

Συστήματα Διακοπών και Απ'ευθείας Προσπέλαση Μνήμης

6.1 Διακοπές

Σε πολλές περιπτώσεις ένα σημαντικό μέρος ή και ολόκληρος ο χρόνος του μE καταναλώνεται στην εξυπηρέτηση περιφερειακών συσκευών, όποτε αυτές το χρειαστούν.

Αυτό μπορεί να επιτευχθεί με τη μέθοδο της συνεχούς σάρωσης (μέθοδος polling). Σύμφωνα με τη μέθοδο αυτή ο μE πρέπει να παρακολουθεί συνεχώς τις περιφερειακές συσκευές μέσω μιας πόρτας εισόδου, προκειμένου να διαπιστώσει έγκαιρα αν κάποια συσκευή χρειάζεται εξυπηρέτηση. Αυτή η συνεχής παρακολούθηση συνεπάγεται μια χρονική επιβάρυνση, η οποία είναι τόσο μεγαλύτερη όσο μεγαλύτερη είναι η συχνότητα με την οποία οι περιφερειακές συσκευές ζητούν να εξυπηρετηθούν.

Είναι φανερό ότι χρειάζεται άλλος τρόπος σύμφωνα με τον οποίον ο επεξεργαστής θα είναι ολοκληρωτικά αφιερωμένος στην εκτέλεση του κύριου προγράμματος και θα ασχολείται με κάποια περιφερειακή συσκευή, μόνον όταν αυτή ζητήσει εξυπηρέτηση. Αυτό μπορεί να επιτευχθεί με τη μέθοδο των διακοπών που θα εξετάσουμε στην αμέσως επόμενη παράγραφο.

6.1.1 Εξυπηρέτηση Περιφερειακών Συσκευών με τη Μέθοδο των Διακοπών. Πλεονεκτήματα έναντι της μεθόδου Polling

Σύμφωνα με τη μέθοδο των διακοπών, όταν μια περιφερειακή συσκευή χρειασθεί εξυπηρέτηση, έχει τη δυνατότητα να ειδοποιήσει η ίδια απ' ευθείας τον μE , ενεργοποιώντας κάποια ακίδα του, που προορίζεται ειδικά γι' αυτό το σκοπό. Ο μE απαντώντας άμεσα στην αίτηση για εξυπηρέτηση, διακόπτει το πρόγραμμα που εκτελείται εκείνη τη στιγμή και εκτελεί μια συγκεκριμένη υπορουτίνα, που έχουμε γράψει για την εξυπηρέτηση αυτής της συσκευής. Όταν ολοκληρωθεί η εκτέλεση αυτής της υπορουτίνας, το πρόγραμμα που διακόπηκε, συνεχίζεται από το σημείο που έγινε η διακοπή. Στην περίπτωση αυτή λέμε ότι η συσκευή προκάλεσε διακοπή ή ότι έκανε αίτηση διακοπής και ο μE , αφού πρώτα αναγνώρισε τη διακοπή, την εξυπηρέτησε. Η υπορουτίνα που εκτελείται

σ'αυτήν την περίπτωση ονομάζεται υπορουτίνα εξυπηρέτησης διακοπής. Η ειδική ακίδα μέσω της οποίας προκαλείται η διακοπή ονομάζεται είσοδος διακοπής.

Το γεγονός ότι ο μE δεν χρειάζεται πλέον να παρακολουθεί συνεχώς κάποια πόρτα εισόδου, συνεπάγεται εξοικονόμηση χρόνου, γιατί μπορεί να αφιερώσει όλο το χρόνο του στο κύριο πρόγραμμα. Θα ασχοληθεί με κάποια περιφερειακή συσκευή μόνον όταν αυτή ζητήσει εξυπηρέτηση, υποβάλλοντας αίτηση διακοπής. Εκτός όμως από την εξοικονόμηση χρόνου η μέθοδος των διακοπών απλουστεύει τη δομή του προγράμματος. Το τμήμα προγράμματος που είδαμε στη παράγραφο 4.7, το οποίο εξετάζει το bit κατάστασης της συσκευής και καλεί το πρόγραμμα εξυπηρέτησης, δεν χρειάζεται πλέον αφού μπορεί να γραφεί σε μορφή ρουτίνας εξυπηρέτησης, που καλείται αυτόματα όταν προκληθεί διακοπή. Το κύριο πρόγραμμα μπορεί να γραφεί εντελώς ανεξάρτητα χωρίς να γίνεται καμία αναφορά (κλήση) στη ρουτίνα εξυπηρέτησης.

Επίσης μεταφορά δεδομένων με βάση διακοπές για σύνδεση του μE με σχετικά αργές συσκευές E/E, όπως A/D μετατροπέας, εκτυπωτές κλπ. γίνεται με την εξής τεχνική. Ο μE πρώτα αρχικοποιεί τη συσκευή E/E και συνεχίζει την εκτέλεση της κύριας διαδικασίας. Όταν ολοκληρωθεί η διαδικασία για παράδειγμα της μετατροπής από αναλογική τάση σε ψηφιακή μορφή ενός A/D μετατροπέα, τότε η συσκευή προκαλεί διακοπή στο μE . Η ρουτίνα εξυπηρέτησης της διακοπής αναλαμβάνει τη λήψη και επεξεργασία του δεδομένου. Μετά την ολοκλήρωσή της επιστρέφει στο κύριο πρόγραμμα. Αν και η διαδικασία είναι παρόμοια με κλήση υπορουτίνας το κύριο πρόγραμμα δεν περιέχει εντολή `CALL`. Το πρόγραμμα μπορεί να διακοπεί σε τυχαίο σημείο. Η ρουτίνα εξυπηρέτησης της διακοπής δεν διαφέρει από τις άλλες ρουτίνες του συστήματος και πρέπει να τερματίζεται όπως αυτές με την εντολή `RET`.

6.1.2 Διεύθυνση Διακοπής

Με βάση τα παραπάνω θα μπορούσε να θεωρηθεί, ότι η διακοπή είναι ουσιαστικά η κλήση μιας υπορουτίνας που γίνεται από το υλικό και όχι από το πρόγραμμα. Όταν όμως μια υπορουτίνα καλείται μέσα από το πρόγραμμα με μια εντολή `CALL` δίνεται πάντοτε η διεύθυνση της μνήμης απ' όπου αυτή αρχίζει. Στην περίπτωση όμως της διακοπής η υπορουτίνα εξυπηρέτησης καλείται όταν ενεργοποιηθεί η είσοδος διακοπής από την περιφερειακή συσκευή χωρίς να δίνεται καμία διεύθυνση. Τίθεται επομένως το ερώτημα σε ποιά διεύθυνση της μνήμης θα μεταφερθεί ο έλεγχος του προγράμματος προκειμένου να εκτελεστεί η ρουτίνα εξυπηρέτησης. Η διεύθυνση αυτή είναι καθορισμένη από τον κατασκευαστή του μE για κάθε τύπο διακοπής και ονομάζεται διεύθυνση διακοπής. Συνεπώς η ρουτίνας

εξυπηρέτησης πρέπει να φορτωθεί στη μνήμη αρχίζοντας από αυτή τη συγκεκριμένη διεύθυνση. Φυσικά με μια εντολή JMP Νέα Διεύθυνση μπορούμε να μεταθέσουμε το σημείο εκκίνησης της ρουτίνας εξυπηρέτησης.

Οι πιο προχωρημένοι επεξεργαστές (16 και 32 bits) έχουν έναν πίνακα διανυσμάτων διακοπών. Στον πίνακα αυτό για κάθε τύπο διακοπής αντιστοιχεί μια διεύθυνση, που μπορεί να τοποθετηθεί στον πίνακα από τον χρήστη μέσω προγράμματος.

6.1.3 Προτεραιότητα Διακοπών

Συνήθως οι περιφερειακές συσκευές ενός μΥ είναι περισσότερες από μία γι' αυτό και οι περισσότεροι μΕ διαθέτουν περισσότερες από μία εισόδους διακοπής. Για κάθε είσοδο καθορίζεται από τον κατασκευαστή η διεύθυνση διακοπής όπου πρέπει να φορτωθεί η αντίστοιχη ρουτίνα εξυπηρέτησης. Επίσης πολλές συσκευές μπορούν να είναι συνδεδεμένες σε μία κοινή γραμμή διακοπής

Οι πολλές εισόδους διακοπής σε συνδυασμό με τις πολλαπλές συσκευές καθιστούν απαραίτητο τον καθορισμό προτεραιότητας για την περίπτωση που περισσότερες από μία συσκευές ζητήσουν ταυτόχρονα εξυπηρέτηση. Εδώ περιλαμβάνεται και η περίπτωση που μία συσκευή ζητάει εξυπηρέτηση ενώ εξυπηρετείται μια άλλη. Η προτεραιότητα των εισόδων θα καθορίσει και τη σειρά με την οποία θα γίνει η εξυπηρέτηση των συσκευών. Η προτεραιότητα καθορίζεται από τον κατασκευαστή και δεν μπορεί να μεταβληθεί. Μερικοί επεξεργαστές όμως επιτρέπουν την αλλαγή της προτεραιότητας των διακοπών μέσα από το πρόγραμμα.

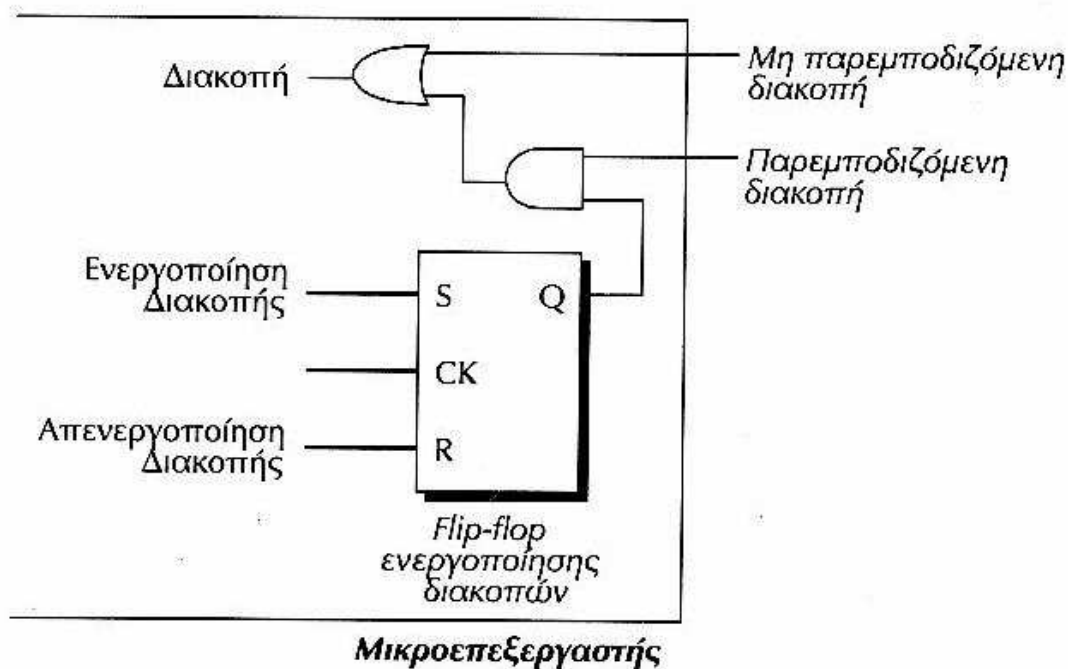
6.1.4 Ενεργοποίηση και Απενεργοποίηση του Μηχανισμού των Διακοπών. Παρεμποδιζόμενες Διακοπές

Ορισμένες φορές, στο πρόγραμμα που εκτελείται υπάρχουν κρίσιμα τμήματα τα οποία θα πρέπει να εκτελεστούν χωρίς διακοπής. Για παράδειγμα αυτό θα μπορούσε να συμβεί στην περίπτωση που υπάρχουν χρονικοί περιορισμοί για ολοκλήρωση της εκτέλεσης των διαφόρων τμημάτων του προγράμματος. Στην περίπτωση αυτή είναι αναγκαίο να απενεργοποιηθεί για κάποιο διάστημα ο μηχανισμός των διακοπών. Όταν συμβεί αυτό ο μΕ αγνοεί οποιαδήποτε αίτηση διακοπής μέχρι ο μηχανισμός των διακοπών να ενεργοποιηθεί και πάλι. Η ενεργοποίηση και η απενεργοποίηση γίνεται με ειδικές εντολές, που πρέπει να ενσωματωθούν στα κατάλληλα σημεία του προγράμματος.

Άλλες φορές όμως δεν χρειάζεται να απενεργοποιηθεί όλος ο μηχανισμός των διακοπών αλλά να παρεμποδιστούν επιλεκτικά κάποιες

απ' αυτές. Οι περισσότεροι μΕ μας παρέχουν τη δυνατότητα να παρεμποδίζουμε, να θέτουμε μάσκα διακοπών, όπως αλλιώς λέγεται, σε όλες ή σε κάποιες από τις εισόδους διακοπών. Οι εισόδοι αυτοί ονομάζονται παρεμποδιζόμενες διακοπές επειδή μπορούν να παρεμποδιστούν με τη χρήση μάσκας. Όπως και στην περίπτωση της ολικής απενεργοποίησης του μηχανισμού των διακοπών, το ρεπερτόριο εντολών των μΕ περιέχει ειδικές εντολές για την τοποθέτηση και την αφαίρεση της μάσκας στις διακοπές. Έτσι το ποιές διακοπές θα παρεμποδιστούν και σε ποιά χρονική στιγμή είναι αποκλειστική ευθύνη του προγράμματος.

Στο σχήμα 6.1 φαίνεται το κύκλωμα, με το οποίο τίθεται μάσκα σε μια είσοδο διακοπής.



Σχήμα 6.1 Είσοδοι διακοπών με μάσκα και χωρίς μάσκα

Οι εισόδοι S και R του flip-flop ενεργοποιούνται ή απενεργοποιούνται από την από τις ειδικές εντολές, με τις οποίες τίθεται ή αφαιρείται η μάσκα στις εισόδους διακοπών.

6.1.5 Διαδικασία εξυπηρέτησης Διακοπής

Στη συνέχεια θα εξετάσουμε λεπτομερέστερα τι συμβαίνει σε έναν επεξεργαστή όταν αναγνωριστεί μία αίτηση διακοπής:

- Ολοκληρώνεται η εντολή την οποίαν εκτελούσε ο μΕ τη στιγμή που ενεργοποιήθηκε η είσοδος διακοπής.
- Απενεργοποιούνται όλες οι διακοπές

- Σώζεται στη στοίβα η κατάσταση του μE . Στους περισσότερους μE σαν κατάσταση θεωρείται και σώζεται ο μετρητής προγράμματος και ο καταχωρητής σημαιών. Ο μετρητής προγράμματος είναι απαραίτητο να σωθεί επειδή δείχνει από ποιά εντολή πρέπει να συνεχιστεί η εκτέλεση του προγράμματος μετά την εκτέλεση της ρουτίνας εξυπηρέτησης της διακοπής. Επίσης σώζεται και ο καταχωρητής σημαιών γιατί η διακοπή μπορεί να συμβεί πριν από μια εντολή που σχετίζεται με τις σημαίες, για παράδειγμα άλμα υπό συνθήκη.
- Εκτελείται η υπορουτίνα εξυπηρέτησης της διακοπής. Η υπορουτίνα αυτή πρέπει να γραφεί έτσι ώστε στο ξεκίνημά της να σώζει πλήρως την κατάσταση του μE μιας και όπως είδαμε στο προηγούμενο βήμα σώζεται συνήθως μόνο ο μετρητής προγράμματος και ο καταχωρητής σημαιών. Συγκεκριμένα όλοι οι καταχωρητές των οποίων οι τιμές τροποποιούνται από την υπορουτίνα εξυπηρέτησης πρέπει να σωθούν. Ο πιο απλός τρόπος για να γίνει αυτό είναι να σωθούν στη στοίβα με τη χρήση της εντολής **PUSH**. Ακόμη πρέπει οπωσδήποτε να σωθεί και ο καταχωρητής σημαιών στη στοίβα στην περίπτωση που αυτό δεν γίνεται αυτόματα από τον μE . Όταν η υπορουτίνα ολοκληρωθεί και πρίν επιστρέψει πρέπει να ανακτηθούν από τη στοίβα όσοι καταχωρητές σώθηκαν στην αρχή. Η ανάκτηση θα πρέπει να γίνει με την αντίστροφη σειρά που έγινε η αποθήκευση. Ακόμη θα πρέπει οπωσδήποτε να ενεργοποιηθούν με τη χρήση ειδικής εντολής στο πρόγραμμα οι διακοπές, αφού απενεργοποιήθηκαν αυτόματα στο βήμα 2.
- Ανακτάται η κατάσταση του μE , που αποθηκεύτηκε στο βήμα 3 και επιστρέφεται ο έλεγχος στο πρόγραμμα που διακόπηκε στην επόμενη εντολή μετά από αυτήν, στην διάρκεια της οποίας συνέβει η διακοπή.

Από τα παραπάνω είναι προφανές ότι υπάρχει μία χρονική επιβάρυνση (timing overhead), που οφείλεται στα βήματα 1, 2, 3 και 5 και στις εντολές για το σώσιμο και την ανάκτηση των καταχωρητών που πρέπει να περιλαμβάνει η υπορουτίνα εξυπηρέτησης στο βήμα 4. Αυτή η επιβάρυνση πρέπει να ληφθεί υπόψιν όταν πρόκειται να διαλέξουμε για την εξυπηρέτηση περιφερειακών συσκευών ανάμεσα στη μέθοδο των διακοπών και τη διαδοχική σάρωση (μέθοδος polling). Στη μέθοδο polling η χρονική επιβάρυνση οφείλεται στον βρόγχο που εξετάζει συνεχώς αν κάποια συσκευή χρειάζεται εξυπηρέτηση. Το βασικό κριτήριο επιλογής ανάμεσα στις δύο μεθόδους είναι η συχνότητα με την οποίον γίνονται αιτήσεις για εξυπηρέτηση από τις περιφερειακές συσκευές. Όταν η συχνότητα αυτή είναι μεγάλη η μέθοδος polling είναι η πιο κατάλληλη ενώ η μέθοδος των διακοπών ενδείκνυται στην περίπτωση που οι διακοπές συμβαίνουν σποραδικά.

Στην παράγραφο αυτή είδαμε τι συμβαίνει όταν αναγνωριστεί η διακοπή. Στην επόμενη παράγραφο θα εξετάσουμε τον μηχανισμό αναγνώρισης των διακοπών.

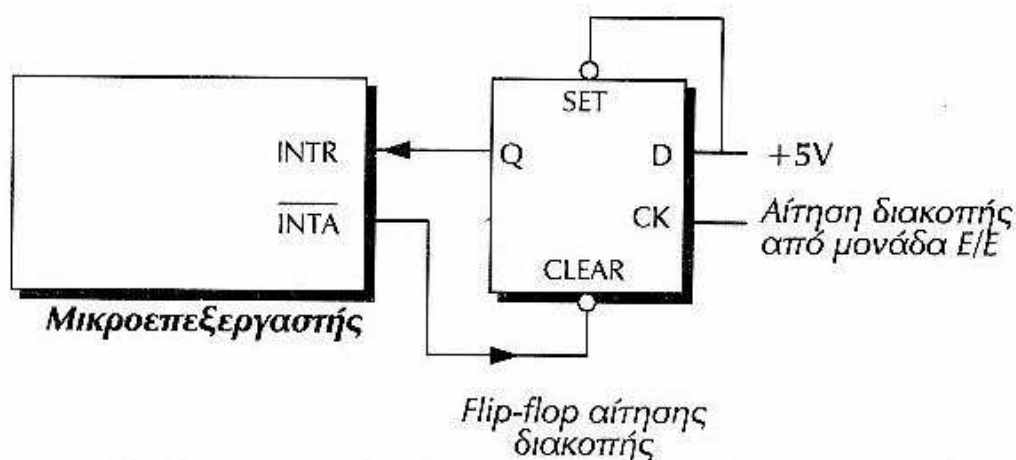
6.1.6 Αναγνώριση Διακοπής από το Υλικό του μE .

Είσοδοι Διακοπής Ευαίσθητες στο επίπεδο της τάσης

Μετά την ολοκλήρωση και του τελευταίου κύκλου μηχανής κάθε εντολής, ο μE , εξετάζει το επίπεδο της τάσης ("1" ή "0") σε όλες τις εισόδους διακοπών στις οποίες δεν έχει τεθεί μάσκα, για να δει αν κάποια από αυτές είναι ενεργοποιημένη. Αν ο μE αναγνωρίζει μια διακοπή εξετάζοντας το επίπεδο της τάσης λέμε ότι αυτές οι εισόδους είναι ευαίσθητες στο επίπεδο της τάσης (level sensitive). Όταν αναγνωριστεί η διακοπή, η περιφερειακή συσκευή θα πρέπει να απενεργοποιήσει αυτή την είσοδο γιατί μετά την ενεργοποίηση των διακοπών στο πέρας της ρουτίνας εξυπηρέτησης ο μE θα αναγνωρίσει και νέα διακοπή. Το αποτέλεσμα θα είναι αλληπάλληλες διακοπές στην ίδια είσοδο.

Οι αιτήσεις όμως για διακοπές από τις περιφερειακές συσκευές υποβάλλονται ασύγχρονα στο μE , και μπορεί να συμβούν σε οποιοδήποτε σημείο του προγράμματος. Έτσι αν η αίτηση υποβληθεί στη μέση της εκτέλεσης μιας εντολής θα ολοκληρωθεί πρώτα και ο τελευταίος κύκλος μηχανής της εντολής και στη συνέχεια θα αγνωριστεί η διακοπή. Συνεπώς, η είσοδος διακοπής θα πρέπει να μείνει ενεργοποιημένη μέχρι να αναγνωριστεί και όταν γίνει αυτό θα πρέπει να απενεργοποιηθεί.

Το κύκλωμα του σχήματος 6.2 μπορεί να χρησιμοποιηθεί για να επιτευχθεί αυτό.



Σχήμα 6.2 Εξωτερικό κύκλωμα για την αναγνώριση διακοπής

Όπως φαίνεται στο σχήμα, ένα flip-flop παρεμβάλεται μεταξύ της συσκευής και της εισόδου διακοπής. Όταν συσκευή ζητήσει εξυπηρέτηση η τάση στην είσοδο CLK του flip-flop, που συνδέεται με αυτήν, μεταβαίνει

από "0" σε "1". Αυτό έχει σαν αποτέλεσμα η έξοδος Q του flip-flop να έρθει σε κατάσταση "1" και θα παραμείνει σ' αυτήν ακόμη και αν η έξοδος της συσκευής που συνδέεται με το CLK γίνει και πάλι "0". Όταν η διακοπή αναγνωριστεί από το μE θα πρέπει η έξοδος Q να γίνει "0" δηλαδή το flip-flop να γίνει CLEAR. Όπως φαίνεται στο σχήμα, αυτό επιτυγχάνεται από το σήμα εξόδου $\overline{\text{INTA}}$ του μE , το οποίο γίνεται "0" μόλις αναγνωριστεί η διακοπή. Εξοδο σαν την $\overline{\text{INTA}}$ (INTerrupt Acknowledgement) παρέχουν σχεδόν όλοι οι μE για να γνωστοποιήσουν ότι αναγνώρισαν μια διακοπή.

Αφού το flip-flop γίνει CLEAR, η είσοδος INTR παραμένει σε κατάσταση "0" ακόμη και αν η αίτηση διακοπής της συσκευής παραμένει σε κατάσταση "1". και με αυτόν τον τρόπο αποφεύγονται η αλληπάληλες διακοπές. Σε περίπτωση που η συσκευή αυτή θέλει να προκαλέσει μια καινούργια διακοπή, θα πρέπει η έξοδος της να μεταβεί πρώτα σε κατάσταση "0" και μετά πάλι σε κατάσταση "1".

Αρκετοί επεξεργαστές περιλαμβάνουν εσωτερικά το κύκλωμα του σχήματος 6.2 διευκολύνοντας έτσι τη σύνδεση περιφερειακών συσκευών. Στην περίπτωση αυτή η αίτηση διακοπής δεν αναγνωρίζεται από το επίπεδο της τάσης αλλά το μέτωπο ενός παλμού που στέλνει η περιφερειακή συσκευή. Αυτό άμεση συνέπεια της λειτουργίας του κυκλώματος του σχήματος 6.2, όπου για να αναγνωριστεί η διακοπή πρέπει η τάση στην είσοδο CLK να μεταβεί από "0" σε "1" (το μέτωπο του παλμού) χωρίς να χρειάζεται να παραμείνει σε κατάσταση "1" μέχρι να αναγνωριστεί από τον μE . Στη συνέχεια όμως πρέπει να επανέλθει πάλι σε κατάσταση "0" ώστε να μπορεί να προκληθεί νέα διακοπή. Γι' αυτό και μια τέτοια είσοδος διακοπής ονομάζεται ακμοπυροδότητη (edge triggered).

Το κύκλωμα του σχήματος 6.2 μπορεί και να λειτουργήσει χωρίς να υπάρχει η έξοδος $\overline{\text{INTA}}$. Στην περίπτωση αυτή μια πόρτα εξόδου μπορεί να χρησιμοποιηθεί γι' αυτόν το σκοπό και η είσοδος CLR του flip-flop θα συνδεθεί σε κάποιο bit της πόρτας. Όταν αναγνωριστεί η διακοπή η υπορουτίνα εξυπηρέτησης θα πρέπει να εκχωρήσει την κατάλληλη τιμή στο bit αυτό γνωστοποιώντας έτσι την αναγνώριση της διακοπής.

6.1.7 Είσοδος Πολλαπλών Διακοπών

Η ύπαρξη περισσότερων από μία εισόδων διακοπών δε λύνει το πρόβλημα της σύνδεσης πολλών συσκευών με τον μE μέσω διακοπών, διότι εξαιτίας του περιορισμένου αριθμού ακίδων που μπορούν να προστεθούν σε ένα ολοκληρωμένο ο αριθμός των εισόδων διακοπής είναι αρκετά μικρός, το πολύ τρεις ή τέσσερις εισοδοί. Για το λόγο αυτό οι περισσότεροι μE διαθέτουν μια γενική είσοδο διακοπής INTR και τον αντίστοιχο σήμα εξόδου $\overline{\text{INTA}}$ για την αναγνώριση της διακοπής. Στην είσοδο INTR μπορούν να συνδεθούν πολλές περιφερειακές συσκευές.

Οποιαδήποτε από αυτές τις συσκευές χρειαστεί εξυπηρέτηση, ενεργοποιεί την είσοδο INTR και αναμένει την ενεργοποίηση του σήματος $\overline{\text{INTA}}$ από τον μE , γεγονός που δείχνει ότι αυτός έχει αναγνωρίσει τη διακοπή. Όταν συμβεί αυτό, η συσκευή τοποθετεί έναν αριθμό των 8 bits πάνω στο διάδρομο δεδομένων γνωστοποιώντας έτσι στον μE την ταυτότητά της.

Ο αριθμός αυτός ονομάζεται αριθμός διακοπής και με βάση αυτόν ο μE καλεί αυτόματα τη ρουτίνα εξυπηρέτησης που έχει γραφεί γι' αυτή τη διακοπή. Σε κάθε αριθμό διακοπής αντιστοιχεί και μια συγκεκριμένη διεύθυνση στη μνήμη όπου πρέπει να φορτωθεί η αντίστοιχη ρουτίνα εξυπηρέτησης. Με τον τρόπο αυτόν ο αριθμός των διακοπών, που μπορεί να αναγνωρίσει ένας μE , αυξάνεται σημαντικά χωρίς να απαιτούνται επιπλέον εισόδοι για διακοπές.

Είναι προφανές ότι η ανάγνωση του αριθμού διακοπής από τον διάδρομο δεδομένων είναι μέρος της διαδικασίας αναγνώρισης και εξυπηρέτησης της διακοπής, γίνεται αυτόματα από τον μE και δεν απαιτείται κάποια σχετική εντολή E/E στο πρόγραμμά μας.

6.2 Το Σύστημα Διακοπών του μE 8085

Η σχεδίαση του 8085 υποστηρίζει ικανοποιητικά την επικοινωνία των περιφερειακών συσκευών και του μE μέσω διακοπών. Διαθέτει τέσσερεις εισόδους διακοπών καθώς και μία είσοδο πολλαπλών διακοπών.

Κατά την εκκίνηση, ο μηχανισμός των διακοπών είναι αρχικά απενεργοποιημένος και γι' αυτό πρέπει να ενεργοποιηθεί με την εκτέλεση της κατάλληλης εντολής μέσα από το πρόγραμμα. Στη συνέχεια θα εξεταστούν αναλυτικότερα οι εισόδοι διακοπών και θα περιγραφούν και οι εντολές που παρέχει ο 8085 για τον χειρισμό των διακοπών.

6.2.1 Είσοδοι Παρεμποδιζόμενων Διακοπών

Ο μE 8085 διαθέτει τρεις εισόδους RST 7.5, RST 6.5 και RST 5.5 παρεμποδιζόμενων διακοπών. Η RST 7.5 έχει την υψηλότερη προτεραιότητα και RST 5.5 τη μικρότερη. Από τις εισόδους αυτές μόνο η RST 7.5 είναι ακμοπυροδότητη ενώ οι RST 6.5 και RST 5.5 είναι ευαίσθητες στο επίπεδο της τάσης.

Οι διευθύνσεις μνήμης που αντιστοιχούν σε αυτές τις διακοπές είναι 3CH για την RST 7.5, 34H για την RST 6.5, και 2CH για την RST 5.5. Οι διευθύνσεις αυτές βρίσκονται πολύ κοντά ή μία στην άλλη, μόλις 8 bytes. Το γεγονός αυτό δεν επιτρέπει να φορτωθούν οι υπορουτίνες εξυπηρέτησης αρχίζοντας από τις θέσεις αυτές. Γι' αυτό στις παραπάνω διευθύνσεις φορτώνονται εντολές άλματος, οι οποίες μεταφέρουν τον έλεγχο στις

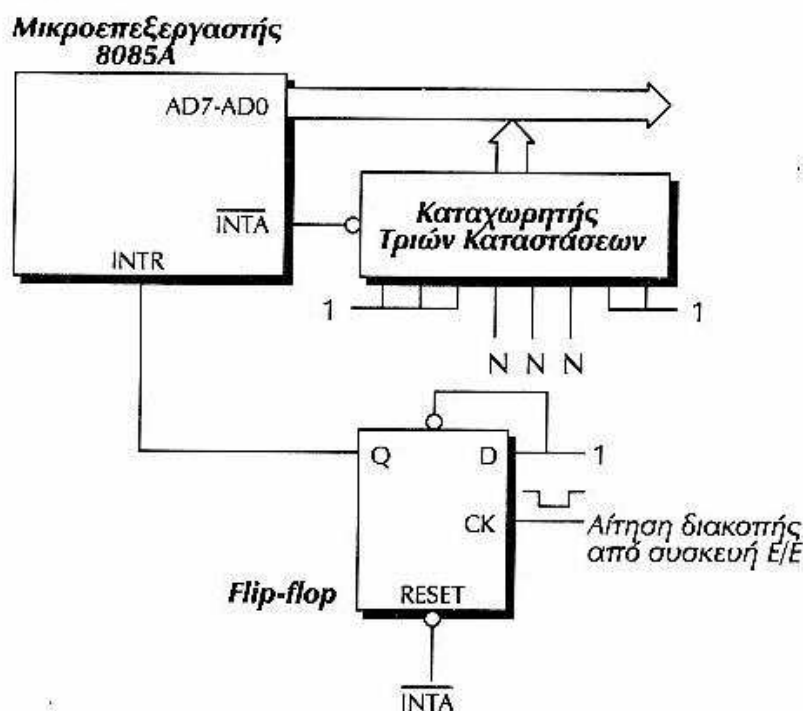
ρουτίνες εξυπηρέτησης, μόλις αναγνωρισθεί η αντίστοιχη διακοπή. Έτσι οι κλήση των υπορουτινών αυτών γίνεται έμμεσα.

Με τη χρήση μάσκας διακοπών αυτές οι εισοδοι μπορούν να παρεμποδιστούν επιλεκτικά.

6.2.2 Είσοδος πολλαπλών διακοπών

Ο 8085 διαθέτει μία είσοδο πολλαπλών διακοπών την INTR καθώς και την έξοδο \overline{INTA} , που γνωστοποιεί την αναγνώριση της διακοπής από τον μΕ. Η είσοδος INTR έχει μικρότερη προτεραιότητα και από τις τρεις εισόδους που αναφέρθηκαν στην προηγούμενη παράγραφο και είναι ευαίσθητη στο επίπεδο της τάσης. Η διεύθυνση διακοπής για την INTR καθορίζεται, όπως είδαμε και στην παράγραφο 6.1.8 από τον κωδικό που τοποθετεί η περιφερειακή συσκευή πάνω στο διάδρομο δεδομένων μόλις λάβει το σήμα \overline{INTA} . Πιο συγκεκριμένα εκτελείται κύκλος ανάκλησης εντολής. Συνεπώς το εξωτερικό κύκλωμα θεωρητικά μπορεί να θέσει τον κωδικό οποιασδήποτε εντολής. Πρακτικά όμως ο κωδικός αυτός είναι είτε η εντολή CALL ή συνηθέστερα η εντολή RST n ($0 \leq n \leq 7$), που είδαμε στην παράγραφο 5.10. Στη πρώτη περίπτωση η διεύθυνση της διακοπής είναι αυτή που συνοδεύει την εντολή CALL ενώ στη δεύτερη είναι n×8. Η κλήση της αντίστοιχης υπορουτίνας εξυπηρέτησης γίνεται και πάλι έμμεσα, μέσω της κατάλληλης εντολής άλματος, που πρέπει να έχει φορτωθεί στη διεύθυνση διακοπής.

Στο σχήμα 6.3 φαίνεται η διασύνδεση μιας συσκευής με τον 8085 μέσω τις εισόδου INTR.



Σχήμα 6.3 Κύκλωμα για την υλοποίηση ανυσματικής διακοπής στον 8085

Στο διάδρομο δεδομένων τοποθετείται η εντολή **RST** μέσω ενός απομονωτή τριών καταστάσεων (tri-state buffer) ο οποίος ελέγχεται από το σήμα $\overline{\text{INTR}}$. Υπενθυμίζεται ότι η δυαδική μορφή της εντολής **RST** (n) είναι 11NNN111 όπου NNN το όρισμα n (0 έως 7) της εντολής σε δυαδική μορφή. Το flip-flop χρησιμοποιείται επειδή η είσοδος **INTR** είναι ευαίσθητη στο επίπεδο της τάσης.

Η διακοπή **INTR** δεν επιδέχεται μάσκα. Ο μόνος τρόπος για να παρεμποδιστεί αυτή η διακοπή είναι με την απενεργοποίηση του μηχανισμού των διακοπών (Εντολή **DI:Disable Interrupts**).

6.2.3 Η διακοπή **TRAP**

Η είσοδος διακοπής **TRAP**, είναι μια ειδική είσοδος διακοπής. Έχει την υψηλότερη προτεραιότητα απ' όλες τις διακοπές του 8085, δεν επιδέχεται μάσκα και δεν παρεμποδίζεται ακόμα και στην περίπτωση που απενεργοποιηθεί ο μηχανισμός των διακοπών. Κατά την εκκίνηση η **TRAP** είναι η μόνη διακοπή στην οποία αποκρίνεται ο 8085. Πρόκειται δηλαδή για μια διακοπή την οποία ο 8085 θα εξυπηρετήσει οπωσδήποτε και μάλιστα άμεσα αφού έχει την υψηλότερη προτεραιότητα και δεν υπάρχει τρόπος να παρεμποδιστεί.

Προφανώς η διακοπή αυτή προορίζεται για περιφερειακές συσκευές που επαγρυπνούν για την εμφάνιση έκτακτων καταστάσεων, όπως για παράδειγμα πτώση της τάσης τροφοδοσίας, εμφάνιση βλαβών κλπ. Τέτοιου είδους καταστάσεις μπορεί να προκαλέσουν καταστροφή τόσο των δεδομένων που διαχειρίζεται ο $\mu\text{Υ}$, όσο και του ίδιου υλικού και γι' αυτό το λόγο η αντιμετώπιση από τον $\mu\text{Ε}$ πρέπει να είναι άμεση.

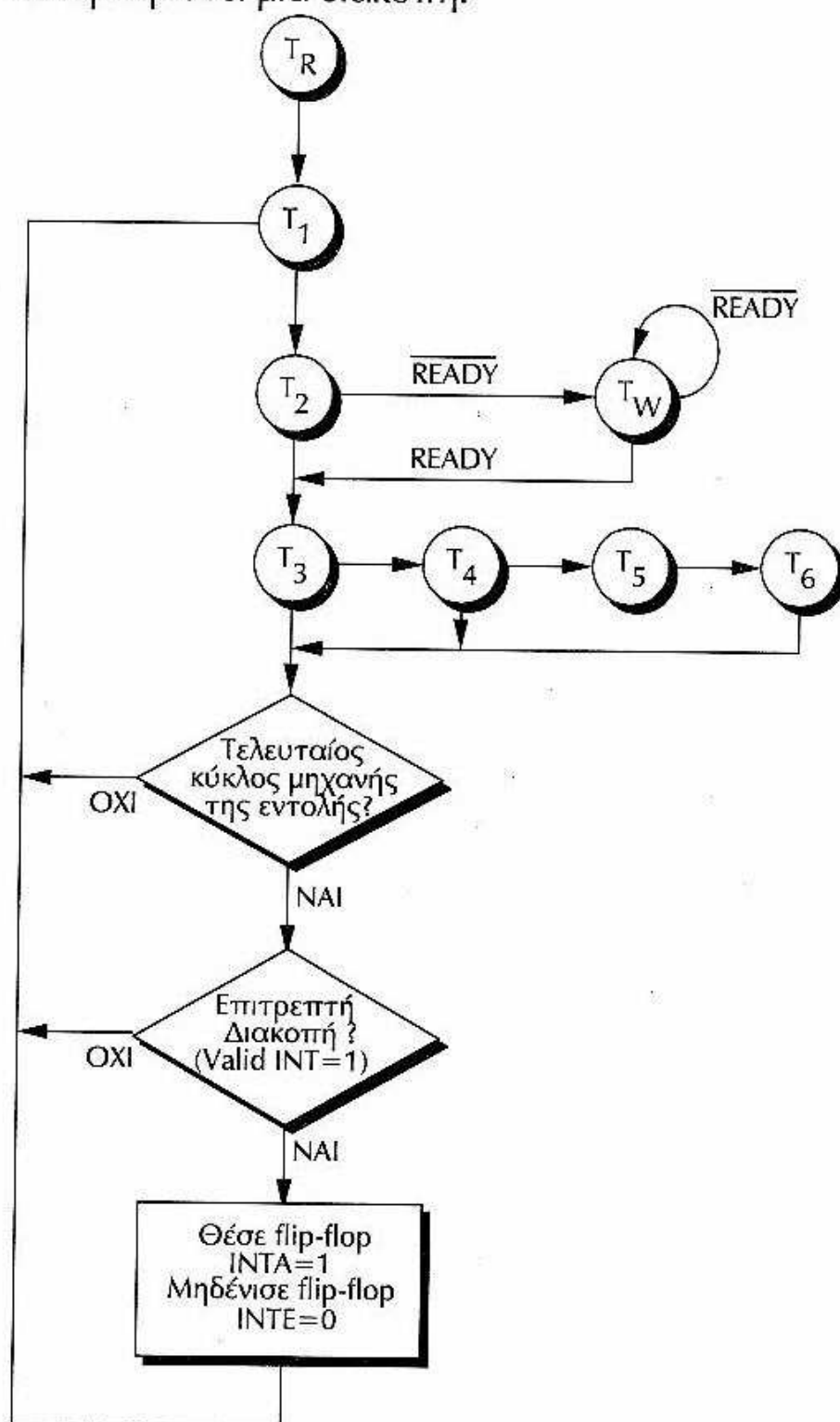
Για να αποφευχθεί εσφαλμένη πυροδότηση εξαιτίας θορύβου η είσοδος **TRAP** είναι ευαίσθητη και στο θετικό μέτωπο του παλμού και στο επίπεδο της τάσης. Αυτό σημαίνει ότι για να αναγνωριστεί από τον $\mu\text{Ε}$, θα πρέπει η τάση στην είσοδο **TRAP** να μεταβεί από "0" σε "1" και να παραμείνει σε κατάσταση "1" μέχρι τη στιγμή της αναγνώρισής της. Η διεύθυνση διακοπής για την **TRAP** είναι 024H.

6.2.4 Αναγνώριση διακοπών από τον 8085

Στο σχήμα 6.4 φαίνεται το διάγραμμα καταστάσεων ενός κύκλου μηχανής του 8085, στο οποίο έχει συμπεριληφθεί και ο έλεγχος των διακοπών.

Όπως φαίνεται και στο σχήμα, ο 8085 ελέγχει την ύπαρξη αίτησης για επιτρεπτή διακοπή μετά τον τελευταίο κύκλο μηχανής κάθε εντολής, οπότε και αποτιμάται η συνθήκη **VALID INT**. Η συνθήκη αυτή έχει την ακόλουθη μορφή:

$\text{VALID INT} = \text{TRAP} + \text{INTE} \times [\text{INTR} + \text{RST7.5} \times \overline{\text{M7.5}} + \text{RST6.5} \times \overline{\text{M6.5}} + \text{RST5.5} \times \overline{\text{M5.5}}]$
 όπου Mn.5 είναι το bit μάσκας για τη διακοπή RSTn.5 και $\overline{\text{Mn.5}}$ είναι το συμπλήρωμά του. Η συνθήκη αυτή παίρνει την τιμή 1 αν εκκρεμεί κάποια διακοπή. Αν ο μE βρίσκεται σε κατάσταση HALT ο έλεγχος γίνεται στο τέλος του κάθε παλμού ρολογιού. Μόλις αναγνωριστεί μια διακοπή ένα εσωτερικό flip-flop INTA, που παίζει το ρόλο σημαίας, γίνεται "1" δείχνοντας έτσι ότι έχει αναγνωριστεί μια διακοπή.



Σχήμα 6.4 Απλοποιημένος κύκλος διακοπής του 8085

Επίσης ένα άλλο εσωτερικό flip-flop το INTE (INTerrupt Enable), το οποίο ενεργοποιεί ή απενεργοποιεί το μηχανισμό των διακοπών γίνεται "0" απενεργοποιώντας έτσι τις διακοπές εκτός από την TRAP. Στη συνέχεια, ο επεξεργαστής μπαίνει σε έναν ειδικό κύκλο μηχανής για την εξυπηρέτηση των διακοπών.

Αν η διακοπή που αναγνωρίστηκε είναι μια από τις TRAP, RST 7.5, RST6.5, RST 5.5, εκτελείται κύκλος μηχανής για τις διακοπές αυτές, που είναι ο "άργος" κύκλος (BI, Bus Idle) (βλέπε παραγρ. 2.5.3.) Στη διάρκεια αυτού του κύκλου προετοιμάζεται για εκτέλεση μια παραλλαγή της εντολής RST(n). Η εντολή αυτή ονομάζεται εσωτερική RST επειδή δεν συμπεριλαμβάνεται στο ρεπερτόριο εντολών του επεξεργαστή. Η εκτέλεση αυτής της εντολής που έχει σαν αποτέλεσμα τις ακόλουθες ενέργειες:

((SP)-1) \leftarrow (PCH)

((SP)-2) \leftarrow (PCL)

(SP) \leftarrow (SP)-2

(PC) \leftarrow Διεύθυνση της διακοπής που αναγνωρίστηκε

Το περιεχόμενο του μετρητή προγράμματος (PC) σώζεται στη στοίβα και στη συνέχεια η διεύθυνση που αντιστοιχεί στη διακοπή που αναγνωρίστηκε εκχωρείται στον PC. Η τελευταία ενέργεια έχει σαν αποτέλεσμα ο έλεγχος του προγράμματος να μεταφερθεί στη διεύθυνση διακοπής και από εκεί με την εντολή άλματος στη ρουτίνα εξυπηρέτησης της διακοπής. Από τα παραπάνω βλέπουμε ότι η μοναδική διαφορά της εσωτερικής RST από την RST(n) είναι ότι ο έλεγχος του προγράμματος δεν μεταφέρεται στη διεύθυνση n×8 αλλά στη διεύθυνση της διακοπής που αναγνωρίστηκε.

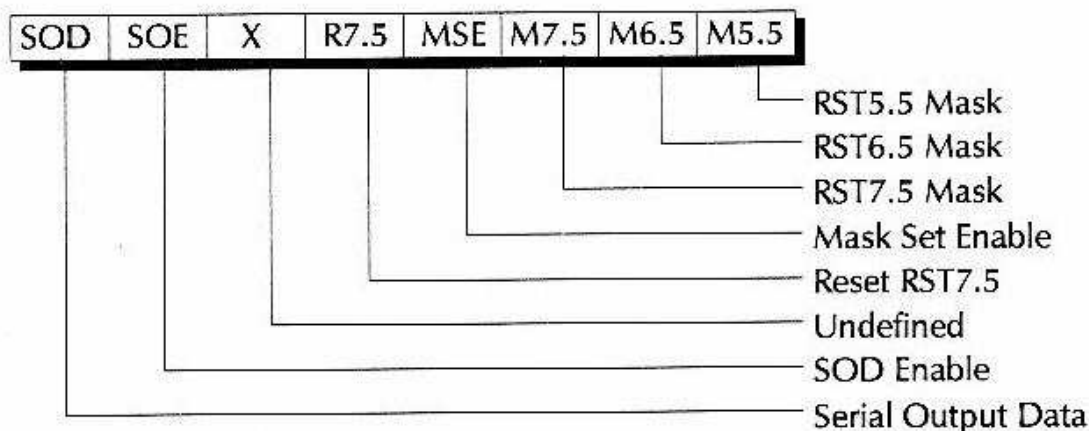
Αν η διακοπή αναγνωρίστηκε στην είσοδο INTR τότε ο επεξεργαστής εισέρχεται στον κύκλο μηχανής INA (INTerrupt Acknowledgement). Στη διάρκεια αυτού του κύκλου ενεργοποιείται το σήμα $\overline{\text{INTA}}$, και η περιφερειακή συσκευή, που προκάλεσε τη διακοπή, ανταποκρινόμενη στο σήμα αυτό τοποθετεί στον διάδρομο δεδομένων την εντολή CALL ή RST (n) όπως είδαμε και στην παράγραφο 6.2.2.

6.2.5 Ενεργοποίησης και Απενεργοποίησης του Μηχανισμού των Διακοπών. Παρεμπόδιση Διακοπών

Οι εντολές EI (Enable Interrupt) και DI (Disable Interrupt) ενεργοποιούν και απενεργοποιούν αντίστοιχα το μηχανισμό των διακοπών επηρεάζοντας το flip-flop INTE. Υπενθυμίζεται, ότι όταν ο 8085 ξεκινάει ή όταν προκαλείται διακοπή, ο μηχανισμός των διακοπών απενεργοποιείται

αυτόματα και συνεπώς απαιτείται η εκτέλεση της **EI**, για να λειτουργήσουν οι διακοπές.

Η επιλεκτική παρεμπόδιση των διακοπών, γίνεται με την εντολή **SIM** (Set Interrupt Mask). Η εντολή **SIM** χρησιμοποιεί το περιεχόμενο του συσσωρευτή, στον οποίον πρέπει προηγουμένως να έχει τοποθετηθεί η κατάλληλη τιμή. Το περιεχόμενο αυτό πρέπει να έχει τη μορφή που φαίνεται στο σχήμα 6.5.



Cycles: 1, States:4

Σχήμα 6.5 Το περιεχόμενο του καταχωρητή A για την εντολή **SIM**

Με τα bit 0 έως 2, αν αυτά λάβουν την τιμή 1, παρεμποδίζονται κάποιες ή και οι τρεις διακοπές RST 5.5 6.5 και 7.5 αντίστοιχα. Τα τρία αυτά bits αποτελούν και τη μάσκα διακοπών. Για να ενεργοποιηθεί όμως αυτή η μάσκα διακοπών όταν εκτελεστεί η **SIM**, πρέπει το bit 3 (**MSE**, Mask Set Enable) να έχει τιμή 1 διαφορετικά η **SIM** δεν έχει καμία επίδραση στις διακοπές αυτές. Για παράδειγμα οι παρακάτω εντολές επιτρέπουν τις διακοπές στις εισόδους RST 7.5 και RTST 5.5 ενώ παρεμποδίζονται οι διακοπές στην είσοδο RST 6.5.

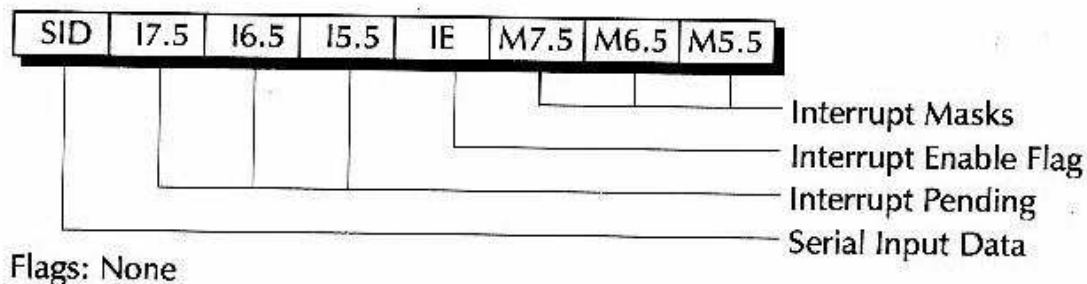
```
MVI  A,00001010b    ;Φόρτωση τη μάσκα διακοπών στο συσσωρευτή
SIM                                ;Εκτέλεσε την εντολή SIM
```

Με το bit 4, αν αυτό πάρει την τιμή 1, μπορούμε να ακυρώσουμε μια αίτηση διακοπής που εκκρεμεί στην είσοδο RST 7.5. Για να το επιτύχει αυτό η **SIM** θέτει το flip-flop που συνδέεται στην ακμοπυροδότητη είσοδο RST 7.5 σε κατάσταση RESET (λογικό "0"). Τέλος τα bit 6 και 7 δεν έχουν καμία σχέση με τις διακοπές γι' αυτό και δεν θα εξεταστούν εδώ.

Ο 8085 μας δίνει τη δυνατότητα να δούμε τη μάσκα διακοπών, που έχει τεθεί από κάποια εντολή **SIM**, καθώς επίσης και, εφόσον βέβαια είμαστε σε κατάσταση απενεργοποίησης των διακοπών, αν εκκρεμούν

αιτήσεις διακοπών στις εισόδους RST 7.5, 6.5 και 5.5. Όλα αυτά μπορούν να γίνουν με την εντολή RIM (Read Interrupt Mask).

Όταν εκτελεστεί η εντολή RIM, τοποθετεί τις παραπάνω πληροφορίες στον συσσωρευτή. Το περιεχόμενο του συσσωρευτή μετά την εκτέλεση της RIM φαίνεται στο σχήμα 6.6.



Σχήμα 6.6 Το περιεχόμενο του καταχωρητή A για την εντολή RIM

Τα bit 0 έως 2 περιέχουν τη μάσκα που έχει τεθεί στις τρεις παρεμποδιζόμενες εισόδους διακοπών. Εξετάζοντας το bit 3 μπορούμε να δούμε αν ο μηχανισμός των διακοπών είναι ενεργοποιημένος. Το bit αυτό περιέχει την τιμή του flip-flop INTE (INTerrupt Enable). Από τα bit 4 έως 6 μπορούμε να δούμε σε ποιές από τις εισόδους RST 7.5, 6.5, 5.5 εκκρεμούν διακοπές (αν το αντίστοιχο bit είναι 1). Τέλος το bit 7 δεν έχει σχέση με τις διακοπές, γι' αυτό και δεν εξετάζεται εδώ.

6.2.6 Μορφή υπορουτινών εξυπηρέτησης διακοπών για τον 8085

Όπως είδαμε στην παράγραφο 6.2.5, όταν αναγνωριστεί μια διακοπή, ο 8085 εκτελεί είτε μία εσωτερική RST, είτε την εντολή RST (n). Και στις δύο περιπτώσεις ο 8085 σώζει στη στοίβα τον PC, έτσι ώστε μετά το τέλος της ρουτίνας εξυπηρέτησης να μπορεί να συνεχίσει με την επόμενη εντολή από το σημείο που έγινε η διακοπή.

Η ρουτίνα εξυπηρέτησης θα πρέπει στο ξεκίνημά της να σώσει στη στοίβα την κατάσταση του με δηλαδή τους καταχωρητές συμπεριλαμβανομένου και του καταχωρητή σημαιών. Βέβαια για να μειωθεί η χρονική επιβάρυνση από την κλήση της υπορουτίνας, θα πρέπει να σωθούν μόνο οι καταχωρητές, τα περιεχόμενα των οποίων πρόκειται να τροποποιηθούν. Για παράδειγμα αν η υπορουτίνα εξυπηρέτησης χρησιμοποιεί τους καταχωρητές B, D και H, τότε πρέπει να αρχίσει με τις ακόλουθες εντολές:

```
PUSH PSW
PUSH B
PUSH D
PUSH H
```

Για να συνεχιστεί ομαλά η εκτέλεση του προγράμματος που είχε διακοπεί θα πρέπει να αποκατασταθεί η κατάσταση του μΕ, όπως ακριβώς ήταν πριν από την εκτέλεση της υπορουτίνας εξυπηρέτησης. Και φυσικά θα πρέπει να ενεργοποιηθεί ο μηχανισμός των διακοπών, αφού με την αναγνώριση της διακοπής απενεργοποιήθηκε αυτόματα από τον μΕ. Έτσι αν υπορουτίνα άρχισε με τις παραπάνω εντολές θα πρέπει να κλείσει με τις ακόλουθες:

```
POP  H
POP  D
POP  B
POP  PSW
EI
RET
```

Παρατηρείστε ότι στην παραπάνω ακολουθία εντολών η καταχωρητές ανακτώνται από τη στοίβα με την αντίστροφη σειρά.

Τέλος τα χαρακτηριστικά των εισόδων διακοπών του 8085 παρουσιάζονται συνοπτικά στον πίνακα 6.1.

Πίνακας 6.1 Τα χαρακτηριστικά διακοπών του 8085 (INTEL CORP.)

Συμβολισμός	Περιγραφή και λειτουργία
TRAP	Trap: Η διακοπή Trap είναι μια μη παρεμποδιζόμενη διακοπή. Αναγνωρίζεται την ίδια χρονική στιγμή όπως συμβαίνει και με τις INTR και RST 5.5_7.5. Δεν επηρεάζεται από καμιά μάσκα ή εντολή ενεργοποίησης διακοπής. Έχει την υψηλότερη προτεραιότητα απ' όλες τις διακοπές.
RST 7.5 RST 6.5 RST 5.5	Διακοπές επανεκκίνησης (Restart Interrupts): Αυτές οι τρεις εισοδοί διακοπών έχουν τον ίδιο χρονισμό με την INTR. Η προτεραιότητα αυτών των διακοπών είναι η ίδια με τη σειρά που αυτές αναγράφονται στην πρώτη στήλη αυτού του πίνακα. Αυτές οι διακοπές έχουν υψηλότερη προτεραιότητα από την INTR. Επιπλέον σε κάθε μια ξεχωριστά μπορεί να τεθεί μάσκα με την εντολή SIM.
INTR	Αίτηση διακοπής (Interrupt Request): Χρησιμοποιείται σαν είσοδος διακοπής γενικού σκοπού. Ενεργοποιείται μόνο μετά τον τελευταίο κύκλο μηχανής μιας διακοπής και κατά τη διάρκεια των καταστάσεων Hold και Halt. Όταν ενεργοποιηθεί ο PC δεν αυξάνεται αλλά εκπέμπεται το σήμα <u>INTA</u> . Στην συνέχεια εκτελείται μια εντολή CALL και ο έλεγχος μεταφέρεται στην ρουτίνα εξυπηρέτησης της διακοπής. Η διακοπή INTR μπορεί να ενεργοποιηθεί και να απενεργοποιηθεί με εντολές του Software. Απενεργοποιείται μόλις το σύστημα γίνει Reset και επίσης αμέσως από πρόκληση διακοπής.

INTA	Αναγνώριση Διακοπής (Interrupt Acknowledge): χρησιμοποιείται αντί του σήματος \overline{RD} κατά τη διάρκεια του κύκλου μηχανής, που ακολουθεί μετά την ενεργοποίηση της διακοπής INTR. Χρησιμοποιείται για να ενεργοποιήσει τον ελεγκτή διακοπών 8259A ή κάποια άλλη πόρτα διακοπών. Στη συνέχεια δίνονται δύο παραδείγματα όπου φαίνεται η χρήση και η αξία των διακοπών.
------	---

6.2.7 Εξυπηρέτηση πολλών περιφερειακών από μια είσοδο διακοπής με διαδικασία σάρωσης (polling)

Στην περίπτωση πολλών περιφερειακών συσκευών που πρέπει να εξυπηρετηθούν από μια είσοδο διακοπής μπορούμε να εφαρμόσουμε την παρακάτω τεχνική. Συνδέουμε τις γραμμές αίτησης εξυπηρέτησης όλων των συσκευών μέσω μιας πύλης OR στη διαθέσιμη είσοδο διακοπής. Όταν προκαλείται διακοπή από κάποια συσκευή η ρουτίνα εξυπηρέτησης διακοπής εξετάζει μέσω μιας θύρας εισόδου τα περιεχόμενα ενός καταχωρητή κατάστασης (STATUS), που δείχνει ποιά συσκευή προκάλεσε τη διακοπή. Η αναζήτηση αυτή γίνεται με τη διαδικασία της σάρωσης (Polling). Στη συνέχεια, όταν διαπιστωθεί ποιά συσκευή προκάλεσε τη διακοπή καλείται η αντίστοιχη ρουτίνα εξυπηρέτησης.

Στο σχήμα 6.7 φαίνεται το διάγραμμα ροής μιας διαδικασίας polling, η οποία έχει οκτώ συσκευές εισόδου με τις σημαίες διαθεσιμότητας δεδομένων συνδυασμένα σ'ένα byte κατάστασης (STATUS).

Το πρόγραμμα που ακολουθεί ελέγχει τη σημαία κατάστασης κάθε συσκευής εισόδου και στέλνει τον έλεγχο σε μία υπορουτίνα εξυπηρέτησης για την συσκευή της οποίας η σημαία έχει ενεργοποιηθεί. Θέτει επίσης προτεραιότητα για τις οκτώ συσκευές με τη σειρά με την οποία ελέγχει τα bits αίτησης εξυπηρέτησής τους. Η συσκευή εισόδου 7 έχει την υψηλότερη προτεραιότητα, δεδομένου ότι είναι η πρώτη που εξετάζεται.

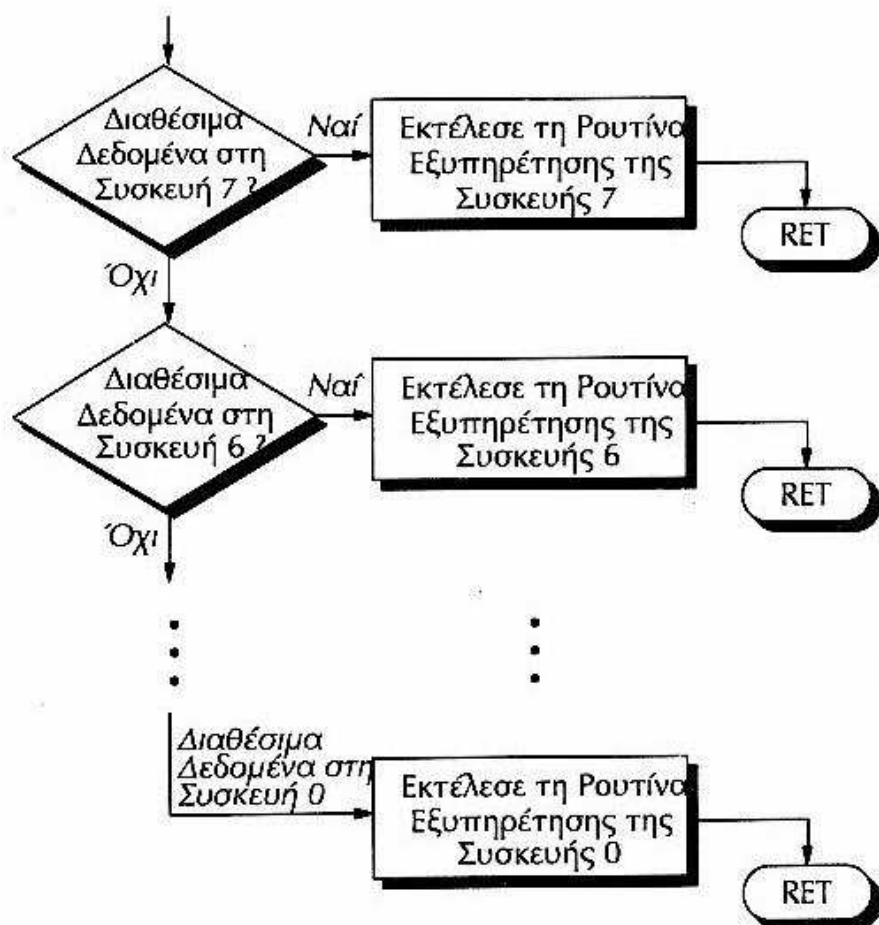
Ακολουθεί πρόγραμμα που κάνει συνεχή σάρωση (polling) 8 συσκευών:

IN	STATUS	;είσοδος κατάστασης
RLC		;απομόνωση 8ου bit
JNC	DEV6	
CALL	SRV_DEV7	;έλεγχος για εξυπηρέτηση συσκευής 7
RET		
DEV6:	RLC	;ομοίως για τη συσκευή 6
JNC	DEV5	
CALL	SRV_DEV6	
RET		
DEV5:	RLC	;ομοίως για τη συσκευή 5
JNC	DEV4	
CALL	SRV_DEV5	

```

    RET
DEV4:RLC                                ;ομοίως για τη συσκευή 4
    JNC DEV3
    CALL SRV_DEV4
    RET
DEV3:RLC                                ;ομοίως για τη συσκευή 3
    JNC DEV2
    CALL SRV_DEV3
    RET
DEV2:RLC                                ;ομοίως για τη συσκευή 2
    JNC DEV1
    CALL SRV_DEV2
    RET
DEV1:RLC                                ;ομοίως για τη συσκευή 1
    JNC DEV0
    CALL SRV_DEV1
    RET
DEV0:RLC                                ;ομοίως για τη συσκευή 0
    JNC EXIT
    CALL SRV_DEV0
EXIT:RET

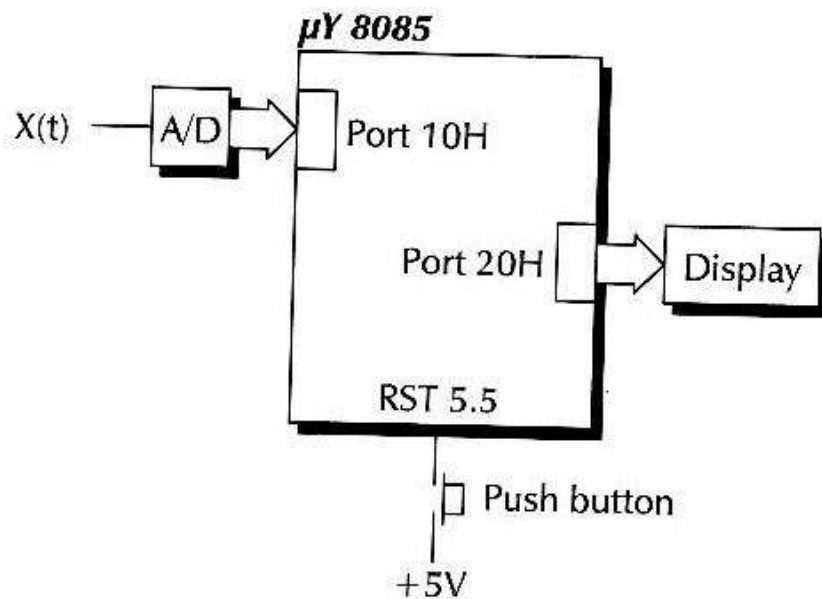
```



Σχήμα 6.7 Διάγραμμα ροής ρουτίνας εξυπηρέτησης 8 συσκευών.

Παράδειγμα 1ο

Δίνεται μY -Σ 8085 που διαθέτει πόρτα εισόδου (10H) με συνδεδεμένο έναν A/D (συνεχούς μετατροπής) και πόρτα εξόδου (20H) με συνδεδεμένα δυο 7-segments για απεικόνιση 8 bits δεδομένων. Ένα push button συνδέεται στο ένα άκρο του με την ακίδα RST5.5 και στο άλλο με το λογικό 1. Εστω τώρα ότι εκτελείται συνεχώς κάποιο κύριο πρόγραμμα και θέλουμε σε οποιαδήποτε επιθυμητή χρονική στιγμή πατώντας το push button να διακόπτεται το κύριο πρόγραμμα και να απεικονίζεται στο 7-segment display η τιμή του αναλογικού σήματος. Η απεικόνιση αυτή να διαρκεί έως ότου πατηθεί για δεύτερη φορά το push button οπότε και να έχουμε επιστροφή στο κύριο πρόγραμμα. Η διαδικασία αυτή να μπορεί να επαναλαμβάνεται όσες φορές επιθυμούμε. Να δοθεί η αναγκαία υπορουτίνα εξυπηρέτησης διακοπής. Το σύστημα που χρησιμοποιούμε φαίνεται στο σχήμα 6.8.



Σχήμα 6.8 Κύκλωμα απεικόνισης δεδομένων από A/D

Η ρουτίνα εξυπηρέτησης θα είναι :

```
002C:      JMP  ADC ή ADC*
002D: BYTE  1      ;Αρχικά στις 002D-002E υπάρχει η διεύθυνση
002E: BYTE  2      ;ADC, με κάθε πάτημα του διακόπτη στη συνέχεια
                  ;η διεύθυνση αυτή θα εναλλάσσεται με την ADC*.
```

;Τμήμα της ρουτίνας εξυπηρέτησης για το πρώτο πάτημα του διακόπτη
ADC:

```
PUSH PSW      ;Χρησιμεύει για το πρώτο πάτημα του
PUSH B         ;διακόπτη οπότε είναι αναγκαία η φύλαξη
```

PUSH D		;των καταχωρητών και του PSW.
PUSH H		
LXI H, ADC*		;Τοποθετείτε στη θέση μνήμης 002DH την
SHLD 002DH		;διεύθυνση ADC* ώστε στο επόμενο πάτημα
		;του διακόπτη να εκτελεσθεί η εντολή JMP ADC*.
EI		;Ενεργοποίηση διακοπών για να επιτραπεί το
		;δεύτερο πάτημα του διακόπτη
WAIT:		
IN 10H		;Διαβάζουμε τη τιμή του X(t) και την
OUT 20H		;απεικονίζουμε στη θύρα 20H
JMP WAIT		;Η απεικόνιση επαναλαμβάνεται μέχρι να
		;ξαναπατηθεί για δεύτερη φορά ο διακόπτης

;Τμήμα της ρουτίνας εξυπηρέτησης για το δεύτερο πάτημα του διακόπτη
ADC* :

LXI H, ADC		;Προετοιμασία για την επόμενη διακοπή που
SHLD 002DH		;θα είναι περιττή
POP H	} Δείτε και τον ποζι 2 POP H	;Αποκατάσταση των καταχωρητών
POP H		
POP D		
POP B		
POP PSW		
RET		;Επιστροφή στο κυρίως πρόγραμμα

Όπως φαίνεται παραπάνω, η ρουτίνα εξυπηρέτησης της διακοπής RST5.5, αρχίζει από την διεύθυνση 002CH και αποτελείται από δύο τμήματα. Το πρώτο τμήμα, το οποίο εκτελείται στο πρώτο πάτημα του διακόπτη και αρχίζει από τη διεύθυνση ADC. Το τμήμα αυτό μπαίνει σε ένα βρόγχο και απεικονίζει συνεχώς την τιμή του αναλογικού σήματος μέχρι να συμβεί διακοπή από το δεύτερο πάτημα του διακόπτη, οπότε εκτελείται το δεύτερο τμήμα, που εξασφαλίζει την ομαλή επιστροφή στο κύριο πρόγραμμα. Το τμήμα αυτό αρχίζει από τη διεύθυνση ADC*.

Η μετάβαση στο κατάλληλο τμήμα κάθε φορά που συμβαίνει διακοπή, επιτυγχάνεται με την εντολή JMP που υπάρχει στην αρχή της ρουτίνας εξυπηρέτησης. Η εντολή JMP αποτελείται από 3 bytes. Η διεύθυνση στην οποία θα γίνει το άλμα περιέχεται στα 2 τελευταία bytes και στις διευθύνσεις 002DH και 002EH. Στην αρχή η εντολή JMP δείχνει στο πρώτο τμήμα. Έτσι με το πρώτο πάτημα του διακόπτη εκτελείται το τμήμα ADC. Το τμήμα αυτό τροποποιεί την εντολή JMP, καταχωρώντας τη διεύθυνση ADC* στη θέση μνήμης 002DH και 002EH, έτσι ώστε στο δεύτερο πάτημα του διακόπτη να εκτελεστεί το τμήμα ADC*. Το τμήμα

αυτό με τη σειρά του τροποποιεί την εντολή **JMP**, έτσι ώστε να δείχνει πάλι στο τμήμα **ADC** και έτσι η όλη διαδικασία να μπορεί να επαναληφθεί.

Το τμήμα της ρουτίνας **ADC*** εξασφαλίζει την ομαλή επιστροφή στο κύριο πρόγραμμα αποκαθιστώντας τους καταχωρητές με τις κατάλληλες εντολές **POP**. Υπάρχει όμως το ακόλουθο πρόβλημα. Όπως αναφέρθηκε και προηγουμένως, το τμήμα **ADC*** ενεργοποιείται με το δεύτερο πάτημα του διακόπτη διακόπτοντας την εκτέλεση του βρόγχου **WAIT**, που εκτελείται εκείνη τη στιγμή από το τμήμα **ADC**. Η διακοπή αυτή έχει σαν αποτέλεσμα να τοποθετηθεί στη στοίβα η τιμή του **PC**. Η τιμή αυτή πρέπει να απομακρυνθεί προκειμένου να ακολουθήσει η ομαλή αποκατάσταση των καταχωρητών. Γι' αυτό ακριβώς το λόγο η εντολή **POP** Η εκτελείται δύο φορές. Την πρώτη φορά απομακρύνει την τιμή του **PC** από τη στοίβα ενώ η δεύτερη αποκαθιστά τον καταχωρητή **H-L**.

Παράδειγμα 2ο

Στο παρακάτω πρόγραμμα πραγματοποιείται η μέτρηση του αριθμού των διακοπών τύπου **RST5.5** στον μY 8085. Σαν μετρητής χρησιμοποιείται ο καταχωρητής **B**. Η εκτέλεση του προγράμματος αρχίζει από τη διεύθυνση **START**. Ζητούνται τα παρακάτω: α) Επιτρέπει το πρόγραμμα την παραπάνω διακοπή και γιατί; β) Πόσες διακοπές μπορεί να αντέξει το πρόγραμμα χωρίς να αυτοκαταστραφεί; Δικαιολογείστε. γ) Να προταθούν λύσεις που θεραπεύουν τυχόν αδυναμίες του προγράμματος. δ) Να δοθεί το πρόγραμμα σε τυπικά ορθή μορφή.

```
RST5.5:
    JMP  LOOP1
    .
    .
START:
    LXI  SP, LOOP2+5
    MVI  B, FFH
LOOP1:
    INR  B
    MVI  A, 0EH
    SIM
    EI
LOOP2:
    JMP  LOOP2
```

α) Το πρόγραμμα επιτρέπει την διακοπή **RST5.5** γιατί πριν από την εντολή **SIM** έχει φορτωθεί στον **A** η τιμή 00001110 όπου $\text{bit}3=1$, που σημαίνει

επίτρεψη των διακοπών που ορίζουν τα bits 0-2 και bit0=0 που σημαίνει επίτρεψη της διακοπής RST5.5.

β) Φαινομενικά το πρόγραμμα αντέχει μόνο μία διακοπή και αυτό γιατί με τη δεύτερη, η αποθήκευση του PC στη στοίβα θα καταστρέψει το δεύτερο και τρίτο byte της εντολής `JMP LOOP2`. Επειδή όμως οι διακοπές συμβαίνουν όταν εκτελείται ο βρόγχος `LOOP2`, η τιμή του PC που αποθηκεύεται στη στοίβα, είναι η διεύθυνση `LOOP2`. Έτσι με τη δεύτερη διακοπή η στοίβα θα περιέχει τα ακόλουθα:

```

LOOP2:JMP          ;Κώδικας εντολής JMP
LOOP2+1:(LOW BYTE) ;Το λιγότερο σημαντικό byte της διεύθυνσης LOOP2
LOOP2+2:(HIGH BYTE);Το περισσότερο σημαντικό byte της διεύθυνσης LOOP2
LOOP2+3:PCL
LOOP2+4:PCH
SP=LOOP2+5

```

Βλέπουμε δηλαδή, ότι στο δεύτερο και τρίτο byte εντολής `JMP` θα αποθηκευτεί η διεύθυνση `LOOP2`, η οποία όμως προϋπήρχε σ' αυτή τη θέση. Έτσι και η δεύτερη διακοπή θα εξελιχθεί ομαλά. Το πρόβλημα θα εμφανισθεί στην τρίτη διακοπή.

γ) Το πρόβλημα που παρουσιάζει το παραπάνω πρόγραμμα είναι, ότι η στοίβα μεγαλώνει συνεχώς. Και αυτό συμβαίνει γιατί όταν προκαλείται μία διακοπή και τοποθετείται η τιμή του PC στη στοίβα δεν εκτελείται στο τέλος η εντολή `RET` που θα επαναφέρει τη στοίβα στην αρχική της μορφή. Μια λύση στο πρόβλημα είναι να αυξάνει ο SP κατά δύο σε κάθε μέτρηση-διακοπή.

```

LOOP1: INR    B
        INX    SP
        INX    SP
        MVI    A, 0EH
        . . .

```

δ) Η τυπικά ορθή μορφή στην οποία θα μπορούσε να γραφεί το πρόγραμμα είναι η ακόλουθη:

```

RST5.5:JMP    LOOP1
        . . .
;Κύριο πρόγραμμα
START:MVI     B, 0H
LOOP2:JMP     LOOP2

```

```
;Ρουτίνα εξυπηρέτησης της διακοπής
```

```
LOOP1:
```

```
    INR    B
```

```
    MVI    A, 0EH
```

```
    SIM
```

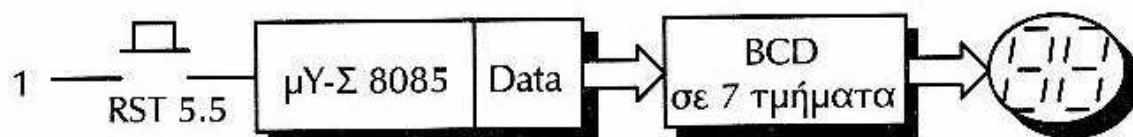
```
    EI
```

```
    RET
```

Το πρόγραμμα αποτελείται από το κύριο πρόγραμμα, που αρχικοποιεί το μετρητή και στη συνέχεια μπαίνει σε βρόγχο αναμονής, και τη ρουτίνα εξυπηρέτησης, η οποία, όταν καλείται, αυξάνει απλώς το μετρητή. Το πρόβλημα με τη συνεχή αύξηση του μεγέθους της στοίβας δεν υπάρχει πλέον, αφού η τιμή του PC, που τοποθετείται στη στοίβα όταν συμβαίνει διακοπή, απομακρύνεται από αυτήν με την εντολή **RET**.

Παράδειγμα 3ο

Να υλοποιηθεί σε $\mu\text{Y-}\Sigma$ 8085 χρονόμετρο δευτερολέπτων. Το χρονόμετρο θα λειτουργεί με ένα πλήκτρο που με κάθε περιττό πάτημά του ο χρόνος θα ξεκινάει από το μηδέν ενώ με κάθε άρτιο πάτημα ο χρόνος θα σταματάει και θα εμφανίζεται σταθερά στο display. Το πλήκτρο αυτό να συνδεθεί στην είσοδο διακοπής RST 5.5. Δίνεται υπορουτίνα DELAY καθυστέρησης 50msec.



Το πρόγραμμα για την υλοποίηση του χρονομέτρου δίνεται με δύο τρόπους. Στον πρώτο τρόπο η λειτουργία του χρονομέτρου υλοποιείται από το κύριο πρόγραμμα ενώ στο δεύτερο τρόπο η λειτουργία πραγματοποιείται αποκλειστικά από την υπορουτίνα εξυπηρέτησης.

1ος Τρόπος

Στο πρόγραμμα που ακολουθεί η θέση μνήμης 3000H χρησιμοποιείται για την αποθήκευση της τιμής του μετρητή με τη μορφή δύο BCD ψηφίων ενώ η 3001H χρησιμοποιείται σαν σημαία που με την τιμή 1 δείχνει ότι μέχρι εκείνη τη στιγμή έχει συμβεί περιττός αριθμός διακοπών ενώ με την τιμή 0 ο αριθμός αυτός είναι άρτιος. Η ρουτίνα εξυπηρέτησης της διακοπής RST 5.5 αντιστρέφει απλώς τη σημαία στη θέση μνήμης 3001h. Έτσι όταν η σημαία είναι 0 (άρτιος αριθμός διακοπών) η επόμενη διακοπή που θα συμβεί θα είναι άρτια και η σημαία θα πάρει την τιμή 1. Επίσης όταν η διακοπή είναι περιττή μηδενίζεται ο μετρητής ώστε το χρονόμετρο να ξεκινάει από την αρχή.

```

MVI A, 0E      ;Αρχικοποίηση μάσκας διακοπών, σημαίας και μετρητή.
SIM
MVI A, 00
STA 3000      ;Αρχικοποίηση μετρητή
MVI A, 00
STA 3001      ;Αρχικοποίηση σημαίας
EI

```

LOOP:

```

CALL DISPLAY
LDA 3001      ;Αν ο αριθμός των μέχρι τώρα διακοπών είναι
CPI 0         ;άρτιος συνέχισε να τυπώνεις το περιεχόμενο
JZ LOOP       ;του μετρητή στο Display χωρίς να τον αυξάνεις.
CALL DELAY1   ;Καθυστέρηση ένα sec
CALL INC_COUNTER
JMP LOOP

```

;Η ρουτίνα Display εμφανίζει την τιμή του μετρητή

DISPLAY:

```

LDA 3000H
OUT DATA
RET

```

;Η υπορουτίνα DELAY1 δημιουργεί καθυστέρηση 1 sec

DELAY1:

```

MVI B, 20d

```

INL:

```

CALL DELAY
DCR B
JNZ INL
RET

```

;Η ρουτίνα INC_COUNTER αυξάνει το μετρητή του χρονομέτρου κατά 1. Επίσης αν ο μετρητής έχει την τιμή 60 η ρουτίνα αυτή τον μηδενίζει.

INC_COUNTER:

```

LDA 3000H
INR A
DAA
STA 3000H
CPI 60H      ;Αν ο μετρητής φτάσει την τιμή 60 (περιέχει
JNZ OUTL     ;δηλαδή τα BCD ψηφία 6 και 0, γι' αυτό η τιμή 60
XRA A        ;κατά τη σύγκριση είναι HEX) μηδενίζεται.

```

OUTL:

```

STA 3000H
RET

```

Στη συνέχεια ακολουθεί η υπορουτίνα εξυπηρέτησης για τη διακοπή RST 5.5.

RST 5.5:

```

PUSH PSW

```

LDA	3001H	;Αντιστρέφεται η σημαία που δείχνει αν έχουμε
XRI	01	;άρτιο ή περιττό αριθμό διακοπών
.STA	3001H	
CPI	01	;Εξετάζεται η σημαία διακοπών.
JNZ	END	;Αν ο αριθμός των διακοπών είναι άρτιος επιστρέφει
XRA	A	;Διαφορετικά μηδενίζεται ο μετρητής και επιστρέφει
STA	3000H	
END: POP	PSW	
EI		
RET		

2ος Τρόπος

Στο δεύτερο τρόπο την υπολογιστική εργασία για τη λειτουργία του χρονομέτρου αναλαμβάνει αποκλειστικά η υπορουτίνα εξυπηρέτησης. Το κύριο πρόγραμμα που ακολουθεί εκτελείται μόνο κατά την εκκίνηση του προγράμματος μέχρι να συμβεί η πρώτη διακοπή.

;Κύριο πρόγραμμα

MVI	A, 0EH	;Αρχικοποίηση διακοπών
SIM		
MVI	C, 0	;Αρχικοποίηση μετρητή και σημαίας
MVI	A, 0	
EI		
LOOP:		
JMP	LOOP	;Αναμονή διακοπών

Ακολουθεί η υπορουτίνα εξυπηρέτησης. Η υπορουτίνα αυτή περιλαμβάνει δύο τμήματα. Το τμήμα ODD το οποίο κάνει τη χρονομέτρηση, και το οποίο εκτελείται όταν συμβεί περιττή διακοπή και το τμήμα EVEN για τις άρτιες διακοπές, το οποίο τυπώνει συνεχώς την τιμή του χρονομέτρου στο DISPLAY. Ο καταχωρητής C χρησιμοποιείται σαν σημαία, η οποία όταν έχει την τιμή 1 δείχνει ότι ο αριθμός των διακοπών, που έχει συμβεί μέχρι τώρα είναι περιττός. Η σημαία αυτή εξετάζεται στην αρχή της υπορουτίνας εξυπηρέτησης για να αποφασιστεί ποιο από τα δύο τμήματα της υπορουτίνας θα εκτελεστεί. Ο καταχωρητής A χρησιμοποιείται σαν μετρητής δύο BCD ψηφίων.

RST	5.5:	
POP	H	;Απομάκρυνση από τη στοίβα του περιεχομένου
		;του PC που αποθηκεύτηκε με τη διακοπή
MOV	D, A	;Προσωρινή αποθήκευση της τιμής του μετρητή
MOV	A, C	
XRI	01	;Αντιστροφή της τιμής της σημαίας
MOV	C, A	
CPI	1	

```

JNZ  EVEN      ;Αν ο καταχωρητής C είναι 0, ο αριθμός των διακοπών
                ;είναι άρτιος και εκτελείται το τμήμα EVEN.
ODD:
XRA  A          ;Μηδενισμός του μετρητή
EI                    ;Επίτρεψη διακοπών
LOOP1:
CALL DISPLAY
CALL DELAY
CALL INC_COUNTER
JMP  LOOP1
EVEN:
MOV  A,D        ;Ανάκτηση της τιμής του μετρητή
EI
LOOP2:
CALL DISPLAY
JMP  LOOP2

```

Η ρουτίνες DELAY, INC_COUNTER και DISPLAY, έχουν δοθεί κατά την περιγραφή του πρώτου τρόπου.

Εξετάζοντας την υπορουτίνα εξυπηρέτησης βλέπουμε ότι δεν υπάρχει εντολή RET, και αυτό γιατί όλη η λειτουργία του χρονομέτρου υλοποιείται αποκλειστικά από αυτήν χωρίς τη συμμετοχή του κύριου προγράμματος. Όταν συμβεί μια νέα διακοπή η υπορουτίνα εξυπηρέτησης διακόπτεται και ξαναεκτελείται από την αρχή. Αυτό έχει σαν αποτέλεσμα με κάθε διακοπή να τοποθετείται στη στοίβα ο Program Counter χωρίς να απομακρύνεται ποτέ από αυτήν αφού δεν εκτελείται ποτέ η εντολή RET. Για το λόγο αυτό εκτελείται η εντολή POP H στην αρχή της υπορουτίνας, η οποία απομακρύνει το PC από τη στοίβα και αποφεύγεται η υπερχειλίσιση της.

Παράδειγμα 4ο

Μια συσκευή εισόδου είναι συνδεδεμένη στη θύρα 80H ενός μΥ συστήματος 8085. Πριν από κάθε δεδομένο, που πρόκειται να στείλει η συσκευή αυτή, προκαλεί διακοπή τύπου RTST 5.5. Να γραφεί πρόγραμμα που να εισάγει 64 δεδομένα από τη συσκευή και στη συνέχεια να σταματά την εισαγωγή δεδομένων και να υπολογίζει το μέσο όρο. Τα δεδομένα είναι θετικοί αριθμοί των 8 bits.

```

;Κύριο πρόγραμμα
MVI  A, 0EH      ;Αρχικοποίηση της μάσκας διακοπών και του
SIM                    ;μετρητή δεδομένων C
LXI  H, 0        ;Μηδενισμός συσσωρευτή
MVI  C, 64d
EI                    ;Ενεργοποίηση διακοπών

```

ADDR:

MOV	A, C	;Βρόγχος αναμονής μέχρι να διαβαστούν όλα τα
CPI	0	;δεδομένα, οπότε ο μετρητής C θα έχει την τιμή 0
JNZ	ADDR	
DI		;Οι διακοπές απενεργοποιούνται αφού έχουν
		;διαβαστεί και τα 64 δεδομένα
DAD	H	;Υπολογισμός του μέσου όρου με ολίσθηση 2
DAD	H	;θέσεων προς τα αριστερά του ζεύγους H-L. Αυτό
		;επιτυγχάνεται με την πρόσθεση του ζεύγους H-L
HLT		;δύο φορές στον εαυτό του.

;Ρουτίνα εξυπηρέτησης

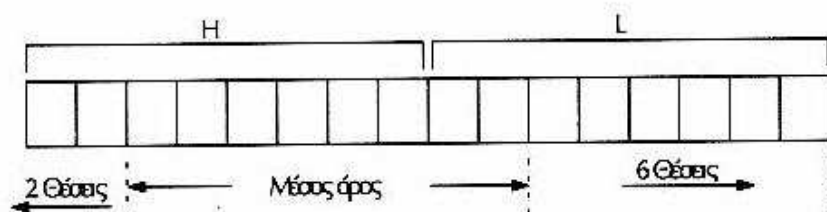
002C:

PUSH	PSW	
IN	80h	
MVI	D, 0	
MOV	E, A	
DAD	D	;Πρόσθεση δεδομένων
DCR	C	;Ελάττωση μετρητή
POP	PSW	
EI		
RET		

Στο παραπάνω πρόγραμμα, η υπορουτίνα εξυπηρέτησης της διακοπής διαβάσει το δεδομένο και το προσθέτει στον αθροιστή ενώ το κύριο πρόγραμμα, όταν ολοκληρωθεί η ανάγνωση και των 64 δεδομένων υπολογίζει το μέσο όρο. Σαν αθροιστής χρησιμοποιείται το ζεύγος καταχωρητών H-L. Ο καταχωρητής C χρησιμοποιείται σαν μετρητής για τα 64 δεδομένα. Κάθε δεδομένο που διαβάζεται από τη ρουτίνα εξυπηρέτησης τοποθετείται σαν αριθμός δύο byte (με το πιο σημαντικό byte μηδενισμένο) στο ζεύγος καταχωρητών D-E και στη συνέχεια προστίθεται στο ζεύγος H-L.

Για τον υπολογισμό του μέσου όρου απαιτείται η διαίρεση του διπλού καταχωρητή H-L με τον αριθμό $64=2^6$.

Επειδή ο διαιρέτης είναι δύναμη του 2, το πηλίκο μπορεί να υπολογισθεί με δεξιά ολίσθηση κατά 6 θέσεις. Όπως φαίνεται στο παρακάτω σχήμα, το ίδιο μπορεί να επιτευχθεί με αριστερή ολίσθηση κατά 2 θέσεις μόνο που ο μέσος όρος αντί να βρεθεί στον καταχωρητή L μεταφέρεται στον καταχωρητή H.



Μετά την ολίσθηση ο καταχωρητής H θα περιέχει το ακέραιο μέρος του αποτελέσματος ενώ ο L το κλασματικό μέρος εκφρασμένο σαν πολλαπλάσιο το $1/256$. Για παράδειγμα, έστω ότι μετά την πρόσθεση το ζεύγος H-L περιέχει την τιμή 258. (0000 0001 -0000 0010). Μετά την ολίσθηση θα περιέχει την τιμή 0000 0100 - 0000 1000 (H=4 και L=8). Δηλαδή το ακέραιο μέρος είναι 4 και το κλασματικό $8 \times 1/256 = 0,03125$. Άρα ο μέσος όρος είναι 4,03125.

Παράδειγμα 5ο

Ένα μΥ-Σ 8085 δέχεται δεδομένα από τη θύρα εισόδου DATA. Κάθε νέο δεδομένο προκαλεί διακοπή RST 5.5. Τα δεδομένα αγνοούνται μέχρι να εμφανιστεί η τιμή 01. Η τιμή αυτή και τα δεδομένα που ακολουθούν μέχρι την τιμή 02, καταχωρούνται σε έναν πίνακα που αρχίζει από τη διεύθυνση DTXT. Να δοθεί το πρόγραμμα.

1ος τρόπος

;Κύριο πρόγραμμα

LXI H, DTXT

;Φόρτωσε το ζεύγος καταχωρητών H-L με την
;διεύθυνση του πίνακα.

MVI A, 0Eh

;Αρχικοποίησε τη μάσκα διακοπών και τις σημαίες

SIM

;και ενεργοποίησε τις διακοπές.

MVI C, 0

;Σημαία έγκυρου δεδομένου

MVI B, 0

;Σημαία παρουσίας νέου δεδομένου

EI

LOOP:

MOV A, B

;Βρόγχος αναμονής

CPI 01

JNZ LOOP

;Αν B=0, δεν υπάρχει δεδομένο για αποθήκευση
;και η αναμονή συνεχίζεται

MOV A, D

MOV M, A

;Αποθήκευσε το δεδομένο και αύξησε τη

INX H

;διεύθυνση κατά 1, έτσι ώστε να δείχνει στο

MVI B, 0

;επόμενο στοιχείο του πίνακα.

CPI 02

;Αν το δεδομένο ήταν ο αριθμός 2, η διαδικασία

JNZ LOOP

;τερματίζεται

DI

HLT

;Ρουτίνα εξυπηρέτησης

002C:

PUSH PSW

IN	DATA	
MOV	D, A	
CPI	01	; Αν το δεδομένο που διαβάστηκε είναι 1 τότε δώσε την
JZ	SET	;τιμή 1 στους καταχωρητές B και C
MOV	A, C	; Αλλιώς αν και ο καταχωρητής C
CPI	01	;δεν έχει την τιμή 1 τότε δεν έχουμε έγκυρο δεδομένο
JNZ	EXIT	;Διαφορετικά (για τιμή καταχ. C ίσο με 1)
SET: MVI	B, 01	; τίθεται 1 στον καταχωρητή B για να πληροφορήσουμε
MVI	C, 01	; το κύριο πρόγραμμα ότι διαβάστηκε δεδομένο προς
EXIT: POP	PSW	; αποθήκευση
EI		;Επιστροφή στο κύριο πρόγραμμα
RET		

Η υπορουτίνα εξυπηρέτησης διαβάζει κάθε δεδομένο που στέλνεται στον μΥ-Σ, ενώ το κύριο πρόγραμμα δέχεται αυτό το δεδομένο μέσω του καταχωρητή D και το αποθηκεύει στον πίνακα. Ο καταχωρητής C χρησιμοποιείται σαν σημαία, η οποία δείχνει πότε τα δεδομένα που διαβάζει η ρουτίνα εξυπηρέτησης πρέπει να αποθηκευθούν. Η σημαία αυτή ενεργοποιείται (λαμβάνει την τιμή 1) και παραμένει όταν διαβαστεί δεδομένο με τιμή 01. Μέσω του καταχωρητή B, η ρουτίνα εξυπηρέτησης πληροφορεί το κύριο πρόγραμμα, ότι ενεργοποιήθηκε και διάβασε κάποιο δεδομένο που χρειάζεται αποθήκευση. Έτσι το κύριο πρόγραμμα εκτελείται ένας βρόγχος αναμονής, έως ότου ο καταχωρητής B γίνει 1. Τότε το κύριο πρόγραμμα λαμβάνει από την υπορουτίνα το δεδομένο μέσω του καταχωρητή D, το αποθηκεύει στον πίνακα, μηδενίζει τον καταχωρητή B και ξαναμπαίνει στο βρόγχο αναμονής περιμένοντας νέο δεδομένο. Η διαδικασία τερματίζεται όταν το κύριο πρόγραμμα λάβει από την ρουτίνα εξυπηρέτησης δεδομένο με τιμή 02.

2ος τρόπος

;Κύριο πρόγραμμα		
LXI	H, DTXT	;Φόρτωσε τους καταχ. H-L με την διεύθυνση του πίνακα
MVI	A, 0EH	;Αρχικοποίησε τη μάσκα διακοπών
SIM		;και ενεργοποίησε τις διακοπές.
EI		
MVI	C, 0	;Αρχικοποίησε τη σημαία έγκυρου δεδομένου
WAIT:		
JMP	WAIT	;βρόγχος αναμονής
;Ρουτίνα εξυπηρέτησης		
002C:		
IN	DATA	;Διάβασε το δεδομένο
MOV	D, A	;Φύλαξέ το στον καταχωρητή D
MOV	A, C	

```

CPI 01      ;Αν η σημαία είναι 1 σημαίνει ότι είμαστε μετά το 01
JZ  STORE   ;Αποθήκευσε το δεδομένο.
MOV \ A, D   ;Επανάκτηση δεδομένου
CPI 01      ;Αν η σημαία είναι 0 εξέτασε αν το δεδομένο είναι 01
JNZ EXIT     ;Αν όχι αγνόησέ το, διαφορετικά θέσε τη σημαία
MVI C, 1     ;αποδοχής 1 και αποθήκευσε το δεδομένο

STORE:
MOV A, D     ;Επανάκτηση δεδομένου
MOV M, A     ;Αποθήκευση δεδομένου
INX H        ;Αύξησε το δείκτη του πίνακα
CPI 02      ;Αν το δεδομένο είναι το 02 τερματίζεται το
JNZ EXIT     ;πρόγραμμα
DI
EXIT:HLT
EI
RET

```

Με το δεύτερο τρόπο βλέπουμε ότι απλουστεύεται κατά πολύ η διαδικασία, αφού τώρα ο έλεγχος και η αποθήκευση των δεδομένων γίνεται αποκλειστικά από τη ρουτίνα εξυπηρέτησης. Το κύριο πρόγραμμα κάνει απλώς της αναγκαίες αρχικοποιήσεις και στη συνέχεια μπαίνει σε βρόγχο αναμονής. Ο τερματισμός του προγράμματος γίνεται από τη ρουτίνα εξυπηρέτησης με την εντολή **HLT**.

Ο καταχωρητής C χρησιμοποιείται από τη ρουτίνα εξυπηρέτησης ως σημαία αποδοχής δεδομένου, δηλαδή δείχνει αν έχει ήδη διαβαστεί το 01, οπότε το δεδομένο αποθηκεύεται στον πίνακα.

Παράδειγμα 6ο

Για την υλοποίηση χρονομέτρου λεπτών και δευτερολέπτων με τη χρήση µΥ 8085, μια γεννήτρια παλμών συχνότητας 50 Hz συνδέεται στην είσοδο διακοπών RST 5.5 του µΕ. Επίσης δύο ενδείκτες 7 τμημάτων (7-Segment Display) συνδέονται στις θύρες 20H και 21H για την απεικόνιση των λεπτών και των δευτερολέπτων αντίστοιχα. Να δοθεί η υπορουτίνα εξυπηρέτησης και η εντολές αρχικοποίησης του κύριου προγράμματος. Η ρουτίνα εξυπηρέτησης να γραφεί έτσι ώστε το χρονόμετρο να λειτουργεί παράλληλα με το κύριο πρόγραμμα.

```

PULSES      DB (?)
SEC          DB (?)
MIN          DB (?)

;Κύριο πρόγραμμα. Αρχικοποίηση χρονομέτρου
XRA A       ;Μηδένισε τους μετρητές παλμών, λεπτών
STA PULSES  ;και δευτερολέπτων
STA SEC

```

```

STA MIN
MVI A, 0EH           ;Επίτρεψε τη διακοπή RST 5.5
SIM
EI

```

```

...
;Υπόλοιπες εντολές του κύριου προγράμματος
...

```

;Ρουτίνα εξυπηρέτησης

RST 5.5:

```

    PUSH PSW

```

INR_SEC:

```

    CALL DISPLAY           ;Απεικόνισε τα λεπτά και τα δευτερόλεπτα

```

```

    LDA PULSES

```

```

    INR A

```

```

    STA PULSES

```

```

    CPI 50D

```

;Αν δεν έχει φτάσει στην τιμή 50, η ρουτίνα

```

    JNZ EXIT

```

;τερματίζει. Διαφορετικά μηδενίζεται και

```

    XRA A

```

;ακολουθως αυξάνεται ο μετρητής των

```

    STA PULSES

```

;δευτερολέπτων

```

    LDA SEC

```

```

    INR A

```

;Αύξησε το μετρητή των δευτερολέπτων

```

    DAA

```

;κανονικοποίησε σε BCD μορφή

```

    STA SEC

```

```

    CPI 60H

```

;Αν δεν έχει φτάσει στην τιμή 60H,

```

    JNZ EXIT

```

;είναι αποθηκευμένος σε BCD μορφή) η

```

    XRA A

```

;ρουτίνα τερματίζει. Διαφορετικά μηδενίζεται και

```

    STA SEC

```

;ακολουθως αυξάνεται ο μετρητής των λεπτών

```

    LDA MIN

```

```

    INR A

```

;Αύξησε το μετρητή των λεπτών

```

    DAA

```

;κανονικοποίησε σε BCD μορφή

```

    STA MIN

```

```

    CPI A, 60h

```

;Αν δεν έχει φτάσει στην τιμή 60H, η

```

    JNZ EXIT

```

;ρουτίνα τερματίζει. Διαφορετικά μηδενίζεται

```

    XRA A

```

```

    STA MIN

```

EXIT:

```

    POP PSW

```

```

    EI

```

```

    RET

```

;Με την ακόλουθη υπορουτίνα εμφανίζονται στους ενδείκτες τα λεπτά και
;τα δευτερόλεπτα

DISPLAY:

```

LDA  MIN
OUT  20H
LDA  SEC
OUT  21H
RET

```

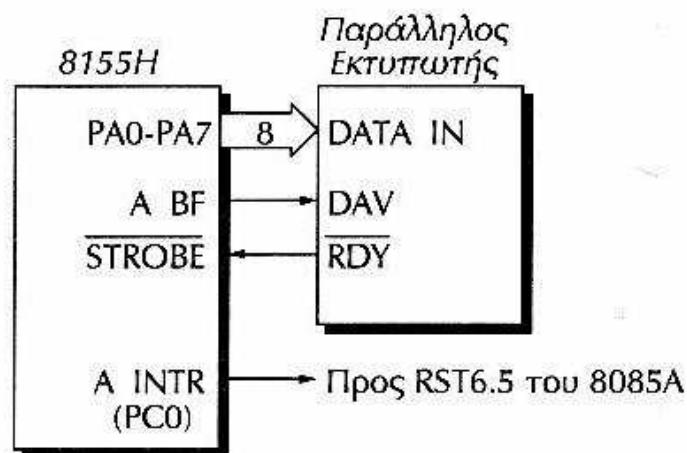
Για να μπορεί το χρονόμετρο να λειτουργεί παράλληλα με τη διαδικασία που εκτελεί το κύριο πρόγραμμα, η υλοποίηση πραγματοποιείται αποκλειστικά από τη ρουτίνα εξυπηρέτησης της διακοπής. Το κύριο πρόγραμμα μηδενίζει μόνο τους μετρητές κατά την εκκίνηση. Για τον ίδιο λόγο οι μετρητές αυτοί φυλάγονται σε θέσεις μνήμης και όχι σε καταχωρητές. Οι θέσεις μνήμης που χρησιμοποιούνται σαν μετρητές στο παραπάνω πρόγραμμα είναι, PULSES για τους παλμούς της γεννήτριας, SEC για τα δευτερόλεπτα και MIN για τα λεπτά.

Κάθε φορά που παράγεται ένας παλμός από τη γεννήτρια, προκαλείται διακοπή και εκτελείται η ρουτίνα εξυπηρέτησης, η οποία απεικονίζει το χρόνο και αυξάνει το μετρητή των παλμών. Αν αυτός λάβει την τιμή 50, αυτό σημαίνει ότι πέρασε ένα δευτερόλεπτο, αφού η γεννήτρια λειτουργεί με συχνότητα 50 Hz. Στην περίπτωση αυτή μηδενίζεται ο μετρητής και αυξάνει κατά 1 ο μετρητής των δευτερολέπτων. Με τη σειρά του, αν και αυτός ο μετρητής λάβει την τιμή 60, μηδενίζεται και αυξάνει ο μετρητής των λεπτών. Όταν λάβει και αυτός την τιμή 60, μηδενίζεται και το χρονόμετρο ξεκινάει από την αρχή.

Παράδειγμα 7ο

Σε μΥ-8085 είναι παράλληλα συνδεδεμένος εκτυπωτής μέσω του ολοκληρωμένου 8155 όπως φαίνεται στο παρακάτω σχήμα. Ο εκτυπωτής πληροφορεί τον 8155 ότι είναι έτοιμος να δεχτεί έναν χαρακτήρα στέλνοντας έναν παλμό μέσω της ακίδας \overline{RDY} . Ο 8155 με τη σειρά του προκαλεί διακοπή 6.5 στον 8085 μέσω της ακίδας A INTR, οπότε ενεργοποιείται η αντίστοιχη ρουτίνα εξυπηρέτησης και γράφει τον χαρακτήρα στην πόρτα A του 8155. Τέλος ο 8155 ενεργοποιεί το σήμα A BF (Buffer Full) ειδοποιώντας έτσι τον εκτυπωτή, ότι πρέπει να λάβει τον χαρακτήρα μέσω των γραμμών $PA_0 - PA_7$. Όταν ο χαρακτήρας τυπωθεί, ο εκτυπωτής στέλνει νέο παλμό \overline{RDY} και η διαδικασία επαναλαμβάνεται.

Οι χαρακτήρες που στέλνονται για εκτύπωση, αποθηκεύονται από το κύριο πρόγραμμα σε μια περιοχή μνήμης (Buffer, Απομονωτή), ξεκινώντας από τη διεύθυνση TEXT_BUF. Επίσης στη θέση μνήμης NUM τοποθετείται το πλήθος των χαρακτήρων που είναι αποθηκευμένοι. Να δοθεί η ρουτίνα εξυπηρέτησης καθώς και το τμήμα του κύριου προγράμματος, που κάνει τις αναγκαίες αρχικοποιήσεις κάθε φορά που ο απομονωτής είναι έτοιμος για εκτύπωση. Οι διευθύνσεις που αντιστοιχούν στον καταχωρητή κατάστασης και στη θύρα A του 8155 είναι 20H και 21H.



Σχήμα 6.9. Διασύνδεση του εκτυπωτή με τον 8155

;Κύριο πρόγραμμα. Εντολές αρχικοποίησης

```
LXI H, TEXT_BUFF ;Αρχικοποίησε τον δείκτη του πίνακα
SHLD POINTER
MVI A, 00010101B ;Επέτρεψε διακοπές για την πόρτα A
OUT 20H ;του 8155 (ακίδα A INTR)
```

;Ρουτίνα εξυπηρέτησης

RST 6.5:

```
PUSH PS ;Φύλαξε τις σημαίες και τους καταχωρητές
PUSH H ;που θα χρησιμοποιηθούν
LHLD POINTER ;Διάβασε το χαρακτήρα από τον απομονωτή
MOV A, M ;στον οποίο δείχνει ο δείκτης
OUT 21H ;Γράψε το χαρακτήρα στην πόρτα του 8155
INX H ;Αύξησε και φύλαξε το δείκτη του πίνακα
SHLD POINTER
LDA NUM ;Μείωσε το μετρητή χαρακτήρων στο Buffer
DCR A
STA NUM
CPI 0 ;Αν δεν υπάρχουν άλλοι χαρακτήρες
JNZ CONT ;απένεργοποίησε τις διακοπές για τη θύρα A
MVI 00000101B ;του 8155 στέλνοντας την κατάλληλη
OUT 20H ;εντολή στον καταχωρητή C/S
```

CONT:

```
POP H ;Αποκατέστησε σημαίες και καταχωρητές
POP PSW
EI
RET
```

Στο παραπάνω πρόγραμμα χρησιμοποιείται η θέση μνήμης POINTER για τη φύλαξη του δείκτη, που δείχνει κάθε φορά ποιός χαρακτήρας από τον απομονωτή πρέπει να τυπωθεί. Επίσης στη θέση μνήμης NUM φυλάγεται ο αριθμός των bytes, που περιέχει ο απομονωτής. Κάθε φορά

που ο απομονωτής γεμίζει με δεδομένα, το κύριο πρόγραμμα ενημερώνει τη NUM και στη συνέχεια εκτελείται το τμήμα του κύριου προγράμματος, που δόθηκε παραπάνω. Το τμήμα αυτό αρχικοποιεί τον POINTER, ώστε να δείχνει στην αρχή του απομονωτή. Επίσης επιτρέπει τις διακοπές για την θύρα A του 8155, που προκαλούνται στον επεξεργαστή μέσω της ακίδας A INTR. Για να γίνει αυτό η κατάλληλη λέξη καταχωρείται στον C/S.

Η ρουτίνα εξυπηρέτησης αυξάνει το δείκτη και μειώνει το NUM, αφού τυπώθηκε ένας χαρακτήρας. Στη συνέχεια αν το NUM είναι 0, δηλαδή δεν υπάρχουν άλλοι χαρακτήρες για εκτύπωση, η ρουτίνα πριν επιστρέψει απενεργοποιεί τις διακοπές για την πόρτα A του 8155. Η διακοπές αυτές θα επιτραπούν και πάλι από το κύριο πρόγραμμα με τις παραπάνω εντολές, μόλις αυτό γεμίσει πάλι το απομονωτή με χαρακτήρες.

Στο πρόγραμμα χρησιμοποιούνται δύο εντολές του 8155. Η εντολή 00010101 με την οποία επιλέγεται ο τρόπος λειτουργία ALT3 για τον 8155 (bits 3,2=01), καθορίζεται ότι η θύρα A θα χρησιμοποιηθεί σαν έξοδος (bit 0=1) και επιτρέπονται οι διακοπές για την θύρα A (bit 4=1). Για τη δεύτερη εντολή 00000101, ισχύουν τα ίδια εκτός από το ότι η διακοπές απενεργοποιούνται (bit4=0).