



UNIVERSIDAD AUTONOMA DE SAN LUIS POTOSÍ

FACULTAD DE INGENIERÍA

ÁREA DE CIENCIAS DE LA COMPUTACIÓN

“LABORATORIO PROGRAMACIÓN DE SISTEMAS”

**PRACTICA 11:
“Paso 2 – SIC Extendida”**

**PROFESOR:
ING. AGUSTÍN HERNÁNDEZ GARCÍA**

ALUMNO: Braulio Alejandro García Rivera

CLAVE ÚNICA: 239196

FECHA: 06/12/2020

SEMESTRE: 2020-2021/I

1. INTRODUCCIÓN

En este documento se explica cómo fue la elaboración del paso 2 de la SIC Extendida mediante los métodos correspondientes para su correcto funcionamiento.

2. INSTALACIÓN Y CONFIGURACIÓN

En Visual Studio 2017 se instalaron las siguientes extensiones:

- Antlr4Code
- ANTLR Language Support

También se instalaron las bibliotecas de Antlr4 como paquetes NuGet:

- Antlr4 v4.6.6
- Antlr4.CodeGenerator v4.6.6
- Antlr4.Runtime v4.6.6

La configuración se realizó en base a la explicación del profesor y los ejemplos de la plataforma didacTIC.

3. INTERFAZ

4. BANDERAS N, I, X, B, P, E DESPLAZAMIENTO/DIRECCIÓN

Para las banderas se utilizó un arreglo de caracteres con las etiquetas donde dependiendo el modo de direccionamiento se empezaban a rellenar las posiciones correspondientes para poder generar una sola variable con posiciones detectables y determinar el código objeto correcto.

5. ENSAMBLADO INSTRUCCIONES FORMATO 3 Y 4

El ensamblado de las instrucciones se generó utilizando una variable con las banderas y dos métodos para calcular si es relativo a la base y al contador y por medio de banderas detectar de que tipo es o si esta fuera del rango y de esta manera poder cargar los datos correctos, si es formato tres solo se carga el desplazamiento y en formato cuatro la dirección correspondiente.

6. REGISTROS DE MODIFICACION

Sistema Interprete

Codigo Fuente

Linea	Etiqueta	Instruccion	Operando
1	SUM	START	0h
2	FIRST	LDX	#0
3		LDA	#0
4		LDB	#TABLE2
5		BASE	TABLE2
6	LOOP	ADD	TABLE.X
7		ADD	TABLE2.X
8		TIX	COUNT
9		JLT	LOOP
10		+STA	TOTAL
11		RSUB	
12	COUNT	RESW	1
13	TABLE	RESW	2000
14	TABLE2	RESW	2000
15	TOTAL	RESW	1
16		END	FIRST

Archivo Intermedio

Linea	CP	Etiqueta	Instruccion	Direccion	Error	Codigo Obj
1	0	SUM	START	0h	No	--
2	0	FIRST	LDX	#0	No	050000
3	3		LDA	#0	No	010000
4	6		LDB	#TABLE2	No	694000
5	9		BASE	TABLE2	no	--
6	9	LOOP	ADD	TABLE.X	No	1BA013
7	C		ADD	TABLE2.X	No	1BC000
8	F		TIX	COUNT	No	2F200A
9	12		JLT	LOOP	No	3B2FF0
10	15		+STA	TOTAL	No	F102EFF*
11	19		RSUB		No	4F0000
12	1C	COUNT	RESW	1	No	--
13	1F	TABLE	RESW	2000	No	--

Simbolo	Direccion
FIRST	0
LOOP	9
COUNT	1C
TABLE	1F
TABLE2	178F
TOTAL	2EFF

Resultados

Archivo Obj

Tamaño del archivo: 2F02

HSUM 000000002F02

T000000180500000100006940001BA0131BC0002F200A3B2FF0F102EFF4F0000

M00001605+SUM

E0000000

Errores

Para las modificaciones al momento de detectar un asterisco al momento de ensamblar se guarda el CP en una lista auxiliar para después proceder a relocalizarlo con los registros M.

7. GENERACION DEL VALOR HEXADECIMAL DESPLAZAMIENTO NEGATIVO EN INSTRUCCIONES DE FORMATO 3

Para trabajar los valores negativos al momento de ensamblar las instrucciones se utilizaron variables tipo long donde permitía el correcto guardado y verificación de valores negativos.

8. PROBLEMAS

Un problema encontrado fue al momento de detectar si una etiqueta de la tabla de símbolos era válida para poderla convertir a hexadecimal, esto se solucionó utilizando `IsNumeric` que devuelve un `True` si es una cadena que contiene un número hexadecimal válido.

9. POSIBLES MEJORAS

Una posible mejora seria al momento de generar los códigos objeto para poder hacerlo más eficiente al momento de la generación y que conlleva bastantes pasos a realizar.

10. CONCLUSIÓN

En esta práctica se implementó el paso 2 de la SIC Extendida donde se tuvieron diferentes métodos para poder generarlo correctamente y lograr un análisis

correcto de los archivos .sx para poder visualizar su respectivo código objeto, así como los registros correspondientes.