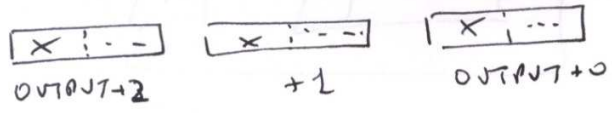


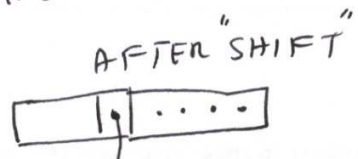
①

INPUT-BIN DB ?

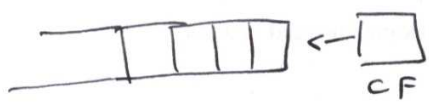
OUTPUT-BCD DB 3 DUP (?)



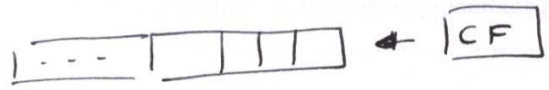
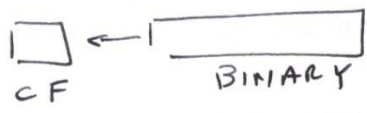
~~FOR THE~~ TO SHIFT & JUST CHECK ~~LOWER~~ BIT IN POSITION 5 (I CAN SET CARRY FLAG EQUAL TO THAT BIT) AND THEN DO A ROTATE THROUGH CARRY



ACTUAL "SHIFT"



FOR RIGHTMOST FIELD I PUT THE BIT TO BE INSERTED FROM RIGHT IN CF



FIRST BLOCK

BASICALLY A "SHIFT" IS A ROTATE THROUGH CARRY PLUS A COPY OF 5TH POSITION BIT INTO CF FOR NEXT (RIGHT TO LEFT) BLOCK.

(2)

NUMBER-SHIFTS EQU 8

MOV AL, INPUT_BIN

XOR DL, DL

XOR BX, BX

XOR CL, CL

NEXT_SHIFT: INC CL

CMP CL, NUMBER-SHIFTS

JE END-CONVERSION-ALMOST

SHL AL, 1

RCL BL, 1

TEST BL, 00010000B

~~JZ~~ SET_CF_TO_0-A

STC

RCL BH, 1

JMP NEXT_BLOCK-A1

SET_CF_TO_0-A: CLC

NEXT_BLOCK-A: RCL BH, 1

NEXT_BLOCK-A1: TEST BH, 00010000B

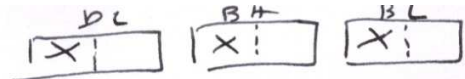
JZ SET_CF_TO_0-B

STC

JMP NEXT_BLOCK-B

SET_CF_TO_0-B: CLC

NEXT_BLOCK-B: RCL DL, 1



; AL stores a copy of input

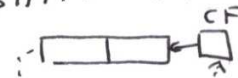
; DL: BX
store a copy of output

; CL COUNTING # OF SHIFTS

; IF ALREADY DONE 8 SHIFTS EXIT TO END-CONVERSION-ALMOST PROCESS

; STILL IN ALGORITHM OF CONVERTING EXTRACTING NEXT MSB FROM BINARY (IN CF)

; "SHIFT" FIRST BLOCK



; NEED TO CHECK BIT IN POSITION 5 AND COPY IN CF

; IF I AM HERE IT IS BECAUSE BIT IS 1 AND I HAVE TO SET CF TO 1

; HAVE HERE CF OK FOR BLOCK #2 (TO DO SAME STUFF)

; THIS IS LAST BLOCK NO NEED TO PROPAGATE CARRY FURTHER
; end of shift part

(3)

; entering clock part

~~AND BL, 00001111B~~ ; ~~each lower block~~
AND BL, 00001111B ; ZEROING HIGHER PART
OF BL (LOWER BLOCK)

CMP BL, 5

; TESTING WITH 5

~~JXX~~ JXX LESS-5-A

; JUMP IF LESS THAN
5

ADD BL, 3

; I AM HERE BECAUSE
NEED CORRECTION

LESS-5-A: AND BH, 00001111B

; NEXT BLOCK

CMP BH, 5

; again

JXX LESS-5-B

; JUMP IF LESS THAN 5

ADD BH, 3

; CORRECTION

LESS-5-B: AND BH, 00001111B

; last block

CMP DL, 5

; again

JXX LESS-5-C

; JUMP IF LESS THAN 5

ADD DL, 3

; CORRECTION

LESS-5-C: JMP NEXT-SHIFT ; ENDED CORRECTION

END-CONVERSION-ALMOST :

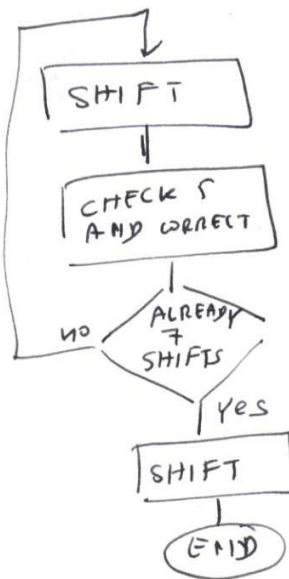
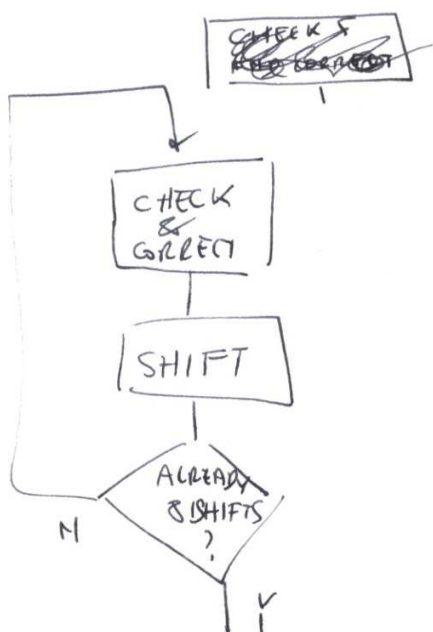
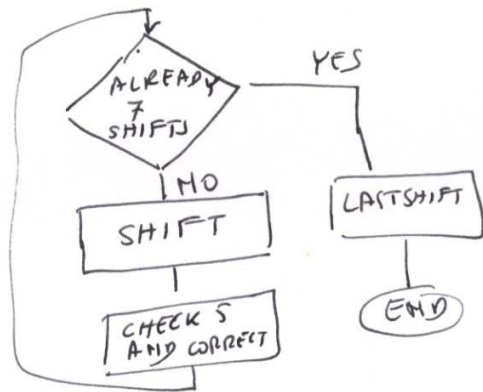
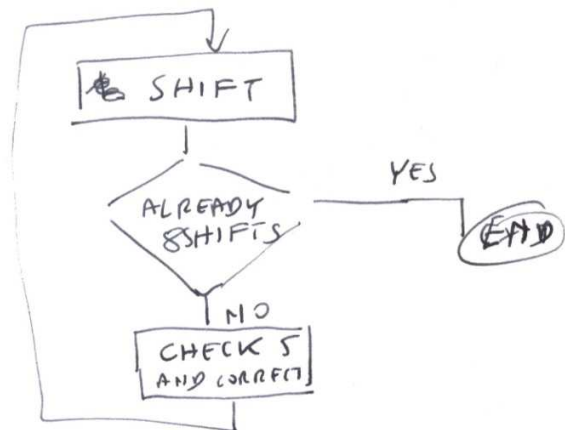
; I STILL MISS THE
LAST SHIFT
BECAUSE I INCREMENT
CL BEFORE SHIFTING

here the shift
code part.

MOV OUTPUT-BCD, BL
MOV OUTPUT-BCD+1, BH
MOV OUTPUT-BCD+2, DL

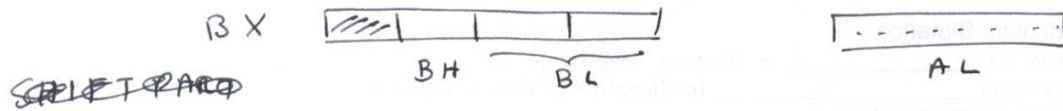
; done copy result into
OUTPUT

4



⑤ INPUT_BIN DB (?)
 OUTPUT_BIN DB 2 DUP (?) packed for.

for packed for - shift is easy difficult is compare and correct.



SHIFT

SHL AL, 1

MSB IN CF

RCL BX, 1

CF IN LSB OF BX

COMPARE AND CORRECT PART

FIRST BLOCK (LOW BL)
~~FOR DL, DL~~ ; use DL as correct to be added
 MOV DH, BL ; make a copy into DH
 AND CH, 00001111B ; reset higher part.
 CMP DH, 5

JXXX NOCORR-1 ; if less than 5 no correction
 MOV DH, 3 ; put a 3 into lower DH

NO_CORR-1 : MOV DH, BL ; to correct lower BL
 AND DH, 11110000B ; now test higher
 CMP DH, 5 * 2⁴ ; extracted higher

JXXX NOCORR-2 ; if less no correction

ADD DL, 3 * 2⁴ ; correction, i.e. a 3 in higher DL

NO_CORR-2 : ADD BL, DL ; correction performed
 MOV DH, BH ; now last block

XOR DH, 00001111B

CMP DH, 5

JXXX NOCORR-3

ADD BH, 3

NOCORR-3 :

; compare and correction performed.