Instruction Set H08B User's Manual

H08B Instruction Index

Instruction		Description	Cycles	Status Affected	Ref Page
BYTE-ORIENTED FILE REGISTER OPERATIONS					
ADDC	f,d,a	Add W, F and C and place the result to W or F.	1	C,Z	5
ADDF	f,d,a	Add W and F and place the result to W or F.	1	C,Z	6
ADDL	k	Add constant k and W and place the result to W.	1	C,Z	7
ANDF	f,d,a	AND operation W and F and place the result to W or F.	1	Z	8
ANDL	k	AND operation K and W and place the result to W.	1	Z	9
CLRF	f,a	Clear F	1	None	16
COMF	f,d,a	Complement F and place the result to W or F.	1	Z	17
CPSE	f,a	If F=W, skip the next instruction	1(2)(3)	None	18
CPSG	f,a	If F>W, skip the next instruction.	1(2)(3)	None	19
CPSL	f,a	If F<W, skip the next instruction.	1(2)(3)	None	20
DCF	f,d,a	Subtract 1 of F value, place the result to W or F.	1	C,Z	22
DCSUZ	f,d,a	Subtract 1 of F value, if the result≠0, skip the next instruction and place it to W or F.	1(2)(3)	None	23
DCSZ	f,d,a	Subtract 1 of F value, if the result =0, skip the next instruction and place it to W or F.	1(2)(3)	None	24
INF	f,d,a	Add 1 to F value and store the result to W or F.	1	C,Z	26
INSUZ	f,d,a	Add 1 to F value, if the result≠0, skip the next instruction and place it to W or F.	1(2)(3)	None	27
INSZ	f,d,a	Add 1 to F value, if the result=0, skip the next instruction and place it to W or F.	1(2)(3)	None	28
IORF	f,d,a	Inclusive OR W and F, and place the result to W or F.	1	Z	29
IORL	k	Inclusive OR constant k and W, and place the result to W.	1	Z	30
MVF	f,d,a	Move W value to F (d=1) or move F value to W(d=0).	1	None	32
MVL	k	Move constant k to W.	1	None	33
RETL	k	Put the top stack value to PC and configure W value as k. The main program will execute from current PC value.	2	None	37
RLF	f,d,a	Rotate left F and place the result to W or F.	1	Z	38
RLFC	f,d,a	Rotate F value and C and place the result to W or F.	1	C,Z	39

Remark f Register b Register b bit

n Memory address k 8 bit constant

d Data storage; d = 0 means the data is saved in accumulator W；d = 1 means it is saved in register f.

a Memory address of data storage ,a=0 is saved in memory address (000H~0FFH)

H08B Instruction Fast Index (continued)

Instruction		Description	Cycles	Status Affected	Ref Page
BYTE-ORIENTED FILE REGISTER OPERATIONS					
RRF	f,d,a	Rotate right f and place the result to W or F.	1	Z	40
RRFC	f,d,a	Rotate right f and C and place the result to W or F.	1	C,Z	41
SETF	f,a	Configure F value as 0xFF.	1	None	42
SUBC	f,d,a	Subtract w of F value and reverse C, then place the result to W or F.	1	C,Z	44
SUBF	f,d,a	Subtract w of F value and place to result to W or F.	1	C,Z	45
SUBL	k	Subtract constant k and W and place the result to W.	1	C,Z	46
SWPF	f,d,a	Switch the high 4 bit and low 4 bit value and save the result to W or F.	1	None	47
TFSZ	f,a	Test whether a specific bit value of F is 0. If so, skip the next instruction.	1(2)(3)	None	48
XORF	f,d,a	Exclusive OR W and F and place the result to W or F.	1	Z	49
XORL	k	XOR constant k and W, then places the result to W.	1	Z	50

Remark				
	f	Register	b	Register b bit
	n	Memory address	k	8 bit constant

d Data storage; d = 0 means the data is saved in accumulator W；d = 1 means it is saved in register f.

a Memory address of data storage ,a=0 is saved in memory address (000H~0FFH)

H08B Instruction Fast Index (continued)

Instruction		Description	Cycles	Status Affected	Ref Page
CONTROL OPERATIONS					
CALL	n,s	Save the PC value of next instruction to the uppermost stack layer and jump to address n.	2	None	15
CWDT		Clear watch dog timer to 0	1	TO	21
IDLE		Go to standby mode	1	IDLEB	25
JMP	n	Jump unconditionally to address n.	2	None	31
NOP		Blank instruction	1	None	34
RET	s	Return to main program from vice program and place the uppermost stack value into PC.	2	None	35
RETI	s	Return to main program from interrupt vice program and place the uppermost stack value placed into PC. Main program will be executed from current PC value.	2	GIE	36
SLP	f,a	Go to sleep mode		PD	43
BIT-ORIENTED FILE REGISTER OPERATIONS					
BCF	f,b,a	Configure a specific bit of F as 0.	1	None	10
BSF	f,b,a	Configure a specific bit of F as 1.	1	None	11
BTGF	f,b,a	NOT a specific bit of F.	1	None	12
BTSS	f,b,a	Test whether a specific bit value of F is 1. If so, skip the next instruction.	1(2)(3)	None	13
BTSZ	f,b,a	Test whether a specific bit value of F is 0. If so, skip the next instruction.	1(2)(3)	None	14

Remark	f	Register	b	Register b bit
	n	Memory address	k	8 bit constant
	d	Data storage; d = 0 means the data is saved in accumulator W；d = 1 means it is saved in register f.		
	a	Memory address of data storage ,a=0 saved in memory address (000H~0FFH)		

ADDC ADD w and Carry bit to f

Syntax: ADDC f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (W) + (f) + (Status<C>) → destination

Status Affected: C, Z

Description: Add accumulator W value, register f value and carry flag C together and place the result to d appointed register；

If d = 0, the operation result will be placed into accumulator W ；

If d = 1, the operation result will be placed into register f；

If a = 0, the operation result will be place into RAM address (000H~0FFH).

Words: 1

Cycles: 1

Example 1: ADDC REG, 0, 0

Before Instruction:	**After Instruction:**
W=001H	W=020H
REG(080H)=01FH	REG(080H)=01FH
C=Z=0	C=Z=0

Remark: d=0, the execution result will be placed into accumulator W.

Example 2: ADDC REG, 1, 0

Before Instruction:	**After Instruction:**
W=001H	W=001H
REG(070H)=00EH	REG(070H)=010H
C=1, Z=0	C=Z=0

Remark: d=1, the execution result will be placed into register f.

ADDF ADD w to F

Syntax: ADDF f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (W) + (f) → destination

Status Affected: C, Z

Description: Add accumulator W value and register f value together, then place the result to d appointed register；

If d = 0, the operation result will be placed into accumulator W ；

If d = 1, the operation result will be placed into register f；

If a = 0, the operation result will be place into RAM address (000H~0FFH).

Words: 1

Cycles: 1

Example 1: ADDF REG, 0, 0

Before Instruction:	**After Instruction:**
W=001H	W=020H
REG(080H)=01FH	REG(080H)=01FH
C=Z=0	C=Z=0

Remark: d=0, the execution result will be placed into accumulator W..

Example 2: ADDF REG, 1

Before Instruction:	**After Instruction:**
W=001H	W=001H
REG(080H)=01FH	REG(080H)=020H
C=Z=0	C=Z=0

Remark: d=1, the execution result will be placed into register f. The default value is a=0, if program a=0, then this argument can not be added into this program.

ADDL ADD Literal to w

Syntax: ADDL k

Operands: 0 ≦ k ≦ 255

Operation: (W) + K → W

Status Affected: C, Z

Description: Add accumulator W value and k value, and place the result to accumulator W.

Words: 1

Cycles: 1

Example 1: ADDL 00FH

Before Instruction:	**After Instruction:**
W=001H	W=010H
C=Z=0	C=Z=0

Example 2: ADDL 00FH

Before Instruction:	**After Instruction:**
W=071H	W=080H
C=Z=0	C=Z=0

Example 3: ADDL 00FH

Before Instruction:	**After Instruction:**
W=081H	W=090H
C=Z=0	C=OV=Z=0

Example 4: ADDL 00FH

Before Instruction:	**After Instruction:**
W=0F1H	W=000H
C=Z=0	C=Z=1

Remark: he execution result > 0FFH, carry flag, C=1. If the result is 000H, the zero flag, Z=1.

ANDF AND w with F

Syntax: ANDF f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (W) AND (f) → destination

Status Affected: Z

Description: Logic AND accumulator W value and register f value, and place the result to d appointed register.

If d = 0, place the result to accumulator W ;

If d = 1, place the result to register f ;

IF a = 0, place the result to RAM address (000H~0FFH).

Words: 1

Cycles: 1

Example 1: ANDF REG, 0

Before Instruction:	**After Instruction:**
W=055H	W=000H
REG(080H)=0AAH	REG(080H)=0AAH Z=1, C=0
C=Z=0	

Remark: If the result is 000H, the zero flag, Z=1. If the default value is a=0, when a=0, it is alright to not adding this argument into the program.

Example 2: ANDF REG, 1, 0

Before Instruction:	**After Instruction:**
W=080H	W=080H
REG(070H)=0FFH	REG(070H)=080H
C=Z=0	C=Z=0

ANDL AND Literal with w

Syntax: ANDL k

Operands: 0 ≦ k ≦ 255

Operation: (W) AND k → W

Status Affected: Z

Description: Logic AND accumulator W value and k value, and place the result to accumulator W.

Words: 1

Cycles: 1

Example 1: ANDL 0A0H

Before Instruction:	**After Instruction:**
W=055H	W=000H
C=Z=0	Z=1, C=0

Remark: If the result is 000H, the zero flag bit Z=1.

Example 2: ANDL 0FF0H

Before Instruction:	**After Instruction:**
W=080H	W=080H
C=Z=0	C=Z=0

BCF Bit Clear F

Syntax: BCF f, b, a

Operands: 0 ≦ f ≦ 255 ; 0 ≦ b ≦ 7 ; a € (0)

Operation: 0 → f < b >

Status Affected: None

Description: Clear register f configuration as 0.

Words: 1

Cycles: 1

Example 1: BCF REG,2

Before Instruction:

REG(080H)=1111 1111B

After Instruction:

REG(080H)=1111 1011B

Remark: Clear BIT2of register REG as 0, other bits remain unchanged.

BSF Bit Set F

Syntax: BCF f, b, a

Operands: 0 ≦ f ≦ 255 ; 0 ≦ b ≦ 7 ; a € (0)

Operation: 1 → f < b >

Status Affected: None

Description: Configure register f value as 1.

Words: 1

Cycles: 1

Example 1: BSF REG,2

Before Instruction:

REG(080H)=00000000B

After Instruction:

REG(080H)=00000100B

Remark: Configure BIT2 of REG as 1, other bits remain unchanged.

BTGF Bit ToGgle F

Syntax: BTGF f, b, a

Operands: 0 ≦ f ≦ 255 ; 0 ≦ b ≦ 7 ; a € (0)

Operation: $\overline{\textbf{(f < b >)}} \rightarrow f < b >$

Status Affected: None

Description: Complement a specific bit of register.

Words: 1

Cycles: 1

Example 1: BTGF REG, 7, 0

Before Instruction:	**After Instruction:**
REG(080H)=0111 1111B	REG(080H)=1111 1111B

Remark: Complement BIT7 of register, REG.

BTSS Bit Test and Skip if Set

Syntax: BTSS f, b, a

Operands: 0 ≦ f ≦ 255 ; 0 ≦ b ≦ 7 ; a € (0)

Operation: skip if (f < b >)=1

Status Affected: None

Description: Compare whether a specific bit of register is 1. Skip the next instruction if the bit is 1. If not, execute the next instruction.

Words: 1
Cycles: 1(2)(3)

Example 1:

```
BTSS    REG, 7, 0
MVL     001H
NOP
```

Before Instruction:

WREG(02CH)=000H

REG(080H)=0FFH

After Instruction:

WREG(02CH)=000H
REG(080H)=0FFH

Remark: BIT7of register, REG is 1, so skip the next instruction..

Example 2:

```
BTSS            REG, 7, 0
MVL     001H
NOP
```

Before Instruction:

WREG(02CH)=000H

REG(080H)=07FH

After Instruction:

WREG(02CH)=001H

REG(080H)=07FH

Remark: BIT7of register, REG is 0, so the program can execute the next instruction.

BTSZ Bit Test and Skip if Zero

Syntax: BTSZ f, b, a

Operands: 0 ≦ f ≦ 255 ; 0 ≦ b ≦ 7 ; a € (0)

Operation: skip if (f < b >)=0

Status Affected: None

Description: Compare whether a specific bit of register is 0. Skip the next instruction if the bit is 0. If not, execute the next instruction.

Words: 1
Cycles: 1(2)(3)

Example 1:

```
BTSZ        REG, 7, 0
MVL     001H
NOP
```

Before Instruction:

WREG(02CH)=000H

REG(080H)=0FFH

After Instruction:

WREG(02CH)=001H
REG(080H)=0FFH

Remark: BIT7 of register, REG is not 0, so the program can execute the next instruction.

Example 2:

```
BTSZ        REG, 7, 0
MVL     001H
NOP
```

Before Instruction:

WREG(02CH)=000H

REG(080H)=07FH

After Instruction:

WREG(02CH)=000H
REG(080H)=07FH

Remark: BIT7of register, REG is 0, so skip the next instruction.

CALL subroutine CALL

Syntax: CALL n

Operands: 0 ≦ n ≦ 2047(07FFH);

Operation: (PC) + 1 → TOS, n → PC,

Status Affected: STKPTR<STKFL>, STKPTR<STKOV>, Pstatus<SKERR>.

Description: Call vice program, the maximum call range is 2Kbytes memory range.

If the stack is the last layer of the specific product after calling vice program, flag bit, STKFL will be configured as 1.

Under SBMSET1<7>=0 condition, STKOV flag will be configured as 1 when stack full and CALL instruction is executed, SKERR will be configured as 1. PC operates normally.

Under SBMSET1<7>=1 condition, STKOV flag will be configured as 1 when stack full and CALL instruction is executed, SKERR will be configured as 1. IC will be reset and PC will return to 000H.

When STKFL or STKOV occurs, either one is cleared; the other one will be cleared.

TOS is not readable for users.

Words: 2

Cycles: 2

Example 1:

```
LABEL:   CALL     NEXT
         .
         .
         .
NEXT:    NOP
```

Before Instruction:

PC = address (LABEL)

After Instruction:

PC= address (NEXT)

TOS= address (LABEL + 2)

CLRF CLeaR F

Syntax: CLRF f, a

Operands: 0 ≦ f ≦ 255 ; a € (0)

Operation: 000H → f

Status Affected: None

Description: Clear register f value.

Words: 1

Cycles: 1

Example 1: CLRF REG,0

Before Instruction:

REG(080H)=055H

After Instruction:

REG(080H)=000H

Remark: Clear register, REG value as 0.

COMF COMplement F

Syntax: COMF f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: $\overline{(f)}$ → destination

Status Affected: Z

Description: The value of register f is complemented and the result is stored in d appointed register.
If d = 0, the result is stored in accumulator W ;
If d = 1, the result is stored in register f ;
If a = 0, the result is stored in RAM address (000H~0FFH).

Words: 1

Cycles: 1

Example 1: COMF REG, 0, 0

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=000H
REG(080H)=0FFH	REG(080H)=0FFH
Z=0	Z=1

Example 2: COMF REG, 1, 0

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=055H
REG(070H)=055H	REG(070H)=0AAH
Z=0	Z=0

CPSE ComPare f with w, Skip if f Equal w

Syntax: CPSE f, a

Operands: 0 ≦ f ≦ 255 ; a € (0)

Operation: skip if (f) = (W)

Status Affected: None

Description: Compare register value and accumulator W value. If the value is equivalent, skip the next value. If the value is > or <, continue the next instruction.

Words: 1

Cycles: 1(2)

Example 1:

```
CPSE    REG, 0
MVL     001H
NOP
```

Before Instruction:

WREG(02CH)=000H

REG(080H)=000H

After Instruction:

WREG(02CH)=000H
REG(080H)=000H

Remark: The value of register f and accumulator W is the same, skip the next instruction.

Example 2:

```
CPSE    REG, 0
MVL     001H
NOP
```

Before Instruction:

WREG(02CH)=000H

REG(070H)=07FH

After Instruction:

WREG(02CH)=001H
REG(070H)=07FH

Remark: The value of register f and accumulator W is different, continue the next instruction.

CPSG ComPare f with w, Skip if f Greater than w

Syntax: CPSG f, a

Operands: 0 ≦ f ≦ 255 ; a € (0)

Operation: skip if (f) > (W)

Status Affected: None

Description: Compare the register value and accumulator W value.
If register value is greater than accumulator W value, skip the next instruction..
If register value is smaller than accumulator W value, continue the next instruction.

Words: 1

Cycles: 1(2)

Example 1:

```
CPSG    REG, 0
MVL     00FH
NOP
```

Before Instruction:

WREG(02CH)=005H

REG(080H)=006H

After Instruction:

WREG(02CH)=005H

REG(080H)=006H

Remark: The value of register f is greater than accumulator W content, skip the next instruction.

Example 2:

```
CPSG    REG, 0
MVL     00FH
NOP
```

Before Instruction:

WREG(02CH)=005H

REG(070H)=005H

After Instruction:

WREG(02CH)=00FH
REG(070H)=005H

Remark: The value of register f equals to accumulator W content, continue the next instruction.

CPSL ComPare f with w, Skip if f Less than w

Syntax: CPSL f, a

Operands: 0 ≦ f ≦ 255 ; a € (0)

Operation: skip if (f) < (W)

Status Affected: None

Description: Compare the register value and accumulator W value.
If register value is smaller than accumulator W value, skip the next instruction.
If register value is greater or equivalent to accumulator value, continue the next instruction.

Words: 1

Cycles: 1(2)

Example 1:

```
CPSL    REG, 0
MVL     00FH
NOP
```

Before Instruction:

WREG(02CH)=005H

REG(080H)=004H

After Instruction:

WREG(02CH)=005H
REG(080H)=004H

Remark: Register f value is smaller than accumulator W value, skip the next instruction.

Example 2:

```
CPSL    REG, 0
MVL     00FH
NOP
```

Before Instruction:

WREG(02CH)=005H

REG(070H)=005H

After Instruction:

WREG(02CH)=00FH
REG(070H)=005H

Remark: Register f value is equivalent to accumulator value, continue the next instruction.

CWDT Clear WatchDog Timer

Syntax: CWDT

Operands: None

Operation: 000H → Watch dog counter

Status Affected: Pstatus<TO

Description: Clear CWDT value as 0

Words: 1

Cycles: 1

Example 1: CWDT

Before Instruction:	**After Instruction:**
WDT counter = ???	WDT counter = 000H Pstatus<TO> = 0

DCF DeCrement F

Syntax: DCF f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f) – 1 → destination

Status Affected: C, Z

Description: Subtract 1 of register f value, and place back the result to d appointed register.
If d = 0, the result is placed to accumulator W ;
If d = 1, the result is placed to register f ;
If a = 0, the result is placed to RAM address (000H~0FFH).

Words: 1

Cycles: 1

Example 1: DCF REG, 0, 0

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=0FEH
REG(080H)=0FFH	REG(080H)=0FFH
C=Z=0	C= 1, Z=0

Remark: If C has not been borrowed, C=1 ;

Example 2: DCF REG, 1, 0

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=055H
REG(070H)=000H	REG(070H)=0FFH
C=Z=0	C=Z=0

Remark: C has been borrowed, C=0 ;

Example 3: DCF REG, 1, 0

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=055H
REG(080H)=080H	REG(080H)=07FH
C=Z=0	C=1, Z=0

Remark: C has not been borrowed, C=1

Example 4: DCF REG, 0, 0

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=000H
REG(080H)=001H	REG(080H)=001H
C=Z=0	C=Z=1

Remark: C has not been borrowed, C=1 ; the result after execution is 0, so Z=1.

DCSUZ DeCrement f, Skip if Un-Zero

Syntax: DCSUZ f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f) – 1 → destination, skip if destination≠0

Status Affected: None

Description: Compare the decremented register value with 0. If register value is not equal to 0, skip the next instruction. If the value is 0, continue the next instruction and place the result to d appointed register.
If d = 0, the result is placed in accumulator W ;
If d = 1, the result is placed in register f ;
If a = 0, the result is placed in RAM address (000H~0FFH).

Words: 1

Cycles: 1(2)(3)

Example 1:

```
DCSUZ   REG, 1, 0
MVL     00AH
NOP
```

Before Instruction:	**After Instruction:**
WREG(02CH)=00FH	WREG(02CH)=00AH
REG(080H)=001H	REG(080H)=000H

Remark: If the result is 0, continue to execute the next program, and the result will be placed in register REG.

Example 2:

```
DCSUZ   REG, 0, 0
MVL     00AH
NOP
```

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=0FFH
REG(070H)=000H	REG(070H)=000H

Remark: If the result is not 0, skips the next instruction and places the result to accumulator W.

DCSZ DeCrement f, Skip if Zero

Syntax: DCSZ f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f) – 1 → destination, skip if destination=0

Status Affected: None

Description: Compare the decremented register value with 0. If the register value equals to 0, skip the next instruction. If not, continue the next instruction and place the result back to d appointed register.
If d = 0, place the result to accumulator W ;
If d = 1, place the result to register f ;
If a = 0, place the result to RAM address (000H~0FFH).

Words: 1

Cycles: 1(2)(3)

Example 1:

```
DCSZ    REG, 0, 0
MVL     00AH
NOP
```

Before Instruction:

WREG(02CH)=00FH

REG(080H)=001H

After Instruction:

WREG(02CH)=000H
REG(080H)=001H

Remark: The result is 0, so skip the next instruction.

Example 2:

```
DCSZ    REG, 1, 0
MVL     00AH
NOP
```

Before Instruction:

WREG(02CH)=055H

REG(070H)=000H

After Instruction:

WREG(02CH)=00AH
REG(070H)=0FFH

Remark: The result is not 0, so continue the next program and place the result to register REG.

IDLE IDLE mode

Syntax: IDLE

Operands: None

Operation: CPU Halt

Status Affected: Pstatus<IDLEB>

Description: CPU accesses to idle mode, program blank instruction executes action.
It is recommended to add NOP instruction after IDLE instruction.

Words: 1

Cycles: 1

Example 1: IDLE
NOP

Before Instruction:

Pstatus<IDLEB>=0

After Instruction:

Pstatus<IDLEB>=1

IDLE
NOP………program break here

INF INcrement F

Syntax: INF f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f) ＋ 1 → destination

Status Affected: C, Z

Description: Add 1 to the content of register f and place the result to d appointed register.
If d = 0, the result is placed to accumulator W ；
If d = 1, the result is placed to register f ；
If a = 0, the result is placed to RAM address (000H~0FFH).

Words: 1

Cycles: 1

Example 1: INF REG, 0, 0

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=000H
REG(080H)=0FFH	REG(080H)=0FFH
C=Z=0	C=Z=1

Remark: C is carried, so C=1. After execution, the result equals to 0, so Z=1.

Example 2: INF REG, 1, 0

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=055H
REG(070H)=00FH	REG(070H)=010H
C=Z=0	C=Z=0

Example 3: INF REG, 1, 0

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=055H
REG(080H)=07FH	REG(080H)=080H
C=Z=0	C=Z=0

INSUZ INcrement f, Skip if Un-Zero

Syntax: INSUZ f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f) ＋ 1 → destination, skip if destination≠0

Status Affected: None

Description: Compare the register incremented value with 0. If the register value is not 0, then skip the next instruction. If the value equals to 0, then continue executing next instruction and store the result to d appointed register.
If d = 0, the result is placed to accumulator W ；
If d = 1, the result is placed to register f ；
If a = 0, the result is placed to RAM address (000H~0FFH).

Words: 1

Cycles: 1(2)(3)

Example 1:

```
INSUZ     REG, 1, 0
MVL       00AH
NOP
```

Before Instruction:	**After Instruction:**
WREG(02CH)=00FH	WREG(02CH)=00AH
REG(080H)=0FFH	REG(080H)=000H

Remark: The result is 0, so continue the next program. The result will be placed back to register REG.

Example 2:

```
INSUZ     REG, 0, 0
MVL       00AH
NOP
```

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=001H
REG(070H)=000H	REG(070H)=000H

Remark: The result is not 0, so skip the next instruction. The result will be stored in accumulator W.

INSZ INcrement f, Skip if Zero

Syntax: INSZ f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f) ＋ 1 → destination, skip if destination=0

Status Affected: None

Description: Compare the incremented register value with 0. If the value is 0, skip the next instruction. If not equals to 0, continue the next instruction and store the result to d appointed register.
If d = 0, the result is placed to accumulator W ;
If d = 1, the result is placed to register f ;
If a = 0, the result is placed to RAM address (000H~0FFH).

Words: 1

Cycles: 1(2)(3)

Example 1:

```
INSZ    REG, 0, 0
MVL     00AH
NOP
```

Before Instruction:

WREG(02CH)=00FH

REG(080H)=0FFH

After Instruction:

WREG(02CH)=000H
REG(080H)=0FFH

Remark: The result is 0, skip the next instruction.

Example 2:

```
INSZ    REG, 1, 0
MVL     00AH
NOP
```

Before Instruction:

WREG(02CH)=055H

REG(070H)=000H

After Instruction:

WREG(02CH)=00AH
REG(070H)=001H

Remark: The result is not 0, so continue the next instruction and place the result to register REG.

IORF Inclusive OR w with F

Syntax: IORF f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (W) OR (f) → destination

Status Affected: Z

Description: Inclusive OR accumulator W value and register f value and place the result to d appointed register.
If d = 0, the result is placed to accumulator W ;
If d = 1, the result is placed to register f ;
If a = 0, the result is placed to RAM address (000H~0FFH).

Words: 1

Cycles: 1

Example 1: IORF REG, 0, 0

Before Instruction:

WREG(02CH)=055H

REG(080H)=0AAH

Z=0

After Instruction:

WREG(02CH)=0FFH

REG(080H)=0AAH

Z=0

Example 2: IORF REG, 1, 0

Before Instruction:

WREG(02CH)=00FH

REG(070H)=0F0H

Z=0

After Instruction:

WREG(02CH)=00FH

REG(070H)=0FFH

Z=0

IORL Inclusive OR Literal with w

Syntax: IORL k

Operands: 0 ≦ k ≦ 255

Operation: (W) OR k → W

Status Affected: Z

Description: Inclusive OR accumulator W value and k and place the result back to accumulator W.

Words: 1

Cycles: 1

Example 1: IORL 055H

Before Instruction:

WREG(02CH)=0AAH

Z=0

After Instruction:

WREG(02CH)=0FFH

Z=0

Example 2: IORL 000H

Before Instruction:

WREG(02CH)=000H

Z=0

After Instruction:

WREG(02CH)=000H

Z=1

Remark: The result is 0, so Z=1.

JMP unconditional JuMP

Syntax: JMP n

Operands: 0 ≦ n ≦ 2047(07FFH)

Operation: n → PC

Status Affected: None

Description: Unconditionally jump to the appointed address n.

Words: 2

Cycles: 2

Example 1:

```
LABEL:   JMP NEXT
         .
         .
         .
NEXT:    NOP
```

Before Instruction:

PC = address (LABEL)

After Instruction:

PC= address (NEXT)

MVF MoVe F to w or MoVe w to F

Syntax: MVF f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f) → W, or (W) → f

Status Affected: None

Description: Move register f value to accumulator W; or move accumulator W value to register f.
If d = 0, it means moving register f value to accumulator W ;
If d = 1m, it means moving accumulator W value to register f ;
a = 0 or a =1 configuration must be determined by register f address of RAM :
If a = 0, it means register f existing in the appointed RAM address of 000H~0FFH.

Words: 1

Cycles: 1

Example 1: MVF REG, 0, 0

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=0AAH
REG(080H)=0AAH	REG(080H)=0AAH
REG1(170H)=0FFH	REG1(170H)=0FFH

Remark: d=0, it means moving register REG value to accumulator W.

Example 2: MVF REG1, 1, 0

Before Instruction:	**After Instruction:**
WREG(02CH)=055H	WREG(02CH)=055H
REG1(170H)=0FFH	REG1(170H)=055H
REG(080H)=0AAH	REG(080H)=0AAH

Remark: d=1, it means moving accumulator W value to register f.

MVL MoVe Literal to w

Syntax: MVL k

Operands: 0 ≦ k ≦ 255

Operation: k → W

Status Affected: None

Description: Move constant k to accumulator W.

Words: 1

Cycles: 1

Example 1: MVL 0FFH

Before Instruction:

WREG(02CH)=000H

After Instruction:

WREG(02CH)=0FFH

NOP No OPeration

Syntax: NOP

Operands: None

Operation: No operation

Status Affected: None

Description: No operation is executed, only delay for 1 instruction time.

Words: 1

Cycles: 1

Example 1: NOP

Remark: Blank instruction, only executing the delay time of 1 instruction cycle.

RET RETurn from subroutine

Syntax: RET

Operands: None

Operation: (TOS) → PC,

Status Affected: STKPTR<STKUN>, Pstatus<SKERR>

Description: Leave vice program and store stack pointer register value to PC.
When vice program is not called and STKPTR=000H, executes instruction RET may result to IC reset and STKUN flag may be configured as 1.
SKERR flag will be configured as 1.
TOS is not readable for users.

Words: 1

Cycles: 2

Example 1: RET

Before Instruction:	**After Instruction:**
None	PC=TOS

RETI RETurn from Interrupt

Syntax: RETI

Operands: None

Operation: (TOS) → PC, 1 → GIE

Status Affected: GIE, STKPTR<STKUN>, Pstatus<SKERR>

Description: Return form interrupt and store the stack pointer register value to PC. Interrupt enable pin is configured again as 1.
When vice program is not called and STKPTR=000H, execute RETI instruction will lead to IC reset. STKUN flag will be configured as 1 so as SKERR flag.
TOS is not readable for users.

Words: 1

Cycles: 2

Example 1: RETI 1

Before Instruction:
None

RETL RETurn Literal to w

Syntax: RETL k

Operands: 0 ≦ k ≦ 255

Operation: k → W, (TOS) → PC

Status Affected: STKPTR<STKUN>, Pstatus<SKERR>

Description: Return from vice program to main program. While returning the instruction, constant k value will be loaded to accumulator W.
This instruction is often implemented in look-up-table function.
When vice program is not called and STKPTR=000H, executing RETL instruction may reset the IC. STKUN flag will be configured as 1, so as SKERR flag.

Words: 1

Cycles: 2

Example 1:

```
LABEL:  MVL     001H
        CALL    TABLE
        .
        .
        .
TABLE:  ADDF    PCLATL, 1, 0
        RETL        055H
        RETL        0AAH
```

Before Instruction:

WREG(02CH)=001H

After Instruction:

WREG(02CH)=0AAH

Remark: While returning to main program, constant k will be loaded to accumulator W.
This example presents how to write Offset value of PCLATL to determine the value of TABLE.

RLF Rotate Left F (no carry)

Syntax: RLF f, d, a

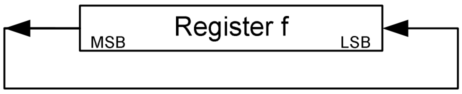
Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f<n>) → destination <n+1 >,

(f<7>) → destination < 0 >

Status Affected: Z

Description: Rotate left f value.
If d = 0, the result is placed to accumulator W ;
If d = 1, the result is placed to register f ;
If a = 0, the result is placed to RAM address (000H~0FFH).



Words: 1

Cycles: 1

Example 1: RLF REG, 1, 0

Before Instruction:

WREG(02CH)=00FH

REG(080H)=0AAH

Z=0

After Instruction:

WREG(02CH)=00FH

REG(080H)=055H
Z=0

Example 2: RLF REG, 0, 0

Before Instruction:

WREG(02CH)=00FH

REG(07FH)=000H

Z=0

After Instruction:

WREG(02CH)=000H

REG(07FH)=000H
Z=1

Example 3: RLF REG, 0, 0

Before Instruction:

WREG(02CH)=00FH

REG(080H)=055H

Z=0

After Instruction:

WREG(02CH)=0AAH

REG(080H)=055H
Z=0

RLFC Rotate Left F through Carry

Syntax: RLFC f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

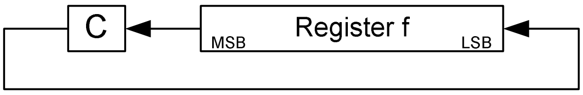
Operation: (f<n>) → destination <n+1 >,

(f<7>) → Status< C >,

Status< C > → destination < 0 >

Status Affected: C, Z

Description: Rotate left register f value and carry flag C.
If d = 0, the result is placed to accumulator W ;
If d = 1, the result is placed to register f ;
If a = 0, the result is placed to RAM address (000H~0FFH).



Words: 1

Cycles: 1

Example 1: RLFC REG, 1, 0

Before Instruction:

WREG(02CH)=00FH

REG(080H)=0AAH

C=Z=0

After Instruction:

WREG(02CH)=00FH

REG(080H)=054H
C=1, Z=0

Example 2: RLFC REG, 0, 0

Before Instruction:

WREG(02CH)=0FH

REG(070H)=0EAH

C=Z=0

After Instruction:

WREG(02CH)=0D4H

REG(070H)=0EAH
C=1, Z=0

Example 3: RLFC REG, 1, 0

Before Instruction:

WREG(02CH)=00FH

REG(070H)=080H

C=Z=0

After Instruction:

WREG(02CH)=00FH

REG(070H)=000H
C=Z=1

RRF Rotate Righ F (no carry)

Syntax: RRF f, d, a

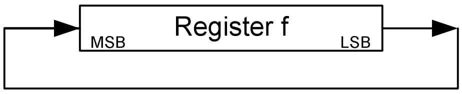
Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f<n>) → destination <n - 1 >,

(f<0>) → destination < 7 >

Status Affected: Z

Description: Rotate right the register f value.
If d = 0, the result is placed to accumulator W ;
If d = 1, the result is placed to register f ;
If a = 0, the result is placed to RAM address (000H~0FFH).



Words: 1

Cycles: 1

Example 1: RRF REG, 1, 0

Before Instruction:	After Instruction:
WREG(02CH)=00FH	WREG(02CH)=00FH
REG(080H)=0AAH	REG(080H)=055H
Z=0	Z=0

Example 2: RRF REG, 0, 0

Before Instruction:	After Instruction:
WREG(02CH)=00FH	WREG(02CH)=000H
REG(07FH)=000H	REG(07FH)=000H
Z=0	Z=1

Example 3: RRF REG, 0, 0

Before Instruction:	After Instruction:
WREG(02CH)=00FH	WREG(02CH)=0AAH
REG(080H)=055H	REG(080H)=055H
Z=0	Z=0

RRFC Rotate Right F through Carry

Syntax: RRFC f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

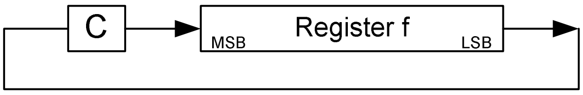
Operation: (f<n>) → destination <n-1 >,

(f<0>) → Status< C >,

Status< C > → destination < 7 >

Status Affected: C, Z

Description: Rotate right the register f value and carry flag C.
If d = 0, the result is placed to accumulator W ；
If d = 1, the result is placed to register f ；
If a = 0, the result is placed to RAM address (000H~0FFH).



Words: 1

Cycles: 1

Example 1: RRFC REG, 1, 0

Before Instruction:

WREG(02CH)=00FH

REG(080H)=0AAH

C=Z=0

After Instruction:

WREG(02CH)=00FH

REG(080H)=055H
C=Z=0

Example 2: RRFC REG, 0, 0

Before Instruction:

WREG(02CH)=00FH

REG(07FH)=055H

C=1, Z=0

After Instruction:

WREG(02CH)=0AAH

REG(07FH)=055H
C=1, Z=0

Example 3: RRFC REG, 1, 0

Before Instruction:

WREG(02CH)=00FH

REG(07FH)=001H

C=Z=0

After Instruction:

WREG(02CH)=00FH

REG(07FH)=000H
C=Z=1

SETF SET F

Syntax: SETF f, a

Operands: 0 ≦ f ≦ 255 ; a € (0)

Operation: 0FFH → f

Status Affected: None

Description: Configure all the contents of register f to 1.
a = 0 or a =1 is determined by RAM address of register f：
If a = 0, it means that register f exists in the appointed RAM address of 000H to 0FFH

Words: 1

Cycles: 1

Example 1: SETF REG, 0

Before Instruction:

WREG(02CH)=00FH

REG(080H)=0AAH

After Instruction:

WREG(02CH)=00FH
REG(080H)=0FFH

SLP enter SLeeP mode

Syntax: SLP

Operands: None

Operation: 1 → PD

Status Affected: Pstatus<PD>

Description: CPU accesses into sleep mode, oscillator stop operating.

Words: 1

Cycles: 1

Example 1: SLP
NOP

Before Instruction:	**After Instruction:**
PD=0	PD=1

SUBC SUBtract w from f with Carry

Syntax: SUBC f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f) － (W) － $\overline{(C)}$ → destination

Status Affected: C, Z

Description: Subtract accumulator W, carry flag C's reversed value of register f and place the result to d.
If d = 0, the result is placed to accumulator W ;
If d = 1, the result is placed to register f ;
If a = 0, the result is placed to RAM address (000H~0FFH).

Words: 1

Cycles: 1

Example 1: SUBC REG, 0, 0

Before Instruction:

WREG=001H

REG(080H)=001H

C=1, Z=0

After Instruction:

WREG=000H

REG(080H)=001H
C=Z=1

Remark: C has not been borrowed, so C=1. The result is 0, so Z=1.

Example 2: SUBF REG, 1, 0

Before Instruction:

WREG=000H

REG(07FH)=080H

C=Z=0

After Instruction:

WREG=000H

REG(07FH)=07FH
C=1, Z=0

Remark: C has not been borrowed, so C=1.

SUBF SUBtract w from F

Syntax: SUBF f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f) － (W) → destination

Status Affected: C, Z

Description: Subtract accumulator W value of register f and place the result to d.
If d = 0, the result is placed to accumulator W ；
If d = 1, the result is placed to register f ;
If a = 0, the result is placed to RAM address (000H~0FFH).

Words: 1

Cycles: 1

Example 1: SUBF REG, 0, 0

Before Instruction:	**After Instruction:**
WREG=001H	WREG=000H
REG(080H)=001H	REG(080H)=001H
C=Z=0	C=Z=1

Remark: C has not been borrowed, so C=1. The result is 0, so Z=1.

Example 2: SUBF REG, 1, 0

Before Instruction:	**After Instruction:**
WREG=001H	WREG=001H
REG(07FH)=080H	REG(07FH)=07FH
C=Z=0	C=1, Z=0

Remark: C has not been borrowed, so C=1 ; .

SUBL SUBtract w from Literal

Syntax: SUBL k

Operands: 0 ≦ k ≦ 255

Operation: K − (W) → W

Status Affected: C, Z

Description: Subtract constant k and accumulator W value and place the result back to accumulator W.

Words: 1

Cycles: 1

Example 1: SUBL 001H

Before Instruction:	**After Instruction:**
WREG=001H	WREG=000H
C=Z=0	C=Z=1

Remark: C has not been borrowed, so C=1. The result is 0, so Z=1.

Example 2: SUBL 080H

Before Instruction:	**After Instruction:**
WREG=001H	WREG=07FH
C=Z=0	C=1, Z=0

Remark: C has not been borrowed, so C=1.

Example 3: SUBL 07FH

Before Instruction:	**After Instruction:**
WREG=0FFH	WREG=080H
C=Z=0	C= Z=0

Remark: C has been borrowed, so C=0.

Example 4: SUBL 000H

Before Instruction:	**After Instruction:**
WREG=001H	WREG=0FFH
C=Z=0	C=Z=0

Remark: C has been borrowed, so C=0

SWPF SWaP F

Syntax: SWPF f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (f<3:0>) → destination<7:4>

(f<7:4>) → destination<3:0>

Status Affected: None

Description: Switch high and low 4 bit value of register f.
If d = 0, the result is placed to accumulator W ；
If d = 1, the result is placed to register f ；
If a = 0, the result is placed to RAM address (000H~0FFH).

Words: 1

Cycles: 1

Example 1: SWPF REG, 1, 0

Before Instruction:	**After Instruction:**
WREG=001H	WREG=001H
REG(080H)=05AH	REG(080H)=0A5H

TFSZ Test F, Skip if Zero

Syntax: TFSZ f, a

Operands: 0 ≦ f ≦ 255 ; a € (0)

Operation: skip if f = 0

Status Affected: None

Description: If register f value is 0, skip the next instruction. If the value is unequal to 0, the next instruction is executed.

a = 0 or a =1 is determined by RAM address of register f：

If a = 0, it means that register f exists in appointed RAM address of 000H to 0FFH.

Words: 1

Cycles: 1(2)(3)

Example 1:

```
TFSZ    REG, 0
MVL     00FH
NOP
```

Before Instruction:

WREG(02CH)=005H

REG(080H)=000H

After Instruction:

WREG(02CH)=005H
REG(080H)=000

Remark: Register f value is 0, skip the next instruction.

Example 2:

```
TFSZ    REG, 0
MVL     00FH
NOP
```

Before Instruction:

WREG(02CH)=005H

REG(070H)=001H

After Instruction:

WREG(02CH)=00FH
REG(070H)=001H

Remark: Register f value is not 0, continue the next instruction.

XORF eXclusive OR w with F

Syntax: XORF f, d, a

Operands: 0 ≦ f ≦ 255 ; d € (0, 1) ; a € (0)

Operation: (W) XOR (f) → destination

Status Affected: Z

Description: Exclusive OR the constant k and accumulator w and place the result back to d.
If d = 0, the result is placed to accumulator w；
If d = 1, the result is placed to register f；
If a = 0, the result is placed to RAM address (000H~0FFH).

Words: 1

Cycles: 1

Example 1: XORF REG, 0, 0

Before Instruction:	After Instruction:
WREG(02CH)=0AAH	WREG(02CH)=0FFH
REG(080H)=055H	REG(080H)=055H
Z=0	Z=0

Remark: XOR: if both values equal, the result is 0；if both values are unequal, the result is 1.

Example 2: XORF REG, 1, 0

Before Instruction:	After Instruction:
WREG(02CH)=0FFH	WREG(02CH)=0FFH
REG(070H)=0FFH	REG(070H)=000H
Z=0	Z=1

Remark: The result is 0, so Z=1.
XOR: if both values equal, the result is 0；if both values are unequal, the result is 1.

Example 3: XORF REG, 0, 0

Before Instruction:	After Instruction:
WREG(02CH)=000H	WREG(02CH)=000H
REG(080H)=000H	REG(080H)=000H
Z=0	Z=1

Remark: The result is 0, so Z=1.
XOR: if both values equal, the result is 0；if both values are unequal, the result is 1.

XORL eXclusive OR Literal with w

Syntax: XORL k

Operands: 0 ≦ f ≦ 255

Operation: (W) XOR k → W

Status Affected: Z

Description: Exclusive OR the constant k and accumulator w and place the result back to accumulator W.

Words: 1

Cycles: 1

Example 1: XORL 055H

Before Instruction:

WREG(02CH)=0AAH

Z=0

After Instruction:

WREG(02CH)=0FFH
Z=0

Remark: XOR: if both values equal, the result is 0；if both values are unequal, the result is 1.

Example 2: XORL 0FFH

Before Instruction:

WREG(02CH)=0FFH

Z=0

After Instruction:

WREG(02CH)=000H
N=0

Remark: The result is 0, so Z=1.
XOR: if both values equal, the result is 0；if both values are unequal, the result is 1.

Example 3: XORL 000H

Before Instruction:

WREG(02CH)=000H

Z=0

After Instruction:

WREG(02CH)=000H
Z=1

Remark: The result is 0, so Z=1.
XOR: if both values equal, the result is 0；if both values are unequal, the result is 1.

REVISION HISTORY

以下描述本文件差異較大的地方，而標點符號與字形的改變不在此描述範圍。

Version	Page	Summary
V01	ALL	First Version
V02	ALL	Layout revision Add in Page 21 Pstatus<TO> and Example description. Add in Page 44 Pstatus<PD>.