CINCINNATI MILACRON

CIP/2200

SYSTEMS SUMMARY

Publication CTL-377

AUGUST 10, 1973

I N D E X

Page

CIP/2200 SYSTEM SUMMARY
- General Description..... 1
- System Features..... 1
- Software Features..... 2

CENTRAL PROCESSING UNIT
- Central Processing Unit Description..... 2
- Central Processing Unit Specifications..... 6
- CIP/2200 Instructions..... 8

PERIPHERALS
- Peripherals Supported..... 13
- Disk Units..... 13
- Printers..... 14
- Card Reader/Punch..... 16
- Magnetic Tape..... 16
- TTY..... 17
- Paper Tape Reader/Punch..... 17
- Communications Controllers..... 17

CiMOS-22 DISK OPERATING SYSTEM
- CiMOS-22 Summary..... 19
- Job Control Language..... 20
- Data Management..... 20
- File Access Capability..... 22
- Relocating Assembler..... 23
- RPG II..... 24
- Linkage Editor..... 24
- Text Editor..... 25
- Program Debugging..... 27
- Utilities..... 28

CIP/2200

General Description

The CIP/2200 is a powerful general purpose byte oriented minicomputer designed for stand alone processing, remote terminal, or dedicated system applications. It is designed to service a complete complement of peripheral devices with software support via CiMOS-22, a comprehensive disk oriented operating system.

The CIP/2200 with its TTL logic elements including MSI has achieved a cost/performance that allows many tasks previously assigned to large computer systems, or hard wired controllers to be achieved much more efficiently on a CIP/2200.

The main memory of the CIP/2200 consists of 8-bit/byte core memory with a 1.1 microsecond full cycle time. Main memory is expandable in 8K modules to a maximum memory size of 32K bytes, accessible using direct, relative, index, indirect, and literal addressing modes. The byte oriented operation, coupled with a complete set of arithmetic formats including binary, decimal, and multi-precision, allow maximum memory utilization and character manipulation. Programs can be written with much greater flexibility using the instruction repertoire of 119 basic commands.

Many features unavailable or available at extra cost on other computers are standard features of the CIP/2200. The I/O structure consists of a micro-programmed serial I/O interface, a byte I/O facility, firmware supported DIRECT MEMORY CHANNEL transfers, and the capability of adding up to two DIRECT MEMORY ACCESS Processors which allow up to two Direct Memory Processors which allow data transfers at up to 910,000 bytes per second. A priority interrupt system allows 6 internal and up to 64 external interrupts. In addition, the CIP/2200 control stack facility provides a method of saving and restoring the computer state information which greatly increases programming ease and flexibility in an interrupt environment.

Operator control is provided by a functional system control panel which is mounted on the front of the main-frame chassis. This panel may be used to display and modify registers, and to control program executions.

A very extensive disk operating system is provided which facilitates user programming and operations. Included in the Disk Operating System are RPG II, Assembler, Linkage Editor, Text Editor, and Library facilities.

System Features

- Byte oriented architecture
- Expandable core memory (8K to 32K bytes); 1.1 microsecond cycle
- 119 basis instructions
- Direct, relative, index, indirect, literal addressing

- 5 operational registers including hardware index
- Binary, decimal, fixed point, and multiprecision arithmetic formats
- Complete I/O facilities including serial, byte, Direct Memory Channel and Direct Memory Access for high speed transfers
- Hardware priority interrupt with up to 64 external interrupts
- Operator's system control panel
- Full range of peripherals

Software Features

- Complete disk operating system
- RPG II
- Relocating Assembler
- Linkage Editor
- Library maintenance
- Sequential, random, and indexed files
- Text Editor
- RJE support

CENTRAL PROCESSING UNIT DESCRIPTION

Central Processing Unit

The CIP/2200 has an 8-bit hardware data path and memory. The CPU registers, however, are 16 bits in length and include the A-register (accumulator), B-register (auxiliary accumulator), X-register (index), P-program counter, and S-(machine status). The instruction set includes a complete set of 16-bit register-to-memory and register-to-register binary arithmetic instructions.

The CIP uses microprogramming techniques which allow implementation of more powerful instructions, greatly expanding the applications and functions accommodated. A control stack mechanism saves the machine state when entering subroutines or interrupt service routines. The program counter, machine status, A, B, and X registers are saved.

The disk IPL greatly reduces the time required to load the basic bootstrap routine. The loader is a firmware routine which moves up to 32K bytes of core image data from the disk to main memory and transfers control to a specified location to load the remaining software. The full 32K bytes can be loaded in less than 220 ms using disk IPL.

Input/Output Operations

The CIP/2200 I/O structure provides several I/O facilities that enable it to be a powerful I/O processor. Features of the I/O structure include a micro-programmed serial I/O interface, byte I/O facility, firmware support for Direct Memory Channel transfers, and the capability of attaching two independent Direct Memory Access (DMA) processors. The Direct Memory Channel allows microprogram controlled high speed data transfer to occur concurrently with program execution. The DMC will allow external devices to transfer data at up to a total of 25,000 bytes/second without affecting response to internal interrupts. Data transfer rates up to 86,000 bytes/second can be accomplished from buffered I/O devices with a correspondingly longer response time to internal interrupts. Up to 910,000 bytes per second can be transferred via the Direct Memory Access Unit which is an independent hardware controller that competes with the CPU for use of main memory time. Priority interrupts are used by device controllers to increase the efficiency of the I/O operations.

Core Memory

The CIP/2200 magnetic core memory is constructed in modules of 8,192 bytes (- bit) up to a maximum of 32,768 bytes with a 1.1 microsecond full cycle. Programs stored in core memory can directly address all of core memory. The hardware index register can also be loaded with a bias value to allow indexed addressing within core.

Decimal arithmetic is an important feature of the CIP/2200. Decimal numbers appear in memory as byte strings up to 16 digits in length.

Variable length operations offer more efficient use of memory allowing binary arithmetic to be performed on 8, 16, 24, or 32 bit data without resorting to multiple precision software routines. String moves allow up to 256 bytes of data to be moved by one instruction. In memory-to-memory operations the registers are not altered.

Interrupts

The CIP/2200 interrupt system is responsive to two types of interrupts. Internal interrupts are generated by the computer on the occurence of an operational fault, a console interrupt, or by the interval timer. External interrupts are generated by an I/O device or another externally supplied signal. When an interrupt occurs, the computer responds by executing a subroutine transfer of control to the address specified by an interrupt transfer location. This method of interrupting the execution of one program and starting execution of another program allows the computer to respond to external events and execute the appropriate portions of the program on a priority basis.

Disk IPL

The CIP/2210 computer uses the CIP/2200 instruction set, but in addition allows disk IPL (Initial Program Load).

The disk IPL greatly reduces the time required to load the basic bootstrap routine. The loader is a firmware routine which moves up to 32K bytes of core image data from the disk to main memory and transfers control to a specified location to load the remaining software. The full 32K bytes can be loaded in less than 220 ms using disk IPL.

CENTRAL PROCESSING UNIT SPECIFICATIONS

General - A general purpose byte oriented minicomputer designed for stand alone processing, remote terminal or dedicated system applications.

Memory - Magnetic core, 8-bit byte, 1.1 microsecond full cycle. Expandable from 8K to -2K bytes in increments of 8K.

Addressing - Direct, indirect, relative, and indexed.

Instructions - 119 basic commands.

Instruction Execution Time (Microseconds)

Add Logical	11.7
Add Decimal	84.5 + 7.5 per digit
Load/Store	12.1/11.2
Branch on Condition	11.4/12.2
Jump	10.1
Move Character	67.7 + 5.5 per character
Skip	6.8/7.7
Translate	72.8 + 20.0 per character
Compare Logical Character	55.5 + 8.6 per character

Registers

P - Program Counter	16 bits
A - Accumulator	16 bits
B - Auxiliary accumulator	16 bits
X - Index	16 bits
S - Status register	8 bits

Input/Output

Serial I/O - 10 bytes/second

Direct Memory Channel - The DMC will allow external devices to transfer data at up to a total of 25,000 bytes/second without affecting response to internal interrupts. Data transfer rates up to 86,000 bytes/second can be accomplished from buffered I/O devices with a correspondingly longer response time to internal interrupts.

Direct Memory Access - 910,000 bytes per second

System Control Panel

16 Data/Address entry switches
15-bit memory address display
10-bit display of ROM address

System Control Panel - continued

8-bit display of A-BUS
16-bit display of ROM instruction
4 sense switches
Clock, reset, interrupt, save, switches
key off - enable - disable switch

Dimensions

8-3/4 X 19 X 23 inches

Power

240 W, 115-230 VAC, 50-60 Hz

Environment

0° - 50°C temperature range
0% - 90% noncondensing humidity range

CIP/2200 INSTRUCTIONS BY ASSEMBLER FORMAT

MNEMONIC	MACHINE INST. CLASS	OPCODE (HEX)	NAME
I CONTROL, OPERANDS: NONE			
HLT	C	00	HALT
TRP	C	01	TRAP
ESW	C	02	ENTER SENSE SWITCHES
DIN	C	04	DISABLE INTERRUPT SYSTEM
EIN	C	05	ENABLE INTERRUPT SYSTEM
DIT	C	06	DISABLE INTERVAL TIMER
EIT	C	07	ENABLE INTERVAL TIMER
RO1	V	08	RESET OV, SET WL=1
RO2	V	09	RESET OV, SET WL=2
RO3	V	0A	RESET OV, SET WL=3
RO4	V	0B	RESET OV, SET WL=4
S01	V	OC	SET OV, SET WL=1
SO2	V	OD	SET OV, SET WL=2
SO3	V	OE	SET OV, SET WL=3
SO4	V	OF	SET OV, SET WL=4
NOP	C	34	NO OPERATION
XIT	C	5E	ROM EXIT
II EXTENDED CONTROL, OPERANDS: NONE			
SAV	T	5F00	SAVE MACHINE STATE
RET	T	5F01	RETURN
III REGISTER OPERATE, OPERANDS: NONE			
IAB	B	03	INTERCHANGE A & B
DCA	B	23	DECREMENT A
DCB	B	27	DECREMENT B
TAB	B	2B	TRANSFER A TO B
TBA	B	2F	TRANSFER B TO A
IAX	B	35	INTERCHANGE A & X
IBX	B	36	INTERCHANGE B & X
ORA	B	40	OR B TO A
XRA	B	41	EXCLUSIVE OR B TO A
ORB	B	42	OR A TO B
XRB	B	43	EXCLUSIVE OR A TO B

MNEMONIC	MACHINE INST. CLASS	OPCODE (HEX)	NAME
III. CONTINUED			
INX	B	44	INCREMENT X
DCX	B	45	DECREMENT X
AWX	V	46	ADD WORD LENGTH TO X
SWX	V	47	SUBTRACT WORD LENGTH FROM X
INA	B	48	INCREMENT A
INB	B	49	INCREMENT B
OCA	B	4A	1'S COMPLEMENT A
OCB	B	4B	1'S COMPLEMENT B
TAX	B	4C	TRANSFER A TO X
TBX	B	4D	TRANSFER B TO X
TXA	B	4E	TRANSFER X TO A
TXB	B	4F	TRANSFER X TO B
IV. CALL			
CALL	T	6A88	CALL
V. CONDITIONAL SKIP, OPERANDS: ADDR			
SOV	T	10	SKIP IF OVERFLOW SET
SAZ	T	11	SKIP IF A=0
SBZ	T	12	SKIP IF B=0
SXZ	T	13	SKIP IF X=0
SAN	T	14	SKIP IF A NEGATIVE
SXN	T	15	SKIP IF X NEGATIVE
SAB	T	16	SKIP IF A=B
SAX	T	17	SKIP IF A=X
NOV	T	18	SKIP IF OVERFLOW NOT SET
NAZ	T	19	SKIP IF A≠0
NBZ	T	1A	SKIP IF B≠0
NXZ	T	1B	SKIP IF X≠0
NAN	T	1C	SKIP IF A NOT NEGATIVE
NXN	T	1D	SKIP IF X NOT NEGATIVE
NAB	T	1E	SKIP IF A≠B
NAX	T	1F	SKIP IF A≠X

MNEMONIC	MACHINE INST. CLASS	OPCODE (HEX)	NAME
VI. SHIFT, OPERANDS: N ($0 \leq N \leq 32$)			
RLA	S	20	ROTATE LEFT A
RLB	S	21	ROTATE LEFT B
RLL	S	22	ROTATE LEFT LONG
LRA	S	24	LOGICAL RIGHT SHIFT A
LRB	S	25	LOGICAL RIGHT SHIFT B
LRL	S	26	LOGICAL RIGHT SHIFT LONG
ALA	S	28	ARITHMETIC LEFT SHIFT A
ALB	S	29	ARITHMETIC LEFT SHIFT B
ALL	S	2A	ARITHMETIC LEFT SHIFT LONG
ARA	S	2C	ARITHMETIC RIGHT SHIFT A
ARB	S	2D	ARITHMETIC RIGHT SHIFT B
ARL	S	2E	ARITHMETIC RIGHT SHIFT LONG
VII. ESCAPE, OPERANDS, N ($0 \leq N \leq F7_{16}$)			
XT2	C	5F08-5FFF	SECONDARY ROM EXIT
VIII. EXTENDED IMMEDIATE, OPERANDS: I ($-128 \leq I \leq 127$)			
AXI	B	5F02	ADD TO INDEX IMMEDIATE
RTN	T	5F03	RETURN DISPLACED
IX. MAIN STORAGE IMMEDIATE, OPERANDS (I,N), ADDR(X)			
AWI	B	50	ADD TO WORD IMMEDIATE
BOC	T	51	BRANCH ON CONDITION
MVI	MI	52	MOVE IMMEDIATE
CLI	MI	53	COMPARE LOGICAL IMMEDIATE
TMI	MI	54	TEST UNDER MASK IMMEDIATE
SMI	MI	55	SET BITS UNDER MASK IMMEDIATE
CMI	MI	56	CLEAR BITS UNDER MASK IMMEDIA
IMI	MI	57	INVERT BITS UNDER MASK IMMEDI
X. MAIN STORAGE TO MAIN STORAGE, OPERANDS:TADDR(L,X),SADDR(X) ($1 \leq L \leq 256$)			
MVL	MM	5C	MOVE CHARACTER STRING LEFT
MVR	MM	5D	MOVE CHARACTER STRING RIGHT
EDT	MM	5F04	EDIT AND MARK
CLC	MM	5F05	COMPARE LOGICAL CHARACTER

MNEMONIC	MACHINE INST. CLASS	OPCODE (HEX)	NAME
XI. UNCONDITIONAL SKIP, OPERANDS: ADDR (x)			
JMP	T	60-67	JUMP
RTJ	T	68-6F	RETURN JUMP
XII. LOGICAL, OPERANDS: ADDR (x)			
ANA	B	DO-D7	AND MEMORY TO A
ANV	V	D8-DF	AND VARIABLE
XIII. BINARY ARITHMETIC, OPERANDS: ADDR (x)			
IWM	B	70-77	INCREMENT WORD IN MEMORY
DWM	B	78-7F	DECREMENT WORD IN MEMORY
MST	B	90-07	MULTIPLY STEP
DST	B	98-9F	DIVIDE STEP
ADA	B	AO-A7	ADD TO A
ADV	V	A8-AF	ADD VARIABLE
SBA	B	BO-B7	SUBTRACT FROM A
SBV	V	B8-BF	SUBTRACT VARIABLE
XIV. LOAD AND STORE, OPERANDS: ADDR (x)			
LDX	B	80-87	LOAD X
STX	B	88-8F	STORE X
LDB	B	CO-C7	LOAD B
STB	B	C8-CF	STORE B
LDA	B	EO-E7	LOAD A
LDV	V	E8-EF	LOAD VARIABLE
STA	B	FO-F7	STORE A
STV	V	F8-FF	STORE VARIABLE

MNEMONIC	MACHINE INST. CLASS	OPCODE (HEX)	NAME
XV. TRANSLATE, OPERANDS. N, TADDR (L,X),SADDR(X) (0≤N≤255)(1≤L≤256)			
TRM	MM	5F06	TRANSLATE UNDER MASK
TTM	MM	5F07	TRANSLATE AND TEST UNDER MASK
XVI. DECIMAL, OPERANDS: TADDR (R,X),SADDR (SL,X) (1≤TL,SL≤16)			
ADD	MM	58	ADD DECIMAL
SBD	MM	59	SUBTRACT DECIMAL
MSD	MM	5A	MULTIPLY STEP DECIMAL
DSD	MM	5B	DIVIDE STEP DECIMAL
XVII. I/O SERIAL, OPERANDS: NONE			
IBS	IO	30	INPUT BYTE SERIALLY
OBS	IO	38	OUTPUT BYTE SERIALLY
XVIII. I/O REGISTER, OPERANDS: N1,N2(0≤N1≤7, 0≤N2≤31)			
IBA	IO	31	INPUT BYTE TO A
IBB	IO	32	INPUT BYTE TO B
OBA	IO	39	OUTPUT BYTE FROM A
OBB	IO	3A	OUTPUT BYTE FROM B
XIX. I/O MAIN STORAGE, OPERANDS: N1,N2,ADDR(X) (0≤N1≤7,0≤N2≤31)			
IBM	IO	33	INPUT BYTE TO MAIN STORAGE
OBM	IO	3B	OUTPUT BYTE FROM MAIN STORAGE

MACHINE INSTRUCTION CLASSES

B - BINARY ARITHMETIC AND LOGICAL
S - SHIFT
V - VARIABLE WORD LENGTH AND ASSOCIATED
MI - MAIN STORAGE IMMEDIATE
MM - MAIN STORAGE TO MAIN STORAGE
T - TRANSFER OF CONTROL
C - CONTROL
IO - INPUT/OUTPUT

PERIPHERALS

The CIP/2200 interfaces a complete line of peripheral devices to provide a complete data processing systems capability. In addition the CiMOS-22 disk operating system supports these peripherals via user oriented software routines.

The CIP/2200's input/output structure and priority interrupt capability allow optimum use of various peripheral transfer rates without seriously impeding system throughput. The microprogrammability of the CIP/2200 also will allow additional devices to be interfaced when needed.

PERIPHERAL DEVICES SUPPORTED

- Disk Drives
- Character Printer
- Line Printers
- 80-Column Card Reader
- 96-Column Card Reader/Punch
- Magnetic Tape Drives
- CRT's
- TTY's
- Paper Tape Reader/Punch
- Asynchronous Communications Controllers
- Synchronous Communications Controllers

DISK UNITS

Controller Features:

- Interfaces from 1 to 4 drives
- Data transfers via DMA and status control via I/O Bus
- Full error checking on hardware and data

- Seek and data operations may be given at the same time on one drive
- Simultaneous seeks in a multidrive system

Disk Drive Features:

- Removable disk cartridge plus fixed disk
- Removable cartridge allows one drive to handle many applications normally requiring two devices
- 1 head per disk surface
- Capacity - 406 cylinders (203 on fixed, 203 removable)
  - 4,915,200 Bytes/Drive
  - 256 Bytes/Sector
  - 24 Sectors/Track
- Access Time - Track to Track 35 MS
  - Average Random Move 95 MS
  - Average Rotational Delay 20 MS
- Data Transfer Rate - 195,000 Bytes/Second
- Track Density - 100 TPI

CHARACTER PRINTERS

- Speeds - 165 characters per second or 125 lines per minute
- Character Structure - 9 X 7 dot matrix, 10 point type eqv.
- Character Buffer - 132 characters
- Printing Structure - 132 characters/line, 6 lines/inch
- Character Set - 64 characters (USASCII)
- Paper Feed - Sprocket Feed to 14 7/8" width
- Number of Copies - Original plus 4 copies

- Vertical Format Control
- Paper Runaway Inhibit
- Form Feed

LINE PRINTERS

- Speeds - up to 200 lines/minute
- Buffer Size - 132 characters
- Printing Structure - 132 characters/line, 6 lines/inch
- Character Set - 64 characters (USASCII)
- Paper Feed - Sprocket feed to 14 7/8" width
- Paper Slew Rate - 27.5 inches/second
- Number of Copies - Original plus 5 copies
- Vertical Format Control - 8 Channel
- Form Feed

80 COLUMN CARD READER

- Speed - Reads at 600 CPM
- Capacity: Input - 600 cards
  Output - 1000 cards
- Drum Transport Mechanism

96 COLUMN CARD READER/PUNCH

- Speeds - Reads at 300 CPM
  Punches at 60-120 CPM
  Read/Punch at 60-120 CPM
- Fully buffered input/output
- Dual Input hoppers and output stackers
- Capacity: Primary Input - 600 cards
  Secondary Input - 400 cards
  Primary output - 400 cards
  Secondary output - 400 cards

MAGNETIC TAPE SYSTEMS

Tape Controller

- Interfaces 1 to 4 magnetic tape drives
- NRZI or phase encoded drives

- IBM or ANSCII compatible

Magnetic Tape Drives

- 7 or 9 track
- Speed 25 inches per second
- Automatic reel seating hold-down hubs
- Read after write
- 800/1600 BPI
- 14.4 MS start/stop time
- 10½ inch reel

TTY's

- ASR-33 TTY
- 10 Characters per second
- 72 Characters per line
- Local or remote transmission

Paper Tape Reader/Punch

- Reads up to 300 characters per second
- Punches up to 240 characters per second
- Reads and punches 5, 6, 7, and 8 level code on 1 inch wide tape

Asynchronous Communications Controller

- Communicates with six asynchronous serial devices
- Speeds from 110 to 4800 baud
- Full duplex operation

- Odd or even parity
- Selectable word length (5, 6, 7, or 8 bits)
- Selectable stop bits (1 or 2)

Synchronous Communications Controller

- Operates with synchronous modems with speed range from 1200 to 9600 baud
- Dedicated or dial up lines
- Half or full duplex
- Complete error checking capabilities

CiMOS-22 DISK OPERATING SYSTEM

The CiMOS-22 Operating System is the Cincinnati Milacron disk-oriented operating system for the CIP/2200 minicomputer. This operating system functions interactively with a single user, that user being both programmer and operator. The user directs the operation of the system using a Job Control Language which allows files to be created, cataloged, modified, and deleted, and also allows programs to be loaded, executed, and cancelled. Files may contain data, libraries of source programs, libraries of object programs, or executable programs in core image form.

The language processors included with the CiMOS-22 Operating System are an RPG II Compiler and a Relocating Assembler. Also provided are a Text Editor for creating and modifying files of source programs and data, a Library Maintenance Program for coordinating the object output of the language processors, and a Linkage Editor for creating executable programs and overlays in core image form.

A major service of the CiMOS-22 Operating System is supplied by its Input/Output Control System. Support is provided for random, sequential, and indexed-sequential disk files, and for devices such as card reader, printer, and paper tape reader-punch. Sequential files on disk may have records containing up to 32,968 characters. Operations appropriate to each device or file type are provided, including OPEN, CLOSE, READ, WRITE, READ BACKWARDS, SKIP, UPDATE, and DELETE. When System Generation is performed, various components of the Input/Output Control System can be declared resident in core memory to conserve disk access time or transient to conserve core memory space.

Data Management facilities of the CiMOS-22 Operating System include disk space allocation and deallocation, disk directory maintenance, and eight independent types of file protection including read, write, update, and execute protection. The Data Management facilities allow many programs to be written in a device-independent manner, so that the decision to use, for example, a printer or a sequential disk file for output can be made by the operator when a program is executed without modifying or even notifying the user program.

Several utility programs and commands are supplied with the CiMOS-22 Operating System. They allow a disk pack to be initialized, reorganized, or copied, and files to be copied, "dumped," and "restored." Debugging commands allow program execution to be break-pointed, and the contents of the operating registers and of core memory to be displayed or modified.

A minimally configured version of the CiMOS-22 Operating System uses the first 8K of core memory. At least another 8K of core memory is required by the language processors, so the minimum core memory requirement is 16K. Modular design of the system allows various components to be made core resident when System Generation is performed; exercising this option requires 24K or more of core memory but does greatly improve throughput.

JOB CONTROL LANGUAGE

The CiMOS-22 System provides user control via its Job Control Language. Communication with the system is established and maintained through the system console. Whenever the system is loaded, the Job Control Language prints a request for a command. The user can either execute or debug programs selectively using JCL. In the area of program execution he can:

1. Create or delete a disk file.
2. Change file names and file protection.
3. Catalog a file or remove it from the catalog.
4. Set the system time and date.
5. Set user switches as required.
6. Execute a selected program.
7. Specify options to be in effect during program execution.
8. Define new option keywords.
9. Cancel a job.
10. Assign reference file names to actual files.

In the area of program debugging he can:

1. Display, with the option of changing the contents, the A, B, X, and S registers, specified main storage locations.
2. Display extended areas of main storage.
3. Display address of entries in the user symbol table.
4. Perform hexadecimal arithmetic, giving both sum and difference of two operands.
5. Load a program and test it.

In summary, JCL enables the user to establish and conduct a smooth-flowing jobstream. Programs and utility functions can be executed in any appropriate sequence. Furthermore, the sequence may be altered easily as changes and conditions dictate.

DATA MANAGEMENT

Data Management is a major function of the CiMOS-22 System that involves organizing, cataloging, storing, retrieving and maintaining data. The system accepts two primary formats of data, ANSCII and System Binary. Use

of ANSCII data allows easy communication between the user and the computer system, and allows the saving and restoring of ANSCII character strings. System binary format is the simplest form of data using only "0" and "1" bit patterns in the form of byte strings. System Binary format provides a means of storing byte strings and retrieving them without changing the format of the string.

To manipulate data the data management programs must have a great deal of information about the data. This information can be supplied by the users program or via the system console. The CiMOS-22 System allows both user program and system console manipulation of data. It also provides the capability of fixed or variable record length files.

Disk packs play a major role in the system. They are used to hold executable programs, the system itself, data, and are used for temporary work space. In order to facilitate disk file manipulation, the following capabilities are provided:

1. Volume table of contents
2. Volume passwords
3. Creation/expiration dates
4. File protection to prevent:
   a. reading a file
   b. writing records to a file
   c. updating records in a file
   d. deleting records from a file
   e. executing a file
   f. deleting a file
   g. moving a file
   h. record deletion check
5. Disk catalog
6. Resident catalog
7. File assignment table
8. Device table

An important feature of the data management portion of the CiMOS-22 System is device independence. Device independence allows programming input/output of data without knowing device characteristics until execution takes place. Actual device assignment takes place via console commands at execution time.

The CiMOS-22 System allows sequential, random and indexed file organizations. Sequential files can contain either fixed or variable length records, although the system treats all records as if they were variable length. Random files must be processed directly with addressing specified by the user. All random records in a file must be of the same length.

FILE ACCESS CAPABILITY

Data management for the CiMOS-22 System consists of several routines which establish files and efficiently schedule and control the transfer of data between main storage and input/output devices. File establishment and control operations include:

1. Create a file
2. Open a file or device
3. Close a file or device
4. Delete a file
5. Halt an input/output operation

Data transfer and manipulation operations for sequential files include:

1. Read the next sequential record
2. Read the previous sequential record
3. Write new sequential record
4. Delete the previously read record
5. Update the previously read record
6. Skip records in a sequential file

Data transfer and manipulation operations for random files include:

1. Read record by relative record number
2. Write record by relative record number
3. Delete the previously read record
4. Update the previously read record

Additional special operations are available for special I/O devices such as printers.

Console communications are provided to allow the user's program to communicate with the console keyboard/printer. These operations are:

$TYPE - for typing messages on the console printer.
$ACCEPT - for accepting entries from the keyboard.
$REPORT - for typing error messages and accepting a user response.

The ability to overlap input/output and program processing is also available under all file access methods that are interrupt driven. This capability allows the program to process data while waiting for an input/output operation to be completed.

Language Processors

The CiMOS-22 Operating System has two language processors, RPG II and a Relocating Assembler.

Relocating Assembler

The assembler is a component of the CiMOS-22 System. The operating system provides the assembler with input/output, library, and other services needed in assembling a source program. The assembler program (ASM/2200) processes the symbolic language in order to translate the symbolic instructions, assign storage locations, and perform auxiliary functions necessary to produce an executable machine language program. The output of the assembler program is an object module, a machine language translation of the source program. The assembler furnishes a printed listing of the original source statements, the machine code representation of the source, and error diagnostics.

ASM/2200 is a disk oriented assembler. It reads the source program from cards, disk, or tape, analyzes the fields of the statements, and produces the object code.

The assembler is available in two versions: 8K and 12K core required. The 12K assembler is much faster than the 8K version since many modules are core resident.

ASM/2200 in addition to its symbolic assembler instructions also has the following pseudo instructions:

MNEMONIC	OPERANDS	NAME
ABS		set absolute mode
CMN	N(0_N_256)	define common storage
% CON		continue (name required)
DC	DATA	define constant
DS	N(0_N_256)	define storage
EJECT		eject page
END	ADDR	end
ENT	NAME1,...,$NAME_n$	define entry symbol
EQU	ADDR	equate (name required)
EXT	NAME1,...,$NAME_n$	define external symbol
% GTO	SYMB	go to
IDENT	NAME	program indentification
% IF	I,SYMB	if
% IFN	I,SYMB	if not

MNEMONIC	OPERANDS	NAME
ORG	ADDR	origin
PRINT	I	print control
REL		set relocatable mode
SET	ADDR	set
SPACE	N	space listing
TITLE	C'CHARACTER STRING	set title

RPG II

RPG II is a highly flexible, problem solving language. RPG stands for Report Program Generator and is a simple language designed primarily for programming ease in solving basic calculations, file updates, and written reports.

The RPG II compiler for the CiMOS-22 System will accept source statements from cards, disk, or tape and convert them to a machine language object program. The object program is then used by the CiMOS-22 System to process information according to programmer specifications.

RPG II uses the following specifications in order to perform a job:

1. Control card and File Description Specifications
2. Extension and Line Counter Specifications
3. Input Specifications
4. Calculation Specifications
5. Output Specifications

These specifications along with the necessary control statements constitute the input source data for the RPG compiler. The output of the RPG compiler includes a listing of the input source, the generated object code listing, symbolic table listing, and diagnostic error listing. The generated object code can be directed to disk or tape.

Linkage Editor

The Linkage Editor combines separately assembled object modules into a single load module. The object modules are in relocatable form; that is, their addresses are identified for later modification to absolute addresses.

The names of the object modules to be combined are provided by the user. The Linkage Editor then locates the modules, and links them into a single absolute load module that is a random file on disk. This load module can then be loaded by the system and executed.

The Linkage Editor under the CiMOS-22 System provides the ability to have an overlay structure. The user may divide his program into a base module and any number of overlays. The Linkage Editor places these segments on disk so that, at execution time, the base module is automatically loaded into main storage and the overlays are loaded only when requested via supervisor call from the base module.

Individual object modules, or an entire library of object modules, can be included in a load module. The Linkage Editor allows the user to place commonly used routines in object module format in a library and then request that they be linked selectively into a single load module.

Text Editor

The Text Editor (TED) for the CiMOS-22 System is a program for creating and modifying source program libraries and data files. TED operates on an ANSCII input text library or file and produces a single ANSCII output text library. New records and corrections to old records are entered through the console keyboard. The various commands recognized by TED allow the user to interact with the system to enter, merge, copy, modify and delete complete files, programs, or records.

TED allows the user to define two input data files. These files may be disk files or I/O devices such as card or paper tape. In addition the console can always be an input source. All text is read by TED into a 96-column work buffer, one record at a time, and written to the output file.

Once a record is in the work buffer it may be displayed or modified before it is written to the output file. Modification is performed much as it is done on a keypunch. The user may duplicate, skip, or change the various fields of a record before writing it to the output file.

TED has an extensive command set with which it manipulates the text. There are four major catagories of TED commands:

1. A — Assign and open read file
   CT — Clear tabs
   EJ — End of job
   ES — End of job step
   FC — File create
   FD — File delete
   R — Select read file
   ST — Set tabs
   TF — Advance to top of form
   W — Open write file

2. Commands which address the text of the input file only:

D	Display next record
LB	List to beginning character string
LC	List to contained character string
LI	List to program IDENT
LL	List to end of file (library)
LP	List programs
LR	List records
SB	Skip to beginning character string
SC	Skip to contained character string
SI	Skip to program IDENT
SL	Skip to end of file (library)
SP	Skip programs
SR	Skip records
SS	Skip to start of file (library)

3. Commands which address the text of the output file only:

BS	Backspace record
EC	Enter records continuously
EL	End of file (library)
EP	End of program
ER	Enter records
P	Print, or suppress printing

4. Commands which address the text of both input and output files:

CB	Copy to beginning character string
CC	Copy to contained character string
CI	Copy to program IDENT
CL	Copy to end of file (library)
CP	Copy programs
CR	Copy records
MC	Modify records continuously
MR	Modify records

The commands listed above may be used in any operational sequence provided the required file or files have been assigned. The categories in which the commands are placed merely provide a cursory overview of their functions and should prove helpful to the user in determining their proper applications.

All operations are ultimately terminated when an end of file is detected. Character String and Record commands are terminated when the system detects an end of program mark. At the completion of the Program IDENT and Character Strings commands, TED displays the contents of the work buffer on the console keyboard/printer.

Program Debugging

The CiMOS-22 Operating System is equipped with the capability of debugging or modifying the user's program at two levels, the source module and the load module. Through use of Text Editor (TED) the user may correct errors in the syntax of his program, and through use of the Job Control Language (JCL) the user may correct errors in the logic of his program.

The description of the Text Editor (TED) and corresponding commands is discussed under TEXT EDITOR (see Index).

The Job Control Language is also described (see Index), but the following JCL commands are available exclusively for load module debugging:

LO	Load user program
C	Continue execution from a specified location
MD	Display main storage
A	Open A register
B	Open B register
X	Open X register
S	Open S register
+	Open main storage offset register
LMO	Open load module
MO	Open main storage
H	Perform hexadecimal arithmetic
=	Display address from user symbol table

In addition to source and load module modification, the user has at his disposal for program debugging three types of main storage dumps:

SNAPSHOT -

The snapshot dump consists of a hexadecimal printout of storage locations designated by the user via a Display Main Storage (MD) command. The dump can be printed on the system printer or the console.

JOB CANCEL DUMP -

The job cancel dump consists of a hexadecimal printout of storage locations designated by the user by means of the Job Cancel command. The Job Cancel dump provides the stack pointer, stack contents, and user communication vectors in addition to the main storage printout.

SYSTEM ERROR DUMP -

When an error in the system is detected, the system automatically terminates the job and prints a dump of

all main storage. The format is the same as the Job Cancel dump. The system dump is printed on the system printer.

Utilities

The CiMOS-22 Operating System includes a number of utility programs for disk and library maintenance.

Initialize Disk (IND)

The initialize disk utility prepares a disk pack for the CiMOS-22 System by checking for read/write capability and recording necessary control information on it. It also deletes all data from a previously used disk pack.

The initialize disk utility establishes, with the aid of user input through the console, the necessary Volume Table of Contents (VTOC) and the VTOC entries for files initially created on each pack. The initialize disk program functions on either fixed or removable packs on any drive. Packs may be initialized in data or IPL (Initial Program Load) format.

IND allows the user at execution time to specify the allowed size of the VTOC, the volume number of the pack, the password for protection if desired, and the owner's name.

Disk Reorganization (DRG)

The disk reorganization utility allows the user to rearrange the files on disk storage to free unused space resulting from file deletion. Disk reorganization eliminates unusable storage areas by rearranging disk files and available free storage.

VTOC and Catalog List (VCL)

The program VCL provides the user with the ability to list the volume table of contents, and the disk catalog. The file name, size, type, and other pertinent data is listed on the printer or console.

Library Maintenance Program (LMP)

The program LMP enables the user to create libraries of relocatable object modules for input to the linkage editor. He may also modify existing libraries, adding or deleting modules as desired.

A library is created by copying or merging relocatable object modules, produced by language translators, into a single library file with each module delimited by an end of program indicator. A module is added to the library by copying the original library up to the point where the new module is to be inserted, copying the new module, and then copying the remainder of the library.

To facilitate this procedure LMP may use up to four input files.

LMP has available the following commands:

P	Reassign the printer file
F	Top of form command
M	Set top of form message
A	Open input file; assign file number
FC	File create
W	Open output file
R	Read a file
C	Copy n number of modules (optional list, copy up to modules, copy library)
SL	Skip library
S	Skip n modules
D	Display IDENT
L	List up to module
LL	List library
E	Close output file
FD	File delete
ES	End step
EJ	End of job

Copy Files (CPY)

CPY is a utility that enables the user to copy files, or to list the contents of a specified file on the system printer. Files may be copied from disk to disk, or from disk to another medium and back again.

The following commands control CPY:

A	Assign file name, organization, format, etc.
L	List the input files on the printer
C	Open the output file and copy the input file to it
S	Open specified sequential output file and copy assigned input random file to it
R	Restore a file previously saved via "S" command
ES	End step
EJ	End job

Duplicate Disk (DUP)

The DUP utility provides a simple method of system backup via copying the contents of one disk pack to another.

Disk Sort (SORT)

The SORT program provides the capability to sort fixed or variable length records in ascending or descending order. Control fields can be specified either ascending or descending and can be mixed. Input records can be packed or unpacked. Output file is in sequential unpacked format. To sort records of maximum size (4096 bytes) will normally require 24K bytes of main memory. File size to be sorted is limited only by the available work-space on disk.