



**EISCAT
TECHNICAL
NOTE**

**Pocket Manul for programming
the EISCAT digital correlator
by
Terrance Ho**

**KIRUNA
Sweden**

POCKET MANUAL FOR PROGRAMMING
THE EISCAT DIGITAL CORRELATOR

by

TERRANCE HO
MAX PLANCK INSTITUT
POSTFACH 20
D-3411 KATLENBURG-LINDAU 3
W. GERMANY

EISCAT Technical Note 81/28
Printed in Sweden
EISCAT Scientific Association
Kiruna, March 1981
ISSN 0349-2710

TABLE OF CONTENTS

	PAGE
I. INTRODUCTION	1
II. BASIC PRINCIPLES	1
III. THE DATA FIELD OF THE CORRELATOR	3
IV. THE PROGRAM FIELD OF THE CORRELATOR	4
V. PRO-INSTRUCTION SET	5
VI. APB/APM INSTRUCTION SET	10
VII. ARI/ACC INSTRUCTION SET	11
VIII. OUT INSTRUCTION SET	15
IX. I/O INSTRUCTION SET	17
X. THE STATUSWORD	20
XI. THE CONTROL WORD	22
XII. CORRSIM	23
XII.1 START OF CORRSIM	23
XII.2 EDITOR COMMANDS (CORRSIM)	23
XII.3 CORRLO COMMANDS	24
XIII. CORRTEST	25
XIII.1 START OF CORRTEST	25
XIII.2 EDITOR COMMANDS (CORRTEST)	25
XIII.3 USE OF FUNCTION PROTEST	25
XIII.4 USE OF FUNCTION ARITEST	26
XIV. REFERENCES	27
APPENDIX A	27

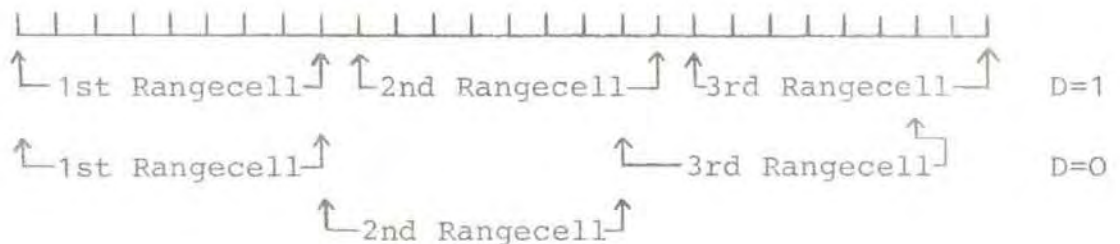
I. INTRODUCTION

This manual is intended for the "advanced" programmer who understands the principles of programming the correlator and is familiar with the software system. The aim of this manual is to consolidate all the commands and instructions available in the EISCAT correlator software system. Only brief explanations will be given when necessary. For more detailed explanations the reader should refer to the list of references.

II. BASIC PRINCIPLES

In the parameter set for the fixed subroutines and programs the following conventions are used:

- 1) The No. of Samples-1 in a Rangepcell means the No. of Samples-1 taken from a particular scattering volume.
- 2) The No. of Rangepcells-1 for Time Average means the number of scattering volumes (ranges) used for a particular time average.
- 3) The No. of Lags-1 in a Rangepcell means that the number of lags can be less than or equal to the number of samples.
- 4) The Rangepcell increment (D) means that overlapping of rangepcells can be obtained. If $D=1$ there is no overlapping. If $D \leq 0$ overlapping is obtained (see diagram).



The LCR1 register should be reloaded with 1).

The LCR2 register should be reloaded with 2).

The LCR3 register should be reloaded with 3).

The DATAI register should be used for the number of 64 BIT words that are to be transferred from the correlator to the NORD 10 computer. Note that the Transfer program must always start at location 32 in the RAM of the correlator.

The start address of the samples read from the Buffer memory is always assumed to be zero and the start address in the correlator result memory is always zero. The number of Start Computes counted is placed after the last sample in both channels as a negative number in the result memory.

III. THE DATA FIELD OF THE CORRELATOR

The Data Field of the correlator consists of the following registers:

<u>REGISTER NAME</u>	<u>ADDRESS</u>
STATUS WORD (STAT)	1
START ADDRESS REGISTER (SAR)	4
BASE ADDRESS REGISTER, APB (BAR)	5
DATAI REGISTER, APB	6
REGISTER STACK, APB (APBRS)	16,0 - 16,15
REGISTER STACK, APM (APMRS)	17,0 - 17,15
LOAD REGISTER FOR LOOP COUNTER 1 (LCR1)	18
TEMPORARY STORAGE FOR VALUE OF LOOP COUNTER 1 (LCR1A)	-
LOAD REGISTER FOR LOOP COUNTER 2 (LCR2)	19
LOAD REGISTER FOR LOOP COUNTER 3 (LCR3)	20
"CORRELATOR READY" REGISTER (CRA)	63

IV. THE PROGRAM FIELD OF THE CORRELATOR

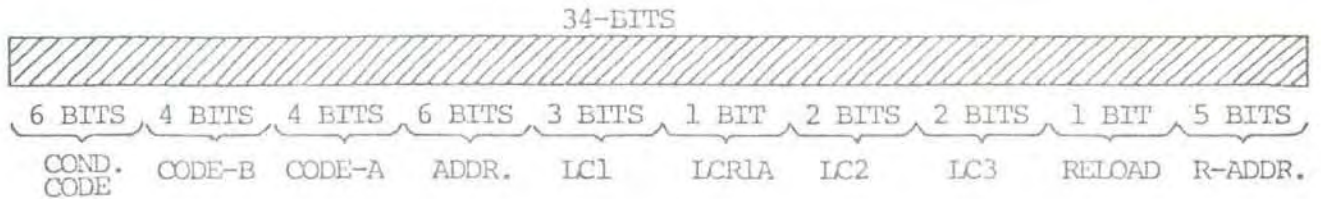
The Program Field of the correlator consists of 7 different functions which operate in parallel. These are:

- PRO-INSTRUCTION: Controls the execution of the correlator program and reload of the programmable registers in the data field (SAR, BAR, LCR1, LCR2, LCR3).
- APB-INSTRUCTION: Controls the addressing of the Buffer memory (data input source) and generation of reload values to the registers in the data field (APB register stack).
- APM-INSTRUCTION: Controls the addressing of the result memory (data storage) and generation of reload values to the registers in the data field (APM register stack).
- ARI-INSTRUCTION: Controls arithmetical functions on data from the Buffer memory.
- ACC-INSTRUCTION: Controls arithmetical accumulation of processed data and data from the result memory.
- OUT-INSTRUCTION: Controls direct memory access (DMA) from the result memory to the computer.
- I/O-INSTRUCTION: Controls data/address communication in a multi-correlator system.

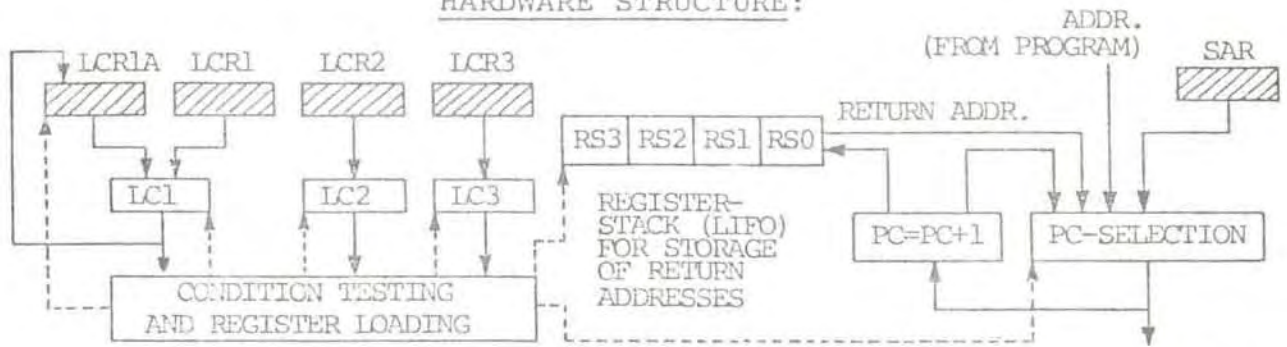
V PRO-INSTRUCTION SET

ABBREVIATIONS: LC - Loop counter; LCR - Load register for loop counter; LIFO - Last in first out; PC - Program counter.

INSTRUCTION-WORD SEPARATION:



HARDWARE STRUCTURE:



▨ : REGISTERS IN DATA-FIELD

PC = SUB.ADDR. FOR NEXT INSTRUCTION READ-OUT FROM PROGRAM-MEMORY

All loop-counters have 12 bits with data-range: 0-4095

CONDITION CODE INSTRUCTIONS:

<u>INTEGER CODING</u>	<u>TEST CONDITION</u>
32, 40, 48, 56	USE CODE A UNCONDITIONALLY
57	IF LC1=0 USE CODE B ELSE USE CODE A
58	IF LC2=0 USE CODE B ELSE USE CODE A
59	IF LC1 OR LC2=0 USE CODE B ELSE USE CODE A
60	IF LC3=0 USE CODE B ELSE USE CODE A
61	IF LC1 OR LC3=0 USE CODE B ELSE USE CODE A
62	IF LC2 OR LC3=0 USE CODE B ELSE USE CODE A
63	IF LC1 OR LC2 OR LC3=0 USE CODE B ELSE USE CODE A
50	IF LC2≠0 USE CODE B ELSE USE CODE A
51	IF LC1=0 OR LC2≠0 USE CODE B ELSE USE CODE A
54	IF LC2≠0 OR LC3=0 USE CODE B ELSE USE CODE A

INTEGER CODING

TEST CONDITION

55	IF LC1=0 OR LC2≠0 OR LC3=0 USE CODE B ELSE USE CODE A
44	IF LC3≠0 USE CODE B ELSE USE CODE A
45	IF LC1=0 OR LC3≠0 USE CODE B ELSE USE CODE A
46	IF LC2=0 OR LC3≠0 USE CODE B ELSE USE CODE A
47	IF LC1=0 OR LC2=0 OR LC3≠0 USE CODE B ELSE USE CODE A
38	IF LC2 OR LC3≠0 USE CODE B ELSE USE CODE A
39	IF LC1=0 OR LC2≠0 OR LC3≠0 USE CODE B ELSE USE CODE A
27	IF LC1=0 USE CODE B ELSE IF LC2=0 USE CODE A ELSE CONTINUE
30	IF LC3=0 USE CODE B ELSE IF LC2=0 USE CODE A ELSE CONTINUE
29	IF LC1 OR LC3=0 USE CODE B ELSE CONTINUE
31	IF LC1 OR LC3=0 USE CODE B ELSE IF LC2=0 USE CODE A ELSE CONTINUE
19	IF LC1=0 USE CODE B ELSE IF LC2≠0 USE CODE A ELSE CONTINUE
22	IF LC3=0 USE CODE B ELSE IF LC2≠0 USE CODE A ELSE CONTINUE
23	IF LC1 OR LC3=0 USE CODE B ELSE IF LC2≠0 USE CODE A ELSE CONTINUE
11	IF LC1≠0 USE CODE B ELSE IF LC2=0 USE CODE A ELSE CONTINUE
14	IF LC3≠0 USE CODE B ELSE IF LC2=0 USE CODE A ELSE CONTINUE
13	IF LC1 OR LC3≠0 USE CODE B ELSE CONTINUE
15	IF LC1 OR LC3≠0 USE CODE B ELSE IF LC2=0 USE CODE A ELSE CONTINUE
3	IF LC1≠0 USE CODE B ELSE IF LC2≠0 USE CODE A ELSE CONTINUE
5	IF LC3≠0 USE CODE B ELSE IF LC2≠0 USE CODE A ELSE CONTINUE
7	IF LC1 OR LC3≠0 USE CODE B ELSE IF LC2≠0 USE CODE A ELSE CONTINUE

CODE A/CODE B INSTRUCTIONS:

<u>INTEGER CODING</u>	<u>FUNCTION</u>	<u>COMMENTS</u>
0	PC=PC+1, POP STACK	CONTINUE AND DELETE LAST RETURN ADDRESS
1	PC=RETURN ADDRESS, POP STACK	RETURN AND DELETE LAST RETURN ADDRESS
2	PC=JUMP ADDRESS, POP STACK	JUMP AND DELETE LAST RETURN ADDRESS
3	PC=SAR, POP STACK	GOTO SAR AND DELETE LAST RETURN ADDRESS
4,12	PC=PC+1	CONTINUE
5,13	PC=RETURN ADDRESS	RETURN
6,14	PC=JUMP ADDRESS	JUMP
7,15	PC=SAR	GOTO SAR
8	PC=PC+1, PUSH STACK	CONTINUE AND SET RETURN ADDRESS
9	PC=RETURN ADDRESS, PUSH STACK	RETURN AND SET RETURN ADDRESS
10	PC=JUMP ADDRESS, PUSH STACK	JUMP AND SET RETURN ADDRESS
11	PC=SAR, PUSH STACK	GOTO SAR AND SET RETURN ADDRESS

LC1 INSTRUCTIONS:

<u>INTEGER CODING</u>	<u>FUNCTION</u>	<u>COMMENTS</u>
0	NOOP	NO OPERATION
1	LC1=LC1-1	DECREMENT LOOP COUNTER
2	LC1=LCR1	LOAD LOOP COUNTER
3	LC1=LCR1A	LOAD LOOP COUNTER
4	IF LC1=0 THEN LC1=LCR1, LC2=LC2-1 ELSE LC1=LC1-1	CONDITIONAL LOAD/DECREMENT
5	IF LC1=0 OR LC3=0 THEN LC1=LCR1A ELSE LC1=LC1-1	CONDITIONAL LOAD/DECREMENT
6	IF LC1=0 THEN LC1=LCR1 ELSE LC1=LC1-1	CONDITIONAL LOAD/DECREMENT
7	IF LC1=0 THEN LC1=LCR1A ELSE LC1=LC1-1	CONDITIONAL LOAD/DECREMENT

LC2 INSTRUCTIONS:

<u>INTEGER CODING</u>	<u>FUNCTION</u>	<u>COMMENTS</u>
0	NOOP	NO OPERATION
1	LC2=LC2-1	DECREMENT LOOP COUNTER
3	LC2=LCR2	LOAD LOOP COUNTER

LC3 INSTRUCTIONS:

<u>INTEGER CODING</u>	<u>FUNCTION</u>	<u>COMMENTS</u>
0	NOOP	NO OPERATION
1	LC3=LC3-1	DECREMENT LOOP COUNTER
2	LC3=LCR3	LOAD LOOP COUNTER
3	IF LC3=0 THEN LC3=LCR3 ELSE LC3=LC3-1	CONDITIONAL LOAD/ DECREMENT

LCR1A INSTRUCTIONS:

<u>INTEGER CODING</u>	<u>FUNCTION</u>	<u>COMMENTS</u>
0	NOOP	NO OPERATION
1	LCR1A=LC1	REGISTER LOAD

RELOAD INSTRUCTIONS:

<u>INTEGER CODING</u>	<u>FUNCTION</u>	<u>COMMENTS</u>
0	NOOP	NO OPERATION
1	REGISTER RELOAD	RELOAD VALUE FROM APB

RELOAD ADDRESSES:

<u>INTEGER CODING</u>	<u>FUNCTION</u>	<u>COMMENTS</u>
4	RELOAD SAR	IF RELOAD=1
5	" BAR, APB	" "
18	" LCR1	" "
19	" LCR2	" "
20	" LCR3	" "

PROGRAMMING NOTES

Only the given instruction values should be used.

Only a maximum of 4 return addresses can be stored.

When the push stack option is used (set up return address) the stored value is the present PC value+1.

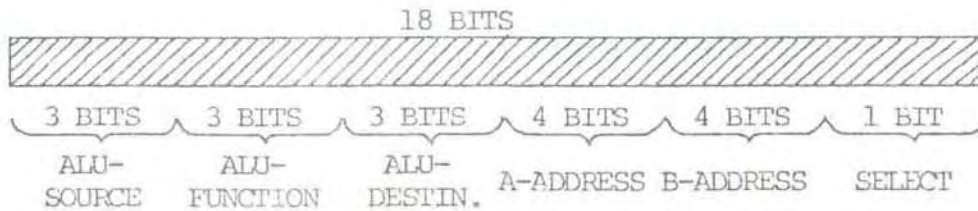
When the RELOAD option is used the next program instruction must not contain the RELOAD option or a load LOOP COUNTER instruction. The correlator needs 2 clock cycles for a RELOAD instruction.

Program Location 0 cannot be used for a register reload.

VI. APB/APM INSTRUCTION SET

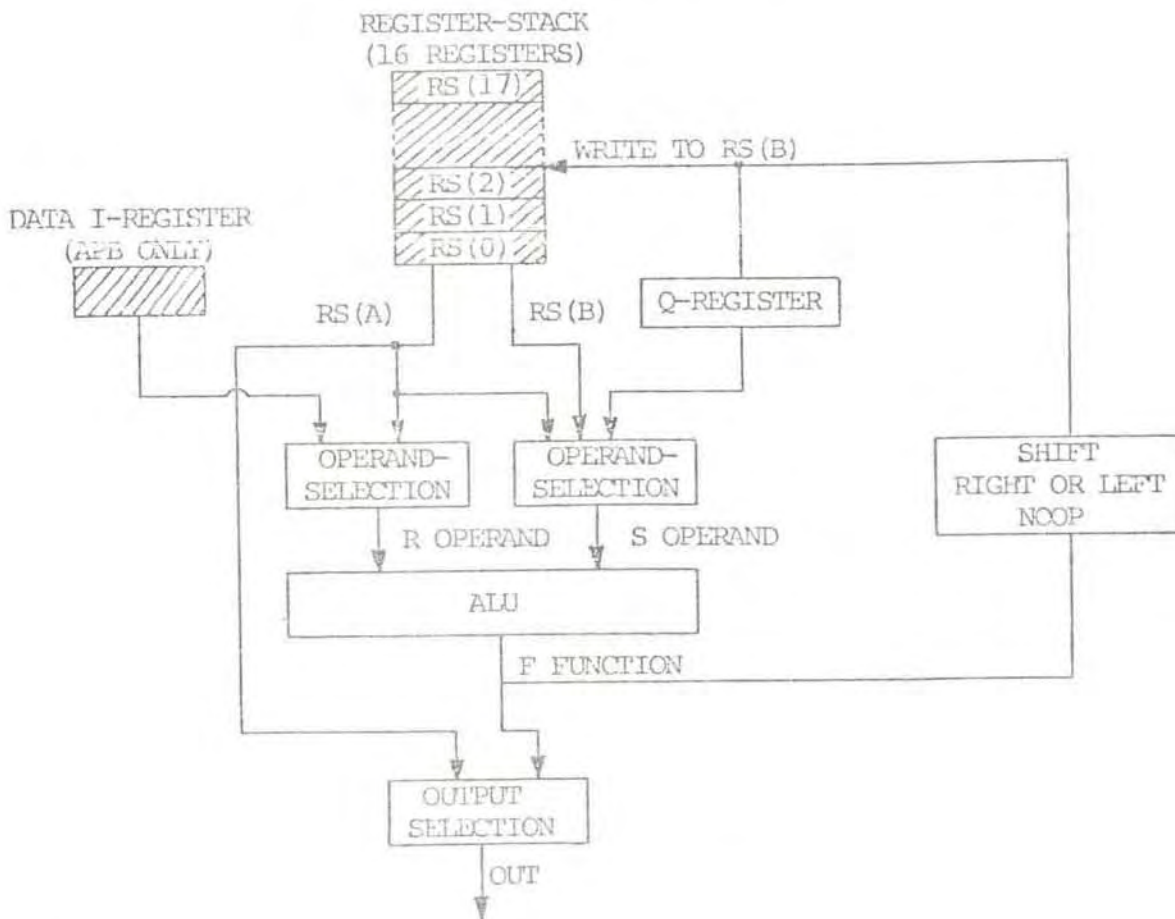
ABBREVIATIONS: APB - Address processor for Buffer memory;
 APM - Address processor for result memory;
 ALU - Arithmetical logical unit.

INSTRUCTION-WORD SEPARATION:



APM does not have the SELECT sub-instruction.

HARDWARE STRUCTURE:



For APB: Address-bus to buffer memory or reload value to registers.

For APM: Read/write address to result memory.

Numeric range for APB: 0 - 65535 (16 BITS)

Numeric range for APM: 0 - 4095 (12 BITS)

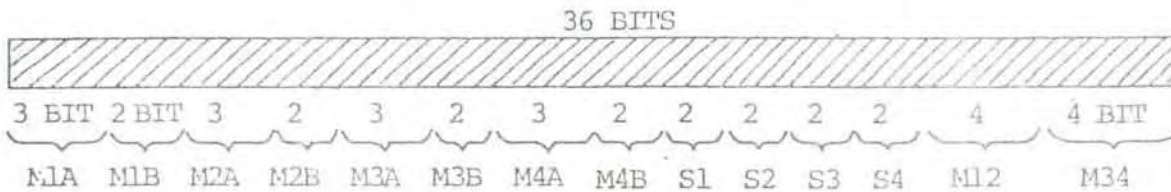
	<u>INTEGER CODING</u>	<u>FUNCTION</u>
<u>ALU-SOURCE:</u>	0	R=RS (A) , S=Q
	1	R=RS (A) , S=RS (B)
	2	R=0, S=Q
	3	R=0, S=RS (B)
	4	R=0, S=RS (A)
	5	R=DATAI, S=RS (A)
	6	R=DATAI, S=Q
	7	R=DATAI, S=0
<u>ALU-FUNCTION:</u>	0	F=R+S
	1	F=S-R
	2	F=R-S
	3	F=R OR S
	4	F=R AND S
	5	F= \bar{R} AND S
	6	F=R EXOR S
	7	F=R EXNOR S
<u>ALU-DESTINATION:</u>	0	Q=F, OUT=F
	1	OUT=F
	2	RS (B) =F, OUT=RS (A)
	3	RS (B) =F, OUT=F
	4	RS (B) =F/2, Q=Q/2, OUT=F
	5	RS (B) =F/2, OUT=F
	6	RS (B) =2F, Q=2Q, OUT=F
	7	RS (B) =2F, OUT=F
A-ADDRESS/B-ADDRESS	0-15	A-ADDRESS IS ALWAYS A READ ADDRESS. B-ADDRESS IS A READ AND WRITE ADDRESS (THE SAME REGISTER STACK LOCATION CAN ALSO BE MODIFIED).
<u>SELECT:</u>	0	NOOP
	1	WHENEVER RS(B) IS REFERENCED, THE B-ADDRESS IS INHIBITED AND RS(LC1) IS USED INSTEAD. (LC1=LOOP COUNTER 1)

Note that the APM does not have the SELECT option.

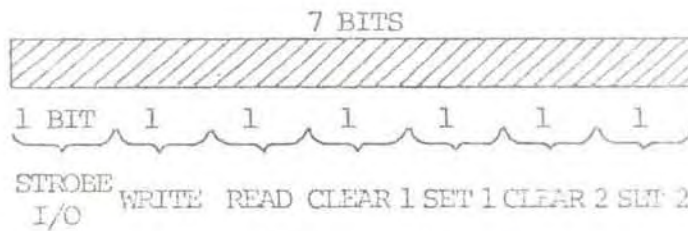
VII. ARI/ACC INSTRUCTION SET

ABBREVIATIONS: ARI - Arithmetical part of data processing;
 ACC - Data accumulators; ALU - Arithmetical
 logical unit; M1A - Operand A selection to
 multiplier 1; M1B - Operand B selection to
 multiplier 1; S1 - Strobe signals to multiplier 1;
 M12 - Arithmetical function on results from
 multiplier 1 and 2.

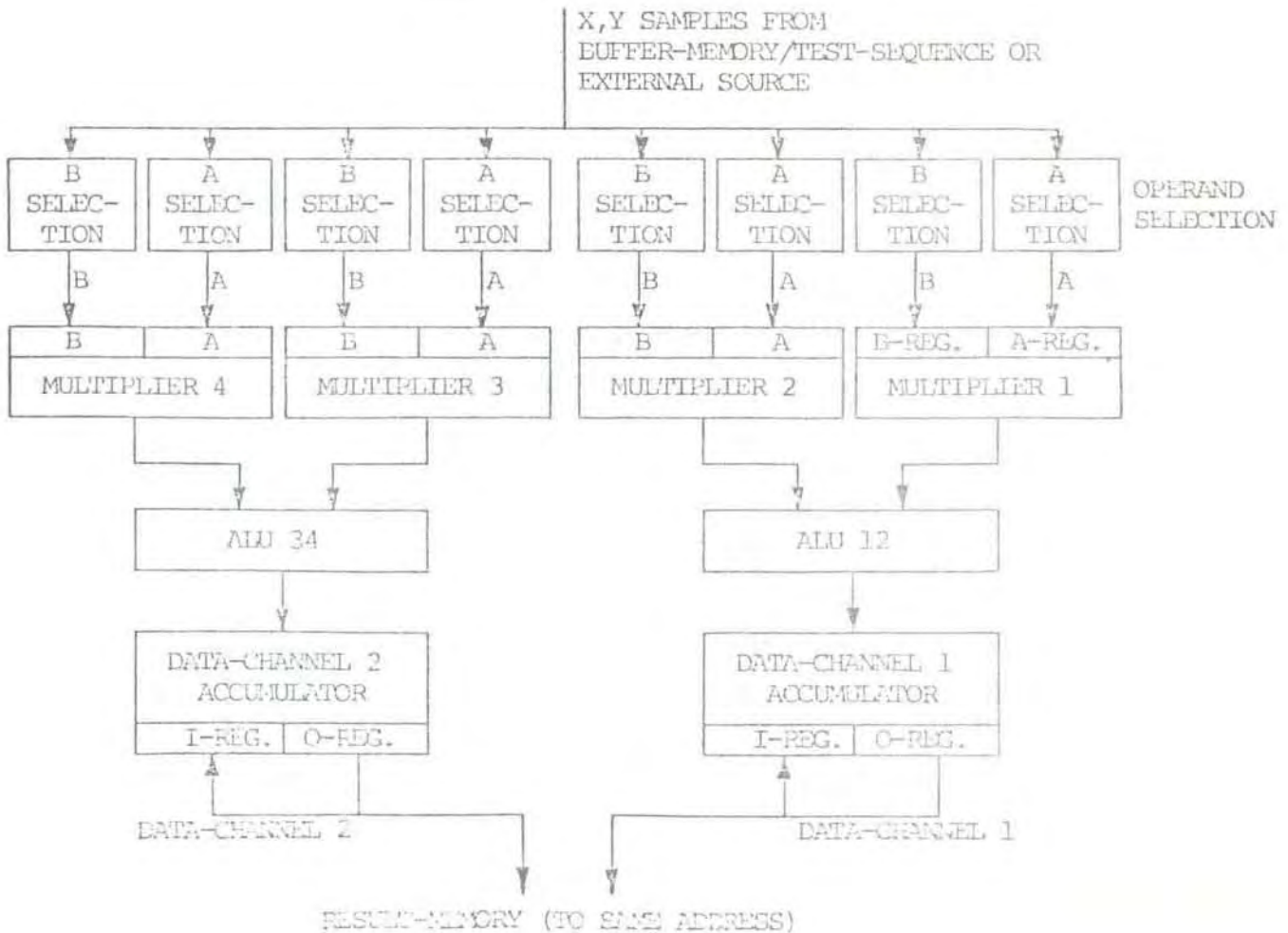
ARI INSTRUCTION-WORD SEPARATION:



ACC INSTRUCTION-WORD SEPARATION:



HARDWARE STRUCTURE:



	INTEGER CODING	FUNCTION	COMMENTS
<u>M1A:</u>	0	A=X INTERNAL	SAME FOR ALL MULTIPLIERS
	1	A=Y INTERNAL	
	2	A=X EXTERNAL	
	3	A=Y EXTERNAL	
	4-7	A=1	
<u>M2B:</u>	0	B=X INTERNAL	SAME FOR ALL MULTIPLIERS
	1	B=Y INTERNAL	
	2	B=X EXTERNAL	
	3	B=Y EXTERNAL	
<u>S1:</u>	0	NOOP	SAME FOR ALL MULTIPLIERS
	1	LOAD NEW OPERAND INTO A-REG.	
	2	LOAD NEW OPERAND INTO B-REG.	
	3	LOAD NEW OPERANDS INTO A- AND B-REGS.	
<u>M12:</u>	5	ALU 12=M2	SAME FOR ALU 34
	6	ALU 12=M1-M2	
	9	ALU 12=M1+M2	
	12	ALU 12=-1	
	15	ALU 12=M1	
<u>STROBE I/O:</u>	0	NOOP	DATA CHANNEL 1 AND 2 ACCUMULATORS ARE OPERATED IN PARALLEL
	1	STROBE NEW CONTENT INTO I/O REGS.	
<u>WRITE:</u>	0	NOOP	
	1	WRITE O-REG. TO RESULT MEMORY	
<u>READ:</u>	0	INTERNAL ACCUMULATION SELECTED	DEPENDENT ON VALUE OF SET 1 AND SET 2
	1	READ RESULT MEMORY TO I-REG. OR SET I-REG.=0	

	<u>INTEGER CODING</u>	<u>FUNCTION</u>	<u>COMMENTS</u>
<u>SET 1:</u>	0	IF READ=1 AND SET 2=0 THEN RESET I-REG (=0)	SET 1 IS ACTIVE UNTIL CLEAR 1=1
	1	IF READ=1 THEN GENERATE READ	
<u>CLEAR 1:</u>	0	NOOP	
	1	RESET SET 1=0	
<u>SET 2:</u>	0	ENABLE SET 1 CONTROL OF READ	SET 2 IS ACTIVE UNTIL CLEAR 2=1
	1	INHIBIT SET 1 CONTROL OF READ	SET 2 CAN ALSO BE SET FROM STATUSWORD
<u>CLEAR 2:</u>	0	NOOP	
	1	RESET SET 2=0	

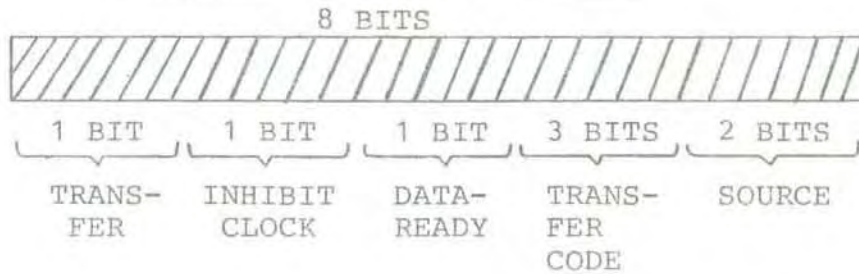
PROGRAMMING RESTRICTIONS:

It is allowed to select data read, accumulate and data write to the same result memory in one instruction cycle. However, it is not allowed to have a read and/or write operation to the same memory location in the next instruction cycle. This is due to the pipelining structure of the correlator.

VIII. OUT INSTRUCTION SET

ABBREVIATIONS: DMA - Direct memory access; LS - Least significant part; MS - Most significant part.

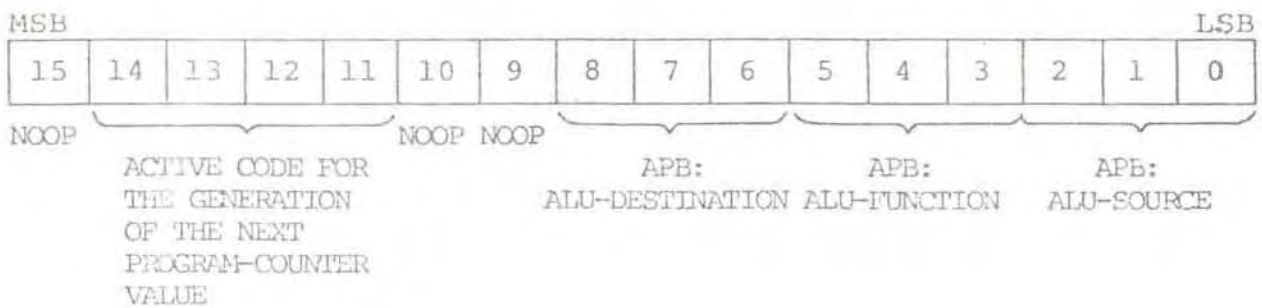
INSTRUCTION-WORD SEPERATION:



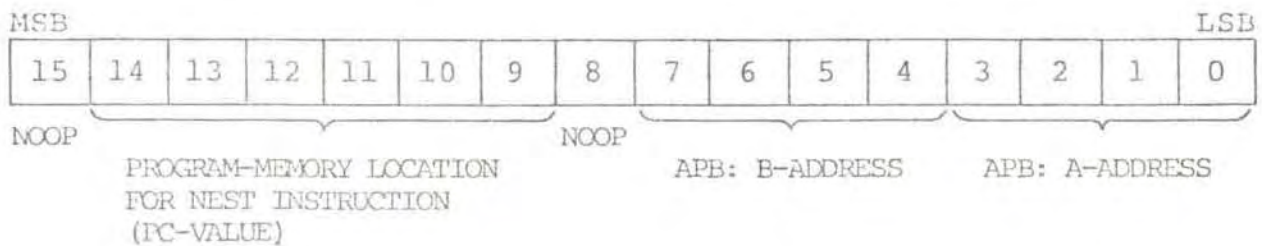
	INTEGER CODING	FUNCTION	COMMENTS
<u>TRANSFER:</u>	0	NOOP	TRANSFER NOT SELECTED
	1	ENABLE CORRELATOR FOR TRANSFER	MUST BE SET FOR TRANSFER
<u>INHIBIT CLOCK:</u>	0	ENABLE INTERNAL CLOCK OPERATION	
	1	INHIBIT CLOCK AND PROGRAM IS STOPPED	REQUIRES THAT DATA-READY IS <u>ALWAYS</u> SET
<u>DATA-READY:</u>	0	NOOP	
	1	SET "DATA-READY" TO CAMAC	"DATA-RECEIVED" FROM CAMAC. PROGRAM ADVANCES <u>ONE</u> INSTRUCTION CYCLE
<u>TRANSFER-CODE:</u>	0	STATUS WORD	ACTIVE ON DMA-CHANNEL
	1	CONTROL WORD	AND FRONT PANEL
	2	RES. MEM, DATA-CHAN1:LS	GIVEN BY SOURCE INSTR.
	3	RES. MEM, DATA-CHAN1:MS	" " " "
	4	RES. MEM, DATA-CHAN2:LS	" " " "
	5	RES. MEM, DATA-CHAN2:MS	" " " "
	6	TEST-WORD 1	
7	TEST-WORD 2		

<u>SOURCE:</u>	<u>INTEGER CODING</u>	<u>FUNCTION</u>	<u>COMMENTS</u>
	0	MEM. VALUES FROM MASTER MODULE	
	1	MEM. VALUES FROM SLAVE 1	} MUST BE USED ONLY FOR TRANSFER CODES: 2-5
	2	MEM. VALUES FROM SLAVE 2	
	3	MEM. VALUES FROM SLAVE 3	

TEST-WORD 1 STRUCTURE:



TEST-WORD 2 STRUCTURE:



PROGRAMMING RESTRICTIONS:

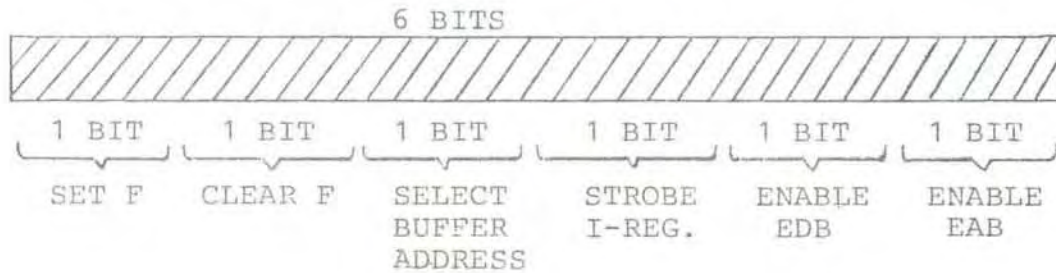
All Transfer programs should start at memory location 32. This value corresponds to the entry-point automatically loaded when the real-time command "data-transfer" is received from the radar-controller.

The hardware construction requires that before TRANSFER is set to 1 there must be at least 2 dummy instructions before it and that before TRANSFER is set to 0 (end of Transfer program) there must be at least 4 dummy instructions without INHIBIT CLOCK before it.

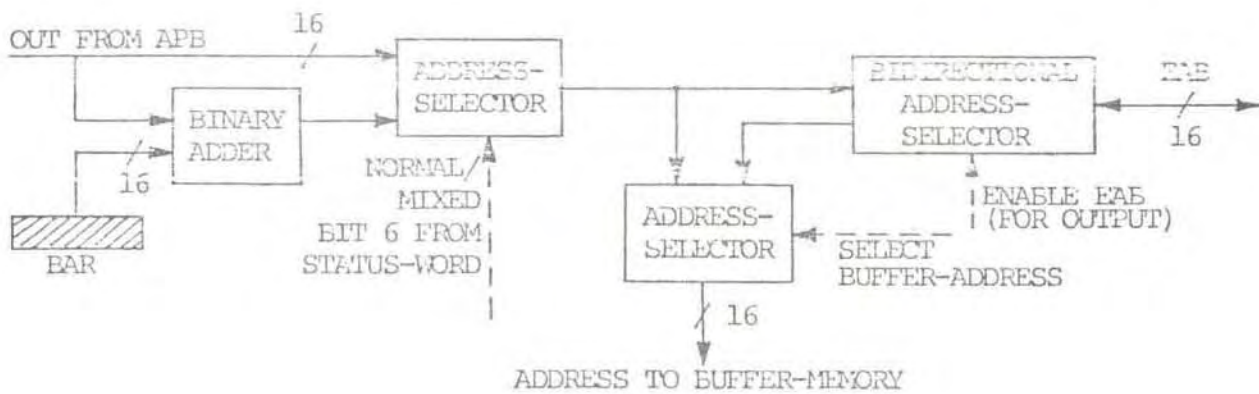
IX. I/O INSTRUCTION SET

ABBREVIATIONS: BAR - Base address register; EDB - External data bus;
 EAB - External address bus; APB - Address processor
 for buffer memory.

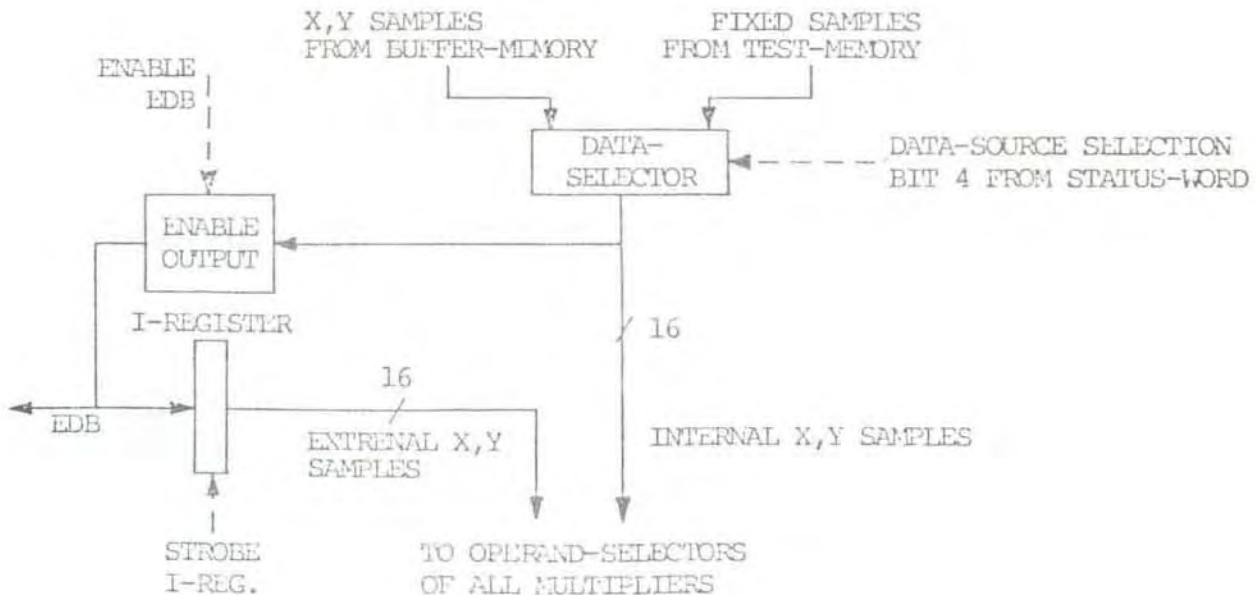
INSTRUCTION-WORD SEPARATION:



BUFFER-MEMORY ADDRESS GENERATION:



DATA-INPUT STRUCTURE:



	<u>INTEGER CODING</u>	<u>FUNCTION</u>
<u>SET F:</u>	0	NOOP
	1	SET FLAG
<u>CLEAR F:</u>	0	NOOP
	1	CLEAR FLAG
<u>SELECT BUFFER ADDRESS:</u>	0	BUFFER ADDRESS GENERATED BY INTERNAL HARDWARE
	1	BUFFER ADDRESS GERERATED BY EAB
<u>STROBE I-REG.:</u>	0	NOOP
	1	STROBE
<u>ENABLE EDB:</u>	0	NOOP
	1	INTERNAL X,Y SAMPLES ACTIVE ON EDB
<u>ENABLE EAB:</u>	0	NOOP
	1	INTERNAL ADDRESS ACTIVE ON EAB

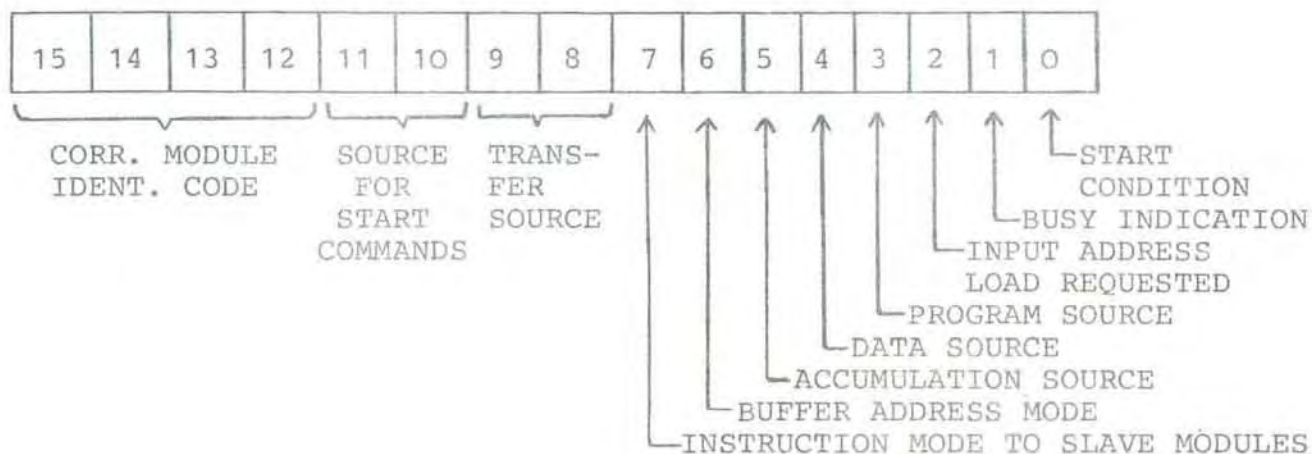
When the normal addressing mode is selected, set by bit 6 in the Statusword, the first internal address selector is always the output value from the APB. When the mixed addressing mode is selected the first internal address selector generates the output value from the APB plus the content of BAR in the first half of the instruction cycle, in the second half the output value from the APB is generated.

The operand register of the multipliers are (when selected) strobed at the end of the second half of the instruction cycle. This gives that when the internal address is selected for the buffer memory the samples with the address reference output value from the APB are always strobed (independent of normal/mixed mode). The I-register connected to the EDB is strobed at the end of the first half of the instruction cycle and can be used for temporary storage of samples with the address reference output value from the APB plus the content of BAR (only when mixed addressing mode is selected).

The correlator can generate a flag signal (rear side connector) which can be set/cleared by the SET F/CLEAR F bits in the instruction word. Both the external data and address busses (EDB and EAB) are bidirectional and are intended for use in multi-correlator systems for possible data/address value transfers between modules.

X. THE STATUSWORD

STATUSWORD FUNCTIONS



<u>BIT</u>	<u>FUNCTION</u>	<u>COMMENTS</u>
15-12 (P1)	CORRELATOR MODULE IDENT. CODE (0-15)	MUST BE SET TO VALUE SHOWN ON FRONT PANEL
11-10 (P2)	0: FRONT PANEL 1: RADAR CONTROLLER 2: COMPUTER	ENABLE THE GIVEN SOURCE FOR STARTING THE CORRELATOR
9-8	0: MASTER MODULE 1: SLAVE 1 2: SLAVE 2 3: SLAVE 3	SET BY TRANSFER INSTRUCTION
7 (P3)	0: SAME ARI INSTRUCTIONS TO SLAVES 1: MODIFIED OPERAND SELECTIONS TO SLAVE MULTIPLIERS	SWITCHES INTERNAL DATA TO EXTERNAL DATA
6 (P4)	0: NORMAL ADDRESS MODE 1: MIXED ADDRESS MODE	SEE I/O INSTRUCTIONS
5 (P5)	0: START EXPERIMENT 1: CONTINUE EXPERIMENT	IF SET 1=0, ACC INSTR. GENERATE SET 2=1
4 (P6)	0: INTERNAL DATA FROM BUFFER MEMORY 1: INTERNAL DATA FROM TEST MEMORY	SEE I/O INSTRUCTIONS
3 (P7)	0: PROGRAMMABLE MEMORY 1: MEMORY FOR FIXED INTERNAL PROGRAMS	RAM MODULES USE OF PROM WITH SAR

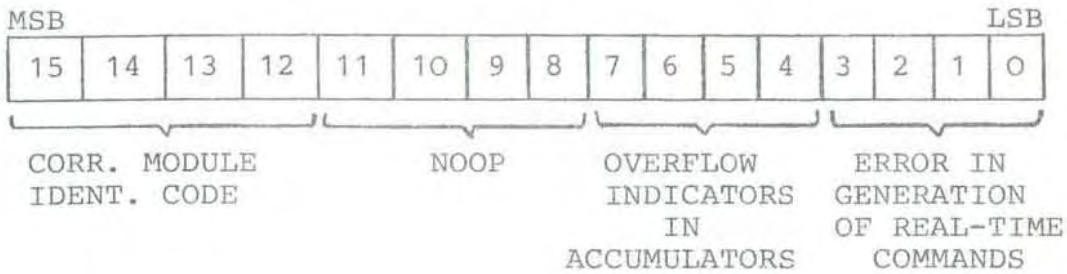
<u>BIT</u>	<u>FUNCTION</u>	<u>COMMENTS</u>
2	0: READY FOR LOADING REGS./ PROG. MEMORY LOCATIONS 1: ADDRESS TO DATA OR PROGRAM FIELD LOADED	
1	0: CORRELATOR NOT ACTIVE 1: CORRELATOR BUSY	SIGNAL ALSO AVAILABLE AT CONNECTOR ON REAR SIDE
0	0: NOT READY FOR ACCEPTING START COMMAND 1: CORRELATOR READY FOR START	SET BY DATA FIELD ADDRESS 63 (CRA)

P denotes which functions can be set in the Statusword.

In CORRIM the functions must be entered in the following order:
P1,P2,P3,P4,P5,P6,P7,. Note that each parameter must be separated
with a comma.

XI. THE CONTROL WORD

CONTROL WORD FUNCTIONS



The given function is active when bit is set:

<u>BIT</u>	<u>FUNCTION</u>
0	INPUT PROGRAM REQUEST HAS BEEN ISSUED BUT CORRELATOR BUSY
1	START SIGNAL RECEIVED FROM COMPUTER OR RADAR CONTROLLER BUT CORRELATOR IN MANUAL OPERATION
2	START SIGNAL RECEIVED BUT CORRELATOR BUSY
3	START SIGNAL RECEIVED BUT CORRELATOR NOT READY
4	ACCUMULATOR OVERFLOW IN SLAVE 1
5	ACCUMULATOR OVERFLOW IN SLAVE 2
6	ACCUMULATOR OVERFLOW IN SLAVE 3
7	ACCUMULATOR OVERFLOW IN MASTER MODULE

When set the error bits remain active until the real time command correlator reset is given.

The Control word is shown on the front panel.

XII. CORRSIM

CORRSIM offers a systematic way of real-time correlator programming and with the additional options for interactive communication with the correlator creates an effective system for external hardware testing and control.

XII.1 START OF CORRSIM

The program is stored on the user-file (CORR-TEST)CORRSIM and is started with the SINTRAN command: (CORR-TEST)CORRSIM or simplified: (CO-TE)CORRSIM

XII.2 EDITOR COMMANDS (CORRSIM)

A	Address assignments of data/program fields of the correlator.
D	Delete data and program fields.
DD	Delete data field.
DP	Delete program field.
E,EX,EXIT	Exit from EDITOR.
F,FIN	Terminate CORRSIM.
H,HELP	Lists valid commands for EDITOR.
LD	All defined data registers are displayed.
LP	All defined program locations are displayed in Octal as RAM modules.
LPD	All defined program locations are decoded.
LPS:PRO	All defined program instructions are displayed.
LPS:APB	All defined instructions for buffer address processor are displayed.
LPS:APM	All defined instructions for result memory address processor are displayed.
LPS:ARI:	All defined instructions for correlator arithmetic are displayed.
LPS:ACC	All defined instructions for accumulators are displayed.
LPS:I/O	All defined instructions for input/output bus handling are displayed.
LPS:OUT	All defined instructions for DMA output are displayed.
PDP	All defined data registers and program locations in Octal as RAM modules are printed.
PPS	All defined program functions are printed separately.
PPD	All defined program locations are printed decoded.
PEND	To obtain output on line-printer (ie after PDP,PPS,PPD).

R Read from file.
SDF Data registers are defined.
SPS:PRO Program instructions are defined.
SPS:APB Buffer address processor instructions are defined.
SPS:APM Result memory address processor instructions are defined.
SPS:ARI Correlator arithmetic instructions are defined.
SPS:ACC Accumulator instructions are defined.
SPS:I/O Input/output instructions are defined.
SPS:OUT DMA output instructions are defined.
W Write to file.

XII.3 CORRLO COMMANDS

ADC Set A/D converter.
B Start the correlator.
BUFF Set buffer memory.
C Status of CAMAC word-count and memory-address registers are displayed.
D All defined registers in the data field are displayed.
DISP Sets the output on LED displays of the buffer memory.
E Exit from CORRLO.
F Terminate CORRSIM.
G Real and Imaginary parts of data are displayed.
H,HELP Lists valid commands for CORRLO.
I Data is displayed as transferred from correlator.
INT Sets processor in integer mode.
L Load the correlator.
LM Outputs the MODE status.
M Set mode for data transfer from correlator.
NODMA Initialize CORRSIM to "grab" data from RT-COMMON.
OCT Sets processor in Octal mode.
P Direct jump to EDITOR.
PRINT Outputs data on line printer.
R Reset the correlator.
S Set value, either correlator READY or STATUSWORD.
V Displays data on the terminal.

XIII. CORRTEST

CORRTEST offers a systematic way of testing correlator programs off-line by simulating the correlator hardware and producing outputs of each micro-instruction as it loops through the micro-program. Furthermore the data results can be stored on a file which can be compared directly with the data results from the correlator when using CORRSIM.

XIII.1 START OF CORRTEST

The program is stored on the user-file (CORR-TEST)CORRTEST and is started with the SINTRAN command: (CORR-TEST)CORRTEST or simplified: (CO-TE)CORRTEST

XIII.2 EDITOR COMMANDS (CORRTEST)

A	Address assignments of the data field of the correlator.
D	Delete data and program fields.
DD	Delete data field.
E,EX,EXIT	Exit from EDITOR.
F,FIN	Terminate CORRTEST.
H,HELP	Lists all valid commands for EDITOR.
LD	All defined data registers are displayed.
R	Read from file.
SDF	Data registers are defined.
W	Write to file.

NOTE: With the R command an Entry-Point in the correlator memory is not asked for.

XIII.3 USE OF FUNCTION PROTEST

PROTEST allows the user to have a "real-time" simulation of the correlator hardware. The program will execute cycle by cycle all defined micro-program instructions except those of the OUT instructions and tests of validity of those instructions are performed. An output mapping of the program-counter, the LIFO register stacks, the loop counters, the LCR1A register, output values of the APB and APM processors, the values of the multipliers, the values of the ALU's, the values of the data channels and an

indication of whether a read/write was made from/to the result memory is given.

At the end of the simulation a calculation is made of the required CPU time when executed in the physical system.

XIII.4 USE OF FUNCTION ARITEST

ARITEST simulates the arithmetical hardware section of the correlator for fixed programs and uses the same registers as defined for the basic subroutines, see appendix A.

The following programs are available:

- PROG1: POWER PROFILE PROGRAM (VER.1)
- PROG2: POWER PROFILE PROGRAM (VER.2)
- PROG3: SINGLE PULSE (NO. OF LAGS .EQ. NO. OF SAMPLES)
- PROG4: SINGLE PULSE (NO. OF LAGS .LE. NO. OF SAMPLES)
- PROG5: MULTI PULSE PROGRAM
- PROG6: CROSS CORRELATION (NO. OF LAGS .EQ. NO. OF SAMPLES)
- PROG7: CROSS CORRELATION (NO. OF LAGS .LE. NO. OF SAMPLES)

ACKNOWLEDGEMENT

I would like to thank the EISCAT Scientific Association for sponsering this work and to all of the staff in Tromsø for their help.

XIV. REFERENCES

- HANS-JØRGEN ALKER: "A PROGRAMMABLE CORRELATOR MODULE FOR THE EISCAT RADAR SYSTEM."
EISCAT Technical Notes No. 79/11, 1979
- TERRANCE HO and HANS-JØRGEN ALKER: "SCIENTIFIC PROGRAMMING OF THE EISCAT DIGITAL CORRELATOR (REVISED)."
EISCAT Technical Notes No. 81/24, 1981
- TERRANCE HO: "PROGRAMS CORRSIM, CORRTEST: SYSTEM FOR PROGRAM DEVELOPMENT AND SOFTWARE SIMULATION OF EISCAT DIGITAL CORRELATOR. USER'S MANUAL."
EISCAT Technical Notes No. 81/25, 1981
- TERRANCE HO: "INSTRUCTION MANUAL FOR EISCAT DIGITAL CORRELATOR (REVISED)."
EISCAT Technical Notes No. 81/26, 1981
- TERRANCE HO: "STANDARD SUBROUTINES AND PROGRAMS FOR EISCAT DIGITAL CORRELATOR."
EISCAT Technical Notes No. 81/27, 1981

APPENDIX A

The parameters which have to be defined for the basic sub-routines are given here. Note that when using ARITEST the status-word must also be defined. A dummy correlator IDENT-CODE can be given in order to define the statusword.

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: POWER PROFILE SUBROUTINE (VERSION 1)

FILE-NAME (NORD 10): PROGØ:DATA

PROGRAM DESCRIPTION:

DATA CHANNEL 1 ZERO LAG ESTIMATION $K_r = \sum_{i=0}^{N-1} (X_{i+(N+D-1)(r-1)}^2 + Y_{i+(N+D-1)(r-1)}^2)$

DATA CHANNEL 2 MEAN VALUE ESTIMATION $M_r = \sum_{i=0}^{N-1} (X_{i+(N+D-1)(r-1)} + Y_{i+(N+D-1)(r-1)})$

WHERE N=NO. OF SAMPLES IN RANGECELL

D=OVERLAP FACTOR (=1 FOR NO OVERLAPPING AND ≤ 0 FOR OVERLAPPING)

r=1,2,...,M RANGECELLS FOR TIME AVERAGE

RESTRICTIONS

MINIMUM NO. OF SAMPLES IN RANGEDATA: 1

NOTES

1. The formulae above are given with respect to the way in which the X, Y samples are read from the buffer memory.
2. The Q registers of the APB, APM processors must be defined with the start addresses of the buffer, result memories in the main program.
3. The LCR1 register must be reloaded with the APBRS(15) register and the LCR2 register must be reloaded with the APBRS(14) register in the main program.

START ADDRESS FOR PROGRAM: 1

PROGRAM-MEMORY LOCATIONS USED: 1 - 6

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: POWER PROFILE SUBROUTINE (VERSION 1)

FILE-NAME (NORD 10): PROGØ:DATA

REGISTER NAME	REGISTER ADDRESS	PARAMETER
SAR	4	START ADDRESS OF SUBROUTINE
APB RS(15)	16,15	NO. OF SAMPLES-1 IN RANGECELL
APB RS(14)	16,14	NO. OF RANGECELLS-1 FOR TIME AVERAGE
APB RS(13)	16,13	RANGECELL INCREMENT (=1 FOR NO OVERLAPPING OF RANGECELLS)
APB RS(12)	16,12	SAMPLE INCREMENT (NORMALLY=1)
APM RS(15)	17,15	INCREMENT (=1)

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: POWER PROFILE SUBROUTINE (VERSION 2)

FILE-NAME (NORD 10): PROG1:DATA

PROGRAM DESCRIPTION:

DATA CHANNEL 1 ZERO LAG ESTIMATION $K_r = \sum_{i=0}^{N-1} (X_{i+(N+D-1)(r-1)}^2 + Y_{i+(N+D-1)(r-1)}^2)$

DATA CHANNEL 2 MEAN VALUE ESTIMATION $M_r = \sum_{i=0}^{N-1} (X_{i+(N+D-1)(r-1)} + Y_{i+(N+D-1)(r-1)})$

DATA CHANNEL 1 MEAN VALUE X ESTIMATION $M_{X,r} = \sum_{i=0}^{N-1} X_{i+(N+D-1)(r-1)}$

DATA CHANNEL 2 MEAN VALUE Y ESTIMATION $M_{Y,r} = \sum_{i=0}^{N-1} Y_{i+(N+D-1)(r-1)}$

WHERE N=NO. OF SAMPLES IN RANGECELL

D=OVERLAP FACTOR (=1 FOR NO OVERLAPPING AND ≤0 FOR OVERLAPPING)

r=1,2,...,M RANGECELLS FOR TIME AVERAGE

RESTRICTIONS

MINIMUM NO. OF SAMPLES IN RANGEDATA: 1

NOTES

1. The formulae above are given with respect to the way in which the X, Y samples are read from the buffer memory.
2. The Q registers of the APB, APM processors must be defined with the start addresses of the buffer, result memories in the main program.
3. The LCR1 register must be reloaded with the APBRS(15) register and the LCR2 register must be reloaded with the APBRS(14) register in the main program.
4. In a particular range cell the zero lag estimation K_r and M_r are computed first and the mean value estimation $M_{X,r}$ and $M_{Y,r}$ second.

START ADDRESS FOR PROGRAM: 1

PROGRAM-MEMORY LOCATIONS USED: 1 - 6

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: POWER PROFILE SUBROUTINE (VERSION 2)

FILE-NAME (NORD 10): PROG1:DATA

REGISTER NAME	REGISTER ADDRESS	PARAMETER
SAR	4	START ADDRESS OF SUBROUTINE
APB RS(15)	16,15	NO. OF SAMPLES-1 IN RANGECELL
APB RS(14)	16,14	NO. OF RANGECELLS-1 FOR TIME AVERAGE
APB RS(13)	16,13	RANGECELL INCREMENT (=1 FOR NO OVER-LAPPING OF RANGECELLS)
APB RS(12)	16,12	SAMPLE INCREMENT (NORMALLY=1)
APM RS(15)	17,15	INCREMENT (=1)

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: SINGLE PULSE SUBROUTINE (NO. OF LAGS .EQ. NO. OF SAMPLES)

FILE-NAME (NORD 10): PROG2:DATA

PROGRAM DESCRIPTION:

DATA CHANNEL 1

$$\text{Re}\{K_{L,r}\} = \sum_{i=0}^{N-L-1} (X_{i+(N+D-1)(r-1)} X_{i+L+(N+D-1)(r-1)}^+ Y_{i+(N+D-1)(r-1)} Y_{i+L+(N+D-1)(r-1)}^-)$$

DATA CHANNEL 2

$$\text{Im}\{K_{L,r}\} = \sum_{i=0}^{N-L-1} (X_{i+L+(N+D-1)(r-1)} Y_{i+(N+D-1)(r-1)}^- X_{i+(N+D-1)(r-1)}^+ Y_{i+L+(N+D-1)(r-1)}^-)$$

WHERE N=NO. OF SAMPLES IN RANGECELL

L=0,1,2,...,N-1

D=OVERLAP FACTOR (=1 FOR NO OVERLAPPING AND ≤0 FOR OVERLAPPING)

r=1,2,...,M RANGECELLS FOR TIME AVERAGE

RESTRICTIONS

MINIMUM NO. OF SAMPLES IN RANGEDATA: 1

NOTES

1. The formulae above are given with respect to the way in which the X, Y samples are read from the buffer memory.
2. The Q registers of the APB, APM processors must be defined with the start addresses of the buffer, result memories in the main program.
3. The LCR1 register must be reloaded with the APBRS(15) register and the LCR2 register must be reloaded with the APBRS(14) register in the main program.

START ADDRESS FOR PROGRAM: 1

PROGRAM-MEMORY LOCATIONS USED: 1 - 6

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: SINGLE PULSE SUBROUTINE (NO. OF LAGS .EQ. NO. OF SAMPLES)
FILE-NAME (NORD 10): PROG2:DATA

REGISTER NAME	REGISTER ADDRESS	PARAMETER
SAR	4	START ADDRESS OF SUBROUTINE
APB RS(15)	16,15	NO. OF SAMPLES-1 IN RANGECELL
APB RS(14)	16,14	NO. OF RANGECELL-1 FOR TIME AVERAGE
APB RS(13)	16,13	RANGECELL INCREMENT (=1 FOR NO OVER-LAPPING OF RANGECELLS)
APB RS(12)	16,12	SAMPLE INCREMENT (NORMALLY=1)
APB RS(11)	16,11	TEMPORARY STORAGE
APM RS(15)	17,15	RANGECELL INCREMENT (=NO. OF LAGS COMPUTED)
APM RS(14)	17,14	INCREMENT (=1)
APM RS(13)	17,13	TEMPORARY STORAGE

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: SINGLE PULSE SUBROUTINE (NO. OF LAGS .LE. NO. OF SAMPLES)

FILE-NAME (NORD 10): PROG3:DATA

PROGRAM DESCRIPTION:

DATA CHANNEL 1

$$\text{Re}\{K_{L,r}\} = \sum_{i=0}^{N-L-1} (X_{i+(N+D-1)(r-1)} X_{i+L+(N+D-1)(r-1)} + Y_{i+(N+D-1)(r-1)} Y_{i+L+(N+D-1)(r-1)})$$

DATA CHANNEL 2

$$\text{Im}\{K_{L,r}\} = \sum_{i=0}^{N-L-1} (X_{i+L+(N+D-1)(r-1)} Y_{i+(N+D-1)(r-1)} - X_{i+(N+D-1)(r-1)} Y_{i+L+(N+D-1)(r-1)})$$

WHERE N=NO. OF SAMPLES IN RANGECELL

L=0,1,2,...,P P<=N-1

D=OVERLAP FACTOR (=1 FOR NO OVERLAPPING AND <=0 FOR OVERLAPPING)

r=1,2,...,M RANGECELLS FOR TIME AVERAGE

RESTRICTIONS

MINIMUM NO. OF SAMPLES IN RANGEDATA: 1

MINIMUM NO. OF LAGS IN RANGEDATA: 2

NOTES

1. The formulae above are given with respect to the way in which the X, Y samples are read from the buffer memory.
2. The Q registers of the APB, APM processors must be defined with the start addresses of the buffer, result memories in the main program.
3. The LCR1 register must be reloaded with the APBRS(15) register, the LCR2 register must be reloaded with the APBRS(14) register and the LCR3 register must be reloaded with the the APBRS(13) register in the main program:

START ADDRESS FOR PROGRAM: 1

PROGRAM-MEMORY LOCATIONS USED: 1 - 7

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: SINGLE PULSE SUBROUTINE (NO. OF LAGS .LE. NO. OF SAMPLES)

FILE-NAME (NORD 10): PROG3:DATA

REGISTER NAME	REGISTER ADDRESS	PARAMETER
SAR	4	START ADDRESS OF SUBROUTINE
APB RS(15)	16,15	NO. OF SAMPLES-1 IN RANGECELL
APB RS(14)	16,14	NO. OF RANGECELLS-1 FOR TIME AVERAGE
APB RS(13)	16,13	NO. OF LAGS-1 IN RANGECELL
APB RS(12)	16,12	RANGECELL INCREMENT (=1 FOR NO OVER-LAPPING OF RANGECELLS)
APB RS(11)	16,11	SAMPLE INCREMENT (NORMALLY=1)
APB RS(10)	16,10	TEMPORARY STORAGE
APM RS(15)	17,15	RANGECELL INCREMENT (=NO. OF LAGS COMPUTED)
APM RS(14)	17,14	INCREMENT (=1)
APM RS(13)	17,13	TEMPORARY STORAGE

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: MULTI PULSE SUBROUTINE

FILE-NAME (NORD 10): PROG4:DATA

PROGRAM DESCRIPTION:

DATA CHANNEL 1 $\text{Re}\{K_{L,r}\} = X_{S+r-1}X_{S+L+r-1} + Y_{S+r-1}Y_{S+L+r-1}$

DATA CHANNEL 2 $\text{Im}\{K_{L,r}\} = X_{S+L+r-1}Y_{S+r-1} - X_{S+r-1}Y_{S+L+r-1}$

LET J_0 = POSITION OF 1st PULSE

J_1 = SAMPLE DIFFERENCE BETWEEN 1st AND 2nd PULSE

\vdots

J_{N-1} = SAMPLE DIFFERENCE BETWEEN 1st AND Nth PULSE

WHERE $S = J_0, J_1, J_2, \dots, J_{N-2}$

$L = J_1 - S, J_2 - S, \dots, J_{N-1} - S$ WITH THE RESTRICTION $L > 0$

$r = 1, 2, \dots, M$ RANGE CELLS FOR TIME AVERAGE

RESTRICTIONS

MINIMUM NO. OF ELEMENT PULSES IN PULSE GROUP: 2

MAXIMUM NO. OF ELEMENT PULSES IN PULSE GROUP: 13

NOTES

1. The formulae above are given with respect to the way in which the X, Y samples are read from the buffer memory.
2. The Q registers of the APB, APM processors must be defined with the start addresses of the buffer, result memories in the main program.
3. The LCR1 register must be reloaded with the APBRS(15) register and the LCR2 register must be reloaded with the APBRS(14) register in the main program.
4. The zero lag is not computed in this algorithm, therefore to calculate the number of lags computed use the formula: $N(N-1)/2$ where N is the number of element pulses in the pulse group.

START ADDRESS FOR PROGRAM: 1

PROGRAM-MEMORY LOCATIONS USED: 1 - 6

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: MULTI PULSE SUBROUTINE

FILE-NAME (NORD 10): PROG4:DATA

REGISTER NAME	REGISTER ADDRESS	PARAMETER
SAR	4	START ADDRESS OF SUBROUTINE
APB RS(15)	16,15	NO. OF PULSES-1 IN RANGECELL
APB RS(14)	16,14	NO. OF RANGECELLS-1 FOR TIME AVERAGE
APB RS(13)	16,13	RANGECELL INCREMENT (NORMALLY=1)
APB RS(12)	16,12	TEMPORARY STORAGE
⋮	⋮	⋮
APB RS(1)	16,1	SAMPLE DISTANCE BETWEEN 2nd LAST AND 1st PULSE
APB RS(0)	16,0	SAMPLE DISTANCE BETWEEN LAST AND 1st PULSE
APM RS(15)	17,15	INCREMENT (=1)

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: CROSS CORRELATION SUBROUTINE (NO. OF LAGS .EQ. NO. OF SAMPLES)

FILE-NAME (NORD 10): PROG5:DATA

PROGRAM DESCRIPTION:

DATA CHANNEL 1

$$\text{Re}\{K_{L,r}\} = \sum_{i=0}^{N-L-1} (X_{i+(N+D-1)(r-1)} X_{i+S+L+(N+D-1)(r-1)} + Y_{i+(N+D-1)(r-1)} Y_{i+S+L+(N+D-1)(r-1)})$$

DATA CHANNEL 2

$$\text{Im}\{K_{L,r}\} = \sum_{i=0}^{N-L-1} (X_{i+S+L+(N+D-1)(r-1)} Y_{i+(N+D-1)(r-1)} - X_{i+(N+D-1)(r-1)} Y_{i+S+L+(N+D-1)(r-1)})$$

WHERE N=NO. OF SAMPLES IN RANGECELL

L=0,1,2,...,N-1

S=SAMPLE DIFFERENCE BETWEEN THE TWO SETS OF DATA

D=OVERLAP FACTOR (=1 FOR NO OVERLAPPING AND ≤0 FOR OVERLAPPING)

r=1,2,...,M RANGECELLS FOR TIME AVERAGE

RESTRICTIONS

MINIMUM NO. OF SAMPLES IN RANGEDATA: 1

NOTES

1. The formulae above are given with respect to the way in which the X, Y samples are read from the buffer memory.
2. The Q registers of the APB, APM processors must be defined with the start addresses of the buffer, result memories in the main program.
3. The LCR1 register must be reloaded with the APBRS(15) register and the LCR2 register must be reloaded with the APBRS(14) register in the main program.
4. Only half of the correlation function can be obtained with this scheme.
5. The Single Pulse autocorrelation scheme (see PROG2:DATA) can be obtained by setting APBRS(12)=0.

START ADDRESS FOR PROGRAM: 1

PROGRAM-MEMORY LOCATIONS USED: 1 - 3

MICRO-PROGRAM FOR DIGITAL CORRELATOR

AUTOR: TERRANCE HO

DATE: 6/8/80

PROGRAM NAME: CROSS CORRELATION SUBROUTINE (NO. OF LAGS .EQ. NO. OF SAMPLES)

FILE-NAME (NORD 10): PROG5:DATA

REGISTER NAME	REGISTER ADDRESS	PARAMETER
SAR	4	START ADDRESS OF SUBROUTINE
APB RS(15)	16,15	NO. OF SAMPLES-1 IN RANGECELL
APB RS(14)	16,14	NO. OF RANGECELLS-1 FOR TIME AVERAGE
APB RS(13)	16,13	RANGECELL INCREMENT (=1 FOR NO OVER-LAPPING OF RANGECELLS)
APB RS(12)	16,12	START ADDRESS OF 2nd FIELD-START ADDRESS OF 1st FIELD
APB RS(11)	16,11	SAMPLE INCREMENT (NORMALLY=1)
APB RS(10)	16,10	TEMPORARY STORAGE
APM RS(15)	17,15	RANGECELL INCREMENT (=NO. OF LAGS COMPUTED)
APM RS(14)	17,14	INCREMENT (=1)
APM RS(13)	17,13	TEMPORARY STORAGE

EISCAT publications

F. du Castel, O. Holt, B. Hultqvist, H. Kohl and M. Tiuri:

A European Incoherent Scatter Facility in the Auroral Zone (EISCAT).

A Feasibility Study ("The Green Report") June 1971. (Out of print).

O. Bratteng and A. Haug:

Model Ionosphere at High Latitude, EISCAT Feasibility Study, Report No. 9.

The Auroral Observatory, Tromsø July 1971. (Out of print).

A European Incoherent Scatter Facility in the Auroral Zone, UHF System and Organization ("The Yellow Report"), June 1974.

EISCAT Annual Report 1976. (Out of print).

P.S. Kildal and T. Hagfors:

Balance between investment in reflector and feed in the VHF cylindrical antenna.

EISCAT Technical Notes No. 77/1, 1977.

T. Hagfors:

Least mean square fitting of data to physical models.

EISCAT Technical Notes No. 78/2, 1978.

T. Hagfors:

The effect of ice on an antenna reflector.

EISCAT Technical Notes No. 78/3, 1978.

T. Hagfors:

The bandwidth of a linear phased array with stepped delay corrections.

EISCAT Technical Notes No. 78/4, 1978.

Data Group meeting in Kiruna, Sweden, 18-20 Jan. 1978

EISCAT Meetings No. 78/1, 1978

EISCAT Annual Report 1977

H-J. Alker:

Measurement principles in the EISCAT system

EISCAT Technical Notes No. 78/5, 1978

EISCAT Data Group meeting in Tromsö, Norway 30-31 May, 1978

EISCAT Meetings No. 78/2, 1978.

P-S. Kildal:

Discrete phase steering by permuting precut phase cables.

EISCAT Technical Notes No. 78/6, 1978

EISCAT UHF antenna acceptance test.

EISCAT Technical Notes No. 78/7, 1978.

P-S. Kildal:

Feeder elements for the EISCAT VHF parabolic cylinder antenna.

EISCAT Technical Notes No. 78/8, 1978.

H-J. Alker:

Program CORRSIM: System for program development and software simulation of EISCAT digital correlator, User's Manual.

EISCAT Technical Notes No. 79/9, 1979.

H-J. Alker:

Instruction manual for EISCAT digital correlator.

EISCAT Technical Notes No. 79/10, 1979

H-J. Alker:

A programmable correlator module for the EISCAT radar system.

EISCAT Technical Notes No. 79/11, 1979.

T. Ho and H-J. Alker:

Scientific programming of the EISCAT digital correlator.

EISCAT Technical Notes No. 79/12, 1979.

S. Westerlund (editor):

Proceedings EISCAT Annual Review Meeting 1969. Part I and II,
Abisko, Sweden, 12-16 March 1979.
EISCAT Meetings No. 79/3, 1979.

J. Murdin:

EISCAT UHF Geometry.
EISCAT Technical Notes No. 79/13, 1979.

T. Hagfors:

Transmitter Polarization Control in the EISCAT UHF System.
EISCAT Technical Notes No. 79/14, 1979.

B. Törustad:

A description of the assembly language for the EISCAT digital
correlator.
EISCAT Technical Notes No. 79/15, 1979.

J. Murdin:

Errors in incoherent scatter radar measurements.
EISCAT Technical Notes No. 79/16, 1979.

EISCAT Digital Correlator. TEST MANUAL.
EISCAT Technical Notes No. 79/17, 1979.

G. Lejeune:

A program library for incoherent scatter calculation.
EISCAT Technical Notes No. 79/18, 1979.

K. Folkestad:

Lectures for EISCAT Personnel, Volume I
EISCAT Technical Notes No. 79/19, 1979.

Svein A. Kvalvik:

Correlator Buffer-Memory for the EISCAT Radar system
EISCAT Technical Notes. No. 80/20.

P.-S. Kildal:

EISCAT VHF Antenna Tests

EISCAT Technical Notes No. 80/21

J. Armstrong:

EISCAT Experiment Preparation Manual

EISCAT Technical Notes No. 80/22

A. Farmer:

EISCAT Data Gathering and Dissemination

EISCAT Technical Note 80/23

Terrance Ho and Hans-Jørgen Alker:

Scientific Programming of the EISCAT Digital Correlator (Revised)

EISCAT Technical Note 81/24

Terrance Ho:

Programs Corrsim, Corrtest: System for Program Development and
Software Simulation of EISCAT Digital Correlator. User's manual.

EISCAT Technical Note 81/25

Terrance Ho:

Instruction Manual for EISCAT Digital Correlator (Revised).

EISCAT Technical Note 81/26

Terrance Ho:

Standard Subroutines and Programs for EISCAT Digital Correlator.

EISCAT Technical Note 81/27

Terrance Ho:

Pocket Manual for Programming the EISCAT Digital Correlator.

EISCAT Technical Note 81/28

K. Folkestad:

Lectures for EISCAT Personnel, Volume II.

EISCAT Technical Note 81/29

