República Bolivariana de Venezuela

Ministerio del Poder Popular para la Educación

Universidad “Simón Bolívar”

Sede Litoral

Microprocesadores

Utilización de Interrupciones y Subrutinas para la Encriptación y Desencriptación de un Texto

Profesor:

Armengol, Alberto

Estudiante:

16-00131 Mayora, Adrián

Camurí Grande, abril de 2021

Programa Realizado:

Programa que encripta y desencripta un texto mostrado en una pantalla, utilizando el teclado del simulador granada.

```
.define
             texto E000h               ; Dirección comienzo pantalla

; -------------------------------
; PROGRAMA PRINCIPAL
; -------------------------------
             .org 1000h
BUCLE: EI                              ;Habilito interrupciones
            JMP BUCLE                  ;Bucle constante, solo sale con interrupción

; ----------------------------------------------------------------
; RUTINA DE INTERRUPCIÓN QUE DETECTA SI: A)SE PRECIONA TECLA "E" ENCRIPTA, B)SI SE
PRCIONA LA TECLA "D"
; DESMCRIPTA Y C)SI SE PRECIONA CUALQUIER OTRA TECLA, REGRESA AL BUCLE PRINCIPAL
; ----------------------------------------------------------------
.org 0034h
             IN 00                     ;Leemos lo que este en el puerto 0
             CPI 'E'                   ;Comparamos la tacla presionada con el
                                       ;ASCII de E
             JNZ PASAR1                ;Si no es E, ver si es D
             INR E                     ;Si es E aviso de que encripté
             CALL ENCRIPTA             ;Encriptar el texto
             JMP PASAR2                ;Luego de encriptar regreso al programa
                                       ;principal
PASAR1:      MOV A, E                  ;
             CPI  00h                  ;Reviso si ya he encriptado al menos 1 vez
             JZ PASAR5                 ;Si no he encriptado coloco un codigo ASCCI
                                       ;distinto al de D
             IN 00h                    ;Si ya encripté leo nuevamente el caracter
             JMP PASAR6                ;Salto a comparar directamente el caracter con
                                       ;el ASCII de D
PASAR5:      MVI A,30h                 ;Coloco un codigo ASCCI cualquiera distinto del
                                       ;de D
PASAR6:      CPI 'D'                   ;Comparamos la tacla presionada con el
                                       ;ASCII de D
             JNZ PASAR2                ;Si no es D volver al programa principal
             DCR E                     ;Si es D aviso que desemcripté
             CALL DESENCRIPTA          ;Encriptar el texto
PASAR2:      RET                       ;Regrsar al programa principal
```

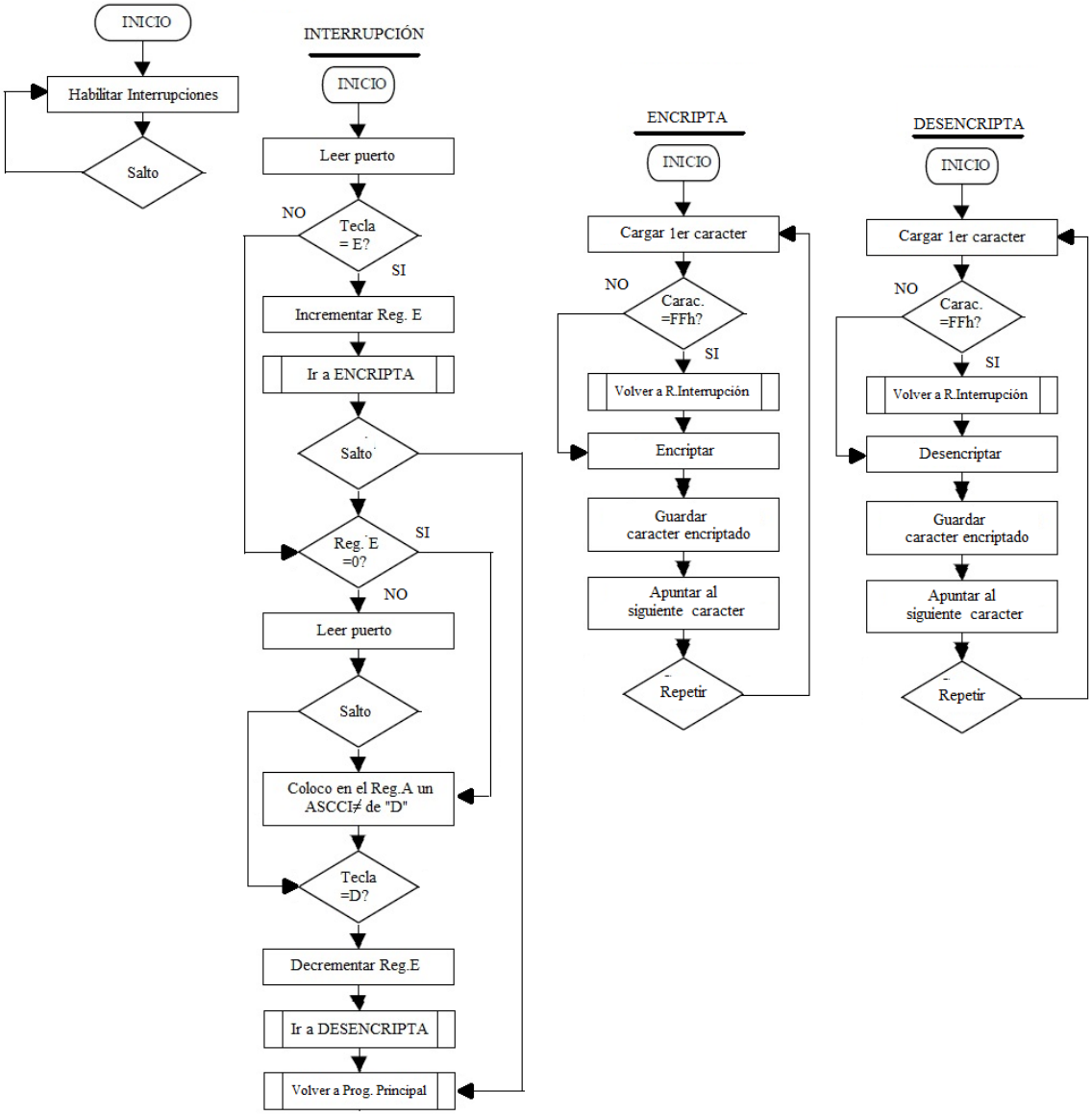
```
; ------------------------------------------------
; SUBRUTINA QUE ENCRIPTA EL TEXTO
; ------------------------------------------------
ENCRIPTA:
            LXI H, texto              ;Cargamos dirección origen del texto
repite_1:   MOV A,M                   ;Copiamos en A el primer caracter
            CPI FFH                   ;Comprobamos si el codigo ASCII de la letra no es FFh
            JNZ PASAR3                ;SI no es FFh encriptar caracter
            RET                       ;Si es FFh volver a la rutina de interrupción
PASAR3:     ADI 07h                   ;Encriptar caracter
            MOV M,A                   ;Guardamos el caracter encriptado
            INX H                     ;Incrementar el apuntador
            JMP repite_1              ;continuar con el siguiente caracter


; ----------------------------------------------
; SUBRUTINA QUE DESENCRIPTA EL TEXTO
; ----------------------------------------------
DESENCRIPTA:
            LXI H, texto              ;cargamos dirección origen
repite_2:   MOV A,M                   ;Copiamos en A la primera caracter
            CPI FFH                   ;Comprovamos si codigo ASCCI de la letra no es FFh
            JNZ PASAR4                ;SI no es FFh desencriptar caracter
            RET                       ;Si es FFh volver a la rutina de interrupcion
PASAR4:     SUI 07h                   ;Desencriptar carater
            MOV M,A                   ;Guardamos el caracter encriptado
            INX H                     ;incrementar el apuntador
            JMP repite_2              ;continuar con el siguiente caracter



.data E000h
DB 41h, 64h, 72h, 69h, 61h, 6Eh, 20h, 4Dh, 61h, 79h, 6Fh, 72h, 61h, 20h, 43h, 61h, 72h,
6Eh, 65h, 74h, 3Ah, 31h, 36h, 30h, 30h, 31h, 33h, 31h, FFh
```

Diagrama de Flujo:



Análisis del programa realizado:

Procederemos a realizar el análisis del programa por partes:

Parte I: Programa principal

Como vemos en el cuadro de abajo, comenzamos definiendo como texto a la dirección origen donde se almacenaran los caracteres que se mostraran en la pantalla, seguido, tenemos el programa principal que almacenamos a partir de la dirección 1000h.

Dentro del programa principal podemos ver un bucle que se repetirá indefinidamente, y solo se podrá salir de él con una interrupción:

```
.define
            texto E000h              ;Dirección comienzo de la
                                     ;pantalla

; -------------------------------
; PROGRAMA PRINCIPAL
; -------------------------------
.org 1000h

BUCLE:      EI                       ;Habilito interrupciones
            JMP BUCLE                ;Bucle constante, solo sale
                                     ;con interrupción
```

Parte II: Rutina De Interrupción

Una vez pulsemos cualquier tecla del teclado, se producirá un interrupción RST 6,4, lo cual hará un llamado a la dirección 0034h, donde se encuentra nuestra rutina de interrupción:

```
org 0034h
            IN 00                    ;Leemos lo que este en el puerto 0
            CPI 'E'                  ;Comparar la tacla presionada con el
                                     ;ASCII de E
            JNZ PASAR1               ;Si no es E, ver si es D
            INR E                    ;Si es E aviso de que encripté
            CALL ENCRIPTA            ;encriptar el texto
            JMP PASAR2               ;Luego de encriptar regreso al
                                     ;programa principal
PASAR1:     MOV A, E
```

Ahora bien, como vemos el programa comienza leyendo el puerto 00h , de forma que justo después de pulsar cualquier tecla, tengamos en el acumulador el código ASCII de la tecla que se pulsó.

```
org 0034h
            IN 00h                  ;Leemos lo que este en el puerto 0
            CPI 'E'                 ;Comparar la tacla presionada con el
                                    ;ASCII de E
            JNZ PASAR1              ;Si no es E, ver si es D
            INR E                   ;Si es E aviso de que encripté
            CALL ENCRIPTA           ;encriptar el texto
            JMP PASAR2              ;Luego de encriptar regreso al
                                    ;programa principal
PASAR1:     MOV A, E
;
```

Justo después de esto, comprobamos si la tecla que se pulsó es la tecla “E”, y en caso que lo sea llamar a la subrutina “ENCRIPTA” para encriptar el texto, sino seguimos el programa normalmente.

También como vemos, justo al confirmarse que la tecla pulsada fue “E” se incrementa el registro E, esto lo hacemos por 2 cosas:

1) Para saber cuántas veces se a encriptado el texto, ya que el registro E solo se va a incrementar cada vez que encriptemos

2) Para saber si no hemos realizado ninguna encriptación.

Esto último tendrá su utilidad más adelante.

```
.org 0034h
            IN 00                   ;Leemos lo que este en el puerto 0
            CPI 'E'                 ;Comparar la tacla presionada con el
                                    ;ASCII de E
            JNZ PASAR1              ;Si no es E, ver si es D
            INR E                   ;Si es E aviso de que encripté
            CALL ENCRIPTA           ;encriptar el texto
            JMP PASAR2              ;Luego de encriptar regreso al
                                    ;programa principal
PASAR1:     MOV A, E
```

También como vemos, se añadió un salto incondicional después de de la llamada a la subrutina "ENCRIPTA", esto se hizo para que justo después de regresar de la subrutina, regresemos directamente al bucle del programa principal, y así obligar a usuario que está utilizando el teclado a pulsar otra tecla para poder seguir el programa, de forma que no se produzcan errores:

```
.org 0034h
            IN 00                   ;Leemos lo que este en el puerto 0
            CPI 'E'                 ;Comparar la tacla presionada con el
                                    ;ASCII de E
            JNZ PASAR1              ;Si no es E, ver si es D
            INR E                   ;Aviso de que encripte
            CALL ENCRIPTA           ;si es E encriptar el texto
            JMP PASAR2              ;Luego de encriptar regreso al
                                    ;programa principal
PASAR1:     MOV A, E
            CPI  00h                ;Reviso si ya he encriptado al menos
                                    ;1 vez
            JZ PASAR5               ;si no he encriptado coloco un
                                    ;codigo ASCCI ;distinto al de D
            IN 00                   ;Si ya encripte leo nuevamente el
                                    ;caracter
            JMP PASAR6              ;Salto a comparar el caracter con el
                                    ASCCI de D
PASAR5:     MVI A,30H               ;Coloco un codigo ASCII cualquiera
                                    ;distindo del de D
PASAR6:     CPI 'D'                 ;Comparamos la tacla presionada con
                                    ;el ASCII de D
            JNZ PASAR2              ;Si no es D volver al
                                    ;programaprincipal
            DCR E                   ;Aviso que desemcripte
            CALL DESENCRIPTA        ;Si es D encriptar el texto
PASAR2:     RET                     ;Regrsar al programa principal
```

Ahora como vemos, si la tecla que se pulso no fue la “E” el programa comprobara si la tecla que se pulso es la “D” y en caso afirmativo desencriptará el texto, sin embargo, antes de esto, se añadió una serie de instrucciones para comprobar si antes de desencriptar el texto, ya se había sido encriptado antes, de modo que no se desencripte el texto si no se a encriptado anteriormente.

Aquí entra en juego el registro E, ya que si este está en 0 significa que no se ha realizado ninguna encriptación (por lo tanto no podemos realizar la desencriptación), y si no es 0 significa que si se realizado alguna encriptación (por lo tanto si podemos desencintar el texto):

```
.org 0034h
             IN 00                    ;Leemos lo que este en el puerto 0
             CPI 'E'                  ;Comparar la tacla presionada con el
                                      ;ASCII de E
             JNZ PASAR1               ;Si no es E, ver si es D
             INR E                    ;si es E aviso de que encripté
             CALL ENCRIPTA            ;encriptar el texto
             JMP PASAR2               ;Luego de encriptar regreso al
                                      ;programa principal
PASAR1:      MOV A, E
             CPI  00h                 ;Reviso si ya he encriptado al menos
                                      ;1 vez
             JZ PASAR5                ;si no he encriptado coloco un
                                      ;codigo ASCII distinto al de D
             IN 00                    ;Si ya encripté leo nuevamente el
                                      ;caracter
             JMP PASAR6               ;Salto a comparar el caracter con el
                                      ASCCI de D
PASAR5:      MVI A,30H                ;Coloco en A un codigo ASCCI
                                      ;cualquiera distindo del de D
PASAR6:      CPI 'D'                  ;Comparamos la tacla presionada con
                                      ; el ASCII de D
             JNZ PASAR2               ;Si no es D volver al
                                      ;programaprincipal
             DCR E                    ;Si es D aviso que desemcripte
             CALL DESENCRIPTA         ;Encriptar el texto
PASAR2:      RET                      ;Regrsar al programa principal
```

Luego de comprobar si anteriormente se a encriptado el texto o no, ahora si, procedemos a comprobar si la tecla que se pulsó es la “D”, como vemos, si no es D retornamos al bucle del programa principal, y si es D decrementaremos el registro E, para señalar que vamos a realizar una desencriptación del texto y llamamos a la subrutina “DESEMCRIPTA” para desencriptar el texto, nótese que justo después de volver de la subrutina retornaremos a al programa principal, gracias a la instrucción RET.

```
.org 0034h
             IN 00                    ;Leemos lo que este en el puerto 0
             CPI 'E'                  ;Comparar la tacla presionada con el
                                      ;ASCII de E
             JNZ PASAR1               ;Si no es E, ver si es D
             INR E                    ;Aviso de que encripte
             CALL ENCRIPTA            ;si es E encriptar el texto
             JMP PASAR2               ;Luego de encriptar regreso al
                                      ;programa principal
PASAR1:      MOV A, E
             CPI  00h                 ;Reviso si ya he encriptado al menos
                                      ;1 vez
             JZ PASAR5                ;si no he encriptado coloco un
                                      ;codigo ASCCI ;distinto al de D
             IN 00                    ;Si ya encripte leo nuevamente el
                                      ;caracter
             JMP PASAR6               ;Salto a comparar el caracter con el
                                      ;ASCII de D
PASAR5:      MVI A,30H                ;Coloco en A un codigo ASCCI
                                      ;cualquiera distindo del de D
PASAR6:      CPI 'D'                  ;Comparamos la tacla presionada con
                                      ;el ASCII de D
             JNZ PASAR2               ;Si no es D volver al
                                      ;programa principal
             DCR E                    ;Aviso que desemcripté
             CALL DESENCRIPTA         ;Si es D encriptar el texto
PASAR2:      RET                      ;Regrsar al programa principal
```

Parte III: subrutina de encriptación

En este caso, comenzamos directamente la subrutina, colocando en el apuntador la dirección de origen del texto, como vemos no es necesario utilizar las instrucciones PUSH y POP en esta subrutina , ya que como mencionamos anteriormente, luego de finalizar la subrutina de "ENCRIPTA" habrá un salto que nos regresará directamente al bucle del programa principal, donde no necesitaremos de ningún registro hasta ahora utilizado:

```
ENCRIPTA:
            LXI H, texto              ;cargamos dirección origen del texto

repite_1:   MOV A,M                   ;Copiamos en A el primer caracter
            CPI FFH                   ;Comprobamos si el codigo ASSCI del
                                      ;caracter no es FFh
            JNZ PASAR3                ;SI no es FFh encriptar caracter
            RET                       ;Si es FFh volver a la rutina de
                                      ;interrupcion
PASAR3:     ADI 7H                    ;Encriptar carater
            MOV M,A                   ;Guardamos el caracter encriptado
            INX H                     ;Incrementar direccion

            JMP repite_1              ;continuar con el siguientes
                                      ;caracter
```

Como vemos, luego de apuntar el origen del texto, procedemos a copiar en el acumulador el primer carácter del texto, y luego se añadió un condicional, de modo que si el código ASCII del carácter que estamos leyendo es igual FFh, finalice la subrutina y retornemos al bucle del programa principal, sino, encriptamos el caracter, guardamos el carácter encriptado en la misma dirección de donde sacamos el caracter sin encriptar, incrementamos el par de registros HL, para apuntar al siguiente carácter y volvemos repetir este ciclo, hasta encontrar el código ASCII de FFh, que esta al final del texto.

```
ENCRIPTA:
            LXI H, texto              ;cargamos dirección origen deltexto

repite_1:   MOV A,M                   ;Copiamos en A el primer caracter
            CPI FFH                   ;Comprobamos si el codigo ASSCI del
                                      ;caracter no es FFh
            JNZ PASAR3                ;SI no es FFh encriptar caracter
            RET                       ;Si es FFh volver a la rutina de
                                      ;interrupcion
PASAR3:     ADI 07h                   ;Encriptar carater
            MOV M,A                   ;Guardamos el caracter encriptado
            INX H                     ;Incrementar direccion

            JMP repite_1              ;continuar con el siguiente
                                      ;caracter
```

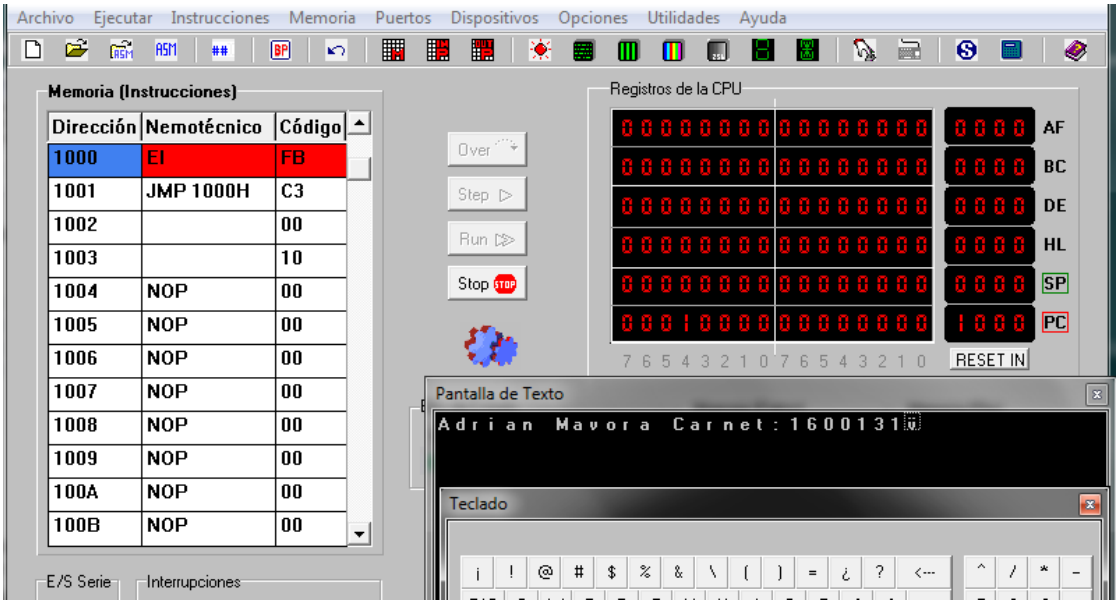
Parte IV: Subrutina de Desencriptación

La subrutina de desencriptación es exactamente igual a la de encriptación, con la diferencia de que se realiza el proceso inverso de encriptar, como vimos en la subrutina de "ENCRIPTA" simplemente se sumo 07h al carácter que se iba a encriptar, por lo tanto para desencriptar se debe restar 07h al carácter que se va a desencriptar:

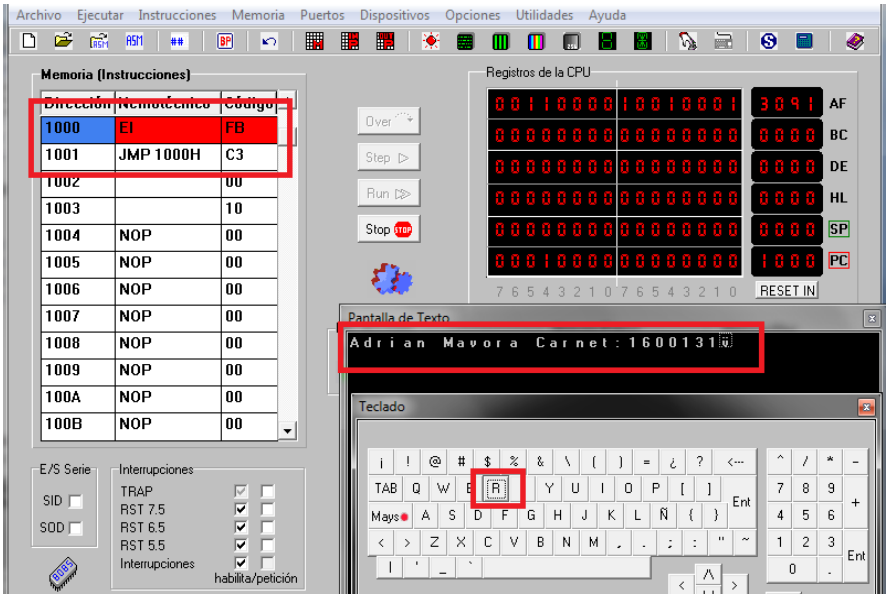
```
DESENCRIPTA:
            LXI H, texto            ;cargamos dirección origen
repite_2:   MOV A,M                 ;Copiamos en A el primera caracter
            CPI FFH                 ;Comprobamos si el codigo ASCCI del
                                    ;caracter no es FFh
            JNZ PASAR4              ;SI no es FFh desencriptar caracter
            RET                     ;Si es FFh volver a la rutina de
                                    ;interrupcion
PASAR4:     SUI 07h                 ;Desencriptar carater
            MOV M,A                 ;Guardamos el caracter encriptado
            INX H                   ;incrementar direccion
            JMP repite_2            ;continuar con el sguiente caracter
                                    ;continuar con el sguiente caracter
```

Proceso en el simulador Granada:

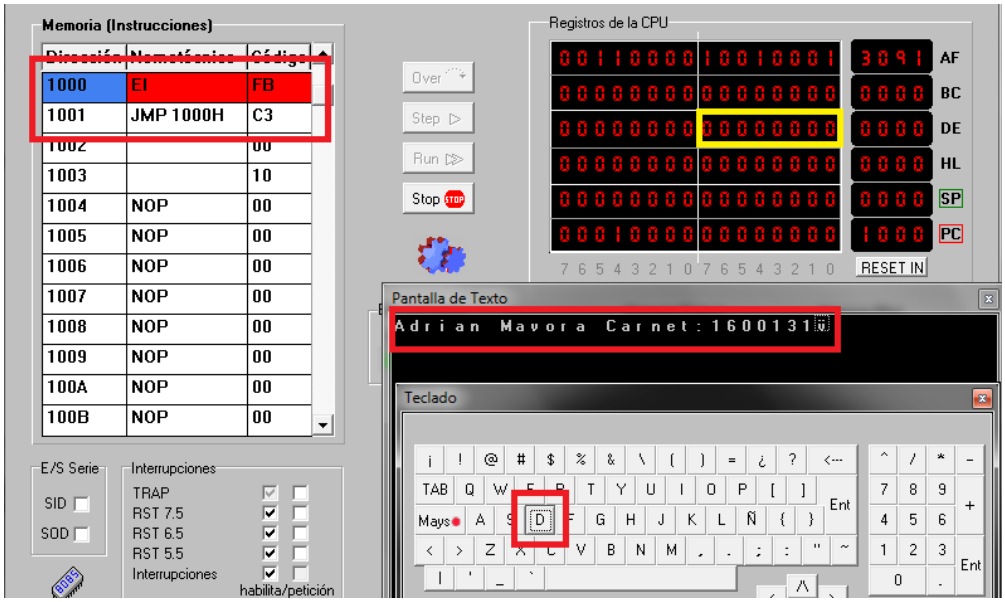
Como vemos en la imagen de abajo, al correr el programa, este empieza a ejecutar el bucle principal, esperando a que presionemos alguna tecla:



Ahora antes de realizar la primera encriptación, primero comprobamos, si al presionar cualquier tecla que no sea E o D, se produce algún error, y como vemos no ocurre nada, ya que la rutina de interrupción está diseñada para que regrese al bucle del programa principal si se preciona cualquier tecla que no sea E o D, como se explicó anteriormente:

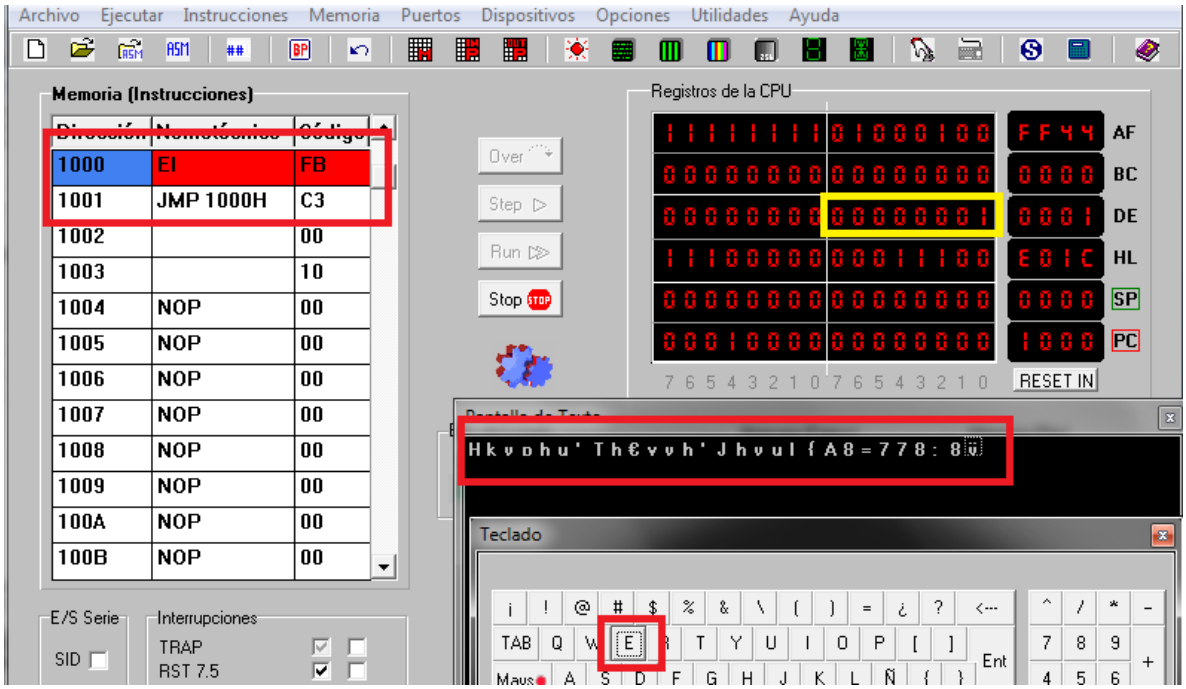


Ahora como vemos en la imagen de abajo, se presiono a la tecla D antes de desemcriptar, y podemos observar, no sucedió nada ya que el registro E se encuentra en 00h, y como vimos anteriormente, el programa está hecho para que si esto ocurre, nos regrese al bucle del programa principal, hasta que pulsemos otra tecla:



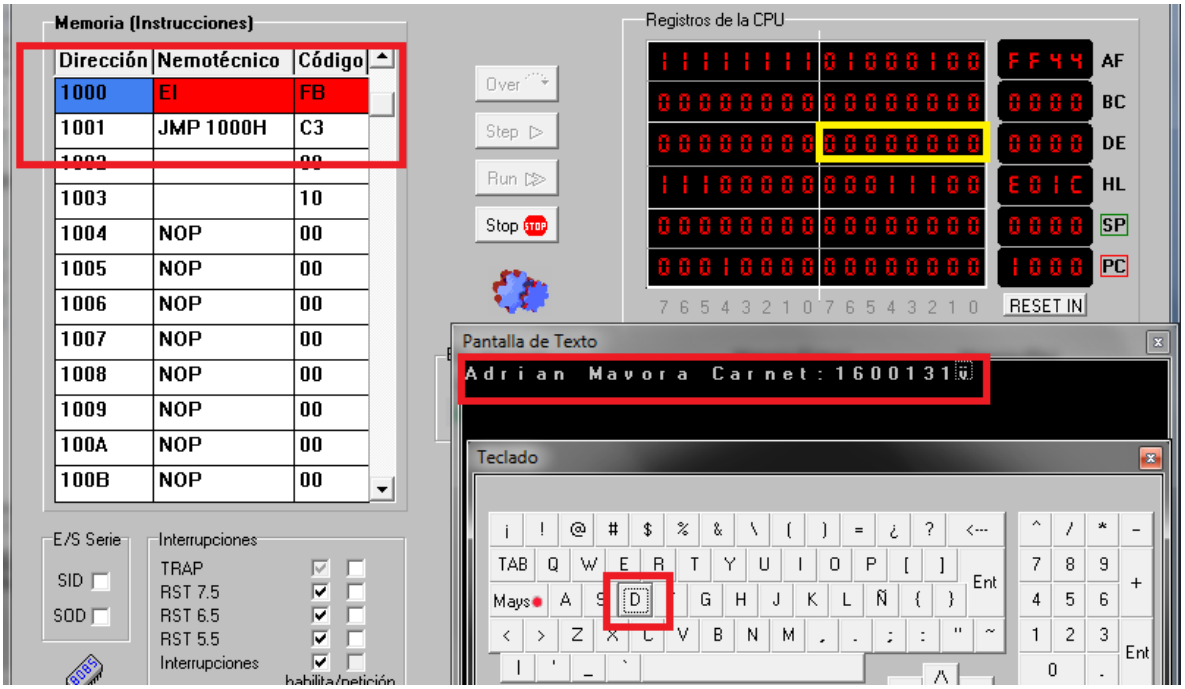
Finalmente, al presionar la tecla E del teclado se encripta el texto mostrado en la pantalla y también se incrementa el registro E, señalándonos que se a realizado una encriptación.

Y podemos ver en la imagen de abajo, luego de encriptar, regresamos al bucle del programa principal, donde el programa espera el siguiente pulso de tecla:



Ya que realizamos una encriptación, el registro E ya no es 0, por lo tanto como vemos en la imagen de abajo al presionar la tecla D, ahora si pudimos desencriptar el texto.

También podemos ver que se decremento el registro E, para señalar que se realizó una desencriptación, y así no poder seguir desencriptando el texto, ya que al decrementar el registro E, este vuelvió 0.



Referencias consultadas:

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