AMRL-TR-65-97

A DEMONSTRATION HYBRID COMPUTER FOR REAL-TIME FLIGHT SIMULATION

MARK E. CONNELLY
OLEG FEDOROFF

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 1965

BEHAVIORAL SCIENCES LABORATORY
AEROSPACE MEDICAL RESEARCH LABORATORIES
AEROSPACE MEDICAL DIVISION
AIR FORCE SYSTEMS COMMAND
WRICHT-PATTERSON AIR FORCE BASE, OHIO

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Requests for copies of this report should be directed to either of the addressees listed below, as applicable:

Federal Government agencies and their contractors registered with Defense Documentation Center (DDC):

DDC
Cameron Station
Alexandria, Virginia 22314

Non-DDC users (stock quantities are available for sale from):

Chief, Input Section
Clearinghouse for Federal Scientific & Technical Information (CFSTI)
Sills Building
5285 Port Royal Road
Springfield, Virginia 22151

Change of Address

Organizations and individuals receiving reports via the Aerospace Medical Research Laboratories automatic mailing lists should submit the addressograph plate stamp on the report envelope or refer to the code number when corresponding about change of address or cancellation.

Do not return this copy. Retain or destroy.

800 - July 1965 - 448-49-1059

BLANK PAGE

A DEMONSTRATION HYBRID COMPUTER FOR REAL-TIME FLIGHT SIMULATION

MARK E. CONNELLY
OLEG FEDOROFF

FOREWORD

This report is a summary review of the system design, construction and demonstration of a hybrid flight simulator carried out by the Electronic Systems Laboratory of the Massachusetts Institute of Technology, Cambridge, Massachusetts, over the period from May, 1961 to August, 1964. The work was sponsored by the Behavioral Sciences Laboratory of the Aerospace Medical Research Laboratories, Aerospace Medical Division, Wright-Patterson Air Force Base, Ohio, under Contract AF 33(616)-8363. The research was in support of Project No. 6114 and Task No. 611408

Mark E. Connelly was the principal investigator at M.I.T. The intelligent and patient guidance of Don R. Gum, Contract Monitor, and his associates Carl McNulty and Robert Cameron is gratefully acknowledged. William B. Goeckler was the Contract Monitor at the inception of the project. The authors further acknowledge the many contributions made by Henry Kerr, Louis Krasny, Alf Solbakken, and James Tang to the completion of the work.

This technical report has been reviewed and is approved.

WALTER F. GRETHER, PhD
Technical Director
Behavioral Sciences Laboratory

ABSTRACT

A hybrid, real-time simulation facility has been designed, constructed, and demonstrated, using as a test vehicle the complete aerodynamic and engine equations for a high-performance military aircraft. The analog-digital configuration employs peripheral analog equipment to represent a linear, skeleton version of the aircraft and the PDP-1 digital computer to carry out engine simulation, decision management, and corrections for nonlinear effects.

To provide an all-digital reference against which the hybrid simulation could be compared, the aircraft model, which in general scope is identical to the F-100A model used in the UDOFT studies, was also simulated on the PDP-1 alone. It was found that the solution rate of 20 per second employed in the all-digital study could be reduced to one per second without deleterious effects when the hybrid configuration was used. Such a reduction demonstrates that supplementing a digital computer by relatively inexpensive analog peripheral equipment in the manner suggested in this report substantially increases the real-time capacity of the digital computer in complex simulation applications. Moreover, because a number of key variables are computed continuously in the analog domain, the introduction of analog equipment results in a net decrease in the complexity of the interface, particularly in the number of analog storage devices required.

The solid-state analog sub-system, designed and built at M.I.T., represents an order of magnitude reduction in cost and size relative to commerical analog computers of equivalent capacity. This improvement is largely due to a novel photoresistive multiplier developed by the project. Wide-bandwidth analog elements used on a time-shared basis are employed at the interface to generate running sums of products. This capability, however, is not an essential feature of the present hybrid concept.

This report presents a detailed description of the hybrid design, the F-100A model, the system components, the PDP-1 programs, and the test phase. Tutorial discussions of scaling, key algorithms, model making, logical design, and the state-of-the-art in hybrid components are also included.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS AND TABLES

LIST OF ILLUSTRATIONS AND TABLES (Continued)

EXPLANATION OF TERMS

A. Basic Aerodynamics

a	linear acceleration
a	speed of sound
a_o	speed of sound at sea level
α	angle of attack
A	pertaining to ailerons
AEA, AER, AEP, AAF	aeroelastic functions
β	sideslip angle
b	pertaining to body axes
b	wing span
BL, BR	left brake; right brake
BER	aeroelastic function
c	mean aerodynamic chord (M.A.C.)
C	nondimensional aerodynamic coefficient
c.g.	center of gravity
d	migration of c.g. from 35 percent M.A.C.
D	drag (negative x axis)
DT	pertaining to drop tank
DC	pertaining to drag chute
δA	aileron deflection
δA_t	aileron trim
δH	horizontal stabilizer deflection
δH_t	horizontal stabilizer trim
δR	rudder deflection
δJ	speed brake deflection (SB)
F	force
f	nonlinear function
g	acceleration due to gravity
GE	pertaining to ground effect
h	altitude
H	pertaining to horizontal stabilizer
i, j, k	unit vectors along x, y, and z axes
I_x, I_y, I_z	moments of inertia
I_{xz}	product of inertia
i	pertaining to inertial axes
i	pertaining to instantaneous value
l, m, n	direction cosines with respect to x, y, and z inertial axes
l, m, n	pertaining to roll, pitch, and yaw moments about x, y, z axes
L, M, N	moment components along x, y, and z axes
L	lift (negative z axis)
LG	pertaining to landing gear
l_{cw}	moment arm of gravity vector about main wheels
m	mass
Ma	Mach number
n	load factor (g's)
NW	pertaining to nosewheel

p, q, r	components of angular rate along x, y, and z body axes
ρ	air density
q_1	dynamic pressure
R/C	rate of climb
R	pertaining to rudder
S	wing area
s	pertaining to stability axes
t	time
T	thrust
T	moment (torque)
u, v, w	components of velocity along x, y, and z body axes
V	velocity
w	pertaining to wind axes
X, Y, Z	force components along x, y, and z axes
x, y, z	pertaining to x, y, and z axis components
x_e, x_n	east and north coordinates
θ, ψ, ϕ	Euler angles in elevation, heading and roll

B. Engine and Hydraulic Systems

A/B	afterburner
Δ	increment or decrement
Δs	stick-movement
δT	throttle setting (also ξ)
δSH	fore-and-aft stick deflection
δPR	rudder pedal deflection
ef	pertaining to external fuel (drop tank fuel)
EF	emergency fuel
f	pertaining to fuel
F_n	thrust
G_1, G_2	hydraulic pressure of No. 1 and No. 2 systems
hpl	hydraulic pressure limit (HPLL)
HM	hinge-moment
N_2	revolutions per minute
P	hydraulic pressure
pf	percent thrust (PFN)
pr	pertaining to rudder pedal
Q_i	quantity of ice
RPM	engine revolutions per minute
S_i	icing rate
tas	true airspeed
TPT	tailpipe temperature
wa	pertaining to airduct
W_f	fuel flow rate
WM	pertaining to windmilling effect

C. Miscellaneous

AC	accumulator
AD	analog-to-digital
ADC	analog-to-digital converter (ENG)

CM	core memory
CRT	cathode ray tube (CRO)
DA	digital-to-analog
DAC	digital-to-analog converter
DEC	Digital Equipment Corp.
DSO	discrete output
DSI	discrete input
FF	flip-flop
IO	in-out register
MA	memory address register
MB	memory buffer register
MBD	memory buffer register decoder
OPG	operate gates instruction
PC	program counter
TSAL	time-sharing assignment level
→	replaces
C(Y)	contents of memory register or arithmetic element Y
(X)	register whose content is X

BLANK PAGE

CHAPTER I

OBJECTIVES OF THE SYSTEM DESIGN AND SYSTEM DEMONSTRATION

1.1 INTRODUCTION

On May 1, 1961, the Air Force Aeronautical Systems Division initiated a one-year design study with the M.I.T. Electronic Systems Laboratory to determine the system design and logical organization of a combined analog-digital computer suitable for the real-time simulation of complex dynamic systems. In addition, a digital program based on the functional characteristics of this hybrid computer was to be prepared for the F-100A aircraft and further development work carried out on a family of high-speed analog computing elements.

The objectives of this first year effort were to indicate the feasibility and the potential advantages of the hybrid simulator with an acceptable degree of confidence to justify the construction and demonstration of such a computer. The system design was also to be in sufficient detail to serve as the basis for the construction phase to follow.

The results of this study were encouraging enough to warrant undertaking actual construction of a demonstration hybrid simulator incorporating the PDP-1 digital computer, a solid-state analog computer designed and built at M.I.T., single analog-digital conversion channels, analog sampling and holding gates activated under computer control, and a family of wide-bandwidth analog computing elements operating at the interface between the analog and digital domains. This hybrid system, which is shown in Fig. 1.1, was completed and tested by May, 1964. The steady-state performance and transient results showed good agreement with a previous all-digital simulation of the F-100A on the PDP-1 alone and with the results obtained on essentially the same F-100A model with UDOFT. Moreover, this fidelity was maintained at solution rates as low as one per second compared to the 20 solutions per second employed in the all-digital simulation using trapezoidal integration. As a consequence, it was concluded that the addition of a modest amount of analog equipment on the periphery of a digital computer in the manner suggested in this report substantially increases the real-time capacity of the digital computer. In addition, since certain computer variables are required in the analog domain for cockpit displays and a number of pilot controls originate in the analog domain, a judicious division of the computation load between the analog and digital equipment reduces the analog-digital information traffic and reduces the amount of interface equipment, particularly storage gates.

This report is a terminal document summarizing the results of the entire program. It includes a treatment of the F-100A model, the hybrid system design, the all-digital and hybrid programs, component development, and experimental test results. Although the discussion is focused on a specific vehicle, the F-100A, the report is intended to be more generally helpful in the following areas:

a) The establishment of a quantitative measure of the relative fidelity and real-time capacity of all-digital and hybrid simulation computers. By extrapolation, it is hoped that this measure will be useful as a guide in the design of simulation facilities in general, particularly those directed to the solution of more complex models.

b) An evaluation of the present state-of-the-art- in hybrid computing elements, both with respect to performance and to cost. As an example, the 75-amplifier solid-state computer designed and built for the M.I.T. hybrid system represents an order of magnitude reduction in cost and size relative to commerical analog computers of the same capacity. This improvement is due largely to a novel photoresistive multiplier developed by the project. In addition, the performance characteristics of analog-digital conversion units, wide-bandwidth pulsed analog units, sampling and holding gates, and display devices have been assessed experimentally. Moreover, further insight has been gained with respect to the interface control logic and the order code requirements imposed on the digital portion of the system.

c) A tutorial exposition of the model-making and programming procedures suitable for digital and hybrid computation. Included in the report are discussions of binary analog and digital scaling, flow charts, key algorithms such as those used in function generation, the effect of solution rate, time and memory requirements, display, input-output, and utility routines.

d) The hybrid configuration described here was designed to be useful for a wider range of scientific and engineering problems and it is hoped that the report will stimulate efforts to investigate these other applications.

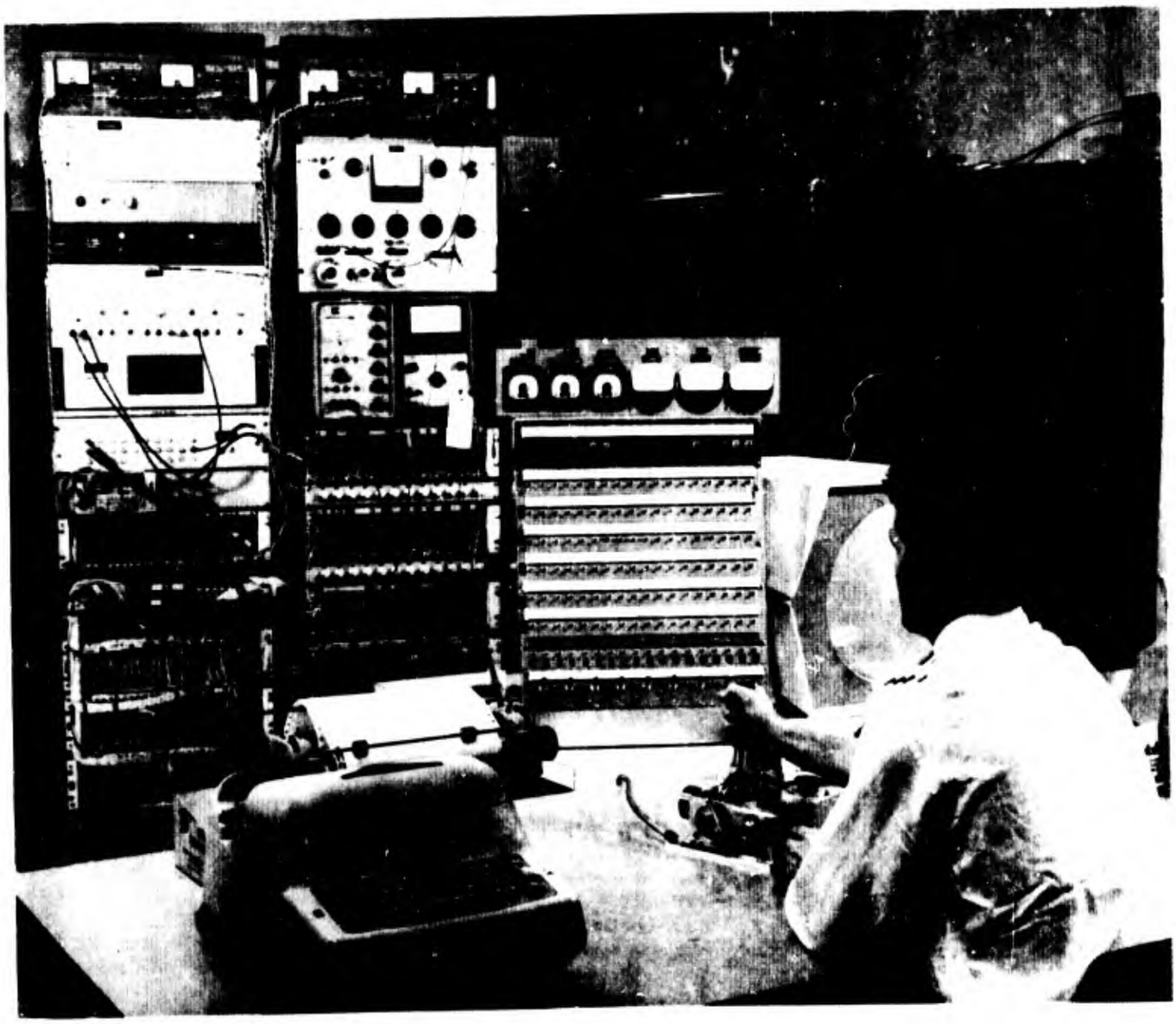


Fig. 1.1 M.I.T. Hybrid Demonstration Computer

1.2 HISTORICAL AND TECHNICAL REVIEW

Prior to undertaking the present work for the Air Force Aerospace Medical Research Laboratories, the M.I.T. Electronic Systems Laboratory carried out a four-year research program on the use of analog-digital computers for operational flight trainers under the sponsorship of the U.S. Naval Training Device Center. In the course of this work, a number of system designs were generated for comparative evaluation and several representative analog-digital systems were tested experimentally. In 1958, the first of these systems, employing analog integrators linked to the Whirlwind digital computer by analog-digital conversion equipment, successfully simulated an aircraft with six degrees of freedom in real-time.*[1, 2] In that same year, a second experimental configuration demonstrated the feasibility of using high-speed, time-shared analog computing elements under digital sequence control.[3]

Under the sponsorship of the Air Force Cambridge Research Laboratories, a system was assembled in 1960 in which pulsed-analog elements were operated under the control of the M.I.T. TX-0 digital computer. The data and program for all calculations were stored in the TX-0 core memory. Analog-digital conversion channels provided for the transfer of data between the analog and digital domain. The TX-0 hybrid system was employed successfully for function generation,[4] correlation of radar blip data,[5] and trigonometric resolution.[6]

The terminal report for the Naval Training Device Center in 1961 presented a critical review of this and other systems and circuit design studies and demonstrations up to that point.[7] In addition, this report outlined a suggested pulsed analog-digital configuration for real-time aircraft simulation.

On the basis of this prior experience, the rapid technical advances of transistor machines, and the success of the Sylvania UDOFT computer[8, 9] in simulating the complete F-100A model in 1960, we concluded that a distinct reversal had occurred in the competitive position of analog and digital techniques for solving large-scale, real-time problems.[10] Our hypothesis was that relatively low cost digital computers were available that would be capable of solving F-100A class problems in real time and that such machines were competitive in price with analog computers.

To substantiate this hypothesis, Krasny programmed the complete F-100A aircraft equations on a real-time basis using a modified TX-0 order code. Krasny found that the most time-consuming route through his F-100A program required 46.87 msec. This is sufficient to guarantee a solution rate of 20 per second, which is adequate for real-time aircraft simulation since the natural aircraft frequencies lie below 2 cps. The total program, data and instructions, occupied 7233 registers of memory.[11]

The TX-0 is a solid-state digital computer built in 1956 by the M.I.T. Lincoln Laboratory. Since it is an experimental, one-of-a kind machine, it was not possible to assign a realistic cost figure to it. Starting with the Digital Equipment Corporation PDP-1 in 1960, however, more than a dozen commercial machines have been offered in the price range of $45,000 to $120,000 that are fast enough to handle the F-100A class of problems on a real-time basis.[14] With the advantages in accuracy, flexibility, dynamic range, reliability, size, and power consumption, and the manifest superiority in generating nonlinear functions and decision management, there has subsequently been a pronounced trend to digital techniques in the simulation field. As a further indication of this trend, all recent military and commercial flight trainer contracts have been based on digital implementation. In the case of general-purpose simulation facilities, it is even economically advantageous to pay a considerably higher initial cost for a digital computer, since in any computing facility the hardware investment is soon exceeded by the costs of operation, maintenance, and problem preparation. The optimum strategy favors a reduction of these continuing expenses.

A simple and direct economic comparision today between digital and analog techniques is complicated by the advent of the multi-programmed or time-shared mode of digital operation. As an illustration of this mode, the all-digital F-100A program described in this report can be run on the PDP-1 with two other users on the machine simultaneously, each user independently interacting with his program from a separate typewriter console. With such a capability, the cost of the machine can be divided among the separate users. For large, expensive computer installations, the possibility of scores of individuals using the machine simultaneously greatly reduces the cost to each user. The hybrid configuration suggested in this report diminishes the digital computing load to the point where several complex, real-time problems can be included in the multitude of simultaneous users. For custom-built installations to solve specific real-time problems, the hybrid configuration permits the use of a slower, less expensive digital computer. Several small digital computers like the Digital Equipment Corporation PDP-8, the Computer Control Company DDP 116 and the Scientific Data Systems SDS 92 are now available in the price range from $18,000 to $30,000.

---

* Superscripts refer to numbered items in the References.

Clearly the competitive situation between analog and digital techniques is by no means static. Whereas the price of central processors and core memory capacity has gone down substantially in the last four years, the introduction of solid-state circuitry in analog computers has made the previous digital advantages with respect to cost, size, power consumption, and reliability less pronounced. In 1960, commercial analog computer consoles with a full complement of multipliers and nonlinear equipment averaged over $1000 per amplifier. In the solid-state low voltage analog computers presently offered, however, this figure has been cut roughly in half. The analog computer described in this report indicates that further substantial reductions in the cost of analog equipment are possible, especially in situations not demanding the ultimate in accuracy and flexibility. In addition, the wide-bandwidth capability of analog equipment is now being exploited more fully in repetitive and iterative operating modes, often under the control of digital logic elements.[15] Notwithstanding the significant improvements in the analog state-of-the art, the digital advantages with respect to accuracy, dynamic range, and flexibility remain unchallenged, and steady progress continues toward higher operating speeds and more powerful order codes.

1.3 BASIC FEATURES OF THE SYSTEM DESIGN

The evidence available in 1961, therefore, indicated that a basic digital computer with analog-digital conversion equipment offered substantial advantages over the traditional analog implementation of large-scale simulations. A reasonable question at that point was whether analog and pulsed-analog techniques could contribute in any way to the effectiveness of the computer being considered. The unique and interesting features of the present design are based on the conviction that the answer to this question is affirmative.

One disturbing aspect of the Krasny study was that the F-100A problem approached the practical limit of the capability of the PDP-1 at a solution rate of 20 per second. Since the F-100A represents only a moderately complex model compared to multi-engine aircraft weapon systems and spacecraft, a simulation facility with a much greater real-time capacity would be desirable.

The most obvious solution is to handle the larger problems by operating several basic computers in parallel and exchanging information between them. A second approach is to increase the computing speed of a single machine. Sylvania has completed a comprehensive study of this possibility.[12, 13]

However, the M.I.T. simulation is based on a third approach, namely, to reduce the solution rate requirements of those portions of the problem carried out on a sampled basis by simulating the fundamental dynamic loops on a continuous basis in the analog domain. A moderate number of standard analog components peripheral to the PDP-1 simulate the aircraft force, moment, and Euler angle loops on a continuous, linear skeleton basis. Correction terms for the nonlinearities characteristic of the aircraft model are inserted into each of these loops on a sampled basis. These corrections are based on data stored in the digital memory, hence the inherent flexibility to change or modify models readily is retained. The correction technique is fully explained in Chapter III.

Assigning selected parts of the aircraft model to the analog domain reduces the computing load on the digital computer, particularly in the case of the fast dynamic loops which would have to be solved digitally at a relatively high solution rate. A carefully-planned division of the computation load also reduces the traffic of analog-digital information and the amount of interface equipment.

Pulsed-analog elements at the analog-digital interface are used to compute running sums of products and for channeling and storing analog information under digital program control. The pulsed-analog system is employed to multiply aerodynamic functions by control inputs in the analog domain, thus avoiding the frequent encoding of the rapidly-varying control signals. As stated previously, the digital part of the system is the PDP-1 computer operated by the M.I.T. Department of Electrical Engineering. The PDP-1 is a commercial outgrowth of the TX-0 computer with somewhat superior speed and order code features. Single channels of digital-to-analog and analog-to-digital conversion equipment provide for the exchange of analog and digital data at rates compatible with the operating speed of the PDP-1.

The sample and sample and hold gates in the pulsed analog subsystem are activated under digital program control, a special order having been added to the PDP-1 for this purpose. Seventy-two events in the analog domain (up to nine at a time) are controllable with the special order. In addition, the order is used to start the analog-to-digital conversion process. The digital-to-analog converter is connected directly to the accumulator of the PDP-1, hence any number appearing in the accumulator is converted to the corresponding analog voltage automatically without the necessity of a program instruction. Extra logic has also been added at the interface so that the standard PDP-1 in-out transfer instruction can be used to strobe the contents of any of seven 18-bit toggle switch registers or the encoder output into the in-out register. The in-out transfer instruction is also employed to load an 18-bit discrete output register, which in turn drives a bank of indicator lamps. These provisions are necessary to handle the discrete input and discrete output requirements of the F-100A simulation.

The pilot's continuous control inputs corresponding to the aileron and horizontal stabilizer are derived from a spring-loaded joystick with potentiometer transducers. Similar manual inputs for rudder and throttle, as well as aileron and stabilizer trim, are mounted on the same unit. Continuous outputs needed by the pilot are presented either on special panel meters (airspeed, altitude, rate of climb, acceleration) or on a CRT display (simulated attitude, heading).

In both the all-digital and hybrid programs, some increase in computing efficiency has been effected in the formulation of the F-100A model and in the choice of algorithms. Without sacrificing any of the detail of the Melpar and UDOFT F-100A model, a formulation more suitable for digital calculations has been evolved. For example, the generation of nonlinear functions is based on interpolation between discrete points stored in the digital memory. This procedure simplifies the data processing involved in the preparation of the problem and facilitates changes in the nonlinear details of the model. A single set of breakpoints is used for each independent variable to reduce the time spent in level-selecting. Wind axes instead of body axes are used for the force equations; Euler angles are employed instead of direction cosines. These improved model making and programming procedures are explained more fully in Chapters II and IV.

BLANK PAGE

CHAPTER II

THE F-100A AIRCRAFT MODEL

2.1 INTRODUCTION

In general scope, the F-100A model described in this chapter is identical to that used by Sylvania in the UDOFT digital simulation program.[9,17] The information flow diagram of Fig. 2.1 indicates the comprehensiveness of the UDOFT model. However, the UDOFT model was based on the equations employed by the Melpar analog OFT, hence was strongly influenced by the limitations of analog equipment. This is particularly true in the case of function generation, where Sylvania used complicated sums and products of functions of one variable to represent functions of more than one variable. In the M.I.T. aerodynamics simulation, function generation is based on interpolation between discrete points stored in the digital memory. To employ this technique, it was necessary to return to the original North American data on the F-100A aerodynamic characteristics.[16] Unfortunately, time did not permit the recasting of the Pratt and Whitney J57 engine model into a more rational form for digital computation, hence the engine simulation is still based on the Melpar-Sylvania model.[17]

Other changes in the M.I.T. model are that wind axes are used for the force equations instead of body axes and Euler angles are used instead of direction cosines to describe the orientation of the aircraft in space. Since the wind axes follow the motion of the aircraft center of gravity through space, they are a more stable computing reference than the body axes, all three of which are subject to rapid fluctuations.

A few minor simplifications have been incorporated, particularly with respect to the functional dependence of nonlinearities. These changes are discussed as they are introduced.

Prudent model-making, of course, is an integral part of any effective simulation, but the optimum model is, in part, determined by the computation techniques employed. In this particular study, the governing factors were fidelity to the actual aircraft, computation time, efficient use of computer memory, minimal data processing prior to the simulation, and ease of incorporating changes during the simulation.

Some question might be raised concerning the choice of the F-100A, an obsolete aircraft, as the subject of the demonstration. It was felt that this aircraft presented the same basic computational problems that a more up-to-date model would present, and that the organization of the data and flow graphs would be greatly facilitated by the previous work carried out by Sylvania on the UDOFT program. In addition, the Melpar and UDOFT simulation results and the extensive flight test data on the aircraft itself provide standards against which the hybrid system may be judged. Moreover, all data pertaining to the F-100A is now unclassified, and this permits the free exchange of information inside and outside of M.I.T.

To provide a reference against which the hybrid computer's operating characteristics and fidelity could be compared, the same F-100A model has been simulated in two ways, first with an all-digital program on the PDP-1 and second with the hybrid system described in Chapter III.

2.2 AERODYNAMIC EQUATIONS OF MOTION

Four axis systems are commonly employed in writing the equations of motion of an aircraft. These are the body, stability, wind and inertia axis systems. The inertia axes are fixed in inertia space, although for aircraft, the earth is an adequate local reference. The x-y inertia plane may be taken perpendicular to the gravity vector, let us say, with the x axis fixed in the direction of true north.

The other three axis systems have their origin at the nominal center of gravity of the aircraft at the 35 percent M.A.C. point. The body axes are fixed with respect to the aircraft, the x and z axes lying in the plane of symmetry. The three moment equations are invariably written in terms of these axes. The x wind axis is coincident with the velocity vector of the aircraft and the z wind axis lies in the planes of symmetry. Intermediate between the wind and body axes are the stability axes, the x stability axis being the projection of the x wind axis on the plane of symmetry. The angle between these two axes is the sideslip angle β; the angle between the x stability axis and the x body axis is the angle of attack α. The definition of these important aerodynamic angles is illustrated in Fig. 2.2.

In the present F-100A model, the orientation of the wind axis system with respect to the inertia axis system is specified by three Euler angles in pitch (θ_w), roll (ϕ_w) and heading (ψ_w). Frequently, Euler angles describing the orientation of the body axes are encountered (θ_b, ϕ_b, and ψ_b). A master diagram relating these various definitions is shown in Fig. 2.3.

As stated previously, direction cosines were employed in the UDOFT simulation instead of Euler angles. For the purposes of the hybrid demonstration, the added complexity of implementing the direction cosine equations with analog equipment just to be able to handle the rather unusual flight condition of $\theta_w = \pm 90^\circ$, did not seem to be worth the extra expense and effort required.

Aircraft aerodynamic data is customarily presented in terms of the stability axes. Since the moment equations are with respect to the body axes and the force equations are with respect to the wind axes, the stability axis coefficients should be modified, in theory at least, to yield correct forces and moments.

The present model assumes that the aerodynamic forces derived from stability axis data can be used directly in the wind axis summation of forces since the normal flight condition is with zero sideslip, consequently the wind and stability axis systems normally coincide. However, the normal flight condition is not necessarily with zero angle of attack, so a correction of the yaw and roll moment terms is necessary before they can be summed in the body axes. In the UDOFT program, the rotation of both force and moment components from stability axes to body axes takes place in the computer. These operations consume machine time and add to the memory requirement. As pointed out in Simulation of Aircraft,[18] however, it is also possible to modify stability axis aerodynamic coefficients on paper prior to their insertion into the computer, so that they can be used directly in the body axes. In the moment equations that follow, the original stability axis coefficients C_{ℓ_p}, C_{n_p}, C_{ℓ_r} and C_{n_r} have been so modified for use in the body axes. Because of their more complicated functional dependence, the coefficients $C_{\ell_{\delta A}}$, $C_{n_{\delta A}}$, $C_{\ell_{\delta R}}$, $C_{n_{\delta R}}$, C_{ℓ_β}, and C_{n_β} are rotated in the machine.

The aerodynamic equations used in the all-digital and hybrid simulation studies are as follows:

Total Aerodynamic Moment about x Body Axis (Roll)

$$L_b = \frac{\rho V^2 Sb}{2}\left[C_{\ell_{\delta A}}(\alpha, Ma)\cdot AEA(Ma, q_1)\cdot \delta A + C_{\ell_{\delta R}}(|\alpha|)\cdot f_2(Ma)\cdot AER(Ma, q_1)\cdot \delta R + C_{\ell_\beta}(Ma)\cdot f_3(\alpha)\cdot \beta + C_{\ell_p}(|\alpha|)\cdot AEP(Ma, q_1)\cdot \frac{pb}{2V} + C_{\ell_r}(|\alpha|)\cdot \frac{rb}{2V}\right] \quad (2.1)$$

Total Aerodynamic Moment about y Body Axis (Pitch)

$$M_b = \frac{\rho V^2 Sc}{2}\left[C_m(C_L, Ma) + AAE(Ma, q_1)\cdot C_L + C_{m_{\delta H}}(Ma, h)\cdot \delta H + C_{m(wa)_\alpha}(Ma)\cdot \alpha\cdot \delta T + C_{m(DT)}(Ma) + C_{m(DC)} + C_{m(GE)} + C_{m_{\delta J}}(Ma)\cdot \delta J + C_{m_q}(Ma, h)\cdot \frac{qc}{2V} + C_{m_{\dot\alpha}}(Ma, h)\cdot \frac{\dot\alpha c}{2V}\right] \quad (2.2)$$

where the ground effect term $C_{m(GE)} = k'(h - 25)\cdot C_L$ $\quad h \leq 25$ feet

$C_{m(GE)} = 0$ $\quad h > 25$ feet

The landing gear pitch term $C_{m(LG)}(\alpha)$ has been neglected.

Total Aerodynamic Moment about z Body Axis (Yaw)

$$N_b = \frac{\rho V^2 Sb}{2}\left[C_{n_\beta}\cdot \beta + C_{n_{\delta R}}(Ma)\cdot BER(Ma, q_1)\cdot \delta R + C_{n_{\delta A}}(\alpha)\cdot \delta A + C_{n_p}(|\alpha|)\cdot AEP(Ma, q_1)\cdot \frac{pb}{2V} + C_{n_r}(|\alpha|)\cdot \frac{rb}{2V}\right] \quad (2.3)$$

DISCRETE INPUTS (42)

TEMPERATURE HOT
START CRANK
START FIRE
EMERGENCY FUEL ON
MAIN FUEL REGULATOR FAILED
AFTERBURNER ON AND NOZZLE FAILED CLOSED
AFTERBURNER OFF AND NOZZLE FAILED OPEN
COCKPIT TEMPERATURE MASTER SWITCH
CANOPY AND WINDSHIELD DEFROST LEVER
WINDSHIELD ANTI-ICE
AFTERBURNER ON
NO FUEL DEPLETION
DROP TANK JETTISON
REFUEL DROP TANKS
DROP TANK PRESSURE
MAIN TANK REFUEL
MAIN TANK DUMP
CENTER OF GRAVITY LOCK
LANDING GEAR IN MOTION
SPEED BRAKE DUMP
SPEED BRAKE IN
SPEED BRAKE OUT
UTILITY HYDRAULIC FAIL
DRAG CHUTE DEPLOYED
TRUE AIRSPEED LOCK
ROLL ANGLE LOCK
AUTOPILOT
NOSE WHEEL STEERING
INCREASE ALTITUDE
DECREASE ALTITUDE
CABIN PRESSURE 2 75 P S I
CABIN PRESSURE 5 00 P S I
ALTITUDE LOCK
HYDRAULIC SYSTEM NO 1 FAIL
HYDRAULIC SYSTEM NO 2 FAIL
EMERGENCY HYDRAULIC SYSTEM OPERATING
HYDRAULIC SYSTEM NO 1 TO ANALOG OUTPUT
HYDRAULIC SYSTEM NO 2 TO ANALOG OUTPUT
PITOT ICE
ZERO MODE
FREEZE MODE
YAW DAMPER ON
ROUGH AIR
GUIDE VANE ANTI-ICE

ANALOG INPUTS (9)

RIGHT BRAKE FORCE
LEFT BRAKE FORCE
THROTTLE POSITION
AILERON POSITION
HORIZONTAL STABILIZER POSITION
RUDDER POSITION
AIRPORT ELEVATION
BAROMETRIC PRESSURE SETTING
ICING RATE

DISCRETE OUTPUTS (15)

DROP TANKS FULL
LANDING GEAR DOWN
LANDING GEAR UP
SPEED BRAKE IN
SPEED BRAKE OUT
SPEED BRAKE IN MOTION
DYNAMIC PRESSURE HIGH
STALL WARNING
STALL
LAND-AIR
CRASH
HYDRAULIC SYSTEM NO 1 FAIL
HYDRAULIC SYSTEM NO 2 FAIL
STABILIZER FROZEN
AILERON FROZEN

ANALOG OUTPUTS (38)

DROP TANK FUEL QUANTITY
MAIN TANK FUEL QUANTITY
TAILPIPE TEMPERATURE
INDICATED ALTITUDE (± SIN ± COS)
ALTITUDE ABOVE GROUND (± SIN ± COS)
CABIN ALTITUDE
HYDRAULIC PRESSURE
NORMAL ACCELERATION
INDICATED AIRSPEED
MACH NUMBER
BALL ANGLE
RATE OF CLIMB
GROUND SPEED
TRUE AIRSPEED
PITCH ANGLE
ROLL ANGLE (± SIN ± COS)
TRUE HEADING (± SIN ± COS)
RUDDER HINGE MOMENT
ENGINE R.P.M
TURNING RATE
ROLLING RATE
FUEL FLOW
ANGLE OF ATTACK
ICE QUANTITY
ELEVATOR HINGE MOMENT
PITCHING RATE

Adapted from Sylvania
Final Report FR77-1N

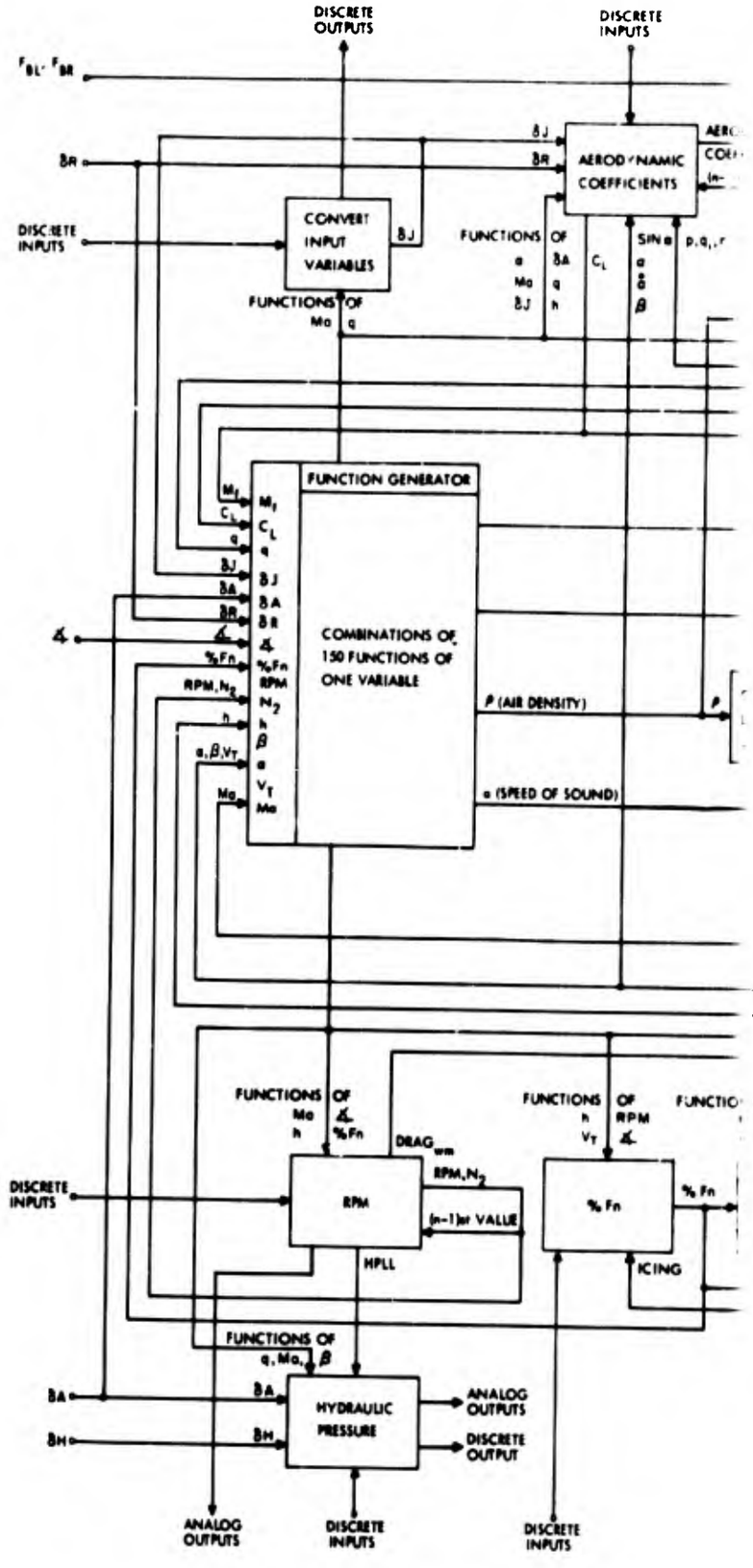


Fig.

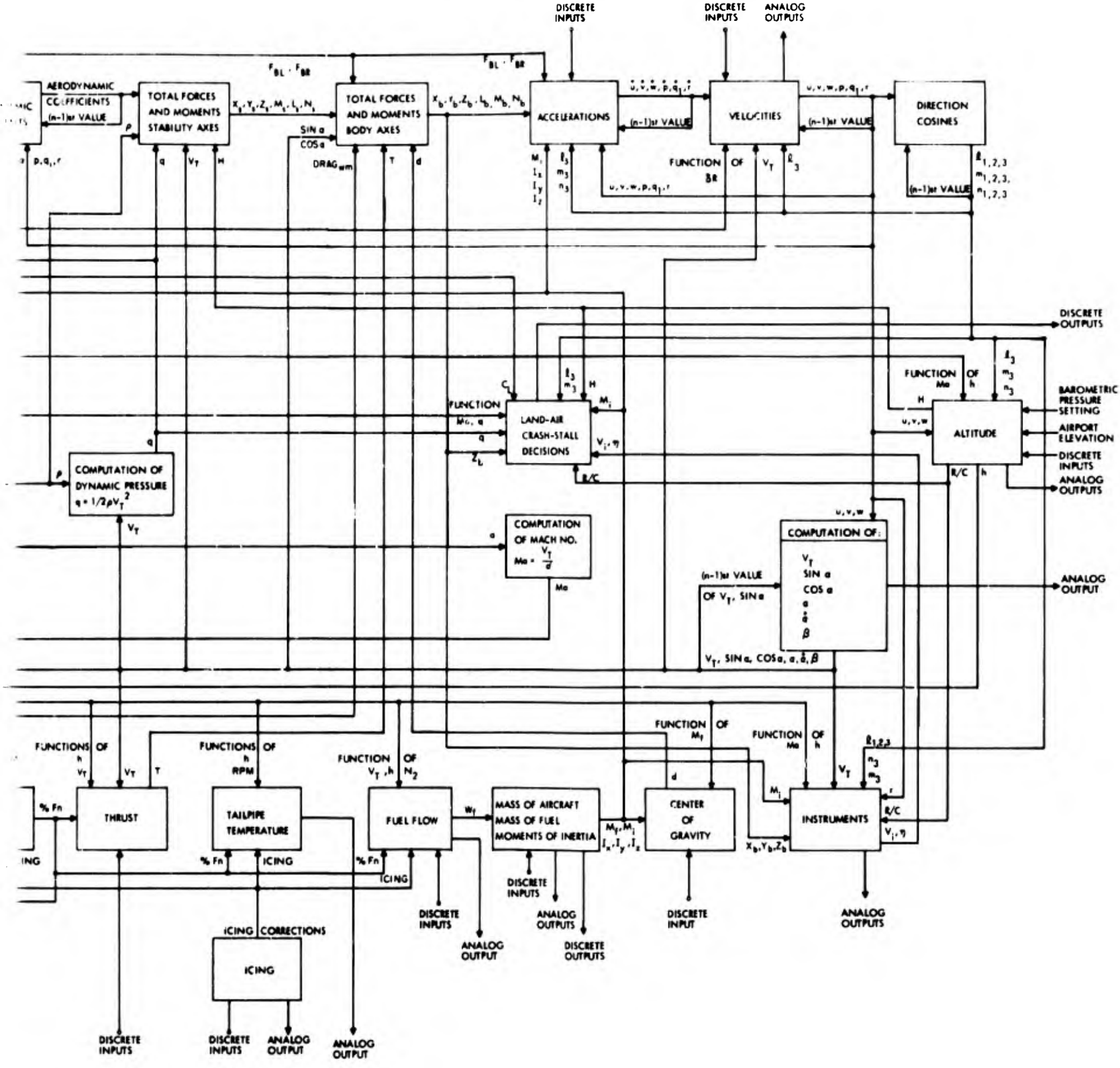


Fig. 2.1 Information Flow in UDOFT F100A Model

B

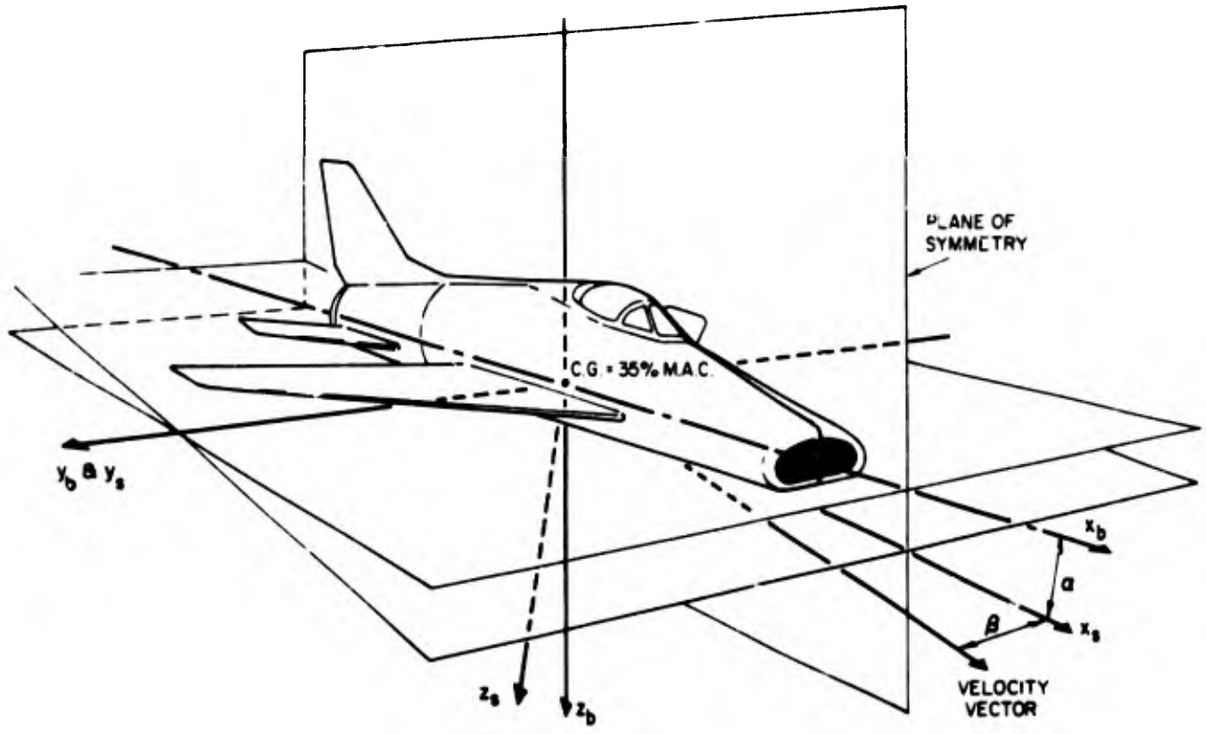


Fig. 2.2 Definition of Body and Stability Axes

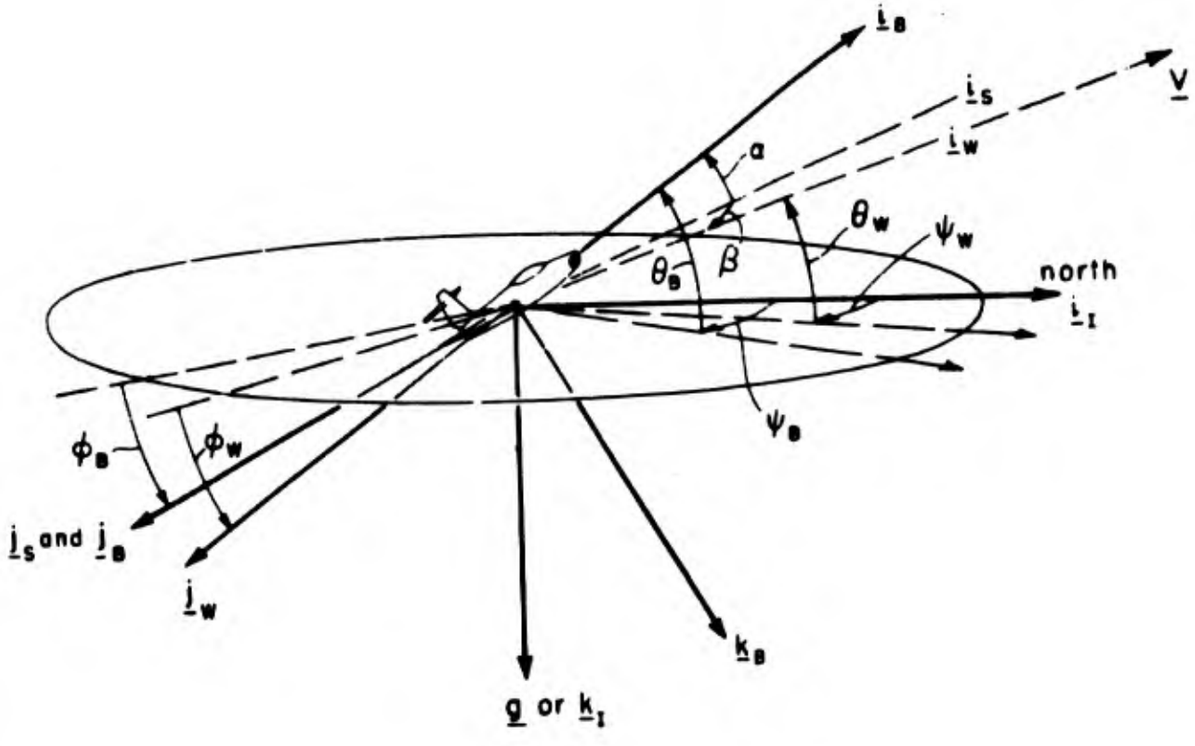


Fig. 2.3 Master Diagram of Euler Angles and Axis Systems Used in Aircraft Simulation

Instead of incorporating a $C_{n(wa)}$ term directly, its effect is added to the $C_{n\beta}$ term. Because of the need for more breakpoints for proper representation, when $\alpha > 0$

$$C_{n_p} = C_{n_p}(|\alpha|)$$

$$C_{\ell_r} = C_{\ell_r}(|\alpha|)$$

but when $\alpha < 0$

$$C_{n_p} = af\,4(|\alpha|)$$

$$C_{\ell_r} = -C_{\ell_r}(|\alpha|)$$

Total Aerodynamic Force along X Wind Axis (-Drag)

$$X_w = -\frac{\rho V^2 S}{2}\left[C_D(|C_L|, Ma) + C_{D(DT)}(Ma) + C_{D(DC)} + C_{D(LG)} + C_{D_{\delta J}}(Ma)\cdot \delta J\right] \qquad (2.4)$$

Total Aerodynamic Force along Y Wind Axis

$$Y_w = \frac{\rho V^2 S}{2}\left[C_{y_\beta}\cdot\beta + C_{y_{\delta R}}(Ma)\cdot BER(Ma, q_1)\cdot \delta R\right] \qquad (2.5)$$

The coefficients $C_{y_p}(\alpha)$ and $C_{y(DT)}$ have been neglected.

Total Aerodynamic Force along Z Wind Axis (-Lift)

$$Z_w = -\frac{\rho V^2 S}{2}\left[C_{L_\alpha}(Ma, h)\cdot\alpha + C_{L_{\delta H}}(Ma, h)\cdot\delta H + C_{L_{\delta J}}(Ma)\cdot\delta J + C_{L(DT)}(\alpha)\cdot f_1(Ma) + C_{L(GE)}\right] \qquad (2.6)$$

where the ground effect term

$$C_{L(GE)} = k(25 - h)\cdot C_L \qquad h \leq 25 \text{ feet}$$

$$C_{L(GE)} = 0 \qquad h > 25 \text{ feet}$$

Total Forces in the X, Y, and Z Wind Axes

To obtain the total force in the wind axes, the effect of thrust, windmill drag, and gravity must be added to the aerodynamic forces. The thrust line is at an angle of 3° below the x body axis. When the aircraft is on the ground, terms must be included for the wheel brake and wheel friction forces.

$$F_{xw} = X_w + (T - \text{Drag WM})\cos\alpha + (T - \text{Drag WM})\sin 3^\circ \sin\alpha - mg\sin\theta_w - (F_{BL} + F_{BR}) - 600 \qquad (2.7)$$

$$F_{yw} = Y_w + mg\cos\theta_w \sin\phi_w = ma_y \qquad (2.8)$$

$$F_{zw} = Z_w + (T - \text{Drag WM})\cdot\sin 3^\circ\cos\alpha - (T - \text{Drag WM})\sin\alpha + mg\cos\theta_w\cos\phi_w = ma_z \qquad (2.9)$$

Total Moments in the X, Y, and Z Body Axes

At touchdown with $h = 0$, the variables θ_w, $\sin\theta_w$, $\dot{h}$, a_y, $\dot{p}$, p, $\dot{\phi}_w$, ϕ_w, and a_z are clamped at zero. F_{zw} is still computed, however, because this force, acting with a moment arm ℓ_{cw} about

the pivot point of the main wheels, causes the nosewheel to settle to the ground. When the nosewheel contacts the ground ($\alpha < 5^o$), an elastic stop term is introduced in the pitch equation to counterbalance the pitch-down moments. On the ground, the moment arm is computed from

$$\ell_{cw} = 1.095 \cos\alpha - 5.93 \sin\alpha \tag{2.10}$$

The total moments in the body axes are obtained by adding the effects of the thrust line passing 1.33 feet above the aircraft center of gravity, the activation of the yaw damper, and the moments exerted by the wheels and brakes when the aircraft is on the ground. Pitch moments due to the migration of the center of gravity are neglected because they are equivalent to small trim changes by the pilot.

$$T_{yb} = M_b - 1.33\,T - F_{zw}\ell_{cw} + \Delta M_{NW} \tag{2.11}$$

where

$$\Delta M_{NW} = 0 \qquad \alpha \geq 5^o$$

$$\Delta M_{NW} = k'''(5^o - \alpha) - k''\dot{\alpha} \qquad \alpha < 5^o$$

$$T_{xb} = L_b \tag{2.12}$$

$$T_{zb} = N_b + 420\,V\dot{\beta} + 6.21\,(F_{BR} - F_{BL}) - 1250\,V \cdot r \tag{2.13}$$

where

$$6.21\,(F_{BR} - F_{BL}) = 0$$

$$1250\,V \cdot r = 0 \qquad \text{if} \quad h > 0$$

The yaw damper term $420\,V\dot{\beta}$ is zero if the yaw damper is off. If the nosewheel is firmly on the ground ($\alpha < 4^o$) and the nosewheel steering is activated, $\dot{r}$ is set to zero and the following relation is used to determine heading:

$$\dot{\psi}_w = r = f_{pr}(\delta R) \cdot V$$

The total forces and moments are then used to find the derivatives $\dot{V}$, $\dot{\alpha}$, $\dot{\beta}$, $\dot{p}$, $\dot{q}$, and $\dot{r}$

$$\begin{aligned}
F_{xw} &= mV \\
F_{yw} &= ma_y = mV\,[\dot{\beta} + r - p\alpha] \\
F_{zw} &= ma_z = mV\,[\dot{\alpha} - q + p\beta] \\
T_{yb} &= I_y\dot{q} + (I_x - I_z)\,rp \\
T_{xb} &= I_x\dot{p} \\
T_{zb} &= I_z\dot{r} + (I_y - I_x)\,pq
\end{aligned} \tag{2.14}$$

In the roll equation, the dynamic term $(I_y - I_z)\,qr$ has been neglected. In the pitch equation, the term $I_{xz}\,p^2$ was eliminated because no information could be located on the value of the product of inertia I_{xz}. The derivatives $\dot{V}$, $\dot{\alpha}$, $\dot{\beta}$, $\dot{p}$, $\dot{q}$, and $\dot{r}$ are integrated to give V, α, β, p, q, and r, except that on the ground the special conditions already specified go into effect.

Wind Euler Angles

The Euler angles describing the orientation of the wind axes are given by:

$$\dot{\psi}_w = \frac{1}{\cos\theta_w}\left[\frac{a_y}{V}\cos\phi_w - \frac{a_z}{V}\sin\phi_w\right] \tag{2.15}$$

$$\dot{\theta}_w = -\left[\frac{a_y}{V}\sin\phi_w + \frac{a_z}{V}\cos\phi_w\right] \tag{2.16}$$

$$\dot{\phi}_w = p + \dot{\psi}_w \sin\theta_w \tag{2.17}$$

Because of the indeterminacy at $\theta_w = \pm 90^\circ$, flight conditions in the M.I.T. F-100A model are limited to $|\theta_w| \leq 45^\circ$. Consequently, in pitch the following series approximation is used in the all-digital program:

$$\sin \theta_w = \theta_w - \frac{\theta_w^3}{6} \tag{2.18}$$

$$\cos \theta_w = 1 - \frac{\theta_w^2}{2} \tag{2.19}$$

Since angle of attack is similarly limited in range, the same approximation is employed to yield

$$\sin \alpha = \alpha - \frac{\alpha^3}{6} \tag{2.20}$$

$$\cos \alpha = 1 - \frac{\alpha^2}{2} \tag{2.21}$$

However, the roll angle ϕ_w and the heading ψ_w have unlimited rotation. Consequently, in the all-digital program, $\sin \phi_w$, $\cos \phi_w$, $\sin \psi_w$, and $\cos \psi_w$ are computed directly from $\dot{\phi}_w$ and $\dot{\psi}_w$ using a numerical algorithm discussed in the chapter on programming.

Miscellaneous

Other variables computed in the aerodynamics model include:

$$\dot{h} = \text{rate of climb} = V \sin \theta_w \tag{2.22}$$

$$h = \text{altitude} = \int \dot{h}\, dt \tag{2.23}$$

$$\frac{a}{a_0} = \left(1 - \frac{h}{35{,}400}\right) .1375 + .8625 \qquad h < 35{,}400 \text{ feet}$$

$$\frac{a}{a_0} = .8625 \qquad h \geq 35{,}400 \text{ feet}$$

where

a = speed of sound at altitude h

a_0 = speed of sound at sea level = 1130 ft/sec.

$$\text{Mach} = \frac{V}{a} \tag{2.24}$$

$$\text{Air Density} = \rho = \text{rho}(h)$$

$$\text{Dynamic Pressure} = q_1 = \frac{1}{2} \rho V^2 \tag{2.25}$$

For obvious economic reasons, no attempt was made in the M.I.T. demonstration to duplicate the actual F-100A cockpit complete with all flight instruments. Since no instruments were provided to display them, the following variables were not computed:

Indicated pressure altitude
Indicated airspeed
Cabin altitude
Ball angle
Indicated Mach
Magnetic heading

In a complete hybrid OFT, these variables would probably be computed in the analog domain, since the cost of the analog equipment required to do so would be less than the cost of the sample gates and sample-and-hold gates required to bring the contributory information into the digital computer and to store the digital results in analog form for activating the cockpit instruments. Indeed, the conversions from true airspeed, true Mach, and true altitude to indicated airspeed, indicated Mach, and indicated pressure altitude are standard relationships that would seem to warrant the development of special electromechanical devices for carrying out these conversions in the simulator cockpit instruments themselves.

Control Forces and Hinge Moments

A similar, but somewhat more complicated, situation arises in the simulation of the control forces required on the stick and rudder pedals and the resultant control surface deflections. In both the all-digital and hybrid simulations, a simple spring-loaded joy stick was employed, mechanically coupled to two potentiometers. The potentiometer output responding to fore-and-aft motions of the stick corresponded to deflections of the horizontal stabilizer (δH). The potentiometer output responding to lateral motion of the stick corresponded to deflections of the aileron surface (δA). Separate knob-operated potentiometers produced voltages corresponding to the rudder surface deflection (δR) and throttle setting (δT).

In the UDOFT F-100A model, the total rudder pedal force (F_{p_T}) was given by

$$F_{p_T} = F_p(\delta PR) + \frac{2}{15}\,\frac{|HM_R|}{HM_R}\,[|HM_R| - 4330] \qquad HM_R \geq 4330$$

$$F_{p_T} = F_p(\delta PR) \qquad HM_R < 4330 \qquad (2.26)$$

$$F_{p_T} = F_p(\delta PR) + \frac{2}{15}\,HM_R \qquad \text{Utility hydraulic failed}$$

where F_p is a nonlinear function of the pedal deflection δPR.

$$HM_R = \text{Rudder Hinge Moment} = 128\,q_1\,C_{h_R} = q_7(q_1)\cdot m_{43}(Ma)\left[b_1(\beta) - 1.28925\,\delta R\left(1 - \frac{|\beta|}{16}\right)\right] \qquad (2.27)$$

The actual rudder surface deflection (δR) is a nonlinear function of δPR. Since variable-load rudder pedals are not included in the M. I. T. System, the force F_{p_T} is not computed. However, the rudder hinge moment (HM_R) is calculated.

In UDOFT, the fore-and-aft (stabilizer) stick force (F_{H_T}) is given by

$$F_{H_T} = F_H(\delta SH) + 2.85(n - 1) \qquad (2.28)$$

where F_H is a nonlinear function of stick fore-and-aft deflection (δSH)

n = load factor in g's

The actual horizontal stabilizer deflection (δH) is a nonlinear function of the stick deflection δSH. Since no provisions are made to vary the stick force in the M. I. T. system, F_{H_T} is not computed.

However, both the stabilizer hinge moment and the aileron hinge moment are needed to test the adequacy of the hydraulic pressure, hence these moments are computed using the following UDOFT relations in which angle of attack (α) and control surface deflections (δH, δA) are in degrees:

$$HM_H = \text{Horizontal Stabilizer Hinge Moment} = 4050\,q_1\,Ch_H \qquad (2.29)$$

$$C_{h_H} = [0.0111\,\alpha\cdot f_{53}(Ma) + 0.025\,\delta H\cdot f_{54}(Ma)]\;935\,f_8(q_1) \qquad (2.30)$$

$$HM_A = \text{Aileron Hinge Moment} = 500\,q_1\,Ch_A \qquad (2.31)$$

$$C_{h_A} = 0.0186\,\alpha\,f_{52}(Ma) + 0.018\;|\delta A|\,f_{51}(Ma) \qquad (2.32)$$

The hydraulic pressure required to provide these hinge moments is given by:

$$P_{H_H} = \frac{\pi}{500}\,H_{M_H} = -\,q_8(q_1)[\alpha\cdot m_{53}(Ma) + \delta H\cdot m_{54}(Ma)] \qquad (2.33)$$

$$P_{H_A} = \frac{\pi}{500}\,H_{M_A} = q_1\cdot[|\delta A|\cdot m_{51}(Ma) + \alpha\cdot m_{52}(Ma)] \qquad (2.34)$$

where, by combining terms and converting degrees to radians, the UDOFT functions $f_{53}(Ma)$, $f_{54}(Ma)$, $f_8(q_1)$, $f_{52}(Ma)$ and $f_{51}(Ma)$ have been replaced by

$$m_{53}(Ma) = 0.0111(57.3)935\,f_{53}(Ma) = 595\,f_{53}(Ma)$$

$$m_{54}(Ma) = 0.025(57.3)935\,f_{54}(Ma) = 1340\,f_{54}(Ma)$$

$$q_8(q_1) = \frac{\pi}{500}(4050)q_1\,f_8(q_1) = 25.44\,q_1\,f_8(q_1) \qquad (2.35)$$

$$m_{52}(Ma) = \pi(0.0186)(57.3)\,f_{52}(Ma)$$

$$m_{51}(Ma) = \pi(0.018)(57.3)\,f_{51}(Ma)$$

2.3 ENGINE EQUATIONS

The major remaining aspects of the M.I.T. F-100A model describe the characteristics of the Pratt and Whitney J57-P-7 Turbo-Wasp engine. Early in the program, preliminary discussions were held with Pratt and Whitney engineers with the intent of converting from the bizarre Melpar-Sylvania engine model to the equations used successfully by Pratt and Whitney in its own design work. Unfortunately, time did not permit this change to a model more suitable for digital and hybrid computation. As a consequence, the equations employed in both the digital and hybrid simulations are identical to the UDOFT formulation.[9,17]

The engine simulation is complicated by the large number of discrete inputs and decisions that influence the sequence of calculations. The overall effect can only be understood by referring to both the equations that follow and the flow charts for the engine program.

RPM (Percent)

$$\text{Normal RPM} = N_2 = t_2(\delta T)[(0.979 - p_{31}(h))m_{50}(Ma) + p_{31}(h)] \tag{2.36}$$

$$\text{Minimum governing RPM} = N_2 = p_{32}(h) \tag{2.37}$$

$$\text{Windmilling RPM} = N_2 = m_{49}(Ma) \cdot p_{33}(h) \tag{2.38}$$

$$\text{Emergency fuel RPM} = N_2 = f_2(pfn)$$

$$\text{Windmilling drag} = D_{wM} = m_{48}(Ma) \cdot p_5(h) \tag{2.39}$$

Percent Thrust (PFN)%/100

Minimum governing RPM

$$pfn = r_2(rpm)$$

Emergency fuel

$$pfn = v_{12}(tas) \cdot p_{27}(h) \cdot t_4(\delta T) + t_5(\delta T) \tag{2.40}$$

Normal

$$pfn = t_6(\delta T) \tag{2.41}$$

Tailpipe Temperature (TPT), °C

$$TPT = TT7 \text{ (lagged)}$$

Normal

$$TT7 = 1024\left[(K - K_1)\left(\frac{pfn - 0.2}{0.8}\right) + K_1\right]$$

$$K_1 = [p_{34}(h) - p_{35}(h)]\frac{V}{1000} + p_{35}(h) \qquad \text{if } 0 \leq V < 1000$$

$$K_1 = p_{34}(h) \qquad \text{if } 1000 \leq V \leq 1100$$

$$\begin{aligned} K &= 0.5 && \text{if nozzle closed} \wedge \text{A/B off} \\ &= 0.4 && \text{if nozzle open} \wedge \text{A/B off} \\ &= 0.9 && \text{if nozzle closed} \wedge \text{A/B on} \\ &= 0.5 && \text{if nozzle open} \wedge \text{A/B on} \end{aligned} \tag{2.42}$$

Starting surge

$$TT7 = r_5(rpm)$$

Thrust (F_n)

$$F_n = F_n' \cdot PFN \tag{2.43}$$

Normal, non-A/B

$$F_n' = [p_{20}(h) - p_{19}(h)] \frac{tas}{600} + p_{19}(h) \qquad 0 \leq tas < 600$$

$$F_n' = [p_{21}(h) - p_{20}(h)] \frac{tas - 600}{400} + p_{20}(h) \qquad 600 \leq tas < 1000 \tag{2.44}$$

$$F_n' = p_{21}(h) \qquad 1000 \leq tas < 1100$$

Limit, non-A/B

$$F_n' = 10{,}000 \left[\frac{v_4(tas) + p_{22}(h)}{1 + p_{22}(h)} \right] \tag{2.45}$$

Normal, A/B

$$F_n' = [p_{24}(h) - p_{23}(h)]\, v_7(tas) + p_{23}(h) \tag{2.46}$$

Limit, A/B

$$F_n' = [p_{26}(h) - p_{25}(h)] v_9(tas) + p_{25}(h) \tag{2.47}$$

Fuel Flow (W_f) lbs/hour

Normal fuel flow, non-A/B

$$W_f = [W_{500} - W_0] \frac{tas}{500} + W_0 \qquad \text{if } 0 \leq tas < 500 \text{ knots}$$

$$W_f = [W_{1000} - W_{500}] \frac{tas - 500}{500} + W_{500} \qquad \text{if } 500 \leq tas < 1000$$

$$W_f = W_{1000} \qquad \text{if } 1000 \leq tas < 1100 \tag{2.48}$$

$$W_0 = [p_{10}(h) - p_9(h)] \left(\frac{1 - pfn}{0.8}\right) + p_9(h) \qquad \%\, F_n > 20\,\%$$

$$W_0 = p_{10}(h) \qquad \%\, F_n \leq 20\,\%$$

$$W_{500} = [p_{12}(h) - p_{11}(h)] \left(\frac{1 - pfn}{0.8}\right) + p_{11}(h) \qquad \%\, F_n > 20\,\%$$

$$W_{500} = [p_{12}(h) \qquad \%\, F_n \leq 20\,\%$$

$$W_{1000} = [p_{14}(h) - p_{13}(h)] \left(\frac{1 - pfn}{0.8}\right) + p_{13}(h) \qquad \%\, F_n > 20\,\%$$

$$W_{1000} = p_{14}(h) \qquad \%F_n \leq 20\,\% \tag{2.49}$$

Limit, Non-A/B

$$W_f = 10{,}000 \left[\frac{v_5(tas) + p_{15}(h)}{1 + p_{15}(h)} \right] \tag{2.50}$$

Normal, A/3

$$W_f = \{[p_{17}(h) - p_{16}(h)] \cdot v_8(tas) + p_{16}(h)\}\, pfn \tag{2.51}$$

Limit, A/B

$$W_f = v_{10}(tas) + p_{18}(h) \tag{2.52}$$

Minimum governing RPM

$$W_f = r_1(rpm)$$

Icing Corrections

Fuel Flow Icing Correction = $0.1702\, Q_i\, W_f$,max 2040 lbs/hour) (2.53)
(added to fuel flow W_f)

where

Q_i = quantity of ice at time $t = \int_0^t S_i\, dt$

S_i = icing or deicing rate (7 fixed values)

Tailpipe Temperature Icing Correction = $0.1710\, Q_i(TPT)$ (max 170°C)
(added to tailpipe temperature TPT)

Thrust Icing Correction = $1 - 0.485\, Q_i$ (max 52 percent)
(thrust and limit thrust multiplied by this factor)

Mass of Fuel

The drop tanks can be refueled (discrete input) at the rate of 237 lbs/sec and fuel is transferred (discrete input: drop tank pressure) from the drop tanks to the main tanks at the rate of 2.78 lbs/sec.

$$M'_{ef} = \text{weight of drop tank fuel} = \int_0^t (\Delta_1 - \Delta_2)\, dt \quad \text{drop tank jettison off} \tag{2.54}$$

$$= 0 \quad \text{drop tank jettison on}$$

Δ_1 = 237 lbs/sec (refuel drop tanks on)

= 0 (refuel drop tanks off)

Δ_2 = 2.78 lbs/sec (drop tank pressure on)

= 0 (drop tank pressure off)

The main tanks can be refueled or dumped (discrete inputs) at the rate of 80 lbs/sec and the depletion rate is calculated in the fuel flow routine (W_f).

$$M'_f = \text{weight of main tank fuel} = \int_0^t (-W_f + \Delta_2 + \Delta_3)\, dt \tag{2.55}$$

Δ_3 = 80 lbs/sec (main tank refuel on)

= -80 lbs/sec (main tank dump on)

= 0 (otherwise)

Mass and Moments of Inertia

Converting the fuel weights (pounds) to fuel mass (slugs), the total instantaneous aircraft mass is given by:

$$M_i = M_0 + M_f + M_{ef} + M_{at}$$

M_0 = mass of empty aircraft = 591 slugs

M_f = mass of internal fuel (max 153 slugs) (2.56)

M_{ef} = mass of external fuel (max 111 slugs)

M_{dt} = mass of drop tanks (12.4 slugs)

The moments of inertia are given by:

$$I_x = 10,200 + 100\, M_{ef} \quad \text{slug-ft}^2 \tag{2.57}$$

$$I_y = 57,000 \quad \text{slug-ft}^2 \tag{2.58}$$

$$I_z = 60,000 + 75\,(M_f + M_{ef}) \quad \text{slug-ft}^2 \tag{2.59}$$

Hydraulics

The F-100's two independent flight control hydraulic systems are simulated, as well as the emergency source of hydraulic pressure that works in conjunction with the No. 1 system. The two main systems have independent control valves and activating cylinders, hence in normal operation each supplies half the load required by the control surfaces.

$G2_n$ = hydraulic pressure No. 2 system (time step n) = $G2_{n-1} + \Delta P - (K_2 + 2.6)\,\Delta S$

= $hp\ell$ (if $G2_n < 600$ psi and system No. 2 has not failed (max 3000 psi)

ΔP = pressure increment = 0.1 $hp\ell$

= 0 (system No. 2 failed)

ΔS = stick movement = $-\,|\delta H_n - \delta H_{n-1}| - |\delta A_n - \delta A_{n-1}|$

K_2 = [No. 2 fail + 1/8 (No. 1 fail)] ΔP Fail = 1, Not Fail = 0

$\frac{hp\ell}{(rpm)}$ = lower limit of hydraulic pressure (if system has not failed) (2.60)

$G1_n$ = hydraulic pressure No. 1 system (time step n) = $G1_{n-1} + \Delta p' - (K_1 + 2.6)\,\Delta S$ (max 3300 psi)

$\Delta P' = \Delta P$ No. 1 system only

$\Delta P' = \Delta P - 5$ emergency only

$\Delta P' = \Delta P + 5$ emergency and system No. 1

ΔP = 0.1 $hp\ell$

= 0 (system No. 1 failed)

K_1 = [No. 1 fail + $\frac{1}{8}$ (No. 2 fail)] ΔP Fail = 1, Not Fail = 0 (2.61)

The emergency system can be activated by the pilot (discrete input) or automatically if the rpm drops below 40 percent. It will not operate unless the dynamic pressure exceeds 7 lbs/ft^2.

The hydraulic pressure demanded by the aileron and stabilizer hinge moments computed previously is then compared with the available hydraulic pressure. If the demand is greater than the capacity, the stick would be frozen in a fully-instrumented OFT cockpit.

if $G1_n + G2_n < P_{HA}$ δA frozen (discrete output)

if $G1_n + G2_n < P_{HH}$ δH frozen (discrete output)

2.4 DIGITAL FLOW CHARTS

By themselves, the aerodynamic and engine equations just presented offer an incomplete picture of the total F-100A model. The actual set of calculations performed at any instant is a function of numerous decisions and discrete inputs that reflect failures; emergency conditions; the effects of landing gear, speed brakes, drop tanks, yaw damper, afterburner, and nosewheel steering; ground, air, and transition conditions; and even improper sequences of operating steps, as in the case of the engine starting procedure. In addition, many variables are automatically set to their saturation level by the program whenever this level is exceeded.

To describe these aspects of the simulation model, the engine and aerodynamic equations must be supplemented by the flow charts used in the hybrid and all-digital programs. These flow charts are made available in Appendix B and Appendix C.

2.5 SUMMARY OF F-100A MODEL

A reasonably comprehensive description of the F-100A model used in the M.I.T. simulation studies has been presented for the following reasons:

a) To provide a background for understanding the hyprid system design and the programs discussed in Chapters III and IV.

b) To indicate the preliminary steps that must be carried out in model making, data analysis, and flow charting before a successful hybrid or digital simulation can be undertaken.

c) To make available in quantitative terms a measure of the size of the real-time problem involved in the M.I.T. demonstrations. It is hoped that this measure, in conjunction with the data on memory consumption and computation times presented in Chapter IV, will be useful in estimating the computational requirements imposed by future real-time simulation problems of a similar category.

In summary, the demonstration is based on a six-degree-of-freedom simulation of an F-100A aircraft over the complete range of operating flight conditions from ground taxiing to service ceiling. Included in the model are the engine static and dynamic characteristics, fuel consumption, significant aerodynamic nonlinearities corrected for aeroelastic effects where necessary, atmospheric changes, numerous malfunctions, and pilot inputs, both discrete and continuous. Broken down, the model involves:

73 functions of one variable

13 functions of two variables

13 independent variables on which nonlinear functions depend (pfn, V, δT, rpm, $|\beta|$, h, Ma, C_L, $|C_L|$, α, $|\alpha|$, δR, q_1)

50 discrete inputs

15 discrete outputs

9 analog inputs (all-digital program)

30 analog outputs (all-digital program)

15 integrations ($\dot{V}$, $\dot{h}$, $\dot{\theta}_w$, $\dot{\alpha}$, $\dot{\beta}$, $\dot{p}$, $\dot{q}$, $\dot{r}$, $\sin \dot{\phi}_w$, $\cos \dot{\phi}_w$, $\sin \dot{\psi}_w$, $\cos \dot{\psi}_w$, drop tank fuel depletion, main tank fuel depletion, icing rate)

CHAPTER III

THE HYBRID SYSTEM DESIGN

3.1 INTRODUCTION

In practically all complex dynamic simulation problems, certain basic operations must be performed regardless of whether all-digital or hybrid computing techniques are employed. For example, an all-digital simulator must generally provide at least one analog-to-digital conversion channel with multiplexed inputs in order to introduce manual analog control signals into the digital calculations. If time skew is to be avoided, each multiplexer input must be preceded by a sample-and-hold gate so that all analog samples correspond to the same instant of time. One or more digital-to-analog conversion channels must be provided to supply analog signals for meters, displays, and plotters. Digital registers with individual converters or sample-and-hold gates are needed to store such information for use in the analog domain. Means are required to introduce discrete inputs into the digital calculations, and to make discrete outputs available for driving indicators and relays, and for controlling events in the analog domain. Finally, the all-digital simulation requires some measure or control of time intervals, since digital integration is based on quadrature formulas in which the time increment is an important factor.

It can be seen from the discussion above that the addition of analog computing equipment to the digital simulator does not complicate the basic functional interface requirements in any significant way. The analog-digital conversion requisites remain essentially the same, and analog mode control and gate activation control are merely forms of discrete output. Indeed, the employment of analog integrators eliminates the need for a digital clock to measure or control time increments. In fact, if the M.I.T. hybrid simulation of the F-100A may be taken as a typical case, the introduction of analog equipment results in a net decrease in the complexity of the interface, particularly in the number of analog storage devices required.

A simplified block diagram of the complete M.I.T. hybrid demonstration computer is shown in Fig. 3.1. The principal computing element is the PDP-1, a general-purpose digital computer manufactured by the Digital Equipment Corporation. The M.I.T. PDP-1 has a 4096-word core memory supplemented by a magnetic-drum memory with a capacity of 22 fields of 4096 words each. The core memory cycle time is five microseconds, hence two-cycle instructions such as add and subtract take 10 microseconds. Multiply commands are executed in 14 to 25 microseconds; divides require 30 to 40 microseconds. The PDP-1 word length is 18 bits with data words represented in the one's complement system.

Optional features installed on the Department of Electrical Engineering machine are a DEC Type 30 CRT display (1024 x 1024 points), for which the plotting rate is 20,000 points per second, and an auxiliary Type 32 light pen. A single-channel sequence break function is also included. The order code for the PDP-1 is listed in Appendix A. Standard inputs to the computer are via a paper tape reader and on-line typewriter; standard outputs are via a paper tape punch and typewriter print-out.

In the hybrid system, the digital computer is employed to calculate the engine equations (thrust, fuel flow, tailpipe temperature, RPM, hydraulics, aircraft mass, icing, etc.), to generate nonlinear functions by interpolation between discrete points stored in memory, and to manage decisions (branching operations, saturation limits, etc.).

The analog sub-system is employed to represent, on a continuous basis, a linear, skeleton version of the aircraft (Force equations, Moment equations, Euler angles, Altitude and Navigation coordinates). Corrections for nonlinear characteristics are generated in the digital and pulsed analog equipment and inserted in this analog skeleton to give a net effect corresponding to the nonlinear aircraft.

The pulsed analog sub-system, utilizing wide-bandwidth analog elements which can be time-shared by the use of high-speed sample and sample-and-hold gates operating under program control, is employed to compute the running sums of products that constitute the nonlinear correction signals mentioned above.

A detailed description of the analog sub-system, the pulsed analog sub-system, and the interface logic necessary to implement the discrete input-outputs and the control signals for the gates and the encoder is presented in the sections that follow.

3.2 ANALOG SUB-SYSTEM

The complete block diagram of the analog sub-system is given in Fig. 3.2. The solid-state computer employed in this skeleton representation of the aircraft was designed and built at M.I.T. A detailed description of individual circuit elements, including the novel photoresistive multiplier, is presented in Chapter V.

The block diagram is largely self-explanatory. Coefficient pots are used to implement the constant values of the aerodynamic coefficients and other parameters which constitute the skeleton. The method suggested by Howe and Gilbert[58] is employed for the generation of the sines and cosines of the unlimited Euler angles ϕ and ψ, but since the elevation angle θ is restricted to $\pm 45^{\circ}$, $\sin\theta$ is approximated by θ and $\cos\theta$ is generated in a simple diode function generator. At specified stages in landing and takeoff, various ground effects are switched in or out of the analog configuration under manual or program control. These are:

Deceleration due to ground friction and foot brakes

Heading rate due to differential foot brake action and nosewheel steering

Pitch effect of a_z with main wheel as pivot

Variables clamped at zero $(\frac{a_z}{V}, \frac{a_y}{V})$

Integrator outputs reset to initial condition zero $(h, \beta, \theta, r, p, \sin\phi, x_e, x_n)$

Integrator outputs reset to initial condition machine one $(\cos\phi)$

In addition, the velocity integrator is limited to positive values of velocity output by a diode clamp at all times. The yaw damper effect may be switched in or out manually.

One noteworthy feature of the analog sub-system that warrants further discussion is the use of binary scaling. Powers of two are employed for scaling in the PDP-1 program, of course, because scaling adjustments then require only 5 microsecond shift instructions instead of 25 microsecond multiply instructions. In a hybrid system, binary scaling is also recommended for the analog calculations to simplify the scaling problem on data transferred between the analog and digital domains.

The binary scaling procedure used in the M.I.T. hybrid system is straightforward. Given the largest possible value of a variable, the power of two which is just larger than this maximum is the value assigned to machine one. Physically, machine one is a nominal 10 volts in the analog domain. For example, in scaling aircraft velocity V:

V_{max} = 1520 ft/sec

Next higher power of two = 2^{11} = 2048

$\bar{V}$ = velocity in machine units, the maximum value of which is unity

$$V = 2^{11} \cdot \bar{V} = 2048\,\bar{V}$$

Thus, when the analog voltage representing $\bar{V}$ is at 10 volts, $\bar{V}$ will have a value of unity in machine units, and the corresponding value of airspeed V is 2048 ft/sec.

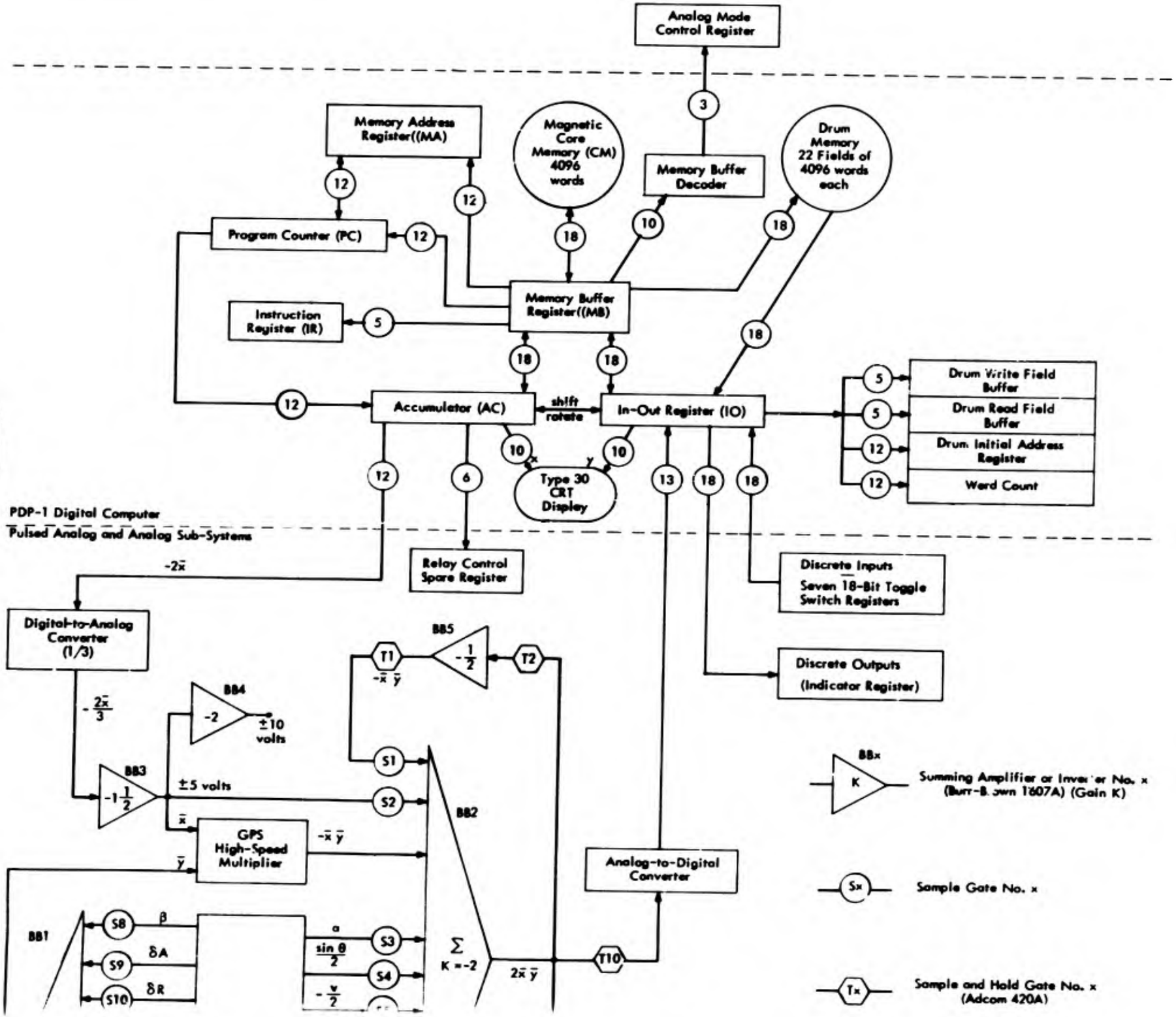
As a further illustration, the simplified X-force equations below will be scaled:

$$\dot{V} = -g\sin\theta + \frac{T}{m} - \frac{\rho V^2}{2}\,\frac{S}{m}\,C_D \qquad (3.1)$$

$$V = \int \dot{V}\,dt \qquad (3.2)$$

Before deriving the equivalent equations which can be implemented directly on the analog computer, the maximum values of all variables must be known or estimated:

A


Analog Mode Control Register
3
Memory Address Register((MA)
Magnetic Core Memory (CM) 4096 words
Memory Buffer Decoder
Drum Memory 22 Fields of 4096 words each
12
12
18
10
18
Program Counter (PC)
12
Memory Buffer Register((MB)
Instruction Register (IR)
5
18
18
18
12
Accumulator (AC)
shift rotate
In-Out Register (IO)
5
Drum Write Field Buffer
5
Drum Read Field Buffer
12
Drum Initial Address Register
12
Word Count
10
10
12
6
Type 30 CRT Display
13
18
18
PDP-1 Digital Computer
Pulsed Analog and Analog Sub-Systems
Relay Control Spare Register
-2x̄
Discrete Inputs
Seven 18-Bit Toggle Switch Registers
Digital-to-Analog Converter (1/3)
BB5
T1
T2
Discrete Outputs
(Indicator Register)
BB4
BB3
±10 volts
±5 volts
S1
S2
BB2
GPS High-Speed Multiplier
Analog-to-Digital Converter
BB1
S8
S9
S10
S3
S4
T10
BBx
Summing Amplifier or Inverter No. x (Burr-Brown 1607A) (Gain K)
Sx
Sample Gate No. x
Tx
Sample and Hold Gate No. x (Adcom 420A)

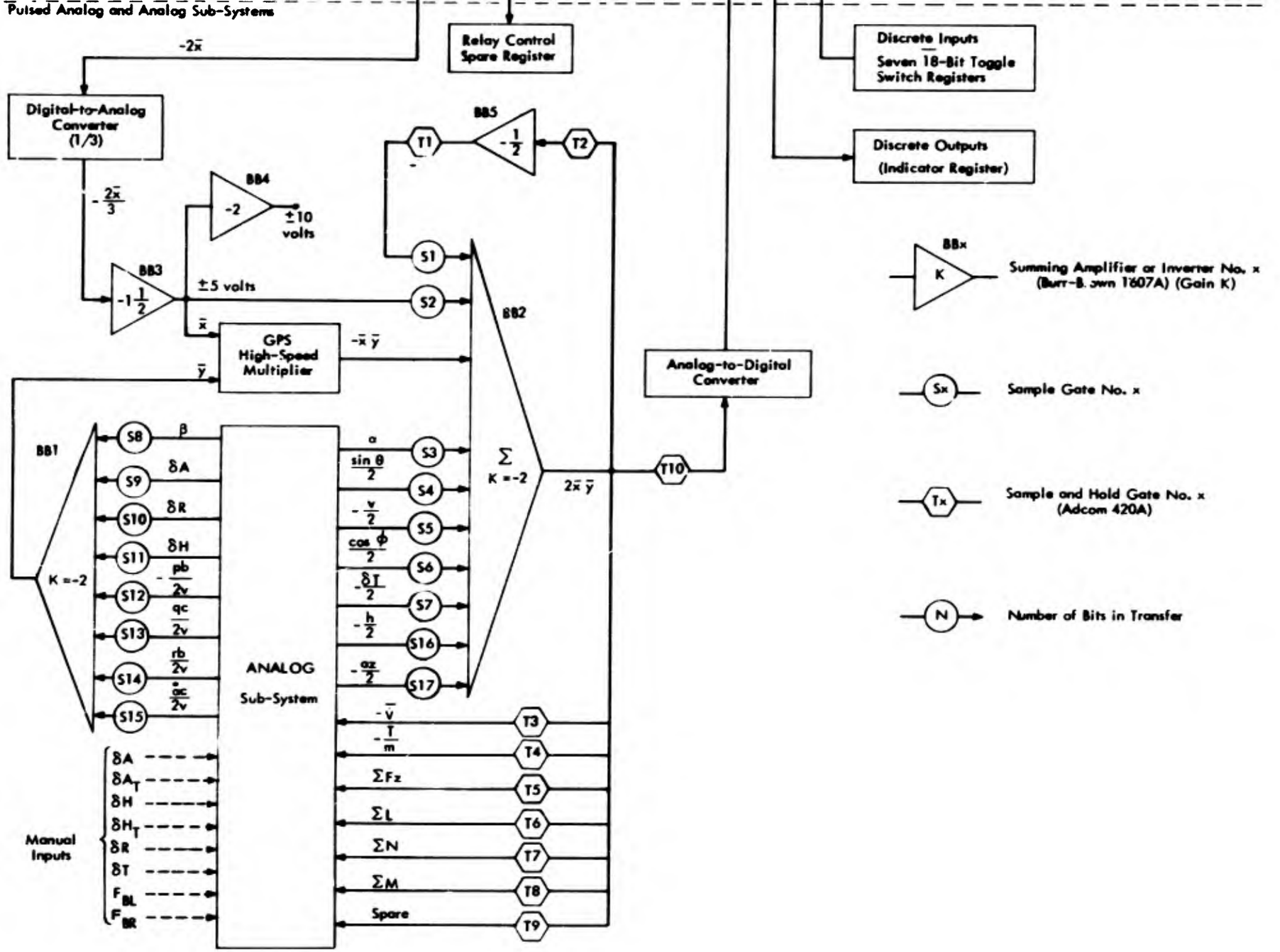


Fig. 3.1 MIT-Hybrid Demonstation Computer

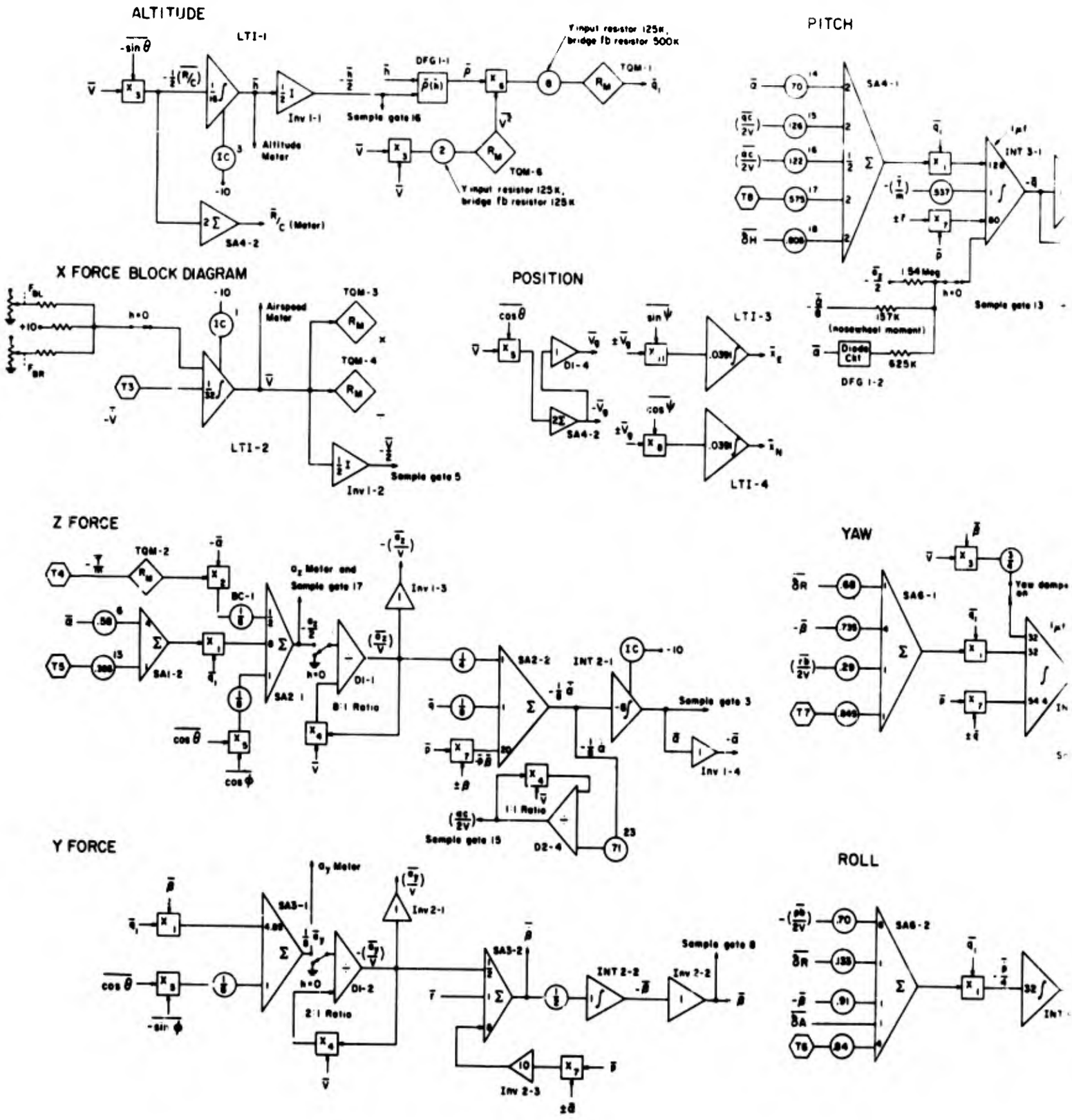


Fig. 3.2 Analog S

A

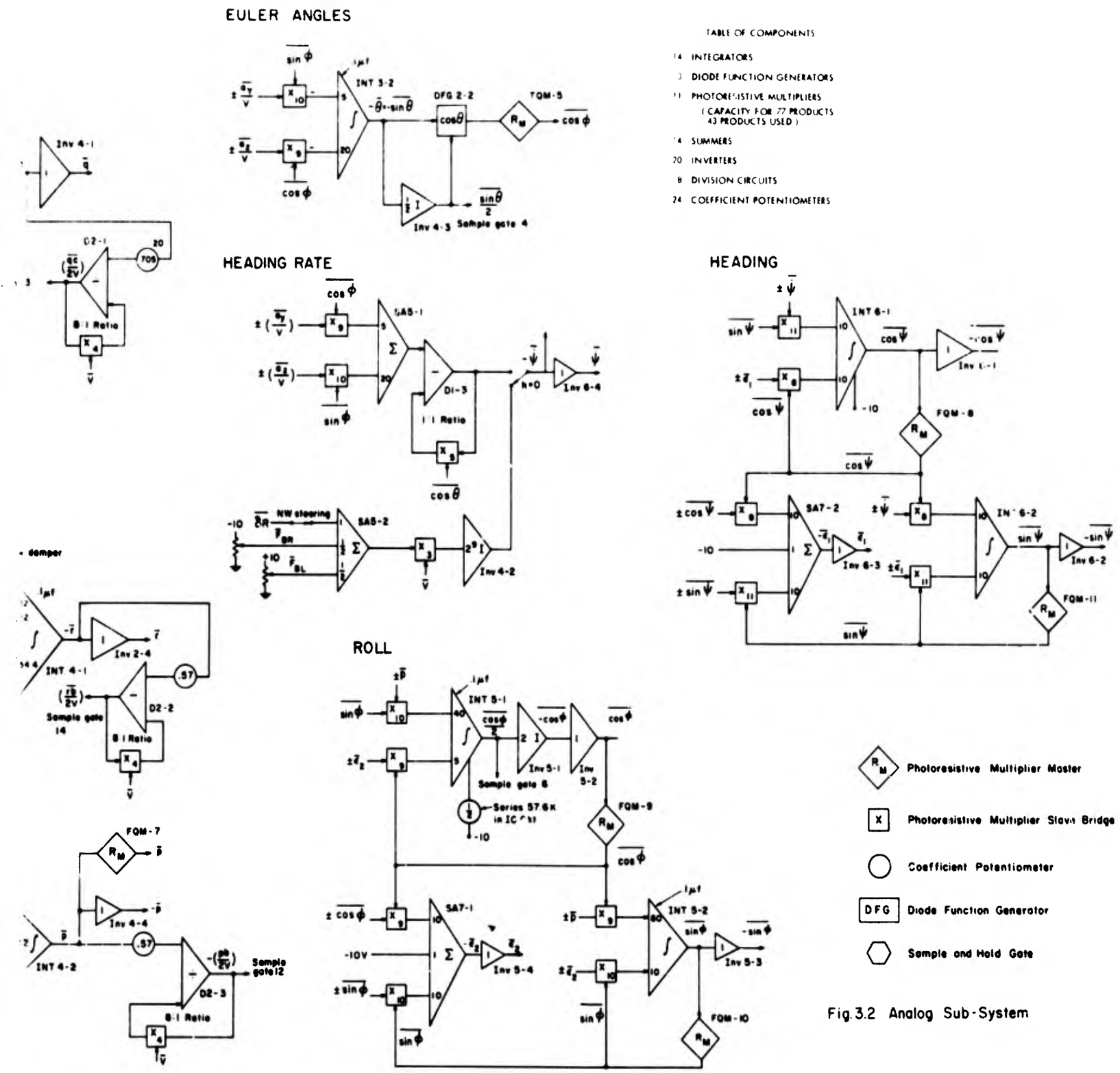


Fig.3.2 Analog Sub-System

B

Table 3.1

Illustrative Analog Scaling

Variable	Maximum Value	Next highest Power of Two	Equivalent Term
V	1520 ft/sec	$2^{11} = 2048$	$2^{11} \cdot \overline{V}$
$\dot{V}$	40 ft/sec^2	$2^6 = 64$	$2^6 \cdot \overline{\dot{V}}$
$\sin\theta$	1	$2^0 = 1$	$\overline{\sin\theta}$
T/m	25.4 ft/sec^2	$2^5 = 32$	$2^5 \cdot \overline{(T/m)}$
$1/2\ \rho V^2 = q_1$	1600 lbs/ft^2	$2^{11} = 2048$	$2^{11} \cdot \overline{q_1}$

For the purposes of this illustration, the following constant parameters may be assumed:

S = Wing Area = 376 ft^2

m = Aircraft Mass = 745 slugs

C_D = Drag Coefficient = 0.017

g = Acceleration of Gravity = 32 ft/sec^2 = 2^5

Substituting equivalent terms in the original equations, one obtains:

$$2^6 \cdot \overline{\dot{V}} = -2^5\,\overline{\sin\theta} + 2^5\,\overline{(T/m)} - 2^{11}\overline{q_1}\,(0.00855) \tag{3.3}$$

$$2^{11}\,\overline{V} = 2^6 \int \overline{\dot{V}}\,dt \tag{3.4}$$

The last relation may be restated as

$$\overline{V} = 2^{-5} \int \overline{\dot{V}}\,dt \tag{3.5}$$

Hence, if we wish the output to be $\overline{V}$, the input to a gain of one integrator ($1/RC = 1$) must be $-2^{-5} \cdot \overline{\dot{V}}$. From Eq. 3.3, the integrator input is given by:

$$-2^{-5}\,\overline{\dot{V}} = 2^{-6}\,\overline{\sin\theta} - 2^{-6}\,\overline{(T/m)} + 0.0085\,\overline{q_1} \tag{3.6}$$

The analog implementation of Eqs. 3.5 and 3.6 would be as shown in Fig. 3.3.

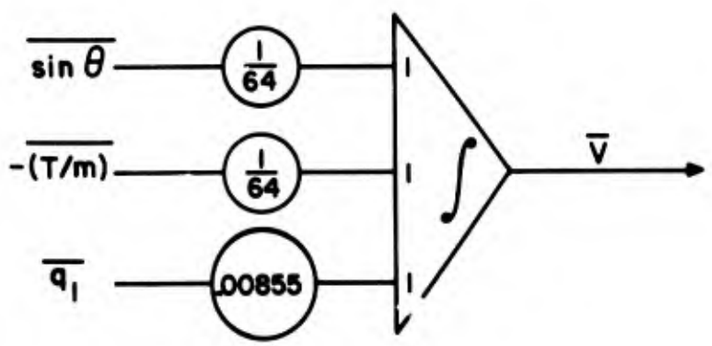


Fig. 3.3 Illustrative Implementation of Simplified X-Force Equation

As originally conceived, the analog skeleton airplane, given engine thrust as an input, was capable of 6-degree-of-freedom flight independent of the digital computer. When the hybrid system was first assembled, the thrust-drag-gravity summation was carried out in the analog domain, much in the manner indicated in the simplified illustration above. However, it was quickly discovered that acceptable steady-state performance fidelity was difficult to obtain with this configuration because the X-force summation, which is the most important factor in performance under normal flight conditions, was being computed at low analog voltage levels. At these low levels, small absolute errors had a serious percentage effect.

Returning to the illustrative example, a typical flight situation will clearly demonstrate the problem. For level flight at Mach 0.9 and altitude 25,000 ft, the following machine variables result:

$$q_1 = 444 \text{ lbs/ft}^2 \qquad \bar{q}_1 = 0.216 \text{ (2.16 volts)}$$

$$\sin\theta = 0 \qquad \overline{\sin\theta} = 0$$

$$T/m = 3.79 \text{ ft/sec}^2 \qquad (T/m) = (0.118 \text{ volts})$$

Under these conditions, the signal inputs to the velocity integrator in Fig. 3.3 are

$$2^{-6}\,\overline{\sin\theta} = 0$$

$$2^{-6}\,(\overline{T/m}) = 0.00184 \text{ (0.0184 volts)}$$

$$0.00855\,\bar{q}_1 = 0.00184 \text{ (0.0184 volts)}$$

The most important performance calculation was thus being carried out with signals in the low millivolt range and acceptable fidelity could not be achieved.

The problem was overcome by two changes incorporated in the present hybrid configuration shown in Fig. 3.2:

(a) The X-force summation (thrust-drag-gravity) was transferred to the digital domain with the resultant $\dot{V}$ decoded and stored at the input to the velocity integrator.

(b) The one volt/sec/volt integrator for velocity was replaced by a long-term integrator with a 1/32 volt/sec/volt gain. This increased the voltage levels at the input and reduced drift.

One minor drawback to these changes, however, was that with the X-force summation carried out in the PDP-1, the analog skeleton airplane could no longer fly independently of the digital computer.

Not included in the analog sub-system block diagram are the pilot control station outputs corresponding to the stick movements (δA, δH), the rudder movements (δR), the throttle position (δT), and the aileron and horizontal stabilizer trim settings (δA_t, δH_t). Four operational amplifiers mounted on a modified quad inverter card are used as buffer stages for these control signals. The δH and δH_t buffer amplifier incorporates a diode limiting circuit that simulates the non-symmetrical characteristics of the F-100A horizontal stabilizer. Also not included in Fig. 3.2 are the six panel meters used to monitor airspeed, rate of climb, altitude, normal acceleration (a_z), lateral acceleration (a_y), and one variable of choice. These units with their series calibration resistors are mounted on the toggle switch console. The meters indicating rate of climb and altitude can be manually switched to expanded-scale operation when these variables approach zero.

3.3 CORRECTING THE ANALOG SKELETON FOR NONLINEAR EFFECTS

The most distinctive feature of the M.I.T. hybrid design is the manner in which the advantages of continuous analog computation are retained, yet the generation of nonlinear functions is accomplished in the digital computer based on data stored in the digital memory. To illustrate this technique, the roll equation may be taken as an example. From Eqs. 2.1 and 2.2

$$\dot{p} = \frac{\rho V^2 Sb}{2I_x}\left[C_{\ell_{\delta A}}(\alpha, M_A)\cdot AEA(M_A, q_1)\cdot \delta A + C_{\ell_{\delta R}}(|\alpha|)\cdot f_2(M_A)\cdot AER(M_A, q_1)\cdot \delta R\right.$$

$$\left. + C_{\ell_\beta}(M_A)\cdot f_3(\alpha)\cdot \beta + C_{\ell_p}(|\alpha|)\cdot AEP(M_A, q_1)\frac{pb}{2V} + C_{\ell_r}(|\alpha|)\cdot \frac{rb}{2V}\right] \qquad (3.7)$$

Each nonlinear aerodynamic function in the roll equation is represented in the continuous analog domain by a constant coefficient corresponding to the most common value assumed by that function in normal flight. The true value of the nonlinear function is computed in the digital domain and the difference between this and the approximate value is inserted into the continuous analog computation on a sampled basis. The net effect is that of using the true value of the function in the computation. In practice, the sum of all such corrective terms for the roll equation may be stored on a single high-speed storage gate.

To be specific, let us examine a single nonlinear function in the roll equation, namely $SbC_{\ell_\beta}(M_A) \cdot f_3(\alpha)/I_x$. In the analog skeleton, this term is approximated by a constant, typical value $SbC_{\ell_{\beta_o}}/I_{x_o}$. The difference between the true value and this approximate value

$$Sb\left(\frac{C_{\ell_\beta}(M_A) \cdot f_3(\alpha)}{I_x} - \frac{C_{\ell_{\beta_o}}}{I_{x_o}}\right)$$

is computed in the digital domain. This corrective term is decoded, multiplied by the analog variable β in the pulsed analog equipment, and the result stored on a sample-and-hold gate as shown in Fig. 3.4. The net effect at the output of the summing amplifier corresponds to the true nonlinear aerodynamic function:

$$\underbrace{Sb\left(\frac{C_{\ell_\beta}(M_A) \cdot f_3(\alpha)}{I_x} - \frac{C_{\ell_{\beta_o}}}{I_{x_o}}\right)\beta}_{\text{Sampled Correction}} + \underbrace{\frac{SbC_{\ell_{\beta_o}}}{I_{x_o}}\beta}_{\substack{\text{Continuous} \\ \text{Term}}}$$

$$= \underbrace{\frac{SbC_{\ell_\beta}(M_A) \cdot f_3(\alpha)}{I_x}\beta}_{\text{Net Effect}} \qquad (3.8)$$

The choice of values for $C_{\ell_{\beta_o}}$ and I_{x_o} should be such that under the most common flight condition, say cruising, the stored corrective term is near zero. That is:

$$\frac{C_{\ell_\beta}(M_A) \cdot f_3(\alpha)}{I_x} \approx \frac{C_{\ell_{\beta_o}}}{I_{x_o}} \qquad (3.9)$$

The extension of the technique to all the terms of the roll equation is obvious. Correction factors are computed in the PDP-1 for $C_{\ell_{\delta A}}$, $C_{\ell_{\delta R}}$, C_{ℓ_β}, C_{ℓ_p}. These factors are multiplied by the appropriate analog variables in the pulsed analog equipment and the running sum of products is stored on sample-and-hold gate T6. These pulsed analog operations will be described in detail in Section 3.5. In the continuous analog paths, typical constant values of these coefficients are used, producing the situation shown in Fig. 3.5. Note that C_{ℓ_r}, being a second-order term, is not implemented in the continuous analog configuration, but is merely added to the running sum of corrective terms.

As shown in the analog sub-system block diagram Fig. 3.2, four basic analog calculations (Z-force, roll, pitch, and yaw) each require one such storage gate for the insertion of corrective terms on a sampled basis. Under typical flight conditions, all of these corrective terms should be near zero. The X-force summation, as explained previously, is carried out completely in the PDP-1 and the resultant derivative $\dot{V}$ is stored at the input to the velocity integrator. In addition, the principal output of the digital engine calculations (T/m) must also be stored as an input to the analog sub-system.

The Y-force calculation was simplified by eliminating the $C_{y_{\delta R}}(M_A) \cdot BER(M_A, q_1) \cdot \delta R$ term, which is small relative to the $C_{y_\beta} \cdot \beta$ term. The resultant equation then involves no function generation and it is implemented completely in the analog domain.

It will be noted by observant readers that some of the corrective terms calculated in the digital-pulsed analog domain vary rapidly and, therefore, theoretically, should be updated at a fast solution rate. The functions dependent on angle of attack and the multiplications of corrective terms by δA, δH, δR, $\frac{pb}{2V}$, $\frac{rb}{2V}$, $\frac{\delta c}{2V}$, $\frac{qc}{2V}$ and β are in this category. Other terms are slowly-varying and obviously need only be updated at a reduced solution rate. The aeroelastic corrections,

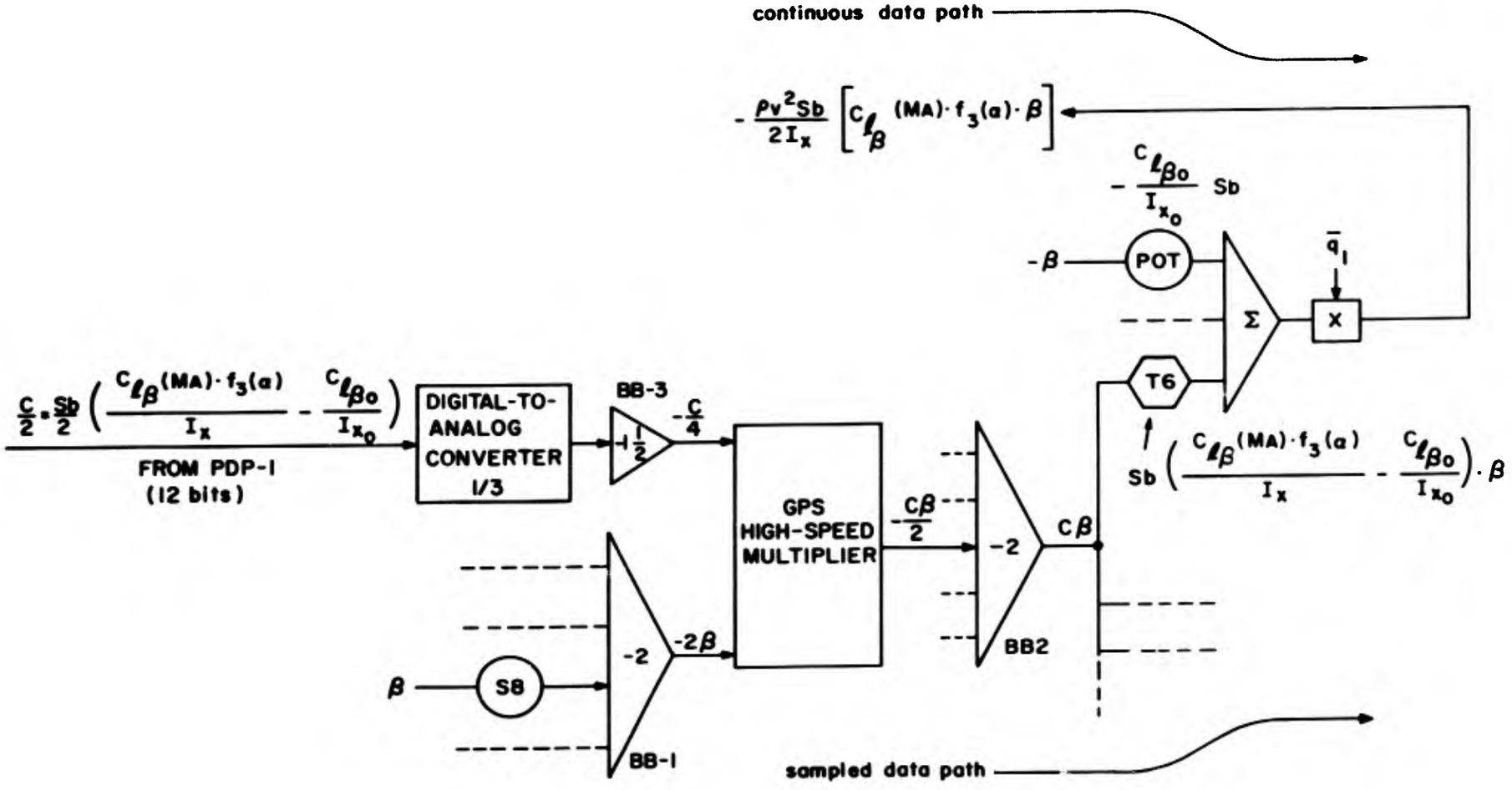
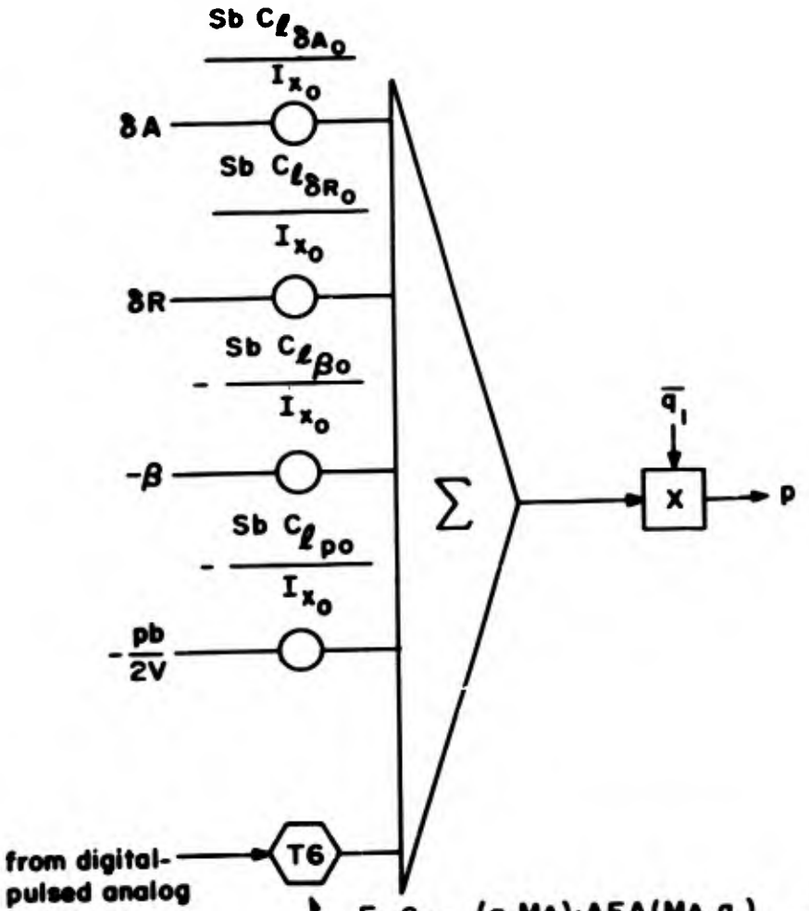


Fig. 3.4 Correction of Constant C_{ℓ_β} Coefficient for Nonlinear Effects



$$+\left(\frac{C_{\ell\beta}(MA)\cdot f_3(\alpha)}{I_x}-\frac{C_{\ell\beta o}}{I_{x_o}}\right)\cdot\beta+\left(\frac{C_{\ell p}(|\alpha|)\cdot AEP(MA,q_1)}{I_x}-\frac{C_{\ell po}}{I_{x_o}}\right)\cdot\frac{pb}{2V}+\left(\frac{C_{\ell_r}(|\alpha|)}{I_x}\right)\cdot\frac{rb}{2V}\Bigg]$$

Fig. 3.5 Correction of All Constant Coefficients in Roll for Nonlinear Effects

the thrust, the aircraft mass, the moment of inertia, I_x, and the Mach dependence of the various nonlinear functions are in this category.

In practice, the expected fast-solution rate requirement on the high-frequency corrective terms turns out to be unnecessary. First, transitions from one steady-state flight condition to another are relatively slow and can be handled at reduced solution rates. Steady-state performance is the only aspect of the simulation requiring close accuracies. Second, once a steady-state condition is established, true values of all the dynamic aerodynamic coefficients corresponding to that condition are in effect within one solution cycle at most. Thirdly, the continuous, skeleton representation of all the basic dynamic loops produces acceptable transient results even though the sampling rate at which nonlinear corrections are introduced might seem to be low relative to the natural frequencies involved. Bear in mind that the tolerances on transient reproduction are substantially less demanding than the tolerances on steady-state performance.

3.4 INTERFACE LOGIC

In this section we will describe the design and operation of the digital logic that places the interface functioning under PDP-1 program control. Included is a discussion of the implementation of the discrete input and output and timing control hardware.

A. Physical Layout

The digital interface was constructed using Digital Equipment Corporation (DEC) modules. Certain additional special-purpose components, unavailable from commercial sources, were fabricated on blank DEC cards compatible with the rest of the modules used in the interface. The basic logical elements in the hybrid system are described in Chapter V.

Figure 3.6 indicates the placement of each of the units mounted on the interface rack. Panels A, B, C, and D are standard DEC taper pin wired mounting panels having a capacity of 25 modules each. The module sockets on each panel are numbered from 1 to 25, left to right, looking at the wiring side of the panel. Each module socket has 22 pins lettered A through Z with G, I, O, and Q omitted. Thus, to uniquely specify a pin on one of the DEC panels on the interface rack, one must designate a letter (panel identifier), a number (module socket identifier), and another letter (pin identifier). For example, B22E specifies the E pin on the 22nd module socket from the left on panel B. Panel E, which contains the Burr Brown high-speed operational amplifiers, has some 40 taper pin connections and 8 BNC connections on it. The nomenclature of these connections will be discussed later. The two 50 conductor cables connecting the interface equipment to the PDP-1 are Cable A and Cable B. The nomenclature of wires in these cables is a pair of letters, either CA or CB, specifying the cable, followed by a number from 1 to 50, specifying the conductor number. The other nine connectors on the connector panel are associated with the input-output console and will be described later. The digital logic block-diagrams have the location of each pin labeled. The location of a given module is shown adjacent to the module-type number; the pin designation is shown adjacent to the given wire as it enters or leaves the given module. If the source or destination of a given wire is not included in a given drawing, the pin designation of the other end of the wire is given in square brackets (like [B22F]).

The digital interface equipment falls conveniently into two categories. One category includes the discrete inputs, discrete outputs, clock circuit and mode switching logic, all of which are under the control of the PDP-1 iot instruction. The second category is the operate gates or opg instruction implementation. The main features of the system interface logic are presented in Fig. 3.7.

B. Time-sharing Assignment Level (TSAL)

One feature which is common to both the opg and iot instructions is control of operability by an enable level. Throughout the logical diagrams, a control level called TSAL appears. TSAL stands for time-sharing assignment level. This interlock level has been included to protect the hybrid equipment from being operated by other users when the PDP-1 is in a time-shared mode of operation. When the simulation program is alone on the machine, TSAL is temporarily supplied by the circuit shown in Fig. 3.8. If the TSAL is assertive (ground), then the equipment performs as described below; otherwise the equipment does not respond to any command.

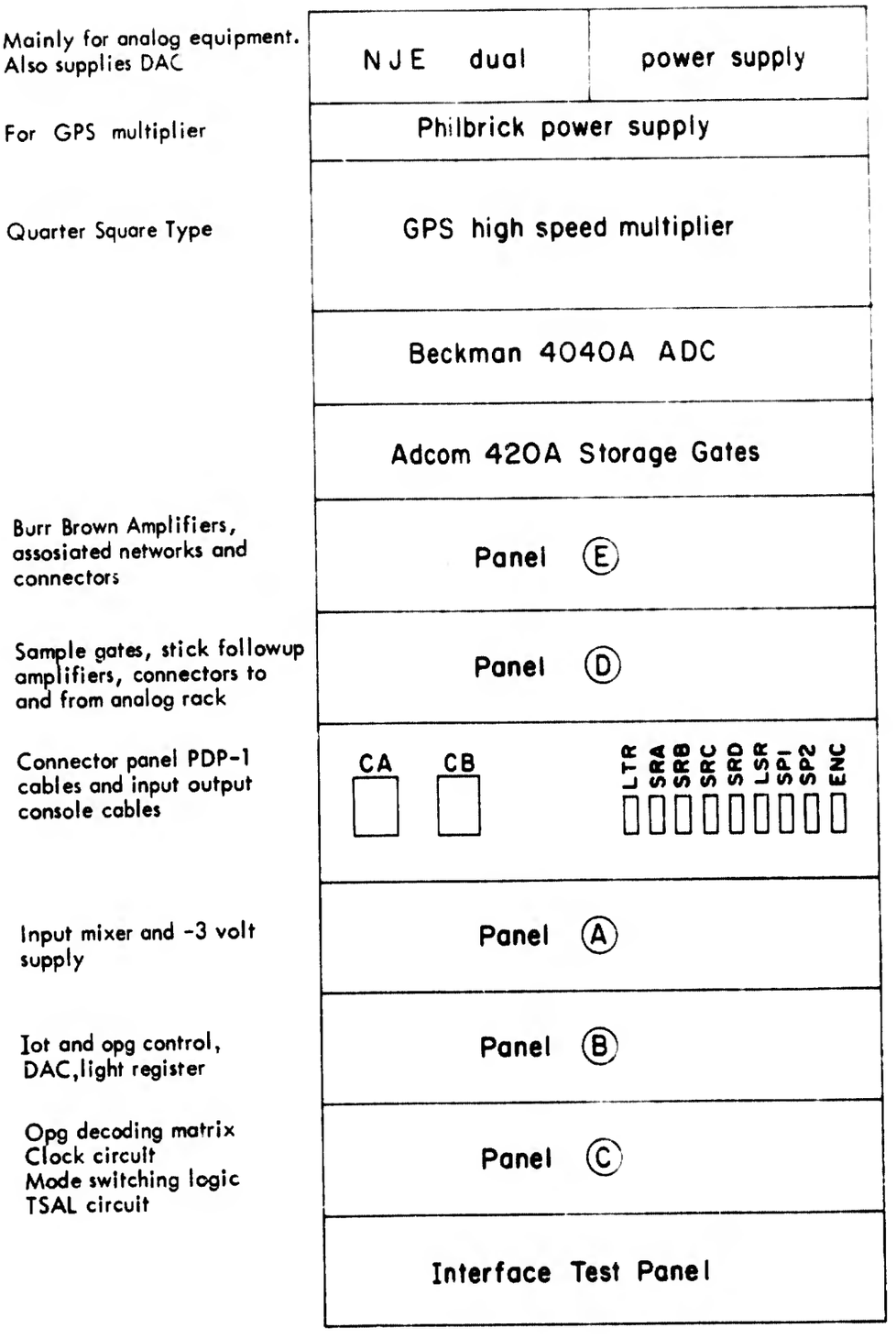


Fig. 3.6 Layout of Interface Rack

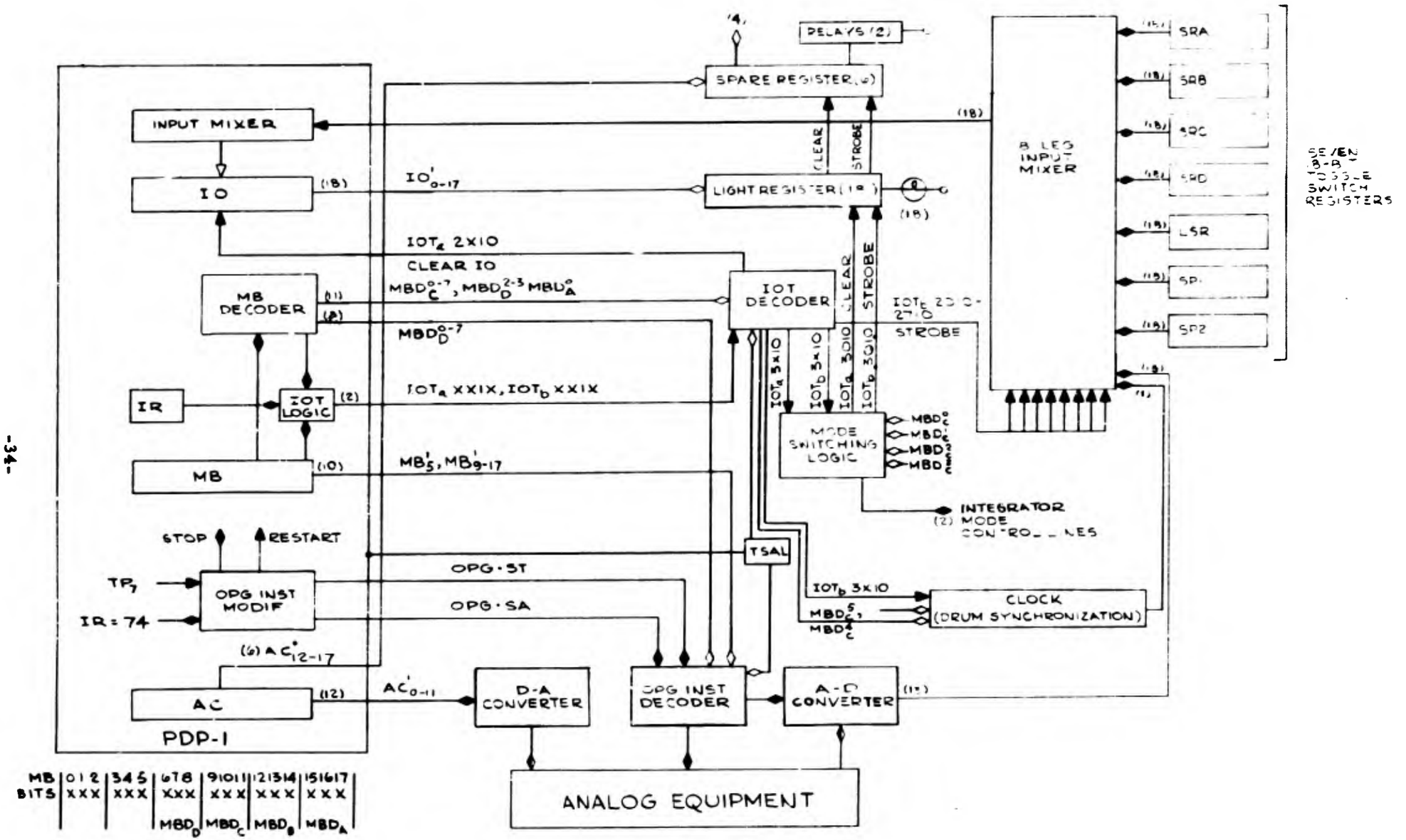
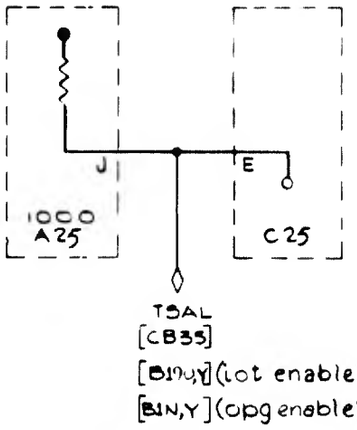


Fig. 3.7 System Interface Diagram



NOTES:
1. IF THE JUMPER MODULE (C25) IS INSTALLED, THE ENABLE LEVEL IS ASSERTIVE; OTHERWISE IT IS NEGATIVE.
2. THE CENTRAL PROCESSOR END OF THE TIME SHARING ASSIGNMENT LEVEL [TSAL] SIGNAL LEAD IS LEFT UNCONNECTED.

Fig. 3.8 TSAL Circuit

C. Digital Inputs

Digital inputs to the PDP-1 include the discrete inputs, the output of the analog-to-digital converter (ADC), and the output of the clock flip-flop. These inputs represent a total of eight 18-bit words which are transmitted to the PDP-1. Since only one 18-bit channel was made available at the PDP-1 installation for the hybrid work, a multiplexing-type operation external to the PDP-1 was used which effectively gave eight input channels. This multiplexer is actually an 8-leg input mixer. Upon issuance of one of eight special iot commands, the levels presented to the selected leg or channel of the input mixer are transmitted in parallel to the IO. A block schematic of the input mixer appears in Fig. 3.9.

Seven of the eight legs of the input mixer are occupied by the discrete input toggle switch registers. Bits 0 through 12 of the eighth leg come from the output of the ADC. Bit 17 of the eighth leg comes from the clock flip-flop. The pulse which strobes the selected leg of the mixer is generated in the PDP-1 by the timing chain, gated through the PDP-1 iot logic and finally gated through the iot decoder shown in Fig. 3.10. It was found necessary to keep certain iot decoder transistor emitters from going positive by connecting diodes from those emitters to ground. The diodes themselves were mounted on a separate card and wired to the decoder.

The operational command in the PDP-1 to strobe a leg of the input mixer is (octal) 722X10 where the digit X (0 through 7) selects the desired leg. The entire IO is set to zero before ones are transferred through the selected leg.

Each leg of the input mixer is wired to a 25-pin connector on the cable connector panel (see Fig. 3.6). In addition to the 18 input mixer leads, each such 25-pin connector has one lead fed by a -3 volt supply and one lead grounded. The eighth leg of the input mixer is connected to a module socket in addition to being connected to a 25-pin connector. The 25-pin socket labels corresponding to the given octal command codes are outlined in the following table. The socket labels are

Table 3.2

Digital Input Commands

Command	Socket Label	
722010	SRA	
722110	SRB	
722210	SRC	
722310	SRD	
722410	LSR	(lever switch register)
722510	SP1	
722610	SP2	
722710	ENC	$ADC_{0-12} \rightarrow IO_{0-12}$ Clock State $\rightarrow IO_{17}$

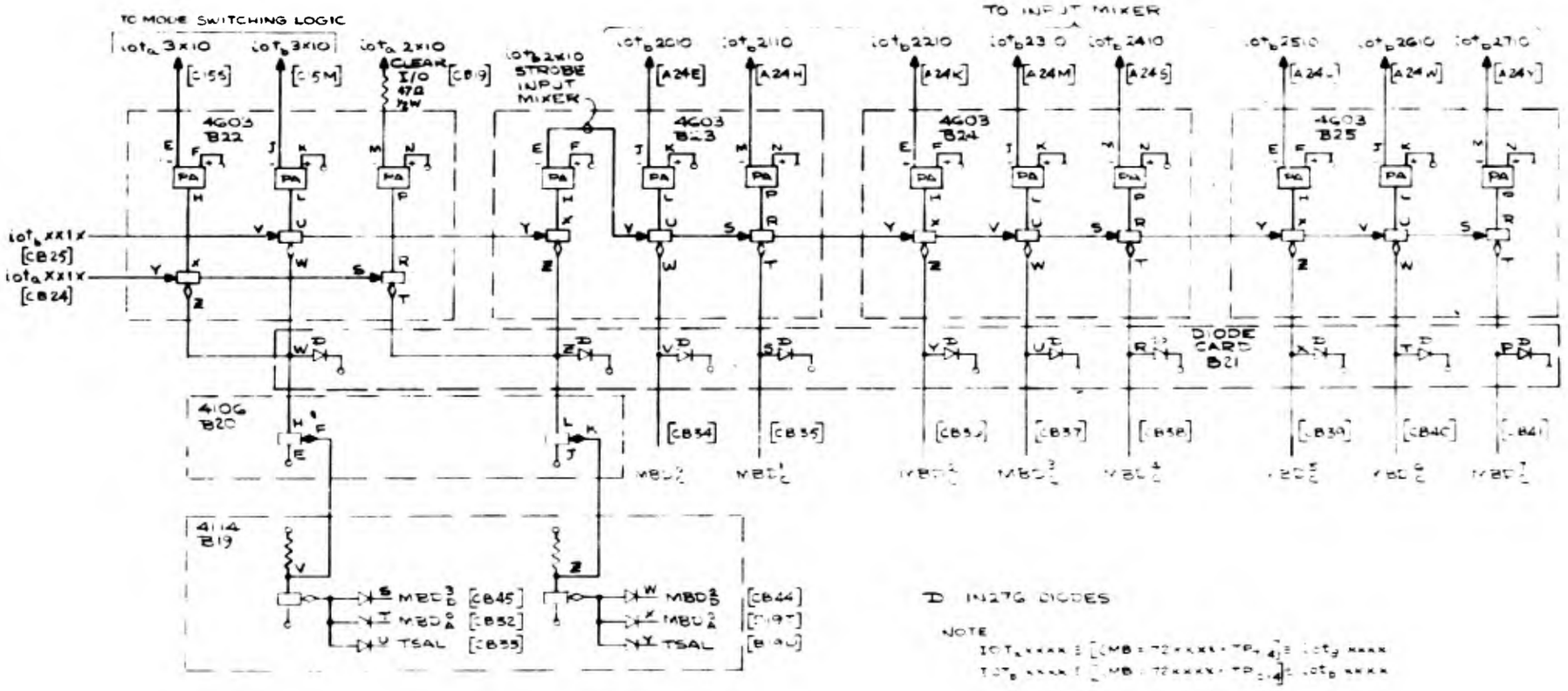


Fig. 3.10 I O T Gating and Amplification

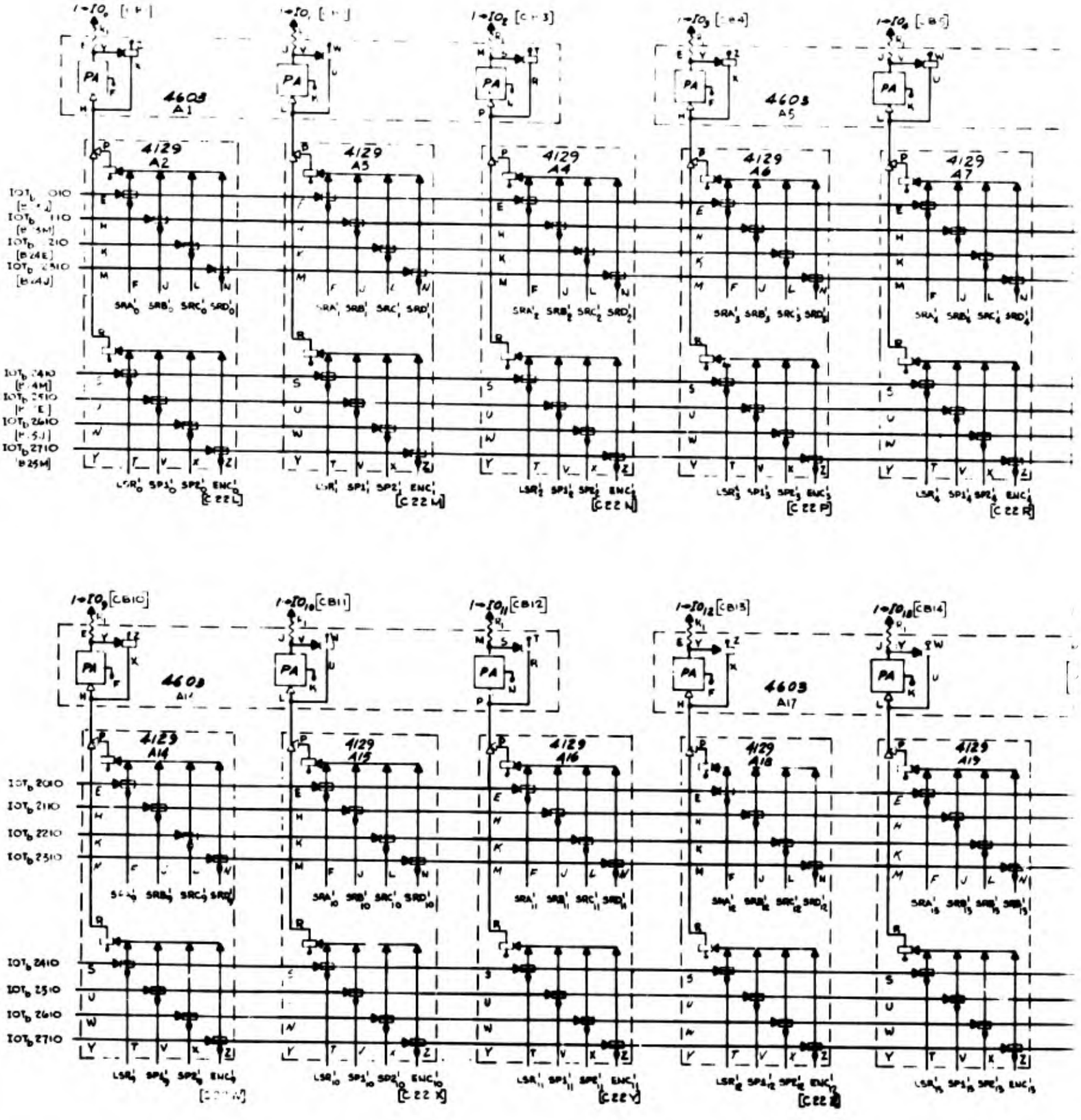
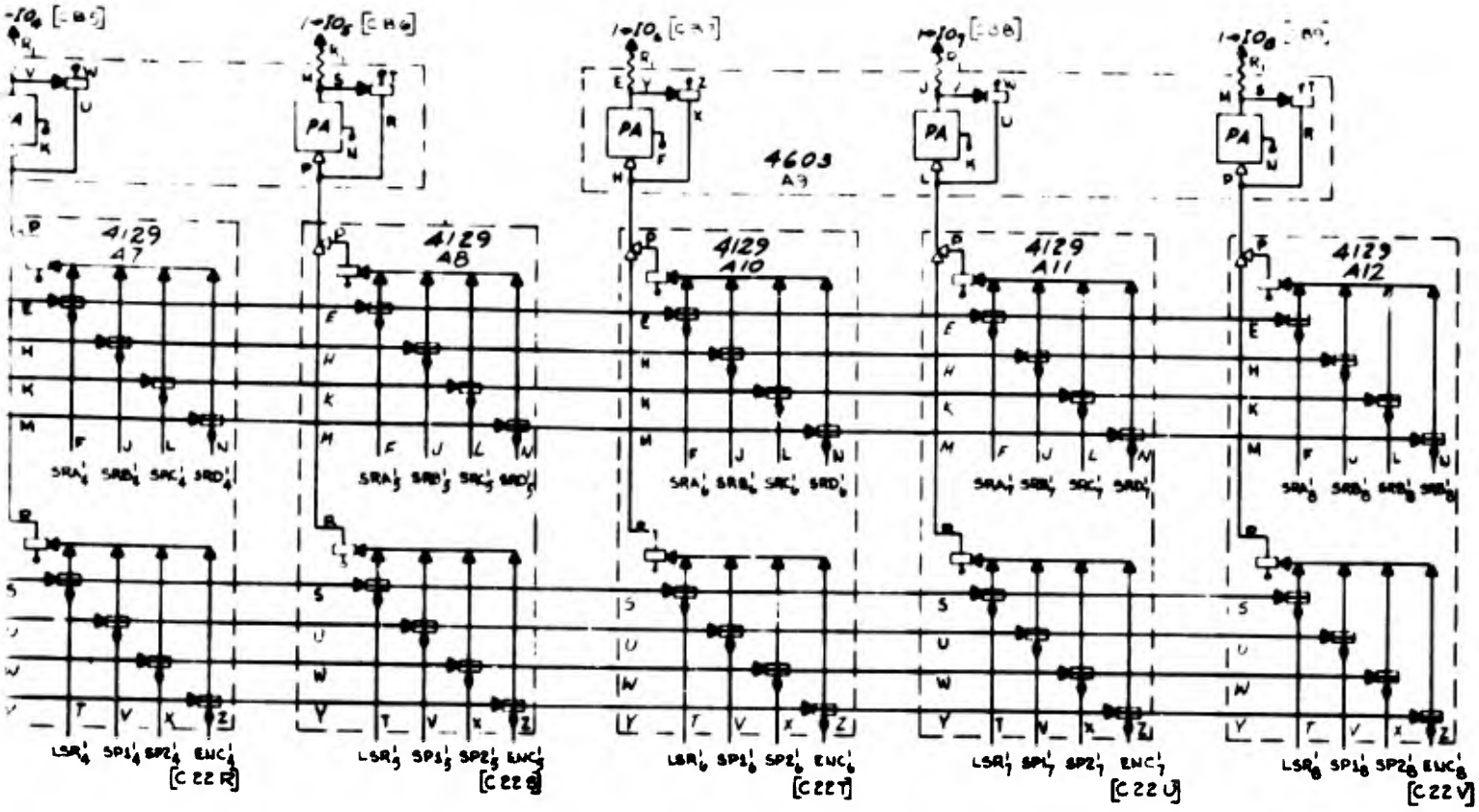


Fig. 3.9 Input Mixer for Digital

A



NOTE: R_1 = 47 Ω ½ W

NOTE
CABLE SOCKET PIN NUMBERS FOR LEVEL INPUTS TO MIXER ARE THE CORRESPONDING SWITCH LEVEL NUMBERS, INCREASED BY ONE (EX. LEVEL LSR_0 COMES FROM PIN LSR_1)

al Mixer for Digital Inputs

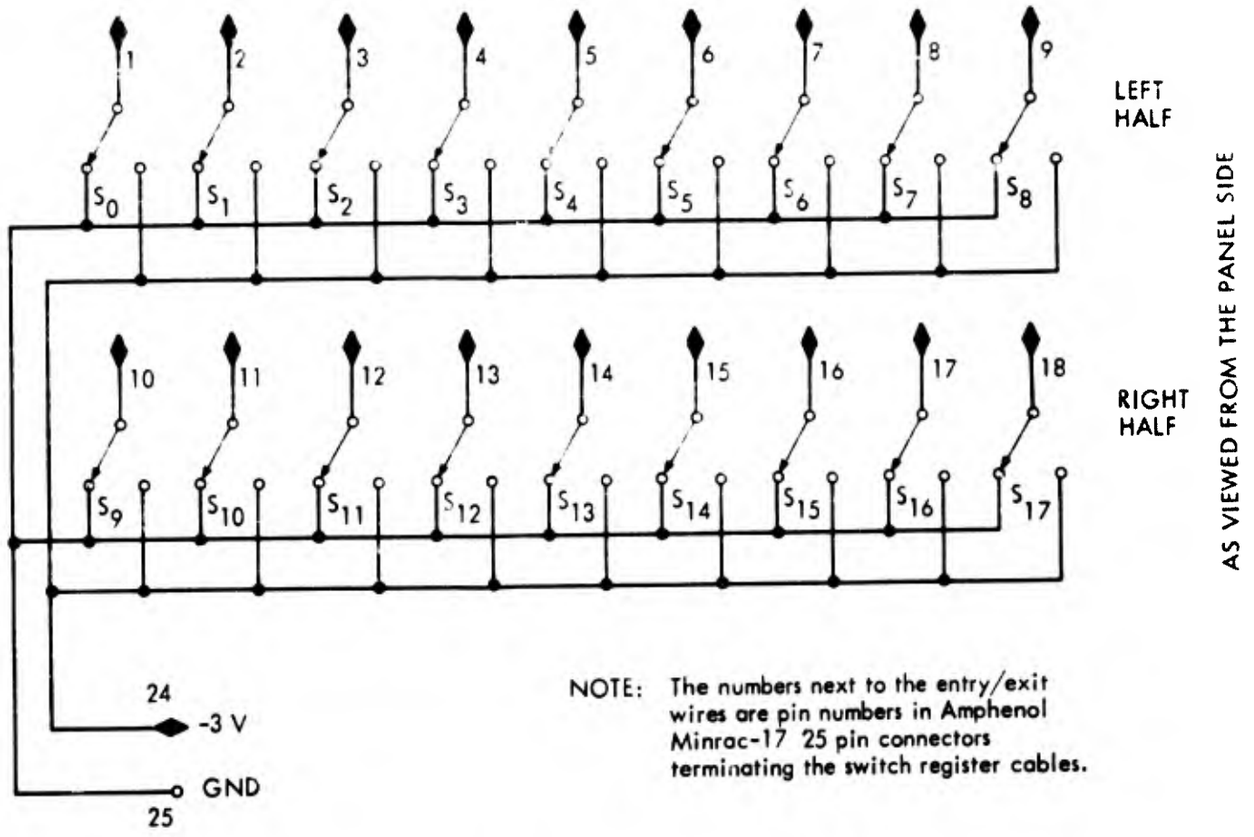


Fig. 3.11 Discrete Input Toggle Switch Register Wiring

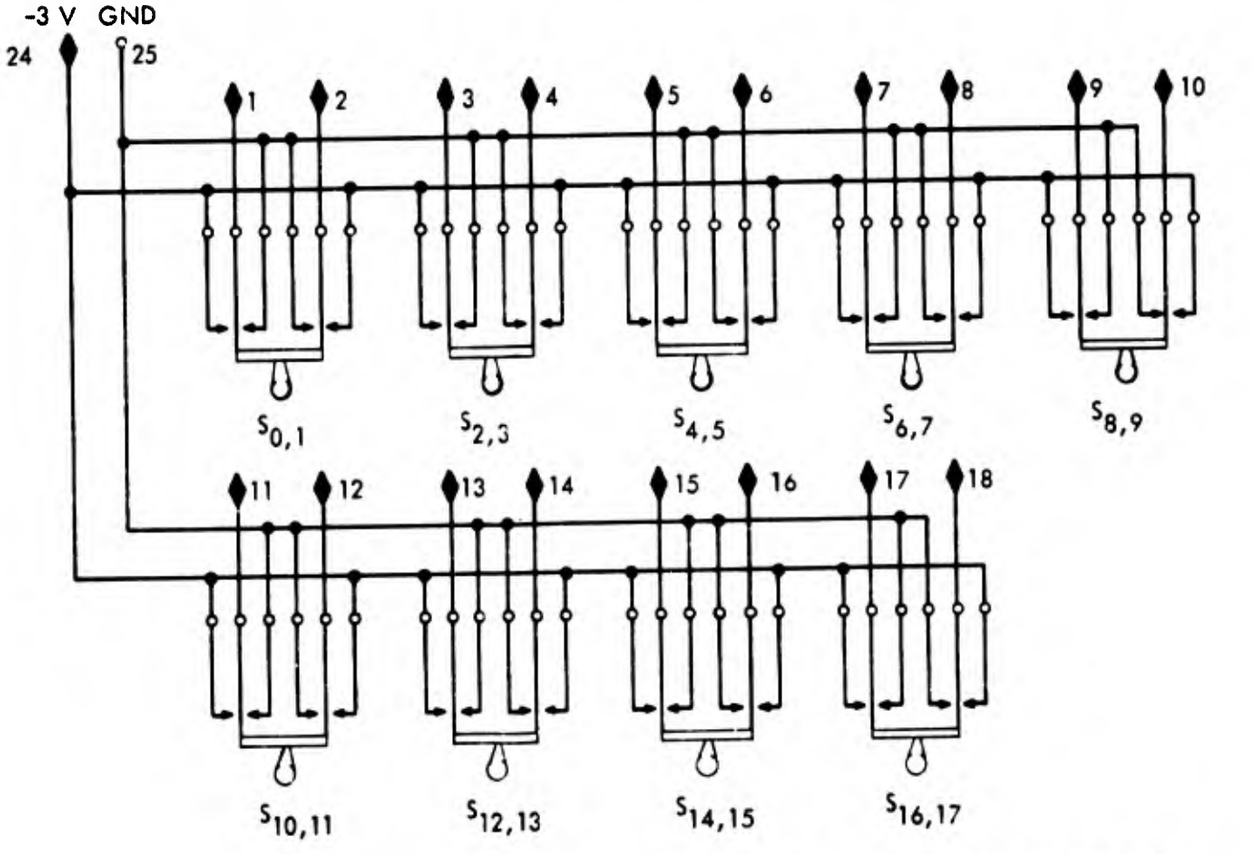


Fig. 3.12 Lever Switch Panel Circuit Diagram

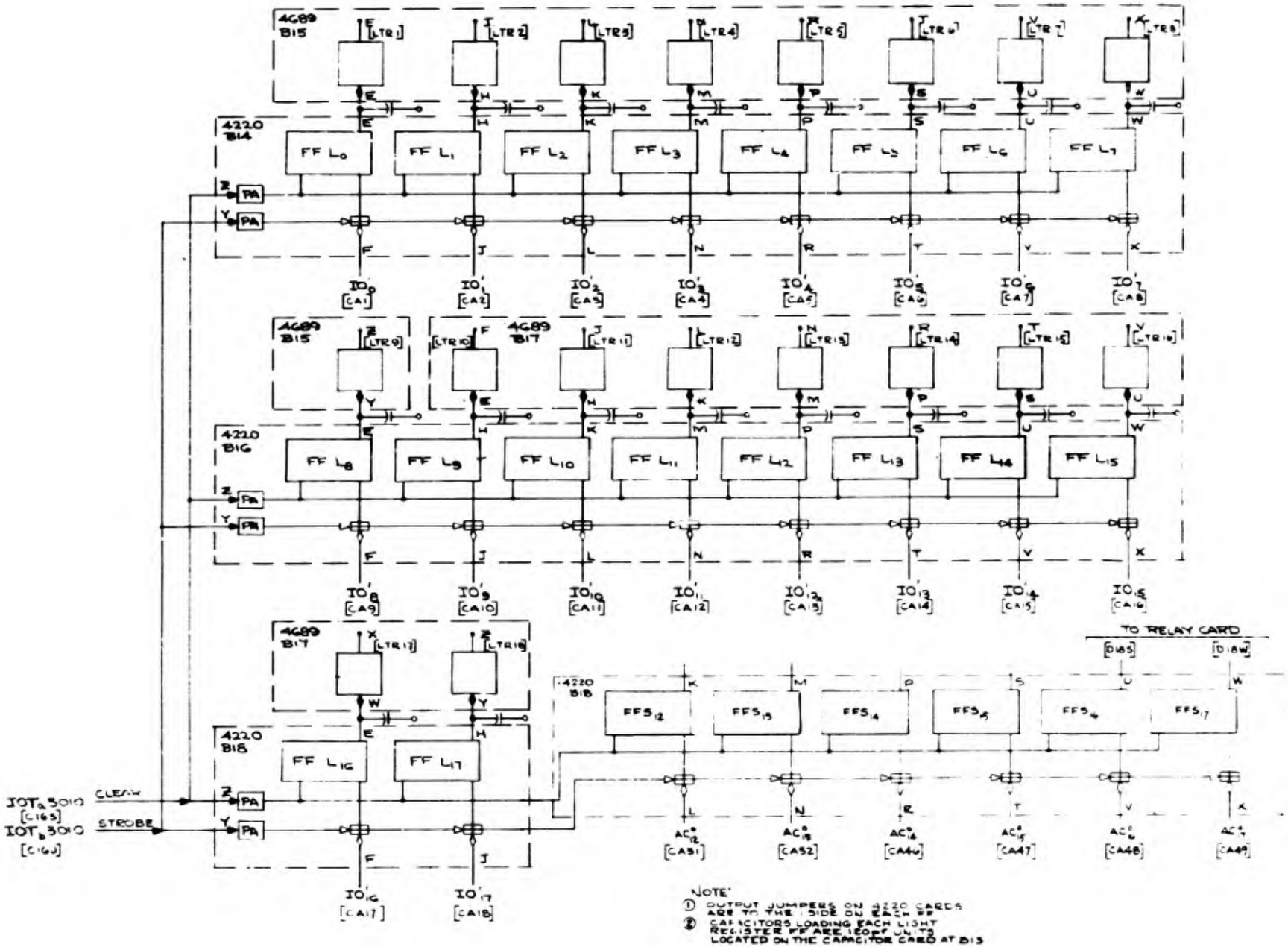


Fig. 3.13 Light Register; Spare Register

identical with the mnemonic instructions chosen to represent the octal commands in the F-100 programs.

The seven discrete input toggle switch registers are mounted on a panel of the input-output console. The wiring of a typical toggle switch register is shown in Fig. 3.11. When a given switch is in the down, off, or zero position, a ground level is supplied to the capacitor diode gate corresponding to the given switch. The ground level blocks the passage of the strobe pulse through the gate, thereby inhibiting the setting of the corresponding IO flip-flop. When the given switch is up, on or one, a -3 volt level goes to the given capacitor diode gate, allowing the strobe pulse to pass and set the corresponding IO flip-flop to one. Each switch register is wired to a separate cable which plugs into one of the eight 25-pin connectors feeding the input mixer.

An extra panel of nine three-position lever switches was built to conveniently accommodate the incrementation and decrementation of variables in the all-digital F-100 program. Each switch controls two bits of an input mixer leg. For example, the left-most switch turns bit 0 on when up, bit 1 on when down, and neither on when centered or neutral. The circuit diagram of this panel of switches is shown in Fig. 3.12. When in use, the output of the lever switch panel is plugged into the connector used for the seventh toggle switch register.

D. Digital Outputs

A 24-bit flip-flop register is the main digital output means. The 24-bit register is operationally divided into an 18-bit register and a 6-bit register. The 18-bit register is called the light register because each of its outputs is connected through an indicator driver to a lamp on the input-output console. The 6-bit register is called the spare register. Its existence is purely a useful by-product of circumstance. The block schematic of the light and spare registers appears in Fig. 3.13.

The light register is loaded from the IO by the (octal) 723010 command. The loading command first clears the light register and then, about 3 μsec later, sets those flip-flops of the light register that have a one in the corresponding flip-flop of the IO. The spare register is loaded from the AC bits 12 through 17 at the same time, by the same command, and by the same process that loads the light register from the IO.

Bits 16 and 17 of the spare register control two utility relays on a special module housing four relays and their associated drivers. The remaining two relays are used in the integrator mode control logic which will be discussed later. Figure 3.14 gives the schematic diagram of the relay card.

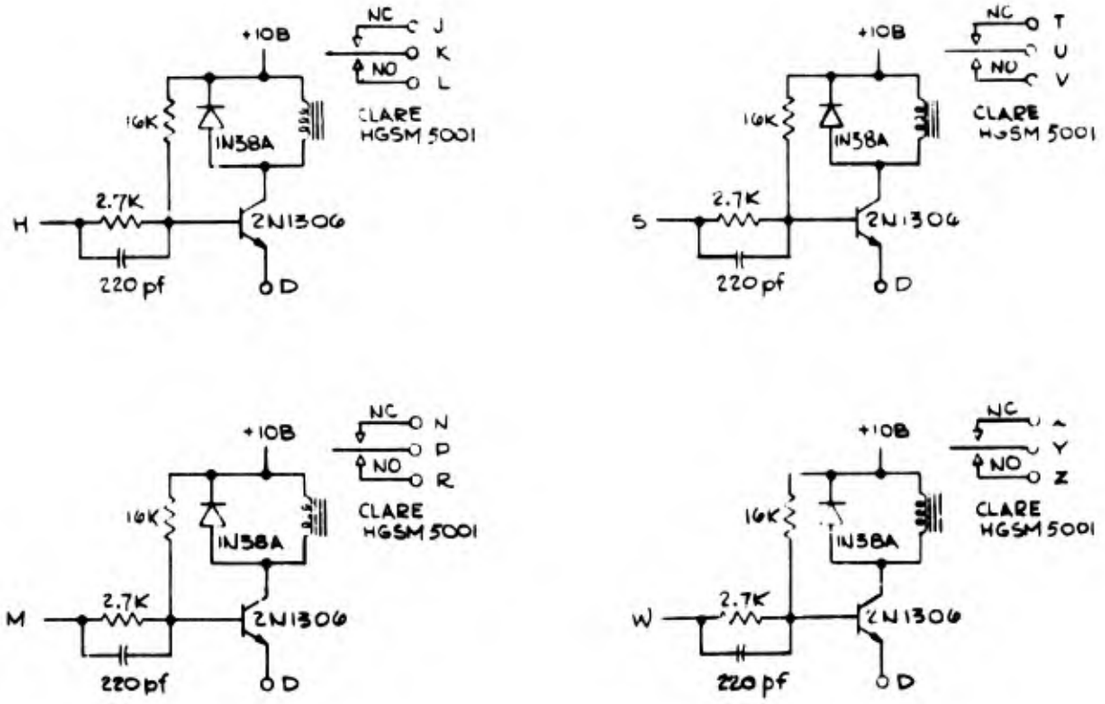


Fig. 3.14 Relay Circuit

The 120 pf capacitors shunted across the outputs of the light register flip-flops were found to be necessary to balance the capacitive load on those unbuffered flip-flops. The capacitors in question were conveniently mounted on a blank module and wired to the flip-flops using the normal taper pin wiring technique. When an application for the four unused spare register outputs arises, care must be taken to assure sufficient capacitive loading of these outputs to prevent nondeterministic behavior.

The outputs of the indicator drivers reach the lights on the input-output console through a 25-pin connector (labeled LTR). The indicator driver circuit is such that the lights act as collector loads to the output transistors. Thus, one side of each indicator light is connected to the collector of a driver transistor; the other side of each light is tied to the -15 volt supply. Therefore, the -15 volt supply is also wired to connector LTR.

The decoding of the 723010 instruction is somewhat involved; therefore, in this case, we shall trace the path followed by the command pulses. The initial decoding takes place in the IOT gating unit (Fig. 3.10) and yields the pulses for command 723X10. These command pulses go to the mode switching logic (Fig. 3 16) and are there decoded to 723010. From the mode switching logic, the final command pulses enter the light and spare register in Fig. 3.13. The decoding that takes place in the mode switching logic is for hardware economy.

E. Clock

A clock or interval timer was found necessary to provide an intermediate 50 millisecond time reference in the initial hybrid F-100 program, which operated on a 100 ms. base cycle synchronized with the magnetic drum storage unit. The mechanization for the clock is such that the beginning of the 50 ms. interval is set by an iot command. The end of the given interval is known to the program through the status of a flip-flop feeding bit 17 of the encoder (ENC) leg of the input mixer. The block schematic of the clock circuit appears in Fig. 3.15. The two iot commands associated with the clock circuit are:

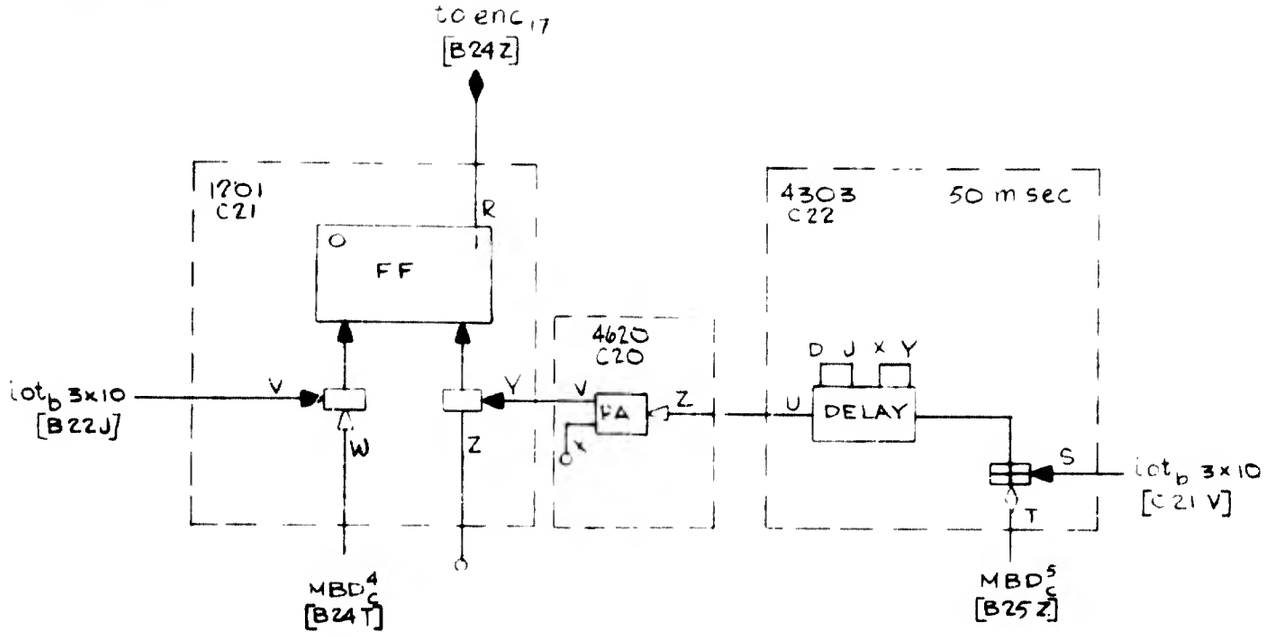


Fig. 3.15 Clock Circuit

723410	zck	zero the clock flip-flop
723510	sck	start clock timing

This circuit could directly supply the 1 or 16 channel sequence break systems with the required request and status information if such operation were ever desired.

F. Mode Switching Logic

In the operation of the hybrid analog-digital computation system, it is often convenient to switch the operating mode of an analog integrator under digital program control. Examples of digital mode switching utility occur in fast repetitive operation and in stopping a running problem and examining its status.

Figure 5.5 gives the schematic diagram of an integrator used in the system. Relays K_1 and K_2 are the mode switching relays. The states of the relays determine the integrator mode as follows:

mode	K_1	V_1	K_2	V_2
operate	closed	0	open	0
hold	open	-15	open	0
reset	open	-15	closed	-15

The logic shown in Fig. 3.16 has been designed so that three different instructions set the mode logic to one of the three states. The given instructions are as follows:

Mnemonic	Op code	Description
rsi	723110	reset integrators
opi	723210	operate integrators
hdi	723310	hold integrators

In switching into any mode but operate, switching transients are non-critical. When switching from reset to operate, one must be careful that K_2 opens before K_1 closes. If this condition is not met, transients may occur that cause inaccuracies. Thus, a DEC 4301 delay circuit prevents K_1 from closing before K_2 opens. Common mode control lines (V_1 and V_2) service all the integrators in the analog system.

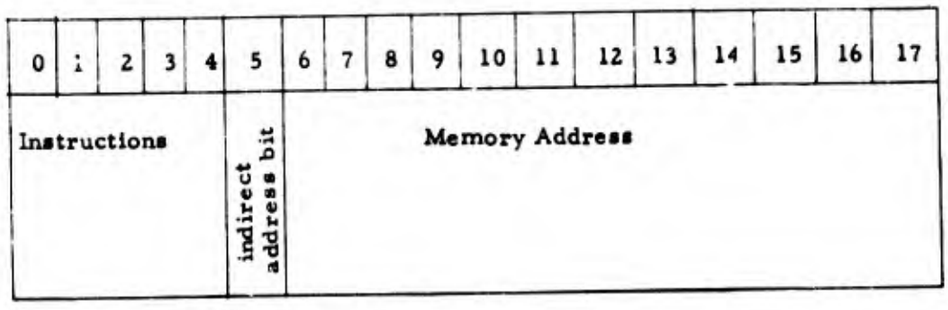
The drive capability of the mode state flip-flops must be boosted by the relays shown in Fig. 3.16 to meet this requirement. Switch "Auto-Man" allows manual control of integrator modes for debugging purposes.

Finally, two channels of the DEC 4603 pulse amplifier shown in Fig. 3.16 are used for clearing and strobing inputs into the light and spare registers. The instruction IOT 3010 first clears these registers and then strobes IO_{0-17} and AC_{12-17} respectively into them.

G. Operate Gates Instruction

The principal logical provision added to the PDP-1 to implement the analog-pulsed analog-digital demonstration was a direct operate gates instruction to activate the storage and sample gates under digital program control. The address part of this instruction designates the gates to be activated.

The 18-bit PDP-1 instruction has the format indicated below:



The instruction codes (octal) not used in the PDP-1 are 00, 14, 36, and 74, the last being in the 60-70 group of augmented (5 μsec) instructions. The computer will normally halt if any of these instructions are used.

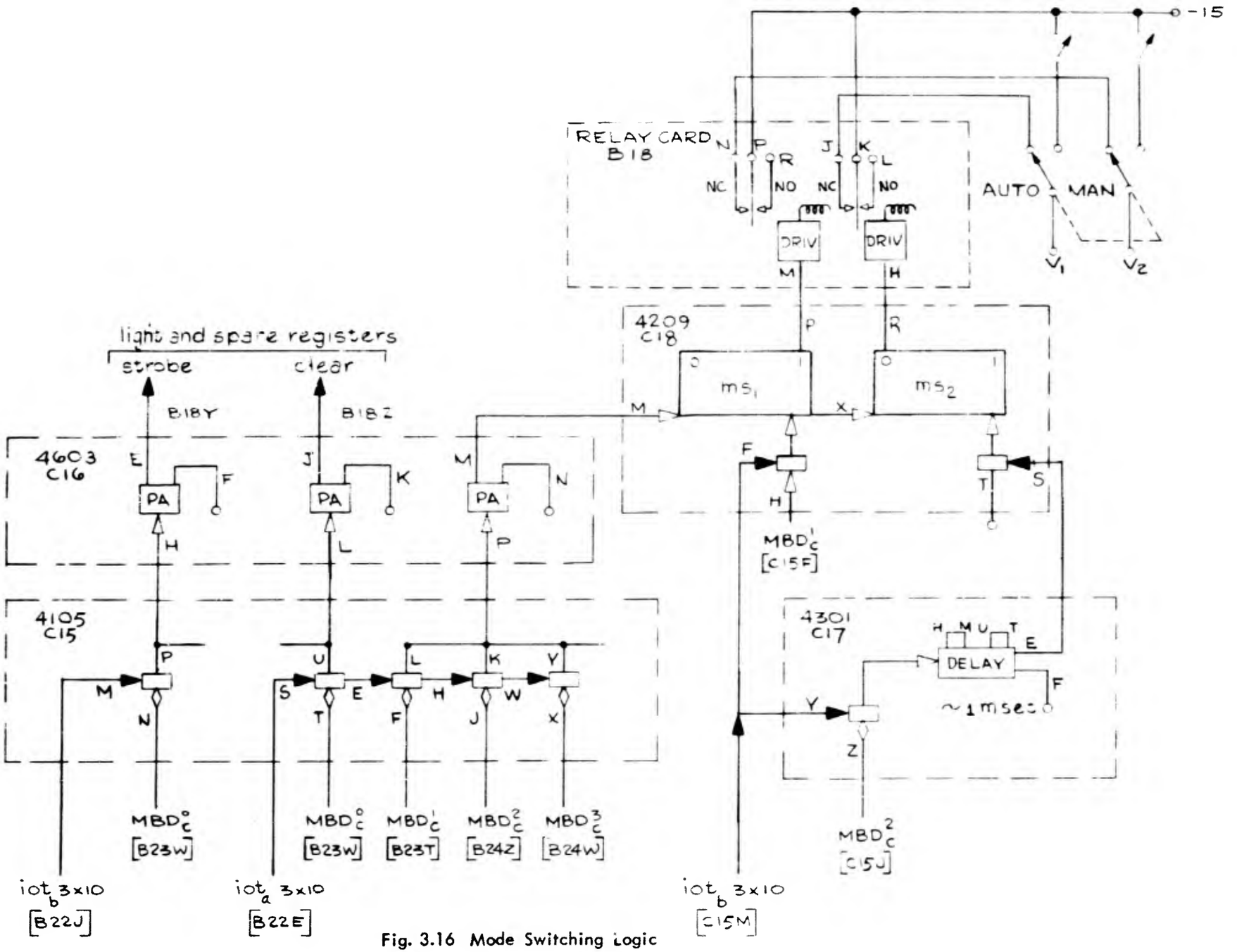


Fig. 3.16 Mode Switching Logic

The hybrid design uses one of these instructions (74) as an operate gate order and as an encoder control. Bits 0-6 carry the instruction designation 74 (75 if the indirect address bit is one). Bits 6, 7, and 8 are decoded to place a control level on one of eight lines. The state of individual bits from 9 to 17 control the levels on another set of nine lines, corresponding respectively to the decimal digits 1, 2, ... 9. The combination of the eight lines and the nine lines give the ability to initiate 72 distinct operations in the analog domain, up to nine at a time. Each of these operations may be assigned a decimal number 1, 2, 3, ... 9, 11, 12, ... 78, 79, the least significant decimal digit indicating which of the 9 bits 9 through 17 control the operation, the most significant decimal digit indicating which of the eight control levels determined by bits 6, 7, and 8 control the operations. To illustrate:

$$740001_{octal} = 111 \mid 100 \mid 000 \mid 000 \mid 000 \mid 001_{binary} = \text{activate operation 9}$$

$$744001_{octal} = 111 \mid 100 \mid 100 \mid 000 \mid 000 \mid 001_{binary} = \text{activate operation 49}$$

$$745061_{octal} = 111 \mid 100 \mid 101 \mid 000 \mid 110 \mid 001_{binary} = \text{activate operations 59, 55, 54}$$

The pulsing of the encoder in this scheme is controlled by the indirect address bit, that is, the instruction 75_8 will store a sampled analog voltage at the input to the encoder and start the encoding action, but the instruction 74_8 will not.

When the operate gates instruction is used, the address part of the instruction, which designates the gates to be activated, appears in MB_{6-17} at time pulse 4 of cycle zero (TPO.4). The instruction code 74_8 is not transferred from MB_{0-5} to the instruction register until TPO.5, hence the activation pulses to the selected gates can occur no sooner than TPO.6. In the present system, TPO.7 initiates the activation of selected gates. The storage gates remain in the activated state for 15 μsec. The sample gates remain in the activated state for 13 μsec. These activate periods are individually adjustable for each class of gates. Storage gates must be turned off a short time before the sample gates to prevent the turn-off transient from leaving an erroneous voltage on the storage capacitor.

One of the storage gates (T10) controlled by the operate gates instruction is at the input of the analog-to-digital encoder. If the indirect address bit, in this case MB_5, is a one, this gate will be activated at TPO.7 and de-activated 13 μsec later, just like all other storage gates. At turn-off, the voltage to be encoded will be held on the T10 storage capacitor. The transition of this particular gate from the activated to de-activated state will automatically start the encoding action, which requires some 50 μsec to complete in the Beckman 4040A. The computer may perform other instructions while waiting for the completion of the encode. When ready, the contents of the encoder register are strobed by an IOT instruction (mnemonic symbol ENC) into the PDP-1 In-out register.

The operate gates hardware is split logically into two sections. One section is internal to the PDP-1 central processor and provides instruction timing information, while the second section is housed in the equipment rack of Fig. 3.6 and provides the decoding logic. First consider the section internal to the PDP-1. The block schematic of the operate gates timing control unit appears in Fig. 3.17. Assume for the following description that the control switch is set such that the opg instruction operates. In the complementary position of the switch, opg is considered an illegal instruction. For any instruction code other than 74_8, no action occurs. On instruction 74_8 the circuit of Fig. 3.17 sends a level to the PDP-1 stop logic to cause the machine to halt at the end of the current memory cycle. At time pulse TPO.7, the two delay units begin their timing cycles. The level output of each of the delay units is sent via cable A from the PDP-1 to the hybrid equipment racks. When the ST (13 μsec) delay finishes its timing cycle, it generates a pulse to notify the PDP-1 proceed logic to initiate the next memory cycle. Figure 3.18 shows the timing involved with the opg levels.

The two opg timing levels control the on times of the two types of analog gates. The sample gates must remain on for a longer period than the storage gates so that the voltages on the storage gates are not disturbed by the turn-off transients of the sample gates. Experimentally, it was found that for completely reliable operation, the storage gates should remain in the "track" condition for 13 μsec. The sample gates are turned off about 2 μsec after the storage gates are set to the "hold" condition. Thus, the 13 μsec delay level is the $\overline{OPG} \cdot$ ST ACTIVE level for controlling the storage gates. The 15 μsec delay level is the $\overline{OPG} \cdot$ SA ACTIVE level for controlling the sample gates.

In addition to the two opg timing levels, the states of memory buffer register bits 5 through 17 are transmitted to the equipment racks. The states of the three bits MB_6 through MB_8 are transmitted in a one-out-of-eight decoded form from the memory buffer decoder D (MBD$_D$). When these three types of levels first enter the external opg logic, the levels are buffered to give them driving capability. Then the buffered levels drive the opg decoding matrix. The block schematic of the external opg logic appears in Fig. 3.19.

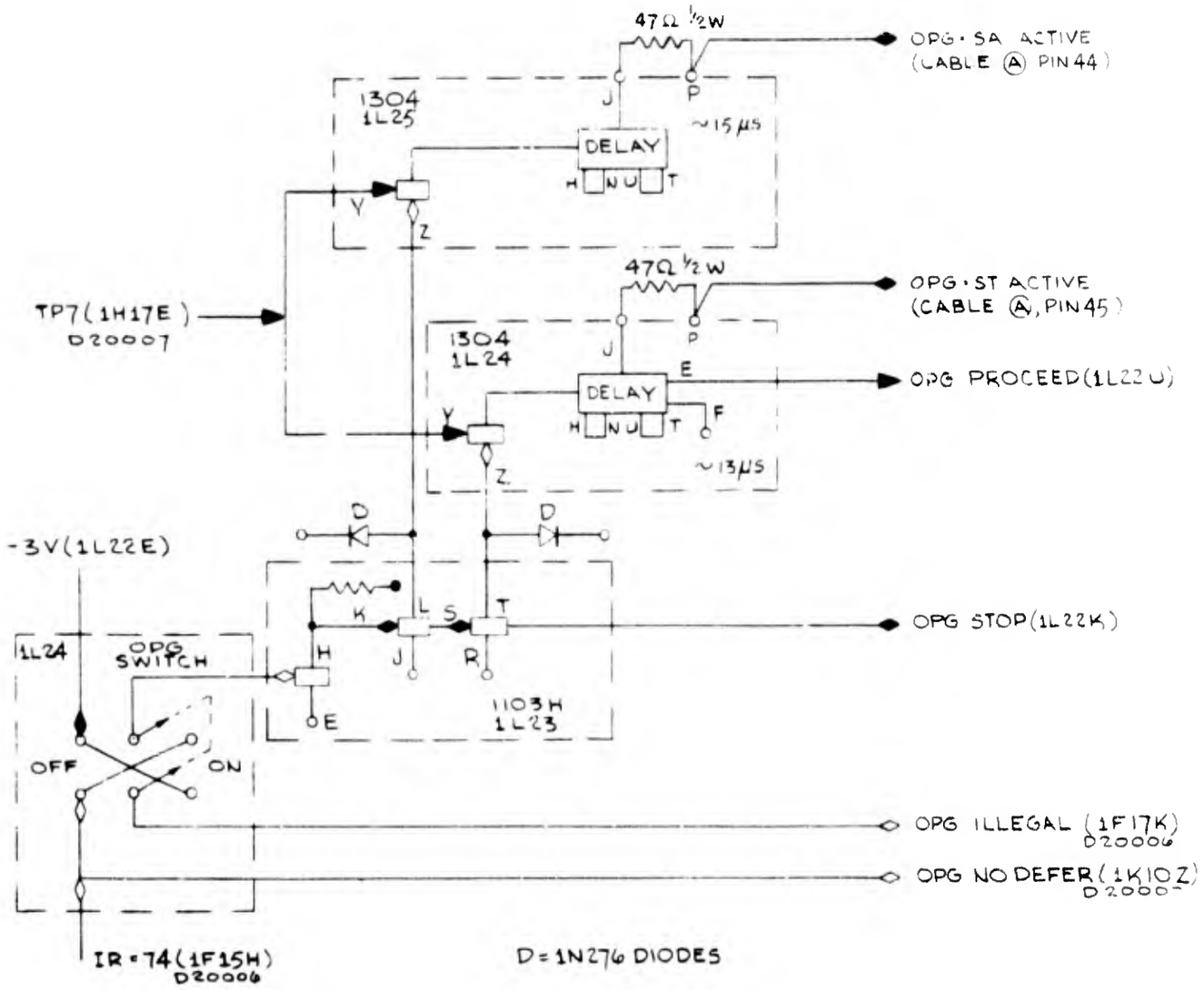


Fig. 3.17 CPU Modification for OPG Instruction

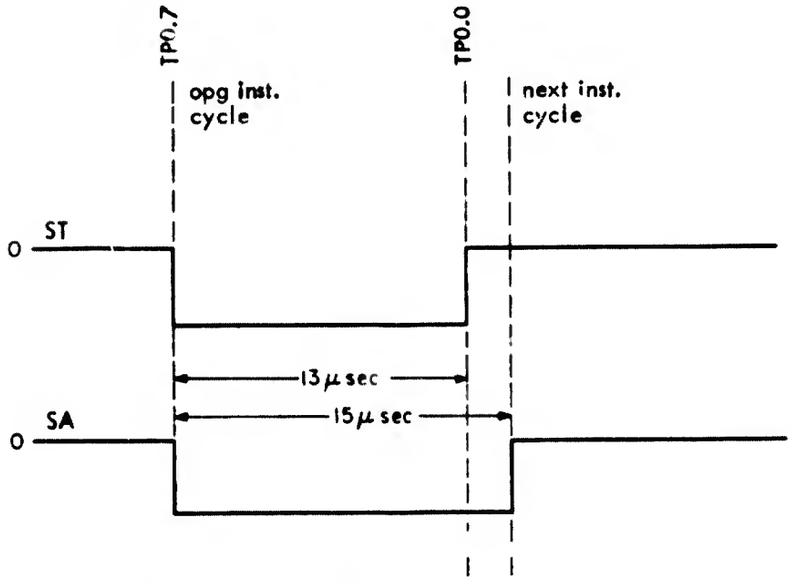
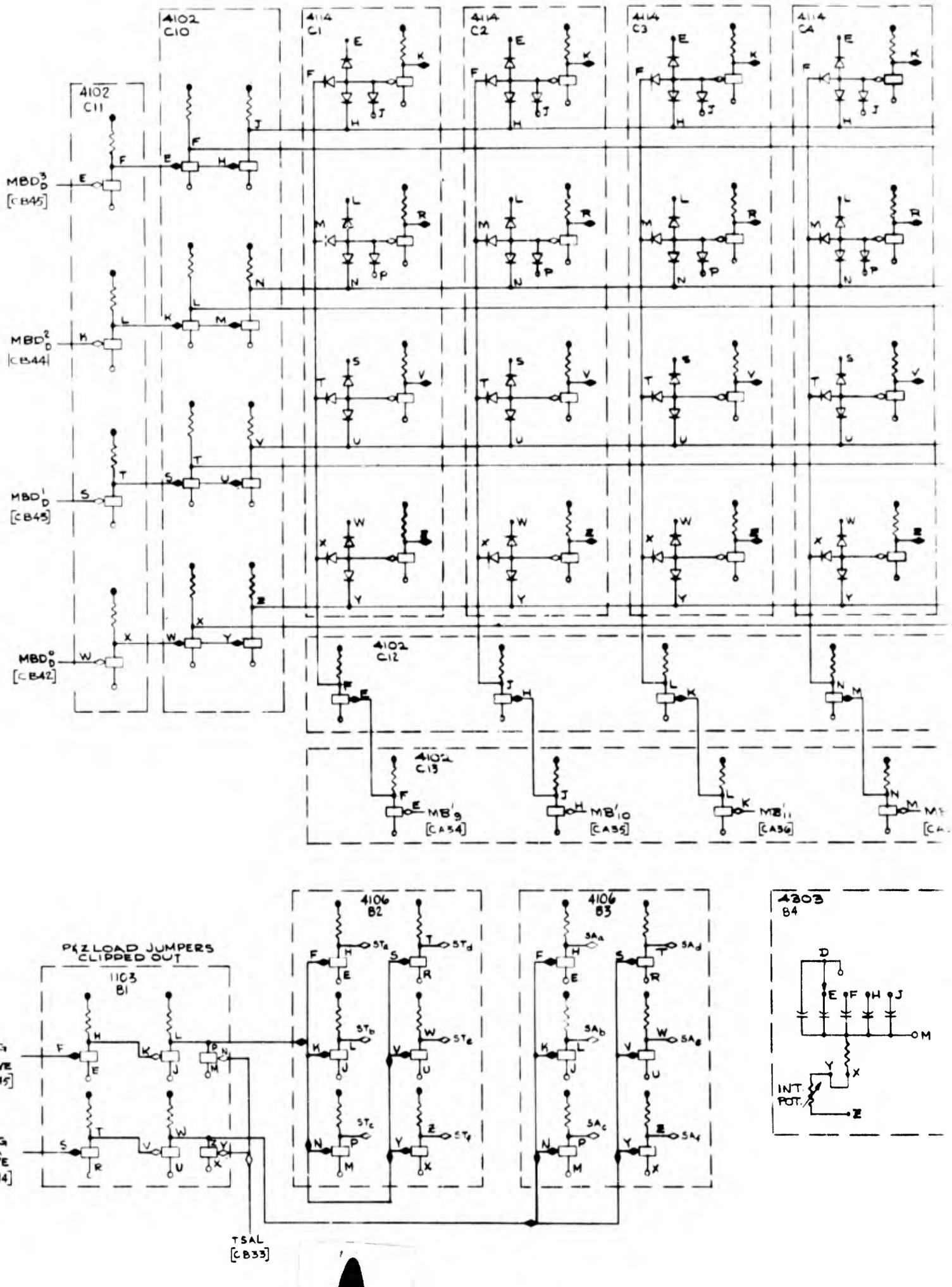
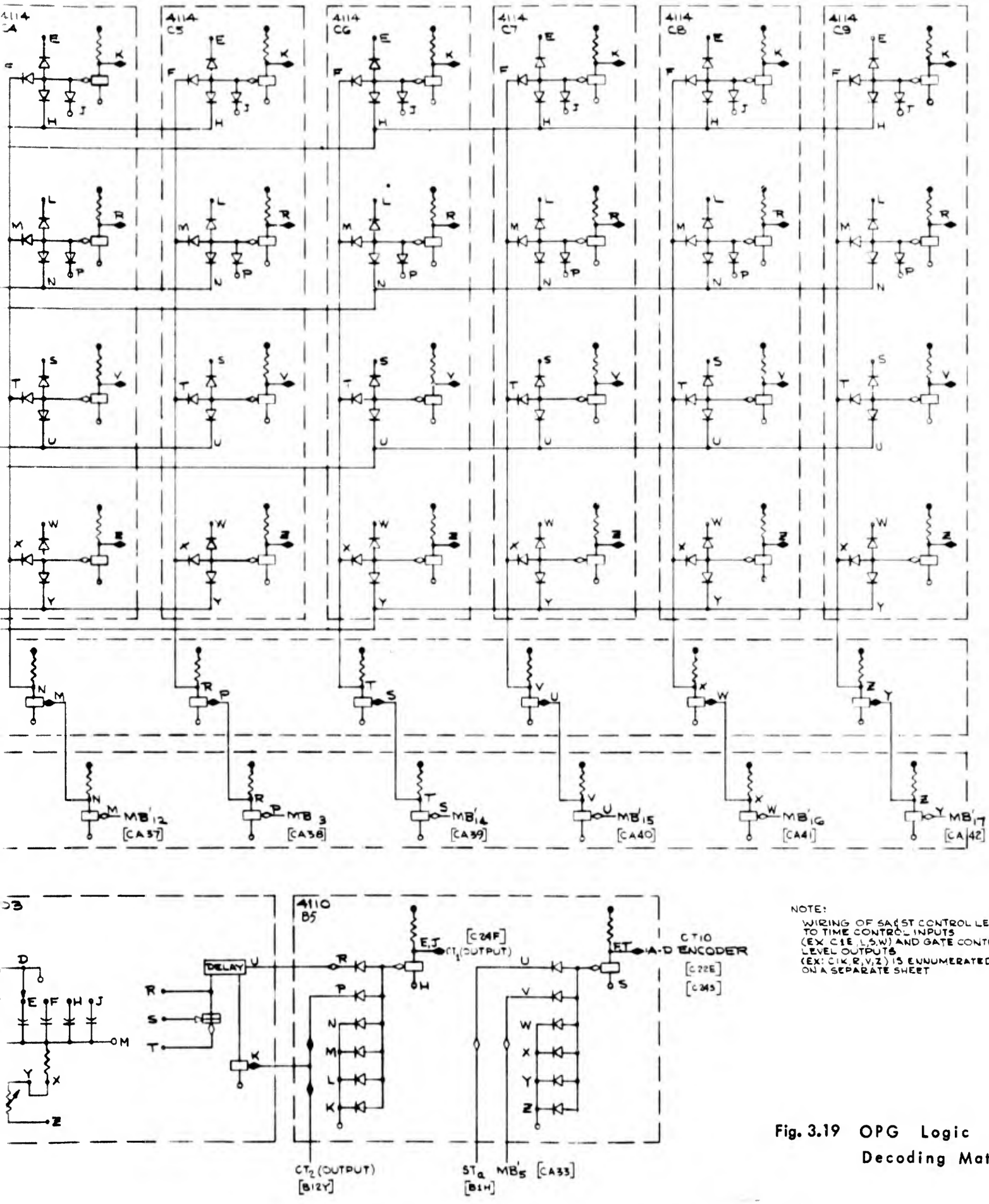


Fig. 3.18 OPG Level Timing


4102
C10
4102
C11
4114
C1
4114
C2
4114
C3
4114
CA
4102
C12
4102
C13
MBD3D
[CB45]
MBD2D
[CB44]
MBD1D
[CB43]
MBD0D
[CB42]
MB'9
[CA54]
MB'10
[CA55]
MB'11
[CA56]
PRELOAD JUMPERS
CLIPPED OUT
1103
B1
4106
B2
4106
B3
4303
B4
OPG
ST
ACTIVE
[CA45]
OPG
SA
ACTIVE
[CA44]
TSAL
[CB33]
INT.
POT.

A



NOTE:
WIRING OF SA(ST CONTROL LEVELS TO TIME CONTROL INPUTS (EX. C1E, L,S,W) AND GATE CONTROL LEVEL OUTPUTS (EX: C1K, R,V,Z) IS ENUMERATED ON A SEPARATE SHEET

Fig. 3.19 OPG Logic and Decoding Matrix

B

The OPG SA and ST ACTIVE levels go through effectively three stages of buffering before they enter the decoding matrix proper. The first two stages are composed of fast 5 mc. type inverters to minimize delays. The function of these first two stages is two-fold. First, since the driving capability of the opg timing levels as they come from the central processor is insufficient to drive six 500 kc. inverters each at the end of the cable running from the PDP-1 to the racks, a non-inverting buffer is necessary. Second, the logical AND of the enabling TSAL with each of the opg timing levels must be implemented. Thus, this second stage of buffering acts not only as an inverter but also as half an AND gate. The OPG SA and ST ACTIVE levels each drive six 500 kc. inverters after the levels leave the first two stages of buffering. These inverters act as the third stage of buffering on the opg timing levels. Once the timing levels are fully buffered, their names change from $\overline{OPG} \cdot SA\ ACTIVE$ and $\overline{OPG} \cdot ST\ ACTIVE$ to SA_n and ST_n respectively, where n may be a, b, c, d, e, or f. SA_n and ST_n proceed into the decoding matrix as described below.

Each of the MB and MBD_D levels are doubly buffered before they enter the decoding matrix. Double buffering is required here for retention of the proper polarity. In the case of each of the MBD_D levels, there are two second stages of buffering. Each of these stages drives about half of the corresponding decoding matrix row since a single inverter could not drive the whole matrix row alone.

The decoding matrix itself consists of 36 three-input AND gates arranged in four sets of nine each. Some of the gates have a fourth input which is simply fed a constant assertive level (ground). Each of the AND gates requires three inputs:

1. a buffered MBD_D level,
2. a buffered MB bit level, and
3. a buffered opg timing level.

The first two of the three inputs select whether the given AND gate will pass the opg timing level when it occurs. The timing level input is determined by the type of analog gate that the output of the given AND gate drives.

If the indirect address bit MB_5 is a one for a given execution of opg, the storage gate T10 associated with the analog-to-digital converter and the ADC itself will be activated. Thus, MB_5 is ANDed with the ST opg timing level and then sent to control the ADC storage gate and the ADC. The ADC triggers on the trailing edge of the ST control level.

The pulsed analog running sum loop formed by T1 and T2 in Fig. 3.1 is controlled by a special circuit composed of a DEC 4303 integrating delay and a DEC 4110 diode gate The operation of the DEC 4303 is essentially as follows. The output is normally a logical zero. When the input rises from a zero to a one, the output follows and remains a one for a certain time after the input returns to zero. The time is set by the internal potentiometer setting and the choice of timing capacitor. Now, the overall circuit output is a one if the output of the DEC 4303 is a one and its input is a zero. The input to the special control circuit is the control level for the input storage gate T2 of the temporary storage gate pair. The output of this control circuit is the control level for the output gate T1 of the temporary storage pair. Thus, when some value is to be stored in the temporary storage pair, the input T2 storage gate is selected by an opg instruction. When the value of the output voltage of the input gate has settled and the input gate returns to its normal hold state, the output gate T1 is switched by the special control circuit to the track state and acquires the value of the voltage just stored in the input storage gate. Figure 3.20 illustrates the timing of the special control circuit.

The total opg decoding matrix capability of the PDP-1 is (8 x 9) 72 gates. However, for economic reasons, only half the total matrix (four sets of nine gates each) was implemented. The implemented set used decoded MBD_D levels 0 through 3.

Nothing has yet been said about the specific connection of opg timing levels to AND gate level inputs or the connection of AND gate outputs to analog gates. Clearly, these connections are highly problem-oriented. That is, for the F-100 problem, one particular combination of timing inputs and AND outputs was used. However, in general, for any other problem class, a different arrangement of these inputs and outputs would be preferable.

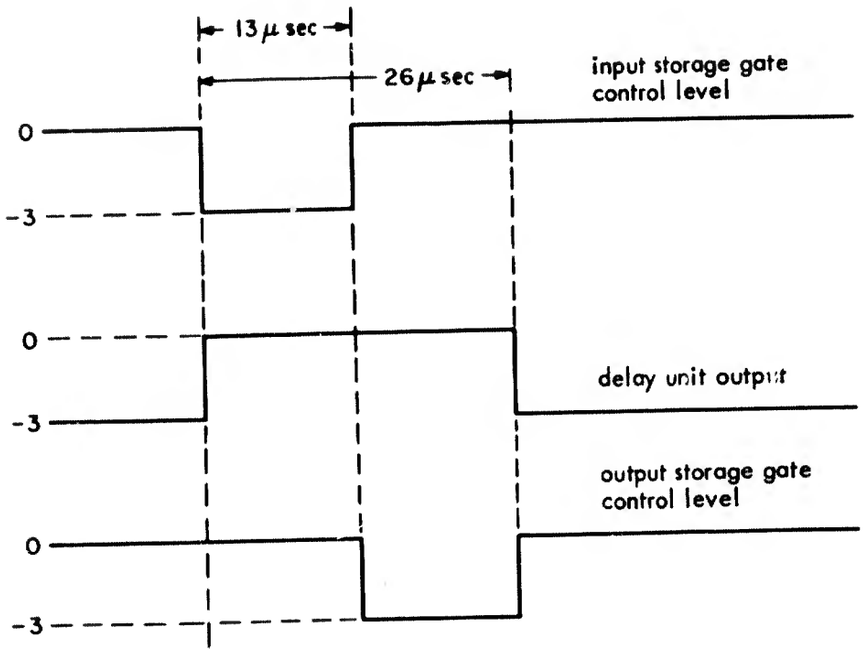


Fig. 3.20 Temporary Analog Storage Circuit Control Timing

In the F-100 simulation configuration, the use of each of the sample and storage gates is listed below.

Table 3.3

Analog Variables Assigned to Gates			
SAMPLE GATES			
S1	Output of T1	S10 δR	Rudder deflection
S2	Digital-analog converter buffer output	S11 δH	Horizontal stabilizer deflection
S3 α	Angle of attack	S12 pb/2V	Roll rate
S4 sin $\theta/2$	Rate of climb	S13 qc/2V	Pitch rate
S5 -V/2	True air speed	S14 rb/2V	Yaw rate
S6 cos $\phi/2$	Roll angle cosine	S15 αc/2V	Angle of attack rate
S7 $-\delta t/2$	Throttle	S16 -h/2	Pressure altitude
S8 β	Sideslip angle	(S17) $-a_z/2$	(Z acceleration)
S9 δA	Aileron deflection		
SAMPLE and HOLD GATES			
T1	Transfer from T2	T6 ΔL	Roll moment
T2	Temporary Storage	T7 ΔN	Yaw moment
T3 ΣF_x	X force	T8 ΔM	Pitch moment
T4 -T/m	Thrust/mass	T9	Special meter
T5 ΔF_z	Z force	T10	Analog-digital converter

The choice of assignments in the opg decoding matrix depends upon which analog gates must be activated simultaneously. Those gates which must be activated simultaneously must all appear in one group of gates. A group of gates consists of the nine gates which are all activated by an opg instruction having the same bit configuration in positions 6 through 8. An analog gate can be a member of more than one group of gates. However, membership in more than one group entails additional digital circuitry (opg OR circuits). Therefore, the number of gates belonging to more than one group should be minimized. Furthermore, members of a given group need not be required to be activated simultaneously. Thus, two small groups can be joined into a large group, as long as the total membership of a group does not exceed nine.

We shall omit the tedium of enumerating all the pairs and triples of gates which must be activated simultaneously in various parts of the F-100 hybrid simulation program. Table 3.4 gives the assignments of opg instruction groups and bits to the various sample and storage gates in the F-100 configuration. Note that gates S1, T2, and S2 are the only gates which are members of more than one group. The control levels for these analog gates come from OR circuits which logically

Table 3.4

OPG Bit Assignments

Decoded Instruction Bits 6 - 8	9 (400)	10 (200)	11 (100)	12 (40)	13 (20)	14 (10)	15 (4)	16 (2)	17 (1)
0	S1	T2	S8	S9	S10	S12	S14	T6	T7
1	S1	T2	S2	S11	S13	S15	T3	T5	T8
2	T4	T9	S2	S3	S4	S5	S6	S7	S16
3	(S17)								
4									
5		OPG decoding matrix No. 2 not implemented							
6									
7									

Note: 1 bit (10000) controls T10 and analog-digital converter

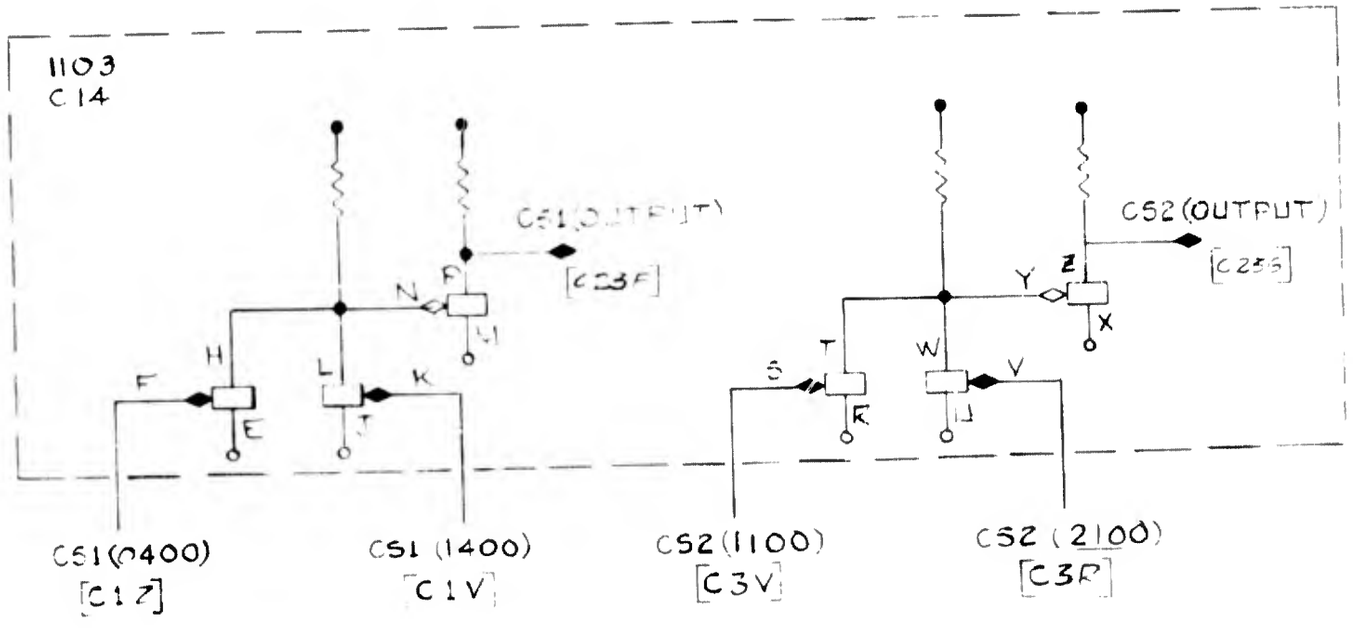
unite control levels from the opg decoding matrix. Figure 3.21 shows the block schematics of the three opg OR circuits necessary for the F-100 problem.

The matrix entry wiring for an analog gate requiring no OR circuit entails specifying the timing control level type and source location input to the matrix entry and the destination of the matrix entry output. The sources of timing control signals are the two sets of buffer transistors shown at the bottom of Fig. 3.19. Each transistor has the capacity to drive no more than seven timing level inputs to decoding matrix entries. This loading limitation must be taken into account when planning the setup of the matrix. The actual wiring lists for the decoding matrix for the F-100 problem are given in Tables 3.5 and 3.6. The module socket pin connections refer to Figs. 3.19 and 3.21.

H. X-Y Plotter Control

The two low order bits of the spare register drive two of the relays on the relay card. These two relays are used to control the pen and motors of an Electro-Instruments 500 X-Y plotter. The plotter, under digital control, was used to obtain the graphs of dynamic aircraft responses as simulated by the all-digital program. The analog inputs to the plotter originated from DAC's on the DEC-type 30 CRT display unit during initial runs. Later, when the Adcom unit was installed, two of the storage gates in the Adcom were used to drive the plotter analog inputs. The digital program used to control the plotter was PLOTEX, described in Chapter IV.

The control of the operation of the plotter pen and motors is as follows. When an ldl (IOT 3010) command is given with $AC_{17} = 0$, the motors are shut off. When $AC_{17} = 1$, the motors are turned on. If $AC_{16} = 0$ and the issuance of an ldl, the pen is raised. When $AC_{16} = 1$, the pen is lowered.



NOTE: LOAD RESISTOR JUMPERS ON PINS H & T ARE REMOVED.

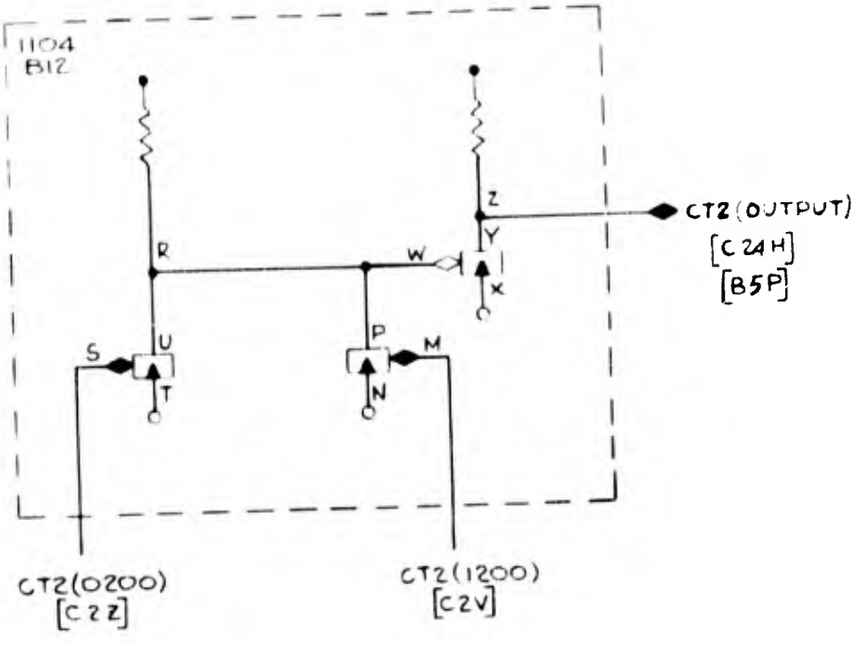


Fig. 3.21 OPG "OR" Circuit

Table 3.5

Sample Gate Control Level Wiring for F100 Configuration

Control Levels	Activating Instructions	Timing Level Source	Timing Level Input	Control Level Matrix Output	Destination	OR Circuit Output	Destination
CS1	OPG 0400	B3Z	C1W	C1Z	C14F	C14P	C23F
	OPG 1400	B3Z	C1S	C1V	C14K	↑	
CS2	OPG 1100	B3Z	C3S	C3V	C14S	C14Z	C23H
	OPG 2100	B3Z	C3L	C3R	C14V	↑	
CS3	OPG 2040	B3W	C4L	C4R	C23J	—	—
CS4	OPG 2020	B3W	C5L	C5R	C23K		
CS5	OPG 2010	B3P	C6L	C6R	C23L	—	
CS6	OPG 2004	B3P	C7L	C7R	C23M		
CS7	OPG 2002	B3P	C8L	C8R	C23N		
CS8	OPG 0100	B3Z	C3W	C3Z	C23P		
CS9	OPG 0040	B3W	C4W	C4Z	C23R		
CS10	OPG 0020	B3T	C5W	C5Z	C23S		
CS11	OPG 1040	B3W	C4S	C4V	C23T	—	—
CS12	OPG 0010	B3T	C6W	C6Z	C23U	—	—
CS13	OPG 1020	B3W	C5S	C5V	C23V	—	—
CS14	OPG 0004	B3T	C7W	C7Z	C23W		
CS15	OPG 1010	B3T	C6S	C6V	C23X	—	—
CS16	OPG 2001	B3P	C9L	C9R	C23Y	—	—
(CS17	OPG 3400	B3P	C1E	C1K	C23Z	—	)

NOTE:

A CABLE USING BLANK DEC MODULES AS CONNECTORS CARRIES THE CS SIGNALS FROM C23 to D1. THE CONNECTIONS CORRESPOND PIN-WISE (C23F to D1F).

Table 3.6

Sample and Hold Gate Control Level Wiring for F100 Configuration

Control Levels	Activating Instructions	Timing Level Source	Timing Level Inputs	Control Level Matrix Output	Destination	OR Circuit Output	Destination
CT1	(Delayed CT2)	B12Z B4U	B5P B5R	B5J	C24F	-	-
CT2	OPG 0200	B2Z	C2W	C2Z	B12S	B12Z	C24H B5P
	OPG 1200	B2Z	C2S	C2V	B12M	↑	-
CT3	OPG 1004	B2W	C7S	C7V	C24J	-	-
CT4	OPG 2400	B2Z	C1L	C1R	C24K	-	-
CT5	OPG 1002	B2W	C8S	C8V	C24L	-	-
CT6	OPG 0002	B2W	C8W	C8Z	C24M	-	-
CT7	OPG 0001	B2W	C9W	C9Z	C24N	-	-
CT8	OPG 1001	B2W	C9S	C9V	C24P	-	-
CT9	OPG 2200	B2Z	C2L	C2R	C24R	-	-
CT10	OPG 1	B1H	B5U	B5T	C22E C24S	-	-

NOTE:

A CABLE USING A BLANK DEC MODULE AS A CONNECTOR AT THE SOURCE END AND A 25 PIN AMPHENOL MINRAC 17 AT THE DESTINATION CARRIES THE CT SIGNALS FROM C24 TO THE TRIGGER INPUT SOCKET (J19) ON THE BACK OF THE ADCOM ANA-LOK HOUSING.

I. Digital-to-Analog Converter

The function of the digital-to-analog converter (DAC) is simply to convert a number in the PDP-1 accumulator to a voltage proportional to that number. The DAC used in the hybrid computer is composed of six DEC modules, four level amplifiers, a ladder network and an inverter. The block schematic of the DAC appears in Fig. 5.22.

The DAC is designed to work into a 2,000 ohm load and yields a ground voltage for a number $+3777_8$ in the high order 12 bits of the accumulator and -6.667 volts for a number of -3777_8 in the high order bits of the AC. Zero in the AC gives an output of -3.333 volts.

The level amplifiers convert the imprecise logic levels coming from the AC into fairly precise 0 to -10 volt levels to drive the ladder network. The level amplifiers actually put out voltages of approximately -0.002 and -9.998 volts for the nominal 0 and -10 volt outputs, respectively. These errors are caused by the saturation resistances of the transistor switches in the level amplifiers. These errors are compensated by the variable gain and offset of the follow-up Burr-Brown amplifier in the analog interface. Furthermore, the DAC follow-up amplifier resistance network is set up so that the final voltage range of the DAC buffered output is ±5 volts. A second Burr-Brown buffer amplifier produces a ±10 volt output.

The inverter on the sign feeding bit 12 of the ladder network is used so that the +0 and -0 in the 1's complement system yield the same output voltage from the DAC.

J. Interface Test Panel

The interface test panel was built to simulate some of the actions of the PDP-1 for off-line checking of the digital interface. The panel has the capability of generating a pair of pulses simulating the iot a and b pulses and a pair of levels simulating the opg timing levels. These two signals can be generated either in a "single cycle" mode under the control of a push button or repetitively under the control of a free running clock. Two 8-position rotary switches simulate two of the memory buffer decoders. There is a 25-pin Amphenol connector mounted on this panel to allow one of the discrete input switch registers to simulate one of the three major registers (AC, IO, MB) in the PDP-1. Figure 3.22 gives the circuit and layout details of the interface test panel.

K. Module Placement and Main Cables

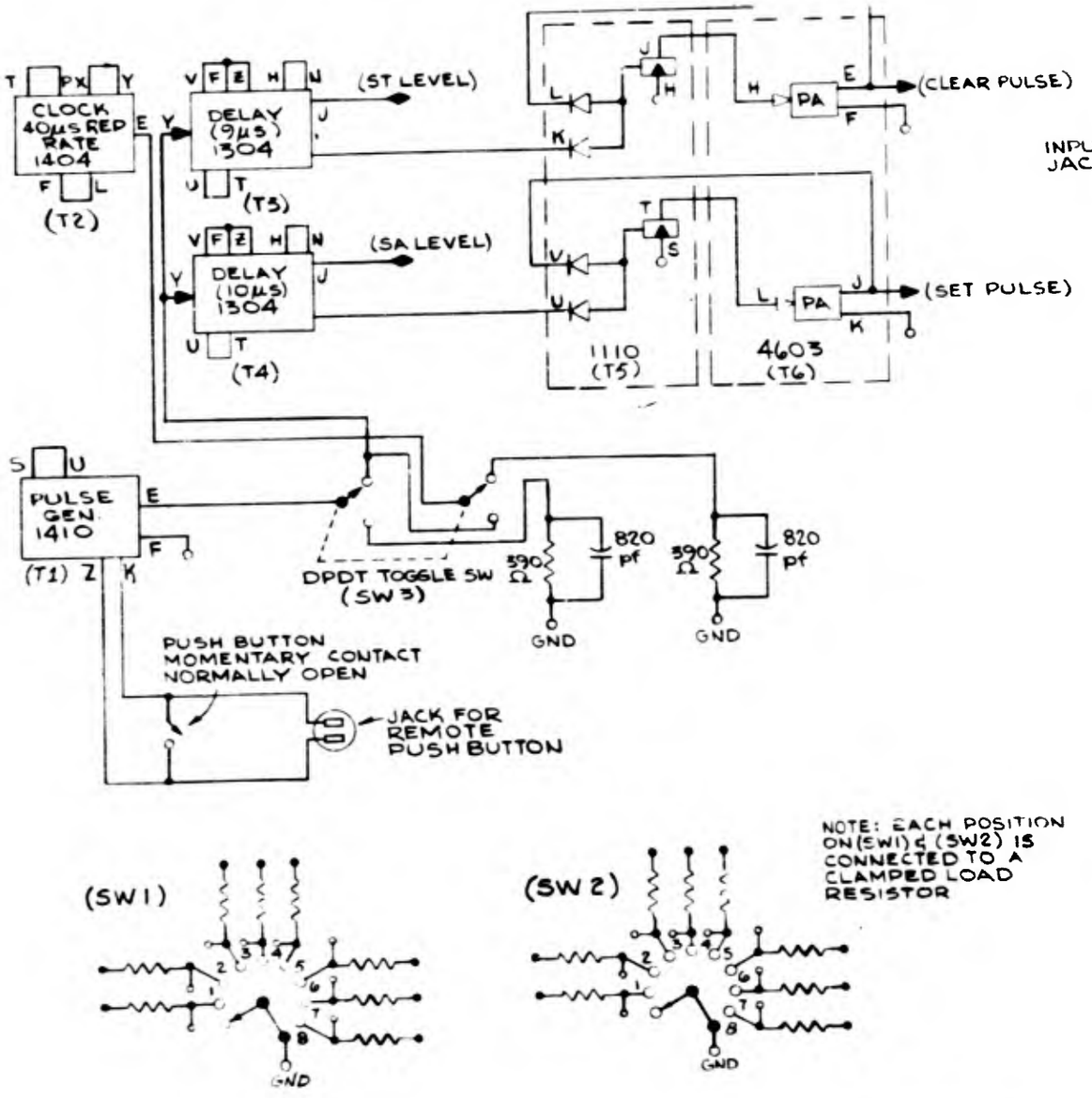
The actual module placement of the units described in Sections C through G is shown in Fig. 3.23. The two cables providing communication between the PDP-1 central processor and the digital interface are composed of 50 conductors; each conductor is individually shielded.

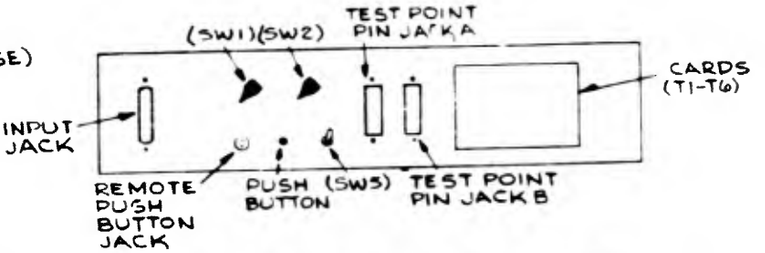
3.5 GENERATING RUNNING SUMS WITH THE PULSED ANALOG EQUIPMENT

In Section 3.3 the technique of modifying the linear aircraft by inserting nonlinear corrective terms into the analog computation on a sampled basis was described. At that time, it was simply stated that these corrective terms were generated in the digital and pulsed-analog equipment. However, until the functioning of the interface logic, particularly the operate gate (opg) instruction, had been outlined, it was not possible to go into further detail on this aspect of the hybrid system's operation. The presentation of the material in Section 3.4 has eliminated this obstruction, so we can now proceed with a more thorough discussion of a typical hybrid program sequence.

For convenience, a summary statement of all the special instructions added to the PDP-1 order code by the hybrid logic is given in Table 3.7.

To illustrate the operation of the pulsed-analog equipment generating the running sum of products, an excerpt from the actual hybrid program is listed in Table 3.8 with explanatory notes. This program fragment computes the first two terms of the roll correction signal given in Fig. 3.5. Neglecting the constant factor Sb, which is actually incorporated in the analog scaling, the desired correction is:





TEST PANEL - FRONT VIEW

TEST POINT PIN JACK A			
B	PIN 2 (INPUT JACK)	A	PIN 1 (INPUT JACK)
D	PIN 4 "	C	PIN 3 "
F	PIN 6 "	E	PIN 5 "
J	PIN 8 "	H	PIN 7 "
L	PIN 10 "	K	PIN 9 "
N	PIN 12 "	M	PIN 11 "
R	PIN 14 "	P	PIN 13 "
T	PIN 16 "	S	PIN 15 "
V	PIN 18 "	U	PIN 17 "
X	PIN 25 "	W	PIN 24 "
Z	GND "	Y	-3 VOLTS

TEST POINT PIN JACK B			
B	-15 VOLTS	A	+10 VOLTS
D	-3 VOLTS	C	-3 VOLTS
F	GND	E	GND
J	POS 2 (SW2)	H	POS 1 (SW2)
L	POS 4 "	K	POS. 3 "
N	POS. 6 "	M	POS. 5 "
R	POS. 8 "	P	POS. 7 "
T	POS. 2 (SW1)	S	POS. 1 (SW2)
V	POS. 4 "	U	POS. 3 "
X	POS. 6 "	W	POS. 5 "
Z	POS. 8 "	Y	POS. 7 "

Fig. 3.22 Interface Test Panel

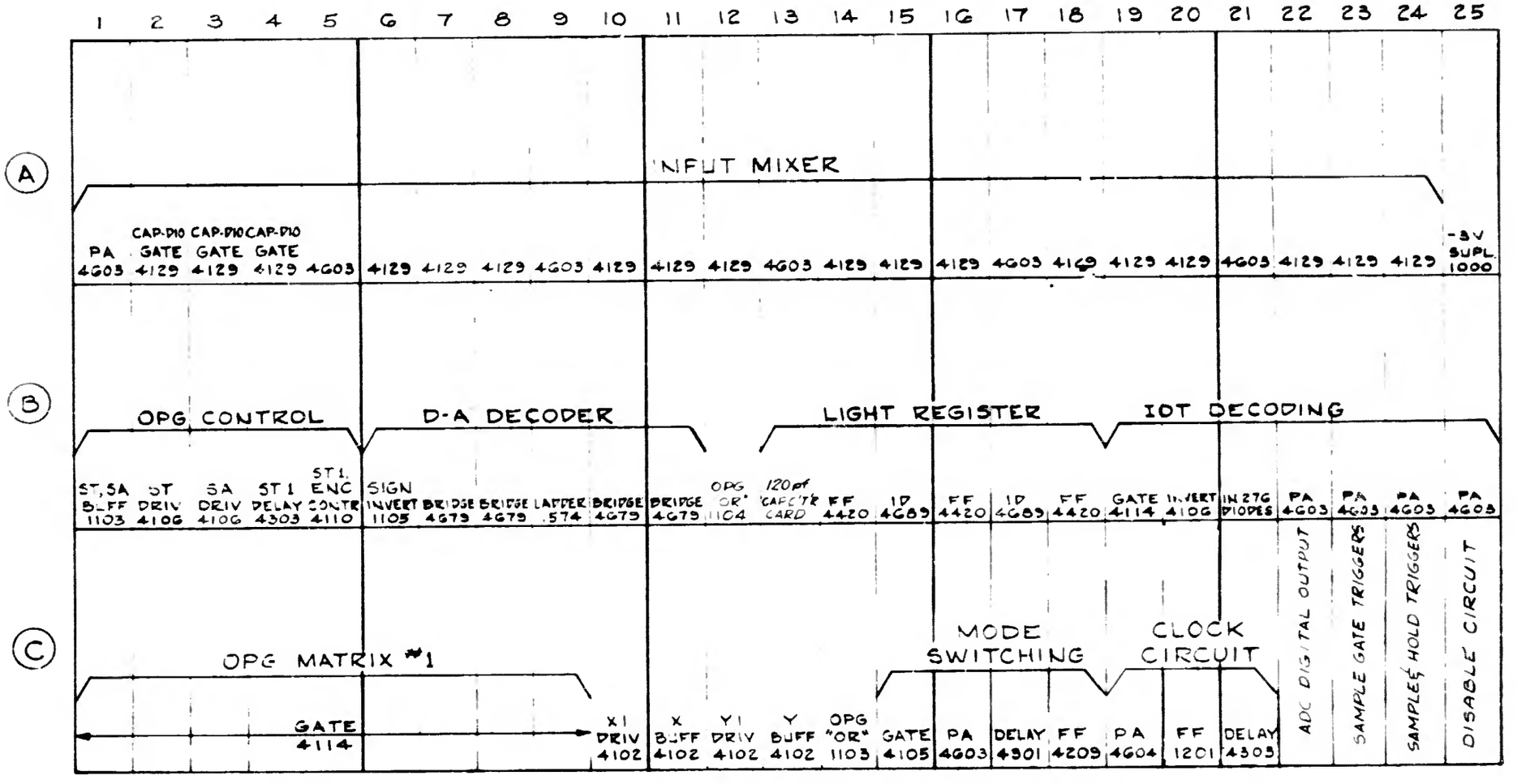


Fig. 3.23 DEC Module Placement, Hybrid Computer

Table 3.7

Special Hybrid Instructions

Instruction	Octal Code	Explanation	Operate Time (μsec)
opg xxxx	74 xxxx	operate gates specified by address xxxx as per Table 3.4	15.7
opg i xxxx	75 xxxx	operate gates specified by address xxxx plus gate T10 -- deactivation of T10 automatically starts ADC conversion process	15.7 (conversion continues for additional 50 μsec)
iot 2010	722010	toggle switch register $SRA_{0-17} \rightarrow IO_{0-17}$	5
iot 2110	722110	toggle switch register $SRB_{0-17} \rightarrow IO_{0-17}$	5
iot 2210	722210	toggle switch register $SRC_{0-17} \rightarrow IO_{0-17}$	5
iot 2310	722310	toggle switch register $SRD_{0-17} \rightarrow IO_{0-17}$	5
iot 2410	722410	lever switch register $LSR_{0-17} \rightarrow IO_{0-17}$	5
iot 2510	722510	toggle switch register $SP1_{0-17} \rightarrow IO_{0-17}$	5
iot 2610	722610	toggle switch register $SP2_{0-17} \rightarrow IO_{0-17}$	5
iot 2710 = enc	722710	encoder output $ADC_{0-12} \rightarrow IO_{0-12}$ clock state $\rightarrow IO_{17}$	5
iot 3010 = ldl	723010	load light register and spare register $IO_{0-17} \rightarrow LR_{0-17}$ $AC_{12-17} \rightarrow SR_{12-17}$ if $AC_{16} = 0$ x-y plotter pen raised $AC_{16} = 1$ pen lowered $AC_{17} = 0$ x-y plotter motor off $AC_{17} = 1$ motor on	5
iot 3110 = rsi	723110	reset analog integrators	5
iot 3210 = opi	723210	operate analog integrators	5
iot 3310 = hdi	723310	hold analog integrators	5
iot 3410 = zck	723410	zero clock flip-flop	5
iot 3510 = sck	723510	start clock timing	5

$$\left[\frac{C_{\ell_{\delta A}}(\alpha, Ma)\cdot AEA(Ma, q_1)}{I_x} - \frac{C_{\ell_{\delta A_o}}}{I_{x_o}}\right]\cdot \delta A$$

$$+\left[\frac{C_{\ell_{\delta R}}(|\alpha|)\cdot f_2(Ma) AER(Ma, q_1)}{I_x} - \frac{C_{\ell_{\delta R_o}}}{I_{x_o}}\right]\cdot \delta R \qquad (3.10)$$

To follow the illustrative program, it will be necessary to refer to the special hybrid instructions in Table 3.7, to the standard PDP-1 order code given in Appendix A, and to the hybrid system configuration shown in Fig. 3.1. In tracing through the program, it should be kept in mind that the summing amplifiers, inverters, and the GPS multiplier all invert the input signal (reverse sign). In addition, an awkward feature of the present system is the ±5 volt limit on the sample gates. As a consequence, scale factors of one half are introduced to reduce signal levels to ±5 volts before passage through a sample gate and scale factors of two are introduced to raise signal levels to ±10 volts after passage through a sample gate. This situation is hopefully temporary since the extraneous scale changing would be eliminated in a system having ±10 volt sample gates.

To simplify the illustrative program in Table 3.8, a harmless subterfuge has been perpetrated which should be noted. The aerodynamic coefficients $C_{\ell_{\delta A}}$ and $C_{\ell_{\delta R}}$ were originally specified in the stability axes. For the summation of roll moments in body axes, these coefficients are corrected for nonlinear effects and rotated by the digital program. The actual resultant values stored in memory are:

lea, $\quad C_{\ell_{\delta A}}(\alpha, Ma)\cdot AEA(Ma, q_1) - \alpha\cdot C_{n_{\delta A}}(\alpha)$

ler, $\quad C_{\ell_{\delta R}}(|\alpha|)\cdot f_2(Ma) AER(Ma, q_1) - \alpha\cdot C_{n_{\delta R}}(Ma) BER(Ma, q_1)$

The original motivation in performing the calculation of running sums with the pulsed-analog equipment was to avoid the frequent encoding of such multiplicative terms as δH, δR, δA, $\frac{pb}{2V}$, $\frac{rb}{2V}$, $\frac{qc}{2V}$, $\frac{\dot{\alpha}c}{2V}$, and β. These analog-to-digital conversions would be necessary if the running sums constituting the correction signals were computed in the PDP-1. A more basic motivation, perhaps, was to obtain experimental data on the characteristics of high-speed analog computing techniques as a state-of-the-art benchmark.

In retrospect, the use of pulsed analog equipment to form running sums was of marginal utility, the total saving in running time over digital calculations being less than one ms. Since the PDP-1 can perform other operations while waiting for an analog-to-digital conversion to be completed, the total time chargeable to each encode, say δA, would be:

opg i 0040	$\delta A \rightarrow$ T10, encode started	13 μsec
.....		
enc	$\delta A \rightarrow$ IO	5 μsec
dio y	$\delta A \rightarrow$ y	10 μsec
	Total	28 μsec

In each computation cycle, therefore, the total time "lost" in encoding the eight variables listed above would be only 224 μsec. Further time would be saved because of the parallel operation and faster multiply of the pulsed analog configuration, but clearly the pulsed analog capability is not an essential feature of the present hybrid aircraft simulator design.

Table 3.8

Illustrative Hybrid Program
Generating Two Terms of Sampled Correction Term in Roll

Instruction	Explanation
lac lea	$C_{l_{\delta A}}(a, Ma) \cdot AEA(Ma, q_1) \rightarrow AC$
mul kk6	$\frac{1}{I_x} \cdot C_{l_{\delta A}}(a, Ma) \cdot AEA(Ma, q_1) \rightarrow AC$
sub lza	$\frac{C_{l_{\delta A}}(a, Ma) \cdot AEA(Ma, q_1)}{I_x} - \frac{C_{l_{\delta A_o}}}{I_{x_o}} \rightarrow AC$
opg 240	Activate Gates T2 (T1) and S9
	$-\left(\frac{C_{l_{\delta A}}(a, Ma) AEA(Ma, q_1)}{I_x} - \frac{C_{l_{\delta A_o}}}{I_{x_o}}\right) \cdot \delta A \rightarrow T1$
lac ler	$C_{l_{\delta R}}(\lvert a \rvert) \cdot f_2(Ma) AER(Ma, q_1) \rightarrow AC$
mul kk6	$\frac{1}{I_x} \cdot C_{l_{\delta R}}(\lvert a \rvert) \cdot f_2(Ma) \cdot AER(Ma, q_1) \rightarrow AC$
sub lzr	$\frac{C_{l_{\delta R}}(\lvert a \rvert) \cdot f_2(Ma) AER(Ma, q_1)}{I_x} - \frac{C_{l_{\delta R_o}}}{I_{x_o}} \rightarrow AC$
opg 620	Activate Gates S10, S1, and T2 (T1)
	$-\left(\frac{C_{l_{\delta A}}(a, Ma) AEA(Ma, q_1)}{I_x} - \frac{C_{l_{\delta A_o}}}{I_{x_o}}\right) \cdot \delta A$
	$+\left(\frac{C_{l_{\delta R}}(\lvert a \rvert) \cdot f_2(Ma) AER(Ma, q_1)}{I_x} - \frac{C_{l_{\delta R_o}}}{I_{x_o}}\right) \cdot \delta R \rightarrow T1$
-------	Continue to generate remaining terms of Roll correction, depositing final running sum in T6
Previously calculated functions stored in memory	
lea,	$C_{l_{\delta A}}(a, Ma) \cdot AEA(Ma, q_1)$
kk6,	$\frac{1}{I_x}$
lza,	$\frac{C_{l_{\delta A_o}}}{I_{x_o}}$
lzr,	$\frac{C_{l_{\delta R_o}}}{I_{x_o}}$
ler,	$C_{l_{\delta R}}(\lvert a \rvert) \cdot f_2(Ma) AER(Ma, q_1)$

BLANK PAGE

CHAPTER IV

DIGITAL AND HYBRID PROGRAMS

4.1 ALL-DIGITAL PROGRAM

(A) Introduction

The all-digital F-100A simulation program was written primarily to provide a reference with which to compare the hybrid F-100A simulation program. The three major areas of comparison are volume of storage required, running time per iteration, and minimum iteration rate necessary to achieve transient response fidelity. Secondly, substantial parts of the all-digital program were transferred directly to the hybrid program, since those sections of the F-100A model handled purely in the digital domain remained essentially unchanged. Finally, the all-digital program provided a preliminary check on the aircraft data used for the F-100A model.

The all-digital F-100A simulation program is composed of routines and subroutines which perform the following general functions:

(a) Nonlinear function generation

(b) Integration

(c) Decisions

(d) Linear algebraic computations

Routines to generate nonlinear functions of one and two variables use tables containing only function values at selected breakpoints, with intermediate values being obtained by linear interpolation. Level (zone) selection and slope computation routines for each independent variable are isolated from specific function generation routines. Only one level selection is performed on a given independent variable during each cycle of computation regardless of the number of functions depending on the variable since a common set of breakpoints are employed for all functions of a given variable.

Integration is handled by three different methods. Most integrations use a subroutine based on the trapezoidal formula. Roll (ϕ) and heading (ψ) angles are generated as the corresponding sines and cosines by a digital Gilbert-Howe algorithm.[58] Finally, in the engine routines, ice accumulation and mass of fuel employ simple rectangular integration.

Decisions and linear algebraic computations are handled in a generally straightforward manner which will be discussed later in detail

To give some idea of the programming and data analysis effort involved, the following approximate schedule is presented. Preliminary data reduction from North America,[16] UDOFT[17] and M.I.T. Electronic Systems Laboratory[18] data began in June, 1962. Subsequently, the programming necessary to simulate the longitudinal mode aerodynamic performance of the F-100A aircraft as a function of thrust, horizontal stablilzer defection and an aggregate of discrete inputs was undertaken. This task was completed by April 1963, after which data was processed and a program written for the lateral mode characteristics. In September, 1962, the programming of the Melpar-UDOFT engine model commenced based on Sylvania's UDOFT flow charts and Melpar's function data sheets. The engine program was essentially complete by July, 1963, at which time the integration of the longitudinal and lateral mode aerodynamic programs with the engine program began. For September, 1963, through March, 1964, a large number of mildly subtle bugs were worked out of the overall all-digital simulation program and transient and steady-state performance tests were run on the programmed aircraft model. The hybrid program evolved from the all-digital program in early 1964 with final testing completed by June, 1964.

It should be noted that the two people primarily responsible for the data analysis and programming were simultaneously involved in parallel activities (course work, theses, logical design, and test). Moreover, it is clear that any subsequent simulation of another aircraft could be programmed much more expeditiously than was the case on this baptismal effort.

(B) Program

Organization: The all-digital simulation program was broken down into three major parts, aero part 1, the engine program, and aero part 2. The 4,096 word core memory of the MIT Electrical Engineering Department PDP-1 was somewhat inadequate to store the entire F-100A all-digital program. Fortunately, an auxiliary magnetic drum memory was available having 22 fields of 4,096 words each Parts of the simulation program were stored on the drum and read into core memory when requir d. The data, the function tables, and aero part 1 remained in core at all times when the program was running. The engine routine and aero part 2 were each read into an area of core called the "swapped" area once each cycle. An initialization routine to set initial values of variables was called from the drum into the swapped area when necessary. An output routine, coded especially for the F-100A simulation program, was also available from the drum and was brought into the "swapped" area at the operator's option. Any changes in the simulation that were necessary were made on the binary version of the program through the use of a symbolic debugging program called D^3T or Drum Digital Debugging Tape. Entry into D^3T was by depressing a key. Finally, an X-Y plotter routine called poltex was available off the drum for plotting transient responses of the programmed aircraft model. Each block of the simulation program will be discussed individually later in this chapter.

(C) Function Generation

Most of the aerodynamic coefficients governing the transient and steady state response of the F-100A aircraft are non-linear functions of one of thirteen independent variables such as Mach number, altitude, and dynamic pressure. In some cases, coefficients are functions of two such variables. In both the digital and hybrid simulations, non-linear functions are computed by linear interpolation between breakpoint values stored in a table. A given independent variable need not have its breakpoints evenly spaced. In most cases, the breakpoints are unevenly distributed in order to achieve the best fit with the function set. All functions of a given variable have the same set of breakpoints.

Level Selection and Functions of One Variable. The well-known formula is used for computing functions of one variable.

$$f(X_i) = \frac{X_i - X_{n-1}}{X_n - X_{n-1}} [f(X_n) - f(X_{n-1})] + f(X_{n-1}) \qquad (4.1)$$

Each of the variables is described in Fig. 4.1.

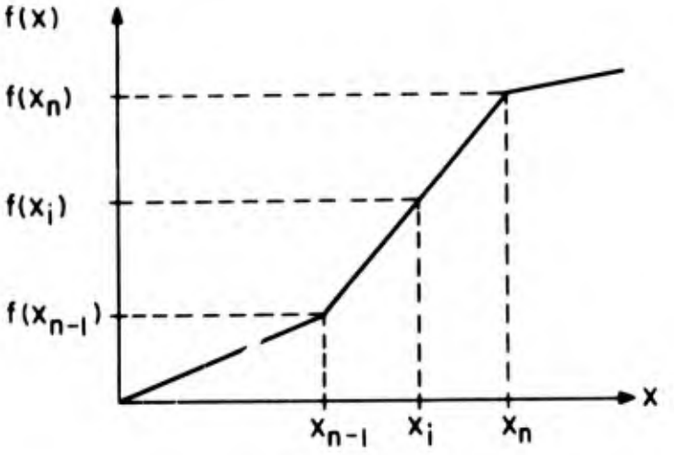


Fig. 4.1 Function of One Variable Breakpoint Convention

The actual computation of the function value in a given cycle of computation is broken into two parts. Prior to the computation of any functions of a given variable, that variable is "level selected". The level selection process consists essentially of two parts. First, the index n is determined, and second, the "slope" $\frac{X_i - X_{n-1}}{X_n - X_{n-1}}$ is computed. The index determination is carried out by repetitive application of the inequality 4.2 until some pair (X_n, X_{n-1}) are found to satisfy the inequality.

$$X_{n-1} \leq X_i < X_n \qquad (4.2)$$

The actual process depends upon having a list of breakpoints (X_j) ordered from highest to lowest. Each X_j is then tested in descending order to see whether Eq. 4.3 holds.

$$X_i < X_j \qquad (4.3)$$

The first X_j which does not satisfy Eq. 4.3 corresponds to X_{n-1} and the X_j next highest on the list (the breakpoint previously tested) corresponds to X_n. Once the index n is known, it is a trivial job to compute the slope, given the table of breakpoints and the value of X_i. Level selection is one of the major contributors to running time, each such operation requiring 215 + 40K microseconds, where K is the number of breakpoint comparisions necessary before the proper zone is located. This timing includes the slope calculation.

Let us emphasize that once the index n and the slope $\frac{X_i - X_{n-1}}{X_n - X_{n-1}}$ are determined for the independent variable X_i, they are used to compute all the nonlinear function values of that variable in a given solution cycle. For example, three separate functions of X_i would be generated by Eqs. 4.4a, 4.4b, and 4.4c.

$$f(X_i) = \frac{X_i - X_{n-1}}{X_n - X_{n-1}} \left[f(X_n) - f(X_{n-1}) \right] + f(X_{n-1}) \qquad (4.4a)$$

$$g(X_i) = \frac{X_i - X_{n-1}}{X_n - X_{n-1}} \left[g(X_n) - g(X_{n-1}) \right] + g(X_{n-1}) \qquad (4.4b)$$

$$h(X_i) = \frac{X_i - X_{n-1}}{X_n - X_{n-1}} \left[h(X_n) - h(X_{n-1}) \right] + h(X_{n-1}) \qquad (4.4c)$$

Obviously, a separate table of function values is needed to compute each function.

The main drawback to using the same set of breakpoints for all functions of a given variable is in determining a common breakpoint placement that is suitable for all the functions involved. Non-linear functions of some variables have common characteristics. For example, functions of Mach number tend to vary most abruptly in the vicinity of Mach 1.0; hence, more breakpoints were placed in the transonic range than in either the subsonic or the supersonic ranges. The Mach breakpoints finally chosen were 0.0, 0.2, 0.8, 0.9, 0.95, 1.05, 1.4, and 1.7. On the other hand, where functions of a given variable had few common characteristics, the breakpoints were spaced more uniformly, as in the case of altitude. The altitude breakpoints chosen are essentially standards in the aircraft industry for data: 0, 10, 15, 25, 35, 45, and 60 thousand feet.

The level selection, function of one variable, and function of two variable subroutines are open subroutines. Each time a computation of the given form is required during the execution of the program, the proper sequence of instructions are coded into the program. A closed subroutine is one in which all computations of a given form are performed by the same set of instructions in some remote location. The data input and output and return location are communicated to the closed subroutine through its calling sequence.

The level selection subroutine operates on a table of data containing (a) the value of the independent variable under consideration, (b) the number of breakpoints used in functions of the variable, (c) the ordered list of independent variable breakpoint values, and (d) locations for the zone index and for the computed slope. Thus, in the Midas assembler language (which is described in Appendix A), a call for the open subroutine for level selection (named "levsel") contains only the symbolic location for the first entry in the table and the number of independent variable breakpoints in the list. For example, the expression "levsel pal, 7" in the program calls for a predetermined sequence of thirty instructions to be carried out level selecting the independent variable pressure altitude. The first entry in its table of seven breakpoints and calculated parameters is at the symbolic address "pal". The slope $\frac{h - h_{n-1}}{h_n - h_{n-1}}$ is also computed by the sequence of instructions. A detailed flow diagram of the levsel subroutine appears in Fig. 4.2.

After completion of the level selection and slope generation procedure, the technique for generating functions of one variable is quite straightforward. The flow diagram of the open subroutine which generates functions of one variable is shown in Fig. 4.3. The subroutine occupies 10 registers in memory and requires 125 μ seconds to carry out.

Functions of Two Variables: The technique for generating functions of two variables is a logical extension of the one variable technique. In the one variable case, the value of the independent variable (X), lay on a line segment (X_{n-1}, X_n). In the two variable case, the values of the independent variables (X, Y) lie in a rectangle $(X_{n-1}, X_n), (Y_{p-1}, Y_p)$. To obtain F(X, Y), three linear interpolations indicated in Eqs. 4.5a, 4.5b, and 4.5c are performed.

$$f(X, Y_{p-1}) = \left[f(X_n, Y_{p-1}) - f(X_{n-1}, Y_{p-1}) \right] S_x + f(X_{n-1}, Y_{p-1}) \qquad (4.5a)$$

$$f(X, Y_p) = \left[f(X_n, Y_p) - f(X_{n-1}, Y_p) \right] S_x + f(X_{n-1}, Y_p) \qquad (4.5b)$$

$$f(X, Y) = \left[f(X, Y_p) - f(X, Y_{p-1}) \right] S_y + f(X, Y_{p-1}) \qquad (4.5c)$$

S_x and S_y are the slopes computed by the level selection subroutines for the variables X and Y respectively. Figure 4.4 gives the detailed flow diagram for the generation of functions of two variables. Each such subroutine occupies 22 memory registers and consumes 260 μ seconds of running time.

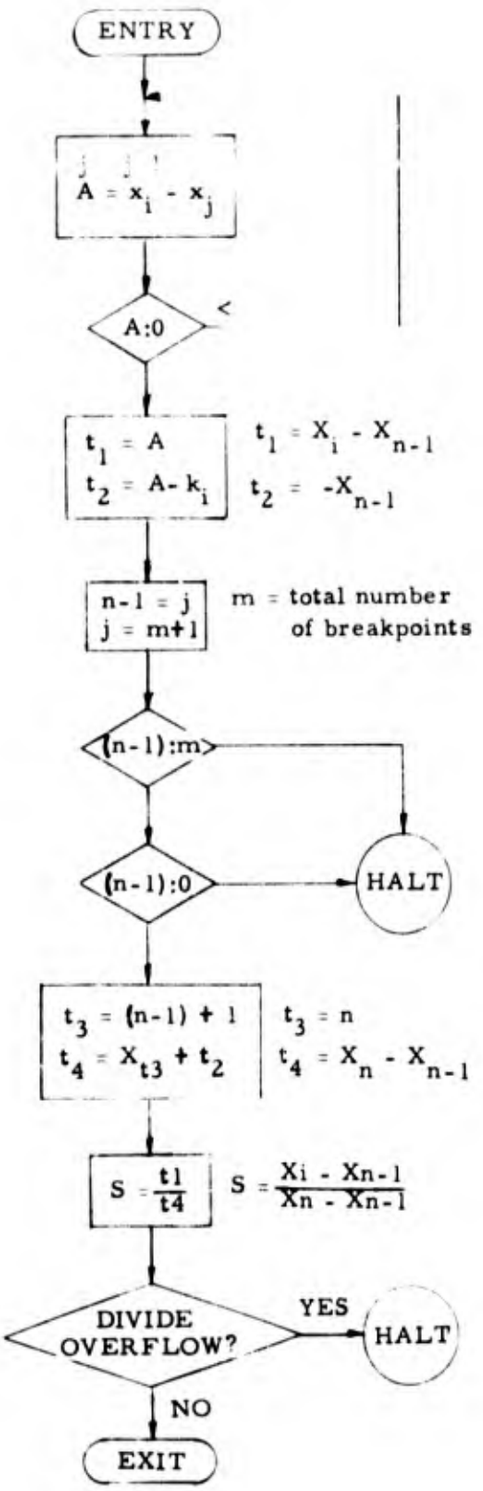


Fig. 4.2 Level Selection and Slope Computation Open Subroutine Flow Diagram

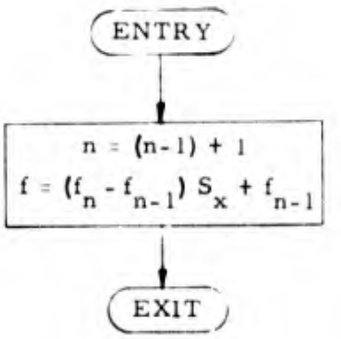


Fig. 4.3 Open Subroutine for Generating Functions of One Variable

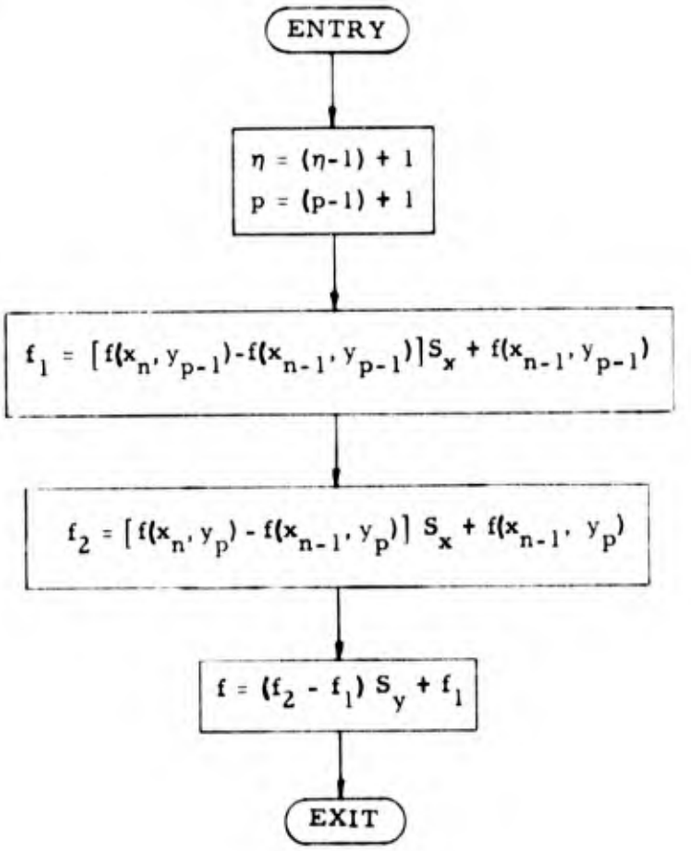


Fig. 4.4 Open Subroutine for Generating Functions of Two Variables

(D) Integration

In the all-digital F-100A simulation, three integration methods are used. These methods are rectangular, trapezoidal, and a digital Gilbert-Howe method for obtaining sines and cosines of angles from angular rates. The choice of method in a particular application was a compromise between speed and accuracy.

Rectangular Formula: The applications where the rectangular method was used consisted in relatively non-critical computations such as fuel flow integration to account for fuel depletion and aircraft mass change, and ice accumulation on engine air intake guide vanes to account (mainly) for loss of power due to icing. A further explanation of the fuel flow integration is merited due to the tremendous dynamic range of the mass relative to the fuel increment over 50 msec intervals. The integration for mass of fuel is done in two stages. The first stage consists in integrating the fuel flow rate using a scaling such that the minimum discernable rate is 2^{-17} lbs/50 msec or about .055 lbs/hr. When a one-pound increment or decrement of fuel has been accumulated by this summation process, the "fuel tank" (register containing the mass of fuel remaining) is incremented or decremented by one pound. In essence, a double-length word is being employed to handle the large dynamic range from the maximum total mass of fuel to the minimum increment or decrement in mass that occurs during one solution cycle. A more complete discussion of scaling and word length requirements is presented in References 7 and 11.

Trapezoidal Formula: All the integrations involved in the airframe dynamics, with the exception of ϕ(roll angle) and ψ (heading angle), are computed using the trapezoidal formula given in Eq. 4.6.

$$Y_{n+1} = Y_n + \frac{h}{2}(3\dot{Y}_n - \dot{Y}_{n-1}) \tag{4.6}$$

A closed subroutine is used to perform these integrations. Roundoff error is essentially eliminated by using a double precision technique which saves the low-order half of the double-length product formed in the multiplication. Some roundoff error occurs due to the limited word length of the time step h and of the rates themselves.

A further double precision technique is required in the cases of two variables having wide dynamic ranges relative to their rates of change. These variables are airspeed and altitude. The following criterion was adopted. If the derivative of the variable ($\dot{V}$ and rate of climb, respectively) is non-zero in the machine, then the value of the variable should change. The 18-bit word length of the machine is about four bits too short to implement the above criterion directly, since the integration subroutine does not change the scaling of the rate when converting it to an increment. A method was devised in which an intermediate variable supplied the needed bits, with the intermediate variable being operated upon by the integration subroutine. When the value of the intermediate variable reaches a sufficent magnitude to affect the main variable, the correct value is added to the main variable and a corresponding value subtracted from the intermediate variable value.

Digital Gilbert-Howe Method: The roll and heading angles may take on any value from 0 to 2π radians in the model chosen for this simulation. Furthermore, the values of these angles are of no direct use in any computation involved in this simulation. The sines and cosines of these angles are of considerable importance, however, both computationally and in terms of driving indicators. Hence, the decision was made to integrate the angular rates directly to form the respective sine-cosine pairs. The method chosen was Massey's extension[6] of an analog technique developed by Gilbert and Howe at the University of Michigan.[58] The original Gilbert-Howe differential equations are given in Eqs. 4.7a and 4.7b.

$$\frac{d}{dt}(\text{Cos }\theta) = -\text{Sin }\theta \cdot \dot{\theta} - \mu\epsilon \text{ Cos }\theta \tag{4.7a}$$

$$\frac{d}{dt}(\text{Sin }\theta) = \text{Cos }\theta \cdot \dot{\theta} - \mu\epsilon \text{ Sin }\theta \tag{4.7b}$$

where $$\epsilon = \text{Sin}^2\theta + \text{Cos}^2\theta - 1$$

and μ = a feedback term governing the rate of an exponential decrease on ϵ.

In his work in this area, Massey showed that Eqs. 4.8a and 4.8b are the difference equations equivalent to Gilbert and Howe's differential equations.

$$\text{Cos }\theta_{n+1} = \text{Cos }\theta_n - \text{Sin }\theta_n \cdot \dot{\theta}_n \cdot \Delta t - \mu\epsilon \text{ Cos }\theta_n \Delta t \tag{4.8a}$$

$$\text{Sin }\theta_{n+1} = \text{Sin }\theta_n + \text{Cos }\theta_n \cdot \dot{\theta}_n \cdot \Delta t - \mu\epsilon \text{ Sin }\theta_n \Delta t \tag{4.8b}$$

where $$\epsilon = \text{Sin}^2\theta_n + \text{Cos}^2\theta_n - 1$$

and μ is a feedback factor.

Massey further showed that the product $\mu \Delta t$ has an optimum value of 1/2 for the case $\Delta t \neq 0$.

For the purpose of this simulation, the results obtained from Eqs. 4.8a and 4.8b were sufficient. Only when the roll rate approached the structural limit of the aircraft (10 rad/sec or 1/2 rad/ solution), did the equation pair produce results in error by about 10 percent.

(E) Decisions

In the all-digital simulation of the F-100A, the effect of a number of cockpit controls and control system functions are (or are idealized to be) discrete, i.e., on-off. These effects are represented as branch points in the simulation program. In this section, discrete cockpit control panel inputs and outputs are discussed and the classes of decisions used in this program are classified.

Arithmetic Decisions: The PDP-1 instruction code contains no conditional jump (transfer) instructions. All branching is done with a micro-programmable skip instruction in conjunction with an unconditional jump instruction. The nine conceivable arithmetic comparisons which can be made are tabulated in Fig. 4.5. In certain cases when c(x) is a constant, the sample instruction sequences

lac x sas y jmp z → 25 μsec c(x) ≠ c(y) 20 μsec ↓ c(x) = c(y)	lac x sad y jmp z → 25 μsec c(x) = c(y) 20 μsec ↓ c(x) ≠ c(y)
lac x sub y spq jmp z → 30 μsec c(x) ≤ c(y) 25 μsec ↓ c(x) > c(y)	lac x sub y szm jmp z → 30 μsec c(x) > c(y) 25 μsec ↓ c(x) ≤ c(y)
lac x sub y spa jmp z → 30 μsec c(x) < c(y) 25 μsec ↓ c(x) ≥ c(y)	lac x sub y sma jmp z → 30 μsec c(x) ≥ c(y) 25 μsec ↓ c(x) < c(y)
lac x 20 μsec sza jmp z → c(x) ≠ 0 15 μsec ↓ c(x) = 0	lac x sza i jmp z → 20 μsec c(x) = 0 15 μsec ↓ c(x) ≠ 0
lac x lio x spa 20 μsec spi jmp z → c(x) < 0 jmp z ↓ 15 μsec c(x) ≥ 0 ↓	lac x lio x sma 20 μsec spi i jmp z → c(x) ≥ 0 jmp z ↓ 15 μsec c(x) < 0 ↓

Fig. 4.5 Arithmetic Comparisons

may be modified slightly to save both computation time and memory space. The saving is gained by using the PDP-1 law or load accumulator with (immediate) instruction. An example follows.

Using lac	Using law
lac X	law constant
sub Y	sub Y
sma	sma
jmp z → 30 μs	jmp z → 25 μs
.	.
.	.
.	.
25 μs	25 μs
running time	running time
X, constant	

The saving using the law instruction amounts to 5 μsec/decision and one register regardless of whether the jump is or is not performed. One use of this technique affords no significant saving. However, with repeated use, a considerable advantage may result. The restriction that the constant must be less than 4096 hampers the application of this technique in less than 25 percent of the possible cases.

Discrete Inputs: A panel containing 7 rows of 18 switches each was built to provide a flexible discrete input capability for the PDP-1 computer as applied to the F-100A simulation problem. The interface logic for the panel was arranged so that when one of seven iot (input-output transfer) commands is issued, the state of one of these rows of switches is placed in the input-output register (IO) on a bit-for-bit basis. The bit of the IO corresponding to an on switch would be set to a one. The off switches cause corresponding IO bits to be set to zero.

Once the state of the row of switches is in the IO register, it is a relatively simple matter to branch the program according to individual switches. An instruction sequence which tests the fourth switch of the second row would be as follows:

```
(5μsec)  srb     /iot 2110, switch row b
(5μsec)  ril 3s  /rotate bit 3 (or 4th bit) into sign position
(5μsec)  spi     /skip on + IO (or zero in sign position)(switch off)
(5μsec)  jmp z   /go to "z" if switch on
         xxx     /next instruction executed if switch off
```

Over 100 sequences such as this occur in the all-digital F-100A simulation. From the programmer's viewpoint, a more convenient way of handling the discrete input switches would be to implement a class of skip commands such that each skip would refer to a single switch. If such a command were available, the program sequence for testing the 4th switch of the second row of switches would be:

```
(5μsec)b04     /skip if switch 4 row b is zero
(5μsec)jmp z   /go to z if switch is on
       xxx     /next instruction
```

Obviously, the number of instructions required to execute this function with the new instruction has been halved and the execution time has been either halved or cut by a factor of three, depending on the chosen branch.

Discrete Outputs: A panel of 18 lights, each of which is driven by a flip-flop, is the implementation of the discrete output indicators for the F-100A simulation on the PDP-1. The lights are not individually controllable but must be set up as a group. The lights are in an on or off state according to the bits of the IO register of the PDP-1 when the iot command controlling the loading of the light flip-flop register is issued. If the bit corresponding to a given light is a one at the time of the issuance of the iot, then the light is turned on; if that bit is a zero, then the light is turned off.

A core memory register is reserved to contain the desired state for all the discrete output lights and its contents are transferred into the light register once each cycle. In this manner, discrete outputs may be set at any point in the program, yet the results of decision-making computations are available in memory for succeeding portions of the program to test and branch on. In the present F-100 simulation program, a typical coded sequence to set bit 5 in the discrete output word on the basis of an arithmetic decision would be as follows:

```
                    .
                    .
lac varbl       /c(varbl) →AC              (10 μ sec)
sub const       /c(AC) - c(const)→ AC      (10 μ sec)
lia             /c(AC)→ IO                 ( 5 μ sec)
lac disout      /c(disout) →AC             (10 μ sec)
and (-10000)    /0 →AC5                    (10 μ sec)
spi i           /skip if C(IO) < 0         ( 5 μ sec)
ior (10000)     /1 →AC5                    (10 μ sec)
dac disout      /c(AC) →disout             (10 μ sec)
                    .
                    .
```

totals: 70 μ sec, o registers

where: "disout" contains the discrete output word
"varbl" contains some variable
"const" contains some constant
and $c(disout)_5$ is set if $c(varbl) \geq c(const)$

One suggested alternative to reduce the number of instructions necessary to produce a given discrete output entails implementing separate set and clear instructions for each bit of the discrete output register. A program sequence corresponding to the example above, using the "new" instructions, would be as follows:

```
                    .
                    .
                    .
lac varbl       /c(varbl)→AC               (10 μsec)
sub const       /c(AC) - c(const)→AC       (10 μsec)
cl05            /clear D.O. bit 5          ( 5 μsec)
sma             /skip if c(AC) 0           ( 5 μsec)
st05.           /set D.O. bit 5            ( 5 μsec)
                    .
                    .
```

totals: 35 μ sec, 5 registers

The availability of these instructions would save about 35 percent of the number of instructions used for discrete output sequences and reduce discrete output running time by 50 percent.

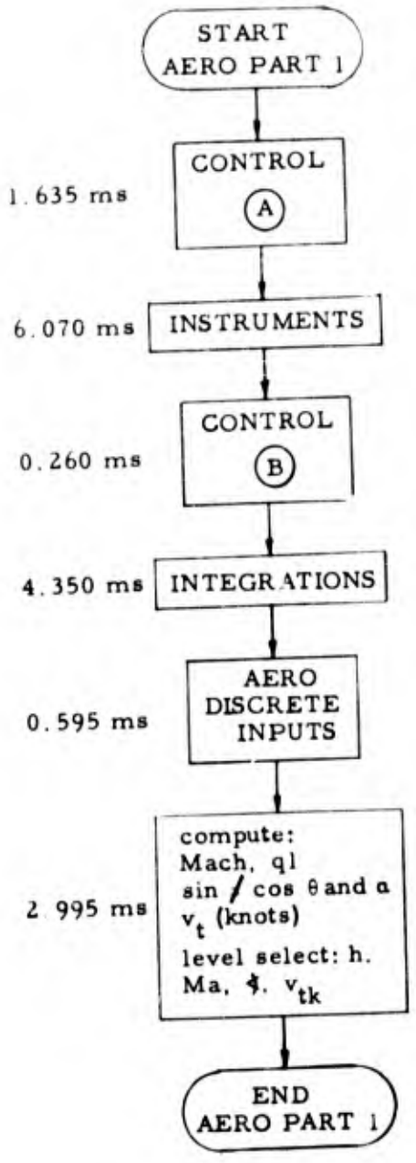
(F) Basic Program

The all-digital F-100A simulation program is divided into three main blocks. These are tagged Aero Part 1, Engine, and Aero Part 2. The gross flow diagram of each of these blocks is given in Figs. 4.6a, 4.6b, and 4.6c. Due to core memory size restriction, not all the program resides in core simultaneously. That part of the simulation program not in core memory at a given time is available from an auxiliary magnetic drum storage unit. Data and variables remain in core at all times when the simulation program is running. Furthermore, Aero Part 1 remains in core continuously to minimize the volume of traffic between the core memory and the drum memory.

Certain additional routines have been written to support experimentation with the program which are not strictly elements of the simulation in that they perform input-output and initialization functions unnecessary in the normal running of the simulation program. Included in these service routines are (a) a symbolic debugging routine (DDDT) (b) a flexible cathode ray tube/typewriter/punched paper tape arithmetic output routine, (c) an X-Y plotter output routine, and (d) an initialization routine.

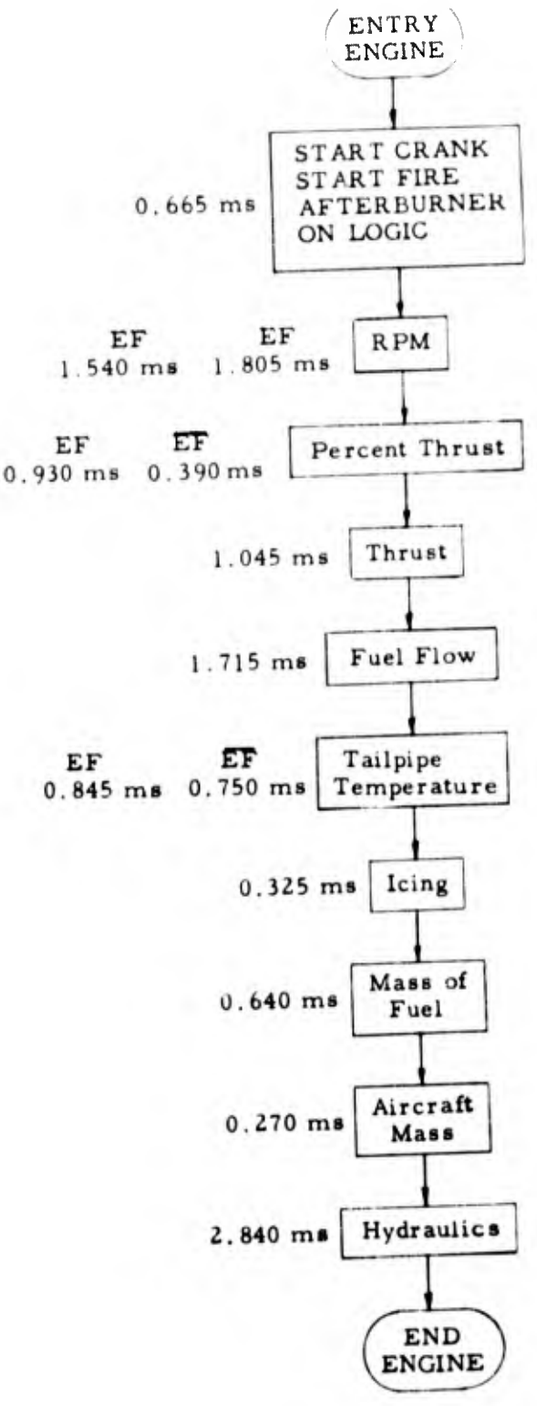
Timing: Normally, in a simulation such as this, cockpit instruments are handled on an analog output basis. During the early development of the all-digital program, no analog output equipment was available, hence, instruments were simulated by bar meters and rotating lines displayed on the cathode-ray oscilloscope. These simulated pilot displays are shown in Fig. 4.7. Only the indicated dots compose the display; all the lettering and dimensioning does not appear on the CRT and is for explanation only.

The lack of analog equipment in the early stages of the digital simulation testing also necessitated simulating analog inputs in a rather unconventional manner. Core memory registers



Total Normal W.C. Time: 15.905 ms

Fig. 4.6 a) Flow and Worst Case Running Time
Aero Part 1, All-Digital



EF → emergency fuel system on
$\overline{EF}$ → emergency fuel system off

Total Normal W.C. time: 10.445 ms
Total E.F. W.C. time: 10.815 ms

Fig. 4.6 b) Flow and Worst Case Running Time
Engine, All-Digital

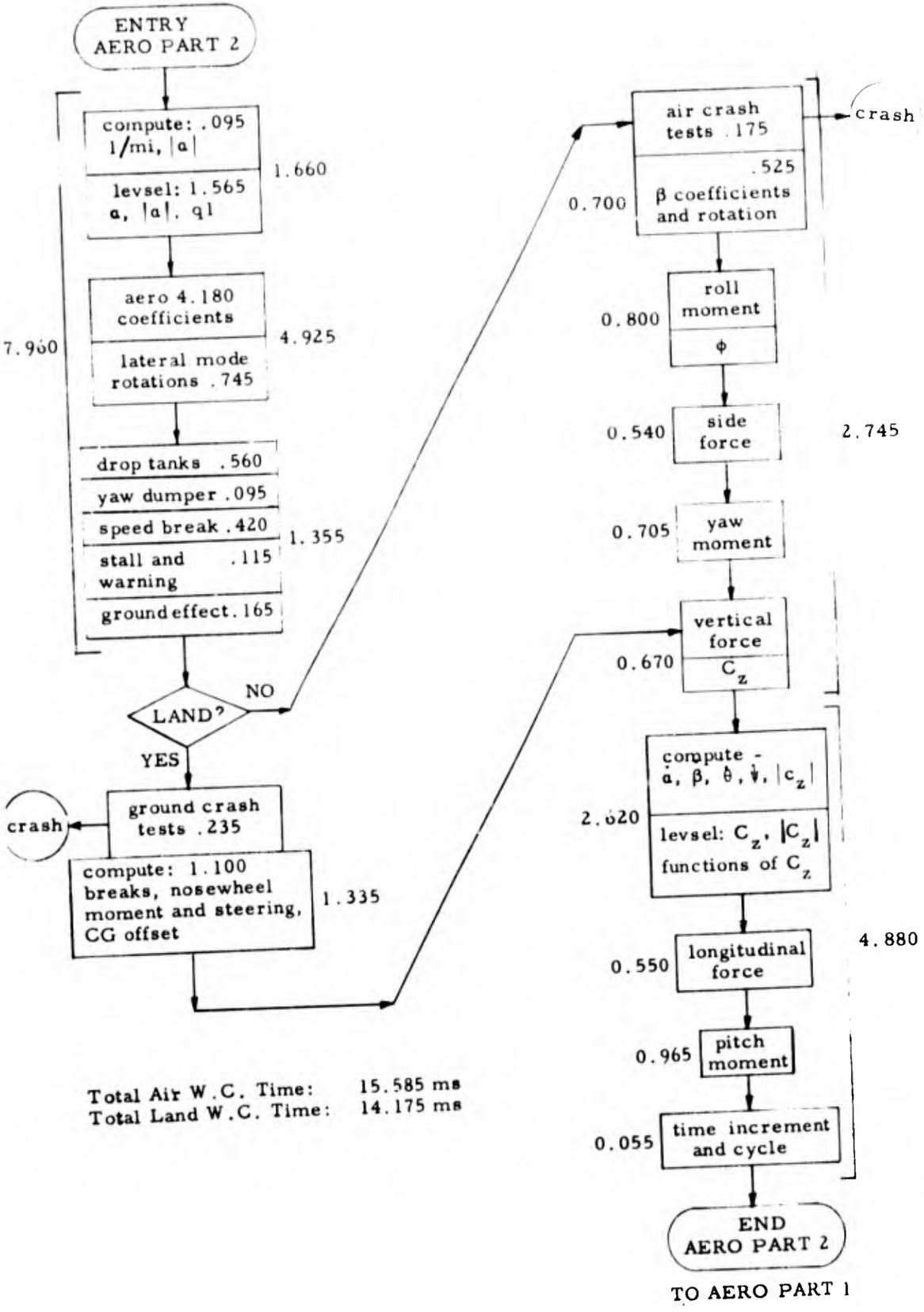


Fig. 4.6 c) Flow and Worst Case Running Time
Aero Part 2, All Digital

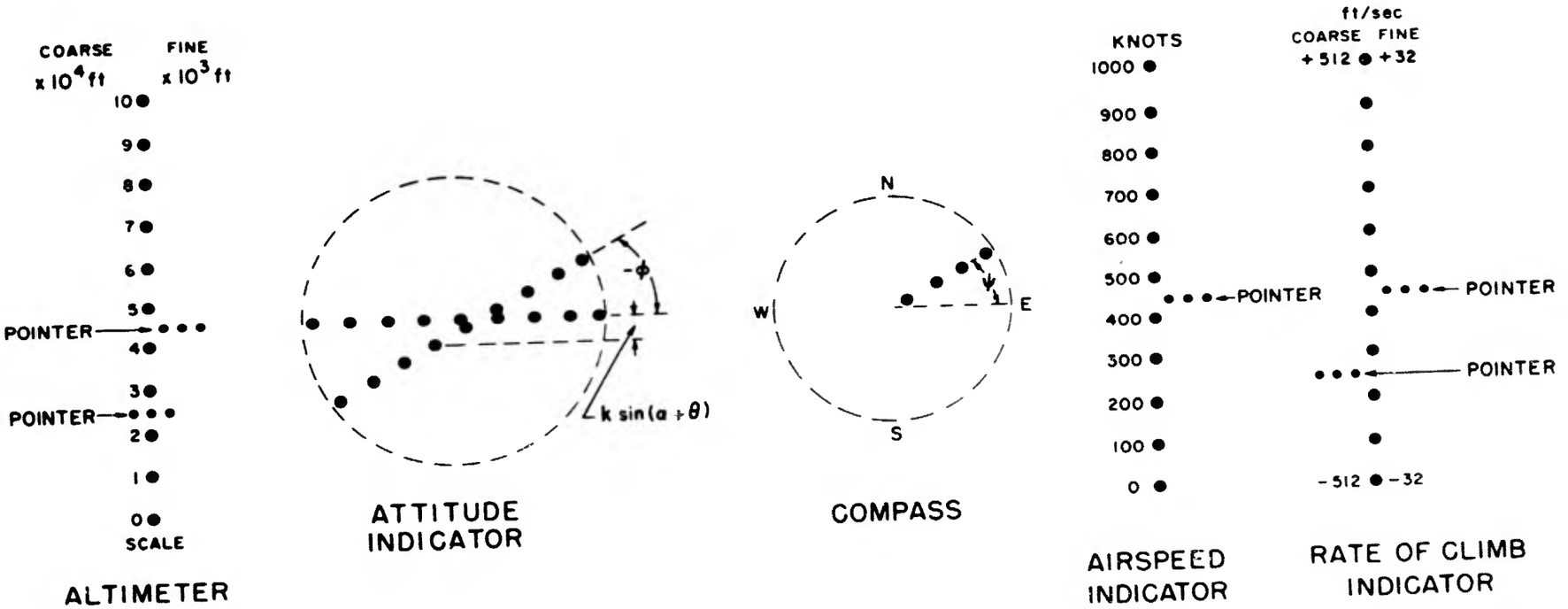


Fig. 4.7 Digitally Simulated Instruments

were reserved to hold quantities representing positions of the four analog controls (horizontal stabilizer, ailerons, rudder, and throttle). Of these, the latter three are handled in the following manner. Three, 3-position switches, whose states are sampled once each computing cycle in a manner entirely parallel to discrete input switches, control programmed incrementation or decrementation of the values stored in each of the three registers. The three possible positions of each switch require only two corresponding IO bits to specify. In the present implementation, one bit on causes incrementation of the control variable, the other bit on causes decrementation of the control variable, and neither bit on causes no change in the corresponding variable. It is impossible to turn both bits on with the given switches.

The horizontal stabilizer requires a somewhat different method as its position and rate of change are rather critical in the longitudinal mode equilibrium. This variable is also stored in a memory register the contents of which are incremented and decremented under pilot control. However, in the case of the horizontal stabilizer, two three-position switches control the changes in the stored quantity, one causing a relatively large rate of change and one causing a small rate of change. The coarse switch allows fast changes in δH and the fine switch allows precise positioning of δH.

Another abnormality of the all-digital simulation is the use of the auxiliary drum memory. The maximum access time for a block of information written on the drum is 33 1/3 milliseconds. The transfer rate, however, is 8.16 μsec/word. Thus, if the position of required information on the drum is optimized, a practical minimum access time of 250 μsec can be achieved. During the running of the simulation, the two major blocks called Engine and Aero Part 2 are read from the drum alternately. In one complete computation cycle, each of these blocks is read once. The two blocks contain a total of 2618 instructions (words) and therefore require slightly in excess of 21.8 msec per cycle for access and reading. With an 8K core memory, this time would be saved since the whole program would fit into core at once.

The timing of the all-digital simulation program is closely related to the drum timing. The computation cycle time goal set by UDOFT and other digital studies was 50 msec/cycle. In the present instance, if the worst-case program running time could be reduced below 27.7 msec/cycle, the total running time of the program would be such that two computation cycles could occur for every three revolutions of the drum (100 msec). However, the actual running time of the F-100A program was approximately 42 msec/cycle (including about 6 msec for instrument display on the CRT). Thus, one cycle of computation occupied two drum revolutions or 66 2/3 msec. Figure 4.8 shows the breakdown of time for one cycle of computation in "normal" worst-case running. The term "normal" worst-case running time implies that the worst-case route is traversed through the simulation program proper and that none of the utility or service programs are entered. This "normal" time, furthermore, is actually the time required to compute one cycle with the service option discrete input switches set so that none of the service routines are entered.

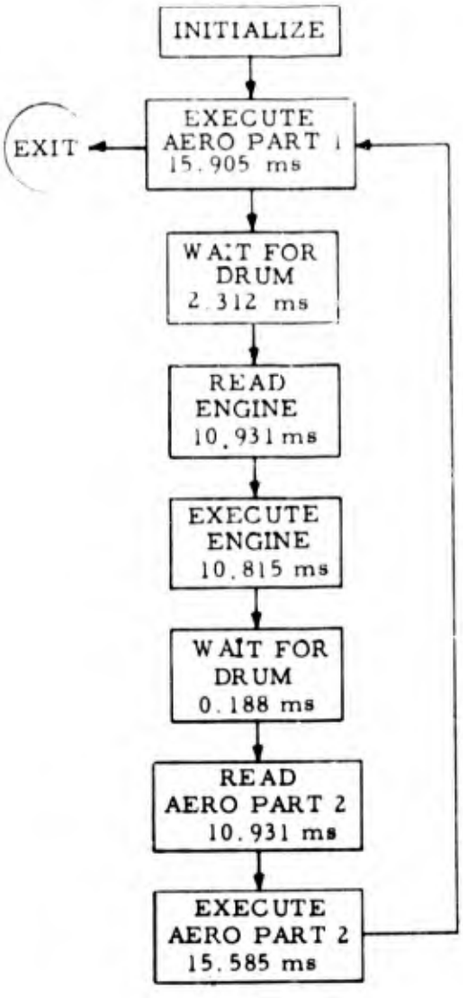
Three areas where running time can be considerably shortened by relieving purely formal restrictions on the program are these: (1) magnetic drum storage usage, (2) analog inputs, and (3) analog outputs. From the foregoing discussion, doubling core memory size to avoid using the slow auxiliary storage saves 24.3 msec in running time. Utilizing analog equipment described elsewhere in this report for analog inputs and outputs, saves roughly 7 msec. Thus, with these three spurious restrictions removed, the intrinsic worst-case running time of the F-100A problem on the PDP-1 would be roughly 35 msec's per solution cycle.

Memory Space: The PDP-1 4K memory is divided into three major areas in the all-digital F-100A simulation program. These areas are (a) data, (b) resident program, and (c) swapped program.

The data area contains a block of registers holding variables, breakpoint tables for level selected variables, function tables for non-linear functions, and the integration subroutine calls. The fourth item is included in the data area because the calling sequence for the integration subroutine contains registers in which variable quantities are held. The resident program area contains subroutines and Aero Part 1.

The swapped area is the one which holds, when necessary, each of the program blocks contained in the auxiliary drum memory. When the simulator is running, this area is alternately occupied by Aero Part 2 and the Engine. At times other than "normal" running, the output program, the initialization program, or the X-Y plotter program may occupy this area. The symbolic debugging program, DDDT (or D^3T), when operating, occupies essentially all of core and utilizes an unused field of the drum to store the simulation program.

Figure 4.9 shows the allocation of core memory to each of the program blocks. Figure 4.10 shows the placement of various routines on the first six fields of the drum memory.



Program W.C. Time:	42.305	ms
Drum Transfer Time:	21.862	ms
"Waste" Time:	2.500	ms
Total Cycle Time:	66.667	ms

Fig. 4.8 Worst Case Running Time Distribution
All-Digital F100 Simulation Program

(G) Special Routines

Initialization Routine: The all-digital F-100A initialization routine performs the function of setting initial conditions for a "flight" of the aircraft model. There are two entry points to the initialization routine. One entry point causes the model to be stopped on the ground with the engine shut down. The second entry point puts the model in straight and level flight at 20,000 feet at Mach 0.7. An entry point is selected by one of two switches on the discrete input console. The initialization routine executes its task repetitively until both of the discrete input switches controlling entry to the routine are turned off. The state of the model is determined by the last control switch to be turned off.

It was much easier to initialized the all-digital model with switch-activated control surface rates than it was to initialize the all-digital version with analog stick inputs. In the latter configuration, the operator had to set the correct control surface positions manually before straight and level flight could be maintained. In the case of the switch-activated control inputs, the initialization program set the surfaces to the correct positions. Figure 4.11 shows a flow diagram of the initialization routine.

Output Routine: The flexible output routine written for the F-100A simulation programs was not intended as an output medium for real-time simulation due to the intrinsic speed limitations of input/output devices available with the PDP-1. Conceivably, with a high-speed channel option and a 1,000 lpm line printer, results from every other computation cycle could be recorded and real-time simulation maintained. The existing routine, however, has options to utilize the typewriter, the paper tape punch and the cathode ray oscilloscope (CRO) display unit, all of which are on-line units.

The output routine is capable of transmitting up to 54 computed variables in 16 different combinations to the selected output device. Which of the possible 16 combinations is employed is determined by the setting of four switches on the discrete input panel. The output medium (typewriter, tape, or CRO) is also selected by switch settings on the discrete input panel. If the

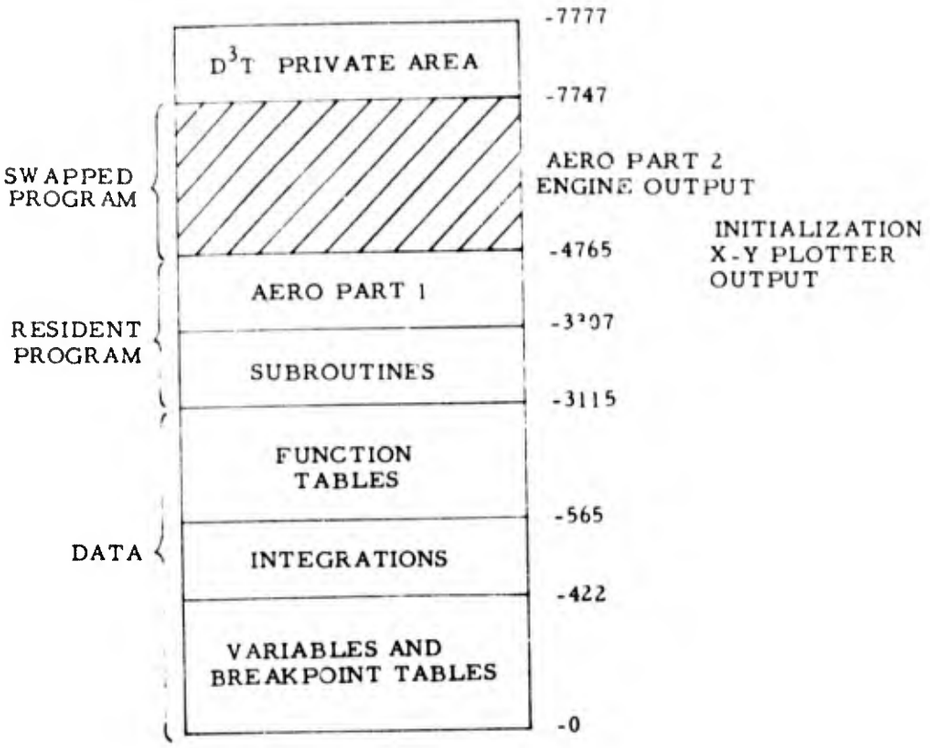


Fig. 4.9 Core Memory Allocation, All-Digital Program

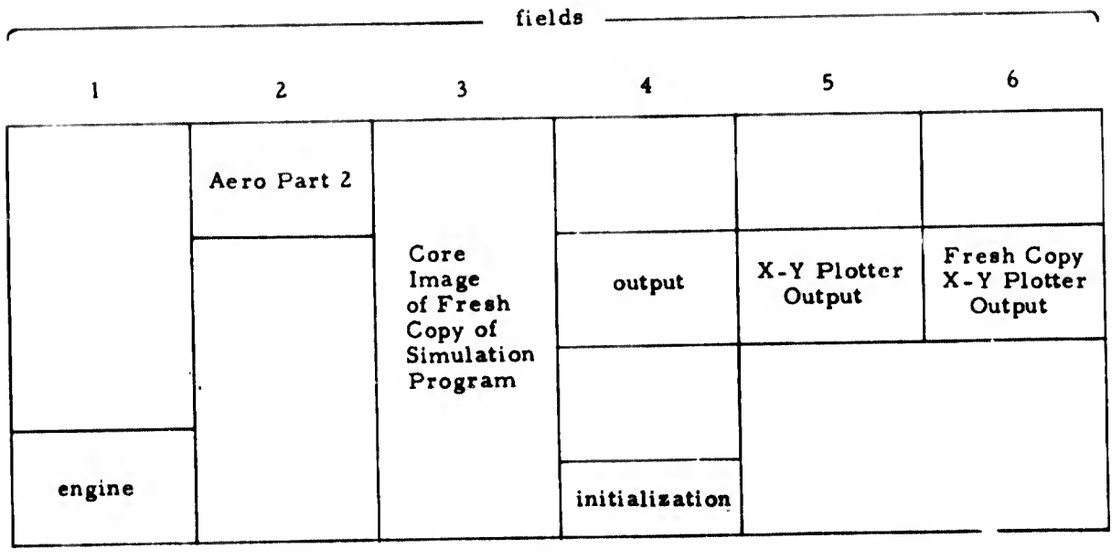


Fig. 4.10 Drum Memory Allocation

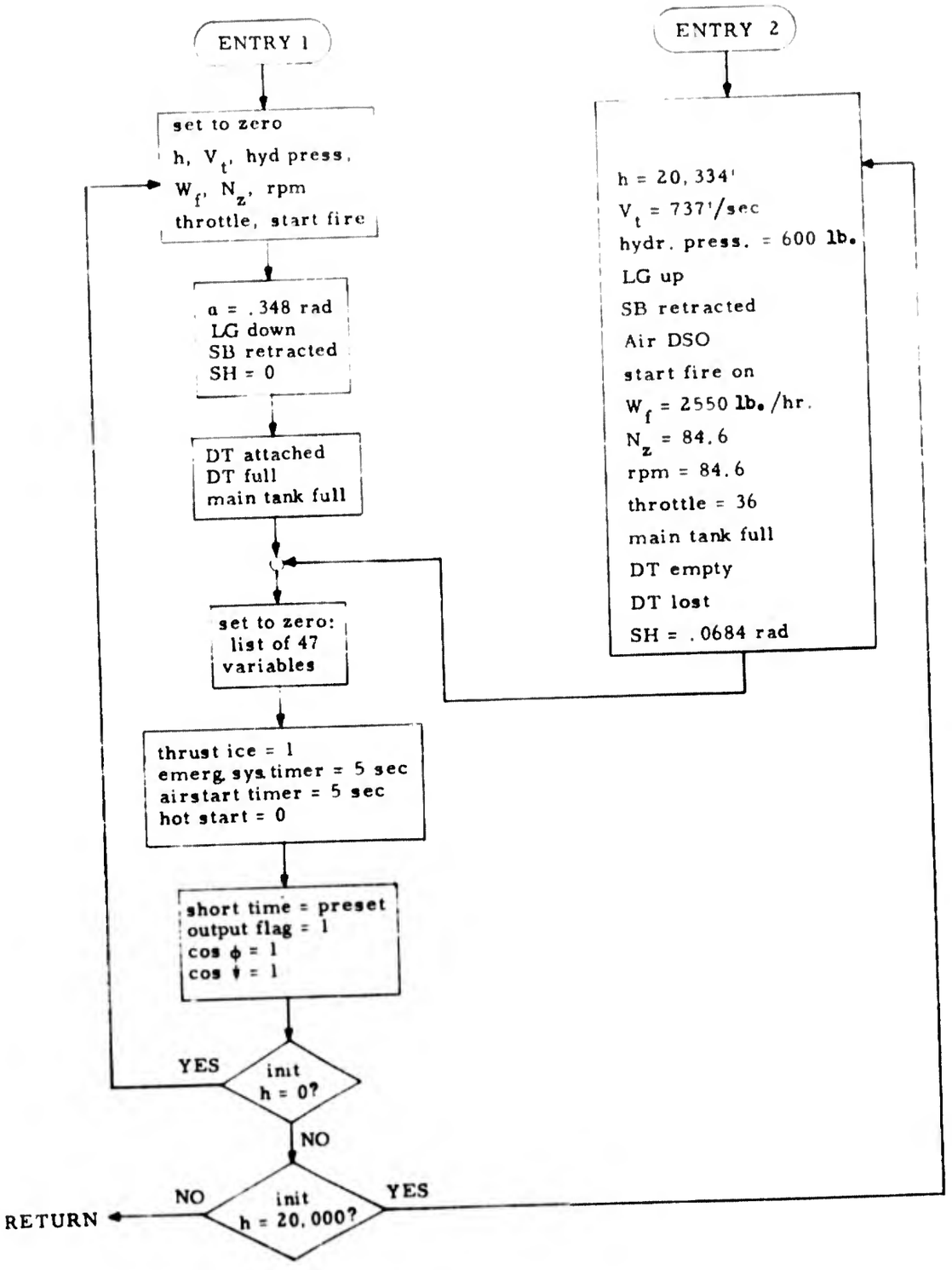


Fig. 4.11 Flow Diagram of Initialization Routine

output medium is either tape or typewriter, normally only variable values will be transmitted. However, a heading to identify variables in the final typed copy can be obtained with a discrete input switch option. The CRO output option displays alphanumeric characters on the CRO screen so that the values of the selected variables and their names are readable directly. The CRO output consumes less than 2 percent of the time required for a comparable typed output. A repetitive mode of operation (FREEZE) is available as a discrete input switch option for the CRO output. Another optional repetitive mode of operation found to be of value is called STEP FREEZE, which operates in conjunction with the FREEZE option. The FREEZE mode causes the output routine to repetitively display the names and values of the selected set of variables until the STEP FREEZE discrete input switch is turned on. At switch-on time the computer enters a dynamic halt condition until the STEP FREEZE switch is turned off. At switch-off time the computer executes one cycle of computation and re-enters the repetitive display (FREEZE) loop. Photographs of typical CRO outputs are shown in Fig. 4.12. A detailed flow diagram of the output routine appears in Figs. 4.13a through 4.13e.

Plotter Output Routine: The plotter output routine (PLOTEX) performs the following functions: (a) auxiliary computation for plotting variables, (b) plotting axes, (c) decreasing the speed of the simulation to allow the plotter pen to accurately reproduce the desired output curve, (d) decreasing the independent variable step width and interpolating the dependent variable for plotting accuracy, and (e) computing a stimulus for the airframe model. The operation of PLOTEX is blocked out in Fig. 4.14.

PLOTEX was used mostly for plotting tansient responses of the airframe to standard control surface input pulses as described in Chapter VI. These stimuli are functions of time and must be synchronized with the output plotting. Thus, the stimuli are computed by PLOTEX. The control surface to be pulsed, the pulse amplitude, the shape and zero-offset of the stimulus, the variable to be plotted, the scaling and offset of the plotted variable, and the time (or other independent variable) and its scaling and offset are parameters that must be inserted into PLOTEX using D^3T before the plot can be run.

4.2 HYBRID PROGRAM

(A) General

The hybrid F-100A simulation program is patterned strongly after the all-digital F-100A simulation program. In addition to the transfer of a large part of the aerodynamic computation to the analog domain, several other organizational changes have been incorporated in the digital computer part of the hybrid program. The analog portion of the hybrid simulation is described in Chapter III. The present chapter describes the organization, timing, and memory requirements of the hybrid program, and also compares the time and memory requirements of the hybrid and all-digital versions of the F-100A simulation program.

(B) Hybrid Programming Techniques

The programming procedures employed to transmit information between the analog and digital domains and to carry out arithmetic sequences with the pulsed analog equipment have been described in Chapter III. These functions and the outright transfer of a portion of the digital calculations to the analog subsystem constitute the principal innovations in the hybrid program. To complete the picture, however, the method of coping with the ground-air transition must be discussed.

A technique was needed for forcing certain variables in the aerodynamic computation to prespecified values when the aircraft is on the ground. Furthermore, the hybrid program had to include correct handling of the land-air and air-land transitions. The variables which needed special attention were altitude (h), pitch angle (θ), roll angle (ϕ), roll rate (p), and yaw rate (r). All of these must be clamped to zero when the aircraft is on the ground and when the aircraft just touches the ground on landing. However, on take-off, θ must be free to increase to allow a rate of climb ($r/c = v_t \sin\theta$) to develop if altitude is to increase. Furthermore, θ must not be allowed to become negative when h = 0 since this implies that the aircraft is somehow burrowing into the ground.

The method chosen for handling these situations monitors the pitch angle rate $\dot{\theta}$ ($\dot{\theta} = a_z/V_t$ when $\phi = 0$) by looking at a_z and deciding digitally whether the affected analog integrators should be in a reset state or in an operate state. In the reset integrator state, all the above-mentioned variables are forced to zero and in the operate integrator state, the variables are free to follow their respective rates. Qualitatively, if $a_z < 0$, then the tendency of the aircraft is to nose up or take-off dictates switching to the operate integrator state. If $a_z > 0$, then the tendency of aircraft is to nose down, which dictates switching to the reset state. In actuality, the switching tests were set up with a small dead zone at $a_z = 0$ to avoid relay chatter in the integrators and to avoid the attendant transients in the analog circuits. A graphic representation of the switching tests programmed for the hybrid simulation appears in Fig. 4.15.

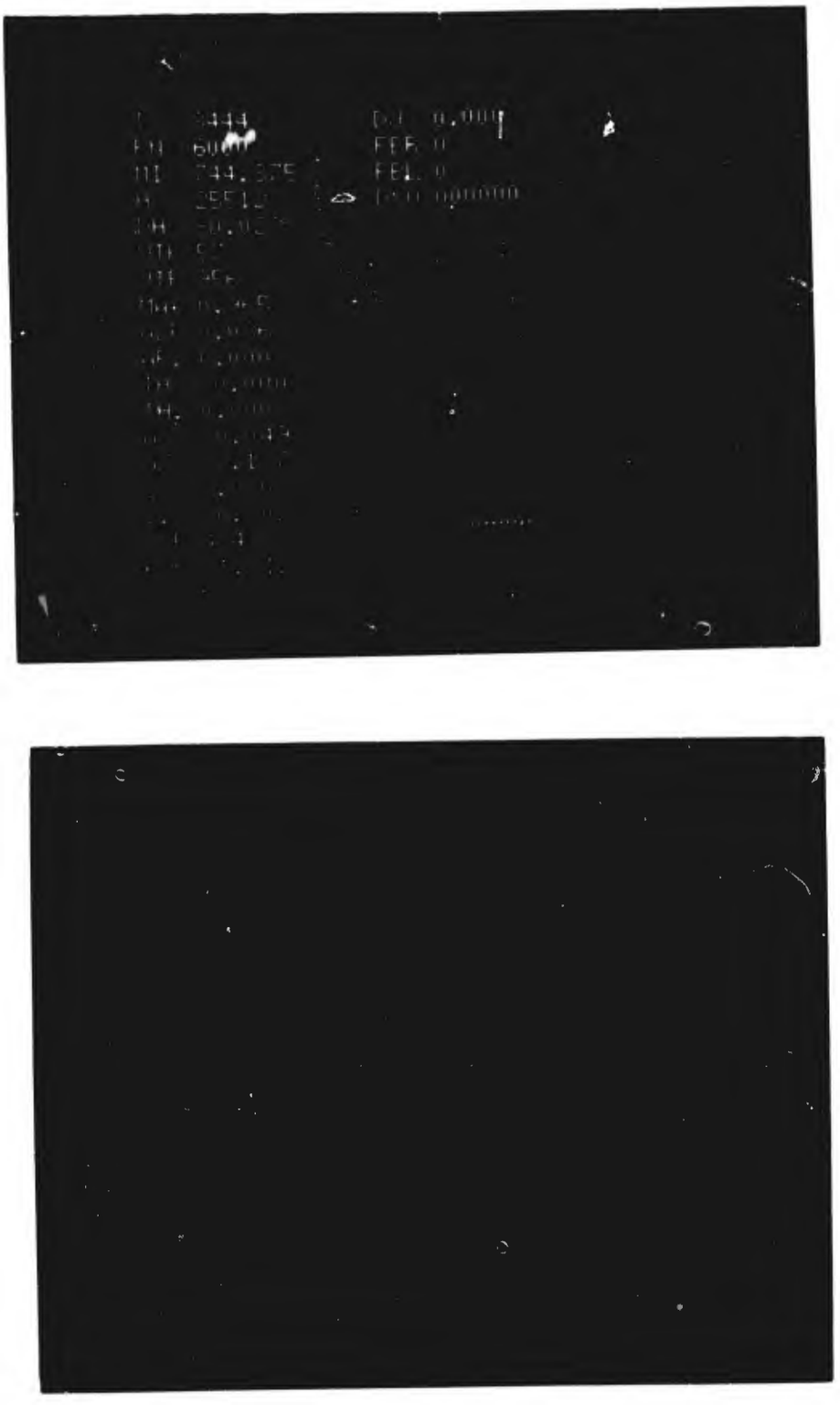


Fig. 4.12 CRO Output Routine Results

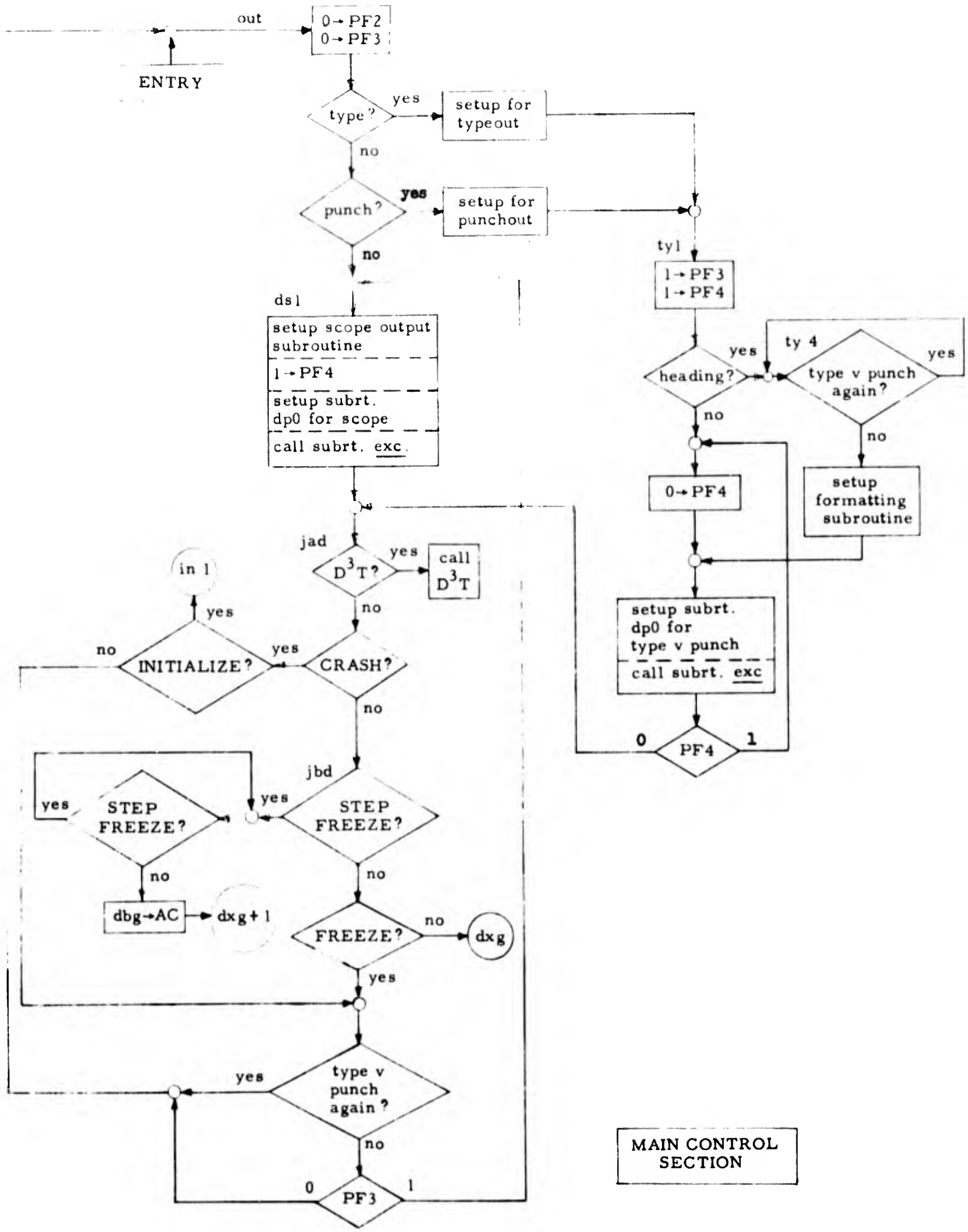
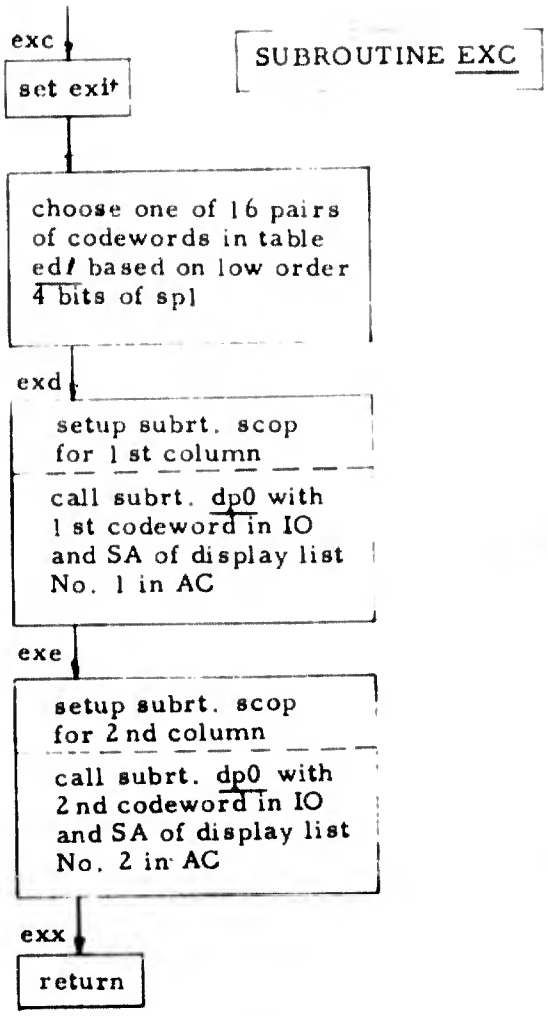


Fig. 4.13 a) Output Routine Flow Diagram, Part 1



note:

a 1 in a codeword bit position selects the corresponding display list entry for output

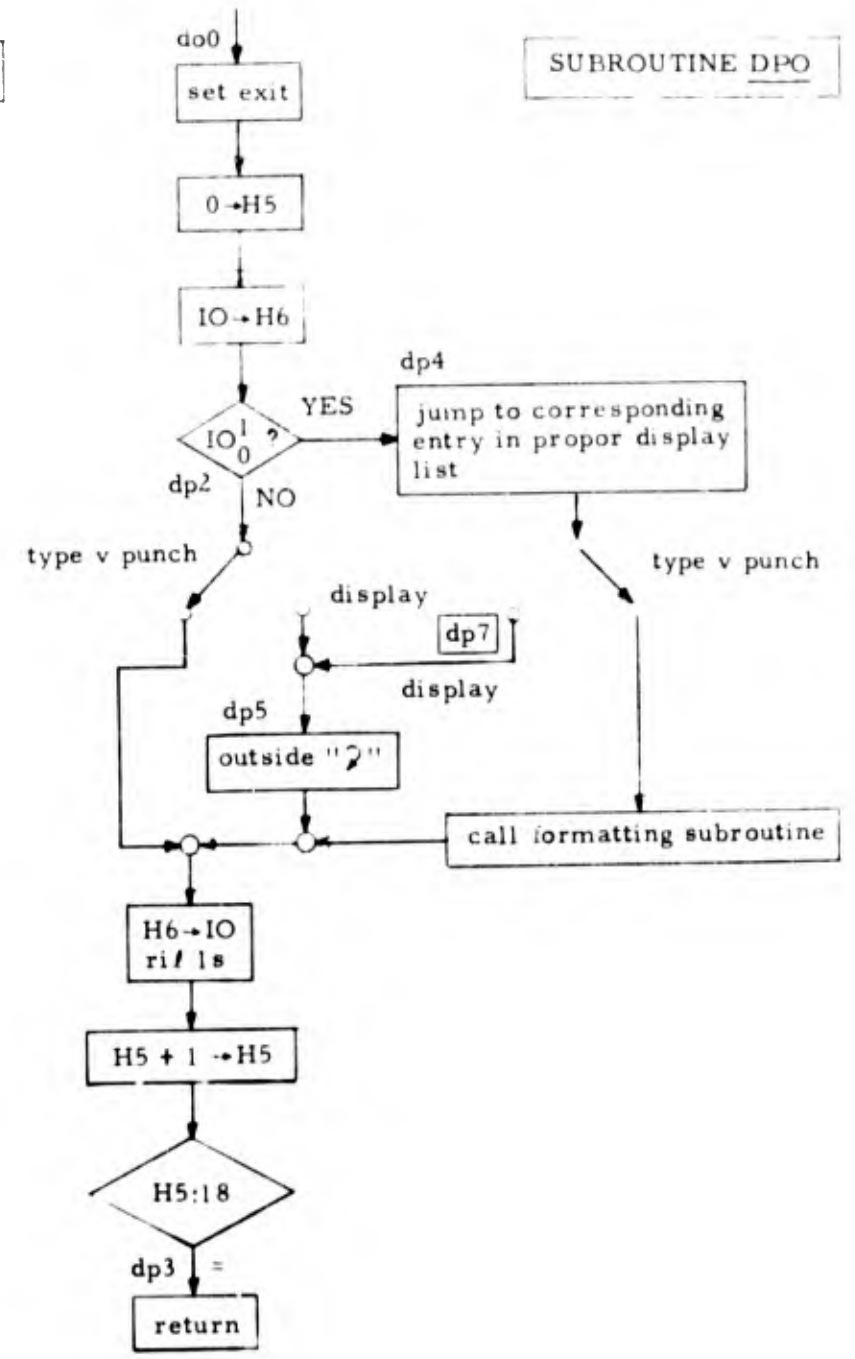


Fig. 4.13 b) Output Routine Flow Diagram, Part 2

NOTES:

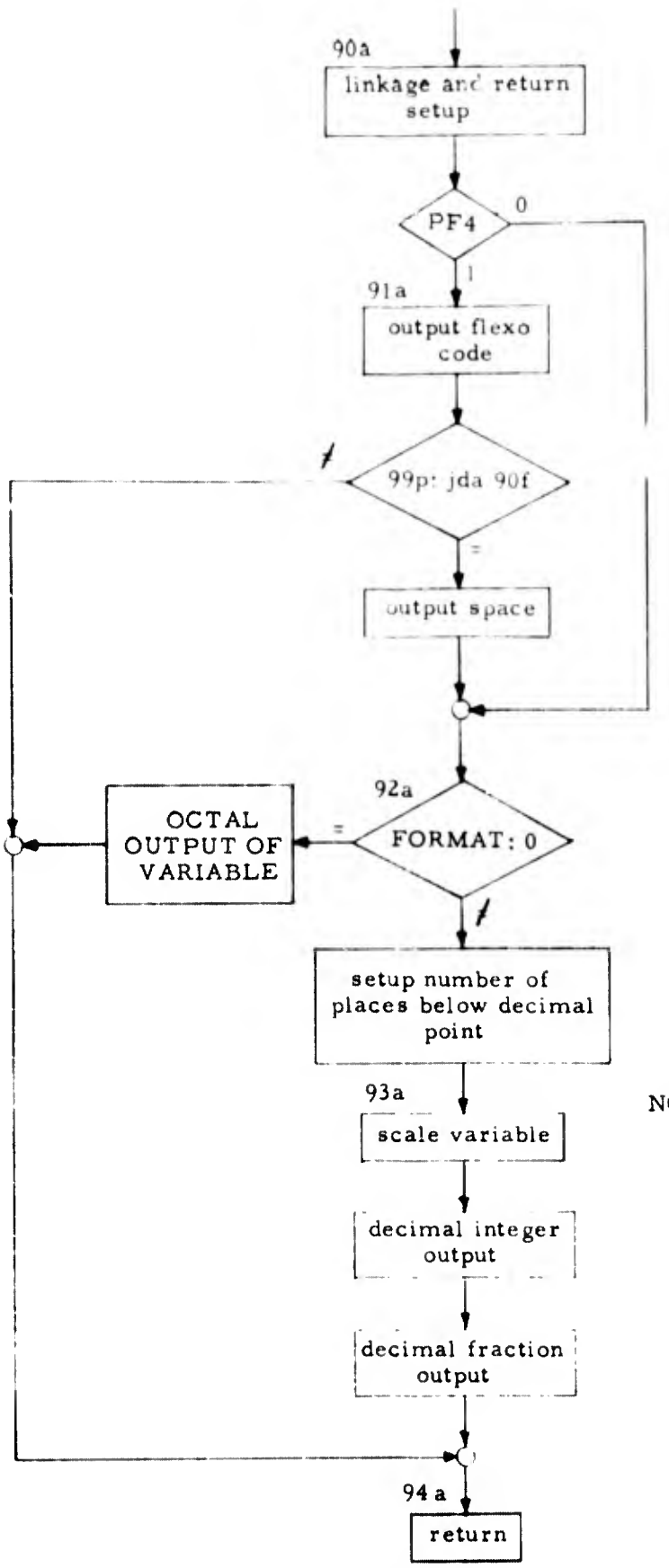
1. Formatting subroutine allows up to and including 7 columns to be typed or punched for a pagewidth Every 7th call to the subroutine produces a carriage return All other calls produce tabs

2. Each display list entry contains the following information:

   a) variable
   b) scaling
   c) number of places after decimal point
   d) flexo code for variable name

   A jump to a display list entry causes the variable to be loaded into the AC and H main output subrt. (90a) to be called using the cal instruction.

Fig. 4.13 c) Output Routine Flow Diagram, Part 3



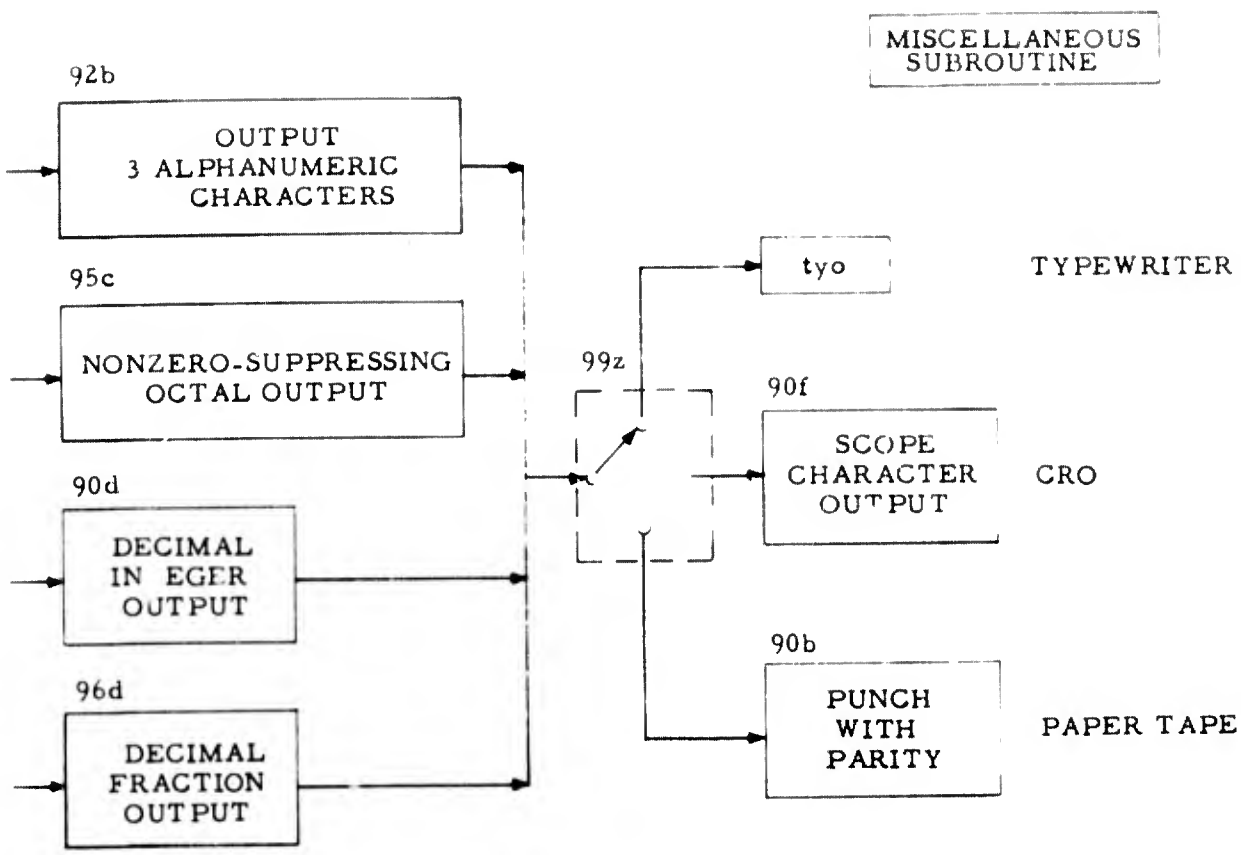
SUBROUTINE 90a

NOTES:

1 Bits 9, 10 and 11 of the cal instruction contain the format code "n"
- (n = 0) → non zero sup octal output
- (n ≠ 0) → decimal output with n-1 places below decimal point

2 Bits 12-17 of the cal instruction indicate scaling from b0 (pure integer) to b17 (pure fraction) or further. No negative b-numbers are allowed.

Fig. 4.13 d) Output Routine Flow Diagram, Part 4



note: maximum subroutine depth attained is six.

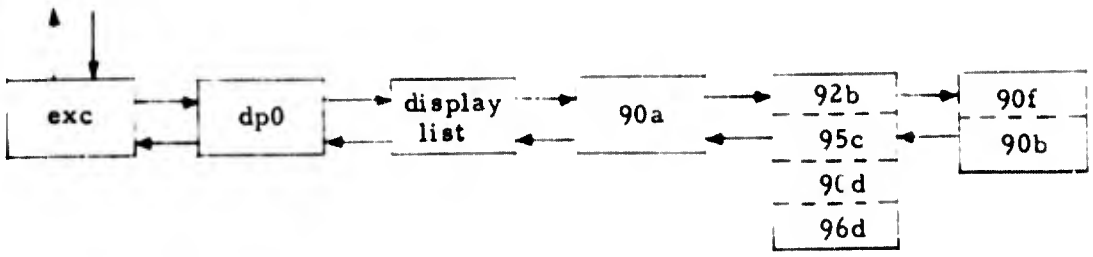


Fig. 4.13 e) Output Routine Flow Diagram, Part 5

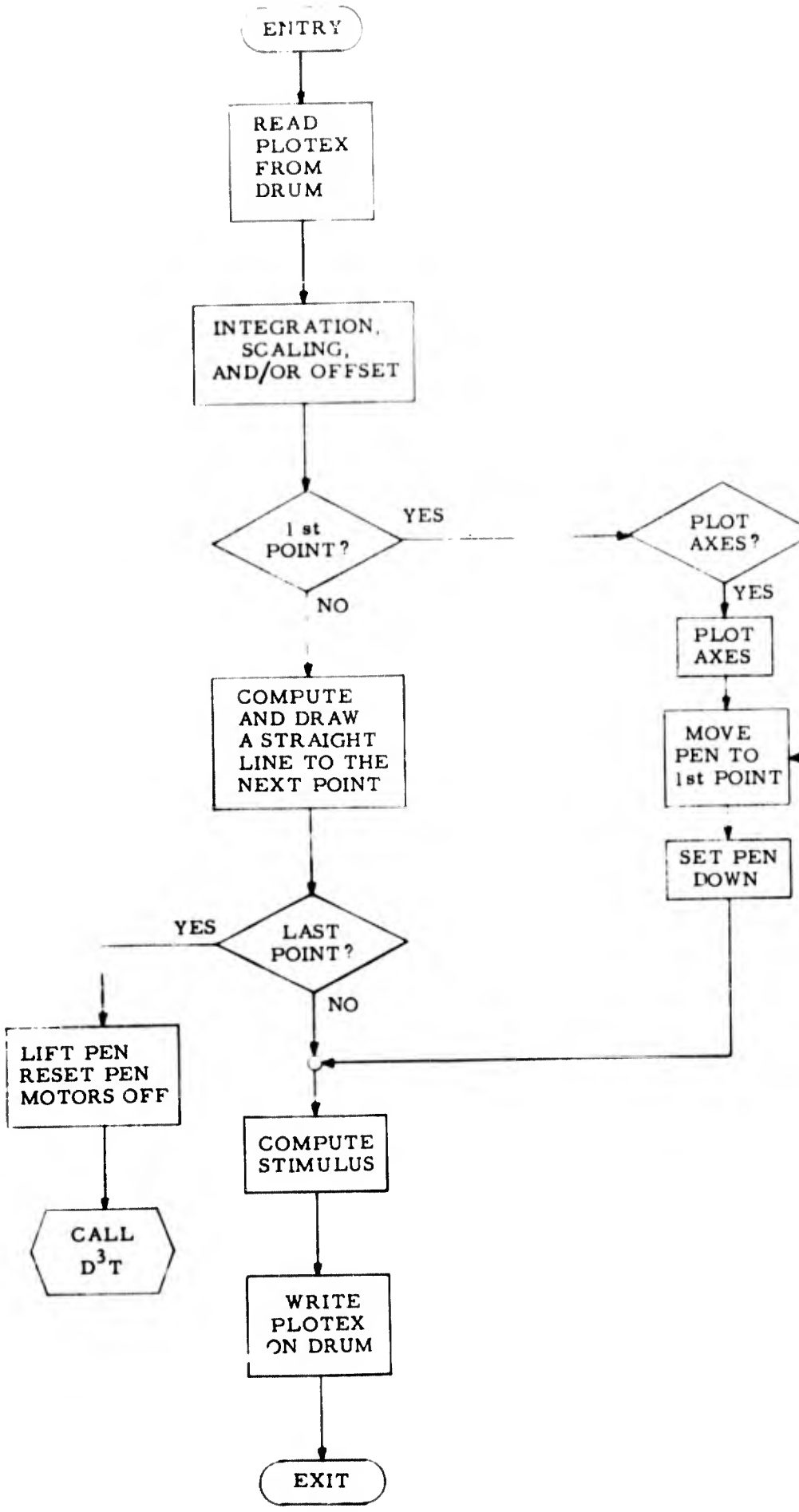


Fig. 4.14 Flow Diagram of PLOTEX

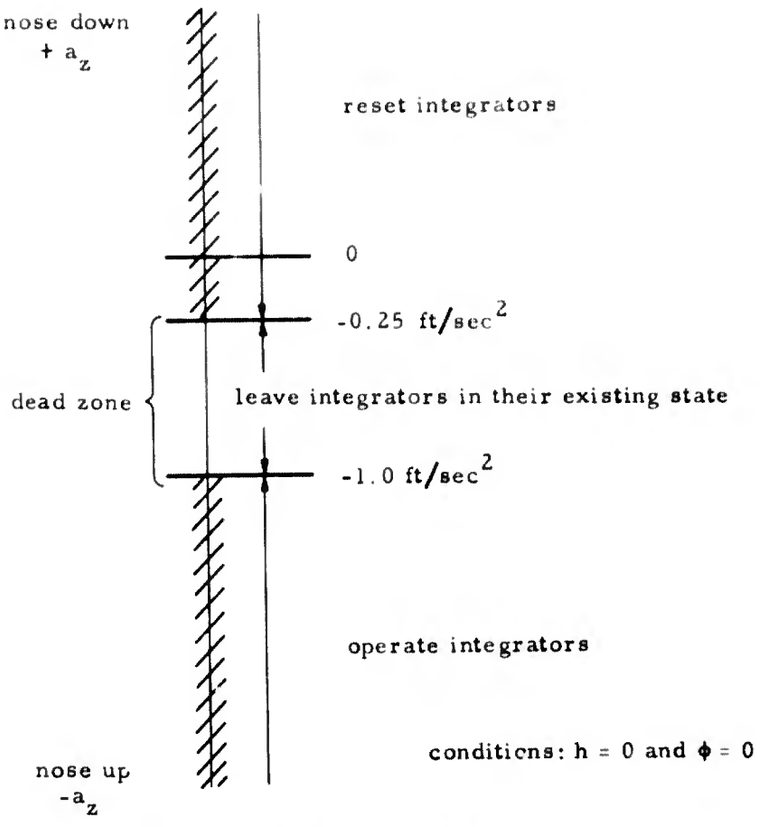


Fig. 4.15 Land-Air Transition Integrator Switching

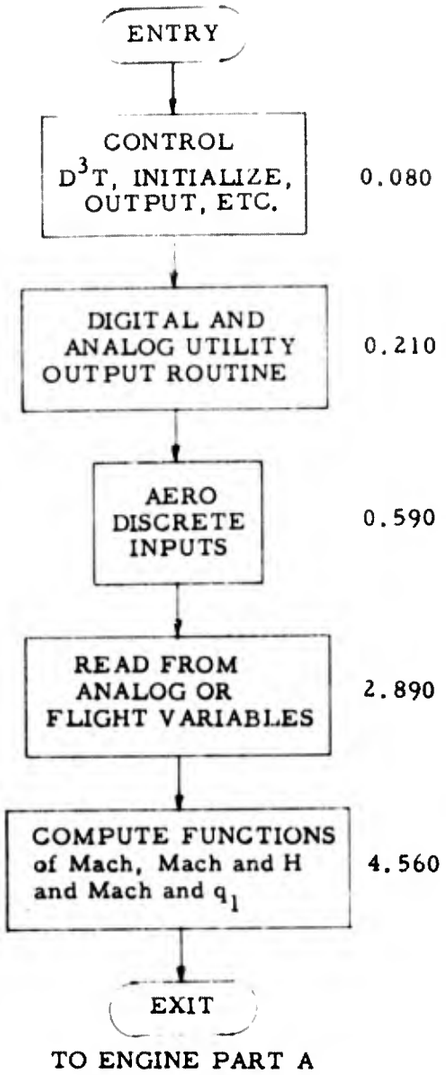
(C) Hybrid Program Organization

The main change in program organization which occurred as the hybrid F-100A program evolved from the all-digital F-100A program, entailed grouping digital computations involving only slowly-varying variables into one block and computations involving both fast-changing and slow-changing variables into another block. Based on the customary guide that solution rates must be ten or twenty times greater than the natural frequencies, the routines servicing slow variables should then need no more than five executions a second and those servicing fast variables (up to 1 or 2 cps) should need no more than twenty executions a second.

The actual organization of the hybrid program is blocked out in Figs. 4.16 through 4.19. Note that now Aero Part 1 contains all of the "slow" computations extracted from Aero Part 2 in addition to control functions and a short special output function routine. Furthermore, the original hybrid organization of the engine program required three segments of approximately equal length. The first engine segment, engine part a, was to remain in core memory constantly, while the second and third segments (parts b and c) occupied the swapped area of core alternately. A reorganization of the original hybrid program required the three segments of the engine program to occupy the swapped area in triangular alternation. In any case, the drum transfer times required are separated from running times in the timing analysis. The sequence of execution of the sub-blocks in the hybrid program is shown in Fig. 4.19. It is clear from this figure that the "slow" routines are executed at half the frequency of the "fast" routines. Throughout the testing of the hybrid program, this frequency ratio remained constant, though the base frequency (the frequency of the "fast" routines) was varied.

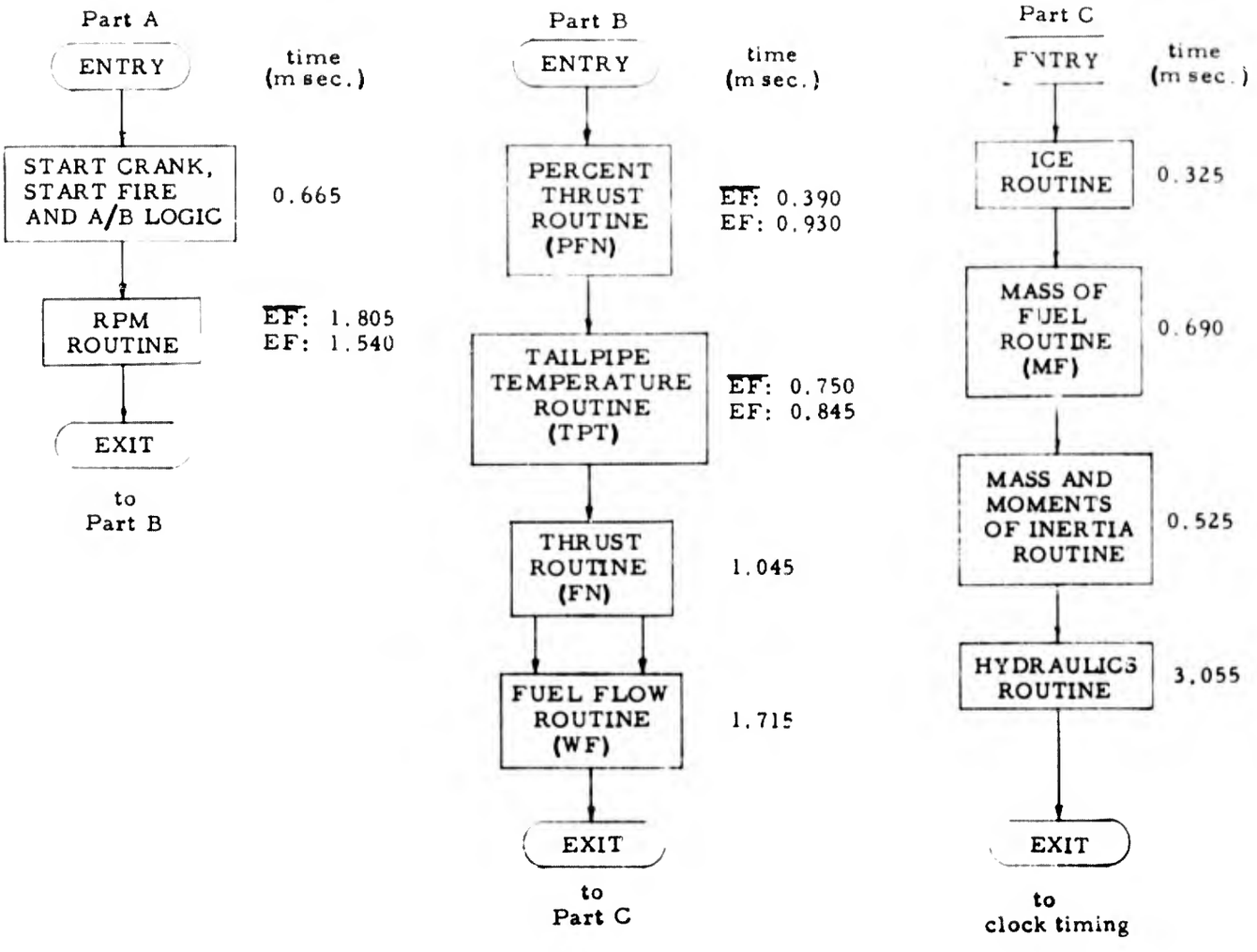
(D) Hybrid Program Timing and Memory Space

Figure 4.19 gives the execution times for the hybrid program. The table of running times is broken down into "long cycles" (those including execution of the "slow" routines) and "short cycles". Averages are included to facilitate direct comparison with the all-digital program. Unfortunately, the all-digital program was never split into "fast" and "slow" segments so that a comparison between the average hybrid time and the all-digital time would be unfair in that the all-digital program could also be split in a manner analogous to the hybrid program with an attendant time saving. However, a meaningful time comparison can be made directly between the "long" cycle time of the hybrid program (27 ms.) and the intrinsic all-digital time (35 ms.). Thus, the immediately obvious effect of transferring part of the digital computation load to the analog domain is to reduce the running time by 8 ms. The more significant effect on solution rate requirements will be discussed in Chapter VI.



Total Normal Worst Case time = 8.330 msec.

Fig. 4.16 Hybrid Aero Part 1 Routine

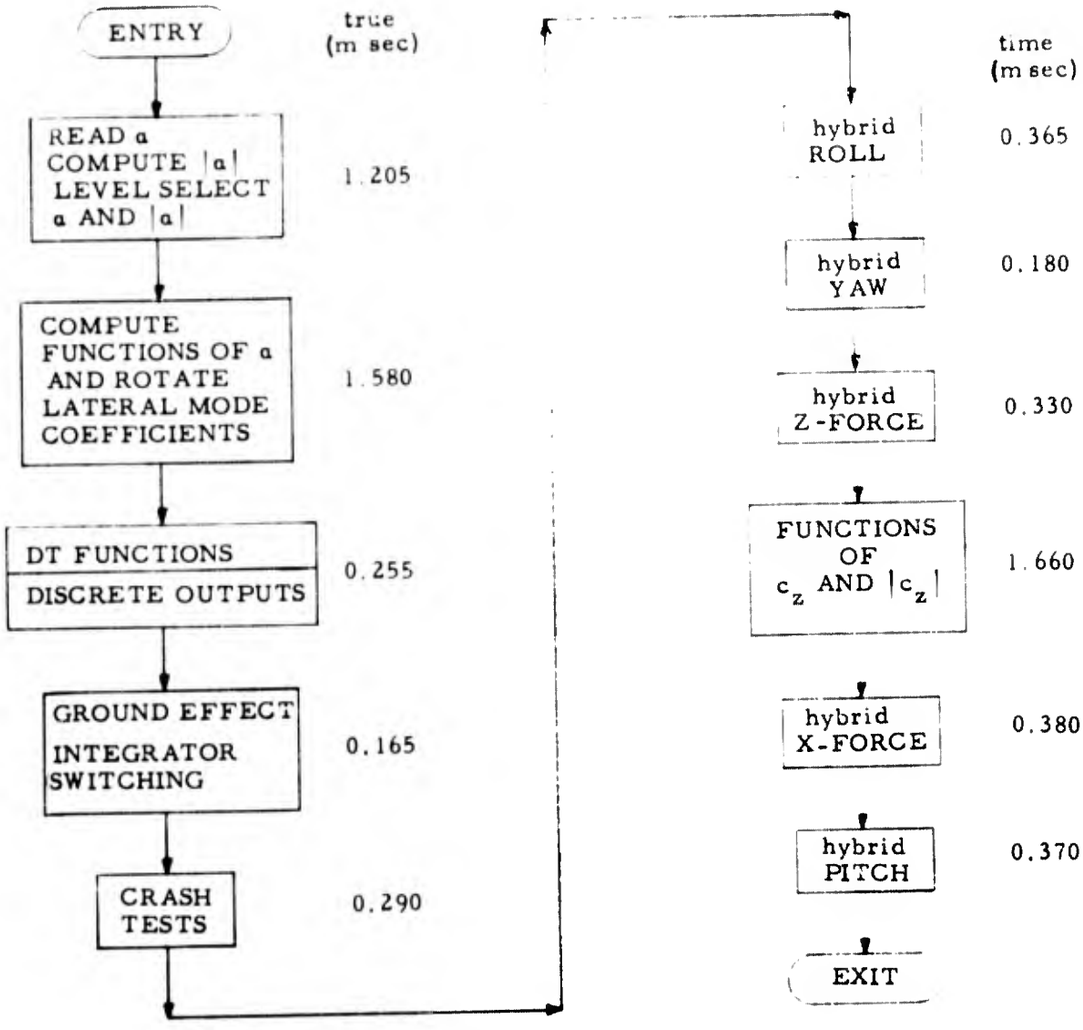


Times are normal worst case running times.

Total $\overline{EF}$ = 10.965 ms
EF = 11.335 ms

EF → emergency fuel system on
$\overline{EF}$ → emergency fuel system off

Fig. 4.17 Hybrid Engine Routines



Total Normal Worst Case Running Time = 6.780 msec.

Fig. 4.18 Hybrid Aero Part 2 Routine

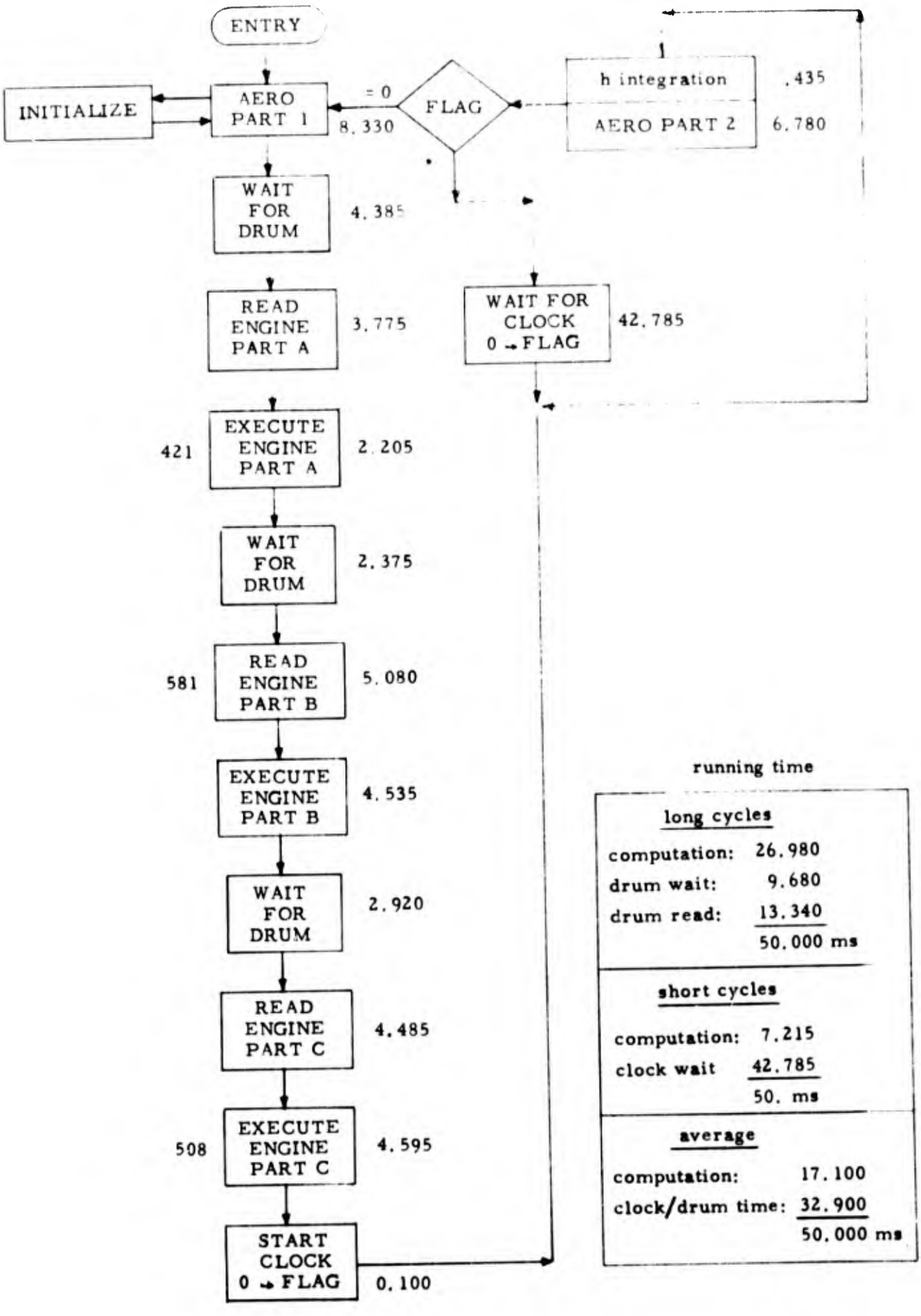


Fig. 4.19 Gross Flow and Worst Case Running Time Breakdown; Hybrid Program

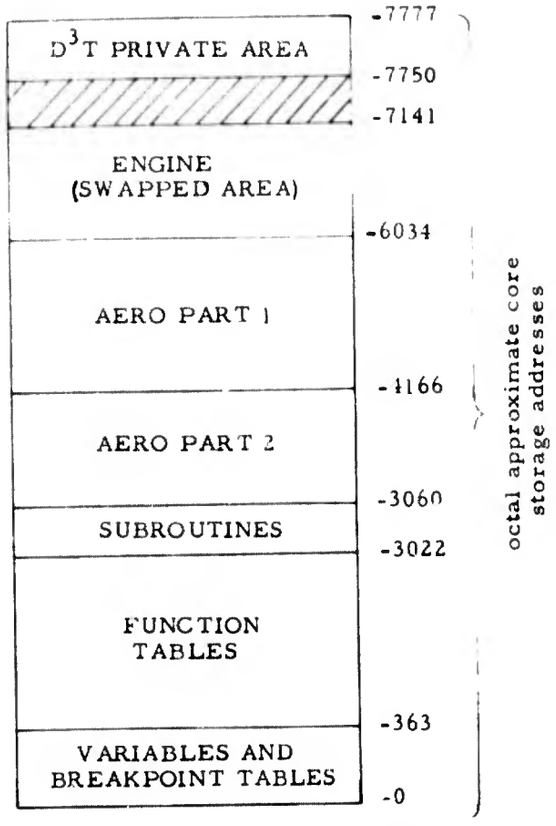


Fig. 4.20 Core Memory Allocation, Hybrid Program

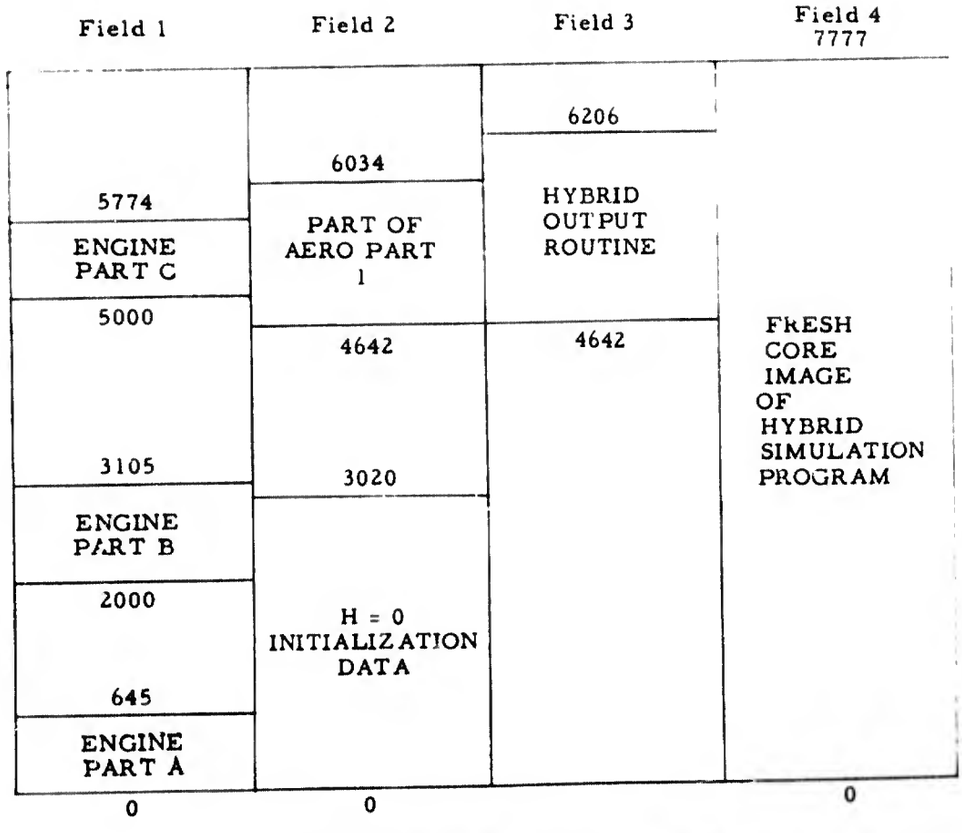


Fig. 4.21 Drum Memory Allocation, Hybrid Program

Figure 4.20 shows the layout of the hybrid program in core storage. The shaded area occupying some 600_8 registers (384_{10}) near the top of storage was reserved in a later revision for possible program modifications to avoid time-consuming reassembly of the whole program. Figure 4.21 describes the utilization of the lower four drum storage fields by the hybrid program. Other drum fields may be used in running this program for, say, initialization at present conditions other than ($H = 0$, $V_t = 0$) and for utility programs such as D^3T.

Figure 4.22 gives the storage and time comparisons between the all-digital and hybrid simulation programs. This comparison table is based on a hybrid "fast" routine iteration rate of 20 solutions/sec, no drum transfer times or waits, and no "short" cycles where the execution of the "slow" routines is omitted.

Memory Requirements	All-Digital	Hybrid	Hybrid Savings
Variables and Breakpoint Tables	274.	241.	- 33.
Function Tables	1355.	1311.	- 44.
Subroutines	121.	30.	- 91.
Aero Part 1	839.	934.	+ 95.
Aero Part 2	1309.	582.	-727.
Engine	1469.	1510.	+ 47.
TOTALS	5367.	4608.	-759.
Running Times (m.s.)			
Aero Part 1	15.905	8.430	- 7.475
Aero Part 2	15.585	7.215	- 8.370
Engine	10.815	11.335	+ .520
TOTAL Computer Time	42.305	26.980	- 15.325

Fig. 4.22 Comparison of Time and Memory Requirements Between the All-Digital and Hybrid F-100A Programs

BLANK PAGE

CHAPTER V

COMPONENTS

5.1 INTRODUCTION

One of the major objectives of the M. I. T. effort has been to reflect the present state of the art in hybrid computing elements with consideration given to both the technical and economic aspects. Since the study and construction phases of the project coincided with the period in which acceptable solid-state analog and pulsed analog components became a practical reality, this objective has necessitated quick adaption to the rapidly changing technical situation. To illustrate this, the Electronic Systems Laboratory had available at the start of the study phase a family of pulsed-analog units developed under previous contracts.[7] For the most part, these circuits employed vacuum tube and solid-state combinations, since earlier efforts to develop completely solid-state units had not been successful. For example, four graduate theses on transistor operational amplifiers had been underwritten without a suitable pulsed-analog design being produced. At the time, only second order improvements in existing tube circuits seemed feasible. On this premise, the development of a Nuvistor operational amplifier[19] and a tube sample-and-hold gate[20] was initiated at the start of the program. In a similar vein, when the requirement for a standard analog subsystem arose, the project planned to procure one of the many commercial tube units then available. However, midway in the design phase, it became apparent that improvements in the performance characteristics of transistors and lower transistor prices made it reasonable to think in terms of a completely solid-state system, both with respect to the analog and pulsed-analog equipment. The advantages of such a system in size, reliability, and power consumption could not be ignored even though the switch to solid-state circuitry meant scrapping previous tube designs.

On several other occasions, new technical developments forced the project to revise prior decisions on implementation. The success of the photoresistive multiplier, for example, dictated the replacement of the quarter-square multiplier and servo multipliers originally specified by the more economical photoresistive devices. The appearance of the Adcom sample-and-hold gate having a permanent holding capability clearly represented an important breakthrough that made existing temporary-hold gates obsolescent. The installation at M. I. T. of a PDP-1 computer, a commercial outgrowth of the TX-0 with superior speed and programming features, caused us to abandon the one-of-a-kind TX-0 in spite of a substantial prior investment in programs and experiments.[4, 5, 6, 11]

Ultimately, of course, the desire to incorporate the very latest equipment in a system and the timetable for completing the system come into conflict. As a consequence, the ideal of creating a true state-of-the-art system can never be realized in practice. However, in a development program of this sort, it is critically important that a degree of flexibility be permitted in utilizing new technical advances as they become available, especially when the state-of-the-art is changing rapidly. A case in point is the UDOFT development over the period from 1950 to 1960 where the implementation was rigidly and artificially restricted to the tube technology in existence at the time of the computer's inception.

The term "state-of-the-art" implies not only that a certain level of performance is achievable, but also that this performance is achievable at a reasonable cost and with reasonable consistency. Attaining the ultimate in performance by applying the philosophy that money is no object and the best is none too good, leads not to distinction, but to the poorhouse. In truth, the research and development sport loses much of its flavor and challenge unless some economic constraints are applied. The M. I. T. effort hopefully demonstrates that an effective hybrid computation facility can be assembled at a reasonable cost and that the poor man's approach of making a system only as good as it has to be (with adequate margins of safety) has much to recommend it.

Numerous examples from the present hybrid configuration illustrate these quasi-economic aspects of systems design. At the time a choice was made, several 10 bit, packaged analog-to-digital converters were available with encoding times ranging from 2 microseconds to 108 microseconds. The unit finally selected (Beckman 4040C) had an encode time of 30 microseconds, hence it did not represent the state-of-the-art as far as conversion rates are concerned. However, it was less expensive than all faster converters, it offered a margin of safety, being accurate to ±12 bits, and it incorporated several conservative engineering features that indicated more reliable operation. Moreover, since only nine analog variables were encoded in the hybrid F-100A program and since the PDP-1 could perform other operations while waiting for the encoding action to be completed, ultra-fast conversion was clearly not a necessity. Restraint was similarly exercised in limiting the interface system to one analog-to-digital converter and one digital-to-analog converter, even though present commercial practice often favors numerous parallel conversion channels in each direction.

In every technology, a level of performance is reached beyond which it becomes increasingly difficult and expensive to effect improvement. For analog computation, this situation occurs in the transition from 0. 1 percent accuracy to 0. 01 percent accuracy. In retrospect, the project's early decision to accept components with a nominal accuracy of only 0. 1 percent reduced the cost of the

analog portion of the system by at least an order of magnitude. To cite the main advantages that resulted from this basic decision:

a. A ±10 volt reference could be used instead of ±100 volts and simple, low voltage, solid-state amplifiers without chopper stabilization became acceptable.

b. Metal film resistors accurate to 0.1 percent (but matchable to better tolerances) could be employed in place of precision wire-wound resistors costing five to 10 dollars apiece. Similar savings were achieved in the choice of power supplies, integrating capacitors, and precision test equipment.

c. Ovens, air conditioning, and other elaborate temperature compensation devices were not necessary.

d. The lower voltage and accuracy requirements permitted the use of photoresistive multipliers in place of servo and quarter-square multipliers. As will be shown, the photoresistive multiplier greatly reduces the number of operational amplifiers needed in the analog sub-system.

e. It was evident from the beginning that the high-speed, pulsed analog interface equipment should be based on a ±10 volt reference, not only because this was a convenient level for sample gates, sample-and-hold gates, encoders, and decoders to operate but also because it resulted in lower impedance and signal levels and wider bandwidths. Selecting the same reference for the standard analog system enabled us to share its ±10 volt and ±15 volt power supplies with the pulsed analog equipment and also to eliminate the scale changing that would otherwise be necessary at the interface between the two systems.

The analog sub-system designed and constructed at M.I.T. for the hybrid demonstration utilized 74 operational amplifiers in implementing the following computing functions:

14 integrators

3 diode function generators

11 photoresistive multipliers (capacity for 77 products; 43 products used)

14 summers

25 inverters

8 division circuits

24 coefficients pots

The total cost of component parts for the analog system, operational amplifiers and power supplies included, was roughly $12,000. Physically, some 38 plug-in cards were involved, each card requiring, on the average, one technician-day to wire. A representative group of cards is shown in Fig. 5.1 and the overall analog sub-system without power supplies is shown in Fig. 5.2.

To duplicate this computing capacity with a commercial machine using quarter-square multipliers would require 115 operational amplifiers servicing the following complement:

14 integrators

3 diode function generators

43 quarter-square multipliers

14 summers

41 inverters

43 operational amplifiers (quarter-square output buffers)

24 coefficient pots

The extra inverters and operational amplifiers are needed because the quarter-square multiplier demands bipolar inputs for both four-quadrant and two-quadrant operations and each product demands an output buffer stage. In terms of a typical commercial computer, the Electronic

Associates TR-48, which is also solid-state and employs a ± 10 volt reference, this computing capacity would require three partially expanded consoles costing over $60,000. This figure does not reflect the extra 34 products which are available, but not used, on the M. I. T. analog computer. An almost fully expanded Electronic Associates 231R, a typical commercial ± 100 volt machine with nominal 0.01 percent accuracy, would have a comparable capacity and would cost over $90,000.

In all fairness, it should be pointed out that the M. I. T. analog computer does not have all the convenience features of the commercial machines, such as pre-patch panels and push-button readout of selected variables. The direct comparison above is merely intended to demonstrate that substantial raw analog computing capacity of moderate accuracy is now obtainable at a relatively low cost. Neither does the discussion imply that 0.01 percent analog accuracies are a fetish based solely on professional vanity and commercial specmanship. In a purely analog simulation, scaling must be such as to accommodate the largest possible value of each variable and this scaling frequently is based on violent, uncommon transient conditions. As a result, most calculations are carried out with voltages considerably smaller than full scale. Whereas a 20 millivolt error corresponds to only 0.01 percent of ± 100 volts, such an error is 2 percent of a signal that normally operates at 1 volt levels. One of the advantages of a hybrid system, of course, is that the calculations involving extreme dynamic range can be performed more accurately on the digital portion of the system.

Carrying this economic discussion to a conclusion, the main elements of the M. I. T. analog-digital interface were:

Beckman 4040A analog-to-digital encoder	$ 4,800
Digital Equipment Corp. logic for implementing operate gates instruction, discrete in-out, analog mode switching.	3,970
Five Burr-Brown wide-bandwidth 1607A amplifiers	975
DEC 722 power supply	305
Adcom sample and hold gates (10 channels)	7,450
Nine dual sample gate cards (18 channels)	920
DEC digital-to-analog converter	690
GPS wide-bandwidth quarter square multiplier with external power supply	2,000
Interface total	$21,110

Thus, in round numbers, the total cost of the interface and analog equipment added to the PDP-1 digital computer for the M. I. T. hybrid demonstration was $35,000. The PDP-1 computer itself with a 4096-word core memory today costs $100,000, so that the total hybrid system represents an investment of $135,000. However, even this figure does not represent the state of the art from a technical or economic viewpoint. The much faster PDP-7, having a 1.75 microsecond memory access time, costs only $45,000 with a 4096 word memory.

For an all-digital simulation, the interface costs would be considerably higher than the hybrid figures cited above. For example, UDOFT had 38 analog outputs, which, if implemented with Adcom sample and hold gates, would alone cost over $28,000. It is clear, in the case of the F-100A demonstration at least, that the peripheral analog equipment more than pays for itself in reduced interface expenditures. In addition, and this is the principal point of the whole demonstration, the analog skeleton operating on a continuous basis effectively increases the real-time capacity of a given digital computer by at least an order of magnitude through the reduction in solution rates. Additional memory must be added to the computer, of course, to actually realize this increase.

In view of the wide range of problems that are being considered for hybrid implementation today, it is impossible to draw general conclusions on system costs. However, utilizing an inexpensive digital computer like the PDP-8 and off-the-shelf interface and analog components, minimal hybrid systems should be obtainable for less than $50,000. Utilizing a fast computer like the PDP-7, very powerful hybrid combinations with adequate display devices should be obtainable in the price range from $100,000 to $300,000. These figures assume that engineering development costs have been minimized by eschewing elaborate custom-built features.

The remainder of this chapter will discuss the various hybrid system components individually.

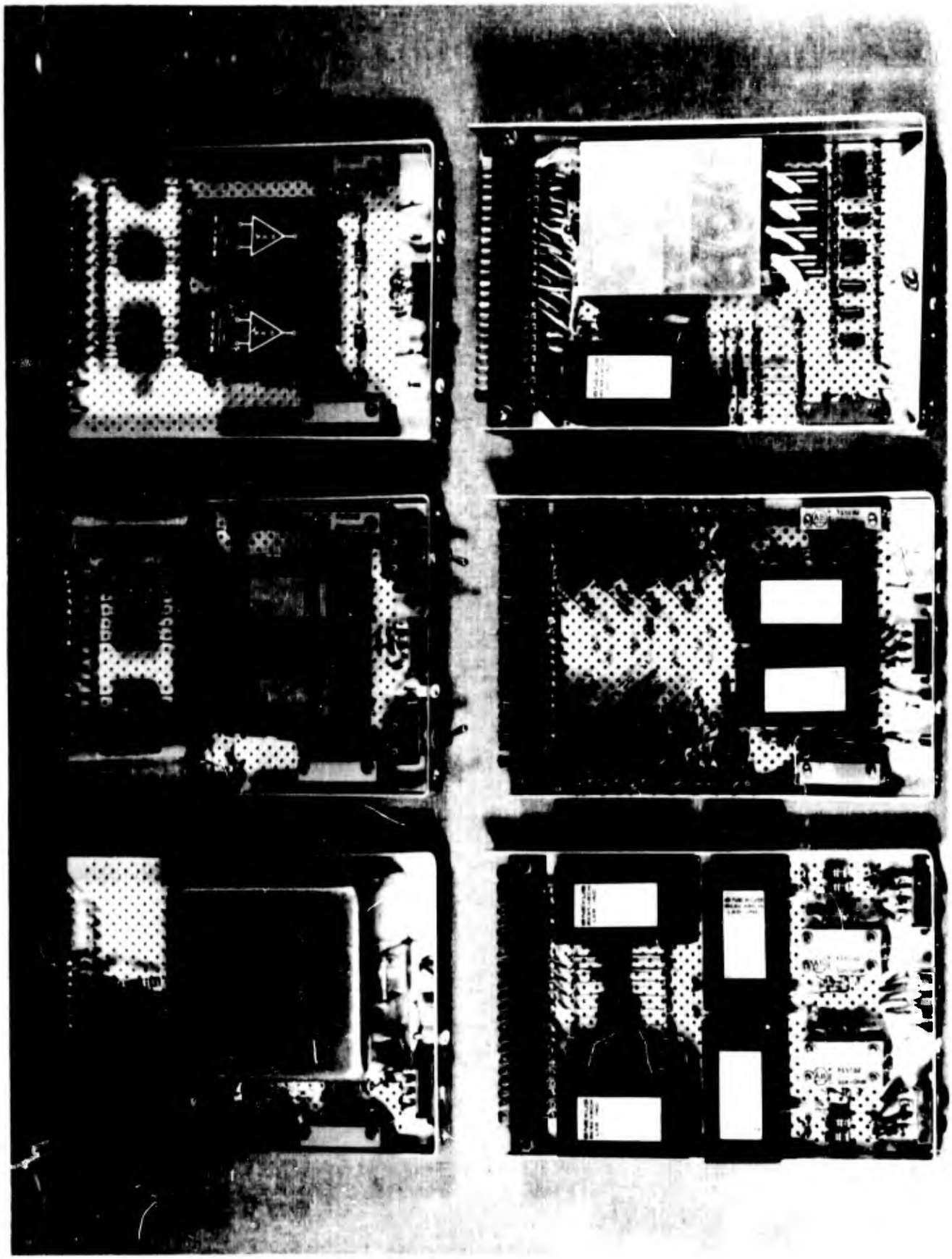


Fig. 5.1 Representative Group of Analog Cards

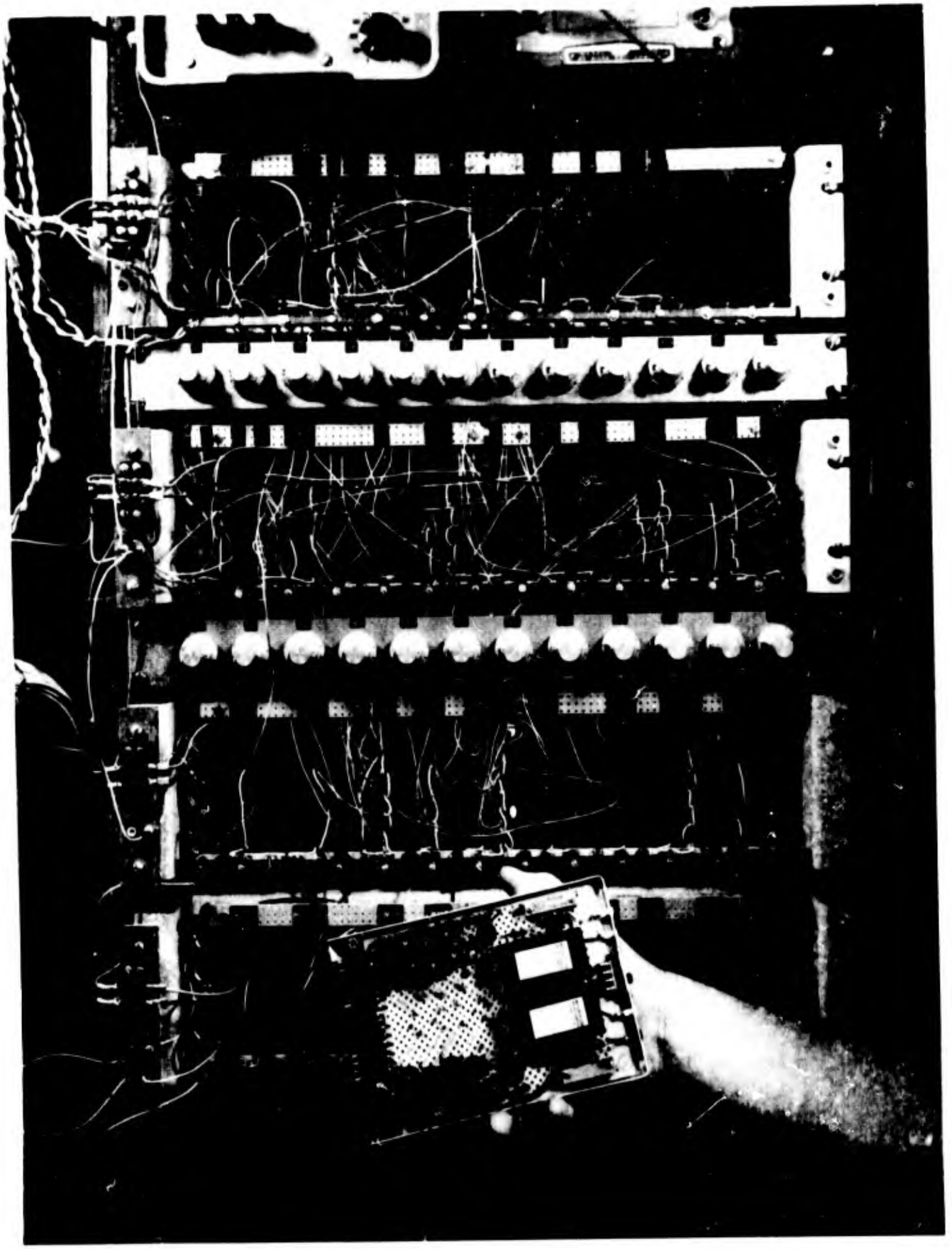


Fig. 5.2 Analog Subsystem

5.2 OPERATIONAL AMPLIFIERS

In 1962, the first solid-state encapsulated operational amplifiers were introduced having voltage and current offset characteristics competitive with chopper-stabilized amplifiers. In adddition to their small size and economy, these amplifiers offered almost instantaneous recovery from overdrive and overload, absence of residual chopper noise, and elimination of the AC chopper drive requirement. After a series of tests, the high gain, differential input SGL-6, SGL-8 and SS12 plug-in units manufactured by Nexus were adopted as the basic building blocks of the M. I. T. analog subsystem. The wide bandwidth Burr-Brown 1607A was later selected for use in the pulsed analog interface. The principal characteristics of these units are listed in Table 5.1. The rapid progress in planar transistor technology has made it possible to better this performance substantially today. By careful design and compensation, temperature coefficients of 0.1 microvolts/°C and 0.1 nanoamps/°C should be obtainable.[21]

Table 5.1

Operational Amplifier Characteristics

	Nexus SGL-6	Nexus SGL-8	Nexus SS 12	Burr-Brown 1607A
Output voltage	±11	±11	±11	±10
Output current (ma)	±20	±20	±3	±30
Power requirements				
Volts	±15	±15	±15	±15
Ma	±30	±30	±10	±12 Quiescent
Typical open-loop gain	500,000	500,000	100,000	31,600
Common mode input voltage	±3	±10	±3	----
Offset voltage temp. coeff. (μv/°C)	10	10	5	±25
Offset current (nanoamps)	40	40	±5	±10
Offset current temp. coeff. (na/°C)	2	2	0.2	3
Unity gain crossover frequency	1 mc	1 mc	1.5 mc	20 mc
Supply/offset voltage rejection ratio	5000	5000	10,000	7500
Price	$80.00	$95.00	$125.00	$195.00

5.3 SUMMING AMPLIFIERS

Two summing amplifiers with their associated input and feedback resistors, and with voltage and current offset balance circuits, are mounted on a single card. The circuit schematic is given in Fig. 5.3.

Mounting terminals are provided on the card for six input channels to each summing amplifier, the ratio of the feedback resistor R_f to the input resistor determining the effective gain of individual channels. Since the analog subsystem was scaled in powers of two, input and feedback resistors were generally chosen from the binary sequence 125K, 250K, 500K, 1 meg, and 2 meg. Occasionally, a non-binary channel gain would be employed to avoid the use of an extra coefficient potentiometer. Metal film resistors accurate to ±0.1 percent, but measured and matched to closer tolerances, were used for all input and feedback applications.

To place both amplifiers in the current balance or voltage balance configuration without disturbing the taper pin problem patching, a double pole-three position switch was mounted on the rear edge of each card as well as appropriate test points. Typically, long-term offsets are less than one millivolt at the output.

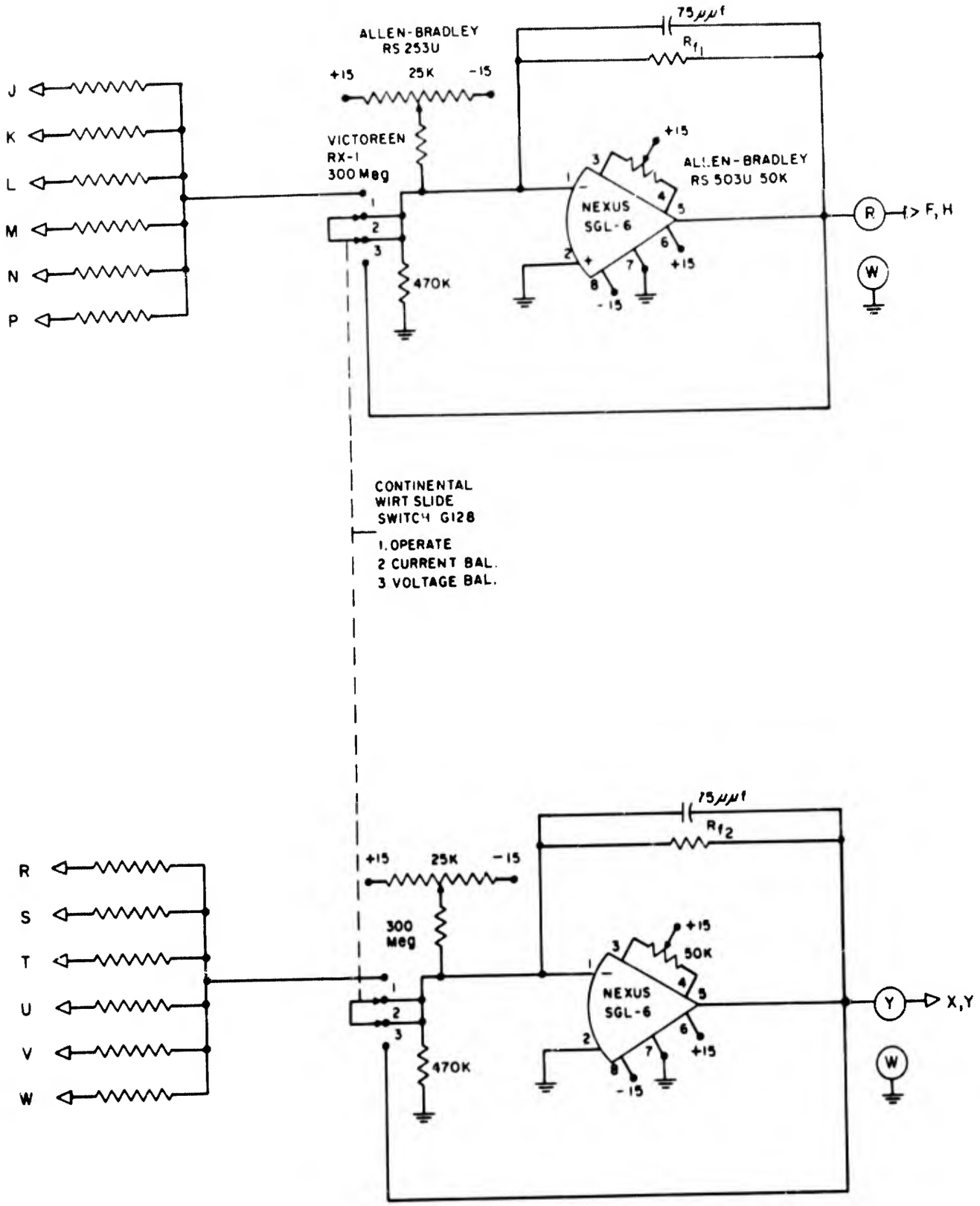


Fig. 5.3 Dual Summing Amplifier

5.4 QUAD INVERTERS

Because of the smaller number of input and feedback resistors required, it was possible to mount four inverters on a single card. The quad inverter schematic is given in Fig. 5.4. As in the case of the summing amplifiers, provisions were made for current and voltage offset balance without disturbing the problem patching. The small feedback capacitor was included to give an added margin of safety against instability. In special situations, a few inverters were employed with other than unity gain. In addition, the inverter buffer for the horizontal stabilizer control input included a diode limiting circuit.

5.5. DUAL INTEGRATORS

Two integrators were mounted on a card, complete with integrating capacitors, mode switching relays, current and voltage balance circuits, and input networks. The circuit schematic is presented in Fig. 5.5. Nominally, four input channels were provided for each integrator, although it is relatively simple to shift channels from one integrator to the other if necessary. The standard feedback element was a 0.5 microfarad polystyrene capacitor with a ±0.1 percent tolerance. With this capacitance value, a 2-megohm input resistor constituted a unity gain channel (1 volt/sec/volt input). For larger gains up to 10 volts/sec/volt, input resistances down to 200K were employed. For gains greater than this, a 0.1 microfarad feedback capacitor was installed.

The mode switching relays No. 1 and No. 2 could be activated by the digital computer or by external manual switches. The correspondence between relay states and integrator mode is as follows (a bar indicating absence of relay excitation):

operate	$\bar{1} \cdot \bar{2}$
hold	$1 \cdot \bar{2}$
reset	$1 \cdot 2$

Under manual control, the hold mode could always be interposed between the reset and operate modes, thus avoiding switching transients leading to false initial conditions. Under digital control, however, the mode switching command was much faster than the relay operation times, consequently a delay had to be inserted between the deactivation of relay No. 2 and deactivation of relay No. 1 to guard against relay No. 1 closing before relay No. 2 opened in the transition to operate.

For system test purposes, it was found convenient to lock individual integrators at various initial conditions. To accomplish this, toggle switch reset circuits were added to each card capable of over-riding the external mode control signals.

With a feedback capacitor, there exists the possibility of exceeding the maximum allowable common mode voltage (± 3 volts) at the input of the SGL-6 (for example, if the output is shorted to ground with the capacitor charged to ± 10 volts) For this reason, the Nexus SGL-8 amplifier with a common mode limit of ± 10 volts is employed in all integrators. Later Nexus amplifiers incorporate internal protective networks to prevent such amplicides.

5.6 LONG-TERM INTEGRATORS

When long-term integrations are encountered, such as the integration of ground velocity components to obtain aircraft position, or when the problem scaling requires integrator gains substantially less than unity, as arises in the integration of rate of climb to obtain altitude, the best computational results are procured with a very large feedback capacitor. In the first case, the large capacitor reduces the predominant drift error due to the finite input current (I_g) of the operational amplifier.[22] The output voltage due to this current is given approximately by

$$e_o = \frac{I_g t}{C}$$

With an amplifier input current of one nano amp, an integrator having a 0.5 microfarad feedback capacitor would exhibit an output voltage drift of 7.2 volts over a one-hour period. The use of a 10-microfarad feedback capacitor reduces this drift by a factor of twenty to 0.36 volts.

In the second situation, where the problem scaling demands substantial attenuation of the signal at the integrator input, the use of a large capacitor enables the designer to achieve fractional values of integrator gain (1/RC) without resorting to excessively high values of resistance (R) in the input channel, or without preceding the integrator by a coefficient potentiometer that reduces the input signal to unacceptably low levels.

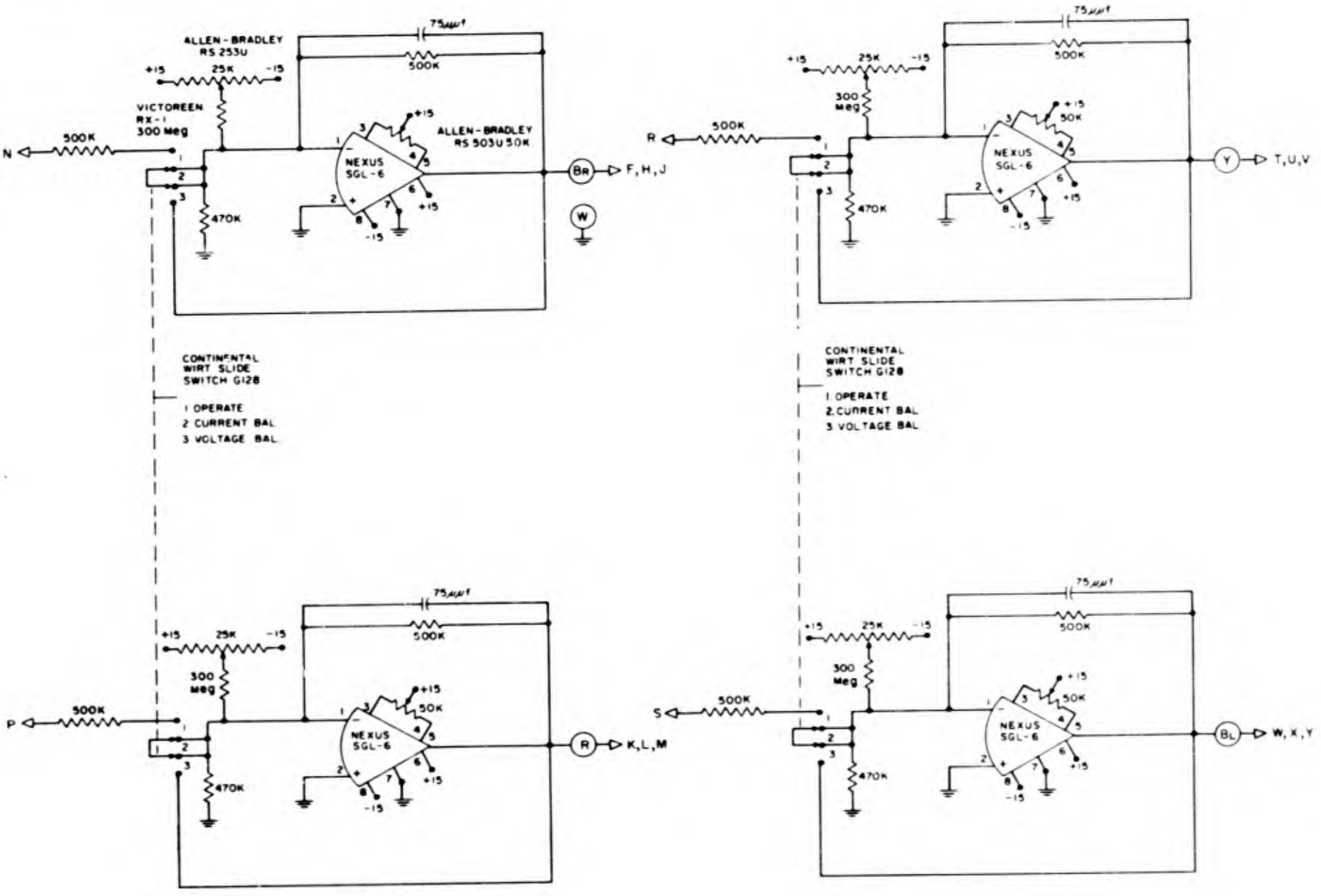


Fig. 5.4 Quad Inverter

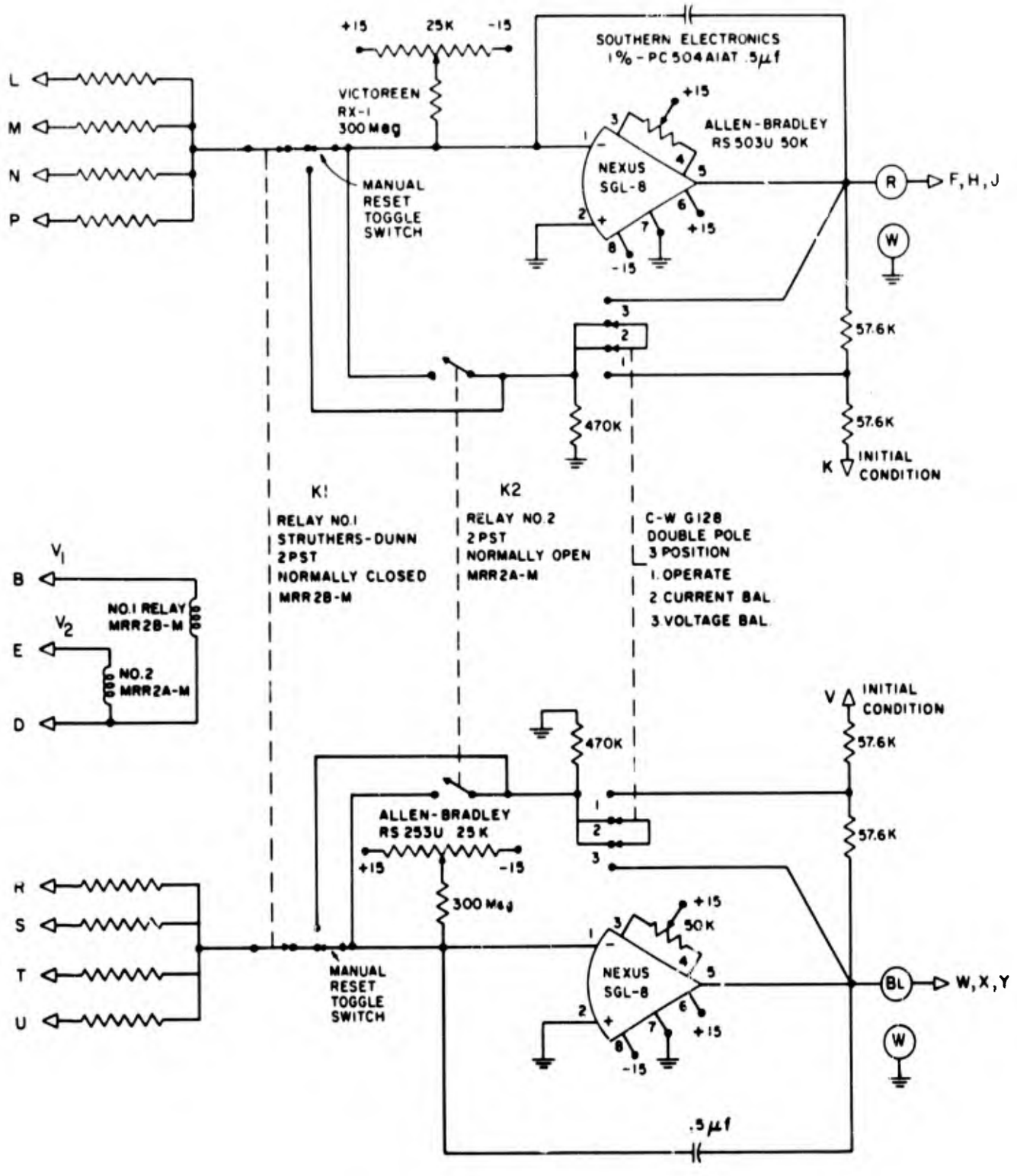


Fig. 5.5 Dual Integrator

For such occasions as these, a special integrator was designed employing a 10-microfarad polystyrene feedback capacitor with 0.1 percent tolerance. Because of the size of the capacitor only one integrator could be mounted on a single card. The current and voltage balance circuits and the mode switching features are the same as for the standard dual integrators. The circuit schematic is given in Fig. 5.6.

5.7 VARIABLE DIODE FUNCTION GENERATOR (DUAL)

Although in a hybrid system it is generally preferable to generate nonlinear functions in the digital domain, situations occasionally arise in which separate sample and hold gates would be required to store individual functions used in the analog domain. In such cases, it is more economical to employ a diode function generator directly. The three nonlinearities in the F-100A analog configuration so generated are air density, cos θ, and the nosewheel pitch moment on the ground.

Two diode function generators, each with five segments and an offset voltage, are mounted on a single card. Up to three segments of one unit, however, can be transferred to the other diode function generator. Each segment has slope and breakpoint adjustments as indicated in the circuit schematic Fig. 5.7.

5.8 PHOTORESISTIVE MULTIPLIER

The generation of nonlinear functions of one or more variables and multiplication have always been the most awkward and expensive of the basic operations in analog computation. In recent years, the problem of large scale function generation has found a satisfactory solution in the application of digital techniques, but analog multipliers continue to constitute a limitation in accuracy and an economic limitation because of the large number of operational amplifiers necessary to service each product. Operating under a rather limited budget, the Electronic Systems Laboratory simulation project has always been painfully aware of these ficiencies and, consequently, has investigated a number of alternative solutions to the multiplication problem.[26,28,29,30,31,59] The Air Force approval to proceed with the construction of the hybrid demonstration system finally brought the situation to a head, however, since the successful completion of the system with the funds allocated clearly required a more economical means of analog multiplication than the standard quarter-square technique.

Under normal cruising and high-speed flight conditions, the thrust-drag summation is the principle relationship that determines aircraft performance. By carrying out this calculation in the digital computer, we eliminated the only analog multiplication $(\rho V^2/2)\, C_D$ requiring extreme accuracy. The remaining 43 products had three noteworthy characteristics:

1. Only moderate accuracy (better than 1 percent) and relatively low frequency response (less than 2 cps) were demanded.

2. Over half of the products were two quadrant rather than four quadrant.

3. All of the products involved one of 11 key analog variables and, characteristically, each of these variables was a factor in four, five, or six products.

Using these characteristics as a guide, the senior author devised the photoresistive multiplier employed in the demonstration system. The description of this device presented here must be regarded as an interim report, since the tight schedule on system development and test did not permit a full exploration of various refinements in the multiplier design.

In its present form, the photoresistive multiplier is capable of producing seven products of one common variable. The principles of its operation may be readily ascertained by referring to the circuit schematic Fig. 5.8. An array of eight photoconductive cells R_{P_1}, R_{P_2}... R_{P_8} are mounted in a light chamber illuminated by a single miniature lamp. Variations in the voltage (e_L) applied to this lamp cause variations in the level of illumination on the cells and this, in turn, changes the effective resistance of the cells. Each cell is a resistive element in a separate voltage divider circuit, the output of which also changes with the illumination level. The voltage divider (FH), incorporating the master cell R_{P_5}, is employed in a closed-loop configuration with a high-gain operational amplifier to vary the lamp voltage as a function of the multiplier common input Y. Steady-state is achieved when the output of the master divider circuit balances out the input Y. The feedback arrangement forces the lamp voltage and the level of illumination to be such that the master cell resistance R_{P_5} satisfies the following equilibrium relationship:

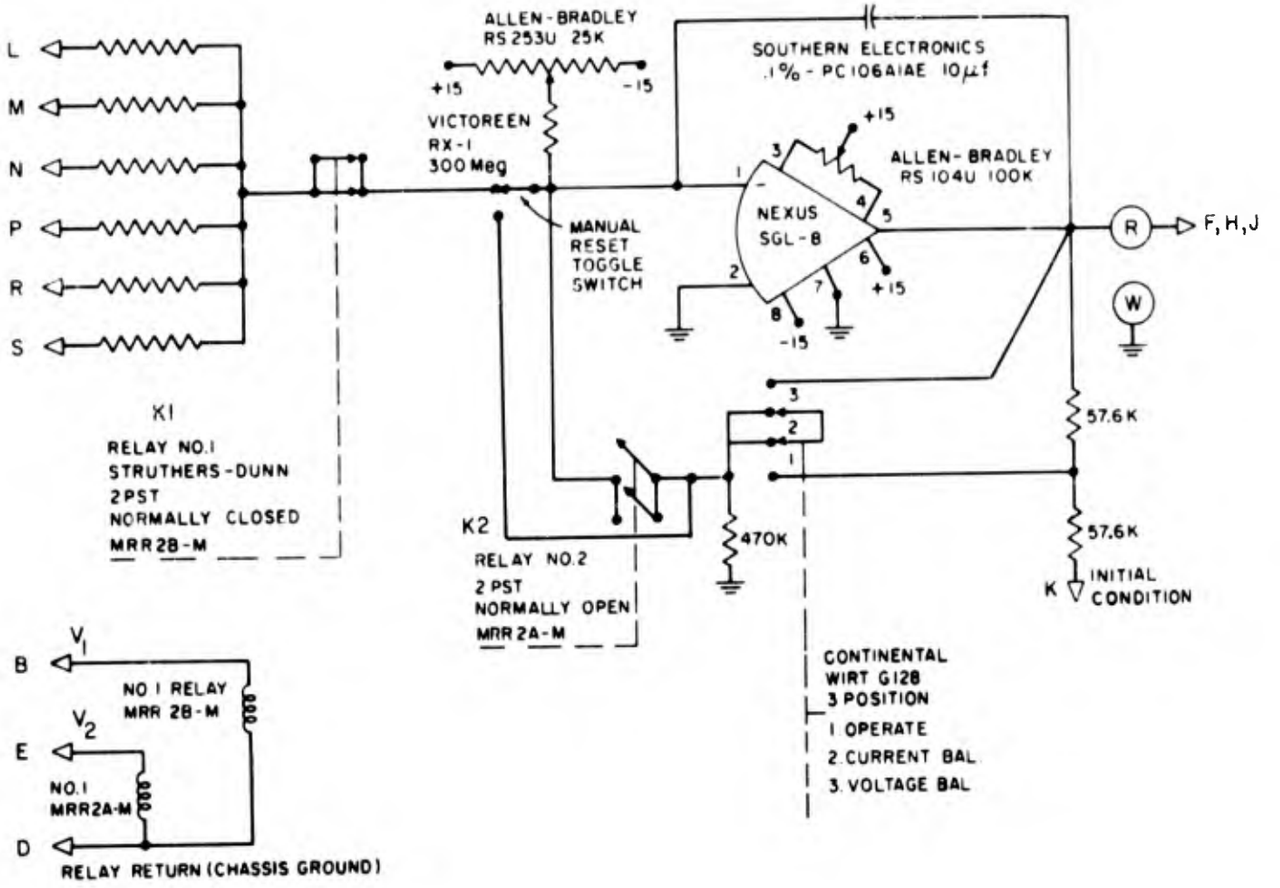


Fig. 5.6 Long-Term Integrator

① Diode Inserted with Desired Polarity Between Flea Connectors on Board

② Diode Inserted Here to Add More Segments to DFG No. 2

③ Jumper Patched to Proper Pin Jack for Desired Breakpoint Polarity

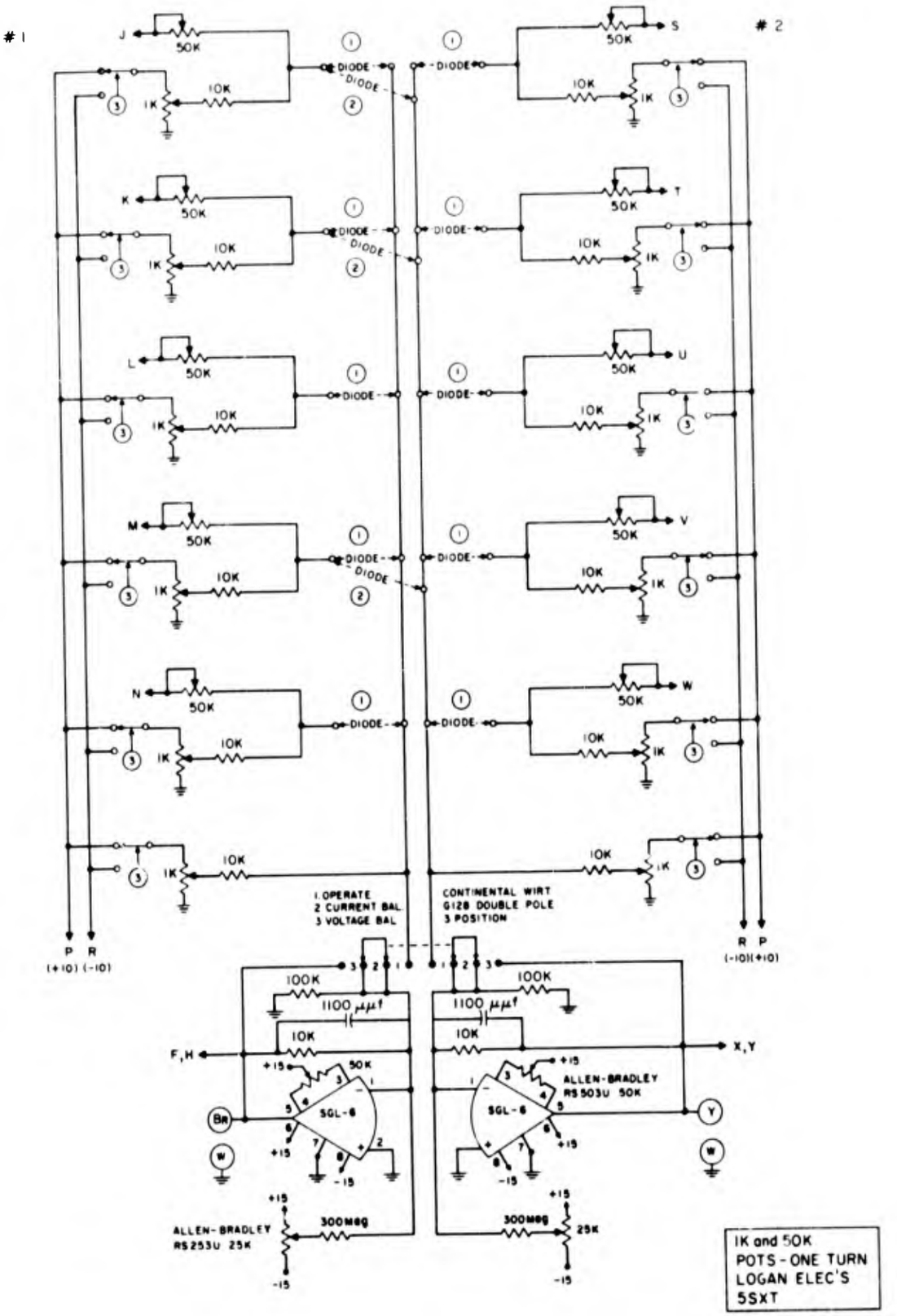


Fig. 5.7 Dual Diode Function Generator

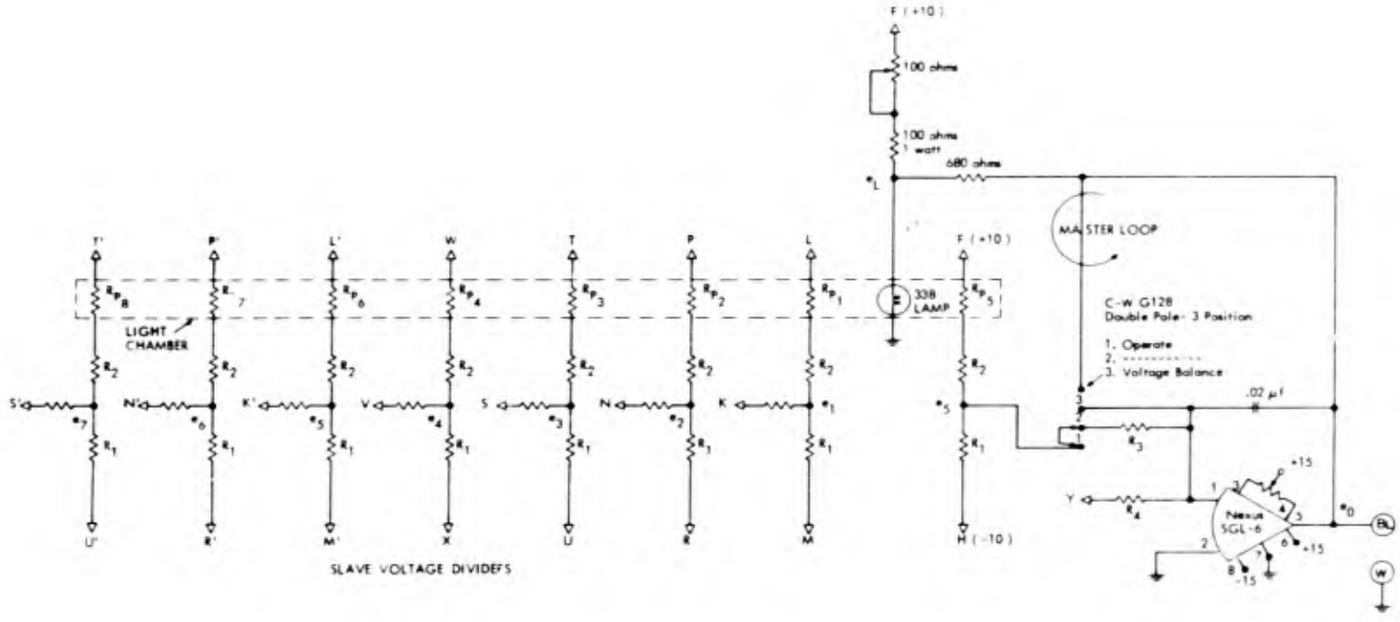


Fig. 5.8 Four Quadrant Photoresistive Multiplier

$$e_5 = -\frac{R_3}{R_4} Y = \frac{10 R_3 [R_1 - R_{P_5} - R_2]}{R_1 R_3 + (R_{P_5} + R_2)(R_3 + R_1)} \tag{5.1}$$

Ideally, the seven slave cells are subject to the same illumination level as the master cell and, ideally, their resistance values follow the resistance of the master exactly. As a consequence, a slave circuit excited by the multiplicand voltage ± V produces an output proportional to the product Y V. To illustrate this, assume that the slave voltage divider L M has the multiplicand + V applied at terminal L and its inverse - V at terminal M. If the loading at the output terminal of the divider is R_3 as in the case of the master divider, the output voltage e_1 is given by

$$e_1 = \frac{V R_3 [R_1 - R_{P_1} - R_2]}{R_1 R_3 + (R_{P_1} + R_2)(R_3 + R_1)} \tag{5.2}$$

If slave cell one has tracked the master cell five exactly

$$R_{P_1} = R_{P_5} \tag{5.3}$$

Hence from Eq. 5.1 and Eq. 5.2

$$\frac{R_3 [R_1 - R_{P_1} - R_2]}{R_1 R_3 + (R_{P_1} + R_2)(R_3 + R_1)} = \frac{e_1}{V} = \frac{R_3 [R_1 - R_{P_5} - R_2]}{R_1 R_3 + (R_{P_5} + R_2)(R_3 + R_1)} = -\frac{R_3}{R_4} \frac{Y}{10} \tag{5.4}$$

from which

$$e_1 = -\frac{R_3}{R_4} \left(\frac{YV}{10}\right) \tag{5.5}$$

The slave output e_1 is directly proportional to the product Y V. A positive sign for the product is obtainable simply by reversing the excitation signals at L and M. In a similar fashion, if the other slave dividers are excited by the multiplicands ± X, ± W, etc., outputs are produced proportional to the products Y X, Y W, etc. The gain stage necessary to bring each product up to full scale (± 10 volts) can be associated with a summing amplifier or any other computing element following the multiplication; hence this stage cannot be properly charged against the multiplier itself.

After extensive experimentation with various photoresistive elements, starting originally with the Raytheon lamp - photocell combination (Raysistor), the Clairex cadmium sulfide CL605 L - 020 unit was finally selected for use in the demonstration multipliers. The spectral response of this cell closely matches the response of the human eye, peaking at 5500Å. A cell consists of a light-sensitive grid on an aluminum oxide substrate mounted in a hermetically-sealed glass cylinder 0.245 inches in diameter and 0.50 inches in length. The maximum voltage rating is 60 volts and the maximum allowable power dissipation is 75 milliwatts. The variation of cell resistance as a function of illumination level for the standard C L605 L cell is shown in Fig. 5.9. In order to reduce the variation of resistance with applied voltage, the C L605 L-020 with a somewhat thicker 20-mil grid was employed in the actual multiplier, but the resistance plot for this unit is similar to that shown.

Referring to the circuit diagram Fig. 5.8, with R_1 = 86.6 K, R_2 = 57.6 K, R_3 = 200 K, and R_4 = 2 megohms, the values of master cell resistance R_{P_5} corresponding to the full range of Y inputs are:

	Y = + 10 volts	Y = 0 volts	Y = - 10 volts
e_5	- 1 v.	0 v.	+ 1 v.
R_{P_5}	48.75 K	29 K	12.98 K

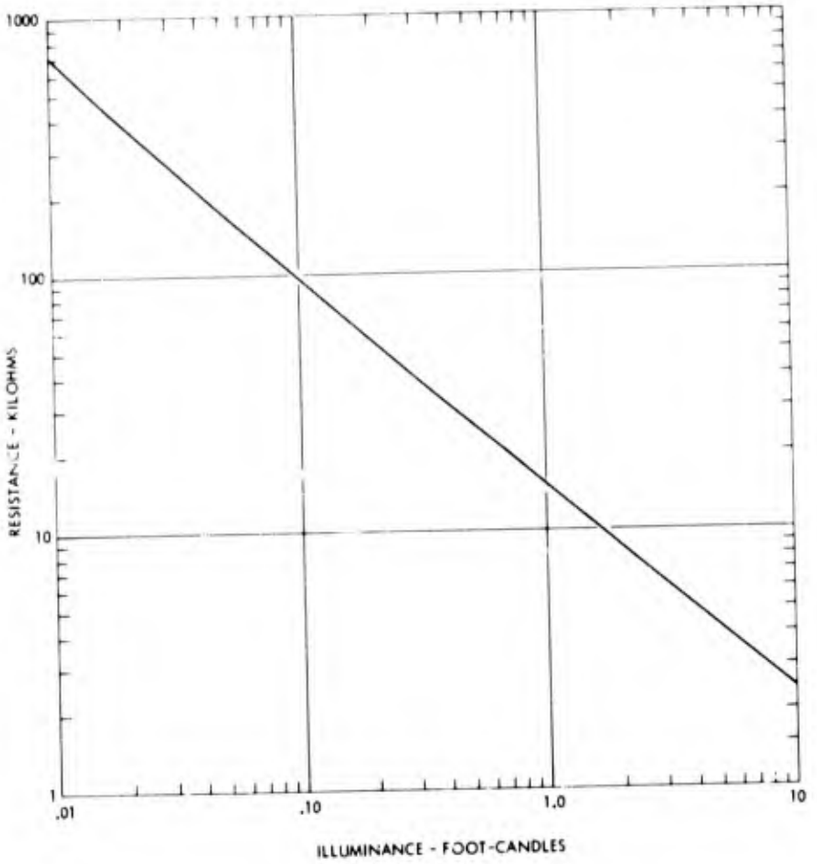


Fig. 5.9 Resistance of Clairex 605 L Cell vs. Illumination Level

It was found experimentally that the Chicago Miniature T - 1-3/4 incandescent lamp type 338 rated at 2.7 volts and 60 milliamps could produce illumination levels within the light cavity corresponding to this range of cell resistance. The voltage-current characteristic of the 338 lamp is given in Fig. 5.10. Superimposed on the plot are load lines for the bleeder circuit, which supplies most of the current to the lamp, and the 680 ohm feedback circuit at the saturation limits of the operational amplifier (± 12 volts). For proper functioning, the range of lamp operating conditions indicated on this plot must correspond to a range of photocell resistances from 12.98 K to 48.75 K. In practice, to utilize the amplifier output range to full advantage, the bleeder circuit resistance is adjusted to produce a zero output at e_5 when the operational amplifier output e_0 is zero.

There are numerous fine points in the design that cannot be treated in detail here, but which warrant mention. The primary problem, of course, is exact matching of all eight cells of an array. Clairex, as a special service, will test and match groups of cells to within ± 8 percent at a standard illumination of two foot-candles. Within each such group, further selection is possible to yield sets of eight matched to better than ± 1 percent. Tailoring of cell characteristics with external resistors is a further possibility. Finally, the distance of each cell of an array from the light source can be set individually by a fine screw adjustment.

Temperature variations from cell to cell are minimized by mounting the entire array in a solid block of aluminum. Ideally, even though photoconductive characteristics are temperature sensitive, if all cells are at a common temperature, mismatches due to temperature variation should be minimized. Cadmium sulfide has a better temperature coefficient than cadmium selenide.

Photocell resistences are also generally dependent on the long-term light history of a cell. Figure 5.11 indicates the variation in conductance at a given light level depending on whether the measurement occurs immediately after a prolonged exposure to darkness or a prolonged exposure to 30 foot-candles of illumination. One of the reasons for selecting a cadmium sulfide cell instead of cadmium selenide cell was the substantial reduction in this light history phenomenon. However, since all the cells of an array are exposed to the same light history, this effect is secondary. For maximum accuracy, it is probably advantageous to leave the multiplier lamps on at all times.

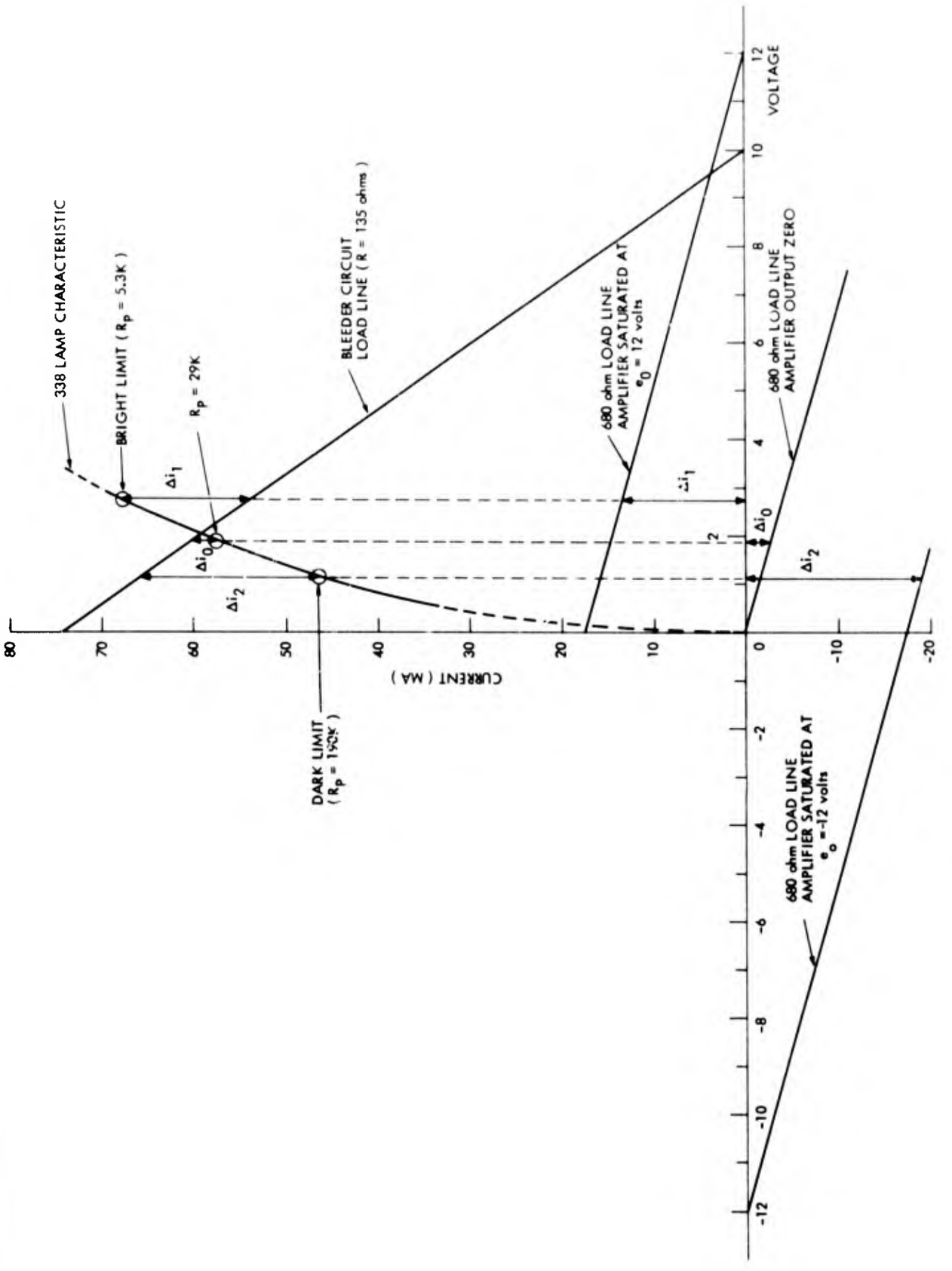


Fig. 5.10 Type 338 Lamp Characteristics with Bleeder and Amplifier Load Lines Superimposed

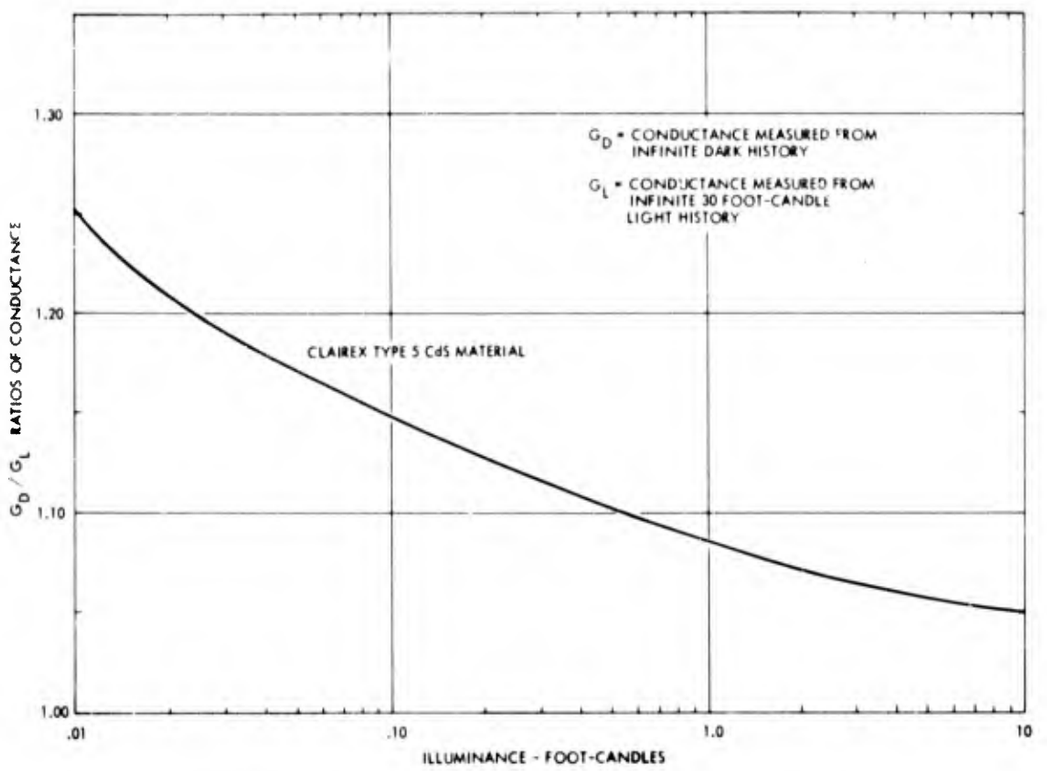


Fig. 5.11 Variation of Conductance with Light History

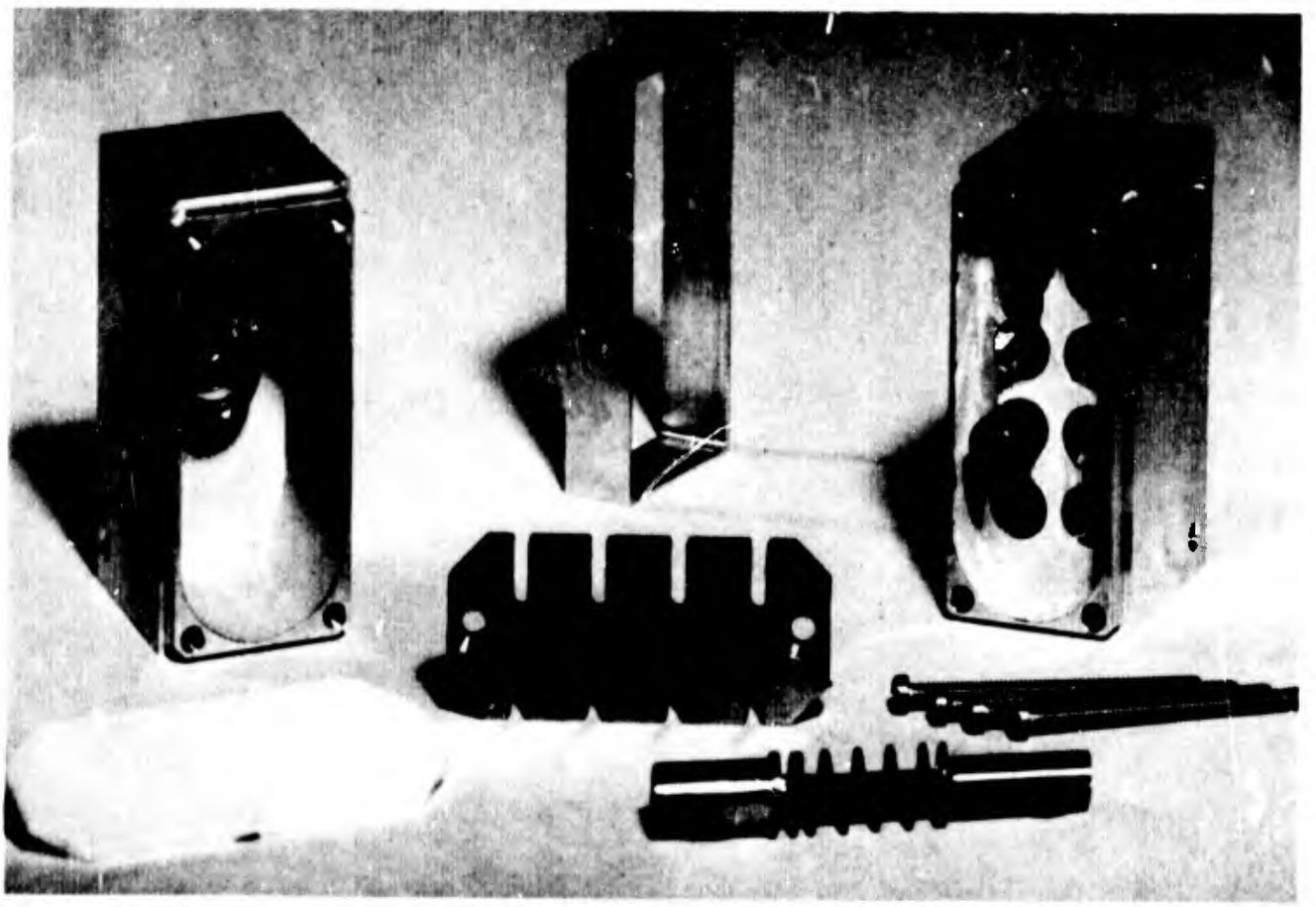


Fig. 5.12 Disassembled Light Cavity

To assure a more even illumination of the cells, a translucent plastic or glass divider may be inserted in the light chamber between the lamp and the cells and a frosted cap may be employed on the lamp. For the same purpose, the chamber itself is coated with a special white diffusion paint. A disassembled light chamber is shown in Fig. 5.12.

The bandwidth of the multiplier is limited primarily by the response of the lamp and the cell, the response time of the latter being a function of the light level as follows:

Light Level (foot-candles)	Rise Time (sec's)	Decay Time (sec's)
0.01	2.8	1.3
0.1	0.30	0.22
1	0.07	0.06
10	0.02	0.02
100	0.007	0.014

Obviously, to obtain the fastest operation of the cell, the level of illumination should be as high as possible. Cadmium selenide cells are generally faster than cadmium sulfide cells, but this advantage does not offset their inferior temperature and hysteresis characteristics.

On the other hand, miniature lamps with thin filaments operating at low currents are the fastest, hence the requirements for high light levels and quick lamp response are in conflict. A change in light output from 10 percent rated value to 90 percent might take 30 milliseconds in a typical lamp. Such rise times, of course, correspond to substantial phase shifts at low signal frequencies. As a consequence, to insure the closed-loop stability of the master cell control circuit, a feedback capacitor is employed to reduce the gain of the operational amplifier at high frequencies. The net result of these factors is that the common multiplier input signal Y is limited to a low frequency range comparable to that achieved in servomechanisms. The multiplicand signals, however, merely excite a resistive divider network, hence these signals may be in the kilocycle range.

Many of the practical problems mentioned above have been discussed in greater detail in the M.I.T. thesis submitted by Lt. (j.g.) Thomas Anderson.[60]

One of the main advantages of the photoresistive multiplier is that the two-quadrant version does not require both the multiplicand voltage and its inverse for each product. The circuit connections when the multiplier Y is restricted to one polarity are shown in Fig. 5.13. As in the four-quadrant circuit, the value of Y controls the light level by means of a feedback loop. Steady-state is achieved when the output (e_5) of the voltage divider balances out the input Y. In the two-quadrant case, however, Y is always negative or always positive, hence the master divider network can be returned to ground and need only be excited by either the +10 or the -10 volt reference supply. Similarly, the slave networks are also returned to ground and need only be excited by the multiplicand voltage.

With a positive Y input and the master divider excited by the -10 volt reference, the steady-state conditions for the master loop are:

$$e_5 = -\frac{R_3}{R_4} Y = -\left(\frac{10 R_1 R_3}{R_1 R_3 + R_{P_5}(R_1 + R_3)}\right) \tag{5.6}$$

If the loading at the output terminal of the slave divider is also R_3 and if the slave network is excited by the multiplicand V, the output voltage e_1 is given by

$$e_1 = \frac{V R_1 R_3}{R_1 R_3 + R_{P_1}(R_1 + R_3)} \tag{5.7}$$

If slave cell one has tracked master cell five exactly

$$R_{P_1} = R_{P_5} \tag{5.8}$$

Hence from Eqs. 5.6 and 5.7

$$\frac{R_1 R_3}{R_1 R_3 + R_{P_5}(R_1 + R_3)} = \frac{R_3}{R_4}\frac{Y}{10} = \frac{R_1 R_3}{R_1 R_3 + R_{P_1}(R_1 + R_3)} = \frac{e_1}{V} \tag{5.9}$$

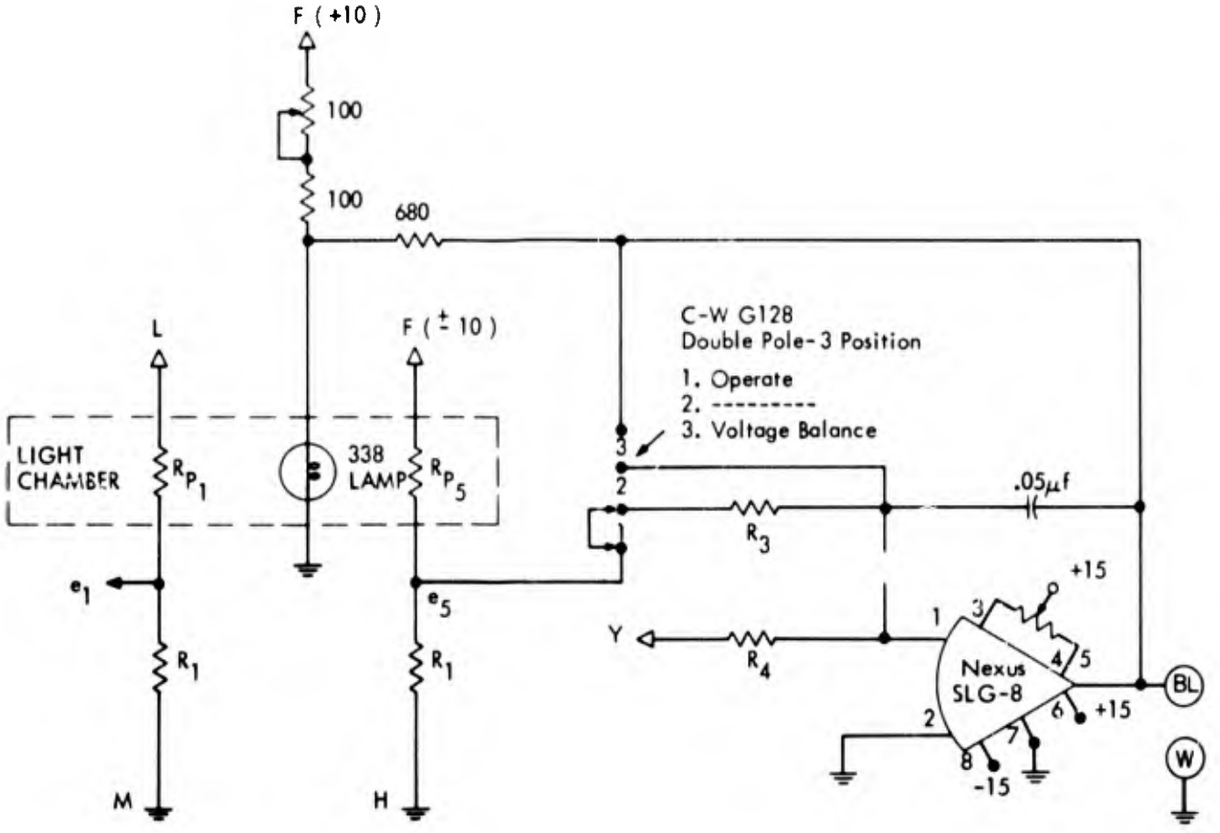


Fig. 5.13 Two-Quadrant Photoresistive Multiplier

From which

$$e_1 = \frac{R_3}{R_4} \frac{YV}{10} \tag{5.10}$$

The slave output is directly proportional to the product YV. A negative product is obtainable by using the inverse of V to excite the voltage divider.

One problem with the two-quadrant circuit is immediately apparent. As the multiplier Y approaches zero, the photocell resistance R_{P_5} must become very much larger than R_1 if the output of the divider network is to approach zero. In practice, this condition can only be approximated, a typical set of values being $R_1 = 20K$ and $R_{P_5}(max) = 20$ megohms, for which the master divider output would be 10 millivolts. This is an acceptable deviation from the ideal value of zero.

5.9 QUAD DIVISION CIRCUIT

Two positive variables, aircraft velocity and cos θ, are involved in division operations, which are implemented in the analog subsystem by using two-quadrant photoresistive multipliers in the feedback configuration shown in Fig. 5.14. Each division requires only a single slave network of the multiplier controlled by the divisor and an operational amplifier to complete the feedback loop. Assuming that LM is one of the slave voltage dividers in a two-quadrant multiplier controlled by the positive variable Y, at steady-state the output (e_1) of this network is forced to obey the relation:

$$e_1 = -\frac{R_5}{R_6} X \tag{5.11}$$

From the analysis of the two-quadrant multiplier (Eqs. 5.6 through 5.10), e_1 can also be expressed in terms of the output voltage (e_0) of the feedback amplifier which excited the network LM:

$$e_1 = \frac{R_3}{R_4} \frac{Y e_0}{10} \tag{5.12}$$

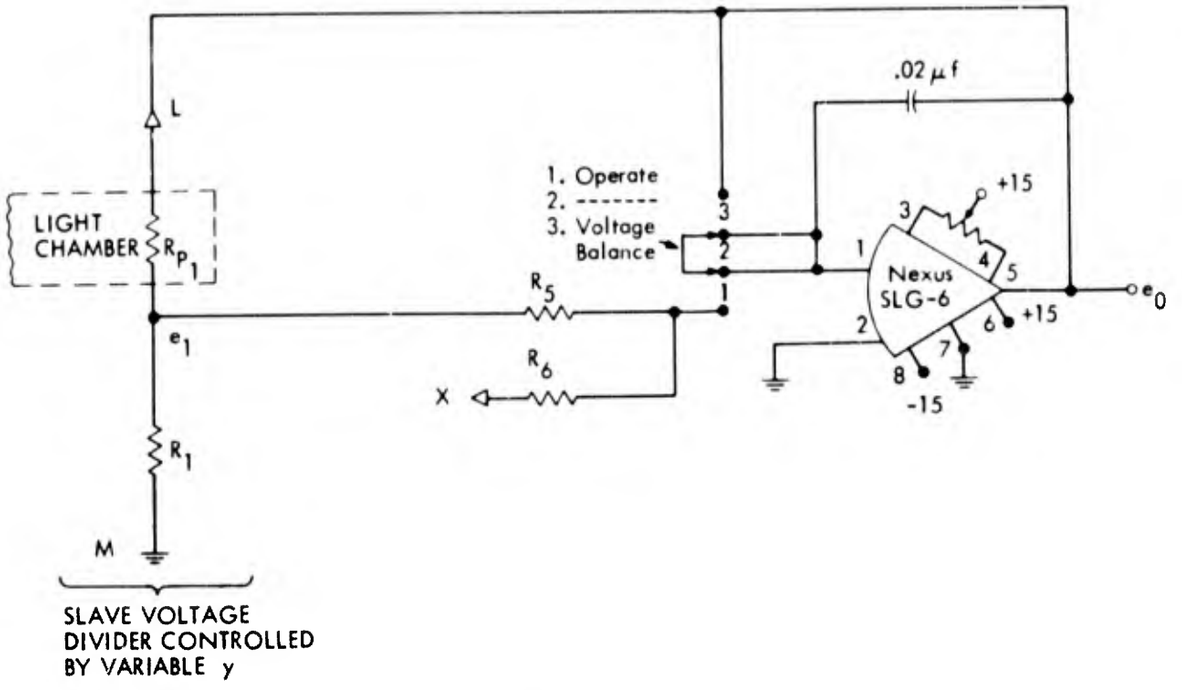


Fig. 5.14 Division Circuit

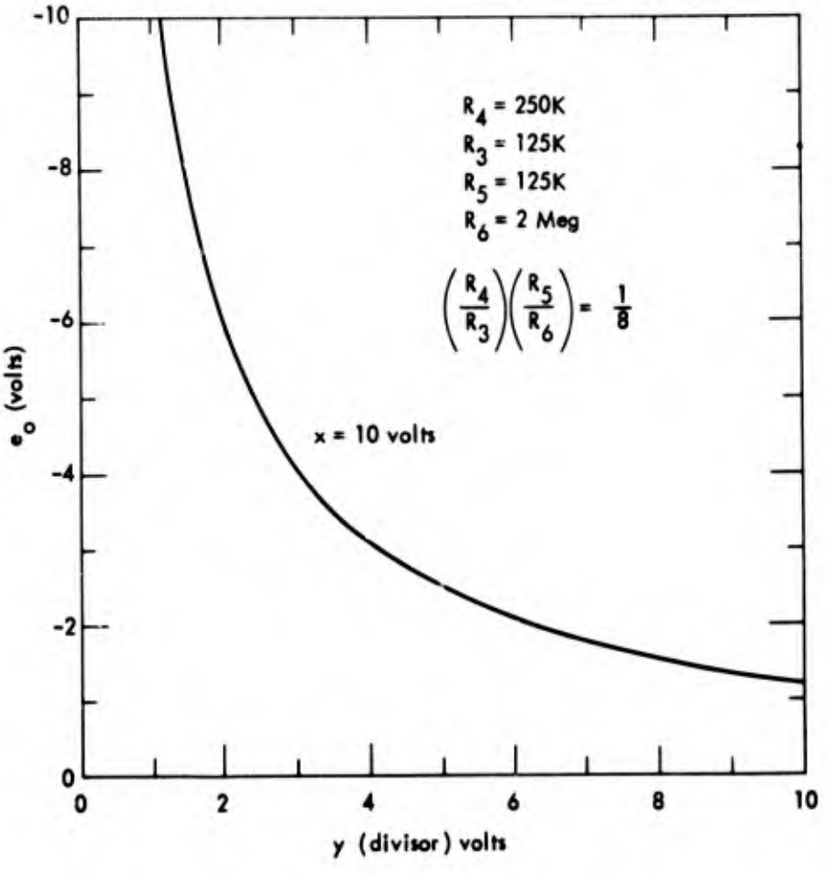


Fig. 5.15 Typical Division Circuit Characteristic

Equating Eqs. 5.11 and 5.12, then solving for e_0, one obtains

$$e_0 = -10\left(\frac{R_4}{R_3}\right)\left(\frac{R_5}{R_6}\right)\frac{X}{Y} \tag{5.13}$$

The amplifier output is thus proportional to the quotient X/Y, the constant of proportionality being controlled by the resistance ratios $\frac{R_4}{R_3}$ and $\frac{R_5}{R_6}$. For the case in which $\left(\frac{R_4}{R_3}\right)\left(\frac{R_5}{R_6}\right) = \frac{1}{8}$, the quotient e_0 is plotted in Fig. 5.15 as a function of the divisor Y. Four of the operational amplifiers used to implement division loops are mounted on a single card. The circuit diagram for this quad division card is given in Fig. 5.16.

5.10 PULSED ANALOG MULTIPLIER

Historically, one of the chronic problems in the field of high-speed analog computation has been that of providing an accurate, wide-bandwidth multiplier. Considerable research effort has been expended in this area, both at M.I.T.[23,24,25,26,27,28,29,30,31] and elsewhere.[32,33,34] At the present time, the state of the art is represented by the GPS quarter-square multipliers having a half-power (-3db) bandwidth of 400 kc and a static error of less than 75 millivolts over the full scale of ±100 volts. On special order, GPS recently produced an ultra-fast quarter-square multiplier with a 10-megacycle bandwidth, but with a static accuracy of only 0.5 percent of full scale (±40 volts).[35] Such a bandwidth corresponds to 1° of phase shift at 2 megacycles. The amplifiers employed in this unit had a closed-loop bandwidth in excess of 30 megacycles.

The multiplier selected for the pulsed analog interface was a GPS model MU-500E, a quarter-square unit modified to operate at ±10 volts full scale. A block diagram of this device is given in Fig. 5.17. Each of the diode shaping networks utilizes eleven straight-line segments in matching a square characteristic. Static tests on the multiplier verified the accuracy specification of 0.25 percent of full scale. Settling time for a 10-volt step input was less than 10 microseconds.

The GPS multiplier and its associated power supply (Philbrick R-100B: ±300 volts) are the only units in the M.I.T. demonstration system employing vacuum tubes. A solid-state design of comparable performance could have been developed by the project based on the earlier work of Dumas,[28] Leith,[29] and Solbakken,[31] but limitations of manpower and time prevented such a development.

5.11 SAMPLE AND HOLD GATES

Another classic circuit problem of hybrid and high-speed analog computation has been the development of sample and hold gates having a large ratio of hold time to sample time. Technically, this problem is closely related to the design of electronic mode controls for analog integrators and of boxcar circuits in radar work.

Generally, sample and hold gates have been based on the two approaches shown in Fig. 5.18; the first, an open-loop series connection of switch, capacitive storage element, and output buffer, and the second, a closed-loop configuration in which the storage capacitor is placed in the feedback path of an operational amplifier. In both arrangements, providing a large charging current through the transmission gates, low leakage off the capacitor in the hold state, overall linearity, drift stability, and acceptable switching noise presents rather challenging design problems. As a consequence, considerable research effort has been expended in this area.[4,6,9,20,36,37,38,39,40] Unfortunately, as is the case with many interesting technical fields evolving rapidly, the best work done under commercial sponsorship has often not been documented in the literature. Current commercial practice favors variations of the closed-loop configuration with the sample times ranging from 5 to 15 microseconds and hold times (0.1 percent decay) from 5 to 50 milliseconds. To operate synchronously with a digital computer, a sample time comparable to the memory access time of the computer is desirable. In the latest digital machines, core memory access times run from 1 to 5 microseconds. In simulation work, the minimum hold time is determined by the solution rate. For example, the all-digital F-100A simulation, run at 20 solutions per second, would require a 50 millisecond hold time. For the hybrid F-100A simulation, however, run at only one solution per second, a one-second hold would be necessary to avoid extraneous gate upkeep operations by the digital computer. Moreover, simulation studies are frequently interrupted for output displays, data print-outs, and program changes. These interruptions might last for several minutes, hence corresponding hold times would be convenient.

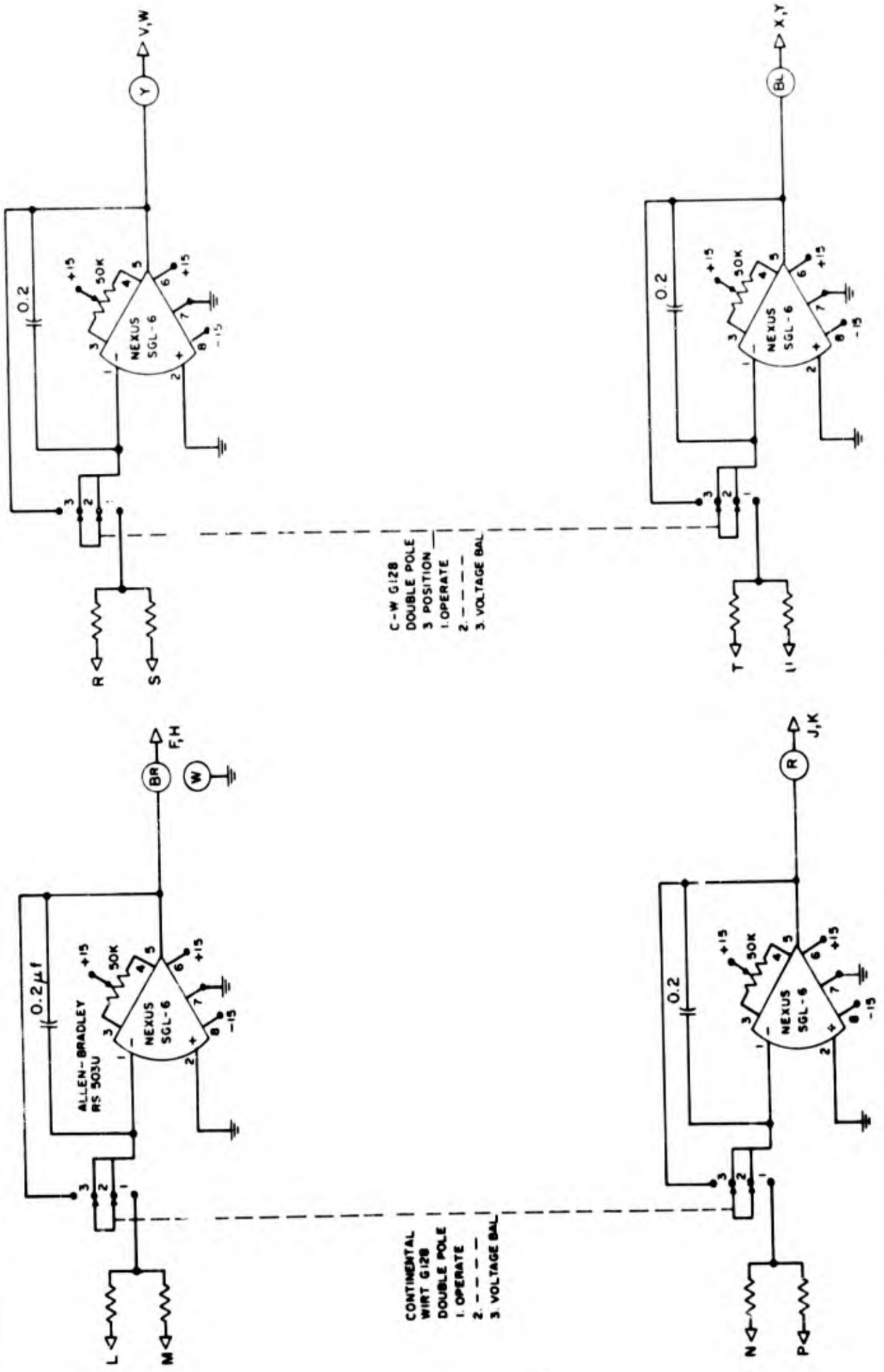


Fig. 5.16 Quad Division Circuit Card

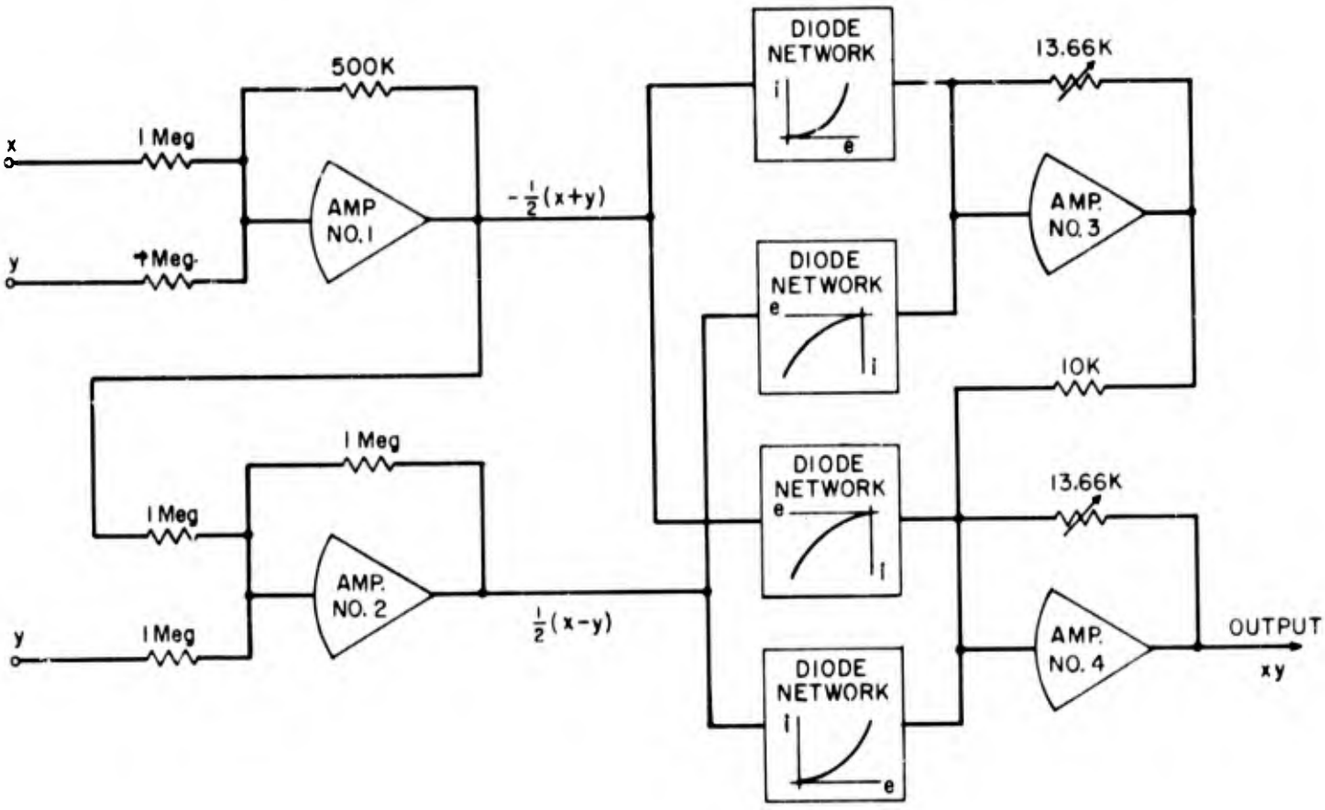
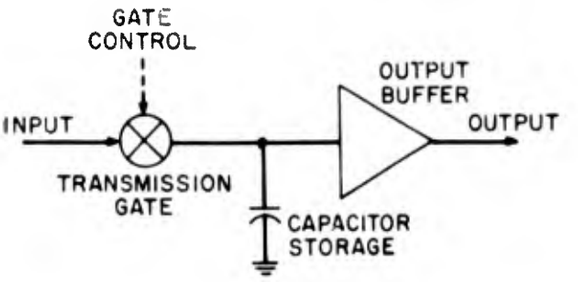
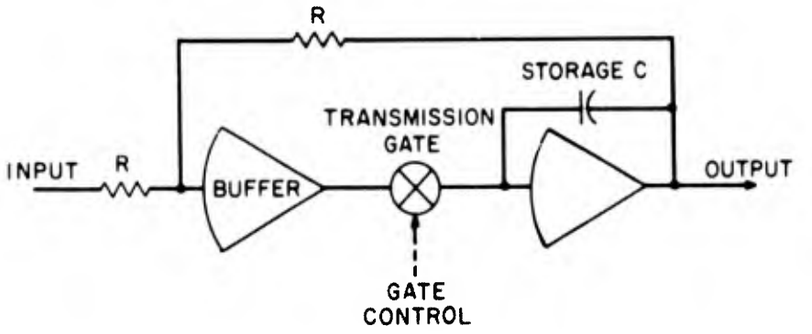


Fig. 5.17 GPS Quarter Square Multiplier



a) OPEN-LOOP SAMPLE AND HOLD GATE



b) CLOSED-LOOP SAMPLE AND HOLD GATE

Fig. 5.18 Two Standard Sample and Hold Configurations

With all these factors considered, the project selected 10 Adcom sample and hold gates for the demonstration system. This unit stores a full-scale ±10 volt input in 10 microseconds and holds the stored value indefinitely within ±0.05 percent of full scale. Permanent holding is achieved by replenishing the charge on the storage capacitor whenever the stored voltage decays outside the ±0.05 percent track. The correction signal is derived by converting small voltage changes to corresponding time changes measured with respect to a stable, high frequency oscillator. Each gate has an output current capability of ±10 ma. with an output impedance of less than 1/2 ohm. For the M.I.T. system, the standard Adcom model 420A was modified to accept a -3 volt level as the track command and a zero volt level as the hold command.

An alternative method of achieving permanent holding is to provide a separate digital register, digital-to-analog converter, and buffer amplifier for each stored value. Although improved accuracies up to 15 bits (±0.01 percent) are obtainable with this technique, the cost per channel is substantially higher than that for sample and hold gates. For example, a 12-bit (±0.05 percent) digital-to-analog converter, complete with flip-flop register, reference supply, and a Burr-Brown 1607A buffer amplifier, can be assembled with standard Digital Equipment Corporation (DEC) modules for roughly $1000/channel. The Adcom sample and hold gates, having comparable accuracy, average $745/channel. If greater accuracy is required, Scientific Data Systems (SDS) offers a packaged 15-bit (±0.01 percent) A-D converter with a digital register, power supplies, and buffer amplifier at $2575/channel. Sample and hold gates accurate to ±0.01 percent are available, but not with an indefinite holding capability.

The inherent speed and accuracy of digital-to-analog ladder networks is being further exploited in devices for multiplying a digital number by a variable analog voltage. Entire hybrid computing systems based on such devices are now offered.[41,42,43,44]

Advances in transistor technology and the introduction of new electronic configurations, such as the Diamond circuit,[45] indicate that significant improvements can be effected in sample and hold gate performance. Sampling times of tens of nanoseconds and hold times of hours have already been achieved with experimental circuits.[46,47] For most practical purposes, a hold time of even one hour constitutes an indefinite holding device. The future of such circuits depends on whether they can be made to perform accurately and reliably at a price significantly lower than the Adcom gate.

5.12 SAMPLE GATES

In pulsed-analog applications, a sample gate having the attributes of a single pole-double throw switch is desired to permit the simultaneous selection of one or more input channels to a summing amplifier, but with unselected channels effectively shorted to ground. The required functioning is indicated diagramatically in Fig. 5.19. In the M.I.T. demonstration system, a series shunt transistor gate designed by Solbakken[48] was employed, two gates being mounted on a single DEC printed circuit card. The series shunt arrangement reduces the signal feed through

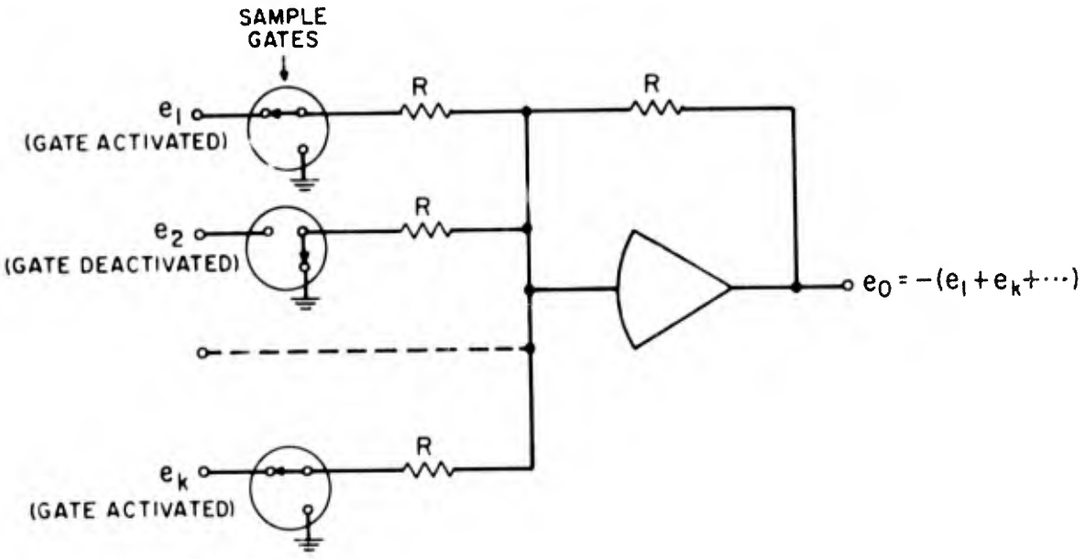


Fig. 5.19 SPDT Action of Sample Gates

in the deactivated state and eliminates variations in drift, bandwidth, output impedance, channel gains, and noise with the number of channels activated. The circuit schematic for a single sample gate is shown in Fig. 5.20.

The successful operation of this circuit requires a very low impedance source driving the gate. This is the case in the flight simulation problem where the sample gates are normally fed from operational amplifiers with output impedances well below one ohm. Since the transistor leakage current is of very little importance in this configuration, high-frequency grown junction germanium transistors with low saturation resistance and offset voltage can be used. In a simple series-shunt arrangement, the base current of transistor Q_1, would vary with different input voltages as much as 1:11. Since storage and fall time of the collector current increases almost linearly with base current, it is advantageous to stabilize the base drive in order to achieve minimum on and off time for a given transistor over the whole range of input voltages. Clamping action is obtained by means of the transistor Q_2. By using a transistor instead of a diode only, the necessary clamping current drawn from the operational amplifier is reduced by the current gain of the transistor. Q_2 is a high gain germanium transistor able to withstand 25 volts collector-base voltage. The diode in the emitter lead of Q_2 increases and stabilizes the base drive voltage for Q_1; and since it is a high-speed diode, it allows Q_2 to be slow. When the switch is closed, the error introduced is the saturation-resistance and voltage of Q_1. When the switch is open, the load resistance is connected to ground through the saturation resistance of Q_3. The offset voltage of Q_3 can be effectively trimmed out by the variable emitter bias.

In Fig. 5.21, the dynamic characteristics of the switch are shown. The output voltage is measured at the collector of transistor Q_3. The speed of response for ±10 volt input signal is about 2 μsec (within 0.1 percent of final value). The static error introduced due to a non-ideal series-transistor is maximum 3 mV. When the switch is on, a driving resistance of one ohm will, in addition, introduce a maximum 1.0 mV error.

When used in connection with an operational amplifier, the series-shunt gate, switches the summing resistor between the output of the low impedance drive amplifier and ground so that the effective resistance between summing node and ground remains the same when the gate is on and off. This eliminates changes in the response of the amplifier as a function of the number of gates enabled.

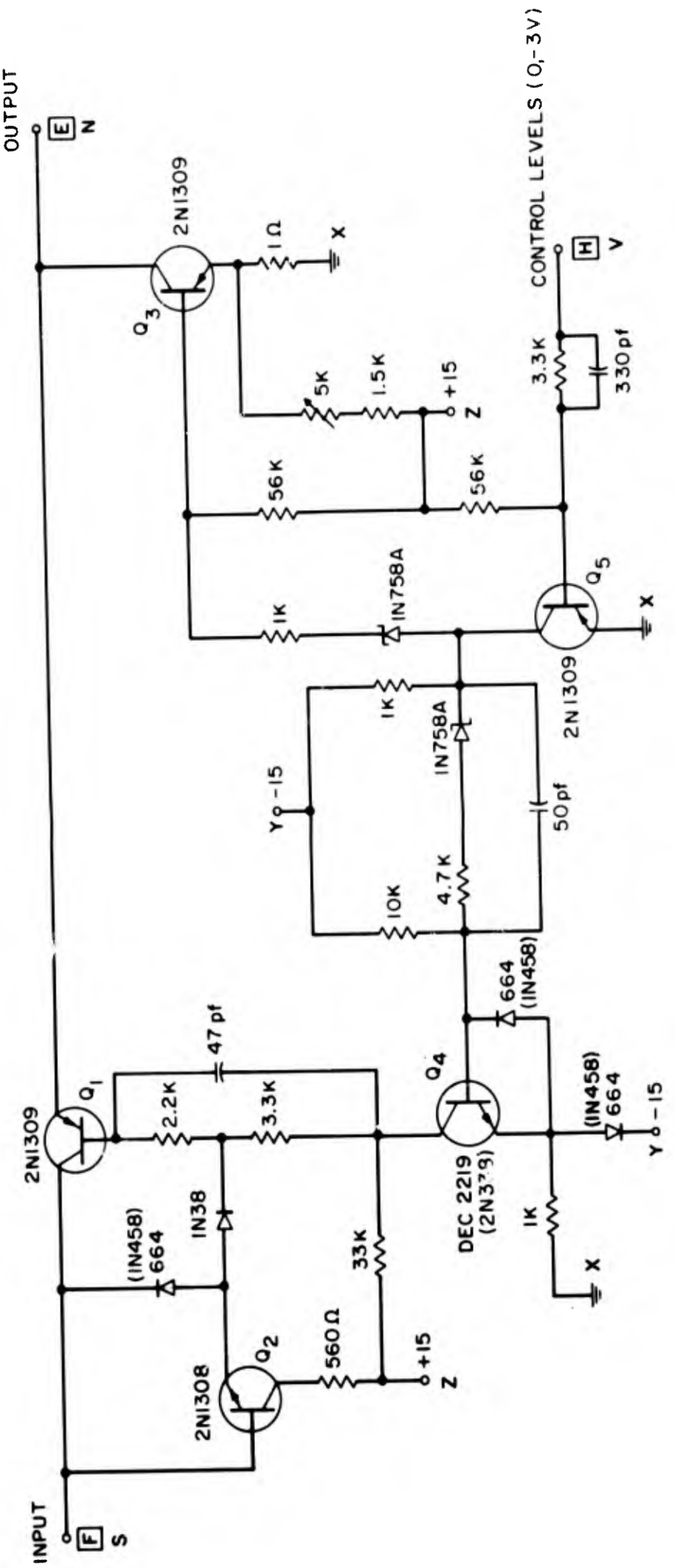
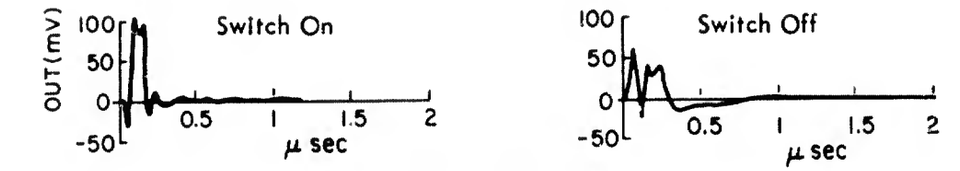
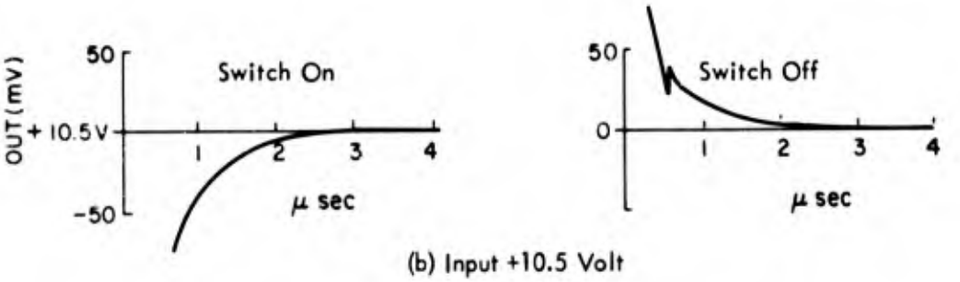


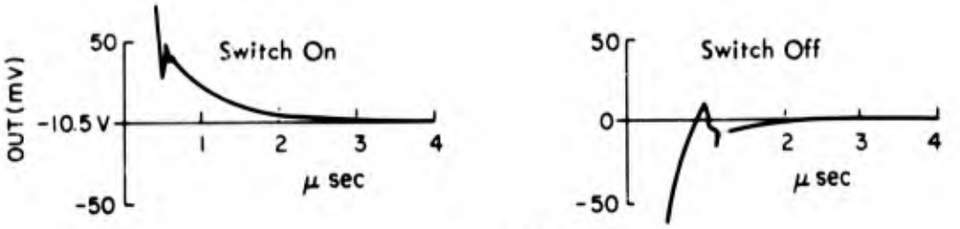
Fig. 5.20 Series - Shunt Sample Gate



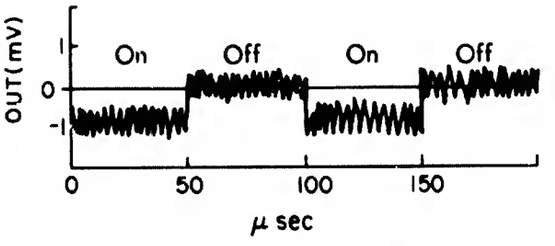
(a) Input Grounded



(b) Input +10.5 Volt



(c) Input -10.5 Volt



(d) Input Grounded

Fig. 5.21 Response of Series-Shunt Switch

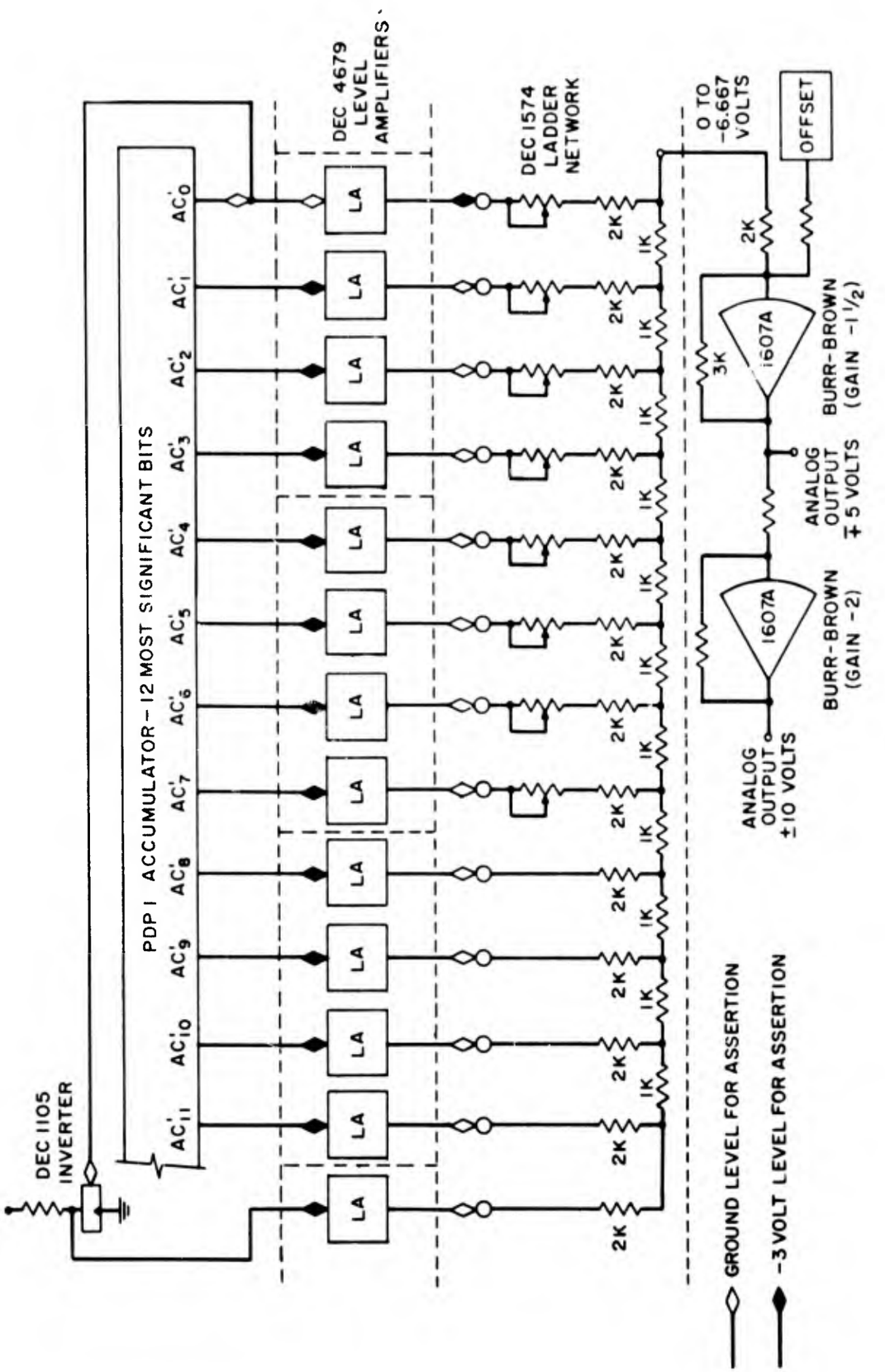


Fig. 5.22 Digital-to-Analog Conversion Circuits

When the input to a gate is grounded and control signal is applied, spikes will appear at the output as shown in Fig. 5.21. For the flight simulation problem, a sample gate and a storage gate normally operate together and the storage gate is switched to the hold state before the sample gate is turned off. Therefore, only the spike occurring when the sample gate is turned on could possibly cause trouble. But since the error due to the spike lasts only 0.5 μsec, the effect is negligible.

In fabricating the Solbakken gate on a printed circuit card, the Digital Equipment Corporation, in some cases, substituted what we thought to be equivalent circuit elements. However, when the cards were delivered, it was discovered that these changes had introduced a slight nonlinearity for the larger input signals. For this reason, scaling of signals in the pulsed analog interface was readjusted so that no sample gate would transmit signals greater than ± 5 volts.

Sampling, of course, is one of the basic operations of hybrid computation and data processing and extensive research has been carried out in this area.[4,6,20,36,37,38,45,46,47,49,50,51] Most commercial emphasis has been placed on the special case of multiplexing, that is, the selection of one channel out of many. For this application, accuracies of 0.01 percent with settling times less than 15 microseconds are obtainable at roughly $150 per channel. Submicrosecond switching at lower accuracies has also been achieved.

5.13 DIGITAL-TO-ANALOG CONVERTER

In the demonstration system, as stated previously, the contents of the twelve most significant bits of the PDP-1 accumulator control a digital-to-analog converter directly, that is, any binary number placed in these bit positions is converted within 2.5 microseconds to an equivalent analog voltage without extra programming steps being required. To effect the conversion, a resistive divider network is connected to the accumulator flip flops as shown in Fig. 5.22, the network being weighted so that each bit contributes to the output voltage in proportion to its value.[52,53] Since flip-flop voltage levels are not usually very precise, level amplifiers are placed between the flip-flops and the divider network. These level amplifiers effectively switch the divider network inputs between ground and a stable -10 volt source provided by the system's precision reference supply. A simplified schematic of the level amplifier is shown in Fig. 5.23. Four such amplifiers are mounted on each of the four DEC 4679 cards employed. The resistive divider is the DEC 1574 ladder network card.[54]

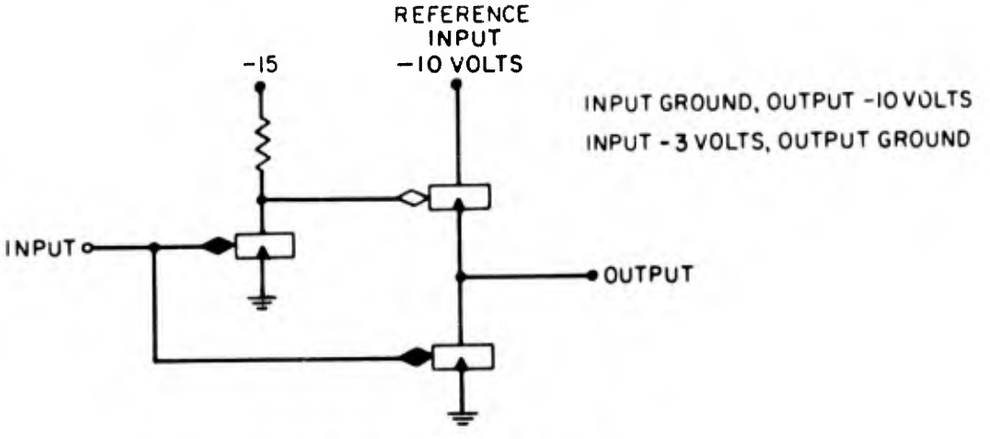


Fig. 5.23 DEC 4679 Level Amplifier

The PDP-1 operates on the one's complement number system, that is, negative numbers are represented as the one's complement of the corresponding positive numbers. Bit 0 is the sign bit which is zero for positive numbers and one for negative numbers. Thus, the most positive number that can be represented by the most significant 12 bits of the accumulator (sign included) would be

$$011\ \ 111\ \ 111\ \ 111 = 2^{11} - 1$$

Zero has two forms

$$\begin{cases} 000\ \ 000\ \ 000\ \ 000 = +\text{zero} \\ 111\ \ 111\ \ 111\ \ 111 = -\text{zero} \end{cases}$$

The most negative number expressible with the same 12 bits would be the one's complement

$$100\ \ 000\ \ 000\ \ 000 = -2^{11} + 1$$

For accumulator bits AC'_1 through AC'_{11}, a bit value of one causes a -3 volt input to the corresponding level amplifier, but in the case of the accumulator sign bit AC'_0, a bit value of one causes a ground input to the level amplifier. This inversion is necessary to make the range of numbers above correspond in a continuous fashion to the ladder's output voltage range. The correction signal derived from AC'_0 and applied to the last level amplifier of Fig. 5.22 brings the analog voltages corresponding to ±zero to within half of the least significant bit of each other.

5.14 ANALOG-TO-DIGITAL CONVERTER

Probably no area of hybrid computation has received the benefit of as much development effort as the basic function of analog-to-digital conversion.[52,54,55,56] As a consequence, a broad selection of commercial encoders are available today with encode times as short as one microsecond at relatively low resolutions and resolutions to ±14 bits at slower conversion speeds.

The solid-state Beckman 4040C converter purchased for the demonstration system employs a standard successive approximation technique (bit-by-bit trials). It is a compromise design representing neither the ultimate in speed or in accuracy, having a resolution of ±12 bits and an encode time of 30 microseconds requiring two microseconds per bit trial. Actually, a temporary unit on loan from Beckman was used in the demonstration tests and this unit had a conversion time of 50 microseconds. It will be replaced by the faster converter when the latter becomes available. The binary output, which is available either in serial or parallel form, utilizes the one's complement number system. Since DEC logic is employed in transferring encoder outputs to the PDP-1 in-out register, the output levels have been changed from 0, -12 volts to 0, -3 volts. The input voltage is bipolar (±10 volts) and the input impedance is 13.33 K. The encoder has its own internal precision reference supply. Five operating modes are possible; external clock-external start, external clock-automatic start, internal clock-automatic start, internal clock-manual start, and internal clock-external start. The latter mode is employed in the demonstration system, a transition from a -3 volt level to ground at the external asynchronous start input initiating the conversion process. When a conversion is completed, the encoder puts out a level change on a separate output line.

5.15 DIGITAL LOGIC

Because the DEC PDP-1 digital computer was the central element in the hybrid demonstration system, all interface and control logic employed fully-compatible Digital Equipment Corporation modules mostly chosen from the 500 kilocycle 4000 series. The characteristics of these logical elements are explained in detail in the DEC module handbook,[57] but to assist the reader in understanding the logic block diagrams presented in this report, a brief description will be given here of the principal units incorporated in the demonstration system.

DEC logic is based on the use of static flip-flops, logical operations with levels, and pulse sampling. The logical voltage levels are -3 volts and ground. The correspondence between the logic states one or zero and the voltage levels of -3 and ground is indicated at each point in a logic diagram by a diamond symbol. A solid diamond ◆ denotes a -3 volt level for assertion and a hollow diamond ◇ denotes a ground level for assertion. Similarly, a solid arrow —▶ denotes a negative pulse and an open arrow —▷ a positive pulse. In the logic diagrams of this report, an open circle —○ is often used to denote ground and the emitter designation is usually omitted from the inverter symbol ▭▲▭.

Most logical operations are performed with saturating PNP transistor inverters as shown in Fig. 5.24a. When a negative (-3 volt) level is applied to the base input, the transistor becomes saturated and the output is essentially shorted to ground. When the base input is positive or at ground level, the transistor is open-circuited and the output, because of the clamped load resistor, is at -3 volts. The emitter terminal can be used as a second logical input in the manner shown. Complex Boolean operations are implemented by a cascade of inverter gates in series-parallel combinations. Logical levels can be sampled and read into a flip-flop by applying the level to the emitter of an inverter and the sampling pulse to the base, using the pulse output at the collector to set the flip-flop.

DEC flip flops are set to a desired state by momentarily shorting the corresponding input line to ground through inverters. When the input on the zero side has been shorted to ground, the zero output is set to the -3 volt level and the one output is at ground. This is, by definition, the zero state. Conversely, when the flip-flop is in the one state, the one output is at -3 volts and the zero output is at ground. There is a built-in delay between the time a flip-flop is pulsed and the resultant change in its output. As a consequence, flip-flop states can be sensed at the same time they are pulsed. Some flip-flops, such as the 4209, have inputs that may be treated either as gated or as direct inputs. In the first instance, the input signal must come from the collector of one or more inverter pulse gates. In the second instance, the input is driven directly by a DEC standard 0.4 microsecond positive pulse. Some flip-flops also have a complement input which changes the state of the flip-flop regardless of its present state.

Diodes may be added to the inverter input to form a logical AND or a logical OR gate as shown in Fig. 5.24b. When the diodes are in the reverse direction, the circuit operates as an AND gate for negative inputs and an OR gate for ground inputs.

DEC pulse amplifiers are used to amplify and standardize pulses, and also for gating pulses. The DEC symbol is shown in Fig. 5.24c. Since the outputs are from pulse transformers with both terminals available, either positive or negative pulses (2.5 volts amplitude, 0.4 microseconds duration) may be obtained, depending on which terminal is grounded. As noted above, positive pulses are needed to direct, set, and clear flip-flops without inverters, and negative pulses are needed for setting, clearing, and complementing flip-flops with inverters. The input to the pulse amplifier proper must come from the collector of one or more inverter pulse gates.

DEC delay units are monostable flip-flops. When the input terminal is shorted to ground by a standard pulse, the level output terminal changes from its normal ground level to -3 volts for a fixed period of time which is adjustable. In addition, a standard pulse is produced at the pulse output terminal when the level output returns to zero volts at the end of the delay period.

A binary-to-octal decoder converts three bits of binary level information into eight lines of octal information, with the selected output line at a negative voltage and the other seven output lines at ground. The DEC symbol is shown in Fig. 5.24c. As an illustration, for a binary input of 000, the input lines L, J and F are at ground and the output R is the selected output line (-3 volts).

For low-speed (500 kc) logical applications, DEC also offers positive and negative capacitor-diode gates, the latter unit being used rather extensively in the demonstration system. The circuits for the basic gates are shown in Fig. 5.24d. In both configurations, the capacitor and resistor inputs must be held at a constant voltage level for a few time constants prior to pulsing for the capacitor to become fully charged. In the positive capacitor-diode gate, the level input must be grounded if a pulse is to be passed through the gate. Moreover, the pulse input must be negative (or a negative-going level change) in order to momentarily forward bias the diode and produce a positive output pulse. In the negative capacitor-diode gate, the level input must be -3 volts if a pulse is to be passed through the gate. The pulse input must be a standard negative pulse (or negative-going level change) in order to forward bias the diode momentarily and produce a positive-going pulse (relative to -3 volts) at the output. Since the pulse inverter is on the output side, the negative capacitor-diode gate has a much greater driving capability. The positive gates are used for driving unbuffered flip-flops only; the negative gate will drive any 4000 series buffered flip-flop, pulse amplifier, or delay unit.

DEC cites the following advantages for the capacitor-diode gate over a simpler inverter:

1. The level gate input presents no DC load.

2. The RC time constant of the gate requires that the conditioning level be present a certain amount of time before the pulse occurs. This introduces a delay between the application of a new gate level and the time the gate is conditioned, and allows the sampling of unbuffered flip-flops at the same time the flip-flop is being changed.

3. The capacitor-resistor combination differentiates a level change. This is useful when it is desired to use a level change to trigger a unit which normally operates from a pulse.

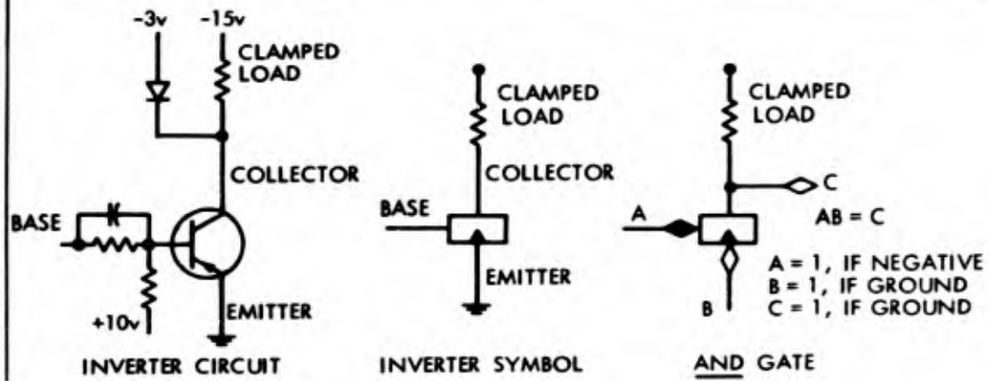
The type 4129 card employed in the demonstration system contains two circuits, each with four negative capacitor-diode gates having an OR input into a single pulse inverter. The symbol for one of these circuits is also shown in Fig. 5.24d.

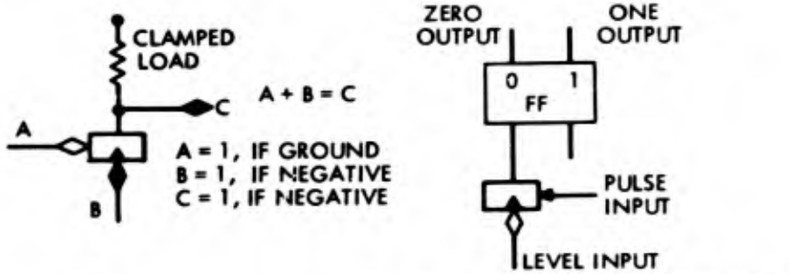
5.16 POWER SUPPLIES

One of the hazards of systems design is the proliferation of power supplies necessary to satisfy the assorted requirements of voltage level, regulation, output capacity, and noise decoupling. Because a reasonable effort was made to standardize on a single set of convenient supply voltages, it is felt that the demonstration system represents a relatively efficient implementation in this regard.

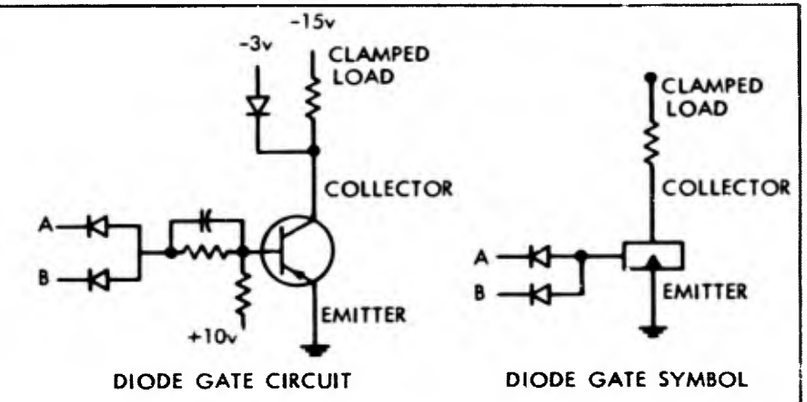
Both the Burr-Brown 1607A and the Nexus operational amplifiers require ±15 volt supplies for which two NJE RB-18-3 MP solid-state units were employed. This power supply has an output range 0 to 18 volts, adjustable with a resolution of 10 millivolts, and an output current capability of 3 amps. Load and line regulation are both 0.01 percent or one millivolt, whichever is greater. The crossover point between constant voltage and constant current operation can be

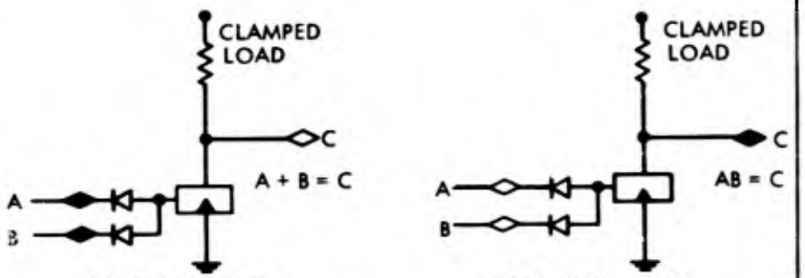
A



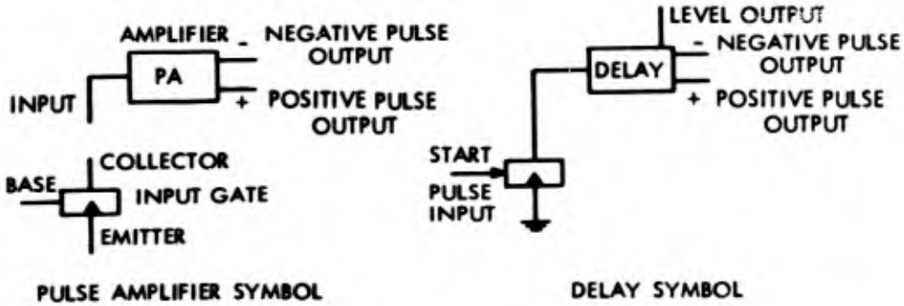


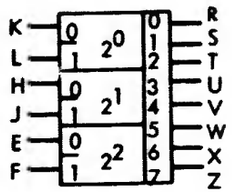
(a) INVERTER LOGIC



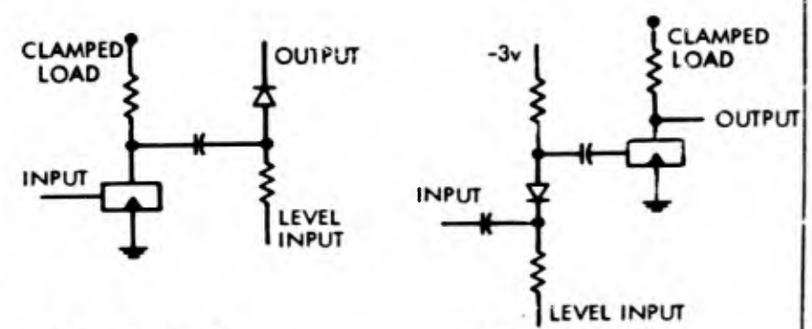


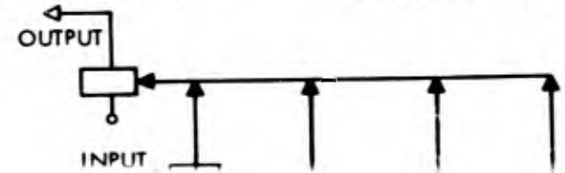
(b) DIODE LOGIC

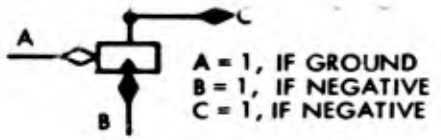




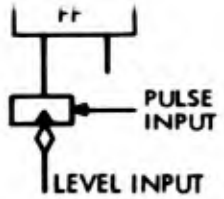
BINARY-TO-OCTAL DECODER SYMBOL





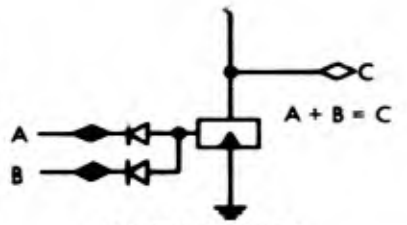


OR GATE

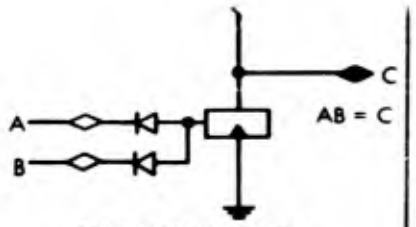


LEVEL SAMPLING INTO FLIP-FLOP

(a) INVERTER LOGIC

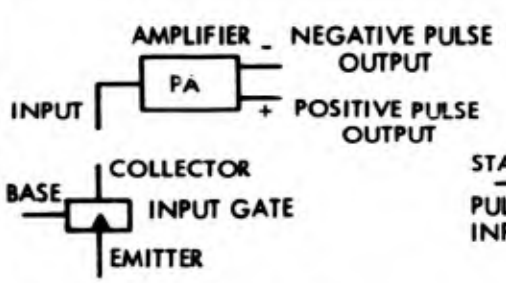


OR DIODE GATE

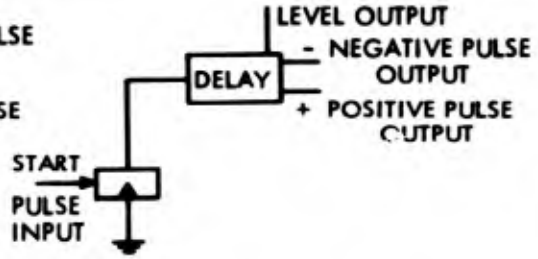


AND DIODE GATE

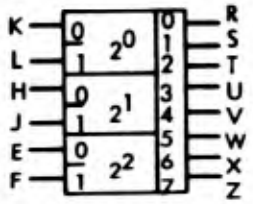
(b) DIODE LOGIC



PULSE AMPLIFIER SYMBOL

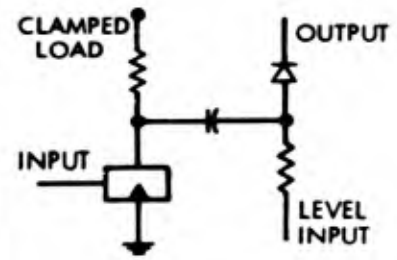


DELAY SYMBOL

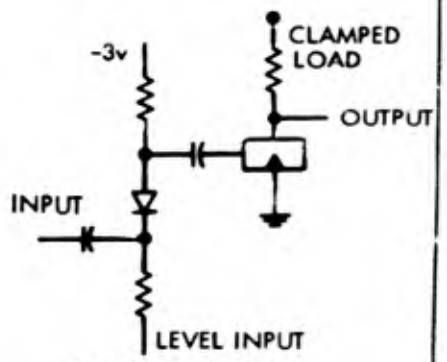


BINARY-TO-OCTAL DECODER SYMBOL

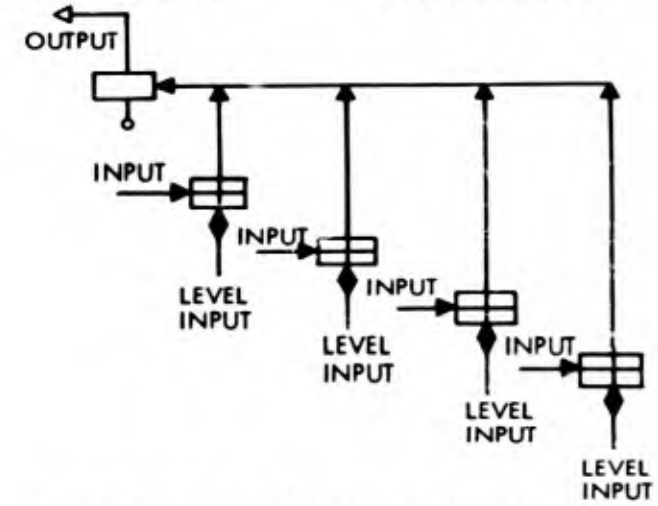
(c) MISCELLANEOUS LOGIC



BASIC POSITIVE CAPACITOR-DIODE GATE



BASIC NEGATIVE CAPACITOR-DIODE GATE



SYMBOL FOR 4129 NEGATIVE CAPACITOR-DIODE GATE

(d) CAPACITOR DIODE GATE

B

Fig. 5.24 DEC Digital Logic

preset at any point within the rated voltage and current range. A second pair of these supplies provides the ±10 volts used as machine one references in the analog system.

All digital logic cards are provided with +10 volts and -15 volts from a standard DEC 722 power supply having rather rough regulation, but good switching noise rejection. Up to 7 amperes can be supplied by the +10 volt output and up to 6.5 amperes by the -15 volt output. However, the capacity of these two outputs is interrelated, the constraining relation being:

$$5 I_{10} + 6 I_{15} = 44$$

The Adcom sample and hold gates and the Beckman analog-to-digital encoder have internal power supplies. The GPS quarter square multiplier, however, must be provided with ±300 volts. A Philbrick R-100 B unit satisfies this requirement.

BLANK PAGE

CHAPTER VI

EXPERIMENTAL AND DEMONSTRATION TEST RESULTS

6.1 INTRODUCTION

The successful simulation of a complex aircraft model using custom-built computing equipment of novel design requires meticulous attention to the details of the model, the data, the program, and the equipment. For this reason, each stage in the completion of the M.I.T. hybrid demonstration computer has involved a parallel test phase of some magnitude.

In the development of the analog equipment, for example, testing began on individual components, such as the metal-film resistors and the photoresistive cells. These elements had to be measured and matched to achieve required accuracies. Subsequently, each analog card had to be tested individually, each sub-group of analog cards (Altitude, Lift, Pitch, etc.) was checked, and finally, the entire analog system was tested as a skeleton airplane. Similarly, individual elements of the pulsed-analog interface such as the analog-digital conversion units, the GPS Multiplier, the Burr-Brown amplifiers, the sample gates, and the sample and hold gates were tested statically and dynamically. Under PDP-1 program control, groups of pulsed-analog elements were then checked by circulating analog voltages around a closed loop, a procedure which magnifies small offsets and gain errors. The discrete in-out and opg logic was also verified on a bit-by-bit basis.

Prior to the preparation of programs, aircraft and engine data of mixed authenticity was received from North American Aviation, Melpar, Sylvania, Pratt and Whitney, and the U. S. Air Force; hence, ample opportunities existed here for misinterpretation and error. To guard against this, each aircraft parameter was spot-checked on a desk calculator and cross-checked with other sources to assure that reasonable values were being used. With respect to the programming itself, building-block routines, MACRO definitions, and subroutines were first written and checked. Following this, all-digital programs for the engine and the aerodynamic equations were written and tested separately. These were then combined into the final all-digital simulation program, the steady-state and transient performance of which was compared with available F-100A data and the UDOFT results. An abbreviated set of these tests was repeated with the hybrid system, noting the effect of solution rate.

Underlying these more obvious requirements for component, program, and system tests was a subsidiary need for establishing precise measurement standards and procedures. In addition, utility programs had to be written to run an x-y plotter under PDP-1 control, to plot graphs and tabulate data on the CRT display, to establish histograms of the analog-digital conversion dispersion, to convert data conveniently betwixt the binary, octal, and decimal number systems, and to print out selected information on demand.

Much of the detailed labor referred to above is of limited general usefulness, hence will not be described in detail in the present report. Nonetheless, this labor represents a massive exercise in low prudence which must be carried out if a successful simulation is to be achieved. Prophets of push-button problem solving and painless programming should be made aware of these seemingly chronic and unavoidable burdens before disillusionment sets in.

6.2 ALL-DIGITAL TESTS

As stated previously, the objectives of the all-digital simulation were:

(a) To check the validity of the F-100A model and data.

(b) To provide a reference against which the hybrid simulation could be compared.

(c) To provide a core program which could be modified to serve in the hybrid simulation.

The results of the steady-state tests on the engine portion of the simulation were compared with Pratt and Whitney data on the J-57 (JT-3) engine.[61,62] The comparison demonstrated in these plots is actually a measure of the quality of Melpar's interpretation of the original Pratt and Whitney engine data. In general, agreement is fair-to-good, though, in some cases, the errors are considerable. A selection of typical engine simulation test results is presented for an altitude of 15,000 feet in Figs. 6.1 through 6.4. Corresponding curves were also obtained at sea level, 25,000 ft, 35,000 ft, 45,000 ft, and 55,000 ft.

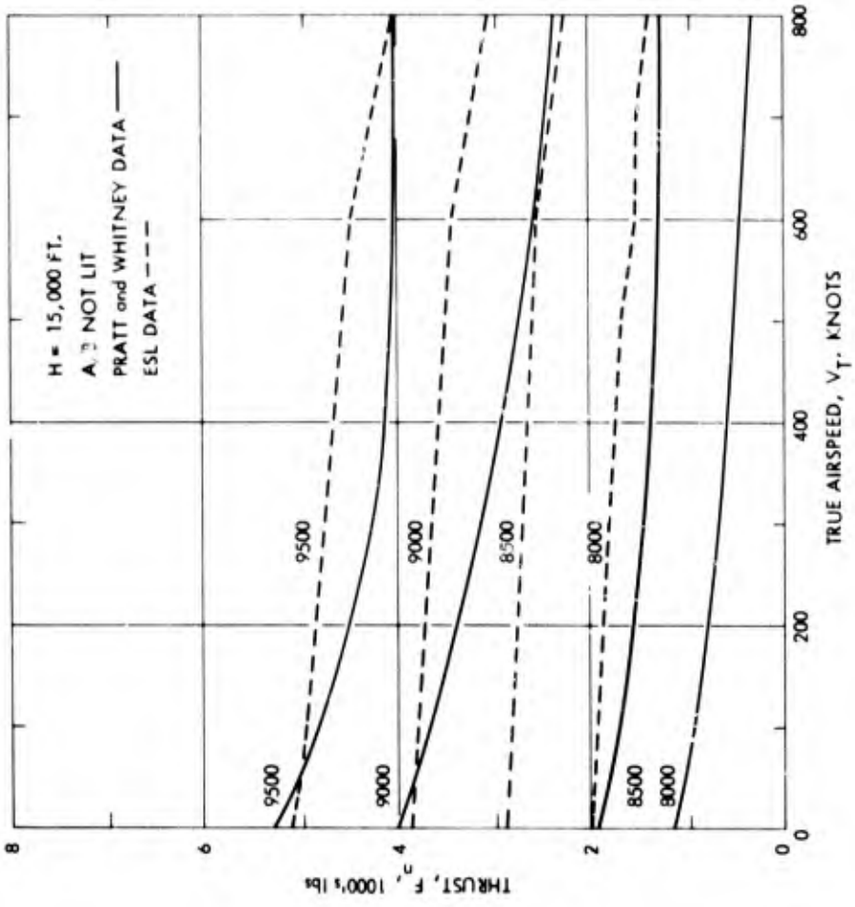


Fig. 6.2 Thrust vs. Airspeed and RPM

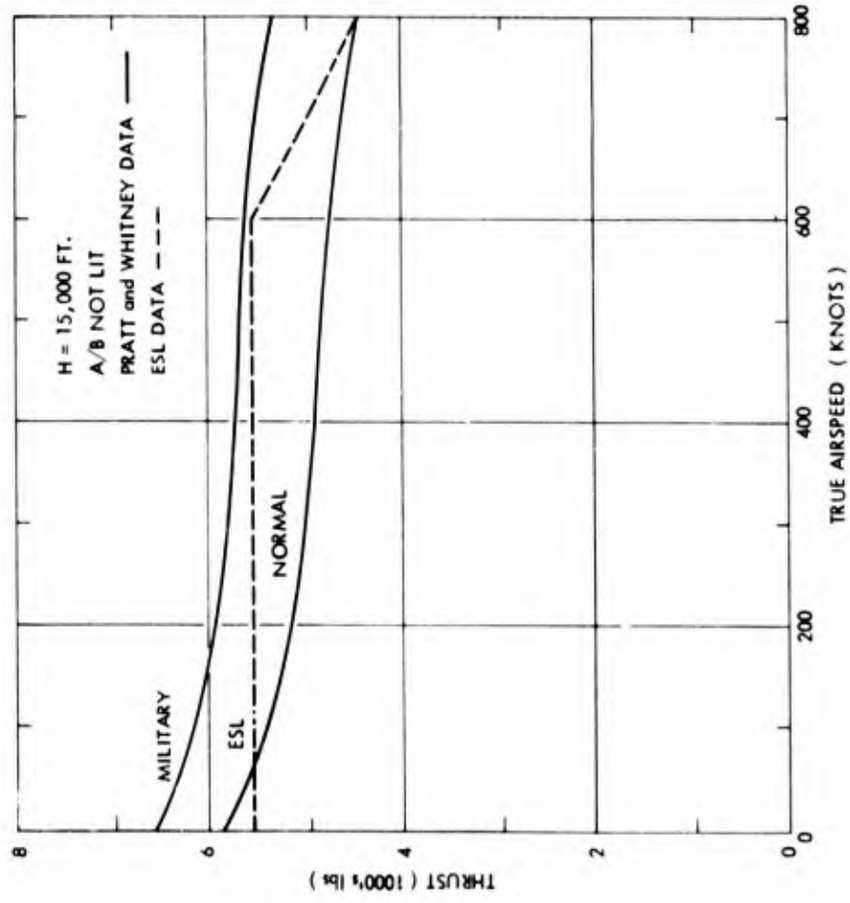


Fig. 6.1 Maximum Thrust vs. Airspeed

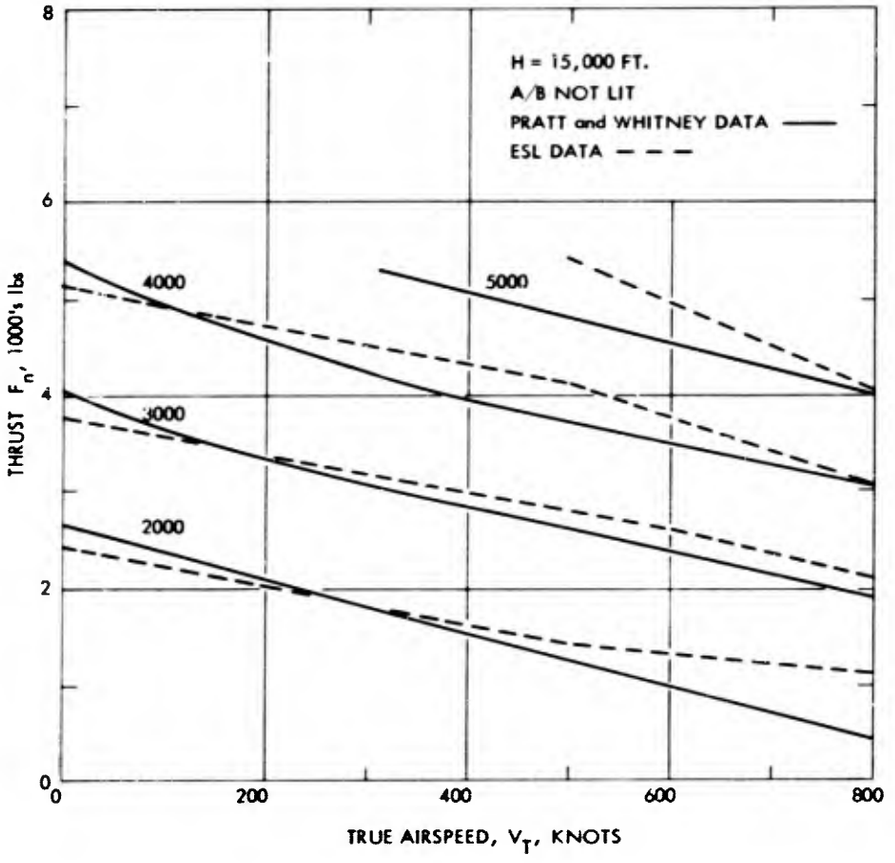


Fig. 6.3 Thrust vs. Airspeed and Fuel Flow

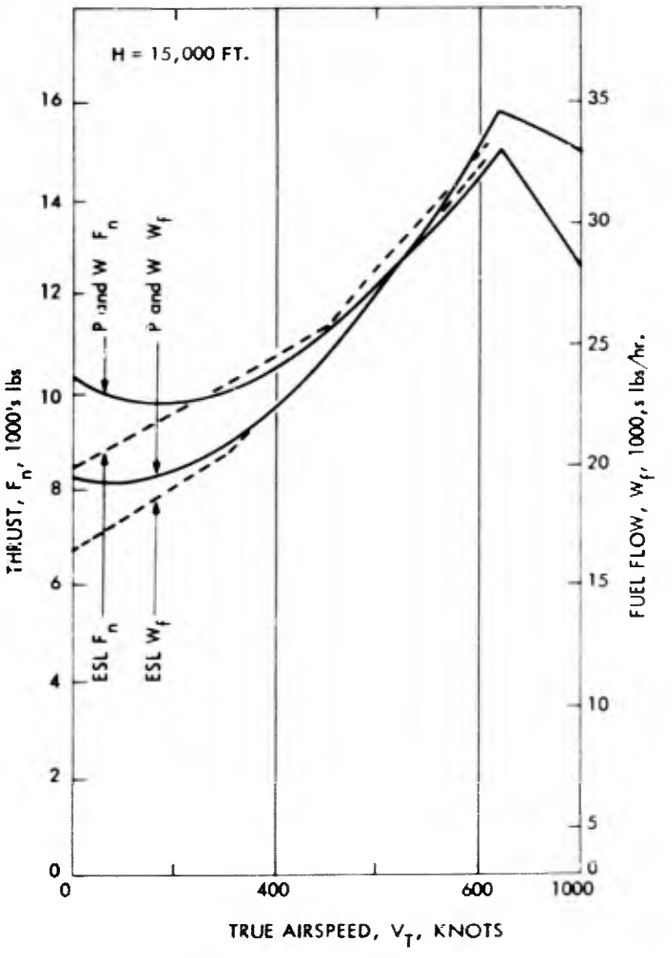


Fig. 6.4 Maximum A/B Thrust and Fuel Flow vs. Airspeed

The all-digital steady-state tests performed in the F-100A airframe simulation were directed toward plotting the thrust-airspeed relationship for the same range of altitudes and for a given aircraft mass. A typical curve is shown in Fig. 6.5, again for the 15,000 ft condition. In addition, this plot demonstrates the influence of different values of Mach breakpoint in the region of Mach 1.0 on the speed-thrust profile. Such discrepancies could be eliminated by using more breakpoints near Mach 1.0, but this increases the memory requirement. The corresponding UDOFT speed-thrust profile is super-imposed on Fig. 6.5.

Some difficulty was encountered in obtaining these plots. Even when the initial conditions of the computer model were set reasonably close to the final steady-state, a prolonged transient occurred before equilibrium was reached. To avoid wasting time in obtaining the thrust-airspeed data, a special temporary routine was coded which would force the model to a steady-state in just 10 percent of the time required earlier. To find the thrust required for straight and level flight at a given altitude and airspeed, the special routine adjusted thrust until the X acceleration was zero, adjusted the angle of attack (α) until the Z acceleration was zero, and adjusted the horizontal stabilizer position until the pitch acceleration ($\dot{q}$) was zero. During this processing, all rates which were inputs to integrations were forced to zero. Once the steady-state had been reached, the integrations could be reinstated with perfect freedom and the aircraft model would remain in the previously forced steady-state.

The all-digital dynamic tests were run using the techniques developed in the earlier Whirlwind studies, namely, stimulating the model with a standard (well-defined) input and recording the response.[18] The set of standard stimuli used with the present all-digital simulation program are trapezoidal control surface deflections of one second duration having the following amplitudes (in radians):

	Rudder δR	Aileron δA	Stabilizer δH
Mild	0.05	0.01	--
Moderate	0.25	0.10	0.10

The rise and fall rate for all inputs is one radian per second. The responses recorded for δH stimuli were angle of attack (α) and pitch rate (q). The responses recorded for δA and δR stimuli were sideslip angle (β) and roll angle (ϕ). Thus for a given airspeed and altitude, 10 plots were taken. Two plots describe the longitudinal mode dynamic characteristics and eight describe the lateral mode dynamic characteristics. Such sets were obtained at Mach 0.4, 0.7 and 1.0+ and at altitudes of sea level, 15,000 ft, 24,000 ft, and 35,000 ft. Typical results are shown in Figs. 6.6 through 6.9. Unfortunately, the method of exciting transients and the variables recorded in the UDOFT tests were different, so a direct comparison of dynamic responses is not possible. The frequencies and damping characteristics, however, generally match the UDOFT results rather well. At very low velocities (Mach 0.4), the δR and δH stimuli excited lightly-damped and even underdamped lateral oscillations. However, the fact that the yaw damper was inoperative probably caused this effect. No comparable results could be found in outside sources at this low airspeed; hence, no check was possible.[63,64]

While the transient tests were in progress, the effect of varying solution rate was measured over the range of altitudes and stimuli. The results at 15,000 feet for a longitudinal and lateral transient are given in Figs. 6.10 and 6.11. These plots indicate that even with the simple trapezoidal integration formula, reasonably faithful transients are produced at solution rates as low as eight per second. More extensive testing would be necessary to verify the adequacy of such a low rate; but these results do raise the possibility that digital aircraft simulators are now artificially overburdened when operated at 20 solutions per second.

6.3 HYBRID TESTS

As discussed in the programming chapter, the hybrid program is in many essential features identical to the all-digital program. Function generation and the calculation of the engine equations, for example, are still carried out digitally. As a consequence, the engine performance in the hybrid simulation is exactly the same as that obtained in the digital simulation. Of course, the digital computer no longer computes the relationships transferred to the analog skeleton, among which are the basic force, moment, and Euler angle integrations. However, the digital program must now direct the calculation of running sums of products in the pulsed analog equipment by activating gates in the proper sequence. In addition, the transition between ground and air states in landing and takeoff is no longer handled by simple branch operations in the digital program. Relays must be activated setting up the main wheel and nosewheel effects in the analog domain.

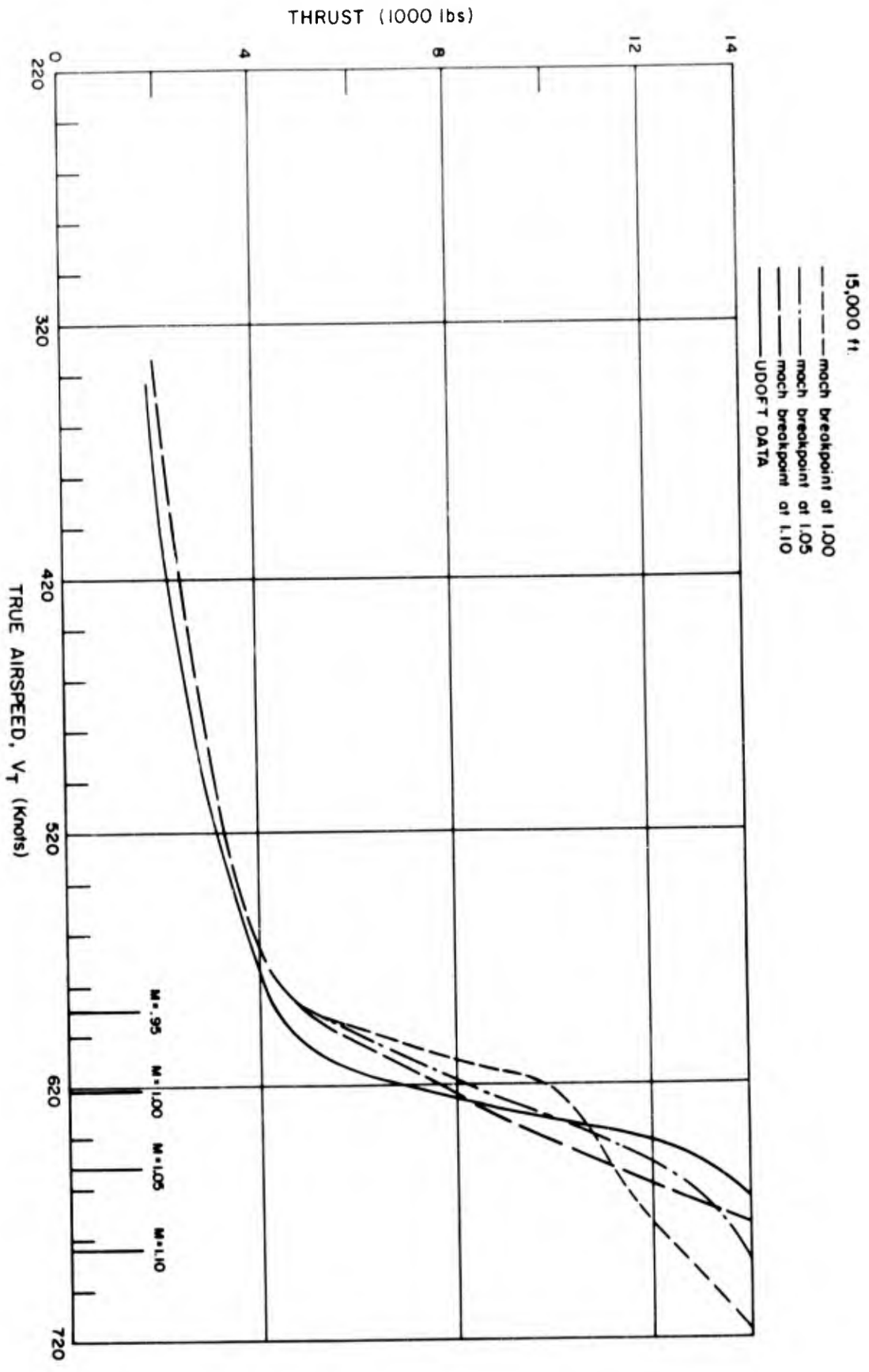


Fig. 6.5 Effect of Mach Breakpoint on Speed-Thrust Profile

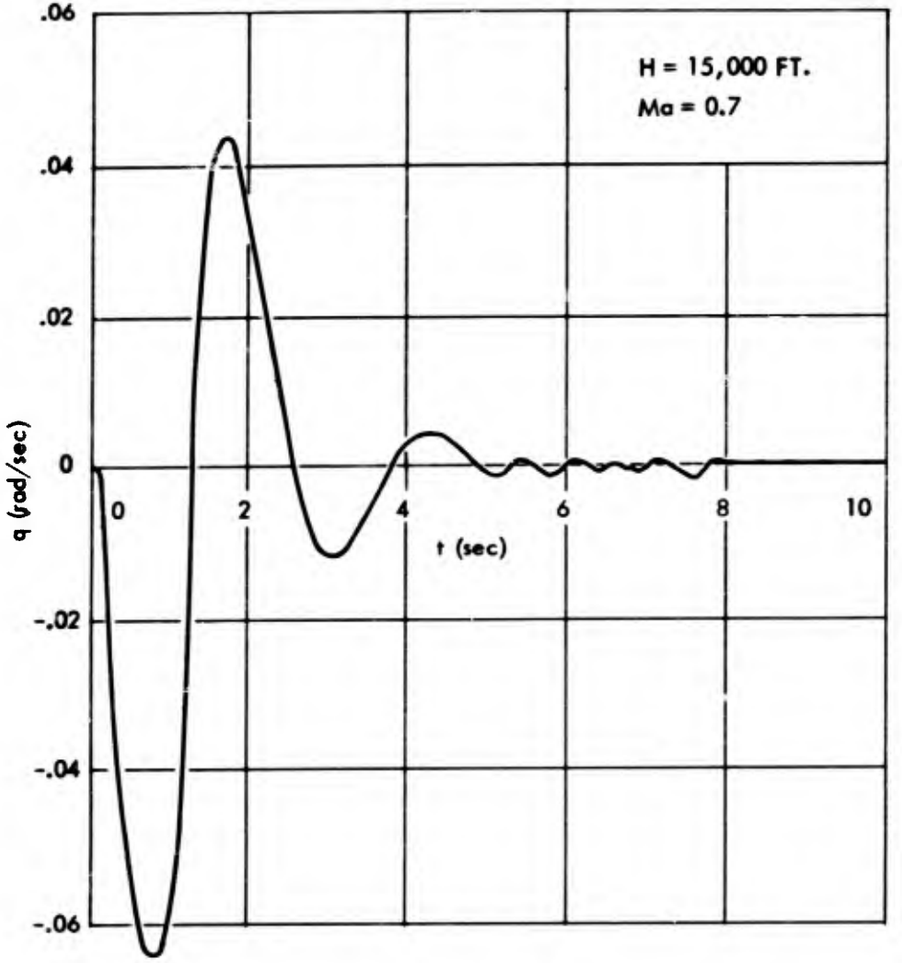


Fig. 6.6 Pitch Rate Response to δH Pulse

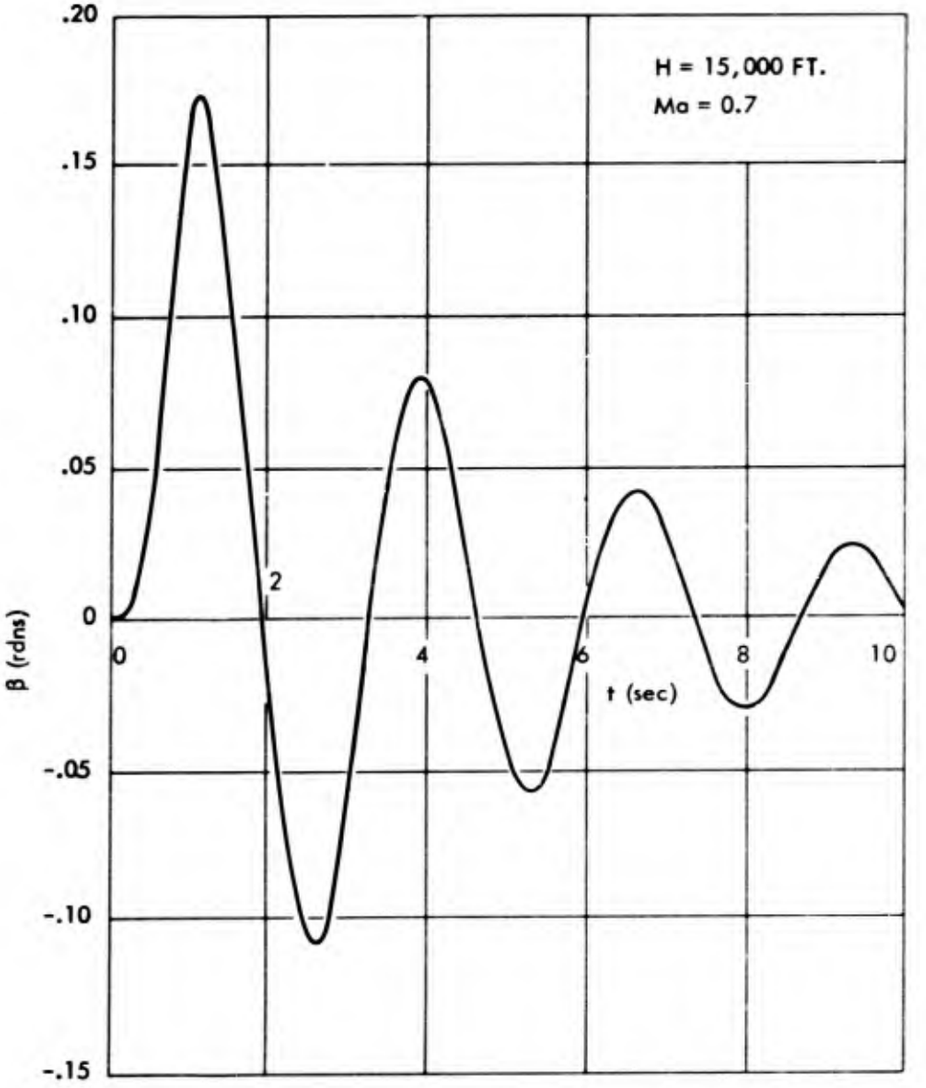


Fig. 6.7 Sideslip Response to Moderate δR Pulse

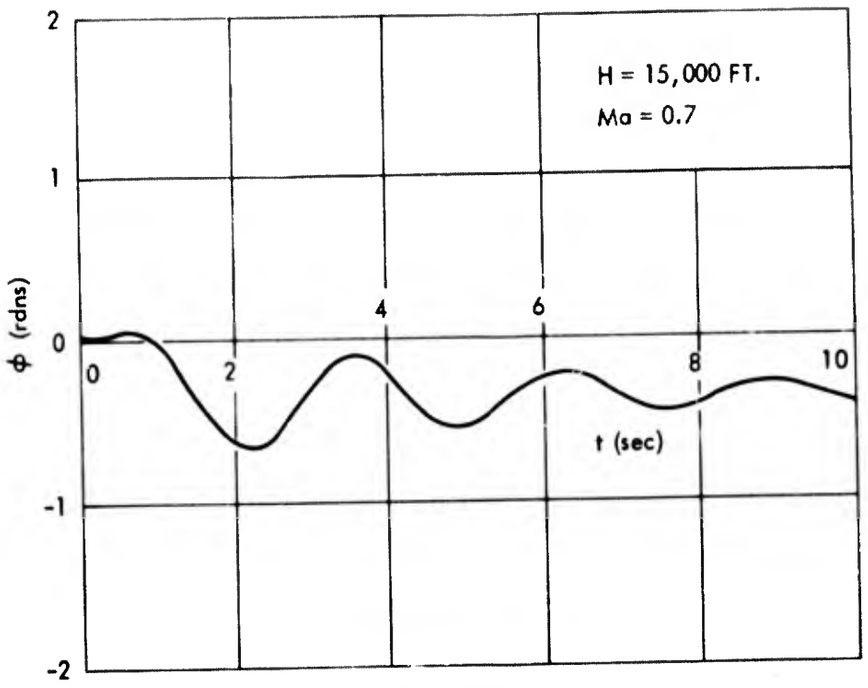


Fig. 6.8 Roll Response to Moderate δR Pulse

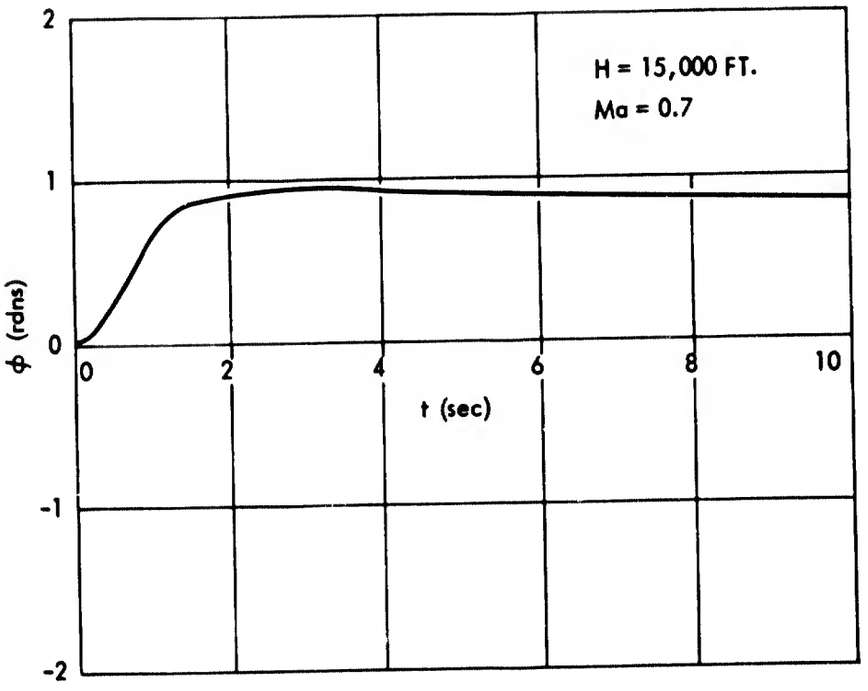


Fig. 6.9 Roll Response to Moderate δA Pulse

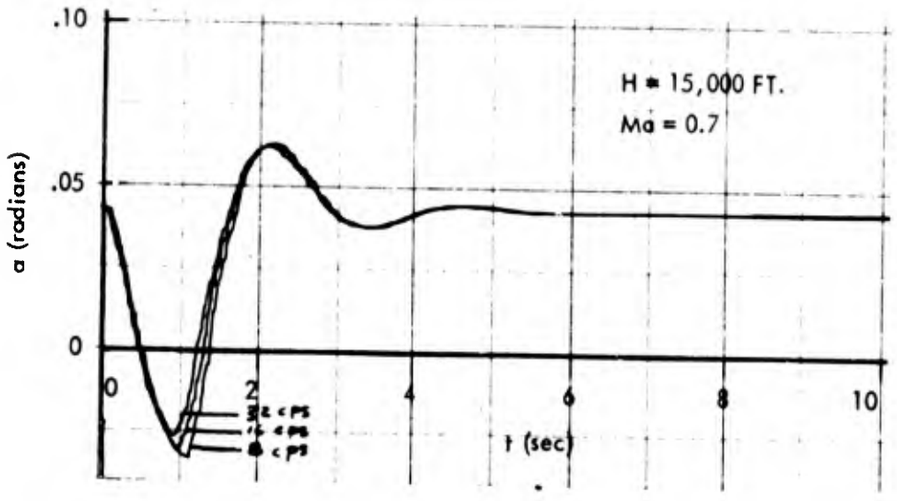


Fig. 6.10 Effect of Solution Rate on δH Pulse Transient

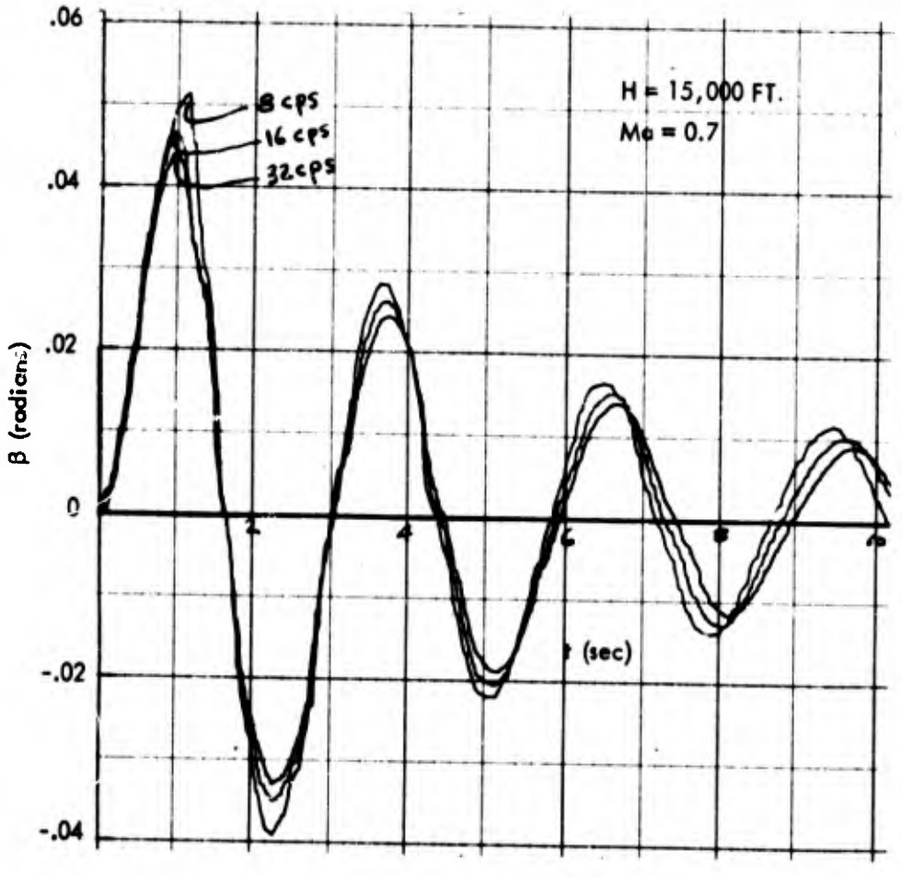


Fig. 6.11 Effect of Solution Rate on δR Pulse Transient

It has long been intuitively obvious that many of the variables in aircraft simulation, such as Mach number, dynamic pressure, and the engine parameters, change relatively slowly; hence, these variables need not be calculated as frequently as the rapidly-changing quantities such as angle of attack. This fact was not taken advantage of in the digital simulation, where all subdivisions of the program were solved at the same solution rate. However, in the hybrid program all the more frisky variables were grouped together in the Aero Part II routine and this routine was traversed twice for every traversal of the remainder of the program.

When the hybrid system was first assembled as a working unit and the scaling and sign errors and other bloopers had been purged from the program, the initial performance test results were not in good agreement with the all-digital results. At that time, the X-force summation (Thrust-Drag) was being carried out in the analog domain and, as discussed previously, the scaling was such that under normal cruise conditions this critical calculation had to be performed at low signal levels (tens of millivolts). Transferring the thrust-drag-gravity summation to the digital computer eliminated this problem.

A typical result with the improved hybrid configuration is the speed-thrust profile shown in Fig. 6.12 on which is superimposed the corresponding all-digital plot. This data was obtained at a basic solution rate of 20 solutions per second. The rate was lowered in gradual steps to one solution per second, at which frequency the Aero Part II routine was being solved twice each second. No changes in the steady-state performance or transient characteristics were discernible. In theory, of course, once steady-state flight conditions are established, it is possible to halt the digital computer altogether and the analog skeleton will continue to fly on undisturbed. At low solution rates, however, secondary limitations become significant; for example, an unrealistic delay between an action by the pilot and the observable consequences of this action. Subject to this lower bound, the hybrid configuration suggested in this report successfully reduces the computing load on the digital computer by a direct assumption of part of this load, and indirectly, by substantially lowering the solution rate requirement placed on the digital computer.

Fig. 6-12 Comparison of Digital and Hybrid Speed-Thrust Profiles

REFERENCES

1. E.W. Pughe - Logical Design of a Real-Time Analog-Digital Simulator; M.S. Thesis; M.I.T. Electrical Engineering Dept.; August, 1957.

2. Kennedy, Moroney, and Morse - An Experimental Analog-Digital Flight Simulator; Final Demonstration Report 45-2; Project 7591; M.I.T. Servomechanisms Laboratory; January, 1959.

3. Binsack, Brean, Connelly, Dumas, Gocht, and Lee - The Pulsed-Analog Demonstration Computer; Final Demonstration Report 45-1; Project 7591; M.I.T. Servomechanisms Laboratory; April, 1959.

4. J. Binsack - A Pulsed Analog and Digital Computer for Function Generation; M.I.T. Electronic Systems Laboratory; Report 8494-R-2; October, 1960.

5. G. Bell - The Pulsed-Analog System for Evaluating Correlation Functions for Radar; M.I.T. Electronic Systems Laboratory; Report 8494-R-1; August, 1960.

6. J. Massey - A Time-Shared Analog Trigonometric Resolver; M.S. Thesis; M.I.T. Department of Electrical Engineering; June, 1960.

7. M. Connelly - Analog-Digital Computers for Real-Time Simulation; M.I.T. Electronic Systems Laboratory; Final Report ESL-FR-110; June, 1961.

8. W. Dunn, C. Eldert, and P. Levonian - A Digital Computer for Use in an Operational Flight Trainer - IRE Transactions on Electronic Computers, Vol. EC-4, June, 1955 (UDOFT: Universal Digital Operational Flight Trainer).

9. Julian Wargo, et al. - The UDOFT Flight Simulation System - Technical Documentary Report AMRL-TDR-63-133-6570th Aerospace Medical Research Laboratories - WPAFB - December, 1963.

10. M. Connelly - Real-Time Analog-Digital Computation; IRE Transactions on Electronic Computers; Vol. EC-11, No. 1; February, 1962.

11. L. Krasny - The Functional Design of a Special-Purpose Digital Computer for Real-Time Flight Simulation; M.I.T. Electronic Systems Laboratory; Report ESL-R-118; August, 1961.

12. E.L. Perry - Submicrosecond Simulation Computer Study Program, Part I, Requirements and Techniques Study, MRL-TDR-62-27 (I); Sylvania Electric Products, Inc.; May, 1962.

13. E.L. Perry - Submicrosecond Simulation Computer Study Program, Part II, Preliminary Design and System Synthesis; MRL-TDR-62-27(II); Sylvania Electric Products, Inc.; October, 1962.

14. K. Andresen and Duncan Ewing - A Study of Digital Computers for a Real-Time Training Simulation Research System - Technical Documentary Report AMRL-TDR-64-22 - Aerospace Medical Research Laboratories - WPAFB - May, 1964.

15. G. Korn - The Impact of Hybrid Analog-Digital Techniques on the Analog-Computer Art - Proc. IRE - p. 1077 - May, 1962.

16. Aerodynamic Data for the Design of the F100A Simulator - North American Aviation-NA-53-592 - July, 1953.

17. J. Prutsalis - UDOFT Simulation Program - Sylvania Electronic Systems - Needham, Mass. - Final Report FR-77-IN - May, 1960.

18. M. Connelly - Simulation of Aircraft - M.I.T. Servomechanisms Laboratory - Report 7591-R-1 - February 15, 1958.

19. Arthur Delagrange - Memory Driver Amplifier - D.S.R. 8823 - Memorandum No. 1-8 June, 1962.

20. Craig Bergman - A High Speed Sample and Hold Circuit - B.S. Thesis, M.I.T. Department of Electrical Engineering - May 18, 1962.

21. Hoffait and Thornton - Limitations of Transistor DC Amplifiers Proc. IEEE, Vol. 52, No. 2, - February, 1964.

22. M. Connelly - Computers for Aircraft Simulation - M.I.T. Electronic Systems Laboratory - Report 7591-R-2 - December, 1959.

23. A.B. Macnee - An Electronic Differential Analyzer - Proc. IRE - Vol. 37, No. 11 - November, 1949.

24. E.J. Angelo - An Electron-Beam Tube for Analogue Multiplication - Sc.D. Thesis, M.I.T., 1952.

25. Miller, Soltes, and Scott - Wideband Analog Function Multiplier - Electronics, Vol. 28 - February, 1955.

26. W.A. Pollock - A High-Speed Analog Multiplier Using a Beam-Deflection Tube - M S. Thesis, M.I.T. Department of Electrical Engineering - September, 1961.

27. S. Giser - An All-Electronic High-Speed Multiplier - M.I.T. Instrumentation Laboratory Report R-67 - November, 1953.

28. H. Dumas - A Pulsed Analog Multiplier-Divider - M.S. Thesis, M.I.T. Department of Electrical Engineering - May, 1958.

29. F. Leith - High-Speed Pulsed-Analog Multiplier - M.I.T. Electronic Systems Laboratory - D.S.R. 8494 - Memorandum No. 15 - June, 1960.

30. A.S. Blum - An Analog Multiplier-Divider Using Logarithms - M.S. Thesis - M.I.T. Department of Electrical Engineering - May, 1960.

31. A. Solbakken - Four-Quadrant Analog Multipliers - M.I.T. Electronic Systems Laboratory - D.S.R. 8823 - Memorandum No. 2 - September, 1962.

32. Chance, Williams, et al. - A Quarter-Square Multiplier Using a Segmented Parabolic Characteristic - Rev. Sci. Inst. - Vol. 22, No. 9 - September, 1951.

33. M. Fisher - A Wide Band Analogue Multiplier Using Crystal Diodes - Electronic Engineering - December, 1957.

34. L. Lofgren - Analog Multiplier Based on the Hall Effect - J. Appl. Physics - Vol. 29, No. 2 - February, 1958.

35. D. Bruck - Analog Multiplication from DC to Mc - Inst. and Control Systems - August, 1962.

36. R. Gocht - The Design of High Speed Analog Sample and Storage Gates - M.S. Thesis - M.I.T. Department of Electrical Engineering - August, 1959.

37. Nyder and Given - The Development of Several High Speed Linear Transmission Gates - B.S. Theses - M.I.T. Department of Electrical Engineering-May, 1958.

38. Millman and Puckett - Accurate Linear Bi-Directional Diode Gates - Proc. IRE - Vol. 43, No. 1 - January, 1955.

39. T. Brubaker - Precision Analog Memory Has Extended Frequency Response - Electronics - September 29, 1961.

40. W. Eddins - Sample and Hold Circuits - 1958 IRE National Convention Record - Vol. 6, Part 5.

41. D. Wortzman - Use of a Digital-Analog Arithmetic Unit Within a Digital Computer - Proc. Eastern Joint Computer Conf. - 1960.

42. G. Birkel - Mathematical Approach to Hybrid Computing - IRE National Symposium on Space Electronics and Telemetry - 1959.

43. M.C. Burns - High-Speed Hybrid Computer - IRE National Symposium on Space Electronics and Telemetry - 1959.

44. T. Hagan - Ambilog Computers - Adage Co. - Boston, Massachusetts - February, 1962.

45. Baker, McMahon, and Burgess - The Diamond Circuit - M.I.T. Lincoln Laboratory Technical Report No. 300 - 30 January, 1963.

46. J.K. Roberge - Experimental Results with a Diamond Circuit Sampling Chain - M.I.T. Electronic Systems Laboratory TM-3 - September 7, 1963.

47. A. Hakimoglu and R. Kulvin - Sampling Ten Million Words a Second - Electronics - February 7, 1964.

48. A. Solbakken - Design of High-Speed Electronic Switches - M.I.T. Electronic Systems Laboratory - D.S.R. 8823 - Memorandum 3 - January 11, 1963.

49. H. Bourland - An Analog-Digital Multiplier-Divider for Function Generation - M.S. Thesis - M.I.T. Department of Electrical Engineering - August, 1957.

50. K.G. Williams - A Transistorized Electronic Switch - U.S. Naval Ordinance Laboratory-NOLTR 62-139 - 5 September, 1962.

51. T.A. Brubaker - A Non-Saturating Transistor Switch for Analog/Hybrid Instrumentation and Computers - University of Arizona (Department of Electrical Engineering) - ACL Memorandum 78 - May, 1963.

52. A.K. Susskind, editor - Notes on Analog-Digital Conversion Techniques - M.I.T. Press - 1958.

53. J.S. Hill - A Fast Digital-Analog Converter and Its Accuracy Limitations - M.S. Thesis - M.I.T. Department of Electrical Engineering - November, 1959.

54. Analog-Digital Conversion Handbook - Digital Equipment Corporation, Maynard, Mass. 1964.

55. D. Savitt - A High-Speed Analog-to-Digital Encoder - M.S. Thesis - M.I.T. Department of Electrical Engineering - August, 1957 (condensed version published in IRE-PGEC-Vol. EC-8, No. 1 - March, 1959.

56. J. Yates - A Low Cost, Solid-State Analog-Digital Converter - B.S. Thesis - M.I.T. Department of Electrical Engineering - May, 1960.

57. Digital Modules - Digital Equipment Corporation Manual A-705-Maynard, Massachusetts (new editions issued periodically).

58. R. Howe, E. Gilbert - A New Resolving Method for Analog Computers - WADC Technical Note 55-468 - January, 1956.

59. E.A. Goldberg - A Generator for Functions of Two Variables - M.S. Thesis - M.I.T. Department of Electrical Engineering - August, 1956.

60. T. Anderson - An Analog Electro-Optical Multiplier - M.S. Thesis - M.I.T. Department of Electrical Engineering - September, 1963.

61. Gas Turbine Installation Handbook, JT3 Turbojet Engines - Vol. II - Pratt and Whitney Aircraft Co. - East Hartford, Conn.-July, 1956.

62. JT3 (J57) Gas Turbine (Specifications, Performance, Correction Factor Curves) - Pratt and Whitney Aircraft Company - Last Revision-October, 1958.

63. L.W. Plum - UDOFT Test Report (Sections II, V, VI, VII, VIII, IX) - Sylvania Electronic Systems - Needham, Mass. - April, 1960.

64. R.M. Howe - An Investigation of the Relative Effect of Stability Derivatives on the Dynamic Characteristics of the F100A - Department of Aeronautical Engineering, University of Michigan - WADCTN 57-297 - December, 1957.

BLANK PAGE

APPENDIX A

ABBREVIATED PDP-1 INSTRUCTION LIST

BASIC GROUP

Instruction	Explanation	Oper. Time (μsec)
add Y	Add C(Y) to C(AC)	10
adm Y	Add C(AC) to C(Y)	10
and Y	Logical AND C(Y) with C(AC)	10
cal Y	Equals jda 100	10
dac Y	Deposit C(AC) in Y	10
dap Y	Deposit contents of address part of AC in Y	10
dio Y	Deposit C(IO) in Y	10
dip Y	Deposit contents of instruction part of AC in Y	10
div Y	Divide	40 max
dzm Y	Deposit zero in Y	10
idx Y	Index (add one) C(Y), leave in Y and AC	10
ior Y	Inclusive OR C(Y) with C(AC)	10
iot Y	In-out transfer, see below	
isp Y	Index and skip if result is positive	10
jda Y	Equals dac Y and jsp Y + 1	10
jdp Y	Deposit C(PC) in Y jump to Y + 1	
jmp Y	Take next instruction from Y	5
jsp Y	Jump to Y and save program counter in AC	5
lac Y	Load the AC with C(Y)	10
lai	Transfer C(IO) to AC	
law N	Load the AC with the number N	5
law-N	Load the AC with the number -N	5
lia	Transfer C(AC) to IO	
lio Y	Load IO with C(Y)	10
mul Y	Multiply	25 max
opr	Operate, see below	5
sad Y	Skip next instruction if C(AC) ≠ C(Y)	10
sas Y	Skip next instruction if C(AC) = C(Y)	10
sft	Shift, see below	5
skp	Skip, see below	5
sub Y	Subtract C(Y) from C(AC)	10
swp	C(IO) to AC C(AC) to IO	
xct Y	Execute instruction in Y	5+
xor Y	Exclusive OR C(Y) with C(AC)	10

OPERATE GROUP

cla	Clear AC	5
clf	Clear selected Program Flag (f = flag no.)	5
cli	Clear IO	5
clo	Clear overflow	5
cma	Complement AC	5
hlt	Halt	5
lap	Load AC with Program Counter	5
lat	Load AC from Test Word switches	5
nop	No operation	5
stf	Set selected Program Flag	5

IN-OUT TRANSFER GROUP

Perforated Tape Reader

rpa	Read Perforated Tape Alphanumeric
rpb	Read Perforated Tape Binary
rrb	Read Reader Buffer

Perforated Tape Punch

ppa	Punch Perforated Tape Alphanumeric
ppb	Punch Perforated Tape Binary

Alphanumerical On-Line Typewriter

Instruction	Explanation	Oper. Time (μ sec)
tyo	Type Out	
tyi	Type In	

Precision CRT Display Type 30

dpy	Display One Point	

SKIP GROUP

sma	Skip on minus AC	5
sni	Skip on non-zero IO	5
spa	Skip on plus AC	5
spi	Skip on plus IO	5
sza	Skip on ZERO (+0) AC	5
szf	Skip on ZERO flag	5
szo	Skip on ZERO overflow (and clear overflow)	5
szs	Skip on ZERO sense switch	5

SHIFT/ROTATE GROUP

ral	Rotate AC left	5
rar	Rotate AC right	5
rcl	Rotate combined AC and IO left	5
rcr	Rotate combined AC and IO right	5
ril	Rotate IO left	5
rir	Rotate IO right	5
sal	Shift AC left	5
sar	Shift AC right	5
scl	Shift combined AC and IO left	5
scr	Shift combined AC and IO right	5
sil	Shift IO left	5
sir	Shift IO right	5

PDP-1 SYSTEMS PROGRAMS

The PDP-1 has four major systems programs. Two of these are assembly programs, one an on-line debugging routine, and the fourth a symbolic tape editor.

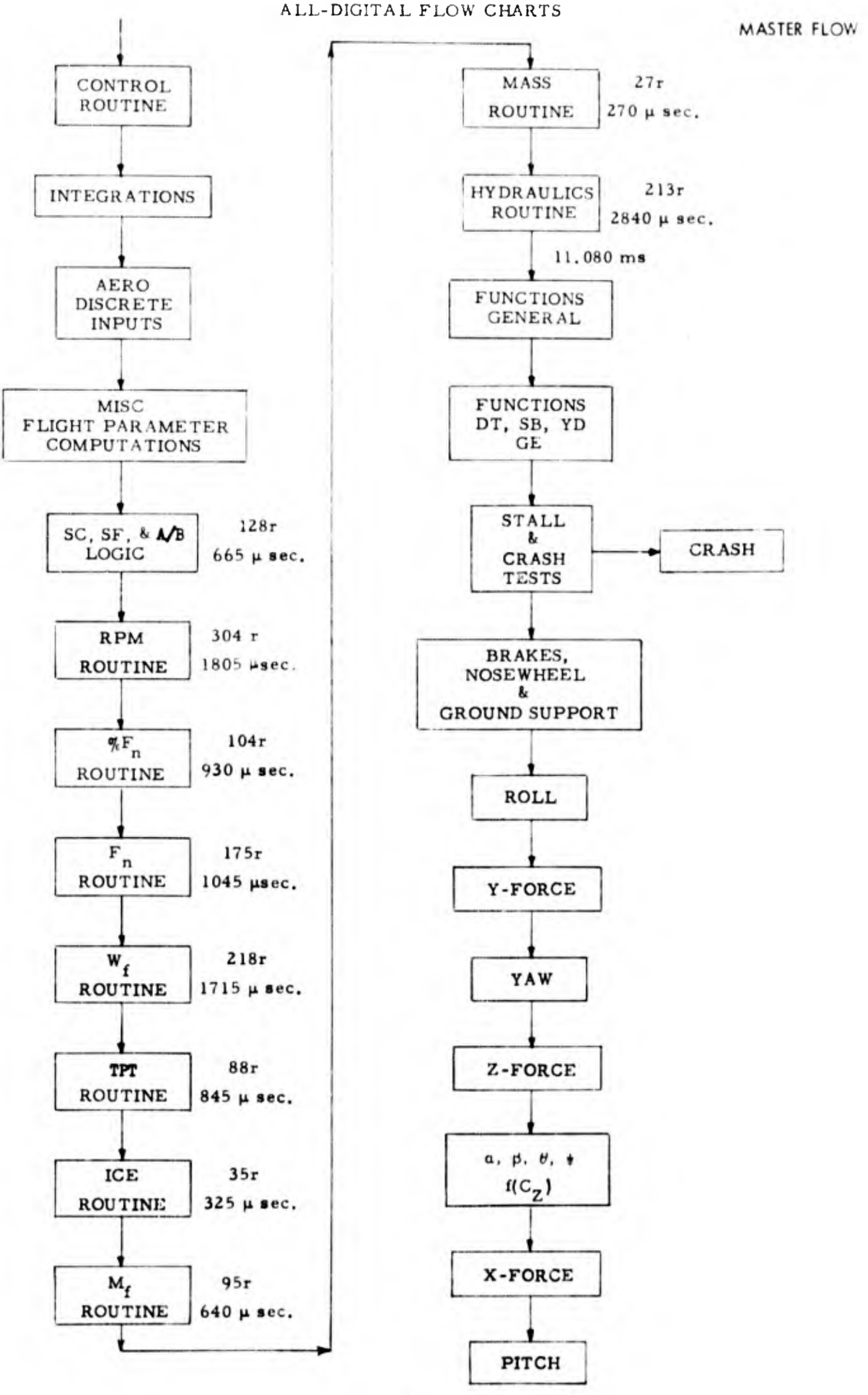
MACRO and MIDAS, the two assembly programs, assemble a symbolic input tape to produce a binary machine language tape (also a symbol tape and symbol listing if desired). MACRO is more often used as it is large enough to handle most ordinary programs. MIDAS is slightly more powerful with greater symbol storage capacity and relocatable facilities. Both assemblers have the feature of being able to define a "MACRO" (a long string of machine instructions) and to call this string whenever needed with only the use of the macro name.

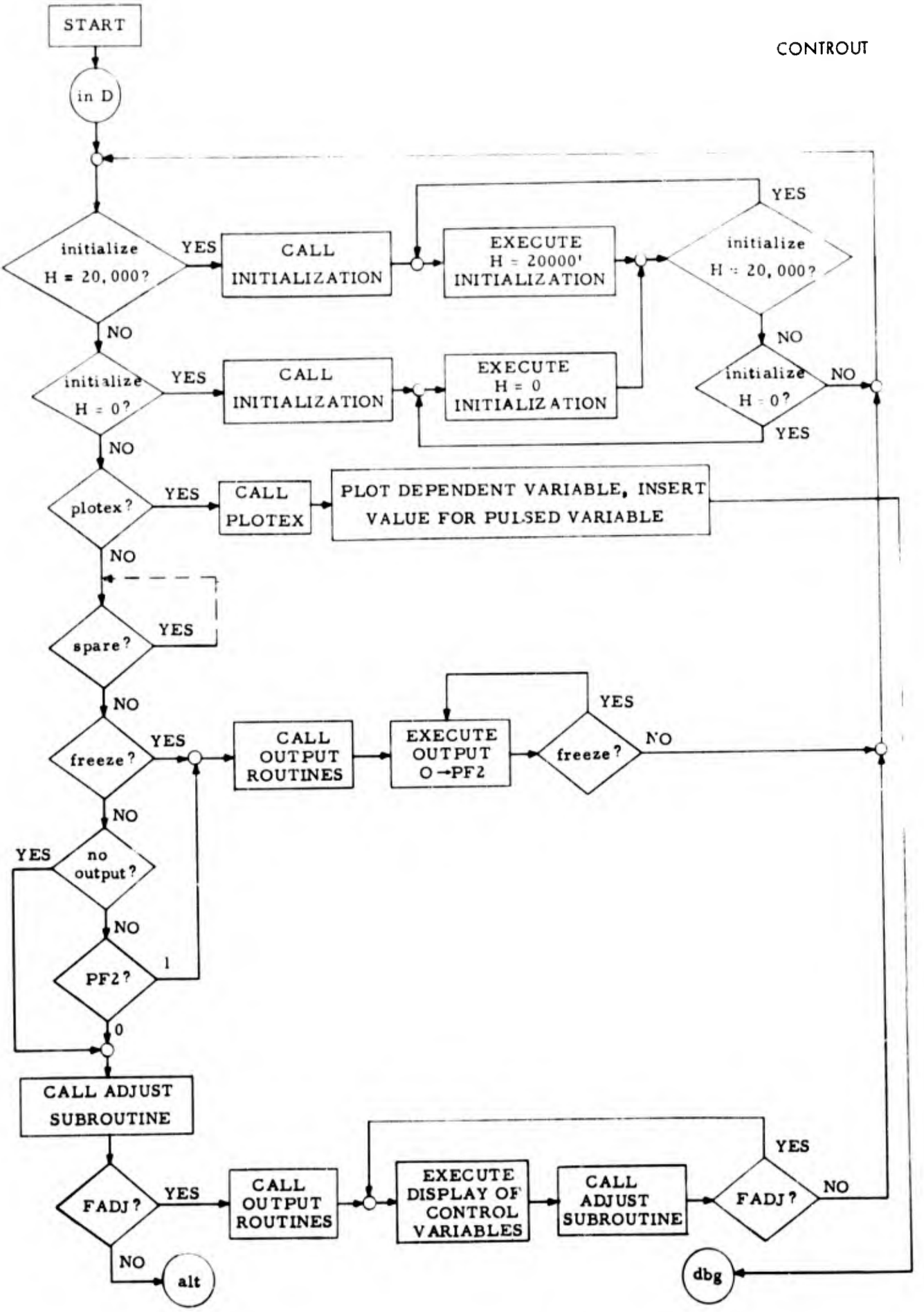
DDT (or Digital Debugging Tape) is a very powerful on-line debugging routine. Its features include on-line interrogation and modification of the binary program in symbolic form (using the symbol tape from MIDAS or MACRO), word searches, verification of tape against core storage, breakpoint insertion, reading and punching of binary machine language tapes, and others.

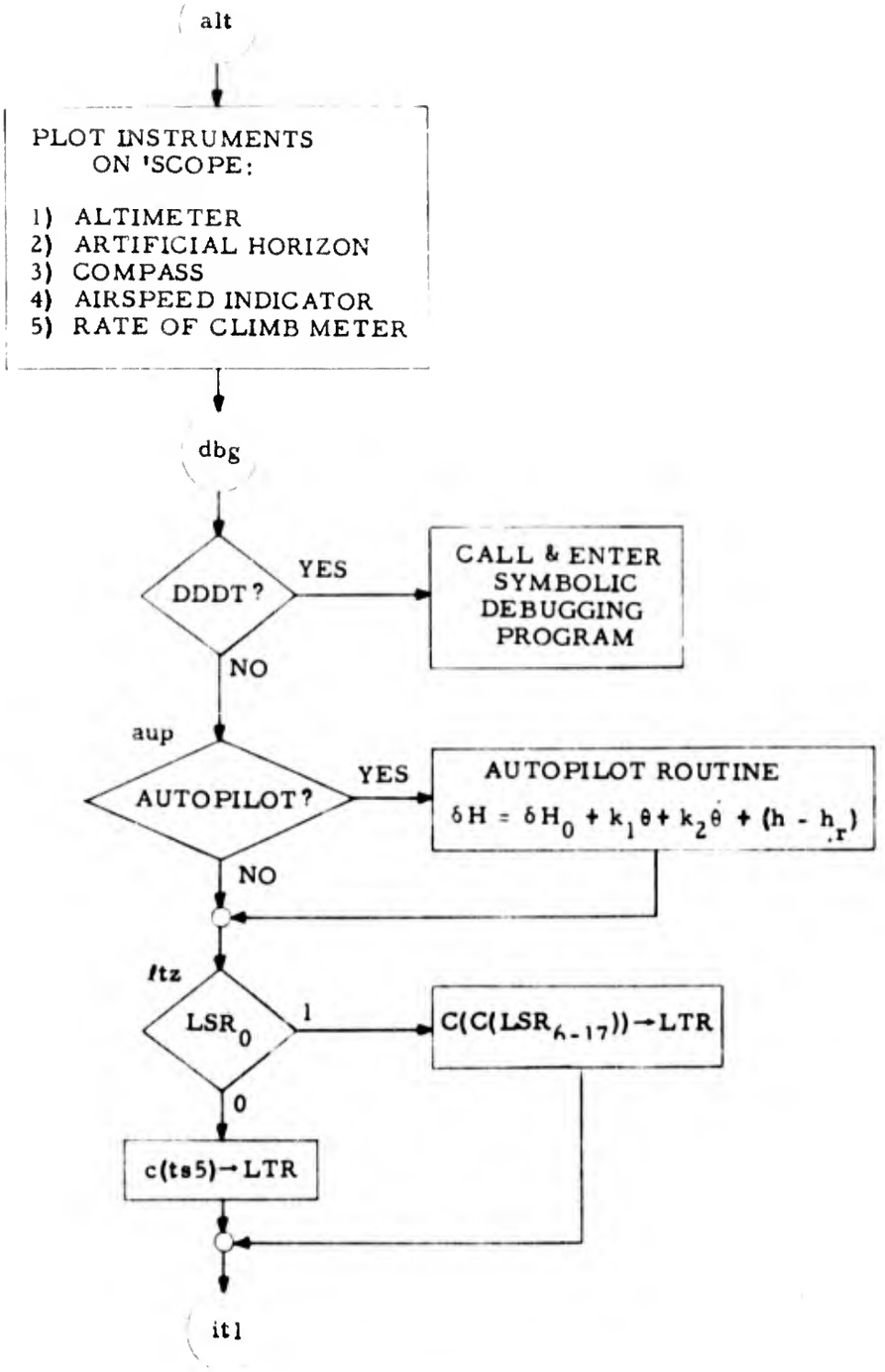
EXPENSIVE TYPEWRITER is the symbolic tape editor. It allows the user to modify, punch, or list his symbolic program on a line by line basis so he would not have to resort to retyping the entire program or making patch corrections in the final binary form.

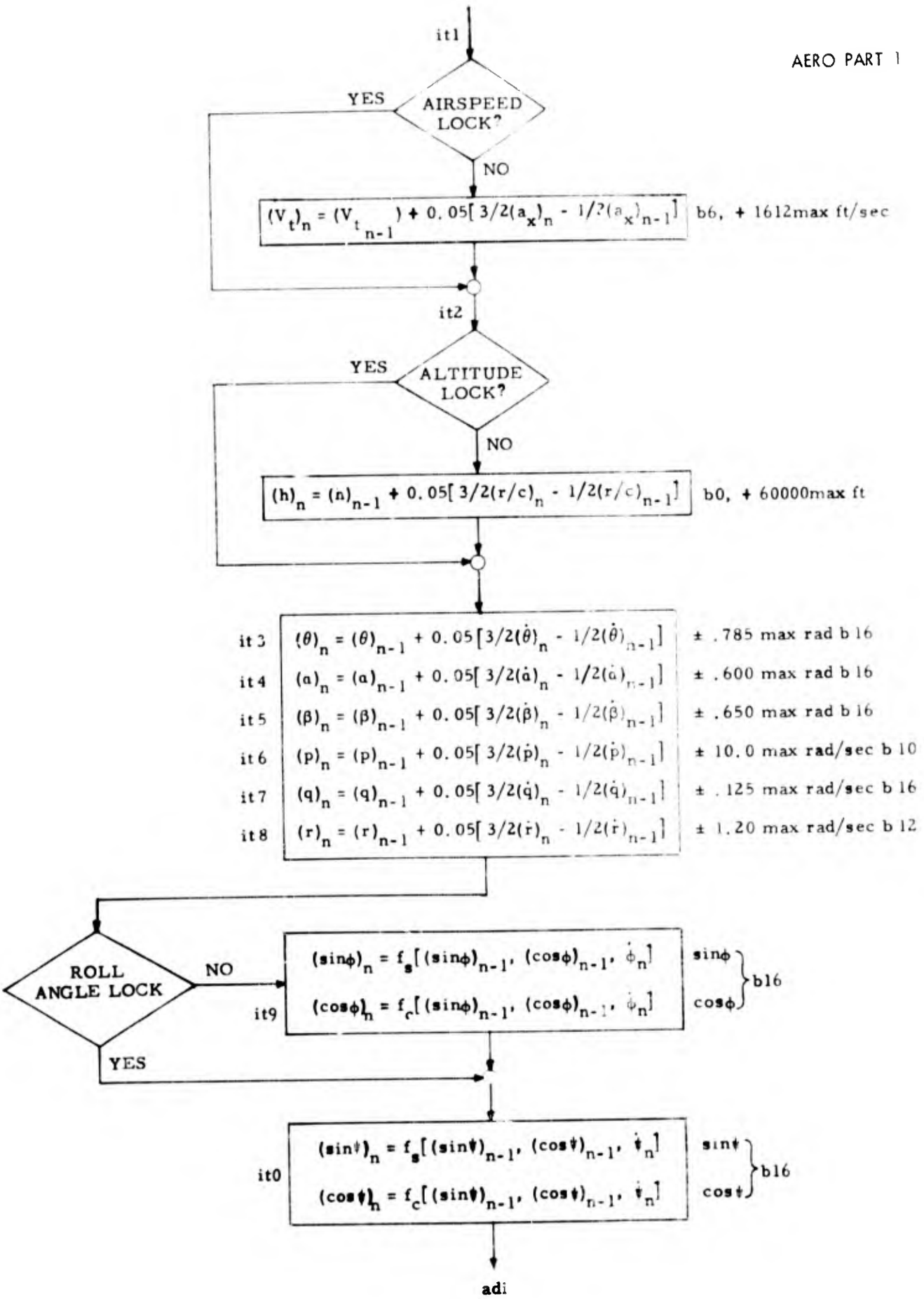
APPENDIX B

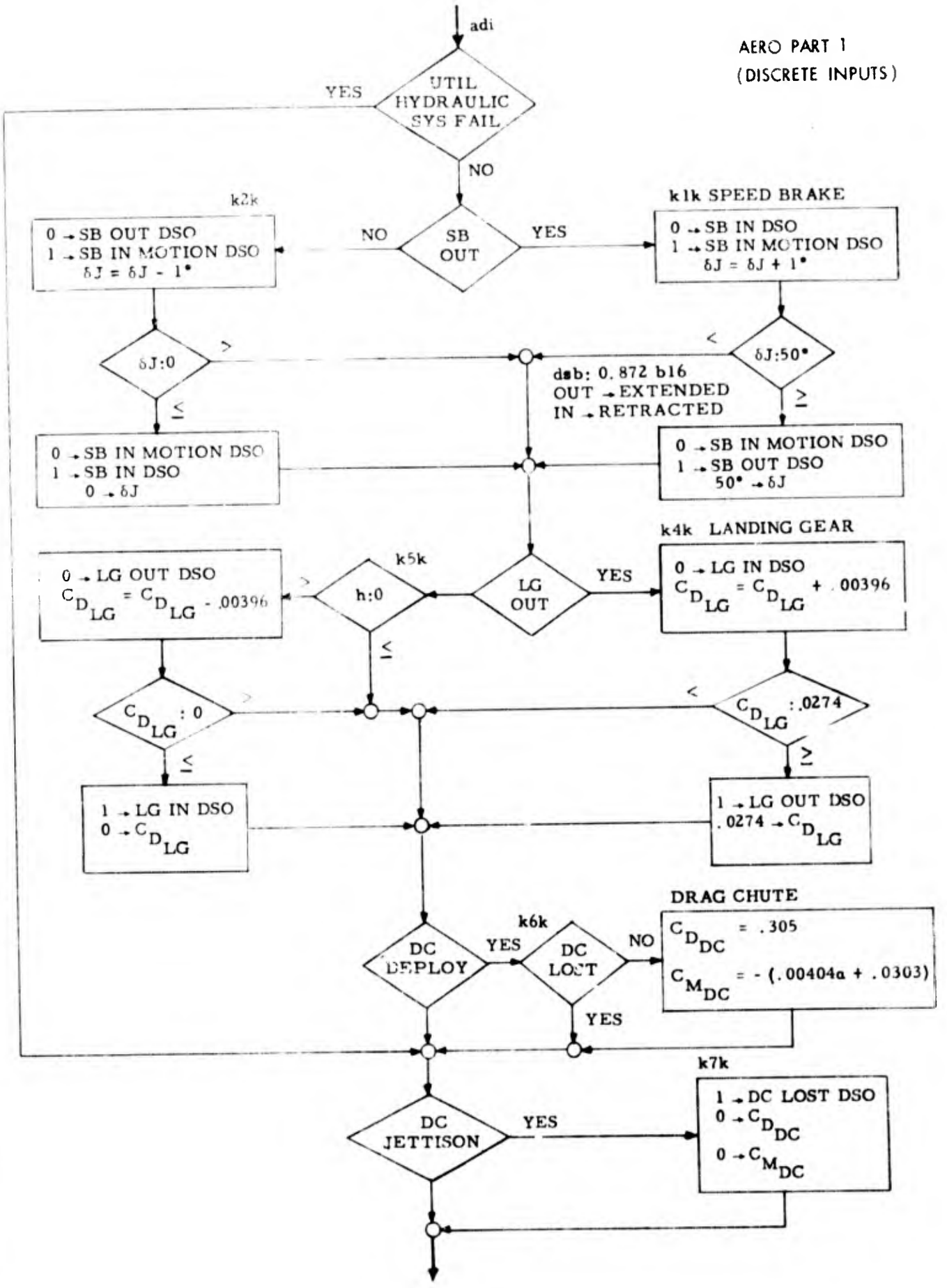
ALL-DIGITAL FLOW CHARTS

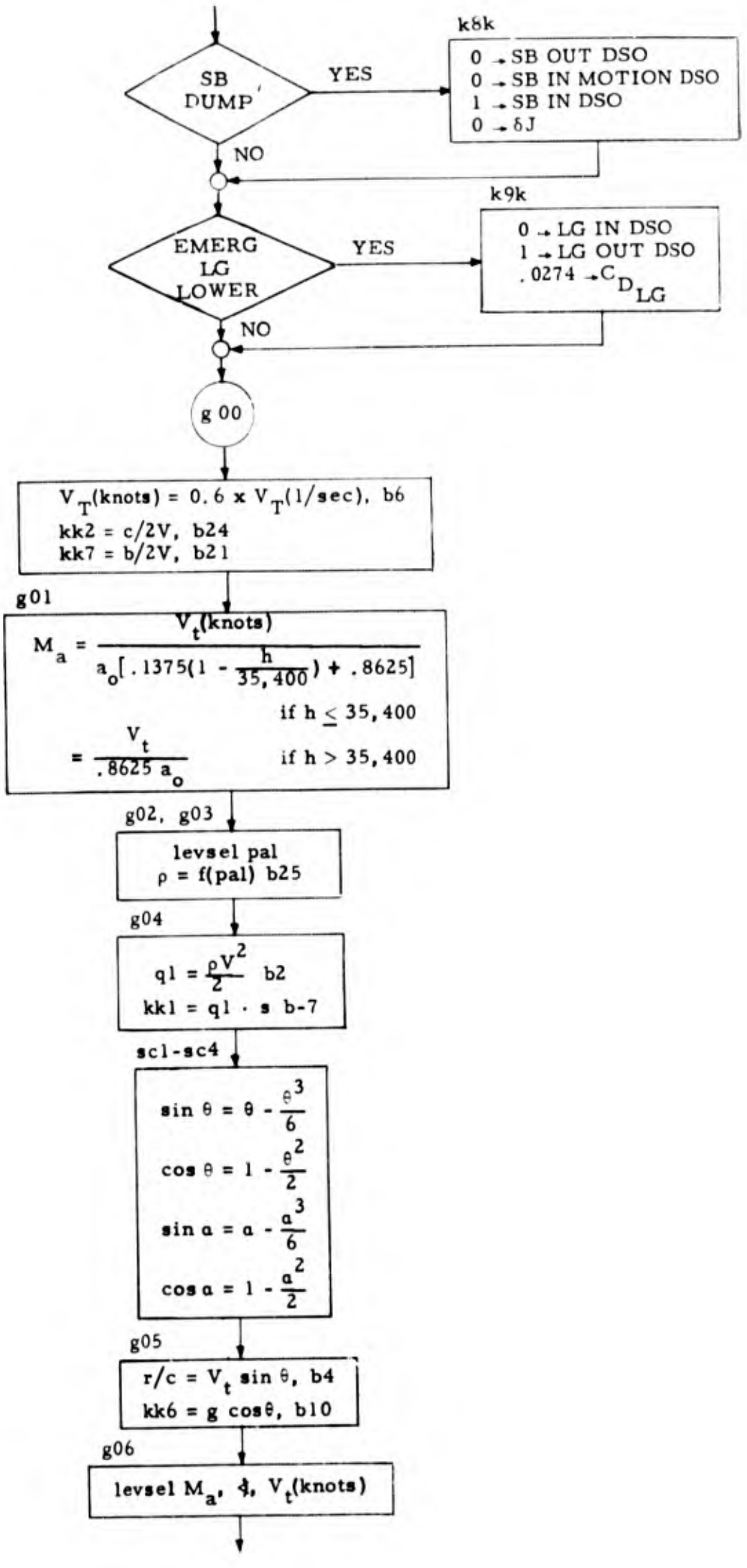


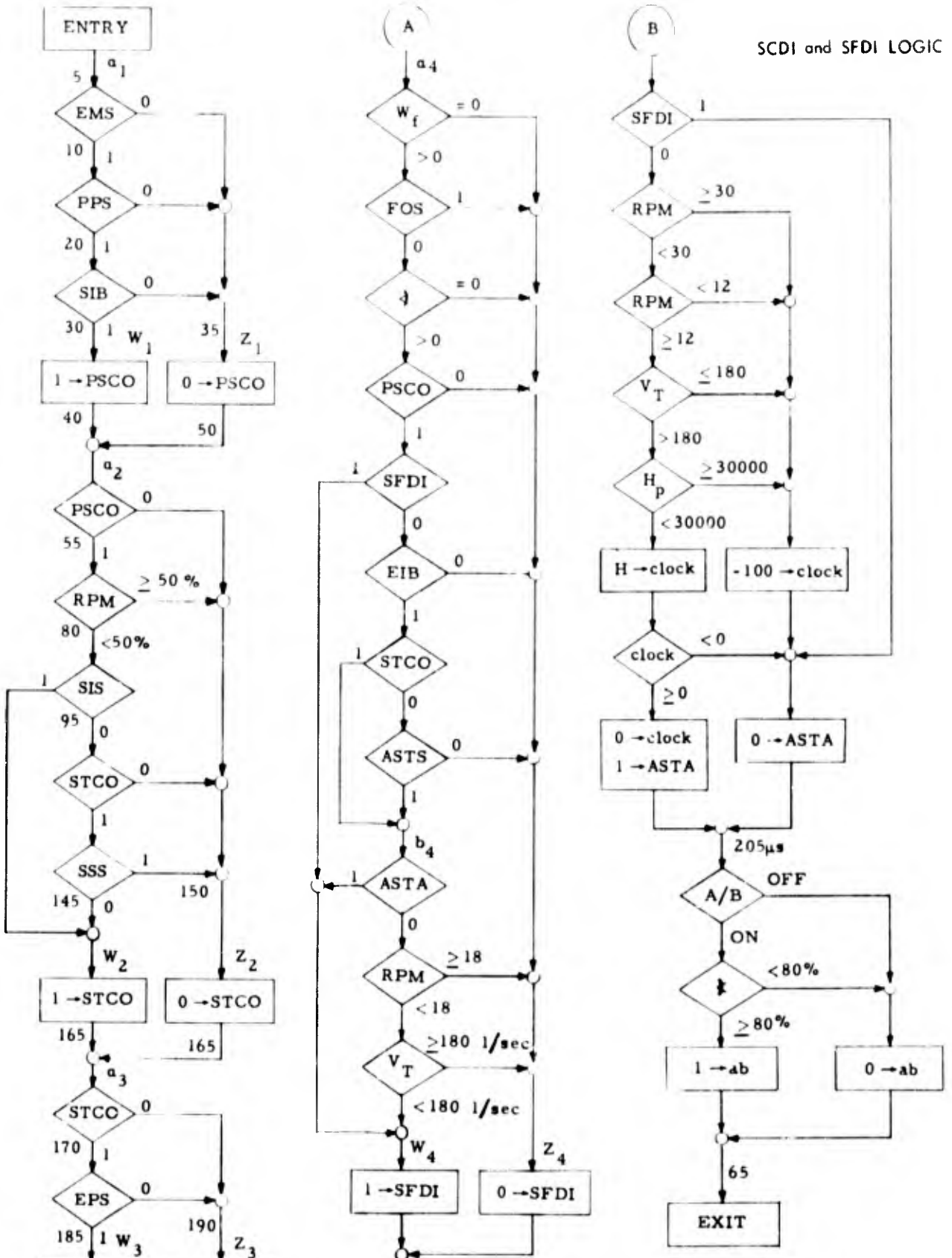

START
in D
initialize H = 20,000?
YES
CALL INITIALIZATION
EXECUTE H = 20000' INITIALIZATION
initialize H = 20,000?
YES
NO
initialize H = 0?
YES
CALL INITIALIZATION
EXECUTE H = 0 INITIALIZATION
initialize H = 0?
NO
YES
NO
plotex?
YES
CALL PLOTEX
PLOT DEPENDENT VARIABLE, INSERT VALUE FOR PULSED VARIABLE
NO
spare?
YES
NO
freeze?
YES
CALL OUTPUT ROUTINES
EXECUTE OUTPUT O→PF2
freeze?
YES
NO
NO
no output?
YES
NO
PF2?
1
0
CALL ADJUST SUBROUTINE
FADJ?
YES
CALL OUTPUT ROUTINES
EXECUTE DISPLAY OF CONTROL VARIABLES
CALL ADJUST SUBROUTINE
FADJ?
YES
NO
NO
alt
dbg


alt
PLOT INSTRUMENTS ON 'SCOPE:
1) ALTIMETER
2) ARTIFICIAL HORIZON
3) COMPASS
4) AIRSPEED INDICATOR
5) RATE OF CLIMB METER
dbg
DDDT?
YES
CALL & ENTER SYMBOLIC DEBUGGING PROGRAM
NO
aup
AUTOPILOT?
YES
AUTOPILOT ROUTINE
δH = δH0 + k1θ + k2θ̇ + (h - hr)
NO
ltz
LSR0
1
C(C(LSR6-17))→LTR
0
c(ts5)→LTR
itl


it1
AIRSPEED LOCK?
YES
NO
$(V_t)_n = (V_t)_{n-1} + 0.05[3/2(a_x)_n - 1/2(a_x)_{n-1}]$
b6, + 1612max ft/sec
it2
ALTITUDE LOCK?
YES
NO
$(h)_n = (h)_{n-1} + 0.05[3/2(r/c)_n - 1/2(r/c)_{n-1}]$
b0, + 60000max ft
it3 $(\theta)_n = (\theta)_{n-1} + 0.05[3/2(\dot{\theta})_n - 1/2(\dot{\theta})_{n-1}]$ ± .785 max rad b16
it4 $(\alpha)_n = (\alpha)_{n-1} + 0.05[3/2(\dot{\alpha})_n - 1/2(\dot{\alpha})_{n-1}]$ ± .600 max rad b16
it5 $(\beta)_n = (\beta)_{n-1} + 0.05[3/2(\dot{\beta})_n - 1/2(\dot{\beta})_{n-1}]$ ± .650 max rad b16
it6 $(p)_n = (p)_{n-1} + 0.05[3/2(\dot{p})_n - 1/2(\dot{p})_{n-1}]$ ± 10.0 max rad/sec b10
it7 $(q)_n = (q)_{n-1} + 0.05[3/2(\dot{q})_n - 1/2(\dot{q})_{n-1}]$ ± .125 max rad/sec b16
it8 $(r)_n = (r)_{n-1} + 0.05[3/2(\dot{r})_n - 1/2(\dot{r})_{n-1}]$ ± 1.20 max rad/sec b12
ROLL ANGLE LOCK
NO
YES
it9 $(\sin\phi)_n = f_s[(\sin\phi)_{n-1}, (\cos\phi)_{n-1}, \dot{\phi}_n]$ sinφ
$(\cos\phi)_n = f_c[(\sin\phi)_{n-1}, (\cos\phi)_{n-1}, \dot{\phi}_n]$ cosφ
b16
it0 $(\sin\psi)_n = f_s[(\sin\psi)_{n-1}, (\cos\psi)_{n-1}, \dot{\psi}_n]$ sinψ
$(\cos\psi)_n = f_c[(\sin\psi)_{n-1}, (\cos\psi)_{n-1}, \dot{\psi}_n]$ cosψ
b16
adi


adi
AERO PART 1
(DISCRETE INPUTS)
UTIL HYDRAULIC SYS FAIL
YES
NO
k2k
0 → SB OUT DSO
1 → SB IN MOTION DSO
δJ = δJ - 1°
NO
SB OUT
YES
k1k SPEED BRAKE
0 → SB IN DSO
1 → SB IN MOTION DSO
δJ = δJ + 1°
δJ:0
>
≤
δJ:50°
<
≥
dsb: 0.872 b16
OUT → EXTENDED
IN → RETRACTED
0 → SB IN MOTION DSO
1 → SB IN DSO
0 → δJ
0 → SB IN MOTION DSO
1 → SB OUT DSO
50° → δJ
k4k LANDING GEAR
k5k
0 → LG OUT DSO
$C_{D_{LG}} = C_{D_{LG}} - .00396$
h:0
LG OUT
YES
0 → LG IN DSO
$C_{D_{LG}} = C_{D_{LG}} + .00396$
$C_{D_{LG}}$: 0
$C_{D_{LG}}$: .0274
1 → LG IN DSO
0 → $C_{D_{LG}}$
1 → LG OUT DSO
.0274 → $C_{D_{LG}}$
DRAG CHUTE
k6k
DC DEPLOY
YES
DC LOST
NO
$C_{D_{DC}}$ = .305
$C_{M_{DC}}$ = - (.00404α + .0303)
YES
k7k
DC JETTISON
YES
1 → DC LOST DSO
0 → $C_{D_{DC}}$
0 → $C_{M_{DC}}$


SB DUMP
YES
k8k
0 → SB OUT DSO
0 → SB IN MOTION DSO
1 → SB IN DSO
0 → δJ
NO
EMERG LG LOWER
YES
k9k
0 → LG IN DSO
1 → LG OUT DSO
.0274 → $C_{D_{LG}}$
NO
g 00
V_T(knots) = 0.6 x V_T(1/sec), b6
kk2 = c/2V, b24
kk7 = b/2V, b21
g01
$M_a = \frac{V_t(\text{knots})}{a_o[.1375(1 - \frac{h}{35,400}) + .8625]}$ if $h \leq 35,400$
$= \frac{V_t}{.8625\, a_o}$ if $h > 35,400$
g02, g03
levsel pal
ρ = f(pal) b25
g04
$q1 = \frac{\rho V^2}{2}$ b2
kk1 = q1 · s b-7
sc1-sc4
$\sin\theta = \theta - \frac{\theta^3}{6}$
$\cos\theta = 1 - \frac{\theta^2}{2}$
$\sin\alpha = \alpha - \frac{\alpha^3}{6}$
$\cos\alpha = 1 - \frac{\alpha^2}{2}$
g05
r/c = $V_t \sin\theta$, b4
kk6 = g cosθ, b10
g06
levsel M_a, ϕ, V_t(knots)

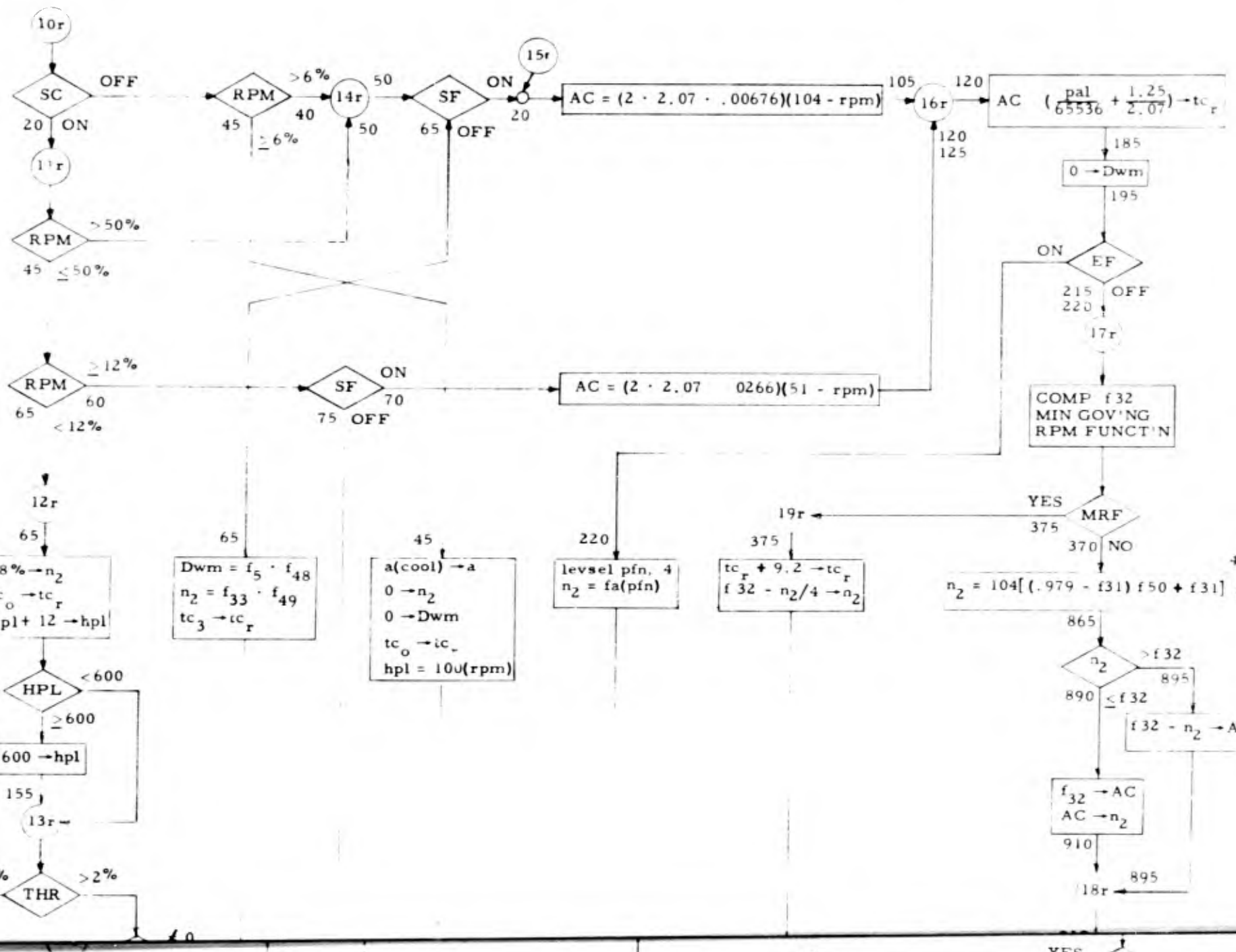


SCDI and SFDI LOGIC
UDOFT: WIRED
PDP-1: PROGRAMMED
W.C. TIME = 665μs
128 regs.

ts4:
0 - EMS
1 - PPS
2 - SIB
3 - SIS
4 - SSS
5 - EPS
6 - FOS
7 - EIB
8 - ASTS

A

R P M ROUTINE
(MOD UDOFT)



B

65
18% → n_2
tc_o → tc_r
hpl + 12 → hpl
HPL: < 600 / ≥ 600
600 → hpl
155
13r
≤ 2% / THR / > 2%
hp: ≠ 0 / = 0
a(cool) → a
200
a(hot) → a
205
640

65
Dwm = $f_5 \cdot f_{48}$
$n_2 = f_{33} \cdot f_{49}$
tc_3 → tc_r

45
a(cool) → a
0 → n_2
0 → Dwm
tc_o → tc_r
hpl = 100(rpm)

220
levsel pfn, 4
n_2 = fa(pfn)

375
tc_r + 9.2 → tc_r
f 32 - $n_2/4$ → n_2

19r
YES
MRF
375
370 NO

495
n_2 = 104[(.979 - f31) f50 + f31] f2

865
n_2: > f32 / ≤ f32
890
895
f 32 - n_2 → AC
f_{32} → AC
AC → n_2
910
18r
895

910
AC → mgf
920

75
145
735
450
20 r (entry to rpm lag)

1000
1000
955
945

OFF
AB
945
940 ON

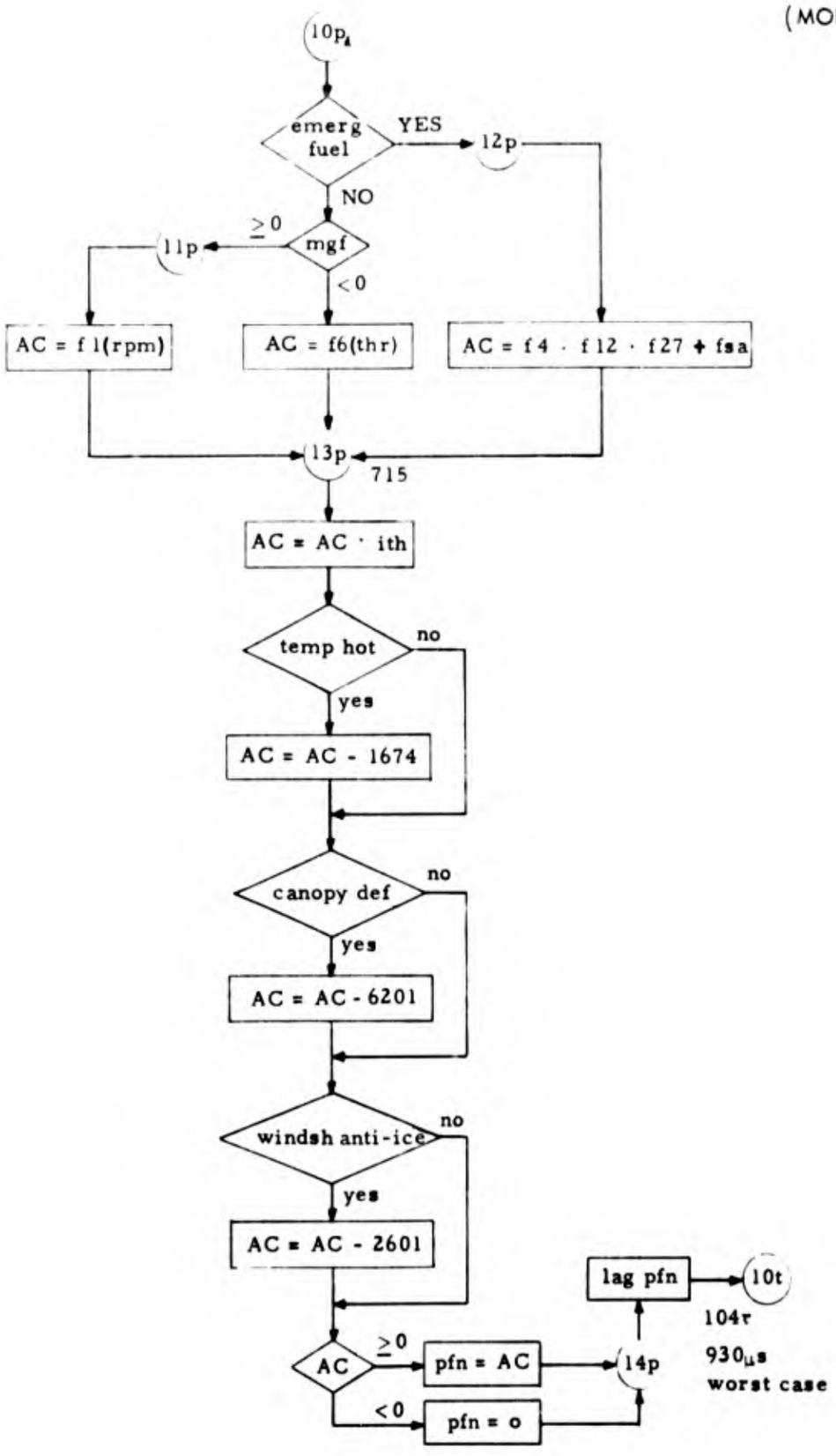
NO
NFC
955
950 YES
0.8n_2 → n_2
1000

0
tt 1 = $(\frac{1}{tcr})n_2 + (1 - \frac{1}{tcr})$ rpm
tt 2 = tt 1 - rpm
AC = |tt 2|
200
AC: < / > 1/2 %
215
tt 2 → IO
1/2 % → AC
220 ≤ 1/2 %
235
tt 1 → AC
235
IO: < 0 / ≥ 0
- AC → AC
245
AC = AC + rpm
255
255
AC → rpm
265
21r
levsel rpm
XIT
1805 μs

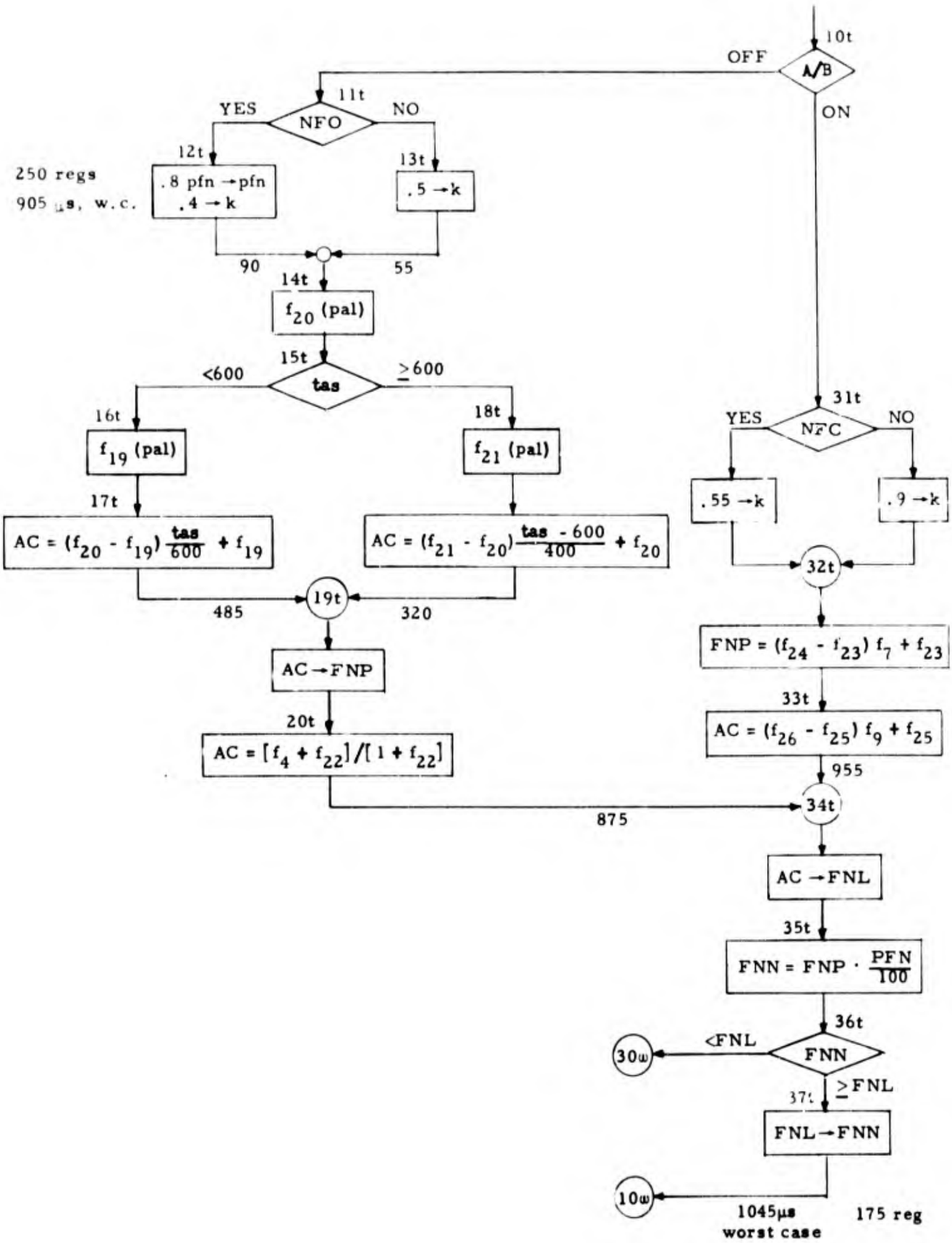
1.805 ms
worst case
304r

BLANK PAGE

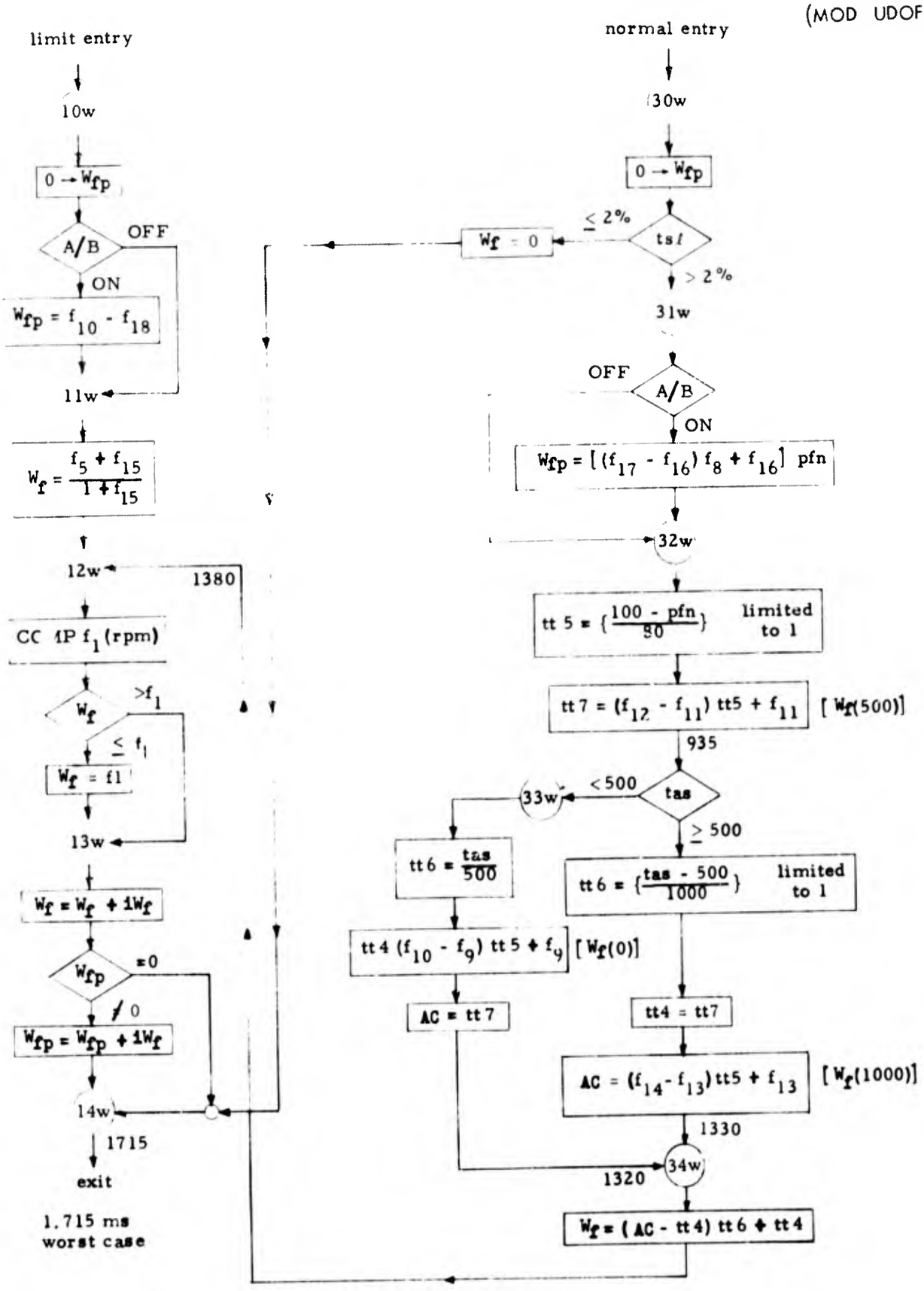
pfn
(MOD UDOFT)



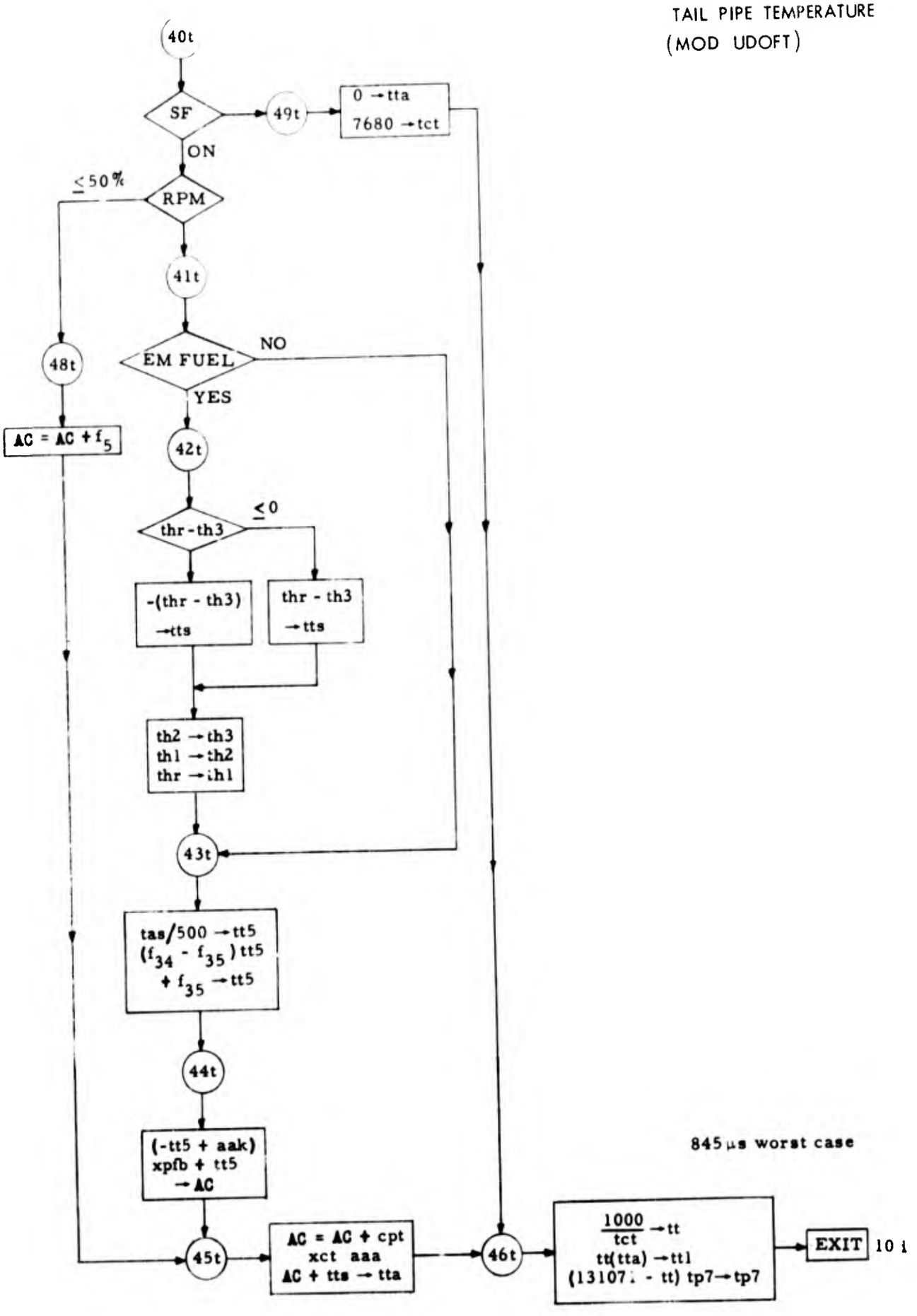
THRUST
(MOD UDOFT)



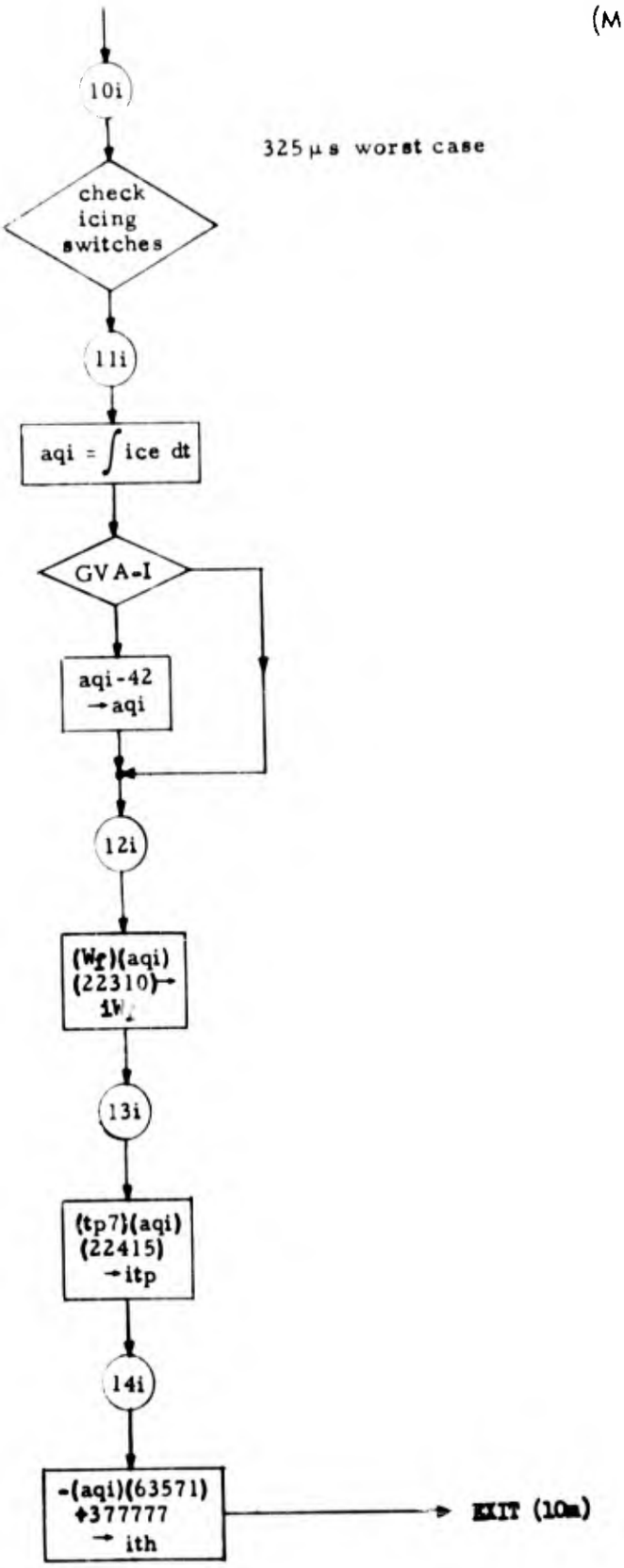
FUEL FLOW
(MOD UDOFT)



TAIL PIPE TEMPERATURE
(MOD UDOFT)

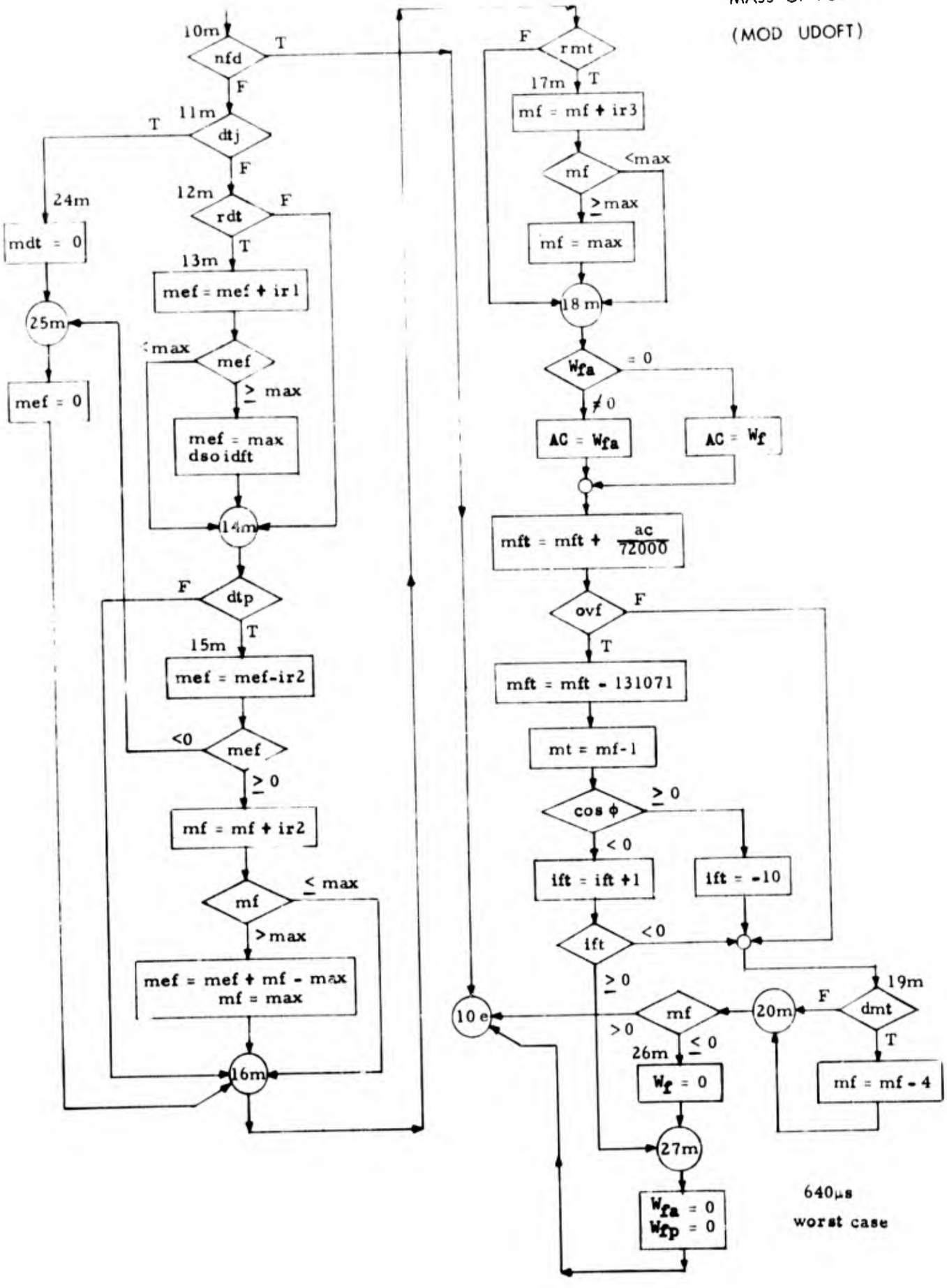


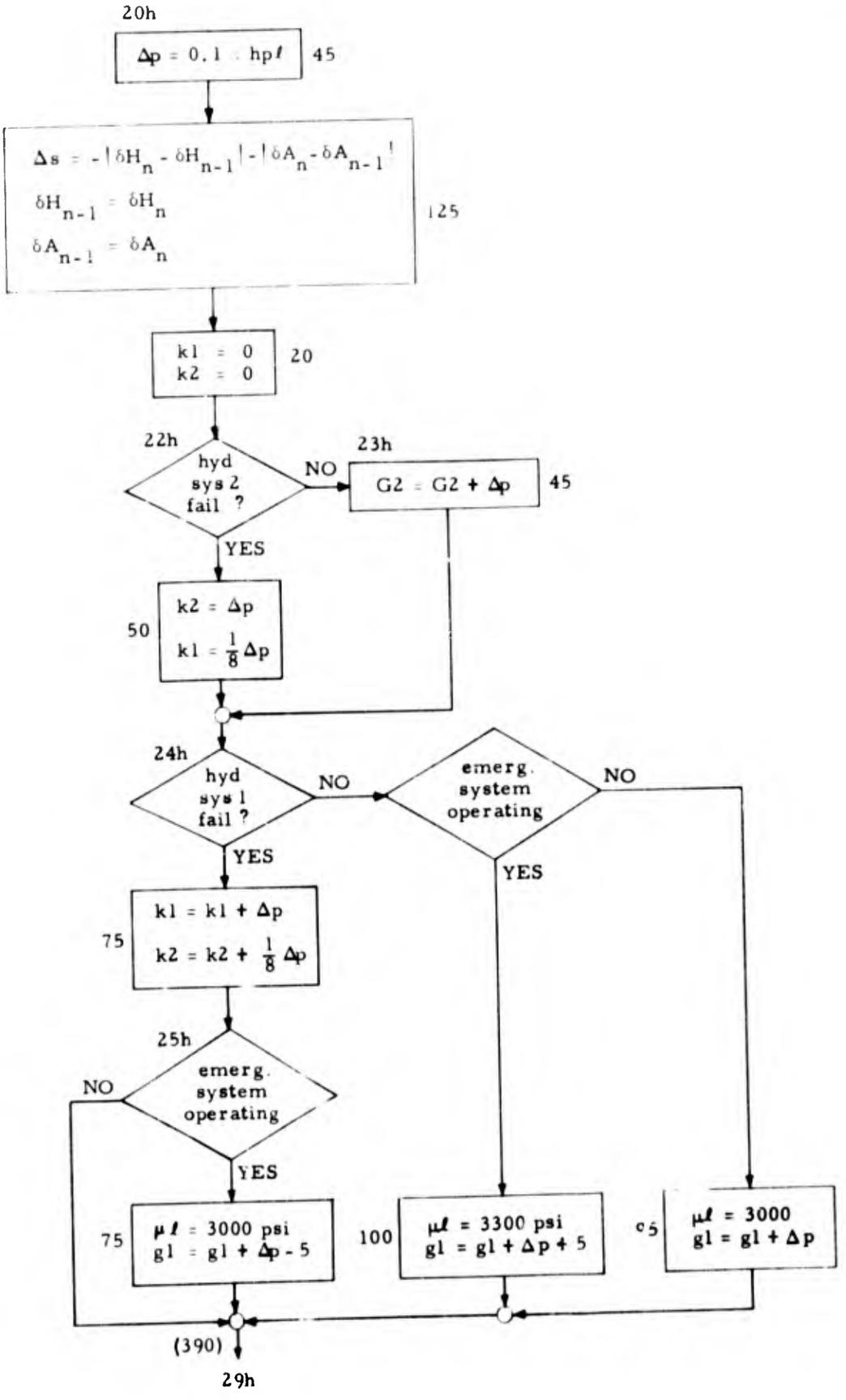
ICE ROUTINE
(MOD UDOFT)

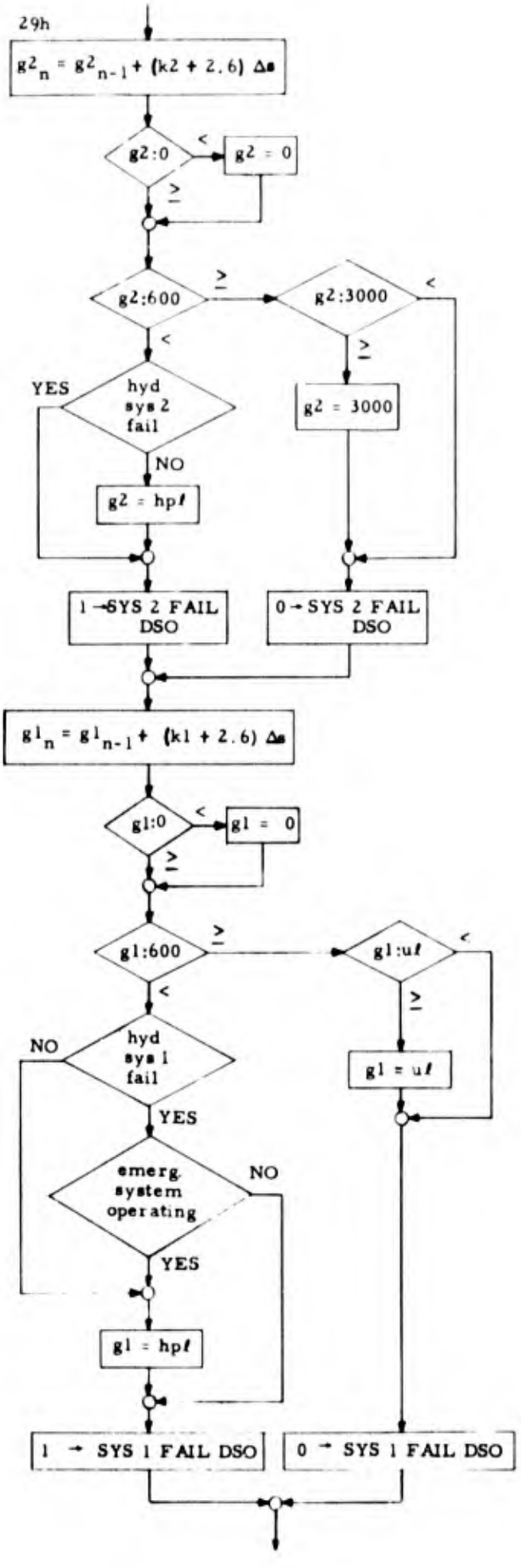


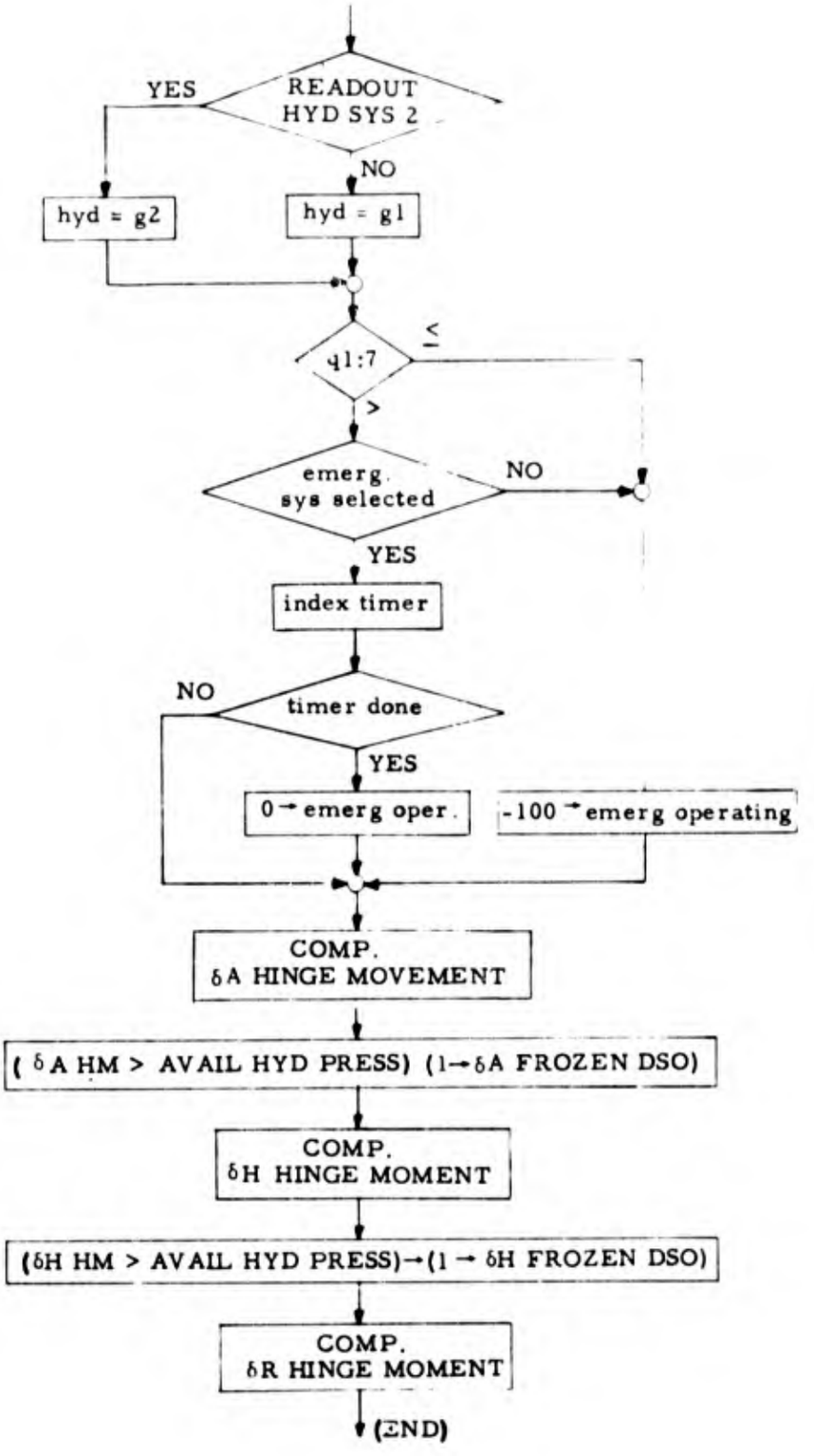
MASS OF FUEL ROUT

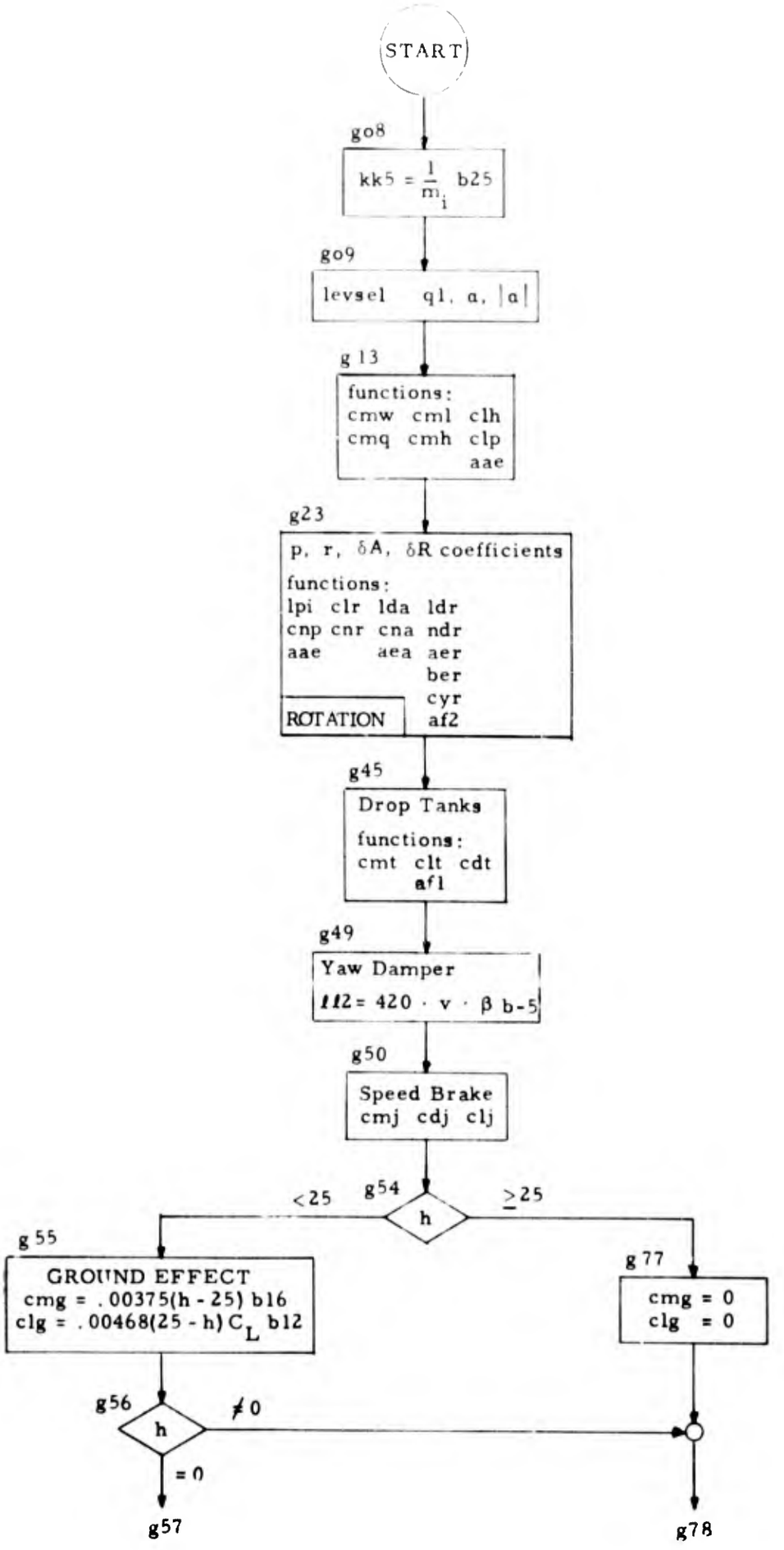
(MOD UDOFT)

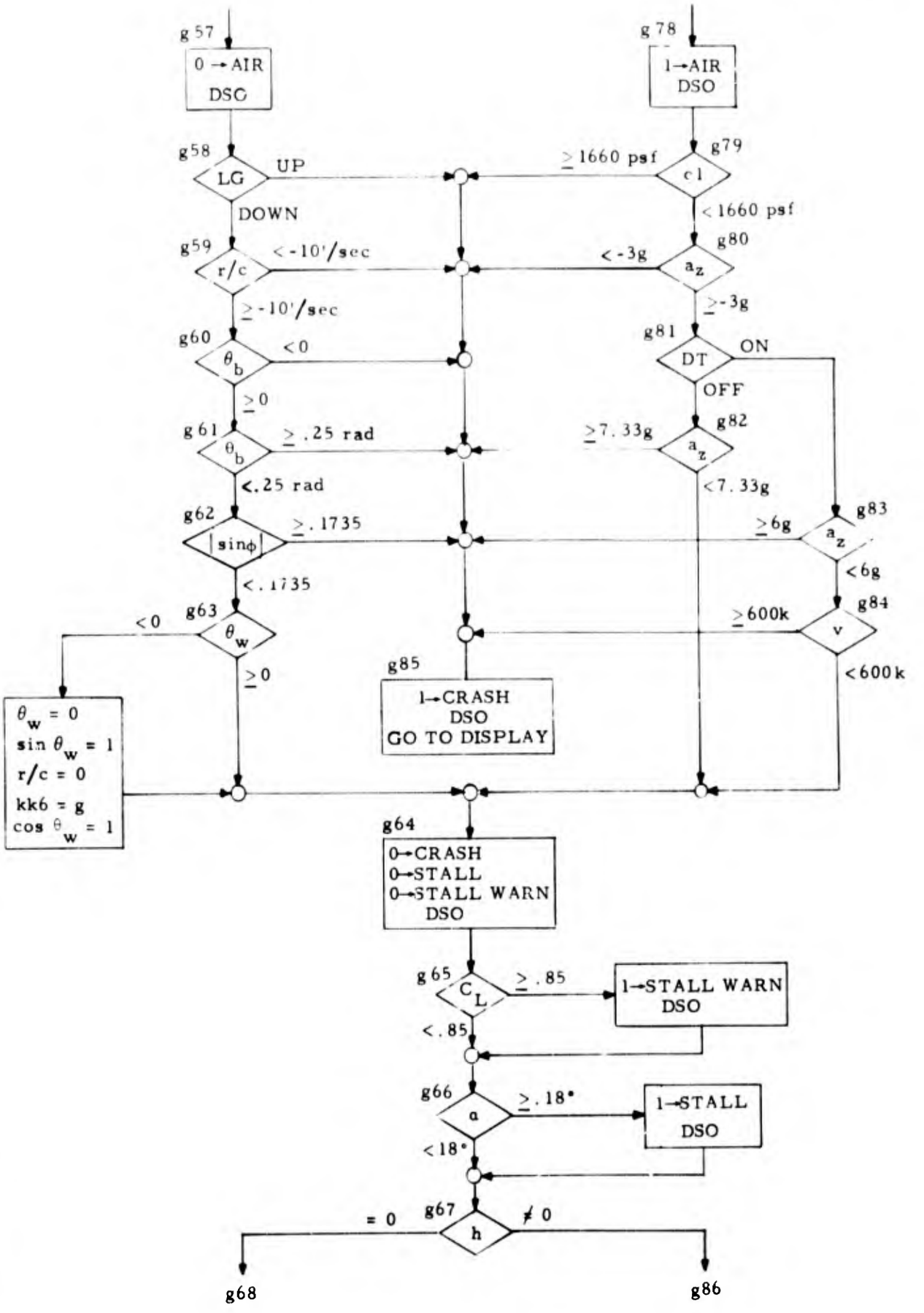


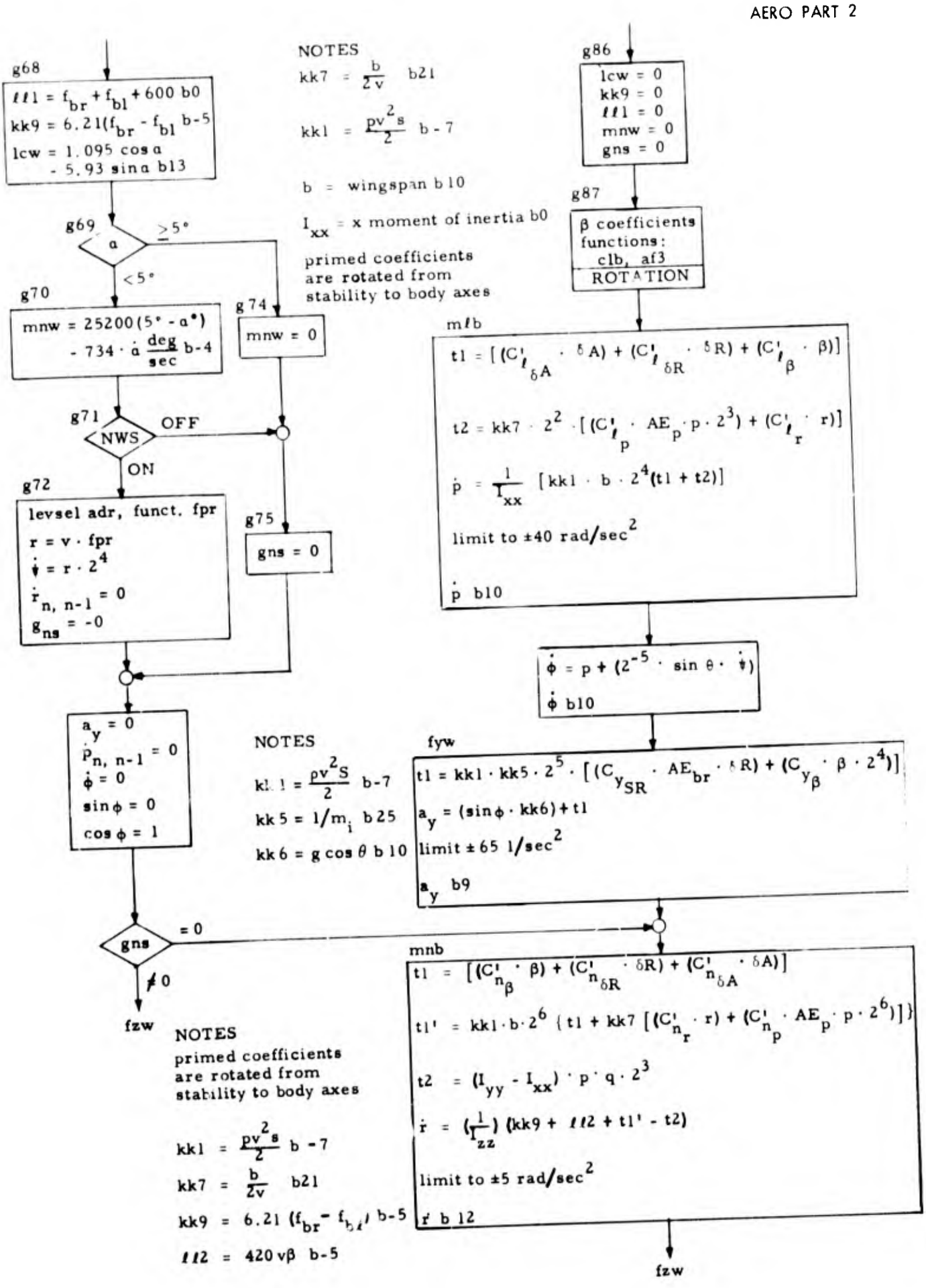

20h
Δp = 0.1 · hpℓ
45
Δs = -|δH_n - δH_n-1| - |δA_n - δA_n-1|
δH_n-1 = δH_n
δA_n-1 = δA_n
125
k1 = 0
k2 = 0
20
22h
hyd sys 2 fail ?
NO
23h
G2 = G2 + Δp
45
YES
50
k2 = Δp
k1 = 1/8 Δp
24h
hyd sys 1 fail ?
NO
emerg. system operating
NO
YES
YES
75
k1 = k1 + Δp
k2 = k2 + 1/8 Δp
25h
emerg. system operating
NO
YES
75
μℓ = 3000 psi
gl = gl + Δp - 5
100
μℓ = 3300 psi
gl = gl + Δp + 5
65
μℓ = 3000
gl = gl + Δp
(390)
29h


29h
$g2_n = g2_{n-1} + (k2 + 2.6)\,\Delta s$
g2:0
<
g2 = 0
≥
g2:600
≥
g2:3000
<
<
≥
YES
hyd sys 2 fail
g2 = 3000
NO
g2 = hpf
1 → SYS 2 FAIL DSO
0 → SYS 2 FAIL DSO
$g1_n = g1_{n-1} + (k1 + 2.6)\,\Delta s$
g1:0
<
g1 = 0
≥
g1:600
≥
g1:uf
<
<
≥
NO
hyd sys 1 fail
g1 = uf
YES
emerg. system operating
NO
YES
g1 = hpf
1 → SYS 1 FAIL DSO
0 → SYS 1 FAIL DSO


READOUT HYD SYS 2
YES
NO
hyd = g2
hyd = g1
q1:7
≤
>
emerg. sys selected
NO
YES
index timer
NO
timer done
YES
0 → emerg oper.
-100 → emerg operating
COMP. δA HINGE MOVEMENT
(δA HM > AVAIL HYD PRESS) (1→δA FROZEN DSO)
COMP. δH HINGE MOMENT
(δH HM > AVAIL HYD PRESS)→(1 → δH FROZEN DSO)
COMP. δR HINGE MOMENT
(END)


START
go8
kk5 = 1/m_i b25
go9
levsel q1, α, |α|
g 13
functions:
cmw cml clh
cmq cmh clp
aae
g23
p, r, δA, δR coefficients
functions:
lpi clr lda ldr
cnp cnr cna ndr
aae aea aer
ber
cyr
ROTATION af2
g45
Drop Tanks
functions:
cmt clt cdt
af1
g49
Yaw Damper
ℓℓ2 = 420 · v · β b-5
g50
Speed Brake
cmj cdj clj
g54
h
<25
≥25
g 55
GROUND EFFECT
cmg = .00375(h - 25) b16
clg = .00468(25 - h) C_L b12
g 77
cmg = 0
clg = 0
g56
h
≠0
= 0
g57
g78


g57
0 → AIR DSO
g58
LG
UP
DOWN
g59
r/c
< -10'/sec
≥ -10'/sec
g60
θb
< 0
≥ 0
g61
θb
≥ .25 rad
<.25 rad
g62
|sinφ|
≥ .1735
< .1735
g63
θw
< 0
≥ 0
θw = 0
sin θw = 1
r/c = 0
kk6 = g
cos θw = 1
g78
1→AIR DSO
g79
c l
≥ 1660 psf
< 1660 psf
g80
az
< -3g
≥ -3g
g81
DT
ON
OFF
g82
az
≥ 7.33g
< 7.33g
g83
az
≥ 6g
< 6g
g84
v
≥ 600k
< 600k
g85
1→CRASH DSO
GO TO DISPLAY
g64
0→CRASH
0→STALL
0→STALL WARN
DSO
g65
CL
≥ .85
< .85
1→STALL WARN DSO
g66
α
≥ .18°
< 18°
1→STALL DSO
g67
h
= 0
≠ 0
g68
g86


g68
$\ell\ell 1 = f_{br} + f_{bl} + 600$ b0
$kk9 = 6.21(f_{br} - f_{bl})$ b-5
lcw = 1.095 cos α − 5.93 sin α b13
g69
α
≥5°
<5°
g70
mnw = 25200(5° − α°) − 734 · α̇ deg/sec b-4
g71
NWS
OFF
ON
g72
levsel adr, funct. fpr
r = v · fpr
ψ̇ = r · 2^4
ṙ n, n-1 = 0
g ns = -0
g74
mnw = 0
g75
gns = 0
a y = 0
Ṗ n, n-1 = 0
φ̇ = 0
sin φ = 0
cos φ = 1
gns
= 0
≠ 0
fzw
NOTES
kk7 = b/2v b21
kk1 = ρv²s/2 b-7
b = wingspan b10
I xx = x moment of inertia b0
primed coefficients are rotated from stability to body axes
NOTES
kk1 = ρv²S/2 b-7
kk5 = 1/m i b25
kk6 = g cos θ b10
NOTES
primed coefficients are rotated from stability to body axes
kk1 = ρv²s/2 b-7
kk7 = b/2v b21
kk9 = 6.21 (f br − f bl) b-5
ℓℓ2 = 420 vβ b-5
g86
lcw = 0
kk9 = 0
ℓℓ1 = 0
mnw = 0
gns = 0
g87
β coefficients functions: clb, af3
ROTATION
mlb
t1 = [(C'ℓδA · δA) + (C'ℓδR · δR) + (C'ℓβ · β)]
t2 = kk7 · 2² · [(C'ℓp · AEp · p · 2³) + (C'ℓr · r)]
ṗ = (1/I xx) [kk1 · b · 2⁴(t1 + t2)]
limit to ±40 rad/sec²
ṗ b10
φ̇ = p + (2⁻⁵ · sin θ · ψ̇)
φ̇ b10
fyw
t1 = kk1 · kk5 · 2⁵ · [(C ySR · AEbr · δR) + (C yβ · β · 2⁴)]
a y = (sin φ · kk6) + t1
limit ± 65 1/sec²
a y b9
mnb
t1 = [(C'nβ · β) + (C'nδR · δR) + (C'nδA · δA)]
t1' = kk1 · b · 2⁶ {t1 + kk7 [(C'nr · r) + (C'np · AEp · p · 2⁶)]}
t2 = (I yy − I xx) · p · q · 2³
ṙ = (1/I zz) (kk9 + ℓℓ2 + t1' − t2)
limit to ±5 rad/sec²
ṙ b12
fzw

NOTES

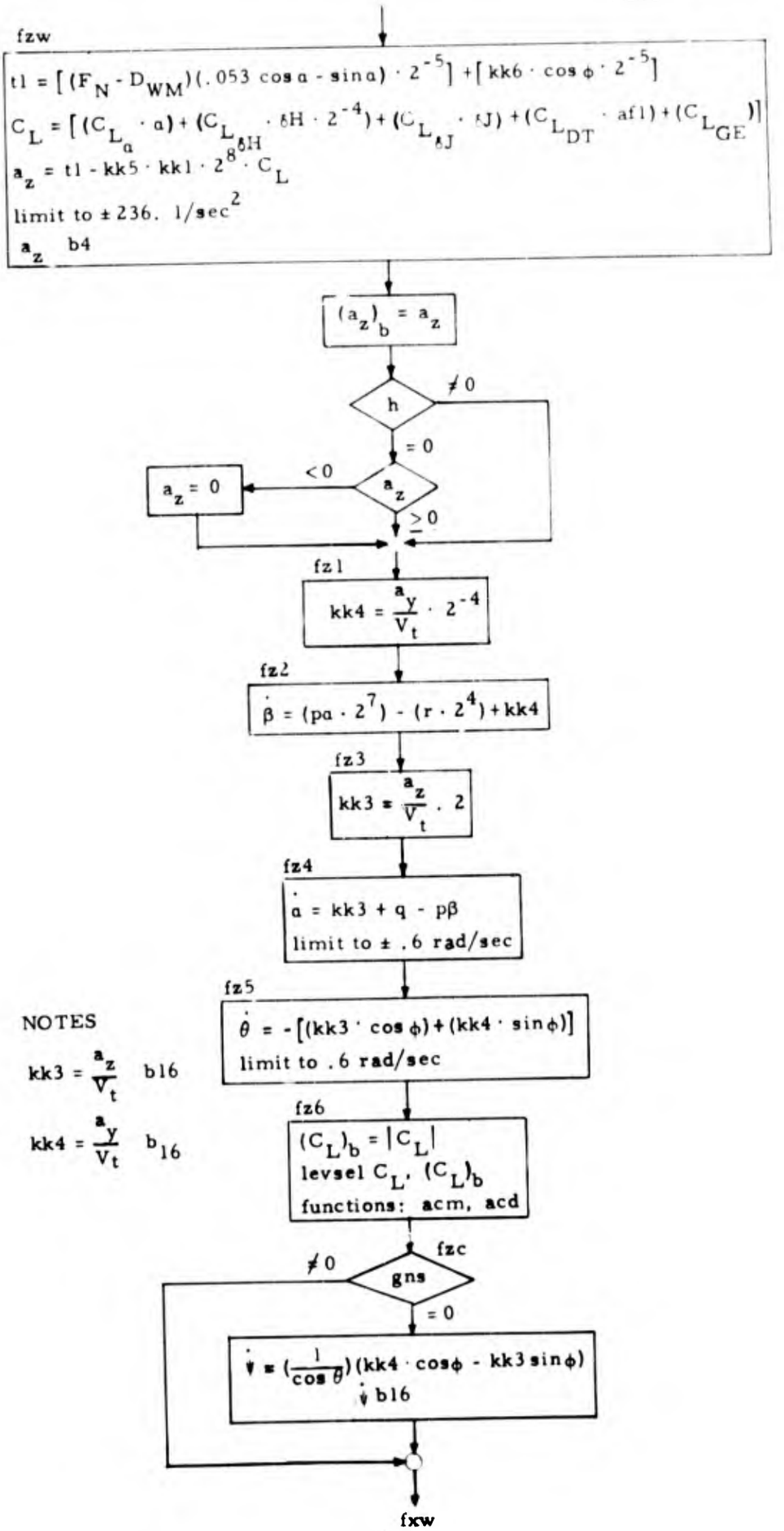
$kk6 = g \cos\theta$ b10

$kk5 = 1/m_i$ b25

$kk1 = \frac{\rho V^2 S}{a^2}$ b-7

$kk4 = \frac{a_y}{V_t}$ b16

$kk3 = \frac{a_z}{V_t}$ b16



NOTES

$kk3 = \frac{a_z}{V_t}$ b16

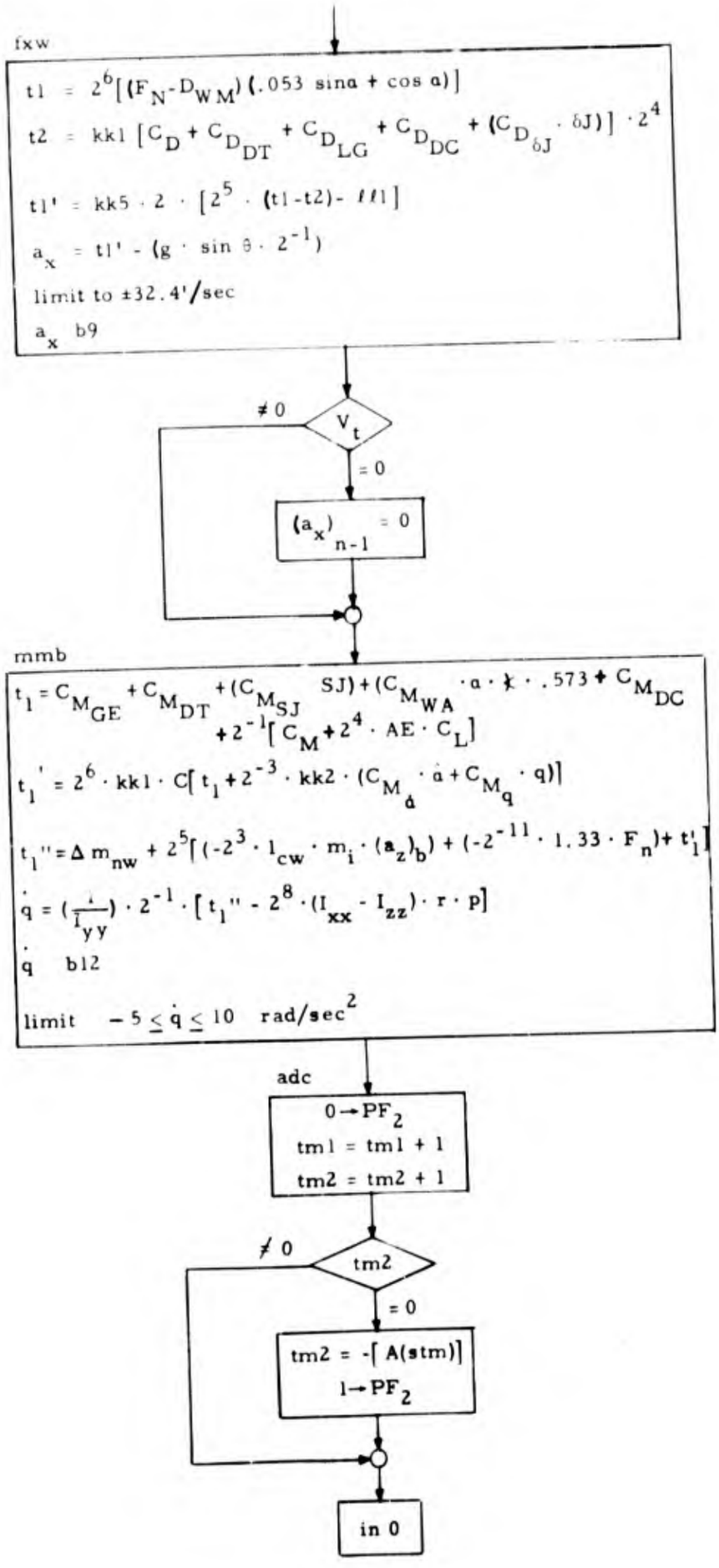
$kk4 = \frac{a_y}{V_t}$ b16

NOTES

$kk1 = \frac{pv^2 s}{2}$ b-7

$kk5 = 1/mi$ b25

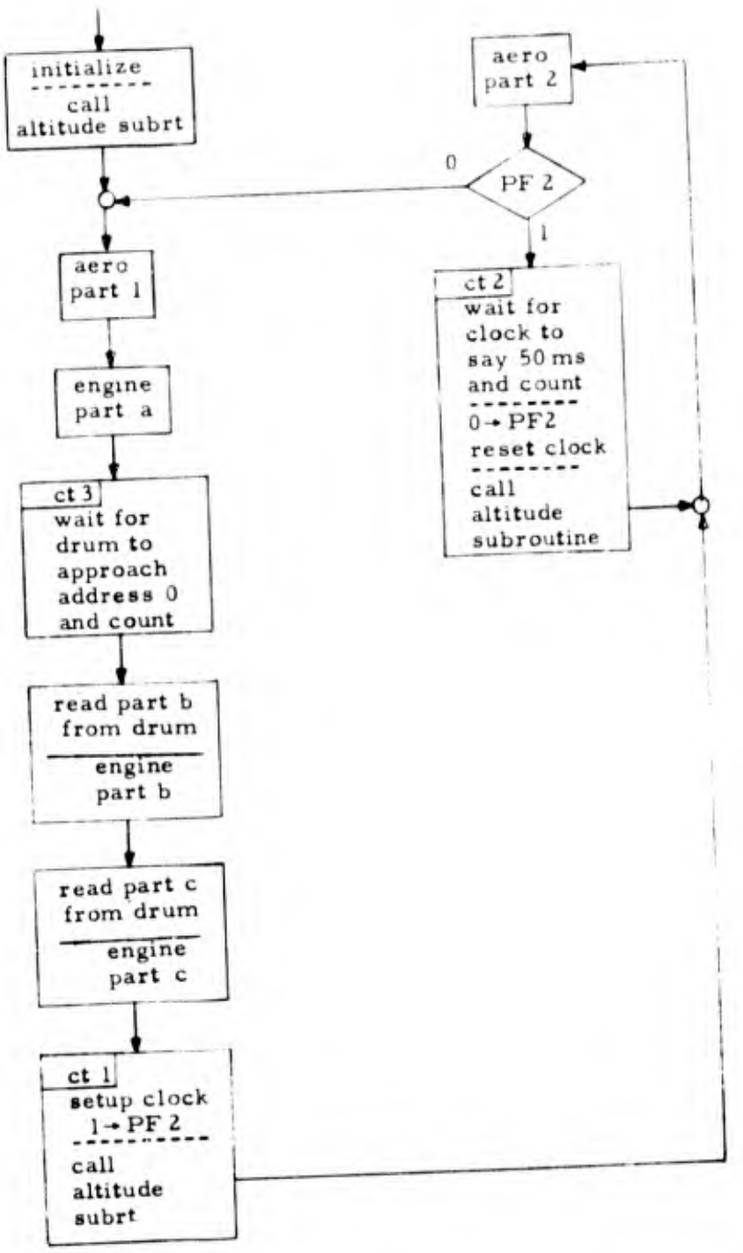
$lll = f_{br} + f_{bl} + 600$ b0

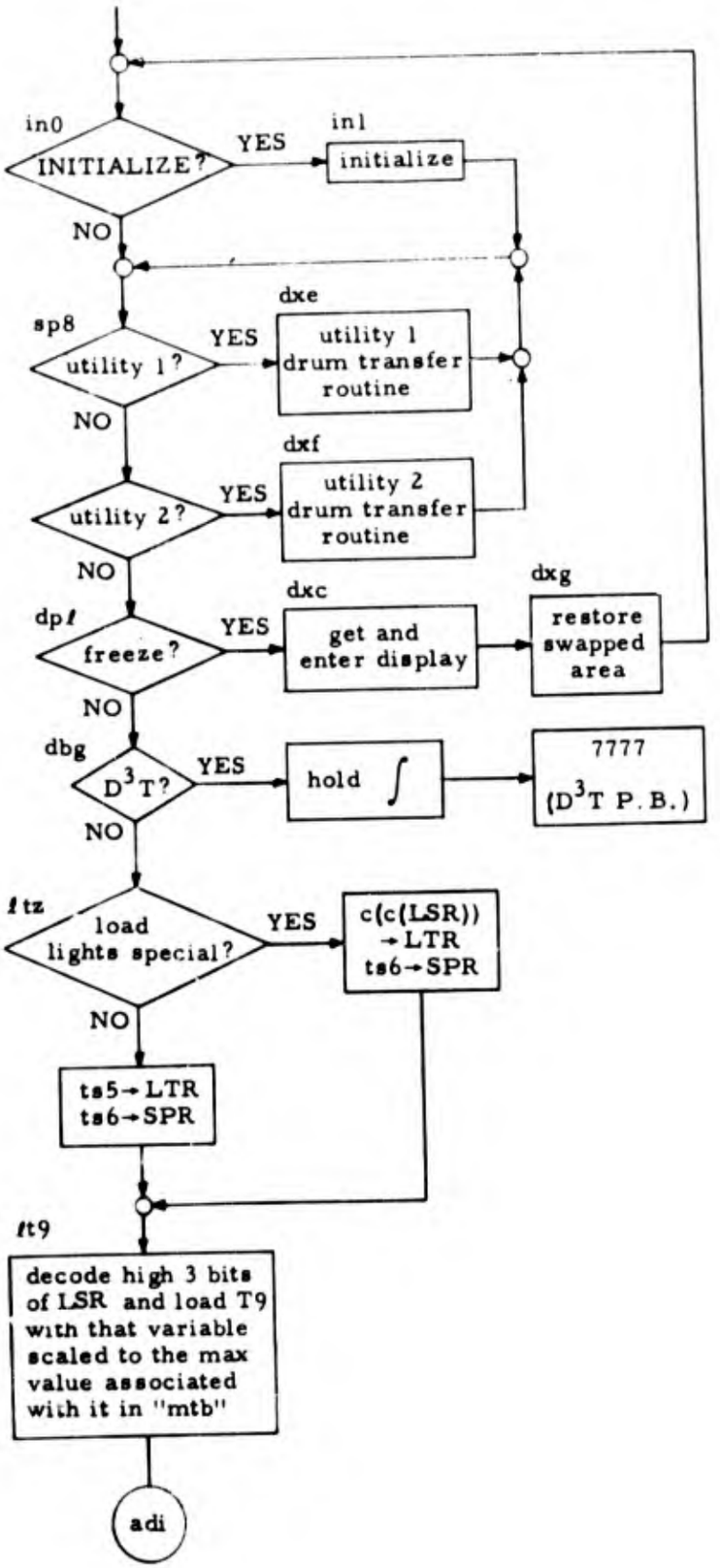


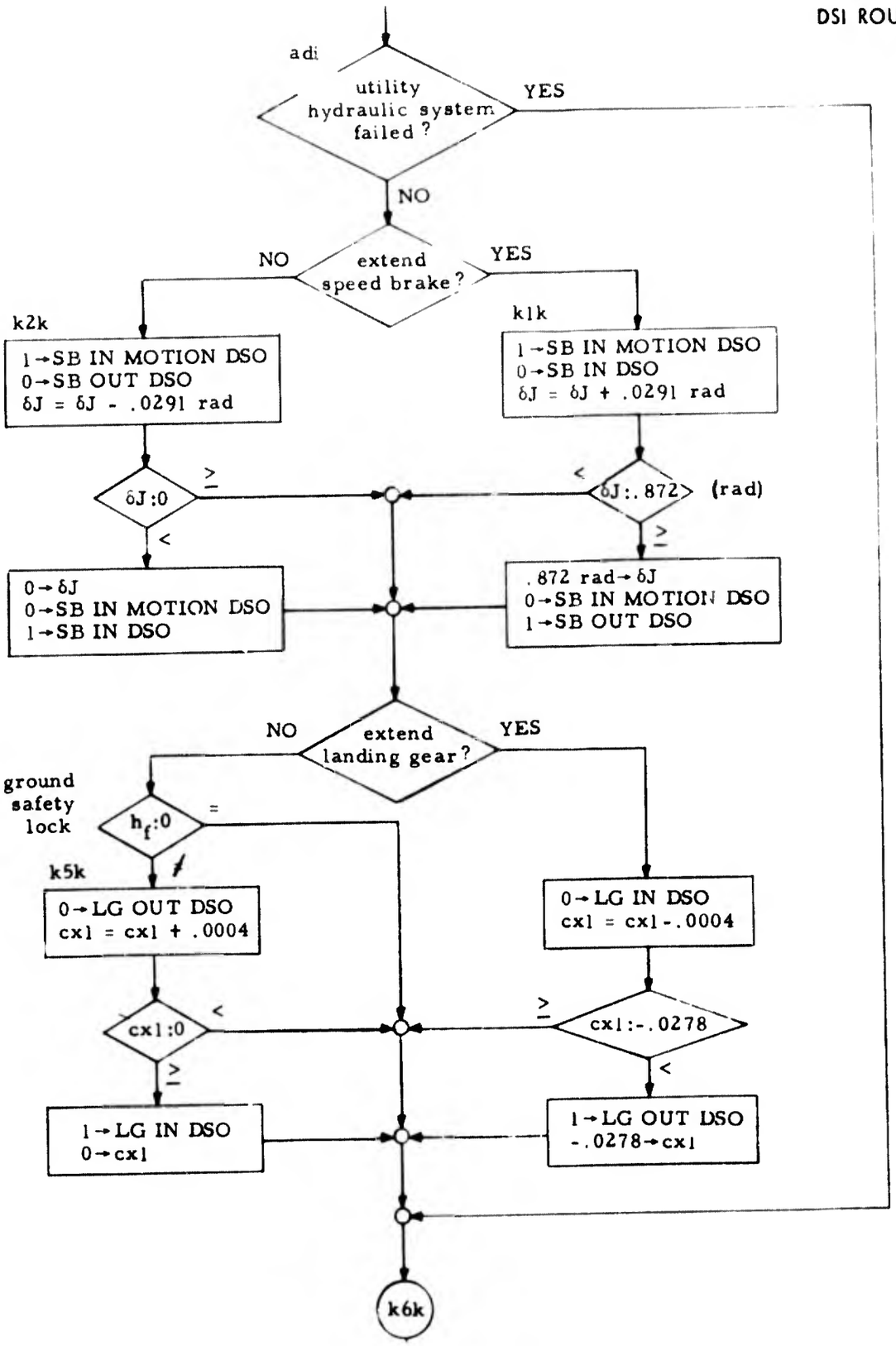
APPENDIX C

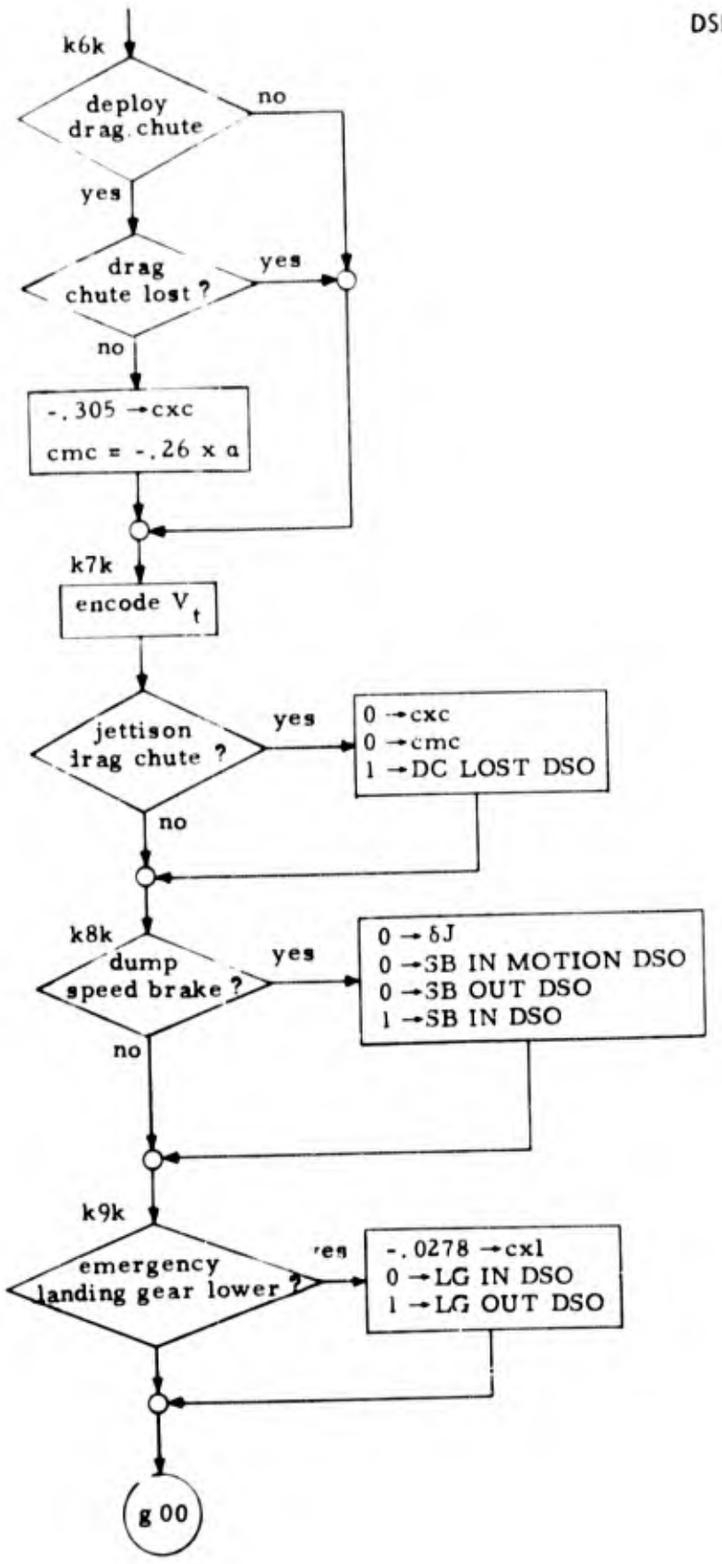
HYBRID FLOW CHARTS

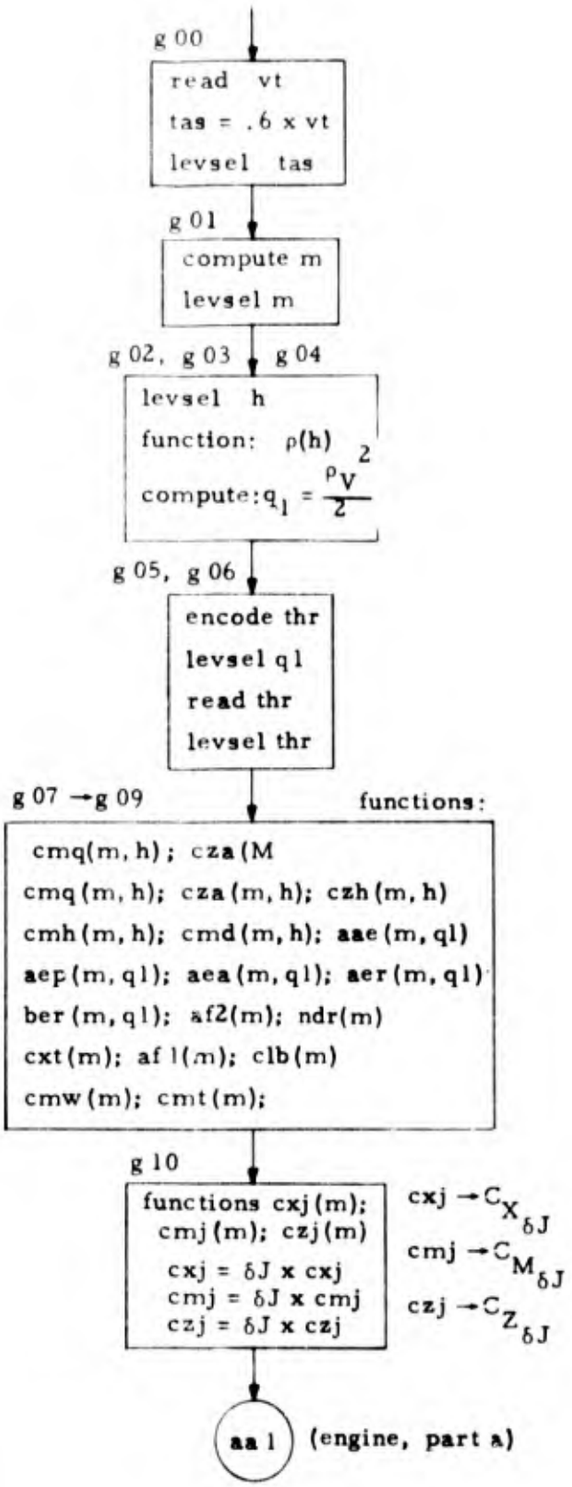
GROSS FLOW OF HYBRID F100




in0
INITIALIZE?
YES
in1
initialize
NO
dxe
sp8
utility 1?
YES
utility 1 drum transfer routine
NO
dxf
utility 2?
YES
utility 2 drum transfer routine
NO
dxc
dxg
dpℓ
freeze?
YES
get and enter display
restore swapped area
NO
dbg
D3T?
YES
hold ∫
7777
(D3T P.B.)
NO
ℓtz
load lights special?
YES
c(c(LSR)) → LTR
ts6 → SPR
NO
ts5 → LTR
ts6 → SPR
ℓt9
decode high 3 bits of LSR and load T9 with that variable scaled to the max value associated with it in "mtb"
adi


adi
utility hydraulic system failed ?
YES
NO
extend speed brake ?
NO
YES
k2k
1→SB IN MOTION DSO
0→SB OUT DSO
δJ = δJ - .0291 rad
k1k
1→SB IN MOTION DSO
0→SB IN DSO
δJ = δJ + .0291 rad
δJ:0
≥
<
δJ:.872
(rad)
<
≥
0→δJ
0→SB IN MOTION DSO
1→SB IN DSO
.872 rad→δJ
0→SB IN MOTION DSO
1→SB OUT DSO
NO
extend landing gear ?
YES
ground safety lock
h_f:0
=
≠
k5k
0→LG OUT DSO
cx1 = cx1 + .0004
0→LG IN DSO
cx1 = cx1 - .0004
cx1:0
<
≥
≥
cx1:-.0278
<
1→LG IN DSO
0→cx1
1→LG OUT DSO
-.0278→cx1
k6k


k6k
deploy drag chute
no
yes
drag chute lost ?
yes
no
-.305 → cxc
cmc = -.26 x α
k7k
encode V_t
jettison drag chute ?
yes
0 → cxc
0 → cmc
1 → DC LOST DSO
no
k8k
dump speed brake ?
yes
0 → δJ
0 → SB IN MOTION DSO
0 → SB OUT DSO
1 → SB IN DSO
no
k9k
emergency landing gear lower ?
yes
-.0278 → cxl
0 → LG IN DSO
1 → LG OUT DSO
g 00



tas → V_t (knots)

vt → vt → V_t ('/sec)

m → Mach

$$tt = \begin{cases} a_0\left[.8625 + .1375\left(1 - \frac{h}{35,400}\right)\right] & \text{if } h \leq 35,400 \\ .8625\ a_0 & \text{if } h > 35,400 \end{cases}$$

$$m = \frac{tas}{tt}$$

q 1 → q, dynamic pressure

thr → ST

cmq → C_{m_q}

cza → C_{z_α}

czh → $C_{z_{\delta H}}$

cmh → $C_{m_{\delta H}}$

cmd → $C_{m_{\dot{\alpha}}}$

aae → AE

aep → AE_p

aea → AE_a

aer → AE_{ar}

ber → AE_{br}

ndr → $C_{n_{\delta R}}$

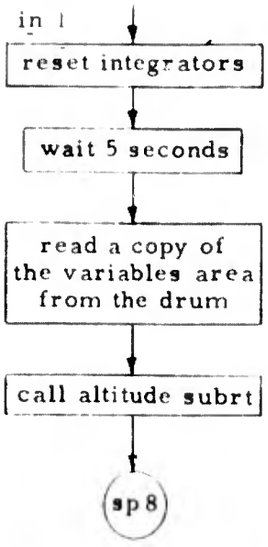
cxt → $C_{x_{DT}}$

clb → C_{l_β}

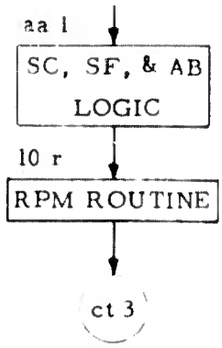
cmw → $C_{m_{WA}}$

cmt → $C_{m_{DT}}$

INITIALIZE

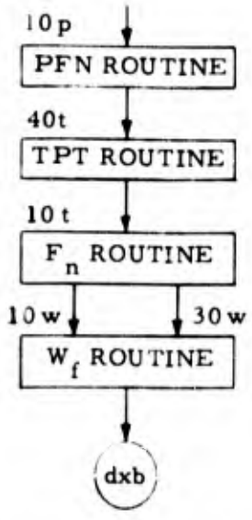


ENGINE PART A



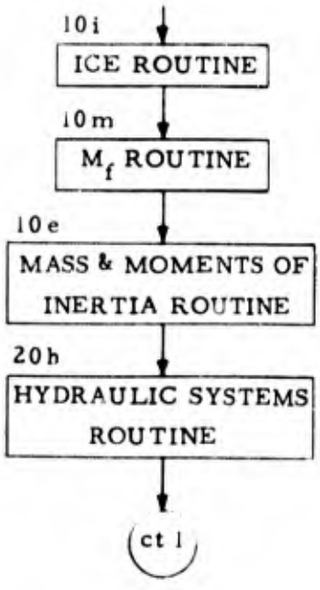
[CONTROL TIMING]

ENGINE PART B

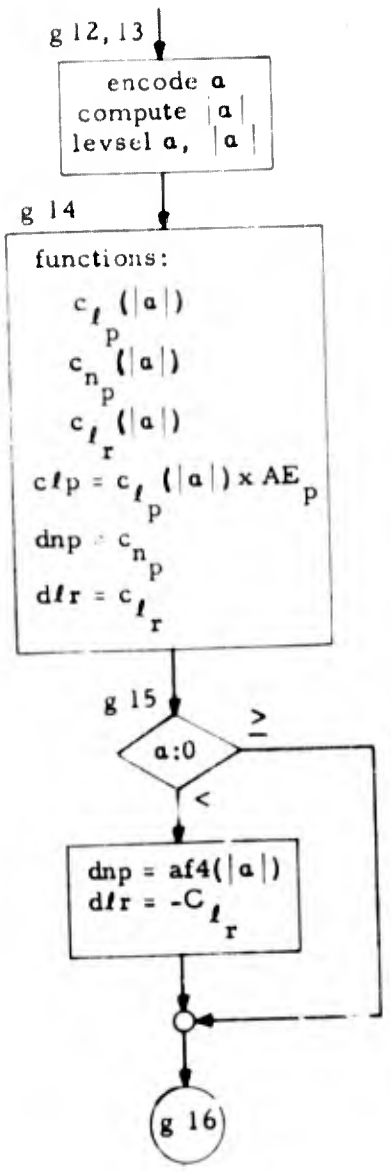


[DRUM READ ROUTINE]

ENGINE PART C



[CONTROL TIMING]



$b\ell p \rightarrow |\alpha|$

$c\ell p \rightarrow c_{\ell_p}$

$cnp \rightarrow c_{n_p}$

$c\ell r \rightarrow c_{\ell_r}$

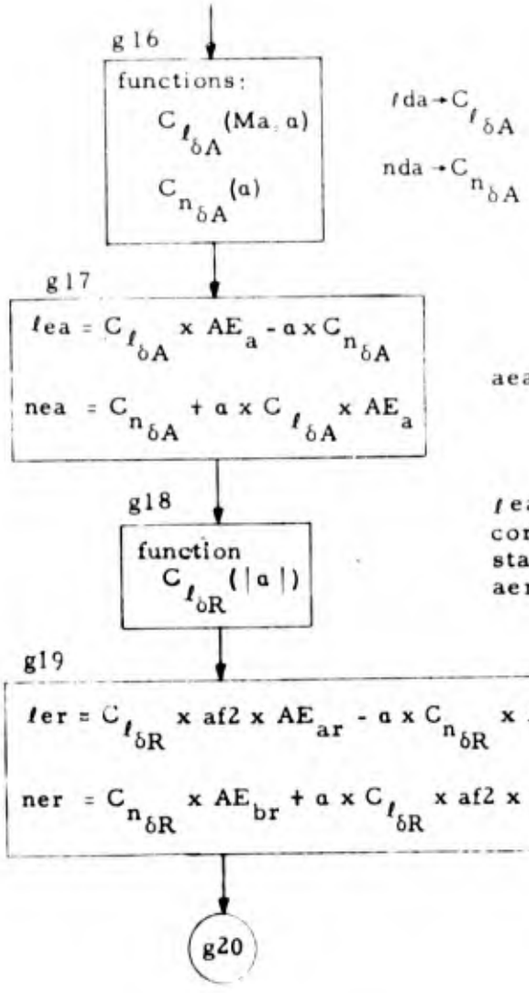
$$aep \rightarrow AE_p \rightarrow \frac{\left(\frac{\partial C_\ell}{\partial \frac{pb}{2v}}\right) F}{\left(\frac{\partial C_\ell}{\partial \frac{pb}{2v}}\right) R}$$

C_{n_r} is assumed constant and is handled in the analog portion

af4(|α|) is the C_{n_p} for $\alpha < 0$

C_{ℓ_r} is assumed symmetric about $\alpha = 0$

dnp and dnr are corrected values of C_{n_p} and C_{n_r} respectively



$\ell da \rightarrow C_{\ell_{\delta A}}$

$nda \rightarrow C_{n_{\delta A}}$

$$aea \rightarrow AE_a \rightarrow \frac{(\frac{\partial C_\ell}{\partial_{\delta A}})_F}{(\frac{\partial C_\ell}{\partial_{\delta A}})_R}$$

ℓea and nea are ℓda and nda corrected for rotation from stability to body axes and for aeroelastic effects

$\ell dr \rightarrow C_{\ell_{\delta R}}(|\alpha|)$

$ndr \rightarrow C_{n_{\delta R}}(Ma)$

ℓer and ner are ℓdr and ndr corrected for rotation and AE af2(Ma) is a correction on $C_{\ell_{\delta R}}(|\alpha|)$

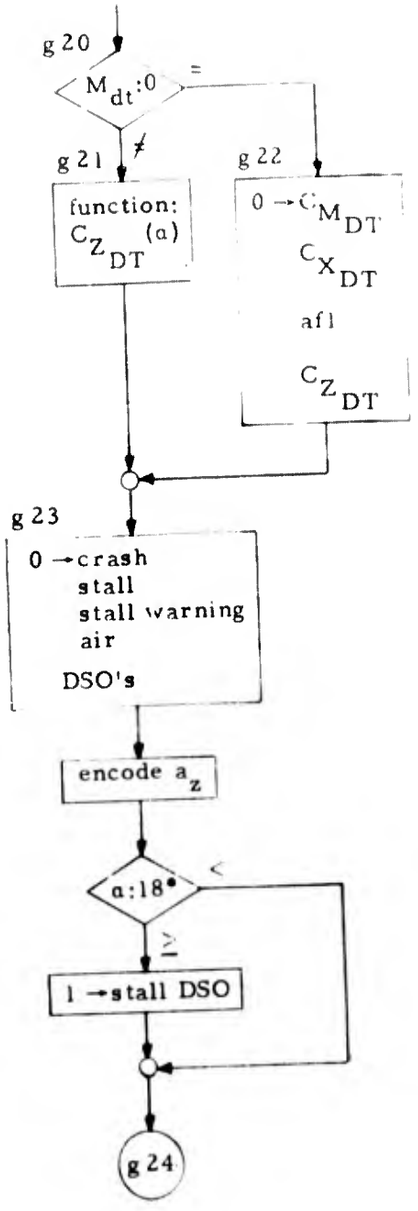
rotation computations assume:

$\sin \alpha \approx \alpha$

$\cos \alpha \approx 1$

$$aer \rightarrow AE_{ar} \rightarrow \frac{(\frac{\partial C_\ell}{\partial_{\delta R}})_F}{(\frac{\partial C_\ell}{\partial_{\delta R}})_R}$$

$$ber \rightarrow AE_{br} \rightarrow \frac{(\frac{\partial C_n}{\partial_{\delta R}})_F}{(\frac{\partial C_n}{\partial_{\delta R}})_R}$$



mdt → M_{DT}

czt → $C_{Z_{DT}}(\alpha)$

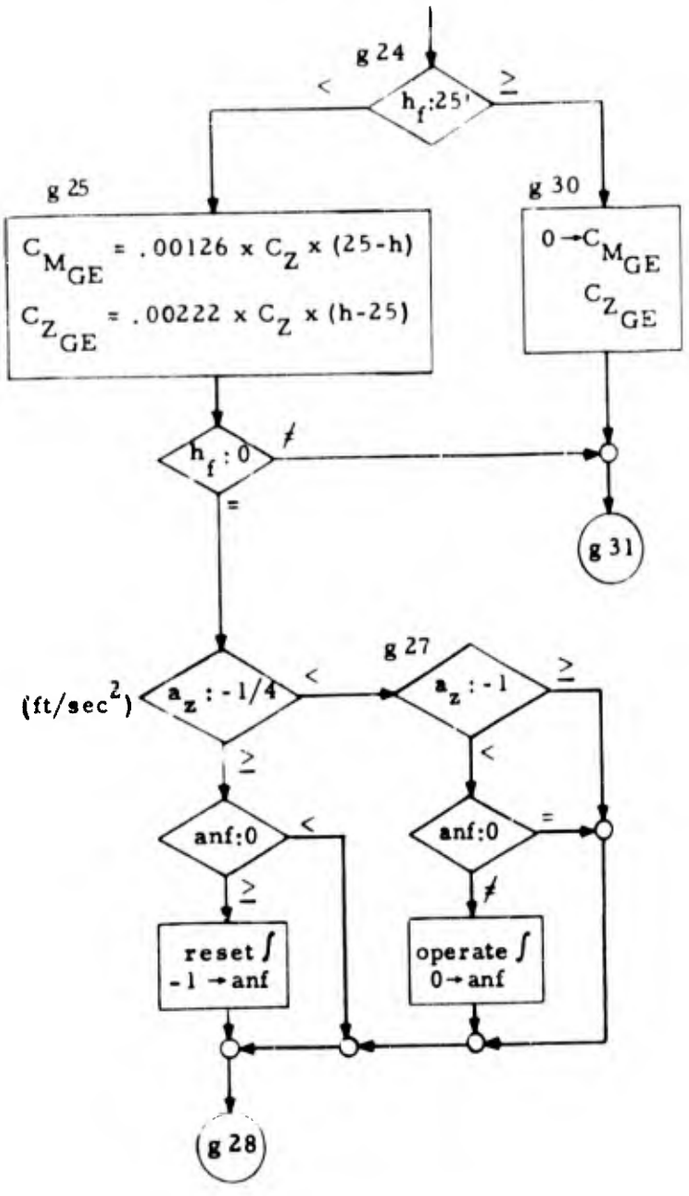
cmt → $C_{M_{DT}}(Ma)$

cxt → $C_{X_{DT}}(Ma)$

afl → afl(Ma)

DSO → discrete output

az → a_z

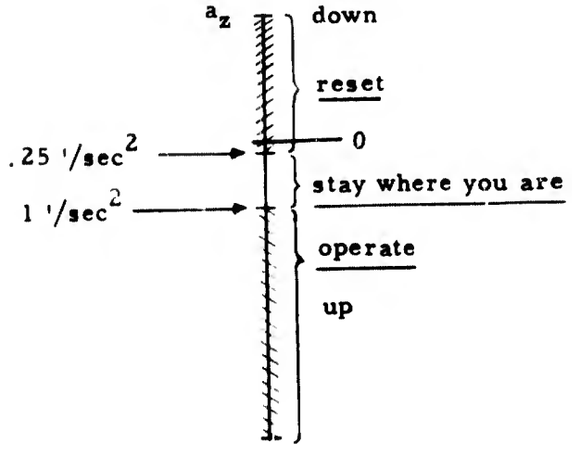


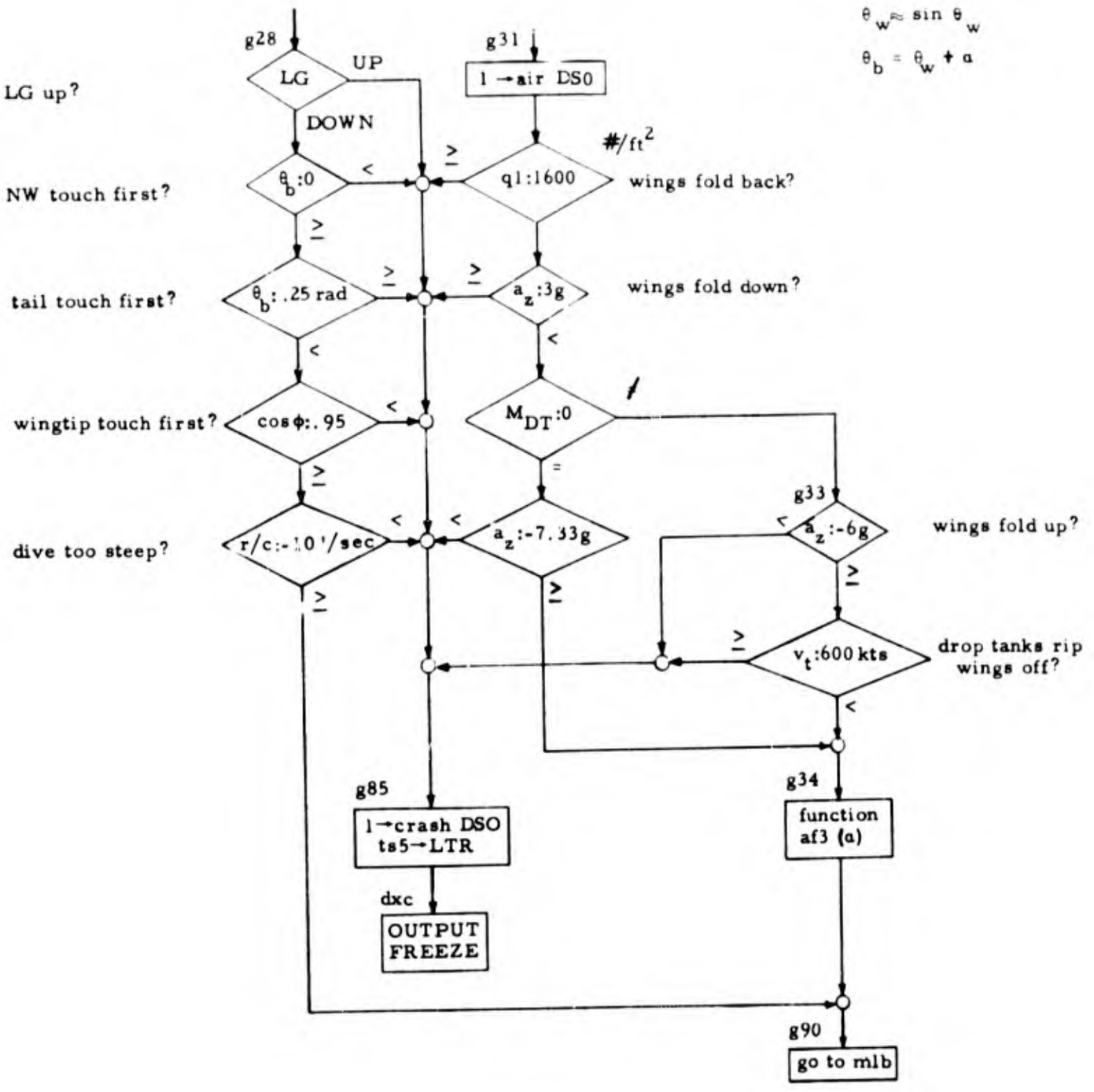
pbl → h_f → fine altitude
(scaled b 6)

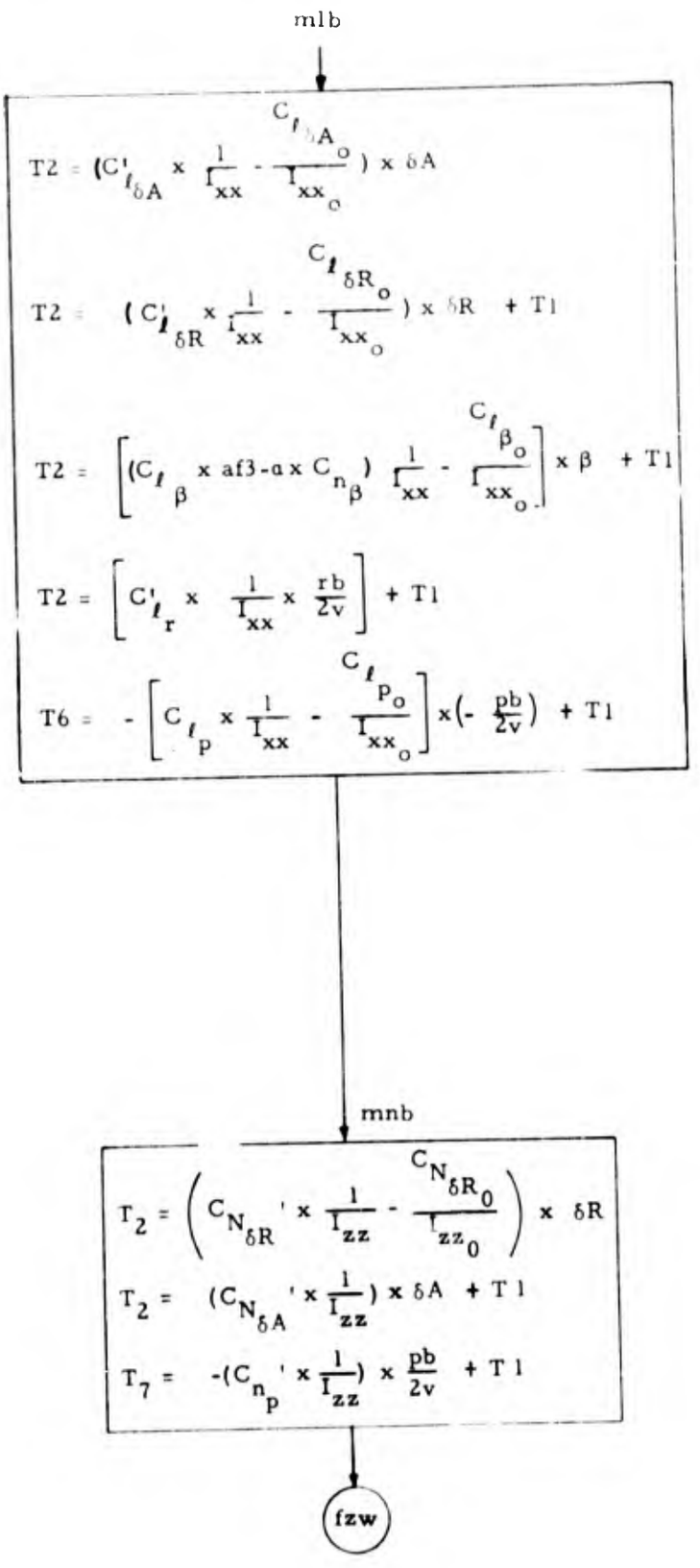
anf: flag register indicating the state of the integrators:

-1 → reset
0 → operate
1 → hold

this set of tests does the landing and take off clamping and unclamping




$\theta_w \approx \sin \theta_w$
$\theta_b = \theta_w + \alpha$
g28
LG
UP
DOWN
LG up?
g31
1 → air DS0
#/ft²
θ_b:0
NW touch first?
q1:1600
wings fold back?
θ_b:.25 rad
tail touch first?
a_z:3g
wings fold down?
cosϕ:.95
wingtip touch first?
M_{DT}:0
g33
r/c:-10°/sec
dive too steep?
a_z:-7.33g
a_z:-6g
wings fold up?
v_t:600 kts
drop tanks rip wings off?
g85
1 → crash DSO ts5 → LTR
dxc
OUTPUT FREEZE
g34
function af3 (α)
g90
go to mlb



primed coefficients are corrected for AE and rotation

lea → $C'_{\ell_{\delta A}}$

kk6 → $1/I_{xx}$

lza → $\frac{C_{\ell_{\delta A_o}}}{I_{xx_o}}$

ler → $C'_{\ell_{\delta R}}$

lzr → $\frac{C_{\ell_{\delta R_o}}}{I_{xx_o}}$

cnb → C_{n_β}

cℓb → C_{ℓ_β}

af3(a) is correction on C_{ℓ_β}(Ma)

zℓb → $\frac{C_{\ell_{\beta_o}}}{I_{xx_o}}$

dℓr → C'_{ℓ_r}

cℓp → C'_{ℓ_p}

zℓp → $\frac{C_{\ell_{p_o}}}{I_{xx_o}}$

primed coefficients are corrected for AE and rotation

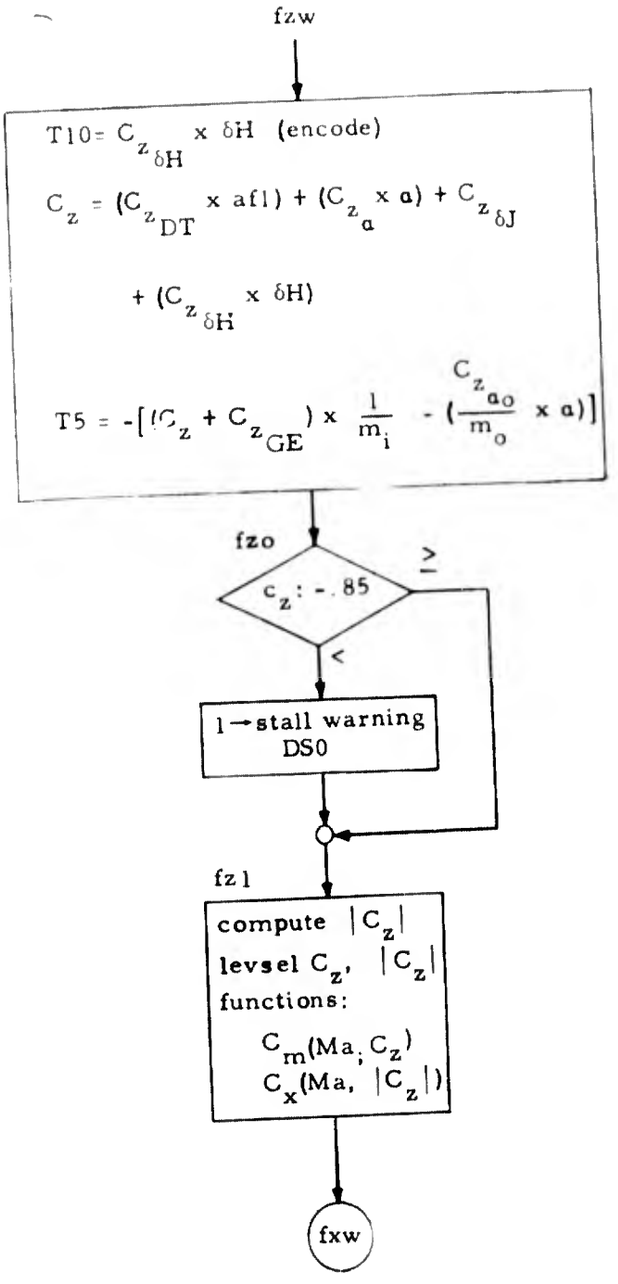
ner → $C_{N_{\delta R}}$

kk7 → $1/I_{zz}$

nzr → $\frac{C_{N_{\delta R_0}}}{I_{zz_0}}$

nea → $C_{N_{\delta A}}$

dnp → C_{n_p}



czh → $C_{z_{\delta H}}$

czt → $C_{z_{DT}}$ (a)

afl(Ma) is a correction on $C_{z_{DT}}$

czj → $C_{z_{\delta J}} \times \delta J$

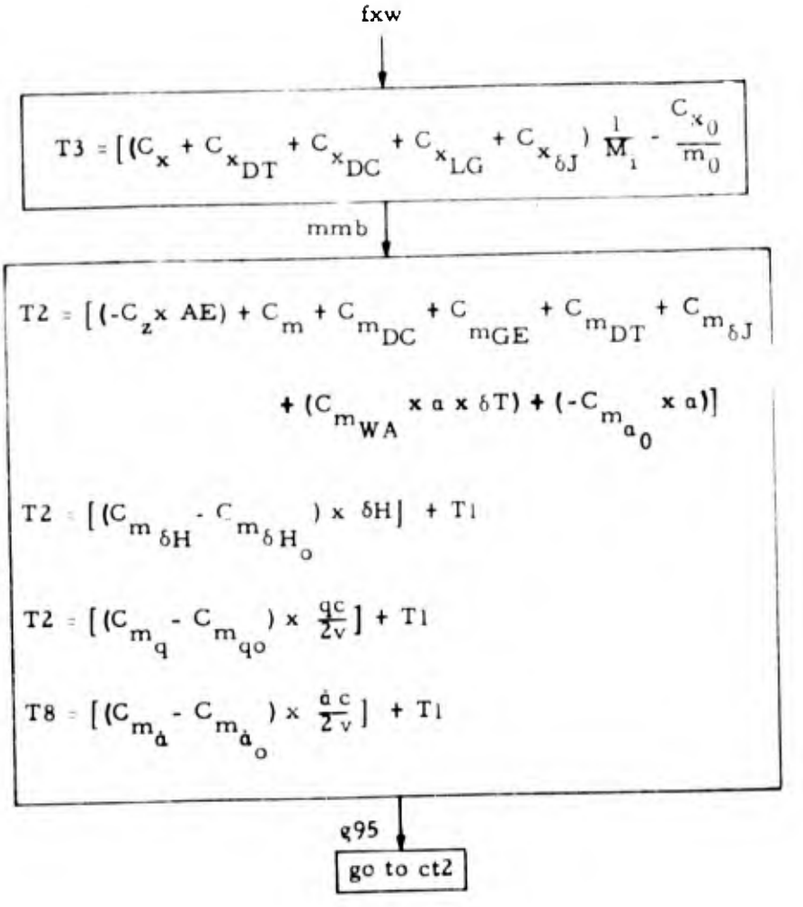
czg → $C_{z_{GE}}$

zza → $\frac{C_{z_{a_o}}}{m_o}$

cz → C_z

cm → C_m

cx → C_x



cx→C_x

cxt→$C_{x_{DT}}$

cxc→$C_{x_{DC}}$

cxl→$C_{x_{LG}}$

cxj→$C_{x_{\delta J}} \times \delta J$

zx→C_{x_o}/m_o

aae→AE→$\frac{\Delta C_m}{\Delta C_z}$

cm→C_m

cmc→$C_{m_{DC}}$

cmg→$C_{m_{GE}}$

cmt→$C_{m_{DT}}$

cmj→$C_{m_{\delta J}} \times \delta J$

cmw→$C_{m_{WA}}$

alp→α

thr→δT

zmα→$-C_{m_{\alpha_o}}$

cmh→$C_{m_{\delta H}}$

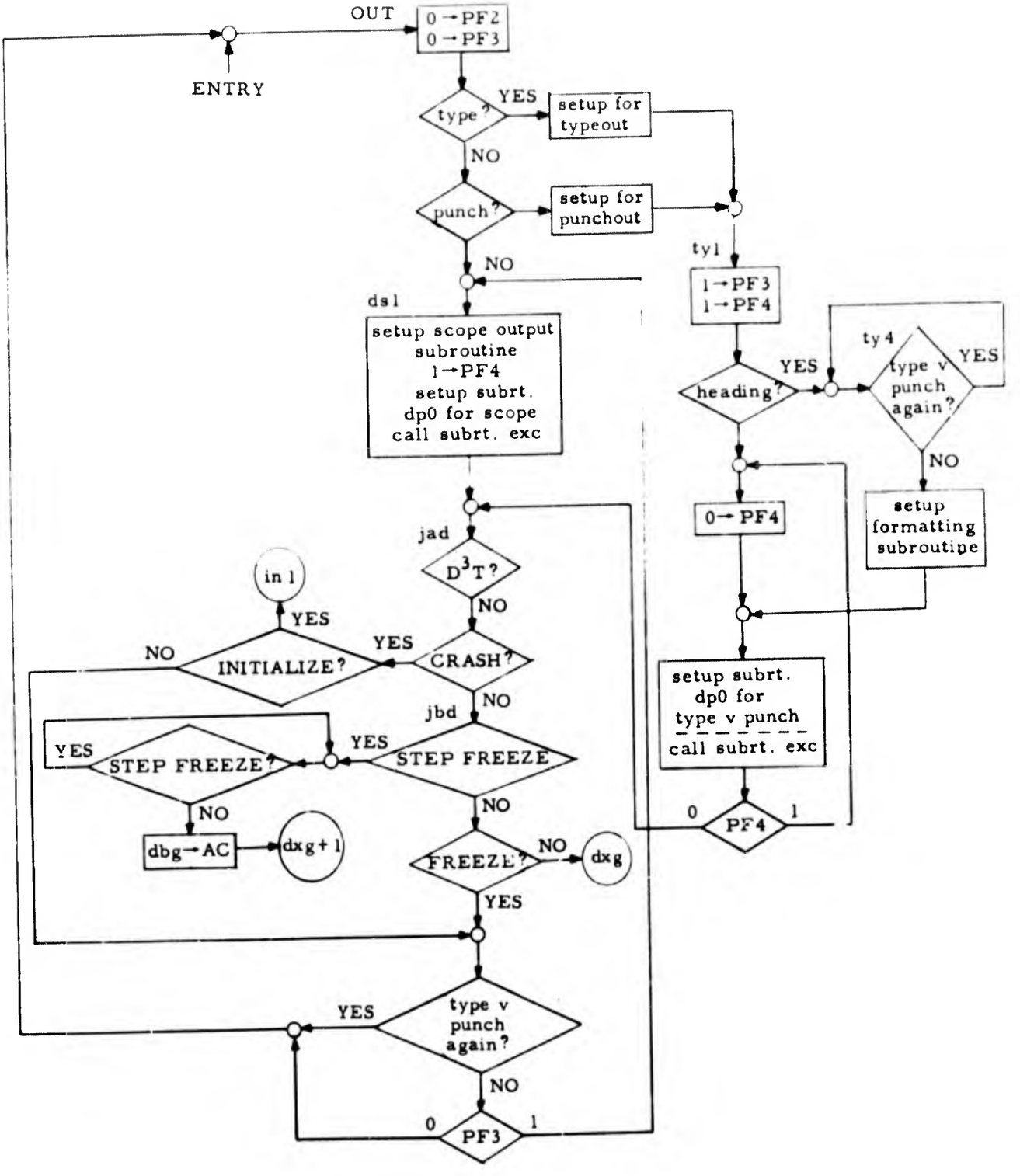
zmh→$C_{m_{\delta H_o}}$

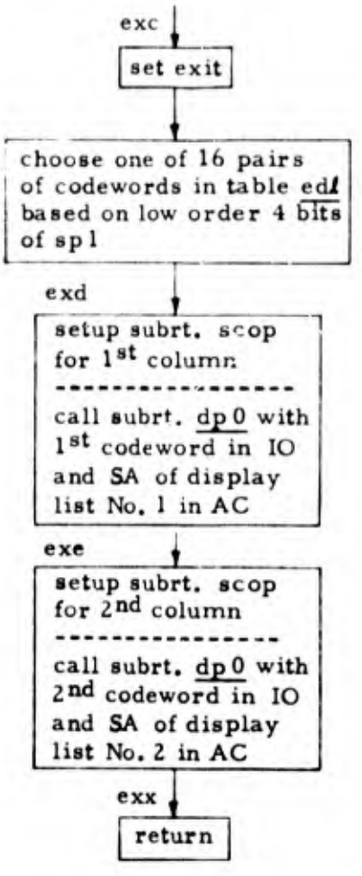
cmq→C_{m_q}

zmq→$C_{m_{q_o}}$

cmd→$C_{m_{\dot{\alpha}}}$

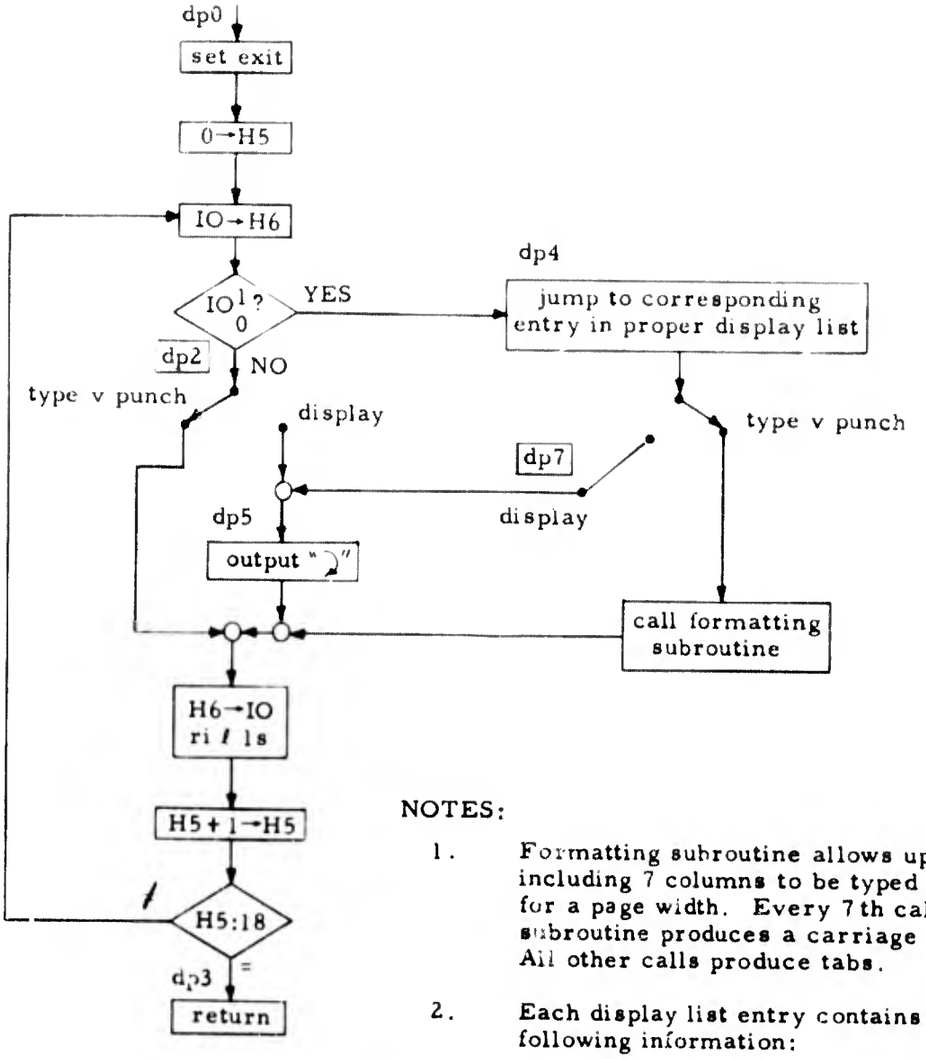
zmd→$C_{m_{\dot{\alpha}_o}}$


OUT
0 → PF2
0 → PF3
ENTRY
type?
YES
setup for typeout
NO
punch?
setup for punchout
NO
tyl
1 → PF3
1 → PF4
dsl
setup scope output subroutine
1 → PF4
setup subrt. dp0 for scope
call subrt. exc
ty4
heading?
YES
type v punch again?
YES
NO
0 → PF4
setup formatting subroutine
jad
D3T?
NO
in 1
YES
INITIALIZE?
NO
YES
CRASH?
NO
jbd
setup subrt. dp0 for type v punch
call subrt. exc
YES
STEP FREEZE?
YES
STEP FREEZE
NO
NO
dbg → AC
dxg + 1
FREEZE?
NO
dxg
YES
0
PF4
1
type v punch again?
YES
NO
0
PF3
1



note:

a 1 in a codeword bit position selects the corresponding display list entry for output



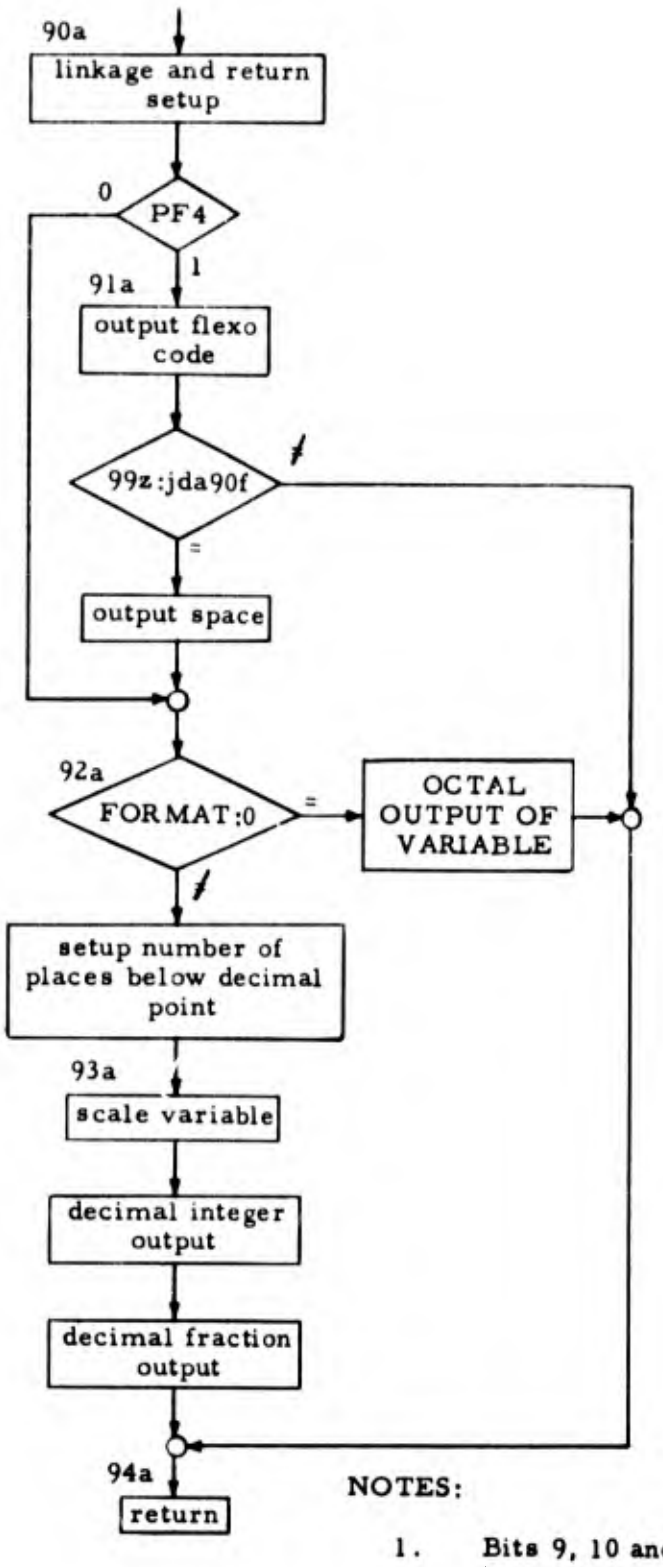
NOTES:

1. Formatting subroutine allows up to and including 7 columns to be typed or punched for a page width. Every 7th call to the subroutine produces a carriage return. All other calls produce tabs.

2. Each display list entry contains the following information:

   a) variable
   b) scaling
   c) number of places after decimal point
   d) flexo code for variable name

   A jump to a display list entry causes the variable to be loaded into the AC and the main output subrt. (90a) to be called using the "cal" instruction.



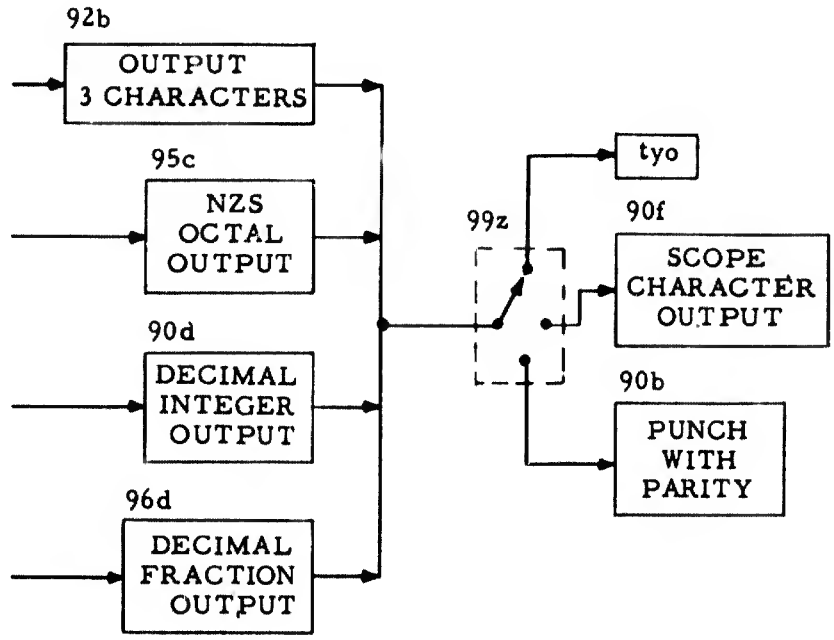
NOTES:

1. Bits 9, 10 and 11 of the cal instruction contain the format code "n"

   (n=0)→non zero sup. octal output

   (n≠0)→decimal output with n-1 places below decimal point

2. Bits 12-17 of the cal instruction indicate scaling from b0 (pure integer) to b17 (pure fraction) or further. No negative b-numbers are allowed.



note: maximum subroutine depth attained is six