GEORGIA INSTITUTE OF TECHNOLOGY OFFICE OF CONTRACT ADMINISTRATION

RESEARCH PROJECT INITIATION



Date: October 7, 1975

Sector Barrow

1 63.14

AL SCH BOOK

GP allow Cont

Project Title: An Instruction & Research Laboratory for Syndetic Digital-Analog Computation in Science & Engineering Education

Project No: G-36-615

Principal Investigator Mr. Alton P. Jensen

Sponsor: National Science Foundation

Agreement Period: From Until the site 1 P.

An and the second

Hidden.

Type Agreement: Grant No. EPP75-18315 and the second state of th

Amount: \$50,800 (Cost-sharing waived)

Reports Required: Final Substantive Report

Sponsor Contact Person (s):

HALL BELLEVIL

Y

1

Administrative Matters

(thru OCA) Mr. F. G. Naughten Grants Manager - Area 4 National Science Youndation Washington, D. C. 20550

(202) 632-5965

School of Information & Computer Science Assigned to: 1. 5 MA COPIES TO: Land the real sector sector

a saire

Principal Investigator School Director

Dean of the College Director, Research Administration

Director, Financial Affairs (2)

Security-Reports-Property Office Patent Coordinator

RA-3 (9-75) **新四次**4月

Library **第一日本日本** Rich Electronic Computer Center

×-135

184 F. 19.

Photographic Laboratory

Project File 7. . . C) 2 3 10 1 Other

and the farth of the

1.00

GEORGIA INSTITUTE OF TECHNOLOGY OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION

Nost 1

Date: 1/24/78

Project Title: An Instruction & Research Laboratory for Syndetic Digital-Analog Computation in Science & Engineering Education

Project No: G-36-615

Project Director: Dr. Alton P. Jensen

Sponsor: National Science Foundation

Effective Termination Date: 11/30/76 (Grant Expiration)

Clearance of Accounting Charges: N/A - All have cleared.

Grant/Contract Closeout Actions Remaining: None

Final Invoice and Closing Documents

Final Fiscal Report

Final Report of Inventions

Govt. Property Inventory & Related Certificate

Classified Material Certificate

Other

Assigned to: Information & Computer Science (School/Laboratory)

COPIES TO:

4 (3/76)

Project Director Division Chief (EES) School/Laboratory Director Dean/Director—EES Accounting Office Procurement Office Sécurity Coordinator (OCA)

Reports Coordinator (OCA)

Library, Technical Reports Section Office of Computing Services Director, Physical Plant EES Information Office Project File (OCA) Project Code (GTR1)

N 79. 4

Other

PROJECT NUMBER: ED75-18315

DATE AWARDED: September 1, 1975

PROJECT TITLE: "AN INSTRUCTION AND RESEARCH LABORATORY FOR SYNDETIC DIGITAL-ANALOG COMPUTATION IN SCIENCE AND ENGINEERING EDUCATION"

PROJECT DIRECTOR: A. P. Jensen

PROJECT ADDRESS: School of Information and Computer Science Georgia Institute of Technology Atlanta, Georgia 30332

PURPOSE:

The purpose of this project has been to study the feasibility of and the imperatives for a computing laboratory which unifies the concepts of analog and digital computing in information and computer science.

AUDIENCE:

The feasibility study of a Syndetic Computing Laboratory was intended to provide data and results which would permit programs in information and computer science education to determine whether or not an important area of emphasis is being neglected.

INNOVATION:

The School of Information and Computer Science of Georgia Institute of Technology has been supported by the National Science Foundation to investigate the feasibility of establishing an Instruction and Research Laboratory for Syndetic Digital-Analog Computation in Science and Engineering Education.

The impetus for the proposal which led to the funding of this effort came from a recognition within the School that information and computer science education has historically focused sharply on the technology of digital computers. With the explosive emergence of large scale digital computers, higher order language concepts, and the sophisticated integration of hardware and software concepts, information and computer science programs have had their hands full keeping pace with the development of an information technology which has become increasingly commited to digital or discrete data concepts. Educationally, this focus was appropriate and not particularly damaging in the early years of development. Further, the "damage" of effecting a program whose gross product was the illusion of a discrete digital world was ameliorated by the fact that a large percentage of the students seeking education

... in invitation and computer science came from a science and engineering background. But now, the picture is different; particularly different in programs of undergraduate education where there is a serious danger of producing a practitioner whose view of the world is a world of discrete and precise date; a practitioner who has not been exposed to notions which contrast accuracy with precision and who has not been exposed to the reactions which take place in continuous systems where data do not exist in a discrete, repeatable form except within the bounds of measurement error. Today's computer science graduate is not prone to question the randomness of a pseudo-random number generator which produces a fixed sequence of "random" numbers for a given "seeding." As repeatable as such a sequence is, it is said to be random enough. In short, today's computer science graduate, undergraduate or Ph.D., is apt to have been ingrained with the notion that bits exist in nature and they are not being taught to question and understand the nature of things which exist in nature as continuous functions. Since this education process ignores a large segment of reality in its focus on the discrete world of digital computers, there has been little need to teach or foster consideration of continuous functions in information and computer science. Such concepts have been left to engineering and the physical sciences where they must be confronted.

As the age-of-computers has emerged the basic concepts of computers have undergone little change. The papers of John Von Neuman are pertinent today. Little new computer technology has been developed since the 1950's. However, that technology has undergone a packaging and economic revolution; vast computer power has been placed in the hands of scientists, engineers, hobbyists, and tinkerers in the form of integrated circuits and things which can be built from them.

In view of this diffusion and dispersal of computer technology into the hands of the willing learners waiting eagerly to complete the cybernetic reality of having their very own expandable, controllable information engines as para-intellectual extensions or reflections of themselves, it seems imperative that Schools of Information and Computer Science must recognize the urgency of the challenge to produce graduates who have a facility not only with digital computers and digital processes but with computers in the perspective of information in its many dimensions and representations. Since information as it exists in real world processes is continuous in nature it is also natural that the newly dispersed technology must be conditioned to deal with it at its points of origin in the form in which it exists. This will require information and computer scientists who are prepared to construct or compose computerlike machines which solve specific problems.

What must these computer composers know? How can they be produced? Given existing programs in information and computer

computers? Or, is there a new role for the information and computer science major? If so, what must this information and computer science student know?

The innovative issuance of this project is the addressing of these questions via an extensive survey of organizations and individuals and a testing of the hypothesis that there is a new role for information and computer science graduates involving a "binding" of digital and analog processes called Syndetic Digital-Analog Computing.

EVALUATION:

In that this project is a feasibility study not intended to produce an end product other than perhaps a proposal, its ultimate evaluation will be effected through peer review. Thus far, presentations have been limited but favorably received.

MATERIALS:

As this is a feasibility study, only limited facilities have been developed to support testing the central hypothesis. Among these is a small microcomputer system based on a Motorola 6800 processor chip. This system (the NSF-6800) is designed to provide a low cost instructional vehicle which interfaces digital and analog systems and supports laboratory experiments through which students are able to assess the problems and tradeoffs of analog and digital processes. For instance, some laboratory experiments require comparisons of digital integration with analog integrations in which the results of standard digital routines are compared with the results of analog integration using both standard analog integrators and analog integrators built by the students from standard integrated circuits. These comparisons involve standard and arbitrary non-linear functions generated through the facilities of the NSF-6800. A number of such experiments have been developed and are undergoing testing prior to becoming an approved element of the curriculum.

PROBLEMS:

The nature of a feasibility study is that some apriori notions are to be assessed through some prescribed procedures. This study was committed to conducting a survey. The most serious operational problem encountered was the design, distribution and analysis of the survey data.

The most serious conceptual problem has been related to the generation of an objective staff willing to assess the issues and problems of Syndetic Digital-Analog computing without the polarities and predispositions inherent in the devotees of analog and digital systems who have so long sought to eliminate each other.

ADDITIONAL COMMENTS:

A survey of over 5,000 firms, organizations and individuals has been conducted. This survey sought to characterize the computing community (organizations and individuals) and sought rerequirements and computer science education. While responses continue to trickle in (with over 500 responses at this time) there appears to be a minuscule recognition of analog computing; there is a strongly felt need for more cost effective modeling; and, a strong implication that most computer science graduates are not being prepared to solve real world problems. Analysis of the survey responses has not been completed at this time.

The hypothesis that there is a role for information and computer science graduates involving a "binding" of digital and analog processes has been tested in an environment which requires the binding of analog and digital processes in the physical construction or composition of computer-like machines which solve very specific problems and which interface the domain of continuous information called the real world.

Information and computer science students from varying backgrounds (Music to Engineering) have been challenged with problems which require a unified application of analog and digital processes. These students have demonstrated that the modular synthesis of functions concept requisite to building higher order hardware functions or systems from lower order functional blocks can be mastered in both digital and analog areas. In fact, it appears that the enterprise of such composition employing standard units with fixed rules of use and application is a very productive way of illustrating the true nature of structured programming as it relates to software production and management. Hence this experience illustrates to ICS students the nature of product engineering and demands an orderly approach to the composition of higher and higher order modules which must be operating correctly before the next stage is initiated. It is somehow almost startling to most students that debugging or troubleshooting must proceed from the "known" rather than the "suspect."

Another observation is that the concept of true and instantaneous parallelism evidenced in analog models is a notion for which traditional information, and computer science students have little intuition. They may be equipped to understand and manage parallel sequential processes and still not comprehend or relate to naturally combinatorial circuits. Thus the true nature and importance of feedback as a dynamic function-production concept is not apparent to them. Through the study of amplifiers applied as summers, and integrators, this notion of feedback can be illustrated, appreciated, and exploited as information relevant phenomena.

It seems both imperative and obvious that people involved in handling information in the framework of the future must know about the tradeoffs and benefits of both analog and digital processes.

GEORGIA INSTITUTE OF TECHNOLOGY

ATLANTA, GEORGIA 30332

OFFICE OF THE DIRECTOR OF FINANCIAL AFFAIRS

September 20, 1977

Division of Grants and Contracts National Science Foundation Washington, D. C. 20550

Gentlemen:

Enclosed in triplicate is the final fiscal report for Grant Number EPP75-18315.

If you have any questions or desire additional information, please let us know.

Sincerely yours,

Evan Crosby // Associate Director of Financial Affairs

EC/bs enclosures:

cc: Dr. A. P. Jensen Dr. V. Slamecka Mr. E. E. Renfro Mr. A. H. Becker File G-36-615, G-36-318

Washington, D.C. 20550	DATION BUI	RESEA	RCH GR FISCAL	ANT REPORT		Form Budg	n Approved et Bureau No. 99-R0013
Ple	ase read instructions of	on reverse si	ide carefull	y before co	ompleti	ng this form.	
STITUTION AND ADDRESS		NSF PROG	RAM		GR	ANT PERIOD	10 5/31/76 *
Georgia Institute of Technology Technology & Syst Atlanta, Georgia GRANT NUMBER IMOS.) FPP75-18315 PRINCIPAL INVESTIGATOR IMOS.) Imos.)			Systems	REPORTING PERIOD from 9/1/75 to 8/31/77 ** R(S) GRANTEE ACCOUNT NUMBER G-36-615			
			GATOR(S)				
A SALARIES AND WAGES NSF Funded			Funded Mar	Months	NS	FAWARD	CUMULATIVE GRANT
1. Senior Personnel	Cal. Acad. Summ.		Summ,	BUDGET		EXPENDITURES Do Not Round	
a. 1 (Co)Princip	al Investigator(s)		5	Long at i	\$	9,253	
b. 1 Faculty Ass	ociates				121	6,375	
Sub-Total					\$	15,628	\$ 6,806.34
2. Other Personnel (N	Ion-Faculty)						
a. Research A:	ssociates-Postdoctoral						
b. Non-Faculty	y Professionals				÷		
c. 2 Graduate St	tudents					2,100	
d. Pre-Baccalay	ureate Students						
e.] Secretarial -	-Clerical					5,625	
f. Technical, S	shop, and Other		v	on 10			
TOTAL SA	LARIES AND WAGES				\$	23,353	\$ 23,467.73
B. STAFF BENEFITS I	F CHARGED AS DIREC	CT COST				2,048	1,708.23
C. TOTAL SALARIES, WAGES, AND STAFF BENEFITS (A + B)					2	25,401	\$ 25, 175.96
D. PERMANENT EQUIP	MENT						
							7 764 10
E. EXPENDABLE EQUI	PMENT AND SUPPLIE	s			-	2,700	7,764.13
F. TRAVEL 1. DOMESTIC (INCLUDING CANADA) 2. FOREIGN					5,000	1,2/2.91	
G. PUBLICATION COSTS						2,500	52,25
H. COMPUTER COSTS	F CHARGED AS DIRE	CT COST					
I. OTHER DIRECT COS	STS						
							/
J. TOTAL DIRECT COSTS (C through 1)					\$	35,601	\$ 34,265.25
K. INDIRECT COSTS	. 68	3% of \$2	3.467.73	3 =			
65% of Salarie	es & Wages	te ne cara	\$19	5.958.05		hilling and a solo	15 050 05
L. TOTAL COSTS (J plus K)					\$	<u>15,179</u> 50,780	15,958.05 s 50,223.30
L. TOTAL COSTS (J più							ALL ALE WORK SHE ALE THE AND
L. TOTAL COSTS (J più							Sugar Constraint Con Prost Prairie
M. AMOUNT OF THIS A	WARD (ROUNDED)				\$	50,800	
M. AMOUNT OF THIS A	WARD (ROUNDED)				5	50,800	
M. AMOUNT OF THIS A	WARD (ROUNDED)				<u>s</u>	50,800	
N. CUMULATIVE GRAM	WARD (ROUNDED) NT AMOUNT ANCE (N. BUDGET MIR				\$	50,800	s 576.70
L. TOTAL COSTS (J plu M. AMOUNT OF THIS # N. CUMULATIVE GRAM O. UNEXPENDED BAL EMARKS: Use extra sheet if r	WARD (ROUNDED) NT AMOUNT ANCE (N. BUDGET MI) necessary	NUS L. EXPR			5	50,800 FOR NSF	s 576.70 USE ONLY
L. TOTAL COSTS (J plu M. AMOUNT OF THIS # N. CUMULATIVE GRAM O. UNEXPENDED BAL EMARKS: Use extra sheet if r NO COST extension	WARD (ROUNDED) NT AMOUNT ANCE (N. BUDGET MIR necessary n till 11/30/76				\$	50,800 FOR NSF Final Fiscal	s 576.70 USE ONLY Report Accepted
L. TOTAL COSTS (J ph M. AMOUNT OF THIS # N. CUMULATIVE GRAM O. UNEXPENDED BAL EMARKS: Use extra sheet if r No cost extension * No obligations wa	WARD (ROUNDED) NT AMOUNT ANCE (N. BUDGET MIR necessary n till 11/30/76 ere. incurred ou	Granted	ENDITURE)	6	\$ \$ Gra	50,800 FOR NSF Final Fiscal I	\$ 576.70 USE ONLY Report Accepted
M. AMOUNT OF THIS / M. AMOUNT OF THIS / N. CUMULATIVE GRAM O. UNEXPENDED BALL EMARKS: Use extra sheet if r No cost extension * No obligations we of 9/1/75 through	WARD (NOUNDED) NT AMOUNT ANCE (N. BUDGET MIN necessary n till 11/30/76 ere. incurred ou h 11/30/76	vUs∟EXPR 5 granted tside ti	enditure) d 6/24/7 ne grant	76 period	\$ \$ Gra By	50,800 FOR NSF Final Fiscal I	s 576.70 USE ONLY Report Accepted _Remains Open Date
L. TOTAL COSTS (J ph M. AMOUNT OF THIS # N. CUMULATIVE GRAM O. UNEXPENDED BAL EMARKS: Use extra sheet if r No cost extension * No obligations we of 9/1/75 through	WARD (ROUNDED) NT AMOUNT ANCE (N. BUDGET MIR necessary n till 11/30/76 ere. incurred ou h 11/30/76.	vUs∟EXPR igranted itside ti	t 6/24/7 ne grant	76 period	\$ \$ Gra By Gra	50,800 FOR NSF Final Fiscal I ant Closed	s 576.70 USE ONLY Report Accepted _Remains Open Date tion Section, Area
L. TOTAL COSTS (J ph M. AMOUNT OF THIS / N. CUMULATIVE GRAM O. UNEXPENDED BAL/ EMARKS: Use extra sheet if r No cost extension * No obligations we of 9/1/75 through	WARD (ROUNDED) NT AMOUNT ANCE (N. BUDGET MIR necessary n till 11/30/76 ereincurred ou h 11/30/76. NVESTIGATOR	granted tside t	d 6/24/7 d 6/24/7 de grant	6 period	\$ Gra By Gra	50,800 FOR NSF Final Fiscal I ant Closed	S 576.70 USE ONLY Report Accepted _Remains Open Date tion Section, Area
L. TOTAL COSTS (J ph M. AMOUNT OF THIS / N. CUMULATIVE GRAM O. UNEXPENDED BAL/ EMARKS: Use extra sheet if r No cost extension * No obligations we of 9/1/75 through	WARD (ROUNDED) NT AMOUNT ANCE (N. BUDGET MIR necessary n till 11/30/76 ereincurred ou h 11/30/76. NVESTIGATOR	granted tside th TYPED C Alton	enditure) d 6/24/7 ne grant PR PRINTER P. Jens	76 period	\$ Gra Bv, Gra	50,800 FOR NSF Final Fiscal I ant Closed ants Administrat	S 576.70 USE ONLY Report Accepted _Remains Open Date tion Section, Area DATE 9/16/77
L. TOTAL COSTS (J ph M. AMOUNT OF THIS / N. CUMULATIVE GRAM O. UNEXPENDED BAL/ EMARKS: Use extra sheet if r No cost extension * No obligations we of 9/1/75 through GNATURE/OF PRINCIPAL IN I CERTIEY THAT A	WARD (NOUNDED) NT AMOUNT ANCE (N. BUDGET MIN necessary n till 11/30/76 ere. incurred ou h 11/30/76. NVESTIGATOR ALL EXPENDITURES F	granted granted tside th rypep c Alton EEPORTED	enditure) d 6/24/7 ne grant P. Jens ARE FOR A	76 period PAME	\$ Gra By, Gra	50,800 FOR NSF Final Fiscal I ant Closed ants Administrat	s 576.70 USE ONLY Report Accepted _Remains Open _Date tion Section, Area DATE DATE DATE MACCORDANCE
L. TOTAL COSTS (J ph M. AMOUNT OF THIS / N. CUMULATIVE GRAM O. UNEXPENDED BAL/ EMARKS: Use extra sheet if r No cost extension * No obligations we of 9/1/75 through GNATURE/OF PRINCIPAL IN I CERTIEY THAT A WITH TH	WARD (NOUNDED) NT AMOUNT ANCE (N. BUDGET MIN necessary n till 11/30/76 ere. incurred ou h 11/30/76. NVESTIGATOR ALL EXPENDITURES F HE AGREEMENTS SET	granted granted tside t TYPED C Alton FORTH IN	enditure) d 6/24/7 ne grant P. Jens ARE FOR A THE APPL	76 period Paname PPROPRIAT ICATION A	\$ Gra By, Gra TE PUR ND AW	50,800 FOR NSF Final Fiscal I ant Closed ants Administrat POSES AND IN ARD DOCUME	s 576.70 USE ONLY Report Accepted _Remains Open _Date tion Section, Area DATE DATE 0ATE VACCORDANCE NTS
M. AMOUNT OF THIS / M. AMOUNT OF THIS / N. CUMULATIVE GRAM O. UNEXPENDED BAL/ EMARKS: Use extra sheet if r No cost extension * No obligations we of 9/1/75 through GNATURE OF PRINCIPAL IN WITH TH GNATURE OF AUTHORIZED	WARD (NOUNDED) NT AMOUNT ANCE (N. BUDGET MIN necessary n till 11/30/76 ere incurred ou h 11/30/76. NVESTIGATOR ALL EXPENDITURES F HE AGREEMENTS SET D OFFICIAL	TYPED C Alton FORTH IN TYPED C Alton FORTH IN TYPED C Evan (ARE FOR A THE APPL Crosby,	PPROPRIAT CATION AT Associa	\$ Gra By, Gra ITE PUR ND AW, ITLE te Di 1 Aff	50,800 FOR NSF Final Fiscal I ant Closed ants Administrat POSES AND IN ARD DOCUME rector of airs	s 576.70 USE ONLY Report Accepted _Remains Open _Date _Date Date Date Date PJ/16/77 N ACCORDANCE NTS DATE 9/20/27
L. TOTAL COSTS (J ph M. AMOUNT OF THIS / N. CUMULATIVE GRAM O. UNEXPENDED BAL/ EMARKS: Use extra sheet if r No cost extension * No obligations wi of 9/1/75 through GNATURE OF PRINCIPAL IN WITH TH GNATURE OF AUTHORIZED	WARD (NOUNDED) WT AMOUNT ANCE (N. BUDGET MIN necessary n till 11/30/76 ere incurred ou h 11/30/76. NVESTIGATOR ALL EXPENDITURES F HE AGREEMENTS SET D OFFICIAL	TYPED C Alton FORTH IN TYPED C Alton FORTH IN TYPED C Evan (FOR NS	ARE FOR A THE APPL R PRINTED Crosby, F USE	76 period PPROPRIATICATION AN NAME & T Associa inancia	\$ Gra By Gra TE PUR ND AW. ITLE te Di 1 Aff	50,800 FOR NSF Final Fiscal I ant Closed ants Administrat POSES AND IN ARD DOCUME rector of airs Unex	s 576.70 USE ONLY Report Accepted _Remains Open _Date _Date ion Section, Area DATE 9/16/77 NACCORDANCE NTS DATE 9/20/27 pended
L. TOTAL COSTS (J ph M. AMOUNT OF THIS / N. CUMULATIVE GRAM O. UNEXPENDED BAL/ EMARKS: Use extra sheet if r NO COST extension * NO obligations wi of 9/1/75 through GNATURE OF PRINCIPAL IN I CERTIEY THAT A WITH TH GNATURE OF AUTHORIZED Drgan. Code F.Y. Fund D	WARD (ROUNDED) NT AMOUNT ANCE (N. BUDGET MIN necessary n till 11/30/76 ere. incurred ou h 11/30/76. NVESTIGATOR ALL EXPENDITURES R HE AGREEMENTS SET D OFFICIAL Frog. Code Ob. Class	Granted stride the stride the str	A CONSTRUCTION OF THE APPL CONSTRUCTION OF THE APPL COSSING THE APPL COSSI	6 period PPROPRIATION AN CATION AN CATION AN Associa inancia Amd, h	\$ Gra By Gra TE PUR ND AW. ITLE te Di 1 Aff nst. Cod	50,800 FOR NSF Final Fiscal I ant Closed Ints Administrat POSES AND IN ARD DOCUME rector of airs unex e Bala	S 576.70 USE ONLY Report Accepted _Remains Open _Date _Date DATE 9/16/77 NACCORDANCE NTS DATE 9/20/27 pended since Trans. Lot

PROJECT TITLE:

AN INSTRUCTIONAL AND RESEARCH LABORATORY FOR SYNDETIC ANALOG-DIGITAL COMPUTATION IN SCIENCE AND ENGINEERING EDUCATION

by

A. P. Jensen John Gehl David F. Kerzel Sinan Tumer Kate Covington Mac Van Nguyen P. C. Hankamer

PROJECT DIRECTOR: A. P. Jensen PROJECT NUMBER: ED75-18315 DATE AWARDED: September 1, 1975

PROJECT ADDRESS: SCHOOL OF INFORMATION AND COMPUTER SCIENCE GEORGIA INSTITUTE OF TECHNOLOGY ATLANTA, GEORGIA 30332

FINAL REPORT

September, 1977

CONTENTS

Foreword and Acknowledgmentsiii
1. SYNDETIC COMPUTATION: A Project Overview, with a Prognosis, Recommendations and a Definition of Syndetic Computing
2. STATE OF THE ART: A Perspective on the State of Analog/Hybrid Computing10
3. SOME TYPICAL STUDENT PROJECTS FOR A COURSE IN SYNDETIC COMPUTING:
A. Table Function Generator
B. Music Box
C. Integrator Comparisons101
4. INTRODUCTORY LABORATORY MANUAL FOR A COURSE IN SYNDETIC COMPUTATION
5. NOTES FROM THE REAL WORLD The Results of a Survey of the Data Processing Community
6. AN INSTRUCTIONAL AND RESEARCH FACILITY: Outline of a Proposed Facility for Syndetic Computation
7. THE EDUCATIONAL CHALLENGE OF SYNDETIC COMPUTING:

A Proposal to the National Science Foundation (Submitted to NSF under separate cover.)

Foreword and Acknowledgments

This project has the distinction of being among a few whose prosecution has led to a repudiation of most of the tenets which motivated the original proposal entitled, "An Instruction and Research Laboratory for Syndetic Digital-Analog Computation in Science and Engineering Education," submitted March 13, 1975.

The original proposal was based on a view of the computing economy which saw an increasing cost consciousness as the leverage on change. This view coupled with supportive evidence from the field of hybrid computing in its efforts to gain the momentum necessary to produce an Advanced Hybrid Computing System formed the premise on which a broad program of laboratory development was proposed.

As a consequence of that original proposal, this feasibility study was initiated. But the results of this study fail to strongly support the proposed laboratory development which would have involved a large-scale hybrid computer system and a whole new class of terminals necessary to its exploitation.

Therefore, this study has averted a possible mistake and has provided the opportunity to enquire into the dimensions of the problems and the virtue of more actively infusing notions of analog form information into existing programs of information and computer science. It has been shown that this infusion can be accomplished via a carefully structured laboratory program involving nominal costs. The development of such a program is proposed.

A number of people and organizations have contributed to the findings of this project in a number of ways. Among these are

iii

the following: Dr. Phillip H. Enslow of the School of Information and Computer Science at Georgia Tech who provided valuable background material on his survey of hybrid computing in Europe; Dr. Mel Corley of the School of Mechanical Engineering at Georgia Tech who generously loaned the project Comdyna Analog Computers and associated function generation equipment; Mr. Rubin and Mr. Mabry of EAI Inc. who provided the project a EAI-180 hybrid computer; Dr. Ronald Uhlig of the Department of the Army Research Command and various members of his staff who reviewed the survey form, provided mailing lists, and helped in many ways: Dr. Robert Howe of the University of Michigan and Dr. F. J. Ricci of the Defense Communications Agency who reviewed the survey form and made helpful suggestions on its presentation and content; Mr. Floyd Nixon of Martin Marietta of Orlando, Florida who gave support, guidance and criticism on request; Mr. Yoshikazu Fujiyoshi, president of Hitachi Shibaden Corporation of American and his technical staff who shared their views of the hybrid computer technology in a day-long session; Dr. Willard Fey of the School of Industrial and Systems Engineering of Georgia Tech who provided consulting, guidance and assistance in issues of Dynamo, modeling and simulation; Dr. Don Martin, chairman of Computer Science at North Carolina State University, who provided guidance in the issues of introducing analog computer programming in a non-engineering environment; Dr. Frank Stelmack of Union College of Schenectedy, New York who provided a forum in which various issues of analog/hybrid computing were reviewed; the many survey respondents who offered comments, guidance and criticism commensurate with a sense of professional responsibility, and the numerous students who enthusiastically (for the most part) accepted the challenge to consider some problems which were new to them and who provided valuable inputs to the project.

iv

1. SYNDETIC COMPUTATION: A Project Overview, with a Prognosis, Recommendations

The purpose of this project has been to study the feasibility of and the imperatives for a computing laboratory which unifies the concepts of analog and digital computing in information and computer science.

and a Definition of Syndetic Computing

The impetus for the proposal which led to the funding of this effort came from a recognition that information and computer science education has historically focused sharply on the technology of digital computers. Historically, the explosive emergence of large-scale digital computers, higher order language concepts, and the sophisticated integration of hardware and software concepts, have resulted in information and computer science programs having their hands full keeping pace with the development of an information technology which has become increasingly committed to digital or discrete data concepts. Educationally, this focus was appropriate and not particularly damaging in the early years of development. Further, the "damage" of effecting a program whose gross product was the illusion of a discrete digital world was ameliorated by the fact that a large percentage of the students seeking education in information and computer science came from a science and engineering background. But now, the picture is different; particularly different in programs of undergraduate education where there is a serious danger of producing a practitioner whose view of the world is a world of discrete and precise data; a practitioner who has not been exposed to notions which contrast accuracy with precision and who has not been exposed to the reactions which take place in continuous systems where data do not exist in a discrete, repeatable form except within the bounds of measurement error. Today's computer science graduate is not prone to question the randomness of a pseudo-random number generator which produces a fixed sequence of "random" numbers for a given "seeding." As repeatable as

such a sequence is, it is said to be random enough." In short, today's computer science graduate, undergraduate or Ph.D., is apt to have been ingrained with the notion that bits exist in nature and they are not being taught to question and understand the nature of things which exist in nature as continuous functions. Since this education process ignores a large segment of reality in its focus on the discrete world of digital computers, there has been little need to teach or foster consideration of continuous functions in information and computer science. Such concepts have been left to engineering and the physical sciences where they must be confronted.

As the age-of-computers has emerged the basic concepts of computers have undergone little change. The papers of John Von Neuman are pertinent today. Little new computer technology has been developed since the 1950's. However, that technology has undergone a packaging and economic revolution; vast computer power has been placed in the hands of scientists, engineers, hobbyists, and tinkerers in the form of integrated circuits and things which can be built from them.

In view of this diffusion and dispersal of computer technology into the hands of the willing learners waiting eagerly to complete the cybernetic reality of having their very own expandable, controllable information engines as para-intellectual extensions or reflections of themselves, it seems imperative that schools of Information and Computer Science must recognize the urgency of the challenge to produce graduates who have a facility not only with digital computers and digital processes but with computers in the perspective of information in its many dimensions and representations. Since information as it exists in real world processes is continuous in nature it is also natural that the newly dispersed technology must

In Harold Stone's very excellent "Introduction to Computer Architecture" one author applauds the repeatability of random number generators as facilitating "debugging simulation programs".

be conditioned to deal with it at its points of origin in the form in which it exists. This will require information and computer scientists who are prepared to construct or compose computer-like machines which solve specific problems.

What must these computer composers know? How can they be produced? Given existing programs in information and computer science what changes need to be effected? Must computer composers be engineers? Must everyone become facile in the compositions of computers? Or, is there a new role for the information and computer science major? If so, what must this information and computer science student know?

The innovative issuance of this project has been the addressing of these questions via an extensive survey of almost 5,000 organizations and individuals and a testing of the hypothesis that there is a new role for information and computer science graduates involving a "binding" of digital and analog processes called Syndetic Digital-Analog Computing. This survey sought to characterize the computing community (organizations and individuals) and sought reactions to issues of modeling economics, analog-hybrid computing requirements and computer science education. The survey (see section 5 of this report) revealed that there is a miniscule recognition of analog computing; there is a strongly felt need for more cost-effective modeling, and a strong implication that most computer science graduates are not being prepared to solve real world problems.

The hypothesis that there is a role for information and computer science graduates involving a "binding" of digital and analog processes has been tested in an environment which requires the binding of analog and digital processes in the physical construction of or composition of computer-like machines which solve very specific problems and which interface the domain of continuous information called the real world. (See section 2 of this report.)

Information and computer science students from varying backgrounds (Music to Engineering) have been challenged with problems which require a unified application of analog and digital processes. (See section 3 of the report.) These students have demonstrated that the modular synthesis of functions concept requisite to building higher order hardware functions or systems from lower order functional blocks can be mastered in both digital and analog areas. In fact, it appears that the enterprise of such composition employing standard units with fixed rules of use and application is a very productive way of illustrating the true nature of structured programming as it relates to software production and management. Hence this experience illustrates to ICS students the nature of product engineering and demands an orderly approach to the composition of higher and higher order modules which must be operating correctly before the next stage is initiated. It is somehow almost startling to most students that debugging or troubleshooting must proceed from the "known" rather than the "suspect."

Another observation is that the concept of true and instantaneous parallelism evidenced in analog models is a notion for which traditional information and computer science students have little intuition. They may be equipped to understand and manage parallel sequential processes and still not comprehend or relate to naturally combinatorial circuits. Thus the true nature and importance of feedback as a dynamic function-production concept is not apparent to them. Through the study of amplifiers applied as summers, and integrators, this notion of feedback can be illustrated, appreciated and exploited as information-relevant phenomena.

It seems both imperative and obvious that people involved in handling information in the framework of the future must know

about the tradeoffs and benefits of both analog and digital processes.

Some Conclusions

The experience gained from the present research effort clearly indicates that information and computer science students (both graduate and undergraduate) can:

- a. understand and apply analog computer programming with ease and to their surprise. Analog computers being comprised of functional standard units or blocks require a discipline in their use that underscores and emphasizes the analysis required in structured programming and thereby enriches the programming education process.
- b. understand and apply integrated circuits in the composition of small-scale digital computers and digital control processes. The process by which this is accomplished is similar to the process by which analog computers are programmed; i.e., there exists a finite collection of functional units (in this case, microprocessors are viewed as being higher-order primitives) with known input and output characteristics and the task is to effect an interconnection of these units in order to generate a higher-order functional unit which solves a problem. This problem-solving process also emphasizes the analysis required for structured programming and enriches the programming education process.
- c. understand and use current-day amplifiers and analog circuits to construct the basic functional components of analog computers: thereby making it possible to develop special-purpose analog functions for specific data acquisition environments and/or simulations.
- d. be taught syndetic computing concepts without the use of large-scale analog or hybrid computers. Small-scale analog computers costing \$1,500-\$2,000 are adequate for teaching analog programming concepts and tradeoffs; such systems available for augmenting teaching stations which

include LSI digital components can be used to illustrate the central issues of computer structures; parallel processing; data acquisition, managements and control, and process control.

However, it should be understood also that the results of the project's survey of opinion within the data processing community (see section 5 of the report):

- a. fail to support the notion that the cost of modeling simulation is a recognized problem at this time.
- b. indicate that the current investments in analog/hybrid computing facilities will be held essentially constant or tend toward being phased-out over the next five years.
- c. indicate that most modeling and simulation is being done in FORTRAN rather than in the simulation languages.
- d. failed to identify any real concern for the solution bandwidth limitations of current digital computers; hence, the principal motivation for hybrid computation seems to exist in isolated circumstances.

A Prognosis

The results of the survey combined with current trends in the computer-related technology strongly support the following view of the future:

a. The number of large-scale hybrid computer facilities will continue to get smaller. The rationale for this prediction is based on the observation that large-scale hybrid

computing is a highly specialized technology that has endured as a result of dramatic payouts in very specific cases. Such cases have been in areas where the computational bandwidth of conventional computers could not meet the strict time limits and economic considerations imposed by the problem area. In the future, more and more of the current hybrid computer applications will give way to cheaper and faster digital systems which can be dedicated to specific functions and subfunctions of existing problems. Digital differential analyzers, complex networks of pseudo-parallel processors and architectural innovations such as Denelcore and others continue to promise they will gradually replace conventional hybrid systems. This implies that there is a "moving boundary" beyond which only the specialists are operative in their quest for problem solutions which lie outside the time and economic bounds of digital systems.

Ъ. The need for a knowledge of analog techniques will increase with the continuing success of a large-scale integration and microprocessor technology. Here, too, is a moving boundary area. As LSI becomes more successful, it will be possible to solve more and more problems with less and less hardware units. With this trend, hardware solutions will move closer and closer to the data acquisition interface and will therefore be confronted with the problems of real time process control and the acquisition, storage and management of data. In other words, future systems will deal with real world problems in real world terms--this implies analog data (input and output) and strict economic constraints. Such systems will be computer-like, composed of MSI and LSI components, and will be restricted in function (tailored to the task at hand) rather than general purpose. In this setting the tradeoffs between cost and solution bandwidth will remain critical. For instance, in a problem setting in which several analog inputs must be summed and integrated and transmitted to a central computer, one might apply several analog to digital converters sampled under control of a microcomputer of bandwidth M, sample the data, sum it digitally, integrate it numerically and then transmit it. On the other hand, several A/D converters under control of a microcomputer of bandwidth M' might be used to sample the data which is then multiplexed by

the microcomputer and transmitted to the central computer. A third possibility would be to sum the analog signals, integrate the sum by an analog integrator, use one A/D converter controlled by a microprocessor of bandwidth M'' and then transmit the digital data to the central site. These three approaches would yield quite different costs in hardware, software, and complexity; but, by virtue of the analog data itself would yield equivalently accurate results.

While this example is related principally to an industrial setting in today's technology, such scenarios will become commonplace in the setting of personal computing. One needs only to review the last three months of Byte, Kiloband and similar popular periodicals devoted to hobbyists and personal computing to see that this reality is clearly emerging in the entertainment, speech synthesis, music production, data sensing and device control areas.

It is true that decreasing costs of digital components will continue to move the analog boundary closer and closer to the information source; but information persists in being continuous in nature and must be converted to digital data for digital processing. Relative to total system costs, A/D and D/A conversion will persist in being a significant factor for some time to come.

Récommendation

Information and Computer Science students should be exposed to both analog and digital computing concepts at the functioning component (LSI/MSI module) level. This exposure should begin early in an undergraduate program and should be maintained through a comprehensive laboratory program that spans the undergraduate ICS curriculum.

Graduate students in ICS who enter the program from other disciplines should be provided an experiential laboratory component

which provides insight into the hardware and firmware milieu of modern computers which exploit microelectronics in a variety of architectures.

Syndetic Computing Definition

SYNDETIC COMPUTING is defined as: The practice of binding together standard integrated circuits:

- a. in order to construct computer-like devices which process both analog and digital data;
- b. while effecting the methodology/time/cost tradeoffs necessary to a cost effective solution to a specific data processing problem.

2. STATE OF THE ART:

A Perspective on the State of Analog/Hybrid Computing

Since simulation is the substance of analog/hybrid computing, a perspective must consider the question, what are the common problems among simulations employing digital simulation languages and simulations employing analog computers? When this question is confronted, one immediately encounters the problem of function generation which has been the bane of analog simulation from the beginning; otherwise, the problem of expressing model solutions in analog form is fully as natural (or more so) as expressing these models in languages such as Dynamo or Mimic.

Among the problems encountered in confronting students with the issues examined in this paper is the distinction between "accuracy of solution" and the "precision of operations". The natural tendency is, of course, to translate precision and repeatability into illusions of accuracy as contrasted with the inherent accuracy of natural variability of physical systems.

Another aspect of analog computing that emerges is the direct association between functions and dedicated functional blocks. In such an environment one is always conscious of the physical limits of the system being employed; are there enough integrators? enough pots? etc.? In the perspective of large digital systems, there is always the illusion of plenty. If a model runs slowly or if turnaround is slow, it is natural to react with the notion that someone else is using too much of the system--whoever <u>they</u> may be.

One consequence of the endeavors associated with this section was a realization that in the instructional processes the model is the goal--not its use; therefore, very few models are run for any significant period of time or range of parameters. Under these conditions, the expense of digital processing is obscured by its flexibility and convenience. The tragically missing ingredient in the student's experience is the issue of cost. This issue could be carefully examined by requiring a cost analysis of a model of a given scope implemented in digital form on both a large computer system and a small dedicated computer system and then on an analog system but most classroom settings do not have access to such a range of facilities.

(In a syndetic computing setting the model can be translated into dedicated microelectronic componenets in such a way that the cost of each function could be clearly enunciated and tradeoffs effected against some nominal cost goal. On this basis complete life cycle costs for models can be illustrated easily.)

In the pages which follow, the current state of hybrid computing, digital and analog simulations are explored and presented. The main headings that are used to divide this treatment of the topic are:

- I. STATE OF THE ART IN ANALOG/HYBRID COMPUTATION
- II. THE APPLICATION AND COMPARISON OF TWO DIGITAL CONTINUOUS SIMULATION LANGUAGES
- III. REPORT OF ANALOG-DIGITAL COMPARISON USING AUTO SUSPENSION SYSTEM PROBLEM

This section of the report then concludes with an appendix and a bibliography.

INTRODUCTION

1. STATE OF THE ART IN ANALOG/HYBRID COMPUTATION

- I.1 Analog Computing
- I.2 Hybrid Computing
- I.3 Users and Applications of Hybrid Computing
- I.4 Languages for Digital Simulation
- I.5 Systems Software for Analog/Hybrid Computation
- I.6 Advanced Hybrid Computer System
- I.7 A Comparison of Digital and Hybrid Computing
- I.8 Hybrid Computing in Education
- II. THE APPLICATION AND COMPARISON OF TWO DIGITAL CONTINUOUS SIMULATION LANGUAGES
 - II.1 Introduction
 - II.2 General Description of MIMIC and DYNAMO
 - a) DYNAMO Language
 - b) MIMIC Language
 - II.3 Sample Runs
 - II.4 Analog Conversion
- III. REPORT OF ANALOG-DIGITAL COMPARISON USING AUTO SUSPENSION SYSTEM PROBLEM
 - III.1 Sample Problem Used
 - III.2 Selection of Simulation Technique
 - III.3 DYNAMO Model
 - a) Specifications of DYNAMO Program
 - b) Results of DYNAMO Simulation
 - III.4 Analog Simulation of Auto Suspension System
 - a) Mathematical Model of Auto Suspension System
 - b) Block Diagram of Auto Suspension System
 - c) Amplitude and Time Scaling
 - d) Patching and Static Checking
 - e) Computer Implementation and Output
 - III.5 Comparison Between Digital and Analog Simulation Outputs
 - III.6 Comparison Between Analog and Digital Simulation

APPENDIX

BIBLIOGRAPHY

INTRODUCTION

Today's computer practitioner probably perceives the real world as being measurable in discrete values, and this illusion of a digital world surely influences his commitment to strictly digital problem solving. The domination of digital computation among computer professionals and users is attributed to the development of higher level languages and business users' dependency on the digital computer, and is evidenced by the information and computer science curriculums of educational institutions.

Many computation problems are involved with the simulation or control of physical systems. It has been said that (Davison):

"...nature behaves in an analog fashion. The variables we measure, filter, and record are analog variables; chemical reactors, heat exchangers, and other dynamic physical systems simply do not behave in a discrete manner."

So it seems paradoxical that today many analog problems are being solved in a strictly digital discrete data environment.

For many computer professionals the understanding of analog computing is limited to a few concepts which contrast digital and analog computing. A synthesis of digital and analog concepts, rather that an antithesis of these ideas, would be appropriate for solving many problems which are neither strictly analog nor strictly digital in nature.

This paper takes a state-of-the-art look at analog/hybrid computing. Particular attention is paid to the potential of an advanced hybrid computing system, the applications of analog/hybrid computing, and the role of analog/ hybrid computing in education.

Two simulation projects that have been completed recently at Georgia . Institute of Technology are described here. One problem compared the modeling

of systems in two different digital simulation languages. The two models, the Retail Store Inventory System and the Health Care Delivery Systems of the USA, were each simulated in MIMIC and DYNAMO. MIMIC and DYNAMO are described in detail and the possibilities of implementing the two models on an analog computer are investigated.

The second problem was a simulation of an automobile suspension system. The system was simulated on a digital computer using the DYNAMO simulation language and the preparation of the model for computation is described in the report. The system was also simulated on an analog computer available at Georgia Tech; preparation steps included drawing the block diagram, amplitude and time scaling, patching and static checking. It was found that the relevant information about the system could be found from both simulations. Both techniques were able to handle the differential equations that describe the model; but the digital computer was more precise while the analog computer offered continuous graphic output.

I.1 ANALOG COMPUTING

Analog computers represent numbers by corresponding physical quantities such as continuously varying voltages or shaft rotations and these representations can be translations of physical conditions such as flow, pressure, temperature, or voltage. Each step of a computation is performed by a separate unit or component (such as an integrator or summer) and the connected units perform their operations simultaneously. The continuous nature of quantity representation allows the values and parameters of a problem to be altered easily, while the simultaneous operation of the components effects a fast solution time. The ability to vary quantities easily and to quickly see the result of the alterations, often in a graphical form, enables a sort of intimate relationship between the user and the problem solving. Analog computing has important uses in solving certain classes of problems that involve differential equations and in accompanying real-time system simulation.

Analog computing is often defined by comparing and contrasting it with digital computing. In a digital computer numerical representation is discrete and operations are performed serially. A complex problem will probably require a longer time to solve by the digital computer but does not restrict the solution time of the analog computer. In the past, analog computers have been difficult to program because of scaling and patching operations, and because of a lack of good debugging aids. Scaling relates the problem variables to the physical system variables and scale factors are selected on the basis of the calculated or estimated maximum values of the problem variables. Patching is physically connecting the analog computer's components by means of patch cords and according to a circuit diagram representative of the problem to be solved. A variety of high languages makes the programming of a digital computer easier and quicker.

Analog computation is often less precise in its calculations because of the scaling problem, the performance of the electronic elements, and the lack of repeatability (the ability to repeat the problem and get the exact same result.) Yet because of the variability in numerical representation and calculation, the analog computer may be considered as giving a more accurate picture of a real world situation; this is an important consideration in simulation studies. Another difference between analog and digital computation is that analog computers are available to only one user for the duration of his problem solving.

I.2 HYBRID COMPUTING

Analog computers offer the advantages of speed, parallel operation, ease of altering parameters, and cheaper processing cost while digital computers offer the advantages of long-term storage capacity, greater precision, ease of programming, good debugging aids, and repeatability in problem solving.

A hybrid computer, which combines analog and digital computing, has been a means of utilizing some of the advantages offered by both. Some writers have considered an analog computer with digital logic or a digital computer with analog elements as being hybrid computers, but traditionally the hybrid computer has consisted of an analog computer and a digital computer linked through an interface system (Vemuri).



FIG. 1. HYBRID COMPUTER

In a program, such as a simulation, the analog computer acts as a subroutine in a digitally oriented computer program. The digital computer assigns the control and the analog circuitry acts as auxiliary equipment (ibid). The linkage or interface performs the analog to digital and the digital to analog conversions of data representation and generates the control signals to keep both machines synchronized.

One of the first attempts at a hybrid computer was in 1955 when engineers at the Space Technology Labs and Convair tried to combine the analog and digital computers. They wanted to simulate the high frequency dynamics of the Atlas missile (ICBM) on the analog and the long-term integrations of the trajectory on the digital computer (<u>Advanced</u>). This required the building of interfaces - analog to digital converters and digital to analog converters.

In the decade of the sixties, the development of digital computing received widespread encouragement and funding. Much less effort was spent on analog research and development. Analog computing has not reached the state of development that digital computing has attained. Business-oriented requirements have encouraged digital technology, research and development. Digital computers have also been popular in the academic world as can be seen by the use of digital computing in science and technology courses and by the emphasis on digital computing in computer science curriculums. The digital manufacturers have, through educational programs, increased public awareness, interest, and confidence in digital computing as the means of problem solving.

The public is well aware of the role the digital computer played in the U.S. Space program. The concept of time-sharing has enabled many people to use the digital computer. The development of higher level languages has facilitated programming and made digital computing accessible to a large number

of students, business people, scientists, and engineers. The large computer manufacturers, like IBM, Control Data, and Burroughs, have done so well that their accumulated resources enable the continued growth of their industry (Advanced....).

I.3 USERS AND APPLICATIONS OF HYBRID COMPUTING

It is estimated that half of the analog/hybrid installations are used by people in education. Other users are in the fields of aerospace, chemicals, transportation, and utilities. The aerospace industry needs analog/hybrid computing for aircraft adaptive control systems, terrain avoidance systems, and space vehicle simulations. Hybrid computers have been used in the Skylab and the Space Shuttle space programs and for the support of aircraft design. A large airline has shown that the digital computer alone is simply not fast enough for some of the required calculation of input, much of which is in analog format.

Analog computers are used for measurements by the chemical process industry in chemical reactors and heat exchangers and by the bioengineers in EKG and EEG data analysis studies. For example, in the hydrocarbon processing industry, analog/hybrid computing is used for determining unknown parameters in process dynamics, process control, and process optimization.

In the field of communication, wave propagation studies and antenna pattern calculations use analog computing. Electronic music studios depend on analog signal-generating devices under digital control, but interfacing the composer to the hybrid system has been a problem because of the difficulty in defining a programming language to describe musical events. Presently,

. 18

a composer is required to specify in the program all the analog components needed to generate the desired sonic effect and to specify the temporal coordinates to locate the event in time.

The main application field of hybrid systems is the simulation of continuous systems. Simulating systems that are characterized by ordinary differential equations can be done efficiently with analog computing. Simulation is the construction of a model which assumes or has the appearance and/or behavior of the system or device without the reality. For a wide gamut of problem situations, an analog computer is capable of real-time solution speeds; that is, it can respond to inputs and produce outputs at the same speed as would the actual system. Analog computing also allows the possibility of on-line monitoring of the system behavior. Because of the speed advantage, the user can immediately modify his simulation on the basis of an observed solution. Certain groupings of analog components are associated with corresponding blocks of the physical system being studied. This provides greater insight into the problem solving. Hybrid computing is an effective and efficient medium in which to solve certain types of problems and simulations such as these (Stephenson):

- studying dynamic systems that are described by simultaneous differential equations having widely differing parameters.
- sets of simultaneous differential equations where the parameters are varied from one solution to the next

3. systems with discretely and continuously changing variables

As an example, analog/hybrid computing has aided the simulation of the seeker, guidance and control systems of missiles, their pre-flight and post-flight, and their flight safety.

A person who wishes to use hybrid computing for a simulation or to solve

- convert the mathematical model of the physical problem to a schematic diagram showing how the analog components are to be connected with patchcords
- determine the appropriate scale factors to make sure none of the components will exceed its voltage range
- 3. compute potentiometer settings or digital coefficient unit settings
- 4. assign the necessary components
- 5. write the digital program
- 6. write the necessary interface program to transfer data between the digital and analog computers.

The credibility of a simulation depends on the validity and the utility of the results. A simulation run may return very precise data which is repeatable. But this may not be an accurate description of an unrepeatable, variable real world situation, and therefore the results would not be truly valid. Since a digital simulation often cannot give an accurate picture, it's usefulness as an imitator of a real world problem should be considered by a decision maker who is dependent on the simulation run's results and who needs an accurate set of data which may not be repeatably precise.

1.4 LANGUAGES FOR DIGITAL SIMULATION

The advantages of analog computing for certain types of problem solving are recognized but the difficulty in programming these problems on the analog computer has encouraged the development of digital computer programming languages which simulate an analog computer for the user. The language construction includes predefined functional "blocks", each of which functions as an analog component such as a summer, an integrator, or a function generator. The user

interconnects these blocks and defines the initial conditions and parameters to be associated with each block. The blocks are considered to be executed in parallel whereas the control statements are in sequential execution. In effect, this is a simulation of a hybrid system where the control and set-up would be done by the digital part and the actual problem solving done by the analog part of the system.

Such simulation languages include the following:

MIDAS	developed by Martin Marietta Modified Integration Digital Analog Simulation
DYNAMO	Developed for the IBM 360/370 series
MIMIC	a recent version of MIDAS that is FORTRAN-oriented
CSMP	Continuous System Modeling Program; FORTRAN-oriented
CSSL	based on MIMIC

The digital languages which provide for analog simulation have provided good target languages for hybrid systems software.

I.5 SYSTEMS SOFTWARE FOR ANALOG/HYBRID COMPUTATION

An important early development in hybrid software was a digital computer program to automate the programming of the analog computer; this program was APACHE. APACHE was developed at Euratom in Italy and was successfully used for simulations that were run on the EAI PACE 23IR analog computer. It used a FORTRAN-like set of statements as an input to a compiler whose output was a patching list for the analog computer. Hand-patching was still required of the programmer. APACHE never gained great acceptance because of the program's machine dependence but it was a source of ideas for later development.

APSE (Automatic Programming and Scaling of Equations) is a simplified version of APACHE. This FORTRAN-like program is designed to receive as input

the differential equations which define the problem and the data to define the calculation of scale factors, constants, and the initial problem conditions. APSE reduces the equations to a form more suitable for solution on the analog computer. APSE also scales the dependent and independent variables, performs a static check on the equations, and assigns the analog components required for the problem solution. The user receives from APSE a listing of the component allocations, a listing of potentiometer values, and a table of static check values. Hand-patching is still required of the programmer. APSE cannot handle multivariable functions and it requires a large computer for running. Because it allows for the worst possible combinations of variables, its scaling is often too conservative. Its advantages are the automatic documentation and the automatic static checks.

Another hybrid compiler is ACTRAN, which was developed under the direction of P.E. Rook in England. ACTRAN uses S/360 CSMP as its input language and also obtains a digital check solution which is used for scaling. It is written in FORTRAN and was intended to provide more complete support than APSE. HIFIPS (Hybrid Formula Interpreting and Programming System) was developed in the Netherlands. Its objectives were to eliminate all hardware dependent terms and allow the writing of algorithms that represented parallel, simultaneous simulations. HIFIPS and ACTRAN, which are similar systems, have been combined as HCS1. HCS1 is intended to be relatively machine independent.

PATCH, a compiler developed at Washington State University, is CSMP-like and demonstrates the extendability of a high level simulation such as CSMP to hybrid compilation (Rigas). The compiler outputs the patching information and potentiometer settings. PATCH is not tied to a specific manufacturer's computer.

HYTRAN, developed by EAI, is basically a static-checking and documentation system. EAI developed it and extended it into what is known as HOI, Hytran

Operations Interpreter. It is an interpreter which acts directly on source statements and is therefore highly interactive but rather slow. It provides four operations: problem definition, consistency check, static check, and run-time supervision. Because of some obscure codes and commands, it has been difficult to use and to comprehend.

1.6 ADVANCED HYBRID COMPUTER SYSTEM

Hybrid computing problems remain. A user needs to know a great deal about hybrid programming in the program writing. The programmer is still required to wire the patchboard and set the scale factors. There is no means for sharing the analog computer among multiple users and analog-todigital and digital-to-analog conversions are expensive and degrade the Precision of the system. In the 1960's, the U.S. Army Materiel Command (AMC) formed a working group to evaluate the technology in hybrid computation and to formulate plans for an Advanced Hybrid Computer System which would fuse into one system the best features of the analog and digital modes of computing and add features unique to hybrid computing. The capability of sending analog computer signals over phone lines, the development of high-level analog programming languages, and the developments in solid-state switching devices have encouraged these ideas and plans.

The proposed AHCS will have an electronic switch matrix for automatic patching of the computing elements, a simplified programming language which could be used by non-hybrid personnel, digitally controlled scaling, and a time-shared system with remote terminal access. It was thought that the proposed AHCS would have speed capabilities of over 600 million operations a second (Saucier). The AHCS configuration looks like this (Mawson):



FIG. 2

ADVANCED HYBRID COMPUTING SYSTEM
Two ADC (analog to digital conversion) systems, micro-computers and associated mass storage devices are to be used to convert analog solutions to digital form. The digital portion of the system would perform the auto-patching and the scaling and other duties now performed by digital computer software systems, such as compiling, storing programs, scheduling, and communicating with the remote terminals.

A compiler which will translate the user's program into the necessary switch codes to interconnect the analog components, set the digital coefficient units, and set the diode function generators, in short set up the analog problem, is needed (Warshawsky, Howe). The object is to bring hybrid programming to the level faced by a digital computer user when he employs a simulation language such as MIMIC or CSMP. Some hybrid system software has been developed though none is extensive enough to support the AMC's proposed AHCS.

If the AHCS can accomplish automatic patching, eliminate the manual interaction, provide for multiple and remote users, and construct feasible software, then the payoffs will be reduced computing cost and increased computer power.

1.7 A COMPARISON OF DIGITAL AND HYBRID COMPUTING

Simulating an analog computation on a digital computer does not take advantage of the speed and cost benefits of the analog computer. The U.S. Army Materiel Command developed a quantitative measure, a mathematical representation of the performance characteristics of the computer, to be used to project computing requirements for solving certain types of problems. The WSP (Wolin-Saucier-Peak) Scientific Mix was given representative problems from governmental and industrial labs and it measured the computation time

needed to solve the problems with both present-day digital and hybrid computing facilities. The following examples demonstrate the findings (Wolin):

		Computer time in minutes				
		Analog	Digital			
AEC	reactor core dynamics;	10	200			
	200 diff. equations	(EAI 8900)	(IBM 360/75)			
Army	vehicle traversing a	1	15			
	cross country terrain	(EAI 231R)	(IBM 360/44)			
Navy	real-time torpedo	10	cannot be run			
	launch simulation	(EAI 8800)	digitally			

The speed advantage of analog computing is evident. The WSP results were verified by utilizing the simulation experiences of various governmental and industrial labs on digital and hybrid computers; the actual computation times were compared with the estimated times and the results were favorable (Wolin).

Another comparison of digital and hybrid computing was made by a group in England. The problem was to interatively solve a large set of first-order differential equations which represented the mass balance of a commercial crude oil distillation unit. The equations were solved on the hybrid computer in the Cambridge Control Engineering Group and on an IBM 370 digital computer for comparison (Stojak). The hybrid computer was composed of a PACE 231R - V linked to an ICL/Elliot 4130. The hybrid computer scheme demonstrated the faster solution time but the solution time on the IBM 370 was only marginally slower. This seemed to demonstrate that digital integration on a digital computer comparable to the IBM 370 can be very close to the solution time of hybrid computation. Approximately two man-weeks were required to successfully implement the CSMP package on the 370 while about three man-months were needed to complete the hybrid scheme.

These comparisons bring to light some important points that must be considered by a programmer who has to decide whether to use digital or hybrid computing for solving a particular problem. The computer run-time in hybrid computation is faster than that of digital computation. But the program preparation time for the hybrid is usually much longer. If a problem is to be run many times, then the hybrid computer is the best choice, but if a problem is to be run only a few times, then the digital solution is cheaper. Changing parameters is simpler and quicker on the hybrid but changing the schematic configuration is time-consuming with hybrid computation. It must be remembered that present hybrid software demands that the programmers handle the actual analog component connections manually with patchcords. The programmer must also decide on the tolerance range that he wishes for the results. Digital solutions show greater precision and repeatability. A final consideration for the programmer would be the applicability of hybrid or digital computation to his problem.

I.8 HYBRID COMPUTING IN EDUCATION

Demonstrating the feasibility of an alternative to the digital computer for problem solving will need to occur in educational institutions. Students should be shown how to evaluate and find the most expedient means for getting solutions to problems. Educating the user community in the use of hybrid computing can be done practically by graduating students. Also, communicating to the user public evidence of how hybrid computers have 'paid off' in some projects will help to create credibility for and confidence in the technique of hybrid computing.

Presently, few computer science and engineering core curriculums include more than one course in analog/hybrid computing. Many computer science depart-

ments do not have any analog/hybrid equipment. Often students who graduate with computer science degrees have not used or even seen an analog computer. The student's approach toward saving time and money in problem solving depends on improving the digital program rather than on finding an alternative means of solution. There is a lack of awareness by the student of the role that analog/hybrid computers can play in problem solving and simulation.

Some schools have acknowledged that the use of an analog computer in coursework can enhance a student's understanding of physical phenomena - in engineering, medicine, chemistry, economics, etc. North Carolina State University has installed 16 analog terminals that can be used by students who have no knowledge of analog programming. The terminals connect to a number of pre-patched problems and allow the student to vary up to 8 parameters in a specific simulation. The student's output is displayed graphically on a terminal oscilloscope.

The University of Michigan has been using small analog computers for teaching courses in dynamics, automatic control, and differential equations. An AD-4 analog system and a PDP-9 digital system comprise the hybrid computer. The student works at a terminal and has direct control of problem parameters and can see the effect of parameter changes immediately and graphically on the terminal storage oscilloscope. Each problem must be patched by the instructor. A switch matrix that could be digitally controlled was installed; a simple compiler for the PDP-9 permitted the programmer to enter the program in simulation language and the PDP-9 assigned the analog modules and developed the autopatch switch code (Fogarty, Howe). This software system is known as Autopatch I. The instructor must still scale the problem, but the hybrid computer can be used by students who do not know anything about analog programming.

The University of Michigan hybrid system is important because, in a small way, it demonstrates a solution to several problems the AHCS is attempting to solve - automatic analog component patching and hybrid computer time-sharing. A solid switch matrix of 768 switches is used to interconnect 8 analog integrators, 6 multipliers, and 32 coefficient devices. It allows analog circuits for solving differential equations to be electronically patched and completely set up in less than 20 milliseconds under digital control. This prototype system is small and does not include such components as function generators, comparators, and parallel logic elements. The main functions of the digital computer are to compile the analog patching instructions, cause the autopatching and coefficient setting to occur and handle the time-sharing of the five graphic terminals.

It is hoped that other colleges and universities will undertake instruction of and the means for a wider gamut of problem solving and simulation methods. An aim of information and computer science departments should be to instill in future computer practitioners a more synergetic view of analog and digital computing. It is hoped that future professionals will not view the real world as being a discrete digital world. It could be envisioned that a practitioner who is called upon to solve a problem might not immediately begin coding a digital program but rather think of a problem solution that might involve digital and analog components which could be hardwired. Because of the economic availability and the convenient applicability of integrated circuit chips, this concept of problem solving seems feasible and practical.

The necessity of using computers in problem solving environments is evident. Using the best means of computing and finding the fastest, cheapest, and most efficient means is the challenge that needs to be met. The capability for choice of computing instrument is the beginning of meeting that challenge.

II. THE APPLICATION AND COMPARISON OF TWO DIGITAL CONTINUOUS SIMULATION LANGUAGES

II.2 INTRODUCTION:

With the flexibility and power introduced by digital computers, compilers, interpreters and translators can be written to simulate analog/ hybrid models in digital computers. MIMIC and MIDAS are two widely known languages which can solve systems of differential equations and state equations respectively. DYNAMO, which simulates dynamic feedback models of business, economic, and social systems, is a language for translating and running continuous models which are represented by difference equations and simple linear first order differential equations.

Of the several languages available to solve systems of ordinary differential equations, two of them were selected for comparative analysis: MIMIC and DYNAMO. The "Retail Store Inventory System" and the "Health Care Delivery Systems of USA" were two sample models involved in the analysis. The main goal of the analysis was to search for the possible difficulties and problems an analyst may encounter during the stage of patching the analog computer from the model simulated in the digital computer.

II.2 GENERAL DESCRIPTION OF MIMIC AND DYNAMO:

a) DYNAMO Language.

DYNAMO was developed by the industrial dynamics group at M.I.T. for simulating dynamic feedback models. It has been designed for the person who is problem-oriented rather than computer-oriented.

The DYNAMO language includes time subscripts that describe a very simple integration scheme (rectangular or Euler's method) which is also very efficient when great precision is not critical.

DYNAMO was primarily designed to solve continuous models which are constructed by aggregate flows. The basic tool of continuous simulation is integration (accumulation), which is the process that relates a quantity to its time rate of change. We can indicate the present time by the subscript K, the earlier time by the subscript J. The elapsed time between J and K is called DT, and we can write the equation:

Quantity \cdot K = Quantity \cdot J + DT * rate of change

In DYNAMO an integral cannot be computed by calculating the above expression once. The interval DT must be computed by dividing the hour into infinitely many small intervals and repeating the calculation for each infinitesimal interval. Unfortunately, the digital computer cannot integrate accurately and the actual interval must be approximated by some other method that works very well when great precision is not required. This method is to compute the rate of change at time J and assume that it is constant over the interval J to K. The accuracy can be controlled by the choice of the size of DT.

Flows of goods into an inventory, men into the labor force, water into a tank, and current into a capacitor are typical examples of flows. These flows are called rates and are the entities moving from one place to another without being created or destroyed in the process.

The basic structure of a DYNAMO model consists of a number of reservoirs, or levels, interconnected by flow paths. The rates of flow are controlled by decision functions that depend upon the condition in the system. The levels represent the accumulation of various entities in the system such as inventories of goods, unfilled orders, number of employees, gallons of water, etc. The current value of accumulation represents the aggregated (integrated) difference

between the input and the output flow for that level. The instantaneous flow between these levels is represented by rates while the dependency of flow rates on levels is controlled by decision functions.

Mathematically, a rate is represented by the derivative of a variable. Since a DYNAMO model relates levels to rates, it follows that this model is actually a set of linear differential equations. Usually, the model is constructed from simple linear differential equations with constant coefficents.

Representation of time in a DYNAMO model is achieved by the name TIME which is a reserved word. The instant at which the present calculations are being made is referred to as TIME \cdot K. The previous instant at which calculations were made is TIME \cdot J and the next instant following TIME \cdot K is TIME \cdot L. The interval just passed is called the JK interval, and the interval coming up is the KL interval. Since the calculations are made for uniform intervals of time the JK and KL intervals are always the same size. The symbol DT discussed in the above paragraphs designates this time interval and the magnitude of the interval is selected by the user.

The program evaluates the values of all variables at successive intervals of DT time units. If DT is too large, the results become inaccurate because they do not follow accurately the continuous interaction between the variables.

As the magnitude of DT gets smaller, the solution accuracy is increased at the expense of increasing the amount of computer time used. No definite method can be introduced for determining the tradeoff between these two factors. It is conventional to try different values of DT to determine whether results are significantly affected by the choice of value. After a certain point, decrementing the value of DT will not effect the precision at the expense of computer time.

Because of the existence of delays in the system, changes do not occur simultaneously. It is necessary to determine the length of the DT interval according to the magnitude of delays. It is customary to select DT no larger than half of the smallest delay length in the system.

It is not always possible to define the model solely in terms of levels and rates. It may be necessary to introduce auxiliary variables, table functions, macros and intrinsic functions. These digital computer features bring powerful capabilities for solving and evaluating the continuous dynamic systems.

b) MIMIC Language

The MIMIC language was designed to facilitate solving systems of ordinary differential equations on a digital computer. The MIMIC language was structured to accept the mathematical description of the problem as data which can be developed from a block diagram.

The important feature of MIMIC language is its parallelism. Because of this distinctive attribute of the language, MIMIC is referred to as a digitalanalog simulator or a continuous systems simulator. Therefore, statements defining a MIMIC program can be written in any order since the system is evaluated in parallel. The sorting algorithm available provides the parallelism which is contradictory to the more general digital programming languages.

MIMIC language, contrary to DYNAMO language, is mathematically oriented; therefore a MIMIC program can be made to resemble closely the original problem statement. Instead of formulating the concept of the problem as in DYNAMO, the user must develop the mathematical formulation of this model before writing the MIMIC program. The format of the MIMIC language facilitates entering the mathematical formulation as program statements.

A variable step integration routine is used to perform all integrations. The magnitude of the step size is out of the control of the user and it is internally provided by the system. This step size is automatically varied by the requirements of the problem to insure that the integration error does not exceed a given bound at any one time.

A set of arithmetic operators and a set of functions are provided by the MIMIC language. Also, the MIMIC language contains all the flexibilities and capabilities of a digital computer language by providing table functions, subroutine features, intrinsic functions and logical switches.

Time increment is continuous in the system and it is achieved by the MIMIC program internally. In practice the time step is very small relative to DT in the DYNAMO program. The MIMIC language lacks the facility to refer to the variables that occurred during the previous time step, because all the variables are calculated continuously with very small time steps starting from time zero.

II.3 SAMPLE RUNS:

At the present time two languages are available on the CYBER-70 at Georgia Tech to represent and evaluate the behavior of dynamic systems: DYNAMO and MIMIC. Two sample models have been selected and programmed by using these two languages.

One of the examples was a rudimentary and straightforward model of a "Retail Sector". This model is similar to an ordinary production distribution system. The customer, who is exogenous to the model, places orders upon the retail sector for an aggregate product. These customer orders reside in a "pool" of unfilled orders until they are filled from inventory. The retail sector orders replacements for the items sold and corrects the inventory to the

desired level, which is several weeks of average sales. These orders from retail are filled after a fixed delay.

One very simple subsystem consists of the flow of customer orders into the retail backlog. The other subsystem consists of the flow of goods from the distributor to the retail inventory and then on to the customer. The retail sector model is illustrated in Figure 3.

The computer listings in the appendix display two different formulations of the Retail Sector Model that use the languages mentioned above. For this trivial problem, the difference between the outputs of the two different codings is negligible and it can be attributed to the selection of DT and the difference in the methods of integration.

The second sample model was the simulation of Health Care Delivery System of the U.S. The original model was constructed within the limits of DYNAMO formulation. The programs written in MIMIC language for the same model created some problems during the execution. This is not an indication of superiority of DYNAMO over MIMIC, because the original model was biased toward DYNAMO formulation.

The Health Care Delivery System Model of the U.S. is illustrated in Figure 4. The computer listings (in the appendix) depict both DYNAMO and MIMIC codings of the model. In the DYNAMO program narrative explanation of each variable is given.

The MIMIC version of the model gave ambiguous results. The percent of increase in some of the major variables was estimated by taking the derivatives with the intrinsic routine available in MIMIC. The derivative is suppose to calculate the slope of change, but after the initial time point, it did not evaluate the rest of the values for following time points. When the derivatives



- IAR Inventory Actual at Retail
- SSR Shipments Sent From Retail
- IDR Inventory Desired at Retail
- DFR Delay Filling Orders

were removed from the program and replaced by difference calculation, the results were printed but the program was not simulating the model in question.

It might be a wrong approach to the problem to contemplate the correctness of the derivation routine available in MIMIC (version 1 modification levels). One should realize that the model was originally developed for DYNAMO and then converted into MIMIC formulation. It is possible that this particular model is not suitable for simulation by the MIMIC language. This assumption could be justified from the modelling point of view. The system model must be suitable for expression in differential equation form; in other words, the approach to MIMIC simulation must originate from Analog Simulation Block Diagrams.

II.4 ANALOG CONVERSION:

To convert the two models into analog computer formulations, three major problems must be solved: physical limitations, time scaling, and amplitude scaling.

a) Physical Limitations: The model must be expressed in the form of differential equations. A simple model like Retail Sector can be analyzed as systems of differential equations. The MIMIC version of representation simulates the analog computer model of the problem. Such a model would require six integrators, one function generator and approximately eight divisors, eight summers and two multipliers.

The Health Care Delivery Model requires more components. The illustration in Figure 4 depicts the secondary variables which influence the level (integration) equations. This structure of the model would necessitate a considerable number of summers, multipliers and divisors. One needs 10 integrators, two function generators and approximately 15 summers, 10 multi-



FIGURE - 4 REVISED FLOW DIAGRAM pliers and 17 divisors. The main assumption at this point is that the MIMIC programs of the above models represent exactly the analog models of the problems. If these problems were reanalyzed and the analog block diagrams were developed, these component requirements might decrease. At present the analog systems available at the school are not capable of solving these problems because of technical limitations.

b) Time Scaling: In these models we are dealing with large values (some in the millions) like population, gross national income, etc. The main limitation of an analog computer is the accuracy of constants. In order to represent a number with a large magnitude like millions or thousands, a very large time scale factor (K) must be chosen. Since scale factor K is multiplied by problem time to evaluate the computer time, the cost of computer time will rise by a tremendous factor.

c) Amplitude Scaling: Since we require the outcome of each amplifier to be limited over the maximum possible fraction of its dynamic range, we need to know the maximum amplitude of each variable in the problem. The amplitude scale factor, reference voltage, and expected maximum of a particular system variable are related by the following equation:

amplitude scale factor = reference voltage (V)
maximum expected value

When the maximum expected value is very large, the scale factor approaches zero and consequently affects the precision of the solution.

The nature of the problems will require much computer time because of the large time scale factor and will eventually output inaccurate results because of very small amplitude scale factors.

Because of the above difficulties the advantage of pure analog solution of these models is questionable. Expensive device requirements and cumbersome scaling manipulations prove the obvious advantage of digital computer solutions of these problems.

On the other hand, an analyst must realize that digital computers follow serial operations. The results may be ambiguous when one attempts to solve a continuous system, in which the operations are parallel, with digital computation methods. The problems encountered in the MIMIC model of Health Care Delivery System might be attributed to operational characteristics of the problem itself and the operational characteristics of the device which is evaluating the solution of the simulation. While these two characteristics are contradictory to each other, in simple systems like Retail Sector Model, contradiction may not cause any problem; but for more complex systems like Health Care Delivery System where various positive and negative feedback loops are coupled with each other, parallelism is the main constraint of the problem.

Continuous large scale system simulation like the Health Care Delivery System Model mentioned above demands the following attributes: parallelism, accuracy and proper devices to construct the model. The analog solution of the problem needs a large number of physical devices and the necessity of scaling may result in loss of precision and thus in unreliable solutions. The parallelism constraint plays an important role in large scale continuous system simulation as we observed in the MIMIC simulation of Health Care Delivery System Model. The Derivative Routine, which is supposed to evaluate the incremental changes of variables in parallel with the other operations, created major problems during the simulation run.

Digital computers provide precision, eliminate the necessity of time and amplitude scaling, and handle the mathematical operations which will eliminate the used extra devices. Analog computers offer the capability of parallelism. As far as the results of the analysis of Retail Sector and Health Care Delivery System models are concerned, the combination of digital and analog computers would give the most efficient solutions for the simulation of a large scale continuous system.

III. REPORT OF ANALOG-DIGITAL COMPARISON USING AUTO SUSPENSION SYSTEM PROBLEM

III.1 SAMPLE PROBLEM USED: (Ferdinand Ricci, <u>Analog/Logic Programming</u> and Simulation, Appendix A)

Automobile Suspension System: Coupled mechanical systems usually require careful study and experimentation to guarantee their correct behavior. This problem presents a particular type of coupled system, a simplified version of an automobile suspension system, for simulation on the computer.

The Simplified illustration of the system is drawn below in Fig. 5





The spring action of the tire provides us with one equation and the actions of the auto spring and shock absorber another. A force-balance on the entire system yields the following second order differential equations:

$$M_1 \ddot{x}_1 + D(\dot{x}_1 - \dot{x}_2) + K_1 (x_1 - x_2) = 0$$

$$M_2 \ddot{x}_2 + D(\dot{x}_2 - \dot{x}_1) + K_1 (x_2 - x_1) + K_2 (x_2 - x_3) = 0$$

 x_3 is a step function that explains the behavior of a car riding up onto a curb. The quantities K_1 , K_2 , M_1 , M_2 , D and x_3 are constants for a given computer run. The following datawere used during our simulation runs:

$$M_1 = 25 \text{ slugs}$$

$$M_2 = 2 \text{ slugs}$$

$$x_3 = 0.2 \text{ ft.}$$

$$D = 20 \text{ lb/ft/sec} \quad (\text{Sensitivity runs: 100 lb/ft/sec and 200 lb/ft/sec})$$

$$K_1 = 1000 \text{ lb/ft} \quad (\text{Sensitivity runs: 500 lb/ft})$$

$$K_2 = 5000 \text{ lb/ft} \quad (\text{Sensitivity runs: 2500 lf/ft})$$

III.2 SELECTION OF SIMULATION TECHNIQUE:

There are normally two considerations in selecting a simulation language: the operational characteristics of the language and its problem oriented characteristics. DYNAMO is a language that is oriented toward problems formulated in terms of nonlinear differential or difference equations. Variables are continuous in their assumed ranges. DYNAMO was developed for defining physical, industrial and social models in which the variables are dynamic in nature. A continuous physical inflow into a physical accumulation and a continuous outflow from an accumulation accomplish the dynamic behavior of the system.

The flow of physical units (information, distance, inventory, population, electric charges, etc.) through the feedback system creates the time patterns; this is the basic philosophy behind DYNAMO modeling. The simplest DYNAMO structure is composed of accumulations, flows and forces which form a closed feedback loop and which create time patterns. The modification of the loops is achieved by time patterns. The structure operated through time creates either a trend pattern or an oscillation pattern. The trend pattern can be in two different forms: growing or declining. The most common form of oscillation pattern is sinusoidal curve.

The feedback dynamics philosophy can be summarized in five basic functions which the physical systems perform:

- 1) the accumulation of time rates of flow
- 2) the establishment of goals with which accumulations are compared
- the creation of forces based on accumulations or on the relationship between goals and accumulations
- 4) the creation of flows (the transfer of units through time from one accumulation to another) as a result of the application of force to accumulations
- 5) the physical or information transformation through time of one type of accumulation unit into another.

These five fundamental functions relate to variables at any level of aggregation in such a way that feedback loops are created.

The totality of relationships between the variables that form the feedback loops is called the structure of the system. The time histories or patterns of variation of the variables are collectively called the performance patterns of the system. Through time the loops create the time patterns; however, it should be noted that the way the loops are structured also changes through time. The structure changes are created by the evaluation and impact of the performance pattern on the components of the system. This structure is illustrated in Fig. 6.

The most prominent language designed to represent processes by continuous models is DYNAMO which represents the dynamic physical (or social) systems as having several levels (accumulations which resemble integrations in mathematical models), flows (rates resembling derivatives) that transport the contents of one level to another, decision functions (force) that control



FIG. 6

the rates of flows between levels, and finally information channels that connect the decision functions to the levels.

Levels and rates are related by simple difference equations:

L (t+1) = L(0) +
$$\sum_{j=1}^{N} (R_{in} (J) - R_{out} (J)) \Delta t$$

t

The level at time t+1 is the level at t=0 plus the cumulative flow (inflow less outflow) to present. Δt is the time step during the occurrence of level change in accumulation.

Force (decision equation) changes rates, e.g.

$$R(t+1) = \frac{L(t)}{K}$$

as a function of a level in the system. A rate equation determines the immediately forthcoming action.

The automobile suspension system, with its dynamic nature, can be defined in terms of differential equations. In my opinion DYNAMO is one of the most suitable contemporary digital computer languages by which the system can be represented. Based on the DYNAMO model a FORTRAN program can be written; unfortunately such an attempt will require extra programming effort and it is an inefficient utilization of the Feedback Model. The orientation of the DYNAMO language is completely designed to program a given Feedback Model.

For its modelling and programing features and its capability of representing differential equations, DYNAMO seems to be the most conformable digital simulation language by which the automobile suspension system can be simulated.

III.3 DYNAMO MODEL:

Momentum and displacement of each mass (M_1 and M_2) are the basic components of the system and they are represented by level equations. Flow into the displacement block is velocity and flow into the momentum block is made up of various force components exerted on the system. These force components are represented by rate equations in the program.

The basic Feedback Model of the automobile suspension system illustrated in Fig. 5 is drawn in Fig. 7. In the model there are four accumulations (levels) which are defined by integration mathematically. The development of the DYNAMO equations was based upon the following physical relationships:

Let Φ_1 = Momentum of the chassis

 Φ_2 = Momentum of the wheel and axle Momentum is defined as follows:

 $\Phi = \int (Force) dt \quad slug-ft/sec$





FIG. - 7

FLOW DIAGRAM OF AUTOMOBILE SUSPENSION SYSTEM

Consequently velocity can be defined as follows

Velocity =
$$\frac{\Phi}{Mass}$$
 $\frac{ft}{sec}$

Net force exerted on the chassis is

$$M_1\ddot{x}_1 = D(\dot{x}_2 - \dot{x}_1) + K_1(x_2 - x_1)$$

and net force exerted on the wheel and axle is

$$M_2\ddot{x}_2 = D(\dot{x}_1 - \dot{x}_2) + K_1(x_1 - x_2) + K_2(x_3 - x_2).$$

Therefore the momentum equations for each mass can be written as follows:

$$\Phi_1 = \int (F_D + FK_1) dt$$

and

$$\Phi_{2} = \int (-F_{D} - FK_{1} + FK_{2}) dt$$
where $F_{D} = D(\dot{x}_{2} - \dot{x}_{1})$

$$FK_{1} = K_{1}(x_{2} - x_{1})$$

$$FK_{2} = K_{2}(x_{3} - x_{2})$$

The velocity equations are formed by the momentum equations:

$$\frac{\mathrm{d}\mathbf{x}_1}{\mathrm{d}\mathbf{t}} = \mathbf{V}_1 = \dot{\mathbf{x}}_1 = \frac{\Phi_1}{M_1}$$
$$\frac{\mathrm{d}\mathbf{x}_2}{\mathrm{d}\mathbf{t}} = \mathbf{V}_2 = \dot{\mathbf{x}}_2 = \frac{\Phi_2}{M_2}$$

The main point to be noted in the DYNAMO formulation is that the equations are generated in reverse fashion relative to physical and mathematical sciences. In other words, it is assumed that there is a continuous flow of velocity which creates the accumulation or displacement. Positive growth of the displacement level stagnates after some span of time because of the resisting forces against limitless accretion. Displacement is accrued by the flow of velocity and momentum is created by the aggregation of forces. The incoming flow into the accumulation boxes (Fig. 7) is controlled by the levels which create positive or negative

feedback loops. Inflow of velocity is monitored by the level of momentum, and inflow or outflow of force is adjusted by the present level of displacement. The dotted lines in Fig.7 depict the course of control on each component. The control action has either positive or negative effect. If the alteration of one variable causes the change of the other variable in the same direction it is called a positive relation. If the effect is in the reverse direction it is called a negative relation. In other words, if an increase in the value of one variable causes the other variable to increase, then the influence of the former variable on the latter is positive; otherwise the influence is negative.

The gain of the feedback loops is found by the algebraic product of each influence in the loop. Fig. 8 illustrates the feedback loops that create the oscillations in the automobile suspension system. There are five negative and three positive feedback loops which are coupled. Negative loops have controlling function whereas the positive loops indicate the growth. When controlling loops are coupled with positive feedback loops, highly oscillatory system behavior is expected. The oscillation declines when the gain of negative feedback loops dominates the growth that is the gain of positive feedback.

a) Specifications of DYNAMO Program:

The Feedback Model of the automobile suspension system was programmed using the second version of DYNAMO on the CDC CYBER 70 Computer. The time span of simulation was two seconds. The selection of the size of DT, the interval of time between the present and the previous time instants, requires a compromise between a large DT which demands slightly less computer time and a small DT which assures numerical accuracy. The magnitude of DT deter-





INFLUENCE DIAGRAM OF BASIC VARIABLES OF AUTO SUSPENSION SYSTEM

mines the accuracy of integration while the reliability increases at the expense of computer time. The number of intervals is evaluated by simply dividing the total simulation time into the length of increment.

Each simulation run consisted of the original set and three sensitivity sets of parameters. In each rerun the value of one parameter, spring constant or damping constant was varied. The execution time includes four runs with different parameters. Three different time intervals were selected and three simulation runs were performed using different (DT) values. Then the percent of increase of computation time versus the change in accuracy was compared and tabulated. The accuracy comparision point was selected randomly at the time point of one second and the values of displacement of the chasis were compared. The results are shown in Table-1.

Table-2 displays the results obtained when the spring and damping constants were altered. The magnitude (amplitude) and oscillation period (length of one cycle) were compared for the displacement and velocity of the chasis and wheel. The results are depicted in Table-2.

b) Results of DYNAMO Simulation:

RUN	DT (sec)	NO. OF INTERVALS LENGHT/DT	TOTAL EXECUTION TIME FOR 4 RUNS ORIGINAL + 3 SENS. (sec)	PERCENT CHANGE OF TIME	VALUE OF X1 AT TIME 1 SEC (ft)	PERCENT CHANGE IN ACCURACY	
(1)	0.002	1,000	14.09 CP	-	82.30	-	
(2)	0.0 0 02	10, 000	19.71 CP	39.8%(1)	86.17	4.7%(1)	
(3)	0.00002 100,000		88.00 CP	524% (1)	86.52	5.1%(1)	
				346% (2)		0.4%(2)	

TABLE - 1

COMPARISION OF ACCURACY VERSUS COMPUTER TIME

Observation of Table-1 asserts the fact that after a certain point, increasing the number of time intervals by smaller choice of DT does not make a considerable contribution to accuracy. As a matter of fact negligable increase of accuracy does not compensate for the tremendous magnification of computer time. 4.7% correction of accuracy costs 39.8% increase in computer time when the number of intervals jumps to 10,000 from 1,000 whereas 5.1% progress in accuracy overcharges 5.24 times as much as the original run. After a reasonable number of intervals the improvement in accuracy is asymptotic with catastrophic rise in computer time. 346% of upward shift in computer time would not compensate 0.4% improvement in accuracy. In our opinion, after 10,000 time intervals the values of the variables will start to converge to a certain limit which is supposed to be the theoretical value. Therefore 10,000 time intervals will be a reasonable (DT = .0002) selection for the auto suspension system simulation. With three sensitivity runs, the total computer time is 19.71 seconds for CDC CYBER 70 computer, including printing and plotting each variable at 100 and 200 time points respectively.

Tabulated results in Table-2 can be used for a better understanding of the system and for the comprehension of the response of the auto suspension system to various parameters involved in the system. The second simulation run with 10,000 time intervals is selected for comparision of different parameters.

Observation of Table-2 shows the response of the system to the basic parameters. The Damping constant (D) and spring constants (K_1 and K_2) have opposite effects on the system. As the spring constants decrease, less oscillatory behavior with low amplitude-smoothed curves is observed. This situation exists when the damping constants are high. Therefore for the best automobile suspension systems, springs with large values of K and shock

53			Ко	D	Maximum Value (ft) Maximum Amplitude (ft)			Longest Cycle (Seconds) Number of Peaks				
	NO.	1b/ft	1b/ft	lb/ft/sec	×1	×2	$\dot{x}_1 \text{ ft/s}$	x ₂ ft/s	×1	*2	×1	ż ₂
	1 1000.	1000	5000.	20.	. 375	.257	1.1	7.96	1.05	0.11	1.07	0.12
		1000.			. 375	.257	2.04	13.65	2	7	2	8
	2 500.	500	500. 2500.	20.	.365	.277	.768	5.15	1.4	0.16	1.45	. 16
		500.			. 365	.277	1.379	8.35	1	5	2	4
	3 1000.	5000.	100.	. 309	.222	1.03	5.04	0.98	3.02	.55	1.01	
				. 309	.222	1.496	5.78	1	2	1	2	
	4	1000.	5000.	200.	.275	.231	1.36	3.51	0.80		.35	
					.275	.231	1.62	3.62	1	0,1	1	1

TABLE - 2

RESULTS OF DYNAMO SIMULATION

,

- : Almost horizontal _

absorbers with a low value of damping constant must be considered to obtain the best performance from the system. In other words, a transient state with high amplitudes slowly entering into a steady state is desired rather than immediate transformation from transient to steady state. In this case the first simulation run seems to give the best performance with the specified parameters.

III.4 ANALOG SIMULATION OF AUTO SUSPENSION SYSTEM:

Analog Computers are electromechanical devices that use many types of components in performing their electrical analogies. The components may be linear or non-linear depending on the function they serve. Before simulating the auto suspension system several basic steps were followed in order to eliminate erroneous results. These programming procedures of an analog computer are illustrated in Fig. 9 in the form of a flow diagram.



FIG. 9

PROGRAMMING PROCEDURES OF ANALOG COMPUTERS

The steps shown in Fig. 9 will be followed step by step in the following paragraphs.

A) Mathematical Model of Auto Suspension System

As it was indicated in section 1, the spring action of the tire provides us with one equation and the action of the auto spring and shock absorber another. By referring to Fig. 5, it is possible to write the force-balance equations as follows:

(1)
$$M_1\ddot{x}_1 + D(\dot{x}_1 - \dot{x}_2) + K_1(x_1 - x_2) = 0$$

(2) $M_2\ddot{x}_2 + D(\dot{x}_2 - \dot{x}_1) + K_1(x_2 - x_1) + K_2(x_2 - x_3) = 0$

where M_1, M_2, K_1, K_2 , and D are constants and x_3 is a step function.

In order to develop a block diagram of the system in the following section, equations (1) and (2) must be manipulated as follows:

(3)
$$\ddot{x}_1 = \frac{D}{M_1} (\dot{x}_1 - \dot{x}_2) - \frac{K_1}{M_1} (x_1 - x_2)$$

(4) $-\ddot{x}_2 = \frac{D}{M_2} (\dot{x}_2 - \dot{x}_1) + \frac{K_1}{M_2} (x_2 - x_1) + \frac{K_2}{M_2} (x_2 - x_3)$

Equations (3) and (4) were solved for the highest order of derivatives. From these equations it is possible to draw the unscaled block diagram of the system. (Fig. 10)

B) Block Diagram of Auto Suspension System:

Fig. 10 displays the block diagram of the system generated from equations (3) and (4). The unscaled version of the block diagram can be seen by ignoring the dark lines on the figure.

The system requires four integrators, three summers and three inverters. Besides ten amplifiers, the system requests eleven potentiometers after the





Unscaled Diagram

Scaled Diagram

scaling factors enter into the system; this will be explained in the following paragraphs. The available analog unit (COMDYNA GP-6) was not sufficient to provide all the necessary components, so two of the same units were connected together; one served as the master computer and the other one was used as the slave computer.

C) Amplitude and Time Scaling:

The need for amplitude and time scaling arises because of the incompatibility of the analog computer with natural phenomena. The analog computer can only vary between plus and minus one machine unit (±10 volts in COMDYNA GP-6), whereas physical systems have amplitudes which can be very large or very small. The analog computer operates in one time frame, whereas physical systems occur over a long span of time or over very short periods of time. From the analysis point of view, amplitude and time scaling are indispensable for the whole cycle of the system to be observed on the display of the computer within the voltage limits of the machine and time limits of man and the machine.

The maximum pot setting value in the COMDYNA GP-6 analog computer is plus or minus one unit. In order to keep the variation of variables within one machine unit, variables would have to be divided by the maximum possible value of the variable:

$$\frac{x}{x \max} \leq 1$$

The equation for the highest order derivative takes the following form where (a) and (b) are constants:

$$\frac{d}{dt} \begin{bmatrix} \frac{1}{x} \\ \frac{1$$

The outputs of all amplifiers are now between plus and minus one machine unit. The representation of the parameters on the computer will be deter-

$$\frac{d}{dt} \begin{bmatrix} \frac{\dot{x}}{\dot{x} \max} \end{bmatrix} = -\begin{pmatrix} \frac{b}{a} \end{pmatrix} \begin{bmatrix} \frac{\dot{x}}{\dot{x} \max} \end{bmatrix} - \begin{pmatrix} \frac{C \ x \max}{a \ \dot{x} \max} \end{pmatrix} \begin{bmatrix} \frac{x}{x \max} \end{bmatrix}$$
(Gain) (Pot Set) Amplifier (Gain) (Pot Set) Amplifier
Output Output

The coefficients in front of the amplifier outputs are the product of the amplifier gains and potsettings. The amplifier outputs determine the amplitude scaling and the coefficients determine the time scaling. The maximum amplifier gain of the COMDYNA GP-6 computer is 10 times. If the coefficients appearing in the equation are not possible to obtain, that is, if the product of the potsetting and gain is not between 0.05 and 10 $(0.05 \leq (Pot Setting)(Gain) \leq 10)$, then time scaling is necessary. A potentiometer must be placed between all integrators in an analog computer circuit to obtain a scaled diagram.

For time scaling we must accept some constant β , such that:

Time scale factor = $\beta = \frac{\Upsilon}{t} = \frac{Computer Time}{Problem Time}$

The maximum values of all derivatives can be obtained for a general nth order derivative by simply solving for the natural frequency

$$w_n = \sqrt[n]{\frac{a_s}{a_n}}$$

where the general homogeneous differential equation

$$a_{n} \frac{dx^{n}}{dt^{n}} + a_{n-1} \frac{dx^{n-1}}{dt^{n-1}} + \dots + a_{0} x = 0$$

$$\left| x^{n} \max \right| \approx w_{n}^{n} x(0)$$

From equations (1) and (2), the maximum estimated values of higher order derivatives of auto suspension system can be evaluated after forming equations (5) and (6)

(5)
$$M_1 \ddot{x}_1 + D\dot{x}_1 + K_1 x_1 = K_1 x_2 + \dot{x}_2 D$$

(6) $M_2 \ddot{x}_2 + D\dot{x}_2 + (K_1 + K_2) x_2 = K_1 x_1 + K_2 x_3 + D\dot{x}_1$

Undamped natural frequencies of the equations using the original data are:

$$w_1 = \frac{2}{\sqrt{\frac{K_1}{M_1}}} = \sqrt{\frac{1000}{25}} \approx 6.32 \text{ and } w_2 = \sqrt{\frac{K_1 + K_2}{M_2}} = \sqrt{\frac{6000}{2}} \approx 54.8$$

where the nth order derivative is 2.

We let $w_1 = 5$ and $w_2 = 50$.

Max $\dot{x}_1 = w_1 \text{ Max } x_1 \approx 5$ and Max $\dot{x}_2 = w_2 \text{ Max } x_2 \approx 50$. The estimated value of displacements x_1 and x_2 is one.Maximum estimated values for $[x_1 - x_2]$ $[\dot{x}_1 - \dot{x}_2]$ and $[x_2 - x_3]$ are 2 ft, 50 ft and 2 ft respectively.

By using the scaled variables, equations (3) and (4) can be written as follows:

$$\begin{array}{rcl} (3a) & \frac{d}{dt} \left[\frac{\dot{x}_1}{5} \right] = & \frac{50D}{5M} \left[\frac{\dot{x}_2 - \dot{x}_1}{50} \right] & + & \frac{2}{5} \frac{K_1}{M_1} \left[\frac{x_2 - x_1}{2} \right] \\ (4a) & - & \frac{d}{dt} \left[\frac{\dot{x}_2}{50} \right] & = & \frac{50D}{50M_2} \left[\frac{\dot{x}_2 - \dot{x}_1}{50} \right] & + & \frac{2}{50} \frac{K_1}{M_2} \left[\frac{x_2 - x_1}{2} \right] & + & \frac{2K_2}{50} \frac{x_2 - x_3}{2} \end{array}$$

Equations (3a) and (4a) are the properly scaled equations; however, one must look at the integrator gains to be certain that they are between 0.05 and 10. Examination of the maximum values of the coefficients shows that some of them are greater than 10:

$$\frac{10 \text{ D}}{\text{M}_1} = 8 \qquad \frac{\text{K}_2}{25 \text{ M}_2} = 100 \qquad \frac{\text{D}}{\text{M}_2} = 10 \qquad \frac{2 \text{ K}_1}{5 \text{ M}_1} = 16 \qquad \frac{\text{K}_1}{25 \text{ M}_2} = 20$$

Therefore time scaling is required. Equations (3a) and (4a) can be rewritten by choosing β as the time scale factor:

$$\frac{d}{d\tau} \left[\frac{\dot{x}_{1}}{5}\right] - \frac{10 \text{ D}}{M_{1}\beta} \left[\frac{\dot{x}_{2} - \dot{x}_{1}}{50}\right] + \frac{2 \text{ K}_{1}}{5 \text{ M}_{1}B} \left[\frac{x_{2} - x_{1}}{2}\right] \quad (3b)$$

$$\frac{d}{d\tau} \left[\frac{\dot{x}_{2}}{50}\right] = \frac{D}{M_{2}\beta} \left[\frac{\dot{x}_{2} - \dot{x}_{1}}{50}\right] + \frac{K_{1}}{25 \text{ M}_{2}\beta} \left[\frac{x_{2} - x_{1}}{2}\right] + \frac{K_{2}}{25 \text{ M}_{2}\beta} \left[\frac{x_{2} - x_{3}}{2}\right] \quad (3b)$$

$$(3b)$$

where $\tau = \beta t$ $dt = \frac{d\tau}{\beta}$

 β = 10 seems to be the most reasonable value for the time scaling factor; then the maximum coefficient drops to 10 and this is a suitable value for the COMDYNA GP-6 analog computer.

D) Patching and Static Checking.

The scaled block diagram of the system is shown in FIG. 10 with dark scaling manipulations. The addition of potentiometers between integrators and into summers makes the scaling complete.

The block diagram in FIG. 10 was used to patch the panel of the analog computer. The potentiometer settings for the coefficients are evaluated as follows:

 $D = 20 \ 1b/ft/sec$ $K_1 = 1000 \ 1b/ft/$ $K_2 = 5000 \ 1b/ft$

$$\frac{10 \text{ D}}{M_1 \text{B}} = .8(1) \star \qquad \frac{D}{M_2 \beta} = 1.(1)$$

$$\frac{2K_1}{5M_1 \beta} = 1.6 = 1.6(10) \qquad \frac{K_1}{25 \text{ M}_2 \beta} = 2 = .2(10)$$

$$\frac{K_2}{25 \text{ M}_2 \beta} = 10 = 1(10)$$
$$D = 20 \ 1b/ft/sec , \quad K_1 = 500 \ 1b/ft \quad K_2 = 2500 \ 1b/ft$$
$$\frac{2K_1}{5M_1\beta} = 0.8 = .8(1) \qquad \frac{K_1}{25M_2\beta} = 1 = 1.(1)$$
$$\frac{K_2}{25M_2\beta} = 5 = .5(10)$$

The coefficients for the sensitivity runs are calculated as following:

A second simulation run was performed by rearranging the potentiometer settings of (7), (10) and (12) according to the above calculations.

In order to check the sensitivity of the system to the changes in the damping constant, the value of D is raised to 100 lb/ft/sec. Thus the coefficient settings will be as follows:

$$D = 100 \ 1b/ft/sec$$
 $K_1 = 1000 \ 1b/ft$ $K_2 = 5000 \ 1b/ft$

$$\frac{10D}{M_1\beta} = 4 = .4(10) \qquad \qquad \frac{D}{M_2\beta} = 5 = .5(10)$$

The output of potentiometer 9 must enter into a 10 volt gain of the third amplifier. Unfortunately amplifier 3 has only two 10 volt gain inlets which are already used by potentiometer ((10) and (12)) outlets. Therefore due to the technical limitation of the COMDYNA GP-6 analog computer we are not able to use damping constants larger than 20 lb/ft/sec.

The computer time of 20 sec. was selected for convenience. Since the time scaling factor is 10, the total run reflects the system performance during 2 seconds of actual time.

Potentiometer setting for Pll is calculated as follows:

 $x_3 = 0.2$ ft. = (12)(0.2) = 2.4 inch inch = 2 volts \Rightarrow 2.4 inch = 4.8 volts $\frac{x_3}{x} = 2.4$ volts = .24(10) is the setting of potentiometer 11.

E) Computer Implementation and Output:

Two different simulation runs were performed to analyze the sensitivity of the system to the changes of spring constants. Due to the technical limitation of the COMDYNA GP-6 analog computer, it was not possible to analyze the sensitivity of the system to the changes of the damping constant D.

Appendix 1 and Appendix 2 display the outputs of the analog computer diverted to an X-Y plotter. The plots show the values of x_1 , x_2 , $\dot{x}_1/5$ and $\dot{x}_2/50$ by plotting the outputs of amplifiers (2), (7), (8) and (3) respectivity.

Table-3 shows the results of analog computation with two runs. To make the results comparable with Table-2 the values were converted to feet, and the outputs of potentiometers (8) and (3) were multiplied by 5 and 50 respectively.

III.5 COMPARISON OF DIGITAL AND ANALOG SIMULATION OUTPUTS:

Table-2 and Table-3 are the two key tables for the comparision of results. Detection of sensitivity of the system to the changes of spring constants was achieved with analog computation and with DYNAMO simulation in a similar fashion. When the spring constants are decreased by half, fewer oscillations occur with longer cycles. With larger spring constants the system achieves moderate transformation from a transient state to a steady state.

DVDV	К ₁	K ₂	D	Maximum Value (ft) Maximum Amplitude (ft)					Longest Cycle (seconds) Number of Peaks				
RUN	15/1t	1b/it	lb/it/sec	×1	*2	x ₁ ft/s	x ₂ ft/sec	×1	*2	*1	*2		
1	1 1000. 5000.	5000	20	.371	.295	1.1	7.91	.92	0.09	0.95	.1		
1		5000.	20.	.371	.295	2.02	13.75	2	7	2	8		
2	2 500. 250	. 2500.	2500.	20.	.362	.275	0.81	5.20	1.2	0.15	1.2	.13	
L			20.	.362	.275	1.45	8.34	2	5	2	5		

TABLE - 3

RESULTS OF ANALOG COMPUTER SIMULATION

The first two runs of the analog and DYNAMO simulations are similar, as shown in Table-2 and Table-3. On the other hand the occurrence of the peak points shows some minor discrepancies. (APP-1 and APP-3). For example the first peak of $x_1(APP-1)$ occurs at 0.45 second of the analog simulation whereas $x_1(APP-3)$ occurs at 0.52 second of the DYNAMO simulation. This difference can be attributed to various factors such as inaccuracy of analog computation, manual selection of the starting point of the X-Y plotter, approximation of the discrete graph of the DYNAMO plot and truncation of numbers after six significant digits by the DYNAMO conversion processor.

Although there are some negligable incongruities between the DYNAMO and analog simulations, it was possible to get the relevant information about the system from both types of simulation. The effect of discrepancies would have no impact on the design procedure of the system. It is possible to get the print-outs of results accurate up to six significant digits, but on the plots the printer of the digital computer prints at discrete points and the locations of the points in the output array are selected by approximation.

On the other hand the plots obtained by the analog computer diverted to the X-Y plotter are accurate and continuously plotted against time. Continuity eliminates the necessity of approximation. With the present equipment in the Georgia Tech Information and Computer Science laboratory, it was not possible to print the values at certain intervals of time with the analog computer.

We must consider the approximation in the discretely varying digital computers when continuously varying dynamic systems are simulated. For reliable approximations the selection and specification of the size of the integration step depends upon the decision of the analyst, and cannot be controlled by the digital computer.

Three common operations were found between digital and analog simulation: (1) Initializing Operations, (2) Dynamic Computing Operations, and (3) Terminating Operations. Table-4 summarizes these three common operations in a comparative manner.

In our opinion it is not possible to decide which method is superior to the other one, because each programming method has its own attributes. The selection of a simulation method depends upon the characteristics of the particular model; i.e., the auto suspension system simulation model can be expressed in terms of differential equations so that programming was straightforward by analog and DYNAMO techniques with some deficiencies. These deficiencies can be eliminated by implementing a complementary system that combines certain characteristics of each method. In a combined or hybrid system, the computational precision of digital computers would complement the continuous plotting characteristic of an analog computer to obtain better results; a central processor connected to an analog unit may eliminate the necessity of time scaling and amplitude modification.

In order to simulate a system by an analog computer, it is a requirement that the system be expressed in terms of differential equations, and this is one of the limitations of the analog computer. A complemented "hybrid" system may eliminate this deficiency of an analog system.

		ANALOG SIMULATION		DIGITAL SIMULATION
Initializing Operations	1. 2. 3. 4. 5. 6.	Draw Block Diagram Make Time and Magnitude Scaling Scale the differential equations Patch Set potentiometers Set initial conditions	1. 2. 3. 4. 5. 6. 7.	Draw flow diagram and influence diagram. Define the variables. Write the program. Specify initial conditions. Select integration time step size. Specify time duration of run and other limits. Assign values to the constants and IC.
Dynamic Computing Operations	1.	With the mode control in OP, the computer simultaneously carries out the required operations of addition, integration, multi- plications and inversion, as required to effect the desired simulated behavior of the physical systems being studied. Parallel processing is available, there- fore all the components can be simulated at the same time.	1.	The digital computer, according to the instruc- tions of the simulation program, executes sequen- tially the required numerical operations to add, integrate, multiply, etc., as required to effect the desired simulated behavior of the physical system being studied. Digital simulation is a serial process. Each component is simulated sequentially according to precedence relationship.
Terminating Operations	1. 2. 3.	Observe the output of the simulated behavior using CRT display or X-Y plotter. Change mode control to RESET. Modify potentiometer settings, initial conditions, etc.	1.	Print and plot the results of the simulated behavior at the printer. Change or modify the constants and rerun the program.



· ·

. . . .

· .

10	NCH = JOLT S																					
· ···· · ····		-																	<u>н</u> р. т			2
																						1
•••••••		×1/5								+								1 - 1				
2			-															· · · ·				 •
		X	\						/							1	······································			:		1.1 . 1.1 1 .
· · · · · · · · · · · · · · · · · · ·	. ×.	L						/											1			
	5							/		-	\mathbf{N}											
	VIII	$(\land \land$														+	21	(++ + ₂ - 2 + 1 1		: 	•	
· · · · · · · · · · · · · · · · · · ·			V Y				1/	4	•								1	\		2	SEC	
	1-1						/					$\left \right\rangle$				1						•
- 1				λ		/							\mathbf{X}^{-}			/						 : .
·····				\mathbf{X}		/-												∔⊷'				•
···2 -	: 			-	/	4							-		· ·							;
• • • • •														<i>D</i> =	: 20 16	: fff	sec	į				· · · ·
	.									-				K. =	1000.	16/4	t. ct.	1				• • •
														×3 =	24	volts		2.f.f				
							APPe	NDIX	16					ζ= β=	-20 : 10.	Sec.						
······	·····														17		· · · · · ·			·		
	· · · · · · · · · · · · · · · · · · ·																					
																					1. : : 1	





. •

47 1323

er alle Multer in Avit in Kolonia. Avidente i Octobri de Tradicio de

- J.W. Atwood. A Highly Flexible Hybrid Computation Facility. 1973 Summer Computer Simulation Conference, Montreal. July 17, 18, 19, 1973. pp. 216-218.
- J.K. Bishop, et al. Hybrid Computation: Fourth Generation and Communications. 1975 Summer Computer Simulation Conference, San Francisco. July 21, 22, 23, 1975. pp. 249-256.
- W.M. Hester Requirements for a Remote Access Hybrid Computer Terminal Utilizing Common Grade Telephone Lines. Research Report, Florida Technological University. Orlando, Florida. 1974.
- R.M. Howe, R.B. Hollstein. Time Shared Hybrid Computers: A new concept in Computer-Aided Design. Proc. IEEE, LX: January, 1972. pp. 71-77.
- D.E. Hyndman. Analog and Hybrid Computing. Pergamon Press. New York. 1970.
- D.G. Hoeschele. Analog-to-Digital/Digital-to-Analog Conversion Techniques. Wiley, New York. 1968.
- D.C. Martin. Telecommunications and Terminals. Advanced Hybrid Computer Systems. U.S. Army Materiel Command. June, 1974.
- D.C. Martin. The Analog Is Dead. Simulation, XV. September, 1970. pp. 125-130.
- J.L. Melsa, et al. A Telephone Modem for a Hybrid Computer. SWIEEECO Record of Technical Papers. April, 1973. pp. 341-345.
- A.I. Rubin, et al. A Future for Hybrids? <u>Research/Development</u>, XXVI. May, 1975. pp. 38-40, 42, 44.
- G.J. Sauer, et al. Hybrid Computation: Fourth Generation and Communications. 1975 Summer Computer Simulation Conference, San Francisco, July 21, 22, 23, 1975. pp. 249-256.
- Technology Evaluation of a Heterogeneous Element Processor. Report Submitted to the Department of the Army by Denelcor, Inc. May, 1974.
- V. Vemuri. Analog and Hybrid Computation. <u>Computer Science</u>. ed. A.F. Cardenas, et al. Wiley, New York. 1970.

APPLICATIONS OF ANALOG/HYBRID COMPUTATION

- P.J. Anderson, et al. Skylab Mission Simulator Facility. 1973 Summer Computer Simulation Conference, Montreal. July 17, 18, 19, 1973. pp. 219-247.
- J.B. Beyer, V. Sokolov. Application of the Hybrid Computer to the Study of the Large Signal Noise Measure in IMPATT Microwave Amplifiers. 1975 Summer Computer Simulation Conference, San Francisco. July 21, 22, 23, 1975. pp. 226-240.

- H.D. Dunkle. CalComp Plotting An Aid to Hybrid Simulation Validation. 1975 Summer Computer Simulation Conference, San Francisco. July 21, 22, 23, 1975. pp. 195-198.
- J.M. Gaillard, et al. Principles of Automatic Analysis of Sleep Records with a Hybrid System. <u>Computers and Biomedical Research</u>, VI. February, 1973. pp. 1-13.
- W.G. Hardison, et al. Use of Hybrid Computers to Analyze Behavior of Detailed Models of Biological Systems. <u>Computers in Biology and Medicine</u>, IV. June, 1974. pp. 3-17.
- N.E. Pobanz. A Peek at Hybrid Computation in the Nuclear Power Plant Industry. 1974 Summer Computer Simulation Conference, Houston. July 9, 10, 11, 1974. pp. 144-151.
- F.J. Ricci. <u>Analog/Logic Programming and Simulation</u>. Hayden. Rochelle Park, New Jersey. 1972.
- R.E. Stephenson. <u>Computer Simulation for Engineers</u>. Harcourt Brace Jovanovich. New York. 1971.
- R.B. Wavell, J.K. Van Hook. Validation and Verification Procedures for Hybrid Computer Simulation Models. 1975 Summer Computer Simulation Conference, San Francisco, July 21, 22, 23, 1975. pp. 1228-1234.

DIGITAL SIMULATION

COMDYNA, Inc. Analog Simulation and Computation Programming Manual. GP-6.

- J.R. Emshoff, R.L. Sisson. <u>Design and Use of Computer Simulation Models</u>. Hayden. Rochelle Park, New Jersey. 1970.
- W.R. Fey. The Dynamics of Simultaneous Inflow and Outflow Control. Proceedings of the Tenth Annual Meeting of the Southeastern Institute of Management Science, Miami, October 31 - November 1, 1974.
- W.R. Fey. Primary and Secondary Control of Accumulations. Unpublished paper. Georgia Institute of Technology. 1974.
- G. Gordon. <u>System Simulation</u>. Prentice-Hall. Englewood Cliffs, New Jersey. 1969.
- A.L. Pugh. DYNAMO II User's Manual. MIT Press. Cambridge, Massachusetts. 1973.

HYBRID SOFTWARE AND AUTO-PATCHING

M.S. Elzas. HL1 or Towards a Unique Language for All Continuous System Simulation (An Introduction to the HCS1 Compiler System). Proc. A.I.C.A. 7th Conference on Hybrid Simulation, Prague . August, 1973. pp. 65-70.

- P.H. Enslow, Jr. Hybrid Computer Research in Europe. Special Symposium on Advanced Hybrid Computing, San Francisco. July 23-24, 1975. pp. 167-172.
- P. Grogono. MUSYS Software for an Electronic Studio. <u>Software Practice</u> and <u>Experience</u>, III. October-December, 1973. pp. 369-383.
- G. Hannauer, A. Asthana. Recent Advances in Automatic Patching Software. Special Symposium on Advanced Hybrid Computing, San Francisco. July 23-24, 1975. pp. 31-36.
- C.J. Kelley. CHAMP: The Cranfield Hybrid Automatic Maintenance Program. The Radio and Electronic Engineer, XLI. October, 1971. pp. 439-452.
- J.J. Martin, III. Automatic Patching of Analog/Hybrid Computers via Interactive Graphics. 1975 Summer Computer Simulation Conference, San Francisco. July 21, 22, 23, 1975. pp. 265-275.
- T.I. Oren. Digital Simulation Languages for Combined Systems An Overview. 1973 Summer Computer Simulation Conference, Montreal. July 17, 18, 19, 1973. pp. 346-353.
- R. Peale, et al. APSE. A Market Analysis and Strategy for the E.E.C. A Study carried out by a project team from the Manchester Business School on Behalf of the National Computing Centre. March, 1974.
- H.B. Rigas, D.J. Coombs. PATCH: Analog Computer Patching from a Digital Simulation Language. <u>IEEE Transactions on Computers</u>. October, 1971. pp. 1140-1146.
- P.E. Rook. ACTRAN, An Analogue/Hybrid Compiler. Proc. AICA-IFIP 6th Conference on Hybrid Simulation. Munich. September, 1970.
- P.E. Rook. The HCS-1 Hybrid Compiler: How Well Does It Work? Special Symposium on Advanced Hybrid Computing, San Francisco. July 23-24, 1975. pp. 63-70.
- D.W. Russell. Automated Analogue Patching Using List Processing and a Hybrid Interface. Kybernetes, II. 1973. pp. 95-100.

HYBRID VS. DIGITAL COMPUTATION

- R.D. Benham. Evaluation of Hybrid Computer Performance on a Cross Section of Scientific Problems. AEC Research and Development Report. January, 1970.
- T.M. Khalil, et al. A Comparative Study of Digital and Hybrid Simulation of Production Lines. Simulation, XVIII. February, 1972. pp. 75-78.
- P.F. Stojak. Hybrid and Digital Computation Results in Multicomponent Distillation Simulation. <u>The Computer Journal</u>, XVI. November, 1973. pp. 368-374.

L. Wolin. The Wolin-Saucier-Peak (WSP) Scientific Mix, A Quantitative Method for Comparing Hybrid and Digital Computer Performance. Special Symposium on Advanced Hybrid Computing, San Francisco. July 23-24, 1975. pp. 181-188.

ADVANCED HYBRID COMPUTER SYSTEM

- Advanced Hybrid Computer Systems. Report prepared by Applied Dynamics Computer Systems for U.S. Army Materiel Command. May, 1974.
- R.M. Howe. The Next Generation of Hybrid Computer Systems. Proc. of the National Electronics Conference, XXIX. October, 1974.
- D.C. Martin. Telecommunications and Terminals Advanced Hybrid Computer Systems. Prepared for the U.S. Army Materiel Command. June, 1974.
- J.B. Mawson. An EAI Discussion of an AHCS. Special Symposium on Advanced Hybrid Computing, San Francisco. July 23-24, 1975. pp. 17-30.
- H.B. Rigas. Software Technology Advanced Hybrid Computer Systems. Prepared for the U.S. Army Materiel Command. June, 1974.
- A. Saucier. An Overview of Advanced Hybrid Computing. Special Symposium on Advanced Hybrid Computing, San Francisco. July 23-24, 1975. pp. 157-160.
- L.M. Warshawsky, R.M. Howe. Operating Characteristics of the Applied Dynamics AHCS. Special Symposium on Advanced Hybrid Computing, San Francisco. July 23-24, 1975. pp. 3-10.

HYBRID/ANALOG COMPUTATION IN EDUCATION

- W.S. Adams. A Research Oriented Hybrid Facility. Special Symposium on Advanced Hybrid Computing, San Francisco. July 23-24, 1975. pp. 107-111.
- The Analog Computer as a Teaching Tool in Physics. <u>American Journal of Physics</u>, XLI. May, 1973. pp. 622-631.
- D.L. Cohn, J.L. Melsa. Hybrid Computer in Graduate and Undergraduate Education. SWIEEECO Record of Technical Papers. April, 1973. pp. 479-481.
- R.A. Davidson. Hybrid Made Easy through the Use of Educational Simulation Terminals. Special Symposium on Advanced Hybrid Computing, San Francisco. July 23-24, 1975. pp. 95-106.
- L.E. Fogarty, R.M. Howe. A Review of Educational Experience with an Automatically Patched Hybrid Computer System. Special Symposium on Advanced Hybrid Computing, San Francisco. July 23-24, 1975. pp. 87-94.

- A.P. Jensen, J. Gehl. An Instructional and Research Facility for Syndetic Digital-Analog Computation. Special Symposium on Advanced Hybrid Computing, San Francisco. July 23-24, 1975. pp. 83-86.
- M.C. Mulder. Model Curricular for Four-Year Computer Science and Engineering Programs: Bridging the Tar Pit. <u>Computer</u>, VIII. December, 1975. pp. 28-31.
- G.A. Rahe. A Computer Research Laboratory for Engineering Laboratory Instruction. 1973 Summer Computer Simulation Conference, Montreal. July 17, 18, 19, 1973. pp. 254-258.

3. SOME TYPICAL STUDENT PROJECTS FOR A COURSE IN SYNDETIC COMPUTING

The two papers which comprise the bulk of this section span a range of applications of syndetic computing.

The first paper examines the design and construction of a digital function generator which would permit the use of tabular data in a setting which would exploit analog functions in the construction of a model. This paper was written by a graduate student in ICS whose undergraduate degree was in Industrial and Systems Engineering and who had no electronics background or experience beyond sophomore physics but who was strong in modeling and simulation.

The second paper illustrates the design of an electronic music box in which the outputs of tuned oscillators is controlled and mixed digitally to produce organ-like quality. This device was completely designed, built and demonstrated by a graduate student in ICS whose undergraduate degree was in music and who had no experience with electronics beyond the use of appliances.

While these projects may not be exactly typical of the set of such projects that have been accomplished by ICS students in the course of this study, they represent significant accomplishments which answer the question of what can be accomplished via syndetic computing studies. From this evidence we contend that students without electronic experience can be taught the concepts necessary to enrich and underscore computer understanding to the point that computers can be composed to perform specific functions.

Ancillary to these projects and others of a related nature, several integrators were designed using inexpensive (cost less

than \$1.00) operational amplifiers. These integrators were compared with those of several analog computers costing from \$1,000 for a four integrator unit to several thousand dollars for a mini-hybrid computer. It was found that students could build integrators for a component cost of 15-20 dollars, integrators that competed very favorably with those of commercially available analog computers. The design of two such integrators is shown in figures 3C.1 through 3C.3. This fact is of significance to the issues of syndetic computing.

3A. Table Function Generator

.

78

.

•

1. Introduction:

The problem which triggered this project was the problem of "Retail Sector Inventory" which was originally implemented in DYNAMO. The Retail Sector Inventory Problem has been modeled as Analog block diagram (see Appendix-3), but unfortunately due to technical limitations such as unavailability of a table function generator on the existing Analog computer at ICS laboratory, it was not possible to run the Analog version of the problem.

A new technology which was developed and became apparent during the last decade, Hybrid computation could be one of the solutions for the Retail Sector Inventory problem which was unsolvable with Analog methods due to technical limitations. Since the same problem has been solved already with digital methods with lack of precision, why not combine both analog and a digital device together to realize the advantages of both in a complementary way and thereby have a superior simulator which solves problems with dynamic variables continuously changing through time. This combined method is named as Hybrid Technique that is a coupled system of Analog and Digital components.

2. Problem Definition:

The table function which was not available in Analog System, can be generated and stored in a memory in digital computation methods. One of the proper ways of generating the table values is to connect the memory device to the output signals of a mini computer. In this project the table values are stored into memory manually.

The memory consisted of 1024 bits and it was divided into 10-bit words. The first word contains the value of the function generated from the first value of the independent variable. The successive words in the memory are corresponding to the succeeding values of the independent variable in incremental order.

The requested information extracted from the proper memory location by shift registers is stored in a 10-bit latch. The stored information is transferred into an eight-bit multiply digital-to-analog converter which is coupled with an MC1408 amplifier. The output of the operational amplifier is the analog signal (voltage level) of the digital information.

Several gates are included in the model to synchronize proper timing. The timing chart is shown in Appendix-2.

3. Components of the System:

Appendix-1 shows the flow-diagram of the model which is the integrated combinations of the following components: a) 2102 MOS Random Access Memory (7416): The 2102 is a 1024-word by 1-bit Static Random Access Memory element using normally off N-channel MOS devices integrated on a monolithic array. The data is read out nondestructively and has the same polarity as the input data.

As it can be seen from the Timing Chart (App.-2) "Write" process occurs during the interval of 8th and 12th clock pulses. Addressing is triggered during the whole 16 clock cycles, the address in the memory is incremented by 1 after every 16 clock cycles.

To obtain "low" Read/Write pulse inverted C, and D outputs of 4-bit binary counter are NAND'ed and the result is OR'ed with SW1 which is at "low" level during "write". Therefore when D is at high level and C is at low level and when the switch 1 is "off", the signal entering into pin 3 of MOS is low which causes the data to be written in the location indicated by row and column selection bits.

During "Read" cycle, Switch 1 is "ON" position, thus the Read/Write signal is always high.

Switch-2 is used to select 0 and 1 bits of the words to be written into the memory.

b) 4-Bit Binary Counters (7493):

5,4-Bit Binary counters are used in the model. The first one is directly connected to the clock output. Its main function is to delay the clock pulses 16 cycles. Once cycle is divided by 16, the period of the clock is multiplied by 16. This extended clock pulse triggered the address reference bits to be incremented by 1. In other words, first 7493 IC causes the address bits in the memory to be incremented by one pointing to the next bit.

The other four 7493 IC's are used to address 32 rows and 32 columns of the RAM. Two coupled 7493's are used as divide by 32 adder. Each coupled chip has capability to address up to 32 columns or rows. Once the 31 columns are referenced, the row number is incremented by 1 and column counter is cleared to zero.

c) Divide by 10 counter (7490):

This counter is used to load 7475 latch from the shift register. The shift process is completed after incrementing the address bits of RAM 10 times (160 clock cycles). This counter generates a pulse which indicates the end of shift, and sends that pulse to the Monostable Multivibrator. 7490 chip counts up to 10 which is equivalent to the available storage area in one word. The output signal is inverted to start counting during the low to high transition (Appendix-2) of the voltage level.

d) Monostable Multivibrator (SN74123):

This circuit is often called a one-shot circuit. Although it may be triggered into its other state, it will return and remain in its stable state until triggered again.

To achieve data input to a D-Latch, a Hi pulse is required on the clock input of the latch. An SN 74123 could generate this pulse, however, since each 7475 has two clock inputs the maximum fanout of the 74123 may be exceeded with only a few latches. Therefore a 7440 buffer is used to insure sufficient fanout.

When there is a low to high transition in 7490 a single pulse is generated by the two SN 74123 circuits to insure that the D-Latch receives the information from the shift register after 10 successive shifts are completed. (Appendix-2)

e) 5-Bit Shift Registers (7476):

Two 7476 circuits are coupled together to transfer information starting from most significant bit of the referenced 10 bit word. All the parallel present inputs are grounded and serial inputs are received from D out pin of MAR. Parallel transfer of this information is accomplished with the load signal of the SN 74123 circuit at the end of address reference of the last bit in the Word. The D output of the first 7493 circuit in the input clock pulses to the shift registers. Switch-4 is used to clear the registers before any operation.

f) Quad Latch (7475):

3 Quad D-Latches are used to accomplish parallel and simultaneous transfer of data from Shift registers to D/A converter. Unless new input is entered in 7475 the content of the circuit does not alter. The change occurs in the contents of the latches whenever a high pulse is applied to the clock input.

g) Eight-Bit Multiplying D/A Converter (MC 1408 LC):

The circuit consists of a reference current amplifier, an R-2R ladder and eight high-speed current switches. The switches are noninverting in operation, therefore a high state on the input turns on the specified output current component. The switch uses current steering for high speed and a termination amplifier consisting of an active load gain stage with unity gain feedback. The termination amplifier holds the parasitic capacitance of the ladder at a constant voltage during switching, and provides a low impedence termination of equal voltage for all legs of the ladder.

An MC 1741 operational amplifier is also a part of the D/A system. The amplifier gain is 0.5 (1K/2K) and amplifier has inverting structure. 5230B diode was used to generate reference voltage. +15V input current was dropped to 4.7V. The configuration of D/A converter is in Appendix-1.

Since MC1408 can hold only 8 bit word, only the most significant bits available in 10-bit Q-latch are wired to D/A. The last two bits of memory are not used.

The voltage output of the whole circuit is evaluated as follows:

Summation Constant= $\frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \dots + \frac{A_8}{256}$

where A's are the values (1 or 0) of each memory bit. $V_0 = \frac{Vref}{R}$ (R) (Summation constant) - $\frac{Vref}{22R}$ (R)

where Vref = 4.7V R = 1K ohms R = 1K ohms R = 2.2K ohms

 ${\rm V}_{\rm O}$ is the output voltage of the whole model which is supposed to be the analog signal of the digital information stored in the Q-Latch.

4, Sample Problem:

The Retail Sector Inventory Problem formulized in Appendix-3 in the form of differential equations is impossible to solve on the available analog computer at ICS laboratory.

The Table Function Generator could be used to store the tables necessary and extract the proper value from the storage area. In this experiment Requisitions Received at the Retail sector over 10 week period (Fig. 1) are stored in memory in the form of a table function. The values are stored bit by bit in 10 bit words successively. The most significant bit must be the very first bit of each word. The last two lowest bits of each word are not used because the available D/A converter accepts only 8 bits as input.

As a sample problem the following table function is stored in the memory:

Week	Requisitions Received At Retail	Binary Number to be Stored
0	80	0101000000
- 1	95	0101111100
2	160	101000000
	225	1110000100
1. S	21.9	1111100100
1 2	253	111110100
6	238	1110111000
7	229	1110010100
8	185	1011100100
o o	123	0111101100
10	76	0100110000

Table - 1

5. Results

The output values of V_0 are shown in Table-2. The theoretical values of V_0 were evaluated according to the calculated summation factors and resistor values given in section 3-g. The difference between the actual and the theoretical values can be observed and compared in Figure-2.

Requisitions Received At Retail Sector	Summation Factor	Theoretical V _o (volts)	Observed V (volts)
80	• 312	.667	.6
95	•371	- 391	- 38
160	.625	.801	8
225	.878	1.990	2.1
249	.972	2.432	2.6
253	•988	2.50	2.7
238	.929	2.230	2.4
229	.894	2.07	2.2
185	.722	1.257	1.4
123	.480	.120	.6
76	.296	745	7

Table-2

6. Conclusion and Future:

This is not a complete analog-digital-analog interface, because analog to digital conversion was not dealt with in the project. This project can be considered as a preliminary study of a simple hybrid system.

The model could easily be wired to a small computer so that the table values of complex functions would be evaluated and stored directly from a digital computer.

The addressing into a particular word of the memory must be achieved directly from the analog computer. Voltage levels from analog device must be converted into digital signals (bits) by an analog-digital converter. The Table Function Generator is a primitive start to an Analog-Digital interface. This model can be extended to programmable function generators and a separate control unit can monitor the digital and analog signals. Eventually we can end up with a design in which programmed minicomputers may be controlling the analog devices with the aid of small interfaces between them, thus such a design will be a complete hybrid system.





Figure-2



	Next			
-	Address D., of 7493		L_	
			 	•
	FOR WRITE HODE	č-	ò	· ·
	Sw1 = 0			r 1
	DATA FOL	DATA CAN CHANGE	DATA CAN CHANGE	
	Sw1=0			
	DATA FOR			
	READ NODE	DATH NOT VALID - HIGH	MPEDENCE	:
	· · · · · · · · · · · · · · · · · · ·			
	· · ·			
86 · .	NEXT ADDRESS Dout OF 7493 leycle = 16 elock cycle			
	· · · ·			
	INVERTED OUTPUT OF 7490. SIGNAL			
	is divided by 10			
•	LARN LATCH	Π		
			· · · · · · · · · · · · · · · · · · ·	•
	IO BIT DATA			• •
	IS AVAILABLE IN THE OUTPUT	LII IO BIT WORD	U Ne XI WORD	• - 4 7
	: е			•
			; · · · · · ·	
	•••	ΑΡΡΕΝΟΙΧ ΙΙ ΤΙΜΠ	NG CHART	
				•

•

.

1 . •

APPENDIX-3

The "Retail Sector Inventory Problem" formulated in Dynamo Model was described in <u>Dynamo III User's Manual</u> by Alexander L. Pugh.

the model can be formulated in ordinary differential equations as follows:

- (1) $\frac{d (UOR)}{dt} = RRR \frac{1}{DFR}$ (UOR) Initial Condition of UOR = (DFR) (RRR) (2) $d (IAR) = DELAY(PSR) = \frac{UOR}{DFR}$ (3) IDR = (AIR) (SMOOTH) (RRR)
- (4) PRS = SMOOTH (RRR) + (IDR IAR)/DIR

where UOR = Unfilled Orders SSR = Shipment Sent from Retail DFR = Delay Filling Orders IDR = Inventory Desired at Retail PSR = Purchase Orders Sent from Retail DIR = Delay Adjusting Inventory IAR = Actual Inventory at Retail RRR = Requisitions Received at Retail SMOOTH = Smoothing function for requisitions to give more weight to the recently received items to make realistic prediction. DELAY = Pipeline Delay Function.

Four equations described above are not suitable for easy implementation to an analog device with several user defined functions. The Table Function Generator built in this project can easily be used to generate the proper signals to the analog device. 3B. A Music Box - A Syndetic Computing Project

The intent of this project was to construct a music box using IC chips. The music box would play a melody (with harmony) that would be stored in a memory and would repeat itself.

Twelve oscillators, made with operational amplifiers and tuned with capacitors and resistors, provided the following chromatic pitches:



The following diagram shows the construction of an oscillator:



The following table gives the frequency for each of the twelve pitches and the capacitor and resistors needed to obtain that frequency:

<u>pitch</u>	freq.	capacitor	resistor	
c #	277	.015 mfd	150k, 8.2k, 10k var.	•
đ	294	.068	27k, 10k var.	
d #	311	.033	47k, 4.7k, .lk	
e '	33 0	.033	5 6k, 1. 8k	
f	349	.033	43k, 8.2k	
f #	370	.047	36k, 5.6k, 10k var.	
g	392	.047 .	36%, 10% var.	
g #	415	.047	27k, 10k var.	
a	440	.015	75k, 12k, 10k var.	

<u>pitch</u>	freq.	capacitor	resistor	
a #	466	.033 mfd	4.7k, 10k,	10k var.
Ъ	494	.033	27k, 6.8k,	lOk var.
с	523	.068	10k, 4.7k,	lOk var.

A switch for turning each pitch on or off, i.e., permitting or inhibiting the oscillator's current flow into an output bus, was built with a transistor:



When the pitch line is low, current flows from the oscillator to the output bus, causing that pitch to 'sound'. And, to have silence, all the pitch lines must be high.

The pitch and rhythmic aspects of the melody are stored in a one-bit memory. Sixteen bits of memory are needed for every sounding note or chord of the melody: 12 bits for control of the pitch lines to the oscillators and four bits for control of the duration. The four duration bits are loaded into a presettable counter which counts down and gives a signal when the counting reaches zero, when the duration for that note or chord is complete.

Sixteen durations are possible; one duration $(1111_2 - which$ would be used infrequently) was used to signal completion of the melody. If a sixteenth note has a duration of one count, then

these 15 durations can be used by the melody:

A presettable counter, set to count to 16, controls the access of 16 bits of memory at a time. Memory is addressed 16 times and there are 16 shifts of memory data into a serial-toparallel register. Since there is a delay between the addressing of a memory word and the appearance of the data at the memory output, the addressing and shifting pulses must be staggered; this is done by a decoder whose input is the output of a counter counting a fast clock.



This shifting is done first so that the data in location zero can be accessed.

While a note or chord is sounding, the 16 bits for the next note are being accessed and shifted into a register. This procedure of addressing and shifting 16 bits must be completed within the shortest duration which is one count. The decoder also pulses a counter which divides the count by 16 and this division becomes the input to the presettable duration counter.



Once the 16 bits for the next note have been accessed, the addressing and shifting are inhibited by a latch until the present sound event is finished. A signal of completion from the duration counter enables a control circuit which performs the following operations in order:

- load the 15 bits for the next sound event from the serial-to-parallel register into another register.
- stop the counting that controls the shifting and addressing
- load the presettable counter that cycles 16 times for the shifting and addressing
- 4. load the duration counter
- 5. reset the divider counter
- open the latch to permit the address and shift operations
- 7. start the counting for shifting and addressing
- 8. disable the control circuit

The ordering of these events is controlled by a decoder fed by a counter which counts a fast clock.

The timing diagram for the control circuit is on the next page.



When 1111₂ is encountered as the programmed duration, the initial circuit is activated and this begins a repeat of the melody. The initial circuit clears the address counters and the shift registers and sets the control circuit latches. It enables the clock and then activates the control circuit which will reset the 16-cycle counter and the shift/address decoder counter. The initial circuit can also be activated by a 'start' pulser. The music box listener may halt the melody with a 'stop' pulser that disables the clock.

The initial circuit timing diagram is on the following page.

clock: (low - disabled)		
start pulser		
reset address registers		
clear serial-to- parallel registers		
clear parallel-to- parallel registers		
set control and address latches		
open control circuit		
start clock	·	

A voltage controlled oscillator is the clock for the music box circuit.



Since this clock eventually is used by the duration counter, the frequency of this clock determines the tempo of the music and is dependent on the capacitor value.

The first melody that was played on this music box was an adapted excerpt from "Rambling Wreck" by Frank Roman. The music played was the following:


It was programmed to sound like this:



The rests between repeated notes help to articulate the notes. This is the actual program as stored in memory; the horizontal lines indicate the location of bar lines.

PITCHES

DURATION

с	Ъ	a <i>∄</i>	a	s /‡	g	$\mathbf{f}_{H}^{(i)}$	f	е	$\mathbf{d}_{n}^{\mathcal{U}}$	đ	$\mathbf{C}_{\widetilde{H}}^{T_{2}^{\prime}}$		I.		
0	l	l	1	0	1	l	1	1	1	1	1	0	1	0	0
1	l	0	1	l	1	1	1	l	0	1	1	0	0	1	0
1	1	l	1	0	1	1	1	1	1	1	1	0	0	1	1
1	1	1	l	1	l	1	1	1	l	1	1	0	0	0	1
l	l	1	l	0	1	1	l	1	1	l	1	0	0	0	1
1 1	1 1	1 1	1 1	1 0	1 1	1 1	1 1	1 1	1 1	1 1	1 1	0 0	0 1	0 0	1 0
1	1	0	1	1	1	1	1	1	0	1	1	0	0	1	0
0	1	1	1	0	1	l	1	1	1	1	1	0	0	1	1
1	l	l	l	l	1	1	l	1	,l	l	1	0	0	0	l
0	l	l	l	l	l	l	l	1	l	l	1	0	0	0	l
l	l	1	1	l	1	1	l	1	1	l	1	0	0	0	1
0	l	l	l	0	1	1	l	1	1	l	1	0	0	l	0
1	l	0	l	l	1	1	l	1	1	1	1	0	0	1	0
1	1	1	1	0	1	1	1	1	1	1	1	0	0	1	0
1	l	0	1	l	1	1	0	1	l	1	0	0	0	1	0
0	1	1	l	1	1	1	1	1	1	1	1	0	0	1	0
l	1	0	1	1	1	1	0	1	1	1	0	0	0	1	0
1	1	1	1	0	1	1	1	1	0	1	1	0	1	0	0
1	1	1	1	1	0	1	1	1	0	1	1	0	0	1	0
1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0
1	1	l	l	1	l	1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

silence

signal that melody is done



.



CONTROL CIRCUIT

r,

.



< 0



3C. Integrator Comparisons



FICHRE SC. 1





FIGURE 3C.3

104

EAI-180

DRAWN M.P. DATE DESIGNER SHEET SO.

4. INTRODUCTORY LABORATORY MANUAL FOR A COURSE IN SYNDETIC COMPUTATION

The following manual illustrates introductory material for entry into syndetic computing. As shown here, syndetic computing requires a firm footing in digital logic but little or no background in electronics beyond concepts normally covered in sophomore physics.

The experiments covered in this manual are distilled from several quarters of experience. The first manual used was a commercially available "cookbook" which produced marginal results. Detailed "recipes" could be followed without learning anything. This was revealed when students first encountered an early version of the frequency counter experiment which has provided a logical transition from purely digital concepts to issues of analog form data.

This manual has been used in parallel with the "cookbook" for two quarters where students were given the option of using either the completely standard manual or this manual which was in obvious transition. Most students who started out with the "cookbook" ran into trouble with the frequency counter and the following analog related experiments (experiments 26-29). Those who used this manual from the beginning had little or no difficulty in taking the principles illustrated here and extending these into higher order syndetic systems.

These experiments are intended to be performed independently in a building block fashion. Very little guidance is provided in a classroom sense. Major topics are introduced in advance of the scheduled experiments and classroom discussion of problems and experiences is encouraged. A major project is defined by the time

all of the experiments have been performed and then the students are required to design, prototype, test and demonstrate an agreed-upon syndetic system. A course grade of "A" is given only if the end item performs as advertised.

Introductory

SYNDETIC COMPUTING LABORATORY MANUAL An Introduction to Syndetic Analog/Digital Systems

School of Information and Computer Science Georgia Institute of Technology Atlanta, Georgia 30332

Preface

This manual is designed to present the fundamentals of syndetic computing. The fundamental assumption on which this manual is predicated is that information and computer science students with little or no knowledge of electronics can acquire the facility to develop and build syndetic analog/digital systems from standard integrated circuits. This of course presumes that such students have some background in Boolean Algebra or basic logic functions and a notion of basic electric circuits.

There are twenty-nine experiments to be performed in the order presented. Some of these experiments require the knowledge gained in previous experiments to be applied. These previous experiments should have collectively supplied all of the required background material.

This manual is intended for independent, (work at your own pace) individual study in the proximity of a laboratory staff person; however, this staff person should not be expected to give guidance except in cases which have been carefully explored and presented. Answers to most questions will be questions.

Both this laboratory and this manual are in a state of development. Comments, suggestions and corrections are invited; they should be written and given to the laboratory staff person in attendance or forwarded to your course instructor.

INDEX

Experiment 1 The Digi Designer Experiment 2 Inverter Experiment 3 2 Input NAND Gate Experiment 4 Multi Input NAND Gate Experiment 5 2 Input AND Gate Experiment 6 2 Input OR Gate Experiment 7 AND-OR Functions Experiment 8 Exclusive OR Binary Addition Experiment 9 Experiment 10 Binary Subtraction Experiment 11 Arithmetic Logic Unit Experiment 12 Reset-Set Latch Experiment 13 Clocked Latches Experiment 14 D-Type Flip Flop Experiment 15 J-K Master-Slave Flip Flop Experiment 16 Asynchronous Binary Counter Experiment 17 Shift Register Experiment 18 Synchronous Binary Counter UP/DOWN Binary Counter Experiment 19 Experiment 20 Decimal Scaling Experiment 21 BCD-to-Decimal Decoder Experiment 22 BCD to 7-Segment Decoder/Driver Experiment 23 One Shot Multivibrators Data Selector/Multiplexer Experiment 24 Experiment 25 Eipolar Random Access Memory

Experiment 26 Frequency Counter Experiment 27 Inverting Amplifier Experiment 28 Digital to Analog Converters Experiment 29 Analog to Digital Conversion

TITLE: The Digi Designer

PURPOSE: To become familiar with the Digi Designer functions.

PROCEDURE:

- Plug the Digi Designer line cord into a power receptical. Turn the power switch ON. The pilot lamp should turn on. If at any time problems related to the Digi Designer arise report them to your Lab Instructor.
- 2) Prepare 4 test wires. Use Number 24 Solid Wire with 1/4 inch of the insulation striped off each end of the wire. Wires such as this will be used to connect various bread board points in all experiments.
- 3) All lamp monitors should be off at this time.
- 4) The lamp monitors should light as their individual inputs are connect to +5 volts. This lamp on condition represents a logical '1' condition. In general, the +5v source can be thought of as a constan logical '1' source. If a 1000 Ohm resistor (Brown Black Red) is used between the +5v power supply and the input (up to 25 inputs can be supplied by one 1000 Ohm resistor) it will reduce the chances of chip failure due to transients on the power supply. Verify that a 1000 Ohm resistor between +5v and the lamp monitor input results in a '1'.
- 5) The Ground terminal on the Digi designer is a logical '0'. This may be used as a constant '0' input for chips. When a lamp monitor input is connected to Ground the lamp will remain OFF indicating a logical '0'.
- 6) Connect the outputs of the 4 function switches to the 4 lamp monitors
- 7) Verify the following operation on all switches:

Switch	Lamp	logic	level
GRD	OFF	•0•	FALSE
+5 v	ON	111	TRUE

- 8) Connect the 4 outputs from the two push buttons to the lamp monitors.
- 9) Observe the push button operation on the lights. Notice that the two outputs of each switch are complements.
- 10) Rotate the Clock Knob fully counter-clockwise to OFF.
- 11) Connect the 2 Clock ouputs to two lamp monitors.
- 12) Turn the knob one click clockwise. This is a 1Hz Clock source. Note that the two outputs are complements of each other.

13) As the Clock knob is turned clockwise the clock rate increases. Note that at 100Hz (100 pulses per second) individual pulses can no longer be seen.

•

•

.

.

a.

5F.,

TITLE: Inverter

FURPOSE: To observe the simplest logic function. To learn how to find the functional parts of an IC.

PROCEDURE:

- ·

- 1) Flug the SN7404 into the Digi Designer socket so that its two rows of terminals are on opposite sides of the groove in the socket.
- 2) Using the drawings supplied with this experiment find pin 7 of the SN7404 and connect it to ground (GRD) with a short wire. Connect pin 14 to +5.0v (VCC) with another wire.
- 3) Connect an input (pin 1) to one of the logic switches. Connect the related output (pin 2) to one of the lamp monitors.
- 4) Remember that: GND = 'O' And +5v = '1'Light Off = 'O' And Light On = '1'

5) Now verify the inversion function is performed by this chip.

6) Check all other sections of the chip.







SPE	ED/PA	CKAGE AV	AILABILITY
54	F,W	74	A.F
54H	F,W	74H	A,F
54LS	F,W	74LS	A,F
54S	F.W	74S	A,F

PIN CONFIGURATION





SWITCHING CHARACTERISTICS $v_{CC} = 5V$, $T_A = 25^{\circ}C$

		54/74		54/74H			54/74LS			54/74S			
TEST CONDITIONS		Ci_⇔15pF Ri_⇒400Ω			CL 25pF RL 280Ω			$C_L \ge 15 pF$ $R_L \ge 2k\Omega$			CL=15pF RL=280Ω		
PARAMETER	MIN	ТҮР	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	ТҮР	MAX	TINU
Propagation delay time tpLH Low-to-high		12	22		6	10		5	15	2	3 CL=50r 4.5	 4.5 	ns
tp _{HL} High-to-low		8	15		6.5	10		9	15	2	3 CL≞50p 5	5 5 	ns

Load circuit and typical waveforms are shown at the front of section.

TITLE: 2 Input NAND Gate

PURPOSE: To observe input/output logic relations of a simple NAND Gate.

PROCEDURE:

- Plug a SN7400 into the Digi Designer socket. Pin 7 is Ground. Pin 14 is +5.0v.
- 2) Connect NAND gate inputs (pins 1 and 2) to logic input switches. Connect the output of that gate (pin 3) to a lamp monitor.

3) Verify NAND function with a truth table.

4) Check other sections.

5) Verify that the inversion function can be done with this gate.

QUAD 2-INPUT NAND GATE

LOGIC AND CONNECTION DIAGRAM

DIP (TOP VIEW)



Positive logic: $Y = \overline{AB}$



SCHEMATIC DIAGRAM (EACH GATE)

.



Component values shown are typical

RECOMMENDED OPERATING CONDITIONS

	9	N00XM/5400	хм	9			
PARAMETER	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Supply Voltage V _{CC}	4.5	5.0	5.5	4.75	5.0	5.25	Volts
Operating Free-Air Temperature Range	-55	25	125	0	25	70	°C
Normalized Fan-Out from Each Output, N		<u> </u>	10	1		10	U.L.

X = package type; F for Flatpak, D for Ceramic Dip, P for Plastic Dip, See Packaging Information Section for packages available on this product.

ELECTRICAL CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (Unless Otherwise Noted)

			LIMITS			1	TEST
SYMBOL	PARAMETER	MIN.	TYP. (Note 2)	MAX.	UNITS	TEST CONDITIONS (Note 1)	FIGURE
VIH	Input HIGH Voltage	2.0			Volts	Guaranteed Input HIGH Voltage	1
VIL	Input LOW Voltage			0.8	Volts	Guaranteed Input LOW Voltage	2
Voн	Output HIGH Voltage	2.4	3.3		Volts	$V_{CC} = MIN., I_{OH} = 0.4 \text{ mA}, V_{IN} = 0$	8V 2
VOL	Output LOW Voltage		0.22	0.4	Volts	VCC = MIN., IOL = 16 mA, VIN = 2.0) V 1
- In s	Input HIGH Current			40	μA	VCC = MAX., VIN = 2.4 V Fach In	aut A
чн	input man conent			1.0	mA	V _{CC} = MAX., V _{IN} = 5.5 V	
- IIL	Input LOW Current			-1.6	mA	V _{CC} = MAX., V _{IN} = 0.4 V, Each In	out 3
	Output Short Circuit Current	-20		-55	mA	9N00/5400	6
¹ OS	(Note 3)	-18		-55	mA	9N00/7400 VCC - MAX.	
ICCH	Supply Current HIGH		4.0	8.0	mA	V _{CC} = MAX., V _{IN} = 0V	6
ICCL	Supply Current LOW		12	22	mA	V _{CC} = MAX., V _{IN} = 5.0 V	6

SWITCHING CHARACTERISTICS (TA = 25°C)

SYMBOL	PARAMETER		LIMITS		UNITS	TEST CONDITIONS	TEST
STMBUL	FARAMETER	MIN.	TYP.	MAX.			FIGURE
^t PLH	Turn Off Delay Input to Output		11	22	ns -	V _{CC} = 5.0 V	_
^t ₽HL	Turn On Delay Input to Output		7.0	15	กร	R _L = 400Ω	

NOTES:

(1) For conditions shown as MIN, or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.

(2) Typical limits are at $V_{CC} = 5.0 V, 25 C.$

(3) Note more than one output should be shorted at a time.

TITLE: Multi Input NAND Gate

PURPOSE: To observe a complex gate.

PROCEDURE:

- 1) Plug the SN7420 into the socket. Pin 7 is Ground and pin 14 is +5v.
- 2) Verify the 4 input NAND operation with truth tables generated using the input switches and indicator lights.
- 3) Verify that this gate can be used as a 3 Input NAND, 2 Input NAND, or Inverter.



Component values shown are typical.

RECOMMENDED OPERATING CONDITIONS

	9	N20XM/5420	хм	g			
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Supply Voltage V _{CC}	4.5	5.0	5.5	4.75	5.0	5.25	Volts
Operating Free-Air Temperature Range	-55	25	,125	0	25	70	°C
Normalized Fan Out From Each Output, N	· · · · · ·		10			10	U.L.

X = package type; F for Flatpak, D for Ceramic Dip, P for Plastic Dip. See Packaging Information Section for packages available on this product.

ELECTRICAL CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (Unless Otherwise Noted)

			LIMITS		UNITS Volts		TECT
SYMBOL	PARAMETER	MIN.	TYP. (Note 2)	MAX.		TEST CONDITIONS (Note 1)	FIGURE
ViH	Input HIGH Voltage	2.0				Guaranteed Input HIGH Voltage	1
VIL	Input LOW Voltage	1		0.8	Volts	Guaranteed Input LOW Voltage	2
∨он	Output HIGH Voltage	2.4	3.3		Volts.	VCC = MIN., IOH = -0.4 mA, VIN = 0.8 V	2
VOL	Output LOW Voltage		0.22	0.4	Volts	V _{CC} = MIN., I _{OL} = 16 mÅ, V _{IN} = 2.0 V	1
				40	μA	V _{CC} = MAX., V _{1N} = 2.4 V	4
чн	input nigh curait			1.0	mA	V _{CC} = MAX., V _{IN} = 5.5 V	
11	Input LOW Current			-1.6	mA	V _{CC} = MAX., V _{IN} = 0.4 V, Each Input	3
······	Output Short Circuit Current	-20		-55	mA	9N20/5420	
OS	(Note 3)	-18		-55	mA	9N20/7420 CC = MAX.	5
1ссн	Supply Current HIGH	1	2.0	4.0	mA	V _{CC} = MAX., V _{IN} = 0 V	6
CCL	Supply Current_LOW		6.0	11	mA	V _{CC} = MAX., V _{IN} = 5.0 V	6

SWITCHING CHARACTERISTICS (TA = 25 C)

SYMBOL	PARAMETER		LIMITS		UNITS	TEST CONDUTIONS	TEST	
STWBUL	PANAMETER	MIN.	TYP.	MAX.			FIGURE	
^t PLH	Turn Off Delay Input to Output		12	22	ns	V _{CC} = 5.0 V		
^t PHL	Turn On Delay Input to Output	[8.0	15	ns	նլ = 15 թԲ Բլ = 400Ω		

NOTES:

(1) For conditions shown as MIN, or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.

(2) Typical limits are at $V_{CC} = 5.0 \text{ V}$, 25°C (3) Not more than one output should be shorted at a time.

TITLE: 2 Input AND Gate

PURPOSE: To look at the AND function.

PROCEDURE:

- 1) Plug the SN7408 into the socket. Connect power and Ground. Connect any section to input switches and a lamp monitor.
- 2) Verify AND operation.
- 3) Check other sections.
- 4) Make an AND function block out of a 2 Input NAND (SN7400) and an Inverter (SN7404).

5) Verify proper logical operation of this block with switches and light:

QUAD 2-INPUT AND GATE LOGIC AND CONNECTION DIAGRAM SCHEMATIC DIAGRAM (EACH GATE) DIP (TOP VIEW) FLATPAK (TOP VIEW) Vcc O LE-C 1610 2540 1011 o٠ operation A (O-B O Positive logic: Y = AB 3512 O GND

Component values shown are typical.

RECOMMENDED OPERATING CONDITIONS

· · · · · · · · · · · · · · · · · · ·	9	N08XM/5408	хм	9			
PARAMETER	MIN.	TYP.	MAX.	MIN,	TYP.	MAX.	
Supply Voltage V _{CC}	4.5	5.0	5.5	4.75	5.0	5.25	Volts
Operating Free-Air Temperature Range	55	25	125	0	25	70	^C
Normalized Fan Out from Each Output, N			10			10	U.L.

X = package type; F for Flatpak, D for Ceramic Dip, P for Plastic Dip. See Packaging Information Section for packages available on this product. ELECTRICAL CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (Unless Otherwise Noted)

SYMBOL VIH VIL VOH VOL IIH IIH			LIMITS				TEST
	PARAMETER	MIN.	TYP. (Note 2)	MAX.	UNITS	TEST CONDITIONS (Note 1)	FIGURE
VIH	Input HIGH Voltage	2.0			Volts	Guaranteed Input HIGH Voltage	75
VIL	Input LOW Voltage			0.8	Volts	Guaranteed Input LOW Voltage	77
VOH	Output HIGH Voltage	2.4	. ,		Volts	V _{CC} = MIN., I _{OH} = -0.8 mA, V _{IH} = 2.0 V	75
VOL	Output LOW Voltage	1		0.4	Volts	V _{CC} = MIN., I _{OL} = 16 mA, V _{IL} = 0.8 V	77
				40	μА	V _{CC} = MAX., V _{IN} = 2.4 V	70
ЧН	Input HIGH Current			1.0	mA	V _{CC} = MAX., V _{IN} = 5.5 V	/8
I HL	Input LOW Current	-		-1.6	mA	VCC = MAX., VIN = 0.4 V, Each Input	79
los	Output Short Circuit Current (Note 3)	-40		-100	mĄ	V _{CC} = MAX.	80
ICCH	Supply Current HIGH			20	mA	V _{CC} = MAX., V _{IN} = 5 V	81
ICCL	Supply Current LOW		· · · ·	32	mA	V _{CC} = MAX., V _{IN} = 0 V	81

SWITCHING CHARACTERISTICS (TA = 25 C)

EXMODI			LIMITS			TECT CONDITIONS	TEST
STINBUL	PARAMETER	MIN.	TYP.	MAX.	UNITS	TEST CONDITIONS	FIGURE
t₽LH	Turn Off Delay Input to Output		17.5	40	ns	$V_{CC} = 5.0 V$	
. ^t ₽HL	Turn On Delay Input to Output		12	25	ns	RL = 400Ω	

NOTES:

(1) For conditions shown as MIN, or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.

(2) Typical limits are at V_{CC} 5.0 V, 25 C

(a) Not come there are not a standard by about of a time

TITLE: 2 Input OR Cate

PURPOSE: To observe basic OR functions.

PROCEDURE:

1) Mount SN7432 in Digi Designer socket. Fin 7 is Ground and pin 14 is +5.0 volts.

2) Verify OR operation.

3) Check other sections of the chip.

- 4) Using only 2 Input NAND Gates (SN7400) build a block which does the OR function. (Truth tables and/or deMorgans thrm. should help)
- 5) Verify the new blocks OR function.
- 6) Mount a SN7402 (2 Input NOR) in the socket. Connect power as before.

7) verify 2 Input NOR operation.

8) Modify your CR gate made from NAND gates to do NOR function.

9) Verify operation.

QUAD 2-INPUT NOR GATE

LOGIC AND CONNECTION DIAGRAM

DIP (TOP VIEW)



Positive logic: $Y = \overline{A+B}$





.

Component values shown are typical.

RECOMMENDED OPERATING CONDITIONS

	9	N02XM/5402	ХМ	9				
PARAMETER	MIN.	TYP. MAX.		MIN.	TYP. MAX.		UNITS	
Supply Voltage VCC	4.5	5.0	5.5	4,75	5.0	5.25	Volts	
Operating Free-Air Temperature Range	55	25	125	0	25	70	°C	
Normalized Fan-Out from Each Output, N			10		· · · · ·	10	U.L.	

X = package type; F for Flatpak, D for Ceramic Dip, P for Plastic Dip. See Packaging Information Section for packages available on this product. ELECTRICAL CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (Unless Otherwise Noted)

			LIMITS				
SYMBOL	PARAMETER	MIN.	TYP. {Note 2}	MAX.	UNITS	TEST CONDITIONS (Note 1)	FIGURE
⊻ін	Input HIGH Voltage	2.0			Volts	Guaranteed Input HIGH Voltage	8
VIL	Input LOW Voltage		_	0.8	Volts	Guaranteed Input LQW Voltage	9
VOH	Output HIGH Voltage	2.4	3.3		Volts	V _{CC} = MIN., I _{OH} = -0.4 mA, V _{IN} = 0.8 V	9
VOL	Output LOW Voltage		0.22	0.4	Volts	V _{CC} = MIN., I _{OL} = 16 mA, V _{IN} = 2.0 V	10
				40	μА	V _{CC} = MAX., V _{IN} = 2.4 V	12
ЧН				1.0	mA	VCC = MAX., VIN = 5.5 V	
	Input LOW Current			-1.6	mA	V _{CC} = MAX., V _{IN} = 0.4 V, Each Input	11
	Output Short Circuit Current	-20		~55	mA	9N02/5402	
'0S	(Note 3)	18		55	mA	9N02/7402 VCC - MAX.	1.3
1ссн	Supply Current HIGH		8.0	16	mA	VCC = MAX., VIN = 0 V	14
ICCL	Supply Current LOW		14	27	mA	V _{CC} = MAX., V _{IN} = 5.0 V	14

SWITCHING CHARACTERISTICS (TA = 25°C)

SYMBOL	BARANETER	LIMITS		TEST CONDITIONS	TEST		
	PARAMETER	MIN.	TYP.	MAX.		TEST CONDITIONS	FIGURE
^t PLH	Turn Off Delay Input to Output		12	22	ns	V _{CC} = 5.0 V	<u> </u>
^t PHL	Turn On Delay Input to Output		B.0	15	ns	RL = 400Ω	

NOTES:

(1) For conditions shown as MIN, or MAX,, use the appropriate value specified under recommended operating conditions for the applicable device type.

(2) Typical limits are at $V_{CC} = 5.0 \text{ V}, 25^{\circ}\text{C}$. (3) Not more than one output should be shorted at a time.

TITLE: AND-OR Functions

PURPOSE: TO look at possible implementations of AND-OR functions.

PROCEDUPE:

1) Mount a SN7408 and a SN7432 on the Digi Designer. Connect Power and Ground to the chips.

2) Build this circuit:



3) Connect 4 input switches and a output indicator light.

4) Verify circuit operation.

- 5) Using only a package of 2 Input NAND Gates (SN7400) and deMorgans theorem duplicate this function.
- 6) Verify circuit operation.

TITLE: Exclusive OR

PURFOSE: To look at the Exclusive OR function (XOR).

PROCEDURE:

1) Using AND, NAND, OR, and Inversion gates design a circuit which will act as a 2 input XOR block.

2 Input XOP Truth Table

XOR В * A × * * * * 0 0 0 * 0 1 * 1 1 0 1 n 1 1

2) Implement it on the Digi Designer using SN7408, SN7400, SN7432, and/or SN7404 gates in any combination needed.

3) Verify the operation of your block.

4) Mount a SN7486 gate on the Digi Designer.

5) Verify that any section of this chip also performs a XOR function.

6) Change XOR to XNOP (Not Exclusive OR) by inverting one input



7) Verify operation.

8) Change XOR to XNOR by inverting output.



9) Verify operation.

10) Observe that the XNOR function is the Equivalence function.

LPTTL/SSI 9L86 LOW POWER QUAD EXCLUSIVE OR GATE

DESCRIPTION – The Low Power TTL/SSI 9L86 consists of four Exclusive OR Gates. Designed for low power, medium speed operation, the 9L86 is useful in large number of code conversion, parity generation/checking and comparison applications. The exclusive OR gate produces an output when the inputs are complementary. The Boolean expression for the device is: $Z = A\overline{B} + \overline{A}B$.

TYPICAL PROPAGATION DELAY OF 25 ns
TYPICAL POWER DISSIPATION OF 25 mW

- INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS
- ALL CERAMIC "HERMETIC" 14-LEAD DUAL IN-LINE AND FLAT PACKAGES
- TTL COMPATIBLE

PIN NAMES

INPUTS (Pins 1, 2, 4, 5, 9, 10, 12, 13) OUTPUTS (Pins 3, 6, 8, 11) LOADING HIGH LOW 0.75 U.L. 0.38 U.L. 10 U.L. 2.5 U.L.

1 Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW.





LOGIC SYMBOL

FLATPAK (TOP VIEW)



TITLE: Binary Addition

PURPOSE: To look at Dinary Half-adder and Full-adders

PROCEDURE:

 Design a logic circuit which will perform the function of a Binary Half-adder.

1 Plus 0 and Carry = 11 = Plus -1 and Carry = 01 0 Carry = 0O Plus 0 = 0 and

• A truth table should make the solution obvious.

2) Implement your design.

3) Verify proper operation.

4) Build a full-adder using your Half adder design.



- 5) Verify this change to be a true Full-adder with the aid of truth tables.
- 6) Verify the Full-adder operation.

TITLE: Binary Subtraction

PURPOSE: To look at a less common arithmetic function implementation. PROCEDURE:

1) Design a logic circuit which will perform the function of Binary Half-subtraction.

Α	Minus	В	=	Difference	and	Borrow
0		0		C		0
0		1		1		1
1		0		1		0
1		1		0		۵

Note similarities to Half-adder.

2) Implement your design.

3) Verify its operation.

- 4) Design a Full-subtractor using two Half-subtractors.
- 5) Implement your Full-subtractor design.

6) Verify its operation.

TITLE: Arithmetic Logic Unit

PURPOSE: To introduce a complex function block and develop skills in reading data sheets.

PROCEDURE:

- 1) Mount SN74181 Arithmetic Logic Unit (ALU) and a SN7404 on the Digi Designer.
- 2) Extra switch inputs will be needed to allow easy access to the 5 function select lines and the two 4 input data paths. Use extra Digi Designers if availiable (TIE ALL GROUNDS TOGETHER) or use external switch register. It will be convienient to use a push button for the carry input.
- 3) There are 8 outputs associated with this chip. Display the function outputs on the lamp monitors. Using SN7404 Inverters build four more logic monitors.



Verify proper LED (Light Emitting Diode) orientation by applying a high logic level to the inverters input. This will cause the LED to light if it is properly oriented. If it does not light reverse its two leads and try again. Use these indicators for carrys and A=B.

4) Verify all truth tables in the data sheet.

4-BIT ARITHMETIC LOGIC UNIT

DESCRIPTION

The S54/N74LS181 arithmetic logic unit (ALU)/function generators have a complexity of 75 equivalent gates on a monolithic chip. These circuits perform 16 binary arithmetic operalions on two 4-bit words as shown in Tables 1 and 2. These operations are selected by the four function-select lines (S0, S1, S2, S3) and include addition, subtraction, decrement, and straight transfer. When performing arithmetic manipulations, the internal carries must be enabled by applying a low-level voltage to the mode control input (M). A full carry look-ahead scheme is made available in these devices for fast, simultaneous carry generation by means of two cascade-outputs (pins 15 and 17) for the four bits in the package. When used in conjunction with the 182 full carry look-ahead circuit, high-speed arithmetic operations can be performed. 54/74181

If high speed is not of Importance, a ripple-carry input $\{C_n\}$ and a ripple-carry output $\{C_n \neq 4\}$ are available. However, the ripple-carry delay has also been minimized so that arithmetic manipulations for small word lengths can be performed without external circuitry.

The S54/N74LS181 will accommodate active-high or active-low data if the pin designations are interpreted as follows: Subtraction is accomplished by 1's complement addition where the 1's complement of the subtrahend is generated internally. The resultant output is A-B-1 which requires an end-around or forced carry to provide A--B.

The S54/74LS181 can also be utilized as a comparator. The A = B output is internally decoded from the function outputs (F0, F1, F2, F3) so that when two words of equal magnitude are applied at the A and B inputs, it will assume a high level to indicate equality (A = B). The ALU should be in the subtract mode with $C_n = H$ when performing this comparison. The A = B output is open-collector so that it can be wire-AND connected to give a comparison for more than four bits. The carry output (C_n +4) can also be used to supply relative magnitude information. Again, the ALU should be placed in the subtract mode by placing the function select inputs S3, S2, S1 S0 at L, H, H, L, respectively.

These circuits have been designed to not only incorporate all of the designer's requirements for arithmetic operations, but also to provide 16 possible functions of two Boolean variables without the use of external circuitry. These logic functions are selected by use of the four function-select inputs (S0, S1, S2, S3) with the mode-control input (M) at a high level to disable the internal carry. The 16 logic functions are detailed in Tables 1 and 2 and include exclusive-OR, NAND, AND, NOR, and OR functions.



FUNCTIONAL BLOCK DIAGRAM

5jijn0fiCS

4-BIT ARITHMETIC LOGIC UNIT

ALU SIGNAL DESIGNATIONS

The S54/N74LS181 can be used with the signal designations of either Figure 1 or Figure 2.

The logic functions and arithmetic operations obtained with signal designations as in Figure 1 are given in Table 1; those obtained with the signal designations of Figure 2 are given in Table 2.

54/74181

.

...

PIN NUMBER	2	1	23	22	21	20	19	18	9	10	11	13	7	16	15	17
Active-high data (Table 1)	Ao	Bo	A1	B1	A ₂	B ₂	A3	B ₃	Fo	F1	F2	F3	С'n	Cn+4	X	Y
Active-low data (Table 2)	Āo	B ₀	Ā1	B ₁	Ā2	B ₂	Ā3	Бз	Ē ₀	Ē1	Ē ₂	Ē3	Cn	Cn / 4	P	Ĝ



TABLE 1

					ACTIVE-HIGH DATA						
SELECTION				M. H	M=L: ARITHMETIC OPERATIONS						
S 3	S 2	S 1	S0	LOGIC	C _n -H (NO CARRY)	C _n - L (WITH CARRY)					
L	L	L	L	F-Ā	F≃A	F=A PLUS 1					
L	L	L	н	F=A+B	F≖A+B	F=(A+B) PLUS 1					
L	L	н	L	F-AB	F=A+B	F=(A+B) PLUS 1					
L,	L	н	н	F=0	F=MINUS 1 (2's COMPL)	F ZERO					
L	н	L	L	F + AB	F∞A PLUS AB	F-A PLUS AB PLUS 1					
L	н	L	н	F≖B	F-(A+B) PLUS A	F=(A-B) PLUS AB PLUS 1					
L	н	н	L	F= A 🕀 B	F-A MINUS B MINUS 1	F≖ A MINUS B					
L	н	н	н	F-AB	F≖AB MINUS 1	F= AB					
н	L	L	L	F=Ā+B	F-A PLUS AB	F=A PLUS AB PLUS 1					
н	L	L	н	F-A @ B	F=A PLUS B	FA PLUS B PLUS 1					
н	L	н	ι	F=B	F-(A+B) PLUS AB	F~(A+B) PLUS AB PLUS 1					
н	L	н	н	F-AB	F-AB MINUS 1	• F= A9					
н	н	L	ι	F =1	F-A PLUS A*	F - A PLUS A PLUS 1					
н	н	L	н	F≈A+B	F=(A+B) PLUS A	F=(A+B) PLUS A PLUS 1					
н	н	н	L	F= A+ B	F∞(A+B) PLUS A	F. (A+B) PLUS A PLUS 1					
н	Η´	н	н	F=A		F=A					

*Each bit is shifted to the next more significant position.

IITLE: Reset-Set Latch

PURPOSE: To examine the most basic memory element.

PROCEDURE:

1) Build a Reset-Set Latch



Familiarize your self with the latch operation during state changes.

- Use the two normally hi push button outputs for the inputs to the Latch. Display the two inputs and the two outputs on the four lamp monitors.
- 3) Develop a complete truth table for the latch.
- 4) Observe the ambiguous state of both Set and Reset being low.

5) Note that changes happen on high to low transtions.

TITLE: Clocked Latches

PURPOSE: To examine data latches with a clock input. PROCEDURE:

1) Build a Clocked Set-Reset Latch.



2) Observe the operation of this latch.

3) Note that changes only happen while the clock level is high.

4) Build the D-Type Latch from the Set-Reset Latch.



- 5) Observe the operation of this Latch.
- 6) Mount a SN7475 on the Digi Designer. Note that power and ground are not on the standard pins.
- 7) Observe the operation of one section.
- 8) is this a D-Type Latch ?
QUAD BISTABLE LATCH

SWITCHING	CHARACTERISTICS	VCC = 5V, TA = 25°C	
-----------	-----------------	---------------------	--

				54/74	1	:	54/74L	.s	
TEST CONDITIONS			C F	:L≖15 L_=40	pF 0Ω		ະ ຊູ - 2k	pF Ω	
PARAMETER	FROM INPUT	TO OUTPUT	MIN	түр	MAX	MIN	Түр	MAX	UNIT
Iw Width of enabling pulse						20			
tSetup Input setup time High level				7	20	20			
Low level Hold input hold time High Level			0	14	20	0		•	
Low level Propagation delay time			0	6					
IPLH Low-to-high IPHL High-to-low	D	a		16 14	30 25		15 9	27 17	ns
IPLH Low-to-high	D	ā		24 7	40 15		12 7	20 15	ns
tpHL Low-to-high tpHL High-to-low	G	٥		16 7	30 15		15 14	27 25	ns
IPLH Low-to-high IPHI High-to-low	G	ã		16 7	30 15		16 7	30 15	ns

Load circuit and typical waveforms are shown at the front of section

PARAMETER MEASUREMENT INFORMATION



VOLTAGE WAVEFORMS

NOTES:

- NOTES: A The pulse generators have the following characteristics: $Z_{OUT} = 50$ R; for pulse generator A, PRR ≤ 500 kHz, for pulse generator B, PRR $\leq I$ MHz. Positions of D and G input pulses are varied with respect to each other to verify setup times. B C_L includes restributed and μ g capacitance. C All blodes are 11/3064 D When measuring propagation delay times from the D input, the corresponding G input

must be held high

E Vref = 1 3V.

SPEED/PACKAGE AVAILABILITY

54	F	74	B,F
54LS	F,W	74LS	B,F

DESCRIPTION

This latch is Ideally suited for use as temporary storage for binary information between processing units and input/output or indicator units. Information present at a data (D) Input is transferred to the Q output when the enable (G) is high and the Q output will follow the data input as long as the enable remains high. When the enable goes low, the information (that was present at the data input at the time the transition occurred) is retained at the Q output until the enable is permitted to go high.

BLOCK DIAGRAM



TRUTH TABLE (Each Latch) LOGIC 54/74

	(Each Latch)] -
t _n	t _n	+1	
D	a	ā	7
1	1	0	
0	0	1	

54/7475

NOTES

minal

254

1 , r_n = bit time before clock purce 2 , $r_n + 1$ = bit time after clock purce 3. These voltages are with respect to network ground terms

54/74LS

INP	INPUTS		PUTS
D	G	Q	ā
L	н	L	н
н	н	н	L
x	L	0	ā

H = high level L = low revel X - 100 . Hevar

 \boldsymbol{D}_0 = the level of $\boldsymbol{\Omega}$ before the high-to-low transition of \boldsymbol{G}



FLIP-FLOP LOGIC DIAGRAM







PIN CONFIGURATION



TRIPLE 3-INPUT NAND GATE

LOGIC AND CONNECTION DIAGRAM

DIP (TOP VIEW)

Positive logic: $Y = \overline{ABC}$



FLATPAK (TOP VIEW)





Component values shown are typical.

RECOMMENDED OPERATING CONDITIONS

	9	N10XM/5410	XM	9N10XC/7410XC			
PARAMETER	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	UNITS
Supply Voltage V _{CC}	4.5	5.0	5.5	4,75	5.0	5.25	Volts
Operating Free-Air Temperature Range	55	25	125	0	25	70	°C
Normalized Fan-Out from Each Output, N			10			10	U.L.

X = package type; F for Flatpak, D for Ceramic Dip, P for Plastic Dip. See Packaging Information Section for packages available on this product. ELECTRICAL CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (Unless Otherwise Noted)

			LIMITS				TCOT
SYMBOL	PARAMETER	MIN.	TYP. (Note 2)	MAX.	UNITS	TEST CONDITIONS (Note 1)	FIGURE
 VIН	Input HIGH Voltage	2.0			Volts	Guaranteed Input HIGH Voltage	1
VIL	Input LOW Voltage			0.8	Volts	Guaranteed Input LOW Voltage	2
VOH	Output HIGH Voltage	2.4	3.3		Volts	VCC = MIN., IOH = -0.4 mA, VIN = 0.8 V	2
VOL	Output LOW Voltage		0.22	0.4	Volts	V _{CC} = MIN., I _{OL} = 16 mA, V _{IN} = 2.0 V	1
				40	μA	V _{CC} = MAX., V _{IN} = 2.4 V	
ЧH				1.0	mA	$V_{CC} = MAX., V_{IN} = 5.5 V$	
կլ	Input LOW Current			-1.6	mA	V _{CC} = MAX., V _{IN} = 0.4 V, Each Input	3
	Output Short Circuit Current	-20		-55	mA	9N10/5410	6
'0S	(Note 3)	-18		-55	mA	9N10/7410 VCC = MAX.	5
Іссн	Supply Current HIGH		3.0	6.0	mA	V _{CC} = MAX., V _{IN} = 0 V	6
ICCL	Supply Current LOW		9.0	16.5	mA	V _{CC} = MAX., V _{IN} = 5.0 V	6

SWITCHING CHARACTERISTICS (TA = 25°C)

	BADAMETED	LIMITS				TEST	
SYMBOL PARAMETER MIN. TYP. MAX. UNITS			FIGURE				
^t PLH	Turn Off Delay Input to Output		11	22	ns	V _{CC} = 5.0 V	Δ
^t PHL	Turn On Delay Input to Output		7.0	15	ns	RL = 400Ω	

NOTES:

(1) For conditions shown as MIN, or MAX,, use the appropriate value specified under recommended operating conditions for the applicable device type.

. . . .

(2) Typical limits are at $V_{CC} = 5.0 V$, $25^{\circ}C$.

.....

(3) Not more than one output should be shorted at a time,

TITLE: D-Type Flip Flop

FURPOSE: To look at a basic Flip Flop and observe edge triggered function.

PROCEDURE:

1) Build the D-Type Flip Flop out of two SN7410 gates.



- $\sqrt{}$
- 2) Supply the inputs Set, Reset, and Data from data switches. The clock input should come from one of the push buttons. Display the outputs on the lamp monitors.
- 3) Verify operation.

D	S	R	Q(t)	Q(t+1)	
x	0	1	0	0	·
Х	1	0 -	1	1	
X	0	0	1	1	This is a undefined state
0	1	1	1	0	
0	1	1	ŋ	0	
1	1	1	1	1	
1	1	1	0	1	

Q(t) is before positive clock transition. Q(t+1) is after positive clock transitior. Data is stored on positive clock transition.

4) Mount a SN7474 (Dual D-Type Flip Flop) on the Digi Designer. Connect one sections inputs to switches and the outputs to lamp monitors as before. 5) Verify operation.

6) Build a Pulse Catcher.



7) Observe its operation.

DUAL D TYPE EDGE TRIGGERED FLIP-FLOP

DESCRIPTION – The 9N74/5474, 7474 are edge triggered dual D type flip-flops with direct clear and preset inputs and both Q and \overline{Q} outputs. Information at the input is transferred to the outputs on the positive edge of the clock pulse. They are designed for use in medium to high speed applications.

Clock triggering occurs at a voltage level of the clock pulse and is not directly related to the transition time of the positive going pulse. After the clock input threshold voltage has been passed, the data input (D) is locked out and information present will not be transferred to the output.

The 9N74/5474, 7474 have the same clocking characteristics as the 9N70/5470, 7470 gated (edge triggered) flip-flop circuits. They can result in a significant saving in system power dissipation and package count in applications where input gating is not required.

LOGIC AND CONNECTION DIAGRAM

DIP (TOP VIEW).





SCHEMATIC DIAGRAM (EACH FLIP-FLOP)



Positive logic:

LOW input to preset sets Q to HIGH level LOW input to clear sets Q to LOW level Preset and clear are independent of clock

Component values shown are typical.

TRUTH TABLE (Each Flip-Flop)

^t n	tn+1				
INPUT	Ουτρυτ	Ουτρυτ			
D	Q	ā			
L	۰ L	н			
н	н	L			

NOTES:

 $t_n = bit$ time before clock pulse, $t_{n+1} = bit$ time after clock pulse.

LOGIC DIAGRAM (EACH FLIP-FLOP)



TITLE: J-K Master-Slave Flip Flop

PURPOSE: To learn how Master-Slave Flip Flops work internally.

OPERATION:

..

This Flip Flop samples the input data as long as the Clock input is high but it only changes its output state as the Clock changes from high to lo



Single Clock pulse

Time 1 Isolate slave from master Time 2 Allow J and K inputs into Naster Time 3 Isolate J and K inputs from Master Time 4 Transfer data from master to slave

PROCEDURE:

1) Build the following J-K Flip Flop circuit.



2) Verify the functional truth table.

J	ĸ	Q(t)	q(t+1)
0	0	9	Q
Ö	1	ū	0
1	0	Ğ.	1
1	1	C	NOT C

- 3) Mount a SN7476 (Dual J-K Flip Flop) on the Digi Designer. Note that the Power and Ground inputs are not standard.
- 4) Apply high levels to Preset and Clear inputs.
- 5) Verify J-K Flip Flop operation.

.....

•

6) Determine operation of Preset and Clear inputs. Are these inputs asynchronous with respect to the clock?

.

DUAL JK MASTER/SLAVE FLIP-FLOP WITH SEPARATE PRESETS, CLEARS AND CLOCKS

DESCRIPTION — The HSTTL/SSI 9H76/54H76, 74H76 is a High Speed Dual JK Master/Slave flip-flop with separate presets, separate cle and separate clocks. Inputs to the master section are controlled by the clock pulse. The clock pulse also regulates the circuitry which connethe master and slave sections. The sequence of operation is as follows: 1) Isolate slave from master. 2) Enter information from J and K inputto master. 3) Disable J and K inputs. 4) Transfer information from master to slave. Logic state of J and K inputs must not be allowed to char when the clock pulse is a HIGH state.



Component values shown are typical.

TRUTH TABLE

t	t _{n+1}	
J	к	Q
L	L	Qn
L	н	L
н	L	н
н	н	ā,

NOTES:

 $t_n = Bit time before clock pulse.$ $<math>t_{n+1} = Bit time after clock pulse.$



CLOCK WAVEFORM





TITLE: Asynchronous Binary Counter

PURPOSE: To look at a basic finary asynchronous Counter. PROCEDURE:

1) Build the following digital system using SN7474 Flip Flops.



Clock comes from a normally low push button Clear comes from normally high push button Display Flip Flop outputs on the lights

- 2) Press clear button. All lights should go off.
- 3) As the Clock button is pushed the counter will increment.
- 4) Verify count sequence.
- 5) Observe the clear function at various times in the sequence.
- 6) Modify this system to be a Binary UP Counter.
- 7) Mount a SN7493 in the Digi Designer socket. Note power and ground are not standard on this chip.

RO (Pins 2 and 3) are the clear inputs (Clear=1, Count=0) Clock to Input A Connect Output A to Input B

8) Verify that this chip has the same function as 4 D-Type Flip Flops connected as a 4 bit Asynchronous Binary Counter. Note any difference.

TTL/MSI 9393/5493, 7493 4-BIT BINARY COUNTER

DESCRIPTION ~ The TTL/MSI 9393/5493, 7493 is a 4-Bit Binary Counter consisting of four master/ slave flip-flops which are internally interconnected to provide a divide by two counter and a divide-by-eight counter. A gated direct reset line is provided which inhibits the count inputs and simultaneously returns the four flip-flop outputs to a LOW level. As the output from flip-flop A is not internally connected to the succeeding flip-flops the counter may be operated in two independent modes:

- 1. When used as a 4-bit ripple-through counter, output Q_A must be externally connected to input \overline{CP}_{B} . The input count pulses are applied to input \overline{CP}_{A} . Simultaneously divisions of 2, 4, 8 and 16 are performed at the Q_A , Q_B , Q_C , and Q_D outputs as shown in the truth table.
- 2. When used as a 3-bit ripple-through counter, the input count pulses are applied to input \overline{CP}_{B} . Simultaneous frequency divisions of 2, 4, and B are available at the Q_B , Q_C , and Q_D outputs. Independent use of flip-flop A is available if the reset function coincides with reset of the 3-bit ripple-through counter.

These circuits are completely compatible with TTL and DTL logic families.

PIN NAMES

 $\mathbf{Q}_{\mathbf{A}}, \mathbf{Q}_{\mathbf{B}}, \mathbf{Q}_{\mathbf{C}}, \mathbf{Q}_{\mathbf{D}}$

L

.

....

н

н

1

 $\frac{R_0}{CP_A}$

<u>C₽</u>B

COUN

10 ...

12

	LOADING
Reset-Zero Input	1 U.L.
Clock (Active LOW going edge) Input	2 U.L.
Clock (Active LOW going edge) Input	2 U.L.
Dutputs	10 U.L.

1 Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW

NOTES:

- 1. Output Q_A connected to input \overline{CP}_B . 2. To reset all outputs to LOW level both $R_0(1)$ and R0(2) inputs must be at HIGH level state.
- 3 Either (or both) reset inputs Ro(1) and Ro(2) must be at a LOW level to count,







NC - No Internal Connection Positive logic See Truth Table

TITLE: Shift Register

PUPPOSE: To examine shifting operations using Flip Flops.

PROCEDURE:

1) Build the following shift register circuit using two SN7474 Flip Flc



- 2) Display the four outputs on lamp monitors. Use a normally low push button for the clock input and a normally high push button for the clear input. Use a data switch for the Data Input.
- 3) Observe various shifting operations.
- 4) Mount a SN7494 on the Digi Designer. Note power and ground are non-standard.
- 5) Compare the operation of this chip to the shift register previously built.
- 6) Rout the data from the last Flip Flop bacck into the Data Input of the shift register.
- 7) Observe the operation of this recirculating memory system.
- 8) Devise a method to load data into this data stream.
- 9) Implement your method.

TTL/MSI 9391/5491, 7491 8-BIT SHIFT REGISTER

DESCRIPTION - The TTL/MSI 9391/5491, 7491 is a serial-on, serial-out, 8-Bit Shift Register utilizing TTL technology. It is composed of eight RS master/slave flip-flops, input gating, and a clock driver. The register is capable of storing and transferring data at clock rates up to 18 MHz while maintaining a typical noise-immunity level of 1.0V. Power dissipation is typically 175 mW; a full fan out of 10 is available from the outputs.

Single-rail data and input control are gated through inputs A and B and an internal inverter to form the complementary inputs to the first bit of the shift register. Drive for the internal common clock line is provided by an inverting clock driver. Each of the inputs (A, B, and CP) appear as only one TTL input load.

The clock pulse inverter/driver causes these circuits to shift information to the output on the positive edge of an input clock pulse, thus enabling the shift-register to be fully compatible with other edge-triggered synchronous functions.

PIN NAMES		LOADING
А, В	Data Inputs	1 U.L.
Q, ₫	Data Output	10 U.L.
CP	Clock Input	1 U.L.

1 Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW

н

TRUTH TABLE								
. t	t _{n+8}							
Α	B	٥						
L	L	L						
L	н	L						

н

н

NOTES: t_n = Bit time before clock pulse. t_{n+8} = Bit time after 8 clock pulses.





TITLE: Synchronous Binary Counter

PURPOSE: To examine the differences in synchronous counters.

DISCUSSION:

In the Asynchronous Binary Counter each counter (Flip Flop) gets its cl signal from the previous stage. This means that the counters output do not change simultaneously. The synchronous counter over comes this problem by determining the next state between clock pulses. The clock only moves the new state to the output of the Flip Flops.

PROCEDURE:

1) Build the following synchronous counter using SN7476 (J-K Flip Flops and a SN7408 (2 input AND).



Clock Input

- 2) Step through the counting sequence to verify operation.
- 3) Mount a SN74163 Synchronous Counter and a S7404 Inverter on the Digi Designer.

Enable T enable counting and carry out Enable P enable counting Load parallel load (on low level) Clear clear all flip flops Ripple Carry Output display this on extra LED

Compare operation of this counter to the first one. Note how the clear is different and the operation of the enables and load lines.

4) Duplicate the count and load sequence in the data sheet.

5) Modify the counter to have the following sequence:

ا میں میں میں میں ا

6) Check operation of counter.

7) Modify the counter to act as a shift register.

8) Check this operation too.



DESCRIPTION

This synchronous presettable binary counter features an internal carry look-ahead for applications in high-speed counting designs. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincident with each other when so instructed by the count-enable inputs and internal gating. This mode of operation eliminates the output counting spikes which are normally associated with asynchronous (rlpple clock) counters. A buffered clock input triggers the four flip-flops on the rising (positive-going) edge of the clock input waveforms.

This counter is fully programmable; that is, the outputs may be preset to either level. As presetting is synchronous, setting up a low level at the load input disables the counter and causes the outputs to agree with the setup data after the next clock pulse regardless of the levels of the enable inputs. The clear function for the 54/74LS 163 is synchronous and a low level at the clear input sets all four of the flip-flop outputs tow after the next clock pulse, regardless of the levels of the enable inputs. This synchronous clear allows the count length to be modified easily as decoding the maximum count desired can be accomplished with one external NAND gate. The gate output is connected to the clear input to synchronously clear the counter to 0000 (LLLL).

SYNCHRONOUS 4-BIT BINARY COUNTER

The carry look-ahead circuitry provides for cascading counters for n-bit synchronous applications without additional gating. Instrumental in accomplishing this function are two countmable inputs and a ripple carry output. Both count-enable inputs (P and T) must be high to count, and input T is fed forward to enable the ripple carry output. The ripple carry output thus enabled will produce a high-level output pulse with a duration approximately equal to the high-level portion of the Q_A output. This high-level overflow ripple carry pulse can be used to enable successive cascaded stages. Transitions at the enable P or T inputs are allowed regardless of the level of the clock input.

The 54/74LS163 features a fully independent clock circuit. Changes made to control inputs (enable P or T, load or clear) that will modify the operating mode have no effect until clocking occurs. The function of the counter (whether enabled, disabled, loading or counting) will be dictated solely by the conditions meeting the stable setup and hold times.

PARAMETER MEASUREMENT INFORMATION TYPICAL CLEAR, PRESET, COUNT, AND INHIBIT SEQUENCES

Mustrated below is the following sequence:

- 1. Clear outputs to zero
- 2. Preset to binary twelve
- 3. Count to thirteen, fourteen, fifteen, zero, one, and two
- 4. Inhibit





VOLTAGE WAVEFORMS

NOTES

- ▲ The input pulses are supplied by generators having the following characteristics. PBR ≤ 1. We, duty cycle \leq 50%, Z_{OUI} \approx 50 1, 163, $I_{f} \leq$ 10 ns; and for 'LS160 thru. LS163, $I_{f} \leq$ 15 ns; $I_{f} \leq$ 6 ns; and for 'LS160 thru. LS163, $I_{f} \leq$ 15 ns; $I_{f} \leq$ 6 ns; and for 'LS160 thru. LS163, $I_{f} \leq$ 15 ns, $I_{f} \leq$ 6 ns; and for 'LS160 thru.
- C Vret = 13 V.

FIGURE 2-SWITCHING TIMES



2

54/74163

VOLTAGE WAVEFORMS

NOTES:

A. The input pulses are supplied by a generator having the following characteristics. PRR ≤ 1 MHz, duty cycle $\leq 50\%$, $Z_{out} \approx 50\%$, $t_f \leq 15$ ns, $t_f \leq 6$ ns. Vary PRR to measure $t_{max} = 8$. Outputs O_D and carry are tested at $t_0 + 16$ where t_0 is the bit time when all outputs are

low C. V_{ref} = 1.3 V.

Load circuit is shown at the front of the book

FIGURE 1-SWITCHING TIMES

TYPICAL APPLICATION DATA

N-BIT SYNCHRONOUS COUNTERS

This application demonstrates how the look-ahead carry circuit can be used to implement a high-speed n-bit counter. The 54/74LS163, will count in binary. Virtually any count mode (modulo-N, N1-toN2, N1-to-maximum) can be used with this fast look-ahead circuit.



TITLE: UP/DOWN Binary Counter

PURPOSE: To observe a UP/DOW! Counter and gain familiarity with sequential logic component data sheets.

FROCEDURE:

· · · · · · ·

.

- Mount the SN74192 on the Digi Designer. Connect switches and lights as needed.
- 2) Duplicate Load and Count sequences on the data sheet.
- 3) Add a second SN74192 to make a 8 bit UP/DOWN Counter.
- 4) Verify all counting operations with this extended counter.

SYNCHRONOUS DECADE UP/DOWN COUNTER

PIN CONFIGURATION

SPEED/PACKAGE AVAILABILITY

в

54	F,W	74	₿
54LS	F.W	74LS	В

DESCRIPTION

This monolithic circuit is a synchronous reversible (up/down) counter having a complexity of 55 equivalent gates. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincidently with each other when so instructed by the steering logic. This mode of operation eliminates the output counting spikes which are normally associated with asynchronous (ripple-clock) counters.

The outputs of the four master-slave flip-flops are triggered by a low-to-high-level transition of either count (clock) input. The direction of counting is determined by which count inputs is pulsed while the other count input is high.

All four counters are fully programmable; that is, each output may be preset to either level by entering the desired data at the data inputs while the load input is low. The output will change to agree with the data inputs independently of the count pulses. This feature allows the counters to be used as modulo-N dividers by simply modifying the count length with the preset inputs.

A clear input has been provided which forces all outputs to the low level when a high level is applied. The clear function is independent of the count and load inputs. The clear, count, and load inputs are buffered to lower the drive requirements. This reduces the number of clock drivers, etc., required for long words.

These counters were designed to be cascaded without the need for external circuitry. Both borrow and carry outputs are available to cascade both the up- and down-counting functions. The borrow output produces a pulse equal in width to the count-down input when the counter underflows. Similarly, the carry output produces a pulse equal in width to the count-down input when an overflow condition exists. The counters can then be easily cascaded by feeding the borrow and carry outputs to the count-down and count-up inputs respectively of the succeeding counter.



54/74192

SWITCHING CHARACTERISTICS VCC = 5V, TA = 25°C

					54/74			54/74LS		
TEST C	ONDITIONS			CL = 15pF RL =400Ω			CL 15pF RL ≤2k			
PARAM		FROM INPUT	TO OUTPUT	MIN	түр	MAX	MIN	TYP	MAX	UNIT
fCount		Count Up	Carry	25	32		25	32		MHz
tw	Width of pulse			ľ			20			ns
tSetup	Input setup time				ł		201			ns
tHold	Input hold time	} 1					to			กร
Propaga	tion delay time			'			ł			
^t PLH	Low-to-high	Countup	Carry		17	26	1	17	26	ns
		Count Down	Borrow		16	24		16	24	{ }
		Either Count	a		25	38		25	38	
		Load	Q		27	40		27	40	
^t PHL	High-to-low	Count Up	Carry		16	24		16	24	ns
	-	Count Down	Borrow]	16	24		16	24	
		Either Count	Q		31	47		31	47	1
		Load	Q	ļ	29	40		29	40	
		Clear	Q		22	35		22	35	

SYNCHRONOUS DECADE UP/DOWN COUNTER

54/74192

BLOCK DIAGRAM



PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT



VOLTAGE WAVEFORMS

NOTES:

- A The pulse generators have the following characteristics: Z_{out} ≈ 50 Ω, and for the data pulse generator PRR ≤ 500 kHz, duty cycle ~ 50%; for the load pulse generator PRR is two times data PRR, duty cycle ~ 50%.
- G C includes probe and ing capacitance C Dodes are IN3064. D I and I \leq 7 ns E V_{ref} is 1.3 volts

FIGURE 1-CLEAR, SETUP, AND LOAD TIMES

TYPICAL CLEAR, LOAD, AND COUNT SEQUENCES

una. Alimter

a sala a

Illustrated below is the following sequence:

- 1. Clear outputs to zero.
- 2. Load (preset) to BCD seven.
- 3. Count up to eight, nine, carry, zero, one, and two.
- 4. Count down to one, zero, borrow, nine, eight, and seven.



NOTES

Clear overrides load, data, and count inputs. When counting up, count-down input must be high, when counting down, count up input A B must be high

TITLE: Decimal Scaling

PURPOSE: To introduce BCD Counters.

PROCEDURE:

- 1) Mount a SN7490 Decade Counter on the Digi Designer. Power and Ground are not on standard pins.
- 2) Apply high inputs to pin 3 (RO) and pin 7 (R9). Apply levels derived from the data switches to the other RO and R9 input pins. Supply inputs A and B from the push buttons. Display all four outputs on the lamp monitors.

3) Determine count sequence for input A.

4) Determine count sequence for input B.

5) Connect the two sections to generate one pulse for every ten input pulses. The output pulse should be high for five input pulses and low for five more input pulses.

6) Connect the two sections to count in BCD.

- 0 0000 1 0001 2 0010 3 0011 4 0100 5 0101 6 0110
- 7 0111
- 8 1000
- 9 1001
- 0 0000
- 7) Observe the two clear function (RO and R9).

8) Cascade two of these units to divide the input rate by 100.

9) Verify operation. It may help in this step to use an oscilloscope. If you do not know how to use the scope ask your Lab Instructor.

TTL/MSI 9390/5490, 7490 DECADE COUNTER

DESCRIPTION – The TTL/MSI 9390/5490, 7490 is a Decade Counter which consists of four dual rank, master slave flip-flops internally interconnected to provide a divide by-two counter and a divide-by-five counter. Count inputs are inhibited, and all outputs are returned to logical zero or a binary coded decimal (BCD) count of 9 through gated direct reset lines. The output from flip-flop A is not internally connected to the succeeding stages, therefore the count may be separated into these independent count modes:

- A. If used as a binary coded decimal decade counter, the CP_{BD} input must be externally connected to the Q_A output. The CP_A input receives the incoming count, and a count sequence is obtained in accordance with the BCD count for nine's complement decimal application.
- **B.** If a symmetrical divide-by-ten count is desired for frequency synthesizers or other applications requiring division of a binary count by a power of ten, the Q_D output must be externally connected to the \overline{CP}_A input. The input count is then applied at the \overline{CP}_{BD} input and a divide-by-ten square wave is obtained at output Q_A .
- C. For operation as a divide-by-two counter and a divide-by-five counter, no external interconnections are required. Flip-flop A is used as a binary element for the divide-by-two function. The CPBD input is used to obtain binary divide-by-five operation at the QB, QC, and QD outputs. In this mode, the two counters operate independently; however, all four flip-flops are reset simultaneously.

PIN NAMES		LOADING
R ₀	Reset-Zero Inputs	1 U.L.
Rg	Reset-Nine Inputs	1 U.L.
ĊΡ _Δ	Clock Input	2 U.L.
CP _{BD}	Clock Input	4 U.L.
$\mathbf{Q}_{\mathbf{A}}, \mathbf{Q}_{\mathbf{B}}, \mathbf{Q}_{\mathbf{C}}, \mathbf{Q}_{\mathbf{D}}$	Outputs	10 U.L.

1 Unit Load (U.L.) = 40µA HIGH/1.6mA LOW.





Positive logic: See Truth Tal NC - NO internal connection

TITLE: BCD-to-Decimal Decoder

PURPOSE: A quick look at a pasic decoder.

PROCEDURE:

- 1) Build a BCD counter using a SN7490.
- 2) Rout its outputs to the inputs of a SN7442 BCD-to-Decimal Decoder.
- 3) Note how the outputs change as the counter changes. It may be helpful to build six extra logic monitors from a SN7404 and some LED's so that all outputs can be seen simultaneously.

TTL/MSI 9352/5442, 7442 9353/5443, 7443 • 9354/5444, 7444 FOUR LINE TO TEN LINE DECODER (1-OF-10)

DESCRIPTION — These monolithic Decimal Decoders consist of eight inverters and ten 4-input NAND gates. The inverters are connected in pairs to make BCD input data available for decoding by the NAND gates. Full decoding of valid input logic insures that all outputs remain off for all invalid input conditions.

The TTL/MSI 9352/5442, 7442 BCD-to-decimal; TTL/MSI 9353/5443, 7443 excess 3-to-decimal; and TTL/MSI 9354/5444, 7444 excess 3 gray-to-decimal decoders feature familiar transistor-transistor logic (TTL) circuits with inputs and outputs compatible for use with other TTL and DTL circuits. The dc noise margins are typically 1.0V and power dissipation is typically 140 mW. Full fan out of 10 is available at all outputs.

LOĂDING 1 U.L.

1 U.L.

1 U.L.

10 U.L.

A, B, C, D A, B, C, D A, B, C, D 3 to 3

BCD Inputs (9352) Excess 3 Inputs (9353) Excess 3 Gray Inputs (9354) Decimal Output



1 Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW





TITLE: BCD to 7-Segment Decoder/Driver

PURPOSE: To look at a simple method of displaying numerical data.

DISCUSSION:

Many times a digital system needs to display some form of numeric information. One relatively inexpensive method is to use a 7-Segment Display. This is a group of seven line segments each illuminated by a seperate LED. With the aid of a Decoder/Driver, which accepts standard BCD inputs, the seven segments can display all digits C through 9. Most Decoder/Drivers have two other features, one which will blank leading zeros and another which will test all segments.

PROCEDURE:

- 1) Build a BCD counter using a SN7490.
- 2) Add on a SN7448 7-Segment Decoder/Driver and a DL704 display unit with the required 390 Ohm resistors (Orange White Brown).



- 3) Verify the count sequence and proper numeric display.
- 4) Add a SN7475 Quad D-Type Latch between the counter and the Decoder/Driver. Supply the Clock to all sections from a normally low push button.
- 5) Pushing the button will transfer the counter data into the latch for displaying.
- 6) Verify that the informatin in the counter can change with out affecting the diplayed data until the button is pushed.
- 7) Observe the oreration of the Planking option and Lamp Test option of the Decoder/Driver.

TTL/MSI 9357A/5446, 7446 9357B/5447, 7447 BCD TO 7-SEGMENT DECODER/DRIVER

DESCRIPTION – The 9357A/5446, 7446 and 9357B/5447, 7447 are TTL, BCD to 7-segment Decoder/Drivers consisting of NAND gates, input buffers and seven AND-OR-INVERT gates. They offer active LOW, high sink current outputs for driving indicators directly. Seven NAND gates and one driver are connected in pairs to make BCD data and its complement available to the seven decoding AND-OR-INVERT gates. The remaining NAND gate and three input buffers provide lamp test, blanking input/ripple-blanking output and ripple-blanking input.

The circuits accept 4-bit binary-coded-decimal (BCD) and, depending on the state of the auxiliary inputs, decodes this data to drive a 7-segment display indicator. The relative positive-logic output levels, as well as conditions required at the auxiliary inputs, are shown in the truth tables. Output configurations of the 9357A/5446, 7446; 9357B/5447, 7447 are designed to withstand the relatively high voltages required for 7-segment indicators. The 9357A/5446, 7446 outputs will withstand 30 V and the 9357B/5447, 7447 outputs will withstand 15 V with a maximum reverse current of 250 μ A. Indicator segments requiring up to 20 mA of current may be driven directly from the 9357A/5446, 7446 or 9357B/5447, 7447 high performance output transistors. Display patterns for BCD input counts above nine are unique symbols to authenticate input conditions.

The 9357A/5446, 7446; 9357B/5447, 7447 incorporate automatic leading and/or trailing-edge zero-blanking control (RBI and RBO). Lamp test (\overline{LT}) of these types may be performed at any time when the BI/BRO node is a HIGH level. Both contain an overriding blanking input (BI) which can be used to control the lamp intensity or to inhibit the outputs.

N NAMES		LOADING
, 8, C, D	BCD Inputs	1 U.L.
BĪ	Ripple Blanking Input	1 U.L.
T	Lamp Test Input	1 U.L.
I/RBO	Blanking Input or	2.6 U.L.
	Ripple Blanking Output	5 U.L.
to g	Outputs	12.5 U.L.

PI

Α

R

E

B

ī,







(D) When the blanking input/ripple-blanking output (BI/ABO) is open or held at a HIGH level, and a LOW level is applied to lamp test input, all segment outputs go to a LOW level.

RECOMMENDED OPERATING CONDITIONS

		935 935	7AXM/54 7BXM/544	4GXM 17XM	935 935	7AXC/744 7BXC/744	6XC 7XC	UNITS
		MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Supply Voltage V _{CC} (See Note 3)		4.5	5.0	5.5	4.75	5. 0	5.25	Volts
Operating Free-Air Temperature Range		-55	25	125	0	25	70	Ċ
Normalized Fan Out From	Outputs a through g to Series 54/74 Loads			24			24	
Normalized Fan Out From	BI/RBO Node to Series 54/74 Loads			5.0			50	
Output Sink Current, IOL:	Outputs a through g			40			40	rηĄ
	BI/RBO Node			8.0			8.0	mA

X = package type; F for Flatpak, D for Ceramic Dip, P for Plastic Dip. See Packaging Information Section for packages available on this product.

	935	7A/5446,	7446	939	57B/5447,	7447	
PARAMETER Continuous Voltage at Outputs à through ĝ	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	UNITS
Continuous Voltage at Outputs à through ĝ			30		•	15	Volts

;

,



FND-500



110

TITLE: One Shot Multivibrators

PURPOSE: To examine asynchronous timing devices.

DISCUSSION:

A One Shot Multivibrator is an edge triggered device. It generates a pules starting on the triggering edge and lasting for preset period of time. The period of time is set by a resistor and a capacitor connected to the One Shot.

Inp	uts	5	OUtp	outs			
A	В		Q	Q'			
1	Х		0	1			
X	9		n	1			
0					low to high	level	trigger
!	1				high to low	level	trigger
Pul	se	Width	= 0.	.31*R*(C*(1+1/R)		
Pul	s e	width	in r	۱s			

Pulse width in ns R in K Ohm C in pF

PROCEDURE:

1) Mount a SN74123 One shot on the Digi Designer and build the following circuit:



Use 4.7% Ohm resistors (Yellow Violet Red) for each of the resistors and 1800 pF for the capacitors. Run the clock at 10KHz.

2) Using a Dual Trace Oscilloscope display one clock cycle on channel

- 3) Using the second channel display the output of the first One Shot.
- 4) Using channel 2 displiay measure the pulse width. Calculate the pulse width for the component values used and compare to mesured values.

1 / 1

5) Using channel 2 display the output of the second One Shot. Notice how this pulse starts after the first pules finishes.

.

.

.

6) Change the resistor to double the delay of the first One Shot.

7) Verify the time change.

RETRIGGERABLE MONOSTABLE MULTIVIBRATOR WITH CLEAR

SPEED/PACKAGE AVAILABILITY

54 F.W 74 B.F

DESCRIPTION

These monolithic TTL retriggerable monostable multivibrators feature dc triggering from gated low-level-active (A) and high-levelactive (B) inputs, and also provide overriding direct clear inputs. Complementary outputs are provided. A full fan-out to 10 normal-Ized Series 54/74 loads is available from each of the outputs at the low logic level, and in the high-level state, a fan-out of 20 is available. The retrigger capability simplifies the generation of output pulses of extremely long duration. By triggering the input before the output pulse is terminated, the output pulse may be extended. The overriding clear capability permits any output pulse to be terminated at a predetermined time independently of the timing components R and C.

Figure A illustrates triggering the one-shot with the high-level-active (B) inputs.

SWITCHING CHARACTERISTICS VCC = 5V, TA = 25°C

				54/74		ł	
TEST CONDITIONS							
PARAMETER	FROM INPUT	TO OUTPUT	MIN	түр	MAX	UNIT	
^t w(min)				1			
t _w Width of pulse			C _{ext} = 1000pF R _{ext} = 10KΩ 54/74123				
			3.08	3.42 54/7412	3.76 3A	μS	
			2.76	3.03	3.37		
t _{w(in).}			40			ns	
Rext External timing			(54)			1	
resistance			5		25	kΩ	
	1		5		50		
			1				

TRUTH TABLE

INP	JTS	OUTPUTS					
	8	Q	Q				
н	x	L	н				
x	L	ι	н				
L	t	<u></u>					
	н	<u>~</u>	~~				

NOTES.

A. H = high level (steady-state) 1 ~ low level (steady state), 1 = transition from tow to high level 1 = transition from high to low level, - one high-level pulse, - one low-level pulse, X - irrelevant (any input, including (ransitions)

B NC + no internal connection

Cext and Rext (positive)

TYPICAL CHARACTERISTICS





† These values of resistance exceed the maximums recommended for use over the full temperature range of the 554122 and 554123

NOTE:

When using electrolytic capacitor insure that minimum rating is 20 volts so that 5% reverse voltage rating is 10 volt or greater





54/74123

54/741234

TITLE: Data Selector/Multiplexer

PURPOSE: To look at Multiplexing operations and function generation using data selectors.

PROCEDURE:

1) Mount a SN74151 Data Selector in the Digi Designer socket.

Supplt Data Select inputs from the switches Supply Strobe from a push button Supply 8 Data inputs from an external switch register Display the output on a light

- Observe Data Selector/Multiplexer operation by addressing various input lines with select switches.
- 3) Observe the operation of the strobe line.
- 4) Use the Data Selector to generate the following function:

Input	Output
000	1
001	1
010	1
011	0
100	1
101	C
110	0
111	1

Note that any function of three input variables can be implemented with a 8 to 1 Data Selector.

5) Use the Data Selector and as few Inverters as possible to implement the following function of four variables:

.

Output
1
1
1
0
С
1
1
· 0
1
0
0
0
1
1
0
1

.

.

6) Can any function of four variables be implemented with a 8 to 1 Data Selector?

165

TTL/MSI 93150/54150, 74150 93151/54151, 74151 • 93152/54152, 74152 16-INPUT AND 8-INPUT MULTIPLEXER

DESCRIPTION – The 93150/54150, 74150 is a 16-Input Multiplexer which features active LOW strobe and internal select decoding. A HIGH at the strobe input forces the output HIGH regardless of input conditions.

The 93151/54151, 74151 is an 8-Input Multiplexer with active LOW strobe, internal select decoding and comptementary outputs.

The 93152/54152, 74152 is an 8-Input Multiplexer with internal select decoding and a single inverted output.

In each of the multiplexers data is routed from a particular input to the output according to the binary code applied to the select inputs.

Typical power dissipations are: 93150/54150, 74150 - 200 mW; 93151/54151, 74151 - 145 mW; 93152/54152, 74152 - 130 mW.

PIN NAMES E0 to E15 D0 to D7 S A, B, C, D W Y

NOTE: 1 U.L. = 40 µA HIGH/1.6 mA LOW.

Data Inputs Data inputs Strobe (Enable) Input Data Select Inputs Data Output Data Output LOADING 1 U.L. 1 U.L. 1 U.L. 1 U.L. 10 U.L. 10 U.L.





^{*}Pro-assignments for these circuits are the same for all packages.



TRUTH TABLES

93150/54150, 74150

											NPUTS										OUTPUT
D	c	8	A	STROBE	Eo	E 1	E2	E 3	E4	E 5	E6	E,	Eg	Eg	E 10	611	E 17	£13	E14	F 15	W
х	×	×	×	н	x	x	×	x	X	×		×	X	x	×	- x -	×	x	×	X	H
L	L	L	L	ļι	L	x	x	x	x	×	×	x	x	×	×	x	×	×	x	x	н
L	ι	Ł	ι.	L	н	×	×	х	x	×	×	x	x	×	×	×	×	×	x	x	ι
L	ι	L	н	L	х	L	x	×	x	×	×	×	×	×	X	х	×	x	×	×	H
L	L	I.	н	L	×	н	x	x	х	×	×	×	X	×	x	×	×	×	x	×	L
Ł	L	н	L	L	×	х	- L -	×	×	×	×	×	ĸ	×	×	x	x	×	×	¥	н
L	L	н	L	L	ί×	×	н	×	×	×	х	×	×	×	×	×	×	×	×	×	ι
L	L	н	н	ι	x	x	×	L	×	×	×	×	×	×	x	×	x	×	×	×	н
L	L	н	н	L	×	×	×	н	×	x	×	×	×	×	×	×	×	×	¥	×	L 1
L	н	L.	L	[ι	×	×	×	×	L	x	×	×	×	×	×	×	×	×	×	×	н
L	н	L	L	L	×	×	×	×	н	×	×	×	×	×	×	×	×	×	×	×	t
ι	н	L.	н	ι	×	×	×	×	×	L	×	×	×	×	×	×	X.	х	×	×	н
L	н	L	н	L	×	×	×	×	×	н	×	×	×	×	×	×	×	×	×	×	ι
L	н	н	L	ι	×	×	×	×	x	ĸ	L	×	×	×	×	×	×	×	x	×	н
L	н	н	L	L	×	x	×	×	×	×	н	x	×	×	×	x	×	×	×	×	ι
L	н	н	н	ι	X	x	×	×	×	×	×	L.	×	x	×	x	×	×	×	×	H H
L	н	н	н	ι ι	×	×	×	x	. ×	x	×	н	×	×	. х	×	×	×	×	×	۱.
н	L,	L	L	L	×	x	x	× ′	×	×	×	×	L	x	×	x	×	×	×	x	н
н	L	L	L	ι	×	x	x	x	×	×	×	×	н	×	×	×	×	х	×	×	(+
н	L	L	н	L	x	x	x	×	×	x	×	×	×	L	×	x	×	x	×	×	н
н	L	L	н	ι	x	x	x	x	×	x	x	x	×	н	×	×	×	x	×	×	L.
н	L	н	L) L	x	×	x	x	×	×	×	x	×	×	L	×	×	×	×	x	н
н	L	н	ΞL.	L	х	x	x	×	×	×	×	×	×	×	н	×	×	×	×	x	L
н	L	н	н	L	x	x	×	x	×	x	×	x	×	×	×	L	x	×	x	×	н
н	L.	н	н	ι	x	×	×	x	×	×	×	×	×	×	x	н	×	x	×	×	L
н	н	L	L	L	×	×	x	×	×	×	×	×	×	×	×	x	L	×	×	×	н
н	н	L	L	L	X	х	х	×	×	×	×	×	×	x	×	×	н	×	×	×	L
н	н	L	н	L	X	×	x	×	x	×	×	×	x	x	×	x	×	L	x	×	н
н	н	L	н	1 L	×	x	×	x	x	x	×	×	×	×	x	×	×	н	×	x	L
н	н	н	L	1 L	X	×	×	х	X	x	x	x	X	×	x	×	×	×	L	x	н
н	н	н	L	(L	X	×	×	x	x	x	x	x	x	x	×	x	×	×	н	x	l L
н	н	н	н	Lι	×	×	×	x	x	x	×	×	x	x	×	×	×	×	×	L	н
н	н	н	н	L	X	х	x	×	×	х	×	×	×	×	×	×	×	×	×	н	L

When used to indicate an input condition, X = LOGICAL H or LOGICAL L.

93151/54151, 74151 AND 93152/54152, 74152

				_	0	NPUTS						OUTP	uts
c	B	A	STROBE(1)	Do	Di	D ₂	03	04	D5	06	D7	Y(5)	W
x	х	х	н	x	x	x	×	×	x	×	X	L	H
L	L	L	L	L	×	x	х	×	×	x	x	L	н
	L	L	L	н	×	×	×	×	×	×	х	н	ļι
L	L	н	L	x	L	х	×	х	×	х	· x) L	ļн
	L	н	L	X	н	×	x	×	×	×	×	н	ļι
L	н	L	L	x	х	L	×	х	×	х	×	L	ļн
L	н	L	ι	×	x	н	x	x	×	×	×	н	j L
L	н	н	ι	×	×	x	L	x	x	×	×	ι	⊢
	н	н	ίι.	×	x	×	н	x	×	x	×	н	ι L
4	L	L	ι	X	×	×	x	L	х	×	×	L L	1
ł	L	L	L	x	×	x	×	н	×	×	×	н	1
ł	L	н	ιL	x	×	х	×	х	ι	×	x	ίι.	1
ł	L	н	Ι ι	x	×	x	х	×	н	×	×	н	1 1
4	н	L	ι .	x	х	x	x	×	x	L	x	ι	•
H.	н	ι	ι ι	x	×	×	х	×	x	н	x	н	1
н	н	н	ι ι	x	х	×	×	×	x	×	L	L	1.
H.	н	н	1 1	x	x	х	×	×	x	×	н) н	1

ABSOLUTE MAXIMUM RATINGS (above which the useful life may be impaired)

Storage Temperature Temperature (Ambient) Under Bias VCC Pin Potential to Ground Pin *Input Voltage (dc) *Input Current (dc) Voltage Applied to Outputs (Output HIGH) -0.5 V to +V_{CC} value Output Current (dcl (Output LOW)

*Either Input Voltage limit or Input Current limit is sufficient to protect the inputs.

-65 °C to +150 °C

-55°C to +125°C

-0.5 V to +7.0 V

-0 5 V to +5 5 V

+30 mA

-30 mA to +5.0 mA

TITLE: Bipolar Random Access Memory

PURPOSE: To look at a 16X4 memory element.

PROCEDURE:

. . <u>.</u>...

1) Build the following system :

The SN7493 is the address source for this memory. The 4 LED's display the address of the cell being accessed. The 4 resistors from the memory output to the +5.0v supply are Pull UP resistors. This chip has Open Collector outputs which allow a f of buss operation, but require Pull Up resistors.

....

2) Load some data into the memory.

3) Read it back.

4) Modify the data stored in the memory.


TTL MEMORY 93403 64-BIT FULLY DECODED READ/WRITE MEMORY FORMERLY 4103

SN 7482

DESCRIPTION — The 93403 is a high speed 64-Bit Read/Write Memory organized 16 words by four bits. Four address lines are buffered and decoded "on chip" for word selection. The 93403 is made with TTL circuitry and all inputs are equivalent to one TTL load.

OPERATION – When the 93403 receives a LOW at the Chip Select (\overline{CS}) input, the binary address IA0, A1, A2 and A3} is decoded to select one of sixteen 4-bit words. If the Write Enable (\overline{WE}) is at a HIGH level, the contents of the selected word are non-destructively read out and the sense amplifier outputs ($\overline{O_0}$, $\overline{O_1}$, $\overline{O_2}$ and $\overline{O_3}$) reflect the state of the stored data in the four bits of the selected word. If the Write Enable is LOW, the data present on the Data Input lines (D_0 , D_1 , D_2 and D_3) is written into the four bits of the selected word. Note that there is inversion through the device in a read operation.

- OUTPUT WIRED-OR CAPABILITY
- ON CHIP DECODING
- NON-DESTRUCTIVE READOUT
- CHIP SELECT FOR SYSTEM WORD EXPANSION
- TTL COMPATIBLE

ABSOLUTE MAXIMUM RATINGS (above which the useful life may be impaired)

Storage Temperature	-65°C to + 150°C
Temperature (Case) Under Bias	-55°C to + 125°C
V _{CC} Pin Potential to Ground	-0.5 V to + 8.0 V
Input Pin Voltage	-1.5 V to + 5.5 V
Current Into Output Terminal	100 mA
Output Voltage (external circuit dependent)	-0.5 V to + 8.0 V





EXPERIMENT 26

TITLE: Frequency Counter

PURPOSE: To combine basic system parts to build a functioning digital system.

DESCRIPTION:

A frequency counter is a device which counts the number of events in a given Period of time. Our frequency counter will count up to 999 events in a one second period. The system will operate in the following manner:



The system will have five basic parts:



PROCEDURE :

 Using one of the Crystal time Base Units build a 1 cycle per second time base using SN7490 counters. The clear input of these counters, and the gate control of the Crystal Time Base must all be connected together.

2) Build the following control circuit after becoming familiar with is operation. The D Type Flip Flop is a pulse catcher that forms a very accurate 1 second window for event counting. The two One Shot control the system between counting cycles.



System Timing Diagram:



3) Build a three digit BCD counter (SN7490 counters) with latches (SN7475) between the counters and the 7-Segment Decoder/Drivers (SN7448). All latches have a common clock 'Load Latch' and all the counters have a common clear line.



4) Verify the system operation first by counting the frequency of the Digi Designer Clock. Then compare this measurement to that made with a Laboratory Frequency Counter. 5) Modify the frequency counter to become a period counter. Count the number of 1/1000 seconds between two events.

.

6) Verify the operation of this by calculating the period of the Digi Designer Clock.

•

EXPERIMENT 27

TITLE: Inverting Amplifier

PURPOSE: To observe an Inverting Amplifier built from an Operational Amplifier.

PROCEDURE:

Build the following Inverting amplifier using a uA741
 Operational Amplifier. Use power supply voltages of + and - 10.0
 volts. R1 = R2 = 1 K Ohm. Use a 10 K Ohm Pot and two 10 K Ohm
 resistors to supply the input to the amplifier.



2) Verify the Inverting Amplifier formula by mesuring input and output voltages with the voltmeter over the input range obtained from the Fot.

-(V out) * R1 = (V in) * R2

- 3) Verify that the input (V-) is a virtual ground with the voltmeter.
- 4) Modify the circuit so (V out) = $-2 \times (V \text{ in})$
- 5) Verify operation over input range.

µA741 FREQUENCY-COMPENSATED OPERATIONAL AMPLIFIER FAIRCHILD LINEAR INTEGRATED CIRCUIT

CONNECTION DIAGRAMS

8-LEAD METAL CAN

(TOP VIEW)

GENERAL DESCRIPTION — The μ A741 is a high performance monolithic Operational Amplifier constructed using the Fairchild Planar* epitaxial process. It is intended for a wide range of analog applications. High common mode voltage range and absence of latch-up tendencies make the μ A741 ideal for use as a voltage follower. The high gain and wide range of operating voltage provides superior performance in integrator, summing amplifier, and general feedback applications. Electrical characinities of the μ A741 and F are identical to MIL M-38510/101



EXPERIMENT 28

TITLE: Digital to Analog Converters

PURPOSE: To observe two basic types of DAC.

FROCEDURE:

1.

1) Build the following Weighted Current Source DAC. The input switches are the Digi Designer switches. (V ref) is the Digi Designer +5.0 v power supply. R = 1000 Ohms use multiple 1000 Ohm resistors in series to get other values. The Op Amp is a uA741 Power supply voltages for the Op Amp are + and - 10.0 volts.



Verify operation of DAC by observing its output on a voltmeter.
 Determine step size and step linearity.

4) Build the following R-2R DAC using the same basic parts.



EXPERIMENT 29

TITLE: Analog to Digital Conversion

PURPOSE: To observe the operation of a Servo Type ADC.

PROCEDURE:

 Build the following Servo Type ADC using the R-2R DAC. The digital input for the DAC now comes from the outputs of a four bit binary counter. A data latch is provided so that the counting part of the cycle is not passed to the output.



2) Verify the operation of this ADC.

3) Observe the linearity and step size of this system.

5. NOTES FROM THE REAL WORLD:

The Results of a Survey of the Data Processing Community

Figures 1, 2, 3, 4 and 5 show the cover letter and the survey instrument. Figure 6 shows the turnaround document requested for follow-up. The instruments were enclosed with a self-addressed, un-franked envelope and mailed to almost 5,000 computer professionals in the U.S., South America, Europe and Japan. Less than 10% were returned because of bad addresses. Of the 465 responses, 210 gave permission for personal contact, 214 indicated a desire to receive a newsletter and 63 indicated a willingness to participate in a workshop. In all, there were 296 turnaround documents returned. Of those responses which recorded the sex of the respondent there were 438 males and 16 females. The age group of the respondents is indicated in the following table:

Age	<26	26- 30	31-35	36- 40	41-45	46-60	>60
Total	8	46	103	88	82	121	16

Figure 7 shows the work experience distribution by class of position and by employer type and shows the average number of years spent in each position class.

Figure 8 shows the distribution of programming language proficiency for the respondents.

Figure 9 shows the distribution of answers for the questions indicated. The extreme right hand column shows the ratio of the number of responses for which a question was not answered to the total number of responses.

On the whole these data show that the survey was well received by a highly qualified body representing computing technology. The response rate was almost 10%. (A response rate of better than 5% is considered outstanding by most surveying organizations.)

With regard to the meaning of the tabulated responses, it is dangerous to take an individual question and make inference. For instance, question number 35 is a key question; its tabulated data show that the responses to the question indicated:

> 35a. Investment in analog/hybrid to be increased = 96 35b. Investment in analog/hybrid to be decreased = 48 35c Investment in analog/hybrid to remain the same = 215

Unless these responses are analyzed to adjust for the fact that those people who have no analog/hybrid facilities will remain the same, their implication can be misleading. Analysis showed that only 159 responses were recorded for organizations which have analog/hybrid facilities (see question No. 27). Of these, only 69 answered that their investment would remain the same; 29 indicated their investment would be increased; 22 indicated a decrease; and 39 chose not to respond. Of the 267 who answered question 27 with a no (indicating no current investment in analog/hybrid) only 55 indicated an increase in analog/hybrid; of these 55 responses only 20 answered questions 10 and 13 affirmatively.

A detailed correlation analysis has not been performed and does not seem warranted. All of the responses have been screened and the general patterns seem to hold. The above analysis of question 35 responses confirmed what was not obvious from the tabulated data but which was clearly the impression gained by screening each response.

Seorgia Institute of Echnology school of information and computer science / (404) 894-3152 / ATLANTA, GEORGIA 30332

Dear Colleague:

I would like to ask you to participate in a survey being conducted to determine the computing needs of science and engineering education. The study is being made by the School of Information and Computer Science, Georgia Institute of Technology.

You were selected because you have been identified as an individual who is knowledgeable about the computer world and who is serving his or her organization in a senior executive, technical, academic or research capacity.

I hope that you will complete the form personally; however, if you feel that someone else in your organization would be a more appropriate respondent for the survey, please have that person complete the form in your place.

The questionnaire is simple, and should take no more than a few minutes of your time. I hope that you will be kind enough to give me those minutes, and return the form as soon as possible to:

A.P. Jensen Principal Research Engineer School of Information and Computer Science Georgia Institute of Technology Atlanta, Georgia 30332

Should your answers to any of questions 41, 42, 54 or 55 be YES, the enclosed self-addressed card should be completed and returned to me so that I may include your name on a new mailing list or contact you personally.

Thank you sincerely for your cooperation.

Yours truly,

Jensen

APJ/me

FIGURE 1

Please mail completed questionnaire to:

.

A.P. Jensen School of Information & Computer Science Georgia Institute of Technology Atlanta, GA 30332

Ι.	BACKGROUND INFORMATION
1.	Your job title:
2.	Primary business of your organization or company:
3.	Piease list some of your major job responsibilities and duties:
	· · · · · · · · · · · · · · · · · · ·
4.	Please list type and number of personnel you supervise:
	Type of Personnel Number
5.	Does your organization or company: a Maintain its own computation facilities b Contract for computation services c have some additional or different arrangement. Please explain:
6.	Does your job position primarily entail:
	a Providing computer support facilities to other persons or units in your organization.
	b Utilizing computer facilities maintained by another unit or contract facility.
	c Neither of the above. Flease explain:
7.	What is the trade name of the main computer(s) used by your organization:
8.	Based on relative expenditures, estimate the extent to which your organization employs computers for each of the following purposes:
	a7 Business
	b% Scientific
	c% Computer Science Education
	d % Computer Support Services for Education in Engineering and Science
	e viner: Please explain:

- 9. Estimate the extent to which your organization employs each of the following computer procedures:
 - a. ____% Information Storage and Retrieval
 - b. Z Fiscal and accounting control
 - c. % Payroll
 - d. % Inventory Control
 - e. ____ % Statistical and Scientific Analysis
 - f. % Simulation and/or Mathematical Modeling
 - g. % Other: Please describe:

II. <u>SIMULATION AND MODELING</u> (In the following sections, please answer YES or NO, if possible)

10. Does your organization use its computation resources to perform mathematical modeling and/or simulation?

- 11. Are your mathematical models generally programmed in a procedure-oriented language such as FORTRAN, ALGOL, BASIC, etc.?
- 12. Are your modeling programs often executed with changes in the problem parameters but not in the model itself?
- 13. Is the cost of computation time a serious consideration in running your modeling programs?
- 14. Is computer center turn-around in batch mode operation a serious deterrent to the use of modeling in your organization?
- 15. Does your organization have a satisfactory capability for running its modeling programs interactively?
- 16. Would better analysis be obtained if these models were run more frequently?
- 17. Is your computing environment such that a knowledge of discrete step processes is sufficient to effect a relationship with the processes to be modeled?
- 18. Are continuous system simulation languages such as DYNAMO, MIMIC, etc. used routinely in your modeling and simulation studies? ______
- 19. Does digital simulation generally satisfy your "real time" requirements in the use of models?
- 20. Are the actual processes controlled within your environment essentially continuous with respect to time?
- 21. Is your environment confronted with high data rate sampled data problems?
- 22. Typically, in your environment, is a sampled data problem concerned with more than 50,000 bits per second per data point acquisition?
- 23. Typically, in your environment, is a sampled data problem concerned with less than 12,000 bits per second per data point acquisition?
- 24. Would your organization like to increase the extent to which computation facilities are used to perform modeling and/or simulation studies?
- 25. Is your organization adequately staffed to carry its required modeling and/or simulation studies?
- 26. Will your organization's modeling and simulation activity increase significantly in the next three to five years?
- III. USE OF ANALOG/HYBRID COMPUTATION AND CONTROL EQUIPMENT
- 27. Does your organization currently employ some type of analog/hybrid computation equipment on an in-house or contract basis?
- 28. Does your central computing facility manage an analog/hybrid computer?
- 29. Does your organization procure analog/hybrid computation services from outside agencies as needed?
- 30. Has someone in your organization been identified as having central responsibility for analog/hybrid computation?
- 31. Are your analog/hybrid facilities operated as service centers?
- 32. Are your analog/hybrid facilities confined to specific applications?
- 33. Is your use of analog/hybrid equipment principally at the data gathering interface of a digital control system?
- 34. Estimate the degree to which your application of analog/hybrid equipment is in the following areas:
 - a. ____ % Process Control
 - b. ____ % General Purpose Modeling and Analysis
 - c. ____% Other: Please specify

35. In the next five years, the use of analog/hybrid equipment by your organization will probably (Circle one): a. be increased

b. be decreased

c. remain the same

26	To the encoderation is shown a trend to (Chark and).
36.	In the organization is there a trend to (check one):
	a phase out analog/hybrid process control centers
	b make new investments in analog/hybrid process control centers
27	To these a recording and in your experiention for engles/hubbid computation compliation?
37.	The new experimentary have black to add enclos/hubid computation computation capabilities:
30.	bees your organization have plans to add analog/hybrid computation services to your computation facility?
39,	bees your organization use computation equipment to carry out some type of process control?
40.	Does your organization use computers in the area of general purpose modeling and analysis?
41.	Does your organization expect that large scale integration will have a significant impact on the future of analog computers?
	Explain:
	May I contact you for further information?
42.	Does your organization forsee major improvements in analog implementation through the use of Analog I.C.'s?
	Explain:
	May I contact you for further information?
IV.	COMPUTER SCIENCE EDUCATION
43.	For your organizational needs in the area of modeling and/or simulation do the computer science majors you hire generally display a sufficient comprehension of process dynamics?
44.	In what areas of preparation do you think present computer science educational programs are deficient?
45.	In what areas of preparation do you feel that computer science education programs should reduce their training
	efforts?
40.	in what areas do you find that your organization has to give computer science graduates an excessive amount of extensive on-the-job training?
	• •
Thank	you for your help in completing this inquiry. It will enhance our ability to interpret the results of this
surve tion	y if you will supply us with the descriptive data asked for below. As you might expect, none of this informa- will be used in a manner that will allow any individual respondent to be identified.
v.	DESCRIPTIVE DATA
47.	JEX: I F
48.	Age: 25 or Younger 46 - 50
	- 26 - 30 - 51 - 55 - 60
	36 - 40 61 - 65
	41 - 45 Over 65

FIGURE 4

185

.

.

•

-

.

49. Academic Degrees, if any

4

Degree		Year Granted
· · · · · · · · · · · · · · · · · · ·		
Please indicate the approxim work settings.	ate number of years experies	nce you have had in the following positions ork
Dendedan - Vm-	Set	ting (C. J)
rosition irs.	Exp. Co	
Manager/Administrator		1. rederal Govt. Civilian 2. Federal Govt. Military
Scientist		3. Other Govt.
Engineer		4. University or other Education Institution
Educator		5. Business
Consultant		6. Industry
Other: Specify		/. Other: Specify
Computer Operations Background:		
Area	Yrs. Exp.	work Setting (Same Codes as above)
Operations		
Recilities Management		
Software, bocumentation		
 No Experience 2. Some Knowle LANGUAGE An Assembler Languag 	dge 3. Moderate Experience	e 4. Proficient 5. High Level of Mastery RATING
FORTRAN		
ALGOL		
BASIC	•	
API.		
COBOL		
Simulation Languages	Such as GPSS Dynamo MIMI	
Other: Specify	back as orboy bytamo, intere	
Are there any other comments you	would like to add about any	of the topics mentioned in this survey?
· · · · · · · · · · · · · · · · · · ·		
Would you like to receive a perio concepts are related in the inter	dic newsletter dealing with est of improving the cost ef	topics in which Analog and Digital Computer ffectiveness of modeling and simulation?
Would you be willing to participa and science should take steps to	te in a workshop devoted to preserve the methodologies o	determining whether education in engineering of Analog computers?

.

A.P. Jensen, Principal Research Engineer School of Information & Computer Science Georgia Institute of Technology Atlanta, Georgia 30332

FIGURE 5

You may contact me personally for further information.

I would like to receive a Newsletter pertaining to cost effective modeling and simulation.

I would be willing to participate in a Workshop pertaining to the importance of analog problem solving methodologies and the future.

	FOLD HERE		
NAME :	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	
TITLE:			
ADDRESS:			
			•
		······································	
		ZIP:	
PHONE NUMBER:			
	· · · · · · · · · · · · · · · · · · ·		

FIGURE 6

WORK EXPERIENCE

Position	Avg. Yrs.*	Fed. Govt.	Military	Govt.	Univer- sity	Business	Industry	Other
Momt /Adm	5	26	13	7	34	81	130	9
Scientiet	2	18	8	3	34	10	42	6
Engineer	4	14	7	4	15	38	151	8
Educator	2		4	2	131	1	7	1
Concultant	1	10	4	4	10	33	47	6
Other	Ô	7	6	2	7	18	17	6
Operations	1	15	7	6	34	29	53	4
Facilities Mont	1	13	2	3	30	23	48	6
Software/Doc.	4	16	7	7	[,] 35	44	112	4

. .

,

~

* This column shows the average time in position for all responses.

.

,

188

•

Language	anguage None Some Moder		Moderate	Proficient	High Level		
ASSEM	50	84	66	73	82		
FORTRAN	18	70	76	115	126		
ALGOL	150	7 9	37	32	14		
BASIC	66	101	83	61	46		
APL	153	9 5	39	12	11		
COBOL	143	9 5	48	25	16		
SIM.	124	110	57	23	16		
OTHER	15	- 11	28	26	22		

COMPUTER LANGUAGE EXPERIENCE *

* The numbers in each column show the number of respondents who rated themselves at each level of proficiency in each language.

FIGURE 8

SURVEY RESULTS

•

Question			No	<u>Yes</u>	<u>N.A.</u>
Number	No	Yes	Yes+No	Yes+No	Total
51	7	415	. 02	.98	.09
SR	32	188	. 15	. 85	.53
50	16	67	.19	.81	.82
50 64	21	220	.09	.91	.48
6B	11	194	.05	• 95 [°]	.56
6C		96	.03	.97	.79
10	52	402	.11	. 89	.02
11	34	377	.08	.92	.12
12	71	321	. 18	. 82	.16
13	240	169	.59	.41	.12
14	299	104	.74	.26	.13
15	145	261	. 36	.64	.13
16	195	141	.58	. 42	.28
17	95	210	.31	. 69	.34
18	317	67	.83	. 17	.17
19	73	296	.20	.80	.21
20	133	222	. 37	.63	.24
21	219	148	.60	.40	.21
22	251	68	. 79	.21	.31
23	101	208	.33	.67	.34
24	148	224	.40	.60	.20
25	147	246	. 37	.63	. 15
26	135	237	.36	• 64	.20
27 ·	267	159	.63	.37	.08
[·] 28	332	52	.86	.14	. 17
29	329	49	.87	.13	. 19
30	253	115	.69	. 31	.21
31	233	58	.80	.20	. 37
32	161	112	.59	.41	.41
33	185	68	.73	.27	.46
36A	8	67	.11	. 89	.84
36B	11	5 3	.17	.83	. 86
36C	7	170	.04	.96	.62
37	230	139	.62	.38	.20
38	338	29	.92	.08	.21
39 '	181	199	.48	.52	. 18
40	53	340	.13	.87	. 15
41A	1 15	96	.55	.45	.55
41C	71	120	. 37	.63	.59
42A	115	59	.66	. 34	.63
42C	55	86	• . 39	.61	. /0
43	125	111	.53	.47	.49
54	167	211	.44	.56	.18
55	286	71	. 80	. 20	.23

PERCENTAGE BREAKDOWNS*

,

1

.

-

.

Quest.		0-5%	6-10%	11-20%	21-40%	41-60%	61-80%	81-90%	91-95%	96-100%
	Computing Environment									
8A	Business	148	36	35	61	43	39	34	17	52
8B	Scientific	142	37	39	55	49	48	35	19	41
8C	Computer Sc. Ed.	407	12	13	19	7	4	0	0	3
8D	Computer Support for Sc.&	395	21	18	21	7	2	0	0	1
8E	Other Eng.	391	15	8	14	13	10	2	2	10
	Application Areas									
9A	Inf. Storage & Return	219	60	69	51	29	16	13	3	5
9B	Fiscal & Accounting	253	46	46	50	33	9	12	3	13
9C	Payroll	342	-50	23	7	6	5	8	2	22
9D	Inventory Control	338	38	31	24	14	6	5	1	8 [`]
9E	Statistical & Sc. Analysis	211	40	52	77	41	29	6	0	9
9F	Simulation & Modeling	226	57	55	47	31	25	11	3	10
9G	Other	365	14	24	19	25	12	3	0	3
	Analog/Hybrid Applications									
34 A	Process Control	378	11	11	12	5	11	8	3	26
34B	General Purpose Modeling	358	13	14	· 8	7	18	12	2	33
34C	Other	421	7	7	5	1	. 4	4	2	14

* The numbers in each % column indicate the number of responses for each question which requested a reply in the form of a percentage.

*

4

FIGURE 10

6. AN INSTRUCTIONAL AND RESEARCH FACILITY: Outline of a Proposed Facility for Syndetic Computation

Abstract

This paper broadly outlines the development of an instructional and research facility intended to promote "syndetic" computational techniques that bind and integrate analog and digital processes in a single problem-solving environment. The program planned for development has among its objectives the following: 1) to develop, specify and evaluate the scope and nature of analog and digital computer facilities needed to support programs of education sensitive to the real world and its new demands on information processing; and 2) to develop a comprehensive instructional component of the computer science curriculum imparting competence in the design and application of analog and digital components in the composition of syndetic systems.

Introduction

Mathematics is a science of counts and measures, a science, according to <u>Webster's Third New International Dictionary</u>, "that deals with the relationship and symbolism of numbers and magnitudes and that includes quantitative operations and the solution of quantitative problems." Nor does the computational nature of mathematics detract from the grandeur of that science. To the contrary, a survey of the entire mathematical literature will reveal that both of the major historical tendencies of mathematics --the inclination of mathematical thought to build upon repetitious steps that submit to automation, and its inclination to reduce

abstract reasoning to physical models -- are directly related to mathematics as a formal discipline in which it is possible to begin with axioms and deduce consequences; indeed, it is certainly the case that Turing's foundational work on computation not only contributed to the modern concept of the computer but also to the fuller understanding of mathematics as a unified whole.¹

However, the wholeness and unity of mathematics is not always evident in the historical development of calculating devices -as witness the recent history of electronic analog and digital computing machines. An analog computer, of course, is a device in which numbers are measured by physical quantities and in which equations or mathematical relations are represented by distinct components corresponding to the individual mathematical operations, such as integration, addition or multiplication. Digital computers count discretely, never varying or responding in degrees but only to distinct signals. Developments in analog computation have been almost entirely eclipsed by the far more vigorous exploitation, refinement and promotion of digital concepts. Thus, the rapid advances in digital computer technology over the past 15 years have virtually smothered the earlier debate (mainly in the 1950's) about the relative merits of analog and digital computing. The success of digital computers has now so clearly won domination over the field of computation that today no major computer science program places any emphasis on analog computation. For that matter, the ACM-68 Computer Science Curriculum course chart pays it only token homage (in a single block whose only antecedent is introductory programming and which has no successor blocks). Furthermore, the major digital computer manufacturers have not found the demand for machines with analog features to be sufficient to justify entry into the market.

The basic limitation of analog computation has always been its limited precision, which results from employing quantities that can vary continuously in magnitude; the consequence of such variations is not only inaccuracies in individual operations, but also the combined effects that further downgrade the precision of the calculations and their representation. As a result, the primary value of analog computation has been to make available an electronic model that can be used for experimentation, rather than to obtain a precise solution of mathematical problems as such. Attempts to overcome this limitation have been largely responsible for the development of hybrid computers, in which, in the typical case, numerical information is transferred back and forth between conventionally circuited analog and digital equipment, by means of appropriate analog-to-digital and digital-to-analog data conversion hardware.

The advocates of hybrid computing have fared scarcely better than did those who had championed the earlier analog techniques. Although the practitioners of hybrid computing endure (by virtue of their economic advantage in narrow specialized problem areas), changes in the economy negatively affected major hybrid computer laboratories in aerospace industries established during the prosperous times of the 1960's and early 1970's. A similar fate came to most of the hybrid computer enterprises within academic institutions. Those enterprises were generally sustained by the energies of individual workers; they suffered from the underdevelopment of early technology; and they were inadequately funded (and hence never achieved a scale which might allow success). As a result, they never became a part of the central thrust in computing at their institutions. Thus, academic hybrid computing in the United States has all but disappeared in the face of institutional commitments to central digital facilities.

The Future of Hybrid Technology

At the beginning of this project, there were excellent reasons to believe that a new appreciation of hybrid technology was about to occur. Of these reasons, the economic ones were, if not paramount, at least compelling. With the economy stagnating, and with the digital computer failing to meet the computing power requirements of economic, sociological and ecological modeling, attention was in specific instances beginning to be focused on more economical means of certain computations. In such situations, the naturally parallel characteristics of analog computation offer cost advantages of hundreds-to-one over large-scale digital computers. (In many cases, this advantage is thousands-to-one.) It also seemed that there were signs of a new appreciation of the possibilities of hybrid computation in European countries, where it seemed to be realized that not all work could be done economically on the digital system alone; the vitality of analog and hybrid computing seemed widely evident throughout Europe, both in industry and in academic/research institutions.²

The probable re-emergence of hybrid technology was further augured by proposed developments involving the utilization of analog and hybrid techniques. For example, experience at Reed College had clearly demonstrated the viability of a system which used timesharing and communications capabilities to distribute the enormous solution bandwidth of analog computation to multiple users at remote locations.³ Along the same lines but on a far more ambitious scale, the Army Materiel Command had designed an "Advanced Hybrid Computing System."⁴ These two developments shared an appreciation for the obvious advantages which accrue when the analog subsystem of a hybrid can be programmed automatically from the digital computer. Some of the advantages advertised were: 1) the elimination of patchboards for multiple problem storage

(thereby achieving a substantial cost saving and obtaining the convenience of conventional digital storage of programs for the analog subsystem); 2) the use of a digital processor control of the analog subsystem (thereby allowing the implementation of high-level compiler systems and opening up the computer power to the nonspecialist); and 3) the achievement of tremendous computing speed that measures problem changeover in milliseconds (thereby opening up a whole new area for exploiting the speed of hybrid computation -- i.e., the operation of hybrid computers on a timeshared basis using remote terminals). In short, both the Reed College and the AMC endeavors took advantage of the fact that recent developments in solid-state switching devices had established a base for easily programmed hybrid computers which had the capability of being efficiently distributed among many users and which now make it possible to offer the inherent man-machine communication and interaction advantages of hybrid terminals in solving dynamic problems.

Another approach to the problem of configuring a hybrid computing system was to develop specialized digital equipment to perform accurate computations on a serial basis in interacting with the analog equipment on a multiplexed basis, so that the results of the digital computation appear serially in different parts of the analog device in a sequence fast enough to be equivalent to analog components. Another approach was to develop hardware and software for a parallel digital processor, which employed parallel computational elements to perform calculations simultaneously and thus handle, digitally, computational tasks historically reserved for analog or hybrid components. An interesting example of the latter approach was Denelcor's proposal to develop a so-called "Heterogeneous Element Processor."⁵ The Denelcor plan sought

to overcome the various shortcomings of analog hardware (such as scaling problems, repeatability problems and problems associated with the difficulty of programming hybrid devices), and to obtain in their place all the corresponding advantages of digital computation: floating point hardware, precision, stored program capability and communication with the computer via a high-level language. According to this plan, the parallel digital processor should contain a suitable number of pipelined compact modules (1.e., modules having an internal set of operations that minimize bus burden) which communicate information from one to another via high-speed data buses. These data transfers must be free-running from register to register in order to avoid slow-speed memory accessing. The processor can be loaded and operated under program control from a minicomputer and can communicate I/O to the minicomputer through a buffer memory which also can be accessed by the bus. Analog I/O can be implemented with MDAC first-hand registers and ADC demultiplexed buffer registers which are also accessed by the bus. A parallel processor with these features would for many applications run faster than real-time with good accuracy, and would produce repeatable data. Such ambitions have only partially materialized over the last two years as a result of higher order complexities introduced in the scheduling, control and programming of such systems.

Syndetic Computation

Imaginative developments and plans such as those cited above, (i.e., developments in autopatching, communications and digital bus structure concepts) have not shown the progress necessary to revitalize large-scale hybrid computing. The hoped-for revitalization manifesting itself in the creation of a new and

unified computation technology has not emerged. Nor does it appear that it will. At best it seems destined to seek and find those specialized applications which demand it by virtue of their natures.

This does not eliminate the need to educationally recognize the important precepts of which hybrid computing was built -the need to combine and use the merits of both analog and digital computers. The word <u>hybrid</u>, from the Greek, alluded originally to the casual alliance of a tame sow and a wild boar. It is not necessary to specify a precise mapping of this relationship into the world of analog and digital computation to know that the word is all too descriptive of the uneven history of hybrid computation, which has been more a mixture than a blend. What is needed now is a new approach entirely, a <u>syndetic</u> approach which truly binds and integrates analog and digital devices and concepts into a single, unified problem-solving information processing environment, reflective of the unity and richness of mathematics itself.

Toward this end, the School of Information and Computer Science, Georgia Institute of Technology, desires to complement its already existing, well-developed Information and Computer Science Laboratory with a comprehensive instructional program in syndetic digital-analog computation. The new program would have among its objectives, the following:

> To develop, specify and evaluate the scope and nature of analog and digital computer facilities needed to support programs of education sensitive to the real world and its new demands on information processing; and,

2. To develop a comprehensive laboratory instructional component of the computer science curriculum imparting competence in the design and application of analog and digital components in the composition of syndetic systems.

To accomplish these objectives, a comprehensive laboratory program is required. A laboratory curriculum which spans the undergraduate program and which "capstones" the professional masters degree program is called for. The facilities of this laboratory while extensive need not be extremely costly but must be flexible and not committed to any given computer of any scope or type. Nor should the laboratory be committed to a specific "learning station" concept. Rather, it should include a large collection of devices analagous to the Bunsen burners and test tubes of a chemistry lab.

The "test tubes" of a syndetic computing laboratory are the integrated circuits, prototyping sockets, power supplies, light emitting diodes, crystals, resistors, capacitors, operational amplifiers, transducers, etc., required to build small digital computers, integrators, digital-to-analog converters, analog-todigital converters, clocks, etc., in the process of generating special purpose computer systems and models.

With these facilities, it would be possible to illustrate to students the relationships between analog and digital processes and to build up digital systems which illustrate how a computer really works. At one point in computing history the cardboard CARDIAC computer was effective in illustrating conceptually how a computer works. Today, it is neither necessary nor adequate to illustrate the conceptual workings of a computer. A simple functioning machine can be built which involves stored program memory, control store concepts, working registers with visual

displays and variable speed clocks that allow a student to actually see each register transfer take place. Once the workings of the system are clear then by running it at increased speeds, it can be analyzed with respect to limiting conditions and its economic productivity for specific classes of problems.

What is envisioned is a laboratory which provides simple work benches with power outlets and good lighting, a controlled storage of hardware components and selected higher order facilities such as those that permit the programming and erasure of PROM's, a staff which is competent at building illustrative and functioning devices as needed, and a carefully developed and maintained series of laboratory experiments and demonstrations.

Defining a work station as a prepared area at a classical work bench large enough to support a student's activity (approximately an area 3 ft. by 4 ft.), an initial investment in materials, supplies and equipment of \$2,000 and an annual maintenance cost of \$500 per station would permit twelve students per quarter to have access to a syndetic computing experience. (This assumes two three-hour sessions per week per student and a team of two students per session.) Ten such work stations would serve 120 students per quarter or possibly 4 different courses of 30 students each.

The laboratory program would serve three undergraduate and one graduate course. In addition, the lab would provide demonstrations supportive of introductory courses in computer structures. The three undergraduate courses would comprise one third-quarter sophomore course, one second-quarter junior course and one firstquarter senior course. The graudate course would be predicated on prerequisites which would prevent its being taken prior to the second or third quarter of a four-quarter graduate program.

This laboratory program would require the support of a full time engineering technician capable of interacting professionally with students and faculty.

Pursuant to the objective set forth in this paper and carefully coordinated with other aspects of the information and computer science program, a proposal is being prepared for submission to the National Science Foundation. This proposal will request approximately \$50,000 to support the development of the curriculum, laboratory manuals and facilities of a syndetic computing laboratory.

References

- 1. F. J. Murray, "Mathematics as a Calculatory Science," Encycl. Brit., 15th ed. (1975), 11:671-696.
- P. H. Enslow, Jr., <u>Hybrid Computer Technology in Europe:</u> <u>A State of the Art Survey</u>. ERO Technical Report ERO-1-74, U.S. Army Research and Development Group (Europe), London, England, February 15, 1974.
- 3. F. D. Tarbutt, Proposal for a Pilot, Medium-Size Hybrid Computer System for Reed College. Proposal submitted to the Special Project Section, Office of Computing Activities, National Science Foundation, by Reed College, Portland, Oregon, May 10, 1968. Also: F. D. Tarbutt, <u>Development of a Medium-Sized Time-Shared Analog Computer: A Study of its</u> <u>Application to Undergraduate Instruction</u>. Detailed Final Report, NSF Grant GJ-32, Reed College, Portland, Oregon, February 29, 1972.
- A. G. Edwards, et al., <u>Advanced Hybrid Computer Systems</u>.
 U.S. Army Materiel Command, Alexandria, Virginia, June 1973.
- 5. Denelcor, Inc., <u>Technology Evaluation of a Heterogeneous</u> <u>Element Processor: Final Report</u>. Denelcor, Inc., Denver, Colorado, May 1, 1974.