Kuwait University
College of Engineering andPetroleum
Computer Engineering Department

CpE-<Course No. 0612363>: INTROD.TO EMBEDDED SYSTEMS

Semester: < Spring 2021-2022 >

Section < No. 04A/ATTEND >

Assignment No. < Mini Project >

Student Name: <Yassine Serrid, Ammar Kesseibi>

Student Id: <2181156439, 2181156215>

Instructor Name: <Dr. Faridah Ali>

TA Name: < Eng. Aish Dabees>

Date: <4th Of May 2022>

Contents:

Table Of Figures:

Tables:

Kuwait University

Computer Engineering Department

CpE-363 INTROD.TO EMBEDDED SYSTEMS

Spring 2021-2022

Car Maintenance System

Project Description

Title: *Design and implementation of an 8051-microcontroller based system to implement a **Car Maintenance System**.*

Summary Of The Problem Description:

In this project we are going to implement an 8051-program using Keil/Edsim51 to implement a Car Maintenance System following the requirements given the project description. To enhance programming skills in assembly language programming. And design and implement a software simple microcontroller-based application. To develop skills necessary for teamwork to accomplish a project.

1- Specifications:

This semester project is designing and implementing an 8051-microcontroller based system to implement ***Car Maintenance System***. The system will get its inputs from the memory locations we fill them with value before running the program.

1.1- How the System Work (Project Description):

1. In the beginning of the program, we will store the user selected values in the memory locations given in the project description.
2. In the memory location 50H we are going to store the desired category there is two possibilities either 1 for selection category 1 or 2 for selection category 2.
3. In the memory location 51H we are going to store the desired option for the user if he wants the maintenance services either 2 for individual car or 1 for group of cars
4. If the user selected a group of cars, we will the store the number of cars in the memory location 52H according to out limitations of the seat numbers for
5. If we store value 1 in the 50H memory location the user selected category 1:
6. Check The Value In 51H (Either 1 for more than car or 2 for one car).

5.1. The user selected (2) for individual car:

5.1.1. Check Inputs Of Additional Services

5.1.2. Decrement the slots of category 1 stored in location 59H by one.

5.1.3. MOV The Cost Of Category 1 Stored In 63H For One Car To The Final Cost Stored In 58H

5.1.4. Golden Scent (Either 1 For Yes Or 2 For No) Stored In 53H

5.1.5. The user Selected 1 (Yes) for Golden Scent:

5.1.6. ADD The Cost Of This Addition Stored In 5BH Then Add It To The Final Cost Stored In 58H And Convert It To Decimal

5.1.7. The user Selected 2 (No) for Golden Scent:

5.1.8 Convert The Cost Stored In 58H To Decimal

5.1.9. END

5.2. The user selected (1) for group of cars:

5.2.1. If Number Of Cars Stored In 52H Greater Than 65 End The Program

5.2.2. Check The Number Of Cars Stored In 52H > 1 if =1 ACALL Individual AND < 65

5.2.3. Check Inputs Of Additional Services

5.2.4. DEC Category 1 Slots Stored In Location 59H According To The Number Of Cars And Convert It To Decimal

5.2.5. MUL The No. Of Cars with The Cost Of This Addition Stored In 5BH Then ADD It To The Final Result Stored In 58H And Convert It To Decimal

5.2.6. Golden Scent (Either 1 For Yes Or 2 For No) Stored In 53H

5.2.7. The user Selected 1 (Yes) for Golden Scent:

5.2.8. MUL The No. Of Cars with The Cost Of This Addition Stored In 5BH Then ADD It To The Final Result Stored In 58H

5.2.9. The user Selected 2 (No) for Golden Scent:

5.2.10. Convert The Cost Stored In 58H To Decimal

5.2.11. END

5.3. The user selected a value =! 1 OR 2:

5.3.1. END

7. If we store value 2 in the 50H memory location the user selected category 2:
8. Check The Value In 51H (Either 1 for more than car or 2 for one car).
   8.1. The user selected (2) for induvial car:
   8.1.1. Check Inputs Of Additional Services
   8.1.2. DEC Category 2 Slots Stored In Location 5AH By One And Convert It To Decimal

   8.1.3. MOV The Cost Of Category 2 Stored In 61H For One Car To The Final Cost Stored In 58H

   8.1.4. Remove The Old Brake Shoe (Either 1 For Yes Or 2 For No) Stored In 54H

   8.1.5. The user Selected 1 (Yes) for Remove The Old Brake Shoe:

   8.1.6. ADD The Cost Of This Addition Stored In 5CH Then Add It To The Final Cost Stored In 58H And Convert It To Decimal

   8.1.7. The user Selected 2 (No) for Remove The Old Brake Shoe:

   8.1.8. Check Next Additional Service

   8.1.9. The user selected a value =! 1 OR 2:

   8.1.10. END

   8.1.11. Unscrew The Wheels (Either 1 For Yes Or 2 For No) Stored In 55H

   8.1.12. The user Selected 1 (Yes) for Unscrew The Wheels:

   8.1.13. ADD The Cost Of This Addition Stored In 5DH Then Add It To The Final Cost Stored In 58H And Convert It To Decimal

   8.1.14. The user Selected 2 (No) for Unscrew The Wheels:

   8.1.15. Check Next Additional Service

   8.1.16. The user selected a value =! 1 OR 2:

   8.1.17. END

   8.1.18. Install New Brake Pads (Either 1 For Yes Or 2 For No) Stored In 56H

   8.1.19. The user Selected 1 (Yes) for Install New Brake Pads:

   8.1.20. ADD The Cost Of This Addition Stored In 5EH Then Add It To The Final Cost Stored In 58H And Convert It To Decimal

   8.1.21. The user Selected 2 (No) for Remove The Old Brake Shoe:

   8.1.22. Check Next Additional Service

8.1.23. The user selected a value =! 1 OR 2:

8.1.24. END

8.1.25. Check Master Cylinder (Either 1 For Yes Or 2 For No) Stored In 57H

8.1.26. The user Selected 1 (Yes) for Check Master Cylinder:

8.1.27. ADD The Cost Of This Addition Stored In 5FH Then Add It To The Final Cost Stored In 58H And Convert It To Decimal

8.1.28. The user Selected 2 (No) for Check Master Cylinder:

8.1.29. Convert The Cost Stored In 58H To Decimal

8.1.30. END

8.1.31. The user selected a value =! 1 OR 2:

8.1.32. END

8.2. The user selected (1) for group of cars:

8.2.1. Check The Number Of Cars Stored In 52H > 1 if =1 ACALL Individual AND < 80

8.2.4. Check Inputs Of Additional Services

8.2.2. DEC Category 2 Slots Stored In Location 5AH According To The Number Of Cars

8.2.3. MUL The Cost Of Category 2 Stored In 60H Group Of Cars And MUL Them With The Number Of Cars To The Final Cost Stored In 58H

8.2.5. Remove The Old Brake Shoe (Either 1 For Yes Or 2 For No) Stored In 54H

8.2.6. The user Selected 1 (Yes) for Remove The Old Brake Shoe:

8.2.7. MUL The No. Of Cars with The Cost Of This Addition Stored In 5CH Then ADD It To The Final Result Stored In 58H Convert It To Decimal

8.2.8. The user Selected 2 (No) for Remove The Old Brake Shoe:

8.2.9. Check Next Additional Service

8.1.23. The user selected a value =! 1 OR 2:

8.1.24. END

8.2.10. Unscrew The Wheels (Either 1 For Yes Or 2 For No) Stored In 55H

8.2.11. The user Selected 1 (Yes) for Unscrew The Wheels:

8.2.12. MUL The No. Of Cars with The Cost Of This Addition Stored In 5DH Then ADD It To The Final Result Stored In 58H Convert It To Decimal

8.2.13. The user Selected 2 (No) for Unscrew The Wheels:

8.2.14. Check Next Additional Service

8.1.23. The user selected a value =! 1 OR 2:

8.1.24. END

8.2.15. Install New Brake Pads (Either 1 For Yes Or 2 For No) Stored In 56H

8.2.16. The user Selected 1 (Yes) for Install New Brake Pads:

8.2.17. MUL The No. Of Cars with The Cost Of This Addition Stored In 5EH Then ADD It To The Final Result Stored In 58H Convert It To Decimal

8.2.18. The user Selected 2 (No) for Remove The Old Brake Shoe:

8.2.19. Check Next Additional Service

8.1.23. The user selected a value =! 1 OR 2:

8.1.24. END

8.2.20. Check Master Cylinder (Either 1 For Yes Or 2 For No) Stored In 57H

8.2.21. The user Selected 1 (Yes) for Check Master Cylinder:

8.2.22. MUL The No. Of Cars with The Cost Of This Addition Stored In 5FH Then ADD It To The Final Result Stored In 58H Convert It To Decimal

8.2.23. The user Selected 2 (No) for Check Master Cylinder:

8.1.29. Convert The Cost Stored In 58H To Decimal

8.2.24. END

8.3. The user selected a value =! 1 OR 2:

5.3.1. END

9. Check The Value In 51H (Either 1 for more than car or 2 for one car if != 1 or 2).
10. END

1.2- System Data ,Limitations And Error Handling:

In our implementation we assuming that will input the number of seats, the category and the cost of each category and service according to the selected category will be counted based on the following tables as following:

Table 1. Initial Ticket Cost.

Category	Price for each Car (KD)	Number of Available Seats
1- Oil Change &Spark Plug Replacement	Group of Cars: 20 KD- Individual Car 25 KD	65
2- Engine Oil & Battery Check& Brakes Check	Group of Cars: 10 KD- Individual Car 15 KD	80

Table 2. Additional Services Cost.

Category	Service	Cost (KD)
1	1. Golden Scent	10
2	1. Remove the Old Brake Shoe	30
	2. Unscrew the wheels	20
	3. Install new Brake Pads	50
	4. Check master cylinder	15

Some of the Limitations in the program:

1- The user can't choose number of cars greater than the number of slots available in the category

2- If the user selected group of cars option and stored the value 1 in memory location 52H ACALL directly individual procedure.

3- Take in consideration (Large number of slots selected and additional services) the result of multiplication may exceeds 8-Bits store the remaining the memory location 48H.

3.1- The largest cost is 10,000 KD for buying all slots of category 2 with its additional services which is (0010 0111) (0001 0000) in binary or 2710H, when we store in the 58H which is the total cost it will be Only 00 because the memory location only takes 8 bits and when we convert it to BCD it will be 4bits+4bits = 00

4- Take in consideration the CY (Carry Flag) and clear it when required.

5- Watch out the inputs from user it should follow the requirements.

2- Project Implementation:

1- The system must be implemented using 8051 assembly language using Keil Or Edsim51.

2- After developing the complete codes sequence we are going to calculate and analysis the code based on their execution time in term of machine cycle (MC) and memory required to store the program in terms of bytes. First we are going to calculate for each instruction it's machine cycle (MC) and the number of bytes required to store the program in the ROM. After that we will sum the machine cycle (MC) and the number of bytes for the whole program.

3- We calculated for each line it's required bytes and MC as a comment in the program, in calculating the machine cycle we used Crystal frequency X_{tal} = 11.0592MHZ ➔ 12/11,0592MHZ = 1.085us

4-

2 x 1.085 = 2.17 x 156 -> 338.52

1 x 1.085 = 1.085 x 141 -> 152.985

4 x 1.085 = 4.34 x 8 -> 34.72

8 x 1.085 = 8.68 x 26 -> 225.68

16 x 1.085 = 17.36 x 4 -> 69.44

total time delay = 821.345 us

total execution time (Machine cycle MC) = 2 x 156 + 1 x 141 + 4 x 8 + 8 x 26 + 16 x 4 = 757

byte:1 x 122

byte:2 x 166 -> 332

byte:3 x 47 -> 141

total memory required to store the program in terms of bytes = 595 bytes

Table 3. Total MC Cycle and Memory Required To Store The Program.

MC	Bytes
Time delay = T_{MC} x 1.085 Time delay = 757 x 1.085 = 821.345 us	595 Bytes

3- Appendix:

3.1-The Program Flow Chart:

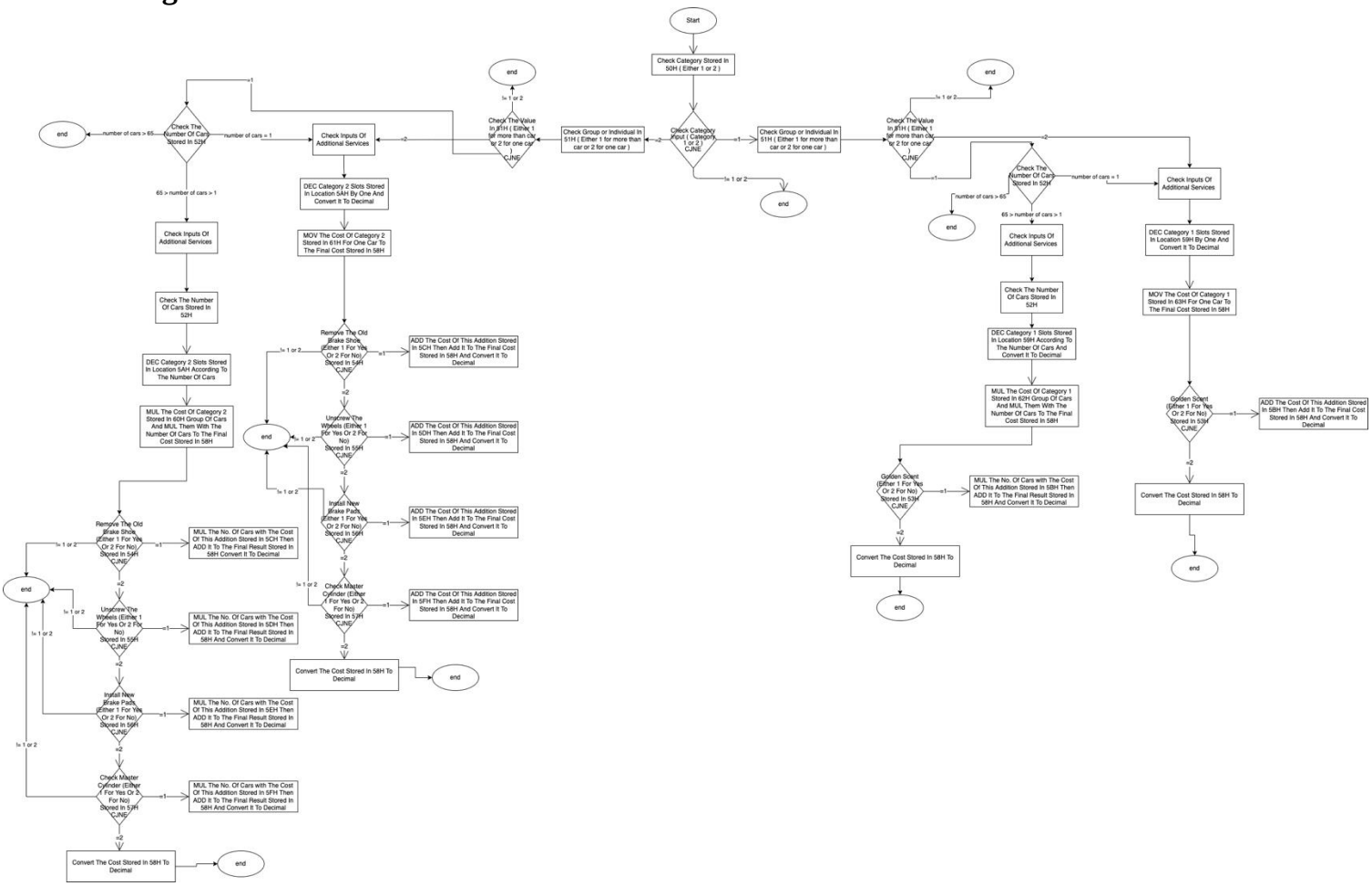

Figure 1. Flow Chart Of The 8051 Program.

3.2- Test Cases:

3.2.1- Best Case: (Selecting Category 2 for individual car without buying extra services “Least Cost” = 15KD)

(The available Slots = 1 – no of available slots to this category = 79)

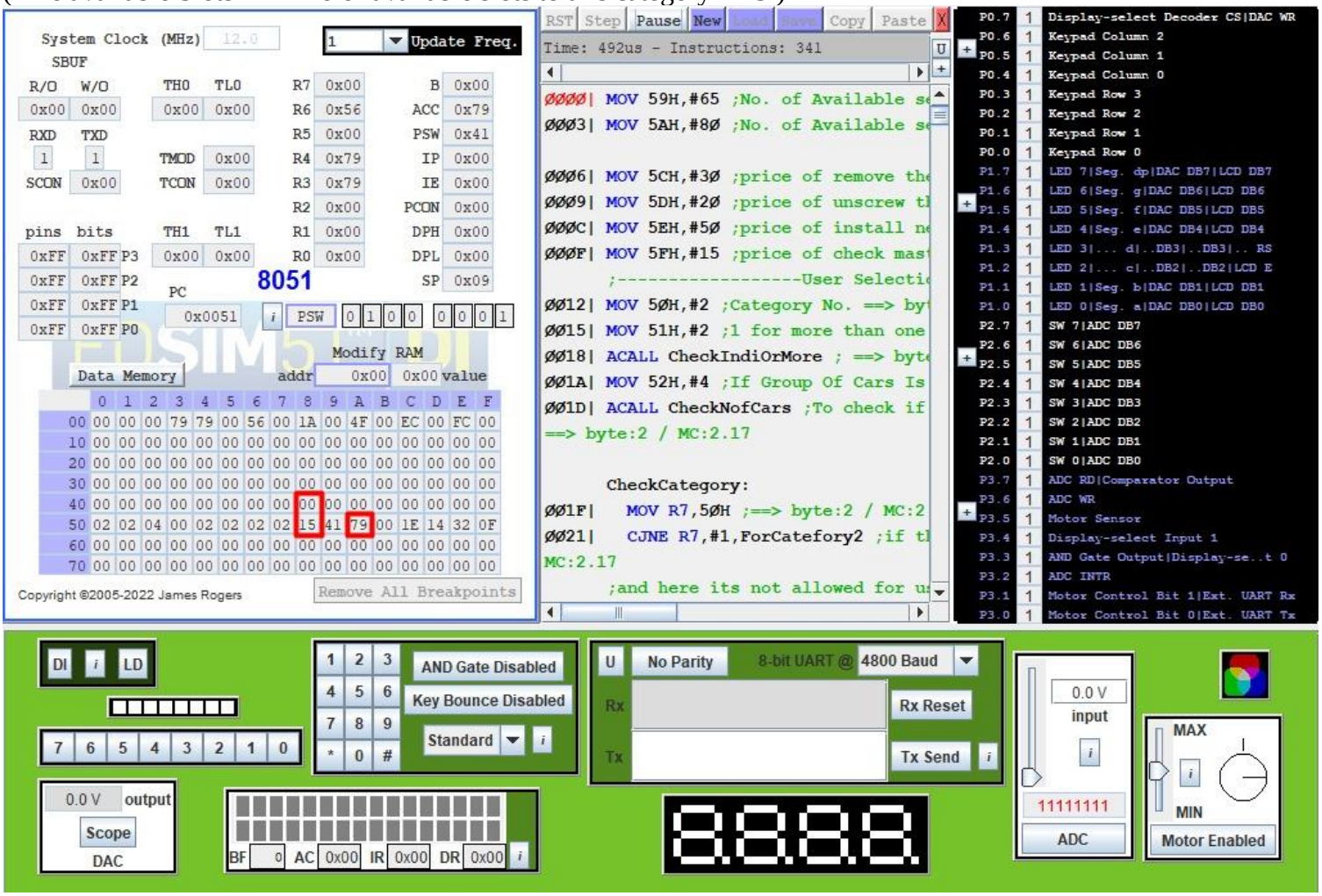

Figure 2. Best Case Outputs Of Memory Locations.

3.2.2- Worst Case: (Selecting Category 2 80 cars and buy all its additional services “Greatest Cost” = 10000KD = 2710H)

(The available Slots = 80 no of cars – no of available slots to this category = 0)

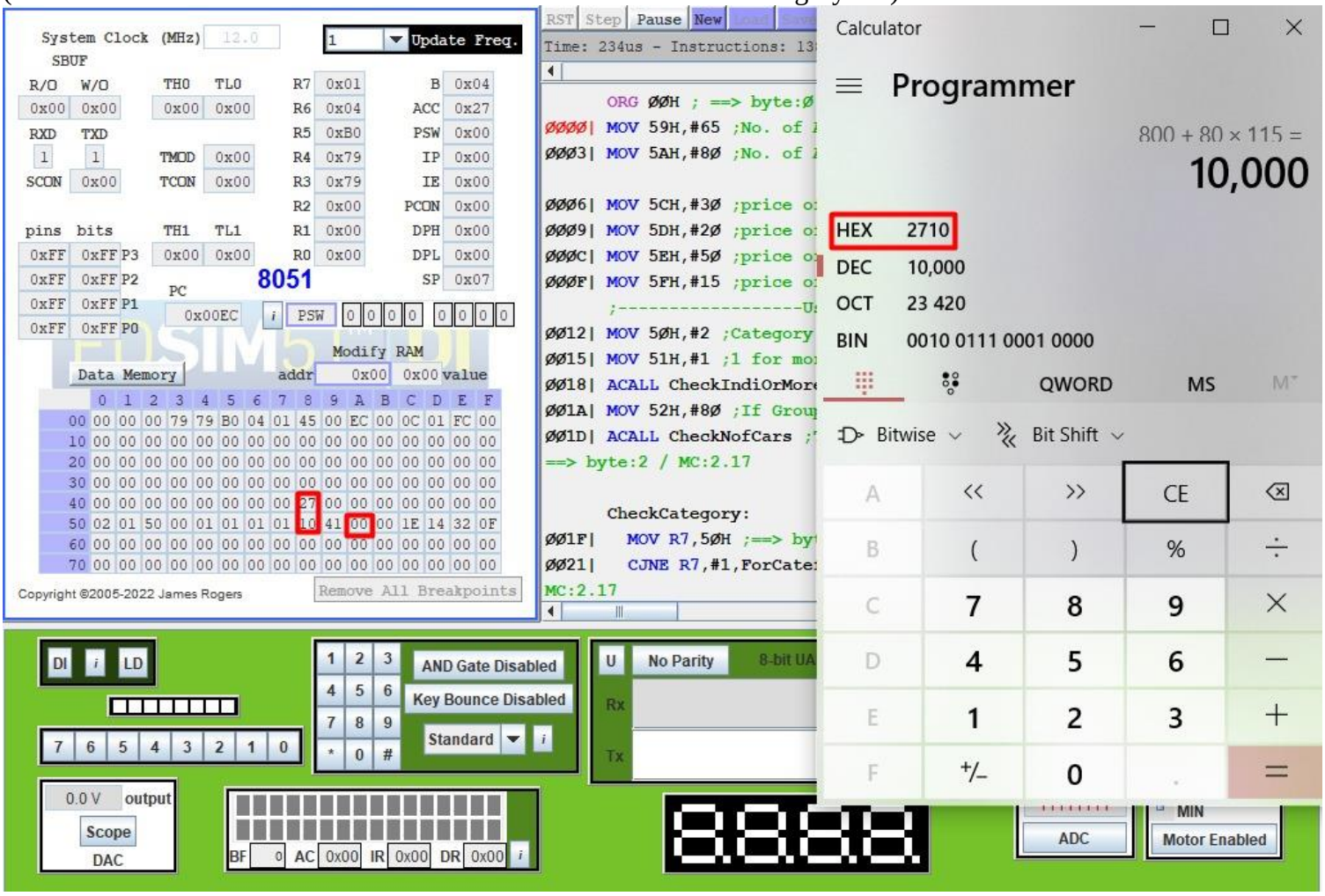

Figure 3. Worst Case Outputs of Memory Locations.

3.2.3- Random Case1: (Selecting Category 1 for 7 cars with buying the additional service = 210KD)

(The available Slots = 7 no of cars – no of available slots to this category = 58)

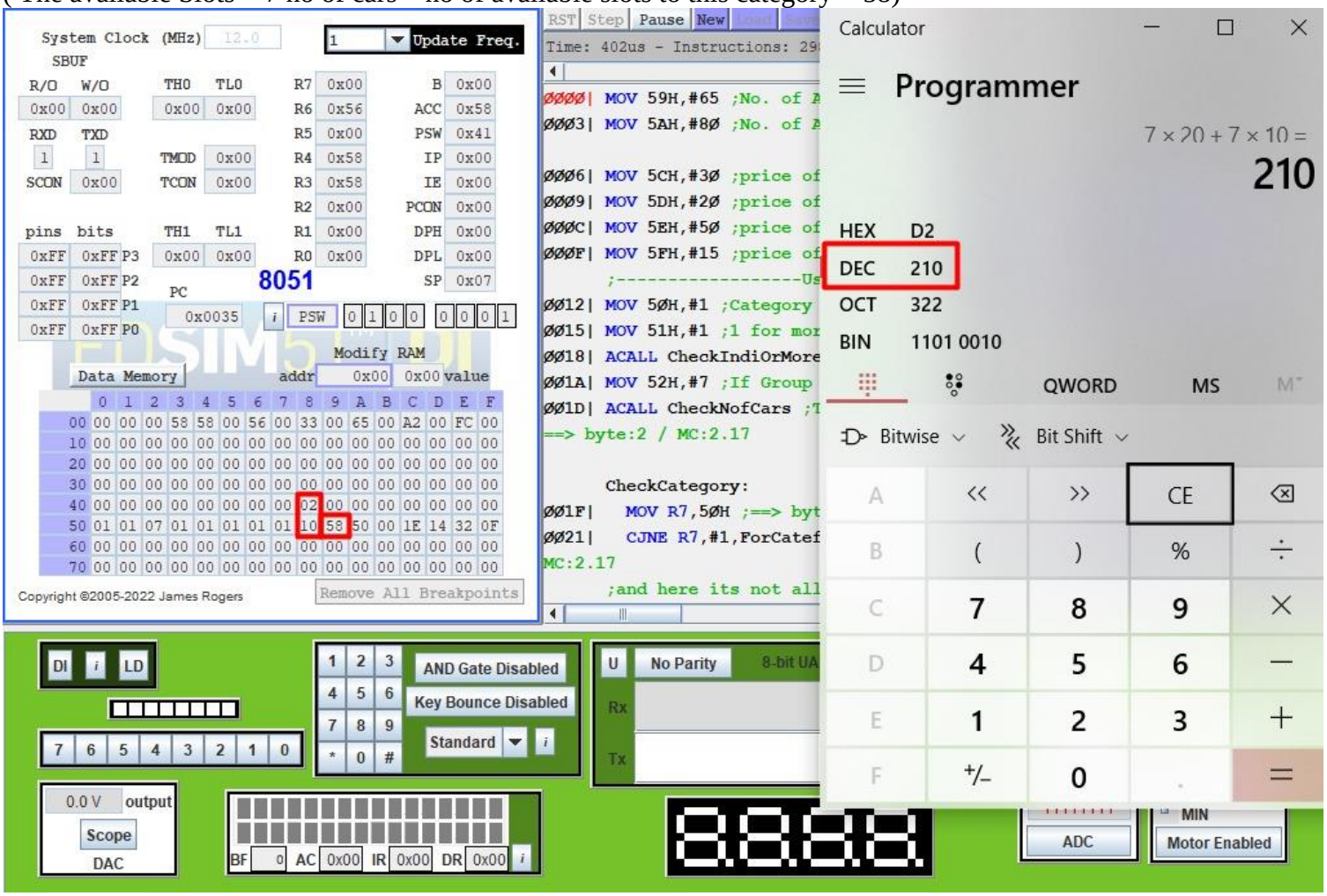

Figure 4. Random Case 1: Outputs of Memory Locations.

3.2.4- Random Case2: (Selecting Category 1 for individual car with the additional service = 35KD)

(The available Slots = 1 – no of available slots to this category = 64)

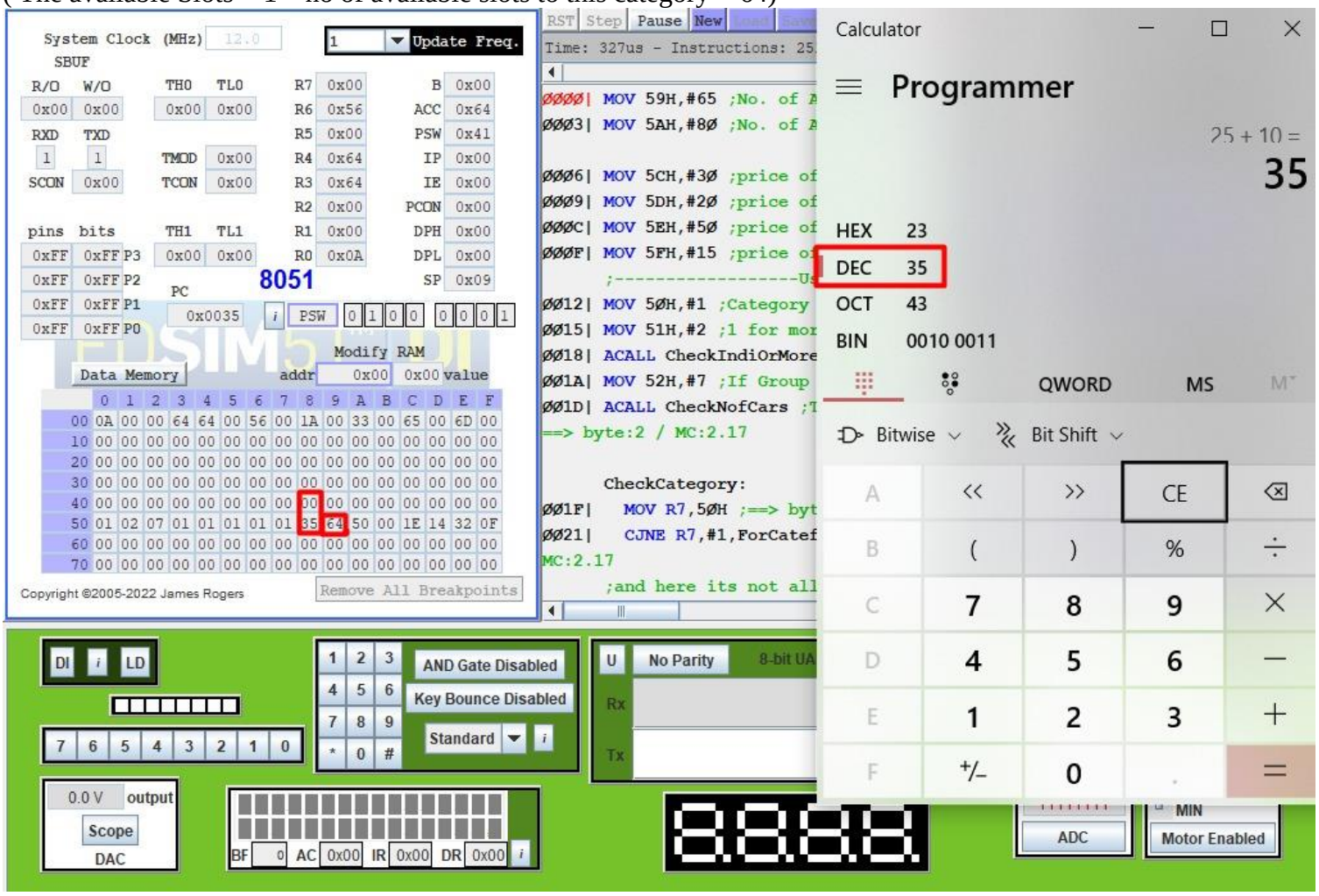

Figure 5. Random Case 2: Outputs of Memory Locations.

3.3- The Source Code Of The 8051 Program:

```
ORG 00H ; ==> byte:0 / MC:0 Final
MOV 59H,#65 ;No. of Available seats For Category 1 ==> byte:3 / MC:2.17
MOV 5AH,#80 ;No. of Available seats For Category 2 ==> byte:3 / MC:2.17

MOV 5CH,#30 ;price of remove the old brake ==> byte:3 / MC:2.17
MOV 5DH,#20 ;price of unscrew the wheels ==> byte:3 / MC:2.17
MOV 5EH,#50 ;price of install new brake ==> byte:3 / MC:2.17
MOV 5FH,#15 ;price of check master cylinder ==> byte:3 / MC:2.17
;------------------User Selection To Start Up----------------------;
MOV 50H,#2 ;Category No. ==> byte:3 / MC:2.17
MOV 51H,#1 ;1 for more than one car 2 individual ==> byte:3 / MC:2.17
ACALL CheckIndiOrMore ; ==> byte:2 / MC:2.17
MOV 52H,#8 ;If Group Of Cars Is Choosen This Is The Number Of Cars ==> byte:3 /
MC:2.17
ACALL CheckNofCars ;To check if the number of Car is 1 it will convert 51H
automatically to individual ==> byte:2 / MC:2.17

CheckCategory:
     MOV R7,50H ;==> byte:2 / MC:2.17
     CJNE R7,#1,ForCatefory2 ;if the user selected category 2 go to category 2
selections ==> byte:3 / MC:2.17
;and here its not allowed for user select the num of cars if he selected individual

;------------------User Selection For Status Of Category1: For Additional Services
START----------------------;
ForCatefory1:
     MOV 53H,#1 ;1 For Yes For Additional Service And 2 For No ==> byte:3 /
MC:2.17
     ACALL Category1 ;go to catogery 1 ==> byte:2 / MC:2.17
     MOV R3,58H ; ==> byte:2 / MC:2.17
     ACALL CeckToConvert ;to convert to decimal==> byte:2 / MC:2.17
     MOV 58H,R3 ; ==> byte:2 / MC:2.17
     MOV R3,59H ; ==> byte:2 / MC:2.17
     ACALL ConvertToDecimalLower2 ;==> byte:2 / MC:2.17
     MOV 59H,R3 ; ==> byte:2 / MC:2.17
     SJMP $ ; ==> byte:2 / MC:2.17
;------------------User Selection For Status Of Category2: For Additional Services
START----------------------;
ForCatefory2:
     MOV 54H,#2 ;1 for yes to remove the old brake 2 for no ==> byte:3 / MC:2.17
     MOV 55H,#1 ;1 for yes to unscrew the wheels 2 for no ==> byte:3 / MC:2.17
     MOV 56H,#2 ;1 for yes to install new brake 2 for no ==> byte:3 / MC:2.17
     MOV 57H,#2 ;1 for yes to check master cylinder 2 for no ==> byte:3 / MC:2.17
     ACALL Category2 ; ==> byte:2 / MC:2.17
     MOV R3,58H ; ==> byte:2 / MC:2.17
     ACALL CeckToConvert ;to convert to decimal ==> byte:2 / MC:2.17
     MOV 58H,R3 ; ==> byte:2 / MC:2.17
     MOV R3,5AH ; ==> byte:2 / MC:2.17
     ACALL ConvertToDecimalLower2 ;to convert to decimal==> byte:2 / MC:2.17
     MOV 5AH,R3 ; ==> byte:2 / MC:2.17
     SJMP $  ;==> byte:2 / MC:2.17
```

```
;------------------User Selection For Status Of Category2: For Additional Services
END-----------------------;
CheckNofCars:
      MOV A,52H ;==> byte:2 / MC:1.085
      XRL A,#01H;to check if the user selected the num of card = 1 it will convert
51H automatically to individual ==> byte:2 / MC:1.085
      JNZ Good ;if the num of cars > 1 ==> byte:2 / MC:2.17
      MOV 51H,#2 ;==> byte:3 / MC:2.17
Good:
      RET ;==> byte:1 / MC:2.17

CheckIndiOrMore:
      MOV R7,51H ;==> byte:2 / MC:2.17
      CJNE R7,#1,CheckCategory  ;==> byte:3 / MC:2.17
      RET


;------------------Category1: START----------------------;
Category1:
      ACALL IndividualCategory1 ;Go To The IndividualCategory1 ==> byte:2 / MC:2.17
      RET ; ==> byte:1 / MC:2.17
;------------------Category1: Individual Car START----------------------;
IndividualCategory1:
      MOV R6,51H ;Check If 1 Car Or One Car ==> byte:2 / MC:2.17
      CJNE R6,#2,GroupCategory1 ;Check If 1 Car Or One Car if 51H != 2 It Means One
Car ==> byte:3 / MC:2.17
      ACALL CheckGoldenScentIndividual ;It Mean 51H = 1 ==> byte:2 / MC:2.17
      RET ; ==> byte:1 / MC:2.17
CheckGoldenScentIndividual:
      MOV A,#65 ;The Number Of Slots Of Category1 ==> byte:2 / MC:1.085
      MOV R0,#1 ;Since It's One Car ==> byte:2 / MC:1.085
      SUBB A,R0 ;The Number Of Slots Of Category1 - 1 ==> byte:1 / MC:1.085
      MOV 59H,A ;Store The Resuilt In 59H ==> byte:2 / MC:1.085
      CLR A ;Clear A ==> byte:1 / MC:1.085
      MOV R0,#0 ;Clear R0 ==> byte:2 / MC:1.085
      MOV R5,53H ; Check If Additional Service Needed From The Inputs ==> byte:2 /
MC:2.17
      CJNE R5,#1,OnlyTheCategoryCostIndividual ; If 53H != 1 It Mean We Only Take
The Category Cost In Consedertion ==> byte:3 / MC:2.17
      MOV R0,#10 ;Put The Cost Of The Additional Service In Register R0 ==> byte:2
/ MC:1.085
      MOV A,#25 ;We Store The Cost Category 1 Of One Car In A ==> byte:2 / MC:1.085
      ADD A,R0 ; Cost Of The Additional Service + Cost Category 1 Of One Car ==>
byte:1 / MC:1.085
      MOV 58H,A ;Store The Value  ==> byte:2 / MC:1.085
      CLR A ;Clear A ==> byte:1 / MC:1.085
      RET ; ==> byte:1 / MC:2.17
OnlyTheCategoryCostIndividual:
      MOV 58H,#25 ;We Store The Cost Category 1 Of One Car In 58H Memory Location
As Required ==> byte:3 / MC:2.17
      RET ; ==> byte:1 / MC:2.17
;------------------Category1: Individual Car END----------------------;

;------------------Category1: Group Of Cars START----------------------;
```

```
GroupCategory1:
      MOV R6,51H ;Check If 1 Car Or One Car ==> byte:2 / MC:2.17
      CJNE R6,#1,ENDD1 ;Check If 1 Car Or more than One Car if 51H != 1 It Means
the user selection Ain't 1 or 2 ==> byte:3 / MC:2.17
      CLR C ;Clear The Carry Flag ==> byte:1 / MC:1.085
      MOV R5,52H ;==> byte:2 / MC:2.17
      MOV A,#66 ;since the number of available seats is 65 ==> byte:2 / MC:1.085
      SUBB A,R5 ;check if the number of cars greater than 65 ==> byte:1 / MC:1.085
      JB Acc.7,ErrorCategory1 ;go to ErrorCategory1 if the result of sub of the No.
of cars from the available seats is positive ==> byte:3 / MC:2.17
      MOV A,#65 ;The Number Of Slots Of Category1 ==> byte:2 / MC:1.085
      MOV R0,52H;Number Of Cars Selected ==> byte:2 / MC:2.17
      SUBB A,R0 ;Number Of Slots - Number Of Cars Selected ==> byte:1 / MC:1.085
      MOV 59H,A ;Store The Resuilt In 59H ==> byte:2 / MC:1.085
      CLR A     ;Clear A ==> byte:1 / MC:1.085
      ACALL CheckGoldenScentGroup;ACALL CheckGoldenScentGroup ==> byte:2 / MC:2.17
      RET ;==> byte:1 / MC:2.17

ErrorCategory1:
      CLR A ;if the numnber of cars is greater than 65 ==> byte:1 / MC:1.085

ENDD1:
      SJMP $ ;==> byte:2 / MC:2.17

CheckGoldenScentGroup:
      MOV R3,53H ; ==> byte:2 / MC:2.17
      CJNE R3,#1,OnlyTheCategoryCostGroup ;Check If With Additional Services Or No
==> byte:3 / MC:2.17
      MOV B,52H ;Put The Number Of Cars In The Register B ==> byte:2 / MC:2.17
      MOV A,#10 ;Put The Cost Of The Additional Service In Register A ==> byte:2 /
MC:1.085
      MUL AB ;Number Of Cars x Cost Of The Additional Service ==> byte:1 / MC:4.34
      MOV 48H,B ;==> byte:2 / MC:2.17
      MOV R0,A ;Store The Resuilt In The R0 ==> byte:1 / MC:1.085
      CLR A ;Clear A ==> byte:1 / MC:1.085
      MOV B,#0H ;Clear B ==> byte:2 / MC:1.085
      MOV A,52H ;Put The Number Of Cars In The Register A ==> byte:2 / MC:1.085
      MOV B,#20 ;The Cost Of The Category for group ==> byte:2 / MC:1.085
      MUL AB ;Number Of Cars x Cost Of The Category ==> byte:1 / MC:4.34
      MOV R7,A ;==> byte:21 / MC:1.085
      MOV A,48H; ==> byte:2 / MC:1.085
      ADD A,B; ==> byte:1 / MC:1.085
      MOV 48H,A; ==> byte:2 / MC:1.085
      MOV A,R7; ==> byte:1 / MC:1.085
      ADD A,R0 ;(Number Of Cars x Cost Of The Additional Service) + (Number Of Cars
x Cost Of The Category) byte:1 / MC:1.085
      JC Loop1 ;==>byte:2 / MC:2.17
      SJMP Loop2 ;==> byte:1 / MC:1.085
      Loop1:
      INC 48H ;==> byte:2 / MC:1.085
      Loop2:
      MOV 58H,A;Store The Resuilt In The 58H ==> byte:2 / MC:1.085
      DEC A; ==> byte:1 / MC:1.085
      CLR A; ==> byte:1 / MC:1.085
```

```
      MOV R0,#0; ==> byte:2 / MC:1.085
      RET; ==> byte:1 / MC:2.17
OnlyTheCategoryCostGroup:
      MOV A,52H ;Put The Number Of Cars In The Register A ==> byte:2 / MC:1.085
      MOV B,#20 ;The Cost Of The Category ==> byte:2 / MC:1.085
      MUL AB ;Number Of Cars x Cost Of The Category ==> byte:1 / MC:4.34
      MOV 48H,B ;==> byte:2 / MC:2.17
      MOV 58H,A ;Store The Resuilt In The 58H ==> byte:2 / MC:1.085
      CLR A ;Clear A ==> byte:1 / MC:1.085
      MOV B,#0H ;Clear B ==> byte:2 / MC:1.085
      RET; ==> byte:1 / MC:2.17
;------------------Category1: Group Of Cars END----------------------;

;------------------Category1: END----------------------;

;------------------Category2: START----------------------;
Category2:
      MOV R7,50H ;==> byte:2 / MC:2.17
      CJNE R7,#2,ENDD2 ;Check The Gategory Input ( 1 or 2 ) if 50H != 2 Go To ENDD
==> byte:3 / MC:2.17
      ACALL IndividualCategory2;==> byte:2 / MC:2.17
      RET;==> byte:1 / MC:2.17

IndividualCategory2:
      MOV R6,51H ;Check If 1 Car Or One Car ==> byte:2 / MC:2.17
      CJNE R6,#2,GroupCategory2 ;Check If 1 Car Or One Car if 51H != 2 It Means One
Car ==> byte:3 / MC:2.17
      ACALL CheckEngineOil ;==> byte:2 / MC:2.17
      ACALL RemoveBrake ;==> byte:2 / MC:2.17
      ACALL Unscrew ;==> byte:2 / MC:2.17
      ACALL Install ;==> byte:2 / MC:2.17
      ACALL MasterCylinder ;==> byte:2 / MC:2.17
      RET ;==> byte:1 / MC:2.17

GroupCategory2:
      MOV R6,51H ;Check If 1 Car Or One Car ==> byte:2 / MC:2.17
      CJNE R6,#1,ENDD2 ;Check If 1 Car Or One Car if 51H != 2 It Means One Car ==>
byte:3 / MC:2.17
      ACALL CheckEngineOil ;==> byte:2 / MC:2.17
      ACALL RemoveBrake ;==> byte:2 / MC:2.17
      ACALL Unscrew ;==> byte:2 / MC:2.17
      ACALL Install ;==> byte:2 / MC:2.17
      ACALL MasterCylinder ;==> byte:2 / MC:2.17
      RET ;==> byte:1 / MC:2.17

ENDD2:
      SJMP $ ;==> byte:2 / MC:2.17

CheckEngineOil:
      MOV R7,51H ;==> byte:2 / MC:2.17
      CJNE R7,#1,IndividualEngineOil ;if the value in 51H != 1 thats mean one
car==> byte:3 / MC:2.17
      SJMP MoreEngineOil ;if the value in 51H=1 thats mean more than 1 car ==>
byte:2 / MC:2.17
```

```
IndividualEngineOil:
      MOV 58H,#15 ;the price of Engine oil for 1 car ;==> byte:3 / MC:2.17
      MOV R6,5AH ;the No. of available seats ;==> byte:2 / MC:2.17
      DEC R6 ;Decrement the available seats by 1 ==> byte:1 / MC:1.085
      MOV 5AH,R6 ;==> byte:2 / MC:2.17
      RET ;==> byte:1 / MC:2.17
MoreEngineOil:
      CLR C ;==> byte:1 / MC:1.085
      MOV R7,#10 ;the price of seats for more than one car==> byte:2 / MC:1.085
      MOV R5,52H;the number of cars ==> byte:2 / MC:2.17
      ;------------------To check if the number of cars > avalable seats end the
program----------------------;
      MOV A,#81;==> byte:2 / MC:1.085
      SUBB A,R5;==> byte:1 / MC:1.085
      JB Acc.7,ErrorCategory2 ;if the number is negative thats mean the number of
cars > 80==> byte:3 / MC:2.17
      ;-------------------------------------------------------------------------
---------------------------;
      CLR A ;==> byte:1 / MC:1.085
      CLR C ;==> byte:1 / MC:1.085
      MOV R6,52H ;num of cars ==> byte:2 / MC:2.17
      MOV A,5AH ;available seats==> byte:2 / MC:1.085
      SUBB A,R6 ;available seats - num of cars ==> byte:1 / MC:1.085
      MOV 5AH,A ;restore it ==> byte:2 / MC:1.085
      CLR C ;==> byte:1 / MC:1.085
      MOV A,R7 ;==> byte:1 / MC:1.085
      MOV B,R6 ;==> byte:1 / MC:1.085
      MUL AB ;price of engine oil * num of cars ==> byte:1 / MC:4.34
      MOV 58H,A;store the cost in 58H ==> byte:2 / MC:1.085
      MOV 48H,B ;store the high bits in 58H==> byte:2 / MC:2.17
      RET ;==> byte:1 / MC:2.17
ErrorCategory2:
      CLR A ;==> byte:1 / MC:1.085
      SJMP $ ;if the num of car > available seats==> byte:2 / MC:2.17

RemoveBrake:
      MOV R5,54H ;to check 1 for yes for remove breake 2 for no==> byte:2 / MC:2.17
      CJNE R5,#1,MEOW3 ;if not 1 go to meow 3==> byte:3 / MC:2.17
      MOV R7,51H ;to check if individual or more than one car==> byte:2 / MC:2.17
      CJNE R7,#1,IndividualRemoveBrake ;==> byte:3 / MC:2.17
      SJMP MoreRemoveBrake ;==> byte:2 / MC:2.17
IndividualRemoveBrake:
      MOV A,58H ;==> byte:2 / MC:1.085
      ADD A,5CH ;add the price of remove brake==> byte:2 / MC:1.085
      MOV 58H,A ;move the new cost in 58H==> byte:2 / MC:1.085
      RET ;==> byte:1 / MC:2.17
MoreRemoveBrake:
      CLR C ;==> byte:1 / MC:1.085
      MOV A,52H ;the num of cars ==> byte:2 / MC:1.085
      MOV B,5CH ;price of remove brake ==> byte:2 / MC:2.17
      MUL AB ;price of remove brake * num of cars==> byte:1 / MC:4.34
      MOV R5,A ;==> byte:1 / MC:1.085
      MOV R6,B ;byte:1 / MC:1.085
      MOV A,58H ;==> byte:2 / MC:1.085
```

```
      ADD A,R5 ;the initial cost + the price of remove brake * mun of cars ==>
byte:1 / MC:1.085
      JNC MEOW1 ;==> byte:2 / MC:2.17
      INC 48H ;==> byte:2 / MC:1.085
MEOW1:
      MOV 58H,A ;if there is no carry just store the result in 58H ==> byte:2 /
MC:1.085
      MOV A,48H ;==> byte:2 / MC:1.085
      ADD A,R6 ;==> byte:1 / MC:1.085
      MOV 48H,A ;==> byte:2 / MC:1.085
      RET;==> byte:1 / MC:2.17
MEOW3:
      MOV R5,54H ;if 2 for remove brake return if != 1 or 2 go To END ==> byte:2 /
MC:2.17
      CJNE R5,#2,ENDD2 ;==> byte:3 / MC:2.17
      RET ;==> byte:1 / MC:2.17

;------------------------------------ Unscrew START ------------------------------
------------;
;---------------------------- the same idea of Remove brake  ----------------------
------------;

Unscrew:
      MOV R5,55H ;==> byte:2 / MC:2.17
      CJNE R5,#1,MEOW4 ;==> byte:3 / MC:2.17
      MOV R7,51H ;==> byte:2 / MC:2.17
      CJNE R7,#1,IndividualUnscrew ;==> byte:3 / MC:2.17
      SJMP MoreUncrew ;==> byte:2 / MC:2.17
IndividualUnscrew:
      CLR C ;==> byte:1 / MC:1.085
      MOV A,58H ;==> byte:2 / MC:1.085
      ADD A,5DH ;the price of unscrew ==> byte:2 / MC:1.085
      MOV 58H,A ;==> byte:2 / MC:1.085
      RET ;==> byte:1 / MC:2.17
MoreUncrew:
      CLR C ;==> byte:1 / MC:1.085
      MOV A,52H ;==> byte:2 / MC:1.085
      MOV B,5DH ;==>byte:1 / MC:1.085
      MUL AB ;==> byte:1 / MC:4.34
      MOV R5,A ;==> byte:1 / MC:1.085
      MOV R6,B ;==>byte:1 / MC:1.085
      MOV A,58H ;==> byte:2 / MC:1.085
      ADD A,R5 ;==> byte:1 / MC:1.085
      JNC MEOW2 ;==> byte:2 / MC:2.17
      INC 48H ;==> byte:2 / MC:1.085
MEOW2:
      MOV 58H,A ;==> byte:2 / MC:1.085
      MOV A,48H ;==> byte:2 / MC:1.085
      ADD A,R6 ;==> byte:1 / MC:1.085
      MOV 48H,A ;==> byte:2 / MC:1.085
      RET
MEOW4:
      MOV R5,55H ;==> byte:2 / MC:2.17
      CJNE R5,#2,ENDD ;==> byte:3 / MC:2.17
```

```
      RET ;==> byte:1 / MC:2.17


;----------------------------------- Install START ------------------------------
------------;
;------------------------------ the same idea of Unscrew  ------------------------
------------;

Install:
      MOV R5,56H ;==> byte:2 / MC:2.17
      CJNE R5,#1,POV ;==> byte:3 / MC:2.17
      MOV R7,51H ;==> byte:2 / MC:2.17
      CJNE R7,#1,IndividualInstall ;==> byte:3 / MC:2.17
      SJMP MoreInstall ;==> byte:2 / MC:2.17
IndividualInstall:
      CLR C ;==> byte:1 / MC:1.085
      MOV A,58H ;==> byte:2 / MC:1.085
      ADD A,5EH ;==> byte:2 / MC:1.085
      MOV 58H,A ;==> byte:2 / MC:1.085
      RET ;==> byte:1 / MC:2.17
MoreInstall:
      CLR C ;==> byte:1 / MC:1.085
      MOV A,52H ;==> byte:2 / MC:1.085
      MOV B,5EH ;==> byte:2 / MC:2.17
      MUL AB ;==> byte:1 / MC:4.34
      MOV R5,A ;==> byte:1 / MC:1.085
      MOV R6,B ;==> byte:1 / MC:1.085
      MOV A,58H ;==> byte:2 / MC:1.085
      ADD A,R5 ;==> byte:1 / MC:1.085
      JNC POV2 ;==> byte:2 / MC:2.17
      INC 48H ;==> byte:2 / MC:1.085
POV2:
      MOV 58H,A ;==> byte:2 / MC:1.085
      MOV A,48H ;==> byte:2 / MC:1.085
      ADD A,R6 ;==> byte:1 / MC:1.085
      MOV 48H,A ;==> byte:2 / MC:1.085
      RET
POV:
      MOV R5,56H ;==> byte:2 / MC:2.17
      CJNE R5,#2,ENDD ;==> byte:3 / MC:2.17
      RET ;==> byte:1 / MC:2.17

;----------------------------------- Master Cylinder START ----------------------
--------------------;
;----------------------------------- the same idea of install  ------------------
--------------------;

MasterCylinder:
      MOV R5,57H ;==> byte:2 / MC:2.17
      CJNE R5,#1,POV3 ;==> byte:3 / MC:2.17
      MOV R7,51H ;==> byte:1 / MC:2.17
      CJNE R7,#1,IndividualMasterCylinder ;==> byte:3 / MC:2.17
      SJMP MoreMasterCylinder ;==> byte:2 / MC:2.17
IndividualMasterCylinder:
```

```
      CLR C ;==> byte:1 / MC:1.085
      MOV A,58H ;==> byte:2 / MC:1.085
      ADD A,5FH ;==> byte:2 / MC:1.085
      MOV 58H,A ;==> byte:2 / MC:1.085
      RET ;==> byte:1 / MC:2.17
MoreMasterCylinder:
      CLR C ;==> byte:1 / MC:1.085
      MOV A,52H ;==> byte:2 / MC:1.085
      MOV B,5FH ;==> byte:2 / MC:1.085
      MUL AB ;==> byte:1 / MC:4.34
      MOV R5,A ;==> byte:1 / MC:1.085
      MOV R6,B ;==> byte:1 / MC:1.085
      MOV A,58H ;==> byte:2 / MC:1.085
      ADD A,R5 ;==> byte:1 / MC:1.085
      JNC POV4 ;==> byte:2 / MC:2.17
      INC 48H ;==> byte:2 / MC:1.085
POV4:
      MOV 58H,A ;==> byte:2 / MC:1.085
      MOV A,48H ;==> byte:2 / MC:1.085
      ADD A,R6 ;==> byte:1 / MC:1.085
      MOV 48H,A ;==> byte:2 / MC:1.085
      RET
POV3:
      MOV R5,57H ;==> byte:2 / MC:2.17
      CJNE R5,#2,ENDD ;==> byte:3 / MC:2.17
      RET   ;==> byte:1 / MC:2.17

ENDD:
      SJMP $ ;==> byte:2 / MC:2.17
;------------------Category2: END-----------------------;


;the idea here is there is two loops the first one is the numbers between 100 to
299
;so here iam checking the number to know which loop iam going to use
CeckToConvert:
      MOV A,R3 ;==> byte:1 / MC:1.085
      SUBB A,#100 ;==> byte:2 / MC:1.085
      JNB Acc.7,ConvertToDecimalLower ;if the number is greater than 100 so go to
first loop ==> byte:3 / MC:2.17
      MOV 48H,#0;==> byte:3 / MC:2.17
      SJMP ConvertToDecimalLower2 ;if smaller go to the second one==> byte:2 /
MC:2.17

;------------------First loop to convert to decimal----------------------;

ConvertToDecimalLower:
      MOV A,R3 ;==> byte:1 / MC:1.085
      SUBB A,#200 ;==> byte:2 / MC:1.085
      JNB Acc.7,greater;if the number is greater than 200 so make the value of 48H
(hight bits) = 2 ==> byte:3 / MC:2.17
      SJMP SecondCheck ;to check if the number between 100 and 200 make the value
in 48H = 1 ==> byte:2 / MC:2.17
greater:
```

```
      Mov 48H,#2 ;==> byte:3 / MC:2.17
      SJMP BeRamadan ;==> byte:2 / MC:2.17
SecondCheck:
      MOV A,R3 ;==> byte:1 / MC:1.085
      SUBB A,#100 ;==> byte:2 / MC:1.085
      JNB ACC.7,greater2 ;==> byte:3 / MC:2.17
      SJMP BeRamadan ;==> byte:2 / MC:2.17
greater2:
      MOV 48H,#1 ;==> byte:3 / MC:2.17
;the idea here as we took in digital when i want to convert a binary num to decimal
if the first value in right equals 1
;i add 1 if the second value in the binary num from right is 1 i add 2
;for example if then num in hex is 25 so in binary is 0010 0101
;in this case i will add 1+4+32
BeRamadan:
      MOV R5,#8 ;because i have 8 bits ==> byte:2 / MC:1.085
      MOV R6,#1 ;the initial value i want to add is 1 ==> byte:2 / MC:1.085
      MOV R4,#0 ;the result of converting will be here ==> byte:2 / MC:1.085
      MOV R7,A ;==> byte:1 / MC:1.085
Hi2:
      MOV A,R7 ;==> byte:1 / MC:8.68
      CLR C ;==> byte:1 / MC:8.68
      RRC A ;==> byte:1 / MC:8.68
      MOV R7,A ;==> byte:1 / MC:8.68
      JNC Hi3 ;==> byte:2 / MC:17.36
      MOV A,R4 ;==> byte:1 / MC:8.68
      ADD A,R6 ;==> byte:1 / MC:8.68
      DA A ;==> byte:1 / MC:8.68
      CLR C ;==> byte:1 / MC:8.68
      MOV R4,A ;==> byte:1 / MC:8.68
Hi3:
      MOV A,R6 ;==> byte:1 / MC:8.68
      ADD A,R6 ;==> byte:1 / MC:8.68
      DA A ;==> byte:1 / MC:8.68
      MOV R6,A ;==> byte:1 / MC:8.68
      DJNZ R5,Hi2 ;==> byte:2 / MC:17.36
      MOV A,R4 ;==> byte:1 / MC:1.085
      ADD A,#1 ;==> byte:2 / MC:1.085
      DA A ;==> byte:1 / MC:1.085
      MOV R3,A ;==> byte:1 / MC:1.085
      RET ;==> byte:1 / MC:2.17

;------------------second loop to convert to decimal----------------------;
;------------------this loop for numbers between 0 and 100----------------------;
;------------------the same idea of the first loop----------------------;
ConvertToDecimalLower2:
      MOV A,R3 ;==> byte:1 / MC:1.085
      MOV R5,#8 ;==> byte:2 / MC:1.085
      MOV R6,#1 ;==> byte:2 / MC:1.085
      MOV R4,#0 ;==> byte:2 / MC:1.085
      MOV R7,A ;==> byte:1 / MC:1.085
Hello2:
      MOV A,R7 ;==> byte:1 / MC:8.68
      CLR C ;==> byte:1 / MC:8.68
```

```
      RRC A ;==> byte:1 / MC:8.68
      MOV R7,A ;==> byte:1 / MC:8.68
      JNC Hello3 ;==> byte:2 / MC:17.36
      MOV A,R4 ;==> byte:1 / MC:8.68
      ADD A,R6 ;==> byte:1 / MC:8.68
      DA A ;==> byte:1 / MC:8.68
      CLR C ;==> byte:1 / MC:8.68
      MOV R4,A ;==> byte:1 / MC:8.68
Hello3:
      MOV A,R6 ;==> byte:1 / MC:8.68
      ADD A,R6 ;==> byte:1 / MC:8.68
      DA A ;==> byte:1 / MC:8.68
      MOV R6,A ;==> byte:1 / MC:8.68
      DJNZ R5,Hello2 ;==> byte:2 / MC:17.36
      MOV A,R4 ;==> byte:1 / MC:1.085
      MOV R3,A ;==> byte:1 / MC:1.085
      RET ;==> byte:1 / MC:2.17
END
```

4- Conclusion:

At the end we can conclude that 8051 assembly language can be used widely in different aspects of our daily life and we can program a very powerful programs using this low-level language.

5- References:

1- Lecture Note