6

Copy No.

NASA PROGRAM APOLLO WORKING PAPER NO. 1197

HARDWARE AND SOFTWARE REQUIREMENTS FOR THE PROPULSION FLIGHT ANALYSIS COMPUTERS

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS
April 1, 1966

NASA PROGRAM APOLLO WORKING PAPER NO. 1197

HARDWARE AND SOFTWARE REQUIREMENTS FOR THE PROPULSION FLIGHT ANALYSIS COMPUTERS

PREPARED BY

Joe M. Thames, Jr.

Joe M. Thames, Jr.
AST, Propulsion Analysis Section

AUTHORIZED FOR DISTRIBUTION

Warren Gillespie, Jr.
for Maxime A. Faget
Assistant Director for Engineering and Development

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS
April 1, 1966

CONTENTS

FIGURES

HARDWARE AND SOFTWARE REQUIREMENTS FOR THE

PROPULSION FLIGHT ANALYSIS COMPUTERS

By Joe M. Thames, Jr.

INTRODUCTION

The purpose of this document is to identify the hardware and software requirements of the propulsion flight analysis computers and to discuss them in more detail than the treatment given in reference 1. This is not meant to be a detailed equipment specification. Therefore, specific reference to equipment types must be regarded as conceptual guidelines only.

Reference 1 presents an overall plan for the flight evaluation of the Apollo propulsion systems. Emphasis is placed on inflight evaluation (postfiring analysis). The means for such analysis is to be provided by a special purpose data processing system, the "Flight Analysis Computer." Figure 1 illustrates the application of such hardware in the overall mission analysis network.

The computer arrangement, presented in reference 1, is given in figure 2. It consists of a large "main-frame" computer, the "propulsion flight analysis computer," and a smaller satellite-computer, the "propulsion analysis control computer."

The functions of the computer system are:

a. To perform the essential calculations required for flight-time propulsion analysis, utilizing real-time firing data;

b. To present the results in a manner which facilitates understanding and interpretation; and

c. To perform the necessary calculations to predict ensuing mission consequences.

OPERATION OF THE FLIGHT ANALYSIS COMPUTER

The computer system will be required to operate in three different modes, each fulfilling a particular purpose:

a. Job-shop operation mode,

b. Inflight analysis mode, and

c. Analysis simulation mode.

It is important that facilities be provided which permit peripheral hardware segregation for each mode. This could be most easily accomplished using a time-sharing, multiprocessing computer system. However, realistic delivery dates for such hardware would not permit their utilization for the early manned Apollo missions. Simultaneous processing with the main frame computer is therefore excluded for the present.

Without time sharing, an interrupt-restore capability is required for continuous processing. When operating in the job-shop mode, the main frame could be interrupted by the satellite for inflight analysis or analysis simulation processing. Control would be immediately returned to the job-shop mode upon completion of the flight analysis jobs.

Job-Shop Operation

Normal data processing operation could be accomplished in an off-line manner in which jobs would be batch processed by the main frame. When interrupted, control would be temporarily transferred to the flight analysis mode (inflight analysis or analysis simulation). Upon completion of the flight analysis job, control would be restored to the job-shop program to enable it to continue execution from the point of interruption. The only loss to the original job would be turn-around time.

Inflight Analysis Operation

Immediately following the firing of the propulsion system, telemetered data would be evaluated by the flight analysis programs. The primary program, REALBEPP, is to be contained in auxiliary storage, with its associated empirical model data, ready to be loaded into the main frame for execution (see fig. 2). The flight data (reduced real-time data) necessary for the execution are received directly from an outside source (the data reduction computer). Actually, such data could be accumulated in auxiliary storage and verified by the satellite computer, prior to interruption of the main frame. When sufficient data (for a particular engine firing) have been received and verified, the main frame would be interrupted. REALBEPP would be loaded from auxiliary storage and would begin processing the flight data. The interruption would be triggered by the satellite operator through activation of a special instruction. All output of the main frame would be transmitted directly to the satellite (core-to-core) or to auxiliary storage common to both machines. All peripheral processing would subsequently be conducted by the satellite computer. Upon completion of the REALBEPP execution, the original (job-shop) program would be restored to execution automatically. The satellite would continue processing independent of the main frame.

Subsequent operation of the satellite would be concerned with the interpretation of the REALBEPP output. Certain plots and displays would be generated to aid in the interpretation. A means for quick-load-and-go programing would be available for on-the-spot programs to aid in the interpretation (for example: calculation of the mean and standard deviation of a data set).

Following output interpretation, the resulting information would be transmitted to the Flight Control personnel. This would be accomplished through various types of visual aids such as plots, text prints, et cetera.

To predict mission consequences of the evaluated firing and to extrapolate mission performance, new trajectories would be projected based on the interpreted firing results. This could be accomplished as part of the REALBEPP analysis. It is more likely, however, that this would be accomplished by another program in a different computer, namely the RTCC Real Time Trajectory Simulator. If so, an additional satellite function would involve transmission of information to the RTCC.

Analysis Simulation

In any complex flight system, much attention is usually directed toward the detection and resolution of malfunctions. Indeed, most of the flight instrumentation is strategically placed for this purpose. Often, however, malfunctioning instruments tend to add to the mystery rather than aid in its resolution.

The flight analysis method (BEPP), by sampling and comparing trajectory data and propulsion data with reference to applicable physical laws, offers a universal diagnostic capability. All that is lacking is prior knowledge of how the particular malfunctions manifest themselves in the output of the BEPP programs (malfunction signatures).

The malfunction signatures can be established by analysis simulation. Malfunctions would be simulated by the Propulsion Mission Simulator Program PREBEPP. The results of PREBEPP would be used to input the inflight analysis program REALBEPP. The results of REALBEPP would provide the required signatures.

Computer operation will be similar to the actual flight time operation of the inflight program. However, PREBEPP results will be substituted for telemetered data. Simulation would not involve the Mission Control Center in all cases. It is therefore desirable that facilities be provided for performing simulations in the Propulsion Building as well.

CONCEPTUAL HARDWARE CONFIGURATIONS

Configuration 1

The logical expansion of the figure 2 hardware configuration is given in figure 3. This concept, designated Configuration 1, involves three separate digital computers: the main frame and two satellites, a control satellite, and a job-shop satellite.

Control satellite.- The control satellite, in this configuration, is largely independent of the other computers. It contains an independent bank of auxiliary storage, together with an array of peripheral equipment. The minimum required peripheral equipment involves a card reader, a high-speed line printer, one on-line, and one off-line high-resolution plotter. Flight data would be transmitted from the data reduction source to the auxiliary storage of the control satellite for subsequent editing and verification (utilizing the satellite and its peripheral display equipment). Following substantiation of the real-time data, the main frame would be interrupted, supplied with edited data, and directed to perform the REALBEPP analysis. REALBEPP output would be transmitted directly to the control satellite for peripheral processing.

Job-shop satellite.- The job-shop satellite would be the standard type peripheral processor necessary to promote the off-line utilization of the main frame computer. The size and speed of this machine would depend upon the type of job-shop data processing it would be required to perform.

Advantages of Configuration 1.- The primary advantage of Configuration 1 is facility of software development. This would only involve modifications to existing executive programs. Main frame interruption could be accomplished easily with a minimum of overlap between the two computers. Peripheral equipment would be completely segregated. Rapid development would be possible because of system simplicity. The link between installations would only involve a telephone line.

Disadvantages of Configuration 1.- The primary disadvantage is hardware expense. Because two complete computer installations would be involved with three computers and segregated peripheral gear, hardware expense appears to be larger than that of the other configurations.

Analysis simulation in this case would necessarily involve both installations, similar to actual flight analysis.

Configuration 2

The minimum hardware outlay, Configuration 2, is illustrated in figure 4. A single multi-purpose satellite computer is involved with the main frame in this configuration. The satellite performs all of the flight analysis functions plus the job-shop peripheral processing functions. The satellite would possess two different console units. The remote unit would be located in the MCC with all of the peripheral gear necessary for flight analysis. This console would be used both for simulation and flight analysis. The adjacent console would be implemented for job-shop and imulation uses.

Advantages of Configuration 2.- Configuration 2 is attractive because hardware expense is minimized, and the most efficient hardware utilization would be realized. The equipment required in the propulsion support area of MCC would also be minimized, since computation, storage, and control modules of the satellite would be located in the Propulsion Building. Simulations would not require the MCC equipment, but could be conducted independent of MCC.

Disadvantages of Configuration 2.- Complexity of software would be the principal disadvantage of Configuration 2. The satellite's executive program would necessarily be unique, because it would be required to do several different jobs (some more-or-less simultaneously). Some time-sharing features would probably be necessary for efficiency. Software development would be much more time consuming than that of Configuration 1. Special peripheral hardware would probably be required also. Special coaxial cable would be necessary for the connection of the satellite and its remote equipment.

Configuration 3

An effective compromise of Configurations 1 and 2 is realized in Configuration 3. The remote satellite console idea is retained. However, the job-shop processing role is relegated to an independent satellite in this case. The control satellite is therefore used only for flight analysis functions.

Advantages of Configuration 3.- Similar to Configuration 1, the software development would be relatively simple. It could be accomplished by modifying existing software. Rapid development would be possible because of system simplicity. Similar to Configuration 2, the peripheral equipment required in the MCC would be minimized, and simulations would not necessarily involve MCC.

Disadvantages of Configuration 3.- Hardware cost is again the main disadvantage to this configuration. In addition, coaxial cables would be required for connecting the remote console and its associated equipment to the control satellite.

Discussion and Recommendations

Configuration 3 represents the most success-oriented and at the same time the most practical configuration. Although Configuration 2 represents slightly less hardware expense, it is believed that the software development for such a configuration could not be develcped and verified in time to be effective for flight analysis of the block I missions. The software development for Configuration 3, however, could be completed in approximately 2 months by the computer manufacturer.

Simulation could be conducted from either building without disturbing the job-shop operation (except for actual machine interruption). Job-shop operators and flight analysis operators would be in different groups.

SOFTWARE REQUIREMENTS

Interrupt-Load-Restore Feature

The job-shop operation of the flight analysis computers would be conducted in accordance with standard MSC practice. To perform flight analysis or flight simulation without disturbing job-shop operation, an interrupt-load-restore feature could be utilized. The functions of this feature are:

a. To interrupt the main frame operation,

b. To transfer the operating program, its data, and all of the machine registers to auxiliary storage,

c. To load and execute the flight analysis programs,

d. To transmit flight analysis output data to auxiliary storage or to the control satellite, and

e. To restore the job-shop program to execution (at the point of interruption) upon completion of the flight analysis execution.

The interrupt-load-restore feature would be a part of the main frame software. The special mode would be triggered by an instruction from the control satellite. All subsequent operation of the main frame would be independent of the satellite except for the possibility of output transmission.

When in the flight analysis operation mode, the software must be capable of protecting the peripheral equipment that was used in the job-shop mode. This is best accomplished by segregating the peripheral hardware and using two different executive programs (monitors). An explanation of how this might be accomplished is given below.

Figure 6 is an expansion of Configuration 3 which illustrates the use of segregated peripheral equipment. The auxiliary storage is presented as two banks of magnetic tape units, bank A for job-shop processing and bank B for flight analysis processing. Bank A would be referenced exclusively by the job-shop monitor and the job-shop satellite, while bank B would be referenced exclusively by the flight analysis monitor and the control satellite.

<u>Job-shop monitor</u>.- The main frame executive program for job-shop operation, Monitor A, would be a standard software package for the particular computer involved. Under normal operation, Monitor A would recognize the existence of only the tape units in bank A. A special direct interrupt instruction from the control satellite would cause Monitor A to dump core memory and all machine registers onto one of the bank A units (specially reserved for this purpose). Monitor A would then read in the flight analysis monitor (Monitor B) from a special bank B unit.

Flight analysis monitor.- Monitor B, upon loading, would gain control of the main frame and would proceed to dump Monitor A onto the specially reserved bank A tape. The flight analysis programs would be subsequently loaded and executed. All input/output processing would involve either the control satellite or the tape units in bank A. Upon completion of the flight analysis tasks, Monitor A would be loaded from the special bank A tape unit. Monitor A would then reload the interrupted program and return control to it at the interruption point.

Multiple day program usage.- Several mission simulator programs (thermal analysers, et cetera) in current usage require more than 24 hours running time for simulation of spacecraft missions. For maintenance reasons, computers cannot be operated continuously for such extended periods, and special provisions for piecewise execution must be provided in each program. This would not be necessary if the interrupt-load-restore feature were available, since piecewise operation could then be implemented through monitor control.

Special Language Compilers

Expanded hardware utility.- The growth of problem oriented languages (Fortran, Algol, et cetera) has greatly expanded computer utility for scientific data processing. Such source languages are designed to evaluate explicit algebraic expressions in a systematic manner, and are therefore quite useful in the numerical solution of engineering problems. However, most of these languages are sorely inadequate for input/output and list processing. Because of this, most large scientific programs must be supplemented by subroutines that are written in machine oriented language (assembly language). This is especially true when the program utilizes experimental test data which must be edited and filtered (and sometimes recalibrated).

For flight analysis computing, a source (problem-oriented) language is desired which is suitable for both test data processing and scientific computing. Source languages such as Fortran and Algol can be sufficiently extended if mixing of machine assembly code is allowed in each program. This characteristic could be easily included in most currently used compilers. In addition to, or in place of, imbedded machine code, the capability for generating and calling macro routines from the source language is a highly desirable attribute. Such an attribute would provide the source-language programer with complete machine utilization capability.

Formula manipulation capability.- One of the newest and most promising developments in problem-oriented language capability is formula manipulation. Heretofore, scientific data processing has been limited to the use of numerical methods. Many engineers have either never learned, or have had to ignore, analytic and theoretical approaches to problems because they have been forced to utilize numerical techniques exclusively.

The ability of a digital computer to manipulate mathematical formulas serves to reactivate all of the powerful classical methods of mathematical physics. In existence now are languages which facilitate formula simplification, expansion, and factoring, as well as analytical differentiation and integration. Programs written in such languages are capable of power series and function series (Fourier, Bessel, et cetera) approximation and the implementation of analytical transformations (LaPlace, Fourier).

FORMAC (ref. 2), IBM's contribution, is an extension of FORTRAN which provides it with the ability to perform a certain amount of symbolic computation. Like FORTRAN, it is somewhat limited in scope and flexibility.

A more general and flexible language, Formula Algol (ref. 3), has been developed at Carnegie Institute of Technology. This language is an extension of Algol 60, which facilitates list processing and pattern recognition, as well as formula manipulation.

A language of similar capability, MATHLAB, is being developed at MIT and MITRE corporation (ref. 4). MATHLAB is designed for use in an integrated man/machine arrangement with the aim of providing instantaneous formulation/solution capability to the scientist.

In the propulsion flight analysis effort, extensive use of the formula-manipulation capability is envisioned. The ability to modify formulas at execution time relieves the propulsion-system model designer, who is constantly harassed by hardware design changes. Many existing simulation programs become obsolete because modifications cannot keep pace with changes in propulsion hardware. Analytical formulation of the partial derivatives required in the BEPP analysis comprises a large part of the program development work. The ability to change the partial derivatives without program modification provides complete flexibility in the selection of primary and secondary parameters for the BEPP process. With formula manipulation, propulsion simulation models could be utilized as input data rather than as built-in portions of the BEPP programs.

With complex hardware installations, such as the one proposed, programing flexibility is mandatory for efficient machine utilization. Such flexibility is adequately provided by the use of language compilers which possess the foregoing characteristics.

REFERENCES

1. Thames, Joe M., Jr.: Flight Analysis of the Apollo Propulsion Systems. NASA Apollo Working Paper (to be published).

2. Sammet, J. E.; and Bond, E. R.: Introduction to FORMAC. IEEE Transactions on Electronic Computers, Vol. EC-13, August 1964, pp. 386-394.

3. Perlis, A. J.; Iturriaga, R.; and Standish, T.: A Preliminary Sketch of Formula Algol. Presented at the University of Michigan Summer Conference Course on Automatic Programing, June 1965.

4. Engelman, C.: MATHLAB: A Program for On-Line Machine Assistance in Symbolic Computations. Proceedings - Fall Joint Computer Conference, 1965.

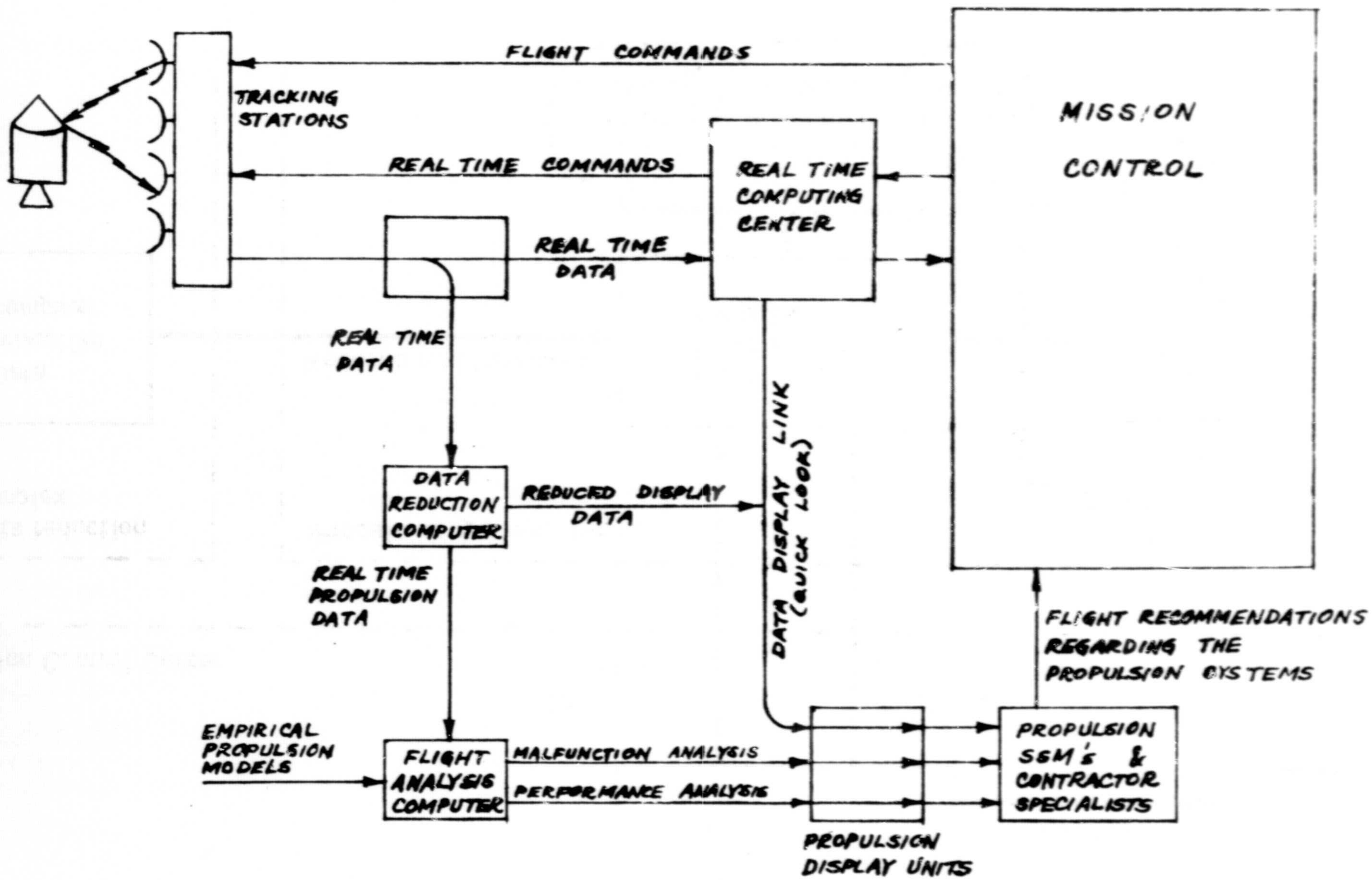

Figure 1.- Communication network for inflight propulsion analysis.

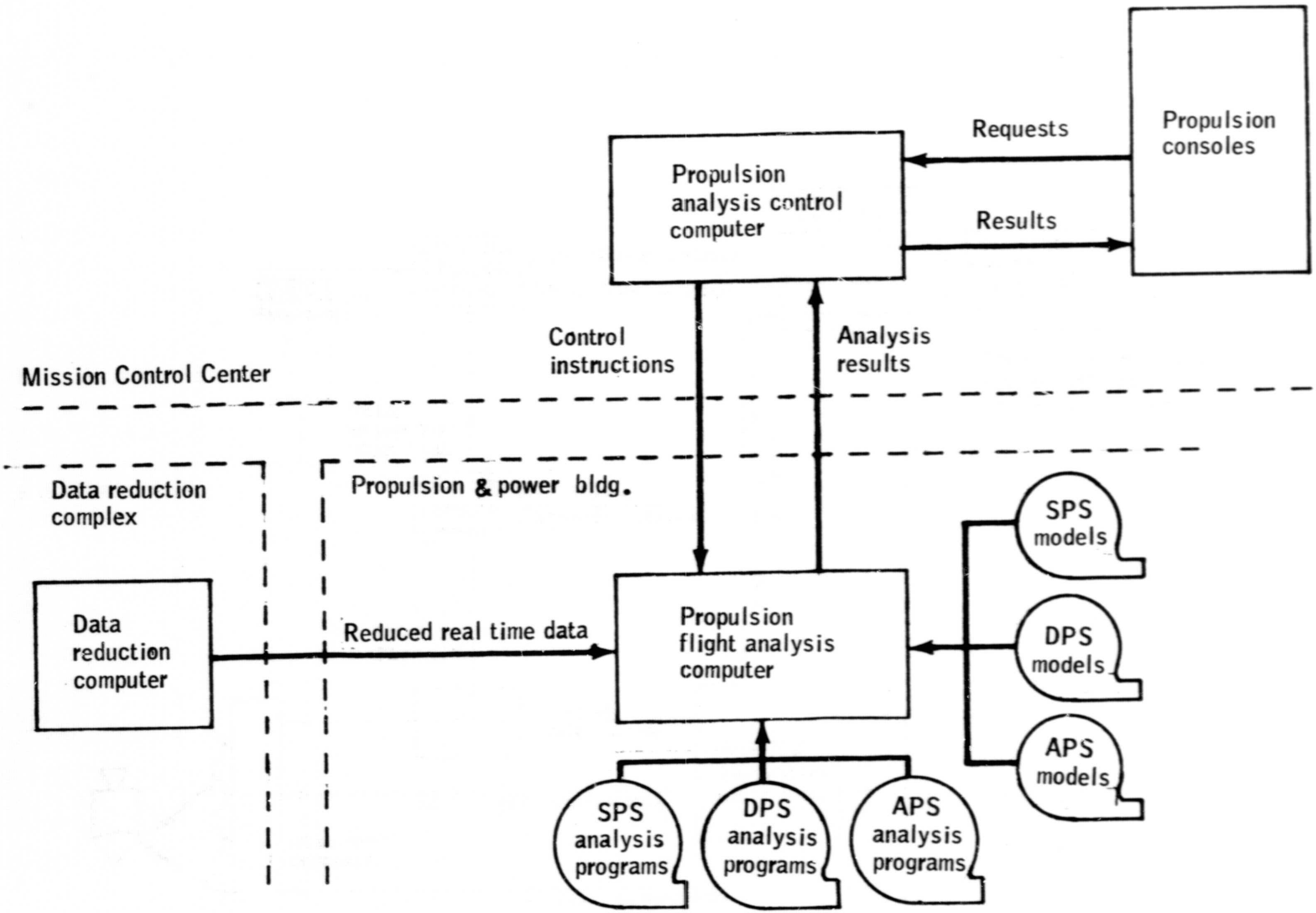

Figure 2.- Special purpose computer configuration.

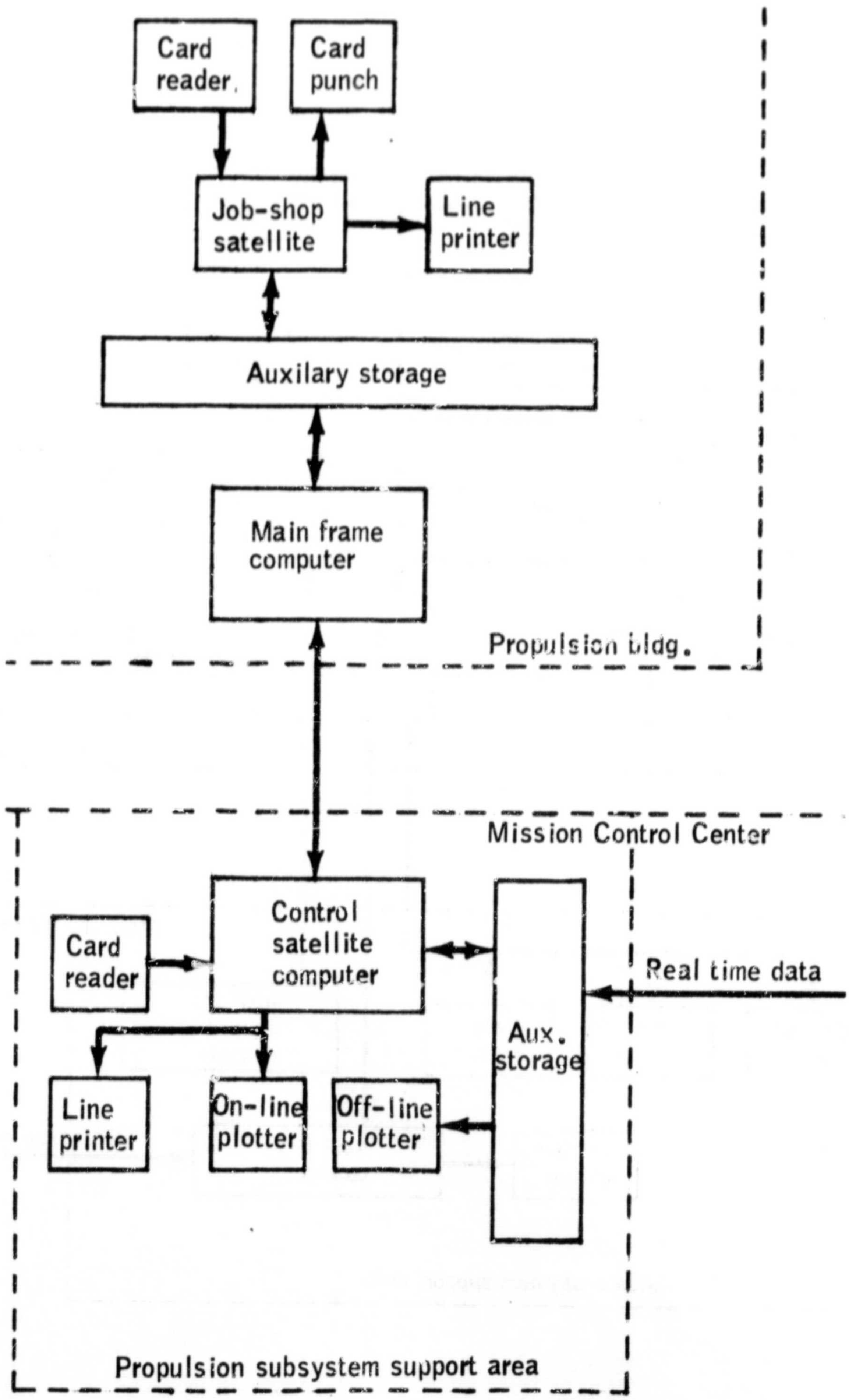

Figure 3.- Configuration 1.

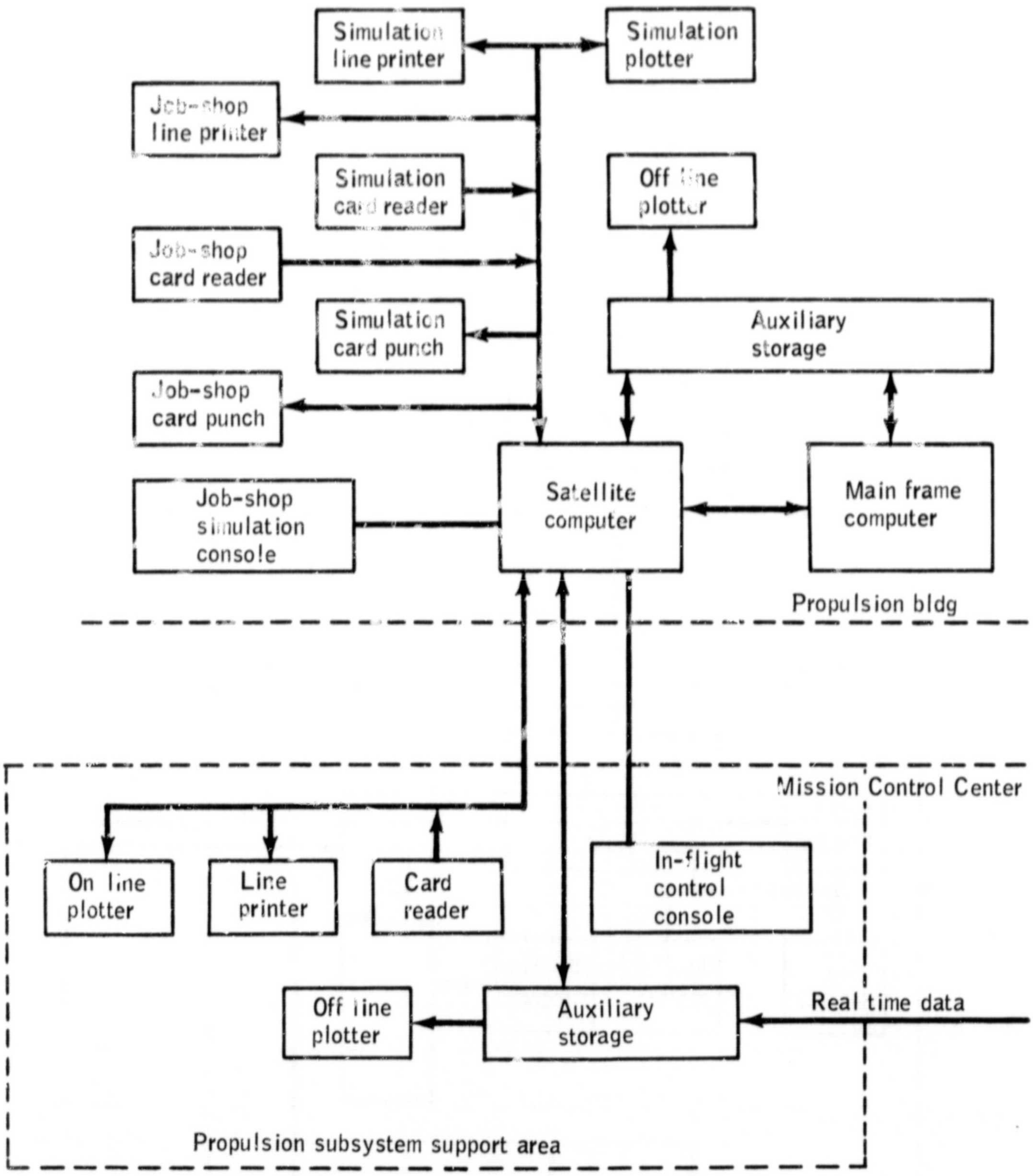

Figure 4.- Configuration 2.

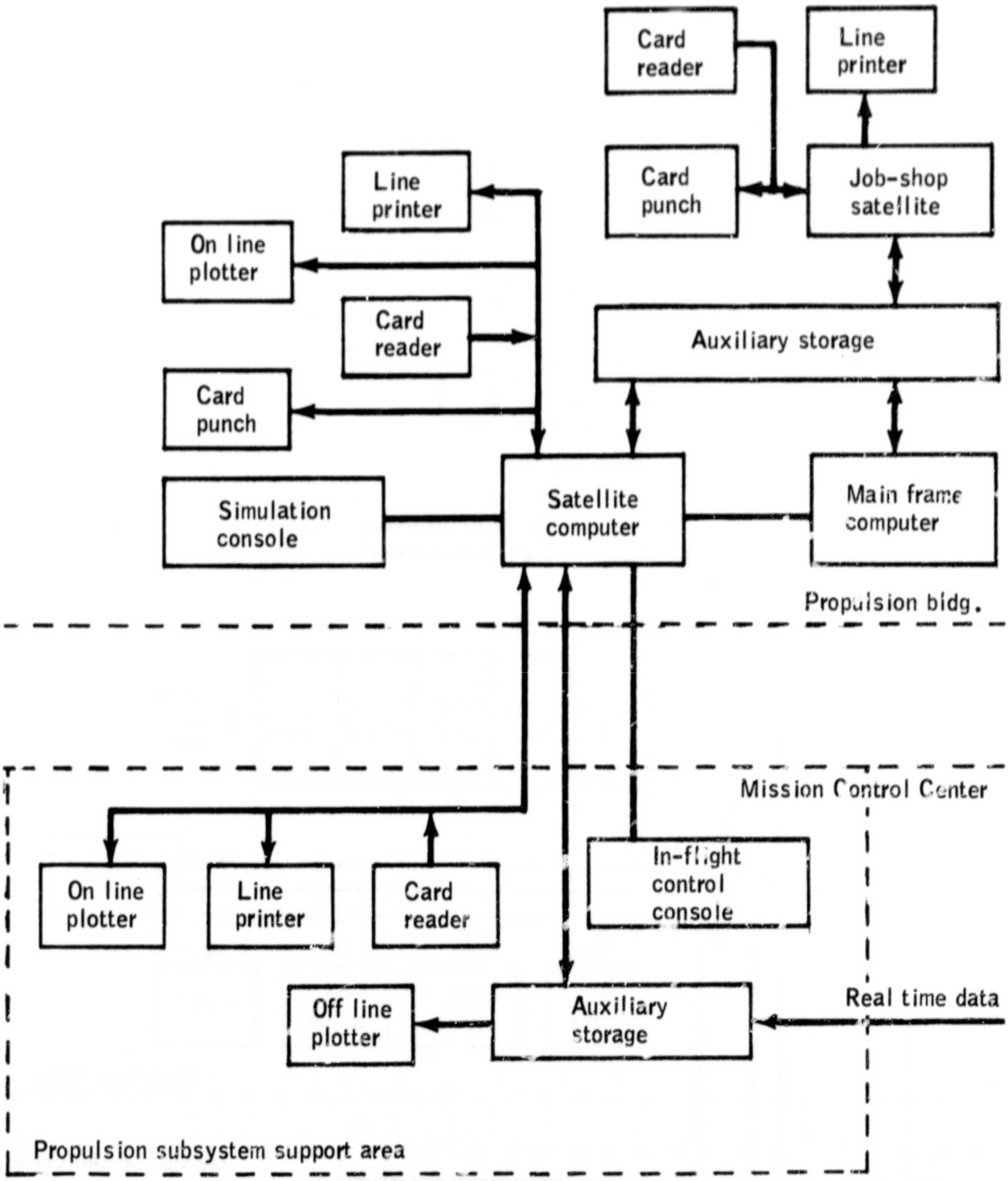

Figure 5.- Configuration 3.

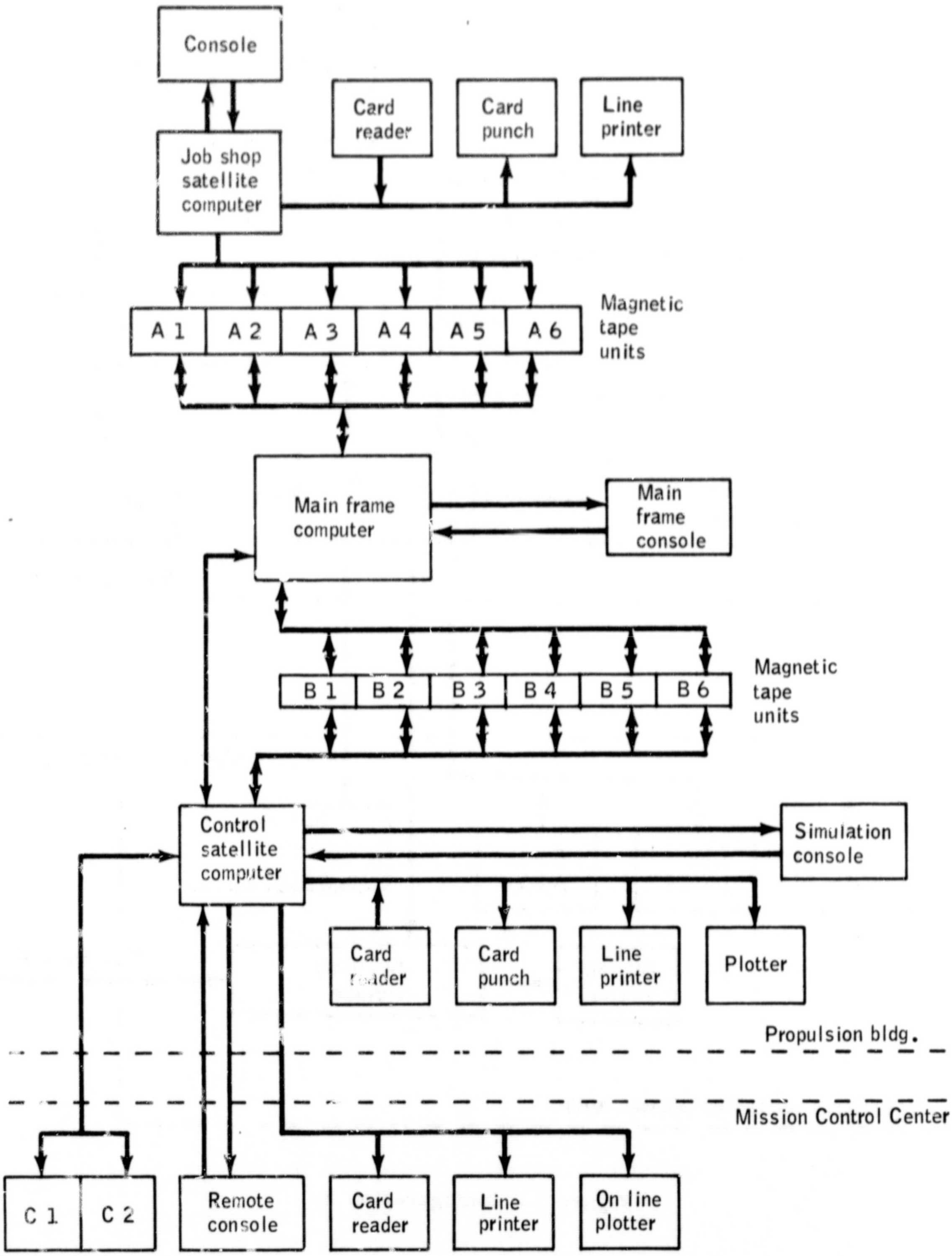

Figure 6.- Configuration 3 with hardware segregation.