THE AES-80 MICROPROCESSOR

REFERENCE MANUAL

AUTOMATIC ELECTRONIC SYSTEMS INC., 5455 PARE ST., MONTREAL 309 CANADA, TEL. (514) 735-6581
AES DATA INC., P.O. BOX 143 ST ALBANS, VERMONT 05478, TEL. (802) 524-3660

8Ø

MICROPROCESSOR

REFERENCE MANUAL

AUTOMATIC ELECTRONIC SYSTEMS INC.,
5455 Pare Street
Montreal 309

TEL.: (514) 735-6581

AES MICROPROCESSOR

TABLE OF CONTENTS

1. SYSTEM DESIGN FEATURES

1.Ø. General Characteristics

The AES microprocessor is a byte oriented general purpose computer designed primarily for OEM use in dedicated applications. The main features and characteristics which describe its operation can be summarized as follows:

- Instruction Memory Size - up to 4Ø96 12-bit words
- Data Memory Size - up to 4Ø96 8-bit words
- Tri-State Data bus
- 2k direct addressing of Instruction Memory
- 1k direct addressing of Data Memory
- 24Ø nanosecond instruction time (full cycle) assuming all bipolar memory
- Single clock cycle (1 state) per full instruction cycle
- 6 Registers
- All modes of 8-bit ALU under software control
- Easy interchange and intermix of memory types and speeds within the same processor
- Both parallel and serial I/O capability
- 16 level automatic push down stack for routine linkage
- Comprehensive instruction set. (49 basic one word instructions plus 43 arithmetic and logic instructions)
- TTL Integrated Circuitry
- Operating Temperature Ø°C to 7Ø°C or -55°C to 125°C depending upon grade of IC's used.

The microprocessor executes one complete instruction during one cycle of the basic timing clock. There are no sub-cycle time slots, or states, used within the basic instruction cycle. Consequently, the power of the individual instructions are somewhat less than those of a higher level processor, such as a "minicomputer". However these higher level instructions are available by writing "micro-programs" or "micro-routines" and give the flexibility of, in effect, writing an instruction set in addition to writing higher level routines.

1.1. Physical Configuration

The basic AES microprocessor is contained in one AES modular system unit. The MSU is a standard package configuration with the capability of containing 9 plug in cards. All cards are 7.ØØ" x 7.35" and spaced Ø.6" apart. The cards are interconnected either by an artwork or by a wire-wrap backplane. The microprocessor MSU contains the following cards:

1) Timing Generator
2) Control Logic A
3) Control Logic B
4) 1Ø24 x 8 Data Memory (RAM)
5) 2Ø48 x 12 Instruction Memory (ROM)

Most configurations leave a slot reserved for a maintenance and control interface card that interfaces the microprocessor to a program development and control console, thus enabling the operator to monitor and/or control the microprocessor during maintenance, test or programming.

The remaining 3 positions can be used for a variety of functions. For example if more ROM or RAM capability is required, then a combination of these cards can be inserted into the remaining positions.

Another slot may be reserved for the serial Input/Output control card. There are two versions of this card. In the simpler version the I/O address select, data and control lines are all TRI-state* outputs and inputs. The more complex version of the I/O card is identical to the simpler one in all respects except that differential line drivers and receivers are used in place of the TRI-state logic. The latter card, whose primary purpose is increasing the allowable load on the I/O bus, is used in cases where the I/O bus is longer than 2Ø ft and/or when more than 5 I/O channels are used. The remaining two slots may be filled by two parallel I/O interface cards, or 1 interface card and one parallel I/O buffer expander. The parallel I/O buffer card is meant (as in the case of the complex serial I/O card) for increasing the allowable I/O bus loading.

The ROM memory card may be removed and in its place a ROM simulator interface card inserted. This card is used to interface the microprocessor to a ROM simulator which is a part of the program development and control console. The ROM simulator, which is actually a fast bipolar random access memory, behaves exactly as a ROM card where the microprocessor is concerned. It is, however, possible to write instructions into the ROM simulator via an ASR 33 TTY, or a tape reader, thus enabling the programmer to develop and verify the final software before masking the program into the Read Only Memory.

1.2. System Organization

The AES microprocessor is a bus organized machine designed around a data transfer concept. An 8-bit TRI-state processor bus is used as the main highway for data traffic between registers and 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 of Figure 1-1.

1.3. Instruction Memory

Commands from the read-only instruction memory control all aspects of the microprocessor operation and all commands 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

---

* National Semiconductor Corp. TM.

upon the particular instruction decoded, literal data can be outputted directly from ROM.

1.4. Data Memory

The Data Memory is available with from 1 to 16 256 word by 8-bit modules for a maximum capacity of 4Ø96 words. These modules can be selected from any of the following types and intermixed within the same processor.

Type 1 High speed parallel bipolar scratch pad memory. This type of 256 word memory module has a cycle time faster than the basic machine cycle time of 24Ø nanoseconds. Thus data can be read out of or written into this memory during a single machine clock cycle.

Type 2 Non volatile random access read/write core memory module. This type of memory module has a cycle time of 1 micro-second.

Type 3 MOS static random access read/write memory module. This module is partially powered by a rechargeable battery for standby power applications so that the memory remains non volatile for up to 48 hours after external power is removed.

Type 4 Bipolar read only memory module.

Type 5 Special purpose function modules.

As can be seen from figure 1-1, data is both read out of and written into data memory via the high speed 8-bit Tri-State processor bus. Thus, for real time applications, where time is at a premium, memory modules of type 1 should be used because of its speed. This type of memory module also satisfies the need for high speed general storage and working registers.

Memory module types 2 or 3 should be used in cases where momentary power failure can occur but where it is necessary to retain data stored into data memory prior to the power off condition.

Bipolar ROM modules may be used within the data memory bank for storing constants and other predefined data which will never be altered during a program.

Special purpose hardware options such as hardware multiply or fast parallel I/O may also be accommodated by using preselected Data memory address slots. This will be more fully explained in sections 1.1Ø, 3.Ø, 3.3 and 3.4.

1.5. Arithmetic Logic Unit (ALU)

The arithmetic logic unit operates on two 8-bit variables the tri-state 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.

1.6. Registers

1.6.Ø. P-Register: The 12-bit P (Program Counter) register indicates the address of the next instruction to be fetched out of instruction memory. The P-register automatically increments by one after the execution of each instruction except in the following cases. If the previous instruction was one of the following:

- a conditional branch instruction whose jumping criteria has been met,

- an unconditional jump instruction, or

- a jump to subroutine instruction,

the least significant 11 bits of the P register are loaded with the ROM output data, thus defining the current page jump address. Similarly if the previous instruction was a return from subroutine instruction, the P-register is loaded with the return address last stored into the push-down stack.

1.6.1. A-Register: The 12-bit A (Data Memory Address) register holds the address of the data memory cell being read from or written into. During a RAM address command, its contents may be altered and the 1Ø-bit operand field of the instruction loaded into the least significant 1Ø bits of the A register.

In addition, three instructions alter the A-register so that:

- the contents of the A register may be incremented,

- the least significant 8 bits of the A register may be loaded with the data present on the TRI-state processor bus, or

- the most significant 4 bits of the A register may be loaded with the 4 least significant bits of the processor bus.

1.6.2. L-Register: The 8-bit L (Literal) register is loaded with the 8-bit literal field of a literal data instruction. If the output of the L register is enabled, an 8-bit literal is available on the processor bus.

1.6.3. 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.

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

a) partially loaded by the 4 least significant bits of the ALU output,

b) partially loaded by the 4 most significant bits of the ALU output,

c) completely, loaded by all 8 output bits of the ALU,

d) rotated right by one bit, or

e) cleared.

1.6.5. 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. To output an 8-bit character onto the serial I/O data bus, data is loaded into the U register from the TRI-state processor bus. When an I/O output command is initiated, the data from the U-register is automatically shifted out onto the I/O data bus. The data is also recirculated back into the U-register so that the character can be retransmitted if necessary. To input a character from the I/O data line, an input command is initiated and the 8-bit character is automatically shifted into the U-register. Appropriate instructions permit the U register to be merged (inclusive "OR" ed) with the data on the TRI-state processor bus, and also to be cleared.

The U-register is also a temporary storage buffer of data on the TRI-state processor bus. It is commonly used as a temporary storage for one byte of a 2 byte address pointer (2 word indirect address) from data memory.

1.7. Decision Flags

Within the microprocessor there are various status flag bits which are addressed by the appropriate decision instruction. These decision flags are tested by the microprogram to determine whether (or not) a conditional branch operation is implemented. If the decision instruction is decoded, for example, as "branch if decision flag 5 = logic 1", the next instruction will be interpreted as a jump address, or ignored, depending upon whether the decision flag was logic "1" or logic "Ø" respectively. Each decision flag may be tested for a logic "1" or a logic "Ø" condition permitting the use of both branch if "Ø" and branch if "1" instructions.

1.8. Push Down Stack

The AES microprocessor has a 16 level automatic push down stack which is used for routine linkage. When a jump to subroutine command is read from the instruction memory, the contents of the P register plus "2" is stored into the push down stack as a return address to be used when returning from the subroutine. After the return address is stored, the stack is virtually pushed down so that a lower level return address may be stored. This occurs when the subroutine itself calls up yet another subroutine. The next instruction will then automatically be interpreted as a jump address defining the starting location of the subroutine. When a return from subroutine command is read from ROM, the return address will be transferred to the P register from the push down stack during the next machine cycle. When an overflow (e.g. greater than 16 subroutine levels) or an underflow occurs in the push down stack, a decision flag is set.

1.9. Instruction Timing

The basic clock is a 12.5 MHz crystal oscillator. The frequency of this oscillator is divided down by 3, producing the microprocessor master clock. This clock has a pulse width of 8Ø nano-sec. and a total period of 24Ø nano-sec. as shown in Fig. 1.2. This period forms the basic machine cycle time and all instructions are executed within this time slot. In some cases, where the instruction or data memory has an access time longer than 16Ø n sec, a memory ready flag from the memory device being accessed may be used to lengthen or delay the master clock in increments of 8Ø nano-sec. This feature of having a variable cycle time is normally not used. It does, however, provide the flexibility of intermixing both slow and fast memory types within the same microprocessor without completely slowing down the master clock to accommodate the slowest memory module. It should be noted that only the leading and trailing edges of the master clock pulse are used for strobing or setting the various logic functions throughout the control logic of the microprocessor. No other sub-clocks delay lines or one-shots are used for timing purposes. When the P register is incremented or loaded, the new instruction address is available. After the memory access time has been reached, data is available for instruction decoding. Both the leading and trailing edges of the next clock pulse are used for executing the instruction just read. The detailed timing diagrams for the various instructions are provided in section IV.

1.1Ø. Input/Output Interface

The AES microprocessor I/O interface provides the necessary timing and control to communicate with both low and high speed peripheral devices. The serial or low speed I/O bus consists of 8 address lines, (defining the peripheral device address), one read/write line, one load strobe line, 3 I/O clock lines, one flag status line, one interrupt flag, one interrupt acknowledge line and one serial data

line. This I/O bus is used to transfer 8 bit serial characters into or out of the microprocessor at rates up to one character every 9.12 micro-seconds.

As soon as the selected peripheral has been addressed, and the data to be transferred is ready either in the U-register (for transmitting) or on the addressed I/O device (for receiving), 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. As soon as the I/O transfer is complete, the I/O ready decision flag is set. This enables the microprocessor to branch when I/O is complete. In addition to the serial I/O, a parallel I/O capability is available. This is normally used as a means of providing hardware processor options such as hardware multiply/divide or sine/cosine function hardware 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 parallel I/O bus consists of 12 address lines, 8 I/O data lines, a write strobe, a read enable line, an interrupt flag, a device ready flag and an interrupt acknowledge line. The I/O address and data lines are the same as those used for accessing the microprocessor data memory, i.e. the 8 bit TRI-state processor bus is the same as the I/O data bus, and the A-register outputs are also the 12 addressing lines for the parallel devices. A power reset pulse is sent to all peripherals both on the serial and parallel I/O bus when the power on reset condition is present.

A more detailed description of both the serial and parallel I/O structure is given in section III.

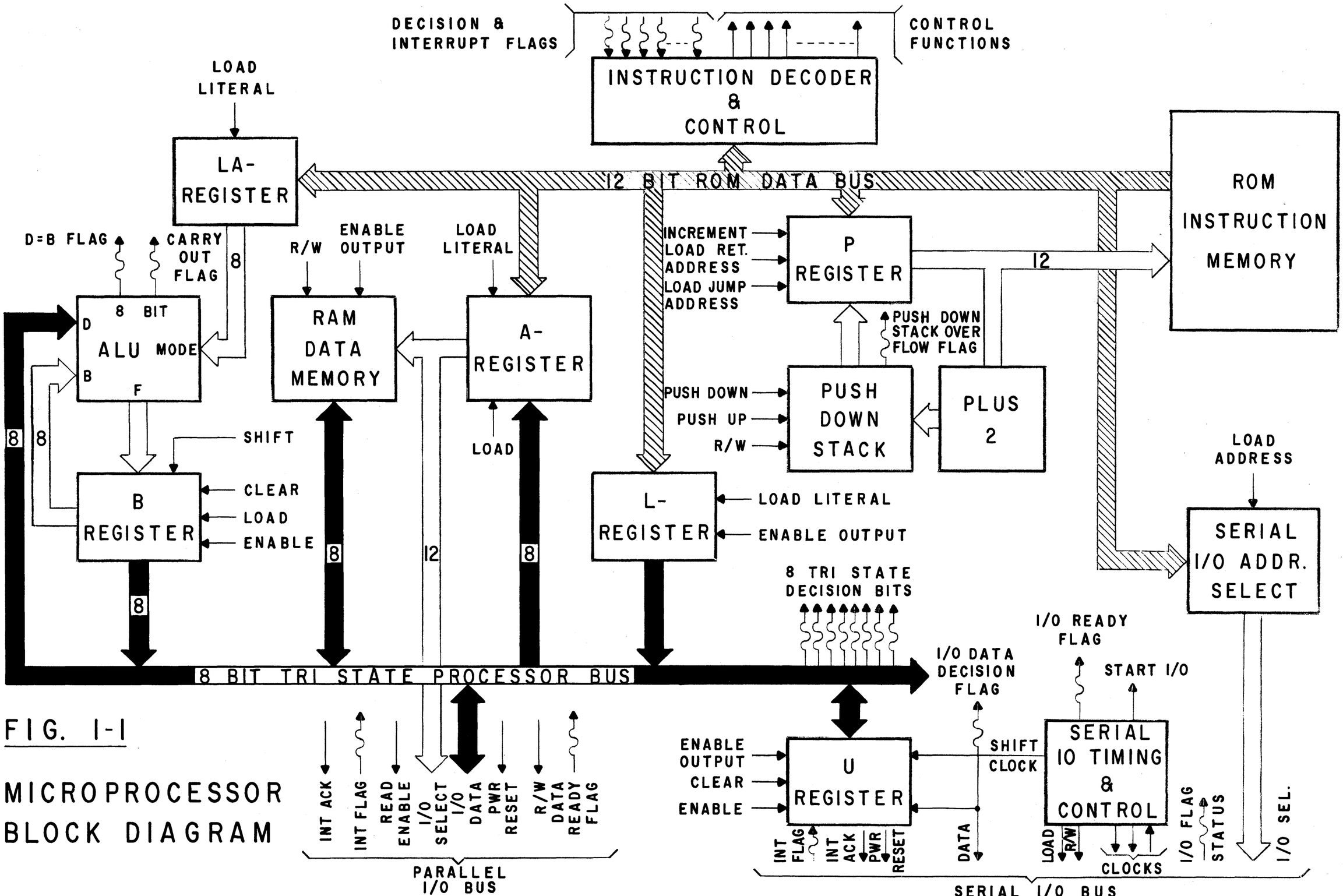


FIG. 1-1

MICROPROCESSOR BLOCK DIAGRAM

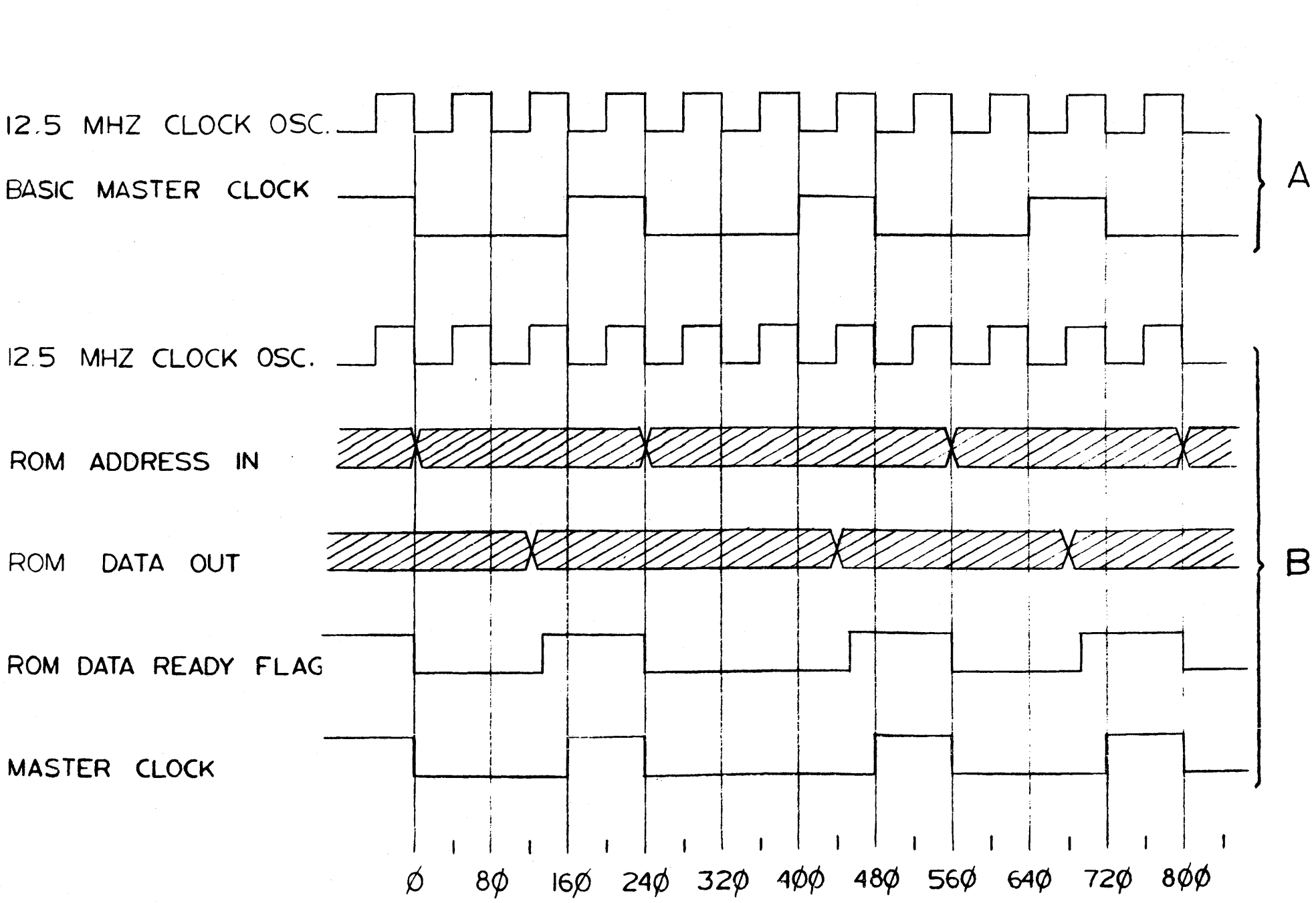


FIG. 1.2 MASTER CLOCK TIMING (A) CONSTANT ACCESS TIME LESS THAN 160N SEC.
(B) VARIABLE ACCESS TIME

II. Micro-Instruction Repertoire

The AES micro-processor has 92 basic one-word instructions, all executable in 24Ø nanoseconds. This section describes all of these micro-instructions. With each description is a diagram showing the format of the command, the mnemonic used in referencing it and also the two character ASCII code. The latter is used as the binary paper tape format when loading the ROM simulator via an ASR 33 TTY or paper tape reader.

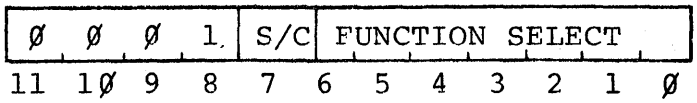
2.Ø. Instruction Formats

There are six basic instruction formats. Each micro-instruction is 12 bits in length and is contained in a single read-only memory location.

The formats are for function, decision, ROM address, RAM address, Data Literal and ALU literal micro-instructions.

2.Ø.Ø. Function

The function micro-instructions have the following format:



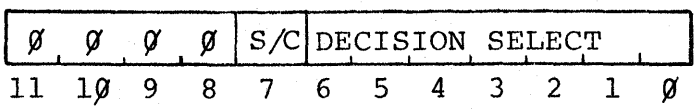
The function type of micro-instruction is used for setting or clearing various control signals within the machine. The simplest type of function is a latching function. In this case, execution of the instruction implies setting or clearing the logic signal defined by the function select code, depending upon whether bit 7 is a 1 or a Ø respectively. The second type of function is a strobing function. This type of instruction sets the addressed function line for one machine cycle period only. This type of function is used for strobing latches and clocking registers etc. Bit 7 is not decoded in this case.

The third type is a mutually exclusive function. The setting of one function within a mutually exclusive group also implies the clearing of all other functions within that group. If one mutually exclusive function is cleared, all functions within that group will also be cleared.

The last type of function is a strobe, branch type of instruction. This behaves like a strobe function, but in addition the next instruction is unconditionally treated as a branch address. The only function of this type is the jump to subroutine instruction explained in section 2.11.

2.Ø.1. Decision

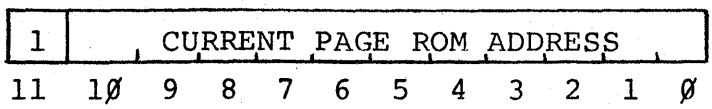
The decision micro-instruction has the following format.



When bits 8-11 of the micro-instruction are Ø, the first seven bits form a selection address to interrogate one of 2^7 possible decision flags. If the flag is equal to the value of bit 7, the next micro-instruction will be treated as a branch address.

2.Ø.2. ROM Address

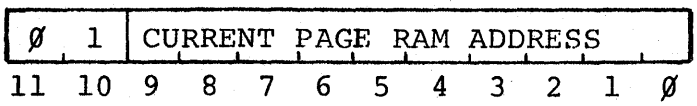
The ROM address micro-instruction has the following format:



The ROM address micro-instruction is defined by a 1 in bit 11. After the execution of a valid decision or a strobe branch function, the least significant 11-bits of the next word of memory data coming from ROM is loaded into the P register. Bit 11 of the P register remains unchanged, A ROM address command located anywhere else in a micro-program is ignored. The combination of a decision followed by a ROM address can be thought of as a two word conditional branch instruction.

2.Ø.3. RAM Address

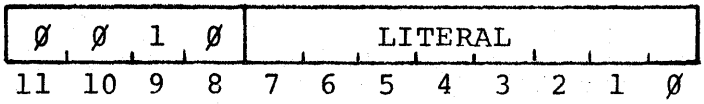
The RAM address micro-instruction has the following format:



When bits 11 and 1Ø are Ø and 1 respectively, the remaining bits are stored into the least significant 1Ø-bits of the A register. Bits 1Ø and 11 of the register are left unchanged.

2.Ø.4. Data Literal

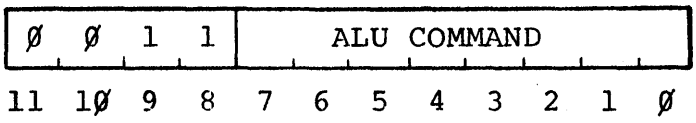
The Data Literal address micro-instruction has the following format:



When bits 11, 1Ø, 9 and 8 are Ø, Ø, 1 and Ø respectively, the 8 bit literal, defined by bits Ø-7, is stored into the L-register.

2.Ø.5. ALU Literal

The ALU Literal address micro-instruction has the following format:

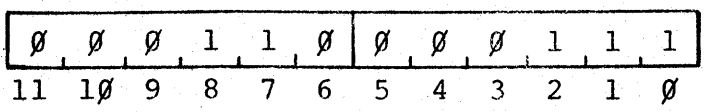


In a similar fashion to the data literal micro-instruction, the ALU literal command enables the 8 bit literal defined by bits Ø-7 onto the LA register.

2.1. Terms, Mnemonics and Symbols

The AES microprocessor's instruction memory consists of 12-bit words. For convenience of programming, entering data, printing out, and preparing punched paper tape, the 12 bits are organized into two modified ASCII characters of 6 bits each. These characters are a subset of the USA Standard Code for Information Interchange (USASCII). A conversion table between 6 bit binary to 2 number octal to ASCII is given in Appendix 1. To use this table , the 12 bit instruction is split into two 6 bit characters. The modified ASCII representation of the instruction is simply the most significant 6 bit character followed by the least significant character represented in modified ASCII.

For example:
The modified ASCII representation of the following function



is FG.

Some of the symbols and terms used in the description of instructions are:

$A_1=A_2$	Set contents of A_1 equal to A_2. A_2 is left unchanged.
JIS,xxx	Jump if flag xxx is set.
JIC,xxx	Jump if flag xxx is clear.
#	Logical OR operation.
.	Logical AND operation.
↑	Logical EXCLUSIVE OR operation.
A'	Logical Complement of A.
+	Arithmetic plus operation.
-	Arithmetic minus operation.

The rest of section II is devoted to describing the AES Microprocessor instructions. The detailed functional programming block diagram shown in Fig. 2.1. may be used as a guide in understanding these commands.

2.2. Load Data Bus Instructions

Mnemonic	ASCII	Description
D=L	F@	Load data bus with contents of the L-register.
D=M	FA	Load data bus with data memory output.
D=U	FB	Load data bus with contents of the U-register.
D=B	FC	Load data bus with contents of the B-register.

These four instructions form a mutually exclusive function set.

2.3. Load Literal Buffer Instruction

Mnemonic	ASCII	Description
L = xxx	H α I α J α K α	Load L-Register with octal number xxx (xxx may take on the values Ø - 377₈). α will be one of the 64 ASCII characters listed in Fig. 2.1.

2.4. ALU Mode Instructions

This group comprises the instructions for performing logical and arithmetic operations on two 8-bit quantities namely D (8 bit processor data bus) and B (B-register output), and providing an 8 bit quantity as an output, termed F. This output may be stored into the B register by using the appropriate instruction. The ALU mode of instructions may further be subdivided into the following groups.

Loading Instructions

Mnemonic	ASCII	Description
F=D	L@	Set the ALU output equal to the data bus.
F=D'	LP	Set the ALU output equal to the complement of the data bus.
F=B	LZ	Set the ALU output equal to the B-register output.
F=B'	LU	Set the ALU output equal to the complement of the B-register output.
F=-1	LC	Set the ALU output equal to minus 1 (2's complement), i.e. all bits of F are logic 1.
F=Ø	LS	Set all bits of the ALU output to logic Ø.

Logic Instructions

Mnemonic	ASCII	Description
F=D#B	LA	Set the ALU output equal to the logical or of D and B.
F=D#B'	LB	Set the ALU output equal to the logical or of D and $\overline{B}$
F=D'#B	LX	Set the ALU output equal to the logical or of $\overline{D}$ and B.
F=D'#B'	LT	Set the ALU output equal to the logical or of $\overline{D}$ and $\overline{B}$.
F=D.B	L[	Set ALU output equal to the logical and of D and B.
F=D.B'	LW	Set ALU output equal to the logical and of D and $\overline{B}$.
F=D'.B	LR	Set ALU output equal to the logical and of $\overline{D}$ and B.
F=D'.B'	LQ	Set ALU output equal to the logical and of $\overline{D}$ and $\overline{B}$.
F=D↑B	LV	Set ALU output equal to the exclusive or of D and B.
F=D↑B'	LY	Set ALU output equal to the exclusive or of D and $\overline{B}$.

Arithmetic Instructions

Mnemonic	ASCII	Description
F=D+D	LL	Set ALU output equal to D plus D.
F=D+B	LI	Set ALU output equal to D plus B.
F=D+D+1	L,	Set ALU output equal to D plus D plus 1.
F=D+B+1	L)	Set ALU output equal to D plus B plus 1.
F=D-B	L&	Set ALU output equal to D minus B. (negative numbers are in 2's complement).

Mnemonic	ASCII	Description
F=D-B-1	LF	Set ALU output equal to D minus B minus 1.
F=D+1	L(space)	Set ALU output equal to D plus 1.
F=D-1	LO	Set ALU output equal to D minus 1.

Combined Logical and Arithmetic Instructions

In the following group of instructions, it is assumed that logical operations are done before the arithmetic ones.

Mnemonic	ASCII
F=D#B+D	LM
F=D#B'+D	LN

Mnemonic	ASCII	Mnemonic	ASCII
F=D#B+1	L!	F=D#B'+1	L"
F=D#B+D+1	L-	F=D#B'+D+1	L.
F=D.B+D	LH	F=D.B'+D	LD
F=D.B+D+1	L(	F=D.B'+D+1	L$
F=D.B-1	LK	F=D.B'-1	LG
F=D#B+D.B'	LE	F=D#B'+D.B	LJ
F=D#B+D.B'+1	L%	F=D#B'+D.B+1	L*

Shift Rotate Instructions

Mnemonic	ASCII	Description
F=BSL	LL	Set the ALU output equal to the B-register shifted left one bit. The LSB of the ALU output is set to Ø.
F=BRL	ML	Set the ALU output equal to the B register rotated left one bit. Thus the LSB of the ALU output is equal to the MSB of the B-register.
EBR	N@	Set the B-register into the rotate mode.

The first two shift rotate instructions require that the B-register is enabled onto the data bus (D=B Is the last load data bus instruction to be executed).

2.5. Load Accumulator Instructions

Mnemonic	ASCII	Description
B=Ø	PH	Clear the accumulator or B-register.
B=F	FI	Load the B-register with the output of the ALU.
B=FH	FJ	Load the B-register's most significant 4 bits with those of the ALU output. The least significant 4 bits of the B register are left unchanged.
B=FL	FK	Load the B-register's least significant 4 bits with those of the ALU output. The most significant 4 bits of the B-register are left unchanged.
B=BRR	FI	Rotate the B-register right one bit.

These instructions all fall under the category of strobe functions. The last preceding ALU Literal command before a B=BRR instruction must be EBR. On the other hand, the last preceding ALU Literal before B=F, B=FH or B=FL must not be EBR. B=Ø will clear the accumulator in both cases.

2.6. RAM Address Instructions

Mnemonic	ASCII	Description
A=xxx	$\alpha\beta$	Load A register with address xxx (xxx is the 1Ø bit RAM address and may take on the values Ø - 1777_8). Depending on the address chosen, α will be one of the following ASCII characters: (P, Q, R, S, T, U, V, W, X, Y, Z, [, \ ,] , ↑ , or ←) and β will be one of the 64 ASCII characters listed in Appendix 1. Bits 1Ø and 11 of the A-register are left unchanged.
AL=D	FE	The data bus is loaded into the 8 least significant bits of the A-register. Bits 8, 9, 1Ø and 11 of the A-register are left unchanged.
AH=D	FF	Bits 8, 9, 1Ø & 11 of the A-register are loaded with bits

		Ø, 1, 2 & 3 of the data bus respectively.
A=A+1	F:	Increment the contents of the A-register.

2.7 Store Into RAM Instruction

Mnemonic	ASCII	Description
M=D	FL	Load data bus into RAM. The address in RAM must be previously defined by a RAM address instruction. The data memory is normally in the read mode and the M = D instruction, which is a strobe function, sends a write pulse to the RAM for the duration of the machine cycle.

2.8. Conditional Branch Instructions

This group comprises the instructions that direct the program to a nonsequential address for execution of the instruction located there. As previously shown in section 2.Ø.1, bits Ø to 6 of the ROM data define which of the possible 27 decision flags will define the logical condition for execution of the jump. The jump address is contained in the next word from instruction memory. Each of the following branch instructions comes in complementary pairs, i.e. jumping when the decision flag is either set or cleared.

Mnemonic	Mnemonic for Complement	ASCII	ASCII for Complement	Description
JIS,BR7	JIC,BR7	BC	@C	Jump if B-register bit 7 is Set/Clear.
JIS,CRY	JIC,CRY	BD	@D	Jump if ALU carry output flag is Set/Clear. This flag is a "1" when there is an overflow or underflow during an ALU addition or subtraction respectively.
JIS,D=B	JIC,D=B	BE	@E	Jump if data bus and B-register are equal/not equal. ALU must be in F=D-B-1 mode. (see pg.16).
JIS,DBØ	JIC,DBØ	BH	@H	
JIS,DB1	JIC,DB1	BI	@I	
JIS,DB2	JIC,DB2	BJ	@J	Jump if bit n of the
JIS,DB3	JIC,DB3	BK	@K	data bus is set/clear
JIS,DB4	JIC,DB4	BL	@L	where n = Ø to 7
JIS,DB5	JIC,DB5	BM	@M	
JIS,DB6	JIC,DB6	BN	@N	
JIS,DB7	JIC,DB7	BO	@O	

Mnemonic	Mnemonic for Complement	ASCII	ASCII for Complement	Description
JIS,PDS	JIC,PDS	BW	@W	Jump if push down stack overflow flag is Set/ Clear. This flag is set when the 16 levels of push down stack are either overflowed or underflowed. It is automatically cleared when the microprocessor is in the PORC condition. Thus, a reset instruction will clear it. (see pg. 22).

The remaining conditional branch instructions are described in the section on Input/Output.

2.9. Set Page Instructions

As shown in Section 2.Ø.2, the least significant 11-bits of a ROM address instruction enables the direct addressing of 2Ø48 memory locations. For addressing all the 4Ø96 locations, the following instructions should be used.

Mnemonic	ASCII	Description
PG=Ø	DT	The most significant bit (bit 11) of the parallel input to the P-register is set to Ø. This means that the next valid decision to take place will cause a jump to page Ø of the instruction memory. The page is always set to Ø automatically after power goes on.
PG=1	FT	Bit 11 of the P-register parallel input is set to 1 so that the next branch will be to the upper page of instruction memory.

It should be noted that a PG=Ø or PG=1 instruction does not change the page at the time of the command, but rather defines the page to be jumped to at the next branch.

2.1Ø. Unconditional Jump Instruction

Mnemonic	ASCII	Description
JMP	B@	This instruction causes an unconditional jump to the address defined by the ROM address pointer following it and to the page number last set by the set page instruction.

2.11. Jump to Subroutine Instruction

Mnemonic	ASCII	Description
JSR	FØ	This loads an address two greater than that in the P register into the push down stack. After this return address is stored, the stack is pushed down ready to accept another return address. The next instruction from ROM will be treated as an unconditional branch address to which the P register will be set. This instruction is a strobe branch function.

2.12. Return from Subroutine Instruction

Mnemonic	ASCII	Description
RET	FN	This instruction causes the push down stack to "push up", thus revealing the last return address stored into it. This return address is then enabled onto the P register parallel input. In addition to the next instruction from ROM being executed, the return address will be loaded into the P register causing a return jump to the subroutine calling program.

2.13. NOP and HALT Instructions

Mnemonic	ASCII	Description
NOP	@@	No operation is performed by this instruction except that a one machine cycle delay of 24Ø n. seconds results.
HLT=Ø	F(space)	If a maintenance and control chassis is interfaced to the microprocessor, any one of the 16 HALT instructions will stop the processor master clock. In the case where no maintenance and control chassis is connected to the microprocessor all HALT instructions are ignored and treated as NOPs.
HLT=1	F !	
HLT=2	F"	
HLT=3	F#	
HLT=4	F$	
HLT=5	F%	
HLT=6	F&	
HLT=7	F'	
HLT=1Ø	F(	
HLT=11	F)	
HLT=12	F*	
HLT=13	F+	
HLT=14	F,	
HLT=15	F-	
HLT=16	F.	
HLT=17	F/	

2.14. PORC and the RST Instruction

The PORC or power on reset circuit is used to provide the required hardware initialization when first turning the microprocessor on. When power is first turned on, PORC condition exists within the microprocessor for approximately 1ØØ milli seconds. During this condition the following is done.

a) The master clock oscillator is disabled from the microprocessor.

b) The P and A registers are cleared (set to address Ø).

c) A power reset pulse is sent to all peripherals both on the serial and the parallel I/O bus.

d) The instruction memory page function is set to Ø.

e) The CLK, LD, R/W and RBC functions are all cleared. These functions will be explained in the section on Input/Output.

f) The interrupt acknowledge function is set to Ø.

g) The push down stack is set to subroutine level Ø and the overflow PDS flag is cleared.

h) The master interrupt flag is disabled.

The PORC condition may also be initiated in two other ways. The first way is to press the reset button on the maintenance and control chassis, assuming it is connected to the microprocessor. The second way is to execute a reset instruction.

Mnemonic	ASCII	Description
RST	F8	Strobe the microprocessor into the PORC condition.

2.15. Real Time Clock and the RTC Instruction

The real time clock provides the setting of the decision flag at a crystal-controlled timing rate. The timing is derived from the microprocessor internal master clock which is divided down by some integral number as determined by optional strapping. This clock frequency may be strapped into the master interrupt circuit (see sec. 3.1.) for use in interrupt mode, or may be used as a decision flag to be detected under program control. Although only one decision flag (RTC) is mentioned, more are available as an option.

Mnemonic	Mnemonic for Complement Instruction	ASCII	ASCII for Complement Instruction	Description
JIS,RTC	JIC,RTC	BF	@F	Jump if real time clock is set/clear.

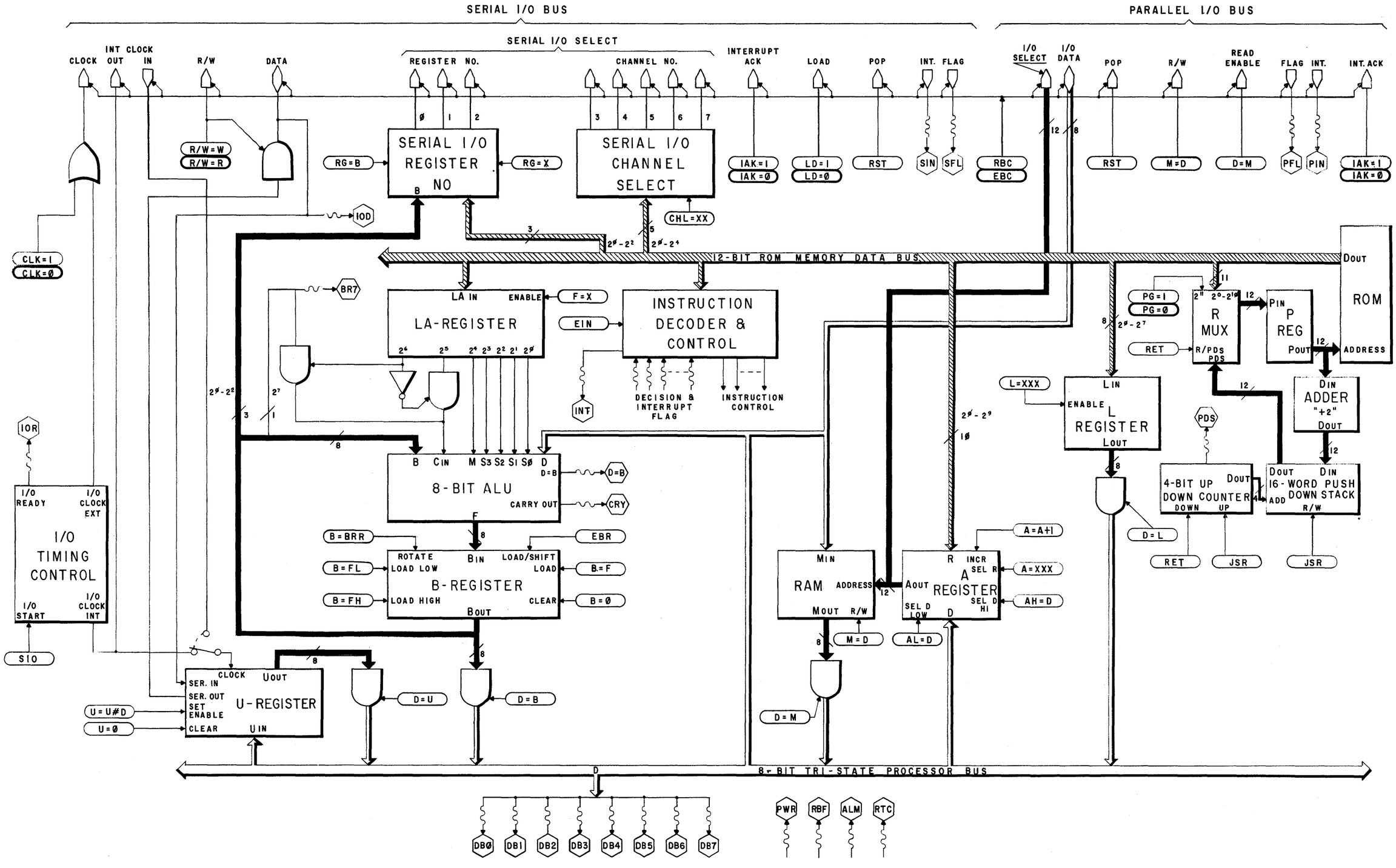


FIG. 2.1 — FUNCTIONAL PROGRAMMING BLOCK DIAGRAM FOR MICROPROCESSOR

III. Input/Output

The AES Input/Output system provides a powerful and flexible interface between the microprocessor and the peripheral devices that are connect to it.

The I/O system may be subdivided into two principal parts namely (a) serial I/O and (b) parallel I/O. A typical I/O system configuration is shown in Fig.3-1.

3.Ø. Serial I/O

3.Ø.1. I/O Bus Structure

The I/O bus that interconnects the microprocessor to the external device has the following structural elements:

a) Address: 8 lines defining an 8 bit I/O select code for addressing up to 256 8 bit registers. 5 lines are used for defining one of 32 channels and the remaining 3 are used for selecting one of 8 registers within the channel.

b) Control: 5 lines called LOAD, R/W, CLOCK, IAK and POP. The first three signals control the transfer of data to or from the external device and the fourth signal acknowledge an interrupt request. The POP signal is the power on pulse generated during a PORC condition.

c) Flags: A status flag corresponding to the I/O select code may be read on one line of the I/O bus. A serial I/O interrupt flag line is also available. This is the logical "OR" of all the interrupt flags within the system.

d) Data: One bi-directional line is available on the serial I/O bus for transferring 8-bit serial characters to or from the microprocessor.

e) Propagation Delay: Two additional clock lines exist within the serial bus. This permits the reading of serial data into the microprocessor to be independent of the physical length of the bus. The clock which is used to shift I/O data through the U-register is "transmitted" onto the I/O bus from the microprocessor and "reflected" from the addressed peripheral. The return clock line is then used so shift serial data into the U-register.

Drivers and receivers are contained in the microprocessor and in the external peripherals for signalling over the serial I/O bus. Three options concerning the types of line drivers or receivers are available to the user.

Type 1: For simple systems (maximum of 2 I/O channels, proximity of less than 5 ft.), all unidirectional bus lines use standard TTL gates as both drivers and receivers and bi-directional lines use TRI-state* TTL logic gates as drivers and standard TTL gates as receivers. For this system configuration no serial Input/Output control card is necessary.

Type 2: For medium sized systems (maximum of 5 I/O channels, proximity of less than 2Ø ft.) all bus lines use TTL TRI-state gates for both drivers and receivers. Each signal line is twisted with a ground line to form a twisted pair.

Type 3: For large systems, using more I/O than types 1 and 2, all bus lines make use of "Party Line" differential line drivers and receivers. These drivers are able to drive up to 1ØØØ ft. of bus line at standard TTL speeds.

3.Ø.2 Modular System Unit Bus Interface

The modular system unit (MSU) bus interface card is designed to interface up to 8 I/O peripheral interface cards to the microprocessor serial I/O bus (see Fig.3.2). This card is intended for use in systems of type 3 and its primary function is to interface the differential driver and receiver signals to standard TTL logic levels. This card also decodes the eight address select lines to form 4 group,8 channel and 8 register linear select signals for enabling the I/O cards within the modular system unit.

The I/O devices within the MSU provide an I/O selected signal which indicates if one of them has been addressed. When this signal is high, the I/O flag status is enabled onto the I/O bus. When the I/O selected line is high and, in addition, the R/W line is in the read condition, both the data line and microprocessor I/O clock are enabled onto the I/O bus.

3.Ø.3 Typical I/O Board

A typical MSU module contains 9 I/O cards which communicate with an AES microprocessor through an MSU bus interface card. A typical example of the type of I/O cards is an 8 IN/8 OUT card. The block diagram of such a card is shown in Fig.3-3. The following is a description of the functions of the various block interconnections.

The 8 IN/8 OUT provides the following capabilities:

1) Reading status of 8 external points and inputting the information to the microprocessor.

---

* National Semiconductor Corp. TM.

2) Outputting drives (sinks) to 8 external points

The input and output of this information is under complete microprocessor program control.

The block labelled I/O INTERFACE LOGIC provides the media of interpreting microprogrammed sequences as well as decoding the destination address of these sequences. The address decode is a strappable option. Consequently, if a string of commands contains the address of the 8 IN/8 OUT board then READ/WRITE action is initiated depending on the request . For example, if the request is to read status from 8 external points then the 8 points in question are strobed into the 8 BIT SHIFT REGISTER and a second sequence would automatically shift the data onto the serial I/O bus and into the microprocessor. Similarly the outputting of data follows the same pattern.

Several other features are provided for by the card. Consider first the 8 input lines. The status input lines are isolated from the logic with photocoupled isolators. This separates the logic from any power or ground line noise induced on the status lines. Furthermore, each line has an input filter which can protect against induced electrical transients (up to 3.5kV). A third feature is provided in the signal conditioning circuitry. This basically prevents any slowly changing status signals from causing the logic to operate in its linear region. Linear region operation would generate oscillations in the logic.

The output lines are also isolated from the logic and are protected against electrical transients. An important feature for the outputs is the capability of providing either a source of power for the 8 external points or of sinking current from these same 8 sources. This source/sink capability is an option on the card.

3.Ø.4 Serial I/O Interrupt

All I/O interface cards within a Modular System Unit having an interrupt flag, output this signal which is logically "OR"ed within the MSU to form the interrupt flag line. The interrupt flag lines from all Modular System Units are, similarly, logically "OR"ed to provide the serial I/O interrupt flag to the microprocessor.

This flag may be used to generate a master interrupt within the microprocessor.

3.1 Interrupt Structure

Eight decision flag bits are provided to designate a particular interrupt condition. When any of these bits are high, the master interrupt flag goes high. The master interrupt flag bit may be tested by the microprogram to detect the interrupt condition. When the microprocessor has recognized the interrupt request, it may respond accordingly.

The normal procedure for acknowledging an interrupt is to regularly monitor the master interrupt flag. This can be done by executing a decision instruction on the interrupt flag whenever returning from a subroutine. One must assume, of course, that the maximum allowable interrupt response time is not exceeded by the maximum subroutine execution time.

Normally, the interrupt flag is low and only 2 machine cycle times (48Ø n seconds) are wasted looking at the flag. If, however, the flag is high, the microprocessor can proceed to execute an interrupt servicing routine which checks the 8 interrupt flags in order of their priority.

The master interrupt flag may be disabled or enabled by using the appropriate instructions.

Once the flag responsible for generating the interrupt is found, it may be serviced. The interrupting flag should, however, be cleared prior to servicing the interrupt so that higher level interrupts may be monitored while servicing the lower level one.

Decision flags may be assigned to be interrupt status flags by strapping them into the master interrupt circuit. Flags normally assigned as interrupts are:

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

3.2 Parallel I/O

As explained in section 1.1Ø, the data memory and parallel I/O peripherals share the same high-speed bus. Thus the microprocessor views parallel I/O peripheral devices as active memory locations which perform special functions. There are some differences, however, between the way the data memory and the parallel I/O devices are operated upon by the microprocessor.

It may be seen in Fig. 2.2, there are 3 control and 2 flag lines on the parallel I/O bus that are not used for accessing data memory. These are:

a) POP: Power on pulse generated during a PORC condition. It is used for resetting all I/O devices.

b) PFL: I/O status flag usually indicates that I/O device being addressed is ready.

c) PIN: I/O interrupt flag used to indicate that at least one I/O device is interrupting.

d) IAK: Interrupt acknowledge signal used for clearing the interrupt flag of the I/O device being addressed.

e) RBC: This line does not go onto the I/O bus, but rather, is used to disable the microprocessor from both the serial and parallel I/O busses.

In some cases the parallel I/O status flag may be used to delay the master clock until the device is ready, as explained in section 1.9.

3.3 Multiprocessor Configuration

A capability exists whereby the AES microprocessor can disable itself from both the serial and parallel I/O busses. When a relinquish bus flag is sensed during an interrupt acknowledge routine, it is possible to disable all I/O line drivers and receivers from the I/O busses by executing a relinquish bus control instruction. This command is useful when there are 2 microprocessors on the same bus and where one of them is waiting on standby to take over bus control in the case of microprocessor failure.

3.4 I/O Instructions

The I/O instruction group is used for all communication between the computer and the peripheral devices that supply and receive data.

The RAM address instructions and the M=D and D=M instructions are also used for selecting, Read/Write control and read enabling of parallel I/O devices. As these have already been explained in sections 2.2, 2.6 and 2.7, they will be omitted from the following description.

3.4.1 U-Register Instructions

Mnemonicm	ASCII	Description
U=Ø	FG	Clear the U-Register (serial I/O register).
U=U#D	FD	Form the logical "OR" function of the D bus and U-register words and store this into the U-register.

3.4.2 I/O Register Select Instruction

Mnemonic	ASCII	Description
RG=Ø	FX	Set the least significant 3 bits of the serial I/O select address to the octal number n where n = Ø to 7. This defines the serial I/O register number.
RG=1	FY	
RG=2	FZ	
RG=3	F[	
RG=4	F\	
RG=5	F]	
RG=6	F↑	
RG=7	F←	
RG=B	FM	Set the least significant 3 bits at the serial I/O select address equal to the least significant 3 bits of the B-register output.

These instructions form a mutually exclusive function set.

3.4.3. I/O Channel Select Instruction

The following instructions set the most significant 5 bits of the serial I/O select address to the number N, where N = Ø to 31. This defines the serial I/O channel number.

Mnemonic	ASCII	Mnemonic	ASCII
CHL=Ø	G@	CHL=16	GP
CHL=1	GA	CHL=17	GQ
CHL=2	GB	CHL=18	GR
CHL=3	GC	CHL=19	GS
CHL=4	GD	CHL=2Ø	GT
CHL=5	GE	CHL=21	GU
CHL=6	GF	CHL=22	GV
CHL=7	GG	CHL=23	GW
CHL=8	GH	CHL=24	GX
CHL=9	GI	CHL=25	GY
CHL=1Ø	GJ	CHL=26	GZ
CHL=11	GK	CHL=27	G[
CHL=12	GL	CHL=28	G\
CHL=13	GM	CHL=29	G]
CHL=14	GN	CHL=3Ø	G↑
CHL=15	GO	CHL=31	G←

3.4.4 Serial I/O CLOCK, LOAD and R/W Instructions

Mnemonic	ASCII	Description
CLK=Ø	DU	Clear the I/O clock line
CLK=1	FU	Set the I/O clock line
LD=Ø	DV	Clear the I/O load line
LD=1	FV	Set the I/O load line
R/W=R	DW	Set the I/O R/W line to Ø (read)
R/W=W	FW	Set the I/O R/W line to 1 (write)

These instructions are all latching functions.

3.4.5 Relinquish Bus Control Instructions

Mnemonic	ASCII	Description
RBC	FP	Disable all serial and parallel I/O line drivers and receivers
EBC	DP	Enable all serial and parallel I/O line drivers and receivers. These are automatically enabled during a PORC condition.
JIS,RBF	BP	Jump if relinquish bus flag is set.
JIC,RBF	@P	Jump if relinquish bus flag is clear.

3.4.6 Serial I/O Timing Instructions

Mnemonic	ASCII	Description
SIO	FO	Start the automatic transfer of serial data between the U-register and the addressed serial I/O device. Data will be transferred into or out of the U-register depending on the status of the R/W line. This is a strobe function.

Mnemonic	ASCII	Description
JIS,IOR JIC,IOR	BB @B	Jump if the I/O ready flag is set/clear. This flag is normally high. As soon as the SIO instruction is executed, the IOR flag goes low and remains there until the transfer of data is completed.

3.4.7 Interrupt Instructions

Mnemonic	ASCII	Description
JIS,SIN JIC,SIN	BQ @Q	Jump if the serial interrupt flag is set/clear.
JIS,PIN JIC,PIN	BS @S	Jump if the parallel interrupt flag is set/clear.
DIN	DQ	Disable the master interrupt
EIN	FQ	Enable the master interrupt
JIS,INT JIC,INT	BU @U	Jump if the master interrupt is set/clear.
IAK=1 IAK=Ø	FR DR	Interrupt acknowledge. This latching function is used for setting or clearing both serial and parallel interrupt and/or status flags. In order to clear a serial interrupt or status flag, the interrupting I/O device must be addressed and the R/W line equal to write before the IAK instructions are executed. For acknowledging a parallel status or interrupt, read must be enabled (D=M) and the proper I/O device addressed before the IAK pulse is sent. An IAK pulse is generated by following an IAK=1 instruction by IAK=Ø.

3.4.8 Other I/O Instructions

Mnemonic	ASCII	Description
JIS,IOD JIC,IOD	BG @G	Jump if serial I/O bus data line is set/clear.
JIS,SFL JIC,SFL	BR @R	Jump if serial I/O status flag is set/clear.
JIS,PFL JIC,PFL	BT @T	Jump if parallel I/O status flag is set/clear.
JIS,ALM JIC,ALM	BA @A	Jump if external alarm is set/clear. This alarm flag decision line is not part of the serial or I/O bus but is reserved for I/O independent purposes such as operator interrupts or console alarms.
JIS,PWR JIC,PWR	BV @V	Jump if the power fail interrupt flag is set/clear. This flag should be connected to the "Power supply on" signal available in some power supplies. This signal is the result of comparing the preregulator voltage with a reference and will predetect a supply drop before at occurs. This flag remains low until after the supply voltage is set and goes low before the supply voltage drops.

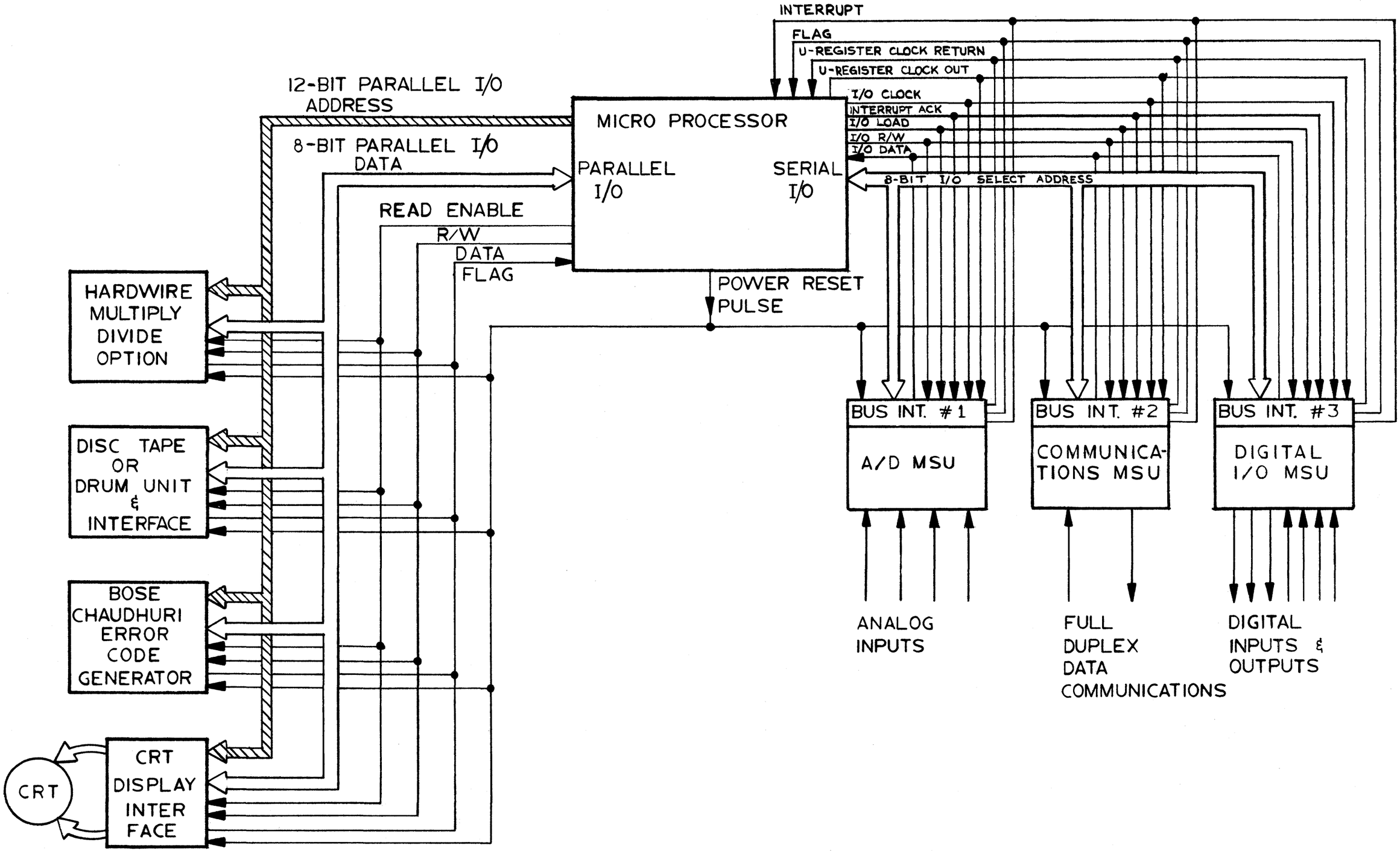


FIG. 3-1: TYPICAL I/O SYSTEM CONFIGURATION

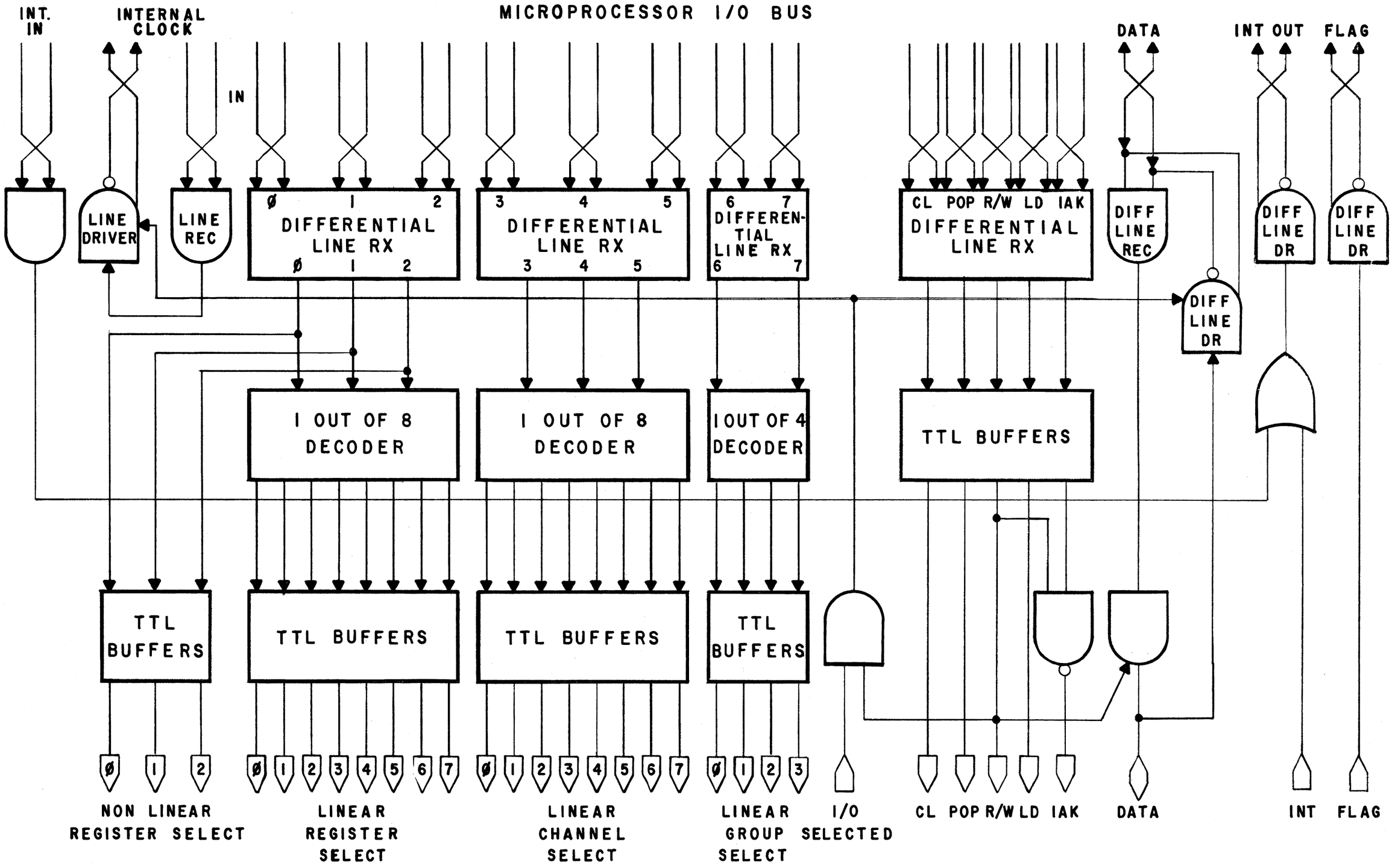


FIG. 3-2 — MSU BUS INTERFACE

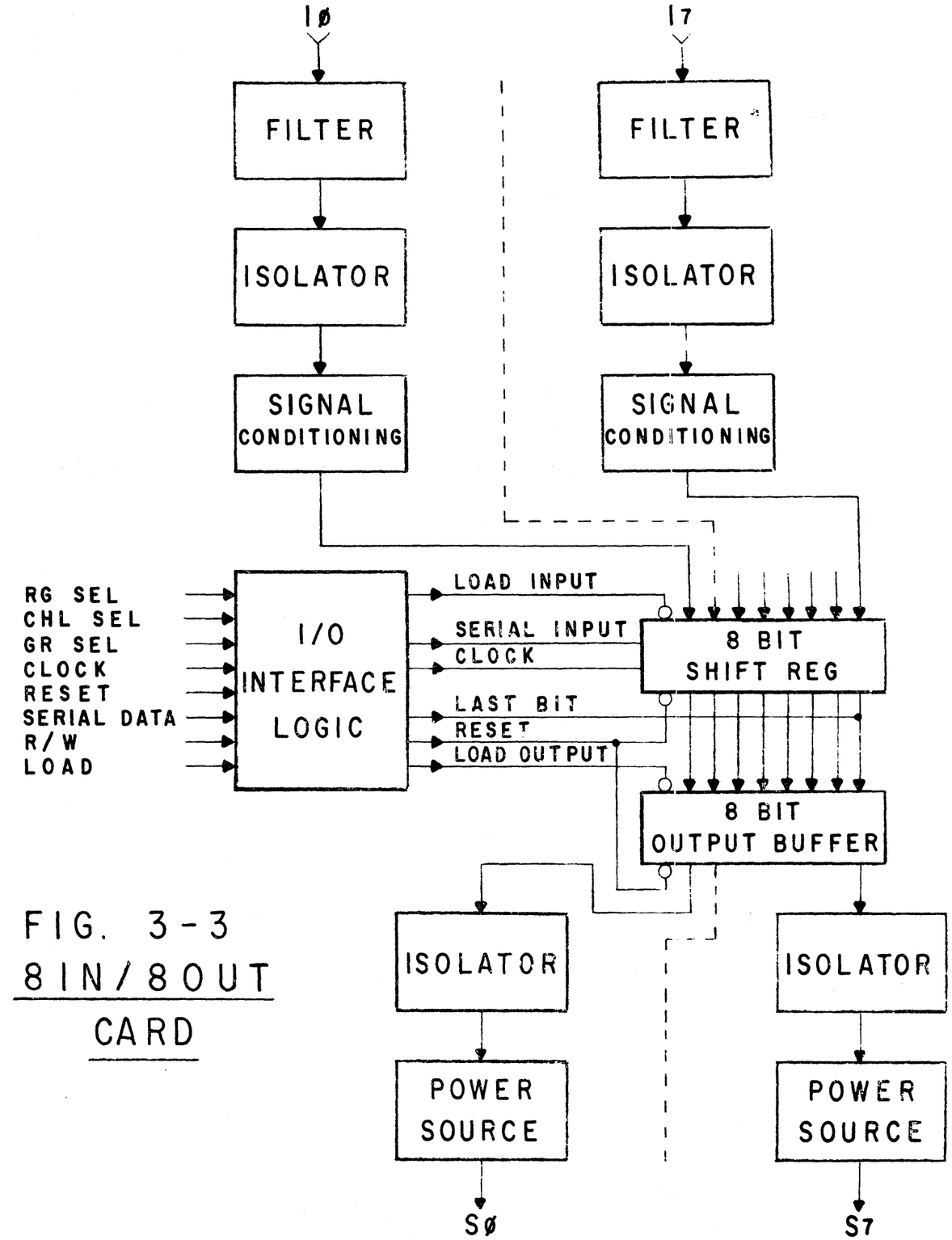


FIG. 3-3
8 IN/8 OUT
CARD

IV. Timing

The AES microprocessor operates on a basic 24Ø nanosecond machine cycle. That is, a full execution cycle (read instruction from memory and execute instruction) is performed in each 24Ø n. second time interval (except in some special cases in which the period is extended: these cases were discussed in section 1.9). The timing diagrams presented in the following sections each correspond to a small program listed at the bottom of the diagram.

4.Ø Function and Literal Timing

The program shown on figure 4.1 consists of a series of function and literal instructions. These are listed in an order which will demonstrate the strobing and latching properties of the command.

4.1 Branch Instruction Timing

The program shown in figure 4.2 consists of a series of unconditional and conditional jump instructions. The F = D - B-1 and D = B commands are used in order to set the D = B flag high.

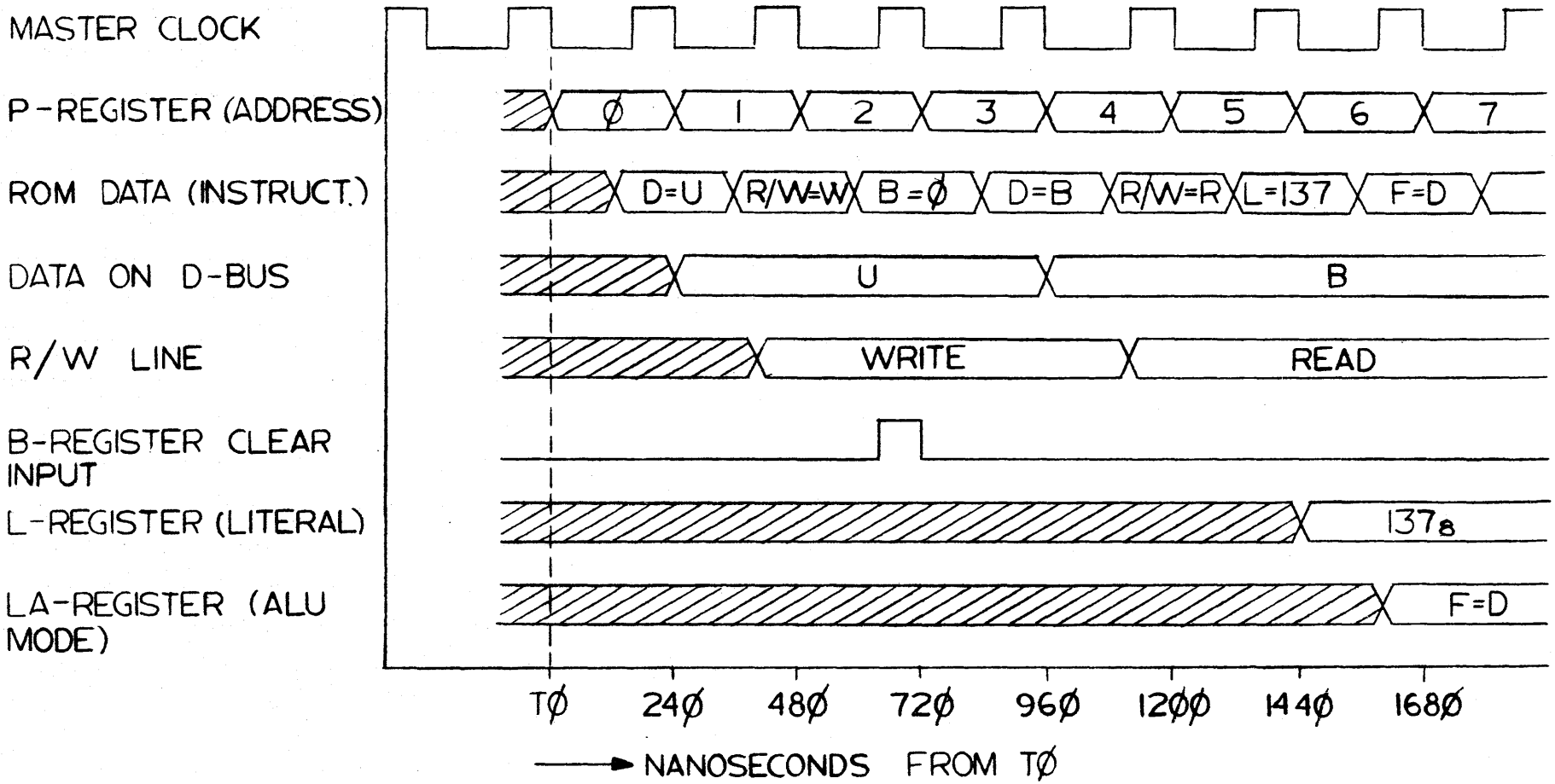
4.2. Subroutine Timing

The program shown in figure 4.3 illustrates the execution of a routine having 2 levels of subroutine. Both jumps and returns from subroutine are demonstrated.

4.3. I/O Timing

Two I/O routines are illustrated. The first, figure 4.4, reads an 8-bit character into the microprocessor from a serial I/O device. It is assumed that the character being read has already been loaded into the external device transfer register. The serial I/O is first set into the read mode and the channel and register are chosen. The reading is started by a start I/O instruction and the character is automatically shifted into the U-register. The microprocessor then runs in a small program loop waiting for the I/O ready flag. As soon as this flag is set, the contents of the U-register is enabled onto the data bus where it may be used for program requirements.

The second I/O routine (figure 4.5) describes the writing of an 8-bit character into a particular register of a channel and the transferring of it into the output buffer (see figure 3.3). It should be noted that for writing onto an I/O device, the output I/O clock precedes the U-register clock rather than following it, as is the case for reading a character.



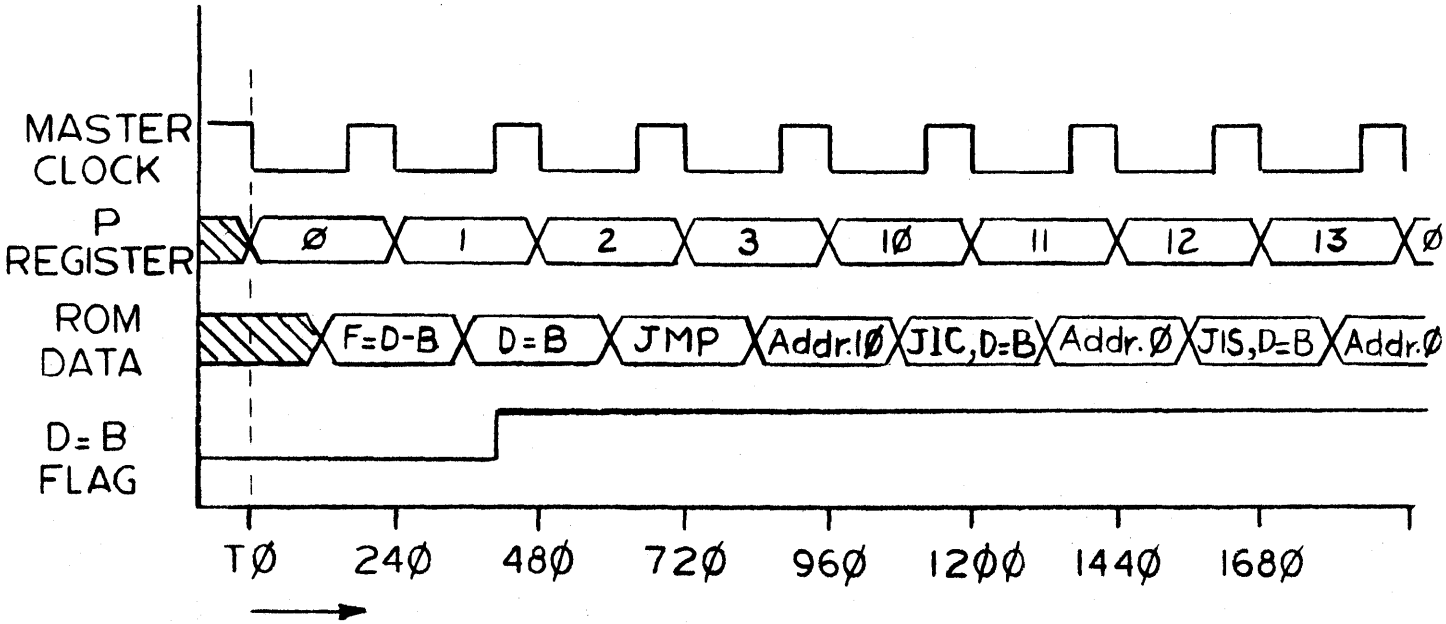
PREVIOUSLY DEFINED

INSTRUCTION TYPE	PROGRAM ADDRESS	PROGRAM INSTUCTION
MUTUALLY EXCLUSIVE FN.	Ø	D=U
LATCHING FN.	1	R/W=W
STROBE FN.	2	B=Ø
M.E. FN.	3	D=B
LATCHING FN.	4	R/W=R
DATA LITERAL	5	L=137
ALU LITERAL	6	F=D

FIG 4.1 FUNCTION TIMING DIAGRAM

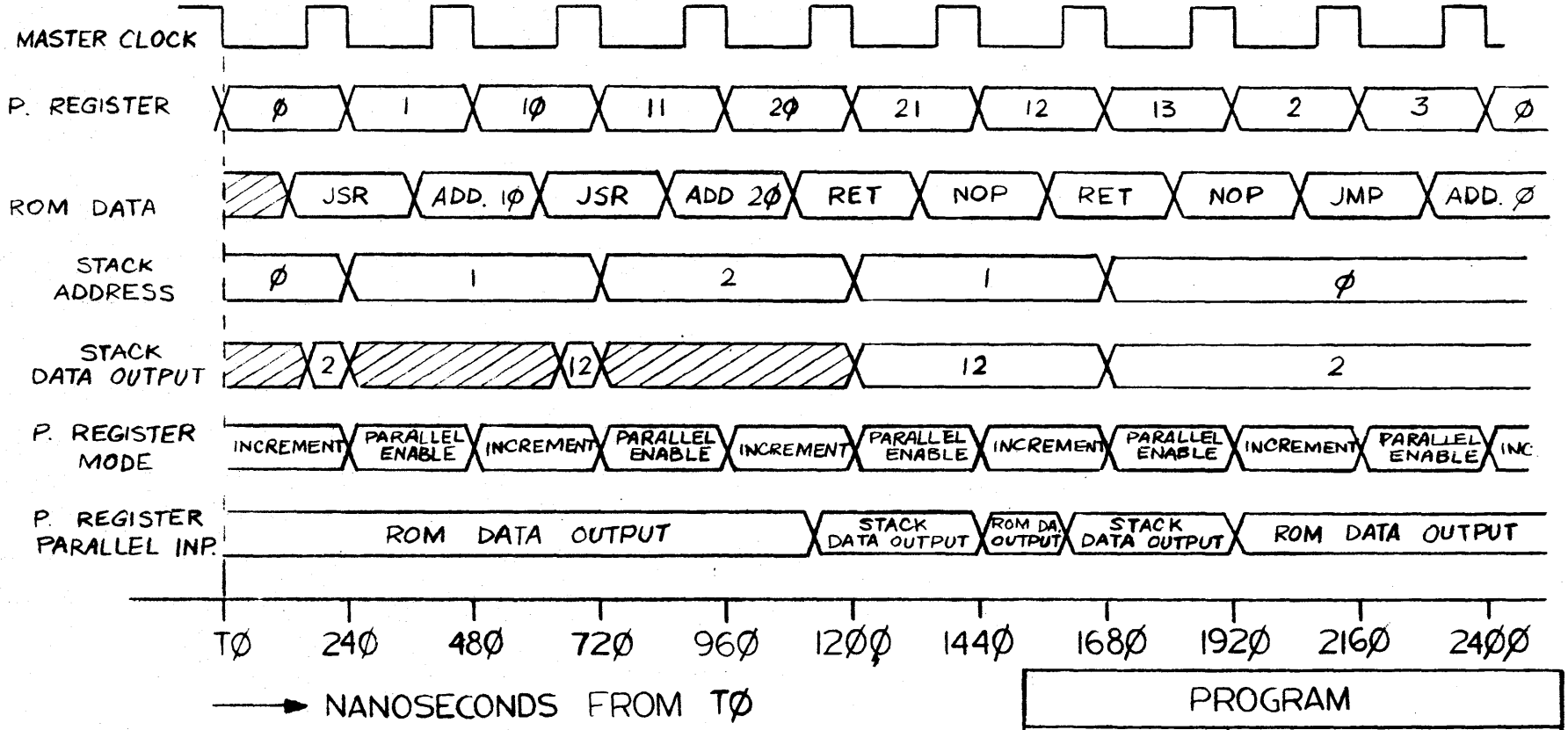
FIG. 4-2

BRANCH INSTRUCTION TIMING



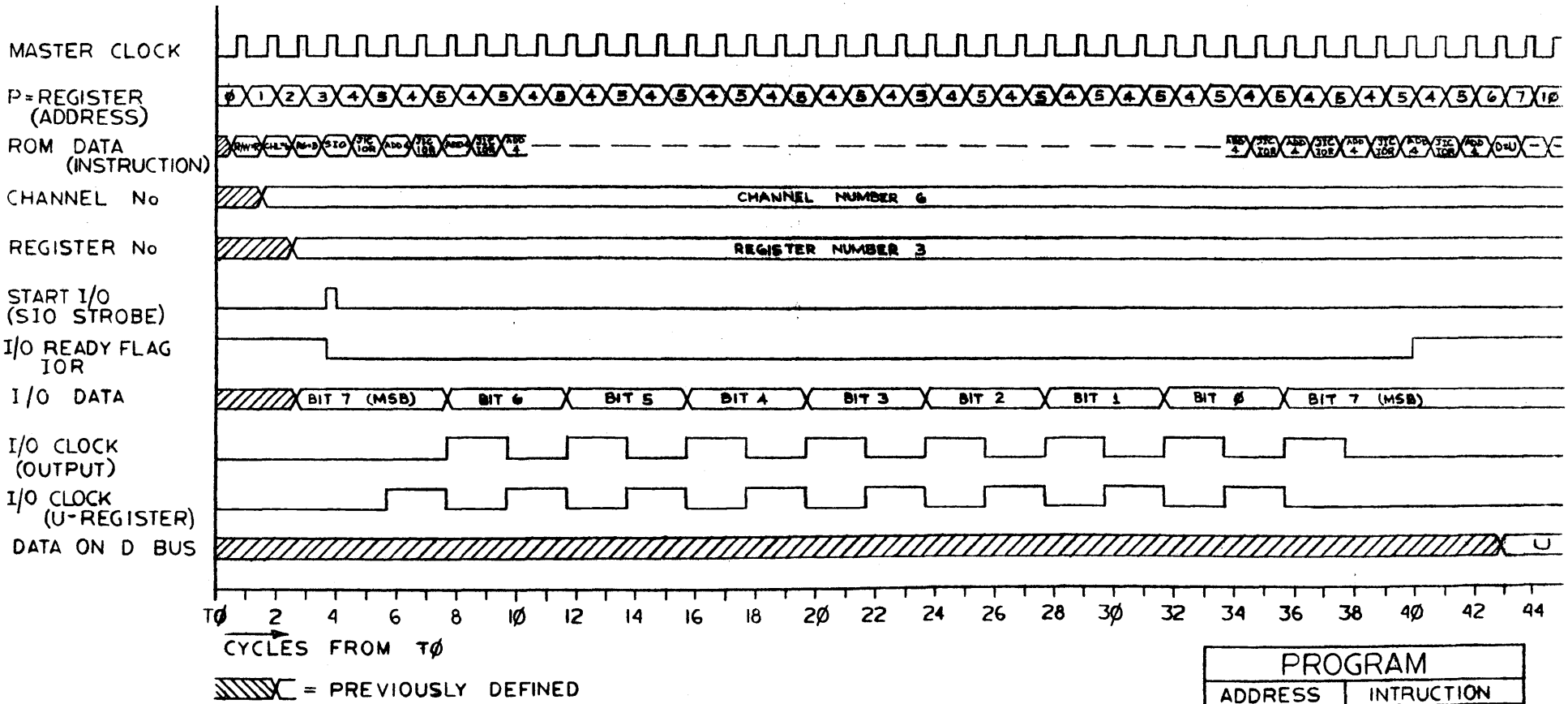
= previously defined

PROGRAM	
ADDRESS	INSTRUCTION
Ø	F=D-B
1	D=B
2	JMP
3	ADDRESS 1Ø
⋮	⋮
1Ø	JIC, D=B
11	ADDRESS Ø
12	JIS, D=B
13	ADDRESS Ø
⋮	⋮



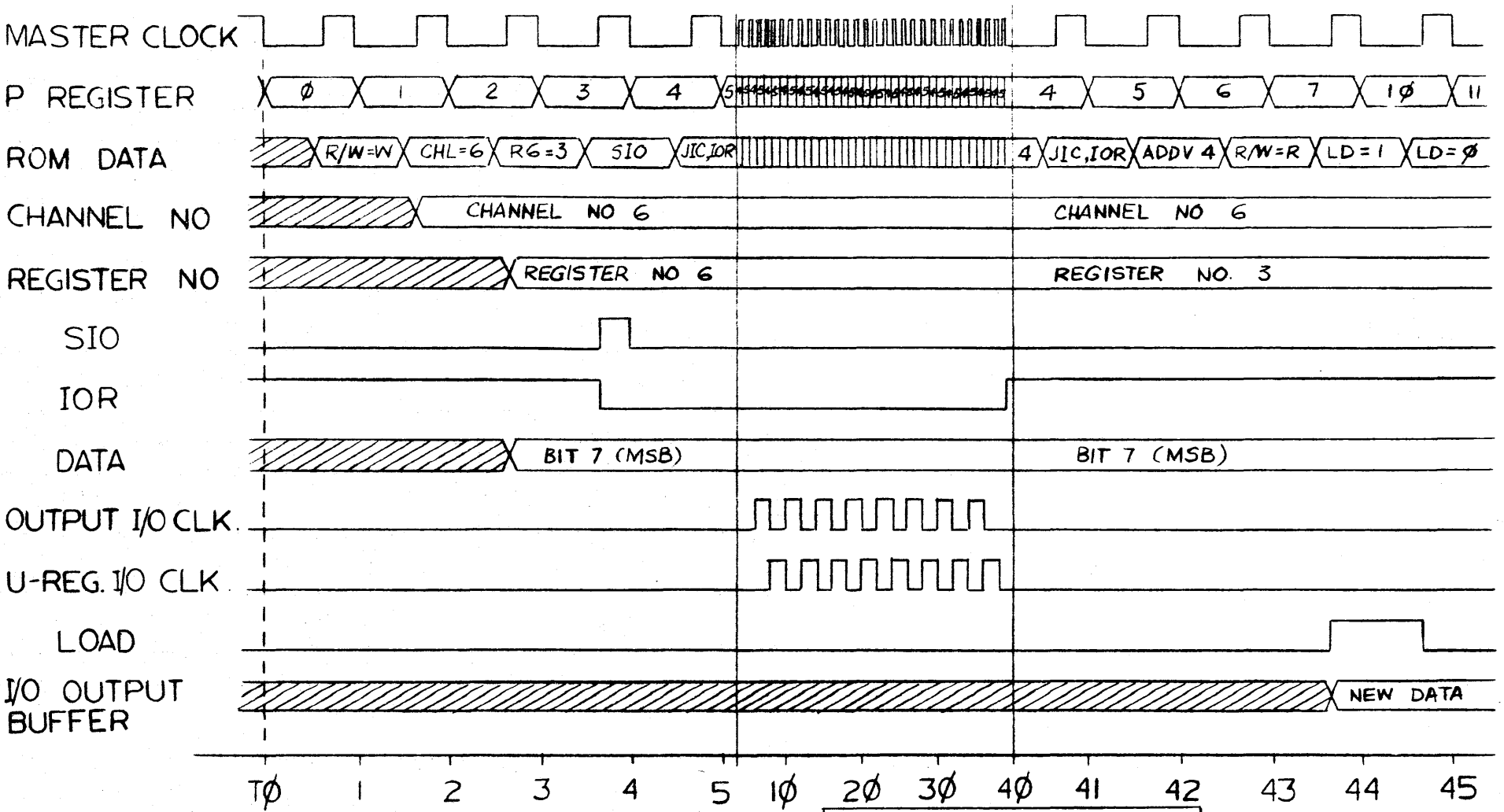
PROGRAM	
ADDRESS	INSTRUCTION
Ø	JSR
1	ADD. 1Ø
2	JMP
3	ADD. Ø
1Ø	JSR
11	ADD. 2Ø
12	RET
13	NOP
2Ø	RET
21	NOP

FIG. 4.3 SUBROUTINE TIMING DIAGRAM



PROGRAM	
ADDRESS	INTRUCTION
Ø	R/W = R
1	CHL = 6
2	RG = 3
3	SIO
4	JIC, IOR
5	ADDRESS 4
6	D = U

FIG. 4.4 READING A CHARACTER



PROGRAM	
ADDRESS	ISTRUCTION
Ø	R/W=W
1	CHL=6
2	RG=3
3	SIO
4	JIC IOR
5	ADD. 4
6	R/W=R
7	LD=1
1Ø	LD=Ø

FIG. 4.5 WRITING A CHARACTER

V. Program Development and Control Console (PDCC)

A PDCC the front panel of which is shown on figure 5-1 is available for programming testing and maintenance of the AES microprocessor. When the console is connected to a microprocessor, the operator is capable of executing the microprogram in a step-by-step fashion. It is not however required for normal operation of the system.

There are three fundamental ways in which the PDCC can be interfaced with an AES microprocessor. The first is used when monitoring a microprocessor that is part of a larger system. In this case, the PDCC is used primarily as a passive display and the only active control that the console has over the microprocessor is the ability to halt, reset, single step and set a command address. The microprogram is limited to that stored in the microprocessor ROM and there is no way of altering an instruction. This configuration requires a PDCC interface card for interfacing the microprocessor unit to the PDCC.

The second configuration is used for checking a microprocessor without the instruction ROM. In addition to the PDCC interface card, a ROM Interface card is necessary. This card is inserted into the Read Only Memory Slot and is also connected to the ROM simulator which is part of the PDCC. The ROM simulator is a high speed random access bipolar memory which behaves like a ROM as far as the microprocessor is concerned. Its contents, however, are alterable by loading instructions into it via the maintenance and control chassis switch register, an ASR 33 teletype or a punched tape reader.

The third configuration is one in which the microprocessor cards themselves are inserted into the PDCC. This combined console provides an AES microprocessor package which may be used in developing the software necessary for the final dedicated application.

The Program Development and Control Console contains the following elements:

1. Four octal displays, each having four digits, for displaying ROM DATA, ROM ADDRESS, RAM ADDRESS and the switch register contents, respectively.

2. One 2-digit octal display for subroutine level indication.

3. An 8-bit data display.

4. A bank of 26 indicators, indicating the following conditions:
   RUN, HALT AND PORC

functions:

RELINQUISH BUS CONTROL, INTERRUPT ACKNOWLEDGE, START I/O and MASTER INTERRUPT ENABLE

and decision and interrupt flags:

I/O Ready, I/O Data, D=B, ALU Carry Output, ALU Carry Input, Relinquish Bus Flag, Serial Interrupt, Serial Status Flag, Parallel Interrupt, Parallel Status Flag, Master Interrupt, Power Fail, Real Time Clock, I/O Clock, U-register Clock, I/O Load, I/O R/W, and Alarm and Push Down Stack Flag.

5. Data Display Select Switches. These seven interlocked switches select the register or bus to be displayed on the data display indicators. The displays which can be selected are: the D-bus, U-register, B-register, Serial I/O address select, L-register, LA-register and the 8-bits of Data Memory output.

6. A numerical keyboard numbered from Ø to 7 with a CLEAR pushbutton. These nine keys constitute the switch register keyboard and the numbers are serially shifted into the switch register octal display as keyed. The CLEAR pushbutton clears the switch register of all numbers keyed.

7. RUN Switch. This switch places the microprocessor in the run mode, thus causing it to execute instructions.

8. HALT Switch. This switch stops the microprocessor at the end of the current machine cycle.

9. STEP Switch. This switch executes one machine cycle each time it is pressed.

1Ø. RESET Switch. This switch places the microprocessor into the PORC condition. When the switch is released, the microprocessor remains in a halt condition.

11. LOAD ADDRESS Switch. This switch transfers the switch register contents into the P-register, thereby setting up the address of the next instruction to be performed.

12. LOAD MEMORY Switch. This switch stores the contents of the switch register into the instruction memory location specified by the current value of the P-register. The P-register is automatically incremented after operation of the Load Memory Switch, to simplify storing data into consecutive locations. In the case where the instruction memory is ROM, depressing this switch merely increments the P-register.

13. DISPLAY MEMORY Switch. This switch causes the ROM data octal numbers on the PDCC to display the contents of the location specified by the P-register. The P-register is automatically incremented after operation of the Display Memory Switch.

14. LOAD PROGRAM Switch. Depressing this switch automatically loads a binary tape into the ROM Simulator.

15. LIST PROGRAM Switch. Depressing this switch automatically lists a program stored in the ROM Simulator. The listing will be in the ASCII format with a carriage return and line feed after each instruction, and will begin at the address specified by the P-register.

16. MAIN POWER Switch. This three position switch turns power (OFF), (ON) and disables all other control switches (LOCK).

17. A lamp test switch. This switch permits the testing of all indicator lights.

The following switches are located on the inside of the front panel.

18. An external clock connector used for providing an external source for the master clock oscillator.

19. An Internal/External switch for selecting the source of the master timing clock.

2Ø. TTY/READER Switch. If both the teletype and tape reader options are available, the position of this switch determines which of these two devices is used to load binary tapes.

21. PARITY/NO PARITY. Enables the confirmation of an even parity bit on all binary tapes loaded.

22. FWD/REV. Switch. Determines the direction of tape flow through the tape reader when loading a program.

23. ECHO/NO ECHO. Enables or disables automatic printing of tapes read in though TTY tape reader.

24. DISABLE/ENABLE BUS. Clears or sets the RBF flag.

AES - 80

AUTOMATIC ELECTRONIC SYSTEMS INC

PRESS TO OPEN

								RAM ADDRESS	ROM DATA								
								0234	3056								
PORC	ALM	PWR	PDS	RBC	RBF	RTC	D=B			PAR FLAG	PAR INTPT	ACK INTPT	MAST INTPT	ENABL INTPT	ALU CARRY IN	ALU CARRY OUT	U-REG CLOCK
DATA DISPLAY								SWITCH REGISTER	ROM ADDRESS								
								7260	5304								
7	6	5	4	3	2	1	Ø			START	READY	CLOCK	R/W	DATA	LOAD	FLAG	INTPT
SUB-ROUTINE LEVEL 07															OFF	ON	LOCK

LIST PROG	LOAD PROG	D BUS	U REG	B REG	I/O SEL	L REG	LA REG	RAM DATA	Ø	1	2	3	CLEAR	LOAD MEM	DISP MEM	LOAD ADD	STEP	RUN	HALT	RESET	ALM TEST	LAMP TEST
									4	5	6	7										

VI. Software Development

6.1 AES 8Ø-ASSEMBLER

The AES-8Ø Assembler allows programmers to write their micro instruction programs in a symbolic language.

The assembler input is a source program which can be on:

- paper tape,
- punched card,
- magnetic tape, or
- disc.

The output consist of:

- an object program punched on paper tape
- a source program listing which can be displayed on a C.R.T terminal or typed on an ASCII printing device (teletype, line printer, etc.).

The assembler is a 2 pass system or a 3 pass system if the punch and the printer are mechanically linked together as in the ASR-33 teletype.

The assembler accepts 1-9 instructions and micro instructions and 5 pseudo-instructions.

Instructions and pseudo-instruction operands can be either numeric (octal or decimal), symbolic (1 to 5 alpha-numeric characters) or a combination of both: symbolic plus or minus a numeric displacement.

The listing includes line-number, address (octal) instruction in both codes: 6-bit ASCII and octal, and the source statement.

Error count and instruction count are provided.

The AES-8Ø assembler is actually a cross-assembler and it exists in different versions:

1) Cross-assemblers written in popular mini computer assemblers. These we can run with most 4k computers and a teletype.

2) Cross-assembler written in FORTRAN II. Can run on any computer with a minimum of 8K of core and the proper combination of peripherals (at least one paper tape punching device).

3) Self-assembler to be used in the AES 8Ø Program Development and Control Console with an external memory of 4k X16.

The present version of this 2 pass cross-assembler includes the following features:

1. Symbolic Addressing:
   Examples:
   For ROM addresses..... JMP Label
   For RAM addresses..... A = Label
   For I/O Channels..... CHL = Label
   For Program Origin..... ORG Label

2. Symbolic Literals:
   Example:..... L = Label

3. Octal and Decimal Constants:
   Examples..... JSR 377
   JMP .-1
   A = D3
   L = D1ØØ

4. Symbolic Addresses and Literals Modified by a Constant:
   Examples: JMP Label - 1ØØØ
   L = Label - D3Ø

5. Diagnostics:
   For: Double defined labels
   Undefined labels
   Illegal labels
   Symbol table overflow
   Constant overflow
   Illegal addresses.

6. Listing: In addition to the original source instructions the assembler provides, from left to right:
   - Source tape line numbers (decimal)
   - Instruction memory address (octal)
   - Instruction memory data both in octal and modified ASCII.

   Page numbering is also provided.

7. Listing Controls: During the second pass, the operator has the choice of printing only the page number, only the line number or the complete listing. This is done by enabling the proper bits of the switch register and has effect even while the listing is taking place.

   If the program does not begin with an origin statement, the assembler will request one from the operator. In addition, the operator can select one of the following options at the beginning of pass 2.

   - the listing only
   - the binary tape only
   - both listing and binary tape
   - none of these (diagnostics only).

8. Label Formats: 1 to 5 alphanumeric characters may be used the first of which must be alphabetic (A to Z and @).

9. Label Field: Labels must begin in column 1. The remaining columns up to and including column 6 must be filled with spaces.

10. Mnemonic Filed: Mnemonics must begin in column 7 and are terminated either by a space or a carriage return.

11. Operand Fields: The operand of a branch, channel or origin instruction begins in column 11 and is terminated by a space or carriage return. The operand of a literal and data memory address instruction begins in column 9 and is similarly terminated by a space or a carriage return.

12. Comment Field: A comment may be inserted anywhere after a space, but before a carriage return, terminating a mnemonic or an operand.

13. Comment Line: An entire line may be devoted to a comment providing the first column contains an *.

---

* a listing sample follows

```
0161 2126 J@ 1200 ITOC  L=200    FLASHER FLAG MASK
0162 2127 @F 0006       JIC, RTC TX1 TEST CLOCK
     2130 1+ 6153
0163 2131 FA 0601       D=M
0164 2132 BO 0217       JIS, DB7 TX2 TEST FLASHER FLAG
     2133 1) 6151
0165 2134 F0 0660       JSR SET      SET     "       "
     2135 0K 6013
0166 2136 F0 0660       JSR INC      INC TOC (SAME WD)
     2137 0G 6007
0167 2140 JN 1216       L=216
0168 2141 F@ 0600       D=L
0169 2142 LF 1406       F=D-B-1   SET ALU
0170 2143 @D 0004       JIC, B=D TX2   B=D ?
     2144 1) 6151
0171 2145 F0 0660       JSR CLRWD
     2146 0U 6025
0172 2147 F0 0660       JSR SCF
     2150 1/ 6157
0173 2151 FN 0616 TX2   RET
0174 2152 @@ 0000       NOP
0175 *
0176 2153 F0 0660 TX1   JSR CLEAR
     2154 0V 6026
0177 2155 FN 0616       RET
0178 2156 @@ 0000       NOP
0179 *
0180 *
0181 * SCAN FAILURE SUBROUTINE
0182 *
0183 *
0184 2157 Q@ 2100 SCF   A=100
0185 2160 HP 1020       L=20
0186 2161 F0 0660       JSR SET AUDIO ALARM
     2162 0K 6013
0187 2163 Q) 2151       A=TXBUF+1   STATION ADDRESS
0188 2164 F0 0660       JSR BTOL
     2165 0< 6074
0189 2166 FA 0601       D=M
0190 2167 @K 0013       JIC, DB3 AS1 TEST MSB
     2170 1< 6174
0191 2171 Q3 2163       A=163
0192 2172 B@ 0200       JMP AS2
     2173 1= 6175
0193 2174 Q2 2162 AS1   A=162
0194 2175 LA 1401 AS2   F=D#B
0195 2176 FI 0611       B=F
0196 2177 FC 0603       D=B
0197 2200 FL 0614       M=D
0198 2201 PI 2011       A=FLAG1
0199 2202 HH 1010       L=10
0200 2203 F0 0660       JSR SET TX FLAG
     2204 0K 6013
0201 2205 FN 0616       RET
0202 2206 @@ 0000       NOP
0203 *
0204 *
```

6.2 TRUTH TABLE GENERATOR

Once a microprocessor program has been completely debugged, a set of ROM modules containing the successful program may be ordered.

This can be easily done by using the Truth Table generator program.

This program accepts a microprocessor object tape as input.

Its output is a 6-page truth table for every block of 256 word of ROM.

Presently the Read Only Memory is organized around 256 x 4 bit elements and the truth table is divided into 3 parts: right, middle, and left sections to form 256 x 12 bit words.

However the program can be easily modified to accommodate other ROM formats.

6.3 STANDARD LIBRARY

A series of often used subroutines is available presently and consists mostly of communication oriented tasks. Additions to this library are continuously made.

Here are some samples:

TRANS	Transfer a word from 1 place to another
INC	Increment a word
SET	Set bit(s) of a word
CLEAR	Clear bit(s) of a word
CLRWD	Clear a word (all bits)
SETWD	Set a word (all bits)
RAMØ	Erase all the RAM (used after power on)
ALF	Rotate the B-register 4 bits left (=right)
BTOL	Convert a binary number into linear
BCD 2	Convert 2 BCD digits into binary
CODE	Compute an 8 bit Bose Chaudhuri error code
WRITE	Output a word
READ	Input a word
LTOB	Linear to binary
BTBCD	12 bit binary to 4 BCD digits
SHFTR	16 bit word right shift of 1 bit
STORI	Store B INDIRECT (16 bit POINTER)
LOADI	Load B INDIRECT (16 bit POINTER)
INCDB	Increment a 16 bit word
DADD	Add 2 16 bit words
SHFTL	16 bit word left shift of 1 bit.

6.4 DIAGNOSTICS

A complete set of diagnostic programs allows a microprocessor user to check and trouble shoot a microprocessor when it is connected to a maintenance panel.

APPENDIX 1

BINARY TO OCTAL TO ASCII CONVERSION

BINARY	OCTAL	ASCII		BINARY	OCTAL	ASCII	
000000	00	@		100000	40		(SPACE)
000001	01	A		100001	41	!	
000010	02	B		100010	42	"	
000011	03	C		100011	43	#	
000100	04	D		100100	44	$	
000101	05	E		100101	45	%	
000110	06	F		100110	46	&	
000111	07	G		100111	47	'	(APOS.)
001000	10	H		101000	50	(	
001001	11	I		101001	51	)	
001010	12	J		101010	52	*	
001011	13	K		101011	53	+	
001100	14	L		101100	54	,	(COMMA)
001101	15	M		101101	55	-	
001110	16	N		101110	56	.	
001111	17	O		101111	57	/	
010000	20	P		110000	60	0	
010001	21	Q		110001	61	1	
010010	22	R		110010	62	2	
010011	23	S		110011	63	3	
010100	24	T		110100	64	4	
010101	25	U		110101	65	5	
010110	26	V		110110	66	6	
010111	27	W		110111	67	7	
011000	30	X		111000	70	8	
011001	31	Y		111001	71	9	
011010	32	Z		111010	72	:	
011011	33	[	(SHIFT K)	111011	73	;	
011100	34	\	(SHIFT L)	111100	74	<	
011101	35	]	(SHIFT M)	111101	75	=	
011110	36	↑	(SHIFT N)	111110	76	>	
011111	37	←	(SHIFT O)	111111	77	?	

FIG. A.1

TABLE OF BINARY TO OCTAL TO ASCII CONVERSION

APPENDIX 2

OPERATION CODE LIST

MNEMONIC	DESCRIPTION	CODE ASCII	CODE OCTAL
	LOAD DATA BUS INSTRUCTIONS		
D=L	L-REGISTER ONTO D BUS	F@	Ø6ØØ
D=M	DATA MEMORY ONTO D BUS	FA	Ø6Ø1
D=U	U-REGISTER ONTO D BUS	FB	Ø6Ø2
D=B	B-REGISTER ONTO D BUS	FC	Ø6Ø3
	RAM ADDRESS INSTRUCTIONS		
A=①	① INTO L.S. 1Ø-BITS OF A-REGISTER ②③		2ØØØ until 3777
AL=D	D BUS INTO L.S. 8-BITS OF A-REGISTER	FE	Ø6Ø5
AH=D	L.S. 4 BITS OF D BUS INTO M.S. 4 BITS OF A-REGISTER	FF	Ø6Ø6
A=A+1	INCREMENT A-REGISTER	F:	Ø672
	LOAD ACCUMULATOR INSTRUCTIONS		
B=Ø	CLEAR B-REGISTER	FH	Ø61Ø
B=F	ALU INTO B-REGISTER	FI	Ø611
B=FH	ALU M.S. 4 BITS INTO B-REGISTER	FJ	Ø612
B=FL	ALU L.S. 4 BITS INTO B-REGISTER	FK	Ø613
B=BRR	ROTATE B-REGISTER RIGHT	FI	Ø611
EBR	ENABLE B-REGISTER ROTATE	N@	16ØØ
	LOAD LITERAL BUFFER INSTRUCTION		
L=④	④ INTO L-REGISTER	⑤③	1ØØØ until 1377
	STORE INTO RAM INSTRUCTION		
M=D	D BUS INTO RAM	FL	Ø614
	U-REGISTER INSTRUCTIONS		
U=Ø	CLEAR U-REGISTER	FG	Ø6Ø7
U=U#D	'OR' CONTENTS OF U AND D-REGISTER AND STORE INTO U	FD	Ø6Ø4

ALU MODE INSTRUCTIONS ⓒ

MNEMONIC	CODE ASCII	CODE OCTAL	MNEMONIC	CODE ASCII	CODE OCTAL
F=D	L@	14ØØ	F=D'	LP	142Ø
F=B	LZ	1432	F=B'	LU	1425
F=-1	LC	14Ø3	F=Ø	LS	1423
F=D#B	LA	14Ø1	F=D#B'	LB	14Ø2
F=D'#B	LX	143Ø	F=D'#B'	LT	1424
F=D.B	L[	1433	F=D.B'	LW	1427
F=D'.B	LR	1422	F=D'.B'	LQ	1421
F=D↑B	LV	1426	F=D B'	LY	1431
F=D+D	LL	1414	F=D+B	LI	1411
F=D+D+1	L,	1454	F=D+B+1	L)	1451
F=D-B	L&	1446	F=D-B-1	LF	14Ø6
F=D+1	L (SPACE)	144Ø	F=D-1	LO	1417
F=D#B+D	LM	1415	F=D#B'+D	LN	1416
F=D#B+1	L!	1441	F=D#B'+1	L"	1442
F=D#B+D+1	L-	1455	F=D#B'+D+1	L.	1456
F=D.B+D	LH	141Ø	F=D.B'+D	LD	14Ø4
F=D.B+D+1	L(	145Ø	F=D.B'+D+1	L$	1444
F=D.B-1	LK	1413	F=D.B'-1	LG	14Ø7
F=D#B+D.B'	LE	14Ø5	F=D#B+D.B	LJ	1412
F=D#B+D.B'+1	L%	1445	F=D#B'+D.B+1	L*	1452

MNEMONIC	DESCRIPTION	CODE ASCII	CODE OCTAL
	SHIFT ROTATE INSTRUCTIONS ⑦		
F=BSL	SET ALU TO B SHIFTED LEFT. SET L.S.B. TO Ø	LL	1414
F=BRL	SET ALU TO B REGISTER ROTATED LEFT.	ML	1514
JMP	UNCONDITIONAL JUMP	B@	Ø2ØØ
JSR	JUMP TO SUBROUTINE	FØ	Ø66Ø
RET	RETURN FROM SUBROUTINE	FN	Ø616
NOP	NO OPERATION	@@	ØØØØ
PG=Ø	NEXT BRANCH TO PAGE Ø	DT	Ø424
PG=1	NEXT BRANCH TO PAGE 1	FT	Ø624
HLT=Ø	H	F (SPACE)	Ø64Ø
HLT=1	A	F!	Ø641
HLT=2	L	F"	Ø642
HLT=3	T	F#	Ø643
HLT=4		F$	Ø644
HLT=5	I	F%	Ø645
HLT=6	N	F&	Ø646
HLT=7	S	F'	Ø647
HLT=1Ø	T	F(	Ø65Ø
HLT=11	R	F)	Ø651
HLT=12	U	F*	Ø652
HLT=13	C	F+	Ø653
HLT=14	T	F,	Ø654
HLT=15	I	F-	Ø655
HLT=16	O	F.	Ø656
HLT=17	N	F/	Ø657

MNEMONIC FOR FLAG	DESCRIPTION OF BRANCH INSTRUCTION FLAG	CODE JIS	JIC
ALM	EXTERNAL ALARM	BA Ø2Ø1	@A ØØØ1
IOR	SERIAL I/O READY	BB Ø2Ø2	@B ØØØ2
BR7	M.S.B OF B-REGISTER	BC Ø2Ø3	@C ØØØ3
CRY	CARRY OUTPUT OF ALU	BD Ø2Ø4	@D ØØØ4
D=B	D BUS EQUAL TO B-REGISTER	BE Ø2Ø5	@E ØØØ5
RTC	REAL TIME CLOCK	BF Ø2Ø6	@F ØØØ6
IOD	SERIAL I/O DATA	BG Ø2Ø7	@G ØØØ7
DBØ	BIT Ø OF D BUS	BH Ø21Ø	@H ØØ1Ø
DB1	BIT 1 OF D BUS	BI Ø211	@I ØØ11
DB2	BIT 2 OF D BUS	BJ Ø212	@J ØØ12
DB3	BIT 3 OF D BUS	BK Ø213	@K ØØ13
DB4	BIT 4 OF D BUS	BL Ø214	@L ØØ14
DB5	BIT 5 OF D BUS	BM Ø215	@M ØØ15
DB6	BIT 6 OF D BUS	BN Ø216	@N ØØ16
DB7	BIT 7 OF D BUS	BO Ø217	@O ØØ17
RBF	RELINQUISH BUS FLAG	BP Ø22Ø	@P ØØ2Ø
SIN	SERIAL I/O INTERRUPT	BQ Ø221	@Q ØØ21
SFL	SERIAL I/O STATUS	BR Ø222	@R ØØ22
PIN	PARALLEL I/O INTERRUPT	BS Ø223	@S ØØ23
PFL	PARALLEL I/O STATUS	BT Ø224	@T ØØ24
INT	MASTER INTERRUPT FLAG	BU Ø225	@U ØØ25
PWR	POWER FAIL FLAG	BV Ø226	@V ØØ26
PDS	PUSH DOWN STACK OVERFLOW	BW Ø227	@W ØØ27

MNEMONIC	DESCRIPTION	CODE ASCII	CODE OCTAL
	I/O REGISTER SELECT INSTRUCTIONS		
RG=B	SET I/O REGISTER NO. TO 3 L.S. BITS OF B-REG.	FM	Ø615
RG=Ø	SET I/O REGISTER NO. TO Ø	FX	Ø63Ø
RG=1	SET I/O REGISTER NO. TO 1	FY	Ø631
RG=2	SET I/O REGISTER NO. TO 2	FZ	Ø632
RG=3	SET I/O REGISTER NO. TO 3	F[	Ø633
RG=4	SET I/O REGISTER NO. TO 4	F\	Ø634
RG=5	SET I/O REGISTER NO. TO 5	F]	Ø635
RG=6	SET I/O REGISTER NO. TO 6	F↑	Ø636
RG=7	SET I/O REGISTER NO. TO 7	F←	Ø637
	SERIAL I/O CLOCK, LOAD AND R/W INSTRUCTIONS		
CLK=Ø	CLEAR I/O CLOCK	DU	Ø425
CLK=1	SET I/O CLOCK	FU	Ø625
LD =Ø	CLEAR I/O LOAD	DV	Ø426
LD =1	SET I/O LOAD	FV	Ø626
R/W=R	SET I/O TO READ	DW	Ø427
R/W=W	SET I/O TO WRITE	FW	Ø627
	OTHER FUNCTION INSTRUCTIONS		
RBC	RELINQUISH BUS CONTROL	FP	Ø62Ø
EBC	ENABLE BUS CONTROL	DP	Ø42Ø
SIO	START SERIAL I/O	FO	Ø617
RST	RESET	F8	Ø67Ø
CHL=⑤	SET I/O CHANNEL TO ⑧	G⑨	Ø7ØØ until Ø737
DIN	DISABLE INTERRUPT	DQ	Ø421
EIN	ENABLE INTERRUPT	FQ	Ø621
IAK=Ø	CLEAR INTERRUPT ACKNOWLEDGE	DR	Ø422
IAK=1	SET INTERRUPT ACKNOWLEDGE	FR	Ø622

N O T E S

① - 1Ø BINARY BITS (Ø - 1777_8)

② - ONE OF THE FOLLOWING ASCII CHARACTERS:
PQRSTUVWXYZ[\]↑←

③ - ONE OF THE 64 ASCII CHARACTERS OF TABLE A-1.

④ - 8 BINARY BITS (Ø - 377_8).

⑤ - ONE OF THE FOLLOWING ASCII CHARACTERS: HIJK

⑥ -

#	LOGICAL OR
.	LOGICAL AND
↑	LOGICAL EXCLUSIVE OR
A'	LOGICAL COMPLEMENT OF A
+	ARITHMETIC PLUS OPERATION
-	ARITHMETIC MINUS OPERATION

⑦ - THOSE INSTRUCTIONS REQUIRE THAT D=L IS THE LAST DATA BUS INSTRUCTION EXECUTED.

⑧ - ANY DECIMAL NUMBER FROM Ø-31

⑨ - ONE OF THE 32 ASCII CHARACTERS ON THE LEFT SIDE OF FIGURE A-1.