Informe de laboratorio 3 Electrónica digital 2

Abstract

Se implementó un código en lenguaje de máquina para LC3 vía el simulador de este procesador, se utilizó Asembler para hacer dicho código. Tal código consiste en capturar N datos numéricos por consola siendo N también dado por el usuario, guardarlos en memoria y hacer las operaciones de ordenarlos, sacar el menor y los múltiplos de 8, luego de hacer dichas operaciones se debe de mostrar el resultado al usuario y permitirle ingresar de nuevo los datos al usuario.

Todo esto se hace de manera modular llamando subrutinas y haciendo comunicación por memoria principalmente.

Pseudocódigo

**Programa para capturar datos**

Rs: Max chrs

R1: Valor final

R2: Ch ingresado

GetChar getc

If(R0 == 10)

JUMP OUTVALUE

Elsif(R0 == 1 && R2 ¡= 0)

JUMP GETCHAR

Elsif(R0 < X30 || R0 > X39)

JUMP GETCHAR

ELSIF(R2 >= R3)

JUMP GETCHAR

OUT = R0 &XF

R1 = R1\*10 + R0

R2++

MEMORIA <= R1

El anterior algoritmo me valida el número y me guarda en memoria

**Programa para capturar varios datos**

For (i = 0; i<N; I++)

Memory(i) = getchar

End;

Programa para menú

R1 = LOADMEMORY

LEA R0, DISPLAYMENU

PUTS

GETCHAR

IF(R0 == x31)

JUMPFUN1

ELSIF(R0 == x31)

JUMPFUN2

ELSIF(R0 == x32)

JUMPFUN3

ELSE

HALT

DISPLAYMENU =”Menu ingrese opción”

**Función que imprime datos**

If(OP == 0)

LDI R0

PRINTCHAR

ELSIF(OP == 1)

FOR(I = 0, I <N; I++)

FOR(I = 0, I <LENNUM; I++)

NUM = NUM%10\*\*i

PRINTCHAR

END

END

JMPGETNUM

PRINTCHAR: PRINT(ASCII(NUM+X0030))

**Función que ordena (con esta misma cojo la del número menor ya que en la primera posición de memoria del arreglo organizado está el número).**

n = len(A)

**loop**

swapped = **false**

**for** i = 1 **to** n-1 inclusive **do**

**if** A[i-1] > A[i] **then**

swap( A[i-1], A[i] )

swapped = **true**

**end** **if**

**end** **for**

**until** **not** swapped

**end** **procedure**

Se utilizó el bubble sort como algoritmo ya que es el más sencillo de implementar sin necesidad de hacer operaciones complejas que ayudan a hacer su código en LC3

**Conclusiones**

Programar en ensamblador requiere de una especial atención a los registros e instrucciones además de requerir un esfuerzo mayor por parte del programador ya que es un lenguaje muy limitado y se necesita traducir del lenguaje humano a una serie muy pequeña de instrucciones que cada una tiene claras limitaciones no solamente en las funciones que puede hacer sino también con los valores que lo puede hacer, así que a la hora de programar también se debe tener en cuenta estas limitaciones en el tamaño de los datos que se pueden representar y el tamaño de los datos que cada instrucción da. Esta manera de programar da la mayoría de la responsabilidad al programador por lo que generalmente encontrar errores es relativamente sencillo si se sabe lo que el programa está haciendo claramente ya que la máquina no interpretará nada extra ni habrá espacio para las incertidumbres. También es difícil entender código en este lenguaje especialmente si se está iniciando o si no está bien comentado, ya que todo el programa depende de todo por lo que generalmente para entender un programa de estos se debe entender en su totalidad y no por partes ya que estas partes en las que no se entiende se puede modificar partes que si se tenían entendidas.