

2

INTRODUCIENDO AL MICROPROCESADOR 8085 Y MICROCONTROLADOR 8051

Debido al avance en el estado de arte en la tecnología semiconductor, fue posible para Intel el rediseñar el 8080 e incluir en un solo paquete los integrados 8224 y 8228. El resultado fue el microprocesador 8085. Además hay varias diferencias cuando se compara con el viejo 8080, la similitud más importante es que el 8085 es compatible en el código de objeto con el 8080. Esto significa que todo programa escrito para el 8080 también correrá sobre el 8085 sin ninguna alteración.

La familia de microcontroladores MCS-51, construida por Intel, se basa principalmente en tres circuitos integrados compatibles pin a pin. Estos son: 8031/8051/8751 los cuales, a su vez, son compatibles con el 8085. Estos tres chips se diferencian entre sí en la memoria de programa interna de la siguiente forma:

- El 8031 no tiene memoria interna de programa, por lo que le hará falta una memoria externa para la ejecución de un programa.
- El 8051 tiene 4 Kbytes de memoria ROM interna, programable en fábrica mediante máscara.
- El 8751 tiene 4 Kbytes de memoria EPROM interna, es decir programable y borrrable.

2.1 MÓDULO CPU DEL 8085 Y 8051

Uno de los problemas que los diseñadores del 8085 y 8051 (y desde luego, de todos los diseñadores de microprocesador y microcontroladores) tuvieron que encarar fue el limitado número de pines disponibles en el encapsulado estándar “*Dual-In-Line*” (DIP). La mayoría de los fabricantes decidieron el encapsulado de 40 pines como un buen compromiso entre un componente que puede ser manejado sin romperse y aún así proveer muchas líneas de acceso al mismo integrado.

Otra solución a este problema es la de compartir los pines del encapsulado para varias funciones. Esto fue lo que los diseñadores del 8085 y del 8051 escogieron. Mantuvieron el encapsulado estándar de 40 pines, pero combinaron las líneas de dirección de orden bajo (A0~A7) con las líneas del bus de dato (D0~D7). Las líneas resultantes son etiquetadas **AD0~AD7**.

Para que este esquema trabaje una nueva señal es necesaria para identificar cuando AD0~AD7 representa datos y cuando ellas representan la dirección de orden bajo. Intel llama a esta línea **ALE** por “*Address Latch Enable*”. Cuando **ALE** esta **alto**, AD0~AD7 lleva la dirección de orden bajo; cuando **ALE** conmuta a **bajo**, AD0~AD7 representa el bus de datos bidireccional.

COMPUTADORES DIGITALES

Introduciendo al Microprocesador 8085 y Microcontrolador 8051

La figura 2.1 provee la distribución de los pines y su descripción para el 8085.

SÍMBOLO	TIPO	NOMBRE Y FUNCIÓN																																												
$\overline{S0}, S1, \overline{IO/M}$	O	<p>Estado del Ciclo de Máquina</p> <table border="1"> <thead> <tr> <th>$\overline{IO/M}$</th> <th>S1</th> <th>S0</th> <th>ESTADO</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Escritura a Memoria</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Lectura a memoria</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Escritura a I/O</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Lectura a I/O</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Búsqueda del Cod-Op</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Búsqueda del Cod-Op</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Reconocimiento de la Interrupción</td> </tr> <tr> <td>•</td> <td>0</td> <td>0</td> <td>Halt</td> </tr> <tr> <td>•</td> <td>X</td> <td>X</td> <td>Hold</td> </tr> <tr> <td>•</td> <td>X</td> <td>X</td> <td>Reset</td> </tr> </tbody> </table> <p>•= tercer estado (alta impedancia) X= no especificado</p> <p>S1 puede ser usado como estado avanzado de $\overline{R/W}$. $\overline{IO/M}$, S0 y S1 se vuelven válidos al comienzo de un ciclo de máquina y permanecen estables a través del ciclo. El flanco descendente de ALE puede ser usado para fijar el estado de estas líneas.</p>	$\overline{IO/M}$	S1	S0	ESTADO	0	0	1	Escritura a Memoria	0	1	0	Lectura a memoria	1	0	1	Escritura a I/O	1	1	0	Lectura a I/O	0	1	1	Búsqueda del Cod-Op	1	1	1	Búsqueda del Cod-Op	1	1	1	Reconocimiento de la Interrupción	•	0	0	Halt	•	X	X	Hold	•	X	X	Reset
$\overline{IO/M}$	S1	S0	ESTADO																																											
0	0	1	Escritura a Memoria																																											
0	1	0	Lectura a memoria																																											
1	0	1	Escritura a I/O																																											
1	1	0	Lectura a I/O																																											
0	1	1	Búsqueda del Cod-Op																																											
1	1	1	Búsqueda del Cod-Op																																											
1	1	1	Reconocimiento de la Interrupción																																											
•	0	0	Halt																																											
•	X	X	Hold																																											
•	X	X	Reset																																											
\overline{RD}	O	Control de Lectura: Un nivel bajo sobre \overline{RD} indica que la memoria o dispositivo de I/O seleccionado será leído y que el bus de datos está disponible para la transferencia de dato, se encontrará en el tercer estado durante los modos Hold, Halt y durante RESET.																																												
\overline{WR}	O	Control de Escritura: Un nivel bajo sobre \overline{WR} indica que el dato sobre el bus de datos será escrito en la localidad de memoria o I/O seleccionado. El dato se configura en el flanco ascendente de \overline{WR} . Se encuentra en el tercer estado durante los modos Hold, Halt y durante RESET.																																												
A0-A15	O	Bus de Dirección: Los ocho bits más significativos de la dirección de memoria o los ocho bits de la dirección de I/O, los cuales se mantienen en el tercer estado durante los modos Hold, Halt y durante RESET.																																												
AD0-AD7	I/O	Bus Dirección/Dato Multiplexado: Los ocho bits más bajo de la dirección de memoria (o dirección de I/O) que aparece sobre el bus durante el primer ciclo de reloj (estado T) de un ciclo de máquina. Entonces se convierte en el bus de datos durante el segundo y tercer ciclo de reloj.																																												
ALE	O	Habilitador del Cerrojo de Dirección: Ocurre durante el primer reloj de estado de un ciclo de máquina y habilita la dirección para ser fijada dentro del cerrojo a bordo de los periféricos. El flanco descendente de ALE es iniciado para garantizar la configuración y los tiempos de sostenimiento para la información de la dirección. El flanco descendente de ALE puede también ser utilizado para centellear (strobe) la información del estado. ALE nunca se pone en el tercer estado.																																												
READY	I	Ready: Si READY está alto durante el ciclo de lectura o escritura, indica que la memoria o periférico está listo para enviar o recibir data. Si READY está bajo, la CPU esperará un número entero de ciclos de reloj para que READY vaya a alto antes de que se complete el ciclo de lectura o escritura. READY se debe conformar para especificar la configuración y tiempos de sostenimiento.																																												
HOLD	I	Hold: Indica que otro maestro hace la petición del uso de los buses de dirección y dato. La CPU, en el momento de recibir la petición de "hold", relegará el uso del bus tan pronto como se complete la transferencia actual en el bus. Los procesos internos pueden continuar. El procesador puede retomar el bus solamente después que se haya removido el "hold". Cuando el "hold" es reconocido, las líneas de dirección, dato, \overline{RD} , \overline{WR} e $\overline{IO/M}$ están en tercer estado.																																												
HLDA	O	Reconocimiento de Hold: Indica que la CPU ha recibido la petición de HOLD y que relegará el bus en el siguiente ciclo de reloj. HLDA va hacia abajo después que la petición de "Hold" es removida. La CPU toma el bus a la mitad del ciclo de reloj después que HLDA va hacia abajo.																																												
INTR	I	Petición de Interrupción: Es utilizado como interrupción de propósito general. Es verificado solamente durante el siguiente y el final del ciclo de reloj de una instrucción y durante los estados de Hold y Halt. Si está activo, el Contador de Programa (PC) se inhibirá del incremento y el \overline{INTA} se presentará. Durante este ciclo una instrucción de RESTART o																																												

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

		CALL puede ser insertada para saltar a la rutina de servicio de interrupciones. La INTR es habilitada o deshabilitada por software. Es deshabilitada por el Reset e inmediatamente después una interrupción es aceptada.
\overline{INTA}	O	Reconocimiento de la Interrupción: Es utilizado en vez de (y tiene la misma temporización como) \overline{RD} durante el ciclo de instrucción después que un INTR es aceptado. Puede ser utilizado para activar el Chip de Interrupción 8259A o cualquier otro puerto de interrupción.
RST 5.5 RST 6.5 RST 7.5	I	Interrupciones de Reinicio: Estas tres interrupciones tienen la misma temporización que la INTR excepto que causan que sea insertada automáticamente una Reinicialización interna. Las prioridades de estas interrupciones se ordenan de la siguiente forma: TRAP, RST7.5, RST6.5, RST5.5 e INTR. En adición, estos pueden ser individualmente enmascarados utilizando la instrucción SIM.
RESET OUT	O	Salida de Reset: "Reset Out" le indica a la CPU que será reinicializada. Puede ser utilizada como Reset del sistema. La señal es sincronizada al reloj del procesador y dura un número entero de períodos de reloj.
X_1, X_2	I	X_1 y X_2 : son conectados a un cristal, LC o red RC para manejar el generador de reloj interno. X_1 puede ser también una entrada de reloj externa desde una compuerta lógica. La frecuencia de entrada es dividida por 2 para darle al procesador una frecuencia de operación interna.
CLK	O	Reloj: Salida de reloj para ser usada como reloj de sistema. El período del reloj es el doble del período de entrada de X_1 y X_2 .
SID	I	Línea de Entrada de Data Serial: La data en esta línea es cargada en el bit 7 del acumulador cada vez que una instrucción RIM es ejecutada.
SOD	O	Línea de Salida de Data Serial: La salida de SOD es inicializada o reinicializada como es especificada por la instrucción SIM.
Vcc		Alimentación: Fuente de alimentación de +5V
Vss		Tierra: Referencia
TRAP	I	Trap: La interrupción trap es una interrupción de Reinicio no mascarable. Es reconocida en el mismo tiempo que INTR o RST5.5-7.5. No es afectada por cualquiera máscara o habilitación de interrupción. Tiene la más alta prioridad de cualquiera interrupción.
$\overline{RESET_IN}$	I	Reset In: Inicializa el Contador de Programa a cero y reinicializa la habilitación de la interrupción y los flip-flops de HLDA. Las líneas de control y los buses de dirección y datos están en tercer estado durante el RESET y debido a la naturaleza asíncrona del RESET, las banderas y los registros internos del procesador pueden ser alterados por el RESET con resultados impredecibles. El $\overline{RESET_IN}$ es una entrada de disparo-Schmitt, permitiendo una conexión a una red RC para un retardo de RESET de "power-on". Mientras se eleva la potencia (power-up), $\overline{RESET_IN}$ debe permanecer bajo por lo menos 10 ms después de alcanzar un Vcc mínimo. Para una apropiada operación de "reset" después de la duración de "power-up", $\overline{RESET_IN}$ deberá ser mantenida baja por un mínimo de tres períodos de reloj. La CPU es mantenida en la condición de "reset" todo el tiempo que $\overline{RESET_IN}$ sea aplicado.

Figura 2. 1 Definiciones y Números de Pin del Microprocesador 8085

¿Qué se ha ganado con este esquema de multiplexión? Varias interrupciones de entrada nuevas (entradas especiales al microprocesador que causan la suspensión de la tarea presente temporalmente) han sido provistas dándole al 8085 una considerable flexibilidad que el 8080 cuando da servicio a las interrupciones. En adición, dos nuevas líneas de I/O – SID y SOD – son provistas y permiten una rudimentaria capacidad de entrada y salida serial. Finalmente, información del estatus es proveída via los dos pines S0 y S1. Ellos codifican las actividades del bus como siguen:

COMPUTADORES DIGITALES

Introduciendo al Microprocesador 8085 y Microcontrolador 8051

S1	S0	
0	0	Pare
0	1	Escriba
1	0	Lea
1	1	Busque

Por la combinación de estas señales con IO/\overline{M} es posible identificar el tipo de ciclo de máquina en progreso. Debido a que S0 y S1 son presentados tempranamente en el ciclo de máquina, ellas pueden ser usadas para generar señales de control en adelanto de lectura y escritura. Esta técnica es útil para incrementar el tiempo de acceso a memoria permitida por el microprocesador.

La figura 2.2 provee la distribución de los pines y su descripción para el 8051.

SÍMBOLO	TIPO	NOMBRE Y FUNCIÓN
$\overline{V_{ss}}$		Tierra: Referencia del circuito
V_{cc}		Alimentación: Fuente de alimentación de +5V
PUERTO 0	I/O	Puerto 0: El puerto 0 es una puerta de entrada/salida bidireccional de drenador abierto de 8 bits. Además actúa como parte baja del bus de direcciones y bus de datos multiplexado (en combinación con la señal ALE) durante la lectura y escritura de la memoria externa. También se usa para programar y verificar la memoria interna del chip. Este puerto puede admitir/suministrar a 2 cargas TTL normales o a 8 cargas LSTTL.
PUERTO 1	I/O	Puerto 1: El puerto 1 es una puerta de entrada/salida cuasibidireccional de 8 bits. Se usa como parte baja (byte bajo) del bus de direcciones durante la programación y verificación de la memoria. Este puerto puede admitir/suministrar a 1 carga TTL normal.
PUERTO 2	I/O	Puerto 2: El puerto 2 es una puerta de entrada/salida cuasibidireccional de 8 bits. Se usa como parte alta (byte alto) del bus de direcciones para la lectura y escritura de la memoria externa. También se usa durante la programación y verificación de la memoria interna como parte alta del bus de direcciones. El puerto 2 puede admitir/suministrar a 1 carga TTL normal.
PUERTO 3	I/O	Puerto 3: El puerto 3 es una puerta de entrada/salida cuasibidireccional de 8 bits. Contiene además los pines de interrupción externa ($\overline{INT0}$ e $\overline{INT1}$), las entradas de los contadores ($T0$ y $T1$), los pines del puerto serie UART (RXD y TXD) y los pines de lectura/escritura de la memoria externa (RD y WR) usadas para diversas funciones. El "latch" de salida de una función debe activarse para que funcione dicha función. Este puerto puede admitir/suministrar a 1 carga TTL normal. Las funciones especiales asignadas al puerto 3 son: <ul style="list-style-type: none"> ➤ P3.0 $RXD/DATA$. Pin de entrada de datos serie en modo asíncrono, o pin de entrada/salida serie de datos en modo de registro de desplazamiento. ➤ P3.1 $TXD/CLOCK$. Pin de salida de datos serie en modo asíncrono, o salida de reloj en modo de registro de desplazamiento. ➤ P3.2 $\overline{INT0}$. Entrada de la interrupción 0 o entrada de control para el contador 0. ➤ P3.3 $\overline{INT1}$. Entrada de la interrupción 1 o entrada de control para el contador 1. ➤ P3.4 $T0$. Entrada del contador 0. ➤ P3.5 $T1$. Entrada del contador 1. ➤ P3.6 WR. Señal de control de escritura del bus de datos, desde el puerto 0 a la memoria externa de datos RAM. ➤ P3.7 RD. Señal de control de lectura del bus de datos, desde la memoria externa al puerto 0.
RST/VPD	I	Rst/Vpd: Un paso de "0" a "1" en este pin (aproximadamente 3 voltios), reinicializa el microcomputador. Una pequeña resistencia interna permite el "Power-On-Reset", usando sólo un condensador entre el pin y alimentación.
ALE/PROG	I/O	Ale/Prog: Esta salida activa el "latch" de enclavamiento de la parte baja del bus de direcciones, cuando se accede a la memoria externa. También se usa como entrada de pulsos en la programación de la EPROM interna.
\overline{PSEN}	O	PSEN: (Program Store Enable). Esta salida saca una señal de control, que habilita el bus

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

		de datos para la lectura de la memoria externa de programa EPROM/ROM.
\overline{EA} / VDD	I	\overline{EA} / VDD : Cuando \overline{EA} se mantiene a nivel alto (de TTL), el microcomputador ejecuta instrucciones desde la EPROM/ROM interna cuando el Contador de Programa (PC) está por debajo de 4096 (4Kbyte); si \overline{EA} se mantiene a nivel bajo (de TTL), el microcomputador busca todas las instrucciones en la memoria externa de programa EPROM/ROM. Este pin se utiliza también para recibir los 21 V necesarios para la programación de la EPROM interna.
XTAL 1	I	Xtal 1 : Es una entrada al amplificador de alta ganancia del oscilador. Se puede usar como fuente del oscilador un cristal de cuarzo. Cuando se utiliza un generador externo, este pin debe conectarse a tierra.
XTAL 2	O	Xtal 2 : Es una salida del amplificador del oscilador, y es necesario cuando se utiliza como fuente de oscilación un cristal de cuarzo. Se usa como entrada cuando se utiliza un generador externo.

Figura 2. 2 Definiciones y Números de pin del Microcontrolador 8051

2.2 MODELO DE PROGRAMACIÓN PARA EL MICROPROCESADOR 8085 Y MICROCONTROLADOR 8051

Un entendimiento del módulo CPU es esencial cuando su meta es el diseñar y construir un sistema particular de microcomputador. Sin embargo, cuando la meta es el desarrollar “software” para un procesador particular (esto es, programas de aplicación y control), un conocimiento del módulo CPU no es tan importante.

Por ejemplo, para escribir un programa en lenguaje ensamblado del 8085 no necesitamos saber como la señal de reloj es generada, o que la información del estatus se presenta al comienzo de cada ciclo de máquina. Lo que necesitamos es un conocimiento del trabajo interno del mismo microprocesador 8085. Esta información usualmente es presentada en la forma de “**Modelo de Programación**”.

El modelo de programación es un diagrama de los registros internos y banderas dentro de la CPU. Recuerde que ya hemos visto un registro especial en el Capítulo 1: el *Acumulador*. En general, el registro es una localidad de almacenamiento en la CPU. El acumulador es especial debido a que la data en este registro puede ser manipulado matemáticamente o lógicamente.

2.2.1 Un Modelo de Programación para el 8085 y el 8051.

La figura 2.3 es un modelo de programación para el microprocesador 8085. El 8085 cuenta con seis registros de 8 bit de propósito-general llamados registros B, C, D, E, H y L. El registro A es el acumulador y el registro F es un registro especial de bandera.

También tiene dos registros de 16 bits etiquetados registro SP y registro PC. El registro PC es el Contador de Programa (Program Counter) y sus 16 bits de largo son consistentes con el bus de direcciones de 16 bits del 8085. Como se mencionó en el Capítulo 1, el Contador de Programa mantiene la dirección de la siguiente instrucción que la CPU deberá buscar desde la memoria.

El registro SP es llamado Puntero de Pila (Stack Pointer). También mantiene una dirección de 16 bits, pero esta dirección es interpretada como apuntando hacia una sección de memoria especial llamada la Pila (Stack). El área de Pila de memoria puede ser usada para suministrar un

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

cercano número ilimitado de registros de propósito general externo al microprocesador. Es también usado para salvar la dirección de retorno cuando una instrucción de llamada a subrutina es dada.

A	F
B	C
D	E
H	L

STACK POINTER
PROGRAM COUNTER

Figura 2. 3 Modelo de Programación para el Microprocesador 8085

Recuerde que toda data para el 8085 es de 8 bits de largo. Por consiguiente, cuando la CPU busca una instrucción desde la memoria, utilizará una dirección de 16 bits, pero solo retornará 8 bits o un byte de data. Los registros de propósito general son para almacenar bytes de data, pero los registros “Stack Pointer” y “Program Counter” son usados para especificar localidades de memoria o direcciones.

Algunas de las instrucciones del 8085 permiten que los registros de propósito general se agrupen en *registros pares*. Entonces tenemos el “par BC”, el “par DE” y el “par HL”. Note que solo estos son permitidos. Cuando se utilizan de esta forma, el registro par puede representar números de 16 bits.

En la figura 2.4 se ilustra las cinco condiciones de bandera del 8085 y se provee la definición para cada una. Estas banderas son actualmente flip-flops de 1 bit que puede ser inicializada a un 1 lógico o reinicializada a un 0 lógico. En general, cada instrucción del 8085 afectará todas, algunas, o ningunas de estas banderas.

S	Z	-	AC	-	P	-	CY
----------	----------	----------	-----------	----------	----------	----------	-----------

Figura 2. 4 Registro de Banderas

- Signo (S): Si el bit más significativo del resultado de la operación tiene el valor 1, esta bandera se inicializa (1); de otra forma se reinicializa (0).
- Cero (Z): Si el resultado de una instrucción tiene el valor 0, esta bandera de inicializa (1); de otra forma se reinicializa (0).
- Acarreo Auxiliar: Si la instrucción causa un acarreo del bit 3 hacia el bit 4 del valor resultante, el acarreo auxiliar se inicializa (1); de otra forma se reinicializa (0). Esta bandera es afectada por de una sola precisión, sustracciones, incrementos, decrementos, comparaciones, y operaciones lógicas, pero es usado principalmente en adiciones e incrementos precedidos por una instrucción DAA (Ajuste a Decimal del Acumulador).

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

Paridad (P): Si la suma del módulo 2 de los bits del resultado de la operación es cero, (por ejemplo, si el resultado tiene paridad par), esta bandera se inicializa (1); de otra forma se reinicializa (0) (por ejemplo, si el resultado tiene paridad impar).

Acarreo (CY): Si la instrucción resulta en un acarreo (de una adición), o un préstamo (de una sustracción o comparación) del bit más significativo, esta bandera se inicializa (1); de otra forma se reinicializa (0).

La figura 2.5 es un modelo de programación para el microcontrolador 8051. Los registros de funciones especiales (SFR) incluyen:

- Registros Aritméticos (A, B, PSW).
- Punteros (SP, DPH, DPL).
- Registros de interface entre la CPU y funciones periféricas (SCON, SBUF...).

SIMBOLO	NOMBRE	DIRECCIÓN
*ACC	ACUMULADOR	E0H
*B	REGISTRO B	F0H
*PSW	PALABRA DE ESTADO DE PROGRAMA	D0H
SP	PUNTERO DE PILA	81H
DPTR	PUNTERO DE DATOS (2 BYTES)	
DPL	BYTE BAJO	82H
DPH	BYTE ALTO	83H
*P0	PUERTO 0	80H
*P1	PUERTO 1	90H
*P2	PUERTO 2	A0H
*P3	PUERTO 3	B0H
*IP	CONTROL DE PRIORIDAD DE INTERRUPCIONES	B8H
*IE	CONTROL DE HABILITACIÓN DE INTERRUPCIONES	A8H
TMOD	CONTROL DE MODOS DE TEMP./CONT.	89H
*TCON	CONTROL DE TEMPORIZADORES/CONTADORES	88H
TH0	BYTE ALTO DEL TEMP./CONT. 0	8CH
TL0	BYTE BAJO DEL TEMP./CONT. 0	8AH
TH1	BYTE ALTO DEL TEMP./CONT. 1	8DH
TL1	BYTE BAJO DEL TEMP./CONT. 1	8BH
*SCON	CONTROL DEL PUERTO SERIE	98H
SBUF	BUFFER DE TRANSMISIÓN DEL PUERTO SERIE	99H
PCON	CONTROL DE ALIMENTACIÓN	87H

* *Direccionable bit a bit*

Figura 2. 5 Modelo de Programación para el Microcontrolador 8051 (Registros de Funciones Especiales)

El registro A es el registro acumulador y su localización en la memoria interna RAM es ACC (Dirección E0H). El registro B está dedicado a servir de origen y destino de datos durante las

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

operaciones de multiplicación y división. En el resto de las instrucciones es un registro de propósito general (Dirección F0H).

El registro Puntero de Datos (DPTR) es un registro de 16 bits, es la concatenación de los registros DPH y DPL. El DPTR se usa en direccionamientos del tipo registro indirecto para:

- Mover constantes de la memoria de programa.
- Mover variables de la memoria externa de datos.
- Saltar a cualquiera de las direcciones de los 64 Kbytes de memoria de programa.

El registro Puntero de Pila (SP) es un registro de 8 bits la cual contiene la dirección en la que se depositó el último byte en la pila. Esta es también la dirección del primer byte que puede sacarse de la pila. Puede ser modificado por software.

La figura 2.6 muestra el Registro de Palabra de Estado de Programa (PSW), este registro nos indica la palabra de estado, es direccionable bit a bit y cada uno de ellos nos muestra el estado de una bandera (flag).

CY	AC	F0	RS1	RS0	OV	-	P
-----------	-----------	-----------	------------	------------	-----------	----------	----------

SIMBOLO	DIRECCIÓN	NOMBRE Y SIGNIFICADO
CY	D7H	Bandera de Acarreo, activado/desactivado por software o hardware durante ciertas instrucciones de aritmética/lógica.
AC	D6H	Bandera de Acarreo Auxiliar, activado/desactivado por hardware durante instrucciones de suma o resta, para indicar acarreo o transporte negativo (borrow out: dígito producido en sustracción directa cuando la diferencia entre los dígitos es menor que 0) del bit 3.
F0	D5H	Bandera 0. Bandera de estado definible por el usuario. Puede ponerse a 0 ó 1 o comprobar su estado por software.
RS1 RS0	D4H D3H	Control de selección de bando de registros (bits 1 y 0), activado/desactivado por software para determinar el banco de registros con el que se quiere trabajar*.
OV	D2H	Bandera de rebosamiento, activado/desactivado por hardware durante operaciones matemáticas, para indicar rebosamiento.
-	D1H	Reservado.
P	D0H	Bandera de paridad, activado/desactivado por hardware en cada instrucción para indicar un número par/impar de “unos” en el acumulador.

* Los contenidos de RS1 y RS0 seleccionan el bando de registros con el que se quiere trabajar. Hay que colocar el número del banco en binario, siendo el bit de mayor peso RS1 y el de menor peso RS0. En caso de no indicar nada, el sistema selecciona por defecto el 0.

Figura 2. 6 Registro de Palabra de Estado de Programa (PSW)

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

2.3 INTRODUCIENDO EL JUEGO DE INSTRUCCIONES DEL 8085 Y 8051

El juego de instrucciones de un microprocesador o microcontrolador representa los códigos únicos que la CPU puede interpretar, decodificar, y ejecutar. Recordando el modelo de CPU del Capítulo 1, estos códigos son buscados en la unidad de memoria durante el ciclo de máquina M1, almacenados en el registro de instrucción, decodificados por la CPU, y finalmente, ejecutados.

Los códigos de instrucción causan que la data se mueva entre los registros de propósito general, memoria, y dispositivos de I/O. Ellos permiten además que se desarrollen operaciones matemáticas y lógicas. El juego completo de todos los códigos de operación se le refiere como “**Juego de Instrucciones**”. Para el microprocesador 8085 son 244 códigos de operación únicos que el procesador puede reconocer, para el microcontrolador 8051 son 242.

Cuando se programa al microprocesador o microcontrolador es absolutamente necesario que el programador entienda el modelo de programación para ese procesador o controlador. En adición, se debe ser cuidadoso de todos los tipos de instrucción en el juego de instrucciones de ese procesador o controlador.

Tómese el tiempo de mirar en el Capítulo 1 los juegos de instrucciones. Incluidos ahí están los códigos de operación en hexadecimal, Información sobre los estados T y la cantidad de byte.

La tarea antes de usarla es el de familiarizarse con esta información para que podamos comenzar a escribir programas sencillos (y algunos no tan sencillos) en lenguaje de máquina y ensamblado.

2.3.1. Tipos de Instrucciones.

Considere los siguientes cinco grupos de instrucciones del 8085.

⇒ GRUPO DE TRANSFERENCIA DE DATOS

Las instrucciones en este grupo mueven la data de Registro a Registro, de Memoria a un Registro, o de Registro a Memoria. La data transferida puede ser de 8 o 16 bits (uno o dos bytes) de largo.

LXI RP,d16	STAX rp	MOV r,m
LDAX rp	SHLD Adr	MOV m,r
LHLD Adr	STA Adr	MVI r,d8
LDA Adr	MOV r1,r2	MVI m,d8

⇒ GRUPO ARITMÉTICO

Este grupo incluye todos los comandos matemáticos, tales como suma, resta, incremento y decremento. Nuevamente las operaciones de 8 o 16 bits son posibles.

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

ADD r	ADD m	ADC r	ADC m	ADI d8	ACI d8
SUB r	SUB m	SBB r	SBB m	SUI d8	SBI d8
INR r	INR m	INX rp	DCR r	DCR m	DCX rp
DAD rp					

⇒ **GRUPO LÓGICO Y ROTACION**

Este grupo incluye los operadores lógicos AND, OR, y XOR; comandos de rotación y comparación.

ANA r	XRA r	ORA r	CMP r
ANA m	XRA m	ORA m	CMP m
ANI d8	XRI d8	ORI d8	CPI d8
RLC	RRC	RAL	RAR

⇒ **GRUPO DE BIFURCACIÓN**

Las instrucciones de bifurcación son también referidas como comandos de salto. Este comando causa que el control del programa sea transferida a una nueva dirección rompiendo la secuencia normal de flujo del programa. La instrucción de salto puede ser CONDICIONAL O INCONDICIONAL.

JMP Adr	JC Adr	JM Adr	CZ Adr	CPE Adr	RNZ	RPO
JNZ Adr	JPO Adr	PCHL	CNC Adr	CP Adr	RZ	RPE
JZ Adr	JPE Adr	CALL Adr	CC Adr	CM Adr	RNC	RP
JNC Adr	JP Adr	CNZ Adr	CPO Adr	RET	RC	RM

⇒ **GRUPO DE PILA, ENTRADA/SALIDA, Y CONTROL DE MÁQUINA**

PUSH r	POP r	OUT d8	NOP	DI
PUSH PSW	POP PSW	IN d8	HLT	EI

El juego de instrucciones del 8051 se puede dividir según las especialidades:

- Instrucciones Aritméticas
- Instrucciones Lógicas
- Instrucciones de Transferencia de Datos
- Instrucciones Booleanas
- Instrucciones de Salto

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

⇒ **INSTRUCCIONES ARITMÉTICAS**

MNEMÓNICO	OPERACIÓN
ADD A,<byte>	$A = A + \text{<byte>}$
ADDC A,<byte>	$A = A + \text{<byte>} + C$
SUBB A,<byte>	$A = A - \text{<byte>} - C$
INC A	$A = A + 1$
INC <byte>	$\text{<byte>} = \text{<byte>} + 1$
INC DPTR	$DPTR = DPTR + 1$
DEC A	$A = A - 1$
DEC <byte>	$\text{<byte>} = \text{<byte>} - 1$
MUL AB	$B:A = B \times A$
DIV AB	$A = \text{int} [A/B]$ $B = \text{Mod} [A/B]$
DA A	Ajuste a Decimal

Al leer se observa que:

- Se puede incrementar o decrementar un byte en la memoria de datos interna sin el concurso del acumulador.
- Una de las instrucciones INC opera sobre los 16 bits del puntero de datos (DPTR).
- La instrucción MUL AB multiplica el contenido del acumulador con el dato situado en el registro B y el producto, en formato de 16 bits, aparece en los registros Acumulador y B (sobreescribe a los operandos).
- La instrucción DIV AB divide el contenido del acumulador con el dato situado en B, apareciendo el resultado, cociente, en el acumulador y el resto en el registro B.
- La instrucción DA A ajusta el resultado en binario puro producido por las instrucciones ADD y ADDC a formato BCD. Esta instrucción no convierte el contenido del acumulador en BCD, sino que hace el ajuste BCD de la suma (resultado), por lo tanto, esta instrucción debe ir inmediatamente después de la instrucción ADD o ADDC.

⇒ **INSTRUCCIONES LÓGICAS**

MNEMÓNICO	OPERACIÓN
ANL A,<byte>	$A = A \text{ .AND. } \text{<byte>}$
ANL <byte>,A	$\text{<byte>} = \text{<byte>} \text{ .AND. } A$
ANL <byte>,#data	$\text{<byte>} = \text{<byte>} \text{ .AND. } \text{\#data}$
ORL A,<byte>	$A = A \text{ .OR. } \text{<byte>}$
ORL <byte>,A	$\text{<byte>} = \text{<byte>} \text{ .OR. } A$
ORL <byte>,#data	$\text{<byte>} = \text{<byte>} \text{ .OR. } \text{\#data}$
XRL A,<byte>	$A = A \text{ .XOR. } \text{<byte>}$
XRL <byte>,A	$\text{<byte>} = \text{<byte>} \text{ .XOR. } A$
XRL <byte>,#data	$\text{<byte>} = \text{<byte>} \text{ .XOR. } \text{\#data}$
CLR A	$A = 00H$
CPL A	$A = \text{.NOT. } A$

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

RL A	Rote ACC a la izquierda 1 bit
RLC A	Rote a la izquierda a través del acarreo
RR A	Rote ACC a la derecha 1 bit
RRC A	Rote a la derecha a través del acarreo
SWAP A	Intercambie los nibbles en A

- Las primeras instrucciones corresponden a las operaciones AND, OR, EXOR y NOT.
- Continuando con el juego de instrucciones, aparecen instrucciones de manipulación de la información, como las instrucciones de Rotación y de Intercambio (SWAP).
- Las instrucciones de Rotación RL A y RR A desplazan el acumulador 1 bit a la izquierda y derecha, respectivamente, sin afectar al bit de acarreo.
- Las instrucciones de Rotación RLC A y RRC A desplazan el contenido del acumulador implicando al bit de acarreo.
- La instrucción SWAP A intercambia los nibbles en el acumulador. Generalmente se utiliza en las manipulaciones de números en BCD.

⇒ **INSTRUCCIONES DE TRANSFERENCIA DE DATOS**

➤ **Sobre la RAM Interna**

MNEMÓNICO	OPERACIÓN
MOV A,<src>	A = <src>
MOV <dest>,A	<dest> = A
MOV <dest>,<src>	<dest> = <src>
MOV DPTR,#data 16	DPTR = 16 bit CONSTANTE INMEDIATA
PUSH <src>	INC SP: MOV “@SP”,<src>
POP <dest>	MOV <dest>,”@SP”: DEC SP
XCH A,<byte>	ACC y <byte> INTERCAMBIAN DATA
XCHD A,@Ri	ACC y @Ri INTERCAMBIAN NIBBLE BAJO

- La instrucción MOV <dest>,<src> permite transferir datos entre alguna de las dos RAM interna o el espacio de localización del SFR. Este tipo de instrucción permite la transferencia sin el concurso del acumulador.
- La instrucción MOV DPTR,#data 16 bits, permite transferir datos de 16 bits para inicializar el DPTR (Data Pointer = puntero de datos) en el tratamiento de tablas en la memoria de programas o para acceder a los datos de la memoria externa.
- La instrucción PUSH <src> actúa incrementando el SP (Stack Pointer = puntero de la pila) y copiando el dato dentro de la pila. La instrucción POP <dest> actúa decrementando el SP y reponiendo el dato en su registro. La pila está situada dentro de la RAM interna, por defecto en el banco de registros 1, pero puede posicionarse en el área SCRATCH PAD.
- La instrucción XCH A,<byte>, intercambia los datos del acumulador y del byte direccionado. La instrucción XCHD A,@Ri es similar a la anterior, pero sólo implica en el intercambio el nibble bajo; es una instrucción especializada en la manipulación de datos en el código BCD.

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

➤ **Sobre la RAM Externa**

DIRECCIÓN CON	MNEMÓNICO	OPERACIÓN
8 bits	MOVX A,@Ri	LEA MEMORIA EXTERNA RAM @Ri
8 bits	MOVX @Ri,A	ESCRIBA MEMORIA EXTERNA RAM @Ri
16 bits	MOVX A,@DPTR	LEA MEMORIA EXTERNA RAM @DPTR
16 bits	MOVX @DPTR,A	ESCRIBA MEMORIA EXTERNA RAM @DPTR

Conviene observar que para este tipo de accesos siempre interviene el acumulador como registro fuente o destino.

El usuario debe escoger entre el tipo de instrucción MOVX A,@Ri o MOVX @Ri,A (siendo Ri, bien el registro R0 o R1 del bando de registros seleccionado), direccionando con 8 bits, o bien utilizar un direccionamiento de 16 bits, usando el registro DPTR (MOVX A,@DPTR o MOVX @DPTR,A) y tener que sacrificar el *puerto 2*, cuando sólo se va a utilizar un pequeño espacio de memoria RAM. Existe una solución intermedia que permite direccionar sólo unos pocos Kbytes de RAM externa, pero obteniendo algunas líneas hábiles del *puerto 2* como entradas y salidas (E/S). De todas formas la solución adecuada debe ser función de las características que necesita la aplicación, suponiendo, en principio, que la implementación de la misma sobre la elección de los Microcontroladores 8051 sea la correcta.

Las líneas de control de lectura y escritura, RD y WR (pines 16 y 17), sólo son utilizadas cuando se ejecutan las instrucciones MOVX; si no se va a utilizar memoria RAM externa, es obvio que se ganarán unas líneas extras de E/S.

➤ **Tratamiento de Tablas**

MNEMÓNICO	OPERACIÓN
MOVC A,@A + DPTR	LEA LA MEMORIA DE PROGRAMA EN (A + DPTR)
MOVC A,@A + PC	LEA LA MEMORIA DE PROGRAMA EN (A + PC)

La primera instrucción de la tabla permite la lectura en tablas con accesos no superiores a 256. El acceso deseado es cargado en el acumulador, y el “Data Pointer” (DPTR) establece el puntero de comienzo de la tabla. La lectura es cargada dentro del registro acumulador.

La segunda instrucción es similar a la primera, salvo que es el “Program Counter” (PC) el que actúa como indicador de la dirección base de la tabla.

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

⇒ **INSTRUCCIONES BOOLEANAS**

MNEMÓNICO	OPERACIÓN
ANL C,bit	C = C .AND. bit
ANL C,/bit	C = C .AND. .NOT. bit
ORL C,bit	C = C .OR. bit
ORL C,/bit	C= C .OR. .NOT. bit
MOV C,bit	C = bit
MOV bit,C	bit = C
CLR C	C = 0
CLR bit	bit = 0
SETB C	C = 1
SETB bit	bit = 1
CPL C	C = .NOT. C
CPL bit	bit = .NOT. bit
JC rel	SALTE SI C = 1
JNC rel	SALTE SI C = 0
JB bit,rel	SALTE SI bit = 1
JNB bit,rel	SALTE SI bit = 0
JBC bit,rel	SALTE SI bit = 1; CLR bit

⇒ **INSTRUCCIONES DE SALTO**

Un programa es una secuencia de instrucciones que el Contador de Programa (PC) rompe en función de:

- a) Haber concluido la secuencia de instrucciones y salta a otra o recomienza la misma. Este es un **Salto Incondicional**.
- b) La Palabra de Estado del Programa (PSW) o del estado de uno o más bits de la entrada/salida de periféricos. Este es un **Salto Condicional**.

➤ **Instrucciones de Salto Incondicional**

La tabla muestra la lista resumida de los saltos incondicionales. Aunque sólo figure la instrucción <<JMP addr>>, de hecho hay tres tipos: SJMP, LJMP y AJMP, que difieren en el formato de la dirección de salto.

MNEMÓNICO	OPERACIÓN
JMP addr	SALTE A LA addr
JMP @A + DPTR	SALTE A A + DPTR
CALL addr	LLAME A LA SUBRUTINA EN addr
RET	RETORNE DESDE LA SUBRUTINA
RETI	RETORNE DESDE LA INTERRUPCIÓN
NOP	No-OPERACIÓN

COMPUTADORES DIGITALES

Introduciendo al Microprocesador 8085 y Microcontrolador 8051

SJMP rel: (Short Jump). La dirección destino del salto viene dada por un <<offset relativo>>, igual que en las instrucciones de salto booleanas. La instrucción tiene dos bytes, correspondientes al código de operación y al byte de offset relativo. El rango del salto está limitado a -128 y +127 bytes relativos al primer byte siguiente a la instrucción de salto.

LJMP addr16: (Long Jump). La dirección destino del salto viene dada por una constante de 16 bits. La instrucción está formada por tres bytes. 1 byte para el código de operación y 2 para la dirección de salto. Por tanto, la dirección de destino está ubicada dentro del área de los 64K de la memoria de programa.

AJMP addr11: (Absolute Jump). La dirección destino del salto viene dada por una constante de 11 bits. La instrucción tiene 2 bytes, uno para el código de operación que también contiene en sí misma 3 de los 11 bits de direcciones, y otro byte que contiene los 8 bits bajos de la dirección de destino.

Por tanto, el destino del salto tiene que estar comprendido dentro del bloque de los 2 Kbytes referenciado respecto a la siguiente instrucción a la del salto.

JMP @A + DPTR: Se trata de una instrucción de salto indirecto. Suma el byte contenido en el acumulador con los 16 bits del puntero de datos (DPTR) y carga el resultado de la suma en el contador de programa (PC). Esta será la dirección para la subsiguiente búsqueda de la instrucción.

Siguiendo con las instrucciones incondicionales, de la tabla, ésta muestra la instrucción CALL addr que tiene 2 formatos, LCALL y ACALL, que difieren en la forma en la que la CPU implementa la dirección donde se encuentra la subrutina.

LCALL addr16: Utiliza un formato de 16 bits para direccionar la subrutina y esa área de direccionamiento es el máximo de 64 Kbytes de la memoria de programas.

ACALL addr11: Utiliza un formato de direccionamiento de 11 bits, por lo que la subrutina debe estar dentro del área de los 2 Kbytes, a partir de la siguiente instrucción.

➤ Instrucciones de Salto Condicional

La tabla muestra la lista de las instrucciones de salto condicional. Estas instrucciones son las que van a permitir al microcontrolador tomar decisiones (siguiendo secuencialmente una parte del programa u otra) ante una propuesta binaria, que en el lenguaje coloquial se contesta con una afirmación o negación. La propuesta de bifurcación de la secuencia del programa se halla implícita dentro del código de operación y es el operando el que señala la dirección del salto en el formato de <<offset relativo>>, por lo que distancia del salto está limitada a -128 y +127 bytes referida a la instrucción siguiente a la del salto condicional. Trabajando con el programa ensamblador es suficiente señalar con una etiqueta la dirección del salto o con una constante de 16 bits.

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

MNEMONICO	OPERACION
JZ rel	SALTA SI A = 0
JNZ rel	SALTA SI A <> 0
DJNZ <byte>,rel	DECREMENTA Y SALTA SI NO ES CERO
CJNE A,<byte>,rel	SALTA SI A <> <byte>
CJNE <byte>,#data,rel	SALTA SI <byte> <> #data

2.4 MODOS DE DIRECCIONAMIENTO

8085	MNEMONICO	OPERACION
DIRECTO REGISTRO REGISTRO INDIRECTO INMEDIATO		
Directo	LDA 1000H	A <--- (1000H)
Registro	MOV A,B	A <--- (B)
Registro Indirecto	MOV A,M	A <-- (HL)
Inmediato	MVI A,00	A <--- 00
8051	MNEMONICO	OPERACION
DIRECTO INDIRECTO REGISTRO IMPLÍCITO INMEDIATO INDEXADO		
Directo	ADD A,3B	A <--- (A) + (3B)
Indirecto	ADD A,@R0	A <--- (A) + (R0)
Registro	MOV R3,A	(R3) <-- (A)
Implícito	INC A	A <--- (A) +1
Inmediato	MOV A,#FFH	A <--- FFH
Indexado	MOVC A, @A + DPTR	A <-- ((A) + (DPTR))

COMPUTADORES DIGITALES

Introduciendo al Microprocesador 8085 y Microcontrolador 8051

➤ **Direccionamiento Directo**

El operando se especifica en la instrucción por un campo de dirección de 16 bits, para el 8085 y de 8 bits para el 8051. Continuando con el 8051 sólo la RAM interna de datos (primeros 128 bytes) y la zona de SFR (Registro de Funciones Especiales) se pueden direccionar de esta forma.

➤ **Direccionamiento Indirecto**

La instrucción especifica un registro par, para el 8085, que contiene la dirección del operando. Para el 8051 la instrucción especifica un registro que contiene la dirección del operando. Tanto la memoria RAM interna (256 bytes/128 bytes) como la externa se pueden direccionar indirectamente.

Los registros para direccionar sobre el mapa de 8 bits pueden ser el R0 y R1 del banco de registros seleccionados, o el "Stack Pointer". El registro para direccionar sobre 16 bits sólo puede ser el Puntero de Datos (Data Pointer = DPTR).

➤ **Direccionamiento por Registro**

Los microcontroladores 8052/8051 contiene cuatro bancos de registros seleccionados por los bits 3 y 4 del PSW, y cada banco de registros tiene ocho registros del R0 al R7. El propio código de operación de la instrucción especifica con qué registro se opera; es decir, cuando la instrucción es ejecutada se accede a uno de los 8 registros del banco seleccionado.

Para el 8085 el procedimiento se hace con cualquiera de los 7 registros disponibles (A, B, C, D, E, H, L).

➤ **Direccionamiento Inmediato**

Al código de operación le sigue una constante en la memoria de programas.

➤ **Direccionamiento Implícito**

Algunas instrucciones especifican, implícitamente, el registro sobre el que van a operar, como el acumulador, el puntero de datos, etc. No necesitan especificar el operando porque está implícito en el código de operación.

➤ **Direccionamiento Indexado**

Este direccionamiento sólo es posible en la memoria de programas y sólo permite la lectura. Es utilizado para la lectura de tablas. Un registro base de 16 bits (el DPTR o el Contador de Programa) apunta a la base de la tabla y el contenido del acumulador es el offset que permite acceder a la lectura de esa posición de la tabla. Es decir, la dirección

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

de la tabla que se va a acceder está formada por la suma del acumulador y el puntero base.

Se pueden utilizar otros tipos de direccionamiento indexado en los <<saltos>> de instrucción. En este caso la dirección de destino del salto se computa como la suma del puntero base y el dato del acumulador.

2.5 PONIENDO TODO JUNTO: UN EJEMPLO DE PROGRAMACIÓN

Ahora utilicemos nuestro nuevo conocimiento de los juegos de instrucciones para escribir un programa de microcomputador que adicione dos números; ¡no es exactamente “Space Invaders” pero es un comienzo!

8085

Ejemplo 2.1

Escriba un programa que adicione el contenido de las localidades de memoria 0700H y 0701H y deje la suma en la localidad de memoria 0702H. Escriba el programa iniciando en la localidad de memoria 0600H y muestre los mnemónicos.

Solución. La figura 2.7 es la solución con el 8085. Note como este programa ha sido presentado. Un formulario especial identifica el autor, nombre, y la data del programa e incluye seis campos para el programa mismo. Estos son el [campo de direcciones](#), el [campo de código objeto](#), el [campo de etiqueta](#), el [campo de código de operación](#), y el [campo de comentario](#).

Nombre del Programa _____							
Autor _____				Fecha _____			
Dirección	Cód. Objeto			Etiqueta	Cód-op	Operando	Comentarios
0600	3A	00	07	SUM:	LDA	0700H	;BUSQUE 1 ^{er} NUM.
0603	47				MOV	B,A	; GUARDE EN B
0604	3A	01	07		LDA	0701H	; BUSQUE 2 ^{do} NUM.
0607	80				ADD	B	; COMPUTE SUMA
0608	32	02	07		STA	0702H	; GUARDE SUMA
060B	76				HLT		; PARE

Figura 2.7 Solución de Programa del 8085 para el Ejemplo 2.1

COMPUTADORES DIGITALES

Introduciendo al Microprocesador 8085 y Microcontrolador 8051

Por el momento sólo nos importará la porción con el lenguaje ensamblado del programa e ignoraremos los campos de dirección y de código de objeto. La primera instrucción, LDA 0700H, está hecha del código de operación (Cód-op) LDA y del operando 0700H. Como indica el campo de comentarios, la instrucción es utilizada para buscar el primer número que se sumará (el número almacenado en 0700H).

La etiqueta SUM: es opcional y es utilizada aquí simplemente para darle a la rutina de adición un nombre.

Debido a que la adición debe ocurrir en el acumulador, la segunda instrucción transfiere el byte de dato al registro B para almacenamiento temporal. Ahora la instrucción LDA 0701H trae el segundo número y estamos preparados para la adición: ADD B (adicione los contenidos de los registros A y B). La suma, la cual se deja en el registro A, es almacenada en la localidad de memoria 0702H con la instrucción de direccionamiento directo: STA 0702H. HLT termina el programa.

A este punto la versión de lenguaje ensamblado del programa se completa. ¡Pero por supuesto el 8085 no puede ejecutar mnemónicos! Lo que hemos generado se llama *código fuente*, pero lo que el microprocesador quiere se llama *código objeto*. Tenemos ahora dos opciones. Si hay disponible un ensamblador del 8085, el código fuente puede ser escrito dentro del computador y el ensamblador le dice que ensamble el programa. Si no existe un ensamblador disponible, tendremos que “ensamblar a mano” el programa.

Ensamblar a mano es una buena manera de aprender la programación en lenguaje de máquina pero es tedioso para programas largos. Para ensamblar a mano este programa, refiérase al juego de instrucciones del 8085 (figura 1.7) la cual provee un listado de todas las instrucciones del 8085. Mire la instrucción LDA. Verá: LDA a, 3A a. Recuerde que los números de 16 bits son introducidos “al revés”, se introduce 3A 00 07 a la izquierda del Cód-op LDA bajo en campo etiquetado “Código Objeto”. Continuando de esta manera, se busca los códigos hex para cada instrucción. Finalmente, podemos completar el campo de direcciones, la cual debe comenzar en la dirección 0600H. Note como la dirección de orden baja se incrementa de instrucción a instrucción. No olvide que se está contando en hex, lo que significa que después de la dirección 0609 viene la dirección 060A (¡no 0610!).

El programa está listo para ser probado. Si se utiliza un entrenador de microprocesador, puede utilizar el programa monitor para introducir los códigos hex dentro de la memoria iniciando en la dirección 0600H. Antes de ejecutar el programa coloque dos números en las localidades 0700H y 0701H. Después de correr el programa, la suma correcta deberá estar almacenada en la dirección 0702H. Si tiene disponible el simulador también podrá probar el programa.

Ejemplo 2.2

Sume los contenidos de las localidades de memoria 0700H y 0701H. Coloque la suma en la localidad de memoria 0702H.

COMPUTADORES DIGITALES

Introduciendo al Microprocesador 8085 y Microcontrolador 8051

Discusión. Este problema fue resuelto en el ejemplo anterior usando direccionamiento directo. En este ejemplo utilizaremos direccionamiento indirecto de registro. Esta técnica utiliza el registro par HL como puntero a una localidad de memoria. El comando INR L puede ser usado para cambiar la posición del puntero.

```
2500 A.D. 8085 Macro Assembler - Version 4.05a
-----

Input Filename : ADICION.asm
Output Filename : ADICION.obj

1          ; ADICION DE 8 BIT PARA EL 8085
2          ;
3          ; ESTE PROGRAMA ADICIONARA LOS CONTENIDOS
4          ; DE LAS LOCALIDADES DE MEMORIA 0700H Y
5          ; 0701H Y ALMACENARA LA SUMA EN 0702H
6          ;
7 0600      .ORG    0600H      ; INICIE EN LA LOCALIDAD 0600H
8 0600      LXI    H,0700H    ; HL APUNTA HACIA EL PRIMER BYTE
9 0603      MOV    A,M        ; GUARDE EL PRIMER BYTE EN A
10 0604     INR   L          ; AHORA APUNTE AL SEGUNDO BYTE
11 0605     ADD   M          ; SUMELO A "A"
12 0606     INR   L          ; AVANCE EL PUNTERO
13 0607     MOV   M,A        ; ALMACENE LA SUMA AQUI
14 0608     HLT   ; PARE
15 0609     END

Lines Assembled : 15          Assembly Errors : 0

Figura 2. 8 Programa 2 Adición de 8 bit para el 8085
```

Nuevas Instrucciones:

ORG El comando ORG es una directriz del lenguaje ensamblado. Instruye al ensamblador de iniciar el ensamblado de los mnemónicos siguientes en la dirección especificada en el campo de operando (en este programa la dirección 0600H). Puede haber cualquier número de comandos ORG dentro de un programa particular.

LXI H,Addr El byte 3 de la instrucción es transferido dentro del registro H. El byte 2 de la instrucción es transferido dentro del registro L. Esto es una operación de cargar un registro par de 16 bits usando el modo de direccionamiento inmediato. Otras formas son LXI B, LXI D, y LXI SP.

INR L El contenido del registro L es incrementado en 1. Cualquiera de los registros de propósito general puede ser especificado. Similarmente, **DCR r** decrementa un registro.

MOV A,M El contenido de la localidad de memoria, cuya dirección está en los registros H y L, es transferido al registro A. Esta es una instrucción de un solo byte usando el modo de direccionamiento indirecto de registro. Reduce dos bytes y seis ciclos de reloj comparado con la técnica de direccionamiento directo (LDA Addr). Puede ser utilizado con cualquiera de los registros de propósito general (MOV B,M, MOV C,M, etc.).

COMPUTADORES DIGITALES

Introduciendo al Microprocesador 8085 y Microcontrolador 8051

MOV M,A El contenido del registro A es transferido a la localidad de memoria cuya dirección está en los registros H y L. Esta instrucción también utiliza el direccionamiento indirecto de registro. Puede ser utilizado con cualquiera de los registros de propósito general (MOV M,B, MOV M,C, etc.).

ADD M El contenido de la localidad de memoria cuya dirección está contenida en los registros H y L es sumado al contenido del acumulador. El resultado es colocado en el acumulador. Esta instrucción usa el direccionamiento indirecto de registro.

HLT El procesador es detenido. Los registros y las banderas no son afectados. Usted querrá reemplazar esta instrucción con RET o JMP de regreso al monitor. Sino, solo un “reset” reinicializará el computador.

END END es otra directriz del ensamblador (opcional) y debe ser la última declaración en el programa de lenguaje ensamblado. Todas las declaraciones después de END serán ignoradas por el ensamblador.

Solución de Programa. La figura 2.8 provee una solución para este ejemplo. El programa inicia cargando el registro par HL con la dirección del primer byte. Entonces el comando MOV A,M trae este byte al acumulador. Incrementando L resulta que HL apunta al segundo byte. ADD M adiciona esto al acumulador. Un segundo INR L causa que HL apunte a la localidad de almacenamiento de la suma y MOV M,A realiza esto. Las declaraciones que inicien con punto y como (;) son comentarios y son ignorados por el ensamblador.

Resumen. Este programa resuelve el problema de adición presentado en el ejemplo 2.1 primeramente, con siete instrucciones versus seis para esa solución. Sin embargo, el número total de bytes ha sido reducido desde 12 a 9 debido al uso del direccionamiento indirecto de registro.

8051

Para los ejemplos de programa con el microcontrolador 8051, se utilizarán los mismos ejemplos y modos de direccionamiento que se vieron en los ejemplos 2.1 y 2.2 del 8085, tomando en cuenta que el 8051 tiene memoria interna.

Ejemplo 2.3

Escriba un programa que adicione el contenido de las localidades de memoria 0008H y 0009H y deje la suma en la localidad de memoria 000AH. Escriba el programa iniciando en la localidad de memoria 0050H y muestre los mnemónicos.

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

```
MCS-51 MACRO ASSEMBLER      ADD1          08/11/98    PAGE    1

DOS 7.10 (038-N) MCS-51 MACRO ASSEMBLER, V2.3
OBJECT MODULE PLACED IN ADD1.OBJ
ASSEMBLER INVOKED BY:  C:\ASEMBLER\APBLDR\ASM51.EXE ADD1.TXT

LOC  OBJ          LINE      SOURCE
0000                1      ORG 0000H
0000 020050        2      LJMP 0050H
0050                3      ORG 0050H
0050 E508          4      MOV A,08H
0052 F8            5      MOV R0,A
0053 E509          6      MOV A,09H
0055 28            7      ADD A,R0
0056 F50A          8      MOV 0AH,A
0058 00            9      NOP
                10     END

MCS-51 MACRO ASSEMBLER      ADD1          08/11/98    PAGE    2

SYMBOL TABLE LISTING
-----

NAME      TYPE  VALUE      ATTRIBUTES

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

Figura 2. 9 Problema 3 adición de 8 bit para el 8051

COMPUTADORES DIGITALES
Introduciendo al Microprocesador 8085 y Microcontrolador 8051

Ejemplo 2.4

Sume los contenidos de las localidades de memoria 0008H y 0009H. Coloque la suma en la localidad de memoria 000AH.

```
MCS-51 MACRO ASSEMBLER      ADD2                08/11/98    PAGE    1

DOS 7.10 (038-N) MCS-51 MACRO ASSEMBLER, V2.3
OBJECT MODULE PLACED IN ADD2.OBJ
ASSEMBLER INVOKED BY:  C:\ASSEMBLER\APBLDR\ASM51.EXE ADD2.TXT

LOC  OBJ          LINE      SOURCE
0000                1      ORG 0000H
0000 020050        2      LJMP 0050H
0050                3      ORG 0050H
0050 7808          4      MOV R0,#08H
0052 E6            5      MOV A,@R0
0053 08            6      INC R0
0054 26            7      ADD A,@R0
0055 08            8      INC R0
0056 F6            9      MOV @R0,A
                   10     END

MCS-51 MACRO ASSEMBLER      ADD2                08/11/98    PAGE    2

SYMBOL TABLE LISTING
-----

N A M E      T Y P E  V A L U E      A T T R I B U T E S

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

Figura 2. 10 Solución de Programa del 8051 para el Ejemplo 2.4