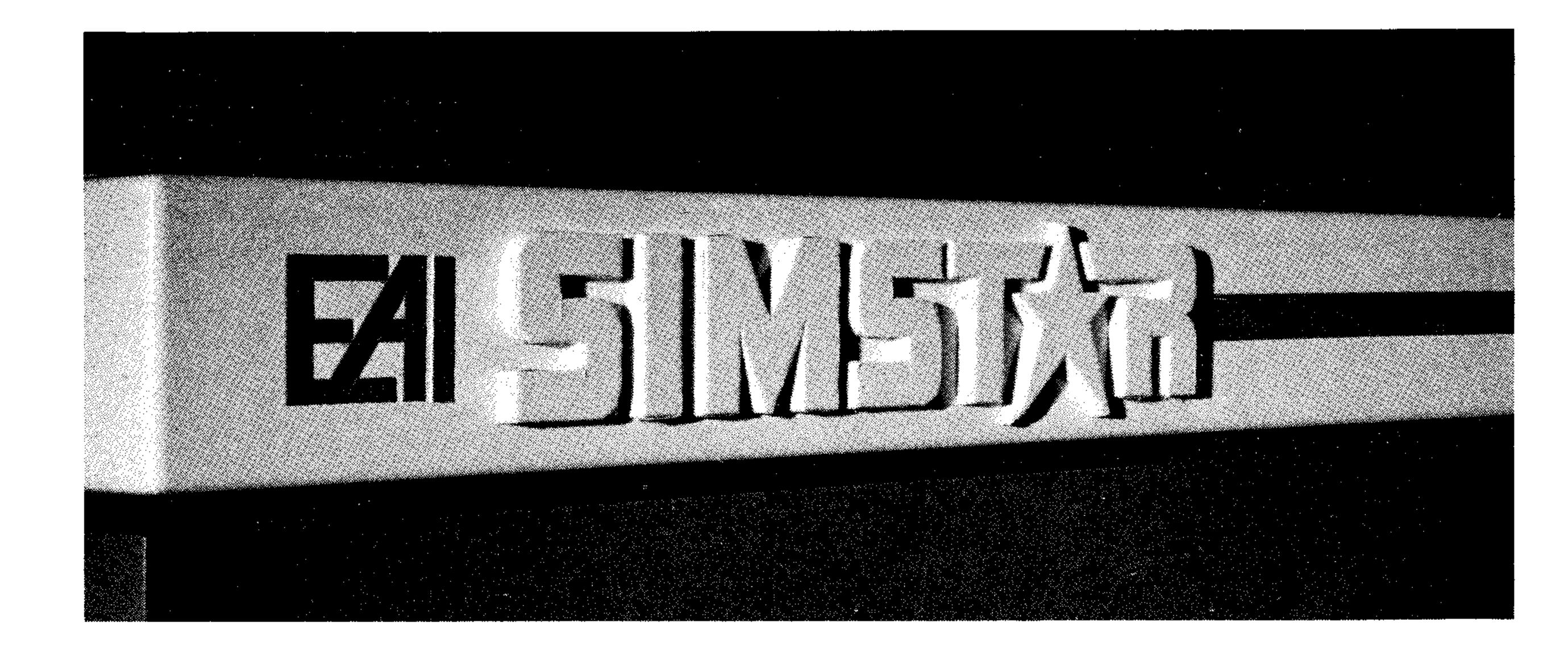


Bulletin No. 04 July 1983

## SIMSTAR – AN ATTACHED MULTIPROCESSOR FOR DYNAMIC SYSTEM ENGINEERING



This paper was presented July 13 at the 1983 Summer Computer Simulation Conference in Vancouver, British Columbia, Canada.



# Electronic Associates, Inc.

SIMSTAR - AN ATTACHED MULTIPROCESSOR FOR DYNAMIC SYSTEM ENGINEERING

> By: J. Paul Landauer Electronic Associates, Inc. West Long Branch, NJ 07764

#### ABSTRACT

This paper describes a new multiprocessor that has been developed by
Electronic Associates, Inc. for scientific
analysis of dynamic systems. Proven parallel and sequential computing methods are
integrated in SIMSTAR(TM) to provide a
unique capability for mixed continuous/
discrete system simulation and signal
processing. In contrast to the earlier
manually programmed analog and hybrid
computers, SIMSTAR is a completely automatic device driven from high level
software in a Host data processing
computer.

The requirements for dynamic system simulation in different fields is developed in the paper to establish a criteria for measurement of SIMSTAR performance relative to alternate computers. New concepts in system architecture, component technology, and system communication features are described. An overview of the SIMSTAR programming system is covered to see the flow from the simulation language input to the program segments for the various processors. Also, a brief discussion of program operation from the Host terminals through a run-time executive is described.

## INTRODUCTION

SIMSTAR is a new computing tool for engineering analysis of dynamic systems. By combining the latest linear and discrete integrated circuit technologies, the earlier hybrid computer concepts have been extended into an automatic, high performance device which can be attached to a range of medium scale data processing systems. Initially, the Host system is one of the GOULD S.E.L. 32 series. Applications of SIMSTAR include conception, evaluation, and optimization of dynamic physical systems in all of the engineering fields. By integrating a high-speed, economical digital arithmetic processor with a unique automatic, stored-program parallel processor, equivalent computing speeds over 200 million operations per second are obtained.

The parallel processing system can be expanded to over 400 mathematical computing blocks which are interconnected by a solidstate switch matrix. Functions implemented in these blocks include continuous integration, linear arithmetic, non-linear operations, and logical processing. For most applications, the digital arithmetic processor is assigned those equations which represent the slowly changing environment. The system, being designed to operate in this environment, is usually modeled on the parallel processor which can automatically handle state variable discontinuties. simultaneous algebraic relationships, and natural frequencies over 1 kHz. The complexities of sophisticated numerical integration techniques can be avoided by use of this combined approach.

Since the SIMSTAR Multiprocessor is completely automatic in operation, it can be programmed in the same fashion as an automatic data processing machine. The user may prepare programs either in FORTRAN or in a high level Continuous System Simulation Language. Setup and interaction with SIMSTAR is handled by the built-in digital arithmetic processor.

## THE APPLICATION

The design and evaluation of complex dynamic systems has been one of the most challenging engineering tasks since the time of Sir Isaac Newton. Many different mathematical and computer means have been developed as aids. The primary mathematical tool has been modeling by systems of ordinary and partial differential equations along with supplementary algebraic equations. When these models are solved, the process is often called "Simulation," although this term is rather ambiguous, since it is also used for simple animation of an environment for display or training purposes.

During the last twenty-five years, the computer tools for solving these models have been primarily combinations of electronic analog and digital machines which are usually termed hybrid computers. Early analog computers used mechanical devices which were relatively unreliable. The first digital computers were very slow, expensive, and relatively difficult to program. Though the 1970's, both of these technologies evolved so that powerful, economical digital, analog, and hybrid systems became available. During this same span, the complexity of applications increased at the same rate so that today simulation is still the most demanding computer application.

SIMSTAR is a quantum step forward in computer technology to meet the increasing demands of engineering simulation. It can be effectively used in large-scale simulation laboratories in the Aerospace, Nuclear, and Electrical fields as well as central scientific/engineering computer facilities in high technology companies. Convenience features available on the Host computer such as color graphics, network access, and Computer Aided Engineering / Design tools can be used with the SIMSTAR attached processor. Accordingly, complete integrated system design studies can be carried-out amongst different divisions of companies.

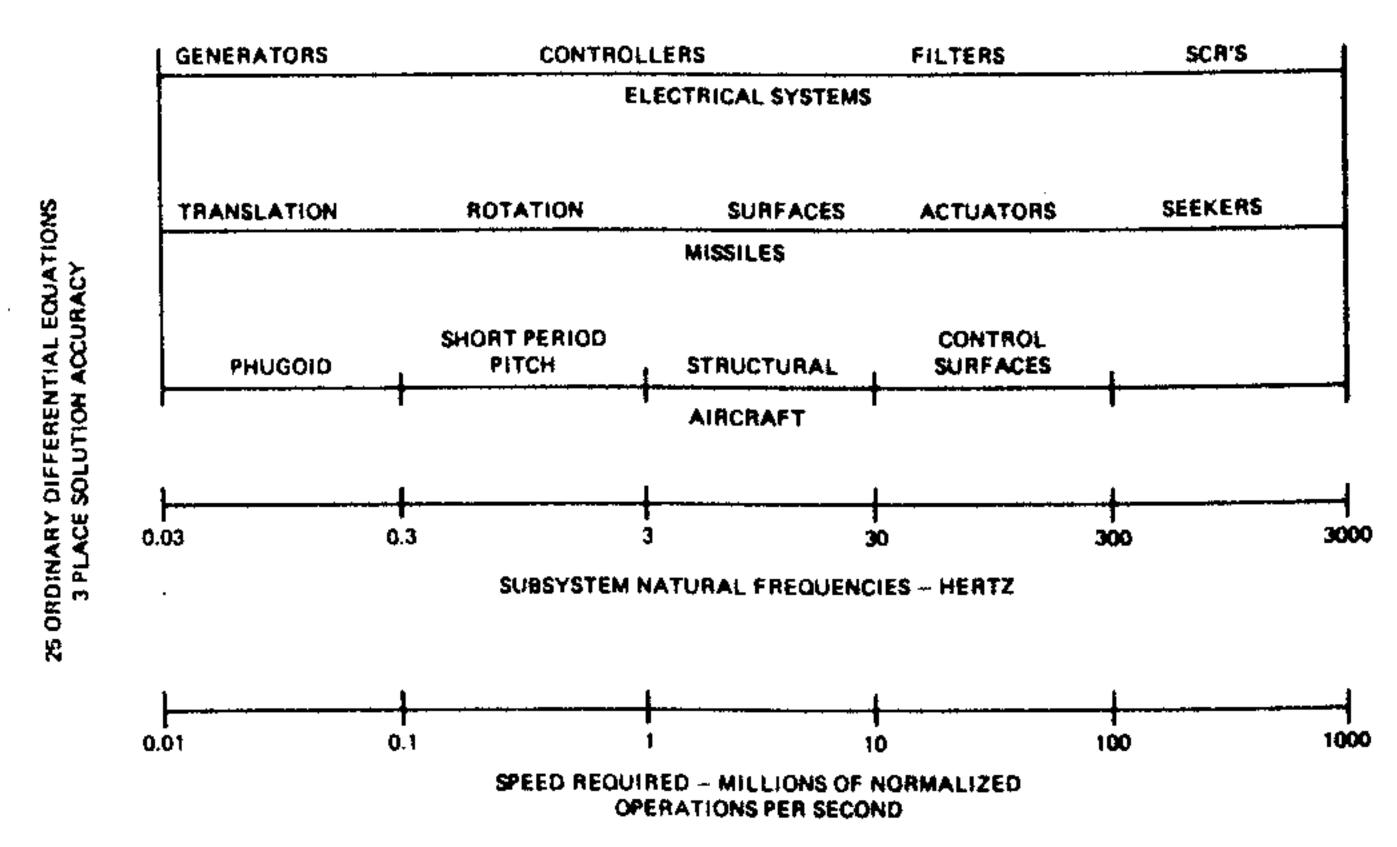


Figure 1 - The Application Requirement

## Analysis of the Requirement

A display of the requirements in three major fields of application is shown in Figure 1. The abscissa of this chart is divided into five decades of computer speed in terms of millions of "Normalized Operations-Per-Second" (NOPS), which is a simple method of comparing processor performance for this class of application. A Normalized Operation (NOP) is essentially a simple memory reference instruction such as LOAD or ADD. Multiply or Divide are usually counted as 3 NOPs. If we assume that we need to solve a system of 25 ordinary differential equations with appropriate non-linearities to three place solution accuracy, the equivalent natural frequencies of the system can be determined as shown in the chart for the various speeds shown. That is, real-time simulation of

this system operating at a natural frequency of about 3 hertz will require a digital processor performing approximately one million NOPS. If the frequency increases to 30 hertz, the equivalent processing speed will go to 10 million NOPS for the same problem complexity and accuracy. Assuming a single integration method, the speed requirement is proportional to the natural frequencies of the subsystems being modeled.

For aircraft, the Phugoid Mode calculations operate at about 0.01 hertz. The Short Period Pitch frequency is about 1 hertz, Structural dynamics are in the range of 20 hertz, and the Control Surface deflections will range from about 30 to 100 hertz. If the Control Surfaces are represented by a system of 25 transfer functions with limiters, the real-time solution will require about 20 million NOPS. For missile system simulations, the frequencies also vary thoughout this entire range from the Translational equations at less than 0.2 hertz to the small infrared or radio frequency Seekers at over 300 hertz. In between are the Rotational equations at about 2 hertz, the Control Surface dynamics at about 20 hertz, and the Actuator response which is about 50 hertz.

Another important application for SIMSTAR is modeling electrical power systems for power inverters, battery storage systems, and general power control of rotating machinery. Again, the frequencies vary from the mechanical time constants of generators through the speed controllers, the line filters, and the Silicon Controlled Rectifiers (SCR) which are in the DC-AC inverters. Proper representation of these systems can require a computing capability of over 100 million NOPS.

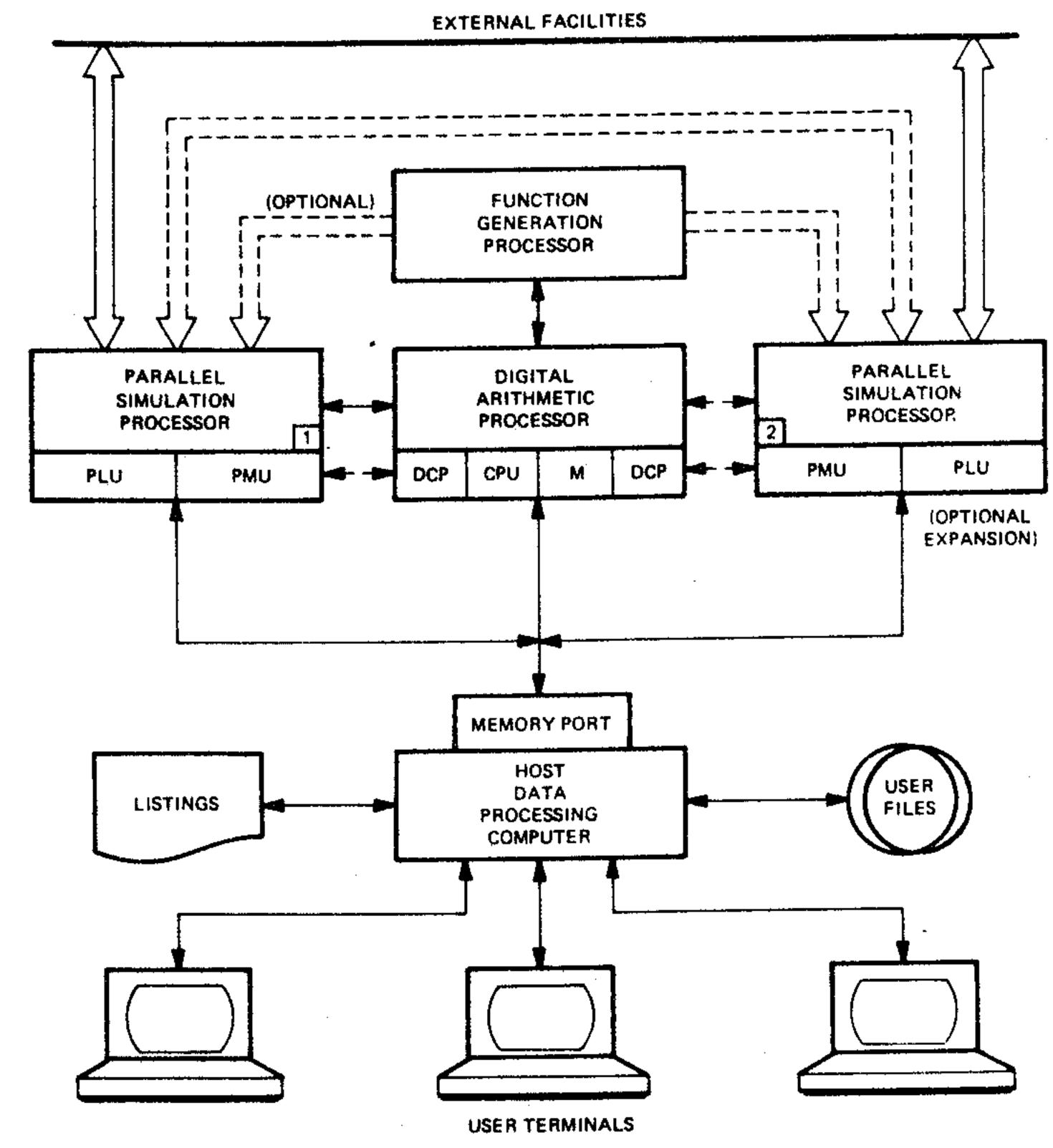


Figure 2 - The SIMSTAR System Architecture

#### MULTIPROCESSOR ARCHITECTURE

A block diagram of the SIMSTAR dual multi- processor attached to a Host computer is shown in Figure 2. In most real-time simulation laboratories, the SIMSTAR model must interface to various external facilities to test actual subsystems or for human interaction. Various types of continuous display and recording devices may also be used. Parallel analog and binary signals as well as a digital data port are available.

The basic SIMSTAR multi-processor, which is attached to a Host computer, is composed of a single Parallel Simulation Processor (PSP) and the Digital Arithmetic Processor (DAP). The second Parallel Simulation Processor and the Function Generation Processor are optional devices in the system and provide parallel extension of the computational power. Included in the PSP is a Parallel Logic Unit (FLU) and a Parallel Math Unit (PMU), which provide the heart of the computing power of SIMSTAR. Sequential digital computing is provided by the DAP composed of a 32 bit CPU and MOS memory. With an optional Floating Point Accelerator (FPA), this processor is approximately equivalent to a VAX 11/780. A basic setup and monitoring interface capability between the PSP and the DAP is provided as an inherent part of the minimum SIMSTAR system. In addition, a Data Conversion Processor (DCP) can be added to the DAP for high speed/accuracy direct memory data communication with the PSP. With this interface, the PSP can also communicate data directly to/from the Host at high speed if it is needed as a part of a computational task. All programming of the DCF is done using Channel Programs setup by the DAP.

The user operates from terminals on the Host and uses the normal file handling capability of that system. Also, high-level software to load the SIMSTAR processor executes on the Host, and appropriate listings can be obtained. The object program can then be downloaded into the DAP and the PSP to operate the simulation. The DAP, in turn, will load the Function Generation Processor (FGP) with appropriate data and programs.

In most applications, one of the PLU's will act as the master timing control for the entire multiprocessor program, since it has a programmable crystal clock and interval timing capability built-in. This device interrupts the DAP for time-critical processing. Data produced is stored in the DAP memory from which it can be accessed by the Host for display on graphics terminals, listings, graphical hard copy, or permanent storage on the disk.

The subsystem models developed in the research and development department on SIMSTAR can be made available to many

design engineers by establishing an applications library on the Host digital data processing system. These libraries may be further used by an Executive routine to provide Problem Oriented Languages. In this way, design engineers can make effective use of Simulation methodologies without the need to develop new mathematical models each time.

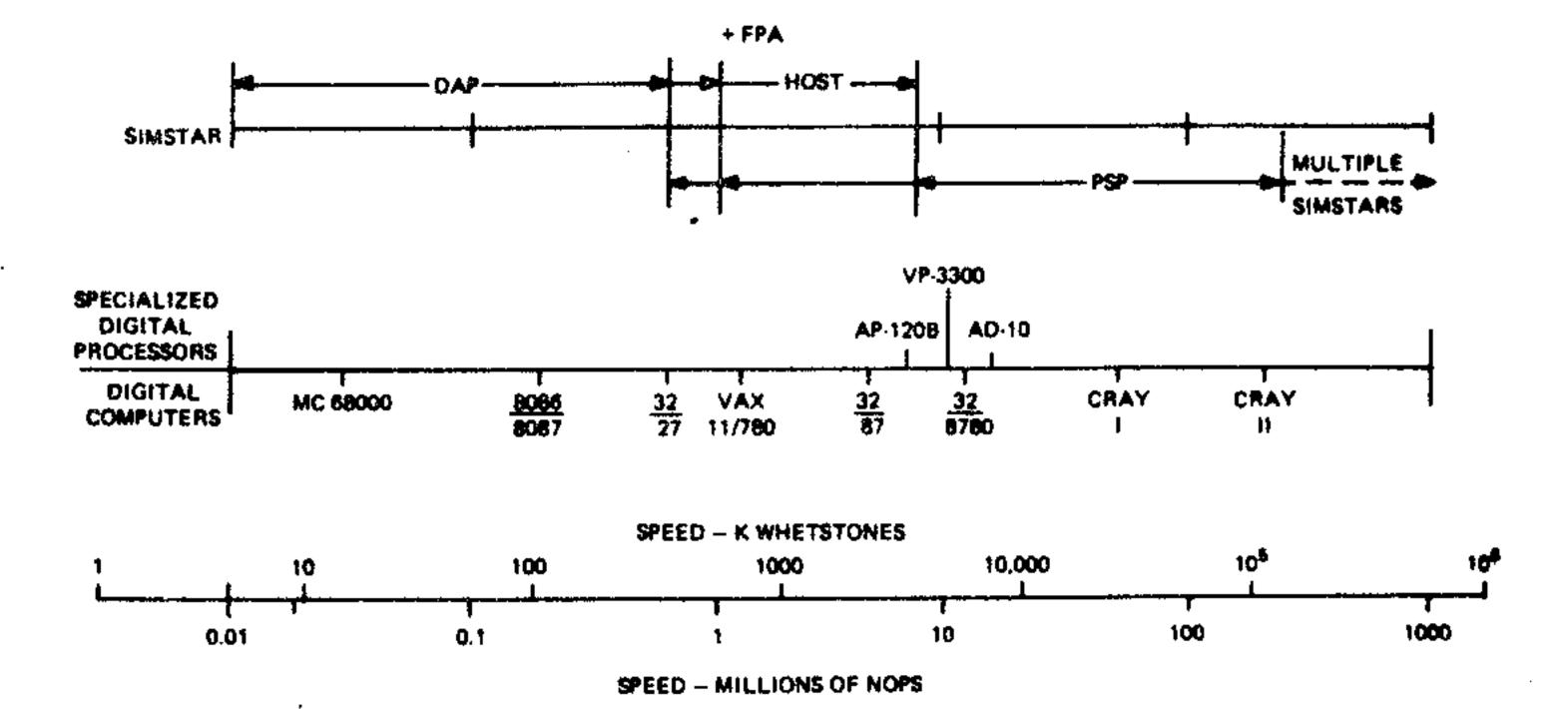


Figure 3 - Comparison to Typical Commercial Computers and Processors

#### Alternate Computer Implementations

The next chart, shown in Figure 3, presents typical commercial computers and specialized digital processors as compared to the SIMSTAR attached multiprocessor. The abscissa is again the speed required in Normalized Operations-Per-Second ranging from 10,000 to 1 billion. The corresponding speed in Kilo Whetstones is also shown on this chart since most computer performance is quoted in terms of the Whetstone benchmarks. A speed of 1 million NOPS is about 700,000 Whetstones.

The various computers being considered throughout this performance range go from the Motorola 68000, which is a 16/32 bit LSI processor, to the Cray II, which is a quadruple 60 bit floating-point machine with 4 nanosecond cycle time. Costs of these digital processors range from about \$4,000 for the 68000 to about \$25 million for the Cray II. It can be seen that the DAP covers the range from 10,000 NOPS to about 1 million NOPS when it is enhanced by the Floating-Point Accelerator (FPA). The equivalent speed of the DAP with the FPA is about 670K Whetstones. In contrast, a VAX 11/780 is about 800K Whetstones. If digital processing performance greater than this is required, the user can apply the Host digital computer to the simulation. The chart shows that one could use a GOULD S.E.L. 32/8780 dual processor for this function to increase the digital floatingpoint performance of SIMSTAR to about 8 million NOPS.

The Parallel Simulation Processor is the only practical device for speed requirements from 20 million NOPS to 200 million NOPS. Of course, the PSP can also be used at lower speeds to overlap with the Host or the Digital Arithmetic Processor.

If speeds greater than 200 million NOPS are required, one can perform these calculations on the PSP at somewhat reduced accuracy without concern about numerical instability. Applications requiring up to nearly 10 kHz can be handled in real time on the PSF if one can live with up to 5% error. In the future, it is planned that the system could be expanded to 3 SIMSTAR dual multiprocessors in a system for a combined equivalent performance of over 500 million Normalized Operations—Per—Second.

Obviously, one would not use Cray computers in this type of application since the cost is prohibitive. However, a possible alternate is one of the specialized high-speed digital processors. For example, the Applied Dynamics AD-10 can be viewed to operate in the range of 10-20 million NOPS. The sixteen bit fixed point operation may not be sufficient since one often requires greater resolution and range for slower variables. A new floating-point version of the AD-10 has been announced, but no systems are yet installed. The digital arithmetic processor in SIMSTAR provides real-time 32 bit floating-point capability for variables and 64 bit precision for integration.

Other high-speed specialized digital processors which have been used to perform various simulation tasks include the Floating-Point Systems AP-120-B and the FPS-100. These devices are somewhat slower than the AD-10 but have full floating-point computation capability and some specialized software useful for simulation.

The VP-3300 is a version of the earlier MAP-300 manufactured by CSPI. This unit operates through a Common Memory interface to GOULD S.E.L. 32 computers which eliminates the usual program and data transfer overhead. Specialized software for scientific computation is also available for this device. A version of this processor can be added to a SIMSTAR system for function generation or coordinate transformation.

All of these digital devices can assist in real-time simulation only to 20 or 30 hertz. Above this, the SIMSTAR Parallel Simulation Processor is the only practical choice. Of course, if one can justify not modeling the higher frequency terms, the slower devices can be used.

It can be seen in Figure 3 that SIMSTAR is designed to provide the appropriate type computing devices for the accuracy and speed requirements needed in many large scale simulations. If a particular model has lower frequencies and does not demand real time, the problem can be time-scaled upward to take full advantage of the extremely high-speed performance of the PSP. This excess of computing power simplifies the task of the simulation engineer since he does not have to optimize the utilization of the processors or be concerned about complex numerical integration methods.

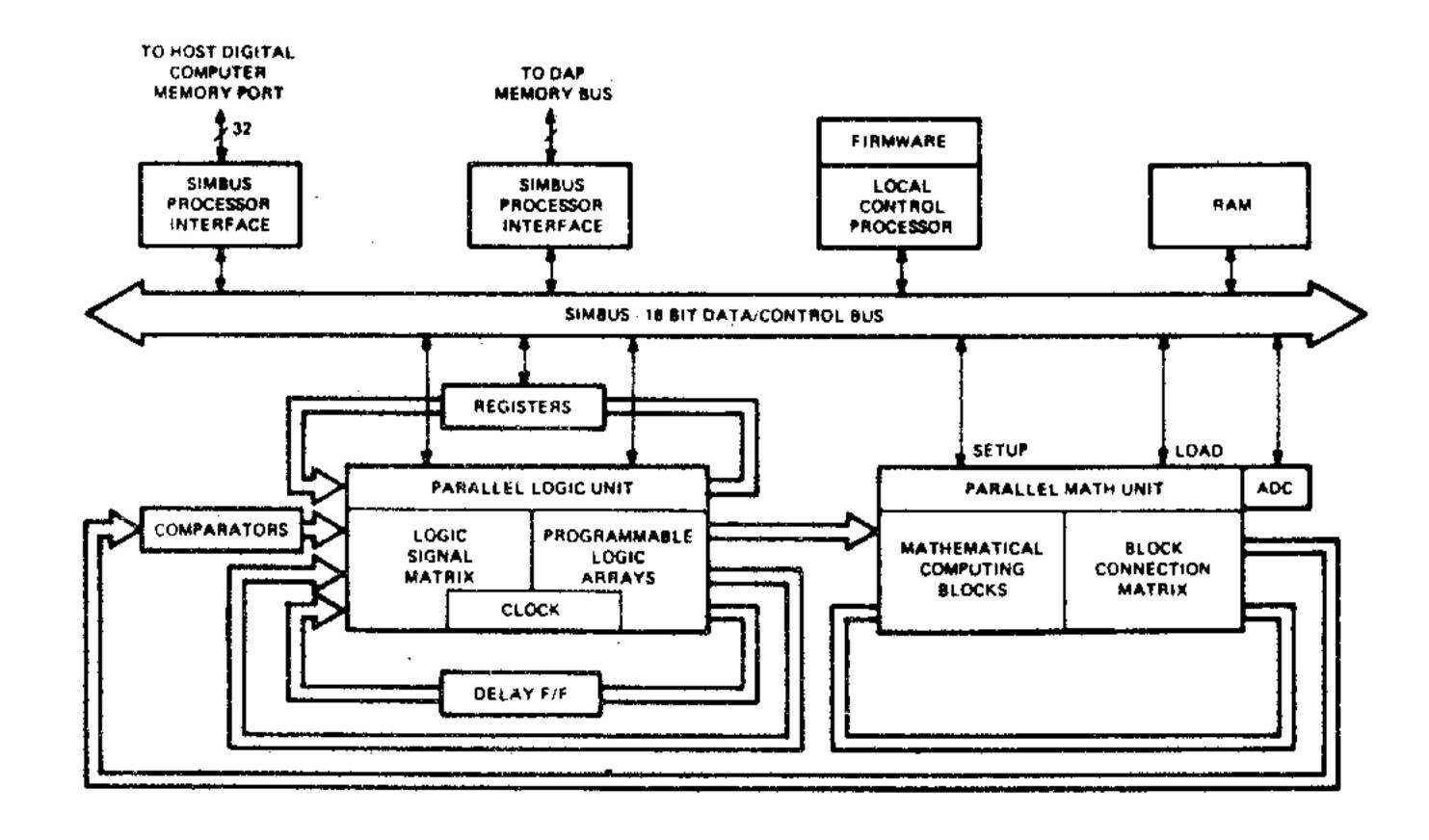


Figure 4 - Functional Block Diagram of the SIMSTAR Parallel Simulation Processor

#### PARALLEL SIMULATION PROCESSOR

The Parallel Simulation Processor (PSP) is the main computing device in the SIMSTAR system. It incorporates high speed, continuous Mathematical Computing Blocks and a set of Programmable Logic Arrays to solve a model composed of nonlinear differential equations combined with switching and control as needed. The concept of dedicating computing devices to specific terms in particular equations has been successfully used for many years in analog and hybrid computing methodologies. SIMSTAR incorporates these proven concepts into a totally new machine which provides automatic operation from a Host digital computer.

A functional block diagram of the PSP as implemented in the SIMSTAR system is shown in Figure 4. The Parallel Logic Unit is composed of the Logic Signal Matrix and the Programmable Logic Arrays. Mathematical Computing Blocks combined with the Block Connection Matrix make up the Parallel Mathematical Unit of the PSP. The double lines represent parallel communication paths to handle many mathematical/ logical variables simultaneously. Selected continuous mathematical variables produced by the PMU can go to the Comparators which test for a specified threshold level. If the variable exceeds that level, a logic signal is generated through the Logic Signal Matrix to implement control functions in the PLU. Finally, the logical states out of the PLU drive the switching devices and mode sequencing of the Mathematical Computing Blocks.

Interprocessor communication between the Local Control Processor (LCP) and the various computing and monitoring systems in the PSP is handled by the 16 bit SIMBus. This is an adaptation of the Intel Multibus. The SIMBus Processor Interface (SPI) units provide memory mapped communication between the Host, the DAP, the RAM on the SIMBus, and the various storage devices in the PMU and PLU.

Loading and setup of all components in the PSP is handled by the LCP. That is, conversion from floating-point data in the Host or the DAP to the optimum gain settings and configuration selection is performed automatically by the LCP. Unused elements of each hardware macro are defaulted out so the user need not be concerned. System monitoring for overrange and signal instability is handled by the LCP. Also, the autorange intelligence in the ADC to provide high resolution readout is performed by the LCP. Finally, the Local Control Processor performs the periodic automatic device calibration in the PMU.

#### Parallel Mathematical Unit

In addition to automatic, high-speed computing capability, SIMSTAR includes innovative component and system design to ensure accurate results over a wide dynamic range. Essentially, the same concepts used for floating-point digital computation using REAL numbers is employed inside the computing blocks of the Parallel Mathematical Unit. A continuous computing device such as the PMU employs electrical signals to represent dynamically changing variables. All information may be considered normalized so that the maximum value of a signal is near unity. Although an exponent cannot be carried with the signal between math blocks, within a block, pseudo floating-point methods are used to provide full four digit accuracy over a greater range than ever possible before. In particular, non-linear blocks which produce the product, quotient, or square root of signals employ an automatic change of exponent to extend the accurate dynamic range by 16-to-1. This operation is completely transparent to the user.

Since a signal is accurate to about five digits, this provides much greater range than used in the earlier analog computers. The monitoring system uses an Autoranging feature to allow accurate readout of smaller signal values. Parameter data loaded into coefficient devices is in floating-point and the built-in microprocessor firmware establishes an optimum combination of binary fraction value and gain for each run. Integrator gain can range from 10 ~ to 10 ~ with full four digit resolution. That is, the user's program simply sets the gain as a floatingpoint number in this range and the appropriate internal settings, gains, and integrating devices are selected by the LCP for optimum accuracy.

All of these pseudo floating-point features combine to make the SIMSTAR system much more User Friendly than any other device for simulation. Furthermore, the accuracy of solution for typical applications is at least ten times that available with the earlier hybrid computers.

#### Parallel Logic Unit

Problem timing and control is provided by the PLU in combination with the built-in crystal clock and the timing registers shown in the feedback around the PLU. Of course, a continuous variable representing time in the problem can be produced using one of the Integrating computing blocks. The implementation of sequential logic in the PSP is done by using the delay flip-flops (F/F), which are provided in the feedback of the PLU as shown in Figure 4.

### The SIMBus/Local Control Processor

Setup of the PSP and digital run-time communication between processors is performed through the SIMBus, which is a 16 bit high-speed implementation of the Intel Multibus. This bus is also used to communicate with the Host digital computing system in a memory mapped fashion. The Local Control Processor (LCP) with its associated Firmware is shown above the SIMBus in Figure 4. This device translates commands from the user program referencing Math Computing Blocks to the actual implementation of those functions on the hardware macros of the PSP. The LCP provides the on-board intelligence in the PSP for any binary communication required from user programs to the physical computing subsystems. Also, a Random Access Memory (RAM) on the SIMBus is loaded by the LCP during the setup or operational phases of the PSP. This memory, as well as many registers throughout the PSP, can be accessed through the SIMBus Processor Interface (SPI) which is a 32 bit memory port to the Digital Arithmetic Processor. As shown in Figure 4, a second Simulation Processor Interface is used to connect the SIMBus to the Host digital computer memory bus. This allows the Host to read the binary status of the entire PSP and store it on disk. To restore a program, a transfer from the disk file back into these memory locations will re-establish the PSP operation very rapidly.

For readout of the signals from the Mathematical Computing Blocks, an autoranging Analog-to-Digital Converter (ADC) is attached to the Block Connection Matrix. The LCP can acquire this data and translate it into appropriate formats for floating-point transfer back to either the DAP or the Host.

## PSP Computing Performance

As discussed under Applications, engineering simulation studies demand computing performance which exceeds that available with any standard digital computer system. In SIMSTAR, this speed requirement is met by employing a large set of parallel computing blocks which operate continuously upon signals. Earlier in this paper, the equivalent performance of a SIMSTAR attached multiprocessor was quoted up to 200 million Normalized Operations—Per-Second (NOPS). This was based upon a

typical mix of components on a unit having dual parallel processors operating at an average solution frequency of 300 hertz.

DIGITAL OPERATION HARDWARE MACRO	ADD/SUB/AND	X; MULT/DIV	LOAD	STORE	SIGN TEST	NOP'S PER PASS	NOP'S PER STEP
LIMITED INTEGRATOR/TS	7	- <del>6</del>	- <del>15</del>	4	1	14	- 44 56
SUMMER/SWITCH	2	2	2	1	1	12	48
SUMMER/LIMIT	2	3	2	1	1	15	60
MULTIPLIER (DIV/SQRT)		1	2	1	<del></del>	6	24
3 INPUT MULT	-	2	2	1		8	32
SINE/COSINE	3	8	5	1		33	132
f(1)	4	2	6	1	2	19	76
f(2)	10	6	13	1	4	46	184
COMPARATOR	1		2	1	1	5	20

NOPS - Normalized Operations per Second

Figure 5 - Equivalent Digital Operations for SIMSTAR Hardware Macro Components

This equivalent performance is derived (1) by estimating the number of normalized operations required for a digital processor to perform the same computations as each of the components of the SIMSTAR parallel processor. The chart in Figure 5 shows nine basic component types including functions of two (or more) variables with the NOPs required for a single integration pass and a fourth order Runge-Kutta integration step. If a single pass predictor/ corrector integration method were adequate, the NOPs per Pass could be used. However, this type of method can introduce large errors for discontinuities in state variables.

COMPONENT	TYPICAL #/PP	EQUIV. NOP/RK4	TOTAL NOP/RK4
LIMITED INTEGRATOR/TS	40	100	4000
SUMMER SWITCH	48	48	2304
SUMMER LIMIT	24	60	1440
MULTIPLIER DIV/SQRT	24	24	576
THREE INPUT MULTIPLIER	16	32	512
SINE/COSINE	6	132	792
f(1)	24	76	1824
f(2) or f(3), f(4)	16	184	2944
COMPARATOR	24	20	480
PARALLEL LOGIC UNIT	1		~500
······································	15372		

AT 20 STEPS/CYCLE (0.1% ERROR) 300,000 NOP/CYCLE AT 300 HERTZ (AVG.) (0.1% SOL ERROR) 90 MILLION NOP/SECOND)

Figure 6 - Single SIMSTAR Parallel Frocessor Performance

On the next chart (Figure 6), the total SIMSTAR PSP performance is calculated based upon the typical number of each of these components which can be programmed in parallel. Including 500 NOPs per RK-4 step for the PLU, over 15,000 normalized operations are required per integration step. If we assume 20 steps/cycle, RK-4 integration will result in about 0.1% error. A digital processor would need to compute 300,000 NOPs per solution cycle to match the performance of a fully expanded PSP. If we assume an average frequency of 300 hertz for signals in the model on the PSP, the typical error per component is less than 0.1%. The equivalent digital speed required for this computation would need to be about 90 million NOPS.

A fully expanded SIMSTAR multiprocessor can include two PSPs operating in
parallel plus a Vector Processor and the
Digital Arithmetic Processor. Also, if
needed the Host digital processors can
contribute to the simulation task. Therefore, in total, the system performance at
this speed would be approximately 200
million NOPS.

#### DIGITAL ARITHMETIC PROCESSOR

Sequential processing in SIMSTAR is performed by the Digital Arithmetic Processor (DAP) which provides up to 1 million NOPS performance. As shown in the block diagram (Figure 7), the DAP is built around a System Bus which has a bandwidth of over 6 million 32 bit words per second. A sophisticated 32 bit CPU is available on this bus. This CPU has firmware for floating-point arithmetic in both single and double precision. A basic Integrated Memory Module (IMM) containing 1 megabyte of 600 nanosecond MOS memory must be connected to the bus. This is further expandable within a SIMSTAR to a maximum of 2 megabytes. The addition of the optional Floating Point Accelerator (FPA) increases the speed of the DAP from 430 K Whetstones to 668 K Whetstones for the single precision Whetstone benchmark. For the double precision Whetstone benchmark, the equivalent speeds are 204 K Whetstones

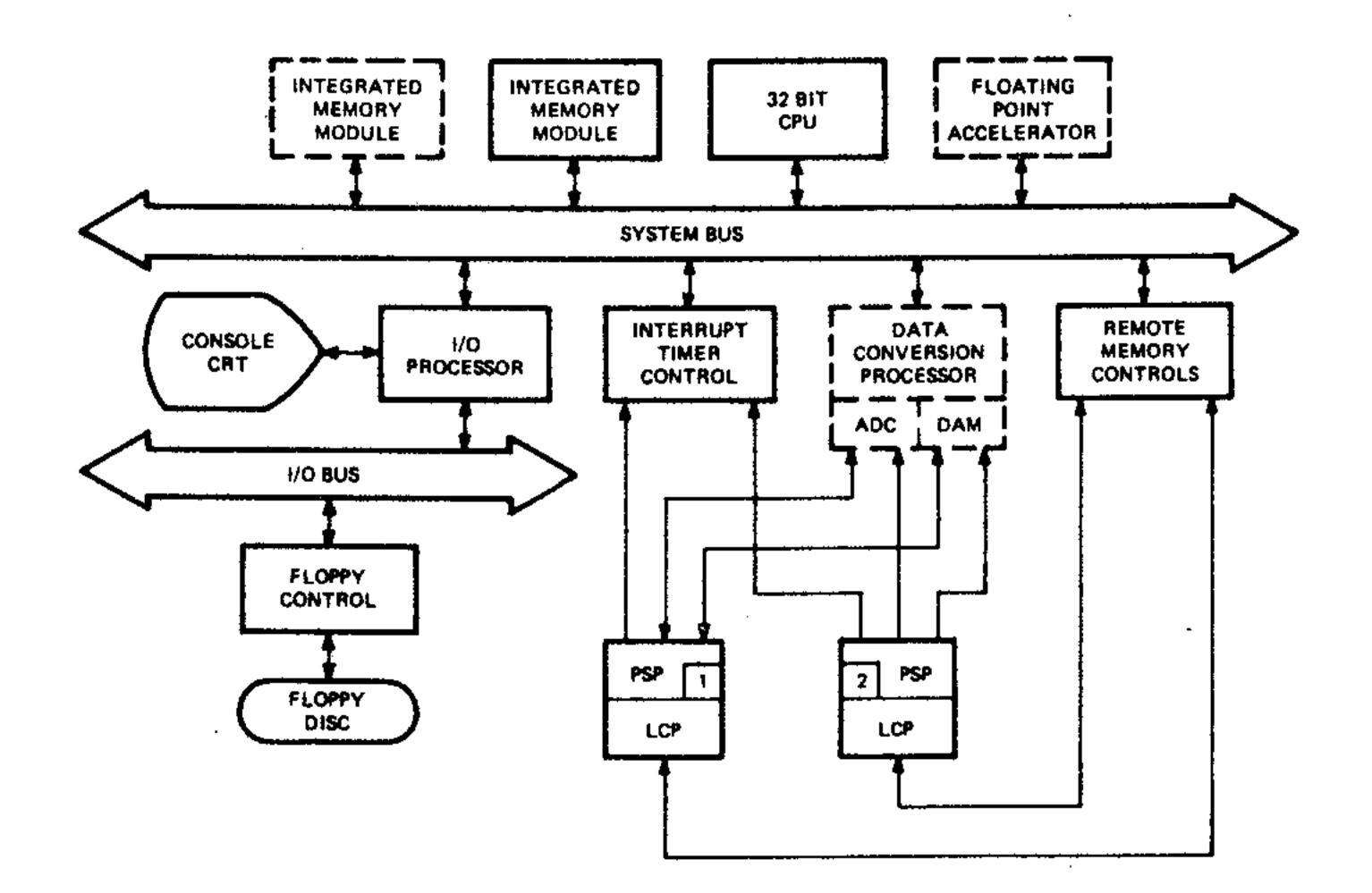


Figure 7 - Digital Arithmetic Processor

without the FPA and 465 K Whetstones with the FPA. A single precision floating point add is performed in 1.65 microseconds, while a single precision floating point multiply is performed in 2.25 microseconds.

I/O devices are shown attached to the bottom of the System Bus. The basic DAP is operated through an Input/Output Processor, which translates the System Bus to a simple 16 bit I/O Bus. This Bus controls the floppy disk which is used to boot up the SIMSTAR Operating System in the DAF. A user can add a console CRT to the I/O Processor to provide local operation of the SIMSTAR multiprocessor. The rest of the devices attached to the System Bus are interfaces to the Parallel Simulation Processor in SIMSTAR. The Interrupt/Timer Control provides a programmable, priority interrupt capability into the CPU as well as a variety of timers for controlling the DAP and Data Conversion Processor operation. The Data Conversion Processor (DCP) is an optional high speed data conversion system to communicate with one or two parallel simulation processors. This is an intelligent, programmable device in which up to 6 tasks may be activated simultaneously to convert continuous signals to floating-point data. Also, six additional tasks can be activated to convert floating-point data to continuous signals. In a SIMSTAR system with two PSPs, half the DCP tasks are controlled by each PLU.

The basic interface between the DAP and the PSP is through the Remote Memory Controls (RMC) which are a memory mapped means of accessing the various data devices in the PSP and the LCP. The RMC device is setup to consume part of the extended memory addressing range of the CPU so that programs may simply store floating-point data into specified memory locations and the LCP will perform the necessary operations to control or setup the PSP. As was previously described in the PSP section, the Host computer will also communicate through the SIMBus in the PSP so that it can down-load into the DAP local memory. An appropriate executive is provided with SIMSTAR to take a block of data/program at a time through this Remote Memory Control for transfer to the appropriate local memory of the DAP.

## FUNCTION GENERATION PROCESSOR

One of the most important functional requirements in simulation applications of the SIMSTAR multiprocessor is the representation of empirical data such as that derived from wind tunnel tests of aerospace vehicles, steam tables for nuclear reactor simulations, and compressor maps for turbine engine simulation. Many types of special-purpose devices and computer programs have been developed to implement this requirement. In SIMSTAR, single variable function generation devices are incorporated in the PMU and an efficient software package is provided for the DAP to

represent multi-variable functions. In addition, one can add a Function Generation Processor (FGP) which extends the performance of SIMSTAR for this requirement by up to 2 decades of speed.

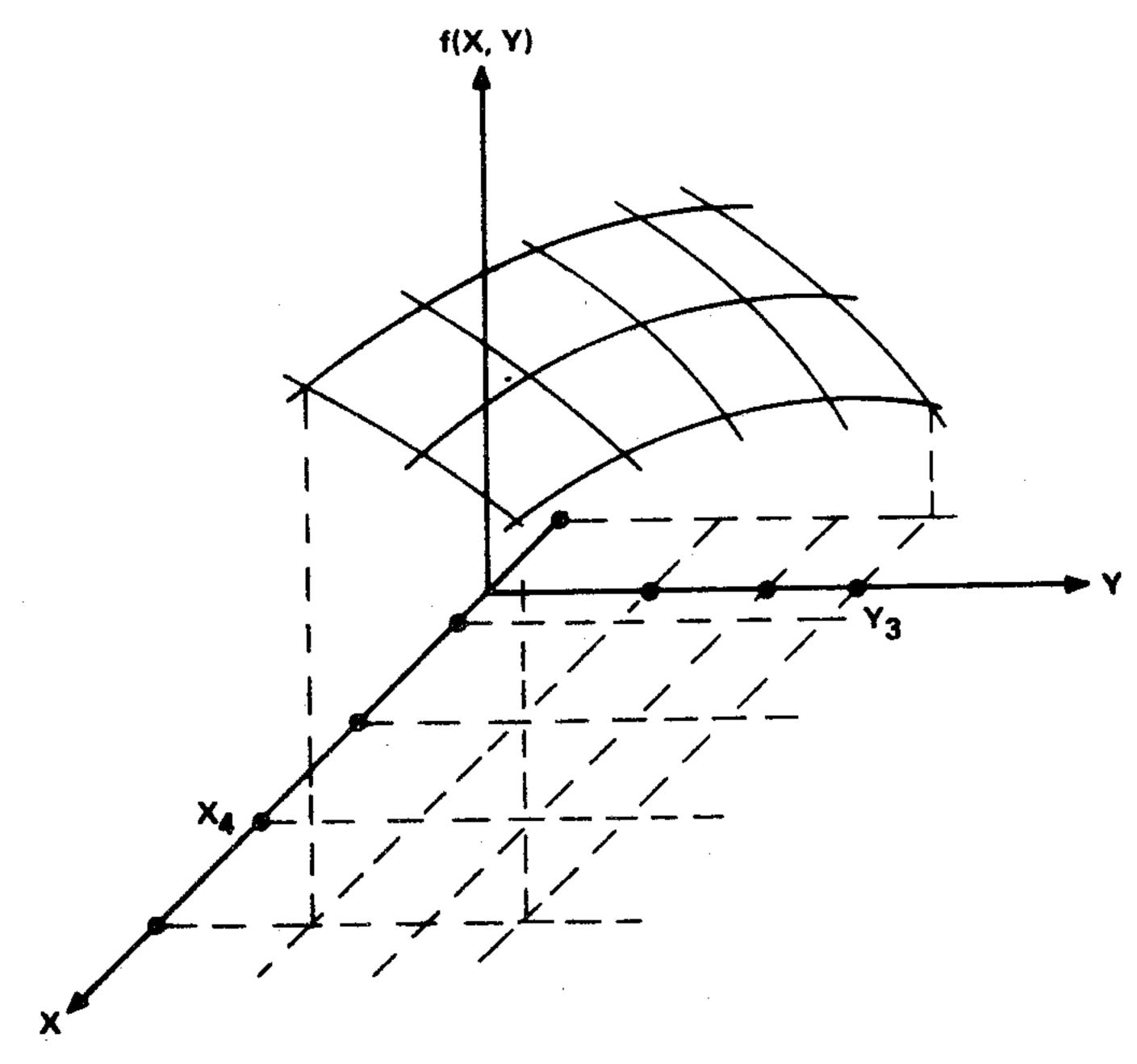


Figure 8 - A Function of Two Variables

A function of 2 variables, as shown graphically in Figure 8, is a surface which can be defined by a set of ordered triples (F, X, Y). Usually, the data is aligned on a rectangular grid so that the function values may be stored in a two dimensional array F( I, J ) in which the subscripts represent grid values of the independent variables. Function generation is a table look-up and interpolation process; that is, given an instantaneous value of X and Y, the computing device must compute an address in the two dimensional array to pick up the appropriate four adjacent points and perform the necessary interpolation on that "surface" to approximate the original continuous function. This requirement extends to functions of 3 variables to interpolate between surfaces and up to 6 variables in some applications. In a typical data set for an aerospace vehicle, a mixture of 50-100 multi-variable

## Basic\_SIMSIAR

DSFG F1 DIGITALLY SET FUNCTION GENERATORS

Continuous function of one variable with 45 or 221 equally spaced breakpoints.

A total of up to 24 DSFGs are provided in each Parallel Mathematical Unit.

FGSYS F1,2,3,4 FUNCTION GENERATION SYSTEM

Specialized software package for DAP to compute discrete floating-point function values at high speed. Compatible with

Figure 9a - Basic SIMSTAR Function

Generation

DCP for use with PSP. Also operates on

Digitally Set Function Generators (DSFG) which provide continuous generation of functions of 1 variable. The breakpoints can be selected for either 45 or 221 equally spaced values between -1.1 and +1.1. These units are setup through the LCP and operate completely independent of any other processor during the problem solution.

For the representation of multi-variable functions, the FGSYS — Function Generation System software package is available for the Digital Arithmetic Processor (or the Host data processing system). This package uses a unique code generation process for a high level of user convenience while obtaining optimum code efficiency at run-time. The internal operations of the run-time code are automatically optimized to eliminate redundant processing for common function argument sets.

Optional Eunstion Generation Processor (FGP)

MVFG F2,3,4

MULTIVARIABLE FUNCTION BENERATORS

Continuous function of two to four variables. Setup by DAP and run independent.

VPFGSYS F1,2,3,4 &(5)

VECTOR PROCESSOR FUNCTION GENERATION SYSTEM

High speed floating-point array processor and software operating from DAP. Compatible with Data Conversion Processor for use with PSP.

Figure 9b - Elements of the Function Generation Processor

An optional Function Generation Processor can be added to SIMSTAR as mentioned in the previous architecture section. This sub-system is composed of two different elements as shown in Figure 9b. First, the specialized Multi-Variable Function Generation (MVFG) subsystem is available which provides continuous function generation for 2, 3, or 4 variables. Second, a Vector Processor Function Generation System (VPFGSYS) is available for floating-point function lookup and interpolation at about 10 times the speed of the DAP. This unit operates independent of the DAP once it is setup and can communicate directly with the Data Conversion Processor.

A block diagram of a function of 2 variables implemented in the MVFG is shown in Figure 10. The continuous signals (X,Y) produced by Mathematical Computing Blocks in the PMU are connected by the switch matrix to the X and Y Input Normalizers.

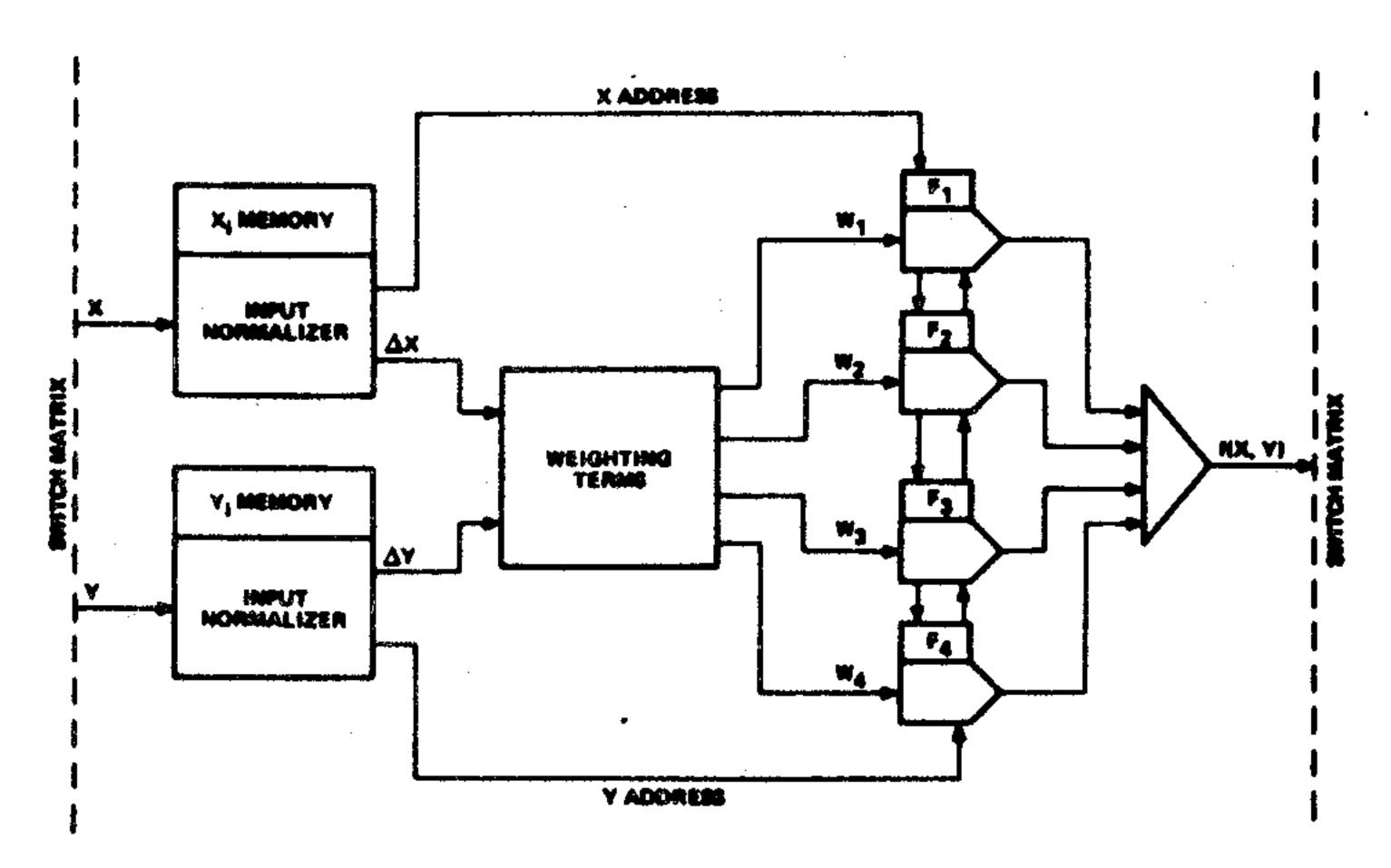


Figure 10 - F(X,Y) Block in MVFG

These elements of the MVFG transform the continuous signal into an Address and a continuous increment as defined by the breakpoint data storage X, and Y,. continuous increments,  $\triangle$  X and  $\triangle$  Y go to the Weighting Terms computing subsystem to produce the continuous signals W , - W , which feed the four Digital-to-Analog Multipliers (DAMs). Each of the DAMs has a portion of the function storage allocated to it and the appropriate cells are selected by the X address and the Y address. The four DAM outputs are summed in the final output device to produce the continuous signal f(X,Y). This signal is then fed to the switch matrix of the PMU to be distributed to the appropriate Mathematical Computing Blocks.

A complete MVFG subsystem will typically be composed of up to 6 pairs of these continuous function generation units to compute 12 f(X,Y)'s. In addition to functions of 2 variables as described here, the MVFG module can be re-configured automatically so that a full MVFG unit can provide 6 functions of three variables or 3 functions of four variables. For a function of four variables, up to 8096 words of function data storage are available. The signals which act as arguments to the MVFG can contain computing frequencies up to 1 kHz with accurate computation of the function.

For large sets of multi-variable functions with high frequency arguments, the VPFGSYS is available for SIMSTAR. This is a Vector Processor based hardware / software subsystem which is compatible with the DAP on a Common Memory interface. This subsystem is programmed in FORTRAN on the Host using a standard library supplied by EAI. The function data and control array are down-loaded into the Vector Processor by the SIMSTAR Setup program. When the run-time task is activated in the DAP, it commands the Vector Processor to produce the specified functions from instantaneous argument values produced either by the DAP program or, through the DCP, from sampled continuous signals in the Parallel Math Unit.

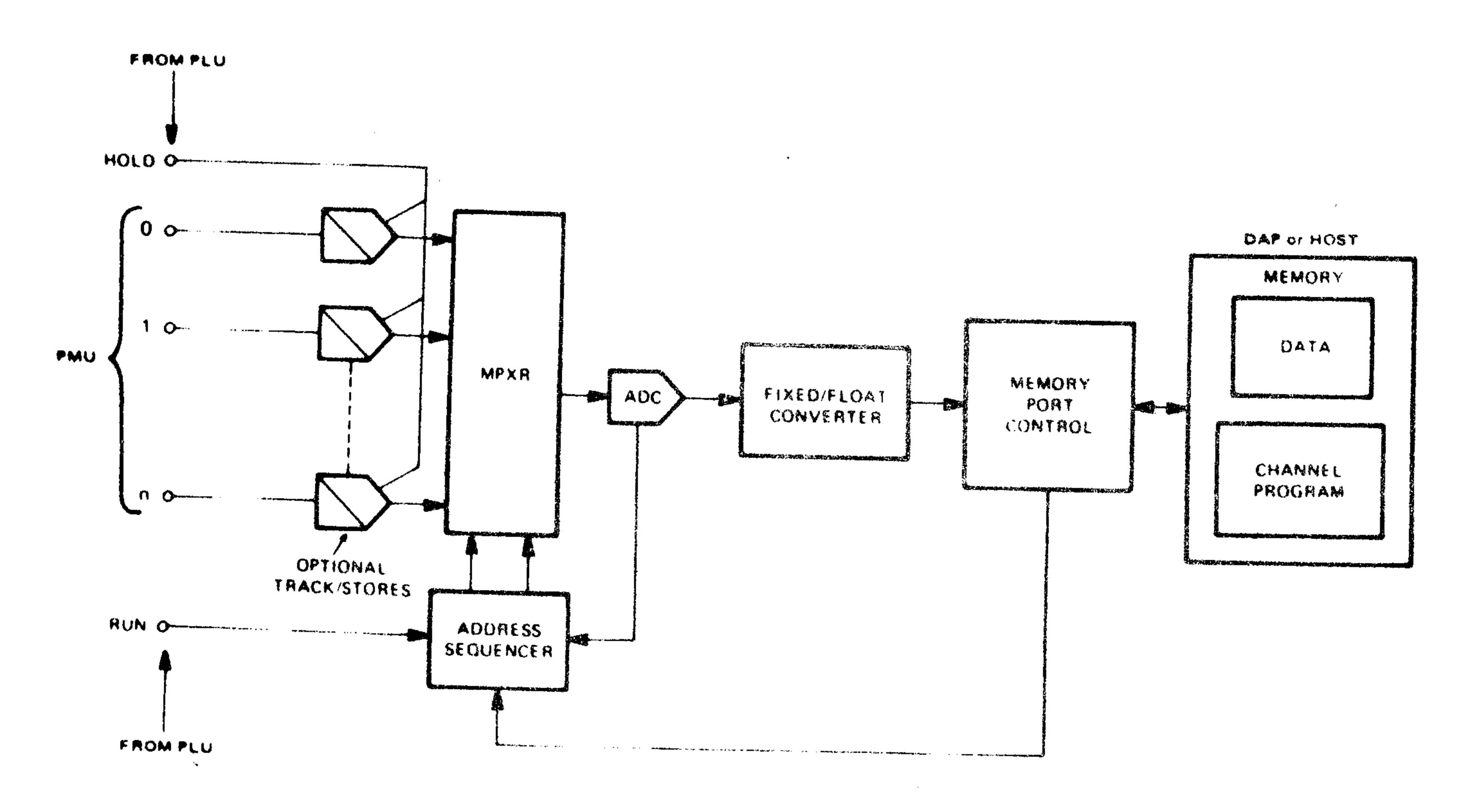


Figure 11a - A/D Signal Conversion

## DATA\_CONVERSION FROCESSOR

One of the critical communication paths in the SIMSTAR multiprocessor is between the Parallel Math Unit and the Digital Arithmetic Processor. This is performed at high speed with a minimum of digital processor overhead by the Data Conversion Processor (DCP). The use of Shared Memory between the DAP and the Host also allows the DCP to transmit data to/from the memory on the SIMBus which is directly accessible from the Host. This allows extensive use of graphic displays and recording with minimum overhead.

A simplified sketch of the A/D signal conversion subsystem is shown in Figure 11a. Note that the PLU logic program controls the conversion process. The Address Sequencer can be driven directly from the RUN line or from a Timer in the Memory Port Control.

For those applications in which all signals from the PMU must be sampled at the same instant of time, optional Track/Stores also controlled by the PLU can be added. The A/D converter output goes through the Fixed/Float Converter so that the data stored in memory is correct for FORTRAN processing. A simple FORTRAN routine prepares the Channel Program to control the sequencing of this conversion process. Sampled data can flow into memory at up to 300,000 words per second with no intervention from the DAP program.

The D/A Signal Generation, as shown in Figure 11b, is essentially a mirror image of the A/D Conversion. A Channel Program is established in the DAP memory while the Data Buffer can be in either the DAP or Host. When a conversion task is activated in the DAP, the Memory Fort Control is setup and data begins to flow from memory through the Float/Fixed Converter into specified DAM's. Again, the PLU program controls the process both to activate the RUN and to control the double buffered transfer through the D/A Multipliers (DAMs). The inputs and outputs of the DAMs are connected to the solid-state switch matrix in one or both of the PMUs.

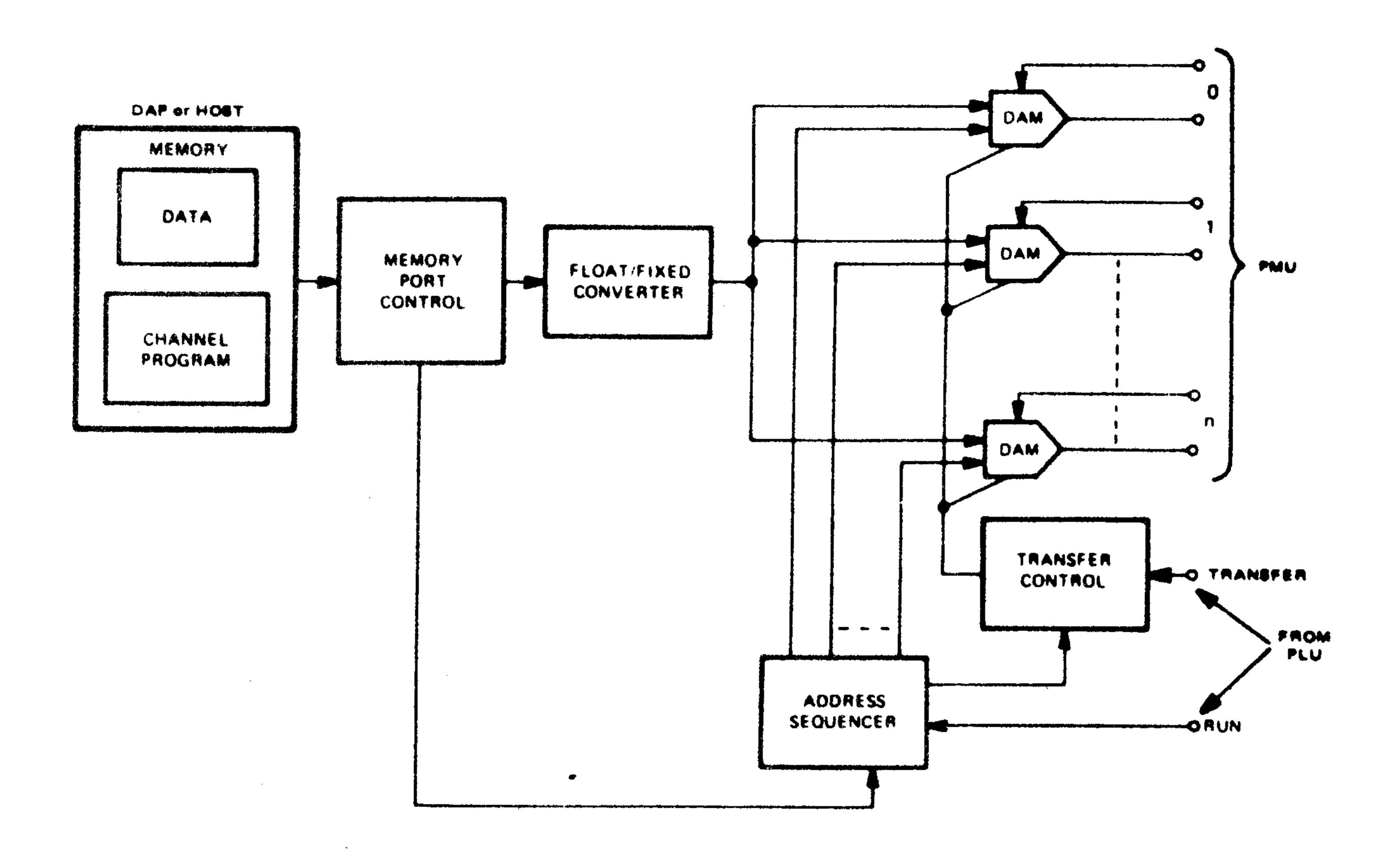


Figure 11b - D/A Signal Generation

## THE SIMSTAR PROGRAMMING SYSTEM

A flow chart of the complete SIMSTAR programming process is shown in Figure 12. First, the user starts with a problem definition. Here we assume that the requirement is to implement a new mathematical model for which existing application libraries do not exist. That is, the process is equivalent to developing a new FORTRAN program for a data processing computer. The highest level entry to the

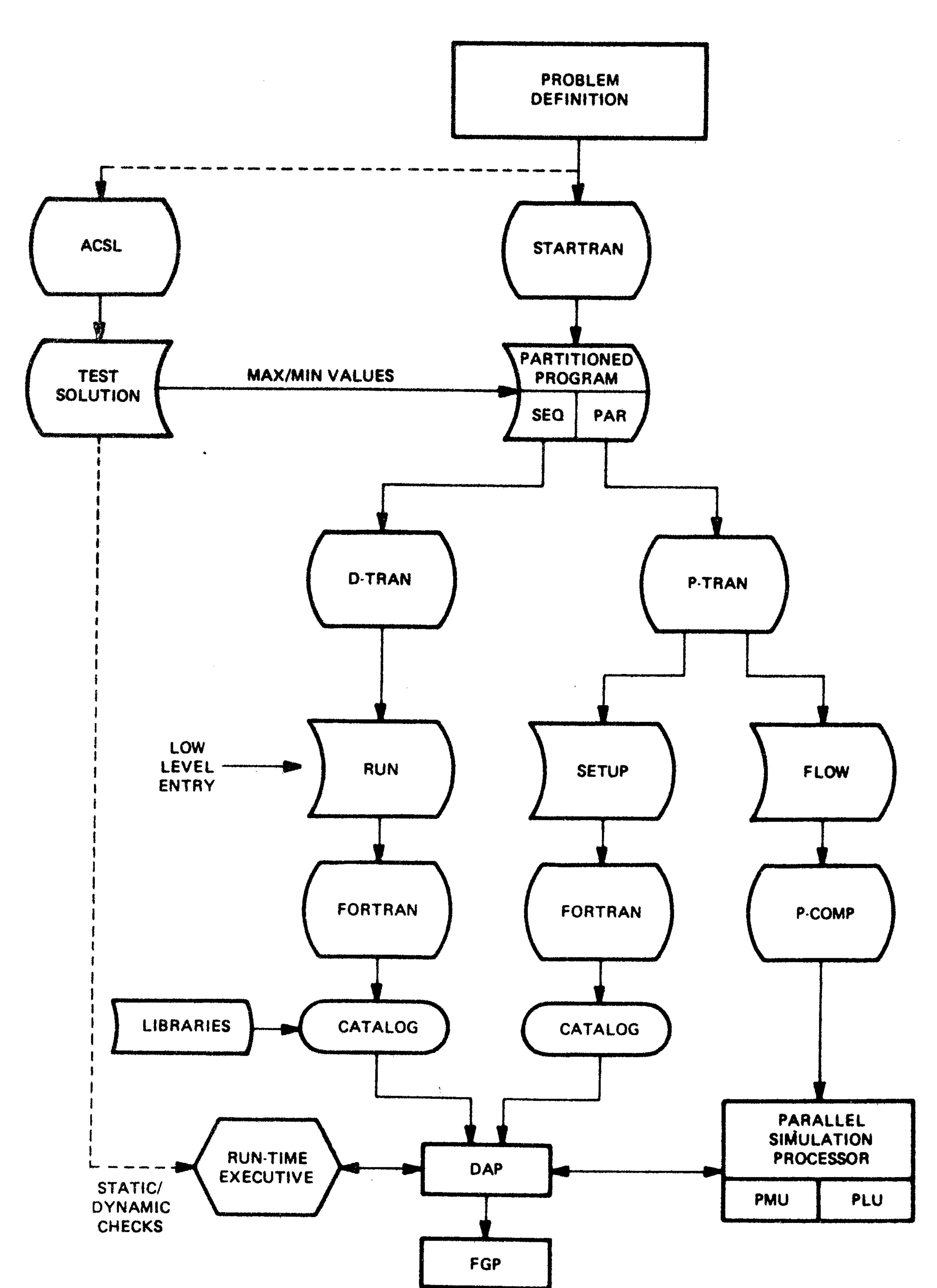


Figure 12 - SIMSTAR Programming Process

SIMSTAR programming system is the STARTRAN programming language, which is an extended version of the Continuous System Simulation Language standard (2). This language allows the user to specify differential equations in a natural form which is a superset of FORTRAN. For SIMSTAR, the user need only add a set of declarations to identify the variables to be produced on the PSP as distinguished from the DAP. The first stage of STARTRAN performs the necessary syntax analysis and partitions the problem into a sequential and a parallel part. For processing the sequential part, a high level language processor (D-TRAN) is provided to translate the differential equations into a FORTRAN program. This process also includes the generation of the control regions of the simulation for initialization and post-run processing.

The parallel region of the model is translated by the STARTRAN front-end into a set of simulation language statements for P-TRAN, which handles the reduction of these equations into a parallel processing structure for the mathematical computing blocks of SIMSTAR. The output of P-TRAN is a language for representing the connections between these mathematical blocks and logical operators which will be allocated to the Parallel Math Unit and the Parallel Logical Unit of the PSP. Also, P-TRAN produces a FORTRAN program including all of the necessary data to setup the mathematical blocks of the PSP as well as the FGP.

The combination of the RUN, SETUP, and FLOW files represent a low-level entry to the SIMSTAR system. That is, users can prepare these routines by hand and then process the files to setup and run the SIMSTAR system. For instance, in Problem-Oriented Languages, it is often easier to produce this lower level input than to try to produce the higher level for STARTRAN. Also, some efficiencies in the use of the various processors may be possible if the user optimizes the program entered at this level.

The digital RUN portion for the DAP is compiled by FORTRAN, combined with various libraries provided with SIMSTAR, and cataloged to create a Run-time task for the DAP. Also, the SETUP file is compiled by FORTRAN and cataloged using other libraries to build a separate task. The FLOW file that defines the topological connection between the mathematical and logical computing elements is processed through the P-COMP compiler to produce the necessary matrix images and binary patterns to operate on the PMU and PLU.

The user runs this combined simulation at a terminal on the Host using the "Run-Time Executive" which operates in the DAP. This executive first activates the setup task to load the Function Generation Processor, the Parallel Mathematical Unit, and the Farallel Logic Unit for the specific data of the problem. This also includes calls to routines to input the files created by P-COMP. Then, the user can ini-

tiate runs using a standard test case from the Run-time Executive to display a test solution. The test case can be defined as a part of the original problem definition for STARTRAN and run through ACSL (3) to generate a dynamic check solution. The solution may also be sliced at various points to statically check through the SIMSTAR multiprocessor solution and verify proper operation. When the solution has been checked, the user may enter run-time conditions into the executive which automatically sets up those conditions on the SIMSTAR processor. All problem information is referenced symbolically so the user need not be concerned with the actual allocation of computing blocks or memory in the various processors. A section of the Run-Time Executive also runs in the Host to generate graphical displays of results on an appropriate terminal.

#### CONCLUSION

A new attached multiprocessor for engineering simulation studies has been described. Termed SIMSTAR by Electronic Associates, Inc., the specialized system can provide computing performance in real-time that exceeds that available from any other commerical computer. Initially, SIMSTAR may be attached to any of the GOULD S.E.L. digital computers as a Host. Total automatic operation combined with the latest high density integrated circuit technology ensures reliable, deterministic performance. Programming of the various processors which make up SIMSTAR may be done either from FORTRAN or a high level Continuous System Simulation Language.

This system is designed to be expandable on a modular basis to meet the increasing computation requirements for engineering simulation for the next decade. The initial versions of SIMSTAR to be delivered in 1984 will provide for up to 200 million NOPS equivalent speed. Later system expansion will increase the performance to over 500 million NOPS. Combined with the new sophisticated graphics systems and data base management tools on Host computers, SIMSTAR should help lead the way into a new revolution in technology expected in the late 1980s and '90s.

## References

- (1) Landauer, J. P., Howe, R. M., "A Quantitative Method of Speed Comparison Between Analog/Hybrid and Digital Computers," IEEE Computer, July, 1976, pp. 31-36.
- The SCI Continuous

  System Simulation Language

  (C.S.S.L.), Simulation Vol. 9, No. 6,

  December 1967, pp 281-303.
- (3) MITCHELL AND GAUTHIER, ACSL Advanced

  Continuous Simulation Language,
  Second Edition, Mitchell and
  Gauthier, Associates, 1975, 150
  pages.

Printed in U.S.A.