1978

A digital analog of an analog computer

Michael D. Clader
Iowa State University

INFORMATION TO USERS

This material was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

5. PLEASE NOTE: Some pages may have indistinct print. Filmed as received.

Xerox University Microfilms
300 North Zeeb Road
Ann Arbor, Michigan 48106

A digital analog of an analog computer

by

Michael D. Clader

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of

The Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major: Electrical Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1978

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

This dissertation describes a digital analog of an analog computer, which is a special purpose digital machine constructed along the lines of an analog machine. It performs parallel computation and is applicable to the class of problems normally run on conventional analog equipment.

Background

Accompanying advancing technology in the computer field is a constant effort towards development of machines with greater flexibility and speed. This movement is largely concentrated in the digital area due to technological advances in design and fabrication of digital integrated circuitry. All computers have benefited to a certain degree; however, the basic concept of analog computation limits the extent to which improvements in that area can be made because of the nature of the continuous signal and the resulting system susceptibility to stray noise perturbations plus transmission line effects at high frequencies. Great improvements, however, have been made relative to hybridization of the computer system. There exists a great deal of software and hardware specifically designed to make available efficient communications between digital and analog machines[1]. This then results in the ability to use the digital circuitry of the general purpose digital computer for analog purposes. A good example of this is the development of algorithms for digital machines which perform functions such as automatic time and magnitude scaling of analog problems[2].

Among developmental concepts in the ongoing effort to improve hybrid performance of a system are patching routines which have been added to the digital system software to enable a programmer to merely specify equations to be implemented on the analog board as in APACHE[3] and PATCH[4]. This results in a component list from which one can then patch the board. A next step in development is to fabricate a device which would take the patching information and using a suitable matrix of switches, perform the interconnecting of components[5]. Such a device has not been built in large form due to the cost and problems involved in the construction of a switching matrix of sufficient flexibility to be useful for a reasonable range of applications[6]. The cost of mechanical switches coupled with the unavailability of a suitable alternative such as electronic or solid state switches are the most serious problems.

The usefulness of an automatic patching device could be great if the switching matrix could be fabricated at a reasonable cost. Foremost among the advantages is that of storing programs in a medium other than the rather expensive patch cords and patch boards and the error free patching of a set of equations. Such a device could be designed so as to not allow improper interconnections. It would, therefore, alleviate device failure because of patching errors introduced primarily by uninformed users. In addition, a high level programming language could be modified at the micro-level language as can be done in a digital machine[7]. This capability would enable rapid changes in the patching programs which would allow for a variable structure hybrid program to exist.

Such a capability would be useful in system identification routines which could be configured to iterate through system dynamics' structures. Another useful function would be to multiplex the analog ro perform error prediction and correction for multiple digital numerical programs in a multi-job system[8].

Since the component cost of this matrix currently appears to be prohibitive, other means of arriving at the goal of automatic patching have been used. Digital circuitry has been developed which has propagation delays so far below those experienced a few years ago that the practice of turning to discrete numerical methods for solving continuous problems has again taken place. A major problem which still remains is that of overcoming a lack of parallel capability in digital machines. Efforts to gain some steps here have led to the link between two mini-class computers so as to make use of the performance gained by using two central processors in parallel. This work has been reported by Korn[9].

The next step toward a completely automatic high performance analog machine is to design a machine which would contain all digital components but, be in practice, an analog machine. Such a device should have the components arranged much as they are now on an analog board with the exception that the interconnections would be handled by an array of digital components: a switching array using the latest technological developments in reliable switching circuitry. The signal levels are low as compared to analog circuitry. Noise perturbations to the system by the switch matrix would be nonexistent as would signal degradation.

A most important factor is the cost. Many simple digital switches are much lower in cost than the reed relays or FET bipolar switches which are necessary to switch analog voltages.

Some work has been done which investigates various aspects of using parallel digital hardware[9, 10, 11]. This does not address all of the aspects of hybrid operation, but does clearly demonstrate the feasibility of the general concept. The next logical hardware step is to deal with the entire system.

This dissertation will deal with a proposed design of such a machine composed of a large array of very small digital arithmetic units which can function as a system like an analog machine.

Statement of the Problem

The objectives of this dissertation are twofold: the conceptual creation of a parallel digital processor, which is an analog of an analog computer, and an examination of the feasibility of such a system in terms of cost and performance.

Such a machine must have computation capability at least equal to existing analog machines. Practically, it must be configured much the same as present analog machines with some variances due to its construction and features. Such a machine should have some features which offer improvement over present analog computers such as automatic patching, high quality user definable functions, greater dynamic range of simulation, and less maintenance.

The preliminary feasibility of this machine is tested by formulating a machine design and then making a component by component

performance and cost comparison. Then, a further comparison is made by taking example problems and examining the timing and making performance comparisons with the same problems as patched on an analog machine. Finally, performance comparison is made with large scale general purpose digital simulation.

Further examination into the feasibility of the machine is made by commenting on the various areas of application in which this machine appears to be advantageous. This is extended by addressing some of the economic aspects of the machine in the context of the complete cost of ownership.

Review of Literature

The existing literature which relates to the concept of new and more flexible parallel analog and digital computation deals primarily, but not exclusively, with automatic patching and parallel digital mini-computers and their applications.

Automatic patching systems have been developed and two papers describe the state of this development.

A paper which is somewhat tutorial is "Automatic Patching for Analog and Hybrid Computers"[6]. In it, Mr. Hannauer describes the modularization of a large analog computer for the purpose of eliminating the large number of switches required if it is required that all interconnections on the patch board be made possible. While doing this, he shows how program flexibility can be maintained for a great many problems. He concludes with the description of the construction of a prototype unit for a portion of the EAI 681 analog machine.

Reported in "A Practical Automatic Patching System for a Time-Shared Hybrid Computer"[5], by Shoup and Adams, is the construction of another hardware patching unit for the EAI 681 analog processor. They modularize the machine into segments containing what they have observed to be a nominal level of high density patching. Designed so as to eliminate the inordinately large number of interconnections necessary for the entire machine, these modules provide the patching flexibility necessary for most applications. Included is an example problem and how this device patches it. In concept, they develop the same hardware as Mr. Hannauer described above. They do, however, go a bit farther with some data for support which points out the optimal size for one of these modules.

The following papers address the concept of parallel digital processors and their use in simulation tasks.

An interesting treatment of the use of digital components for parallel analog type use is found in "Parallel Digital Differential Analyzer with Arbitrary Stored Interconnections" by Dawoud and El-Araby[10]. This paper presents a differential analyzer using an associative memory as the elements of the component array of integrators. These are updated at each time cycle and outputs are routed according to the equation being simulated and resulting pathways in the interconnections.

A summary paper by S. E. Scrupski, "Coming: Cheap, Powerful Computer Electronics"[12], outlines current industrial practices in the area of parallel digital simulation. He treats activity at Grumman

Aerospace, IMS Associates, and Realisations Etudes Electroniques (France) which is focused on parallel micro-processors used in simulation. The difficulties lie primarily with devising suitable algorithms which can share tasks among the elements on a real-time basis. Further description of the work at Grumman is found in the article, "Micros can beat the IBM 370/168"[13].

An interesting presentation of this concept is made by A. V. Kalyayev in "Homogeneous Digital Analogous Structures with Programmable Commutation"[11]. In this paper, the discussion centers around the integration process and the device to device communications. The majority of his discussion centers upon a decision tree algorithm which would perform the interconnections. His idea is one of real-time reconfiguration of the array as program solution progresses. The array structure and its elements are programmable which would lend itself to micro-processor design.

The parallel processor system described by Korn in "Back to Parallel Computation: Proposal for a Completely New On-Line Simulation using Standard Mini-Computers for Low Cost Multiprogramming"[9] is composed of two PDP-11 mini-computers. This system produces a digital simulation of differential equations employing the parallelism of two machines to achieve the high speed operation. The primary design goal of the system was low cost. These machines, without peripherals, can be purchased for a very low cost in comparison to a parallel analog processor. The basic mode of operation is to share the computation responsibilities of the simulation between the two processors.

Although not digital, the equipment described by Wahlstrom and Juslin[14] follows a design concept similar to that of the parallel hardware functions. It is, in concept, a non-flexible function which is less expensive than the corresponding highly flexible analog machine implementation. Unfortunately, this lack of flexibility becomes quite expensive as fabrication costs are quite high for high reliability analog equipment. An extension of their work would be more viable if one were to consider the use of numerous second order or other units which could serve to be macro components for an analog system.

If automatic patching were to exist, software would need to be developed to properly utilize the capabilities. A small amount of software does exist which is applicable in this area, but it represents a small fraction of that needed.

The information presented by Green et al in "APACHE - A Breakthrough in Analog Computing"[3] is a method for relieving the operator of the tedium of devising an analog program given a set of equations to be simulated. This is taken as an input to another algorithm by East in "A Completely Automatic Hybrid Computer"[15] which produces a set of outputs which could drive an automatic patching device such as described by Hannauer[6] or Shoup and Adams[5].

Rigas and Coombs report, in "PATCH: Analog Computer Patching from a Digital Simulation Language"[4] the development of software which would produce a patching list for the CSMP as used on the IBM System 360. They describe the patching and scaling algorithms with some examples

of their implementation.

Further work described in "Interactive Simulation Language for Hybrid Computers", by Benham and Taylor[7], concerns itself with a high level language for doing hybrid simulation. One of their goals was to produce software which would allow the treating of the analog system in a manner similar to the digital. They enumerate the various advantages of such a capability.

"Program Generation System for Modern Hybrid Computers" by Landauer[1] also addresses the high level language concept. He shows an example compiler algorithm which allows the user, from a high level language, to set up virtually all of a hybrid simulation.

This last group of papers deals with some conceptual ideas which revolve around the availability of a highly flexible analog processor.

The tutorial paper by Cannon, "Magnitude and Time Scaling of State Variable Equations for Analog/Hybrid Computation"[2], presents an algorithm for time and magnitude scaling of an analog simulation. This is an algorithm useable by a digital machine as a generatable program which scales the analog simulation. Presented is a generalized routine which can be easily specified to a particular problem.

Another application which could prove useful in a number of applications of a flexible hybrid computer is presented by Karplus and Russell in "Increasing Digital Computer Efficiency with the Aid of Error-Correcting Analog Subroutines"[8]. This paper describes the use of a simple analog circuit to provide a high speed low accuracy computation for systems of linear algebraic equations. Used is a digital

iterative scheme that greatly increases overall system speed by relieving the digital processor of the tedious matrix operations. The extension of this is to apply the idea to problems of higher complexity involving the solution of differential equations on analog machines.

The paper by Rubin and Keene, "The Future of Hybrid Computation"[16], presents the economic tradeoffs of hybrid as compared to totally digital computation. The results are favorable to hybrid systems when they are used in a time-shared mode. They point out here the advantages of automatic patching and size if these were possible. Mention is also made of using an analog machine as a peripheral device to a host machine. The net result of this report is that an analog machine used in a hands-off mode as a peripheral device would greatly increase the overall system capability of a scientific digital machine.

The above literature outlines a general trend towards hybrid simulation capability which is highly flexible. Much work is being done to make analog interactions cheaper, faster and easier; these to meet the growing requirements for hybrid computation services. With the advent of highly sophisticated digital circuitry, the trend is towards its use in this area to pick up where the analog machine leaves off. Major rethinking is required and, as described by Crosbie[17] at the IMACS Congress, the entirety of the economic questions must be addressed. Savings in time or money must be carefully analyzed, as they sometimes become hidden from view in particular situations. So, as digital alternatives become more viable, they must be weighed against the analog in not only performance but cost of total system ownership.

As the above references address the general area of analog computers, the following papers present a general concept of computation within which the proposed machine of this paper is contained. This concept, referred to as data-flow processing, is basically a method of topological description of computation systems which specifically lends itself to describing parallel computation. It is used for either hardware or software and affords one the ability to analyze highly parallel and ordinarily complex systems in a straightforward fashion.

The paper presented by Miller[18], "A Comparison of Some Theoretical Models of Parallel Computation", describes the beginnings of the Petri-Net, which is a method of describing parallelism or concurrency of operation. He then includes the E-Net and a parallel schema which extends the theory toward more practical consideration.

E-Nets are described in detail by Noe and Nutt[19] in "Macro E-Nets for Representation of Parallel Systems". They outline and describe the theory of E-Nets as a method for analyzing the parallel computation scheme. They show how they can describe deadlocks and faults just as real systems have them. This renders them capable of being used in design of systems.

Further theory is developed by Dennis and Misunas[20] in "Preliminary Architecture for a Basic Data-Flow Processor". They present the theory of the data-flow machine. They also present a candidate system which is described by the data-flow theory. Their example is a multiple processor machine with software which determines data-flow paths, at execution time.

Further application of the concept is made by Wulf and Bell in "C.mmp-A Multi-Mini-Processor"[21]. They present the scheme with parallel PDP-11 computers and develop the theoretical concept.

The hardware concept of a host digital eith differential analyzer peripheral to it is presented by Gilbert and Morse in "Digital Simulator Replaces Analog Portion of Hybrid Computer"[22]. They present the concurrency in a single processor as a digital differential analyzer hosted by a NOVA mini-computer. Their approach is to replace the analog machine with digital technology. It is limited to a single processor but does show the concept of the implementation of digital circuitry for previously analog signal processing.

A higher degree of parallelism, more in line with the data-flow concept, is presented by Arnold and Page in "A Hierarchical Restructurable Multi-Microprocessor Architecture"[23]. They show a system composed of bit slice micro-processors in an array connected by a single data bus. Control is effected by a separate bus utilizing a circular bus technology.

It is within the data-flow processor concept that there is application to the analog and hybrid computation area. This concept, when coupled with a specific implementation, can address the various shortcomings of analog systems as well as extend the digital data-flow scheme into that general area. The machine proposed in this paper allows for parallel control and high concurrency of data manipulation. Software governs the data routing while data transmission and operation are hard wired resulting in the time critical aspects to be hardware

executed. This allows for very high data rate capability with flow alteration from a completely independent structure.

MACHINE DESIGN

The research machine described herein is a digital analog of an analog computer which is designed about a "highly parallel" structure. Parallel computation is achieved with very small special purpose arithmetic units interconnected by an externally controlled bus oriented communications array. In concept, it is a direct parallel to an analog processor and, therefore, an analog of it.

First, the machine macro structure will be discussed, followed by presentations on the individual computing elements addressing both performance and cost.

In addition to individual components, a brief discussion is made of the concepts involved with interfacing this machine with the digital host and with the external world. No specific design is presented.

Last is a design and discussion of the switch array. This is a critical portion of the total system from the hardware performance standpoint. A hardware test was conducted to demonstrate the feasibility of multiplexing data at the rates required in the switch array. This is described in the Hardware Testing Results section.

The machine concepts presented, in the judgment of the author, represent an original, significant contribution to the evolving field of analog computation.

Basic Machine Architecture

The basic architecture is established for super fast computation: parallelism. Beyond that, however, the problem begins to lose its definition as the multitude of tradeoffs enter the picture. There are

a great many concepts to follow in parallel architecture design. The data-flow and control-flow perspectives are two major divisions which encompass most designs. One can design to optimize either or include both depending upon various constraints.

In choosing the parallel architecture which applies best to this area, there are several issues which must be considered. These include hardware complexity, software complexity, and method of control. The hardware complexity issue revolves around difficulty of construction, maintenance, and modification. Software complexity is essentially defined by ease of use. The control method issue encompasses the style of the structure, its relation to data manipulation, and system throughput sensitivity to that relation. Consideration of these is generally sufficient in system design.

Various outside constraints will drive decisions in these areas. In the software area, manpower cost and level of competence are at issue. Production capability dictates hardware. Control schemes are impacted mostly by application.

In this proposed machine, the following goals were chosen: Highly parallel arithmetic operation, simplified programming, hardware control at execution time, and software control elsewhere. This set resulted in the architecture presented. The software is straightforward and interfaces well with new developments. Avoided are the difficult partitioning problems associated with other parallel designs. The hardware is modular allowing straightforward construction and modification. Control software only alters data-flow paths which are

fixed during execution time. Further, at execution time all arithmetic activity as well as data transmission are hardware functions.

Conceptually, the proposed machine is designed as shown in Figure 1 with the major components being the array of the computing components, the switch array, the structure for providing input and output signals (digital-to-analog, digital-to-digital, and analog-to-digital), a section for performing the mapping function between a patching diagram and the switch array, and the section dedicated to the control of these various sections.

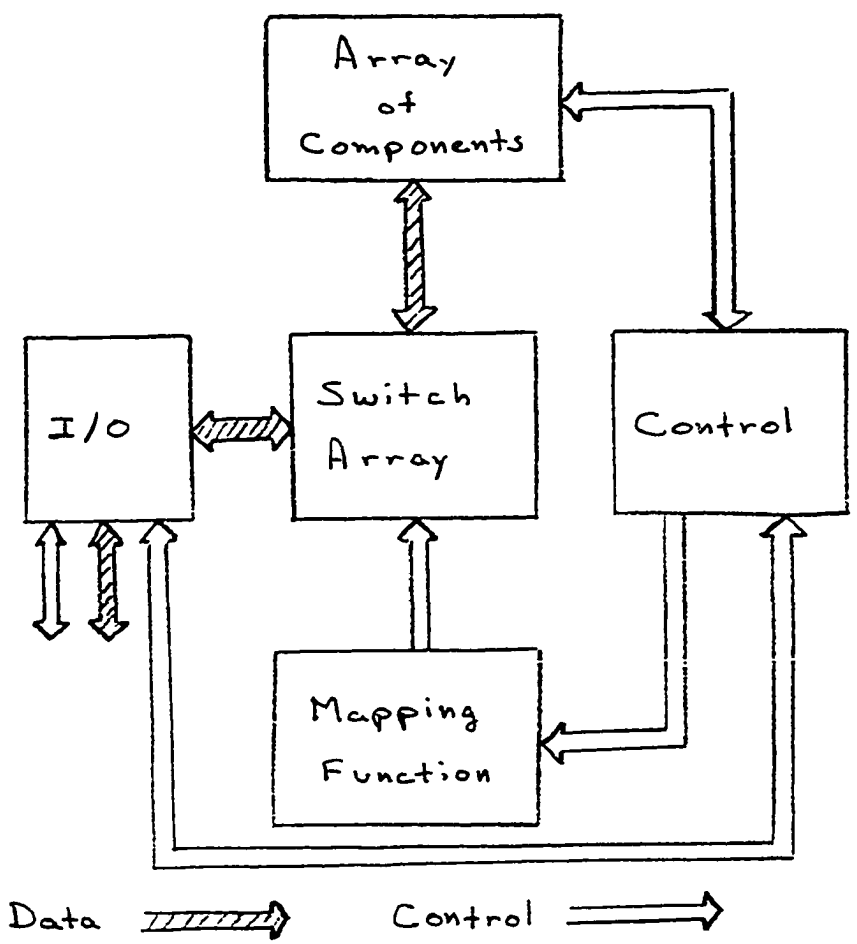


Figure 1 Digital Machine

The array of components is mated with the switch array to provide for minimal wiring and other cost saving considerations. This array is arranged similar in concept to the arrangement of components on a real analog machine, but in a configuration which more readily permits operation from a switch array. These components employ scaled binary 2's complement arithmetic. The word size chosen is 16 bits, however, this could be easily modified. The choices of 16 bits and 2's complement, however, more easily permits interfacing of this machine to any of the large number of mini-computers now on the market which also use the same word size. These types of computers are often used in the hybrid applications, although machines with other characteristics are also found.

The switch array is arranged in a modular fashion much as was done by Hannauer[6] and Shoup and Adams[5]. This results in a sufficiently flexible capability to patch problems while avoiding the burden of allowing every possible connection at all times. As an example, a switch array which would allow any component to be connected to any other component on the EAI 681 analog computer would require many tens of thousands of switches, a number which would not be reasonable. The use of low level logic switches would decrease the burden presented by this large number of analog switches for an analog machine, and, further, the number of interconnections which need be allowed does not approach that magnitude. In addition, a module uses multiplexed busses to further reduce the number of switches and bus lines required.

The mapping function section is linked to the switch arrangement and design, since the mapping function which it provides is also a function of the switch array arrangement. This section provides a mapping which, when given inputs from the control section describing the interconnections (i.e. amplifier 1 to integrator 2 through potentiometer 4), effects these connections. It executes a mapping function which is designed specific to the component layout, by selecting the components necessary while taking into consideration space and improper patching. The mapping function itself is fairly straightforward, given input and output, and the mathematical algorithms for its implementation exists. In a way consistent with the idea of high level languages as referred to in the introduction, the mapping function is modifiable at any time by use of external program loading. This allows for more efficient patching of various types of programs. For instance, if the system to be patched consisted of an array of second order equations, the mapping function could be programmed to respond to arrays of these elements rather than the usual elements composed of single components. Thus its time would be spent in formulating the array of macro elements rather than constructing the second order equations themselves. If the system of equations were not at array of second order but some other more complex set, the mapping function is changeable. The implication here is that of using different algorithms for interpreters with the capability to take the same set of information and perform the patching in different manners.

The heart of the machine, which consists of these three components, requires several structural iterations before an optimal design would be found. However, the technical procedures and implementation are straightforward. The use of low level digital logic provides the situation in which the cost of adding another one thousand switches, for instance, does not increase the cost significantly. As a result, optimal design is not necessarily one of minimal components but one of providing the most flexibility for a given cost.

The control section is one which interacts with the component array as does the switch array, only for control purposes. This emulates the operate, initial condition, hold, and time scale switches existing on present analog computers. Its control of the array of components includes time scaling and multiplication factor changing, etc., much as present analog machines use the time scale switches or control pins on the patch boards to switch capacitors or integrators. The control of the mapping function consists of transmitting data pertaining to interconnections to the required block. It is this control section which is directly manipulated by the external digital machine using some type of software programs such as PATCH[4] or APACHE[3] patching diagram programs. Much like the changing of the mapping function, the control section can be changed using read only memories or programmable read only memories to effect the change in the controlling scheme used for either the mapping function, the switch array, or the component array. This provides, then, a second area of micro-language manipulation of the system. The input-output structure block is straightforward. It

The input-output structure block is straightforward, It consists of the necessary digital-to-analog and analog-to-digital converters to interface the machine to the external world in terms of data. The data coming from the external devices through this I/O structure into the switch array are routed to the various components by means of that switch array as modified by the mapping function.

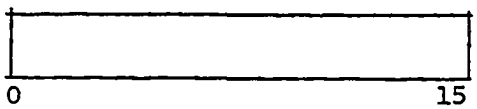
Control Unit

The control unit is the executive for the entire machine. It performs mode control as well as channeling of configuration control and host machine interrupt information.

A 16 bit word was chosen so as to maintain compatibility with mini-class digital machines. Further discussion about data word size is found in the Component Design and Performance Comparison section. Due to the addressing requirements, an entire word is needed for patching. Patching involves both addressing of switch matrix registers as well as constant registers and the 16 bit constants themselves. Further discussion of the details of the patching function and the bit fields of the patch words may be found in the Switch Array section.

Mode control requires one word also. In addition to the control of the modes, this word enables interrupts and specifies the addresses of the channels from which interrupts will issue. This word is decoded as follows:

Table 1 Control Word Format



BIT #	
0-2	MODE 111 Patch/Constant Set 110 Time Scale 100 Run, Hold, Reset
3-4	00 Reset 01 Hold 10 Hold 11 Run
5	Enable bits 6-7 into interrupt register
6-7	00 Turn All Interrupts Off 01 Interrupt on RESET 10 Interrupt on HOLD 11 Enable interrupt from specified channel (bits 8-12)
8-12	Interrupt address channel
13-15	Time Scale Mode 000 One Times Standard 001 Two Times Standard 010 Four Times Standard

The interrupt system allows signals to be transmitted for the purpose of interrupting host processing by any of three pathways. The first two are the occurrences of the RESET and HOLD modes. The third, via one of a set of channels, is connected to the comparators or parallel logic system through the patch array. This allows for interrupts to be generated by various conditions which are determined in the software.

When in the PATCH mode, the control of the machine is determined by the state of a patch/no-patch flip-flop. The state is changed by the first set word when the patch mode is commanded. It returns control to the control section when the patch command containing all ones is processed. At this point, all future instructions are interpreted as control until another command is given to resume in the PATCH mode.

Figure 2 shows the structure of this control section.

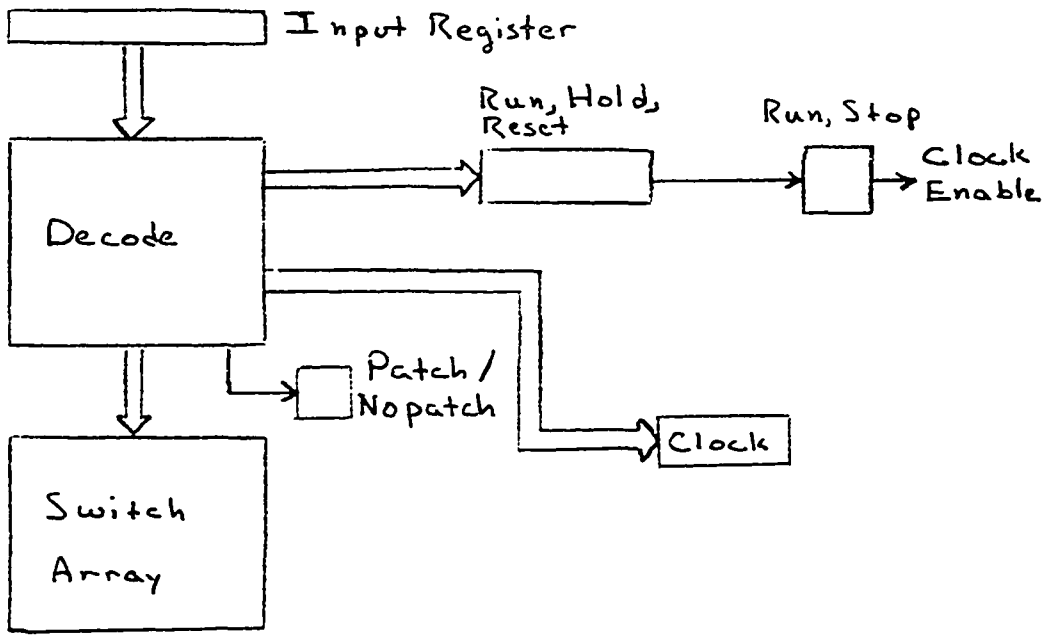


Figure 2 Control Section

The word format used for patching is outlined in the Switch Array section discussion.

Component Design and Performance Comparisons

The feasibility of this digital analog of an analog machine depends upon the performance capabilities of the individual computing components. For a digital simulation, and especially for real-time problems, there must be performance at least equivalent in individual components to an analog machine or a simulation is not reasonable to consider. The following discussions use performance measurements of components contained in the EAI 681 analog computer and compare these with designed capabilities of the digital machine. They will cover only the major components. Those considered are the integrator, summer, multiplier, function generator, attenuator, and a few miscellaneous units.

A design note must be made here which applies to all the components discussed below. A data word of 16 bits was chosen primarily for two reasons. The first is to facilitate data transfer with a mini-computer as well as a pseudo 32 bit machine. The second is the accuracy and resolution required in the computation. A quantified requirement of universal application is difficult to arrive at, however, an estimate can be made. For present analog computation, the four significant decimal digits are not only the limit of the system but generally adequate for the work to be done. This is exceeded in this machine. The resolution here (one in 2^{16}) is sufficient for the

class of problems which are typically addressed in simulation and, for the exceptions which are few, it is more reasonable to employ host machine computation and only at the point in the simulation that it is required. The performance limits imposed by a larger word are generally not compensated for by increased resolution.

Although Schottky technology devices are used in the design of this machine, a second technology is also useable. This technology is emitter coupled logic, which compares favorably with Schottky in performance.

Aside from a slightly lower gate propagation delay for ECL. there are other aspects which are to be considered. Emitter coupled logic has about a 50% greater power consumption. This remains relatively constant with frequency while power is a function of frequency in the Schottky devices. This increase is due primarily to overlap switching of the output transistors. Schottky and ECL power consumption cross at about 32 MHz. Schottky is greater than ECL beyond that frequency.

A second aspect of operation is that regarding transmission line driving capability. Emitter coupled logic is capable of driving lower impedance lines than is Schottky. Also, the difference in propagation delay in long lines increases with length giving a greater advantage to ECL with increasing frequency.

Careful design with regard to transmission, power supply, and ground lines are necessary in the use of ECL devices. These present problems due to very short rise times on the logic signals. In a few places, this technology is useful in this proposed machine as well as common TTL in other areas. Inclusion of these remains for future work.

Integrating Unit

The first element examined is the integrator which, in analog form, is composed of a high gain amplifier with resistor input and capacitor feedback plus the necessary compensation circuitry for voltage holding and time scaling.

EAI factory specifications for this device show a 0.5 degree phase shift at 1 kHz which corresponds to a 1.4 microsecond delay of a sinusoid of that frequency. Of course this delay does not exist independent of frequency, but it is a representative number for showing the time delay performance of the device. The use of a digital adder to perform successive additions for integration results in a 16 bit add delay of 50 nanoseconds when employing the zero-level carry-lookahead technique and using Schottky type devices. By adding only a few nanoseconds of delay for register shifting, time and magnitude scaling can be implemented. With only this 50 nanosecond delay, it is possible to use a digital word larger than 16 bits in order to increase the number of significant digits (and resolution) and still remain within performance constraints if such a requirement were to exist in the future.

Figure 3 shows the conceptual design of this device showing registers and shift points required for control and operation.

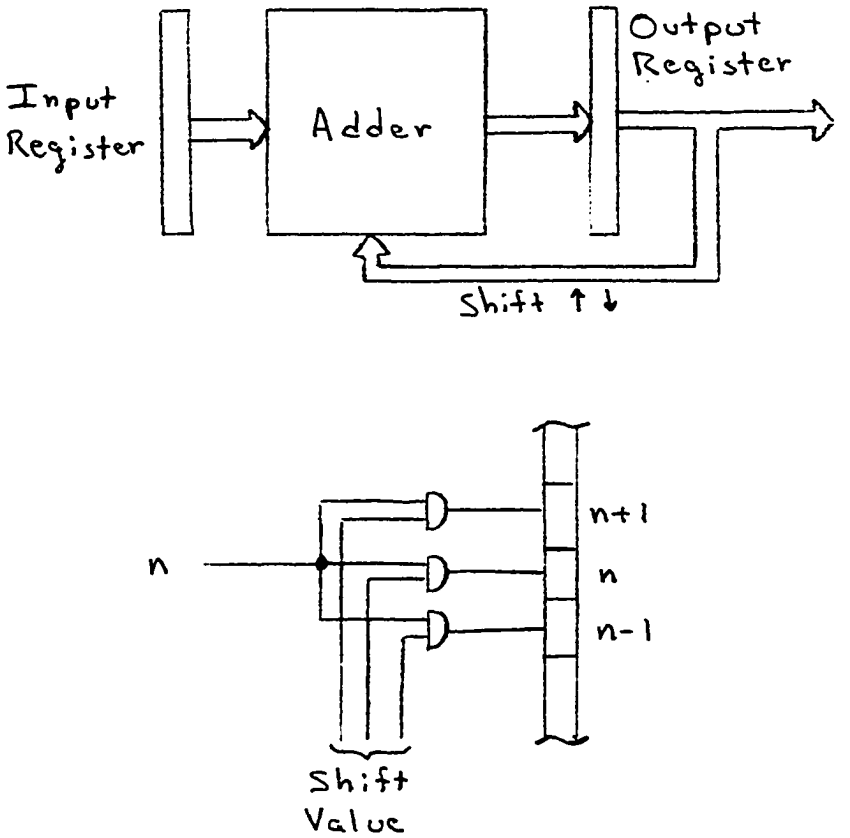


Figure 3 Integrating Unit

Timing is incorporated by using a standardized or normalized clock throughout the entire machine and the clock ticks will be used to initate the add cycle. The adder is an asynchronous device and must have clock controlled gates on the inputs or outputs as would the other devices so that proper synchronization can be maintained among devices. As indicated in Figure 3, the gain or time scale shifts may be controlled by a constant shift number being applied to gates for the input register.

An integrate rate of 1 MHz has been chosen so as to allow signals with a bandwidth up to 10 kHz without severe quantization errors.

Generally, present applications do not require a bandwidth above this but, if they would, time scaling is certainly applicable. For ease of implementation, the clock interval is a power of two. Further, it must be equal to or less than the 1.4 microsecond interval of the analog device. In order to use a clock frequency allowing additions at the rate of one every 1.4 microseconds or less, the following modifications are made to the basic design. A twenty bit adder is added to the most significant bit of the 16 bit adder to accumulate the carries generated from it as shown in Figure 4. This allows a full scale input to

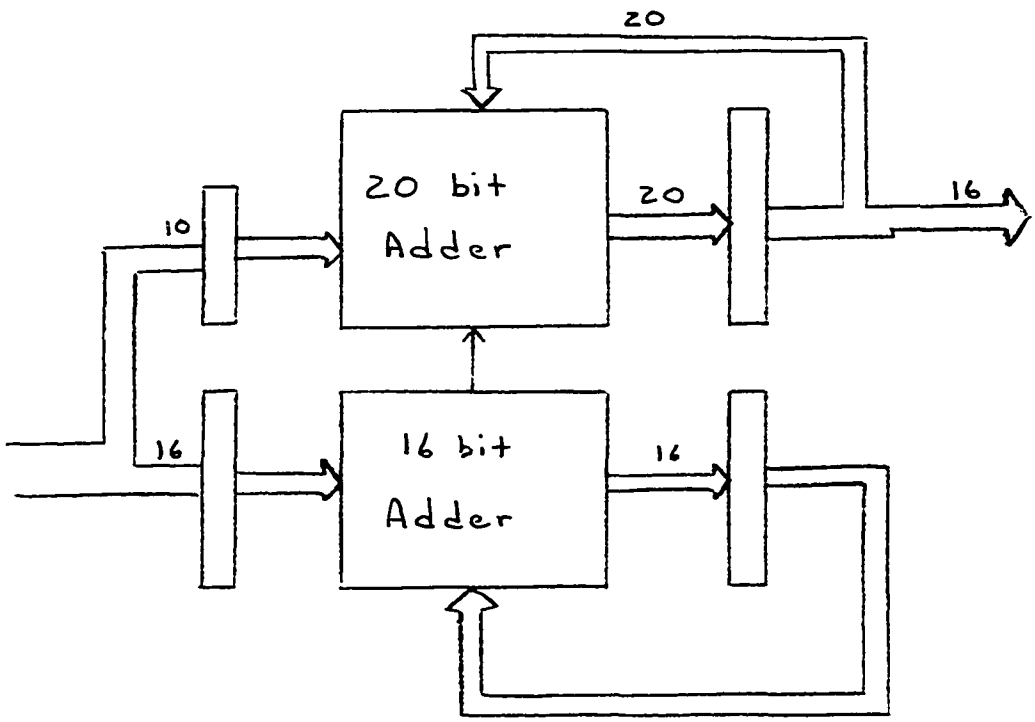


Figure 4 Expanded Integrator

accumulate to a full scale output in 2^{20} counts. For this to occur in one second, results in a 0.954 microsecond clock rate, This

modification allows for the inclusion of the delta t in the recursive

$$\sum f(t)\,\Delta t \approx \int f(t)\,dt$$

equation. Delta t in this design is $1/2^{20}$. The division by 2^{20} is accomplished by using the most significant 16 bits of the 36 bit word as the output. The shifting on the input is still possible and with 954 nanoseconds available, internal register shifting is feasable with clock subcycles or other timing pulses. This method, while requiring time, allows for a savings of a number of components for each integrator. Final decision on which method is more useful is left for future work and must consider the various constraints presented by hardware implementation.

With the addition of these ten bits to the input register, a 2^{10} factor on the input plus up to four times the clock frequency for synchronization results in a total gain/time scale factor of 2^{12} or 4096. This is higher also considering the use of gains on the attenuating units greater than one.

An important feature of this device, and a resulting advantage over its analog counterpart, is its ability to hold a value when in the HOLD mode. This is simply achieved by disabling the clock, at which point the output register, containing the last value, receives no further input strobes and does not change value. The importance of this lies in the necessity of a holding capability of the integrator over a long period of time. Analog devices have compensations built in for this purpose, but the voltage holding is still accomplished by

using a capacitor charged to a desired value. At best, this holding ability does not last more than a few minutes before the signal begins to noticeably deteriorate. This time is a function of the capacitor chosen for time scaling and its leakage current which is itself a function of age and temperature.

Another feature of not quite as great importance is that of a rapid initial condition set. In high speed repetitive operation, this becomes a problem[24] in that the analog device has a capacitor which must be charged to the initial value. Depending upon the time scale chosen, and thus the capacitor value, the time required for charging the capacitor varies. The programmer must program around this lag in order to keep the simulation resetting properly. The digital device, on the other hand, requires only the setting of the output register for initial condition, resulting in an operation time on the order of ten nanoseconds independent of any previous state. Conditions such as overload or time scale chosen have no effect.

The next point is that of the overload. Special compensation circuitry is required on the analog device to prevent component damage due to saturation or overload. Unfortunately, diode limiters must be used in parallel with the analog device so that a simulation may run even though various elements reach this overload state. These diodes allow integrators to integrate down from the overload condition without the "sticking" experienced when no limiters are used. The digital device never saturates or sticks in that state. Minimal circuitry is required, however, to allow the limit condition rather

than have the adder reset itself and continue counting or wrap around. This addition is a comparator circuit which disables the clock if the maximum value is reached and the sign of the input is the same as the output.

In addition to overload limit, there is the requirement for multi-input capability. Placing a tree of adders on the input allows for more than one input as in Figure 5b. Since each add circuit requires only 50 nanoseconds, the same synchronization timing mentioned above is used maintaining the overall device delay time. Using adders of 18 bits further increases capability by allowing the inputs to sum to greater than full scale, a feature possible with analog integrators.

This digital device is constructed to have performance greater than the corresponding analog device considering not only signal propagation delays but increased resetting, holding, and scaling capability. Gain changing during execution time is also possible with shift values changeable from the control section.

Summing Unit

The next component to consider is the summing amplifier. In analog form, this is composed of a high gain amplifier with resistor input and feedback.

The basic summing amplifier on the EAI 681 analog computer has a 0.8 degree phase shift at 10 KHz corresponding to a 222 nanosecond delay. The digital counterpart of this device is essentially the same as the integrator except that no consideration need be made for the feedback or scaling necessary for the high frequency repetitive addition.

The summer is constructed in the same fashion with multiple inputs and gains and uses the same clock rate as that for the integration process for gating signals through the device.

The only component and time consuming aspect of the adder used here is that of multiple inputs. With the basic adder using 50 nanoseconds for the full 16 bits, each input over the basic two would require an additional adder and 50 nanoseconds of operation time. Following the example of the analog component which has six inputs, results in a digital device composed of five adders and a 250 nanosecond operation time (Figure 5a). In this example, the digital device has a lag of approximately 30 nanoseconds longer than the corresponding analog device. One alternative is construction of multi-input adders which would decrease this addition time. Another alternative is construction of a tree structure as in Figure 5b which uses the same basic addition unit while providing for increased speed on a multi-input device. Six inputs, for example, would require 150 nanoseconds to complete operation. Also, this could be expanded to eight units, as shown, with no increase in total execution time. Beyond this expansion, every 50 nanosecond increase in total time would enable a doubling of the number of inputs. Due to the construction of the summing amplifier on the analog machine, the sum of two of the inputs could be greater than full scale since the input values are summed at a current junction on the input. This is duplicated on the digital device too with the change on the input adders of allowing a 17 or 18 bit number on the sum thus allowing a partial sum of two or

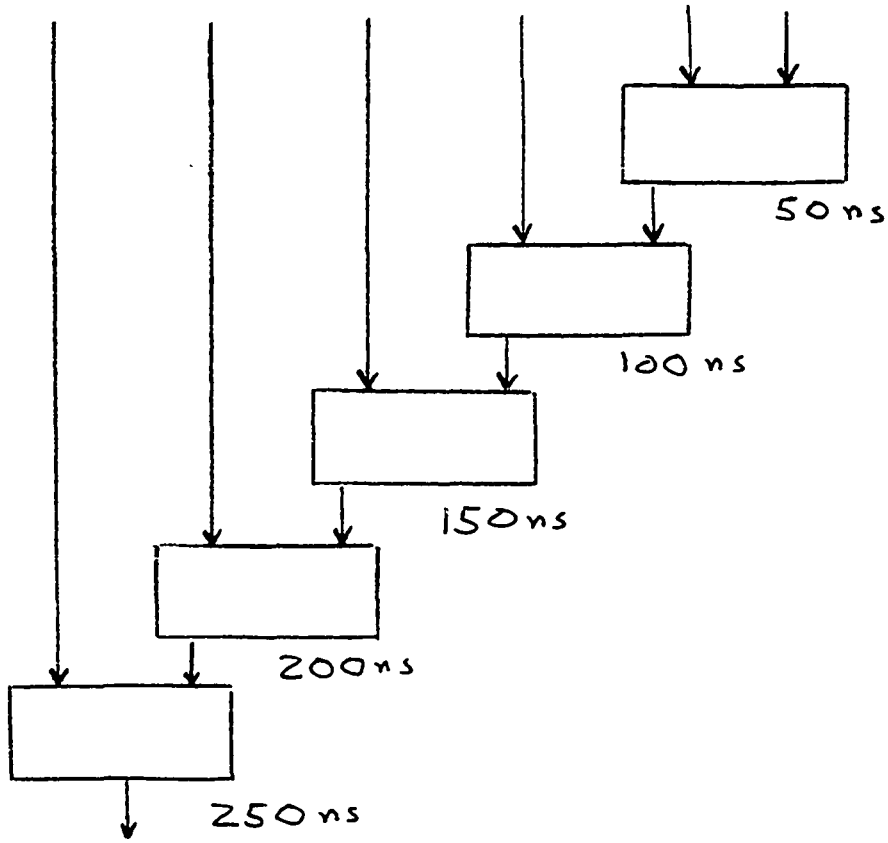


Figure 5a Summer

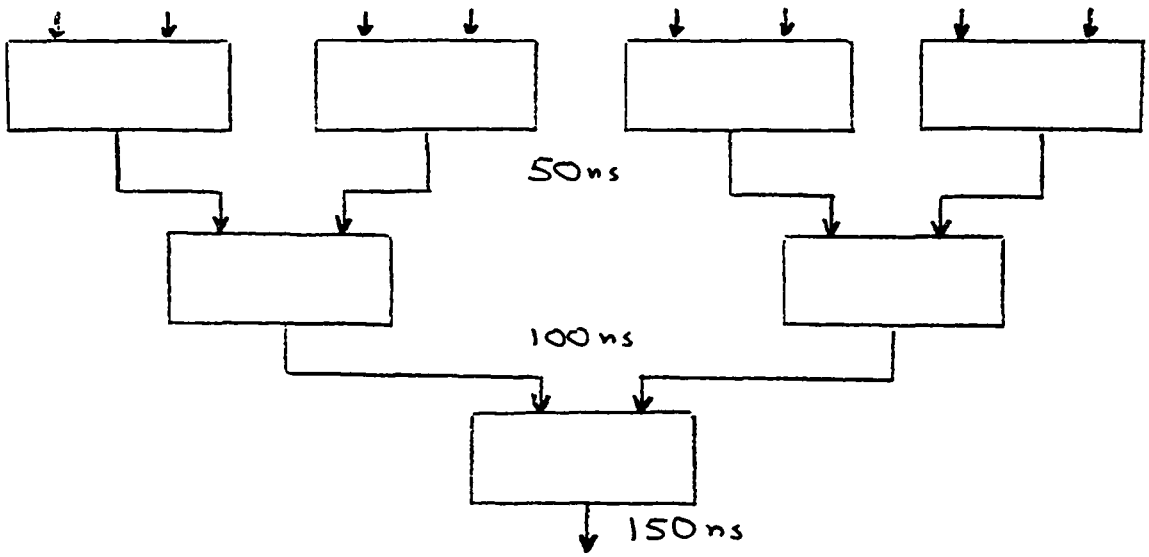


Figure 5b Summer

four times full scale. Such a design allows the partial sum to be greater than full scale even though the total sum is less than full scale.

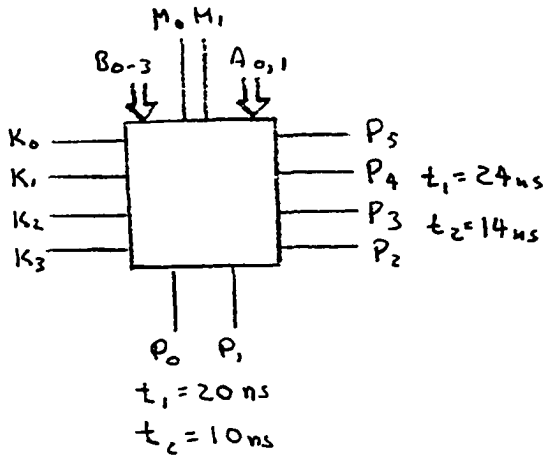
A summing amplifier in the digital form has performance capability superior to its analog counterpart. While improving on some characteristics, such as phase shift and gain accuracy, more have been added such as more inputs and the relieving of saturation ("stick") as in the integrator.

Multiplying Unit

A component quite difficult to implement in an analog machine is the multiplier. Digital multiplication of integer quantities is reasonably straightforward using recently developed solid state circuitry and can achieve speeds comparable to those of analog devices.

With a 0.2 degree phase shift at 1 KHz, the analog multiplier is displaying a delay of 556 nanoseconds. For a digital device, using twenty 2X4 bit multipliers cascaded to provide a scaled binary 16X16 bit multiplication with 16 bit output (Figure 6), a product can be formed in 118 nanoseconds, worst case. This is, of course, a significant improvement in the delay, but the more significant feature is that of accuracy. The analog multiplier is a network of resistors and diodes in the feedback path of a high gain amplifier. This network approximates, by proportional voltages, the process of multiplication. Accuracy is dependent upon the precision of these components and their respective temperature coefficients. The digital device, however, provides accuracy for the full 16 bits, or equivalently, on a ten volt

A+B occur at $t_1 = 0$
K+M occur at $t_2 = 0$



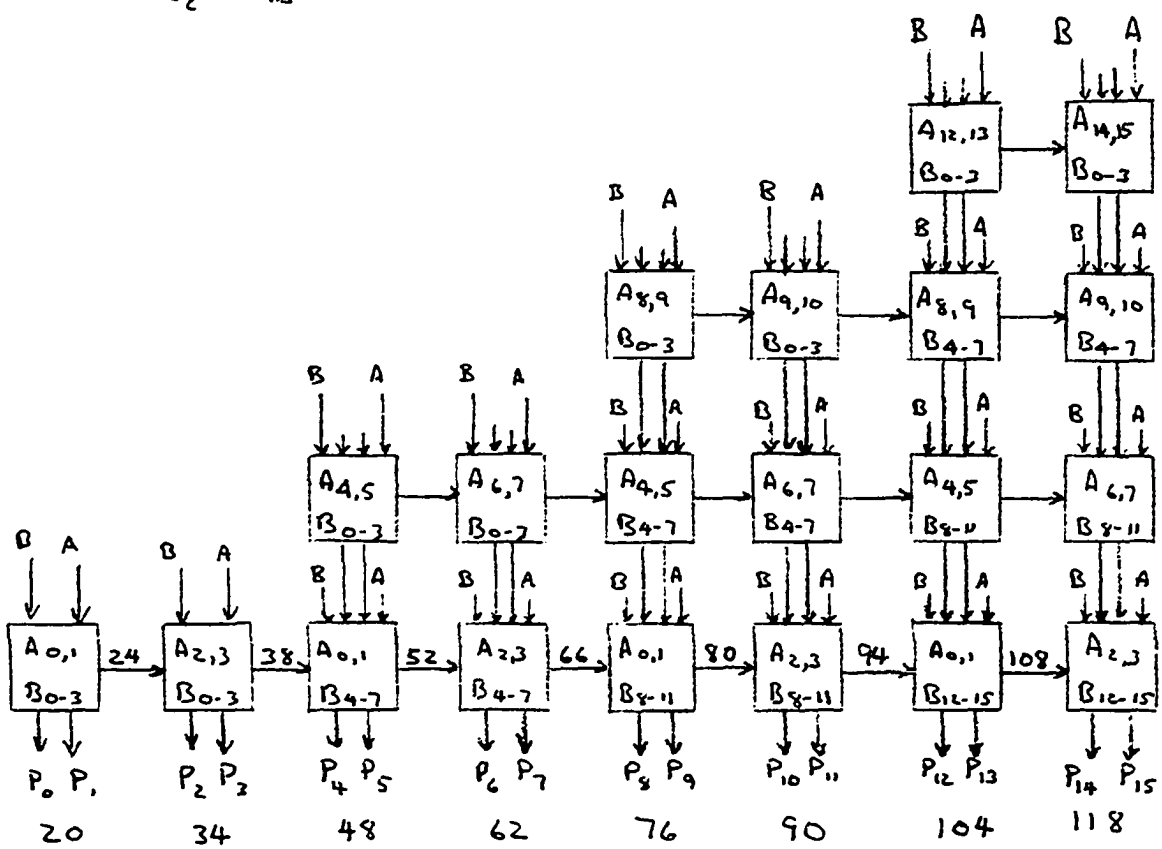


Figure 6 Multiplier.
(Time delays shown in nanoseconds.
Critical path only.)

analog machine, the nearest 0.3 millivolts. Additionally, there is no noise or temperature generated perturbation affecting its function. Aside from static multiplication errors in the analog device, the noise makes it particularly critical in feedback paths of some system simulations. So critical, in fact, that slight noise could result in system response as if there were poles in the right-half-plane of the simulated system which is, of course, an unstable condition.

The digital multiplier outperforms the analog counterpart in all aspects of its operation. In some instances, the ability to simulate a given system could possibly be impacted significantly if such a device were available.

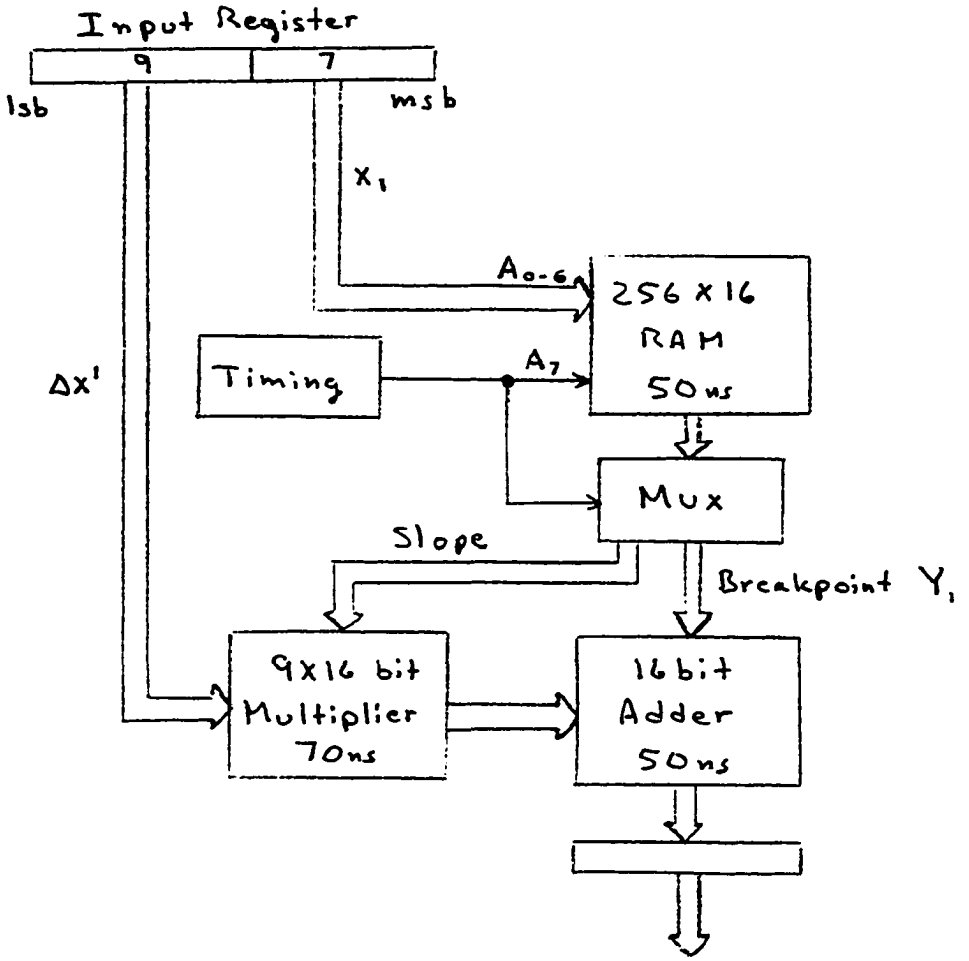
Function Generator

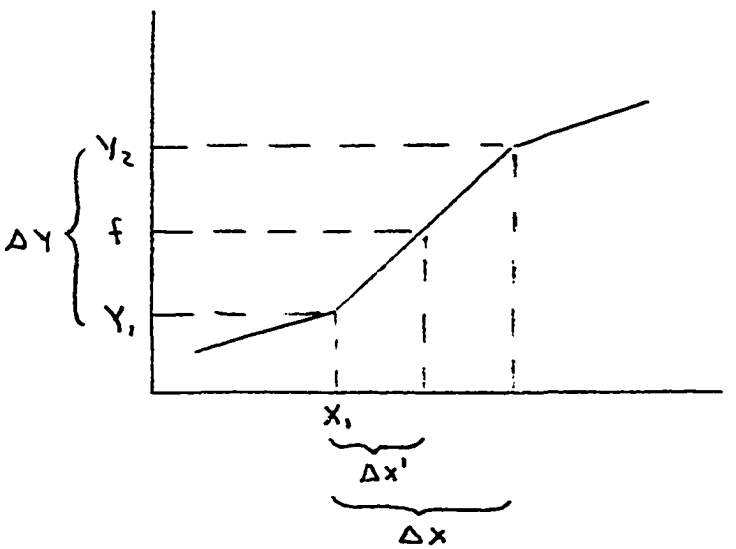
The most expensive and, therefore, rare element of an analog machine is the function generator. This device takes several forms on existing machines. There are available hard wired functions such as sine/cosine and log/exponential while the user definable functions are implemented on either manual diode or digitally controlled function generators. The names of the latter two imply their method of setup, either manually or by the digital host machine.

The manual generator provides for a function description using either ten or twenty (selectable) straight line segments individually definable. The digitally controlled generator provides sixteen segments. This obviously puts a limitation upon the programmer as to what functions are implementable, and, what is required in some cases, is judicious selection of the segment positions in order to most closely

approximate the function desired. The time delays experienced in this device are nominally 1.4 microseconds for the sine/cosine units, 2.2 microseconds for log/exponential units, and greater than 2 microseconds for the digitally controlled generators.

As shown in Figure 7, a proposed example of a digital method of implementing this device consists of using a programmable read only memory, random access memory, or other solid state memory which contains a table of function values and slopes between those values. A number placed in the input register has the high order seven bits decoded into an address in the memory which results in a function value from that cell. The address, therefore, is the abscissa value. The addition of the eighth bit addresses the next cell in memory allowing the slope of the function on that line segment to be accessed. The low order nine bits of the input are the delta x from the reference point to the value in question. The slope is multiplied by this value and the result added to the function point. This table lookup method does not provide a different function approximation than the analog device. However, with 128 segments, the function description capability is greatly enhanced and accuracy, therefore, increased. Additionally, there are none of the problems associated with this device as there are with the analog devices that use diodes and variable resistors to achieve the segmentation. As with the digital multiplier, there are no component precision requirements or temperature effects to consider.





$$Slope = \frac{\Delta Y}{\Delta X}$$

$$f(X_1 + \Delta X') = Y_1 + \frac{\Delta Y}{\Delta X} * \Delta X'$$

Figure 7 Function Generator

For ease of operation, this type of device can be programmed in two ways. First, common functions such as the trigonometric and exponential can be implemented permanently using the burn-in type ROM's and, secondly, user definable functions by RAM which can be changed by the control section at run time or during setup.

The only point of advantage which the analog device has over the digital is the capability of having a set of line segments with randomly spaced abscissa values. The digital device requires evenly spaced values for the abscissa at the breakpoints on the segments. Considering, however, the fact that the number of segments in the digital device is so much greater than the analog, this ability of random segment placement does not present itself as much of an advantage. Additionally, the even spacing more readily lends itself to computation of a function by the digital processor where constant abscissa segments are more easily computed. If more sophistication were involved, such as some curve approximation method to compute these line segments, the even spacing is more simply dealt with by existing numerical methods.

Considering the propagation delay, the total delay for this device is approximately 170 nanoseconds using a 50 nanosecond solid state memory. A slower memory access results in a correspondingly greater time delay.

This unit enjoys a large margin of improvement over its analog counterpart. It has capability which appears advantageous over the analog version.

Constant Attenuating Unit

The simplest and most inexpensive component on the analog machine is the potentiometer. It is manufactured in two forms. One, the more common servo driven potentiometer which is settable from the console by pushing buttons corresponding to the value of the coefficient. The other, which is a more recent development, is called a digital coefficient attenuator and is settable from the digital processor and contains a register with the coefficient value as sent from this digital processor.

The servo set potentiometer is a two quadrant device having only one sign on its coefficient. The digitally set device is available as either a two or four quadrant device. The best way to implement this device using digital components for this research machine is with the multiplier circuit as described previously. One addition to it increases its capability plus giving it slightly more flexibility than its analog counterpart. This change allows coefficients greater than one. This is quite simply done by allowing a shift on the output register as shown in Figure 8. Such a difference adds flexibility to the digital analog so as to somewhat simplify the mapping function necessary for the switch matrix. That is, it alleviates the need for patching amplifiers with potentiometers in the feedback loop when gains other than one are required.

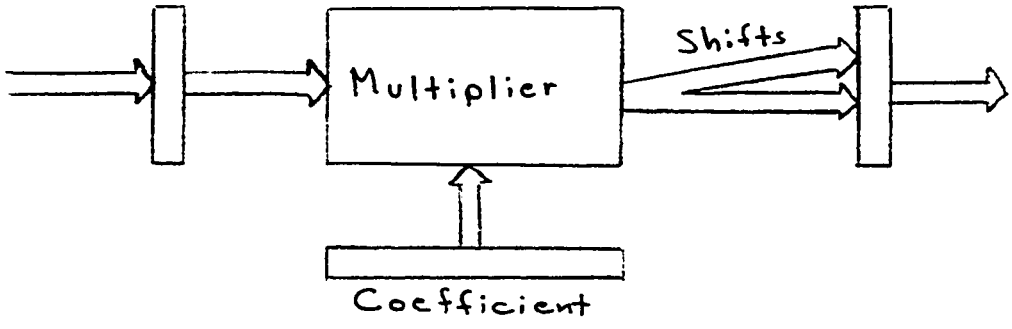


Figure 8 Constant Attenuating Unit

Time delay advantage from 277 to 120 nanoseconds plus coefficients greater than one lend an overall advantage to the digital over the analog attenuator.

Miscellaneous Components

In addition to the previously mentioned components which comprise the major part of an analog machine, there are a number of smaller components used for various types of control.

One such device is the analog-to-digital comparator. This device generates a logic signal of one or zero, depending upon the sign of the sum of the input voltages. This digital signal appears on the parallel logic section and is used to control the mode of the machine or subsets of it as well as count events. This device, in digital form, consists of an adder with decoding on the output to determine if the sum is equal to, less than, or greater than zero. Hysteresis is quite easily added by decoding more than one bit of the sum. Such a device allows two input signals to be summed and compared to a

programmatically determined value. The resultant signal can then be used to control various elements of a simulation.

The implementation of the digital-to-analog switch and function relay equivalents on this digital machine is the same as the analog machine. The device, in both cases, is a 16 bit multi-input controllable multiplexer which is already the functional capability of the switching array.

The parallel logic portion of the analog machine, whose function is to act as a control for the analog devices, consists of counters, logic gates, differentiators, and mono-stable multivibrators. These devices, all being digital, map over to the digital machine on a roughly one-to-one basis. Some of these functions are handled by the control section of the machine while others lend themselves to more flexibility if included in the components section and interconnected via the switch array.

Implementation of devices in this category is reasonably straightforward and much the same as those which exist on an analog machine.

Interface Components

Another of the critical aspects to useful hybrid computation and also more expensive is the interface structure. This is composed mainly of the analog-to-digital and digital-to-analog converters as well as the mode control, pot setting, and component addressing hardware.

With the completely digital machine, there would be no need for these devices for interfacing with a digital machine. The only requirement would be for standard input/output or direct memory access channel controllers to allow the digital computer to efficiently pass data back and forth with this digital device. Such controllers are easily designed and constructed by users possessing the required technical competence, thus making the design and implementation of an interface with a wide range ot general purpose digital computers much easier. Since a typical digital computer is a serial machine, the best way to effect data transfer with this machine is with a multiplexer timed to run at a speed compatible with the particular I/O channel used. Multiplexer speeds of 20 nanoseconds allow a 50 MHz throughput to a data channel which is in excess of speeds typically foune in present digital systems.

Considering the use of this digital machine as a real-time simulation tool for physiological experimentation where man or animals are in the loop, interfacing is required between this machine and the simulator console. Typically, the controls used in such simulations are some type of manual control stick on a potentiometer or force transducer which presents itself as a variable resistance which results in the use of a continuous voltage as a control signal. The use of this type of device requires an analog-to-digital converter of the type mentioned above. Additionally, many instruments used as indicators on these consoles require an analog voltage. This is presented by digital-to-analog converters.

While the interfacing between the digital parallel processor and the digital host machine does not require normal analog/digital interface equipment, some is required for interfacing to the external world. Fortunately, the point at which there is an extremely high data rate requires only high speed digital multiplexers, while the analog/digital interfacing is required only where a comparatively low data rate and resolution is utilized.

The general structure of this machine lends itself to a level of interfacing which is quite a bit easier to deal with on the user level, thus making it a flexible piece of equipment to use.

Switch Array

The single most critical unit of this research machine is the switch array. this array has the task of routing data among the units of the module at a high rate. The routing paths are programmed by inputs from the host machine as configured by the mapping function block. An example patch is described later. The cycle time was chosen as 238 nanoseconds or one-fourth the integrate cycle time. Within this time, all signals will be transmitted to their destinations. This unit must necessarily perform as fast as is allowable with Schottky devices or other high speed technology devices.

The proposed switch array is constructed about what shall be called a module as described in Shoup and Adams[5] and composed as in Table 2.

Table 2 Module Component List

DEVICE	INPUTS	OUTPUTS
Integrators		
1 - 3 input	3	1
7 - 1 input	7	7
Summers		
4 - 3 input	12	4
3 - 2 input	6	3
Multipliers (4)	8	4
Function Generators (2)	2	2
Digital Inputs	0	4
Comparators (2)	4	0
Intermodule Trunks (6)	6	6

A simple array of switches necessary to allow all interconnections requires an excessively large number of gates (23,808) and wires (1264), and considering the present equipment dimensions possible and density of wiring, these requirements are beyond practical consideration. The architecture of this unit must, therefore, be bus oriented with the digital signals being multiplexed.

This arrangement is formed with four sixteen bit digital busses as shown in Figure 9. Data is multiplexed on to each bus from 8 of the 31 devices in the module by multiplexers composed of sixteen eight-to-one multiplexer IC's such as the Fairchild 93S12 which is a Schottky device. Data from the busses to the component inputs is gated by using eight dual four-to-one multiplexer IC's (Fairchild 93S253) for

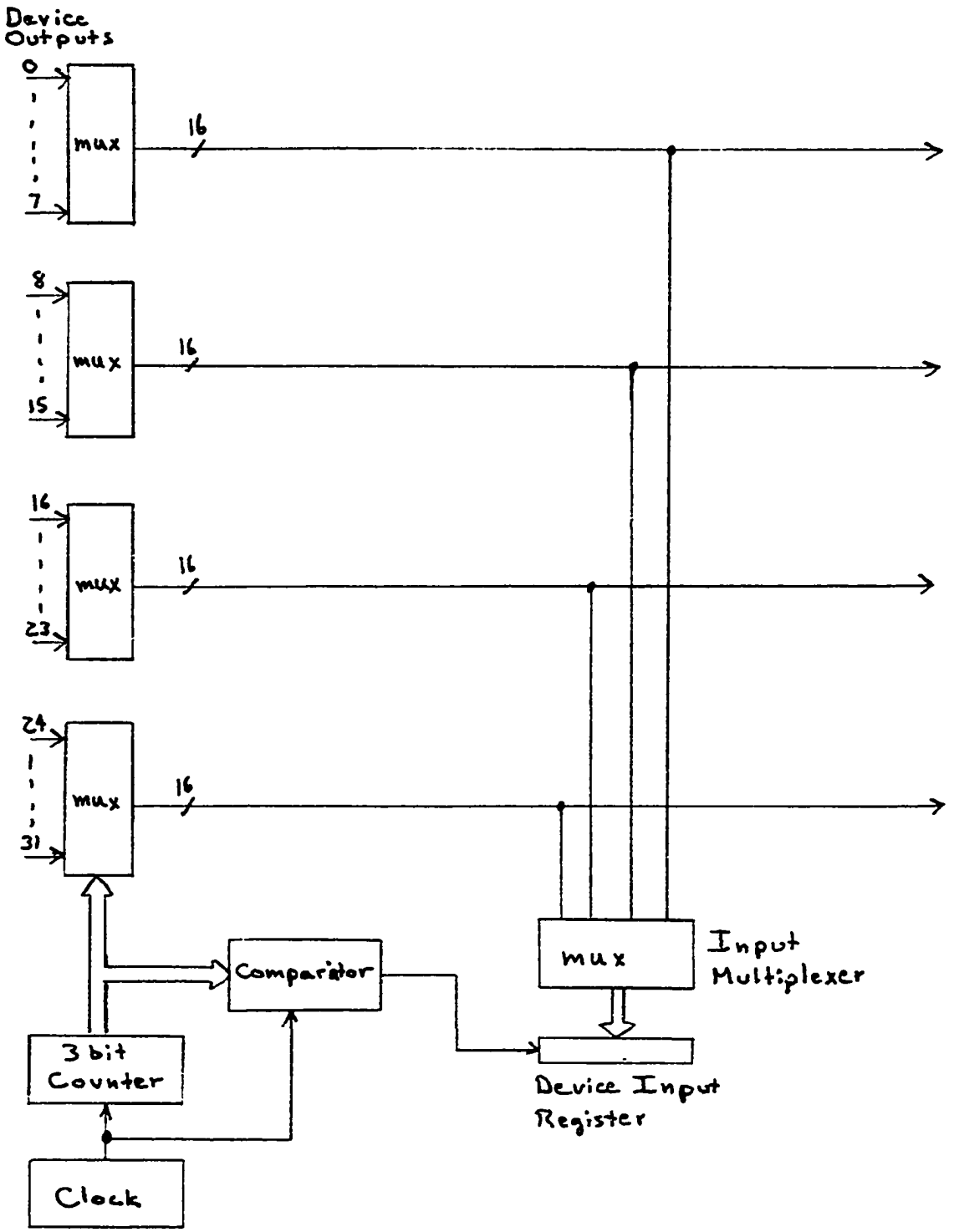


Figure 9 Switch Array

each component input. Input registers are appropriately strobed to accept data from these devices.

In addition to these devices are control circuits which accomplish the addressing of the multiplexers and strobing of the component input registers. The clock drives a counter which has three bits providing for the eight states necessary for the addresses of the input multiplexers. Since the multiplexers for the inputs are able to accept data from any of the busses for a particular patching arrangement, but not change at the same rate as the output multiplexers, the address for each one is held in a register set by the mapping function section of the machine. The strobes for the input registers, therefore, are only a function of the output multiplexer addresses plus a three bit address which is loaded into the comparator register. When the proper address is reached by the output multiplexer, a strobe pulse is sent to the device input register. The timing for the multiplexing is shown in Figure 10.

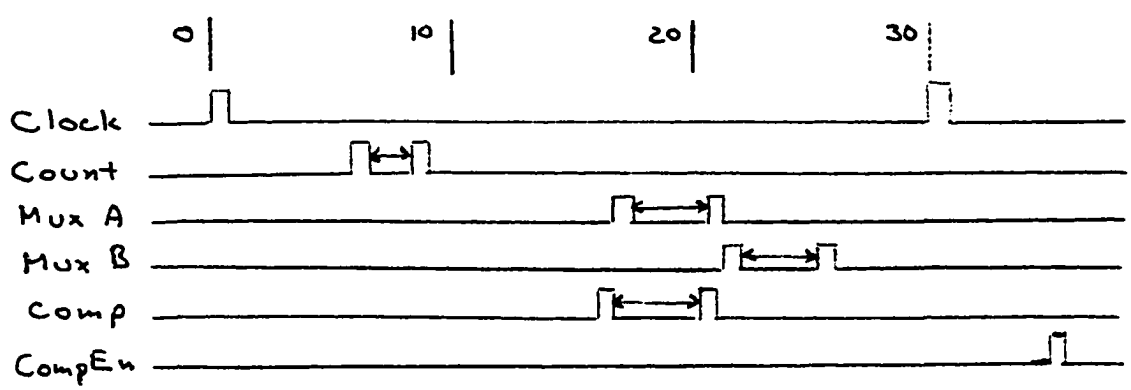


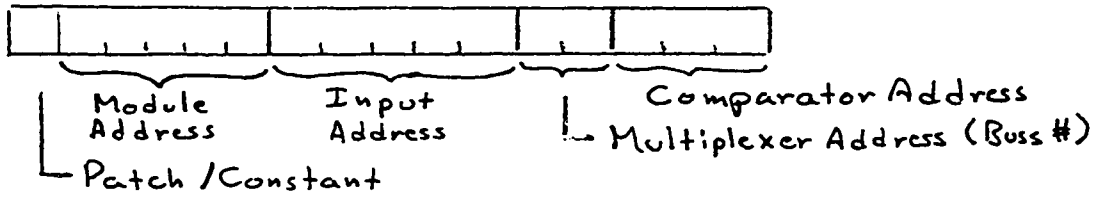
Figure 10 Switch Array Timing Diagram

Clock pulses occur every 30 nanoseconds. The pulses shown for the components of the system indicate when a complete signal is available on the output. Variations shown in the times are due to the differences between turn on and turn off values quoted for each device. The data from a multiplexer going into an input register is not used until the input register is strobed by the comparator which does so when the proper address for an output is present in the counter and the clock pulses the enable input.

The following is given as an example for the patching from one component to another:

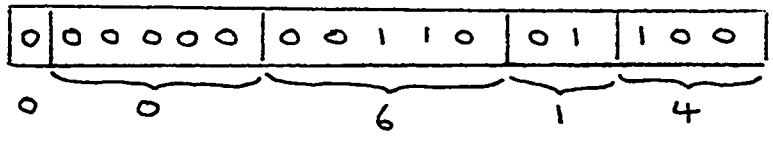
Problem: To patch unit 12 to the input of unit 6.

The patch word 160000_8 places the machine into the patch mode. The following patch word is decoded as follows:



The input address is 6. The multiplexer address is 1, since unit 12 is output on the second bus. Since unit 12 is the fifth output on multiplexer two, the select address is 4.

The resulting patch word for this connection follows:



A schematic for this action is shown below:

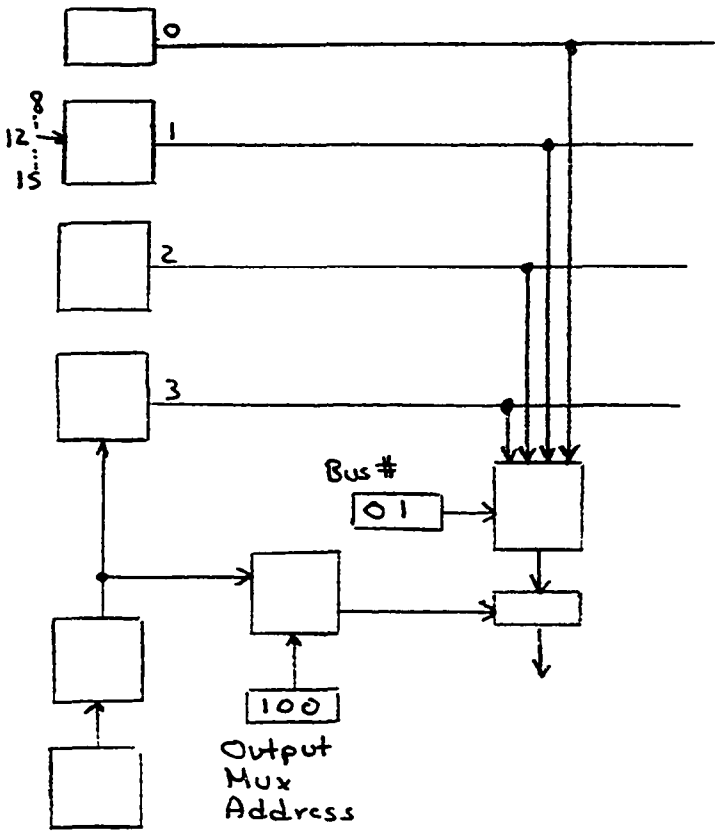


Figure 11 Example Signal Patching

Cost Comparisons

This section will deal with comparisons of cost between the individual components of the analog machine and the proposed digital equivalent. A complete comparison is difficult since the analog machine requires a great deal of switching circuitry, power supplies, and cabinetry, some of which is included in the individual prices. Likewise, the digital version is difficult to price accurately since it is not in existence and, therefore, the added cost of power supplies,

cabinets, circuit boards, connectors, and wiring cannot be estimated with a high degree of accuracy. Much of that can only be determined after final construction and minimization decisions have been made. The intent of this comparison is to show that the digital approach is economically viable.

The cost computation for the digital machine will be made here using prices for Schottky devices. The cost comparisons are meant to be a gross comparison only so as to give an estimate of the costs of these two types of machines. A general rule to follow is the application of a factor of three to seven to the component costs to determine sale price of the finished product. This gives approximate costs for production equipment. The prices used for the analog machine are those from EAI[25] and, for the digital, representative digital devices.

The analog integrator with the capability of time scale control costs $1,430 per unit. This does not include the $500 for control circuitry for the time scale option. The digital counterpart, using one adder, a counter for the extension, and necessary registers, costs in the neighborhood of $150, exclusive of the board, connectors, and power supply.

The analog multiplier base price is $1,800. The digital version, composed of twenty small 2X4 bit multipliers plus two registers, has a total calculated cost of $340.

The analog summer individual price is $600 to $945 depending upon the extras in the tray such as track/store capability, limiter

connections, zero limiter, or digitally controlled limiter connections. The summer in digital form, comprised of the adders and necessary registers, is calculated at $380.

The potentiometer cost is $120 for a servo set unit or $350 for a digitally controlled attenuator. The digitally controlled unit requires cabinetry, power supplies, and mounting racks in addition to the basic machine console. Prorated across a fully expanded machine, this results in $110 per unit, which computes to $460 per unit total. The calculated digital unit cost is the same as the multiplier since it is the same unit except for a different input connection to the register. One of its two inputs comes from a register with the other from the signal busses. This cost is $380.

The analog function generator in the analog machine costs $1,300, $1,375, $1,475, and $3,700 for manual diode, sine/cosine, log/exponential, and digitally set diode function generators respectively. The digital function generator further requires an extra $1,500 per unit in quantities of four for cabinet, power supplies, and mounting racks. As in the case of the digitally controlled attenuators, this is extra to the machine console. The proposed digital machine function generator is calculated at $480, including adder, multiplier, memory, and registers.

The cost remaining for the analog machine is the price of the main console. The basic unit, which includes the cabinet, digital voltmeter, addressing hardware, and power supply, is $35,000. The comparable

unit for the digital machine is far less since only digital displays and switching are required with the cabinet and power supplies. Additionally, there is no need for noise immune circuitry, relays, or gold plated switches throughout.

As can be seen, the individual costs of the digital units are much below those of the analog except for the potentiometer. It must be pointed out, however, that the analog prices are off-the-shelf prices and the digital are for the integrated circuit chips only. Even with these differences, however, the estimate does indicate the cost competitiveness of digital techniques.

The cost of the complete module shown in Table 2 is computed in Table 3. The various analog components were chosen which possess functional capability most like the digital analog.

Table 3 Single Module Cost Comparison

Device	Analog	Digital	X3	X7
Integrator	$1930	$150	$450	$1050
Summer	945	380	1140	2660
Pot	460	380	1140	2660
Function Gen.	5200	480	1440	3360
Mult.	1800	340	1020	2380
	Complete Module			
Integrators (8)	$15440		$3600	$8400
Summers (7)	6615		7980	18620
Mult. (4)	7200		4080	9520
Func. (2)	10400		2880	6720
	$39655		$18540	$43260

HARDWARE TESTING RESULTS

The circuit in which the digital hardware is performing at maximum rate and, hence the critical portion of the system, is the switch array bus multiplexing. In the previous discussions of the switching array, minimal consideration was given to the details of requirements for actual hardware. The following discussion addresses some of these considerations for a test case breadboard of the multiplexed bus.

Considering the design shown in Figure 9, several lines will extend from the driving source across the backplane to all the component boards. The total length would probably not exceed 24 inches except possibly for any extra length required for board removal or other access. These lines include the clock, address, and data bus lines. Each of the propagation times for these plus the required line driver delay time must be included in the timing calculations. The line drivers are required to drive the large number of multiplexer inputs (48) as well as comparators; the fan-out capability of the multiplexers being only ten. The Fairchild 9S140 line driver was chosen as an appropriate device as its fan-out rating allows it to drive 30 multiplexer inputs. For each line requiring a driver, each driver can drive half of the total components, resulting in a load of 24 inputs. This leaves each driver with an excess capability of 6 inputs and keeps it well below its designed maximum. The modified schematic is shown in Figure 12 with each required driver included.

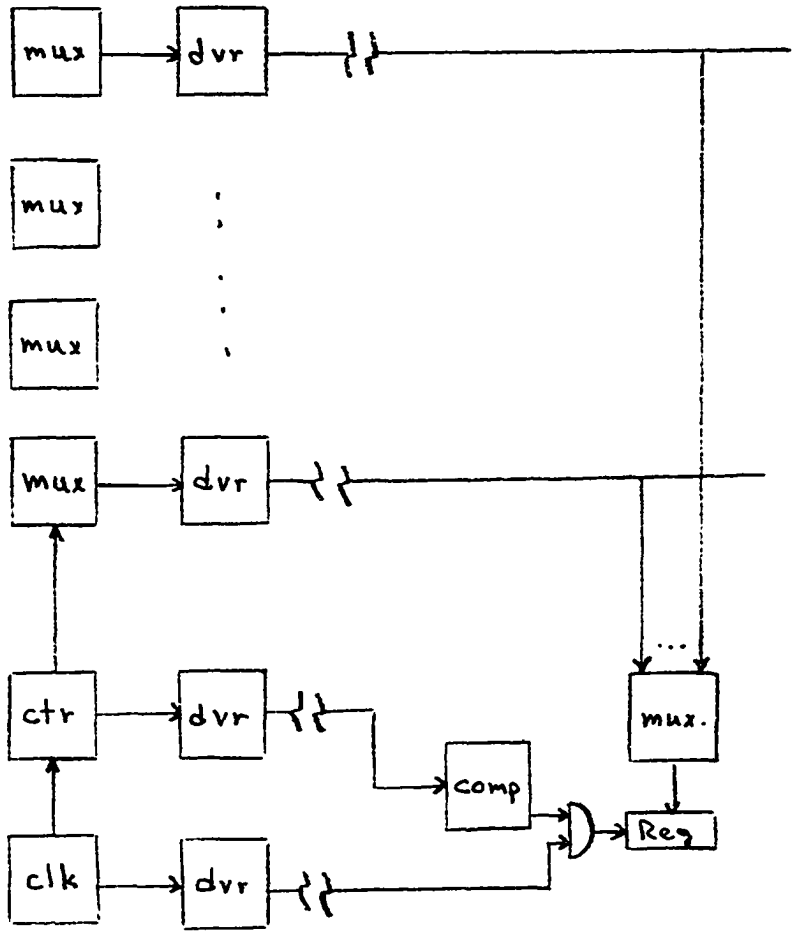


Figure 12 Switch Array with Line Drivers

The configuration chosen for the test is shown in Figure 13. The transmission line used is a round conductor ribbon cable with the signal on the center conductor and all other conductors connected together and used as signal returns. This cable was chosen as an appropriate medium for this short distance requirement because it reasonably approximates backplane wiring. Other possibilities

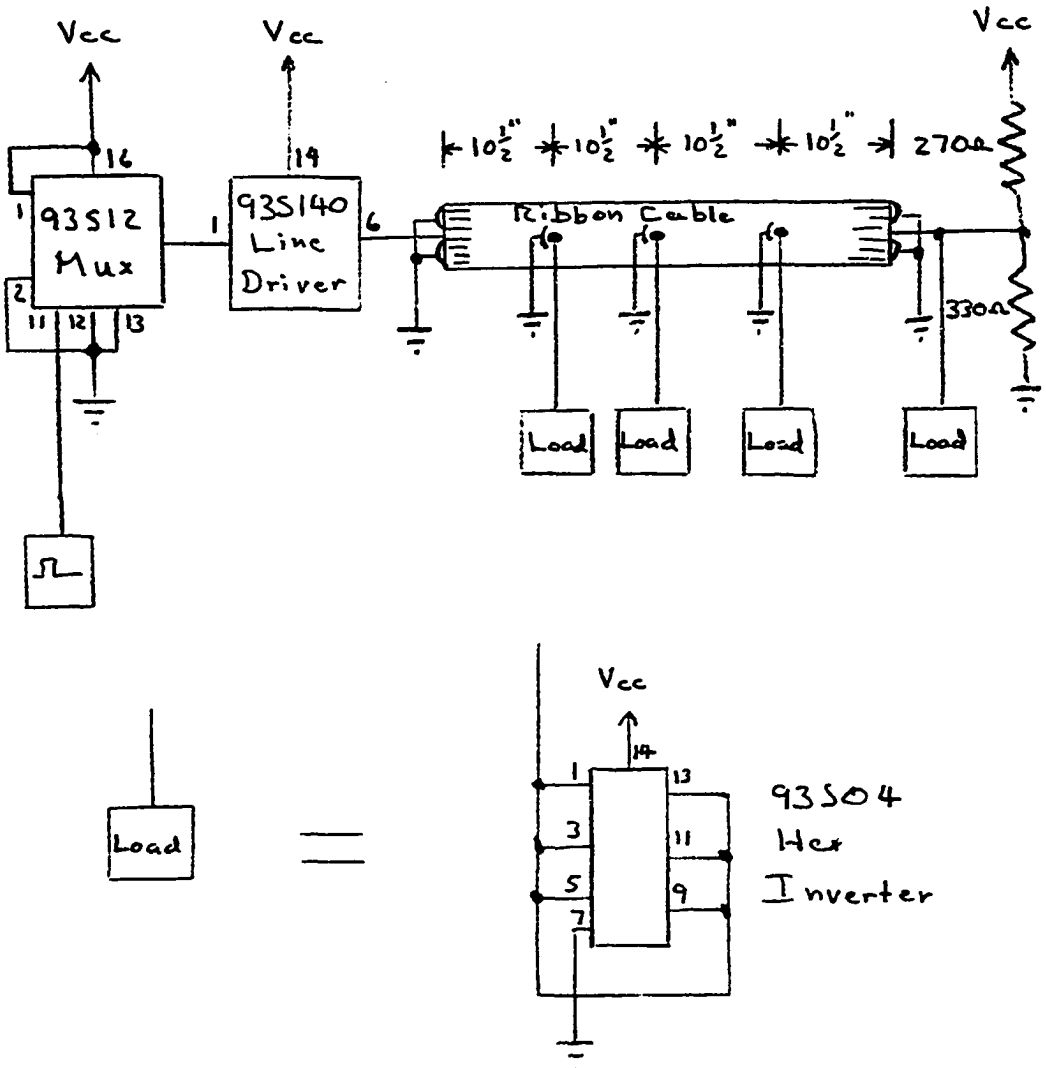


Figure 13 Hardware Test Setup

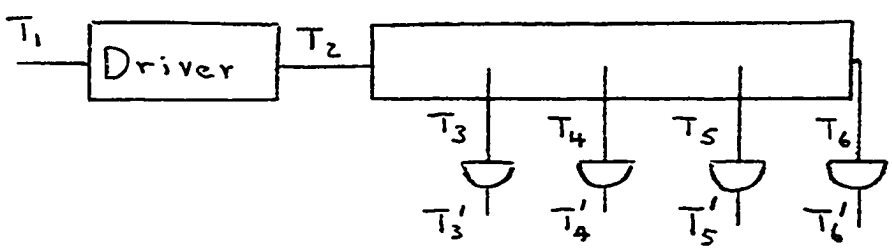
include a flat conductor ribbon cable, a twisted pair ribbon, or a ribbon with a ground plane. These were not tested here.

The components used in the test were the multiplexer, line driver, and hex inverter chips. The clock was connected to the address input of the multiplexer with two inputs grounded. An input word was connected to the device so that, as the clock cycled, the output

changed between a logic "0" and "1". The line driver received the output and drove the transmission line. At the far end of the line, a termination resistance was used. The total length of the cable was 42 inches and a six input load was connected every 10½ inches resulting in a uniformly distributed load as an actual machine would have.

The oscilloscope used was a Tektronix 7904 which has a 200 MHz bandwidth. The voltages were measured through a resistance which caused the scope to appear as a six load device to the line. Each load was disconnected as the measurement was made at the particular port in the cable so as to keep the six load per port distribution constant.

The following test points were used:



The following table shows the delay times measured at each test point. The time was measured from threshold to threshold at the +1.5 volt level. Times are in nanoseconds.

	line pos	going neg		line pos	going neg
T1 to T2	16	16	T2 to T3	14	20
T3	18	25	T4	20	22
T4	22	26	T5	24	24
T5	27	28	T6	25	24
T6	28	28			
T1 to T3'	31	30	T2 to T3'	27	26
T4'	34	32	T4'	30	28
T5'	38	34	T5'	32	30
T6'	40	40	T6'	36	38

These figures include the 10 nanosecond delay between the two probes since the test leads to the scope were of different length. Analysis shows a 3-4 nanosecond delay between each load point resulting in a 13 nanosecond delay for the entire length of the cable. If the cable were driven from the center with loads distributed on either side, roughly half the delay would be realized.

The conclusion is that the 30 nanosecond clock period can be maintained as is shown in the timing diagram, Figure 14. The signal

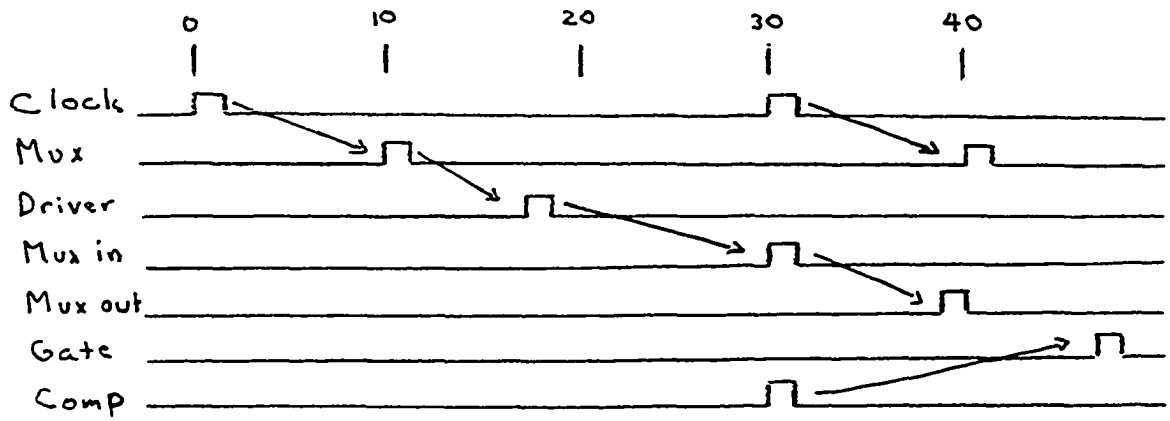


Figure 14 Hardware Test Timing Diagram

from the gate to the input register strobe is required to be 7 nanoseconds after the appearance at the inputs of the desired signal for proper triggering of the register. Additional gates or equivalent are required for providing additional delay to meet this constraint.

This breadboard test is close to actual conditions, and improvement of the propagation in the cable beyond minimal levels is not anticipated in a prototype implementation. Improvement would, however, be realized in suppressing the high frequency ringing when the circuit is constructed on printed circuit boards with appropriate considerations for that medium.

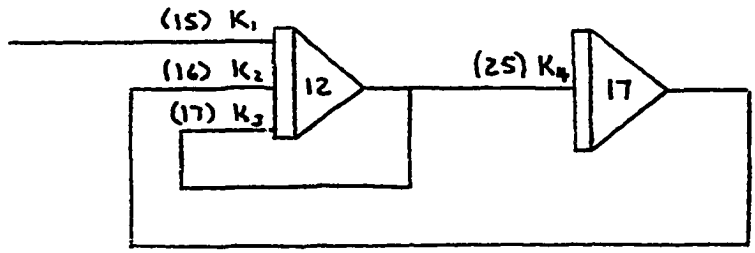
APPLICATIONS

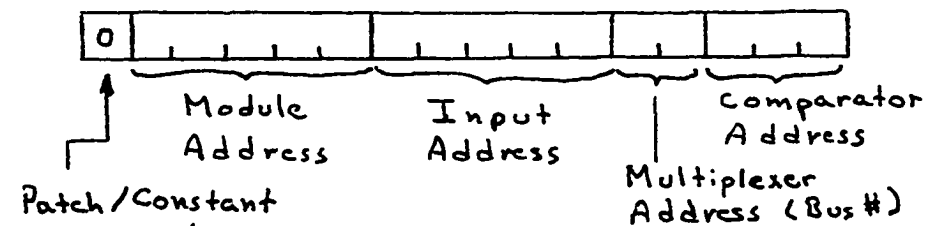
Consideration is now given to various aspects of the machine specific to given applications. First, a sample generic program showing how patching would be achieved is discussed followed by examples of specific problems. Then, applications in a macro system are discussed.

Sample Patching Program

The performance of the patching function requires a mapping function to convert a generic patching diagram to one specific to the hardware of this machine. An example is given which patches a second order equation which utilizes two integrators and four constants (quasi potentiometers). The patch scheme is shown at the top of Figure 15. The patch/constant mode control is shown, as is return from patch to control. Also shown is a brief run time scenario to display how execution would be handled. This represents a complete setup and execution of a problem excluding the patching required for I/O to and from the system. This would most likely be best handled through the trunks to the I/O structure or another patched equation in another module.

The mapping function would have as its output the set of words shown in Table 4 with decoding as shown in Figure 15 and Table 1.





Attenuator Word Format

a: 1 | attenuator address

b: constant: 2's complement

Control Word Format

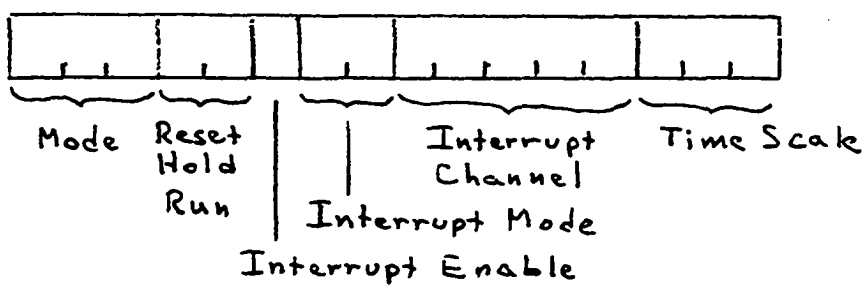


Figure 15 Patch Word Format

The following buffer of control words would effect the patching of the above problem as well as run a solution with an intermediate HOLD condition.

Table 4 Sample Patch/Run Program

1:	1110000000000000	Set patch/constant mode
2:	1000000000000001	Constant set K1
3:	0110011000001101	K1 set
4:	1000000000000010	Constant set K2
5:	1001110100010001	K2 set
6:	1000000000000011	Constant set K3
7:	0110101111000101	K3 set
8:	1000000000000100	Constant set K4
9:	1010011101010011	K4 set
10:	0000001100101100	Patch I12 to I17 (25)
11:	0000001000101100	Patch I12 to I12 (17)
12:	0000001000010001	Patch I17 to I12 (16)
13:	0000000111111010	Patch Ext 1 to I12 (15)
14:	1111111111111111	Return to control
15:	1100000000000000	Set time scale = 1
16:	1000000000000000	RESET
17:	1001100000000000	RUN
18:	1001000000000000	HOLD
19:	1000000000000000	RESET

Referring to Figure 9, integrator 12 is output 12, integrator 17 is output 17, K1 is input 15, K2 is 16, K3 is 17, and K4 is 25. Also, ext 1 is intermodule trunk 1 which is output 26.

Further details of the control function are shown in the control unit description and the patching in the switching array section. Entry 18 would actually be preceded by some condition code to determine when the HOLD should be executed, most likely an internal time clock in the host digital machine or possibly an interrupt generated by a condition within this machine.

Example Problems

As a further feasibility test, an example simulation is examined. This is the well-known Van der Pol's equation.

$$\ddot{y}-(1-y^2)\dot{y}+y=0$$

The patch diagram for this on an analog machine is as follows:

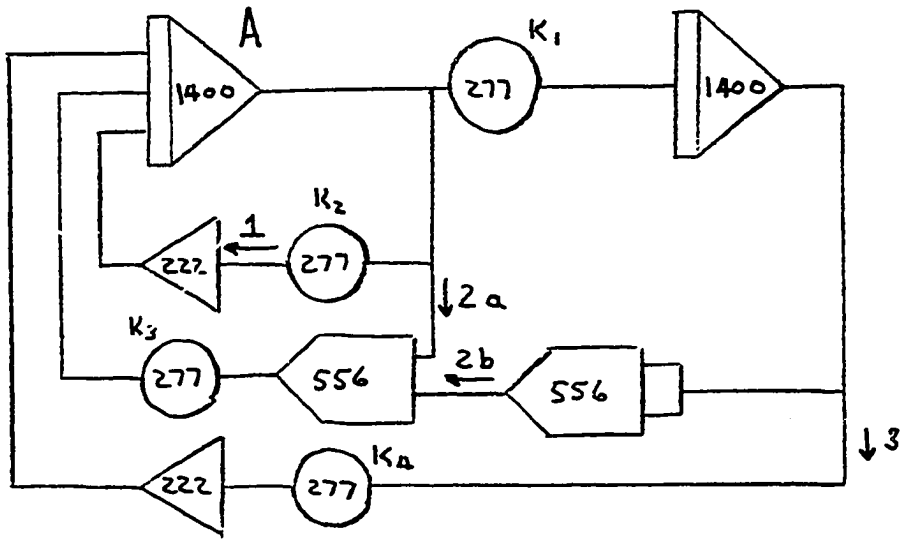
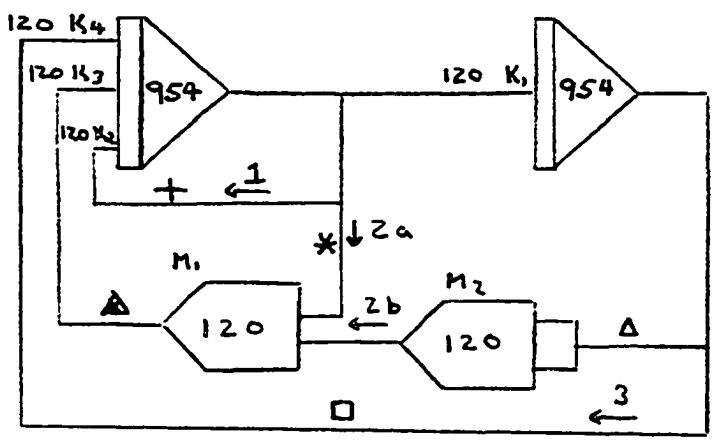


Figure 16 Van der Pol's Analog Diagram

The patching on the digital machine would be as follows:



All times shown are delays in nanoseconds.

Figure 17 Van der Pol's Digital Diagram

The point of consideration is the timing of the simulation and resulting skew at the multiple input devices. As shown in the analog patch diagram, there are four paths for the signals to take to make a circuit from point A and return. The timings and skews on the multiple inputs are shown in Table 5.

Table 5 Van der Pol's Equation Time Delays

Point A to A time delays in nanoseconds				
Path	1	2a	2b	3
Analog	1899	2233	4466	3576
Digital	954	954	1908	1908
Time skew at integrator one relative to the K2 input				
Time in nanoseconds				
	Analog	Digital		
K3	2567	954		
K4	1677	954		
Time skew at multiplier in nanoseconds				
Analog	2233			
Digital	1194			

Utilizing the timing method described earlier (integrator unit description), Figure 18 shows the timing for the set of components necessary for the simulation of the digital machine. All devices are synchronized at four times the integrator add rate, or every 238 ns. This results in the outputs of these devices being changed every 238 nanoseconds according to their inputs. The integrator add pulse occurs

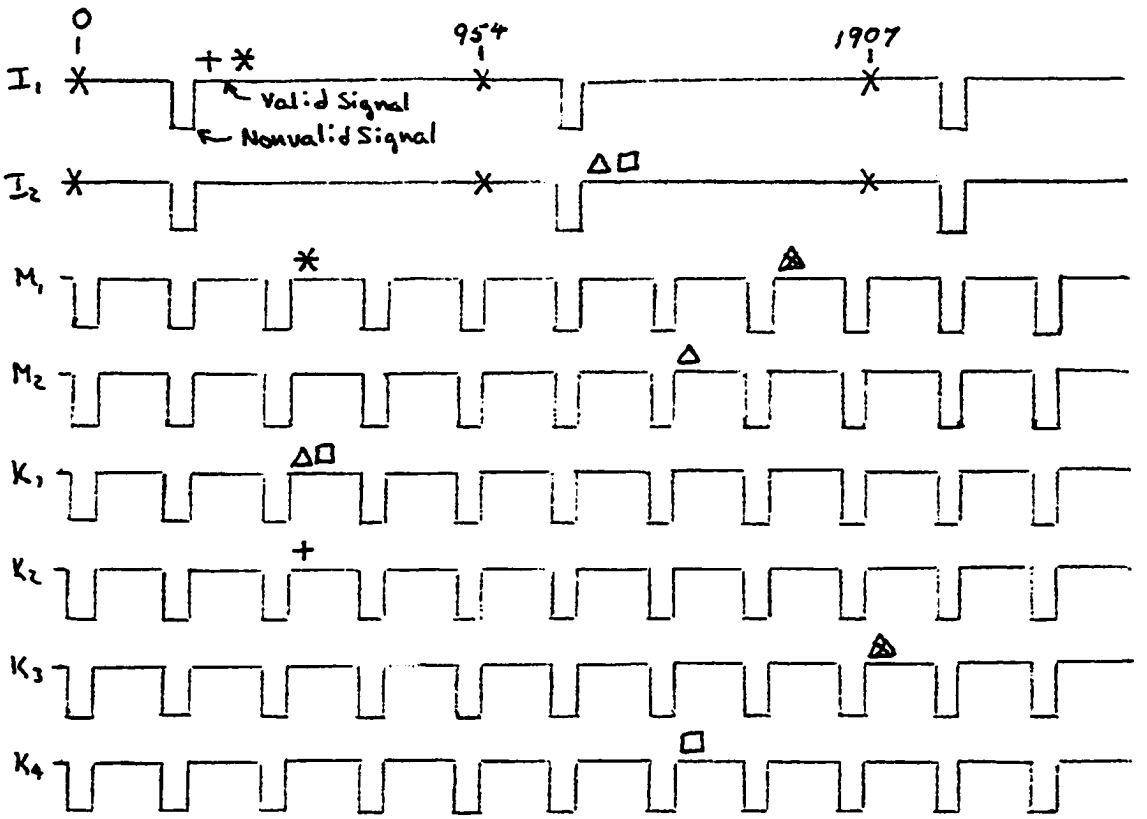


Figure 18 Van der Pol's Equation Timing Diagram

every 954 nanoseconds with its output available 238 nanoseconds after each add pulse. The timing for each of the three pathways are indicated by the symbols. The resulting path delays are also shown on Table 5. It must be pointed out on the skews for the digital machine for the integrator inputs that the two inputs from K3 and K4 are available only one integrator add cycle later than the K2 input. Since the integrator cannot use the inputs until the add pulse comes, the delay must be computed by how many add pulses delay there is from when the integrator should have received the input. For path one, there is no delay since the input is ready for the next add pulse. For

paths two and three, the inputs from K3 and K4 are ready only one add pulse after K2, therefore, resulting in the 954 nanosecond delay rather than the delays considering only propagation which are 954 and 1335 nanoseconds for K3 and K4 respectively.

Example two is a third order equation of the form

$$y''' + k_1y'' + k_2y' + k_3y = f(t)$$

and is patched on an analog machine as follows:

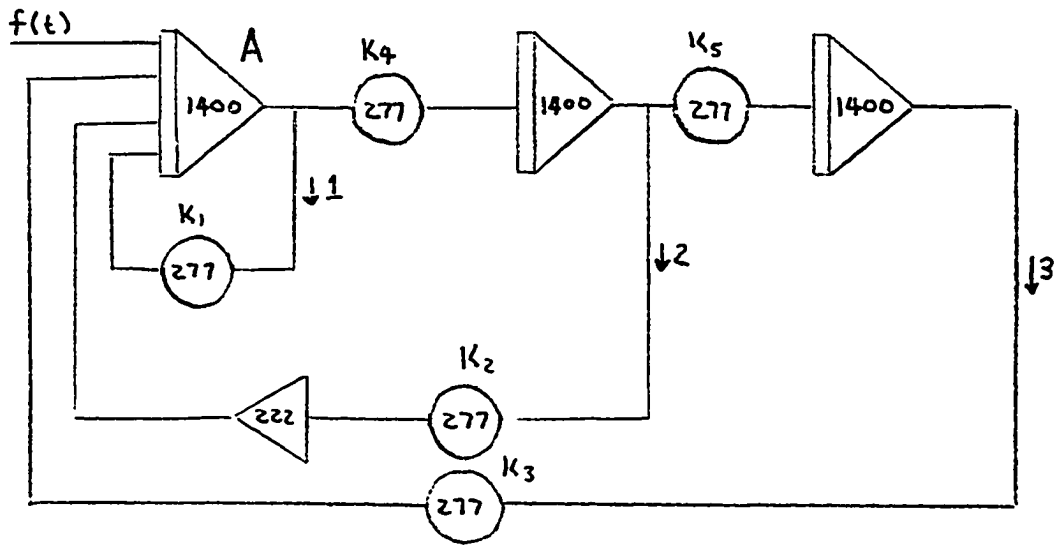


Figure 19 Third Order Equation Analog Diagram

On the digital machine, this equation would be patched as follows:

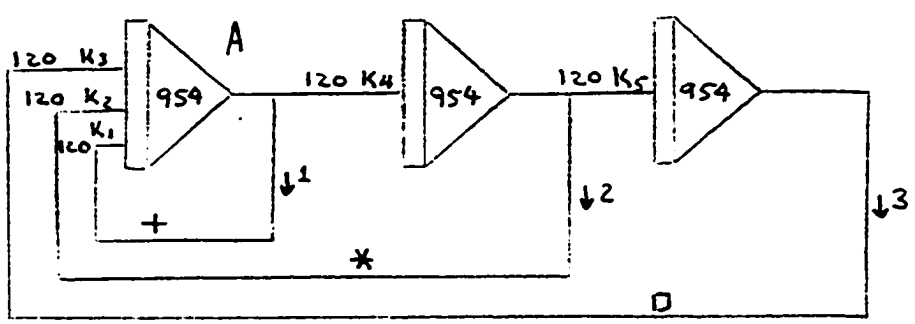
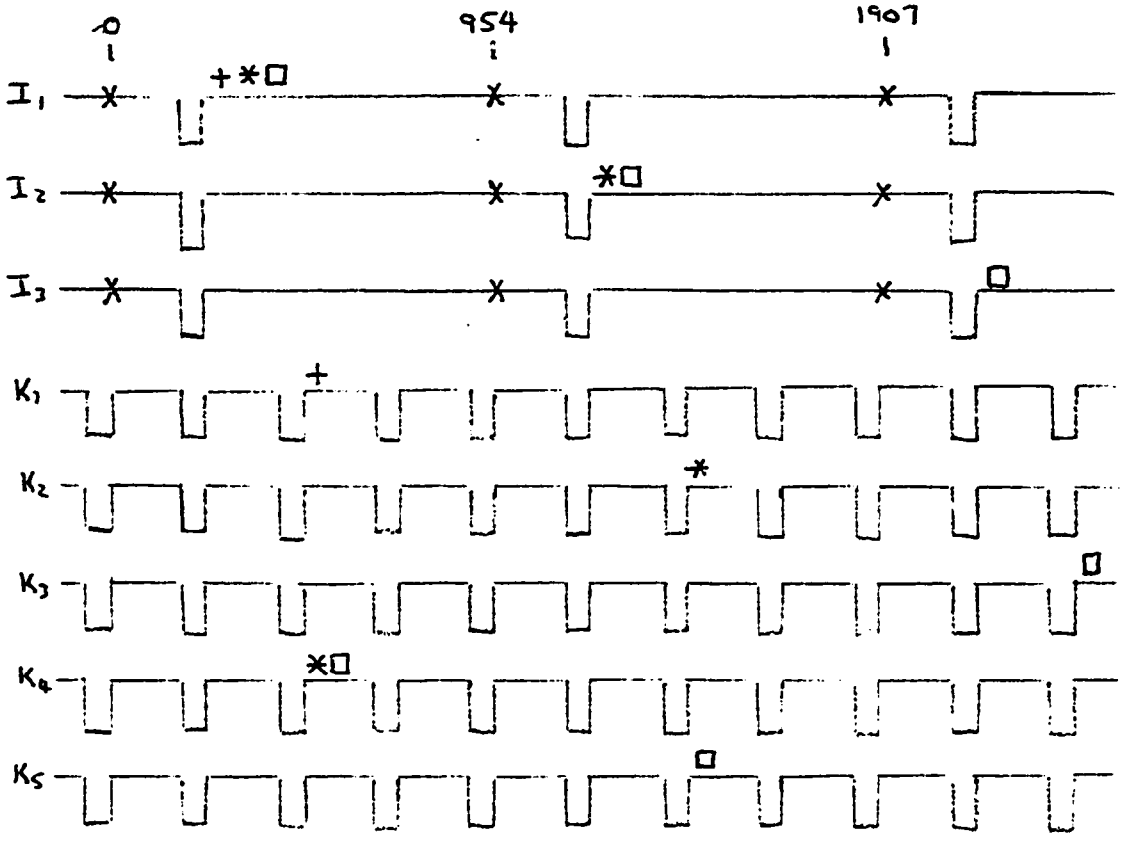


Figure 20 Third Order Equation Digital Diagram

The timing for the three paths shown on the diagram are in Figure 21.



Valid signal available 

Symbols refer to Figure 20

Figure 21 Third Order Equation Timing Diagram

The time delays and skews presented at the input to the first integrator are shown in Table 6.

Table 6 Third Order Equation Time Delays

Point A to A time delays in nanoseconds			
Path	1	2	3
Analog	1677	3576	5031
Digital	954	1907	2862
Time skew at first integrator relative to K1 input times in nanoseconds			
	Analog	Digital	
K2	1899	954	
K3	3354	1907	

Again, as in the first example, the digital time skews are computed based on the number of integrator add pulses difference between the availability of two inputs.

These examples show that for this patched problem, the total performance is significantly different from the analog machine. These reductions of both loop propagation delays and skew at the multiple input devices have a significant effect on the stability of simulations.

As a last example, a metric is applied to general purpose digital machines to generate a figure of merit for doing simulation of continuous systems. An estimate by a large Air Force hybrid computer lab was made for a typical six degree of freedom simulation of a missile. This estimate was based on the bandwidth requirements of the

flight control system and was deemed sufficient to pass all the frequencies of interest. The number was computed as 100-200 hertz. A computation was made of the total number of instructions required and their type, resulting in a digital load which the machine must handle for one complete loop through the simulation equations. The elements included multiplies, adds, trig functions, and functions of one, two, and three variables. With the above bandwidth, and using a rectangular integration scheme, the number of passes per second is required to be 10 - 20 thousand. This would allow one hundred passes per cycle of the highest frequency of interest. The resulting figure of merit was 100-200 million operations per second. For a comparison with existing digital machines, the CDC 6600 computer is capable of three million operations per second. The Sigma 7, which they currently employ in the hybrid facility, is capable of 0.3 million operations per second.

A further indication of speed is found by using the simulation language MIMIC[26]. A sample run was made in the above Sigma 7 using Van der Pol's equation. The interval was decreased until the simulation ran in real-time. This interval was found at one millisecond, which is then equivalent to a ten hertz bandwidth. The difference between this machine and the CDC 6600 is roughly one order of magnitude which would put the example at 100 microseconds or a 100 hertz bandwidth.

The above calculations give an estimate of the calculation capability of two general purpose machines employed in differential equation solutions in a real-time mode. Given a complex set of

equations to simulate, the limits of the general purpose machine are clear. The analog machine and the machine described in this paper are capable of executing these examples in real-time. Of course, as the complexity decreases and the bandwidth decreases, the two capabilities would converge.

System Applications

The computer systems which could make efficient use of this proposed digital machine vary widely in application. These include those used by industrial, governmental, and educational institutions which do scientific computation. Some specific applications include such simulations as aircraft, chemical processes, mechanical control systems, and population flow. Quite obviously, there are more too numerous to mention, but these span the spectrum of hybrid system users.

There are a great many small industrial organizations who use computer facilities to simulate their products or production processes for any necessary changes or updates which might be required. This machine appears useable to advantage in this area. Aside from the obvious advantage of automatic patching (no board), personnel trained in the use of analog or hybrid machine operation are not needed. These personnel are typically difficult to find and are subsequently relatively expensive to employ. The capability needed, instead, is that of a good digital programmer with necessary mathematical capability and those possessing this are more numerous.

Utilizing the small module capability, any number of modules can be combined without purchasing an entire fixed size machine as is

presently required. Cost is about $150K to $200K for a medium sized analog system. A digital system, which can be very closely tailored to a specific requirement much as general purpose digital computers are, results in the ability to purchase only the size required. The capability for rapid program change would also lend itself to more experimentation with programming on the part of the programmers without a need for more patch boards and cords.

Software costs for this new type of system can be significant. The magnitude of the development depends upon what degree of micro-level programming is to be included. As a minimum, the system can be patched as an analog system is with only minor software required for decoding. The software required for a mapping function which results in a high level programming language represents the other end of the spectrum. Such a language would allow user access to this system at a level which makes all patching and control totally transparent to him.

The relatively small maintenance required for a digital system is an important asset. For large systems, this is quite significant because analog components, due to their design, require close scrutiny to keep their performance within factory tolerances, and, therefore, useful in accurate simulation work. For personnel time alone, maintenance cost can be on the order of $10K to $15K per year.

A second potential area of use for this proposed digital analog machine is in an educational institution which is usually limited rather severely by funds. Typically, time and trained personnel are not

a problem. The points of greatest advantage are maintenance, size tailoring, expansion capabilities, and internal modification capability. The first, maintenance, is a great advantage since no sophisticated test equipment or large parts inventory is required nor are specially trained personnel. The capability of specific size tailoring will allow a greater flexibility in application to small instructional machines or larger simulation tools. Expansion capability allows for a basic system to be effectively expanded as grants or contracts become available with no appreciable loss in overall performance or maintainability of the system because of an ad hoc nature. The internal modification capability lends itself quite readily to use as a valuable instructional tool for digital system or sequential circuit design and, therefore, is useable as an excellent test bed for research involving advanced digital circuits design. Secondarily, there is the advantage from the automatic patching which would allow for a great many users both familiar and unfamiliar with analog systems.

Another potential application is that of real-time simulation involving a man-in-the-loop simulator. This type operation would not greatly benefit from the speed capabilities of the digital system since the frequency range typically encountered is quite low, less than 50 Hz. A feature required, on the other hand, is good long term stability. As pointed out in the design discussion, individual components are quite stable as noise does not effect their operation. This is particularly advantageous if integrators with small inputs or divide or square root circuits are used. These, and especially the latter two

in analog form, are extremely sensitive to noise and generate enough of their own noise to be marginally stable. In aircraft simulations, these circuits can severely and adversely affect the entire simulation if not operating as expected. As many of these applications involve very large control systems, large time delays and skews of inputs result in the analog signals which can force the system into an unstable condition. A fourth, but not quite as critical circuit, is the sample and hold circuit. The digital system has the capability of holding a variable indefinitely. It must be pointed out that the digital system, when operating a simulation such as this, would receive noisy signals through the analog-to-digital converters used to input control signals from the human operator control stick.

In general, the proposed digital machine appears to provide a greater throughput per unit cost for a wide variety of applications than existing analog machines. This is due to the lower life cycle costs of digital circuits. While the initial cost is possibly lower on the component level, only the full development of such a system would yield a comparison sufficient for a full economic analysis. Performance should not be discounted, however, and many features in this machine could offer a significant advantage over analog machines.

SUMMARY AND CONCLUSIONS

In summary, this machine appears viable. It represents many features specifically available with digital technology which, on the basis of this analysis, offer an improvement over existing analog technology.

Included is automatic patching with flexibility and high speed reconfiguration. Automatic patching is not presently possible as an economically viable technology on present analog machines. Further significance is found with the possibility of increased performance in certain critical areas which currently require judicious programming techniques. These include high speed repetitive operation, initial condition set, sample and hold, and function generation. These represent the weakest areas of performance for analog machines and appear easily dealt with by means of the proposed digital technology. Yet further technological advancement appears feasible in the signal transmission to host digital machines since an entirely digital medium is realized. This could overcome the inherent limitations in analog/digital interfacing when used with high data rates. With the entirely digital technology, analog interfacing is necessary only at low data rate points, therefore, increasing total system reliability. The last contribution to analog technology development is in the area of cost of operation and ownership. Digital technology is inherently less cumbersome in maintenance, repair, and operation.

The above areas, in total, represent a significant contribution to

the area of analog and hybrid computer technology and further work toward the development of such a machine would appear useful.

AREAS REQUIRING FURTHER INVESTIGATION

This dissertation presents the concept of a highly parallel digital processor which could significantly impact the field of analog and hybrid computation. Although the concept is complete, a great amount of further work in the area of hardware implementation is required. This involves many theoretical aspects of hardware as well as firm design decisions. In general, this is in the frame of the development of a prototype of the proposed machine.

Hardware Implementation

In pursuing this development, other hardware considerations such as board densities, cabinet capacity and sizes, and power supply requirements need be made on a completely designed system before evaluation of physical parameters is realistic beyond an approximation. Board density alone is a parameter which requires a certain effort, not only of circuit design and component selection, but possibly computer aided design of the boards themselves. Board densities will effect the size and proportions of cabinets necessary to house the system as will power supply requirements. Additionally, connector pin savings or costs are part of the process and can have a large impact on total system cost. Further, board care need be exercised in designing proper circuitry which can handle the high frequency multiplexing without degrading the signal.

As the hardware becomes defined, the size of the machine and its smallest units can be determined so as to affect the expansion capability as well as size tailoring advantages. Some of this

necessarily involves the marketing decisions to be made as with any such machine.

Bandwidth Increase

Another area of work, which could prove interesting and has a potentially high payoff, involves added circuitry which would provide an extrapolation lookahead capability. The application of this would be to provide an extrapolation forward in time of a derivative or other function which is a function of time. The effect of providing such a capability would be that of decreasing loop delays in a particular equation. This would enhance the already low delays available due to the synchronization in the system and drive them to zero for many cases. The suggestion would be to construct a second dimension on the switch array which would handle transfer of the lookahead data. Two levels of complexity might be possible with this: one, lookahead on all components with a feedforward from all of them, or secondly, only on the integrators. The mathematical utility of this would necessarily need defining within the hardware constraints.

Another and related concept is one of a modified integration scheme which would employ trapezoidal or some other computation. This would lead to increased bandwidth throughput over the rectangular method currently in the design. Additional circuitry and timing considerations should be added as storage is required of past data.

Miscellaneous Additions

Another major area needing further work is that of the

mapping function. As stated earlier, its task is to take a patch diagram and generate the necessary interconnections while utilizing as much of the system capacity as possible. Work done on the auto patch system alone at EAI has yielded only a 90% utilization rate to date. Certainly a better figure would be desirable.

In the switch array section, performance enhancement might be achieved by employing a fiber optics transmission line system for the multiplexed busses. This medium is capable of high speed digital signal transmission without the inductance effects of wire. For this application, it could result in reduced propagation delays and improved signal reliability.

Lastly, maturing microprocessor developments will lead to systems and devices that will become increasingly useful and cost and performance effective in the control of many of the elements of this machine. While their instruction fetch and execute times are still too slow for the arithmetic functions, they can provide a great deal of control flexibility.

LITERATURE CITED

[1]J. P. Landauer. "Program Generation System for Modern Hybrid Computers." Trans. of IMACS. 17 (Oct. 1975): 294-299.

[2]M. R. Cannon. "Magnitude and Time Scaling of State Variable Equations for Analog/Hybrid Computation." SIMULATION. 21 (July 1973): 23-28.

[3]C. Green, H. D. Hoop, and A. Debroux. "APACHE - A Breakthrough in Analog Computing." IRE Tran. Electron. Comp. EC-11 (Oct. 1962): 699-706.

[4]H. B. Rigas and D. J. Coombs. "PATCH: Analog Computer Patching from a Digital Simulation Language." IEEE Trans. on Comp. C-20, No. 10 (Oct. 1971): 1140-1146.

[5]J. F. Shoup and W. S. Adams. "A Practical Automatic Patching System for a Time-Shared Hybrid Computer." SIMULATION. 18 (April 1972): 142-148.

[6]G. Hannauer. "Automatic Patching for Analog and Hybrid Computers." SIMULATION. 12 (May 1969): 219-232.

[7]R. D. Benham and G. R. Taylor. "Interactive Simulation Language for Hybrid Computers." SIMULATION. 28 (Feb. 1977): 49-55.

[8]W. J. Karplus and R. A. Russell. "Increasing Digital Computer Efficiency with the Aid of Error-Correcting Analog Subroutines." IEEE Trans. on Comp. C-20, No. 8 (Aug. 1971): 831-837.

[9]G. A. Korn. "Back to Parallel Computation: Proposal for a Completely New On-Line Simulation using Standard Mini-Computers for Low Cost Multiprogramming." SIMULATION. 19 (Aug. 1972): 37-45.

[10]D. S. Dawoud and N. Z. El-Araby. "Parallel Digital Differential Analyzer with Arbitrary Stored Interconnections." IEEE Trans. on Comp. C-22, No. 1 (Jan. 1973): 41-46.

[11]A. V. Kalyayev. "Homogeneous Digital Analogous Structures with Programmable Commutation." Trans. of IMACS. 18 (Oct. 1976): 245-255.

[12]S. E. Scrupski. "Coming: Cheap, Powerful Computer Electronics." Electronics. 48 (Dec. 11, 1975): 67-68.

[13]"Micros can beat the IBM 370/168, News Update." Electronics. 49 (Feb. 5, 1976): 41-42.

[14]B. Wahlstrom and K. Juslin. "Simulation with Hard-Wired Analog Subprograms." SIMULATION. 28 (April 1977): 107-111.

[15]W. H. East. "A Completely Automatic Hybrid Computer." Data Processing Magazine. 7 (July 1965): 30-35.

[16]A. I. Rubin and D. H. Keene. "The Future of Hybrid Computation." SIMULATION. 25 (Aug. 1975): 55-59.

[17]R. E. Crosbie. "Simulation - Is it worth it." Survey Lecture at the 8th AICA Congress on Simulation of Systems. IMACS Newsletter. 18, No. 2 (April 1976): 5-6.

[18]R. E. Miller. "A Comparison of Some Theoretical Models of Parallel Computation." IEEE Trans. on Computers. C-22 (Aug. 1973): 710-717.

[19]J. D. Noe and G. J. Nutt. "Macro E-Nets for Representation of Parallel Systems." IEEE Trans. on Computers. C-22 (Aug. 1973): 718-727.

[20]J. B. Dennis and D. P. Misunas. "Preliminary Architecture for a Basic Data-Flow Processor." 2nd Annual Symposium on Computer Architecture Proceedings. (Jan. 1975): 126-132.

[21]W. A. Wulf and C. G. Bell. "C.mmp-A Multi-Mini-Processor." AFIPS Proceedings, FJCC, (Fall 1972): 765.

[22]B. R. Gilbert and M. J. Morse. "Digital Simulator Replaces Analog Portion of Hybrid Computer." Computer Design. 15, No. 4 (April 1976): 91-97.

[23]R. G. Arnold and E. W. Page. "A Hierarchical Restructurable Multi-Microprocessor Architecture." 3rd Annual Symposium on Computer Architecture Proceedings. (Jan. 1976).

[24]K. Janac. "Initial Condition Insertion Circuit for Hybrid Multiplexing Systems." Trans. of IMACS. 18 (April 1976): 109-114.

[25]Electronics Associates Inc. GSA Contract Data Processing Price List for Fiscal Year 1975. West Long Branch, N. J.: Electronics Associates Inc., 1975.

[26]F. J. Sansom and H. E. Petersen. "MIMIC Programming Manual." Aeronautical Systems Division, Wright-Patterson Air Force Base, Systems Engineering Group Tech Report SEG-TR-67-31, July 1967.

ACKNOWLEDGEMENTS

Special gratitude goes to my wife, Pat, for her patience in the trials and tribulations associated with a "student" husband who also holds a full-time job. Her reinforcement after countless weekends of work provided significant motivation in undertaking a major role in family matters. Lastly, and certainly not least, her constant orientation towards the Lord's will for our family and her help in my maintaining a proper perspective is to be praised.

The author also wishes to express his sincere appreciation to Dr. A. V. Pohm for his guidance during the graduate program and dissertation research. Beyond normal guidance, patience is required and was freely given for this project which was accomplished while on full-time employment with the Department of the Air Force. The educational value was greatly enhanced by his allowing the freedom of independent decision making.