



UNIVERSIDAD SIMÓN BOLÍVAR  
VICERRECTORADO ACADÉMICO  
DECANATO DE ESTUDIOS TECNOLÓGICOS  
COORDINACIÓN DE TECNOLOGÍA ELÉCTRICA Y ELECTRONICA  
LABORATORIO DE MICROPROCESADORES

### **Práctica #3**

Diseño de programa para el microprocesador 8085 utilizando el simulador Granada  
“Programa donde se pueda ingresar a través del teclado el número de su cedula y luego es  
mostrado en un display”  
*(Tecnología Electrónica)*

**Autor:** Adrian Mayora

**Carnet:** 16-00131

**C.I:** 26.180.109

**Profesor:** Alberto Armengol

**Camurí Grande, julio de 2022**

# ÍNDICE

Índice.....	
Índice de Figuras.....	2
Introducción .....	3
Instrucciones .....	4
Diseño del programa en el simulador granada.....	5
Programa de inicio.....	5
Subrutina de interrupción .....	6
Subrutinas cedula y cedila1 .....	7
Subrutina mostrar .....	8
Subrutina fin .....	10
Subrutina retardo .....	11
Video del programa simulado en granada .....	11
Diagrama del Circuito diseñado.....	12
Conexiones del diagrama diseñado.....	13
Consideraciones acerca de los dispositivos externos al 8085 y cambios al programa.....	15
Programa rediseñado tomando en cuenta factores externos al 8085.....	16
Conclusion .....	22
Referencias Consultadas .....	23

## ÍNDICE DE FIGURAS

Figura 1. Montaje del circuito diseñado.....	12
Figura 2. Conexiones del circuito .....	13
Figura 3. Conexiones de los puertos .....	14

## INTRODUCCIÓN

El presente informe se enfocará en la creación de un programa en el simulador granada capaz de leer los números de una cedula introducidos al microprocesador 8085 a través de un teclado numérico, estos números serán guardados en una región de memoria llamada lista, y que luego al presionar la tecla \* del teclado podamos ir mostrando en un display cada uno de los números presionados anteriormente, al finalizar de mostrar el último número el display se encenderá u apagará 5 veces para indicar que el programa finalizó, por último se explicará el circuito necesario para su implementación.

## INSTRUCCIONES

**a)** Utilizando solamente el simulador desarrollado en la Universidad de Granada (Simulador Granada) elabore un programa donde pueda ingresar a través del teclado el número de su cedula (lo puede finalizar con cualquier otro ASCII que no pertenezca al código de los 10 dígitos). Naturalmente el simulador ingresará esos dígitos en código ASCII en alguna porción de memoria que llamaremos LISTA.

Una vez introducidos los números, entre los que puede haber repetidos, hay que ordenarlos en forma ascendente. Una vez que la lista ha sido ordenada, se debe poder mostrar en un display o visualizador siete segmentos cada uno de los dígitos ordenados cada vez que se presione un pulsador (para el simulador ese pulsador será la tecla \*) al llegar al último de los números de la lista lo mostrará hasta el próximo pulso, entonces encenderá y apagará el visualizador 5 veces a una frecuencia claramente visible y finalizará el programa.

**b)** Usando un simulador como por ejemplo el PROTEUS, haga el circuito real para implementar el problema anterior. Para ello habrá que asumir algunas cosas y hacer algunas modificaciones prácticas:

1 - PROTEUS no tiene una MPU equivalente al  $\mu P$  8085 por lo tanto use cualquier chip con huella 40 pin DIP (naturalmente no es operativo y no podrá simularse) para efectos de las conexiones se respetará la función de cada una de las patas del  $\mu P$ . (identifique la funcionalidad cada una de las patas que necesite usar con su nombre)

2- Para substituir el teclado del simulador se usará uno tipo matriz numérico de 12 teclas (habría que hacer una pequeña rutina en el 8085 para transformar la salida de ese teclado a su correspondiente código ASCII para hacerlo compatible con el resto del programa ya hecho en la sección anterior.)

3- El pulsador para mostrar los dígitos en el display seguirá siendo la tecla \*.

4- La salida seguiría siendo un visualizador 7-segmentos pero como los microprocesadores carecen de puertos como tales tendrá que auxiliarse con otros circuitos. (Puede usar 8355/8755, un 8155 o un registro tipo latch como el 74LS374 o similar y probablemente necesitará un decodificador BCD→7-segmentos. También que para poder cargar el programa requiere de alguna memoria. No olvide explicar lo que hace, por qué y cómo lo hace.

Recuerde que el uso de Proteus en este caso solo es para hacer el diagrama el cual NO SE PODRÁ SIMULAR, por ello podría usar otra herramienta informática para trazar el diagrama. Recuerde incluir en su informe un "PrintScreen" de su diagrama. De la misma manera en el informe deben incluir un video o un vínculo del programa funcionando en el simulador (no debería exceder de unos 30 segundos, recuerde que hay muchas aplicaciones para grabar la pantalla de su computador, yo uso Vokoscreen, si su equipo no aguantara alguna de estas aplicaciones, puede grabar con un celular y con la aplicación de WhatsApp o Telegram bajar el video a su computadora y/o directamente subirlo a su GoogleDrive)

## DISEÑO DEL PROGRAMA EN EL SIMULADOR GRANADA

### Programa de inicio

Se encarga básicamente de inicializar el programa, primero asegurándonos de que el display este apagado, luego configurando los registros que se utilizaran como contadores y apuntando la dirección donde se comenzaran a guardar los números de la cedula, para luego entrar en un bucle, donde solo se saldrá si se pulsa una tecla.

```
; -----  
;PROGRAMA DE INICIO  
; -----  
    .org 1000h  
    MVI A,00H           ;me aseguro que el display este apagado  
    OUT 07H  
    MVI A,08H           ;contador 1 (indica cuantos datos he guardado)  
    MOV C,A  
    MVI A,08H           ;contador 2 (indica cuantos datos he mostrado en el  
                        ;display)  
    MOV B,A  
    LXI H,2000H         ;apunto la dirección donde guardare los números de la  
                        ;cedula  
  
BUCLE: EI               ;Habilito interrupciones  
        JMP BUCLE       ;Bucle constante, solo sale con interrupción
```

## Subrutina de interrupción

Se ejecuta cuando se presiona una tecla y se encarga de guardar en una lista los dígitos de un numero de cedula y luego mostrarlos en un display.

```
; -----  
; RUTINA DE INTERRUPCION:  
; -----  
  
.org 0034h  
    DI                ;deshabilito interrupciones  
    DCR C             ;indico que guarde el primer digito  
    MOV A,C  
  
    CPI 00H           ;si es el digito que guarde es el 8  
    JZ  CEDULA        ;pasar a mostrar los dígitos introducidos  
  
    CPI 55H           ;si ya introduje lo 8 dígitos  
    JZ  CEDULA2       ;ver si lo tecla que se presiono es *  
  
    IN 00H            ;sino leer el digito  
    MOV M,A           ;guardo digito  
    INX H             ;aviso que ya lei el digito  
    JMP BUCLE         ;espero a que se intrudusca el siguiente
```

## Subrutinas cedula y cedila1

Se encargan de avisar al programa de interrupción que ya se introdujeron los 8 dígitos de la cedula y ahora debe mostrarlos en el display.

```
-----  
;SUBROUTINAS CEDULA Y CEDILA1:  
-----  
  
CEDULA:      IN 00H          ;leo el ultimo digito  
              MOV M,A        ;guardo el ultimo digito  
              LXI H,2000H     ;apunto al inicio de la lista  
              MVI A,56H       ;indico que ya se guardaron todos los dígitos  
  
              MOV C,A  
              JMP BUCLE       ;espero a que se presione *  
  
CEDULA2:     MVI A,56H       ;indico que ya se guardaron todos los dígitos  
              MOV C,A  
              IN 00H          ;veo que tecla se pulso  
              CPI '*'        ;veo que tecla se pulso  
              JZ  MOSTRAR     ;si se pulso el * mostrar el digito en el  
display      JMP BUCLE       ;sino esperar a que se pulse *
```



## Subrutina mostrar

Se encarga de transformar el código ASCII de los dígitos de la lista en su equivalente 7 segmentos y mostrarlos en el display.

```
-----  
;subrutina MOSTRAR:  
-----  
  
MOSTRAR:      MOV A,M           ;veo el digito que indique la pila  
               CPI '0'          ;veo cual de los 10 digitos posible  
               JZ CERO          ;fue el pulsado  
               CPI '1'  
               JZ UNO  
               CPI '2'  
               JZ DOS  
               CPI '3'  
               JZ TRES  
               CPI '4'  
               JZ CUATRO  
               CPI '5'  
               JZ CINCO  
               CPI '6'  
               JZ SEIS  
               CPI '7'  
               JZ SIETE  
               CPI '8'  
               JZ OCHO  
               CPI '9'  
               JZ NUEVE  
               JMP BUCLE  
  
CERO:          MVI A,77H        ;si fue 0  
               OUT 07H          ;mostrar el código 7seg en el display  
               INX H            ;apunto al siguiente digito  
               CALL RETARDO     ;espero un momento  
               DCR B            ;indico que ya mostré un digito  
               CPI 00H          ;veo si ya mostré todos los dígitos  
               JZ FIN           ;si es así ir a finalizar  
               JMP BUCLE        ;sino esperar al siguiente pulso de *  
  
UNO:           MVI A,44H        ;se repite lo mismo para los demas números  
               OUT 07H  
               INX H  
               CALL RETARDO  
               DCR B  
               MOV A,B  
               CPI 00H  
               JZ FIN  
               JMP BUCLE  
  
DOS:           MVI A,3EH  
               OUT 07H  
               INX H  
               CALL RETARDO  
               DCR B  
               MOV A,B  
               CPI 00H  
               JZ FIN  
               JMP BUCLE
```

```

TRES:      MVI A,6EH
           OUT 07H
           INX H
           CALL RETARDO
           DCR B
           MOV A,B
           CPI 00H
           JZ  FIN
           JMP BUCLE

CUATRO:    MVI A,4DH
           OUT 07H
           INX H
           CALL RETARDO
           DCR B
           MOV A,B
           CPI 00H
           JZ  FIN
           JMP BUCLE

CINCO:     MVI A,6BH
           OUT 07H
           INX H
           CALL RETARDO
           DCR B
           MOV A,B
           CPI 00H
           JZ  FIN
           JMP BUCLE

SEIS:      MVI A,7BH
           OUT 07H
           INX H
           CALL RETARDO
           DCR B
           MOV A,B
           CPI 00H
           JZ  FIN
           JMP BUCLE

SIETE:     MVI A,46H
           OUT 07H
           INX H
           CALL RETARDO
           DCR B
           MOV A,B
           CPI 00H
           JZ  FIN
           JMP BUCLE

OCHO:      MVI A,7FH
           OUT 07H
           INX H
           CALL RETARDO
           DCR B
           MOV A,B
           CPI 00H
           JZ  FIN
           JMP BUCLE

NUEVE:     MVI A,6FH
           OUT 07H
           INX H
           CALL RETARDO
           DCR B
           MOV A,B
           CPI 00H
           JZ  FIN
           JMP BUCLE

```

## Subrutina fin

Esta subrutina se encarga de apagar y encender el display 5 veces para indicar el fin. del programa.

```
;-----  
;subrutina FIN:  
;-----  
  
FIN:  
  
    MVI A,08H  
    OUT 07H  
    CALL RETARDO  
    MVI A,00H  
    OUT 07H  
    CALL RETARDO  
  
    MVI A,08H  
    OUT 07H  
    CALL RETARDO  
    MVI A,00H  
    OUT 07H  
    CALL RETARDO  
  
    MVI A,08H  
    OUT 07H  
    CALL RETARDO  
    MVI A,00H  
    OUT 07H  
    CALL RETARDO  
  
    MVI A,08H  
    OUT 07H  
    CALL RETARDO  
    MVI A,00H  
    OUT 07H  
    CALL RETARDO  
  
    MVI A,08H  
    OUT 07H  
    CALL RETARDO  
    MVI A,00H  
    OUT 07H  
    CALL RETARDO  
  
    MVI A,08H  
    OUT 07H  
    CALL RETARDO  
    MVI A,00H  
    OUT 07H  
    CALL RETARDO  
  
    HLT
```

## Subrutina retardo

Se encarga de realizar un retardo que se pueda visualizar en el display.

```
;-----  
;subrutina RETARDO  
;-----  
  
RETARDO:      MVI E,FFH      ;carga en el registro E FFh  
NEXT:         MVI D,20H      ;carga en el registro D 20h  
BACK:         DCR D          ;decremento el registro D  
              JNZ BACK       ;sino es 0 el registro de seguir  
              ;decrementando  
              DCR E          ;si es 0 decrementar el registro E  
              JNZ NEXT       ;sino es 0 el registro E volver a decrementar  
              ;el registro D hasta que sea 0  
              RET            ;si es 0 el registro E retornamos
```

## Video del programa simulado en granada:

[https://drive.google.com/file/d/1g\\_87pHcdl1Lza3LHfqmQmHLzU5hzYqMc/view?usp=drive\\_web](https://drive.google.com/file/d/1g_87pHcdl1Lza3LHfqmQmHLzU5hzYqMc/view?usp=drive_web)

## DIAGRAMA DEL CIRCUITO DISEÑADO

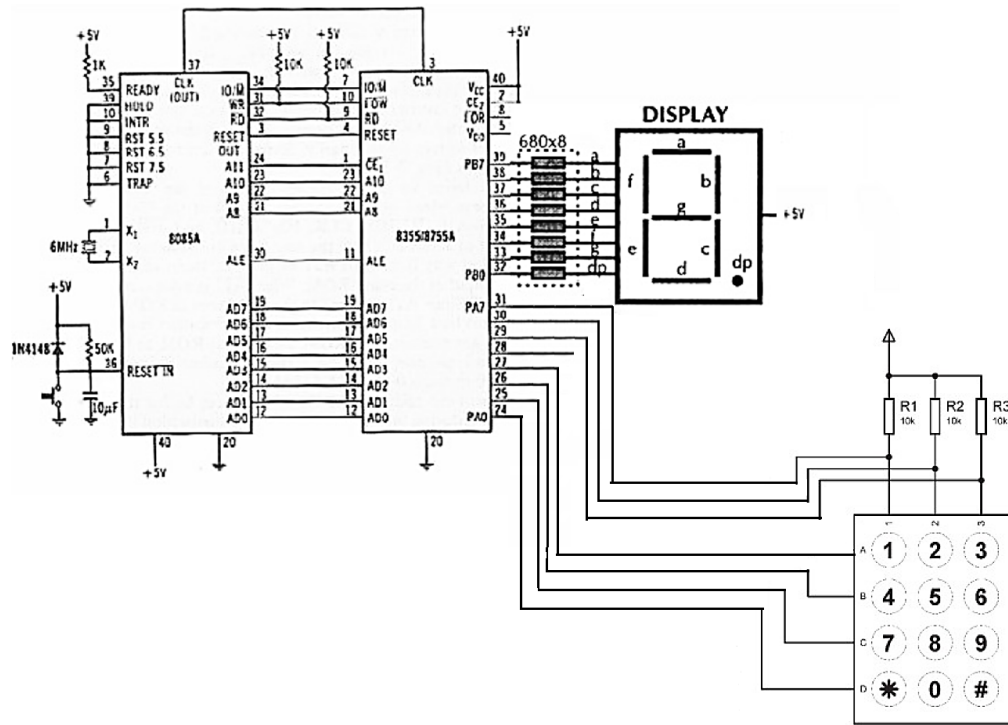


Figura 1. Montaje del circuito diseñado - Imagen de autoría propia rediseñada de un circuito dado por el profesor en cursos anteriores

## CONEXIONES DEL DIAGRAMA DISEÑADO

Una de las principales características que presenta el microprocesador 8085 es poseer un bus de datos (de 8 líneas) multiplexado con la parte baja del bus de direcciones. Este multiplexado quiere decir que en un instante determinado tendremos en las 8 líneas del bus la información perteneciente a una dirección y un instante más tarde tendremos en esas mismas líneas la información perteneciente a un dato. (Vazquez, C)

Corresponde por tanto a la circuitería externa, separar estas señales de modo apropiado, tal que no se produzcan interferencias por informaciones erróneas en el resto de los componentes del sistema. De esta forma se obtendrá una separación total entre el bus de datos (8 bits) y el bus de direcciones (16 bits). (Vazquez, C)

Como sabemos el microprocesador 8085 dispone de una línea de “control” o “sincronización” denominada ALE, por medio de la cual suministra un pulso de tensión de nivel lógico 1 cuando se presenta la parte baja de una dirección en el bus AD0,...,AD7 y cuando aparece un dato la señal ALE pasa al estado lógico 0. Deberemos por tanto emplear esta señal de control para separar estas informaciones utilizando un circuito integrado que actúe de cerrojo (latch) y realice la separación física. (Vazquez, C)

Por otra parte, **CE** se encuentra conectado a la línea 11 del bus de direcciones, por lo cual, sabiendo que el direccionamiento E/S es de 8 bits, los cuales se duplican en el bus de direcciones de 16 bits, tendremos que A11=A3 deberán ser igual 0 si se quiere configurar los puertos. Por último, IO/M, RD, WR se encuentran conectados de forma que el microprocesador pueda avisarle 8355 que tipo instrucción se está ejecutando.

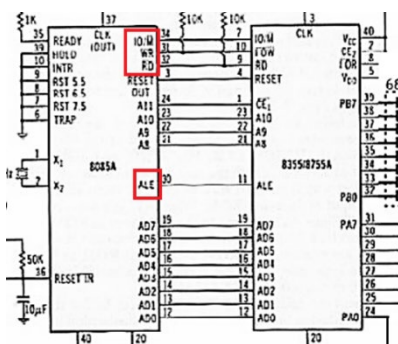


Figura 2. Conexiones del circuito - Imagen de autoría propia rediseñada de un circuito dado por el profesor en cursos anteriores

Por último, podemos ver en la figura de abajo que en el puerto B del 8355 se conectó el display ánodo común (cuadro rojo) y en el puerto A el teclado (cuadro verde).

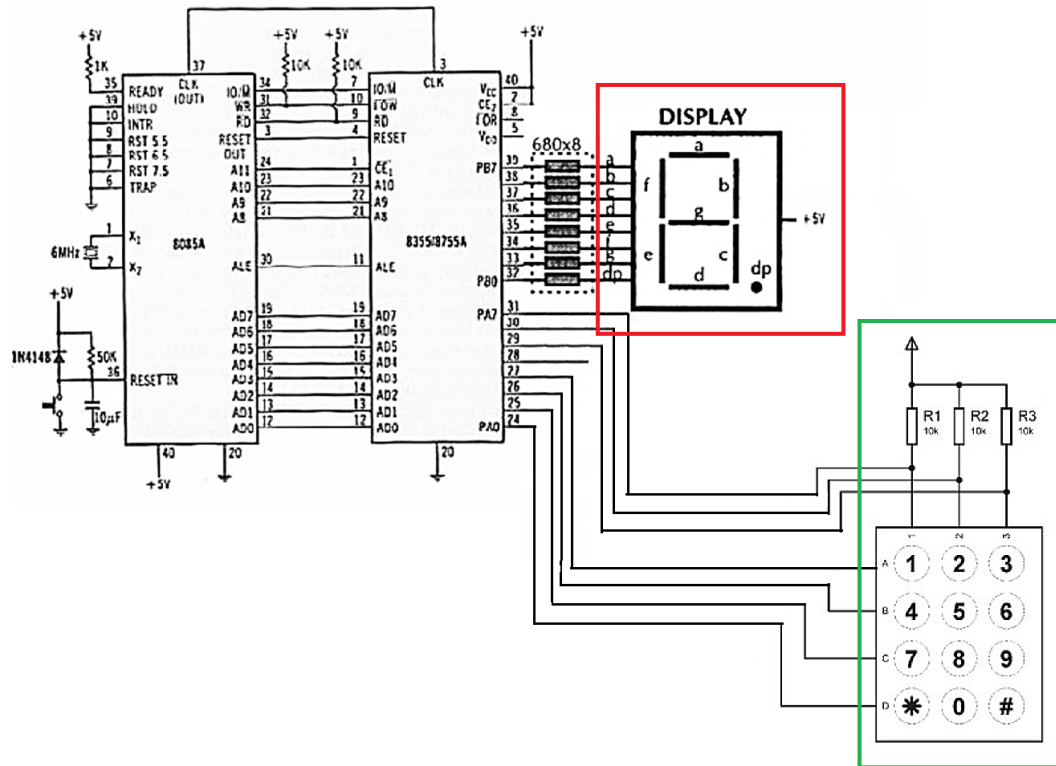


Figura 3. Conexiones de los puertos - Imagen de autoría propia rediseñada de un circuito dado por el profesor en cursos anteriores

## CONSIDERACIONES ACERCA DE LOS DISPOSITIVOS EXTERNOS AL 8085 Y CAMBIOS AL PROGRAMA

El programa hecho anteriormente fue para un caso específico utilizando el simulador granada el cual no toma en cuenta los dispositivos externos al 8085, en esta sección se tomarán en cuenta los dispositivos que se mostraron en el diagrama diseñado y los cambios que se deben realizar al programa.

Primero como vimos para sustituir el teclado del simulador se usará uno tipo matriz numérico de 12 teclas (habría que hacer una pequeña rutina en el 8085 para transformar la salida de ese teclado a su correspondiente código ASCII para hacerlo compatible con el resto del programa ya hecho en la sección anterior.)

Además, se deberán hacer un cambio en la configuración de los puertos ya que estamos utilizando el 8355, recordando que la configuración de puertos se hace con los bits A1 y A2:

	A <sub>1</sub>	A <sub>0</sub>
Puerto A	0	0
Puerto B	0	1
DDR A	1	0
DDR B	1	1

Donde cada línea de cada puerto se puede programar independientemente con un 0 para entrada o un 1 para salida. Las salidas tienen su latch interior, o sea la salida permanece hasta que se cambie

Por último, se realizarán cambios en el código 7 segmentos ya que el simulador utiliza un código diferente al de un display 7 segmentos real ánodo común,

Todos estos cambios se muestran en el programa rediseñado tomando en cuenta factores externos al 8085 en las paginas siguientes:



## PROGRAMA REDISEÑADO TOMANDO EN CUENTA FACTORES EXTERNOS AL 8085

```

; -----
;PROGRAMA DE INICIO: se encarga de inicializar el programa apagando el display,
;primero configurando los puertos, luego los registros que se utilizaran como
;contadores y apuntando la dirección donde se comenzara a guardar los números de
;la C.I, luego de esto entrará en un bucle hasta que se presione una tecla, que
;finalizara guardando en el registro E el código ASCII de la tecla presionada
; -----
                .org 0000h
                MVI A,0FH                ;mitad entrada mitad salida
                OUT 02H                  ;configuro el puerto A
                MVI A,FFH                ;todo salida
                OUT 03H                  ;configuro el puerto B

                MVI A,FFH                ;me aseguro que el display este apagado
                OUT 05H
                MVI A,08H                ;contador 1 (indica cuantos datos he
                                         ;guardado)

                MOV C,A
                MVI A,08H                ;contador 2 (indica cuantos datos he mostrado
                                         ;en el display)

                MOV B,A
                LXI H,2000H              ;apunto la dirección donde guardare los
                                         ;numeros de la C.I

BUCLE:
                MVI A,0FFh
                OUT 00H                  ;desselecciono todas la columnas y filas

                MVI A,0EFh
                OUT 00H                  ;coloca en bajo el pin A4 (escaneo culmna 1)
                IN 00H                   ;leo el puerto A
                CPI 0FEH                 ;si el pin A0=0 envia el numero 0 de lo
                                         ;contrario salta fila siguiente

                JZ UNO1
                CPI 0FDH                 ;si el pin A1=0 envia el numero 2 de lo
                                         ;contrario salta fila siguiente

                JZ DOS1
                CPI 0FBH                 ;si el pin C2=0 envia el numero 4 de lo
                                         ;contrario salta fila siguiente

                JZ TRES1

                MVI A,06Fh
                OUT 00H                  ;coloco en B5 el pin A5 (escaneo columna 2)
                IN 00H
                CPI 0FEH
                JZ CUATRO1
                CPI 0FDH
                JZ CINCO1
                CPI 0FBH
                JZ SEIS1

                MVI A,0BFh
                OUT 00H
                IN 00H
                CPI 0FEH
                JZ SIETE1
                CPI 0FDH
                JZ OCHO1
                CPI 0FBH
                JZ NUEVE1

```

```

MVI A,07Fh
OUT 00H
IN 00H
CPI 0FEH
JZ ASTERISCO
CPI 0FDH
JZ CERO1
CPI 0FBH
JZ NUMERAL
JMP BUCLE ;vuelvo a inicio

CERO1: MVI E,'0'
        JMP GUARDAR
UNO1:  MVI E,'1'
        JMP GUARDAR
DOS1:  MVI E,'2'
        JMP GUARDAR
TRES1: MVI E,'3'
        JMP GUARDAR
CUATRO1: MVI E,'4'
        JMP GUARDAR
CINCO1: MVI E,'5'
        JMP GUARDAR
SEIS1:  MVI E,'6'
        JMP GUARDAR
SIETE1: MVI E,'7'
        JMP GUARDAR
OCHO1:  MVI E,'8'
        JMP GUARDAR
NUEVE1: MVI E,'9'
        JMP GUARDAR
ASTERISCO: MVI E,'*'
        JMP GUARDAR
NUMERAL: MVI E,'#'
        JMP GUARDAR

; -----
;SUBROUTINA GUARDAR: se encarga de guardar en una lista los dígitos
;de un numero de cedula y luego mostrarlos en un display
; -----

GUARDAR:
        DI ;deshabilito interrupciones
        DCR C ;indico que guarde el primer digito
        MOV A,C

        CPI 00H ;si es el digito que guarde es el 8
        JZ CEDULA ;pasar a mostrar los dígitos introducidos

        CPI 55H ;si ya intrduje lo 8 dígitos
        JZ CEDULA2 ;ver si lo tacla que se presiono es *

        MOV A,E ;sino leer el digito
        MOV M,A ;guardo digito
        INX H ;aviso que ya lei el digito
        JMP BUCLE ;espero a que se introduzca el siguiente

```

```

;-----
;SUBROUTINAS CEDULA Y CEDILA1: se encargan de avisar al programa principal que ya
;se introdujeron los 8 digitos de la cedula y ahora debe mostrarlos en el display
;-----

CEDULA:      MOV A,E          ;leo el ultimo digito
              MOV M,A          ;guardo el ultimo digito
              LXI H,2000H      ;apunto al inicio de la lista
              MVI A,56H        ;indico que ya se guardaron todos los digitos

              MOV C,A          ;espero a que se presione *
              JMP BUCLE

CEDULA2:     MVI A,56H        ;indico que ya se guardaron todos los digitos
              MOV C,A          ;veo que tecla se pulso
              CPI '*'          ;si se pulso el * mostrar el digito en el
                              ;display
              JZ  MOSTRAR      ;sino esperar a que se pulse *
              JMP BUCLE

;-----
;subrutina MOSTRAR: se encarga de transformar el codigo ASCII de los digitos
;de la lista en su equivalente 7seg y mostrarlos en el display.
;-----

MOSTRAR:     MOV A,M          ;veo el digito que indique la pila
              CPI '0'          ;veo cual de los 10 digitos posible
              JZ  CERO          ;fue el pulsado
              CPI '1'
              JZ  UNO
              CPI '2'
              JZ  DOS
              CPI '3'
              JZ  TRES
              CPI '4'
              JZ  CUATRO
              CPI '5'
              JZ  CINCO
              CPI '6'
              JZ  SEIS
              CPI '7'
              JZ  SIETE
              CPI '8'
              JZ  OCHO
              CPI '9'
              JZ  NUEVE
              JMP BUCLE

CERO:        MVI A,C0H        ;si fue 0
              OUT 07H          ;mostrar el codigo 7seg en el display
              INX H            ;apunto al siguiente digito
              CALL RETARDO     ;espero un momento
              DCR B            ;indico que ya mostre un digito
              CPI 00H          ;veo si ya mostre todos los digitos
              JZ  FIN          ;si es asi ir a finalizar
              JMP BUCLE        ;sino esperar al siguiente pulso de *

```

UNO:	MVI A,FAH	;se repite lo mismo para los demas numeros
	OUT 07H	
	INX H	
	CALL RETARDO	
	DCR B	
	MOV A,B	
	CPI 00H	
	JZ FIN	
	JMP BUCLE	
DOS:	MVI A,0A4H	
	OUT 07H	
	INX H	
	CALL RETARDO	
	DCR B	
	MOV A,B	
	CPI 00H	
	JZ FIN	
	JMP BUCLE	
TRES:	MVI A,0B0H	
	OUT 07H	
	INX H	
	CALL RETARDO	
	DCR B	
	MOV A,B	
	CPI 00H	
	JZ FIN	
	JMP BUCLE	
CUATRO:	MVI A,099H	
	OUT 07H	
	INX H	
	CALL RETARDO	
	DCR B	
	MOV A,B	
	CPI 00H	
	JZ FIN	
	JMP BUCLE	
CINCO:	MVI A,92H	
	OUT 07H	
	INX H	
	CALL RETARDO	
	DCR B	
	MOV A,B	
	CPI 00H	
	JZ FIN	
	JMP BUCLE	
SEIS:	MVI A,82H	
	OUT 07H	
	INX H	
	CALL RETARDO	
	DCR B	
	MOV A,B	
	CPI 00H	
	JZ FIN	
	JMP BUCLE	

```

SIETE:      MVI A,46H
            OUT 07H
            INX H
            CALL RETARDO
            DCR B
            MOV A,B
            CPI 00H
            JZ  FIN
            JMP BUCLE

OCHO:       MVI A,00H
            OUT 07H
            INX H
            CALL RETARDO
            DCR B
            MOV A,B
            CPI 00H
            JZ  FIN
            JMP BUCLE

NUEVE:      MVI A,10H
            OUT 07H
            INX H
            CALL RETARDO
            DCR B
            MOV A,B
            CPI 00H
            JZ  FIN
            JMP BUCLE

;-----
;subrutina FIN: se encarga de apagar y encender el display 5 veces
;para indicar el fin del programa.
;-----

FIN:        MVI A,08H
            OUT 05H
            CALL RETARDO
            MVI A,00H
            OUT 05H
            CALL RETARDO

            MVI A,08H
            OUT 05H
            CALL RETARDO
            MVI A,00H
            OUT 05H
            CALL RETARDO

            MVI A,08H
            OUT 05H
            CALL RETARDO
            MVI A,00H
            OUT 05H
            CALL RETARDO

```

```

MVI A,08H
OUT 05H
CALL RETARDO
MVI A,00H
OUT 05H
CALL RETARDO

MVI A,08H
OUT 05H
CALL RETARDO
MVI A,00H
OUT 05H
CALL RETARDO
HLT

;-----
;subrutina RETARDO
;-----

RETARDO:
MVI E,FFH           ;carga en el registro E FFh
NEXT:  MVI D,20H      ;carga en el registro D 20h
BACK:  DCR D          ;decremento el registro D
      JNZ BACK        ;sino es 0 el registro de seguir
      ;decrementando
      DCR E           ;si es 0 decrementar el registro E
      JNZ NEXT        ;sino es 0 el registro E volver a decrementar
      ;el registro D hasta que sea 0
      RET            ;si es 0 el registro E retornamos

```

## CONCLUSION

Como vimos, se diseñó y un programa capaz de leer un numero de cedula introducido en el microprocesador 8085 a través del teclado numérico, luego guardarlos y que por último al presionar la tecla \* en el teclado se pudiera mostrar cada dígito que se presionó en un display, esto se pudo realizar con éxito utilizando el simulador granada y se mostró un video donde se veía su funcionamiento.

Posteriormente diseñamos el circuito real que fuera capaz de realizar este mismo programa, donde introducimos factores externos al 8085, al hacer esto tuvimos que rediseñar el programa tomando en cuenta estos factores externos, por lo cual este último programa no se pudo simular, ya que el simulador granada no está diseñado para tomarlos en cuenta y los simuladores más comunes como lo son proteus o multisim no cuentan con una MPU equivalente al microprocesador 8085.

## REFERENCIAS CONSULTADAS

Armengol, A. (2020). *Tarea conexión del 8355/8755*. Caracas, Venezuela: Universidad Simón Bolívar.

Vazquez, Celestino (1999) “Microprocesador 8085, Curso del microprocesador 8085-A fabricado por Intel”