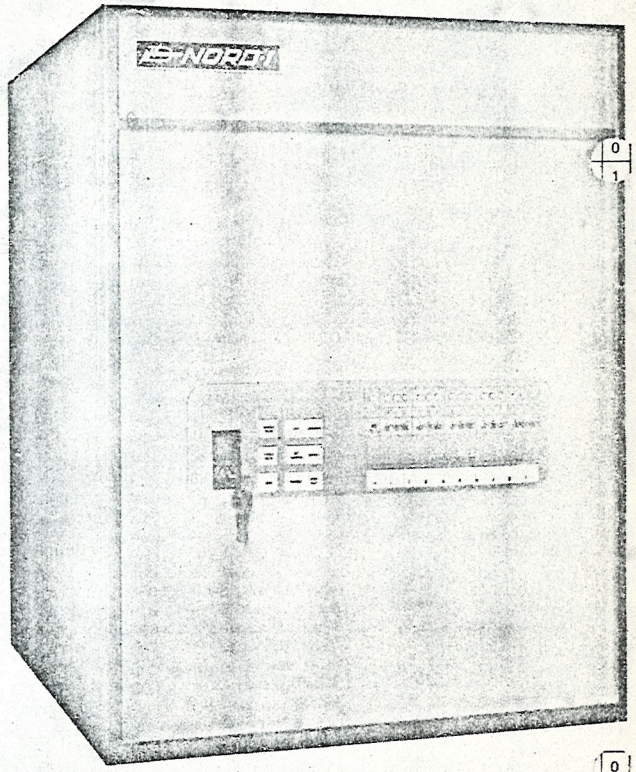
MAC COMMANDS

:	Following immediately after a symbol causes the value of the last symbol to be printed out
:	Following after a space or an octal number causes the value of the whole expression to be printed out
"	Set MAC in library mode
#	Form a 16 bit number equal to the 8 bit teletype codes of the following two characters
# #	Form a 16 bit number where bits 0–7 are equal to the following 8 bit teletype code. Bits 8–15 equal zero
# # # #	Form a 32 bit word containing up to 5 letters
%	Demarcation between instruction and comment
)LINE @	Set MAC in on-line mode
)TAPE $	Set MAC in off-line mode
!	Start execution of assembled program. The starting address must precede the command
←	Preceded by a symbol, will delete the reference to this symbol in the symbol table. If no reference is found, symbol in the US-table is deleted
)WRUS ?	Write undefined symbols
)FILL &	The value of expressions following a (, will automatically be stored in locations following current location
)CLEAR	Reset MAC tables
)PUNCH	Punch the content of the locations between lower and upper boundary, defined by two numbers, separated by the character <. Output on high speed punch
)PRINT	Same as)PUNCH, but output is on teletype
)BPUN	Binary dump of locations between lower and upper boundary must be followed by a symbol specifying the starting address. The program will automatically start after it has been read in
)ZERO	Clear locations between lower and upper boundary
)WMNE	Write the mnemonics allowed in MAC
)WLOC	Write the symbols in symbol table
)OUTM	Set MAC to outmode
)INMO	Set MAC to inmode
)WRITE	Followed by a string of symbols on the same line will cause the symbols and their octal values to be printed out during assembly-time, when in write mode
)NWRT	Reset the)WRTM command
)WRTM	Set MAC to write mode
)KILL	Followed by a string of symbols on the same line will delete the corresponding references in the symbol table. If a symbol is undefined, the last reference in the US-table is deleted
)PCL	Followed by a symbol, will cause all symbol references after this reference to be deleted
)DEC	Set MAC to decimal mode
)OCT	Set MAC to octal mode
)CORE	Print upper and lower boundary of available core
)LIST	List all defined symbols and their values
)CHANGE	Change all instructions or part of instruction between lower and upper boundary in this way: OLD / Old content NEW / New content MASK / Ones in bits to be changed a < b ; upper and lower boundary)CHANGE Only masked bits will be searched for and changed

BREAKPOINT

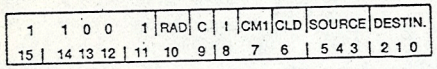
.	Set breakpoint in location specified before this command
!	Start execution in location specified before this command
1!	Step one forward in program sequence
0!	Continue without more breakpoints
)RBP	Reset breakpoint

REFERENCE DATA:



A/S NORSK DATA-ELEKTRONIKK

REGISTER OPERATION



Arithmetic operations, RAD 1:
C, O and Q may be affected by following instructions

RADD	146000	Add source to destination;	(D) = (D) + (S)
RSUB	146600	Subtract source from destination;	(D) = (D) — (S)
COPY	146100	Register transfer;	(D) = (S)
AD1	000400	Also add one to destination;	(D) = (D) + 1
ADC	001000	Also add old carry to destination;	(D) = (D) + C

Logical operations, RAD = 0:

SWAP	144000	Register exchange;	(S) = (D); (D) = (S)
RAND	144400	Logical AND to destination;	(D) = (D) ∧ (S)
REXO	145000	Logical exclusive OR;	(D) = (D) ∀ (S)
RORA	145400	Logical inclusive OR;	(D) = (D) ∨ (S)
CLD	000100	Clear destination before op.;	(D) = 0
CM1	000200	Use one's complement of source;	(S) = (S)o

Combined instructions:

EXIT	146142	= COPY SL DP,	Return from subroutine
RCLR	146100	= COPY,	Register clear
RINC	146400	= RADD AD1,	Register increment
RDCR	146200	= RADD CM1,	Register decrement
SMIL	146302	= COPY CM1 DP,	Starting address of MAC

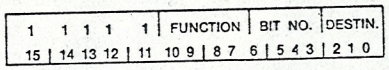
Specify source register S:

SD	000010	D-register as source
SP	000020	P-register as source
SB	000030	B-register as source
SL	000040	L-register as source
SA	000050	A-register as source
ST	000060	T-register as source
SX	000070	X-register as source
	000000	Source value equals zero

Specify destination register D:

DD	000001	D-register as destination
DP	000002	P-register as destination
DB	000003	B-register as destination
DL	000004	L-register as destination
DA	000005	A-register as destination
DT	000006	T-register as destination
DX	000007	X-register as destination
	000000	No operation

BIT OPERATION INSTRUCTIONS



Operation instruction:

BSKP	175000	Skip next location if specified condition is true;	P = P+1
BSET	174000	Set specified bit equal to specified condition.	
BSTA	176200	Store and clear K;	(B) = K; K = 0
BSTC	176000	Store complement and set K;	(B) = K_o ; K = 1
BLDA	176600	Load K;	K = (B)
BLDC	176400	Load bit complement to K;	K = $(B)_o$
BANC	177000	Logical AND with bit complement;	K = K ∧ (B)o
BORC	177400	Logical OR with bit complement;	K = K ∨ (B)o
BAND	177200	Logical AND to K;	K = K ∧ (B)
BORA	177600	Logical OR to K;	K = K ∨ (B)

Specify condition:

ZRO	000000	Specified bit equals zero;	(B) = 0
ONE	000200	one;	(B) = 1
BAC	000600	K;	(B) = K
BCM	000400	Complement specified bit;	(B) = $(B)_o$

Specify bit no:

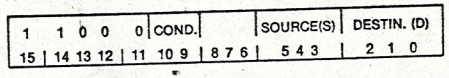
0	000000	Specifies bit in dest. reg.;	B = 0
10	000010		B = 1
20	000020		B = 2
⋮	⋮		⋮
170	000170		B = 15

For destination (D) mnemonics see section for register operations. D = 0 specifies STS-register.

Specify control flip-flop:

SSK	000020	Specifies; one bit accum.;	B = K
SSZ	000030	floating point ov.fl.;	B = Z
SSQ	000040	dynamic ov.fl.;	B = Q
SSO	000050	static ov.fl.;	B = O
SSC	000060	carry;	B = C
SSM	000070	multishift link;	B = M

SKIP INSTRUCTIONS



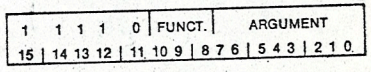
SKP	140000	Skip next location if specified condition is true;	P = P + 1

Specified condition:

EQL	000000	Equal to
UEQ	002000	Unequal to
GRE	001000	Greater or equal to
LST	003000	Less than
IF	000000	May be used freely to obtain
0	000000	easy readability

For Source and Destination mnemonics see section for Register operations.
The destination register (D) should be specified before the source register (S).

ARGUMENT INSTRUCTION



Function:

SAA	170400	Set argument to A;	A = ARG
AAA	172400	Add argument to A;	A = A + ARG
SAX	171400	Set argument to X;	X = ARG
AAX	173400	Add argument to X;	X = X + ARG
SAT	171000	Set argument to T;	T = ARG
AAT	173000	Add argument to T;	T = T + ARG
SAB	170000	Set argument to B;	B = ARG
AAB	172000	Add argument to B;	B = B + ARG

Argument is a signed number ranging from −128 to +127.

DEFINITION

Registers:

A	Accumulator, Main register
D	Extension register
T	Temporary register
L	Link register
X	Index register (post-index)
B	Base register (pre-index)
P	Program counter
STS	(F) Status word contains C, Q, O, Z, K, M and PIL
PID	Priority interrupt detect register
PIE	Priority interrupt enable register
MPR	Memory protect register
OPR	Operator panel register

Status word:

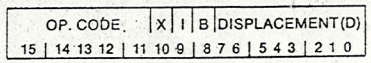
Bit no.

K	2	One bit accumulator
Z	3	Floating point overflow
Q	4	Dynamic overflow
O	5	Static overflow
C	6	Carry indicator
M	7	Multishift
PIL	8–11	Interrupt level indicator

Abbreviations:

EL	Effective location
EW	Effective word
DA	Double accumulator
FA	Floating accumulator
DW	Double word
FW	Floating word
∧	Logical AND
∨	Logical inclusive OR
∀	Logical exclusive OR
( )	The content of

MEMORY REFERENCE INSTRUCTIONS



Effective address:

	000000	Address relative to P;	EL = P ± D
,X	002000	Address relative to X;	EL = X ± D
I	001000	Indirect address;	EL = (P ± D)
,X I	003000	Post-indexing;	EL = (P ± D) + X
,B	000400	Address relative to B;	EL = B ± D
,X ,B	002400	Address relative to B and X;	EL = B ± D + X
I ,B	001400	Pre-indexing;	EL = (B ± D)
,X I ,B	003400	Pre- and Post-indexing;	EL = (B ± D) + X

Store instructions:

STZ	000000	Store zero;	(EL) = 0
STA	004000	Store A;	(EL) = A
STT	010000	Store T;	(EL) = T
STX	014000	Store X;	(EL) = X
MIN	040000	Memory increment and skip if zero;	(EL) = (EL) + 1

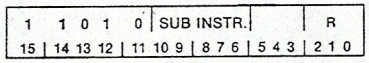
Load instructions:

LDA	044000	Load A;	A = (EL)
LDT	050000	Load T;	T = (EL)
LDX	054000	Load X;	X = (EL)

Arithmetical and logical instructions:

ADD	060000	Add to A (C, O and Q may also be affected);	A = A + (EL)
SUB	064000	Subtract from A (C and Q may also be affected);	A = A − (EL)
AND	070000	Logical AND to A;	A = A ∧ (EL)
ORA	074000	Logical inclusive OR to A;	A = A ∨ (EL)
MPY	120000	Multiply integer (O and Q may also be affected);	A = A * (EL)

Transfer instructions:

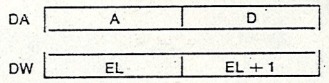


TRA	150000	Transfer specified register to A
TRR	150100	Transfer A to specified register
MCL	150200	Masked clear to specified register
MST	150300	Masked set to specified register

Specify register R:

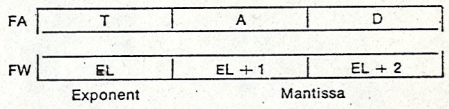
STS	000001	Status word. (PIL will not be affected by any instruction)
OPR	000002	Operator panel register
MPR	000003	Memory protection register
PID	000006	Priority interrupt detect register
PIE	000007	Priority interrupt enable register

Double word instructions:



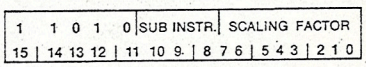
STD	020000	Store double word;	(DW) = DA
LDD	024000	Load double word;	DA = (DW)

Floating instructions:



STF	030000	Store floating accum.;	(FW) = FA
LDF	034000	Load floating accum.;	FA = (FW)
FAD	100000	Add to floating accum. (C may also be affected);	FA = FA + (FW)
FSB	104000	Subtract from floating accum. (C may also be affected);	FA = FA − (FW)
FMU	110000	Multiply floating accum. (C may also be affected);	FA = FA * (FW)
FDV	114000	Divide floating accum. (Z and C may also be affected);	FA = FA / (FW)

Floating conversion:



NLZ	151400	Convert the number in A to a floating number in FA
DNZ	152000	Convert the floating number in FA to a fixed point number in A
NLZ + 20	151420	Integer to floating conversion
DNZ − 20	152360	Floating to integer conversion

The range of scaling factor is −128 to 127 which gives converting range from 10^{-39} to 10^{39}.

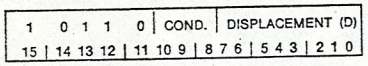
SEQUENCING INSTRUCTIONS

Unconditional jump:

For instruction word format and effective address, see section for memory reference instructions.

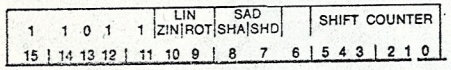
JMP	124000	Jump;	P = EL
JPL	134000	Jump to subroutine;	L = P; P = EL

Conditional jump:



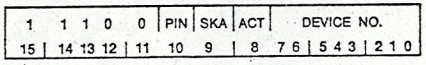
JAP	130000	Jump if: A is positive;	P = P ± D if: A ≥ 0
JAN	130400	A is negative;	A < 0
JAZ	131000	A is zero;	A = 0
JAF	131400	A is nonzero;	A ≠ 0
JXN	133400	X is negative;	X < 0
JXZ	133000	X is zero;	X = 0
JPC	132000	Increment X and jump if positive; X = X + 1; P = P ± D if X ≥ 0	
JNC	132400	Increment X and jump if negative; X = X + 1; P = P ± D if X < 0	

SHIFT INSTRUCTIONS



SHT	154000	Shift T-register
SHD	154200	Shift D-register
SHA	154400	Shift A-register
SAD	154600	Shift A- and D-register connected
	000000	Arithmetic shift. During right shift bit 15 is extended. During left shift zeros are shifted in from right
ROT	001000	Rotational shift. Most and least significant bits are connected together
ZIN	002000	Zero end input
LIN	003000	Link end input. The last vacated bit is fed to M after every shift instruction
SHR		Shift right; gives negative shift counter

INPUT - OUTPUT CONTROL

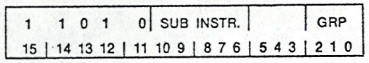


IOT 160000 Operate specified device according to function

Function:

ACT	000400	Activate the specified device
SKA	001000	Skip next instruction if device is ready
PIN	002000	Enable interrupt of specified device
SNI	000000	Skip next instruction if no interrupt

Interrupt control:



ION	150500	Turn on interrupt system
IOF	150400	Turn off interrupt system
INTEN	150540	Enable interrupt
INTDS	150440	Disable interrupt

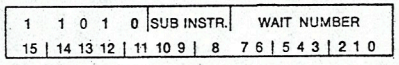
Specify Group:

GRP1	000001	Group one
GRP2	000002	Group two
GRP3	000004	Group three
GRP4	000010	Group four

Device no. for group one:

	Input	Output
Teletype I	RKE; 2	PNT; 3
Teletype II	4	5
Teletype III	10	11
20 mS real time clock	6	
High speed punch		PFA; 7
High speed reader	REA; 22	
Card reader	42	

Halt instruction:



WAIT 151000

I Interrupt off:
Halt the program. The program will continue by depressing continue button.

II Interrupt on:
Give up priority. If there are no interrupt requests on any level, the program on level zero is entered.

It is legal to specify a WAIT NUMBER less than 377.

MODEL 33 ASR/KSR TELETYPE CODE (ASCII) IN BINARY FORM

HOLE PUNCHED = MARK = 1
NO HOLE PUNCHED = SPACE = 0

			7	6	5	4	3		2	1	0
@	SPACE	NULL/IDLE				0	0		0	0	0
A	!	START OF MESSAGE				0	0		0	0	1
B	"	END OF ADDRESS				0	0		0	1	0
C	#	END OF MESSAGE				0	0		0	1	1
D	$	END OF TRANSMISSION				0	0		1	0	0
E	%	WHO ARE YOU				0	0		1	0	1
F	&	ARE YOU				0	0		1	1	0
G	'	BELL				0	0		1	1	1
H	(	FORMAT EFFECTOR				0	1		0	0	0
I	)	HORIZONTAL TAB				0	1		0	0	1
J	*	LINE FEED				0	1		0	1	0
K	+	VERTICAL TAB				0	1		0	1	1
L	,	FORM FEED				0	1		1	0	0
M	-	CARRIAGE RETURN				0	1		1	0	1
N	.	SHIFT OUT				0	1		1	1	0
O	/	SHIFT IN				0	1		1	1	1
P	0	DCO				1	0		0	0	0
Q	1	READER ON				1	0		0	0	1
R	2	TAPE (AUX ON)				1	0		0	1	0
S	3	READER OFF				1	0		0	1	1
T	4	(AUX OFF)				1	0		1	0	0
U	5	ERROR				1	0		1	0	1
V	6	SYNCHRONOUS IDLE				1	0		1	1	0
W	7	LOGICAL END OF MEDIA				1	0		1	1	1
X	8	S 0				1	1		0	0	0
Y	9	S 1				1	1		0	0	1
Z	:	S 2				1	1		0	1	0
[	;	S 3				1	1		0	1	1
\	<	S 4				1	1		1	0	0
]	=	S 5				1	1		1	0	1
↑	>	S 6				1	1		1	1	0
←	?	S 7				1	1		1	1	1

MOST SIGNIFICANT BIT
LEAST SIGNIFICANT BIT

0	0	SAME
0	1	SAME
1	0	SAME
1	1	SAME

RUB OUT

PARITY