

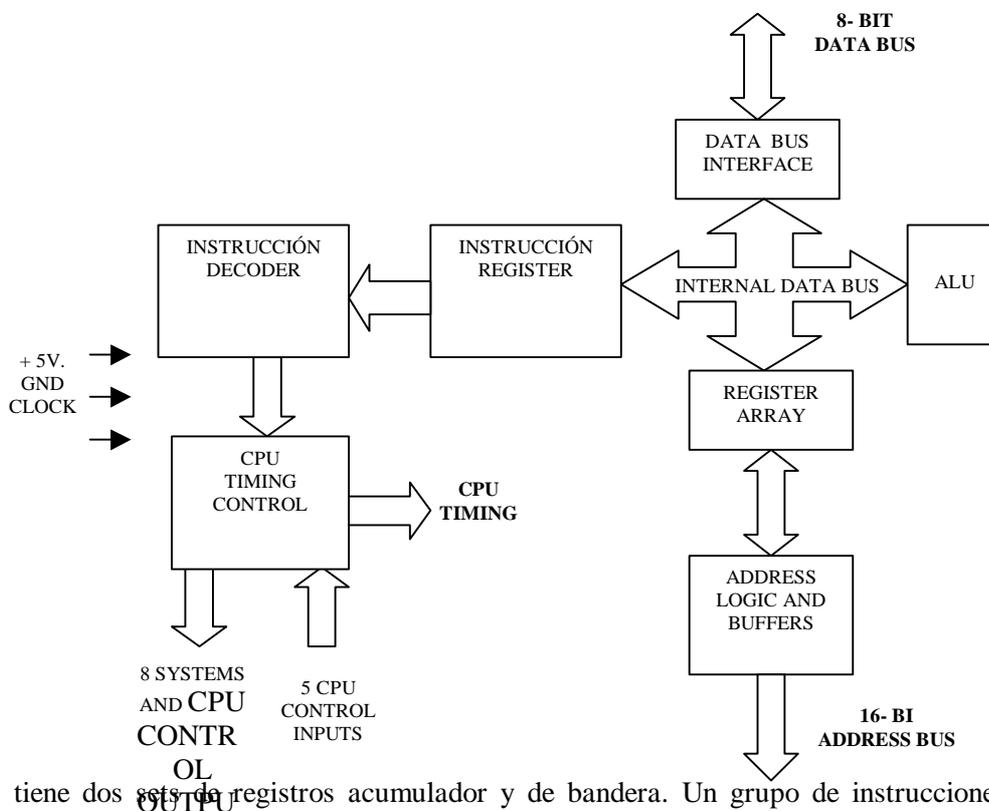
Z-80 CPU (ZILOG)

CARACTERÍSTICAS:

- ◆ El set de instrucciones contiene 158 instrucciones. Están incluidas las 78 instrucciones del 8080 y se mantiene la compatibilidad de software con el 8080.
- ◆ Reloj de 8, 6, 4 y 2.5 MHz. Para el Z80H, Z80B, Z80A y Z80 CPU, resultando una rápida ejecución de instrucciones con la consecuente transferencia elevada de datos.
- ◆ El extenso set de instrucciones incluye operaciones con palabras, bit, byte y cadena de caracteres. Búsqueda y transferencias de bloques a la vez mediante indexado y direccionamiento relativo, resultando el más competente y poderoso procesador de datos en la industria de los microcomputadores.
- ◆ El microprocesador Z80 y la familia asociada de periféricos controladores pueden ser enlazados por un sistema vectorizado de interrupciones. Este sistema podría ser Daisy-Chained que permita la implementación de un esquema de interrupciones prioritario, se requiere poca lógica adicional para el acoplamiento.
- ◆ Set duplicado de registros de banderas y de propósito general.
- ◆ Dos registros índices de 16 bits.
- ◆ Contador de refresco de memoria dinámica.

DESCRIPCION GENERAL

Los registros internos contienen 208 bits de memoria lectura/escritura que son accesibles por el programador. Estos registros incluyen 2 sets de 6 registros de propósito general, los cuales podrían ser usados individualmente como registros de 8 bits o como pares de registros de 16 bits.



Además tiene dos sets de registros acumulador y de bandera. Un grupo de instrucciones de intercambio que hacen que el registro principal o alternativo sean accesibles al programador. El set alternativo permite operaciones en modo foreground- background o podría ser reservado para respuesta muy rápida de interrupciones.

El Z80 además contiene un Stack Pointer, contador de programas, 2 registros índices, un registro de refresco (contador), y un registro de interrupciones. La CPU es fácil de incorporar en un sistema, puesto que requiere una simple fuente de energía de 5 volt.

FAMILIA DEL MICROPROCESADOR Z – 80

Zilog tiene diseñado cinco componentes que proporcionan un vasto soporte para el microprocesador Z – 80. Estos son:

- PIO (Paralled Input / Output), opera en ambos modos de transferencia: de datos I / O (con hand shake), y en el modo bit (sin hand shake). El PIO podría ser configurado como interfaz con dispositivos periféricos paralelos standard como impresoras y teclados.
- CTC (Counter / Timer Circuit).
- DMA (Controlador de acceso directo a memoria), proporciona una puerta bidireccional para operaciones de transferencia de datos y tiene la facultad de terminar la trayectoria de datos como resultado de un acuerdo preestablecido.
- SIO (Serial Input / Output Controller), ofrece dos canales, capaces de operar en una variedad de modos programables, para ambos es síncrona y asíncrona la comunicación, incluyendo Bi – Sync y SDLC.
- DART (Dual Asynchonus Receiver / Trasmmitter), dispotivo que provee comunicación serie asíncrona a bajo costo. Tiene dos canales y una interface de control de modem completa.

REGISTROS DE LA CPU

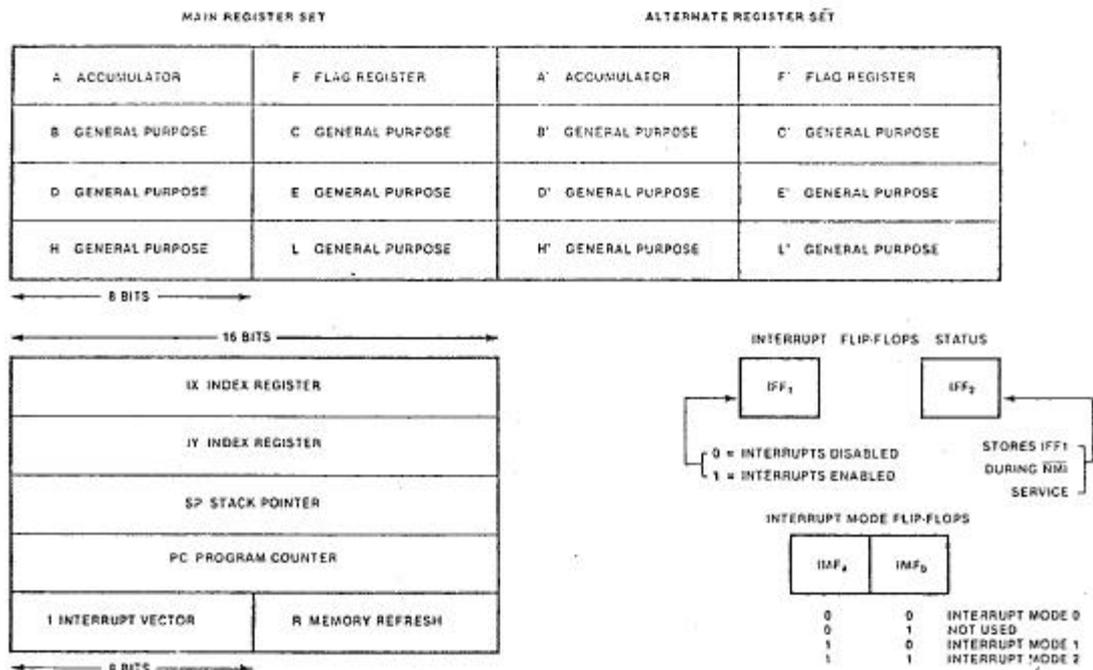


Figure 4. CPU Registers

La figura 4 muestra tres grupos de registros dentro de la CPU Z – 80:

- El primer grupo consiste de un set duplicado de registros de 8 bit: un set principal y un set alternativo. Ambos set constan de: registro acumulador, registro de banderas y seis registros de propósito general. La transferencia de datos, entre estos sets duplicados de registros, se efectúa mediante el uso de instrucciones de intercambio. El resultado es velocidad de

respuesta para interrupciones e implementación fácil y eficiente de técnicas versátiles de programación como procesamiento de datos background-foreground.

- El segundo grupo de registros consiste de 6 registros con funciones asignadas. Estos son: I (registro de interrupción), R (registro de refresco), IX e IY (registros índices), SP (Stack Pointer) y el PC (contador de programa)
- El tercer grupo consiste de 2 flip-flops de estado de interrupciones, mas un par adicional de flip-flops. los cuales ayudan a identificar el modo de interrupción en un determinado momento.

La tabla 1 provee información adicional de estos registros.

Tabla 1 – Registros de la CPU

Registro	Longitud en bits	Comentario
A,A' Acumulador	8	Almacena un operando ó el resultado de una operación.
F,F' Banderas	8	Ver set de instrucciones.
B,B' De propósito general	8	Puede ser usado individualmente o junto con C formando un registro de 16 bits.
C,C' De propósito general	8	Ver B.
D,D' De propósito general	8	Puede ser usado individualmente o junto con E formando un registro de 16 bits
E,E' De propósito general	8	Ver D.
H,H' De propósito general	8	Puede ser usado individualmente o junto con L formando un registro de 16 bits
L,L' De propósito general	8	Ver H.
		<i>Nota.</i> Los sets (BC), (DE) Y (HL) se combinan de la siguiente manera: B-Byte mas significativo. C-Byte menos significativo D-Byte mas significativo. E-Byte menos significativo. H-Byte mas significativo. L-Byte menos significativo
I Registro de interrupción.	8	Almacena los 8 bits más significativos de la dirección de memoria para procesamiento de interrupciones vectorizadas.
R Registro de refresco.	8	Provee, en forma transparente, el refresco de memorias dinámicas. Este registro se incrementa automáticamente y se coloca en el bus de direcciones durante cada ciclo de búsqueda de instrucción.
IX Registro índice.	16	Usado para direccionamiento indexado.
IY Registro índice.	16	Usado para direccionamiento indexado.
SP Apuntador de pila.	16	Contiene la dirección de la parte superior de la pila. Ver push o pop en el set de instrucciones.
PC Contador de programa.	16	Contiene la dirección de la próxima instrucción.
IFF ₁ -IFF ₂	Flip-flops	Toman los valores 1 ó 0 (set o reset) para indicar el estado de interrupción. Ver fig. 4
IMFa- IMFb	Flip-flops	Indican el modo de interrupción. Ver fig. 4.

INTERRUPCIONES: OPERACIÓN GENERAL

La CPU acepta dos señales de interrupción de entrada: $\overline{\text{NMI}}$ y $\overline{\text{INT}}$. La NMI es la no enmascarable y tiene la mas alta prioridad. La INT es una interrupción de menor prioridad y requiere ser habilitada por software para poder operar. INT puede ser conectada a múltiples dispositivos periféricos en una configuración wired-or.

El Z80 tiene un modo de respuesta único para el servicio de interrupción no enmascarable. La interrupción enmascarable INT, tiene disponible tres modos de respuesta programable:

- Modo 0: Similar a microprocesador 8080
- Modo 1: Servicio de interrupción de periféricos para uso con sistemas que no sean 8080/Z80.
- Modo 2: Esquema de interrupción vectorizada, usualmente Daisy – chained para uso de dispositivos periféricos compatibles y de la familia Z80.

La CPU interrumpe el servicio muestreando las señales $\overline{\text{NMI}}$ e $\overline{\text{INT}}$ en el flanco de subida del último pulso de reloj de una instrucción, luego el proceso de atención de interrupción depende del tipo de interrupción que fue detectada. Los detalles sobre la respuesta a interrupciones se muestran en la sección de diagramas de tiempo de la CPU.

Interrupción no enmascarable ($\overline{\text{NMI}}$):

Las interrupciones no enmascaradas no pueden ser deshabilitadas por programas de control y por lo tanto serán aceptadas en cualquier momento por la CPU.

$\overline{\text{NMI}}$ es generalmente reservada solo para atender tipo de interrupciones de muy alta prioridad, tal como cerrar ordenadamente después que una falla de alimentación fue detectada.

Después del reconocimiento de la señal $\overline{\text{NMI}}$ (suponiendo $\overline{\text{BUSREQ}}$ no activa) la CPU salta para reiniciar al la dirección 0066H. Normalmente el software de inicio guarda en esta dirección la rutina de servicio de interrupción.

Interrupción enmascarable ($\overline{\text{INT}}$):

Independientemente del modo de interrupción elegido por el usuario, el Z-80 responde a la entrada de una interrupción enmascarable siguiendo un mismo ciclo de tiempo.

Después que la interrupción ha sido detectada por la CPU (suponiendo que las interrupciones están habilitadas y $\overline{\text{BUSREQ}}$ es no activo) comienza un ciclo especial (M1) de búsqueda en el cual se activa $\overline{\text{IORQ}}$ (en vez de $\overline{\text{MREQ}}$, como en un ciclo normal M1) ,además este ciclo especial M1 va prolongado automáticamente por dos estados WAIT, para permitir el tiempo necesario de reconocimiento de solicitud de interrupción.

Operación de interrupcion en modo 0

Este modo es similar al procedimiento de servicio de interrupción del microprocesador 8080. el dispositivo que interrumpe coloca una instrucción en el bus de datos, esta es normalmente una instrucción de reinicio, la cual realizará una llamada para seleccionar una de la ocho posiciones de reinicio en la pagina cero de memoria. Distinto al 8080, la CPU Z – 80 responde a la instrucción de llamada con un solo ciclo de reconocimiento de interrupción seguido de dos ciclos de lectura de memoria.

Operación de interrupción en modo 1

La operación e modo 1 es muy similar al de $\overline{\text{NMI}}$. La principal diferencia es que el modo 1 de interrupción tiene solo una dirección de reinicio, 0038H.

Operación de interrupción en modo 2

Este modo de interrupción a sido diseñado para utilizar mas eficientemente la capacidad del microprocesador Z – 80 junto con la familia de periféricos asociados. El dispositivo que interrumpe selecciona la dirección de comienzo de la rutina de interrupción.

Esto lo hace colocando un vector de 8 – bits en el bus de datos, durante el ciclo de reconocimiento; la CPU forma un puntero usando estos 8 bits como los más bajos y el contenido del registro I como los 8 bits mas altos, este (puntero) apunta a una entrada de una tabla de direcciones de rutinas; la CPU entonces salta a la rutina que esta en esa dirección.

Esta flexibilidad en la selección de la dirección de rutinas, permite a los dispositivos periféricos usar distintos tipos de rutinas; estas rutinas pueden ser ubicadas en cualquier posición disponible de memoria. Puesto que el dispositivo que interrumpe proporciona el byte inferior del vector de dos byte, el bit 0 (A) debe ser cero.

Prioridad de las interrupciones (Interrupciones anidadas y Daisy chaining)

En una configuración daisy-chain, la prioridad de interrupciones de cada dispositivo periférico es determinada por la ubicación física del mismo, cada dispositivo tiene una línea de entrada de habilitación de interrupciones (IEI) y una línea de salida de habilitación de interrupción (IEO), la cual se conecta al siguiente dispositivo de menor prioridad. El primer dispositivo en la configuración (Daisy Caín) tiene su entrada IEI conectada a un nivel alto, este es el que tiene la más alta prioridad, y a cada dispositivo subsiguiente le corresponde una menor prioridad. Este arreglo le permite a la CPU seleccionar la interrupción de mayor prioridad entre varios periféricos que interrumpan simultáneamente; el dispositivo que interrumpe deshabilita la línea IEO del próximo dispositivo de menor prioridad hasta que este (el de menor prioridad) haya sido atendido, después de la atención la línea IEO se levanta, permitiendo a los periféricos de menor prioridad que soliciten atención o interrupción.

Operación de habilitar / deshabilitar interrupciones

Los dos Flip – Flops, IFF₁ y IFF₂, se usan para indicar a la CPU el estado de interrupción. La operación de los dos Flip – Flops se describe en la tabla 2. para mas detalles remitirse a Z – 80 CPU Technical Manual y Z – 80 Assenbly Language Manual.

Tabla 2: Flip – Flops de estado

Activo	IFF ₁	IFF ₂	Comments
CPU Reset	0	0	Maskable interrupt INT disabled
DI instrucción execution	0	0	Maskable interrupt INT' disabled
EI instrucción execution	1	1	Maskable interrupt INT' disabled
LD A, I instrucción execution	•	•	IFF ₂ → Parity Flag
LD A, R instrucción execution	•	•	IFF ₂ Parity Flag
Accept NMI'	0	•	Maskable interrupt INT' disabled
RETN instrucción execution	IFF ₂	•	IFF ₁ IFF ₂ at completion of an NMI' service routine

SET DE INSTRUCCIONES

El microprocesador tiene una de las más potentes y versátiles sets de instrucciones que cualquier microprocesador de 8 bits. Este incluye operaciones singulares tal como el movimiento de un bloque, lo cual permite transferencia de datos, rápidos y eficientes, dentro de la memoria o entre la memoria y una I/O. También permite operaciones sobre cualquier posición de la memoria. A continuación se da un resumen de set de instrucciones, el cual muestra el lenguaje mnemónico assembler, la operación, los estados de bandera y comentarios sobre cada instrucción.

Las instrucciones se dividen en las siguientes categorías:

- Carga de 8 bits.
- Carga de 16 bits.
- Intercambio, transferencia y búsqueda de bloques.
- Operaciones aritméticas y lógicas de 8 bits.
- Aritméticas de propósito general y control de CPU.
- Operaciones aritméticas de 16 bits.
- Solución y desplazamiento.
- Bit set, reset y operaciones de testeo.
- Saltos (jumps).

- Llamadas, retornos y reinicios.
- Operaciones de entrada y salida.

Se implementa una variedad de *modos de direccionamiento* que permiten una rápida y eficiente transferencia de datos entre varios registro, ubicaciones de memoria y dispositivos de I/O. Estos modos de direccionamiento son:

- Inmediato.
- Inmediato extendido.
- Pagina cero modificado.
- Relativo.
- Extendido.
- Indexado.
- Registro.
- Registro indirecto.
- Implícito.
- Bit.

Tablas de los grupos de sets de instrucciones y del resumen de operaciones de banderas

Mnemonic	Symbolic Operation	Flags					Opcode			Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments			
		S	Z	H	P/V	N	C	76	543						210		
LD r, r'	r ← r'	*	*	X	*	X	*	*	*	01	r	r'		1	1	4	r, r' Reg
LD r, n	r ← n	*	*	X	*	X	*	*	*	00	r	110		2	2	7	000 B
												←n→				001 C	
LD r, (HL)	r ← (HL)	*	*	X	*	X	*	*	*	01	r	110		1	2	7	010 D
LD r, (IX+d)	r ← (IX+d)	*	*	X	*	X	*	*	*	11	011	101	DD	3	5	19	011 E
										01	r,	110				100 H	
												←d→				101 L	
LD r, (Y+d)	r ← (Y+d)	*	*	X	*	X	*	*	*	11	111	101	FD	3	5	19	111 A
										01	r	110					
												←d→					
LD (HL), r	(HL) ← r	*	*	X	*	X	*	*	*	01	110	r		1	2	7	
LD (IX+d), r	(IX+d) ← r	*	*	X	*	X	*	*	*	11	011	101	DD	3	5	19	
										01	110	r					
												←d→					
LD (Y+d), r	(Y+d) ← r	*	*	X	*	X	*	*	*	11	111	101	FD	3	5	19	
										01	110	r					
												←d→					
LD (HL), n	(HL) ← n	*	*	X	*	X	*	*	*	00	110	110	36	2	3	10	
												←n→					
LD (IX+d), n	(IX+d) ← n	*	*	X	*	X	*	*	*	11	011	101	DD	4	5	19	
										00	110	110	36				
												←d→					
												←n→					

8-BIT LOAD GROUP (Continued)

Mnemonic	Symbolic Operation	Flags							Opcode			No. of Bytes	No. of M Cycles	No. of T States	Comments		
		S	Z	H	P/V	N	C	76	543	210	Hex						
LD (Y+d), A	(Y+d) ← n	.	.	X	.	X	.	.	.	11	111	101	FD	4	5	19	
										00	110	110	3E				
LD A, (BC)	A ← (BC)	.	.	X	.	X	.	.	.	00	001	010	0A	1	2	7	
LD A, (DE)	A ← (DE)	.	.	X	.	X	.	.	.	00	011	010	1A	1	2	7	
LD A, (nn)	A ← (nn)	.	.	X	.	X	.	.	.	00	111	010	3A	3	4	13	
LD (BC), A	(BC) ← A	.	.	X	.	X	.	.	.	00	000	010	02	1	2	7	
LD (DE), A	(DE) ← A	.	.	X	.	X	.	.	.	00	010	010	12	1	2	7	
LD (nn), A	(nn) ← A	.	.	X	.	X	.	.	.	00	110	010	32	3	4	13	
LD A, I	A ← I	†	†	X	0	X	IFF	0	.	11	101	101	ED	2	2	9	
										01	010	111	57				
LD A, R	A ← R	†	†	X	0	X	IFF	0	.	11	101	101	ED	2	2	9	
										01	011	111	5F				
LD I, A	I ← A	.	.	X	.	X	.	.	.	11	101	101	ED	2	2	9	
										01	000	111	47				
LD R, A	R ← A	.	.	X	.	X	.	.	.	11	101	101	ED	2	2	9	
										01	001	111	4F				

NOTE: IFF, the content of the interrupt enable flip-flop, (IFF₂), is copied into the P/V flag.

16-BIT LOAD GROUP

Mnemonic	Symbolic Operation	Flags							Opcode			No. of Bytes	No. of M Cycles	No. of T States	Comments		
		S	Z	H	P/V	N	C	76	543	210	Hex						
LD dd, nn	dd ← nn	.	.	X	.	X	.	.	.	00	dd0	001		3	3	10	dd Pair
																	00 BC
																	01 DE
																	11 SP
LD IX, nn	IX ← nn	.	.	X	.	X	.	.	.	11	011	101	DD	4	4	14	
										00	100	001	21				
LD IY, nn	IY ← nn	.	.	X	.	X	.	.	.	11	111	101	FD	4	4	14	
										00	100	001	21				
LD HL, (nn)	H ← (nn+1) L ← (nn)	.	.	X	.	X	.	.	.	00	101	010	2A	3	5	16	
LD dd, (nn)	dd _H ← (nn+1) dd _L ← (nn)	.	.	X	.	X	.	.	.	11	101	101	ED	4	6	20	
										01	0d1	011					

NOTE: (PAIR)_H, (PAIR)_L refer to high order and low order eight bits of the register pair respectively, e.g., BC_H = C, AF_H = A.

16-BIT LOAD GROUP (Continued)

Mnemonic	Symbolic Operation	Flags					Opcode			No. of Bytes	No. of M Cycles	No. of T States	Comments						
		S	Z	H	P/V	N/C	76	543	210					Hex					
LDIX, (nn)	$IX_H \leftarrow (nn+1)$.	.	X	.	X	.	.	.	11	011	101	DD	4	6	20			
	$IX_L \leftarrow (nn)$	00	101	010	2A						
LDIY, (nn)	$IY_H \leftarrow (nn+1)$.	.	X	.	X	.	.	.	11	111	101	FD	4	6	20			
	$IY_L \leftarrow (nn)$	00	101	010	2A						
LD(nn), HL	$(nn+1) \rightarrow H$ $(nn) \rightarrow L$.	.	X	.	X	.	.	.	00	100	010	22	3	5	16			
LD(nn), dd	$(nn+1) \rightarrow dd_H$ $(nn) \rightarrow dd_L$.	.	X	.	X	.	.	.	11	101	101	ED	4	6	20			
LD(nn), IX	$(nn+1) \rightarrow IX_H$.	.	X	.	X	.	.	.	11	011	101	DD	4	6	20			
	$(nn) \rightarrow IX_L$	00	100	010	22						
LD(nn), IY	$(nn+1) \rightarrow IY_H$.	.	X	.	X	.	.	.	11	111	101	FD	4	6	20			
	$(nn) \rightarrow IY_L$	00	100	010	22						
LDSP, HL	$SP \rightarrow HL$.	.	X	.	X	.	.	.	11	111	001	F9	1	1	6			
LDSP, IX	$SP \rightarrow IX$.	.	X	.	X	.	.	.	11	011	101	DD	2	2	10			
										11	111	001	F9						
LDSP, IY	$SP \rightarrow IY$.	.	X	.	X	.	.	.	11	111	101	FD	2	2	10			
PUSH qq	$(SP-2) \rightarrow qq_L$.	.	X	.	X	.	.	.	11	qq0	101		1	3	11	qq		
	$(SP-1) \rightarrow qq_H$																00	BC	
	$SP \rightarrow SP-2$																10	HL	
PUSH IX	$(SP-2) \rightarrow IX_L$.	.	X	.	X	.	.	.	11	011	101	DD	2	4	15	11		
	$(SP-1) \rightarrow IX_H$									11	100	101	E5						
	$SP \rightarrow SP-2$																		
PUSH IY	$(SP-2) \rightarrow IY_L$.	.	X	.	X	.	.	.	11	111	101	FD	2	4	15			
	$(SP-1) \rightarrow IY_H$									11	100	101	E5						
	$SP \rightarrow SP-2$																		
POP qq	$qq_H \leftarrow (SP+1)$.	.	X	.	X	.	.	.	11	qq0	001		1	3	10			
	$qq_L \leftarrow (SP)$																		
	$SP \rightarrow SP+2$																		
POP IX	$IX_H \leftarrow (SP+1)$.	.	X	.	X	.	.	.	11	011	101	DD	2	4	14			
	$IX_L \leftarrow (SP)$									11	100	001	E1						
	$SP \rightarrow SP+2$																		
POP IY	$IY_H \leftarrow (SP+1)$.	.	X	.	X	.	.	.	11	111	101	FD	2	4	14			
	$IY_L \leftarrow (SP)$									11	100	001	E1						
	$SP \rightarrow SP+2$																		

NOTE: (PAIR)_H, (PAIR)_L refer to high order and low order eight bits of the register pair respectively, e.g., BC_L = C, AF_H = A.

EXCHANGE, BLOCK TRANSFER, BLOCK SEARCH GROUPS

Mnemonic	Symbolic Operation	Flags				Opcode			Hex	No. of Bytes	No. of M. Cycles	No. of T. States	Comments			
		S	Z	H	PV	N	C	78						543	210	
EX DE, HL	DE ↔ HL	•	•	X	•	X	•	•	•	11 101 011	EB	1	1	4		
EX AF, AF'	AF ↔ AF'	•	•	X	•	X	•	•	•	00 001 000	08	1	1	4		
EXX	BC ↔ BC'	•	•	X	•	X	•	•	•	11 011 001	D9	1	1	4	Register bank and auxiliary register bank exchange	
	DE ↔ DE'	•	•	X	•	X	•	•	•							
	HL ↔ HL'	•	•	X	•	X	•	•	•							
EX (SP), HL	H ↔ (SP + 1)	•	•	X	•	X	•	•	•	11 100 011	E3	1	5	19		
	L ↔ (SP)	•	•	X	•	X	•	•	•							
EX (SP), IX	IX _H ↔ (SP + 1)	•	•	X	•	X	•	•	•	11 011 101	D0	2	6	23		
	IX _L ↔ (SP)	•	•	X	•	X	•	•	•	11 100 011	E3					
EX (SP), IY	IY _H ↔ (SP + 1)	•	•	X	•	X	•	•	•	11 111 101	FD	2	6	23		
	IY _L ↔ (SP)	•	•	X	•	X	•	•	•	11 100 011	E3					
LDI	(DE) ↔ (HL)	•	•	X	0	X	1	0	•	11 101 101	ED	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)	
	DE → DE + 1									10 100 000	A0					
	HL → HL + 1															
	BC → BC - 1															
LDIR	(DE) ↔ (HL)	•	•	X	0	X	0	0	•	11 101 101	ED	2	5	21	If BC ≠ 0	
	DE → DE + 1									10 110 000	B0	2	4	16	If BC = 0	
	HL → HL + 1															
	BC → BC - 1															
	Repeat until BC = 0															
LDD	(DE) ↔ (HL)	•	•	X	0	X	1	0	•	11 101 101	ED	2	4	16		
	DE → DE - 1									10 101 000	A8					
	HL → HL - 1															
	BC → BC - 1															
LDDR	(DE) ↔ (HL)	•	•	X	0	X	0	0	•	11 101 101	ED	2	5	21	If BC ≠ 0	
	DE → DE - 1									10 111 000	B8	2	4	16	If BC = 0	
	HL → HL - 1															
	BC → BC - 1															
	Repeat until BC = 0															
CPI	A - (HL)	1	1	X	1	X	1	1	•	11 101 101	ED	2	4	16		
	HL → HL + 1									10 100 001	A1					
	BC → BC - 1															

NOTE: ① PV flag is 0 if the result of BC - 1 = 0, otherwise PV = 1.
 ② PV flag is 0 only at completion of instruction.
 ③ Z flag is 1 if A = HL, otherwise Z = 0.

EXCHANGE, BLOCK TRANSFER, BLOCK SEARCH GROUPS (Continued)

Mnemonic	Symbolic Operation	Flags			Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments					
		S	Z	H	P/V	N	C	76					543	210	Hex		
CPIR	A ← (HL)	1	1	X	1	X	1	1	•	11	101	101	ED	2	5	21	If BC ≠ 0 and A ≠ (HL)
	HL ← HL + 1									10	110	001	B1	2	4	16	If BC = 0 or A = (HL)
	BC ← BC - 1 Repeat until A = (HL) or BC = 0																
CPD	A ← (HL)	1	1	X	1	X	1	1	•	11	101	101	ED	2	4	16	
	HL ← HL - 1									10	101	001	A9				
	BC ← BC - 1																
CPDR	A ← (HL)	1	1	X	1	X	1	1	•	11	101	101	ED	2	5	21	If BC ≠ 0 and A ≠ (HL)
	HL ← HL - 1									10	111	001	B9	2	4	16	If BC = 0 or A = (HL)
	BC ← BC - 1 Repeat until A = (HL) or BC = 0																

NOTE: ① P/V flag is 0 if the result of BC - 1 = 0, otherwise P/V = 1.
 ② P/V flag is 0 only at completion of instruction.
 ③ Z flag is 1 if A = (HL), otherwise Z = 0.

8-BIT ARITHMETIC AND LOGICAL GROUP

Mnemonic	Symbolic Operation	Flags			Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments								
		S	Z	H	P/V	N	C	76					543	210	Hex					
ADD A, r	A ← A + r	1	1	X	1	X	V	0	1	10	000	r	1	1	4	r Reg.				
ADD A, n	A ← A + n	1	1	X	1	X	V	0	1	11	000	110	2	2	7	000 B				
																				001 C
																				010 D
																				011 E
ADD A, (HL)	A ← A + (HL)	1	1	X	1	X	V	0	1	10	000	110	1	2	7	011 E				
ADD A, (IX+d)	A ← A + (IX+d)	1	1	X	1	X	V	0	1	11	011	101	DD	3	5	19	100 H			
																				101 L
																				111 A
ADD A, (IY+d)	A ← A + (IY+d)	1	1	X	1	X	V	0	1	11	111	101	FD	3	5	19				
										10	000	110								
ADC A, s	A ← A + s + CY	1	1	X	1	X	V	0	1		001						s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction. The indicated bits replace the 000 in the ADD set above.			
SUB s	A ← A - s	1	1	X	1	X	V	1	1		010									
SBC A, s	A ← A - s - CY	1	1	X	1	X	V	1	1		011									
AND s	A ← A ∧ s	1	1	X	1	X	P	0	0		100									
OR s	A ← A ∨ s	1	1	X	0	X	P	0	0		110									
XOR s	A ← A ⊕ s	1	1	X	0	X	P	0	0		101									
CP s	A ← s	1	1	X	1	X	V	1	1		111									

8-BIT ARITHMETIC AND LOGICAL GROUP (Continued)

Mnemonic	Symbolic Operation	Flags					Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments		
		S	Z	H	P/V	N	C	78	543	210					Hex	
INC r	r ← r+1	↑	↑	X	↑	X	V	0	*	00	r	<u>100</u>	1	1	4	
INC (HL)	(HL) ← (HL)+1	↑	↑	X	↑	X	V	0	*	00	110	<u>100</u>	1	3	11	
INC (IX+d)	(IX+d) ← (IX+d)+1	↑	↑	X	↑	X	V	0	*	11	011	101	DD	3	6	23
	00									110	<u>100</u>					
INC (IY+d)	(IY+d) ← (IY+d)+1	↑	↑	X	↑	X	V	0	*	11	111	101	FD	3	6	23
	00									110	<u>100</u>					
DEC m	m ← m-1	↑	↑	X	↑	X	V	1	*			<u>101</u>				

NOTE: m is any of r, (HL), (IX+d), (IY+d) as shown for INC. DEC same format and states as INC. Replace 100 with 101 in opcode.

GENERAL-PURPOSE ARITHMETIC AND CPU CONTROL GROUPS

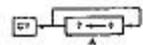
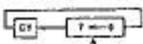
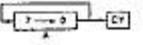
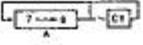
Mnemonic	Symbolic Operation	Flags					Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments			
		S	Z	H	P/V	N	C	78	543	210					Hex		
DAA	⊕	↑	↑	X	↑	X	P	*	↑	00	100	111	27	1	1	4	Decimal adjust accumulator.
CPL	A ← A	*	*	X	1	X	*	1	*	00	101	111	2F	1	1	4	Complement accumulator (one's complement).
NEG	A ← 0 - A	↑	↑	X	↑	X	V	1	↑	11	101	101	ED	2	2	8	Negate acc. (two's complement).
										01	000	100					
CCF	CY ← CY	*	*	X	X	X	*	0	↑	00	111	111	3F	1	1	4	Complement carry flag.
SCF	CY ← 1	*	*	X	0	X	*	0	1	00	110	111	37	1	1	4	Set carry flag.
NOP	No operation	*	*	X	*	X	*	*	*	00	000	000	00	1	1	4	
HALT	CPU halted	*	*	X	*	X	*	*	*	01	110	110	78	1	1	4	
DI *	IFF ← 0	*	*	X	*	X	*	*	*	11	110	011	F3	1	1	4	
EI *	IFF ← 1	*	*	X	*	X	*	*	*	11	111	011	F8	1	1	4	
IM 0	Set interrupt mode 0	*	*	X	*	X	*	*	*	11	101	101	ED	2	2	8	
										01	000	110					
IM 1	Set interrupt mode 1	*	*	X	*	X	*	*	*	11	101	101	ED	2	2	8	
										01	010	110					
IM 2	Set interrupt mode 2	*	*	X	*	X	*	*	*	11	101	101	ED	2	2	8	
										01	011	110					

NOTES: ⊕ converts accumulator content into packed BCD following add or subtract with packed BCD operands.
 IFF indicates the interrupt enable flip-flop.
 CY indicates the carry flip-flop.
 * indicates interrupts are not sampled at the end of EI or DI.

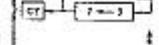
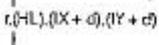
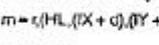
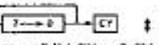
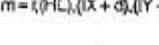
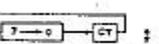
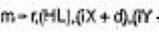
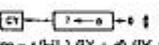
16-BIT ARITHMETIC GROUP

Mnemonic	Symbolic Operation	S	Z	Flags				Opcode			Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments		
				H	P/V	N	C	76	543	210					Reg.	Reg.	
ADD HL, ss	HL ← HL + ss	*	*	X	X	X	*	0	†	00	ss1	001	1	3	11	ss Reg. 00 BC	
ADCHL, ss	HL ←															01 DE	
	HL + ss + CY	†	†	X	X	X	V	0	†	11	101	101	ED	2	4	15	10 HL 11 SP
SBC HL, ss	HL ←																
	HL - ss - CY	†	†	X	X	X	V	1	†	11	101	101	FD	2	4	15	
ADD IX, pp	IX ← IX + pp	*	*	X	X	X	*	0	†	11	011	101	DD	2	4	15	pp Reg. 00 BC
										01	pp1	001					01 DE 10 IX 11 SP
ADD IY, rr	IY ← IY + rr	*	*	X	X	X	*	0	†	11	111	101	FD	2	4	15	rr Reg. 00 BC
										00	rr1	001					01 DE
INC ss	ss ← ss + 1	*	*	X	*	X	*	*	*	00	ss0	011	1	1	6	01 DE	
INC IX	IX ← IX + 1	*	*	X	*	X	*	*	*	11	011	101	DD	2	2	10	10 IY
										00	100	011	23				11 SP
INC IY	IY ← IY + 1	*	*	X	*	X	*	*	*	11	111	101	FD	2	2	10	
										00	100	011	23				
DEC ss	ss ← ss - 1	*	*	X	*	X	*	*	*	00	ss1	011	1	1	6		
DEC IX	IX ← IX - 1	*	*	X	*	X	*	*	*	11	011	101	DD	2	2	10	
										00	101	011	2B				
DEC IY	IY ← IY - 1	*	*	X	*	X	*	*	*	11	111	101	FD	2	2	10	
										00	101	011	2B				

ROTATE AND SHIFT GROUP

Mnemonic	Symbolic Operation	S	Z	Flags				Opcode			Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments		
				H	P/V	N	C	76	543	210							
RLCA		*	*	X	0	X	*	0	†	00	000	111	07	1	1	4	Rotate left circular accumulator.
RLA		*	*	X	0	X	*	0	†	00	010	111	17	1	1	4	Rotate left accumulator.
RRCA		*	*	X	0	X	*	0	†	00	001	111	0F	1	1	4	Rotate right circular accumulator.
RRA		*	*	X	0	X	*	0	†	00	011	111	1F	1	1	4	Rotate right accumulator.

ROTATE AND SHIFT GROUP (Continued)

Mnemonic	Symbolic Operation	Flags					Opcode			No. of Bytes	No. of M Cycles	No. of T States	Comments		
		S	Z	H	P/V	N/C	76	543	210					Hex	
RLC r		†	†	X	0	X	P	0	†	11 001 011	CB	2	2	8	Rotate left circular register r.
RLC (HL)		†	†	X	0	X	P	0	†	11 001 011	CB	2	4	15	r Reg. 000 B 001 C 010 D 011 E 001 H 101 L 111 A
RLC (IX+d)		†	†	X	0	X	P	0	†	11 011 101	DD	4	6	23	
RLC (IY+d)		†	†	X	0	X	P	0	†	11 001 011	CB	4	6	23	
RL m		†	†	X	0	X	P	0	†	00 000 110					
RRC m		†	†	X	0	X	P	0	†	00 000 110					
RR m		†	†	X	0	X	P	0	†	00 000 110					
SLA m		†	†	X	0	X	P	0	†	00 000 110					
SRA m		†	†	X	0	X	P	0	†	00 000 110					
SRL m		†	†	X	0	X	P	0	†	00 000 110					
RLD		†	†	X	0	X	P	0	†	11 101 101	ED	2	5	18	Rotate digit left and right between the accumulator and location (HL).
RLD		†	†	X	0	X	P	0	†	01 101 111	6F				
RLD		†	†	X	0	X	P	0	†	11 101 101	ED	2	5	18	The content of the upper half of the accumulator is unaffected.
RLD		†	†	X	0	X	P	0	†	01 100 111	67				

Instruction format and states are as shown for RLCs. To form new opcode replace 000 or RLCs with shown code.

BIT SET/RESET AND TEST GROUP

Mnemonic	Symbolic Operation	Flags						Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments				
		S	Z	H	P/V	N	C	76	543	210	Hex								
BIT b, r	$Z \leftarrow r_b$	X	1	X	1	X	X	0	•	11	001	011	CB	2	2	8	r Reg. 000 B		
BIT b, (HL)	$Z \leftarrow (HL)_b$	X	1	X	1	X	X	0	•	11	001	011	CB	2	3	12	001 C		
										01	b	r	010 D						
BIT b, (X+d) _b	$Z \leftarrow (X+d)_b$	X	1	X	1	X	X	0	•	11	011	101	DD	4	5	20	011 E		
										11	001	011	CB				100 H		
										← d →			01				b	r	101 L
BIT b, (Y+d) _b	$Z \leftarrow (Y+d)_b$	X	1	X	1	X	X	0	•	11	111	101	FD	4	5	20	b Bit Tested 000 0		
										11	001	011	CB				001 1		
										← d →			01				b	r	010 2
										← d →			01				b	r	011 3
SET b, r	$r_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	001	011	CB	2	2	8	100 4		
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	001	011	CB	2	4	15	101 5		
										11	b	r	110 6						
SET b, (X+d)	$(X+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	23	111 7		
										11	001	011	CB						
										← d →			11				b	r	110
SET b, (Y+d)	$(Y+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	111	101	FD	4	6	23			
										11	001	011	CB						
										← d →			11				b	r	110
RES b, m	$m_b \leftarrow 0$ m = r, (HL), (X+d), (Y+d)	•	•	X	•	X	•	•	•	11	b	110	10						

To form new opcode replace **11** of SET b, s with **10**. Flags and time states for SET instruction.

NOTE: The notation m_b indicates location m, bit b (0 to 7).

JUMP GROUP

Mnemonic	Symbolic Operation	Flags					Opcode			Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments	
		S	Z	H	P/V/N	C	75	543	210						
JP nn	PC ← nn	*	*	X	*	X	*	*	*	11 000 011	C3	3	3	10	cc Condition 000 NZ (non-zero) 001 Z (zero) 010 NC (non-carry) 011 C (carry) 100 PO (parity odd) 101 PE (parity even) 110 P (sign positive) 111 M (sign negative)
JP cc, nn	If condition cc is true PC ← nn, otherwise continue	*	*	X	*	X	*	*	*	11 cc 010	3	3	10		
															→ n →
JR e	PC ← PC + e	*	*	X	*	X	*	*	*	00 011 000	18	2	3	12	
JR C, e	If C = 0, continue If C = 1, PC ← PC + e	*	*	X	*	X	*	*	*	00 111 000	38	2	2	7	If condition not met.
JR NC, e	If C = 1, continue If C = 0, PC ← PC + e	*	*	X	*	X	*	*	*	00 110 000	30	2	2	7	If condition not met.
JP Z, e	If Z = 0, continue If Z = 1, PC ← PC + e	*	*	X	*	X	*	*	*	00 101 000	28	2	2	7	If condition not met.
JR NZ, e	If Z = 1, continue If Z = 0, PC ← PC + e	*	*	X	*	X	*	*	*	00 100 000	20	2	2	7	If condition not met.
JP (HL)	PC ← HL	*	*	X	*	X	*	*	*	11 101 001	E9	1	1	4	
JP (IX)	PC ← IX	*	*	X	*	X	*	*	*	11 011 101	D0	2	2	8	
JP (IY)	PC ← IY	*	*	X	*	X	*	*	*	11 111 101	FD	2	2	8	
DJNZ, e	B ← B - 1 If B = 0, continue If B ≠ 0, PC ← PC + e	*	*	X	*	X	*	*	*	00 010 000	10	2	2	8	If B = 0
												2	3	13	If B ≠ 0

NOTES: e represents the extension in the relative addressing mode.
 e is a signed two's complement number in the range < -128, 129 >.
 e - 2 in the opcode provides an effective address of pc + e as PC is incremented by 2 prior to the addition of e.

CALL AND RETURN GROUP

Mnemonic	Symbolic Operation	Flags					Opcode			No. of Bytes	No. of Cycles	M. States	No. of T States	Comments			
		S	Z	H	P/V	N/C	76	543	210						Hex		
CALL nn	(SP-1)→PC _H (SP-2)→PC _L PC ← nn	•	•	X	•	X	•	•	•	11	001	101	CD	3	5	17	
CALL cc,nn	If condition cc is false: continue, otherwise same as CALL nn	•	•	X	•	X	•	•	•	11	cc	100		3	3	10	If cc is false.
														3	5	17	If cc is true.
RET	PC _L ← (SP) PC _H ← (SP+1)	•	•	X	•	X	•	•	•	11	001	001	CD	1	3	10	
RET cc	If condition cc is false: continue, otherwise same as RET	•	•	X	•	X	•	•	•	11	cc	000		1	1	5	If cc is false.
														1	3	11	If cc is true.
																	cc Condition
																	000 NZ (non-zero)
																	001 Z (zero)
																	010 NC (non-carry)
																	011 C (carry)
																	100 PO (parity odd)
																	101 PE (parity even)
																	110 P (sign positive)
																	111 M (sign negative)
RST p	(SP-1)→PC _H (SP-2)→PC _L PC _H ← 0 PC _L ← p	•	•	X	•	X	•	•	•	11	t	111		1	3	11	t p
																	000 00H
																	001 06H
																	010 10H
																	011 18H
																	100 20H
																	101 28H
																	110 30H
																	111 38H

NOTE: ¹RETn loads IFF₂ → IFF₁

INPUT AND OUTPUT GROUP

Mnemonic	Symbolic Operation	Flags					Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments				
		S	Z	H	P/V/N	C	76	543	210	Hex								
IN A, (n)	A ← (n)	•	•	X	•	X	•	•	•	•	11	011	01	DB	2	3	11	n to A ₀ ~ A ₇ Acc. to A ₈ ~ A ₁₅
IN C, (C)	r ← (C) <i>if r = 110 only the flags will be affected</i>	1	†	X	†	X	P	0	•	•	11	101	101	ED	2	3	12	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
IN I	(HL) ← (C) B ← B - 1 HL ← HL + 1	X	†	X	X	X	X	1	X	•	11	101	101	ED	2	4	16	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
											10	100	010	A2				
IN R	(HL) ← (C) B ← B - 1 HL ← HL + 1 Repeat until B = 0	X	1	X	X	X	X	1	X	•	11	101	101	ED	2	5 (if B ≠ 0)	21	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
											10	110	010	B2				
IN D	(HL) ← (C) B ← B - 1 HL ← HL - 1	X	†	X	X	X	X	1	X	•	11	101	101	ED	2	4	16	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
											10	101	010	AA				
IN DR	(HL) ← (C) B ← B - 1 HL ← HL - 1 Repeat until B = 0	X	1	X	X	X	X	1	X	•	11	101	101	ED	2	5 (if B ≠ 0)	21	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
											10	111	010	BA				
OUT (n), A	(n) ← A	•	•	X	•	X	•	•	•	•	11	010	011	D3	2	3	11	n to A ₀ ~ A ₇ Acc. to A ₈ ~ A ₁₅
OUT (C), r	(C) ← r	•	•	X	•	X	•	•	•	•	11	101	101	ED	2	3	12	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
OUT I	(C) ← (HL) B ← B - 1 HL ← HL + 1	X	†	X	X	X	X	1	X	•	11	101	101	ED	2	4	16	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
											10	100	011	A3				
OUT R	(C) ← (HL) B ← B - 1 HL ← HL + 1 Repeat until B = 0	X	1	X	X	X	X	1	X	•	11	101	101	ED	2	5 (if B ≠ 0)	21	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
											10	110	011	B3				
OUT D	(C) ← (HL) B ← B - 1 HL ← HL - 1	X	†	X	X	X	X	1	X	•	11	101	101	ED	2	4	16	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
											10	101	011	AB				
OUT DR	(C) ← (HL) B ← B - 1 HL ← HL - 1 Repeat until B = 0	X	1	X	X	X	X	1	X	•	11	101	101	ED	2	5 (if B ≠ 0)	21	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
											10	111	011					

NOTES: ① If the result of B - 1 is zero, the Z flag is set, otherwise it is reset.
 ② Z flag is set upon instruction completion only.

SUMMARY OF FLAG OPERATION

Instructions	D ₇			D ₀			Comments		
	S	Z	H	P/V	N	C			
ADD A, s; ADDA, s	↑	↓	X	↑	X	V	0	1	8-bit add or add with carry.
SUB s; SBCA, s; CP s; NEG	↑	↓	X	↑	X	V	1	1	8-bit subtract, subtract with carry, compare and negate accumulator.
AND s	↑	↓	X	↑	X	P	0	0	Logical operation.
OR s; XOR s	↑	↓	X	0	X	P	0	0	Logical operation.
INC s	↑	↓	X	↑	X	V	0	*	8-bit increment.
DEC s	↑	↓	X	↑	X	V	1	*	8-bit decrement.
ADD DD, ss	*	*	X	X	X	*	0	↑	16-bit add.
ADCHL, ss	↑	↓	X	X	X	V	0	↑	16-bit add with carry.
SBCHL, ss	↑	↓	X	X	X	V	1	↑	16-bit subtract with carry.
RLA; RLCA; RRA; RRCA	*	*	X	0	X	*	0	↑	Rotate accumulator.
RL m; RLC m; RR m; RRC m; SRA m; SRL m	↑	↓	X	0	X	P	0	↑	Rotate and shift locations.
RLD; RRD	↑	↓	X	0	X	P	0	*	Rotate digit left and right.
DAA	↑	↓	X	↑	X	P	*	↑	Decimal adjust accumulator.
CPL	*	*	X	1	X	*	1	*	Complement accumulator.
SCF	*	*	X	0	X	*	0	1	Set carry.
CCF	*	*	X	X	X	*	0	↑	Complement carry.
IN r (C)	↑	↓	X	0	X	P	0	*	Input register indirect.
INI; IND; OUT; OTUD	X	↑	X	X	X	X	1	*	Block input and output. Z = 1 if B ≠ 0, otherwise Z = 0.
INIR; INDR; OTIR; OTDR	X	↑	X	X	X	X	1	*	Block input and output. Z = 1 if B ≠ 0, otherwise Z = 0.
LDI; LDD	X	X	X	0	X	↑	0	*	Block transfer instructions. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LDIR; LDDR	X	X	X	0	X	0	0	*	Block transfer instructions. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
CPI; CPIR; CPD; CPDR	X	↑	X	X	X	↑	1	*	Block search instructions. Z = 1 if A = (HL), otherwise Z = 0. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LDA I; LDA, R	↑	↓	X	0	X	IFF	0	*	IFF, the content of the interrupt enable flip-flop, (IFF ₂), is copied into the P/V flag.
BIT b, s	X	↑	X	1	X	X	0	*	The state of bit b of location s is copied into the Z flag.

Notación simbólica:

Símbolo	Operación
S	Bandera de signo, S=1 si el MSB (bit más significativo) del resultado es 1.
Z	Bandera de cero, Z=1 si el resultado de la operación es 0.
P/V	Bandera de overflow o paridad. Paridad (P) y Overflow (V) comparten la misma bandera. Las operaciones lógicas afectan esta bandera con la paridad del resultado, mientras que las operaciones aritméticas afectan esta bandera con el overflow del resultado. Si p/v contiene paridad: P/V=1 si el resultado de la operación es par y P/V=0 si el resultado es impar. Si P/V contiene overflow: P/V=1 si el resultado de la operación produce un overflow y P/V=0 si no se produce un overflow.
H*	Bandera de acarreo-mitad. H=1 si una operación de suma o resta produce un acarreo en, o desde, el bit 4 del acumulador.
N*	Bandera de suma-resta. N=1 si la operación previa fue una resta.
C	Bandera carry/linn. C=1 si la operación produce un acarreo del MSB del operando o resultado.
↕	La bandera es afectada de acuerdo al resultado de operación.
.	La bandera no es modificada por la operación.
O	La bandera es reseteada por la operación.
½	La bandera es seteada por la operación.
X	La bandera indeterminada.
V	La bandera P/V es afectada según el overflow del resultado de operación.
P	La bandera P/V es afectada según la paridad del resultado de operación.
r	Cualquier registro de la CPU: A, B, C, D, E, H, L.
s	Cualquier posición de 8 bits, para todos los modos de direccionamiento permitido por la instrucción particular.
ss	Cualquier posición de 16 bits, para todos los modos de direccionamiento permitido por la instrucción particular.
ii	Cualquiera de los registros índices IX ó IY.

R	Contador de refresco.
n	Valor de 8 bits comprendido entre <0,255>
mn	Valor de 16 bits comprendido entre <0,65535>

*Las banderas H y N se usan junto con la instrucción de ajuste decimal (DAA) para proporcionar resultados correctos en formato BCD condensado, permitiendo sumar y restar usando operandos con formato BCD condensado.

DESCRIPCIÓN DE PINES

- **A0 – A15.** Bus de direcciones (salida, activa en alto, de 3 estados). Forma un bus de direcciones de 16 bits. El bus de direcciones provee las direcciones para realizar intercambios en el bus de datos de la memoria (hasta 64 Kbytes) y en los dispositivos de I/O.
- **BUSREQ.** Solicitud de bus, (entrada, activa en bajo). Solicitud de bus tiene mayor prioridad que NMI y siempre se la reconoce al finalizar el ciclo de máquina actual. BUSREQ obliga al bus de direcciones, al bus de datos y a las señales de control MREQ, IORQ, RD y WR de la CPU a pasar a estado de alta impedancia para que otros