

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΠΕΞΕΡΓΑΣΤΗ 8085



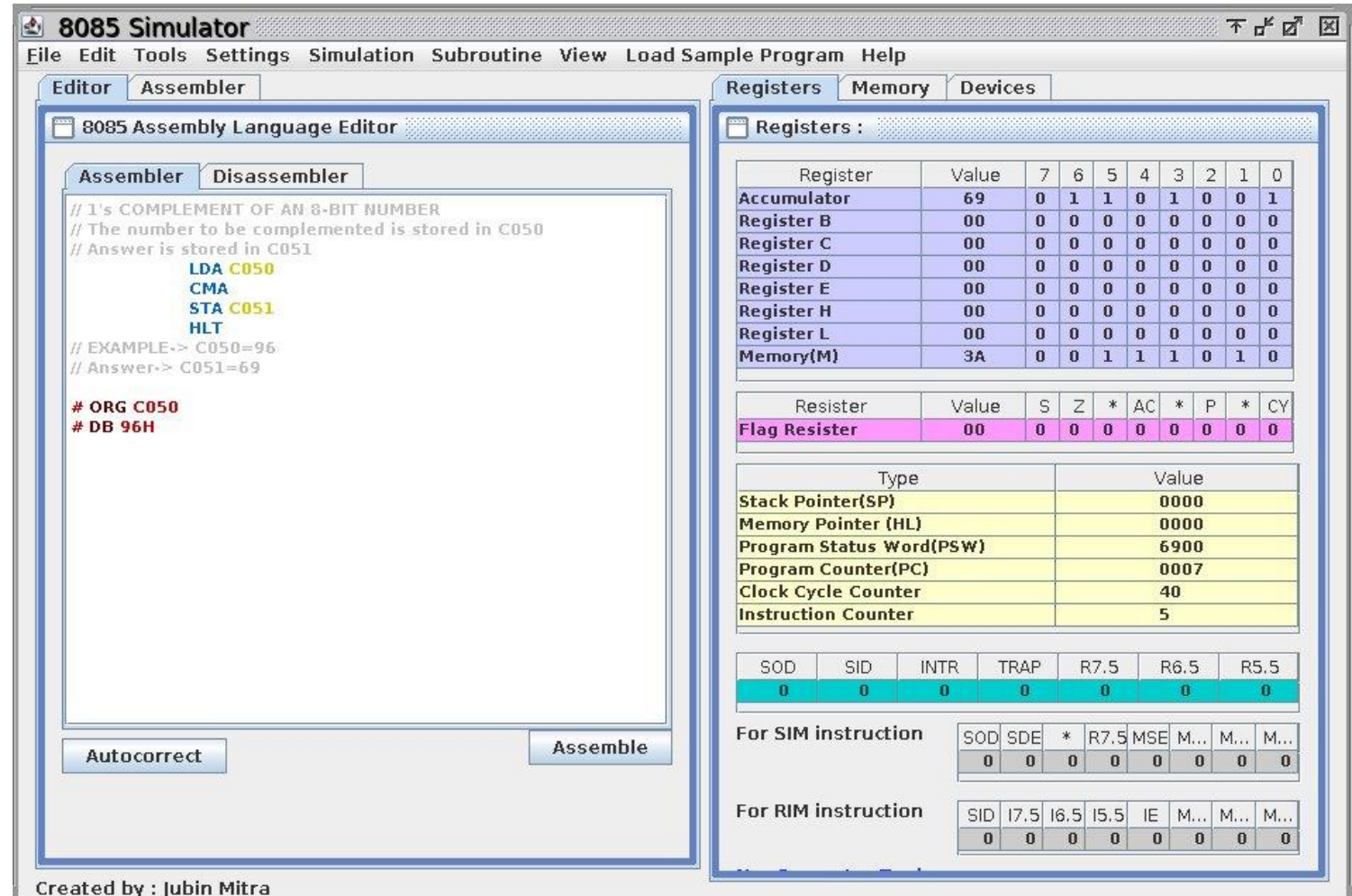
ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΎΠΟΛΟΓΙΣΤΩΝ
ΦΡΟΝΤΙΣΤΗΡΙΑΚΗ ΔΙΑΛΕΞΗ 1

ΠΡΟΣΟΜΟΙΩΣΗ 8085

- Προτεινόμενα προγράμματα:
 - 8085 Simulator by JUBIN MITRA
 - <https://github.com/8085simulator/8085simulator.github.io>
 - Απλό jar, δεν χρειάζεται εγκατάσταση παίζει αν υπάρχει JAVA
 - TSIK Microlab Simulator
 - Έχει ανέβει στο Helios
 - Χρειάζεται παλιά version του .NET παίζει μέχρι Windows 10 γίνεται προσπάθεια εκσυγχρονισμού του

8085 SIMULATOR BY JUBIN MITRA

- Simulates Registers, I/Os, Delays, Interrupts, Memory
- Detailed documentation



8085 SIMULATOR BY JUBIN MITRA

- Assembler Editor
 - Disassembler Editor
 - Assembler Workspace
 - Memory Editor
 - I/O Editor
 - Interrupt Editor
 - Debugger
 - Simulator
 - Helper
- Printing
 - Assembler Content
 - Register Bank
 - Flag Register
 - Stack Pointer (SP)
 - Memory Pointer (HL)
 - Program Counter (PC)
 - Clock Cycle Counter
 - Special blocks for monitoring Flag register and the usage of SIM and RIM instruction

8085 SIMULATOR BY JUBIN MITRA

- Helper
 - Code wizard tool
 - Examples
 - User manual
- Insert DELAY Subroutine TOOL
 - Helps generating delay subroutine with user defined delay
- Interrupt Service Subroutine TOOL
 - Set memory values at corresponding vector interrupt address
- Number Conversion Tool
 - Interconversion tool for Hexadecimal, decimal and binary numbers

ΕΠΑΝΑΛΗΨΗ ΟΡΙΣΜΩΝ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ

- Ποιες είναι οι τρεις κύριες μονάδες ενός ψηφιακού υπολογιστή;
 - η κεντρική μονάδα επεξεργασίας (CPU)
 - η μονάδα μνήμης
 - Οι συσκευές εισόδου/εξόδου (I/O)
- Ποιες είναι οι διαφορετικές εργασίες που κάνει η CPU σε κάθε δεδομένο χρονική στιγμή;
 - Ανάγνωση ή εγγραφή στη μνήμη
 - Ανάγνωση ή εγγραφή I/O
 - Μια εσωτερική δραστηριότητα

ΕΠΑΝΑΛΗΨΗ ΟΡΙΣΜΩΝ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ

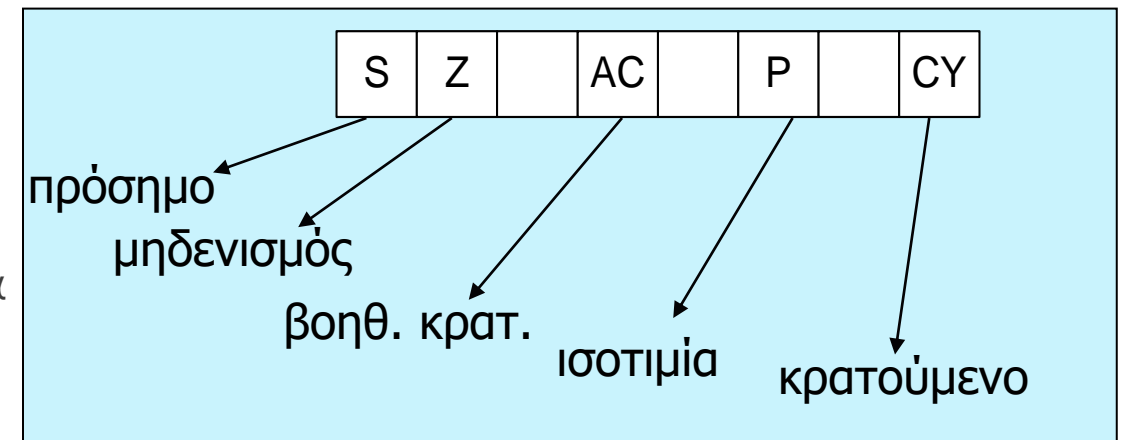
- Ένα πρόγραμμα είναι γραμμένο είτε σε δυαδικό είτε σε δεκαεξαδικό κώδικα (γλώσσα μηχανής).
 - Συμβολικός κώδικας για κάθε εντολή για ευκολία (γλώσσα assembly).
- Τι είναι «μήκος λέξης» (word length)?
 - Ο αριθμός των bit που ένας υπολογιστής αναγνωρίζει και μπορεί να επεξεργαστεί κάθε φορά
- Πώς μεταφράζεται η assembly σε γλώσσα μηχανής?
 - Ή «με το χέρι» ή με χρήση του assembler προγράμματος

ΚΑΤΑΧΩΡΗΤΕΣ 8085

- Αρχιτεκτονική 8085 - 8bit
- Instruction set 74 εντολών
- Καταχωρητές των 8bits
 - A (accumulator), B, C, D, E, H, L
 - Ζεύγη καταχωρητών B-C, D-E, H-L
 - Stack pointer SP
 - Program Counter (PC)
- Accumulator: είναι ο πιο σημαντικός από όλους τους καταχωρητές του 8085. Οποιαδήποτε δεδομένα, είσοδος/έξοδος προς/από τον μικροεπεξεργαστή πραγματοποιείται μέσω αυτού. Ο καταχωρητής συσσωρευτή χρησιμοποιείται ευρέως για αριθμητική, λογική, αποθήκευση και λειτουργίες περιστροφής.
- Το ζεύγος καταχωρητών HL μπορεί να χρησιμοποιηθεί ως δείκτης δεδομένων ή δείκτης μνήμης.

Ο ΚΑΤΑΧΩΡΗΤΗΣ ΣΗΜΑΙΩΝ

- (S): – Εάν το MSB του αποτελέσματος μιας λειτουργίας είναι 1, αυτή η σημαία είναι 1, διαφορετικά μηδενίζεται.
- (Z): – Εάν το αποτέλεσμα μιας εντολής είναι μηδέν, αυτή η σημαία γίνεται 1, διαφορετικά 0.
- (AC): – Εάν υπάρχει μεταφορά του bit 3 και στο bit 4 ως αποτέλεσμα την εκτέλεση μιας αριθμητικής πράξης, γίνεται 1 διαφορετικά μηδενίζεται.
- (CY): – Εάν μια οδηγία έχει ως αποτέλεσμα μεταφορά (για προσθήκη) ή δανεισμό (για αφαίρεση ή σύγκριση) από το bit D7, τότε αυτή η σημαία είναι 1, διαφορετικά 0.
- (P): – Αυτή η σημαία ορίζεται όταν το αποτέλεσμα μιας λειτουργίας περιέχει ζυγό αριθμό 1 και μηδενίζεται διαφορετικά.



ΠΑΡΑΔΕΙΓΜΑ 1

- Δίνεται αριθμός των 8 bit στη θέση μνήμης 0040H. Να βρεθούν τα δεκαεξαδικά ψηφία υψηλότερης και χαμηλότερης τάξης (σε δυαδική παράσταση) και να αποθηκευτούν στις θέσεις 0041H και 0042H αντίστοιχα.

0040: X Y

0041: 0 X

0042: 0 Y

ΠΑΡΑΔΕΙΓΜΑ 1 - ΛΥΣΗ

LXI H,0040

MOV A,M

MOV B,A

RRC

RRC

RRC

RRC

ANI 0F

INX H

MOV M,A

MOV A,B

ANI 0F

INX H

MOV M,A

HLT

; Κρατώ αντίγραφο

; 4 περιστροφές δεξιά

; Το υψηλότερης αξίας Hex ψηφίο

; πάει στη χαμηλότερη θέση

; Απομόνωση X, 0F16 = 00001111

; Αποθηκεύω το X στη θέση 0041H

; Επαναφέρω

; Απομονώνω το χαμηλότερης

; Hex αξίας ψηφίο Y

; Αποθηκεύω το Y στη θέση 0042H

; Τέλος

0040: X Y

0041: 0 X

0042: 0 Y

ΠΑΡΑΔΕΙΓΜΑ 1 – ΕΝΤΟΛΕΣ 1

- Μετακίνηση δεδομένων – Φόρτωση απ' ευθείας δεδομένου 16 bit σε ζεύγος καταχωρητών

- Εντολή **LXI** (**L**oad **R**egister **P**air **I**mmEDIATE)

LXI **RP, Data**

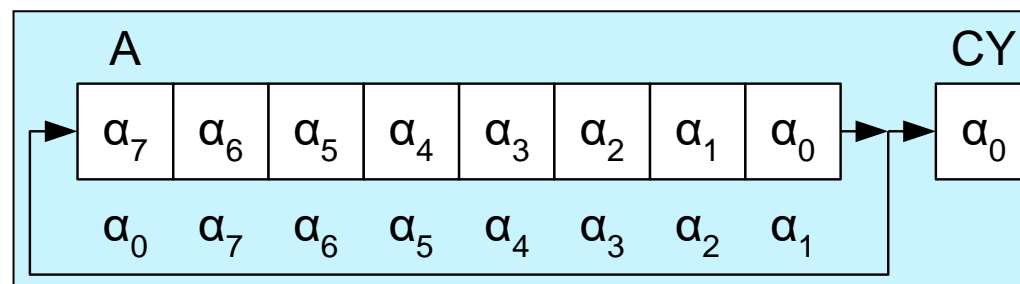
- Μετακίνηση δεδομένων – Μεταξύ καταχωρητών και μνήμης

- Εντολή **MOV M, K**

MOV **M, K** (από Καταχωρητή σε Μνήμη)

- Εντολή δεξιάς περιστροφής

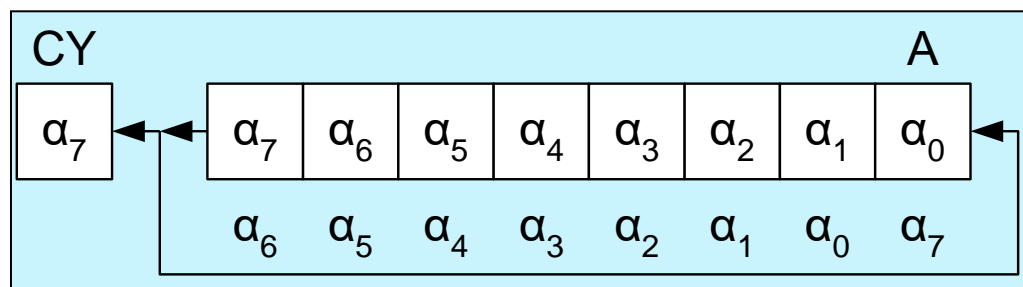
RRC



1 κύκλος

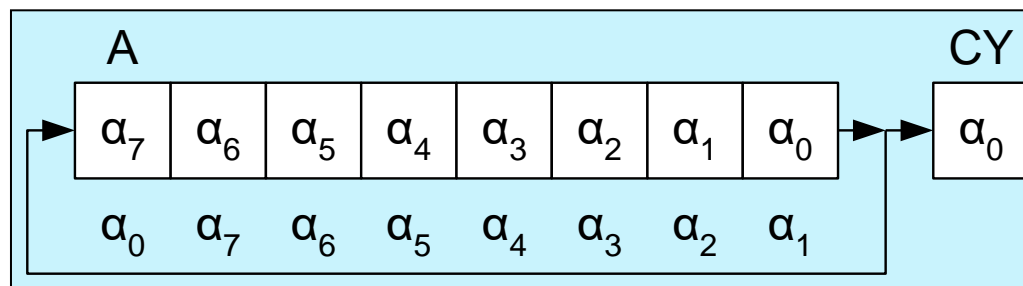
ΕΝΤΟΛΕΣ ΠΕΡΙΣΤΡΟΦΗΣ

RLC



1 κύκλος

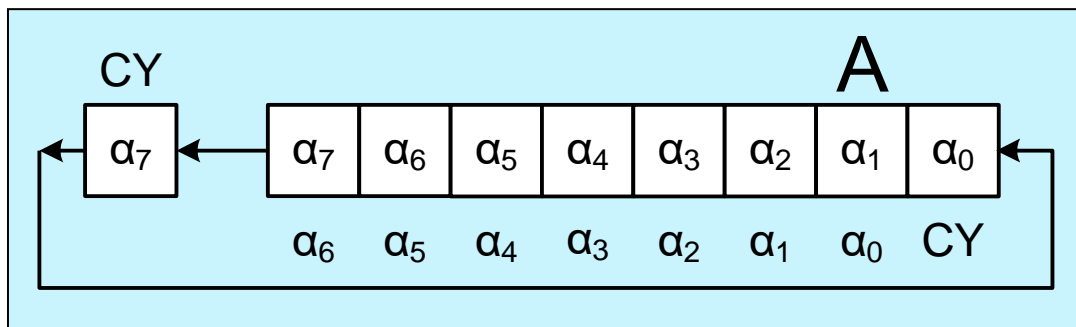
RRC



1 κύκλος

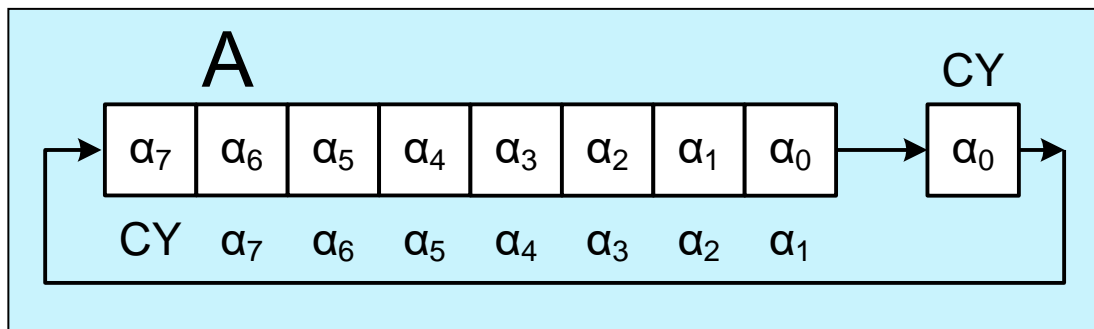
ΕΝΤΟΛΕΣ ΠΕΡΙΣΤΡΟΦΗΣ Μ'ΕΣΩ ΚΡΑΤΟΥΜ'ΕΝΟΥ

RAL



1 κύκλος

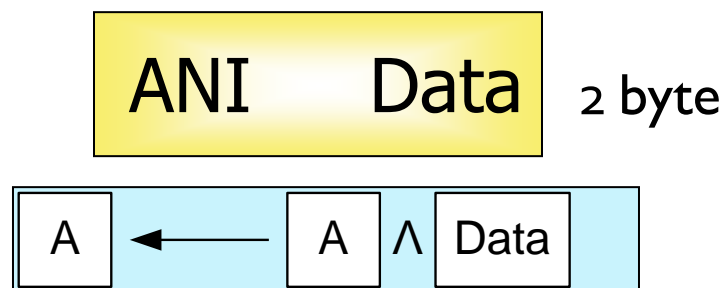
RAR



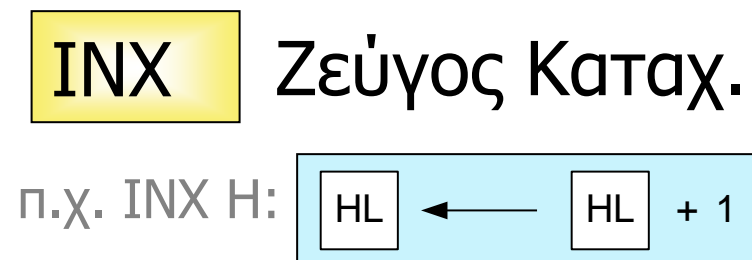
1 κύκλος

ΠΑΡΑΔΕΙΓΜΑ 1 – ΕΝΤΟΛΕΣ 2

- Πράξη AND – Με αριθμό

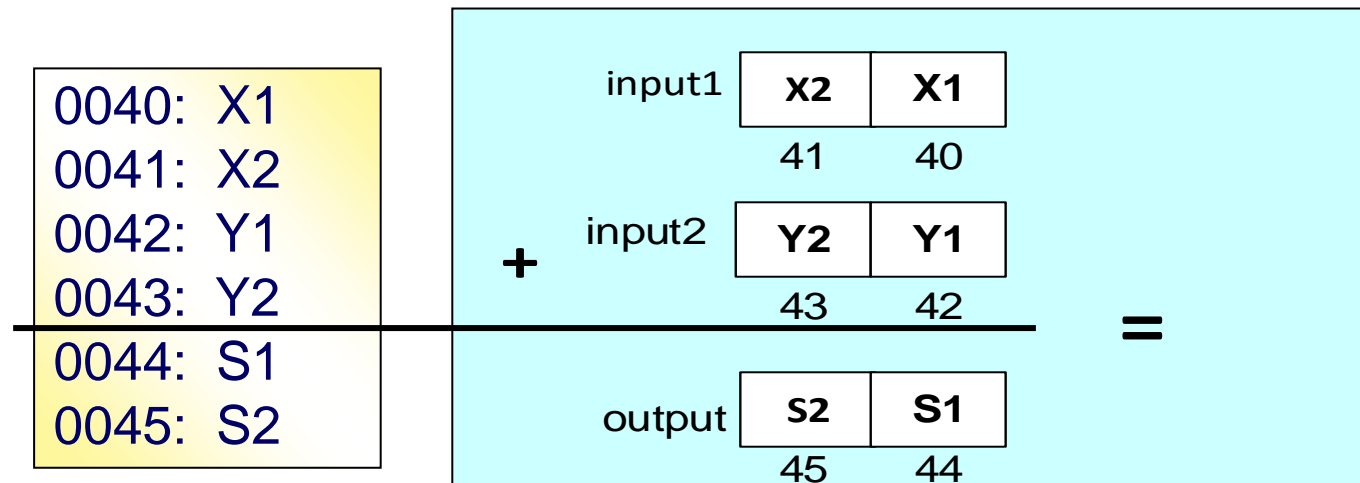


- Αύξηση ζεύγους καταχωρητών (INX)



ΠΑΡΑΔΕΙΓΜΑ 2

- Αριθμός των 16 bits που βρίσκεται στις θέσεις 40 και 41 της μνήμης να προστεθεί στον αριθμό των 16 bits που βρίσκεται στις θέσεις 42 και 43 της μνήμης. Το αποτέλεσμα να τοποθετηθεί στις θέσεις 44 και 45. Τα περισσότερα σημαντικά bytes βρίσκονται στις θέσεις 41, 43 και 45. Να βρεθεί και δεύτερος τρόπος.



ΠΑΡΑΔΕΙΓΜΑ 2 – ΛΥΣΗ 1

LXI H,0040H

MOV A,M ; (A) = X1

INX H

INX H ; (HL) = 0042H

ADD M ; Προσθέτουμε τα λιγότερο
σημαντικά bytes X1+Y1=S1

MOV B,A ; (B) = S1

DCX H ; (HL) = 0041H

MOV A,M ; (A) = X2

INX H

INX H ; (HL) = 0043H

ADC M ; Προσθ. τα περισσότερα
σημαντικά bytes X2+Y2+CY

INX H ; (HL) = 0044H

MOV M,B ; Αποθήκευση S1

INX H ; (HL) = 0045H

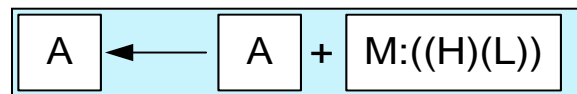
MOV M,A ; Αποθήκευση
αποτελέσματος S2

HLT ; Τέλος προγράμματος

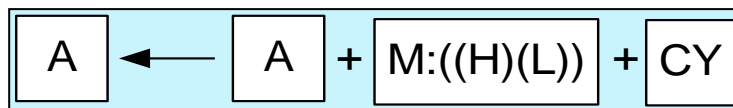
ΠΑΡΑΔΕΙΓΜΑ 2 – ΕΝΤΟΛΕΣ 1

- Αριθμητικές εντολές – Πρόσθεση δεδομένου από τη μνήμη στον καταχωρητή A

ADD M



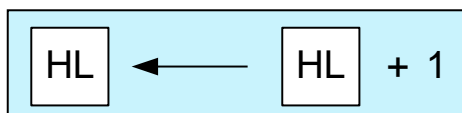
ADC M



- Αριθμητικές Εντολές – Εντολές που επιδρούν σε ζεύγη καταχωρητών

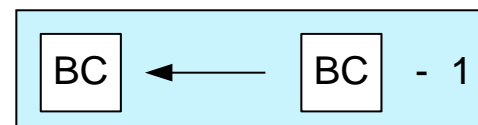
INX

π.χ. INX H:

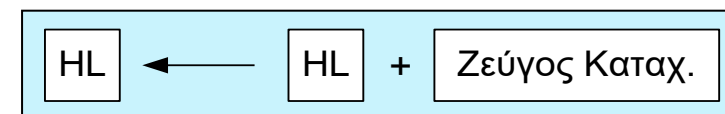


DCX

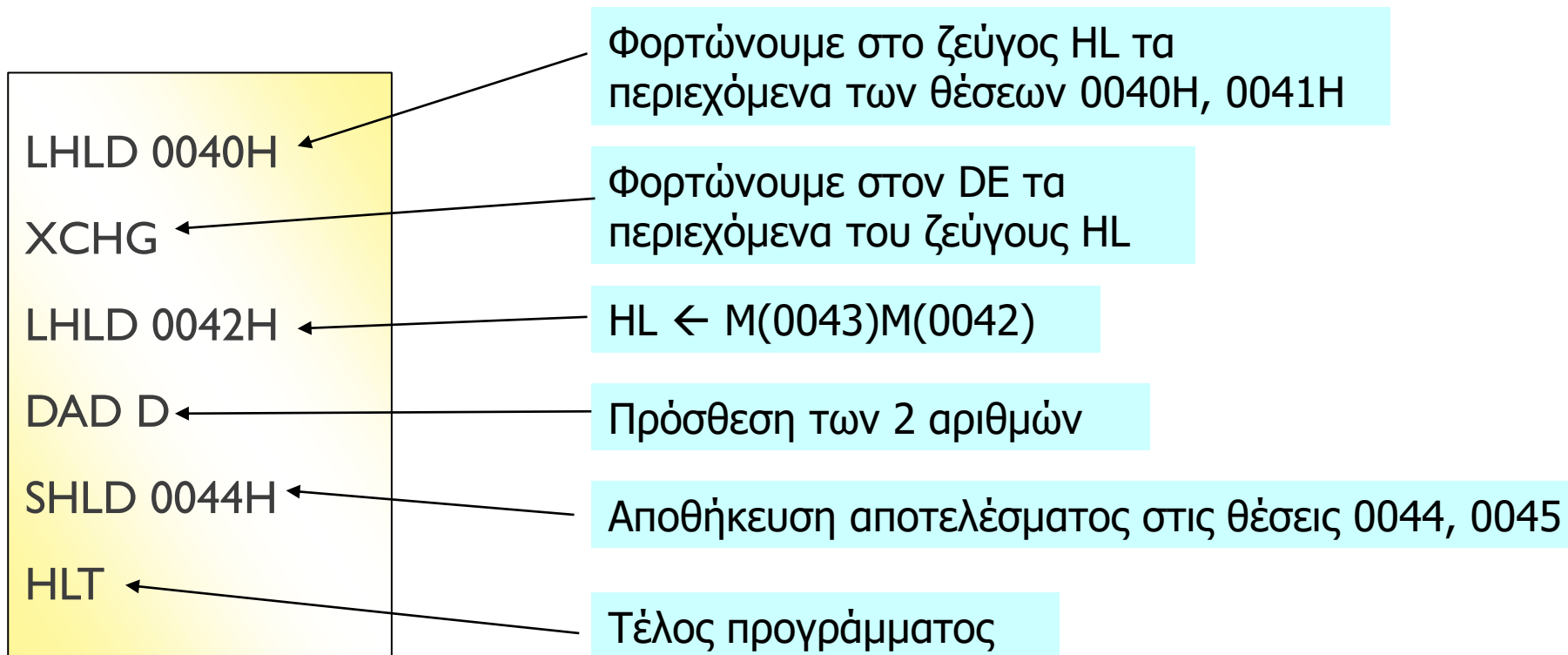
π.χ. DCX B:



DAD



ΠΑΡΑΔΕΙΓΜΑ 2 – ΛΥΣΗ 2

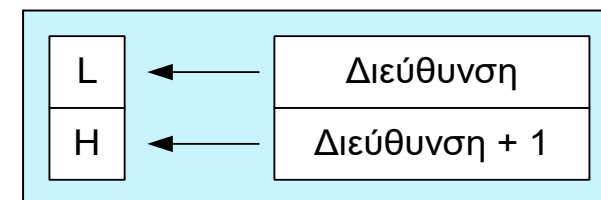


ΠΑΡΑΔΕΙΓΜΑ 2 – ΕΝΤΟΛΕΣ 2

- Μετακίνηση Δεδομένων από/προς ζεύγος H-L

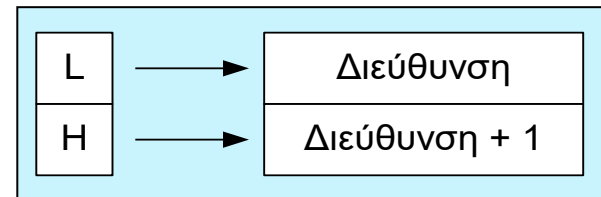
LHLD Διεύθυνση (2 bytes)

- 5 κύκλοι
- 16 Καταστάσεις



SHLD Διεύθυνση (2 bytes)

- 5 κύκλοι
- 16 Καταστάσεις



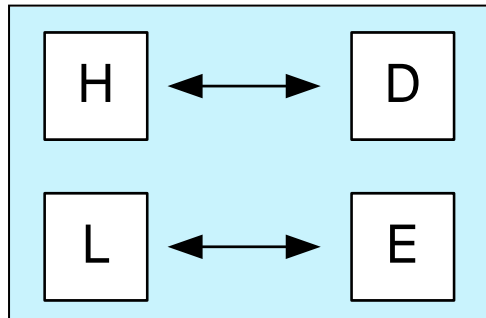
ΠΑΡΑΔΕΙΓΜΑ 2 – ΕΝΤΟΛΕΣ 3

- Μετακίνηση δεδομένων μεταξύ ζευγών καταχωρητών

XCHG (Χωρίς όρισμα)

- 1 κύκλος
- 4 Καταστάσεις

1 1 1 0 1 0 1 1 opcode

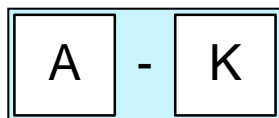


Αμοιβαία ανταλλαγή
περιεχομένου των
καταχωρητών HL και DE

ΣΥΓΚΡΙΣΕΙΣ

- Εντολές Σύγκρισης (γίνεται αφαίρεση αλλά το αποτέλεσμα δεν καταχωρείται, το περιεχόμενο του A δεν αλλάζει, παραμένει όμως η ενημέρωση των σημαιών)

CMP Καταχωρητής



- ⊕ CY=1 αν $A < K$
- ⊕ CY=0 αν $A \geq K$
- ⊕ Z=1 αν $A = K$

1 κύκλος

CMP M

Σύγκριση με δεδομένο στη μνήμη

2 κύκλοι

CPI Data 2 byte

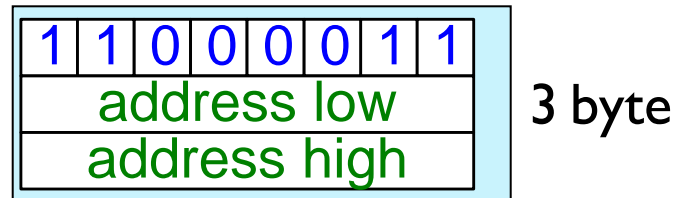
Σύγκριση με άμεσο δεδομένο

2 κύκλοι

ΕΝΤΟΛΕΣ ΑΛΜΑΤΟΣ

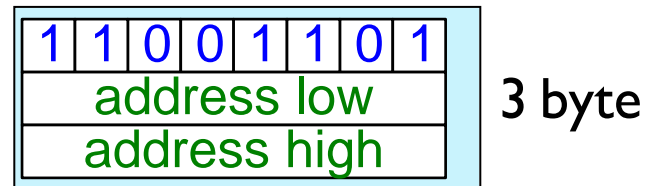
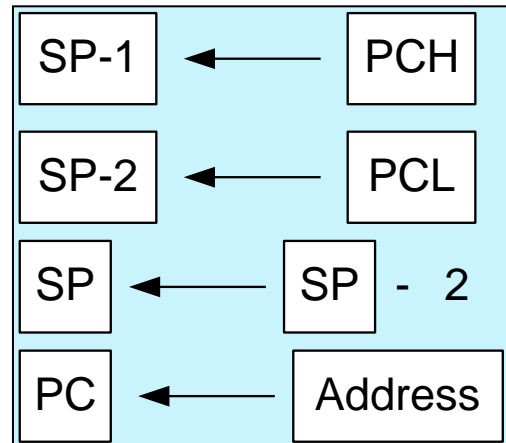
- Εντολές άλματος – Άνευ συνθήκης

JMP Address (2 bytes)



3 κύκλοι

CALL Address (2 bytes)



5 κύκλοι

Κλήση ρουτίνας
Επιστροφή με **RET**

ΕΝΤΟΛΕΣ ΑΛΜΑΤΟΣ 2

- Εντολές άλματος υπό συνθήκη:

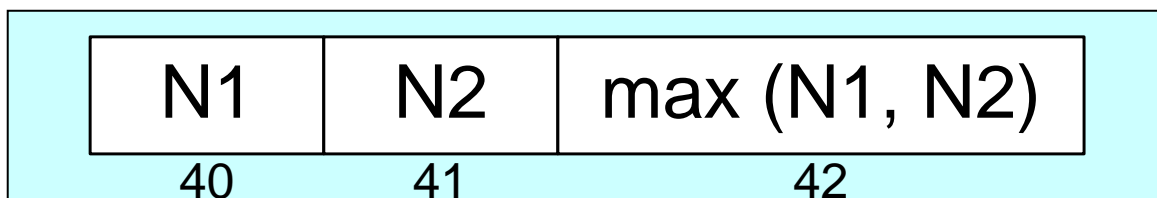
JNZ Z=0	JNC CY=0	JPO P=0	JP S=0	Address	2 ή 3 κύκλοι
JZ Z=1	JC CY=1	JPE P=1	JM S=1	Address	

Συνθήκη (που όταν ισχύει γίνεται άλμα)

- **NZ** – Μη μηδενικό αποτελ. (Z=0)
- **NC** – Όχι κρατούμενο (CY=0)
- **PO** – Ισοτιμία περιττή (P=0)
- **P** – Θετικό αποτέλεσμα (S=0)
- **Z** – Μηδενικό αποτέλεσμα (Z=1)
- **C** – Κρατούμενο (CY=1)
- **PE** – Ισοτιμία άρτια (P=1)
- **M** – Αρνητικό αποτέλεσμα (S=1)

ΠΑΡΑΔΕΙΓΜΑ 3

- Τα περιεχόμενα των θέσεων 0040H και 0041H της μνήμης είναι απλοί δυαδικοί αριθμοί (χωρίς πρόσημο). Να προσδιοριστεί ο μεγαλύτερος από τους δύο αριθμούς και να αποθηκευτεί στη θέση 0042H.



0040: N1
0041: N2
0042: max

ΠΑΡΑΔΕΙΓΜΑ 3 - ΛΥΣΗ

LXI H, 0040H

MOV A,M

INX H

CMP M ; Συγκρίνονται οι 2 αριθμοί

JNC DONE ; Άλμα αν ο 1ος είναι μεγαλύτερος

; N1(A) > N2(M) => CY=0

MOV A,M ; διαφορετικά (N1 < N2) προετοιμάζεται ο 2ος

DONE:

INX H ; (H)(L) ← 0042H

MOV M,A ; Αποθήκευση μεγαλύτερου στη 0042H

HLT

ΠΑΡΑΔΕΙΓΜΑ 4

- Στο πρόγραμμα αυτό προσδιορίζεται το μεγαλύτερο ενός συνόλου αριθμών. Το πλήθος των αριθμών βρίσκεται στη θέση 0041H της μνήμης και οι αριθμοί αρχίζουν από τη θέση 0042H. Το αποτέλεσμα καταχωρείται στη θέση 0040H.

```
0040: max  
0041: n  
0042: x1  
0043: x2  
...  
...  
: xn
```

ΠΑΡΑΔΕΙΓΜΑ 4 - ΛΥΣΗ

```
LXI H, 0041H
MOV B,M      ; (B) ← πλήθος αριθμών
SUB A        ; Μηδενίζεται ο A (A = A-A)
FOR:
INX H
CMP M        ; Είναι ο επόμενος αριθμός > A
JNC NEXT     ; δηλ. του τοπικού μεγίστου;
MOV A,M      ; Ναι, αντικαθίσταται ο μέγιστος.
NEXT:
DCR B        ; Αλλιώς προχωράμε στον έλεγχο
JNZ FOR      ; του επόμενου αριθμού
STA 0040H
HLT
```

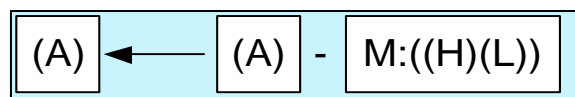
ΠΑΡΑΔΕΙΓΜΑ 4 – ΕΝΤΟΛΕΣ 1

SUB Καταχωρητής

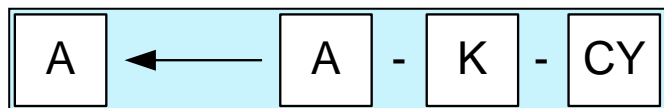


- Αριθμητικές Εντολές – Αφαίρεση με ή χωρίς δανεικό

SUB M

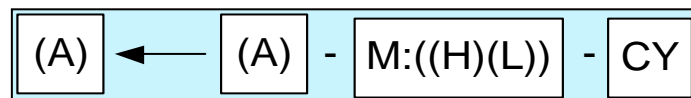


SBB Καταχωρητής



1 κύκλος

SBB M



2 κύκλοι

ΠΑΡΑΔΕΙΓΜΑ 4 – ΕΝΤΟΛΕΣ 2

- Αριθμητικές Εντολές - Εντολές αυξομείωσης κατά 1 – (Αντίστοιχα INX και DCX για ζεύγη!)

INR Καταχωρητής

$(K) \leftarrow (K) + 1$

DCR Καταχωρητής

$(K) \leftarrow (K) - 1$

1 κύκλος

INR M

$((H)(L)) \leftarrow ((H)(L)) + 1$

DCR M

$((H)(L)) \leftarrow ((H)(L)) - 1$

ΠΑΡΑΔΕΙΓΜΑ 4 – ΕΝΤΟΛΕΣ 2

- Αριθμητικές Εντολές - Εντολές αυξομείωσης κατά 1 – (Αντίστοιχα INX και DCX για ζεύγη!)

INR Καταχωρητής

$(K) \leftarrow (K) + 1$

DCR Καταχωρητής

$(K) \leftarrow (K) - 1$

1 κύκλος

INR M

$((H)(L)) \leftarrow ((H)(L)) + 1$

DCR M

$((H)(L)) \leftarrow ((H)(L)) - 1$

(flag Z):– Εάν το αποτέλεσμα μιας εντολής είναι μηδέν, αυτή η σημαία γίνεται 1, διαφορετικά 0.

ΠΑΡΑΔΕΙΓΜΑ 4 - ΛΥΣΗ

LXI H, 0041H

MOV B,M ; (B) \leftarrow πλήθος αριθμών

SUB A ; Μηδενίζεται ο A ($A = A - A$)

FOR:

INX H

CMP M ; Είναι ο επόμενος αριθμός $> A$

JNC NEXT ; δηλ. του τοπικού μεγίστου;

MOV A,M ; Ναι, αντικαθίσταται ο μέγιστος.

NEXT:

DCR B ; Αλλιώς προχωράμε στον έλεγχο

JNZ FOR ; του επόμενου αριθμού

STA 0040H

HLT

(flag Z):— Εάν το αποτέλεσμα μιας εντολής είναι μηδέν, αυτή η σημαία γίνεται 1, διαφορετικά 0.