

## Εργαστήριο Δομής και Λειτουργίας Μικροϋπολογιστών

### Βοήθημα εκτέλεσης εργαστηριακής άσκησης 5: Εξωτερικές διακοπές του 8051

Ο κώδικας για την παρούσα εργαστηριακή άσκηση σας δίδεται κατά το μεγαλύτερο μέρος του έτοιμος, προκειμένου να εξοικονομηθεί χρόνος. Έτσι το βάρος θα πέσει στα σενάρια εκτέλεσης του κώδικα («τεσταρίσματα»).

#### Λειτουργία του προγράμματος

Το πρόγραμμα της παρούσας άσκησης πρέπει να κάνει τα ακόλουθα:

- Να μετακινεί έναν άσσο από αριστερά προς τα δεξιά, κυκλικά και διαρκώς, στην πόρτα **P0**.
- Όταν φθάσει σήμα διακοπής  $\overline{INT0}$ , τότε να αναβοσβήνουν όλα μαζί τα bits της πόρτας **P1** πέντε φορές.
- Όταν φθάσει σήμα διακοπής  $\overline{INT1}$ , τότε να μετακινείται ένας άσσος στην πόρτα **P2** από τα δεξιά στα αριστερά μία φορά.

#### Σύντομη περιγραφή των συνιστώντων μερών του προγράμματος

- **Κυρίως πρόγραμμα.** Ξεκινά στη διεύθυνση **0100h**.
  - Αρχικοποιεί τον stack pointer ώστε να «δείχνει» τη διεύθυνση 70h (καλύτερα αυτή την τιμή να την μεταβάλλουμε σε 50h, ώστε η στοίβα να έχει άφθονο χώρο επέκτασης προς τα επάνω).
  - Ενεργοποιεί τη δυνατότητα εξυπηρέτησης της εξωτερικής διακοπής 0, κάνοντας 1 το bit **EX0** (Enable eXternal 0).
  - Απαιτείται ακόμη η ενεργοποίηση του «κεντρικού διακόπτη» ελέγχου των διακοπών. Αυτό γίνεται κάνοντας 1 το bit **EA** (Enable All).
  - Τέλος, κάνει 1 το bit **IT0** (Interrupt Type 0), ώστε η εξωτερική διακοπή 0 να είναι **ακμοπυροδότητη (edge triggered)**.
  - Ακολουθούν εντολές **NOP** (πάνω σε αυτές θα προσθέσουμε κώδικα αργότερα), καθώς και ο κώδικας διαρκούς κυκλικής μετακίνησης του άσσου από τα αριστερά προς τα δεξιά.
- **Ρουτίνα ROUT1.** Ξεκινά στη διεύθυνση **0200h**. Η ρουτίνα αυτή καλείται κάθε φορά που φθάσει σήμα διακοπής  $\overline{INT0}$  και όπως λέμε «εξυτηρετεί» τη διακοπή.
  - Η ρουτίνα φροντίζει για το άναμμα και σβήσιμο των bits της πόρτας **P1** πέντε φορές και για το λόγο αυτό η λειτουργία της στηρίζεται σε βρόχο. Χρησιμοποιεί τον καταχωρητή **R2 (διεύθυνση 02h)** ως μετρητή επαναλήψεων του βρόχου. Για το λόγο αυτό, πριν κάνει οτιδήποτε άλλο, φυλάσσει την τιμή του **R2** στη στοίβα και αποκαθιστά αυτή την τιμή από το stack πριν επιστρέψει στο κυρίως πρόγραμμα.
- **Ρουτίνα ROUT2.** Ξεκινά στη διεύθυνση **0300h**. Η ρουτίνα αυτή εξυτηρετεί τη διακοπή  $\overline{INT1}$ .
  - Η ρουτίνα φροντίζει για τη μετακίνηση ενός άσσου από τα δεξιά προς τα αριστερά στην πόρτα **P2** μία φορά. Χρησιμοποιεί τον συσσωρευτή και τον καταχωρητή **R2 (διεύθυνση 02h)**, τους οποίους φροντίζει να προφυλάξει με τη βοήθεια της στοίβας.

- **Ρουτίνα WAIT.** Ξεκινά στη διεύθυνση **0400h**. Η ρουτίνα αυτή είναι μία κλασική ρουτίνα καθυστέρησης. Προσαρμόστε τις τιμές **#32h**, **#30h**, αν χρειάζεται, ανάλογα με την ταχύτητα του υπολογιστή σας. Η ρουτίνα χρησιμοποιεί τους καταχωρητές **R2 (διεύθυνση 02h)** και **R3 (διεύθυνση 03h)**, του οποίους και προστατεύει με τη βοήθεια της στοίβας.
- Παρατηρήστε ακόμη ότι στη διεύθυνση **0003h** υπάρχει η εντολή **LJMP ROUT1**. Όταν φθάσει σήμα διακοπής  $\overline{INT0}$ , η ροή του προγράμματος μεταφέρεται αυτόματα στη διεύθυνση **0003h**. Από εκεί, με χρήση της LJMP, μεταβαίνουμε στη διεύθυνση μνήμης με ετικέτα **ROUT1 (δηλ. στη διεύθυνση 0200h της ROM)**, όπου και βρίσκεται ο κώδικας εξυπηρέτησης της διακοπής.

### Εργαστηριακή εκτέλεση και τεσταρίσματα

#### **Βήματα προετοιμασίας.**

- Στη διεύθυνση μνήμης **0000h** προσθέστε την εντολή **LJMP 0100h**. Κάθε φορά που κάνουμε **Reset Cpu** (από το κύριο μενού του προσομοιωτή) ο Program Counter (**PC**) λαμβάνει την τιμή **0000h**. Η εντολή **LJMP**, που μόλις προσθέσαμε, εξασφαλίζει την «αυτόματη» μετάβαση στο κυρίως πρόγραμμα μετά από Reset της CPU.
- Τροποποιήστε τον stack pointer ώστε να «δείχνει» στη διεύθυνση **50h**. Με άλλα λόγια, αντικαταστήστε την εντολή στη διεύθυνση **0100h** της ROM με την εντολή **MOV SP,#50h**. Αυτό γίνεται προκειμένου να υπάρχει αρκετός χώρος, ώστε να μεγαλώσει η στοίβα χωρίς να συμβεί ποτέ υπερχείλιση της.

#### **Δουλειά με την (ακμοπυροδότητη) εξωτερική διακοπή 0**

- Πραγματοποιήστε **Reset Cpu** και, στη συνέχεια, εκτελέστε το πρόγραμμα με **F1**. Θα δείτε το άσσο στην πόρτα **P0** να κινείται διαρκώς από αριστερά προς τα δεξιά.
- Αν η κίνηση του άσσου είναι πολύ αργή ή πολύ γρήγορη στον υπολογιστή σας, μεταβείτε στη ρουτίνα καθυστέρησης στη διεύθυνση **0400h** και επιλέξτε καταλληλότερες τιμές για τον διπλό βρόχο καθυστέρησης. Όταν η κίνηση του άσσου αποκατασταθεί σε ένα σχετικά αργό, αλλά παρατηρήσιμο, ρυθμό, προχωρήστε στο επόμενο βήμα.
- Στείλτε σήμα διακοπής  $\overline{INT0}$ , **αλλάζοντας την τιμή του bit P3.2** της πόρτας **P3** από **1 σε 0 (αρνητική ακμή)**. Το bit **P3.2** είναι το **τρίτο από δεξιά**. Τι παρατηρείτε;
- Στείλτε σήμα διακοπής  $\overline{INT0}$  εκ νέου, ώστε να εκτελεστεί η ρουτίνα ROUT1 και φροντίστε να είστε αρκετά γρήγοροι ώστε να στείλετε **ένα ακόμα σήμα διακοπής  $\overline{INT0}$** , καθώς η ROUT1 «τρέχει». Τι παρατηρείτε;
- Αν δεν στείλουμε άλλο σήμα διακοπής  $\overline{INT0}$ , τι παρατηρείτε; Πώς εξελίσσεται η εκτέλεση του προγράμματος;

#### **Εκτέλεση του προγράμματος βήμα-βήμα**

- Πραγματοποιήστε **Reset Cpu** και, στη συνέχεια, εκτελέστε το πρόγραμμα βήμα-βήμα με **F10 έως ότου φθάσετε στην πρώτη εντολή NOP**.
- Στο σημείο αυτό στείλτε σήμα διακοπής  $\overline{INT0}$ . Παρατηρήστε ότι το bit-σημαία **IE0** (στα δεξιά της οθόνης, στην γκρι περιοχή) λαμβάνει αυτόματα την τιμή 1, για να καταγράψει **το αίτημα της διακοπής**.
- Πατήστε **F10** άλλη μία φορά. Σε ποιο σημείο συνεχίζει η εκτέλεση του προγράμματος και τι τιμή έχει τώρα το **IE0**; Πατήστε το **F10** μερικές φορές ακόμη και συνεχίστε με την ενότητα που ακολουθεί.

### Δουλειά με την (επιπεδοπυροδοτήτη) εξωτερική διακοπή 0

Προκειμένου να μετατρέψουμε τη διακοπή 0 σε επιπεδοπυροδοτήτη (level triggered), αρκεί να εκτελέσουμε την εξής μετατροπή:

- Στη διεύθυνση της ROM **0107h** αντικαθιστούμε την εντολή **SETB IT0** με την **CLR IT0**.

Στη συνέχεια προχωρούμε όπως και νωρίτερα:

- Πραγματοποιήστε **Reset Cpu**.
- Φροντίστε, **πριν τρέξετε το πρόγραμμα**, να επαναφέρετε, **χειροκίνητα**, την πόρτα **P3** στην κατάσταση στην οποία περιέχει μόνο άσους.
- Εκτελέστε το πρόγραμμα με **F1**.
- Στείλτε σήμα διακοπής  $\overline{INT0}$  **μηδενίζοντας το bit P3.2** της πόρτας **P3**. Το bit **P3.2** είναι το **τρίτο από δεξιά**. Κρατήστε το bit αυτό στην τιμή μηδέν και **μην** το μετατρέψετε εκ νέου σε 1. Τι παρατηρείτε; Παρατηρήστε και την τιμή του bit **IE0** (στα δεξιά της οθόνης στην γκρι περιοχή). Τι τιμή έχει αυτό το bit;
- Κάντε το bit **P3.2** **ίσο με τη μονάδα** (αυτό σημαίνει ότι παύετε να στέλνετε σήμα διακοπής  $\overline{INT0}$ ). Τι τιμή έχει τώρα το bit **IE0**; Τι παρατηρείτε όσον αφορά στην εκτέλεση του προγράμματος;

### Εκτέλεση του προγράμματος βήμα-βήμα

- Πραγματοποιήστε **Reset Cpu** και στη συνέχεια εκτελέστε το πρόγραμμα βήμα-βήμα με **F10** έως ότου φθάσετε στην πρώτη εντολή **NOP**.
- Στο σημείο αυτό, στείλτε σήμα διακοπής  $\overline{INT0}$ . Παρατηρήστε ότι το bit-σημαία **IE0** (στα δεξιά της οθόνης, στην γκρι περιοχή) λαμβάνει αυτόματα την τιμή 1, για να καταγράψει το αίτημα της διακοπής.
- Πατήστε **F10** άλλη μία φορά. Σε ποιο σημείο συνεχίζει η εκτέλεση του προγράμματος και τι τιμή έχει τώρα το **IE0**; Ποια η διαφορά σε σχέση με το σενάριο της ακμοπυροδοτήσης; Πατήστε το F10 μερικές φορές ακόμη και συνεχίστε με την ενότητα που ακολουθεί.

### Δουλειά με τις (ακμοπυροδοτήτες) εξωτερικές διακοπές 0, 1

- Κατ' αρχήν θα πρέπει να επαναφέρουμε την εξωτερική διακοπή 0 στην κατάσταση της ακμοπυροδοτήσης. Για να γίνει αυτό μεταβαίνουμε εκ νέου στη διεύθυνση της ROM **0107h** και αντικαθιστούμε την εντολή **CLR IT0** με την **SETB IT0**.
- Επιπλέον, προσθέτουμε τις εξής γραμμές κώδικα:
  - Στη διεύθυνση μνήμης ROM **0109h** την εντολή **SETB EX1** (για να ενεργοποιηθεί η δυνατότητα εξυπηρέτησης εξωτερικής διακοπής 1)
  - Στη διεύθυνση μνήμης ROM **010Bh** την εντολή **SETB IT1** (για να ρυθμιστεί η εξωτερική διακοπή 1 ως ακμοπυροδοτήτη).
- Επίσης, στη διεύθυνση μνήμης ROM **0013h** προσθέτουμε την εντολή **LJMP ROUT2 (ισοδύναμα, την εντολή LJMP 0300h)**. Αυτό διότι, όταν φθάσει σήμα διακοπής  $\overline{INT1}$ , η ροή εκτέλεσης του προγράμματος μεταφέρεται αυτόματα στη διεύθυνση **0013h**, από όπου με την εντολή **LJMP** την εκτρέπουμε στη **ROUT2 (δηλ. στη διεύθυνση 0300h της ROM, όπου και βρίσκεται το κύριο μέρος του κώδικα εξυπηρέτησης της διακοπής)**.

Στη συνέχεια προχωρούμε με παρόμοιο τρόπο όπως προηγουμένως:

- Πραγματοποιήστε **Reset Cpu**.
- Φροντίστε, **πριν τρέξετε το πρόγραμμα**, να επαναφέρετε, **χειροκίνητα**, την πόρτα **P3** στην κατάσταση στην οποία περιέχει μόνο άσους.

- Εκτελέστε το πρόγραμμα με **F1**.
- Ενεργοποιήστε τη διακοπή  $\overline{INT1}$  **αλλάζοντας την τιμή του bit P3.3 της πόρτας P3 από 1 σε 0 (αρνητική ακμή)**. Το bit P3.3 είναι το **τέταρτο από δεξιά**. Τι παρατηρείτε;
- Επιβεβαιώστε ότι εξακολουθεί να είναι δυνατός ο χειρισμός της διακοπής  $\overline{INT0}$ , **αλλάζοντας την τιμή του bit P3.2 της πόρτας P3 από 1 σε 0 (αρνητική ακμή)**. Το bit P3.2 είναι το  **τρίτο από δεξιά**.
- 

#### Ταυτόχρονη δουλειά με τις (ακμοπυροδότητες) εξωτερικές διακοπές 0, 1

- Πραγματοποιήστε **Reset Cpu**.
- Φροντίστε, **πριν τρέξετε το πρόγραμμα**, να επαναφέρετε, **χειροκίνητα**, την πόρτα **P3** στην κατάσταση στην οποία περιέχει μόνο άσσους.
- Εκτελέστε το πρόγραμμα με **F1**.

Στη συνέχεια δοκιμάστε τα εξής σενάρια:

- 1) Στείλτε σήμα διακοπής  $\overline{INT0}$  και, **καθώς η ρουτίνα εξυπηρέτησής της εκτελείται**, στείλτε σήμα διακοπής  $\overline{INT1}$ . Τι παρατηρείτε; Μπορεί η μία διακοπή να διακόψει τη ρουτίνα εξυπηρέτησης της άλλης; Γιατί;
- 2) Το ανάποδο. Στείλτε σήμα διακοπής  $\overline{INT1}$  και, **καθώς η ρουτίνα εξυπηρέτησής της εκτελείται**, στείλτε σήμα διακοπής  $\overline{INT0}$ . Τι παρατηρείτε; Μπορεί η μία διακοπή να διακόψει τη ρουτίνα εξυπηρέτησης της άλλης; Γιατί;
- 3) Στείλτε σήμα διακοπής  $\overline{INT0}$  και, **καθώς η ρουτίνα εξυπηρέτησής της εκτελείται**, στείλτε γρήγορα τόσο σήμα διακοπής  $\overline{INT1}$ , όσο και σήμα διακοπής  $\overline{INT0}$  **εκ νέου**. Τι παρατηρείτε; Ποια η σειρά εξυπηρέτησης των δύο νέων αιτημάτων διακοπής;
- 4) Το ανάποδο. Στείλτε σήμα διακοπής  $\overline{INT1}$  και, **καθώς η ρουτίνα εξυπηρέτησής της εκτελείται**, στείλτε γρήγορα τόσο σήμα διακοπής  $\overline{INT0}$ , όσο και σήμα διακοπής  $\overline{INT1}$  **εκ νέου**. Τι παρατηρείτε; Υπάρχει κάποια διαφορά με το **σενάριο 3** παραπάνω, όσον αφορά στη σειρά εξυπηρέτησης των νέων αιτημάτων διακοπής;

#### Ταυτόχρονη δουλειά με τις (ακμοπυροδότητες) εξωτερικές διακοπές 0, 1 όταν η 1 έχει υψηλότερη προτεραιότητα

Προκειμένου να δώσουμε **υψηλότερη προτεραιότητα** στην εξωτερική διακοπή 1, πρέπει να εκτελέσουμε την εξής μετατροπή:

- Στη διεύθυνση της ROM **010Dh** να προσθέσουμε την εντολή **SETB PX1** (priority external 1).

Κατόπιν συνεχίζουμε όπως και προηγουμένως:

- Πραγματοποιήστε **Reset Cpu**.
- Φροντίστε, **πριν τρέξετε το πρόγραμμα**, να επαναφέρετε, **χειροκίνητα**, την πόρτα **P3** στην κατάσταση στην οποία περιέχει μόνο άσσους.
- Εκτελέστε το πρόγραμμα με **F1**.

Δοκιμάζουμε τα εξής σενάρια:

1. Στείλτε σήμα της (**υψηλής προτεραιότητας**) διακοπής  $\overline{INT1}$ . Όσο **εκτελείται η ρουτίνα εξυπηρέτησής της**, στείλτε σήμα της (**χαμηλής προτεραιότητας**) διακοπής  $\overline{INT0}$ . Τι παρατηρείτε; Μπορεί η διακοπή  $\overline{INT0}$  να διακόψει τη ρουτίνα εξυπηρέτησης της διακοπής  $\overline{INT1}$ ;

2. Το ανάποδο. Στείλτε σήμα της **(χαμηλής προτεραιότητας)** διακοπής  $\overline{INT0}$ . Όσο **εκτελείται η ρουτίνα εξυπηρέτησής της**, σήμα της **(υψηλής προτεραιότητας)** διακοπής  $\overline{INT1}$ . Τι παρατηρείτε; Διακόπτει η  $\overline{INT1}$  τη ρουτίνα εξυπηρέτησης της  $\overline{INT0}$ ;
3. Στείλτε σήμα διακοπής  $\overline{INT0}$  και, στη συνέχεια και σχετικά γρήγορα, στείλτε σήματα διακοπής  $\overline{INT1}$ ,  $\overline{INT0}$  και  $\overline{INT1}$ . Τι παρατηρείτε και γιατί;