M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

A

B
A

A + B

C
A + B

$(A + B) \times C$

D
$(A + B) \times C$

E
D
$(A + B) \times C$

$D \times E$
$(A + B) \times C$

$(A + B) \times C + D \times E$

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

Opcode	Mode	Address or operand

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

Memory

250	Opcode	Mode
251	ADRS	
252	Next instruction	
	Program	
500	800	
	Data	

PC = 250

ACC

Opcode: Load ACC
Mode: Direct address
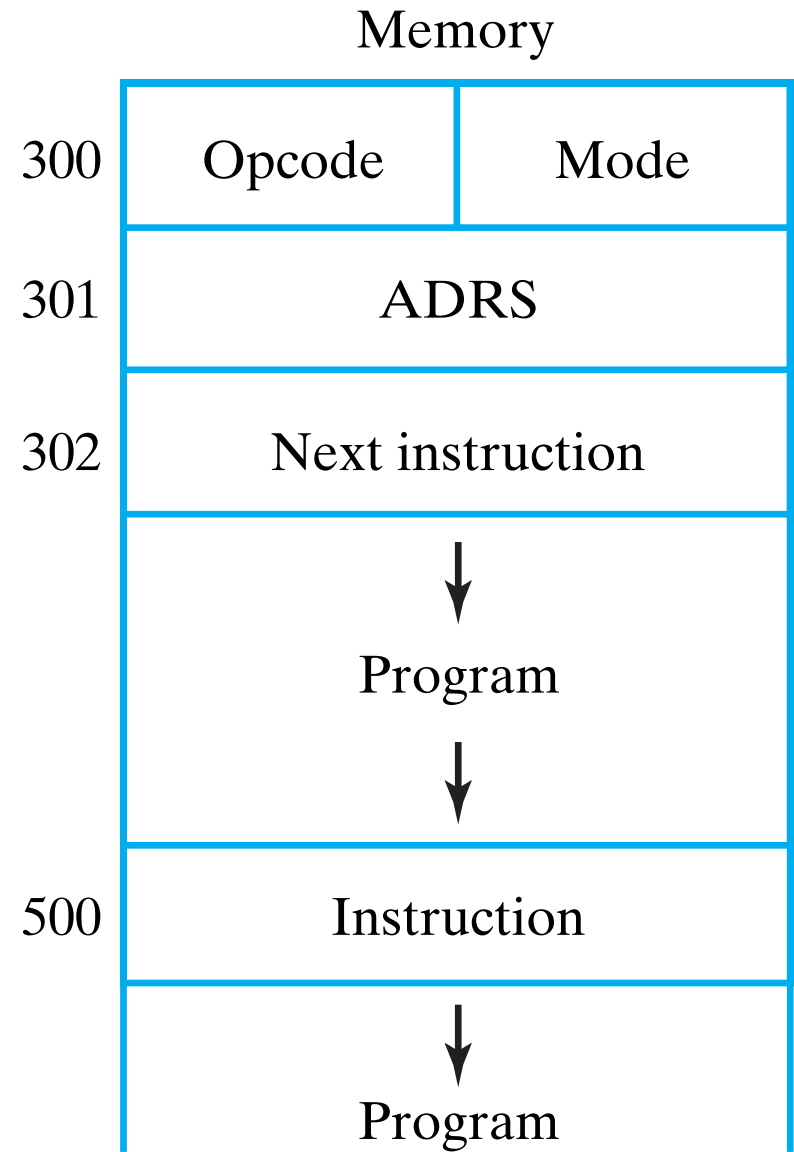ADRS: 500
Operation: $ACC \leftarrow 800$

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

Memory

300 | Opcode | Mode |

301 | ADRS |

302 | Next instruction |

| Program |
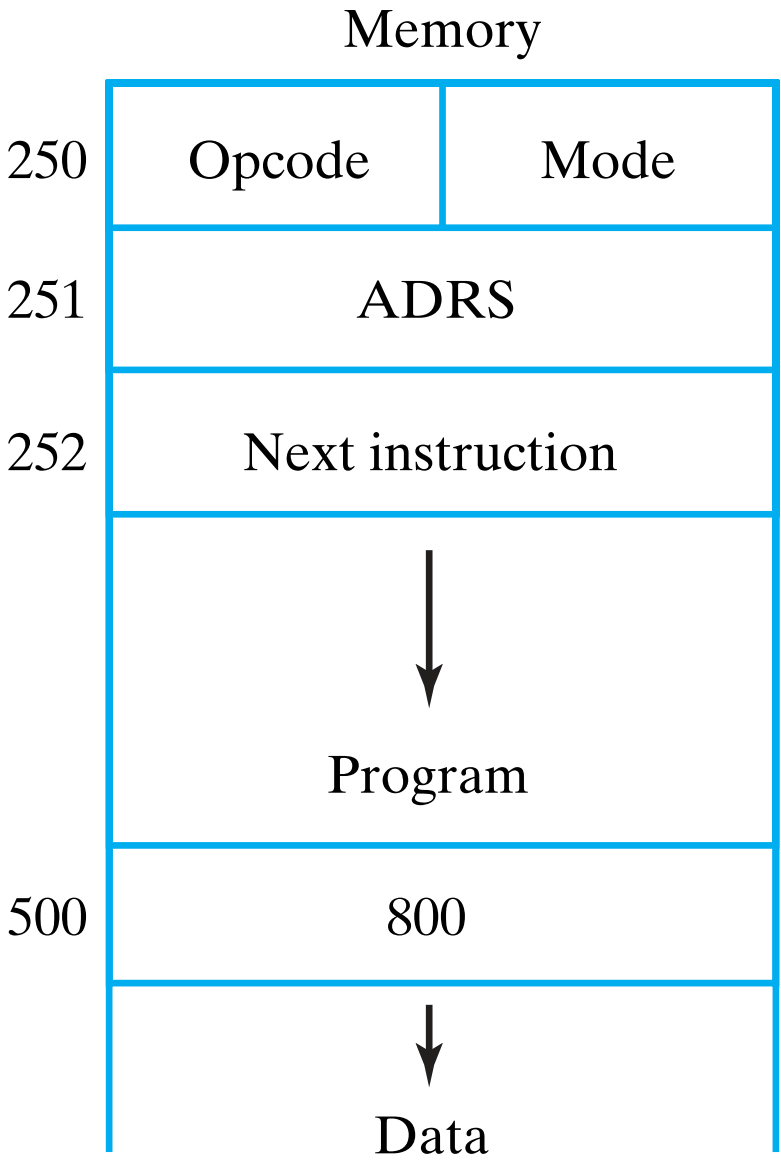
PC = 300

ACC

Opcode: Branch if ACC = 0
Mode: Direct address
ADRS: 500
Operation: PC ← 500 if ACC = 0
                PC ← 302 if ACC ≠ 0

500 | Instruction |

| Program |

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

PC = 250

R1 = 400

ACC

Opcode: Load to ACC

Memory

250	Opcode	Mode
251	ADRS or NBR = 500	
252	Next instruction	
400	700	
500	800	
752	600	
800	300	
900	200	

T 10-1

□ TABLE 10-1
Symbolic Convention for Addressing Modes

Addressing Mode	Symbolic Convention	Register Transfer	Refers to Figure 10-6	
			Effective Address	Contents of *ACC*
Direct	LDA ADRS	$ACC \leftarrow M[ADRS]$	500	800
Immediate	LDA #NBR	$ACC \leftarrow NBR$	251	500
Indirect	LDA [ADRS]	$ACC \leftarrow M[M[ADRS]]$	800	300
Relative	LDA $ADRS	$ACC \leftarrow M[ADRS + PC]$	752	600
Index	LDA ADRS (R1)	$ACC \leftarrow M[ADRS + R1]$	900	200
Register	LDA R1	$ACC \leftarrow R1$	—	400
Register-indirect	LDA (R1)	$ACC \leftarrow M[R1]$	400	700

□ TABLE 10-2
Typical Data Transfer Instructions

Name	Mnemonic
Load	LD
Store	ST
Move	MOVE
Exchange	XCH
Push	PUSH
Pop	POP
Input	IN
Output	OUT

Memory

Address

	100
C	101
B	102
A	103
	104

SP = 101

R1

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e
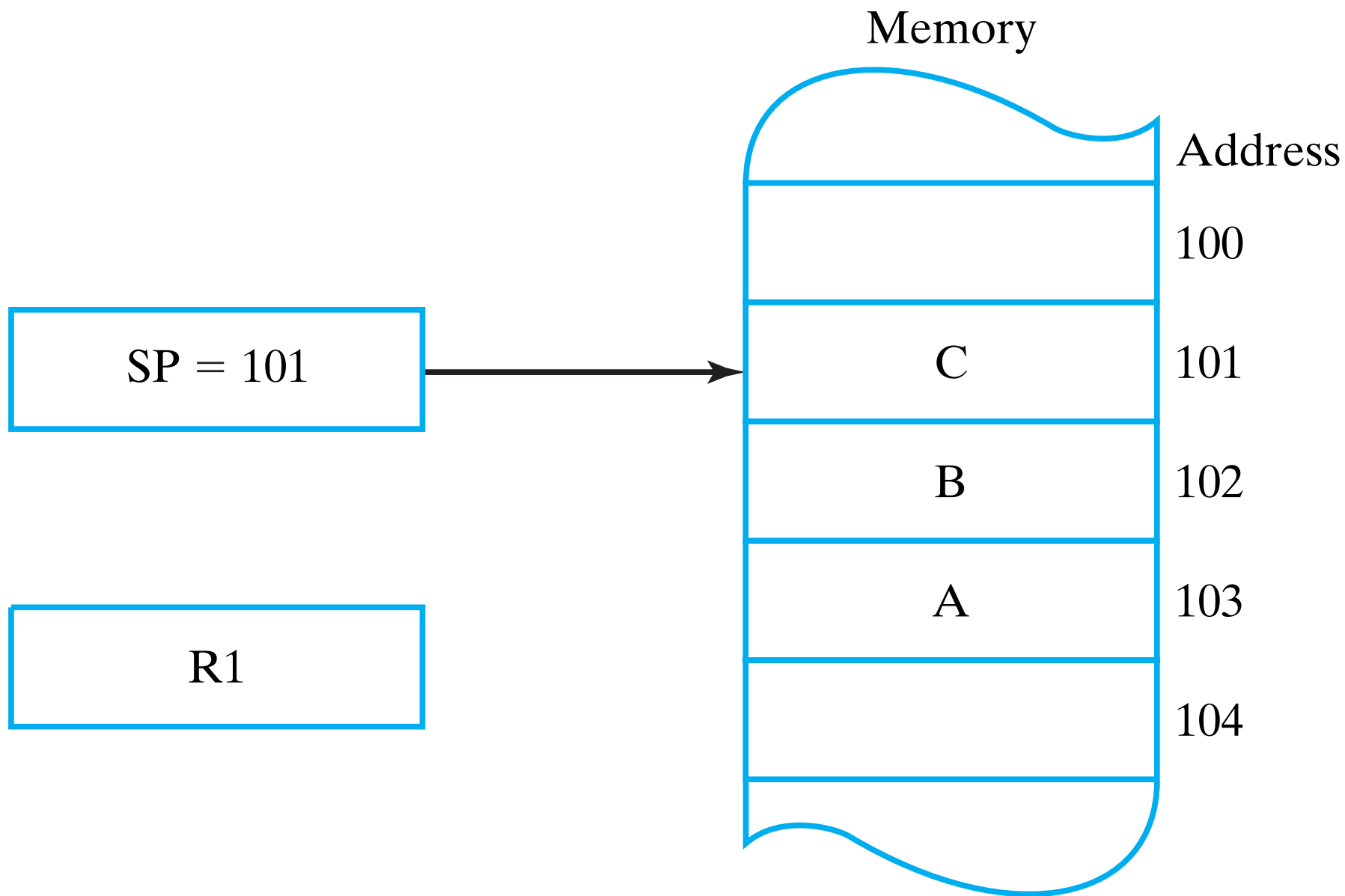
☐ TABLE 10-3
Typical Arithmetic Instructions
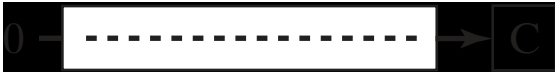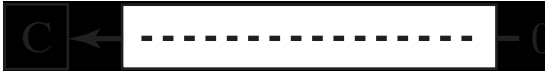
Name	Mnemonic
Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Add with carry	ADDC
Subtract with borrow	SUBB
Subtract reverse	SUBR
Negate	NEG

☐ TABLE 10-4
Typical Logical and Bit-Manipulation Instructions

Name	Mnemonic
Clear	CLR
Set	SET
Complement	NOT
AND	AND
OR	OR
Exclusive-OR	XOR
Clear carry	CLRC
Set carry	SETC
Complement carry	COMC

□ TABLE 10-5
Typical Shift Instructions

Name	Mnemonic	Diagram
Logical shift right	SHR	
Logical shift left	SHL	
Arithmetic shift right	SHRA	
Arithmetic shift left	SHLA	
Rotate right	ROR	
Rotate left	ROL	
Rotate right with carry	RORC	
Rotate left with carry	ROLC	

1	8	23
s	e	f

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

☐ TABLE 10-6
Evaluating Biased Exponents

Exponent E in decimal	Biased exponent $e = E + 127$	
	Decimal	Binary
-126	$-126 + 127 = 1$	00000001
-001	$-001 + 127 = 126$	01111110
000	$000 + 127 = 127$	01111111
$+001$	$001 + 127 = 128$	10000000
$+126$	$126 + 127 = 253$	11111101
$+127$	$127 + 127 = 254$	11111110

□ TABLE 10-7
Typical Program Control Instructions

Name	Mnemonic
Branch	BR
Jump	JMP
Call procedure	CALL
Return from procedure	RET
Compare (by subtraction)	CMP
Test (by ANDing)	TEST

☐ **TABLE 10-8**
Conditional Branch Instructions Relating to Status Bits in the PSR

Branch Condition	Mnemonic	Test Condition
Branch if zero	BZ	$Z = 1$
Branch if not zero	BNZ	$Z = 0$
Branch if carry	BC	$C = 1$
Branch if no carry	BNC	$C = 0$
Branch if minus	BN	$N = 1$
Branch if plus	BNN	$N = 0$
Branch if overflow	BV	$V = 1$
Branch if no overflow	BNV	$V = 0$

☐ **TABLE 10-9**
Conditional Branch Instructions for Unsigned Numbers

Branch Condition	Mnemonic	Condition	Status Bits*
Branch if above	BA	$A > B$	$C + Z = 0$
Branch if above or equal	BAE	$A \geq B$	$C = 0$
Branch if below	BB	$A < B$	$C = 1$
Branch if below or equal	BBE	$A \leq B$	$C + Z = 1$
Branch if equal	BE	$A = B$	$Z = 1$
Branch if not equal	BNE	$A \neq B$	$Z = 0$

*Note that C here is a borrow bit.

☐ **TABLE 10-10**
Conditional Branch Instructions for Signed Numbers

Branch condition	Mnemonic	Condition	Status Bits
Branch if greater	BG	$A > B$	$(N \oplus V) + Z = 0$
Branch if greater or equal	BGE	$A \geq B$	$N \oplus V = 0$
Branch if less	BL	$A < B$	$N \oplus V = 1$
Branch if less or equal	BLE	$A \leq B$	$(N \oplus V) + Z = 1$
Branch if equal	BE	$A = B$	$Z = 1$
Branch if not equal	BNE	$A \neq B$	$Z = 0$

External interrupts

1

2

3

4

Central processing unit (CPU)

End of execution
of instruction

EI

Enable-interrupt
flip-flop

INTACK

Interrupt acknowledge

Interrupt vector
address

IVAD

PC

To memory
stack

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e