Automatic Electronic Systems Inc.

The AES-80 Microprocessor

A Fourth dimension in data acquisition processing and communication

Printed in Canada

1 Hardware

2 Controllers

3 Minicomputers

4 The AES-80 Microprocessor

There used to be three

There were three ways to design a digital data product or system, that is, until the AES-80 Microprocessor.

1) The traditional approach required dedicated hardware — a time consuming exercise involving long lead times for prototyping. Just when you have your product developed a customer with different requirements sends you back to redesign and modify. Often you end up with a proliferation of products, none with enough volume for reasonable manufacturing costs. Why build hardware if a make/buy analysis favours the AES-80 over your own design?

2) A recent approach uses a controller for logical functions. Typically, controllers have limitations: they are slow — with a limited instruction set and inadequate input/output capability. If you have not found a suitable controller, why not look at our Microprocessor?

3) In more complex applications, a minicomputer is a frequent choice for your dedicated processing application — often a case of overkill (and overexpenditure). A mini is a general purpose digital computer and its capabilities are not required in many applications. Furthermore minicomputer instructions do not always result in an optimum approach to a given problem and input/output is relatively complex. Finally, when you add up the cost for a minicomputer plus memory plus interface options, you have a pretty expensive item. Until the AES-80, however, you had no viable alternatives. Why buy a mini when a micro will do?

Go fourth

with the AES-80 Microprocessor, a byte oriented processor at a fraction of the cost of a mini. It is designed for the OEM or sophisticated user in dedicated processing applications. The AES-80 is part of a family of Modular System Units (MSU's) designed for data acquisition, processing and communication. Serial I/O, analog I/O, communication, control, and special purpose modules are designed for "daisy chaining" with the microprocessor MSU to satisfy specific applications.

Why the AES-80?

Using the AES-80, your design cycle is reduced to writing a software program and debugging it on our easy-to-use Program Development and Control Console. We then convert this program to a permanent Read Only Memory (ROM) and supply the completed microprocessor with memory and input/output MSU's ready for installation. Alternatively, our Programming Service is available for the software phase.

You eliminate that costly design and prototyping cycle and get to your market while others are still in the design phase. In short, the AES-80 can make you money.

The AES-80 features exceptionally fast processing-240 nanoseconds full cycle time with bipolar memory. The instruction set of 92 micro-instructions allows great flexibility for solving application problems efficiently. The input/output capability is exceptional with the serial I/O bus capable of a transfer rate of over 100,000 (8-bit) characters per second to or from the microprocessor. Parallel I/O peripherals share the high speed parallel bus with the data memory and communicate at internal processor speeds.

Naturally there is the cost element. Using the AES-80 optimizes the tradeoff between hardware and software: the microprocessor is extremely competitive with standard hardware design and when you compare our prices to those of a minicomputer, well, there is no contest. If the AES-80 will do the job then it will definitely save dollars and that's the name of the game, isn't it?

Features

- 240 nanosecond full cycle time with bipolar memories.
- serial transfer up to 100,000 8-bit characters/second.
- high speed parallel I/O.
- capability of intermixing memory types in both data and instruction memories (bipolar, MOS, ROM, etc.).
- add-on I/O capability with standard MSU's.
- direct accessing to 1K of data memory and 2K of instruction memory.
- easy to use Program Development and Control Console, which supports a teletype or high speed tape reader. No other computing facilities required.
- expandable to 4K ROM and 4K RAM with additional memory accessible on an indirect basis.
- comprehensive set of 92 micro instructions.
- complete software support includes a cross assembler, minicomputer emulator, a self assembler and standard system development routines.
- multiprocessor configuration facilitated with relinquish bus command.

Applications

1 - OEM
2 - End user
3 - Research

- bread boards
- data communication
- process control
- terminals
- machine tool control
- remote concentration
- educational systems
- custom designs
- CRT controllers
- processing
- peripheral control
- switching
- front end preprocessing
- instrument systems
- automated testing
- airline reservations
- point of sale concentration
- simulators
- medical systems
- contour plotting
- spectrum analysers
- batch terminals
- sequence controllers
- line concentrators
- monitors
- lane counters

- **Cut your design time**
- **Eliminate bread boards**
- **Tradeoff software and hardware optimally**
- **Replace minicomputers in dedicated processing applications**
- **Reduce engineering in custom designs**

Architecture

The AES microprocessor is a bus organized machine designed around a data transfer concept. An 8-bit three-state processor bus is used as the main highway for data traffic between registers and the data memory. The source and destination of data travelling along the processor bus is under complete microprogram control. The basic microprocessor elements are shown in the block diagram.

Memory

A data memory word is 8 bits — chosen for its applicability in data communications. To reduce memory costs a 12-bit instruction word is used, a length which is extremely efficient yet allows single word literal instructions.

Commands from the read-only instruction memory control all aspects of the microprocessor operation and are executed in a single machine clock cycle. The 12-bit data from the ROM output bus is fed to an instruction decoder, the output of which determines the logic functions to be performed within the processor during the machine cycle. The ROM data bus also goes to the inputs of various registers within the microprocessor so that, depending upon the particular instruction decoded, literal data can be outputted directy from ROM.

Data is both read out of and written into data memory via the high speed 8-bit three-state processor bus.

Intermixing

A unique feature of the AES-80 microprocessor is the ability to intermix types of memories in both Data and Instruction Memories. Memory modules of 256 words each (8-bit data memory or 12-bit instruction memory) of many types may be intermixed.

- Bipolar ROM or RAM
- MOS ROM or RAM
- Core RAM
- Capacitive ROM
- Special purpose memories

Memory module selection depends on your requirements for processing speed, power failure protection, and program length. Depending on your volume requirements memories are chosen to minimize set up and unit costs. Intermixing provides flexibility for the solution of many application problems.

Arithmetic Logic Unit (ALU)

The arithmetic logic unit operates on two 8-bit variables the processor bus and its own output buffer accumulator. The ALU is capable of performing up to 16 logic operations on its two input variables and a variety of arithmetic operations, the most important being add and subtract. The mode of the ALU is selected by the ALU command register which is set by executing a single ALU literal instruction.

Registers-

P-Register: The 12-bit P (Program Counter) register indicates the address of the next instruction to be fetched out of instruction memory.

A-Register: The 12-bit A (Data Memory Address) register holds the address of the data memory cell being read from or written into.

LA-Register: The 8-bit LA (ALU Command) register is similar to the L-Register in that an 8 bit literal from ROM is loaded into it during an ALU literal instruction. The output of the LA-Register selects the operating mode of the ALU.

B-Register: The 8-bit (ALU Output Buffer) register is the ALU accumulator in which all results of the arithmetic and logical operations are stored.

U-Register: The U (Universal) register is an 8-bit parallel in, parallel out, serial in or serial out register. It is primarily used as the serial I/O buffer register.

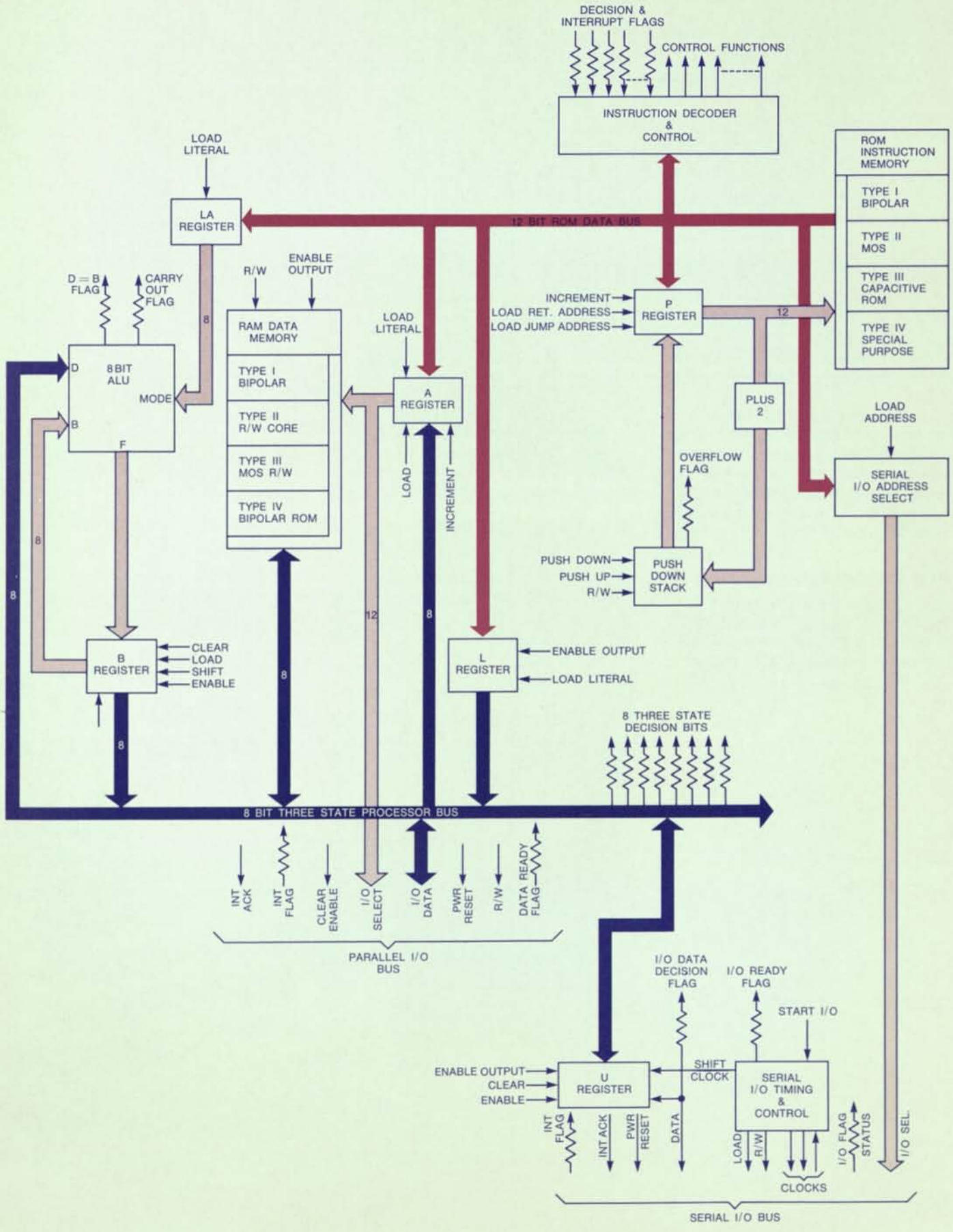
Push Down Stack

The AES microprocessor has an automatic push down stack for routine linkage. With this feature subroutines are written with a minimum of instruction time overhead.

Input / Output

The AES microprocessor serial I/O interface provides the necessary timing and control to communicate with both low and high speed peripheral devices. This I/O bus is used to transfer 8-bit serial characters at rates up to one character every 9.12 microseconds. When a start I/O instruction is executed, the clocking and transfer of I/O data then becomes automatic, with the microprocessor free to execute other instructions during the I/O interval. In addition to the serial I/O, a high speed parallel I/O capability is available. This is normally used as a means of providing fast hardware processor options such as multiply/divide or sine/cosine, etc. This bus is also used as a means of accessing a large data base such as a disc or magnetic tape unit where maximum data throughput is necessary.

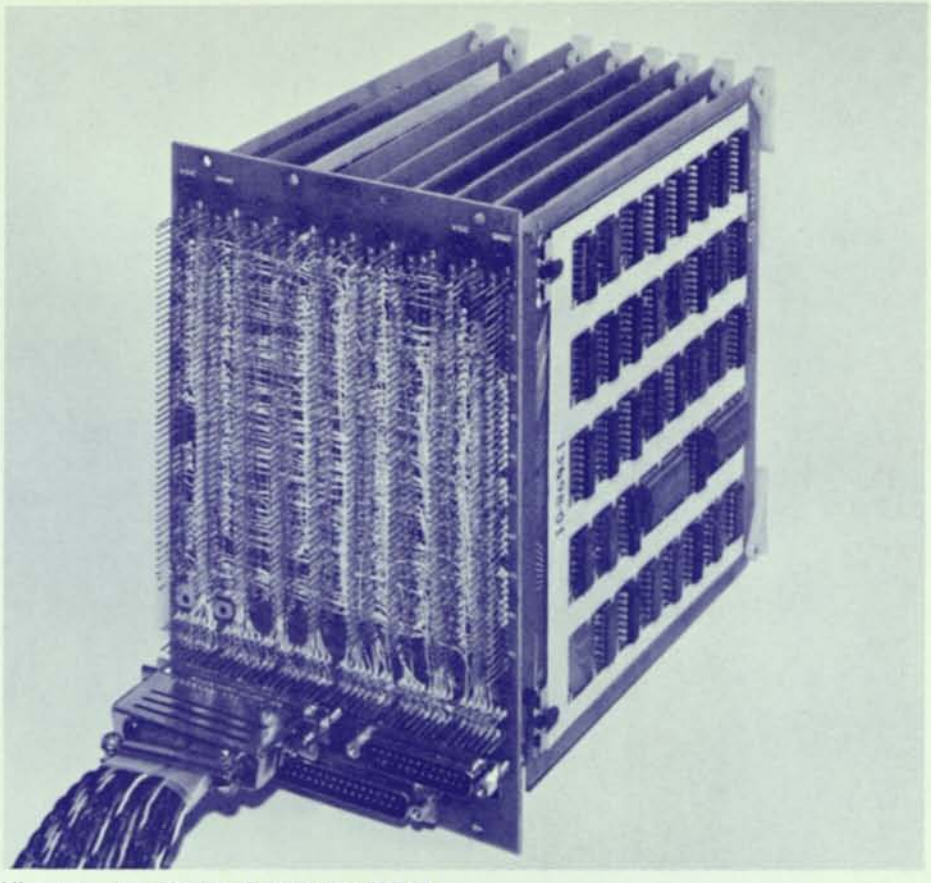
The AES-80 MICROPROCESSOR



Physical Configuration

The basic microprocessor MSU consists of a wire wrap mother board with nine connectors for insertable printed circuit boards. The microprocessor boards consist of 3 logic cards, one ROM card, and one RAM card. Four additional connector slots are available on the mother board. The first of these is reserved for an interface board for communication with the Program Development and Control Console used for program development and checkout.
One position is reserved for serial input/output control which comes in two versions — equipped with three-state inputs and outputs or differential line drivers and receivers. This card increases the load capability on the I/O bus. The remaining two slots may be filled by two parallel I/O interface cards or one interface card and one parallel I/O buffer expander.

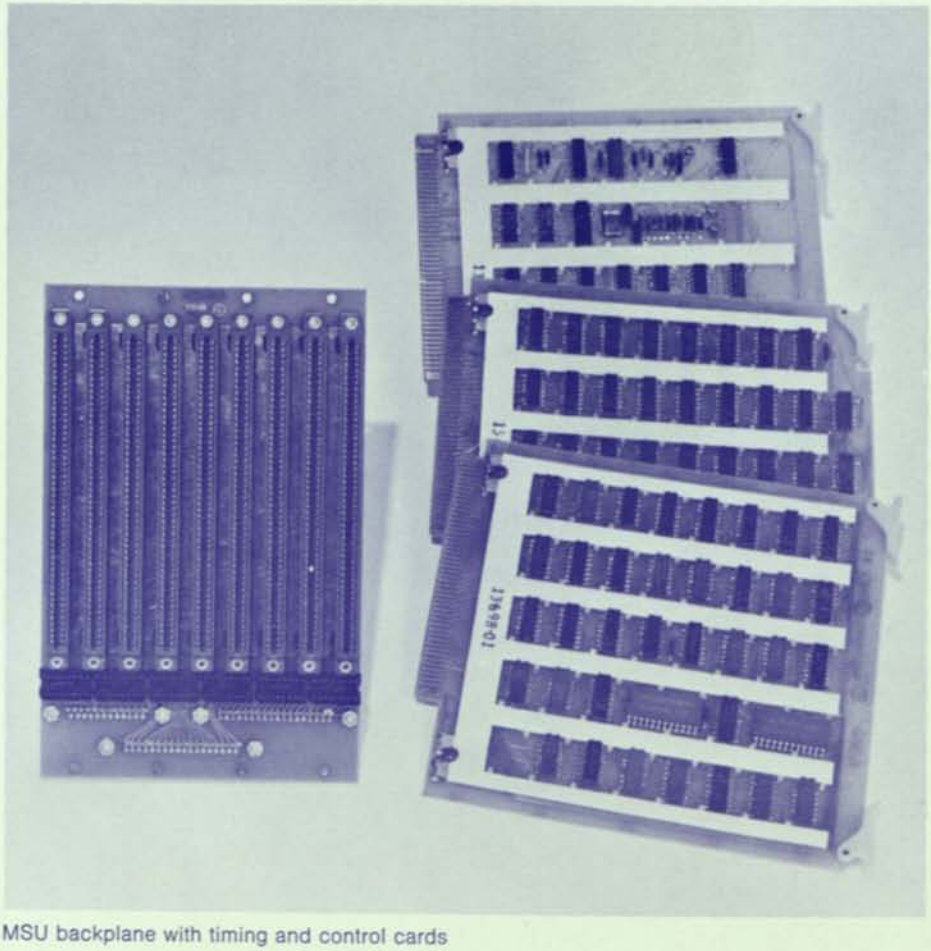
MSU back planes contain three standard connectors — a differential serial I/O connector and 2 parallel I/O connectors. MSU's may be "daisy chained" at distances up to 1000 feet.



Microprocessor Modular System Unit (MSU)

OEM Configurations

The AES-80 is available in packages specifically designed to the mechanical requirements of an OEM user.



MSU backplane with timing and control cards

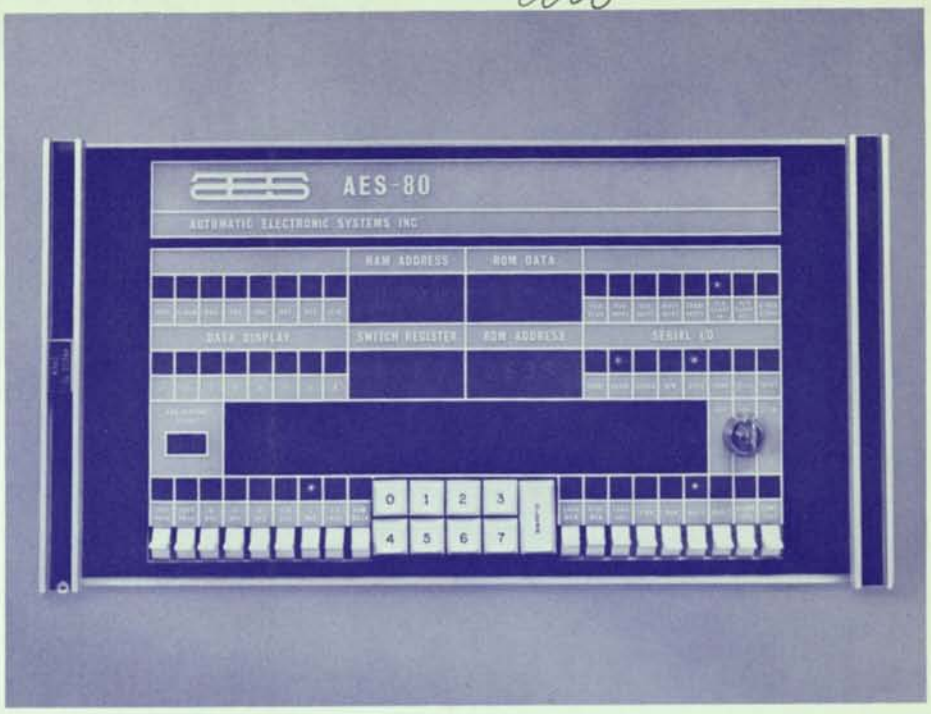
Support

Program Development and Control Console (PDCC)

A unique Program Development and Control Console is available for either program preparation and debugging prior to the "burning in" of a ROM or for passively monitoring an external microprocessor. The configuration used for program development is a microprocessor without ROM but including the interface card for the P.D.C.C. A second interface card is inserted in the Read Only Memory slot of the microprocessor and also connected to the P.D.C.C. Within the P.D.C.C. a high speed random access bipolar memory simulates the permanent ROM. Its contents, however, are alterable by loading instructions into it via the switch registers on the console, an ASR teletype, or a punched paper tape reader. The front panel of the P.D.C.C. contains a complete set of indicators, displays, switches, and controls for program development, including controls for program listing and tape loading.

When used to monitor a microprocessor installed in a larger system, the P.D.C.C. is primarily a passive display; active control is limited to halt, reset, single step and set command address. The microprogram is limited to fixed instruction in ROM unless read/ write memory is intermixed in the Instruction Memory.

A configuration of the P.D.C.C. which provides a convenient packaging arrangement for program development is one in which the microprocessor



Program Development and Control Console (PDCC)

cards themselves are inserted in the P.D.C.C. The combined console is ideal for software development or for prototyping.

Software

A cross assembler is available for software development which allows programming the AES-80 in symbolic language. This cross assembler is itself written in Fortran II or assembler language. Input can be from paper tape, punched card, magnetic tape, or disc. Output consists of an object program on paper tape, a comprehensive listing, and diagnostics. The Fortran version can be executed on any computer with 8K of core and a Fortran compiler; the assembler version is designed for one of the more popular minicomputers with 4K of memory.

A self assembler is available for execution on the Program Development and Control Chassis equipped with a 4K x 16-bit external memory. This configuration permits complete program development in assembler language. No other computer facilities are required.

Once a program is debugged a Truth Table generator is used to generate a truth table for every block of 256 words of ROM permitting easy manufacture of the permanent ROM.

AES-80 Specifications

Characteristics

- general purpose, 8 bit, byte oriented, programmable digital microprocessor
- serial I/O — up to 1 character every 9.12 microseconds
- parallel I/O — on data memory bus at internal processor speeds.
- physically and functionally expandable with Modular System Units

Memory

- 8 bit data memory, 12 bit instruction memory
- either memory available with 1 to 16 256 word modules intermixed from the following types:
  - 1 - Bipolar read/write — 240 nsec cycle time
  - 2 - Random Access read/write Core — 1 microsecond cycle time
  - 3 - MOS random access read/write memory—2 microsecond cycle time
  - 4 - Bipolar Read Only Memory — 240 nsec cycle time
  - 5 - Capacitive Read Only Memory — 240 nsec cycle time
  - 6 - Special purpose
  - 7 - Other

Serial I/O

- 8 address lines, 1 read/write line, 1 load strobe line, 3 I/O clock lines,
  1 flag status line, 1 interrupt flag, 1 IAK line, 1 serial data line, and 1 power on pulse line.
- automatic I/O of 8 bit data stream
- serial I/O can address 256 devices

Parallel I/O

12 address line, 8 I/O data lines, 1 write strobe, 1 read enable, 1 interrupt flag, 1 device ready flag, 1 interrupt line and 1 power on pulse line.

Instruction Set

ALU mode instructions	20
Data bus instructions	4
RAM address instructions	3
Accumulator instructions	4
Register instructions	6
Branch instructions	23
I/O instructions	16
Other instructions	16
TOTAL	92

Environment

0 to 70° C

Dimensions

- MSU backplane 5.25 x 8.5 inches
- Insertable PC boards 7.5 x 7.5 inches
- PDCC 19″W x 8.75H x 17″D

Power

9 amps maximum at 5 VDC for complete microprocessor with 1K x 8 Bipolar RAM and 1K x 12 Bipolar ROM.

Automatic Electronic Systems Inc.
5455 Pare Street
Montreal 309, P. Quebec, Canada
(514) 735-6581

AES Data Inc.
P.O. Box 143
St. Albans, Vermont 05478
(802) 524-3660

AES-90
CRT text editor
a giant step
into a new era
of cost effective
written
communication

Triples typing output – produces documents at less than ⅓ average cost

≈ 12-13K

The AES-90 combines computer power, stored program control and a CRT video display to increase typing productivity by 300% over conventional methods and by 50% over non-CRT text editors.
Just imagine, your typist can produce three times as much typewritten material each day. How?
First of all, the AES-90 virtually eliminates time consuming retyping. Every document is letter perfect. First time. Every time. Second, the AES-90 will type your letters at 175 words per minute, while your secretary performs other tasks: editing, proofreading or preparing new text. The AES-90 can also store over 100 pages of text in its electronic memory. Each document is automatically logged and indexed in an internal directory, ready for instant recall should future revisions or copies be needed.

Familiar keyboard, simple operation
The AES-90 is extremely easy to use. Your secretary simply types on a keyboard similar to that of the familiar IBM Selectric typewriter. As she types, an image of the document appears in large, clear characters on the CRT display. After typing a page, errors or omissions can readily be corrected with pushbutton ease. In fact, so great is the editorial power of the AES-90, that even the most heavily edited text is revised in seconds. Without retyping. Words, lines or paragraphs can be erased, inserted or moved to any location in the main body of the text with easy-to-execute pushbutton commands. Complex functions, such as right margin justification or merging two pages onto one, are accomplished with similar ease. All operations are automatic, and are performed on the displayed text. Once your typist is satisfied, the AES-90 prints out the document – error free – on its own high quality printer.

Replaceable memory discs store up to ~~50~~ 340 words each.

Features
Among the pushbutton editing features of the AES-90 are:

- **Automatic Title Centering** – shifts titles, subheadings or whole paragraphs to centre of page without counting or tabbing;
- **Automatic Underline** – underline words, or groups of words, on command;
- **Word Wrap-around** – eliminates manual carriage return and automatically drops split words to next line;
- **Automatic Numeric Alignment** – permits numbers to be entered from right to left, automatically aligning columns of figures to preset position;
- **Move** – allows transfer of whole blocks of text from one page to any position on same page, or on another.
- **List** – permits a personalized letter to be automatically directed to 2000 different addresses (perfect for direct mailing);
- **Justify** – automatically justifies (eliminates uneven right edge of) text;
- **Erase** – permits erasure of words, lines paragraph, or page, and replacement with other text (if desired);
- **Append** – allows operator to convert two separate pages into one two-column page;
- **Horizontal Page** – allows increase of page width to 128 characters from the normal 80;
- **Delete** – removes any amount of data from body of text; text automatically adjusts to fill vacated space; and
- **Insert** – allows any number of words to be inserted into middle of text; text automatically adjusts to accept new words.

Automatic Electronic Systems Inc.
570 McCaffrey Street
Montreal, Canada H4T 1N1
(514) 341-5030

AES Data Inc.
P.O. Box 143
St. Albans, Vermont 05478
(802) 524-3660

PRINTED IN CANADA

AUTOMATIC ELECTRONIC SYSTEMS INC.

THE AES-80 MICROPROCESSOR

ASSEMBLER REFERENCE MANUAL

Automatic Electronic Systems Inc.
5455 Pare Street
Montreal 309, P. Quebec, Canada
(514) 735-6581

AES Data Inc.
P.O. Box 143
St. Albans, Vermont 05478
(802) 524-3660

8Ø

MICROPROCESSOR

ASSEMBLER

REFERENCE MANUAL

AUTOMATIC ELECTRONIC SYSTEMS INC.,
5455 Pare Street
Montreal 309

TEL.: (514) 735-6581

TABLE OF CONTENTS

CHAPTER 1

1.Ø INTRODUCTION

The A.E.S. Standard Assembler allows programmers to write their microprocessor programs in a symbolic language rather than in machine language, the translation being performed by the Assembler.

The Assembler input is usually a paper tape, punched in ASCII code on an off line teleprinter.

The output consists of:

- another paper tape called the object program punched in the appropriate binary code required by the microprocessor program loader.

- a source program listing which displays pertinent information about the program such as: Line count, instruction addresses, memory content, error messages, etc.

A program can be broken down into several smaller programs which can be assembled separately.

This Assembler manual will be more profitable to the reader already familiar with the general architecture of the microprocessor as described in the hardware manual.

CHAPTER 2

2.Ø DESCRIPTION

2.1 ASSEMBLY PASSES

The Assembler is a 2 or 3 pass system depending on the types of peripheral devices available: if a KSR-33 Teletype is the only output device, 3 passes are required; if the computer system includes a printer and an independent paper perforator, then only 2 passes are required since the listing and the binary tape can be produced simultaneously.

A pass is defined as one complete reading of the source tape (s).

During pass 1, the Assembler builds a label table with the label (or names) found in some of the source statements.

During pass 2/3 the Assembler decodes the instruction mnemonic, searches in the label table the actual address associated with a label used as an operand, prints the listing and/or punches the binary tape and prints the eventual error messages.

2.2 SYMBOLIC ADDRESSING

There are several advantages to writing a program in Assembler rather than in machine language: One obvious one is the use of mnemonics. (E.g: F=D-B) instead of a hard-to remember binary code such as L&(or 1446 Octal). Another one is the inherent redundancy of an Assembler statement which allows error detection and yet another one is the automatic documentation of a program.

By far the most important advantage, however, is due to the use of symbolic addresses instead of explicitely defined absolute addresses. For descriptive purposes, assume a program is written in machine language and that a single instruction must be inserted (or deleted) into this program. Every instruction following the insertion (or deletion) point will be moved by one location, which means that all the

instructions referring to any moved instruction have to be modified if the original flow chart is to be respected. This task can be time consuming and is prone to numerous errors.

Consider the same program written in Assembly language. The source tape is edited instead of the binary tape. Once edited, this new source tape is processed by the assembler and all the memory reference instructions will be automatically updated when necessary.

The same procedure applies if a complete program is moved from one place in memory to another.

A symbolic address is defined by a label (or name) placed in the label field of a statement. To refer to this address the same label is used in the operand field of the referring instruction.

Labels appearing more than once in the label field of a program are considered in error. (Double Defined). Labels appearing in the operand field and never appearing in the label field are also considered in error (Undefined).

2.3 PROGRAM LOCATION COUNTER

For symbolic addressing and documentation purposes the Assembler maintains a program location counter.

This counter is initiated or reinitiated only by an ORIGIN Pseudo-Instruction (ORG) which defines the absolute address of the first instruction of a program or a section of program.

From there on the assembler will assign consecutive locations to every instruction it encounters. During the first pass the Assembler uses this counter to assign absolute addresses to every label it finds in the label field of an instruction and stores both, label and address, into the Label Table.

2.4 INSTRUCTIONS AND PSEUDO-INSTRUCTIONS

Instructions are defined as actual commands to the microprocessor to be executed at program run time. The Assembler always converts them into machine instructions punched on the object tape.

Pseudo-instructions are defined as commands to the Assembler itself. They are not punched on the object tape and they are executed by the Assembler at assembly time.

CHAPTER 3

FORMATS

3.1 STATEMENT FORMATS

There are three types of statements in the A.E.S. Microprocessor Assembler:

- -Comment statements
- -Instruction statements
- -Pseudo-Instruction statements

A statement is always contained in an ASCII line, i.e., a string of no more than 72 ASCII characters terminated by a RETURN CARRIAGE and a LINE-FEED character.

A statement is divided into several fields in order to distinguish between the different types of symbols.

For descriptive purposes, the characters within a line are numbered from 1 to 72 from left to right and designated by Ch x and the space character will be marked as ^ whenever it is necessary to show its place.

3.2 LABELS

A label is a word of 1 to 5 characters whose first one must be alphabetic or one of the following characters: @ [ ] ← ↑

The remaining characters can be any character except a space, a line-feed, a+, a-, a return carriage or a rubout. Non-printing characters will be accepted although it is not recommended to use them.

NOTE:

Since valid operands can be either octal numbers, decimal numbers or symbolic (labels) and a decimal number is of the form DXXXX e.g., D1Ø or D38Ø, labels shall not be of the form D followed by a number.

Examples of valid labels:

```
START
A1Ø
END
[#?
Z
```

Examples of invalid labels:

```
D2
128
#SUB
READER
```

LABEL FIELD

The label field exists in the instruction and in the pseudo-instruction statements. It always begins with Ch. 1 and ends with Ch. 5.

Ch. 6 must always be a space and acts as a separator between the label field and the mnemonic field. If the label has less than 5 characters, the remaining positions must be filled up with spaces.

If no label is used, then the label field must be filled up with spaces.

Examples:

```
START^D=M
^^^^^^F=D
      ...
Z^^^^^RET
      NOP
```

3.3 MNEMONICS

A mnemonic is a conventional word identifying one instruction or pseudo instruction.

The list of valid mnemonics is given in chapter 4.

Their length can be from 3 characters up to 12 characters but they must always begin in Ch. 7. No space is allowed within a mnemonic. The first space, return carriage, or line feed encountered by the Assembler while reading a mnemonic is interpreted as an "end of mnemonic" mark.

Examples:

```
^^^^^^D=F
^^^^^^JSR^SUBRO
LP1^^^F=D-B+1
```

3.4 OPERANDS

3.4.1 There are two types of instruction operands:

3.4.1.1 - Those used in Instruction Memory Reference Instruction (Jump, Jump subroutine, Jump if)
For these instructions the Operand field is separated by one (and only one) space from the Mnemonic field.

Examples:

```
^^^^^^JMP^2300
      JSR^SUBRO
      JIS,RTC^NEXT
```

3.4.1.2 - Those used in Data Memory Reference Instructions, Literal Register Instructions and Channel Reference Instructions. In this case the operand field follows immediately the last character of the Mnemonic (which is always an "=" character)

Examples:

```
^^^^^^L=D10
      L=RAMAD
      A=400
      CHL=TTY
```

3.4.2 There are also two types of pseudo-instruction operands:

3.4.2.1 The first one is used with the original definition and follows the same rules as the first-type of instruction operand (3.4.1.1)

Examples

```
^^^^^^ORG^100
^^^^^^ORG^LAST+1
```

3.4.3.2 The second type is used in the label definition statements following a RAM or an EXT pseudo-instruction. These two pseudos are used to define data memory labels RAM) or instruction memory labels external to the program (EXT).

In both cases the label definition operand begins at Ch. 7.

3.4 OPERANDS

3.4.1 There are two types of instruction operands:

3.4.1.1 - Those used in Instruction Memory Reference Instruction (Jump, Jump subroutine, Jump if) For these instructions the Operand field is separated by one (and only one) space from the Mnemonic field.

Examples:

```
^^^^^^JMP^23ØØ
      JSR^SUBRO
      JIS,RTC^NEXT
```

3.4.1.2 - Those used in Data Memory Reference Instructions, Literal Register Instructions and Channel Reference Instructions. In this case the operand field follows immediately the last character of the Mnemonic (which is always an "=" character)

Examples:

```
^^^^^^L=D1Ø
      L=RAMAD
      A=4ØØ
      CHL=TTY
```

3.4.2 There are also two types of pseudo-instruction operands:

3.4.2.1 The first one is used with the original definition and follows the same rules as the first-type of instruction operand (3.4.1.1)

Examples

```
^^^^^^ORG^1ØØ
^^^^^^ORG^LAST+1
```

3.4.3.2 The second type is used in the label definition statements following a RAM or an EXT pseudo-instruction. These two pseudos are used to define data memory labels RAM) or instruction memory labels external to the program (EXT).

In both cases the label definition operand begins at Ch. 7.

Examples:

```
^^^^^^RAM
WORD1^74
LABEL WORD1+1
      ---
^^^^^^EXT
SUB1^^2ØØØ
S2^^^^SUB1+D2Ø
      ---
      ORGXXX
```

3.4.3 COMMON RULES FOR ALL OPERANDS:

3.4.3.1 No space allowed within an operand.
Operands can be either numeric or symbolic or a combination of both:

3.4.3.1.1 Numeric operands:

- A single numeric term:
- Octal term: a number of no more than 4 octal digits preceded, optionally by a sign (+,-)
- Decimal term: a number of no more than 4 decimal digits preceded by the letter D itself preceded, optionally, by a sign (+,-)

Examples
```
377
+377
-1
D1Ø8
+D1Ø8
-D1Ø8
```

3.4.3.1.2 Symbolic operands:

A label which must be defined elsewhere in the program, i.e., it must appear once and only once in the label field of the program (ch. 1 to ch. 5).

3.4.3.1.3 Symbolic operands modified by a numeric displacement:

A defined label immediately followed by a numeric term. The resulting address is the address assigned to the label displaced by an amount equal to the numeric term.

Examples

```
^^^^^^A=BUFF+2
```

Let's assume that BUFF has been defined as data memory address 1ØØ, then the A - register will point to address 1Ø2 after execution of this instruction.

NOTE:

Since A=A+1 is a special instruction to increment the A - Register, A can be used as a label as long as it is not used with a displacement of + 1.

3.4.4 SPECIAL RULES FOR PSEUDO INSTRUCTION OPERANDS

Since instruction operands are evaluated during the second pass they can refer to labels defined anywhere in the program.

However pseudo instructions operands are always evaluated during the first pass and therefore, if they are symbolic, they must refer to a label defined BEFORE this pseudo is encountered.

Examples: Valid pseudo instruction operands

```
      ...
LASTW^NOP
      RAM
DATA1^1Ø4
DATA2^DATA1+1Ø
DATA3 DATA1+2Ø
      ORG^LASTW+1
      ...
        Illegal pseudo-instruction operands
      ...
      RAM
DATA2^DATA1+1Ø
DATA3^DATA1+2Ø
DATA1^1Ø4
      ORG LASTW+1
      ...
LASTW^NOP
```

3.5 COMMENTS

3.5.1 Comment Statement

An entire line can be devoted to comments, i.e.,

information useful to programmers but ignored by the assembler. To do this an asterisk (*) must be the first character of the line.

Example:

```
*^^THIS IS A COMMENT LINE
```

2.5.2 COMMENT FIELD

The right hand part of an instruction or pseudo-instruction statement can be used for additional comments: in both cases the comment must be separated from the operand, or from the mnemonic if there is no operand, by at least one space.

Examples:

```
PRG1^^F=D^COMMENT
      JSR^SUBRO^COMMENT
      RET            COMMENT
```

3.5.3 COMMENT SIZES

Since the printed listing includes information not originally punched in the source tape the number of characters allowed for a comment is limited accordingly:

- A comment statement cannot exceed 67 characters.
- An instruction statement cannot exceed 54 characters.
- A pseudo-instruction statement cannot exceed 67 characters.

Otherwise the comment characters in excess won't appear on the listing.

CHAPTER 4

MACHINE INSTRUCTIONS

4.1 DATA MEMORY REFERENCE INSTRUCTIONS

A single instruction can set a 1Ø bit address in the 12 bit address register (A).

A=xxx

The two most significant bits (bit 1Ø, 11) of the A register are left unmodified by the A= instruction. Thus one can directly address any one location of a 1Ø24 word data memory page.

The quantity xxx can be either Octal, Decimal or Symbolic. In any case it cannot exceed 1777 (8).

Examples:
```
A=77
A=D1Ø8
A=FLAG+3
```

In order to reach the other 1Ø24 word pages and also to load the A register with a computed address, two other instructions are provided.

AL=D	The data bus is loaded into the 8 LEAST significant bits of the A-register
AH=D	Bits 8, 9, 1Ø, 11 of the A-register are loaded with bits Ø, 1, 2, 3 of the data bus.

Examples: 1) Retrieval of one element of an array

```
A=PTR     Pointer address
D=M       Pointer on data bus
AL=D      Pointer in A-Register
          At this point, retrieved data is on the
          data bus.
```

2) Same requirement but the pointer is now in page Ø and the array in page 2 (256 Wd pages).

```
A=PTR     Pointer address
D=M       Pointer on data bus
F=D       Set ALU for direct load
B=F       Pointer in B-register
L=2       Literal ØØ2
D=L       On data bus
```

AH=D	Load bit 11, 1Ø, 9, 8 of the A-Register with Ø2
D=B	Pointer on data bus
AL=D	Load the 8 LSB of A-register with the pointer.

The A-Register can be incremented

A=A+1

Thus permitting easy scanning of an array:

Example: Find the first even element of an array

```
^^^^^^A=AR(Ø)-1      first element address minus 1
      D=M            Address on data bus
LOOP^^A=A+1          element on data bus
^^^^^^JIS,DBØ LOOP   Test least significant bit
```

4.2 DATA BUS INSTRUCTIONS

Four mutually exclusive latching instructions are available which enable programmers to choose the source of the data present on the bus.

D=L	Literal Register (L) on data bus
D=M	Data Memory location pointed by A on data bus
D=U	Universal Register (U) on data bus
D=B	ALU Register (B) on data bus

4.3 INSTRUCTION MEMORY REFERENCE INSTRUCTION

They are divided into 3 groups. All of them require two memory locations and are executed in two machine cycles.

4.3.1 UNCONDITIONAL JUMP

JMP xxx

The program flow is altered:
The operand defines the address of the next instruction to be executed.

4.3.2 JUMP TO AND RETURN FROM SUBROUTINE

JSR xxx

The operand defines the address of the first instruction of the subroutine.
The return address is automatically stored in the upper location of the push down stack.

Note: Subroutines can be nested up to 16 levels

RET xxx

The next instruction to be executed is the one following immediately the RET instruction. Then the program flow is altered: the last address stored in the push down-down stack defines the address of the next instruction to be executed. The push down stack is pushed-up.

Example: Increment a word in data memory

```
^^^^^^A=WORD        Address of word
^^^^^^JSR^INC       Increment it
      ...
*
*  INCREMENT SUBROUTINE
*
INC^^^D=M      Word on bus
       F=D+1   Set ALU for increment operation
       B=F     LOAD B-Register with incremented value
       D=B     Incremented word on bus
       RET     Initiate the return from subroutine
       M=D     Store incremented word in original address.

*
```

4.3.3 CONDITIONAL JUMP INSTRUCTIONS

They are of the form:

JIS,yyy^xxx or JIC,yyy^xxx
where yyy is a mnemonic defining the condition to be tested
and xxx the address of the next instruction to be executed if the condition is met.
JIS stands for Jump If Set (logical 1)
JIC stands for Jump If Clear (logical Ø)

JIS,BR7 xxx or JIC,BR7 xxx

Jump if B-Register bit-7 is Set/Clear.

Note: This bit can be considered as the sign of a 7 bit word in two's complement form.

JIS,CRY xxx or JIC,CRY xxx

Jump if ALU carry output flag is Set/Clear.
This flag is set whenever the result of an ALU arithmetic operation produces an overflow.

Example: Check if the quantity X present on the data bus is Ø.

```
F=D-1          Set ALU for X-1
JIS,CRY ZERO   If X was Ø, then X -1 = -1 and
               the carry flag is set.
```

JIS,D=B xxx or JIC,D=B xxx	Jump if data bus and B-Register are equal/ different. ALU must be in the F = D-B-1 mode

Example: Check if 2 quantities in memory are equal

```
      A=WORD1        Address of first word
      D=M            Word 1 on data bus
      F=D            Set ALU for a direct load
      B=F            Load word 1 into B-register
      A=WORD2        Word 2 on data bus
      F=D-B-1        Set ALU for compare
      JIS,D=B EQUAL  If equal, the compare flag is
                     set and the jump is executed
```

JIS,DBn xxx or JIC,DBn xxx	Jump if bit n of the data bus is set/clear where n is Ø to 7 (Ø for the least significant bit)

Example: Test an internal flag in position 4 of the data memory word FLAGS

```
A=FLAGS
D=M
JIS,DB4 FLSET   If bit 4 of the word FLAGS
                is set, then jump is executed
```

JIS,PDS xxx or JIC,PDS xxx	Jump if push down stack overflow flag is set/clear.

The remaining conditional jumps will be described in the Input/Output instruction section.

4.3.4 INSTRUCTION MEMORY PAGES

Since 11 bits are used to specify an instruction memory address, we can address directly any location in a 2Ø48 word page. To jump across a page boundary, a page instruction is provided.

PG=Ø PG=1	The next jump (any type) instruction will be made to the location defined by the jump operand but within the specified page (Ø or 1)

Example: Let's assume we are in page Ø
PG = 1
...
JSR 31Ø Jump subroutine to address 431Ø

4.4 ARITHMETIC - LOGIC UNIT INSTRUCTIONS

Generally an arithmetic or logic operation will be performed in two steps:

1) Set the ALU to the selected operation
2) Save the ALU output (called F) in the B-register.

Note: The ALU will remain set for an operation until it is changed by another ALU instruction.

Special characters used in the ALU instruction mnemonics

#	logical OR operation
.	logical AND operation
↑	logical EXCLUSIVE OR operation
'	logical COMPLEMENT of preceeding quantity: B' or (D#B)'
+	arithmetic ADDITION (2's complement)
-	arithmetic SUBTRACTION (2's complement)

Expressed with these symbols the De Morgan's theorem becomes:

(A#B)'=A'.B'
(A.B)'=A'#B'

Knowing these symbols, the ALU mnemonics are self explanatory.

4.4.1 LOADING INSTRUCTIONS

F=D	The output F of the ALU is equal to the : Data bus
F=D'	Data bus complement
F=B	B-register
F=B'	B-register complement
F=-1	Octal 377
F=Ø	Octal ØØØ

4.4.2 LOGIC INSTRUCTIONS

Logical OR			Equivalent to
F=D#B	or	F=B#D	(D'.B')'
F=D#B'	or	F=B'#D	(D'.B)'
F=D'#B	or	F=B#D'	(D.B')'
F=D'#B'	or	F=B'#D'	(D.B)'

Logical AND			
F=D.B	or	F=B.D	(D'#B')'
F=D.B'	or	F=B'.D	(D'#B)'
F=D'.B	or	F=B.D'	(D#B')'
F=D'.B'	or	F=B'.D'	(D#B)'

Exclusive OR

F=D↑B	or	F=B↑D
F=D↑B'	or	F=(D↑B)'

4.4.3 ARITHMETIC INSTRUCTIONS

Addition

F=D+D	or	F=2D
F=D+B	or	F=B+D
F=D+D+1	or	F=2D+1
F=D+B+1	or	F=B+D+1
F=D+1		
F=D+B'		
F=D+B'+1		

Subtraction

F=D-B	F=D-B'
F=D-B-1	F=D-B'-1
F=D-1	

4.4.4 COMBINED LOGIC AND ARITHMETIC INSTRUCTIONS

The logical operation is executed before the arithmetic one.

F=D#B+D	F=D#B'+D
F=D#B+1	F=D#B'+1
F=D#B+D+1	F=D#B'+D+1
F=D.B+D	F=D.B'+D
F=D.B+D+1	F=D.B'+D+1
F=D.B-1	F=D.B'-1
F=D#B+D.B'	F=D#B'+D.B
F=D#B+D.B'+1	F=D#B'+D.B+1

4.4.5 SHIFT ROTATE INSTRUCTIONS

F=BSL	F = B-register shifted left by one bit. LSB of F is set to Ø.
F=BRL	F = B-register rotated left by one bit. Thus the MSB of B becomes the LSB of F. Both instructions require that the B-register output is enabled onto the data bus (D=B).
B=BRR	B is ready for a one bit right rotation independently of the ALU output F.

4.4.6 B-REGISTER INSTRUCTIONS

B=Ø	Clear B-register
B=F	Load B-register with ALU output
B=FH	Load B-register with ALU output 4 most significant bits.
B=FL	Load B-register with ALU output 4 least significant bits.

Note: For B=FH and B=FL the remaining bits of the B-register are left unchanged.

Example: Convert one ASCII digit to binary

```
A=DIGIT     Digit address
D=M         Digit on data bus
F=D         Set ALU for direct load
B=Ø         Clear the B-register
B=FL        Load the 4 LSB in the B-register
```

Or, if the digit is already present in the B-register

```
F=Ø
B=FH        Load Ø's into the 4 MSB of the
            B-register.
```

Note: If the ALU was set in the B=BRR mode then a B=F (or B=FL or B=FH) instruction will actually rotate the B-register one bit right.

4.5 LITERAL AND UNIVERSAL REGISTER INSTRUCTIONS

L=xxxxx	Set the literal register to the value of the operand. Since the L-Register has 8 bits, the operand, once evaluated by the Assembler, should not be greater than 377 (8). If it is, the 8 LSB will be used as operand. No diagnostic will be given.

Examples:

```
L=-1
L=D255
L=377
L=1777
L=LABEL
```

All result in the loading of Octal 377

U=Ø	Clear the universal register.
U=U#D	Load the logical OR of the data bus and the present U-register content.

4.6 INPUT/OUTPUT INSTRUCTIONS

4.6.1 SERIAL I/O

CONTROL LINES:

CLK=Ø CLK=1	Clock line low or high
LD=Ø LD=1	Load line low or high
R/W=R R/W=W	R/W Line low (read) or high (write)

For serial I/O all 8-bit data words pass through the Universal Register.

CHANNEL SELECTION

CHL=xx	xx = any operand (octal, decimal or symbolic) Its value must not exceed 31.

REGISTER SELECTION (One at a time)

RG=x	x = one octal digit (Ø to 7)
RG=B	In this case the register is selected by the 3 least significant bits of the B-register.

TIMING

SIO	Stand for Start Input-Output automatic transfer between the U-Register and the addressed I/O device register. This is a strobe function.
JIS,IOR xxx or JIC,IOR xxx	Test the I/O Ready flag. IOP flag is normally set. The SIO instruction clears it. When the transfer is completed the IOR flag is set again.

Examples: Input one 8 bit word from channel 1Ø, register Ø

```
        ...
        CHL=1Ø          Select channel
        RG=Ø            Select register
        R/W=W
        LD=1
        CLK=1           Load data into I/O register
        CLK=Ø
        LD=Ø
        R/W=R           Initiate serial transfer between I/O
        SIO             Register and U-Register
WAIT^^JIC,IOR WAIT        Finished? No loop
        D=U             Yes, data on bus
        ...
```

Output 8 8-bit words to channel Ø, registers Ø to 7 from 8 consecutive locations in data memory

```
        CHL=Ø           Select channel
        A=AR(Ø)         Address of first word
        B=Ø             Clear B-register
        F=D+1           Set ALU for increment
LOOP^^  D=M             Data on data bus
        U=Ø
        U=U#D           Data in U-Register
        RG=B            Select I/O register
        R/W=W           Initiate
        SIO             Serial transfer
WFLAG^JIC,IOR WFLAG       Wait if I/O Ready flag not set
        R/W=R
        A=A+1           Increment array pointer
        D=B
```

```
B=F             Increment B-register
JIC,DB3 LOOP      Done it 8 times ?  No, loop
...             Yes, exit
```

4.6.2 PARALLEL I/O

No special instruction is devoted to parallel I/O since the I/O registers are considered as data memory locations and, therefore, are operated by the M=D or D=M instructions.

Example: For descriptive purposes, assume the microprocessor is used with the Bose-Chaudhury Error code generator option. This option uses one input register (new data) and one output register (new Bose-Chaudhuri Error code) and their addresses are respectively $6ØØ1_{(8)}$ and $6ØØØ_{(8)}$. New data is supposed to be initially in the U-Register and we want to get the result (new Error Code) in the B-register.

```
L=14
A=1
D=L
AH=D    Set A-register to 6ØØ1
D=U     New data on data bus
M=D     Compute new Error Code
A=Ø     Set A-Register to 6ØØØ
D=M     Error code on bus
F=D
B=F     New Error code in B-register
```

4.6.3 INTERRUPT INSTRUCTIONS

JIS,SIN xxx or JIC,SIN xxx	Jump If the Serial Interrupt flag is Set/Clear.
JIS,PIN xxx or JIC,PIN xxx	Jump If the Parallel Interrupt flag is Set/Clear.
JIS,INT xxx or JIC,INT xxx	Jump If the Master Interrupt flag is Set/Clear.

Note: This flag is set whenever any one of the decision flags, strapped into the interrupt structure, is set. These decision flags are usually chosen from the following.

Push Down Stack flag
Console Alarm flag
Power Fail flag
Relinquish Bus flag
Parallel I/O Interrupt flag
Serial I/O Interrupt flag
Real Time Clock flag

IAK=1 IAK=Ø

Interrupt Acknowledge. Used to clear the device status flag if it is an interrupting device. If the device was the only interrupting device IAK will also clear the serial or parallel interrupt flag. To clear a serial I/O interrupt and status flag:

```
CHL=x
RG=y
R/W=W
IAK=1
IAK=Ø
```

To clear a parallel I/O interrupt and status flag

```
A=z
D=M
IAK=1
IAK=Ø
```

In both cases, if the status flag is not connected to the interrupt structure, only the device status flag will be cleared.

JIS,SFL xxx or JIC,SFL xxx

Jump If Serial I/O Status Flag Is Set/Clear.

JIS,PFL xxx or JIC,PPL xxx

Jump If Parallel I/O Status Flag is Set/Clear.

Prior to testing a status flag, the proper device must be addressed: channel and Register selection for serial I/O Data memory Address Register (A) for parallel I/O.

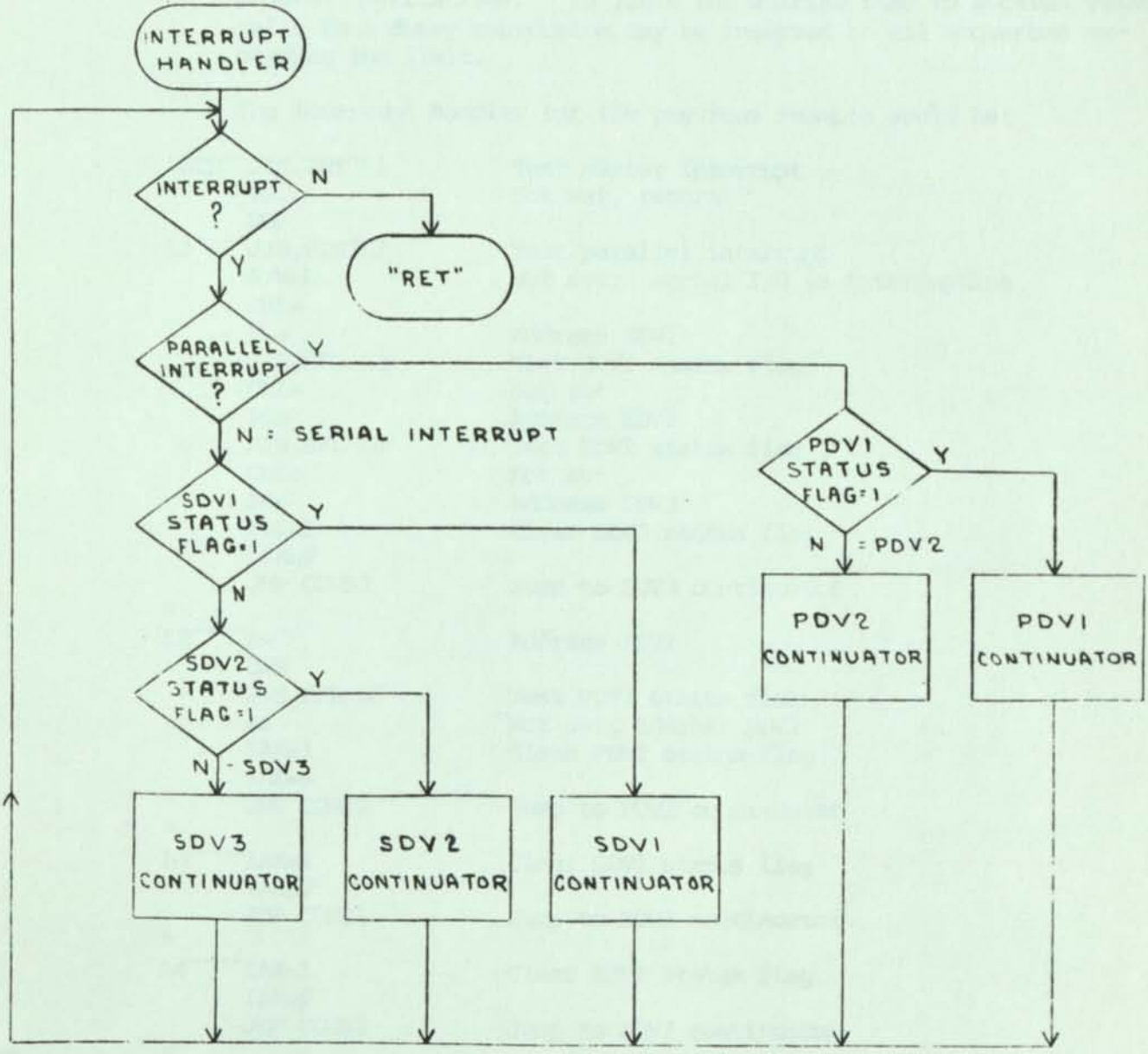
Example: For descriptive purposes, assume a configuration of 3 devices connected to the serial I/O and 2 devices connected to the parallel I/O. All have status flags and are connected to their respective interrupt systems. They are called SDV1, SDV2, SDV3 and PDV1, PDV2 respectively. SDV1, SDV3 and PDV1 are for input, SDV2 and PDV2 are for output. The software system will consist of a background program called "PROCESS" and a foreground program called "ACQUISITION". PROCESS will initiate I/O operations and ACQUISITION will service these I/O operations once they have been initiated. Every device handler is composed of two parts:

1) the Initiator, which will be part of PROCESS
2) the Continuator, which will be part of ACQUISITION

For example simplicity we will assume that ACQUISITION is non-interruptable and thus a reentrant Interrupt Handler is not needed. Communication between PROCESS and ACQUISITION will be accomplished through data buffers and software flags: one BUSY flag for every device handler. This flag will be automatically set by the Initiator and cleared

by the continuator when the I/O operation is completed.
Thus PROCESS is given the ability to know the state of the I/O operations.
Devices will be serviced according to the following priority:

1 PDV1 input
2 PDV2 output
3 SDV1 input
4 SDV2 input
5 SDV3 output



To allow an interrupt to take place all subroutines used by PROCESS return to the calling program by making a direct jump to the interrupt handler entry point subroutine example:

```
SUBRO^...
      ...
      JMP^INTH
```

Thus the maximum time an interrupt request is kept waiting will be determined by the longest instruction sequence without a jump to the interrupt handler. Our experience shows that the average sequence will be around 10 instructions and, without any special attention it has never exceeded 40 instructions in any of our past program applications. To limit the waiting time to a given value, calls to a dummy subroutine may be inserted in all sequences exceeding the limit.

The Interrupt handler for the previous example would be:

```
INTH^^JIS,INT^L1        Test master Interrupt
      RET               Not set, return
      NOP
L1^^^^JIS,PIN^L2        Test parallel interrupt
      R/W=W             Not set:  serial I/O is interrupting
      CHL=
      RG=               Address SDV1
      JIS,SFL L3        Test SDV1 status flag
      CHL=              Not set
      RG=               Address SDV2
      JIS,SFL L4        Test SDV2 status flag
      CHL=              Not set
      RG=               Address SDV3
      IAK=1             Clear SDV3 status flag
      IAK=0
      JMP CONS3         Jump to SDV3 continuator
*
L2^^^^A=                Address PDV1
      D=M
      JIS,PFL L5        Test PDV1 status flag
      A=                Not set, Address PDV2
      IAK=1             Clear PDV2 status flag
      IAK=0
      JMP CONP2         Jump to PDV2 continuator
*
L3^^^^IAK=1             Clear SDV1 status flag
      IAK=0
      JMP CONS1         Jump to SDV1 continuator
*
L4^^^^IAK=1             Clear SDV2 status flag
      IAK=0
      JMP CONS2         Jump to SDV2 continuator
*
L5^^^^IAK=1             Clear PDV1 status flag
      IAK=0
      JMP CONP1         Jump to PDV1 continuator
*
```

4.6.4 OTHER I/O INSTRUCTIONS

JIS,IOD xxx or JIC,IOD xxx	Jump If serial I/O bus data line is Set/Clear. The following instructions are independent of the standard I/O structure.
JIS,ALM xxx or JIC,ALM xxx	Jump If external ALARM flag is Set/Clear
JIS,PWR xxx or JIC,PWR xxx	Jump If POWER fail interrupt flag is Set/Clear
JIS,RTC xxx or JIC,RTC xxx	Jump If Real Time Clock flag is Set/Clear
JIS,RBF xxx or JIC,RBF xxx	Jump If Relinquish Bus Flag is Set/Clear
RBC or EBC	Disable or Enable all serial and parallel I/O line drivers and receivers (Relinquish or Enable Bus Control)

4.6.5 HALT INSTRUCTIONS

HLT=xx	xx=octal number between Ø and 17. The halt instruction is available only with the maintenance and control chassis connected to the micro-processor. Otherwise it is treated as a NOP instruction.

4.6.6 MASTER RESET

RST	Strobe the micro-processor into the PORC condition.

CHAPTER 5

PSEUDO-INSTRUCTIONS

5.1 PROGRAM ORIGIN

During the first pass if the assembler encounters no origin statement before the first non-comment statement, it will request an origin on the teletype. The answer to be keyed-in must be an octal or a decimal number in the range Ø-7777(8).

This feature allows programmers to decide the program location at Assembly time if they choose to do so.

An origin statement has the form:

ORG xxxxx

where xxxxx can be either octal, decimal or symbolic.

In any case, when this operand is evaluated it must yield a result within the range Ø-7777(8)

Any number of ORG statements can be inserted in a program.

Examples.

```
        ORG 1ØØ
        ...             Program part I
        ...
        RET
LSTWD   NOP
        ORG D1Ø
        ...             Program part II
        ...
        ORG LSTWD+1
        ...             Program Part III consecutive
                        to Part I.
```

5.2 DATA MEMORY ADDRESSES

RAM

Since the data memory is usually built with read/write Random Access Memory a "RAM" pseudo-instruction has been created:

```
^^^^^^RAM
LABEL^xxxxx
.............
```

The Assembler will consider all non-comment statements following a "RAM" as pseudo-instructions defining the address associated with a label.

The operand xxxxx of a Label definition pseudo-instruction can be either octal, decimal or symbolic.

The Assembler will stay in the mode where it treats non-comment statements as Label definitions until it encounters an ORG, an EXT or an END statement.

Example

```
LSTWD NOP
*
      RAM
LAB1^^144
LAB2^^LAB1-1Ø
FLAGS^D3
*
*
        ORG LSTWD+1
```

5.3 EXTERNAL LABELS — EXT

When a label is not declared anywhere in the program as an operand (Example: the entry point of a subroutine not included in the program), an "EXT" pseudo-instruction is used to define it.

```
^^^^^^EXT
LABEL^xxxxx
. . . . . .
```

The assembler will consider non-comment statement following an "EXT" as pseudo-instructions defining the address associated with a label.

The operand xxxxx of a Label definition pseudo-instruction can be either octal, decimal or symbolic.

The Assembler will stay in the mode where it treats non-comment statements as Label definitions until it encounters an ORG, a RAM, or an

END statement.

Example

```
...
JSR SUBRO
CHL=TTY
*
EXT
*
SUBRO^277
SUB2^^SUBRO-1
TTY^^^3Ø
      RAM
      ...
```

5.4 END OF PROGRAM

END

A source program can be made of several pieces of tape. Each of them with a leader and a trailer (at least 5 inches of null characters).

As long as the Assembler does not encounter an "END" statement it will assume that there are additional source tapes to come.

As soon as it finds the "END" statement the Assembler stops reading and considers the pass as finished. Anything after an END statement is ignored.

CHAPTER 6

ASSEMBLER INPUT/OUTPUT

6.1 SOURCE PROGRAM

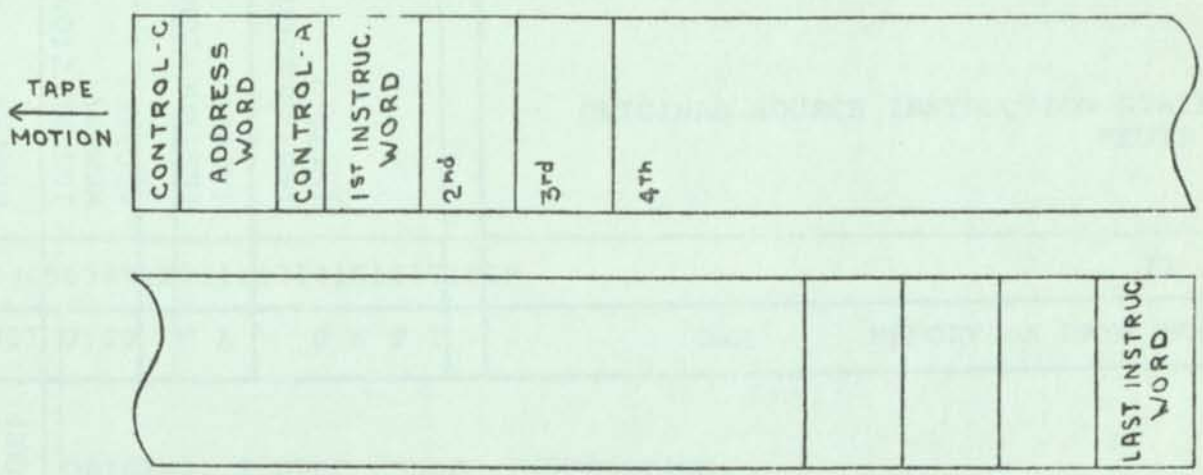
A source program can contain up to 9999 statements.

When a source program is divided into several pieces of tape, the last (non-null) character of any tape must be a return-carriage or a Line feed.

On reader input, a rub-out character is ignored. On keyboard input (Origin request), The Assembler ignores the line which contains a rub-out.

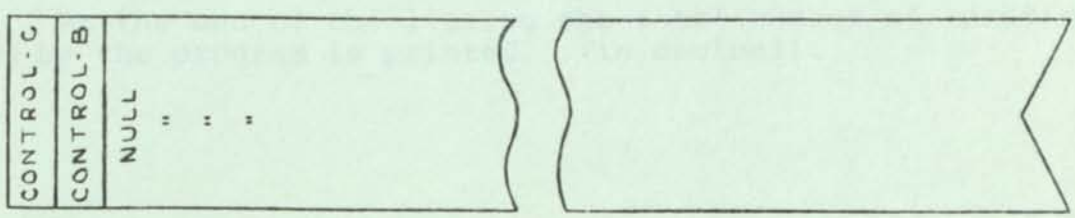
6.2 BINARY TAPE FORMAT

The binary tape is built with consecutive blocks and, of course, a leader and a trailer (null characters). Each block has the following format:



NOTE: There is a block for every ORG in the source program

The trailer has the following format



Every character is provided with even parity (i.e: the sum of all 8 bits must be even).

The address word and the instruction words have the following format:

	FIRST CHARACTER								SECOND CHARACTER							
	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
WORD BIT 11	1Ø	9	8	7	6				5	4	3	2	1	Ø		

Character BIT 7 is always the complement of BIT 6 in order to have a printing character rather than a non-printing one.

See Appendix A for the binary tape character set.

6.3 LISTING FORMAT

Pages are numbered from 1 to 99. The line number is reset to 1 at the beginning of a source tape.

LINE NUMBER	INSTRUCTION MEMORY ADDRESS		INSTRUCTION (ASCII)	INSTRUCTION (OCTAL)		ORIGINAL SOURCE INSTRUCTION STATEMENTS
1234	56789	1Ø	1112	131415161718	19	72
ØØØ1	Ø1ØØ		F A	Ø 6 Ø 1		D=M MEMORY oa DATA BUS

LINE NUMBER	ORIGINAL SOURCE PSEUDO-INSTRUCTION OR COMMENT STATEMENTS
ØØ1Ø	ORG 4Ø
ØØ11	* PROGRAM

At the end of the listing the total number of locations used by the program is printed. (in decimal).

6.4 ERROR MESSAGES

The Assembler will print error messages when it finds syntax errors or it cannot recognize a label or a mnemonic or it cannot evaluate an operand:

These messages are:

ILLEGAL (format error, RAM or EXT block error, literal, channel, or halt, overflow, mnemonic error.

UNDEFINED (an operand cannot be evaluated).

ADDRESS ERROR (address above 3777 for instruction memory or above 1777 for data memory)

LABEL ERROR (label format).

DOUBLE DEFINED (label)

SYMBOL TABLE OVERFLOW

The faulty statement follows the error message.
The total error count for a program is given at the end of every pass (in decimal).

6.5 OPERATING INSTRUCTIONS

6.5.1

Data General Nova Computer: PAPER TAPE SYSTEM Minimum Configuration: 1 NOVA computer with 4K of Memory
1 Teletype ASR-33

LOADING 1-Turn computer ON, and, if available, set the fast tape punch and/or line printer to on.

2-Turn teletype ON-LINE.

3-Set switch Register to 7777 for a 4K Nova computer
or 17777 for a 8K Nova computer
or 27777 for a 12K Nova computer
or 37777 for a 16K Nova computer

4-Put the configured "CROSS-ASSEMBLER" binary tape into the tape reader (teletype reader or, if available, fast tape reader).

5-Press "Reset"

6-Press "Start", the Computer should load the cross Assembler

PASS 1 7-Set Switch Register to 4ØØ

8-Place the source tape into the tape reader

9-Press "Reset"

1Ø-Press "Start"-Computer should read the first tape.

11-If more than 1 tape is to be input (for 1 given program), repeat step 8 and press "Continue" for every tape.

After the last tape, which must contain an "END" statement, the computer will print "N" ERRORS.

PASS 2 12-choose your options:

sw Ø	sw 1	
1	Ø	Listing
Ø	1	Binary tape
1	1	Listing & Binary tape simultaneously providing they come from 2 independent devices.
Ø	Ø	No listing, no binary type: only the error messages.

13-Place the source tape into the tape reader.

14-Press "Continue"

15-If more than 1 tape, repeat steps 13 and 14

16-After the last tape, the computer is ready to execute another Pass 2 (Steps 12 to 16)

NOTE: During Pass 2, if the Listing option has been chosen, one can bypass unwanted sections of the listing by setting SW 14 to 1, thus only the line number and the page number are printed. By setting SW 15 to 1, only the page number is printed. As soon as these switches are reset to Ø the printing reverts to complete listing.

6.6 OBJECT PROGRAM LOADING

Set "PARITY" switch to parity or NO parity check.

6.6.1 TELETYPE LOADING

1. Set teletype/reader switch to "TTY".
2. Place binary tape into teletype tape reader.
3. Press "LOAD PROGRAM".

6.6.2 READER LOADING

1. Set reader switch to "RDR".
2. Place binary tape into tape reader.
3. Press "LOAD PROGRAM".

Note: It will automatically load until one of the following conditions occurs:

-It reads a control-B character (Valid end of loading).

-There is a parity error (if parity switch was set).

-There is a Teletype transmission error.

-The Operator pressed "HALT".

AUTOMATIC ELECTRONIC SYSTEMS INC.

The AES-85 CRT Control System

usually use Motorola

5x7 (Coins 7x9 9x13)

A FAMILY OF PROGRAMMABLE CRT TERMINALS

Automatic Electronic Systems Inc.
5455 Pare Street
Montreal 309, P. Quebec, Canada
(514) 735-6581

AES Data Inc.
P.O. Box 143
St. Albans, Vermont 05478
(802) 524-3660

TABLE OF CONTENTS

1. GENERAL

The AES-85 CRT terminals are programmable units organized around the AES-80 Microprocessor as the controller. Various functions of the terminals are performed by executing programs stored in the Read Only Memory of the microprocessor. The microprocessor controller permits extreme flexibility in configuring the AES-85 systems. The microprocessor equipped with the appropriate program permits the AES-85 terminals to appear like many different devices, to perform a variety of different functions, to conform to various communication protocols and to interface to any computer.

The AES-85 terminal design is modular. Standard modules can be configured in a number of different ways to meet specific requirements. A range of interface cards are available which permits the easy interfacing of the terminals to synchronous or asynchronous modems, serial or parallel keyboards, card reader/punches, tape readers/punches, tape casettes, etc.

2. SYSTEM CONFIGURATIONS

The AES-85 family of terminals consists of three standard units:

(i) AES-85/1: a single channel stand-alone terminal including control electronics, keyboard, monitor and power supply.

(ii) AES-85/2: a four channel system suitable when a number of terminals are concentrated in the same location. A central control electronic unit drives a cluster of four keyboard-monitor combinations. The four terminals share the same microprocessor and communication line resulting in considerable savings in cost per channel. The unit is modular, the number of channels can be expanded one to four in increments of one.

(iii) AES-85/3: a four channel system, similar to AES-85/2 but with higher quality character display. The main application is for public displays. (Air terminals etc.)

3. DESCRIPTION OF SYSTEMS

Each system as described in Par. 2 is built up from these basic functional blocks:

(i) Terminal Controller (TC)
(ii) Processor
(iii) Interfaces

The Terminal controller performs the functions of generating and maintaining the display and controlling the cursor. It operates entirely under the control of the Processor. The Processor controls the operation of a unit by communicating with the Terminal Controller (s) and with the interfaces and by executing routines initiated by received data.

The interfaces perform level conversion, character assembly and disassembly and provide control and timing signals for the connected peripherals.

In the following, a description of the basic functional blocks is given.

3.1 TERMINAL CONTROLLER

The Terminal Controller is constructed on one standard AES-PC board, approximately 11" x 12" in size. The TC consists of two functional sub-units as follows:

(i) Video Sub-unit, consisting of the following circuits:

- Refresh Memory
- Character Generator
- Cursor Generator
- Video Amplifier

The refresh memory is built using MOS RAM memories (2K).

The character generator uses commercially available Read Only Memorys.

(ii) Register Sub-unit, consisting of the following circuits:

- Processor Interface
- Data in/out registers
- Cursor x-y registers
- Refresh Memory R/W controls
- Cursor address decoding
- Various controls. (Refresh ON/OFF, Cursor ON/OFF).

The Terminal Controllers used for the AES-85/1 and AES-85/2 are identical; The Terminal Controller used for AES-85/3 is equipped with a higher resolution character generator.

3.2 PROCESSOR

The processor is constructed on one 11" x 12" PC board. All functions of the AES-85 Terminals are controlled and performed by the Processor by executing programs stored in its Read Only Memory. It is an AES-80 Microprocessor equipped with the required capacity of Read Only Memory and Random Access Memory.

Some salient features of the AES-80 microprocessor are listed here. A detailed description of the microprocessor can be found in the document titled AES-80 Microprocessor Reference Manual.

The AES-80 microprocessor is a byte oriented general purpose processor designed primarily for use in dedicated applications. Its main features can be summarized as follows:

-Instruction memory (ROM) size: up to 4K 12-bit words (expandable to 64K)

-Data Memory (RAM) size: up to 4K 8-bit words.

-240 nanosecond instruction time.

-6 registers

-All modes of 8-bit ALU under software control.

-16 level automatic push down stack for routine linkage.

-64 basic one word instructions plus 24 arithmetic and logic instructions.

Some features of the AES-80 microprocessor renders it particularly suitable as a processor for a programmable terminal.

-Intermixing of different types of memories. The organization of the microprocessor allows the intermixing of memories of different speeds. This feature enables the use of bipolar, MOS or other type of memories as the speed requirements of specific applications dictates.

-Bus structure. The Terminal controllers and interfaces, being connected to the parallel bus of the microprocessor, can be handled as extensions of the Data Memory.

The microprocessor is supported by extensive software packages. Specifically, a standard Assembler is available which allows programmers to write their microprocessor programs in a symbolic language rather than in machine language, the translation being performed by the Assembler.

A detailed description of the assembler can be found in the document titled AES-80 Microprocessor Assembler Reference Manual.

For developing and checking programs and for trouble shooting a Program Development Console is available.

Each AES-85 terminal has provisions for the connection of a Program Development Console, a description of which can be found in the document titled AES-80 Microprocessor Reference Manual.

3.3 INTERFACES

Any AES-85 terminal can be equipped with a variety of interfaces, to a variety of peripherals and communications lines. Presently, three types of interfaces are available. Other types are developed as the requirements arise:

3.3.1 Synchronous Communications Interface

This module interfaces the processor to synchronous modems. It performs the following functions:

- Acquires character sync on reception of two consecutive sync characters.
- Serializes transmitted characters.
- Assembles received characters.
- Transfers characters to and from the microprocessor.
- Provides data and control signal interface for modems according to EIA RS 232-C.

This interface can operate at speeds up to 9600 bits/second.

3.3.2 Asynchronous Communications Interface

This module interfaces the processor to asynchronous modems. It performs the following functions:

- Generates start and stop bits for transmitted characters.
- Strips the start and stop bits from received character.
- Serializes transmitted characters.
- Assembles received characters.
- Transfers characters to and from the microprocessor.
- Provides data and control signal interface for modems according to EIA RS 232-C.

This interface can operate at speeds up to 2400 bits/second.

3.3.3 Keyboard Interface

This module interfaces the processor to two keyboards. Each keyboard interface is itself equipped with two input/outputs.

(i) parallel interface for a keyboard within 50 feet.

(ii) serial interface for a keyboard up to a 1000 feet.

3.3.4 Other Interfaces

Other interfaces for such as: cassettes, ticket printers, and readers are available to meet specific requirements.

4. PERFORMANCE SPECIFICATION

4.1 VIDEO

	AES 85/1	AES 85/2	AES 85/3
Screen size	12"	Note 1	
Character/Line	64 or 80 NOTE 2		48 or 64 NOTE 2
LINES	up to 24 NOTE 2		16
Character Format	5x7 (UPPER CASE CHARACTERS) 5x9 (LOWER CASE CHARACTERS)		7x9 (UPPER CASE) 7x11 (LOWER CASE)
Refresh Rate	60 times/second		
Raster	262 lines non-interlacing		
Video output:	Composite (commercial) Output level: 1V pp min. Output impedance: 75 ohms 5%		
Character Repertoire	96 ASCII		
Cursor	Underline (NOTE 3)		
Number of channels	One	Up to four:	

Note 1: Monitors specified by customer.

Note 2: The various formats are implemented by strapping, by program or by switches.

A change in format does not necessitate change in the program.

Note 3: Other types of cursor can be implemented as options.

4.2 REFRESH MEMORY OPTIONS

Fully equipped, the refresh memory has 8-bits per character. Six of the eight bits are used to specify 64 ASCII characters. The remaining two bits, under program control, can be used for a number of different purposes.

1.) Extending the character repertoire to 96
2.) Memory Protection (Split screen)
3.) Half tone control
4.) Flash control on a character-by-character basis.

4.3 FUNCTIONS

Due to the programmable nature of the terminals, they can execute any functions as specified by the customer. The functions of the terminals are therefore not specified. A list of commonly used functions for which the terminals can be programmed are:

Cursor Controls:	SP
	BSP
	CR
	LF↑
	LF↓
	HOME
Erase:	Line erase
	Field erase
	Frame erase
	Foreground (unprotected) erase.

Editing: Tab
Character insert/delete
Line insert/delete

Transmit: Line
Field
Frame

4.4 COMMUNICATIONS

The terminals can be programmed to comply with any specified communications discipline, using either synchronous or asynchronous transmission. Specifically, the terminals can be programmed to emulate completely the IBM 2260/3270 terminals.

The processing of communications including assembly of messages, decoding of controls and addresses, error checking, retransmission, response to polling etc., is completely under program control.

4.5 ENVIRONMENT

Temperature: $10^{o}C$ - $45^{o}C$ operating

Humidity: 5% - 95% R.H., no condensation.

4.6 POWER

105 - 125 VAC, 60 Hz

4.7 MECHANICAL

AES-85/1: Constructed as a self-contained table top unit, including electronics, monitors, keyboard and power supply.

AES-85/2: Constructed as a 19" rack mountable unit, using the standard AES card cage. The unit includes control electronics and power supply only.

AUTOMATIC ELECTRONIC SYSTEMS INC.

The AES-81/10 Remote Terminal Unit

RTU APPLICATION BULLETIN

Automatic Electronic Systems Inc.
5455 Pare Street
Montreal 309, P. Quebec, Canada
(514) 735-6581

AES Data Inc.
P.O. Box 143
St. Albans, Vermont 05478
(802) 524-3660

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

1.2 PURPOSE

1.3 INTENT

1.4 APPLICABLE DOCUMENTS

1.0 INTRODUCTION

1.0.1 GENERAL

Automatic Electronic Systems Inc. is pleased to submit details of our remote terminal unit for supervisory control and data acquisition applications.

1.0.2 AES CREDENTIALS

Automatic Electronic Systems Inc. specializes in supervisory control and data acquisition equipment. Since inception, AES has developed a comprehensive systems capability in this field. The success that has been achieved is due, in great measure, to furnishing well-proven "of the-shelf" components interconnected in the system configuration. In order to maintain this success, AES has an extensive research and development program to develop new products and to update existing product lines.

AES maintains a full support staff of Sales Engineers and Field Support Engineers for customer liaison. The Field Support Engineers, trained on each system during the production and final test phases, provide on-site assistance during installation and commissioning and serve as a continuous back-up to the users' maintenance staff.

1.0.3 EXPERIENCE

AES' experience includes design, manufacture, test, installation, documentation, computer software, and field support on a wide variety of related equipments, some of which are described in the following paragraphs.

1.0.3.1 Data Acquisition Equipment

AES has provide several Data Acquisition Systems ranging from simple "one-to-one" configurations to multiple remote-single master configurations with conventional displays and/or computer controlled data-logging. A typical system, comprising 32 remote stations and a single master with computer, has been installed with the Manitoba Telephone System for several years and is in current usage.

1.0.3.2 Supervisory Control Equipment

Numerous Supervisory Control Systems are in use with Major Utilities for substation control. A typical configuration, such as the Berri Substation for Hydro Quebec, involves the control of approximately 60 breakers (with auto-reclose)

and the reporting of approximately 200 status points complete with telemetering. Currently in production at the AES facility, is a Supervisory Control and Data Acquisition System for the control of a generating station. This system utilizes CRT display systems and microprocessors of AES manufacture.

Also in production are several supervisory control and data acquisition systems, utilizing microprocessors and complex security techniques, for usage by Telesat Canada in television network control applications via synchronous satellites.

1.0.3.3. Protective Relaying Equipment

AES manufactures a line of sophisticated tone equipment for protective relaying applications. The "S-300 Tele-Protection System", is a current model and several hundred terminals are in use throughout North America at large power utilities. This equipment, incorporating several new techniques to provide security, is recognized as being the best available for this demanding application.

1.0.3.4. Telex Multiplex Equipment

AES has produced Telex Multiplex Equipment for the Canadian Overseas Telecommunications Corporation. This equipment multiplexes approximately 900 telex channels and provides a high speed data stream into a Univac computer system for the control of overseas telex data originating in Canada for transmission to the United Kingdom. In addition, standard systems have been developed for use by the common carriers in their various applications.

1.0.4. PROPOSED EQUIPMENT

The proposed equipment provides a means for data acquisition and control at all levels ranging from simple applications requiring only alarm reporting and annunciation to those applications requiring full control capability with the acquisition of alarms, status and analog values. The architecture is modular, to permit a building block approach for specialized system applications, and to allow for expansion in the field at a later date if necessary.

1.1 SCOPE

1.1.1 This specification describes the performance characteristics of the AES 81/10 programmable REMOTE TERMINAL UNIT (RTU).

1.2 PURPOSE

1.2.1 The purpose of this specification is to describe the operating and technical characteristics of a RTU which is functionally and communications compatible with commonly used Remote Terminal Control systems.

1.3 INTENT

1.3.1 The intent of this specification is to provide sufficient information to permit an effective technical evaluation of the RTU.

1.4 APPLICABLE DOCUMENTS

The following documents are listed for reference purposes only:

1. AES 80 MICROPROCESSOR REFERENCE MANUAL
2. AES 80 MICROPROCESSOR ASSEMBLER REFERENCE MANUAL

2.0 GENERAL

2.1 PURPOSE OF EQUIPMENT

2.2 GENERAL DESCRIPTION

2.0 GENERAL

2.1 PURPOSE OF EQUIPMENT

2.1.1 In general, REMOTE TERMINAL units provide the primary interface with the instrumentation and control signals of an electric power network. As such, the RTU must have the capability to perform both systems and local functions.

2.1.2 System functions include:

a) Communicating with a MASTER site via a suitable communications link,

b) Operating within the constraint of master station computer controlled operating procedures,

c) Performing message security operations to minimize the effects of transmission errors.

2.1.3 LOCAL functions include the following:

a) Provide timed contact closure outputs for operation of two-position devices such as circuit breakers and disconnect switches,

b) Provide timed contact closure outputs for incremental operation of multi-position devices such as tap changers, gates, governor motors and motor operated rheostats,

c) Monitor and store status of contact inputs for reporting of alarms and device positions. This also includes momentary change detection.

d) Provide digital contact outputs, with or without digital-to-analog conversion, for set point control,

e) Count and store impulses representing quantities such as KWH or positive displacement of liquids,

f) Provide multiplexing, analog-to-digital conversion and digital storage for quantities such as watts, vars, voltage and current,

g) Transmit status and data in digital form to the master station upon request.

2.2 GENEPAL DESCRIPTION

A block diagram of a typical data acquisition and control network integrating an AES 81 RTU System is shown on Fig. 2-1.

The AES 81 system operates in what is commonly known as a continuous scanning mode with the master station in command at all times. System direction and programming are provided by the master station, and the RTU responds and replies only when specifically commanded or interrogated by the master station. Transmissions between the master and remote stations are carried on repetitively under control of a master station program determined by the system requirements. In general, this continuous scanning mode of operation will be interrupted only when it is necessary for the master station operator to perform a control function.

This mode of operation provides a high degree of flexibility in that the master station can request and receive, at different intervals, data of varying degrees of importance. For example, some data must be updated as often as every two seconds, while other data will be required only every fifteen munutes or longer. Programs in the master station central processor unit, coupled with the extremely flexible method of addressing data at the RTU can automatically provide the different data retrieval cycles required. Another advantage of the continuous scanning mode of operation is that frequent checks are made on the satisfactory functioning of the channels and remote terminal units. Any failure of the master station to receive a reply to its request from the remote station can instantly be brought to the attention of the operator.

When not communicating with the master station or performing a control function commanded by the master station, the RTU continuously performs an internal routine for the purpose of gathering and storing status and data. The organization and operating mode of the remote terminal unit are such that current status, unreported indications of status change, and data in digital form are all stored in memory and ready for transmission to the master.

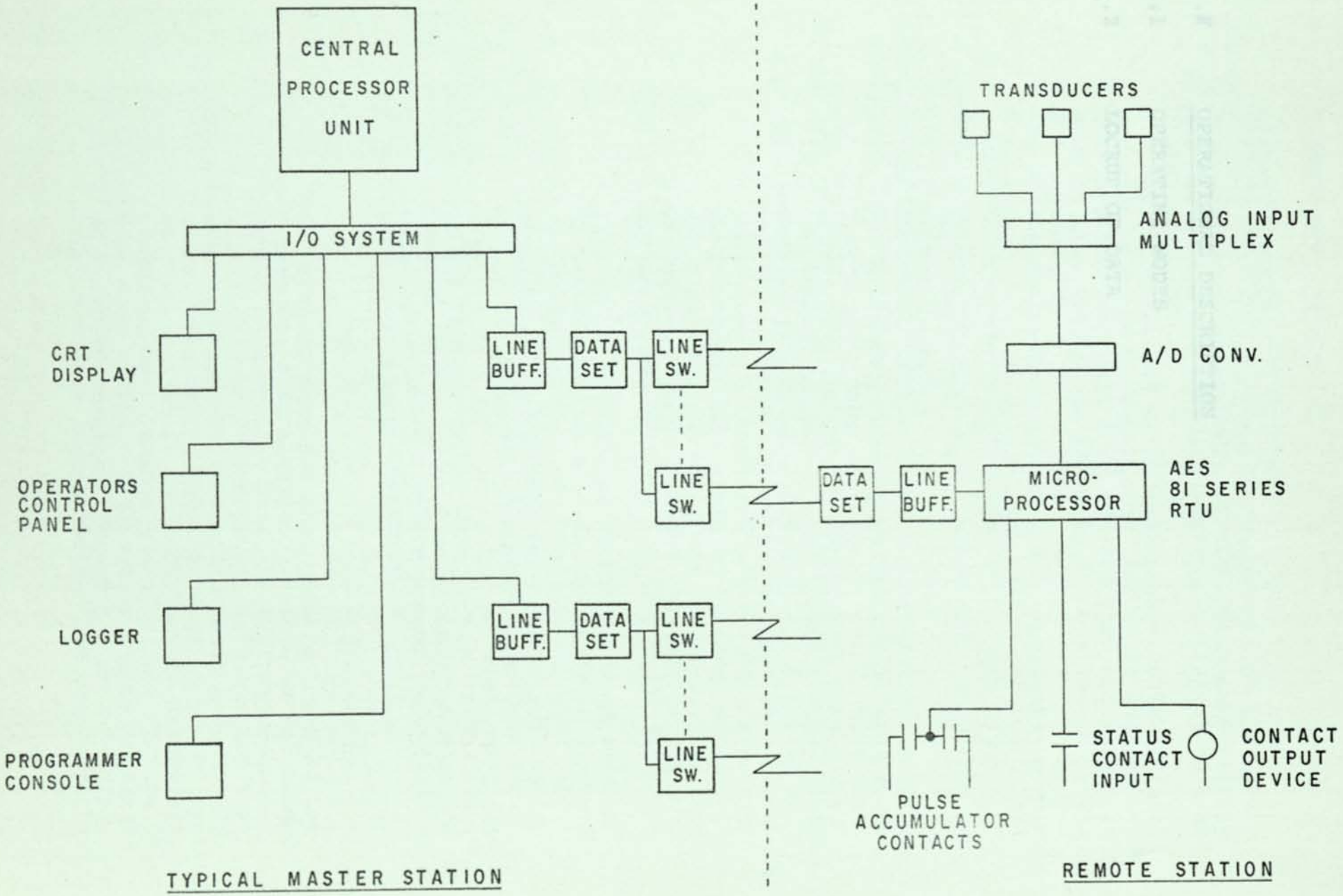


FIGURE 2-1 BLOCK DIAGRAM

3.Ø OPERATIONAL DESCRIPTION

3.1 OPERATING MODES

3.2 LOCKUP OF DATA

3.Ø OPERATIONAL DESCRIPTION

3.1 OPERATING MODES

Three basic operating modes of the system permitted by the RTU design are the data scan mode, control mode and status scan mode.
The design of the AES 81 RTU has emphasized efficient and high speed data acquisition, and it is expected that most applications will use the data scan mode. The control mode is also very basic and will be required on practically all applications.
The status scan mode will be used on applications where frequent updating of data at the master station is not required. These modes of operation are explained in the following.

3.1.1 Data Scan Mode

In this mode of operation the master station can continuously keep its memory updated with the latest data and status information from all of the remote stations of the system. When the master station is not being asked by the operator to perform a control operation the master station can continuously request and receive data from the remote terminals. Thus, up-to-date data and status can always be stored in the master station central processor unit, from which they may be rapidly obtained when required for display or logging.

Data is addressable as described under MESSAGE STRUCTURE AND CODING. In this mode of operation the master station can continuously send out requests with addresses for data in accordance with a preprogrammed routine. Each data reply consists of 1 to 8 data messages containing from 24 to 192 bits of useful information. This data retrieval routine can be run on a continuous basis unless it is required to perform remote control operations or to bring back status indications.

In order to make the most efficient use of the system for data retrieval, status is not transmitted on a routine basis in reply to periodic interrogations. However, it is necessary that status changes be reported without unnecessary delay. Therefore, the organization of the system is such that it is not necessary to transmit status unless a change has occurred. In the data scan mode if there has been no change in status at a remote terminal receiving a data request, the requested data is transmitted and the master station

proceeds to request data from the next remote terminal unit in a similar manner. Now, assume a change of status has occurred in the remote terminal unit receiving a data request. A common change of status flag is set at the remote terminal unit. The remote terminal unit sends the requested data in the usual manner, but the function section of the first data message contains a change of status flag to inform the master station that there is a status change to be reported at a sub-remote of this station. The remote terminal unit will complete the data reply as originally requested. Then the master station can transmit a status request, which is very much like a data request, to the same remote terminal unit to obtain the necessary status changes and present status information.

After the request status and changes have been obtained the master station can proceed with the data scan as before.

Remote stations may each include two or more blocks of data readings, requiring two or more data requests per station and proportionately more time for transmission of data. When this is the case, it may be desirable to use an alternate scanning arrangement in order to avoid too much delay between visits to the individual stations for the purpose of learning of possible status changes. With this alternate scanning arrangement the first block of data would be requested from each remote station on the first scan cycle, then the second block of data from each station on the next scan cycle, and so on until all data has been obtained. Each data reply gives the remote station an opportunity to notify the master station that a status change has occurred.

3.1.2 Control Mode

When a control operation is requested, the normal data scan mode is momentarily interrupted. The master must first select the controlled device to be operated and then transmit the appropriate control (close, trip) message. As a security measure to minimize the time that a point remains selected at the remote station, there are no transmissions between the master and remote stations for control purposes until after the master control has been initiated. After the master control has been activated, the data scan mode can be interrupted permitting the following exchange of messages to occur.

a) A selection message is transmitted from the master to the remote.

b) A checkback message, similar to the selection message, is transmitted from the remote to the master.

c) An activate message, which must agree with the selection message, is transmitted from the master to the remote to cause the operation to be performed.

d) An activate acknowledge message is transmitted from the remote to the master.

Two important security measures are inherent in this message exchange for performing a control operation. The checkback message which is returned by the remote terminal in reply to the selection message transmitted by the master station can be compared for agreement with the selection message which was sent out. Upon verification of the checkback message, the master station can follow with the activate message. The second safeguard is that the activate message must agree with the selection message in order for the remote station to proceed with operation of the device. Upon receipt of the correct activate message the interposing relay is energized for a preset time and is then released. The acknowledge message which is returned to the master station verifies that the activate message was correctly received and the appropriate control action was initiated.

At this point the remote terminal is cleared of the point selection and the control operation. After the controlled device has operated, its new status will be reported to the master station when that remote terminal is again polled during the normal data scan mode. The system normally returns to the data (or status) scan mode immediately after the master station has received the activate acknowledge message from the remote terminal.

3.1.3 Status Scan Mode

The status scan mode may be used when frequent updating of data at the master station is not required. In this mode of operation the master station will continuously scan the remote terminals asking for status (status is addressable in the same manner as data). If there has been no change of status the remote terminal unit will reply with a single message, indicating in the function section of the message that there are no status changes to report. The master station then proceeds to interrogate the next remote terminal unit. If a change of status flag is reported in the initial reply, the master station will then request a report of status and status changes in the same manner as used in the data scan mode.

3.2 LOCKUP OF DATA (Optional)

It is often necessary, for the purpose of making system load and stability studies, to make data readings at all remote stations within a very short span of time. A unique feature of the AES 81 system is its ability to take readings which are very nearly simultaneous at all remote stations. Each remote station, while "at rest", is busy collecting and storing data in digital form at the rate of 25 A/D conversions per second. If each remote station is assumed, for example, to have 12 data readings, at any given time the oldest data in memory will be only one-half second old-or an average of one-fourth second for all readings at the station.

The remote station, upon receipt of a lockup signal from the master station, freezes all data in storage and prevents further A/D conversions from being entered. If lockup signals are sent from the master station to all remote stations at the same time assuming transmission times to all stations to be nearly equal all stations will have stored data whose measurements were all made within the same one-half second interval. After lockup, the master station then requests and obtains the stored

data in the usual manner. Following acquisition of the data, the master station then sends a lockup reset signal to allow resumption of the normal updating of data in storage.

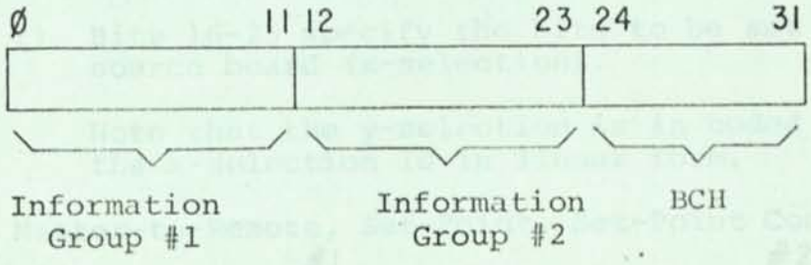
4.Ø MESSAGE STRUCTURE AND CODING

4.Ø MESSAGE STRUCTURE AND CODING

4.1 Basic Message Structure

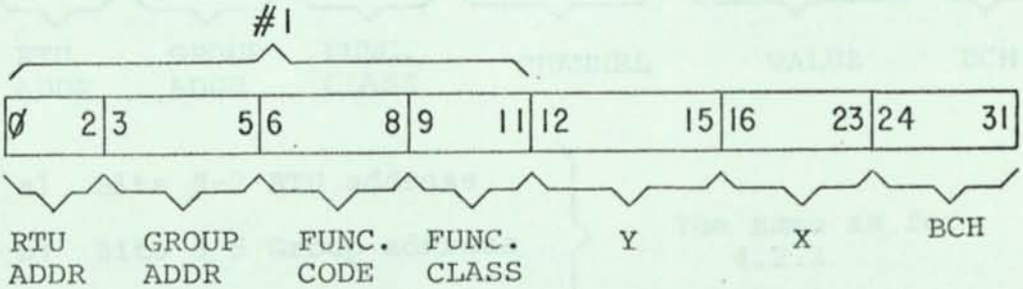
A fixed message length of 32 bits is used. Each message consists of two 12-bit information groups (Ø-11 and 12-23) and eight bits (24-31) for a cyclic parity code check (BCH). All messages transmitted by any station are of this length and structure. In general, octal coding is used for addressing and binary used for data.

Numbers shown above in parenthesis are the numbered bits as shown below:



4.2 Message Formats

4.2.1 Master-to-Remote for selection of a control device.



a) Bits Ø-2 specify the RTU address. A maximum of 12 stations can be addressed (4-15). Leaving 4 common addresses (Ø-3) which should be accepted by all connected stations.

b) Bits 3-5 specify the address of the SUB-RTU as specified by the function class.

c) Bits 6-8 specify the function class. The following classes have been assigned:

Ø: Control selection
1: Momentary activate
2: Latching activate
3: Housekeeping
4: RTU-read
5: RTU-write
6: Set-point control
7: Special

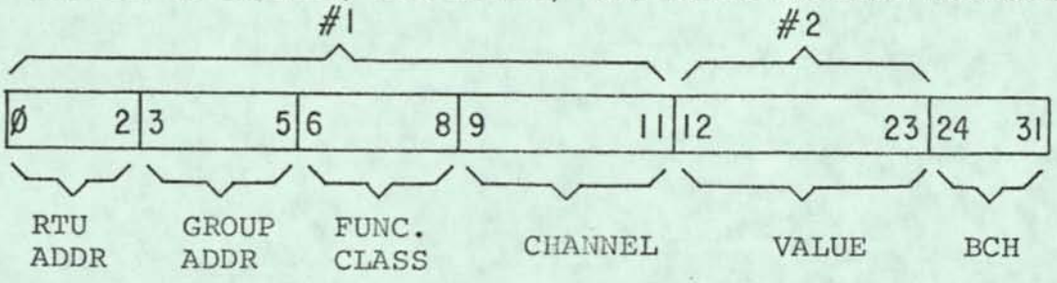
d) Bits 9-11 specify the function code. The interpretation of this code is related to the current function class (see code matrix).

e) Bits 12-15 specify the address of the bit (1/16) to be set in the output sink board (y-selection).

f) Bits 16-23 specify the bits to be set in the output source board (x-selection).

Note that the y-selection is in coded form, and that the x-selection is in linear form.

4.2.2 Master-to-Remote, Set-Point, Set-Point Control Selection



a) Bits Ø-2 RTU address

b) Bits 3-5 Group address

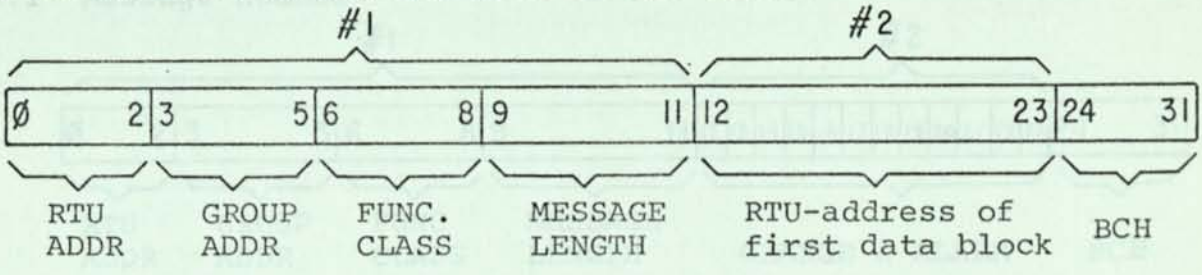
c) Bits 6-8 Function class

} The same as for 4.2.1

d) Bits 9-11 specify the set-point channel address. Up to 4 channels can be addressed to each sub-RTU.

e) Bits 12-23 specify the analog value in sign plus magnitude form. The sign is located in bit-12, and MSB of the magnitude in bit-13.

4.2.3 Master-to-Remote, RTU-read or RTU-write.



a) Bits Ø-2 RTU address

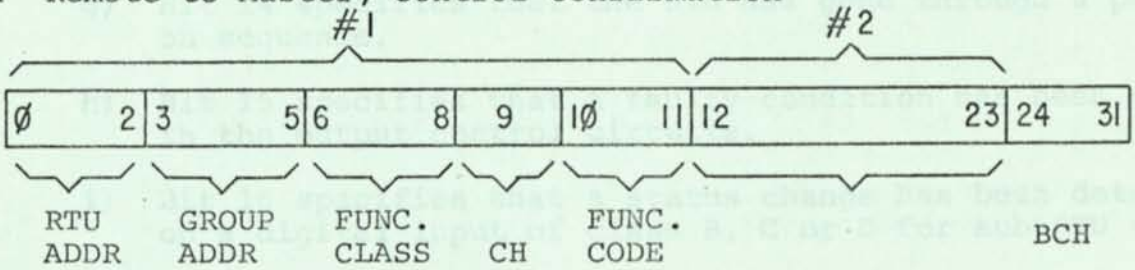
b) Bits 3-5 Group address

c) Bits 6-8 Function Class

} The same as for 4.2.1

d) Bits 9-11 specify the number of pairs of data blocks the received or transmitted message should have. Up to 2 x 8 12-bit data blocks can be requested or transmitted from the master. Note that addresses Ø-3777 are prevented from writing.

4.2.4 Remote-to-Master, Control confirmation



a) Bits Ø-2 As was received

b) Bits 3-5 As was received

c) Bits 6-8 As was received

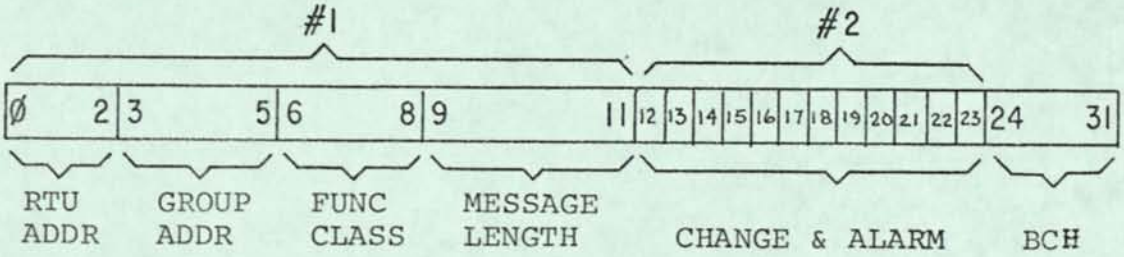
d) Bit 9 specifies a change or alarm condition in the RTU. A "1" indicates that the mentioned condition exists and a Ø indicates a normal condition. The Master can get a better picture of the abnormal condition by examining the header of a Remote-to-Master Interrogation message.

e) Bit 1Ø-11 function code, as received in control selection.

f) 12-23 Selection Addresses, as received in control selection.

4.2.5 Remote-to-Master reply to RTU-read.

4.2.5.1 Message Header



a) Bits Ø-2 RTU-address
b) Bits 3-5 Group address
c) Bits 6-8 Function address
d) Bits 9-11 Message length

} As received in RTU Write Command

e) Bit 12 Specifies general status change condition (COS)

f) Bit 13 specifies that a control action is in process.

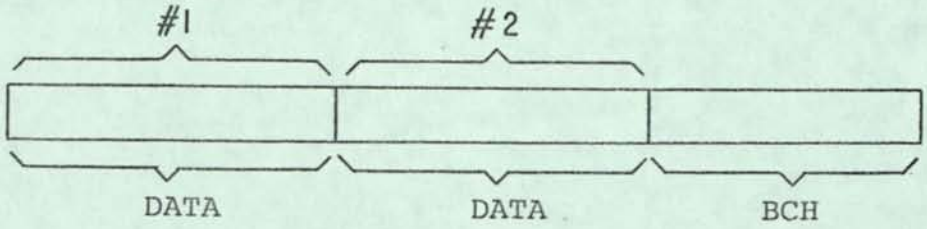
g) Bit 14 specifies that the RTU has gone through a power on sequence.

h) Bit 15 specifies that a faulty condition has been detected in the output control circuits.

i) Bit 16 specifies that a status change has been detected on a digital input of class B, C or D for sub-RTU #Ø.

j) Bit 17 to 23 the same as for i) but for sub-RTU 1-7.

4.2.5.2 Data Message



Note that the address of the first data block in the first data message is specified by the starting address (4.2.3).

4.3 Status Indications

Status and change of status indications are handled in a similar manner to data, and are addressable in blocks of 12 indications. Up to fifteen 12-indication blocks may be obtained in reply to one request. The first reply message will contain the address and the first group of 12 indica-

tions. Subsequent reply messages in a multiple reply will each contain two 12-indication blocks.

In order to report rapid status changes such as trip-reclose or close-trip, each status memory has associated with it a momentary change memory. The momentary change indications are reported in the same manner as the current status. Therefore, for each block of 12 status indications there is a corresponding block of 12 change indications transmitted. The total of fifteen blocks in a reply will include eight blocks of status and seven blocks of change of status.

4.4 BCH (Cyclic Code for Error Detection)

BCH is an acronym formed from the names of Messrs: Bose, Ray-Chaudhuri, and Hocquenghem because of their contribution to error detecting cyclic codes. A cyclic code is so described because of its property which allows the bits of the code to be cyclically shifted in some fashion without losing the identity or effectiveness of the code.

Transmission error detection employs a BCH cyclic code check utilizing eight parity check bits in each message. The use of the BCH coding permits detection of all (1ØØ%) single and double bit errors in a message block, all (1ØØ%) burst error patterns of length 8 bits or less, 94% detection of burst error patterns of length 9 bits, and 97% of all burst error patterns of length 1Ø bits or more. A burst is defined as the number of binary bits between two error bits, including the error bits. All or any number of bits in the burst may be in error.

The BCH code is generated by the primitive polynomial $P(X) = X^8 + X^7 + X^6 + X^Ø$ checking on 4Ø bits. The Framing bits of the message structure are excluded. The method used to generate the 31 bit coded message polynomial, F(X), is as follows:

1) The information polynomial, G(X), consisting of 32 bits is multiplied by X^8 to form $X^8G(X)$.

2) $X^8G(X)$ is divided by the generator polynomial $P(X) = X^8 + X^7 + X^6 + X^Ø$ to give a quotient Q(X) and a remainder R(X).

3) The 4Ø bit coded message polynomial is the sum of $X^8G(X)$ and R(X).

$$F(X) = X^8.G(X)+R(X)$$

In checking, a received message is divided by the generator polynomial P(X). If the resulting remainder is zero, either no error or an undetectable error has occurred. If the

remainder is not zero, an error has been detected.

4.5 Message Exchanges

The following is a descriptive listing of the symbols used in the data blocks that follows:

RTU = RTU Address

G = Group (SUB-RTU) Address

FG = Function Class

FC = Function Code

L = Message Length

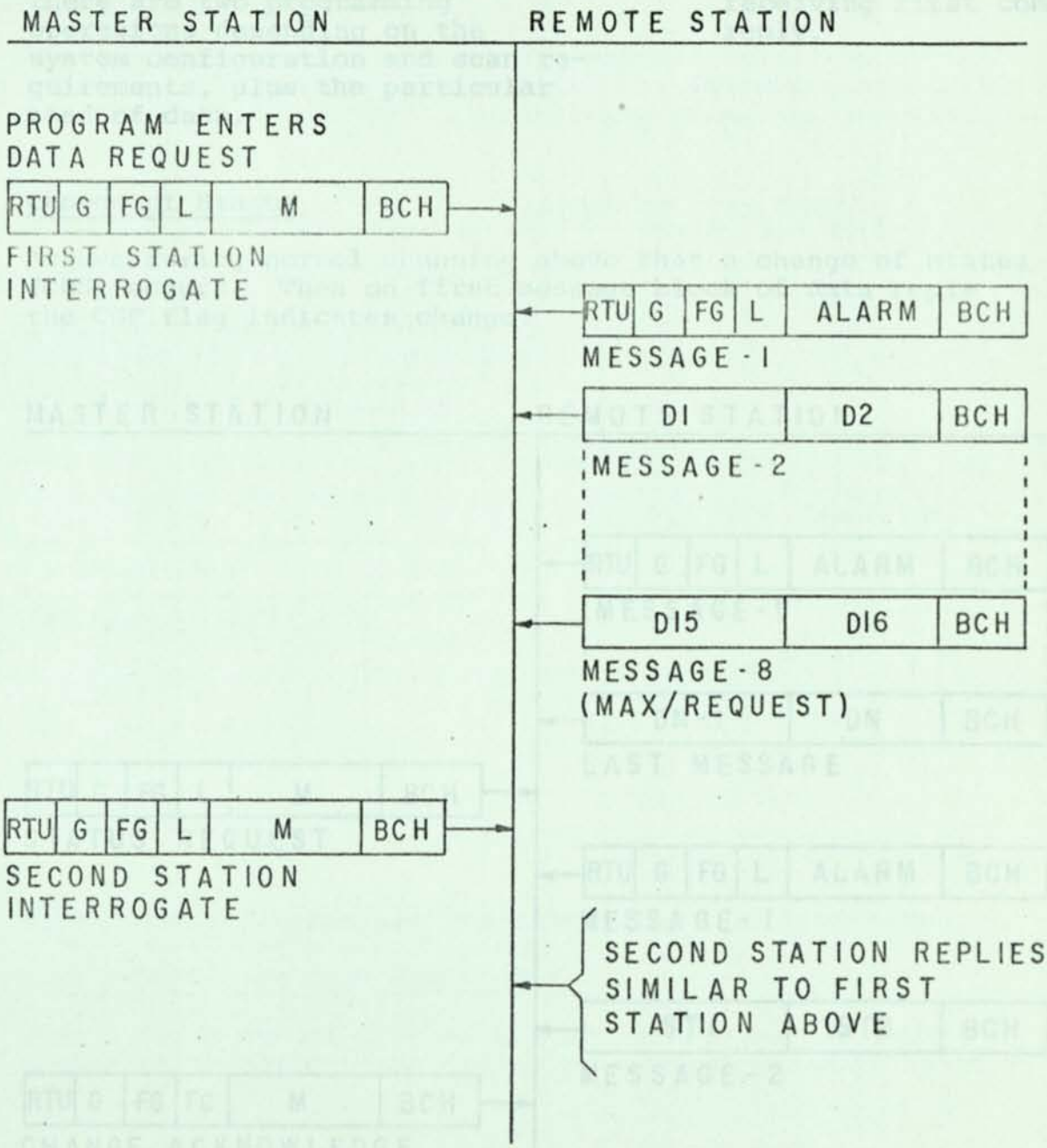
M = Address of First Reading

DX = 12-Bit Data Word

X = X-Selection

Y = Y-Selection

4.5.1 Normal Data Scanning



Sequence continues until last station is scanned or COS flag is reported. For continuous data up-dating, the cycle begins again.

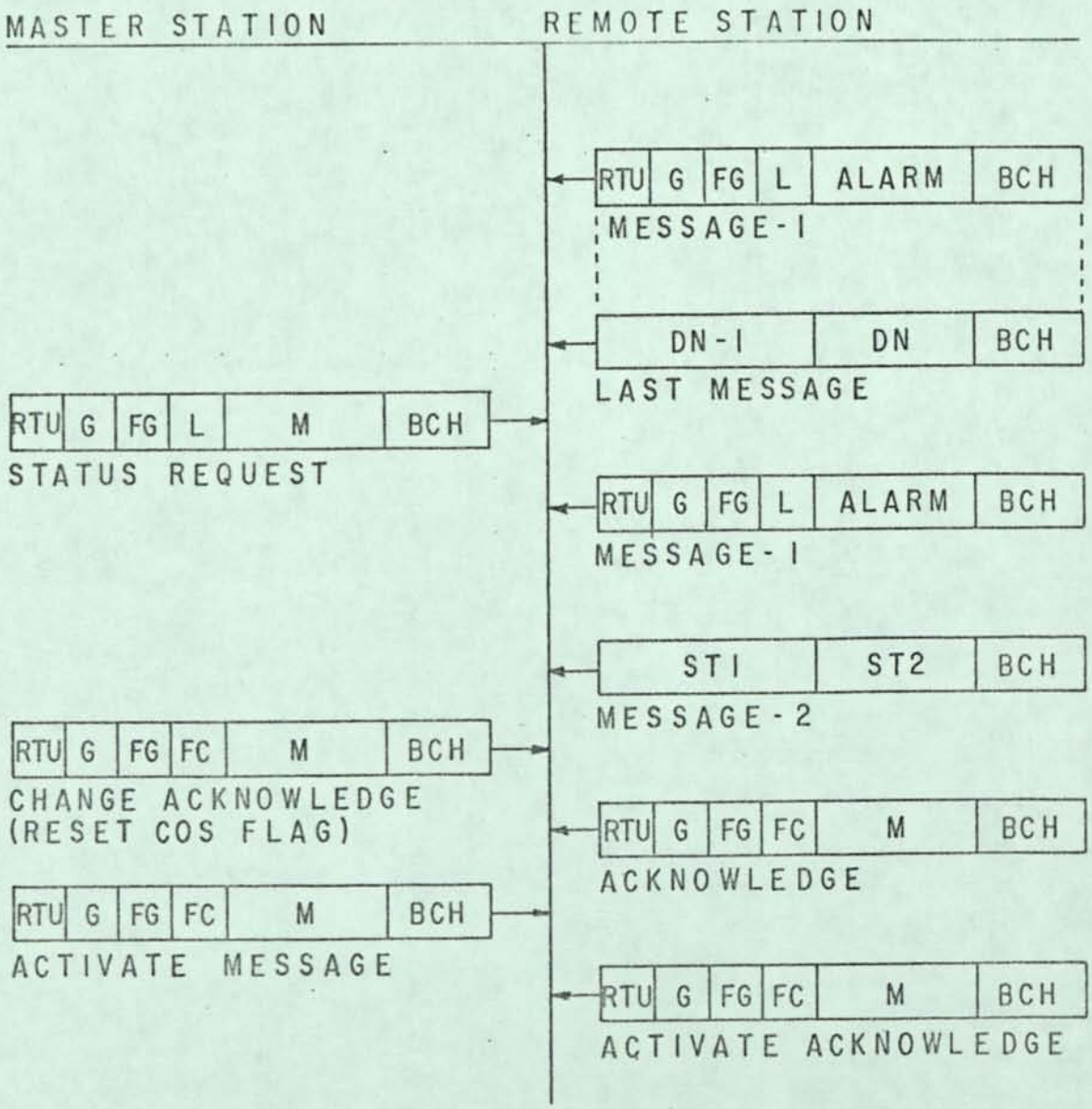
1) Alternate scanning whereby the first 16 data words are requested on the first scan and the second set requested on the next system scan.

Up to 8 Data blocks (16 data words) are sent with each request. When more are required from a station, there are two programming operations depending on the system configuration and scan requirements, plus the particular need of data.

2) Consecutive scanning whereby the second request is made to same station after receiving first complet reply.

4.5.2 Change of Status

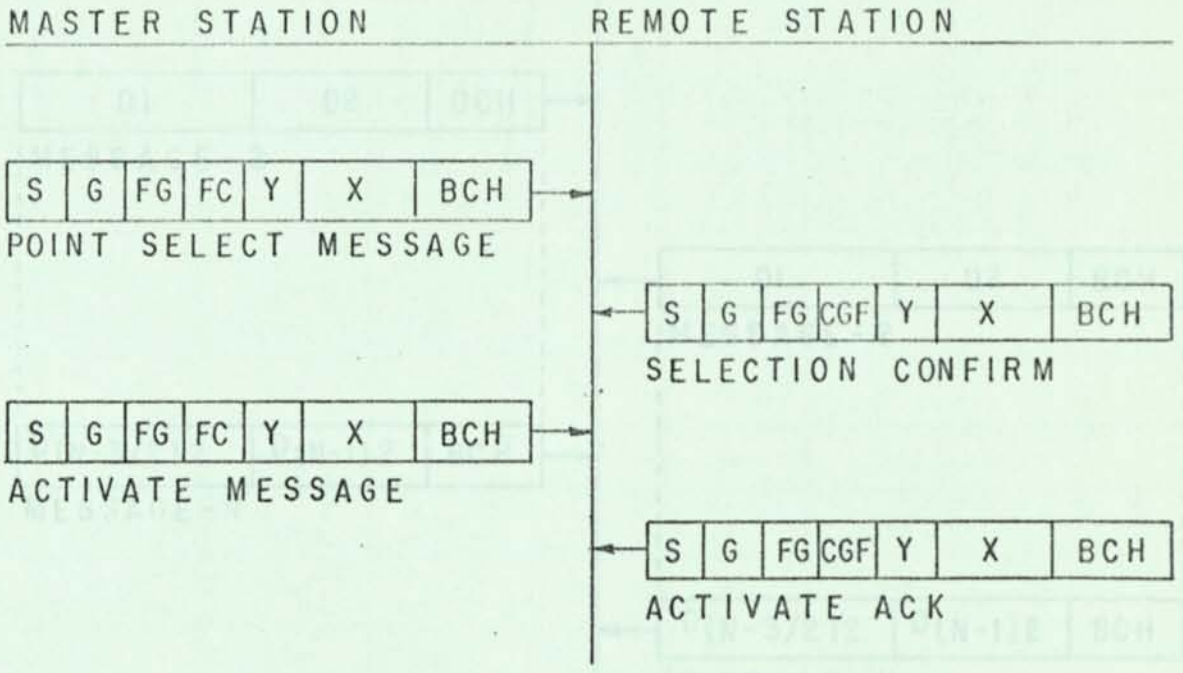
Assume during normal scanning above that a change of status (COS) occurs. Then on first message block of data reply the CGF flag indicates change.



Scan returns to position of data scan where interrupt occurred.

4.5.3 Control Operation

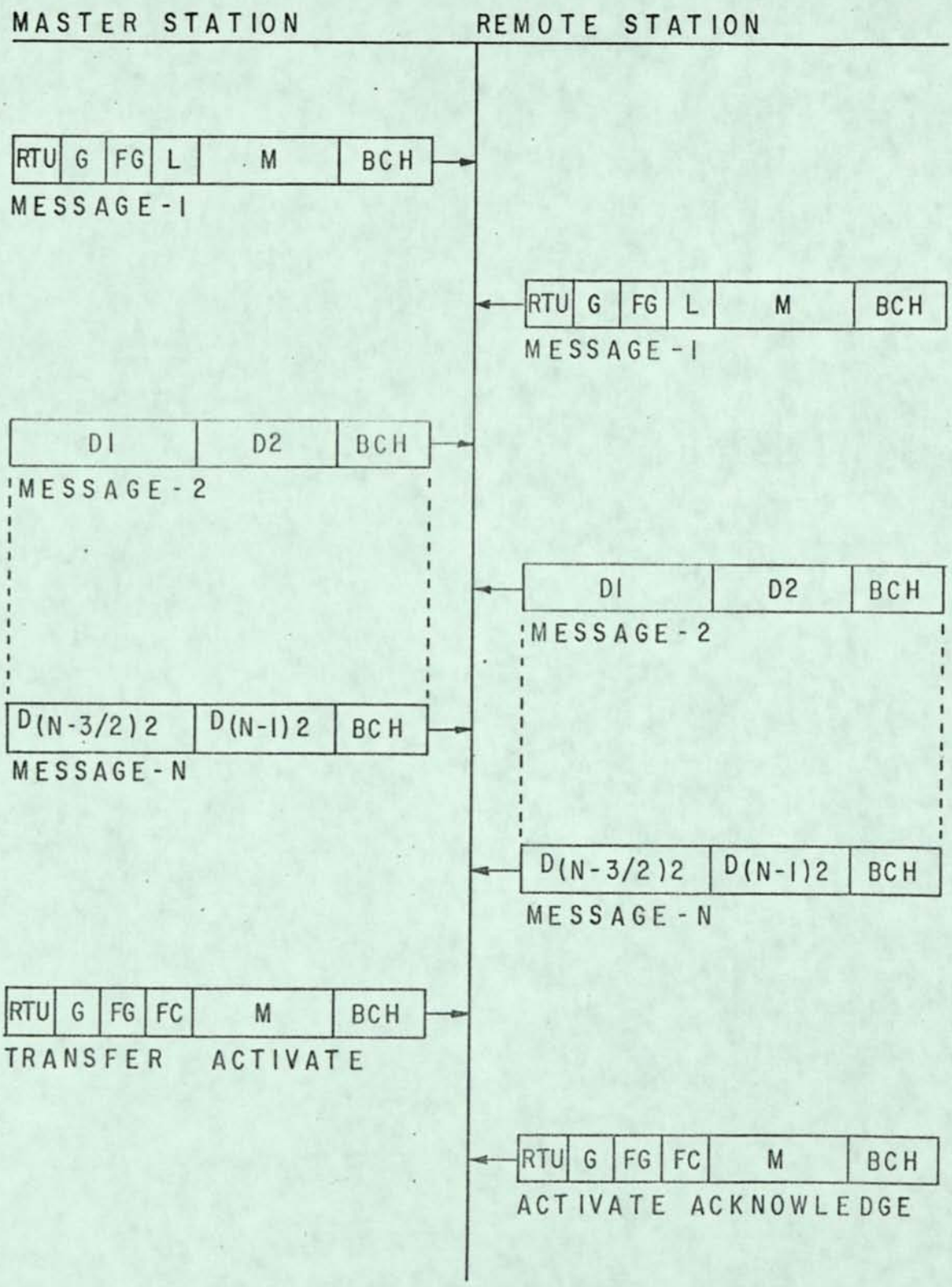
The operator selects a point and control operation on the console. The sequence begins with the following substitute for the next programmed interrogation of associated station.



Data request is then sent to same station as programmed.

4.5.4 Master-To-Remote Data Transferral

When a large amount of data has to be transferred from the master to the remote the following message exchange takes place:


MASTER STATION
REMOTE STATION
RTU G FG L M BCH
MESSAGE-I
RTU G FG L M BCH
MESSAGE-I
D1 D2 BCH
MESSAGE-2
D1 D2 BCH
MESSAGE-2
D(N-3/2)2 D(N-1)2 BCH
MESSAGE-N
D(N-3/2)2 D(N-1)2 BCH
MESSAGE-N
RTU G FG FC M BCH
TRANSFER ACTIVATE
RTU G FG FC M BCH
ACTIVATE ACKNOWLEDGE

Function Group/Code Matrix Bits 6-11

Function Code 9-11	Function Class 6-8							
	ø	1	2	3	4	5	6	7
ø	-	General	General	-	1	1	ø	Automatic Update
1	Open	Open	Open	Status Req.	2	2	1	A/D Lockup
2	Close	Close	Close	Reset of Lockup	3	3	2	PA Lockup
3	Open & Close	Open & Close	Open & Close	Reset of COS	4	4	3	-
4					5	5		-
5					6	6		-
6					7	7		-
7					8	8		-
	Contr. SEL	Mem ACT	Latch ACT.	House-keeping	RTU READ	RTU WRITE	Set-point SELECT	Special
					Message Length		Channel	

4.6 Code assignments for the message formats

Bit No.	Function	Value
Ø-2	RTU-address	
	All stations-lockup	Ø
	Station-1	1
	-2	2
	-3	3
	-4	4
	-5	5
	-6	6
	-7	7
3-5	Group address	
	Sub-RTU-1	Ø
	-2	1
	-3	2
	-4	3
	-5	4
	-6	5
	-7	6
	-8	7

4.7 General Information

4.7.1 If an expected message is not received or if parity check fails, the remote station will take no action.

4.7.2 If any message other than a control execute message is received by the remote station, a selected control point will be reset.

4.7.3 The lockup command is the only message which causes action at all remote stations. Two levels of lockup are defined;

a) Lockup analog points only.

b) Lockup Pulse Counter points only.

4.7.4 Analog data values are twelve bits. The first bit, is sign (1 = +, Ø = -). The remaining eleven bits are the absolute value of the analog reading in binary form with most significant bit following sign bit. The return for the second half of message if the point does not exist will be all "Ø"s. The code for an overrange input

will be all "1"s. Counter values are twelve bits binary.

Upon receipt of an Analog Lockup command the current value of each analog data point will be stored in a digital memory. All data messages will contain that value until the reset of Lockup command is received.

Upon receipt of a Pulse Counter Lockup command the current value of each pulse counter is stored in a digital memory. All data messages will contain that value until the reset of Lockup command is received.

4.8 Error Detection

There is evidence from studies made that the type of error which occurs most frequently in digital transmission over telephone lines is the burst error. Accordingly, for detecting transmission errors the AES-81 system uses a cyclic code which is particularly effective in detecting errors of short bursts. The particular class of codes has been generalized by Bose and Ray-Chaudhuri. In the AES-81 message format there are 24 information bits and 8 check bits which are involved in transmission error detection.

The effectiveness of the code varies with the nature of the errors and is a maximum for errors occurring in bursts up to the number of check bits used, which in this case is 8. A burst error is defined as a sequence of bits in which the first and last bits are in error, and any combination of errors or non-errors occurs in between. The code is effective in detecting errors as follows:

a) All single bit errors.
b) All combinations of two errors
c) All burst errors of eight bits or less.
d) Approximately 99.6% of all other error arrangements.

To provide even more effective detection of transmission errors when performing control functions, checkback is used. Actually, three messages are transmitted which must be in agreement: selection, checkback and activate. This redundancy, plus the cyclic code, provides an effectiveness which is very nearly 1ØØ percent.

4.9 System Security

Each message is exactly 32 bits in length, and starts with two synchronizing characters. The purpose of the synchronizing characters is to give the receiver a starting point for a message so it can begin registering the incoming bits.

In addition to the cyclic code check which is particularly effective in detecting burst errors, the use of select, checkback and activate messages for performing control functions provides an extremely high degree of insurance against improper operations. If an expected message is not received in its proper sequence, or if the cyclic code check indicates a transmission error, the remote station will take no action. If a device is selected for control and the remote station then receives any message other than a control execute message the selected control point is automatically reset.

At the RTU the method of status checking and comparison against a standard routine provides a measure of security over and above straight hardware checking. To be valid a message must be not only BCH acceptable, but it must also be in proper sequence to agree with the remote program.

5.Ø FUNCTIONAL DESCRIPTION

5.1 REMOTE STATION HARDWARE

5.2 PROGRAMMABLE CONTROL UNIT

5.3 I/O STRUCTURE

5.4 CONTROL AND DISPLAY PANEL

5.Ø FUNCTIONAL DESCRIPTION

5.1 REMOTE STATION HARDWARE

The AES 81/2Ø REMOTE TERMINAL UNIT comprises a programmable control unit, a data communications sub-system, a power supply, a digital input multiplexer, an analog input sub-system, and a digital output commutator.

5.2 PROGRAMMABLE CONTROL UNIT

5.2.1 The RTU is controlled by an AES 8Ø microprocessor. This control unit coordinates the data acquisition, control, and communications functions at the RTU via a fixed program resident in READ-ONLY MEMORY.

5.2.2 At the communications interface with the data modem, the control unit receives station scan and control messages and transmits selection and operation verifications as well as complete data telemetry messages.

5.2.3 At the system interface, the control unit scans analog, status, and alarm information through analog and digital input modules and outputs control actions through a digital output commutator.

5.3 I/O STRUCTURE

5.3.1 The control unit communicates with the input/output modules via a high speed party-line I/O bus. The I/O bus is a bi-directional common communication channel containing the data and control lines required for system communication. Communications between the control unit and an I/O module consists of two phases; the first phase is route set-up and the second is data transfer.

5.3.2 Each I/O module contains a party-line interface, mode and address logic, and one or more 8 bit data transfer registers. The control unit views the interface environment as a set of up to 256 full-duplex independently operating transfer registers.

5.3.3 Under RTU program convention, each DATA transfer register is associated with a specific processing function. Hence a given I/O handler communicates with a specific group of registers according to a fixed convention. Thus the strapping of an address on an interface module determines how the control unit will interpret data transfers.

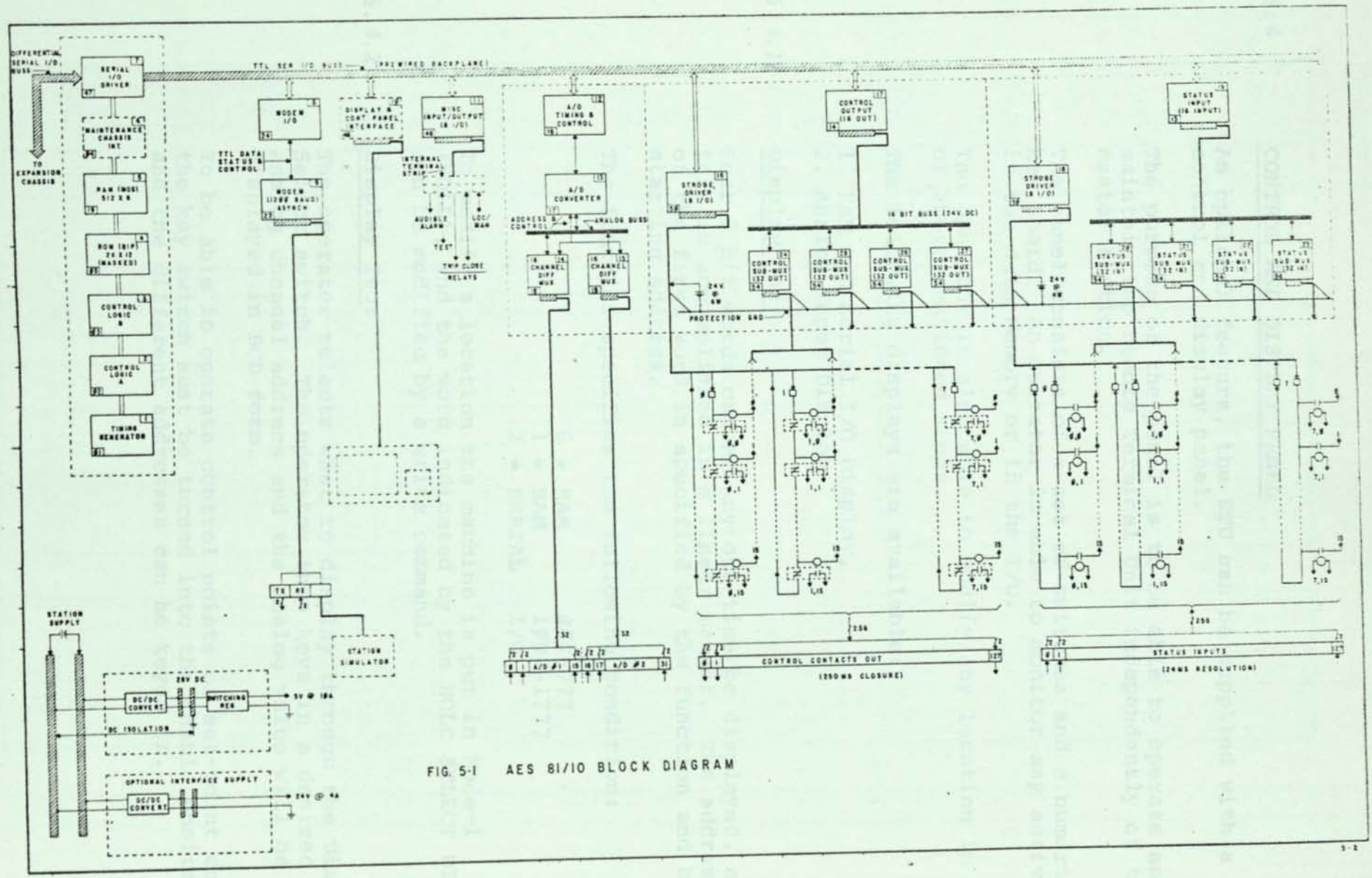


FIG. 5-1 AES 81/10 BLOCK DIAGRAM

5.4 CONTROL AND DISPLAY PANEL

An optional feature, the RTU can be supplied with a control and display panel.

The purpose of the panel is to be able to operate and maintain the Remote Terminal Unit independently of the master station.

The panel consists of a set of switches and a numerical key board. An operator is able to monitor any activity in the data Memory or in the I/O.

The operator is also able to modify any location in any of above mentioned areas.

The following displays are available:

1. RAM and Serial I/O Display,
2. Analog input Display

5.4.1 Display One:

Four 8 bit words can at any one time be displayed, and this is accomplished in a linear manner. The address of the first word is specified by the function and by the starting address.

The function specifies the following condition:

6 =	RAM	ØØØ-777
1 =	RAM	1ØØØ-1777
2 =	SERIAL	I/O

To modify a location the machine is put in Mode-1 (HOLD), and the word indicated by the HOLD SELECT DISPLAY can be modified by a write command.

5.4.2 Display Two:

The operator selects what to display through the Disp. Select switch. The operator then keys in a desired analog channel address and the analog value will be displayed in BCD form.

To be able to operate control points or set-point controls, the key switch must be turned into the enable position, and the different addresses can be keyed in.

AES-81 CONTROL & DISPLAY

AUTOMATIC ELECTRONIC SYSTEMS INC

ADD	7	6	5	4	3	2	1	0
A	O	O	O	O	O	O	O	O
A+1	O	O	O	O	O	O	O	O
A+2	O	O	O	O	O	O	O	O
A+3	O	O	O	O	O	O	O	O

PAR/SER. ADDRESS	KEYBOARD	OCTAL ADDRESS	ANALOG VALUE
0128	5721	026b	+1187 MV

OFF ON LOCK

LOCK

RUN	PAR ADD	INCR	ALARM ACK	LAMP TEST	
HOLD	SER ADD	DECR	TRANS MIT	RESET	

0	1	2	3	4
5	6	7	8	9
+	−		CLEAR	

CONT ADDR	S-P VALUE	CLOSE	LATCH ACT	WRITE	FALSE SEL ADDR
ANA ADDR	S-P ADDR	OPEN	MOM ACT	CANCEL	

6.Ø FIRMWARE

6.1 GENERAL

6.2 FOREGROUND

6.3 BACKGROUND (PROCESSING)

6.2.1 CONTROL SELECTION

6.3.2 SCAN

6.3.3. MESSAGE DECODER

6.3.4 DISPLAY

6.4 DATA MEMORY LAYOUT

6.Ø FIRMWARE DESCRIPTION

6.1 GENERAL

The program stored in the Read-Only Memory (ROM) is divided into a number of program levels.
Two basic levels exist, the foreground and the background program.
The foreground program handles all communication with input/output devices.
The background program consists of task routines and idle routines.

6.2 FOREGROUND

A task routine can only be initiated by an interrupt. When a task routine requires an input or output operation, the routine calls up a desired I/O handler located in the foreground program. The following interrupts have been assigned:

HIGHEST LEVEL	1	= REAL TIME, 1Ø mS.
	2	= COMMUNICATION RECEIVER
	3	= A/D - CONVERTER
	4	= COMMUNICATION TRANSMITTER
LOWEST LEVEL	5	= CONTROL PANEL.

The real time interrupt initiates the basic scanning sequence of all digital inputs, and increments a 16-bit counter. This counter is used throughout the software system for different timers and time out counters.

The communication receiver interrupt is set each time a message has been received by the communication subsystem. The I/O handler associated with this interrupt transfers the received 24 bit data word in three words of eight bits each, and the associated task routine is the message decoder. This routine validates the received message, and according to the content of the message initiates a set of operations. The basic message decoder is shown on flow chart No.2.

The converter interrupt is set each time a new reading is available from one of the four A/D converter subsystems. The associated I/O handler transfers the converted data from the subsystem to the image memory and initiates a new conversion. If the actual reading was related to a set-point control (digital) operation, the handler calculates the difference between the read value and the selected one and depending on the result, the handler generates either increase or decrease control signals for the selected set-point channel.

The transmitter interrupt is set when the transmitter buffer in the communication subsystem is empty and ready to accept a new 2 x 12 bit data message. If more than one message is contained in the response from the remote, the associated handler extracts required data from the image memory and formats three, eight bit words into the two by twelve bit word required by the message structure.

The lowest level of interrupt is associated with the control panel (optional). Each time a key or a switch is pressed on the control panel, the interrupt is set.

6.3 BACKGROUND

The background program consists of a set of low level routines each tied together in a program loop. See flow chart 1. The loop is entered when the system is turned on via the initialize routine. This routine comprises a RAM erase portion which clears all location's in the data memory, and a system configuration routine.
The system conf. routine establishes the current system size by scanning the different I/O registers. While doing so, the routine also allocates buffer space in the data memory for the different input classes as they are scanned. Each class of data is represented by a two byte word (2 x 8 bits), containing the starting address in the data memory for the class and an available flag, which indicates that one or more I/O register's has been connected for that particular class.

The following basic routines are contained in the background program:

6.3.1 THE CONTROL SELECTION ROUTINE:

Is stimulated by either the message decoder or the control panel routine when control of a digital output point has been requested. The routine makes the selection, and indicates to the message decoder that the selected point is operable. This check back operation initiates a confirmation message. Which is generated by the message decoder, but outputted by the transmit routine. When no selection sequence has been initiated, the control selection routine continuously scans the confirmation inputs from the selection relays, and verifies that no false selection has occurred through component failure.
If a point has been false selected, the remote does not reply with a proper selection confirmation message when requested to do so.

6.3.2 SCAN ROUTINES

When not receiving or acting upon a received message, the RTU performs internal scan routines for the purpose of gathering and storing status and data. The scan routine is performed at a rapid rate and consists of looking at the status input contacts and pulse accumulation input contacts to determine whether changes have occurred, multiplexing analog inputs and performing A/D conversions. Current status, indications of status change (one per status point), pulse accumulation count and data in digital form are all stored in read/write memory, ready for transmission to the master station when requested. The scan subroutines that perform these functions are triggered by pseudo interrupts. Included in the scan routine, of course, is the sensing of pseudo interrupts which will trigger subroutines that have to do with transmitting or receiving messages.

The scan routine is initiated by the real time interrupt. The routine is illustrated by flow chart 3.
Before the routine is entered, all digital input's are strobed into the associated I/O-registers.
The scan routine then processes one class of data at a time. Or in other words, for each pass in the background loop the routine executes all required processing for one class of data and then returns the control of the microprocessor to the general background program as shown on Fig.1. The scan routine uses the information generated by the system configuration routine to establish the availability of a specific class of inputs.

6.3.3 MESSAGE DECODER (Flow Chart 2)

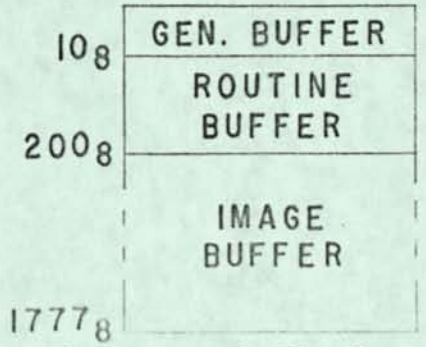
The message decoder routine is called up when a new message has been received by the remote. The routine checks the message for transmission error. If the message is valid, the routine proceeds to determine what action is to be taken by the remote, and also checks if the messages are received in a proper sequence for control selection and activation. The routine then calls up the appropriate subroutines for execution of the selected operation.

6.3.4 DISPLAY

The display routine is used to output selected data to the displays located on the control panel.
For each pass through the background loop, the display routine outputs one eight bit word to be displayed.

6.4 DATA MEMORY LAYOUT

The data memory is divided up in three areas as shown below.



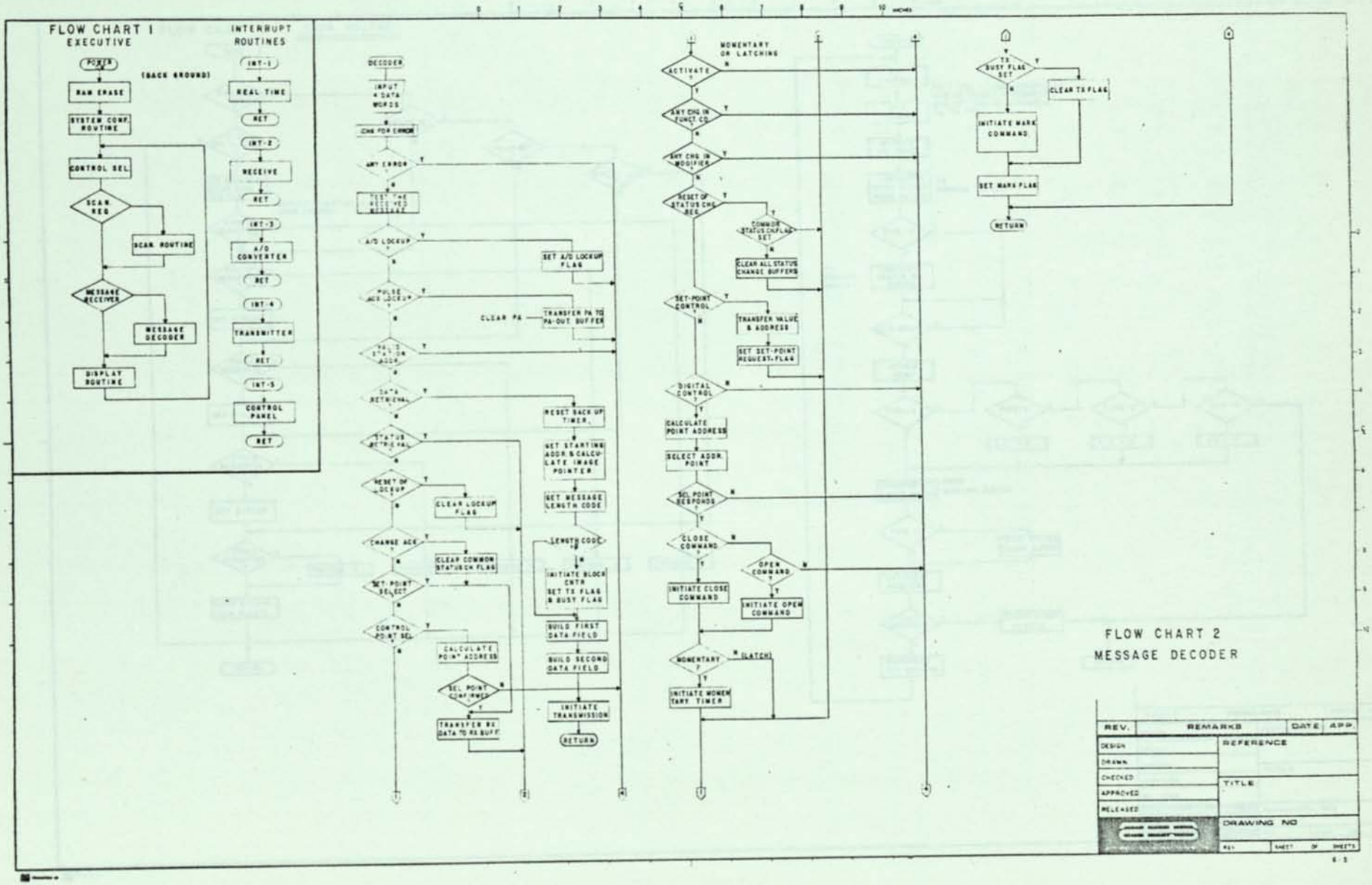
The general buffer area consistsof eight words, and is used as temporary storage of data during execution of a routine.

In the routine buffer area, all pointers and flag's associated with the different task routines are stored.

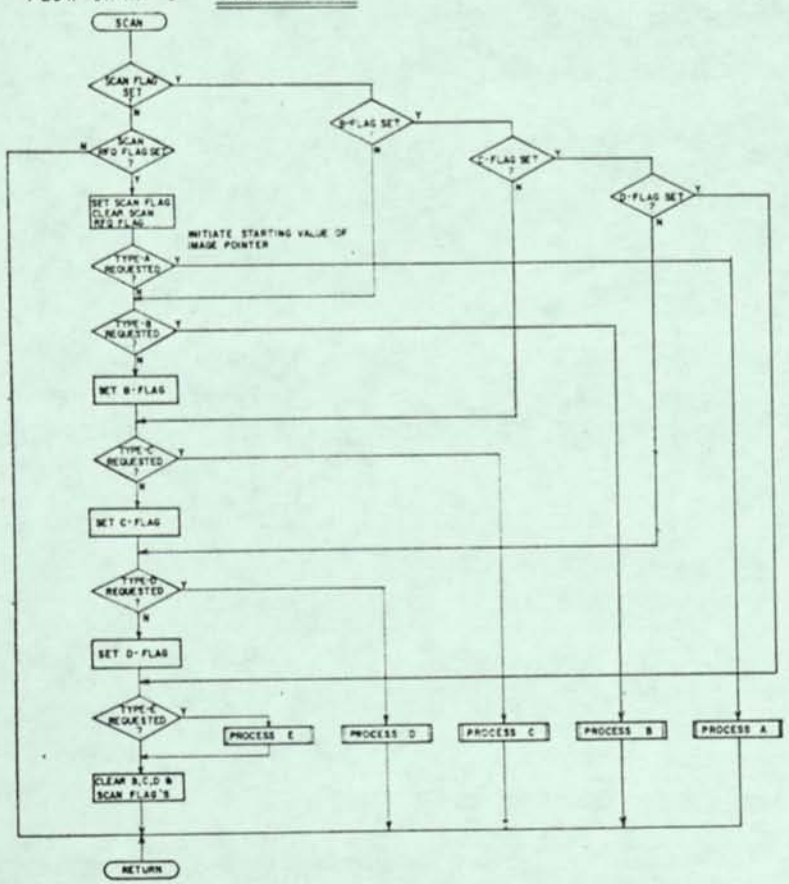
The image buffer is dynamically allocated by the system configuration routine as describe above the following rules are used for allocation of the different input classes.

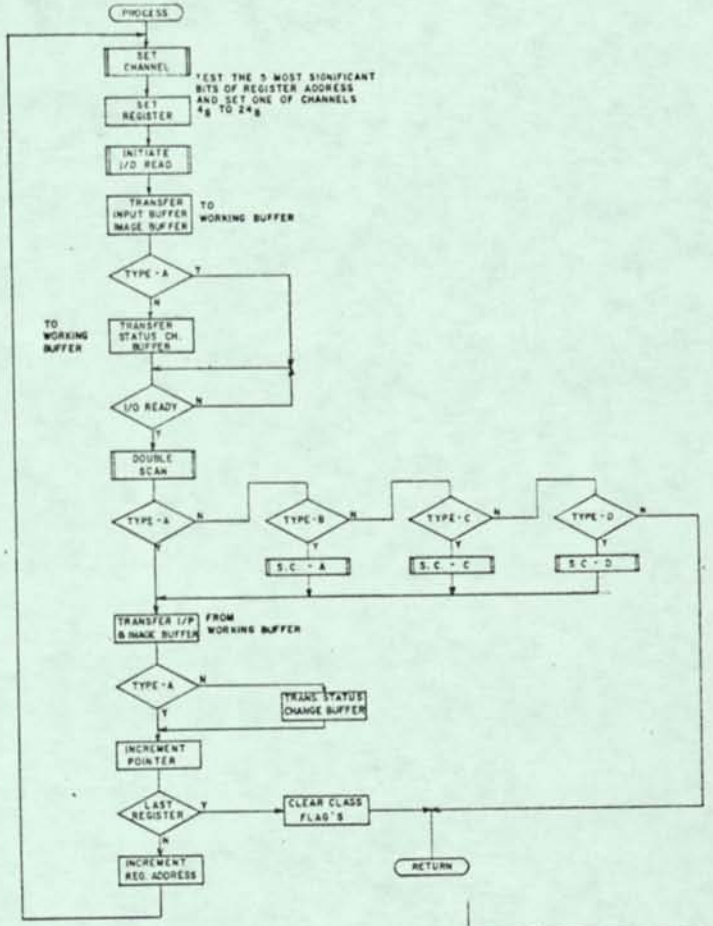
1) All input data except class-D are divided in groups of twelve bits, hence requiring $1\frac{1}{2}$ memory words.
   Class-D is divided in groups of eight bits.

2) All change information is interleaved with the status information, hence a twelve bit status group is followed by a twelve bit change group.


FLOW CHART 1
EXECUTIVE
INTERRUPT ROUTINES
(BACK GROUND)
RAM ERASE
SYSTEM CONF. ROUTINE
CONTROL SEL.
SCAN REQ
SCAN ROUTINE
MESSAGE RECEIVER
MESSAGE DECODER
DISPLAY ROUTINE
INT-1
REAL TIME
RET
INT-2
RECEIVE
RET
INT-3
A/D CONVERTER
RET
INT-4
TRANSMITTER
RET
INT-5
CONTROL PANEL
RET
DECODER
INPUT 4 DATA WORDS
CHK FOR ERROR
ANY ERROR
A/D LOCKUP
SET A/D LOCKUP FLAG
CLEAR PA
TRANSFER PA TO PA-OUT BUFFER
DATA RETRIEVAL
RESET BACK UP TIMER
SET STARTING ADDR & CALCULATE IMAGE POINTER
SET MESSAGE LENGTH CODE
LENGTH CODE
INITIATE BLOCK CNTR
SET TX FLAG & BUSY FLAG
BUILD FIRST DATA FIELD
BUILD SECOND DATA FIELD
INITIATE TRANSMISSION
RETURN
RESET OF LOCKUP
CLEAR LOCKUP FLAG
CHANGE ACK
CLEAR COMMON STATUS CH FLAG
SET POINT SELECT
CONTROL POINT SEL.
CALCULATE POINT ADDRESS
SEL POINT CONFIRMED
TRANSFER RX DATA TO RX BUFF.
MOMENTARY OR LATCHING
ACTIVATE
ANY CHG IN FUNCT CD.
ANY CHG IN MODIFIER
RESET OF STATUS CHG REQ
COMMON STATUS CH FLAG SET
CLEAR ALL STATUS CHANGE BUFFERS
SET-POINT CONTROL
TRANSFER VALUE & ADDRESS
SET SET-POINT REQUEST-FLAG
DIGITAL CONTROL
CALCULATE POINT ADDRESS
SELECT ADDR. POINT
SEL POINT RESPONDS
CLOSE COMMAND
OPEN COMMAND
INITIATE CLOSE COMMAND
INITIATE OPEN COMMAND
MOMENTARY
(LATCH)
INITIATE MOMENTARY TIMER
TX BUSY FLAG SET
CLEAR TX FLAG
INITIATE MARK COMMAND
SET MARK FLAG
RETURN
FLOW CHART 2
MESSAGE DECODER
REV.
REMARKS
DATE
APP
DESIGN
DRAWN
CHECKED
APPROVED
RELEASED
REFERENCE
TITLE
DRAWING NO
SHEET OF SHEETS

FLOW CHART 3 SCAN ROUTINE





REV.	REMARKS	DATE	APP.
DESIGN			REFERENCE
DRAWN			
CHECKED			
APPROVED			TITLE
RELEASED			
			DRAWING NO
			SIZE SHEET OF SHEETS

7.Ø TECHNICAL CHARACTERISTICS

7.1 SERVICE CONDITION

7.2 MEMORY

7.3 I/O BUSS

7.4 MULTIPLEXED CONTACT INPUT

7.5 MULTIPLEXED CONTACT OUTPUT

7.6 MULTIPLEXED ANALOG INPUT

7.7 MODEM

7.8 POWER

7.9 TRANSIENT PROTECTION

7.1 SERVICE CONDITION

NON-OPERATING CONDITION

All units of the system are designed in such a way that no mechanical or electrical damage and no permanent degradation in performance will result when stored or transported in its shipping crate under any combination of the following environmental conditions:

i) TEMPERATURE

From -3ø°C to +9ø°C

ii) HUMIDITY

From zero to 99% without condensation

iii) ALTITUDE

Sea level to 2ø,øøø Feet.

iv) ORIENTATION

Any position

v) SHOCK AND VIBRATION

Transportation and handling over secondary roads.

OPERATING CONDITION

i) TEMPERATURE

From -25°C to +85°C

ii) HUMIDITY

From zero to 99% without condensation

iii) ALTITUDE

From sea level to 15,øøø feet

iv) SHOCK AND VIBRATION

Normal shock and vibration as encountered in a fixed installation.

7.2 MEMORY

i) DATA MEMORY

TYPE = RANDOM ACCESS
SIZE = 512 WORDS x 8 BITS
TECHNOLOGY = MOS

ii) INSTRUCTION MEMORY

TYPE = READ ONLY (ROM), RANDON ACCESS
SIZE = 2k WORDS x 12 BITS
TECHNOLOGY = BIPOLAR (MASKED)

7.3 INPUT/OUTPUT BUSS

Two different busses are used, one buss which is connected to all I/O devices in the main processor chassis, and one buss which connects the processor to expansion chassis. The buss inside the processor chassis uses logic levels according to established standards for TTL (three-state) type of integrated circuits.

The extender bus is of differential type, with a driving capability of 4Ø mA (sink/source) and a common mode voltage rejection of up to ±15V at the receiving end.

7.4 MULTIPLEXED CONTACT INPUT

The input contacts are organized in a matrix, see system drawing. The columns of the matrix are driven by a strobe driver, which enables a set of 16 input contacts. These 16 inputs are submultiplexed together with another set of 16 inputs in a status sub mux board, the outputs (16) of which are connected to the status buss and sensed by a status input board.
All isolation and filtering is done in the status input board.

INPUT CHARACTERISTICS

i) ISOLATION:

a) 2.5KV PEAK between any 2 signal points or between any signal and signal return

b) 35VDC steady state as in a)

c) In case of floating input, common mode signal potential limited to ±3ØØVDC or 6ØØV p-p AC w.r.t logic grd.

ii) RC TRANSIENT LIMITING
FILTER DELAY = 3mS

iii) Signal conditioning time constant variable Ø-3mS

7.5 MULTIPLEXED CONTACT OUTPUT

The digital control outputs are organized in a matrix, as shown on the system drawing.
The select relays are divided in groups of 16. All relays in one group have one side of the coils connected together and are driven by an output of the strobe driver board. The other side of the coil is drive in a linear way by the control sub-mux board.

OUTPUT CHARACTERISTICS

i) ISOLATION: as per 7.4. i)

ii) CLOSURE TIME: 1 sec max; 25Øms typical

7.6 MULTIPLEXED ANALOG INPUT

Signalling conditioning and transient protection on mux board.

RESOLUTION	=	12 BITS
ACCURACY OF FULL SCALE	=	.1% (Ø to 7Ø°C)
INPUT RANGE	=	±1ØV
CONVERSION RATE	=	25 CONV/SEC
DIFF. REJECTION @6ØHz	=	3Ø db
COMMON MODE REJECTION @ 6ØHz @ 1ØØVDC OR 145VAC	=	1ØØ db
No. OF INPUTS/MUX BOARD	=	16
No. OF INPUTS/SUBSYSTEM	=	32
TRANSIENT PROTECTION	SEE	7.4. i)

7.7 MODEM I/O

SPECIFICATIONS

OPERATING MODE:	Full-or half-duplex selected under software control.
DATA RATES:	15Ø, 3ØØ, 6ØØ, 12ØØ baud. Two speeds selectable under software control.
DATA FORMAT:	Two start bits, 29 data bits, one stop bit.
ORDER OF BIT TRANSMISSION:	Low order bit first
DISTORTION INPUT:	Up to 4Ø% per data frame allowed
DISTORTION OUTPUT:	Less than 3% per data frame
DATA SIGNALS:	TTL compatible for connection to AES modem. (Connections on backplane connector) optional EIA RS-232-C interface available on front edge connectors.
CONTROL SIGNALS:	All control signal connections compatible with Bell 1Ø3 and 2Ø2 type modems and are optionally available on front-edge connector (EIA-RS-232-C).
CLOCK:	Crystal controlled frequency of two times selected baud rate available at backplane connector for use by AES modem.
POWER REQUIREMENTS:	+ 5V ± 5% 1.3 A max.
GENERATED USING DC TO DC 5V TO ±15V CONVERTER	+12V ±20% 3Ø mA max.
	-12V ±20% 3Ø mA max.
	Incl. in -5V consumption

OPERATING TEMPERATURE:	$-20^{\circ}C$ to $+85^{\circ}C$
HUMIDITY:	20% - 95% non condensating
PHYSICAL:	One PC Board, size $7\frac{1}{2}$" x $7\frac{1}{2}$"

7.8 MODEM

7.8.1 OUTPUT CHARACTERISTICS

Asynchronous FM signal

MARK frequency	:	2400Hz	1200Hz	Strapping option
SPACE Frequency	:	1200Hz	2400Hz	
Output impedance	:	600ohms ±5%		
Output level	:	Adjustable up to -5dBm		
Transmission rate	:	$\frac{1200 \text{ Baud}}{N}$		

N = Integer number

7.8.2 RECEIVER CHARACTERISTICS

Input signal	:	Asynchronous FM signal
MARK frequency	:	2400Hz reversable by strapping
SPACE frequency	:	1200Hz
Input impedance	:	600Ohms ±5%
Receiver sensitivity	:	Receiver gain can be adjusted for line signal levels between -5dbm and -55dbm
Loss of Carrier Alarm	:	Threshold level for this alarm can be set to any level between 0dbm and -40dbm Interface: TTL compatible Alarm = logic 1

Data rate : Same as being transmitted from opposite end (must be: $\frac{1200 \text{ Baud}}{N}$

N = Integer

Receiver output : TTL compatible

Logic 1 = MARK

Logic Ø = SPACE

Operating modes : Half duplex only over two wire channels. Half duplex or full duplex over four wire channels.

7.8.3 SPECIFICATIONS

7.8.3.1 TRANSMISSION MEDIUM REQUIREMENTS

- 3kHz bandwidth voice channel transmission facilities
- A wire compatible with AT and T type 3ØØ2 unconditioned individual leased channels.

POWER REQUIREMENTS: +5V ±5% 1.ØA

ENVIRONMENTAL:

Temperature -20°C to +85°C
Humidity 10% to 95%
non condensating

Physical
PC Board size 7½" x 7½"

SIGNAL CONNECTIONS
CONNECTIONS TO LINES: on 44 pin front card edge connector

LOGIC LEVEL CONNECTIONS
AND POWER: 1ØØ pin back card edge connector

7.8 POWER

1. SYSTEM POWER

INPUT LEVEL = 1) 24 V
2) 48 V
3) 129 V

POWER REQUIREMENT = 15ØW

2. INTERFACE POWER (OPTIONAL)

INPUT LEVEL = 1) 24V
2) 48V
3)129V

POWER REQUIREMENTS = 1ØW

7.9 TRANSIENT PROTECTION

The inputs and outputs are designed to withstand surge voltage spikes at power network interfaces having a crest of 2.5kV at a frequency of 1.ØMhz to 1.5Mhz. The envelope of the first cycle will delay to 5Ø% of the crest in not less than 6.Øms after the start of the wave.