Division 6 - Lincoln Laboratory
Massachusetts Institute of Technology
Lexington 73, Massachusetts

SUBJECT: THE MULTI-SEQUENCE PROGRAM CONCEPT

To: R. R. Everett

From: W. A. Clark

Date: 5 November 1954

Approved: N. L. Daggett

Abstract: A computer program operating in a single-program-counter machine consists of a single control-sequence. It is possible to construct computing systems with several program-counters operating a multi-sequence program. In a system consisting of one computer operating many asynchronous terminal devices, multi-sequence operation is generally more efficient than one-sequence operation.

Introduction

It is the object of this note to extend certain ideas concerning the sequence structure of computer programs. In particular, the concept of a computing system operating a program with several independent sequences advancing concurrently is applied to the problem of centralizing the control of a set of asynchronous devices. It will be assumed that the reader is reasonably familiar with the principal logical features of computers of the one-address, stored-program class on which these ideas are based.

By "stored program" we mean an encoded sequence of control words held in an addressable storage element. In the usual mode of operation a control word is removed from this storage and transferred to a central control element which decodes the word and distributes command signals to the various elements of the system. In general, the control word will contain the address of another stored word on which specified operations are to be carried out in an arithmetic element or other operation element.

Control Sequencing

The process of obtaining the control word, decoding it, and carrying out the operations it specifies constitutes one step in the execution of a control sequence. Upon completion of any step in the sequence,

The research reported in this document was supported jointly by the Department of the Army, the Department of the Navy, and the Department of the Air Force under Air Force Contract No. AF 19(122)-458.

the address of the control word for the next step is held in a register variously called the "instruction counter" or "program counter". At the maximum operating rate, the advance to the next step in the sequence occurs automatically upon completion of the current step.

The advance rate can be reduced from this maximum and can also be made to depend on timing events in a device external to the central computer, e.g. a manually-operated pushbutton or even another computer. These timing events can occur aperiodically as long as the maximum rate is not exceeded.

Note that the program as we have described it consists of a single control-sequence. One number only, an address held in the program counter, is required to specify the next step of the sequence. This sequence may be very complex, involving cycles and alternative branches through many subsequences, but control will eventually pass through every subsequence if the program is operated long enough.

The Multi-Sequence Program

It is possible to construct programs of more than one sequence. Consider, for example, a computing system consisting of two intercommunicating stored-program computers. If it is assumed that each computer is operating a one-sequence program requiring to some extent the cooperation of the other computer, then it might be said that the computing system is operating a single two-sequence program. Two numbers, the addresses in the program counters, are required to specify the next double-step of the two-sequence program. This idea may be generalized to an N-sequence program operating in an N-computer system (assuming that there are no computer cliques within the system).

A multi-sequence program can also be constructed for a single computer. The general requirement is that the operation of one sequence must not interfere with the operation of any other. In general, this means that the operating registers of the computer must be time-shared by the sequences. A program counter must be provided for each sequence and the computing system must include an element for deciding which sequence is to be advanced during any given control cycle.

These sequences may be time-interleaved in an arbitrary fashion and each sequence can be advanced step by step at its own rate as determined by timing events in a separate clock associated with the sequence.

Application to a System with Asynchronous Terminal Devices

A multi-sequence programmed computer is well-suited to operation in a system with many asynchronous terminal devices. Consider the general control and communication requirements in this kind of system:

A typical terminal device can appear in one of several states at any given time. As an example, a tape unit may be running in reverse, it may be in an erase mode, etc. A control element associated with the device determines the transitions from one state to another and initiates data transfers between the device and the computer. In general, the timing of these transfer and transition events is critical within certain limits which depend upon the characteristics of the device and its particular application. For example, a tape reader must read only at those times when a line of data is under the reading heads and must transfer this data before the next line is read.

Now it is possible to simplify the system by centralizing to a large extent the control of many different devices. In particular, it is possible to incorporate many of these control functions into the control element of the central computer and to program the sequences of transitions and data transfers required to operate the devices. It is necessary only to provide the computer with a means of selecting one device at a time and a repertoire of orders for dealing with data transfers and transitions. The problem is that the existence of different timing requirements for a set of devices makes it difficult to incorporate the control sequences of the entire set into a single one-sequence program without sacrifice of computing efficiency. However, an efficient multi-sequence program can be constructed with a separate minor sequence devoted to the operation of each device and a major sequence which constitutes the main body of the program. The advance rate in any minor sequence can be tied to the particular timing requirements of the device being controlled by the sequence. The major sequence is advanced at the maximum rate possible in the remaining computing time. The timing problem is reduced to one of determining the number of devices which can be operated in the allocated computing time without exceeding the maximum rate of the system.

Many terminal devices are concerned almost exclusively with the transfer of data and can be operated without requiring all of the functions of an arithmetic element. Thus if a separate, simplified operation element is provided to deal with just those functions which are required, then the arithmetic element need not be time-shared between the major sequence and the minor sequences for these devices.

Time-Shared Sub-routines

If a sub-routine is constructed in such a way that none of its control-words requires modification during operation of the program, then it may be time-shared in an arbitrarily inter-leaved fashion among the sequences of a multi-sequence program. It is generally possible to do this given enough separately-stored "index-registers" which modify any operand addresses referred to by the control-words of the sub-routine without altering the control-words themselves. Each sequence using the sub-routine then forms its own operand addresses by referring to its own set of index-registers.

In the case of a computing system with many terminal devices, the possibility of time-sharing sub-routines may mean a considerable saving in program storage space if the system includes several devices which require identical control sequences but different operands. By providing a set of index-registers for each device, only one set of control words need be stored.

As an extreme case, a computing system can be constructed which consists of several identical computers in various phases time-sharing the same program but operating on different sets of data.

Conclusion

Multi-sequence operation offers advantages in the centralization of control in a system consisting of a computer operating asynchronously with many terminal devices. The general concepts of a multi-sequence program may also be of some help in visualizing the operation of a computing system with several intercommunicating computers.

Signed: W A Clark
W. A. Clark

WAC/ahm

Distribution:

D. R. Brown
S. H. Dodd
B. G. Farley
J. W. Forgie
F. E. Heart
R. P. Mayer
T. H. Meisling
K. H. Olsen
W. N. Papian
J. I. Raffel