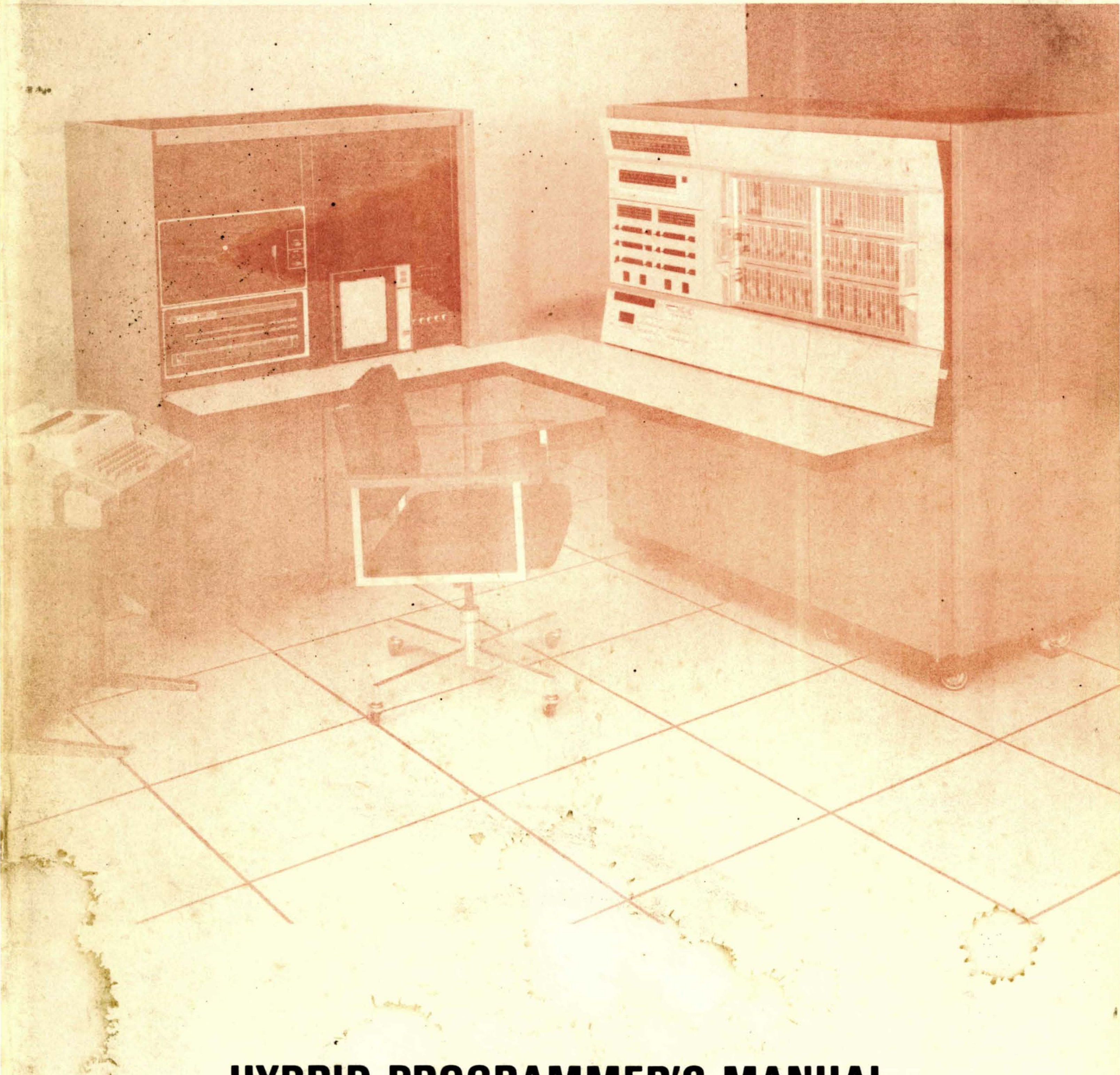
APPLIED DYNAMICS / FIVE



HYBRID PROGRAMMER'S MANUAL

APPLIED DYNAMICS ANALOG/HYBRID SYSTEMS

AD/FIVE

HYBRID COMMUNICATIONS ROUTINES

OPERATORS MANUAL

TABLE OF CONTENTS

HCR OPERATORS MANUAL

ANALOG DATA

FUNCTION ROUTINES

APPENDIX A

PAGE

1.0 INTRODUCTION

1.1 The following routines provide hybrid communications within an AD/FIVE system. They are referred to as HCR's (Hybrid Communication Routines). The routines satisfy the following basic requirements:

a.) The HCR's are USASI Basic Fortran compatible, written in assembly language for each system.

b.) Fixed argument lists are used in all calls to HCR's, therefore all arguments must be present.

c.) All hybrid data will be two's complement and right justified in the word, with the sign bit extended where necessary.

1.2 The HCR's may be separated into two distinct categories, depending upon the interface between the digital and the AD/FIVE. These two categories are basic hybrid routines and hybrid expansion routines.

The basic hybrid category consists of routines which perform the "push-button" functions on the AD/FIVE: Mode Control, General Analog Control, and Reading and Setting Pots and DCU's. This package will be delivered with the basic interface.

The hybrid expansion routines provide communication with the expanded interface, which includes sense, control and interrupt register, MDAC's and ADC channels. All these devices are modules, which are added to the module line rack. Also included are two cycle stealing routines which provide sophisticated communication to the MDAC's and ADC channels, if the cycle stealing option is present in the Interface.

2.0 ERROR DETECTION

2.1 Error Mode Description

The Hybrid Communication Routines (HCR's) can be used in two modes, the Run Mode and the Test Mode. In the Run Mode there is no error detection and the error argument is not accessed so that all HCR's are as fast and efficient as possible.

All error detection is made in the Test Mode where normal program and problem debugging takes place. Whenever an error is detected, the assigned HCR code number and code representing the error type encountered are put into the error argument. This allows control of error lists and action based on program decisions. Immediately thereafter, rather than perform its function incorrectly due to the error, the HCR returns control to the calling program.

Whenever printing is requested, any error detected in an HCR also causes the error routine to print out a message to the operator on the principal print device. The message identifies the HCR in which the error was detected and specifies the type of error. After the error message is printed, control is returned to the calling program as in the non-print mode. The operator may interpret whatever error messages are printed after the program has run to completion.

2.2 Error Argument Word

Detection of an error will cause the following information to be placed within the error argument:

a.) A numeric code for the error type (see Section 2.2.d).

b.) A numeric code for the routine (see Table 1).

c.) In the case of the unique (pot setting) error, the pot address.

In the test mode, if no error is detected, the error argument is set to zero.

The documentation accompanying each set of HCR's for a specified digital computer contains the exact error word description.

TABLE 1: Code numbers assigned to the HCR's for general identification and for flagging in the error word. HCR's 1 - 27 perform testing when in test mode.

HYTST (M, IE)	0
INITA (N)	1
CONSO (N)	2
HON (N)	3
HOFF (N)	4

d.) Printed Error Messages

All error messages on the principal print device will be of the form:

ERROR - (Name of HCR) - (Type of Error)

The error types are:

CODE	TYPE	
0	ADDRESS	- The routine has asked for a non-valid address; or address is incompatible with routine.
1	DATA	- The range of an argument is exceeded.
2	BUSY	- The analog console addressed is in the BUSY mode and cannot be accessed.
3	CANNOT SET POT JKLM	- Where JKLM is the address of the pot.

2.3 FUNCTIONS

2.3.1 Functions are routines which return a number in the accumulator. The only values returned by functions in the Hybrid Communication Library are: plus one and zero. The format of the returned value is always integer. Functions can be used in the following ways:

```
IF (FUNCTION (K +2)), A, B, C
ALPHA = FUNCTION (4)
X = Y ** FUNCTION (I, K(3))
ETC.
```

2.3.2 All of the functions included in the HCR's accept only integer arguments. The arguments may be supplied in any standard way within the program.

2.4 SUBROUTINES

2.4.1 A subroutine can return one or many answers. These are stored in memory and may not even require an argument for operation. Subroutines are used by writing CALL SUBR (X, Y, ...) in which case subroutine SUBR is supplied with arguments X, Y, etc., and granted control of the computer. A check for compatibility between arguments supplied and those acceptable to the subroutine using them is performed, if possible. If any argument type incompatibility is found, an error statement is output.

2.5 ARGUMENTS

2.5.1 In the following description, the argument IE, where it appears, is the location to which error information is returned, when the system is in test mode. Most subroutines treat this argument as an integer without testing it; therefore, it should be specified as such within a program.

2.5.2 The argument D, which may appear following a function, is a dummy argument specified here to satisfy the requirements of Fortran. No form restriction applies, since it is ignored by the function.

2.5.3 Argument IA is a four-digit decimal integer representation of an address comprising the element class, field, area, and component number, in that order. The element class number is taken from the list in Table 2.

TABLE 2: Addressing List for AD/FIVE Components

Type (Class)	Element	Address
0	Amplifier	0KLM
1	Pot Coefficient	1KLM
2	Trunks	2KLM
3	Digital Coefficient	3KLM
4	Non-Linear	4KLM
5	Miscellaneous	5KLM

The element type is the first number of an address; the other three are:

K = 0 or 1	Field Number
L = 0 through 5	Area Number
M = 0 through 9	Component Number

2.5.4 Argument IV is fixed point 2's complement binary.

2.5.5 All other arguments are integer and are defined within the subroutine descriptions.

3.0 BASIC HYBRID ROUTINES

3.1 Fortran Callable Routines

3.1.1 General Control Routines

3.1.1.1 CALL HYTST (M, IE)

This routine, usually called at the start of a program, controls the flags which determine the testing modes. The flags, which are accessible to all hybrid routines, are set depending on the value of M. The argument IE is the error argument used in the Test mode.

Argument M	Resultant Flag Condition
0	Do not change mode - IE is new error word
1	Run mode - no error tests.
2	Test mode - test errors, place error code in word IE and return.
3	Test mode with print out - test errors, place error code in word IE, print out error message on the principal print device and return.

3.1.1.2 CALL INITA (N)

Argument N (0 through 7) specifies the logical console number. The routine sets all system console addressing to N, then initializes the following:

Sets: Time Scale - X1
Analog Mode - IC
Logic Mode - Load
Interval Timer uses thumb wheels
V-Signals to Interval Timer - 1 sec.
All other controls off
and, if modules present, sets:

All MDAC's to update immediate
All ADC channels to sample
Control Line Register to 0
Sense Line Register to 0

Error Messages:

(a) ERROR-INITA - BUSY: Occurs when console is busy.
(b) ERROR-INITA - DATA: Occurs when N is outside allowable range.

3.1.1.3 CALL CONSO (N)

This routine sets control number to N (0 to 7), to enable HCR's to communicate with AD/FIVE Console N.

Error Message:

(a) ERROR-CONSO - DATA: Occurs when N is outside allowable range.

3.1.1.4 CALL HON (N)

The routine HON allows the user to enable Hybrid Communication to AD/FIVE console N.

Error Message:

(a) ERROR-HON - DATA: Occurs if N is outside range.

3.1.1.5 CALL HOFF (N)

The HOFF routine allows the user to disable Hybrid Communication to AD/FIVE console N.

Error Message:

(a) ERROR-HOFF - DATA: Occurs if N is outside range.

3.1.2 Analog Data Routines

3.1.2.1 CALL READ (IA, IV)

The READ routine allows the user to read a digital representation of the analog voltage for any device which is addressable from the AD/FIVE keyboard.

Argument IA is a four digit integer address in which digits may range from 0 through 9 (see Table 2).

Argument IV is an integer variable, in which is returned the value measured at the address IA and read back through the DRM.

Error Messages:

(a) ERROR-READ - ADDRESS: Occurs when analog address is non-valid

(b) ERROR-READ - DATA: Occurs if number of arguments is incorrect.

(c) ERROR-READ - BUSY: Occurs when console is busy.

3.1.2.2 CALL READA (IA (I), IV (J), N)

Arguments IA and IV are arrays of data as described in READ. N is the count of the number of addresses to be read. Last array points used are IA (I+N-1) and IV (J+N-1). READA is identical to READ is all other respects.

Error Messages:

(a) ERROR-READA - ADDRESS: Occurs when address is non-valid.

(b) ERROR-READA - DATA: Occurs if number of arguments is incorrect.

(c) ERROR-READA - BUSY: Occurs when console is busy.

3.1.2.3 CALL STIND (IA, IV)

The routine STIND may be used to set a coefficient into any of the coefficient devices addressable from the AD/FIVE keyboard (Pots and DCU's).

Argument IA is a four digit integer address of a coefficient device in the AD/FIVE CONSOLE.

Argument IV is the coefficient to be set into the addressed device.

Error Messages:

(a) ERROR-STIND - ADDRESS: Occurs when analog address is non-valid.

(b) ERROR-STIND - DATA: Occurs if the number of arguments is incorrect or value is out of range.

(c) ERROR-STIND - BUSY: Occurs when console is busy.

(d) ERROR-STIND - CANNOT SET POT XXXX: Occurs if pot fails to set after two attempts.

3.1.2.4 CALL STINA (IA (I), IV (J), N)

Arguments IA and IV are arrays of data as described in STIND. N is the count of the number of successive array points to be used.

Last array points used are IA (I+N-1) and IV (J+N-1). STINA is otherwise identical to STIND.

Error Messages:

(a) ERROR-STINA - ADDRESS: Occurs when analog address is non-valid.

(b) ERROR-STINA - DATA: Occurs if number of arguments is incorrect or value is out of range.

(c) ERROR-STINA - BUSY: Occurs when console is busy.

(d) ERROR-STINA - CANNOT SET POT XXXX: Occurs if pot fails to set after two attempts.

3.1.3 Mode Control

These routines affect the modes of the analog computer which are defined by various pushbuttons and switches.

3.1.3.1 Analog Modes

	Mode
3.1.3.1.1 CALL IC	IC

The analog console is put into the Initial Condition mode.

Error Message:

ERROR-IC - BUSY: Occurs if console is busy.

3.1.3.1.2 CALL HOLD	HOLD

The analog console is put into the Hold mode.

Error Message:

ERROR-HOLD - BUSY: Occurs if console is busy.

	Mode
3.1.3.1.3 CALL OP	OP

The analog console is put into the Operate Mode.

Error Message:

ERROR-OP - BUSY: Occurs if console is busy.

3.1.3.2 Logic Modes

	Mode
3.1.3.2.1 CALL LOAD	LOAD

The analog logic is put into the Load mode.

Error Message:

ERROR-LOAD - BUSY: Occurs if console is busy.

3.1.3.2.2 CALL STP STOP

The analog logic is put into the Stop Mode.

Error Message:

ERROR-STP - BUSY: Occurs if console is busy.

3.1.3.2.3 CALL RUN RUN

The analog logic is put into the Run Mode.

Error Message:

ERROR-RUN - BUSY: Occurs if console is busy.

3.1.3.3 CALL VER (N)

<u>N</u>	<u>Mode</u>
0	Problem verify off
1	Problem verify on

Error Messages:

(a) ERROR-VER - DATA: Occurs when N is outside allowable range.

(b) ERROR-VER - BUSY: Occurs if console is busy.

3.1.3.4 CALL LEX (N)

<u>N</u>	<u>Mode</u>
0	Logic execute off
1	Logic execute on

Error Messages:

(a) ERROR-LEX - DATA: Occurs when N is outside allowable range.

(b) ERROR-LEX - BUSY: Occurs if console is busy.

3.1.3.5 CALL AUTHD (N)

<u>N</u>	<u>Mode</u>
0	Auto-hold off
1	Auto-hold on

Error Messages:

(a) ERROR-AUTHD - DATA: Occurs when N is outside allowable range.

(b) ERROR-AUTHD - BUSY: Occurs if console is busy.

3.1.3.6 CALL PB (N)

<u>N</u>	<u>Mode</u>
0	PB off
1	PB On

Error Messages:

(a) ERROR-PB - DATA: Occurs when N is outside allowable range.

(b) ERROR-PB - BUSY: Occurs if console is busy.

3.1.3.7 CALL TEST (N)

N	Mode
0	Test function off
1	Test function on

Error Messages:

(a) ERROR-TEST - DATA: Occurs when N is outside allowable range.

(b) ERROR-TEST - BUSY: Occurs if console is busy.

3.1.3.8 CALL STEP

If logic mode is STOP, the logic step function is implemented.

Error Message:

ERROR-STEP - BUSY: Occurs if console is busy.

3.1.3.9 CALL STREF (N)

N	Mode
0	Coefficient values read from coefficient devices
1	Coefficient X input values read from coefficient devices

Error Messages:

(a) ERROR-STREF - DATA: Occurs if N is not in range.

(b) ERROR-STREF - BUSY: Occurs if console is busy.

3.1.3.10 CALL TSCAL (N)

N	Mode
0	Time scale X1
1	Time scale X10
2	Time scale X100
3	Time scale X1000

Error Messages:

(a) ERROR-TSCAL - DATA: Occurs when N is outside allowable range.

(b) ERROR-TSCAL - BUSY: Occurs if console is busy.

3.1.3.11 CALL STITR (I, J, K)

The routine STITR allows the user to set the length of the periods in the interval timer register.

The arguments I, J and K correspond to the number and counts in the A, B and C periods in the interval timer register.

Error Messages:

(a) ERROR-STITR - DATA: Occurs when I, J, or K are outside allowable range.

(b) ERROR-STITR - BUSY: Occurs if console is busy.

3.2.3.12 CALL SELIT (N)

SELIT is called to select the interval timer periods from either the thumb wheel switches or the interval timer register.

<u>N</u> <u>MODE</u>

0 Interval Timer periods selected from thumb wheel switches.

1 Interval Timer periods selected from register.

Error Messages:

(a) ERROR-SELIT - DATA: Occurs when N is outside allowable range.

(b) ERROR-SELIT - BUSY: Occurs if console is busy.

3.1.3.13 CALL SELVS (N)

The routine SELVS is used to select the V-signals which are used to count the periods in the interval timer.

<u>N Mode</u>

0 Selects 1 sec V-pulse into Interval timer

1 Selects 100 m sec V-pulse into Interval timer.

2 Selects 10 m sec V-pulse into Interval timer.

Error Messages:

(a) ERROR-SELVS - DATA: Occurs when N is outside allowable range.

(b) ERROR-SELVS - BUSY: Occurs if console is busy.

3.1.3.14 CALL SETAR (IA)

SETAR is used to set an address into the AD/FIVE address register.

Argument IA is the integer address (see Table 2) to be set into the address register.

Error Messages:

(a) ERROR-SETAR - ADDRESS: Occurs when analog address is non-valid.

(b) ERROR-SETAR - BUSY: Occurs when console is busy.

3.1.3.15 CALL SETC (M)

The routine SETC allows the user to set individual conditions into the control register.

Argument M is the number corresponding to the desired control condition (see Table 3).

Error Messages:

(a) ERROR-SETC - DATA: Occurs if M is not in range.

(b) ERROR-SETC - BUSY: Occurs if console is busy.

TABLE 3: CONTROL CONDITION

M	FUNCTION	OPERATION
0	IC	Place AD/FIVE in Initial Condition mode
1	OP	Place AD/FIVE in Operate mode
2	H	Place AD/FIVE in Hold mode
3	LD	Place AD/FIVE in Load mode
4	RUN	Place AD/FIVE in Run mode
5	STOP	Place AD/FIVE in Stop mode
6	V1S	Place Interval Timer in 1 sec frame
7	V100M	Place Interval Timer in 100ms frame
8	V10M	Place Interval Timer in 10ms frame
9	X1	Place AD/FIVE in X1 time scale
10	X10	Place AD/FIVE in X10 time scale
11	X100	Place AD/FIVE in X100 time scale
12	X1000	Place AD/FIVE in X1000 time scale
13	SET COEF	Initiate a set coefficient operation
14	LST	Initiate a logic step
15	SP	Initiate a DRM sample operation to digitize an analog value
16	LE SET	Set Logic Exec state
17	LE CLR	Clear Logic Exec state
18	AH SET	Set Auto Hold state
19	AH CLR	Clear Auto Hold state
20	PB SET	Transfer PB hole to ASO
21	PB CLR	Remove PB hole from ASO
22	COEFXS	Set Coef X Input state
23	COEFXC	Clear Coef X Input state
24	PV SET	Set Problem Verify State
25	PV CLR	Clear Problem Verify State
26	TEST SET	Set Test state
27	TEST CLR	Clear Test state

TABLE 3: CONTROL CONDITION

M	FUNCTION	OPERATION
28	CLR	Clear the address or data register
29	SCLR	Clear the AD/FIVE system (initialize)
30	AE	Set the Address entry state
31	DE	Set the Data Entry state
32	ENA 15	
33	ENA 14	
34	ENA 13	
35	ENA 12	
36	ENA 11	
37	ENA 10	
38	ENA 9	
39	ENA 8	
40	ENA 7	
41	ENA 6	Hybrid On
42	ENA 5	Hybrid Off
43	ENA 4	Enable MSD of the DRM onto system read bus. Includes MSB, Sign, 1SP.
44	ENA 3	Enable NMSD (BCD) of the DRM onto system read bus.
45	ENA 2	Enable NMSD (BCD) of the DRM onto system read bus.
46	ENA 1	Enable NMSD (BCD) of the DRM onto system read bus
47	ENA 0	Enable LSD (BCD) of the DRM onto system read bus

3.2 Fortran Functions

3.2.1 IOVLD (D)

Tests the overload bit of the status register and returns an arithmetic 0 if off and an arithmetic + 1 if on. D is a dummy argument.

Error Message:

ERROR - IOVLD - BUSY: Occurs when console is busy.

3.2.2 IBUSY (D)

Tests the BUSY bit of the status register and returns an arithmetic 0 if off and an arithmetic + 1 if on. D is a dummy argument.

4.0 HYBRID EXPANSION ROUTINES

4.1 Fortran Callable Routines

4.1.1 Discrete Data

4.1.1.1 CALL SETCL (M, N)

The routine SETCL is used to set or clear one of the sixteen lines in the control register for the AD/FIVE. All other lines are unchanged.

Argument M defines the line number in the control register (0-15).

Argument N is Ø if the line is to be off, and is 1 if the line is to be on.

4.1.1.2 CALL SETSL (M, N)

The routine SETSL is used to set or clear one of the sixteen bits in the sense line register for the AD/FIVE. All other lines are unchanged. This routine is applicable only to the D16.201 Sense Register.

Arguments M and N are the same as in SETCL, except the sense register is referenced.

4.1.1.3 CALL SETCR (M)

SETCR sets a 16 bit image into the control register. This allows more than one line to be set simultaneously.

Argument M is the sixteen bit image to be set into the control register.

4.1.1.4 CALL SENCR (M)

SENCR allows the user to read the sixteen bit image out of the control register. The register is not cleared at the end of the read. The sixteen bit image is placed in the argument M.

4.1.1.5 CALL SETSR (M)

SETSR sets a sixteen bit image into the sense register. This allows more than one bit to be set simultaneously. This routine is applicable only to the D16.201 Sense Register.

Argument M is the sixteen bit image to be set into the sense register.

4.1.1.6 CALL SENSR (M)

SENSR allows the user to read the sixteen bit image in the sense register. The register is cleared at the end of the read.

The sixteen bit image is placed in the argument M.

4.1.1.7 CALL INTRW (M)

The routine INTRW is used to read the 8 bit image in the interrupt register. The register is cleared at the end of the read. The resultant 8 bit image is placed in the 8 low order bits of the variable M. The high order bits are set to zero.

4.1.1.8 CALL COMSY (K, L, M, N)

The routine COMSY is provided to allow the use of the special functions within the AD/FIVE Interface and Communications System. The user may build his own routines not provided in the HCR's.

Argument K is the function and modifier for I/O commands to the interface.

Argument L is the Communications System address of the module to receive or send data.

Argument M is the 8 bit function code to the Communications System.

Argument N is the variable from which data is sent or into which data is received.

4.1.2 MDAC and ADC Control Routines

4.1.2.1 CALL ADCSH (N, M)

The routine ADCSH facilitates control of the ADC Sample-Hold package, if present in the ADC.

Argument N is the channel number of any of the eight channels on a Sample-Hold card.

Argument M is a zero to set all channels on card to sample and is a one to set all channels on card to hold.

Note: Addressing any of the channels on a card sets the whole card into sample or hold.

4.1.2.2 CALL UPDAT (N)

The routine UPDAT allows a chassis of DAC's to be updated simultaneously. This routine is used in connection with STINH with a load data only option (See 4.1.3)

Argument N is the chassis number of the DAC to be updated.

4.1.2.3 CALL DACU (M,N)

The DACU routine permits the user to set up update groups in the DAC Controller.

Chassis number is denoted by argument M.

Argument N is a 16-bit word pattern whose bit positions correspond one-for-one with the 16 DAC's in one chassis.

In the argument N, a 1-bit in a particular position enables that DAC to be updated by calling CONUP. A Ø-bit disables that DAC from updating.

4.1.2.4 CALL CONUP (M)

The CONUP routine, used in conjunction with DACU, allows updating of groups of DAC's through the DAC Controller.

Argument M is the chassis number of the DAC's to be updated.

When CONUP is called, all DAC's in chassis M whose update lines have been enabled by a previous call to DACU, are simultaneously updated.

4.1.3 MDAC and ADC Data Routines

4.1.3.1 CALL SCANH (N, IV, M)

The SCANH routine is used to read ADC-MUX channels in the sequential mode.

Argument N is the number of the channel at which the scan is to begin.

Argument IV is the array into which the data read out of the ADC is placed.

Argument M is the number of ADC-MUX channels to be scanned.

4.1.3.2 CALL READH (N, IV)

The READH routine permits the user to read a single ADC-MUX channel.

Argument N is the number of the channel to be read.

Argument IV is the variable in which the data read from the ADC is placed.

4.1.3.3 CALL STINH (N, M, L)

The routine STINH is used to set an individual MDAC to a specified value. There is also control over immediate or delayed updating of the MDAC.

Argument N is the number of the MDAC to be set.

Argument M is the value to which the MDAC is to be set.

Argument L gives control over updating and external control.

L	Result
0	load data into initial register only
1	load data and update immediate
2	load data and enable external control
3	load data, update immediate, and enable external control.

Note: External control pertains to the updating capabilities through the DAC Controller.

4.1.3.4 CALL STBLK (N, M, L, K)

The STBLK routine allows one or more MDAC's to be set to specified values in the non-sequential mode. As in STINH, control over updating is also given.

Argument N is an array containing the numbers of the MDAC's to be set.

Argument M is an array containing the values to be set into the MDAC's.

Argument L is an array containing the control over updating for each MDAC as in STINH.

Argument K is the number of MDAC's to be set.

4.1.4 Routines using Cycle Stealing

4.1.4.1 CALL STSCN (N, IV, M)

The routine STSCN is identical to SCANH except that the values are read out under cycle stealing.

Note: Before using data read back, completion of cycle stealing should be tested using the function ICMPL.

4.1.4.2 CALL STSEQ (N, M, L, K)

The routine STSEQ facilitates the setting of sequential MDAC's using cycle stealing. The routine uses the DAC Controller.

Arguments N, M, L, and K are the same as in STBLK.

4.2 Fortran Functions

4.2.1 ICMPL (D) D is a dummy argument.
Tests the completion of the routines using cycle stealing. Returns 0 if not complete, 1 if complete and -1 if an error occurred.

4.2.2 ITSTC (M)
Tests line M in the control register. Returns 0 if M is off and 1 if M is on.

4.2.3 ITSCM (M)
Tests the control register under the mask M. Returns 0 if all bits tested are off and 1 if any one is on.

4.2.4 ITSTS (M)
Tests line M in the sense register. Returns same as ITSTC.

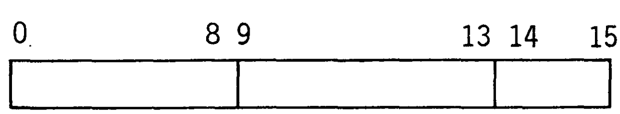
4.2.5 ITSSM (M)
Tests the sense register under the mask M. Returns same as ITSCM.

4.2.6 INTR (M)
Tests line M in the interrupt register. Returns 0 if M is off and 1 if M is on.

4.2.7 INTRM (M)
Tests the interrupt register under the mask M. Returns same as ITSCM.

ERROR WORD FORMATS

For the AD/FIVE/IBM-1130 system and the AD/FIVE/PDP-11 the error word format is:

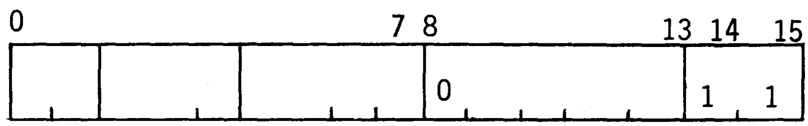


Bits 14 and 15 contain the error type:

Number	Type of Error
0	Address
1	Data
2	Busy
3	Cannot Set Pot

Bits 9-13 contain the HCR number.

In the special case of "Cannot Set Pot" error, the HCR number is either 7 for STIND or 8 for STINA, bits 0 through 7 are free to be used for the pot address. The format is as follows:



Since the error type is 3, we know the address is a pot, and bits 0-7 contain the field, area, and number in decimal format of the pot where the error occurred.

GENERALIZED AD/FIVE HCRS

1. CONTROL TO/THROUGH COMMUNICATIONS SYSTEM

   A. CALL CCSYS(M)

0 3	4 7	8 11	12 15
COMMAND		ADDRESS	

ADDRESSES:		
	00	- DIGITAL INTERFACE
	01	- AD/FIVE CONTROL
	02	- INTERRUPT & STATUS
	03	- ADC SYSTEM
	04	- CONTROL LINE REGISTER
	05	- SENSE LINE REGISTER
	06	
	07	
	08	
	09	
	0A	
	0B	
	0C	
	0D	
	0E	
	0F	
	10 | 1F	- DAC SYSTEM

B. CALL SETIC(M)

Identical to 1.A. with implied address 01.

2. DATA TO/THROUGH COMMUNICATIONS SYSTEM

CALL CDSYS(D,N)

0 3	4 7	8 11	12 15

DATA
(N)

D = Direction:
0 - Write
1 - Read

3. CONTROL AND DATA TO/THROUGH COMMUNICATIONS SYSTEM

CALL CMSYS(M,D,N)

Equivalent to 1.A. followed by 2.

CALL SETIC(M)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1							0	0	0	0	0	0	0	1
CONTROL								ADDRESS							

M	I/O WORD	FUNCTION	
0	4001	IC	Place AD-5 in Initial Condition mode
1	4101	OP	Place AD-5 in Operate mode
2	4201	H	Place AD-5 in Hold mode
3	4301	LD	Place AD-5 in Load mode
4	4401	RUN	Place AD-5 in Run mode
5	4501	STOP	Place AD-5 in Stop mode
6	4601	V1S	Place interval timer in 1 sec frame
7	4701	V100M	Place interval timer in 100 ms frame
8	4801	V10M	Place interval timer in 10 ms frame
9	4901	X1	Place AD-5 in X1 time scale
10	4A01	X10	Place AD-5 in X10 time scale
11	4B01	X100	Place AD-5 in X100 time scale
12	4C01	X1000	Place AD-5 in X1000 time scale

M	I/O WORD	FUNCTION	
13	4D01	SET COEF	Initiate a set coefficient operation
14	4E01	LST	Initiate a logic step
15	4F01	SP	Initiate a DRM sample operation to digitalize an analog value
16	5001	LE SET	Set Logic Exec state
17	5101	LE CLR	Clear Logic Exec state
18	5201	AH SET	Set Auto Hold state
19	5301	AH CLR	Clear Auto Hold state
20	5401	PB SET	Transfer PB hole to ASO
21	5501	PB CLR	Remove PB hole from ASO
22	5601	COEFXS	Set Coef X Input state
23	5701	COEFXC	Clear Coef X Input state
24	5801	PV SET	Set Problem Verify state
25	5901	PV CLR	Clear Problem Verify state
26	5A01	TEST SET	Set Test state
27	5B01	TEST CLR	Clear Test state
28	5C01	CLR	Clear the address or data register
29	5D01	SCLR	Clear the AD-5 system (initialize)

M	I/O WORD	FUNCTION	
30	5E01		Set the Address entry state
31	5F01		Set the Data entry state
32	6001	ENA Ø	Enable LSD(BCD) of the DRM onto system read bus
33	6101	ENA 1	Enable NMSD(BCD) of the DRM onto system read bus
34	6201	ENA 2	Enable NMSD(BCD) of the DRM onto system read bus
35	6301	ENA 3	Enable NMSD(BCD) of the DRM onto system read bus
36	6401	ENA 4	Enable MSD of the DRM onto system read bus. Includes MSB, Sign, Busy, 1SP
37	6501	ENA 5	
38	6601	ENA 6	
39	6701	ENA 7	
40	6801	ENA 8	
41	6901	ENA 9	
42	6A01	ENA 10	
43	6B01	ENA 11	
44	6C01	ENA 12	

M	I/O WORD	FUNCTION	
45	6D01	ENA 13	
46	6E01	ENA 14	
47	6F01	ENA 15	

CONTROL REGISTER

1. CONTROL ONLY

CALL CNRG1(M)

0 1	2	3	4 5	6 7	8 11	12 15
A	B	C	D	E		
CONTROL (M)					ADDRESS	

ADDRESS: 04

CONTROL:

A	00	-	(N/A)
	01	-	Clear sync
	10	-	Set sync
	11	-	(N/A)
B	0	-	(N/A)
	1	-	Clear register
C	0	-	(N/A)
	1	-	Reed register
D	00	-	(N/A)
	01	-	Clear on 1's
	10	-	Set on 1's
	11	-	Load
E	00	-	(N/A)
	01	-	Off
	10	-	On
	11	-	On then off

2. CONTROL AND DATA

CALL CNRG2(M,N)

Equivalent to 1. followed by data N.

SENSE REGISTER

1. CONTROL ONLY

CALL SNRG1(M)

0	1	2	3	4 5	6 7	8 11	12 15
A	B	C	D	E	F		
$\underbrace{\qquad\qquad}_{\text{CONTROL (M)}}$						$\underbrace{\qquad\qquad}_{\text{ADDRESS}}$	

ADDRESS: 05

CONTROL:

A		(N/A)
B	0 -	(N/A)
	1 -	Set Test
C	0 -	(N/A)
	1 -	Clear register
D	0 -	(N/A)
	1 -	Read
E	00 -	(N/A)
	01 -	Off
	10 -	Track
	11 -	Store
F	00 -	(N/A)
	01 -	Disable interrupt
	10 -	Enable interrupt
	11 -	(N/A)

2. CONTROL AND DATA

```
CALL SNRG2(M,N)
```

Equivalent to 1. followed by data N.

INTERRUPT, STATUS, AND DRM DATA

CALL STATS(N)

0 3	4 7	8 11	12 15
3	0	0	2
CONTROL		ADDRESS	

0 3	4 7	8 11	12 15
DATA (N)			

DATA WORD:

- 0 - 7: Interrupts 0 - 7
- 8: Overload status
- 9: Hybrid On status
- 10: (Not used)
- 11: (Not used)
- 12 - 15: DRM data (BCD) for 4 LSD's.

  For MSD:

  - 12: Sign bit
  - 13: DRM Busy status
  - 14: Coef Setting status (1 SECP)
  - 15: Most significant bit

ADDRESS AND DATA ENTRY

CALL ADDA(P,N)

0 3	4 7	8 11	12 15
P	N	ADDRESS	

ADDRESS: 01

P: 0 - Address entry

1 - Data entry

N: BCD digit to be entered

APPLIED DYNAMICS ANALOG/HYBRID SYSTEMS

AD/FIVE

LINKAGE DIAGNOSTIC

OPERATORS MANUAL

TABLE OF CONTENTS

AD/FIVE
LINKAGE DIAGNOSTIC
OPERATOR'S MANUAL

1-1 The Linkage Diagnostic is loaded into the digital computer either as a Fortran or assembly language deck or tape. LKD consists of a main line program, thirteen independent subroutines, and five support subroutines. The lower numbered routines test the hardware most basic to the operation of the system. In general, the higher numbered routines are more extensive and depend greatly on the error-free operation of the hardware tested previously.

1-2 The main line program initializes variables to be used by the subroutines, then types "ENTER CONSOLE NO.". The operator should enter the one-digit number of the console to be tested. The program then requests the operator to indicate whether or not he wishes to pause after each error, or type out all errors with no pause. The operator types 1 for error pause, or 0 for no pause. The program then types "ENTER ROUTINE NO.". The operator should enter the two-digit number of the desired routine. An invalid routine number, such as 00 or 14, causes the program to reinitialize variables so the program is effectively the same as it was at loading. Routine number 99 causes a return to Monitor or an exit. Numbers 01 through 13 cause the associated routine to be called.

1-3 The called routine types its number and name. When an error is encountered, an error message is typed, giving error type, address, register number, etc. If pause mode was selected, the operator types 1 to continue, 2 to return to the beginning of the current routine, or 3 to exit from the current routine. Routines requiring patching or other operator action also expect the above entries.

1-4 Bit patterns used in the tests are formed by moving a "one" through a field of "zeros" and a "zero" through a field of "ones". The field length is equal to the particular register being tested. The tolerance for value tests is initially set to 0.0010 of Reference. (see LKD13).

1-5 Following is a list of the thirteen diagnostic routines, the functions they test and explanations of their error messages.

01 LKD1 - STATUS BIT TEST

By causing appropriate error and non-error conditions, LKD1 tests all bits of the status bus.

Error Messages

ERROR - OVLD IS $\begin{matrix}0\\1\end{matrix}$

ERROR - DRM BUSY IS $\begin{matrix}0\\1\end{matrix}$

ERROR - COM BUSY IS $\begin{matrix}0\\1\end{matrix}$

ERROR - 1SP IS $\begin{matrix}0\\1\end{matrix}$

NOTE: This routine will pause regardless of pause action decided by the operator. (See paragraph 1-2)

02 LKD2 - AD/FIVE CONTROL REGISTER TEST

This routine requests "ENTER (FOUR DIGIT) TEST CYCLES". The operator should enter the number of times the test will be performed, with leading zero (s) if the number is less than 1000. The register is tested by writing a series of data words in, reading them back, and comparing the results with the data sent.

Error Messages

ERROR - AD/FIVE CR - SENT nnnnn ERR eeeee

where nnnnn is decimal representation of the bit pattern sent.

eeeee is the decimal representation of the bits in error.

03 LKD3 - AD/FIVE SENSE REGISTER TEST

This test operates in the same manner as LKD2, except the AD/FIVE sense register is tested with the inputs disabled. This routine is applicable only to the D16.201 Sense Register.

04 LKD4 - CONTROL, SENSE, INTERRUPT REGISTER TEST

The first time LKD4 is called, the total number of each register in the console and "PATCH ACCORDING TO MANUAL" are typed. Patching instructions are:

Control reg. lines 0-15 to sense reg. lines 0-15, control reg. lines ~~8-15~~ 0-7 to interrupt reg. lines 0-7.

The routine then requests "ENTER (FOUR DIGIT) TEST CYCLES". The operator should enter the number of times the test will be performed, with leading zero(s) if the number is less than 1000. The registers are tested by sending a series of data words to the control register, reading the sense and interrupt registers and comparing with the data sent.

Error Messages

ERROR - CLR OR SLR SENT nnnnn ERR eeee

ERROR - INT. REG. SENT nnnnn ERR eeee

where:

nnnnn is the decimal representation of the value sent

eeee is the decimal representation of the bits in error (+ = on, - = off).

05 LKD5 - DAC UPDATE REGISTER TEST

LKD5 uses the valid address table to determine the addresses of all valid DACs. If there are none, a message is typed and the routine is skipped. Each DAC is tested by writing each update code into the update register, reading it back, and comparing with the code sent.

Error Messages

ERROR - REGISTER ADDR. baaa SENT n RECVD n

where b is the component class

aaa is the address

n are the values sent and received.

06 LKD6 - ADC SAMPLE/HOLD REGISTER TEST

If there are no S/H registers, a message is typed and the routine is skipped. Each S/H register is tested by writing each S/H code into the register, reading it back, and comparing with the original.

Error Messages

ERROR - REGISTER ADDR. baaa SENT n RECVD n

where b is component class
aaa is the address
n are the values sent and received.

07 LKD7 - DAC TEST

LKD7 uses the valid address table to determine the addresses of all valid DACs. If there are no DACs a message is typed and the test is skipped. A table of valid DAC addresses and "PATCH ACCORDING TO MANUAL" are typed. Patching instructions are:

Patch all multiplying DACs in the valid address table to plus and minus reference.
Patch all DAC outputs to ADC MUX inputs.
Each DAC is tested by writing a series of values into it, reading them back, and comparing the results with the values sent.

Error Messages

ERROR - ADDR. baaa SENT nnnn ERR eeee

where b is the component class
aaa is the DAC address
nnnn is the values sent and received
±eeee is the amount of the error

08 LKD8 - DCU TEST

LKD8 uses the valid address table to determine the addresses of all valid DCUs. If there are no DCUs a message is typed and the test is skipped.

Error Messages

ERROR - ADDR. baaa SENT nnnn ERR eeee

where b is the component class
aaa is the DCU address
nnnn is the values sent and received
±eeee is the amount of the error

09 LKD9 - SERVO SET POT TEST

LKD9 uses the valid address table to determine valid pot addresses. If there are no pots, a message is typed and the routine is skipped. Otherwise, a table of valid pot addresses and "ENTER (4I4) FOUR POT ADDRESSES" are typed. The operator should enter four pot addresses from the table, with no embedded blanks or other characters. These four pots are tested by sending values to each, reading them back and comparing with the values sent.

Error Messages

ERROR - ADDR baaa SENT nnnn RECVD nnnn

where b is the component class

aaa is the pot address

nnnn are the values sent and received.

10 LKD10 - DRM

LKD10 requests the operator to patch 5 volts into PB and then reads it back through the DRM. The reading is compared within tolerance to 5 volts. LKD10 then requests the operator to patch 1 volt into PB and then reads it back through the DRM. The reading is compared within tolerance to 1 volt.

Error Messages

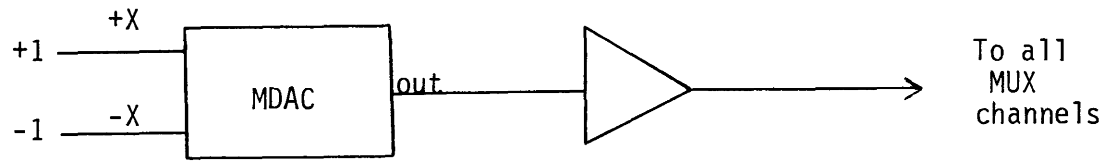
ERROR - DRM EXPECTED nnnnn RECVD nnnnn

where nnnnn are the values sent and received.

11 LKD11 - ADC SYSTEM TEST

LKD11 uses the valid address table to determine a valid DAC, then types "PATCH ACCORDING TO MANUAL", and the DAC address. Patching should be according to the following diagram.

Each ADC channel is tested by sending a series of values to the DAC, reading the ADC channels and comparing values.



Error Messages

ERROR - ADC baaa SENT nnnnn ERR eeee

where b is component type

aaa is the address

nnnnn is the value sent

±eeee is the amount of the error

12 LKD12 - CONSOLE CONTROL

LKD12 tests the action of each pushbutton on the AD/FIVE console for appropriate control from the digital computer. Each button is tested in sequence for on and off state with verification by the operator. When proper operation is confirmed by the operator (typically visually verifying the pushbutton lamps), LKD12 proceeds to the next test upon receipt of an interrupt or the depression of the carriage return key on the typewriter. A pause is automatically built into LKD12 after each test regardless of pause action selected by the operator. Appropriate messages are typed out for the operator identifying the particular condition for verification.

13 LKD13 - SET TOLERANCE

LKD13 types "ENTER (I2) NEW TOLERANCE FOR ANALOG VALUE TESTS", reads the value and types "NEW TOLERANCE IS PLUS OR MINUS nn". The operator should enter desired tolerance expressed as a fraction of reference. Two leading zeros are always assumed. Thus, for a tolerance of .0055 of reference only the 55 must be typed.

APPLIED DYNAMICS ANALOG/HYBRID SYSTEMS

DIVISION OF RELIANCE ELECTRIC COMPANY

BOX 1488, ANN ARBOR, MICHIGAN 48106 / PHONE 313-429-4957 / TWX 810-223-8151 / CABLE COMPUTERS