RAYTHEON

FORM 10-0557 (9-65) BOND

Classification UNCLASSIFIED

DIVISION MISSILE SYSTEMS
Operation Advanced Systems Center
Department Tactical Systems

Contract No.

Distribution aa

To Alan J. Deerfield (S2-29)

File No.

From Steven J. Wallach (S2-29)

Memo No. AADC-75-12

Subject AADC APL MICROPROGRAMMED INTERPRETER

Date 14 May 1975

## 1.0 INTRODUCTION

The AADC DPE directly supports via microcode a subset of APL. The interpreter resides in a 756 x 110 bit microprogrammed control store. The 110 bit wide ROM word structure allows for highly parallel computational algorithms. Aiding the interpreter is a bank of scratchpad registers (16 scratchpads, each 16 bits in length). These scratchpad locations are used to maintain: address pointers for accumulator and memory operand manipulations, and operand counts to control the number of iterations for the various algorithms.

## 2.0 INTERPRETER OVERVIEW

The interpreter is comprised of three basic components: prologue, execution sequence, and epilogue.

### 2.1 Prologue

Before performing the execution sequence of the APL primitives, each opcode invokes generalized subroutines to ensure proper setup of the DPE. These subroutines are invoked from a master subroutine (SDYAD) to minimize total microcode storage.

The setup routines invoked and their use are:

- SBSTP - Set up various internal indicators relating to the B (memory) operand. These include:
  - SP[21:] - address of the first B operand.

- ° SP[20:] - the internal form of the row/column part of the B dimension word. Internally all operands are basically treated as matrices. If B has 2 rows, 4 columns, SP[20:] denotes 1 row, 4 columns. In this way a scalar is denoted by 0 rows, 1 column).

- ° BELF - denotes if B is null.

- ° STORES - a scalar B operand and the dimension word in location 255,254 and 255,255 (double scalar - dimen not stored if B is popped).

- ° SP[29:] - saves P away.

- ° SP[17:] - 255,254 to indicate an available temporary location if needed by the execution sequence.

- SBMO - This ensures that the DPE is idle (in a Wait state).

- Loads IA with the A dimension word, IB with the B dimension word, P cleared to zero, and sets EQHF and EQLF (Arithmetic Unit compare latches). This last action is done so that the BROM Macro can be called upon the return from the master "SDYAD" routine.

  Other function modules of the prologue called as needed are:

  - ° compare the A and B operands for conformal data types. (Complex-Complex e.g., Catenation or Single B e.g., Rotate).

  - ° SPSH - push the opcode and B operand.

  - ° SUN - unpack routine.

- SACST - Accumulator set up.

- ° SP[24] and SP[25] to point to the next accumulator area (A Other) and if necessary stores a real, scalar A in a memory accumulator with a subsequent setting of SP[19] and SP[22] with the starting address of the accumulator. Also, the APQ is cleared.
- ° SP[16] set with 0 rows, 1 column if A is a scalar.

- SPOP - The SPOP routine is entered when the numeric opcode class (e.g., Outer product) is executed when the opcode was just popped.

## 2.2 Execution Sequence

Every opcode's execution sequence is partitioned into two parts:

- Answer dimension generation and semantical correctness.
- Answer generation.

### 2.2.1 Answer Dimension Generation

In this phase the result answer dimension is calculated and placed in SP[18]. Length and Domain checking is also performed. For the opcodes that begin with hex 6, the special Decode branch is the major tool used to perform this action.

### 2.2.2 Answer Generation

All blockmode instructions generate answers using the AP. The blockmode instructions are partitionable into two sets: numerics and non-numerics. The numerics involve performing AP opcodes as specified by the LOAD OPCODE instruction or the generalized dyadic form (this includes transfers, compares, element by element operations, etc.) The non-numerics generally involve memory to memory moves (under precision control) according to the opcode (e.g., Catenation, Take).

The numerics use the SOX routine to route data from the memory to the answer accumulator. The SOX routine commands the AP to execute the opcode indicated by opcode register 1 or 2 (determined by the state of BMSL). SOX has the following conventions:

P is used to read the B operand and points to the next B operand after the return.

IB is used to read the A operand and points to the next A operand after the return.

IA points to the storage address. IA points to its initial value upon return.

The return is to STC high bank. ST2 and ST3 are not used.

The APQ read address points to its initial value upon entry.

SP[30:] is conditionally incremented (full 16 bits) based on P1FF (1-increment). Upon return ADZF is conditioned by SP[30:].

SP[16:] is unconditionally incremented.

The SOX routine may be entered with the B operand already loaded by performing the branch → 892+BCMP (SOX3+∧/BDT). This assumes that B is already loaded in the APQ. The SOX routine loads M with the contents of the APQ and A via IQ.

The non-numerics use the SAP routine to move data from memory to the answer accumulator. The SAP routine is used solely to move data subject to precision control. It accomplishes this by the AP M-register loading from the APQ and then commanding the AP to execute a Blockmode Load. The conventions of SAP are:

P points to the B operand. Upon return P points to the next B operand.

IB points to the answer area. Upon return IB points to the next answer accumulator.

The return is via STC. ST2 and ST3 are unused.

IA is incremented by 1 (full 16 bits). Upon return ADZF is conditioned by IA.

Scratchpad 30 begins reading. Thus SP30 may be used for arithmetic operations upon completion.

The entry points SAPO (RSS 813) assumes AR is set up with P. SAP1 (RSS 782) assumes no AR setup. SAP2 assumes the APQ already has the B operand stored away.

## 2.3 Ending Routine

All Blockmode instructions, upon completion of their execution phase, transfer control to the SEND routine. The SEND routine generates a dimension word to be stored into the answer area, updates scratchpads 16, 19, 22, and 29. If a parenthetical pop is indicated, the SEND routines initiates actions to perform the pop. If no pop is indicated, the next sequential instruction is fetched and executed.

Steven J. Wallach
Ext. 4593

/bgf
dc: E. Aaronson S2-67
S. Nissen S4-75
B. Tanenbaum S2-29