

Experiment No: 10
Code Conversions

Program 18: Write an ALP to convert the contents of memory location 3100h into an ASCII character. The 3100h location contains a single Hex digit (4 MSB is zero). Store the ASCII equivalent in memory location 3200h.

Algorithm:

1. Initialize the source pointer & destination pointer registers
2. Compare the data with 0Ah
3. If it is <0Ah, add 30h else add 37h
4. Store the result in destination pointer register
5. End the program

Source Code:

```
Output 2500ad      ; Linker version command
Org 2000h           ; Starting address
Mov SI,3100h        ; Initialize the source pointer
Mov DI,3200h        ; Initialize the destination pointer
Mov AX,0000h        ; Clear AX reg
Mov AL,[SI]         ; Move the hex data from SI to AL reg
Cmp AL,0Ah          ; Compare it with 0AH
Jc AHEAD           ; Jump if cy=1 to AHEAD
Add AL,'A'-'9'-1 (or) Add AL,07h ; Add 07H to AL reg
AHEAD:Add AL,30h     ; Add 30H to AL reg
Mov [DI],AL         ; Store the result in dest ptr location
Int 3               ; Halt
Org 3100h
Db 09h
```

Result: 3200h 39h

Program 19: Write an ALP to convert the contents of memory location 3100h from ASCII to an equivalent Hexadecimal number. Place the result at memory location 3200h.

Algorithm:

1. Initialize the source pointer & destination pointer registers
2. Subtract 30h from the input data
3. Compare the value with 09h
3. If it is >09h, subtract 07h else do nothing
4. Store the result in destination pointer register
5. End the program

Source Code:

```
Output 2500ad      ; Linker version command
Org 2000h           ; Starting address
Mov SI,3100h        ; Initialize the source pointer
Mov DI,3200h        ; Initialize the destination pointer
Mov AX,0000h        ; Clear AX reg
Mov AL,[SI]         ; Move the hex data from SI to AL reg
Sub AL,30h          ; Subtract 30H from AL reg
Cmp AL,09h          ; Compare value of AL reg with 09H
Jg AHEAD            ; Jump if AL>09H to AHEAD
Jmp DONE            ; Jump DONE
AHEAD:Sub AL,07h     ; Subtract 07H from AL reg
DONE:Mov [DI],AL     ; Store the result in dest ptr location
Int 3               ; Halt
Org 3100h
Db 43h
```

Result: 3200h 0Ch

Program 20: Write an ALP to convert 2 digit packed BCD number into its Binary equivalent number. Packed 2 digit number is stored in memory location 3100h & place the result at memory location 3200h.

Algorithm:

1. Initialize the source pointer & destination pointer registers
2. Split the packed BCD into two unpacked BCD digits BCD1 & BCD2
3. Multiply BCD2 by 10 i.e., 0ah
4. Add BCD1 to the answer in step 2
5. Store the result in destination location
6. End the program

Source Code:

```

Output 2500ad      ; Linker version command
Org 2000h           ; Starting address
Mov SI,3100h        ; Initialize the source pointer
Mov DI,3200h        ; Initialize the destination pointer
Mov AL,[SI]         ; Move the data from SI to AL reg
And AL,0Fh          ; Mask the upper nibble of AL reg
Mov DL,AL           ; Save BCD1 in DL reg
Mov AL,[SI]         ; Move the data from SI to AL reg again
And AL,0F0h         ; Mask the lower nibble of AL reg
Mov CL,04h          ; Initialize the count
Ror AL,CL           ; Rotate right AL CL times to get BCD2
Mov BL,0Ah          ; Move 0AH to BL reg
Mul BL              ; Multiply it with BCD2
Add AL,DL           ; Add it with BCD1
Mov [DI],AL         ; Store the result in dest ptr location
Int 3               ; Halt
Org 3100h
Db 69h

```

Result: 3200 45h

Program 21: Write an ALP to convert Binary number into its 2 digit packed BCD number. The binary number is stored in memory location 2100h & place the result at memory location 2200h.

Algorithm:

1. Initialize the source pointer & destination pointer registers
2. Divide the value with 0Ah
3. Rotate the quotient 4 times & add it with remainder
4. Store the result in destination location

Source Code:

```
Output 2500ad      ; Linker version command
Org 2000h           ; Starting address
Mov SI,2100h        ; Initialize the source pointer
Mov DI,2200h        ; Initialize the destination pointer
Mov AX,0000h        ; Clear AX reg
Mov AL,[SI]         ; Move the data from SI to AL reg
Mov BL, 0Ah         ; Move 0AH to BL reg
Div BL              ; Divide AL with BL
Mov CL,04h          ; Initialize the count
Rol AL,CL           ; Rotate left quotient CL times
Or AL,AH            ; Add it with remainder
Mov [DI],AL         ; Store the result in dest ptr location
Int 3               ; Halt
Org 2100h
Db 45h
```

Result: 2200 69d

Program 22: Write an ALP to convert decimal number into its equivalent 7-segment conversion using XLAT instruction. The 7-segment codes are stored in memory as a lookup table starting from 2100h (use common cathode codes) for 7-segment display.

Algorithm:

1. Initialize the source pointer & destination pointer registers
2. Save the source location address in BX register
3. Take the input in AL register
4. Convert the decimal value into 7-segment code
5. Store the result in destination location
6. End the program

Source Code:

```
Output 2500ad      ; Linker version command
Org 2000h           ; Starting address
Mov SI,2100h        ; Initialize the source pointer
Mov DI,2200h        ; Initialize the destination pointer
Mov BX,SI           ; Copy the source address in BX reg
Mov AL,05h          ; Move the input data in AL reg
Xlat                ; Translate byte in AL from lookup table
Mov [DI],AL         ; Store the result in dest ptr location
Int 3               ; Halt
Org 2100h
Db 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh,
07h, 7fh, 6fh
```

Result: 2200 6dh

Program 23: Write an ALP to convert temperature from degree centigrade into degree Fahrenheit using $C = 5/9 * (F - 32)$.
 $F = 9C/5 + 32 \rightarrow F = 9C/5 + 20h$

Algorithm:

1. Initialize the source pointer & destination pointer registers
2. Multiply the input with 09h
3. Divide it with 05h & add with 20h
4. Store the result in destination location
5. End the program

Source Code:

Output 2500ad	; Linker version command
Org 2000h	; Starting address
Mov SI,2100h	; Initialize the source pointer
Mov DI,2200h	; Initialize the destination pointer
Mov AL,[SI]	; Move the data from SI to AL reg
Mov BL,09h	; Move 09H to BL reg
Mul BL	; Multiply AL with BL regs
Mov CL,05h	; Initialize CL reg
Div CL	; Divide AL with CL reg
Add AL,20h	; Add it with 20H
Mov [DI],AL	; Store the result in dest ptr location
Int 3	; Halt
Org 2100h	
Db 05h	

Result: 2200 29h