

REPORTE PRACTICA NO 3

Rogelio Daniel Gonzalez Nieto

LABORATORIO PROGRAMACION DE SISTEMAS AREA DE CIENCIAS DE LA COMPUTACION

OBJETIVO DE LA PRACTICA

Realizar el paso 1 para el ensamblador de la sic estándar el cual es:

- Asignar direcciones a todas las proposiciones del programa
- Guardar las direcciones asignadas a todas las etiquetas para usarse en el paso 2 (TABSIM)
- Realizar algún procesamiento de las instrucciones para el ensamblador (BYTE, RESW, etc.)

Para el cálculo de direcciones siempre se debe de sumar 3 bits a la última dirección asignada a excepción de cuando se tienen una cadena, en ese caso se suma 1 bíte por cada carácter, un valor hexadecimal se suma 1 bíte por cada 2 caracteres y un valor decimal se multiplica por 3 y se convierte a hexadecimal, ese valor se suma a la dirección.

CODIGO DE LA PRACTICA

```
private void calcularDirecciones()
{
    string dirHexa = "";
    int iDirDec = 0;
    int iPos = 0;
    int iTTotal = 0;
    foreach (var linea in this.ensamblador.programa)
    {
        if (this.ensamblador.programa.IndexOf(linea) == 0)
        {
            linea.sDireccionHEXA = IDirecciones[0].sDireccionHexadecimal;
            dirHexa = IDirecciones[0].sDireccionHexadecimal;
            dirHexa = dirHexa.Trim('H');
            iDirDec = Convert.ToInt32(dirHexa.ToString(), 16);
            //iDirDec = IDirecciones[0].iNumeroDecimal;
        }
        else
        {
            if (this.IDirecciones[iPos].cRegistro == 'C')
            {
                iTTotal = this.IDirecciones[iPos].sPalabra.Length;
                iDirDec = iDirDec + iTTotal;
                dirHexa = $"{iDirDec:X}";
                linea.sDireccionHEXA = dirHexa;
            }
            else if (this.IDirecciones[iPos].cRegistro == 'X' && this.IDirecciones[iPos].sDireccionHexadecimal != null)
            {
                iTTotal = this.IDirecciones[iPos].sDireccionHexadecimal.Length;
                if (iTTotal % 2 == 0)
                {
                    iTTotal = iTTotal / 2;
                }
                else
                {
                    iTTotal = (iTTotal / 2) + 1;
                }
                iDirDec = iDirDec + iTTotal;
                dirHexa = $"{iDirDec:X}";
                linea.sDireccionHEXA = dirHexa;
            }
            else if (this.IDirecciones[iPos].iNumeroDecimal != 0)
            {
                int iNuevo = this.IDirecciones[iPos].iNumeroDecimal * 3;
                iDirDec = iDirDec + iNuevo;
                //iDirDec = iDirDec * iNuevo;
                //dirHexa = $"{iDirDec:X}";
                dirHexa = $"{iDirDec:X}";
                linea.sDireccionHEXA = dirHexa;
            }
        }
    }
}
```

```

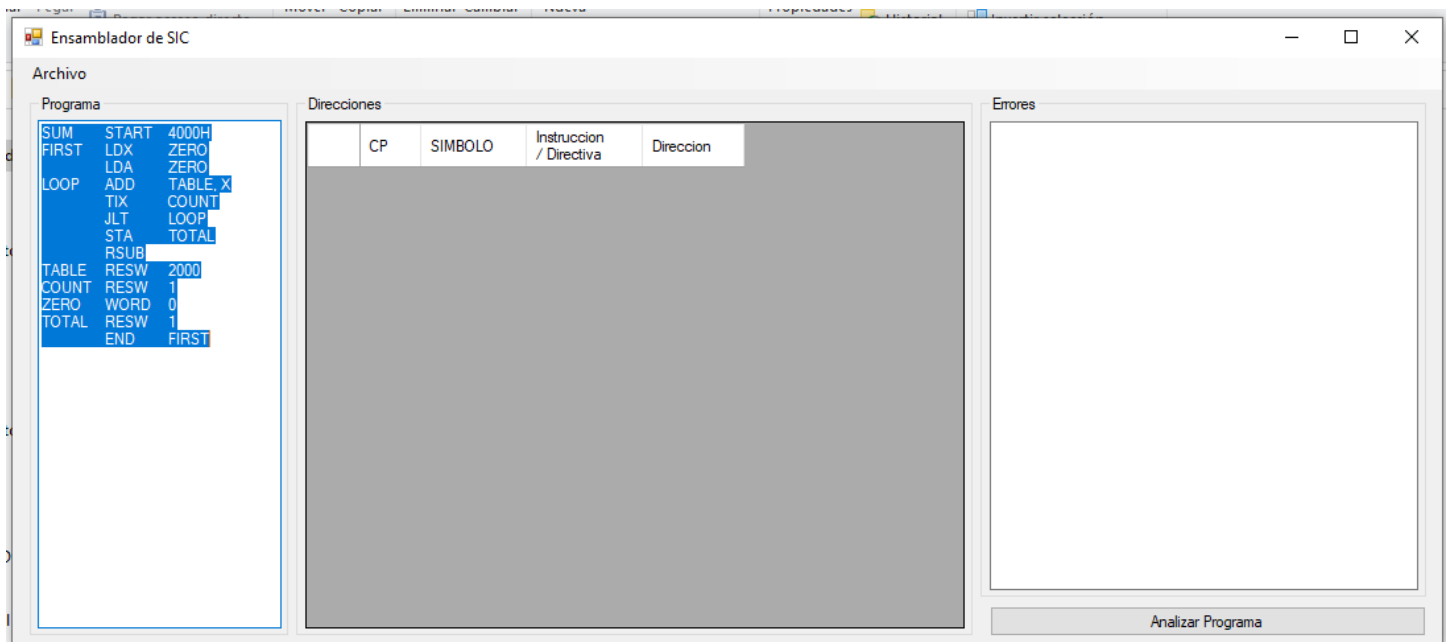
        {
            iDirDec = iDirDec + 3;
            dirHexa = "${iDirDec:X}";
            linea.sDireccionHEXA = dirHexa;
        }
    }
    iPos++;
}

iPos = 1;
foreach (var linea in this.ensamblador.programa)
{

    this.dataGridView1.Rows.Add(linea.sDireccionHEXA, linea.sEtiqueta, linea.sCodigoOp, linea.sDireccion);
    iPos++;
}
}

```

EJECUCION DEL PROGRAMA



REPORTE PRACTICA NO 3

Ensamblador de SIC
— □ ×

Archivo

Programa

```

SUM  START 4000H
FIRST LDX  ZERO
      LDA  ZERO
LOOP  ADD  TABLE, X
      TIX  COUNT
      JLT  LOOP
      STA  TOTAL
      RSUB
TABLE  RESW 2000
COUNT RESW 1
ZERO   WORD 0
TOTAL  RESW 1
      END  FIRST
          
```

Direcciones

	CP	SIMBOLO	Instruccion / Directiva	Direccion
▶	4000H	SUM	START	4000H
	4003	FIRST	LDX	ZERO
	4006		LDA	ZERO
	4009	LOOP	ADD	TABLE,X
	400C		TIX	COUNT
	400F		JLT	LOOP
	4012		STA	TOTAL
	4015		RSUB	
	5785	TABLE	RESW	2000
	5788	COUNT	RESW	1
	578B	ZERO	WORD	0
	578E	TOTAL	RESW	1
	5791		END	FIRST

Errores

Errores de direcciones:

El Programa No Contiene Errores de Etiquetas repetidas
 Errores de instrucciones y/o directivas
 Sin errores de instrucciones y/o directivas

Archivo de errores generado en:
 file:\C:\Users\dany_\Desktop\Ensamblador_SIC\Ensamblador_SIC\bin\Debug\ERRORES.txt

Aceptar

Analizar Programa

	CP	SIMBOLO	Instruccion / Directiva	Direccion
▶	4000H	SUM	START	4000H
	4003	FIRST	LDX	ZERO
	4006		LDA	ZERO
	4009	LOOP	ADD	TABLE,X
	400C		TIX	COUNT
	400F		JLT	LOOP
	4012		STA	TOTAL
	4015		RSUB	
	5785	TABLE	RESW	2000
	5788	COUNT	RESW	1
	578B	ZERO	WORD	0
	578E	TOTAL	RESW	1
	5791		END	FIRST