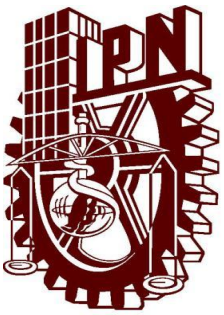

INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

INTRODUCCIÓN A LOS MICROCONTROLADORES

PRACTICA #3

Integrantes:

- ✓ ***Lomelí García Martín***
- ✓ ***Pacchiano Alemán Alain***
- ✓ ***Trejo Martínez Francisco***

PROFESOR: Pérez Pérez José Juan

GRUPO: 3CM3 ***22/11/2016***

MARCO TEÓRICO

Rutinas y subrutinas

Una pila es una estructura de datos caracterizada por que el último dato que se almacena es el primero que se obtiene después. Para gestionar la pila se necesita un puntero a la última posición ocupada de la misma, con el fin de conocer dónde se tiene que dejar el siguiente dato a almacenar, o para saber dónde están situados los últimos datos almacenados en ella. Para evitar problemas, el puntero de pila siempre debe estar apuntando a una palabra de memoria. Por precedentes históricos, el segmento de pila siempre "crece" de direcciones superiores a direcciones inferiores. Las dos operaciones más típicas con esta estructura son:

Transferir datos hacia la pila (push o apilar). Antes de añadir un dato a la pila se tienen que restar 4 unidades al puntero de pila.

Transferir datos desde la pila (pop o desapilar). Para eliminar datos de la pila, después de extraído el dato se tienen que sumar 4 unidades al puntero de pila.

RCALL realiza una llamada relativa a una dirección que se encuentra dentro de PC - 2K + 1 y PC +2K (words). En ensamblador, se usan etiquetas en lugar de operadores relativos. Para los microcontroladores AVR con memoria de programa que no excede 4K words (8K bytes) esta instrucción puede dirigirse a toda la memoria desde cualquier posición de dirección. El Stack Pointer (puntero de pila) utiliza durante el RCALL un esquema de post-decremento.

RJMP realiza un salto relativo a una dirección que se encuentra dentro de PC - 2K + 1 y PC +2K (words). En ensamblador, se usan etiquetas en lugar de operadores relativos. Para los microcontroladores AVR con memoria de programa que no excede 4K words (8K bytes) esta instrucción puede dirigirse a toda la memoria desde cualquier posición de dirección

Código fuente del programa

Esta práctica consistió en la aplicación de rutinas y subrutinas, para que el programa fuera mostrando en pantalla un contador, realizado desde el 0-9 con un cambio cada segundo. Para lograr la espera de un segundo entre cada cambio de mostrar el número en el display se hizo uso de rutinas y subrutinas, específicamente de la llamada ***rjmp*** que nos permitía ir a un código de bloque que duraba exactamente el tiempo deseado en ejecutarse.

La parte 1 consistió en decodificar la entrada (números de 0-9 en binario) y mostrar su respectiva representación decimal en un display de 7 segmentos.

```
.include "m8535def.inc"
.def dato =r17

ldi r21,$3f
ldi r22,$06
ldi r23,$5b
ldi r24,$4f
ldi r25,$66
ldi r26,$6d
ldi r27,$7d
ldi r28,$27
ldi r29,$7f
ldi r30,$6f

ldi r16,0

ldi r18,1

ciclo:
        ldi r17,20
        add r16,r18
        add r17,r16
        out portb,dato
        rjmp delay

continuar:
        ;RCALL delay
        cpi r16,11
        breq reiniciar
        rjmp ciclo


reiniciar:
        ldi r16,0
        rjmp ciclo

delay:
        ldi r17,11
        ldi r19,38
        ldi r20,94
L1:
        dec r20
        brne L1
        dec r19
        brne L1
        dec r17
        brne L1
        rjmp continuar
```

CONCLUSIONES

Lomelí García Martín: En la práctica 3 aprendimos a programar en ensamblador con el uso de rutinas y subrutinas, con llamadas al sistema como rjmp o rcall que nos permitían ejecutar algún bloque de código y al finalizar este regresar al momento cuando lo mandamos a llamar, o incluso ejecutar otro bloque totalmente distinto y localizado en otra parte de la pila del programa.

Pacchiano Alemán Alain: La práctica solo consistió en la implementación de rutinas con las llamadas de rjmp, permitiendo ejecutar algún bloque de código no secuencial.

Trejo Martínez Francisco: En esta ocasión aprendimos a implementar las llamadas de rcall y rjmp, que nos permiten ir a una parte específica de la pila del programa y ejecutar las acciones escritas ahí mismo, y al finalizar ese bloque ir a otro totalmente distinto.