**Table of contest:**

Objective …………………………………………………………….….page 2

Theory …………………………………………………………………...page 2

Calculated values…………………………………………………….page 3-6

Program…………………………………………………………….……Page 3-6

Mnemonics…………………………………………………………....Page 3-6

Flow chart………………………………………….....……………….Pare 3-6

OP Codes ………………………………………………………....……Page 3-6

Questions and Answers………………................………….Page 3-6

Conclusion……………………………………………………………...Page 6e

**Objective**

The objective of this lab report is to write and test an assembly program that displays the moving message “**PEPSI**” on the address field of your sdk-8085 with a moving speed of one character per second.

**Theory**

The theory is using appendix F to write a program to display pass fail and blank using the call command CD02B7.

02B7 is a call command that uses flags to determine whether to display data in address or data field, or both. Display Flag is appointed by register A and the Dot Flag is appointed by register B.

When loading register A with 00 means you are displaying data in the address field.

When loading register A with 01 it means that you are displaying data in data field.

When loading register B with 00 it means there is no dots in the display.

When loading register B with 01 it means that there will be a dot in the right edge of the data field.

Also HL addresses the output to the display.

The CD 02B7 is a 3 byte instruction and it take 18 cycle to execute. All data is destroyed when using the CD call if there is any important data that need to be saved be sure to push them and later you call always pop that data for use.

Writing a program using the JMP commands to keep the program in a loop to display PASS FAIL BLANK in an endless loop. First we have to load the data that represents the characters in the memory location. This way before calling 02B7 we load HL with the beginning of the data that represents the characters.

All characters are represented in the table below.

|  |  |
| --- | --- |
| CHAR | HEX. |
| 0 | 00 |
| 1 | 01 |
| 2 | 02 |
| 3 | 03 |
| 4 | 04 |
| 5 | 05 |
| 6 | 06 |
| 7 | 07 |
| 8 | 08 |
| 9 | 09 |
| A | 0A |
| B | 0B |
| C | 0C |
| D | 0D |
| E | 0E |
| F | 0F |
| H | 10 |
| L | 11 |
| P | 12 |
| I | 13 |
| r | 14 |
| BK | 15 |

**Process**

**Using subroutines output 02B7 and delay 05F1.**

**Draw a clear flow chart of the solution, and convert the flowchart in mnemonics assembler code and ram locations.**

|  |  |  |
| --- | --- | --- |
| Memory location | Hex Character | Data |
| 2000 | BK | 15 |
| 2001 | BK | 15 |
| 2002 | BK | 15 |
| 2003 | P | 12 |
| 2004 | E | 0E |
| 2005 | P | 12 |
| 2006 | S | 05 |
| 2007 | I | 13 |
| 2008 | BK | 15 |
| 2009 | BK | 15 |
| 200a | BK | 15 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MEMORY LOCATION | OP CODE | FLOW CHART | MNEMONICS | BYTES | CYCLES |
| 2010 | 31 | [SP]<-|20C2| | LXI SP, 20C2 | 3 | 10 |
| 2011 | C2 |  |  |  |  |
| 2012 | 20 |  |  |  |  |
| 2013 | 21 | [HL]<-|2000| | LXI H, 2000 | 3 | 10 |
| 2014 | 00 |  |  |  |  |
| 2015 | 20 |  |  |  |  |
| 2016 | E5 | PUSH H | PUSH H | 1 | 12 |
| 2017 | 3E | [A]<-|00| | MVI A , 00 | 2 | 7 |
| 2018 | 00 |  |  |  |  |
| 2019 | 06 | [B]<-|00| | MVI B, 00 | 2 | 7 |
| 201A | 00 |  |  |  |  |
| 201B | CD | CALL 02B7 | CD 02B7 | 3 | 18 |
| 201C | B7 |  |  |  |  |
| 201D | 02 |  |  |  |  |
| 201E | 3E | [A]<-|02| | MVI A, 02 | 2 | 7 |
| 201F | 02 |  |  |  |  |
| 2020 | F5 | PUSH PSW | PUSH PSW | 1 | 12 |
| 2021 | 11 | [DE]<-|F696| | LXI D, F696 | 3 | 10 |
| 2022 | 96 |  |  |  |  |
| 2023 | F6 |  |  |  |  |
| 2024 | 1B | [DE]<-[DE]-|1| | DCX D | 1 | 6 |
| 2025 | 7A | [A]<-[D] | MOV A,D | 1 | 4 |
| 2026 | B3 | [A]<-[A]V[E] | ORA E | 1 | 4 |
| 2027 | C2 | JNZ 2024 | JNZ 2024 | 3 | 7/10 |
| 2028 | 24 |  |  |  |  |
| 2029 | 20 |  |  |  |  |
| 202A | F1 | POP PSW | POP PSW | 1 | 10 |
| 202B | 3D | [A]<-[A]-|1| | DCR A | 1 | 4 |
| 202C | C2 | JNZ 2020 | JNZ 2020 | 3 | 7/10 |
| 202D | 20 |  |  |  |  |
| 202E | 20 |  |  |  |  |
| 202F | E1 | POP H | POP H | 1 | 10 |
| 2030 | 23 | [HL]<-[HL]+|1| | INX H | 1 | 6 |
| 2031 | 3E | [A]<-|0B| | MVI A, 0B | 2 | 7 |
| 2032 | 0B |  |  |  |  |
| 2033 | BD | [A]<->[L] | CMP L | 1 | 4 |
| 2034 | C2 | JNZ 2016 | JNZ 2016 | 3 |  |
| 2035 | 16 |  |  |  |  |
| 2036 | 20 |  |  |  |  |
| 2037 | CF | END | RST1 | 1 | 12 |

TOTAL 40 BYTES

**TIME DELAY**

1 = 330 × { 48 × X10 + 106 + 130}

=> × = 48 × X10 + 236

=>= 48 × X10 = × -236

=> X10 = × -236

=> X10 = 63,126

Hex= F695

E) MODEFY THE FLOW CHART AND PROGRAM SO THAT THE MOVING MESSAGE “PEPSI” OS DISPLAYED 3 TIMES

THE FIRST TIME WITH A MOVING SPEED OF 1 SECOND

THE SECOND TIME WITH A MOVING SPEED OF 2 SECOND

THE 3RD TIME WITH A MOVINS SPEED OF 3 SECONDS.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MEMORY LOCATION | OP CODES | FLOW CHART | MNEMONICS | BYTES | CYCLES |
| 2010 | 31 | [SP]<-20C2 | LXI SP, 20C2 | 3 | 10 |
| 2011 | C2 |  |  |  |  |
| 2012 | 20 |  |  |  |  |
| 2013 | 01 | [BC]<-|0302| | LXI B, 0302 | 3 | 10 |
| 2014 | 02 |  |  |  |  |
| 2015 | 03 |  |  |  |  |
| 2016 | C5 | PUSH B | PUSH B | 1 | 12 |
| 2017 | 21 | [HL]<-|2000| | LXI H, 2000 | 3 | 10 |
| 2018 | 00 |  |  |  |  |
| 2019 | 20 |  |  |  |  |
| 201A | E5 | PUSH H | PUSH H | 1 | 12 |
| 201B | 3E | [A]<-|00| | MVI A, 00 | 2 | 7 |
| 201C | 00 |  |  |  |  |
| 201D | 06 | [B]<-|00| | MVI B,00 | 2 | 7 |
| 201E | 00 |  |  |  |  |
| 201F | CD | CALL 02B7 | CD 02B7 | 3 | 18 |
| 2020 | B7 |  |  |  |  |
| 2021 | 02 |  |  |  |  |
| 2022 | C1 | POP B | POP B | 1 | 10 |
| 2023 | 79 | [A]<-[C] | MOV A,C | 1 | 4 |
| 2024 | C5 | PUSH B | PUSH B | 1 | 12 |
| 2025 | F5 | PUSH PSW | PUSH PSW | 1 | 12 |
| 2026 | 11 | [DE]<-| | | LXI D, XXXX | 3 | 10 |
| 2027 |  |  |  |  |  |
| 2028 |  |  |  |  |  |
| 2029 | 1B | [DE]<-[DE]-|1| | DCX D | 1 | 6 |
| 202A | 7A | [A]<-[D] | MOV A, D | 1 | 4 |
| 202B | B3 | [A]<-[A]V[E] | ORA E | 1 | 4 |
| 202C | C2 | JNZ 2029 | JNZ 2029 | 3 | 7/10 |
| 202D | 29 |  |  |  |  |
| 202E | 20 |  |  |  |  |
| 202F | F1 | POP PSW | POP PSW | 1 | 10 |
| 2030 | 3D | [A]<-[A]-|1| | DCR A | 1 | 4 |
| 2031 | CE | JNZ 2025 | JNZ 2025 | 3 | 7/10 |
| 2032 | 25 |  |  |  |  |
| 2033 | 20 |  |  |  |  |
| 2034 | E1 | POP H | POP H | 1 | 10 |
| 2035 | 23 | [HL]<-[HL]+|1| | INX H | 1 | 6 |
| 2036 | 3E | [A]<-|0B| | MVI A, 0B | 2 | 7 |
| 2037 | 0B |  |  |  |  |
| 2038 | BD | [A]<->[L] | CMP L | 1 | 4 |
| 2039 | C2 | JNZ 201A | JNZ 201A | 3 | 7/10 |
| 203A | 1A |  |  |  |  |
| 203B | 20 |  |  |  |  |
| 203C | C1 | POP B | POP B | 1 | 10 |
| 203D | 0C | [C]<-[C]+|1| | INR C | 1 | 4 |
| 203E | 0C | [C]<-[C]+|1| | INR C | 1 | 4 |
| 203F | 78 | [A]<-[B] | MOV A, B | 1 | 4 |
| 2040 | 3D | [A]<-[A]-|1| | DCR A | 1 | 4 |
| 2041 | C2 | JNZ 2016 | JNZ 2016 | 3 | 7/10 |
| 2042 | 16 |  |  |  |  |
| 2043 | 20 |  |  |  |  |
| 2034 | CF | END | RST1 | 1 | 12 |

TOTAL 53 BYTES

**TIME DELAY**

1 = 330 × { 48 × X10 + 106 + 158}

=> × = 48 × X10 + 264

=>= 48 × X10 = × -264

=> X10 = × -264

=> X10 = 63,125

**HEX= F695**

**CONCLUSION**

In this lab report we have learned how to develop a program that displays a moving massage such as “PEPSI” in the address field with various display speeds. We also have learned to calculate speed time by using the time delay theory.