79/11

EISCAT
TECHNICAL
NOTES

A PROGRAMMABLE CORRELATOR MODULE FOR THE EISCAT RADAR SYSTEM

Hans-Jørgen Alker

KIRUNA
Sweden

A PROGRAMMABLE CORRELATOR MODULE FOR THE EISCAT RADAR SYSTEM

Hans-Jørgen Alker

Report no. 51-78

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, February 1978.

TABLE OF CONTENTS

I. INTRODUCTION

During 1977 a redesign and construction of the digital multibit correlator has been made. For reduction of hardware complexity the multi-channel correlator-system is divided into separate sub-systems, which operate at a smaller level of system complexity. The design concept of parallel data-processing, required for the multi-channel operation, has been changed to a time-serial mode of operation for reduction of hardware elements. A programmable option of system-control has been included for increased flexibility of the system.

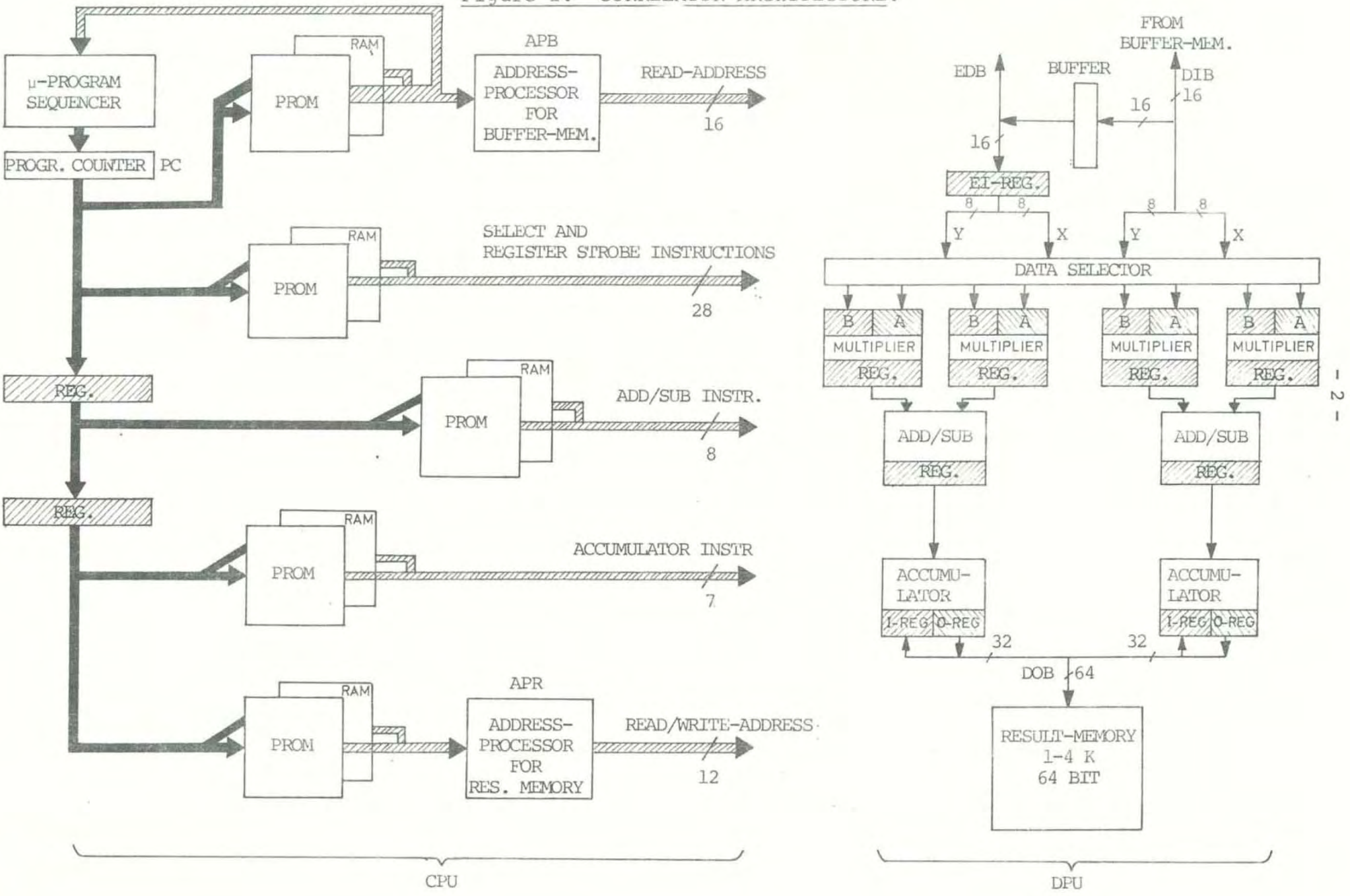
This report describes the architecture of the correlator system with simple block diagrams of the internal modules and may be regarded as a programming model of the system.

II. HARDWARE MODULES OF THE SYSTEM

A block diagram of the system is given in Figure 1. The correlator hard-ware can be split into two processing units, the data-prosessing unit, DPU, and the control-prosessing unit, CPU.

The DPU performs data-processing operations on digitized, signed (2's complement) samples read from the buffer-memory system. The data-input bus, DIB, is 16 bits in parallel , one byte for the x- and y-sample. With the arithmetic elements of the DPU resursive algorithms including complex multiply, add/subtract and accumulation operations are realized. These basic operations are required for auto/cross-correlation measurements and time-average estimation. The data-processing is split into two independent channels for optimum processing speed performance. The correlator has an internal result-memory for scan-to-scan integration without interrupting the on-site computer system. The result-memory is organized in 1 K modules with words of 64 bits, 32 bits for each data-channel. The memory-system is expandable to 4 K (4 K = 4096) word-locations. The data-output bus, DOB, is bidirectional for time-serial read/write operations. For system expansion a bidirectional external data bus, EDB, is realized for transmit/receive operations of data. By use of the data selector mixed loading to the DPU arith-matic from separate buffer-memories is possible.

Figure 1. CORRELATOR ARCHITECTURE.



The control-processing unit, CPU, generates all required control signals within the system including the generation of read addresses to the buffer-memory and read/write addresses to the result memory. The main control unit is the microprogram sequencer which generates the program-counter (PC)value. This is a memory-location pointer which maps the PC value into control instructions to the DPU and feedback to the μ-sequencer for updating the PC value. The sequence of instructions defines a μ-program and is stored in an internal program-memory of the CPU. Two options are selectable for the CPU program memory. With the RAM option (random access memory) the correlator system is μ-programmable. This enables the computer system to control the CPU and DPU on "bit" level by loading a μ-program into the CPU before a radar experiment starts. Within the available program-memory space the RAM can contain several user-defined programs, and a given data-processing algorithm is selectable by defining the initial value of the PC when operations start. With the PROM option (programmable read only memory) a library of predefined, nondestructable programs can be selected. In the construction of the correlator both program-memory options are included and can be selected from the computer. For storage of experiment-defined parameters a distributed register file and small scratch-pad memories located in the address processors are used. Separation of program and data storage maximizes the speed performance.

The DPU and CPU of the correlator are clock synchronized with a cycle time of 150 nsec. This gives an effective data throughput rate of the DPU: 6.67 M samples/sec (complex samples). By time-sequenced operations (pipe-line structure) using temporary storage of data in each stage within the DPU, the given data rate is valid fór all arithmetic operations of the DPU. The result memory is realized with high-speed components and can perform a read and write cycle within the clock period. The given cycle time requires that all real-time commands must be generated by the system itself. In normal operation of the correlator all necessary information is loaded into the system from the computer before the experiment starts, the only required command is a START-COMPUTE signal generated from the radarcontroller after data sampling stops in each radar scan. Preprocessed data can be transfered to the computer, either on direct command from the computer or by software-control within the correlator system. Transfer mode disables data processing, but transfer time can be minimized by DMA (direct memory access)- channel transfer to the computer.

III. DATA PROCESSING UNIT (DPU)

An expanded block diagram of the DPU is shown in Figure 2. Bidirectional, 16 bits external address/data buses, EDB and EAB, have been included in the design for correlator system expansion. Under software control from the CPU the data input bus, DIB, can be connected to the buffer memory or a PROM module, which contains a predefined test sequence (512 values of x-and y-samples). In a test operation this deterministic sequence can be loaded to the DPU for hardware error detection. Internally of the CPU, the address processor for the buffer memory, APB, can generate two 16 bits addresses in parallel . The address-input bus, AIB, generates normally the address which is used in the read-operation of data to the DPU. This bus is called VAB (valid address bus). The other bus from the APB, AIBD (address input bus with displacement) contains the address value of AIB $\pm$ a displacement value. With the buses AIB and AIBD two memory references with a fixed displacement can be generated every clock cycle in parallel . The input transfer of data to the operand registers (A,B) of the multipliers and the input register on EDB are strobed on high $\leftrightarrow$ low transitions of the system clock (50 % duty cycle). The first half of the clock period is termed phase 1, the second phase 2. The input register on EDB is strobed at the end of phase 1 and are under real-time software control. The 8 operand registers of the multipliers have individual strobe signals and the registers are strobed at the end of phase 2. Each operand register can be loaded with either x- or y-samples in DIB or x- or y-samples in EDB. Loading to all operand registers are in parallel

With the given bus handling system several modes of input/output control are available.

MODES OF BUFFER ADDRESS SOURCE

BUFFER ADDRESS SOURCE = NORMAL : VAB = AIB

BUFFER ADDRESS SOURCE = MIXED : phase 1:VAB=AIBD,phase2:VAB=AIBD

BUFFER ADDRESS SOURCE = EXTERNAL: VAB = EAB

With the MIXED mode and enabling the EDB, the input register on EDB can be strobed with samples in memory-reference AIBD in phase 1 and in phase 2 operand registers can be strobed with a combination of samples with reference AIBD and AIB. This structure requires a buffer memory system with access-time half the clock-period and can produce an input transfer of 13.3 M samples/sec (complex samples). The address-source modes NORMAL/MIXED are defined by computer or front panel and are not under real-time control of the CPU. With the use of the EAB the address processor of the CPU can gererate memory references to external memory systems not connected to the DIB. Data transfer are established on the EDB. This external data transfer are under real-time control by software. On request of data from external devices, the EXTERNAL-address mode is used. This request mode is externally cont-rolled by a REQUEST SIGNAL, RS. When RS is active, the address mode changes to EXTERNAL independent of previons mode. After transfer is made, the address mode is redefined.

The bus-handling system constructed enables flexible system-expansion in MASTER/SLAVE and/or multi-correlator configurations.

MODES OF INPUT DATA SOURCE.

INPUT DATA SOURCE = BUFFER-MEMORY:DIB ↔ Buffer memory

INPUT DATA SOURCE = TEST :DIB ↔ PROM system (test sequence)

INPUT DATA SOURCE = EXTERNAL : Mixed DIB, EDB loading to DPU.

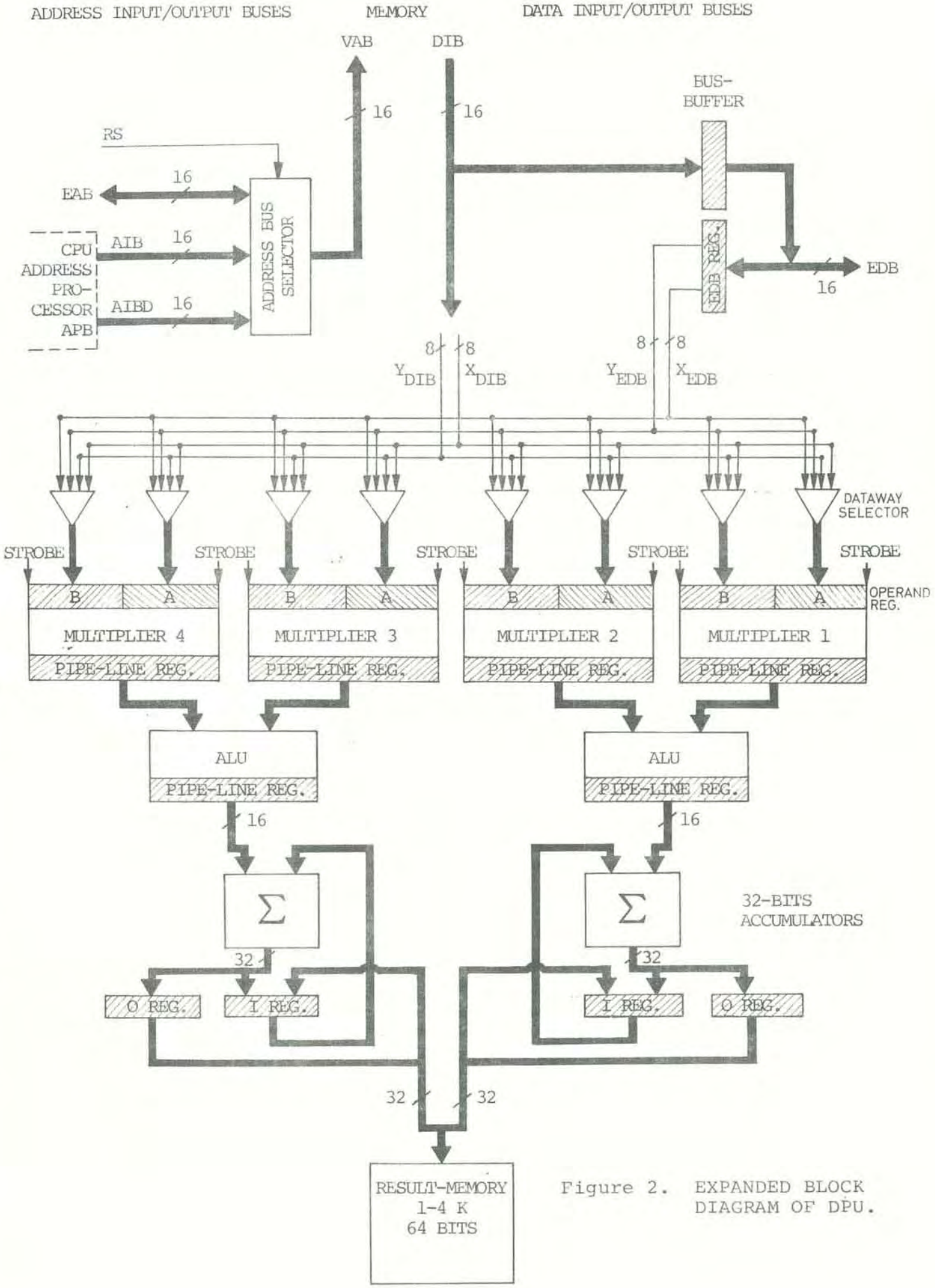


Figure 2. EXPANDED BLOCK DIAGRAM OF DPU.

The EXTERNAL mode does not exclude BUFFER-MEMORY or TEST mode. Mixing of data-inputs are real-time controlled by CPU. This implies that EXTERNAL mode is real-time controlled, the two other modes are not (BUFFER-MEM/TEST are defined by computer or front panel).

The address buses for reading data into the DPU are 16 bits. This implies that one correlator system can address 64 K of buffer memory locations and 64 K of external memory locations through EAB. This means that the correlator system can be directly interfaced to any buffer system of practical size.

The arithmetic/logic part of the DPU consists of a distributed system where different operations are performed in subsequent clock intervals ("pipe-lined" operations). The arithmetic/logic operation in each "stage" is performed in one clock period. With reference to the block diagram in Figure 2 an example of data processing with one sample is sketched in Figure 3. The processing involves the following operations:

1.STAGE: Sample read out from memory, dataway selection and strobe into the operand registers of the multipliers. Each operand register has an individual dataway selection and strobe command. The A register can be loaded with either x- or y-sample on DIB, x- or y-sample on EDB, the value 1, or storing value of the preceding clock period. The B reg. has the same load-functions except for the load 1 operation.

2.STAGE: 4 multiplications (8 by 8 bits) performed on A,B reg. in parallel . Multiplier-results are strobed into registers at the end of clock phase 2.

3.STAGE: The data processing is split into separate datachannels. 2 arithmetic/logic units (ALU's) perform ADD/SUBTRACT operations on pairs of result registers from the multipliers. Also individual result reg. selection is available. The ALU-word format is 16 bits.

4.STAGE: 32 bits accumulation with contents in the result memory is performed in each data channel by reading into the I registers of the accumulators in clock phase 2 of

preceding clock period. In stage 4 two 32-bits binary operations are performed and the results are strobed into the O registers at end of clock phase 2.

5.STAGE: In clock phase 1 the contents of the O registers are written back into the result memory, 64 bits in parallel .

By having the DPU-operations pipe-lined by storing temporary results in each stage, data processing on the next sample can start after ending stage 1 of preceding sample processing, that is, an effective data-processing speed of 6.67 M samples/sec (complex samples) at input. The DPU speed-performance considering arithmetic operations and transfers to/from memory separate, is (effective rates):

DPU ARITHMETIC OPERATIONS : 53.3 MOPERATIONS/SEC

DPU MEMORY-TRANSFERS : 20.0 MTRANSFERS/SEC

With the given DPU structure, the following data-processing restrictions exist:

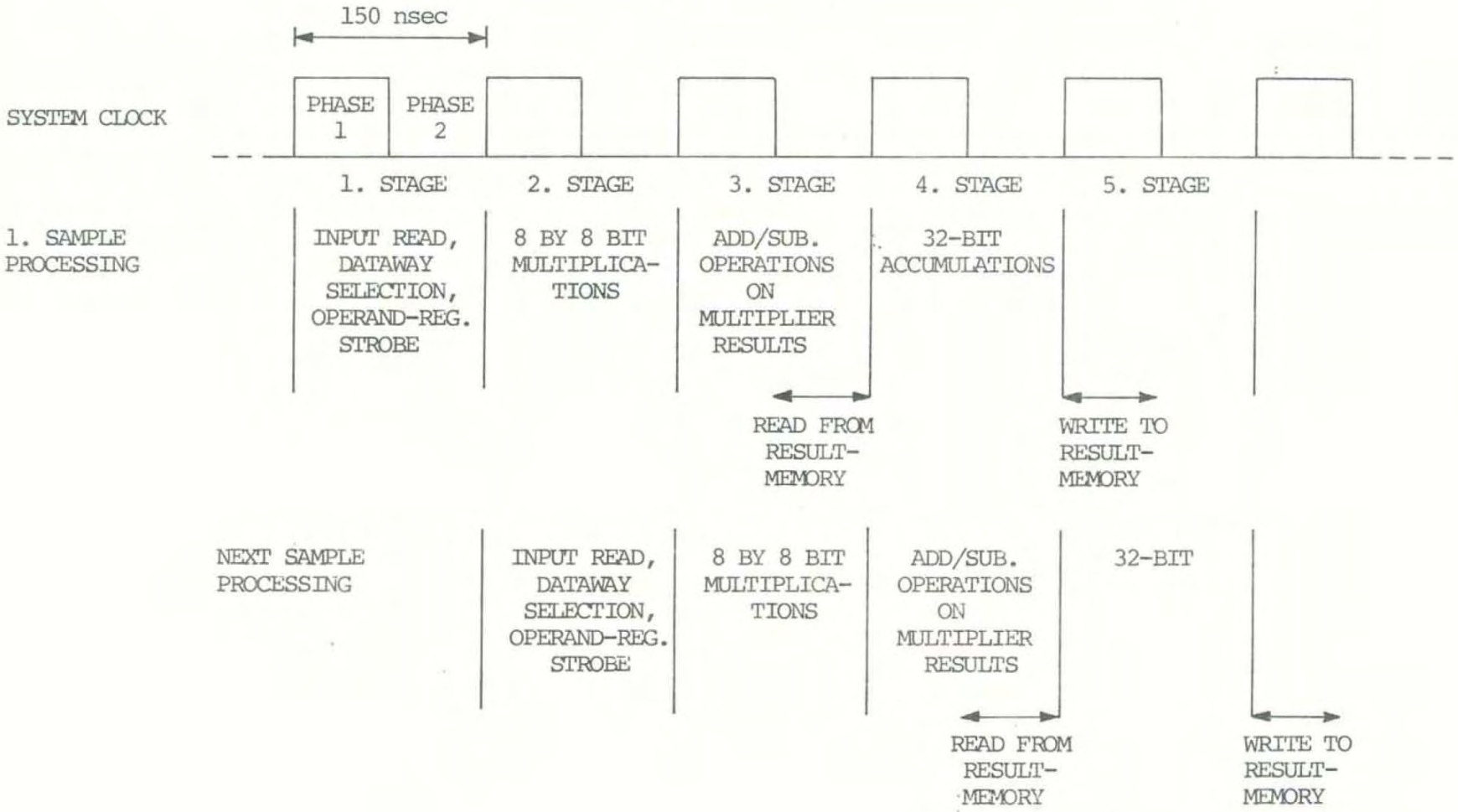
- Control of the 32 bits accumulators are the same for both data-channels.

- READ/MODIFY/WRITE operation referencing the resultmemory is using the same memory address for READ and WRITE transfer.

The real-time control of the DPU is handled by the correlator CPU by clock syncronized control signals to the different modules of the DPU. The CPU of the correlator is a sequence-unit which generates control-signal patterns enabling the DPU to perform a recursive data processing algorithm. The control-signal patterns are stored in a program memory of the CPU. One location in this memory defines all control signals to the DPU for one clock interval.

In the following subchapters requried control signals and programming models of the DPU elements are given.

Figure 3. TIME-SEQUENCED OPERATIONS OF THE D P U .



III.1. DATAWAY SELECTION

A programming model of the dataway selection to the operand registers of multiplier no.1 is shown in Figure 4. A set of 7 control signals, m_1 to m_7, is required for operating the element. The 4 multipliers of the DPU are equal and 4 sets of control signals must be generated.

The operand registers A and B are strobed at the end of phase 2 of the system clock, if the corresponding STROBE signal is set.

The pipe-line register storing the multiplier result is strobed at the end of phase 2.

Decoding rules:

A REG.SELECTION	$m_3m_2m_1$ =	000	$X_{DIB} \rightarrow A$
		001	$Y_{DIB} \rightarrow A$
		010	$X_{EDB} \rightarrow A$
		011	$Y_{EDB} \rightarrow A$
		1ØØ	$1 \rightarrow A$
A REG.STROBE	m_4 =	0	NO STROBE
		1	STROBE
B REG.SELECTION	m_6m_5 =	00	$X_{DIB} \rightarrow B$
		01	$Y_{DIB} \rightarrow B$
		10	$X_{EDB} \rightarrow B$
		11	$Y_{EDB} \rightarrow B$
B REG.STROBE	m_7 =	0	NO STROBE
		1	STROBE

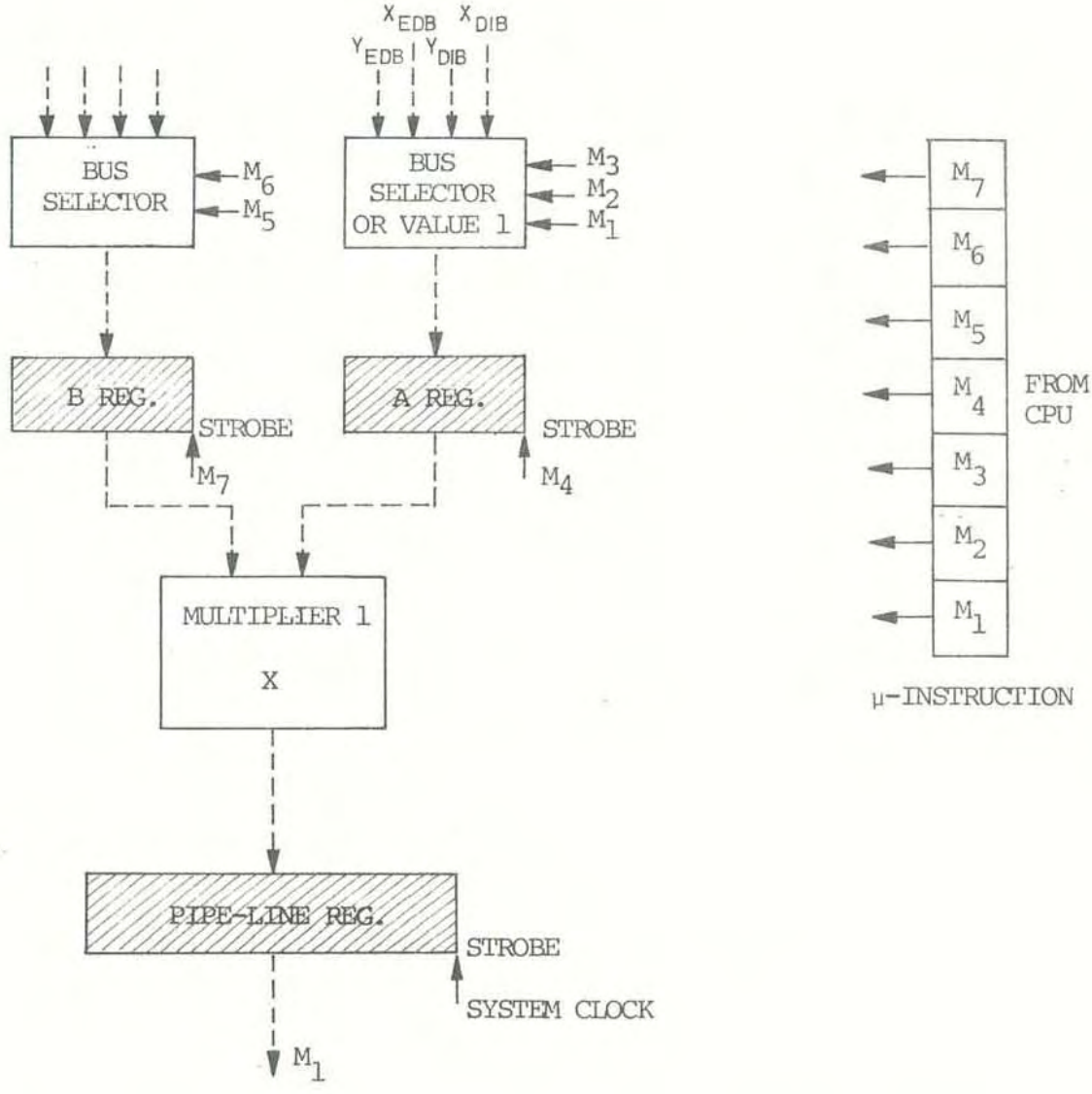


Figure 4. PROGRAMMING MODEL OF DATAWAY SELECTION.

III.2. ARITHMETIC OPERATIONS ON MULTIPLIER RESULTS.

The two arithmetic elements for each data-channel are equal, each requiring a set of 4 control signals. In Figure 5 the ALU-element for multiplier 1 and 2 is shown.

Decoding rules: $a_4a_3a_2a_1$ = 1111 Result M_1

1010 Result M_2

1001 Result M_1+M_2

0110 Result M_1-M_2

0011 Result eq. to -1 (TEST OPTION)

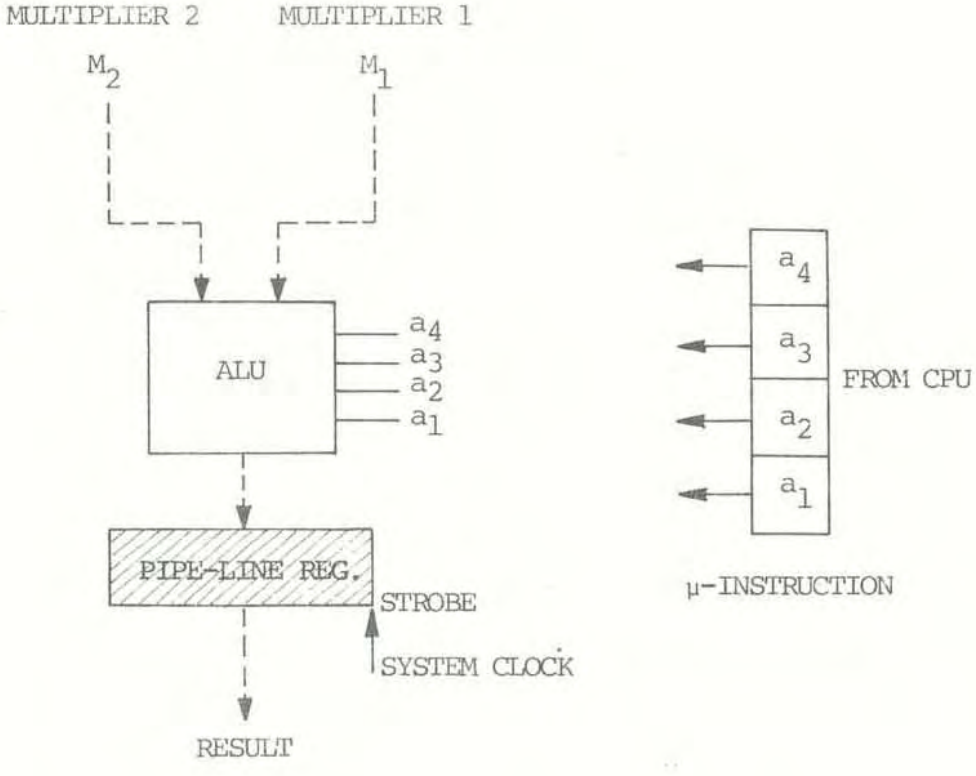


Figure 5. PROGRAMMING MODEL OF THE MULTIPLIER-RESULT ARITHMETIC.

III.3. THE 32-BITS ACCUMULATORS

The 32-bits accumulators of the data-channels are equal, but are controlled by the same set of signals from the CPU. The block diagram of the accumulator is given in Figure 6. Basic operations include result-memory READ/WRITE through the I and O registers and an internal ACCUMULATION mode using the I reg. When the signal c_3 is set, strobing of I/O reg. (in parallel) occurs at the end of clock phase 2. The signal c_1 controls the selection of input to the I register, when set the memory bus is selected as input and with operation by the c_3 signal, a load sequence is generated in clock phase 2. Then c_1 is cleared, the 32-bits binary adder chain is selected as input. When c_2 is set, the O register is enabled for memory bus transfer during the clock phase 1. of the succeeding clock interval after the O register is strobed.

When operating the DPU, two distict modes of computations can be defined: START and CONTINUE EXPERIMENT. The START EXPERIMENT mode corresponds to the first radar-scan processing, where the scan to scan integration should be suppressed. With the given result-memory structure the content of the memory can not be cleared so the START EXP. mode is handled by inhibiting the memory read command (c_1) and instead loading the value O into the I register. This function is controlled by a flip flop (FF1) which is under real-time control by the CPU by generating c_4 (RESET) or c_5 (SET) to FF1. This handling implies that different programs have to be constructed for CPU controlling the two modes. This has been avoided by including a higher priority flip flop (FF2) in the construction. When FF2 is preset by the computer, the FF1 control function is inhibited, and a CONTINUE EXP. mode is performed. When FF2 is reset, the input loading is under control of FF1.

The FF2 is also under real-time control by the signals c_6 (RESET) and c_7 (SET). With this structure all programs can be constructed for START EXP. computations. At the end of the first scan processing the internal program can set FF2, defining a CONTINUE EXP. mode.

By option the computer can preset FF2 before conputations start, defining a CONTINUE EXP. mode without changing the internal program of the CPU. In same way the computer can directly restart a processing by reset of FF2 without changing the internal program.

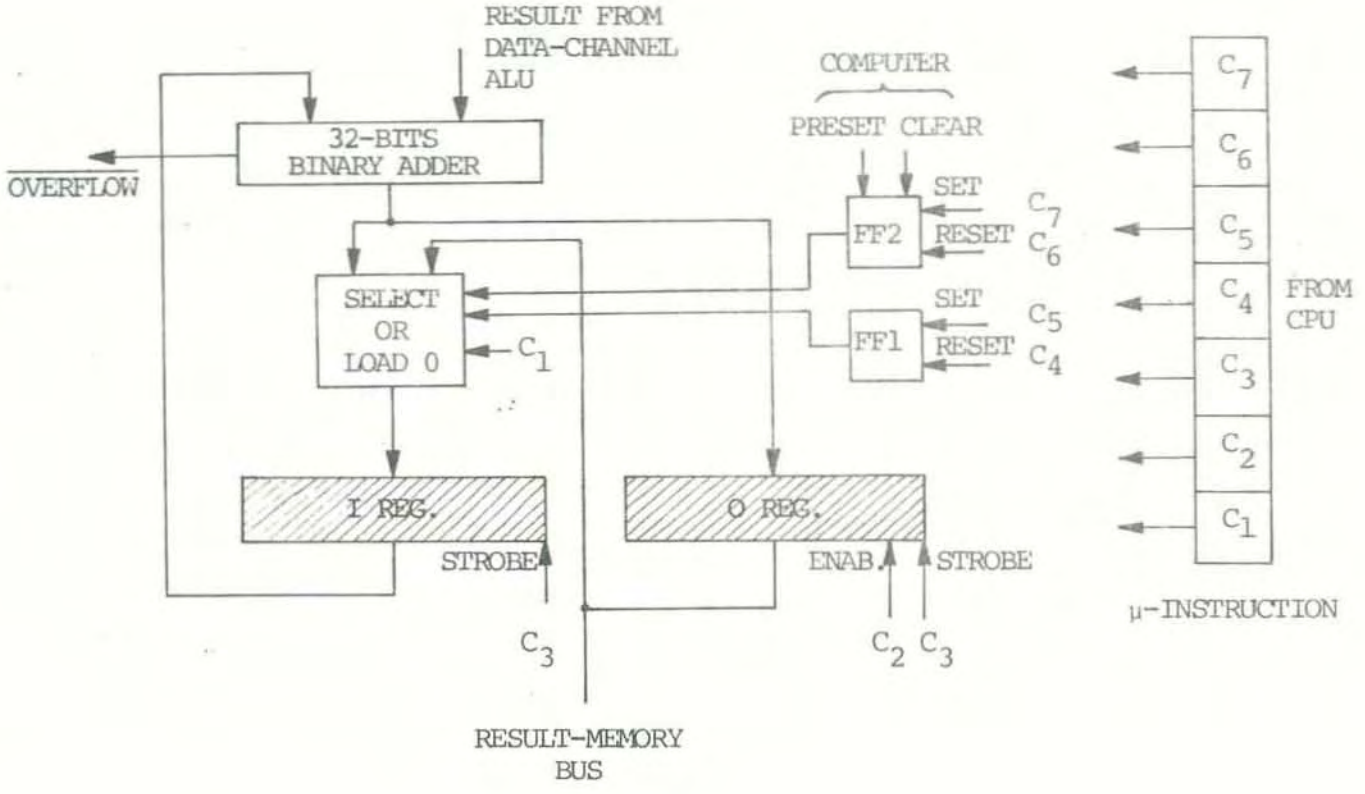


Figure 6. PROGRAMMING MODEL OF THE ACCUMULATOR.

Decoding rules:

MEMORY READ c_1: 0 select internal accumulation

1 select res.memory bus (READ)

MEMORY WRITE c_2: 0 noop

1 output transfer

STROBE SIGNAL c_3: 0 no strobe

1 strobe I/O registers (end of clock phase 2)

FF1 c_5c_4: 00 no change of FF1

01 set FF1, memory read control given by c_1

10 reset FF1, zero loading instead of read

FF2 c_7c_6: 00 no change of FF2

01 set FF2, inhibit FF1-control of c_1

10 reset FF2, enable FF1-control of c_1

IV CONTROL PROCESSING UNIT (CPU)

The main processing operation of the CPU is generation of the requried μ-instructions (43 control lines in parallel) from cycle to cycle of the system clock. These instructions define the DPU operations. The CPU has an internal program memory storing the set of μ-instructions. When memory READ/WRITE operations are defined, the CPU internally generates the corresponding addresses to the buffer and result-memory. For the real-time address processing, 35 additional control lines from the program memory are requried.

The typical data processing algorithms to be used involve recursive data operations: Sample by sample multiplications and accumulations, the same data-processing algorithm is used in a recycled mode on different sample sets (range-gate organization of the samples stored in the buffer memory). For minimalization of the CPU program-memory size, an internal program control has been constructed for conditional/ unconditional program branching and program looping during execution of a program in real-time. 28 control lines from the program-memory are used for this purpose.

Additional "background" operations has to be performed by the CPU. These involve control of the interfaces between the correlator and the computer/radar-controller. The correlator/computer communication can be split into 2 basic operations: Program-load/parameter-definitions transferred from the computer before computations start and transfer of preprocessed data from the result-memory to the computer. Result-memory data-transfer is under real-time program control by the CPU. Communication between the correlator and the radar-controller involves only a transfer of a START COMPUTE signal, generated in real-time from the radar-controller. This signal un-conditioanlly starts a predefined μ-program inside the correlator CPU. After the start has been initiated the CPU controls all internal operations of the correlator without any external communications. The computer can, however, at any time stop internal operations of the CPU by generation of a MASTER RESET signal which unconditionally cancels a running program. Also a PROGRAM INTERRUPT signal can be generated from the computer which forces the CPU to a WAIT mode. The WAIT reset operation is also externally controlled.

The CPU of the correlator has an internal, programmable mode STOP which inhibits the internal system clock. In transfer of data from the correlator to the computer using DMA-channel, the CPU system clock is under direct computer control by the control signals on the DMA-channel. This STOP mode enables single instruction execution inside the CPU for compensation of speed mismatch between the computer system and the correlator. The transfer of a single location of the result-memory (two 32-bit words for the two data channels of the DPU) are divided into transfer of 4 (16-bit) words on the DMA-channel. The transfer of data is programmable and options like half-word transmissions from the result-memory are available. Also, by software control, main operating modes of the correlator and correlator-module identification, internal bus values inside the CPU, can be transferred to the computer. These options enable the user to have computer analysis of new programs in test run.

The CPU has internal error detecting systems which mainly control nummerical overflow of the DPU accumulators (Fig. 6) and external control-signal generation. When an error is detected an ERROR INTERRUPT signal is set at the correlator/computer interface. By request from the computer a CONTROL-WORD can be transferred enabling the error mode to be identified by the computer.

The correlator front panel gives information (diode displays) on present status of the correlator. Front panel functions are also interfaced with the CPU for direct control/programming the system without use of the computer, this option is included mainly for test/service of the correlator.

A block diagram of the CPU hardware architecture is given in Figure 7. The CPU module contains:

- μ-PROGRAM MEMORY with selection of RAM/PROM modules from the computer. Each part consists of 8 separate hardware modules with 64 words of 16 bits.

- μ-PROGRAM SEQUENCER which controls the real-time execution of the μ-program. The μ-program seq. generates a 6-bit program-counter (PC) which gives a 6-bit address reference to the μ-program memory. The PC value is by output from the memory

mapped
into 128 separate control signals in parallell. 28 of these signals are used for definition of next PC value. The μ-instructions to the sequencer include:

- CONDITIONAL/UNCOND. BRANCHING OF PC COVERING THE COMPLETE PROGR. MEMORY

- SUBROUTINE JUMPS (4 LEVELS AVAILABLE)

- FLAG SET/CLEAR

- VECTORIZED JUMP ON INTERRUPT REQUEST FROM COMPUTER

- PROGRAMMABLE WAIT/STOP COMMAND

- PROGRAM LOOPING WITH CONTROL OF 3 SEPARATE PROGRAMMABLE LOOP-COUNTERS.

- CPU contains 2 SEPARATE ADDRESS PROCESSORS for buffer/result-memory address generation. Each processor contains

  - ALU MODULE (16 BITS FOR BUFFER, 12 BITS FOR RES.MEMORY).

  - REGISTER FILE WITH 17 REGISTERS (2 PORTED, READ/WRITE ON SAME CLOCK CYCLE):

  Address range for buffer: 64 K (16-BIT WORDS)
  Address range for res.memory: 4 K (64-BIT WORDS)

- STATUS REGISTER directly programmable from the computer/front panel. The status reg. is 16 bits and the content defines basic operational modes of the correlator.

- CONTROL REGISTER giving status of 8 separate error-indications and correlator-module identification code.

In the following subchapters programming models of the different parts of the CPU are given.

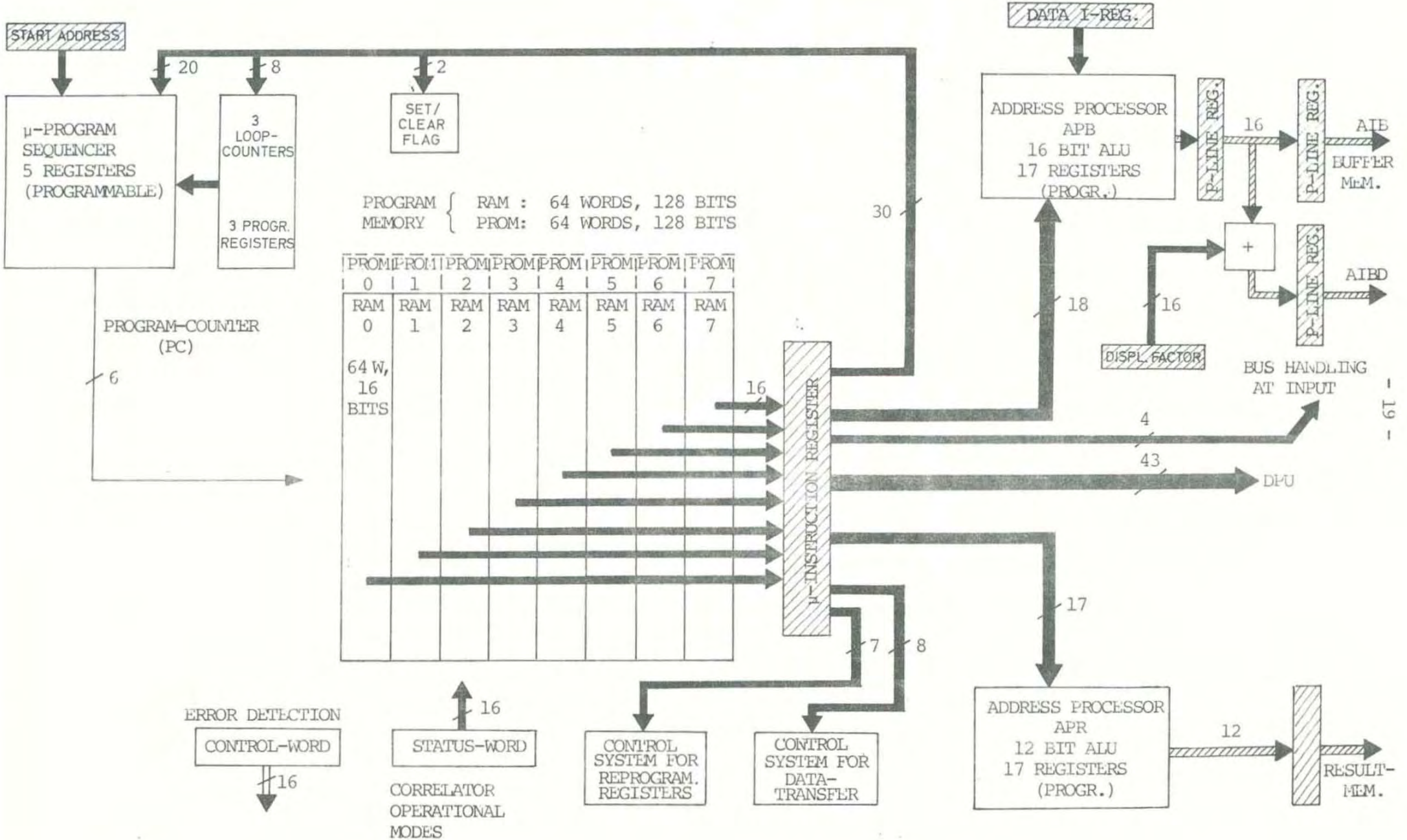


Figure 7. BLOCK DIAGRAM OF CPU HARDWARE

IV.1. MAIN OPERATIONAL MODES OF THE CORRELATOR. DEFINITION OF STATUS-WORD.

The STATUS-WORD is a 16-bit register, which by loading from the computer or front panel, defines the main operational modes of the correlator. Several bits are not programmable but show the present status of the system.

STATUS-WORD DECODING:

BIT 0: CORRELATOR READY
0 → NOT READY
1 → READY

The bit is not programmable from the computer but indicates if the system is ready for axcepting a START COMPUTE signal.

BIT 1: CORRELATOR RUN
0 → IDLE STATUS
1 → μ-PROGRAM RUNNING

The bit is not programmable.

BIT 2: INPUT LOAD REQUEST
0 → NO REQ.
1 → INPUT DATA/PROGRAM LOAD REQUESTED BY COMPUTER/INTERNAL PROGRAM OR FRONT PANEL

The bit is not programmable.

BIT 3: PROGRAM SOURCE
0 → RAM MODULES SELECTED
1 → PROM MODULES SELECRED

The bit can be set by computer/front panel.

BIT 4: DATA SOURCE
0 → DATA FROM BUFFER MEMORY
1 → DATA FROM PROM MODULE (TEST SEQUENCE).

The bit can be programmed.

BIT 5: ACCUMULATION MODE
0 START EXPERIMENT
1 CONTINUE EXPERIMENT

The bit can be programmed (see chapter III.4).

BIT 6: BUFFER ADDRESS SOURCE
0 NORMAL
1 MIXED (see ch. III)

The bit can be programmed.

BIT 7: DPU SIGNALS TO SLAVES 0 → SAME
1 → INVERTED

In a given correlator-system configuration the CPU of one of the modules can be a MASTER module with the others in SLAVE operation. The SLAVE modules need only have the DPU part realized in hardware. Bit 7 in the status-word indicates that the control signals to the DPU modules of the slaves are the same as in the master system, that is data input to the slaves are through the individual DIB bus. When the inverted signal is sent, the data should be entered through the EDB bus in the slaves. The bit is programmable.

BIT 8-9: DATA TRANSFER ENABLE 00 → MASTER MODULE
01 → SLAVE NO.2
10 → SLAVE NO.3
11 → SLAVE NO.1

This mode is only active during a programmed transfer of data to the computer. By the 2-bit selection one of four modules can be selected for output transfer.

BIT 10-11: START EXPERIMENT SOURCE 00 → CONTROLLED BY FRONT PANEL
01 → - " - RADAR-CONTROLLER
10 → - " - COMPUTER

Start compute signal is only accepted when the device is selected.

BIT 12-15: CORRELATOR MODULE IDENTIFICATION NO.

This value is not programmable but is defined by hardware wireing on one of the circuit cards. The module identification no. is used when entering data from the computer into one of the correlator modules in a multi correlator system. The actual setting of this no. is always displayed on the front panel.

IV.2. CONTROL-WORD ORGANIZATION

The control-word is a 16-bit register which contains the present error status. When an error is detected the corresponding error bit is set and can only be reset by a MASTER RESET command.

CONTROL WORD DECODING:

BIT 0. When set: An input load request has been set but the correlator is busy (CORRELATOR RUN actice).

BIT 1. When set: Start compute signal is generated from an external device, but the correlator is in manual operation.

BIT3. When set: Start command generated but CORRELATOR RUN is active.

BIT 3. When set: Start command generated but correlater is not ready.

BIT 4. When set: Accumulator overflow in slave no.3.

BIT 5. When set: Accumulator overflow in slave no. 2.

BIT 6. When set: Accumulator overflow in slave no.1.

BIT 7. When set: Accumulator overflow in master-module.

BIT 12-15.CORRELATOR MODULE IDENTIFICATION NO.

IV.3. µ-PROGRAM EXECUTION

The system for real-time control of the program execution is given in Figure 8. Together with the program memory, the system realizes a feedback system for generation of a new PC value; the progr.-memory reference for the next readout of the 28-bit µ-instruction word in the next clock interval. The execution cycle is synchronized with the system-clock of the correlator. The PC value definition depends on the control signals S_1, S_0, $\overline{FE}$ and PUP from the condition-code test-system. This system tests present status of the loop-counters (LC), and, depending on the result of the test, generates a given set of S_1, S_0, $\overline{FE}$ and PUP. The PC value can either be defined by the T register content, the content of the START ADDRESS register, the last value loaded into the register stack or directly from the µ-program itself by the DIRECT ADDRESS input. The T register is always loaded with the PC value +1 at the end of clock phase 2. The µ-program sequencer contains a 4-register stack in LIFO (last in first out) operation. For stack operation the FILE ENABLE ($\overline{FE}$) and PUSH/POP STACK (PUP) commands are generated. For the µ-sequencer the following decoding is used:

PC VALUE SELECTION S_1, S_0:	00	T REG.
	01	DIRECT ADDRESS (DA)
	10	REGISTER STACK (RS)
	11	START ADDR.REG. (SAR)

SYNCHRONOUS STACK CONTROL $\overline{FE}$, PUP:	1Ø	NOOP
	01	PUSH
	00	POP

PC value and internal next-cycle register states are given in Figure 9. A total of 12 different operations are available.

For generation of the 4 control bits S_1, S_0, $\overline{FE}$, PUP, the condition-code test system uses bit P_1 to P_{14} of the µ-instruction word. These 14 bits are divided into 3 subsets: Code-set A, code-set B and a 6-bit condition code. The organization is shown in Figure 10. Depending on the bit settings in the condition-code, the test system generates either code-set A or code-set B to the µ-program sequencer. In a conditional test operation the decision is depending on the present loop-counters. Also a unconditional mode is available by programming.

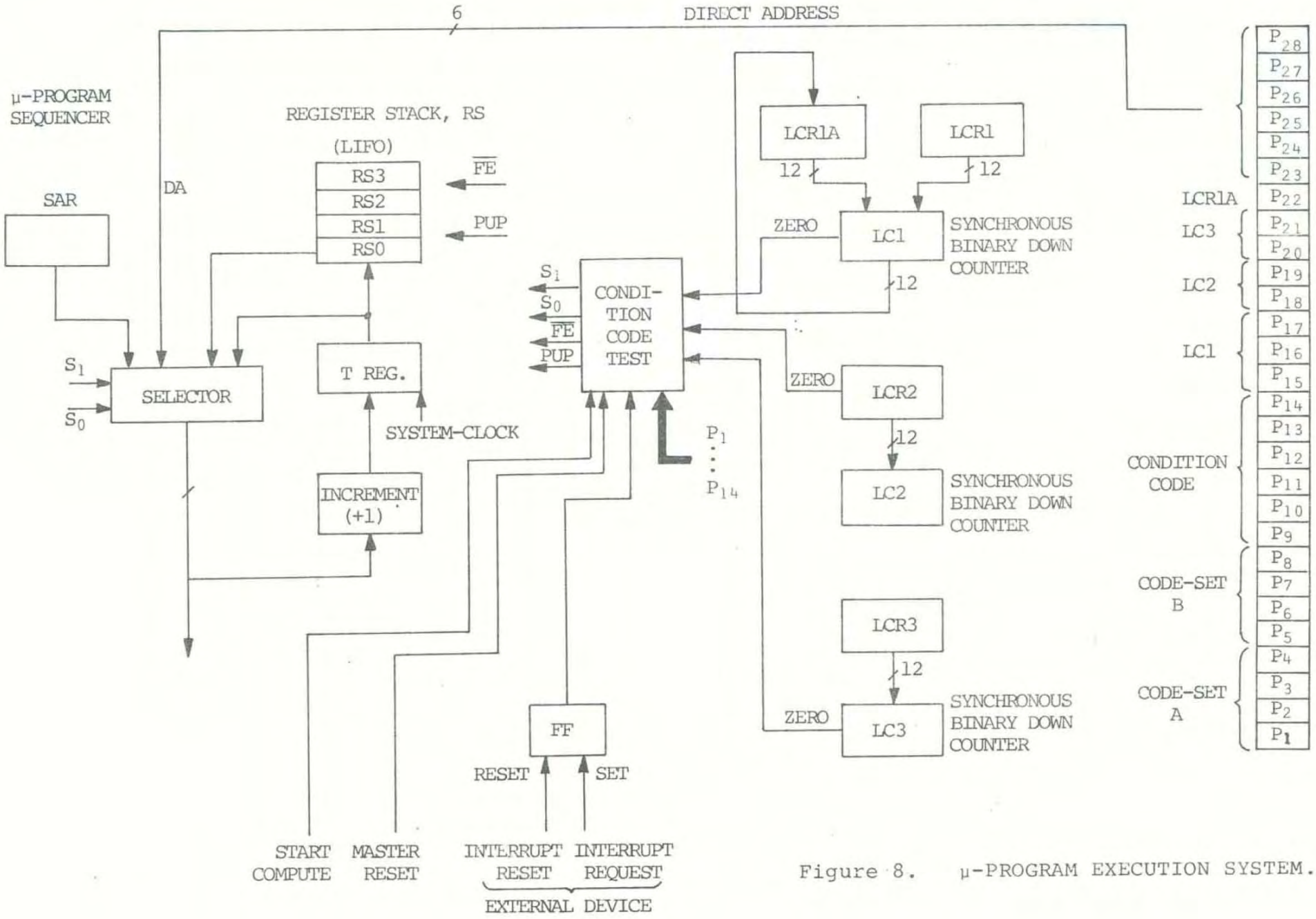


Figure 8. μ-PROGRAM EXECUTION SYSTEM.

The register SAR and loop-counter registers LCR1, LCR2 and LCR3 can be directly loaded from the computer or front panel. By use of the instr. bit P_{22} the LCR1A can be loaded with the present content of the loop-counter LC1.

CLOCK CYCLE	$S_1, S_0, \overline{FE}$, PUP	T REG	DA	RS0	RS1	RS2	RS3	PC-VALUE	COMMENT	PRINCIPLE USE
N N+1	0000 -	J J+1	K K	Ra Rb	Rb Rc	Rc Rd	Rd Ra	J -	POP STACK	DELETE LOOP
N N+1	0001 -	J J+1	K K	Ra J	Rb Ra	Rc Rb	Rd Rc	J -	PUSH T REG.	SET-UP LOOP
N N+1	001Ø -	J J+1	K K	Ra Ra	Rb Rb	Rc Rc	Rd Rd	J -	CONTINUE	CONTINUE
N N+1	0100 -	J K+1	K K	Ra Rb	Rb Rc	Rc Rd	Rd Ra	K -	POP STACK USE DA FOR ADDR.	DELETE LOOP JUMP
N N+1	0101 -	J K+1	K K	Ra J	Rb Ra	Rc Rb	Rd Rc	K -	PUSH T REG. USE DA FOR ADDR.	DIRECT JUMP TO SUBROUTINE
N N+1	011Ø -	J K+1	K K	Ra Ra	Rb Rb	Rc Rc	Rd Rd	K -	USE DA FOR ADDR.	DIRECT JUMP
N N+1	1000 -	J Ra+1	K K	Ra Rb	Rb Rc	Rc Rd	Rd Ra	Ra -	USE ADDR. IN RS0 POP STACK	RETURN FROM SUBROUTINE
N N+1	1001 -	J Ra+1	K K	Ra J	Rb Ra	Rc Rb	Rd Rc	Ra -	USE ADDR. IN RS0 PUSH T REG.	
N N+1	101Ø -	J Ra+1	K K	Ra Ra	Rb Rb	Rc Rc	Rd Rd	Ra -	USE ADDR. IN RS0	END LOOP RETURN
N N+1	1100 -	J SAR+1	K K	Ra Rb	Rb Rc	Rc Rd	Rd Ra	SAR -	POP STACK USE SAR FOR ADDR.	DELETE LOOP, JUMP
N N+1	1101 -	J SAR+1	K K	Ra J	Rb Ra	Rc Rb	Rd Rc	SAR -	USE SAR FOR ADDR. PUSH T REG.	SAR-JUMP TO SUBROUTINE
N N+1	111Ø -	J SAR+1	K K	Ra Ra	Rb Rb	Rc Rc	Rd Rd	SAR -	USE SAR FOR ADDR.	JUMP TO SAR

Figure 9. PC value and internal next-cycle register states for the μ-program sequencer.

CONDITION-CODE INSTRUCTION FORMAT:

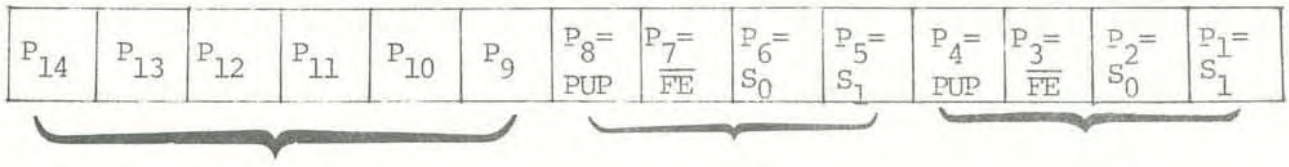


Figure 10.

CONDITION-CODE DECODING:

TEST OPTION I, $P_{14}P_{13}P_{12}P_{11}P_{10}P_9 = 1ØØØØØ$

IF TEST IS TRUE USE CODE-SET B,
ELSE USE CODE-SET A.

TEST CONDITION,	(LC3) P_{13}	(LC2) P_{12}	(LC3) P_{11}	(LC2) P_{10}	(LC1) P_9	
	Ø	Ø	0	0	0	USE CODE A UNCONDITIONALLY
	1	1	0	0	1	LC1 = 0
	1	1	0	1	0	LC2 = 0
	1	1	0	1	1	LC1 OR LC2 = 0
	1	1	1	0	0	LC3 = 0
	1	1	1	0	1	LC1 OR LC3 = 0
	1	1	1	1	0	LC2 OR LC3 = 0
	1	1	1	1	1	LC1 OR LC2 OR LC3 = 0
	1	0	Ø	Ø	Ø	SAME AS BEFORE WITH LC2 ≠ 0
	0	1	Ø	Ø	Ø	SAME AS BEFORE WITH LC3 ≠ 0
	0	0	Ø	Ø	Ø	SAME AS BEFORE WITH LC2, LC3 ≠ 0

TEST OPTION II, $P_{14}P_{13}P_{12}P_{11}P_{10}P_9$ = 0ØØØØØ

IF TEST 1 IS TRUE USE CODE-SET B,
ELSE IF TEST 2 IS TRUE USE CODE-SET A,
ELSE CONTINUE (generate S_1, S_0, $\overline{FE}$, PUP = 0010)

TEST-CONDITION, $P_{13}P_{12}P_{11}P_{10}P_9$

1	1	0	1	1	TEST 1:LC1 = 0, TEST 2:LC2 = 0
1	1	1	1	0	TEST 1:LC3 = 0, TEST 2:LC2 = 0
1	1	1	0	1	TEST 1:LC1 OR LC3 = 0
1	1	1	1	1	TEST 1:LC1 OR LC3 = 0,TEST 2:LC2=0
1	0	Ø	Ø	Ø	CHANGED POLARITIES ON COUNTERS IN TEST 2
0	1	Ø	Ø	Ø	CHANGED POLARITIES ON COUNT. IN TEST 1
0	0	Ø	Ø	Ø	- " - TEST 1 AND TEST 2

The loop-counters have 12 bits and have individual control bits from the µ-instr.word. The following decoding is used:

LC1 INSTRUCTION, $P_{17}P_{16}P_{15}$ =

000	NOOP
001	DECREMENT COUNTER LC1 = LC1 - 1
010	LOAD COUNTER LC1 = LCR1
011	LOAD COUNTER LC1 = LCR1A
100	LC1 = LCR1 AND LC2 = LC2 - 1 IF LC1 = 0, ELSE LC1 = LC1 - 1 AND LC2 NOOP.
101	LC1 = LCR1A IF LC1 OR LC3 = 0, ELSE LC1 = LC1 - 1
110	LC1 = LCR1 IF LC1 = 0, ELSE LC1 = LC1 - 1
111	LC1 = LCR1A IF LC1 = 0, ELSE LC1 = LC1 - 1

LC2 INSTRUCTION, $P_{18}P_{19}$ =

00	NOOP
01	LC2 = LC2 - 1
11	LC2 = LCR2

LC3 INSTRUCTION, $P_{21}P_{20}$ = 00 NOOP
01 LC3 = LC3 - 1
10 LC3 = LCR3
11 LC3 = LCR3 IF LC3 = 0,
ELSE LC3 = LC3 - 1.

LCR1A INSTRUCTION, P_{22} = 0 NOOP
1 LCR1A = LC1

The loop-counters are operated at the end of clock phase 2 of the system clock. This implies that the counters are operated at the end of the μ-instruction execution cycle. The register LCR1A are loaded at the end of clock phase 1 (middle of the ex.-cycle).

The lowest and highest address in the program memory, 00_8 and 77_8 (octal), are used for special status of the correlator system. When PC = 00_8, the incrementer of the μ-sequencer is inhibited, and this location is used as a IDLE STATUS. The program is started by generation of the START COMPUTE signal from either computer/radar-controller or front panel. If the given signal source is enabled by bit-settings in the status-word and a program is not presently running (if CORRELATOR RUN is "0"), the program is started by the automatic operation: PC = SAR. The SAR must always contain the start address for the program. Also, the last instruction of a program must contain a PC = 00_8 jump. A running program can always be cancelled by the MASTER RESET command, this forces the PC = 00_8. The MASTER RESET will reset all flip flops which are set by ecternal devices or under program control. Also the CORRELATOR READY mode is cancelled. A simplified flow diagram of the program execution is given in Figure 11. The CPU has an option of program interrupt. When the interrupt FF is set, an interrupt sequence starts in the first cycle a CONTINUE command ($S_1S_0\overline{FEPUP}$ = 0010) is generated from the test system. The PC value is stored in the register stack and a program break point is performed by the jump PC = 77_8. In location 77_8 the start address of an interrupt-service routine can be stored or the program can be stopped. Break-point with restart on the stored PC value in the register stack is performed by external reset of the interrupt-FF. Except for the locations 00_8 and 77_8 a CORRELATOR RUN signal is set indicating run of a μ-program.

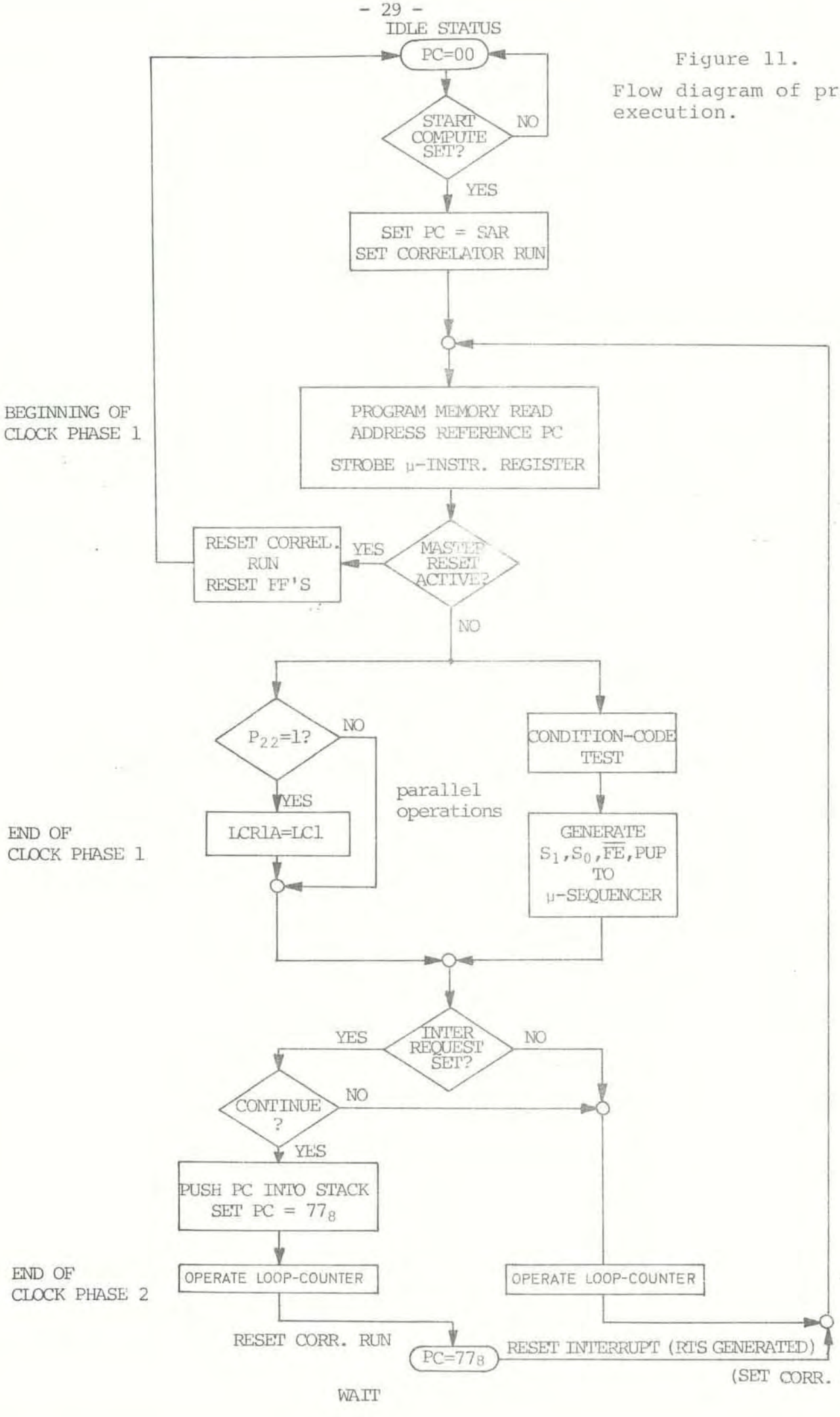


Figure 11.

Flow diagram of program execution.

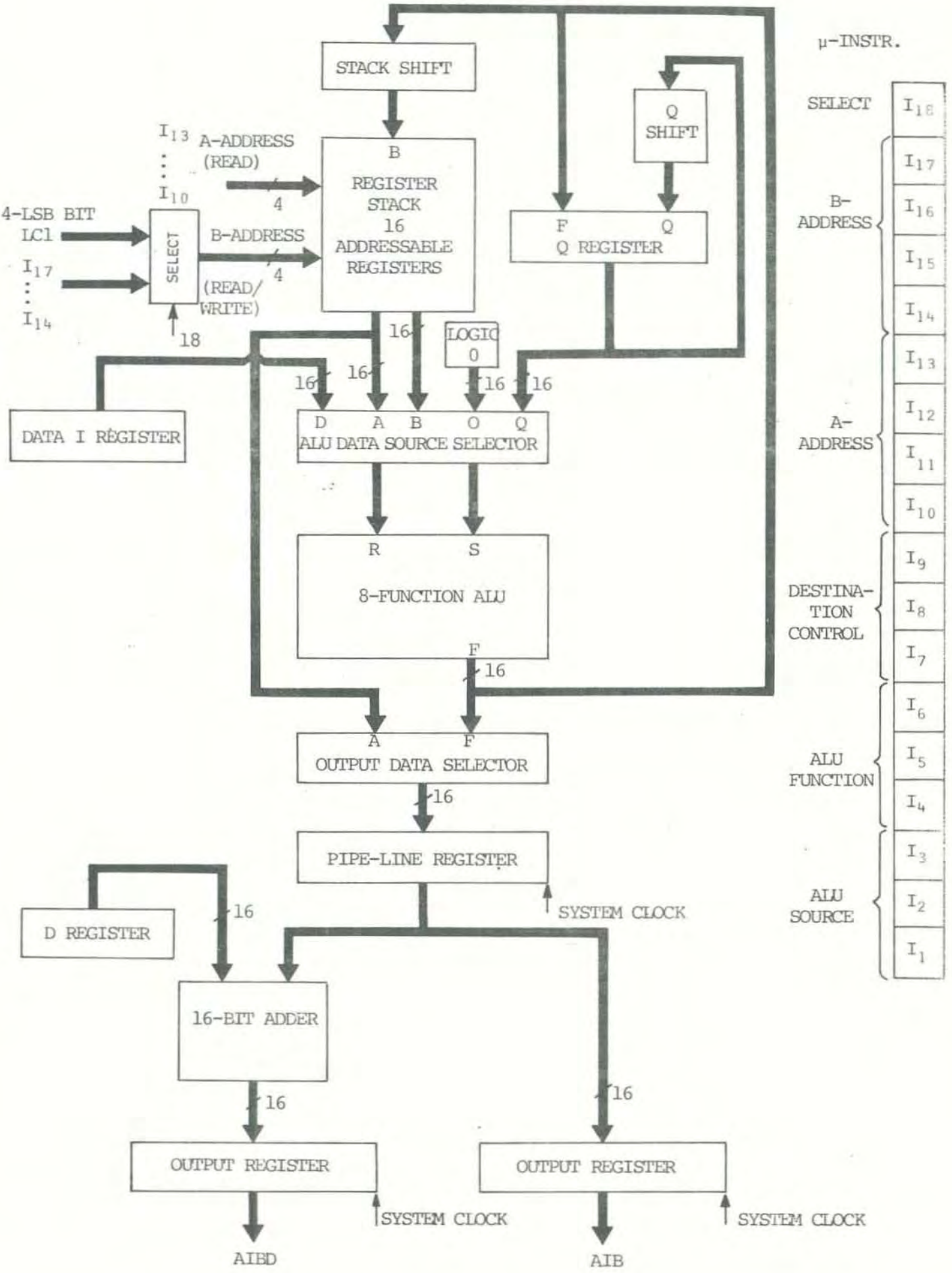
IV.4. THE ADDRESS PROCESSORS APB AND APM

The two address processors of the CPU are functionlly equal. The address processor for the buffer memory, APB, is a 16-bit processor. The resultmemory processor, APM, has 12-bit representation. The block diagram of the APB is given in Figure 12. The APM has simpler construction with the exclusion of the DATA I register and the option of selecting loop-counter value for the B-address. The address bus handling at output is also different as given in Figure 13.

The APB uses a 18-bit μ-instruction word as definition of system operation (APM has 17-bit, without I_{18}). The internal decoding of the μ-instruction word is given in Figure 14. The 16 registers in the register stack, DATA I reg. can be loaded directly from the computer/front panel in the idle status of the correlator. The register stack and the Q register are two-ported which enables a read/modity/write operation with the same stack reference in one execution cycle.

During program run, one can use the address processors in an index relative addressing mode. Before the program starts absolute memory address references together with increment/decrement values can be loaded into the register stack. In program execution, one can compute new values for the address using the arithmetic/logic unit (ALU) and the feedback loop in a recursive addressing mode. With the loop-counter option for B address , APB has a program relative stack reference.

Figure 12. ADDRESS PROCESSOR FOR BUFFER MEMORY (APB)



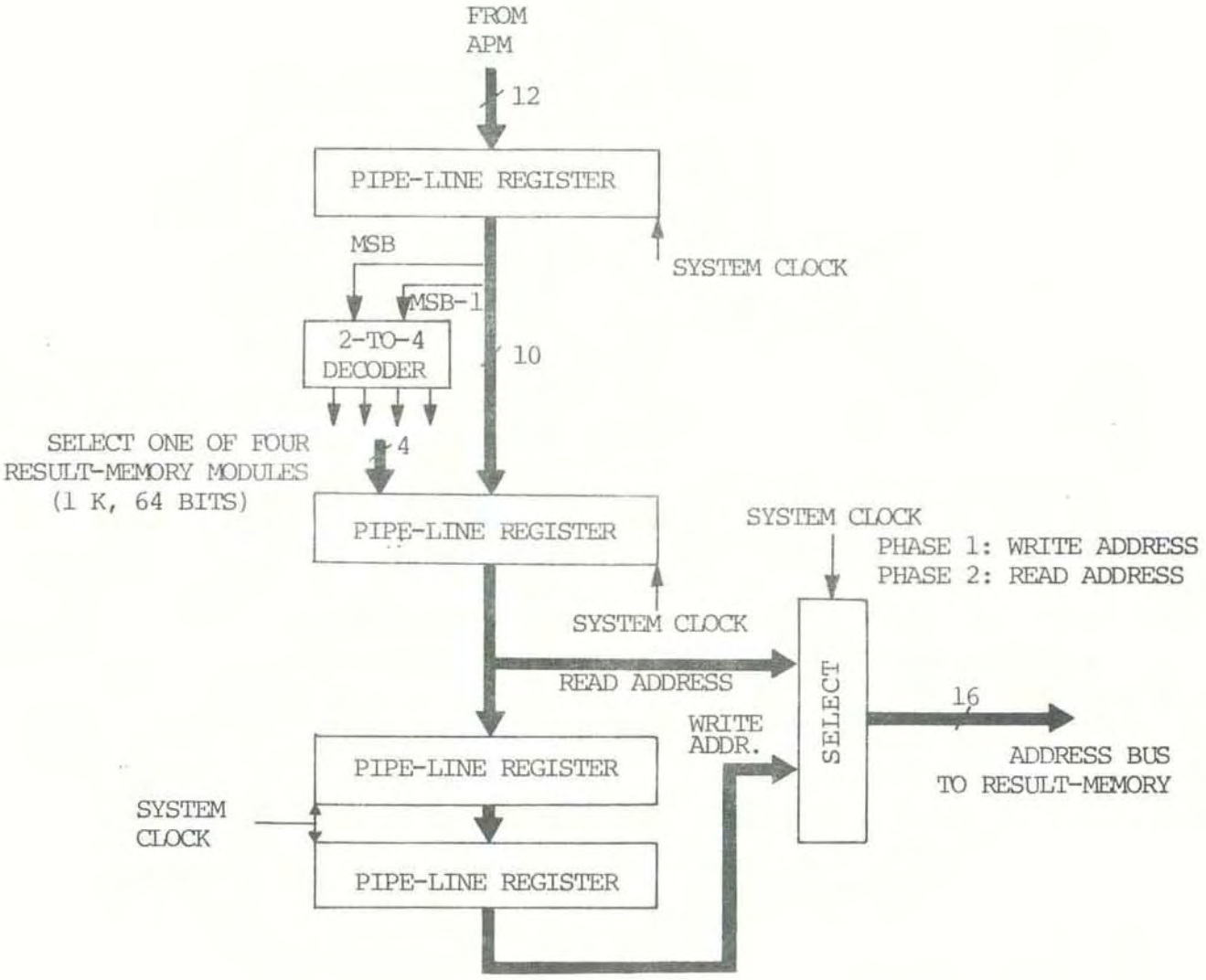


Figure 13. ADDRESS BUS HANDLING FOR THE APM.

MICRO CODE			ALU SOURCE OPERANDS		
I_3	I_2	I_1	OCTAL	R	S
0	0	0	0	A	Q
0	0	1	1	A	B
0	1	0	2	O	Q
0	1	1	3	O	B
1	0	0	4	O	A
1	0	1	5	D	A
1	1	0	6	D	Q
1	1	1	7	D	O

ALU-SOURCE OPERAND CONTROL

MICRO CODE			ALU FUNCTION
I_6	I_5	I_4	OCTAL	
0	0	0	0	R+S
0	0	1	1	S-R
0	1	0	2	R-S
0	1	1	3	R OR S
1	0	0	4	R AND S
1	0	1	5	$\overline{R}$ AND S
1	1	0	6	R-EX-OR S
1	1	1	7	R-EX-NOR S

ALU FUNCTION CONTROL

MICRO CODE				REGISTER-STACK FUNCTION		Q-REG. FUNCTION		OUTPUT
I_9	I_8	I_7	OCTAL	SHIFT	LOAD	SHIFT	LOAD	
0	0	0	0	X	NONE	NONE	F → Q	F
0	0	1	1	X	NONE	X	NONE	F
0	1	0	2	NONE	F → B	X	NONE	A
0	1	1	3	NONE	F → B	X	NONE	F
1	0	0	4	DOWN	F/2 → B	DOWN	Q/2 → Q	F
1	0	1	5	DOWN	F/2 → B	X	NONE	F
1	1	0	6	UP	2F → B	UP	2Q → Q	F
1	1	1	7	UP	2F → B	X	NONE	F

ALU DESTINATION CONTROL

I_{654} \ I_{321}		0	1	2	3	4	5	6	7
	ALU FUNC-TION \ ALU SOURCE	A,Q	A,B	O,Q	O,B	O,A	D,A	D,Q	D,O
0	R+S	A+Q	A+B	Q	B	A	D+A	D+Q	D
1	S-R	Q-A	B-A	Q	B	A	A-D	Q-D	-D
2	R-S	A-Q	A-B	-Q	-B	-A	D-A	D-Q	D
3	R OR S	A∨Q	A∨B	Q	B	A	D∨A	D∨Q	D
4	R AND S	A∧Q	A∧B	O	O	O	D∧A	D∧Q	O
5	$\overline{R}$ AND S	$\overline{A}$∧Q	$\overline{A}$∧B	Q	B	A	$\overline{D}$∧A	$\overline{D}$∧Q	O
6	R EX-OR S	A∀Q	A∀B	Q	B	A	D∀A	D∀Q	D
7	R EX-NOR S	$\overline{A∀Q}$	$\overline{A∀B}$	$\overline{Q}$	$\overline{B}$	$\overline{A}$	$\overline{D∀A}$	$\overline{D∀Q}$	$\overline{D}$

SOURCE OPERAND AND ALU FUNCTION MATRIX

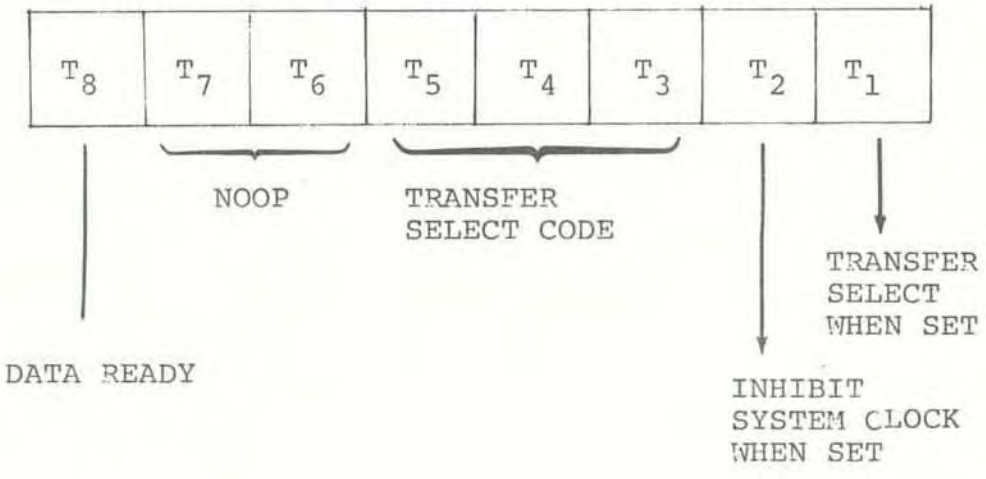
Figure 14. DECODING OF THE INSTRUCTION WORD FOR THE ADDRESS PROCESSORS.

IV.5. OUTPUT TRANSFER OF PROCESSED DATA TO THE COMPUTER.

The NORD-10 computer has a 16-bit memory word and a transfer of one location in the result-memory requires 4 16-bit word transfers to the computer. For reducing the required transfer-time, the communication between the result-memory and the computer should use a direct-memory access, DMA. The present scheme for controlling the DMA-channel transfer uses 2 control signals: A DATA READY signal from the correlator, when data is valid on the DMA bus, and a DATA RECEIVED signal generated from the computer.

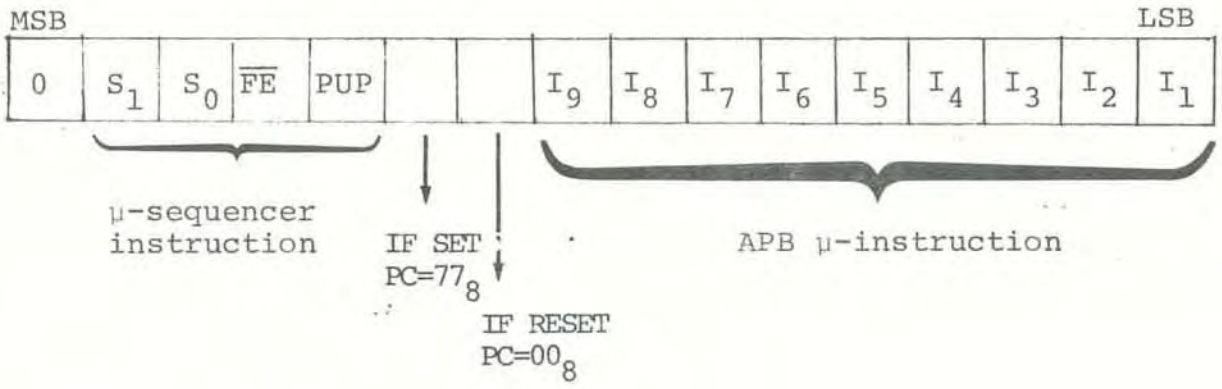
The TRANSFER mode of the correlator is controlled by a μ-program execution. The transfer-program can be processed in connection with other programs or initiated from the computer by reprogramming the START ADDRESS register and generate a START COMPUTE signal. In a multi correlator system, the module enabled is given by bits 8-9 in the STATUS-WORD. The correlator CPU uses a 8-bit μ-instruction word for in real-time controlling the transfer.

μ-INSTRUCTION WORD FOR DATA TRANSFER:

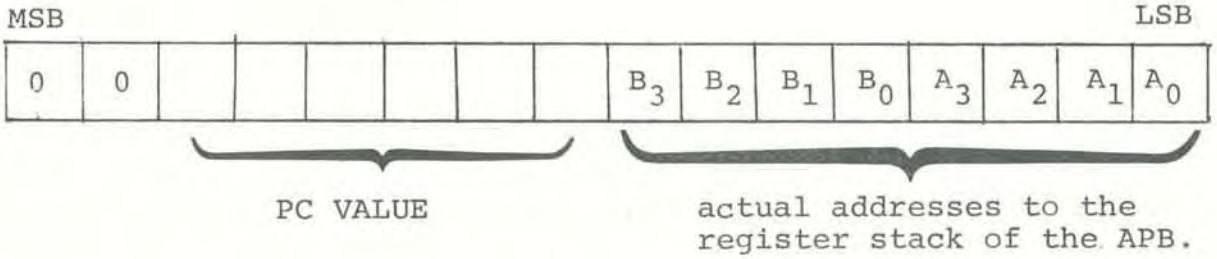


TRANSFER SELECT CODE $T_5T_4T_3$= 000 STATUS WORD
001 CONTROL WORD
010 RES.MEMORY X_{LS} } Data-channel 1.
011 RES.MEMORY X_{MS} }
100 RES.MEMORY Y_{LS} } Data-channel 2.
101 RES.MEMORY Y_{MS} }
110 TESTWORD 1
111 TESTWORD 2

TESTWORD 1 ORGANIZATION:



TESTWORD 2 ORGANIZATION:



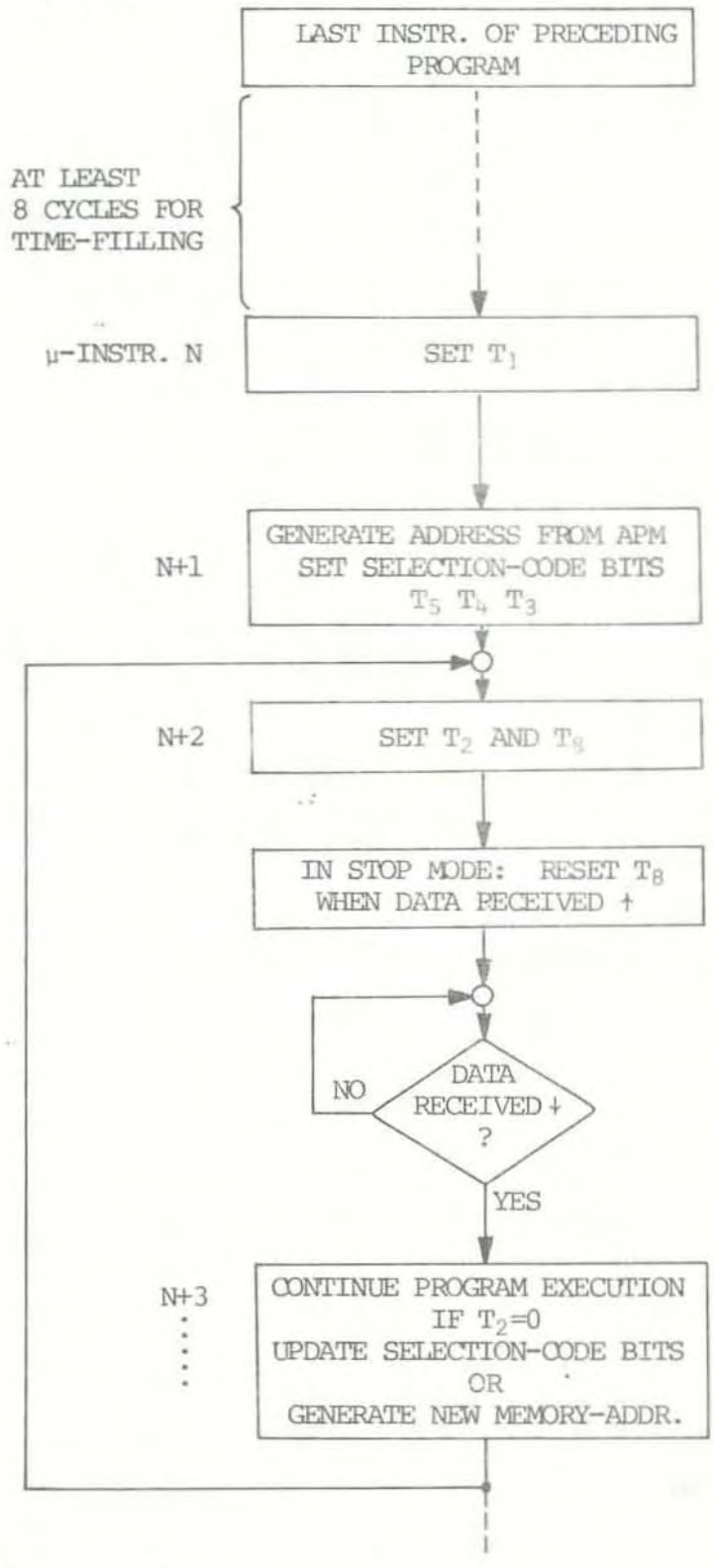
Testword 1 and 2 can be used for tracing the execution of a μ-program.

The μ-instruction bit T_1 is used for internal setup for data transfer and is displayed at the correlator front panel. The bit T_2 is used for synchronize the operations of the correlator with the computer DATA RECEIVED signal. When the bit T_2 is set, the internal system clock is inhibited and the correlator CPU executes one μ-instruction after the high to low transition of the DATA RECEIVED signal from the computer. A flow diagram of a transfer-program and timing-diagram of the DMA control signals are given in Figure 15. With this output transfer handling the correlator can

be interfaced with any external system having a longer cycle-time.

The correlator module is not locally generating the computer memory address when DMA is requested. It is assumed that this operation is performed by the DMA interface.

FLOW DIACRAM:



TIMING DIAGRAM:

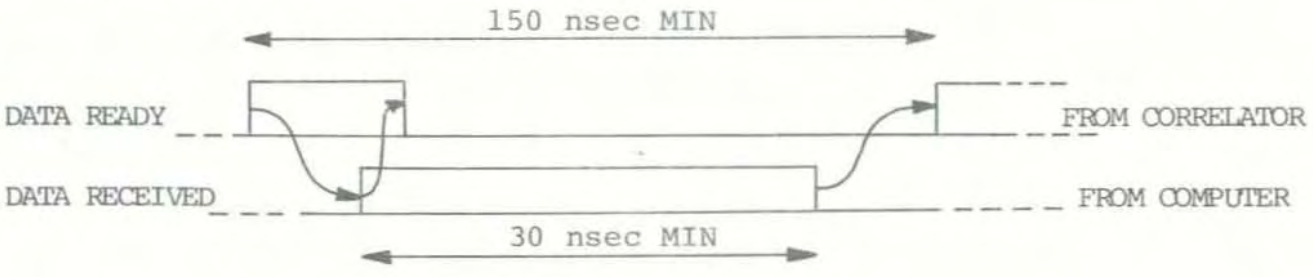


Figure 15. DMA TRANSFER TO COMPUTER.

IV.6. CORRELATOR INTERFACES, INPUT LOAD OF DATA/PROGRAMS TO THE CORRELATOR CPU.

The correlator interfaces with corresponding control signals are given in Figure 16. The interface handling can be split into 4 subgroups:

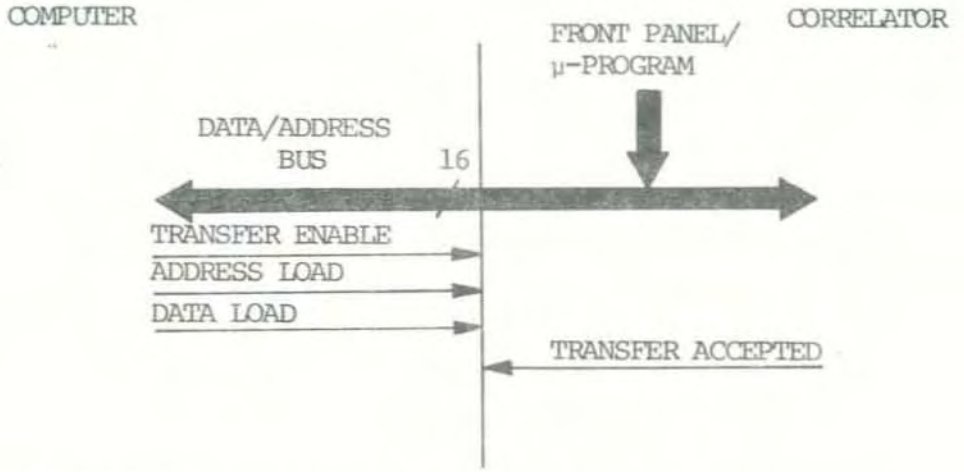
a) Input data/program load of correlator CPU for definition of internal operation.
b) Interrupt handling correlator ↔ computer during execution of program.
c) Real-time signals from radar-controller during a radar experiment execution.
d) Data transfer correlator → computer on DMA-channel.

SUBGROUP A: Input data/program load of correlator CPU

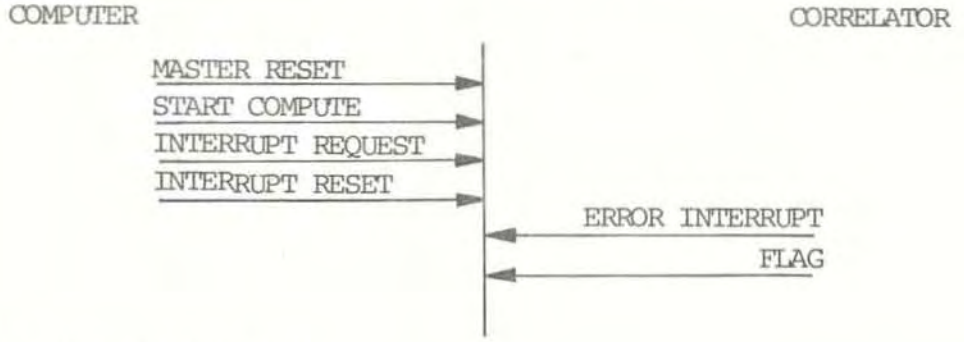
The correlator is equipped with a bidirectional 16-bits data/address bus for input/output transfer. The bus handling is controlled by 3 (active low) signals from the external device: TRANSFER ENABLE, ADDRESS LOAD AND DATA LOAD. TRANSFER ENABLE must always be active when DATA OR ADDRESS LOAD is generated. Input loading is performed on request only if the correlator is in remote control (front panel setting) and the CORRELATOR RUN signal is not active. If these conditions are not fulfilled, errorbits 1 and/or 2 are set and error-interrupt is generated to the external device. Input load to the correlator is performed in two cycles. In the first cycle a transfer of address must be performed . This address defines the dataway selection inside the CPU module. Address trandfer mode is initiated by setting TRANSFER ENABLE and ADDRESS LOAD active (active period 150 $nsec_{MIN}$). The correlator will generate a TRANSFER ACCEPED signal if input-transfer is enabled. Then the second cycle is initiated by setting TRANSFER ENABLE and DATA LOAD active.

Figure 16. SIGNALS IN THE CORRELATOR INTERFACES.

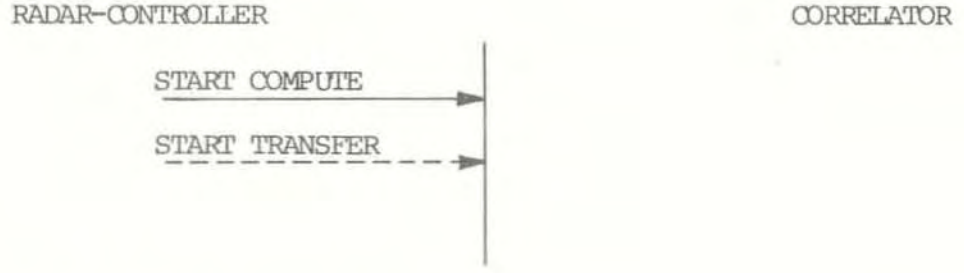
a) PROGRAM/DATA LOAD



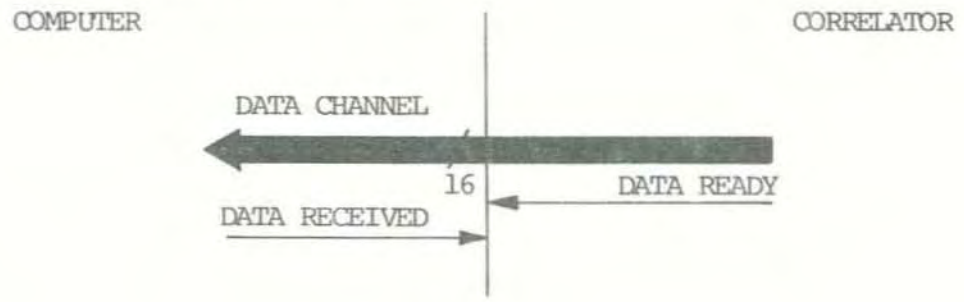
b) INTERRUPT/CONTROL SIGNALS DURING PROGRAM EXECUTION



c) START-INITIALIZATION



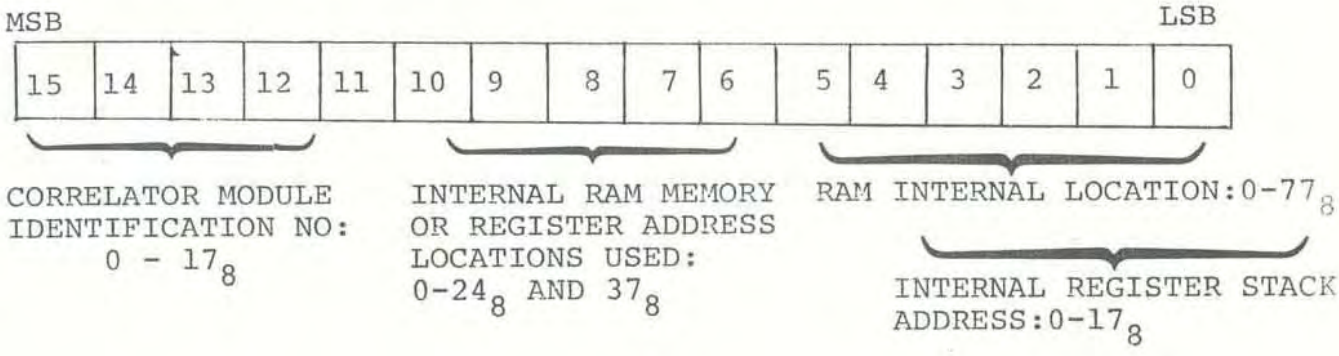
d) TRANSFER OF PROCESSED DATA



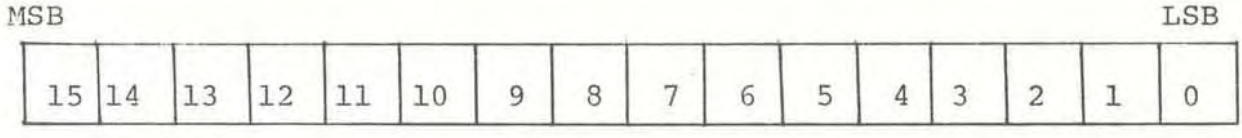
After data is loaded the TRANSFER ACCEPTED is reset and a new transfer can be initiated.

THE TRANSFER WORD FORMAT:

1. CYCLE, TRANSFER ADDRESS (TRANSFER ENABLE/ADDRESS LOAD = 0)



2. CYCLE, TRANSFER DATA (TRANSFER ENABLE/DATA LOAD = 0)



Data must be in binary format and the number of bits used as input depends on the bit-length of the selected module (always the LS-bits are used).

With the addressing decoding used the transfer bus can be connected to 16 different correlator modules.

When manual-control mode is selected from the front panel, the internal bus is directly connected to front panel swithes and input load can be performed.

LIST OF PROGRAMMABLE UNITS INSIDE THE CPU WITH CORRESPONDING IDENTIFICATION ADDRESSES

ADDRESS (OCTAL CODE), BITS 6-10 IN ADDRESS WORD

0: OUTPUT TRANSFER OF STATUS-WORD
1: LOAD STATUS-WORD (16 BITS)
2: OUTPUT TRANSFER OF CONTROL-WORD
3: NOOP
4: LOAD START ADDRESS REGISTER (SAR) FOR PROGRAM (6 BITS)
5: LOAD DISPLACEMENT PARAMETER IN D REGISTER,APB (16 BITS)
6: LOAD DATA I REGISTER, APB (16)
7: NOOP
10: LOAD RAM-MODULE 0, MEMORY ADDRESS GIVEN BY BITS 0-5 (16)
11: - " - 1, - " -
12: - " - 2, - " -
13: - " - 3, - " -
14: - " - 4, - " -
15: - " - 5, - " -
16: - " - 6, - " -
17: - " - 7, - " -
20: LOAD REGISTER STACK IN APB, ADDRESS GIVEN BY BITS 0-3 (16)
21: LOAD REGISTER STACK IN APM, - " - (12)
22: LOAD LCR1 (LOOP COUNTER 1) (12)
23: LOAD LCR2 (- " - 2) (12)
24: LOAD LCR3 (- " - 3) (12)
37: LOAD STATUS COMMAND: CORRELATOR READY (1)

The output transfer (address 0 and 2) is initiated by the 2. cycle control-settings (these transfers can not be controlled by the front panel).

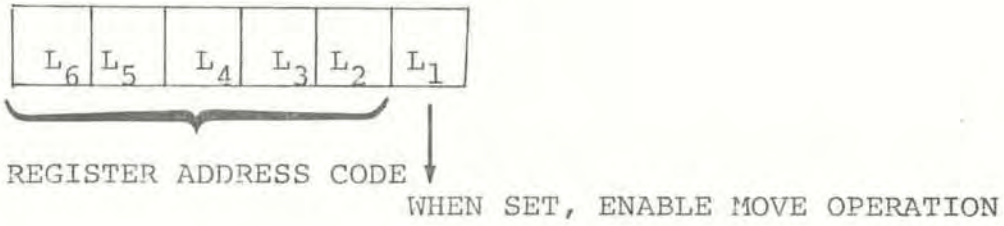
The correlator CPU has a total capasity of

PROGRAM MEMORY: 512 LOCATIONS

DATA MEMORY : 39 LOCATIONS

During program execution (CORRELATOR RUN active), the internal transfer bus is connected to the first pipe-line register at the output of the APB (see Figure 12). By μ-program control, the registers with address 4, 5, 22, 23 and 24 can be reprogrammed by a MOVE instruction command, which enables transfer of data from the register stack of the APB to the given registers.

μ-INSTRUCTION CODE:



The MOVE instruction is executed in 2 clock intervals so the instruction must always be followed by a NOOP-instruction.

SUBGROUP B: Interrupt handling.

The correlator is designed with 2 interrupt signals, which can be handled at separate priority levels in the computer. The error interrupt signal is permanently set after an error is detected and is reset only by the MASTER CLEAR signal. Also a warning signal, FLAG OUTPUT, can be generated. This signal is under μ-program control and can optionally be used for information of program execution.

With the use of MASTER RESET, START COMPUTE and SET/RESET OF PROGRAM INTERRUPT from the computer in a test situation, the computer has a direct control tracing program execution. These signals can be generated from the front panel in manual control.

SUBGROUP C: Real-time signals from the radar-controller.

With the present design only a START COMPUTE signal is generated from the radar-controller. This signal (active high in 150 $nsec_{MIN}$) must be generated for each radar scan to be processed. The signal must be synchronized with the termination of sampling commands to the only analog/digital converters. If the correlator system is prepared for accepting START COMMAND from this source

(bit-setting in the statusword) and the correlator is ready, the START COMMAND initiates a program sequence which sets the PC value equal to the content of the START ADDRESS register (SAR). If a new START COMMAND is generated before the correlator has finished the program execution, an error-bit in the control-word is set.

A problem arises when the correlator scan computations are stopped and the DMA-channel transfer of processed data is initiated. This represents a break point in program execution, and with the present system designed, this can only be handled by a output transfer from the computer redefining the content of the SAR (START ADDRESS register) for reference to a new program routine. The time used for this redefinition is depending on the present priority program status inside the computer. With redesign of the input handling an extra interrupt-line START TRANSFER can be included. In this case the radar-controller must have a program break point in order to generate the START TRANSFER signals. This interrupt handling inside the correlator can be vectorized, that is, the interrupt service routine is the program which transfers data out of the correlator.

SUBGROUP D: DMA-channel transfer is described in chapter IV.5.

IV.7. PROGRAM MEMORY CONFIGURATION, TIME-DIAGRAMS FOR PROGRAM EXECUTION.

One program location in the program memory consist of 128 bits. The bit decoding of the 128-bit μ-instruction word is given in Figure 17.

As given in chapter III, the DATA PROCESSING UNIT (DPU) performs the different arithmetic operations on a sample read from the buffer memory in different clock intervals by having temporary storage registers following each operation. This is sketched in Figure 3. Normally this "pipe-line"-structure implies that a sample processing through the DPU is defined by more then one program location because the processing extend over several clock intervals. This problem has been avoided by also having the program-counter (PC) pipe-lined. With this structure one obtains:

THE μ-INSTRUCTION IN ONE PROGRAM MEMORY LOCATION CORRESPONDS TO ONE SAMPLE PROCESSING THROUGH THE WHOLE DPU.

This means that when programming the correlator for a execution cycle, one need not compensate for the required time-delays inside the DPU.

The internal, fixed time-delays of the control signals after read out of an instruction word from the program memory are given in Figure 18. This diagram is valid when TRANSFER BIT T_1=0 (transfer not selected).

The memory address for output transfer from the result-memory is generated from address processer APM. As given from Figure 18, the execution of a result-memory referenced instruction is performed in clock interval 5, 6 and 8. In order to reduce the required time when output transfer mode is selected (T_1-bit active), the time diagram in Figure 19 is valid.

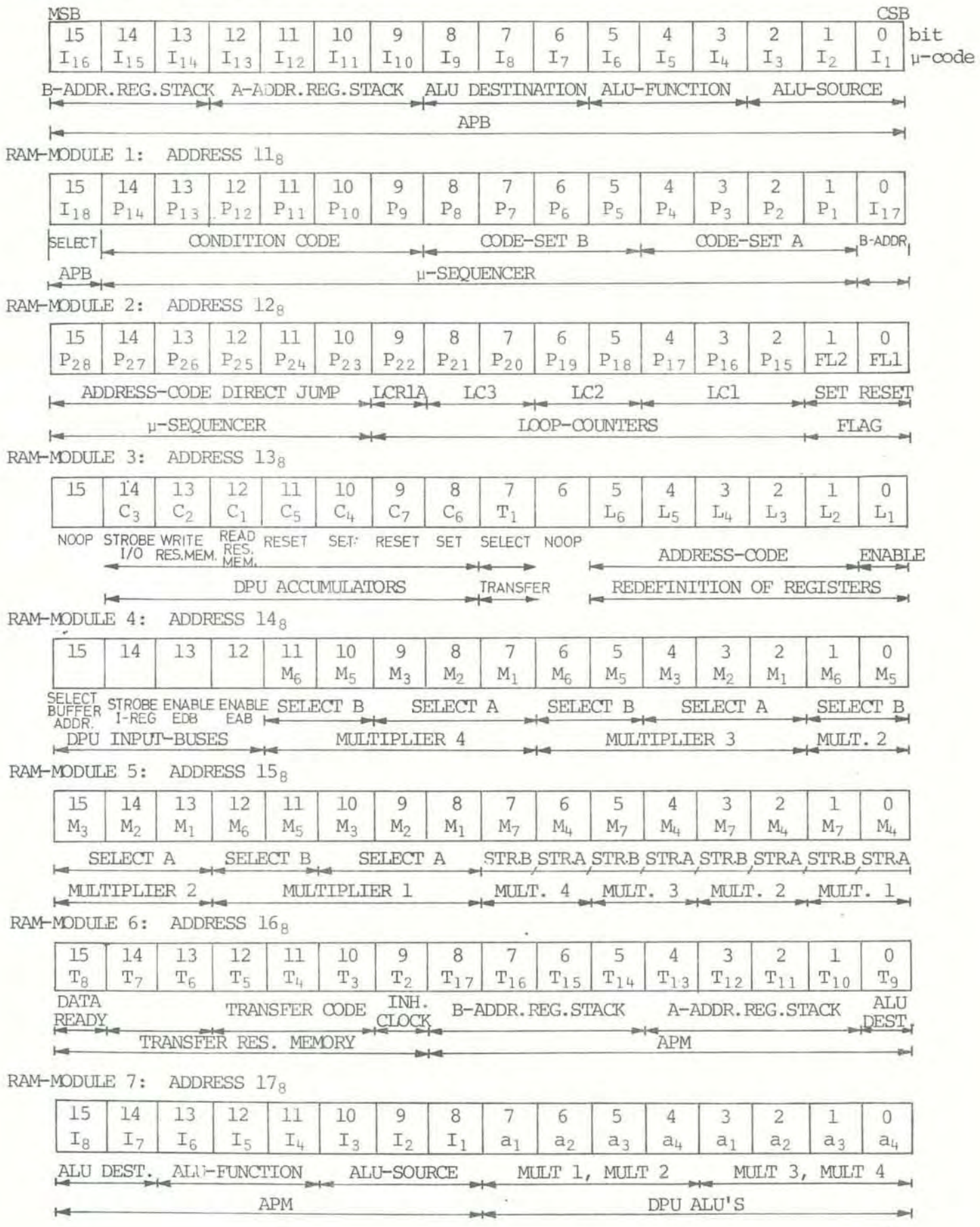


Figure 17. PROGRAM MEMORY CONFIGURATION.

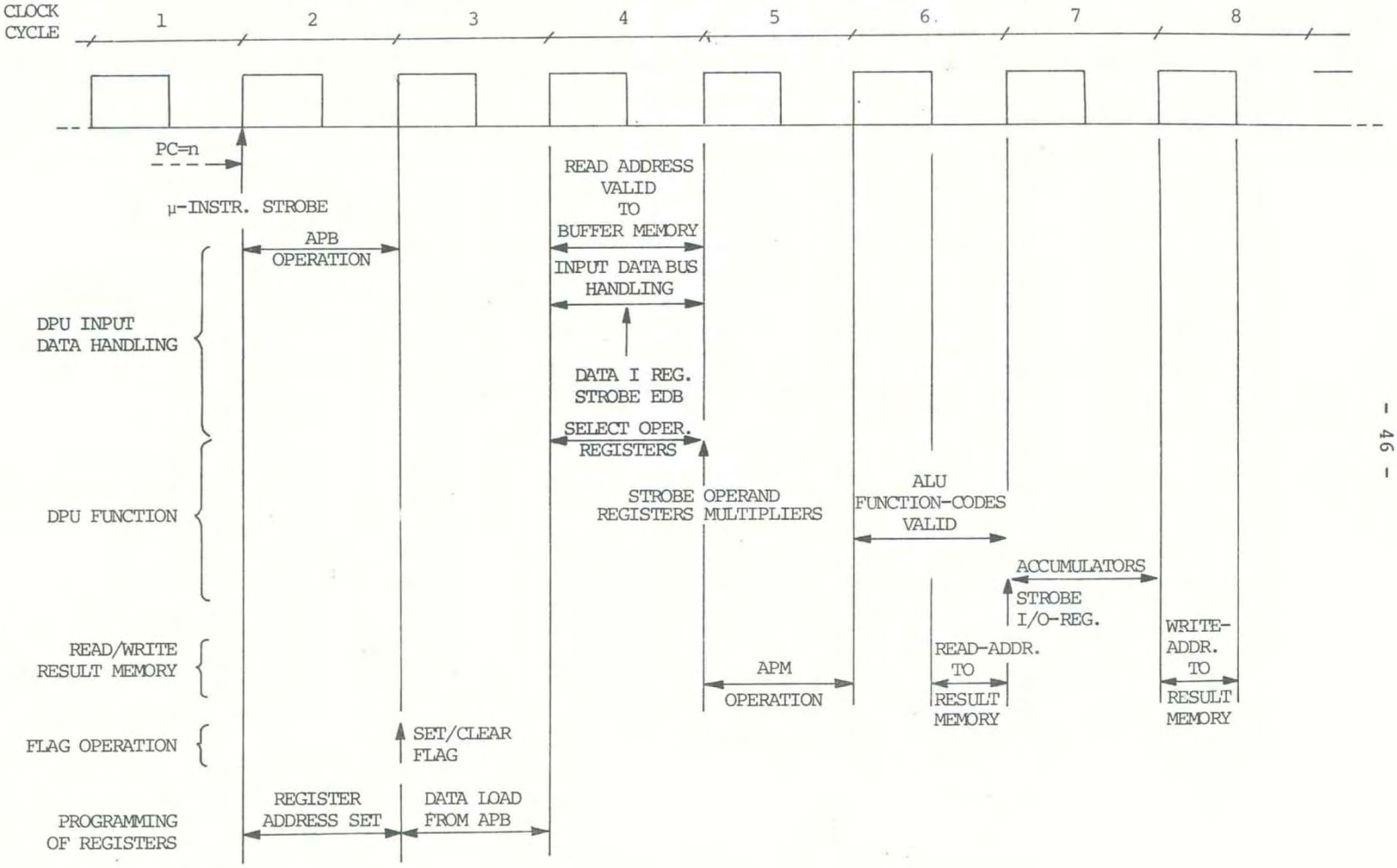


Figure 18. CORRELATOR TIME DIAGRAM FOR ONE PROGRAM CYCLE.

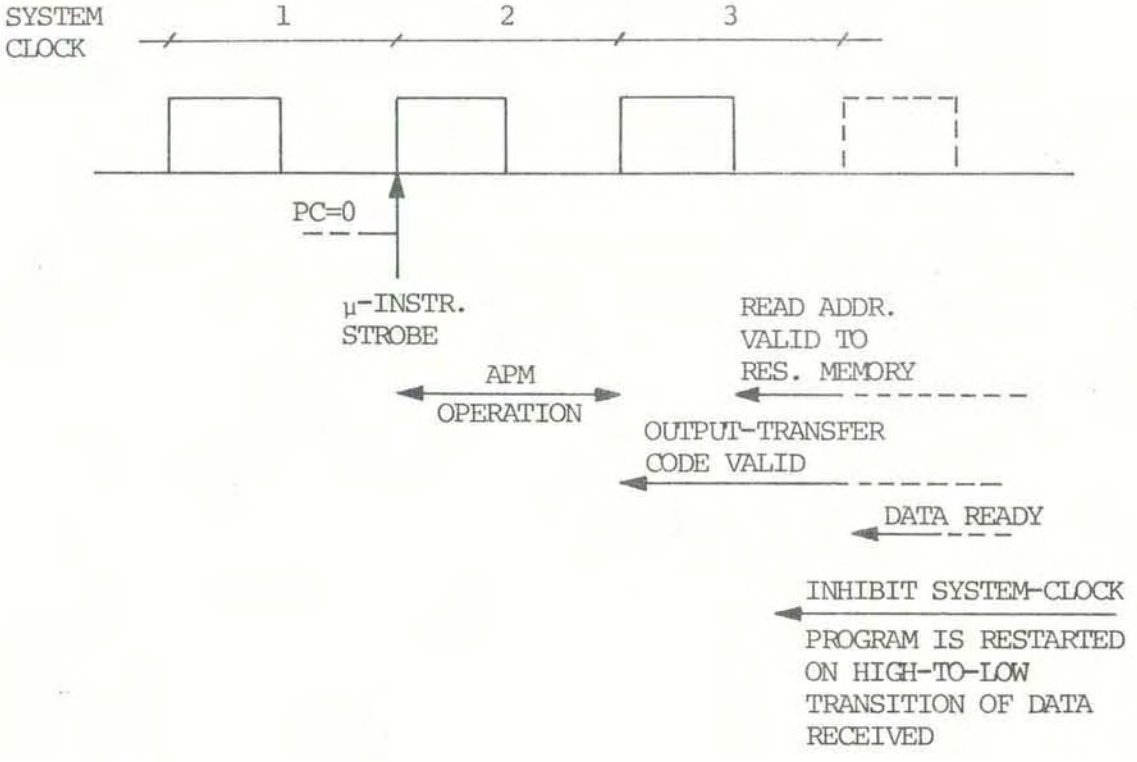


Figure 19. TIME DIAGRAM FOR OUTPUT TRANSFER (T_1,T_2 = 1).

V. EXAMPLES OF PROGRAMMING.

MICRO-PROGRAM FOR DIGITAL CORRELATOR

Author: HANS-JØRGEN ALKER

Date: 13/12-77

Program name: POWER-PROFILE File-name (NORD-10): -

Program version: I

Functional description:

DATACHANNEL 1: ZERO-LAG ESTIMATION $K = \sum_{i=0}^{N-1} (x_i^2+y_i^2)$

DATACHANNEL 2: MEAN-VALUE ESTIMATION $m = \sum_{i=0}^{N-1} (x_i+y_i)$

MINIMUM NUMBER OF SAMPLES IN RANGEDATA: 2

Start-address for program: 01_8

Program-memory locations used: 01_8 - 05_8

Parameters stored:

Register name	Register address		Parameter
SAR	00100	-	START ADDRESS PROGR.
LCR1	10010	-	NO. OF SAMPLES -2 IN RANGEGATE
LCR2	10011	-	NO. OF RANGEGATES -1 IN RADAR-SCAN
APB RS:12	10000	1100	START-ADDR -D_1 BUFFER
APB RS:11	10000	1011	SAMPLE INCREMENT "
APB RS:10	10000	1010	RANGEGATE INCR. D_1 "
APB RS:9	10000	1001	USED FOR TEMPORARY STORAGE
APM RS:12	10001	1100	START-ADDR - K_1 RES MEM
APM RS:11	10001	1011	INCREMENT K_1 " "
APM RS:10	10001	1010	USED FOR TEMPORARY STORAGE

Flag operation: NOT USED

MICRO-PROGRAM FOR DIGITAL CORRELATOR
PROGRAM-MEMORY MODULE 0, 1 AND 2

PROGRAM NAME: POWER-PROFILE
PROGRAM VERSION: I

RAM-MODULE 2 (ADDR. 1010) | RAM-MODULE 1 (ADDR. 1001) | RAM-MODULE 0 (ADDR. 1000)

PROGRAM LOCATION	JUMP-ADDRESS						LOOP-COUNTERS								FLAG		APB	MICRO-PROGRAM SEQUENCER												APB																					
							LCR1A	LC3			LC2			LC1				MODE-CONTROL				LC3			LC2			LC1			INSTR. CODE B					INSTR. CODE A				B-ADDR.				A-ADDR.				ALU DESTIN.			ALU FUNCTION			ALU SOURCE		
															SET	RESET	Select																																				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0	1	1	0	0	0	1	1	0	0
2	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	1	0	1
3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	1	0	0	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1
4	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	1	1	1

MICRO-PROGRAM FOR DIGITAL CORRELATOR
PROGRAM-MEMORY MODULE 3, 4 AND 5

PROGRAM NAME: POWER-PROFILE
PROGRAM VERSION: I

PROGRAM LOCATION	RAM-MODULE 5 (ADDR. 1101)																RAM-MODULE 4 (ADDR. 1100)																RAM-MODULE 3 (ADDR. 1011)															
	MULT. 2 SELECT			MULTIPLIER 1 SELECT					MULT. OPERAND-REG. STROBE								INPUT-BUS				MULTIPLIER 4 SELECT					MULTIPLIER 3 SELECT					M. 2 Select		NOOP	ACCUMULATORS							Transf. select	NOOP	REDEF. OF REGISTERS					
	A			B		A			M4 B	M4 A	M3 B	M3 A	M2 B	M2 A	M1 B	M1 A	Select	Strobe	EDB	EAB	B		A			B		A			B			Strobe	Write	Read	FF1 Re-set	FF1 Set	FF2 Re-set	FF2 Set			ADDRESS-CODE					Enable
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0

MICRO-PROGRAM FOR DIGITAL CORRELATOR
PROGRAM-MEMORY MODULES 6 AND 7

PROGRAM NAME: POWER-PROFILE
PROGRAM VERSION: I

	RAM-MODULE 7 (ADDR. 1111)																RAM-MODULE 6 (ADDR. 1110)																	
PROGRAM LOCATION	APM								ALU OPERATIONS								OUTPUT TRANSFER								ADM									
	ALU DEST.		ALU FUNCTION			ALU SOURCE			M1, M2 1	2	3	4	M3, M4 1	2	3	4	D.R.			SELECT CODE				IN U.	B-ADDRESS				A-ADDRESS				ALU D
0	0	1	1	0	1	1	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0
2	1	1	0	0	0	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	0	0
3	0	1	1	0	1	1	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	0	0	0	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
5	0	1	1	0	1	1	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MICRO-PROGRAM FOR DIGITAL CORRELATOR

Author: HANS-JØRGEN ALKER

Date: 16/9-77

Program name: AUTOCORRELATION **File-name (NORD-10):** -

Program version: I

NO. OF LAG-PARAMETERS (COMPLEX) EQUAL TO NO. OF SAMPLES IN RANGEGATE

Functional description:

DATA-CHANNEL 1: $\mathrm{Re}\{K_e\} = \sum_{i=0}^{N-1-e} (x_i x_{i+e} + y_i y_{i+e})$

DATA-CHANNEL 2: $\mathrm{Im}\{K_e\} = \sum_{i=0}^{N-1-e} (x_i y_{i+e} - x_{i+e} y_i)$

Start-address for program: 06_8

Program-memory locations used: $06_8 - 13_8$

Parameters stored:

Register name	Register address	Parameter
SAR	00100 -	START ADDRESS PROGR.
LCR1	10010 -	NO. OF SAMPLES -1 IN RANGEGATE
LCR2	10011 -	NO. OF RANGEGATES -1 IN RADAR SCAN
APB RS:15	10000 1111	START-ADDR - D_1 BUFFER
APB RS:14	10000 1110	SAMPLE INCR. "
APB RS:13	10000 1101	RANGEGATE INCR D_1 "
APB RS:9	10000 1001	USED FOR TEMPORARY STORAGE
APM RS:15	10001 1111	START-ADDR - K_1 RES MEM
APM RS:14	10001 1110	INCREMENT K_1 " "
APM RS:10	10001 1010	USED FOR TEMPORARY STORAGE

Flag operation:

MICRO-PROGRAM FOR DIGITAL CORRELATOR
PROGRAM-MEMORY MODULE 0, 1 AND 2

PROGRAM NAME: AUTOCORRELATION
PROGRAM VERSION: I

PROGRAM LOCATION	RAM-MODULE 2 (ADDR. 1010)																RAM-MODULE 1 (ADDR. 1001)																		RAM-MODULE 0 (ADDR. 1000)																	
	JUMP-ADDRESS						LOOP-COUNTERS								FLAG		APB	MICRO-PROGRAM SEQUENCER														APB																	
							LCR1A	LC3		LC2		LC1			SET	RESET	Select	MODE-CONTROL			LC3	LC2	LC1	INSTR. CODE B				INSTR. CODE A				B-ADDR.				A-ADDR.				ALU DESTIN.			ALU FUNCTION			ALU SOURCE		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1
6	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	0	1	1	0	0	0	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	1	1	1	0	1	0	1	1	0	0	0	0	0	1
8	0	0	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
10	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	1	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	1

MICRO-PROGRAM FOR DIGITAL CORRELATOR
PROGRAM-MEMORY MODULE 3, 4 AND 5

PROGRAM NAME: AUTOCORRELATION
PROGRAM VERSION: I

RAM-MODULE 5 (ADDR. 1101)

PROGRAM LOCATION	MULT. 2 SELECT A			MULTIPLIER 1 SELECT B		MULTIPLIER 1 SELECT A			MULT. OPERAND-REG. STROBE M4 B	M4 A	M3 B	M3 A	M2 B	M2 A	M1 B	M1 A
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1
8	0	0	1	0	0	0	0	0	1	0	1	0	1	0	1	0
9	0	0	1	0	0	0	0	0	1	0	1	0	1	0	1	0
10	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RAM-MODULE 4 (ADDR. 1100)

PROGRAM LOCATION	INPUT-BUS Select	Strobe	EDB	EAB	MULTIPLIER 4 SELECT B		MULTIPLIER 4 SELECT A			MULTIPLIER 3 SELECT B		MULTIPLIER 3 SELECT A			M. 2 Select B	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
8	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
9	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
10	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

RAM-MODULE 3 (ADDR. 1011)

PROGRAM LOCATION	NOOP	ACCUMULATORS Strobe	Write	Read	FF1 Re-set	FF1 Set	FF2 Re-set	FF2 Set	Transf. select	NOOP	REDEF. OF REGISTERS ADDRESS-CODE					Enable
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
7	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
8	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
10	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
11	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0

MICRO-PROGRAM FOR DIGITAL CORRELATOR
PROGRAM-MEMORY MODULES 6 AND 7

PROGRAM NAME: AUTOCORRELATION
PROGRAM VERSION: I

	RAM-MODULE 7 (ADDR. 1111)																RAM-MODULE 6 (ADDR. 1110)																	
PROGRAM LOCATION	APM								ALU OPERATIONS								OUTPUT TRANSFER								ADM									
	ALU DEST.		ALU FUNCTION			ALU SOURCE			M1, M2 1	2	3	4	M3, M4 1	2	3	4	D.R.			SELECT CODE			IN U.		B-ADDRESS				A-ADDRESS				ALU D
0	0	1	1	0	1	1	1	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	1	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	0		
7	1	1	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0		
8	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0		
9	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0		
10	1	1	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0		
11	0	1	1	0	1	1	1	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

MICRO-PROGRAM FOR DIGITAL CORRELATOR

Author: HANS-JØRGEN ALKLR

Date: 16/9-77

Program name: AUTOCORRELATION File-name (NORD-10): -

Program version: II

Functional description:

- SAME AS VERSION I, BUT NO. OF LAGS CAN BE LESS THAN NO. OF SAMPLES IN RANGEGATE

Start-address for program: 14_8

Program-memory locations used: 14_8 - 21_8

Parameters stored:

Register name	Register address		Parameter
SAR	00100	-	START-ADDRESS PROGR.
LCR1	10010	-	NO. OF SAMPLES-1 IN RANGEGATE
LCR2	10011	-	NO. OF RANGEGATES-1 IN RADAR SCAN
LCR3	10100	-	NO. OF LAG-PAR.-1 IN RANGEGATE
APB RS:15	10000	1111	START-ADDR-D_1 BUFFER
APB RS:14	10000	1110	SAMPLE INCR. "
APB RS:13	10000	1101	RANGEGATE INCR. D_1 "
APB RS:9	10000	1001	TEMPORARY STORAGE
APM RS:15	10001	1111	START-ADDR-K_1 RES MEM
APM RS:14	10001	1110	INCREMENT K_1 " "
APM RS:10	10001	1010	TEMPORARY STORAGE

Flag operation: NOT USED

MICRO-PROGRAM FOR DIGITAL CORRELATOR
PROGRAM-MEMORY MODULE 0, 1 AND 2

PROGRAM NAME: AUTOCORRELATION
PROGRAM VERSION: II

	RAM-MODULE 2 (ADDR. 1010)							RAM-MODULE 1 (ADDR. 1001)							RAM-MODULE 0 (ADDR. 1000)				
	JUMP-ADDRESS	LOOP-COUNTERS				FLAG		APB	MICRO-PROGRAM SEQUENCER						APB				
PROGRAM LOCATION		LCR1A	LC3	LC2	LC1	SET	RESET	Select	MODE-CONTROL	LC3	LC2	LC1	INSTR. CODE B	INSTR. CODE A	B-ADDR.	A-ADDR.	ALU DESTIN.	ALU FUNCTION	ALU SOURCE
0	0 0 0 0 0 0	0	0 0	0 0	0 0 0	0	0	0	1 0 0	0	0	0	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	0 0 1	1 0 1	1 1 1
12	0 0 0 0 0 0	0	1 0	1 1	0 1 0	0	0	0	1 0 0	0	0	0	0 0 0 0	1 0 0 0	1 0 0 1	1 1 1 1	0 1 1	0 0 0	1 0 0
13	0 0 0 0 0 0	0	0 1	0 0	1 0 0	0	0	0	1 1 1	1	0	1	0 1 0 1	1 0 0 0	1 0 0 1	1 1 0 1	0 1 1	0 0 0	0 0 1
14	0 1 0 0 0 0	1	1 1	0 0	1 0 1	0	0	0	1 1 1	1	0	1	0 1 1 0	1 0 1 0	1 0 0 1	1 1 0 0	0 0 0	0 0 0	0 0 1
15	0 0 0 0 0 0	0	1 1	0 0	1 0 1	0	0	0	1 1 1	1	0	1	0 0 0 0	0 1 0 1	0 0 0 0	1 1 1 0	0 0 0	0 0 0	0 0 0
16	0 0 1 1 0 1	0	1 1	0 0	1 0 0	0	0	0	0 0 0	0	1	1	0 1 0 1	0 0 1 0	1 0 0 1	1 1 1 0	0 1 1	0 0 0	0 0 1
17	0 0 0 0 0 0	0	0 0	0 0	0 0 0	0	0	0	0 1 0	0	0	0	0 0 0 0	0 1 1 0	0 0 0 0	0 0 0 0	0 0 1	1 0 1	1 1 1

RAM 3, 4, 5, 6 AND 7 ARE EQUAL TO PROGRAM VERSION I EXCEPT FOR PROGRAM LOCATION

MICRO-PROGRAM FOR DIGITAL CORRELATOR

Author: HANS-JØRGEN ALKER

Date: 21/9-77

Program name: AUTOCORRELATION **File-name (NORD-10):** -

Program version: III

Functional description:

. FOR MULTIPLE-PULSE TRANSMISSIONS.
MAX. NO. OF ELEMENT-PULSES IN PULSE-GROUP: 12

Start-address for program: 22_8

Program-memory locations used: $22_8 - 27_8$

Parameters stored:

Register name	Register address		Parameter
SAR	00100	-	START-ADDRESS PROGR.
LCR1	10010	-	NO. OF SAMPLES (PULSES) -1 IN RANGEGATE
LCR2	10011	-	NO. OF RANGEGATES-1 IN RADAR SCAN
APB RS:15	10000	1111	START-ADDR-D_1 BUFFER
APB RS:14	10000	1110	SAMPLE INCR. "
APB RS:13	10000	1101	RANGEGATE INCR. D_1 "
APB RS:12	10000	1100	TEMPORARY STORAGE
⋮	⋮	⋮	⋮
APB RS:1	10000	0001	SAMPLE DIST. 2. LAST - 3. LAST PULSE
APM RS:0	10000	0000	SAMPLE DISTANCE BETWEEN LAST AND 2. LAST PULSE
APM RS:15	10001	1111	START ADDR-K_1 RES MEM
APM RS:14	10001	1110	INCREMENT K_1 " "
APM RS:10	10001	1010	TEMPORARY STORAGE

Flag operation: NOT USED.

MICRO-PROGRAM FOR DIGITAL CORRELATOR
PROGRAM-MEMORY MODULE 0, 1 AND 2

PROGRAM NAME: AUTOCORRELATION
PROGRAM VERSION: III

	RAM-MODULE 2 (ADDR. 1010)																	RAM-MODULE 1 (ADDR. 1001)																				RAM-MODULE 0 (ADDR. 1000)																			
PROGRAM LOCATION	JUMP-ADDRESS						LCR1A	LOOP-COUNTERS LC3		LC2		LC1			FLAG SET	RESET	APB Select	MICRO-PROGRAM SEQUENCER MODE-CONTROL		LC3	LC2	LC1	INSTR. CODE B				INSTR. CODE A				APB B-ADDR.				A-ADDR.				ALU DESTIN.			ALU FUNCTION			ALU SOURCE			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1
18	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	1	1	1	0	1	1	0	0	0	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	0	1	1	0	0	1	1	0	1	0	1	1	0	0	0	0	0	1
20	0	1	0	1	1	0	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1
21	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1
22	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	

RAM 3, 4, 5, 6 AND 7 ARE EQUAL TO PROGR. VERSION I EXCEPT FOR PROGR. LOCATION

EISCAT SCIENTIFIC ASSOCIATION
S-981 01 KIRUNA 1, SWEDEN
TELEPHONE 0980/187 40
TELEX 8754 GEOFYSK S