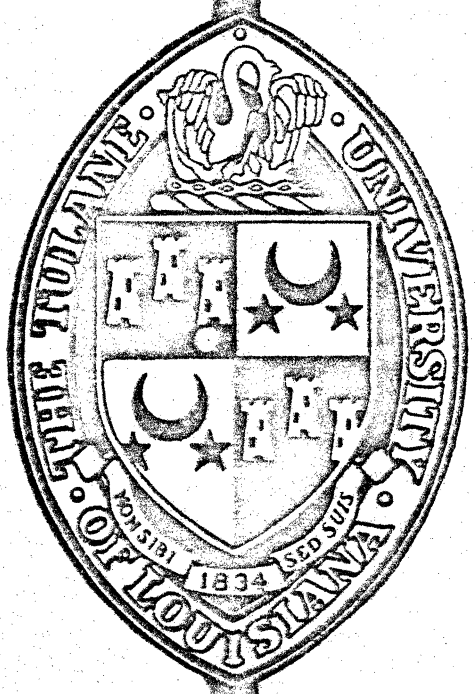
PROCEEDINGS OF THE THIRD MEETING OF THE MINUTEMAN COMPUTER USERS GROUP



REPORT MCUG-3-71

Meeting held
July 19-20, 1971
Miami Beach, Florida

PROCEEDINGS*

OF THE

THIRD MEETING

OF THE

MINUTEMAN COMPUTER

USERS GROUP

Meeting held

July 19-20, 1971

Miami Beach, Florida

Report MCUG-3-71

*Supported in part by *ARMY MEDICAL R&D COMMAND* under Contract DADA 17-71-C-1019 and by *NSF OFFICE OF COMPUTING ACTIVITIES* under Grant GJ-850.

TABLE OF CONTENTS

*Papers presented by Dr. Charles H. Beck, Professor of Electrical Engineering, Tulane University.

MINUTEMAN D17B COMPUTER PROCUREMENT

Approximately 800 Minuteman D17B computers are expected to be declared excess by the USAF through 1974. The original acquisition cost per system was approximately $234,000. These computers can be acquired by qualified agencies, contractors, and grantees as the systems become available through appropriate ADPE reutilization agencies on an "as is" non-reimbursable basis as follows:

DoD Agencies

Contact respective service Hqs. for ADPE Acquisition for approval and for forwarding of Requisition Form 1419 to DARO.

DoD Agency Contractors and Grantees

Contact respective contracting officers for approval and for forwarding of Form 1419 to Defense Supply Agency, DSAH-LSR/DARO, Cameron Station, Alexandria, Virginia 22314.

Civil (Non-DoD) Agencies of the Federal Government

Contact respective Office for ADPE Acquisition for approval and for forwarding of Transfer Order Form 122 to GSA Excess Equipment Utilization Branch, Crystal Mall Bldg. 4, Washington, D.C. 20406.

Civil Agency Contractors and Grantees

Contact respective contracting officers for approval and for forwarding of Form 122 to GSA as listed previously.

Authorized Donees

Contact respective state surplus property offices for acquisition through DHEW Office of Surplus Property Utilization, 4452 DHEW North Bldg., Washington, D.C. 20201.

PART 1. MINUTEMAN D17B COMPUTER DESCRIPTION

Functional Capabilities and I/O

The D17B is a small general-purpose computer. It is completely programable and has the capabilities of: receiving and sampling analog signals, digital data, or pulse-type input signals; logical decision-making and performance of arithmetic operations using an instruction repertoire of 39 machine language instructions; and the transmission of output data in the from of analog, digital and pulse type signals under program control. Figure 1 is a functional block diagram for the D17B which shows how it is possible for the computer to perform the operations described above. Because of the extremely flexible I/O capability of the D17B, it can be quite useful in a control application.[1-8]

Central Processing Unit and Control

Since the D17B is a serial-binary computer, simultaneous access to all the bits of a memory locations is not needed either for instructions or data. Hence, the arithmetic registers need not be constructed entirely of flip-flops. Instead, they are in the form of circulating loops in memory as illustrated in Figure 2. The D17B has four double-rank arithmetic registers which are Accumulator (A), Lower Accumulator (L), Instruction Register (I), and Number Register (N). Because the L-register is addressable, it can be used as rapid-access storage in addition to performing normal arithmetic functions. There are two non-addressable arithmetic registers, the I-and N-registers, which are used without programmer control and one 3-bit pseudo-index (phase) register.

The central processing unit (CPU) has I/O access to four rapid-access memory loops of 1,4,8, and 16 words in addition to the main memory which is

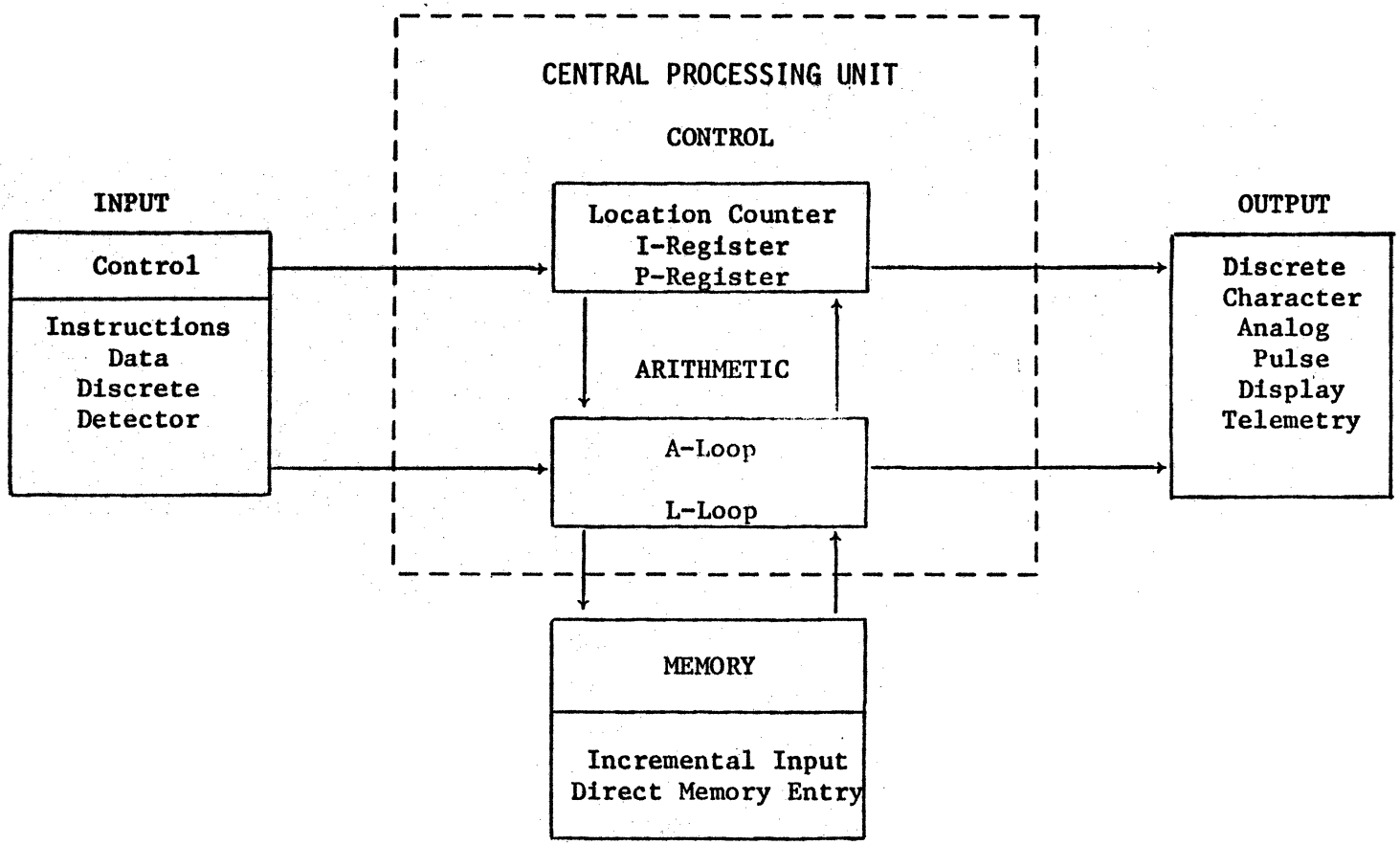


Figure 1. Minuteman D17B computer functional block diagram (conceptual).

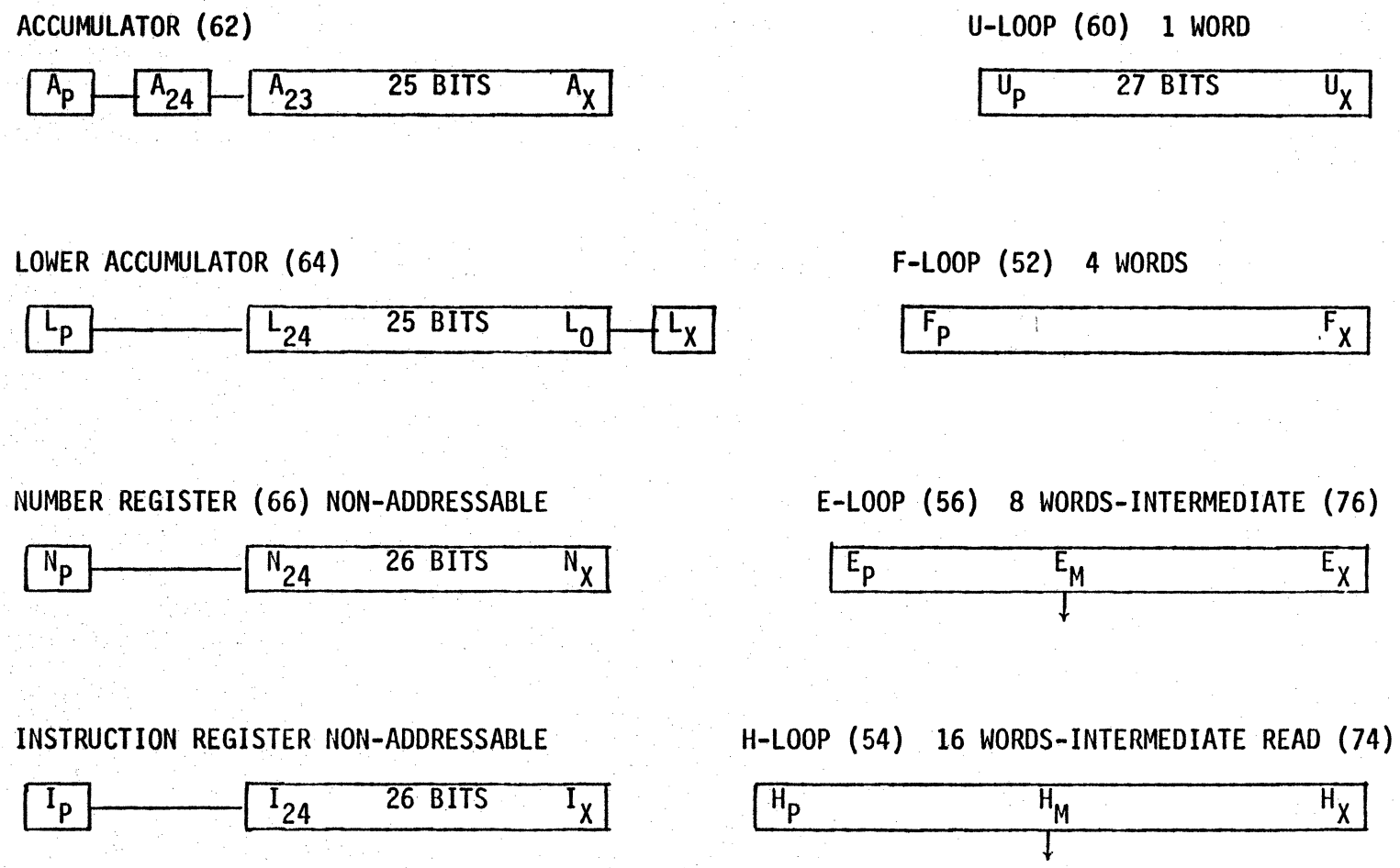


Figure 2. Arithmetic registers and rapid-access memory loops.

arranged in 21 channels of 128 words each. Two input buffer loops of four words each provide additional input capability to memory.

Programmed data channels cause data transfers into the arithmetic registers. All machine functions are processed and interpreted in the CPU. The memory channel address from which the next instruction is to be taken is determined by the location counter. When the CPU is ready to accept another instruction from memory, the address is specified by the channel address stored in the location counter and the sector address specified in the previous instruction.

The phase register can modify the operand address of one of the multiply instructions. This register also serves as a selector switch for choosing one of two pairs of inputs to one of the incremental pulse-type input loops and for selecting one of four external positions for each of the three D-A analog voltage outputs.

The Accumulator holds the results of all arithmetic operations and serves as an output register for parallel digital data, pulse-type signals, D-A analog voltage outputs, and telemetry data. The Lower Accumulator is involved in certain arithmetic, input, and logical operations. A real-time clock is provided by internal timing signals derived from the clock channel of the disc memory.

Specifications [9,10]

The D17B is basically composed of two semi-circular sections. One half contains the power supply circuit cards which generate the various d-c voltages required in the computer and a 400 Hz 3ϕ signal for the 6000 rpm disc memory. The other semi-circular section contains the discrete DRL and DTL logic components of the computer itself. Detailed specifications for the D17B computer are given in Table 1. Figure 3 shows a sketch of the computer section.

TABLE 1. MINUTEMAN D17B COMPUTER SPECIFICATIONS

Manufacturer: Autonetics, a division of North American Rockwell

Model: D17B

Year: 1962

Type: Serial, Synchronous

Number System: Binary, fixed point, 2's complement

Logic Levels: 0 or False, 0V; 1 or True, -10V

Data Word Length (bits): 11 or 24 (double-precision)

Instruction Word Length (bits): 24

Maximum I/O (words/s): 25,600

Number of Instructions: 39 types from a 4-bit op code by using five bits of the operand address field for instructions which do not access memory.

Execution Times:
- Add (us): 78 1/8
- Multiply (us): 546 7/8 or 1,015 5/8 (double precision)
- Divide: (Software)
- (Note: Parallel processing such as two simultaneous single precision operations is permitted without additional execution time.)

Clock Channel: 345.6 KHz

Addressing: Direct addressing of entire memory
Two-address (unflagged) and three-address (flagged) instructions

Memory: Word Length (bits): 24 plus 3 timing
Type: Ferrous-oxide-coated NDRO disc
Cycle Time (us): 78 1/8 (minimal)
Capacity (words): 5,454 or 2,727 (double precision)

Input/Output: Input lines: 48 digital
Output lines: 28 digital
12 Analog
3 Pulse
Program: 800 5-bit char/s

Physical Characteristics:
- Dimensions: 20" high, 29" diam.
- Power: 28V DC at 19A
- Circuits: DRL and DTL. Double copper clad, gold plated, glass fiber laminate, flexible polyurethane coated circuit boards

Software: Minimal delay coding using machine language
modular special-purpose subroutines

Reliability: 5.5 years MTBF

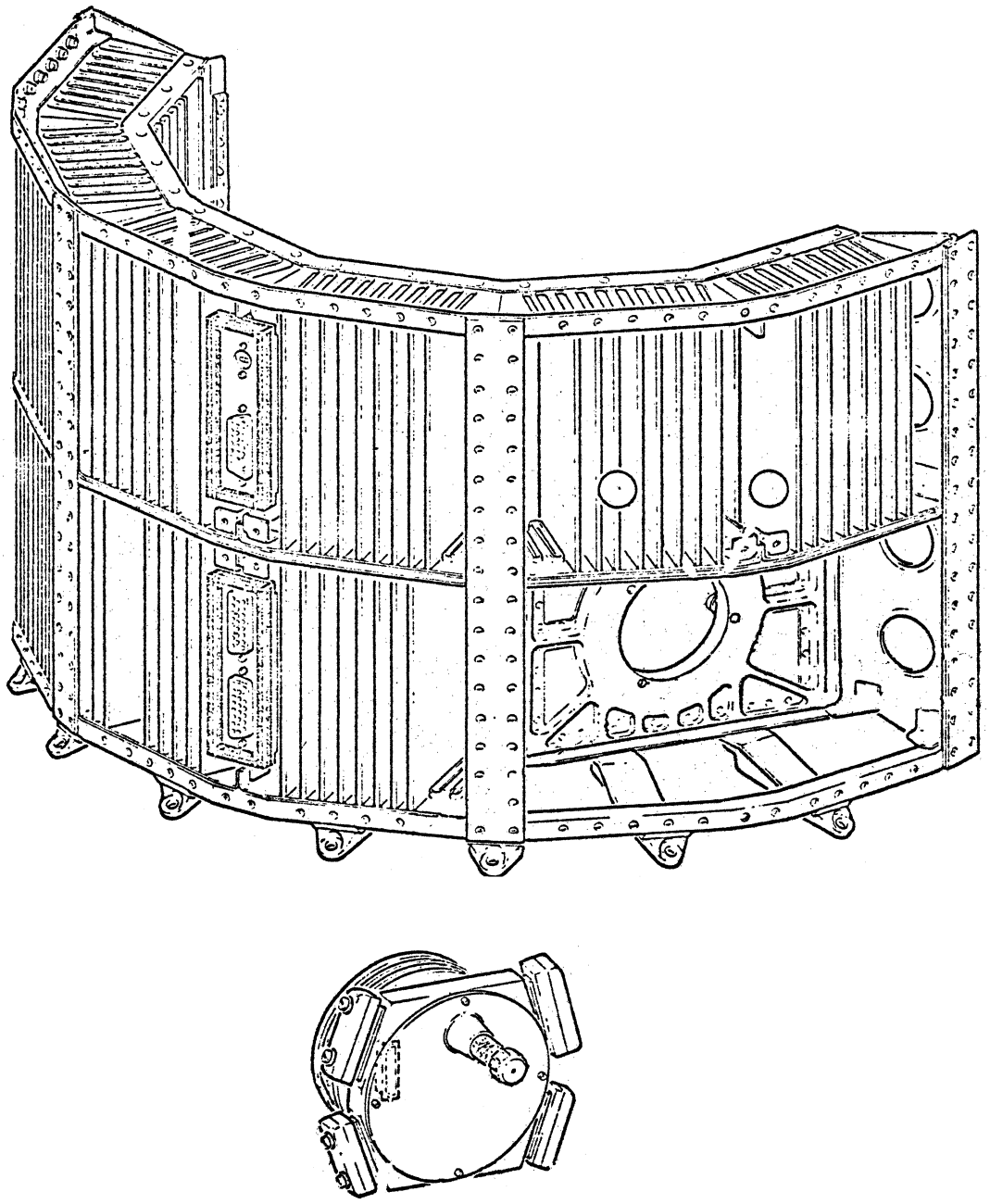


Figure 3. MINUTEMAN D17B MINICOMPUTER SKETCH

References

1. J. Cohen, "Mini-computers," Modern Data, vol. 2, No. 8, 1969.

2. A. Epstein and D. Bessel, "Minicomputers are made of this," Computer Decisions, pp. 10-22, August 1970.

3. E. D. DeCastro, "What can you do with a minicomputer," Industrial Research, November 1969.

4. F. Gruenberger, "Are small free-standing computers here to stay?" Datamation, vol. 12, pp. 67-68, April 1966.

5. W. H. Roberts, "Minicomputer architecture," IEEE Computer Group News, vol. 3, pp. 5-9, July/August 1970.

6. E. Holland, "Minicomputer 1/O and peripherals," IEEE Computer Group News, vol. 3, pp. 10-14, July/August 1970.

7. F. Coury, "A systems approach to minicomputer 1/O," Proc. AFIPS 1970 SJCC, 1970.

8. R. Rinder, "The input/output architecture of minicomputers," Datamation, vol. 16, 119-124, May 1970.

9. D. O. Baechler, "Aerospace computer characteristics and design trends," IEEE Computer Group News, vol. 4, pp. 46-57, January/February 1971.

10. R. T. Ollivier, "A technique for selecting small computers," Datamation, vol. 16, pp. 141-145, January 1970.

PART 2. MINUTEMAN D17B COMPUTER APPLICATIONS

Introduction

Although the D17B does not provide all the desirable features of large general-purpose machines, it does resemble them functionally and it possesses a number of similar features. It is a versatile multi-purpose computer capable of solving a wide range of problems; however, it has limited capability both in storage capacity and computation speed. Unlike the large general-purpose computer which is designed to efficiently process many different programs, the multipurpose D17B is better suited to dedicated or fixed tasks that can be served effectively by economical use of the available memory and speed of execution.[1-7]

To consider potential uses of the D17B computer, it is necessary to categorize various uses for general-purpose electronic digital computers, then to compare them on the basis of degree of similarity to the design purpose of the D17B. Some general applications which coincide with the design of the D17B are listed below:

A. Direct machine perception to, interpretation of, and reaction to physical conditions.

B. Integral to a command/management system uniquely and specifically designed for the control of on-going operations.

C. Mathematical computation and transformations in support of basic research and technical aspects of applied research, development, test, and evaluation in connection with scientific and engineering work.

Computing and Control Applications

The D17B, like commercial minicomputers with small memories, is not well suited for general-purpose computing when compared to a large computer. General-purpose computation in minicomputer terminology refers to stand-alone operation. Some minicomputers are used in stand-alone applications such as control, data acquisition, communication concentrators and processors, peripheral controllers and preprocessors for large computer systems, display controllers, buffer memories, bio-medical monitoring, automated testing, automated instrumentation and telemetry.

In a practical sense, the capability for general computing is determined by the ability to perform a large variety of calculations. This is determined basically by the instruction set. Available subroutines simplify the programming, and assemblers and compilers simplify the task further. The goal in providing general-purpose software for the D17B is to minimize the amount of time, effort, and knowledge required for a user to arrive at a point of useful return for his investment in development of the D17B. But, generality always comes at a price. The D17B is limited at present to a small number of real-time, machine language programs and subroutines.

The D17B is well suited for control applications because of its flexible input/output ability. In terms of a control application the D17B has the following capabilities:

1. Sampling and processing of input data in the form of control signals, digital data, or pulse-type signals.
2. Logical decision-making and performance of arithmetic operations using an instruction repertoire containing 39 types of machine language instructions listed in Table 1 of Part 6.

3. Transmission of output data in the form of analog, digital, and pulse-type signals under program control.

Computer control applications may include monitoring and data processing, start-up and shut-down procedures, and optimal control. The main attributes of computer control are computational speed, storage capability, and decision-making ability. If sufficient computational speed is available, optimal control can be accomplished. The storage capability provides for economical and efficient data recording and processing. Decision-making ability provides the capability for direct digital control.[8-19]

Logical decisions and constraints can be employed in computer control, and the results of intermediate calculations and control actions can be recorded to produce a historical file. The general-purpose capabilities of the D17B permit the control program to be modified and expanded within the limits of memory capacity to fit system growth, new instruments, or changing control policy.

At present, the D17B is being used for waveform analysis in connection with an AutoAnalyzer. The computer performs a self-checkout of the system and then samples the signal from the AutoAnalyzer to determine the peak concentrations.[20]

If the D17B is to be used for control computing applications, it must be capable of not only performing control calculations, but a number of other essectial functions also. For example, raw input data are generally subjected to individual limit checks to detect instrument failures or out-of-normal conditions, averaged or smoothed to minimize the effects of random variations, and then recorded or used in calculations. As a typical example of a limit check in terms of D17B instructions, the following could be executed:

1. DIA - data input to the Accumulator

2. MIM - replace the contents of the Accumulator by the negative of the present magnitude of the contents of the Accumulator
3. ADD - add the limit tolerance to the contents of the Accumulator
4. TMI - transfer on minus

These four instructions would accomplish the limit check by performing a conditional branch. Similar operations could be equally useful for general or special-purpose computing.

It is appropriate that the D17B be considered for dedicated control applications involving control over a single unit or a limited portion of a process. Such an application may not only be appropriate considering the limited memory and execution speed of the D17B, but the system reliability consideration makes the D17B ideally suited to such tasks. Process-wide control may require several interconnected D17B's. The real-time aspect of control applications is compatible with the current requirement of machine language programming for the D17B.[21]

The versatility available with a computer control system involving a general-purpose computer is an important consideration. As new instruments are added and as knowledge of a process increases, better control policies can be developed. Hence, control programs are constantly in need of change. Also, the characteristics of the process will often change as its operation is improved through computer control. Because of these factors, the programmable feature of the D17B is extremely desirable as well as its flexible I/O capabilities, which can accommodate a variety of control devices. The D17B can provide digital, pulse-type, and analog output signals under program control for manipulating process variables. This flexible I/O capability provides for efficient interaction between the D17B and the devices being controlled.

Certain special-purpose applications such as on-line digital data processing, computer interfacing, peripheral buffering, and data monitoring require very little CPU sophistication, limited arithmetic capability, and perhaps low-speed performance compatible with the D17B specifications. The dominant requirement of many special-purpose computer applications relates to the I/O architecture as is the case for control applications. The importance of I/O channels is particularly significant where data is being transmitted continuously between the computer and peripheral devices.

On-line digital data processing often requires that analog information be converted to digital form using an A-D converter. With the 24-bit double precision word of the D17B, the output from two 12-bit A-D converters can be inputted simultaneously under program control. The required speed of I/O transfers and arithmetic for special-purpose data acquisition can be much slower than for control applications because real-time analysis and control response commands are not necessary. Hence, the D17B is flexible enough to be used in these areas formerly requiring special-purpose computers. As requirements change, the D17B can easily be re-programmed. In such fields as medical research, biological studies, and experimental physics, the D17B can be programmed to control the monitoring, measuring, and recording of a variety of quantities such as pressures, flow rates, EKG, and heart rate. Automation of chemical laboratory instruments such as chromatographs, spectrometers, and AutoAnalyzers using the D17B also appears feasible. Calculation of desired parameters, recording of results and graphic display are appropriate applications areas for this computer. Simultaneous measurements of several quantities are possible through the use of sample-and-hold devices, a multiplexer, and an A-D converter.

A flexible, reliable, mobile data monitoring system can be developed using the D17B computer with interface to any of the following: operational amplifiers, sample-and-hold devices, multiplexers, A-D converters, digital voltmeters, counters, CRT displays, plotters, programmable signal generators and power supplies, transducers, and sensors. This combination will provide for the automatic testing of electronics components, IC, logic cards, complete logic assemblies, and other devices and circuits. Programmed transducer testing and high-quality data collection of signal characteristics such as amplitude, current, and phase which can be accomplished at high speeds have significant advantages over manual methods. These techniques are also applicable to non-destructive testing as employed in the inventory of aircraft parts based on the characteristics of the steel as represented by the electrical output of spectrometer-type instruments.

On-line communication is also an important applications area to be considered for the D17B. A data concentration buffer storage system for teletype and other low speed I/O devices can be developed. Programmed multiplexing of parallel information for serial transmission over a narrow-band communication channel is possible since the D17B can provide for changing the scan rate. Preprocessing for analysis and computation by a large-scale computer will also be an appropriate consideration.

Training Devices

There are numerous areas in which the D17B can be utilized as a training device. Because of its size and design, the D17B can be instructive in both hardware and software fields. Although the D17B is a small computer, it has a sizable set of 39 instructions which make programming a non-trivial operation. In addition to a large number of instructions, the D17B has some rather sophisticated software capabilities that make programming a challenge. Some

of these extra features include: flag store instructions, programmable I/O operations, and a fine countdown mode used for program timing. These features make the D17B useful as a trainer for teaching machine language programming techniques.

The D17B's value as a training device may be even more valuable in the hardware field than in the software field. First, it has proven effective as an introductory example in the discussion of machine organization because of its fairly straightforward organization. On the other hand, it could be used as an example of the degree of complexity of an advanced digital system. In this respect, the D17B can be analyzed on any level -- from the basic gate-flip-flop level up to the actual execution of each instruction. If one performs this analysis to the extent where each step of the operation is fully understood, the knowledge gained is far more meaningful than training with textbook material only. This step-by-step analysis has been performed by graduate students in the Systems Laboratory at Tulane University in the areas of both instruction execution and control of the D17B. The results provided, obviously, a far better understanding of the D17B, but also a much better feel for the principles and operations of a digital system.

In general, there is no doubt that the D17B makes an excellent training device in both the hardware and software fields. Of course, there are many more training applications than those mentioned here, some of which have probably not even been investigated yet.

References

1. G. Lapidus, "A look at minicomputer applications," Control Engineering, pp. 82-91, November 1969.

2. D. J. Theis and L. C. Hobbs, "Mini-computers for real-time applications," Datamation, vol. 15, pp. 39-61, March 1969.

3. M. J. Lowenstein, "The minicomputer: The machine with an endless future," Electronic Design, vol. 18, no. 9, April 1970.

4. R. E. Penczer, "Automation in data acquisition," American Laboratory, April 1969.

5. R. L. Hooper and L. D. Amdahl, "Trends in aerospace computers," Datamation, vol. 13, pp. 22-26, November 1967.

6. W. H. Long, "Invasion of the minicomputers," Automation, August, 1969.

7. A. S. Buchman, "Aerospace computers," Advances in Computers, vol. 9, pp. 239-284, 1968.

8. T. M. Stout, "Process Control," Datamation, vol. 12, pp. 22-27, February 1966.

9. R. J. Spinrad, "Automation in the laboratory," Science, vol. 158, p. 55, 1967.

10. M. H. Mueller, L. Heaton, and L. Amiot, "A computer controlled experiment," Research and Development, p. 34, August 1968.

11. R. P. Noonan, "What kind of computer for your plant," Chemical Engineering, pp. 114-116, June 1969.

12. J. A. Jones, "On-line computers: A survey of techniques and concepts applied to low-energy nuclear research," IEEE Trans. Nucl. Sci., p. 576, February 1967.

13. W. D. Gwinn, et al, "On-line control, data collection, and reduction for chemical experiments," J. Comp. Phys., vol. 2, p. 439, 1968.

14. J. W. Frazer, "Instruments and computers," Science and Technology, no. 79, p. 41, July 1968.

15. J. W. Frazer, "Digital control computers," Anal. Chem., vol. 40, p. 26A, 1968, and "Instruments and computers," Science and Tech., p. 41, July 1968.

16. R. A. Edwards and B. H. Polishook, "Laboratory automation based (LAB) systems," Instrument Society of America, 23rd Annual Conference, New York, October 28-31, 1968.

17. H. Cole, "Computer-operated X-ray laboratory equipment," IBM Journal of Research and Development, vol. 13, pp. 5-14, January 1969.

18. R. E. Anderson and J. W. Frazer, "Computer control in chemistry at the awrence Radition Laboratory," J. Comp. Phys. vol. 2, p. 484, 1968.

19. E. Abrahamson, "Minicomputers for large-scale process control," Datamation, vol. 16, pp. 123-126, February 1970.

20. J. N. Kessler, "Where EE and MD link up to prolong life," Electronic Design, vol. 18, no. 4, pp. 24-28, February 1970.

21. R. E. Anderson, "Dedicated computers for instrument control," March 15, 1968, Contract W-7405-eng-48. (UCRL-70638).

PART 3. MINUTEMAN D17B COMPUTER INSTALLATION

1. Remove the top and sides of the packing crate with a wrecking bar.
2. Remove the cap screws around the base of the missle skin and the large round plug which is located under the red dust cap.
3. Lift off the missile skin and gold dome.
4. Remove the magnesium dome from the top of the computer.
5. Remove the plugs from positions J19, J20, and J21 near the computer base (it may be necessary to drill holes in the computer base to gain access to the attachment screws). Remove the screws which attach the stable platform to the D17B. Remove the stable platform by lifting it upward and freeing the attached cables which were connected to J19, J20, and J21. The locations of these jacks are shown in Figure 1.
6. Remove all wires which lead to external connectors, i.e. wires between J1 and TB1, TB2, TB3, TB4, and GND.
7. Disconnect and remove large power diodes on the underside of the computer.
8. Connect a heavy wire (no. 8 or larger) from E3 to E1 to E7. Also connect a building ground (if available) to E3.
9. Connect the ground of a 28V DC, 30A power supply to E3, and the positive terminal momentarily to E2 to determine if the memory is free to rotate smoothly.
10. Remove the panel which covers the memory and locate a cooling fan so that air is forced directly on and around the memory.
11. Check the 3ϕ 400 Hz signal on J16 pins 5, 6, and 7. More detail as to the specifications of this waveform are presented in the Checkout section (Part 4).

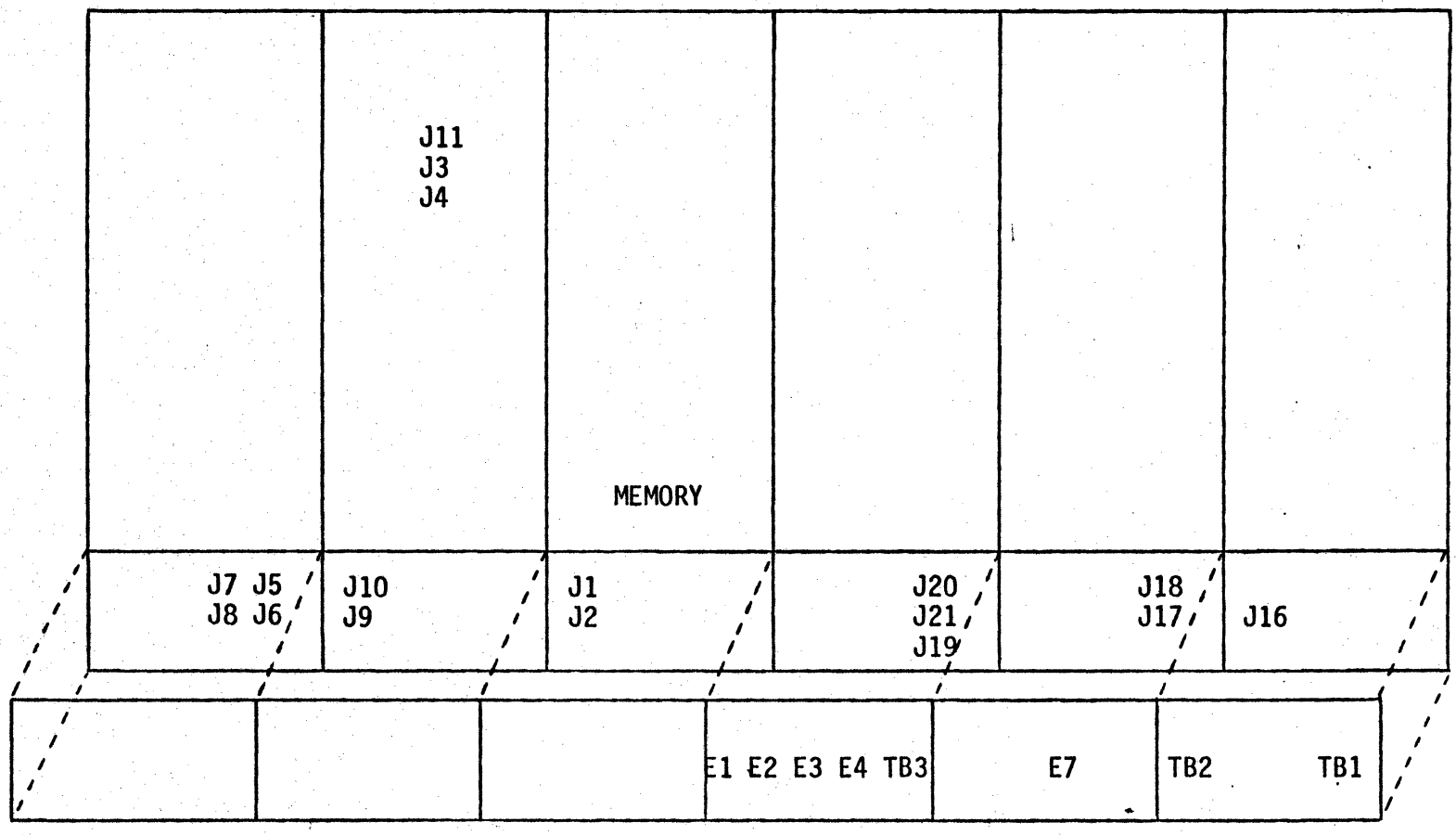


Figure 1. Sketch of jack and terminal locations.

12. Attach permanent cooling and power systems to the computer. One possible cooling system is shown in Figure 2. The two 6" saucer fans are mounted on 3/4" plywood so that cool air is forced down into the computer and consequently through the card slots.

13. A relatively simple control panel may be constructed for the D17B which will allow the operator to supply specific initializing and interactive inputs. These inputs can be supplied via switches and push buttons to cause the conditioning of logic circuitry, logical synchronization, and sequential state transitions between submodes of computer operations. The control inputs and necessary voltages are shown in Figure 3. The voltages required for the panel are -10V and +25V. These voltages are obtained on TB2 pins 77 and 74. The first five functions shown in Figure 2 are I1C-I5C. These functions are the character input lines. I5C is used to generate odd parity over the five lines. A character may be entered into the L-register by setting the desired character on the input lines, I1C-I5C, and applying a timing pulse. The timing waveforms are shown in Figure 4. It is important to note that the values on I1C-I5C must remain constant during the time that TC' is +25V and TC is 0V. The character will be shifted into the L register as TC' falls from +25V to 0V and TC rises from 0V to +25V. The MRC (Master Reset) switch is used to initialize the computer for data entry and for instruction execution. The Disable Discrete (DDC) switch forces all of the discret output lines into the false state and prevents writing into the hot channel (50). Application of the fill signal (FSC) causes the computer to enter the wait sub-mode of the load mode at the first TX time. It is also used to remove parity errors which may have occurred. The Compute or Run-Halt switch puts the computer in the compute or non-compute mode. When the

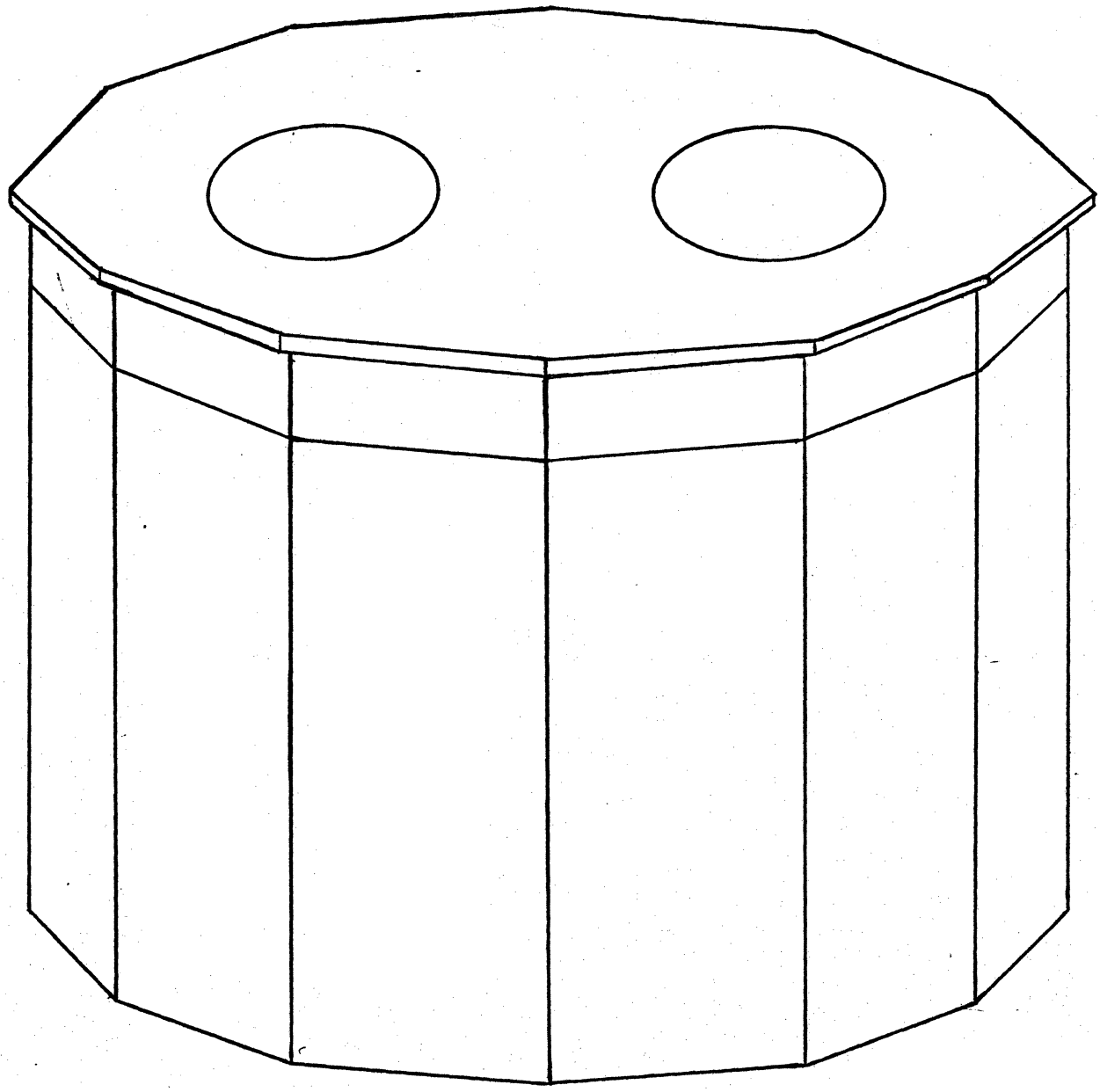


Figure 2. Cooling system arrangement.

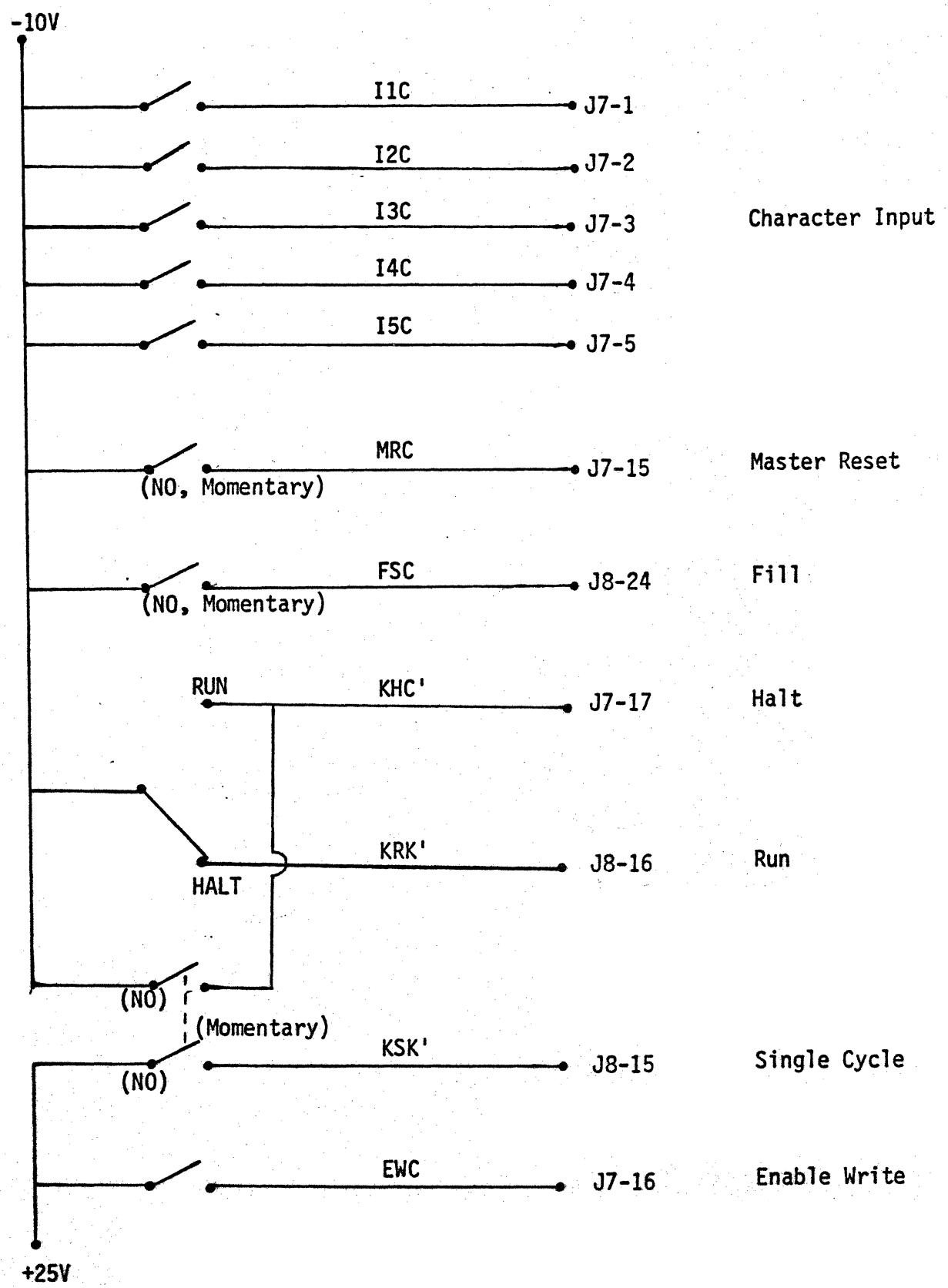


Figure 3. Schematic of manual control panel input switches.

FUNCTION	TERM	LOCATION
Parity Error Output	PK	J8-34
Verify Error Output	PVK	J8-35
Parity-Verify Error	P	J6-25
Timing	TC	J8-38
Timing Prime	TC'	J7-6

WAVEFORMS

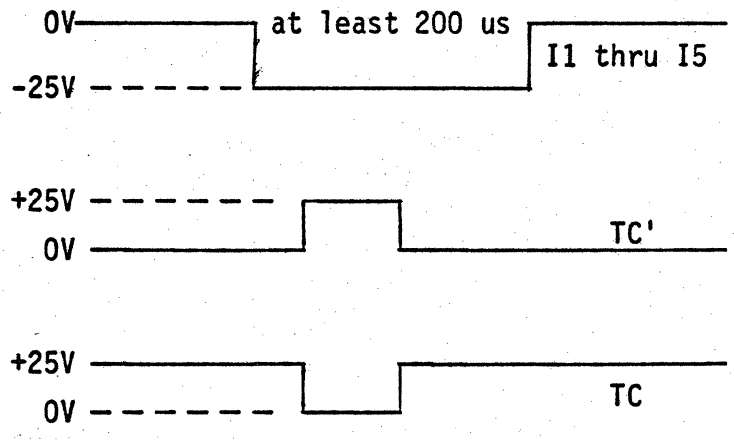


Figure 4. The timing waveforms for character input.

switch is in the Halt position, KRK' (Run prime) is true; in the Run position KHC' (Halt prime) is true. The function KSK' (Single Cycle prime) permits the operator to observe the results of a single instruction execution. When the KSK' switch is depressed, KRK', KHC', and KSK' all become true. EWC (Enable Write) allows writing on the cold channels (00-46 even octal) of memory. If the EWC switch is opened, writing in the cold channels is not possible.

14. Attempt to input information into L-register. At this point the simple test program in section 13 of Checkout (Part 4).

15. As a final step in the D17B installation, an input/output interface between the D17B and a Flexowriter or Teletypewriter may be constructed.[1,2] A Flexowriter model SPD may be used for entering data into the D17 via character input. The output from the SPD consists of five information lines and a timing pulse line. These signals are voltage pulses of +90V and have widths of 30ms. When any key is depressed, a timing pulse is generated. The information lines carry octal characters or control codes which are generated by the Flexowriter when a particular key is depressed. Logic ZERO is OV and logic ONE is +90V. Since the output from the SPD is OV and +90V, electronic interfacing is required in order to remove noise spikes and convert these signals to OV and -10V respectively.
Figure 5 shows a simple and reliable design that interfaces the SPD to the D17. Each information line from the SPD is connected to a potentiometer and a 1K resistor in parallel. The 1K resistor provides a constant load on output to prevent noise spikes. This resistor should be rated at least 2 watts in order to handle the power dissipated. When the SPD output line is OV, the input to the Nand gate is negative and turns the PNP transistor on, so the output is OV. If the SPD output line is +90V, the

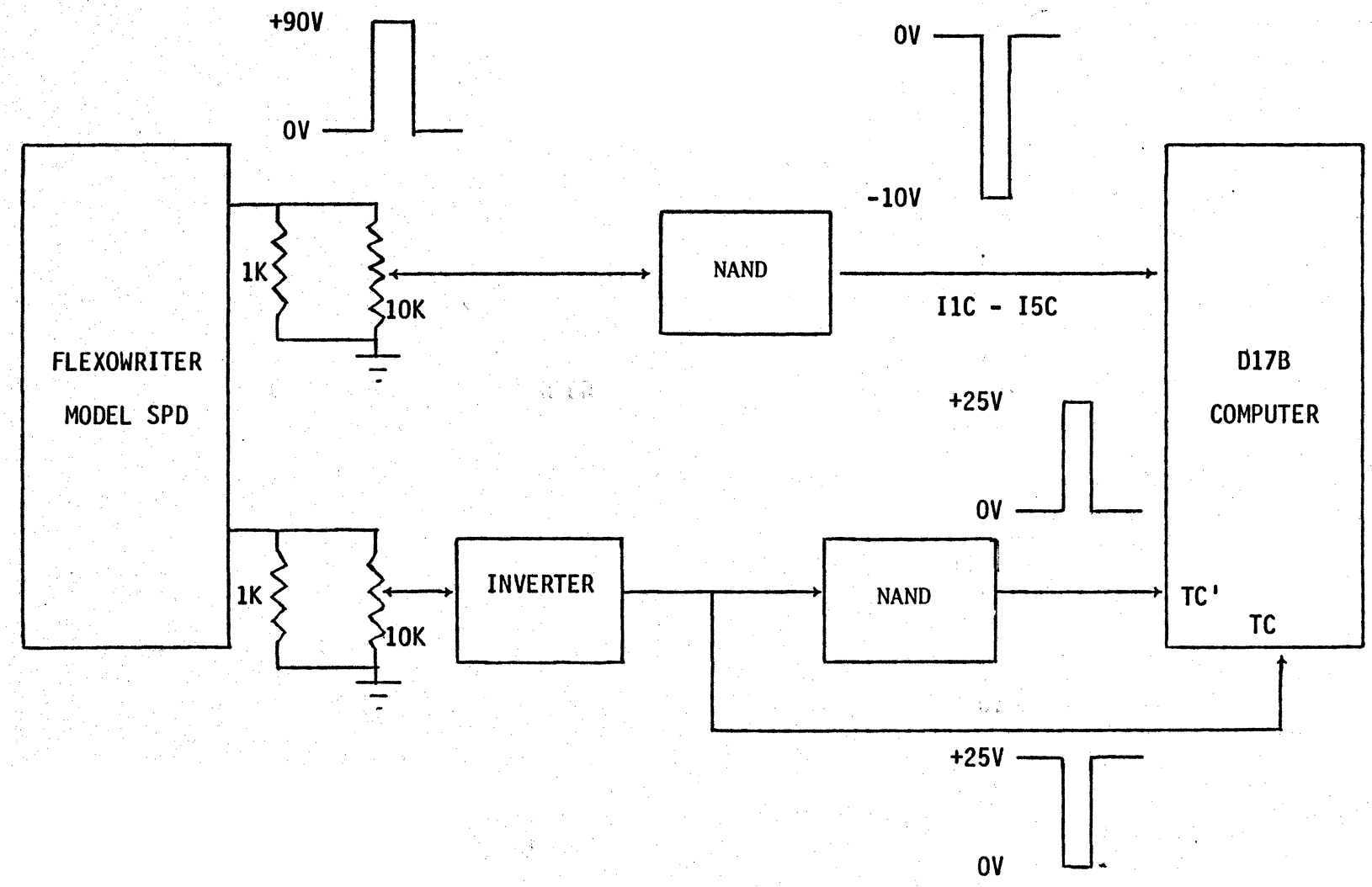


Figure 5. Input interface for D17B--SPD.

input to the gate will be slightly positive. This positive level will cause the transistor to go off and the output will be equal to -10V. The potentiometer is adjusted so that the input to the gate is slightly above 0V for a +90V SPD output.

In order to generate TC and TC', the output of the potentiometer is connected to an inverter. When the SPD timing output line is 0V, the transistor goes off and the output is +25V. For a +90V pulse, the transistor is foward biased and turns on making the output 0V. TC is inverted by a Nand gate to give TC'.

Because the Flexowriter model 2201 has voltage levels of 0V and -90V, some changes must be made in the interface. The information lines need merely to be attenuated from -90V for a logic ONE to -10V. Figure 6 shows the model 2201 interface. The timing interface uses the same circuitry as the SPD but with different bias voltages.

The block diagram for COA execution is shown in Figure 7. Normal recirculation in the A-register is indicated by the right loop because A_c is false and information from A_x is fed directly to A_p. When COA is being executed, the A-register is extended by four flip-flops. A_c is true and the information from A_x is fed to the input of C_4. The four most significant bits (A_{24} -A_{21}) are shifted out to AND gates. The information in the A-register is now shifted to the left by four bits. The J flip-flop generates odd parity for the output. SCT is the COA timing pulse and occurs whenever COA is executing. It is ANDed with the outputs of C_1-C_4 and J to give SC1-SC5. A logical zero is +10V and a -25V pulse is a logical one. The width of the output pulses depends on "S" in the COA instruction. The period between timing pulses depends on the program. Since the A-register contains 24 bits, six COA instructions are required to empty it in groups of 4-bits.

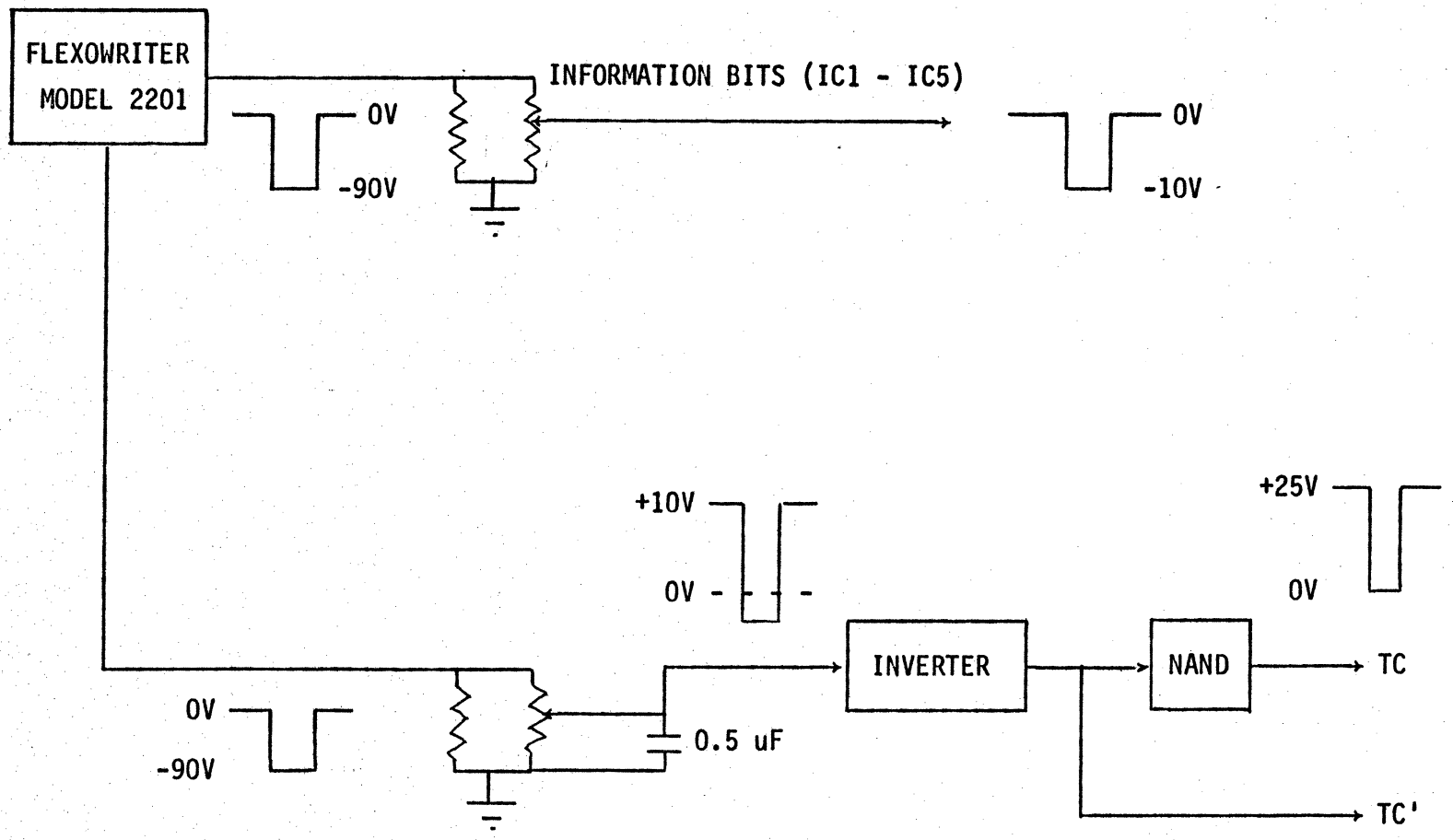


Figure 6. Model 2201 Flexowriter — D17B Interface.

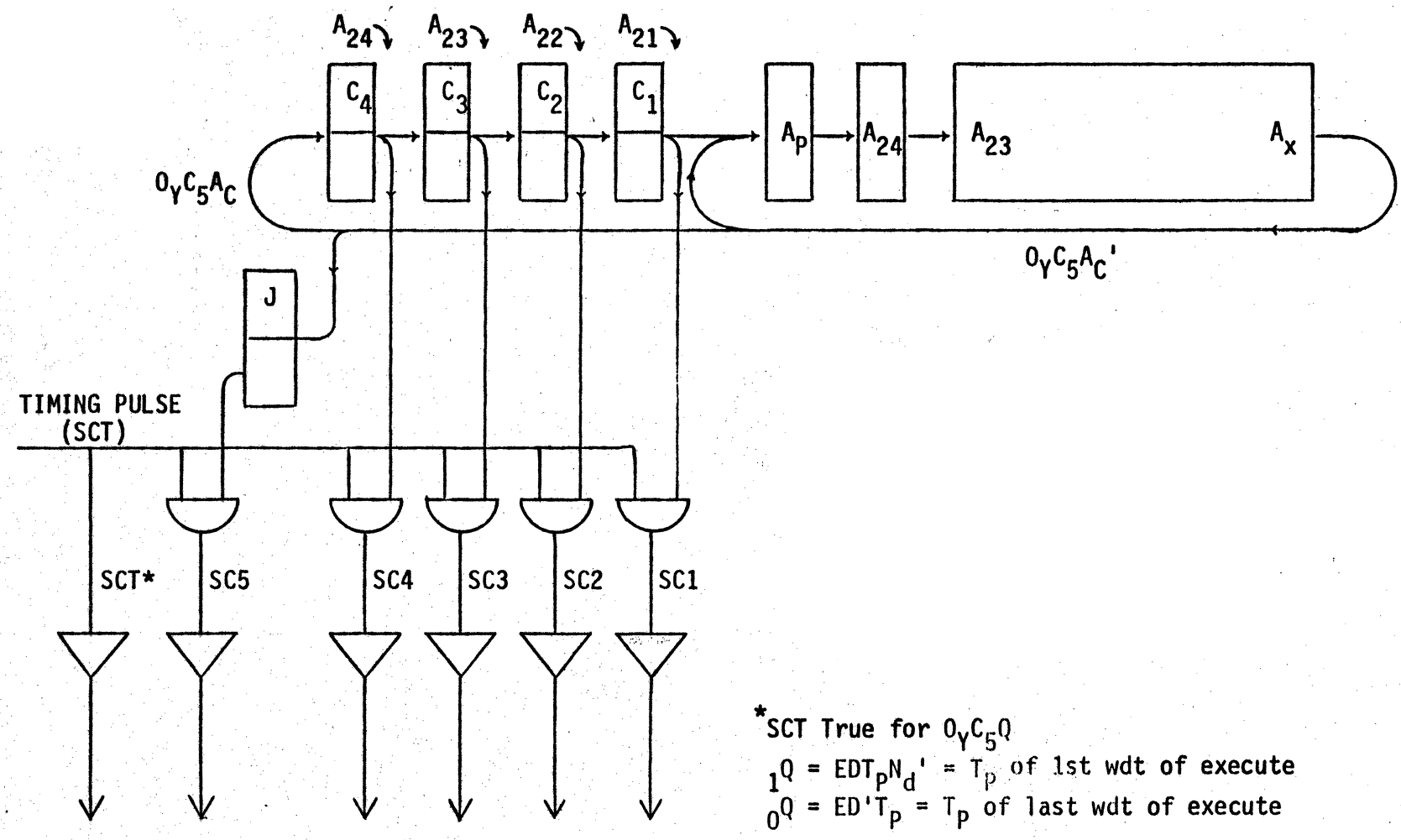


Figure 7. Block diagram for COA execution.

Appropriate programming will allow an effective COA of 3 bits (one octal character) at a time.

The output interfaces for the Flexowriter models 2201 and SPD are the same except for the voltages applied to the relays. The block diagram for the interface is shown in Figure 8. Since the SCT timing pulse is, at most, approximately 3 ms wide, a monostable multivibrator, "one-shot," is required to increase the width of this pulse to at least 30 ms so that the Flexowriter has enough time to respond. When the interface is first turned on, the output of the one-shot is -10V and this resets all five character flip-flops. The input circuits clamp the SC1-SC5 signals to ground to allow manual operation of the flip-flops. When a COA occurs, the inverter and one-shot outputs go from -10V to ground and the set line receives a -25V pulse. The flip-flop is set and the relay driver goes on which applies +90V or -90V to the Flexowriter input. The SPD requires +90V and the 2201 needs -90V for correct operation. The one-shot output is inverted by a NAND gate and fed to a relay driver. The output of the one-shot stays at 0V for about 40 ms, then goes back to -10V, and resets all the flip-flops. Character lines which stay at logic zero (+10V) do not set their flip-flops and their relay drivers remain in the off state.

References

1. G. Saviers, "The minicomputer and the engineer -- peripherals expand your mini's capabilities," Electronic Design, vol. 19, pp. 72-76, June 10, 1971.

2. "Free: used missile computers," EDN/EEE, vol. 16, pp. 12-13, June 1, 1971.

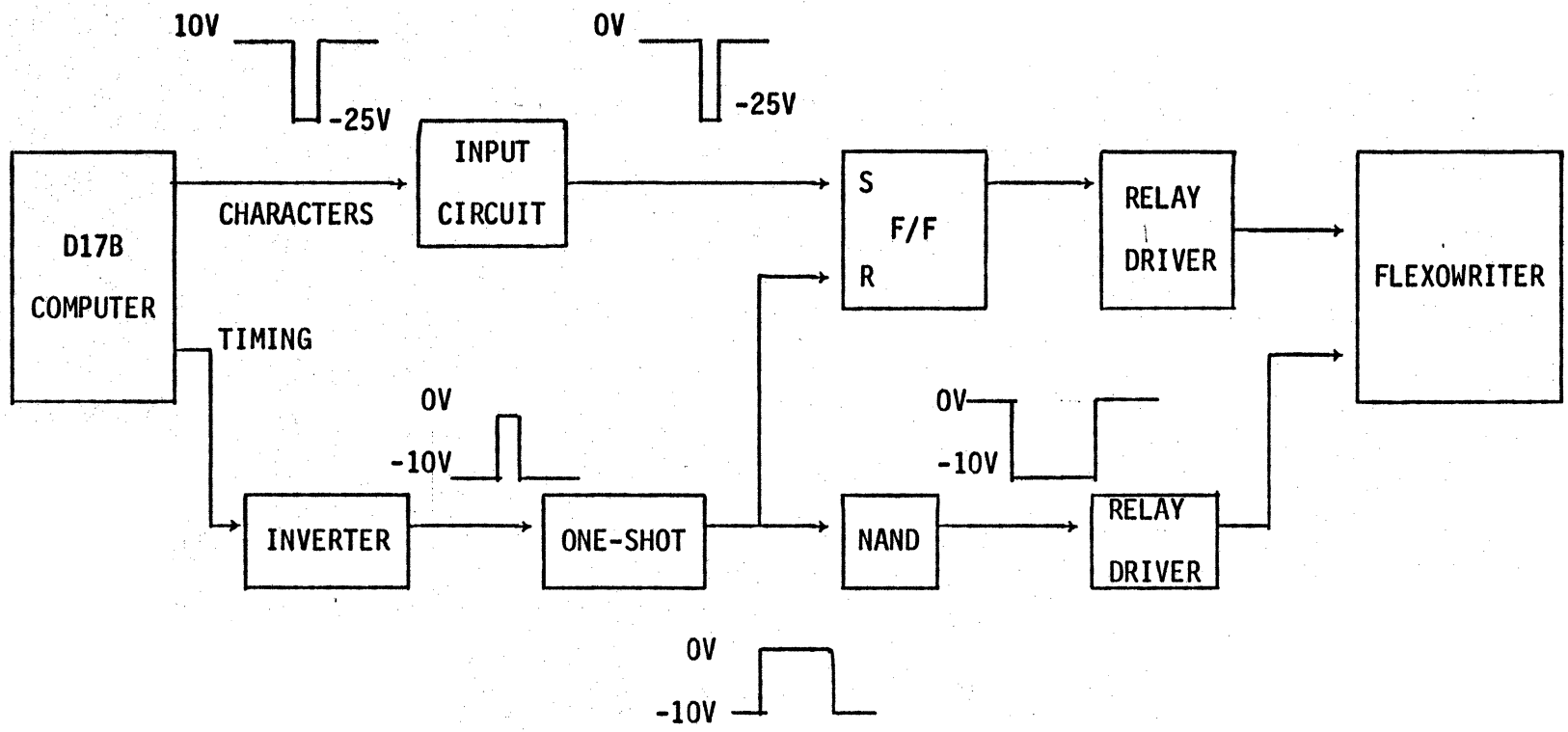


Figure 8. D17B—flexowriter output interface.

PART 4. MINUTEMAN D17B COMPUTER CHECKOUT

1) Apply 28VDC to the D17B computer.

2) Check the internal power supply voltages:

Voltage	Location
2V	TB1B -63
15V	TB2B -67
6V	-68
-35V	-69
- 1V	-70
-25V	-71
- 5V	-72
10V	-73
25V	-74
- 3V	-75
35V	-76
-10V	-77
-28V	-78

3) Monitor the 3ϕ 400 Hz memory voltages at J16 pins 5, 6, 7 on a CRO. (Fig. 1).

4) A 400 Hz audio frequency signal and the high speed mechanical motion of the memory motor should be heard on the memory plate.

5) Monitor the output of the clock channel read amplifier CAC at A6 pin 15. (Fig. 2).

6) Monitor the clock on J3 pin 47. (Fig. 3).

7) Monitor the bit counters B1 through B6 at J3 pins 1 through 6. (Fig. 4.1 through 4.6). They should synchronize on the CRO.

8) Initiate MRC (Master Reset) and monitor J, K, V_c and R_c at J3 pins 13, 14, 35, and 36 respectively: J = -10V, K = 0V, V_c = 0V, and R_c = 0V.

The 3Ø 400Hz memory motor voltages

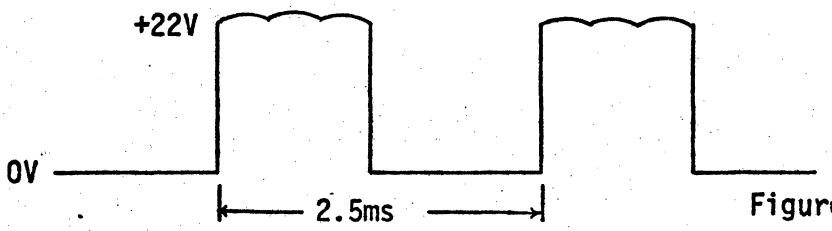


Figure 1.

The clock channel read amplifier CAC

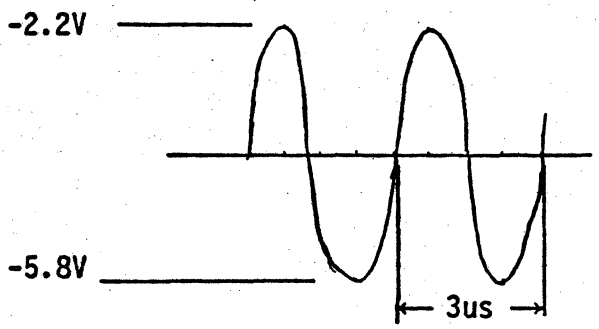


Figure 2.

Clock

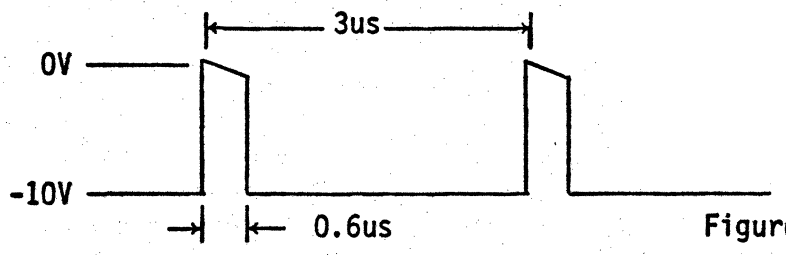


Figure 3.

The bit counters B1 - B6

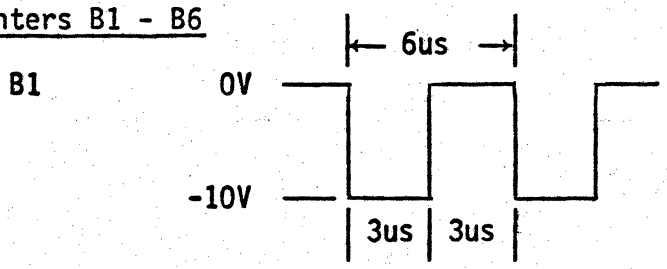


Figure 4.1

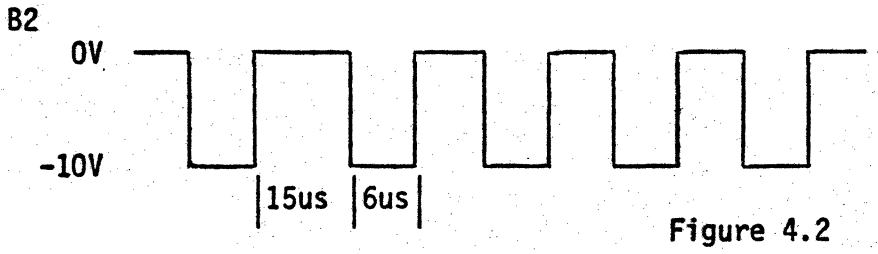


Figure 4.2

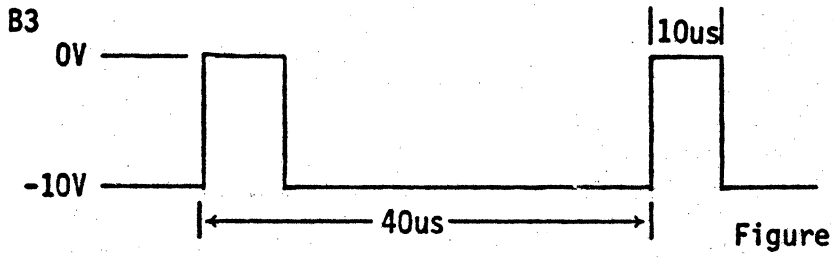


Figure 4.3

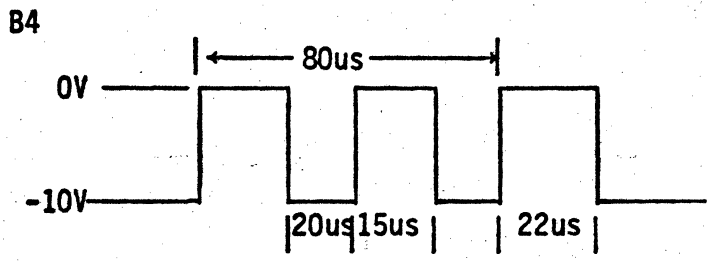


Figure 4.4

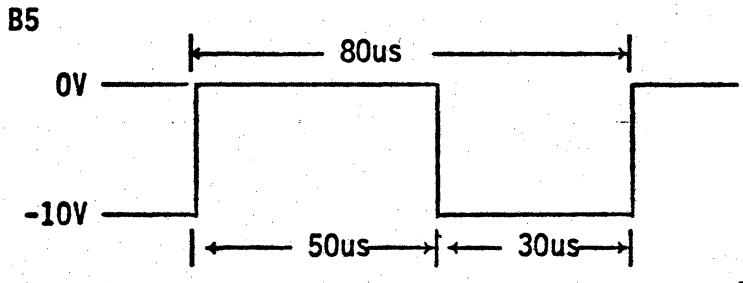


Figure 4.5

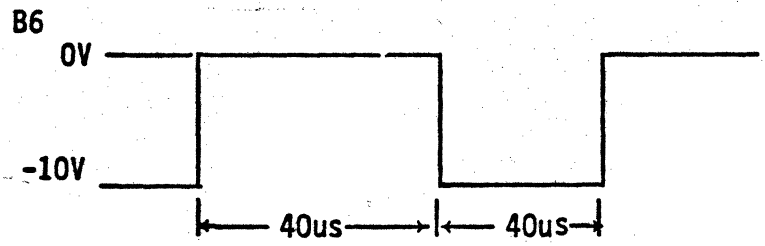


Figure 4.6

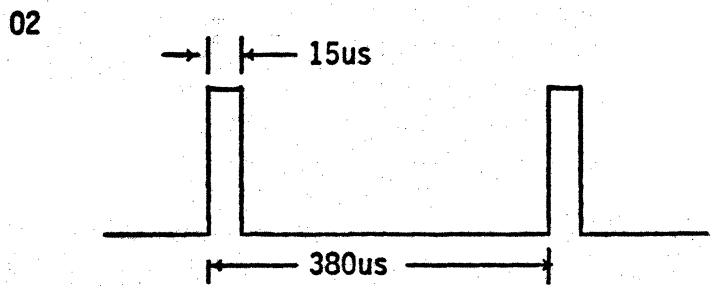


Figure 5.

In the D17B computer, False or logic ZERO is 0V, and True or logic ONE is -10V.

9) Monitor 0_1, 0_2, and 0_4 at J4 pins 11, 12, and 14, respectively: 0_1 = -10V, 0_2 (Fig. 5), 0_4 = -10V.

10) Monitor CP1 through CP4 on J4 pins 18 thru 21. Make I_{1c} true and apply a timing pulse. CP1 should be true while the timing switch is depressed. Repeat the preceeding steps for CP2 through CP4 while making I_{2c} thru I_{4c} true.

11) Monitor L_p at J3 pin 43 using B_1 as a synchronizing signal and load octal numbers via character input. Since the data are shifted in from the left on the CRO, words will be displayed in reverse.

12) Monitor A_p and I_p at J3 pins 41, 42. When a Location code is loaded, the contents of the L-register are transferred to the I-register. If an Enter code is loaded, the contents of the L-register are transferred to the A-register. An Enter code also causes the contents of the I-register to increase by one.

13) Begin the elementary software checkout procedure by loading the Ripple Counter program shown in Table 1 using the Manual Control Panel.

+	Fill Code
44010002E	CLA "1"
64010002E	ADD "1"
00000001E	Data "1"

Table 1. Ripple Counter Program

a. Synchronize the CRO with B_1.

b. Depress MRC and monitor L_p. Set the manual character input switches to represent each character of the first instruction beginning with

the most significant character. Depress the timing switch after each character is set. L_p should display the first instruction in reverse order.

c. Monitor I_p. It should display the instruction 50000000.

d. Monitor A_p. Set the manual character input switches to represent an ENTER code (E), and depress the timing switch. The contents of the L-register should transfer to the A-register and the I-register should increase by one.

e. Repeat steps b through d above for both instructions and the data word.

f. Depress MRC and monitor I_p. The Transfer instruction 5000000 should be displayed on the CRO in reverse.

g. Depress K_{sk} and monitor I_p. It should display the first instruction 44010002 in reverse.

h. Depress K_{sk} and monitor I_p. The ADD instruction 64010002 is displayed. By monitoring A_p, the data 00000001 is displayed.

i. Continue depressing K_{sk}. The I-register should stay unchanged and the A-register should increase by one for each single cycle.

j. Depress MRC and flip the Run toggle switch. Monitor A_p on the CRO. The A-register should start with 00000001 and increase by one count every disc revolution (10 ms).

PART 5. MINUTEMAN D17B COMPUTER TROUBLE-SHOOTING

A. Non-Compute Mode

The sequence of operations for the Load Mode is shown in Table 1. The trouble-shooting procedure is as follows:

1. Apply the 28 VDC power to the D17B computer. All the internal control flip-flops assume random states, and the bit counter is not synchronized with the sector track.
2. Application of a MRC pulse causes the control flip-flops to assume specific steps and the bit counter to synchronize with the sector track.
3. The computer enters the Manual Halt Interlock submode via the Idle 1 submode and begins to oscillate between the Interlock and Prepare to Load submodes.
4. Apply the Initiate Load FSC signal. The computer will enter the Wait submode of the Load mode.
5. Enter the data into the memory via the A and L registers with the proper control codes.
6. If the expected transition does not occur or if it takes place at the wrong time, the appropriate logic equations should be checked to determine where the malfunction has occurred.

Example:

"The A, L, and I registers were monitored on a CRO for programming, but they could not be brought into synchronization." The trouble-shooting procedures are:

1. Examine the bit counters B1 through B6 and the timing signal T_p.
2. The logic signals from B_1 and B_2 were changing state normally; however, the terms B_3, B_4 were a constant logic ONE while B_5, B_6 were ZERO.
3. In order to make the D17B computer transition to Sync. Bit Counter 2 Mode,

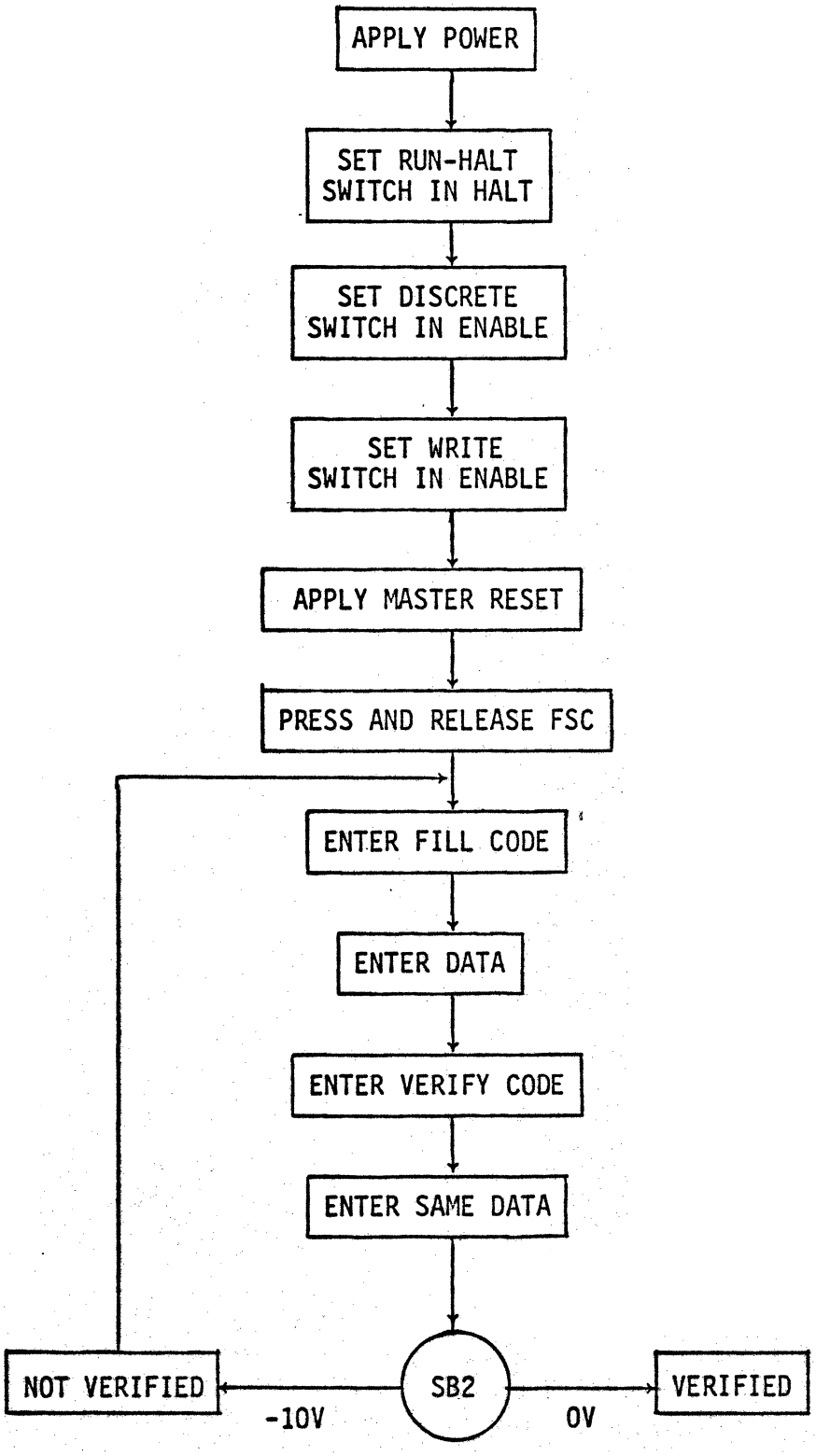


Table 1. Sequence of operations for the load mode.

the respective flip-flop states should be $K'J\ V'_c\ R_c 0_1$.

$$_1 0_1 = K'\ V'_c\ J\ 0'_1\ S\ B_6\ B'_3\ B'_5$$

Since B_6 is a constant ZERO, the registers could not be brought into synchronization.

4. Check the logic equations for $_1B_5$,

$$_1B_5 = B'_6\ B'_5\ B_4\ B'_2\ B_1\ T'_p$$

It is known that $(B'_6\ B'_5\ B_4)$ is true constantly, and it is also known that B_2 and B_1 are working properly. B_5 should be ONE according to the logic equation shown above; however, the output of the gate on the D17B for $_1B_5$ was ZERO.

5. A27 was checked and the faulty AND gate was discovered. Part of A27 is the logic for $_1B_5$.

6. Upon installation of the new circuit board, the problem was eliminated.

B. Compute Mode [1]

The sequence of operations for Compute Mode is shown in Table 2. The trouble-shooting procedures are as in the following:

1. Begin with a software checkout to ensure that the problem is really a hardware malfunction.

2. Monitor the I-register to determine what instruction the computer is implementing. The following steps can be taken:

   a. Synchronize a CRO using the bit counter B_1 and monitor the I_p flip-flop.

   b. Display the contents of the Accumulator and the appropriate memory location and carry out a single-cycle operation.

   c. Re-examine the A-register and the same memory location following the single-cycle execution of the instruction.

   d. Compare the data obtained in steps b and c. Determine why the

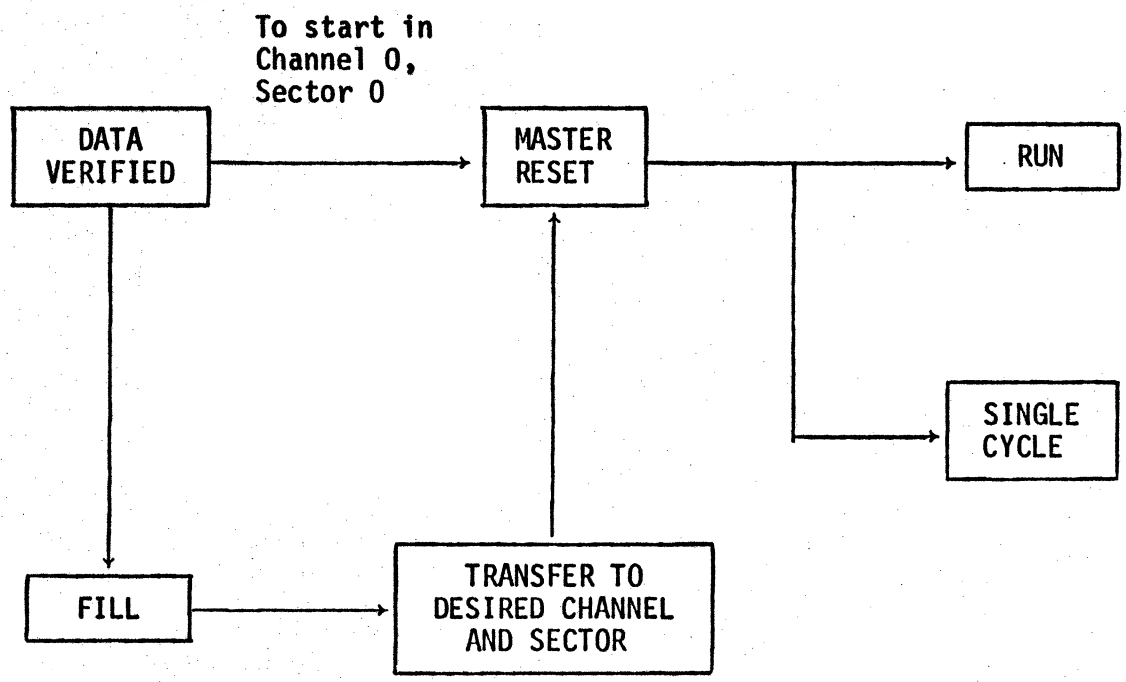


Figure 2. Sequence of operations for the compute mode.

malfunction is occuring and what area should be subjected to further examination from the knowledge of the contents of the respective memory locations before and after instruction execution.

3. Investigate the accuracy with which the machine decodes the I-register as follows:

   a. Check the C_{B5}—C_{B1} and C_5—C_1 flip-flops for the correct channel address.

   b. Check the S_{B3}—S_{B1} and S_3—S_1 flip-flops for the correct flag address.

   c. Check the O_{B4}—O_{B1} and O_4—O_1 flip-flops for the correct operation code.

   d. Check N_d and I_d flip-flops to see if sector agreement occurs by removing the sector disagreement logic from the computer and test statically.

4. Investigate the logic equations and flow diagrams for the individual instructions to identify the malfunctioning logic elements.

5. Remove the necessary circuit boards from the computer and replace with new boards.

Example:

"It was discovered that the computer would not transfer to the proper location during a TMI when the contents of the Accumulator were negative. An associated problem was that the computer would not pick up the correct next instruction during a flagged instruction. The computer would take the next instruction from the first 16 sectors of the channel instead of from the 16 sectors following the operand location." The problem is solved in the following manner:

1. Follow the trouble-shooting procedures up to Step 2. The problem is identified as stated previously.
2. Consult the logic equations. Since the TMI and flagged instruction problems seem to have the same basic origin, it is supposed that some function which involves both is causing the trouble.
3. The logic equation for $_{0}I_{D}$ is checked. The I_{D} flip-flop indicates instruction agreement with various word length channels which involves the selection of the next instruction location.
4. It is found that the term 0_{cx} of $_{0}I_{D}$ is used to force instruction agreement when number agreement is found. The Accumulator is negative, and the instruction is TMI. This is the logic involved in the problem.
5. The card that generated these logic functions was changed (J432 - No. 119 Logic Network), and the problem was solved.

Reference

1. D. A. Babroff, "Avoid pitfalls in computerized testing," Electronic Design, vol. 17, pp. 196-201, August 1969.

PART 6. MINUTEMAN D17B COMPUTER PROGRAMMING

The D17B is a small serial-binary general-purpose computer which was designed as an airborne control computer. It has several features which make machine language programming different than for other general-purpose [1-3] computers. Some special features of D17B computers are:

A. Flag Storing

The D17B has a "flag store" mode which provides for simultaneously storing the previous contents of the Accumulator in certain specified channels, coincident with the execution of an instruction, to perform this frequent operation.

B. Split-Word Arithmetic

The D17B can also be used to perform parrallel- or multi-processing such as the simultaneous execution of two identical single-precision arithmetic operations. This not only provides for the execution of two operations during one word time, but also effectively doubles the memory available for data storage. In single-precision data storage each word is divided into two 11-bit words at the sacrifice of some precision.

C. Access Timing

The serial presentation of information on the rotating disc necessitates careful coding of a program to insure the minimum access time between instructions as well as between an instruction and the data on which it operates. This scheme of coding is referred to as minimal delay coding (MDC).

The execution of each instruction can be considered as being accomplished in the following five phases which are usually common to delay-type memories:

1. Instruction search (IS)
2. Instruction read (IR)

3. Operand search (OS)

4. Operand read (OR)

5. Execute (EX)

Figure 1 shows that the D17B can perform several of these phases simultaneously with increased efficiency compared to sequential operation. This figure assumes MDC of instructions which require an execution time of one word time. The advantage of this minimized access timing is that, once a minimal delay coded program is initiated, the effective completion time of any instruction is equal to the basic execution time of the instruction. If random access addressing were used in the D17B, the search operations (IS and OS) could each require up to 128 word times or one disc revolution of 10 ms. MDC places the next instruction operand at a location which will pass the read head immediately after completion of the current instruction.

D. Fine Countdown

The fine countdown mode of the computer links together the 4-word V incremental input loop and the 1-word U rapid access loop in a special purpose operation. Once the fine countdown is initiated, the operation is performed idependently from the program in memory.

When the fine countdown mode is entered, the 0, +1 and -1 incremental inputs to the V-loop cause the product of the input and the contents of their sector to be algebraically added to the contents of the U-loop. The U-loop, assuming that it contained a positive number via flag storage originally, is counted down at a rate depending upon the sign and magnitude of the numbers placed in the V-loop and upon the sign and frequency of the inputs. When the contents of the U-loop become negative, a discrete output signal (D_{16}) is issued automatically.

"Timing" is an important essence in a real-time program. Certain

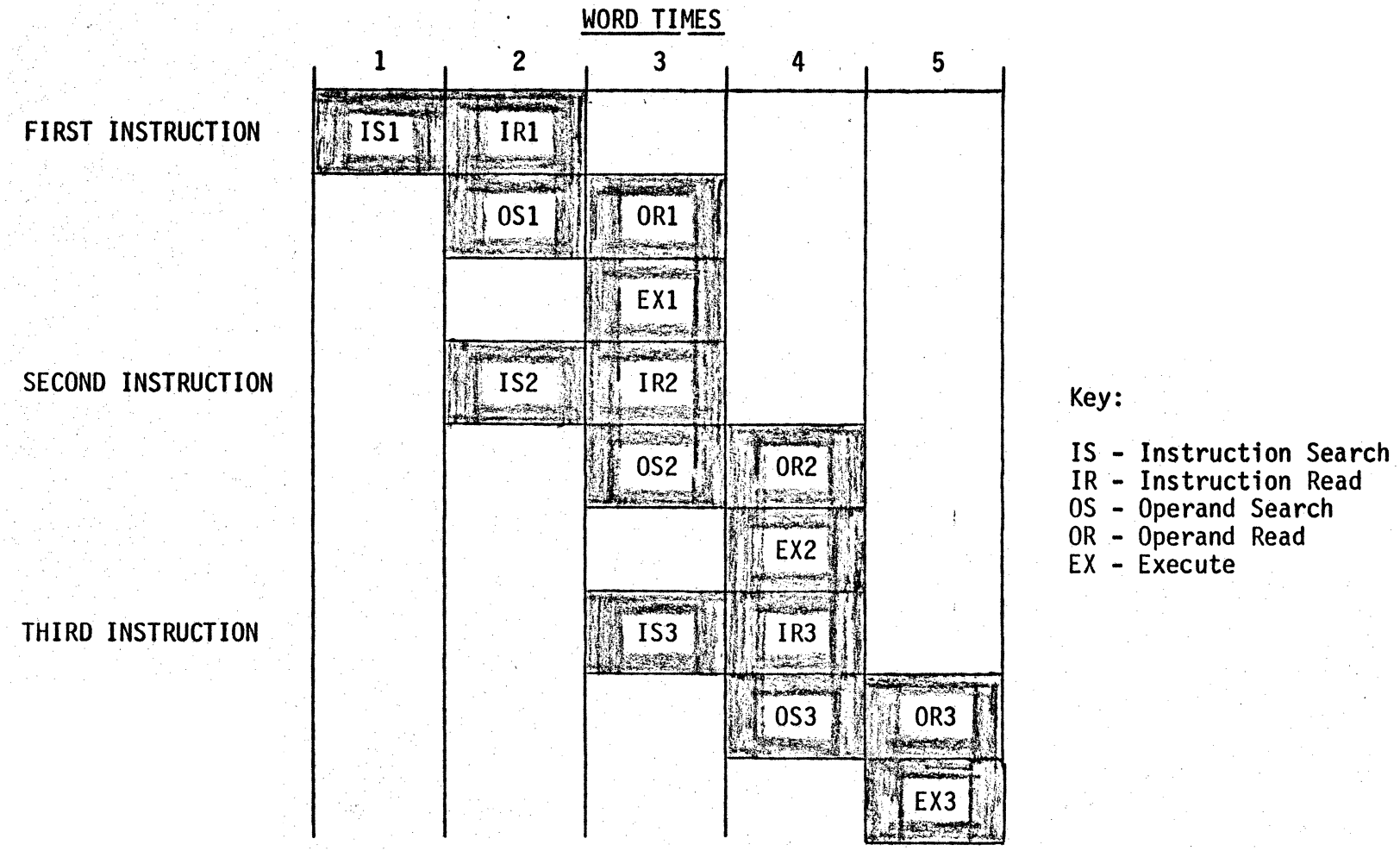


Figure 1. Steps required for instruction completion.

subroutines may take different execution times for different data values. The fine countdown mode can be utilized for timing control when this situation arises. The fine countdown mode is entered immediately before the program transfers to the specific subroutine. Usually the fine countdown mode will continue for the maximum execution time of the subroutine. If the execution of the subroutine finishes before the fine countdown mode ends, the program will be forced into a loop until the discrete output signal brings the program to its proper location. By this technique, the execution times of the subroutine can be equalized and the whole program treated in real-time fashion.

Data and Instruction Word Format

Data are represented in the D17B in the fractional form. For a full-word operand, the sign bit is located in T_{24}. The magnitude of the number is determined by the information located in T_1 through T_{23}. A ONE in T_{24} indicates that the quantity located in bit positions T_1 through T_{23} is negative. Negative numbers are read in the two's complement. By splitting an operand (full-word) in half, we now have two operands. T_{24} and T_{11} are the sign bits and T_{23} to T_{14} and T_{10} to T_1 are the information bits for the left half-word and right half-word respectively. In split-word arithmetic, T_{12} and T_{13} are not used. The data word format for the D17B computer is shown schematically in Figure 2.

Direct addressing of the entire memory of the D17B as illustrated in Figure 3 by using a 12-bit operand address field is a feature of considerable value. A typical two-address (unflagged) D17B instruction has three parts: an operation code and two addresses. One address identifies the operand which fulfills the same function as the address field in a single address machine; the second is the address field S_p which is used to specify the address of the next instruction within the active memory channel. One bit

SIGN	WHOLE NUMBER																							
T_{24}	T_{23}	T_{22}	T_{21}	T_{20}	T_{19}	T_{18}	T_{17}	T_{16}	T_{15}	T_{14}	T_{13}	T_{12}	T_{11}	T_{10}	T_{9}	T_{8}	T_{7}	T_{6}	T_{5}	T_{4}	T_{3}	T_{2}	T_{1}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7			7			7			7			7			7			7			7			

SIGN	10-BIT BINARY FRACTION										X	X	SIGN	10-BIT BINARY FRACTION									
T_{24}	T_{23}	T_{22}	T_{21}	T_{20}	T_{19}	T_{18}	T_{17}	T_{16}	T_{15}	T_{14}	T_{13}	T_{12}	T_{11}	T_{10}	T_{9}	T_{8}	T_{7}	T_{6}	T_{5}	T_{4}	T_{3}	T_{2}	T_{1}
1	1	1	1	1	1	1	1	1	1	1	X	X	1	1	1	1	1	1	1	1	1	1	1
7			7			7			6			3			7			7			7		

Figure 2. D17B data word format.

UNFLAGGED INSTRUCTION (T_{20} = 0)

T_{24} T_{23} T_{22} T_{21}	T_{20}	T_{19} T_{18} T_{17} T_{16} T_{15} T_{14} T_{13}	T_{12} T_{11} T_{10} T_9 T_8	T_7 T_6 T_5 T_4 T_3 T_2 T_1
OPERATION CODE	FLAG	NEXT INSTRUCTION SECTOR ADDRESS	CHANNEL NUMBER	SECTOR NUMBER
OP	F	S_P	C	S

FLAGGED INSTRUCTION (T_{20} = 1)

T_{24} T_{23} T_{22} T_{21}	T_{20}	T_{19} T_{18} T_{17}	T_{16} T_{15} T_{14} T_{13}	T_{12} T_{11} T_{10} T_9 T_8	T_7 T_6 T_5 T_4 T_3 T_2 T_1
OPERATION CODE	FLAG	FLAG STORAGE LOCATION	SECTOR OF NEXT INSTRUCTION	CHANNEL NUMBER	SECTOR NUMBER
OP	F	S_F	S_P	C	S

Figure 3. D17B instruction word formats.

(F-flag) in the address mode field permits the use of two alternate address modes. If the flag bit is ON, an instruction is interpreted as a three-address word. A typical three-address (flagged) instruction has four parts: an operation code and three addresses. One address again identifies the operand; the second is used to specify the channel S_F in which the present contents of the accumulator are to be stored; the third is used to specify the address S_p of the next instruction within the next sixteen successive memory locations in the active channel after the operand agreement. A program in a single address machine is likely to require much more memory than is required by the D17B. The codes for flagged channels are given in Figure 4.

Minimal Delay Coding and Subroutine Linkage

Because information on the memory disc is transferred serially, the location of successive instructions and their operands on the disc is extremely important. With MDC an instruction can be completed in the number of word times equal to the execution time of that instruction. Therefore, to execute a number of sequential instructions in the minimum number of word times, the instructions should be separated by n-1 sectors where n is the execution time, measured in word times, of the first instruction of each pair of instructions considered.[4,5]

For a given instruction, the operand can be read during the word time immediately following the reading of the instruction. Therefore, when writing a program, the operand of each instruction should be located in the sector location of the instruction. In order to maintain the desired instruction locations, the operands should be located in other channels in the correct sector since corresponding sectors in all channels are available during the same word time. It is usually convenient to locate all operands in the same channel.

T_{19}	T_{18}	T_{17}	CODE	FUNCTION
0	0	0	0 0	IDLE
0	0	1	0 2	(F) 4-WORD LOOP
0	1	0	0 4	(T) TELEMETRY
0	1	1	0 6	(50) HOT CHANNEL
1	0	0	1 0	(E) 8-WORD LOOP
1	0	1	1 2	"L" 1-WORD LOOP
1	1	0	1 4	(H) 16-WORD LOOP
1	1	1	1 6	(U) 1- WORD LOOP

Figure 4. Flagged channel coding.