LDW Load word LDW

Syntax LDW dst,src e.g. LDW X,#$1500

Operation dst <= src

Description Load the destination word (16-bit value) with the source word. The dst and src can be a 16-bit register (X, Y or SP) or a memory/data 16-bit value.

Instruction overview

mnem	dst	src	Affected condition flags						
			V	I1	H	I0	N	Z	C
LD	Reg	Mem	-	-	-	-	N	Z	-
LD	Mem	Reg	-	-	-	-	N	Z	-
LD	Reg	Reg	-	-	-	-	-	-	-
LD	SP	Reg	-	-	-	-	-	-	-
LD	Reg	SP	-	-	-	-	-	-	-

N ⇒ R15
Set if bit 7 of the result is set (negative value), cleared otherwise.

Z ⇒ $\overline{R15}.\overline{R14}.\overline{R13}.\overline{R12}.\overline{R11}.\overline{R10}.\overline{R9}.\overline{R8}.\overline{R7}.\overline{R6}.\overline{R5}.\overline{R4}.\overline{R3}.\overline{R2}.\overline{R1}.\overline{R0}$
Set if the result is zero (0x0000), cleared otherwise.

Detailed description

dst	src	Asm	cy	lgth	Op-code(s)					ST7
X	#word	LDW X,#$55AA	2	3		AE	MS	LS		✗
X	shortmem	LDW X,$50	2	2		BE	XX			✗
X	longmem	LDW X,$5000	2	3		CE	MS	LS		✗
X	(X)	LDW X,(X)	2	1		FE				✗
X	(shortoff,X)	LDW X,($50,X)	2	2		EE	XX			✗
X	(longoff,X)	LDW X,($5000,X)	2	3		DE	MS	LS		✗
X	(shortoff,SP)	LDW X,($50,SP)	2	2		1E	XX			
X	[shortptr.w]	LDW X,[$50.w]	5	3	92	CE	XX			✗
X	[longptr.w]	LDW X,[$5000.w]	5	4	72	CE	MS	LS		
X	([shortptr.w],X)	LDW X,([$50.w],X)	5	3	92	DE	XX			✗
X	([longptr.w],X)	LDW X,([$5000.w],X)	5	4	72	DE	MS	LS		

dst	src	Asm	cy	lgth	Op-code(s)					ST7
shortmem	X	LDW $50,X	2	2		BF	XX			✗
longmem	X	LDW $5000,X	2	3		CF	MS	LS		✗
(X)	Y	LDW (X),Y	2	1		FF				
(shortoff,X)	Y	LDW ($50,X),Y	2	2		EF	XX			
(longoff,X)	Y	LDW ($5000,X),Y	2	3		DF	MS	LS		

JRA Jump Relative Always JRA

Syntax JRA dst e.g. JRA loop

Operation PC = PC+lgth

PC <= PC + dst, if Condition is True

Description Unconditional relative jump. PC is updated by the signed addition of PC and dst. Control then passes to the statement addressed by the program counter. Else, the program continues normally.

Instruction overview

mnem	dst	Affected condition flags						
		V	I1	H	I0	N	Z	C
JRA	Mem	-	-	-	-	-	-	-

Detailed description

dst	Asm	cy	lgth	Op-code(s)					ST7
shortoff	JRA $2B	2	2		20	XX			✗

See also: JP

JRxx Conditional Jump Relative Instruction JRxx

Syntax JRxx dst e.g. JRxx loop

Operation PC = PC+lgth

PC <= PC + dst, if Condition is True

Description Conditional relative jump. PC is updated by the signed addition of PC and dst, if the condition is true. Control, then passes to the statement addressed by the program counter. Else, the program continues normally.

Instruction overview

mnem	dst	Affected condition flags						
		V	I1	H	I0	N	Z	C
JRxx	Mem	-	-	-	-	-	-	-

Instruction List

mnem	meaning	sym	Condition	Op-code (OC)	
JRC	Carry		C = 1		25
JREQ	Equal	=	Z = 1		27
JRF	False		False		21
JRH	Half-Carry		H = 1	90	29
JRIH	Interrupt Line is High			90	2F
JRIL	Interrupt Line is Low			90	2E
JRM	Interrupt Mask		I = 1	90	2D
JRMI	Minus	< 0	N = 1		2B
JRNC	Not Carry		C = 0		24
JRNE	Not Equal	<> 0	Z = 0		26
JRNH	Not Half-Carry		H = 0	90	28
JRNM	Not Interrupt Mask		I = 0	90	2C
JRNV	Not Overflow		V = 0		28
JRPL	Plus	>= 0	N = 0		2A
JRSGE	Signed Greater or Equal	>=	(N XOR V) = 0		2E
JRSGT	Signed Greater Than	>	(Z OR (N XOR V)) = 0		2C
JRSLE	Signed Lower or Equal	<=	(Z OR (N XOR V)) = 1		2D
JRSLT	Signed Lower Than	<	(N XOR V) = 1		2F
JRT	True		True		20
JRUGE	Unsigned Greater or Equal		C = 0		24
JRUGT	Unsigned Greater Than	>	C = 0 and Z = 0		22
JRULE	Unsigned Lower or Equal	<=	C = 1 or Z = 1		23
JRC	Carry		C = 1		25
JRULT	Unsigned Lower Than		C = 1		25
JRV	Overflow		V = 1		29

Detailed description

dst	Asm	cy	lgth	Op-code(s)					ST7
shortoff	JRxx $15	1/2	2		Op-code	XX			✗
shortoff	JRxx $15	1/2	3	90	Op-code	XX			✗

ADD Addition ADD

Syntax ADD A,src e.g. ADD A,#%11001010

Operation A <= A+ src

Description The source byte is added to the contents of the accumulator and the result is stored in the accumulator. The source is a memory or data byte.

Instruction overview

mnem	dst	src	Affected condition flags						
			V	I1	H	I0	N	Z	C
ADD	A	Mem	V	-	H	-	N	Z	C

V ⇒ $(A7.M7 + M7.\overline{R7} + \overline{R7}.A7) \oplus (A6.M6 + M6.\overline{R6} + \overline{R6}.A6)$
Set if the signed operation generates an overflow, cleared otherwise.

H ⇒ $A3.M3 + M3.\overline{R3} + \overline{R3}.A3$
Set if a carry occurred from bit 3 of the result, cleared otherwise.

N ⇒ R7
Set if bit 7 of the result is set (negative value), cleared otherwise.

Z ⇒ $\overline{R7}.\overline{R6}.\overline{R5}.\overline{R4}.\overline{R3}.\overline{R2}.\overline{R1}.\overline{R0}$
Set if the result is zero (0x00), cleared otherwise.

C ⇒ $A7.M7 + M7.\overline{R7} + \overline{R7}.A7$
Set if a carry occurred from bit 7 of the result, cleared otherwise.

Detailed description

dst	src	Asm	cy	lgth	Op-code(s)					ST7
A	#byte	ADD A,#$55	1	2		AB	XX			✗
A	shortmem	ADD A,$10	1	2		BB	XX			✗
A	longmem	ADD A,$1000	1	3		CB	MS	LS		✗
A	(X)	ADD A,(X)	1	1		FB				✗
A	(shortoff,X)	ADD A,($10,X)	1	2		EB	XX			✗
A	(longoff,X)	ADD A,($1000,X)	1	3		DB	MS	LS		✗
A	(Y)	ADD A,(Y)	1	2	90	FB				✗
A	(shortoff,Y)	ADD A,($10,Y)	1	3	90	EB	XX			✗
A	(longoff,Y)	ADD A,($1000,Y)	1	4	90	DB	MS	LS		✗
A	(shortoff,SP)	ADD A,($10,SP)	1	2		1B	XX			
A	[shortptr.w]	ADD A,[$10.w]	4	3	92	CB	XX			✗
A	[longptr.w]	ADD A,[$1000.w]	4	4	72	CB	MS	LS		
A	([shortptr.w],X)	ADD A,([$10.w],X)	4	3	92	DB	XX			✗
A	([longptr.w],X)	ADD A,([$1000.w],X)	4	4	72	DB	MS	LS		
A	([shortptr.w],Y)	ADD A,([$10.w],Y)	4	3	91	DB	XX			✗

See also: ADDW, ADC, SUB, SBC, MUL, DIV

LD Load LD

Syntax LD dst,src e.g. LD A,#$15

Operation dst <= src

Description Load the destination byte with the source byte. The dst and src can be a register, a byte (low/high) of an index register or a memory/data byte. When half of an index register is loaded, the other half remains unchanged.

Instruction overview

mnem	dst	src	Affected condition flags						
			V	I1	H	I0	N	Z	C
LD	Reg	Mem	-	-	-	-	N	Z	-
LD	Mem	Reg	-	-	-	-	N	Z	-
LD	Reg	Reg	-	-	-	-	-	-	-

N ⇒ R7
Set if bit 7 of the result is set (negative value), cleared otherwise.

Z ⇒ $\overline{R7}.\overline{R6}.\overline{R5}.\overline{R4}.\overline{R3}.\overline{R2}.\overline{R1}.\overline{R0}$
Set if the result is zero (0x00), cleared otherwise.

Detailed description

dst	src	Asm	cy	lgth	Op-code(s)					ST7
A	#byte	LD A,#$55	1	2		A6	XX			✗
A	shortmem	LD A,$50	1	2		B6	XX			✗
A	longmem	LD A,$5000	1	3		C6	MS	LS		✗
A	(X)	LD A,(X)	1	1		F6				✗
A	(shortoff,X)	LD A,($50,X)	1	2		E6	XX			✗
A	(longoff,X)	LD A,($5000,X)	1	3		D6	MS	LS		✗
A	(Y)	LD A,(Y)	1	2	90	F6				✗
A	(shortoff,Y)	LD A,($50,Y)	1	3	90	E6	XX			✗
A	(longoff,Y)	LD A,($5000,Y)	1	4	90	D6	MS	LS		✗
A	(shortoff,SP)	LD A,($50,SP)	1	2		7B	XX			
A	[shortptr.w]	LD A,[$50.w]	4	3	92	C6	XX			✗
A	[longptr.w]	LD A,[$5000.w]	4	4	72	C6	MS	LS		
A	([shortptr.w],X)	LD A,([$50.w],X)	4	3	92	D6	XX			✗
A	([longptr.w],X)	LD A,([$5000.w],X)	4	4	72	D6	MS	LS		
A	([shortptr.w],Y)	LD A,([$50.w],Y)	4	3	91	D6	XX			✗

CLR Clear CLR

Syntax CLR dst e.g. CLR A

Operation dst <= 00

Description The destination byte is forced to 00 value. The destination is either a memory byte location or the accumulator. This instruction is compact, and does not affect any register when used with RAM variables.

Instruction overview

mnem	dst	Affected condition flags						
		V	I1	H	I0	N	Z	C
CLR	Mem	-	-	-	-	0	1	-
CLR	A					0	1	

N: 0

Cleared

Z: 1

Set

Detailed description

dst	Asm	cy	lgth	Op-code(s)					ST7
A	CLR A	1	1		4F				✗
shortmem	CLR $10	1	2		3F	XX			✗
longmem	CLR $1000	1	4	72	5F	MS	LS		
(X)	CLR (X)	1	1		7F				✗
(shortoff.X)	CLR ($10,X)	1	2		6F	XX			✗
(longoff,X)	CLR ($1000,X)	1	4	72	4F	MS	LS		
(Y)	CLR (Y)	1	2	90	7F				✗
(shortoff,Y)	CLR ($10,Y)	1	3	90	6F	XX			✗
(longoff,Y)	CLR ($1000,Y)	1	4	90	4F	MS	LS		
(shortoff,SP)	CLR ($10,SP)	1	2		0F	XX			
[shortptr.w]	CLR [$10]	4	3	92	3F	XX			✗
[longptr.w]	CLR [$1000].w	4	4	72	3F	MS	LS		
([shortptr.w],X)	CLR ([$10],X)	4	3	92	6F	XX			✗
([longptr.w].X]	CLR ([$1000.w],X)	4	4	72	6F	MS	LS		
([shortptr.w],Y)	CLR ([$10],Y)	4	3	91	6F	XX			✗

See also: LD

JPF Jump far JPF

Syntax JPF dst e.g.:JPF test

Operation PC <= dst

Description The unconditional jump simply replaces the content of the PC by a destination with an extended address. Control then passes to the statement addressed by the program counter. For safe memory usage, this instruction must be used, when the operation crosses a memory section.

Instruction overview

mnem	dst	Affected condition flags						
		V	I1	H	I0	N	Z	C
JPF	Mem	-	-	-	-	-	-	-

Detailed description

dst	Asm
extmem	JPF $2FFFFC
[longptr.e]	JPF [$2FFC.e]

cy	lgth
2	4
6	4

Op-code(s)					ST7
	AC	ExtB	MS	LS	
92	AC	MS	LS		

See also: JP, CALLF

SUB Subtraction SUB

Syntax SUB A,src e.g. SUB A,#%11001010

Operation A <= A- src

Description The source byte is subtracted from the contents of the accumulator/SP and the result is stored in the accumulator/SP. The source is a memory or data byte.

Instruction overview

mnem	dst	src	Affected condition flags						
			V	I1	H	I0	N	Z	C
SUB	A	Mem	V	-	-	-	N	Z	C
SUB	SP	Imm	-	-	-	-	-	-	-

V ⇒ (A7.M7 + A7.R7 + A7.M7.R7) ⊕ (A6.M6 + A6.R6 + A6.M6.R6)
Set if the signed operation generates an overflow, cleared otherwise.

N ⇒ R7
Set if bit 7 of the result is set (negative value), cleared otherwise.

Z ⇒ R7.R6.R5.R4.R3.R2.R1.R0
Set if the result is zero (0x00), cleared otherwise.

C ⇒ $\overline{A7}.M7 + \overline{A7}.R7 + A7.M7.R7$
Set if a borrow request occurred from bit 7, cleared otherwise.

Detailed description

dst	src	Asm	cy	lgth	Op-code(s)				ST7
A	#byte	SUB A,#$55	1	2		A0	XX		✗
A	shortmem	SUB A,$10	1	2		B0	XX		✗
A	longmem	SUB A,$1000	1	3		C0	MS	LS	✗
A	(X)	SUB A,(X)	1	1		F0			✗
A	(shortoff,X)	SUB A,($10,X)	1	2		E0	XX		✗
A	(longoff,X)	SUB A,($1000,X)	1	3		D0	MS	LS	✗
A	(Y)	SUB A,(Y)	1	2	90	F0			✗
A	(shortoff,Y)	SUB A,($10,Y)	1	3	90	E0	XX		✗
A	(longoff,Y)	SUB A,($1000,Y)	1	4	90	D0	MS	LS	✗
A	(shortoff,SP)	SUB A,($10,SP)	1	2		10	XX		
A	[shortptr.w]	SUB A,[$10.w]	4	3	92	C0	XX		✗
A	[longptr.w]	SUB A,[$1000.w]	4	4	72	C0	MS	LS	
A	([shortptr.w],X)	SUB A,([$10.w],X)	4	3	92	D0	XX		✗
A	([longptr.w],X)	SUB A,([$1000.w],X)	4	4	72	D0	MS	LS	
A	([shortptr.w],Y)	SUB A,([$10.w],Y)	4	3	91	D0	XX		✗
SP	#byte	SUB SP,#$9	1	2		52	XX		

See also: SUBW, ADD, ADC, SBC, MUL