AERE - R 9360

United Kingdom Atomic Energy Authority

HARWELL

Using the Am9511 Arithmetic Processing Unit with the Motorola M6800 Microprocessor

M.A. Reid and D.A. Newton
Instrumentation and Applied Physcis Division
AERE Harwell, Oxfordshire
February 1979

PRICE £2.00 net from H.M. Stationery Office

AERE R-9360

Using the Am9511 Arithmetic Processing Unit with the Motorola M6800 Microprocessor

M.A. Reid and D.A. Newton

Summary

A circuit for interfacing the Am9511 Arithmetic Processing Unit to a Motorola M6800 Microprocessor is described, together with a software interface which makes the use of the arithmetic unit by applications programs particularly simple. Examples are given of its use, together with execution timings. A description is given of test software which permits each function of the Am9511 to be operated from the control console of a microcomputer development system.

Instrumentation and Applied Physics Division,
A.E.R.E., Harwell.

February 1979

HL.79/561 (C13)

Contents

ISBN 0-70-580850-5

Introduction

Microcomputers are now widely applied in laboratory and process control instrumentation, in which they are usually required to carry out control and supervision tasks, as well as to perform data aquisition and reduction.

General purpose microprocessors are well suited to the control and supervision requirements because their instruction sets contain many logical and control operations, but they are restricted in their performance of data acquisition and reduction by their poor range of arithmetic instructions; the Motorola M6800 microprocessor for example, offers only eight bit add and subtract arithmetic instructions.

The provision of further arithmetic operations, and their extension to 16 and 32 bit fixed point and 32 bit floating point representation may be achieved either by using software arithmetic packages, known as 'maths packs', or by adding a dedicated arithmetic processing unit to the microcomputer. The software arithmetic packages enable microcomputers to carry out many useful data acquisition and reduction operations, but are too slow for use where significant amounts of real time data processing are required, for example in the preparation of information for graphics displays. The performance of arithmetic devices of the type used in hand calculators is also insufficient for such applications, and in addition they are usually restricted to decimal representation of variables, with serial input and output characteristics which complicate their interfacing to microprocessors.

A dedicated arithmetic device has recently been introduced however, which both has high performance and is easy to interface to many microprocessors. The Am9511 Arithmetic Processing Unit (APU) from Advanced Micro Devices (1) carries out 16 and 32 bit fixed point arithmetic operations, and 32 bit floating point arithmetic and trigonometric functions together with appropriate format interconversions; it offers up to x100 improvements in execution times over software arithmetic routines for comparable operations. It is designed to be compatible with the Intel 8080 and AMD 9080 microprocessor data and control bus standards, but can be interfaced to other microprocessors by appropriate circuits and software techniques.

This report describes a circuit for interfacing the Am9511 APU to a Motorola M6800 Microprocessor(section 2),together with a set of M6800 assembler language software drivers which permit the APU to be used by applications programs for arithmetic operations, without requiring of users a knowledge of its detailed operations(section 3). Examples of the use of the programs are given, together with typical timings for operations executed through the software drivers.

This placing of software drivers between the application program and the APU makes the use of the device straightforward, but adds an execution time penalty to all operations, and also prevents advantage being taken of the detailed properties of the APU to achieve the highest possible calculation speeds in chained operations. Therefore a description is also given (section 4) of the direct use of the APU by an application program, and examples are given of the techniques which may be employed to achieve high calculation throughputs in chained operations where required by applications.

Test software is also described in section 5, which permits a user to exercise each function of the APU from the control console of a microcomputer development system, in order to establish the correct functioning of the software drivers with the APU.

2. Interface Circuit

2.1 Interfacing Requirements of the Am9511

The Am9511 APU is a MOS LSI device which provides high performance fixed and floating point arithmetic and floating point trigonometric operations. It consists of a 16 bit processor with a stored control program, and a user accessible data stack, command register and status register.

An external processor uses the APU for calculations by loading data into the data stack and loading a command into the command register of the APU, which then operates on the data as directed by the command (section 3.3). The status of the APU during its operations is indicated by appropriate bit settings in its status register, as well as by signals on its interface control lines. Results are retrieved from the stack after the completion of a calculation.

The APU interface circuit must therefore provide for reading and writing to the data stack, reading the status register and writing into the command register, as well as servicing the APU interface control lines. The hardware interfacing model of the APU is shown in Fig. 1. All transfers, including data, results, status and command information take place over the 8 bit bidirectional data bus, and are managed by the bus control signals. Table 1 gives the transfers which can be carried out as a function of the bus control signals.

Table 1

APU Transfer Operations

$\overline{\text{SELECT}}$	COMMAND/$\overline{\text{DATA}}$	$\overline{\text{READ}}$	$\overline{\text{WRITE}}$	Operation
1	X	X	X	No operation
0	0	1	0	Enter Data into Stack
0	0	0	1	Read Data from Stack
0	1	1	0	Load Command Register
0	1	0	1	Read Status Register

X = don't care ; 0 = logic 0 ; 1 = logic 1

The data sheets (2) of the Am9511 give the timing of each of these operations. The $\overline{\text{PAUSE}}$ signal is generated by the APU to control the timing of bus transfers by delaying the operation of the external processor during each transfer until the APU is ready. Its use is discussed in more detail in section 2.2.

The interface control signals govern the operation of the APU and enable it to be used in processor interrupt and direct memory access (DMA) structures. The clock input is a continuous signal which drives the APU in all of its operations. The Reset input sets the APU to a known state, and is usually activated during the power up sequence of a system. The completion of an operation by the APU is accompanied by the $\overline{\text{END}}$ signal, which can be used to interrupt the external processor, and also if required by the Service Request signal (SVREQ) which is intended for initiating the transfer of results under the control of a DMA controller to achieve the highest possible data transfer rate. The $\overline{\text{ENDACK}}$ and $\overline{\text{SVACK}}$ inputs acknowledge and clear the $\overline{\text{END}}$ and SVREQ outputs respectively.

These bus control, data bus, and interface control signals are designed to be compatible with the Intel 8080 and AMD 9080 microprocessors, and suitable connections are given by AMD(2). They are however, not all

compatible with other microprocessors, and interface circuits are usually required to match the APU to them. Appropriate circuits for the M6800 microprocessor are given in sections 2.2 and 2.3.

2.2 M6800 Microprocessor Interfacing Considerations

The control and data bus lines of the M6800 microprocessor are either directly compatible with those of the Am9511, or else require only simple interfacing to become compatible, with the exception of the $\overline{\text{PAUSE}}$ output of the Am9511. There is no input on the M6800 which corresponds to the RDY input of the Intel 8080 or AMD 9080, which is the usual connection for the $\overline{\text{PAUSE}}$ line. There is however, an input to the MC6875 microprocessor clock driver which enables slow devices to lengthen a M6800 processor bus cycle, and hence permit data transfers to take place with slow memory or peripherals. The use of this input of the MC6875 (Memory Ready) is shown in Fig. 2; the essential requirement is that the Memory Ready line is brought low by the accessed slow device before the rising edge of the processor phase two clock. The clock line states are then extended in increments of nominal bus cycles for as long as the Memory Ready line is low: this input is sampled at each increment boundary. The processor completes the data transfer one nominal bus cycle after the Memory Ready line has been sampled in the high state again. This input is therefore a suitable connection for the $\overline{\text{PAUSE}}$ output of the Am9511 APU.

There is a restriction however, on the maximum time for which the M6800 clock line states may be prolonged, because of the dynamic nature of the M6800 circuits: this time is 4.5 μsec. The Am9511 bus transfer timings (2) show that most transfers are completed within 2.5 μsec, thus satisfying this restriction; however attempts to access the command register or data stack of the Am9511 during a calculation will bring the $\overline{\text{PAUSE}}$ output low for the rest of the calculation (tens or hundreds of μsec), thus destroying all information within the M6800. It is important therefore, to access only the status register of the Am9511 from a M6800 whilst a calculation is in progress; this state of the APU is detectable by bit settings in its status register, thus enabling data and command transfers to be restricted to the not busy state.

2.3 Circuit Description

The circuit developed for interfacing the Am9511 to the M6800 processor, and meeting the criteria of sections 2.1 and 2.2 is shown in

Fig. 3. It is intended to be compatible with the Motorola EXORcisor(3) microcomputer development system, and therefore has control and data lines buffered by IC5,6,7 from the connections to the EXORcisor bus, SK1. The $\overline{\text{SELECT}}$ line is activated by an address map decoder elsewhere in the system for addresses E7ØØ to E7Ø7 hexadecimal.

All transfers to the Am9511 are intended to take place by programmed I/O for simplicity, hence no use is made of the SVREQ and $\overline{\text{SVACK}}$ DMA connections; furthermore, the standard form of the circuit does not use the $\overline{\text{END}}$ signal to interrupt the M6800 for the same reason. However, provision is made (link LK1) for this feature to be implemented if required by an application.

The 2-4 line decoder IC2 controls bus transfers in conjunction with address line A2 of the M6800, to implement the truth table in table 1. The $\overline{\text{SELECT}}$ and M6800 READ/$\overline{\text{WRITE}}$ lines are used as the address inputs of IC2, which is enabled by the inverted phase two clock; this achieves the required $\overline{\text{READ}}$ and $\overline{\text{WRITE}}$ delays from $\overline{\text{SELECT}}$ (2) The $\overline{\text{READ}}$ signal is also used to control the direction of data flow through the bidirectional buffers IC6, IC7.

The $\overline{\text{PAUSE}}$ output of the Am9511 goes low after the start of a read or write access (2), hence will not be low at the rising edge of phase two and therefore cannot serve as a direct connection to the Memory Ready line in order to stretch the M6800 clock inputs (Fig. 2). The $\overline{\text{SELECT}}$ line and phase two are therefore used to bring Memory Ready low before the rising edge of phase two (IC4-1, IC3-4 and IC5 pin 15) and start the M6800 clock states stretching (Fig 4 point A); $\overline{\text{PAUSE}}$ then appears in time (2) to hold Memory Ready low at each subsequent nominal bus cycle boundary (Fig 4, points B-E) and thus control the bus transfer timing. Fig. 4 gives the timing diagram of this operation. Deselection of the Am9511 after a bus transfer causes the Memory Ready output (IC5 pin 11) to go to a high impedance state, thus permitting other slow peripherals or memory to use the Memory Ready line of the MC8675 clock driver.

3. Software Interface

3.1 Introduction

The software interface routines are designed to handle all the data transfers with the Am9511 APU which are required for initiating calculations and retrieving results, whilst observing the restrictions

laid down in section 2.2 on transfers during a calculation; this enables applications programmes to make use of the performance improvement offered by the Am9511 whilst retaining the simple software interface which is characteristic of software arithmetic packages. The formats of the numbers handled by the APU are described in section 3.2, and the functions provided by the interface software program AM952E(Appendices A&B)are detailed in section 3.3 together with their use. Error handling is dealt with in section 3.4 and execution timing in section 3.5. The detailed description of the software interface is given in section 3.6.

3.2 Format of Data

(a) Fixed Point Data

Fixed point numbers may be of 16 or 32 bits length. In each case the most significant bit is a sign bit; 0 = +ve, 1 = -ve. The ranges of values which can be accommodated are:-

16-bit numbers	-32 768 to 32 767
32-bit numbers	$-(2^{31})$ to $2^{31}-1$

(b) Floating Point Data

Floating point numbers are of 32-bits length, consisting of a 24-bit mantissa, and an 8-bit exponent. The binary point is always to the left of the most significant bit of the mantissa, and numbers are normalised so that the mantissa always has a decimal value in the range 0.5 to 1. The exponent consists of six bits plus a sign-bit and the mantissa sign bit. The number structure is shown in Fig 5, together with an example of the representation of π. Numbers in the ranges $\pm$ (0.5×2^{-64} to $(1-5.96 \times 10^{-8}) \times 2^{+63}$), and zero, may be represented .

3.3 Functions Available

The interface program AM952E is structured as a subroutine of an applications program. It provides a number of arithmetic, trigonometric and utility operations; each of these is accessible as a subroutine entry point in AM952E. The arithmetic operations consist of the 32 bit fixed point entries of table 2, and the 32 bit floating point entries of table 3.

Table 2

32 Bit Fixed Point Arithmetic Operations

Operation	AM952E Entry Point Name	Number of Arguments
Addition	DXADD	3
Subtraction	DXSUB	3
Division	DXDIV	3
Multiplication (least significant 32 bits returned)	DXMULL	3
Multiplication (most significant 32 bits returned)	DXMULH	3

Table 3

32 bit Floating Point Arithmetic Operations

Operation	AM952E Entry Point Name	Number of Arguments
Addition	FPADD	3
Subtraction	FPSUB	3
Division	FPDIV	3
Multiplication	FPMUL	3
Square Root	FSQRT	2
y^x	PWRX	3
Log_{10}	LOG10	2
Log_e	LOGE	2
e^x	EXPX	2

The trigonometric operations are all 32 bit floating point format routines, and are given in table 4.

Table 4

Trigonometric Operations

Operation	AM952E Entry Point Name	Number of Arguments
Sine	FPSIN	2
Cosine	FPCOS	2
Tangent	FPTAN	2
Sin^{-1}	FPASIN	2
Cos^{-1}	FPACOS	2
Tan^{-1}	FPATAN	2

The utility operations consist of format conversions given in table 5, and a number of basic manipulations of the Am9511 APU data stack. These are given in table 6.

Table 5

Format Conversions

Conversion	AM952E Entry Point Name	Number of Arguments
16 bit fixed point to 32 bit floating point	CXTFS	2
32 bit floating point to 16 bit fixed point	CFTXS	2
32 bit fixed point to 32 bit floating point	CXTFD	2
32 bit floating point to 32 bit fixed point	CFTXD	2

Table 6

Am9511 Data Stack Operations

Operation	AM952E Entry Point Name	Number of Arguments
Obtain π (= 3.1415925)	PUPI	1
Clear Data Stack	CLEAN	0
Change Sign of Operand Loaded	CSGNF	2
Retrieve Top of Stack	FETCH	1
Retrieve Next on Stack	EXCHF	1

Each function is accessed by a subroutine call from the applications program; this call must be followed by the appropriate number of 16 bit addresses for the function. The program AM952E returns the result (if any) to the locations starting with the last of these 16 bit addresses, and uses the other 16 bit addresses as pointers to any operands required by the function. Tables 2 through 6 give the number of arguments required for each function. The program AM952E returns to the first location after the addresses on completion of the required operation. An example of the use of the program for the calculation of a square root is provided by the following sequence of M6800 assembler code:

```
        JSR FSQRT  ; Invoke square root function
        FDB #ARG1  ; Address of operand
        FDB #ARG2  ; Address for result
           :       ; return here on completion
```

In this example, ARG1 and ARG2 are labels for two 4-byte storage areas. In a similar manner, a floating point division may be carried out by the sequence

```
JSR FPDIV ; invoke divide function
FDB #ARG1 ; address of operand 1
FDB #ARG2 ; address of operand 2
FDB #ARG3 ; address for result
    :     ; return here on completion
```

The order of the operands is important for some functions. In particular, for all subtractions,

Result = operand 1 - operand 2

and for all divisions

Result = operand 1/operand 2

whilst for the power function

Result = y^x ; y = operand 1, x = operand 2.

In addition, the square root and log operands and operand 1 of the power function must always be positive (see table 7), and all trigonometric functions require angles to be expressed in radians. There are also some restrictions on the magnitude of certain arguments; these are

a) EXPX: Operand must lie in range -2^5 to 2^5

b) FPASIN, FPACOS: Operand must lie in range [-1, 1]

c) PWRX: The value | operand 2 x ln (operand 1) | must be $\leqslant 2^5$.

3.4 Error Handling

The APU checks for the occurence of errors during its operations, and sets bits in its status register at the completion of a calculation according to the error detected. The software interface checks for the presence of an error code in the APU after every calculation, and returns to the calling program with the overflow bit (V bit) of the M6800 processor set if one has been detected. The error code is also returned in bits 0-3 of the B accumulator of the M6800 according to table 7.

Table 7

Error Codes

Code	Error
1 0 0 0	Divide by zero
0 1 0 0	Negative argument (SQ. Root, log, ln, y^x)
1 1 0 0	Argument too large ($\sin^{-1}$, $\cos^{-1}$, e^x, y^x)
X X 1 0	Underflow on result
X X 0 1	Overflow on result

Table 8

Typical Execution Timings for Functions

Function	Execution time (μs)
FPADD	350-400
FPSUB	350-400
FPMUL	370
FPDIV	370
FSQRT	600
PWRX	5000
LOG1Ø	2700
LOGE	2600
EXPX	2500
FPSIN	2500
FPCOS	2500
FPTAN	2800
FPASIN	3800
FPACOS	3900
FPATAN	3100
PUPI	200
CSGNF	250
CXTFS	280
CXTFD	300
CFTXS	280
CFTXD	300
EXCHF	200
CLEAN	90
DXADD	350
DXSUB	350
DXMULL	390
DXMULH	390
DXDIV	390
FETCH	200

The development version program, AM953X (Appendices A&B), also prints an error message on the system console which identifies the type and location of an error in addition to returning the error code to the main program as above.

3.5 Execution Timing

The function execution times given in table 8 represent the total time from the initial call to the software interface by the main program, to the return. The times are typical, because execution times are data dependent, particularly with the trigonometric and log-based functions. In general, large operands lead to slightly longer execution times; also certain values of operand cause shorter execution times with the trigonometric functions.

3.6 Software Description

The software interface program AM952E has to carry out the following operations when it is entered from a calling program, e.g. an applications program.

a) Identify function required and generate appropriate command for Am9511 APU.
b) Fetch any operands required by the function and load them into the APU.
c) Load the command into the APU.
d) Wait for the APU to complete its operation.
e) Test for errors.
f) If no error, retrieve result from APU.
g) Indicate error if detected.
h) Return to calling program.

Flowchart 1 (figure 6) gives the overall structure of the program which carries out the above tasks; appendices A and B give the detailed flow-chart and listing of the program, which requires a total of 329 bytes of memory space.

The function required is designated by the entry point used by the calling program; at each entry point the appropriate command for the Am9511 APU is generated and saved in the M6800 B register.

The program then fetches as many operands as required by the function; this is achieved by branching from the entry point to either the 2 operand fetch and load sequence (LOAD42) or the 1 operand sequence (LOAD41). In the cases where no operands are required (table 6), control is transferred

to the appropriate routine. An operand is fetched and loaded, least significant byte first, by appropriate stack and index register operations; in program AM952E, the data stack of the APU is designated by the label INOUT, which points to the address E700 as required by the circuit of section 2. The order of loading the bytes of an operand is important, because the Am9511 is designed to be compatible with the Am9080 microprocessor which handles addresses in the order low byte, high byte, whereas the M6800 handles addresses in the reverse order.

The command stored in the B register of the M6800 is then loaded into locationCOMND, which is a label in program AM952E for address E704, the command and status register address of the circuit of section 2.

The program waits in an idle loop, testing bit 7 of the APU status register, until the calculation is complete. The status register is then examined for error messages, and any error is treated as described in section 3.4. If no error is found, the result is retrieved from the APU and placed at the address indicated in the address sequence of the calling program.

The return to the calling program at the address after the argument list is achieved by discarding the return address on the stack and returning relative to the argument list address.

A listing of a version of the program which is suitable for development use is also given in appendix B as AM953X. This is similar to AM952E, but has an added section of code which prints error messages on the EXORcisor console; these messages identify the type of error and the place in the calling program where the error occurred. This feature aids the development of programs which use the APU and its software interface for arithmetic operations.

4. Direct Use of the Am9511

The software interface program provides a simple and straightforward way for applications programs to use the Am9511 APU. However it adds an execution time penalty of about 200µsec per calculation, and cannot speed up chained calculations by using the data stack of the APU for storage of intermediate results. Details are therefore given in this section of the direct use of the APU from an applications program, and an example is given of performing a chained calculation using the APU data stack for intermediate storage: it is assumed that the hardware interface to the APU

is as given in section 2 i.e. the data stack is at location E700, and the command and status registers are at E704. References 2 and 4 should be consulted when using the APU directly from an applications program.

The sequence of operations for using the APU for one calculation is:-

a) Load operands into the data stack in the desired order (see section 3.3)

b) Load command

c) Wait for completion of calculation

d) Test for errors

e) Retrieve result if needed.

As an example of direct use, consider adding two 32 bit fixed point numbers called ARG1 and ARG2, and returning the result to ARG3, another 32 bit number.

ARG1 is loaded by the sequence

```
LDA A ARG1+3
STA A $E700      ; $ indicates hexadecimal notation
LDA A ARG1+2
STA A $E700
LDA A ARG1+1
STA A $E700
LDA A ARG1       ; note the order of loading bytes
STA A $E700
```

A similar series of statements with ARG2 substituted for ARG1 will load ARG2 on to the APU data stack. The command for 32 bit fixed point addition is 2C (4), hence the operation is initiated by the sequence

```
LDA A #$2C
STA A $E704
```

The wait loop for testing for completion of the addition can be:-

```
LOOP   TST $E704 ;  Read status register and set
       BMI LOOP  ;  condition codes of M6800
                 :  exit from loop when not busy
                 :
                 :
```

Error testing can be combined with the loop, or else can be performed separately thus:-

```
LDA A$E704       ; obtain APU status register
AND A#%00001111  ; mask off unwanted bits
BNE ERROR        ; branch to error if now zero
```

The result may be retrieved by unloading four bytes from $E700 and storing at ARG3.

In expressions involving more than one operation, time may be saved by leaving intermediate results on the data stack of the APU, where they may be used in subsequent operations (4). This technique is illustrated in the calculation

$$y = b^3$$

For brevity, the loading of an operand x (see example above) can be denoted by LDO x, and similarly a command by LDC z, waiting for not busy by BSY, and result retrieval by RTR y. The sequence for calculating $y = b^3$ then becomes

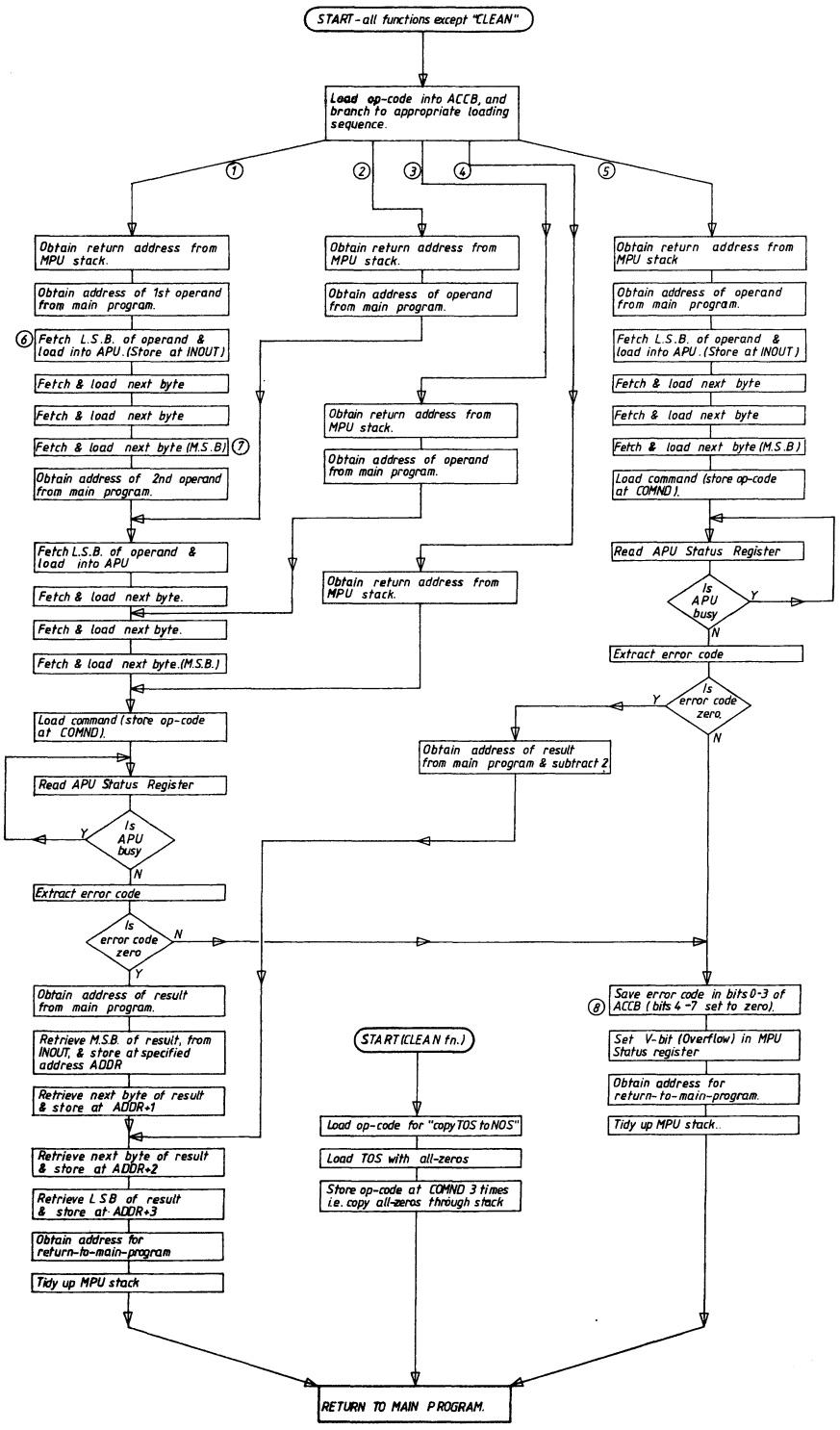
```
LDO  b        ; load operand b into stack
LDC  PTOF     ; copy operand
BSY           ; wait
LDC  PTOF     ; copy operand
BSY           ; wait
LDC FMUL      ; initiate multiplication to form b^2
BSY
LDC FMUL      ; multiply b^2 x b
BSY
RTR y         ; y = b^3
```

The result can be obtained in 320 to 340µsec by this method, against some 700µsec when using the software interface program Am952E. Appendix E gives a further example of a chained operation, the calculation of both roots of a quadratic equation in less than 2.5msec, as against some 8 to 9msec when using the software interface program.

5. Test Software

It is important to provide test software for all hardware and software modules used in computer systems. A test program has therefore been developed which permits a user to verify the operation of the APU and the software interface program from the control console of a microcomputer development system, the Motorola EXORcisor. This check is carried out by permitting the user to select an APU function for execution, and to input data; the test program (AMPRM3, Appendices C, D) passes these data to the software interface, and then returns the result of the calculation

Appendix A FLOWCHART REPRESENTATION OF AM952E PROGRAM



KEY

1. ALL FUNCTIONS REQUIRING 2 4-BYTE OPERANDS
2. " " " 1 " " (except CFTXS)
3. Function CXTFS
4. ALL FUNCTIONS REQUIRING NO OPERANDS.
5. Function CFTXS
6. L.S.B. = least significant byte
7. M.S.B = most " "
8. ACCB = register B of MC6800 MPU

back to the console for verification. The test program is also useful because it illustrates the calling sequence for each function and provides a user with familiarity with the data formats.

The user chooses an APU function for testing by specifying a reference code from a table output by the test program; prompts are then issued for the data required for the function. These data are checked to ensure that only hexadecimal characters or N are input. N causes the function test to be abandoned, and any illegal character causes the data entry to be abandoned, when fresh data may be input.

Operands are entered and results presented in one of the following formats:

a)	16 bit fixed point			NN	NN
b)	32 bit fixed point	NN	NN	NN	NN
c)	32 bit floating point	EE	MM	MM	MM

where NN is a data byte, EE is an exponent byte and MM is a mantissa byte; all bytes are in hexadecimal notation. For example, the decimal number 8191 is entered as either

			1F	FF	(format a)
or	ØØ	ØØ	1F	FF	(format b)
or	ØD	FF	F8	ØØ	(format c)

and -8191 is entered as either

			EØ	Ø1	(format a)
or	FF	FF	EØ	Ø1	(format b)
or	8D	FF	F8	ØØ	(format c)

References

1. Advanced Micro Devices (UK) Ltd.,
   16 Grosvenor Place, London SW1 X7HH

2. AMD Am9511 Data Sheet
   Obtainable from address in (1)

3. 'EXORcisor' is a trade mark of Motorola Ltd.

4. Algorithm Details for the Am9511 APU
   Obtainable from address in (1).

APPENDIX A

Software Interface Flowchart

APPENDIX B(i)

Listing of Program AM952E

```
00001                   ************************************************************
00002                   *PROGRAM FOR USE WITH AM9511 A.P.U.
00003                   ************************************************************
00004                   *
00005                   *GENERAL CALLING SEQUENCE:-
00006                   *
00007                   *  JSR  XXXXX    XXXXX=FUNCTION NAME
00008                   *  FDB  ADD1     ADDRESS OF 1ST OPERAND
00009                   *  FDB  ADD2     ADDRESS OF 2ND OPERAND
00010                   *  FDB  ADDR     ADDRESS OF RESULT
00011                   *  -----         RETURN
00012                   *
00013                   *THERE MAY BE 0,1,OR 2 OPERANDS
00014                   *
00015                   *FUNCTIONS AVAILABLE ARE:-
00016                   *
00017                   * DXADD  ADDITION        (32-BIT FIXED POINT)
00018                   * DXSUB  SUBTRACTION     ( "    " )
00019                   * DXMULL MULTIPLICATION  ( "    " ) RES=LO.HALF
00020                   * DXMULH MULTIPLICATION  ( "    " ) RES=HI.HALF
00021                   * DXDIV  DIVISION        ( "    " )
00022                   * PUPI   CONSTANT PI
00023                   * FPADD  ADDITION
00024                   * FPSUB  SUBTRACTION
00025                   * FPMUL  MULTIPLICATION
00026                   * FPDIV  DIVISION
00027                   * FSQRT  SQUARE ROOT
00028                   * PWRX   POWER  Y**X
00029                   * LOG10  LOG  BASE 10
00030                   * LOGE   LOG  BASE E
00031                   * EXPX   EXPONENTIAL  E**X
00032                   *
00033                   * FPSIN  SINE
00034                   * FPCOS  COSINE
00035                   * FPTAN  TANGENT
00036                   * FPASIN ARCSINE
00037                   * FPACOS ARCCOSINE
00038                   * FPATAN ARCTANGENT
00039                   *
00040                   * CXTFS  CONVERT 16-BIT FX.PT TO 32-BIT FL.PT
00041                   * CFTXS  CONVERT 32-BIT FL.PT TO 16-BIT FX.PT
00042                   * CXTFD  CONVERT 32-BIT FX.PT TO FL.PT
00043                   * CFTXD  CONVERT 32-BIT FL.PT TO FX.PT
00044                   *
00045                   * CSGNF  CHANGE SIGN OF OPERAND LOADED
00046                   * FETCH  RETRIEVE TOP-OF-STACK
00047                   * EXCHF  RETRIEVE NEXT-ON-STACK
00048                   *
00049                   *
00050                   *THE TOP-OF-STACK NUMBER IS ALWAYS RETRIEVED
00051                   *AND STORED AT "ADDR"
00052                   *
00053                   *ALSO:-
00054                   * CLEAN  CLEARS THE A.P.U. STACK
00055                   *
00056                   ************************************************************
```

```
00058                                NAM    AM952E
00059A D800                          ORG    $D800
00060                                OPT    CREF


00062                                ********************************************************
00063                                *FP.APU  PROGRAM ENTRY POINTS
00064                                ********************************************************
00065                                *
00066A D800 C6 1A     A PUPI   LDAB   #$1A      1A IS OPCODE FOR "PI"
00067A D802 30          LDCOM  TSX
00068A D803 EE 00     A         LDX    0,X
00069A D805 09                  DEX
00070A D806 09                  DEX
00071A D807 FF CFF0   A         STX    XKEEP
00072A D80A 20 40 D84C          BRA    STEP1
00073                           *
00074A D80C C6 2C     A DXADD   LDAB   #$2C      ADDITION
00075A D80E 20 6F D87F          BRA    LOAD42
00076                           *
00077A D810 C6 2D     A DXSUB   LDAB   #$2D      SUBTRACTION
00078A D812 20 6B D87F          BRA    LOAD42
00079                           *
00080A D814 C6 2E     A DXMULL  LDAB   #$2E      MULTIPLICATION
00081A D816 20 67 D87F          BRA    LOAD42
00082                           *
00083A D818 C6 36     A DXMULH  LDAB   #$36      MULTIPLICATION
00084A D81A 20 63 D87F          BRA    LOAD42
00085                           *
00086A D81C C6 2F     A DXDIV   LDAB   #$2F      DIVISION
00087A D81E 20 5F D87F          BRA    LOAD42
00088                           *
00089A D820 C6 10     A FPADD   LDAB   #$10      ADDITION
00090A D822 20 5B D87F          BRA    LOAD42
00091                           *
00092A D824 C6 11     A FPSUB   LDAB   #$11      SUBTRACTION
00093A D826 20 57 D87F          BRA    LOAD42
00094                           *
00095A D828 C6 12     A FPMUL   LDAB   #$12      MULTIPLICATION
00096A D82A 20 53 D87F          BRA    LOAD42
00097                           *
00098A D82C C6 13     A FPDIV   LDAB   #$13      DIVISION
00099A D82E 20 4F D87F          BRA    LOAD42
00100                           *
00101A D830 C6 01     A FSQRT   LDAB   #$01      SQ.ROOT
00102A D832 20 46 D87A          BRA    LOAD41
00103                           *
00104A D834 C6 0B     A PWRX    LDAB   #$0B      POWERS
00105A D836 20 47 D87F          BRA    LOAD42
00106                           *
00107A D838 C6 08     A LOG10   LDAB   #$08      LOG 10
00108A D83A 20 3E D87A          BRA    LOAD41
00109                           *
00110A D83C C6 09     A LOGE    LDAB   #$09      LOG E
00111A D83E 20 3A D87A          BRA    LOAD41
00112                           *
```

```
00114A D840 C6 0A    A EXPX   LDAB   #$0A      EXPONENTIAL
00115A D842 20 36 D87A        BRA    LOAD41
00116                         *
00117A D844 C6 00    A FETCH  LDAB   #$00      RETRIEVE TOS
00118A D846 20 BA D802        BRA    LDCOM
00119                         *
00120A D848 C6 19    A EXCHF  LDAB   #$19      RETRIEVE NOS
00121A D84A 20 B6 D802        BRA    LDCOM
00122                         *
00123                         *
00124A D84C 20 6B D8B9 STEP1  BRA    LOADPI
00125                         *
00126                         *
00127A D84E C6 1E    A CFTXD  LDAB   #$1E      FL.PT TO 32-BIT FX.PT
00128A D850 20 28 D87A        BRA    LOAD41
00129                         *
00130A D852 C6 1C    A CXTFD  LDAB   #$1C      32-BIT FX.PT TO FL.PT
00131A D854 20 24 D87A        BRA    LOAD41
00132                         *
00133A D856 C6 15    A CSGNF  LDAB   #$15      CH.SIGN OF OPERAND LOADED
00134A D858 20 20 D87A        BRA    LOAD41
00135                         *
00136                         *
00137A D85A C6 02    A FPSIN  LDAB   #$02      SINE
00138A D85C 20 1C D87A        BRA    LOAD41
00139                         *
00140A D85E C6 03    A FPCOS  LDAB   #$03      COSINE
00141A D860 20 18 D87A        BRA    LOAD41
00142                         *
00143A D862 C6 04    A FPTAN  LDAB   #$04      TANGENT
00144A D864 20 14 D87A        BRA    LOAD41
00145                         *
00146A D866 C6 05    A FPASIN LDAB   #$05      ARCSINE
00147A D868 20 10 D87A        BRA    LOAD41
00148                         *
00149A D86A C6 06    A FPACOS LDAB   #$06      ARCCOSINE
00150A D86C 20 0C D87A        BRA    LOAD41
00151                         *
00152A D86E C6 07    A FPATAN LDAB   #$07      ARCTANGENT
00153A D870 20 08 D87A        BRA    LOAD41
00154                         *
00155A D872 C6 1D    A CXTFS  LDAB   #$1D      C.16-BIT FX.PT TO FL.PT
00156A D874 20 71 D8E7        BRA    KXTFS
00157                         *
00158A D876 C6 1F    A CFTXS  LDAB   #$1F      C.FL.PT TO 16-BIT FX.PT
00159A D878 20 77 D8F1        BRA    KFTXS
00160                         *

00162                         ************************************************************
00163                         *
00164                         *EXTRA ENTRY POINTS MUST BE PLACED BEFORE "PUPI"
00165                         *
00166                         ************************************************************
```

```
00168                    ************************************************************
00169                    *FP.APU  MAIN PROGRAM-LOADING OF OPERAND(S)
00170                    ************************************************************
00171                    *
00172A D87A 30           LOAD41  TSX
00173A D87B EE 00     A          LDX     0,X
00174A D87D 20 21 D8A0           BRA     LOAD4C



00176A D87F 30           LOAD42  TSX
00177A D880 EE 00     A          LDX     0,X
00178A D882 FF CFF0   A          STX     XKEEP
00179A D885 EE 00     A          LDX     0,X

00181A D887 A6 03     A          LDAA    3,X
00182A D889 B7 E700   A          STAA    INOUT
00183A D88C A6 02     A          LDAA    2,X
00184A D88E B7 E700   A          STAA    INOUT
00185A D891 A6 01     A          LDAA    1,X
00186A D893 B7 E700   A          STAA    INOUT
00187A D896 A6 00     A          LDAA    0,X
00188A D898 B7 E700   A          STAA    INOUT

00190A D89B FE CFF0   A          LDX     XKEEP
00191A D89E 08                   INX
00192A D89F 08                   INX

00194A D8A0 FF CFF0   A LOAD4C   STX     XKEEP
00195A D8A3 EE 00     A          LDX     0,X

00197A D8A5 A6 03     A          LDAA    3,X
00198A D8A7 B7 E700   A          STAA    INOUT
00199A D8AA A6 02     A          LDAA    2,X
00200A D8AC B7 E700   A          STAA    INOUT
00201A D8AF A6 01     A IXTFS    LDAA    1,X
00202A D8B1 B7 E700   A          STAA    INOUT
00203A D8B4 A6 00     A          LDAA    0,X
00204A D8B6 B7 E700   A          STAA    INOUT
```

```
00206                    ************************************************************
00207                    *FP.APU MAIN PROGRAM-LOAD COMMANDS AND                     *
00208                    *LOOK FOR ERRORS.  RETRIEVE ANSWER                        *
00209                    ************************************************************
00210                    *
00211A D8B9 F7 E704  A LOADPI STAB   COMND



00213A D8BC B6 E704  A LOOP1  LDAA   COMND
00214A D8BF 48               ASLA
00215A D8C0 25 FA D8BC       BCS    LOOP1
00216A D8C2 46               RORA
00217A D8C3 84 1E    A       ANDA   #%00011110
00218A D8C5 26 5D D924       BNE    WRONG



00220A D8C7 FE CFF0  A       LDX    XKEEP
00221A D8CA EE 02    A       LDX    2,X

00223A D8CC B6 E700  A       LDAA   INOUT
00224A D8CF A7 00    A       STAA   0,X
00225A D8D1 B6 E700  A       LDAA   INOUT
00226A D8D4 A7 01    A       STAA   1,X
00227A D8D6 B6 E700  A IFTXS LDAA   INOUT
00228A D8D9 A7 02    A       STAA   2,X
00229A D8DB B6 E700  A       LDAA   INOUT
00230A D8DE A7 03    A       STAA   3,X



00232A D8E0 FE CFF0  A       LDX    XKEEP
00233A D8E3 31               INS
00234A D8E4 31               INS
00235A D8E5 6E 04    A       JMP    4,X



00237                    ************************************************************
00238                    *FP.APU SUB-PROGRAM FOR CXTFS
00239                    ************************************************************
00240                    *
00241A D8E7 30           KXTFS  TSX
00242A D8E8 EE 00    A          LDX    0,X
00243A D8EA FF CFF0  A          STX    XKEEP
00244A D8ED EE 00    A          LDX    0,X
00245A D8EF 20 BE D8AF          BRA    IXTFS
00246                    *
00247                    ************************************************************
```

```
00249                    ************************************************************
00250                    *FP.APU SUB-PROGRAM FOR CFTXS
00251                    ************************************************************
00252                    *
00253A D8F1 30           KFTXS  TSX
00254A D8F2 EE 00     A         LDX    0,X
00255A D8F4 FF CFF0   A         STX    XKEEP
00256A D8F7 EE 00     A         LDX    0,X


00258A D8F9 A6 03     A         LDAA   3,X
00259A D8FB B7 E700   A         STAA   INOUT
00260A D8FE A6 02     A         LDAA   2,X
00261A D900 B7 E700   A         STAA   INOUT
00262A D903 A6 01     A         LDAA   1,X
00263A D905 B7 E700   A         STAA   INOUT
00264A D908 A6 00     A         LDAA   0,X
00265A D90A B7 E700   A         STAA   INOUT


00267A D90D F7 E704   A         STAB   COMND


00269A D910 B6 E704   A LOOP2   LDAA   COMND
00270A D913 48                  ASLA
00271A D914 25 FA D910          BCS    LOOP2
00272A D916 46                  RORA
00273A D917 84 1E     A         ANDA   #%00011110
00274A D919 26 09 D924          BNE    WRONG


00276A D91B FE CFF0   A         LDX    XKEEP
00277A D91E EE 02     A         LDX    2,X
00278A D920 09                  DEX
00279A D921 09                  DEX
00280A D922 20 B2 D8D6          BRA    IFTXS
00281                    ************************************************************
00282                    *FP.APU  SEND ERROR CODE TO MAIN PROGRAM
00283                    ************************************************************
00284                    *
00285A D924 47           WRONG  ASRA
00286A D925 16                  TAB
00287A D926 07                  TPA
00288A D927 88 02     A         EORA   #%00000010
00289A D929 06                  TAP
00290A D92A FE CFF0   A         LDX    XKEEP
00291A D92D 31                  INS
00292A D92E 31                  INS
00293A D92F 6E 04     A         JMP    4,X
00294                    *
```

```
00296                    ************************************************************
00297                    *
00298           E700   A INOUT  EQU    $E700
00299           E704   A COMND  EQU    $E704
00300           CFF0   A XKEEP  EQU    $CFF0
00301                    *


00303                    ************************************************************
00304                    *CLEAN - CLEAR OUT APU STACK
00305                    ************************************************************
00306                    *
00307A D931 C6 17     A CLEAN  LDAB   #$17
00308A D933 4F                 CLRA
00309A D934 B7 E700   A        STAA   INOUT
00310A D937 B7 E700   A        STAA   INOUT
00311A D93A B7 E700   A        STAA   INOUT
00312A D93D B7 E700   A        STAA   INOUT
00313A D940 F7 E704   A        STAB   COMND
00314A D943 F7 E704   A        STAB   COMND
00315A D946 F7 E704   A        STAB   COMND


00317A D949 39                 RTS
00318                    *
00319                    *
00320                          END
TOTAL ERRORS 00000
```

```
D84E CFTXD  00127*
D876 CFTXS  00158*
D931 CLEAN  00307*
E704 COMND  00211 00213 00267 00269 00299*00313 00314 00315
D856 CSGNF  00133*
D852 CXTFD  00130*
D872 CXTFS  00155*
D80C DXADD  00074*
D81C DXDIV  00086*
D818 DXMULH 00083*
D814 DXMULL 00080*
D810 DXSUB  00077*
D848 EXCHF  00120*
D840 EXPX   00114*
D844 FETCH  00117*
D86A FPACOS 00149*
D820 FPADD  00089*
D866 FPASIN 00146*
D86E FPATAN 00152*
D85E FPCOS  00140*
D82C FPDIV  00098*
D828 FPMUL  00095*
D85A FPSIN  00137*
D824 FPSUB  00092*
D862 FPTAN  00143*
D830 FSQRT  00101*
D8D6 IFTXS  00227*00280
```

```
E700 INOUT  00182 00184 00186 00188 00198 00200 00202 00204 00223
            00225 00227 00229 00259 00261 00263 00265 00298*00309
            00310 00311 00312
D8AF IXTFS  00201*00245
D8F1 KFTXS  00159 00253*
D8E7 KXTFS  00156 00241*
D802 LDCOM  00067*00118 00121
D87A LOAD41 00102 00108 00111 00115 00128 00131 00134 00138 00141
            00144 00147 00150 00153 00172*
D87F LOAD42 00075 00078 00081 00084 00087 00090 00093 00096 00099
            00105 00176*
D8A0 LOAD4C 00174 00194*
D8B9 LOADPI 00124 00211*
D838 LOG10  00107*
D83C LOGE   00110*
D8BC LOOP1  00213*00215
D910 LOOP2  00269*00271
D800 PUPI   00066*
D834 PWRX   00104*
D84C STEP1  00072 00124*
D924 WRONG  00218 00274 00285*
CFF0 XKEEP  00071 00178 00190 00194 00220 00232 00243 00255 00276
            00290 00300*
```

APPENDIX B(ii)

Listing of Program AM953X

```
00001                   ************************************************************
00002                   *PROGRAM FOR USE WITH AM9511 A.P.U.
00003                   ************************************************************
00004                   *
00005                   *GENERAL CALLING SEQUENCE:-
00006                   *
00007                   *  JSR  XXXXX     XXXXX=FUNCTION NAME
00008                   *  FDB  ADD1      ADDRESS OF 1ST OPERAND
00009                   *  FDB  ADD2      ADDRESS OF 2ND OPERAND
00010                   *  FDB  ADDR      ADDRESS OF RESULT
00011                   *  -----          RETURN
00012                   *
00013                   *THERE MAY BE 0,1,OR 2 OPERANDS
00014                   *
00015                   *FUNCTIONS AVAILABLE ARE:-
00016                   *
00017                   * DXADD  ADDITION        (32-BIT FIXED POINT)
00018                   * DXSUB  SUBTRACTION     ( "    " )
00019                   * DXMULL MULTIPLICATION ( "    " ) RES=LO.HALF
00020                   * DXMULH MULTIPLICATION ( "    " ) RES=HI.HALF
00021                   * DXDIV  DIVISION        ( "    " )
00022                   * PUPI   CONSTANT PI
00023                   * FPADD  ADDITION
00024                   * FPSUB  SUBTRACTION
00025                   * FPMUL  MULTIPLICATION
00026                   * FPDIV  DIVISION
00027                   * FSQRT  SQUARE ROOT
00028                   * PWRX   POWER  Y**X
00029                   * LOG10  LOG  BASE 10
00030                   * LOGE   LOG  BASE E
00031                   * EXPX   EXPONENTIAL  E**X
00032                   *
00033                   * FPSIN  SINE
00034                   * FPCOS  COSINE
00035                   * FPTAN  TANGENT
00036                   * FPASIN ARCSINE
00037                   * FPACOS ARCCOSINE
00038                   * FPATAN ARCTANGENT
00039                   *
00040                   * CXTFS  CONVERT 16-BIT FX.PT TO 32-BIT FL.PT
00041                   * CFTXS  CONVERT 32-BIT FL.PT TO 16-BIT FX.PT
00042                   * CXTFD  CONVERT 32-BIT FX.PT TO FL.PT
00043                   * CFTXD  CONVERT 32-BIT FL.PT TO FX.PT
00044                   *
00045                   * CSGNF  CHANGE SIGN OF OPERAND LOADED
00046                   * FETCH  RETRIEVE TOP-OF-STACK
00047                   * EXCHF  RETRIEVE NEXT-ON-STACK
00048                   *
00049                   *
00050                   *THE TOP-OF-STACK NUMBER IS ALWAYS RETRIEVED
00051                   *AND STORED AT "ADDR"
00052                   *
00053                   *ALSO:-
00054                   * CLEAN  CLEARS THE A.P.U. STACK
00055                   *
00056                   ************************************************************
```

```
00058                           NAM     AM953X
0059A D800                      ORG     $D800
00060                           OPT     CREF


00062                   ********************************************************
00063                   *FP.APU  PROGRAM ENTRY POINTS
00064                   ********************************************************
00065                   *
0066A D800 C6 1A     A PUPI     LDAB    #$1A      1A IS OPCODE FOR "PI"
0067A D802 30          LDCOM    TSX
0068A D803 EE 00     A          LDX     0,X
0069A D805 09                   DEX
0070A D806 09                   DEX
0071A D807 FF CFF0   A          STX     XKEEP
0072A D80A 20 40 D84C           BRA     STEP1
00073                   *
0074A D80C C6 2C     A DXADD    LDAB    #$2C      ADDITION
0075A D80E 20 6F D87F           BRA     LOAD42
00076                   *
0077A D810 C6 2D     A DXSUB    LDAB    #$2D      SUBTRACTION
0078A D812 20 6B D87F           BRA     LOAD42
00079                   *
0080A D814 C6 2E     A DXMULL   LDAB    #$2E      MULTIPLICATION
0081A D816 20 67 D87F           BRA     LOAD42
00082                   *
0083A D818 C6 36     A DXMULH   LDAB    #$36      MULTIPLICATION
0084A D81A 20 63 D87F           BRA     LOAD42
00085                   *
0086A D81C C6 2F     A DXDIV    LDAB    #$2F      DIVISION
0087A D81E 20 5F D87F           BRA     LOAD42
00088                   *
0089A D820 C6 10     A FPADD    LDAB    #$10      ADDITION
0090A D822 20 5B D87F           BRA     LOAD42
00091                   *
0092A D824 C6 11     A FPSUB    LDAB    #$11      SUBTRACTION
0093A D826 20 57 D87F           BRA     LOAD42
00094                   *
0095A D828 C6 12     A FPMUL    LDAB    #$12      MULTIPLICATION
0096A D82A 20 53 D87F           BRA     LOAD42
00097                   *
0098A D82C C6 13     A FPDIV    LDAB    #$13      DIVISION
0099A D82E 20 4F D87F           BRA     LOAD42
00100                   *
0101A D830 C6 01     A FSQRT    LDAB    #$01      SQ.ROOT
0102A D832 20 46 D87A           BRA     LOAD41
00103                   *
0104A D834 C6 0B     A PWRX     LDAB    #$0B      POWERS
0105A D836 20 47 D87F           BRA     LOAD42
00106                   *
0107A D838 C6 08     A LOG10    LDAB    #$08      LOG 10
0108A D83A 20 3E D87A           BRA     LOAD41
00109                   *
0110A D83C C6 09     A LOGE     LDAB    #$09      LOG E
0111A D83E 20 3A D87A           BRA     LOAD41
00112                   *
```

```
00114A D840 C6 0A    A EXPX   LDAB   #$0A     EXPONENTIAL
00115A D842 20 36 D87A        BRA    LOAD41
00116                         *
00117A D844 C6 00    A FETCH  LDAB   #$00     RETRIEVE TOS
00118A D846 20 BA D802        BRA    LDCOM
00119                         *
00120A D848 C6 19    A EXCHF  LDAB   #$19     RETRIEVE NOS
00121A D84A 20 B6 D802        BRA    LDCOM
00122                         *
00123                         *
00124A D84C 20 6B D8B9 STEP1  BRA    LOADPI
00125                         *
00126                         *
00127A D84E C6 1E    A CFTXD  LDAB   #$1E     FL.PT TO 32-BIT FX.PT
00128A D850 20 28 D87A        BRA    LOAD41
00129                         *
00130A D852 C6 1C    A CXTFD  LDAB   #$1C     32-BIT FX.PT TO FL.PT
00131A D854 20 24 D87A        BRA    LOAD41
00132                         *
00133A D856 C6 15    A CSGNF  LDAB   #$15     CH.SIGN OF OPERAND LOADED
00134A D858 20 20 D87A        BRA    LOAD41
00135                         *
00136                         *
00137A D85A C6 02    A FPSIN  LDAB   #$02     SINE
00138A D85C 20 1C D87A        BRA    LOAD41
00139                         *
00140A D85E C6 03    A FPCOS  LDAB   #$03     COSINE
00141A D860 20 18 D87A        BRA    LOAD41
00142                         *
00143A D862 C6 04    A FPTAN  LDAB   #$04     TANGENT
00144A D864 20 14 D87A        BRA    LOAD41
00145                         *
00146A D866 C6 05    A FPASIN LDAB   #$05     ARCSINE
00147A D868 20 10 D87A        BRA    LOAD41
00148                         *
00149A D86A C6 06    A FPACOS LDAB   #$06     ARCCOSINE
00150A D86C 20 0C D87A        BRA    LOAD41
00151                         *
00152A D86E C6 07    A FPATAN LDAB   #$07     ARCTANGENT
00153A D870 20 08 D87A        BRA    LOAD41
00154                         *
00155A D872 C6 1D    A CXTFS  LDAB   #$1D     C.16-BIT FX.PT TO FL.PT
00156A D874 20 71 D8E7        BRA    KXTFS
00157                         *
00158A D876 C6 1F    A CFTXS  LDAB   #$1F     C.FL.PT TO 16-BIT FX.PT
00159A D878 20 77 D8F1        BRA    KFTXS
00160                         *


00162                         ***********************************************************
00163                         *
00164                         *EXTRA ENTRY POINTS MUST BE PLACED BEFORE "PUPI"
00165                         *
00166                         ***********************************************************
```

```
00168                    ************************************************************
00169                    *FP.APU  MAIN PROGRAM-LOADING OF OPERAND(S)
00170                    ************************************************************
00171                    *
00172A D87A 30           LOAD41 TSX
00173A D87B EE 00     A         LDX    0,X
00174A D87D 20 21 D8A0          BRA    LOAD4C



00176A D87F 30           LOAD42 TSX
00177A D880 EE 00     A         LDX    0,X
00178A D882 FF CFF0   A         STX    XKEEP
00179A D885 EE 00     A         LDX    0,X

00181A D887 A6 03     A         LDAA   3,X
00182A D889 B7 E700   A         STAA   INOUT
00183A D88C A6 02     A         LDAA   2,X
00184A D88E B7 E700   A         STAA   INOUT
00185A D891 A6 01     A         LDAA   1,X
00186A D893 B7 E700   A         STAA   INOUT
00187A D896 A6 00     A         LDAA   0,X
00188A D898 B7 E700   A         STAA   INOUT

00190A D89B FE CFF0   A         LDX    XKEEP
00191A D89E 08                  INX
00192A D89F 08                  INX

00194A D8A0 FF CFF0   A LOAD4C  STX    XKEEP
00195A D8A3 EE 00     A         LDX    0,X

00197A D8A5 A6 03     A         LDAA   3,X
00198A D8A7 B7 E700   A         STAA   INOUT
00199A D8AA A6 02     A         LDAA   2,X
00200A D8AC B7 E700   A         STAA   INOUT
00201A D8AF A6 01     A IXTFS   LDAA   1,X
00202A D8B1 B7 E700   A         STAA   INOUT
00203A D8B4 A6 00     A         LDAA   0,X
00204A D8B6 B7 E700   A         STAA   INOUT
```

```
00206                    ********************************************************
00207                    *FP.APU MAIN PROGRAM-LOAD COMMANDS AND                 *
00208                    *LOOK FOR ERRORS.  RETRIEVE ANSWER                     *
00209                    ********************************************************
00210                    *
00211A D8B9 F7 E704  A LOADPI STAB   COMND



00213A D8BC B6 E704  A LOOP1  LDAA   COMND
00214A D8BF 48                ASLA
00215A D8C0 25 FA D8BC        BCS    LOOP1
00216A D8C2 46                RORA
00217A D8C3 84 1E    A        ANDA   #%00011110
00218A D8C5 26 5D D924        BNE    WRONG



00220A D8C7 FE CFF0  A        LDX    XKEEP
00221A D8CA EE 02    A        LDX    2,X

00223A D8CC B6 E700  A        LDAA   INOUT
00224A D8CF A7 00    A        STAA   0,X
00225A D8D1 B6 E700  A        LDAA   INOUT
00226A D8D4 A7 01    A        STAA   1,X
00227A D8D6 B6 E700  A IFTXS  LDAA   INOUT
00228A D8D9 A7 02    A        STAA   2,X
00229A D8DB B6 E700  A        LDAA   INOUT
00230A D8DE A7 03    A        STAA   3,X



00232A D8E0 FE CFF0  A        LDX    XKEEP
00233A D8E3 31                INS
00234A D8E4 31                INS
00235A D8E5 6E 04    A        JMP    4,X



00237                    ********************************************************
00238                    *FP.APU SUB-PROGRAM FOR CXTFS
00239                    ********************************************************
00240                    *
00241A D8E7 30           KXTFS  TSX
00242A D8E8 EE 00    A        LDX    0,X
00243A D8EA FF CFF0  A        STX    XKEEP
00244A D8ED EE 00    A        LDX    0,X
00245A D8EF 20 BE D8AF        BRA    IXTFS
00246                    *
00247                    ********************************************************
```

```
00249                    ************************************************************
00250                    *FP.APU SUB-PROGRAM FOR CFTXS
00251                    ************************************************************
00252                    *
00253A D8F1 30           KFTXS  TSX
00254A D8F2 EE 00     A         LDX    0,X
00255A D8F4 FF CFF0   A         STX    XKEEP
00256A D8F7 EE 00     A         LDX    0,X



00258A D8F9 A6 03     A         LDAA   3,X
00259A D8FB B7 E700   A         STAA   INOUT
00260A D8FE A6 02     A         LDAA   2,X
00261A D900 B7 E700   A         STAA   INOUT
00262A D903 A6 01     A         LDAA   1,X
00263A D905 B7 E700   A         STAA   INOUT
00264A D908 A6 00     A         LDAA   0,X
00265A D90A B7 E700   A         STAA   INOUT



00267A D90D F7 E704   A         STAB   COMND


00269A D910 B6 E704   A LOOP2   LDAA   COMND
00270A D913 48                  ASLA
00271A D914 25 FA D910          BCS    LOOP2
00272A D916 46                  RORA
00273A D917 84 1E     A         ANDA   #%00011110
00274A D919 26 09 D924          BNE    WRONG



00276A D91B FE CFF0   A         LDX    XKEEP
00277A D91E EE 02     A         LDX    2,X
00278A D920 09                  DEX
00279A D921 09                  DEX
00280A D922 20 B2 D8D6          BRA    IFTXS
```

```
00282                          ********************************************************
00283                          *FP.APU-IDENTIFY ERRORS
00284                          ********************************************************
00285A D924 47          WRONG  ASRA
00286A D925 B7 CFF2  A         STAA   ASAVK
00287A D928 47                 ASRA
00288A D929 24 08 D933         BCC    NEXT1
00289A D92B CE D979  A         LDX    #MSS10
00290A D92E BD F024  A         JSR    PDATA
00291A D931 20 29 D95C         BRA    EEND
00292                          *
00293A D933 47          NEXT1  ASRA
00294A D934 24 08 D93E         BCC    NEXT2
00295A D936 CE D987  A         LDX    #MSS20
00296A D939 BD F024  A         JSR    PDATA
00297A D93C 20 1E D95C         BRA    EEND
00298                          *
00299A D93E 81 03    A NEXT2   CMPA   #$03
00300A D940 26 08 D94A         BNE    NEXT3
00301A D942 CE D996  A         LDX    #MSS30
00302A D945 BD F024  A         JSR    PDATA
00303A D948 20 12 D95C         BRA    EEND
00304                          *
00305A D94A 81 01    A NEXT3   CMPA   #$01
00306A D94C 26 08 D956         BNE    NEXT4
00307A D94E CE D9B2  A         LDX    #MSS40
00308A D951 BD F024  A         JSR    PDATA
00309A D954 20 06 D95C         BRA    EEND
00310                          *
00311A D956 CE D9D6  A NEXT4   LDX    #MSS50
00312A D959 BD F024  A         JSR    PDATA
00313                          *
00314A D95C CE CFF0  A EEND    LDX    #XKEEP
00315A D95F BD F01E  A         JSR    OUT4HS
00316A D962 BD F021  A         JSR    PCRLF
00317A D965 CE D9EA  A         LDX    #MSS60
00318A D968 BD F027  A         JSR    PDATA1
00319                          *
00320A D96B F6 CFF2  A         LDAB   ASAVK
00321A D96E 07                 TPA
00322A D96F 88 02    A         EORA   #%00000010
00323A D971 06                 TAP
00324A D972 FE CFF0  A         LDX    XKEEP
00325A D975 31                 INS
00326A D976 31                 INS
00327A D977 6E 04    A         JMP    4,X
00328                          *
00329A D979    4F    A MSS10   FCC    /OVERFLOW AT  /
00330A D986    04    A         FCB    4
00331A D987    55    A MSS20   FCC    /UNDERFLOW AT  /
00332A D995    04    A         FCB    4
00333A D996    41    A MSS30   FCC    /ARGUEMENT OUT OF RANGE AT  /
00334A D9B1    04    A         FCB    4
00335A D9B2    4E    A MSS40   FCC    /NEGATIVE ARGUEMENT NOT ALLOWED AT
00336A D9D5    04    A         FCB    4
00337A D9D6    44    A MSS50   FCC    /DIVIDE BY ZERO AT  /
00338A D9E9    04    A         FCB    4
00339A D9EA    07    A MSS60   FCB    7,7,7,7,4
```

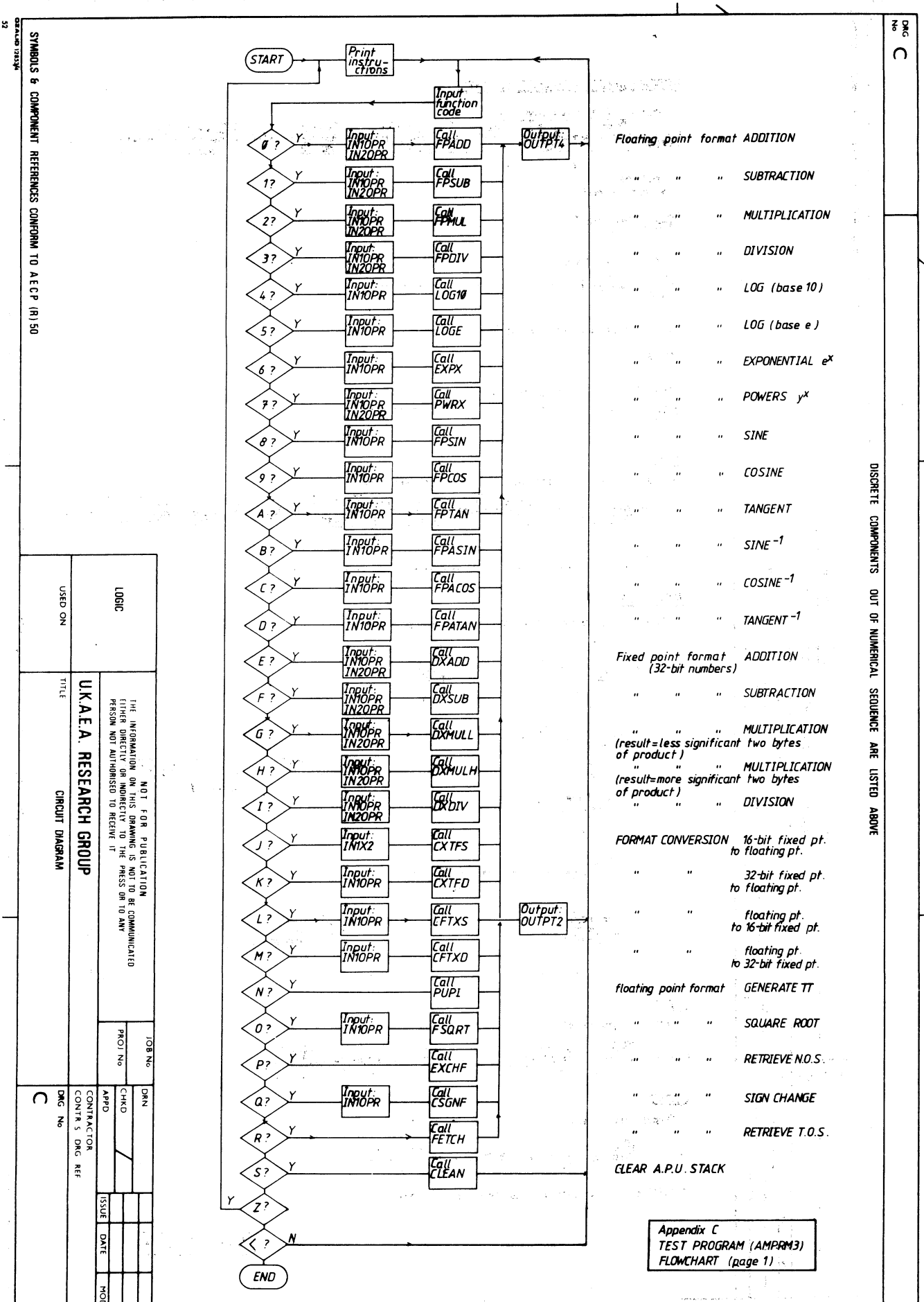
```
00340                          *
00341                          ************************************************************
00342                          *
00343           CFF2  A ASAVK  EQU     $CFF2
00344           E700  A INOUT  EQU     $E700
00345           E704  A COMND  EQU     $E704
00346           CFF0  A XKEEP  EQU     $CFF0
00347           F024  A PIATA  EQU     $F024
00348           F027  A PIATA1 EQU     $F027
00349           F01E  A OUT4HS EQU     $F01E
00350           F021  A PCRLF  EQU     $F021
00351           F0F3  A HOME   EQU     $F0F3
00352                          *


00354                          ************************************************************
00355                          *CLEAN - CLEAR OUT APU STACK
00356                          ************************************************************
00357                          *
00358A D9EF C6 17     A CLEAN  LDAB    #$17
00359A D9F1 4F                 CLRA
00360A D9F2 B7 E700   A        STAA    INOUT
00361A D9F5 B7 E700   A        STAA    INOUT
00362A D9F8 B7 E700   A        STAA    INOUT
00363A D9FB B7 E700   A        STAA    INOUT
00364A D9FE F7 E704   A        STAB    COMND
00365A DA01 F7 E704   A        STAB    COMND
00366A DA04 F7 E704   A        STAB    COMND


00368A DA07 39                 RTS
00369                    *
00370                    *
00371                          END
TOTAL ERRORS 00000


   CFF2 ASAVK  00286 00320 00343*
   D84E CFTXD  00127*
   D876 CFTXS  00158*
   D9EF CLEAN  00358*
   E704 COMND  00211 00213 00267 00269 00345*00364 00365 00366
   D856 CSGNF  00133*
   D852 CXTFD  00130*
   D872 CXTFS  00155*
   D80C DXADD  00074*
   D81C DXDIV  00086*
   D818 DXMULH 00083*
   D814 DXMULL 00080*
   D810 DXSUB  00077*
   D95C EEND   00291 00297 00303 00309 00314*
   D848 EXCHF  00120*
   D840 EXPX   00114*
   D844 FETCH  00117*
   D86A FPACOS 00149*
   D820 FPADD  00089*
   D86A FPASIN 00146*
```

```
D86E FPATAN 00152*
D85E FPCOS  00140*
D82C FPDIV  00098*
D828 FPMUL  00095*
D85A FPSIN  00137*
D824 FPSUB  00092*
D862 FPTAN  00143*
D830 FSQRT  00101*
F0F3 HOME   00351*
D8D6 IFTXS  00227*00280
E700 INOUT  00182 00184 00186 00188 00198 00200 00202 00204 00223
            00225 00227 00229 00259 00261 00263 00265 00344*00360
            00361 00362 00363
D8AF IXTFS  00201*00245
D8F1 KFTXS  00159 00253*
D8E7 KXTFS  00156 00241*
D802 LDCOM  00067*00118 00121
D87A LOAD41 00102 00108 00111 00115 00128 00131 00134 00138 00141
            00144 00147 00150 00153 00172*
D87F LOAD42 00075 00078 00081 00084 00087 00090 00093 00096 00099
            00105 00176*
D8A0 LOAD4C 00174 00194*
D8B9 LOADPI 00124 00211*
D838 LOG10  00107*
D83C LOGE   00110*
D8BC LOOP1  00213*00215
D910 LOOP2  00269*00271
D979 MSS10  00289 00329*
D987 MSS20  00295 00331*
D996 MSS30  00301 00333*
D9B2 MSS40  00307 00335*
D9D6 MSS50  00311 00337*
D9EA MSS60  00317 00339*
D933 NEXT1  00288 00293*
D93E NEXT2  00294 00299*
D94A NEXT3  00300 00305*
D956 NEXT4  00306 00311*
F01E OUT4HS 00315 00349*
F021 PCRLF  00316 00350*
F024 PDATA  00290 00296 00302 00308 00312 00347*
F027 PDATA1 00318 00348*
D800 PUPI   00066*
D834 PWRX   00104*
D84C STEP1  00072 00124*
D924 WRONG  00218 00274 00285*
CFF0 XKEEP  00071 00178 00190 00194 00220 00232 00243 00255 00276
            00314 00324 00346*
```

=

APPENDIX C

Test Software Flowchart

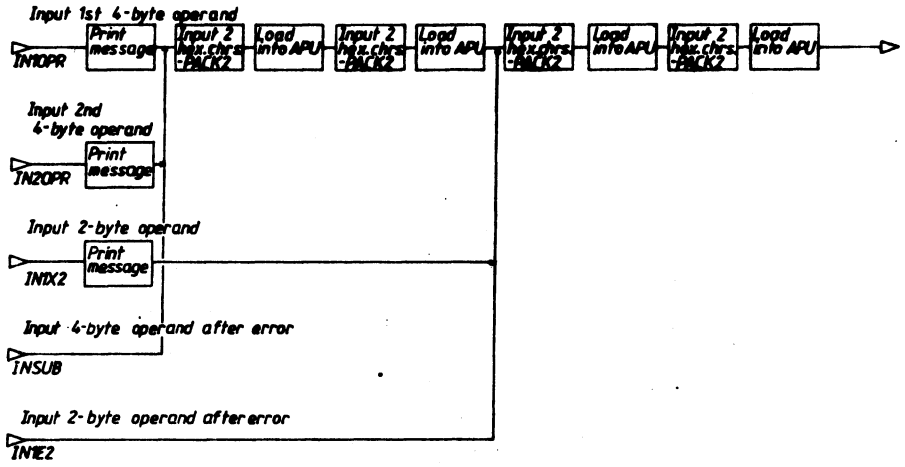

START
Print instructions
Input function code
0 ?
1?
2?
3?
4?
5?
6?
7?
8?
9?
A?
B?
C?
D?
E?
F?
G?
H?
I?
J?
K?
L?
M?
N?
O?
P?
Q?
R?
S?
Z?
L ?
Y
N
Input: IN10PR IN20PR
Input: IN10PR
Input: IN1X2
Call FPADD
Call FPSUB
Call FPMUL
Call FPDIV
Call LOG10
Call LOGE
Call EXPX
Call PWRX
Call FPSIN
Call FPCOS
Call FPTAN
Call FPASIN
Call FPACOS
Call FPATAN
Call DXADD
Call DXSUB
Call DXMULL
Call DXMULH
Call DXDIV
Call CXTFS
Call CXTFD
Call CFTXS
Call CFTXD
Call PUP1
Call FSQRT
Call EXCHF
Call CSGNF
Call FETCH
Call CLEAN
Output: OUTPT4
Output: OUTPT2
END
Floating point format ADDITION
" " " SUBTRACTION
" " " MULTIPLICATION
" " " DIVISION
" " " LOG (base 10)
" " " LOG (base e)
" " " EXPONENTIAL e^x
" " " POWERS y^x
" " " SINE
" " " COSINE
" " " TANGENT
" " " SINE $^{-1}$
" " " COSINE $^{-1}$
" " " TANGENT $^{-1}$
Fixed point format (32-bit numbers) ADDITION
" " " SUBTRACTION
" " " MULTIPLICATION (result=less significant two bytes of product)
" " " MULTIPLICATION (result=more significant two bytes of product)
" " " DIVISION
FORMAT CONVERSION 16-bit fixed pt. to floating pt.
" " 32-bit fixed pt. to floating pt.
" " floating pt. to 16-bit fixed pt.
" " floating pt. to 32-bit fixed pt.
floating point format GENERATE TT
" " " SQUARE ROOT
" " " RETRIEVE N.O.S.
" " " SIGN CHANGE
" " " RETRIEVE T.O.S.
CLEAR A.P.U. STACK
Appendix C
TEST PROGRAM (AMPRM3)
FLOWCHART (page 1)
DISCRETE COMPONENTS OUT OF NUMERICAL SEQUENCE ARE LISTED ABOVE
DRG No C
SYMBOLS & COMPONENT REFERENCES CONFORM TO A E C P (R) 50
DRAWN 12534
52
LOGIC
USED ON
U.K.A.E.A. RESEARCH GROUP
TITLE
CIRCUIT DIAGRAM
THE INFORMATION ON THIS DRAWING IS NOT TO BE COMMUNICATED EITHER DIRECTLY OR INDIRECTLY TO THE PRESS OR TO ANY PERSON NOT AUTHORISED TO RECEIVE IT
NOT FOR PUBLICATION
JOB No
PROJ No
DRN
CHKD
APPD
CONTRACTOR
CONTR'S DRG REF
DRG No C
ISSUE
DATE
MOD

SYMBOLS & COMPONENT REFERENCES CONFORM TO AECP (R) 50

DRG No C

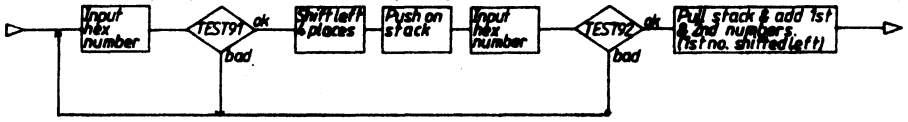
DISCRETE COMPONENTS OUT OF NUMERICAL SEQUENCE ARE LISTED ABOVE

IN1OPR / IN2OPR / IN1X2 / IN1E2 / INSUB -



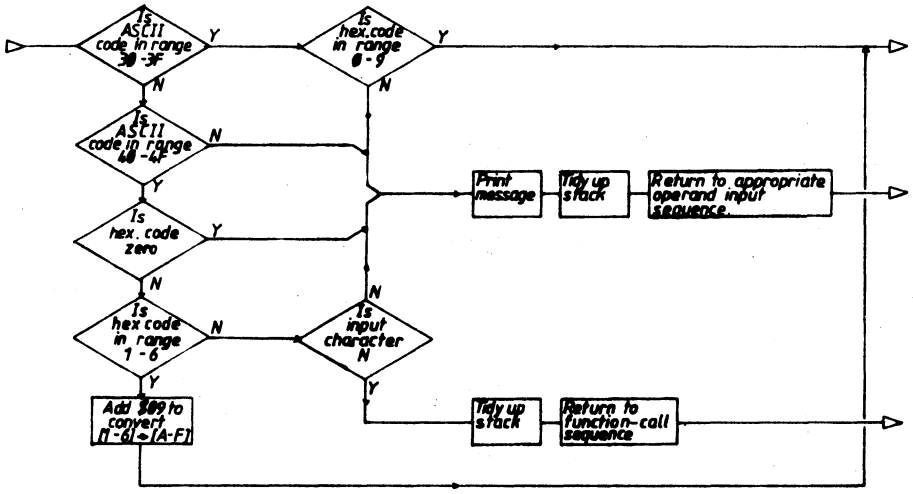
PACK2

Accept console input; pack data 2 characters per byte



TEST91 (TEST92)

Check for erroneous data (non-hexadecimal characters)

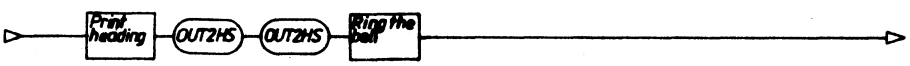


OUTPT1

Output 4-byte result

OUTPT2

Output 2-byte result



OUT2HS prints one byte as two hexadecimal characters

USED ON	LOGIC		
TITLE CIRCUIT DIAGRAM	U.K.A.E.A. RESEARCH GROUP	THE INFORMATION ON THIS DRAWING IS NOT TO BE COMMUNICATED EITHER DIRECTLY OR INDIRECTLY TO THE PRESS OR TO ANY PERSON NOT AUTHORISED TO RECEIVE IT NOT FOR PUBLICATION	JOB No
			PROJ No
DRG No C	CONTRACTOR'S DRG REF	APPD / CHKD	DRN
		ISSUE	
		DATE	
		MOD	

APPENDIX D

Listing of Test Program AMPRM3

```
PAGE  001    AMPRM3

00001                          ********************************************************
00002                          *TEST PROGRAM FOR AM9511 PROGRAM
00003                          *
00004                          *CALLS EACH FUNCTION IN TURN & ALLOWS
00005                          *OPERANDS TO BE ENTERED FROM KEYBOARD
00006                          *DISPLAYS RESULTS
00007                          *
00008                          ********************************************************
00009                          *THIS PROGRAM IS THE SAME AS AMSUM9 EXCEPT THAT
00010                          *THE CONTROL SECTION HAS A DIFFERENT ORIGIN.
00011                          *THIS AVOIDS OVERLAP WITH THE MEMORY TEST PROGRAM
00012                          *THE PROGRAM STARTS AT $B000,AND MAY BE
00013                          *CONTAINED IN TWO 2716-UVPROMS.
00014                          *FIRST PROM STARTS $B000,& CONTAINS 1995 BYTES
00015                          *SECOND PROM STARTS $B800,& CONTAINS 505 BYTES
00016                          ********************************************************
00017                          *
00018                          *
00019                                  NAM     AMPRM3
00020                                  OPT     CREF
00021A B000                            ORG     $B000
00022                          *
00023                          *
00024           F015  A        INCHNP  EQU     $F015
00025           0808  A        RES     EQU     $808
00026           F01B  A        OUT2HS  EQU     $F01B
00027                          *
00028                          *
00029A B000 8E 07FF   A                LDS     #$7FF
00030                          *
00031                          *
00032                          *PRINT OPERATING INSTRUCTIONS
00033                          *
00034A B003 CE B543   A        NEXT9   LDX     #XN16
00035A B006 BD F024   A                JSR     PDATA
00036A B009 BD F021   A        NEXT8   JSR     PCRLF
00037A B00C CE B6B9   A                LDX     #XN17
00038A B00F BD F024   A                JSR     PDATA
00039                          *
00040                          *
00041                          ********************************************************
00042                          *IDENTIFY REQUIRED FUNCTION & JUMP TO
00043                          *APROPRIATE CALLING SEQUENCE
00044                          ********************************************************
00045                          *
00046A B012 BD F015   A                JSR     INCHNP
00047A B015 81 30     A                CMPA    #'0
00048A B017 26 03 B01C                 BNE     SP1
00049A B019 7E B10E   A                JMP     TP3
00050                          *
00051A B01C 81 31     A        SP1     CMPA    #'1
00052A B01E 26 03 B023                 BNE     SP2
00053A B020 7E B129   A                JMP     TP4
00054                          *
00055A B023 81 32     A        SP2     CMPA    #'2
00056A B025 26 03 B02A                 BNE     SP3
00057A B027 7E B14A   A                JMP     TP5
00058                          *
```

```
00059A B02A 81 33     A SP3     CMPA    #'3
00060A B02C 26 03 B031          BNE     SP4
00061A B02E 7E B165   A         JMP     TP6
00062                     *
00063A B031 81 34     A SP4     CMPA    #'4
00064A B033 26 03 B038          BNE     SP5
00065A B035 7E B1BD   A         JMP     TP9
00066                     *
00067A B038 81 35     A SP5     CMPA    #'5
00068A B03A 26 03 B03F          BNE     SP6
00069A B03C 7E B1D3   A         JMP     TP10
00070                     *
00071A B03F 81 36     A SP6     CMPA    #'6
00072A B041 26 03 B046          BNE     SP7
00073A B043 7E B1E9   A         JMP     TP11
00074                     *
00075A B046 81 37     A SP7     CMPA    #'7
00076A B048 26 03 B04D          BNE     SP8
00077A B04A 7E B19C   A         JMP     TP8
00078                     *
00079A B04D 81 38     A SP8     CMPA    #'8
00080A B04F 26 03 B054          BNE     SP9
00081A B051 7E B1FF   A         JMP     TP12
00082                     *
00083A B054 81 39     A SP9     CMPA    #'9
00084A B056 26 03 B05B          BNE     SPA
00085A B058 7E B21B   A         JMP     TP13
00086                     *
00087A B05B 81 41     A SPA     CMPA    #'A
00088A B05D 26 03 B062          BNE     SPB
00089A B05F 7E B237   A         JMP     TP14
00090                     *
00091A B062 81 42     A SPB     CMPA    #'B
00092A B064 26 03 B069          BNE     SPC
00093A B066 7E B253   A         JMP     TP15
00094                     *
00095A B069 81 43     A SPC     CMPA    #'C
00096A B06B 26 03 B070          BNE     SPD
00097A B06D 7E B26F   A         JMP     TP16
00098                     *
00099A B070 81 44     A SPD     CMPA    #'D
00100A B072 26 03 B077          BNE     SPE
00101A B074 7E B28B   A         JMP     TP18
00102                     *
00103A B077 81 45     A SPE     CMPA    #'E
00104A B079 26 03 B07E          BNE     SPF
00105A B07B 7E B337   A         JMP     TP26
00106                     *
00107A B07E 81 46     A SPF     CMPA    #'F
00108A B080 26 03 B085          BNE     SPG
00109A B082 7E B352   A         JMP     TP27
00110                     *
00111A B085 81 47     A SPG     CMPA    #'G
00112A B087 26 03 B08C          BNE     SPH
00113A B089 7E B373   A         JMP     TP28
00114                     *
00115A B08C 81 48     A SPH     CMPA    #'H
00116A B08E 26 03 B093          BNE     SPI
```

```
00117A B090 7E B38E  A          JMP     TP29
00118                       *
00119A B093 81 49    A SPI      CMPA    #'I
00120A B095 26 03 B09A          BNE     SPJ
00121A B097 7E B3A9  A          JMP     TP30
00122                       *
00123A B09A 81 4A    A SPJ      CMPA    #'J
00124A B09C 26 03 B0A1          BNE     SPK
00125A B09E 7E B2A7  A          JMP     TP19
00126                       *
00127A B0A1 81 4B    A SPK      CMPA    #'K
00128A B0A3 26 03 B0A8          BNE     SPL
00129A B0A5 7E B2D3  A          JMP     TP21
00130                       *
00131A B0A8 81 4C    A SPL      CMPA    #'L
00132A B0AA 26 03 B0AF          BNE     SPM
00133A B0AC 7E B2BD  A          JMP     TP20
00134                       *
00135A B0AF 81 4D    A SPM      CMPA    #'M
00136A B0B1 26 03 B0B6          BNE     SPN
00137A B0B3 7E B2E9  A          JMP     TP22
00138                       *
00139A B0B6 81 4E    A SPN      CMPA    #'N
00140A B0B8 26 03 B0BD          BNE     SPO
00141A B0BA 7E B0FD  A          JMP     TP2
00142                       *
00143A B0BD 81 4F    A SPO      CMPA    #'O
00144A B0BF 26 03 B0C4          BNE     SPP
00145A B0C1 7E B186  A          JMP     TP7
00146                       *
00147A B0C4 81 50    A SPP      CMPA    #'P
00148A B0C6 26 03 B0CB          BNE     SPQ
00149A B0C8 7E B326  A          JMP     TP25
00150                       *
00151A B0CB 81 51    A SPQ      CMPA    #'Q
00152A B0CD 26 03 B0D2          BNE     SPR
00153A B0CF 7E B2FF  A          JMP     TP23
00154                       *
00155A B0D2 81 52    A SPR      CMPA    #'R
00156A B0D4 26 03 B0D9          BNE     SPS
00157A B0D6 7E B315  A          JMP     TP24
00158                       *
00159A B0D9 81 53    A SPS      CMPA    #'S
00160A B0DB 27 11 B0EE          BEQ     TP1
00161                       *
00162A B0DD 81 5A    A          CMPA    #'Z
00163A B0DF 26 03 B0E4          BNE     SPZ
00164A B0E1 7E B003  A          JMP     NEXT9
00165                       *
00166A B0E4 81 3C    A SPZ      CMPA    #'<
00167A B0E6 26 03 B0EB          BNE     AGAIN
00168A B0E8 7E F0F3  A          JMP     HOME
00169                       *
00170A B0EB 7E B009  A AGAIN    JMP     NEXT8
00171                       *
00172                       *
```

```
00174                          ************************************************************
00175                          *CALLING SEQUENCES FOR EACH FUNCTION
00176                          ************************************************************
00177                          *
00178A B0EE CE B453  A TP1      LDX     #FN22
00179A B0F1 BD F024  A         JSR     PDATA
00180A B0F4 BD B9E1  A         JSR     CLEAN
00181A B0F7 BD B6CB  A         JSR     MOVE2
00182A B0FA 7E B009  A         JMP     NEXT8
00183                          *
00184                          *
00185A B0FD CE B3CA  A TP2      LDX     #FN0
00186A B100 BD F024  A         JSR     PDATA
00187A B103 BD B800  A         JSR     PUPI
00188A B106     0808 A         FDB     $0808
00189A B108 BD B6D2  A         JSR     OUTPT4
00190A B10B 7E B009  A         JMP     NEXT8
00191                          *
00192                          *
00193A B10E CE B3CF  A TP3      LDX     #FN1
00194A B111 BD F024  A         JSR     PDATA
00195A B114 BD B77A  A         JSR     IN1OPR
00196A B117 BD B7AA  A         JSR     IN2OPR
00197A B11A BD B820  A         JSR     FPADD
00198A B11D     0800 A         FDB     $0800
00199A B11F     0804 A         FDB     $0804
00200A B121     0808 A         FDB     $0808
00201A B123 BD B6D2  A         JSR     OUTPT4
00202A B126 7E B009  A         JMP     NEXT8
00203                          *
00204                          *
00205A B129 CE B3D5  A TP4      LDX     #FN2
00206A B12C BD F024  A         JSR     PDATA
00207A B12F CE B4B7  A         LDX     #XN7
00208A B132 BD F024  A         JSR     PDATA
00209A B135 BD B77A  A         JSR     IN1OPR
00210A B138 BD B7AA  A         JSR     IN2OPR
00211A B13B BD B824  A         JSR     FPSUB
00212A B13E     0800 A         FDB     $0800
00213A B140     0804 A         FDB     $0804
00214A B142     0808 A         FDB     $0808
00215A B144 BD B6D2  A         JSR     OUTPT4
00216A B147 7E B009  A         JMP     NEXT8
00217                          *
00218                          *
00219A B14A CE B3DB  A TP5      LDX     #FN3
00220A B14D BD F024  A         JSR     PDATA
00221A B150 BD B77A  A         JSR     IN1OPR
00222A B153 BD B7AA  A         JSR     IN2OPR
00223A B156 BD B828  A         JSR     FPMUL
00224A B159     0800 A         FDB     $0800
00225A B15B     0804 A         FDB     $0804
00226A B15D     0808 A         FDB     $0808
00227A B15F BD B6D2  A         JSR     OUTPT4
00228A B162 7E B009  A         JMP     NEXT8
00229                          *
00230                          *
00231A B165 CE B3E1  A TP6      LDX     #FN3A
```

```
00232A B168 BD F024  A          JSR    PDATA
00233A B16B CE B4D6  A          LDX    #XN8
00234A B16E BD F024  A          JSR    PDATA
00235A B171 BD B77A  A          JSR    IN1OPR
00236A B174 BD B7AA  A          JSR    IN2OPR
00237A B177 BD B82C  A          JSR    FPDIV
00238A B17A    0800  A          FDB    $0800
00239A B17C    0804  A          FDB    $0804
00240A B17E    0808  A          FDB    $0808
00241A B180 BD B6D2  A          JSR    OUTPT4
00242A B183 7E B009  A          JMP    NEXT8
00243                       *
00244                       *
00245A B186 CE B3E7  A TP7      LDX    #FN4
00246A B189 BD F024  A          JSR    PDATA
00247A B18C BD B77A  A          JSR    IN1OPR
00248A B18F BD B830  A          JSR    FSQRT
00249A B192    0800  A          FDB    $0800
00250A B194    0808  A          FDB    $0808
00251A B196 BD B6D2  A          JSR    OUTPT4
00252A B199 7E B009  A          JMP    NEXT8
00253                       *
00254                       *
00255A B19C CE B3ED  A TP8      LDX    #FN5
00256A B19F BD F024  A          JSR    PDATA
00257A B1A2 CE B4F5  A          LDX    #XN9
00258A B1A5 BD F024  A          JSR    PDATA
00259A B1A8 BD B77A  A          JSR    IN1OPR
00260A B1AB BD B7AA  A          JSR    IN2OPR
00261A B1AE BD B834  A          JSR    PWRX
00262A B1B1    0800  A          FDB    $0800
00263A B1B3    0804  A          FDB    $0804
00264A B1B5    0808  A          FDB    $0808
00265A B1B7 BD B6D2  A          JSR    OUTPT4
00266A B1BA 7E B009  A          JMP    NEXT8
00267                       *
00268                       *
00269A B1BD CE B3F2  A TP9      LDX    #FN6
00270A B1C0 BD F024  A          JSR    PDATA
00271A B1C3 BD B77A  A          JSR    IN1OPR
00272A B1C6 BD B838  A          JSR    LOG10
00273A B1C9    0800  A          FDB    $0800
00274A B1CB    0808  A          FDB    $0808
00275A B1CD BD B6D2  A          JSR    OUTPT4
00276A B1D0 7E B009  A          JMP    NEXT8
00277                       *
00278                       *
00279A B1D3 CE B3F8  A TP10     LDX    #FN7
00280A B1D6 BD F024  A          JSR    PDATA
00281A B1D9 BD B77A  A          JSR    IN1OPR
00282A B1DC BD B83C  A          JSR    LOGE
00283A B1DF    0800  A          FDB    $0800
00284A B1E1    0808  A          FDB    $0808
00285A B1E3 BD B6D2  A          JSR    OUTPT4
00286A B1E6 7E B009  A          JMP    NEXT8
00287                       *
00288                       *
00289A B1E9 CE B3FD  A TP11     LDX    #FN8
```

```
00290A B1EC BD F024  A          JSR     PDATA
00291A B1EF BD B77A  A          JSR     IN1OPR
00292A B1F2 BD B840  A          JSR     EXPX
00293A B1F5     0800 A          FDB     $0800
00294A B1F7     0808 A          FDB     $0808
00295A B1F9 BD B6D2  A          JSR     OUTPT4
00296A B1FC 7E B009  A          JMP     NEXT8
00297                           *
00298                           *
00299A B1FF CE B402  A TP12     LDX     #FN9
00300A B202 BD F024  A          JSR     PDATA
00301A B205 CE B515  A          LDX     #XN10
00302A B208 BD F024  A          JSR     PDATA
00303A B20B BD B77A  A          JSR     IN1OPR
00304A B20E BD B85A  A          JSR     FPSIN
00305A B211     0800 A          FDB     $0800
00306A B213     0808 A          FDB     $0808
00307A B215 BD B6D2  A          JSR     OUTPT4
00308A B218 7E B009  A          JMP     NEXT8
00309                           *
00310                           *
00311A B21B CE B408  A TP13     LDX     #FN10
00312A B21E BD F024  A          JSR     PDATA
00313A B221 CE B515  A          LDX     #XN10
00314A B224 BD F024  A          JSR     PDATA
00315A B227 BD B77A  A          JSR     IN1OPR
00316A B22A BD B85E  A          JSR     FPCOS
00317A B22D     0800 A          FDB     $0800
00318A B22F     0808 A          FDB     $0808
00319A B231 BD B6D2  A          JSR     OUTPT4
00320A B234 7E B009  A          JMP     NEXT8
00321                           *
00322                           *
00323A B237 CE B40E  A TP14     LDX     #FN11
00324A B23A BD F024  A          JSR     PDATA
00325A B23D CE B515  A          LDX     #XN10
00326A B240 BD F024  A          JSR     PDATA
00327A B243 BD B77A  A          JSR     IN1OPR
00328A B246 BD B862  A          JSR     FPTAN
00329A B249     0800 A          FDB     $0800
00330A B24B     0808 A          FDB     $0808
00331A B24D BD B6D2  A          JSR     OUTPT4
00332A B250 7E B009  A          JMP     NEXT8
00333                           *
00334                           *
00335A B253 CE B414  A TP15     LDX     #FN12
00336A B256 BD F024  A          JSR     PDATA
00337A B259 CE B515  A          LDX     #XN10
00338A B25C BD F024  A          JSR     PDATA
00339A B25F BD B77A  A          JSR     IN1OPR
00340A B262 BD B866  A          JSR     FPASIN
00341A B265     0800 A          FDB     $0800
00342A B267     0808 A          FDB     $0808
00343A B269 BD B6D2  A          JSR     OUTPT4
00344A B26C 7E B009  A          JMP     NEXT8
00345                           *
00346                           *
00347A B26F CE B41B  A TP16     LDX     #FN13
```

```
00348A B272 BD F024  A          JSR     PDATA
00349A B275 CE B515  A          LDX     #XN10
00350A B278 BD F024  A          JSR     PDATA
00351A B27B BD B77A  A          JSR     IN1OPR
00352A B27E BD B86A  A          JSR     FPACOS
00353A B281    0800  A          FDB     $0800
00354A B283    0808  A          FDB     $0808
00355A B285 BD B6D2  A          JSR     OUTPT4
00356A B288 7E B009  A          JMP     NEXT8
00357                           *
00358                           *
00359A B28B CE B422  A TP18     LDX     #FN14
00360A B28E BD F024  A          JSR     PDATA
00361A B291 CE B515  A          LDX     #XN10
00362A B294 BD F024  A          JSR     PDATA
00363A B297 BD B77A  A          JSR     IN1OPR
00364A B29A BD B86E  A          JSR     FPATAN
00365A B29D    0800  A          FDB     $0800
00366A B29F    0808  A          FDB     $0808
00367A B2A1 BD B6D2  A          JSR     OUTPT4
00368A B2A4 7E B009  A          JMP     NEXT8
00369                           *
00370                           *
00371A B2A7 CE B429  A TP19     LDX     #FN15
00372A B2AA BD F024  A          JSR     PDATA
00373A B2AD BD B7E5  A          JSR     IN1X2
00374A B2B0 BD B872  A RTN      JSR     CXTFS
00375A B2B3    0800  A          FDB     $0800
00376A B2B5    0808  A          FDB     $0808
00377A B2B7 BD B6D2  A          JSR     OUTPT4
00378A B2BA 7E B009  A          JMP     NEXT8
00379                           *
00380                           *
00381A B2BD CE B42F  A TP20     LDX     #FN16
00382A B2C0 BD F024  A          JSR     PDATA
00383A B2C3 BD B77A  A          JSR     IN1OPR
00384A B2C6 BD B876  A          JSR     CFTXS
00385A B2C9    0800  A          FDB     $0800
00386A B2CB    0808  A          FDB     $0808
00387A B2CD BD B6BC  A          JSR     OUTPT2
00388A B2D0 7E B009  A          JMP     NEXT8
00389                           *
00390                           *
00391A B2D3 CE B435  A TP21     LDX     #FN17
00392A B2D6 BD F024  A          JSR     PDATA
00393A B2D9 BD B77A  A          JSR     IN1OPR
00394A B2DC BD B852  A          JSR     CXTFD
00395A B2DF    0800  A          FDB     $0800
00396A B2E1    0808  A          FDB     $0808
00397A B2E3 BD B6D2  A          JSR     OUTPT4
00398A B2E6 7E B009  A          JMP     NEXT8
00399                           *
00400                           *
00401A B2E9 CE B43B  A TP22     LDX     #FN18
00402A B2EC BD F024  A          JSR     PDATA
00403A B2EF BD B77A  A          JSR     IN1OPR
00404A B2F2 BD B84E  A          JSR     CFTXD
00405A B2F5    0800  A          FDB     $0800
```

```
00406A B2F7    0808  A        FDB    $0808
00407A B2F9 BD B6D2  A        JSR    OUTPT4
00408A B2FC 7E B009  A        JMP    NEXT8
00409                 *
00410                 *
00411A B2FF CE B441  A TP23   LDX    #FN19
00412A B302 BD F024  A        JSR    PDATA
00413A B305 BD B77A  A        JSR    IN1OPR
00414A B308 BD B856  A        JSR    CSGNF
00415A B30B    0800  A        FDB    $0800
00416A B30D    0808  A        FDB    $0808
00417A B30F BD B6D2  A        JSR    OUTPT4
00418A B312 7E B009  A        JMP    NEXT8
00419                 *
00420                 *
00421A B315 CE B447  A TP24   LDX    #FN20
00422A B318 BD F024  A        JSR    PDATA
00423A B31B BD B844  A        JSR    FETCH
00424A B31E    0808  A        FDB    $0808
00425A B320 BD B6D2  A        JSR    OUTPT4
00426A B323 7E B009  A        JMP    NEXT8
00427                 *
00428                 *
00429A B326 CE B44D  A TP25   LDX    #FN21
00430A B329 BD F024  A        JSR    PDATA
00431A B32C BD B848  A        JSR    EXCHF
00432A B32F    0808  A        FDB    $0808
00433A B331 BD B6D2  A        JSR    OUTPT4
00434A B334 7E B009  A        JMP    NEXT8
00435                 *
00436                 *
00437A B337 CE B459  A TP26   LDX    #FN23
00438A B33A BD F024  A        JSR    PDATA
00439A B33D BD B77A  A        JSR    IN1OPR
00440A B340 BD B7AA  A        JSR    IN2OPR
00441A B343 BD B80C  A        JSR    DXADD
00442A B346    0800  A        FDB    $0800
00443A B348    0804  A        FDB    $0804
00444A B34A    0808  A        FDB    $0808
00445A B34C BD B6D2  A        JSR    OUTPT4
00446A B34F 7E B009  A        JMP    NEXT8
00447                 *
00448                 *
00449A B352 CE B45F  A TP27   LDX    #FN24
00450A B355 BD F024  A        JSR    PDATA
00451A B358 CE B4B7  A        LDX    #XN7
00452A B35B BD F024  A        JSR    PDATA
00453A B35E BD B77A  A        JSR    IN1OPR
00454A B361 BD B7AA  A        JSR    IN2OPR
00455A B364 BD B810  A        JSR    DXSUB
00456A B367    0800  A        FDB    $0800
00457A B369    0804  A        FDB    $0804
00458A B36B    0808  A        FDB    $0808
00459A B36D BD B6D2  A        JSR    OUTPT4
00460A B370 7E B009  A        JMP    NEXT8
00461                 *
00462                 *
00463A B373 CE B465  A TP28   LDX    #FN25
```

```
00464A B376 BD F024  A          JSR     PDATA
00465A B379 BD B77A  A          JSR     IN1OPR
00466A B37C BD B7AA  A          JSR     IN2OPR
00467A B37F BD B814  A          JSR     DXMULL
00468A B382     0800 A          FDB     $0800
00469A B384     0804 A          FDB     $0804
00470A B386     0808 A          FDB     $0808
00471A B388 BD B6D2  A          JSR     OUTPT4
00472A B38B 7E B009  A          JMP     NEXT8
00473                           *
00474                           *
00475A B38E CE B46C  A TP29     LDX     #FN26
00476A B391 BD F024  A          JSR     PDATA
00477A B394 BD B77A  A          JSR     IN1OPR
00478A B397 BD B7AA  A          JSR     IN2OPR
00479A B39A BD B818  A          JSR     DXMULH
00480A B39D     0800 A          FDB     $0800
00481A B39F     0804 A          FDB     $804
00482A B3A1     0808 A          FDB     $0808
00483A B3A3 BD B6D2  A          JSR     OUTPT4
00484A B3A6 7E B009  A          JMP     NEXT8
00485                           *
00486                           *
00487A B3A9 CE B473  A TP30     LDX     #FN27
00488A B3AC BD F024  A          JSR     PDATA
00489A B3AF CE B4D6  A          LDX     #XN8
00490A B3B2 BD F024  A          JSR     PDATA
00491A B3B5 BD B77A  A          JSR     IN1OPR
00492A B3B8 BD B7AA  A          JSR     IN2OPR
00493A B3BB BD B81C  A          JSR     DXDIV
00494A B3BE     0800 A          FDB     $0800
00495A B3C0     0804 A          FDB     $0804
00496A B3C2     0808 A          FDB     $0808
00497A B3C4 BD B6D2  A          JSR     OUTPT4
00498A B3C7 7E B009  A          JMP     NEXT8
00499                           *
00500                           *
00501                           ********************************************************
00502                           *HEADINGS FOR EACH FUNCTION,& OTHER MESSAGES
00503                           ********************************************************
00504                           *
00505A B3CA    50    A FN0      FCC     /PUPI/
00506A B3CE    04    A          FCB     4
00507A B3CF    46    A FN1      FCC     /FPADD/
00508A B3D4    04    A          FCB     4
00509A B3D5    46    A FN2      FCC     /FPSUB/
00510A B3DA    04    A          FCB     4
00511A B3DB    46    A FN3      FCC     /FPMUL/
00512A B3E0    04    A          FCB     4
00513A B3E1    46    A FN3A     FCC     /FPDIV/
00514A B3E6    04    A          FCB     4
00515A B3E7    46    A FN4      FCC     /FSQRT/
00516A B3EC    04    A          FCB     4
00517A B3ED    50    A FN5      FCC     /PWRX/
00518A B3F1    04    A          FCB     4
00519A B3F2    4C    A FN6      FCC     /LOG10/
00520A B3F7    04    A          FCB     4
00521A B3F8    4C    A FN7      FCC     /LOGE/
```

```
00522A B3FC    04    A        FCB   4
00523A B3FD    45    A FN8    FCC   /EXPX/
00524A B401    04    A        FCB   4
00525A B402    46    A FN9    FCC   /FPSIN/
00526A B407    04    A        FCB   4
00527A B408    46    A FN10   FCC   /FPCOS/
00528A B40D    04    A        FCB   4
00529A B40E    46    A FN11   FCC   /FPTAN/
00530A B413    04    A        FCB   4
00531A B414    46    A FN12   FCC   /FPASIN/
00532A B41A    04    A        FCB   4
00533A B41B    46    A FN13   FCC   /FPACOS/
00534A B421    04    A        FCB   4
00535A B422    46    A FN14   FCC   /FPATAN/
00536A B428    04    A        FCB   4
00537A B429    43    A FN15   FCC   /CXTFS/
00538A B42E    04    A        FCB   4
00539A B42F    43    A FN16   FCC   /CFTXS/
00540A B434    04    A        FCB   4
00541A B435    43    A FN17   FCC   /CXTFD/
00542A B43A    04    A        FCB   4
00543A B43B    43    A FN18   FCC   /CFTXD/
00544A B440    04    A        FCB   4
00545A B441    43    A FN19   FCC   /CSGNF/
00546A B446    04    A        FCB   4
00547A B447    46    A FN20   FCC   /FETCH/
00548A B44C    04    A        FCB   4
00549A B44D    45    A FN21   FCC   /EXCHF/
00550A B452    04    A        FCB   4
00551A B453    43    A FN22   FCC   /CLEAN/
00552A B458    04    A        FCB   4
00553A B459    44    A FN23   FCC   /DXADD/
00554A B45E    04    A        FCB   4
00555A B45F    44    A FN24   FCC   /DXSUB/
00556A B464    04    A        FCB   4
00557A B465    44    A FN25   FCC   /DXMULL/
00558A B46B    04    A        FCB   4
00559A B46C    44    A FN26   FCC   /DXMULH/
00560A B472    04    A        FCB   4
00561A B473    44    A FN27   FCC   /DXDIV/
00562A B478    04    A        FCB   4
00563A B479    31    A XN1    FCC   /1ST OPERAND=/
00564A B485    04    A        FCB   4
00565A B486    32    A XN2    FCC   /2ND OPERAND=/
00566A B492    04    A        FCB   4
00567A B493    52    A XN3    FCC   /RESULT     =/
00568A B49F    04    A        FCB   4
00569A B4A0    08    A XN4    FCB   8,8,8,8,8,8,8,8,8,8,$2E,$A,$D,7,7,4
00570A B4B0    0A    A XN5    FCB   $A,$D,7,7,4
00571A B4B5    20    A XN6    FCB   $20,4
00572A B4B7    52    A XN7    FCC   /RESULT=1ST OPERAND-2ND OPERAND/
00573A B4D5    04    A        FCB   4
00574A B4D6    52    A XN8    FCC   XRESULT=1ST OPERAND/2ND OPERANDX
00575A B4F4    04    A        FCB   4
00576A B4F5    52    A XN9    FCC   /RESULT=1ST OPERAND**2ND OPERAND/
00577A B514    04    A        FCB   4
00578A B515    41    A XN10   FCC   /ANGLES IN RADIANS/
00579A B526    04    A        FCB   4
```

```
00580A B527    2A    A XN11  FCC   /****MISTAKE!/
00581A B533    04    A       FCB   4
00582A B534    20    A XN12  FCC   /    OPERAND=/
00583A B540    04    A       FCB   4
00584A B541    2E    A XN13  FCB   $2E,4
00585A B543    55    A XN16  FCC   /USE THE FOLLOWING CODES TO CALL FU
00586A B56C    0A    A       FCB   $0A,$0D
00587A B56E    20    A       FCC   / FPADD .0  LOG10 .4  FPSIN .8  FPA
00588A B5A9    0A    A       FCB   $0A,$0D
00589A B5AB    20    A       FCC   / FPSUB .1  LOGE  .5  FPCOS .9  FPA
00590A B5E6    0A    A       FCB   $0A,$0D
00591A B5E8    20    A       FCC   / FPMUL .2  EXPX  .6  FPTAN .A  FPA
00592A B623    0A    A       FCB   $0A,$0D
00593A B625    20    A       FCC   / FPDIV .3  PWRX  .7  TABLE .Z  FET
00594A B660    0A    A       FCB   $0A,$0D
00595A B662    20    A       FCC   / PUPI  .N  FSQRT .0  EXCHF .P  CSG
00596A B69D    0A    A       FCB   $0A,$0A,$0D
00597A B6A0    20    A       FCC   / ESCAPE FROM PROGRAM  .</
00598A B6B8    04    A       FCB   4
00599A B6B9    3E    A XN17  FCB   $3E,$3E,4
00600                  *
00601                  *
00602                  *
00603                  ************************************************************
00604                  *OUTPT2/4  PRINTS RESULTS 16 OR 32 BIT FORMAT
00605                  ************************************************************
00606                  *
00607A B6BC CE B493   A OUTPT2 LDX   #XN3
00608A B6BF BD F024   A        JSR   PDATA
00609A B6C2 CE 0808   A        LDX   #RES
00610A B6C5 BD F01B   A        JSR   OUT2HS
00611A B6C8 BD F01B   A        JSR   OUT2HS
00612A B6CB CE B4B0   A MOVE2  LDX   #XN5
00613A B6CE BD F024   A        JSR   PDATA
00614A B6D1 39                 RTS
00615                  *
00616A B6D2 CE B493   A OUTPT4 LDX   #XN3
00617A B6D5 BD F024   A        JSR   PDATA
00618A B6D8 CE 0808   A        LDX   #RES
00619A B6DB BD F01B   A        JSR   OUT2HS
00620A B6DE BD F01B   A        JSR   OUT2HS
00621A B6E1 BD F01B   A        JSR   OUT2HS
00622A B6E4 BD F01B   A        JSR   OUT2HS
00623A B6E7 CE B4A0   A        LDX   #XN4
00624A B6EA BD F027   A        JSR   PDATA1
00625A B6ED 39                 RTS
00626                  *
00627                  *
```

```
00629                          ************************************************************
00630                          *PACK2  PACKS CHARACTERS FROM KEYBOARD,2 PER BYTE
00631                          *& CONVERTS ASCII TO HEX
00632                          ************************************************************
00633                          *
00634A B6EE BD F015  A PACK2   JSR     INCHNP
00635A B6F1 16                TAB
00636A B6F2 BD B704  A         JSR     TEST91
00637A B6F5 48                ASLA
00638A B6F6 48                ASLA
00639A B6F7 48                ASLA
00640A B6F8 48                ASLA
00641A B6F9 36                PSHA
00642                          *
00643A B6FA BD F015  A         JSR     INCHNP
00644A B6FD 16                TAB
00645A B6FE BD B72F  A         JSR     TEST92
00646                          *
00647A B701 33                PULB
00648A B702 1B                ABA
00649                          *
00650A B703 39                RTS
00651                          *
00652                          *
00653                          ************************************************************
00654                          *TEST91/92  CHECKS KEYBOARD INPUT FOR NON-HEX
00655                          *CHARACTERS & ALLOWS CORRECTIONS
00656                          ************************************************************
00657                          *
00658A B704 84 F0     A TEST91 ANDA    #%11110000
00659A B706 81 30     A        CMPA    #$30
00660A B708 27 1D B727         BEQ     YES3
00661                          *
00662A B70A 81 40     A        CMPA    #$40
00663A B70C 26 47 B755         BNE     ERR4R
00664                          *
00665A B70E 17                 TBA
00666A B70F 84 0F     A        ANDA    #%00001111
00667A B711 27 42 B755         BEQ     ERR4R
00668A B713 81 06     A        CMPA    #$06
00669A B715 2E 03 B71A         BGT     ERR0R
00670A B717 8B 09     A        ADDA    #$09
00671A B719 39                 RTS
00672                          *
00673A B71A 81 0E     A ERR0R  CMPA    #%00001110
00674A B71C 26 37 B755         BNE     ERR4R
00675A B71E 31          RTN10  INS
00676A B71F 31                 INS
00677A B720 31                 INS
00678A B721 31                 INS
00679A B722 31                 INS
00680A B723 31                 INS
00681A B724 7E B009  A         JMP     NEXT8
00682                          *
00683A B727 17          YES3   TBA
00684A B728 84 0F     A        ANDA    #%00001111
00685A B72A 81 09     A        CMPA    #$09
00686A B72C 2E 27 B755         BGT     ERR4R
```

```
00687A B72E 39                  RTS
00688                       *
00689A B72F 84 F0    A TEST92   ANDA    #%11110000
00690A B731 81 30    A          CMPA    #$30
00691A B733 27 17 B74C          BEQ     YES33
00692                       *
00693A B735 81 40    A          CMPA    #$40
00694A B737 26 1B B754          BNE     ERR5R
00695                       *
00696A B739 17                  TBA
00697A B73A 84 0F    A          ANDA    #%00001111
00698A B73C 27 16 B754          BEQ     ERR5R
00699A B73E 81 06    A          CMPA    #$06
00700A B740 2E 03 B745          BGT     ERR1R
00701A B742 8B 09    A          ADDA    #$09
00702A B744 39                  RTS
00703                       *
00704A B745 81 0E    A ERR1R    CMPA    #%00001110
00705A B747 26 0B B754          BNE     ERR5R
00706A B749 33                  PULB
00707A B74A 20 D2 B71E          BRA     RTN10
00708                       *
00709A B74C 17            YES33 TBA
00710A B74D 84 0F    A          ANDA    #%00001111
00711A B74F 81 09    A          CMPA    #$09
00712A B751 2E 01 B754          BGT     ERR5R
00713A B753 39                  RTS
00714                       *
00715A B754 33            ERR5R PULB
00716A B755 FF 0850  A ERR4R    STX     XKEEP
00717A B758 CE B527  A          LDX     #XN11
00718A B75B BD F027  A          JSR     PDATA1
00719A B75E CE B534  A          LDX     #XN12
00720A B761 BD F024  A          JSR     PDATA
00721A B764 31                  INS
00722A B765 31                  INS
00723A B766 31                  INS
00724A B767 31                  INS
00725A B768 30                  TSX
00726A B769 EE 00    A          LDX     0,X
00727A B76B 8C B2B0  A          CPX     #RTN
00728A B76E 27 05 B775          BEQ     ERR6R
00729A B770 FE 0850  A          LDX     XKEEP
00730A B773 20 0E B783          BRA     INSUB
00731                       *
00732A B775 FE 0850  A ERR6R    LDX     XKEEP
00733A B778 20 41 B7BB          BRA     IN1E2
00734                       *
```

```
00736                          ************************************************************
00737                          *IN1OPR  INPUT 1ST OPERAND
00738                          *IN2OPR  INPUT 2ND OPERAND
00739                          *SPACE  FORMATTING FOR KEYBOARD IN/OUTPUT
00740                          ************************************************************
00741                          *
00742A B77A CE B479  A IN1OPR LDX    #XN1
00743A B77D BD F024  A        JSR    PDATA
00744A B780 CE 0800  A        LDX    #$800
00745A B783 BD B6EE  A INSUB  JSR    PACK2
00746A B786 A7 00    A        STAA   0,X
00747A B788 FF 0850  A        STX    XKEEP
00748A B78B CE B541  A        LDX    #XN13
00749A B78E BD F027  A        JSR    PDATA1
00750A B791 FE 0850  A        LDX    XKEEP
00751A B794 BD B6EE  A        JSR    PACK2
00752A B797 A7 01    A        STAA   1,X
00753A B799 BD B7C0  A        JSR    SPACE
00754A B79C BD B6EE  A INHALF JSR    PACK2
00755A B79F A7 02    A        STAA   2,X
00756A B7A1 BD B7C0  A        JSR    SPACE
00757A B7A4 BD B6EE  A        JSR    PACK2
00758A B7A7 A7 03    A        STAA   3,X
00759A B7A9 39                RTS
00760                          *
00761                          *
00762A B7AA CE B486  A IN2OPR LDX    #XN2
00763A B7AD BD F024  A        JSR    PDATA
00764A B7B0 CE 0804  A        LDX    #$804
00765A B7B3 20 CE B783        BRA    INSUB
00766                          *
00767                          *
00768A B7B5 CE B479  A IN1X2  LDX    #XN1
00769A B7B8 BD F024  A        JSR    PDATA
00770A B7BB CE 07FE  A IN1E2  LDX    #$7FE
00771A B7BE 20 DC B79C        BRA    INHALF
00772                          *
00773                          *
00774                          *
00775A B7C0 FF 0850  A SPACE  STX    XKEEP
00776A B7C3 CE B4B5  A        LDX    #XN6
00777A B7C6 BD F027  A        JSR    PDATA1
00778A B7C9 FE 0850  A        LDX    XKEEP
00779A B7CC 39                RTS
```

```
00781                   ************************************************************
00782                   *PROGRAM FOR USE WITH AM9511 A.P.U.
00783                   ************************************************************
00784                   *
00785                   *GENERAL CALLING SEQUENCE:-
00786                   *
00787                   *  JSR  XXXXX     XXXXX=FUNCTION NAME
00788                   *  FDB  ADD1      ADDRESS OF 1ST OPERAND
00789                   *  FDB  ADD2      ADDRESS OF 2ND OPERAND
00790                   *  FDB  ADDR      ADDRESS OF RESULT
00791                   *  -----          RETURN
00792                   *
00793                   *THERE MAY BE 0,1,OR 2 OPERANDS
00794                   *
00795                   *FUNCTIONS AVAILABLE ARE:-
00796                   *
00797                   * DXADD  ADDITION       (32-BIT FIXED POINT)
00798                   * DXSUB  SUBTRACTION    ( "    " )
00799                   * DXMULL MULTIPLICATION ( "    " ) RES=LO.HALF
00800                   * DXMULH MULTIPLICATION ( "    " ) RES=HI.HALF
00801                   * DXDIV  DIVISION       ( "    " )
00802                   * PUPI   CONSTANT PI
00803                   * FPADD  ADDITION
00804                   * FPSUB  SUBTRACTION
00805                   * FPMUL  MULTIPLICATION
00806                   * FPDIV  DIVISION
00807                   * FSQRT  SQUARE ROOT
00808                   * PWRX   POWER  Y**X
00809                   * LOG10  LOG  BASE 10
00810                   * LOGE   LOG  BASE E
00811                   * EXPX   EXPONENTIAL  E**X
00812                   *
00813                   * FPSIN  SINE
00814                   * FPCOS  COSINE
00815                   * FPTAN  TANGENT
00816                   * FPASIN ARCSINE
00817                   * FPACOS ARCCOSINE
00818                   * FPATAN ARCTANGENT
00819                   *
00820                   * CXTFS  CONVERT 16-BIT FX.PT TO 32-BIT FL.PT
00821                   * CFTXS  CONVERT 32-BIT FL.PT TO 16-BIT FX.PT
00822                   * CXTFD  CONVERT 32-BIT FX.PT TO FL.PT
00823                   * CFTXD  CONVERT 32-BIT FL.PT TO FX.PT
00824                   *
00825                   * CSGNF  CHANGE SIGN OF OPERAND LOADED
00826                   * FETCH  RETRIEVE TOP-OF-STACK
00827                   * EXCHF  RETRIEVE NEXT-ON-STACK
00828                   *
00829                   *
00830                   *THE TOP-OF-STACK NUMBER IS ALWAYS RETRIEVED
00831                   *AND STORED AT "ADDR"
00832                   *
00833                   *ALSO:-
00834                   * CLEAN  CLEARS THE A.P.U. STACK
00835                   *
00836                   ************************************************************
```

```
00838A B800                    ORG    $B800


00840                          ********************************************************
00841                          *FP.APU  PROGRAM ENTRY POINTS
00842                          ********************************************************
00843                          *
00844A B800 C6 1A    A PUPI    LDAB   #$1A     1A IS OPCODE FOR "PI".
00845A B802 30         LDCOM   TSX
00846A B803 EE 00    A         LDX    0,X .
00847A B805 09                 DEX
00848A B806 09                 DEX
00849A B807 FF 0850  A         STX    XKEEP
00850A B80A 20 40 B84C         BRA    STEP1
00851                          *
00852A B80C C6 2C    A DXADD   LDAB   #$2C
00853A B80E 20 6F B87F         BRA    LOAD42
00854                          *
00855A B810 C6 2D    A DXSUB   LDAB   #$2D
00856A B812 20 6B B87F         BRA    LOAD42
00857                          *
00858A B814 C6 2E    A DXMULL  LDAB   #$2E
00859A B816 20 67 B87F         BRA    LOAD42
00860                          *
00861A B818 C6 36    A DXMULH  LDAB   #$36
00862A B81A 20 63 B87F         BRA    LOAD42
00863                          *
00864A B81C C6 2F    A DXDIV   LDAB   #$2F
00865A B81E 20 5F B87F         BRA    LOAD42
00866                          *
00867A B820 C6 10    A FPADD   LDAB   #$10     ADDITION
00868A B822 20 5B B87F         BRA    LOAD42
00869                          *
00870A B824 C6 11    A FPSUB   LDAB   #$11     SUBTRACTION
00871A B826 20 57 B87F         BRA    LOAD42
00872                          *
00873A B828 C6 12    A FPMUL   LDAB   #$12     MULTIPLICATION
00874A B82A 20 53 B87F         BRA    LOAD42
00875                          *
00876A B82C C6 13    A FPDIV   LDAB   #$13     DIVISION
00877A B82E 20 4F B87F         BRA    LOAD42
00878                          *
00879A B830 C6 01    A FSQRT   LDAB   #$01     SQ.ROOT
00880A B832 20 46 B87A         BRA    LOAD41
00881                          *
00882A B834 C6 0B    A PWRX    LDAB   #$0B     POWERS
00883A B836 20 47 B87F         BRA    LOAD42
00884                          *
00885A B838 C6 08    A LOG10   LDAB   #$08     LOG 10
00886A B83A 20 3E B87A         BRA    LOAD41
00887                          *
00888A B83C C6 09    A LOGE    LDAB   #$09     LOG E
00889A B83E 20 3A B87A         BRA    LOAD41
00890                          *
00891A B840 C6 0A    A EXPX    LDAB   #$0A     EXPONENTIAL
00892A B842 20 36 B87A         BRA    LOAD41
00893                          *
```

```
00894A B844 C6 00    A FETCH  LDAB   #$00     RETRIEVE TOS
00895A B846 20 BA B802        BRA    LDCOM
00896                         *
00897A B848 C6 19    A EXCHF  LDAB   #$19     RETRIEVE NOS
00898A B84A 20 B6 B802        BRA    LDCOM
00899                         *
00900                         *
00901A B84C 20 6B B8B9 STEP1  BRA    LOADPI
00902                         *
00903                         *
00904A B84E C6 1E    A CFTXD  LDAB   #$1E     FL.PT TO 32-BIT FX.PT
00905A B850 20 28 B87A        BRA    LOAD41
00906                         *
00907A B852 C6 1C    A CXTFD  LDAB   #$1C     32-BIT FX.PT TO FL.PT
00908A B854 20 24 B87A        BRA    LOAD41
00909                         *
00910A B856 C6 15    A CSGNF  LDAB   #$15     CH.SIGN OF OPERAND LOADED
00911A B858 20 20 B87A        BRA    LOAD41
00912                         *
00913                         *
00914A B85A C6 02    A FPSIN  LDAB   #$02     SINE
00915A B85C 20 1C B87A        BRA    LOAD41
00916                         *
00917A B85E C6 03    A FPCOS  LDAB   #$03     COSINE
00918A B860 20 18 B87A        BRA    LOAD41
00919                         *
00920A B862 C6 04    A FPTAN  LDAB   #$04     TANGENT
00921A B864 20 14 B87A        BRA    LOAD41
00922                         *
00923A B866 C6 05    A FPASIN LDAB   #$05     ARCSINE
00924A B868 20 10 B87A        BRA    LOAD41
00925                         *
00926A B86A C6 06    A FPACOS LDAB   #$06     ARCCOSINE
00927A B86C 20 0C B87A        BRA    LOAD41
00928                         *
00929A B86E C6 07    A FPATAN LDAB   #$07     ARCTANGENT
00930A B870 20 08 B87A        BRA    LOAD41
00931                         *
00932A B872 C6 1D    A CXTFS  LDAB   #$1D     C.16-BIT FX.PT TO FL.PT
00933A B874 20 71 B8E7        BRA    KXTFS
00934                         *
00935A B876 C6 1F    A CFTXS  LDAB   #$1F     C.FL.PT TO 16-BIT FX.PT
00936A B878 20 77 B8F1        BRA    KFTXS
00937                         *


00939                         ************************************************************
00940                         *
00941                         *EXTRA ENTRY POINTS MUST BE PLACED BEFORE "PUPI"
00942                         *
00943                         ************************************************************
```

```
00945                    ************************************************************
00946                    *FP.APU  MAIN PROGRAM-LOADING OF OPERAND(S)
00947                    ************************************************************
00948                    *
00949A B87A 30           LOAD41 TSX
00950A B87B EE 00     A         LDX    0,X
00951A B87D 20 21 B8A0          BRA    LOAD4C



00953A B87F 30           LOAD42 TSX
00954A B880 EE 00     A         LDX    0,X
00955A B882 FF 0850   A         STX    XKEEP
00956A B885 EE 00     A         LDX    0,X

00958A B887 A6 03     A         LDAA   3,X
00959A B889 B7 E700   A         STAA   INOUT
00960A B88C A6 02     A         LDAA   2,X
00961A B88E B7 E700   A         STAA   INOUT
00962A B891 A6 01     A         LDAA   1,X
00963A B893 B7 E700   A         STAA   INOUT
00964A B896 A6 00     A         LDAA   0,X
00965A B898 B7 E700   A         STAA   INOUT


00967A B89B FE 0850   A         LDX    XKEEP
00968A B89E 08                  INX
00969A B89F 08                  INX


00971A B8A0 FF 0850   A LOAD4C  STX    XKEEP
00972A B8A3 EE 00     A         LDX    0,X

00974A B8A5 A6 03     A         LDAA   3,X
00975A B8A7 B7 E700   A         STAA   INOUT
00976A B8AA A6 02     A         LDAA   2,X
00977A B8AC B7 E700   A         STAA   INOUT
00978A B8AF A6 01     A IXTFS   LDAA   1,X
00979A B8B1 B7 E700   A         STAA   INOUT
00980A B8B4 A6 00     A         LDAA   0,X
00981A B8B6 B7 E700   A         STAA   INOUT
```

```
00983                    ************************************************************
00984                    *FP.APU MAIN PROGRAM-LOAD COMMANDS AND                     *
00985                    *LOOK FOR ERRORS.  RETRIEVE ANSWER                         *
00986                    ************************************************************
00987                    *
00988A B8B9 F7 E704   A LOADPI STAB   COMND



00990A B8BC B6 E704   A LOOP1  LDAA   COMND
00991A B8BF 48                 ASLA
00992A B8C0 25 FA B8BC         BCS    LOOP1
00993A B8C2 46                 RORA
00994A B8C3 84 1E     A        ANDA   #%00011110
00995A B8C5 26 5D B924         BNE    WRONG



00997A B8C7 FE 0850   A        LDX    XKEEP
00998A B8CA EE 02     A        LDX    2,X

01000A B8CC B6 E700   A        LDAA   INOUT
01001A B8CF A7 00     A        STAA   0,X
01002A B8D1 B6 E700   A        LDAA   INOUT
01003A B8D4 A7 01     A        STAA   1,X
01004A B8D6 B6 E700   A IFTXS  LDAA   INOUT
01005A B8D9 A7 02     A        STAA   2,X
01006A B8DB B6 E700   A        LDAA   INOUT
01007A B8DE A7 03     A        STAA   3,X



01009A B8E0 FE 0850   A        LDX    XKEEP
01010A B8E3 31                 INS
01011A B8E4 31                 INS
01012A B8E5 6E 04     A        JMP    4,X

01014                    ************************************************************
01015                    *FP.APU SUB-PROGRAM FOR CXTFS
01016                    ************************************************************
01017                    *
01018A B8E7 30           KXTFS  TSX
01019A B8E8 EE 00     A        LDX    0,X
01020A B8EA FF 0850   A        STX    XKEEP
01021A B8ED EE 00     A        LDX    0,X
01022A B8EF 20 BE B8AF         BRA    IXTFS
01023                    *
01024                    ************************************************************
```

```
01026                          ************************************************************
01027                          *FP.APU SUB-PROGRAM FOR CFTXS
01028                          ************************************************************
01029                          *
01030A B8F1 30                 KFTXS  TSX
01031A B8F2 EE 00     A               LDX    0,X
01032A B8F4 FF 0850   A               STX    XKEEP
01033A B8F7 EE 00     A               LDX    0,X


01035A B8F9 A6 03     A               LDAA   3,X
01036A B8FB B7 E700   A               STAA   INOUT
01037A B8FE A6 02     A               LDAA   2,X
01038A B900 B7 E700   A               STAA   INOUT
01039A B903 A6 01     A               LDAA   1,X
01040A B905 B7 E700   A               STAA   INOUT
01041A B908 A6 00     A               LDAA   0,X
01042A B90A B7 E700   A               STAA   INOUT


01044A B90D F7 E704   A               STAB   COMND


01046A B910 B6 E704   A LOOP2  LDAA   COMND
01047A B913 48                 ASLA
01048A B914 25 FA B910         BCS    LOOP2
01049A B916 46                 RORA
01050A B917 84 1E     A               ANDA   #%00011110
01051A B919 26 09 B924         BNE    WRONG


01053A B91B FE 0850   A               LDX    XKEEP
01054A B91E EE 02     A               LDX    2,X
01055A B920 09                 DEX
01056A B921 09                 DEX
01057A B922 20 B2 B8D6         BRA    IFTXS
```

```
01059                          ************************************************************
01060                          *FP.APU-IDENTIFY ERRORS
01061                          ************************************************************
01062                          *
01063A B924 47          WRONG  ASRA
01064A B925 47                 ASRA
01065A B926 24 08 B930         BCC    NEXT1
01066A B928 CE B96B  A         LDX    #MSS10
01067A B92B BD F024  A         JSR    PDATA
01068A B92E 20 29 B959         BRA    EEND
01069                          *
01070A B930 47          NEXT1  ASRA
01071A B931 24 08 B93B         BCC    NEXT2
01072A B933 CE B979  A         LDX    #MSS20
01073A B936 BD F024  A         JSR    PDATA
01074A B939 20 1E B959         BRA    EEND
01075                          *
01076A B93B 81 03    A NEXT2   CMPA   #$03
01077A B93D 26 08 B947         BNE    NEXT3
01078A B93F CE B988  A         LDX    #MSS30
01079A B942 BD F024  A         JSR    PDATA
01080A B945 20 12 B959         BRA    EEND
01081                          *
01082A B947 81 01    A NEXT3   CMPA   #$01
01083A B949 26 08 B953         BNE    NEXT4
01084A B94B CE B9A4  A         LDX    #MSS40
01085A B94E BD F024  A         JSR    PDATA
01086A B951 20 06 B959         BRA    EEND
01087                          *
01088A B953 CE B9C8  A NEXT4   LDX    #MSS50
01089A B956 BD F024  A         JSR    PDATA
01090                          *
01091A B959 CE 0850  A EEND    LDX    #XKEEP
01092A B95C BD F01E  A         JSR    OUT4HS
01093A B95F BD F021  A         JSR    PCRLF
01094A B962 CE B9DC  A         LDX    #MSS60
01095A B965 BD F027  A         JSR    PDATA1
01096A B968 7E F0F3  A         JMP    HOME
01097                          *
01098A B96B    4F    A MSS10   FCC    /OVERFLOW AT  /
01099A B978    04    A         FCB    4
01100A B979    55    A MSS20   FCC    /UNDERFLOW AT  /
01101A B987    04    A         FCB    4
01102A B988    41    A MSS30   FCC    /ARGUEMENT OUT OF RANGE AT  /
01103A B9A3    04    A         FCB    4
01104A B9A4    4E    A MSS40   FCC    /NEGATIVE ARGUEMENT NOT ALLOWED AT
01105A B9C7    04    A         FCB    4
01106A B9C8    44    A MSS50   FCC    /DIVIDE BY ZERO AT  /
01107A B9DB    04    A         FCB    4
01108A B9DC    07    A MSS60   FCB    7,7,7,7,4
01109                          *
```

```
01111                    ************************************************************
01112                    *
01113           E700  A INOUT  EQU    $E700
01114           E704  A COMND  EQU    $E704
01115           0850  A XKEEP  EQU    $850
01116           F024  A PIATA  EQU    $F024
01117           F027  A PIATA1 EQU    $F027
01118           F01E  A OUT4HS EQU    $F01E
01119           F021  A PCRLF  EQU    $F021
01120           F0F3  A HOME   EQU    $F0F3
01121                    *

01123                    ************************************************************
01124                    *CLEAN - CLEAR OUT APU STACK
01125                    ************************************************************
01126                    *
01127A B9E1 C6 17     A CLEAN  LDAB   #$17
01128A B9E3 4F                 CLRA
01129A B9E4 B7 E700   A        STAA   INOUT
01130A B9E7 B7 E700   A        STAA   INOUT
01131A B9EA B7 E700   A        STAA   INOUT
01132A B9ED B7 E700   A        STAA   INOUT
01133A B9F0 F7 E704   A        STAB   COMND
01134A B9F3 F7 E704   A        STAB   COMND
01135A B9F6 F7 E704   A        STAB   COMND

01137A B9F9 39                 RTS
01138                    *
01139                    *
01140                          END
TOTAL ERRORS 00000
```

```
B0EB AGAIN  00167 00170*
B84E CFTXD  00404 00904*
B876 CFTXS  00384 00935*
B9E1 CLEAN  00180 01127*
E704 COMND  00988 00990 01044 01046 01114*01133 01134 01135
B856 CSGNF  00414 00910*
B852 CXTFD  00394 00907*
B872 CXTFS  00374 00932*
B80C DXADD  00441 00852*
B81C DXDIV  00493 00864*
B818 DXMULH 00479 00861*
B814 DXMULL 00467 00858*
B810 DXSUB  00455 00855*
B959 EEND   01068 01074 01080 01086 01091*
B71A ERR0R  00669 00673*
B745 ERR1R  00700 00704*
B755 ERR4R  00663 00667 00674 00686 00716*
B754 ERR5R  00694 00698 00705 00712 00715*
B775 ERR6R  00728 00732*
B848 EXCHF  00431 00897*
B840 EXPX   00292 00891*
B844 FETCH  00423 00894*
```

```
B802 LDCOM  00845*00895 00898
B87A LOAD41 00880 00886 00889 00892 00905 00908 00911 00915 00918
            00921 00924 00927 00930 00949*
B87F LOAD42 00853 00856 00859 00862 00865 00868 00871 00874 00877
            00883 00953*
B8A0 LOAD4C 00951 00971*
B8B9 LOADPI 00901 00988*
B838 LOG10  00272 00885*
B83C LOGE   00282 00888*
B8BC LOOP1  00990*00992
B910 LOOP2  01046*01048
B6CB MOVE2  00181 00612*
B96B MSS10  01066 01098*
B979 MSS20  01072 01100*
B988 MSS30  01078 01102*
B9A4 MSS40  01084 01104*
B9C8 MSS50  01088 01106*
B9DC MSS60  01094 01108*
B930 NEXT1  01065 01070*
B93B NEXT2  01071 01076*
B947 NEXT3  01077 01082*
B953 NEXT4  01083 01088*
B009 NEXT8  00036*00170 00182 00190 00202 00216 00228 00242 00252
            00266 00276 00286 00296 00308 00320 00332 00344 00356
            00368 00378 00388 00398 00408 00418 00426 00434 00446
            00460 00472 00484 00498 00681
B003 NEXT9  00034*00164
F01B OUT2HS 00026*00610 00611 00619 00620 00621 00622
F01E OUT4HS 01092 01118*
B6BC OUTPT2 00387 00607*
B6D2 OUTPT4 00189 00201 00215 00227 00241 00251 00265 00275 00285
            00295 00307 00319 00331 00343 00355 00367 00377 00397
            00407 00417 00425 00433 00445 00459 00471 00483 00497
            00616*
B6EE PACK2  00634*00745 00751 00754 00757
F021 PCRLF  00036 01093 01119*
F024 PDATA  00035 00038 00179 00186 00194 00206 00208 00220 00232
            00234 00246 00256 00258 00270 00280 00290 00300 00302
            00312 00314 00324 00326 00336 00338 00348 00350 00360
            00362 00372 00382 00392 00402 00412 00422 00430 00438
            00450 00452 00464 00476 00488 00490 00608 00613 00617
            00720 00743 00763 00769 01067 01073 01079 01085 01089
            01116*
F027 PDATA1 00624 00718 00749 00777 01095 01117*
B800 PUPI   00187 00844*
B834 PWRX   00261 00882*
0808 RES    00025*00609 00618
B2B0 RTN    00374*00727
B71E RTN10  00675*00707
B01C SP1    00048 00051*
B023 SP2    00052 00055*
B02A SP3    00056 00059*
B031 SP4    00060 00063*
B038 SP5    00064 00067*
B03F SP6    00068 00071*
B046 SP7    00072 00075*
B04D SP8    00076 00079*
B054 SP9    00080 00083*
```

```
B05B SPA     00084 00087*
B7C0 SPACE   00753 00756 00775*
B062 SPB     00088 00091*
B069 SPC     00092 00095*
B070 SPD     00096 00099*
B077 SPE     00100 00103*
B07E SPF     00104 00107*
B085 SPG     00108 00111*
B08C SPH     00112 00115*
B093 SPI     00116 00119*
B09A SPJ     00120 00123*
B0A1 SPK     00124 00127*
B0A8 SPL     00128 00131*
B0AF SPM     00132 00135*
B0B6 SPN     00136 00139*
B0BD SPO     00140 00143*
B0C4 SPP     00144 00147*
B0CB SPQ     00148 00151*
B0D2 SPR     00152 00155*
B0D9 SPS     00156 00159*
B0E4 SPZ     00163 00166*
B84C STEP1   00850 00901*
B704 TEST91  00636 00658*
B72F TEST92  00645 00689*
B0EE TP1     00160 00178*
B1D3 TP10    00069 00279*
B1E9 TP11    00073 00289*
B1FF TP12    00081 00299*
B21B TP13    00085 00311*
B237 TP14    00089 00323*
B253 TP15    00093 00335*
B26F TP16    00097 00347*
B28B TP18    00101 00359*
B2A7 TP19    00125 00371*
B0FD TP2     00141 00185*
B2BD TP20    00133 00381*
B2D3 TP21    00129 00391*
B2E9 TP22    00137 00401*
B2FF TP23    00153 00411*
B315 TP24    00157 00421*
B326 TP25    00149 00429*
B337 TP26    00105 00437*
B352 TP27    00109 00449*
B373 TP28    00113 00463*
B38E TP29    00117 00475*
B10E TP3     00049 00193*
B3A9 TP30    00121 00487*
B129 TP4     00053 00205*
B14A TP5     00057 00219*
B165 TP6     00061 00231*
B186 TP7     00145 00245*
B19C TP8     00077 00255*
B1BD TP9     00065 00269*
B924 WRONG   00995 01051 01063*
0850 XKEEP   00716 00729 00732 00747 00750 00775 00778 00849 00955
             00967 00971 00997 01009 01020 01032 01053 01091 01115*
B479 XN1     00563*00742 00768
B515 XN10    00301 00313 00325 00337 00349 00361 00578*
```

```
B527 XN11    00580*00717
B534 XN12    00582*00719
B541 XN13    00584*00748
B543 XN16    00034 00585*
B6B9 XN17    00037 00599*
B486 XN2     00565*00762
B493 XN3     00567*00607 00616
B4A0 XN4     00569*00623
B4B0 XN5     00570*00612
B4B5 XN6     00571*00776
B4B7 XN7     00207 00451 00572*
B4D6 XN8     00233 00489 00574*
B4F5 XN9     00257 00576*
B727 YES3    00660 00683*
B74C YES33   00691 00709*
```

=

Appendix E

Another example of a chained calculation is given here, in which both roots of a quadratic equation are found, with a three to four-fold improvement in processing time. The calculation can be done in as little as 1.8ms, compared with over 6ms using the software interface, and about 60ms using a conventional software "maths pack".

The contents of the APU data stack are shown at each step in the calculation in table 9 . It is important to keep track of operands in the stack and to ensure that the function being used does not interfere with other stack locations being used as temporary storage locations.

For brevity, the following names are used referring to the various program steps already outlined:-

LDO	a	load operand a
LDC	XXXX	load command XXXX
BSY		wait whilst busy
CKE		check for errors
RTR	y	retrieve result y.

The following symbols are used to represent intermediate results:-

$$\alpha = b^2 - 4ac$$

$$\phi_1 = -b + \sqrt{\alpha}$$

$$\phi_2 = -b - \sqrt{\alpha}$$

and the results are:-

$$k_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} = \frac{\phi_1}{2a}$$

$$k_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} = \frac{\phi_2}{2a}$$

where a,b, and c are coefficients in the quadratic equation

$$ax^2 + bx + c = 0$$

In table 9 , the columns give the operation, operand and execution time in microseconds for each step of the calculation, together with the contents of the top three locations of the APU data stack at salient points. The fourth location, bottom of stack, is not shown since it is used as scratch working space by many of the operations of the APU. However, although not reported by the AMD literature on the Am9511, pulling a result off the stack causes it to reappear at the bottom of the stack. Thus the two RTR operations will leave valid data in bottom of stack.

Table 9

Operation	Operand	Execution Time, μsec	Top of Stack	Next on Stack	3rd on Stack
LDO	4	36	4		
LDC	CHSF	7	-4		
BSY		10			
LDO	a	36	a	-4	
LDC	FMUL	7	-4a		
BSY		80-90			
CKE		6			
LDO	c	36	c	-4a	
LDC	FMUL	7	-4ac		
BSY		80-90			
CKE		6			
LDO	b	36	b	-4ac	
LDC	PTOF	7	b	b	-4ac
BSY		20			
LDC	FMUL	7	b^2	-4ac	
BSY		80-90			
CKE		6			
LDC	FADD	7	α		
BSY		30-190			
CKE		6			
LDC	SQRT	7	$\sqrt{\alpha}$		
BSY		400-440			
CKE		6			
LDC	PTOF	7	$\sqrt{\alpha}$	$\sqrt{\alpha}$	
BSY		20			
LDC	CHSF	7	$-\sqrt{\alpha}$	$\sqrt{\alpha}$	
BSY		10			
LDO	b	36	b	$-\sqrt{\alpha}$	$\sqrt{\alpha}$
LDC	FSUB	7	ϕ_2	$\sqrt{\alpha}$	
BSY		40-190			
CKE		6			
LDC	XCHF	7	$\sqrt{\alpha}$	ϕ_2	
BSY		20			
LDO	b	36	b	$\sqrt{\alpha}$	ϕ_2
LDC	FSUB	7	ϕ_1	ϕ_2	

Table 9 (continued)

Operation	Operand	Execution Time, μsec	Top of Stack	Next on Stack	3rd on Stack
BSY		40-190			
CKE		6			
LDO	2	36	2	ϕ_1	ϕ_2
LDC	FDIV	7	$\phi_1/2$	ϕ_2	
BSY		80-100			
CKE		6			
LDO	a	36	a	$\phi_1/2$	ϕ_2
LDC	FDIV	7	$\phi_1/2a$	ϕ_2	
BSY		80-100			
CKE		6			
RTR	k_1	36	ϕ_2		
LDO	2	36	2	ϕ_2	
LDC	FDIV	7	$\phi_2/2$		
BSY		80-100			
CKE		6			
LDO	a	36	a	$\phi_2/2$	
LDC	FDIV	7	$\phi_2/2a$		
BSY		80-100			
CKE		6			
RTR	k_2	36			

Total execution time 1800μs minimum; 2400μs maximum.

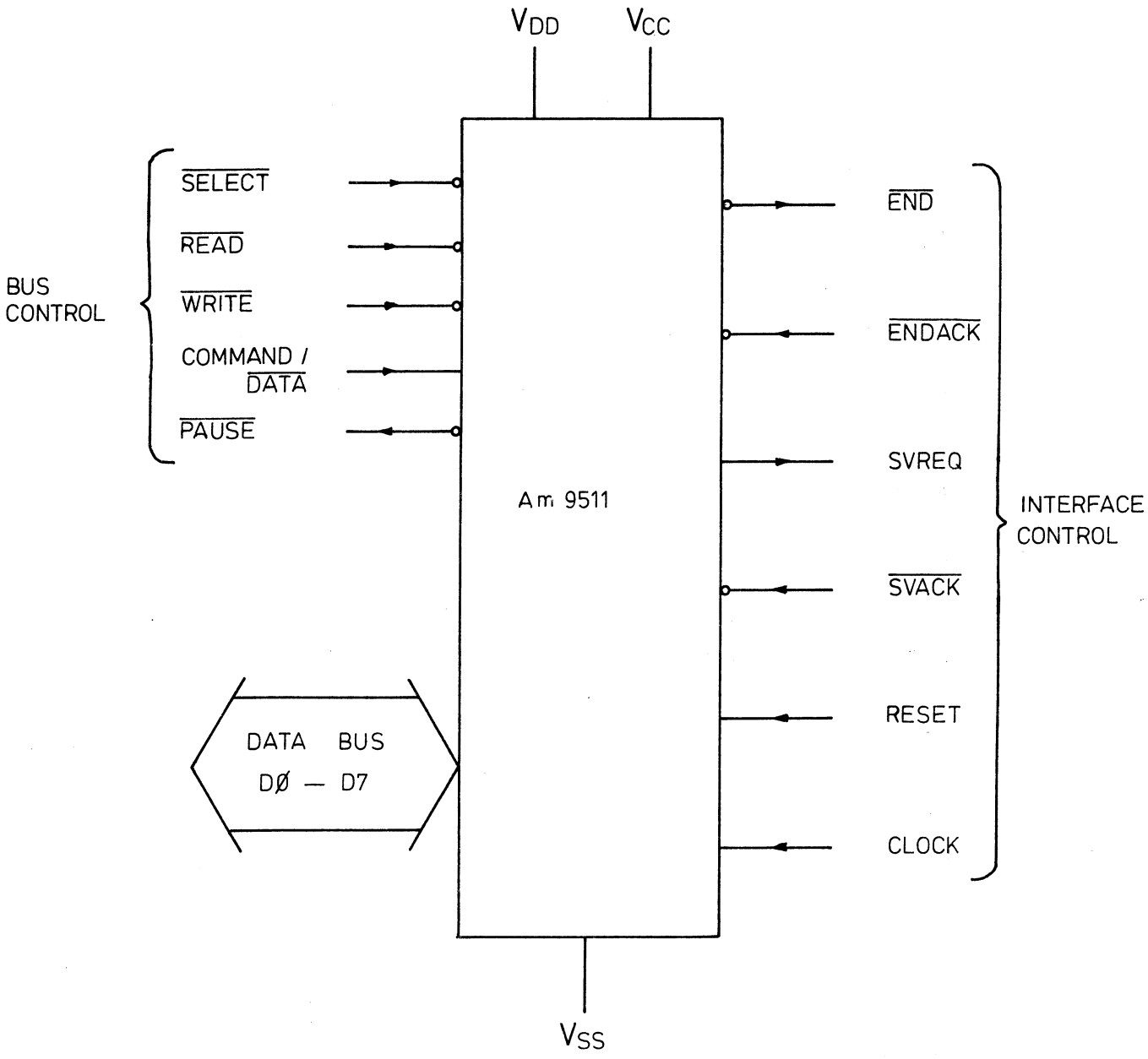


FIG. 1.

APU HARDWARE INTERFACING MODEL.

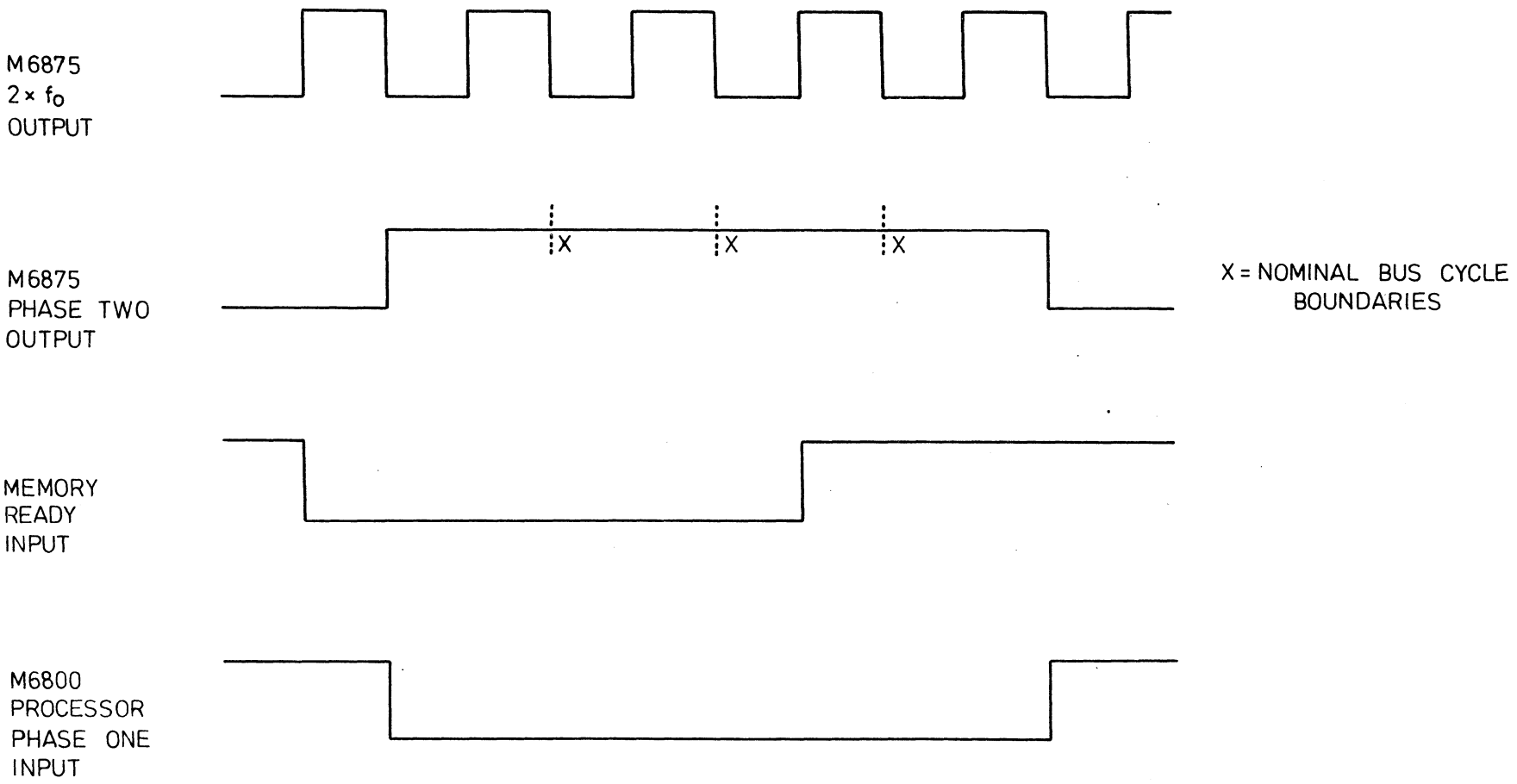
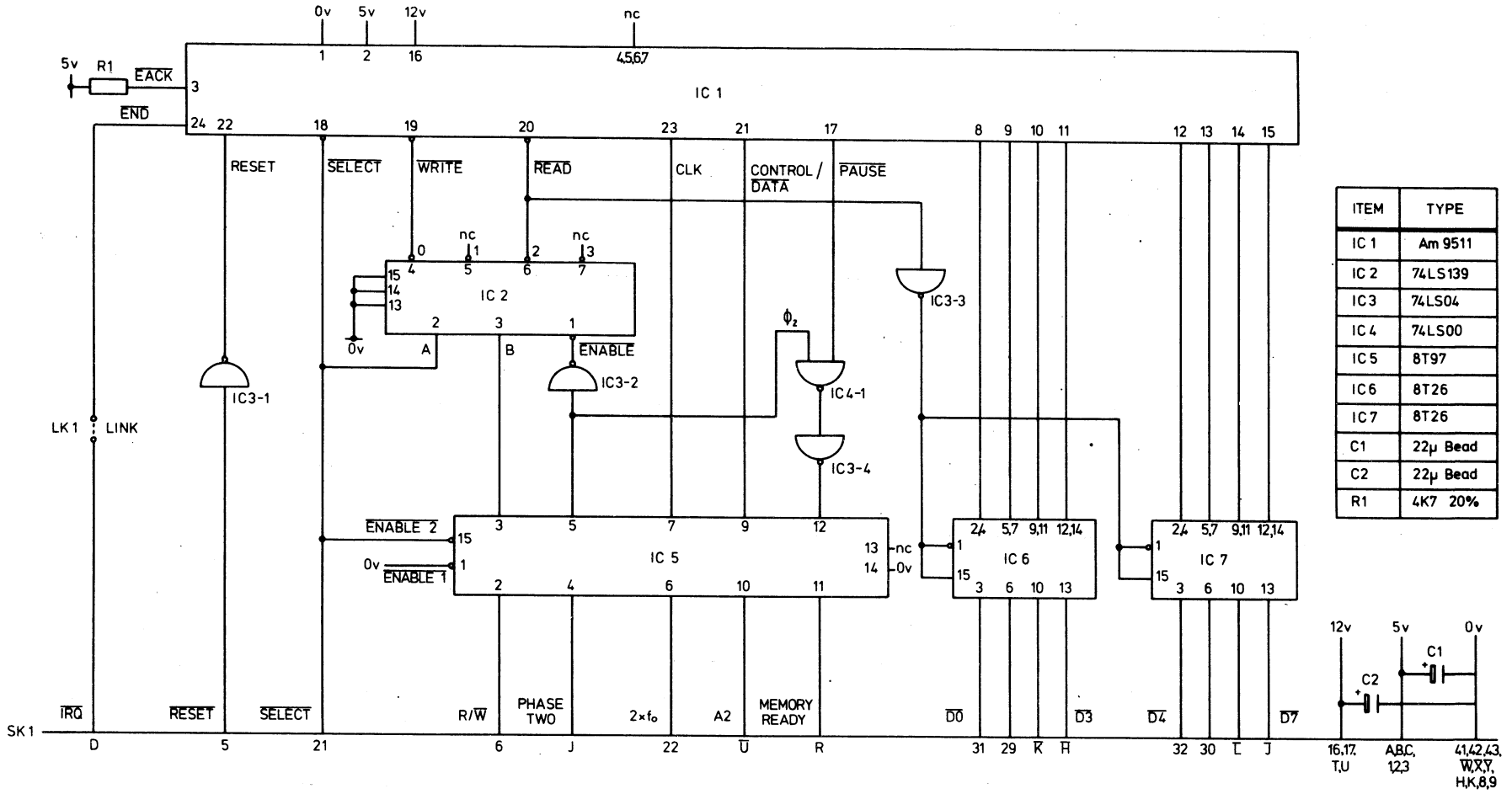


FIG. 2.

MEMORY READY OPERATION.

DRG No C

DISCRETE COMPONENTS OUT OF NUMERICAL SEQUENCE ARE LISTED ABOVE



ITEM	TYPE
IC 1	Am 9511
IC 2	74LS139
IC 3	74LS04
IC 4	74LS00
IC 5	8T97
IC 6	8T26
IC 7	8T26
C1	22μ Bead
C2	22μ Bead
R1	4K7 20%

FIG. 3. INTERFACE CIRCUIT.

SYMBOLS & COMPONENT REFERENCES CONFORM TO AECP (R) 50

LOGIC	NOT FOR PUBLICATION THE INFORMATION ON THIS DRAWING IS NOT TO BE COMMUNICATED EITHER DIRECTLY OR INDIRECTLY TO THE PRESS OR TO ANY PERSON NOT AUTHORISED TO RECEIVE IT	JOB No	DRN		
		PROJ No	CHKD		
			APPD	ISSUE / DATE / MOD	
	U.K.A.E.A. RESEARCH GROUP		CONTRACTOR CONTR S DRG REF		
USED ON	TITLE CIRCUIT DIAGRAM		DRG No C		

OZALID 1283/4

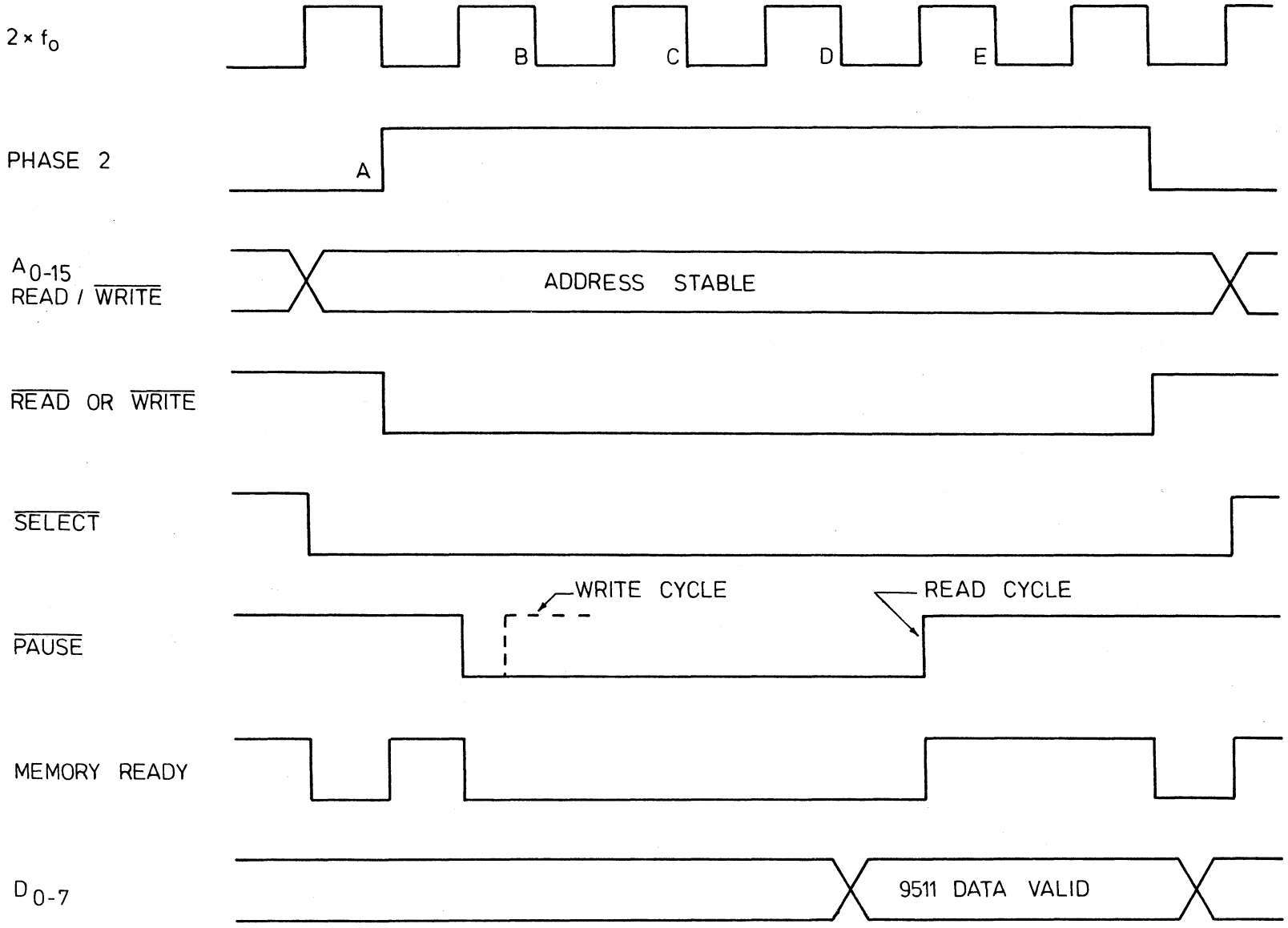
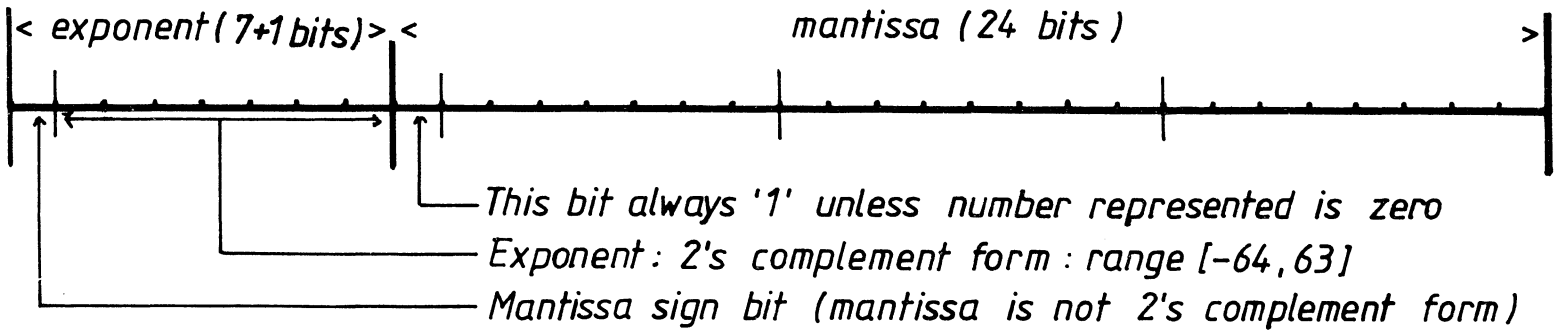


FIG. 4.

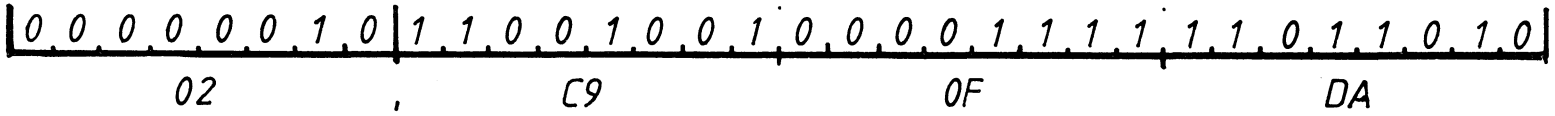
Am 9511 TO MC6800 INTERFACE TIMING.

FIGURE 5 Format of floating point data.



Example : floating point representation of TT.

TT is written as : 02.C9 0F DA :-



$$= 2^{2} (2^{-1}+2^{-2}+2^{-5}+2^{-8}+2^{-13}+2^{-14}+2^{-15}+2^{-16}+2^{-17}+2^{-18}+2^{-20}+2^{-21}+2^{-23})$$

$$= 4 \times 0{,}785\,398\,125\,6$$

$$= 3{,}141\,592\,503$$

TT, value from calculator = 3,141 592 653 6

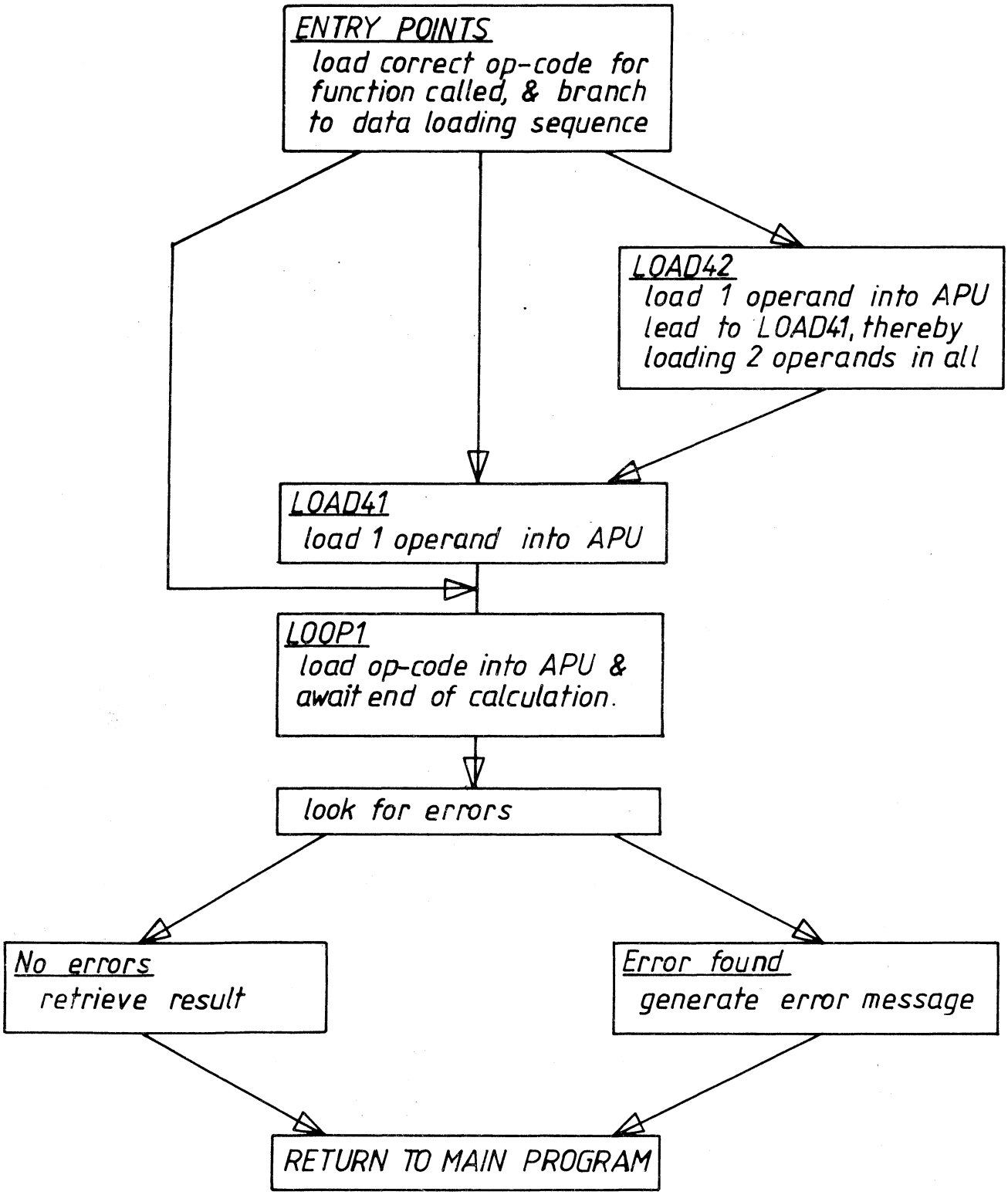


Figure 6 OVERALL VIEW FLOWCHART for AM952E program