NASA CONTRACTOR REPORT

NASA CR-1158

STUDY OF SPACEBORNE MULTIPROCESSING — PHASE II

Volume I — Summary

by L. J. Koczela

Prepared by
NORTH AMERICAN ROCKWELL CORPORATION
Anaheim, Calif.
for Electronics Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1968

NASA CR-1158

STUDY OF SPACEBORNE MULTIPROCESSING — PHASE II

Volume I — Summary

By L. J. Koczela

Issued by Originator as Report No. C6-1476.22/33

Prepared under Contract No. NAS 12-108 by
NORTH AMERICAN ROCKWELL CORPORATION
Anaheim, Calif.

for Electronics Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 — CFSTI price $3.00

FOREWORD

This final report describes the results of the Phase II portion of a study conducted under NASA contract NAS 12-108, "Spaceborne Multiprocessor Study". It was performed by Autonetics, a division of the Aerospace and Systems Group of the North American Rockwell Corporation. The work was administered under the direction of the National Aeronautics and Space Administration, Electronics Research Center, Computer Research Laboratories, Cambridge, Massachusetts. The NASA project manager was Mr. G. Y. Wang. Volume I contains a summary of the study and volume II a detailed technical discussion.

CONTENTS

ILLUSTRATIONS

TABLES

SUMMARY

This final report presents the results of a research study of an advanced multiprocessor computer organization for future space missions. The organization is entitled the "Distributed Array Memory and Processor" computer. A manned Mars lander mission was selected as a representative mission for application and the computer requirements were developed. The organization developed utilizes an array of cells capable of a high degree of computational parallelism. The feasibility of such cells is dependent upon future LSI technology. Research was carried out in the hardware design and software analysis of this organization. This volume contains a summary of the results of the study.

1. INTRODUCTION

The purpose of this study was to investigate an advanced multiprocessor computer organizational concept. This organization is called the "Distributed Array Memory and Processor" and will be referred to as the "distributed processor." The distributed processor evolved from the Phase 1 portion of this study (Reference 1); the purpose of the Phase 1 portion was to evaluate various multiprocessor organizations for future space missions.

This volume contains a summary of the accomplishments during the study; detailed technical considerations may be found in volume II. Prior to developing the organization the computer requirements for a manned Mars mission were defined, LSI semiconductor technology was projected ten years out, and parallelism within the spaceborne computations was investigated.

Figure 1-1 contains a block diagram of the organizational concept. The organization is seen to consist of a number of identical cells interconnected in a particular manner. Each cell consists of a general purpose processor section and a small amount of memory (512 16-bit words) on a single LSI wafer. The cells are divided into groups (4 groups of 20 cells each are considered for the spaceborne application) and these groups are connected by an intergroup bus for communication. Within each group the cells communicate with each other by an intercell bus and by neighbor communication lines. Each group will have one cell designated as a controller cell; the remaining cells can be operated independently or dependently of the controller cell. This organization is thereby capable of simultaneously taking advantage of applied (global control) parallelism and natural (local control) parallelism within computations.

The organization offers considerable advantages in application to future manned space missions: (a) efficiently meeting the widely varying computational loads of different phases of a mission, (b) efficiently mechanizing the diverse requirements of various subsystems of a mission such as a command module and a lander module, (c) an overall net reduction in power due to the ability to turn modules on and off (d) increase in reliability, given that failure rates of dormant equipment are lower

than operating equipment, and (e) enhancement of probability of mission success and availability, due to reconfiguration around failures at a low module level. The latter advantage is most prominent for this organization since it offers many levels of graceful degradation by reconfiguration around individual cell failures.

The organization was subject to a detailed analysis, in particular, the following topics were considered: (a) the development of this distributed organization, (b) the development of the architecture of the organization, (c) input/output methods, (d) failure detection and reconfiguration, and (e) software and executive programming methods.

As a result of this analysis, a preliminary design of the cell was conducted. In addition, a detailed design of the communication system (the inter-cell bus) was accomplished. A reliability simulation using Monte Carlo techniques was also conducted for the manned Mars lander mission; statistics for probability of success and availability were thereby generated.

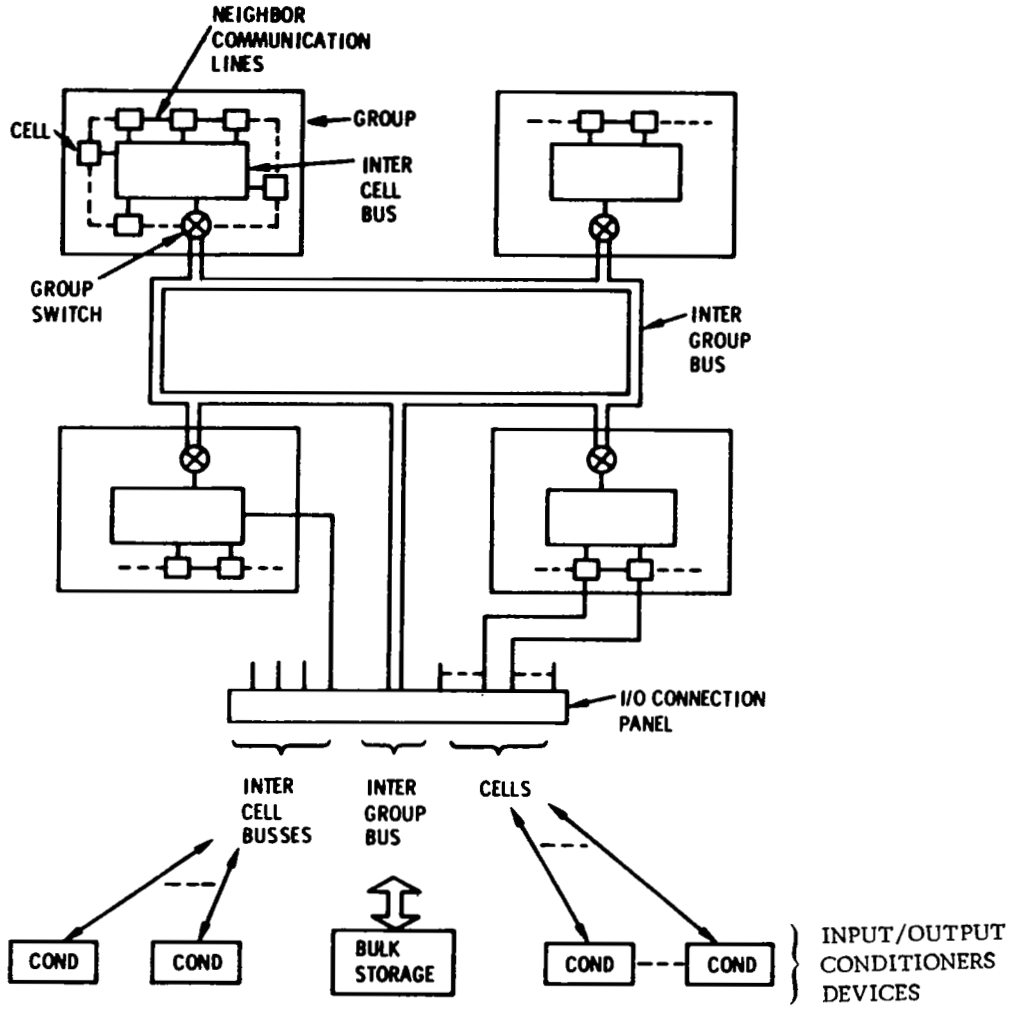


Figure 1-1. Distributed Processor Organization

2. REQUIREMENTS

Manned Space missions in the 1980 time period were selected as the application area for the distributed processor computer organization. A firm base for requirements was established by a detailed investigation of one particular mission; the manned Mars lander mission. This mission spans a broad spectrum of requirements (long duration, widely varying and diverse computational loads, high reliability demands, etc.) so that using it as a base for requirements results in applicability to many other missions in the same time period, such as extended earth orbital space stations.

An analysis of the manned Mars Lander Mission was performed to define the computational functions from which the computer requirements may be defined. The mission profile covers a 420-day period and consists of three primary phases: Trans Mars (120 days), Mars Orbital (40 days), and Trans Earth (260 days). These three primary phases may be subdivided into twenty basic mission phases which begin with earth atmospheric ascent and culminate with earth re-entry; a listing of these twenty phases is given on the horizontal axis shown in Figure 2-1.

The computational functions were defined for each of the mission phases and the computer requirements (speed and storage) were thereby derived as a function of mission phase. Much of the requirements work was accomplished in the phase 1 study and is reported in Reference 1. However, the computational functions associated with the scientific experiments were subject to a further detailed investigation and this is contained in volume 2.

Figure 2-1 and 2-2 contain the computational storage and speed requirements as a function of mission phase. These requirements are those of the individual mission phases only. The requirements were derived assuming a conventional sequential computer organization with indexing, multiple accumulators, a basic instruction repetoire and a 16 bit word length.

There are two other requirements which may be defined for the computer. The first is a reliability requirement; a probability of success of 0.997 for the mission is required. The second is a reconfiguration time requirement of 0.1 second during critical phases such as atmospheric entry. Reconfiguration is defined as having a computational function mechanized and performed correctly in the computer system after some failure in the computer system. During these critical phases there actually is a subset of the computations being performed that may be considered critical (e.g., a portion of the navigation and guidance programs). These are the programs that must be continued with only a 0.1 second interruption.

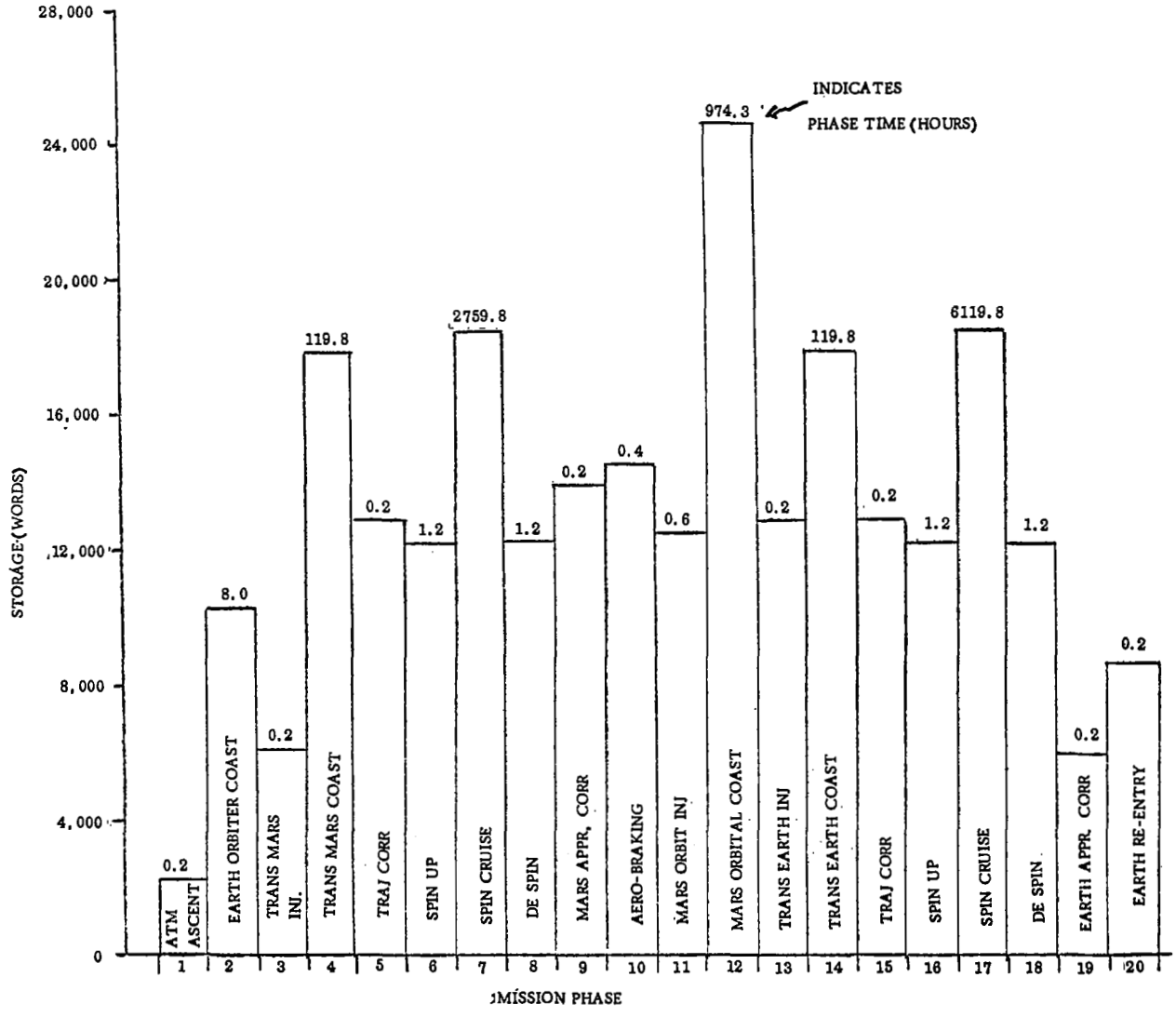


Figure 2-1. Computer Storage Requirements

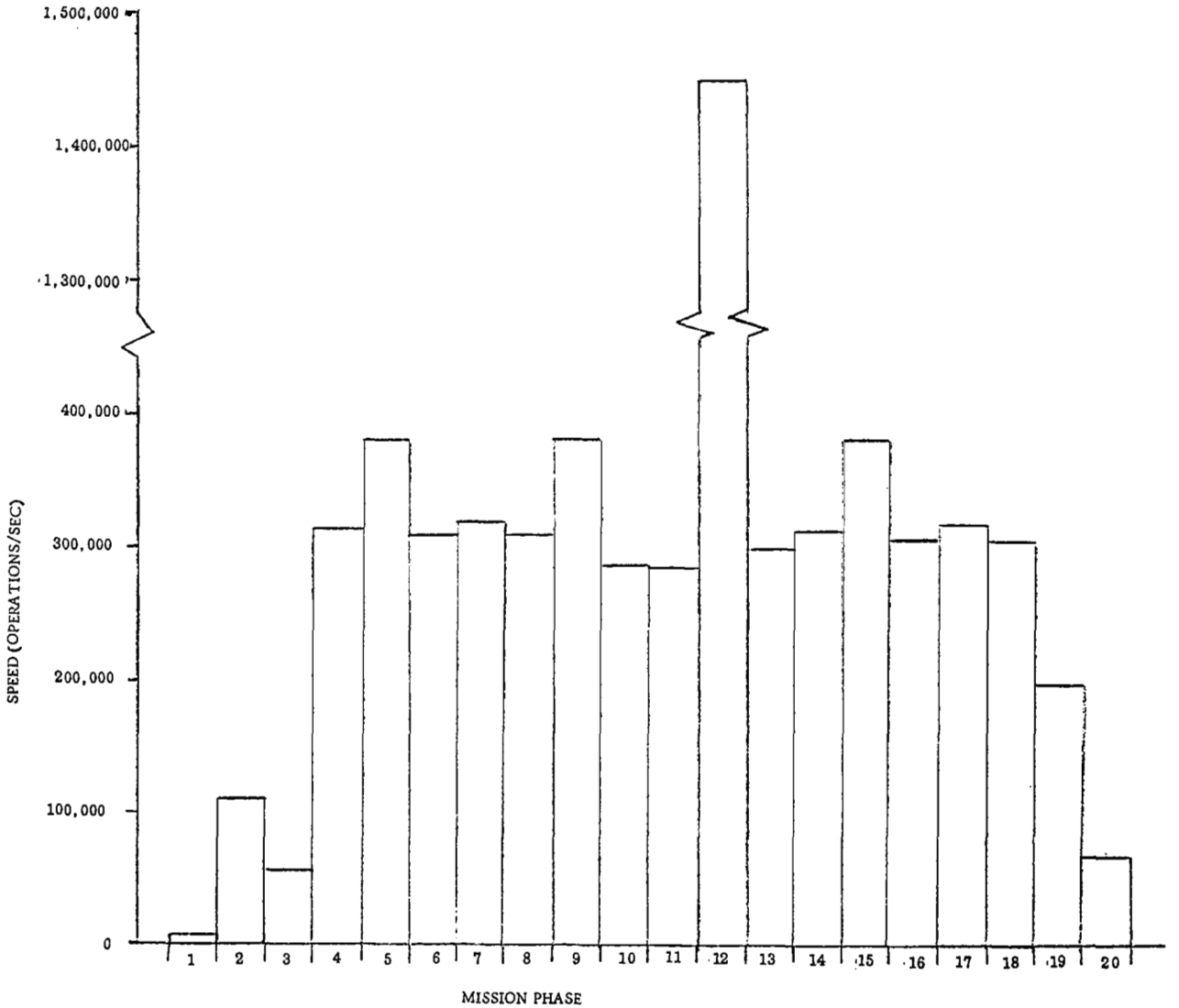


Figure 2-2. Computer Speed Requirements

3. TECHNOLOGY

There are basically two device oriented technologies in existence today that lead the developments in LSI. These are: (1) bipolar and (2) metal-oxide-semiconductor-field effect transistors (MOS-FET). Bipolar devices offer a speed advantage over MOS-FET's while MOS-FET devices offer the advantage of: a simpler process for manufacturing, less area for the same function, and lower power dissipation compared to bipolar.

The distributed processor organization is based upon future developments in MOS technology that will make practical the development of large wafers containing 10^5 or more devices (MOS-FET's). A "cell" in the organization may then be manufactured from a single wafer. Based upon todays state of the art in the two device technologies, a calculation of the required area and power dissipation for a wafer containing 10^5 devices was made (see volume II for details). The bipolar wafer would require 7.0 in^2 and dissipate 250 watts while the MOS wafer would require 1.6 in^2 and dissipate 6.5 watts. At this level of complexity power dissipation is the primary limiting factor that prevents the construction of wafers containing 10^5 devices using todays state of the art. It is also clear that at this level of complexity the advantages of MOS-FET's over bipolar transistors are impressive.

The technology time frame for the distributed processor is approximately ten years from today (late 70's). To extrapolate MOS technology this far in the future is quite difficult since in general semiconductor technology has undergone such drastic changes in the last 6 to 8 years. One particular batch fabrication technology presently being developed is the: heteroepitaxy of thin semiconductor films in which active devices are fabricated; a particular example is the growth (heteroepitaxy) of single crystal silicon-on-sapphire (SOS) by chemical vapor deposition. Application of this technology to MOS (MOS-SOS) results in two basic improvements: an overall power reduction and a speed capability increase.

There are recognized problems in LSI and research and development programs are in progress to solve these problems. This work can be divided into three categories:

1. Computer Aided Design

2. Materials, devices, and process improvements

3. Testing and customizing

Research and development work is currently under way on MOS-SOS devices. Based upon present results this technology was extrapolated ten years out to estimate the size and power dissipation of a wafer containing 10^5 devices. It was estimated that such a wafer would be approximately 1.5 inches in diameter and dissipate 0.125 watts (static and dynamic power). The extrapolations assumed that a maximum device density of 245,000 devices/in^2 would be used in parts of the wafer. This represents an increase of approximately (10x) over present results. Such an improvement should be readily achieved in ten years.

4. PARALLELISM

The distributed processor organization exhibits the capability of a high degree of computational parallelism. For this reason parallelism within computations was investigated and applied to the manned Mars mission computational functions to determine how effectively parallelism within the computations may be utilized.

Two types of parallelism, applied and natural, may be defined as follows. Applied parallelism: The property of a set of computations that enables a number of groups of identical operations within the set to be processed simultaneously on distinct or the same data bases. Natural parallelism: The property of a set of computations that enables a number of groups of operations within the set to be processed simultaneously and independently on distinct or the same data bases. It may be noted from these definitions that applied parallelism is a special case of natural parallelism, since the naturally parallel operations could be groups of identical operations. The distinction between the two types is introduced since it has important implications with regards to computer organization.

An important part of any parallelism analysis is the formation of computation graphs to study the degree of parallelism utilized and the resultant reductions in computation time over a single sequential computational solution. This analysis leads to a curve of computation reduction ratio vs. degree of parallelism or an "effectiveness of parallelism curve". It was shown in volume II that the procedures used in deriving such a curve are similar to those encountered in job shop scheduling or assembly line balancing theory from the field of operations research. Considerable research has been carried out in this field of operations research. The general problems to be solved are: (1) given m jobs or tasks to be performed by n men or assembly stations, determine the sequence or assignment of the jobs or tasks that completes them in the shortest time and (2) given m jobs to be performed, the total time spent by any one man or at any one assembly station is not to exceed T, what is the sequence or ordering of the jobs to minimize n the number of men or assembly stations required.

Evaluation of research carried out on these problems showed that no general solutions exist to the optimum sequencing problem that are applicable to the study of parallelism here. The only solutions to the problem are generally very lengthy computational programs considering all permutations of the solution space to find a minimum; analytical procedures are non-existent for general solutions.

The general procedures to be followed in applying parallelism studies to the computational problem are outlined in volume II. In general one must investigate the computations for applied parallelism and then for natural parallelism.

The general problem is to consider a certain degree of parallelism as being available and then assigning and sequencing the computations so as to achieve a minimum computation time. This procedure involves many permutations of individual computations to attempt to achieve an optimum solution. As the degree of parallelism is varied one may then obtain an "effectiveness of parallelism" curve.

The computational tasks for the manned Mars mission were investigated for parallelism. Curves of the "effectiveness of parallelism" were obtained for both natural and applied parallelism. These curves are given in volume II.

5. DEVELOPMENT OF THE DISTRIBUTED PROCESSOR ORGANIZATION

5.1 INTRODUCTION AND GENERAL DESCRIPTION

The distributed processor organization was developed to provide general purpose computing with very high reliability for long duration space missions. In addition it was desirable for the system to make good use for future circuit and memory technology, to dissipate a small amount of power, and to be easily expandable or contractable so that it could be flexibly applied to a variety of missions. The high reliability requirement can actually be interpreted as a need to continue executing a critical subset of the computations in spite of hardware failures. In other words the computations can be "gracefully degraded." The distributed processor meets this goal by dividing its memory up into a number of modules (approximately 512-16 bit words used per module) and then integrating each memory module with a processor to make an independently operating "cell." These cells can then individually be turned on and off to match the requirements from phase to phase. This will substantially decrease the power dissipation of the system and also increase the reliability (assuming off line or dormant reliability is higher than on line or operating reliability). The distributed processor structure consisting of many cells can meet the expandability requirement by adding a small or large number of cells to match a new set of requirements. The system also makes good use of future circuit and memory technology since each cell can be fabricated on a single LSI (MOS) wafer in the 1980 time frame.

The organization of the distributed processor was shown in Figure 1-1. It is seen that the distributed processor structure consists of cells. These cells are interconnected to form a group and a number of groups are interconnected to form the organization. A cell may be thought of as a small conventional computer that has a relatively small memory. Parallel operation may exist between cells in a group and between the groups in the system. Therefore, the organization appears as a highly parallel computational facility.

The computational tasks that are presently required to be carried out are divided or assigned to various groups and finally to various cells in the groups. A group of cells may carry out a complete task (such as the navigation and guidance computations), a number of small tasks or even part of a large function or program. The tasks assigned to a group are sub-divided or assigned to individual cells within the group for actual computational mechanization. An example of a sub-task that may be assigned to an individual cell is the computation of position and velocity from navigation sensor input data; this sub-task may be part of a larger navigational task assigned to the group. Some of the more important considerations in assigning tasks are minimizing the communication between cells and groups and the utilization of applied and natural parallelism where possible.

The cells in a group and in the entire system are all identical in terms of hardware. However, at any particular time some cells will be functionally operated differently from other cells. Each group will have one cell designated as a controller cell. The remaining cells in the group will be designated as working cells. The cell that is the controller cell will be responsible for controlling the intercell communication bus and providing the executive control of the groups' operation. All of the cells in a group are connected to the inter-cell communication bus. This bus will be a 1/2 word

parallel and serves as the primary means of communication among the cells in a group. The inter-cell bus will be used for communicating data between cells and for the transmission of global instructions and commands to cells. The cell that has the unique function of the controller cell will control the use of the inter-cell bus and will provide all the global instructions and commands.

The working cells in a group can either accept and execute the global instructions that are sent on the inter-cell bus or they may fetch and execute instructions from their own memory. In fact, some cells may be using the global instructions from the inter-cell bus, while others are using instructions from their own individual memory. Since all of the cells are physically identical, this operation will be described by stating that the cells are capable of existing in different states functionally. In this way, both local and global control may be carried out simultaneously within a group. As a result both the natural and applied parallelism inherent in a group's tasks may be efficiently carried out.

As was noted above, the controller cell will control the inter-cell bus. The functions to be carried out on the inter-cell bus include communication of data between cells, the sending of global instructions and commands, and the sending of commands to individual cells. As mentioned above, only cells operating under global control will be using global instructions and commands from the inter-cell bus. However, all of the cells in the group will be responsive to commands sent to individual cells via the inter-cell bus regardless of whether the cell is operating under local or global control (these commands are addressed to one particular individual cell). It may therefore be seen that the controller cell can command one or more cells operating under global control simultaneously while cells operating under local control must be commanded or talked to on an individual basis.

Global control implies that one or more cells are using this type of control. The controller cell provides all control information sent on the inter-cell bus. It may therefore be seen that cells that are being operated in a global control mode are highly dependent on the controller cell in that they are receiving instructions from it and are also responsive to its global commands. However, the cell operated in a local control mode is considerably more independent of the controller cell in that it fetches its instructions from its own memory and must be talked to or commanded individually. It can be seen that the controller cell has the overall control of the cells in a group. The degree of this control can be varied depending on the functional use of each cell. The above discussion is intended to serve only as a brief introduction to the operation of a group of cells; further details will be given in the next section and a considerably more detailed explanation is given in Volume II.

A simplified block diagram of a cell is shown in Figure 5-1. The cell consists of a memory and arithmetic and control section in a similar manner to conventional computers. In addition, the cell will contain logic for inter-cell bus communications and for its own identification; this is the part of the cell that differs significantly from conventional computers.

All storage within a cell is directly addressable and is divided into control registers and storage registers. The storage registers hold the programs and data to be used by this cell and possibly other cells. The control registers are the general processor registers used in the execution of instructions. Instructions, whether they

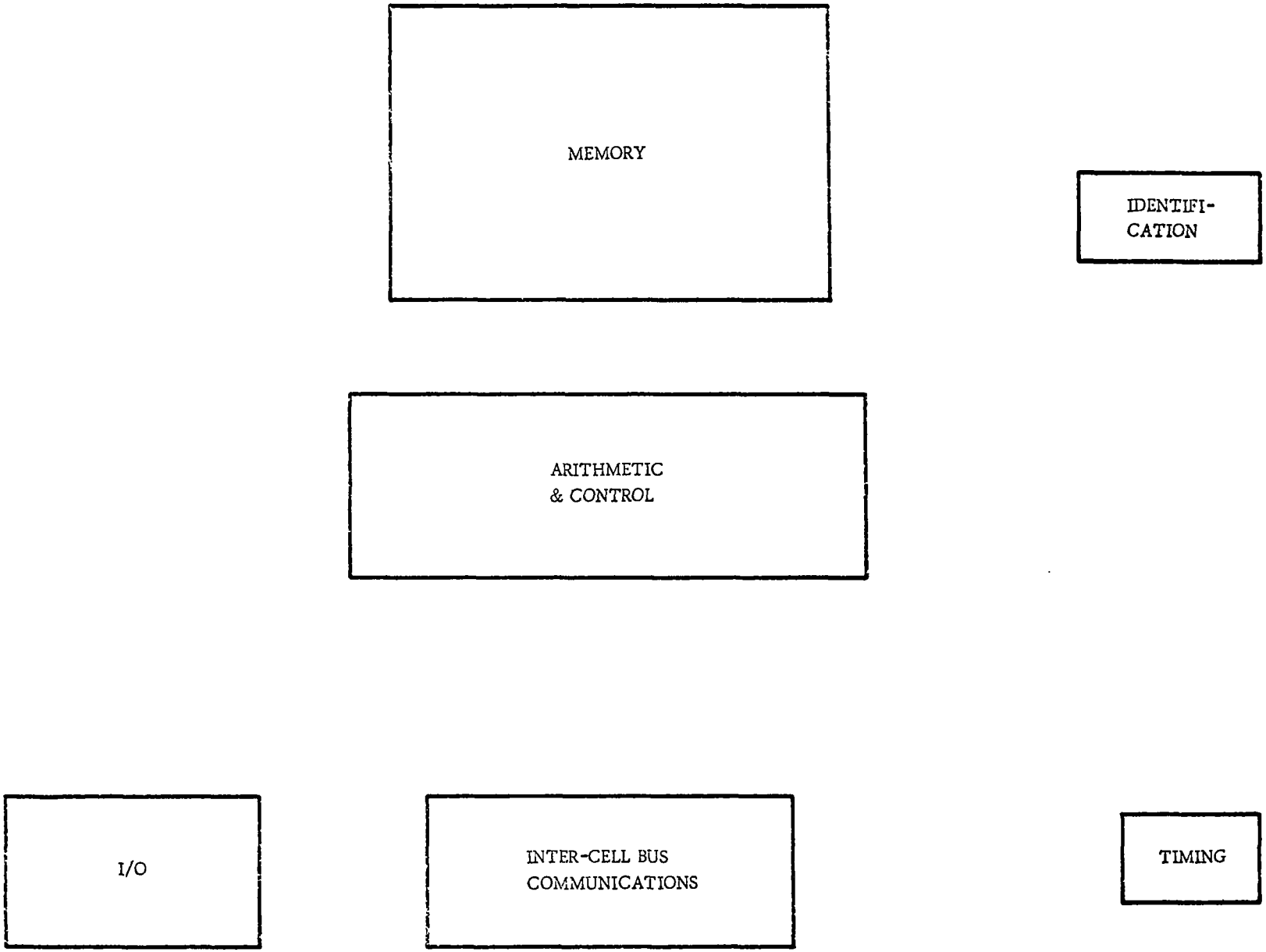


Figure 5-1. General Cell Block Diagram

come from the cells own memory or from the inter-cell bus, are decoded and executed in a manner similar to conventional computers.

In addition to the inter-cell bus communications path, another means of communication is provided for, which is the neighbor communications. Each cell contains a single serial connection to each of its four adjacent neighbors (north, east, south, and west) with wrap-around connections at the edges of the array of cells. This communication means is primarily included to facilitate the global programming of a group where small amounts of data are required to be communicated between cells, e.g., a matrix multiply utilizing applied parallelism.

The operation of a group was briefly outlined above. It was seen that the cells were capable of operating under either local control or global control. The organization consists of a number of groups interconnected by an inter-group communication bus as shown in Figure 1-1. Each of the groups can operate independently and simultaneously of the other groups. Therefore, this introduces another level of parallelism in the organization in that it is possible to simultaneously have more than one group in operation where each group may be utilizing local and global control to carry out its tasks. Each group is connected to the inter-group bus by a group switch as shown in Figure 1-1. The group switch is controlled by the controller cell of the group. One of the groups will contain additional functions in its controller cell that enables it to operate as the system executive group. This will primarily involve coordinating and controlling the communications that take place on the inter-group bus. Since all of the cells are identical, the system executive can be located in any one or more of the groups.

There are many factors that determine the number of cells in a group and the number of groups to be used. From the requirements given in Section 2 and the parallelism studies of Section 4, (see Volume II for details) it has been determined to use 4 groups of 20 cells each (512 words/cell). It should be noted that the system could be designed using a variable number of cells per each group; however, there is some impetus toward using fixed size groups. Such a structure is more symmetrical and as a result, more capable of many levels of graceful degradation in small increments.

Finally, it should be noted that input-output connections are provided directly to each cell and will be connected to the appropriate sensors and conditioners (some may not be used if the cell gets I/O over the bus). In addition, I/O devices may also be connected directly to the communication busses in the distributed processor system.

This section has served to introduce the general concepts of the organization and its general operation. The remaining sections of this report will present detailed descriptions of features and operation of the system. The next section on Group Architecture will clarify the detailed operation of the group and many points presented here.

5.2 GROUP ARCHITECTURE

Architecture means the combining of software and hardware features to make a balanced useful system that will meet the requirements set upon the computing system. The distributed processor system consists of groups, which are made up of cells. Because the cells in a group are connected by neighbor communication lines, and the controller cell can send global instructions to cells, the group is the fundamental unit of the computing system. This section describing the group architecture will describe the features desirable to unify the cells into a working group. All of the topics covered in this section are expanded upon in much greater detail in Volume II.

5.2.1 Cell States

Although all the cells are identical in hardware, a cell always exists functionally in one of seven different and mutually exclusive states. These states are listed in Table 5-1.

Table 5-1. Cell States

1. Permanently failed - power off
2. Shut down - power saving state
3. Independent
4. Dependent under global control (Global State)
5. Dependent under local control
6. Dependent in wait state
7. Controller cell

Independent cells are functionally similar to a conventional computer. These cells fetch all instructions and operands from their memories. The cell that is in the independent state stays in this state until the controller cell sends a command on the inter-cell bus to change states. The independent cells can process problems that are not amenable to global processing.

Dependent cells respond to global instructions and global level commands sent out from the controller cell. A dependent cell exists in one of the states 4, 5, or 6. Which of the three depends upon the level of instructions being sent from the controller cell and the cell's level register contents. (The concept of levels is described below under cell identification.)

A dependent cell in the global state (also called the active state) is receiving instructions from the controller cell via the inter-cell bus.

As mentioned above the instructions being sent from the controller cell to dependent cells are identified as being at a certain level. A dependent cell that is not at the proper level to receive global instructions can idle and not execute instructions. This is the wait state. Therefore, if the controller cell is servicing certain dependent cells, other dependent cells may wait their turn for service.

A dependent cell, instead of waiting for the controller cell to send the instructions for its level, may fetch and execute instructions from its own memory. This is the local control state of a dependent cell.

The seventh state of a cell is the controller state. In this state, a cell controls the inter-cell bus and may issue global instructions. The fundamental ground rule of making all the cells of the same hardware allows any cell to become a controller cell. This gives the advantage that the controller cell functions may be switched among several cells. Thus there is no requirement that all the executive and controller programs fit in one cell. At any time, there is only one cell in the controller cell state in a group.

5.2.2 Cell Identification

The distributed processor computer system has two methods for identifying the individual cells. One method identifies the cells by a level. A cell will be identified as being at one of eight levels. The other method results in each cell being given an identifier, also known as the cell address. Thus a cell has two "names", a common first name (level) and a unique last name (identifier). This concept of having two names is important when discussing the dependent and independent cells.

Independent cells use only one name, their identifier or cell address. The level (or first name) is not used and, although present in a level register, has no meaning unless the cell later assumes a dependent state.

Dependent cells use both names. The controller cell can send out information using a first name (level number) to all the dependent cells; all the dependent cells at this level will respond. If a last name (cell address) is sent, only the cell with this name will respond since each cell has a unique last name regardless of state.

Thus, it can be seen that the controller cell must communicate with independent cells individually. However, the controller cell may communicate with dependent cells individually or to more than one at the same time since the dependent cells respond to levels and more than one cell may have the same level.

5.2.3 Source of Instructions

Unlike a traditional computer that has instructions stored in its memory which is always available to the processor, the distributed processor has different sources of instructions dependent upon the state of a cell. The independent cell receives all of its instructions from its own memory like a traditional computer. The program counter is used to control the fetch of instructions.

The dependent cell in the global state gets its instructions from the controller cell. These cells receive the instructions from the inter-cell bus and then execute them. The controller cell precedes the instructions with a name. The name (level number) is contained in a control word sent on the inter-cell bus. This control word is a prefix to a group of instructions and is the level of all instructions until a new level prefix is sent or another control instruction is sent.

Every dependent cell compares the level prefix sent by the controller cell to the level register contents contained in the cell. If the prefix and the level register contents are different, the cell ignores all the instructions, data, etc. sent by the controller cell, until a new level prefix (or other control word) is put on the bus by the controller cell.

The dependent cell not receiving instructions from the inter-cell bus may fetch instructions from its own memory. This cell is in the dependent local control state. The fetch of instructions is identical to an independent cell. The difference in this cell being that it is constantly examining the bus for a command at its level as noted above.

The controller cell always fetches instructions from its own memory. The instructions destined to be executed by the dependent global cells are not executed by the controller cell. All other instructions are executed by the controller cell and are not sent to the global cells. This will be explained in more detail below under instruction execution.

5.2.4 Source of Addresses

Addresses are calculated from the displacement in an instruction, a base register, and an index register (if specified); this is called a calculated address. Independent cells will obtain all the parameters that make up the address from the cell itself. Dependent global cells will obtain the displacement from the instruction that was sent on the inter-cell bus, the base and index register are always from the cell itself.

In addition to the calculated address, a new concept of a given address is used. A given address is an address that is given following an instruction and is used in place of a calculated address. The use of a given address is specified by a special instruction preceeding a regular instruction.

A dependent global cell recognizes a given address by a special control instruction received on the inter-cell bus. This special instruction is called a GC (global control) format instruction. The format instruction is sent from the controller cell to signal the dependent global cells that a given address, in addition to an instruction, is to be sent on the inter-cell bus. The dependnent global cells will then use the word that follows the instruction as an address to execute the instruction.

The independent, controller cell, and dependent under local control cells may also use the given address concept. In this case, these cells will have a special instruction located and executed immediately before the instruction that is to have a given address used with it. The special instruction is called a CC (controller cell) format instruction. Execution of this format instruction tells the cell that an address will follow the next instruction.

5.2.5 Source of Data

All cells have access to data stored in their memory and may also obtain data from their neighbors (each cell has four neighbor cells). Cells may receive data from outside the group via the cells' I/O line. Cells may also receive data from outside the group or from other cells in the group via the inter-cell bus.

In addition to the conventional means of fetching data to be used in executing an instruction (by an address), another means known as "data follows" is introduced here. This is quite similar to the "given address" concept introduced above. In fact "data follows" is specified in the same manner. Dependent global cells will recognize this by a special GC format instruction received over the inter-cell bus and other cells recognize it by the execution of a special CC format instruction as described above. The use of the "data follows" concept will be interpreted as meaning that the word following the next instruction is to be used as the data to execute the instruction with; the address from the instruction is not used in this case.

It should be pointed out here that there are several other types of the format instructions that could be used: one of these specifies to use the address displacement field of the instruction directly as the data (immediate format), another combines both the given address and the data follows concepts into one format instruction, and finally another one specifies a string of data follows the next instruction.

The format instructions introduced here are called modifiers. This is, they are used to modify the address or data specification (or both) of an instruction.

5.2.6 Execution of Instructions

The instruction repetoire of a cell may be broken down into 12 classes of instructions as shown in Table 5-2. The execution of the instructions depends upon the state of the cell and upon where the addresses, instructions, and data are located. Two of the classes of instructions listed in Table 5-2 are quite unique, namely the CC (Controller Cell) and the GC (Global Control) instructions.

Some of the functions of the CC instructions are: (1) control the inter-cell bus communications section of the cell, (2) generate the GC instructions, (3) generate inter-cell bus communications control signals, (4) control the transmission/execution of instructions in the controller cell, (5) specify a format modification, (6) control the state or level identification of the cell. The cell identified as the controller cell will be capable of executing all the CC instructions; only a very restricted part of the CC instructions may be executed by independent and dependent local cells.

GC instructions are actually instructions or commands sent over the inter-cell bus by the controller cell. They are generated by the execution of a CC instruction and then sent over the inter-cell bus. Some of the functions of the GC instructions are: (1) specify a format modification, (2) control the state of dependent cells on the basis of level identification, and (3) control the state and/or level of individual cells upon the basis of address identification.

A detailed description of the execution of each of the instruction categories for each state of a cell is given in Volume II. Several important points will be mentioned below.

Dependent global cells execute instructions received over the inter-cell bus. A notable difference over conventional instruction execution occurs with the compare and skip instruction categories. The difference being that the program counter is not used in dependent global cells and the results of executing these instructions is to change the level register and change the state to dependent wait. CC instructions are not sent over the inter-cell bus and are therefore not executed in dependent global cells.

Table 5-2. Instruction Categories

1.	LR	Load Register from a memory location
2.	STR	Store Register into a memory location
3.	OPR	An operation is performed between a register and a memory location contents, the results are in a register.
4.	RR	An operation is performed between one register and another register.
5.	R	Single register operation, such as shift.
6.	EXEC	Execute an instruction in a memory location.
7.	COMP	Compare the contents of a memory location (or register) with a register. The results of the comparison are saved in the COMPARISON flip-flops.
8.	SKIP	Test the contents of a memory location (or register) with a register or implied value. The result is true or false.
9.	JUMP	A new sequence of instruction is begun. The jump may be combined with a test to make a conditional jump.
10.	CC	Controller Cell instructions.
11.	GC	Global Control instructions. These instructions control the states and levels of all cells and dependent cell execution of global instructions.
12.	IO	Input-Output instructions. These instructions initiate and control I/O operations.

Dependent cells under local control execute instructions quite similarly to dependent global cells. Of course the instructions in this case are located in the cell's own memory. Cells in this state may execute a very restricted part of the CC instructions namely those to specify a format modification and those to control the state and/or level of the cell. In addition cells in this state will execute certain GC instructions received over the inter-cell bus namely those to control the state of dependent cells on the basis of level identification and those to control the state and/or level of individual cells upon the basis of address identification.

Independent cell instruction execution is quite similar to that in a traditional computer. Just as in the dependent local state only a very restricted part of the CC instructions can be executed. The only GC instructions on the inter-cell bus that will be executed by an independent cell are those that control state and/or level on the basis of cell address identification.

The controller cell operates in basically two modes as far as instruction execution is concerned: the transmit mode and the execute mode. That is the controller cell can either send out instruction over the inter-cell bus or execute them internally. The transmit mode is entered by executing a particular CC instruction. Essentially this instruction causes the controller cell to place the subsequent memory words on the inter-cell bus. The program counter controls the fetch of instructions. The execute mode is also entered by executing a particular CC instruction. In this mode the instruction execution is quite similar to the independent cell. The primary difference occurs in the CC and GC instruction execution. The controller cell can execute all the CC instructions (it is thereby capable of sending out instructions over the inter-cell bus) and the controller cell does not execute any GC instructions since it does not examine the inter-cell bus for any instructions.

It should be mentioned here that not all the format modifications can be used in every instruction category; this topic is treated in detail in Volume II. The material presented in this section is quite complex; however, it is given here to afford some understanding as to how a group operates. For a detailed discussion of the topics given here the reader is referred to Volume II.

5.3 INPUT/OUTPUT

The input/output method is shown in Figure 1-1. Input/Output devices may be connected to the cells directly via a single serial line; to the inter-cell busses (byte parallel), or to the inter-group bus (byte parallel). All connections are made via the I/O connection panel. It is assumed that the sensors will contain their own conditioners so that the signals will be binary digital in nature.

A number of alternative I/O approaches were considered and are reported in detail in volume II. Most of the other approaches utilized separate I/O cells in the system; these approaches have the advantage of eliminating one connection from each cell, however, the system is now more prone to failure due to the specialized I/O cells. The selected approach offers the possibility of using any cell as an I/O cell. Failure in I/O cells may require the plugging/unplugging of connections to affect a reconfiguration, however, one does not lose an entire group or system due to individual failures with the selected approach.

A point that should be brought out here is that in general the system will be used with only several (2 or 3) cells in each group acting as I/O cells. The reason for this is to minimize the need for having to change connections due to reconfigurations. This includes both reconfiguration due to failures and to changes in the computation functions as the mission changes from phase to phase.

6. FAILURE DETECTION AND RECONFIGURATION

Several approaches to detecting failures were considered. The selected approach is a combination of software and hardware methods. The software portion of the detection scheme consists of a central self-test program that is executed by the controller cell and also sent out to the other cells in the group for execution. The other cells will all be in the dependent global state when the self-test program is sent out over the inter-cell bus.. The cell that the program resides in will be the controller cell during the portion that the self-test program is being sent out. Execution of the self-test program results in the cells transmitting certain results over their neighbor lines and checking received results over their neighbor lines. Thus a neighbor cell can report a faulty cell to the controller cell after the self-test program is completed.

The hardware portion of the detection scheme consists of special fault detection hardware such as parity checking, special error detecting codes, feed-back checks on decoders, etc. This type of detection is carried on continuously while the software detection is carried out periodically.

The controller cell or the cell that sends out the self-test program does not send results to a neighbor cell but will send a response word to the group switches upon successfully passing the self-test program. The group switches contain special timing circuitry to time the periodic reception of these response words. Thus the group switches will report on the status of the group.

A high probability of mission success and availability result from the distributed processor system because an individual group can continue to operate in the presence of a failed cell(s), and because the system can continue to operate in the presence of a failed group(s).

For purposes of reconfiguration analysis the mission was subdivided into three basic phases: (1) non-critical, (2) critical, and (3) Mars orbital; each of these phases requires a different computer configuration to meet the requirements. Each phase was analyzed to determine the reconfiguration procedures for the different types of failures that may occur: (1) working cell fails, (2) controller cell fails, (3) group switch fails, (4) executive group fails and (5) conditioner/sensor fails.

Reconfiguration procedures for the non-critical and Mars orbital phases are quite similar. Working cell failures are handled by the controller cell of the group. Controller cell failures are detected in the group switches which are in turn monitored by the executive group; the executive group therefore handles these failures. As a part of the reconfiguration routine here, the entire group that contains the faulty controller cell must be reinitialized. A failure of the executive group results in a reinitialization of the entire system; the reconfiguration procedures can be handled automatically by having a small back-up executive residing in another group or can be initiated by the astronaut via the display/control panel.

Reconfiguration procedures for the critical phases are somewhat different. The critical computations will be redundantly performed by one group, so that in case of failure of the group that is performing the critical computations and outputing of critical signals, the other group may take over outputing of critical signals. This implies that some form of a reliable switch be available at the sensors to switch control signals upon indication from the distributed processor system. Any failure in a group that affects the critical computations will result in the controller cell informing the group switches that the group has failed. The group switches then signal the output switch accordingly. Following this of course, reconfiguration can take place in the failed group by reconfiguration routines similar to those for the non-critical and Mars orbital phases. Therefore, further redundant groups can be made available by reconfiguring after control of the critical output signals has been switched. All of the reconfiguration routines are described in detail in Volume II.

7. CELL AND GROUP SWITCH DESIGN

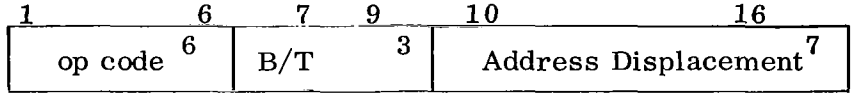
7.1 CELL DESIGN

Processor Features:

The features to be incorporated in the processor section of the cell were investigated. In particular the following were evaluated: (1) multiple accumulators, (2) index-bank registers, (3) word length, (4) instruction word format, (5) macro instructions and (6) byte manipulation instructions. The primary consideration in evaluating the features is their ability to save storage or bits of memory. The reason for this is that the majority of the cell is devoted to storage. It is also desirable to minimize the amount of hardware devoted to the processor section of the cell. The reason for this is that the processor section is characterized by an irregular connection of logical gates and registers. The memory section is characterized by a regular connection structure of gates and registers. Therefore, yields can be increased on cell wafers by easily incorporating redundant memory logic whereas it is relatively difficult to implement redundant logic in the processor section to increase yields. Thus as much of the processor registers as possible are implemented in the memory section of the cell (this includes accumulators and index-bank registers).

Evaluation of multiple accumulator features resulted in the decision to provide four (4) accumulators. The following word lengths were given consideration: 12 bit, 14 bit, 16 bit, and 18 bit. Longer word lengths were not considered since they are primarily useful to provide various features in the processor that increase the speed capability. A 16-bit word length was selected and is shown below:

Bits:



This instruction word format provides for 64 basic op codes for instructions and a 128 word bank from the address displacement. Three bits are provided for the index-bank registers (referred to as B and T registers). There is no real distinction between the two since full word length index and bank registers are used. The selected use of the three B/T bits is to provide for the specification of the use of one of two B registers in conjunction with one or none of three T registers. This scheme provides 5 index-bank registers and the capability of double indexing.

7.1.1 Functional Description

A general block diagram of the cell was shown in Figure 5-1. A detailed block diagram of the cell containing all the processor functional registers is given in Volume II. Four accumulators are stored in memory and one accumulator is provided in hardware to carry out arithmetic and logical operations for the presently used accumulator from memory. The five B/T registers are stored in memory. A lower accumulator, program counter, memory buffer register, and memory address register are provided in hardware. In addition to various control registers the cell also contains a Cell ID register, state register, and a level register; these were

discussed in Section 5. The cell also contains two communication buffer registers, one is used for the inter-cell bus and the other for the I/O and neighbor connections to the cell.

Volume II contains the detailed instruction list for a cell. A summary of the types of instructions was given in Section 5. The use of op code extension results in many more instructions than the basic 64 that can be derived from a 6 bit op code.

There are a variety of internal interrupts that can occur in a cell, one of these being the counting down to 0 of a real time clock register. The detailed steps for processing interrupts is given in Volume II. The mechanization of the neighbor communications is given Volume II. One point that should be noted here is that only one line is used between neighbor cells. This is mechanized by assigning cell ID addresses in a checkerboard pattern so that an evenly addressed cell has all odd addressed neighbors, etc. and allowing evenly addressed cells to make requests at one time and odd cells at the next time, etc.

7.2 GROUP SWITCH DESIGN

Referring to Figure 1-1, one can see the relation of the group switches to the cells and the busses. The group switch controls the information transfer between the inter-cell busses and the inter-group busses. Essentially the group switches look like another cell to the group that is the system executive. This is due to the fact that the system executive group will control the communications over the inter-group bus. This is quite similar to the controller cell in the group controlling the communications over the inter-cell bus.

The main difference between the control of the two busses is that the controller cell receives an instantaneous response to its commands that it sends out to the other cells in the group whereas the executive group does not receive an instantaneous response to its commands sent over the inter-group bus to other groups. This results in different communication system mechanizations for the two busses.

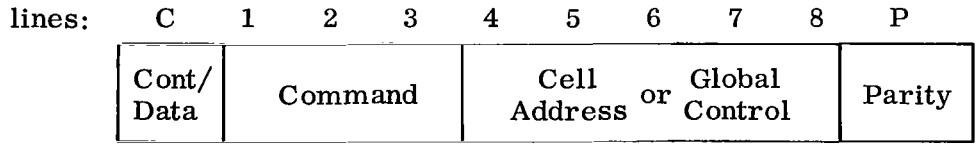
A detailed block diagram of the group switch is given in Volume II. The group switch is relatively simple in terms of complexity. It contains two buffer communications registers, one for each of the busses, an address ID register for identification purposes, a command register for holding commands received from the system executive group, a request register for holding requests from the controller cell of the group that it is a part of, and control circuitry for reception and generation of bus commands. It will also contain BITE timing circuitry. This circuitry consists of a flip-flop that is alternately set-reset by a command from the inter-cell bus. The rate at which it is set will determine the voltage generated across a circuit. Two out of tolerance level detectors across this circuit will check for a high and low value of voltage. If the voltage is out of tolerance a failure signal is issued.

8. COMMUNICATION BUS DESIGN

8.1 INTER-CELL COMMUNICATIONS

The distributed processor system contains two types of busses, the inter-cell and the inter-group bus. Inter-Cell communications within a group are carried out over the group's inter-cell bus. The inter-cell bus is under complete control of the cell in the group designated as the controller cell. This cell will operate the bus by the execution of certain CC instructions.

A total of ten lines are used for the inter-cell bus. One line is used to denote control or data and is designated the control line. Another line is used for parity; future studies may dictate the need for more lines for error control, however for the present only one line is assumed. The remaining eight lines are used for control or data words. All words over the bus are composed of eight-bit bytes whether they are control or data words. The control words are identified as such by use of the control line. Some control words require more than one byte. The format of the first byte of a control word is shown below:



This byte provides for addressing up to 32 cells. Eight basic command codes are provided, however more commands are provided for by using the second byte of the control word for command extension.

Some of the types of inter-cell communications to be carried out include: (1) controller cell sending or receiving data to/from a cell under controller cell's command, (2) controller cell commanding communication between two cells, (3) controller cell scanning cells for communications requests and establishing the requested communcations and (4) controller cell issuing GC (global control) instructions to one or more cells. To accomplish these communications functions, communication bus commands and control words were designed, the detailed design of which is given in Volume II.

In all cases the controller cell issues the proper sequence of control words to accomplish the desired communications. One of the control words is a command to report the communications request of a cell. Upon reception of this command, the cell must respond with its communications request word. This word will identify to/from which cell data is desired. The controller cell completes the requested communications by issuing the proper control words upon examining the request word. The control words comprise basic operations such as input, output, etc. Some of the control words are simply GC instructions as given in Section 5; these are used to command changes in the states of cells, level of cells, etc. A detailed logical design of the formation of the inter-cell bus control words and the mechanization of the commanded operations is given in Volume II.

8.2 INTER-GROUP COMMUNICATIONS

Inter-group communications are accomplished by using the inter-cell and the inter-group busses. This type of communications is somewhat different from that given above for inter-cell communications.

The difference is that the cells in a group are under immediate control of the controller cell and immediately respond to commands over the inter-cell bus. However, the executive group that is in charge of controlling the inter-group bus use does not have the groups (represented by the group switches) under its immediate control. The groups periodically sample the group switches to see if any commands have been placed in a command register by the executive group. (Any group can do this; however only the group acting as the executive group will be issuing such commands.) The groups then respond to these commands. It can be seen that there will be some delay in getting a response to these commands. This delay is primarily dependent upon the rate at which the group switch is sampled by the controller cell of a group for these commands. Once the controller cell picks up these commands, it will immediately respond by commanding the proper cells in its group to begin communications.

The groups will place requests for inter-group communications in the group switch (in a request register). These requests are periodically sampled by the executive group. It is based upon these requests and any individual requests of the executive group, that commands will be placed in the group switches for inter-group bus control.

The types of communications functions to be carried out are similar to those listed for the inter-cell case (except for the GC instructions). To accomplish the desired communications functions various commands were designed for the inter-cell bus and the inter-group bus. It should be noted that the group switch contains an ID (address register) just as a cell does. It will therefore respond to commands addressed to it just as a cell does. However, how it interprets a command may be different than how a cell does. Each command uses the same format as that given previously namely 3 bits for the command code and 5 bits for the address. In addition the group switch interprets the command codes over the inter-cell bus differently from those received over the inter-group bus. The detailed design of the commands is given in Volume II.

9. SOFTWARE ANALYSIS AND DESIGN

The distributed processor has many levels of control. This results in many levels of control (executive action) in the software to program the computer. The executive programs may be considered at three levels: (1) the system executive, (2) the group executive, and (3) the cell executive.

9.1 GROUP EXECUTIVE

Basically, the group executive controls a group of cells. Each group must have allocated one cell as a controller cell; this cell will act as the group executive. Since each cell in a group is identical any cell can perform as a controller cell. The group executive functions will reside in one or more cells in a group (more than one as required due to storage limitations of an individual cell) and one of these cells will be acting as a controller cell at any one time. Although the different programs that comprise the group executive may be in different cells, this set of cells (which always includes the present cell that is in the controller cell state) is called the group executive.

The group executive has many functions. Basically, it must control the operation of the cells in the group and the inter-cell communication bus, it must reconfigure the group resources due to changes in processing requirements and due to failures, and finally it must interface with other groups. The functions may be broken down into six subfunctions:

1. Control system resources
2. Furnish global programs to dependent cells
3. Allocate system resources
4. Test system hardware and software, and respond to any malfunctions
5. Reconfigure system upon
    a. Normal phase change
    b. Malfunction
6. Interface with other groups, including the system executive

Each of the above functions of the group executive were considered in detail in volume II. Some of the more important items will be noted below.

The computational tasks will be performed in the operational cells. These cells may require executive services such as getting I/O data over the inter-cell bus, getting intermediate computational data from other operational cells, setting clocks, etc. The group executive will provide such services under the subfunction of "control system resources." These services will be provided in two modes: periodic and background. Periodic services are those required at fixed

predetermined intervals of time while background services are those fitted in between the periodic according to some priority scheme. The services will also be provided in two ways: by command or request. Command services are those completely controlled by the controller cell, e. g., the controller cell commanding one cell to send another cell certain data. Request services are those requested by the operational cells, e. g., an operational cell requesting data from another cell. The function of controlling the system resources may also be considered as one of controlling the control and data flow over the inter-cell bus.

The periodic control of system resources is handled by a real time clock interrupt routine. Upon an interrupt from this routine the controller cell will service a particular task in one of the cells. This may require the requesting of a communications request word from the cell. The controller cell examines these requests, checks their validity and takes the necessary action to complete the requested service. The background mode of service is quite similar to the periodic except that no real time clock interrupt routine is used. In this mode the cells are all serviced according to some priority scheme.

The cell or cells that contains the group executive contains all the global instructions that are to be executed by the dependent cells. Because these instructions are sent on the inter-cell bus and the controller cell controls the inter-cell bus, the controller cell sends out all the global programs. These global programs may be simply sent out by the controller cell or they may be sent out in response to a request either by the periodic or background service routines mentioned above.

To allocate system resources and schedule the use of these resources, there must be made available to an executive scheduler program a list of the hardware resources in the group and a list of what tasks must be performed by the group. In addition, a list of the software requirements of each task must also be made available. The executive scheduler, using these lists, will begin to allocate the resources to the tasks. Periodic tasks are assigned first according to their priority, then the background tasks are assigned. The first assignment primarily consists of deriving the inter-cell bus usage schedule. The tasks may then be physically assigned to cells.

The group executive will contain the central self test program to test the group hardware and software. This program will be sent from the controller cell to all the other cells in the group by placing these cells in the dependent state.

Reconfiguration is required under two conditions. There are the normal changes due to phase changes in the mission; e. g., new programs are required during the coast phase which are different from the requirements of the Mars orbital phase. Secondly, reconfiguration may be required upon a failure or malfunction of a cell, group, bus, etc. Reconfiguration procedures for phase changes are designed to affect a smooth transition for the periodic programs of one phase to those of another. Reconfiguration around failures results in the executive scheduler program being called and given a new hardware resources table. If the tasks cannot be handled by the group, the system executive will be informed. The system executive will then attempt to reschedule and allocate the tasks throughout the entire computer system.

The group executive will be required to interface with the group that contains the system executive. This interface will involve task assignments, communication of back up data, use of the inter-group bus for communications, etc.

9.2 SYSTEM EXECUTIVE

In addition to the group executive functions presented in the previous section, one of the group executives will also contain additional functions giving it the capability of a system executive. The system executive is designed to coordinate the groups and not really to control them. By this means, the system reliability is increased.

The system executive will contain a complete up-to-date list of the total hardware resources in the computer system. It will be responsible for assigning the tasks throughout the computer system.

The system executive is responsible for control of the inter-group bus in much the same manner as the group executive is responsible for control of the inter-cell bus. The system executive interfaces with the group switches and essentially this is similar to it acting like the group executive and the group switches looking like cells. The system executive will follow a routine that is quite similar to the control system resources routine described above.

The system executive will be forced to handle many cases of reconfiguration: phase changes, faulty assignment of tasks, and failures. The most difficult reconfigurations are due to the latter two cases, both are handled in a similar manner by entering the executive scheduler routine and proceeding to re-assign tasks among the groups.

9.3 CELL EXECUTIVE

Each cell in a group will require a cell executive routine. This routine will vary in size and complexity from cell to cell depending on the specific tasks to be performed in each cell. In any case the cell executive must be a small, minimal program. A major part of the cell executive is that of processing interrupts. Some of the types of interrupts are: (1) machine errors, (2) inter-cell bus I/O, (3) illegal data such as attempted division by zero, (4) illegal operation, and (5) real time clock goes to zero. Each of these interrupts is considered in detail in volume II.

10. RELIABILITY SIMULATION

The distributed processor organization was subject to a reliability simulation for a manned Mars mission application. A Monte Carlo reliability analysis program was used for the simulation. This program generates reliability and availability data based on simulating the organization.

Monte Carlo techniques of analysis refer to the simulation of random variables in a process by the generation of random numbers or sequences. The Monte Carlo program solves the probability of failure equation in reverse. It generates a random number which is evenly distributed between zero and one. It sets this equal to the probability of failure and solves for time to failure. This is done for each piece of equipment (cells, group switches) in the system. The time to failure is compared to the time in the phase of the mission to determine if a failure occurred. If a failure occurred, spare equipment is brought in (if available) to replace the failed equipment.

A different configuration is required for the non-critical, critical, and Mars orbital phases. Each of these phases requires a different simulation model. The mission consists of the following profile: (1) Non Critical, (2) Critical, (3) Mars Orbital, (4) Non Critical, and (5) Critical.

The program starts with the first phase of the mission and examines all the equipment as described above. The program then changes to the next phase of the mission and follows the same procedure for the computer configuration for the new phase. Of course if a mission failure occurs in a critical phase the program is terminated for this mission.

The program collects the statistics for each phase. Critical mission phases are primarily concerned with mission failure (probability of success) and the program will record these failures. Other phases of the mission are concerned with availability of the computer system. For these phases the percentage of the equipment that is operating will be recorded.

When all the phases of the mission have been completed, the process is again repeated. By repeating this process, statistics are collected for probability of success and availability. As more and more runs are made the error in the results decreases.

Numerous cases were simulated varying parameters such as, cell and group switch failure rate, number of spare cells per group, number of spare inter-group busses and group switches, number of spare inter-cell busses, on/off failure rate ratio, and percent of failures affecting the busses. The detailed results of each case are given in volume II. A few of the results will be given below.

Figure 10-1 shows a curve of the average availability over the entire mission as a function of the number of spare cells in each group. The conditions as shown are for an MTBF of 200,000 hours for each cell and group switch, 1 inter-group bus, and 1 inter-cell bus per group. The average availability is 97.25 percent with no spares and rises to 99.6 percent with two spares per group.

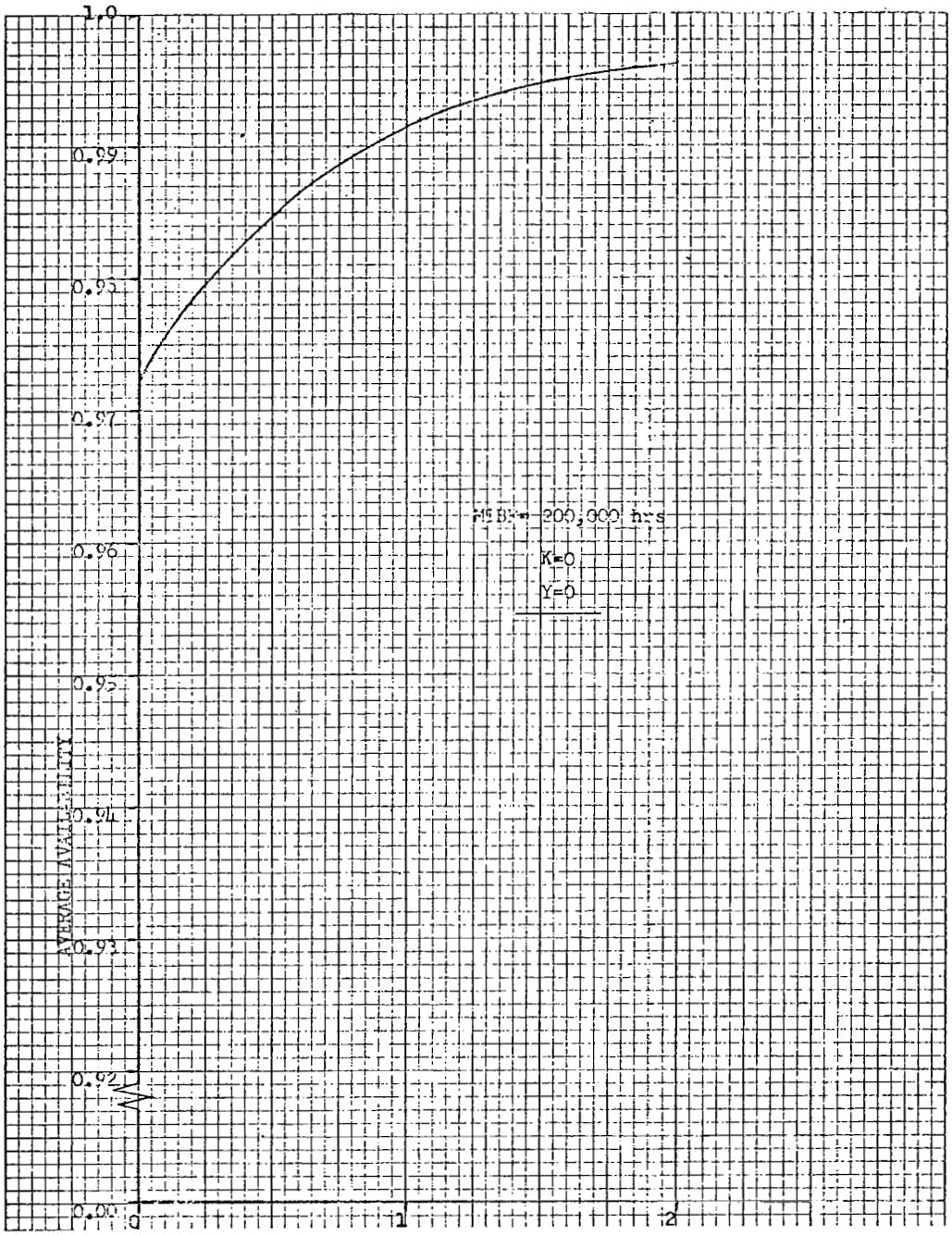


Figure 10-1. Average Availability vs Number of Spare Cells Per Group

The effects of cell and group switch MTBF (on) on average availability are shown in Figure 10-2. These effects are shown parametrically as a function of the number of spare cells per group. The conditions on the curves are no spare busses in the system. The curve for N=2 is dotted between 75,000 hours and 200,000 hours since no point was actually obtained at 75,000 hours.

Probability of success (P_S) was considered for the critical phases of the mission. It should be noted that 1000 runs were made for each case and therefore the results can only be extrapolated to the third decimal point. In all the cases run with MTBF's of 200,000 and 500,000 hours no mission failures resulted. Therefore a flat curve at $P_S = 1.0$ results for these values. With MTBF's of 75,000 hours there were a number of mission failures and the values for P_S ranged from 0.967 to 1.000. Figure 10-3 shows the effects of ON/OFF failure rate ratio on P_S. As may be seen, the only effects were with an ON MTBF of 75,000 hours. The high probability of success of this system is due to the many logical paths available for reconfiguration.

In addition curves were obtained for average availability showing the effects of spare inter-group busses, spare inter-cell busses, on/off failure rate ratio, and percent of failures effecting busses. Availability as a function of mission time was also obtained for a number of cases. Curves for the average number of cell, group switch, and bus failures were also obtained.

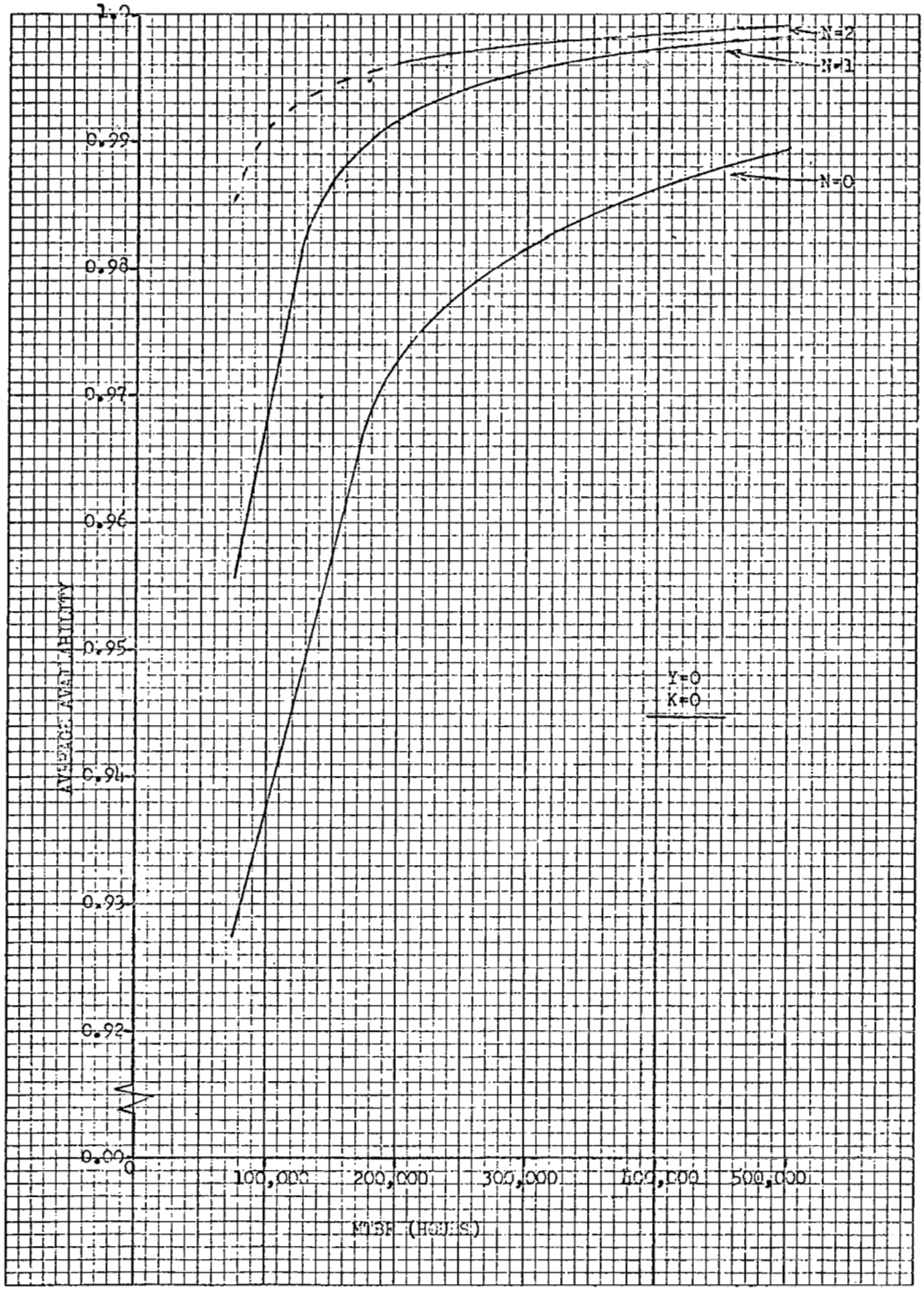


Figure 10-2. Average Availability vs Cell MTBF Varying Number of Spare Cells

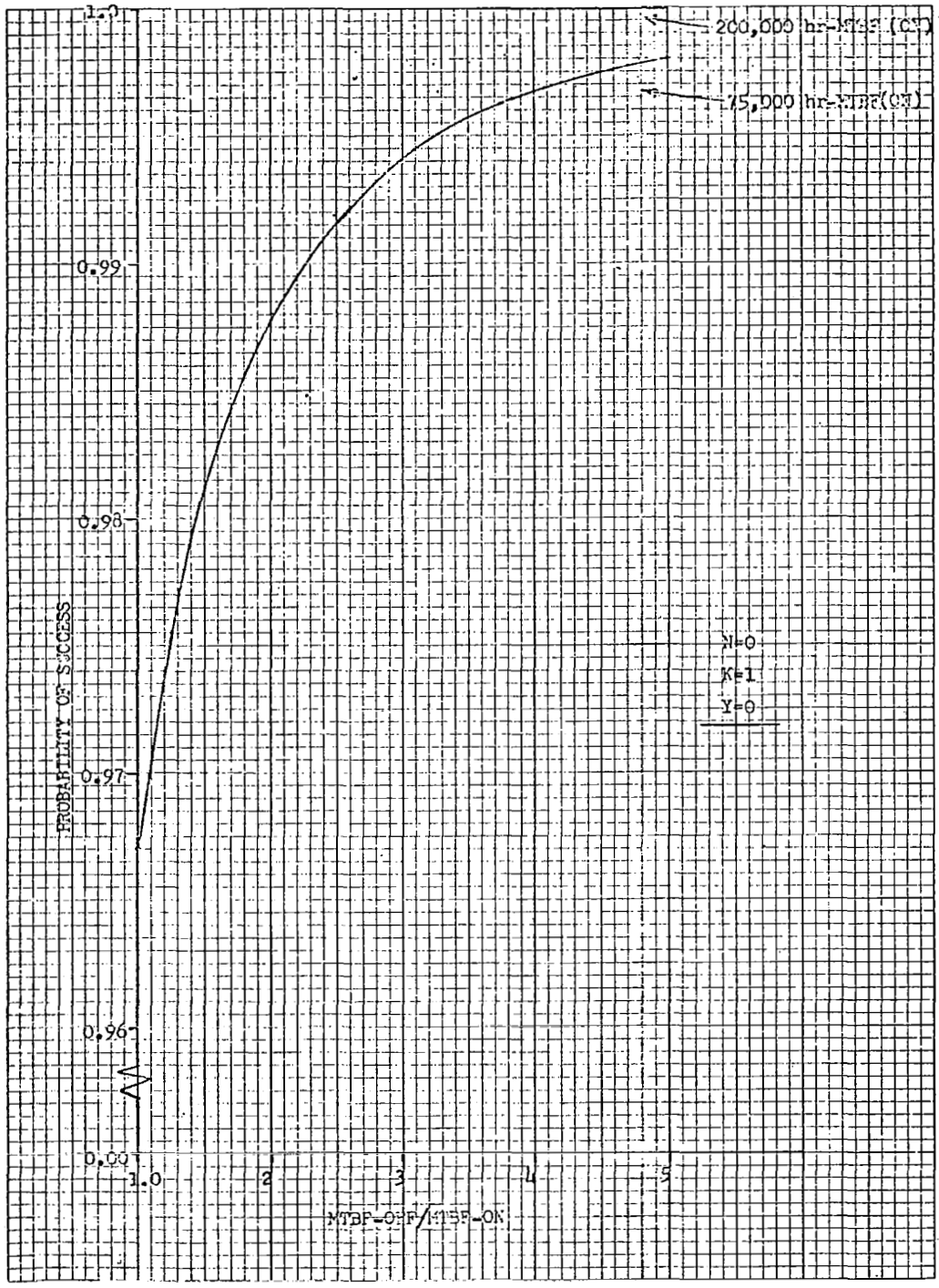


Figure 10-3. Probability of Success vs MTBF-OFF/MTBF-ON

REFERENCES

1. Study of Spaceborne Multiprocessing Final Report, North American Rockwell/Autonetics Division, April 1967, C6-1476.10/33.