A simulation system architecture for time critical realtime applications

K. L. Hall
U. S. Army Missile Command
AMSMI-RD-SS-HS, Bldg. 5400
Redstone Arsenal, Alabama 35898-5252

ABSTRACT

During the past 10 years a simulation system architecture has been evolving in the Advanced Simulation Center (ASC) at the U.S. Army Missile Command which is now providing computational speeds for large dynamic simulation applications in the range of the largest supercomputers. The present simulation system combines general purpose scientific superminicomputers with a grouping of the new SIMSTAR parallel processors. Communication between the elements of the simulation system includes conventional interfaces along with a unique software driven set of switch matrices. The software system for realtime simulation implementation avoids the common bottleneck of how to apply large parallel computational systems to practical scientific applications. A realtime user interface is being developed which combines multiple data display and control.

INTRODUCTION

Providing adequate simulation processor computational speed for Army realtime and hardware-in-the-loop missile applications has proven to be a constant challenge. The developers of mathematical models continue to push for increased detail and fidelity to represent the dynamic behavior of modern missile systems. These demands for additional speed and power from simulation processors have increased at a rate at least as fast as the rate of improved processor technology development. Thus, today, available simulation processor capabilities have the same relationship to requirements as 10 years ago.

For several years it was difficult to specify a numerical value for the magnitude of computational power required to meet the needs of the realtime simulation community. Approximations have been derived to assist in estimating processor requirements in numerical terms (1,2). These estimates are helpful for discussion of the relative performance available from candidate processor systems. As a rough estimate of the present processor requirements, aerospace applications presently active in the Advanced Simulation Center are dictating utilization of processor systems with speed capabilities in the range of 100 to 1000 million normalized operations per second to support full fidelity models in the realtime hardware-in-the-loop environment.

Increased emphasis upon productivity and minimizing missile system development time has led to increased demand for improved human interfaces and sensor stimulation subsystems. This has motivated the development of all electronic data logging, display and documentation along with more convenient man machine interaction for realtime and postprocessing control and display. Database driven scene generation systems have been developed to provide detailed target and background scenes for electronic signal injection into missile sensor systems (3).

HISTORY

During the 1960's and up until 1975, Army missile realtime and hardware-in-the-loop simulation applications depended upon either all analog or a combination of analog and early digital processors to provide the required analytical computational capability. Programming these processors was a manual process. The analog processors were programmed using the patch boards and fixed point scaled equations. Digital programming consisted of highly optimized assembly language codes. In 1972 the Army Missile Command began a development effort to provide high level language programming for processors required for time critical realtime applications. By 1975 the first generation software/hardware system was operational in the Advanced Simulation Center. This system permitted implementation of the digital portion of a hybrid simulation in the Advanced Continuous Simulation Language (ACSL) and provided a high level CSSL type syntax for the analog portion of the simulation. ACSL has continued as the language on the digital and the ACSL syntax has been implemented for the other resources utilized in present simulations.

Hardware has evolved from analog computers such as the EAI 231, EAI 781 and AD-4 with digital computers such as IBM 7094, EAI PACER 100, and CDC-6600 to the present integrated SIMSTAR system. Today the system is expanding to provide additional processing power by adding processors to the integrated system. These added processors include capability to support scene generation and data storage and display.

Human interface capabilities have consisted of CRT displays, strip chart recorders, X-Y plotters and the wide array of indicators and controls provided on hybrid computer systems. All electronic systems are being developed to provide realtime display, postprocessing, recording, and control functions.

Sensors have often received inputs directly from simple as well as complex physical models. These models have been viewed by the sensor system mounted in a three axis flight table. Some simulation/test activities continue to utilize physical models for inputs. However, an increasing percentage of the sensor inputs are being provided by mathematical models and databases constructed using computer aided design tools available commercially for mechanical drawing capture.

DISCUSSION

Factors which might be considered for high speed processor development in the long term are: total processor power and speed, accuracy, dynamic range, total cost, source code transportability, user interface quality, results management and results/implementation documentation. These factors should, ideally, be considered together in developing a processor system. It is desirable, but not usually achievable. that each factor be optimized while all other factors are optimized. The best that can usually be expected is that a few factors may be optimized while work can continue on optimizing the other factors in later versions of the system.

For realtime hardware-in-the-loop applications, power/speed requirements drive the overall system architecture to a parallel structure. For the long term, cost also tends to support a parallel approach since technology is approaching the point of diminishing returns on serial processor speed. Some of the limits on serial processor speed are closely related to reaching the limits of the laws of physics such as the speed of light and minimum spacing of objects in integrated circuits. Parallel processors have presented difficult problems in the software and quality of human interface areas. These parallel systems have been difficult to program and use.

The choice of architecture of multiprocessor systems is critical to achieving maximum available speed. A simple structure such as that shown in Figure 1 might contain many processors, P1 to PN, sharing a single communications bus. So long as minimal communication is necessary the method of interconnection is less critical and the single bus should prove adequate. However, most aerospace realtime simulation applications require high data transfer rates between processors. A common shared memory method of communication between digital processors, as shown in Figure 2, eliminates some of the interface implementation and software problems of the shared bus configuration, but speed-wise reduces to the configuration in Figure 1. To remove the limits of serial communication a configuration such as the one shown in Figure 3 might be considered. In this architecture, each processor can communicate with all other processors bi-directionally. Where as the number of possible processors is large, few actual implementations of this configuration are known to have been attempted.

Accuracy and dynamic range are usually determined by the type and implementation of the individual processors chosen. However, in the case of hybrid or analog processors, these factors are also a function of the communication quality and method.

Total cost is influenced by most of the other factors discussed. It might seem best to pick a single type processor and make all identical. If technical breakthroughs were obtainable as we wish this might be the practical approach. However, as a given technology is pushed to its limit, cost usually goes up non-linearly with achievement of increasing performance. This is especially true as a single or a set of physical limits is reached. Thus to achieve the most processor power for a given cost, a selection of processor technologies might be the best approach. This would allow each processor to be chosen without pushing the processor technology up into the very expensive region and would allow matching the technology best suited for each module implemented in a simulation. Cost is also affected by difficulty of implementation of applications. Mixing many processors, each of a different type, would present a severe programming challenge to the user unless a software system were developed which permitted the user to treat the group of processors as a single computer resource from a single high level language. For maximum benefit to the realtime simulation community the software should permit easy transfer of source code between processor sites as well as between different processor types and manufactures.

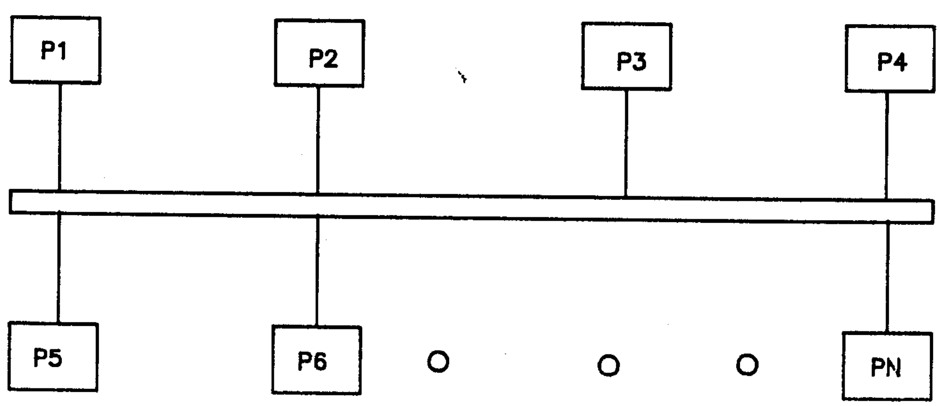

FIGURE 1 – SHARED BUS CONFIGURATION

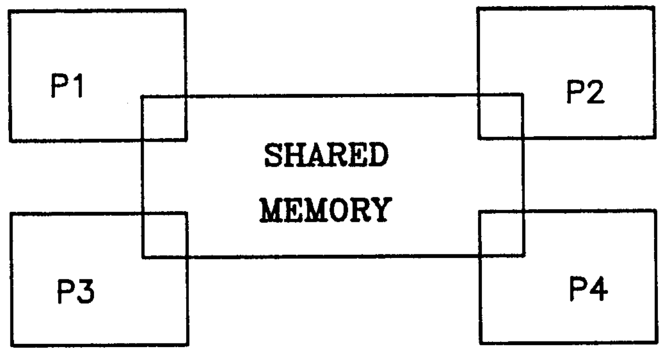

FIGURE 2 – SHARED MEMORY CONFIGURATION

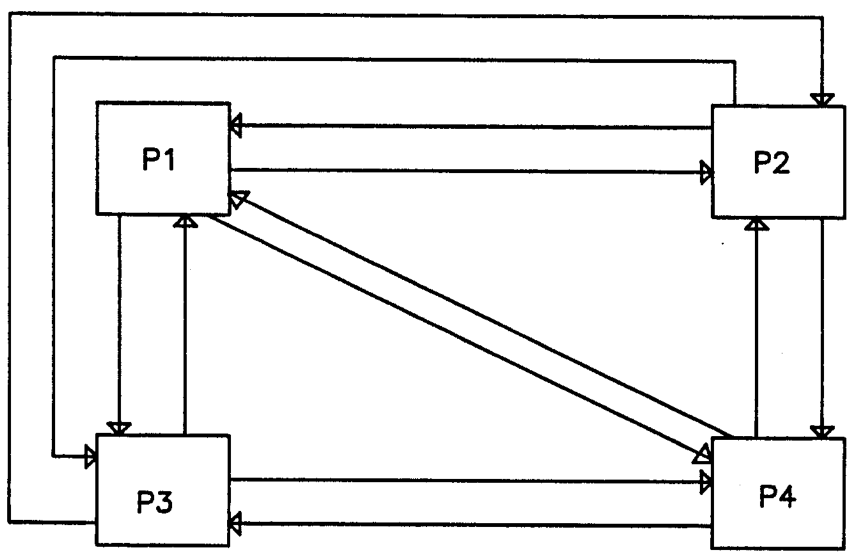

FIGURE 3 – SIMPLE PARALLEL CONFIGURATION

Evolving technology appears to permit development of a special processor system dedicated to the functions of allowing user command and control, storing realtime data, displaying realtime data, retrieving the stored data, displaying, processing the results and feeding the reduced data directly into a publication system for final documentation. A link should be possible between the realtime processor system and the publication system for complete documentation of the simulation implementation.

ASC APPROACH

Over the past 10 years the Advanced Simulation Center has assembled a mix of processor technologies with high level software directed toward the goal of providing an optimum system architecture to meet the Army's realtime, hardware-in-the-loop, missile system simulation requirements. The communications structure for one of the key elements of the system is illustrated in Figure 4. This figure shows the degree of parallelism achieved in each parallel simulation processor unit of the SIMSTAR processor system (2). The switch matrix has 320 inputs and 512 outputs. Each switch controls an analog bus which is approximately equivalent in resolution to one 14 bit parallel digital bus. This switch permits the selection of 320 source processor outputs or external input buses to be switched to any of 512 destination processor inputs or external output buses. Each processor performs one mathematical operation such as a summation/difference, switch, limit, real compare, integration, multiplication/division/square root, sin/cos or function. The configuration of the individual processors and the switch matrix is programmable and is transparent to the source code as well as the user. This processor node provides approximately 100 million normalized operation per second (MNOPS) capability for simulation suited to the accuracy and dynamic range available from SIMSTAR (2). Since a typical application in the ASC requires between 100 and 1000 MNOPS of processor power,

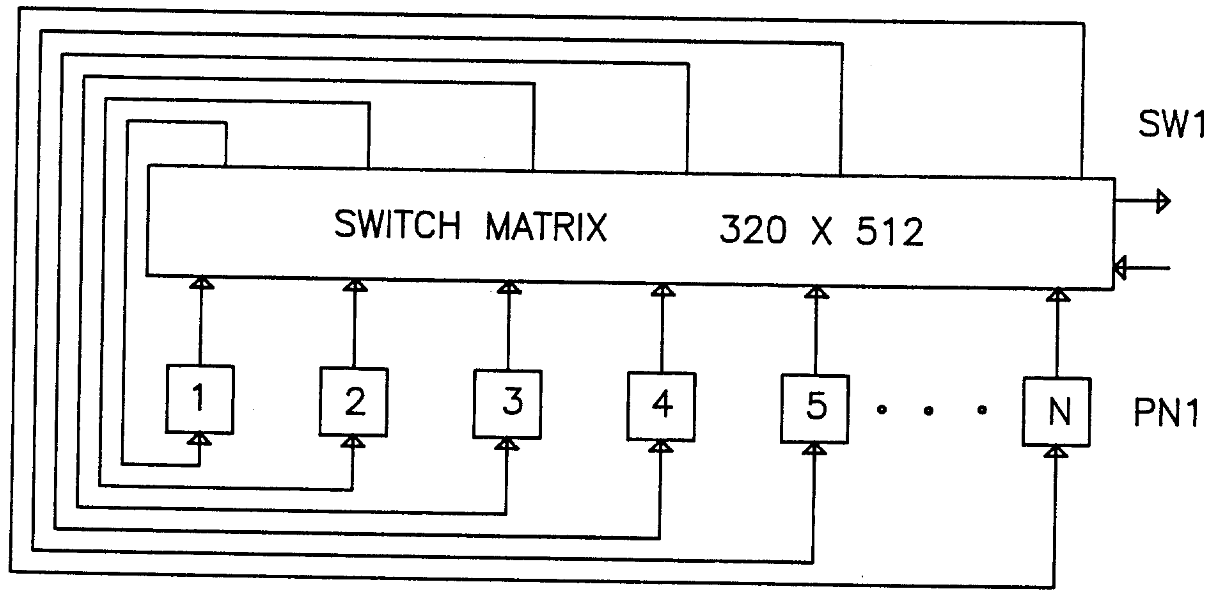

FIGURE 4 – ASC PROCESSOR NODE PN1/SW1

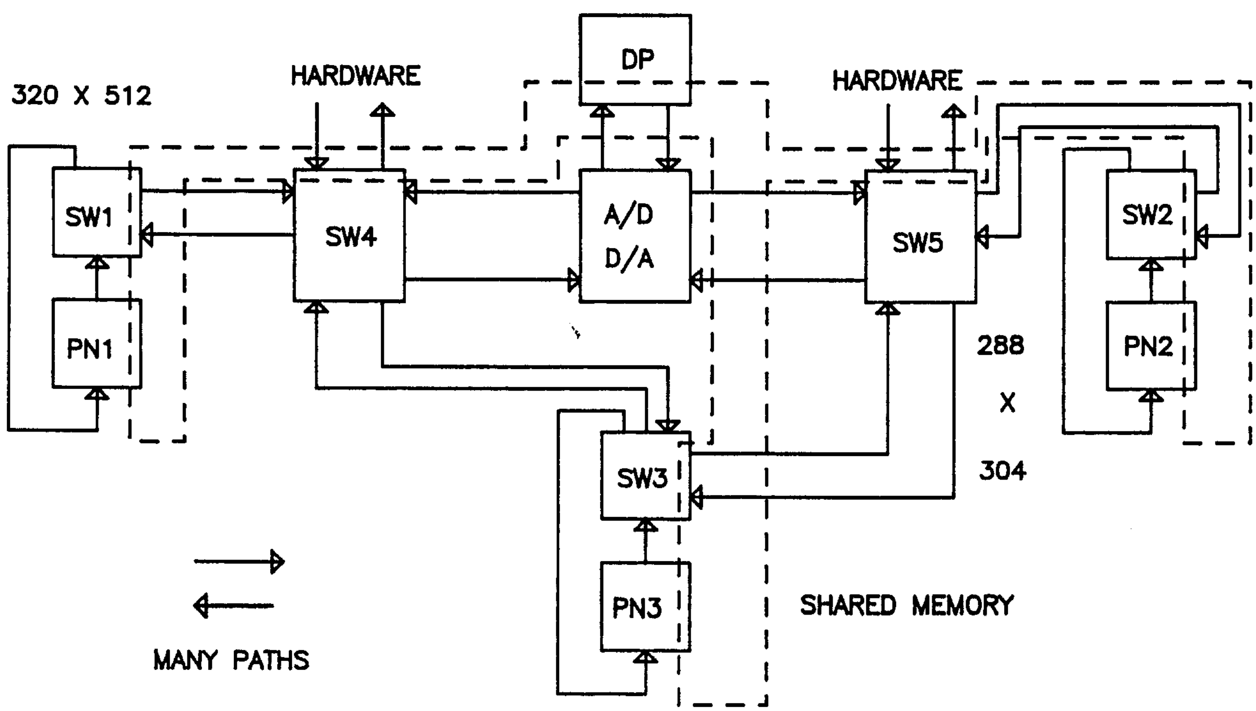

FIGURE 5 – ASC PARALLEL PROCESSOR

more than one node is usually necessary. Figure 5 shows, in a simplified form, the present processor configuration in the ASC. Three parallel simulation processors are connected to a digital processor (DP, Gould 32/8780) by two switch matrices providing approximately 300 MNOPS of processor power. Programming of the entire system is performed through the digital processor using the ASCL language syntax and the usual terminals, display and printer. The total resource set is connected by two methods. First, two switch matrices, each with 288 input buses and 304 output buses connect the three parallel simulation processors and the digital processor through the A/D and D/A converters. The second method is illustrated by the dotted line in Figure 5 and consists of shared memory with all processors mapped into the shared memory. Thus each parallel simulation processor and the interconnection switch matrices are programmed at the binary level by loading a binary image of the shared memory. The two interconnecting switch matrices provide continuous input and output ports for connecting missile hardware. Discrete, parallel and serial digital ports connect directly to the digital processor.

The realtime human interface is being developed to connect directly to shared memory and log/display data and control the realtime simulation by copying data from shared memory and writing data to shared memory.

Scene generation capability being developed connects the scene generation digital processor to the shared memory for inputs and connects the scene generator outputs directly to the sensor hardware (3).

SUMMARY

Over the last 10 years the Advanced Simulation Center at the U.S. Army Missile Command has evolved processor hardware and software for time critical realtime hardware-in-the-loop applications which provides about 300 MNOPS capability for a system cost of less than $5 million dollars. The system is programmed using a common simulation language syntax. Cooling requirements are met by normal, under-the-floor, air conditioning. Processor system power requirements are approximately 45 kilowatts.

REFERENCES

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

2. Landauer, J.P., "SIMSTAR - An Attached Multiprocessor for Dynamic System Engineering", Starlight Bulletin No. 04, July 1983.

3. Hall, K.L., "Realtime Database Driven Scene Generation for Infrared Sensor Applications", 1986 Eastern Simulation Conference Proceedings, March 1986.