 μονάδες**):**

Παρακάτω φαίνεται ο χάρτης της μνήμης προγράμματος «Memory Map» του μικροελεγκτή ATmega16. Να μετατραπούν οι παρακάτω εντολές συμβολικής γλώσσας «Assembly Commands» σε εκτελέσιμο κώδικα «Operational Code» και τα αποτελέσματα να γραφτούν στην δυαδική και δεκαεξαδική μορφή στις αντίστοιχες στήλες «Data» του παρακάτω πίνακα. Οι πράξεις να γίνουν χωρίς την χρήση ηλεκτρονικού υπολογιστή. Παρακάτω δίνονται τα «Op-code» των εντολών που χρησιμοποιούνται στο πρόγραμμα.

Καλή επιτυχία

Ο Εισηγητής

Δρ. Κουλούρας Γρηγόριος
Καθηγητής Εφαρμογών

Word Addressing	Byte Addressing	Data (Hex)	Data (Bin)	Assembly Commands
$0000	$0000			JMP RESET
	$0001			
$0001	$0002			
	$0003			
$0002	$0004			RESET: LDI R16, $40
	$0005			
$0003	$0006			LDI R17, $3F
	$0007			
$0004	$0008			ADD R17, R16
	$0009			
$0005	$000A			MAIN: NOP
	$000B			
$0006	$000C			NOP
	$000D			
$0007	$000E			JMP MAIN
	$000F			
$0008	$0010			
	$000D			
…	…	…	…	…
	…	…	…	…
$1FFF	$3FFE			NOP
	$3FFF			

Σύνταξη:	**Τελεστές:**	**Μετρητής Προγράμματος:**
JMP k	0≤k≤4M	PC ← k
Λειτουργία:	**Μήκος Εντολής:**	**Χρόνος Εκτέλεσης Εντολής:**
PC ← k	2 Word (4 Bytes)	3 Cycles

Status Register (SREG):							
I	**T**	**H**	**S**	**V**	**N**	**Z**	**C**
-	-	-	-	-	-	-	-

32-bit Opcode:			
1001	**010k**	**kkkk**	**110k**
kkkk	**kkkk**	**kkkk**	**kkkk**

Σύνταξη:	**Τελεστές:**	**Μετρητής Προγράμματος:**
LDI Rd, k	16≤d≤31, 0≤k≤255	PC ← PC + 1
Λειτουργία:	**Μήκος Εντολής:**	**Χρόνος Εκτέλεσης Εντολής:**
Rd ← k	1 Word (2 Bytes)	1 Cycles

Status Register (SREG):							
I	**T**	**H**	**S**	**V**	**N**	**Z**	**C**
-	-	-	-	-	-	-	-

16-bit Opcode:			
1110	**kkkk**	**dddd**	**kkkk**

Σύνταξη:	**Τελεστές:**	**Μετρητής Προγράμματος:**
ADD Rd, Rr	0≤d≤31, 0≤r≤31	PC ← PC + 1
Λειτουργία:	**Μήκος Εντολής:**	**Χρόνος Εκτέλεσης Εντολής:**
Rd ← Rd + Rr	1 Word (2 Bytes)	1 Cycles

Status Register (SREG):							
I	**T**	**H**	**S**	**V**	**N**	**Z**	**C**
-	-	<>	<>	<>	<>	<>	<>

16-bit Opcode:			
0000	**11rd**	**dddd**	**rrrr**

Σύνταξη:	**Τελεστές:**	**Μετρητής Προγράμματος:**
NOP	-	PC ← PC + 1
Λειτουργία:	**Μήκος Εντολής:**	**Χρόνος Εκτέλεσης Εντολής:**
-	1 Word (2 Bytes)	1 Cycles

Status Register (SREG):							
I	**T**	**H**	**S**	**V**	**N**	**Z**	**C**
-	-	-	-	-	-	-	-

16-bit Opcode:			
0000	**0000**	**0000**	**0000**

```
;============================== PROGRAM TEMPLATE ================================;
.include "m16def.inc"
.CSEG
.ORG 0
       jmp RESET              ; Reset Handler
.CSEG
.ORG 43
RESET: cli
       ldi r16, high(RAMEND)
       out SPH,r16
       ldi r16, low(RAMEND)
       out SPL,r16
INIT:  ……
       ……
MAIN:  ……
       ……
       rjmp MAIN
;================================================================================;
```