The Lincoln TX-2 Computer Development*

WESLEY A. CLARK†

INTRODUCTION

The TX-2 is the newest member of a growing family of experimental computers designed and constructed at the Lincoln Laboratory of M.I.T. as part of the Lincoln program for the study and development of large-scale, digital computer systems suitable for control in real time. Although, in general characteristics and design philosophy, it owes a great deal to its predecessors, Whirlwind I and the Memory Test Computer, the Lincoln TX-2 incorporates several new developments in components and circuits, memories, and logical organization. It is the purpose of this paper to summarize these new features and to give some idea of the historical development and general design objectives of the TX-2 program. Fig. 1 shows TX-2 in its present development stage.

The development of a $256 \times 256$, switch-driven, magnetic-core memory array was begun and the Philco surface-barrier transistor made its appearance. After some very promising bench experiments with flip-flops and logic circuits, it became apparent that this transistor was potentially well-suited to use in large-scale systems and warranted further study. Accordingly, plans were laid for a succession of experimental digital systems of increasing size and complexity which would make possible the development and evaluation of circuits using the surface-barrier transistors, and which would lead to a computer of advanced design that would be capable of making efficient use of the $256 \times 256$ memory.

A double-rank shift register of eight stages and containing about 100 transistors was constructed and put on life-test in April, 1955. It has since been circulating a fixed pattern almost continuously with no known errors and no natural transistor failures.

As the next step, it was decided to build a small, high-speed, error-detecting multiplier and incorporate marginal checking and other system features. The value of a multiplier as a preliminary model had been well demonstrated by the 5-digit system built during Whirlwind's early development. The shift, carry, count, and complement operations, under closely controlled timing conditions, were felt to be representative of all of the operations in the manipulative elements of the type of computer planned. Accordingly, an 8-bit system using 600 transistors was designed and completed in August, 1955 and has been in nearly continuous operation since. Operating margins are periodically checked, and in steady state operation, the multiplier's error-rate has been about one every two months, or one error per $5 \times 10^{11}$ multiplications at $10^5$ multiplications per second. Most of these errors appear to have been caused by cracks in the printed wiring which open intermittently.

During this period, a better idea of the general characteristics of the projected computer began to develop and the engineers who were designing the $256 \times 256$ memory were encouraged to think in terms of a word of 36 bits. The notion of a logically separate input-output processor was examined and rejected in favor of a minimum buffering scheme in which data is transferred directly to and from the central memory of the computer. The possibility was recognized of programming these transfers by means of additional program sequences and associated program counters, thus taking advantage of the extensive facilities of the central machine itself for processing input-output data.

It was realized that another development step was desirable before attempting such an elaborate 36-bit system. The 8-bit multiplier had produced a certain
measure of confidence and familiarity with circuits, packaging, and techniques of logical design, but there remained the problems associated with communicating with memory units and input-output equipment operating at vacuum-tube levels over relatively large distances from a central machine which operated at transistor levels. It appeared that the memory development, which had now entered the construction phase, would also benefit by a preliminary evaluation of the 256×256 array and its switching, timing, and noise problems in an operating computer of some kind, possibly with a reduced word length. It was, therefore, decided to design and build next a simple machine—in fact, the simplest reasonable machine—in order to bring about an early intermediate closure of the various efforts within the program.

After some thought about the various possible minimal machines, a design was completed in which the word length would be 18 bits—a graceful half of the projected final form. We began to refer to this computer as the TX-0 and to the projected machine as the TX-2. Because the 256×256 memory array required 16 bits for complete addressing, the single-address instruction word of the TX-0 was left with 2 bits in which to encode instructions. The particular set of instructions included three which required a memory address (add, store, and conditional jump) and one which did not. In this last instruction, the remaining 16 bits were used to control certain necessary and useful primitive operations such as clearing and complementing the accumulator, transferring words between registers, and turning on and off input-output equipment.

The TX-0, equipped with a Flexowriter, a paper-tape reader, and a cathode-ray tube display system was completed, except for the memory, in April, 1956. Twenty planes of the 256×256 memory array were installed the following August and the TX-0, now containing about 3600 transistors and 400 vacuum tubes, began to function as a complete computer. Since that time, it has been used to run a variety of testing and demonstration programs, and a symbolic address compiler and other utility programs have been constructed and are currently in use.

Not only has the TX-0 served the evaluational purposes for which it was built, but it has also demonstrated an effectiveness as a usable computer that is somewhat surprising in view of its simplicity. Its relatively high speed of about 80,000 instructions per second and its 65,536-word memory compensate in large measure for the limitations of its instruction code and logical structure.

With the successful completion of the TX-0, the final steps in the development were undertaken in packaging, circuit refinement, and logical design of the TX-2. A great deal had been learned about the performance of the transistors and memory, the types of logical circuits which are practical, techniques of marginal checking, and the lesser system problems such as color scheme selection and the proper location of pencil sharpeners.

As design work progressed, the TX-2 took form as a system of about 22,000 transistors and 600 vacuum tubes. It is an interesting fact that at each step of the development since the shift register, the number of transistors involved was about 6 times the number in the preceding step. This is graphically shown in Fig. 2. At the time of writing, approximately 16 million transistor-hours have accumulated in the shift register, multiplier, and TX-0. There have been two natural deaths and a dozen or so violent ones, primarily due to contact shorting with clip leads and probes.

Fig. 2—Steps in the Lincoln TX-2 development program.

**Design Objectives**

In describing design objectives, it should be pointed out that speed of operation was not the primary consideration to which all other attributes were sacrificed. It would have been possible, at the expense of a few more logic circuits, to increase the speed of multiplication, division, and shift-type operations. Similarly, the operation of the index register system could have been made more efficient at the cost of an additional small, fast memory. The principal objective was rather that of achieving a balance among the factors of speed, reliability, simplicity, flexibility, and general virtue.

A key aspect is that of expandability which, in an experimental computer in an active environment, certainly ranks with the foregoing qualities in importance. The address structure in the TX-2 permits an expansion of the memory by about a factor of 4, partly to allow for new memory developments, such as the transistor-driven 64×64 array which was begun following the completion of TX-0. New instructions and pieces of terminal equipment will certainly be added during the course of future operation. Extra space and spare plugs have been artfully distributed about in constructing the computer frame. Finally, modular construction will permit a fairly easy physical expansion when required.

The result of all this activity has been a computer of relatively large capability. In addition to incorporating high-speed transistor circuits and a large magnetic-core
memory array, the Lincoln TX-2 has two major and distinguishing design characteristics:

1) The structure of the arithmetic element can be altered under program control. Each instruction specifies a particular form of machine in which to operate, ranging from a full 36-bit computer to four 9-bit computers with many variations. Not only is such a scheme able to make more efficient use of the memory in storing data of various word lengths, but it also can be expected to result in greater over-all machine speed because of the increased parallelism of operation.

Peak operating rates must then be referred to particular configurations. For addition and multiplication, these peak rates are given in Table I.

2) Instead of one instruction counter, the TX-2 has 32 such counters which are assigned separately to different users of the computer, who then compete for operating time from instruction to instruction. A special part of the machine selects a particular user based partly on a predetermined priority schedule and partly on the current needs of that user. This multiple-sequence operation, in which many essentially independent instruction sequences interrupt and interleave one another, is an extension of the breakpoint operation found in DYSEAC of the National Bureau of Standards.

The value of these features will have to be assessed during the course of future machine operation. The features themselves are discussed in more detail in the next two papers.

---

**Discussion**

**P. C. Miller** (Logistics Research): When several programs are being run simultaneously, are there any provisions to prevent each of the users from stealing another's storage space?

**Mr. Clark:** This is a provision which, of course, has to be made by the programmer himself. Actually, suppose we were talking about the multisequence type machine that will be discussed in the next paper, I might mention that there are certain things that we can't acquire in the control of the programmer, if the programmer does try to use the same area of storage that another programmer is using, and there is no way at all that we are going to try to attempt to detect this.

**F. S. Preston** (Norden Labs.): What uses are planned for the TX-2?

**Nelson Blachman** (Sylvania EDI) and **Howard Bedford** (North American Aviation): Will the TX-2 Computer be used in any manner by the SAGE system?

**Mr. Clark:** The preliminary planning was done largely in the previous systems which were developed. The actual manpower which went into the design and building of the computer is roughly broken down: three engineers doing logical design; three engineers doing the memory work; and three engineers designing and building the hardware. This is besides shop facilities, drafting facilities, and a good number of very good technicians. It was approximately nine people for one year.

---

**TABLE I**

<table>
<thead>
<tr>
<th>Word Lengths (in bits)</th>
<th>Additions per second</th>
<th>Multiplications per second</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>150,000</td>
<td>80,000</td>
</tr>
<tr>
<td>18</td>
<td>300,000</td>
<td>240,000</td>
</tr>
<tr>
<td>9</td>
<td>600,000</td>
<td>600,000</td>
</tr>
</tbody>
</table>

---

**W. A. Farrand** (Autonetics Div. of N.A.A., Bellflower, Calif.): I would like to know what checking methods are used?

**Mr. Clark:** I am not sure just what checking methods are meant here. So far as the circuits go, we check the memory system by means of a parity loop; this is a very simple check. We expect the memory to be quite reliable in that we do check all of the core memories with the parity; otherwise, there is no checking while the machine is running. We have to check the machine before the machine is actually doing the programming, but this will be gone into later.

**L. Kolbo** (RAND Corp.): Does this machine make use of extract and deposit operations by use of masks?

**Mr. Clark:** The three floating functions that I mentioned originally, the and/or instructions, are not masked instructions; the only one which is, is the masked stored instruction, which is not, strictly speaking, a logical instruction but a digit and memory substitution type instruction.

**W. Heising** (IBM): What is the time to execute typical floating "add" (programmed)?

**Mr. Clark:** The reason why we did not wire floating-point operations into the machine is because we found that with configuration control it is very easy to program these instructions. Floating addition, for instance, takes 7 to 9 instructions, depending on how many there are going to be of them, to be actually executed in sequence to develop in 10 microseconds per instruction. That is, about 70 microseconds are required to execute an interpretive floating addition operation. Multiplication and division are much shorter: they take 3 or 4 instructions apiece.

**R. Frohman** (National Cash Register Co.): It appears that the TX-2 was well and thoroughly planned. Could you please indicate about what amount of time was spent in preliminary planning?

**Mr. Clark:** The preliminary planning was done largely in the previous systems which were developed. The actual manpower which went into the design and building of the computer is roughly broken down: three engineers doing logical design; three engineers doing the memory work; and three engineers designing and building the hardware. This is besides shop facilities, drafting facilities, and a good number of very good technicians. It was approximately nine people for one year.