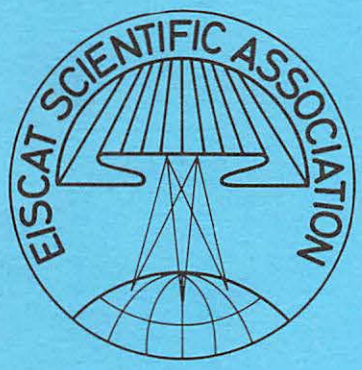
79/10



EISCAT
TECHNICAL
NOTES

INSTRUCTION MANUAL
FOR
EISCAT DIGITAL CORRELATOR

by
Hans-Jørgen Alker

KIRUNA
Sweden

INSTRUCTION MANUAL
FOR
EISCAT DIGITAL CORRELATOR

by

Hans-Jørgen Alker

EISCAT Scientific Association
S-981 01 Kiruna, Sweden
March 1979

This report has also been published by the Institute of Mathematical and Physical Sciences, University of Tromsö, September 1978.

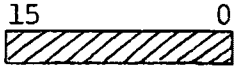
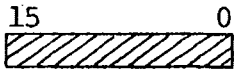
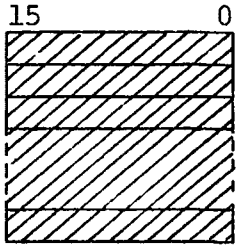
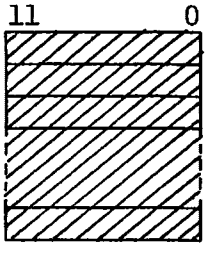
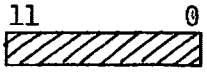
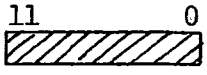
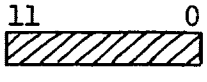
EISCAT Technical Note 79/10
Printed in Sweden
EISCAT Scientific Association
March 1979

PROGRAMMABLE REGISTERS/MEMORY-LOCATIONS

DATA-FIELD IN DIGITAL CORRELATOR

All numeric values are octal.

ABBR.: APB: Address processor for buffer-memory, APM: Address processor for result-memory, C: Computer, F: Front-panel, P: Internal program.

REGISTER NAME		ADDRESS	NUMERIC-RANGE	LOAD-ACCESS
STATUSWORD (STAT)		1,	0-177777	C,F
START-ADDRESS-REGISTER (SAR)		4,	0-77	C,F,P
BASE-ADDRESS-REGISTER, APB (BAR)		5,	0-177777	C,F
DATA I-REGISTER, APB		6,	0-177777	C,F,P
REGISTER-STACK, APB* (APBRS)		20,0, 20,1, 20,2, ⋮ 20,17,	0-177777 " " "	C,F " " "
REGISTER-STACK, APM (APMRS)		21,0, 21,1, 21,2, ⋮ 21,17,	0-7777 " " "	C,F " " "
LOAD-REGISTER FOR LOOP-COUNTER 1 (LCR1)		22,	0-7777	C,F,P
TEMPORARY STORAGE FOR VALUE OF LOOP-COUNTER 1 (LCR1A)		-	0-7777	P
LOAD-REGISTER FOR LOOP-COUNTER 2 (LCR2)		23,	0-7777	C,F,P
LOAD-REGISTER FOR LOOP-COUNTER 3 (LCR3)		24,	0-7777	C,F,P
"CORRELATOR-READY"-REGISTER (CRA)		77,	0-1	C,F

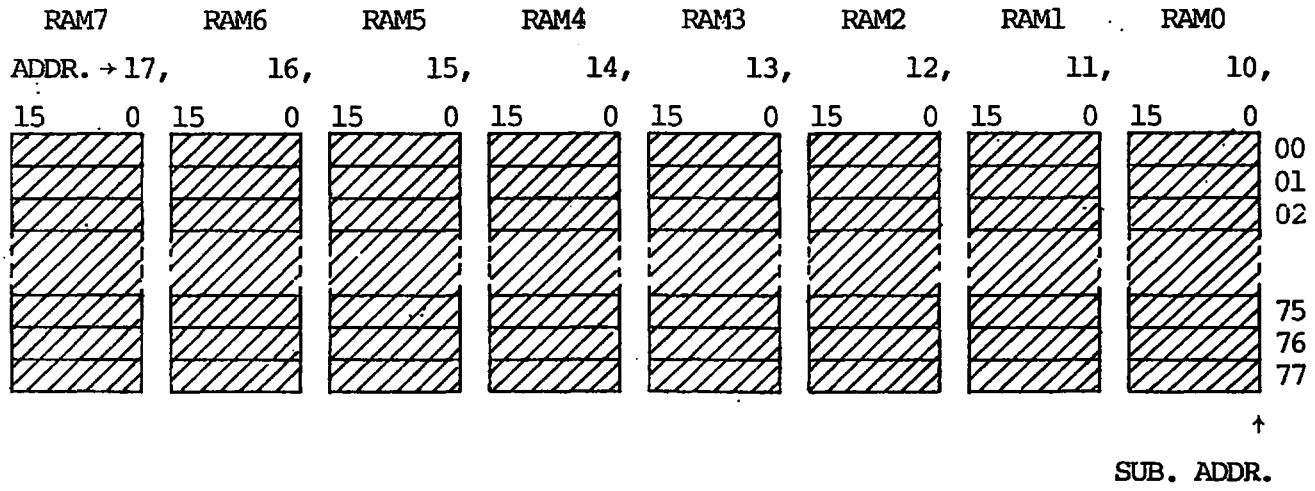
* APBRS must be loaded through the Data I-reg., APB.

PROGRAM-FIELD IN DIGITAL CORRELATOR

All numeric values are octal.

ABBR.: RAM: Random access memory.

PROGRAM-MEMORY STRUCTURE:



When accessed from external source (computer or front-panel) for program set-up, the memory is split into 8 separate pages (RAM0, RAM1,...) with separate address identification. Each location in page is given by the SUB-ADDRESS.

The correlator internal instruction word (128 bits) is defined by parallel read-out from the same SUB.ADDR. in all pages, i.e. Instruction word location is defined by the SUB.ADDR.

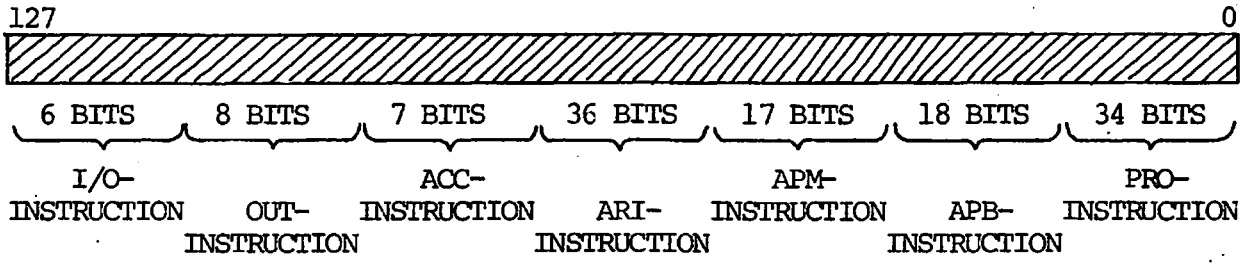
PROGRAM RESTRICTIONS: Location 00 is used for idle-running of the correlator.
Location 77 is used for storage of service-routine.
Pointer when handling external interrupts.

LOCATIONS FREE FOR PROGRAMMING: 01-76.

CORRELATOR INSTRUCTION WORD

The correlator is a multi-processor system where 7 main internal functions are controlled in parallel, the control given by the 128-bits instruction word.

THE INSTRUCTION WORD SEPARATED INTO MAIN INTERNAL FUNCTIONS:



PRO-INSTRUCTION: Controls the execution of the correlator program and re-load of programmable registers in the data-field.

APB-INSTRUCTION: Controls the addressing of the buffer-memory (data input source) and generation of re-load values to the registers in data-field.

APM-INSTRUCTION: Controls the addressing of the result-memory (data storage).

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 computer.

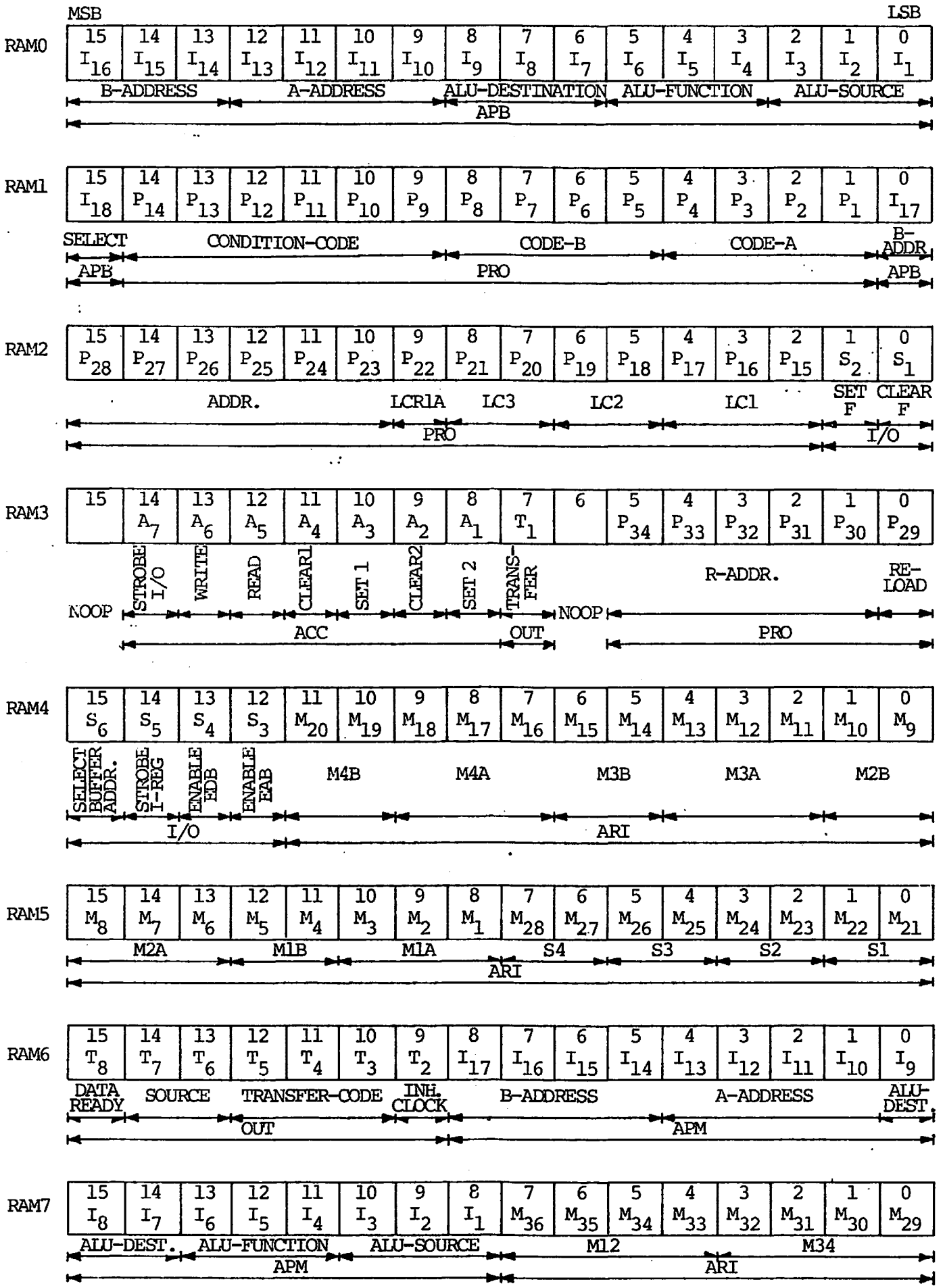
I/O-INSTRUCTION: Controls data/address communication in a multi-correlator system.

The actual bit-assignments for the functions are given on the next page. Due to hardware construction the different functions are scrambled.

CORRELATOR INSTRUCTION WORD

ALLOCATION OF BITS IN MEMORY.

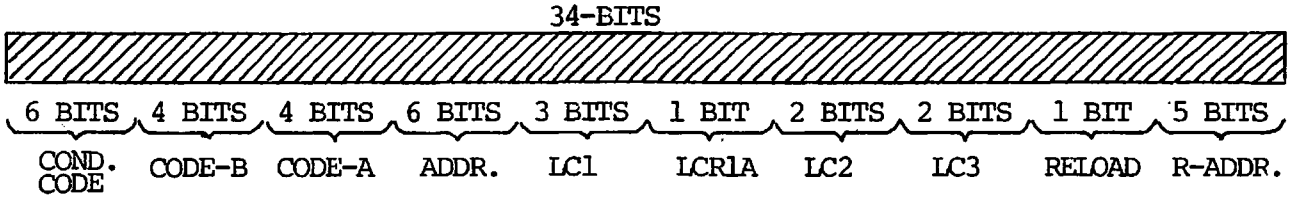
INSTRUCTION-WORD BIT ASSIGNMENTS:



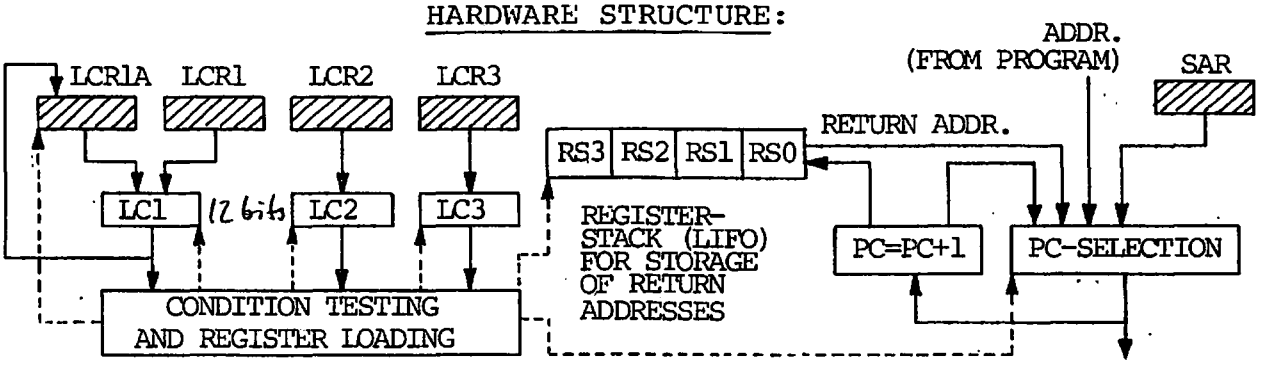
PROGRAM-INSTRUCTIONS

ABBR.: 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

All loop-counters have 12 bits with
data-range: $0\text{-}7777_8$.

COND.CODE, INSTRUCTION BITS $P_{14}\ P_{13}\ P_{12}\ P_{11}\ P_{10}\ P_9$

TEST-STRUCTURE 1. $P_{14} = 1$: If test is true use CODE-B, else use CODE-A.

BINARY CODING: Ø = Free value	P_{13}	P_{12}	P_{11}	P_{10}	P_9		TEST CONDITION
	Ø	Ø	0	0	0		Use CODE-A unconditionally
	1	1	0	0	1	(71)	If LC1 = 0
	1	1	0	1	0	(72)	If LC2 = 0
	1	1	0	1	1	(73)	If LC1 or LC2 = 0
	1	1	1	0	0	(74)	If LC3 = 0
	1	1	1	0	1	(75)	If LC1 or LC3 = 0
	1	1	1	1	0	(76)	If LC2 or LC3 = 0
	1	1	1	1	1	(77)	If LC1 or LC2 or LC3 = 0
	1	0	Ø	Ø	Ø	(6Ø)	Same as before with LC2 ≠ 0
	0	1	Ø	Ø	Ø	(5Ø)	Same " " " LC3 ≠ 0
	0	0	Ø	Ø	Ø	(4Ø)	Same " " " LC2 and LC3 ≠ 0

TEST-STRUCTURE 2. $P_{14} = 0$: If test 1 is true use CODE-B, else if test 2 is true use CODE-A, else continue (PC = PC+1).

BINARY CODING:	P_{13}	P_{12}	P_{11}	P_{10}	P_9		TEST CONDITION
	1	1	0	1	1	(33)	Test 1: LC1 = 0, Test 2: LC2 = 0
	1	1	1	1	0	(36)	Test 1: LC3 = 0, Test 2: LC2 = 0
	1	1	1	0	1	(35)	Test 1: LC1 or LC3 = 0, no test 2
	1	1	1	1	1	(37)	Test 1: LC1 or LC3 = 0, Test 2: LC2 = 0
	1	0	Ø	Ø	Ø	(2Ø)	Test 2: ≠ 0
	0	1	Ø	Ø	Ø	(1Ø)	Test 1: ≠ 0
	0	0	Ø	Ø	Ø	(0Ø)	Test 1 and test 2: ≠ 0

CODE-A/ CODE-B	OCTAL VALUE	FUNCTION	COMMENTS
	0	PC = PC+1, POP STACK	CONTINUE AND DELETE LAST RETURN ADDR.
	1	PC = RETURN ADDR., POP STACK	RETURN AND DELETE LAST RETURN ADDR.
	2	PC = ADDR., POP STACK	JUMP AND DELETE LAST RETURN ADDR.
	3	PC = SAR, POP STACK	GOTO SAR AND DELETE LAST RETURN ADDR.
	4,14	PC = PC+1	CONTINUE
	5,15	PC = RETURN ADDR.	RETURN
	6,16	PC = ADDR.	JUMP
	7,17	PC = SAR	GOTO SAR
	10	PC = PC+1, PUSH STACK	CONTINUE AND SET RETURN ADDR.
	11	PC = RETURN ADDR., PUSH STACK	RETURN AND SET RETURN ADDR.
	12	PC = ADDR., PUSH STACK	JUMP AND SET RETURN ADDR.
	13	PC = SAR, PUSH STACK	GOTO SAR AND SET RETURN ADDR.
LC1:	0	NOOP	
	1	LC1 = LC1-1	DECREMENT COUNTER
	2	LC1 = LCR1	LOAD
	3	LC1 = LCR1A	LOAD
	4	IF LCL = 0: LC1 = LCR1, LC2 = LC2-1, ELSE: LC1 = LC1-1	CONDITIONAL LOAD/DECREMENT
	5	IF LC1 = 0 and LC3 = 0: LC1 = LCR1A, ELSE: LC1 = LC1-1	CONDITIONAL LOAD/DECREMENT
	6	IF LC1 = 0: LC1 = LCR1, ELSE: LC1 = LC1-1	CONDITIONAL LOAD/DECREMENT
	7	IF LC1 = 0: LC1 = LCR1A, ELSE: LC1 = LC1-1	CONDITIONAL LOAD/DECREMENT
LC2:	0	NOOP	
	1	LC2 = LC2-1	
	3	LC2 = LCR2	
LC3:	0	NOOP	
	1	LC3 = LC3-1	
	2	LC3 = LCR3	
	3	IF LC3 = 0: LC3 = LCR3, ELSE: LC3 = LC3-1	
LCR1A:	0	NOOP	
	1	LCR1A = LC1	REGISTER-LOAD
RELOAD:	0	NOOP	
	1	REGISTER-RELOAD	(RELOAD VALUE FROM APB)
R-ADDR.:	4	RELOAD SAR	IF RELOAD = 1
	5	" BAR, APB	" "
	22	" LCR1	" "
	23	" LCR2	" "
	24	" LCR3	" "

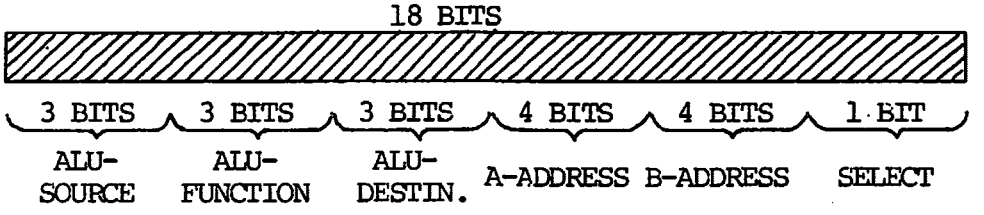
PROGRAMMING RESTRICTIONS: Only the given instruction values should be used. Notice the limitation in storing 4 return addresses. When push-stack option is used (set up return address), the stored value is the present PC-value plus one.

Program-location 0 can not be used for register-reload.

APB/APM-INSTRUCTIONS

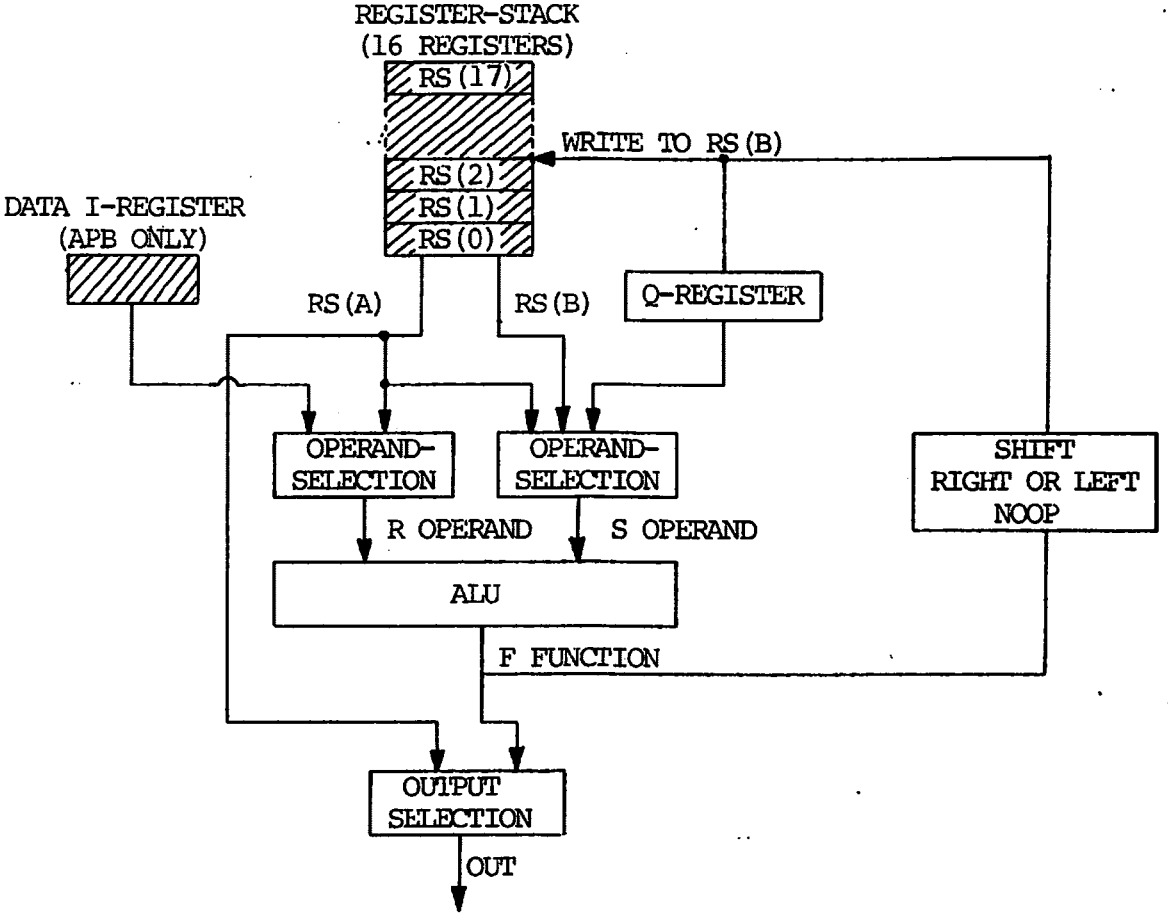
ABBR.: APB: Address processor for buffer-memory, APM: Address processor for result-memory, ALU: Arithmetical-logical unit.

INSTRUCTION-WORD SEPARATION:



APM has not the select sub-instruction.

HARDWARE STRUCTURE:



For APB: Address-bus to buffer-memory or relaod-value to registers.
For APM: Read/write-address to result-memory.

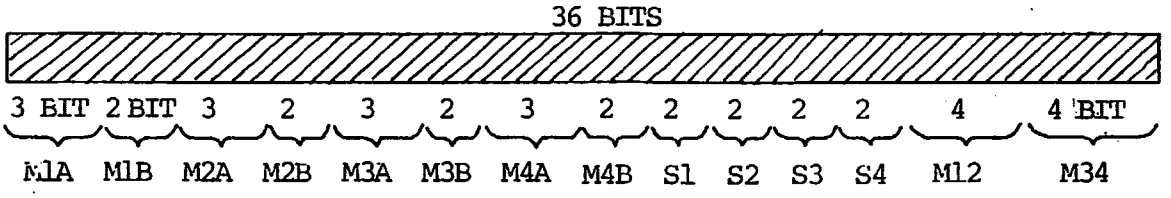
Numeric range for APB: $0\text{-}177777_8$ (16-bits)
Numeric range for APM: $0\text{-}7777_8$ (12-bits)

	OCTAL CODING	FUNCTION
ALU-SOURCE:	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 = DATA I, S = RS(A)
	6	R = DATA I, S = Q
	7	R = DATA I, S = 0
ALU-FUNCTION:	0	F = R+S
	1	F = S-R
	2	F = R-S
	3	F = R OR S
	4	F = R AND S
	5	F = $\overline{R}$ AND S
	6	F = R EXOR S
	7	F = R EXNOR S
ALU-DESTINATION:	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:	0-17	A-ADDRESS IS ALWAYS A READ ADDRESS B-ADDRESS IS READ AND WRITE ADDRESS (ALSO SAME REGISTER-STACK LOCATION CAN BE MODIFIED)
SELECT:	0	NOOP
	1	WHENEVER RS(B) IS REFERENCED, B-ADDRESS IS INHIBITED AND ONE GETS RS(LC1) LC1: LOOP-COUNTER 1

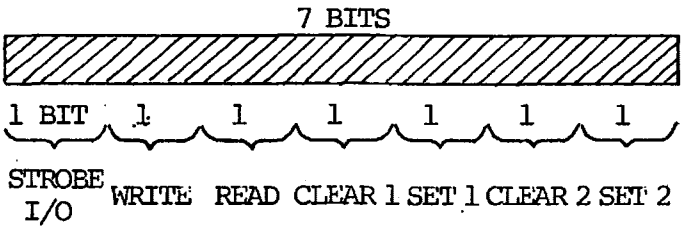
ARI/ACC-INSTRUCTIONS

ABBR.: 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.

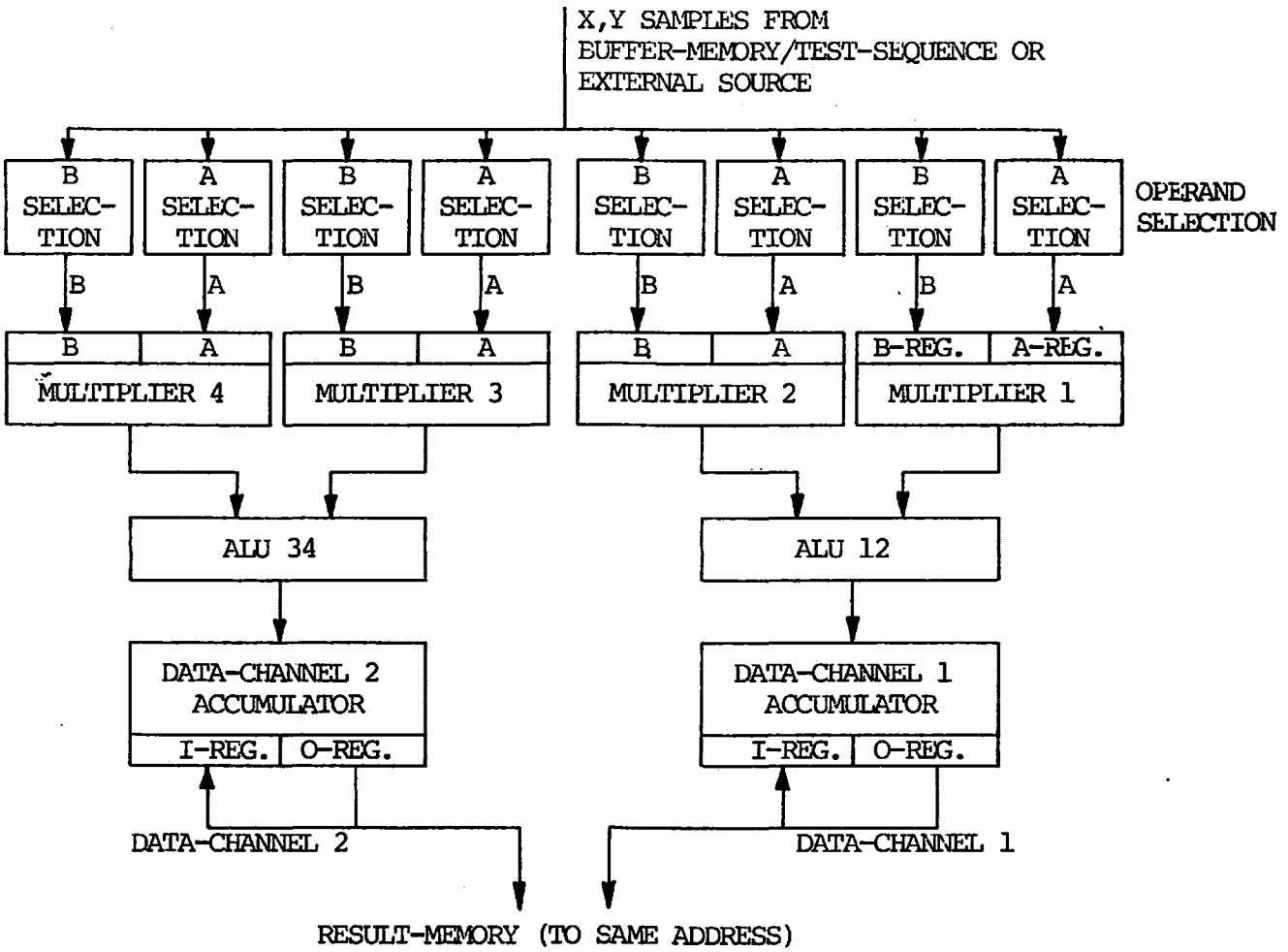
ARI INSTRUCTION-WORD SEPARATION:



ACC INSTRUCTION-WORD SEPARATION:



HARDWARE STRUCTURE:



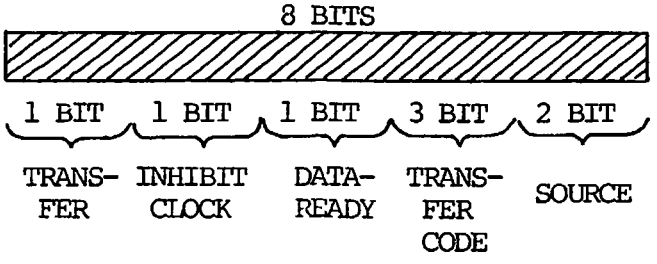
	OCTAL CODING	FUNCTION	COMMENTS
M1A:	0	$A = X_{INTERNAL}$	SAME FOR ALL MULTIPLIERS
	1	$A = Y_{INTERNAL}$	
	2	$A = X_{EXTERNAL}$	
	3	$A = Y_{EXTERNAL}$	
	4-7	A = 1	
M1B:	0	$B = X_{INTERNAL}$	SAME FOR ALL MULTIPLIERS
	1	$B = Y_{INTERNAL}$	
	2	$B = X_{EXTERNAL}$	
	3	$B = Y_{EXTERNAL}$	
S1:	0	NOOP	SAME FOR ALL MULTIPLIERS
	1	LOAD NEW OPERAND INTO A-REG.	
	2	" " " " B-REG.	
	3	LOAD NEW OPERANDS INTO A- AND B-REG.	
M12:	5	ALU 12 = M_2	SAME FOR ALU 34
	6	ALU 12 = M_1-M_2	
	11	ALU 12 = M_1+M_2	
	14	ALU 12 = -1	
	17	ALU 12 = M_1	
STROBE I/O:	0	NOOP	DATA-CHANNEL 1 AND 2 ACCUM.
	1	STROBE NEW CONTENT INTO I/O-REG.	ARE OPERATED IN PARALLEL
WRITE:	0	NOOP	
	1	WRITE O-REG. TO RESULT-MEMORY	
READ:	0	NOOP	
	1	READ RES.-MEM. TO I-REG. OR SET I-REG. = 0	DEPENDENT ON VALUE OF SET 1 and SET 2
SET 1:	0 —	IF READ = 1 and SET 2 = 0: RESET I-REG.	SET 1 IS ACTIVE UNTIL CLEAR 1 = 1
	1	IF READ = 1: GENERATE READ	
CLEAR 1:	0	NOOP	
	1	RESET: SET 1 = 0	
SET 2:	0 —	ENABLE SET 1-CONTROL OF READ	
	1	INHIBIT SET1-CONTROL OF READ	SET 2 IS ACTIVE UNTIL CLEAR 2 = 1. SET 2 CAN ALSO BE SET FROM STATUS-WORD
CLEAR 2:	0	NOOP	
	1	RESET: SET 2 = 0	

PROGRAMMING RESTRICTIONS: Note that it is possible to select data-read, accumulate and data-write to the same res.-mem. in one instruction cycle. It is not allowed to have read and/or write operation to the same memory-location in following instruction cycles.

OUT-INSTRUCTIONS

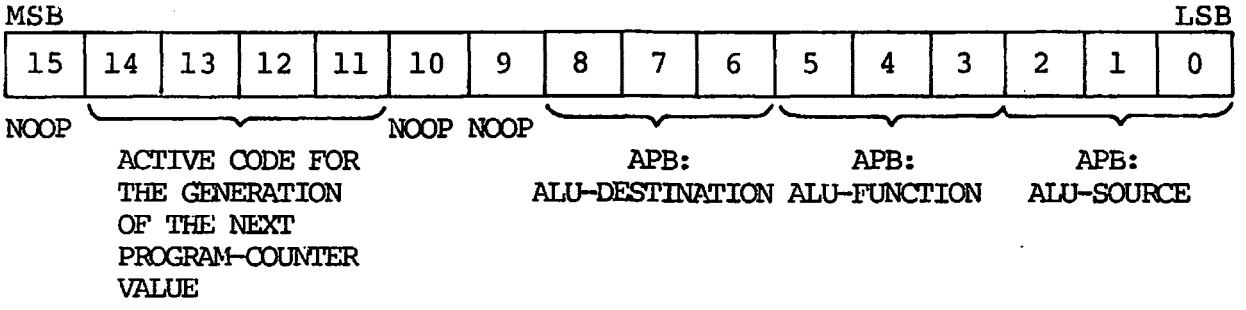
ABBR.: DMA: Direct-memory-access, LS: Least significant part,
MS: Most significant part.

INSTRUCTION-WORD SEPARATION:

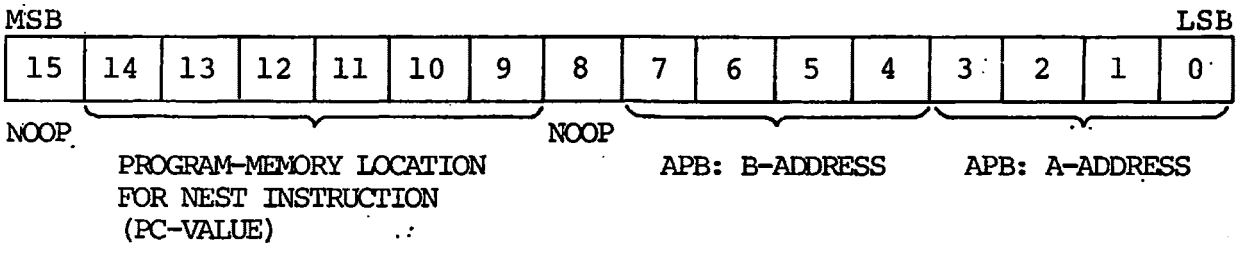


	OCTAL CODING	FUNCTION	COMMENT
TRANSFER:	0	NOOP	TRANSFER NOT SELECTED
	1	ENABLE CORRELATOR FOR TRANSFER	MUST BE SET FOR TRANSFER
INHIBIT CLOCK:	0	ENABLE INTERNAL CLOCK-OPERATION	
	1	INHIBIT CLOCK AND PROGRAM IS STOPPED	REQUIRES ALWAYS THAT DATA-READY IS SET
DATA-READY:	0	NOOP	
	1	SET "DATA-READY" TO CAMAC	"DATA-RECEIVED" FROM CAMAC ADVANCES THE CORRELATOR PROGRAM ONE INSTRUCTION-CYCLE
TRANSFER-CODE:	0	STATUS-WORD	CONTENT ACTIVE ON DMA-CHANNEL AND FRONT-PANEL DISPLAY
	1	CONTROL-WORD	
	2	RES.MEM, DATA-CHANNEL 1: LS	DATA-SOURCE GIVEN BY SOURCE-INSTR.
	3	RES.MEM, DATA-CHANNEL 1: MS	" " " "
	4	RES.MEM, DATA-CHANNEL 2: LS	" " " "
	5	RES.MEM, DATA-CHANNEL 2: MS	" " " "
	6	TESTWORD 1	
	7	TESTWORD 2	
SOURCE:	0	MEM. VALUES FROM MASTER-MODULE	
	1	" " " SLAVE 1	MUST BE USED ONLY FOR TRANSFER-CODES: 2-5
	2	" " " SLAVE 2	
	3	" " " SLAVE 3	

TEST-WORD 1 STRUCTURE:



TEST-WORD 2 STRUCTURE:



PROGRAMMING RESTRICTIONS:

It is recommended that all transfer-programs start at program-memory location 40. This value corresponds to the entry-point automatically loaded when the real-time command "data-transfer" is received from the radar-controller.

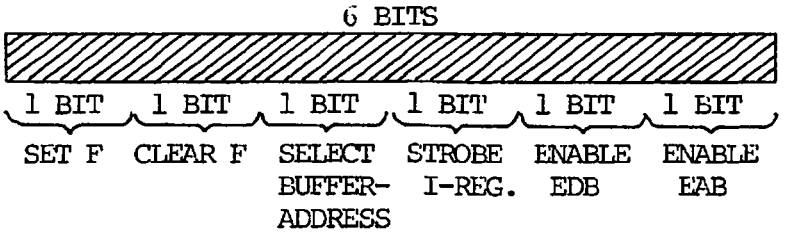
The hardware construction requires that before transfer is set to 0 (end of transfer-program), 6 dummy instructions without inhibit-clock must be inserted.

When PC = 0 out-instruction word is set to zero.

I/O-INSTRUCTIONS

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

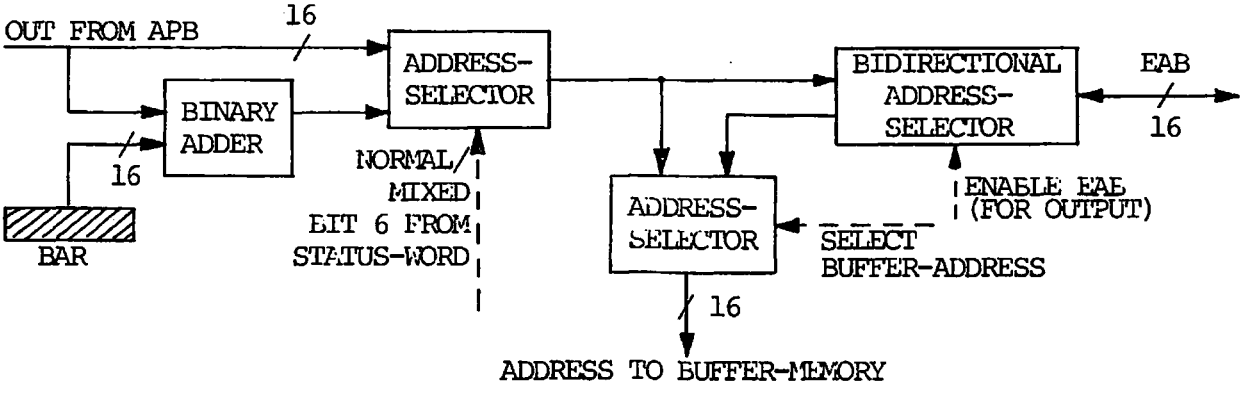
INSTRUCTION-WORD SEPARATION:



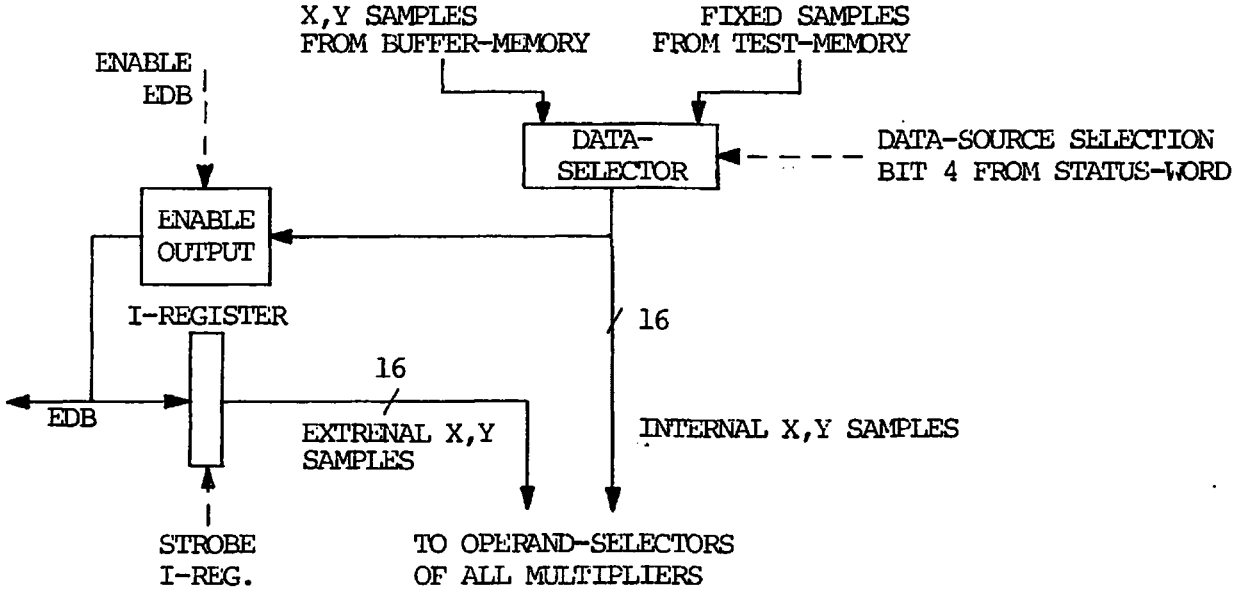
I/O-COMMUNICATION STRUCTURE:

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

BUFFER-MEMORY ADDRESS GENERATION:



DATA-INPUT STRUCTURE:

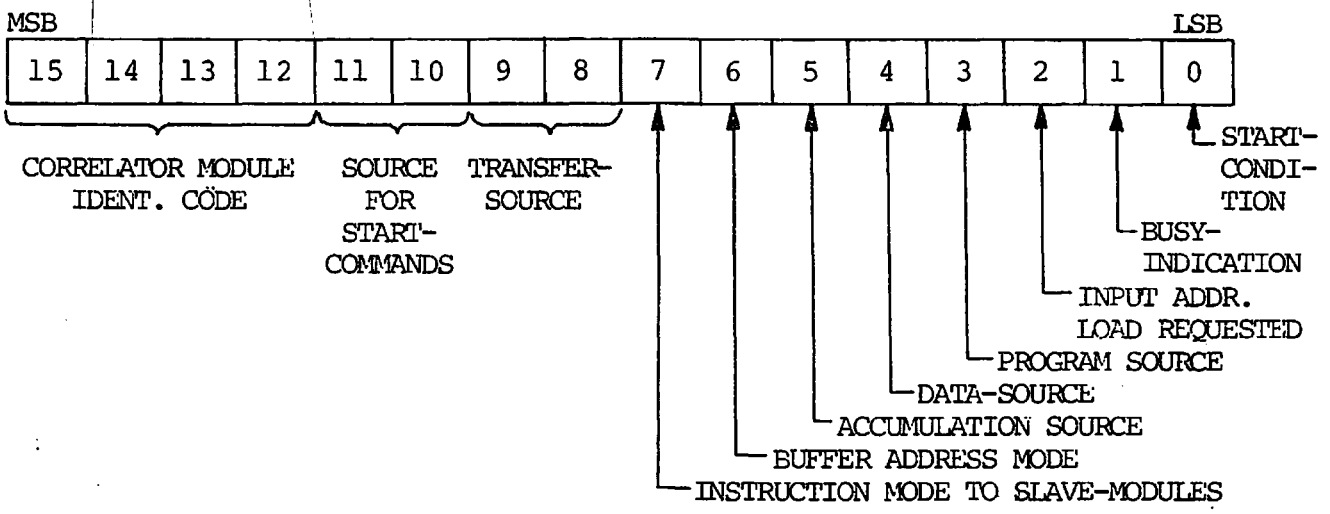


SET F:	OCTAL CODING	FUNCTION	COMMENT
	0	NOOP	
	1	SET FLAG	SIGNAL AT EXTERNAL CONNECTOR
CLEAR F:	0	NOOP	
	1	CLEAR FLAG	
SELECT BUFFER-ADDRESS:	0	BUFFER-ADDRESS GENERATED BY INTERNAL HARDWARE	
	1	BUFFER-ADDRESS GENERATED FROM EAB	
STROBE I-REG.:	0	NOOP	
	1	STROBE	
ENABLE EDB:	0	NOOP	
	1	INTERNAL X,Y SAMPLES ACTIVE ON EDB	
ENABLE EAB:	0	NOOP	
	1	INTERNAL ADDRESS ACTIVE ON EAB	

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

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

STATUS-WORD FUNCTIONS:

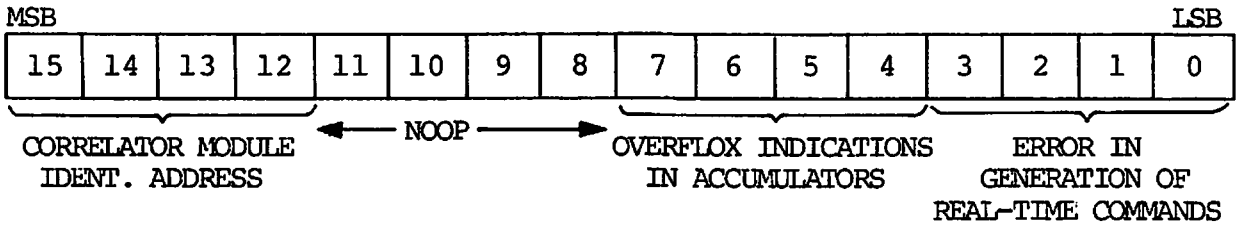


(P) denotes that function can be set by user.

BIT		FUNCTION	COMMENT
15-12:		CORRELATOR MODULE IDENT. CODE ($0\text{-}17_8$)	SET BY INTERNAL SWITCH, DESIMAL VALUE SHOWN ON FRONT-PANEL
11-10:	(P)	00: FRONT-PANEL 01: RADAR-CONTROLLER 10: COMPUTER	ENABLING THE GIVEN SOURCE FOR STARTING THE CORRELATOR
9-8:		00: MASTER-MODULE 01: SLAVE1 10: SLAVE2 11: SLAVE3	SET BY TRANSFER INSTRUCTION
7:	(P)	0 : SAME ARI-INSTRUCTIONS TO SLAVES 1 : MODIFIED OPERAND-SELECTIONS TO SLAVE-MULTIPLIERS	SWITCHES INTERNAL DATA → EXTERNAL DATA
6:	(P)	0 : NORMAL ADDRESS MODE 1 : MIXED ADDRESS MODE	SEE I/O-INSTRUCTIONS
5:	(P)	0 : START EXPERIMENT 1 : CONTINUE EXPERIMENT	IF SET1 = 1, SEE ACC.-INSTRUCTIONS GENERATE SET2 = 1, SEE ACC.-INSTR.
4:	(P)	0 : INTERNAL DATA FROM BUFFER-MEMORY 1 : INTERNAL DATA FROM TEST-MEMORY	SEE I/O-INSTRUCTIONS
3:	(P)	0 : PROGRAMMABLE MEMORY 1 : MEMORY FOR FIXED INTERNAL PROGRAMS	RAM-MODULES SELECTION OF LIBRARY-PROGRAMS WITH SAR
2:		0 : READY FOR LOADING REGISTERS/ PROGR. 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 77_8

All main functions are shown on front-panel.

CONTROL-WORD FUNCTIONS:



Given function is active when bit is set:

BIT	FUNCTION
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 SLAVE3
5:	ACCUMULATOR OVERFLOW IN SLAVE2
6:	ACCUMULATOR OVERFLOW IN SLAVE3
7:	ACCUMULATOR OVERFLOW IN MASTER-MODULE

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

Control-word is shown on the front-panel.

REAL-TIME COMMANDS

All commands from computer or radar-controller are termed real-time commands. These commands have the following internal correlator response when the correlator is in remote-status (switch setting on front-panel).

REAL-TIME COMMAND	SOURCE	INTERNAL RESPONSE
START COMPUTE:	COMPUTER OR RADAR-CONTROLLER	TEST ON CORRELATOR READY AND BUSY AND ENABLING OF START-SOURCE. IF NOT GRANTED: SET ERROR BIT. IF GRANTED: START INTERNAL PROGRAM AT ENTRY-POINT: PC = SAR SET BIT 1 IN STATUSWORD.
START TRANSFER:	RADAR-CONTROLLER	SAME AS FOR START COMPUTE. ENTRY-POINT AT PC = 40_8
RESET:	COMPUTER	SET PC = 00, RESET ERROR BITS IN CONTROL-WORD, RESET CORRELATOR READY AND BIT 1 IN STATUSWORD.
ADDRESS LOAD:	COMPUTER	TEST ON CORRELATOR BUSY. IF NOT GRANTED: SET ERROR BIT. IF GRANTED: LOAD ADDRESS FOR DATA-LOAD TO DATA- OR PROGRAM-FIELD. SET BIT 2 IN STATUSWORD.
DATA LOAD:	COMPUTER	LOADS DATA WITH ADDRESS REFERENCE GIVEN ABOVE. RESET BIT 2 IN STATUSWORD.
INTERRUPT PROGRAM:	EXTERNAL SOURCE	WHEN FIRST CONTINUE-STATEMENT IS REACHED IN THE INTERNAL PROGRAM, THIS STATEMENT ADDRESS IS STORED, AND A JUMP TO PC = 77_8 IS PERFORMED. PROGRAM STOPS.
INTERRUPT RESET:	EXTERNAL SOURCE	INTERNAL PROGRAM IS RESTARTED AT BREAK-POINT GIVEN ABOVE.

REAL-TIME SIGNALS FROM THE CORRELATOR

SIGNAL	FUNCTION
CORRELATOR RUN:	BIT 1 FROM STATUSWORD, ACTIVE WHENEVER PC ≠ 00 AND PC ≠ 77_8
FLAG:	SET/RESET BY PROGRAM (I/O-INSTRUCTION)
DMA-REQUEST:	REQUEST OF COMPUTER DMA-CYCLE, SET BY DATA-READY
DATA-READY:	COMMUNICATION SIGNAL TO CAMAC WHEN DATA IS READY FOR TRANSFER. SET BY OUT-INSTRUCTION
ERROR-INTERRUPT:	ACTIVE WHEN ONE OF THE ERROR BITS IN CONTROL-WORD IS SET.

FRONT-PANEL COMMANDS

All real-time commands can be generated on the front-panel when the correlator is in manual status (front-panel switch).

POWER-ON PROCEDURE:

Set power on and push master reset button. Select status: Manual or remote.

DATA/PROGRAM-FIELD LOADING FROM FRONT-PANEL:

Manual status must be selected. Set addr., sub-addr., on chip-switches:

	11	10	9	8	7	6	5	4	3	2	1	0	SWITCH NO.
ADDRESS FORMAT	x	x	x	x	x	x	x	x	x	x	x	x	
	ADDR.						SUB-ADDR.						

Push address-load (correlator must not be busy) and led-indicator becomes active. Set data value on chip-switches and push data-load. Load-sequence is finished. Repeate for new loading.

DATA-DISPLAY FUNCTIONS ON FRONT-PANEL (MANUAL STATUS):

Set display switch to "Front-panel". Led-indicators are showing:

DISPLAY CODE (OCTAL)

0:	STATUSWORD	
1:	CONTROLWORD	
2:	DATA CHANNEL 1, LS-PART	RES-MEMORY ADDRESS GIVEN BY APM
3:	DATA CHANNEL 1, MS-PART	
4:	DATA CHANNEL 2, LS-PART	
5:	DATA CHANNEL 2, MS-PART	
6:	TESTWORD 1	SEE OUT-INSTRUCTIONS
7:	TESTWORD 2	

When display switch is set to program, internal program-instruction out is active. (For PC = 0 display-code is always 0: Statusword).

DISPLAY OF RESULT-MEMORY CONTENT (MANUAL STATUS)

Set display switch to "Front-panel" and load APM-instruction word 7, 0, 1, ØØ, ØØ into program-location 00. By display-code 2, 3, 4 or 5 and memory-address set on chip-switches, memory-content can be read on the led-display.

This function can be used only when PC = 00.

TEST OPTIONS:

SINGLE-CYCLE PROGRAM-EXECUTION (MANUAL STATUS):

Set front-panel switch to "Clock-single". In this mode single instruction-cycle is generated when "Clock-advance" is pushed. When display-switch is set to "Front-panel" all display selections are available. When a front-panel function is required, push function-switch and then push "Clock-advance".

DMA TRANSFER-PROGRAM TESTING (MANUAL STATUS):

Set switch to "Transfer inhibit". In this mode data-ready to CAMAC is inhibited, and program will stop when transfer-instructions are active. Simulation of CAMAC-response can be made by pushing "Data-received".