
Εξέλιξη των Μικροεπεξεργαστών / Μικροελεγκτών

1.1. Η αρχιτεκτονική των μικροεπεξεργαστών

Οι διάφορες αρχιτεκτονικές των σύγχρονων μικροεπεξεργαστών διακρίνονται με βάση τα ιδιαίτερα χαρακτηριστικά τους και τις ιδιότητες που έχει κάθε σύστημα, που λαμβάνει υπόψη του το συνολικό αριθμό των εντολών και τον τρόπο εκτέλεσης αυτών.

Ένα παρά πολύ διαδεδομένο σύστημα όπως αυτό που χρησιμοποιούμε στους επιτραπέζιους υπολογιστές, στους φορητούς υπολογιστές κ.λπ., είναι η **αρχιτεκτονική CISC** (*Complex Instruction Set Computer – Αρχιτεκτονική Σύνθετου Ρεπερτορίου Εντολών*). Επίσης, μια άλλη αρχιτεκτονική που κυριαρχεί σήμερα στην πλειοψηφία των ηλεκτρονικών συσκευών είναι η **αρχιτεκτονική RISC** (*Reduced Instruction Set Computer – Αρχιτεκτονική Μειωμένου Ρεπερτορίου Εντολών*).

Σε αυτήν βασίζονται σήμερα όλοι οι αυτοματισμοί των βιομηχανικών αυτοματισμών παραγωγής συσκευών και εξαρτημάτων, των οικιακών συσκευών, αυτοκινήτων, φωτογραφικών μηχανών, κινητών τηλεφώνων, αεροπλοΐας, ιπτάμενων drones, διαστημικών συσκευών, κλπ.

Αυτό που ξεχωρίζει έναν μικροεπεξεργαστή CISC από έναν μικροεπεξεργαστή RISC είναι η ύπαρξη στον CISC ενός μεγάλου συνόλου εντολών, στο οποίο συμπεριλαμβάνονται και σύνθετες.

Οι εντολές ενός μικροεπεξεργαστή CISC εκτελούνται σε πάνω από 4 έως 12 κύκλους μηχανής.

Αντίθετα η αρχιτεκτονική RISC φημίζεται για το λιτό και κομψό ρεπερτόριο εντολών, που εκτελούνται σε έναν κύκλο μηχανής!

Ένα άλλο μέρος με βάση το οποίο διακρίνουμε και διαχωρίζουμε τις διατάξεις των μικροεπεξεργαστών αφορά τον τρόπο με τον οποίο πραγματοποιείται η επικοινωνία στη μνήμη προγράμματος και στη μνήμη των δεδομένων.

Υπάρχουν δυο βασικές διατάξεις με τις οποίες πραγματοποιούμε πρόσβαση στη μνήμη προγράμματος και στη μνήμη δεδομένων.

Το πρώτο μοντέλο μνήμης που είναι γνωστό με την ονομασία αρχιτεκτονική **Von Neumann**, ή αρχιτεκτονική **Princeton**, το οποίο σχεδιάστηκε με βάση τις αρχές που διατυπώθηκαν από τον καθηγητή και ερευνητή **Τζον φον Νόιμαν** τη δεκαετία του 1940 στο πανεπιστήμιο του Princeton.

Οι αρχές, οι οποίες αναφέρονται αμέσως παρακάτω, συνιστούν την αρχιτεκτονική **φον Νόιμαν**.

- Τα δεδομένα και οι εντολές των εκτελούμενων προγραμμάτων αποθηκεύονται σε μια μοναδική μνήμη εγγραφής - ανάγνωσης.
- Τα περιεχόμενα της μνήμης αυτής διευθυνσιοδοτούνται κατά κελί, χωρίς να μας ενδιαφέρει ο τύπος των δεδομένων που περιέχεται στα κελιά μνήμης.
- Η εκτέλεση των εντολών του προγράμματος πραγματοποιείται σειριακά (εκτός και αν υπάρχει ρητή διακλάδωση) από μια εντολή στην επόμενη.

Και το δεύτερο μοντέλο με την ονομασία αρχιτεκτονική **Χάρβαρντ (Harvard Architecture)** το οποίο προβλέπει ξεχωριστό χώρο μνήμης για την αποθήκευση προγράμματος και για την αποθήκευση των δεδομένων.

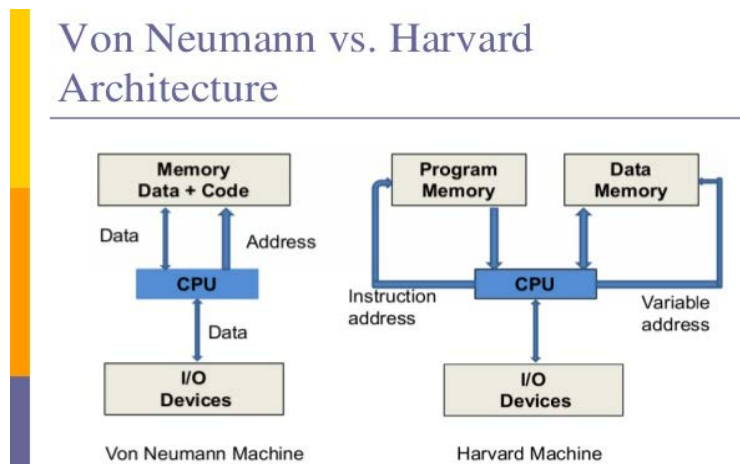
Επίσης το μήκος των λέξεων στο χώρο του προγράμματος δύναται να είναι διαφορετικό από το μήκος των λέξεων στο χώρο των δεδομένων.

Η αρχιτεκτονική αυτή είναι κυρίαρχη σήμερα στους μικροελεγκτές.

Μια άλλη διακριτή διαφορά σε ένα σύστημα μικροεπεξεργαστή είναι ο τρόπος με τον οποίο διαχωρίζει και αποθηκεύει τα δεδομένα εντός της CPU καθώς και οι διάφοροι τρόποι προσπέλασης αυτών.

Έτσι βάσει αυτών που αναφέραμε, διακρίνουμε καθαρά τα βασικά πρότυπα τα οποία εξηγούνται παρακάτω.

Von Neumann vs. Harvard Architecture



Σχήμα 1. Γενικό σχηματικό διάγραμμα αρχιτεκτονικής Von Neumann και Harvard. Παρατηρούμε την αρχιτεκτονική και τις διαφορές από τα δυο μοντέλα.

1.1.1. Μοντέλα λειτουργίας

1. Το μοντέλο στοίβας:

Το μοντέλο λειτουργίας τέτοιας μηχανής βασίζεται στο μηχανισμό στοίβας, η οποία τοποθετεί τις μεταβλητές στην αρχή του σορού και η (ALU – Arithmetic Logic Unit) λαμβάνει τις τιμές από το υψηλότερο μέρος της στοίβας και στη συνέχεια εκτελεί την πράξη. Το αποτέλεσμα των πράξεων αποθηκεύεται και πάλι στη στοίβα σε υψηλότερη θέση

2. Το μοντέλο συσσωρευτή:

Το πρότυπο μηχανής με συσσωρευτή (Accumulator) εκτελεί έναν υπολογισμό ως εξής. Ο ένας από τους δυο τελεστές είναι πάντα ο ίδιος, έτσι όλες οι πράξεις ουσιαστικά εκτελούνται χρησιμοποιώντας τον συσσωρευτή του συστήματος.

3. Το μοντέλο πολλών καταχωρητών:

Στο πρότυπο αυτό των πολλών καταχωρητών η πρόσβαση στη μνήμη επιτυγχάνεται με τη βοήθεια των καταχωρητών ως εξής:

Πρώτα φορτώνουμε τις τιμές των μεταβλητών στους καταχωρητές, στη συνέχεια εκτελούμε την πράξη και το αποτέλεσμα που παράγεται αποθηκεύεται πάλι σε έναν προηγούμενο καταχωρητή και από εκεί στη μεταβλητή προορισμού που ορίζουμε μέσα από το πρόγραμμα.

Στις μέρες μας έχει υιοθετηθεί το μοντέλο των πολλών καταχωρητών. Αυτό οφείλεται στη διαδικασία πρόσβασης των εσωτερικών καταχωρητών η οποία είναι ταχύτερη σε σχέση με την πρόσβαση εξωτερικής μνήμης. Αλλά και στο επίπεδο της μεταγλώττισης έχουμε ευκολότερη πρόσβαση στους εσωτερικούς καταχωρητές. Τέλος, για όλους αυτούς τους λόγους οι σύγχρονοι μικροελεγκτές έχουν ένα αρκετά μεγάλο σύνολο γενικών καταχωρητών.

Σημείωση: Το πώς λειτουργούν τα μοντέλα μηχανών στοίβας, καταχωρητών, οι μνήμες, οι χρονομετρητές κ.λπ. αναφέρονται παρακάτω αναλυτικά στην εξέλιξη του βιβλίου.

1.2. Τι είναι οι μικροελεγκτές και σε τι διαφέρουν από τους μικροεπεξεργαστές.

Ένας μικροελεγκτής είναι ένας μικροεπεξεργαστής με επιπλέον όλα τα απαραίτητα κυκλώματα υποστήριξης.

Τα οποία βασικά κυκλώματα είναι τα εξής:

- Οι μονάδες εισόδου εξόδου, για τη διασύνδεση του συστήματος με τον έξω κόσμο.
- Η μνήμη προγράμματος και δεδομένων για την αποθήκευση του προγράμματος και των αποτελεσμάτων των πράξεών του, (EEPROM, RAM).
- Οι χρονομετρητές (TIMER), και διάφορες άλλες εσωτερικές μονάδες.

Έτσι εάν όλες αυτές οι διακριτές μονάδες τοποθετηθούν μέσα στο σώμα ενός μόνο ολοκληρωμένου τότε έχουμε δημιουργήσει έναν μικροελεγκτή!

Ενώ αντίθετα όλα αυτά τα χαρακτηριστικά που προαναφέρθηκαν απουσιάζουν από έναν μικροεπεξεργαστή ο οποίος χρειάζεται εξωτερικές περιφερειακές μονάδες για την επικοινωνία του με το εξωτερικό περιβάλλον.

Ιστορικά ο πρώτος μικροελεγκτής στον κόσμο κυκλοφόρησε το 1974 από την Texas instruments και ήταν ο TMS1000, ένας 4bit microcontroller με ρολόι χρονισμού στα 400Khz!

Ολοκληρώνοντας και όπως γνωρίσαμε παραπάνω οι διαφορές μεταξύ του μικροελεγκτή και του μικροεπεξεργαστή είναι ακριβείς και διακριτές μεταξύ τους.

Στο μέρος αυτό θα αναφερθούμε στις διάφορες διατάξεις μικροελεγκτών που μπορεί να συναντήσουμε, και να τους χωρίσουμε σε κατηγορίες.

Ας φανταστούμε ότι θέλουμε να κατασκευάσουμε ένα σύστημα ικανό να συνδεθεί με διάφορες εξωτερικές μονάδες. πχ (μονάδες απεικόνισης, (DISPLAY), κινητήρες, Relay, led, κλπ. και έχουμε στη διάθεσή μας μια απλή μονάδα κεντρικής επεξεργασίας (CPU), στην προκειμένη περίπτωση θα χρειαστούμε, για να το κατασκευάσουμε, διάφορες λειτουργικές μονάδες, όπως μνήμη για το πρόγραμμα, μνήμη για τα δεδομένα, (Flash, Ram, Eeprom). Μονάδες διασύνδεσης εισόδου - εξόδου, όπως αναλογικούς μετατροπής (ADC), αναλογικούς συγκριτές κλπ.

Όλες αυτές οι διάφορες μονάδες μας οδηγούν σε ένα σύστημα με πολυάριθμα εξαρτήματα, με μεγαλύτερη δυσκολία σχεδίασης και υλοποίησης του συστήματος που θέλουμε να κατασκευάσουμε, προφανώς με μεγαλύτερη κατανάλωση ισχύος, και ασφαλώς με μεγαλύτερο κόστος κατασκευής.

Αυτό λοιπόν που ξεχωρίζει έναν μικροελεγκτή είναι ότι όλα αυτά τα συστατικά μέρη που αναφέρθηκαν παραπάνω προϋπάρχουν ενσωματωμένα σε ένα μόνο απλό ολοκληρωμένο κύκλωμα!

Συμπέρασμα:

Ο Μικροελεγκτής είναι ένα ολοκληρωμένο κύκλωμα το οποίο διαθέτει κεντρική μονάδα επεξεργασίας, μνήμη, διάφορες περιφερειακές μονάδες, κυκλώματα διαύλων με θύρες εισόδου/εξόδου για την επικοινωνία με το εξωτερικό περιβάλλον, συστήματα χρονισμού, κλπ. Τα παραπάνω χαρακτηριστικά είναι ενσωματωμένα σε ένα μόνο ολοκληρωμένο κύκλωμα (Chip), πράγμα που το καθιστά ικανό να λειτουργήσει αυτόνομα.

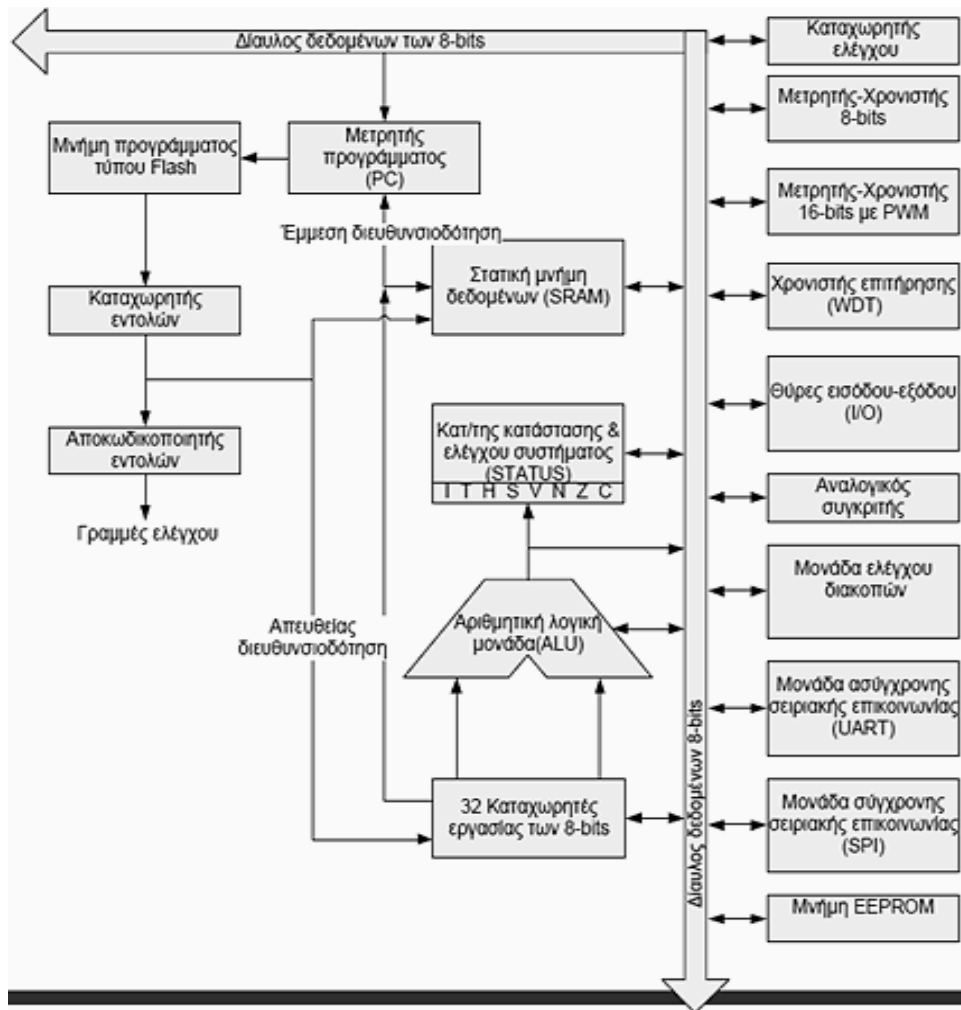
1.3. Τα συστατικά μέρη του μικροελεγκτή και οι λειτουργίες τους

1. Μονάδα κεντρικής επεξεργασίας (CPU)

Η μονάδα αυτή αποτελεί την καρδιά του μικροελεγκτή. Ανακαλεί και αποκωδικοποιεί τα δεδομένα από τη μνήμη προγράμματος υπό μορφή εντολών και κατόπιν τις εκτελεί.

Η μονάδα κεντρικής επεξεργασίας (CPU) αποτελείται από:

- Την αριθμητική λογική μονάδα, (ALU Arithmetic logic unit).
- Τους καταχωρητές, (Registers).
- Τον αποκωδικοποιητή εντολών με τα κυκλώματα χρονισμού και ελέγχου.



Σχήμα 2. Γενικό σχηματικό διάγραμμα ενός Μικροελεγκτή AVR ATmega8-88-328.

2. Μνήμη προγράμματος:

Στη μνήμη αυτή αποθηκεύεται το πρόγραμμα, ουσιαστικά οι εντολές που απαρτίζουν αυτό. Η μνήμη αυτή συναντάται ως μνήμη ταχείας αποθήκευσης (FLASH MEMORY). Χρησιμοποιείται σήμερα κατά κόρον σε όλα τα συστήματα μικροελεγκτών. Το μέγεθος της μνήμης στους μικροελεγκτές γενικά είναι μικρό και εξαρτάται από τον τύπο του μικροελεγκτή. Για τον λόγο αυτό εάν θελήσουμε την εκτέλεση μεγαλύτερου προγράμματος δύναται να προσθέσουμε εξωτερική μνήμη προγράμματος SRAM.

3. Μνήμη SRAM:

Η αλλιώς μνήμη τυχαίας προσπέλασης (Static Random Access Memory). Η μνήμη αυτή χρησιμοποιείται από τον ελεγκτή για την αποθήκευση των δεδομένων.

Η CPU επίσης χρησιμοποιεί τη μνήμη RAM για την αποθήκευση μεταβλητών, επίσης χρησιμοποιεί και τη στοίβα (stack) για την προσωρινή αποθήκευση των διευθύνσεων επιστροφής, ώστε να συνεχίσει απρόσκοπτα την εκτέλεση του προγράμματος από το σημείο διακοπής λόγω εμφάνισης υπορουτίνας, (Subroutine) ή από ρουτίνα διακοπής, (Interrupt routine).

Εδώ θα πρέπει να γνωρίζουμε ότι η μνήμη SRAM διατηρεί τα περιεχόμενά της όσο της παρέχεται ηλεκτρική τάση (ισχύς). Για αυτό τον λόγο η SRAM ονομάζεται και προσωρινή μνήμη ή πτητική μνήμη (volatile memory).

4. Ταλαντωτής χρονισμού:

Ο ταλαντωτής χρονισμού είναι το κύκλωμα με το οποίο καθορίζεται ο ρυθμός εκτέλεσης των εντολών (πρόγραμμα) από τη μνήμη προγράμματος.

Ο ρυθμός καθορίζεται από τη συχνότητα λειτουργίας του ταλαντωτή χρονισμού. Ο ταλαντωτής αυτός υλοποιείται είτε με ένα εξωτερικό στοιχείο χρονισμού, συνήθως με ένα κρύσταλλο (Quartz), ένα κεραμικό στοιχείο (resonator) και σε ορισμένους μικροελεγκτές ακόμα και με ένα εξωτερικό κύκλωμα RC, ή με τον εσωτερικό RC ταλαντωτή του μικροελεγκτή.

5. Σειριακή θύρα επικοινωνίας:

Η θύρα αυτή είναι από τα πλέον σπάνια συστατικά ενός μικροελεγκτή και χρησιμοποιείται για την επικοινωνία του ελεγκτή με διάφορα εξωτερικά κυκλώματα.

Η λειτουργία της θύρας βασίζεται στην τεχνική μορφή της σειριακής μετάδοσης των δεδομένων σε διάφορες ταχύτητες, δηλαδή λαμβάνει δεδομένα από την είσοδό της τα οποία ολισθαίνει προς την έξοδο κατά ένα δυαδικό ψηφίο (bit) τη φορά, σχηματίζοντας με 8 τέτοια δυαδικά ψηφία (bits) μια λέξη που αντιστοιχεί σε ένα (byte).

Οι σειριακές θύρες υπάρχουν σε δυο τύπους.

Την ασύγχρονη σειριακή μετάδοση και την σύγχρονη σειριακή μετάδοση.

Η λειτουργία μιας σύγχρονης σειριακής θύρας χρειάζεται παλμό σήματος συγχρονισμού (clock).

Αντίθετα σε μια ασύγχρονη σειριακή θύρα δεν απαιτείται η ύπαρξη παλμού συγχρονισμού διότι οι πληροφορίες χρονισμού και συγχρονισμού είναι ενσωματωμένες στο σύνολο των δεδομένων που μεταδίδονται σειριακά, με τη χρήση πρόσθετων (bits) μέσω των οποίων πληροφορείται η έναρξη και η παύση μιας συγκεκριμένης μετάδοσης δεδομένων (Start bit και stop bit) αντίστοιχα.

6. Ψηφιακή θύρα εισόδου εξόδου:

Είναι οι θύρες I/O με τις οποίες επικοινωνεί ο μικροελεγκτής από και προς το εξωτερικό περιβάλλον.

Σε μια τέτοια θύρα η ανταλλαγή δεδομένων γίνεται υπό τη μορφή ομάδων των 8 δυαδικών ψηφίων (bits) ή ένα (byte) μια λέξη.

7. Αναλογική θύρα εισόδου εξόδου:

Οι θύρες αυτές είναι μετατροπείς αναλογικού σήματος σε ψηφιακό (**ADC Analog to Digital Converter**), χρησιμοποιούνται για την ανάγνωση των δεδομένων από διάφορους αισθητήρες πχ, Θερμοκρασίας - Πίεσης - Υγρασίας κλπ.

Επίσης στους περισσότερους μικροελεγκτές υπάρχουν και κυκλώματα με διαμόρφωση εύρους παλμών (Pulse Width Modulators ή PWM) μέσα από τις οποίες με τα κατάλληλα φίλτρα τύπου RC μπορούμε να λάβουμε αναλογικές τάσεις. Οι μετατροπείς του τύπου PWM, χρησιμοποιούνται συνήθως για την οδήγηση βηματικών κινητήρων και τη δημιουργία γεννητριών διαφόρων παλμών.

8. Κύκλωμα χρονισμού:

Αυτό το εσωτερικό κύκλωμα χρησιμοποιείται από τον μικροελεγκτή για το χρονισμό διαφόρων γεγονότων. Για παράδειγμα αν επιθυμούμε να αποστείλουμε δεδομένα σε μια εξωτερική συσκευή με ένα συγκεκριμένο ρυθμό, τότε χρησιμοποιούμε το κύκλωμα του χρονισμού ώστε να δημιουργήσουμε το συγκεκριμένο ρυθμό.

Επίσης ένα τέτοιο κύκλωμα δύναται να χρησιμοποιηθεί και για την καταμέτρηση στιγμών είτε εσωτερικά του μικροελεγκτή είτε εξωτερικά δηλαδή σαν μετρητής (Counter) γεγονότων.

9. Ρολόι πραγματικού χρόνου:

Ένα ακόμα κύκλωμα χρονισμού ειδικού σκοπού είναι το ρολόι πραγματικού χρόνου (Real Time Clock η RTC), ο σκοπός του οποίου είναι η διατήρηση

της τρέχουσας ώρας και ημερομηνίας.

Μπορεί να χρησιμοποιηθεί σε διάφορες εφαρμογές όπου χρειαζόμαστε διάφορα στοιχεία με γνώμονα την ημερομηνία ή την ώρα.

10. Κύκλωμα χρονισμού επιτήρησης:

Επίσης ένα ακόμη κύκλωμα ειδικού σκοπού είναι και το κύκλωμα επιτήρησης (**Watchdog Timer ή WDT**). Το κύκλωμα αυτό χρησιμοποιείται συνήθως για να αποφύγουμε την πιθανή κατάρρευση ενός συστήματος με μικροελεγκτή.

11. Κύκλωμα επανεκκίνησης (Reset) ή μηδενισμού:

Το κύκλωμα αυτό μας παρέχει τη δυνατότητα επανεκκίνησης του μικροελεγκτή (Reset).

Το γεγονός αυτό μας εξασφαλίζει ότι όλα τα κυκλώματα και οι εσωτερικές μονάδες του μικροελεγκτή θα εκκινήσουν τη λειτουργία τους από την προκαθορισμένη αρχική κατάσταση.

Η λειτουργία του κυκλώματος επανεκκίνησης βασίζεται στο εσωτερικό κύκλωμα ανίχνευσης βύθισης της τάσης τροφοδοσίας που υπάρχει μέσα στον μικροελεγκτή, το οποίο ελέγχει συνεχώς το επίπεδο της τάσης τροφοδοσίας, εάν ανιχνευθεί κάποια στιγμιαία βύθιση της τάσης τότε αυτόματα θα επανεκκινηθεί ο μικροελεγκτής προστατεύοντας με αυτόν τον τρόπο τα περιεχόμενα της μνήμης και των καταχωρητών από πιθανή αλλοίωση ή βλάβη, γεγονός που θα οδηγούσε τον μικροελεγκτή σε απρόβλεπτη λειτουργία.