UCRL-16199

University of California

Ernest O. Lawrence Radiation Laboratory

TWO-WEEK LOAN COPY

This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 5545

TRIST--A TRI-LEVEL MULTIPROGRAMMING EXECUTIVE INVOLVING REAL-TIME CALCULATIONS AND AN ASYNCHRONOUS ON-LINE DATA SOURCE

Berkeley, California

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

For 1965 Fall Joint Computer Conference

UCRL-16199

UNIVERSITY OF CALIFORNIA

Lawrence Radiation Laboratory
Berkeley, California

AEC Contract No. W-7405-eng-48

TRIST--A TRI-LEVEL MULTIPROGRAMMING EXECUTIVE INVOLVING REAL-TIME CALCULATIONS AND AN ASYNCHRONOUS ON-LINE DATA SOURCE

Carol Osborne, Robert Larson, Thomas Oliver,
Joan Stekler, and Howard S. White

June 15, 1965

TRIST--A TRI-LEVEL MULTIPROGRAMMING EXECUTIVE INVOLVING REAL-TIME CALCULATIONS AND AN ASYNCHRONOUS ON-LINE DATA SOURCE

Carol Osborne, Robert Larson, Thomas Oliver
Joan Stekler, and Howard S. White

Lawrence Radiation Laboratory
University of California
Berkeley, California

June 15, 1965

Abstract

TRIST is a TRI-level multiprogramming SysTem executive written at Lawrence Radiation Laboratory. The TRIST executive is designed to coordinate computer operations with a real-time asynchronous device attached on-line to an IBM 7094-II and to make the computer time between real-time demands available for other uses. The asynchronous device operation requires that the computer be available to accept data when the device is transmitting. Data are transmitted to alternate core buffers. The computer must immediately interchange buffers and process data from one buffer while the other buffer is being filled; however, a large portion of the computer time is also available for processing general programs. Real-time demands for CPU control are met within 500 microseconds. The three priority levels of programs controlled by TRIST are: the A-level or real-time program, which receives data from the on-line device; the B-level programs, which process the data produced by the A level; and the C-level programs, which are general production and debug.

Introduction

TRIST is a TRI-level multiprogramming SysTem executive written at Lawrence Radiation Laboratory. The TRIST executive is designed to coordinate computer operations with a real-time asynchronous device attached on-line to an IBM 7094-II and to make the computer time between real-time demands available for other uses. The asynchronous device operation requires that the computer be available to accept data when the device is transmitting. Data are transmitted to alternate core buffers. The computer must immediately interchange buffers and process data from one buffer while the other buffer is being filled; however, a large portion of the computer time is also available for processing general programs. Real-time demands for CPU (central processing unit) control are met within 500 microseconds.

The three priority levels of programs controlled by TRIST are: the A-level or real-time program, which receives data from the on-line device and has highest priority; the B-level programs, which process the data produced by the A level and have second-order priority; and the C-level programs, which are general production, preproduction, and debug, and have lowest priority.

Machine Configuration

The executive is written for an IBM 7094, model II, with two 7302 core storage modules (65K), and six data channels: one with a 1301 disk file, one with an LRL-built interval timer which traps every 137 milliseconds, one with the on-line device, and three tape-unit channels, one of which also has a printer, punch, and reader. See Fig. 1.

Background-FSD

The on-line device is a Flying-Spot Digitizer (FSD), which is used to digitize pictures of nuclear interactions occurring in bubble chambers. The FSD has buffers that collect coordinates of images along a line on the film and transmit these data to the computer. Scan lines containing about 40 words of data are produced every 2 milliseconds. Once the scan of a picture is begun, the FSD operates asynchronously, and nearly 100 000 words of information may flow into the computer during the several seconds required to measure a large picture. Since the storage available for these data is only a few thousand words, the data must be analyzed and reduced in real time while they are being transmitted.

The real-time program reads data into alternate buffers, processing the data in one buffer while the other buffer is being filled. There are two logical delay periods in the real-time program. One occurs when the program has processed one buffer of data and is waiting for the other buffer to be filled. The other, longer delay occurs when the FSD is advancing the film to the next picture. These CPU times are made available by TRIST for other uses.

Background-System

TRIST maintains complete compatibility with the monitor system which preceded it so that none of the pre-TRIST programs have had to be modified.

The main production programs spatially reconstruct and kinematically analyze bubble chamber events, and require 15 sec of CPU time per event. (An event is defined as an interaction in a bubble chamber, and is seen in the three stereo views as particle tracks in the form of a string of bubbles.) These programs are run as B level when the data are from the FSD, and are run as C level when data are from other film-measuring devices.

TRIST Structure

TRIST logic occupies 20_8K locations of Core A, and TRIST input-output routines plus Core A/Core B communication linkages occupy 5_8K locations of Core B. (See Fig. 2.) The A-level or real-time program occupies Core A from 20_8K up, and the current B- or C-level program occupies Core B from 5_8K up.

TRIST, a TRIST loader, and all programs are stored permanently on the 1301 disk file. To save time and eliminate large amounts of tape I/O (input-output), A- and B-level output data are stored on the disk. The executive maintains storage communication links between A- and B-level programs and among the B-level programs.

TRIST Flow

The general flow chart in Fig. 3 shows that a one-card loader calls a small loader which, in turn, calls TRIST into core. TRIST then calls in and gives control to the A-level program, which starts the FSD processing. The A-level program reduces FSD data, provides diagnostics for pictures rejected, and gives on-line quality control summaries to the operator. A-level output is stored in batches of 15 events on the disk. Whenever the A-level program is not busy, it gives control back to the executive.

TRIST then calls in, initializes, and stores up to twelve B-level programs on designated areas of the disk, from which they are recalled as needed. After all B-level programs have been initialized and stored on the disk, TRIST determines priorities and decides which program will be given control. The controlling program may be a B- or C-level program or the executive itself. (a) If A level has a batch of data ready, TRIST initiates a B-level cycle during which all B-level programs will be run in order on that batch of data. If A level

acquires another batch of data before the B levels complete the current batch, A-level processing is temporarily inhibited. (b) If no data are ready and the B-level cycle is finished, TRIST gives control to C level. The executive and all non-real-time programs, of course, are interrupted whenever the A level requires priority, but the C level also may be interrupted and stored on a disk area whenever there are data for the B levels. When the B levels are finished, the C-level program is then recalled and continued. (c) If B levels are finished and there are no C-level programs, TRIST retains control and waits for more data via a pause loop.

Special Features

Program decks

TRIST can control any combination of the three levels at one time, except that B level requires data from a concurrent A-level program. (See Fig. 4.) The levels and programs to be run are set up as a deck of macro instruction cards which are read by TRIST. Each program deck consists of a program call card, which specifies level and program; a time-charge card; program modifications, if any; and an execute card. The A-level program is always the first program call deck in a run. It is followed by the B-level program decks, if any, and then the C-level decks. Prior to processing the first C-level program, all C-level program decks are read and stored on a disk area, thus freeing the card reader. C-level program calls are than read from the disk when C level has priority.

Priority and entry vectors

TRIST controls the priorities of all three levels via a priority word and a set of flag words which determine the status of data, of the string of B-level programs, and of the current C-level program. The levels are maintained and

entered via a set of four entry vectors (real-time, non-real-time, executive, and core B) in which machine conditions and identification information are saved. The core B (B- and C-level) and executive entry vectors are moved in and out of the non-real-time entry vector as program control is exchanged; thus the non-real-time entry vector contains information pertaining either to the executive or to the core B program. There is only one core B program at any one time, and therefore only one core B entry vector. TRIST may have retired a C-level core B program and may be processing a string of several B-level core B programs, therefore each core B entry vector is stored and recalled with its program.

Time accounting

A time-accounting feature is incorporated into TRIST via a set of clock vectors which are stored and recalled with the individual programs as are the entry vectors. Real-time demands cause TRIST to give control to the A-level program at intervals of approximately 200 msec. As TRIST shifts control among the programs and within itself in program loading and core-B program exchanging, times are saved in the appropriate clock vectors. The executive then produces a final run summary for itself and for each program, including level, time-charge number, time (starting, ending, and used), run type (debug, production, preproduction, etc.), and a completion code (normal, interrupted, restarted, etc.). This run summary information is also stored on the disk for use by a time-accounting program which produces weekly, monthly, and yearly time-accounting data.

Memory-protect package

In addition to its obvious function of protecting the executive from user errors, the memory-protect package is the vehicle for system compatibility between the previous system and TRIST and for multiple subroutine use sequencing. Since all of the standard I/O routines are part of the executive under TRIST, a memory-protect trap occurs when any one of these routines is called,

and the memory-protect package simply reroutes the call to the new executive subroutine.

Multiple-call protection

The memory-protect package also protects all the executive subroutines from multiple calls. Since there are three levels of programs operating, it is possible that the C-level program, for example, will have been in the middle of a subroutine when the higher priority A-level program was given control of the CPU. If the A-level program needed to use this subroutine, there would have been a conflict, but since subroutine calls go through the memory-protect package, the conflict is eliminated. The prior use of the subroutine is brought to completion, the machine conditions of the subroutine exit are saved in the entry vector, and the current call is then allowed to go through. Although TRIST may be running three levels and therefore many programs all together, there are really only two running at any one instant in time (the real-time program and the non-real-time program), thus there is no problem of knowing where to save machine conditions. Furthermore, since the real-time program may never request use of a system subroutine during its actual real-time operation, the above procedure solves all conflicts of calls and does not interfere with the real-time demands.

Sense-switch usage

TRIST reserves all the sense switches for its exclusive use. The console keys are used as sense switches by each program (ten are provided for each level). One sense switch is used to indicate that the keys are being used for this purpose. The only time this switch is off is when the operator needs to indicate the location of a transfer, in which case this key is off and the transfer command is in the keys. Three sense switches designate the program for which operator action is intended (A-level, B- or C-level, or the

executive). A fifth switch is used to interrupt C-level processing. The current C-level program is finished and further C levels are ignored.

Sense lights

The sense lights give an indication of the level that currently has control. One is used for the real-time program, one for B level, one for C level, and one for pause; all off is used to indicate executive control.

Interval timer

Attached to Channel H is an interval timer, which causes a data-channel trap to occur every 137 msec. The timer service routine examines the current machine operating conditions for halts and loops, allows (by means of sense switches) program pauses for operator action (tape requests, etc.), and allows direct operator action (to force a program to pause, to force a restart in a program, to force a dump and interrupt a program, etc.).

Halts

When halts occur in debug-mode operation, a dump is taken and the next program is called. When they occur in production programs, the timer routine attempts to restart the program.

Loops

Loops are detected by an I/O timer word which is part of each program's entry vector. Since all programs are of a data-processing nature, a maximum permissible elapsed time (approximately 4 sec) between I/O requests can be established. The I/O timer word is incremented at each timer trap and cleared when an I/O request is recognized. When the timer word exceeds the established maximum, a statement is printed on-line and the computer operator may take corrective action.

Program operator action

When a program requests operator action, it prints a request on-line

and goes to a "pause" subroutine. (Pause subroutines are provided because real-time demands interfere with halts and because, as stated above, TRIST checks for halts and assumes they are errors.) The operator performs the requested action and puts the specified sense switch down and up. When control has returned to the specified level, the sense switch action is recognized and processing continues.

Direct operator action

When the operator initiates action, he simple puts down the appropriate sense switch which causes acknowledgment and a pause in the specified level only. At this point, the operator may put the sense switch up or may indicate a transfer by entering a command in the keys and then putting the sense switch up. Important communication links exist between the memory-protect routine in its feature of multiple subroutine call protection and the timer service routine when it allows direct operator action. Depending upon the status of the memory-protect routine, the timer routine may not immediately execute a desired transfer, but may set the location for the memory-protect routine to execute.

Skipping C-level programs

TRIST incorporates a provision for skipping either backward or forward in the C-level program deck. A specified console key ("C-LEVEL SKIP") may be put down to cause identification of the next C-level program in the program deck to be printed on-line with its serial number indicating its relative position in the program deck (the fifth program to be run has serial number 5). The operator may select that or any other program in the deck by indicating its serial number, or may terminate all C-level programs by selecting serial number zero.

The actual skipping procedure is as follows. The operator puts down the C-LEVEL SKIP key and reads the on-line for program and serial number. He simply puts the SKIP key up if the desired program was indicated on-line. If he wishes to select a different program, he puts down a C-LEVEL SELECT key, puts the desired serial number in the keys, puts the SELECT key up, and reads the printer message. When he sees the desired program, he puts both the SELECT and SKIP keys up.

Reading additional C-level decks

When all C-level programs have been run or if no C levels were originally read, a C-level deck can be read via a MORE CARDS console key. This key initiates the reading of C-level program decks onto the disk from which they are read as C level gets priority. This means, for instance, that an A-level run can be started and successive C-level runs can be initiated during the continuing A-level run. If desired, a C-level run can be terminated and a higher-priority C-level run can be initiated without interrupting the other levels.

Conclusion

Approximately 17 man-months of effort have gone into the development of the TRIST executive system; however, in one month, the computer time saved by the multiprogramming operation has equalled the development cost. This executive, which has been in operation since February of 1965, successfully coordinates computer operation with the on-line, asynchronous FSD, and simultaneously makes available nearly 60% of the CPU time for other use.

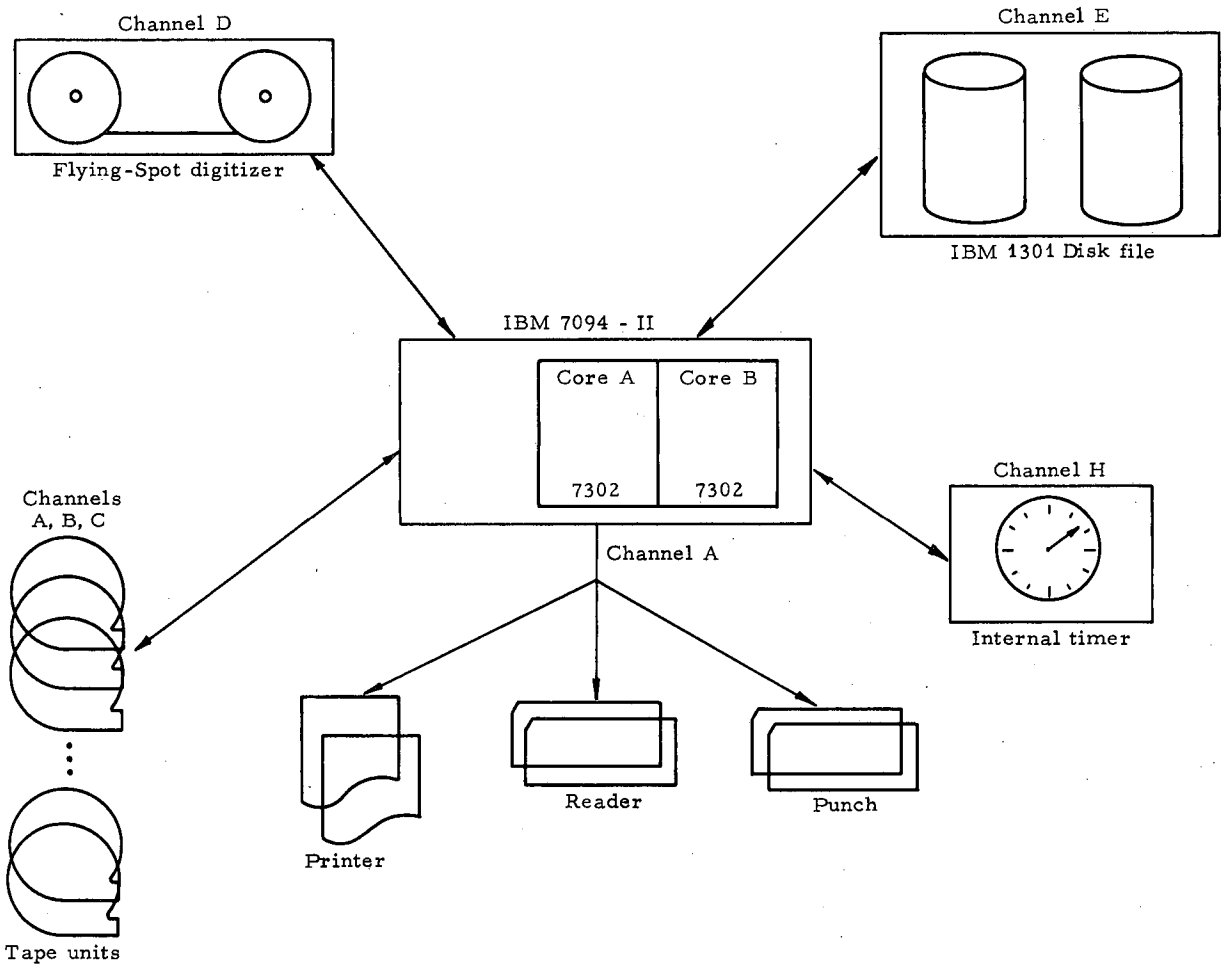

Fig. 1. Machine configurations.

MUB-6646

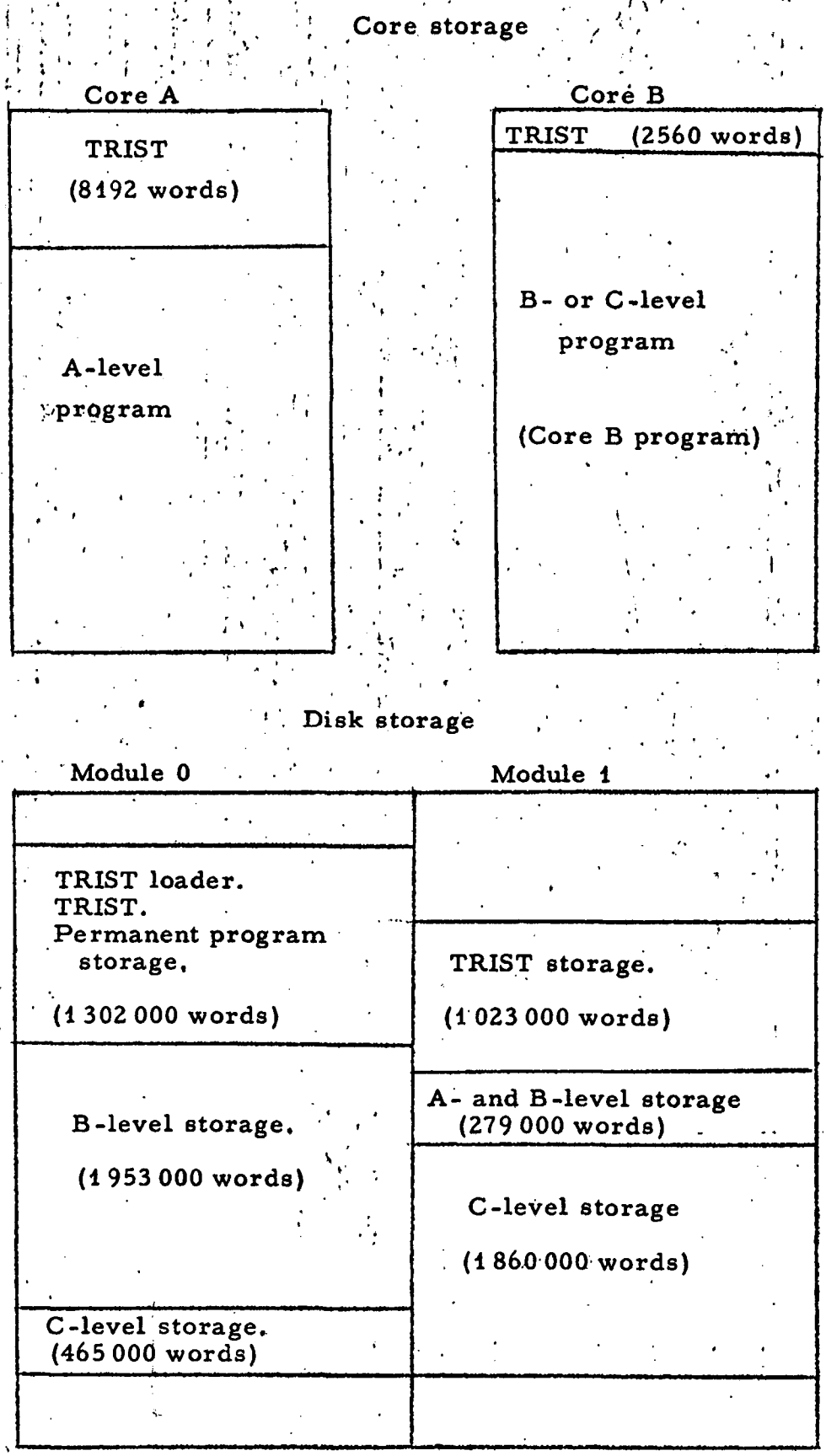

Fig. 2. TRIST structure.

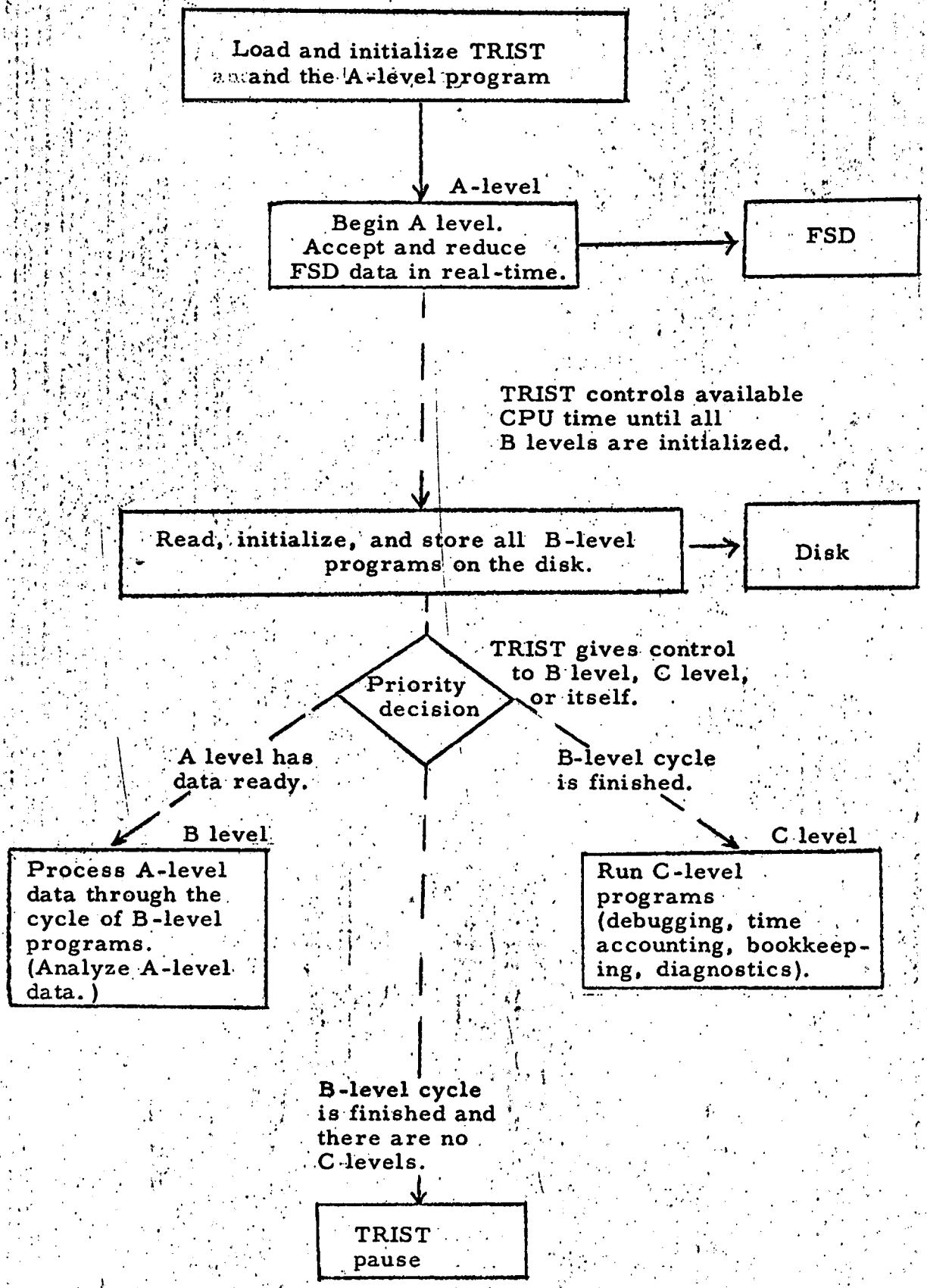

Fig. 3. TRIST flow.

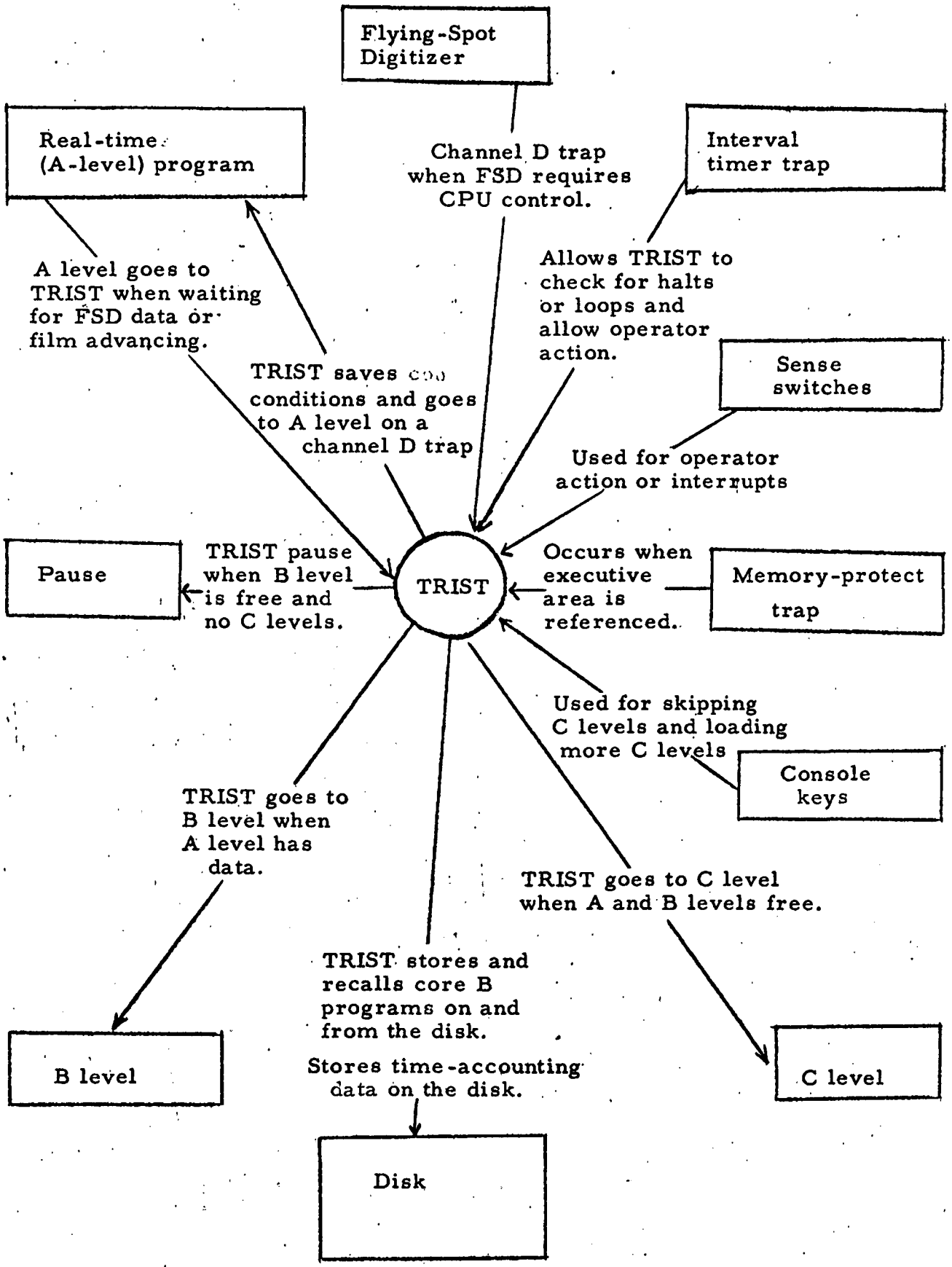

Fig. 4. Special features

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.