OR dst, src

< Logical OR >

Operation : dst← dst OR src

Description : Ors the contents of dst with those of src and loads the result to dst.

(Truth table)

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	○	OR	R, r	1 1 z z 1 r 1 1 1 0 0 R
○	○	○	OR	r, #	1 1 z z 1 r 1 1 0 0 1 1 1 0 #<7:0> #<15:8> #<23:16> #<31:24>
○	○	○	OR	R, (mem)	1 m z z m m m m 1 1 1 0 0 R
○	○	○	OR	(mem), R	1 m z z m m m m 1 1 1 0 1 R
○	○	×	OR<W>	(mem), #	1 m 0 z m m m m 0 0 1 1 1 1 1 0 #<7:0> #<15:8>

Flags :

S	Z	H	V	N	C
*	*	0	*	0	0

S = MSB value of the result is set.
Z = 1 is set when the result is 0, otherwise 0.
H = 0 is set.
V = 1 is set when the parity (number of 1s) of the result is even, 0 when odd. When the operand is 32-bit, an undefined value is set.
N = Cleared to 0.
C = Cleared to 0.

Execution example: OR HL, IX

When the HL register = 7350H and the IX register is 3456H, execution sets the HL register to 7756H.

```
         0111   0011   0101   0000   ←   HL register (before execution)
OR   )   0011   0100   0101   0110   ←   IX register (before execution)
         0111   0111   0101   0110   ←   HL register (after execution)
```

ORCF num, src

< OR Carry Flag >

Operation : CY ← CY OR src<num>

Description : Ors the contents of the carry flag with those of bit num of src and loads the result to the carry flag.

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	×	ORCF	#4, r	1 1 0 z 1 r 0 0 1 0 0 0 0 1 0 0 0 0 # 4
○	○	×	ORCF	A, r	1 1 0 z 1 r 0 0 1 0 1 0 0 1
○	×	×	ORCF	#3, (mem)	1 m 1 1 m m m m 1 0 0 0 1 #3
○	×	×	ORCF	A, (mem)	1 m 1 1 m m m m 0 0 1 0 1 0 0 1

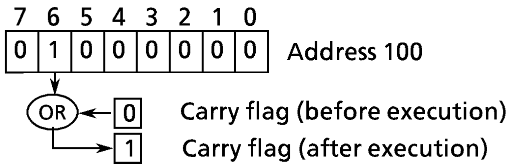
Note : When bit num is specified by the A register, the value of the lower 4 bits of the A register is used as bit num. When the operand is a byte and the value of the lower bits of bit num is from 8 to 15, the result is undefined.

flag :

S	Z	H	V	N	C
–	–	–	–	–	*

S = No change
Z = No change
H = No change
V = No change
N = No change
C = The result of or-ing the contents of the carry flag with those of bit num of src is set.

Execution example: ORCF 6, (100H)
When the contents of memory at address 100H = 01000000B (binary) and the carry flag = 0, execution sets the carry flag to 1.



PAA dst

< Pointer Adjust Accumulator >

Operation : if dst <LSB> = 1 then dst ← dst + 1

Description : Increments dst by 1 when the LSB of dst is 1. Does nothing when the LSB of dst is 0.
Used to make the contents of dst even. With the TLCS-900 series, when accessing 16- or 32-bit data in memory, if the data are loaded from an address starting with an even number, the number of bus cycles is 1 less than that of the data loaded from an address starting with an odd number.

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
×	○	○	PAA	r	1 1 z z 1 r 0 0 0 1 0 1 0 0

Flags :

S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change

Execution example: PAA XIZ
When the XIZ register = 00234567H, execution increments the XIZ register by 1 so that it becomes 00234568H.

POP dst

< Pop >

Operation : dst ← (XSP+) $\left[\begin{array}{ll}\text{In bytes} & : \text{dst}\leftarrow(\text{XSP}),\ \text{XSP}\leftarrow\text{XSP}+1\\ \text{In words} & : \text{dst}\leftarrow(\text{XSP}),\ \text{XSP}\leftarrow\text{XSP}+2\\ \text{In long words} & : \text{dst}\leftarrow(\text{XSP}),\ \text{XSP}\leftarrow\text{XSP}+4\end{array}\right]$

Description : First loads the contents of memory address specified by the stack pointer XSP to dst. Then increments the stack pointer XSP by the number of bytes in the operand.

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	×	×	POP	F	0 0 0 1 1 0 0 1
○	×	×	POP	A	0 0 0 1 0 1 0 1
×	○	○	POP	R	0 1 0 s 1 R
○	○	○	POP	r	1 1 z z 1 r 0 0 0 0 0 1 0 1
○	○	×	POP<W>	(mem)	1 m 1 1 m m m m 0 0 0 0 0 1 z 0

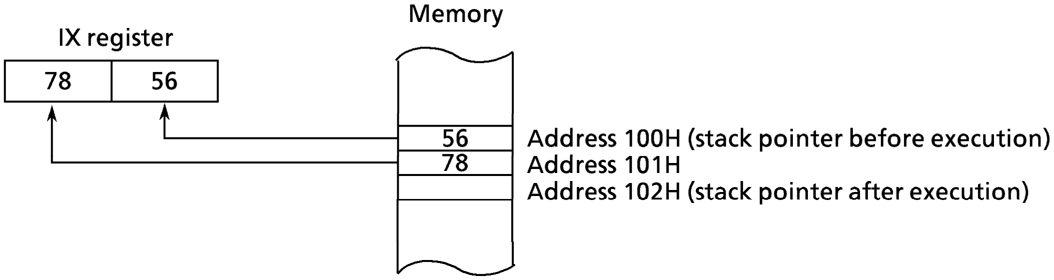
Flags :

S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change
(Note) Executing POP F changes all flags.

Execution example: POP IX

When the stack pointer XSP = 0100H, the contents of address 100H = 56H, and the contents of address 101H = 78H, execution sets the IX register to 7856H and the stack pointer XSP to 0102H.


Memory
IX register
78
56
56
78
Address 100H (stack pointer before execution)
Address 101H
Address 102H (stack pointer after execution)

POP SR

< Pop SR >

Operation : SR←(XSP+)

Description : Loads the contents of the address specified by the stack pointer XSP to status register. Then increments the contents of the stack pointer XSP by 2.

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
×	○	×	POP	SR	0 0 0 0 0 0 1 1

Flags :

S	Z	H	V	N	C
*	*	*	*	*	*

$$\left.\begin{array}{l} S = \\ Z = \\ H = \\ V = \\ N = \\ C = \end{array}\right\}$$ Contents of the memory address specified by the stack pointer XSP are set.

Note1: Please execute this instruction during DI condition. The timing for executing this instruction is delayed by several states than that for fetching the instruction. This is because an instruction queue (4 bytes) and pipeline processing method is used.

Note2: When this instruction is executed, change all bits in the SR. If there are the bits which must not change (example: the minimum mode is not supported for 900/H, therefor, the SR<MAX> register must be set to "1"), prevent the bits from changing.

PUSH SR

<Push SR>

Operation : (−XSP)←SR

Description : Decrements the contents of the stack pointer XSP by 2. Then loads the contents of status register to the memory address specified by the stack pointer XSP.

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
×	○	×	PUSH	SR	0 0 0 0 0 0 1 0

Flags :

S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change

PUSH src

< Push >

Operation : (−XSP) ← src
[In bytes : XSP←XSP−1, (XSP)←src
In words : XSP←XSP−2, (XSP)←src
In long words : XSP←XSP−4, (XSP)←src]

Description : Decrements the stack pointer XSP by the byte length of the operand. Then loads the contents of src to the memory address specified by the stack pointer XSP.

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	×	×	PUSH	F	0 0 0 1 1 0 0 0
○	×	×	PUSH	A	0 0 0 1 0 1 0 0
×	○	○	PUSH	R	0 0 1 s 1 R
○	○	○	PUSH	r	1 1 z z 1 r 0 0 0 0 0 1 0 0
○	○	×	PUSH<W>	#	0 0 0 0 1 0 z 1 #<7:0> #<15:8>
○	○	×	PUSH<W>	(mem)	1 m 0 z m m m m 0 0 0 0 0 1 0 0

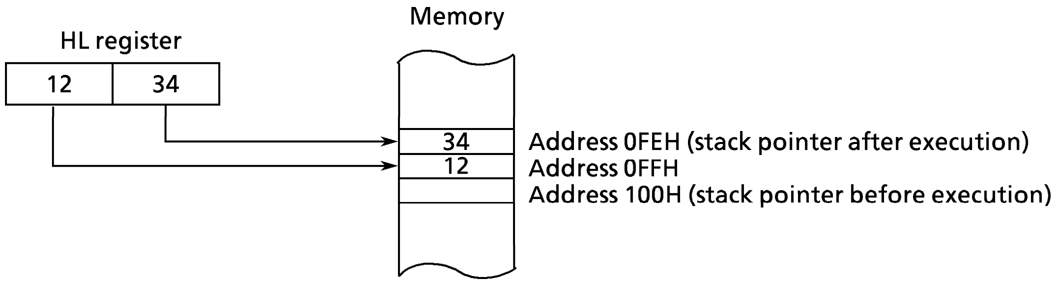
Flags :

S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change

Execution example: PUSH HL

When the stack pointer XSP = 0100H and the HL register = 1234H, execution changes address 00FEH to 34H, address 00FFH to 12H, and sets the stack pointer XSP to 00FEH.


Memory
HL register
12
34
34
12
Address 0FEH (stack pointer after execution)
Address 0FFH
Address 100H (stack pointer before execution)

RCF

< Reset Carry Flag >

Operation : CY ← 0

Description : Resets the carry flag to 0.

Details :

Mnemonic	Code
RCF	0 0 0 1 0 0 0 0

Flags :

S	Z	H	V	N	C
–	–	0	–	0	0

S = No change
Z = No change
H = Reset to 0.
V = Reset to 0.
N = No change
C = Reset to 0.

RES num, dst

< Reset >

Operation : dst <num> ← 0

Description : Resets bit num of dst to 0.

Details :

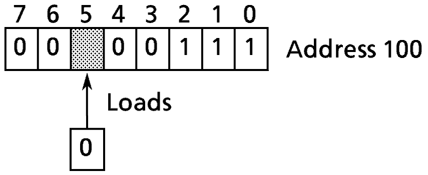
Size			Mnemonic		Code
Byte	Word	Long word			
○	○	×	RES	#4, r	1 1 0 z 1 r 0 0 1 1 0 0 0 0 0 0 0 0 # 4
○	×	×	RES	#3, (mem)	1 m 1 1 m m m m 1 0 1 1 0 #3

Flags :

S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change

Execution example: RES 5, (100H)
When the contents of memory at address 100H = 00100111B (binary), execution sets the contents to 00000111B (binary).



RET condition

< Return >

Operation : If cc is true, then the 32-bit PC ← (XSP), XSP ← XSP + 4.

Description : Pops the return address from the stack area to the program counter when the operand condition is true.

Details :

Mnemonic	Code
RET	0 0 0 0 1 1 1 0
RET cc	1 0 1 1 0 0 0 0 1 1 1 1 c c

Flags :

S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change

Execution example: RET

When the stack pointer XSP = 0FCH and the contents of memory at address 0FCH = 9000H (long word data), execution sets the stack pointer XSP to 100H and jumps (returns) to address 9000H.

RETD num

< Return and Deallocate >

Operation : 32-bit PC ← (XSP), XSP ← XSP + 4, XSP ← XSP + num

Description : Pops the return address from the stack area to the program counter. Then increments the stack pointer XSP by signed num.

Details :

Mnemonic		Code
RETD	d16	0 0 0 0 1 1 1 1
		d<7:0>
		d<15:8>

Flags :

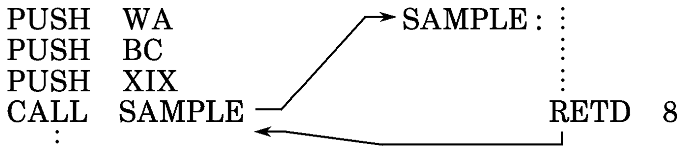
S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change

Execution example: RETD 8

When the stack pointer XSP = 0FCH and the contents of memory at address 0FCH = 9000H (long word data) in minimum mode, execution sets the stack pointer XSP to 0FCH + 4 + 8 → 108H and jumps (returns) to address 9000H.

Usage of the RETD instruction is shown below. In this example, the 8-bit parameter is pushed to the stack before the subroutine call. After the subroutine processing complete, the used parameter area is deleted by the RETD instruction.



```
PUSH  WA            SAMPLE : :
PUSH  BC                     :
PUSH  XIX                    :
CALL  SAMPLE                 RETD  8
  :
```

RETI

< Return from Interrupt >

Operation : SR ← (XSP), 32-bit PC ← (XSP + 2), XSP ← XSP + 6

After the above operation is executed, the 900/H decrement a value of interrupt nesting counter INTNEST by 1.

Description : Pops data from the stack area to status register and program counter.

After the above operation is executed, the 900/H decrement a value of interrupt nesting counter INTNEST by 1.

Details :

Mnemonic	Code
RETI	0 0 0 0 0 1 1 1

Flags :

S	Z	H	V	N	C
*	*	*	*	*	*

S = The value popped from the stack area is set.
Z = The value popped from the stack area is set.
H = The value popped from the stack area is set.
V = The value popped from the stack area is set.
N = The value popped from the stack area is set.
C = The value popped from the stack area is set.

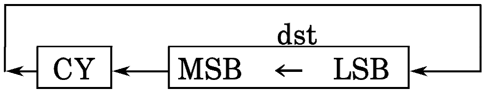
RL num, dst

< Rotate Left >

Operation : {CY & dst ← left rotates the value of CY & dst} Repeat num

Description : Rotates left the contents of the linked carry flag and dst.
Repeats the number of times specified in num.

Description figure:



Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	○	RL	#4, r	1 1 z z 1 r 1 1 1 0 1 0 1 0 0 0 0 0 #4
○	○	○	RL	A, r	1 1 z z 1 r 1 1 1 1 1 0 1 0
○	○	×	RL <W>	(mem)	1 m 0 z m m m m 0 1 1 1 1 0 1 0

Note : When the number of rotates is specified by the A register, the value of the lower 4 bits of the A register is used. Specifying 0 rotates 16 times.
When dst is memory, rotating is performed only once.

Flags :

S	Z	H	V	N	C
*	*	0	*	0	*

S = MSB value of dst after rotate is set.
Z = 1 is set when the contents of dst after rotate is 0, otherwise 0.
H = Reset to 0.
V = 1 is set when the parity (number of 1s) of dst is even after rotate, otherwise 0. If the operand is 32 bits, an undefined value is set.
N = Reset to 0.
C = The value after rotate is set.

Execution example: RL 4, HL
When the HL register = 6230H and the carry flag = 1, execution sets the HL register to 230BH and the carry flag to 0.

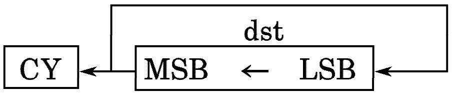
RLC num, dst

< Rotate Left without Carry >

Operation : {CY← dst <MSB>, dst← left rotate value of dst} Repeat num

Description : Loads the contents of the MSB of dst to the carry flag and rotates left the contents of dst. Repeats the number of times specified in num.

Description figure :



Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	○	RLC	#4, r	1 1 z z 1 r 1 1 1 0 1 0 0 0 0 0 0 0 # 4
○	○	○	RLC	A, r	1 1 z z 1 r 1 1 1 1 1 0 0 0
○	○	×	RLC <W>	(mem)	1 m 0 z m m m m 0 1 1 1 1 0 0 0

Note : When the number of rotates is specified by the A register, the value of the lower 4 bits of the A register is used. Specifying 0 rotates 16 times.
When dst is memory, rotating is performed only once.

Flags :

S	Z	H	V	N	C
*	*	0	*	0	*

S = MSB value of dst after rotate is set.
Z = 1 is set when the contents of dst after rotate is 0, otherwise, 0.
H = Reset to 0.
V = 1 is set when the parity (number of 1s) of dst is even after rotate. If the operand is 32 bits, an undefined value is set.
N = Reset to 0.
C = MSB value of dst before the last rotate is set.

Execution example: RLC 4, HL
When the HL register = 1230H, execution sets the HL register to 2301H and the carry flag to 1.

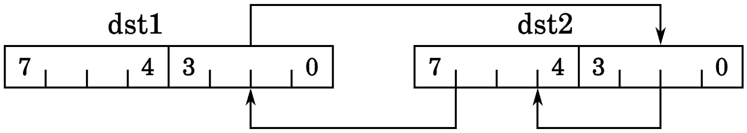
RLD dst1, dst2

< Rotate Left Digit >

Operation : dst1<3:0>←dst2<7:4>, dst2<7:4>←dst2<3:0>,
dst2<3:0>←dst1 <3:0>

Description : Rotates left the lower 4 bits of dst1 and the contents of dst2 in units of 4 bits.

Description figure :



Details :

Size Byte	 Word	 Long word	Mnemonic		Code
○	×	×	RLD	[A,] (mem)	1 m 0 0 m m m m 0 0 0 0 0 1 1 0

Flags :

S	Z	H	V	N	C
*	*	0	*	0	–

S = MSB value of the A register after rotate is set.
Z = 1 is set when the contents of the A register after the rotate are 0, otherwise 0.
H = Reset to 0.
V = 1 is set when the parity (number of 1s) of the A register is even after the rotate, otherwise 0.
N = Reset to 0.
C = No change

Execution example: RLD A, (100H)
When the A register = 12H and the contents of memory at address 100H = 34H, execution sets the A register to 13H and the contents of memory at address 100H to 42H.

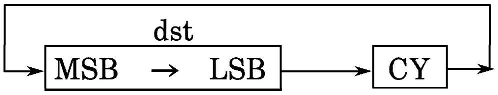
RR num, dst

< Rotate Right >

Operation : {CY & dst ← right rotates the value of CY & dst} Repeat num

Description : Rotates right the linked contents of the carry flag and dst.
Repeats the number of times specified in num.

Description figure:



Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	○	RR	#4, r	1 1 z z 1 r 1 1 1 0 1 0 1 1 0 0 0 0 # 4
○	○	○	RR	A, r	1 1 z z 1 r 1 1 1 1 1 0 1 1
○	○	×	RR <W>	(mem)	1 m 0 z m m m m 0 1 1 1 1 0 1 1

Note : When the number of rotates is specified by the A register, the value of the lower 4 bits of the A register is used. Specifying 0 rotates 16 times.
When dst is memory, rotating is performed only once.

Flags :

S	Z	H	V	N	C
*	*	0	*	0	*

S = MSB value of dst after rotate is set.
Z = 1 is set when the contents of dst after rotate is 0, otherwise 0.
H = Reset to 0.
V = 1 is set when the parity (number of 1s) of dst is even after the rotate, otherwise 0. If the operand is 32 bits, an undefined value is set.
N = Reset to 0.
C = The value after rotate is set.

Execution example: RR 4, HL
When the HL register = 6230H and the carry flag = 1, execution sets the HL register to 1623H and the carry flag to 0.

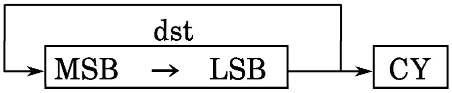
RRC num, dst

< Rotate Right without Carry >

Operation : {CY← dst <LSB>, dst ← right rotate value of dst} Repeat num

Description : Loads the contents of the LSB of dst to the carry flag and rotates the contents of dst to the right. Repeats the number of times specified in num.

Description figure:



Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	○	RRC	#4, r	1 1 z z 1 r 1 1 1 0 1 0 0 1 0 0 0 0 # 4
○	○	○	RRC	A, r	1 1 z z 1 r 1 1 1 1 1 0 0 1
○	○	×	RRC <W>	(mem)	1 m 0 z m m m m 0 1 1 1 1 0 0 1

Note : When the number of rotates num is specified by the A register, the value of the lower 4 bits of the A register is used as the number of rotates.
Specifying 0 rotates 16 times. When dst is memory, rotating is only once.

Flags :

S	Z	H	V	N	C
*	*	0	*	0	*

S = MSB value of dst after rotate is set.
Z = 1 is set when the contents of dst after rotate is 0, otherwise 0.
H = Reset to 0.
V = 1 is set when the parity (number of 1s) of dst is even after rotate, otherwise 0. If the operand is 32 bits, an undefined value is set.
N = Reset to 0.
C = MSB value of dst before the last rotate is set.

Execution example: RRC 4, HL
When the HL register = 1230H, execution sets the HL register to 0123H and the carry flag to 0.

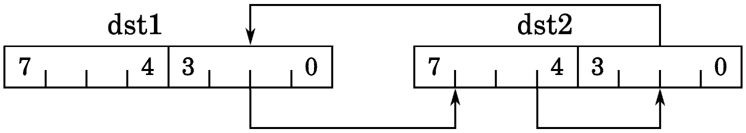
RRD dst1, dst2

< Rotate Right Digit >

Operation : dst1<3:0>←dst2<3:0>, dst2<7:4>←dst1<3:0>,
dst2<3:0>←dst2<7:4>

Description : Rotates right the lower 4 bits of dst1 and the contents of dst2 in units of 4 bits.

Description figure :



Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	×	×	RRD	[A,] (mem)	1 m 0 0 m m m m 0 0 0 0 0 1 1 1

Flags :

S	Z	H	V	N	C
*	*	0	*	0	–

S = MSB value of the A register after rotate is set.
Z = 1 is set when the contents of the A register after rotate is 0, otherwise 0.
H = Reset to 0.
V = 1 is set when the parity (number of 1s) of the A register is even after rotate, otherwise 0.
N = Reset to 0.
C = No change

Execution example: RRD A, (100H)

When the A register = 12H and the contents of memory at address 100H = 34H, execution sets the A register to 14H and the contents of memory at address 100H to 23H.

SBC dst, src

< Subtract with Carry >

Operation : dst ← dst − src − CY

Description : Subtracts the contents of src and the carry flag from those of dst, and loads the result to dst.

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	○	SBC	R, r	1 1 z z 1 r 1 0 1 1 0 R
○	○	○	SBC	r, #	1 1 z z 1 r 1 1 0 0 1 0 1 1 #<7:0> #<15:8> #<23:16> #<31:24>
○	○	○	SBC	R, (mem)	1 m z z m m m m 1 0 1 1 0 R
○	○	○	SBC	(mem), R	1 m z z m m m m 1 0 1 1 1 R
○	○	×	SBC<W>	(mem), #	1 m 0 z m m m m 0 0 1 1 1 0 1 1 #<7:0> #<15:8>

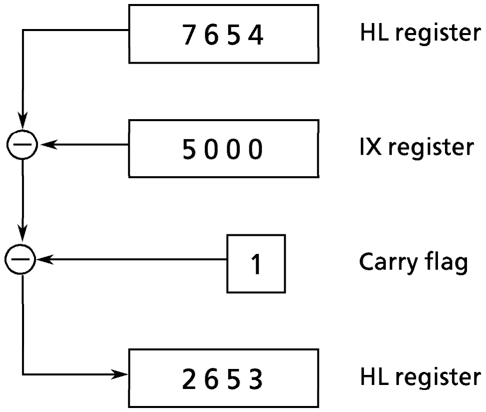
Flags :

S	Z	H	V	N	C
*	*	*	*	1	*

S = MSB value of the result is set.
Z = 1 is set when the result is 0, otherwise 0.
H = 1 is set when a borrow from bit 3 to bit 4 occurs as a result, otherwise 0. When the operand is 32 bits, an undefined value is set.
V = 1 is set when an overflow occurs as a result, otherwise 0.
N = 1 is set.
C = 1 is set when a borrow from the MSB occurs as a result, otherwise 0.

Execution example: SBC HL, IX

When the HL register is 7654H, the IX register = 5000H, and the carry flag = 1, execution sets the HL register to 2653H.



SCC condition, dst

< Set Condition Code >

Operation : If cc is true, then dst ← 1 else dst ← 0.

Description : Loads 1 to dst when the operand condition is true; when false, 0 is loaded to dst.

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	×	SCC	cc, r	1 1 0 z 1 r 0 1 1 1 c c

Flags :

S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change

Execution example: SCC OV, HL
When the contents of the V flag = 1, execution sets the HL register to 0001H.

SCF

< Set Carry Flag >

Operation : CY ← 1

Description : Sets the carry flag to 1.

Details :

Mnemonic	Code
SCF	0 0 0 1 0 0 0 1

Flags :

S	Z	H	V	N	C
–	–	0	–	0	1

S = No change
Z = No change
H = Reset to 0.
V = No change
N = Reset to 0.
C = Set to 1.

SET num, dst

< Set >

Operation : dst <num> ← 1

Description : Sets bit num of dst to 1.

Details :

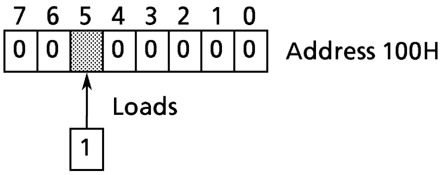
Size			Mnemonic		Code
Byte	Word	Long word			
○	○	×	SET	#4, r	1 1 0 z 1 r 0 0 1 1 0 0 0 1 0 0 0 0 # 4
○	×	×	SET	#3, (mem)	1 m 1 1 m m m m 1 0 1 1 1 #3

Flags :

S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change

Execution example: SET 5, (100H)
When the contents of memory at address 100H = 00000000B (binary), execution sets the contents of memory at address 100H to 00100000B (binary).



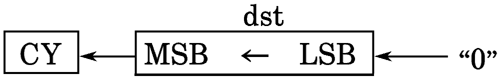
SLA num, dst

< Shift Left Arithmetic >

Operation : {CY ← dst<MSB>, dst ← left shift value of dst, dst<LSB> ← 0} Repeat num

Description : Loads the contents of the MSB of dst to the carry flag, shifts left the contents of dst, and loads 0 to the LSB of dst. Repeats the number of times specified in num.

Description chart:



Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	○	SLA	#4, r	1 1 z z 1 r 1 1 1 0 1 1 0 0 0 0 0 0 #4
○	○	○	SLA	A, r	1 1 z z 1 r 1 1 1 1 1 1 0 0
○	○	×	SLA <W>	(mem)	1 m 0 z m m m m 0 1 1 1 1 1 0 0

Note : When the number of shifts, num, is specified by the A register, the value of the lower 4 bits of the A register is used. Specifying 0 shifts 16 times. When dst is memory, shifting is performed only once.

Flags :

S	Z	H	V	N	C
*	*	0	*	0	*

S = MSB value of dst after shift is set.
Z = 1 is set when the contents of dst after shift is 0, otherwise 0.
H = Reset to 0.
V = 1 is set when the parity (number of 1s) of dst is even after shifting, otherwise 0. If the operand is 32 bits, an undefined value is set.
N = Reset to 0.
C = MSB value of dst before the last shift is set.

Execution example: SLA 4, HL
When the HL register = 1234H, execution sets the HL register to 2340H and the carry flag to 1.

SLL num, dst

< Shift Left Logical >

Operation : {CY ← dst<MSB>, dst ← left shift value of dst, dst<LSB> ← 0} Repeat num

Description : Loads the contents of the MSB of dst to the carry flag, shifts left the contents of dst, and loads 0 to the MSB of dst. Repeats the number of times specified in num.

Description chart: $\boxed{\text{CY}} \longleftarrow \overset{\text{dst}}{\boxed{\text{MSB} \leftarrow \text{LSB}}} \longleftarrow \text{“0”}$

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	○	SLL	#4, r	1 1 z z 1 r 1 1 1 0 1 1 1 0 0 0 0 0 # 4
○	○	○	SLL	A, r	1 1 z z 1 r 1 1 1 1 1 1 1 0
○	○	×	SLL <W>	(mem)	1 m 0 z m m m m 0 1 1 1 1 1 1 0

Note : When the number of shifts, num, is specified by the A register, the value of the lower 4 bits of the A register is used. Specifying 0 shifts 16 times. When dst is memory, shifting is performed only once.

Flags :

S	Z	H	V	N	C
*	*	0	*	0	*

S = MSB value of dst after shift is set.
Z = 1 is set when the contents of dst after shift is 0, otherwise 0.
H = Reset to 0.
V = 1 is set when the parity (number of 1s) of dst is even after shifting, otherwise 0. If the operand is 32 bits, an undefined value is set.
N = Reset to 0.
C = MSB value of dst before the last shift is set.

Execution example: SLL 4, HL
When the HL register = 1234H, execution sets the HL register to 2340H and the carry flag to 1.

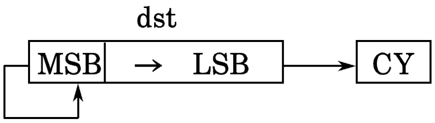
SRA num, dst

< Shift Right Arithmetic >

Operation : {CY ← dst<MSB>, dst ← right shift value of dst, dst <MSB> is fixed}
Repeat num

Description : Loads the contents of the LSB of dst to the carry flag and shifts right the contents of dst (MSB is fixed). Repeats the number of times specified in num.

Description chart:



Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	○	SRA	#4, r	1 1 z z 1 r 1 1 1 0 1 1 0 1 0 0 0 0 # 4
○	○	○	SRA	A, r	1 1 z z 1 r 1 1 1 1 1 1 0 1
○	○	×	SRA <W>	(mem)	1 m 0 z m m m m 0 1 1 1 1 1 0 1

Note : When the number of shifts, num, is specified by the A register, the value of the lower 4 bits of the A register is used. Specifying 0 shifts 16 times. When dst is memory, shifting is performed only once.

Flags :

S	Z	H	V	N	C
*	*	0	*	0	*

S = MSB value of dst after shift is set.
Z = 1 is set when the contents of dst after shift is 0, otherwise 0.
H = Reset to 0.
V = 1 is set when the parity (number of 1s) of dst is even after shift, otherwise 0.
If the operand is 32 bits, an undefined value is set.
N = Reset to 0.
C = LSB value of dst before the last shift is set.

Execution example: SRA 4, HL
When the HL register = 8230H, execution sets the HL register to F823H and the carry flag to 0.

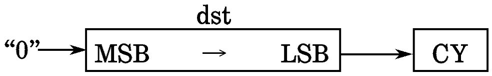
SRL num, dst

< Shift Right Logical >

Operation : {CY ← dst<LSB>, dst ← right shift value of dst, dst <MSB> ← 0} Repeat num

Description : Loads the contents of the LSB of dst to the carry flag, shifts right the contents of dst, and loads 0 to the MSB of dst. Repeats the number of times specified in num.

Description chart :



Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	○	SRL	#4, r	1 1 z z 1 r
					1 1 1 0 1 1 1 1
					0 0 0 0 #4
○	○	○	SRL	A, r	1 1 z z 1 r
					1 1 1 1 1 1 1 1
○	○	×	SRL<W>	(mem)	1 m 0 z m m m m
					0 1 1 1 1 1 1 1

Note : When the number of shifts, num, is specified by the A register, the value of the lower 4 bits of the A register is used. Specifying 0 shifts 16 times. When dst is memory, shifting is performed only once.

Flags :

S	Z	H	V	N	C
*	*	0	*	0	*

S = MSB value of dst after shift is set.
Z = 1 is set when the contents of dst after shift is 0, otherwise 0.
H = Reset to 0.
V = 1 is set when the parity (number of 1s) of dst is even after shift, otherwise 0. If the operand is 32 bits, an undefined value is set.
N = Reset to 0.
C = LSB value of dst before the last shift is set.

Execution example: SRL 4, HL

When the HL register = 1238H, execution sets the HL register to 0123H and the carry flag to 1.

STCF num, dst

< Store Carry Flag >

Operation : dst<num>←CY

Description : Loads the contents of the carry flag to bit num of dst.

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	×	STCF	#4, r	1 1 0 z 1 r 0 0 1 0 0 1 0 0 0 0 0 0 #4
○	○	×	STCF	A, r	1 1 0 z 1 r 0 0 1 0 1 1 0 0
○	×	×	STCF	#3, (mem)	1 m 1 1 m m m m 1 0 1 0 0 #3
○	×	×	STCF	A, (mem)	1 m 1 1 m m m m 0 0 1 0 1 1 0 0

Note : When bit num is specified by the A register, the value of the lower 4 bits of the A register is used. When the operand is a byte and the value of the lower 4 bits of bit num is from 8 to 15, the operand value does not change.

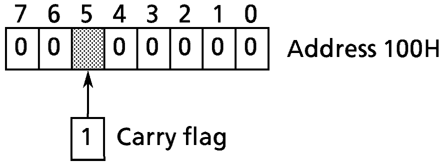
Flags :

S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change

Execution example: STCF 5, (100H)

When the contents of memory at address 100H = 00H and the carry flag = 1, execution sets the contents of memory at address 100H to 00100000B (binary).



SUB dst, src

< Subtract >

Operation : dst←dst−src

Description : Subtracts the contents of src from those of dst and loads the result to dst.

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
○	○	○	SUB	R, r	1 1 z z 1 r 1 0 1 0 0 R
○	○	○	SUB	r, #	1 1 z z 1 r 1 1 0 0 1 0 1 0 #<7:0> #<15:8> #<23:16> #<31:24>
○	○	○	SUB	R, (mem)	1 m z z m m m m 1 0 1 0 0 R
○	○	○	SUB	(mem), R	1 m z z m m m m 1 0 1 0 1 R
○	○	×	SUB<W>	(mem), #	1 m 0 z m m m m 0 0 1 1 1 0 1 0 #<7:0> #<15:8>

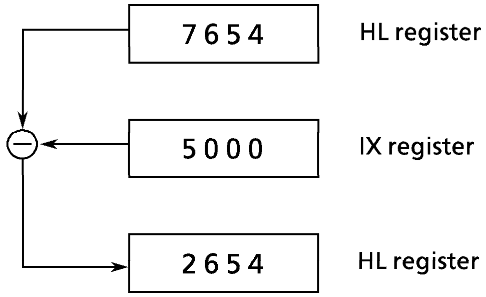
Flags :

S	Z	H	V	N	C
*	*	*	*	1	*

S = MSB value of the result is set.
Z = 1 is set when the result is 0, otherwise 0.
H = 1 is set when a borrow from bit 3 to bit 4 occurs as a result, otherwise 0. When the operand is 32 bits, an undefined value is set.
V = 1 is set when an overflow occurs as a result, otherwise 0.
N = 1 is set.
C = 1 is set when a borrow from MSB occurs as a result, otherwise 0.

Execution example: SUB HL, IX

When the HL register = 7654H and the IX register = 5000H, execution sets the HL register to 2654H.



SWI num

< Software Interrupt >

Operation : 1) XSP←XSP−6
2) (XSP)←SR
3) (XSP+2)←32 bit PC
4) PC←(Address refer to vector+num×4)
Note : Address refer to vector is defined for each product.

Description : Saves to the stack area the contents of the status register and contents of the program counter which indicate the address next to the SWI instruction. Finally, jumps to vector is indicated address refer to vector.

Details :

Mnemonic		Code					
SWI	[#3]	1	1	1	1	1	#3

Note 1 : A value from 0 to 7 can be specified as the operand value. When the operand coding is omitted, SWI 7 is assumed.

Note 2 : The status register structure is as shown below.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSM	IFF2	IFF1	IFF0	MAX	RFP2	RFP1	RFP0	S	Z	“0”	H	“0”	V	N	C

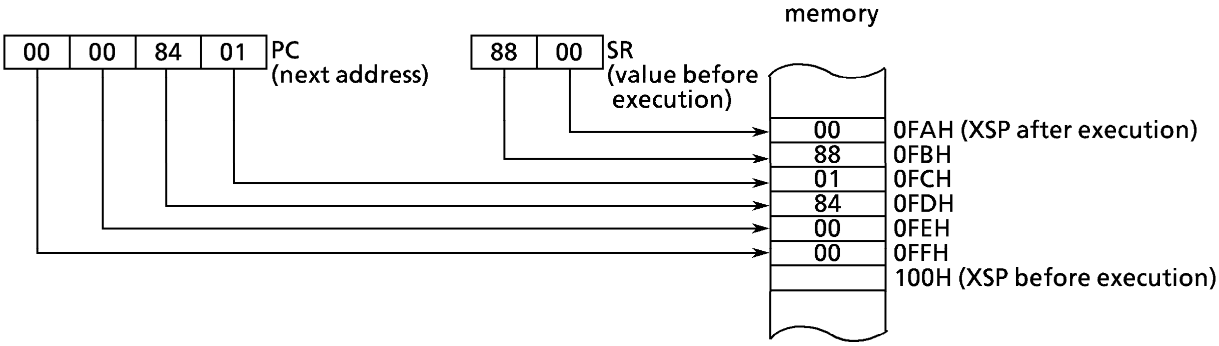
Flags :

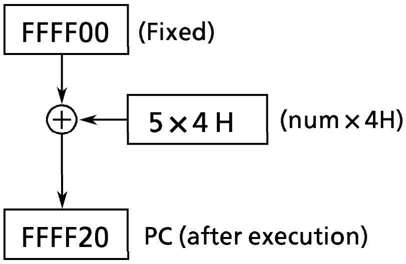
S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change

Execution example: SWI 5

When the stack pointer XSP = 100H, the status register = 8800H, executing the above instruction at memory address 8400H writes the contents of the previous status register 8800H in memory address 00FAH, and the contents of the program counter 00008401H in memory address 00FCH, then jumps to address FFFF20H.





TSET num, dst

< Test and Set >

Operation : Z flag ← inverted value of dst <num>
dst <num> ← 1

Description : Loads the inverted value of the bit num of dst to the Z flag.
Then the bit num of dst is set to "1".

Details :

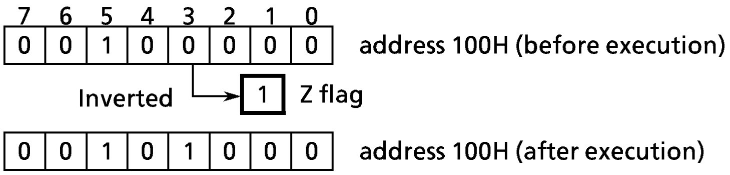
Size			Mnemonic		Code
byte	word	long word			
○	○	×	TSET	#4, r	1 1 z z 1 r 0 0 1 1 0 1 0 0 0 0 0 0 # 4
○	×	×	TSET	#3, (mem)	1 m 1 1 m m m m 1 0 1 0 1 # 3

Flags :

S	Z	H	V	N	C
×	*	1	×	0	–

S = An undefined value is set.
Z = The inverted value of the src <num> is set.
H = Set to 1
V = An undefined value is set.
N = Set to 0
C = No change

Execution example: When the contents of memory at address 100H = 00100000B (binary), TSET 3, (100H) execution sets the Z flag to 1, the contents of memory at address 100H = 00101000B (binary).


7 6 5 4 3 2 1 0
0 0 1 0 0 0 0 0
address 100H (before execution)
Inverted
1
Z flag
0 0 1 0 1 0 0 0
address 100H (after execution)

UNLK dst

< Unlink >

Operation : XSP ← dst, dst ← (XSP+)

Description : Loads the contents of dst to the stack pointer XSP, then pops long word data from the stack area to dst. Used paired with the Link instruction.

Details :

Size			Mnemonic		Code
Byte	Word	Long word			
×	×	○	UNLK	r	1 1 1 0 1 r 0 0 0 0 1 1 0 1

Flags :

S	Z	H	V	N	C
–	–	–	–	–	–

S = No change
Z = No change
H = No change
V = No change
N = No change
C = No change

Execution example: UNLK XIZ

As a result of executing this instruction after executing the Link instruction, the stack pointer XSP and the XIZ register revert to the same values they had before the Link instruction was executed. (For details of the Link instruction, see page 100)

XOR dst, src

< Exclusive OR >

Operation : dst←dst XOR src

Description : Exclusive ors the contents of dst with those of src and loads the result to dst.

(Truth table)

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Details :

	Size		Mnemonic		Code
Byte	Word	Long word			
○	○	○	XOR	R, r	1 1 z z 1 r 1 1 0 1 0 R
○	○	○	XOR	r, #	1 1 z z 1 r 1 1 0 0 1 1 0 1 #<7:0> #<15:8> #<23:16> #<31:24>
○	○	○	XOR	R, (mem)	1 m z z m m m m 1 1 0 1 0 R
○	○	○	XOR	(mem), R	1 m z z m m m m 1 1 0 1 1 R
○	○	×	XOR<W>	(mem), #	1 m 0 z m m m m 0 0 1 1 1 1 0 1 #<7:0> #<15:8>

Flags :

S	Z	H	V	N	C
*	*	0	*	0	0

S = MSB value of the result is set.
Z = 1 is set when the result is 0, otherwise 0.
H = Reset to 0.
V = 1 is set when the parity (number of 1s) of dst is even as a result, otherwise 0. If the operand is 32 bits, an undefined value is set.
N = Cleared to 0.
C = Cleared to 0.

Execution example: XOR HL, IX
When the HL register = 7350H and the IX register = 3456H, execution sets the HL register to 4706H.

```
        0111  0011  0101  0000  ←  HL register (before execution)
XOR)    0011  0100  0101  0110  ←  IX register (before execution)
——————————————————————————————
        0100  0111  0000  0110  ←  HL register (after execution)
```

XORCF num, src

< Exclusive OR Carry Flag >

Operation : CY ← CY XOR src<num>

Description : Exclusive ors the contents of the carry flag and bit num of src, and loads the result to the carry flag.

Details :

Size Byte	 Word	 Long word	Mnemonic		Code
○	○	×	XORCF	#4, r	1 1 0 z 1 r 0 0 1 0 0 0 1 0 0 0 0 0 # 4
○	○	×	XORCF	A, r	1 1 0 z 1 r 0 0 1 0 1 0 1 0
○	×	×	XORCF	#3, (mem)	1 m 1 1 m m m m 1 0 0 1 0 #3
○	×	×	XORCF	A, (mem)	1 m 1 1 m m m m 0 0 1 0 1 0 1 0

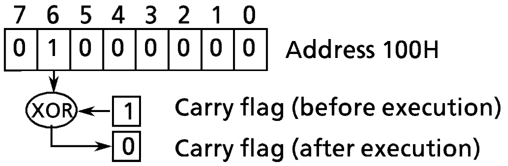
Note : When bit num is specified by the A register, the value of the lower 4 bits of the A register is used. When the operand is a byte and the value of the lower 4 bits of bit num is from 8 to 15, the result is undefined.

Flags :

S	Z	H	V	N	C
–	–	–	–	–	*

S = No change
Z = No change
H = No change
V = No change
N = No change
C = The value obtained by exclusive or-ing the contents of the carry flag with those of bit num of src is set.

Execution example: XORCF 6, (100H)
When the contents of memory at address 100H = 01000000B (binary) and the carry flag = 1, execution sets the carry flag to 0.



ZCF

< Zero flag to Carry Flag >

Operation : CY ← inverted value of Z flag

Description : Loads the inverted value of the Z flag to the carry flag.

Details :

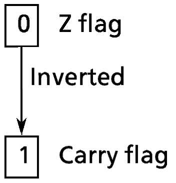
Mnemonic	Code
ZCF	0 0 0 1 0 0 1 1

Flags :

S	Z	H	V	N	C
–	–	×	–	0	*

S = No change
Z = No change
H = An undefined value is set.
V = No change
N = Reset to 0.
C = The inverted value of the Z flag is set.

Execution example: ZCF
When the Z flag = 0, execution sets the carry flag to 1.


0
Z flag
Inverted
1
Carry flag