CENTRAL PROCESSOR
MAINTENANCE SUPPORT INFORMATION
"DIMENSION®" "PRELUDE*" PBX

CONTENTS PAGE

1. GENERAL 2
PROCESSOR 2
COMMON CONTROL BUSES 2
MAIN MEMORY 4
MINIRECORDER 4
MEMORY CONTROL 4
SOFTWARE STRUCTURE 4
INPUT/OUTPUT DECODER 4
ALARM CIRCUIT 4
2. PROGRAMMED SEQUENCE OF COMMON CONTROL OPERATIONS 4
3. PROCESSOR OPERATION 7
EPL-3 (HIGH-LEVEL) INSTRUCTIONS . . . 7
MICROSTORE 8
PROCESSOR DATA MANIPULATION . . . 8
READ (EPL-3) INSTRUCTIONS 12
INTERRUPTS 12
A. High-Level Interrupts 17
B. Microlevel Interrupts 17
CENTRAL PROCESSOR INITIALIZATION . . 17
ALARM CIRCUIT 18

Trademark of AT&T

CONTENTS PAGE

4. MINIRECORDER 21
FUNCTIONAL OPERATION 21
SOFTWARE STRUCTURE 24
5. MAIN MEMORY 28
READ/WRITE OPERATION 28
6. INPUT/OUTPUT (I/O) CIRCUITS 28
I/O PROCESSING 30

Figures

1. DIMENSION PRELUDE PBX—Block Diagram 3
2. Typical EPL-3 Instruction Execution Sequence 8
3. Typical Microinstruction Implementation 9
4. DIMENSION PRELUDE PBX Processor and Memory—Block Diagram 10
5. EPL-3 High-level Instruction 12
6. Typical Microinstruction—Add, Move, Test 13
7. DIMENSION PRELUDE PBX Processor—Functional Block Diagram 14
8. DIMENSION PRELUDE PBX—Timing Diagram 16
9. Alarm Interface Circuitry 19

Printed in U.S.A.

CONTENTS PAGE

10. Alarm Panel Faceplate for LC557 Memory Control/Alarm Circuit Pack 20

11. Special Data Link (LC570B) 22

12. Typical DIMENSION PRELUDE PBX Software Structure 25

13. Maintenance Task—Block Diagram . . 26

14. Call Processing Timing Interval (Typical) 27

15. Typical Timing for a 25-ms Interval With Heavy Traffic Demands 27

16. Memory Controller and Alarm Circuit (LC557) 29

17. EPL-3 I/O Instructions 30

18. Data Channel Time Usage Propagation Delay (Low-Speed) 31

Tables

A. J58891AA-1 Trunk/Control Carrier Circuit Capacity and Position in Carrier . . . 5

B. (I/O Peripherial) Address Selection for DIMENSION PRELUDE PBX 23

1. GENERAL

1.01 This section provides a description of the DIMENSION PRELUDE PBX common control (central processor) which consists of the following elements:

- Processor
- Common control buses
- Main memory
- Tape unit
- Memory control circuit
- Software structure
- Input/Output (I/O) decoding circuits
- Alarm circuit.

1.02 Whenever this section is reissued, the reason(s) for reissue will be listed in this paragraph.

PROCESSOR

1.03 The DIMENSION PRELUDE PBX is an electronic switching system that uses stored program control for call processing and system diagnostics (Fig. 1). The program provided is erasable programmable read only memory (EPROM) located on the LC555 or LC595 circuit pack.

1.04 The call processing functions provided include the various sequential operations required to detect, service, and maintain call status before call initiation, during, and after call completion. Software programs are prepared for each PBX prior to delivery and include the services and features desired by the customer. These programs allow for flexibility in adding new features on customer premises with minimal effort via the Maintenance and Administration Panel (MAAP).

1.05 The processor controls system functions (such as providing dial tone and collecting digits) by executing programmed instructions, one at a time, but in a rapid sequential fashion. All system functions are under control of the processor at all times. Once a command from the processor has been issued, certain circuits (network connections) function autonomously until another command is issued by the processor.

COMMON CONTROL BUSES

1.06 The processor maintains control over the PBX via the common control buses and interface circuits. The common control buses consist of an address bus, bi-directional data bus, and control bus. Physically, the buses consist of connecting wires and logical circuit elements (LC20, LC52, LC53, LC21B, and LC22B) between the processor, memory, and I/O peripherals. These buses are communication links over which all data flows. An address bus is used to select locations in memory or specific I/O circuits for read/write operations. A data bus passes memory contents or I/O status information. The control bus provides a path for function signals (read, write,

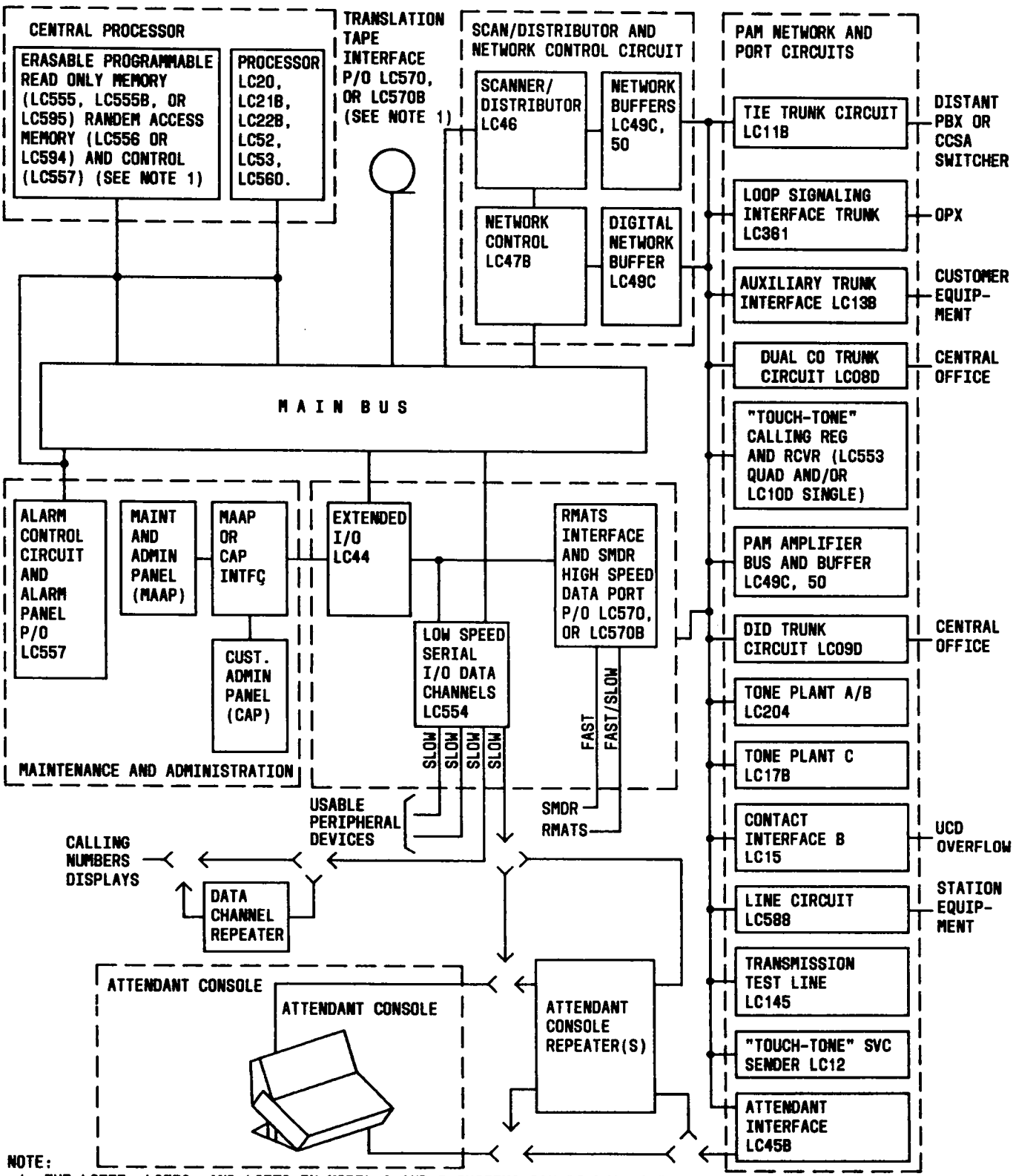


Fig. 1—DIMENSION PRELUDE PBX—Block Diagram

complete, etc) to and from high-level memory or I/O peripherals.

MAIN MEMORY

1.07 The processor and main memory are located in the PBX control/trunk carrier (J58891AA). The primary storage area of the central processor is called main memory. Main memory consists of 16-bit words contained in static random access memory (RAM) and EPROM elements. The LC556 or LC594 provides 18K and 32K words, respectively, of RAM and the LC555 or LC595 provides 88K words of EPROM. Memory usage is determined by customer requirements such as, number of lines, trunks, attendants, and special services, etc, desired.

MINIRECORDER

1.08 In a customer participation situation, a tape unit is provided to load the memory (initialize the system) following sustained power interruptions, or when data is lost for any reason. A tape cartridge contains the translations (system configurations) and status (call processing stimulus records for system monitoring).

1.09 The tape is kept up to date with system administrative changes via the run tape feature of the MAAP.

MEMORY CONTROL

1.10 The memory control circuit (LC557) provides the processor with an interface to main memory. The control circuit decodes the slot within the control/trunk carrier, and then decodes the row-column address to select the proper word in the RAM. The processor uses the control data bus to read from or write to this location.

SOFTWARE STRUCTURE

1.11 The PBX software is structured in a top-down hierarchy, that is, controlled via real-time executive programs. Each executive program is given control at scheduled intervals to perform such functions as digit collection and sending, trunk and button scanning, maintenance functions, etc. During heavy use periods, execution of some lower priority programs may be temporarily delayed. If these delays persist, a fixed work schedule is generated to insure completion of certain maintenance and periodic tasks.

INPUT/OUTPUT DECODER

1.12 The processor communicates with all peripherals via I/O buffer logic. Commands to network circuits in each cabinet, the data link channels, and MAAP interface circuits in the common control carrier pass through the I/O buffer circuits. The decoder circuits determine which peripheral device is to receive command information by:

(1) Decoding control and address inputs from the processor

(2) Latching data from the processor to the address peripherals.

1.13 This section is based on the drawings listed in Part 7. If this section is to be used with equipment or apparatus reflecting a later issue(s) of the drawing(s), reference should be made to SDs and CDs to determine the extent of the changes and the manner in which the section may be affected.

ALARM CIRCUIT

1.14 Software and maintenance circuitry constantly monitor critical functions. When a fault condition is detected, an alarm indication is generated. The alarm indication lights one or more fault indicators on the alarm circuit (LC557) pack.

1.15 The alarm and fault indicators are designated to provide a quick index to specific maintenance procedures. Using the index allows service personnel to return the system to the highest level of service in the least amount of time.

2. PROGRAMMED SEQUENCE OF COMMON CONTROL OPERATIONS

2.01 The DIMENSION PRELUDE PBX uses a 16-bit general-purpose, microprogrammed processor which is physically located in the control/trunk carrier of the basic control cabinet (Table A). Circuit packs LC21B, LC52, LC20, LC53, LC22/B, and LC560 constitute the processor and are located in slots 19 through 24, respectively.

2.02 The system common control (central processor) performs all of the functions necessary to maintain customer service. It monitors network connections, provides status indications to users, and performs operations necessary to implement the sys-

TABLE A

J58891AA-1 TRUNK/CONTROL CARRIER CIRCUIT CAPACITY AND POSITION IN CARRIER

CIRCUIT PACK		NO. OF CIRCUITS PER CP	MAXIMUM NO. OF CPs PER CARRIER	MINIMUM NO. OF CPs PER CARRIER	SLOT NUMBER																						
TYPE	CODE				01	02-08	10	11 12	15	16	17	18	19	20	21	22	23	24	27	29	32	30 33	31 34	35 36	39	40
Attendant Conference	LC06B	1	1	-		X																				
CO Trunk	LC08D	2	7	-		X																				
DID Trunk	LC09D	2	7	-		X																				
TOUCH-TONE Calling Reg and Rec	LC10D	1	7	-		X																				
Tie Trunk	LC11B	2	7	-		X																				
TOUCH-TONE Calling Sender	LC12	2	7	-		X																				
Auxiliary Trunk	LC13B	2	7	-		X																				
Contact Interface B	LC15	8	7	-		X																				
Processor Control	LC20	1	1	1											X											
Processor Control	LC21B	1	1	1									X													
Processor Control	LC22B	1	1	1													X									
Extended I/O	LC44	1	1	1																X						
Scanner/Distributor	LC46	1	1	1																					X	
Network Control	LC47B	1	1	1																						X

TABLE A (Contd)

J58891AA-1 TRUNK/CONTROL CARRIER CIRCUIT CAPACITY AND POSITION IN CARRIER

CIRCUIT PACK		NO. OF CIRCUITS PER CP	MAXIMUM NO. OF CPs PER CARRIER	MINIMUM NO. OF CPs PER CARRIER	SLOT NUMBER																						
TYPE	CODE				01	02-08	10	11 12	15	16 (NOTE)	17	18	19	20	21	22	23	24	27	29	32	30 33	31 34	35 36	39	40
Digital Buffer A	LC49C	1	1	1			X																				
Analog Buffer B	LC50	1	1	1	X																						
Processor Data Ckt	LC52	1	1	1										X													
Processor Interrupt Ckt	LC53	1	1	1												X											
Loop Signaling Intfc	LC361	2	7	-		X																					
TOUCH-TONE Calling Rec and Reg	LC553	4	1	-				X																			
Low Speed Data Channel	LC554	4	3	-																		X	X	X			
88K EPROM Memory	LC555B	1	1	1						X																	
18K RAM Memory	LC556*	1	1	1							X																
Memory Controller/Alarm	LC557	1	1	1								X															
Processor Control	LC560	1	1	1														X									
X-Ray Test	LC569	1	1	-					X																		
RMATS/Tape Interface	LC570B	1	1	1																	X						
32K RAM Memory	LC594	1	1	1							X																
88K EPROM Memory	LC595	1	1	1						X																	

Note: The LC555B is used in single-cabinet systems and the LC595 is used in multicabinet systems.

* The LC556 provides RAM for single-cabinet systems.

tem features. The system features are implemented by executing program instructions which are read from main memory. These instructions are called high-level instructions in order to differentiate them from the microinstructions which are contained in microstore.

2.03 Programs (a series of program instructions) which reside in main memory maintain the sequential flow of operations necessary for system control. Program instructions consist of one or more memory words. The programs are written in a modified electronic switching system programming language called EPL-3. The EPL-3 permits reading, writing, and testing—the three principal capabilities required for call processing, maintenance, and self-diagnostics. These instructions mainly consist of an operation code (op code) field and an operand field. An operation code identifies the function of the instruction, ie, arithmetic, logical, control, etc. The operand field contains data to be used when the function is performed and/or the location where the data is to be recorded.

2.04 The processor uses its common control buses (data, address, and control) to control the receiving and transmission of data, commands, and instructions. Main memory and both network and peripheral data channels are interconnected to the control buses. Common control activity at any time consists of reading programmed instructions from the main memory, decoding the instructions, and implementing the required functions. For example (Fig. 2), starting at time ***t,*** the processor fetches an instruction from main memory and decodes the instruction; at time ***t+1,*** the processor performs a read command by sending data to an I/O device followed by taking status information from the address I/O device; at time ***t+2,*** after interpreting the response from an I/O interface circuit, the processor continues to read additional EPL-3 instructions and records information in the status records of main memory. Finally, at time ***t+N,*** the processor issues a write I/O command and changes the status of peripheral equipment.

2.05 The processor decodes each high-level EPL-3 instruction by executing one or more microinstructions. A broad picture of the internal processor actions for a read I/O sequence is shown in Fig. 3.

2.06 The EPL-3 high-level instructions are read from main memory and appear at the input of the processor via its data bus. The op code field of each high-level instruction defines the starting address in microstore (EPROM) of the particular microinstruction sequence which implements the high-level instruction. The addressed microinstruction is read from microstore and applied to central processing and decoding logic circuits. Each microinstruction is decoded and data provided from the EPL-3 operand field is used in internal control and logic operations. If several microinstructions are required to implement the op code, an address for each subsequent (next) microinstruction is generated internally and the microstore is read again. This sequence of microinstruction addressing and execution continues until the appropriate data manipulations have been performed. For this description, the processor has transferred information via the address, data, and control buses to an I/O decoder.

3. PROCESSOR OPERATION

3.01 A functional diagram of the 16-bit general-purpose, microprogrammed central processor and main memory is shown in Fig. 4. The processor communicates with the network, peripherals, and main memory via its common control buses (address, data, and control). A 16-bit address bus (MA00* through MA15*) permits direct addressing for up to 64K words of main memory and board select (BDSEL0* and 1*) signals select one of the two memory boards. Input/output ports are also addressable via the address bus.

3.02 A 16-bit bi-directional data bus is used to transfer data between the processor and main memory or I/O channels. The control bus generates enable commands and monitors main memory and/or I/O channel responses to the processor commands. External peripheral devices, such as the MAAP, are periodically interrogated by the processor; and certain devices, such as the tape unit may independently signal (interrupt) the processor.

EPL-3 (HIGH-LEVEL) INSTRUCTIONS

3.03 The processor is designed to execute code generated by an EPL-3 compiler. The EPL-3 instructions constitute the (software) programs which are located in main memory. These 16-bit program instructions consist of two primary fields: an operand and an op code field. Figure 5 shows an EPL-3 instruction with the fields labeled. The EPL-3 op code addresses a microinstruction (in microstore) which is

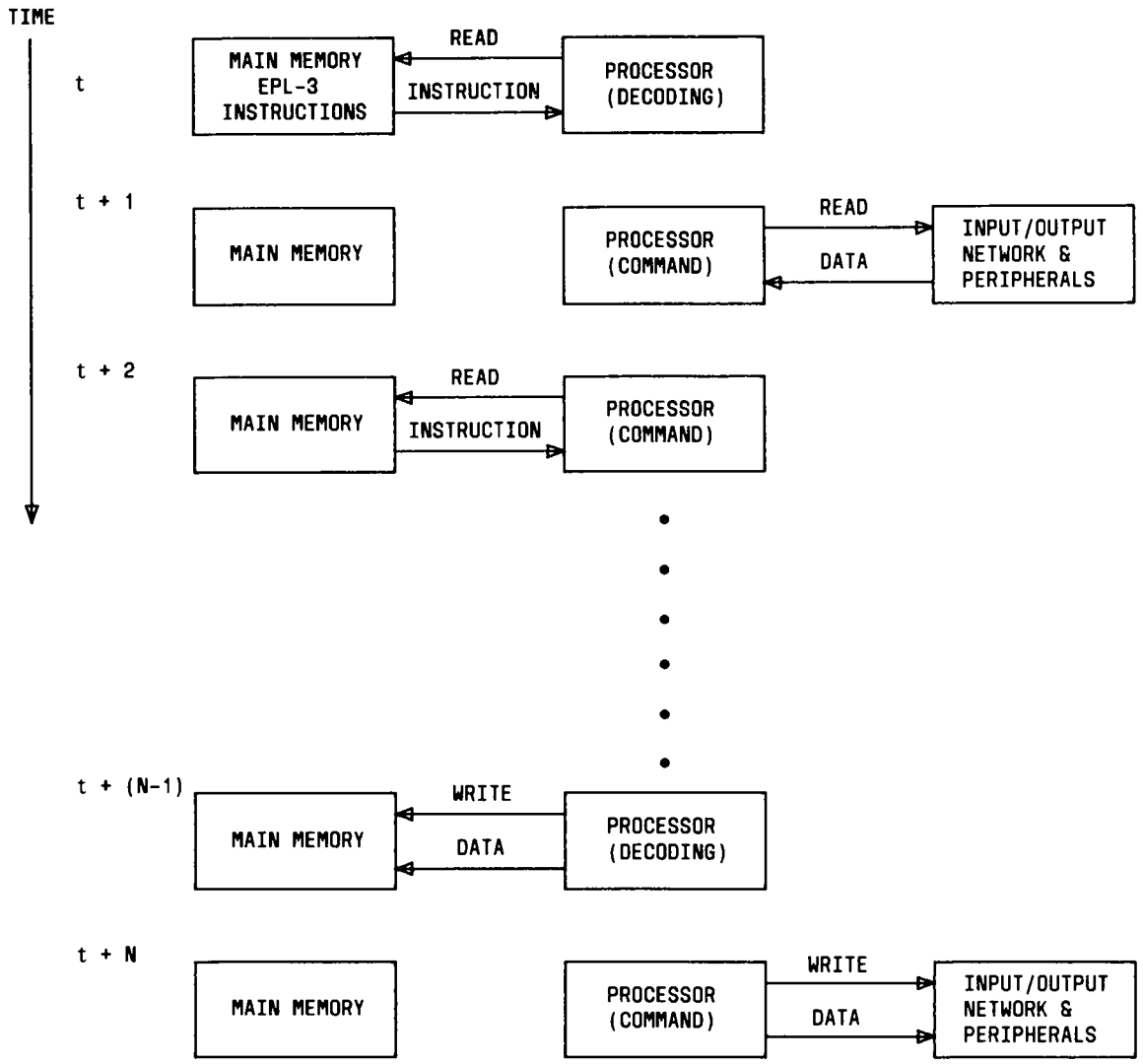


Fig. 2—Typical EPL-3 Instruction Execution Sequence

the beginning of a particular microprogram. The operand (bits 0 through 7) is immediate data which is used by the microinstructions. This addressed microprogram contains the emulation codes for implementing op code (HEX 09).

MICROSTORE

3.04 The microstore contains the microdiagnostics (MDs), the interrupt handler, and the emulation codes (microprograms) in the form of a sequence of microinstructions. The microstore controls the sequence in which a system reload is performed. When a system reload is required, the processor services a microinterrupt which addresses and executes a bootstrap loader routine.

3.05 Figure 6 depicts the three types of microinstructions (ARITHMETIC, MOVE, and TEST). These 24-bit microinstructions are shown with the fields labeled and each field is described in the legend. Each microinstruction field designates a particular operation and/or controls a particular hardware element within the processor (Fig. 7).

PROCESSOR DATA MANIPULATION

3.06 The arithmetic manipulation unit performs arithmetic and logic functions on immediate

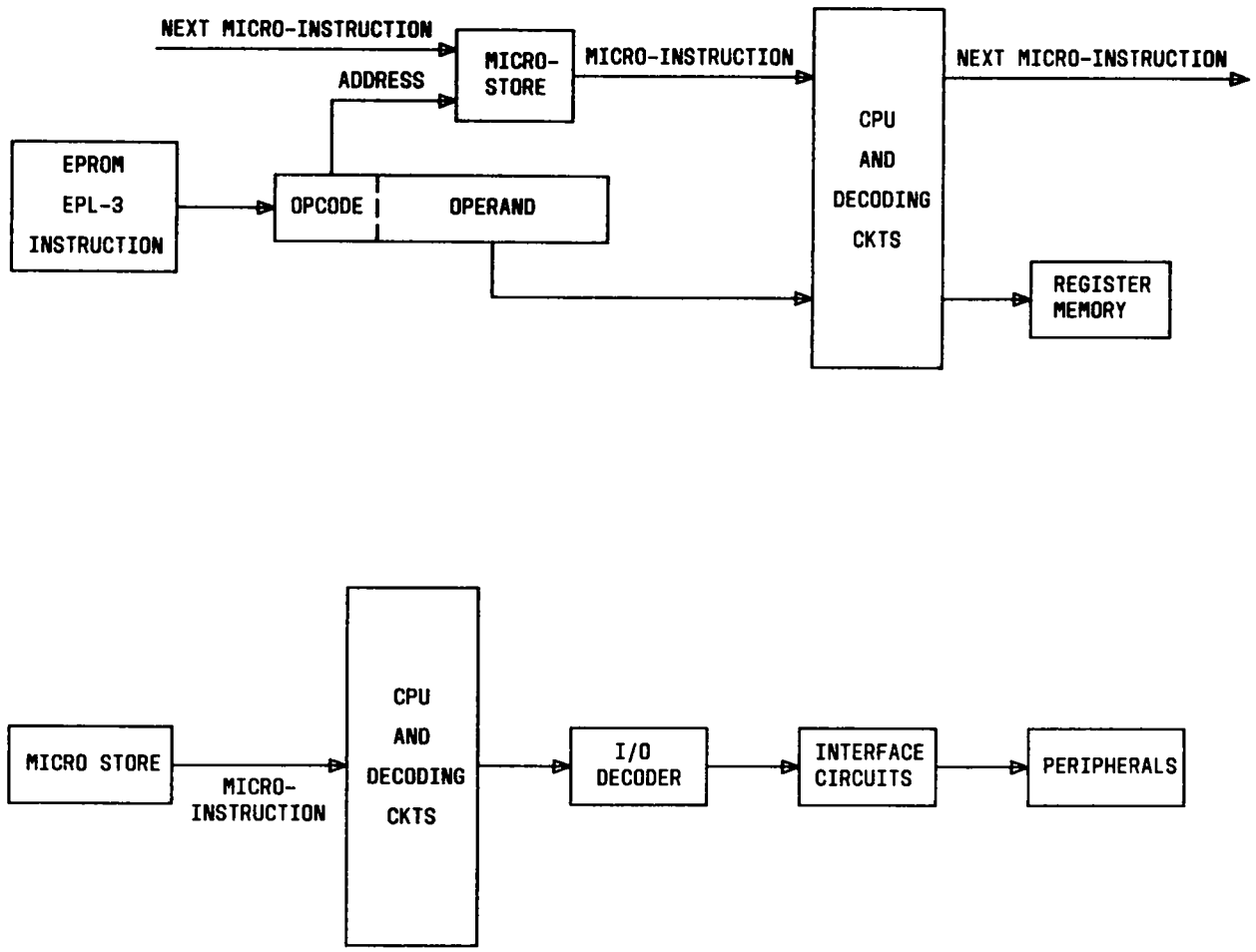


Fig. 3—Typical Microinstruction Implementation

data. All data transferred between elements of the processor must pass through this circuitry. Data movement and manipulation within the processor is accomplished by breaking each 16-bit word (such as from line scanning) into four 4-bit bytes. Time required for processing a data word is called a microcycle (Fig. 8). Each microcycle is divided into four phases designated ***0, 1, 2,*** and ***3.*** Each phase represents a 4-bit byte operation. For example:

- Byte 0—bits 0 through 3 (phase 0)
- Byte 1—bits 4 through 7 (phase 1)
- Byte 2—bits 8 through 11 (phase 2)
- Byte 3—bits 12 through 15 (phase 3).

3.07 Each phase is divided into two parts called subphases. These subphases are called load and clock. During the subphase load, the selected source of data (memory or register) is gated onto source bus #1 (or #2). At the end of load subphase, the 4-bit byte is strobed into ***A*** and/or ***B*** registers. During the subphase clock, data in the arithmetic and manipulation unit is operated upon and placed on the destination bus. Following completion of the subphase clock, data on the destination bus is strobed into a selected destination.

3.08 The arithmetic manipulation unit consists of four functional units: an arithmetic logic unit, A-register, B-register, and the A-ROM. The A-ROM contains preprogrammed arithmetic instructions which are addressable via an ***ARTH*** field. These arithmetic instructions function to set/clear, control

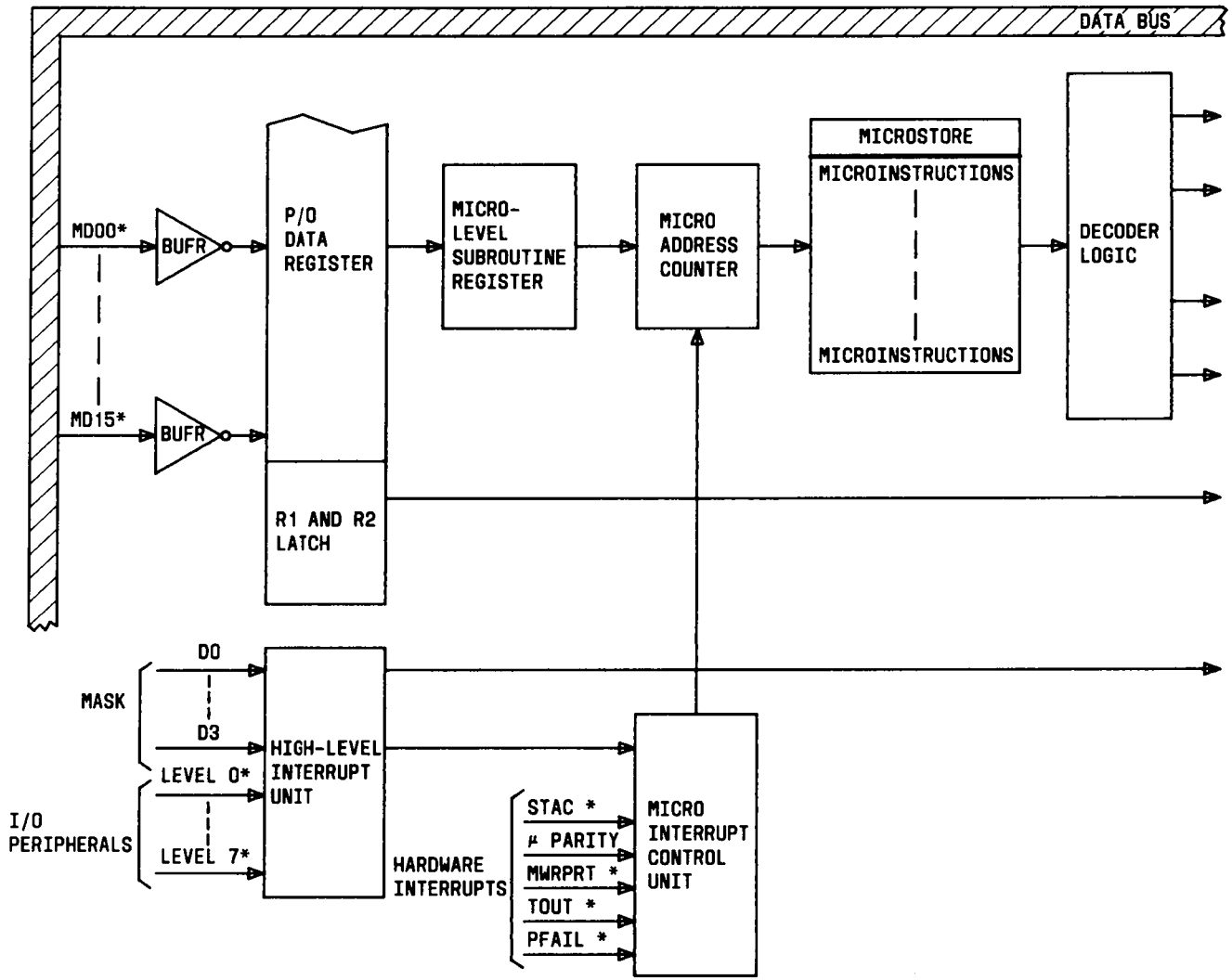


Fig. 4—DIMENSION PRELUDE PBX Processor and Memory—Block Diagram (Sheet 1 of 2)

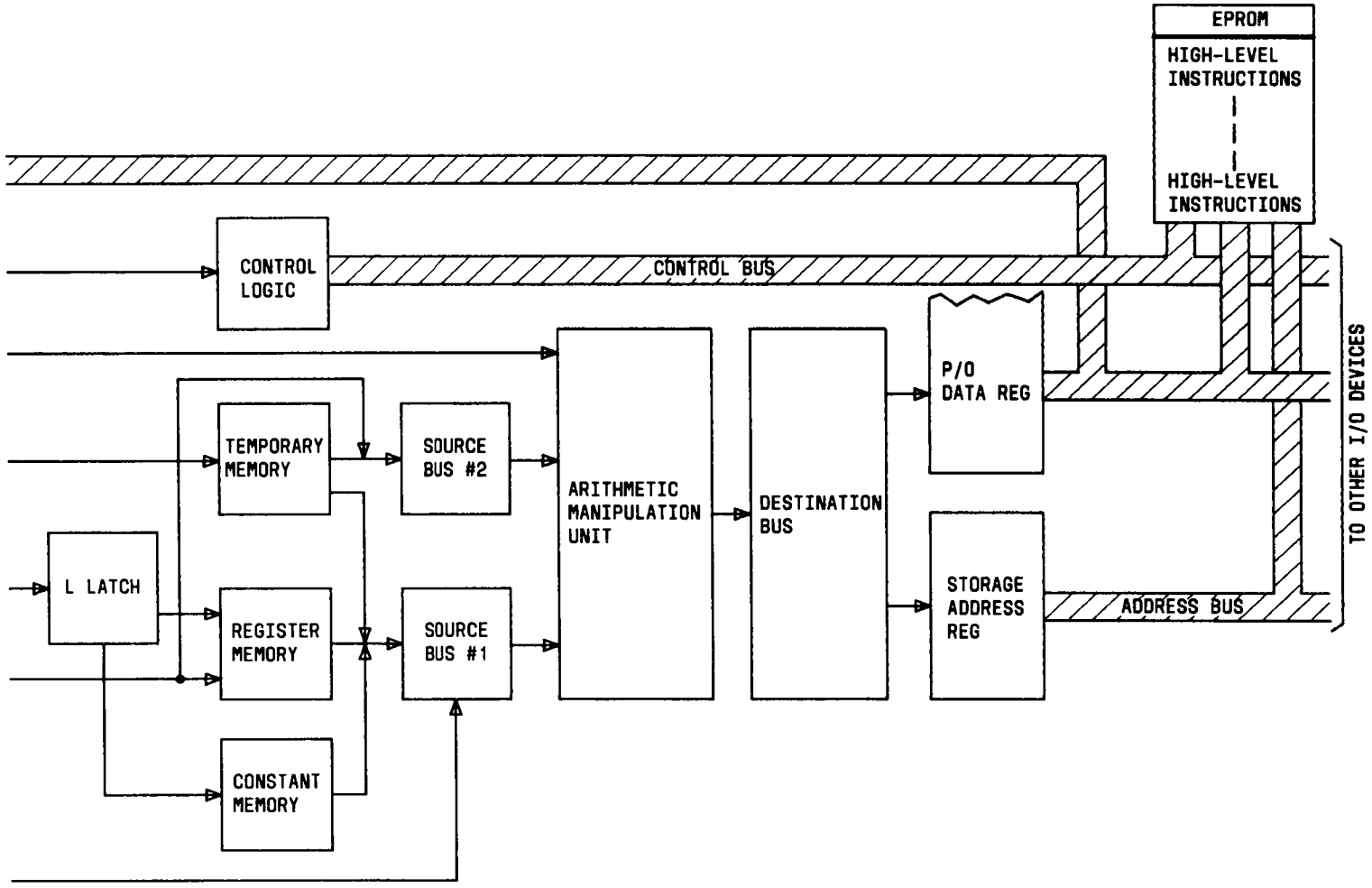
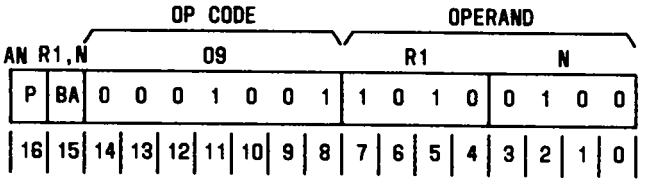


Fig. 4—DIMENSION PRELUDE PBX Processor and Memory—Block Diagram (Sheet 2 of 2)



LEGEND:

1. P = PARITY BIT
   BA = BRANCH ALLOW BIT
2. OP CODE 09 (HEXIDECIMAL) TRANSLATES TO:
   ADD N (DATA) TO THE CONTENTS OF R1 (REGISTER IN REGISTER MEMORY) AND STORE THE RESULTS IN R1.
3. R1 = 4 BITS DEFINING 1 OF 16 GENERAL PURPOSE REGISTERS IN THE REGISTER MEMORY.
   N = 4 BITS OF DATA
4. SET THE CONDITION FLOP (CF) EQUAL TO:
   A. ONE, WHEN THIS INSTRUCTION CAUSES A CARRY BEYOND BIT 15.
   B. ZERO, WHEN THERE IS NO CARRY BEYOND BIT 15.
5. THE ABOVE TRANSLATES TO REGISTER 10 BEING INCREASED BY A VALUE OF 4.

Fig. 5—EPL-3 High-level Instruction

flip-flops, and determine the arithmetic or logic function to be executed.

3.09 The operational sequences during call processing consist of a series of activities performed by the processor as follows:

(1) Reading a high-level instruction or data from main memory (EPROM)

(2) Executing a series of microinstructions to implement the EPL-3 instruction

(3) Issuing commands to peripherals and/or reading or writing into main memory as a result of implementing an EPL-3 instruction.

READ (EPL-3) INSTRUCTIONS

3.10 High-level instructions (EPL-3) are read from EPROM memory under microinstruction control. A READ* command is implemented when contents of the storage address register (Fig. 7) are placed on the address bus and control bus signal READ* is made active low. Contents of an addressed main memory location are strobed onto the processor data bus. When the addressed memory word has been placed (read) onto the data bus, a memory complete signal is returned to the processor. Control flip-flops within the processor (set by previous instructions) identify the next fetched word as an instruction or as a data word.

3.11 All 16 bits, both op code and operand, are transferred via the data bus into the processor input data register. The instruction op code addresses microstore and the operand is latched into the R1-R2 latch. This operand may be used to address the register memory (16 general-purpose registers) or as immediate data, depending on the particular EPL-3 instruction.

INTERRUPTS

3.12 The processor (Fig. 7) uses a queued interrupt structure. This means that because of a hardware event (an interrupt), the program currently operating is temporarily stopped while a routine of higher priority (the interrupt) is serviced. The main memory is utilized in a software-controlled file-structured system. A "hold-get" file consists of a series of consecutive memory words forming a register stack in memory. Each level of the stack is 13 words in length. These stacks store contents of the register memory when EPL-3 subroutines and high-level interrupts are serviced. Programming is such that

ARTH 01

P	OP	CF	ARTH	DES	S2	S1	CONTROL FIELD
23	22 21	20	19 16	15 12	11 9	8 5	4 0

LEGEND:
P = PARITY BIT
OP = INDICATES TYPE OF INSTRUCTION, IE: ARITHMETIC 01.
CF = CONTROL OF CONDITION FLOP-FLIP/FLOP
ARTH = 1 OF 16 ARITHMETIC INSTRUCTIONS - ADDRESS TO AN EPROM
DES = ADDRESS DECODED OF INFORMATION ON DESTINATION BUS
S2,S1 = DATA INPUT - A OR B REG. TEMPORARY MEMORY (TM), OR REGISTER MEMORY (RM)
CONT = CONTROL OF FLIP-FLOPS FOR READ, WRITE, WRITE PROTECT, ETC

A. ARITHMETIC MICROINSTRUCTION

MOV 10

P	OP	CF	EMIT	DES	EMIT	X	S1-S2	CONTROL FIELD
23	22 21	20	19 16	15 12	11 10	9	8 5	4 0

* S1=0; S2=1

LEGEND:
P = PARITY BIT
OP = INDICATES TYPE OF INSTRUCTION, IE: MOVE 10
CF = CONTROL OF CONDITION FLOP-FLIP/FLOP
EMIT = ADDRESS ON L LATCH
DES = ADDRESS DECODED OF INFORMATION ON DESTINATION BUS
* = IF DATA IN FIELD BITS 5 THROUGH 8 IS S1, THIS BIT SET TO 0
IF DATA IN FIELD BITS 5 THROUGH 8 IS S2, THIS BIT SET TO 1
S2,S1 = DATA INPUT - A OR B REGISTER, TEMPORARY MEMORY (TM), OR REGISTER MEMORY (RM)
CONT = CONTROL OF FLIP-FLOPS FOR READ, WRITE, WRITE PROTECT, ETC

B. MOVE MICROINSTRUCTION

TEST 11

P	OP	CF	JUMP CTL	TEST			BRANCH ADDRESS
23	22 21	20	19 18	17 14	13	12	11 0

LEGEND:
P = PARITY BIT
OP = OP CODE INDICATES TYPE OF INSTRUCTION, IE: TEST 11
CF = CONTROL OF CONDITION FLOP-FLIP/FLOP
JUMP CTL = JUMP CONTROL BIT UD18 AND UD19 FORM MICRO LATCH
TEST = TYPE OF TEST CODE

BITS 0 THROUGH 11 DESIGNATE A BRANCH ADDRESS
ON TEST INSTRUCTIONS UD20 ACTIVATES WR10* LEAD, UD13 DEACTIVATES

C. TEST MICROINSTRUCTIONS

Fig. 6—Typical Microinstruction—Add, Move, Test

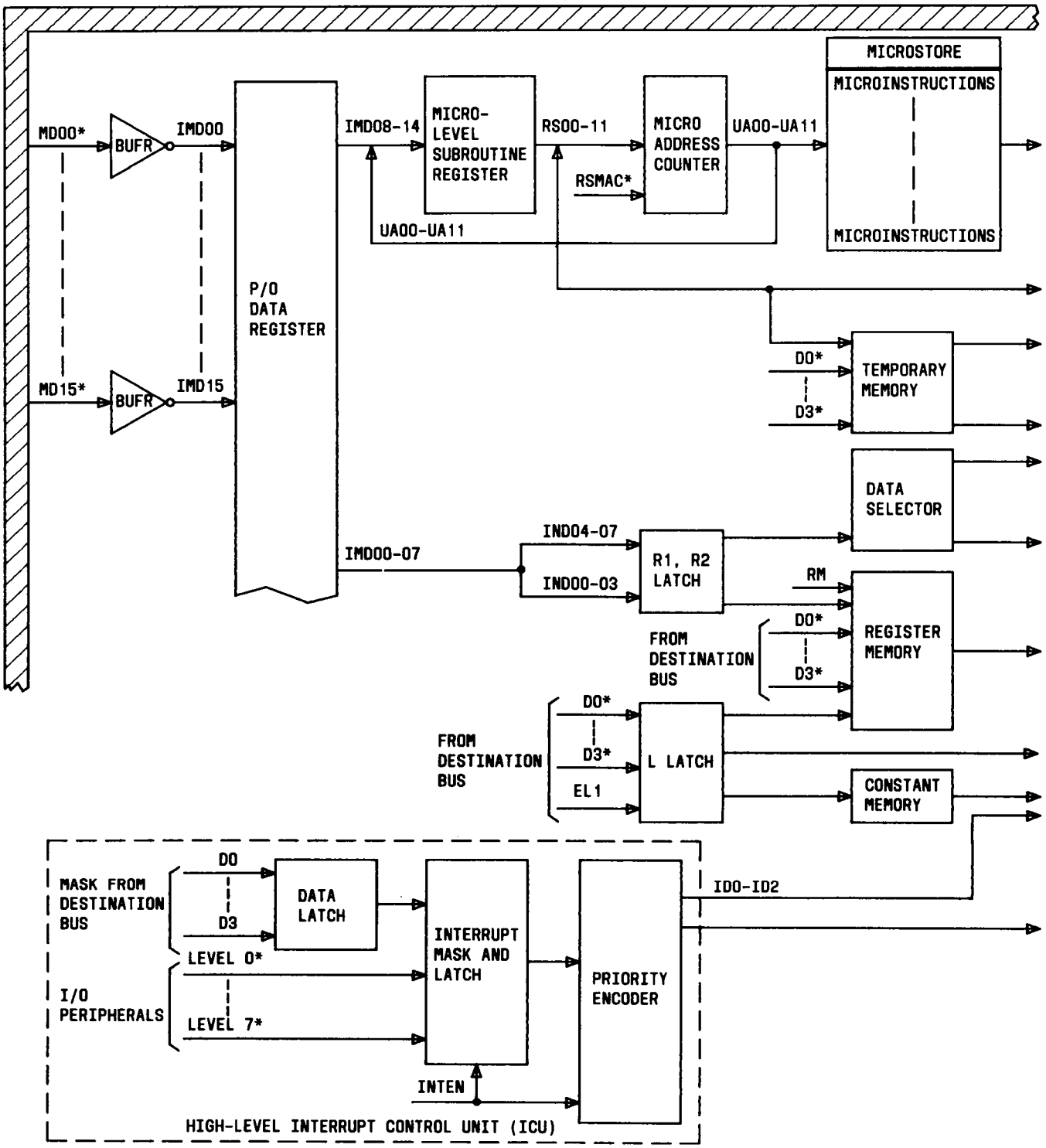


Fig. 7—DIMENSION PRELUDE PBX Processor—Functional Block Diagram (Sheet 1 of 2)

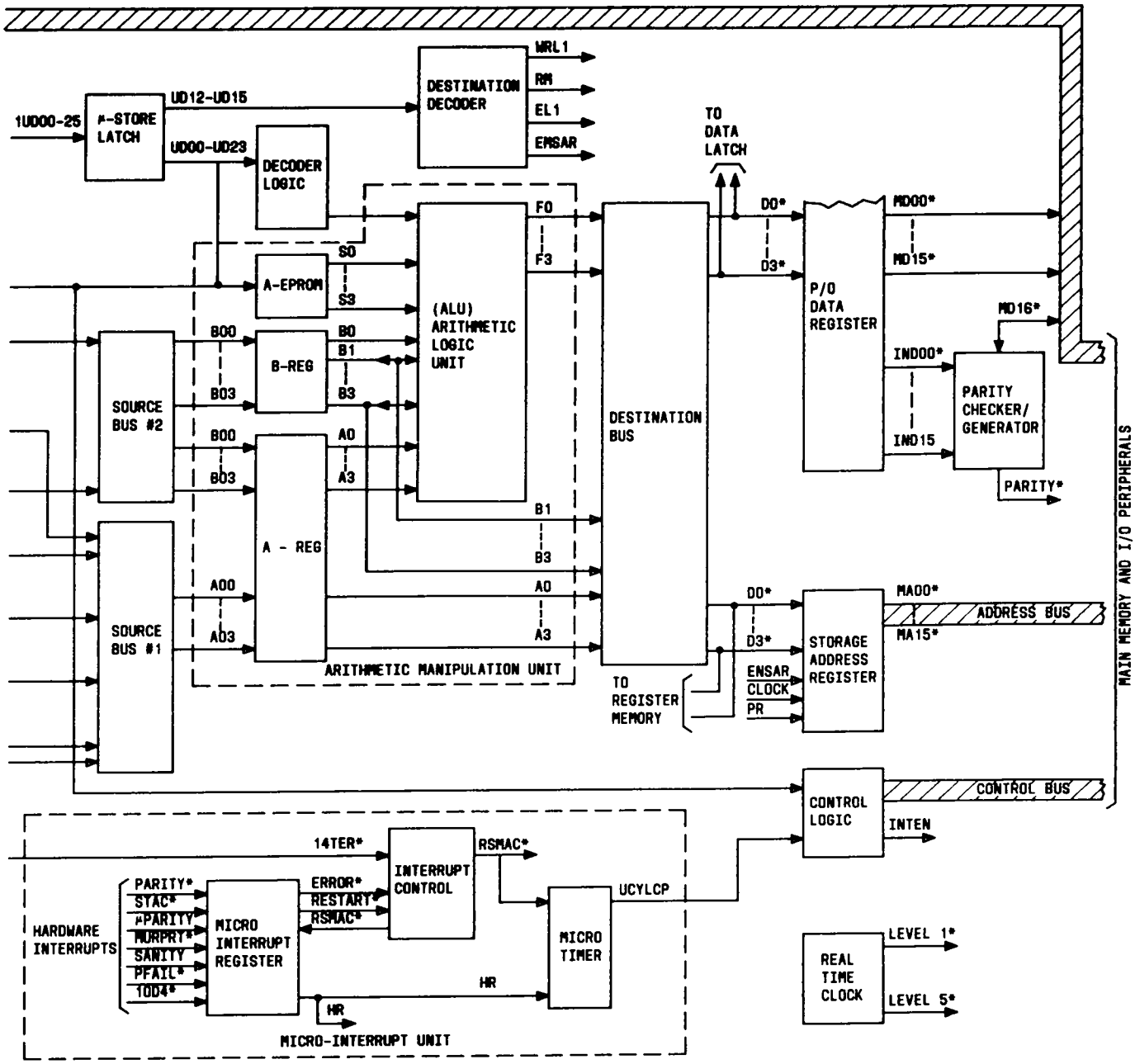


Fig. 7—DIMENSION PRELUDE PBX Processor—Functional Block Diagram (Sheet 2 of 2)

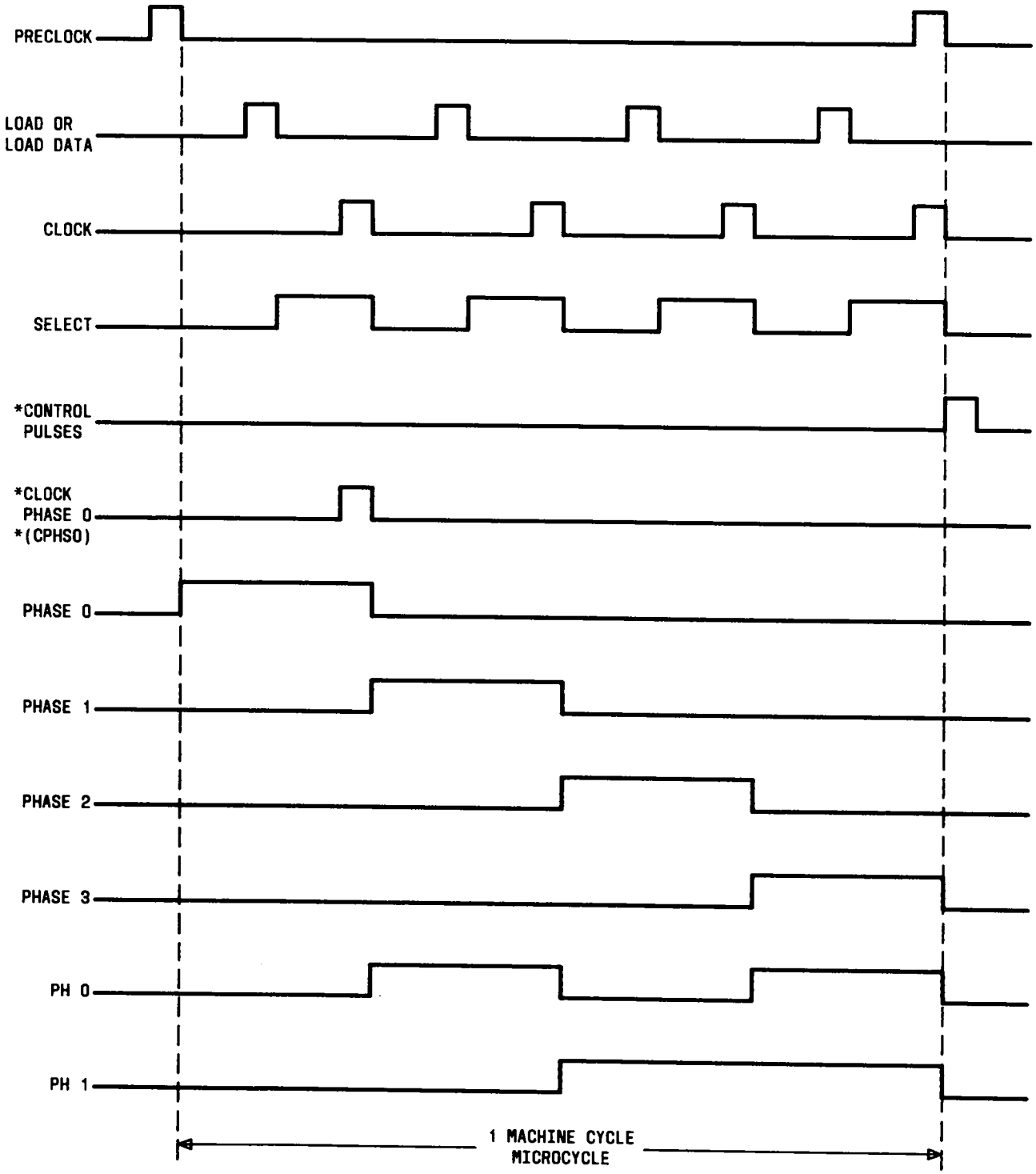


Fig. 8—DIMENSION PRELUDE PBX—Timing Diagram

when the priority routine is serviced, control is returned to the point at which the original program was interrupted. All interrupts are serviced automatically. An interrupt signal could be for any one of the following: real-time clock (RTC), I/O devices, or abnormal/error conditions. The two types of interrupt structures used in the processor are high-level and microlevel. Figure 7 depicts both high-level and microlevel functional interconnections with the processor.

A. High-Level Interrupts

3.13 A high-level interrupt control unit (ICU) is used to exchange interrupt information from peripheral (I/O) devices with the processor. These interrupt signals are arranged in a priority, or queued multilevel sequence (Fig. 7). These interrupts (LEVEL 0* through LEVEL 7*) are signals from I/O hardware units. The ICU is software-controlled and the interrupt levels are maskable.

3.14 The high-level ICU consists of three functional units—a 4-bit data latch, an 8-level interrupt mask and latch, and a priority encoder. Signal INTEN enables the ICU, allowing nonmasked (not blocked) active interrupts to generate INTER*. An INTER* active low input to the interrupt control signals the processor of an interrupt. Encoded interrupt levels are latched onto source bus #1. The interrupting level is rewritten to the data latch and blocks further interrupts at the lower and interrupting level while conditionally enabling interrupts at higher levels.

Signal designations shown with an asterisk (INTER* for example) indicate that the normal logic level is high (1) and the primary function of the lead is performed in low (0) logic state. Lead designations without an asterisk (INTEN) indicate that the logic level is normally low (0) and the primary function of the lead is performed in the high (1) logic state.

B. Microlevel Interrupts

3.15 The microinterrupt control unit is used to service microlevel hardware-generated interrupts in real time. Microinterrupts are fault (abnormal) or signal indications concerning the state of the processor hardware and software operations. The microinterrupt system gives immediate control to diagnostic, recovery, or service programs when microinterrupts are detected.

3.16 Control signal IODN* is the indication that an I/O operation has been performed and the processor may take the peripheral response, process it, and continue operation. The processor contains a hardware timer known as a sanity timer. The sanity timer monitors hardware-software interactions and detects gross malfunctions (errors). The sanity timer has a time period of approximately 400 ms. Base-level maintenance program instructions periodically reset the timer (within its time period) if all is well. Normally, the processor cycles the alarm panel PASS/FAIL lamp on-off in 1-second intervals, giving a heart-beat indication. However, if sanity timer is not triggered, an error condition, signal SANITY, is generated.

3.17 Microinterrupt PARITY* indicates whether the current EPL-3 instruction or data read from memory has correct parity. The alarm interface circuitry monitors power supply voltages and, if these voltages are not maintained within predetermined limits, microinterrupt signal PFAIL* is generated. When signal PFAIL* is active, the processor is halted. Actual power failures may result in the contents of main memory (RAM) being destroyed and require a system reinitialization.

3.18 An example of an interrupt service request would be from the minirecorder interface. LEVEL 1*, an input to the interrupt mask and latch, is physically connected to the minirecorder interface. When the software generates an interrupt signal (INTEN), a microinterrupt (INTER*) may be generated if there is an active nonmasked interrupt. Signal INTER* forces a microlevel interrupt. A recovery and service routine determines the interrupting source (eg, tape interface) and dispenses control to the tape monitor program.

CENTRAL PROCESSOR INITIALIZATION

3.19 When the PBX is first powered up or when subsequent long-term power failures occur, the network and main memory (RAM) must be initialized. Microinterrupt signal RSMAC* (low) may, depending upon the condition for the microinterrupt, force an initialization program to be executed. Signals PWONFF* and PFAIL* may be generated from an actual or pseudo-power failure. Pseudo-power

failures are generated by manually placing the alarm panel (GO/HALT) switch in the HALT position. The processor cannot distinguish between a pseudo-power failure and an actual power failure. In either case, the processor searches memory for a word called the program key. The next sequential address, following the program key, should (normally) contain the inverse of the program key. Following this is a word which contains the address of the area of memory which is active. The processor tests a sampling of this memory and if parity errors are detected, it is assumed that the power failure was long enough to destroy the instructions contained in memory. Following the conclusion that memory contains faulty instructions, a system reinitialization from tape is performed.

3.20 The PBX has a software initialization program which is normally contained in main memory. This program establishes network status as a result of any of the following:

- Short power failures (static) in which contents of the (RAM) main memory are not lost, but network connections were lost
- Termination of a pseudo-power failure generated by the alarm panel (GO/HALT) switch
- Generation of system error(s)
- A microlevel interrupt.

3.21 The off-line (network) initialization program monitors the number of times it is executed and, if the number becomes excessive, a complete system initialization is performed. The system initialization program (an off-line routine) is loaded into the paging buffer, a 1K-word area of main memory, while the tape is being loaded. A system initialization differs from a network initialization in that the tape must be reloaded and all call processing status tables be constructed by interrogating the network connections. The PBX may be initialized automatically during call processing or manually via the alarm panel when the alarm panel SELECT switch is in position 9 and by depressing the ENABLE switch.

ALARM CIRCUIT

3.22 The processor and alarm sensors interface with the alarm circuit by turning off/on light emitting diodes (LED)s mounted on the faceplate. The alarm circuit interfaces with the processor by using a switch to select diagnostic tests.

3.23 The diagnostic tests are programs used by the processor for detection and isolation of component failures. These diagnostic test programs (MD0 through MD9) are contained in microstore (EPROMs) and are individually addressable (ACD0 through ACD3) (Fig. 9) via the alarm panel SELECT switch (Fig. 10). Microdiagnostic (MD) status is indicated by appropriate LEDs on the alarm panel. A brief description of the components checked by each MD procedure is as follows:

- The MD0 is used to verify correct operation of the microlevel subroutine register, microaddress counter, sanity timer, and 20-MHz clock.
- The MD1 is used to validate proper operation of the processor data register.
- The MD2 checks for proper operation of all 16 registers with the register memory.
- The MD3 checks the real-time clock (RTC) and high-level interrupt control unit.
- The MD4 exercises the L-latch and constant memory.
- The MD5 verifies correct operation of the arithmetic manipulation unit by performing a shift (rotate) operation.
- The MD6 checks the temporary memory.
- The MD7 is used to check the I/O data bus.
- The MD8 checks RAM and EPROM memory and its decoding and buffering circuits.
- The MD9 runs one cycle of test (MD0 through MD8) followed by loading of the program tape.

3.24 Depressing the start test button causes the start test (STAC*) lead to go low. The processor detects the low on the STAC* lead and runs the selected test. The results of the test are received by the software alarm interface circuit via data (D0* through D3*) lead. The processor places a low on the

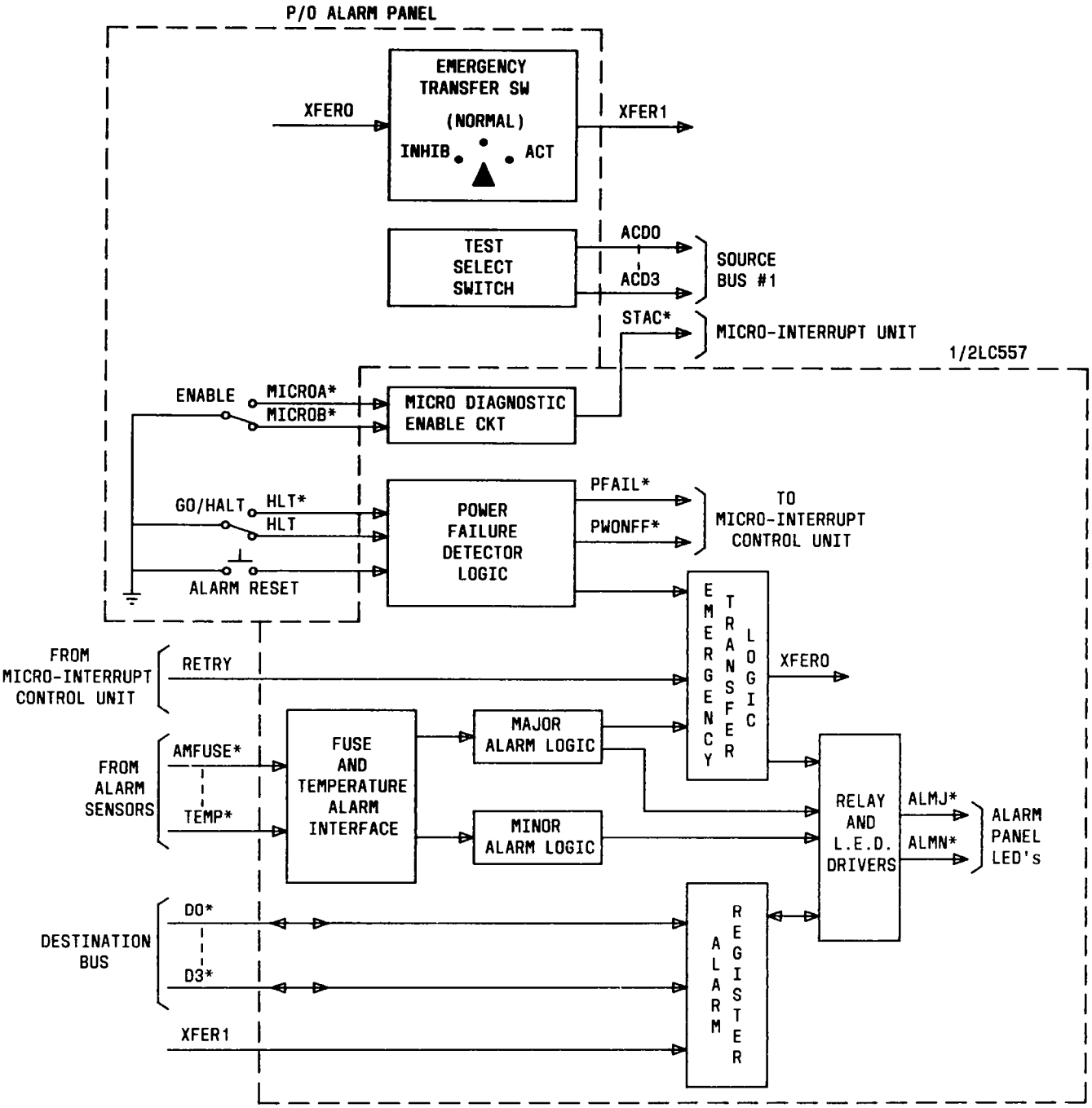


Fig. 9—Alarm Interface Circuitry

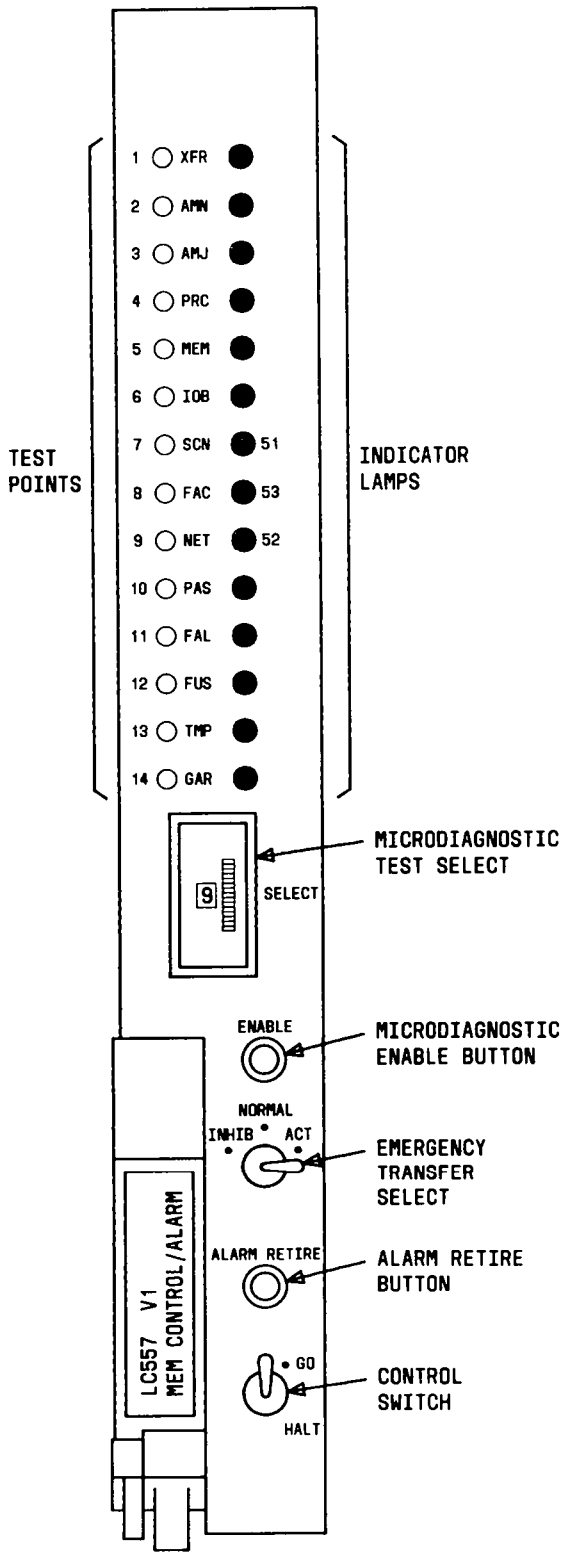


Fig. 10—Alarm Panel Faceplate for LC557 Memory Control/Alarm Circuit Pack

phase 1*, phase 2*, or CPHS0* lead to enter the data in one of three registers located in the software alarm interface circuit. A low on the EWAC* lead enters the data into all three registers.

3.25 The software alarm interface sends alarm status to the alarm driver and display circuit and the major/minor alarm circuit. The driver and display circuit (Fig. 9) uses these outputs to set the off/on status for the following indicators: (1) PASS, (2) FAIL, (3) NETWORK, (4) FACILITY, (5) SCANNER, (6) I/O BUS, and (7) MEMORY.

3.26 The power failure detect circuit also provides alarm status indications to the major/minor alarm circuit. The power failure detect circuit receives input from the following sensors: power on, power fail and simulated power failure, via the PON*, PFAIL*, and PWONOFF* leads, and outputs the status to the major/minor alarm circuit.

3.27 The major/minor alarm circuits receive input from alarm sensors, the software interface circuit, and the power failure detect circuit. The following sensors provide inputs directly to the alarm circuit via: (1) APMJ, (2) APMN, (3) ANMJ*, (4) ANMN*, (5) TEMP*, (6) RETRY, and (7) TOUT leads.

3.28 The output of the major/minor alarm circuit establishes the off/on status of the drivers in the alarm driver and display circuit for the following displays: (1) MAJOR, (2) MINOR, (3) FUSE, and (4) processor (PROC). The major/minor alarm circuit also establishes the status of major and minor alarm indicators at the attendant console, MAAP and night console transfer via the AMJ*, AMN*, ALRMN*, and ALRMJ* leads.

3.29 The faceplate of the alarm circuit is equipped with an alarm clear switch. Depressing this switch places a low on the alarm clear (ALMCLR*) lead to the processor. The processor clears the alarm indicators momentarily then rewrites existing alarm conditions.

3.30 The I/O performing (IOP*) signal is generated by the data channel when a read or write command is received. The signal is decoded by the I/O buffer and directed to the processor via the I/O data (IODN*) lead. The signal informs the processor that the read or write command has been received and decoded.

4. MINIRECORDER

4.01 The DIMENSION PRELUDE PBX provides for an interface with the Braemar minirecorder. The interface is provided in the Electronic Industries Association (EIA) RS232 format at a rate of 9.6k baud. The LC570B provides the circuitry necessary to convert between standard PBX format and EIA format.

4.02 The Braemar unit consists of a serial interface, a microprocessor, and a tape unit in a self-contained carrying case. For operation, the unit requires a standard 110-volt power source and cabling, that provides a ground and five leads for the EIA interface. The serial interface provides the necessary circuitry to convert between EIA format and microprocessor format. The microprocessor acts as a buffer and interprets commands to control the tape unit. The tape unit records (writes on the tape) and reproduces (reads from the tape) digital data.

4.03 The data recorded on the magnetic tape consists of translations and status. This data may be read from or written to and is accessible for modification.

4.04 The tape cartridge is used to maintain a permanent copy of the translation data. Equipment changes, such as line and/or trunk additions, reassignments, or station feature changes require that the translations be changed. The translation portion of the tape may be modified by writing MAAP or attendant-entered changes contained in main memory onto the tape cartridge.

4.05 The processor uses a data channel (LC570B) to interface with the minirecorder. The data channel (Fig. 11) is used to transfer data and the following commands to the recorder.

- The Forward/Rewind commands are decoded to start the tape drive and select tape direction.
- The Read/Write commands are decoded to start the tape drive and select the read or write mode.
- The drive stops after completion of a command.

4.06 The minirecorder uses the data channel to transfer data and the following status indication to the processor.

- Drive not connected.
- Cassette not in place.
- Cassette is write protected.
- Tape is too short.
- Checksum error.
- Block size larger than 2K.

4.07 To initiate the data channel connection, the IOE05 and MA00 and MD11 signals (Table B) are decoded by the data channel control circuit to shift the EIA in/output multiplexer to the tape mode of operation. This places a high on the data set ready (CD) lead and data channel write (BA) lead to the tape unit interface. The tape unit responds over the receive data (BB) lead.

4.08 Signal RDI0* or WRI0* in combination with MA00* and IOE05* generates signal IODN* (Fig. 11). Signal IODN* low indicates that the control bus command (RDIO*/WRIO*) has been initiated at the addressed interface. (See I/O processing in Part 6.)

FUNCTIONAL OPERATION

4.09 Reading from and writing to the minirecorder uses the interface data handling circuits which are listed as follows:

- Processor bus interface and data buffer register
- Transmitter—a parallel-to-serial converter
- Receive—a serial-to-parallel converter
- Write data selector.

4.10 During operation, data is received via the BBT lead in a serial mode at a rate of 9600 baud. The T I/O data register converts the data to a parallel mode. A high placed on the data received (DR) lead of the T I/O control circuit signals the processor that the data is ready to be read.

4.11 The process of sending data to the tape unit is functionally the reverse of receiving data. A high on the transmitter buffer empty lead of the S

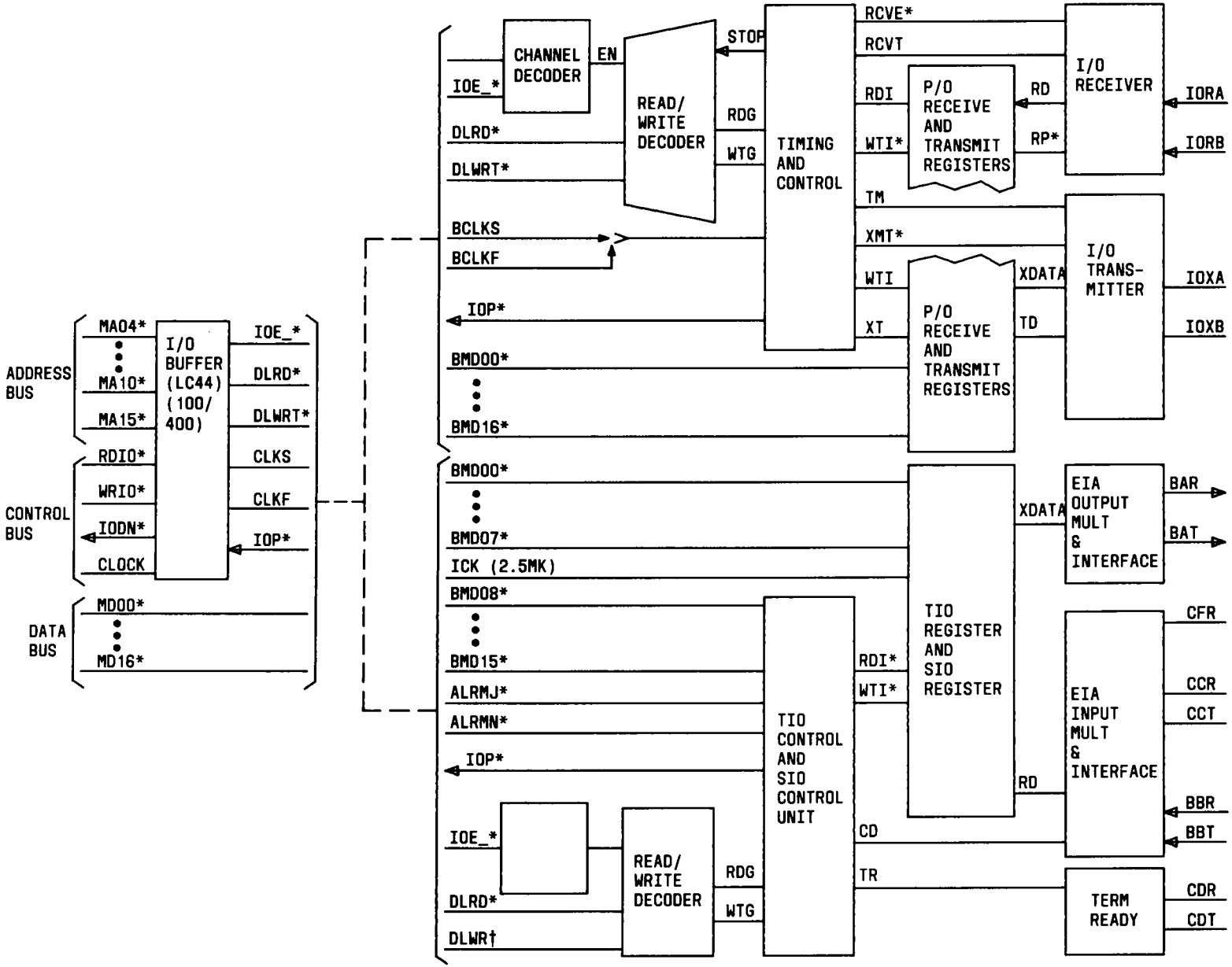


Fig. 11—Special Data Link (LC570B)

TABLE B

(I/O PERIPHERAL) ADDRESS SELECTION FOR "DIMENSION" "PRELUDE" PBX

MAIN ADDRESS BUS (MA__*)								ENABLE LEAD	INTERFACE CIRCUIT ENABLED
15	10	09	08	07	06	05	04		
0	1	1	0	1	1	1	0	MA00	Tape Interface, Command & Signaling
0	1	1	0	1	1	1	1	MA00*	Tape Interface, Data Transfer
1	0	0	0	1	1	1	1	IOE00*	Data Channel A, Slot 30
1	0	0	0	1	1	1	0	IOE01*	Data Channel B, Slot 30
1	0	0	0	1	1	0	1	IOE02*	Data Channel A, Slot 31
1	0	0	0	1	1	0	0	IOE03*	Data Channel B, Slot 31
1	0	0	0	1	0	1	1	IOE04*	Data Channel A, Slot 32†
1	0	0	0	1	0	1	0	IOE05*	Data Channel B, Slot 32
1	0	0	0	1	0	0	1	IOE06*	Data Channel A, Slot 33
1	0	0	0	1	0	0	0	IOE07*	Data Channel B, Slot 33
1	0	0	0	0	1	1	1	IOE08*	Data Channel A, Slot 34
1	0	0	0	0	1	1	0	IOE09*	Data Channel B, Slot 34
1	0	0	0	0	1	0	1	IOE10*	Data Channel A, Slot 35
1	0	0	0	0	1	0	0	IOE11*	Data Channel B, Slot 35
1	0	0	0	0	0	1	1	IOE12*	Data Channel A, Slot 36
1	0	0	0	0	0	1	0	IOE13*	Data Channel B, Slot 36
1	0	0	0	0	0	0	1	IOE14*	Data Channel A, Slot 37
1	0	0	1	—	0	0	0	IOEFF*	Loop Test, LC44, Slot 29

* Indicates lead is active in the low state.

† LC171B circuit pack may replace the LC34B in slot 32 when the RMATS option is utilized.

I/O control circuit indicates to the processor that the S I/O data register is ready to receive data. The data is converted to a serial mode by the S I/O data register then transmitted via the BAT lead at a rate of 9600 baud.

4.12 The tape cartridge is used to maintain a permanent system record of the translations which are peculiar to the particular installation and its feature package.

4.13 The processor has access to two main programs which allow it to communicate with the tape unit—a bootstrap loader (contained in main memory) or the run tape program. During a system initialization, the microcode forces the processor to the beginning address of the bootstrap loader. The bootstrap loader consists of a sequence of instructions which determine the procedure for loading the tape into main memory. The bootstrap loader initiates the loading process from a predetermined tape location.

4.14 The tape monitor program is an on-line (contained in main memory) program to the tape unit which controls the read and write operations. This tape monitor program, while being executed by the processor, generates the address, control, and enable signals. These signals are necessary for monitoring the status of the tape unit and controlling the transfer of data between the processor and minirecorder.

SOFTWARE STRUCTURE

4.15 The PBX programs are structured in a top-down hierarchy (Fig. 12). The software structure is controlled via real-time executive programs. The EXEC-10 is given control every 10 milliseconds by the high-level interrupt unit (Fig. 7). This program collects digits from those lines which are in the originating state and performs digit sending and trunk scanning at 20-ms intervals.

4.16 The EXEC-25 receives control via the 25-ms (LEVEL 5*) interrupt and gives control to the task dispenser. The task dispenser is a real-time supervisory control program which dispenses control to various scheduled tasks in preprogrammed interleaved 25-ms intervals. Some of these tasks include line scanning, MAAP button scanning, and the performing of certain maintenance tasks as controlled by the maintenance task dispenser (M_TSK DSP). Normally, the maintenance task dispenser does not perform any tasks unless craft personnel or RMATS facility is performing maintenance on the PBX.

4.17 The MAAP control and numeric buttons are scanned every 25 ms by MAAP SCAN. Valid button pushes are sensed and the encode for that button is stored in a stimulus buffer. The stimulus buffer is a 64-word read/write storage located in main memory.

4.18 The maintenance task dispenser receives control once each 25-ms interval and schedules four tasks as depicted in Fig. 13. The page monitor translates a MAAP procedure number entered by craft personnel into a page number. The MAAP control routine performs various functions as listed below:

- Updates the MAAP displays.
- Checks the stimulus buffer table to determine whether the table is full. If the stimulus buffer table is full, a WAIT lamp on the MAAP is energized.
- Checks status of the page monitor program, and if necessary, sends a page request to the tape monitor program.

4.19 The tape monitor transmits the page request to the tape unit. The tape unit reads the tape directory block, calculates the physical location of the nonresident program, and transmits the designated program to the paging buffer. A program, such as traffic measurement, may consist of up to 1K words of main memory, the paging buffer limit. A 1K-word block of data consists of four 256-word data blocks. The paging buffer is a multiple use read/write area located in main memory. The maintenance task dispenser gives control to the active (paged in) nonresident program once each 25-ms interval.

4.20 Execution of programs scheduled by the task dispenser, with the exclusion of base level maintenance routines, is known as call processing. The time required to perform call processing is variable because of the dynamic nature of customer requirements and changing traffic demands and can change from one 25-ms period to another 25-ms period. Figure 14 depicts an arranged task schedule and shows typical time usage. During periods of heavy traffic, time for lower priority tasks may be preempted for higher priority tasks.

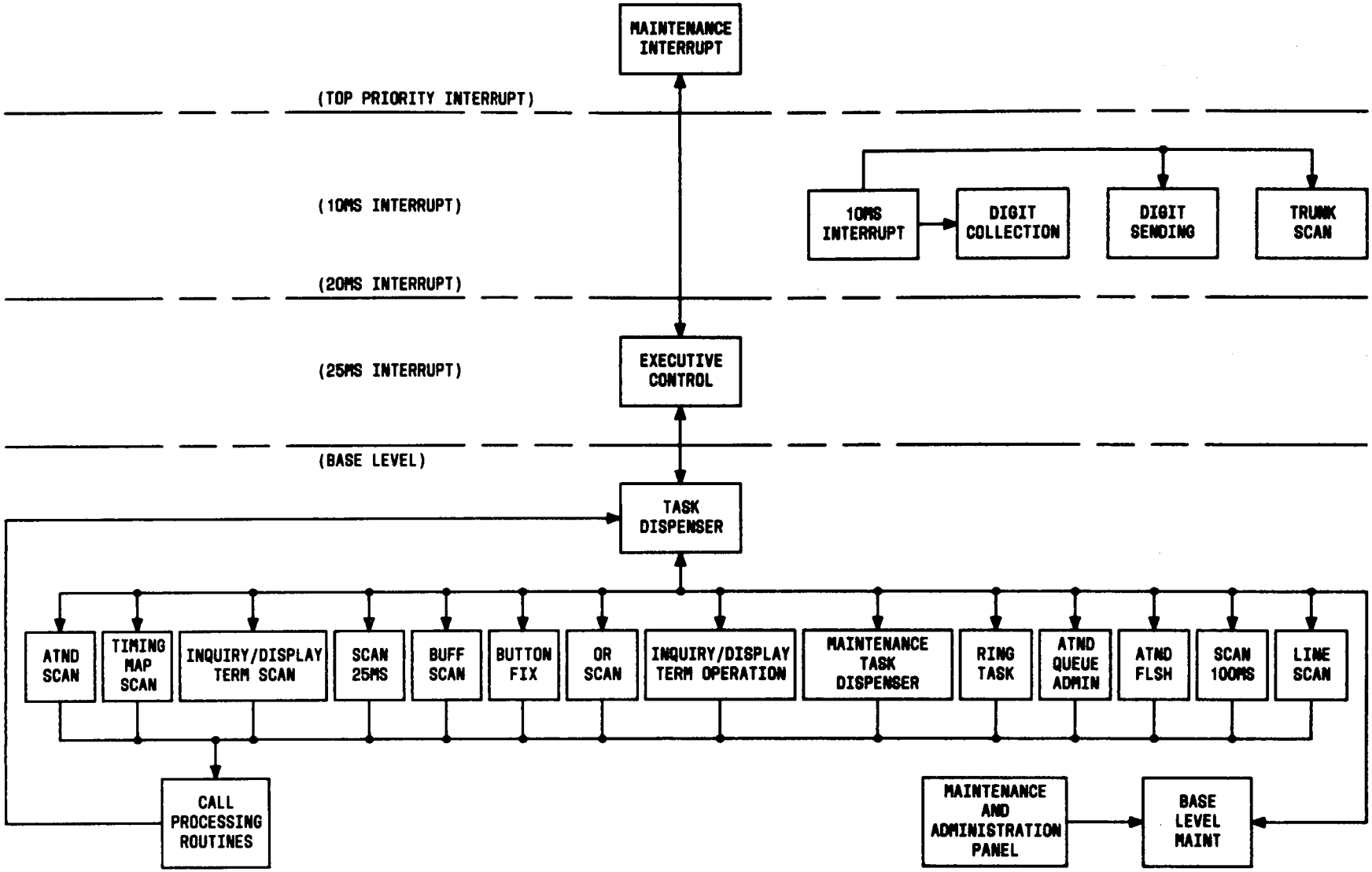
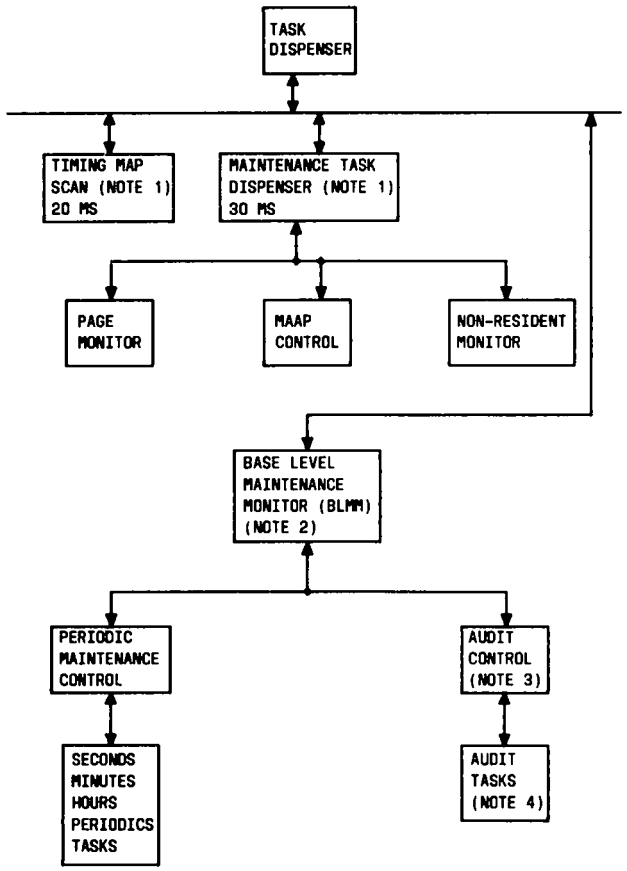


Fig. 12—Typical DIMENSION PRELUDE PBX Software Structure



NOTES:
1. THESE TASKS ARE SCHEDULED AT EVERY 25 MS INTERVAL.
2. BLMM IS GIVEN CONTROL BY THE TASK DISPENSER ONLY AFTER ALL CALL PROCESSING IS FINISHED.
3. AUDIT CONTROL IS THE LAST CONTROLLER TO BE EXECUTED UNDER BLMM FOR EVERY BLMM CYCLE.
4. AUDIT TASKS ARE SCHEDULED IN A CYCLIC (ROUND ROBIN) PATTERN.

Fig. 13—Maintenance Task—Block Diagram

4.21 Certain maintenance and periodic tasks are scheduled by the base level maintenance monitor (BLMM). If call processing requirements are such that these maintenance routines are not executed within a specified period of time, a fixed work schedule is generated, ensuring that base level maintenance is performed. The maintenance tasks include resetting the sanity timer, executing error checking routines which perform audits on the system hardware, etc (Fig. 15). The execution of base level periodic tasks is controlled by timing maps which are maintained in software. The timing maps provide signals as to when the seconds, minutes, and hourly periodic tasks and audits are to be performed.

4.22 Data may be read from the tape cartridge during system initialization or during a verify operation. Verify is one of many error detection schemes built into the tape monitor program. Verify is performed via the tape monitor after translations have been written onto the tape cartridge. It consists of a word-by-word comparison between those translations in main memory and those written on the tape cartridge.

4.23 Data may be written onto the tape cartridge in the form of translation changes. Translation changes or updates may be made by either an attendant when altering station restrictions or by

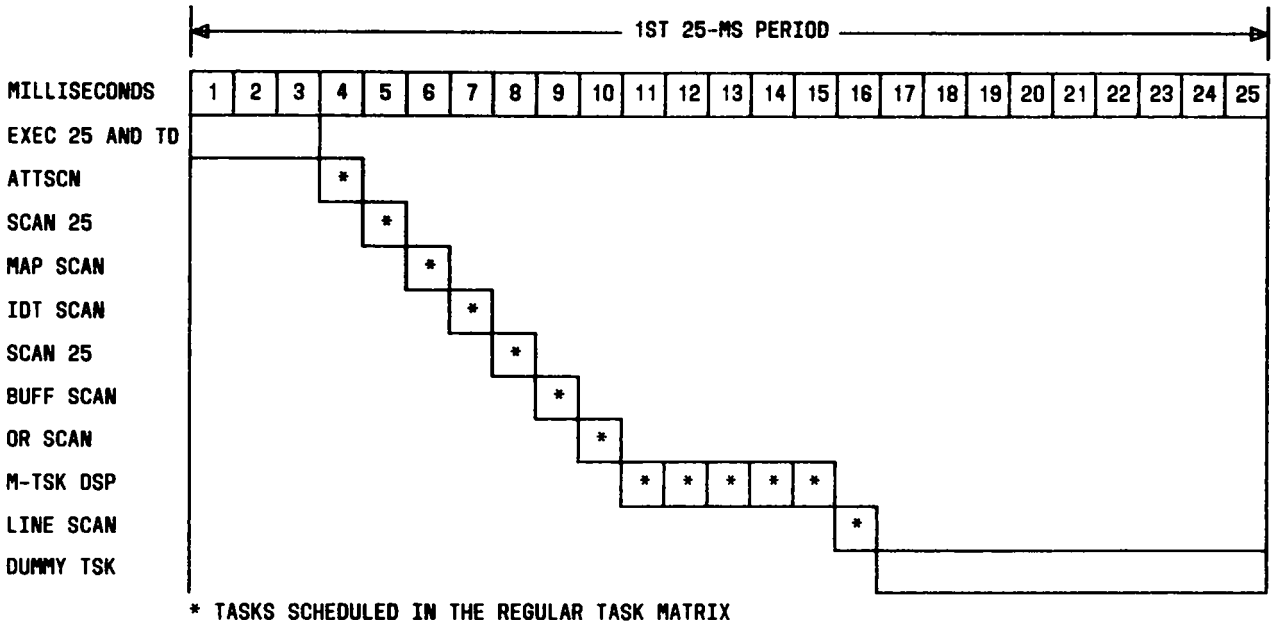


Fig. 14—Call Processing Timing Interval (Typical)

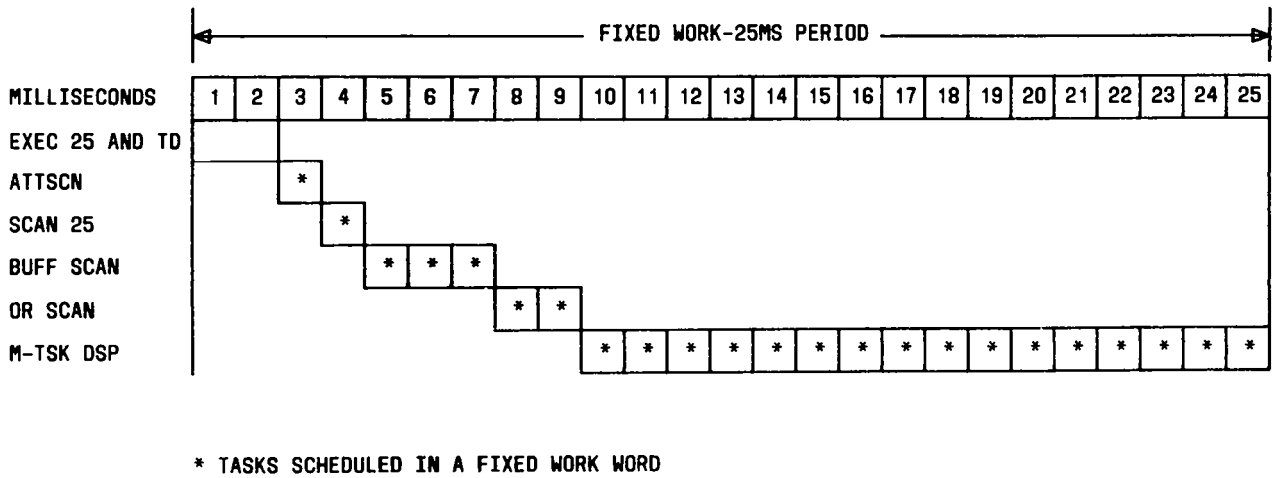


Fig. 15—Typical Timing for a 25-ms Interval With Heavy Traffic Demands

craft personnel making feature, line, or trunk class-of-service changes. A translation mover routine relocates existing translation memory whenever it is necessary to change call processing programs to incorporate new features or to add major line and trunk growth on an existing system.

5. MAIN MEMORY

5.01 Main memory consists of RAM and PROM contained on circuit packs located in the control trunk carrier (Table A). The PROM is contained on LC555 (FP14) or LC595 (FP16) and is read only. The RAM is contained on LC556 (FP14) or LC594 (FP16), is of the static type, and may be read from or written to.

5.02 The PROM functions as a storage for generic code and administration and maintenance data. The processor uses its common control buses for reading data from the PROM memory.

5.03 The primary function of RAM is to provide a rapid and easily accessible read/write storage facility for the processor. This random access storage provides a means for storing system control and call processing programs, translation parameters, and transient data which is used for completing and disconnecting network connections. The processor uses its common control buses for transferring data to and from the RAM memory.

READ/WRITE OPERATION

5.04 The memory control and alarm circuit LC557 (Fig. 16) functions as an interface between the processor and memory and is physically connected to the common control buses. The processor uses the common control buses and the memory control circuit for transferring to/from RAM or reading from PROM. Selecting a particular location, for reading from PROM or reading from/writing to RAM, requires that:

- A memory board within the common control carrier be enabled
- The 16-bit address be decoded, thus selecting a memory word
- A READ*/WRITE* command be generated.

5.05 Memory boards may be enabled by the processor or by the SWAT interface decoding the XM0* and XM1* leads. The SWAT interface is enabled by a low on the DPROM* lead.

5.06 The area of memory to be read is determined and is activated by the board select logic circuit on the LC557B. The area in memory is determined by decoding main address leads MA10*-MA15* and is activated via the BDSEL0*-BDSEL3* leads.

5.07 The board select logic is also controlled by the GID*, INSREQ, and IOE15* leads. A high on the INSREQ lead indicates the data stored in the addressed location is an instruction. A low on the GID* lead indicates the instruction is contained in two words. A low on the IOE15* lead activates the data latches to the I/O ports. The timing logic on the LC557B signals the processor that the memory area, board select, and read instructions have been received and processed by placing a low on the MCMPT* lead.

5.08 The data buffer serves as a temporary storage for data received from or transferred to the processor via the MD00*-MD16* leads or for data received from or transferred to a memory location via the DA00*-DA16* leads.

5.09 The data flow is controlled by the read/write and timing control circuits. Timing is provided by the 20-mHz clock and a low on the READ* lead or the WRITE* lead activates the read/write control circuit. The timing control circuit strobes the 16-bit memory address into memory latches on the selected memory board, via the ADSTB* lead. The MCMPT lead signals the processor when the memory transaction has been completed.

5.10 The procedure for executing a write command requires that the RAM memory be addressed, data be strobed onto the main data bus, and control bus signal WRITE* be generated. Signal WRITE* (indirectly) gates the 17 bits of data through the input data buffer and timing circuit. The 17 bits (DA00 through DA16) are subsequently stored in the addressed memory location.

6. INPUT/OUTPUT (I/O) CIRCUITS

6.01 The processor executes programmed instructions and uses its common control buses to transfer data to and from the interface circuits. Interface circuits are the electronic connection between

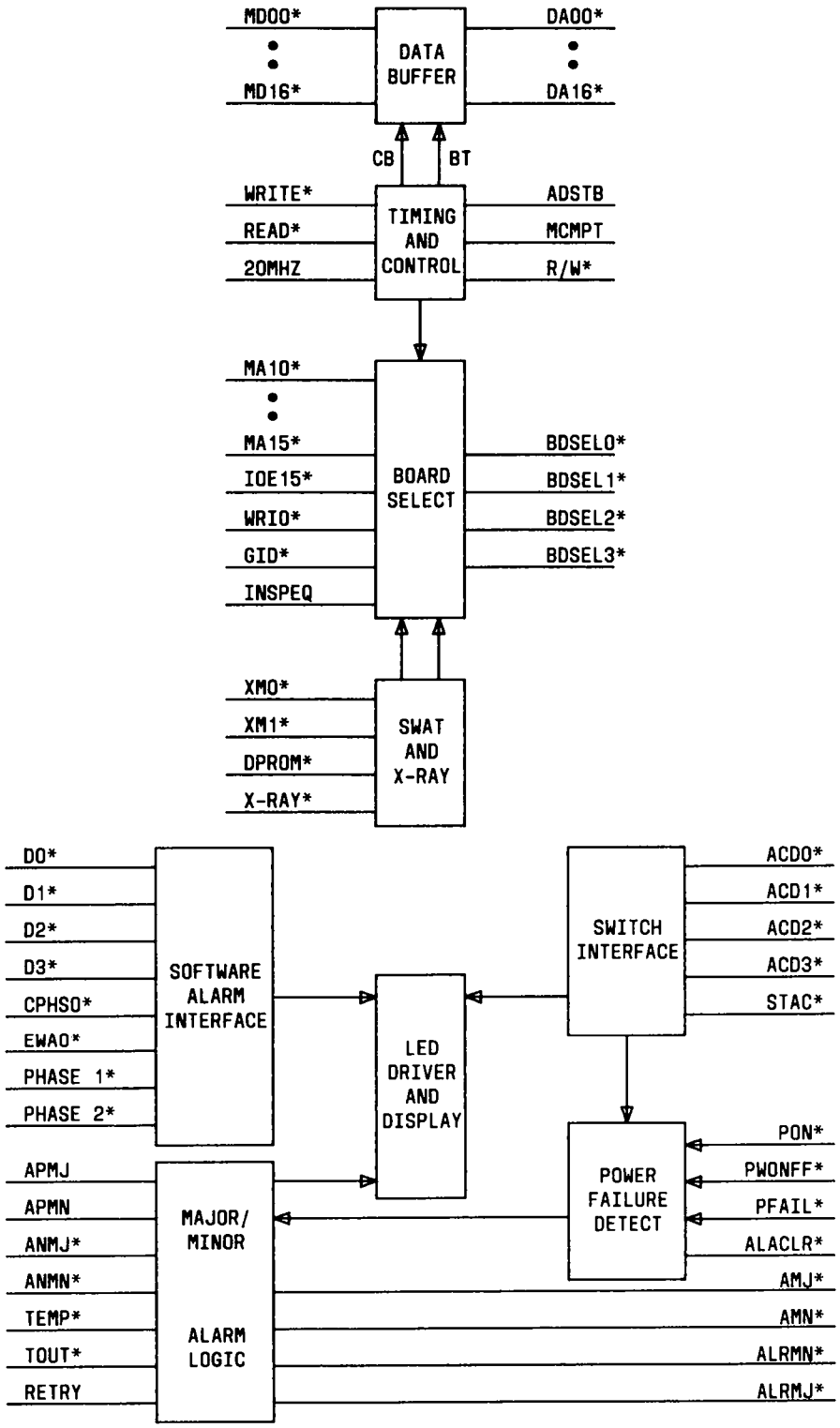


Fig. 16—Memory Controller and Alarm Circuit (LC557)

the processor, network, and I/O peripherals. These include the minirecorder, data channels, MAAP, Remote Maintenance, Administration, and Traffic System (RMATS), etc.

6.02 All peripherals are treated as addressable I/O locations. The processor selects an I/O channel (ie, LC570) via the main address bus and extended I/O buffer. Read/write commands allow data to be transferred via the main data bus. The WRI0* command generates an enable signal (DLWRT*) which gates data from the data bus into an addressed interface circuit.

6.03 Table B shows the I/O address and enable signal leads. Note that the extended I/O buffer is inhibited when MA15* is low and the tape interface is addressable by MA12* and MA13*. The tape interface is enabled for signaling and commands when MA00* is a logic low. Data is transferred when the addressed tape interface receives MA00* active high. Data bus inputs are hard-wired to all I/O circuits as required.

6.04 Certain maintenance oriented functions are periodically performed on the I/O channels. Loop-around testing, a maintenance function, is performed when the common control places the system in the loop-test mode. A loop-test is performed when the processor addresses the writes to an I/O data channel and then generates the loop-test (LTST*) signal, which disables the addressed data channel and allows the processor to read the data previously written to the addressed I/O channel.

6.05 Timing and clock signals (CLKF, CLFS) are distributed to the I/O peripherals. An I/O acknowledge (IOP*) signal is the indication that a RDIO*/WRIO* command has been completed. Signal IOP* is converted by the I/O buffer into signal IODN* which signals the processor, indicating that the command has been performed. An IODN* may also be generated by the bit-swap circuitry, write protect circuitry, minirecorder interface, etc.

I/O PROCESSING

6.06 From the processor perspective, the transferring of data to and from a peripheral is basically a 2-part (interrogate) operation. The processor must first address a peripheral interface (ie, LC554 slow-speed data channel connected to an attendant console) and then execute a pair of EPL-3 instructions sequentially (Fig. 17). When the processor executes a send I/O (SIO) message, 16 bits of data are sent to the addressed peripheral interface. Once data is gated into the addressed peripheral interface, it generates signal IODN*, indicating that the interface has received the data. The processor is then free to execute other instructions, but is programmed to execute take I/O (TIO) messages for a set time, checking to see if status information is available from the attendant console.

6.07 Once the interface generates signal IODN*, the interface functions autonomously and transmits the 17-bit data word bit-synchronously (full duplex), meaning that the peripheral unit (at-

SIO SEND I/O MESSAGE

P	BA	OPCODE = 27*	R1	R2

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TIO TAKE I/O MESSAGE

P	BA	OPCODE = 26†	R1	R2

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

* ONE WORD INSTRUCTION USED TO SEND DATA TO THE ADDRESSED I/O DEVICE.
† ONE WORD INSTRUCTION USED TO TAKE DATA FROM THE ADDRESSED I/O DEVICE.

Fig. 17—EPL-3 I/O Instructions

tendant console) echoes received commands or status back to its interface (LC554) synchronously. Since the slow-speed data channel transmits and receives data serially over a transmission line, there is a time delay which consists of the time required by the interface to transmit and receive 17 bits in full duplex plus all circuitry rise/fall times and transmission propagation delays (Fig. 18).

6.08 Returned status information is the condition of buttons (button pushes) on the console. The processor executes read commands (TIO instructions) at the interface and, upon receiving control bus signal IODN*, takes the data, checks to see if it is the peripheral response (status information) and, if not, repeatedly rereads until status information is obtained. Status information from one scan period is compared with that from a subsequent period and validated. All changes (button pushes) must exist for two subsequent periods to be processed.

6.09 The time from when the processor executes an SIO instruction to the time when the interface responds with IODN* is T-T1 (Fig. 18). This time is typically around 19 μ and cannot be greater than 27.9 μ. The propagation delay consists of the time T1-T2. This time is typically around 100 μ. The tape cartridge provides for a maximum of 160 μ. This allows the attendant console to be placed a greater distance away from the PBX with repeaters in series.

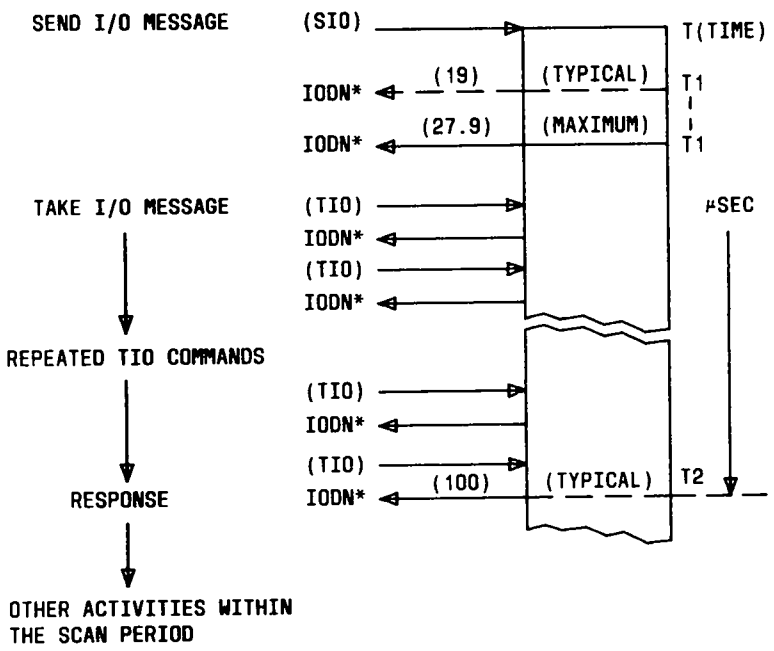


Fig. 18—Data Channel Time Usage Propagation Delay (Low-Speed)