COMPUTER PROGRAMMING AND CODING AT THE HIGH SCHOOL LEVEL

Aaron L. Buchman

This paper concerns a course in computer programming and coding being offered at the Hutchinson Central Technical High School, Buffalo, New York. The course, which runs one term and is open to seniors studying their fourth year of mathematics, was completed by the first group of pupils, and is now being given to the second group, most of whom intend to major in a branch of engineering at college.

About two years ago, the author broached the idea that the field of machine computation contained much material that could prove a challenge to high school seniors with a good mathematical background, and that such a course could interest pupils in a field that showed much promise of absorbing a large number of competent mathematicians. In order to make such a course concrete and provide motivation, he proposed that his pupils build an automatic digital computer. To be an effective teaching device, it was derirable that such a machine contain (1) an input unit able to read from a tape, (2) a control unit which could interpret the taped orders, especially those relating to branching routines, (3) a storage unit capable of storing several numbers, and (4) an arithmetic unit which could do rapid addition, subtraction, multiplication, and perhaps division. The Buffalo Board of Education made one thousand dollars available, and under the guidance of the author and using the plans and circuits he drew up, the pupils in his advanced mathematics classes built a machine having the above specifications. The construction and wiring of the automatic computer, an electromagnetic relay machine, took about six months, and was a fine experience for the pupils involved.

The computer built at the Hutchinson Central Technical High School operates in the binary number system. Its storage section consists of six registers, each of which can store a six digit binary number. A number which is transferred, remains in the original register until cleared from this register by a signal. However, transferring a number from one register to another which already contains a number, will "merge" the contents of the two registers. Panel lights indicate the contents of each register.

The arithmetic unit contains three registers, two operator-registers which hold the numbers that are to be processed, and one output-register which holds the results of the operation. Panel lights also indicate the contents of these registers. Three networks of relays perform six column addition, subtraction, and multiplication respectively, each process in a single operation. To keep the cost down, the division network was eliminated, and division in this machine is accomplished by a taped routine. However, a network for automatic division by two, that is, right shift of one place, was retained. Additional panel lights are used to indicate (1) overflow in addition and multiplication, (2) a negative difference in subtraction, and (3) a non-integral quotient in division by two.

The control section of the computer contains the routing circuits which translate the signals from the tape-reading unit into the various operations of the machine. This section also contains the circuits for testing for overflow, negative difference, and non-integral quotient; and the various gate circuits, which with the test circuits, are used to choose between alternate sub-routines, and thus permit branching to take place.

The tape-reading unit consists of ten rolling contacts, forming a ten channel reader which reads holes in the tape. The channels on the tape are three fourths of an inch wide, and with the three fourths of an inch margins along the edges, result in a tape which is nine inches wide. The details of the signals on this magnified tape are easily visible in a large room, which aids greatly in teaching this topic. Each tape contains an entire program, including all the sub-routines listed in sequence, if the program requires branching. Thus the tape-reading unit reads all the alternate sub-routines, but at each branch point, the control unit sets itself to accept the signals of one sub-routine and to reject the orders or signals of the alternate sub-routine. The periods of dead time which result, clearly indicate the branching in the program. Data and directions may also be fed into the machine manually through its keyboard.

This machine contains no output unit, the answers being read from the panel lights of the output-register. Thus the machine is quite limited in scope but does contain all the essential elements needed to teach concretely, on the high school level, a basic course.

The computer course, as set up by the author, consists of theory and laboratory work, taken alternately. The following syllabus indicates the contents of this high school course.

I THEORY
- A Number systems
  - (1) General construction
  - (2) Decimal system
  - (3) Binary system
    - (a) Conversion
      - (i) Decimal to binary
      - (ii) Binary to decimal
    - (b) Addition table
      - (i) Practice in binary addition
    - (c) Subtraction table
      - (i) Practice in binary subtraction
      - (ii) Complementing a negative answer
    - (d) Multiplication table
      - (i) Practice in binary multiplication
- B Computers
  - (1) History
  - (2) Parts of a modern automatic computer
    - (a) Input devices
    - (b) Storage devices and circuits
      - (i) Address
      - (ii) Transfer of information
    - (c) Arithmetic units
      - (i) Study of addition, subtraction and multiplication networks
    - (d) Control units
      - (i) Study of signal routing networks
      - (ii) Study of test circuits - overflow, negative difference, etc.
      - (iii) Study of gates

(e) Output units

C Programming for an automatic computer

(1) Problem analysis - formulas

(a) Study of program for finding the average of two numbers

(2) Study of possible unit directions - single operation

(3) Programming a simple type problem, such as finding the total area of a rectangular solid

(a) Sequence of orders

(b) Minimum number of orders

(i) Decrease resulting from transformation of formula

(c) Limitations

(d) Pitfalls

(4) Branching and recursive procedures

(a) Study of arbitrary programs containing branch points

(b) Study of program for determining the roots of a quadratic equation by successive bisections of the interval

D Coding

(1) Theory

(2) Study of code sheets

(3) Practice

II LABORATORY EXERCISES

A Pupil to program a simple formula, this program to contain one or more branch points

B Code the program obtained in A

C Pupil to cut a program tape using the coded sequence in B

The program for locating a root of $x^2 + bx - c = 0$, where $0 \leqq r \leqq 7$, is a typical program for this machine and illustrates the kinds of orders the pupils use in their programs, and cut on the tapes they construct. Use the keyboard to enter <u>b</u> in register one, <u>c</u> in register two, and <u>four</u> in register three. The root appears in register six after three cycles of the following taped orders.

1. transfer contents of register three to operator-register one
2. transfer contents of register six to operator-register two
3. merge contents of operator-registers one and two
4. blank operator-register two
5. transfer contents of register one to operator-register two
6. add
7. stop machine if addition overflows
8. blank operator-register two
9. transfer contents of output-register to operator-register two
10. blank output-register
11. multiply
12. stop machine if multiplication overflows
13. blank operator-register one
14. blank operator-register two
15. transfer contents of output-register to operator-register one
16. blank output-register
17. transfer contents of register two to operator-register two
18. subtract

19. blank operator-register two
20. transfer contents of register three to operator-register two
21. blank register three
22. lock control gate X if number in output-register is negative
23. blank output-register
24. divide contents of operator-register two by <u>two</u>
25. lock control gate Y if control gate X is open
26. open control gate X if it is locked
27. blank output-register
28. divide contents of operator-register two by <u>two</u>
29. merge contents of operator-register two and register six
30. open control gate Y if it is locked
31. transfer contents of output-register to register three
32. blank operator-register one
33. blank operator-register two
34. blank output-register

Visits to industrial establishments using automatic computers can do much to interest pupils in careers in the field of machine computing. The computing class at Hutchinson Central Technical High School has been very fortunate in being able to visit the Cornell Aeronautical Laboratory and study the various automatic computers in use there. Personnel and management at this plant were very cooperative in permitting pupils to program for their machines and to run these programs on the computers.

The author believes that offering a course such as is taught at the Hutchinson Central Technical High School, is a step in the direction of the fuller utilization of the scientific and the mathematical abilities of young people. He hopes that this paper indicates what has been done and what can be done at the high school level. Perhaps other schools can take up the ideas outlined in this paper and so give their pupils some very rewarding experiences in an interesting and important branch of mathematics, machine computation.