



ΤΕΙ ΑΘΗΝΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

Όνομα :
Επώνυμο :
Α.Μ. :
Εξάμηνο :
Ημερομηνία : 03/02/2011
Εισηγητής : Κουλούρας Γρηγόριος

Α' ΕΞΕΤΑΣΤΙΚΗ ΠΕΡΙΟΔΟΣ ΧΕΙΜΕΡΙΝΟΥ ΕΞΑΜΗΝΟΥ 2010-2011

ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΕΣ Ι

Θέμα 1^ο

2.5 μονάδες

A) Τι ονομάζουμε «Τεχνική Σωλήνωσης» ή «Pipelining» στους επεξεργαστές; Ποιά είναι τα βήματα εκτέλεσης μιας εντολής σε ένα επεξεργαστή που χρησιμοποιεί τεχνική σωλήνωσης πέντε (5) βημάτων; (1.8 μονάδες)

B) Τι συμπεράσματα βγάζουμε συγκρίνοντας έναν επεξεργαστή που λειτουργεί χωρίς τεχνική σωλήνωσης «non-Pipelined Execution» και έναν που λειτουργεί με την τεχνική σωλήνωσης «Pipelined Execution» πέντε (5) βημάτων;

(0.7 μονάδες)

Θέμα 2^ο

2.5 μονάδες

Ο μικροελεγκτής AVR ATmega2560 ενσωματώνει ένα «μετατροπέα αναλογικού σήματος σε ψηφιακό» «Analog to Digital Converter» (ADC) τεχνολογίας «Διαδοχικών Προσεγγίσεων» ή «Successive Approximation». Σχεδιάστε σε μπλοκ διάγραμμα και περιγράψτε την λειτουργία ενός μετατροπέα αυτής της τεχνολογίας.

(2.5 μονάδες)

Θέμα 3^ο

5.0 μονάδες

A) Να σχεδιαστεί στοιχειώδες κύκλωμα υποστήριξης μικροελεγκτή AVR AT90S8535, που χρονίζεται στα 4MHz 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 Loop». (Θεωρήστε δεδομένη την υπορουτίνα DELAY_100mSEC.)

(1.7 μονάδες)

Γ) Στη συνέχεια γράψτε νέο πρόγραμμα σε συμβολική γλώσσα «Assembly» για τον παραπάνω μικροελεγκτή, που να ελέγχει συνεχώς ένα μπλοκ μνήμης SRAM μεγέθους 19 Bytes που περιέχει μη προσημασμένους αριθμούς 8-bit. Το μπλοκ μνήμης αρχίζει από την διεύθυνση \$0100. Στο τέλος του ελέγχου ο μέγιστος αριθμός θα απεικονίζεται στην δυαδική του μορφή στα 8 LED της πόρτας A.

(2.2 μονάδες)

Δ) Γράψτε τρεις εναλλακτικούς τρόπους μηδενισμού του καταχωρητή γενικής χρήσης «r16» με χρήση μιας εντολής κάθε φορά.

(0.3 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 Loop
;=====
MAIN:    out PORTA,r16       ;
        ldi r18,5           ;
        rcall DELAY_100mSEC ;
        out PORTA,r17       ;
        ldi r18,5           ;
        rcall DELAY_100mSEC ;
        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
;=====

```