

3.4 (1) $1234H + 5678H = 68ACH$	$OF = 0$	$CF = 0$	
$689CH - 5678H = 1234H$	$PF = 0$	$OF = 0$	$CF = 0$ 转移到 N4
(2) $8765H + 7654H = FDB9H$	$OF = 0$	$CF = 0$	
$FDB9 - 7654H = 8756H$	$PF = 1$		转移到 N3
(3) $4325H + 8761H = CA86H$	$OF = 0$	$CF = 0$	
$CA86H - 8761H = 4325H$	$PF = 0$	$OF = 0$	转移到 N4
(4) $4321H + 7762H = BA83H$	$OF = 1$		转移到 N1
(5) $5678H + 1234H = 68ACH$	$OF = 0$	$CF = 0$	
$68ACH - 1234H = 5678H$	$PF = 1$		转移到 N3

3.5 (1) $AX = 1$ (3) $AX = 5$ (5) $AX = C76AH$

3.6 (8) TEST AL 80H	测试最高位
JS N1	$SF = 1$ 最高位为 1
JP N2	$PF = 1$ 低 7 位有偶数个置 1
JNP N3	$PF = 0$ 低 7 位有奇数个置 0
N1: TEST AL 7FH	测试低 7 位
JP N2	$PF = 1$ 低 7 位有偶数个
N3: AND AL 7HH	最高位置 0
EXIT: HLT	
N2 OR AL 80H	最高位置 1

(9) MOV CL 4	
ROL AL CL	
(10) MOV DS 1000H	
MOV DI 0	MOV CX, 00FF
N1: INC DI 1	AGAIN
MOV CX DX	
AND CL FFH	MOV [DI], 0
NOT CL	LOOP AGAIN
JNZ N1	
(11) PUSH AX	
PUSH BX	
POP BX	
POP AX	

3.8 IP = 0087H

3.9 (1)

(1) 采用移位指令:

```
XOR AH, AH    reg, reg      3T
SHL AX, 1     count = 1    2T
MOV BX, AX    reg, reg      2T
SHL AX, 1     count = 1    2T
SHL AX, 1     count = 1    2T
ADD AX, BX    reg, reg      3T
```

共 15T

(2) 采用乘法指令:

```
MOV BL, 10    reg, data   4T
MUL BL        reg(8 bit) 70 → 77T
```

共 74-81T

移位指令更快