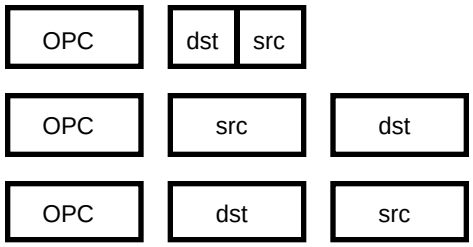
12.6 INSTRUCTION DESCRIPTION AND FORMATS

ADC
ADD WITH CARRY

ADC
Add With Carry

ADC dst, src

Instruction Format:

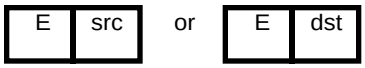


Cycles	OPC (Hex)	Address dst	Mode src
6	12	r	r
6	13	r	Ir
10	14	R	R
10	15	R	IR
10	16	R	IM
10	17	IR	IM

Operation: dst <— dst + src + C
The source operand, along with the setting of the Carry (C) Flag, is added to the destination operand. Two's complement addition is performed. The sum is stored in the destination operand. The contents of the source operand are not affected. In multiple precision arithmetic, this instruction permits the carry from the addition of low order operands to be carried into the addition of high order operands.

Flags:
C: Set if there is a carry from the most significant bit of the result; cleared otherwise.
Z: Set if the result is zero; cleared otherwise.
S: Set if the result is negative; cleared otherwise.
V: Set if an arithmetic overflow occurs, that is, if both operands are of the same sign and the result is of the opposite sign; cleared otherwise.
D: Always cleared.
H: Set if there is a carry from the most significant bit of the low order four bits of the result; cleared otherwise.

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the source or destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Working Register R3 contains 16H, the C Flag is set to 1, and Working Register R11 contains 20H, the statement:

ADC R3, R11
OpCode: 12 3B

leaves the value 37H in Working Register R3. The C, Z, S, V, D, and H Flags are all cleared.

ADC
ADD WITH CARRY

Example: If Working Register R16 contains 16H, the C Flag is not set, Working Register R10 contains 20H, and Register 20H contains 11H, the statement:

ADC R16, @R10
OpCode: 13 FA

leaves the value 27H in Working Register R16. The C, Z, S, V, D, and H Flags are all cleared.

Example: If Register 34H contains 2EH, the C Flag is set, and Register 12H contains 1BH, the statement:

ADC 34H, 12H
OpCode: 14 12 34

leaves the value 4AH in Register 34H. The H Flag is set, and the C, Z, S, V, and D Flags are cleared.

Example: If Register 4BH contains 82H, the C Flag is set, Working Register R3 contains 10H, and Register 10H contains 01H, the statement:

ADC 4BH, @R3
OpCode: 15 E3 4B

leaves the value 84H in Register 4BH. The S Flag is set, and the C, Z, V, D, and H Flags are cleared.

Example: If Register 6CH contains 2AH, and the C Flag is not set, the statement:

ADC 6CH, #03H
OpCode: 16 6C 03

leaves the value 2DH in Register 6CH. The C, Z, S, V, D, and H Flags are all cleared.

Example: If Register D4H contains 5FH, Register 5FH contains 4CH, and the C Flag is set, the statement:

ADC @D4H, #02H
OpCode: 17 D4 02

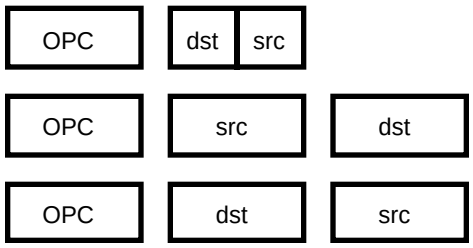
leaves the value 4FH in Register 5FH. The C, Z, S, V, D, and H Flags are all cleared.

ADD
ADD

ADD
Add:

ADD dst, src:

Instruction Format:



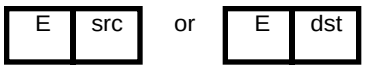
Cycles	OPC (Hex)	Address dst	Mode src
6	02	r	r
6	03	r	Ir
10	04	R	R
10	05	R	IR
10	06	R	IM
10	07	IR	IM

Operation: dst <— dst + src

The source operand is added to the destination operand. Two's complement addition is performed. The sum is stored in the destination operand. The contents of the source operand are not affected.

Flags:
- C: Set if there is a carry from the most significant bit of the result; cleared otherwise.
- Z: Set if the result is zero; cleared otherwise.
- S: Set if the result is negative; cleared otherwise.
- V: Set if an arithmetic overflow occurs, that is, if both operands are of the same sign and the result is of the opposite sign; cleared otherwise.
- D: Always cleared.
- H: Set if there is a carry from the most significant bit of the low order four bits of the result; cleared otherwise.

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the source or destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Working Register R3 contains 16H and Working Register R11 contains 20H, the statement:

ADD R3, R11
OpCode: 02 3B

leaves the value 36H in Working Register R3. The C, Z, S, V, D, and H Flags are all cleared.

ADD
ADD

Example: If Working Register R16 contains 16H, Working Register R10 contains 20H, and Register 20H contains 11H, the statement:

ADD R16, @R10
OpCode: 03 FA

leaves the value 27H in Working Register R16. The C, Z, S, V, D, and H Flags are all cleared.

Example: If Register 34H contains 2EH and Register 12H contains 1BH, the statement:

ADD 34H, 12H
OpCode: 04 12 34

leaves the value 49H in Register 34H. The H Flag is set, and the C, Z, S, V, and D Flags are cleared.

Example: If Register 4BH contains 82H, Working Register R3 contains 10H, and Register 10H contains 01H, the statement:

ADD 3EH, @R3
OpCode: 05 E3 4B

leaves the value 83H in Register 4BH. The S Flag is set, and the C, Z, V, D, and H Flags are cleared.

Example: If Register 6CH contains 2AH, the statement:

ADD 6CH, #03H
OpCode: 06 6C 03

leaves the value 2DH in Register 6CH. The C, Z, S, V, D, and H Flags are all cleared.

Example: If Register D4H contains 5FH and Register 5FH contains 4CH, the statement:

ADD @D4H, #02H
OpCode: 07 D4 02

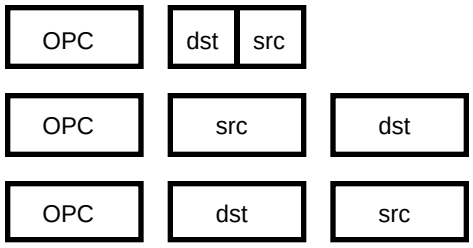
leaves the value 4EH in Register 5FH. The C, Z, S, V, D, and H Flags are all cleared.

AND
Logical AND

AND
Logical AND

AND dst, src

Instruction Format:



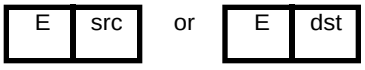
Cycles	OPC (Hex)	Address dst	Mode src
6	52	r	r
6	53	r	Ir
10	54	R	R
10	55	R	IR
10	56	R	IM
10	57	IR	IM

Operation: dst <— dst AND src

The source operand is logically ANDed with the destination operand. The AND operation results in a 1 being stored whenever the corresponding bits in the two operands are both 1, otherwise a 0 is stored. The result is stored in the destination operand. The contents of the source bit are not affected.

Flags:
C: Unaffected
Z: Set if the result is zero; cleared otherwise
S: Set if the result of bit 7 is set; cleared otherwise
V: Always reset to 0
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the source or destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Working Register R1 contains 34H (00111000B) and Working Register R14 contains 4DH (10001101), the statement:

AND R1, R14
OpCode: 52 1E

leaves the value 04H (00001000) in Working Register R1. The Z, V, and S Flags are cleared.

AND
Logical AND

Example: If Working Register R4 contains F9H (11111001B), Working Register R13 contains 7BH, and Register 7BH contains 6AH (01101010B), the statement:

AND R4, @R13
OpCode: 53 4D

leaves the value 68H (01101000B) in Working Register R4. The Z, V, and S Flags are cleared.

Example: If Register 3AH contains the value F5H (11110101B) and Register 42H contains the value 0AH (00001010), the statement:

AND 3AH, 42H
OpCode: 54 42 3A

leaves the value 00H (00000000B) in Register 3AH. The Z Flag is set, and the V and S Flags are cleared.

Example: If Working Register R5 contains F0H (11110000B), Register 45H contains 3AH, and Register 3AH contains 7FH (01111111B), the statement:

AND R5, @45H
OpCode: 55 45 E5

leaves the value 70H (01110000B) in Working Register R5. The Z, V, and S Flags are cleared.

Example: If Register 7AH contains the value F7H (11110111B), the statement:

AND 7AH, #F0H
OpCode: 56 7A F0

leaves the value F0H (11110000B) in Register 7AH. The S Flag is set, and the Z and V Flags are cleared.

Example: If Working Register R3 contains the value 3EH and Register 3EH contains the value ECH (11101100B), the statement:

AND @R3, #05H
OpCode: 57 E3 05

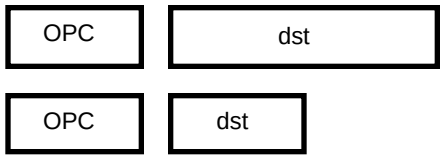
leaves the value 04H (00000100B) in Register 3EH. The Z, V, and S Flags are cleared.

CALL
CALL PROCEDURE

CALL
Call Procedure

CALL dst

Instruction Format:



Cycles	OPC (Hex)	Address Mode dst
20	D6	DA
20	D4	IRR

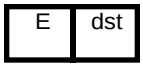
Operation: SP <— SP - 2
@SP <— PC
PC <— dst

The Stack pointer is decremented by two, the current contents of the Program Counter (PC) (address of the first instruction following the CALL instruction) are pushed onto the top of the Stack, and the specified destination address is then loaded into the PC. The PC now points to the first instruction of the procedure.

At the end of the procedure a RET (return) instruction can be used to return to the original program flow. RET will pop the top of the Stack and replace the original value into the PC.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Note: Address mode IRR can be used to specify a 4-bit Working Register Pair. In this format, the destination Working Register Pair operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register Pair RR12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



CALL
Call Procedure

Example: If the contents of the PC are 1A47H and the contents of the SP (Registers FEH and FFH) are 3002H, the statement:

CALL 3521H
OpCode: D6 35 21

causes the SP to be decremented to 3000H, 1A4AH (the address following the CALL instruction) to be stored in external data memory 3000 and 3001H, and the PC to be loaded with 3521H. The PC now points to the address of the first statement in the procedure to be executed.

Example: If the contents of the PC are 1A47H, the contents of the SP (Register FFH) are 72H, the contents of Register A4H are 34H, and the contents of Register Pair 34H are 3521H, the statement:

CALL @A4H
OpCode: D4 A4

causes the SP to be decremented to 70H, 1A4AH (the address following the CALL instruction) to be stored in R70H and 71H, and the PC to be loaded with 3521H. The PC now points to the address of the first statement in the procedure to be executed.

CCF
COMPLEMENT CARRY FLAG

CCF
Complement Carry Flag

CCF

Instruction Format:

	Cycles	OPC (Hex)
OPC	6	EF

Operation: C <— NOT C

The C Flag is complemented. If C = 1, then it is changed to C = 0; or, if C = 0, then it is changed to C = 1.

Flags:
C: Complemented
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Example: If the C Flag contains a 0, the statement:

CCF
OpCode: EF

will change the C Flag from C = 0 to C = 1.

CLR
CLEAR

CLR
CLEAR

CLR dst

Instruction Format:

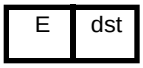
Cycles	OPC (Hex)	Address Mode dst
6	B0	R
6	B1	IR

Operation: dst <— 0

The destination operand is cleared to 00H.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Working Register R6 contains AFH, the statement:

CLR R6
OpCode: B0 E6

will leave the value 00H in Working Register R6.

If Register A5H contains the value 23H, and Register 23H contains the value FCH, the statement:

CLR @A5H
OpCode: B1 A5

will leave the value 00H in Register 23H.

COM
COMPLEMENT

COM
Complement

COM dst

Instruction Format:

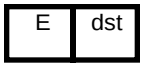
Cycles	OPC (Hex)	Address Mode dst
6	60	R
6	61	IR

Operation: dst <— NOT dst

The contents of the destination operand are complemented (one's complement). All 1 bits are changed to 0, and all 0 bits are changed to 1.

Flags:
C: Unaffected
Z: Set if the result is zero; cleared otherwise.
S: Set if result bit 7 is set; cleared otherwise.
V: Always reset to 0.
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Register 08H contains 24H (00100100B), the statement:

COM 08H
OpCode: 60 08

leaves the value DBH (11011011) in Register 08H. The S Flag is set, and the Z and V Flags are cleared.

Example: If Register 08H contains 24H, and Register 24H contains FFH (11111111B), the statement:

COM @08H
OpCode: 61 08

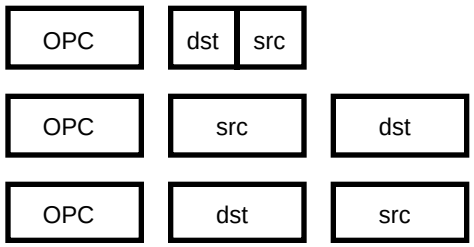
leaves the value 00H (00000000B) in Register 24H. The Z Flag is set, and the V and S Flags are cleared.

CP
COMPARE

CP
Compare

CP dst, src

Instruction Format:



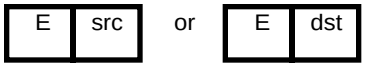
Cycles	OPC (Hex)	Address dst	Mode src
6	A2	r	r
6	A3	r	Ir
10	A4	R	R
10	A5	R	IR
10	A6	R	IM
10	A7	IR	IM

Operation: dst - src

The source operand is compared to (subtracted from) the destination operand, and the appropriate flags are set accordingly. The contents of both operands are unaffected.

Flags:
- C: Cleared if there is a carry from the most significant bit of the result. Set otherwise indicating a borrow.
- Z: Set if the result is zero; cleared otherwise.
- S: Set if result bit 7 is set (negative); cleared otherwise.
- V: Set if arithmetic overflow occurs; cleared otherwise.
- D: Unaffected
- H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the source or destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Working Register R3 contains 16H and Working Register R11 contains 20H, the statement:

CP R3, R11
OpCode: A2 3B

sets the C and S Flags, and the Z and V Flags are cleared.

CP
COMPARE

Example: If Working Register R15 contains 16H, Working Register R10 contains 20H, and Register 20H contains 11H, the statement:

CP R16, @R10
OpCode: A3 FA

clears the C, Z, S, and V Flags.

Example: If Register 34H contains 2EH and Register 12H contains 1BH, the statement:

CP 34H,12H
OpCode: A4 12 34

clears the C, Z, S, and V Flags.

Example: If Register 4BH contains 82H, Working Register R3 contains 10H, and Register 10H contains 01H, the statement:

CP 4BH, @R3
OpCode: A5 E3 4B

sets the S Flag, and clears the C, Z, and V Flags.

Example: If Register 6CH contains 2AH, the statement:

CP 6CH, #2AH
OpCode: A6 6C 2A

sets the Z Flag, and the C, S, and V Flags are all cleared.

Example: If Register D4H contains FCH, and Register FCH contains 8FH, the statement:

CP @D4H, 7FH
OpCode: A7 D4 FF

sets the V Flag, and the C, Z, and S Flags are all cleared.

DA
DECIMAL ADJUST

DA
Decimal Adjust

DA dst

Instruction Format:

Cycles	OPC (Hex)	Address Mode dst
8	40	R
8	41	IR

Operation: dst <— DA dst

The destination operand is adjusted to form two 4-bit BCD digits following a binary addition or subtraction operation on BCD encoded bytes. For addition (ADD and ADC) or subtraction (SUB and SBC), the following table indicates the operation performed.

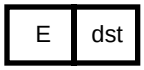
Instruction	Carry Before DA	Bits 7-4 Value (HEX)	H Flag Before DA	Bits 3-0 Value (HEX)	Number Added To Byte	Carry After DA
	0	0-9	0	0-9	00	0
	0	0-8	0	A-F	06	0
	0	0-9	1	0-3	06	0
ADD	0	A-F	0	0-9	60	1
ADC	0	9-F	0	A-F	66	1
	0	A-F	1	0-3	66	1
	1	0-2	0	0-9	60	1
	1	0-2	0	A-F	66	1
	1	0-3	1	0-3	66	1
	0	0-9	0	0-9	00	0
SUB	0	0-8	1	6-F	FA	0
SBC	1	7-F	0	0-9	A0	1
	1	6-F	1	6-F	9A	1

If the destination operand is not the result of a valid addition or subtraction of BCD digits, the operation is undefined.

Flags:
C: Set if there is a carry from the most significant bit; cleared otherwise (see table above).
Z: Set if the result is zero; cleared otherwise.
S: Set if result bit 7 is set (negative); cleared otherwise.
D Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For

example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If addition is performed using the BCD value 15 and 27, the result should be 42. The sum is incorrect, however, when the binary representations are added in the destination location using standard binary arithmetic.

```
  0001 0101 = 15H
+ 0010 0111 = 27H
-----------------
  0011 1100 = 3CH
```

If the result of the addition is stored in Register 5FH, the statement:

DA 5FH
OpCode: 40 5F

adjusts this result so the correct BCD representation is obtained.

```
0011 1100 = 3CH
0000 0110 = 06H
---------------
0100 0010 = 42H
```

Register 5F now contains the value 42H. The C, Z, and S Flags are cleared, and V is undefined.

Example: If addition is performed using the BCD value 15 and 27, the result should be 42. The sum is incorrect, however, when the binary representations are added in the destination location using standard binary arithmetic.

Register 45F contains the value 5FH, and the result of the addition is stored in Register 5FH, the

```
  0001 0101 = 15H
+ 0010 0111 = 27H
-----------------
  0011 1100 = 3CH
```

statement:

DA @45H
OpCode: 40 45

adjusts this result so the correct BCD representation is obtained.

```
0011 1100 = 3CH
0000 0110 = 06H
---------------
0100 0010 = 42H
```

Register 5F now contains the value 42H. The C, Z, and S Flags are cleared, and V is undefined.

DEC
DECREMENT

DEC
Decrement

DEC dst

Instruction Format:

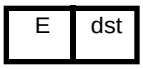
Cycles	OPC (Hex)	Address Mode dst
6	00	R
6	01	IR

Operation: dst <— dst - 1

The contents of the destination operand are decremented by one.

Flags:
C: Unaffected
Z: Set if the result is zero; cleared otherwise
S: Set if the result of bit 7 is set (negative); cleared otherwise
V: Set if arithmetic overflow occurs; cleared otherwise
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Working Register R10 contains 2A%, the statement:

DEC R10
OpCode: 00 EA

leaves the value 29H in Working Register R10. The Z, V, and S Flags are cleared.

Example: If Register B3H contains CBH, and Register CBH contains 01H, the statement:

DEC @B3H
OpCode: 01 B3

leaves the value 00H in Register CBH. The Z Flag is set, and the V and S Flags are cleared.

DECW
DECREMENT WORD

DECW
Decrement Word

DECW dst

Instruction Format:

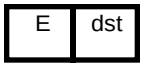
Cycles	OPC (Hex)	Address Mode dst
10	80	RR
10	81	IR

Operation: dst <— dst - 1

The contents of the destination (which must be an even address) operand are decremented by one. The destination operand can be a Register Pair or a Working Register Pair.

Flags:
C: Unaffected
Z: Set if the result is zero; cleared otherwise
S: Set if the result of bit 7 is set (negative); cleared otherwise
V: Set if arithmetic overflow occurs; cleared otherwise
D: Unaffected
H: Unaffected

Note: Address modes RR or IR can be used to specify a 4-bit Working Register Pair. In this format, the destination Working Register Pair operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register Pair R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Register Pair 30H and 31H contain the value 0AF2H, the statement:

DECW 30H
OpCode: 80 30

leaves the value 0AF1H in Register Pair 30H and 31H. The Z, V, and S Flags are cleared.

Example: If Working Register R0 contains 30H and Register Pairs 30H and 31H contain the value FAF3H, the statement:

DECW @R0
OpCode: 81 E0

leaves the value FAF2H in Register Pair 30H and 31H. The S Flag is set, and the Z and V Flags are cleared.

DI
DISABLE INTERRUPTS

DI
Disable Interrupts

DI

Instruction Format:

	Cycles	OPC (Hex)
OPC	6	8F

Operation: IMR (7) <— 0

Bit 7 of Control Register FBH (the Interrupt Mask Register) is reset to 0. All interrupts are disabled, although they remain “potentially” enabled. (For instance, the Global Interrupt Enable is cleared, but not the individual interrupt level enables.)

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Example: If Control Register FBH contains 8AH (10001010) (interrupts IRQ1 and IRQ3 are enabled), the statement:

DI
OpCode: 8F

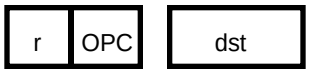
sets Control Register FBH to 0AH (00001010B) and disables these interrupts.

DJNZ
DECREMENT AND JUMP IF NON-ZERO

DJNZ
Decrement and Jump if Non-zero

DJNZ r, dst

Instruction Format:



Cycles		OPC (Hex)	Address Mode dst
12	if jump taken	rA	RA
10	if jump not taken	(r = 0 to F)	

Operation: r <— r - 1;
If r <> 0, PC <— PC + dst

The specified Working Register being used as a counter is decremented. If the contents of the specified Working Register are not zero after decrementing, then the relative address is added to the Program Counter (PC) and control passes to the statement whose address is now in the PC. The range of the relative address is +127 to –128. The original value of the PC is the address of the instruction byte following the DJNZ statement. When the specified Working Register counter reaches zero, control falls through to the statement following the DJNZ instruction.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Note: The Working Register being used as a counter must be one of the Registers from 04H to EFH. Use of one of the I/O ports, control or peripheral registers will have undefined results.

Example: DJNZ is typically used to control a “loop” of instructions. In this example, 12 bytes are moved from one buffer area in the register file to another. The steps involved are:

- Load 12 into the counter (Working Register R6).
- Set up the loop to perform the moves.
- End the loop with DJNZ.

The assembly listing required for this routine is as follows:

```
        LD R6, 12          ;Load Counter
LOOP:   LD R9, @R6         ;Move one byte to
        LD @R6, R9         ;new location
        DJNZ R6, LOOP      ;Decrement and Loop until
                           counter = 0
```

EI
ENABLE INTERRUPTS

EI
Enable Interrupts

EI

Instruction Format:

	Cycles	OPC (Hex)
OPC	6	9F

Operation: IMR (7) <— 0

Bit 7 of Control Register FBH (the Interrupt Mask Register) is set to 1. This allows potentially enabled interrupts to become enabled.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Example: If Control Register FBH contains 0AH (00001010) (interrupts IRQ1 and IRQ3 are selected), the statement:

EI
OpCode: 9F

sets Control Register FBH to 8AH (10001010B) and enables IRQ1 and IRQ3.

HALT
HALT

HALT
Halt

HALT

Instruction Format:

	Cycles	OPC (Hex)
OPC	6	7F

Operation: The HALT instruction turns off the internal CPU clock, but not the XTAL oscillation. The counter/timers and the external interrupts IRQ1, IRQ2, and IRQ3 remain active. The devices are recovered by interrupts, either externally or internally generated.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Note: In order to enter HALT mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. The user must execute a NOP immediately before the execution of the HALT instruction.

Example: Assuming the Z8 is in normal operation, the statements:

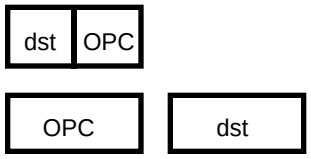
```
NOP
HALT
OpCodes:  FF  7F
```

place the Z8 into HALT mode.

INC
INCREMENT

Inc
Increment

Instruction Format:



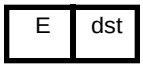
Cycles	OPC (Hex)	Address Mode dst
6	rE	r
6	20	R
6	21	IR

Operation: dst <— dst + 1

The contents of the destination operand are incremented by one.

Flags:
C: Unaffected
Z: Set if the result is zero; cleared otherwise.
S: Set if the result of bit 7 is set (negative); cleared otherwise.
V: Set if arithmetic overflow occurs; cleared otherwise.
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Working Register R10 contains 2AH, the statement:

INC R10
OpCode: AE

leaves the value 2BH in Working Register R10. The Z, V, and S Flags are cleared.

Example: If Register B3H contains CBH, the statement:

INC B3H
OpCode: 20 B3

leaves the value CCH in Register CBH. The S Flag is set, and the Z and V Flags are cleared.

Example: If Register B3H contains CBH and Register BCH contains FFH, the statement:

INC @B3H
OpCode: 21 B3

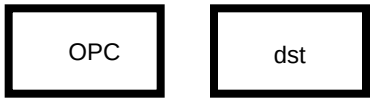
leaves the value 00H in Register CBH. The Z Flag is set, and the V and S Flags are cleared.

INCW
INCREMENT WORD

INCW
Increment Word

INCW dst

Instruction Format:



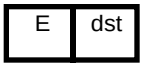
Cycles	OPC (Hex)	Address Mode dst
10	A0	RR
10	A1	IR
10	A0	R

Operation: dst <— dst - 1

The contents of the destination (which must be an even address) operand is decremented by one. The destination operand can be a Register Pair or a Working Register Pair.

Flags:
- C: Unaffected
- Z: Set if the result is zero; cleared otherwise.
- S: Set if the result of bit 7 is set (negative); cleared otherwise.
- V: Set if arithmetic overflow occurs; cleared otherwise.
- D: Unaffected
- H: Unaffected

Note: Address modes RR or IR can be used to specify a 4-bit Working Register Pair. In this format, the destination Working Register Pair operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register Pair R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Register Pairs 30H and 31H contain the value 0AF2H, the statement:

INCW 30H
OpCode: A0 30

leaves the value 0AF3H in Register Pair 30H and 31H. The Z, V, and S Flags are cleared.

Example: If Working Register R0 contains 30H, and Register Pairs 30H and 31H contain the value FAF3H, the statement:

INCW @R0
OpCode: A1 E0

leaves the value FAF4H in Register Pair 30H and 31H. The S Flag is set, and the Z and V Flags are cleared.

IRET
INTERRUPT RETURN

IRET
Interrupt RETURN

IRET

Instruction Format:

	Cycles	OPC (Hex)
OPC	16	BF

Operation: FLAGS <— @SP
SP <— SP + 1
PC <— @SP
SP <— SP + 2
IMR (7) <— 1

This instruction is issued at the end of an interrupt service routine. It restores the Flag Register (Control Register FCH) and the PC. It also re-enables any interrupts that are potentially enabled.

Flags:
C: Restored to original setting before the interrupt occurred.
Z: Restored to original setting before the interrupt occurred.
S: Restored to original setting before the interrupt occurred.
V: Restored to original setting before the interrupt occurred.
D: Restored to original setting before the interrupt occurred.
H: Restored to original setting before the interrupt occurred.

Example: If Stack Pointer Low Register FFH currently contains the value 45H, Register 45H contains the value 00H, Register 46H contains 6FH, and Register 47 Contains E4H, the statement:

IRET
OpCode: BF

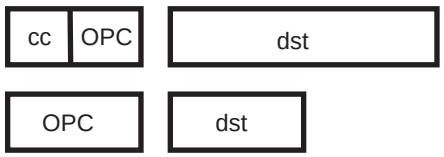
restores the FLAG Register FCH with the value 00H, restores the PC with the value 6FE4H, re-enables the interrupts, and sets the Stack Pointer Low to 48H. The next instruction to be executed will be at location 6FE4H.

JP
JUMP

JP
JUMP

JP cc, dst

Instruction Format:



Cycles		OPC (Hex)	Address Mode dst
12	if jump taken	ccD	DA
10	if not taken	cc = 0 to F	
8		30	IRR

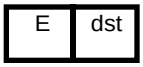
Operation: If cc (condition code) is true, then PC <— dst

A conditional jump transfers Program Control to the destination address if the condition specified by cc (condition code) is true. Otherwise, the instruction following the JP instruction is executed. See Section 12.3 for a list of condition codes.

The unconditional jump simply replaces the contents of the Program Counter with the contents of the register pair specified by the destination operand. Program Control then passes to the instruction addressed by the PC.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Note: Address mode IRR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If the Carry Flag is set, the statement:

JP C, 1520H
OpCode: 7D 15 20

replaces the contents of the Program Counter with 1520H and transfers program control to that location. If the Carry Flag had not been set, control would have fallen through to the statement following the JP instruction.

Example: If Working Register Pair RR2 contains the value 3F45H, the statement:

JP @RR2
OpCode: 30 E2

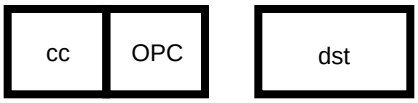
replaces the contents of the PC with the value 3F45H and transfers program control to that location.

JR
JUMP RELATIVE

JR
Jump Relative

JR cc, dst

Instruction Format:



Cycles	OPC (Hex)	Address Mode dst
12 if jump taken	ccB	RR
10 if jump not taken	cc = 0 to F	

Operation: If cc is true, PC <— PC + dst

If the condition specified by the “cc” is true, the relative address is added to the PC and control passes to the instruction located at the address specified by the PC (See Section 12.3 for a list of condition codes). Otherwise, the instruction following the JR instruction is executed. The range of the relative address is +127 to –128, and the original value of the PC is taken to be the address of the first instruction byte following the JR instruction.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Example: If the result of the last arithmetic operation executed is negative, the next four statements (which occupy a total of seven bytes) are skipped with the statement:

JR MI, #9
OpCode: 5B 09

If the result was not negative, execution would have continued with the instruction following the JR instruction.

Example: A short form of a jump –45 is:

JR #–45
OpCode: 8B D3

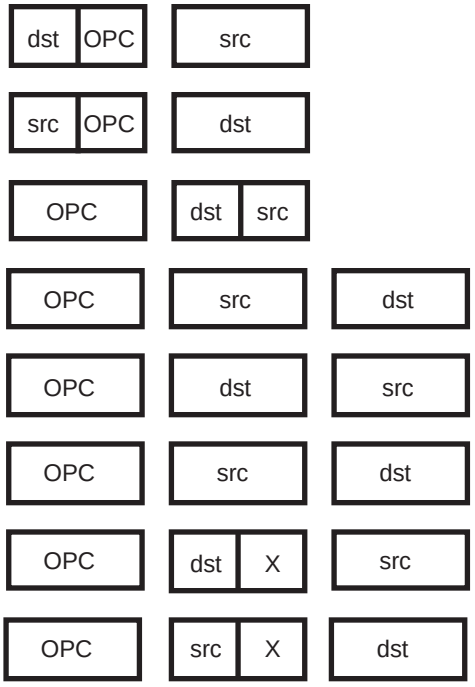
The condition code is “blank” in this case, and is assumed to be “always true.”

LD
LOAD

LD
Load

LD dst, src

Instruction Format:



Cycles	OPC (Hex)	Address Mode dst	Address Mode src
6	rC	r	IM
6	r8	r	R
6	r9 r = 0 to F	R*	r
6	E3	r	Ir
6	F3	Ir	r
10	E4	R	R
10	E5	R	IR
10	E6	R	IM
10	E7	IR	IM
10	F5	IR	R
10	C7	r	X
10	D7	X	r

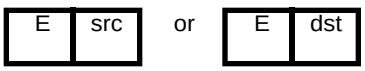
* In this instance, only a full 8-bit register can be used.

Operation: dst <— src

The contents of the source operand are loaded into the destination operand. The contents of the source operand are not affected.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the source or destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



LD
LOAD

Example: The statement:

LD R15, #34H
OpCode: FC 34

loads the value 34H into Working Register R15.

Example: If Register 34H contains the value FCH, the statement:

LD R14, 34H
OpCode: F8 34

loads the value FCH into Working Register R15. The contents of Register 34H are not affected.

Example: If Working Register R14 contains the value 45H, the statement:

LD 34H, R14
OpCode: E9 34

loads the value 45H into Register 34H. The contents of Working Register R14 are not affected.

Example: If Working Register R12 contains the value 34H, and Register 34H contains the value FFH, the statement:

LD R13, @R12
OpCode: E3 DC

loads the value FFH into Working Register R13. The contents of Working Register R12 and Register R34 are not affected.

Example: If Working Register R13 contains the value 45H, and Working Register R12 contains the value 00H the statement:

LD @R13, R12
OpCode: F3 DC

loads the value 00H into Register 45H. The contents of Working Register R12 and Working Register R13 are not affected.

Example: If Register 45H contains the value CFH, the statement:

LD 34H, 45H
OpCode: E4 45 34

loads the value CFH into Register 34H. The contents of Register 45H are not affected.

LD
LOAD

Example: If Register 45H contains the value CFH and Register CFH contains the value FFH, the statement:

LD 34H, @45H
OpCode: E5 45 34

loads the value FFH into Register 34H. The contents of Register 45H and Register CFH are not affected.

Example: The statement:

LD 34H, #A4H
OpCode: E6 34 A4

loads the value A4H into Register 34H.

Example: If Working Register R14 contains the value 7FH, the statement:

LD @R14, #FCH
OpCode: E7 EE FC

loads the value FCH into Register 7FH. The contents of Working Register R14 are not affected.

Example: If Register 34H contains the value CFH and Register 45H contains the value FFH, the statement:

LD @34H, 45H
OpCode: F5 45 34

loads the value FFH into Register CFH. The contents of Register 34H and Register 45H are not affected.

Example: IIf Working Register R0 contains the value 08H and Register 2CH (24H + 08H = 2CH) contains the value 4FH, the statement:

LD R10, 24H(R0)
OpCode: C7 A0 24

loads Working Register R10 with the value 4FH. The contents of Working Register R0 and Register 2CH are not affected.

Example: If Working Register R0 contains the value 0BH and Working Register R10 contains 83H the statement:

LD F0H(R0), R10
OpCode: D7 A0 F0

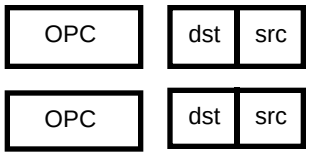
loads the value 83H into Register FBH (F0H + 0BH = FBH). Since this is the Interrupt Mask Register, the LOAD statement has the effect of enabling IRQ0 and IRQ1. The contents of Working Registers R0 and R10 are unaffected by the load.

LDC
LOAD CONSTANT

LDC
Load Constant

LDC dst, src

Instruction Format:



Cycles	OPC (Hex)	Address Mode src	Address Mode dst
12	C2	r	Irr
12	D2	Irr	r

Operation: dst <— src

This instruction is used to load a byte constant from program memory into a Working Register, or vice versa. The address of the program memory location is specified by a Working Register Pair. The contents of the source operand are not affected.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Example: If Working Register Pair R6 and R7 contain the value 30A2H and program memory location 30A2H contains the value 22H, the statement:

LDC R2, @RR6
OpCode: C2 26

loads the value 22H into Working Register R2. The value of program memory location 30A2H is unchanged by the load.

Example: If Working Register R2 contains the value 22H, and Working Register Pair R6 and R7 contains the value 10A2H, the statement:

LDC @RR6, R2
OpCode: D2 26

loads the value 22H into program memory location 10A2H. The value of Working Register R2 is unchanged by the load.

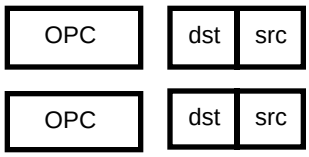
Note: This instruction format is valid only for MCUs which can address external program memory.

LDCI
LOAD CONSTANT AUTO-INCREMENT

LDCI
Load Constant Auto-increment

LDCI dst, src

Instruction Format:



Cycles	OPC (Hex)	Address Mode src	Address Mode dst
18	C3	Ir	Irr
18	D3	Irr	Ir

Operation: dst <— src
r <— r + 1
rr <— rr + 1

This instruction is used for block transfers of data between program memory and the Register File. The address of the program memory location is specified by a Working Register Pair, and the address of the Register File location is specified by Working Register. The contents of the source location are loaded into the destination location. Both addresses in the Working Registers are then incremented automatically. The contents of the source operand are not affected.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Example: If Working Register Pair R6-R7 contains 30A2H, program memory location 30A2H and 30A3H contain 22H and BCH respectively, and Working Register R2 contains 20H, the statement:

LDCI @R2, @RR6
OpCode: C3 26

loads the value 22H into Register 20H. Working Register Pair RR6 is incremented to 30A3H and Working Register R2 is incremented to 21H. A second

LDCI @R2, @RR6
OpCode: C3 26

loads the value BCH into Register 21H. Working Register Pair RR6 is incremented to 30A4H and Working Register R2 is incremented to 22H.

LDCI
LOAD CONSTANT AUTO-INCREMENT

Example: If Working Register R2 contains 20H, Register 20H contains 22H, Register 21H contains BCH, and Working Register Pair R6-R7 contains 30A2H, the statement:

LDCI @RR6, @R2
OpCode: D3 26

loads the value 22H into program memory location 30A2H. Working Register R2 is incremented to 21H and Working Register Pair R6-R7 is incremented to 30A3H. A second

LDCI @RR6, @R2
OpCode: D3 26

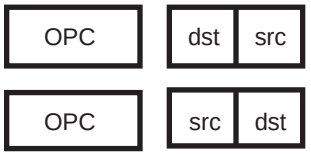
loads the value BCH into program memory location 30A3H. Working Register R2 is incremented to 22H and Working Register Pair R6-R7 is incremented to 30A4H.

LDE
LOAD EXTERNAL DATA

LDE
Load External Data

LDE dst, src

Instruction Format:



Cycles	OPC (Hex)	Address Mode src	Address Mode dst
12	82	r	Irr
12	92	Irr	r

Operation: dst <— src

This instruction is used to load a byte from external data memory into a Working Register or vice versa. The address of the external data memory location is specified by a Working Register Pair. The contents of the source operand are not affected.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Example: If Working Register Pair R6 and R7 contain the value 40A2H and external data memory location 40A2H contains the value 22H, the statement:

LDE R2, @RR6
OpCode: 82 26

loads the value 22H into Working Register R2. The value of external data memory location 40A2H is unchanged by the load.

Example: If Working Register Pair R6 and R7 contain the value 404AH and Working Register R2 contains the value 22H, the statement:

LDE @RR6, R2
OpCode: 92 26

loads the value 22H into external data memory location 404AH

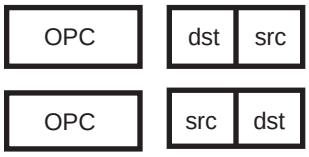
Note: This instruction format is valid only for MCUs which can address external data memory.

LDEI
LOAD EXTERNAL DATA AUTO-INCREMENT

LDEI
Load External Data Auto-increment

LDEI dst, src

Instruction Format:



Cycles	OPC (Hex)	Address Mode src	Address Mode dst
18	83	Ir	Irr
18	93	Irr	Ir

Operation: dst <— src
r <— r + 1
rr <— rr + 1

This instruction is used for block transfers of data between external data memory and the Register File. The address of the external data memory location is specified by a Working Register Pair, and the address of the Register File location is specified by a Working Register. The contents of the source location are loaded into the destination location. Both addresses in the Working Registers are then incremented automatically. The contents of the source are not affected.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Example: If Working Register Pair R6 and R7 contains 404AH, external data memory location 404AH and 404BH contain ABH and C3H respectively, and Working Register R2 contains 22H, the statement:

LDEI @R2, @RR6
OpCode: 83 26

loads the value ABH into Register 22H. Working Register Pair RR6 is incremented to 404BH and Working Register R2 is incremented to 23H. A second

LDEI @R2, @RR6
OpCode: 83 26

loads the value C3H into Register 23H. Working Register Pair RR6 is incremented to 404CH and Working Register R2 is incremented to 24H.

LDEI
LOAD EXTERNAL DATA AUTO-INCREMENT

Example: If Working Register R2 contains 22H, Register 22H contains ABH, Register 23H contains C3H, and Working Register Pair R6 and R7 contains 404AH, the statement:

LDEI @RR6, @R2
OpCode: 93 26

loads the value ABH into external data memory location 404AH. Working Register R2 is incremented to 23H and Working Register Pair RR6 is incremented to 404BH. A second

LDEI @RR6, @R2
OpCode: 93 26

loads the value C3H into external data memory location 404BH. Working Register R2 is incremented to 24H and Working Register Pair RR6 is incremented to 404CH.

Note: This instruction format is valid only for MCUs which can address external data memory.

NOP
NO OPERATION

NOP
No Operation

NOP

Instruction Format:

	Cycles	OPC (Hex)
OPC	6	FF

Operation: No action is performed by this instruction. It is typically used for timing delays or clearing the pipeline.

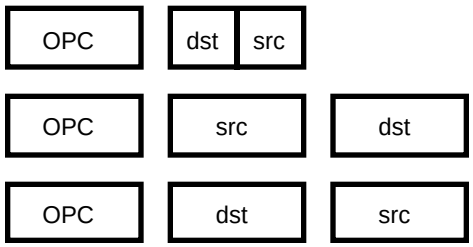
Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

OR
LOGICAL OR

OR
Logical OR

OR dst, src

Instruction Format:



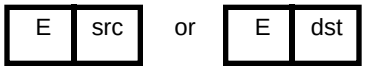
Cycles	OPC (Hex)	Address Mode dst	src
6	42	r	r
6	43	r	Ir
10	44	R	R
10	45	R	IR
10	46	R	IM
10	47	IR	IM

Operation: dst <— dst OR src

The source operand is logically ORed with the destination operand and the result is stored in the destination operand. The contents of the source operand are not affected. The OR operation results in a one bit being stored whenever either of the corresponding bits in the two operands is a one. Otherwise, a zero bit is stored.

Flags:
C: Unaffected
Z: Set if the result is zero; cleared otherwise
S: Set if the result of bit 7 is set; cleared otherwise
V: Always reset to 0
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the source or destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Working Register R1 contains 34H (00111000B) and Working Register R14 contains 4DH (10001101), the statement:

OR R1, R14
OpCode: 42 1E

leaves the value BDH (10111101B) in Working Register R1. The S Flag is set, and the Z and V Flags are cleared.

OR
LOGICAL OR

Example: If Working Register R4 contains F9H (11111001B), Working Register R13 contains 7BH, and Register 7B contains 6AH (01101010B), the statement:

OR R4, @R13
OpCode: 43 4D

leaves the value FBH (11111011B) in Working Register R4. The S Flag is set, and the Z and V Flags are cleared.

Example: If Register 3AH contains the value F5H (11110101B) and Register 42H contains the value 0AH (00001010), the statement:

OR 3AH, 42H
OpCode: 44 42 3A

leaves the value FFH (11111111B) in Register 3AH. The S Flag is set, and the Z and V Flags are cleared.

Example: If Working Register R5 contains 70H (01110000B), Register 45H contains 3AH, and Register 3AH contains 7FH (01111111B), the statement:

OR R5, @45H
OpCode: 45 45 E5

leaves the value 7FH (01111111B) in Working Register R5. The Z, V, and S Flags are cleared.

Example: If Register 7AH contains the value F3H (11110111B), the statement:

OR 7AH, #F0H
OpCode: 46 7A F0

leaves the value F3H (11110111B) in Register 7AH. The S Flag is set, and the Z and V Flags are cleared.

Example: If Working Register R3 contains the value 3EH and Register 3EH contains the value 0CH (00001100B), the statement:

OR @R3, #05H
OpCode: 57 E3 05

leaves the value 0DH (00001101B) in Register 3EH. The Z, V, and S Flags are cleared.

POP
POP

POP
Pop

POP dst

Instruction Format:

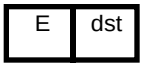
	Cycles	OPC (Hex)	Address Mode dst
OPC dst	10	50	R
	10	51	IR

Operation: dst <— @SP
SP <— SP + 1

The contents of the location specified by the SP (Stack Pointer) are loaded into the destination operand. The SP is then incremented automatically.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If the SP (Control Registers FEH and FFH) contains the value 70H and Register 70H contains 44H, the statement:

POP 34H
OpCode: 50 34

loads the value 44H into Register 34H. After the POP operation, the SP contains 71H. The contents of Register 70 are not affected.

Example: If the SP (Control Registers FEH and FFH) contains the value 1000H, external data memory location 1000H contains 55H, and Working Register R6 contains 22H, the statement:

POP @R6
OpCode: 51 E6

loads the value 55H into Register 22H. After the POP operation, the SP contains 1001H. The contents of Working Register R6 are not affected.

PUSH
Push

PUSH src

Instruction Format:

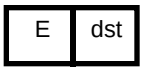
Cycles	OPC (Hex)	Address Mode dst
10 Internal Stack 12 External Stack	70	R
10 Internal Stack 10 External Stack	71	IR

Operation: SP <— SP - 1
@SP <— src

The contents of the SP (stack pointer) are decremented by one, then the contents of the source operand are loaded into the location addressed by the decremented SP, thus adding a new element to the stack.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If the SP contains 1001H, the statement:

PUSH FCH
OpCode: 70 FC

stores the contents of Register FCH (the Flag Register) in location 1000H. After the PUSH operation, the SP contains 1000H.

Example: If the SP contains 61H and Working Register R4 contains FCH, the statement:

PUSH @R4
OpCode: 71 E4

stores the contents of Register FCH (the Flag Register) in location 60H. After the PUSH operation, the SP contains 60H.

RCF
RESET CARRY FLAG

RCF
Reset Carry Flag

RCF

Instruction Format:

	Cycles	OPC (Hex)
OPC	6	CF

Operation: C <— 0

The C Flag is reset to 0, regardless of its previous value.

Flags:
C: Reset to 0
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Example: If the C Flag is currently set, the statement:

RCF
OpCode: CF

resets the Carry Flag to 0.

RET
RETURN

RET
Return

RET

Instruction Format:

	Cycles	OPC (Hex)
OPC	14	AF

Operation: PC <— @SP
SP <— SP + 2

This instruction is normally used to return from a procedure entered by a CALL instruction. The contents of the location addressed by the SP are popped into the PC. The next statement executed is the one addressed by the new contents of the PC. The stack pointer is also incremented by two.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Note: Each PUSH instruction executed within the subroutine should be countered with a POP instruction in order to guarantee the SP is at the correct location when the RET instruction is executed. Otherwise the wrong address will be loaded into the PC and the program will not operate as desired.

Example: If SP contains 2000H, external data memory location 2000H contains 18H, and location 2001H contains B5H, the statement:

RET
OpCode: AF

leaves the value 2002H in the SP, and the PC contains 18B5H, the address of the next instruction to be executed.

RL
ROTATE LEFT

RL
Rotate Left

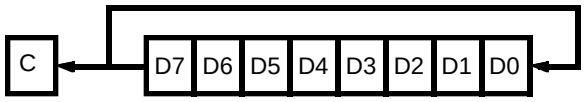
RL dst

Instruction Format:

Cycles	OPC (Hex)	Address Mode dst
6	90	R
6	91	IR

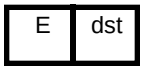
Operation:
C <— dst(7)
dst(0) <— dst(7)
dst(1) <— dst(0)
dst(2) <— dst(1)
dst(3) <— dst(2)
dst(4) <— dst(3)
dst(5) <— dst(4)
dst(6) <— dst(5)
dst(7) <— dst(6)

The contents of the destination operand are rotated left by one bit position. The initial value of bit 7 is moved to the bit 0 position and also into the Carry Flag.



Flags:
C: Set if the bit rotated from the most significant bit position was 1 (i.e., bit 7 was 1).
Z: Set if the result is zero; cleared otherwise.
S: Set if the result in bit 7 is set; cleared otherwise.
V: Set if arithmetic overflow occurred (if the sign of the destination operand changed during rotation); cleared otherwise.
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



RL
ROTATE LEFT

Example: If the contents of Register C6H are 88H (10001000B), the statement:

RL C6H
OpCode: 80 C6

leaves the value 11H (00010001B) in Register C6H. The C and V Flags are set, and the S and Z Flags are cleared.

Example: If the contents of Register C6H are 88H, and the contents of Register 88H are 44H (01000100B), the statement:

RL @C6H
OpCode: 81 C6

leaves the value 88H in Register 88H (10001000B). The S and V Flags are set, and the C and Z Flags are cleared.

RLC
ROTATE LEFT THROUGH CARRY

RLC
Rotate Left Through Carry

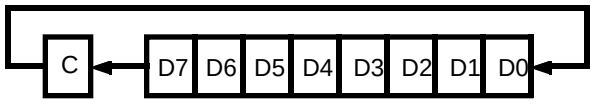
RLC dst

Instruction Format:

Cycles	OPC (Hex)	Address Mode dst
6	10	R
6	11	IR

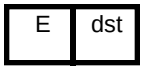
Operation:
C<— dst(7)
dst(0) <— C
dst(1) <— dst(0)
dst(2) <— dst(1)
dst(3) <— dst(2)
dst(4) <— dst(3)
dst(5) <— dst(4)
dst(6) <— dst(5)
dst(7) <— dst(6)

The contents of the destination operand along with the C Flag are rotated left by one bit position. The initial value of bit 7 replaces the C Flag and the initial value of the C Flag replaces bit 0.



Flags:
C: Set if the bit rotated from the most significant bit position was 1 (i.e., bit 7 was 1).
Z: Set if the result is zero; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Set if arithmetic overflow occurred (if the sign of the destination operand changed during rotation); cleared otherwise.
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



RLC
ROTATE LEFT THROUGH CARRY

Example: If the C Flag is reset and Register C6 contains 8F (10001111B), the statement:

RLC C6
OpCode: 10 C6

leaves Register C6 with the value 1EH (00011110B). The C and V Flags are set, and S and Z Flags are cleared.

Example: If the C Flag is reset, Working Register R4 contains C6H, and Register C6 contains 8F (10001111B), the statement:

RLC @R4
OpCode: 11 E4

leaves Register C6 with the value 1EH (00011110B). The C and V Flags are set, and S and Z Flags are cleared.

RR
ROTATE RIGHT

RR
Rotate Right

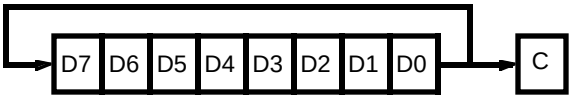
RR dst

Instruction Format:

	Cycles	OPC (Hex)	Address Mode dst
OPC dst	6	E0	R
	6	E1	IR

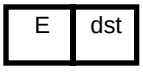
Operation:
C <— dst(0)
dst(0) <— dst(1)
dst(1) <— dst(2)
dst(2) <— dst(3)
dst(3) <— dst(4)
dst(4) <— dst(5)
dst(5) <— dst(6)
dst(6) <— dst(7)
dst(7) <— dst(0)

The contents of the destination operand are rotated to the right by one bit position. The initial value of bit 0 is moved to bit 7 and also into the C Flag.



Flags:
C: Set if the bit rotated from the least significant bit position was 1 (i.e., bit 0 was 1).
Z: Set if the result is zero; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Set if arithmetic overflow occurred (if the sign of the destination operand changed during rotation); cleared otherwise.
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



RR
ROTATE RIGHT

Example: If the contents of Working Register R6 are 31H (00110001B), the statement:

RR R6
OpCode: E0 E6

leaves the value 98H (10011000) in Working Register R6. The C, V, and S Flags are set, and the Z Flag is cleared.

Example: If the contents of Register C6 are 31H and the contents of Register 31H are 7EH (01111110B), the statement:

RR @C6
OpCode: E1 C6

leaves the value 4FH (00111111) in Register 31H. The C, Z, V, and S Flags are cleared.

RRC
ROTATE RIGHT THROUGH CARRY

RRC
Rotate Right Through Carry

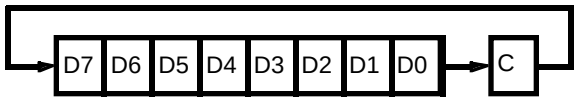
RRC dst

Instruction Format:

Cycles	OPC (Hex)	Address Mode dst
6	C0	R
6	C1	IR

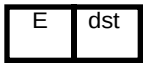
Operation:
C <— dst(0)
dst(0) <— dst(1)
dst(1) <— dst(2)
dst(2) <— dst(3)
dst(3) <— dst(4)
dst(4) <— dst(5)
dst(5) <— dst(6)
dst(6) <— dst(7)
dst(7) <— C

The contents of the destination operand with the C Flag are rotated right by one bit position. The initial value of bit 0 replaces the C Flag and the initial value of the C Flag replaces bit 7.



Flags:
C: Set if the bit rotated from the least significant bit position was 1 (i.e., bit 0 was 1).
Z: Set if the result is zero; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Set if arithmetic overflow occurred (if the sign of the destination operand changed during rotation); cleared otherwise.
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



RRC
ROTATE RIGHT THROUGH CARRY

Example: If the contents of Register C6H are DDH (11011101B) and the C Flag is reset, the statement:

RRC C6H
OpCode: C0 C6

leaves the value 6EH (01101110B) in register C6H. The C and V Flags are set, and the Z and S Flags are cleared.

Example: If the contents of Register 2C are EDH, the contents of Register EDH is 00H (00000000B), and the C Flag is reset, the statement:

RRC @2CH
OpCode: C1 2C

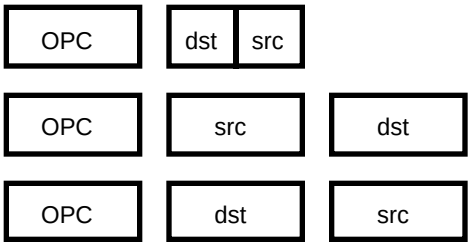
leaves the value 02H (00000010B) in Register EDH. The C, Z, S, and V Flags are reset.

SBC
SUBTRACT WITH CARRY

SBC
Subtract With Carry

SBC dst, src

Instruction Format:



Cycles	OPC (Hex)	Address dst	Mode src
6	32	r	r
6	33	r	Ir
10	34	R	R
10	35	R	IR
10	36	R	IM
10	37	IR	IM

Operation: dst <— dst - src - C

The source operand, along with the setting of the C Flag, is subtracted from the destination operand and the result is stored in the destination operand. The contents of the source operand are not affected. Subtraction is performed by adding the two's complement of the source operand to the destination operand. In multiple precision arithmetic, this instruction permits the carry (borrow) from the subtraction of low order operands to be subtracted from the subtraction of high order operands.

Flags:

C: Cleared if there is a carry from the most significant bit of the result; set otherwise, indicating a "borrow."

Z: Set if the result is 0; cleared otherwise.

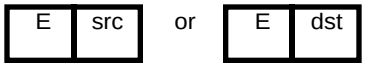
V: Set if arithmetic overflow occurred (if the operands were of opposite sign and the sign of the result is the same as the sign of the source); reset otherwise.

S: Set if the result is negative; cleared otherwise.

H: Cleared if there is a carry from the most significant bit of the low order four bits of the result; set otherwise indicating a "borrow."

D: Always set to 1.

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the source or destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



SBC
SUBTRACT WITH CARRY

Example: Working Register R3 contains 16H, the C Flag is set to 1, and Working Register R11 contains 20H, the statement:

SBC R3, R11
OpCode: 32 3B

leaves the value F5H in Working Register R3. The C, S, and D Flags are set, and the Z, V, and H Flags are all cleared.

Example: If Working Register R15 contains 16H, the C Flag is not set, Working Register R10 contains 20H, and Register 20H contains 11H, the statement:

SBC R16, @R10
OpCode: 33 FA

leaves the value 05H in Working Register R15. The D Flag is set, and the C, Z, S, V, and H Flags are cleared.

Example: :If Register 34H contains 2EH, the C Flag is set, and Register 12H contains 1BH, the statement:

SBC 34H, 12H
OpCode: 34 12 34

leaves the value 13H in Register 34H. The D Flag is set, and the C, Z, S, V, and H Flags are cleared.

Example: If Register 4BH contains 82H, the C Flag is set, Working Register R3 contains 10H, and Register 10H contains 01H, the statement:

SBC 4BH, @R3
OpCode: 35 E3 4B

leaves the value 80H in Register 4BH. The D Flag is set, and the C, Z, S, V, and H Flags are cleared.

Example: If Register 6CH contains 2AH, and the C Flag is not set, the statement:

SBC 6CH, #03H
OpCode: 36 6C 03

leaves the value 27H in Register 6CH. The D Flag is set, and the C, Z, S, V, and H Flags are cleared.

Example: If Register D4H contains 5FH, Register 5FH contains 4CH, and the C Flag is set, the statement:

SBC @D4H, #02H
OpCode: 37 D4 02

leaves the value 4AH in Register 5FH. The D Flag is set, and the C, Z, S, V, and H Flags are cleared.

SCF
SET CARRY FLAG

SCF
Set Carry Flag

SRC

Instruction Format:

	Cycles	OPC (Hex)
OPC	6	DF

Operation: C <— 1

The C Flag is set to 1, regardless of its previous value.

Flags:

C:	Set to 1
Z	Unaffected
S	Unaffected
V	Unaffected
D	Unaffected
H	Unaffected

Example: If the C Flag is currently reset, the statement:

SCF
OpCode: DF

sets the Carry Flag to 1.

SRA
SHIFT RIGHT ARITHMETIC

SRA
Shift Right Arithmetic

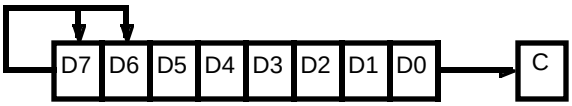
SRA dst

Instruction Format:

6	D0	R
6	D1	IR

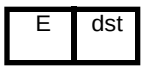
Operation:
C <— dst(0)
dst(0) <— dst(1)
dst(1) <— dst(2)
dst(2) <— dst(3)
dst(3) <— dst(4)
dst(4) <— dst(5)
dst(5) <— dst(6)
dst(6) <— dst(7)
dst(7) <— dst(7)

An arithmetic shift right by one bit position is performed on the destination operand. Bit 0 replaces the C Flag. Bit 7 (the Sign bit) is unchanged and its value is shifted into bit 6.



Flags:
C: Set if the bit rotated from the least significant bit position was 1 (i.e., bit 0 was 1).
Z: Set if the result is zero; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Always reset to 0.
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



SRA
SHIFT RIGHT ARITHMETIC

Example: If the contents of Working Register R6 are 31H (00110001B), the statement:

SRA R6
OpCode: D0 E6

leaves the value 98H (00011000) in Working Register R6. The C Flag is set, and the Z, V, and S Flags are cleared.

Example: If Register C6 contains the value DFH, and Register DFH contains the value B8H (10111000B), the statement:

SRA @C6
OpCode: D1 C6

leaves the value DCH (11011100B) in Register DFH. The C, Z, and V Flags are reset, and the S Flag is set.

SRP
SET REGISTER POINTER

SRP
Set Register Pointer

SRP src

Instruction Format:

Cycles	OPC (Hex)	Address Mode dst
6	31	IM

Operation: RP <— src

The specified value is loaded into the Register Pointer (RP) (Control Register FDH). Bits 7-4 determine the Working Register Group. Bits 3-0 selects the Expanded Register Bank. Addressing of un-implemented Working Register Group, while using Expanded Register Banks, will point to Bank 0.

Example: SRP TD addresses Working Register Group 7 of Bank 0.

Register Pointer (FDH) Contents (Bin)	Working Register Group (Hex)	Actual Registers (Hex)
1111 0000	F	F0-FF
1110 0000	E	E0-EF
1101 0000	D	D0-DF
1100 0000	C	C0-CF
1011 0000	B	B0-BF
1010 0000	A	A0-AF
1001 0000	9	90-9F
1000 0000	8	80-8F
0111 0000	7	70-7F
0110 0000	6	60-6F
0101 0000	5	50-5F
0100 0000	4	40-4F
0011 0000	3	30-3F
0010 0000	2	20-2F
0001 0000	1	10-1F
0000 0000	0	00-0F

SRP
SET REGISTER POINTER

Register Pointer (FDH) Contents (Hex)	Expanded Register Bank (Hex)	Working Registers (Dec)
xxxx 1111	F	R0-R15
xxxx 1110	E	R0-R15
xxxx 1101	D	R0-R15
xxxx 1100	C	R0-R15
xxxx 1011	B	R0-R15
xxxx 1010	A	R0-R15
xxxx 1001	9	R0-R15
xxxx 1000	8	R0-R15
xxxx 0111	7	R0-R15
xxxx 0110	6	R0-R15
xxxx 0101	5	R0-R15
xxxx 0100	4	R0-R15
xxxx 0011	3	R0-R15
xxxx 0010	2	R0-R15
xxxx 0001	1	R0-R15
xxxx 0000	0	R0-R15

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Note: When an Expanded Register Bank , other than Bank 0 is selected, access to the Z8 Standard Register File is possible except for the Port Register and general purpose registers 04H to 0FH.

fpr Register Addresses 0H to FH.

Example: The statement:

SRP F0H OpCode: 31 F0

sets the Register Pointer to access expanded Register Bank 0 and Working Register Group F in the Z8 Standard Register File. All references to Working Registers now affect this group of 16 registers. Registers F0H to FFH can be accessed as Working Registers R0 to R15

Example: The statement:

SRP 0FH
OpCode: 31 0F

sets the Register Pointer to access Expanded Register Bank F, Reg 00H to Reg 0FH, as the current Working Registers. All references to Working Registers now affect this group of 16 registers. These registers are now accessed as Working Registers R0 to R15. Port Registers are now not accessable.

SRP
SET REGISTER POINTER

Example: Assume the RP currently addresses the Control and Peripheral Working Register Group and the program has just entered an interrupt service routine. The statement:

SRP 70H
OpCode: 31 70

retains the contents of the Control and Peripheral Registers by setting the RP to 70H (01110000B). Any reference to Working Registers in the interrupt routine will point to registers 70H to 7FH.

STOP
STOP

STOP
Stop

STOP

Instruction Format:

	Cycles	OPC (Hex)
OPC	6	6F

Operation: This instruction turns off the internal system clock (SCLK) and external crystal (XTAL) oscillation, and reduces the standby current. The STOP mode is terminated by a RESET which causes the processor to restart the application program at address 000CH.

Flags:
C: Unaffected
Z: Unaffected
S: Unaffected
V: Unaffected
D: Unaffected
H: Unaffected

Note: In order to enter STOP mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. The user must execute a NOP immediately before the execution of the STOP instruction.

Example: The statements:

```
NOP
STOP
OpCodes:  FF  6F
```

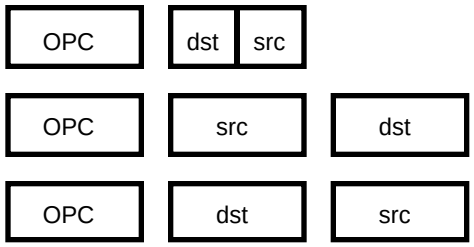
place the Z8 into STOP mode.

SUB
SUBTRACT

SUB
Subtract

SUB dst, src

Instruction Format:



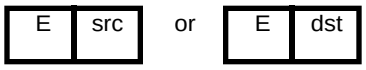
Cycles	OPC (Hex)	Address dst	Mode src
6	22	r	r
6	23	r	Ir
10	24	R	R
10	25	R	IR
10	26	R	IM
10	27	IR	IM

Operation: dst <— dst - src

The source operand is subtracted from the destination operand and the result is stored in the destination operand. The contents of the source operand are not affected. Subtraction is performed by adding the two's complement of the source operand to the destination operand.

Flags:
- C: Cleared if there is a carry from the most significant bit of the result; set otherwise, indicating a "borrow."
- Z: Set if the result is 0; cleared otherwise.
- V: Set if arithmetic overflow occurred (if the operands were of opposite sign and the sign of the result is the same as the sign of the source); reset otherwise.
- S: Set if the result is negative; cleared otherwise.
- H: Cleared if there is a carry from the most significant bit of the low order four bits of the result; set otherwise indicating a "borrow."
- D: Always set to 1.

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the source or destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example:]If Working Register R3 contains 16H, and Working Register R11 contains 20H, the statement:

SUB R3, R11
OpCode: 22 3B

leaves the value F6H in Working Register R3. The C, S, and D Flags are set, and the Z, V, and H Flags are cleared.

SUB
SUBTRACT

Example: If Working Register R15 contains 16H, Working Register R10 contains 20H, and Register 20H contains 11H, the statement:

SUB R16, @R10
OpCode: 23 FA

leaves the value 05H in Working Register R15. The D Flag is set, and the C, Z, S, V, and H Flags are cleared.

Example: If Register 34H contains 2EH, and Register 12H contains 1BH, the statement:

SUB 34H, 12H
OpCode: 24 12 34

leaves the value 13H in Register 34H. The D Flag is set, and the C, Z, S, V, and H Flags are cleared.

Example: If Register 4BH contains 82H, Working Register R3 contains 10H, and Register 10H contains 01H, the statement:

SUB 4BH, @R3
OpCode: 25 E3 4B

leaves the value 81H in Register 4BH. The D Flag is set, and the C, Z, S, V, and H Flags are cleared.

Example: If Register 6CH contains 2AH, the statement:

SUB 6CH, #03H
OpCode: 26 6C 03

leaves the value 27H in Register 6CH. The D Flag is set, and the C, Z, S, V, and H Flags are cleared.

Example: If Register D4H contains 5FH, Register 5FH contains 4CH, the statement:

SUB @D4H, #02H
OpCode: 17 D4 02

leaves the value 4AH in Register 5FH. The D Flag is set, and the C, Z, S, V, and H Flags are cleared.

SWAP
SWAP NIBBLES

SWAP
Swap Nibbles

SWAP dst

Instruction Format:

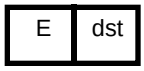
Cycles	OPC (Hex)	Address Mode dst
6	F0	R
6	F1	IR

Operation: dst(7-4) <—> dst(3-0)

The contents of the lower four bits and upper four bits of the destination operand are swapped.

Flags:
C: Unaffected
Z: Set if the result is zero; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Undefined
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Register BCH contains B3H (10110011B), the statement:

SWAP B3H
OpCode: F0 B3

will leave the value 3BH (00111011B) in Register BCH. The Z and S Flags are cleared.

Example: If Working Register R5 contains BCH and Register BCH contains B3H (10110011B), the statement:

SWAP @R5H
OpCode: F1 E5

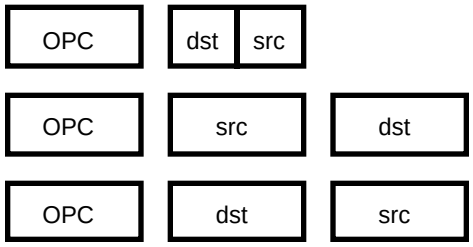
will leave the value 3BH (00111011B) in Register BCH. The Z and S Flags are cleared.

TCM
TEST COMPLEMENT UNDER MASK

TCM
Test Complement Under Mask

TCM dst, src

Instruction Format:



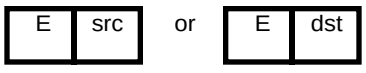
Cycles	OPC (Hex)	Address dst	Mode src
6	62	r	r
6	63	r	Ir
10	64	R	R
10	65	R	IR
10	66	R	IM
10	67	IR	IM

Operation: (NOT dst) AND src

This instruction tests selected bits in the destination operand for a logical 1 value. The bits to be tested are specified by setting a 1 bit in the corresponding bit position in the source operand (the mask). The TCM instruction complements the destination operand, and then ANDs it with the source mask (operand). The Zero (Z) Flag can then be checked to determine the result. If the Z Flag is set, then the tested bits were 1. When the TCM operation is complete, the destination and source operands still contain their original values.

Flags:
C: Unaffected
Z: Set if the result is zero; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Always reset to 0.
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the source or destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



TCM
TEST COMPLEMENT UNDER MASK

Example: If Working Register R3 contains 45H (01000101B) and Working Register R7 contains the value 01H (00000001B) (bit 0 is being tested if it is 1), the statement:

```
TCM R3, R7
OpCode: 62 37
```

will set the Z Flag indicating bit 0 in the destination operand is 1. The V and S Flags are cleared.

Example: If Working Register R14 contains the value F3H (11110011B), Working Register R5 contains CBH, and Register CBH contains 88H (10001000B) (bit 7 and bit 3 are being tested if they are 1), the statement:

```
TCM R14, @R5
OpCode: 63 E5
```

will reset the Z Flag, because bit 3 in the destination operand is not a 1. The V and S Flags are also cleared.

Example: If Register D4H contains the value 04H (000001000B), and Working Register R0 contains the value 80H (10000000B) (bit 7 is being tested if it is 1), the statement:

```
TCM D4H, R0
OpCode: 64 E0 D4
```

will reset the Z Flag, because bit 7 in the destination operand is not a 1. The S Flag will be set, and the V Flag will be cleared.

Example: If Register DFH contains the value FFH (11111111B), Register 07H contains the value 1FH, and Register 1FH contains the value BDH (10111101B) (bit 7, bit 5, bit 4, bit 3, bit 2, and bit 0 are being tested if they are 1), the statement:

```
TCM DFH, @07H
OpCode: 65 07 DF
```

will set the Z Flag indicating the tested bits in the destination operand are 1. The S and V Flags are cleared.

Example: If Working Register R13 contains the value F2H (11110010B), the statement:

```
TCM R13, #02H
OpCode: 66 ED, 02
```

tests bit 1 of the destination operand for 1. The Z Flag will be set indicating bit 1 in the destination operand was 1. The S and V Flags are cleared.

Example: If Register 5DH contains A0H, and Register A0H contains 0FH (00001111B), the statement:

```
TCM @5D, #10H
OpCode: 67 5D 10
```

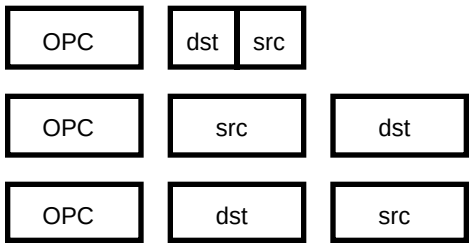
tests bit 4 of the Register A0H for 1. The Z Flag will be reset indicating bit 1 in the destination operand was not 1. The S and V Flags are cleared.

TM
TEST UNDER MASK

TM
Test Under Mask

TM dst, src

Instruction Format:



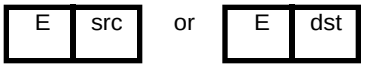
Cycles	OPC (Hex)	Address dst	Mode src
6	72	r	r
6	73	r	Ir
10	74	R	R
10	75	R	IR
10	76	R	IM
10	77	IR	IM

Operation: dst AND src

This instruction tests selected bits in the destination operand for a 0 logical value. The bits to be tested are specified by setting a 1 bit in the corresponding bit position in the source operand (the mask). The TM instruction ANDs the destination operand with the source operand (the mask). The Zero (Z) Flag can then be checked to determine the result. If the Z Flag is set, then the tested bits were 0. When the TM operation is complete, the destination and source operands still contain their original values.

Flags:
C: Unaffected
Z: Set if the result is zero; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Always reset to 0.
D: Unaffected
H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the source or destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Working Register R3 contains 45H (01000101B) and Working Register R7 contains the value 02H (00000010B) (bit 1 is being tested if it is 0), the statement:

TM R3, R7
OpCode: 72 37

will set the Z Flag indicating bit 1 in the destination operand is 0. The V and S Flags are cleared.

TM
TEST UNDER MASK

Example: Working Register R14 contains the value F3H (11110011B), Working Register R5 contains CBH, and Register CBH contains 88H (10001000B) (bit 7 a bit 3 are being tested if they are 0), the statement:

TM R14, @R5
OpCode: 73 E5

will reset the Z Flag, because bit 7 iin the destination operand is not a 0. The S Flag will be set, and the V Flag is cleared.

Example: If Register D4H contains the value 08H (00001000B), and Working Register R0 contains the value 04H (00000100B) (bit 2 is being tested if it is 0), the statement:

TM D4H, R0
OpCode: 74 E0 D4

will set the Z Flag, because bit 2 in the destination operand is a 0. The S and V Flags will be cleared.

Example: If Register DFH contains the value 00H (00000000B), Register 07H contains the value 1FH, and Register 1FH contains the value BDH (10111101B) (bit 7, bit 5, bit 4, bit 3, bit 2, and bit 0 are being tested if they are 0), the statement:

TM DFH, @07H
OpCode: 75 07 DF

will set the Z Flag indicating the tested bits in the destination operand are 0. The S is set, and the V Flag is cleared.

Example: If Working Register R13 contains the value F1H (11110001B), the statement:

TM R13, #02H
OpCode: 76 ED, 02

tests bit 1 of the destination operand for 0. The Z Flag will be set indicating bit 1 in the destination operand was 0. The S and V Flags are cleared.

Example: If Register 5DH contains A0H, and Register A0H contains 0FH (00001111B), the statement:

TM @5D, #10H
OpCode: 77 5D 10

tests bit 4 of the Register A0H for 0. The Z Flag will be set indicating bit 4 in the destination operand was 0. The S and V Flags are cleared.

WDH
WATCH-DOG TIMER ENABLE DURING HALT MODE

WDH
Watch-Dog Timer Enable During HALT Mode

WDH

Instruction Format:

	Cycles	OPC (Hex)
OPC	6	4F

Operation: When this instruction is executed it will enable the WDT (Watch-Dog Timer) during HALT mode. If this instruction is not executed the WDT will stop when entering HALT mode. This instruction does not clear the counter, it just makes it possible to have the WDT function running during HALT mode. A WDH instruction executed without executing WDT (5FH) has no effect.

Flags:
C: Unaffected
Z: Undefined
S: Undefined
V: Undefined
D: Unaffected
H: Unaffected

Note: The WDH instruction should not be used following any instruction in which the condition of the flags is important.

Example: If the WDT is enabled, the statement:

```
WDH
OpCode: .BYTE  4FH
```

will enable the WDT in HALT mode.

Note: This instruction format is valid only for the Z86C04/C08 and Z86E04/E07/E08.

WDT
WATCH-DOG TIMER

WDT
Watch-Dog Timer

WDT

Instruction Format:

	Cycles	OPC (Hex)
OPC	6	5F

Operation: The WDT (Watch-Dog Timer) is a retriggerable one shot timer that will reset the Z8 if it reaches its terminal count. The WDT is initially enabled by executing the WDT instruction. Each subsequent execution of the WDT instruction refreshes the timer and prevents the WDT from timing out.

Flags:
C: Unaffected
Z: Undefined
S: Undefined
V: Undefined
D: Unaffected
H: Unaffected

Note: The WDT instruction should not be used following any instruction in which the condition of the flags is important.

Example: If the WDT is enabled, the statement:

```
WDT
Opcode: .BYTE   5FH
```

refreshes the Watch-Dog Timer.

Example: The first execution of the statement:

```
WDT
OpCode: .BYTE   5FH
```

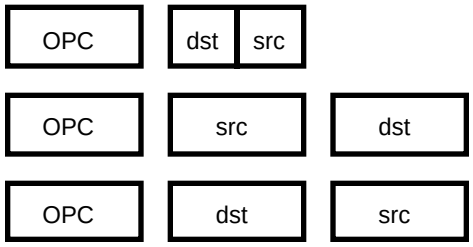
enables the Watch-Dog Timer.

XOR
LOGICAL EXCLUSIVE OR

XOR
Logical Exclusive OR

XOR dst, src

Instruction Format:



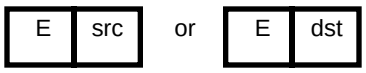
Cycles	OPC (Hex)	Address dst	Mode src
6	B2	r	r
6	B3	r	Ir
10	B4	R	R
10	B5	R	IR
10	B6	R	IM
10	B7	IR	IM

Operation: dst <— dst XOR src

The source operand is logically EXCLUSIVE ORed with the destination operand. The XOR operation results in a 1 being stored in the destination operand whenever the corresponding bits in the two operands are different, otherwise a 0 is stored. The contents of the source operand are not affected.

Flags:
- C: Unaffected
- Z: Set if the result is zero; cleared otherwise.
- S: Set if the result of bit 7 is set; cleared otherwise.
- V: Always reset to 0
- D: Unaffected
- H: Unaffected

Note: Address modes R or IR can be used to specify a 4-bit Working Register. In this format, the source or destination Working Register operand is specified by adding 1110B (EH) to the high nibble of the operand. For example, if Working Register R12 (CH) is the destination operand, then ECH will be used as the destination operand in the OpCode.



Example: If Working Register R1 contains 34H (00111000B) and Working Register R14 contains 4DH (10001101B), the statement:

XOR R1, R14
OpCode: B2 1E

leaves the value BDH (10111101B) in Working Register R1. The Z, and V Flags are cleared, and the S Flag is set.

XOR
LOGICAL EXCLUSIVE OR

Example: If Working Register R4 contains F9H (11111001B), Working Register R13 contains 7BH, and Register 7B contains 6AH (01101010B), the statement:

XOR R4, @R13
OpCode: B3 4D

leaves the value 93H (10010011B) in Working Register R4. The S Flag is set, and the Z, and V Flags are cleared.

Example: If Register 3AH contains the value F5H (11110101B) and Register 42H contains the value 0AH (00001010B), the statement:

XOR 3AH, 42H
OpCode: B4 42 3A

leaves the value FFH (11111111B) in Register 3AH. The S Flag is set, and the C and V Flags are cleared.

Example: If Working Register R5 contains F0H (11110000B), Register 45H contains 3AH, and Register 3A contains 7F (01111111B), the statement:

XOR R5, @45H
OpCode: B5 45 E5

leaves the value 8FH (10001111B) in Working Register R5. The S Flag is set, and the C and V Flags are cleared.

Example: If Register 7AH contains the value F7H (11110111B), the statement:

XOR 7AH, #F0H
OpCode: B6 7A F0

leaves the value 07H (00000111B) in Register 7AH. The Z, V and S Flags are cleared.

Example: If Working Register R3 contains the value 3EH and Register 3EH contains the value 6CH (01101100B), the statement:

XOR @R3, #05H
OpCode: B7 E3 05

leaves the value 69H (01101001B) in Register 3EH. The Z, V, and S Flags are cleared.