



ΤΕΙ ΑΘΗΝΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

Όνομα :
Επώνυμο :
Α.Μ. :
Εξάμηνο :
Ημερομηνία : 23/06/2011
Εισηγητής : Κουλούρας Γρηγόριος

Α' ΕΞΕΤΑΣΤΙΚΗ ΠΕΡΙΟΔΟΣ ΕΑΡΙΝΟΥ ΕΞΑΜΗΝΟΥ 2010-2011

ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΕΣ Ι

Θέμα 1^ο

5.0 μονάδες

A) Τι γνωρίζεται για τις αρχιτεκτονικές **von-Neuman** και **Harvard**; Τι συμπεράσματα βγάζουμε συγκρίνοντας δύο επεξεργαστές που λειτουργούν με τις δύο παραπάνω αρχιτεκτονικές;

(1.6 μονάδες)

B) Περιγράψτε την λειτουργία του **σωρού (stack)** στους μικροελεγκτές AVR. Γράψτε μερικές εντολές στην συμβολική γλώσσα «Assembly» για τους μικροελεγκτές AVR που επιρεάζουν τον Δείκτη Στοίβας (Stack Pointer).

(1.2 μονάδες)

Γ) Ο μικροελεγκτής AVR AT90S8535 ενσωματώνει μια μονάδα Ασύγχρονης Σειριακής Επικοινωνίας ή **UART**. Περιγράψτε εν συντομία την λειτουργία του κύκλωματος εκπομπής και λήψης.

(1.0 μονάδες)

Δ) Έστω ότι η ρύθμιση της σειριακής πόρτας είναι 115200bps 8N1 (8 Data bits, No parity, 1 Stop bit). Σχεδιάστε ένα UART frame (πλαίσιο) όπως μπορεί να έχει καταγραφεί από έναν παλμογράφο κατά την διάρκεια σειριακής επικοινωνίας. Το byte της πληροφορίας που αποστέλλεται είναι το \$F0. Να υπολογίσετε την διάρκεια του ενός bit καθώς και ολόκληρου του frame.

(1.2 μονάδες)

Θέμα 2^ο

5.0 μονάδες

A) Να σχεδιαστεί στοιχειώδες κύκλωμα υποστήριξης μικροελεγκτή AVR AT90S8535, που χρονίζεται στα 16MHz από εξωτερικό κρύσταλλο. Επίσης στην πόρτα A να συνδεθούν οκτώ (8) LED σε συνδεσμολογία θετικής λογικής.

(0.8 μονάδες)

B) Δίνεται το παρακάτω ολοκληρωμένο πρόγραμμα συμβολικής γλώσσας, για τον μικροελεγκτή AVR AT90S8535 του παραπάνω κυκλώματος. Περιγράψτε τι κάνει η κάθε εντολή στα τρία μπλοκ κώδικα που φαίνονται παρακάτω: «Reset Handler», «Initialize» και «Main Program».

(1.8 μονάδες)

Γ) Στη συνέχεια γράψτε νέο πρόγραμμα σε συμβολική γλώσσα «Assembly» για τον παραπάνω μικροελεγκτή, που να ταξινομεί κατά αύξουσα σειρά, ένα μπλοκ μνήμης SRAM μεγέθους 12 Bytes που περιέχει μη προσημασμένους αριθμούς 8-bit. Το μπλοκ μνήμης αρχίζει από την διεύθυνση \$007A. Στο τέλος του ελέγχου ο ελάχιστος αριθμός θα απεικονίζεται στην δυαδική του μορφή στα 8 LED της πόρτας A.

(2.4 μονάδες)

;Note: Start of Program

```
.include "8535def.inc"
.CSEG
.ORG 0
```

;Note: Interrupt Vectors

```
    rjmp RESET           ; Reset Handler
    rjmp EXT_INT0        ; IRQ0 Handler
    rjmp EXT_INT1        ; IRQ1 Hanzdler
    rjmp TIM2_COMP       ; Timer2 Compare Handler
    rjmp TIM2_OVF        ; Timer2 Overflow Handler
    rjmp TIM1_CAPT       ; Timer1 Capture Handler
    rjmp TIM1_CMPA       ; Timer1 CompareA Handler
    rjmp TIM1_CMPB       ; Timer1 CompareB Handler
    rjmp TIM1_OVF        ; Timer1 Overflow Handler
    rjmp TIM0_OVF        ; Timer0 Overflow Handler
    rjmp SPI_STC         ; SPI Transfer Complete Handler
    rjmp UART_RXC        ; UART RX Complete Handler
    rjmp UART_DRE        ; UDR Empty Handler
    rjmp UART_TXC        ; UART TX Complete Handler
    rjmp ADC_CONV        ; ADC Conversion Complete Interrupt Handler
    rjmp EE_RDY          ; EEPROM Ready Handler
    rjmp ANA_COMP        ; Analog Comparator Handler
```

;Note: Reset Handler

```
RESET:    cli                ;
          ldi r16, high(RAMEND) ;
          out SPH,r16         ;
          ldi r16, low(RAMEND) ;
          out SPL,r16        ;
```

;Note: Initialize

```
INIT:     clr r16             ;
          ser r17             ;
          out DDRA,r17        ;
          out PORTA,r16       ;
```

;Note: Main Program

```
CHECK:    ldi r26,low($0060)  ;
          ldi r27,high($0060) ;
          ldi r20,10          ;
          clr r21              ;
CHECK_1:   ld r19,X+           ;
          cpi r19,$AA         ;
          brne CHECK_2        ;
          inc r21              ;
CHECK_2:   dec r20             ;
          brne CHECK_1        ;
          out PORTA,r21       ;
MAIN:     rjmp MAIN           ;
```

;Note: Interrupt Handlers

```
EXT_INT0: reti                ; IRQ0 Handler
EXT_INT1: reti                ; IRQ1 Handler
TIM2_COMP: reti               ; Timer2 Compare Handler
TIM2_OVF:  reti               ; Timer2 Overflow Handler
TIM1_CAPT: reti               ; Timer1 Capture Handler
TIM1_CMPA: reti               ; Timer1 CompareA Handler
TIM1_CMPB: reti               ; Timer1 CompareB Handler
TIM1_OVF:  reti               ; Timer1 Overflow Handler
TIM0_OVF:  reti               ; Timer0 Overflow Handler
SPI_STC:   reti               ; SPI Transfer Complete Handler
UART_RXC:  reti               ; UART RX Complete Handler
UART_DRE:  reti               ; UDR Empty Handler
UART_TXC:  reti               ; UART TX Complete Handler
ADC_CONV:  reti               ; ADC Conversion Complete Interrupt Handler
EE_RDY:    reti               ; EEPROM Ready Handler
ANA_COMP:  reti               ; Analog Comparator Handler
```

;Note: End of Program
