ADVANCED AVIONICS BREADBOARD

EXECUTIVE DESIGN AND IMPLEMENTATION

(NASA-CR-120416) ADVANCED AVIONICS BREADBOARD EXECUTIVE DESIGN AND IMPLEMENTATION (International Business Machines Corp.) 27 p HC $4.50 CSCL 01C

N74-31533

Unclas
G3/03 47334

NAS8-14000
IBM No. 72W-00413

Prepared by: L. R. Adams
IBM Corporation
Federal Systems Div.
Huntsville, Ala. 35805

TABLE OF CONTENTS

Introduction

The Advanced Avionics Breadboard (AAB) Executive evolved from an effort to design and develop an avionics system satisfying requirements specified by the Marshall Space Flight Center. This executive is unique in that it supervises a triple redundant avionics computer system. Three IBM System 4 Pi/CP-2 computers, operating synchronously and executing identical software, comprise the central processors which route data to and from a data bus via an Input/Output Controller. The executive's basic function is to provide application programs with an efficient software structure within which to perform specific avionics application tasks. Although implemented in a triplex Data Management System, the AAB Executive contains the flexibility to be adapted to other systems with minimal change.

Design Concepts

The executive design concepts were developed in accordance with the following considerations:

- To comply with hardware constraints
- To provide inter-program communication between the executive and application programs
- To support predefined time interval processing
- To minimize the impact on software caused by hardware changes
- To enable independent integration and checkout of executive functions and application programs

The modular approach to program design was implemented to facilitate future systems analysis studies by minimizing re-programming efforts to meet evolving baseline requirements. Significant features of the Data Management System (DMS) are the data base, minor-major loop transition function, and prime computer concept.

The AAB data base consists of the interrupt pointers, the I/O communication buffer, and the Communications Vector Table (CVT). These are the only address-dependent modules of the executive. The interrupt pointer addresses and the I/O communication buffer addresses are requirements imposed by the computer architecture. The I/O communication buffer is the limited area of memory addressable by the I/O controller.

The CVT contains the following general types of system information:

- Dynamic system profile and status indicators
- Pointers to I/O communications areas
- Pointers to executive service routines
- Commonly used masks and constants
- Miscellaneous system switches, indicators, and addresses

The executive system controls execution of application programs using an avionics minor-major loop design concept, depicted in Figure 1. The minor loop encompasses such functions as guidance, navigation, and control which in an avionics environment must be performed at known time intervals. Execution of the major loop segments follow each minor loop. To transfer control between the minor loop and a variable number of major loop application segments requires an efficient transition technique. The technique implemented in the AAB DMS is based on the use of a schedule table driven program segment initiator. The minor loop executive functions execute each minor-major loop computation cycle. The minor loop is initiated every 20 milliseconds by the executive upon receipt of an external synchronization interrupt generated by the I/O controller. Due to the cycle stealing requirements of the I/O controller and differences in CPU clock pulse oscillators, this permits closer synchronization than can be maintained under control of software logic. The system is required to be in the machine wait state prior to responding to this synchronization interrupt. The DMS software design reflects this requirement by entering an error routine if the interrupt is received while a program is in execution. The technique used to insure that the computers are in the wait state prior to receipt of the synchronization interrupt is to subdivide each major loop application program into segments which would not overrun the defined 20 ms time slice. This was accomplished by defining a segment's end to be coincidental with the issuance of a call to the executive I/O service routine. A "dummy" entry pointer was provided in the I/O routine to permit a program to relinquish control if it did not have any I/O to be processed. The Transition Program receives control after completion of minor loop processing and makes the final chain connection between minor loop output commands and commands chained for the previous major loop program segments. After clearing the I/O command status word area, the I/O operation is started, the schedule table is accessed, and control is passed to the next program segment.

Once minor-major loop processing begins, the executive scheduling technique interleaves application program execution in a multiprogrammed environment.

Figure 1 illustrates the functional relationship between the executive, minor and major loop programs, and corresponding program processing flow of a single computation cycle.

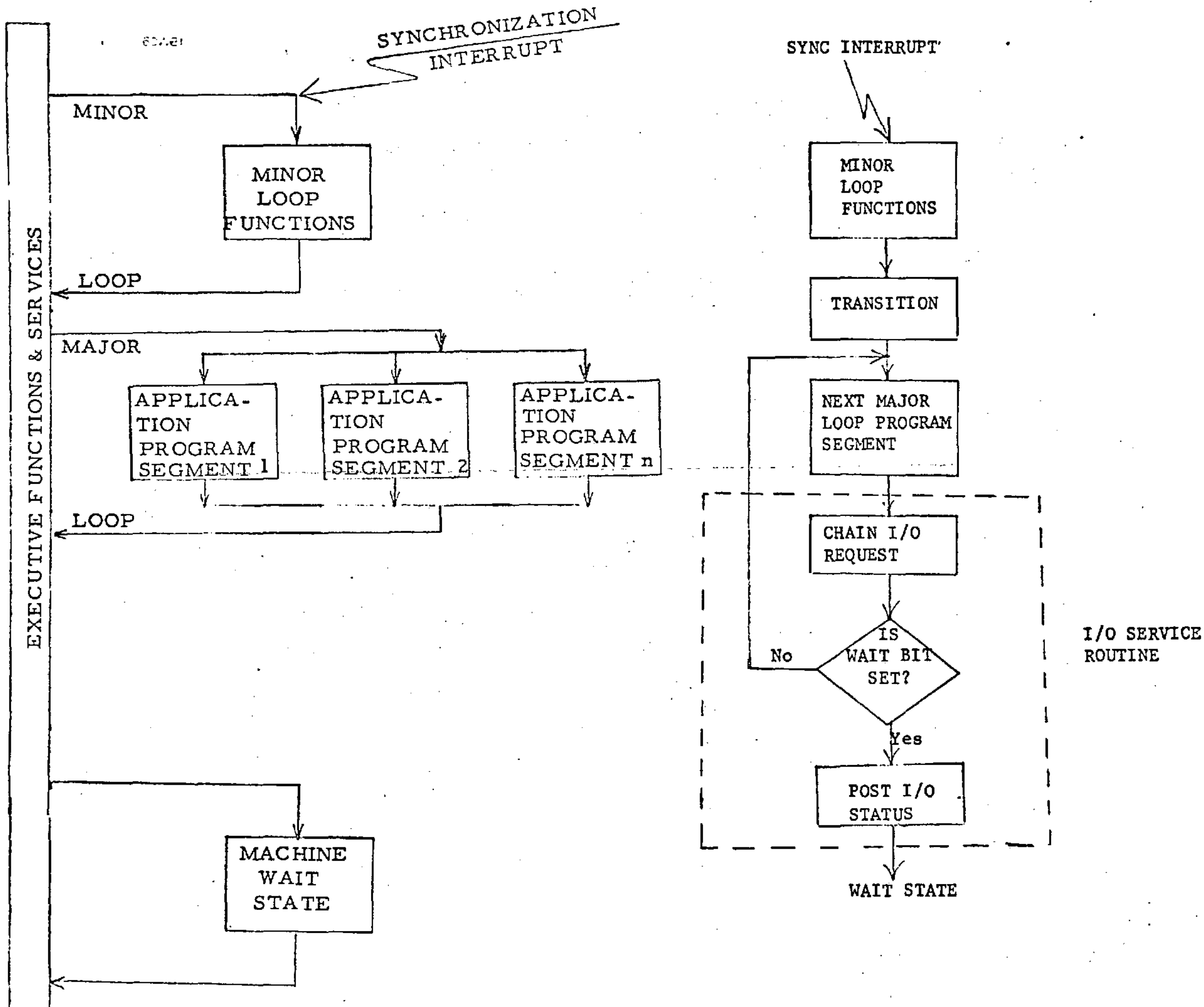

Figure 1. Functional System Design and Single

To assure the validity of all data being output from the computer system a prime selection concept was introduced. The prime computer concept designates one computer to control all output transmitted through the I/O controller. All computers capable of transmitting data to be processed by the I/O controller are defined as active. When I/O has been initiated, data from the active computers is compared within the I/O controller and the data from the prime computer is output to the data bus. If a miscompare occurs during processing by the I/O controller, subsequent commands in the I/O chain and their relative data continue to be analyzed and output if they compare. The executive software maintains the operational status of the computers when the I/O controller comparator logic indicates a failure. Active and/or prime computer reconfiguration is performed as follows:

- If data from the prime computer miscompares, the computer is allowed to remain active but the next computer in sequence is selected as prime. If the system degrades to the status where no computers remain qualified to function as prime the executive indicates the condition and transfers control to its abort function which terminates execution.
- If data from a non-prime active computer miscompares, that computer is disqualified to function as a prime or active computer.

If only two computers are active when a miscompare occurs, the prime output is assumed valid. In this case, the non-prime active computer is made inactive. System messages indicate all reconfiguration action.

Overall Structure

The executive receives control when a power on interrupt (or CPU system reset) occurs. Both hardware and software initialization functions are performed before entering a machine wait state with only the interrupts which establish CPU identity enabled. This state is maintained until a System Interrupt Switch is depressed to initiate system execution. Each computer is interrupted from the wait state to establish its identity and initialize internal timers before enabling interrupts, including the sync interrupt which occurs every 20 milliseconds. The identity of each computer is maintained by a software indicator in the CVT.

The executive performs the following hardware and software initialization functions prior to the first minor-major loop:

- Reset CP-2 interrupts, registers, and discretes
- Initialize dynamic area of the Communications Vector Table (CVT)
- Set application program starting addresses to the initial entry point in all schedule table entries
- Set schedule table pointer in the CVT to the first logical entry
- Enable CPU identity interrupts
- Enter wait state
- Establish computer identities upon receipt of external interrupt
- Reset interval and GO/NO-GO timers
- Enable interrupts, including sync interrupt
- Issue I/O controller master reset
- Enter wait state

The system initialization sequence includes checks to determine which computers can successfully operate as prime. The executive software then selects the prime for system I/O from the available primes in the order 1-2-3. Indicators reflecting this selection as well as which computers are active are maintained in the CVT.

Executive supervisory functions performed every minor loop include:

- Sync interrupt processing
- Prime selection (reconfiguration if necessary)
- Timer support (reset of internal clocks and maintenance of mission elapsed time)
- I/O initiation for current minor loop and previous major loop segment
- Mode selection
- Transfer of control to next application program segment

After the minor loop programs have executed and the output data has been prepared, the Transition Program receives control to start the I/O operations. The Transition Program, via the schedule table, determines the major loop segment to be executed and passes control to that segment.

Modes of Execution

The executive controls the execution of minor loop functions and of each application program in one of three modes:

- Initialization
- Operational
- Off-line

The initialization mode is entered initially to establish the integrity of the on-line configuration.

In the operational mode, the minor loop functions include a self-test routine to verify the operational capability of the computer system logic and a system pulser which monitors the operational status of each device in the system.

The off-line mode differs from the initialization mode only in that a specified application program is executed continuously without sequencing through other programs in the major loop.

Upon entry to each application program segment, the accumulator contains an indication of the current mode. Program sequencing within any mode selected will restart following the termination of all major loop programs unless a new mode selection is pending. Mode changes are initiated by operator input via the typewriter or display unit.

Figure 2 provides a general outline of the processing sequence in the operational mode.

MINOR LOOP

- PROCESS SYNC INTERRUPT
- SELECT PRIME (RECONFIGURE IF NECESSARY)
- UPDATE TIME AND RESET TIMERS
- MONITOR OPERATIONAL STATUS OF EACH DEVICE
- INITIATE MINOR LOOP APPLICATION PROGRAMS (GUIDANCE, NAVIGATION, AND CONTROL)

TRANSITION PROGRAM

- INITIATE I/O FOR CURRENT MINOR AND LAST MAJOR LOOP PROGRAMS
- SELECT MODE
- DEFINE NEXT SEGMENT FOR EXECUTION

MAJOR LOOP

- EXECUTE ONE OR MORE APPLICATION PROGRAM SEGMENTS

POSTING ROUTINE

- POST CURRENT I/O REQUESTS
- ENTER WAIT STATE

Figure 2. Operational Mode Processing Functional Sequence

Program Sequencing

A table-driven scheduling technique was designed and implemented in the AAB system to control execution of application program segments in the correct sequence and current mode selection. A major loop segment is defined as application program execution from:

- the initial entry point through the first I/O call or termination request.
- an I/O call through the next I/O call or termination request.

The schedule table, incorporated into the executive data base, contains one entry for each application program. These entries provide communication and linkage between the executive and application programs. The schedule table entries are chained together by pointers denoting the next sequential physical entry, but the execution sequence for the application program segments is determined by pointers which specify the next logical entry. Since an entry may have multiple logical pointers, each of these pointers has an associated relative exit number. The relative exit numbers, 0 and subsequent numbers in increments of 2, specify which logical pointer is to be accessed in association with the next logical schedule table entry. Each entry provides space that serves as a "save area" for index registers and the instruction counter when the executive I/O routine is called by the application program. After every I/O call, the I/O routine updates the CVT word which contains the pointer to the next logical schedule table entry and its relative exit number. The routine follows this pointer to the next logical entry where it accesses the values to restore registers and the instruction counter, thus giving control to the next program segment in sequence. Any desired sequence of segment execution can be attained by the arrangement of pointers and their relative exit numbers.

Figure 3 illustrates the general format for a schedule table entry and exemplifies a system composed of programs A, B, and C. The arrangement of pointers and exits will cause the programs to gain control in the sequence: A, B, A, C, A, C, A,...

When all application programs have completed execution, the major loop is said to be complete.

Application program identification consists of a decimal value from Ø to 31. Unique to each program, the identification number is used to access the corresponding entry for programs to be bypassed, activated, or executed off-line.

Examination of the schedule table reveals at all times the progress made in executing all major loop application programs. The second word of each entry serves as a group of flags, or indicators, accessed and maintained by various executive routines. Only five of the sixteen bits available have been allocated. They supply the following information:

- The "wait bit" signifies that no further major loop execution should take place until the I/O operations of the current time interval have completed and the next sync interrupt occurs.
- The "terminate bit" indicates that the particular application program has terminated.
- The "bypass allow bit" specifies whether or not the application program can have execution bypassed via system parameter.
- The "off-line bit" reflects a request to execute the related application program in the off-line mode.

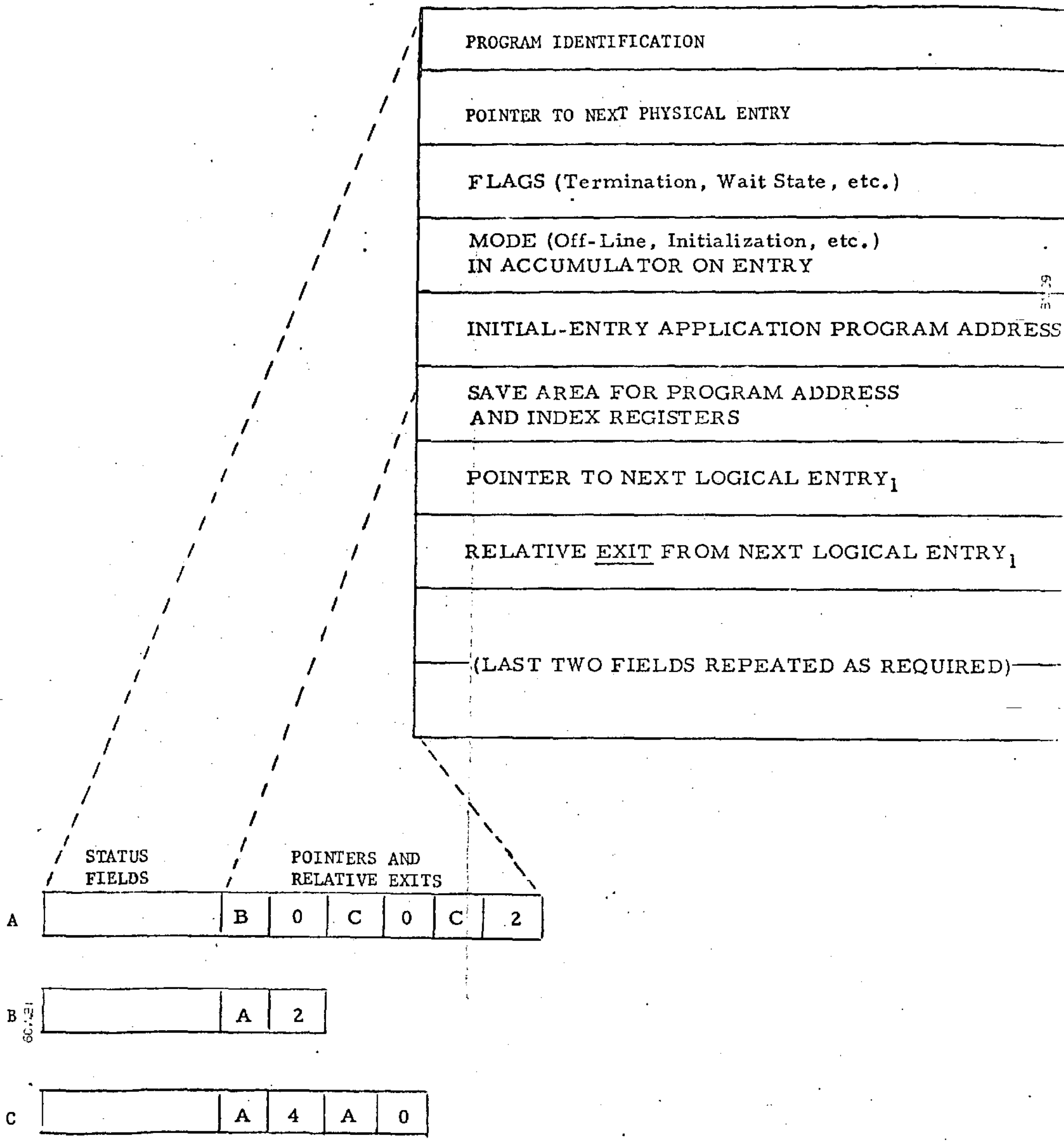

Figure 3. Schedule Table Illustration

The method of recording the status of task completion employs a location in the CVT each of whose bits correspond to a schedule table entry. Every minor loop, the Transition Program compares this 32-bit word to a predefined mask reflecting those application programs expected to terminate.

An application program may expand over multiple major loop segments and may require multiple minor-major loop computation cycles to complete. The interleaving of more than one application program in this manner contributes to maximizing utilization of the computers.

One powerful capability provided by the table-driven scheduling technique is the ability to execute an application program to completion before the next logical application program gains control. This is accomplished by modifying the CVT word which contains the relative exit for the application program currently executing. In the following example this technique is exercised for program B.

A		B	2	C	Ø	C	2
B		B	0	A	2		
C		A	4	A	Ø		

When program B initially gains control, the CVT relative exit value of 2 is saved and replaced with Ø so that program B will regain control following the sync interrup The program will continue to receive control after each sync interrupt until the CVT relative exit of 2 is restored.

Significant advantages of the schedule table and table-driven initiating technique are:

- Capability to independently develop and integrate an application program into the system
- Simple approach to application program segment definition
- Flexibility in application program sequencing
- Lower system overhead than dynamic task selection systems
- Centralized system data base
- Application oriented program control information
- Use of system macros to insure consistent schedule table generation

I/O Supervision

I/O supervision is performed by the I/O Service Routine and the Minor-to-Major Loop Transition executive functions.

The I/O Service Routine chains I/O controller commands together in main memory, thus controlling a critical resource: the area of memory addressable by the I/O controller. The I/O Service Routine saves the execution state of one program segment and gives control to the next segment, following the chain established in the schedule table until encountering an entry with the "wait bit" set. Upon detection of the wait-state indicator, the I/O Service Routine accesses the Posting Routine for analysis and posting of the I/O status for the previous I/O command chain. The I/O controller returns an I/O status word as each command word in a chain is processed. The Start I/O command initiates the I/O sequence and provides the starting address for storage of all I/O status words. Since the I/O Controller stores all I/O status words for an I/O chain in only one buffer area, the Posting Routine assumes the central responsibility for I/O error reporting. After analyzing the I/O status words, the Posting Routine reconfigures the system if necessary and stores the I/O status words in locations known to the respective application programs to enable subsequent I/O analysis. When making a call to the I/O Service Program each application program supplies a status mask to indicate what it expects the I/O status word to be. The rationale for having application programs supply this mask is to allow diagnostic programs to induce deliberate errors without causing the executive to reconfigure as a result of the errors.

The function of the Transition Program in the I/O process is to chain the minor loop output commands with the commands chained by the I/O Service Routine during the previous major loop, clear the I/O status area, and start the I/O operation.

Interrupt Processing

The function of the executive interrupt processing routines is to service the interrupt conditions defined within the functional areas specified below:

- Externally generated Input/Output Controller interrupts
  - System Interrupt - This generates a unique interrupt for each computer. When the identities have been established, the remaining interrupts are enabled so that minor-major loop processing can begin.
  - Synchronization Interrupt - This occurs at fixed intervals to allow software synchronization. If the system is not in the machine wait state upon receipt of the interrupt, and if the typewriter routines are not in progress, control is passed to the executive abort routine. If the typewriter routines are executing when the interrupt occurs, the timer support functions are performed and control is immediately returned to the location where processing was interrupted.
  - Typewriter Input Interrupt - A switch on the typewriter unit causes this interrupt to indicate a request for typewriter input.
  - Restart Interrupt - This simulates the power-on interrupt.

- Internally generated computer interrupts
  - Power On Interrupt - This initiates hardware and software initialization functions. All interrupts except the three "identity interrupts" are then inhibited to insure that computer identity is established before minor-major loop processing begins.

- Interval Timer Interrupt - This will occur only if the I/O controller synchronization interrupt failed or if I/O controller synchronization is simulated. If the timer interrupt resulted from a synchronization interrupt failure, an error indicator will be set and control passed to the executive abort routine.
- Storage Protect and Machine Check Interrupts - Error indicators will be set and a call made to the abort routine.

- Unused System Interrupt Conditions available in the computer
  - Unidentified interrupts will cause an error indicator to be set before control is transferred to the abort routine.

Executive Service Routines

The Executive Service Routines provide a base for application and executive program development. They execute as requested by application or executive programs to provide I/O supervision, application program sequence modification, and services common to several applications. In addition to the I/O Service Routine, routines are available to perform the following functions:

- Abnormal termination - All interrupts are inhibited except the "restart" interrupt, and execution halts. The calling address is placed in the accumulator in order to trace which unrecoverable error occurred.
- Data conversion (hexadecimal to EBCDIC, EBCDIC to hexadecimal, and binary to decimal)
- Data transfer - A specified number of data words is copied from one area of memory to another.
- Discrete processing - Discrete outputs are set or reset according to the parameter received from the calling program, and the current discrete output status is maintained in the CVT.
- Peripheral device address translation - Each I/O command contains a device address which directs the device to accept and transmit data. Symbolic device addresses assembled by the application program are translated to actual addresses contained in a table in the executive data base. To avoid possible retranslation during subsequent execution, a bit in the address format is used to indicate whether the address is symbolic or actual.

- Memory propagation - A specified halfword of data is propagated throughout a specified area of memory.

- Message stacking - Messages to be output via the typewriter or display unit are processed and placed in a buffer to be accessed when the corresponding executive application program receives control.

- Normal termination - The appropriate termination bits are set in the CVT and schedule table before control is given to the next schedule table entry.

- Queueing - To facilitate the processing of stacks or lists of data an element is removed from the beginning of one queue and attached to the end of another. Directory and link pointers are updated accordingly.

- Timer support - The internal timers are reset and, depending on the parameter supplied by the calling program, mission elapsed time is initialized or updated.

Executive Application Programs

The requirement to process executive functions within known time limits (minor-loop) resulted in implementing executive services such as memory utilities, display and typewriter service programs, and the system parameter processor as executive application programs. They execute under control of the executive in the major loop as any other application program. The typewriter service and display service programs provide an interface between the executive and application programs and operator. Programs can request that messages be displayed or typed by calling the respective executive message routine. The messages will be stacked until the associated executive application program gains control. The typewriter and system parameter programs execute regardless of mode selection and cannot be bypassed via parameter, thus providing constant communication between software a operator.

- IBM 2250 Display Unit Service Program

  The 2250 Display Service application programs include that software required to utilize the 2250 display unit. This program's primary functions are to display system messages and to process operator inputs made via the 2250 display unit. The 2250 Display Service application program transfers and displays message text selections from a remote device combined with variable data supplied by the requesting program.

Operator inputs at the 2250 display unit cause interrupt conditions at the 2250 interface. The program detects and processes these inputs to retrieve the operator entered information, then provides the operator with an indication that the data was processed.

- Typewriter Service

To allow unique data to be typed from each computer without inducing data miscompares, all typewriter services are performed independently of the minor loop functions. Program messages and dumps are typed from each active computer, requiring the system to be degraded from a multicomputer environment to one in which a single computer is active at a time. Although the sync interrupt continues to occur, control is immediately returned to the typewriter program which processes its own I/O including prime selection, issuing commands to the I/O controller, and analyzing the I/O status words. When all computers complete their output, the typewriter routines restores the system to its original configuration prior to relinquishing control. An external switch on the typewriter unit generates an interrupt which serves as a request for typewriter input.

The typewriter service also processes the following utilities which can be entered only via the typewriter:

DUMP - Contents of core between two given addresses will be typed.

SNAP - On execution of a given address, the contents of core between two specified addresses will be typed.

DLET - The active SNAP request will be deleted.

LOAD - A given data word will be stored in a given address.

QUIT Typewriter I/O will be bypassed until receipt of the external interrupt which indicates a request for type-~~writer input.~~

- System Parameter Processor

  The System Parameter Routine processes system parameter requests entered via the typewriter and/or 2250 display to initiate the following functions:

  - Change mode of operation
  - Discontinue automatic cycling at completion of the current mode
  - Bypass execution of application programs
  - Re-activate bypassed application programs
  - Select a new prime computer
  - Status system profile (prime and active computer(s), current mode selection)

Mode parameters will remain pending until all programs have completed execution in the current mode. The parameter to discontinue automatic cycling causes the application programs which terminate to remain in that state until further mode direction is received from the operator.

Conclusion

A time-sliced multiprogrammed executive is an effective method of system control in a defined minor-major loop avionics environment. Flexibility of the table-driven program sequencing technique proved to be highly beneficial, especially in the developmental stages of the total system. The modular design of the software makes it especially suited to the support of future research and development activities related to data management systems within an avionics environment.