Αρχιτεκτονική Υπολογιστών

Ενότητα 7: Αποκωδικοποίηση Εντολής x86

Δρ. Μηνάς Δασυγένης
mdasyg@ieee.org
Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών
http://arch.ece.uowm.gr/mdasyg

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Λειτουργία του assembler (1/3)

- Σε κάθε ISA ο προγραμματιστής χρησιμοποιεί μνημονικά *(εντολές assembly)* για να περιγράψει το πρόγραμμά του.
- Ο assembler διαβάζει αυτές τις εντολές και τις μετατρέπει σε MACHINE CODE, δηλαδή σε σειρά από Byte *(με '1' και '0')* στο δυαδικό σύστημα με μια αντιστοίχηση που έχει περιγραφεί από την εταιρία κατασκευής του επεξεργαστή.
- Αυτό γίνεται γιατί ο επεξεργαστής μπορεί και αποκωδικοποιεί μόνο 1 και 0 και όχι εντολές όπως SUB, ADD κτλ.

Η μορφή των εντολών της IA32

- Η μορφή των εντολών x86 αποτελείται από μια σειρά πεδίων μεταβλητού μεγέθους. Κάποια πεδία είναι προαιρετικά, κάποια πεδία απαιτούνται.

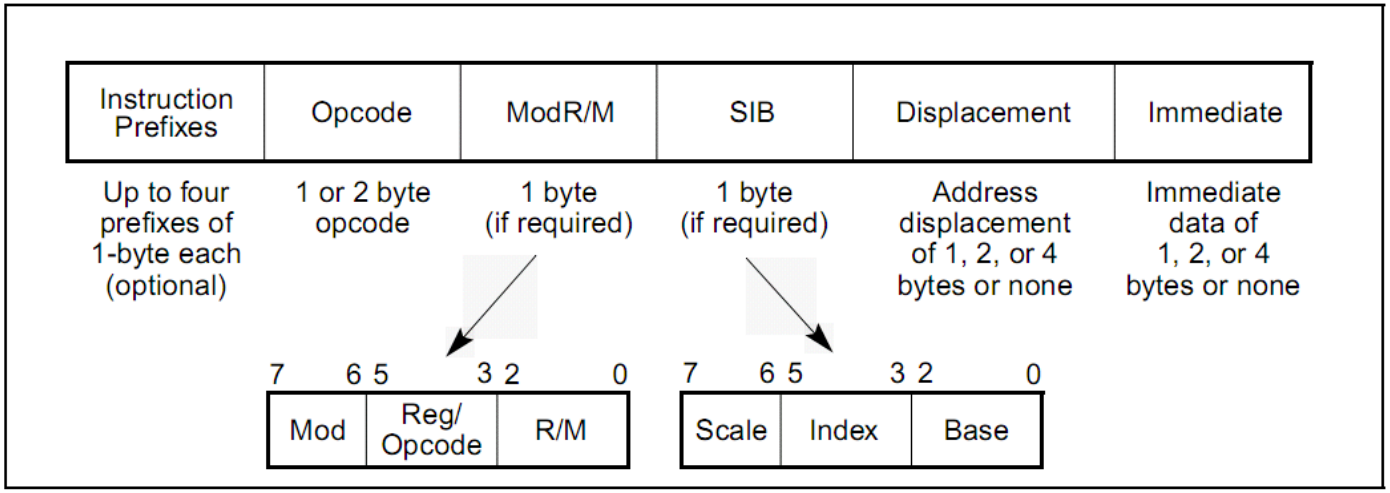
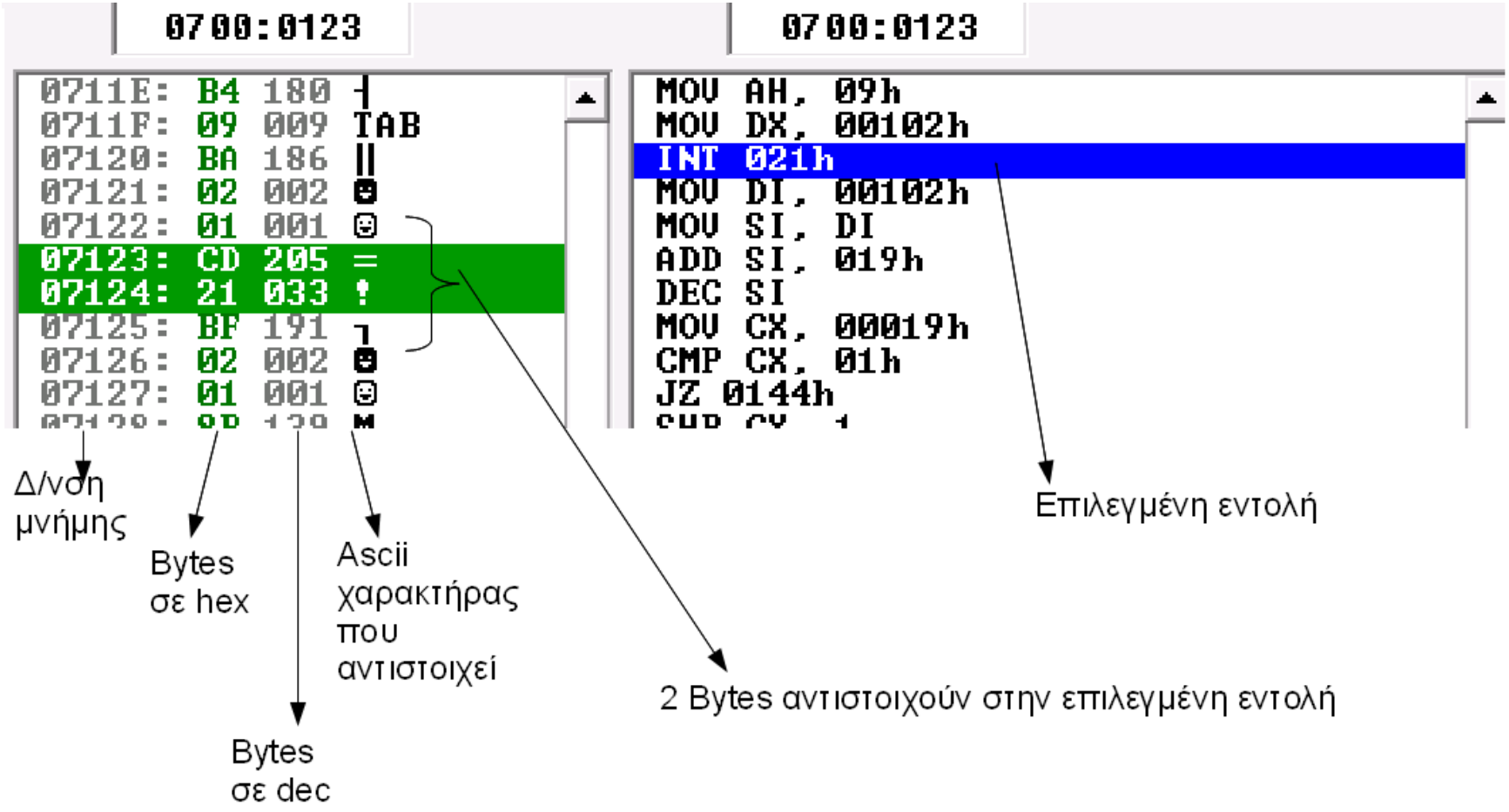


Figure 2-1. Intel Architecture Instruction Format

Λειτουργία του assembler (2/3)

- Για λόγους ευκολίας και οικονομίας χώρου η σειρά των 1 και 0 εμφανίζεται στο listing σε δεκαεξαδικό σύστημα.
- 2 ψηφία στο δεκαεξαδικό σύστημα έχουν μέγεθος 1 Byte.
- Υπάρχει αντιστοίχηση 1 προς 1 με μια εντολή assembly και τον αντίστοιχο machine code.
- Για παράδειγμα η εντολή **INT 21h** έχει το machine code **CD 21** *(2 Byte).*
- Αντιστρόφως, το **CD 21** αντιστοιχεί στην εντολή **INT 21h** *(αν το θεωρήσουμε ως εντολή).*
- Αν το CD 21 το θεωρήσουμε ως χαρακτήρες ASCII είναι ο χαρακτήρας “=” για CDh *(205 δεκαδικό)* και το “**!**” για 21h *(33 δεκαδικό).*

Λειτουργία του Assembler (3/3)



Αποκωδικοποίηση Εντολών

- Στις παρακάτω διαφάνειες θα εξετάσουμε πως μπορούμε να αποκωδικοποιούμε μια εντολή σε MACHINE CODE *(είτε στο δυαδικό, είτε στο δυαδικό, είτε στο δεκαεξαδικό σύστημα)* στην αντίστοιχη εντολή assembly.
- Το πρώτο βήμα είναι να έχουμε το Machine Code στη δεκαεξαδική μορφή, γιατί όλοι οι κατασκευαστές επεξεργαστών, **παρέχουν λίστες με τους machine code στη δεκαεξαδική μορφή.**
- Αν δεν έχουμε λοιπόν το machine code σε δεκαεξαδική μορφή το μετατρέπουμε σε αυτή.

  Π.χ. Το **`205 033`** το μετατρέπουμε σε **`CD 21`**

  Π.χ. Το **`11001110 00100001`** ομοίως σε **`CD 21`**

Machine code και Program Counter

- Έχουμε μια ακολουθία από Byte στη μνήμη.
- Το ζευγάρι CS:IP αντιστοιχεί στο program counter του προγράμματος.
- Το Program Counter (PC) δείχνει πάντα την επόμενη προς εκτέλεση εντολή. Όλοι οι επεξεργαστές έχουν είτε έναν καταχωρητή (PC) είτε δύο καταχωρητές *(για τη x86 τους CS και IP)* που δείχνουν ποια εντολή ΘΑ εκτελεστεί μετά την τρέχουσα εντολή.
- Το machine code είναι μεταβλητού μεγέθους στη x86.

  ➔ Αρχικά έρχεται το πρώτο byte και μόλις αποκωδικοποιηθεί ο επεξεργαστής θα φέρει και όσα ακόμη απαιτούνται.

Βασικές έννοιες του μετρητή προγράμματος

- **Πως βρίσκεται η επόμενη εντολή προς εκτέλεση;**
  - Program Counter (PC): Θέση μνήμης μέσα στον υπολογιστή που δείχνει τη διεύθυνση της θέσης μνήμης όπου περιέχεται η επόμενη εντολή
- **Πως αλλάζει τιμές ο μετρητής;**
  - Σειριακή αύξηση διεύθυνσης μετά την εκτέλεση εντολής. Όσα Byte είναι ο machine code τόσα αυξάνει.
  - Ή μεταπήδηση σε νέα θέση μνήμης *(διακλάδωση).* Αν υπάρχει εντολή διακλάδωσης τροποποιείται *(είτε μειώνεται είτε αυξάνεται)* με τη νέα διεύθυνση μνήμης.
- **Ποια είναι η αρχική του τιμή;**
  - Κατά την εκκίνηση του επεξεργαστή ο μετρητής προγράμματος έχει μια προκαθορισμένη τιμή. Συνήθως εκεί βρίσκεται το BIOS προκειμένου να εκτελεστούν τα διαγνωστικά.
- **Πότε σταματάει να αλλάζει τιμές;**
  - Συμβατικά, ποτέ!
  - Σε περιπτώσεις SLEEP για μείωση κατανάλωσης ενέργειας μπορεί να απενεργοποιηθεί *(αλλά δε γίνεται).*

Πως γίνεται το 'fetch'

- Για παράδειγμα αν το CS:IP δείχνει μια συγκεκριμένη θέση στη μνήμη η οποία έχει το machine code CD, θα έρθει το CD από τη μνήμη μέσα στον επεξεργαστή *(στον καταχωρητή εντολών Instruction Register)*.
- Ο μετρητής προγράμματος αυξάνει κατά 1 *(1 Byte).*
- Μόλις διαβαστεί το CD θα αποφασιστεί ότι η εντολή που ξεκινάει από CD έχει 2 Byte.
- Θα έρθει λοιπόν και το επόμενο Byte από τη μνήμη. Θα αυξηθεί ο μετρητής προγράμματος κατά 1.

Ερωτήσεις κατανόησης

- **Που βρίσκεται η επόμενη εντολή;**
  - Στην εξωτερική μνήμη.
- **Γιατί πρέπει η εντολή να έρθει μέσα στον επεξεργαστή;**
  - Γιατί δε μπορεί να εκτελεστεί από τη μνήμη.
- **Πως ξέρει ο επεξεργαστής ποια είναι η επόμενη εντολή που πρέπει να φέρει από τη μνήμη;**
  - Από το μετρητή προγράμματος *(program counter)*. Στην αρχιτεκτονική x86 ο PC υλοποιείται με το ζευγάρι των καταχωρητών CS:IP.
- **Που τοποθετείται η εντολή όταν έρχεται μέσα στον καταχωρητή;**
  - Μέσα στον καταχωρητή εντολών *(Instruction Register).*
- **Το machine code τι στοιχεία έχει;**
  - Την εντολή *(opcode, operation code)* και τις παραμέτρους της *(operands).*

Αποκωδικοποίηση εντολής στο 8086 (1/9)

Προκειμένου να αποκωδικοποιήσουμε μια εντολή σε machine code, κάνουμε τα εξής:

- Ως υπενθύμιση, τα x86 opcodes είναι όλα 1 Byte και τα υπόλοιπα είναι είτε operands είτε βοηθητικά Bytes.
- Βρίσκουμε στην επίσημη λίστα της Intel για το x86 opcodes την αντίστοιχη γραμμή που ξεκινάει με το πρώτο Byte. Για παράδειγμα έστω τα 2 Bytes **CD 21.**
- Στο αρχείο με τα opcodes βρίσκουμε τη γραμμή που ξεκινάει με **CD.**

Αποκωδικοποίηση εντολής στο 8086 (2/9)

- Βρίσκουμε αυτή τη γραμμή:

```
CC      INT             3
CD      INT             Ib
CE      INTO
CF      IRET
```

- Δηλαδή είναι η εντολή **INT**
- Είναι εντολή με μια παράμετρο *(3η στήλη).*
  - Τύπος Παραμέτρου: **Ib**
- Το επόμενο βήμα είναι η αποκωδικοποίηση της παραμέτρου.

Αποκωδικοποίηση εντολής στο 8086 (3/9)

- Ένδειξη **`Ib`**
- Κοιτώντας στο κείμενο με τα opcodes λίγο παρακάτω βλέπουμε ότι:
  - **I : Immediate data** The operand value is encoded in subsequent bytes of the instruction.
    (Άμεσα δεδομένα. Η παράμετρος βρίσκεται στα επόμενα Bytes του κώδικα μηχανής).
  - **b: Byte argument.**
    (παράμετρος 1 Byte).

Αποκωδικοποίηση εντολής στο 8086 (4/9)

- Αυτό λοιπόν σημαίνει ότι ακολουθεί παράμετρος ενός Byte με τιμή άμεσα προσδιορισμένη από τον κώδικα μηχανής.
- Το Machine Code μας είναι CD 21.
- Το CD είναι ο opcode της εντολής INT.
- Το 21 είναι η άμεση παράμετρος (δλδ, δε χρειάζεται κάποια μετατροπή) στο δεκαεξαδικό σύστημα . Οπότε το machine code αυτό αντιστοιχεί στην εντολή:

```
INT 21h
```

Αποκωδικοποίηση εντολής στο 8086 (5/9)

- Να αποκωδικοποιήσετε την εντολή:

BA 24 00

- Βρίσκουμε το BA στον πίνακα με τα opcodes.
- Βρίσκουμε ότι είναι:

B9	MOV	eCX	Iv
BA	MOV	eDX	Iv
BB	MOV	eBX	Iv
BC	MOV	eSP	Iv

- Δηλαδή, το opcode BA έχει την εξής μορφή:

BA: **MOV eDX Iv**

Αποκωδικοποίηση εντολής στο 8086 (6/9)

B9	MOV	eCX	Iv
BA	MOV	eDX	Iv
BB	MOV	eBX	Iv
BC	MOV	eSP	Iv

- Η εντολή λοιπόν είναι εντολή MOV.
- Είναι εντολή 2 παραμέτρων.
  - Η πρώτη παράμετρος **`eDX`**.
  - Η δεύτερη παράμετρος **`Iv`**.

Αποκωδικοποίηση εντολής στο 8086 (7/9)

- 1η Παράμετρος: **`eDX`.**
  - Είναι ο καταχωρητής DX
    (το e σημαίνει ότι στην επόμενη αρχιτεκτονική x86 την IA32 θα χρησιμοποιούνταν ο καταχωρητής eDX (extended DX)).

Αποκωδικοποίηση εντολής στο 8086 (8/9)

- 2η Παράμετρος **`Iv`**
  - **I: Immediate data** The operand value is encoded in subsequent bytes of the instruction. *(Άμεσα δεδομένα. Η παράμετρος βρίσκεται στα επόμενα Bytes του κώδικα μηχανής).*
  - **v: Word argument.**
    Δηλαδή, είναι μια παράμετρος 2 Byte (1 λέξη ή word είναι 2 Byte).

Αποκωδικοποίηση εντολής στο 8086 (9/9)

- Δηλαδή, τα επόμενα 2 Byte του machine code **(24 00)** αποτελούν μια λέξη και είναι τα άμεσα δεδομένα. Ποια όμως είναι η σειρά των Byte;
  - Είναι 2400h ή 0024h;
  - Η x86 αρχιτεκτονική χρησιμοποιεί την αποθήκευση των λέξεων με μορφή little-endian (πρώτα το LSB). Για αυτό το λόγο πρώτα αποθηκεύεται το χαμηλό Byte που είναι το 24 και στη συνέχεια το υψηλό που είναι το 00. Άρα η λέξη είναι 0024h.
- Συνοψίζοντας, η αποκωδικοποίηση της **BA 24 00** είναι: **MOV DX,0024h**
  (επίσης μπορεί να είναι LEA DX, msg με το msg να είναι στη διεύθυνση 0024h)

Αποκωδικοποίηση εντολής στο 8086 *(3ο παράδειγμα)* (1/8)

- Υπάρχουν κάποιες περιπτώσεις που στο machine code υπάρχουν ένα ακόμη Byte που ονομάζεται modification Byte *(δηλαδή βοηθητικό byte μετατροπής).*
- Το Byte αυτό ακολουθεί το opcode *(δηλαδή είναι το 2ο Byte του machine code)* και ονομάζεται ModR/M Byte.
- Για την αποκωδικοποίηση αυτού του Byte πρέπει να χρησιμοποιήσουμε το σχετικό πίνακα τιμών από τη Intel.
- Το αν υπάρχει ή απουσιάζει το ModR/M **καθορίζεται από τον opcode**. Δηλαδή, αν ο opcode προσδιορίζει ότι απαιτείται τέτοιο Byte τότε το 2ο Byte του machine code είναι αυτό.
- Ακολουθεί παράδειγμα...

Αποκωδικοποίηση εντολής στο 8086 *(3ο παράδειγμα)* (2/8)

- Να αποκωδικοποιήσετε την εντολή: **8E D8**
- Βρίσκουμε το opcode από το αρχείο των opcodes:

8D	LEA	Gv	M
8E	MOV	Sw	Ew
8F	POP	Ev	
9A	NOP		

- Βρίσκουμε την αντίστοιχη γραμμή:

8E MOV Sw Ew

- Δηλαδή είναι μια εντολή MOV.
- Έχει 2 παραμέτρους:
  - 1η παράμετρος **Sw**.
  - 2η παράμετρος **Ew**.

Αποκωδικοποίηση εντολής στο 8086 *(3ο παράδειγμα)* (3/8)

- 1η παράμετρος **`Sw`**:
  - **S**: The reg field of the ModR/M byte selects a segment register.
    *(το πεδίο reg του byte τροποποίησης (ModR/M) επιλέγει έναν καταχωρητή **τμήματος**).*
  - **w**: Word argument.
    (λέξη, δηλαδή 2 Byte). Αυτό μας δείχνει ότι από το πεδίο reg του byte τροποποίησης θα επιλέξουμε το καταχωρητή που έχει χωρητικότητα Word.
- Προσδιορίζεται λοιπόν ότι υπάρχει το Byte (ModR/M) οπότε το 2ο Byte του machine code είναι αυτό. Το 2ο Byte είναι το **`D8`** **.**

Αποκωδικοποίηση εντολής στο 8086 *(3ο παράδειγμα)* (4/8)

- 2η παράμετρος **Ew**
  - **E:** A ModR/M byte follows the opcode and specifies the operand. The operand is either a general-purpose register or a memory address. If it is a memory address, the address is computed from a segment register and any of the following values: a base register, an index register, a displacement.
    (Στο ModR/M byte προσδιορίζεται αυτή η παράμετρος. Είτε θα είναι καταχωρητής είτε θα είναι δ/νση μνήμης).
  - **w:** Word argument.
    (λέξη, δηλαδή 2 Byte). Αυτό μας δείχνει ότι από το πεδίο reg του byte τροποποίησης θα επιλέξουμε το καταχωρητή που έχει χωρητικότητα Word.

Πίνακας ModR/M

- Βρίσκουμε τον πίνακα ModR/M Byte και προσδιορίζουμε τη γραμμή και τη στήλη.

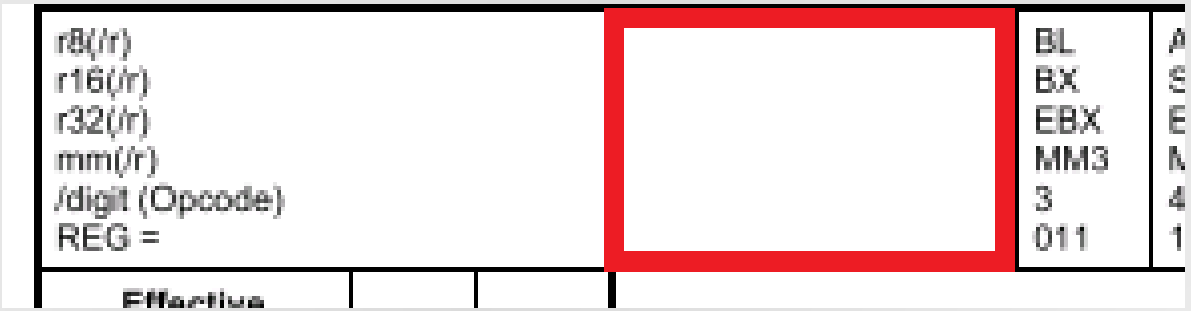
r8(/r)			AL	CL	DL	BL	AH	CH	DH	BH
r16(/r)			AX	CX	DX	BX	SP	BP[1]	SI	DI
r32(/r)			EAX	ECX	EDX	EBX	ESP	EBP	ESI	EDI
mm(/r)			MM0	MM1	MM2	MM3	MM4	MM5	MM6	MM7
/digit (Opcode)			0	1	2	3	4	5	6	7
REG =			000	001	010	011	100	101	110	111
Effective Address	**Mod**	**R/M**	**Value of ModR/M Byte (in Hexadecimal)**							
[BX+SI]	00	000	00	08	10	18	20	28	30	38
[BX+DI]		001	01	09	11	19	21	29	31	39
[BP+SI]		010	02	0A	12	1A	22	2A	32	3A
[BP+DI]		011	03	0B	13	1B	23	2B	33	3B
[SI]		100	04	0C	14	1C	24	2C	34	3C
[DI]		101	05	0D	15	1D	25	2D	35	3D
disp16[2]		110	06	0E	16	1E	26	2E	36	3E
[BX]		111	07	0F	17	1F	27	2F	37	3F
[BX+SI]+disp8[3]	01	000	40	48	50	58	60	68	70	78
[BX+DI]+disp8		001	41	49	51	59	61	69	71	79
[BP+SI]+disp8		010	42	4A	52	5A	62	6A	72	7A
[BP+DI]+disp8		011	43	4B	53	5B	63	6B	73	7B
[SI]+disp8		100	44	4C	54	5C	64	6C	74	7C
[DI]+disp8		101	45	4D	55	5D	65	6D	75	7D
[BP]+disp8		110	46	4E	56	5E	66	6E	76	7E
[BX]+disp8		111	47	4F	57	5F	67	6F	77	7F
[BX+SI]+disp16	10	000	80	88	90		A0	A8	B0	B8
[BX+DI]+disp16		001	81	89	91		A1	A9	B1	B9
[BP+SI]+disp16		010	82	8A	92		A2	AA	B2	BA
[BP+DI]+disp16		011	83	8B	93		A3	AB	B3	BB
[SI]+disp16		100	84	8C	94		A4	AC	B4	BC
[DI]+disp16		101	85	8D	95		A5	AD	B5	BD
[BP]+disp16		110	86	8E	96		A6	AE	B6	BE
[BX]+disp16		111	87	8F	97	9F	A7	AF	B7	BF
EAX/AX/AL/MM0	11	000	C0	C8	D0	D8	E0	E8	F0	F8
ECX/CX/CL/MM1		001	C1	C9	D1	[illegible]	EQ	E9	F1	F9
EDX/DX/DL/MM2		010	C2	CA	D2	DA	E2	EA	F2	FA
EBX/BX/BL/MM3		011	C3	CB	D3	DB	E3	EB	F3	FB
ESP/SP/AHMM4		100	C4	CC	D4	DC	E4	EC	F4	FC
EBP/BP/CH/MM5		101	C5	CD	D5	DD	E5	ED	F5	FD
ESI/SI/DH/MM6		110	C6	CE	D6	DE	E6	EE	F6	FE
EDI/DI/BH/MM7		111	C7	CF	D7	DF	E7	EF	F7	FF

Πίνακας σελίδα 26
Intel Architecture volume 2.

Αποκωδικοποίηση εντολής στο 8086 *(3ο παράδειγμα)* (5/8)

Προσδιορίζεται ότι η γραμμή είναι η:

Επιλέγεται ο μοναδικός καταχωρητής WORD (AX)
Ενώ η στήλη είναι η:



Εδώ θα πρέπει να επιλέξουμε τον καταχωρητή τμήματος. Θα χρησιμοποιήσουμε το REG=011

Αποκωδικοποίηση εντολής στο 8086 *(3ο παράδειγμα)* (6/8)

- Χρησιμοποιούμε το πεδίο **reg=011** επειδή:

  - 1η παράμετρος Sw
    - S :The reg field of the ModR/M byte selects a segment register.

Αποκωδικοποίηση εντολής στο 8086 *(3ο παράδειγμα)* (7/8)

Το 011 από τον πίνακα τμημάτων, βρίσκουμε ότι αντιστοιχεί στον καταχωρητή τμήματος DS.

3-Bit sreg3 Field	Segment Register Selected
000	ES
001	CS
010	SS
011	DS
100	FS
101	GS
110	Reserved*
111	Reserved*

Πίνακας σελίδα 532- Intel Architecture volume 2

Αν θέλαμε γενικό καταχωρητή τότε το πεδίο reg θα έδειχνε...

- Προσοχή, αν απαιτείται **καταχωρητής τμήματος ή γενικός καταχωρητής**.
  Για γενικούς καταχωρητές το πεδίο reg έχει αυτές τις σημασίες:

reg	register	*reg*	register
000	AX	100	SP
001	CX	101	BP
010	DX	110	SI
011	BX	111	DI

Αποκωδικοποίηση εντολής στο 8086 *(3ο παράδειγμα)* (8/8)

- Η πρώτη παράμετρος Sw, δείχνει ότι το πεδίο reg δείχνει τον καταχωρητή τμήματος που χρησιμοποιείται. Δηλαδή, η 1η παράμετρος είναι ο καταχωρητής DS.
- Η δεύτερη παράμετρος (Ew) προσδιορίζει ότι ο καταχωρητής είναι αυτός που φαίνεται στο effective address. Δηλαδή, η 2η παράμετρος είναι ο καταχωρητής AX.
- Η εντολή **`8E D8`** είναι λοιπόν η εντολή:

```
MOV DS,AX
```

Αποκωδικοποίηση εντολής στο 8086 *(4ο παράδειγμα)* (1/4)

- Να αποκωδικοποιήσετε την εντολή: **8C D8**
- Βρίσκουμε το opcode από το αντίστοιχο αρχείο:

8B	MOV	Gv	Ev
8C	MOV	Ew	Sw
8D	LEA	Gv	M

- Βρίσκουμε την αντίστοιχη γραμμή:

  8C MOV Ew Sw
- Δηλαδή είναι μια εντολή MOV.
- Έχει 2 παραμέτρους
  - 1η παράμετρος Ew.
  - 2η παράμετρος Sw.

Αποκωδικοποίηση εντολής στο 8086 *(4ο παράδειγμα)* (2/4)

- 1η παράμετρος **`Ew`** .
  - **E:** A ModR/M byte follows the opcode and specifies the operand. The operand is either a general-purpose register or a memory address. If it is a memory address, the address is computed from a segment register and any of the following values: a base register, an index register, a displacement.
    (Στο ModR/M byte προσδιορίζεται αυτή η παράμετρος. Είτε θα είναι καταχωρητής είτε θα είναι δ/νση μνήμης)
  - **w:** Word argument.
    (λέξη, δηλαδή 2 Byte). Αυτό μας δείχνει ότι από το πεδίο reg του byte τροποποίησης θα επιλέξουμε το καταχωρητή που έχει χωρητικότητα Word.

Αποκωδικοποίηση εντολής στο 8086 *(4ο παράδειγμα)* (3/4)

- 2η παράμετρος **Sw .**
  - **S:** The reg field of the ModR/M byte selects a segment register. *(το πεδίο reg του byte τροποποίησης (ModR/M) επιλέγει έναν καταχωρητή **τμήματος**).*
  - **w:** Word argument. *(λέξη, δηλαδή 2 Byte).* Αυτό μας δείχνει ότι από το πεδίο reg του byte τροποποίησης θα επιλέξουμε το καταχωρητή που έχει χωρητικότητα Word.
- Προσδιορίζεται λοιπόν ότι υπάρχει το Byte *(ModR/M)* οπότε το 2ο Byte του machine code είναι αυτό.

  Το 2ο Byte είναι το **D8 .**

Αποκωδικοποίηση εντολής στο 8086 *(4ο παράδειγμα)* (4/4)

Συνοψίζοντας:

- Η πρώτη παράμετρος είναι καταχωρητής. Από τον προηγούμενο πίνακα διαπιστώνουμε ότι είναι ο AX.
- Η δεύτερη παράμετρος είναι καταχωρητής τμήματος. Με την ίδια διαδικασία του προηγούμενου παραδείγματος, βρίσκουμε ότι το D8 αντιστοιχεί στον καταχωρητή τμήματος DS.
- Η εντολή **`8C D8`** είναι λοιπόν η εντολή:

`MOV AX,DS`

Τέλος Ενότητας

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης