SOLUTIONS MANUAL

M. MORRIS MANO

COMPUTER SYSTEM ARCHITECTURE

Third Edition

CHAPTER 8

8.1

(a) 32 multiplexers, each of size 16 × 1.
(b) 4 inputs each, to select one of 16 registers.
(c) 4-to-16 – line decoder
(d) 32 + 32 + 1 = 65 data input lines
32 + 1 = 33 data output lines.

(e) 4 4 4 6 = 18 bits

SELA	SELB	SELD	OPR

8.2

30 + 80 + 10 = 120 n sec.
(The decoder signals propagate at the same as the muxs.)

8.3

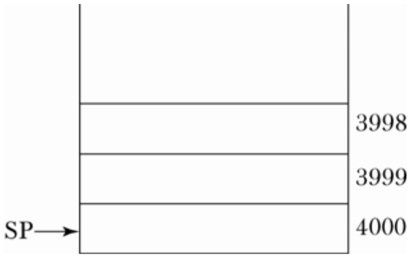
		SELA	SELB	SELD	OPR	Control word
(a)	R1 ← R2 + R3	R2	R3	R1	ADD	010 011 001 00010
(b)	R4 ← $\overline{R4}$	R4	—	R4	COMA	100 xxx 100 01110
(c)	R5 ← R5 –1	R5	—	R5	DECA	101 xxx 101 00110
(d)	R6 ← SH1 R1	R1	—	R6	SHLA	001 xxx 110 11000
(e)	R7 ← Input	Input	—	R7	TSFA	000 xxx 111 00000

8.4

	Control word	SELA	SELB	SELD	OPR	Microoperation
(a)	001 010 011 00101	R1	R2	R3	SUB	R3 ← R1 – R2
(b)	000 000 000 00000	Input	Input	None	TSFA	Output ← Input
(c)	010 010 010 01100	R2	R2	R2	XOR	R2←R2 ⊕ R2
(d)	000 001 000 00010	Input	R1	None	ADD	Output ← Input + R1
(e)	111 100 011 10000	R7	R4	R3	SHRA	R3 ← shrR7

8.5

(a) Stack full with 64 items.
(b) stack empty



8.6

PUSH : M[SP] ← DR
SP ← SP – 1
POP : SP ← SP + 1
DR ← M[SP]

8.7

(a) AB * CD * EF * + +
(b) AB * ABD * CE * + * +
(c) FG + E * CD * + B * A +
(d) ABCDE + * + * FGH + */

8.8

(a) $\dfrac{A}{B-(D+E)*C}$ (b) $A+B\dfrac{C}{D*\ E}$

(c) $\dfrac{A}{B*C}-D+\dfrac{E}{F}$ (d) $(((F + G) * E + D) * C + B) * A$

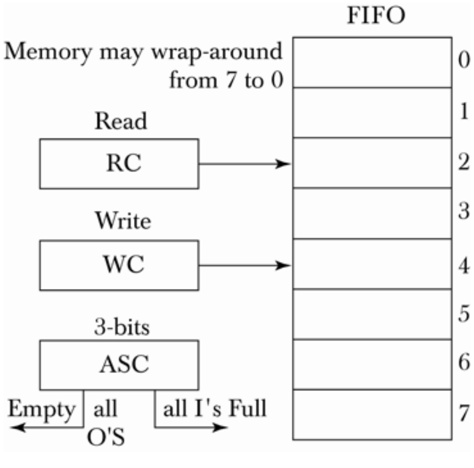
8.9

(3 + 4) [10 (2 + 6) + 8] = 616
RPN : 34 + 2 6 + 10 * 8 + *

				6		10		8		
	4		2	2	8	8	80	80	88	
3	3	7	7	7	7	7	7	7	7	616
3	4	+	2	6	+	10	*	8	+	*

8.10

WRITE (if not full) :
M [WC] ← DR
WC ← WC + 1
ASC ← ASC + 1
READ : (if not empty)
DR ← M [RC]
RC ← RC + 1
ASC ← ASC –1



8.11

8	12	12	=	32bit
op code	Address 1	Address 2		Two address instructions

2^8 = 256 combinations.
256 – 250 = 6 combinations can be used for one address

op code	Address

6×2^{12}

One address instructor

Maximum number of one address instruction:

$= 6 \times 2^{12} = 24{,}576$

8.12

(d) RPN: × AB – C + DE ж F – ж GHK ж + /=

8.13

$256\ K = 2^{8} \times 2^{10} = 2^{18}$

op code	Mode	Register	Address
5	3	6	18

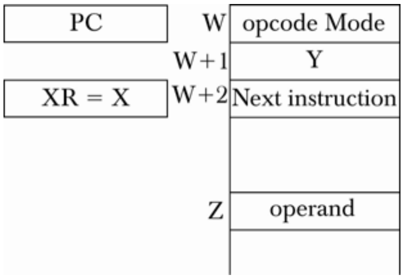
= 32

Address = 18 bits
Mode = 3 ″
Register = 6 ″
27 bits
op code 5
32 bits

8.14

Z = Effective address

(a) Direct: Z = Y
(b) Indirect: Z = M[Y]
(c) Relative: Z = Y + W + 2
(d) Indexed: Z = Y + X



8.15

(a) Relative address = 500 – 751 = – 251
(b) 251 = 000011111011; – 251 = 111100000101
(c) PC = 751 = 001011101111; 500 = 000111110100
PC = 751 = 001011101111
RA = – 251 = +111100000101
EA = 500 = 000111110100

8.16

Assuming one word per instruction or operand.

Computational type	Branch type
Fetch instruction	Fetch instruction
Fetch effective address	Fetch effective address and transfer to PC
Fetch operand	
3 memory references	2 memory references.

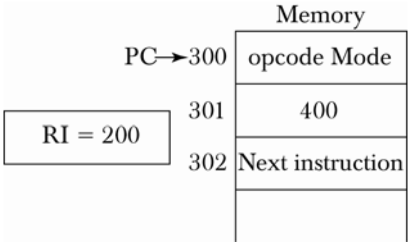
8.17

The address part of the indexed mode instruction must be set to zero.

8.18

Effective address

(a) Direct: 400
(b) Immediate: 301
(c) Relative: 302 + 400 = 702
(d) Reg. Indirect: 200
(e) Indexed: 200 + 400 = 600



8.19

1 = C	0 = C	1 = C	0 = Reset initial carry	
6E	C3	56	7A	
13	55	6B	8F	
82	18	C2	09	Add with carry

8.20

10011100		10011100		10011100	
10101010	AND	10101010	OR	10101010	XOR
10001000		11111110		00110110	

8.21

(a) AND with: 0000000011111111
(b) OR with: 0000000011111111
(c) XOR with: 0000111111110000

8.22

Initial: 01111011 C = 1
SHR: 00111101
SHL: 11110110
SHRA: 00111101
SHLA: 11110110 (over flow)
ROR: 10111101
ROL: 11110110
RORC: 10111101
ROLC: 11110111

8.23

+ 83 = 01010011 – 83 = 10101101
+ 68 = 01000100 – 68 = 10111100

(a) – 83 10101101
+ 68 +01000100
– 15 11110001
(in 2’s complement)

$$\begin{array}{lll} \text{(b)} & & 1\ 0 \text{ carries} \\ & -68 & 10111100 \\ & \underline{-83} & \underline{+10101101} \\ & -151 & \ 01101001 \\ & \ \ \wedge & \\ & -128 & \text{(over flow)} \end{array}$$

$$\begin{array}{ll} \text{(c)} & -68 = 10111100 \\ & -34 = 11011110 \\ & \qquad\ \ \oplus\ = 1 \end{array}$$

$$\begin{array}{ll} \text{(d)} & \ \ -83 = 10101101 \\ & -166 \neq 01011010 \\ & \qquad\ \ \text{Over flow} \end{array}$$

8.24

$Z = F'_0\ F'_1\ F'_2\ F'_3\ F'_4\ F'_5\ F'_6\ F'_7 = (F_0 + F_1 + F_2 + F_3 + F_4 + F_5 + F_6 + F_7)'$

8.25

$$\begin{array}{lll} & & 1\ 1 \\ \text{(a)} & 72 & 01110010 \\ & \underline{C6} & \underline{11000110} \\ & 138 & 00111000 \\ & C = 1 & S = 0\ \ Z = 0\ \ V = 0 \end{array}$$

$$\begin{array}{lll} \text{(b)} & & 0\ 1 \\ & 72 & 01110010 \\ & \underline{1E} & \underline{00011110} \\ & 90 & 10010000 \\ C = 0 & & S = 1\ \ \ Z = 0\ \ V = 1 \end{array}$$

$$\begin{array}{llll} \text{(c)} & 9A = & 10011010 & \text{2's comp.} \\ & & 01100110 & \\ & \underline{72} & \underline{01110010} & \\ & D8 & 11011000 & \end{array}$$

$C = 0\ \ S = 1\ \ Z = 0\ \ V = 1$

(Borrow = 1)

$$\begin{array}{lll} \text{(d)} & 72 = & 01110010 \\ & \underline{8D} & \underline{10001100} \\ & 00 & 00000000 \end{array}$$

$C = 0\ \ S = 0\ \ Z = 1\ \ V = 0$

(e) $C = 0\ \ S = 0\ \ Z = 1\ \ V = 0$

8.26

C = 1 if A < B, therefore C = 0 if A $\geq$ B

Z = 1 if A = B, therefore Z = 1 if A $\neq$ B

For A > B we must have A $\geq$ B provided A $\neq$ B

Or C = 0 and Z = 0 (C’Z’) = 1

For A $\leq$ B we must have A < B or A = B

Or C = 1 or Z = 1 (C + Z) = 1

8.27

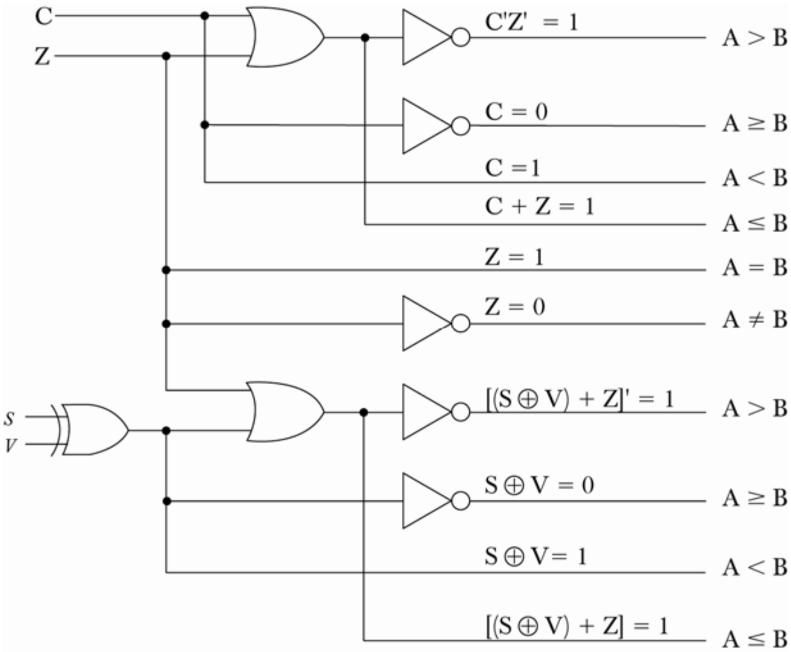
A ≥ B implies that A – B ≥ 0 (positive or zero)
Sign S = 0 if no over flow (positive)
or S = 1 if over flow (sign reversal)
Boolean expression: S'V' + SV = 1 or (S ⊕ V) = 0

A < B is the complement of A ≥ B (A – B negative)
then S = 1 if V = 0
or S = 0 if V = 1 (S ⊕ V) = 1

A > B Implies A ≥ B but not A = B
(S ⊕ V) = 0 and Z = 0

A ≤ B Implies A < B or A = B
S ⊕ V = 1 or Z = 1

8.28



8.29

	Unsigned	Signed
A = 01000001	65	+ 65
B = 10000100	132	– 124
A + B = 11000101	197	– 59

(c) C = 0 Z = 0 S = 1 V = 0
(d) BNC BNZ BM BNV

8.30

(a) A = 01000001 = + 65
B = 10000100 = 132
A – B = 10111101 = – 67 (2's comp. of 01000011)
(b) C (borrow) = 1; Z = 0 65 < 132
A < B
(c) BL, BLE, BNE

8.31

(a)

$$\begin{aligned} A &= 01000001 = +\ 65 \\ \underline{B} &\underline{= 10000100 = -124} \quad + \\ A - B &= 10111101 \quad +189 \ = \underbrace{010111101}_{9\,\text{bits}} \end{aligned}$$

(b) S = 1 (sign reveral)   +189 > 127
Z = 0
V = 1 (over flow)   65 > –124
A > B

(c) BGT, BGE, BNE

8.32

	PC	SP	Top of Stack
Initial	1120	3560	5320
After CALL	6720	3559	1122
After RETURN	1122	3560	5320

8.33

Branch instruction – Branch without being able to return.
Subroutine call – Branch to subroutine and then return to calling program.
Program interrupt – Hardware initiated branch with possibility to return.

8.34

See Sec. 8–7 under “Types of Interrupts”.

8.35

(a) SP ← SP – 1
M[SP] ← PSW
SP ← SP – 1
M[SP] ← PC
TR ← IAD (TR is a temporary register)
PSW ← M[TR]
TR ← TR + 1
PC ← M[TR]
Go to fetch phase.

(a) PC ← M[SP]
SP ← SP + 1
PSW ← M[SP]
SP ← SP+1

8-37

Window Size = L + 2C + G

Computer 1: 10 + 12 + 10 = 32
Computer 2: 8 + 16 + 8 = 32
Computer 3: 16 + 32 + 16 = 64

Register file = (L + C) W + G

Computer 1: (10 + 6) 8 + 10 = 16 × 8 + 10 = 138
Computer 2: (8 + 8) 4 + 8 = 16 × 4 + 8 = 72
Computer 3: (16 + 16) 16 + 16 = 32 × 16 + 16 = 528

8-38

(a) SUB R22, # 1, R22	R22 ← R22 – 1 (Subtract 1)
(b) XOR R22, # –1, R22	R22 ← R22 ⊕ all 1’s (x ⊕ 1 = x′)
(c) SUB R0, R22, R22	R22 ← 0 – R22
(d) ADD R0, R0, R22	R22 ← 0 + 0
(e) SRA R22, # 2, R22	Arithmetic shift right twice
(f) OR R1, R1, R1	R1 ← R1 V R1
or ADD R1, R0, R1	R1 ← R1 + 0
or SLL R1, # 0, R1	shift left 0 times

8-39

(a) JMP Z, # 3200, (RO)	PC ← 0 + 3200
(b) JMPR Z, – 200	PC ← 3400 + (– 200)