

German Mnemonics	English Mnemonics	Program Elements Catalog	Description
=	=	Bit logic Instruction	Assign
)	)	Bit logic Instruction	Nesting Closed
*D	*D	Integer math Instruction	Multiply ACCU 1 and ACCU 2 as Double Integer (32-Bit)
*I	*I	Integer math Instruction	Multiply ACCU 1 and ACCU 2 as Integer (16-Bit)
*R	*R	Floating point Instruction	Multiply ACCU 1 and ACCU 2 as Floating-Point Numbers (32-Bit IEEE-FP)
/D	/D	Integer math Instruction	Divide ACCU 2 by ACCU 1 as Double Integer (32-Bit)
/I	/I	Integer math Instruction	Divide ACCU 2 by ACCU 1 as Integer (16-Bit)
/R	/R	Floating point Instruction	Divide ACCU 2 by ACCU 1 as a Floating-Point Number (32-Bit IEEE-FP)
?D	?D	Compare	Compare Double Integer (32-Bit) ==, <>, >, <, >=, <=
?I	?I	Compare	Compare Integer (16-Bit) ==, <>, >, <, >=, <=
?R	?R	Compare	Compare Floating-Point Number (32-Bit) ==, <>, >, <, >=, <=
+	+	Integer math Instruction	Add Integer Constant (16, 32-Bit)
+AR1	+AR1	Accumulator AR1	Add ACCU 1 to Address Register 1
+AR2	+AR2	Accumulator AR2	Add ACCU 1 to Address Register 2
+D	+D	Integer math Instruction	Add ACCU 1 and ACCU 2 as Double Integer (32-Bit)
+I	+I	Integer math Instruction	Add ACCU 1 and ACCU 2 as Integer (16-Bit)
+R	+R	Floating point Instruction	Add ACCU 1 and ACCU 2 as a Floating-Point Number (32-Bit IEEE-FP)
-D	-D	Integer math Instruction	Subtract ACCU 1 from ACCU 2 as Double Integer (32-Bit)
-I	-I	Integer math Instruction	Subtract ACCU 1 from ACCU 2 as Integer (16-Bit)
-R	-R	Floating point Instruction	Subtract ACCU 1 from ACCU 2 as a Floating-Point Number (32-Bit IEEE-FP)
ABS	ABS	Floating point Instruction	Absolute Value of a Floating-Point Number (32-Bit IEEE-FP)
ACOS	ACOS	Floating point Instruction	Generate the Arc Cosine of a Floating-Point Number (32-Bit)
ASIN	ASIN	Floating point Instruction	Generate the Arc Sine of a Floating-Point Number (32-Bit)
ATAN	ATAN	Floating point Instruction	Generate the Arc Tangent of a Floating-Point Number (32-Bit)
AUF	OPN	DB call	Open a Data Block
BE	BE	Program control	Block End
BEA	BEU	Program control	Block End Unconditional
BEB	BEC	Program control	Block End Conditional
BLD	BLD	Program control	Program Display Instruction (Null)
BTD	BTD	Convert	BCD to Integer (32-Bit)
BTI	BTI	Convert	BCD to Integer (16-Bit)
CALL	CALL	Program control	Call Block from a Library

CALL	CALL	Program control	Block Call
CALL	CALL	Program control	Call Multiple Instance
CC	CC	Program control	Conditional Cal
CLR	CLR	Bit logic Instruction	Clear RLO (=0)
COS	COS	Floating point Instruction	Generate the Cosine of Angles as Floating-Point Numbers (32-Bit)
DEC	DEC	Accumulator	Decrement ACCU 1
DTB	DTB	Convert	Double Integer (32-Bit) to BCD
DTR	DTR	Convert	Double Integer (32-Bit) to Floating-Point (32-Bit IEEE-FP)
ENT	ENT	Accumulator	Enter ACCU Stack
EXP	EXP	Floating point Instruction	Generate the Exponential Value of a Floating-Point Number (32-Bit)
FN	FN	Bit logic Instruction	Edge Negative
FP	FP	Bit logic Instruction	Edge Positive
FR	FR	Timers	Enable Timer (Free)
FR	FR	Counters	Enable Counter (Free) (free, FR C 0 to C 255)
INC	INC	Accumulator	Increment ACCU 1
INVD	INVD	Convert	Ones Complement Double Integer (32-Bit)
INVI	INVI	Convert	Ones Complement Integer (16-Bit)
ITB	ITB	Convert	Integer (16-Bit) to BCD
ITD	ITD	Convert	Integer (16-Bit) to Double Integer (32-Bit)
L DBLG	L DBLG	Load/Transfer	Load Length of Shared DB in ACCU 1
L DBNO	L DBNO	Load/Transfer	Load Number of Shared DB in ACCU 1
L DILG	L DILG	Load/Transfer	Load Length of Instance DB in ACCU 1
L DINO	L DINO	Load/Transfer	Load Number of Instance DB in ACCU 1
L	L	Load/Transfer	Load
L	L	Timers	Load Current Timer Value into ACCU 1 as Integer (the current timer value can be a number from 0 to 255, for example, L T 32)
L	L	Counters	Load Current Counter Value into ACCU 1 (the current counter value can be a number from 0 to 255, for example, L C 15)
L STW	L STW	Load/Transfer	Load Status Word into ACCU 1
LAR1 AR2	LAR1 AR2	Load/Transfer	Load Address Register 1 from Address Register 2
LAR1	LAR1	Load/Transfer	Load Address Register 1 from ACCU 1
LAR1 <D>	LAR1 <D>	Load/Transfer	Load Address Register 1 with Double Integer (32-Bit Pointer)
LAR2 <D>	LAR2 <D>	Load/Transfer	Load Address Register 2 with Double Integer (32-Bit Pointer)
LAR2	LAR2	Load/Transfer	Load Address Register 2 from ACCU 1
LC	LC	Counters	Load Current Counter Value into ACCU 1 as BCD (the current timer value can be a number from 0 to 255, for example, LC C 15)
LC	LC	Timers	Load Current Timer Value into ACCU 1 as BCD (the current counter value can be a number from 0 to 255, for example, LC T 32)
LEAVE	LEAVE	Accumulator	Leave ACCU Stack
LN	LN	Floating point Instruction	Generate the Natural Logarithm of a Floating-Point Number (32-Bit)
LOOP	LOOP	Jumps	Loop
MCR(	MCR(	Program control	Save RLO in MCR Stack, Begin MCR
)MCR	)MCR	Program control	End MCR
MCRA	MCRA	Program control	Activate MCR Area

MCRD	MCRD	Program control	Deactivate MCR Area
MOD	MOD	Integer math Instruction	Division Remainder Double Integer (32-Bit)
NEGD	NEGD	Convert	Twos Complement Double Integer (32-Bit)
NEGI	NEGI	Convert	Twos Complement Integer (16-Bit)
NEGR	NEGR	Convert	Negate Floating-Point Number (32-Bit, IEEE-FP)
NOP 0	NOP 0	Accumulator	Null Instruction 0
NOP 1	NOP 1	Accumulator	Null Instruction 1
NOT	NOT	Bit logic Instruction	Negate RLO
O	O	Bit logic Instruction	Or
O(	O(	Bit logic Instruction	Or with Nesting Open
OD	OD	Word logic Instruction	OR Double Word (32-Bit)
ON	ON	Bit logic Instruction	Or Not
ON(	ON(	Bit logic Instruction	Or Not with Nesting Open
OW	OW	Word logic Instruction	OR Word (16-Bit)
POP	POP	Accumulator	POP
POP	POP	Accumulator	CPU with Four ACCUs
POP	POP	Accumulator	CPU with Two ACCUs
PUSH	PUSH	Accumulator	CPU with Four ACCUs
PUSH	PUSH	Accumulator	CPU with Two ACCUs
R	R	Bit logic Instruction	Reset
R	R	Timers	Reset Timer (the current timer can be a number from 0 to 255, for example, R T 32)
R	R	Counters	Reset Counter (the current counter can be a number from 0 to 255, for example, R C 15)
RLD	RLD	Shift/Rotate	Rotate Left Double Word (32-Bit)
RLDA	RLDA	Shift/Rotate	Rotate ACCU 1 Left via CC 1 (32-Bit)
RND	RND	Convert	Round
RND-	RND-	Convert	Round to Lower Double Integer
RND+	RND+	Convert	Round to Upper Double Integer
RRD	RRD	Shift/Rotate	Rotate Right Double Word (32-Bit)
RRDA	RRDA	Shift/Rotate	Rotate ACCU 1 Right via CC 1 (32-Bit)
S	S	Bit logic Instruction	Set
S	S	Counters	Set Counter Preset Value (the current counter can be a number from 0 to 255, for example, S C 15)
SA	SF	Timers	Off-Delay Timer
SAVE	SAVE	Bit logic Instruction	Save RLO in BR Register
SE	SD	Timers	On-Delay Timer
SET	SET	Bit logic Instruction	Set
SI	SP	Timers	Pulse Timer
SIN	SIN	Floating point Instruction	Generate the Sine of Angles as Floating-Point Numbers (32-Bit)
SLD	SLD	Shift/Rotate	Shift Left Double Word (32-Bit)
SLW	SLW	Shift/Rotate	Shift Left Word (16-Bit)
SPA	JU	Jumps	Jump Unconditional
SPB	JC	Jumps	Jump if RLO = 1
SPBB	JCB	Jumps	Jump if RLO = 1 with BR
SPBI	JBI	Jumps	Jump if BR = 1
SPBIN	JNBI	Jumps	Jump if RLO = 0 with BR
SPBN	JCN	Jumps	Jump if RLO = 0
SPBNB	JNB	Jumps	Jump if RLO = 0 with BR
SPL	JL	Jumps	Jump to Labels
SPM	JM	Jumps	Jump if Minus

SPMZ	JMZ	Jumps	Jump if Minus or Zero
SPN	JN	Jumps	Jump if Not Zero
SPO	JO	Jumps	Jump if OV = 1
SPP	JP	Jumps	Jump if Plus
SPPZ	JPZ	Jumps	Jump if Plus or Zero
SPS	JOS	Jumps	Jump if OS = 1
SPU	JUO	Jumps	Jump if Unordered
SPZ	JZ	Jumps	Jump if Zero
SQR	SQR	Floating point Instruction	Generate the Square of a Floating-Point Number (32-Bit)
SQRT	SQRT	Floating point Instruction	Generate the Square Root of a Floating-Point Number (32-Bit)
SRD	SRD	Shift/Rotate	Shift Right Double Word (32-Bit)
SRW	SRW	Shift/Rotate	Shift Right Word (16-Bit)
SS	SS	Timers	Retentive On-Delay Timer
SSD	SSD	Shift/Rotate	Shift Sign Double Integer (32-Bit)
SSI	SSI	Shift/Rotate	Shift Sign Integer (16-Bit)
SV	SE	Timers	Extended Pulse Timer
T	T	Load/Transfer	Transfer ACCU 1 into Status Word
T	T	Load/Transfer	Transfer
TAD	CAD	Convert	Byte Sequence in ACCU 1 (32-Bit)
TAK	TAK	Accumulator	Toggle ACCU 1 with ACCU 2
TAN	TAN	Floating point Instruction	Generate the Tangent of Angles as Floating-Point Numbers (32-Bit)
TAR	CAR	Load/Transfer	Exchange Address Register 1 with Address Register 2
TAR1	TAR1	Load/Transfer	Transfer Address Register 1 to Address Register 2
TAR1	TAR1	Load/Transfer	Transfer Address Register 1 to ACCU 1
TAR1	TAR1	Load/Transfer	Transfer Address Register 1 to Destination (32-Bit Pointer)
TAR2	TAR2	Load/Transfer	Transfer Address Register 2 to ACCU 1
TAR2	TAR2	Load/Transfer	Transfer Address Register 2 to Destination (32-Bit Pointer)
TAW	CAW	Convert	Change Byte Sequence in ACCU 1-L (16-Bit)
TDB	CDB	Convert	Exchange Shared DB and Instance DB
TRUNC	TRUNC	Convert	Truncate
U	A	Bit logic Instruction	And
U(	A(	Bit logic Instruction	And with Nesting Open
UC	UC	Program control	Unconditional Call
UD	AD	Word logic Instruction	AND Double Word (32-Bit)
UN	AN	Bit logic Instruction	And Not
UN(	AN(	Bit logic Instruction	And Not with Nesting Open
UW	AW	Word logic Instruction	AND Word (16-Bit)
X	X	Bit logic Instruction	Exclusive Or
X(	X(	Bit logic Instruction	Exclusive Or with Nesting Open
XN	XN	Bit logic Instruction	Exclusive Or Not
XN(	XN(	Bit logic Instruction	Exclusive Or Not with Nesting Open
XOD	XOD	Word logic Instruction	Exclusive OR Double Word (32-Bit)
XOW	XOW	Word logic Instruction	Exclusive OR Word (16-Bit)
ZR	CD	Counters	Counter Down
ZV	CU	Counters	Counter Up