D16/M Instruction Set

Addressing Modes

The D16/M processor has eight addressing modes, including Implied Addressing (0 address instructions), as follows:

MNE --	**Implied Addressing**. The instruction operates on the CPU, and there is no memory operand. Instruction operation code (opcode) only.
MNE m	**Immediate Addressing**. The memory word following the instruction opcode is the operand for the instruction.
MNE (m)	**Direct Addressing**. The memory word following the opcode is the address of the operand.
MNE [m]	**Indirect Addressing**. The memory word following the opcode is the address of the address of the operand.
MNE [m++]	**Post-Incremented Indirect Addressing**. As in Indirect Addressing, the memory word following the opcode is the address of the address of the operand. After the instruction executes, the address of the operand is incremented by one.
MNE (m+IX) MNE (m+IY) MNE (m+SP)	**Direct Indexed Addressing**. The memory word following the opcode is the base address of the operand. The physical address of the operand is then the sum of the base address and the contents of Index Register IX (or IY or SP).
MNE [m+IX] MNE [m+IY] MNE [m+SP]	**Indirect Indexed Addressing**. The memory word following the opcode is the address of the base address of the operand. The physical address of the operand is then the sum of the base address and the contents of the Index Register IX (or IY or SP).
MNE [(m++)+IX] MNE [(m++)+IY] MNE [(m++)+SP]	**Post-Incremented Indirect Indexed Addressing.** As in Indirect Indexed Addressing, the memory word following the opcode is the address of the base address of the operand. The physical address of the operand is then the sum of the base address and the contents of the Index Register IX (or IY or SP). After the instruction executes, the base address of the operand is incremented by one.

In the Indexed addressing modes, the sum is a simple addition of the base address and the contents of the requisite Index Register. Any carry resulting from this addition is ignored; that

is, the address will "wrap around," and two's complement numbers in the base address or in the Index Register will produce the correct result.

The assembler permits Indexed address operands to appear in any order; for example, the instruction

```
ADD   (BASE+IX)
```

may also be written as

```
ADD   (IX+BASE)
```

with the same result.

The number of machine cycles required for the execution of each instruction is shown in the Cycles column of the instruction table. A single machine cycle is 250 ns, except for Extended cycles (memory and I/O accesses) which are 500 ns. So, for example, the SET instruction will require five standard cycles plus one extended cycle; a total of six cycles, and the execution time will be 1750 ns (1.75 µs).

D16/M Zero-Address Instructions (44)

Mnemonic	Opcode	Function	Cycles (Extended)
INC	001A	Increment Accumulator	7 (1)
DEC	001B	Decrement Accumulator	7 (1)
COM	001C	One's Complement Accumulator	6 (1)
NEG	001D	Two's Complement Accumulator	6 (1)
SET	001E	Set Accumulator (AC < FFFF)	6 (1)
CLR	001F	Clear Accumulator (AC < 0000)	6 (1)
SHL	0020	Shift Left Accumulator (CF < AC15, AC0 < 0)	6 (1)
LSR	0021	Logical Shift Right Accumulator (AC15 < 0, CF < AC0)	6 (1)
ASR	0022	Arithmetic Shift Right Accumulator (AC15 < AC15, CF < AC0)	6 (1)
ROL	0023	Rotate Left Accumulator through Carry (CF < AC15, AC0 < CF)	6 (1)

ROR	0024	Rotate Right Accumulator through Carry (AC15 < CF, CF < AC0)	6 (1)
LSW	0025	Load Accumulator with Switch Register	6 (1)
SCF	0026	Set Carry Flag (CF < 1)	6 (1)
CCF	0027	Clear Carry Flag (CF < 0)	6 (1)
PSA	0028	Push Accumulator onto Stack	7 (2)
POA	0029	Pop Accumulator off of Stack	6 (2)
PSF	002A	Push Flag Register onto Stack	7 (2)
POF	002B	Pop Flag Register off of Stack	6 (2)
ENI	002C	Enable Interrupts	5 (1)
DII	002D	Disable Interrupts	5 (1)
RTN	002E	Return from Subroutine	6 (2)
RTI	002F	Return from Interrupt, Enabling Interrupts	10 (4)
NOP	0030	No Operation	6 (1)
INX	0035	Increment Index Register IX	6 (1)
INY	0036	Increment Index Register IY	6 (1)
INS	0037	Increment Stack Pointer	6 (1)
DEX	0038	Decrement Index Register IX	6 (1)
DEY	0039	Decrement Index Register IY	6 (1)
DES	003A	Decrement Stack Pointer	6 (1)
MAX	003B	Move Accumulator to Index Register IX	6 (1)
MAY	003C	Move Accumulator to Index Register IY	6 (1)
MAS	003D	Move Accumulator to Stack Pointer	6 (1)
MXA	003E	Move Index Register IX to Accumulator	6 (1)
MYA	003F	Move Index Register IY to Accumulator	6 (1)

MSA	0040	Move Stack Pointer to Accumulator	6 (1)
PSX	0041	Push Index Register IX onto Stack	7 (2)
POX	0042	Pop Index Register IX off of Stack	6 (2)
PSY	0043	Push Index Register IY onto Stack	7 (2)
POY	0044	Pop Index Register IY off of Stack	6 (2)
BSW	0045	Byte Swap Accumulator (AC15-8 < AC7-0, AC7-0 < AC15-8)	20 (1)
BSL	0046	Byte Shift Left Accumulator (AC15-8 < AC7-0, AC7-0 < 00)	12 (1)
BSR	0047	Byte Shift Right Accumulator (AC7-0 < AC15-8, AC15-8 < 00)	12 (1)
BRK	0048	Software Interrupt to Address FFFE	13 (4)
HLT	00FF	Processor Halt	5 (1)

D16/M Memory Reference (One Address) Instructions (27)

Mnemonic	Opcode	Function	Cycles (Extended)
LDA m	0301	Load Accumulator (Immediate)	8 (2)
LDA (m)	0101	Load Accumulator (Direct)	8 (3)
LDA [m]	0501	Load Accumulator (Indirect)	9 (4)
LDA [m++]	0D01	Load Accumulator (Indirect with Increment)	12 (5)
LDA (m+IX)	1101	Load Accumulator (Direct Indexed)	8 (3)
LDA (m+IY)	2101		8 (3)
LDA (m+SP)	3101		8 (3)
LDA [m+IX]	1501	Load Accumulator (Indirect Indexed)	9 (4)
LDA [m+IY]	2501		9 (4)
LDA [m+SP]	3501		9 (4)
LDA [(m++)+IX]	1D01	Load Accumulator (Indirect Indexed with Increment)	12 (5)
LDA [(m++)+IY]	2D01		12 (5)
LDA [(m++)+SP]	3D01		12 (5)

LDS m	0302	Load Stack Pointer	8 (2)
LDS (m)	0102		8 (3)
LDS [m]	0502		9 (4)
LDS [m++]	0D02		12 (5)
LDS (m+IX)	1102		8 (3)
LDS (m+IY)	2102		8 (3)
LDS (m+SP)	3102		8 (3)
LDS [m+IX]	1502		9 (4)
LDS [m+IY]	2502		9 (4)
LDS [m+SP]	3502		9 (4)
LDS [(m++)+IX]	1D02		12 (5)
LDS [(m++)+IY]	2D02		12 (5)
LDS [(m++)+SP]	3D02		12 (5)
STA (m)	0103	Store Accumulator	7 (3)
STA [m]	0503		8 (4)
STA [m++]	0D03		11 (5)
STA (m+IX)	1103		7 (3)
STA (m+IY)	2103		7 (3)
STA (m+SP)	3103		7 (3)
STA [m+IX]	1503		8 (4)
STA [m+IY]	2503		8 (4)
STA [m+SP]	3503		8 (4)
STA [(m++)+IX]	1D03		11 (5)
STA [(m++)+IY]	2D03		11 (5)
STA [(m++)+SP]	3D03		11 (5)
STS (m)	0104	Store Stack Pointer	7 (3)
STS [m]	0504		8 (4)
STS [m++]	0D04		11 (5)
STS (m+IX)	1104		7 (3)
STS (m+IY)	2104		7 (3)
STS (m+SP)	3104		7 (3)
STS [m+IX]	1504		8 (4)
STS [m+IY]	2504		8 (4)
STS [m+SP]	3504		8 (4)
STS [(m++)+IX]	1D04		11 (5)
STS [(m++)+IY]	2D04		11 (5)
STS [(m++)+SP]	3D04		11 (5)

ADD m	0305	Add to Accumulator	7 (2)
ADD (m)	0105		8 (3)
ADD [m]	0505		9 (4)
ADD [m++]	0D05		12 (5)
ADD (m+IX)	1105		8 (3)
ADD (m+IY)	2105		8 (3)
ADD (m+SP)	3105		8 (3)
ADD [m+IX]	1505		9 (4)
ADD [m+IY]	2505		9 (4)
ADD [m+SP]	3505		9 (4)
ADD [(m++)+IX]	1D05		12 (5)
ADD [(m++)+IY]	2D05		12 (5)
ADD [(m++)+SP]	3D05		12 (5)
ADC m	0306	Add to Accumulator with Carry	7 (2)
ADC (m)	0106		8 (3)
ADC [m]	0506		9 (4)
ADC [m++]	0D06		12 (5)
ADC (m+IX)	1106		8 (3)
ADC (m+IY)	2106		8 (3)
ADC (m+SP)	3106		8 (3)
ADC [m+IX]	1506		9 (4)
ADC [m+IY]	2506		9 (4)
ADC [m+SP]	3506		9 (4)
ADC [(m++)+IX]	1D06		12 (5)
ADC [(m++)+IY]	2D06		12 (5)
ADC [(m++)+SP]	3D06		12 (5)
SUB m	0307	Subtract from Accumulator	7 (2)
SUB (m)	0107		8 (3)
SUB [m]	0507		9 (4)
SUB [m++]	0D07		12 (5)
SUB (m+IX)	1107		8 (3)
SUB (m+IY)	2107		8 (3)
SUB (m+SP)	3107		8 (3)
SUB [m+IX]	1507		9 (4)
SUB [m+IY]	2507		9 (4)
SUB [m+SP]	3507		9 (4)
SUB [(m++)+IX]	1D07		12 (5)
SUB [(m++)+IY]	2D07		12 (5)
SUB [(m++)+SP]	3D07		12 (5)

SBC m	0308	Subtract from Accumulator with Borrow	7 (2)
SBC (m)	0108		8 (3)
SBC [m]	0508		9 (4)
SBC [m++]	0D08		12 (5)
SBC (m+IX)	1108		8 (3)
SBC (m+IY)	2108		8 (3)
SBC (m+SP)	3108		8 (3)
SBC [m+IX]	1508		9 (4)
SBC [m+IY]	2508		9 (4)
SBC [m+SP]	3508		9 (4)
SBC [(m++)+IX]	1D08		12 (5)
SBC [(m++)+IY]	2D08		12 (5)
SBC [(m++)+SP]	3D08		12 (5)
AND m	0309	Logical AND with Accumulator	7 (2)
AND (m)	0109		8 (3)
AND [m]	0509		9 (4)
AND [m++]	0D09		12 (5)
AND (m+IX)	1109		8 (3)
AND (m+IY)	2109		8 (3)
AND (m+SP)	3109		8 (3)
AND [m+IX]	1509		9 (4)
AND [m+IY]	2509		9 (4)
AND [m+SP]	3509		9 (4)
AND [(m++)+IX]	1D09		12 (5)
AND [(m++)+IY]	2D09		12 (5)
AND [(m++)+SP]	3D09		12 (5)
ORA m	030A	Logical OR with Accumulator	7 (2)
ORA (m)	010A		8 (3)
ORA [m]	050A		9 (4)
ORA [m++]	0D0A		12 (5)
ORA (m+IX)	110A		8 (3)
ORA (m+IY)	210A		8 (3)
ORA (m+SP)	310A		8 (3)
ORA [m+IX]	150A		9 (4)
ORA [m+IY]	250A		9 (4)
ORA [m+SP]	350A		9 (4)
ORA [(m++)+IX]	1D0A		12 (5)
ORA [(m++)+IY]	2D0A		12 (5)
ORA [(m++)+SP]	3D0A		12 (5)

XOR m	030B	Logical Exclusive-OR with Accumulator	7 (2)
XOR (m)	010B		8 (3)
XOR [m]	050B		9 (4)
XOR [m++]	0D0B		12 (5)
XOR (m+IX)	110B		8 (3)
XOR (m+IY)	210B		8 (3)
XOR (m+SP)	310B		8 (3)
XOR [m+IX]	150B		9 (4)
XOR [m+IY]	250B		9 (4)
XOR [m+SP]	350B		9 (4)
XOR [(m++)+IX]	1D0B		12 (5)
XOR [(m++)+IY]	2D0B		12 (5)
XOR [(m++)+SP]	3D0B		12 (5)
ISZ (m)	010C	Increment and Skip if Zero	9 (4)
ISZ [m]	050C		10 (5)
ISZ (m+IX)	110C		9 (4)
ISZ (m+IY)	210C		9 (4)
ISZ (m+SP)	310C		9 (4)
ISZ [m+IX]	150C		10 (5)
ISZ [m+IY]	250C		10 (5)
ISZ [m+SP]	350C		10 (5)
ISZ [(m++)+IX]	1D0C		13 (5)
ISZ [(m++)+IY]	2D0C		13 (5)
ISZ [(m++)+SP]	3D0C		13 (5)
DSZ (m)	010D	Decrement and Skip if Zero	9 (4)
DSZ [m]	050D		10 (5)
DSZ (m+IX)	110D		9 (4)
DSZ (m+IY)	210D		9 (4)
DSZ (m+SP)	310D		9 (4)
DSZ [m+IX]	150D		10 (5)
DSZ [m+IY]	250D		10 (5)
DSZ [m+SP]	350D		10 (5)
DSZ [(m++)+IX]	1D0D		13 (5)
DSZ [(m++)+IY]	2D0D		13 (5)
DSZ [(m++)+SP]	3D0D		13 (5)
		Any Skipped Instruction after ISZ or DSZ:	4 (1)

JMP m	030E	Unconditional Jump	7 (2)
JMP (m)	010E		8 (3)
JMP [m]	050E		9 (4)
JMP [m++]	0D0E		12 (5)
JMP (m+IX)	110E		8 (3)
JMP (m+IY)	210E		8 (3)
JMP (m+SP)	310E		8 (3)
JMP [m+IX]	150E		9 (4)
JMP [m+IY]	250E		9 (4)
JMP [m+SP]	350E		9 (4)
JMP [(m++)+IX]	1D0E		12 (5)
JMP [(m++)+IY]	2D0E		12 (5)
JMP [(m++)+SP]	3D0E		12 (5)
JOZ m	030F	Jump on Zero Accumulator	9 (2)
JOZ (m)	010F		10 (3)
JOZ [m]	050F		11 (4)
JOZ [m++]	0D0F		14 (5)
JOZ (m+IX)	110F		10 (3)
JOZ (m+IY)	210F		10 (3)
JOZ (m+SP)	310F		10 (3)
JOZ [m+IX]	150F		11 (4)
JOZ [m+IY]	250F		11 (4)
JOZ [m+SP]	350F		11 (4)
JOZ [(m++)+IX]	1D0F		14 (5)
JOZ [(m++)+IY]	2D0F		14 (5)
JOZ [(m++)+SP]	3D0F		14 (5)
JNZ m	0310	Jump on Non-Zero Accumulator	9 (2)
JNZ (m)	0110		10 (3)
JNZ [m]	0510		11 (4)
JNZ [m++]	0D10		14 (5)
JNZ (m+IX)	1110		10 (3)
JNZ (m+IY)	2110		10 (3)
JNZ (m+SP)	3110		10 (3)
JNZ [m+IX]	1510		11 (4)
JNZ [m+IY]	2510		11 (4)
JNZ [m+SP]	3510		11 (4)
JNZ [(m++)+IX]	1D10		14 (5)
JNZ [(m++)+IY]	2D10		14 (5)
JNZ [(m++)+SP]	3D10		14 (5)

JPL m	0311	Jump on Plus Accumulator	9 (2)
JPL (m)	0111		10 (3)
JPL [m]	0511		11 (4)
JPL [m++]	0D11		14 (5)
JPL (m+IX)	1111		10 (3)
JPL (m+IY)	2111		10 (3)
JPL (m+SP)	3111		10 (3)
JPL [m+IX]	1511		11 (4)
JPL [m+IY]	2511		11 (4)
JPL [m+SP]	3511		11 (4)
JPL [(m++)+IX]	1D11		14 (5)
JPL [(m++)+IY]	2D11		14 (5)
JPL [(m++)+SP]	3D11		14 (5)
JMI m	0312	Jump on Minus Accumulator	9 (2)
JMI (m)	0112		10 (3)
JMI [m]	0512		11 (4)
JMI [m++]	0D12		14 (5)
JMI (m+IX)	1112		10 (3)
JMI (m+IY)	2112		10 (3)
JMI (m+SP)	3112		10 (3)
JMI [m+IX]	1512		11 (4)
JMI [m+IY]	2512		11 (4)
JMI [m+SP]	3512		11 (4)
JMI [(m++)+IX]	1D12		14 (5)
JMI [(m++)+IY]	2D12		14 (5)
JMI [(m++)+SP]	3D12		14 (5)
JOC m	0313	Jump on Carry (CF = 1)	9 (2)
JOC (m)	0113		10 (3)
JOC [m]	0513		11 (4)
JOC [m++]	0D13		14 (5)
JOC (m+IX)	1113		10 (3)
JOC (m+IY)	2113		10 (3)
JOC (m+SP)	3113		10 (3)
JOC [m+IX]	1513		11 (4)
JOC [m+IY]	2513		11 (4)
JOC [m+SP]	3513		11 (4)
JOC [(m++)+IX]	1D13		14 (5)
JOC [(m++)+IY]	2D13		14 (5)
JOC [(m++)+SP]	3D13		14 (5)

JNC m	0314	Jump on No Carry (CF = 0)	9 (2)
JNC (m)	0114		10 (3)
JNC [m]	0514		11 (4)
JNC [m++]	0D14		14 (5)
JNC (m+IX)	1114		10 (3)
JNC (m+IY)	2114		10 (3)
JNC (m+SP)	3114		10 (3)
JNC [m+IX]	1514		11 (4)
JNC [m+IY]	2514		11 (4)
JNC [m+SP]	3514		11 (4)
JNC [(m++)+IX]	1D14		14 (5)
JNC [(m++)+IY]	2D14		14 (5)
JNC [(m++)+SP]	3D14		14 (5)
JOV m	0315	Jump on Twos Complement Overflow (OVF = 1)	9 (2)
JOV (m)	0115		10 (3)
JOV [m]	0515		11 (4)
JOV [m++]	0D15		14 (5)
JOV (m+IX)	1115		10 (3)
JOV (m+IY)	2115		10 (3)
JOV (m+SP)	3115		10 (3)
JOV [m+IX]	1515		11 (4)
JOV [m+IY]	2515		11 (4)
JOV [m+SP]	3515		11 (4)
JOV [(m++)+IX]	1D15		14 (5)
JOV [(m++)+IY]	2D15		14 (5)
JOV [(m++)+SP]	3D15		14 (5)
JNV m	0316	Jump on No Overflow (OVF = 0)	9 (2)
JNV (m)	0116		10 (3)
JNV [m]	0516		11 (4)
JNV [m++]	0D16		14 (5)
JNV (m+IX)	1116		10 (3)
JNV (m+IY)	2116		10 (3)
JNV (m+SP)	3116		10 (3)
JNV [m+IX]	1516		11 (4)
JNV [m+IY]	2516		11 (4)
JNV [m+SP]	3516		11 (4)
JNV [(m++)+IX]	1D16		14 (5)
JNV [(m++)+IY]	2D16		14 (5)
JNV [(m++)+SP]	3D16		14 (5)
CSR m	0317	Call Subroutine	9 (3)
CSR (m)	0117		10 (4)
CSR [m]	0517		11 (5)
CSR [m++]	0D17		14 (6)

LDX m	0331	Load Index Register IX	8 (2)
LDX (m)	0131		8 (3)
LDX [m]	0531		9 (4)
LDX [m++]	0D31		12 (5)
LDX (m+IX)	1131		8 (3)
LDX (m+IY)	2131		8 (3)
LDX (m+SP)	3131		8 (3)
LDX [m+IX]	1531		9 (4)
LDX [m+IY]	2531		9 (4)
LDX [m+SP]	3531		9 (4)
LDX [(m++)+IX]	1D31		12 (5)
LDX [(m++)+IY]	2D31		12 (5)
LDX [(m++)+SP]	3D31		12 (5)
LDY m	0332	Load Index Register IY	8 (2)
LDY (m)	0132		8 (3)
LDY [m]	0532		9 (4)
LDY [m++]	0D32		12 (5)
LDY (m+IX)	1132		8 (3)
LDY (m+IY)	2132		8 (3)
LDY (m+SP)	3132		8 (3)
LDY [m+IX]	1532		9 (4)
LDY [m+IY]	2532		9 (4)
LDY [m+SP]	3532		9 (4)
LDY [(m++)+IX]	1D32		12 (5)
LDY [(m++)+IY]	2D32		12 (5)
LDY [(m++)+SP]	3D32		12 (5)
STX (m)	0133	Store Index Register IX	7 (3)
STX [m]	0533		8 (4)
STX [m++]	0D33		11 (5)
STX (m+IX)	1133		7 (3)
STX (m+IY)	2133		7 (3)
STX (m+SP)	3133		7 (3)
STX [m+IX]	1533		8 (4)
STX [m+IY]	2533		8 (4)
STX [m+SP]	3533		8 (4)
STX [(m++)+IX]	1D33		11 (5)
STX [(m++)+IY]	2D33		11 (5)
STX [(m++)+SP]	3D33		11 (5)

STY (m)	0134	Store Index Register IY	7 (3)
STY [m]	0534		8 (4)
STY [m++]	0D34		11 (5)
STY (m+IX)	1134		7 (3)
STY (m+IY)	2134		7 (3)
STY (m+SP)	3134		7 (3)
STY [m+IX]	1534		8 (4)
STY [m+IY]	2534		8 (4)
STY [m+SP]	3534		8 (4)
STY [(m++)+IX]	1D34		11 (5)
STY [(m++)+IY]	2D34		11 (5)
STY [(m++)+SP]	3D34		11 (5)

D16/M Input/Output Instructions (2)

Mnemonic	**Opcode**	**Function**	**Cycles (Extended)**
INP (m)	0118	Input to Accumulator from I/O port	7 (3)
INP [m]	0518		8 (4)
INP [m++]	0D18		11 (5)
INP (m+IX)	1118		7 (3)
INP (m+IY)	2118		7 (3)
INP (m+SP)	3118		7 (3)
INP [m+IX]	1518		8 (4)
INP [m+IY]	2518		8 (4)
INP [m+SP]	3518		8 (4)
INP [(m++)+IX]	1D18		11 (5)
INP [(m++)+IY]	2D18		11 (5)
INP [(m++)+SP]	3D18		11 (5)
OUT (m)	0119	Output from Accumulator to I/O port	7 (3)
OUT [m]	0519		8 (4)
OUT [m++]	0D19		11 (5)
OUT (m+IX)	1119		7 (3)
OUT (m+IY)	2119		7 (3)
OUT (m+SP)	3119		7 (3)
OUT [m+IX]	1519		8 (4)
OUT [m+IY]	2519		8 (4)
OUT [m+SP]	3519		8 (4)
OUT [(m++)+IX]	1D19		11 (5)
OUT [(m++)+IY]	2D19		11 (5)
OUT [(m++)+SP]	3D19		11 (5)