

Actividad de Participación ISA de la LC3

La ISA de la LC-3 tiene las siguientes características:

- Tamaño de palabra: 16 bits. Los bits se miden de derecha a izquierda, comenzando por el bit 0.
- Localidades de memoria: Existen 2^{16} localidades, cada una con una palabra. Se numeran desde el 0 hasta el 65,535 (x0000 a xFFFF en hexadecimal).

Instrucciones

- Miden 16 bits. Se compone de dos cosas: opcodes, que es lo que la instrucción le está pidiendo hacer a la computadora; y los operandos, son quienes la computadora espera que haga la instrucción. Los bits [15:12] especifican el opcode, los bits [11:0] especifican los operandos.
- La ISA de la LC-3 maneja 15 instrucciones diferentes, cada una identificada por un opcode único.
- Los operandos de la LC-3 tienen soportan cinco modos de direccionamiento: inmediato (o literal), registro, relativo al PC, indirecto y Base+desplazamiento.
- La LC-3 soporta códigos de condición: tres registros de 1 bit. N(negativo), Z(cero) y P(positivo). Se establecen con base en el último resultado de la siguiente manera: negativo (N=1; Z, P=0), cero (Z=1, N, P=0) o positivo (P=1; N, Z=0).
- Los tipos de datos que la LC-3 soporta son: enteros en complemento a dos.
- Hay tres tipos diferentes de instrucciones: de operación, movimiento de datos y control.

Instrucciones de Operación

- Las instrucciones de operación procesan datos. La LC-3 tiene tres instrucciones de operación: ADD (opcode=0001), AND (opcode=0101) y NOT (opcode=1001)
- La instrucción NOT ejecuta una operación unaria, requiere un operando de origen. Las instrucciones AND y ADD realizan operaciones binarias, requieren dos operandos de origen.

Instrucciones de Movimientos de Datos

- Transfieren información entre los registros de propósito general y la memoria, y entre los registros y los dispositivos de entrada/salida.
- El proceso de transferir información de la memoria a un registro se llama carga (load), y el proceso de transferir información de un registro a la memoria se llama almacenamiento (store).
- Las instrucciones de movimientos de datos requieren dos operandos, uno de origen y uno de destino.

- Los bits [8:0] contienen los bits de generación de direcciones. Hay cuatro formas de interpretarlos, llamados modos de direccionamiento. Estos son:
 - Modo relativo a PC: LD (opcode=0010) y ST (opcode=0011). Los bits [8:0] especifican un desplazamiento relativo al PC. La dirección de memoria se calcula mediante la extensión de signo de los bits [8:0] a 16 bits y sumar el resultado al PC incrementado.
 - Modo indirecto: LDI (opcode=1010) y STI (opcode=1011). Los bits [8:0] especifican la dirección del operando que se va a cargar o almacenar.
 - Modo Base+desplazamiento: LDR (opcode=0110) y STR (opcode=0111). La dirección del operando se obtiene al sumar un desplazamiento de 6 bits de signo extendido a un registro base. El desplazamiento de 6 bits se toma de la instrucción en los bits [5:0]. El registro base se especifica mediante los bits [8:6]
 - Modo inmediato: Se utiliza sólo con la instrucción de dirección efectiva de carga: LEA (opcode =1110)

Instrucciones de Control

- Cambian la secuencia de las instrucciones que se están ejecutando.
- La LC-3 tiene cinco opcodes que permiten que los flujo secuenciales se rompan: bifurcación condicional, salto condicional, llamada de subrutina, TRAP y retorno de la interrupción.
- La instrucción TRAP es particularmente útil porque permite que un programador meta y saque información de la computadora sin comprender completamente las complejidades de dispositivos de entrada y salida.
- La instrucción JMP (opcode=1100) carga el PC con el contenido del registro especificado por los bits [8:6] de la instrucción.

Ejemplo “Hello World”

```

;-----
; Programa Hello World
; Autor: Martha L. Naranjo
; Fecha: Julio 8 de 2,009
;-----

.ORIG X3000
LEA R1, HELLO      ; R1 apunta a la cadena HELLO
LDR R0, R1, #0     ; R0 contiene el caracter apuntado por R1
AGAINOUT
ADD R1, R1, #1     ; se mueve el apuntador una posicion
LDR R0, R1, #0     ; R0 contiene el caracter apuntado por R1
BRnp AGAIN        ; el caracter es "zero" cuando llega al fin de cadena
HALT
HELLO.STRINGZ     "Hello World"
.END

```

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD ⁺	0001			DR			SR1			0	00		SR2			
ADD ⁺	0001			DR			SR1			1	imm5					
AND ⁺	0101			DR			SR1			0	00		SR2			
AND ⁺	0101			DR			SR1			1	imm5					
BR	0000			n	z	p	PCoffset9									
JMP	1100			000			BaseR			000000						
JSR	0100			1	PCoffset11											
JSRR	0100			0	00		BaseR			000000						
LD ⁺	0010			DR			PCoffset9									
LDI ⁺	1010			DR			PCoffset9									
LDR ⁺	0110			DR			BaseR			offset6						
LEA ⁺	1110			DR			PCoffset9									
NOT ⁺	1001			DR			SR			111111						
RET	1100			000			111			000000						
RTI	1000			000000000000												
ST	0011			SR			PCoffset9									
STI	1011			SR			PCoffset9									
STR	0111			SR			BaseR			offset6						
TRAP	1111			0000			trapvect8									
reserved	1101															