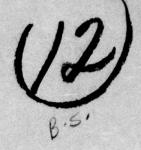
ARMY MISSILE RESEARCH DEVELOPMENT AND ENGINEERING LAB--ETC MULTIPURPOSE DIGITAL MICROPROCESSOR EMULATOR.(U) AD-A031 782 F/G 9/2 MAY 76 J R BROOKSHIRE RG-76-62 UNCLASSIFIED NL 1 OF 2 ADA031782



TECHNICAL REPORT RG-76-62

MULTIPURPOSE DIGITAL MICROPROCESSOR EMULATOR

Jerry R. Brookshire
Guidance and Control Directorate
US Army Missile Research, Development, and Engineering Laboratory
Redstone Arsenal, Alabama 35809

11 May 1976

Approved for public release; distribution unlimited.



U.S. ARMY MISSILE COMMAND

Redstone Arsenal, Alabama 35809

COPY AVAILABLE TO DDG DGES NOT PERMIT FULLY LEGIBLE PRODUCTION

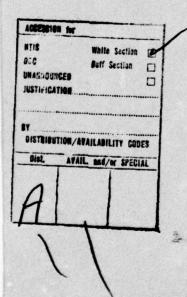


DISPOSITION INSTRUCTIONS

DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

DISCLAIMER

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.



TRADE NAMES

USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES NOT CONSTITUTE AN OFFICIAL INDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.

I. REPORT NUMBER 2. GOVT ACCESSIO	READ INSTRUCTIONS BEFORE COMPLETING FORM			
	N NO. 3. ECIPIENT'S CATALOG NUMBER			
RG-76-62	(4)			
4. TITLE (and Subtitio)	The or REPORT DE SOVE			
MULTIPURPOSE DIGITAL MICROPROCESSOR EMULATOR	Technical report,			
COLITION OF DIGITAL MICROPROCESSOR EMPLATOR,	6. PERFORMING ORG. RESONT HUMBE			
7. AUTHOR(a)	S. CONTRACT OR GRANT NUMBER(s)			
Jerry R./Brookshire				
PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TA			
Commander US Army Missile Command				
Attn: DRSMI-RG Redstone Arsenal, Alabama 35809				
1. CONTROLLING OFFICE NAME AND ADDRESS	18 REPORT DATE			
Commander US Army Missile Command	11 May 176			
Attn: DRSMI-RPR	117			
Redstone Arsenal, Alabama 35809				
14. MONITORING AGENCY NAME & ADDRESS(II different from Controlling Of	fice) 15. SECURITY CLASS. (of this report)			
14. MONITORING AGENCY NAME & ADDRESS(II different from Controlling Of				
14. MONITORING AGENCY NAME & ADDRESS/II different from Controlling Of	Unclassified			
Approved for public release; distribution unlim	Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE			
G. DISTRIBUTION STATEMENT (of this Report)	Unclassified 15a. DECLASSIFICATION/DOWNGRADII SCHEDULE ited.			
Approved for public release; distribution unlim	Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE ited.			
Approved for public release; distribution unlim	Unclassified 15a. DECLASSIFICATION/DOWNGRADII SCHEDULE ited.			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Black 20, if differ	Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE ited.			
Approved for public release; distribution unlim	Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE ited.			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Black 20, if differ	Unclassified 15a. DECLASSIFICATION/DOWNGRADII SCHEDULE ited.			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Black 20, if differ	Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE ited.			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different actions are also as a supplementary notes	Unclassified Isa. DECLASSIFICATION/DOWNGRADING SCHEDULE ited. ent from Report)			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abetract entered in Block 20, if different to the abetract entered in Block 20, if different to the supplementary notes 19. KEY WORDS (Continue on reverse side if necessary and identify by block in Emulator	Unclassified Isa. DECLASSIFICATION/DOWNGRADII SCHEDULE ited. ent from Report)			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different to the abstract entered in Block 20, if different to the supplementary notes 19. KEY WORDS (Continue on reverse side if necessary and identify by block in Emulator Digital microprocessor	Unclassified Isa. DECLASSIFICATION/DOWNGRADING SCHEDULE ited. ent from Report)			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abetract entered in Block 20, if different to the abetract entered in Block 20, if different to the supplementary notes 19. KEY WORDS (Continue on reverse side if necessary and identify by block in Emulator	Unclassified Isa. DECLASSIFICATION/DOWNGRADING SCHEDULE ited. ent from Report)			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different to the abstract entered in Block 20, if different to the supplementary notes 19. KEY WORDS (Continue on reverse side if necessary and identify by block in Emulator Digital microprocessor	Unclassified Isa. DECLASSIFICATION/DOWNGRADING SCHEDULE ited. ent from Report)			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if differ 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block in Emulator Digital microprocessor Guidance control computer	Unclassified Isa DECLASSIFICATION/DOWNGRADING SCHEOULE ited. ent from Report)			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different in the supplementary notes 18. Supplementary notes 19. Key words (Continue on reverse side if necessary and identity by block in Emulator Digital microprocessor Guidance control computer 10. Approach (Continue on reverse side if necessary and identity by block in the control computer	Unclassified Isa. DECLASSIFICATION/DOWNGRADII ited. ent from Report) umber)			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different in the state of the abstract entered in Block 20, if different in the state of the sta	Unclassified Isa DECLASSIFICATION/DOWNGRADII ited. ent from Report) umber) and implemented to support to purpose digital microprocessor			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if differ 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block in Emulator Digital microprocessor Guidance control computer 10. ABSTRACT (Continue on reverse side if necessary and identify by block in This report describes an emulator designed initial software development effort for a multipand which also is to provide a vehicle for expense.	Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEOULE ited. umber) and implemented to support the purpose digital microprocessor rimenting with hardware and so			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if differ 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block in Emulator Digital microprocessor Guidance control computer 10. Astract (Continue on reverse side if necessary and identify by block in a control computer 10. Astract (Continue on reverse side if necessary and identify by block in a control computer 10. Astract (Continue on reverse side if necessary and identify by block in a control computer 10. Astract (Continue on reverse side if necessary and identify by block in a control computer 10. Astract (Continue on reverse side if necessary and identify by block in a control computer describes an emulator designed initial software development effort for a multiplication of the control computer architectural changes proposed for the mice	Unclassified Tis. DECLASSIFICATION/DOWNGRADING SCHEOULE ited. and from Report) and implemented to support the purpose digital microprocessor rimenting with hardware and scroprocessor. The initial test			
Approved for public release; distribution unlim 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if differ 18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block in Emulator Digital microprocessor Guidance control computer 10. ABSTRACT (Continue on reverse side if necessary and identify by block in This report describes an emulator designed initial software development effort for a multipand which also is to provide a vehicle for expense.	Unclassified Tis. DECLASSIFICATION/DOWNGRADING SCHEOULE ited. and from Report) and implemented to support the purpose digital microprocessor rimenting with hardware and scroprocessor. The initial test the guidance control computer			

DD 1 1473 1473 EDITION OF 1 NOV 65 15 OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Extended)

ur sen i incomplete de organismo de la	
i de la companya de	
	The state of the s
	the second secon
	ms bitalena militarena war sar filia ili.
	to Vinghard milesten 19, 25, 15, 265 (American temperature)
	an Plaghero misseran Propinsi Lucio I formativa Mantalah na mase 1965 No. Kasawalansa formativa menga
	ma Pinghero nilasendo y 280 illos i formas y 200 il

CONTENTS

																							Page
1.	Inti	roduci	tion ar	d Bac	kgroun	d.																	3
2.			cessor																				
3.			Contro																				
4.			Routir																				
5.			ary Res																				
6.			on and																				
BIBL	OGRA	PHY.						•	•		•		•	•			•			•			25
Apper	ndix	A.	MULTI	THE PERSON NAMED IN			Section 2		336		1777		-	10000	-	1	100	-					
			ELEMEN	T DEF	INITIO	NS	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	27
Apper	ndix	в.	MULTIE					2010000	211227	Same of the last	1000000	-				1							
			FLOWCH	LARTS.		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	31
Apper	ndix	c.	MULTIE	URPOS	E MICR	OPE	200	CES	SS	OR	E	IUI	A	COF									56

1. Introduction and Background

This report describes an emulator designed and implemented to support the initial software development effort for a multipurpose digital microprocessor, and which also is to provide a vehicle for experimenting with hardware and software architectural changes proposed for the microprocessor. The initial test-bed application for the microprocessor is to be the guidance control computer in the T-6 missile. The flight control software will be developed utilizing this emulator.

The emulator is written in a variation of FORTRAN allowing the use of "structured" statements, including: IF...THEN...ELSE...; DO UNTIL (...)...; DO WHILE (...)...; and IF...THEN...OR IF...THEN...ELSE...

This program is first run through a preprocessor which translates the statements into standard FORTRAN, providing the capability to run the emulator on any host computer which has a standard FORTRAN compiler available. The utilization of the emulator for the T-6 missile flight control software development is to be on the Missile Computer Software and Hardware Center's Raytheon R-520 and the MICOM CDC 6600 computers.

The design objectives of the emulator were to support the immediate software development effort for the T-6 missile application of the microprocessor, while retaining as much flexibility for proposed-change evaluation as possible. To support the software development effort, it was essential that the emulator perform exactly as though it were the microprocessor hardware executing microinstructions. Both considerations. fidelity of microinstruction execution and ease of change, supported the position that the emulator should be highly modular. Additional considerations which influenced the emulator design include: ease of setup and operation; varying levels of detailed output available optionally to support different types of runs; and maximum portability to allow the emulator to run on other host computers. A single run control card was designed to define the desired characteristics of each run, and provides the mechanisms for specifying simulated memory preset values, output options, and run termination conditions (other than errors). For portability, the host-computer-dependent portions of the program were minimized and isolated for simplified conversion to other host computers.

The remainder of this report describes the specific requirements and the specific programming techniques used to meet the requirements, plus a description of some potential extensions to or generalizations from the current design. Since early applications software design and experimentation are often crucial during computer hardware definition and design, it is believed that the concepts provided here can provide an extremely useful tool in the continuing application of microprocessors to weapons control systems.

2. Microprocessor Hardware Organization

From the viewpoint of software, the microprocessor* consists of a 16-bit (reconfigurable up or down in 4-bit increments) arithmetic logic unit (ALU) with a 16-word random access memory (scratchpad), a 32-operation instruction set, expandable to 58 operations by interpretation of a carry-control bit, and further expandable by a 3-bit instruction modifier (Figures 1 and 2). There are two associated memories: 1024 - 16-bit words of main memory; and 1024 - 48-bit words of control memory. Normal execution of a macroinstruction begins with the decoding of the main memory word pointed to by the first word in the scratchpad (software program counter - PC) into two 8-bit addresses. The leftmost eight bits are placed in the control memory address register (H-register), and the rightmost eight bits are placed in the main memory address register (C-register). Both the H- and C-registers are 10 bits in length: The C-register is wired such that this decoding operation causes the two high-order bits to be on, so that the references to main memory following the decode will be offset by 1400 octal. The top two bits of the H-register can be controlled from the execution of a microinstruction.

The 48-bit control memory words (microinstructions) are broken into 25 distinct fields (Table 1). Six of the fields are encoded, and the remaining 19 are considered as discrete control bits. Eighteen of these control bits are now functionally assigned, but additional control functions could be implemented if required by grouping some of the functions under additional encoded fields; i.e., where three bits are now three functions, three bits encoded could be used to control eight functions. Current encoded fields consist of: the 5-bit operation code, a 3-bit modifier code, two 4-bit operands (scratchpad addresses), a 10-bit next control word (H-register) field, and a 3-bit analog/digital (A/D) multiplexer address field.

Macroinstruction execution proceeds by decoding the control memory word pointed to by the H-register after a main-memory decode. The design of the hardware is such that some of the control functions specified by discrete bits are performed first. Then the operation code is executed, followed by the disposition of the result as directed by the instruction modifer, and the functions of the remaining discrete control bits are performed. One of the control bits is used to enable the loading of the H-register from the 10-bit next-H field. Thus, the microinstructions comprising a macrooperation are linked. The last microinstruction of a macrosequence will contain the address of a microinstruction which initiates the next decode of a main memory word through the PC, and then increments the PC by one. Therefore, the main memory control program will be sequential, unless modified by a reload of the PC.

^{*}Monolithic memories 5701/6701 microprocessor.

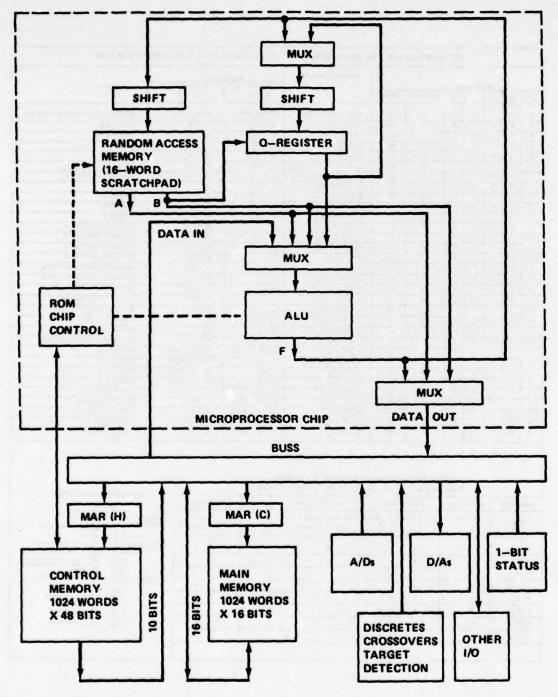


Figure 1. Multi-purpose digital microprocessor simplified block diagram.

			ROM WORD			ALU IN	STRUCTION		O CARRY	RY IN TYPICAL				
',	L	•	15	14	13	DECIMAL	OCTAL				(CM - L)		(CN - H)	USES
			L			0	00	LLLL++	инн + C _N	F	ORCE 111	,	FORCE 000	INITIALIZATION (FORCE 1's OR 8's
			L	L	н	1	01	AND A	В	A	, 8,		A _i B _j	AND A, & B,
		-	L	н	L	2	05	AND D,	18,	D	,		D, 8,	AND D. & B.
	1	-	L	+	H	3	03	OR A, &		A			A ₁ B ₃	OR A, & B,
	-	-	н	L	L	4	04	OR D, &	_	D			D, 8,	OR D ₁ & B ₃
_	_	-	н	L	н	5	05		VE OR A				A, B,	EXCLUSIVE OR A, & B,
_	_	-	н	H	H	6	06		VE OR D, &			_	D _i B _j	EXCLUSIVE OR D. & B.
-	1	H	-	-	-		10	A, + HHE			+1111	-	<u>^1</u>	INVERT D.
-		-	÷	-	H	•	11	B, + HH			. + 1111	-	D ₁	INVERT B.
-	_	-	i	H	T.	10	12	Q + HHH	H+C		+ 1111	-	8,	INVERT Q
-	٠	H	ī	H	H	11	13	A, + LLL		-		-	A, + 0001	2's COMPLEMENT OF A.
-	+	н	н	L	L	12	14	D, + LLL		10		-	D, + 0001	2's COMPLEMENT OF D.
-	+	H	н	L	н	13	15	B,+LLL		8		-	B, + 0001	Z's COMPLEMENT OF B.
-	1	H	н	н	L	14	16	0+1111		0		-	Q+0001	2's COMPLEMENT OF Q
	1	#	н	H	н	15	17	A, + LLL		A			A, + 0001	TRANSFER OR INCREMENT A,
•	1	-	L	L	L	16	20	D + LLL		0			D, +0001	TRANSFER OR INCREMENT D
•	1		L	L	н	17	21	B,+LLL	L + CN	8		_	B ₃ + 0001	TRANSFER OR INCREMENT B,
•	٢		L	н	L	18	22	Q+LLLI	+ CN	0			Q + 0001	TRANSFER OR INCREMENT Q
•	T	1	L	н	L	19	23	A, + HHF		A	+ 1111		A,	DECREMENT OR TRANSFER A
•	t	H	L	1	L	20	24	D, + HH			+ 1111		D ₁	DECREMENT OR TRANSFER DK
•	T	1	н		H	21	25	B, + HHH	H + CN	8	+ 1111		8,	DECREMENT OR TRANSFER B
1			н	H	L	22	26	Q + HHH	H+CN	and the state of the state of	+ 1111		a	DECREMENT OR TRANSFER Q
•			н	н	н	23	27	A,+B,+	CN	A	1 + B J		A, + B, + 00	
•		H	r	L	L	24	30	D, + B, +		D	1 + BJ		D + B + 00	
•		H	7	L	H	25	31	A, +Q+			1 + Q		A, +Q+000	
-		H	-	H	L	26	32	0,+0+			1+0		D1 + Q + 000	
1		H	H -	H	H	27	33	A _I +B _J +			1 - BJ - 0		A _I - B _J	SUBTRACT A, & B,
-	1_	H	H	-	H	29	35	B,+A,+			J - A1 - 0		B _J - A _I D _I - B _J	SUBTRACT D.& B.
÷	_	н	H	H	-	30	36	B,+D,+			B 0		8, - D,	SUBTRACT B, & D,
-	۰	H 1	H	H	H	31	37	D, + Q +			-0-00		D ₁ - Q	SUBTRACT D, & Q
4:	FI	RST	OPER		INST	RUCTION	MODIFI	ERS IN T	HE 8 x 1	B ROM	_			PIN CONFIGURATION
06	4 W	ORC	,	ROM I	VORD	LOAD	CONTROL	SI	IFT CONTE	OL	DAT	A OUT CO	NTROL	PIN CONFIGURATION
_	١,		-	DECIN		LOAD RAI	AB, LOAD	Q SHIF		DON'T	A	B	ALU OUTPUT F	A SOURCE A0 2 30 004 ACCUMULATOR A3 3 30 003 00174 GUTPUTS
		-	+	•		×	+	1	- India	×	- I		×	A ₂ = 4
-	L	H	+	1		×			+	×	×			SLOSEN - 8 22 - CN - 4 IMPPLE CARRY OUTPU
_	н	L	+	2		×	_	1		×		×		10 10 10 10 10 10 10 10
_	+	H	+	3		×	1	×	-				×	METRUCTION 11 13 28 P - CARRY PHOPMONTES
(in)	L		+	•		×		 	×				X	17 18 30 013 DATA HIPUTS
		+	+	5		×	×	×		1			×	14 16 36 012 DATA HERUTS 81 17 30 011
		_	+	6		×	×	1	×	1		-	×	
_	H	L												1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Figure 2. Multipurpose microprocessor instruction set.

TABLE 1. MULTIPURPOSE MICROPROCESSOR INSTRUCTION FORMAT

Bit Positions				Fiel	d Le	ngth	Field Name/Control Function
Field	Decimal	Oct	al			1,110	fallen, mellanetelvenske et e
1	0-4	0-4			5		Instruction Code
2	5-7	5-7	TOTAL S		3		Instruction modifier
3	8-11	10-	13		4		Operand A (scratchpad address)
4	12-15	14-	17		4		Opernad B (scratchpad address)
5	16-25	20-	31		10		Next H address
25	45-57	55-	57		3		A/D multiplexor address
		Discr	ete	Cont	rol	Bits:	(Action if Bit = 1)
6	26	32			1		SR1 = 0
7	27	33			1		SR2 = SR1
8	28	34			1		H Reg = CM addr
9	29	35			1		Main (C) = Buss
10	30	36		1000	1		Buss = Main (C)
11	31	37	3 1 10		1		Buss = ALU data out
12	32	40	7.15	100	1		Fetch enable
13	33	41			1		Data in = Input discretes or micad (KAD)
14	34	42		-10	1		SR1 = 1 if 4 data out = All 0's
15	35	43		E 200	1		LSB of buss = SR1
16	36	44			1		LSB of H = SR2
17	37	45		1	1		Carry (CN) = 1
18	38	46	The		1		Telemetry = Buss
19	39	47			1		C = Buss
20	40	50			1		Select D/A 1 and clock
21	41	51			1		Select D/A 2 and clock
22	42	52	(also		1		Select D/A 3 and clock
23	43	53		N DIV	1		Clock A/D mux addr register
24	44	54			1		Clock A/D conversion start (per bi
	ol Memor Format:	у				13631	
Bit:	0	5	8	12	16	26	45
Field Numbe	a solution of the second	2	3	4	5	16	thru 24 25

3. Emulator Control Structure

The emulator, as now structured, consists of a main program and 25 subprograms which are used for initialization and run control; instruction and control-function simulation, diagnostic and statistical output, and the provision of machine-dependent (Raytheon 520) direct code support. It was also determined to be necessary to provide FORTRAN masking and shifting operations for bit access and manipulation. An exclusive OR function has also been added to the function library of the R-520. These direct code and FORTRAN extensions tend to reduce the portability of the system, but they were minimized and simplified so that the effort raquired to duplicate these capabilities on another host computer can be minimal.

All microprocessor memory, registers, and data paths are represented by FORTRAN COMMON variable names in order to minimize parameter passing in calling sequences. The organization of the R-520 computer is such that REAL FORTRAN variables are represented internally by 48 bits, and integers have 24-bit representation. Thus, a microprocessor control memory word is represented by a real variable name (CNTMEN), and all other microprocessor entities are represented by integer variables. The implicit variable-type identifiers of standard FORTRAN are used exclusively, while at the same time maintaining at least an abbreviation of the name of the elements, preceded where necessary by an integer-type letter (I through N). For example, the two 1-bit registers, SR1 and SR2, are identified in the emulation program as MSR1 and MSR2.

The main program reads the run control card (Table 2) and proceeds to step through all the run-initiation subprograms. These are described in detail later, but consist of initializing all simulated microprocessor memory, registers and data paths, and reading microinstructions and main memory sequence instructions and data values into the appropriate locations of simulated microprocessor memory (Table 3). Execution is then initiated by entering an outer loop (DO WHILE .NOT.FINIS(hed). AND. . NOT. ERROR) in which a main memory access and decode is accomplished. An inner (macro) loop is then entered which is controlled by the contents of the H-field of the microinstructions. The emulator simulates the execution of all functions specified by the microinstructions, and falls out of the inner loop when a next-H = zero is encountered. At this point, the macrocount is incremented and the run termination conditions are examined to determine if the run should go on. If the run is to stop normally, the logical variable FINIS is set to .TRUE.; if a fatal error has been encountered, the logical variable ERROR will have been set to .TRUE. Either condition will cause the simulated run to stop, and a final statistical output routine will be called.

TABLE 2. MULTIPURPOSE MICROPROCESSOR RUN CONTROL CARD

Card Columns	Octal Value	Interpretation
001411110	varue	The order of the o
1-4	1	Pre-clear control memory.
	2	Pre-set control memory, NO-OP, next H = here.
5-8	1	Pre-clear main memory
	2	Pre-set main memory, H = 0, C = here.
9-12	1	Minimum output during execution
	2	Extensive diagnostic output during execution.
13-16	0	No initialization output.
	1	Output simulated memory after load.
17-20	xxxx	Ten-bit initial value of PC. (i.e., starting address of main memory)
21-24	XXXX	Output control-interval time-octal (sec)
25-28	xxxx	Output control-microinstruction count.
29-32	xxxx	Output control-selected macroaddress.
33-36	Run	Termination criteria
Carrier with the	1	H-address stop
matte to kine base	2	C-address stop
	3	Simulated elapsed time
	4	Real elapsed time
	5	Macroinstruction count
37-40	xxxx	Octal termination value associated with above termination criteria.
41-44	1	Load memory cards are in contiguous octal representation
mannin gaph call	2	Load control memory by 25 discrete octal fields

Note: Run control card can be expanded to accommodate more variable considerations - currently the card is read with a (20 04) format.

TABLE 3. MULTIPURPOSE MICROPROCESSOR MEMORY LOAD CARD

N-RUN (11) = 1: Card	amazon langua esculos
<u>Columns</u>	Contents
1	<pre>Key: 1 = main memory load 2 = control memory load 9 = load completed</pre>
9-12	Address: 4 octal digits
15-20	Main memory contents: 6 octal digits
21-36	Control memory contents: 16 octal digits
N-RUN (11) = 2:	
1	<pre>Key: 1 = Main memory load 2 = Control memory load 9 = Load completed</pre>
3-6	Address: 4 octal digits
8-10	Left 8 bits of main memory (octal)
11-13	Right 8 bits of main memory (octal)
15-33	Control memory instruction, by fields (octal)

Note: Memory load card could be extended to register loads or other size memory word loads by adding keys to columns 1 through 8, as desired.

4. Emulator Routine Descriptions

This section describes in detail the actions performed by the individual subprograms of the microprocessor emulator. The sequence of the descriptions follows generally the order in which the routines are called, except that the more general purpose and widely used routines are described last. Flowcharts, listings, and detailed data element definitions are given on the appendices.

a. Subroutine INIT

This routine initializes all nonmemory elements of the simulated hardware to zero, except for scratchpad word one (PC), which is set to the value read from the run control card [NRUN(5)]. In addition, the emulator control variables FINIS and ERROR are set to .FALSE., and operation count and time accumulation variables are set to zero. This routine is called one time per emulator run.

b. Subroutine SETCM

This routine initializes all 1024 words of simulated control memory (CNTMEM), based on an initialization key read from the run control card (NRUN(1)). If the key is 1, all of memory is cleared to zero; if the key is 2, each word in memory is preset to a NO-OP in the op code field, and the H-field is set to point to its own address. With this setting, execution of a microprogram will hang in a one-word loop if it inadvertently gets outside the area where the microprogram under test is loaded. Any other value of the key on the run control card will cause an error message and will abort the run by setting ERROR to .TRUE. This routine is called once per emulator run.

c. Subroutine SETMN

This subroutine is the same as SETCM, except NRUN(2) is the applicable key, and presets main memory (MAINM) rather than control memory.

d. Subroutine LOADM

This subroutine reads data cards which contain the octal address and octal contents of all words of simulated control memory and main memory which are to be loaded for an emulation run. A key in column 1 is used to determine the type memory a given card applies to: key = 1 loads main memory, key = 2 loads control memory, and key = 9 signifies the load is completed; any other key causes an error message to be printed, and the flag ERROR is set to .TRUE. The load cards may contain the 16- or 48-bit contiguous octal value to be placed in memory, or may contain separate fields for each field of the macroinstructions and microinstruction, as preferred. This routine is called once per emulator run.

e. Subroutine DCODM

This subroutine is called to initiate the execution of a macroinstruction by accessing a word of main memory and decoding it. The main memory word accessed is the one whose address is currently in the Program Counter (PC - word one of the simulated scratchpad). Once the address is retrieved, the PC is incremented by one. Then the main memory word is broken into two 8-bit fields utilizing the subroutine GETMM. If an error flag is returned from GETMM, an error message is output and ERROR is set to .TRUE. Otherwise, the left-most 8 bits are incremented by 1400 octal and placed in the simulated H-register, and the righ-most 8 bits are incremented by 1400 octal and placed in the simulated C-register. The original values of the two halves are stored in MINSTR and MOPND respectively. This subroutine is called once for each Macro executed.

f. Subroutine GETMM

This subroutine is generalized to extract any given field from a 16-bit simulated word located in a 24-bit R-520 word. If therefore has five parameters in its calling sequence: (1) the address of the word to be extracted from, (2) the starting bit, decimal 0 through 15, of the field to be extracted, (3) the number of bits to be extracted, (4) the address that the field, right adjusted and zero-filled, is to be stored in, and (5) an error flag which is set to zero when no error is encountered, or to an integer to identify the parameter which contained a recognizable error. This subroutine is normally callled twice per macro, or once for each desired field extraction.

g. Subroutine DCODC

This subroutine is called to break down a 48-bit micro-instruction into its constituent fields (currently 25) to facilitate interpretation and simulated execution of the specified operations and control functions. The selected 48-bit word is that pointed to by the index in the simulated H-register. The extraction of the fields is accomplished by calling GETCM, and the fields are placed in a 25-word vector in COMMON identified as MICRO(25). If an error flag is returned from GETCM, an error message is output and ERROR is set to .TRUE. The subroutine is called once for each microinstruction to be executed.

h. Subroutine GETCM

This subroutine is generalized to extract any given field from a 48-bit (60-bit in CDC version) real word, and return the result right-justified, zero-filled, in a 24-bit integer word. It is therefore currently limited to a 24-bit extraction, but the largest field currently required from a microinstruction is the 10-bit next-H field. The calling parameters are identical to those listed for GETMM, except that the starting bit may be any decimal number from 0 through 47. This subroutine is normally called 25 times per microinstruction, or once for each desired field extraction.

i. Subroutine EXECM

This subroutine manages the complex execution of a microinstruction. It first calls PRBIN for the application of those control
functions which occur prior to instruction interpretation/execution. It
then determines if the register SR2 is to cause the operation code to be
modified, and does so if indicated. It then determines which of seven
functional areas of execution that the operation code falls in, and then
calls the appropriate routine to execute that operation. If the operation
code is outside the allowed range (0 - 31 decimal), then an emulator
error has occurred, a message is output, and ERROR is set to .TRUE.
Otherwise, the subroutine BINRY is called to interrupt and act upon the
binary control fields which apply in time afer instruction execution is
completed. This subroutine is called once per microinstruction.

j. Subroutine PRBIN

The microprocessor hardware design is such that the following control functions are performed, if enabled, prior to instruction execution:

Field (Ref. Figure 2)	Octal Bit No.	Action if Bit = 1 (On)
· · · · · · · · · · · · · · · · · · ·		
6	32	SR1 = 0 (MSR1 = 0)
10	36	Buss = Main(c) (MBUSS - MAINM(MEMC))
13	41	Buss = DATA IN = (Input Discretes) OR (A/D Output)
15	43	LSB or Buss = SR1 (MBUSS = OR (MBUSS, MSR1))
17	45	Carry (Cn) = 1
20	50	Select D/A 1
21	51	Select D/A 2
22	52	Select D/A 3
23	53	Clock A/D Mux Addr Register
24	54	Clock A/D conversion Start (per Bit.)

The subroutine checks each of these fields [micro(6) micro(10),... MICRO(23)] in turn: For each that is set (i.e., equal to 1), the indicated operation (or test and operation) is performed. Most of the rquired actions are the setting of one variable equal to another (i.e., MSR1 = 0), MBUSS = MAINM(MEMC), etc.), but not always. For example, if bit 43 octal = 1, MSR1 is ORed to the MBUSS, so that the only change that if MSR1 = 1 and the LSB of MBUSS is 0, it is changed to 1. The last four controls cause special keys (KDA for D/A, KAD for A/D) to be set at nonzero (1, 2, or 3) for the appropriate D/A register select, or equal to the MUX address [MICRO(25)] for A/D input. Timing for A/D setting delays must be managed by the microprogram. This subroutine is called once per microinstruction.

k. Subroutine BOLN (Boolean)

This is the first operation code execution subprogram, and is called when the effective operation code is in the range 00 to 06 inclusive. The operation specified are all Boolean except that generated by OP code 0. In this and all subsequent OP code execution routines, a temporary variable ITMP is first set to the value resulting from the operation, and the final effect on the status of registers and output lines is determined later, after analysis of the instruction modifier field. The status of the carry in register ICN affects the result of OP code 0. If ICN = 0, then ITMP = 177777, or all bits on. If ICN = 1, then ITMP = 0 (all bits off). For the other OP codes in this subroutine, ICN has no effect.

OP Code	<u>Result</u>
00	ITMP = 177777 if ICN = 0; = 0 if ICN = 1
01	ITMP = (A OPND) and (B OPND)
02	ITMP = (Data in) and (B OPND)
03	ITMP = (A OPND) or (B OPND) Inclusive
04	ITMP = (Data IN) or (B OPND) or
05	ITMP = (A OPND) XOR (B OPND) Exclusive
06	ITMP = (Data IN) XOR (B OPND) or

Note: The data IN lines are represented by the variable IDIN.

This subroutine then examines the instruction modifier field. If the modifier value is seven, then the Q-register (MICQ) is set equal to the resultant value of ITMP, and control is returned to the calling program. If the modifier is not seven, then the scratchpad address pointed to by the B-operand is set to ITMP, and the subprogram INMOD is called to complete the effects of the modifier.

1. Subroutine INVRT (Invert)

This subprogram is called when the effective operation code is in the range 07 to 10 inclusive (07 to 12 octal). In the emulator, the operations are performed first, and then the effects of the carry (ICN) are applied.

OP Code Decimal	Octal		<u>Result</u>		
7	07	ITMP =	Complement	of	(A OPND)
8	10	ITMP =	- Complement	of	(Data IN)
9	11	ITMP =	Complement	of	(B OPND)
10	12		Complement		

Then ICN is examined if = 0, 177777 (all ones) are added to ITMP, and the bits to the left of the 16-bit simulated word length are masked off. If ICN = 1, ITMP is not changed. Then the modifier is examined, and, if = 7, the Q-register is set equal to ITMP, and control is returned. Otherwise, the B-operand word in the scratchpad is set equal to ITMP, and INMOD is called.

m. Subroutine CMPLT (Two's Complement)

This subroutine is entered when the OP code is 11 to 14 and performs almost identical operations to those of INVRT:

OP Code

Decimal	Octal	Result
11	13	ITMP = Complement of (A OPND) + ICN
12	14	ITMP = Complement of (Data IN) + ICN
13	15	ITMP = Complement of (B OPND) + ICN
14	16	ITMP = Complement of (Q REG) + ICN

In this case, if ICN = 0, ITMP is not affected; and if ICN = 1, ITMP is increased by 1, thereby performing a 2's complement on the selected data. Then, if the modifier is 7, the Q-register is set to ITMP. If the modifier is other than 7, the value of the B-operand is set to ITMP and INMOD is called.

n. Subroutine TRINC (Transfer/Increment)

This subroutine is called when the OP code is 15 to 18, with the following results:

OP Code

Decimal Decimal	Octal	<u>Result</u>
15	17	ITMP = (A OPND) + ICN
16	20	ITMP = (Data IN) + ICN
17	21	ITMP = (B OPND) + ICN
18	22	ITMP = (Q REG) + ICN

Here, when carry is 0, the effect is a simple transfer of the contents of the designated operand to the designated B-operand or the Q-register, depending on the modifier. Note that operation 17 (21 octal) will have a no OP effect if the modifier is 0. The emulator still performs the load ITMP from B and stores ITMP back in B, however. If carry = 1, the operation includes an increment. If the modifier is 7, the value in ITMP is stored in the Q-register. Otherwise, the value is stored in the designated B-operand and INMOD is called.

p. Subroutine DECTR (Decrement/Transfer)

This subprogram is called when the operation code is 19 to 22, with results similar to TRINC above:

OP Code

<u>Decimal</u>	Octal	<u>Result</u>
19	23	ITMP = (A OPND)
20	24	ITMP = (Data IN)
21	25	ITMP = (B OPND)
22	26	ITMP = (Q REG)

When the basic operation of loading ITMP is completed, the value of carry is examined. If carry = 0, than all 1's (177777 octal) is added to ITMP, and ITMP is then masked back to 16 bits. Thus the effect is to decrement ITMP by 1, since the action constitutes addition of a minus one. This technique is used in the emulator because that is the way the operation is performed in the microprosessor hardware. If carry = 1, no further change is made yet to ITMP. Here again is a potential no-OP, in OP code 21 (25 octal), when carry is on and the modifier is 0. Again, ITMP is placed in Q-register if the modifier is 7; otherwise, it is placed in the B-operand and INMOD is called.

q. Subroutine SUM (Addition)

This subroutine is entered when the OP code is 23 to 25, and the following operations result:

OP Code

<u>Decimal</u>	<u>Octal</u>	Result
23	27	ITMP = (A OPND) + (B OPND) + ICN
24	30	ITMP = (Data IN) + (B OPND) + ICN
25	31	ITMP = (A OPND) + (Q REG) + ICN
26	32	ITMP = (Data IN) + (Q REG) + ICN

Here, the effect of carry is to increase the sum by one if on (ICN = 1), or no effect if off (ICN = 0). If the instruction modifier value is 7, Q-register is loaded with the value in ITMP, and control is returned to the calling program. Otherwise, the B-operand is loaded from ITMP and INMOD is called.

r. Subroutine DIFF (Subtraction)

This subroutine is called when the operation code is 27 to 31, and is the last of the operation analysis/execution subroutines. The effects are:

OP Code

Decimal	Octal	Result
27	33	ITMP = (A OPND) - (B OPND)
28	34	ITMP = (B OPND) - (A OPND)
29	35	ITMP = (Data IN) - (B OPND)
30	36	ITMP = (B OPND) - (Data IN)
31	37	ITMP = (Data IN) - (Q REG)

The effect carry has on this operation is to reduce the result by one when carry is off (ICN = 0), or no effect when on (ICN = 1). Again, ITMP is stored in the Q-register if the instruction modifier is minus 7; otherwise, ITMP is stored in the B-operand and INMOD is called.

s. Subroutine INMOD (Instruction Modification)

This subprogram is called from the applicable instruction interpretation/execution routine when the instruction modifier is not 7, but 0 to 6. At this point, the value produced by execution of the operation has been placed in the scratchpad word pointed to by the B operand of the microinstruction. The following actions are now taken based on the value of the modifier:

Modifier	<u>Action</u>
0	(Data OUT) = (B OPND)
1 2	(Data OUT) = (A OPND) ('A' Latch).
2	(Data OUT) =old (B OPND), before execution of operation changed its value ('B' Latch).
3	Shift (B OPND) left one bit - zero enters (B OPND) from the right, and the bit shifted off is inverted and ORed to SR1 (MSR1).
	Result is also placed in (Data OUT).
4	Shift (B OPND) right one bit - a one bit enters (B OPND) from left, and the bit shifted off is lost. Result is placed in Data Out.
5	Shift (Q-REG and (B OPND) together left one bit - a one bit enters q from right - MSB of (B OPND) → SR1. Data OUT is then set to the resultant (B OPND) value.
6	Shift (Q-REG) and (B OPND) together right one bit - a one bit enters the MSB of (B OPND), and the bit shifted off of the Q-REG is inverted and ORed to SR1 (MSR1). Data OUT is then set to the new value of (B OPND).

Note: Data OUT is represented by the variable IDOUT. Control is then returned to the calling routine.

t. Subroutine BINRY

The binary control bits which were not examined for required action prior to operation code execution are now evaluated. They are:

Field (Ref. Figure 2)	Octal Bit No.	Action if BIT = 1 (on)
7	33	SR2 = SR1 (MSR2 = MSR1)
8	34	H - REG = CM (addr) (MEMH = MICRO(5))
9	35	MEM (C) = Buss (MAINM (MEMC) = MBUSS)
. 11	37	Buss = ALU Data OUT (MBUSS = IDOUT)
12	40	Fetch (Confirm that H REG = 0; if not, ERROR = .TRUE.)
14	42	SR1 = 1 if Data OUT = 0
16	44	H - REG = SR1 ORed to LSB of H - REG
18	46	Telemetry = Buss (ITELM = MBUSS)*
19	47	C - REG = Buss (MEMC - MBUSS)

*Note: Telemetry bit effect is inverted, i.e., telemetry = buss when bit $46_8 = 0$.

This routine is called once per microinstruction.

u. Subroutine FINAL

This subroutine is entered following the completion of a complete macrooperation, and evaluates the run-termination criteria which were setup by the run control card. If the termination criteria are satisfied, then the logical variable FINIS is set to .TRUE., otherwise no action is taken, and control is returned to the calling routine (MAIN). This subroutine is called once per macro-operation.

v. Subroutine STATS

This subroutine is entered after the simulated microprocessor run has been terminated or aborted. The routine prints out the contents of all registers, data paths, and the 16-word scratchpad, plus the simulated run time, number of microinstructions, and the number of macros executed. Additional output can be added as desired. When control is returned to the main program, the program stops.

w. Subroutine REPT1

This subroutine is called when the level of diagnostic output required is high as indicated on the run control card. Currently, outputs result in the execution of individual microinstruction. It is anticipated that this subroutine will be expanded as more experience is gained in the development of microprograms.

x. Subroutine REPT2

Stub only - not yet implemented.

y. Subroutine XOR

Not used for CDC 6000 version: CDC function XOR.

This subroutine was developed prior to the availability of exclusive OR in the R-520 function library. It performs an exclusive OR on the second or third parameters, and returns the results in the first parameter. It uses the following algorithm:

CALL XOR (IANS, J, K)
IANS = ((J)AND(K))OR((J)AND(K)).

z. Subroutine FLEX

Not used for CDC 6000 version.

This subroutine contains all the required direct code (R-520 FLEXTRAN) for the emulator. Two functions are currently performed, based on the key in parameter one: If key = 1, the subroutine converts a right-adjusted real (double-word) variable to an integer variable. If K = 2, the subroutine increments the H-field (bit 31 octal) of a 48-bit microinstruction by 1, used in presetting control memory.

5. Preliminary Results

The basic design goal of providing a software development and test capability for the digital control computer application of the microprocessor has been met, and the emulator is now serving this capacity. It has been determined that the ability to vary the intermediate output over a very wide range has been difficult to implement, but further work is planned in this area. The current version allows essentially three options of output: (a) a full printout of simulated control and main memory after loading (Table 4), (b) a very detailed printout of the intermediate and final results of each instruction execution (Table 5), and (c) a final status of all registers and data paths after termination (Table 6). The last of these three is currently not under run card control, but the other two are. It is not yet known

TABLE 4. MICROPROCESSOR SIMULATED MEMORY CONTENTS

	LOC	ATION	CONTE	ENTS	3	
DEC		DCTAL	MAIN MEMORY		CONTROL MEMORY	
	38	UU46	00000046			
	39	UU47	00000047		4200001162000000	
	40	0050	00000050		4200001202000000	
	41	U051	00000051		4200001555000000	
	42	U052	00000052		4200001242000000	
	43	U053	00000053		4200001262000000	MANUFACTURE OF STATE OF STATE OF
	14	U054	00000054		4200001302000000	
	45	0055	00000055		4200001322000000	
	46	U056	00000056		4200001342000000	
	47	U057	00000057		4200001362000000	
	48	0060	00000060		4200001402000000	
	49	U061	00000061		4200001422000000	
	50	0062	00000062	-	4200001442000000	NO-OP, NEXT H = HERE
	51	0063	00000063		4200001462000000	
	52	0064	00000064		4200001502000000	
	53	0065	00000065		4200001522000000	
	54	U066	00000066		4200001542000000	
	55	0067	00000067		4200001562000000	
						OW WELL LAND OF SECTION
	56	0070	00000070		4200001602000000	
	57	0071	00000071		4200001622000000	
	58	0072	00000072		4200001642000000	
	59	0073	00000073		4200001662000000	
	60	0074	00000074		4200001702000000	
	61	0075	00000075		4200001722000000	
	62	0076	00000076		4200001742000000	
	63	0077	00000077		4200001762000000	
	64	0100	00067000		4200002002000000	
	65	0101	00000101		3676742052004000	
	66	0102	00000102		4360702102000000	
	67	0103	00000103		4200002632000000	
	68	0104	00000104		0140702136002000	
	69	0105	00000105		4355702156000000	
	70	0106	00000106		4355702176000000	
	71	0107	00000107		4355702216000000	
	72	0110	00000110		4355702236000000	
	73	U111	00000111		4355702256000000	
	74	0112	00000112		4355702276000000	
	75	0113	00000113		4355702316000000	GENERAL PURPOSE
	76	U114	00000114		4355702336000000	MULTIPLY
	77	0115	00000115		4355702356000000	MACRO INSTRUCTION
	78	U116	00000116		4355702376000000	MACKO INSTRUCTION
	79	0117	00000117		4355702416000000	
	80	0120	00000120		4355702436000000	***
	81	0121	00000121		4355702456000000	
	82	U122	00000122		4355702476000000	
					4355702516000000	
	83	0123	00000123		4215702522000000	
	84	0124	00000124			
	85	0125	00000125		3664202542004000	
	86	0126	00000126		42000000002000000	
	87	0127	00000127		3200040002002000	
	88	0130	00000130		4200000002000000	
	89	0131	00000131		3360702642000000	
	90	0132	00000132		0000505105005000	
	91	U133	00000133		4200002662000000	
	92	0134	00000134		4200002702000000	
	93	0135	00000135		4200002722000000	
	94	U136	00000136		4200002742000000	
	95	U137	00000137		4200002762000000	
	0.6	11111	00000140		4200003002000000	

TABLE 5. DETAILED EXECUTION OUTPUT

1413 INSTR AT:1425, H-FIELD = 1426 .

1 N9TR W4 EXECUTE 1 N9TR W4 EXECUTE 1 N9TR W4 EXECUTE 2 CRACTED OP CODE 3 JUL 3 CRATCH PAD(01) 3 CRATCH PAD(11) 3 CRATCH PAD(11)	INSTR U4 EXECUTED, TEMP VALUE = 0134U1.	INSTR MOD = 3 JUST SET DATA-OUT = 00027010.	EXECUTED OF CODE 04 WITH MODIFIEP 3.	13 = 003001	8-REG 15 CUNTENTS INITIALLY = 001400, NOW = 027010	G-REG CONTENTS INITIALLY . 077777, NOW . 077777	IELEM CONTENTS INITIALLY . 001400, NOW . 016005	THE FOLLOWING CM INSTRUCTION HAS JUST BEEN EXECUTED:	05422n3400 n4 03 14 15 H:1406 0 0 1 n 0 1 0 0 0 0 1 1 1 0 0 0 0 0	SCRATCH PAD(10) CONTAINS OUTING . SCRATCH PAD(11) CONTAINS OUTING . SCRATCH PAD(12) CONTAINS OUTING . SCRATCH PAD(13) CONTAINS OUTING . SCRATCH PAD(14) CONTAINS OUTING . SCRATCH PAD(16) CONTAINS OUTING . SCRATCH PAD(10) CONTAINS OUTING . SCRATCH PAD(11) CONTAINS OUTING . SCRATCH PAD(11) CONTAINS OUTING . SCRATCH PAD(11) CONTAINS OUTING . SCRATCH PAD(12) CONTAINS OUTING . SCRATCH PAD(13) CONTAINS OUTING . SCRATCH PAD(12) CONTAINS OUTING .
	TED, TEMP VALUE	JUST SET DATA-OUT	DE 04 WITH MODIF!	A-REG 14 CONTENTS = 003001	NTS INITIALLY .	INITIALLY . 0777	INITIALLY . 0014	CM INSTRUCTION HA	1074670542203490 04	PAD (10) CONTAINS 001000 PAD (11) CONTAINS 001000 PAD (12) CONTAINS 001000 PAD (13) CONTAINS 001000 PAD (13) CONTAINS 001000 PAD (13) CONTAINS 001000 PAD (11) CONTAINS 001000 PAD (11) CONTAINS 001000 PAD (12) CONTAINS 001000 PAD (13) CONTAINS 001000 PAD (14) CONTAINS 001000 PAD (14) CONTAINS 001000 PAD (15) CONTAINS 001000

TABLE 6. FINAL OUTPUT

00001442 0000000 90000000 00000000		,	•	The second second second		
0000000						
00000000		1	-			
00000000						
00177303					-	
90174606						
00076605			-			
0000000						
00171547						
00161212						
00010512						
00177776						
777777						
00002364						
00077777						
00035376						
DATA IN = U0001777, DATA OUT = 00077777, BUSS = 00077777, C-REG = 1777, H-REG = 0000, 0-REG = 177777	BUSS . 000	77777 C-	PEG = 17	777, H-REG :	# 0000 B-R	E.G = 17777
A/D S = 00000000 00000000 00000000 00000000	0000000	0000000	0/A S .	00000000	00000000 00000000 00000000	00000000
	:					
CARRY . U. INPUT DISCRETES . DOCUMENT		1	-			

if the ability to vary more widely the intermediate detail will be worth the additional parameter testing that will be required to implement it, but further analysis is planned.

The structure of the control mechanisms of the emulator was of some concern, and it was not readily obvious what criteria should be used to manage a given emulation run. It was finally decided to use two logical (Boolean) variables, ERROR and FINIS, to control the runs. Both are initialized to .FALSE., and both are tested at the beginning of each simulated Macroinstruction execution. Any error condition in any subroutine causes an error message to be output at that point, and then causes the error variable to be set to .TRUE., which then causes the run to be terminated at the next macro-FETCH stage of execution. Upon completion of a macroinstruction, the normal run-termination conditions defined by the run control card are examined, and if the conditions are met, causes FINIS to be set .TRUE., with the same result. This technique allows the program termination to be controlled, and meets the structured programming criteria of one-entrance, one-exit for the main program. All subroutines have this same characteristic of onein, out-out. In general, it is felt that valuable additional experience with the use of structured FORTRAN forms has been gained.

6. Conclusion and Future Work

The emulator described in this report is currently operational on the Raytheon R-520 and CDC 6600 computers, and is currently supporting the applications software development process for the first multipurpose microprocessor feasibility application, that of guidance control computer on the T-6 missile. It is anticipated that extensions to the emulator will be made during the flight software development. Some of the planned extensions are:

- a) Experimentation with various forms and frequencies of emulator output, and with interactive control of emulator runs.
- b) Experimentation with different microprocessor software structures, such as the development of meaningful operation codes at the main memory level, as opposed to simply pointing to macros and data.
- c) Experimentation with variations in the word size of main and control memory.
- d) Efforts to implement the emulator on other host computers, including the HP 2100.

In addition to the above, but probably later in time, it is anticipated that studies will be made of the potential of generalizing from the current emulator form to allow external definition of instruction/control fields and their meanings. This will involve attempts to support microprogramming for other microprocessors, the accumulation of information on similarities and differences, and the analysis of this data

against the original structure of this emulation effort. Another planned extension will provide an interactive capability for testing and modifying microprograms.

One outstanding characteristic of this work that should be emphasised is the comparative ease of making necessary or experimental changes to the emulator. In a new-processor environment, where both unforeseen and planned hardware changes are frequent, the simplicity of making fast changes to the emulator program has been invaluable. The modular design coupled with the strict adherence to the principals of structured programming are primarily responsible for this capability. More specifically, when a hardware function was initially misunderstood, or later required modification, the exact location of the emulator representation of that function was immediately known, easily isolated, and readily changed. This feature is also expected to play a vital role in future experimentation with proposed hardware design variations, in that the effects as seen by software can be thoroughly evaluated very easily prior to making any actual hardware changes.

BIBLIOGRAPHY

- Aitken, J. D., Sliz, C. J., and Leonard, J. P., <u>T-6 Missile Description</u>, US Army Missile Command, Redstone Arsenal, Ala., Technical Report RG-73-18, 10 August 1973.
- Asquith, C. Frank, <u>T-6 Digital Autopilot Data Processing Analysis and Specification</u>, US Army Missile Command, Redstone Arsenal, Ala., Technical Report RG-75-36, 5 March 1973.
- Copeland, E., Computer Evaluation Techniques, US Army Missile Command, Redstone Arsenal, Ala., Technical Report RG-72-3, January 1972.
- Copeland, D. E., Jones, M. C., and Strickland, M. R., <u>Improved HAWK</u>
 <u>Software Simulation</u>, US Army Missile Command, Redstone Arsenal, Ala.,
 <u>Technical Report RG-75-1</u>, 15 July 1974.

Appendix A. MULTIPURPOSE MICROPROCESSOR EMULATOR: DATA ELEMENT DEFINITIONS

1. Common

CNTMEM (1024	SIMULATED CONTROL MEMORY, 48 BITS/WORD
ICN	SIMULATED CARRY IN - 1 BIT (ACTUAL 24)
ICOUNT	INSTRUCTION COUNT REGISTER, INITIALIZED
	AT O, INCREMENTED BY 1 IN EXECM (MICRO)
IDIN	SIMULATED DATA IN LINES - 16 BITS (ACTUAL 24)
IDOUT	SIMULATED DATA OUT LINES - 16 BITS (ACTUAL 24)
IDSCAR	SIMULATED INPUT DISCRETE REGISTER -
IDOCAR	4 BITS (ACTUAL 24)
ITELM	SIMULATED TELEMETRY REGISTER - 16 BITS
TICLE	(ACTUAL 24)
MATNW (102/)	SIMULATED MAIN MEMORY, 16 BITS/WORD (ACTUAL 24)
MAINM (1024)	SIMULATED DATA BUSS, 16 BITS (ACTUAL 24)
MBUSS	SIMULATED MAIN MEMORY ADDRESS (C) REGISTER
MEMC	
	10 BITS (ACTUAL 24)
MEMH	SIMULATED CONTROL MEMORY ADDRESS (H) REGISTER
(5)	10 BITS (ACTUAL 24)
MICAD (5)	SIMULATED A/D INPUT REGISTERS, 12 BITS
.= (0)	EACH (ACTUAL 24)
MICDA (3)	SIMULATED D/A OUTPUT REGISTERS, 12 BITS
	EACH (ACTUAL 24)
MICQ	SIMULATED Q REGISTER - 16 BITS (ACTUAL 24)
MSCPAD (16)	SIMULATED MICROPROCESSOR SCRATCHPAD
	MEMORY - 16 BITS EACH (ACTUAL 24)
	MSCPAD (1) = PROGRAM COUNTER
	MSCPAD (2) THRU (4) = RETURN ADDRESSES
	MSCPAD (5) THRU (8) = X REGISTERS
	MSCPAD (9) THRU (12) = A REGISTERS
	MSCPAD (13) THRU (16) = B REGISTERS
MSR1	SIMULATED SR1
MSR2	SIMULATED SR2
NRUN (20)	RUN CONTROL PARAMETERS:
	NRUN (1) - PRESET VALUE FOR CONTROL MEMORY
	NRUN (2) - PRESET VALUE FOR MAIN MEMORY
	NRUN (3) - OUTPUT LEVEL
	NRUN (4) - INITIALIZATION OUTPUT LEVEL
	NRUN (5) - INITIAL PC VALUE (START)
	NRUN (6) - OUTPUT CONTROL INTERNAL - TIME
	NRUN (7) - OUTPUT CONTROL INTERVAL - INSTR. COUNT
	NRUN (8) - OUTPUT CONTROL - ADDRESS IN H
	NRUN (9) - TERMINATION CRITERIA
	NRUN (10) - TERMINATION VALUE
	NRUN (11) - FORMAT OF MEMORY LOAD USE
	REMAINING NRUN OPTIONS RESERVED FOR FUTURE USE -
FINIS	LOGICAL: NORMAL RUN TERMINATION FLAG
ERROR	LOGICAL: ERROR TERMINATION FLAG
EARUR	LOUICAL. EXXON IENTIMATION PLAG

MINSTR	MAIN MEMORY INSTRUCTION - 8 BITS (ACTUAL 24)
MOPND	MAIN MEMORY OPERAND - 8 BITS (ACTUAL 24)
MICRO (25)	MICROINSTRUCTION FIELDS: (ALL ACTUAL 24 BITS) MICRO (1) - OPERATION CODE - 5 BITS
	MIC RO (2) - INSTRUCTION MODIFIER - 3 BITS
	MICRO (3) - OPERAND 1 ADDRESS - 4 BITS
	MICRO (4) - OPERAND 2 ADDRESS - 4 BITS
	MICRO (5) - NEXT MICRO (H) ADDRESS - 10 BITS
	MICRO (6) THU (24) - BINARY CONTROL FIELDS - 1
	BIT EACH
	MICRO (25) - A/D MUX ADDRESS - 3 BITS
MCNT	MACRO COUNT - INCREMENTED IN MAIN
STIME	SIMULATED ELAPSED TIME - 750 NSEC/MICRO EXECUTION
ITIME	INTEGER VALUE OF STIME IN SECONDS
IETIM	ELAPEED EMULATION RUN TIME
KAD	KEY THAT D/A MUX HAS BEEN CLOCKED
KDA	KEY TO SELECTED A/D REGISTER

2. Noncommon

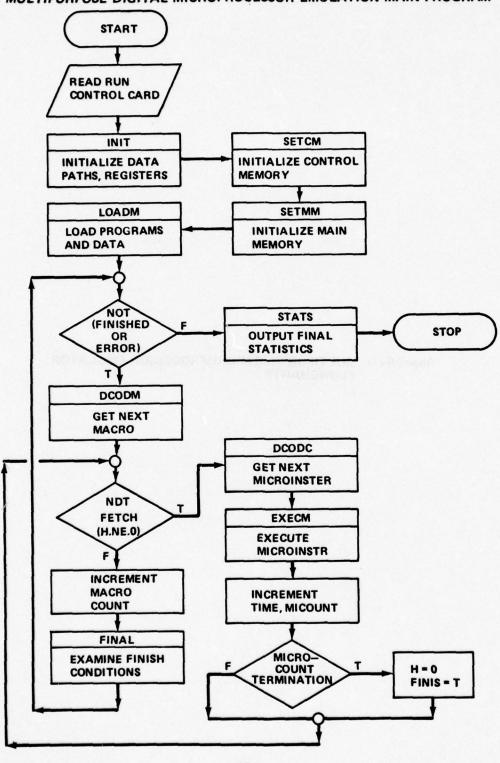
Subprogram	Parameter	
in EMUL (MAIN)	LFTMS	8-BIT MASK (377 OCTAL)
	TINCR	TIME INCREMENT (0.00000075 SEC = 75 NSEC)
	MICS	MICROINSTRUCTION COUNT
in SETCM	REPL /00/	TO PRESET CONTROL MEMORY TO CLEAR
	REPL 2	TO PRESET CM TO NOP TO HERE
	INDX	INDEX THRU MEMORY, 1 - 1024
	ADDR	INCREMENT H FIELD BY ONE (CDC VERSION ONLY)
	MSK	8-BIT MASK
SETMM	INDX	INDEX THRU MEMORY, 1 - 1024
LOADM	MSK 1	MASK FOR INVERSION OF BITS 2, 3 OF OP CODE
	KEY	MAIN OR CONTROL MEMORY, OR END LOAD
	MADR	OCTAL ADDRESS
	MAIN	VALUE FOR MAIN MEMORY
	CNTRL	VALUE FOR CONTROL MEMORY
	K	MAIN MEMORY WORD COUNTER
	L	CONTROL MEMORY WORD COUNTER
DCODC	IERR	CUMULATIVE ERROR FLAG
	IPTR	CALLING PARAMETER FOR MEMH
	MOP	CALLING PARAMETER FOR MICRO(N)
	INDX	MICRO INDEX
	NCOL	FIELD (COLUMN) INDEX
	I FLAG	ERROR RETURN FLAG
	MSK 1	MASK FOR INVERSION OF BITS 2, 3 OF OP CODE
DCODM	IOFF/01400	OFFSET FOR 1st INSTR OF EACH MACRO
	MIN	CALLING PARAMETER FOR MINSTR
	MOP	CALLING PARAMETER FOR MOPND
	2	

GETCM	IADR	ADDRESS OF SOURCE WORD
GETCH	IBITL	LEFT (START) BIT OF DESIRED FIELD
	IBITN	NUMBER OF BITS IN FIELD
	IBYTE	ADDRESS OF TARGET (FIELD) WORD
	I FLAG	ERROR FLAG
	IDR	IADR + 1 TO OFFSET O BASE FOR INDEX
	OPMSK/IMSK	MASK FOR EXTRACTING FIELD
	BYTE	48 - BIT ANSWER FIELD (R-520 VERSION
	BILL	ONLY)
	NMBR	FINAL SHIFT FOR RIGHT ADJUST
	ISTBIT	START BIT OFFSET FOR 60-BIT WORD
		(CDC VERSION ONLY)
GETMM	IADR	
	IBITL	
	IBITN	SAME AS GETCM
	IBYTE	
	I FLAG	
	IDR	
	IBIT	MACHINE-WORD ADJUSTMENT FOR START BIT
	IMASK	MASK FOR EXTRACTING FIELD
	NMBR	FINAL SHIFT FOR RIGHT ADJUST
INMOD	MOD	INSTRUCTION MODIFIER (MICRO(2)) - USED TO SPECIFY DESTINATION OF OUTPUT AND TO CONTROL SHIFT OPERATIONS.
	IA	INDEX OF A OPERAND
	IB	INDEX OF B OPERAND
	KMASK	RIGHTMOST 16-BIT MASK
	IHIBIT	LEFTMOST 1-BIT MASK (HIGH BIT)
XOR	IANS	ARGUMENT 1, RESULT OF EXCLUSIVE OR
(R-520 ONLY	J	ARGUMENT 2, VALUES TO BE EXLUSIVE
	K	ARGUMENT 3, OR ED
EXECM	IOP	OPERATION CODE (MICRO (1))
OPERATIONS	IOP	OP CODE (MICRO (1))
BOLN	LHIGH	OCTAL 177777 (FORCE ALL 1s), 16-BIT MASK
INVRT	MOD	INSTRUCTION MODIFIER (MICRO(2))
CMPLT	IAR	A OPERAND INDEX (MICRO(3))
TRINC	IBR	B OPERAND INDEX (MICRO(4))
DECTR		
SUM	ITMP	TEMPORARY HOLD FOR RESULTS OF
DIFF		EXECUTION UNTIL FINAL FORM / DESTI - NATIONS IS DETERMINED
PRBIN	ILOW	4 LSBs OF DATA IN (IDIN) OR DATA OUT (IDOUT)
BINRY	KOUNT	ELAPSED TIME (OPERATIONS) COUNTER FOR D/A OUTPUT REGISTER SELECTION
	MSK 10	10-BIT MASK FOR ADDRESSES
FLEX	К	KEY TO SELECTED DIRECT CODE OPTION

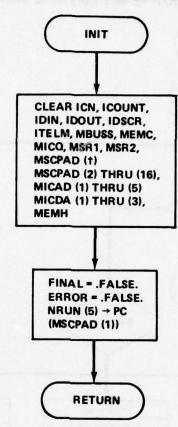
(R-520 ONLY)	FLTB	REAL (48-BIT) PARAMETER IN OR RETURN
FINAL	INTEGR NEND	INTEGER (24-BIT) PARAMETER OR BOTH TERMINATION CRITERIA (NRUN(9))
	NVAL	TERMINATION VALUE (NRUN(10))
REPT1	ITEL2	PREVIOUS VALUE OF TELEMETRY REGISTER
	MT CO2	PREVIOUS VALUE OF Q-REGISTER

Appendix B. MULTIPURPOSE MICROPROCESSOR EMULATOR: FLOWCHARTS

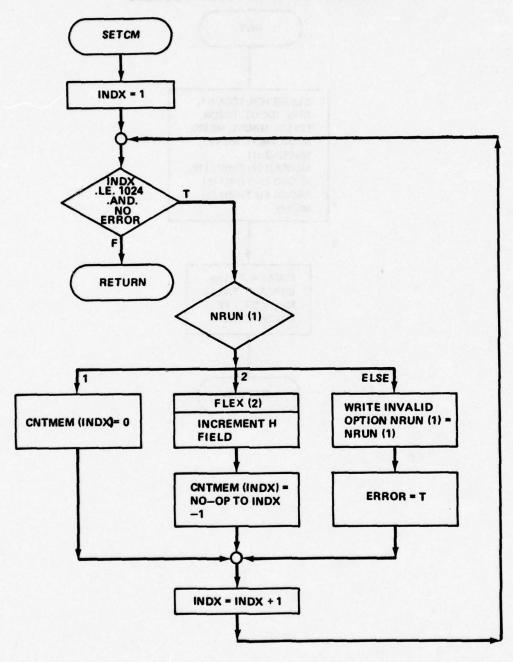
MULTIPURPOSE DIGITAL MICROPROCESSOR EMULATION MAIN PROGRAM



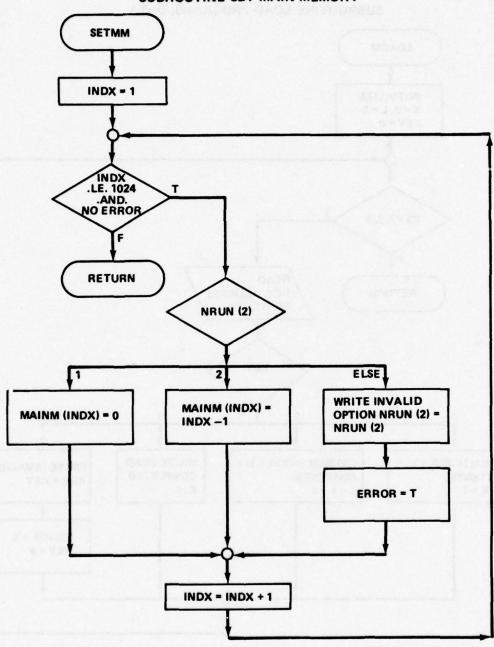
SUBROUTINE INITIALIZE (REGISTERS AND DATA PATHS)



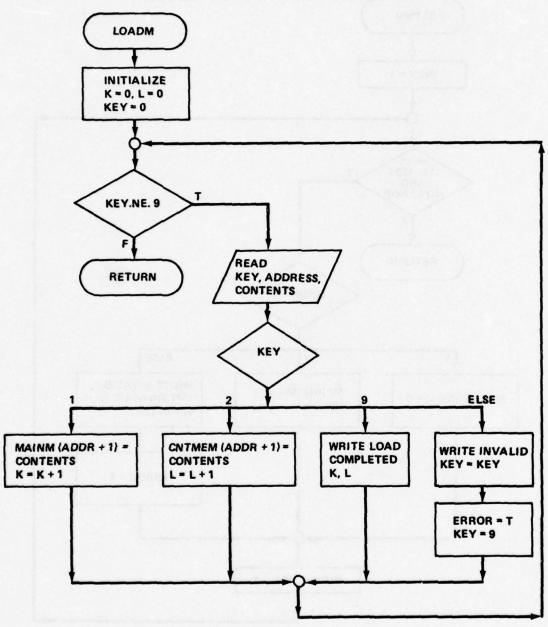
SUBROUTINE SET CONTROL MEMORY



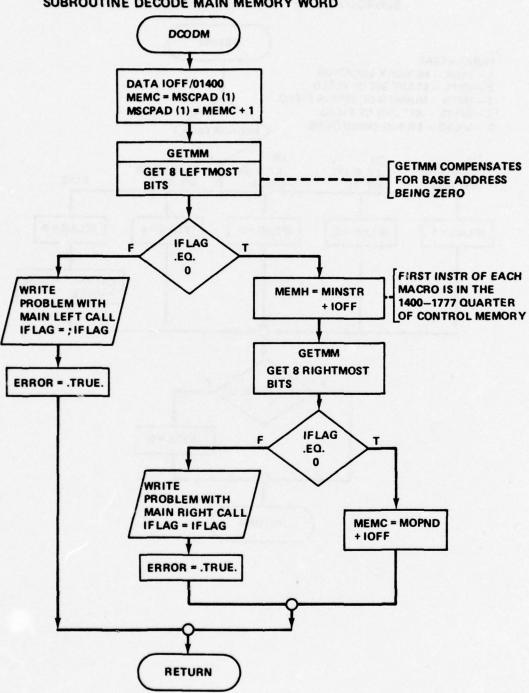
SUBROUTINE SET MAIN MEMORY



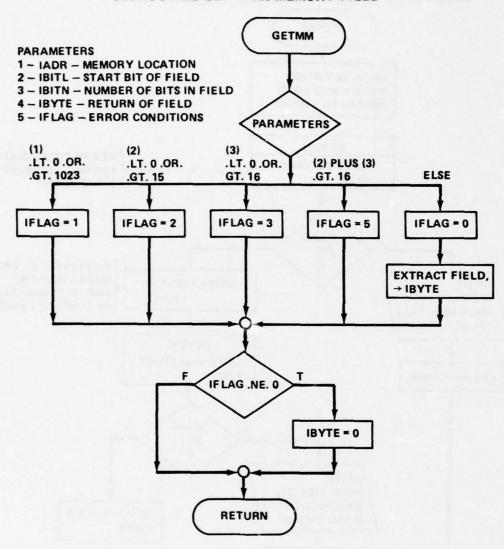
SUBROUTINE LOAD PROGRAMS, DATA



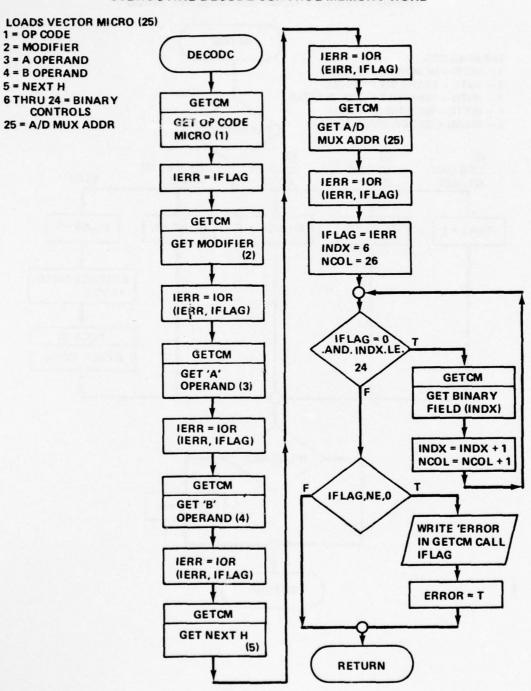
SUBROUTINE DECODE MAIN MEMORY WORD



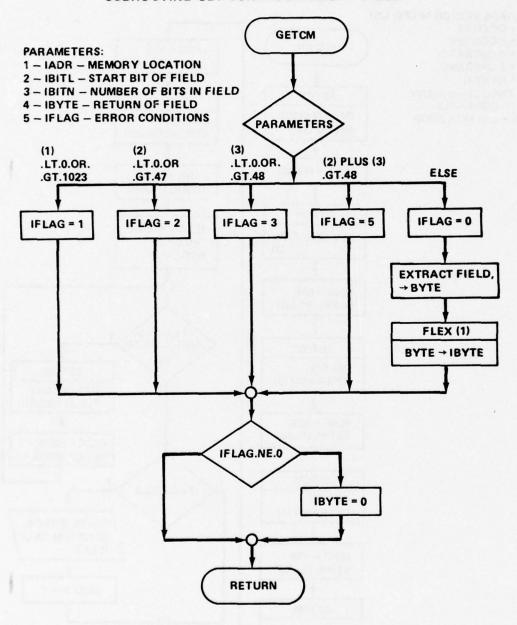
SUBROUTINE GET MAIN MEMORY FIELD



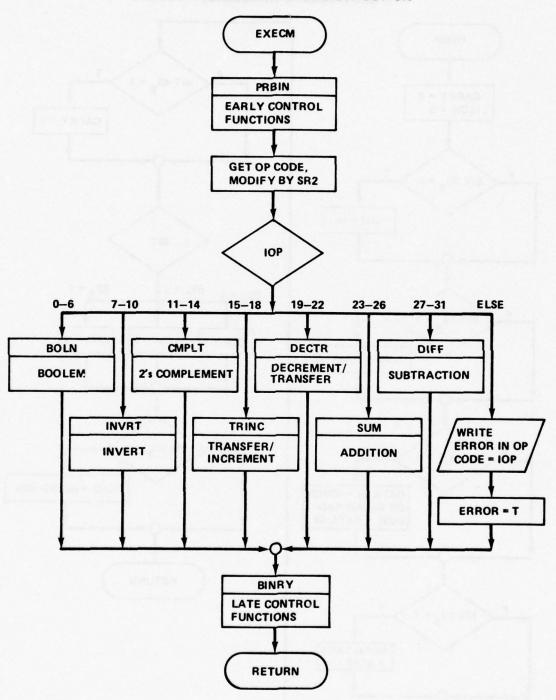
SUBROUTINE DECODE CONTROL MEMORY WORD



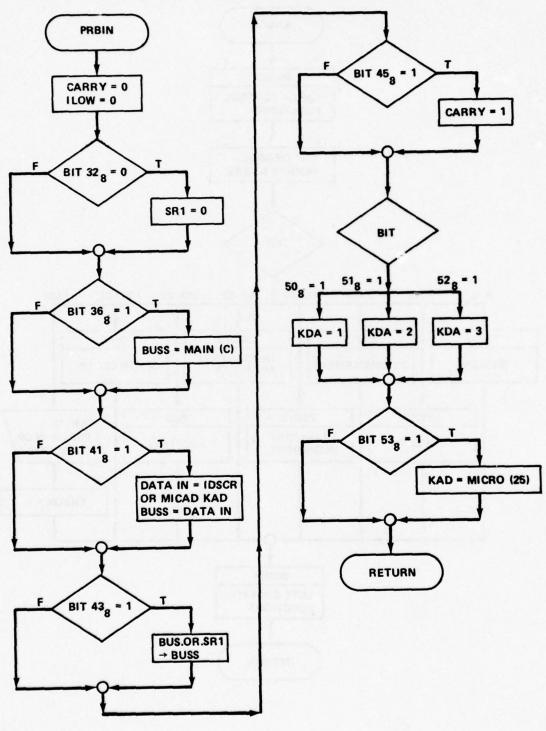
SUBROUTINE GET CONTROL MEMORY FIELD



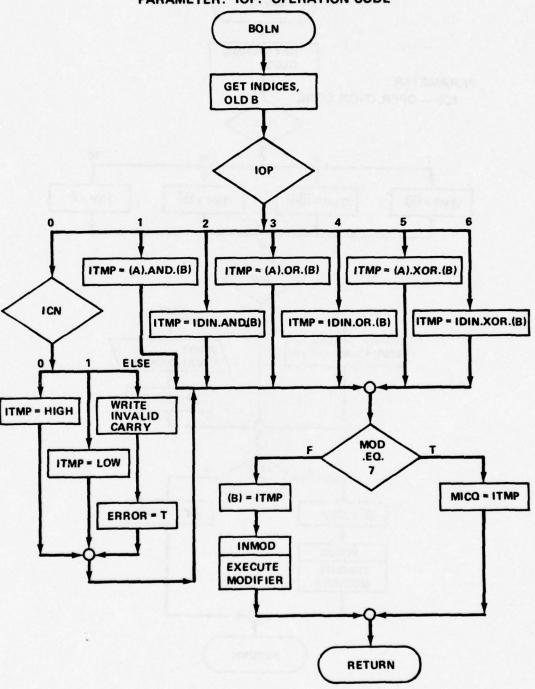
SUBROUTINE EXECUTE MICROISTRUCTION

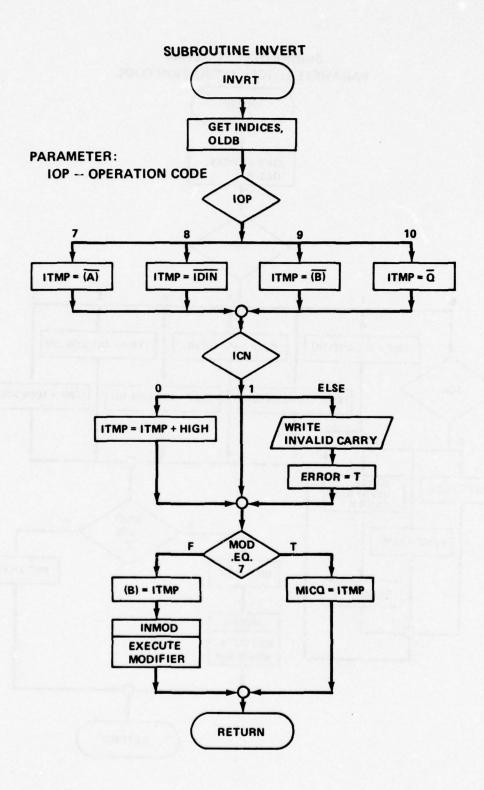


SUBROUTINE PRE-BINARY EARLY CONTROL FUNCTIONS (BIT NUMBERS IN BINARY)

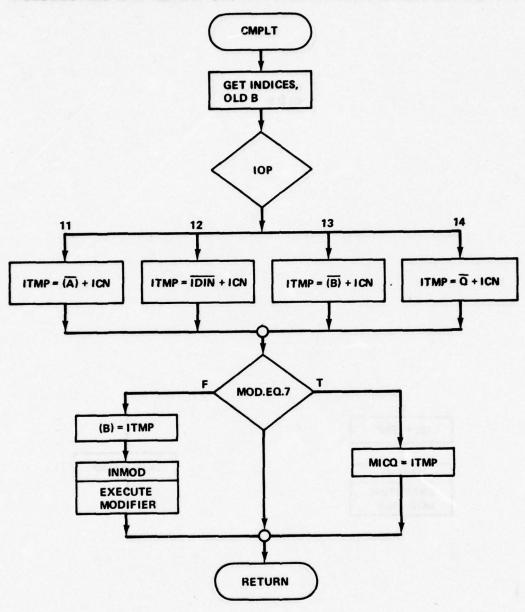


SUBROUTINE BOOLEAN PARAMETER: IOP: OPERATION CODE

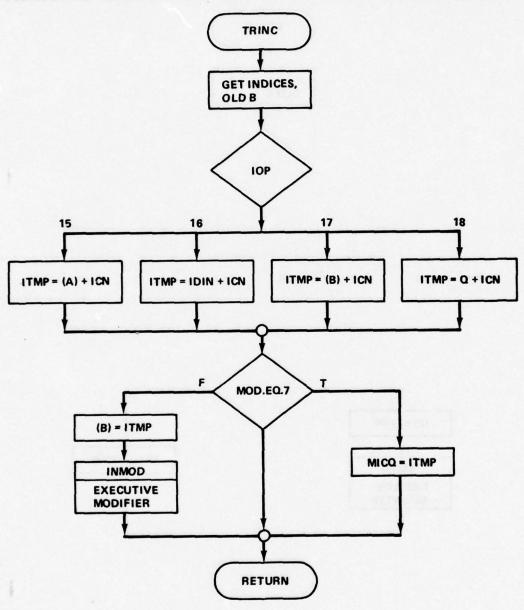




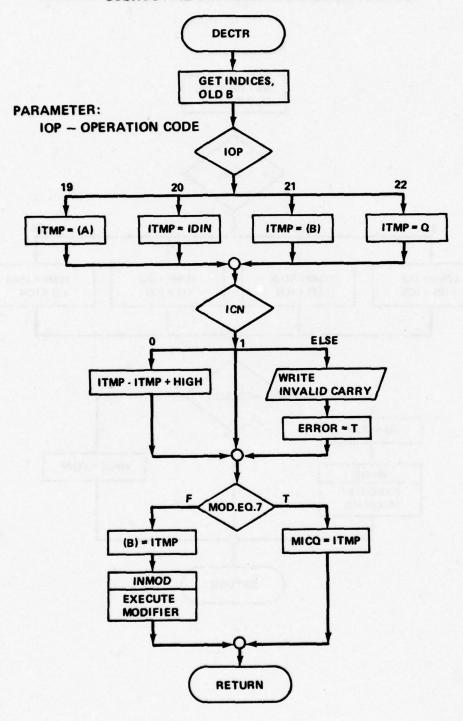
SUBROUTINE COMPLEMENT (2's) PARAMETER: IOP - OPERATION CODE



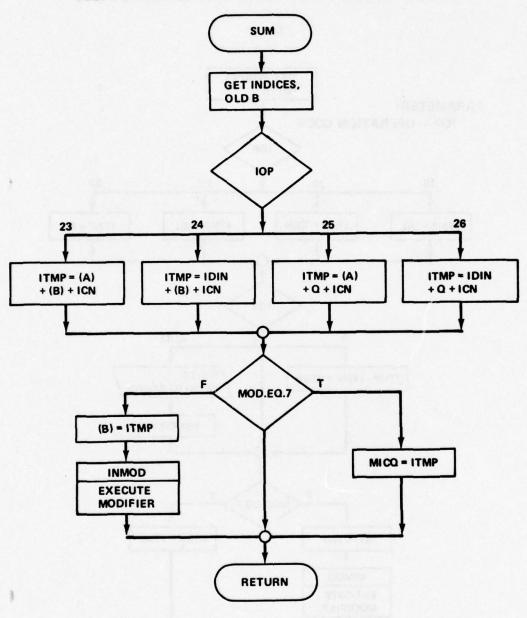
SUBROUTINE TRANSFER/INCREMENT PARAMETER: IOP - OPERATION CODE



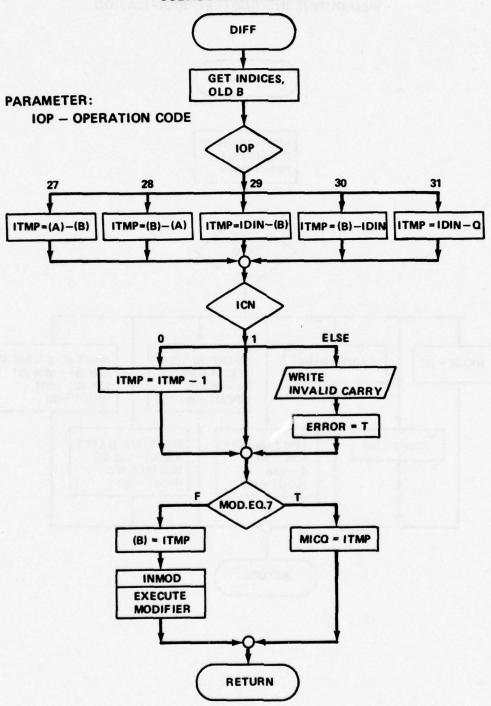
SUBROUTINE DECREMENT/TRANSFER



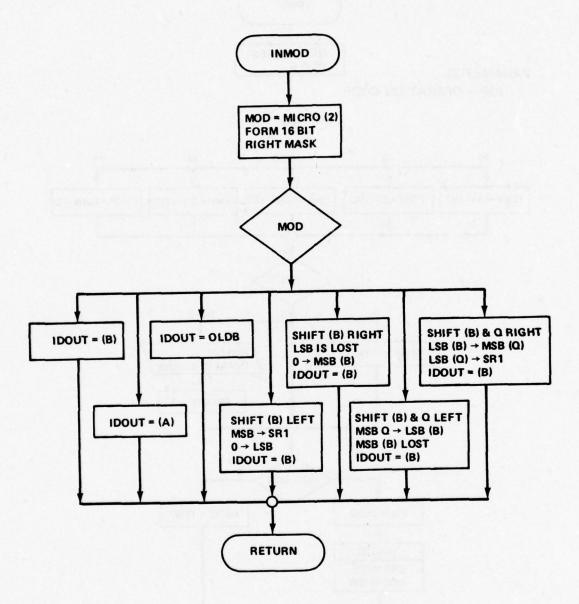
SUBROUTINE ADD PARAMETER: IOP - OPERATION CODE

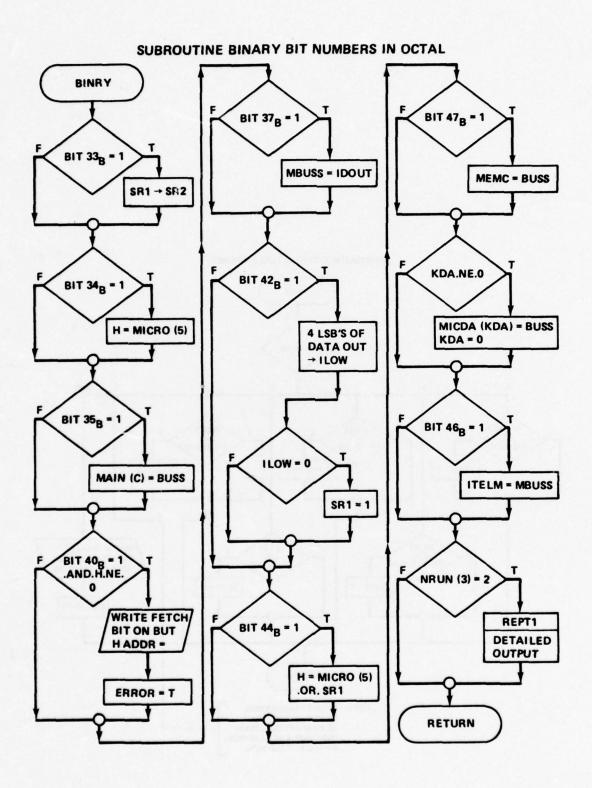


SUBROUTINE SUBTRACT

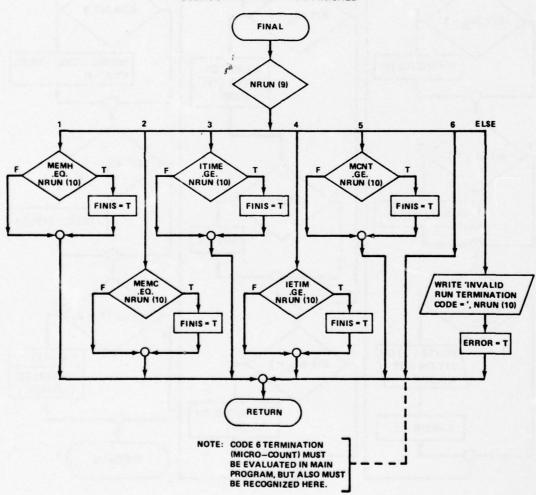


SUBROUTINE INSTRUCTION MODIFICATION

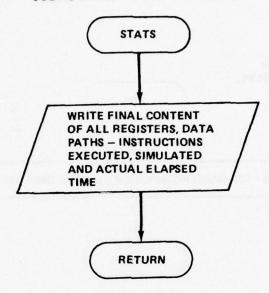




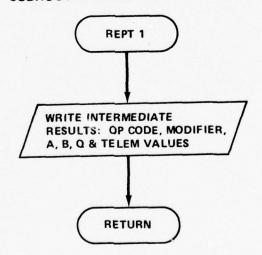
SUBROUTINE CHECK FOR FINISHED



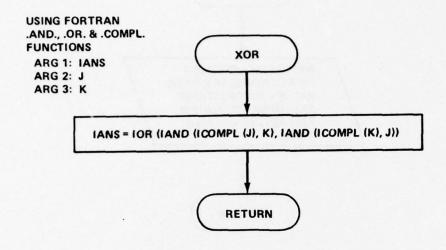
SUBROUTINE FINAL STATISTICS



SUBROUTINE DIAGNOSTIC OUTPUT



SUBROUTINE EXCLUSIVE OR (RAYTHEON R520 VERSION ONLY)



SUBROUTINE FLEXTRAN (RAYTHEON R520 VERSION ONLY) DIRECT R-520 CODE

PARAMETERS:

K - KEY, = 1 FOR FLTB → INTGR = 2 FOR INCREMENT H FIELD FLTB - REAL (48 BIT) VARIABLE INTGR - INTEGER (24 BIT) VARIABLE FLEX K ELSE 2 WRITE DOUBLE LOAD REAL INCREMENT H FUTURE **INVALID FLEX** LOWER HALF → FIELD OF FLTB **EXPANSION** KEY BY 1 INTGR RETURN

Appendix C. MULTIPURPOSE MICROPROCESSOR EMULATOR: PROGRAM LISTINGS: (1) STRUCTURED FORTRAN (2) STANDARD FORTRAN

	PROGRAM EMUL (INPUT, DUTPUT, TAPES = INPUT, TAPEZ=DUTPUT)	000110
200000	06 T 000 1000000000000000000000000000000	000000000000000000000000000000000000000
	PRUGRAM NAME & MULTI-PURPOSE DIGITAL MICROPROCESSOR EMULATOR	+000150
00	T - 5 AUTOPILOT VERSION	*300160
	AUTHOR I JERRY & BROOKSHIRE	.000180
	US ARM MISSILE COMMAND	.000190
0	PROJECT * MULTI-PURPOSE DISITAL MISROPROCESSOR	• 0000
	M ESTABLISHES THE	*00020 *000230
	HAKUMAKE ENVIRONHENT, PRESETS HAIN AND CONTROL HENDRY	*000240
, .	MATERIA, COMUS THE 68-BIT CONTROL MEMORY AND THE 16-BIT	*000250
	MEMORY (MICRO-) INSTRUCTIONS HADEN MEMORY SECTIONS	4000270
00	CONTROL, AND OUTPUTS COMPUTED VALUES PLUS PERFORMANCE AND TIMING DATA UNDER RUN-OPTIONS AS SPECIFIED.	• 90028
	AND SECURE OF THE PROPERTY OF	.000 300
	משוכנים פולקונים . שאוש בעקפעא	*000310
	SUBROUTINES CALLED .	+000330
-	SETCH - PRESET CONTROL MEHORY	900000
	SETAN - PRESET MAIN MEMORY	*000350
1	INIT - INITIALIZE ALL REGISTERS AND DATA PATHS	.000380
		-000370
	DCOUNT - DECODE MAIN MEMORY , SET PC. H, RESET C REGS	+000300
	FIRE STREET START THE CONTROL CHICKS-1 INSTRUCTION	060 000
		1000+
	STATS - GUTPUT FINAL STATISTICS UNDER OPTION CONTROL	***************************************
		• 000430
	:	077000
	20	00000
	CO 29 JULY	04,000
	MOD 02 19 SEP 1975	007000
-	03 6 NOV	10005
5	015000+	*000510
		000530
	Ī	845000
	H(1024) . ICN .	000000
	19SCK . ITELA	000200
~ -	S . MENC . MENA . LICADIGI,	000570
, ,	NEIN(201, FINTS FOODS MINERS MODES	900200
4	THE CARDY OF THE STATE OF THE S	965090
	1003	9000
-	1	00000
0	GET RUN CONTROL PARAMETERS	000630
	-	969640

20 FORMAT (23HIPUN CONTROL CARD = . 2304.)	000670
They a deducated	000000
, '	2000
CALLINI	20000
יייייייייייייייייייייייייייייייייייייי	000710
	1000726
	000730
	940040
14 PM	000 750
COLUMN HERORY CONTENTS (PRE-EASTON)	090760
	02000
WRITE (7. 60)	003780
LOCATION. SX, 15444IN DEC	06 2 0 0 0
•	000000
SHRIGHT, 4x, 14HCONTACL MENORY, 7x, 9HDEC OCT .3x,	010013
. TIMMEMORY LEFT RIGHT. 4x.14 HCONTROL MEMORY // 1	00 9023
00 MHILE (N.LE.1024)	00000
X = X = X	300 840
z	203650
- HAINHUN	20000
*	0.000
	200000
MAIN H MAINTEN	260000
	000000
	000010
MAITE (7, 901NN, NN, HAINHIN), HAINE, HAINR, CNTHERIN),	036000
. N. N. MAINM (NR) , MAINS, CNT MEM (NR)	000030
30 FORMAT (1H , It, 2X,04,03X,96,2X,633,2X,036,3X,016,6X,	076000
. It. 24.04.34.36.24.03.24.03.34.016	063630
- Z H Z	20000
	676060
C MAJOR CONTROL LOOP (EXECUTE MACRO):	066000
-	991000
כשרר סכיסמא	001010
C HINOR CONTROL LOOP (EXECUTE MISRO) +	00102
DO WHILE (MEMH.NE.D)	001030
HIGH HENT	20100
	00000
TE ANDIMATE ED. 21	00100
	901080
OUTPUT EXECUTION DETAILS!	00100
N . N	001100
5) MHOLD, SNIMEHIN), (MICROLI), I = 1,	
(53HGTHE FOLLOWING CY INSTRUCTION HAS JUST	0
	001140
DO WHILE CHELESTON	OCTION .
HRITE (7. 30) K. MSCPAD(M)	001170
30 FORMS (13H SCRATCH PADI . 02, 11H) CONTAINS . 06.	2H .) 001180
	001190
END B0	441740
LOTTE AND TOTAL TOTAL TOTAL MODE MADE	001210

	164. C-REG = .08./94 CARRY = .11.84. SRI = .11.84. SRZ = .001250	11.0H, 5K2 - 0012530
	11, 14,1	441251
	END IF	20100
-	INCREMENT TIME AND MICRO COUNT:	20716
	STIME = STIME + TINCR	001570
-	ITINE = IFIX(STINE)	192100
	MICS = MICS + 1	001290
1	1F (NEUN (3) - EQ. 2)	001100
	2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	101310
-	WRITE (7. 48) STIME, ITCHE, MICS	001320
	FORMAT (1MG, F15.8, 5x, 18, 5x, 15)	001330
-	# 1 ST	001343
	CHESK FOR MICRO-COUNT TERMINATION!	001350
	10 takunt91.Eq. 61	001363
		001370
-	IF tHICS.6E. NRUN(101)	001380
	THEN	060100
1	THE RESERVE TO SERVE THE PROPERTY OF THE PROPE	901100
	FINIS = .TRUE.	014100
1	£kg If	034100
	END IF	001433
-	END DO	001460
	INCREMENT MACRO COUNTS	001450
-		001400
	CALL FINAL	001410
-	END 30	001480
H	IF (NRUN(+).EQ.1)	001490
-	14EN	2000
	PRINT MEMORY CONTENTS (POST-EXECUTION)	001510
-		025700
	WRITE (7.86)	00100
1	00 WHILE (N. LE. 1054)	001500
	7 - X = XX	991560
-		001570
		901580
	LATING - AND CARTE - COTAS	001230
	MALIN - AND CHARACTER -	991600
		001610
	MATHS = AND (MAIN, LFTHS)	301620
	(7. 9.1)	001630
-	A. N. MAINTENS, AMINS, CATHERINE	901640
	N = N + 2	001650
1	1	991669
u	END IF	001670
1 4	CALL STATS	96169
	STOP	001630
		201700

001710	05/1004	FROM RUN CONTROL CARD	001790	001000 018100	1024)	, MSR2 ,	TINSTA . MOPNO . 001650	001890	001910	926190	0011930	001950	996100	001970	061700	002010	002030	005040	000000	90500	062000	002113	002130	002150	062170	962196	061200	002210	002230
SUBROUTINE INIT	02.1700 02.1700 03.1700	ONE TIME INITIALIZATION OF ALL RESISTER AND DATA PATHS INCLUDING SETTING OF PC AND M REGISTERS FROM RUN CONTROL CARD	********************************	FINIS , ERROR	COMMON CNIMENCIACAS . LONG	ADTION	MRUN(26), FINIS , ERROR	 ICOUNT = 3	+	10001 = 0	ITELM = 0	A - 0	KUA = J	11033 - C	MICQ = 0	15R1 = 0	HSCP4D(1) - NRUN(5)	00 WHILE (N.LE.16)	MSCPAO(N) = 6	00 000	DO WILLE (No.LE.5)		END 30 MICDA(1) = 8	HICO4(2) = 0	FINIS - FALSE.		STINE = 0.6	IETIM = 0	KETURN

Common	SUBROUTINE DCODC	002250
0022290 0022390 002339 002349 002339 002339 002339 002339 002339 002339 002339 002349 002339 002349 0023	092700	*** 00 22 60
35. 0 0 2 3 40 0	DECODE CONTROL MEMORY INSTRUCTION POINTED TO BY H-BEG.	*032276 *0322AD
002300 002340	SEPARATES ALL FIELDS INTO INDIVIDUAL MORDS IN THE VECTOR	•002250
FINIS : ERROR CONTENTIALS : ERROR CONTENTIAL	MICRO(25). H-REG IS RESET BY FINA .: SONTENTS OF MICRO(5).	•002300
FINES ERROR ICA ICOUNT IDIA		*** 03 2320
Time	FINIS	002330
IDOUT IDOU	CNTHERCIOSES . ICN . ICOUNT .	002350
MENNICED: FILES: MERGE MINSTE MOPINO 002390 MENNICED: MENNICED: MENNICED: MINSTE MOPINO 002420 MENNICED: MENNICED: MENNICED: MINSTE MOPINO 002420 MENNICED: MENN	SCR , ITELM , MAINHILDS	002360
MEDNICES: MEDIA 1 1973 1 1975	MENG . MENA . MICAULED HICOMICS	005370
	STATE COOPS STATES WORKS	002380
KAD KAD KAD 10L03 0024200 002420 002420 002420 002420 002420 002420 0024200 002420 002420 002420 002420 002420 002420 0024200 002420 002420 002420 002420 002420 002420 0024200 002420	HONT STIME . LTIME .	002400
TCH (IFFR, 0.5 5, MOP, IFLAS) ON CODE BITS Z AND I STORE) INVERTED, SO FIX: 002460 ON CODE BITS Z AND I STORE) INVERTED, SO FIX: 002460 IFLAST (IFFR, 5. 5, MOP, IFLAS) ON (IERR, IERR, IERR	N KAD . KDA .	002410
The color of the	HOKE	005450
The control of the	IFR = 0	002440
IF A STORES INVERTED, SO FIX: 000240 00240 00240 00240 00240 00240 00240 00240 00240 00240 00240 00240 00250 002		992458
		002460
TCH (IPTR, 5, 3, MOP, IPLAS) 002500	8	002470
	-010. 4. 4. ave.	200
OR LIERR, FELAG) 17		005200
TOW IFPR, 0, 4, MOP, IFLAS) 00.220 00.22	R LIERR. IFLAST	002510
	GETCH (IPTR. 8	002520
		002540
1	CALL GETCH (1978, 12, 4, 40P, 1FL46)	002550
OKIERA 16. 10. HOP, IFLAJI 002590 17. HOP OKIERA 16. 10. HOP, IFLAJI 002590 OKIERA 16. 10. HOP, IFLAJI 002630 OKIERA 16. 10. HOP, IFLAGI 002630 OKIERA 16. 10. HOP, IFLAGI 002630 OKIERA 16. HOP E INCA 1	MICRO(4) = HOP	002560
TERM TERM TELMO DESCRIPTION DESCRI	GETCH (TPTR. 16. 10	002500
ORLIERR, IFLGG) ORLIGGO ORLI	0151 = HOP	905599
TERR	= ORCIERR, IFLAG)	005600
ORIEER, FELGO - IERR - IERR - IERR - IERR - IEFLEGEGISTANDS, INDX, IELRGI - OCTON (1976, MCOL, I, MOP, IFLAGI - OCTON (1) - MCOL, I, MOP, IFLAGI - MCOL, I, MOD (1) - 00250 - MCOL, I, MOD (1) - 00270 - MCOL, I, MOD (1	1251 - MOD	002620
### 15ERR 102660 102600		002630
26 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6		002640
### CEFTAGE CO. 6 MOD. 1 MOP. IFLAGI 00250	#00L = 26 TMDX = 6.	382658
CETCH (IPTE MCOL, 1, HOP, IFLAG) 002550 002550 002550 002550 002700 00	HLE HFLAS.	995678
### ##################################	(IPTR.	002690
#6= ENDX + 1	:	002700
#6FEGT## 002728 002728 002729		002710
# # # # # # # # # # # # # # # # # # #	ENO 30	002720
F (RRUN(3)*ER.2) THE FORMAT (15HITHIS INSTRATS , 04, 12H, H-FIELD = , 04, 2H , 0 002 NO LE	1	002730
THEN METE (7-5) NEMH, MIDROLD) FORMAT (15HITHIS INSTRATE , O4, 12H, M-FIELD = , O4, 2H ,) FORMAT (15HITHIS INSTRATE , O4, 12H, M-FIELD = , O4, 2H ,) RITE (7, 10F IPLEC ORBIT (43HITHIALLO CALL TO SETCH FROM DODDC, IFLAG = , IZ.1H,) 00282 RROR = , TRUE, D2284 D2284		002750
MRITE (F. 5) HENH, MISANGS) FORMAT (15HITHIS INSTRAT: 04, 12H, M-FIELD = 04, 2H ,) 02503 02501 FRITE (7, 10) IFLEG 02501 06261 06261 06261 06261 06261 06261 06261 06261	THEN	002760
NO EF RITE (7, 10) IPLEG ORRAT (*3ACINVALED CALL TO SETCH FROM DCODC, IFLAG = , IZ.114,) RROR = "TRUE" DOZAGO	55 HENH, MIDROLS) 15H1THIS INSTR AT: , 04, 12H, H-FIELD	. 04. 2H . 1
RITE 17, 101 1916 TORNIT (+3HINVALIO CALL TO SETCH FROM DCODC, IFLAG = ,I2,1H,) IRROR = (1RUE, 101)	END IF	002893
1	7, 101 19th6	
Character Control of the Control of	MARON - TRUE.	
	ENO IF	20200

59 CARDS OUTPUT.

& ERRORS FOUND IN THIS ROUTINE. 53 34RDS READ.

STAULATE ACTIVITY GENERATED BY INSTRUCTION MODIFIER, MICRO(2) 102910	002880
ACTIVITY GENERATED BY	**002690
	+002910
	*** 002930
opodu street	085340
. ICN .	002960
IDOUT . IDSCR .	0/6200
MACCOLOCION STATEM STATEMENT STATEME	002990
PENES . ERROR . ATHSTR . HOPHO	00200
HCNT .	003010
COMMON KAD , KOA , IOLUB	
DATA KHASK /1777778/	003000
DATA THIBIT/ICCOUR/	003020
TO THE POLICE OF	003060
-	003076
THE REPORT OF THE PROPERTY OF	003000
IDOUT = MSCPAD(18)	903100
OR 15 (MOJ. Eq. 1)	003113
TAEN	003120
	00 31 30
7	003150
	003160
OR IF CHUGEROSS	603170
1	003190
IOIT & SAIP INSCRIPTION - LA	993293
NAT.	003210
HSR1 # T	003230
ELSE	903540
1 - 14CH - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -	00 325 3
HSCPAD (18) = AND ((SHIFF (HSCPA) (18) ,111, KMSK)	99 3290
100JT = MSCPAD(18)	0.25.00
OR IF (HOD: EQ. 4)	32
TIEN STORE OFF OFF PROM 8	9966 99
MSCPAD(18) = SHIFT (MSCPAD(13), -1)	003310
#SCPADITED = OR (MSCPADITED INTERITE	62660
IDOUT = MSCPAD(IB)	40.5340
OR 1F (MOD. EG. 5)	
14EN 14EN 64191 0FP 811 FROM Q-RES 30ES TO USB 0º 81	00 2360
LEFT SHIFT OFF BIT FROM 8 63:S TO SRI!	00 3370
SHE .	1988 00
IF (1817.EQ.C)	993400
	003410

END F F F F F F F F F	1), KHASH R1, LSB OF SE SR1 = 1)		0.000 000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.	
MICO = AND (SHITCH CO, 1)	1), KHASH 11, LSB OF SE SR1 = 1)		00 47 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
MICHAELED MICH	1), KMASK R1, LSB OF SE SR1 = 1)		003470 003470 003490 003503	
MSCPADIES	1), KHASH R1, LSB OF SE SR1 = 1)		003480	
NSCPAD(IB) = OR (NSCPAD(IB), IBIT)	R1, LSB OF SE SR1 = 1)		00 3490	
Thour = HSCPADIIB	ELSE SR1 = 1)		00 3503	
FIGHT SHIFT OFF BIT FROM Q-REG TO SRI, LSB OF B INTO MSB OF QI 003520	SR1, LSB OF B IN			
RIGHT SHIFT OFF BIT FROM Q-REG TO SRI, LSB OF B INTO MSB OF Q1 003530 RIGHT SHIFT OFF BIT FROM Q = 1, SRI = 10; ELSE SRI = 1) 003550 IEST = AND (MSCPAD(IB), 1) 003550 LOBIT = AND (MICQ, 1) 003550 LOBIT = AND (MICQ, 1) 003570 ELSE	SR1, LSB OF B IN		3510	
RIGHT SHIFT OFF BIT FROM Q-REG TO SRI, LSB OF B INTO MSB OF Q1 003530 INTELOW-BIT FROM Q-REG TO SRI, LSB OF B INTO MSB OF Q1 003540 INTELOW-BIT FROM Q-REG TO SRI, LSB OF B INTO MSB OF Q1 003540 LOBIT = SHIFT (1817, 15) I CLOBIT = AND (HICQ, 1) I CON MSCRAD(18) = OR (HICQ, 1) I CALL REPTS I CON MSCRAD(18) = OR (HICQ, 1) I CALL REPTS I CALL	ELSE SR1 = 1)		003520	
INST = AND (HSCPAD(18), 1) 1817 = AND (HSCPAD(18), 1) 1817 = AND (HSCPAD(18), 1) 1 COST 1 COST 1 COST = AND (HSCPAD(18), 1) 1 COST 1 CO	ELSE SR1 = 1)		00 35 30	
IBIT = AND (HSCPADIE), 1) LOBIT = AND (HIGG, 1) IF (LOBIT-EQ.0) IF (HENCY = 1			003540	And the second s
LOGIT = SHIFT (1517, 15) LOGIT = AND (HICQ. 1) LOGIT = AND (HICQ. 1) LOGIT = AND (HICQ. 1) ELSE HNR1 = 1 HNR1 = 1 HNR1 = 1 HNR2 = 0 HNR2 = 0R(HICQ. 1817) ERST = 1RUE. FORM AT (26H INVALIO INSTR MODIFIER = ,04,15H, RUN HALTED.) 103710 FORM AT (1340 INSTR MOD. 1900, 1900, 1900, 1900, 1900, 190770 FORM AT (1340 INSTR MOD. = ,11,21H JJST SET DATA-OUT = ,D8, 1H,) 003750 LAIN CALL REPTE CALL REPTE END IF CALL REPTE CAL			003250	
LOBIT = AND (MICQ, 1) (03570 (03570 THEN MRI = 1 ELSE END IF CRM AT (26H INVALID INSTR MODIFIER = ,04,15H. RUN HALTED.) 103720 FORMAT (13HDINSTR MOD = ,11,21H JJST SET DATA-OUT = ,08, 1H.) 003750 I (MRUN(3)-EQ-2-OR-NRUN(3)-EQ-3) END IF CALL REPTE END IF CALL REPTE CALL REPTE CALL REPTE END IF CALL REPTE CALL REPTE END IF CONDATION OF A CONDATED ON OUT OF A CONDATED OUT			00 3563	
F (LOBIT.EQ.0)			003570	
NSRI = 1			603580	
ELSE HSR1 = 1 HSR1 = 0 HSR1 = 0 HSR1 = 0 HICQ = SHIFT (HICQ, -1) HICQ = OR(MICQ, 18IT) HICC = OR(MICQ, 18IT			003200	
END IF HIGG = SHIFT (HIGG, -1) HIGG = OR (HIGG, -1) H	the state of the s		00366	
## ## ## ## ## ## ## ## ## ## ## ## ##			003610	
END IF MICG = SHIFT (MICG1) MICG = OR(MICG. IBIT) MSCPAD(IB) = SHIFT (MSCPAD(IB), -1) MSCPAD(IB) = SKIFT (MSCPAD(IB), IHIBIT) IDOUT = MSCPAD(IB) ISE MRITE (7, 5) MOD FORM AT (26H INVALIO INSTR MODIFIER = ,04,15H. RUN HALTED.) ERROR = .TRUE. IF MRUN(3).EG.2. MRITE (7, 10) MOD. IDOUT IF MRUN(3).EG.2. OR. NRUN(3).EG.3) GALL REPT2 OU3750 IF CALL REPT2 OU3750 IF				
MICG = SHIFT (MICG, 18IT) MICG = OR(MICQ, 18IT) MICG = OR(MICQ, 18IT) MICG = OR(MICQ, 18IT) MICG = OR(MICG, 18IT) MICG = MICG = MICG MODIFIER = ,04,15H. RUN HALTED.) ERROR = .TRUE. FORM AT (26H INVALIO INSTR MODIFIER = ,04,15H. RUN HALTED.) ERROR = .TRUE. IF MICG MICG MICG MODIFIER MODIFIER			003630	
MSCPAD(IB) = SHIFT (MSCPAD(IB), -1) MSCPAD(IB) = SHIFT (MSCPAD(IB), IHIBIT) INSCPAD(IB) = OR(MSCPAD(IB), IHIBIT) INSCPAD(IB) = OR(MSCPAD(IB), IHIBIT) INSCPAD(IB) = OR(MSCPAD(IB) INSCPAD(IB) = OR(MSCPAD(IB) INSCPAD(IB) = OR(MSCPAD(IB) INSCRIPTION OR SHIPE (7, 10) MOO, IDOUT I			093640	
MSCPAD(IB) = SHIF (MSCPAD(IB), IHIBIT)			103630	
SOUTH			003600	
ELSE WRITE (7, 5) MOD FORM AT (26H INVALIO INSTR MODIFIER = ,04,15H. RUN HALTED.) ERROR = .TRUE. END IF FORMAT (13H0INSTR HOD = ,11,21H JJST SET DATA-OUT = ,D8, 1H.) 003750 If (NRUN(3).EQ.2.OR.NRUN(3).EQ.3) THEN CALL REPT2 END IF RETURN END END OU33810			993689	
#RITE (7, 5) MOD FORM AT (26H INVALIO INSTR MODIFIER = ,04,15H. RUN HALTED.) ERROR = .TRUE. END IF FORMAT (13HDINSTR MOD = ,11,21H JJST SET DATA-OUT = ,08, 1H.) 003740 IF (NRUN(3).EQ.2.OR.NRUN(3).EQ.3) I + CNRUN(3).EQ.2.OR.NRUN(3).EQ.3) CALL REPT2 END IF RETURN 0033810			003693	
ERROR = .TRUE. END IF IF (NRUN(3).EQ.21 MRITE (7, 10) MOO, IDOUT FORMAT (1340 INSTR MOD = ,11,214 JUST SET DATA-OUT = ,08, 14.) 003750 If (NRUN(3).EQ.2.OR.NRUN(3).EQ.3) THEN CALL REPT2 END IF RETURN 003810		The second secon	003700	
ERROR = .TRUE. IF (NRUN(3).EQ.21 MRITE (7, 10) MOO, IDOUT FORMAT (13HDINSTR MOD = ,11,21H JJST SET DATA-OUT = ,08, 1H.) 003750 IF (NRUN(3).EQ.2.OR.NRUN(3).EQ.3) T 1 EN CALL REPT2 END IF RETURN 003810	= .04 ,15H.			
END IF IF (NRUN(3).EQ.21 WRITE (7, 10) MOD, IDOUT FORMAT (1340INSTR MOD = ,11,214 JJST SET DATA-OUT = ,08, 14.) 003750 IF (NRUN(3).EQ.2.OR.NRUN(3).EQ.3) I + EN CALL REPT2 END IF RETURN 003810			093720	
IF (NRUN(3).EQ.21 WRITE (7, 10) MOO, IDOUT FORMAT (1340INSTR MOD = ,11,214 JJST SET DATA-OUT = ,08, 14.) 003750 IF (NRUN(3).EQ.2.OR.NRUN(3).EQ.3) T 1 EN CALL REPT2 END IF RETURN 503810		200	37.30	
HAT (13HDINSTR MOD = ,11,21H JJST SET DATA-OUT = ,08, 1H.) 003750 thrun(3).EQ.2.OR.NRUN(3).EQ.3) 003760 THEN CALL REPT? 003790 IF	100		3740	
CALL REPT2 103790 1 F 1084	DATA-OUT =	14.)	3750	
CALL REPT2 003790 IF 003810		96	3760	
CALL REPT2 003790 IF 003880			003770	
I.F.				
		00	2628	
		000	3810	
SOOK SE SHITTING STAT AT CHING SCOOLS		= ,04,15H. I DATA-OUT = ,	= ,04,15H. RUN HAL	003 003630 093640 003650 003760 003760 003770 003770 003770 003770 003770 003770 003770 003770

SUBROUTINE EXECM	00.38.20
0700	
EXECUTE CONTROL HEMORY INSTRUCTION	098200
080000	000000000000000000000000000000000000000
LOGICAL FINIS . ERROR	
M(1024) . ICN .	. IDIN .
IDOUT , IDSCR , ITELM ,	
3 MICQ . MSCPAD(16) . MSR1	ISRI , MSR2 , 003940
(201, FINIS , ERROR .	. HOPNO
MICRO(25), MCNI , STIME , ITIME	• IETIM 003950
E APPLICABLE CONTROL FUNCTIONS	BEFORE INSTR EXECUTION: 003950
IF (MSR2.EQ.1)	000700
MICRO(1) = OR (MICRO(1), 6)	004010
FORMAT (15HGOP CODE NOM # , 02, 15H DUE	TO SR2 = 1.)
END IF IOP = MICRO(1)	090400
IF (10P.6E.0.4NC.10P.LE.6)	00+010
THEN	00000
OR IF (100,65.7. AND, 100, LE, 10)	004100
THEN CALL INVRT (IOP)	004110
OR 15 (10P.SE.11.4ND.10P.LE.14)	004130
TAEN CAT CAD: T (100)	004143
OR IF (10P.GE.15.AND.10P.LE.18)	004160
THEN CALL TRING (10P)	004170
OR 17 (10P.62.19.4ND.10P.LE.22)	004190
OR IF (10P.6E.23.4NO.10P.LE.26)	004210
CALL SUM (10P)	042400
17EN 17EN 17EN 17EN 17EN 17EN 17EN 17EN	004260
CALL DIFF (10P)	00 4270
ELSE 48176 (7, 18) 109	004293
IO OP CODE: , 12, 22H	ENCOUNTERED IN EXECM. 1 004300
END IF	00 4320
CALL BINRT	04K400

ENIS : ERROR : SIT FINIS = .TRUE. WHEN TRUE . 1004400	
EVALJATE RUN TEKHINATION OPTIONS - SET FINES = .TRUE. WHEN TRUE . 1004420 LOGICAL FINES . ERROR . ICH . TOOUNT . IDIN . 004450 COMMON RAD . MEN . MEN . MEN . 10040 . 004490 NEWLEST . MEN . MEN . MEN . 004490 NEWLEST . MEN . MEN . 004490 NEWLEST . MEN . 004790 NEWLEST . MEN . 004730	2564 394603 394603 394603 394603 394603 394603
LOGIJAL FINIS , ERROR , ICN , ICOUNT , IDIN , D04450 COMMON CNTRHILE&, ITEM , MAINHILD&, D04450 MCQ , MENCA ,	56.00 00 4570
LOGISAL FINIS , ERROR COMMON CONTREMILECATI	564 004570 004570 004630 004663 004663
COMMON CNTMCHILEQUES ICH . ECOUNT, IOIN . 004450 HOUS . IDSCR . ITELN . MAINKLIGAT) . 004470 HIGG . MSCPAD(16) . MSR . MSRZ . 004490 HIGG . MSCPAD(16) . MSRZ . 004490 HIGG . MSCPAD(16) . MSRZ . 004490 HIGG . MSCPAD(16) . 004490 HIGG . MSRZ . 004490 HIGG . MSCPAD(16) . 004490	564 0 0 457 0 0 457 0 0 460 0 0 460 0 0 460
MOUNT LOSCR LTELM MAINH(1024) 004470	566 394 669 394 669 303 663 303 669 6666 6666 6666 6666 6666 6666 6666
HENCY	56570 6594 6693 6694 6693 6994 6994 6994 6994 69
MICHAEL MASCRALES MAST MAST MAST MAST MAST MAST MAST MAS	564 6637 624 6637 624 6637 634 6637 634 6637 634 6637 634 6637
COMMON KAD KAD FIRES FIR	564 394 663 394 663 624 634 634 634 634 634 634 634 634 634 664 66
NEND	564 994 693 994 693 994 693 994 693 994 699
= NRUN(9) = NRUN(10) =	200 40 40 40 40 40 40 40 40 40 40 40 40 4
######################################	25 4 6 6 3 6 6 3 6 6 9 6 6 9 6 6 9 9 6 6 9 9 6 6 9 9 6 6 9 9 6 9 9 6 9 9 6 9
THEM	564 994 994 994 994 994 994 994 994 994 9
IF (MEHM.EQ.MVAL)	00 4570 00 4570 00 4683 00 4683 00 46663
THEN FINIS = .TRUE. 394,	394663 394663 394663 3946663
END IF TRUE. 1.	994693
I = (NEWS.EQ.2)	94669
I THEN EQ. 2) 1 I THEN EQ. NVAL) 1 FEN E G. NVAL) 1 THEN END IF 1 THEN EQ. 3) 1 THEN EQ. 3 THUE. 1 THEN ED. 1 THUE. 1 THUE	994663
THENC.EQ.NVAL)	994669
FINIS = .TRUE, END IF (RND.EQ.3) If (RIME.GE.NVAL) DO4679 DO4679 IF (IIME.GE.NVAL) DO4730 IF (NEND.EQ.4) DO4739 DO4749	30+663
FINIS = .TRUE. 1. (NEND.EQ.3) 1. (ILIME.GE.NVAL) 1. (ITIME.GE.NVAL) 1. (NEND.GO.4)	633
END IF END IF FILTIME.GE.NVAL! 004630 FILTIME.GE.NVAL! 004630 END IF END IF 004730 FIND IF 004730 004743	014660
TEN (IIIME.GE.NVAL) 104609 11	00400
F (IIIME, GE, NVAL) THEN FINIS = .TRUE, END IF IF (NEND, 40, 41) THEN 004730	
FINIS = .TRUE. END IF IF (NEND.=Q.4) 17464	25072
END IF (NEND. ZQ.+) 004730	00+190
IF (NEND-EQ++) 004730	004710
100	
	994.749
11 - 11 - 11 - 11 - 11 - 11 - 11 - 11	004750
INIS = .TRUE.	004770
END IF	004783
OR I - (NEND-E4-5)	004790
IF (MCNT.GE.NVAL)	004313
THEN THE PROPERTY OF THE PROPE	60+626
FINIS = .TRUE.	00400
OR I" (NENO.EQ.6)	004850
THEN	004050
700	004870
73. 1H. 1	0000
CREDE CATACACTOR CENTRALION COOL - 124 100 C	566400
016400	004910
FNG 004930	904920

	SUBROUTINE SETHM	004940
***	026700+	026700* 096700***
	THIS ROUTINE PRESETS MAIN MEMORY TO A VALUE DESIGNATED BY THE	086700*
	RUN CONTROL CARD, INCLUDING THE ABILITY TO SET EVERY LOCATION	066500+
	TO BE A NO-OP POINTING TO ITSELF	*005000
****	02.000.0000) ****************************	
	LOGICAL FINIS . ERROR	005000
	CNTMEM(1624)	005050
1	IDOUT , IDSCR , ITELM , MAINM(1024	090500
2	HBUSS . HEMC . HEMA . LICAD(6), HICDA(3),	005070
m	, MSCPAD(16) , 4SR1 ,	005080
,	26), FINIS , ERRO? , MINSTR ,	002030
יני	STIME , ITIME ,	005100
	10108	005110
	SK /3778/	005120
	x = 1	005130
	DO WHILE (.NOT.ERROR.AND.INDX.LE.1324)	005140
	I (NRUN(2), EQ. 1)	005150
		005160
-	HAINA(INDX) = 0	005170
	UK IF INKUNIZI.EU.ZI	002150
	EMDX = INDX	002 500
	MAINM(INDX) = AND (NEWDX, MSK)	005210
	317	005220
2.	==	, I3, 1H.)
	ENOR - TRUE.	005260
-	INDX = INDX + 1	005270
	END 30	305280
	RETURN	005300

-	
111	
	1
S	
-	
-	
X	
-	
	1
U.S. ARMY MISSLE COMMAND, AMSMI-RGG (MISE)	the same of the sa
S	
U.S.	1
н	
7	
-	
S	
3-	
_	
•	
-	
•	
0	
-	
~	
Ø	
-	
2	
Σ	
0	
_	
C	
w	П
. 1	
S	
S	
-	
-	
Σ	
100	
>	
7	
-	
Œ	
-	Carried and the contract of th
d	
2	
•	
S	
-	
•	
7	
-	
S	
Š	
3	
A MS	
SMAS	
SA MS	
GRAMS	*
JG ZA MS	
JGRAMS	
RUGRAMS	
PROGRAMS	
PROGRAMS	
PROGRAMS	
V PRJGRAMS	
IV PROGRAMS	
IV PROGRAMS	The second secon
IV PROGRAMS	The second secon
IV PROGRAMS	The second secon
N IV PROGRAMS	The second secon
AN IV PROGRAMS	
RAN IV PROGRAMS	
TRAN IV PROGRAMS	The second secon
TRAN IV PROGRAMS	the same of the sa
RTRAN IV PROGRAMS	the same of the sa
DRIRAN IV PROGRAMS	the same of the sa
ORTRAN IV PROGRAMS	the same of the sa
FORTRAN IV PROGRAMS	The second secon
FORTRAN IV PROGRAMS	the same of the sa
FORTRAN IV PROGRAMS	the same of the sa
D FORTRAN IV PROGRAMS	the same of the sa
ED FORTRAN IV PROGRAMS	the same of the sa
RED FORTRAN IV PROGRAMS	
RED FORTRAN IV PROGRAMS	
URED FORTRAN IV PROGRAMS	The second secon
TURED FORTRAN IV PROGRAMS	
TURED FORTRAN IV PROGRAMS	The second secon
CTURED FORTRAN IV PROGRAMS	The second secon
JCTURED FORTRAN IV PROGRAMS	The second secon
TUCTURED FORTRAN IV PROGRAMS	A CONTRACTOR OF THE PROPERTY O
RUCTURED FORTRAN IV PROGRAMS	The second secon
TRUCTURED FORTRAN IV PROGRAMS	The second secon
STRUCTURED FORTRAN IV PROGRAMS	The second secon
STRUCTURED FORTRAN IV PROGRAMS	The second secon
STRUCTURED FORTRAN IV PROGRAMS	The second section of the second seco
STRUCTURED FORTRAN IV PROGRAMS	The second secon
R STRUCTURED FORTRAN IV PROGRAMS	The second secon
OR STRUCTURED FORTRAN IV PROGRAMS	
FOR STRUCTURED FORTRAN IV PROGRAMS	The second secon
FOR STRUCTURED FORTRAN IV PROGRAMS	
FOR STRUCTURED FORTRAN IV PROGRAMS	The second secon
R FOR STRUCTURED FORTRAN IV PROGRAMS	The second secon
R FOR STRUCTURED FORTRAN IV PROGRAMS	The second secon
OR FOR STRUCTURED FORTRAN IV PROGRAMS	The state of the s
SOR FOR STRUCTURED FORTRAN IV PROGRAMS	The state of the s
SOR	The second secon
SSOR FOR STRUCTURED FORTRAN IV PROGRAMS	The second secon
SSOR	The second secon
SOR	The second secon
SSOR	The second secon
SSOR	The second secon
SSOR	
SSOR	
SSOR	The second secon
SSOR	The second secon
SSOR	The second secon
SSOR	The second secon
SSOR	The second secon
SSOR	The second secon
SSOR	

STR 1.2

Name	w)	SUBROUTINE LOADH	00	005710
CONTINUOUS MEMORY THANS TO CONTINUE TO CONTINUE THANS TO CONTINUE THANS TO CONTINUE THANS TO CONTINUE THANS THANS THAN THANS THAN THAN THAN THAN THAN THAN THAN THAN				57.30
CONTINUED TO THE TOTAL NEINARY THREES THRUTHILL TOTAL TOT			00+	5740
COURTIONUS NEARCH THRES INVALLETED COURTIONUS NEARCH THRES INVALLIT TIT COLS 3-12 = ADMESS OFFIL COLS 2-3 = ADMESS IN CONTROL NEARCH CONTROL NEARCH THRES INVALLIT TIT MICRO-FEELD THRESE INVALLIT = 21 COLS 15-12 = ADMESS IN CONTROL NEARCH CONTROL NEARCH THRES INVALLIT TO THE COLS 2-13 = BINARY CONTROL NEARCH THRES IN CONTROL THRES IN CONT	,	COL 1 = 1,	•	5750
COURTEDUDUS NERROR TREADS TRUNKTITH TITE COLS 2-12 = ADDRESS, OCTAL COLS 2-12 = ADDRESS, OCTAL COLS 2-12 = TASTROTTOR TO CONTENTS - OCTAL COLS 2-12 = TASTROTTOR TO CONTENTS - OCTAL COLS 1-2 = TASTROTTOR TO CONTENTS - OCTAL COLS 1-2 = TASTROTTOR TO CONTENTS - OCTAL COLS 1-2 = TASTROTTOR TO COTAL) COLS 2-2 = REXTH ADDRESS TOOTAL CONTON TOOTAL CONTON TOOTAL TOOT TOOTAL READ (12 = 10 TOOTAL CONTON TOOTAL TOOT TOOTAL TOOTAL TOOT TOOTAL TOOTAL	-	1 = 2.	00	5760
COURTINGUES READOR THREES INVOILLE IN THE CONTROL NOTAL CONTROLOGY STATE CONTROL HENDRY CONTROL TO THE COLS 12-12 - ADDRESS, OCTAL COLS 21-35 = CONTROL HENDRY CONTRITS - OCTAL COLS 21-35 = ADDRESS, OCTAL		COL 1 = 9. LOAD COMPLETED		0116
COLS 13-23 = MAINENSE DUTAL COLS 13-23 = MAINENSE DUTAL COLS 21-25 = MONESS, COTAL COLS 21-25 = MONESS, COTAL COLS 31-25 = MONESS, COTAL COLS 32-25 = MONESS, COTAL CONTON MONESO, M	T	CONTIGUOUS MEMORY IMAGES INRUNITITY = 111		27.50
COLS 21-55 = CONTRICT HENORY CONTENTS - D27AL COLS 21-55 = CONTRICT HENORY CONTENTS - D27AL COLS 31-56 = A DORESS, OCTAL COLS 31-57 = A DORESS, OCTAL COLS 31-57 = A DORESS, OCTAL COLS 21-57 = A DORESS, OCTAL COLS 21-57 = A OPERAND COCTAL COLS 22-57 = RENARY CONTENT FOR CONTENTS COLS 22-57 = RENARY CONTROL FIELDS PLUS A/O HUX ADDR COLS 22-57 = RENARY CONTROL FIELDS PLUS A/O HUX ADDR COLS 22-57 = RENARY CONTROL FIELDS PLUS A/O HUX ADDR COLS 22-57 = RENARY CONTROL FIELDS PLUS A/O HUX ADDR COLS 22-57 = RENARY CONTROL FIELDS PLUS A/O HUX ADDR CONTROL CONTROL FIELDS PLUS A/O HUX ADDR CONTROL CONTROL FIELDS PLUS A/O HUX ADDR MUCHAL FINIS FEROR CONTROL CONTROL FIELDS PLUS A/O HUX ADDR MUCHAL FINIS FEROR CONTROL CONTROL FIELDS PLUS A/O HUX ADDR MUCHAL FINIS FEROR M		COLS 5-12 = ADDRESS+ OCIAL		5880
ACCES 1-6 ACCES		21-35		5810
COLS 26 = ADDRESS, OCTAL COLS 21-5 = ADDRESS, OCTAL COLS 21-2 = BYSTROITON TOO FORTAL COLS 21-2 = BYSTROITON TOO FORTAL COLS 22-2 = BENARY CONTROL FIELDS PLUS ALD MUX ADDR COLS 22-2 = BENARY CONTROL FIELDS PLUS ALD MUX ADDR COLS 26-3 = BENARY CONTROL FIELDS PLUS ALD MUX ADDR COLS 26-3 = BENARY CONTROL FIELDS PLUS ALD MUX ADDR COLS 26-3 = BENARY CONTROL FIELDS PLUS ALD MUX ADDR COMMON CONTROL FIELDS PLUS ALD MUX ADDR MOUTH : 105CR : ITEL	-	T-FTFL	0.	5820
COLS 15-15 * INSTRUCTION HODFELR (COTAL) COLS 21 * INSTRUCTION HODFELR (COTAL) COLS 22-23 * NEXT H ADDRESS (COTAL) ***********************************		OLS 3-6 = ADDRESS, OCTAL	00.	5830
COURS 15-19 = A DPERAND (OCTAL) COUS 22-25 = NEXT H ADDRESS (OCTAL) COURS 22-25 = NEXT H ADDRESS (OCTAL) COURS 22-25 = NEXT H ADDRESS (OCTAL) COMPAN HOUSE (CONTROL FILES) COMPAN HOUSE (CONTROL FILES) COMPAN HOUSE (CONTROL FILES) MUCRO (25) MCD (100	-	3015 15-15 = INSTRUCTION (OP CODE - OCTAL)	00.	5843
COURS 24-21 = 8 OPERAND (OCTAL) COUS 26-33 = BINAY CONTROL FIELDS PLUS A/O MUX ADOR COUS 26-33 = BINAY CONTROL FIELDS PLUS A/O MUX ADOR COUS 26-33 = BINAY CONTROL FIELDS PLUS A/O MUX ADOR LOCICAL FINIS	_			5850
COUS 22-25 = NEXT H ADDRESS (OCTAL) COUS 22-25 = NEXT H ADDRESS (OCTAL) COUS 22-25 = NEXT H ADDRESS (OCTAL) COUS 26-33 = BINARY CONTROL FIELDS PLUS A/O MUX ADDR CONTROL CATTER LOSS (OCTAL) LOGICAL CATTER LOSS (OCTAL) LOGICAL CATTER LOSS (OCTAL) NICO COMMON LOSS (OCTAL) NICO COMON LOSS (OCTAL) N	T	JOLS 18-19 = A OPERAND (OCTAL)	0	5860
COMENS SENSE HENCY CONTROL FIELDS PLUS A/D HUX ADDR "005890 COUS 26-33 = BINARY CONTROL FIELDS PLUS A/D HUX ADDR "005890 CONTROL CHINETIES." ERROR "ICH "ITELM" ITELM" 105990 COMMON CHINETIES." ITELM" HENCH "ITELM" 105990 COMMON CHINETIES." ERROR "ICH "ITELM" 105990 MICHOLIS "HENCY "ERROR" 100590 MICHOLIS "HENCY "HENT" 100590 MICHOLIS "HENCY "HOLD "HENT" 100590 MAINTHANDRAID "HENCY "HOLD "HENDER "HORDS OF HAIN AND FELSE THEN "HELE "TO METE "HENCY "HOLD "HENDER LOBDED" "THEN "HENCY "HORD "HENDER "HENDER "HENDER" "THEN "HENCY "HENDER" "HENDER" "THEN "HENCY "HORD "HENDER" "THEN "HENCY "HORD "HENDER" "THEN "HENCY "HENDER" "THEN "HENCY "HENDER" "THEN "HENCY "HENDER" "HENDER" "THEN "HENCY "HENDER" "HENDER" "THEN "HENCY "HENDER" "HENDER" "THEN "HENCY "HENDER" "HENCY "HENDER" "HENDER" "HENDER" "HENDER" "HENDER" "HENCY "HENDER			0	2000
COMMON CATHER CONTRICT CONT		COLS 25-25 = NEXT H ADDRESS (UCTAL)		2890
LOGICAL FINIS ERROR LCN LCOUNT LDIN D05920			00	5900
LOCICAL FINIS . ERROR . ITELN . HAINHIAZ+) . DUSSON COMMON CONTENTION . DUSSON . ITELN . HAINHIAZ+) . DUSSON . DUSSON . DUSSON . DUSSON . METCON .			c c	5920
COMMON CNTMENTINGS, INCN 1 ICOUNT, 101N 1 005940 1		FINIS .	36	5930
I		CNTMFH (1024) . ICN .	. NIGI .	2940
MEUNICEPT FINES MEND MEND MEDA MED	-	IDDUT , IDSCR , ITELM ,	•	0565
MENORAL MESTA MESTA MESTA MESTA MESTA MENORAL MENORAL MESTA MESTA MENORAL MESTA MENORAL MESTA ME	-	. HEHC , HEH+ .		5960
**************************************	•	. (91)0	, MSR2 ,	5970
MCROACESSANCHI SITAE LITTE LELTH UD6600 DIMENSION CNTFLO(6)		. ERRDA .	. HOPNO	2300
DIENNION CATELLO(6) OATH HSKI /IVOGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG	S	MICRO(25), MCNT . STIME .		0660
DATA HSKI / TYPOCOCCOCOUDDS/ KEY = 0 KEY = 0 KEY = 0 L = 3 L = 3 IF (HRUN(11).EQ.1) THEN		KOA .		6010
KEY = 0 KEY = 0 L = 1 If (NRUN(11).EQ.1) DANTE (KEY.NE.3) OR HE (KEY.EQ.1) THEN THE		The way on the case of the cas		9899
		JAIN HONE / Leducacacacacacacacacacacacacacacacacacaca		6030
			96	6640
THEN			0	6050
THEN READ (5, 10) KEY, MADR, MAIN, CNTRL SEAD (5, 10) KEY, MADR, MAIN, CNTRL FORMIT (111, 74, 04, 24, 05, 016) If (KEY, EQ.1) THEN MAINHADRAL) ALINHADRAL) OR IF (KEY, EQ.2) THEN BITS 2, 3 OF OP CODE (0-+): CHIPHETHADRAL) = NOR (MSKIT CNTRL) CHIPHETHADRAL) = XOR (MSKIT CNTRL) CHIPHETHADRAL) OR IF (KEY, EQ.2) THEN KETT (2, 23) K, L OR IF (KEY, EQ.9) THEN KETT (2, 23) K, L COR IF (KEY, EQ.9) THEN ELSE ELSE			D	6064
######################################		THEN		006070
REFO (5, 10) KEY, MAUK, MAIN, CHIKL FORHAT (112, 74, 04, 24, 65, 016) THEN MAINM(HADRAL) = HAIN N	1			50000
F (KEY.EQ.1)				000000
IT (KEYELGAI) HAINM(MADRAI) = MAIN K K L OR IF (KEYELGAI) INVERT BITS 2,3 OF 00 CODE (0-+): C L L L OR IF KEYELGAI THEN WRITE (7, 23) K,L FORMAT (2-MUMEMORY LOAD COMPLETED : , 15, 19H WORDS OF MAIN AN) ELSE ELSE				001000
MAINHINADRAID = HAIN K = K + 1 N = K + 1 N = K + 1 N = K + 1		IF (KET . E.G. 1)		006120
OR IF (KEY-EQ.2) 106.15 10.4): THEN INVERT BITS 2.3 OF 0.0 CODE (0.4): CANTHETHADR-11 = XOR (HSKI» CNFRL) CANTHETHADR-11 = XOR (HSKI» CNFRL) L + 1		ATMM (MADRett) =		000130
OR IF (KEY-EQ.2) THEN INVERT BITS 2,3 OF 02 CODE (0-+)1 CMYMENTHADR-11 = XOR (HSKI; CNTRL) L = L + 1 OR IF (KEY-EQ.3) THEN MRITE (7, 23) K;L CRAMI (2-+)MHENORY LOAD COMPLETED 1, 15, 194 WORDS OF MAIN AND ELSE ELSE	1	1		000140
THEN INVERT BITS 2,3 OF 0° CODE (0-+)! CHIMENTHADRAL! = XOR (HSKIT GNFRL) CORRESS OF CONFROL HEHORY LOADED (1) ELSE				306150
INVERTRE BITS 2,3 OF 02 CODE (0-2): CUTHERHORP 1 = XOR (HSKIT CHTRL) L = L + 1 OR IF (KEV-EQ.9) THEN WRITE (7, 23) K,L FORMAT (2-HUMEHORY LOAD COMPLETED : , 15, 19H HORDS OF HAIN AND ELSE ELSE	-	THEN		09000
CANTETHARDRAID = XOR (MSKIT CNFRL) L = L + 1 THEN THEN THEN WEITE (7, 23) K,L FORMAT (2-MJKEHORY LOAD COMPLETED : , 15, 19M MORDS OF MAIN AND ELSE ELSE		INVERT BITS 2,3 OF 0" CODE (0-+		71977
OR IF (KEY.EQ.9) THEN REITE (7, 23) K.L REITE (2.40HENORY LOAD COMPLETED : , I5, 19H MORDS OF HAIN AND FORMAT (2.40HENORY LOAD COMPROL HEMBRY LOADED. , 17) ELSE		CNTHENTHABRATT = XOR (MSK1 + CNT	180	00190
THEN THEN THEN THEN THEN THEN THEN THEN				¢
FORMAT (2-40HEHORY LOAD COMPLETED : , 15, 19H HORDS OF HAIN AND ELSE ELSE		THEN		006210
FORMAT (2-HUMEMORY LOAD COMPRETED 1 , 15, 19H MORDS OF MAIN AND ELSE	-	elle (7, 23) Kit	the state of the s	
IST 32H 43R3S OF CONFROL MEMORY LOADED. 1771 006.		FORMAT (24HOMENORY LOAD COMPLET	TED : , I5, 19H WORDS	
		15+ 32H 49R3S	FROL MEMORY LOADED	90
		ELSE		96.360

	ERROR = . IRUE. ERROR = . IRUE.
NEW STATE	ERROR = .TRUE.
END DE FROR = .TRUE. END DO FUND DO FUND DO FORMILLI.EQ.2) ON HILE (KEY.NE.9) READ (5, 00) KEY, HADR, HAINL, HAINR, (CNTFLOIT), I = 1,5) ON HILE (KEY.EQ.1) THEN MAINHINDRYL) = OR (SHIFT (CNTFLOIT), 3), CNTFLOIC) THEN CNTRL = OR (SHIFT (CNTFLOIT), 3), CNTFLOIC) CNTRL = OR (SHIFT (CNTRL, 4), CNTFLOIC) INVERT BIS 2,3 OF 3P CODE (0,4); CNTRL = OR (SHIFT (CNTRL, 4), CNTRL) CNTR = OR (SHIFT (CNTRL, 4), CNTRL) CNTR = OR (SHIFT (CNTRL, 4),	END IF END IF END IF END DO OR IF (NRUNLII).EQ.2) OR IF (NRUNLII).EQ.2) OR IF (NRUNLII).EQ.2) OR IF (NEV.NE.9) READ 'S. 40') KEY. MADR. MAINL, MAINR, (CNTFLO(I), I = 1.6) OD 6350 OR AND 'S. 40') KEY. MADR. MAINL, MAINR, (CNTFLO(I), I = 1.6) OD 6350 IF (KEY.EQ.1) THEN MAINRY HADR. 1) = OR (SHIFT (CNTFLO(I), 3), CNTFLO(2) CNTRL = OR (SHIFT (CNTFLO(I), 3), CNTFLO(2) CNTRL = OR (SHIFT (CNTRL, 4), CNTFLO(2) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLO(3) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLO(4) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLO(6) INVERT BITS 2, 30 of 500E CNTRL = OR (SHIFT (CNTRL) OR OR OR OR OR OR
END IF END IF END IF END AND IF END AND IF (KEVANE.3) ON HILE (KEVANE.3) FORMAT (111x,04,1x,03,03,1x,02,02,04,08) FORMAT (111x,04,1x,03,03,1x,02,02,04,08) FORMAT (111x,04,1x,03,03,1x,02,02,04,08) FORMAT (111x,04,1x,03,03,1x,02,02,04,08) FORMAT (111x,04,1x,03,03,1x,02,02,04,08) FORMAT (111x,04,1x,03,03,1x,02,02,04,08) FORMAT (111x,04,1x,03,03,1x,02,02,02,04,08) FORMAT (111x,04,1x,03,03,1x,02,02,02,04,08) END IF END IF END IF KEY = 9 END IF KEY = 9 END IF KEY = 9 FORMAT (2,40) NRUM(11) FORMAT (2,40)	OR IF (NRUN(11) - EQ.2)
OR IF (NRUN(11).EQ.2) 17EN 17EN ON HILE (KEV.NE.9) READ (5, 40) KEV. HADR, HAENL, HAINR, (CNIFLOII), I = 1.6) ON SEAD (5, 40) KEV. HADR, HAENL, HAINR, (CNIFLOII), I = 1.6) FORMAT (11.1X,04.1X,03.03.1X,02.02.04.08) FORMAT (11.1X,04.1X,03.03.1X,02.02.04.08) ON SEAD (5, 40) KEV. HADR, HAENL, HAINR, (CNIFLOII), I = 1.6) ON SEAD (5, 40) KEV. HADR, HAINR, HADR, HADR (10.1) THEN KEV.EQ.2) CNIEL = 0R (SHIFT (3NIFL, 41), CNIFLOI2) CNIEL = 0R (SHIFT (3NIFL, 41), CNIFLOIS) CNIEL = 0R (SHIFT (3NIFL) CNIEL = 0R (SHIFT	OR IF (NRUN(11).EQ.2) OR IF (NRUN(11).EQ.2) READ [5, 40] FKEY, HADR, HAINL, HAINR, (CNTFLOLI), I = 1,6) READ [5, 40] FKEY, HADR, HAINL, HAINR, (CNTFLOLI), I = 1,6) FRANT (I.1.1X.04.1X.03.03.1X.02.01.02.02.04.08) IF (KEY.EQ.1) IF (KEY.EQ.1) IF (KEY.EQ.2) THEN CHTR = OR (SHIFT (CNTFLOLI), 3), CNTFLOL2) CHTR = OR (SHIFT (CNTFLOLY), CNTFLOLY) CHTR = OR (SHIFT (CNTRL, 4), CNTFLOLY) CNTR = OR (SHI
OR IF (NRUN(11).EQ.2) 17EN 1	OR IF (NRUN(11).EQ.2) 14EN READ (5, 40) KEY, MADR, MAINL, MAINR, (CNTFLOLI), I = 1,6) READ (5, 40) KEY, MADR, MAINL, MAINR, (CNTFLOLI), I = 1,6) READ (11,1x,04,1x,03,03,1x,02,01,02,02,04,00) FORMI (11,1x,04,1x,03,03,1x,02,01,02,02,04,00) IF (KEY.EQ.2) THEN R * 4 1 OR IF (KEY.EQ.2) THEN CNTRL = OR (SHIFT (SNTFLD(1), 3), CNTFLD(2) 1 CNTRL = OR (SHIFT (SNTRL, 4), CNTFLD(3) 1 CNTRL = OR (SHIFT (SNTRL, 4), CNTFLD(3) 1 CNTRL = OR (SHIFT (SNTRL, 4), CNTFLD(3) 1 CNTRL = OR (SHIFT (SNTRL, 4), CNTFLD(4) 1 CNTRL = OR (SHIFT (SNTRL, 4), CNTFLD(4) 1 CNTRL = OR (SHIFT (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL = OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL = OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL = OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL = OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL = OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL = OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL = OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL = OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL = OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL = OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL = OR (SNTRL + OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL + OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL + OR (SNTRL, 22), CNTFLD(5) 1 CNTRL = OR (SNTRL + OR (SNTRL + OR (SNTRL, 22), CNTFLD(5) 1 CNTR = OR (SNTRL + OR (SNTRL, 22), CNTFLD(5) 1 CNTR = OR (SNTRL + OR (S
14EN READ (5, 40) KEY, MADR, MAINL, MAINR, (CNTFLO(I), I = 1,6)	19EN
DOM HALE (KEV, EE, 9) READ '(5, 40) KEY, MADR, MAINL, MAINR, (CNTFLD(I), I = 1,5) READ '(5, 40) KEY, MADR, MAINL, MAINR, (CNTFLD(I), I = 1,5) FORMAT (II,1X,04,1X,03,03,1X,02,01,02,02,04,08) IF (KEY, EQ, I) THEN (KEY, EQ, I) THEN (KEY, EQ, I) CNTRL = OR (SHIFT (CNTRL, 4), CNTFLD(2) CNTRL = OR (SHIFT (CNTRL, 4), CNTFLD(1) CNTRL = OR (SHIFT (CNTRL, 2), CNTF	00 MHILE (KEV.NE.9) READ (5, 40) KEY, NADR, NAINL, HAINR, (CNTFLOIT), I = 1,6) FORMA (11,1X,04,1X,03,31,1X,02,01,02,02,04,08) IF (KEV.EQ.1) THEN K = K + 1 OR IF (KEV.EQ.2) THEN K = K + 1 OR IF (KEV.EQ.2) THEN CUTEL = OR (SHIFT (SNTFLOIT), 3), CNTFLOIS) CNTEL = OR (SHIFT (SNTR., 4), CNTFLOIS) CNTEL = OR (SHIFT (SNTR., 10), CNTFLOIS) CNTEL = OR (SHIFT
FEAD (5, 40) KEY, MADR, MAINE, MAINE, CONTFIDIT), I = 1,5)	READ (5, 40) KEY MADR, NAINL, MAINR, (CNTFLOII), I = 1,5)
FORMAI (II, IX, 04, IX, 03, 03, 1X, 02, 02, 02, 04, 08) IF (KEV-EQ.1) NEAL HAINRINGDR*1) = OR (SAIFT (HAINL, 8), HAINR) NEAR HAINRINGDR*1) = OR (SAIFT (HAINL, 8), HAINR) OR IF (KEV-EQ.2) THEN CNTRL = OR (SHIFT (CNTRL) (1), CNTFLD(2)) CNTRL = OR (SHIFT (CNTRL, 4), CNTFLD(2)) CNTRL = OR (SHIFT (CNTRL, 4), CNTFLD(4)) CNTRL = OR (SHIFT (CNTRL, 4), CNTRL, 4), CNTRL, 4), CNTRL, 4) ERROR = TRUE ERROR	FORMAT (II, IX, 04, IX, 03, 03, 1X, 02, 01, 02, 02, 04, 00) IF (KEV EQ. 1) THEN HAINH(HADR+1) = OR (SHIFT (HAINL, 8), HAINR) OR IF (KEV-EQ.2) THEN CNTRL = OR (SHIFT (2NTRL), 4), CNTFLD(2) CNTRL = OR (SHIFT (2NTRL, 4), CNTFLD(3) CNTRL = OR (SHIFT (2NTRL, 4), CNTFLD(4) CNTRL = OR (SHIFT (2NTRL, 4), CNTFLD(4) CNTRL = OR (SHIFT (2NTRL, 4), CNTFLD(6) CNTRL = OR (SHIFT (2NTRL, 22), CNTFLD(6) CNTRL = OR (SHIFT (2NTRL, 4),
TF (KEY.EQ.1)	TF (KEY.EQ.1)
THEN	THEN HAINH(HADR*I) = OR (SHIFT (MAINL, 8), HAINR) K = K + 1 OR IF (KEV-EQ.2) THEN CNTRL = OR (SHIFT (SNFLD(1), 3), CNFLD(2)) CNTRL = OR (SHIFT (SNFL, 4), CNFLD(3)) CNTRL = OR (SHIFT (SNFL) (SNFL) (SNFLD(4)) CNTRL = OR (SHIFT (SNFL) (SNFL) (SNFL) (SNFLD(4)) CNTRL = OR (SHIFT (SNFL) (SNFL) (SNFL) (SNFLD(4)) CNTRL = OR (SHIFT (SNFL) (SNFL) (SNFL) (SNFLD(4)) CNTRL = OR (SHIFT (SNFL) (SNFL) (SNFL) (SNFL) (SNFLD(4)) CNTRL = OR (SHIFT (SNFL) (SNFL) (SNFL) (SNFL) (SNFLD(4)) CNTRL = OR (SHIFT (SNFL) (SNFL
MAINH(HADR+1) = OR (SHIFT (HAINL, 6), MAINR)	HAINH(HADR+1) = OR (SAIFT (HAINL, 8), HAINR)
NET CALLE CALLED CALLE	K = K + 1
OR IF (KEY-EQ.2) THEN CNTRL = OR (SHIFT (CNTFLD(1), 3), CNTFLD(2)) CNTRL = OR (SHIFT (CNTRL, 4), CNTFLD(4)) ELSE END END END END END END END E	OR IF (KEY.EQ.2) THEN CNTRL = OR (SHIFT (CNTFLD(1); 3); CNTFLD(2); CNTRL = OR (SHIFT (CNTRL, 4); CNTFLD(4); CNTRL = OR (SHIFT (CNTRL, 4); CNTFLD(4); CNTRL = OR (SHIFT (CNTRL, 4); CNTFLD(4); CNTRL = OR (SHIFT (CNTRL, 2); CNTRL, 2); CNTRL = OR (SHIFT (CNTRL, 2); CNTRL 2); CNTR = OR (SHIFT (CNTRL, 2); CNTR = OR (SHIFT (CNTRL 2); CNTR = OR (SHIFT (CNTRL, 2); CNTR = OR (SHIFT (CNTRL 2); CNTR = OR (SHIFT (CNTRL 2);
THEN CNTRL = OR (SHIFT (CNTRLD(1), 3), CNTFLD(2)) CNTRL = OR (SHIFT (CNTRL, 4), CNTFLD(3)) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLD(4)) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLD(6)) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLD(6)) CNTRL = OR (SHIFT (CNTRL, 22), CNTFLD(6)) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLD(6)) ENTRE (T, 20) K, L ENTRE (T, 20) K, L ERROR = TRUE	THEN CNTRL = OR (SHIFT (CNTRL); 3); CNTFLD(2); CNTRL = OR (SHIFT (CNTRL, 4); CNTFLD(3); CNTRL = OR (SHIFT (CNTRL, 10); CNTFLD(5); L = L + 1 CNTRL = OR (SHIFT (CNTRL, 10); CNTFLD(5); L = L + 1 CNTRL = OR (SHIFT (CNTRL, 10); CNTFLD(5); CNTRL = OR (SHIFT (CNTRL, 10); CNTRL = OR (SHIFT (CNTRL); OOGS 90 OOGS
CNTRL = OR (SHIFT (2NTFLO(1), 3), CNTFLO(2)) CNTRL = OR (SHIFT (2NTRL, 4), CNTFLO(3)) CNTRL = OR (SHIFT (2NTRL, 4), CNTFLO(4)) CNTRL = OR (SHIFT (2NTRL, 40), CNTFLO(4)) CNTRL = OR (SHIFT (2NTRL, 22), CNTFLO(6)) INVERT BITS 2, 3 OF 2P CODE (0,4); CNTRL = OR (SHIFT (2NTRL, 22), CNTFLO(6)) INVERT BITS 2, 3 OF 2P CODE (0,4); CNTRL = OR (SHIFT (2NTRL, 22), CNTFLO(6)) CNTRL = OR (SHIFT (2NTRL, 22), CNTFLO(6)) CNTRL = OR (SHIFT (2NTRL, 22), CNTFLO(6)) END F EN	CNTRL = OR (SHIFT (3NTFLO(1), 3), CNTFLD(2)) CNTRL = OR (SHIFT (3NTRL, 4), CNTFLD(3)) CNTRL = OR (SHIFT (3NTRL, 4), CNTFLD(4)) LNTREH BITS 2, 3 OF 3P CODE (0,4) 1 CNTHEH (MADR*I) = XOR (MSKI, CNTRL) L = L + 1 3R IF (KEY-EQ.9) THEN WRITE (7, 20) K, L ERROR = .TRUE. END OR
CNTRL = OR (SHIFT (CNTRL, 41, CNTFLO(41)) CNTRL = OR (SHIFT (CNTRL, 41, CNTFLO(41)) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLO(5)) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLO(5)) CNTRENT BITS 2, 3 OF 3P CODE (0.4-1); CNTMENTANDR-1) = X32 (MSKI, CNTRL) L = L + 1 L = L + 1 CNTMENTANDR-1) = X32 (MSKI, CNTRL) CNTMENTANDR-1) = X32 (MSKI, CNTRL) ERROR = .TRUE. END DO END E END DO END E END CNTMENTE (7, 50) NRUN(11) = ,34, 18H, RUN TERMINATED.) 006630 ERROR = .TRUE. ERROR = .TRUE. Ondeso ERROR = .TRUE. Ondeso ERROR = .TRUE. Ondeso ERROR = .TRUE. Ondeso Ondeso Ondeso Ondeso Ondeso Ondeso ERROR = .TRUE. Ondeso Ondeso Ondeso ERROR = .TRUE. Ondeso Ondeso ERROR = .TRUE. Ondeso Ondeso ERROR = .TRUE. Ondeso ERROR = .TRUE. Ondeso Ondeso Ondeso ERROR = .TRUE. Ondeso Ondeso ERROR = .TRUE. Ondeso Ondeso Ondeso Ondeso Ondeso ERROR = .TRUE. Ondeso	CNTRL = OR (SHIFT (SWIRL, 4), CNTFLD(4)) CNTRL = OR (SHIFT (SWIRL, 4), CNTFLD(4)) CNTRL = OR (SHIFT (SWIRL, 10), CNTFLD(5)) CNTRL = OR (SHIFT (SWIRL, 10), CNTFLD(6)) INVERT BITS 2,3 OF 3P GODE (0,4); CNTMEMIMADR41) = XOR (MSKI, CNTRL) CNTMEMIMADR41) = XOR (MSKI, CNTRL) CNTMEMIMADR41) = XOR (MSKI, CNTRL) CNTRL = OR (SHIFT (SWIFT) CWIRL) CNTRL = OR (SWIFT (SWIFT) CWIRL) CNTRL = OR (SWIFT) CWIRL =
CNTRL = OR (SHIFT (CNTRL, 4), CNTFLD(4)) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLD(5)) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLD(5)) CNTRL = OR (SHIFT (CNTRL, 10), CNTFLD(5)) INVERT BITS 2,3 OF 3P CODE (0,4); CNTRHHHMOR*1) = XOR (MSKI, CNTRL) L = L + 1 L = L + 1 L = L + 1 L = L + 1 CNTRL (7, 20) K, L ELSE END IF ERROR = .TRUE. END IF ERROR = .TRUE. END IF ERROR = .TRUE. O06630 ERROR = .TRUE. O06630 ERROR = .TRUE. O06630 ERROR = .TRUE. O06650 END IF ERROR = .TRUE. O06650 END IF ERROR = .TRUE. O06650 END IF ERROR = .TRUE. O06650 ERROR = .TRUE. O06650 END IF	CNTRL = OR (SHIFT (CNTRL, 4), CNTFLD(4)) CNTRL = OR (SHIFT (SNTRL, 22), CNTFLD(5)) CNTRL = OR (SHIFT (SNTRL, 22), CNTFLD(6)) INVERT BITS 2,3 OF 3P CODE (0,+): CNTMEHTMAOR+1) = XOR (MSK1, CNTRL) L = L + 1 CNTMEHTMAOR+1) = XOR (MSK1, CNTRL) L = L + 1 One520 THEN WRITE (7, 20) K, L ERROR = .TRUE. END IF END IF END O END O ELSE WRITE (7, 50) NRUN(11) = .34, 18H, RUN TERMINATED.) 006630 FORMAT (2,HDINVALID NRUN(11) = .34, 18H, RUN TERMINATED.) 006630
CNTRL = OR (SHIFT (2NTRL, 10), CNTFLD(5)) CNTRL = OR (SHIFT (2NTRL, 22), CNTFLD(6)) INVERT BITS 2,3 OF 3P CODE (0,4); CNTMEHTMADR+1) = XO3 (HSK1, CNTRL) L = L + 1 L = L + 1 L = L + 1 L = L + 1 CNTMEHTMADR+1) = XO3 (HSK1, CNTRL) NRITE (7, 20) K, L ELSE END F ERROR = .TRUE. END F END	CNTRL = OR (SHIFT (SNTRL, 10), CNTFLD(5)) CNTRL = OR (SHIFT (SNTRL, 22), CNTFLD(6)) INVERT BITS 2, 3 OF 3P CODE (0,4); CNTMEH(MADR+1) = XOR (MSK1, CNTRL) L = L + 1 3R IF (KEV.EQ.9) THEN RRITE (7, 20) K, L ELSE RRY = 9 ERROR = .TRUE. END JF EN
CNTRL = OR (SHIFT (3MTRL, 22), CNTFLD(6)) INVERT BITS 2,3 OF 3P CODE (0,4): CNTMEH(HAOR+1) = XOR (HSKI, CNTRL) L = L + 1 3R IF (KEY=EQ.9) THEN RRITE (7, 20) K, L END DO ERROR = .TRUE. END DO END DO END DO END DO END DO ERROR = .TRUE. O06630 ERROR = .TRUE. O06650	CNTRL = 0R (SHIFT (SMTRL) 22); CNTFLD(6)) INVERT BITS 2,3 0F 3P (SODE (0,+)); CNTMEH(MAOR+1) = x02 (MSK1, CNTRL) L = L + 1 L = L + 1 L = L + 1 One 520 THEN RETE (7, 20) K, L ELSE END IF END IF END OF ELSE END OF END OF END OF END OF One 530 One 530 ELSE END OF END OF ONE 11 (2, HOINVALID ARUN(11) = .34. 18H. RUN TERMINATED.) 006 530
INVERT BITS 2,3 OF 57 CODE (0.4) 15 CNTMENTED (0.4) 15 CNTMENT (0.4) 17 CNTMENTED (0.4) 17 CNTMENTED (0.4) 17 CNTMENTED (0.4) 18 CNTMENT (0.4) 18 CNTMENTED (0.	INVERT BITS 2,3 OF 3P CODE (0,+): CNTMENTHADR+1) = XOR (MSKI, CNTRL) L = L + 1 L = L + 1 L = L + 1 L = L + 1 One 520 THEN WRITE (7, 33) KEY KEY = 9 ERROR = .TRUE. END IF END OF One 630 One 630 One 630
CNTHEN HANDRALLS 2,3 OF 37 CUCE 10;17;1. L = L + 1 L = L + 1 L = L + 1 L = L + 1 ONS (MSKI, CNTRL) L = L + 1 ONS (MSKI, CNTRL) L = L + 1 ONS (MSKI, CNTRL)	CONTHEMNABLE CONTRACTOR CONTRACTO
L = L + 1 L = L + 1 One case of the control of the case of the c	L = L + 1 L = L + 1 Object
END IF END OF END OF	DR IF (KEY-EQ.9) THEN WRITE (7, 20) K, L ELSE WRITE (7, 30) KEY KEY = 9 ERROR = .TRUE. END IF END DO ELSE WRITE (7, 50) NRUN(11) = .04. 18H. RUN TERMINATED.) 006530
END IF RETTE (7, 50) NRUN(11) = ,04, 18H, RUN TERMINATED,) 006590 END IF END SEND SEND SEND SEND SEND SEND SEND S	ELSE ELSE WRITE (7, 20) K, L ELSE WRITE (7, 30) KEY END DO END DO END DO ELSE END DO END DO END DO ECSE WRITE (7, 50) NRUN(11) = .04. 18H. RUN TERMINATED.) 006630
ELSE END IF END IF END DO ELSE END TE END TE MRITE (7, 50) NRUN(11) FORMAT (2, HDINVALIO NRUN(11) = ,04, 18H, RUN TERMINATED,) 006650 END TE END IF END IF	ELSE ELSE ERROR = .TRUE. END IF END OF ELSE END OF END OF ELSE END OF ELSE END OF ELSE END OF ELSE FORMAT (2, HOINVALID NRUN(11) = .04. 18H. RUN TERMINATED.) 006630
ELSE END IF END OF ELSE END DO ELSE MRITE (7, 50) NRUN(11) FORMAT (2, HDINVALIO NRUN(11) = ,04, 18H. RUN TERMINATED.) 006650 END OF END IF	ELSE WRITE (7, 3) KEY KEY = 9
ELSE WRITE (7, 3) KEY KEY = 9 KEY = 9 END IF END IF END DO E.SE WRITE (7, 50) NRUN(11) = ,04, 18H. RUN TERMINATED.) 006630 END IF E	ELSE KEY = 9
ERROR = .TRUE. END IF END DO E_SE FORMAT (2, HDINVALID NRUN(11) = ,34, 18H. RUN TERMINATED.) 006530 ED650 END IF ERROR = .TRUE. 006530 END IF END IF	WRITE (7, 3) KEY KEY = 9 KEY = 9 ERROR = .TRUE. END IF END IF ELSE WRITE (7, 50) NRUN(11) = .34. 18H. RUN TERMINATED.) 096630
KEY = 9 ERROR = .TRUE. END DO E_SE MRITE (7, 50) NRUN(11) FORMAT (2, HDINVALID NRUN(11) = ,04, 18H. RUN TERMINATED.) 006630 END IF RETURN	KEY = 9 ERROR = .TRUE. END IF END DO E.SE WRITE (7, 50) NRUN(11) = .34. 18H. RUN TERMINATED.) 086630
END IF END DO ELSE WRITE (7, 50) NRUN(11) FORMAT (2, HDINVALID NRUN(11) = ,04, 18H. RUN TERMINATED.) 006630 END IF RETURN	END IF END DO E.SE WRITE (7, 50) NRUN(11) = .04. 18H. RUN TERMINATED.) 006630
END IF ELSE WRITE (7, 50) NRUN(11) FORMAT (2, HOINVALID NRUN(11) = ,04, 18H. RUN TERMINATED.) 006 END IF RETURN	END DO ELSE WRITE (7, 50) NRUN(11) FORMAT (2, HDINVALID NRUN(11) = .04, 18H, RUN TERMINATED.) 006
ELSE WRITE (7, 50) NRUN(11) FORMAT (2, H0INVALIO NRUN(11) = ,04, 18H. RUN TERMINATED.) ERRORTRUE. END IF RETURN	ELSE WRITE (7, 50) NRUN(11) FORMAT (2, HOINVALID NRUN(11) = .34. 18H. RUN TERHINATED.)
ELSE WRITE (7, 50) NRUN(11) FORMAT (2, HDINVALIO NRUN(11) = ,04, 18H. RUN TERMINATED.) ERRORTRUE. END IF RETURN	ELSE #RITE (7, 50) NRUN(11) = .04. 18H. RUN TERMINATED.)
MRITE (7, 50) NRUN(11) FORMAT (2, HDINVALID NRUN(11) = ,04, 18H. RUN TERMINATED.) ERROR = .TRUE. END IF RETURN	MRITE (7, 50) NRUN(11) FORMAT (2,H0INVALID NRUN(11) = .04, 18H, RUN TERMINATED.)
FORMAT (2, HDINVALIO NRUN(11) = ,04, 18H. RUN TERMINATED.) ERROR = .TRUE. END IF RETURN D06660	FORMAT (2.HOINVALLD NRUN(11) = .34. 18H. RUN TERMINATED.
ERROR = .TRUE. 006650 1F 006650 006650	
IF	TRUE.
N\$0	16
	NAN
	END 006670

SUBROUTIN	SUBROUTINE GETCH (IADR, IBITL, IBITN, IBVIE, IFLAG)	006630
*******	002900=================================	*** 106700
USED TO E	USED TO EXTRACT A FIELD FROM THE SIMULATED 48-BIT MICRO-	*006720
PROCE SSOR	PROCESSOR CONTROL MEMORY - PARAMETERS -	•006730
IADR -	IADR - RELATIVE CONTROL MEMORY LOCATION - 0 TO 1023	•006743
13116	ISTIL - STARTING BIT OF DESIRED FIELD - U TO 47	*006750
NITET TO	SETTION MOOD STEEL OF LA AD HISTED	+006770
151 46	TELAG - FORDS DEFINENC - ZERO, NJ : RRJS - 1.2, OR 3 - ERROR IN	*006780
44	PARAMETER 1.2. OR 3. CODE 3. BITS MRAPAROUND MORO	+006790
		*009900
*******	01-10-10-10-1	1006810
1961741	STATE	028900
NOMMON		006840
	SCR TTFLY AAINHIDED	006.850
1 ~	MENG . MENH	000000
-	. MSCPAD(16)	946878
•	(26). FINIS . ERRJR . MOPNO	006800
	STIME . LTIME .	00000
COMMON	. IOLD3	006900
TEST PARA	TEST PARAMETERS FOR VALIDITY	006910
IF (IADR.	IF (IAOR.LT.0.0R.IADR.6T.1023)	006950
MEN		666939
	IFLAG = 1	046900
STI AND	I' I I I I I I I I I I I I I I I I I I	106961
IFL	18116 3-2	806970
OR IF (IB	IF (IBITN.LT.0.0R.IBITN.GT.48)	006980
THEN		066 900
IEL	IFLAG = 3	000,000
OR IFTEIS	OR IFILIBITE + IBITNI, 6T. 45)	
1 JEN		00/020
741	1 L L 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	00700
1		997050
561	GET FIELD -	007060
FOT	ADJUST START BIT FOR 60-BIT 43201	997679
IOI	IOR = IAOR + 1	00700
181	[STBIT = 46 = IBIT	060,000
INS		19710
RIGHT	TOTTE = MRU CONTRENTORY, TASK	007120
NA NA	NMB2 = 48 - (1817t + 1817N)	697130
IBYTE	TE = SHIFT (IBYTE, -NMBR)	
AI ON3	The second secon	997150
IF (IFLAG, NE. 0)	.NE.0)	007160
1464		627209
IBYTE	TE = 0	007180
1 043		961.00

	SUBROUTINE DCODM	007220	
****	092700	****07230	
0		+007250	
	MUKE INSTRUCTION POINTED TO BY	007/00+	
	RESET PC = PC + 1. LOADH-REG FROM IST MALF, C-REG LOADED FROM 2ND MALF OF MAIN MEMORY WORD.	*007270 *007280	
*****	002200+	****007290	
		007310	
	L FINIS , ERROR	007320	
-	105CP TIEN MATURE 1020	007350	
1	A TOOM A TICE!	000.00	
'n	. MSCPAD(16) . MSR1 .	007360	
*	S , ERROR , MINSTR , MOPNO	007370	
ın	MICRO(25), MCNT ,	007380	
		007390	
		007,400	
	MEMC = MSCPADILI	007410	
	+	007:20	
	CARL GETTIN THENCY OF BY TING ITEMS!	001.00	
	VIE H Y ONTE	071200	
	NAT L	037450	
	AENH = AINSTR + IOFF	007470	
	GETHM CHEMC	007480	
-	Z	007490	
	IF (IFLAG.EQ.0)	007500	
		007510	-
	MEMC = MOPND + IOFF	007520	
	HOTTE 17 483 TELAC	005.00	
1.3	1 (43	= . I2. tH.)	
:			
-		007570	
	ELSE	007580	
	HRITE (7, 10) IFLAG	007590	
	ERROR = .TRUE.	09769	
		007610	
-	KELUKN	029/00	
		2000	

	SUBROUTINE GETMM (IAUK, IBITL, IBIIN, IBTIE, ITLAU)	200	
***************************************		*** 107660	
	AND THE RESIDENCE OF THE PARTY	+007670	
USED TO EXTRACT A FIEL		+017680	
PRUCESSOR MAIN MENORY.	PARAMETERS:	069200	
IADR - RELATIVE MEMORY LOCATION - 34SE ZERO	LOCATION - 34 SE ZERO	00//00	
	F DESIRED FIELD - 3 TO 15	01110	
•	IN FIELD	1007730	
	TODGE CONSTITUTE TODGE NAME AND CONTRACTOR	*007740	
IFLAS - ERRUR CONDITIONS - ZERO	METER THE THE METER WEND AROTHER	+307750	-
THANKID LAKE		•007760	
***************************************	0k/k00+++++++++++++++++++++++++++++++++	*********	
		907780	
LOGICAL FINIS , ERROR	KOR	96.199	
H (105	· ICN	978799	-
TOOOL TO	TOSCH , ITELM , MAINMAILE, TOREST	007820	
-	· HENT	967830	-
NEUROSCI FINIS	ERROS . MINSTR .	007840	
ATCRO1251.HCNT	. STIME . ITIME .	007.850	-
COMMON KAD . KDA	100.38	01.7860	
ARTHETERS FOR		017670	
IF (IADR-LT.0.0R. IADR. GT.1123)	61.1323)	007860	
THEN		007890	
AP 16 (1911) 1 0 00 1911 GT 15)	911 61 151	007910	
TAFM	1571151	007 92 3	
16116 = 2	ALT REPORT OF THE PROPERTY OF	00 1933	-
OR IF (IBITN-LT.G.OR. IBITN.GT.16)	BITN. GT. 16)	307940	
THEN		007953	
IFLAG = 3		107963	
OR IF ((IBITL + IBITN)-61-16	.6f.16)	007970	
THEN		00.200	-
F SE	The second secon	00000	
16146 - 0		999019	
IOR = IAOR + 1		00 8023	
-	The second secon	08 60 30	
"	SHIFT (MASK(IBITN), 1917)	00000	
TOTAL THE	- AZO (ZAINT(IOX), INVS()	00000	
NABE = 16 - (IBIIL + IBIIN)	IL * IBIIN)	629979	
CAN TE	OTTES CHICKS	000000	
ENO IF	Commercial Section (Commercial Commercial Co	069630	-
TIEN		008130	
1877£ = d	The second secon	908113	
END IF		008120	
END		008140	
	THE RESERVE THE PARTY OF THE PA	The state of the s	-

SUBROUTINE BOLN (ICP)	. 0 8 1 5 0
	008150
- 1	008180
ALU SIMULATION FOR OP CODES ZERO THRJ SIX (BOOLEAN)	09190
0728974************************************	*308210
OCTAL STATE	008220
CNTHFREIGS	0,2000
IDOUT . IDSCR . ITELM . MAINME	008250
, MEMC , MEM4 ,	008260
. (91)0	038270
S . ERROR . HINSTR .	008280
ALCKUICSION .	00.000
THE CASE OF THE CA	000 300
	000330
HOD = HICKOLET	946946
	008350
IBR = MICRO(4) + 1	069360
IOLO3 = MSCPAD(18R)	0008370
If (10P. Eq. 0)	066930
TIEN	008 390
IF CEN. Eq. 01	010400
237	108410
ITMP = LAIGH	02 hann
	64466
ITHP = 0	008420
	003460
WRITE (7, 10) ICM, IOP	20000
FORMAT (17 HO INVALID CARRY = .08,20 H IN BOLN, OP CODE	= . I2. 1H.) 008490
ENO IF	008500
11.20.01. T. NO.	00000
	008533
OR IF (10P.EQ.2)	000540
TIND - AND FIGURE MCCOAD (TABLE)	008569
OR IF (10P.EQ.3)	008570
TACK CONTRACTOR CONTRA	000590
DO TO LINE OR INSCRIPTION AND ADDITION OF THE LINE OF	96866
NAT.	008610
ITHP = OR (IDIN. MSCPAD(IBR))	009620
OR I" (103.Eq.5)	009630
ITMP = XOR (MSCPAD(IAR), MSC-AD(IBR))	003650
	06866
ITHP = XOR (IDIN, MSCPAD(18R))	699669

TJEN	900	
25 FORMAT (7HOINSTR , 02, 24H EXECUTED, TEMP VALUE = ,06, 1H.)	VALUE = ,06, 1H.) 908733	
IF (MOD.EQ.7)	508740 008750	
HICQ = ITHP	008750	
	008780	
MSCPAD(IBR) = ITMP CALL INMOD (IAR, IBR)	008800	
RETURN	008830	
** DERRORS FOUND IN THIS ROUTINE. 7:: CARDS READ.	008840 64 CARBS OUTPUT.	

FINIS . E RROW CUTHERHIDES . ERROW CUTHERHIDES . IDSCR . ITEL4 . 4AINHIDES . CUTHERHIDES . IDSCR . ITEL4 . 4AINHIDES . CUTHERHIDES . REC . MEAT . MEAZ . MUCACO . MSCPACO . MSCR	SUBROUTINE INVRT (10P)	1008850
ALU SIMULATION FOR OP CODES 7 TARJ 10 (INVERT) LOGIZAL FINIS . ERROR . ICH . CCOUNT . IOIN . 008990 COMMON CONTENTION TAND . 1056R . ITEL . 4 AINH1024) . 008990 COMMON WAD . 1056R . ITEL . 4 AINH1024) . 008990 MICHAGOT . TAND . 10109 . 018990 MICHAGOT . TAND . 10109 . 0189900 MICHAGOT . MICHAGOT . 10109 . 0189900 MICHAGOT . TAND . 10109 . 0189900 MICHAGOT . TAND . 10109 . 0189900 MICHAGOT . TAND . 10109 . 0189900 MICHAGOT . MICHAGOT . 10109 . 0189900 MICHAGOT .		- 0.0 0 8 7 0 - 0.0 0 8 7 0
LOGICAL FINIS . ERROR COMMON . ICON		068800.
LOGIZAL FINIS . ERROR LOGUINI . LOTN DOSSON COMMON CONTENTIST LOS LOTE ALNHILOZA) DOSSON COMMON CONTENTIST LOSCA LOSTA MERCA MER		0108000
COMMON CONTRETIEZ, 1 IECH 1 FOUNTY 1 DON'S 1 D	FINIS . ERROR	066900
HENDES HENDES HENDES HENDES	COMMON CNTMENTICES , ICM , ICOUNT , ION , ICOUNT , ION , ITEL , MAINWILDES	000000
COMMAN	MBUSS , MENC , NEM4 ,	008950
CONTINUED CONTINUED CONTINUED CONTINUED CONTINUED	MRUNIZOT. FINIS . ERROR . HINSTR . HOPMO	096900
DATA LAIGH (1777772 AGA DATA LAIGH (177772 AGA DATA LAIGH (17772 AGA DAT	MICRO(25), HCMT , STIME , ITIME ,	06800
F (18M1 (27 Has 20.2) NETE (7, 9)		009010
HOO = HICKOL2; HOO = HICKOL3; HOO		020500
IAR = MICROL3) * 1 104.03 = MSCPAD(18R) 105.03 = 0.00PL (101N) 106.03 = 0.00PL (101N) 107.00 = 0.00PL (101N) 108.00 = 0.00PL (101N) 109.00 = 0.00PL (101N) 109.00PL (101N) 109.00 = 0.00PL (101N) 109.00PL (101N) 109.00		043600
Intercolor Int		000000
THE COMPLEMENTARY DO DO DO DO DO DO DO D	IOLDS = MSCPAD(19R)	0.0000
THE	1f (10p.£q.7)	000600
OR IF (10P. c. c. of the component of th	-	060803
OR IF (IOP.EQ.9) OR IF (IOP.EQ.9) THEN = COMPL (MIDIN) OR IF (IOP.EQ.9) OR IF (IOP.EQ.9) OR IF (IOP.EQ.9) OR IF (IOP.EQ.1L) OR IF (IOP.EQ.1	IF (10P.	009110
OR IF (10F = COPPL (120A) THE	1-fev	609129
THEN TITP - COMPL (HSCPADITOR) 009173 TEN	OR IF (10P.Eq.9)	009140
OR IF (10P-EQ.1L) 14EN DORIGON (HICQ) END IF IIMP = COMPL (HICQ) END IF IIMP = AND (IIMP, LMIGH) OR IF (ICM, LT.G. OR, ICM, ET.) THEN ITMP = THP * LMIGH) OR IF (ICM, LT.G. OR, ICM, ET.) TEN OR IF (ICM, LT.G. OR, ICM, ET.) FORMAT (17HGINVALIO CARRY = ,35,214 IN INVRT, OP GOOE = ,12,14.) END IF FORMAT (17HGINVALIO CARRY = ,35,214 IN INVRT, OP GOOE = ,12,14.) END IF I (40D, EA.2) FORMAT (17HGINTR, O2, 24H EXECUTED, TEMP VALUE = ,06, 14.) END IF HICQ = ITMP I (40D, EA.2) I (40D, EA.2) I (40D, EA.2) END IF END IF END IF END IF END IF FORMAT (17HGINTR, O2, 24H EXECUTED, TEMP VALUE = ,06, 14.) END IF END IF FORMAT (17HGINTR, O2, 24H EXECUTED, TEMP VALUE = ,06, 14.) END IF END I	TAEN	
END IF ITHP = COMPL (MICG) END IF ITHP = LANGINP, LMIGH) OR IF (TCM.ETG.G) THEN THE P. LMIGH) OR IF (ICM.LT.G.GR.ICM.SF.1) FORMAT (ITHG.INVALIO CARRY = ,28,214 IN INVRT, OP COOE = ,12,14.) FORMAT (ITHG.INVALIO CARRY = ,28,214 IN INVRT, OP COOE = ,12,14.) FORMAT (ITHG.INVALIO CARRY = ,28,214 IN INVRT, OP COOE = ,12,14.) FORMAT (ITHG.INVALIO CARRY = ,28,214 IN INVRT, OP COOE = ,12,14.) FORMAT (ITHG.INVALIO CARRY = ,28,214 IN INVRT, OP COOE = ,12,14.) FORMAT (ITHG.INVALIO CARRY = ,28,214 IN INVRT, OP COOE = ,12,14.) FORMAT (ITHG.INFC = ,12,14) FORMAT	OD IN LIDE ED. LL.	
END IF THE E COMPL (MICQ) END IF ANO(ITMP, LMICM) OP 220 OP 2210 OF 15 TITM = ANO (ITMP, LMICM) OR IF TICK.EC.OR.ICK.EC.II THEN THE TAND (ITMP, LMICH) FORMAT (ITMO INVALIO CARRY = .38.214 IN INVRT, OP COOE = .12.14.) END IF THEN THE TANDER = .78 TOP. ITMP FORMAT (ITMO INVALIO CARRY = .38.214 IN INVRT, OP COOE = .12.14.) END IF THEN TOPER = .18.14.) FORMAT (ITMO INVALIO CARRY = .38.214 IN INVRT, OP COOE = .12.14.) FORMAT (ITMO INVALIO CARRY = .38.214 IN INVRT, OP COOE = .12.14.) FORMAT (ITMO INVERTOR INVERTOR INVERTOR INVIT.) FORMAT (ITMO INVO INV. INVERTOR INVO INV.) END IF THEN TOPORT = ITMP FORMAT (ITMO INV. INV.) END IF THEN TOPORT = ITMP FORMAT (ITMO INV. INV.) FORMAT (ITMO INV. INV.) FORMAT (ITMO INV. INV.) END INVERTOR INVO INV. INV. INV.) END INVERTOR INVO INV. INV. INV. INV. INV. INV. INV. INV.	14£N	669140
IT I I I I I I I I I	:	00 31 90
TECN-62.01 100-62.00 100-92.00 116N 114P = 114P + LMIGH	ITMP = AND(ITMP, LHIGH)	009210
ITHP = ITHP * LMIGH) OR IF (ICH. 200 (IMP. LMIGH)) TEN = AND (IMP. LMIGH) TEN = AND (IMP. LMIGH) FORMAI (I.HGINVALIO CARRY = .38.214 IN IMVRT, OP CODE = .12.14.) END IF WRONG = .FUE. FORMAI (I.HGINVALIO CARRY = .38.214 IN IMVRT, OP CODE = .12.14.) END IF WRONG : .FUE. FORMAI (I.HGIN . 22.1 LOF. ITHP	IF (ICN, EQ. O)	009220
OR If (ITH = AND (ITH), LUIGH) TERM. LITTLE ACTION (ITH) WRITE (17 101 ITH) FORMAT (IZHGIMVALIO CARRY = ,30,214 IN INVRT, OP CODE = ,12,114,) END IF FORMAT (ZHGIMVALIO CARRY = ,30,214 IN INVRT, OP CODE = ,12,114,) FORMAT (ZHGIMVALIO CARRY = ,30,214 IN INVRT, OP CODE = ,12,114,) FORMAT (ZHGIMST, 27,7) FORMAT (ZHGIMST, 24, EXECUTED, TEMP VALUE = ,06, 14,1) FORMAT (ZHGIMST, 11,00,00,00,00,00,00,00,00,00,00,00,00,0	ITHP - ITHP - LAIGH	009240
######################################	:	
# WRITE (T. 10) ICH. 100 END END	1 F	
END IF THEN IS - STUDE. If INTEN IS - CO - STATE 009310 009310 1441 14	FORMAT (17HG INVALED CARRY = ,38,214 IN INVRT, OP CODE =	
F (MRUN(3).EQ.2)	END IF	
######################################	IF (NRUN(3) . EQ. 21 THEN	009320
.E3.7) 009360 0001 = ITHP 0001 = ITHP 0001 = ITHP 0001 = ITHP 009460 009460 009460	FORMAT (THEINSTR , 02, 24H EXECUTED, TEMP VALUE =	
	END IF IF (400.E2.7)	909360 009370
SCHADIBRY = ITHP SALL INHOG (IAK, IBR) 009450	HICA = ITMP	009393
SCHIDTERF = ITMP SALL IMMOD (IAK, ISR) 009440	1000	607600
CALL INHOG (IAK, 18R) 009440	£1.5c	009410
	CALL INNOC (IAK, 18R)	6E+600
	RETURN	057600

ALU SIMULATION FOR OP CODES 11 THRU 14 (THO"S COMPLEHENT) LOGCICAL FINIS : ERROR COMMON CHIEFERIECE'S ITEM (THO"S COMPLEHENT) COMMON CHIEFERIECE'S ITEM (THORIS) HICKORIS HICKORY MICROLES HERO (THE NATIONAL HICKORY) HICKORY HICKORY MICROLES HERO (THE NATIONAL HICKORY) HICKORY HICKORY MICROLES HERO (THE NATIONAL HICKORY) FORMAT (27405UBROUTINE COMPLENTERED.) FORMAT (27405UBROUTINE COMPLENTERED.) FORMAT (27405UBROUTINE COMPLENTERED.) FORMAT (27405UBROUTINE COMPLENTERED.) FORMAT (100-Eq.13) FORMAT (100-Eq.13) FORMAT (110-Eq.13) FORMAT (110-Eq.13) FORMAT (110-Eq.14) FORMAT (110-Eq.14) FORMAT (110-Eq.14) FORMAT (110-Eq.15) FORMAT (110-Eq.15) FORMAT (110-Eq.16) FO	1	SUBROUTINE CMPLT (10P)	009470
ALU SIMULATION FOR OP CODES 11 THRU 14 (TMO"S COMPLEMENT) ************************************			057500
LOGICAL FINIS : ERROR . ICOUNT , IDIN , 009550 COHMON . CWTHEHELSER ERROR . ITELA , HAINHILIZE , 109550 COHMON . CWTHEHELSER ERROR . ITELA , HAINHILIZE , 109550 HIGGS , HENCE		LU SIMULATION FOR OP CODES 11 THRU 14 (TMO"S COMPLEHENT)	0736004
LOGICAL FINIS , ERROR , ICAN , 101N , 009550 COMMON CNTHENISCH , ITELA , MAINHIGATH , 101N , 009550 MINOR CNTHENISCH , ITELA , MAINHIGATH , 100950 MINOR CNTHENISCH , ITELA , MAINHIGATH , 100950 MINOR CNTHENISCH , 100950 MINOR CNTHEN			000000
COMMON CONTENT CONTE		FINIS , ERROR	000000
## ## ## ## ## ## ## ## ## ## ## ## ##		IDOUT , IDSCR , ITELA ,	009570
COMMON KAD FORTALL FIRE	2	S . HENC . HENN .	009580
HICROLESI, HONT STIME TIME TETM 009550	,	COI. PINIS . PRESS . HINSTR .	009600
DATA LHIGH AT77778 AND A TOWN A TOWN AND A TOWN A TOWN AND A TOWN A TOWN AND		MICRO(25), MCNT , STIME! , ITIME ,	019610
FRINKLY CANDAL	00	. AUA	009630
The content of the	1	5)	049640
TAR = MICCO(13) + 1		ENIEKED.	0.0000
IBR = HCROC41 + 1 009500 1003 = HSCPD0(18R) 1003 = HSCPD0(18R) 1003 = HSCPD0(18R) 1003 = HSCPD0(18R) 1009	E H	= MIC30(3)	009670
IOU03	1	*	009600
If (IOP.Eq.il)	1		069600
OR IF (IOP.EQ.12) TIMP = COMPL (IDIN) OR IF (IOP.EQ.12) THEN = COMPL (IDIN) OR IF (IOP.EQ.13) OR IF (IOP.EQ.14) THEN = COMPL (MICQ) END IF THEN = AND (ITMP, LMIGH) THEN = THEP * TAN THEN * THEN * THEP * TAN THEN * THEN * THEP * TAN THEN * THEN * THEP * TH	-		002600
OR IF (10P. GG.12) 14EN OR IF (10P. GG.13) OR IF (10P. GG.13) 15EN OR IF (10P. GG.13) OR IF (10P. GG.13) OR IF (10P. GG.13) ITHP = COMPL (HXCPACITSR)) OR IF (10P. GG.13) ITHP = AND (ITHP, LMIGH) OR IF (HROW 13P. GG.14) OR IF (HROW 13P. GG.17) OR IF (HROW 13P. GG.14) OR IF (HROW 13P			009713
OR IF TIOP.EQ.137 OR IF TIOP.EQ.137 THEN OR IF (IOP.EQ.14) OR IF (IOP.EQ.17) OR IF (o	IF (IOP.	069730
OR IF time conflictions The	-	-	009740
OR IF (IOP-EQ.14) OR IF (IOP-EQ.14) THEN = COMPL (MICQ) END IF ITHP = AND (ITMP, LMIGH) ITHP = AND (ITMP, LMIGH) THEN = TITHP + LMIGH) OR 9550 ITHP = AND (ITMP, LMIGH) OR 9550 OR 955	0	R IF (10P.Eq.13)	
OR IF LIDP. GG.LTP. (M.SCFACILDEX.) 146N END IF ITHP = AND (ITHP. LNICH) OD 9350 IF (NO. EQ.?) FORMAT (? MOINSTR * 02, 24H EXECUTED, TEMP VALUE = ,06, 14,) FORMAT (? MOINSTR * 02, 24H EXECUTED, TEMP VALUE = ,06, 14,) FORMAT (? MOINSTR * 02, 24H EXECUTED, TEMP VALUE = ,06, 14,) FORMAT (? MOINSTR * 02, 24H EXECUTED, TEMP VALUE = ,06, 14,) FORMAT (? MOINSTR * 18 P		-	0 // 600
ITHP = COMPL (MICQ) END IF ITHP = AND (ITMP, LMIGH) ITH = AND (ITMP, LMIGH) OP9949 ITH (NRUM (3) EQ.2) FORMIT (7 MOINSTR * 02, 24H EXECUTED, TEMP VALUE = .06, 14,) FORMIT (7 MOINSTR * 02, 24H EXECUTED, TEMP VALUE = .06, 14,) FORMIT (7 MOINSTR * 02, 24H EXECUTED, TEMP VALUE = .06, 14,) FORMIT (7 MOINSTR * 02, 24H EXECUTED, TEMP VALUE = .06, 14,) FORMIT (7 MOINSTR * 02, 24H EXECUTED, TEMP VALUE = .06, 14,) FORMIT (7 MOINSTR * 18,) FORMIT (7 MOINSTR * 18,) END IF E	0	IF (IOP	009790 069690
END IF ITHP = AND (ITHP, LHIGH) ITHR WATTE (T. 25) IOP, ITHP FORMAI (7 HOLNSIR & 02, 24H EXECUTED, TEMP VALUE = .06, 14.) FORMAI (7 HOLNSIR & 02, 24H EXECUTED, TEMP VALUE = .06, 14.) FORMAI (7 HOLNSIR & 02, 24H EXECUTED, TEMP VALUE = .06, 14.) FORMAI (7 HOLNSIR & 02, 24H EXECUTED, TEMP VALUE = .06, 14.) FORMAI (7 HOLNSIR & 02, 24H EXECUTED, TEMP VALUE = .06, 14.) FORMAI (7 HOLNSIR & 18.)		ITHE	
ITHP = AND (ITHP + 150 ITHP = AND (ITHP + 150 ITHE = AND (ITHP + 150 + 1	1	NO IF TMP = AND (ITMP, LMIGH)	009830
### TE (NRUN (3) - Eq. 2) THEN WEITE (7, 25) IOP, ITHP END IF (NO. EQ. 7) THEN IS (NO. EQ. 7) THEN IS 2 - ITHP IS 2 - ITHP IS 2 - ITHP IS 3 - ITHP IS 4 - ITHP IS 5 - ITHP IS		1	009850
THEN THEN THEN THEN THEN TO SEN AT (7 HOINSTR , 02, 24H EXECUTED, TEMP VALUE = ,06, 14,) END 1F (HOD.EQ.7) THEN HIG3 = ITMP HIG4 = ITMP HSCPADIISR1 = ITMP CALL IMHOD (IAR, IBR) ELSE END IF ELSE HSCPADIISR1 = ITMP CALL IMHOD (IAR, IBR) 009998	-	2	999699
END IF (HOD.EQ.7) 1 (HOD.EQ.7) 1 (LODI = ITHP 1 (LODI = ITH	75	RITE (7, 25) IOP, IIMP ORMAI (7HOINSIR, 02, 24H EXECUTED, TEMP VALUE =	68
MIC2 = ITHP MIC2 = ITHP I LOUT = ITHP CALL INHOO (IAR, IBR) 10990			9966
100UT = 17MP 009 0		THEN MIG2 = ITMP	609926
CALL INHOD (IAR, IBR) CALL INHOD (IAR, IBR) CALL INHOD (IAR, IBR) CALL INHOD (IAR, IBR)		S	649949
IF		CALL INNOO (IAR, 19R)	139973
	0 2	- X	966690

SUBROUTINE TRING (IDP)	010010
000010	0000000
ALU SIMULATION FOR OP CODES 15 THRJ 18 (TRANSFER/INCREMENT)	*010050
020070.	0.00000
LCGICAL FINIS . ERROR	010090
CNTHENCIO241 . ICN .	010100
. IDSCR , ITEL4 .	010110
3 MICO . NSCPAD(16) . MSR1 . MSR2 .	010120
NRUN(20), PINIS , ERROR , HINSTR , HOPING	010140
MICRO(25), MCNT	010150
CONHON KAD , KDA , IOLDS	01010
DATA LHIGH /I///E/	010170
5 FORMAT (27HDSUBROUTINE TRING ENTERED.)	01010
HOD = MICRO121	819290
IAR = MICRO(3) + 1	010210
- MICROCA)	010220
IOLO3 = MSCPAD(IBR)	010230
IF (108-52-15)	910240
THEN	010251
ITHP = MSCPAD(IAR) + ICN	010260
OR IF (IOP.EQ.16)	010270
TAEN TOTAL TOTAL	616283
101 - 1101 - 1001	067010
NAT I CLUSTER IN THE PROPERTY OF THE PROPERTY	010300
ITHE = HSCPAO(18R) + ICM	010320
EQ. 181	010330
THEN MICQ + ICN	01.0350
EWO IF	010360
ITHP = AND (ITHP, LMIGH)	010370
IF INKUNISTICES	010 300
MOTIF (7, 25) 100, 11MP	010401
25 FORMAT (THOINSTR ,02,24H EXECUTED, TEMP VALUE = ,08, 1H.)	
END IF 1F (400.F3.7)	010420
	010443
MIC2 = ITMP	013453
F. C.	018.71
HSCPAD(18R) = 11MP	019490
	010490
RETURN	010510
7	010520

SUBROUTINE DECTR (10P)	010530
055070	0501020
ALU SIMULATION FOR OP CODES 19 THRU 22 (DECREMENT/TRANSFER)	. 0105010
065070+	065010***
LOGITAL FINIS . ERROR	010610
CNTHERCIO24) , ICN .	010620
. IDSCR . ITELM .	010630
. MICAULES	010650
HICHOLOGIAN CTIME . ITIME . IETH	010650
COMMON KAD . KOA . IOLOS	010680
HIG4 /1777778/	010690
IF (NRUN(3)-EQ.2) WRITE (7, 5)	010700
במשמו באונסטינטו באונסטינטו באונים מבטוי באונים ו	910726
IAR = MICRO(3) + 1	010730
IBR = MICRO(4) + 1	010760
IOLOB = MSCPAD(18R)	010750
IF (10P.EQ.191	010770
TOTAL TANDON TO THE TANDON TO	010783
OR IF (10P-EQ-26)	010790
TIES - TOTA	010810
08 IF (108.50.21)	010820
THEN	010830
TIMP & MSCPADLIBRI	010040
OK IF (IUF-E4-CC)	01065
ITHP = MICG	010873
TE (TCM. FO. N.)	010890
TAEN	010910
ITHP = ITHP + LHIGH	11,091.0
DO TO LICENTA LAND CHIMM, CANON	010930
	010910
WRITE (7, 10) ICM, IOP	
TOWART LITHURING CHERT - 100-11 IN UCCIN, OF CODE - 114-114	010070
IF (NRUN(3), EQ.2)	91090
TAEN	066010
MRITE (7, 25) IOP, ITMP SE COMMAT (7MDINSTO . 32, 24H EXECUTED. TEMP VALUE = .06.	14.) 011013
END IF	195
IF (400.£2.7)	011030
MICO # 11MP	011050
1001 - ITHP	07.1060
6.56	011070
CALL TWOOD (TAR. TAR.)	011093
FIG 14	OTITO CONTRACTOR CONTR
RETURN	011110

ALU SIMULATION FOR OP COUES 23 THRU 26 (ADD) **********************************
ALU SIMULATION FOR DP CODES 23 THRU 26 (ADD) **********************************
CONTROL CONT
LOGGCAL FINIS , ERROR CONTENTISE, ERROR CONTENT , IDIN , 011200
The component of the control of the count of the component of the compon
IDDUIT
##
METON OF THE NEW TOOLS OF THE NEW TO
WANTERS FINES FINES FINES FINES COMMON KAD STREE ITTHE ITTHE COMMON KAD STREE ITTHE ITTHE FORMAT 1277777 17 184 FORMAT 127408MOUTINE SUM ENTERGO, FORMAT 127408MOUTINE FORMATINE 127408MOUTINE FORMAT 127408MOUTIN
DUTATION (ALCOLOS), MENT (25), MENT (25), MENT (25), MENT (25), MENT (25), MENT (25), MENT (27), ME
DITALLHICK ALTYTYTH CONTROLL OF THE CONTROLL O
F CARLOW 13.
FORMAT (ZPHOSUBROUTINE SUM ENTERED. 1 10.00
HOO = HCROCE! HOO HO
IAR = MCRO(3) + 1 18R = MCRO(4) + 1 18R = MCRO(4) + 1 18R = MCRO(4) + 1 19L 10P.Eq.23 17E 10P.Eq.23 17E 10P.Eq.25 17E 10
18R = HCRO(4) + 1 11350 1150 1150 11350 1150
IQLOB
IF (10P.EQ.23) 146N OR IF (10P.EQ.24) THN = HSCPAD(19R) + ICN OR IF (10P.EQ.24) THN = IDIN + HSCPAD(19R) + ICN OR IF (10P.EQ.25) THE HSCPAD(19R) + ICN OR IF (10P.EQ.25) THE HSCPAD(19R) + ICN OR IF (10P.EQ.25) THE HSCPAD(19R) + ICN OR IF (10P.EQ.25) THEN = ANO (ITHP, LHIGH) FORMAT (7HCINSTR, 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR, 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR, 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR, 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR, 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR, 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR, 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR, 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR, 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR, 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR, 02, 24H EXECUTED, TEMP VALUE = ,06, 14.)
14EN
OR IF (IOP.EQ.24) 146N 174N
(IOP.EQ.24) NTDP. = IDIN + MSCPAD(IBR) + ICN (IOP.EQ.25) ITMP = MSCPAD(IAR) + MICQ + ICN (IOP.EQ.26) ITMP = IDIN + MICQ + ICN AND (ITMP, LHIGH) OUI1450 UN13).EQ.27 N EQQ4 = ITMP ITMP = IDIN + MICQ + ICN 011450 011550 011550 011550 011550 011550 011550 011550 011550 011550 011550 011550 011550 011550
THEN
ITTP- = IDIN + MSCPAD(IBR) + ICN ITTP- = TAIN + MSCPAD(IBR) + ICN ITTP- = TASCPAD(IBR) + MTCQ + ICN ITTP- = IDIN + MTCQ + ICN ON (ITMP, LMIGH) ON (ITMP
THEN THEN THEN THEN THEN THEN THEN THEN
OR IF (IOP.EQ.26) 146N END IF ITHP = IDIN + MICQ + ICN END IF ITHP = IDIN + MICQ + ICN END IF ITHP = AND (ITHP, LMIGH) IF (MRNM13)-EQ.2) THEN END IF FORMAT (74CINSTR , 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) IF (MOD.EQ.7) IF (MOD.EQ.7) INCQ = ITMP END IF END IF IDOUT = ITMP END IF END IF IDOUT = ITMP END IF IDOUT = ITMP
ITAP = IDIN + HICQ + ICN
THEN END IF ITHP = IDIN + HICQ + ICN G11490 ITHPRACE THEN THEN ITHER THEN THEN THEN END IF FORMER (7 25) 10P, ITHP END IF FORMER (7 40), 24H EXECUTED, TEHP VALUE = ,06, 14.) ITHEN ITHER THEN THEN END IF END IF END IF END ITHP END ITHP END ITHP END ITHP END ITHP O11550
ITHP = IDIN + HICQ + ICN AND (ITHP, LHIGH) UN13:=64.21 UN11:=64.21 UN115:E4.25
END F AND (ITHP, LHIGH) IF (NRUN(3)-Eq.2) THEN FORMAT (7HCINSTR , 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR , 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR , 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR , 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR , 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR , 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) FORMAT (7HCINSTR , 02, 24H EXECUTED, TEMP VALUE = ,06, 14.)
I THEN THE THE THE THE THE THEN THEN THEN
IT HEN WEIFE (7, 25) 10P, ITHP 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) END IF (MOD.EQ.7) IF (MOD.EQ.7) ALCQ = ITHP E.SE E.SE
FORMAT (7HCINSTR , 02, 24H EXECUTED, TEHP VALUE = ,06, 14.) FORMAT (7HCINSTR , 02, 24H EXECUTED, TEHP VALUE = ,06, 14.) IF (H00.EQ.7) IF (H00.EQ.7) INIQ = ITHP IDOUT = ITHP E.SE
FORMAT (THEINSTR , 02, 24H EXECUTED, TEMP VALUE = ,06, 14.) LEO IF (HOD.Eq.7) THEN THEN TIMP LOGUT = ITHP E.SE 01150
END IF (HOD.EQ.7) 011540 11550
HEQ = ITMP 011550 HOUT = ITMP 01150
ICQ = ITMP DGJT = ETMP 011
00sf = EFMP
410
ANCERO LIBERT IN LIBERT
98)
SELUCAN ALLOCAN

SUBRUUITNE DIFF	(10P)	011650
***************************************	029770 ***********************************	******011670
		*011680
ALU SIMULATION FL	ALU SIMULATION FOR OP CODES 27 THEJ 31	• 111690
C	012110	012110
LOGICAL FINIS	ERROR	011730
	1624) , ICN , ICOURT , IDIN ,	011740
	SCR , ITELM ,	011750
	HEMH . MICADIS) .	011760
3 HICG	MSCPAD(16) . MSR1 ,	011770
	S . ERROR . HINSTR .	011780
COHMON KAR	ALCKUICATONIO DILATE O LITTLE O LEITH	011/30
HYGH		01110
15 (NRUN(3), EQ. 2) WRITE (7.	WEITE (7. 5)	011820
5 FORMAT (27HD SUBROUTINE DIFF		011630
HOD - HICROLES		011040
"	-	011850
IBR = MICRO(4) + 1		011060
ICLO3 = MSCPAD(IBR)	8	011870
IF (10P.EQ.27)	The state of the s	011880
		011
417	- HOCTADITARY + ANDICOMPT (MSCHADILURY), LHIGHT + ICA	
TAFN		011310
CHI	= MSCPAD(IBR) + AND (COMPLIMSCOAD(IAR)), LHIGH) + ICN	
OR IF (10P.EQ.29)		011940
		011950
THE	- IDIN + ANDICOMPLINSCPADITORI) + LHIGH) + ICN	011960
OR IF (10P. EQ. 30)		011970
THEN THEN		966110
TIME HISCH	TIME # MSCPAULIBRY * ANDICORPLITUINY, LAIGHT * ICN	666110
TAEN TAEN		010010
1	TOTA + DADICOMP (HICS); HIGH) + ICN	01.600
		012030
ITHP - AND (ITHP, LHIGH)	THI 6H)	012040
IF (NRUN (3). EQ. 2)		012050
TER		012060
WRITE (7, 2		
	FORMAT (THEINSTR , DZ, 24H :XECUTED, TEMP VALUE = ,C	106, 14,1 012000
		012090
ביונית		01211
I TEN		012113
IDOUT = ITHP	a	012130
£.3£		012140
MSCPAD(IBR) = ITHP	= ITMP	012150
CALL INHOBITAR, 198)	IAR, 198)	912169
ENO IF		012170
AE OVA		201710

CANTON OF THE PRETATION REDUIRED PRIOR TO	105	012210 01220 012240 012240 012250
EXECUTE THOSE CONTROL FUNCTIONS REQUIRED PRIOR TO 1012280 LOGICAL FINIS " ERROR " ICN " ICOUNT", IOIN " 0112280 COMMON CINTENTION" INCENTRALICATION " 0112280 LOGICAL FINIS " ERROR " ICN " ICOUNT", IOIN " 0112380 " HOUSS " HEMM " HEMM " HARHITOSA" " 0112380 " HOUSS " HEMM " 100 " 1012390 " HOUSS " HOUSS " HINS " 10108 " HOUSS " HOUSS " HINS " 10108 " HOUSS " HOUSS " HINS " 1012390 " HOUSS " HOUSS" HINS " 1012390 " HOU	LOGICAL FINIS ERROR LOGICAL FINIS ERROR COMMON CONTENTOR 10 CONTENTOR 10 CONTENTOR 10 CONTENTOR 10 CONTENTOR 10 CONTENTOR 1	
EXECUTE THOSE CONTROL FUNCTIONS REQUIRED PRIOR TO 1012280 LOGICAL FINIS FRROR TATION AND EXECUTION 1012280 LOGICAL FINIS FRROR 1 TEM MICHIGAN 101230 COMMON LODGY LOSG LOS	EXECUTE THOSE CONTROL FUNCTIONS REQUIRED INSTRUCTION INTERPRETATION AND EXECUTION COMMON CNIMENTIALLY COMMON CONTRACTOR COMMON CONTRACTOR COMMON TODOUT 105CR 1 TELM 1 MUCG 1 MSCPARTIAL	
INSTRUCTION INTERPRETATION AND EXECUTION 1012289 1012389 1	LOGICAL FINIS "ERROR "ICH "COMMON CONTRACTOR" ICH "COMMON CONTRACTOR" ITEM " Z MBUSS NERG NERG NERG NERG NERG NERG NERG NERG	
CCHCCAL FINIS ERROR COUNT LOUN COUNT	LOGICAL FINIS , ERROR , ICN , COMMON GNITHER (1054) , ICN , HELM	
CCHMON CONTENT CONTE	LOGICAL FINIS , ERROR , ICN , COMMON CNTREM(1024) , ICN , ITELM , BUSS , MENC , HEHI , MENC , HEROR , HEHI , MENC , HEROR , HE	***
COMMON CANTER ERROR COMMON CANTER ERROR COMMON CANTER ERROR COMMON CANTER ERROR ERRO	LOGICAL FINIS , ERROR , ICN , COMMON CNIMENTIALS, ITELM , I HOUST , ITELM , HEND , HEN	
COMMON GNTHEN(1024) , ICN , ICOUNT , IDIN , 012300 MUDOL	COMMON GNIMEN(1024) ICN . 1001 1056 . ITEM . HBUSS . MEMC . MEM . MICK . MSCPAD(16)	012290
MOUNT 105CR 17ELM MAINHT1024) 101230 MUSS MENG MERM MISSA MESM MISSA MESM MISSA MESM MISSA MESM MES	TOOUT , IDSCR , ITEH , NBUSS , MENC , MEN4 , MENC ,	
HBUSS HEHC HEHH HIGADIGI) HIGADIGIS HIGADIGIS HBUSS HEHC HEHH HIGADIGIS HIGADIGIS HIGADIGIS HBUSS HSCRAPADIGIS HIGADIGIS HIGADIGIS HIGADIGIS THE HIGADIGIS HOPO THE HIGADIGIS HIGADIS HOPO THE HIGADIGIS HIGADIS THE HIGADIGIS HIGADIS HIGADIS THE HIGADIS HOPO THE HIGADIS THE HIGADIS HOPO THE HIGADIS	MBUSS , MEHC , MEH ,	
MEDICAL MEDICAL MESTA MESTA MESTA MESTA MESTA MEDICAL MEDI	MICG , MSCPAD(16)	
NBUN(20), FINIS ERD3 NISTR NOPNO 012340	Charte Coope Charter Cooperation	-
THE THE TETTH 012850 THE THE THE THE TETTH 012850 THE	NKUNICOLI FINTS . ERRUY . HINSIK .	
CROTTED: EQ.11 MSR1 = 1 CROTTED: EQ.11 MSR1 = 1 CROTTED: EQ.11 MSR1 = 1 D12360 CROTTED: EQ.11 MSR2 = 1 D12420 CROTTED: EQ.11 MSR2 = 012470 CROTTED: EQ.11 MSR2 = 012570 CROTTED: E	, STIME , ITIME ,	
CROCLES.EG.11 HSR1 = 2 CROCLES.EG.11 HSR2 = ALINHKHEHC + 1) HBUSS = HAINHKHEHC + 1) CROCLES.EG.11 CROCLES.EG	KAD . KDA .	012360
CRO(10):-Eq.1) MBUSS = MAINHUMEHG + 1) LDIN = MBUSS CRO(13):-Eq.1) CRO(15):-Eq.1) MCCRO(15):-Eq.1) MCCRO(17):-Eq.1) MCCRO(17):-Eq.1		012370
MANUSS = MAINHUMENC + 1) MANUSS = MANUSS = MANUSS + 1) MANUSS = MANUSS = MANUSS + 1) MANUSS = MANUSS + 1) MANUSS = MANUSS + 1) MANUSS = MAINHUMENC + 1) MANUSS = MANUSS + 1) MANUSS = MANUSS + 1) MANUSS = MAINHUMENC + 1) MANUSS = MANUSS + 1) MANUSS + 1) MANUSS = MANUSS + 1) MANUSS		012380
MBUSS = MINHURENC + 1) CRO(13) = EQ 1 MINHURENC + 1) CRO(13) = EQ 1 MINSS = OR (MBUSS, MSR1) 012470 CRO(13) = EQ 1 MINSS = OR (MBUSS, MSR1) 012490 CRO(13) = EQ 1 MINSS = OR (MBUSS, MSR1) 012490 MINGROTIS = EQ 1 012490 MINGROTIS = EQ 1 012590 MINGROTIS = EQ 1 01259	THEN	015340
CRO(13).Eq.1) CRO(15).Eq.1) CRO(15	MBUSS = MAINMEMEN + 1)	01.2400
### CROST CROST FOR CAT FOR CA	IDIN = MBUSS	015 +10
(MICADIKAD), 108C2)) 012430 USS = OR (MBUSS, MSR1) 012490 012290 012500 012560 012560 012560 012560 012560 012560	END IF	012420
USS = OR (MBUSS, MSR1) 012470 012490 012590 012590 012590 012560 012560 012610 012610 012610 012610 012610 012610 012610 012610 012610	F (MICRO(13).EQ.1)	012430
USS = OR (MBUSS, MSR1) 012470 01240 01250 01250 01250 01250 01253 01253 01254 012619 012619 012619 012619	THEN	012440
USS = OR (MBUSS, MSR1) 012400 012500 012500 012530 012530 012560 012560 012610 012610 012610 012610 012610 012610 012610	HBUSS - NOT TOK INICADIKADI, IUSCAL	012420
USS = OR (MBUSS, MSR1) 012899 012899 012899 012899 012860 012860 012860 012860		
01250 01250 01250 01250 01250 01250 012610 012610 012610 012610 012610 012610		
012500 012500 012530 012530 012560 012610 012610 012650 012650 012650		042498
012530 012530 012560 012560 012610 012650 012650 012650 012650		012500
012540 012560 012560 012610 012620 012620 012650 012650 012650		012510
012530 012560 012560 012610 012610 012620 012650 012660	KOA * 1	
012560 012560 012610 012610 012610 01260 01260	OR IF IMICROICIF EQ. 11	012530
### ### ### ### ######################	TAEN	012543
(MICRO(22).EQ.1) 012560 N (MA = 3 E KOA = 3 CRO(23).EQ.1) 012619 CRO(23).EQ.1) 012620 CRO(23).EQ.1) 012660 IF (KAD. = MICRO(25) 012660	K0A * 2	012550
MA = 3 E KOA = 3 KOA = 0 012610 CR0(23) • Eq. 1) KAO = MICRO(25) If tkAD. Eq. 61 KAD = 3 012650	OR IF (MICRO(22), EQ. 1)	012560
KGA = 3 KGA = 0 GRO(23).EQ.1) GRO(23).EQ.1) GRO = MICRO(25) GRO = MICRO(25) IF (KAD.EQ.6) KAD = 3 012660	TYEN	012570
CRO(23).Eq.1) CRO(23).Eq.1) CRO(23).Eq.1) CRO(23).Eq.1) O12610 O12650 O12660 O12660 O12660 O12660	KOA = 3	012583
CRO(23) - EQ.1) 012610 CRO(23) - EQ.1) 012620 KAD = MICRO(25) 012660 IF = (KAD. EQ.6) KAD = 3 012660	11.3C	01250
CRO(23).Eq.1) MAD * MICRO(25) IF [KAD.EQ.6] KAD = 3 012650 012650		
MAD * MICRO(25) IF (KAD.EQ.6) KAD = 3 012650 012670		012620
KAD * MICRO(25) If tkAD.Eq.61 KAD * 3 012660	THEN	012630
012660		012640
The second state of the second		012660
	&FTUR.	012670
	END	012680

### BI NARY BITS 27-45 ###################################	
LOGISAL FINIS , ERROR , ICN , ICOUNT , IDIN , COMMON IDOUT , IDSCR , ITELM , MAINH(1024) , HBUSS , NEMC , HEMY , MICAD(6), MICAN(3), MICAD(6), MICAN(4), MIC	012730 +012740 *112750
CNTHENTIGE4) ICN , ECOUNT, IDIN , IDOUT , IDSCR , ITELM , MAINH1024) , MBUSS , MENG , MENH , MICAD(6) , MICDA(3), MICQ , MSCPAD(16) , MSL , MSR2 , MRUNICOT, FINES , ERROR , MINSTR , MOPNO ,	31.27.60 01.27.70
IDOUT , IDSCR , ITELM , MAINH(1024) , HOUSS , MEM , METALO, MICAGO, MICOA(3), MICQ , MICAGO , MINSTR , MOPND ,	012780
. MENO , MENY , MICAD(6), MICDA(3), MICAD(16), MICAD(16), MICAD(16), MINSTR , MODIO , MINSTR , MODIO ,	012790
201, FINIS , ERROR , MINSTR , MOPND ,	012800
THE PROPERTY AND A STATE OF THE PARTY OF THE	010210
MICRO(25), MCNT , STIME, , ITIME , IETIM	012830
COMMON KED , KDA , IOLDS	012840 112850
	012850
1) MBUSS = IDOUT	012670
(HICROIT), EQ. 11 MSR2 - MSR1	012880
IF (MICRO(8).EQ.1)	012890
HEM = MICRO(5)	012910
ELSE IF (NRUN(3), EQ.2) HRITE (7, +) MEMH	012920
	FEB. 1812940
	012950
MICRO(5) = 0	912960
ERROR = .TRUE.	012970
IF (HICRO(9), EQ. 1)	012990
T-t-EN	013030
7 = 7 N	013010
THE	013030
FORMAI (17 HUMAIN MEMORY LOG , 04, 14H CHANGED FROM , 06,	6, 4H TO , 013050
06; 194 84 CONFROL BIT 35, 1	013868
END IF HAINMEN = HBUSS	613073
	013090
ROLIZT.EQ.1.AND.HEMH.NE.01	013100
THEN	013113
FORMAT (14-HDERROR: FETCH BIT ON BUT M-ADOR = , 06, 1M.)	013120 013130 043440
11	
RO(14).EQ.1) K ALU OUTPUT; IF = ZERO, SET SR1 = 1:	013170
IF (IDOUT.EQ.6) MSR1 = 1	013190
END IF IF (MICRO(16).EQ.1)	013210
THEN * OR (MICRO(51, MSRI)	013220 013233

30 FORMAT (29MBNEXT H MODIFIED 3Y MSR1 TO: , 04)	013250
1 (3	013260
IF (MICRO(19) -EQ-1) MEMC = AND (MBUSS, MSK10)	013280
IF (ALCKO(16)-EQ-U) LIELA = ABUSS	013300
THE	013310
HICDA(KDA) = HBUSS	013320
END IF	013340
IF (NRUN(3), EQ. 2) CALL REPT1	013350
RETURN	013360
ENG	0.0010
D ERRORS FOUND IN THIS ROUTINE. 69 CARDS READ.	68 CARDS OUTPUT.

		013390	
****	00m2100xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	013400	
0 0	OUTPUT FINAL CONTENTS AND STATISTICS	*013410 *013420	
* * * *	***************************************	*013430	
3	GOOD TIME	013450	
	CNIMEM (1024) . ICN . ICOUNT . IOIN .	013470	
1	IDOUT , IDSGR , ITELM , MAINM(1024)	013480	
2	S , MEMC , MEMH , MICAD(6), MICOA(3),	013490	
3	MSCPAD(16) . MSR1 . MSR2 .	013500	
+ u	MICROSOF MINIS , ERROR , MINSTR , MOPNO ,	013510	
`	10003	013530	
	(7, 10) STIME, ICOUNT, MCNT, MSCPAD, IDIN, IDOUT, MBUSS,	013540	
•	MEMC, MEMH, MICQ, MICAD, MICDA, ICN, IDSCR, MSRI, MSR2	013550	
10	FORMAT (26HIRUN TERMINATED AT TIME : ,FI5.8,15H SECONDS AFTER ,(,013560	
	OR , 15,17H M43R0S EXECUTED. //	013570	
	D CONTAINS:,/1H , 15 (/5x, 08), //	013580	
•		013590	
•	= ,04,10H, H-RES = ,04,10H, Q-REG = , 06, //	013600	
•	,5(2x,08),10H, 0/4 S = ,3(2x,08),//	013610	
•	RY = ,12, 20H, INPUT DISCRETES = , 08, //	113620	
	946 MSR1 = , 11, 94, MSR2 = , 11)	013630	
	RETURN	013640	
	END	013650	

DETAILED OUTPUT FROM MICRO-INSTRUCTION EXECUTION LOGICAL FINIS , ERROR COMMINION CONTROLS , MAINHOLD , MAINHO	SUBROUTINE REPTI	013660
DETAILED OUTPUT FROM MICRO-INSTRUSTION EXECUTION LOGICAL FINIS , ERROR COMMON TOOL TOOL TOOL TOOL TOOL TOOL TOOL TO		U135/U
DETAILED DUTPUT FROM MICRO-INSTRUCTION EXECUTION ***********************************		•013690
LOGICAL FINIS , ERROR , ICN , ICOUNT , IDIN , IDOUT , IDOUT , IDSCR , ITELN , HAINHIGE+) , HICDA(3), HENCE , H	ON MICRO-INSTRUCTION	*013700
LOGICAL FINIS , ERROR COMMON , ICOUNT , IDIN , COMMON CONTENTION , IDSCR , ITELM , MAINHILO24) , MBUSS , MEMC , MEMH , MICADIG) , MICADIG , MICADI		
COMMON CNTMENTING, ERROR COMMON CNTMENTOQ4) MEDUC , IDSCR , ITELN , HAINHIDG4) MEDUC , HSCPAD(16) MEDUC , HSCPAD(17) MEDUC , HSCPAD(17) MEDUC , HSCPAD(17) MEDUC , HSCPAD(17) MEDUC , MSTR, MOD, MSCPAD(18), MSCPAD(18), MSCQPAD(18), MSCPAD(18) METE (7,16) INSTR, MOD, MSCPAD(18), MSCPAD(18), MSCQPAD(18), MSCQC, C2, C2, C3, C3, C3, C3, C4, C4, C4, C4, C4, C4, C4, C4, C4, C4		013730
COMMON CNTHEM 11024) 10001	FINIS , ERROR	013740
IDOUT IDSCR ITELM MAINHILD24) MBUSS MBUSS MBUSS MBCD MBCD	CNTMEM (1024) . ICN . ICOUNT .	013750
MBUSS , MEMC , MEMH , MICADIO), MICADIO), MICADIO), MICADIO , MSR2 , MICADIO , MSR2 , MICRO(25), MCN1 , ERRO? , MINST , MOFNO , MICRO(25), MCN1 , STIME: , ITIME , IETIM , MICADIO , KDA , IOLOB , STIME: , ITIME , IETIM , INST = MICRO(1) , MCD = MCD = MICRO(1) , MCD =	. IDSCR . ITELM . MAINM(1024	013760
MICA	S , MEMC , NEM4 , MICAD(6),	013770
MECHICAL TIMES : ERROS : TIME : LETIM COMMON KAD : KDA : 10L08 INSTR. = MICRO(1) MOD = MICRO(1) MICRO(1) I = IA - I MICRO(1) + MICRO(1B) MICRO(1B) MICRO(1B) MICRO2 CORMAT (1846EXECUTED OP CODE ,02,134 MITH MODIFIER ,11,144 / PORMAT (1846EXECUTED OP CODE ,02,134 MITH MODIFIER ,11,144 / TH A-REG .02:12H CONTENTS INITIALL = .05,84, NOW = .06, N ITEL2 = ITELM MICRO = MICRO ERRORS FOUND IN THIS ROUTINE . 33 CARDS READ 2 SUBROUTINE REPT2 BETURN ERRORS FOUND IN THIS BOUTINE . 33 CARDS READ 2 RETURN ERRORS FOUND IN THIS BOUTINE . 33 CARDS READ 2 RETURN END	, MSCPAD(16) , MSR1 , MSR2	313780
COMMON MICRO(25), MCNI , STIME: , ITIME , IETIM LNSTR = MICRO(1) HOD = MICRO(2) IA = MICRO(1) HOD = MICRO(2) IA = MICRO(2) IA = MICRO(4) + 1 LB = MICRO(4) +	FINIS , ERRDS , MINSTR ,	013790
COMMON KAD , KDA , IOLOB INSTR = HICRO(1) HOD = MICRO(2) IA = MICRO(3) + 1 IB = MICRO(4) + 1 IB = MICRO(4) + 1 I = IA - 1 J = IB - 1	MICRO(25), MCNT , STIME , ITIME ,	013800
INST? = HICRO(1) HOD = MICRO(2) HOD = MICRO(3) + 1 IA = MICRO(4) + 1 IB = MICRO(4) + 1 I = IA - 1 J = IA - 1 J = IB - 1 HRITE (7,16) INSTR,MOD,1, HSCPAD(IA), J, IOLOB, MSCPAD(IB), MICQ2, TH A-REG, 02,12H CONTENTS = ,05,777 H B-REG, 02, 12H CONTENTS INITIALLY = ,05,777 H B-REG, 02, 12H CONTENTS INITIALLY = ,05,777 H B-REG, 02, 12H CONTENTS INITIALLY = ,05,84, NOM = ,06,77	KAD , KDA ,	013810
HOD = MICRO(2) 1	11	013820
IA = MICRO(3) + 1 IB = MICRO(4) * 1 I = IA - 1 J = IB - 1 MRITE (7,16) INSTR,MOD,I, MSCPAD(IA), J, IOLOB, MSCPAD(IB), MICQ2, TELP (1846EXECUTED OF CODE , OZ,134 MITH MODIFIER , II,114, / 74 M = REG , OZ, 124 CONTENTS INITIALLY = , OS, 84, NOM = , OS, / 284 TELEM CONTENTS INITIALLY = , OS, 84, NOM = , OS, / 284 TELEM CONTENTS INITIALLY = , OS, 84, NOM = , OS, / / 284 TELEM CONTENTS INITIALLY = , OS, 84, NOM = , OS, / / 284 TELEM CONTENTS INITIALLY = , OS, 84, NOM = , OS, / / 284 TELEM CONTROLS SUBROUTINE REPTS SUBROUTINE REPTS DEFAILED OUTPUT FROM INSTRUCTION MODIFICATION AND CONTROLS REION, END. FERDAN FOUND IN THIS BOUITINE.	"	01.38.30
18 = HICROIG) + 1 1 = 1A - 1 J = 1B - 1 J = 1B - 1 HRITE (7,10) INSTR,MOD,11, MSCPAD(IA),J,IOLOB,MSCPAD(IB),MICQ2, 1 IEL2, ITELM 1 IEL2, ITELM 22H G-REG .02;12H CONTENTS = .05;7/7 PH B-REG02, 28H Q-REG CONTENTS INITIALLY = .05; 84, NOM = .06, 1/ 28H Q-REG CONTENTS INITIALLY = .06,8H, NOM = .06, 1/ AICQ2 = HICQ RETURN END ERRORS FOUND IN THIS ROUTINE. 33 CARDS READ. SUBROUTINE REPT2 UETAILED GUIPUT FROM INSTRUCTION MODIFICATION AND CONTROLS RETURN END SUBROUTINE THIS POULTINE. 4, CARDS READ. FERDAN FOUND IN THIS POULTINE. 4, CARDS READ.	IA = MICRO(3) + 1	013840
I = IA - I J = IB - I HRITE (7,10 INSTR,MOD,I, MSCPAD(IA),J,IOLOB,MSCPAC(IB),MICQ2, , ITEL2, ITEL4 FORMAT (1846EEECUTED OF CODE,O2,134 MITH MODIFIER, 11,114, / TH A-REG,O2,12H CONTENTS = ,05,77 7H B-REG, C2, 22H CONTENTS INITIALLY = ,05,77 7H B-REG, C3, 28H Q-REG CONTENTS INITIALLY = ,05,84, NOM = ,06,7/ 28H Q-REG CONTENTS INITIALLY = ,06,84, NOM = ,06,7/ MICQ2 = HICQ RETURN END ERRORS FOUND IN THIS ROUTINE, 33 CARDS READ. SUBROUTINE REPT2 UEFAILED OUTPUT FROM INSTRUCTION MODIFICATION AND CONTROLS REIURN END ERRORS FOUND IN THIS BOUTINE, 3 CARDS READ.	IB = MICROGAT + 1	013850
J = 18 - 1 WRITE (7,1G) INSTR,MOD,1, MSCPAD(IA),J,IOLOB,MSCPAD(IB),MICQ2, ITEL2, ITELM TH A-REG 102,12H CONTENTS 22H CONTENTS INITIALLY = ,J5,7/7 7H B-REG , / C2, / 28H Q-REG CONTENTS INITIALLY = ,J5,84, NOM = ,J6,7/7 / HECQ = ,J6,84, NOM = ,J6,7/7 / HECQ	-	
MRITE (7,16) INSTR, MOD, 1, MSCPAD(IA), J, IOLOB, MSCPAD(IB), MICQ2, ITEL2, ITEL4 ITEL2, ITEL4 ITEL2, ITEL4 ITEL5, ITEL4 ITEL5, ITEL4 ITEL5, ITEL4 ITEL5, INT A-REG, O2, 12H CONTENTS 28H Q-REG CONTENTS INITIALLY = ,05,8H, NOM = ,06,71 28H Q-REG CONTENTS INITIALLY = ,06,8H, NOM = ,06,71 ITEL2 = ITELM MICQ2 = MICQ ERRORS FOUND IN THIS ROUTINE, 33 CARDS READ. SUBROUTINE REPT2 SUBROUTINE REPT2 BETURN ERRORS FOUND IN THIS BOUTINE, 33 CARDS READ. SUBROUTINE REPT2 BETURN ERRORS FOUND IN THIS BOUTINE.		000010
FORMAT (1846EXECUTED OF CODE ,02,134 WITH MODIFIER ,11,114. / TH A-REG .02,124 CONTENTS = ,03,7/7 74 B-REG02, 224 CONTENTS INITIALLY = ,05,84, NOM = ,06,7/ 284 TELEM CONTENTS INITIALLY = ,06,84, NOM = ,06,7/ MICQ2 = ITELM MICQ2 = MICQ RETURN END ERRORS FOUND IN THIS ROUTINE. 33 CARDS READ. SUBROUTINE REPT2 DETAILED GUIPUT FROM INSTRUCTION MODIFICATION AND CONTROLS RETURN FORMAT (1846EXECUTION AND CONTROLS FORMAT (1846EXECUTION AND CONTROLS) FORMAT (1846EXECUTION AND CONTROLS)	(7.16) TNOTO	
FORMAT (1846EECTED OF CODE ,02,134 MITH MODIFIER ,11,114, 7 H 4-REG ,02,12H CONTENTS = ,05,77 7H 8-REG , C2, 22H CONTENTS INITIALLY = ,05,77 7H 8-REG , C2, 22H CONTENTS INITIALLY = ,05,84, NOM = ,06,77 7H 8-REG , C2, 28H Q-REG CONTENTS INITIALLY = ,06,84, NOM = ,06,71 7H ICQ2 = HICQ	TACE TACE	
SUBROUTING REPTS SUBROUTING R	* TIELC, LIELM	0.200
SUBROUTINE REPTS FOUND IN THIS BOUTINE. SUBROUTINE REPTS FOUND THIS BOUTINE. SUBROUTINE REPTS FOUND THIS BOUTINE. SUBROUTINE REPTS FETURA END FETURA FET	TIONUE ACCOUNT WITH MULTIPER	00000
SUBROUTINE REPT?	AFREG +UC+12H CONIENTS - +US+// TH BFREG +	OTECTO
SUBROUTINE REPT?	CONTENTS INTITACE = 133, 51, NON = 100, 1	
ITEL2 = ITELM HICQ2 = HICQ RETURN END ERRORS FOUND IN THIS ROUTINE, 33 CARDS READ. SUBROUTINE REPT2 USETALED GUIPUT FROM INSTRUCTION MODIFICATION AND CONTROLS RETURN END FERRORS FOUND IN THIS BOUTINE.	THE CONTENTS INTITALLY = , UB, 84, NOM = , UB,	
HIGGS = HIGG RETURN END ERRORS FOUND IN THIS ROUTINE. 33 CARDS READ. SUBROUTINE REPT2 DETAILED OUTPUT FROM INSTRUCTION MODIFICATION AND CONTROLS RETURN END FROM THIS POSITINE.	28H IELEM CONIENIS INTITALLY = , UB, 8H, NUM =	013940
RETURN RETURN END ERRORS FOUND IN THIS ROUTINE. 33 CARDS READ. SUBROUTINE REPT? DETAILED OUTPUT FROM INSTRUCTION MODIFICATION AND CONTROLS FEDAN FORDS FOUND IN THIS BOUTINE.	11	013950
ERRORS FOUND IN THIS ROUTINE. 33 CARDS READ. SUBROUTINE REPT2 DETAILED OUTPUT FROM INSTRUCTION MODIFICATION AND CONTROLS RETURN END FROM THIS BOUTINE.	•	013900
ERRORS FOUND IN THIS ROUTINE. 33 CARDS READ. 2 SUBROUTINE REPT2 DETAILED OUTPUT FROM INSTRUCTION MODIFICATION AND CONTROLS RETURN END FERDAN FOUND IN THIS BOUTINE.	KEIUKN	013970
SUBROUTINE REPTZ BETURN SUBROUTINE REPTZ UETAILED UNIPUT FROM INSTRUCTION MODIFICATION AND CONTROLS RETURN END FERDAN FORMAN F		
SUBROUTINE REPTZ DETAILED OUTPUT FROM INSTRUCTION MODIFICATION AND CONTROLS RETURN END FROM STATES AND THIS POILTINE.	ROUTINE. 33 SARDS READ.	CARDS OUTPUT.
SUBROUTINE REPT2 DETAILED OUTPUT FROM INSTRUCTION MODIFICATION AND CONTROLS RETURN ENU FROM IN THIS POULTNE.		
ON INSTRUCTION MODIFICATION AND CONTROLS	SUBROUTINE REPT2	013990
RETURN END FRRDS FOUND IN THIS PRITTING.		014000
FRRDS FOUND IN THIS POSITING.	RETURN	014010
FRANCE FORM IN THIS POSITING. A CARDS READ.	2	
ENTRY OF THE PROPERTY OF THE P	ERRORS FOUND IN THIS ROUTINE. + CARDS READ.	3 CARDS OUTPUT.

	H EMUL (14°UT, JUTPUT, TAPES = INPUT, TAPET=
	CONMON CNIMERCIDES, , ICN , ICOUNT, IDIN , IDOUT , IDSCR , ITELM , MAINM(1924) ,
5	MBUSS , MEMC , MEMH , MICAD(6),
	. ITIME . IETIM
	. IOL 08
10	DATA LFTHS /3778/
	10 FORMAT (20 0 4)
	WRITE (7, 20) NRUN
4	20 FORMAT (23HIRUN SONTROL CARD = , 2304)
	CALL INIT
	CALL SETON
25	MICS = 0
	IF (=NOT)
	(INCOLOR OF TOTAL OF
	1 I N
52	
	tel
	SHRIGHTAN DECODED, VIIM DEC DCT, 3X, 12MMENDRY LEFT , SHRIGHT, 4X, 14HCONTROL MEMORY, ZX, 9HDFC DCT , 3X,
-	ITHMEMORY LEFT RIGHT, 4X, 14HCONTROL MEMO
30	99998 IF(.NDT.
	66 0
	7 . 2 . 2
35	CRUMINITAL TOTAL
,	MAINL = SHIFT (MAIN, -8)
	HATAR - AND (HAIN, LFINS)
	TO THE TENER OF TH
94	STELL STEEL
	MRITE (7, 90) NN, 4N, MAINH(N), MAINE, CHINE, CNTHEM(N),
	90 FORMAT (IN TREEX-04-12/3-10/3-10/3-10/4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
4.5	OTO440747074470744714747 7 + Z = Z
	-
A property and the same	99997 CONTINUE
	99999 CONTINUE
25	SASSA INC. NOT. (FINIS. OR. ERROR!)
	. 160 10 99995
	CALL DCODM
	יייייייייייייייייייייייייייייייייייייי
55	150 10 9993

66 150 TO 3932 65 150 TO 3932 65 150 TO 3932 65 150 TO 3932 66 150 TO 3932 67 150 TO 3932 68 150 TO 3932 70 150 TO 3932 70 150 TO 3932 70 150 TO 3932 71 150 TO 3932 72 150 TO 3932 73 150 TO 3932 74 17		CALL EXECH IFC.NOT.	655
## MOD ## 9994 ## MOD ## POD ## 9994 ## MOD		. (NRUN(3) . EQ.21	95
15 FURNAL 153 HAJO, CHYRENIN (MISCOIL), I 1 25) 16 FURNAL 153 HAJO, CHYRENIN (MISCOIL), H H JOH, ZX, Z011X.111///) 17 FURNAL 153 HAJO, CHYRENIN (MISCOIL), H H JOH, ZX, Z011X.111///) 18 FURNAL 153 HAJO, CHYRENIN (MISCOIL), H H JOH, ZX, Z011X.111///) 19 FURNAL 153 HAJO, CHYRENIN (MISCOIL), H H JOH, ZX, Z011X.111///) 18 FURNAL 11, HOW, H JOH, HOW, HOW, HOW, TOW, H JOH, HOW, HOW, TOW, H JOH, HOW, HOW, HOW, HOW, HOW, HOW, HOW, H		26666 01 091 × 1	25
15 FORMAT 153 HOTHE FOLLOWING CM INSTRUCTION HAS JUST BEEN EXCUTED. 160 10 99991 IF. NOT. 160 10 99991 IF. NOT. 160 10 99991 CONTINUE 160 10 10 10 10 10 10 10 10 10 10 10 10 10		HRITE (7, 15) M43LD, CNTHEMIN), (MICRO(I), I = 1, 25)	£ 6
## 1 //14 + 00 - 5x.016.413x.021.44	-	S FORMAT 193 HUTHE FOLLOWING ON INSTRUCTION HAS JUST BEEN EXECUTED!	
### 16			93
*** **********************************	6666	n = 1 I F(.NOT.	. · · ·
** ** ** ** ** ** ** ** ** ** ** ** **			9.5
36 FORMAT (134) SCRAICH PA)(*, 02* 114) CONTAINS - 06* 24 *, 1 H = H + 1. 50 FORMAT (114) CAP (1 + 02* 114) CONTAINS - 06* 24 *, 1 H = H + 1. 50 FORMAT (114) CAP (1 + 02*) FORMAT (114) FOR (1 + 02*) FOR FOR (1		. 160 TO 99990	65
## # # # # # # # # # # # # # # # # # #			
99990 CONTINUE F H 1 1 1 1 1 1 1 1 1		WRITE (7, 30) K, MSCPAD(M)	12
00 10 9999		TOKARI (13H SCRAIGH PRJ) ; UC; 11H) CONININS ; US;	
99990 CNITUUE 50 FORMAT (11H0DATA IN = .08.134, DATA DUT = .08.44, BUSS = .08, 10		1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	
##IFE (1 504 101W 10045 WENC, 10W, 10S2 10S3 10			25
50 FORMAT LITHODATA IN = .08.134, DATA DUT = .08.94, BUSS = .08, 11. 14.) 99992 CONTINUE FIRE = STIME + TINCR FIRE = FIRESTIME; HIGS = HIGS + 1 FIRES = FIRES = HIGS + 1 FIRES		MELLE 17, 501 JOIN, IDOUT, HEUSS, NEHC, JCN, HSRI, HSR2	- 52
11. 14.) 99992 CONTINUE 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14.) 11. 14. 11. 14.) 11. 14. 11. 1		1 FORMAT (11H0DATA IN = .08,13H, DATA OUT = .08,9H, BUSS = .08,	11
99992 CDNTINUE SITHE = SITHE + ITACR ITACR		194, G-2:G = ,00,/94 CARRY = ,11,64, SR1 = ,11,94, SR2 =	7.8
99992 CONTANUE ITHE = IFIX(5TL=1) INCS = MCS + 1 INCS + 1		. II, 14.)	6.
STIME = STIME + TINCR	-	2 CONTINUE	9.6
HICS = HICS + 1 IF (*NOT*, 100 10		STIME = STIME + TINGR	1:
TF (-NOT) = TLUS = TLUS = TRE			
#RITE (7, 40) STRME, ITIME, MICS #RITE (1, 40) STRME, ITIME, MICS #ROWISHUE #ROWING (10, 9998) #RITE (1, 40)		HICS - HICS + 1	200
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
##ITE (7, 40 STIME, ITIME, MICS 9988 CONTINUE FF. NOT. (NEUN(9) . Eq. 5) (NEUN(9) . Eq. 5) (HESS-SE, MRUM112) (HESS-SE, MRUM12) (HESS-SE,		150 00 0000	96
99949 CDNINUE (RINN(9).EQ.50. (RINN(9).EQ.50. (RINN(9).EQ.50. (RINN(9).EQ.50. (RINN(9).EQ.50. (RINN(9).EQ.50. (RINN(9).EQ.50. (RINN(9).EQ.50. (RINN(9).EQ.50. (RINN(9).EQ.10. (RINN(9)		WRITE (7, 40) STIME, ITIME, MICS	87
99949 CONTNUE IF (-NOT) - (-N	*	6 FORHAT (1Hd. #15.6, 5x, 16, 5x, 15)	86
IF (NOT 1 1 1 1 1 1 1 1 1	9666	9 CONTINUE	8.6
	-	If the No.	2
If (. NOT. 57500 (1005) 56. MRUNILD1) 160 TO 99987 FEMN = 0 9987 9998 CONTINUE 60 TO 99996 9999 CONTINUE 60 TO 99996 9999 CONTINUE 60 TO 99996 60 TO 99996		. (NKUN(9).E4.3)	16
## CAL FINAL CALL FINA		IF (.NOT.	33
#ENH = 0 9987 #ENH = 0 FINIS = .TRUE. 99987 CONTRIVE 99985 CONTRIVE 99985 CONTRIVE 60 TO 99996 60 TO 99996 10 TO 99996 99995 CONTRIVE 11 CALL FINAL CALL FINAL CALL FINAL AN = 1 FINAL W =		• (HICS, 6E, NRUN(191)	*
### ### ##############################		.160 TO 99987	35
99984 CONTINUE 99984 CONTINUE 99993 CONTINUE 100 TO 99994 100 TO 99996 100 TO 9999999999999999999999999999999999			
99997 CONTANUE 99995 CONTANUE 99993 CONTANUE 60 TO 99996 60 TO 99996 99995 CONTANUE 1			
99995 CONTINUE 60 10 99994 99995 CONTINUE 61 FINAL 62 FO 99996 99995 CONTINUE 1 KRUN(4), EQ.1) 1 K I I W. I I I I I I I I I I I I I I I I	9666		9 6
99993 CONTINCE CALL FINAL CALL FINAL CALL FINAL GO 70 99996 99995 CONTINCE * 160 70 99966 N = 1 ** 160 70 99966 N = 1 ** 160 70 99966 ** 160 70 99966 ** 160 70 99966 ** 160 70 99966 ** 160 70 99966 ** 160 70 99966 ** 160 70 99966 ** 160 70 99966 ** 160 70 99966 ** 160 70 90 99966 ** 160 70 90 99966 ** 160 70 90 99966 ** 160 70 90 99966 ** 160 70 90 99966 ** 160 70 90 99966 ** 160 70 90 99966 ** 160 70 90 99966 ** 160 70 90 99966 ** 160 70 90 90 90 90 90 90 90 90 90 90 90 90 90	3666	11 11	441
#GNT = #GNT + 1 GAL FINAL GO TO 9996 99995 CONTINCE * MRUN(4).EQ.1) * * * * * * * * * * * * * * * * * * *	0000	F -	101
CALL FIMAL GOTO 9996 99995 CONTRUCE IF.NOT NETE (7.00) 99985 IF.NOT NETE (7.00)		- 1	19.2
09 99 5 CONTINUE * (NRUN(*).EQ.1) * 160 F0 99966 N = 1 ** N =		CALL FINAL	113
99995 CONTINUE 181.NOT. 180.TO 99966 18 1 N 1 N 2 N 2 N 2 N 2 N 2 N 2 N 2 N 2 N		96666 01 09	+61
# # # # # # # # # # # # # # # # # # #			115
. (NRUN(+).EQ.1) . 160 TO 99986 N = 1		If t. NOT.	136
N = 1 199965 N = 1 N =		. (NRUN(+) .EQ.1)	23
99965 IF(.NOT. (M.LE.L.)241		*160 T0 99986	
99965 IF(.NOT. (W.L.E.1.)24)			60:
IP (• NO)			1:
	2666	TE C. NOI.	111
		• • • • • • • • • • • • • • • • • • • •	311

	בייסו			
115			115	
	MAIN = MAINH(N)		116	
	MAINL = SHIFT (MAIN, -8)		111	
	MAINR = AND (MAIN, LFTMS)		118	
	HAIN = MAINH(NR)	The second secon	119	
120	MAINJ = SHIFT (MAIN, -8)		120	
	MAINS - AND (MAIN, LPINS)		121	
	MRITE (7. 90) NN. NN. MAINH(N), MAINE, MAINR, CNTMEM(N),		122	
	N.N. MAINN(NR), MAINS, CNTHEM (NR)		123	
			124	
25	60 10 99985	and the same and the same of t	125	
	99984 CONTINUE		126	
	99986 CONTINUE	-	127	
	CALL STATS		128	
-	STOP	and a part of the property of the part of	129	
130	G Z u		130	

	SUBROUTINE INIT	EINIT	0000					131	
	COMMON	CNTHEM (1324)	24)	. ICN		TCOUNT .	. NIGI	133	N. F.
	•	1000I	IDSCR	, ITELM		MAINM(1024)		13	
5	7 10	M TCO	MSC PAD (16)	16) MEMH	• •	MICAD (6), MICDA(3)	MICDA(3),	135	
		NRUN(2:1)	1	FRROR	-	HINSTR .	HOPPID .	131	
	2	MICRO(25)		. STINE	•	ITINE .	IETIM	138	•
	COMMON	KAD .	KJA	100.08		the state of the same of the same of the same of		£1)	6
10	ICN = O							1,0	
	ICOUNT =								4.
	n = NIOT							1.4	7
	- 1000T							77,	
1.5	TELM = 3				-			4	
	KAD = 0							146	. 9
	KOA = U		-		-		-	147	
	MBUSS = 0							148	
	HENC = 0							541	-
. 02	MENH # 0							150	
	MCG = 0			-				151	1
	MSK1 = 0							152	
	MSCPAD(1)	I NPIN (2)						123	
	2 - N			-			-	135	-
	99999 IF (.NOT.							156	
		(Note 15)				The same of the sa		4	
-	See of Date	28.6		,				158	
30	1 + Z = Z							150	
	66 66 01 09	66			-	•)		151	
	30998 CONTINUE							152	
	1 = N							51	
	9999/ IF (. NO! .							91	• •
	0 TO 090	946						100	
	MECHOCK							1	
	T + Z Z							151	•
	60 10 99997		-			***************************************		159	•
40	ENNITHOD 96666					-		171	
	MECDACE	0 =			-			-	
	MICDA(2)	0 =						17.	2
									-
-	FINIS = .	FALSE.						174	•
	, "	TALSE.						176	
	الت	0.0					The same of the sa	111	
	ITINE = 0							178	
	TETTH - 0		-	-		-		64	

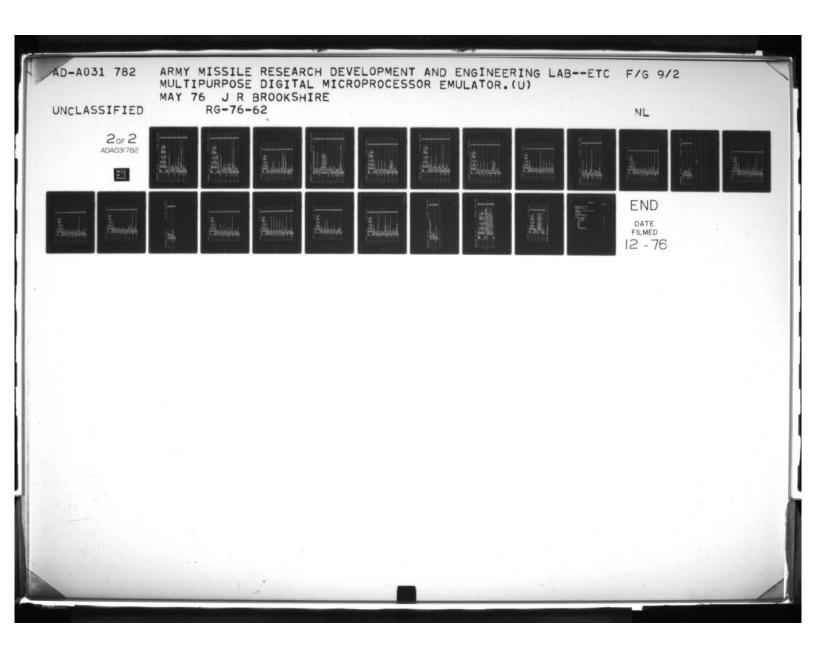
	TINE UCOUS	192
		195
5	. MENC	196
	FRROR . MINSTR , MOPNO	138
	MICRO(25). HONT . STINE . ITIME .	189
	. KDA .	191
7.0	VOO TENT OFF	192
		193
	CALL GETCH (IPTR., 3, 5, MOP, IFLAG)	194
	HEGROLI'S E XOR (MSKI, 40P)	196
12	CALL GETCH (IPTR, 5, 3, MOP, IFLAG)	197
		661
		133
	CALL SETON (IPT4: 0 4 4 NOP. ITCAO)	201
50	HICKOLD TO THE AG	212
	CALL GETCH (IPTR. 12, 4, MOP, IFLAG)	233
	MICRO(4) = NOP	\$02
	IERR = OR(IERR, IFLAG)	612
62	CALL SEICH (1P1: 15, 10, HOP, I'CAG)	212
	HECKOLO = NOTERO TELAC	23.8
	CALL GETCH (IPTR, +5, 3, MOP, IFLAG)	519
	MCRO(25) = MOP	210
30	JERR = DROIERRY IFLAGS	213
	NCOL = 26	21.3
	INOX = 6	*12
96	99999 IT (*NOT 1751-AC 1751	21.6
20	96666 01 091	21.7
	CALL GETCH (IPPT), NCOL, 1, MOP, IFLAG)	613
	MICRO(INDX) = HOP	220
4.0	INDX = INDX + 1	221
	66666 01 05	555
	99998 CONTINUE	22.3
	154 (NOT)	225
6.9	160 10 99997	22.6
	IF (.NOT.	227
	, (NRUNI37.EQ.2)	229
	. 10 03330	639
90	T (15HLT4	231
	99996 CONTINUE	232
	50005 CO 109 99995	234
	WRITE (7, 10) IF.AS	235
-		211
	ERROR * . TRUE.	238

007 1 105CR 1 ITELM 1 CQ 1 HENC 1 HENC 1 CQ 1 HSPAC 1 HENC 1 CQ 163 HENC 1 CQ 153 HENC 1 CQ 123 HENC 1 CQ 123 HENC 1 CQ 123 HENC 1 CQ 12 HENC 1 CQ 13 HENC 1 CQ 13 HENC 1 CQ 14 HENC 1 CQ 16 HENC 1 CQ 17 HENC 1 CQ 1	ICDA(3). ISR2 .
HENCS . HERC . HEHU . HIGGACISJ . HERC . HS. POLIGO . HIGGACISJ . HG. M. AS. AS. AS. AS. AS. AS. AS. AS. AS. AS	
SA 999 999 999 999 999 999 999 999 999 9	•

- E C C C C C C C C C C C C C C C C C C	
11181 1181 1181 1181 1181 1181 1181 11	250
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	152
1000 000 000 000 000 000 000 000 000 00	367
99999999999999999999999999999999999999	254
99999999999999999999999999999999999999	255
28 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	530
8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	752
NS N N N N N N N N N N N N N N N N N N	663
99999999999999999999999999999999999999	250
29999999999999999999999999999999999999	251
11.00 000 000 000 000 000 000 000 000 00	292
99999999999999999999999999999999999999	563
2000 000 000 000 000 000 000 000 000 00	561
2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	552
11.5 KILL 11.5 K	4:6
11.6 11.8 11.8 11.8 11.8 11.8 11.8 11.8	25.8
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	652
11.6 11.6 11.6 11.6 11.6 11.6 11.6 11.6	27.0
1.6 199999999999999999999999999999999999	27.2
11.00 000 000 000 000 000 000 000 000 00	273
169 169 169 169 169 169 169 169 169 169	472
18 1 18 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	275
181 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	220
18 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	27.
181 184 00 184 00 00 00 00 00 00 00 00 00 00 00 00 00	12
18 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6	291
1189 1189 1189 1189 1189 1189 1189 1189	281
1 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	
94846	294
0 95 0	53
936	586
986	762
99992 IF (*NO F. (MOO. EQ.5)	867
99992 IF (*NOT. (MOD.EQ.5)	230
. (MOD.EQ.5)	291
	262
1960 10 10 10 10 10 10 10 10 10 10 10 10 10	293
1811 = SHIFT (MSCPAULIST) -151	166

		298
9	99990 CONTINUE	300
	1 2	335
	IBIT = SHIFT (MISQ, -15)	313
	MICQ = AND (SMIFF(MICQ, 1) , KMASK)	304
65	MICQ = OR(MICQ, 1)	315
		3)6
	MSCPAD(18) = OR (MSCPAD(18), 18IT)	33.7
	IDOUT = MSCPAD(IS)	33.8
	8666 0 09	319
2	99991 IF(•NOT•	31.0
		311
	CELEBOTORION ON THE	316
	SHIFT (TAIT, 15)	3.5
22		315
	IF(:NOT.	31.6
		31.7
	78666 01 091	90 6
	T I TACE	91.9
00	99987 CONTINUE	321
		33.5
	99986 CONTINUE	323
	MCQ = SHIFT (MI33, -1)	324
85	MICQ = OR(MICQ, ISIT)	325
	= SHIFT (MSCPAD(I	326
	MSCPAU(18) = UKINSCPAU(18) THIELD	327
	TOUT - ASCRAULS	0.52
06	NO SECOND	33.0
:		331
	5 FORMAT (26H INVALID INSTR MODIFIER = ,04,15H, RUN HALTED.)	33.2
	ERROR = .TRUE.	333
10	1	334
26	TO INCOME CONTRACTOR (*) TO MOUS TOUGH	335
	IF (. NOT.	337
	. (NRUN(3) .EQ.2.03. NRUN(3) . EQ.3)	338
	.)60 10 99985	339
100	000	0.0
	99999 CON INGE	341
		31.3

173 CR 176 1		SUBRUTINE EXECM	355
10017 115CR	1	FIGURE . ICH . ICOUNT .	346
## ## ## ## ## ## ## ## ## ## ## ## ##		IDOUT , IDSCR , ITELM .	347
NEGOLIS NESPADILS NESPADILS		. MENC . MEMH .	
NEGROES FINE FINE NEGRO NEGR		MICQ . MSCPAD(16) . MSR1 . MSR2	34.9
MICRO(23, MONT , SITHE , ITTHE , ITTHE , ITTHE , LELIN *EQ.1)		NRUN(23), FINIS , ERROR , MINSTR ,	220
EG.11 199999 197011.61 197011.61 19709999 197099999 19706-11.4ND.1DP.LE.16) 19999 1977 19706-11.4ND.1DP.LE.14) 19999 19999 19999 19999 19999 19999 19999 19999 19999 19999 19999 19999 19999 199999 19		MICRO(25), MONT , STIME , ITIME ,	351
### ##################################			353
### ### ##############################		בערו	391
99999 131-EQ.21 WRITE (T. 201 HICROCII) 131-EQ.21 WRITE (T. 201 HICROCII) 131-EQ.21 WRITE (T. 201 HICROCII) 131-EQ.21 WRITE (T. 201 HICROCII) 131-EQ.21 WRITE (T. 201 HICROCII) 131-EC.01 AND.IDP.LE.10 131-EC.11 AND.IDP.LE.10 131-EC.11 AND.IDP.LE.22 131-EC.11 AND.IDP.LE.22 131-EC.11 AND.IDP.LE.22 131-EC.11 AND.IDP.LE.23 131-EC.11 AND.IDP.LE.23 131-EC.11 AND.IDP.LE.31 131-EC.11 AND.		. (MSR2.EQ.1)	355
# GRAND.IDP.LE.61 15.40.27 1.40.120.LE.61 15.40.27 1.40.120.LE.61 15.40.27 1.40.120.LE.10 15.40.27 1.40.120.LE.10 15.40.27 1.40.120.LE.10 15.40.27 1.40.120.LE.22 15.40.120.LE.22 15.40.120.LE.22 15.40.120.LE.22 15.40.120.LE.23 15.40.120.LE.23 15.40.LE.23 15.40.LE.23 16.60.23 1.40.LE.24 16.60.23 1.40.LE.25 16.60.23 1.40.LE.25 16.60.23 1.40.LE.25 16.60.23 1.40.LE.23 16		.160 T0 99999	336
139-50.21 MRITE if, 201 MICROII) 139-60.21 13-100P 230E NOM = , 02, 16H DUE TO SR2 = 1,) 139-50.20 139-50		(MIS 20(1)	357
15 HQDP 130E NOM = , 02, 15H DUE TO SR2 = 1.) GRO(1) GE.Q.AND.IOP.LE.6) 99999 1977 1076-GE.11.ND.IOP.LE.14) 99994 LT (IOP) 9999 1076-GE.15.AND.IOP.LE.22) 9999 1076-GE.15.AND.IOP.LE.22) 9999 1076-GE.23.AND.IOP.LE.22) 9999 1076-GE.23.AND.IOP.LE.31) 1076-GE.23.AND.IOP.LE.31) 1076-GE.23.AND.IOP.LE.31) 1076-GE.23.AND.IOP.LE.31) 1076-GE.23.AND.IOP.LE.31) 1076-GE.27-AND.IOP.LE.31) 1076-GE.31-AND.IOP.LE.31)		t7, 20) MICRO(1)	23.0
GE.O.AND.IOP.LE.6) N. (IOP) N. (IOP) 199999 1977 1976 1977 1979		FORMAT (13HQOP 330E NOM = , 02, 16H DUE TO SR2 = 1.	656
GE_GLAND.IOP.LE.6) 99996 N (IOP) 99996 RT (IOP) 99997			3 2 2
GE.O.AND.IOP.LE.6) 99996 1000 00-CE-7.AND.IOP.LE.10) 1990 1997 1000 19994 1100 19994 1100 19997 1100 19997 1100 19999 1100 19999 1100 19999 1100 19999 1100 19999 1100 19999 1100 19999 1100 19999 1100 19999 1100 19999 1100 19999 1100 19999 1100 19999 1100 100		TIONAL STATE	385
99998 N (10P1) 19996 19996 19996 19996 199999 19999 19999 19999 19999 19999 19999 19999 19999 199999 19999 19999 19999 19999 19999 19999 19999 19999 199999 19999 19999 19999 19999 19999 19999 19999 19999 199		TTOP OF AMOUND IT AND	363
## (IOP) 99-97 RR (IOP) 99-96 RR (IOP) 99-96 LI (IOP) 99-96 LI (IOP) 99-96 FR (IOP) 99-97 FR (IO			354
997 997 9986 9986 9986 9986 9986 9989 9989 9989 9989 99899 9		CALL BOLN (10P)	365
00-06-7-400-10P-16-10) 19956 181 (10P) 19957 1-1 (10P) 19959 1-1 (10P) 19959 199594 10P) 199595 10P-66-13-4N0-13P-16-22) 199595 10P-66-13-4N0-13P-16-22) 199595 10P-66-23-4N0-13P-16-31) 199595 10P-66-27-4N0-13P-16-31) 199595 10P-66-27-4N0-13P-16-31) 199595 10P-66-27-4N0-13P-16-31) 199595 10P-66-27-4N0-13P-16-31) 199595 1-10P-10P-10P-10P-10P-10P-10P-10P-10P-10		60 10 99997	356
00-66-7-403-100-16-101 997 887 (100) 9995 1.7 (100) 9995 1.8 (100) 9995 1.8 (100) 9995 1.9 (100) 9995 1.9 (100) 9995 1.9 (100) 9995 1.9 (100) 9995 1.9 (100) 9995 1.9 (100) 9995 1.9 (100) 9995 1.9 (100) 9995 1.9 (100) 9995 1.9 (100) 9995 1.9 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100) 9997 1.0 (100)		-10X-) HI 6666	357
9996 997 997 997 997 998 9994 9994 9996 9996 9996 9997 9999 9999		(10P.5E.7.4ND.10P.cE.10)	388
RT (10P) 997 11 (10P) 9984 12 (10P) 9984 13 (10P) 9984 14 (10P) 9984 16 (10P) 9989 17 (10P) 19992 18 (10P) 19992 18 (10P) 19992 19992 19992 19992 19993 19992 19993 1999		. 160 TO 99996	359
997 19 GE.11.ND.1DP.LE.14) 19 GE.15.AND.1DP.LE.20) 99 GE.15.AND.1DP.LE.22) 99 GE.23.AND.1DP.LE.20) 19 GE.27.AND.1DP.LE.20) 19 GE.27.AND.1DP.LE.31) 19 GE.27.AND.1DP.LE.31) 10 GE.27.AND.1DP.LE.31) 11 GP) 12 GE.27.AND.1DP.LE.31) 13 GE.27.AND.1DP.LE.31) 14 GE.27.AND.1DP.LE.31) 15 GE.27.AND.1DP.LE.31) 16 GE.27.AND.1DP.LE.31) 17 GE.27.AND.1DP.LE.31) 18 GE.27.AND.1DP.LE.31) 18 GE.27.AND.1DP.LE.31) 18 GE.27.AND.1DP.LE.31)		CALL INVRT (10P)	310
99.GE.11.1ND.IDP.LE.14) 9995 LI (IOP) 9994 9994 1007 9999 9994 11009) 9995 11009) 11009) 11009 1		2666 01 09	37.1
10P.66.11.1ND.1DP.LE.14) 39395 110P) 39395 110P) 39395 110P) 39391 110P.66.13.4ND.1DP.LE.22) 313996 110P) 310P.66.23.4ND.1DP.LE.26) 313997 110P.66.27 AND.1DP.LE.31) 313997 110P.66.27 AND.1DP.LE.31) 313997 110P.66.27 AND.1DP.LE.31) 313997 110P.66.27 AND.1DP.LE.31) 313997 3139	-	99996 IFf.NOT.	372
99997 10P.GE.13.4ND.10P.LE.10) 99994 10P.GE.13.4ND.10P.LE.22) 10P.GE.23.4ND.10P.LE.20) 99992 10P.GE.27.4ND.10P.LE.31) 99991 11DP.GE.27.4ND.10P.LE.31) 11DP.GE.27.4ND.10P.LE.31] 11DP.GE.27.4ND.10P.LE.31] 11DP.GE.27.4ND.10P.LE.31] 11DP.GE.27.4ND.10P.LE.31] 11DP.GE.27.4ND.10P.LE.31] 11DP.GE.27		. (IOP.GE.11.4ND.IDP.LE.14)	37.3
PUT (10P) 199-56.15.4ND.10P.LE.18) 999-94 10P.66.19.4ND.10P.LE.22) 999-95 10P.66.23.4ND.10P.LE.22) 10P.66.27.4ND.10P.LE.31) 999-97 10P.66.27.4ND.10P.LE.31)		360 10 99995	246
100-66-13-1400.10P-16-18) 109-66-13-1400.10P-16-18) 109-66-13-1400.10P-16-22) 109-96-13-1400.10P-16-22) 109-96-13-1400.10P-16-31) 109-99-93-1400.10P-16-31) 109-99-97-16-16-18-18-18-18-18-18-18-18-18-18-18-18-18-		CALL CAPLY (10P)	316
10P.GE.15.4ND.10P.LE.18) 99994 10P.GE.13.4ND.13P.LE.22) 99995 110P.GE.23.4ND.13P.LE.26) 99992 110P.GE.27.4ND.13P.LE.31) 99991 110P.GE.27.4ND.13P.LE.31) 99991 110P.GE.27.4ND.13P.LE.31) 99997 110P.GE.27.4ND.13P.GE.27.4ND.13P.GE.31) 99997 110P.GE.27.4ND.13P.GE.27.4ND.13P.GE.31) 99997 110P.GE.27.4ND.13P.GE.27.4ND.13P.GE.31) 99997 110P.GE.27.4ND.13P.GE.27.4ND.13P.GE.31) 99997 110P.GE.27.4ND.13P.GE.31) 99997 110P.GE.27.4ND.13P.GE.31] 110P.GE.27.4ND.13P.GE.31] 110P.GE.27.4ND.13P.GE.31] 110P.GE.27.4ND.13P.GE.31] 110P.GE.27.4ND.13P.GE.31] 110P.GE.27.4ND.13P.GE.31] 110P.GE.27.4		101 0	37.7
99994 10P.6E.19.4ND.13P.LE.22) 99995 10P.6E.23.AND.13P.LE.28) 99997 10P.6E.27 AND.13P.LE.31) 99997 118H3INMLL3 OP GODE: , I2, 22H ENGOUNTERED IN EXECH.) 118H3INMLL3 OP GODE: , I2, 22H ENGOUNTERED IN EXECH.)		49450 LT. 100. GF. 13. 4ND. TOP. 16. 18.	37.6
### (###) 9997 9997 100 GC.19.4NO.13P.LE.22) 99993 110P.GE.27 AND.13P.LE.31) 99997 110P.99997		160 00 000	37.9
10P.6E.13.4ND.13P.LE.22) 99993 10P.6E.23.4ND.13P.LE.26) 99991 10P.6E.27.4ND.13P.LE.31) 99991 10P.6E.27.4ND.13P.LE.31) 99997 118HQINNLL3 OP CODE: , 12, 22H ENCOUNTERED IN EXECM.) 118HQINNLL3 OP CODE: , 12, 22H ENCOUNTERED IN EXECM.) 118HQINNLL3 OP CODE: , 12, 22H ENCOUNTERED IN EXECM.) 118HQINNLL3 OP CODE: , 12, 22H ENCOUNTERED IN EXECM.) 118HQINNLL3 OP CODE: , 12, 22H ENCOUNTERED IN EXECM.) 118HQINNLL3 OP CODE: , 12, 22H ENCOUNTERED IN EXECM.)	-	CALL TRING (10P)	330
10P.6E.19.AND.13P.LE.22) 99993 10P.6E.23.AND.13P.LE.34) 99992 10P.6E.27 AND.13P.LE.31) 99997 118H31NALL3 OP GODE: , I2, 22H ENGOUNTERED IN EXECH.) 118H31NALL3 AP GODE: , I2, 22H ENGOUNTERED IN EXECH.) 118H31NALL3 AP GODE: , I2, 22H ENGOUNTERED IN EXECH.) 118H31NALL3 AP GODE: , I2, 22H ENGOUNTERED IN EXECH.)		20 10 99997	391
IOP.6E.13.4NO.13P.LE.22) CIR (10P) 10P.GE.23.4ND.13P.LE.26) 10P.GE.27 AND.13P.LE.31) 10P.GE.27 AND.13P.LE.31) 10P.GE.27 AND.13P.LE.31) 11P.GE.27 AND.13P.LE.31) 11P.GE.37 AND.13P.LE.31)	-	99994 IFt.NOT.	335
999993 10P.GE.23.AND.IDP.LE.26) 99992 10P.GE.27 AND.IDP.LE.31) 99991 FF (10P) 7 129 139 118HAINVALL3 OP GODE: , IZ, 22H ENCOUNTERED IN EXECM.) 118HAINVALL3 NEW COUNTERED IN EXECM.) 118HAINVALL3 NEW COUNTERED IN EXECM.)		. (IOP. GE. 19.4NO.13P. LE. 22)	253
TOTAL TUCK! TOP-GE-23.AND.IDP-LE.26) 99992 TOP-GE-27 AND.IDP-LE.31) 99997 FF (IDP) 9997 999		29499	195
10P.GE.23.AND.IDP.LE.26) 10P.GE.27.AND.IDP.LE.31) 99997 10P.GE.27 AND.IDP.LE.31) 99997 11SHAINVILI3 OP CODE: , I2, 22H ENCOUNTERED IN EXECH.) 12 TAUGE. 13 TAUGE. 14 TAUGE. 15 TAUGE. 16 TAUGE. 17 TAUGE. 18 TAUGE. 18 TAUGE. 18 TAUGE.		כשרר חבמוא נוסה	100
10P.GE.23.AND.13P.LE.26) 99992 10P.GE.27 AND.13P.LE.31) 99991 FF (10P) FF (/6666 DI DO	387
150 TO 9999 151 TO 9999		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	38.8
GGLC 504 (10P) GGTO 99397 -16 TO 99391 -16 TO 99991 GGTO 99997 G		101-101-101-101-101-101-101-101-101-101	33.9
CONTO 99397 IF ** NOT. INC. 100 99397 ** 160 10 99397 ** 160 10 99397 ** 160 10 99397 ** 160 10 99397 ** 160 10 10 10 10 10 10 10 10 10 10 10 10 10		2666 11 1000	39.0
TF. NOT. (100-6E-27 AND. 13P-LE. 31) (ALL DIFF (10P) (AP 09 9997 (AP 07 99997 (AP 10 9997 (CALC SON	391
.10 (100,6E-27 AND.13P.LE.31) .10 (100,9E-27 AND.13P.LE.31) .10 (100) 60 (1	-	10 23331	345
-150 T0 99991. CALL DIFF (10P) CANTUNE FROM TO 19997 FROM TETTE (1. 10) FROM TO 100 TO 100 CONTINUE CONTINUE CALL BINRY ICOUNT = ICOUNT + 1		99995 IF : 91. GC. 27 AND I D. I S. 31.)	393
CALL DIFF (IOP) 60 TO 9997 60 TO 9997 60 TO 9997 FORMATI LIBRALINALED OP CODE: , I2, 22M ENCOUNTERED IN EXECM.) FORMATI LIBRALINUE RAGE. CONTINUE RAGE. CLAL BINNY ICOUNT = ICOUNT + 1		100000	394
COT 10 99997 CONTINUE WARTHE (F. 10) 130 FORMATI (18HAINVALL) OP CODE: , IZ, 22H ENCOUNTERED IN EXECH.) FORMATI (18HAINVALL) CONTINUE TRUE CONTINUE TRUE CLAL BINNY ICOUNT = ICOUNT + 1		TACAC OF DATA	335
MATER (* 10) 130 MATER (* 10) 130 ERGOR (* 1806. CONTINUE CALL BIANT ICOUNT = ICOUNT * 1		101 C0000 04 00	396
MRITE (T. 10) 130- FORMAI (ISHDINALE) OP CODE: , I2, 22M ENCOUNTERED IN EXECM.) CONTINUE RUG. CALL BINK ICCOUNT = ICCOUNT + 1			397
FORMAT (IGHAINVALL) OP CODE: , IZ, ZZM ENCOUNTERED IN EXECH.) CONTINUE TRUE. CONTINUE TRUE. CONTINUE TRUE. ICOUNT = ICOUNT + 1		WETTE (7. 10) 10*	398
ERFOR " "TRUE" CONTINUE CLALL BINK" ICOUNT " ICOUNT " 1		FORMAT (18HGINVALES OF CODE: , 12, 22M ENCOUNTERED IN	339
CONTINUE CONT = ICOUNT + 1	-	ERKOK TRUE.	659
COUNT + 1			177
T TCOCK + T		CALL BINK	70.4
			0.00



	5	Sult Time and													
	35	LOGICAL	u	•	ERROR								24		
		COMMON	CNTHEN (40 24)	1402	41		ICN	•	ICOUNT		TOTA	•	475	2	
	1		IDOOI	•	IDSCR	•	ITELM	•	MAINM (1024)	1024		•	676	•	
2	N M		MBUSS MICO		MENC .		HEHH	• •	MICAU(6)	•	MICDA(3)	31.	67.8		
			Norman		21019		2000	•	MINGTO		THOU M		200		
	rw		MICRO(25)	. 6	. HCMT		STINE	• •	ITINE		IETIN	•	190	•	
	3	COMMON	KAD	-	KON		10,08						164		
10	8	DATA MSK ,	/3778/										482	~	
	TI DODGO TE	INOX = 1											200	h -	
			-	-		-						-			
	90.	160 10 999		A CAN	• NOT SECULLAR DE LAUDANCE - 10 C4	•	16.19	•					964	0	
15	1	IF t. NOT.										-	181	-	-
	•	CNRUNC	(NRUN(2) .EQ.1)	-									864		
	91.	. 160 TO 99	1666				-			-			669	-	
	Z.	MAINME IND											130	•	
	8	666 01 09	9										*		
20	99997 IF (.NOT.	C. NOT.											264	~ ·	
		-	JN (2) . 54. 2)	12.5											
	091.	2	253												
	¥ ₹	MAINM INDX	CNE = ()	200	(NEMD X.	MSK)							964		
23	05	666 01 09	9										64		
	99995 CD	CONTINUE											964		
	HR	WRITE (7,	10) NRJN(2)	JNIC									664	6	-
	10 60	FORMAT (4			SALLED W.	HILH	INVAL	0 03	INVALID OPTION NRUN(2)	NON	= 12	, I3,	1H.) 50	_	
	1	ERROR = .	RUE.										501		
30	99996 CU	7											36		
	09	00 TO 9999	1 . 46										165		
	99998 CO	CONTINUE RETURN											505		
	•														

	1			1				
	DETCAL FIN	TNTS FREDR					519	
	1	ett 0.2	TCN	TCOUNT	UNT	TOTH	910	
		UT . IDSCR	. ITELM	. HAI	HAINM(1024)	. (4)	511	
5			. MENH	. MIC	HICADIE).	MICDA(3).	51.2	
	3 HICG		-	. HSR1	1 .	MSR2 .	513	
	4 NRU	NRUN (2,3, FINIS	. ERROR	. HENST	STR.	HOPNO .	-	
	5 MIC	1. 19	. STINE	. ITINE	HE .	IETIM	515	
	COMMON KAD	, KOA	. IOL 08				\$1.6	
10	DATA REPL / JB	1, REPL	77777620000	0 08/			217	
	-	2036033087					816	
							619	
66666	L			-			926	
	•	NOT. ERROR. AND. IND X. LE. 1024)	ND X. LE. 1024	•			521	
-51	160 10 99998				-		255	
	IF (. NOT.						523	
	(NRUN(1)	0.11			-		176	
	160 10 99997						525	
	CNTHEHILMOXI	2 8:3		1			936	
20	96666 OT 05						527	
49997	1						9.29	
		1.50.21					529	
	100 00 000						63	
	12 - 050	4 4000					531	
	אבי בני - אבי בני							
6	-	2742					766	
							556	
66666	CONTINUE						25	
	7. 1	O) NEGN(T)						
	HOS.	DSETON CALLED WITH INVALID OPTION NRUN(1)	ITH INVALID	05110	A ARCA	,	, 13,1H. 1 536	
30	ERROR = . TRUE.						537	
96666	CONTIN	20502000					956	
		-					939	
#0000	GO TO 99999						341	
566								
2	END						543	20-03- Ex

26666 56666 56666 82 86666 91		545
26666 56666 98 98 98 98 98 98 98 98 98 98 98 98 98	TOTA	946
2	. (42	51.7
266666 56666 98 86666 86666	, MICOA(3),	946
86666 86666 86666 86666	HORES .	
\$6666 \$6666 \$6666 \$7 \$6666 \$7 \$6666	IETIN	188
26666 56666 76666 82 86666		366
26666 26666 28666 28666 28666		222
26666 56666 98 98 98 98 98 98 98 98 98 98 98 98 98		33
26666 56666 98 86666 86666		966
86666 86666 82 86666 83 86666		537
26666 56666 98 86666 98486 98486		33.0
26666 26666 26666 22 26666 27 26666		166
26666 56666 98 86666 02		25
26666 56666 9£ 22 23 24666		28.2
866666 86666 82 84666		986
26666 56666 86666 88466		595
26666 56666 76666 88	AND THE REAL PROPERTY AND THE PERSON NAMED IN COLUMN TWO IS NOT THE PERSON NAMED IN COLUMN TWO IS N	956
26666 56666 98 86666		186
26666 22 26666 22 26666		8:3
26666 26666 26666 22 22		916
26666 56666 82		2.1
CUTTEN (MADR*1) = XOR (MSKI, CNTRL) L L L L L L L C O 9999 GO TO 99993 MITE (7, 20) 20 FORMAT 12 WHENPORT LOAD COMPLETED 1, 15, 1987 20 FORMAT 12 WHENPORT LOAD COMPLETED 1, 15, 1987 20 FORMAT 12 WHENPORT LOAD COMPLETED 1, 15, 1987 20 FORMAT 12 WHENPORT LOAD COMPLETED 1, 15, 1987 99993 CONTINUE 99999 CONTINUE 99999 CONTINUE 99999 CONTINUE 99999 CONTINUE 60 FO 99992		573
26666 26666 27 28		975
26666 26666 26666		976
26666 26666 26666		946
26666 26666 26666		57.9
20 FORHAT IZUMPENET LOAD COMPLETED 1 . 13, 19H 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		591
26868 26868 26868	ROS OF HAIN AND	386
99993 CONTINUE 38 FORMAT (*916 INVALCO * KEY = 9 * REROR = 'FRUE, 99995 CONTINUE 99997 CONTINUE 99997 CONTINUE 99999 TO * 60 FO * 5990		166
39 FDRMAT (1996 INVALE) (EV = 9 (EV =		515
KEV = 9 ERROR = 1FUE 99995 CONTINUE 99997 CONTINUE 99999 FC + 99992 99999 IFC + 100	. 040 C430 -	986
ERROR ERROR OF 00 100 70999 100 70999 100 70999		886
99995 CONTIL 99997 CONTIL 60 FG 60 FG		559
99997 CONTIL		
99997 CONTI		366
99999 IF . NOT.		593
- (MCMC11) - 23- £1		335
		*
.160 TO 99991		537

	40 FORMAT (II,1X,04,1X,03,03,1X,02,01,02,02,04,08)	631
3	. (KEY.EQ.1)	63 t
	MAINH(MADR+1) = 38 (SHIFT (MAINL, 8). MAINR)	616
99		60 8
	-	610
	(SHIFT: (CNTFLO(1),	612
70	CNTRL = OR (SHIFT, (CNTRL, 4), CNTFLD(3)) CNTRL = OR (SHIFT, (CNTRL, 4), CNTFLD(4))	£13 419
	- OR (SHIFT (CNF2L, 10))	615
	CNIME # OK CONT. CONT. CONT. CNIME CONT.	617
75	+ +	618
•	60 10 99967 99986 IF(.NOT.	620
		621
0.0	-	623
•	99965 CONTINUE MRITE (7, 30) KEY	625
99	ERROR = .TRUE.	629
	99987 CONTINUE GO TO 99990	630
	99969 CONTINUE GO TO 99992	632
	99991 CONTINUE MRITE (7, 50) NRJN(11) 50 FORMAT (20H0TMVALTO NRHWILL) =Oli. 184. Blin Terminated.	633 4 G
	ERROR = .TRUE.	636
96	SSS CONTINUE OF THOM	538

			-	NATION.	-	THE PARTY PARTY AND THE ACT	131	680	
	1901901	FINTS						641	
	NUMPOL	-	-	TEN		TEMINT .	TOTAL	66.2	
	Townson .	TOOM		TTEIN		MATUM (4024)		643	
-	•						at Charter	9 49	
	u m	MICO . MSC	MSC PAD (16)			MSR1	MSR2	6.5	
				audas		. desare	CHOOM	828	
	•	251.		STINE		ITINE .	IETIN	249	
	CONNON			10.08				848	
	TECANOT							6.49	
	- 10 mm	5 0 00 C 00 C			1			888	
	TO TO S	O TO SASSA						651	
					1			492	
	I = OF TO							7.0	
	05555 01 09	0.6						199	
12	STONE TARGET		:	:					
		TRITL-LT. J. OK. IBITL-61.47)	IIL .61					623	
	1666 OI 091°	168		,				000	
	IFLAG = 2							759	
	1	96						969	
20	99997 IF(.NOT.							629	
		test tract. 3. or. fettrast. 461	ITHOST	1940	-			959	
	. 160 TO 99996	966						651	
	IFLAG = 3		-		-			269	
	60 10 99998	98						663	
62	99996 IF (- NOT -	1						150	
	÷ .	[TL + 131TN).6T.46)	T. 48)					655	
	•160 TO 99995	\$66			-			989	
	IFLAG = 5							65.7	
		96	-		1			999	
30	99995 CONTINUE							69	
	IPLAG = 0							-	
	IOR = IAOR	+						571	
		•	1			888		7.0	
		SHIFT (MASK(IBITN),		ISTBIT				673	
35	18VTE = A	NO CONTHEM LEDED.	THE SK	×	1			5	
	NABR = 48	E	2					675	
		HIPT (IBVIE, -NHOR	KHOR						
	99998 CONTINUE								
	IF to NOT.								
3	· (IFLAG	(IFLAG.NE.0)						3	
	6 01 091.	***						000	
								160	
	30 NT 1 NOT 96666				CALL US				

	SUBROUTINE DCODM	689
	LOGICAL FINIS , ERROR	
	. TON .	· NIO
	1 IDOUT , IDSCR , ITELM , MAINM(1824)	
5	Z NBUSS . NEAC . NENH . HICAD(6)	. MICDA(3),
	, MS3PAD(16) .	
	4 NRUN(2:), FINIS , ERROR , HINSTR	
	S) . MCNT .	• IETIM 692
	KDA	
1.0	OFF /14008/	169
-	3	
		969
	Ŧ	469
	m MTM	869
7		659
23		
	5	
	2	15:
	II	77
	CALL SETHH (MENC, 8, 8, NOP, IFLAS)	
50	MOPNO = MOP	
	5	216
	96666 0	
		• ; ;
53	60 10 99997	
	99998 CONTINUE	710
	WRITE (7, 10) IFLAG	
	10 FORMAT (43HGINVA_CD CALL TO GETMM FROM DCODM, IFLAG	= , I2, 1H.)
	ERROR = .TRUE.	713
30		716
	00 TO 99996	574
	99999 CONTINUE	716
	MRITE 17, 10) IPLAS	
	ERROR = .TRUE.	917
35	99996 CONTINUE	226

	INE GETHH	TINDE, IBITL, IBITN, IBTE, IFLAGI	IBYTE, IFLAGI	224
	_			653
	COMMON CNIMENTICES	TTELE	MATCHETT .	72.4
-	-			736
•		0(16)	MSR1 . MSR2 .	722
	1633			728
	-	. STIME .	ITINE , IETIM	624
	CONNON KAD , KDA	10,08		750
10	IF(.NOT.			731
	ĸ	1023)	to the second	732
	99999 · 160 TO 99999			733
				136
				(35
57	99999 IF L. NOT.			952
	. (IBITL.LT. D. DR. IBITL.ST. 15)	L.5T. 15)		15
	199997			220
	STERS = E			
26	99997 IFC.NOT.			762
		N.67.16)		742
	0			743
		The second secon		3.2
63	99990 IFTENDIS	151		27.2
		167.		7.60
				6.2
				750
30	99995 CONTINUE			751
	6			155
	TOTAL TAUK + 1			562
	IMASK = SHIFT (MASK(IBITN) - IBIT)). IBIT)		755
35	E AND	IMASKI		73.6
) - 91	2		757
		BRI		956
	99998 CONTINUE			967
77	LTICACIO.			751
2	150 10 4444			
				763
	99994 CONTINUE			184
•	RETURN			7 7 50
42	ZNO			00.

	SUBROUTIN	1	TOPI			1				2.6	
	LOGICAL	STATE	TO ERROR		200	-	TOOM	101		752	
	1		TOSCR	• •	ITELM		MAINMELD	(1324)		77.0	
	- 2	MBUSS	. MENC	•	HEMH		MICADIST.	. MICDA(3)	A (3) .	77.1	
	•	MICO	. MSCPAD	9110	-	•	HSR1	HSR2	•	772	•
		MRONE	200	•	CERCOR	•	HENSTR .	TETT		776	
	CONNON	KAD	ACX.	• •	10.00	-	-			773	
10	DATA LHI	SH /17777	781							77.5	
	S FORMAT (FORMAT (27HBSUBROJINE BOLN	JIINE BO	10	ENTERED.				-	718	-
		CROLES				1		-		524	-
		CRO(3) +								750	
57	18K = #1	HICRO(4) +		-	-	-				162	
•	101.08 =	IOLOB = MSCPAD(132)	2	-		1			-	787	
	(TOP-EQ. 0)	£0.01								784	
	01 001				-	1				566	
20										796	
	. tICH.EQ.0)	EQ. 0)	-	-		-				2	
	.160 10	96666								798	
	ITHP = LHIGH	HIGH	-			-	-	-		662	
		166								21	
62	9998 IP1. NOT.	TON FO. 11								747	
	9666 01 094-	9666	-	-	-	1		-	-	733	
	ITHP = 0									734	
		166				-		-		56.	
30	99996 CONTINUE									133	
	MRITE (7. 10)	TAN ICA	40I %								
	16 FORMAT (5	ID CARRY		102 , 00	*	805 N. OP	€ 3000	. 12v 1	-	
	99997 CONTINUE									930	
-	60 10 9	566								160	
*	SASSA TELEMOTE	1								360	
	46969 TO 99994	7666								36	
	TEMP = 1	TEMP - AND INSCREDITARY. MSCPADITORY	OLTAR!	#SCP	A DET BR	+	-	-	-	618	
•	CO TO 99995	366								836	•
	39994 TPf. NOT.		1	-	-		-	-		18	
	2	(10P. EQ.2)								8	•
	2666 01 001	20 10 99993	0 00001	100							
	DUN - JULY	MAC 1014	Hackan	6		-		-	-		
	99943 IF (MOT.									3 6	
		110P. Eq. 31	-		-	1	-	-	-	14	9
	2	26661				. 3				16	•
		OF THECPASTIARIS HECPADISERS	TARRE .	5	101388	-		986	100	410	
20	CREER 101 09	295	-			-	-			1	
		(10P. EQ.4)									
	16666 01 0911	16661	-	-	-	-	-	-		10	
	ITMP = 0	OR CIDIN, MSCPADCIBRY	45CPAD (1	8	-					629	
90	99991 TE (. NOT.	-	1000000	0			100			82	
		Contract of the last of the la									

N. S.	091.	0 10 99990						429	
	11	P = X02	(IAR)	(MSCPAD(IAR), MSCPAD(IBR)				629	
90	3	60 10 99995			-	-		939	
	99990 IF (.NOT	C. NOT.						827	
	•	(10P.Eq.6)	-					0.0	
		8666 01						6 3 0	
1 12	-	8	(IDIN, HSCPAD(IER)	Leki				0.60	
65	99989 CONTINUE	NTINGE						631	
	99995 CI	NITHOE						269	
	7	TE (NO ! .						200	
		INRUNIS) .EQ.21						400	
	•	66						650	
94	25 FO	FORMAT (THOINSTS .	02.	H EXECUTED.	24H EXECUTED, TEMP VALUE =	.90.	5. 1H.)	837	
								63.6	
	11	IF(.NOT.						639	
	•	(MOD. EQ. 7)						0+8	
75	31.	.160 TG 99987						148	
	=	===						700	
	#	IDOUT = ITHP						238	
		60 TO 99986							
	33301	9						4	
3	2	CALL INHON (TAR. 192)	38)					64.4	
	99984 Chullan							848	
	33300 C	RETURN						6 4 8	
	END							930	

	TINE INVRT (10P)		991
	COMMON CATACACTURES ICA . I	ICOUNT, IDIN	953
	S . MENC . MENH .	9	959
The state of the s	. MSSPAD(16)		836
	MICROLESS, MICAT , STIME , I	ITINE . IETIM	83.6
	COHHON KAD , KDA , 100.06	Constitution of the consti	650
	IF INRUNIS , EG. 21 WELTE (7. 5)	the state of the second	631
•	FORMAT (27HOSUBROJTINE INVRT		952
77	TAB - HICRO(2)		699
	186 = HICRO(6) + 1		* 100
	IOL08 = MSCPAD(132)		936
	1ft.NOT.		-8:
	110110 10 10000		000
20	0		970
	96666 01 09	The second secon	97.1
66666	99999 IF(.NOT.		872
	. 100 TO 99997		874
	ITHP - COMPL (IDIN)		975
TA STATE OF THE ST			976
16666	99997 IFf. NOT.		***
	160 10 00006		
	ITHP = COMPL (MSCPAD(I3R))		930
90000	00 TO 99996		
	(10P.EQ.10)	The state of the s	893
	.160 TO 99995		486
90000	SOSOS CONTINUE	the constraints the same of the constraints of the	846
86666	CONTINUE		168
	ITMP = ANDLITMP, LHIGH)		88
	Fft. NOT.		660
3	· (ICM.EQ.0)		969
			632
			993
	GO TO 99993		169
*****	STORY LT. MOT. ST. 13		A35
	-160 TO 99992		697
	401 .NCI (9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	969
=	WALLED CARRY	,00, £14 IN INVRY, OF CODE - , 12, 14,	***
16	CONTINE		256
99993			915
-	1F1 -NOT.		93
	160 to 10001		

15666	916 911 911 911 912 915 915 916 916 919
	911 912 913 915 916 918 919
35666	91.5 91.6 91.6 91.6 91.9 91.9
35666	916 919 919
CALL INHOD (IAR, IBR)	0 16 6 16
99989 CONTINUE RETURN	

	SUBROUTINE CHELT (139) LOGICAL FINIS , ERROR COMMON CNTHEM(1024) , ICN , IDOUT , IDSCR , ITELH , HARRS , MERC , HEM		921 922 923 924 924
	MICQ , HS2PAD(16) NRUN(2:), FINIS , MICRO(25), MCVT ,	, HSR2 ,	926
	ION KAD , KDA , ILMISH /177778/ INTUNEST :EQ.27 WRITE (7, 9		929 930 931
	MOD = M12/CR0(3) + 1 ISR = MICRO(4) + 1 IQLOB = MSCPAD(132) IFF.NOT. (100-Eq. 11)		935
56666	.160 TO 99999 IIMP = COMPL(MSC)A)(IAR)) 60 TO 99996 10 TOP-Eqc12)		6 2 2 2 8 3 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
16666	ITHP - COMPL (131M) 60 T0 9998 60 T0 9998 (100 - EQ.13) 60 T0 99996		₩ 9 K 9 6
96666	ITHP = COMPL (MSCPAD(IBR))		950 950 950 950 950 950 950 950 950
1666	J CONTINUE ITHP = AND (ITHP, LHIG4) ITHP = AND (ITHP, LHIGH) IF (LNOT, CRUCK)		958
\$58 \$888	HRITE (7 HRITE (7 FORMAT (CONTINUE IF (NOT	.UE = ,06, 14.)	6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
66666	2 1 2 2 2 2 2 3 3 5 5 5 5 5 5 5 5 5 5 5 5 5		99 97 97 97 97
26666	CONT INUE		37.6

SUBROUTINE FRICE LOGICAL ETINIS : EROR COMMON INSERT : MICOS : MESPADLIS : MESPADLIS : MESPADLIS : MESPADLIS : MESPADLIS :
--

•	SUBROUTINE	DECTR 11	140				1635	
•	LOGICAL	FINIS	ERROR				1036	
•		IDOUT	IDSCR .	ITELM	MAINM(1824)		1038	
	- 2	MBUSS .	METC	MEMH	MICAD (6) .		1039	-
		MICO.	MSS PAD (1	19	HSR1 .		10.01	
	•	NRUNIE: 1 .	PINIS .	ERROR	HINSTR .	HOPHO .	1401	
	5	MICRO(25)	. HCAT	STIME	. ITIME .	IETEN	10,5	
	DATA LITER	KAU 41777793	. KUM	100.08			7,00	
	TE CADIN	TE CADINET ED 21 APTE CT	41 5410	-				-
	5 FORMAT ((27 HOSUBADUTINE DECTA ENTERED.	INE DECT	ENTERED.	-		10:6	
		HICKOLES				-	1001	
San San Strate S	IAR = MI	MIC20(3) + 1					1048	
13	18R = MI	MICROTAL . I	-	-			1049	
	101.08 =	= MSCPAD(134)		-	-		1050	
	CIOP	(TOP.F3.19)					1021	
	160 10	66666				-	1053	
36	ITHP = H	ITHP = MSCPAD(IAR)					1054	
	60 16 9999	966	-	-			1035	and the same and the same
56666	99999 IF 1. NOT.						1036	
The same of the sa		110P. EQ. 201				-	1057	-
THE PERSON NAMED IN	2	1668					1058	
62	NICT - CHAI	NIOI NIOI					6697	
9000	100 101 100 PE	930		-	-	and the same of th	1001	-
		110P. EQ. 211					1052	
	.160 TO 99996	9666		-			1063	-
30	ITHP = M	ITHP . MSSPAD(184)					1991	
90000	60 10 99996	966					1965	
		(10P. EQ. 22)				Control of the last of the las	1691	
		5666					1039	
35	TTHP =	MISQ	-				1059	-
55666	77.0						1070	
6666							141	
	IF C. NOT.						1072	
	150 TO 00004	10000					141.9	
	-	THP . LATS.		- Company	-		5440	
	ITHP = AP	E	HIG41				1075	
		183					197	
76666	Te (. NOT	10 10 101 et 9 11 10					1078	
	.150 TO 99992	3992	17.10.101				6 601	
-	WRITE (7.	. 16) ICA. 10P	401				1601	
77			CARRY .	,08,21H IP	DESTR. OP	,08,214 IN DESTR, OP CODE = ,12,14.1		
26666 05	3 CONTINUE						4601	
	IF to MOT.						1895	-
	. CHRUN	RUN(3).69.21					9607	
	. 160 10	94991	1140				1691	
*		- C3 131.		CKCOMPCA	See frequited from use me	40	9507	
19991				-		1001	1036	

SUBROUTINE DECTR 74/74 OPT=1	275 A 45 C 1 4 5 C 1 4
. (HOD: E2.7)	103
ALCO - 114P	1681
IDOUT = ITHP	2601 C601
99990 CONTINUE	1037
CALL INHOD (IAR, 13R)	6601
99989 CONTINUE RETURN	1011

CORPORT CONTRACT		NE SUM (10P)		1113
The control of the	-	Cutues 11261	2010	1135
### ##################################		TOOUT TOSCE TIFIN	1241	1116
# # # # # # # # # # # # # # # # # # #	- 5	. MENG . MENH .	. MTCDACTI.	11117
### ##################################		MICQ . MSSPAD(16)	. MSR2	1136
5 5 5 5 5 5 5 5 5 5	-	12.3, FINIS , ERROR ,		1119
DUTA LHIGH /17778 / 100.08 DUTA LHIGH /17778 / 100.08 DUTA LHIGH /17778 / 100.08 FORMAI (27483083/JINE SUM ENTERED.) INR = HIGROIN : 1 INR =		MICROLEST, MC47 , STIME ,	. IETIM	1119
STATEMENT STAT		ON KAD . KOA .	The second secon	1111
FORMATINE TOTAL NETTER	10	LHIGH /1777783/		1112
The control of contr		TO THE UNIST OF COST WRITE IT.	And the second s	1113
TAR = 41280(13+) TAR = 4		FURNAI (27 HUSUBAUSITAE SUN		*111
100-62.33 100-62.33 100-62.33 100-62.33 100-62.33 11MP = HSCPAD(183) + MSCPAD(18R) + ICN 100-62.23 11MP = HSCPAD(183) + MSCPAD(18R) + ICN 100-62.23 11MP = HSCPAD(18R) + MSCPAD(18R) + ICN 100-62.23 11MP = MSCPAD(18R) + MSCPAD(18R) + ICN 11MP = MSCPAD(18R) + MSCPAD(18R) + ICN 11MP = MSCPAD(18R) + MSCPAD(18R) + ICN 11MP =				6111
10 to 0	-		the state of the s	1116
117.001. 100.00.231 11	:	TOTAL BEST OF THE PROPERTY OF		****
Tido 99999 17th		Total Total		
1100 99999 1100				1120
114P = MSCPAD(1A2) + MSCPAD(1BR) + ICN 1160	1	50000 00 0000		
99999 Ff. NOT 160 TO 99998 160 TO 99997 170 - 110 - 120 -	36	MSCDANITASI + MSCDANITADI		1133
99999 JF(.NOT. - 160 TO 99990 - 160 TO 9990 - 160 TO 990 - 160 TO 90 - 160		National Party of the Party of		1151
100 100				4154
99997 IFT : 101 N SCPACOLIBRI * ICN GD 10 99996 110			The second secon	
ITT		110 To agas?		1123
GD TO 99994 GD TO 99994 150 TO 99996 150 TO 99996 If he = NSCPADILIAR) + MICQ + ICN GD TO 99995 160 TO 99999 160 TO 99995 160 TO 99992 160 TO 99998				11:0
99997 IFT.NOT. 110P = MSCPAD(IAR) + MICQ + ICN 11MP = MSCPAD(IAR) + MICQ + ICN 99996 GT		THE HALL BEEN	(1311
11		1		1129
1700 TO 99996 1700				1133
ITHP = NSCPADITAR) + MEQ + ICN 100 TO 99995 100 TO 99995 100 TO 99995 11	-	96666 01-091	The second second second second second	1131
99996 GF (NOT. 60)-60,-26) 100 10 99995 110 10 10 99995 110 10 10 99995 110 10 10 99994 110 10 10 99994 110 10 10 99994 110 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 99995 110 10 10 99995 110 10 10 99995 110 10 10 10 10 10 10 10 10 10 10 10 10	30	DUIAR! + MECQ		1132
99996 Iff.NOTe. 150 T0 99995 150 T0 99995 150 T0 99995 150 T0 99995 150 T0 99999 150 T0 P9999 150 T0 T0 P9999 150 T0 P999		_		1193
150 TO 9995 170		IF (. NOT.		1134
99995 CONTINUE 99995 CONTINUE 11th = 101N * H132 + 15N 99995 CONTINUE 15 NO (ITH** LHISM) 15 NO (ITH** LHISM) 15 NO (ITH** LHISM) 15 NO (ITH** NO (ITH** LHISM) 15 NO (ITH** NO (ITH** NO (ITH** NO (ITH**) 15 NO (ITH** NO (ITH** NO (ITH**) 15 NO (ITH** NO (ITH** ITH** 15 NO (ITH** 15 NO		. (10P. Eq. 261	-	1135
99995 INTO = 101N + MISS + 1CN 99995 CONTINUE 17 = AND (ITH-, LNISH) 17 = AND (ITH-, LNISH) 18 = TO TO 99994 18 = AND (ITH-, LNISH) 18 =				1136
99995 CONTINUE 999995 CONTINUE 1	35	ITHP = IOIN + MISS		1137
### 17 17 17 17 17 17 17 1				1138
If the = AND (ITM*, LMISH) If the = AND (ITM*, LMISH) ** (NRUN(I) - EQ - 21 ** (NRUN(I) - EQ - 21 ** (NRUN(I) - EQ - 21 ** (NO - EQ - 29) ** (NO - EQ - 29) ** (NO - EQ - 29) ** (NO - ETHP ** (NO - ETHP ** (DOUT - ETHP		Ľ		1139
I NOTE NOT		ITHP & AND (ITHO). LAISH)		1163
## 150 TO 99994 ## 125 107, ITHP 29 TORNAT (THEINST , DE, EWH EXECUTED, TEMP VALUE = ,067 _ 141,] 9999, CONTINUE		16.03.15.00.15.		2011
### (7, 25) 10P, 17HP 25 PORMA! (7401NS14 , 06, 24H EXECUTED, 1EHP VALUE = 1067 1141) 27 PORMA! (7401NS14 , 06, 24H EXECUTED, 1EHP VALUE = 1067 1141) 39994 CONTINUE 39995 CONTINUE 39999 CONTINUE 39999 CONTINUE 39999 CONTINUE 39999 CONTINUE 39999 CONTINUE 39999 CONTINUE	-	160 TO 99994	and the second s	
25 FORMAT (FMBINST .)0; 24H EXECUTED. TEMP VALUE . ,06r 14t.) 9999, CONTINUE 17 (MOD.EQ.7) 10 10 99992 17 (MOD.EQ.7) 18 (MOD.EQ.7) 18 (MOD.EQ.7) 19 (7. 241 103		34:
99994 CONTINUE ###		PHOTONSTA	- 1961	2411
# # # # # # # # # # # # # # # # # # #		CONTINUE		1166
. (MOD.EQ.7) .160 TO 99993 MCG = ITHP EDOUT = ITHP EGPATINE MSCPAD(I3R) = ITHP CALE EMBOO (IAR, I3R) 99992 CONTINUE	- 51			111.7
## 150 TO 99993 ### ############################		. (MOD.EQ.7)		1158
HCCQ = ITMP 1001 - 11MP 60 10 9992 MSCPAD(13R) = ITMP CALL IMMOD (1AR, 13R) 99992 CONTINUE				1119
1001 - 11MP 10 10 9992 10 9992 10 9993 CONTINUE 11 10 10 10 10 10 10 10 10 10 10 10 10 1		-		1150
GD 10 9992 9993 CONTINUE NSCPAD(I3R) = II4P CALL FAMOD (IAR, I3R) 99992 CONTINUE		10001 - 11HP		151
CONTINUE MSCPACISR) = IT4P MSCLE FWOO (IAR, E3R) CONTINUE	20	60 10 99992		1152
MSCPDG(IR) = IT4P CONTAND (IAR, I3R) CONTAND		_		1153
CALL EMMOD (IAR, IBR)				1154
CONTINUE				1155

	110	6617
	CORRON CRIMENTARY ICH . ICOUNT . ICOUNT	1150
	IDOUT . IJSCR . ITELM . MAINMIID24	1152
	S MEUSS , MERC , MENH , MICAO(65, MICDA(3),	1165
	(2.3, PINIS . CRROR . HINSTR .	6911
	STIME . ITIME . INTER . INTER . INTER	1156
	HIGH /1777/63/	
	IF THRUN(31.EQ.21 HRITE 17. 9	1189
•	FORMA	11/0
	IAR = MICRO(3) + 1	1172
	-18k = #10R0(4) + 1-	1173
	IOLOB = MSCPAD(I3%)	2
	. (IOP.EQ.27)	1176
	1	-
	MOUNT AND THE PROPERTY AND THE PROPERTY OF THE	977
66666	99999 IF (.NOT.	1130
	- 150 TO 90007	1111
		-
	8666	1697
15666	19997-IF CANT.	1195
	**************************************	1136
	ITMP = IOIN + ANJCOMPL (MSCPAD (IBR)), LHIGH) + ICN	1188
98666	60 TO 9996 99956 IF(.NOT.	1130
	- (100 at 0.00	1611
		1192
	9666	1617
estate.	. (10P.Eq.31)	1136
		1137
- 0000	CARTILLE SOIN + ANJACONFLATCOL, LAIGH) + LON	1138
96666	CONTINUE	1530
	1747 - 440 (1747, . #164)	-
		123
	66 0	1884
36	FORMAT (7-10THST) . 17- 24H EXECUTED. TEMP VALUE = .06. 1H.)	1216
2000	CONTINUE	181
	IFC.NOT.	6021
	.160 10 99992	1210
THE RESERVE OF	ACC = 17HP	rat.
	to to the	Ich
26666	CONTINUE	121
	+8CP40(19K) = 114P	444
16666		
-		1171

	74000 370400 0013	****
	ERZOR	1221
	CNTHEN(1024) . ICN .	183
	1 IDOUT . IDSCR , ITELM , MAINW(ID24) .	1223
	. MSCPAD(16) . MSR1 .	1225
	12:3 , PINIS , ERROR , HINSTR ,	144
	S MICAO(25) MCAT , STIME , ITIME , IETIM	1227
16	CRO(6).EQ.13 MSR1 = 0	1229
-	1.	1230
	S	1231
	.160 to 99999	1211
15		1234
	99999 CONTINUE	1235
	iff c.NOT.	1236
	3	1237
20	MBUSS = NOT (OR (MICADIKAD), IOSCR))	1239
	Section 1	1240
	99998 CONTINUE	1771
	IF CHICRO(15)-E3-11 MEUSS # OR (MBUSS, MSR1)	2421
	TO T	
0		1245
		154
	704 = 1	1247
30	99997 IF (. NOT.	1249
	. tHEOROTE17-53-17	1631
	2	1521
	CD TO 9030A	1253
35	99995 IF (:NOT.	1254
	. (MICRO(221.62.1)	5521
	#180 10 9999¢	1257
	9666 01 09	1250
,,	99954 CONTINUE	1239
-	poppe Continue	125
		201
	. (MCRO(23).Eq.4)	1263
2	. 160 to 99993	1927
-	KAD = MICRO(25)	1255
	99993 CONTINUE	1257
	KETUKK	1650

DOUGHOUNG THEN FROR 170m 170m 1770			
UGUICAL FINES EROR ICA ICANIT IDIN		BINKY	0.23
CONTROL CHITCLES ZCH ZCONT ZCONT		FINIS , ERROR	1271
2		CHIMEN(1024) . ICM . ICOUNT .	2.21
STATE STAT		. IJSCK . LIELM .	16/3
SCHOOL SCHOOL STATES SCHOOL STATES SCHOOL STATES SCHOOL STATES SCHOOL STATES S	•	THE PROPERTY OF	100
MCMCACATA MCMC		The state of the s	
		MENT STIME TITME	122
ONT A 100 14		101 08 0 101 08 08 0 101 08 08 0 101 08 08 0 101 08 08 0 101 08 0	****
	77	111137	1279
			1941
F (MERCATITECALT) MSKE = MSKE		11) . E3.1) MBUSS	1521
		710EG. 13 HSR2 =	1835
HIGGOLD 160			1293
######################################	61	. [HECROIDS.EQ.13	1631
### ##################################			1235
GG TO 99999 CONTINUE FINANT 1214 H-RES LOAD BET NOT SET AT LOC . O4.154 - RUN HALTED. 1 FINANT 1214 H-RES LOAD BET NOT SET AT LOC . O4.154 - RUN HALTED. 1 FINANT 1214 H-RES LOAD BET NOT SET AT LOC . O4.154 - RUN HALTED. 1 FILE OF THE OF SET AT RES LOAD BET NOT SET AT LOC . O4.154 - RUN HALTED. 1 FILE OF THE OF SET AT RES LOAD BET NOT SET AT LOC . O4.154 - RUN HALTED. 1 FILE OF THE OF SET AT RES LOAD BET NOT SET AT SET AT CONTROL SIT 35. 1 FILE OF THE OF SET AT CONTROL SIT 35. 1 FILE OF THE OF SET AT SET ON ROL SIT 35. 1 FILE OF THE OF SET AT SET ON ROL SIT 35. 1 FILE OF THE OF SET AT SET ON ROL SIT 35. 1 FILE OF THE OF SET AT SET ON ROL SIT 35. 1 FILE OF THE OF SET AT SET ON ROL SIT 35. 1 FILE OF THE OF SET AT SET ON ROL SIT 35. 1 FILE OF THE OF SET AT SET ON ROL SIT 35. 1 FILE OF THE OF SET AT SET ON ROL SIT ON ROL SIT SET AT S			1256
### CONTINUE FIRE		10 9	1297
	•	Jin Jack	
### ### #### #########################	20	N(3) .EG.2) MRITE (7	1289
#ERN = 0 #EROR = .TRUE. ##EROR = .TRUE. ##ERO		CALM Nº RES LOAD BET MOT SET AT	9621
### FROM ## TRUE. ** SERVER ## TRUE. ** INCORDINGUE 1			1231
ERROR = .TRUE. 156. TO 99997 18 - NEW 0 99999 18 - NEW 1 99999		161 E	484
99999 CONTINUE If .NOT. . 160 TO 99997 N			1233
If wor.		1	
** ** ** ** ** ** ** ** ** ** ** ** **			1235
150 TO 99997		(NCRO(9) -FO.13	1206
		. 160 10 99997	1237
IFT.NOT.			1238
### (WRUNTS) - EQ-21 150 TO 3999 WRITE (12 91 MEND, MAINHIN), HBUSS FORMAT (17 HBWAIN MEND, MAINHIN), HBUSS FORMAT (17 HBWAIN MEND, MENN, ME. 01, 14H CHANGED FROM, 06, 4H TO 0 99996 CONTINUE FF. NOT CONTINUE FROM TIME (17, 10) MEN FF. NOT MENN TIME (17, 10) MEN FF. NOT F	30	IF C. NOT.	1239
STRING 39996 STRING ST		· thrunist .Eq.t.	1530
######################################		96666	181
5 FORMAT (17 HOMAIN MEMORY LOC, 04, 14H CHANGED FROM, 06, 4H TO, 99996 CONTINUE 99996 CONTINUE 100001121-11-1005 1000001121-11-1005 100001121-11-1005 1000001121-11-1005 1000001121-11-1005 1000001121-11-1005 100000000000000000000000000000		#BUSS	138
99996 CONTINUE 06, 19H 57 CONTROL 517 35.) 99996 CONTINUE 48093 99997 CONTINUE 48093 151-180 TO 99999 15-180		. 04, 14H CHANGED FROM , 06, 4H TO	1333
99996 CONTINUE 99997 CONTINUE - 16. NOT 160 TO 99995 - 150 TO 99995 - 150 TO 99995 - 160 TO 99996 - 16	-	. 06, 19H SY CONTROL SIT 35. 1	1334
99997 MITHER HOUSE - (MICROLIZ) - E3.1. AND. MENH, ME. D) - (MICROLIZ) - E3.1. AND. MENH, ME. D) - 160 TO 99999 - WRITE (7, 10) MENH - 17 MEN - TRUE - 99999 CONTINUE - INTERCOL - TRUE - 99999 CONTINUE - INTERCOL - TRUE - 160 TO 99999 - 17 LEDOT - E4.1) - 160 TO 99999 - 17 LEDOT - E4.1) - 160 TO 99999 - 17 LEDOT - E4.1) - 160 TO 99999 - 17 LEDOT - E4.1) - 160 TO 99999 - 17 LEDOT - E4.1) - 160 TO 99999 - 17 LEDOT - E4.1) - 160 TO 99999 - 17 LEDOT - E4.1) - 160 TO 99999 - 17 LEDOT - E4.1) - 17 LEDOT - E4.1) - 180 TO 99999 - 180 TO	6		1305
99997 CANTINUE (FIG TO 12) - E3.1.AND. NENH. NE. D) (FIG TO 12) - 99995 (FIG TO 13) - FETCH - BIT ON BUT H-ADDR - , OG. 144.) (FIG TO 13) - FETCH - FETCH - BIT ON BUT H-ADDR - , OG. 144.) (FIG TO 13) - FETCH - FETCH - BIT ON BUT H-ADDR - , OG. 144.) (FIG TO 14) - FETCH - FETCH - BIT ON BUT H-ADDR - , OG. 144.) (FIG TO 15) - FETCH - FETCH - BIT ON BUT H-ADDR - , OG. 144.) (FIG TO 19994 -) - FETCH - BIT - BIT ON BUT - BIT		Ī	263
### ##################################	*	1997 CONTINUE	1337
. (MICRO(12).E3.1.AND.MENH.NE.0) WRITE (1. 10) MEA WRITE (1. 10) MEA LE PORMAT (34HERR341 FETCH BIT ON BUT N-ADDR - , Ofg. 2H.) ERROR = .TRUE. FRANCE (100011).EQ.1) . 160 TO 3999. IF .NOT. ON INC. (100011.Eq.1) - 160 TO 3999. IF .NOT. (150011.Eq.1) I	-		1386
##ITE (7, 10) ##34 ##ITE (10001-1001) ##ITE (10001-1001-1001) ##ITE (10001-1001-1001-1001-1001-1001-1001-10	94	. (MICRO(12).62.1.AND.MEMN.NE.D)	1309
### ##################################		5,00 10 99995	1361
16 PARMAT 134HEERDAY FETCH BIT ON BUT H-ADDR = , Off ENt.			1311
ERROR = .TRUE. 1		T CHARGERED CO PETCH BYT ON BUT	13.6
9999 CONTINUE If (NOT) 160 TO 39994 1		= .TRUE.	1313
			1314
### 19994			1315
9999 DT 10007-64-81 4381 = 1 9999		1.0001111.00.1	1316
10 - 110000-20-01 4381 = 1 9999- CONTINUE (MC CRO(16). E 2.1)		350 10 3004	1317
99994 CONTINUE TF1:NOT: - NACRO16)-E3.1) - 160-TO 99993 NEMH - OR (MICRO(5), MSR1) - SECHALL SEC. 22 MRSR1 - OR 1			
TOT: NOT: (MCRO116).E3.1) - 166 - 19993 NENN = OR (NICRO15). MSR1) S. FROMAT (2948MF A. MR16 F. R. NO. MCM.		CONTINUE	1319
. (MCCRO(16).E2.1) .160-T0 3993 .WEN = OR (MCCRO(5). MSR1) .F ANRWIZI.EG.21 MRIE 17, 301 MENN .FORMAT (29MANE 4 MOTEFER AV MSB1 TO: 0.04)			
			121
MENN = OR (NICRO(5) MSR1) 181 MENN FARM (2) MANEY A MOTETER AV MSS TO . OR)	1		•
FERNAL COMMENT A MONTETED BY MENT TO . ON 1		9	1331
FORMAT (2940NEXT 4 HORIFIED BY MAR! TO: . OF !	*	Res 2	1331

SUBKOULT	SUBROUTINE BINKT	14/14	OFIEL		CCCLIANCE NIL		
	41	THE CROSSON.	£2.4)	HENC = AND (MBUSS, MSK10) ITELM = MBUSS	3 (30)	1328	
99	IF.	(KDA.NE.0)		IF(.NDT.		1330	
	09(°	TO 99992	18.35			1332	
65	404 99952 CON	TINUE			1 - 60 1 - 63 1	1334	
	RETURN RETURN	TNRUN(3).EC	121 21	EQ. 21 CALL REPT1		1336	
	ENO		P. C.			1931	

												-	
	SUBI	SUBROUTINE STA	STATS									338	
	1007	LOGICAL F	FINIS ,	, ERROR	~				1			1339	
	COMMON		CNTHENT	10241		ICN	151	ICOUNT .	TOTA	•		1340	
	1		I DOOL	IDSCR		ITELH	. MA	MAINM(1024)	(3:	. 1323		1341	
	2		MBUSS	NENC		MEMH	. HI	340 t61.	MICADIES. MICDALS	(3),		1345	
	m		MICQ .	. MSCP	MSCPAD(16)		. MSR1	. 12	MSR2	•		343	
			NRUNIES	PINIS	000	ERROR		MINSTR .	HOPHO	•		1	
			MICRO(25) . MCNT	S) - HCNT	•	STIME	. ITI	ITIME .	IETIH			1345	
	COMMON		KAD	ACX		101.08						3+6	
10	WRIT	.23		ME, ICO	UNT, H	CNT. M	C PAD.	IDIN.	IDOUT,	STIME, ICOUNT, MCNT, MSCPAD, IDIN, IDOUT, MBUSS,		1347	
		1	1	NE NE	. MICO	. HICAL	P. MICH	DA . 151	1. LOSCR	. MSRI .		1348	
	18 FOR	18 FORMAT (25H1RU	HIRUN T	RHINAT	ED AT	TIME :	9F15	.8.15H	SECOND	IN TERMINATED AT TIME : "F15.8,15H SECONDS AFTER		13,46	
		18	TH HILL	MICROS OR . 19,17H HACROS EXECUTED.	. 15.	17H MAL	KOS E	recoret	1	-		32	
	•	211	21HOSCRATCHPAD CONTAINS:,/1H , 16(/5X, 08),	CHPAD C	ONTAIN	S1,/1H	, 16(15x, C	19:18			1351	
	•	111	LIHODATA	N	08 . L 3H	, DATA	= 100	. 08	9H. BU	TA IN = , 06,13H, DATA OUT = , 06, 9H, 8USS = , 06,	8.	1355	
		101	H. C-REL	40° = 5	. 164.	H-REG =	. 00.	10H. Q.	-REG =	10H. C-REG = .04.16H, H-REG = .04.10H, Q-REG = . 06, //		1353	
		46	0470 S	512x	1. 180.	0H. D/	5 =	,312×.	11.181			1354	
		111	11HO CARRY = .12, 20 H, INPUT DISCRETES = .		12. 20	H. INPL	IT DIS	CRETES	•	08. //		1355	
			THE WAR	1	. 11 . am. HSR2 =	582 =						33.6	
20	RETURN											1357	
	CMD		The second secon						-			250	Contract of the Contract of th

	SUBROUTINE	THE REPTI						100	
	LOGICAL		ERROR					1350	
	COMMON	CNTHENCE	142	. ICM	. ICOUNT .		TOIN .	1351	-
		I DOUT	IDSCR	. ITELM	. MAINM	MAINM (1024)	•	1352	
5	2	HBUSS .	MENC	. HENH	. MICAD	(6) . HJ	MICADIES, MICDAL 25,	1363	-
	•	MICO.	9	(91)	. MSR1		MSR2 .	1364	
		CC23 MONN	CINI	CRROR	. HENST		HOPHO	1369	
		MICRO(25)	•	. STINE	. ITINE		TIN	1356	
	CONTON	KAO	KOK	100.08	-			1381	
- 11	THE WALL	INSIN - MICHOLIN						1358	
	- 004	DO - MICKOLZI				-		1369	-
		IN MICKOLS! + 1.						1378	
	19 - HICKOLA	CROCKS . L.						161	
		1						1372	
3	7 24 107	T. A. Tuero	200		40.00				
	MALIE	HALLE VILLE INTENTION LINGUITATION CONTROL CONTROL CONTROL	1000 T 6 H3C	COLALION	T OF DE PE	Cranta	1 . THE CUZ . MI		
	10 FORMAT	(18H0EXECUTED OP CODE .OZ.15H HITH MODIFIED .11.1H. //	D OP COD	F -02-15H	WITH MOD	TETED	T1.11H. //	1373	
		S S S S S S S S S S S S S S S S S S S	- 85 . OF 1 21 CONTENTS						
82		22H CONTENTS INITIALLY .	S INITIA		.06. 8H. NOM =		.06. //	1378	
	•	28H Q-RES CONTENTS ENETTALLY	ONTENTS	+	1			1370	-
	•	28H TELEN SONTENTS INITIALLY .	ONTENTS	ENITIALLY		H. NOM .		•	
	Ifft =	FFETH					1		-
	HICOS #	MICO						1382	
62	RETURN				-	-		1983	
	CNO							1394	
	SUBROUT	SUBROUTINE REPTE				-		1333	
	ME I UKM		The state of the s					1386	
	2							1381	
Page 2 Day	18.18.18								
\$207aaaaaaa	LIBEGE III	40 OH 3	11111	1					
**********		/// END OF LEST ////	11111						

DISTRIBUTION	X	
	No. of Copies	
Defense Documentation Center	6	
Cameron Station		
Alexandria, Virginia 22314	6	
Commander		
US Army Materiel Development and Readiness		
Command	1	
ATTN: DRCRD	ī	
DRCDL		
5001 Eisenhower Avenue		
Alexandria, Virginia 22333		
DRSMI-FR, Mr. Strickland	1	
-LP, Mr. Voigt	1	
-R, Dr. McDaniel	1	
Dr. Kobler		
-RG, Mr. Huff	1	
-RGG,	10	
-RGC	1	
-RGL	1	
-RGP	1 1	
-RGT		
-RGN	1	
-RBD	3 1	
-RPR (Record Copy)		
(Reference Copy)	1	