**Microprocessor Lab Report (Assignment - 1)**

Bisakh Mondal 001810501079

Format: Question | Assembly Code (Memory Address, OpCode, Mnemonics, Comments)

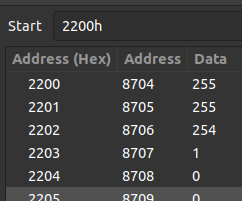
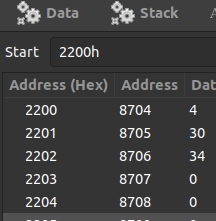
(\* The programs run in simulator seamlessly, though while running into 8085 it's better to use **RST5** instead of **HLT** to return the control to monitor program, without stopping the processor by **HLT**)

**(\* Simulator: GNUSim8085, ( here Only the numbers which can be stored in memory is decimal (not hex format(maybe for better/easy user's interaction))**

**So for BCD operations** [**this**](https://www.sim8085.com/) **online simulator has been used)**

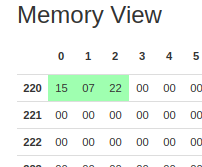
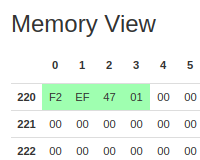
**(Q1)** Load the contents of memory location 2200Hand 2201H into Registers. Add this registers and store it into location 2202H and 2203H

|  |
| --- |
| 4200 3A 00 022 LDA 2200H ; Load Content from 2200H into accumulator  4203 47 MOV B, A; Move accumulator to B register   4204 3A 01 022 LDA 2201H; Load Content from 2201H into accumulator  4207 0E 00 MVI C, 00; Reset C register for Carry   4209 80 ADD B; Addition   420A D2 0E 042 JNC NOCARRY; check is carry is generated 420D 0C INR C; Increment C register if CY is set 420E 32 02 022 NOCARRY: STA 2202H; Store Sum into 2202H   4211 79 MOV A, C; Move carry into accumulator   4212 32 03 022 STA 2203H; Store Carry   4215 76 HLT; |



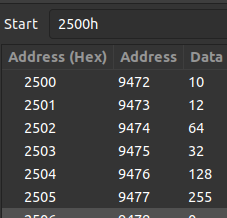
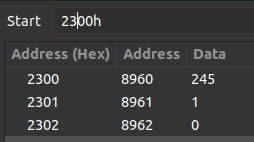
**(Q2)** Repeat 1 for BCD Numbers.

|  |
| --- |
| 4200 2A 00 022 LHLD 2200H; Load HL pair with direct addressing from memory   4203 0E 00 MVI C, 00H; Initiate C register with 0 for carry 4205 7D MOV A, L; Load first byte from HL pair into Accumulator   4206 84 ADD H; Addition with Second byte.    4207 27 DAA; Decimal adjust after addition for BCD conversion   4208 D2 0C 042 JNC NOCARRY; check is carry is generated 420B 0C INR C; Increment C register if CY is set   420C 32 02 022 NOCARRY: STA 2202H; Store Sum into 2202H 420F 79 MOV A,C; Move carry into accumulator   4210 32 03 022 STA 2203H; Store Carry   4213 76 HLT; |

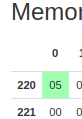
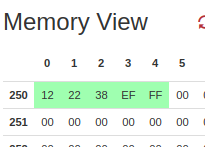
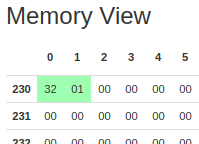
**(Q3)** Find Sum of N numbers stored in consecutive location starting from 2500H, the value of N is stored into 2200H, Store the result into 2300H and 2301H

|  |
| --- |
| 4200 21 00 025 LXI H,2500H; Store 2500H in HL register Pair, pointing N numbers  4203 0E 00 MVI C, 00H; Initiate Carry count   4205 3A 00 022 LDA 2200H; loading N into Accumulator   4208 47 MOV B, A; Move N to B register   4209 AF XRA A; Exclusive OR Accumulator with itself to clear it's value   420A 86 LOOP1: ADD M; Add Content of Memory location pointed by HL pair   420B D2 0F 042 JNC NOCARRY; Check if carry generated 420E 0C INR C; Increment C register if CY is set   420F 23 NOCARRY: INX H; Increment the HL pair( Extended register pair).   4210 05 DCR B; Decrement the N by 1 4211 C2 0A 042 JNZ LOOP1; If Zero flag is set, i.e. N is zero, we are done adding.   4214 32 00 023 STA 2300H; Store Sum into 2300H   4217 79 MOV A,C; 4218 32 01 023 STA 2301H; Store Carry into 2301H   421B 76 HLT; |

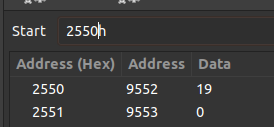
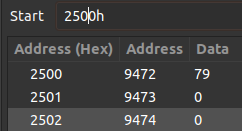
**(Q4)** Repeat 3 For BCD NUMBERS.

|  |
| --- |
| 4200 21 00 025 LXI H,2500H; Store 2500H in HL register Pair, pointing N numbers   4203 0E 00 MVI C, 00H; Initiate Carry count 4205 3A 00 022 LDA 2200H; loading N into Accumulator   4208 47 MOV B, A; Move N to B register   4209 AF XRA A; Exclusive OR Accumulator with itself to clear it's value   420A 86 LOOP1: ADD M; Add Content of Memory location pointed by HL pair   420B 27 DAA; Decimal adjust after addition for BCD conversion   420C D2 10 42 JNC NOCARRY; Check if carry generated 420F 0C INR C;  4210 23 NOCARRY: INX H; Increment the HL pair( Extended register pair). 4211 05 DCR B; Decrement the N by 1 4212 C2 0A 042 JNZ LOOP1; If Zero flag is set, i.e. N is zero, we are done adding.   4215 32 00 023 STA 2300H; Store Sum into 2300H 4218 79 MOV A,C;    4219 32 01 023 STA 2301H; Store Carry into 2301H 421C 76 HLT; |

**(Q5)** Find the sum of least significant 4 bits and Most Significant 4 bits of a byte stored in 2500H and store the result into 2550H

|  |
| --- |
| 2200 3A 00 025 LDA 2500H; Load Content from 2500H into accumulator  2203 57 MOV D, A; keeping backup due to multiple AND operation   2204 E6 0F ANI 00FH; AND operation with least 4 set bits( 0FH = 15( Decimal)= 00001111(Binary))   2206 47 MOV B,A; Cache least 4 bits into B register   2207 7A MOV A,D; Since content changed, load Accumulator freshly from backup    ; 4 times ROTATE LEFT ACCUMULATOR to take 4 MSB into LSB 2208 07 RLC; 2209 07 RLC; 220A 07 RLC; 220B 07 RLC;   220C E6 0F ANI 00FH; AND Immediate with 15 in decimal to get 4 most Significant bits   220E 80 ADD B; Add 4bit MSB to 4 bit LSB  220F 32 50 25 STA 2550H; Store the result in 2550H 2212 76 HLT; |

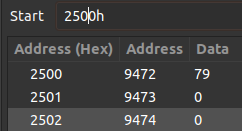
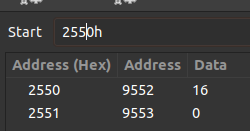
****

**79 = 0100 1111 (4 + 15)**

**(Q6)** Repeat 5 for BCD

|  |
| --- |
| 3200 3A 00 025 LDA 2500H; Load Content from 2500H into accumulator     ; fist have to achieve the binary representation of the BCD   3203 16 00 MVI D, 00H; Most sig bit of The BCD 3205 4F MOV C, A; Move hex to C reg   3206 AF XRA A; Clear Accumulator   3207 C6 01 LOOP: ADI 01; ADD immediate 01 with the Accumulator for the value Stored in C   3209 27 DAA; Subsequently it will convert into BCD 320A D2 0E 032 JNC NOCARRY; 320D 14 INR D; 320E 0D NOCARRY: DCR C; 320F C2 07 032 JNZ LOOP; 3212 82 ADD D; As a contains any overflowing bits;    ;A now contains the BCD  3213 57 MOV D,A; keeping backup due to multiple AND operation   3214 E6 0F ANI 00FH; AND operation with least 4 set bits( 0FH = 15( Decimal)= 00001111(Binary)) 3216 4F MOV C,A; Cache least 4 bits into C register   3217 7A MOV A,D; Since content changed, load Accumulator freshly from backup   3218 E6 F0 ANI 0F0H; AND immediate with F0H ie (11110000) to extract the most sig 4 bits   321A CA 21 32 JZ SKIP; if zero flag is set i.e. the number is less than 32 no need to rotate the content just to get 0   321D 07 RLC; 321E 07 RLC; 321F 07 RLC; 3220 07 RLC;   3221 81 SKIP: ADD C; Add 4bit MSB to 4 bit LSB    3222 32 50 25 STA 2550H; Store the result in 2550H 3225 76 HLT; |

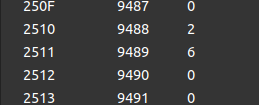
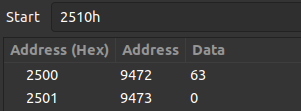
**79 in BCD 0111 1001 (7+9) = 16**

** **

**(Q7)** Write a Program to count number of 0’s and 1’s of a byte stored in 2500H, Store the result into 2510H and 2511H

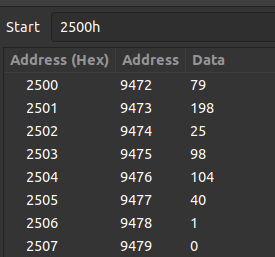
LDA 2500H; Load accumulator with the Number.

|  |
| --- |
| 4200 3A 00 025 LDA 2500H; Load accumulator with the Number.   4203 47 MOV B, A; Keeping a backup in B register 4204 0E 08 MVI C, 008H; Move Immediate 08 ( the number of bits) to C   4206 16 00 MVI D,00H; Initiating with 0 to count set bits   4208 E6 01 LOOP: ANI 001H; And operation with 01 to check the LSB is it set or not? 420A CA 0E 042 JZ NOSETBIT; is Zero flag is set, The the bit was not set 420D 14 INR D; Increase set bit count 420E 78 NOSETBIT: MOV A,B; Reloading the number to Accumulator   420F 07 RLC; Rotate left accumulator by 1 bit. 4210 47 MOV B,A; Store the new number to B for next use. 4211 0D DCR C; Decrease counter.  4212 C2 08 042 JNZ LOOP;   4215 3E 08 MVI A, 008H; For number of unset bits. 4217 92 SUB D; Substract the number of set bits from a byte(8 bit) 4218 32 10 25 STA 2510H; Store the unset bits count in 2610H 421B 7A MOV A, D; Mov D accumulator content to Accumulator. 421C 32 11 25 STA 2511H; Store set bits into 2511H   421F 76 HLT; |



(**Q8**) Write a program to add two 16 bit binary numbers.

|  |
| --- |
| 2200 21 00 025 LXI H, 2500H; Store 2500H in HL register pair, the first 16 bit number.  2203 16 00 MVI D,00; Initiating Carry Count 2205 01 02 025 LXI B, 2502H; Store 2502H in BC register pair, the second 26 bit number.   2208 0A LDAX B; Load accumulator with content pointed by BC register pair. 2209 86 ADD M; Add with Content pointed by HL pair 220A 32 04 025 STA 2504H; Store the sum of the first 8 bit to 2504H.    ; Increment the HL and BC pair( Extended register pair). 220D 23 INX H; 220E 03 INX B;   220F 0A LDAX B; Again Load accumulator with content pointed by BC register pair. 2210 8E ADC M; Add with Content pointed by HL pair and the carry(if generated) by previous addition.   2211 D2 15 22 JNC NOCARRY; Check if any carry generated 2214 14 INR D; Increase the carry count 2215 32 05 025 NOCARRY: STA 2505H; Store the next 8 bit sum to 2505H   2218 7A MOV A,D; Move D(carry) to accumulator 2219 32 06 025 STA 2506H; Store the result in 2500H    221C 76 HLT; |



(**Q9**) Repeat 8 for BCD Numbers.

|  |
| --- |
| 2200 21 00 025 LXI H, 2500H; Store 2500H in HL register pair, the first 16 bit number.  2203 01 02 025 LXI B, 2502H; Store 2502H in BC register pair, the second 26 bit number.   2206 16 00 MVI D, 00H; Initiate Carry Count. 2208 0A LDAX B; Load accumulator with content pointed by BC register pair. 2209 86 ADD M; Add with Content pointed by HL pair 220A 27 DAA; Decimal adjust after addition for BCD conversion   220B 32 04 025 STA 2504H; Store the sum of the first 8 bit to 2504H.    ; Increment the HL and BC pair( Extended register pair). 220E 23 INX H; 220F 03 INX B;   2210 0A LDAX B; Again Load accumulator with content pointed by BC register pair. 2211 8E ADC M; Add with Content pointed by HL pair and the carry(if generated) by previous addition. 2212 27 DAA; Decimal adjust after addition for BCD conversion 2213 D2 17 22 JNC NOCARRY; Check if any carry generated   2216 14 INR D; Increase the carry count 2217 32 05 025 NOCARRY: STA 2505H; Store the next 8 bit sum to 2505H 221A 7A MOV A,D; Move D(carry) to accumulator  221B 32 06 025 STA 2506H; Store the result in 2500H  221E 76 HLT; |

