

Practica #3 Direccionamientos Indexados y Relativos

Objetivo

Complementar junto con la práctica 2 la generación de código máquina de las instrucciones con direccionamiento indexado y relativo, teniendo en cuenta los siguientes requerimientos.

Requerimientos.

- **Generación de Código**

- Indexado de 5 bits "IDX" (X,Y,SP,PC de -16 a 15)
 - Buscar en TABOP el valor del mnemónico.
 - Leer el código maquina correspondiente al mnemónico.
 - Calcular el código máquina faltante y mostrarlo en la columna "Código Máquina"
 - Ejemplo para encontrar el valor de "xb":
 - **LDAA -1,X -----> LDAA oprx0_xysp -----> A6 xb**
 - La siguiente tabla muestra la estructura de un IDX de 5 bits, donde "Postbyte Code" representa la estructura para la generación del código máquina faltante.

Postbyte Code	Source Code Syntax	rr = X, Y, SP, PC
rr0nnnnn	,r	X = 00, Y = 01, SP = 10, PC = 11
	n,r	n = -16 a 15
	-n,r	

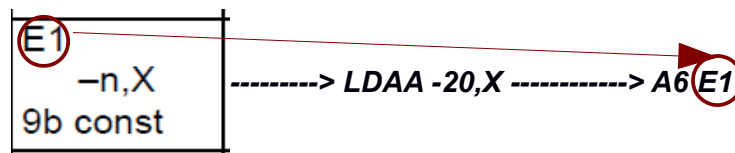
- En nuestro ejemplo **X = 00** y **n = -1**, el cual, *utilizando el complemento a 2 de 5 bits para convertir a negativo, n = 11111*.
- Una vez obtenidos los valores de "rr" y "n", suplirlos en el "postbyte code", quedando de la siguiente manera **00011111**, el cual en hexadecimal es **1F**.
- El resultado del código máquina será **A6 1F**.
- Lo anterior se puede comprobar de la siguiente manera:
 - Para encontrar el valor de "xb" hay que revisar la tabla 1 (*Indexed Addressing Mode Postbyte Encoding (xb)*) de la hoja de referencia del microcontrolador "HC12" (*CPU12 Reference Guide*).



- Indexado de 9 bits "IDX1" (X,Y,SP,PC de -256 a 255)
 - Buscar en TABOP el valor del mnemónico.
 - Leer el código máquina correspondiente al mnemónico.
 - Calcular el código máquina faltante y mostrarlo en la columna "Código Máquina".
 - Ejemplo para encontrar el valor de "xb ff":
 - **LDAA -20,X -----> LDAA oprx9,xysp -----> A6 xb ff**
 - La siguiente tabla muestra la estructura de un IDX1 de 9 bits, donde "Postbyte Code" representa la estructura para la generación del código máquina faltante.

Postbyte Code	Source Code Syntax	rr = X, Y, SP, PC
111rr0zs	n,r	X = 00, Y = 01, SP = 10, PC = 11
	-n,r	z = 0, s = (0 positivo, 1 negativo)

- En nuestro ejemplo **X = 00** y **n = -20**, el cual, está en el rango de los negativos, por lo tanto **s = 1**. Aquí "z" siempre es 0.
- Una vez obtenidos los valores de "rr" y "s", suplirlos en el "postbyte code", quedando de la siguiente manera **11100001**, el cual en hexadecimal es **1F**.
- El resultado del código máquina para "xb" es **E1**
 - Para encontrar el valor de "xb" hay que revisar la tabla 1 (*Indexed Addressing Mode Postbyte Encoding (xb)*) de la hoja de referencia del microcontrolador "HC12" (*CPU12 Reference Guide*).



- "ff" es el valor de "n", el cual es **-20**. Obtenemos el absoluto de "n" y se convierte a hexadecimal de 8 bits, teniendo así **14**.
- El resultado del código máquina es **A6 E1 14**.

- Indexado de 16 bits "IDX2" (X,Y,SP,PC de 0 a 65535)
 - Buscar en TABOP el valor del mnemónico.
 - Leer el código máquina correspondiente al mnemónico.
 - Calcular el código máquina faltante y mostrarlo en la columna "Código Máquina".
 - Ejemplo para encontrar el valor de "xb ee ff":
 - **LDAA 258,X -----> LDAA oprx16,xysp -----> A6 xb ee ff**
 - La siguiente tabla muestra la estructura de un IDX2 de 16 bits, donde "Postbyte Code" representa la estructura para la generación del código máquina faltante.

Postbyte Code	Source Code Syntax	rr = X, Y, SP, PC
111rr0zs	n,r	X = 00, Y = 01, SP = 10, PC = 11 z = 1, s = 0

- En nuestro ejemplo **X = 00** y **n = 258**. Aquí **z = 1** y “s” siempre es 0.
- Una vez obtenidos los valores de “rr”, suplirlos en el “postbyte code”, quedando de la siguiente manera **11100010**, el cual en hexadecimal es **1F**.
- El resultado del código máquina para “xb” es **E2**
 - Para encontrar el valor de “xb” hay que revisar la tabla 1 (*Indexed Addressing Mode Postbyte Encoding (xb)*) de la hoja de referencia del microcontrolador “HC12” (*CPU12 Reference Guide*).



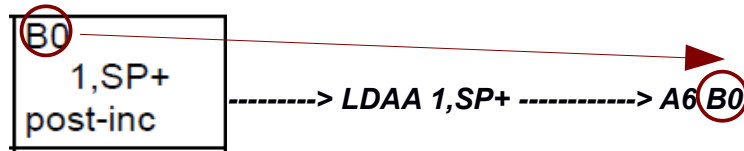
- “ee ff” es el valor de “n”, el cual es **258**, el cual se convierte a hexadecimal de 16 bits, dado **01 02**.
- El resultado del código máquina es **A6 E1 01 02**.

- Indexado PRE/POST INC/DEC “IDX” (X,Y,SP de 1 a 8 y de -1 a -8)
 - Buscar en TABOP el valor del mnemónico.
 - Leer el código maquina correspondiente al mnemónico.
 - Calcular el código máquina faltante y mostrarlo en la columna “Código Máquina”.
 - Ejemplo para encontrar el valor de “xb”:
 - **LDAA 1,SP+ -----> LDAA oprx0_xysp -----> A6 xb**
 - La siguiente tabla muestra la estructura de un IDX pre/pos, donde “Postbyte Code” representa la estructura para la generación del código máquina faltante.

Postbyte Code	Source Code Syntax	rr = X, Y, SP, PC
rr1pnnnn	n,-r	X = 00, Y = 01, SP = 10
p = 0 (pre)	n,+r	valores de “n”
p = 1 (pos)	n,r-	8=0111 -1=1111
	n, r+	7=0110 -2=1110
		6=0101 -3=1101
		5=0100 -4=1100

		4=0011 -5=1011
		3=0010 -6=1010
		2=0001 -7=1001
		1=0000 -8=1000

- En nuestro ejemplo **SP+ = 10**, **p = 1** y **n = 1**, el cual según la tabla le corresponde a **0000**.
- Una vez obtenidos los valores de “rr”, suplirlos en el “postbyte code”, quedando de la siguiente manera **10110000**, el cual en hexadecimal es **B0**.
- El resultado del código máquina para “xb” es **B0**.
 - Para encontrar el valor de “xb” hay que revisar la tabla 1 (*Indexed Addressing Mode Postbyte Encoding (xb)*) de la hoja de referencia del microcontrolador “HC12” (*CPU12 Reference Guide*).



- Indexado de Acumulador “IDX” (A,B,D y X,Y,SP,PC)
 - Buscar en TABOP el valor del mnemónico.
 - Leer el código maquina correspondiente al mnemónico.
 - Calcular el código máquina faltante y mostrarlo en la columna “Código Máquina”.
 - Ejemplo para encontrar el valor de “xb”:
 - **LDA A,X -----> LDA oprx0_xysp -----> A6 xb**
 - La siguiente tabla muestra la estructura de un IDX pre/pos, donde “Postbyte Code” representa la estructura para la generación del código máquina faltante.

Postbyte Code	Source Code Syntax	rr = X, Y, SP, PC
111rr1aa	A,r	X = 00, Y = 01, SP = 10
	B,r	valores para aa:
	D,r	A = 00, B = 01, D = 10

- En nuestro ejemplo **A = 00** y **X = 00**
- Una vez obtenidos los valores de “rr” y “aa”, suplirlos en el “postbyte code”, quedando de la siguiente manera **11100100**, el cual en hexadecimal es **E4**.
- El resultado del código máquina para “xb” es **E4**.

- Para encontrar el valor de “xb” hay que revisar la tabla 1 (*Indexed Addressing Mode Postbyte Encoding (xb)*) de la hoja de referencia del microcontrolador “HC12” (*CPU12 Reference Guide*).



- Relativo de 8 y 16 bits
 - Buscar en TABOP el valor del mnemónico.
 - Leer el código máquina correspondiente al mnemónico.
 - Calcular el código máquina faltante y mostrarlo en la columna “Código Máquina”.
 - Ejemplo para encontrar el valor de “rr”:
 - **BRA 15 -----> BRA rel8 -----> 20 rr**
 - Se lee el valor de operando, y a este se le resta el valor del siguiente contador de localidades. Ejemplo: el contador de localidades está en “0000”, el siguiente valor de la localidad es “0002” (ya que la instrucción BRA requiere 2 bytes), entonces se hace la resta de $15 - 2 = 13$, convertido a hexadecimal queda como **0D**. Por lo tanto “rr” valdrá **0D**, siendo el resultado del código máquina de **20 0D**.
 - Si el operando fuera una etiqueta, se toma el valor de la “tabla de símbolos” y se le resta el valor del siguiente contador de localidades.

prueba.asm

```

Limpia      JSR      A,PC
              LBRA    3052
              SWI
              SBCB    Limpia
              DS.W    20
              JSR      -12,X
              ORG     26378
              ANDB    212,PC
              JSR      17073,Y
EORB        @271
LDAA        #%11101101
              BVC     $6659
              JSR     7,SP+
              END

```

Salida:

Contador	Código Máquina	Etiqueta	Instrucción	Operando
0000	15 FC	Limpia	JSR	A,PC
0002	18 20 0B E6		LBRA	3052
0006	3F		SWI	
0007	F2 00 00		SBCB	Limpia
000A			DS.W	20
0032	15 14		JSR	-12,X
0034			ORG	26378
670A	E4 F8 D4		ANDB	212,PC
670D	15 EA 42 B1		JSR	17073,Y
6711	D8 B9		EORB	@271
6713	86 ED		LDAA	# %111011 01
6715	28 42		BVC	\$6659
6717	15 B6		JSR	7,SP+
6719			END	