CHAPTER 14 ADDRESS GENERATION

14.1 Instruction Address

In addition to automatic address incrementation each time an instruction is executed, a number of instruction execution order control modes can be used, as described below.

14.1.1 Direct addressing

Two-byte or four-byte immediate data among the instruction's bytes is directly loaded into the PC or both the PS and PC to become a branch address.

This mode is used to execute the following instructions.

```
CALL   far-proc
CALL   memptr16
CALL   memptr32
BR     far-label
BR     memptr16
BR     memptr32
```

14.1.2 Relative addressing

One-byte or two-byte immediate data among the instruction's bytes is added to the PC as a signed displacement value to become a branch address.

When eight-bit displacement is given, a sign extension is made and the resulting 16-bit data is added to the PC.

The PC contents when the addition is made indicate the starting address of the next instruction.

This mode is used to execute the following instructions.

```
CALL   near-proc
BR     near-label
BR     short-label
Conditional branch instruction    short-label
```

14.1.3 Register addressing

The contents of any desired 16-bit register specified in the 3-bit register specification field among the instruction's bytes are loaded into the PC as a branch address.

Unlike data, all eight of the 16-bit registers (AW, BW, CW, DW, IX, IY, SP, and BP) can be used.

This mode is used to execute the following instructions.

```
                      Description example
CALL   regptr16       CALL   AW
BR     regptr16       BR     BW
```

14.1.4 Register indirect addressing

The word or double word contents of the memory addressed by the 16-bit register (IX, IY, or BW) specified in the register specification field among the instruction's bytes are loaded into the PC or both the PC and PS as a branch address.

```
                          Description example
CALL   memptr16       CALL   WORD     PTR [IX]
CALL   memptr32       CALL   DWORD    PTR [IY]
BR     memptr16       BR     WORD     PTR [BW]
BR     memptr32       BR     DWORD    PTR [IX]
```

Remark When WORD PTR is described, memptr16 opcode is generated by the assembler. When DWORD PTR is described, memptr32 opcode is generated by the assembler.

14.1.5 Index addressing

One-byte or two-byte immediate data among the instruction's bytes is the signed displacement value, and is added to the 16-bit register (IX or IY) which serves as an index register. The word or double word contents of the memory addressed by the result are loaded into the PC as a branch address.

This mode is used to execute the following instructions.

```
                        Description example
CALL   memptr16       CALL   var [IX] [2]
CALL   memptr32       CALL   var [IY]
BR     memptr16       BR     var [IY]
BR     memptr32       BR     var [IX+4]
```

Remark If variable var has a word attribute, memptr16 opcode is generated by the assembler. If it has a double word attribute, memptr32 opcode is generated by the assembler.

14.1.6 Based addressing

One-byte or two-byte data among the instruction's bytes is the signed displacement value and is added to the 16-bit register (BP or BW) which serves as a base register. The word or double word contents of the memory addressed by the result are loaded into the PC as a branch address.

This mode is used to execute the following instructions.

		Description example	
CALL	memptr16	CALL	var [BP+2]
CALL	memptr32	CALL	var [BP]
BR	memptr16	BR	var [BW] [2]
BR	memptr32	BR	var [BP]

Remark If variable var has a word attribute, memptr16 opcode is generated by the assembler. If it has a double word attribute, memptr32 opcode is generated by the assembler.

14.1.7 Based index addressing

One-byte or two-byte data among the instruction's bytes is used as a signed displacement value. This value, the contents of the 16-bit register (BP or BW) which serves as a base register, and the contents of the 16-bit register (IX or IY) which serves as an index register are added together. The word or double word contents of the memory addressed by the result are loaded into the PC as a branch address.

This mode is used to execute the following instructions.

		Description example	
CALL	memptr16	CALL	var [BP] [IX]
CALL	memptr32	CALL	var [BW+2] [IY]
BR	memptr16	BR	var [BW] [2] [IX]
BR	memptr32	BR	var [BP+4] [IY]

Remark If variable var has a word attribute, memptr16 opcode is generated by the assembler. If it has a double word attribute, memptr32 opcode is generated by the assembler.

14.2 Memory Operand Address

A number of addressing modes for registers and memory to be handled when executing instructions can be used, as described below.

14.2.1 Register addressing

The register to be handled is addressed by the contents of the register specification field (reg = 3-bit field or sreg = 2-bit field) among the instruction's bytes.

One of eight word registers (AW, BW, CW, DW, BP, SP, IX, and IY) and eight byte registers (AL, AH, BL, BH, CL, CH, DL, and DH) is specified in a combination of reg and one bit (W) among the instruction's bytes that specifies a word or byte.

One of four segment registers (PS, SS, DS0, and DS1) is specified in sreg.

The opcode of an instruction may specify a particular register.

This mode is used to execute instructions having the following operand identifiers.

Identifier	Description
reg	AW, BW, CW, DW, SP, BP, IX, IY, AL, AH, BL, BH, CL, CH, DL, DH
reg16	AW, BW, CW, DW, SP, BP, IX, IY
reg8	AL, AH, BL, BH, CL, CH, DL, DH
sreg	PS, SS, DS0, DS1
acc	AW, AL

Description example

```
MOV    When reg, reg'
MOV    BP, SP
MOV    AL, CL
```

14.2.2 Immediate addressing

One-byte or two-byte immediate data among the instruction's bytes is to be handled as it is.

This mode is used to execute instructions having the following operand identifiers.

Identifier	Description
imm	8-bit or 16-bit immediate data
imm16	16-bit immediate data
imm8	8-bit immediate data
pop-value	16-bit immediate data

For imm, 8-bit or 16-bit data is determined by the assembler which decides the imm value described in the operand or another operand attribute described at the same time. Word/byte specification bit W is also determined.

Description example

```
MOV    When reg, imm
MOV    AL, 5; byte

MUL    When reg16, reg16', imm16
MUL    AW, BW, 1000H
```

14.2.3 Direct addressing

Immediate data among the instruction's bytes addresses the memory to be manipulated.

This mode is used to execute instructions having the following operand identifiers.

Identifier	Description
mem	16-bit variable specifying 8-bit or 16-bit memory data
dmem	16-bit variable specifying 8-bit or 16-bit memory data
imm4	4-bit variable indicating the bit field data's bit length

Description example

```
MOV    When mem, imm
MOV    WORD_VAR, 2000H

MOV    When acc, dmem
MOV    AL, BYTE_VAR
```

14.2.4 Register indirect addressing

The 16-bit register (IX, IY, or BW) specified in the memory specification field (mod or mem) among the instruction's bytes addresses the memory to be handled.

This mode is used to execute instructions having the following operand identifiers.

Identifier	Description
mem	[IX], [IY], [BW]

Description example

```
SUB    When mem, reg
SUB    [IX], AW
```

14.2.5 Automatic increment/decrement addressing

This mode belongs to register indirect addressing. After the default register contents address the memory to be manipulated, the register contents are automatically incremented or decremented (by one for byte processing or by two for word processing).

In other words, this addressing function enables automatic address updating for the next processing of a byte or word operand.

The direction flag (DIR) specifies increment or decrement addressing. If DIR = 0, increment addressing is specified; if DIR = 1, decrement addressing is specified.

This addressing mode is always applicable to default registers, and is used to execute instructions having the following operand identifiers.

Identifier	Default register
dst-block	IY
src-block	IX

The addressing mode and the counter which counts the number of byte/word operand processing repetitions (CW) are used in combination for block data processing control.

14.2.6 Index addressing

One-byte or two-byte immediate data among the instruction's bytes is the signed displacement value, which is added to the 16-bit register (IX or IY) that serves as an index register, and the result addresses the memory operand to be manipulated.

This addressing mode is useful for accessing array-type data. The displacement indicates the array start address and the index register contents determine the array element location.

This mode is used to execute instructions having the following operand identifiers.

Identifier	Description
mem	var [IX], var [IY]
mem16	var [IX], var [IY]
mem8	var [IX], var [IY]

Description example

```
TEST     When mem, imm
TEST     BYTE_VAR [IX], 7FH
TEST     BYTE_VAR [IX+8], 7FH
TEST     WORD_VAR [IX] [8], 7FFFH
```

Remark When variable var has a byte attribute, byte operand is specified and when it has a word attribute, word operand is specified. In either case, the corresponding opcode is generated by the assembler.

14.2.7 Based addressing

One-byte or two-byte immediate data among the instruction's bytes is the signed displacement value, which is added to the 16-bit register (BP or BW) that serves as a base register, and the result addresses the memory operand to be manipulated.

This addressing mode is useful for accessing structure-type data stored in a number of memory locations. The base register indicates the structure start address and the displacement selects one element in the structure.

This mode is used to execute instructions having the following operand identifiers.

Identifier	Description
mem	var [BP], var [BW]
mem16	var [BP], var [BW]
mem8	var [BP], var [BW]

Description example

```
SHL      When mem, 1
SHL      BYTE_VAR [BP], 1
SHL      WORD_VAR [BP+2], 1
SHL      BYTE_VAR [BP] [4], 1
```

Remark When variable var has a byte attribute, byte operand is specified and when it has a word attribute, word operand is specified. In either case, the corresponding opcode is generated by the assembler.

14.2.8 Based index addressing

One-byte or two-byte immediate data among the instruction's bytes is the signed displacement value. This value is added to the contents of the 16-bit register (BP or BW) which serves as a base register and the contents of the 16-bit register (IX or IY) which serves as an index register. The result addresses the memory operand to be manipulated.

This addressing can point to one data unit by changing both the base register contents and index register contents, and thus is very useful for accessing structure-type data that contains array-type data. In other words, the base register indicates the structure's starting address, the displacement value indicates the offset from the structure's starting address to the starting address of the array-type data, and the index register indicates the array data element location.

This mode is used to execute instructions having the following operand identifiers.

Identifier	Description
mem	var [base register] [index register]
mem16	var [base register] [index register]
mem8	var [base register] [index register]

Description example

```
PUSH    When mem16
PUSH    WORD_VAR [BP] [IX]
PUSH    WORD_VAR [BP+2] [IX+6]
PUSH    WORD_VAR [BP] [4] [IX] [8]
```

14.2.9 Bit addressing

Three-bit or four-bit immediate data among the instruction's bytes or the low-order three or four bits of the CL register specifies one bit of the 8-bit or 16-bit register or the memory to be manipulated.

If an instruction using this addressing mode is executed, only a particular bit of the specified register or memory can be tested (to determine 0 or 1), set, cleared, or inverted without considering other bit contents. This means that byte or word data need not be provided for one-bit manipulation such as set or reset when using the AND or OR instruction.

This mode is used to execute instructions having the following operand identifiers.

Identifier	Description
imm4	Word operand bit number
imm3	Byte operand bit number
CL	CL

Description example

```
  TEST1    reg8, CL
  TEST1    AL, CL
  NOT1     reg8, imm3
  NOT1     CL, 5
  CLR1     mem16, CL
  CLR1     WORD_VAR [IX], CL
  SET1     mem16, imm4
  SET1     WORD_VAR [BP], 9
```

14.2.10 Special function register addressing

One-byte immediate data among the instruction's bytes addresses the special function register to be manipulated as offset (unsigned) from the top of the special function register area.

This addressing mode is applicable only to the BTCLR instruction.

Identifier	Description
sfr	8-bit variable specifying 8-bit special function register

Description example

```
  BTCLR    EXIC, 7, 45
```

Remark The special function registers are mapped in memory space (between addresses xxF00H and xxFFFH where xx is the IDB register value). For details, see section **3.5.3 Special function register area.**