ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Εργαστήριο
Μικροελεγκτές - Ενσωματωμένα Συστήματα

1ος ΚΥΚΛΟΣ

Μικροελεγκτές Atmel – AVR

ΑΣΚΗΣΗ 2η

Διαχείριση της Μνήμης

Γ. Καλτσάς

Καθηγητής

Αθήνα 2019

Άσκηση 2η: Διαχείριση της Μνήμης.

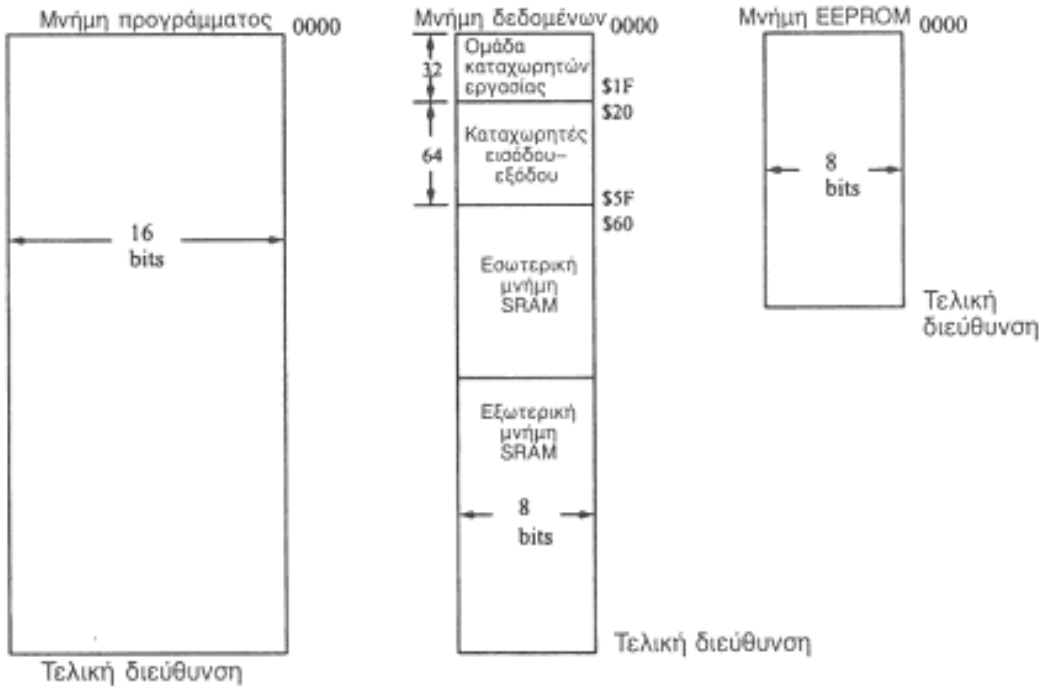
1. Η Αρχιτεκτονική της μνήμης στην οικογένεια AVR.

Οι μικροελεγκτές της οικογένειας AVR έχουν σχεδιαστεί σύμφωνα με την αρχιτεκτονική Harvard. Αυτό σημαίνει ότι διαθέτουν διαφορετικούς εσωτερικούς διαύλους για την μνήμη προγράμματος και για την μνήμη δεδομένων. Ο δίαυλος στον οποίον συνδέεται η μνήμη δεδομένων, είναι ένας δίαυλος των 8-bit και ο οποίος συνδέει τις περισσότερες από τις εσωτερικές περιφερειακές μονάδες με την ομάδα των ειδικών καταχωρητών ελέγχου. Ο δίαυλος στον οποίον συνδέεται η μνήμη προγράμματος έχει μήκος 16-bit και τροφοδοτεί μόνο τον καταχωρητή εντολών.

Η μνήμη προγράμματος αποτελεί έναν συνεχή χώρο μιας μνήμης τύπου flash. Η ακριβής χωρητικότητα της μνήμης αυτής διαφέρει από μικροελεγκτή σε μικροελεγκτή. Ο μικροελεγκτής AT90S1200, που αποτελεί και το βασικό μέλος της οικογένειας αυτής, διαθέτει μνήμη προγράμματος χωρητικότητας του 1Kbyte η οποία είναι οργανωμένη σε 512 x 16-bits, ενώ ο μικροελεγκτής Mega103 διαθέτει μνήμη προγράμματος των 128 Kbytes η οποία είναι οργανωμένη σε 64K x 16-bits. Η πρόσβαση στην μνήμη προγράμματος πραγματοποιείται σε κάθε μια περίοδο του σήματος χρονισμού και η αντίστοιχη εντολή που ανακαλείται φορτώνεται στον καταχωρητή εντολών.

Από την άλλη πλευρά, η μνήμη δεδομένων του συστήματος διαιρείται σε πολλούς διαφορετικούς τύπους. Στο σχήμα που ακολουθεί φαίνονται οι διάφοροι χάρτες μνήμης όπως διατάσσονται σε έναν τυπικό επεξεργαστή AVR. Η μνήμη δεδομένων χωρίζεται συνολικά σε πέντε διαφορετικά τμήματα:

1. Ένα τμήμα μήκους 32 bytes που αντιστοιχεί στην ομάδα των καταχωρητών εργασίας (register file) των 8-bits. Η ομάδα αυτή των καταχωρητών υπάρχει σε όλους τους μικροελεγκτές της οικογένειας AVR.
2. Ένα τμήμα 64 καταχωρητών εισόδου - εξόδου των 8-bits. Δε διαθέτουν όλοι οι μικροελεγκτές της σειράς AVR και τους 64 παραπάνω καταχωρητές. Μερικοί μικροελεγκτές διαθέτουν περισσότερους τέτοιους καταχωρητές, ανάλογα με το πλήθος των εσωτερικών περιφερειακών μονάδων. Οι καταχωρητές αυτοί αποτελούν ουσιαστικά τμήμα της ενσωματωμένης μνήμης SRAM και η προσπέλασή τους πραγματοποιείται στις διευθύνσεις μεταξύ $20 έως $5F ή σαν καταχωρητές εισόδου-εξόδου στις διευθύνσεις $00 έως $3F. Στις περισσότερες περιπτώσεις οι καταχωρητές αυτοί αντιμετωπίζονται ως καταχωρητές εισόδου - εξόδου παρά ως τμήμα της μνήμης SRAM.
3. Ένα τμήμα εσωτερικής στατικής μνήμης (SRAM). Η μνήμη αυτή περιλαμβάνεται σχεδόν σε όλους τους μικροελεγκτές της σειράς AVR, εκτός από τα βασικά μέλη όπως ο AT90S1200. Η χωρητικότητα της μνήμης αυτής κυμαίνεται από 128 bytes μέχρι και 4 Kbytes. Η μνήμη SRAM χρησιμοποιείται εκτός από την αποθήκευση μεταβλητών και ως στοίβα (stack) του συστήματος. Σε εκείνους του μικροελεγκτές της σειράς AVR οι οποίοι δε διαθέτουν εσωτερική μνήμη SRAM, η στοίβα χρησιμοποιείται μόνο για την αποθήκευση των διευθύνσεων επιστροφής μετά από κάποια κλήση στο κύριο πρόγραμμα. Αυτός ο τύπος της στοίβας μπορεί να αποθηκεύσει μέχρι το πολύ 3 διευθύνσεις επιστροφής.



Ο χάρτης μνήμης των μικροελεγκτών AVR.

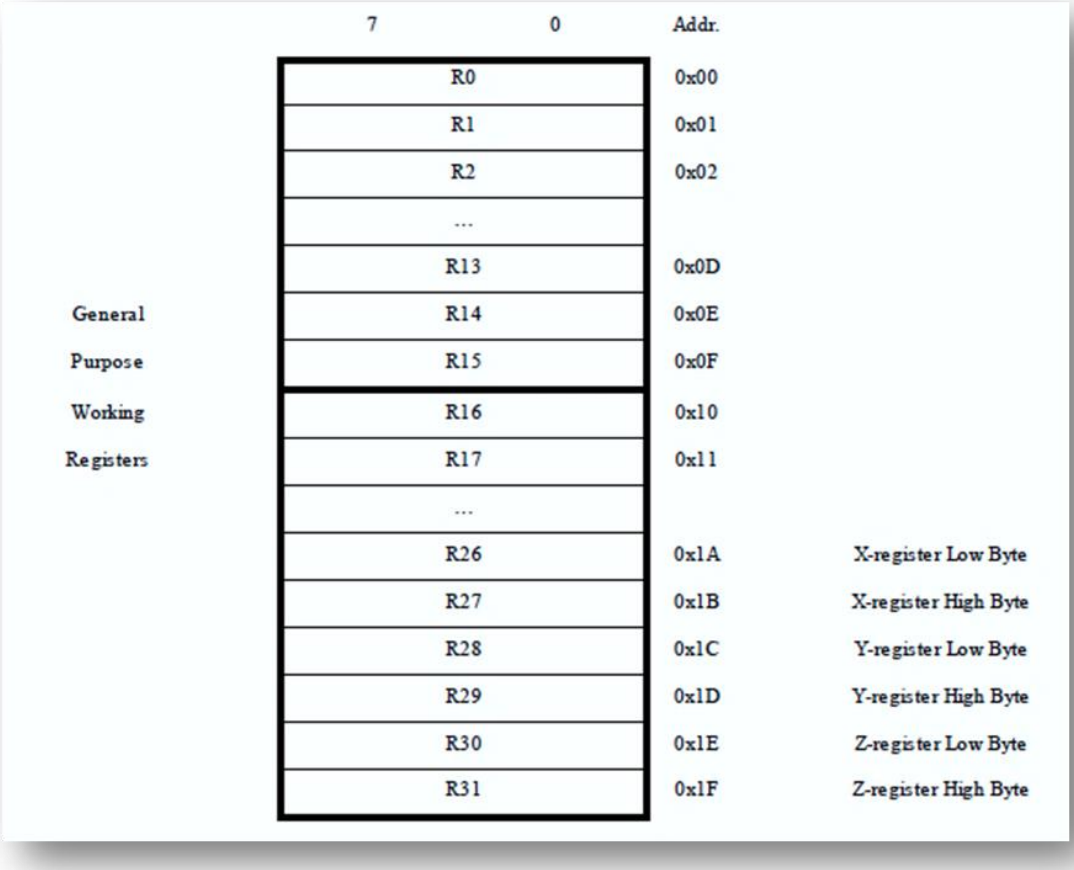
4. Ένα τμήμα εξωτερικής στατικής μνήμης SRAM. Η μνήμη αυτή χρησιμοποιείται μόνο στους μεγαλύτερους επεξεργαστές της σειράς AVR. Στους επεξεργαστές αυτούς που διαθέτουν θύρες πρόσβασης σε εξωτερικό δίαυλο δεδομένων και διευθύνσεων μπορεί να χρησιμοποιηθεί οποιαδήποτε ποσότητα εξωτερικής μνήμης SRAM επιθυμεί ο χρήστης.
5. Ένα τμήμα εσωτερικής μνήμης τύπου EEPROM. Αυτός ο τύπος της εσωτερικής μνήμης είναι διαθέσιμος σχεδόν σε όλους τους μικροελεγκτές AVR. Η χωρητικότητα της μνήμης αυτής κυμαίνεται από 64 bytes έως και 4 Kbytes στους διάφορους μικροελεγκτές της σειράς. Η EEPROM μπορεί να διαβαστεί και να γραφεί από οποιοδήποτε πρόγραμμα. Τονίζεται το γεγονός ότι η διαδικασία ενός κύκλου ανάγνωσης της μνήμης EEPROM είναι ταχύτερη από την αντίστοιχη διαδικασία ενός κύκλου εγγραφής. Επίσης θα πρέπει να γνωρίζουμε ότι η μνήμη EEPROM μπορεί να δεχθεί περίπου μέχρι και 100000 κύκλους εγγραφής.

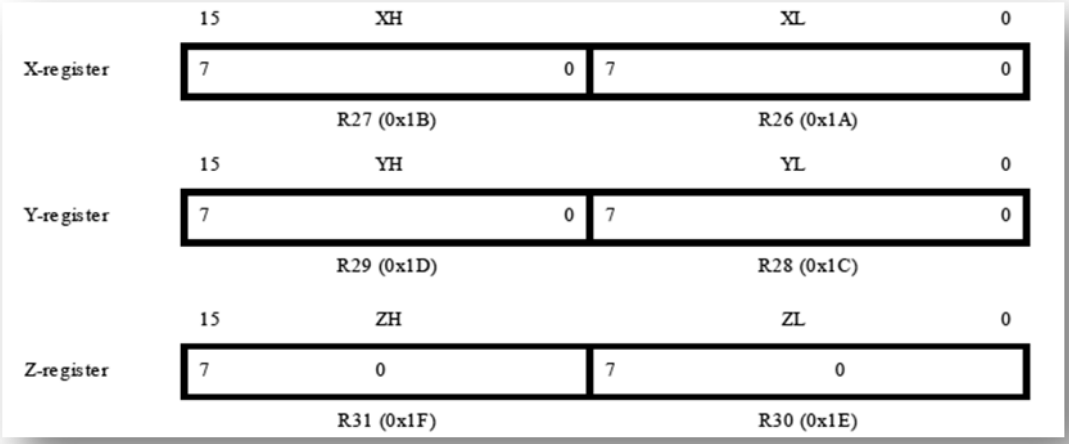
2. Πρόσβαση στη Μνήμη.

Από την άλλη πλευρά, η πρόσβαση στην μνήμη SRAM απαιτεί δύο περιόδους του κεντρικού σήματος χρονισμού. Αυτό οφείλεται στο γεγονός ότι η πρόσβαση στην SRAM χρησιμοποιεί έναν καταχωρητή δείκτη για τον προσδιορισμό της διεύθυνσης στην SRAM. Ο καταχωρητής δείκτη είναι ένας από τους καταχωρητές που ορίζονται από τα ζεύγη των καταχωρητών εργασίας X, Y και Z. Το χρονικό διάστημα που αντιστοιχεί στην πρώτη περίοδο του κεντρικού ρολογιού απαιτείται για την πρόσβαση στην ομάδα των καταχωρητών εργασίας και για την λειτουργία πάνω στα περιεχόμενα του καταχωρητή δείκτη (οι εντολές πρόσβασης στην SRAM επιτρέπουν και μια επιπλέον λειτουργία αύξησης της τιμής της τρέχουσας διεύθυνσης της μνήμης που βρίσκεται στον καταχωρητή δείκτη, πριν ή μετά από την εκτέλεση της κύριας εντολής). Προς το τέλος της πρώτης περιόδου του ρολογιού, η ALU εκτελεί αυτόν

τον απαιτούμενο υπολογισμό και η τιμή της διεύθυνσης που προκύπτει χρησιμοποιείται για την πρόσβαση μιας συγκεκριμένης θέσης στην SRAM όπου πρόκειται να λάβει χώρα κάποια εγγραφή (ή ανάγνωση).
Η μνήμη SRAM είναι διαθέσιμη στους περισσότερους από τους μεγάλους μικροελεγκτές της οικογένειας AVR. Οι τιμές της χωρητικότητας της μνήμης αυτής κυμαίνονται ανάλογα με την χρησιμοποιούμενη διάταξη, από 128 bytes μέχρι και 4 Kbytes. Η πρόσβαση στην μνήμη αυτή επιτυγχάνεται με τη βοήθεια ενός μεγάλου αριθμού εντολών που διαθέτει η γλώσσα των μικροελεγκτών AVR, είτε απευθείας, είτε έμμεσα με χρήση κάποιου καταχωρητή δείκτη. Επίσης, η μνήμη SRAM χρησιμοποιείται και για την λειτουργία της στοίβας.

Οι καταχωρητές-δείκτες X, Y, Z δεν είναι διακριτά μέρη του υλικού αλλά απαρτίζονται από την ένωση υπαρχόντων καταχωρητών. Ειδικότερα ο καταχωρητής-δείκτης X απαρτίζεται από την ένωση των δύο 8bit καταχωρητών R26 (low byte) και R27 (high byte) ώστε να σχηματιστεί ένας νέος καταχωρητής 16bit. Ανάλογα οι καταχωρητές-δείκτες Y, Z συντίθεται από την ένωση των υπαρχόντων καταχωρητών R28 (low byte) και R29 (high byte), για τον Y και R30 (low byte) και R31 (high byte) για τον Z αντίστοιχα. Η συσχέτιση των καταχωρητών-δεικτών με τους υπάρχοντες καταχωρητές απεικονίζεται στα δύο παρακάτω σχήματα.





3. Ανάλυση Εργαστηριακής Άσκησης

1) Δημιουργήστε κατάλληλο πρόγραμμα σε συμβολική γλώσσα που να διαβάζει από την πόρτα Α 10 συνεχόμενα bytes και να τα αποθηκεύει σε συνεχόμενες θέσεις μνήμης αρχίζοντας από την $0280. Τροποποιήστε το πρόγραμμα ώστε να αποθηκεύονται αυτόματα οι αριθμοί από 1-10 σε διαδοχικές θέσεις μνήμης που αρχίζουν από την $0280.

2) Δημιουργήστε κατάλληλο πρόγραμμα σε συμβολική γλώσσα, που να μεταφέρει 10 bytes της μνήμης που αρχίζουν στη διεύθυνση $0260 σε 10 bytes της μνήμης που αρχίζουν στη διεύθυνση $0280.

3) Δημιουργήστε κατάλληλο πρόγραμμα σε συμβολική γλώσσα, που να μεταφέρει 10 bytes της μνήμης που αρχίζουν στη διεύθυνση $0260 σε 10 bytes της μνήμης που αρχίζουν στη διεύθυνση $0280 και αντιστρόφως.

4) Δημιουργήστε κατάλληλο πρόγραμμα σε συμβολική γλώσσα, που να μεταφέρει 10 bytes της μνήμης που αρχίζουν στη διεύθυνση $0260 σε 10 bytes της μνήμης που αρχίζουν στη διεύθυνση $0280 και αντιστρόφως. Η τοποθέτηση να γίνει με την αντίστροφη σειρά, δηλαδή το πρώτο byte να πάει στην τελευταία θέση και το τελευταίο byte στην πρώτη και στις δύο περιοχές.

5) Δημιουργήστε κατάλληλο πρόγραμμα σε συμβολική γλώσσα, που να ελέγχει 20 bytes της μνήμης που αρχίζουν στη διεύθυνση $0280. Αν η τιμή τους είναι μικρότερη του 128 να μην τα αλλοιώνει, αλλιώς να τα μηδενίζει.

6) Δημιουργήστε κατάλληλο πρόγραμμα σε συμβολική γλώσσα, που να αντιστρέφει τη σειρά 20 bytes της μνήμης που αρχίζουν στη διεύθυνση $0280.

7) Δημιουργήστε κατάλληλο πρόγραμμα σε συμβολική γλώσσα, που να ανιχνεύει κλειδί μήκους ενός byte σε περιοχή της μνήμης που αρχίζει στη διεύθυνση $0280 και

έχει μήκος 32 bytes. Στην πόρτα Β να εμφανίζεται ο αριθμός των κλειδιών στο συγκεκριμένο διάστημα μνήμης. Τροποποιείστε το ανωτέρω πρόγραμμα έτσι ώστε να λαμβάνετε και τις διευθύνσεις της μνήμης στις οποίες βρίσκεται το ανιχνευόμενο κλειδί.

8) Δημιουργήστε κατάλληλο πρόγραμμα σε συμβολική γλώσσα, που να ανιχνεύει κλειδί μήκους δύο bytes σε περιοχή της μνήμης που αρχίζει στη διεύθυνση $0280 και έχει μήκος 32 bytes. Στην πόρτα Β να εμφανίζεται ο αριθμός των κλειδιών στο συγκεκριμένο διάστημα μνήμης. Τροποποιείστε το ανωτέρω πρόγραμμα έτσι ώστε να λαμβάνετε και τις διευθύνσεις της μνήμης στις οποίες αρχίζει το ανιχνευόμενο κλειδί.

ΠΑΡΑΡΤΗΜΑ

Ανάλυση Χρησιμοποιούμενων Εντολών

LD - Load Indirect from SRAM to Register using Index X

Περιγραφή:
Φορτώνει ένα byte έμμεσα από την SRAM σε καταχωρητή. Η τοποθεσία στην SRAM καθορίζεται από τον X (16 bits) καταχωρητή. Η προσπέλαση στην μνήμη περιορίζεται στην τρέχουσα σελίδα της SRAM των 64K bytes. Για να προσπελαθεί μία άλλη SRAM σελίδα, ο RAMPX καταχωρητής στην περιοχή I/O πρέπει να αλλάξει. Ο X καταχωρητής μπορεί είτε να αφεθεί ο ίδιος μετά την λειτουργία ή μπορεί να αυξηθεί ή να μειωθεί. Αυτά τα πλεονεκτήματα ταιριάζουν ιδιαίτερα για επεξεργασία πινάκων και χρήση του καταχωρητή X σαν δείκτη σωρού (stack pointer).

Using the X pointer

Λειτουργία:	**Σχόλιο:**
(i) Rd ← (X)	X: Unchanged
(ii) Rd ← (X), X ← X + 1	X: Post incremented
(iii) X ← X – 1, Rd ← (X)	X: Pre decremented

Σύνταξη:	**Τελεστές:**	**Μετρητής προγράμματος:**
(i) LD Rd, X	0 ≤ d ≤ 31	PC ← PC + 1
(ii) LD Rd, X+	0 ≤ d ≤ 31	PC ← PC + 1
(iii) LD Rd,-X	0 ≤ d ≤ 31	PC ← PC + 1

Παράδειγμα:
```
clr r27 ; Clear X high byte
ldi r26,$20 ; Set X low byte to $20
ld r0,X+ ; Load r0 with SRAM loc. $20(X post inc)
ld r1,X ; Load r1 with SRAM loc. $21
ldi r26,$23 ; Set X low byte to $23
ld r2,X ; Load r2 with SRAM loc. $23
ld r3,-X ; Load r3 with SRAM loc. $22(X pre dec)
```
Words: 1 (2 bytes)
Cycles: 2

LD (LDD) - Load Indirect from SRAM to Register using Index Y

Περιγραφή:
Φορτώνει ένα byte έμμεσα από την SRAM σε καταχωρητή. Η τοποθεσία στην SRAM καθορίζεται από τον Y (16 bits) καταχωρητή. Η προσπέλαση στην μνήμη

περιορίζεται στην τρέχουσα σελίδα της SRAM των 64K bytes. Για να προσπελαθεί μία άλλη SRAM σελίδα, ο RAMPY καταχωρητής στην περιοχή I/O πρέπει να αλλάξει. Ο Y καταχωρητής μπορεί είτε να αφεθεί ο ίδιος μετά την λειτουργία, ή μπορεί να αυξηθεί ή να μειωθεί. Αυτά τα πλεονεκτήματα ταιριάζουν ιδιαίτερα για επεξεργασία πινάκων και χρήση του καταχωρητή Y σαν δείκτη σωρού (stack pointer).

Using the Y pointer:

Λειτουργία:	**Σχόλιο:**
(i) Rd ← (Y)	Y: Unchanged
(ii) Rd ← (Y), Y ← Y + 1	Y: Post incremented
(iii) Y ← Y – 1, Rd ← (Y)	Y: Pre decremented
(iiii) Rd ← (Y+q)	Y: Unchanged, q: Displacement

Σύνταξη:	**Τελεστές:**	**Μετρητής προγράμματος:**
(i) LD Rd, Y	0 ≤ d ≤ 31	PC ← PC + 1
(ii) LD Rd, Y+	0 ≤ d ≤ 31	PC ← PC + 1
(iii) LD Rd,-Y	0 ≤ d ≤ 31	PC ← PC + 1
(iiii) LDD Rd, Y+q	0 ≤ d ≤ 31, 0 ≤ q ≤ 63	PC ← PC + 1

Παράδειγμα:

```
clr r29 ; Clear Y high byte
ldi r28,$20 ; Set Y low byte to $20
ld r0,Y+ ; Load r0 with SRAM loc. $20(Y post inc)
ld r1,Y ; Load r1 with SRAM loc. $21
ldi r28,$23 ; Set Y low byte to $23
ld r2,Y ; Load r2 with SRAM loc. $23
ld r3,-Y ; Load r3 with SRAM loc. $22(Y pre dec)
ldd r4,Y+2 ; Load r4 with SRAM loc. $24
```

Words: 1 (2 bytes)
Cycles: 2

LD (LDD) - Load Indirect From SRAM to Register using Index Z

Περιγραφή:
Φορτώνει ένα byte έμμεσα, από την SRAM σε καταχωρητή. Η τοποθεσία στην SRAM καθορίζεται από τον Z (16 bits) καταχωρητή. Η προσπέλαση στην μνήμη περιορίζεται στην τρέχουσα σελίδα της SRAM των 64K bytes. Για να προσπελαθεί μία άλλη SRAM σελίδα, ο RAMPZ καταχωρητής στην περιοχή I/O πρέπει να αλλάξει. Ο Z καταχωρητής μπορεί είτε να αφεθεί ο ίδιος μετά την λειτουργία, ή μπορεί να αυξηθεί ή να μειωθεί. Αυτά τα πλεονεκτήματα ταιριάζουν ιδιαίτερα για χρήση σαν δείκτη σωρού (stack pointer) του Z καταχωρητή, όμως επειδή ο Z pointer μπορεί να χρησιμοποιηθεί για έμμεσες κλήσεις υπορουτίνας, έμμεσες υπερπηδήσεις και εξέταση πινάκων, είναι συχνά πιο βολικό να χρησιμοποιούνται ο X ή ο Y σαν ένας δείκτης σωρού (stack pointer). Για χρήση του Z pointer για εξέταση πινάκων στην μνήμη προγράμματος δείτε την εντολή LPM.

Using the Z pointer:

Λειτουργία:	**Σχόλιο:**

(i) Rd ← (Z) Z: Unchanged
(ii) Rd ← (Z), Z ← Z + 1 Z: Post increment
(iii) Z ← Z –1, Rd ← (Z) Z: Pre decrement
(iiii) Rd ← (Z+q) Z: Unchanged, q: Displacement

Σύνταξη:	**Τελεστές:**	**Μετρητής προγράμματος:**
(i) LD Rd, Z	0 ≤ d ≤ 31	PC ← PC + 1
(ii) LD Rd, Z+	0 ≤ d ≤ 31	PC ← PC + 1
(iii) LD Rd,-Z	0 ≤ d ≤ 31	PC ← PC + 1
(iiii) LDD Rd, Z+q	0 ≤ d ≤ 31, 0 ≤ q ≤ 63	PC ← PC + 1

Παράδειγμα:
```
clr r31 ; Clear Z high byte
ldi r30,$20 ; Set Z low byte to $20
ld r0,Z+ ; Load r0 with SRAM loc. $20(Z post inc)
ld r1,Z ; Load r1 with SRAM loc. $21
ldi r30,$23 ; Set Z low byte to $23
ld r2,Z ; Load r2 with SRAM loc. $23
ld r3,-Z ; Load r3 with SRAM loc. $22(Z pre dec)
ldd r4,Z+2 ; Load r4 with SRAM loc. $24
```
Words: 1 (2 bytes)
Cycles: 2

LDS - Load Direct from SRAM

Περιγραφή:
Φορτώνει ένα byte από την SRAM σε ένα καταχωρητή. Μία 16 bits διεύθυνση πρέπει να δοθεί. Η προσπέλαση μνήμης είναι περιορισμένη στην τρέχουσα SRAM σελίδα των 64K bytes. Η εντολή LDS χρησιμοποιεί τον καταχωρητή RAMPZ για να προσπελάσει μνήμη μεγαλύτερη των 64K bytes.

Λειτουργία:
(i) Rd ← (k)

Σύνταξη:	**Τελεστές:**	**Μετρητής προγράμματος:**
(i) LDS Rd,k	0 ≤ d ≤ 31, 0 ≤ k ≤ 65535	PC ← PC + 2

Παράδειγμα:
```
lds r2,$FF00 ; Load r2 with the contents of SRAM location
$FF00
add r2,r1 ; add r1 to r2
sts $FF00,r2 ; Write back
```
Words: 2 (4 bytes)
Cycles: 3

LPM - Load Program Memory

Περιγραφή:

Φορτώνει ένα byte που καθορίζεται από τον Z καταχωρητή στον καταχωρητή 0 (R0). Η μνήμη προγράμματος είναι οργανωμένη σε λέξεις των 16 bits και το L.S.Bit του δείκτη Z (16 bits) κάνει επιλογή ανάμεσα στο low byte (0) ή στο high byte (1) της λέξης της μνήμης προγράμματος. Αυτή η εντολή μπορεί να δώσει διεύθυνση στα πρώτα 64K bytes (32K words) της μνήμης προγράμματος.

Λειτουργία:	**Σχόλιο:**
(i) R0 ← (Z)	Z points to program memory

Σύνταξη:	**Τελεστές:**	**Μετρητής προγράμματος:**
(i) LPM	None	PC ← PC + 1

Παράδειγμα:

```
clr r31 ; Clear Z high byte
ldi r30,$F0 ; Set Z low byte
lpm ; Load constant from program
; memory pointed to by Z (r31:r30)
```

Words: 1 (2 bytes)
Cycles: 3

ST - Store Indirect From Register to SRAM using Index X

Περιγραφή:
Αποθηκεύει ένα byte έμμεσα από έναν καταχωρητή στην SRAM. Η διεύθυνση της SRAM καθορίζεται από τον X (16 bits) καταχωρητή. Η πρόσβαση στην μνήμη είναι περιορισμένη στην τρέχουσα σελίδα της SRAM των 64K bytes. Για να υπάρχει πρόσβαση σε μία άλλη σελίδα της SRAM, ο RAMPX καταχωρητής στην περιοχή I/O πρέπει να αλλάξει. Ο X καταχωρητής μπορεί να παραμείνει ο ίδιος μετά την λειτουργία, ή μπορεί να αυξηθεί ή να μειωθεί. Αυτά τα χαρακτηριστικά ταιριάζουν ιδιαίτερα για χρήση σαν stack pointer του καταχωρητή X.

Using the X pointer:

Λειτουργία:	**Σχόλιο:**
(i) (X) ← Rr	X: Unchanged
(ii) (X) ← Rr X ← X+1	X: Post incremented
(iii) X ← X - 1 (X) ← Rr	X: Pre decremented

Σύνταξη:	**Τελεστές:**	**Μετρητής προγράμματος:**
(i) ST X, Rr	$0 \le r \le 31$	PC ← PC + 1
(ii) ST X+, Rr	$0 \le r \le 31$	PC ← PC + 1
(iii) ST -X, Rr	$0 \le r \le 31$	PC ← PC + 1

Παράδειγμα:

```
clr r27 ; Clear X high byte
ldi r26,$20 ; Set X low byte to $20
st X+,r0 ; Store r0 in SRAM loc. $20(X post inc)
st X,r1 ; Store r1 in SRAM loc. $21
ldi r26,$23 ; Set X low byte to $23
st X,r2 ; Store r2 in SRAM loc. $23
```

```
st –X,r3 ; Store r3 in SRAM loc. $22(X pre dec)
```
Words: 1 (2 bytes)
Cycles: 2

ST (STD) - Store Indirect From Register to SRAM using Index Y

Περιγραφή:
Αποθηκεύει ένα byte έμμεσα από ένα καταχωρητή στην SRAM. Η διεύθυνση της SRAM καθορίζεται από τον Y (16 bits) καταχωρητή. Η πρόσβαση στην μνήμη είναι περιορισμένη στην τρέχουσα σελίδα της SRAM των 64K bytes. Για να υπάρχει πρόσβαση σε μία άλλη σελίδα της SRAM, ο RAMPY καταχωρητής στην περιοχή I/O πρέπει να αλλάξει. Ο Y καταχωρητής μπορεί να παραμείνει ο ίδιος μετά την λειτουργία, ή μπορεί να αυξηθεί ή να μειωθεί. Αυτά τα χαρακτηριστικά ταιριάζουν ιδιαίτερα για χρήση σαν stack pointer του καταχωρητή Y.

Using the Y pointer:

Λειτουργία:	**Σχόλιο:**
(i) (Y) ← Rr	Y: Unchanged
(ii) (Y) ← Rr, Y ← Y+1	Y: Post incremented
(iii) Y ← Y – 1, (Y) ← Rr	Y: Pre decremented
(iiii) (Y+q) ← Rr	Y: Unchanged, q: Displacement

Σύνταξη:	**Τελεστές:**	**Μετρητής προγράμματος:**
(i) ST Y, Rr	$0 \le r \le 31$	PC ← PC + 1
(ii) ST Y+, Rr	$0 \le r \le 31$	PC ← PC + 1
(iii) ST -Y, Rr	$0 \le r \le 31$	PC ← PC + 1
(iiii) STD Y+q, Rr	$0 \le r \le 31$, $0 \le q \le 63$	PC ← PC + 1

Παράδειγμα:
```
clr r29 ; Clear Y high byte
ldi r28,$20 ; Set Y low byte to $20
st Y+,r0 ; Store r0 in SRAM loc. $20(Y post inc)
st Y,r1 ; Store r1 in SRAM loc. $21
ldi r28,$23 ; Set Y low byte to $23
st Y,r2 ; Store r2 in SRAM loc. $23
st –Y,r3 ; Store r3 in SRAM loc. $22(Y pre dec)
std Y+2,r4 ; Store r4 in SRAM loc. $24
```
Words: 1 (2 bytes)
Cycles: 2

ST (STD) - Store Indirect From Register to SRAM using Index Z

Περιγραφή:
Αποθηκεύει ένα byte έμμεσα από ένα καταχωρητή στην SRAM. Η διεύθυνση της SRAM καθορίζεται από τον Z (16 bits) καταχωρητή. Η πρόσβαση στην μνήμη είναι περιορισμένη στην τρέχουσα σελίδα της SRAM των 64K bytes. Για να υπάρχει πρόσβαση σε μία άλλη σελίδα της SRAM, ο RAMPZ καταχωρητής στην περιοχή I/O πρέπει να αλλάξει. Ο Z καταχωρητής μπορεί να παραμείνει ο ίδιος μετά την λειτουργία, ή μπορεί να αυξηθεί ή να μειωθεί. Αυτά τα χαρακτηριστικά ταιριάζουν ιδιαίτερα για χρήση σαν stack pointer του Z καταχωρητή, αλλά επειδή ο Z

καταχωρητής μπορεί να χρησιμοποιηθεί για έμμεσες κλήσεις υπορουτινών, έμμεσες υπερπηδήσεις και εξέταση πινάκων είναι συχνά πιο βολικό να χρησιμοποιείται ο X ή ο Y pointer σαν ένας stack pointer.

Using the Z pointer:

Λειτουργία:	**Σχόλιο:**
(i) (Z) ←Rr	Z:Unchanged
(ii) (Z) ← Rr, Z ← Z+1	Z: Post incremented
(iii) Z ← Z – 1, (Z) ← Rr	Z: Pre decremented
(iiii) (Z+q) ← Rr	Z: Unchanged, q: Displacement

Σύνταξη:	**Τελεστές:**	**Μετρητής προγράμματος:**
(i) ST Z, Rr	0 ≤ r ≤ 31	PC ← PC + 1
(ii) ST Z+, Rr	0 ≤ r ≤ 31	PC ← PC + 1
(iii) ST -Z, Rr	0 ≤ r ≤ 31	PC ← PC + 1
(iiii) STD Z+q, Rr	0 ≤ r ≤ 31, 0 ≤ q ≤ 63	PC ← PC + 1

Παράδειγμα:

```
clr r31 ; Clear Z high byte
ldi r30,$20 ; Set Z low byte to $20
st Z+,r0 ; Store r0 in SRAM loc. $20(Z post inc)
st Z,r1 ; Store r1 in SRAM loc. $21
ldi r30,$23 ; Set Z low byte to $23
st Z,r2 ; Store r2 in SRAM loc. $23
st -Z,r3 ; Store r3 in SRAM loc. $22(Z pre dec)
std Z+2,r4 ; Store r4 in SRAM loc. $24
```

Words: 1 (2 bytes)
Cycles: 2

STS - Store Direct to SRAM

Περιγραφή:
Αποθηκεύει ένα byte από ένα καταχωρητή στην SRAM. Μία 16-bit διεύθυνση πρέπει να δοθεί. Η πρόσβαση στην μνήμη είναι περιορισμένη στην τρέχουσα σελίδα της SRAM των 64K bytes. Η εντολή STS χρησιμοποιεί τον RAMPZ καταχωρητή για να έχει πρόσβαση σε μνήμη μεγαλύτερη των 64K bytes.

Λειτουργία:
(i) (k) ← Rr

Σύνταξη:	**Τελεστές:**	**Μετρητής προγράμματος:**
(i) STS k,Rr	0 ≤ r ≤ 31, 0 ≤ k ≤ 65535	PC ← PC + 2

Παράδειγμα:

```
lds r2,$FF00 ; Load r2 with the contents of SRAM location
$FF00
add r2,r1 ; add r1 to r2
sts $FF00,r2 ; Write back
```

Words: 2 (4 bytes)
Cycles: 3