

tion

qn-10

value

array after declare,

10 20

Addressing mode start

(code)?

~~address~~ (Syntax)

parent code name

002/10.2

10.2-1

Bin multiplication

Qh-9 (V_r, S)

Q.1 - MUL and IMUL

Effect of MUL/IMUL on status flags:

SF, ZF, HF, PF undefined

CF, OF: After

Ex - Q.1

Q.3 - Divide

10.2
90.2
10.2

Fig 7.2 (136)

→ Left shift from carry flag set value of zero?

Ex - 7.7;

Ex - 7.8;

Application of LS is multiplication

2 → 10

100 → 4 (2x2)

1000 → 8 (2x2x2)

→ SHL, SAL same

→ RSI

Ex 7.1 (36) Division

Fig 7.5 (Rotate left)

Rotate
7.6 (page 140)

Fig 7.7 REL

Fig 7.8 -

Fig 7.4 → Binary

8 → Partition (fir base)

9 → Optimal phase replacement alg

10 → best recently used

11 → RSA algorithm (encryption)

6 → Bounded Buffer

7 → Counting Der

Linux command -

Ch-8
8.1 (152)

Fig 8.1
POP POPF

8.2 Stack application
→ input reverse

→ 8.4 → down and set

Example-6.1 (V.V.I) 50%
 6.3 - Unconditional jump
 6.4.1 (112)

Example-6.2 (V.V.I)

EX-6.6

EX-6.8 (For)

EX-6.9 (while loop)

Chapter Summary

Ch-7

Logic, Shift and Rotate

$$\bar{A} \oplus B = \bar{A}B + \bar{A}\bar{B}$$

7.1.3 → Destination

From these

4. The AND can be

clear → and 0

Preserve → and 1

2. OR

3. XOR V.V.4

EX-R.2

EX-R.4

→ Clearing a Register (134)

→ Testing a Register for zero

cmp cx, d

→ NO

→ R.2.3-V.V.15

Shift left

24 (math) (35)

X(0)	W ₈ ⁰	W	W	W
X(1)	W ₈ ⁰	W	W	W
X(2)	W ₈ ⁰	W	W	W
X(3)	W ₈ ⁰	W	W	W
X(4)	W ₈ ⁰	W	W	W
X(5)	W ₈ ⁰	W	W	W
X(6)	W ₈ ⁰	W	W	W
X(7)	W ₈ ⁰	W	W	W

Fig 7.2 (136)

Left shift from carry
 flag and value of zero?

EX-R.7:

EX-R.8:

Application of LS in multiplication

2 → 10
 100 → 4 (2X)
 1000 → 8 (2X2)

→ SHL, SAL same

R.2.2

→ RSI

EX-R. (1.36) Division

Fig 7.5 (Rotate left)

Rotate R.6 (page 140)

Fig 7.7 RCL

Fig 7.8

Fig 7.4 → Binary

Ch-1	8-1
Fig 7.1	8-2
Fig 7.2	8-2
Fig 7.3	8-2
Fig 7.4	8-2
Fig 7.5	8-2
Fig 7.6	8-2
Fig 7.7	8-2
Fig 7.8	8-2
Fig 7.9	8-2
Fig 7.10	8-2
Fig 7.11	8-2
Fig 7.12	8-2
Fig 7.13	8-2
Fig 7.14	8-2
Fig 7.15	8-2
Fig 7.16	8-2
Fig 7.17	8-2
Fig 7.18	8-2
Fig 7.19	8-2
Fig 7.20	8-2
Fig 7.21	8-2
Fig 7.22	8-2
Fig 7.23	8-2
Fig 7.24	8-2
Fig 7.25	8-2
Fig 7.26	8-2
Fig 7.27	8-2
Fig 7.28	8-2
Fig 7.29	8-2
Fig 7.30	8-2
Fig 7.31	8-2
Fig 7.32	8-2
Fig 7.33	8-2
Fig 7.34	8-2
Fig 7.35	8-2
Fig 7.36	8-2
Fig 7.37	8-2
Fig 7.38	8-2
Fig 7.39	8-2
Fig 7.40	8-2
Fig 7.41	8-2
Fig 7.42	8-2
Fig 7.43	8-2
Fig 7.44	8-2
Fig 7.45	8-2
Fig 7.46	8-2
Fig 7.47	8-2
Fig 7.48	8-2
Fig 7.49	8-2
Fig 7.50	8-2
Fig 7.51	8-2
Fig 7.52	8-2
Fig 7.53	8-2
Fig 7.54	8-2
Fig 7.55	8-2
Fig 7.56	8-2
Fig 7.57	8-2
Fig 7.58	8-2
Fig 7.59	8-2
Fig 7.60	8-2
Fig 7.61	8-2
Fig 7.62	8-2
Fig 7.63	8-2
Fig 7.64	8-2
Fig 7.65	8-2
Fig 7.66	8-2
Fig 7.67	8-2
Fig 7.68	8-2
Fig 7.69	8-2
Fig 7.70	8-2
Fig 7.71	8-2
Fig 7.72	8-2
Fig 7.73	8-2
Fig 7.74	8-2
Fig 7.75	8-2
Fig 7.76	8-2
Fig 7.77	8-2
Fig 7.78	8-2
Fig 7.79	8-2
Fig 7.80	8-2
Fig 7.81	8-2
Fig 7.82	8-2
Fig 7.83	8-2
Fig 7.84	8-2
Fig 7.85	8-2
Fig 7.86	8-2
Fig 7.87	8-2
Fig 7.88	8-2
Fig 7.89	8-2
Fig 7.90	8-2
Fig 7.91	8-2
Fig 7.92	8-2
Fig 7.93	8-2
Fig 7.94	8-2
Fig 7.95	8-2
Fig 7.96	8-2
Fig 7.97	8-2
Fig 7.98	8-2
Fig 7.99	8-2
Fig 7.100	8-2

20-08-23 Assembly Language

ch-6

6.1 - Jump

6.2 - JNZ → Jump if not zero present
→ The cmp instruction pass 108

instruction स्मारा काई

6.2. Conditional Jumps

Table 6.1 - Conditional Jumps
Signed, unsigned

→ Exercise चेक रेजिस्टर में रेजिस्टर
and value. तो चेक compare
करा जाय ?
• Generally jump, conditional
no statements after,

Table - 4.2 v.v.I

→ Emulatoz

24.5% (download) 0800

ADD 5110 (201590 TH3)

(11.9)

INC (X6)

FTB 4.7 (provisional demand
negation)

১৭.২.১ - ৬
জি১৫২
৬
৬

→ 4.7.2 -

3. 7. 4. 7

→ G. R. H

→ 4.9 - First program

(page 82)

Program Listing. Pam 4-

→ Fig. 8.8 (VVπ)

→ Summary

5-20

↓
[৬৭]
আমি কেবল
তারি

→ Augs 05, 96

→ 5.2.4x

overflood - 4. ~~ଅନୁ~~ - ~~ଅନୁ~~
gamma overflow.

4EX-5.1 (v.v-T (98) pag)

→ Flag Register operation
Control Flag

16-68-23-05

File management

File - attribute -
Def: 1 - Mom

→ Nam

↓
+

→ toobahnqu

size

Operat

→ open

→ creation

→ transcode / truncate

→ Deposition

File access mode - 3 types

1. Sequential Access

2. Direct Access

3. Indexed Access

Directing Structure



holds me
up

t_1	a_1
t_2	1
t_3	1

Discretory
① single level

→ Assembly language pro (23)
Program Listing pg. 11.

DW → Define word.
page (28)

Chapter Summary (V.V.I)

Ch-2

→ Bin
→ Hex
→ 2.2-V.V.I
→ 2.4

→ One's complement V.V.I
→ Two's " (Ex)

→ 2.8 Ex

→ Subtraction

→ Summary

Ch-3

→ 3.1

→ Microprocessor 8086 and 8088
page 51 (IT Govt. Mem)

→ Bitwise, classification
generation wise

→ 3.2.1
→ 3.2.2

→ Physical registers and ports

→ 3.3.1 (53)

→ AX

→ DX

* Segment - offset address

Ex-3.2

Fig 3.3.1, 3.3.2

फिगर 3.3.1 और 3.3.2

Summary

Ch-4

* 4.1 statements

name operation operand com

→ 4.1.3 (Field Desc.)

→ 4.2

Number
errors

variable
Byte variable

→ name DB

→ 4.3.2

→ 4.3.3

→ 4.4 (Equ)

→ 4.5 4.1 Mov Ax
→ Exchanging.

Ch-1

- Bytes and words
- Bit position
- Fig 4.4 - low byte high
- Fig 4.5
operation of BUS

* Unidirectional Bitstream

→ Fig 4.6 - V.V.I

Details → Flag (Everything)

→ machine lang and assembly
language (Difference)