

Microprogramación

Ing. Ronald Caravaca Mora

IF4000 - Arquitectura de Computadores
Informática Empresarial

6 de noviembre de 2021



UNIVERSIDAD DE
COSTA RICA

- 1 Arquitectura de un Microprocesador
- 2 Microarquitectura
- 3 Registros
- 4 Modelo de programación

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡

Arquitectura de un Microprocesador

La memoria guarda dos tipos de información: las **instrucciones**, que son el tipo de operaciones que el computador ejecuta y los **datos** que son los valores numéricos necesarios para ejecutar las instrucciones. Algunas arquitecturas almacenan toda la información, instrucciones y datos, en una sola memoria, arquitectura es conocida como **von Newmann**. Otras arquitecturas almacenan las instrucciones en una memoria separada de los datos y esta arquitectura es conocida como **Hardvard**.

Arquitectura de un Microprocesador

Los periféricos se utilizan para transferir información desde y hacia la CPU. Los periféricos son, por ejemplo, el teclado, el mouse, la impresora, el monitor, el micrófono, la cámara, etc. De manera mas específica los periféricos son los componentes electrónicos que permiten a la CPU comunicarse con los dispositivos externos.

Arquitectura de un Microprocesador

Los buses son las rutas de conexión de las diferentes partes de un computador. En este sentido los buses son las conexiones físicas entre la CPU, la memoria y los dispositivos periféricos. Existen tres tipos de buses principales:

- 1 Bus de datos
- 2 Bus de direcciones
- 3 Bus de control

- 1 Arquitectura de un Microprocesador
- 2 Microarquitectura**
- 3 Registros
- 4 Modelo de programación

Microarquitectura

Una microarquitectura es la lógica digital que permite ejecutar un conjunto de instrucciones. Es la implementación combinada de registros, memoria, unidades aritméticas lógicas, buses y cualquier otro bloque lógico digital. Todo esto, en conjunto, forma el procesador.

Una microarquitectura combinada con un conjunto de instrucciones (ISA) y el modelo de programación, constituye la arquitectura del computador en su conjunto.

1 Arquitectura de un Microprocesador

2 Microarquitectura

3 Registros

Contador de programa

Acumuladores

Registro de status

Puntero de pila

4 Modelo de programación

1 Arquitectura de un Microprocesador

2 Microarquitectura

3 Registros

Contador de programa

Acumuladores

Registro de status

Puntero de pila

4 Modelo de programación

Contador de programa

Este registro sigue la ejecución del programa. Al final de la ejecución de cada instrucción, el contador de programa PC (del inglés "Program Counter"), contiene la dirección de la localización de memoria que almacena la próxima instrucción del programa. El contador de programa se va incrementando durante la ejecución de la instrucción. Finalmente, el PC contiene la dirección que apunta a la siguiente instrucción a ejecutarse. En algunos casos, el contenido de PC puede ser cambiado por el mismo programa. En esta forma se transfiere la ejecución a otra sección del programa.

1 Arquitectura de un Microprocesador

2 Microarquitectura

3 Registros

Contador de programa

Acumuladores

Registro de status

Puntero de pila

4 Modelo de programación

Acumuladores

Los acumuladores, son los registros principales para la manipulación de datos. Estos registros tiene la función de almacenar el resultado de operaciones lógicas y aritméticas, calculadas en la ALU, que involucran uno o dos operandos a la vez. En muchas de las arquitecturas, uno de los operandos proviene de un acumulador y el segundo de la localización de memoria especificada por la instrucción. El resultado de la operación se mantiene usualmente en un acumulador. El computador puede probar el resultado en el acumulador para determinar si contiene alguna condición especial, por ejemplo si es cero, negativo, positivo etc. y tomar diferentes decisiones basado en el resultado de la prueba.

1 Arquitectura de un Microprocesador

2 Microarquitectura

3 Registros

Contador de programa

Acumuladores

Registro de status

Puntero de pila

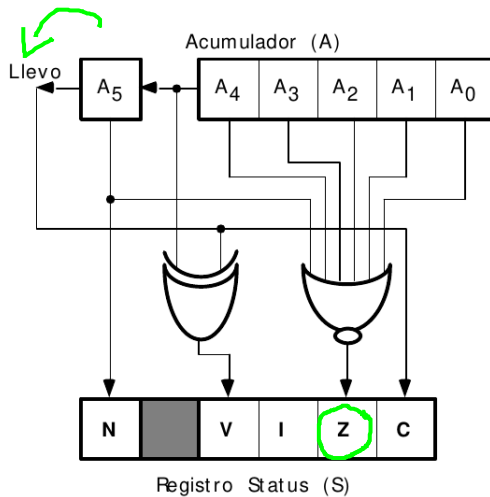
4 Modelo de programación

Registro de status

Este registro tiene la función de almacenar informaciones importantes para la programación, generalmente referentes al contenido de un acumulador o al estado de la CPU.

Por ejemplo, se puede tener un registro de status S, formado por cinco flip flops, cuatro de los cuales retienen información acerca del resultado de la última operación con el acumulador A, y el restante se utiliza para memorizar una acción de control pertinente a la CPU. Los primeros cuatro son, los flip flops de: signo, N; rebase, V; cero, Z y acarreo, C. El último se llama flip-flop de inhibición de interrupción I.

Registro de status



1 Arquitectura de un Microprocesador

2 Microarquitectura

3 Registros

Contador de programa

Acumuladores

Registro de status

Puntero de pila

4 Modelo de programación

Puntero de pila

Es el encargado de direccionar una zona especial de la memoria denominada pila. La pila es un conjunto bien definido de posiciones de memoria, en la cual los datos entran (se apilan) y salen (se desapilan), siguiendo la regla EL ULTIMO QUE ENTRA ES EL PRIMERO QUE SALE. En inglés se le conoce como "stack" la regla es "Last In First Out." LIFO. Su función primordial es la de servir como medio para almacenar la dirección de retorno después de la ejecución de una subrutina.

1 Arquitectura de un Microprocesador

2 Microarquitectura

3 Registros

4 Modelo de programación

Codificación

Clasificación Mnemotécnica

Clasificación de las instrucciones

Modos de direccionamiento

Modelo de programación

El programa de un computador, está compuesto de un número dado de instrucciones elementales ordenadas en una secuencia determinada por el programador. Cada instrucción elemental o básica, realiza una operación tal como SUME, MULTIPLIQUE o ALMACENE.

Para hacer programas con instrucciones elementales no es necesario conocer en detalle la estructura interna de la máquina. Sin embargo, es necesario conocer su organización lógica básica y distinguir las partes de la máquina que son accesibles al programador.

A las partes de la máquina que están directamente ligadas a la programación se les denomina modelo de programación. Estas son: memoria, contador de programa, acumulador, registro de status y puntero de pila.

Ejemplo de un programa

Se desea hacer un programa sencillo para sumar los contenidos de las localizaciones número 300, 301 y 302. Las instrucciones que se utilizan para realizar este programa son las siguientes:

- HLT: Esta instrucción hace que se detenga la ejecución del programa.
- ADD XXXX: Esta instrucción suma el contenido de la localización de memoria con dirección XXXX, al contenido del acumulador A. El resultado queda en el acumulador.
- CLA: Esta instrucción pone en cero al contenido del acumulador.

Ejemplo de un programa

Localización	Contenido
200	CLA
201-203	ADD 300
204-206	ADD 301
207-209	ADD 302
210	HLT
...	...
300	15
301	22
302	08

1 Arquitectura de un Microprocesador

2 Microarquitectura

3 Registros

4 Modelo de programación

Codificación

Clasificación Mnemotécnica

Clasificación de las instrucciones

Modos de direccionamiento

Codificación

En un computador digital, todas las operaciones se realizan en el sistema de números binarios. Sin embargo, durante el procedimiento de escritura de un programa, las instrucciones se pueden expresar en una forma que sea más manejable para el programador. A este cambio de representación de las instrucciones, se le llama CODIFICACION. En general, las instrucciones se pueden codificar en cuatro formas distintas, a saber: BINARIA, OCTAL, HEXADECIMAL, MNEMONICA.

Codificación binaria

Localización	Contenido
%000011001000	%000101
%000011001001	%010001
%000011001010	%101100
%000011001011	%000100
%000011001100	%010001
%000011001101	%101101

Codificación octal

Localización	Contenido
@0310	@05
@0311	@21
@0312	@54
@0313	@04
@0314	@21
@0315	@55
@0316	@04

Codificación hexadecimal

Localización	Contenido
\$0C8	\$05
\$0C9	\$11
\$0CA	\$2C
\$0CB	\$04
\$0CC	\$11
\$0CD	\$2D
\$0CE	\$04

1 Arquitectura de un Microprocesador

2 Microarquitectura

3 Registros

4 Modelo de programación

Codificación

Clasificación Mnemotécnica

Clasificación de las instrucciones

Modos de direccionamiento

Codificación Mnemotécnica

Codificación	Operación	Dirección
Binaria	%010001	%101000000111
Octal	@21	@5007
Hexadecimal	\$11	\$A07
Mnemónica	ADD	DATO

DATO se define igual a \$A07 en otra sección del programa.

Codificación Mnemotécnica

La codificación simbólica es muy conveniente para el programador, pero no puede ser entendida por el computador. El único lenguaje que puede entender el computador es la codificación binaria. Por esto es necesario hacer la traducción al equivalente binario, en una de las dos formas siguientes:

- Manualmente, empleando tablas con el mnemotécnico y el código binario equivalente.
- Mediante un programa de computador especial llamado ENSAMBLADOR (“assembler”).

Codificación Mnemotécnica

A un programa especificado así, se le llama **PROGRAMA FUENTE**. El ensamblador compara cada instrucción mnemotécnica con una tabla de instrucciones y las reemplaza con el código binario equivalente. También determinará la dirección correspondiente a cada etiqueta mediante un contador propio del ensamblador, que se inicia en la primera posición del programa traducido. A este proceso de traducción se le llama **ENSAMBLAJE DEL PROGRAMA** y el resultado de este proceso se denomina **PROGRAMA OBJETO**.

1 Arquitectura de un Microprocesador

2 Microarquitectura

3 Registros

4 Modelo de programación

Codificación

Clasificación Mnemotécnica

Clasificación de las instrucciones

Modos de direccionamiento

Clasificación de las instrucciones

Los computadores contienen una gran variedad de instrucciones determinadas según el diseño de la CPU. Para facilitar el estudio de las mismas, generalmente se clasifican de diferentes formas. Una manera de clasificar las instrucciones se basa en la parte del computador a la que se refiere la instrucción. Empleando este procedimiento se obtienen los siguientes tipos:

- 1 Instrucciones de Referencia a Memoria.
- 2 Instrucciones de Referencia a Registros.
- 3 Instrucciones de Referencia a E/S.

Clasificación de las instrucciones

Otro modo de clasificar las instrucciones se basa en la función que realiza cada instrucción:

- 1 Movimiento de datos, instrucciones lógicas y aritméticas.
- 2 Instrucciones de control.
- 3 Instrucciones de subrutina.
- 4 Instrucciones de referencia a registros.
- 5 Instrucciones de E/S.

Ejemplo de set de instrucciones

Tipo de Instrucción	Mnemo-técnico	Número de palabras	Operación realizada	Banderas alteradas
Movimiento de Datos, Aritméticas y Lógicas	LDA	2,3	A ? (M)	N - - Z -
	STA	3	M ? (A)	- - - -
	ADD	2,3	A ? (A) + (M)	N V - Z C
	SUB	2,3	A ? (A) - (M)	N V - Z C *
	AND	2,3	A ? (A) ? (M)	N - - Z -
	ORA	2,3	A ? (A) ? (M)	N - - Z -
	PHA	1	(A) ?	- - - -
Control	PLA	1	? A	N - - Z -
	HLT	1	Detenga la CPUCR ? DI	- - - -
	NOP	1	Operación nula ? DI+1	- - - -
	PHS	1	(S) ?	- - - -
	PLS	1	? S	N V I Z C
	RTI	1	? PC; I ? 0; ? (PC)	- - I -
	JMP	3	? M	- - - -
	BEQ	2	(Z)=1 ? M; (Z)=0 ? DI+2	- - - -
	BNE	2	(Z)=0 ? M; (Z)=1 ? DI+2	- - - -
	BCS	2	(C)=1 ? M; (C)=0 ? DI+2	- - - -
	BCC	2	(C)=0 ? M; (C)=1 ? DI+2	- - - -
	BPL	2	(N)=0 ? M; (N)=1 ? DI+2	- - - -
	BMI	2	(N)=1 ? M; (N)=0 ? DI+2	- - - -
	BVS	2	(V)=1 ? M; (V)=0 ? DI+2	- - - -
	BVC	2	(V)=0 ? M; (V)=1 ? DI+2	- - - -
	JSR	3	DI+3 ? ; ? M	- - - -
	RTS	1	? PC; ? (PC)	- - - -
Referencia a Registros	CLA	1	A ? 00	N - - Z -
	CPA	1	A ? (A)	N - - Z -
	INA	1	A ? (A) + 1	N V - Z C
	DCA	1	A ? (A) - 1	N V - Z C *
	ROL	1	A ? A4..A0,C; C ? A5	N - - Z C
	ROR	1	A ? C,A5..A1; C ? A0	N - - Z C
	TAP	1	P ? (A)	- - - -
	TPA	1	A ? (P)	N - - Z -
	SEC	1	C ? 1	- - - - C
	CLC	1	C ? 0	- - - - C
	SEI	1	I ? 1	- - I -
	CLI	1	I ? 0	- - I -
Entrada/Salida	INP	2	A ? (PUERTO)	N - - Z -
	OUT	2	PUERTO ? (A)	- - - -

DEBO=ACARREO Si C=1, no ha ocurrido un debo.

1 Arquitectura de un Microprocesador

2 Microarquitectura

3 Registros

4 Modelo de programación

Codificación

Clasificación Mnemotécnica

Clasificación de las instrucciones

Modos de direccionamiento

Modos de direccionamiento

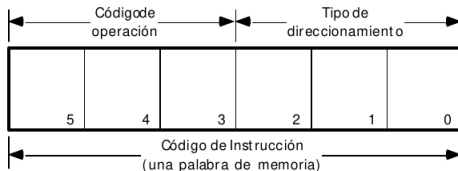
Se llaman modos de direccionamiento, a los diferentes métodos que tiene la CPU para obtener los operandos de la memoria principal.

La forma general usada hasta ahora en instrucciones de referencia a memoria es la siguiente:

<Código de instrucción> M

donde M es la dirección de memoria donde se encuentra el operando.

Modos de direccionamiento



BIT -> 210

- 000 - direccionamiento inmediato
- 001 - direccionamiento absoluto
- 010 - direccionamiento relativo
- 011 - direccionamiento indirecto
- 100 - direccionamiento implícito
- 101 - direccionamiento de acumulador

Direccionamiento inmediato




Las instrucciones que usan en este tipo de direccionamiento, son instrucciones de dos palabras. La primera, como en todos los modos de direccionamiento, es el código de instrucción y la segunda, es ya el operando. Es decir, no se requiere buscar en la memoria el operando, sino que viene incorporado como parte de la instrucción. Así por ejemplo,

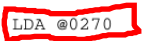
ADD #20

Este tipo de direccionamiento está diseñado específicamente para la operación con constantes, puesto que el programa generalmente no se altera durante su ejecución. Se utilizan para instrucciones aritméticas.

Direccionamiento absoluto

El direccionamiento absoluto permite direccionar cualquier localización de la memoria en forma directa. Las instrucciones con este modo de direccionamiento, son instrucciones de 3 palabras. La segunda y tercer palabra, forman el desplazamiento y definen una dirección efectiva de 12 bits para localizar el segundo operando. La segunda palabra es la parte menos significativa de la dirección y la siguiente aporta la parte más significativa.

Localización	Contenido	Acumulador
 @0100	LDA*	X
@0101	@70	X
@0102	@02	@23
...
 @0270		@23

* LDA = @01
Forma simbólica:  LDA @0270

Direccionamiento relativo

Las instrucciones que utilizan este modo de direccionamiento son únicamente las instrucciones de salto condicional y emplean dos palabras de memoria. La dirección relativa de salto, es un desplazamiento de una palabra, que contiene un número con signo expresado en complemento a dos. La dirección efectiva, la calcula la CPU sumando el desplazamiento al contenido del PC.

	Localización	Contenido	Forma Simbólica
	@0010	@25	INA
	@0011	@02	BEQ @03
	@0012	@03	
	@0013	@01	LDA @4201
	@0014	@01	
	@0015	@42	
	@0016	@32	BCC @71
	@0017	@71	

Nota: @71 = -7

111001

Direccionamiento indirecto

El desplazamiento de dos palabras en las instrucciones con direccionamiento indirecto, apunta a la parte baja de la dirección del operando; y no al operando, la parte alta se encuentra en la siguiente palabra. Por este motivo, es necesario hacer dos accesos a memoria en los que se busca la dirección efectiva, antes de proceder a la localización del operando. En la escritura de programas simbólicos, se encerrará la dirección entre paréntesis redondos para indicar el direccionamiento indirecto.

Direccionamiento Absoluto			Direccionamiento Indirecto		
Loc.	Cont.	Acum.	Loc.	Cont.	Acum.
@0030	@50	@00	@0030	@50	@00
@0031	@06	...	@0031	@06	...
...
@0100	@21 *	...	@0100	@23 **	...
@0101	@30	...	@0101	@30	...
@0102	@00	@50	@0102	@00	@17
...
@0650	@17	...	@0650	@17	...

Direccionamiento implícito

Este direccionamiento se refiere a instrucciones de referencia a registros y específicamente a las banderas de condición. La instrucción que emplea el direccionamiento implícito está formada de una sola palabra que contiene el código de instrucción. Se le llama direccionamiento implícito, porque la instrucción en si ya trae la dirección del operando y entonces no es necesario especificar un desplazamiento.

Direccionamiento de acumulador

Las instrucciones con este tipo de direccionamiento son de una sola palabra y se refieren a acciones sobre el acumulador. En cierto sentido, estas instrucciones tienen direccionamiento implícito, pero, ya que el acumulador es un registro tan importante, se le distingue con un direccionamiento especial.

Tipo de direccionamiento (D2-D0 octal)		Código de operación (D5-D3 octal)							
		0	1	2	3	4	5	6	7
Inmediato	0	LDA	•	ADD	SUB	AND	ORA	•	•
Absoluto	1	LDA	STA	ADD	SUB	AND	ORA	JMP	JSR
Relativo	2	BEQ	BNE	BCS	BCC	BMI	BPL	BVS	BVC
Indirecto	3	LDA	STA	ADD	SUB	AND	ORA	JMP	JSR
Implícito	4	•	•	SEC	CLC	SEI	CLI	•	•
Acumulador	5	CLA	CPA	INA	DCA	ROL	ROR	PLA	PHA
Control	6	TPA	TAP	RTI	RTS	HLT	NOP	PLS	PHS
Entrada/Salida	7	INP	OUT	•	•	•	•	•	•

EJEMPLO: La instrucción `ADD indirecta es @23`.

Decodificación de instrucciones

Este bloque recibe como entrada el contenido de RI para determinar la operación y el modo de direccionamiento de la instrucción. Su salida es leída por la unidad de control para iniciar los pasos necesarios en la ejecución de la instrucción.

- Lo ejecuta la unidad de control
- Realiza microoperaciones
- A través de microinstrucciones
- Microprograma

Decodificación de instrucciones

Para ejecutar una instrucción se efectúan un total de microoperaciones que en su conjunto se denominan el CICLO DE INSTRUCCION. Este ciclo se divide en tres estados llamados de BUSQUEDA, de DEMORA y de EJECUCION (fetch, decode, execute). A su vez, cada uno de tales estados se completa en uno o más CICLOS DE MÁQUINA, dependiendo del tipo de instrucción. El ciclo de máquina está formado por todas aquellas microoperaciones realizadas en la CPU entre dos accesos a memoria sucesivos. Cada acceso a memoria principal se le conoce como CICLO DE MEMORIA.

Diagrama de flujo de la unidad de control

Estructura de Computadores Digitales - RSF/MAVE - Agosto 1997

40

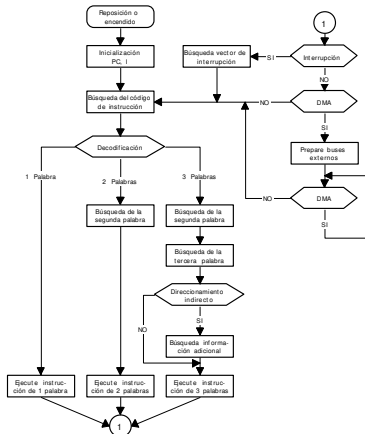


Figura 2.11