AN ABSTRACT OF THE THESIS OF

Roderick Smit Mesecar for the Electrical Engineer

(Name) (Degree)

Date thesis is presented May 8, 1964

Title THE GALAXY COMPUTER LOGIC CONTROL SYSTEM

Abstract approved / (Major Professor)

The logic control system of a computer automatically sequences logic operations which direct how and when memory stored instructions and data are processed in the digital registers.

This thesis contains circuit schematics and test results of a nanosecond response logic-control system for the Galaxy Computer. The circuit designs have made possible level-logic operation rates over a million per second.

The use of a voltage level-shift scheme with low capacitance and complementary emitter-follower current-amplifier accounts for most of the speed improvement. The circuits are designed to be race-free and maintain the same rise and fall time whether operating at low or high frequency. Ability to switch large signal currents improves the fan-out capabilities of the logic networks, thereby increasing the number of functions that can be performed on a particular logic step. Interlocking control in a single logic-control system allows several systems to be coupled together for concurrent operation.
A scheme like the logic-control system can also be used wherever there is a need for a fast, versatile preprogrammed electronic switch.
THE GALAXY COMPUTER CONTROL SYSTEM

by

RODERICK SMIT MESECAR

A THESIS

submitted to

OREGON STATE UNIVERSITY

in partial fulfillment of
the requirements for the
degree of

ELECTRICAL ENGINEER

June 1964
APPROVED:

Professor of Electrical Engineering

Head of Department of Electrical Engineering

Chairman of School Graduate Committee

Dean of Graduate School

Date thesis is presented May 8, 1964

Typed by Jane Sim
ACKNOWLEDGMENTS

The work in this thesis was part of the Galaxy Computer development program done cooperatively by the Electrical Engineering and Mathematics Departments at Oregon State University. The engineering program management was under the direction of Professor Louis N. Stone. The author wishes to thank those members of the Galaxy staff who contributed their comments to the design of the Galaxy Computer logic control system.
# THE GALAXY COMPUTER CONTROL SYSTEM

_by_ Roderick Smit Mesecar

## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreward</td>
<td>1</td>
</tr>
<tr>
<td>Galaxy Computer Description</td>
<td>3</td>
</tr>
<tr>
<td>System Design</td>
<td>4</td>
</tr>
<tr>
<td> Introduction</td>
<td>4</td>
</tr>
<tr>
<td> Logical Oscillator</td>
<td>7</td>
</tr>
<tr>
<td> Cycler</td>
<td>10</td>
</tr>
<tr>
<td> Micrologic</td>
<td>13</td>
</tr>
<tr>
<td>System Modules</td>
<td>17</td>
</tr>
<tr>
<td> Introduction</td>
<td>17</td>
</tr>
<tr>
<td> Logical Oscillator Circuit</td>
<td>18</td>
</tr>
<tr>
<td> Driver Circuits</td>
<td>24</td>
</tr>
<tr>
<td> Cycler Circuits</td>
<td>30</td>
</tr>
<tr>
<td> Micrologic Circuits</td>
<td>32</td>
</tr>
<tr>
<td> Circuit Components</td>
<td>36</td>
</tr>
<tr>
<td>Conclusion</td>
<td>38</td>
</tr>
<tr>
<td>Suggested Thesis Topics</td>
<td>39</td>
</tr>
<tr>
<td>Bibliography</td>
<td>41</td>
</tr>
<tr>
<td>Glossary</td>
<td>42</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Fig.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Simplified Computer System Diagram.</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Logical Oscillator Block Diagram.</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Cycler Signal-Flow Diagram.</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>Sample Micrologic Function Block Diagram.</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>Logical Oscillator Circuit Diagram.</td>
<td>20</td>
</tr>
<tr>
<td>6a, b</td>
<td>Logical Oscillator Output Waveforms.</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>Lock-Delayed-Unlock Circuit.</td>
<td>22</td>
</tr>
<tr>
<td>8a, b, c, d</td>
<td>Several Configurations of Level-Shift and Driver Circuits.</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>Coaxial Cable - Gate Driver Circuit.</td>
<td>29</td>
</tr>
<tr>
<td>10</td>
<td>Circuit Diagram for Two Logic States of Cycler.</td>
<td>31</td>
</tr>
<tr>
<td>11</td>
<td>Abnormal Sequence Circuit.</td>
<td>33</td>
</tr>
<tr>
<td>12</td>
<td>Cycler Logic State Output Waveforms.</td>
<td>33</td>
</tr>
<tr>
<td>13</td>
<td>Circuit Diagram for Micrologic in Figure 4.</td>
<td>35</td>
</tr>
</tbody>
</table>
THE GALAXY COMPUTER CONTROL SYSTEM

FOREWORD

In 1946 J. von Neumann (1), H. H. Goldstine, and A. W. Burks, from Princeton University's Institute for Advanced Study, submitted a report to Army Ordnance for the logical design of an electronic computer. The report, titled "Preliminary Discussion for the Development of an Electronic Computer", set down principles that profoundly influenced subsequent computer development. The IAS computer system organization was developed as a parallel direct-coupled asynchronous machine with a stored program.

Undoubtedly the most significant concept introduced in the report, and the one which makes the modern computer a self-contained sequentially operating unit, is the stored program. Computers before the IAS machine operated with programs on external switches, wired patch boards or punched tape. This style of program control was painfully slow and awkward to use.

J. von Neumann's proposal was to place the program or instructions and the numbers in a quick access memory and then, when initiated, the computer would sequence the instructions* automatically to problem completion. This was done at computational

* See Glossary
rates limited by the internal program efficiency and electronic circuits.

This early concept stimulated other computer designers to develop a multitude of schemes for internal computer control. Accordingly, descendants of this early machine use many varied and hybrid methods of control, very few of which are described in literature. To some degree, this lack of publication is due to the complexity of the systems which are difficult to coherently describe. Also, some manufacturers consider the rights and descriptions of their particular control methods as company proprietary and, therefore, do not publish.

The Galaxy (2) Computer, recently designed at Oregon State University, and which also descends from the early IAS machine, was in need of a fast and flexible control system. Useable experience by the people at the Center for Computer Research, University of Chicago, dictated a certain approach to the control-system design. The degree of concurrent operation and individual logic circuit speeds in the Galaxy Computer required special considerations in control-circuit designs to meet the desired level of program efficiency.
GALAXY COMPUTER DESCRIPTION

The overall design of the Galaxy Computer is university oriented. Features are included in the design that would be omitted in most commercial machines. Such out of the ordinary instructions as significant-digit-floating-point-arithmetic were included to strengthen machine capabilities for the user. The instruction lists are large enough to make efficient compilers for ALGOL, FORTRAN and most other problem-oriented languages.

Add to the powerful instruction list concurrent operation, a one-hundred and fifty nanosecond asynchronous adder, one-megacycle memory, 58 binary digit word length with high-speed transistorized logic circuits and there is a brief description of the Galaxy Computer.
SYSTEM DESIGN

Introduction

This thesis is written to document the Galaxy Computer logic-control system and the implementation of this concept with electronic components. It is reduced to a skeleton system to better illustrate how it can function as a flexible, modular logic-control system. Several such systems can be interconnected to enlarge the computer's computational power with more concurrent operation. The design attempts to illustrate race-free* control circuits that use saturated level-logic in a hybrid synchronous-asynchronous interconnection.

Figure 1 illustrates a simple computer system with a single control system. This diagram was set up to help define major sections of the machine and give some idea of the interconnection of circuits to be discussed later. Any diagram of this form does not do justice to the complex maze of wires and components a computer really is.

Typically, data to be processed and instructions on how it is to be processed are stored in the memory. Through a simple program, an instruction in memory is fetched and placed in the register. The decode circuits attached to the register set up paths or gates

* See Glossary.
Figure 1. Simplified Computer System Diagram
in the wired micrologic. The collection of pre-set paths represents instructions as to what internal processing is to be done on the stored data.

Selection of a sequence for the performance of each instruction is done by the cycler. With the cycler it is possible to skip certain amounts of non-applicable instructions to make the computer more efficient. Each selected output of the cycler is timed by the logical oscillator. A semi-fixed time period is allowed by the logical oscillator for each instruction's operation.

In the Galaxy Computer the memory, the input/output equipment, the arithmetic system and the executive control (overall control) individually have control systems with more digital registers similar to the one illustrated. One cannot easily show the overlap between these independently operating units without describing each instruction in the program manual.

Stepping through the logic functions of each segment of the control system in more detail will give an insight to the flexibility and speed of the feedback of logic signals for positioning, transferring and sequencing of data in a computer.
Logical Oscillator

The logical oscillator generates two logically defined square waveforms that act as the timing periods for most of the instruction performance. The two signals are called the red and green states. Generally, positive setting of gates and movements of data is done on the red states. A turning off and recovery of the gates occurs on the green states. Sequential red-green data transmissions can be made, if there is no common logic involved. For most operations, the two states are of equal time periods, but this is not necessary.

Figure 2 is not an instructive diagram for referring to the operation of the oscillator. It does indicate, however, the most significant input controls and logic functions. Internally, the oscillator has independent timing circuits to adjust the frequency of red-green states and to cause them, if necessary, to be unsymmetrical. This same circuit can also be used to put a fixed delay between the red-green states if a particular timing operation requires it. These adjustments can be easily made by turning dials. Frequency variations of red-green can be adjusted so that the oscillator is operating at one cycle or ten million cycles per second. The one-step control makes possible the selection of red-green states for extended periods by hand-operated switches. This feature allows, as will be shown later, the stepping of the logic through each
Inputs From Micrologic And Asynchronous Circuits

Switch One-Step Control From Maintenance Console

Oscillator One-Step Control

Red Lock Delay Unlock

Red Lock Unlock

Red Cycle Timing Circuits

Green Cycle Timing Circuits

Timed Circuit Output Red Green

Final Output Stage

To Cycler

Figure 2. Logical Oscillator Block Diagram
instruction for location of program or wiring errors.

Most input signals to the logical oscillator are from the micrologic, special toggles*, asynchronous* complete circuits, other control systems, etc. Often it is desirable on some logical operations to stop the sequence. This can be done with lock-delayed-unlock controls.

The oscillator is set to step the instructions at some frequency. If a particular operation requires more time than allowed on a red-green state, the micrologic locks (stops) the oscillator until the operation can be completed. The locked period can be controlled in two ways: One, the operation for which the lock was initiated can return a complete signal (asynchronous), restarting the oscillator. Two, if a known delay period is required, the micrologic can initiate a delayed-unlock which adds a fixed time to that particular red-green state.

A second lock circuit has no delay associated with it. When an unlock signal is presented, the logical oscillator will immediately restart. These controls function the same for both red and green states; however, a red and green control should not be requested simultaneously.

Besides the micrologic and the logical oscillator, the lock-delayed-unlock functions can be activated by special toggles or

*See Glossary.
another control system.

The red-green states are sequenced through an output stage which insures that they do not occur concurrently. This is to prevent a red-green logic race condition. Oscillator red-green outputs are wired only to the cycler.

Cycler

Red-green inputs, from the logical oscillator, are made very distinct and selective in the cycler. This unit represents many toggles and gates that selectively sequence the red-green logic states to the micrologic. Where there are fifty sections of micrologic, there are fifty cycler outputs and only one of the fifty is ever on during a logical oscillator state. The orderly sequence of outputs can be altered by the micrologic with an abnormal sequence function.

The period of the red-green states is the same as for the logical oscillator. Recurrence of a particular state is equal to the logical oscillator frequency divided by the number of cycler states. This function and the cycler operation can be shown with Figure 3. The diagram has illustrated the equivalent of four cycler states. It is possible, however, to extend the number of states to at least fifty: Rk, Gk, Rk + 1, Gk + 1...Rk + 50, Gk + 50.
Figure 3. Cycler Signal-Flow Diagram
Assume the logical oscillator is running, all toggles are set to zero, and then a one is put into the Rk toggle. Setting of Rk closes gate G1 which allows the red state Rk to pass. Note that all other gates are open. The Rk signal once past G1 sets a one in G3. Even though G3 is set, there is no conducted signal because the oscillator is in the Rk state. When the logical oscillator flips to the green state, the Gk signal is conducted. Rk no longer exists and the Gk signal sets the Rk toggle to zero, thus opening G1. Gate G3 also sets Rk + n toggle (n = 1) to one so that when the logical oscillator flips, Rk + 1 appears at G5. On the Rk + 1 state, toggle Rk and G2 set toggle Gk to zero. This sequence is continued n times. To keep the unit continually recycling, gate Gk + n should be connected to the Rk toggle set-to-one input.

In particular, it should be noted that the gate to conduct next is pre-set by the preceding cycler state. This feature makes the activation of that cycler state very fast. When all of the micro instructions are wired into the machine, it is sometimes possible to single out an instruction which is common to several algorithms*. To make the control system more efficient on some algorithms, it is necessary to jump some of the cycler steps to where the common instruction is located. The jump or break in the normal cycler sequence is controlled by the micrologic. An abnormal

*See Glossary.
sequence circuit is activated and the jump is made as though the
cycler was going on in the normal sequential manner.

The abnormal sequence works in this way. First, gate G4
has to be inhibited and toggle Rk + n set to zero. Toggle Rk + n had
been automatically set to one at the beginning of state Gk. A one is
then set in the toggle Rk + (n + x) where x is the jump location
toggle. This all happens during the Gk state so that when the next
logical oscillator red state comes up, it will be the Rk + (n + x)
cycler state. Jumps can be made either to a higher or lower
numbered cycler state.

Micrologic

The individually designated outputs (Rk, Gk etc.) from the
cycler become the bases for timing operations of the micrologic
gates. Connected to each of the cycler outputs are a group of gates
which perform the minute details demanded by the programed
instructions. Many gates may be connected to a cycler state but
only a portion of the gates are used during a particular instruction.
Activation of specific gates for the performance of one step in the
program is done through the instruction register and decode matrix
as indicated in Figure 1. The decode pre-sets gates that further
logically NOR or NAND more functions together to obtain a desired
end operation. The block diagram of Figure 4 is an example of how the gates can be connected to perform an operation. From the cycler outputs, the gates can be branched in many forms so long as certain general rules are followed. One limitation is the number of series gates. There is a limit of three gates unless a driver circuit is added (Figure 4) and then the effective number of gates is six.

Each gate represents a NAND logic function with nine possible inputs. Logic NORing is done with parallel gates with a limit of six.

Timing signals are transmitted through the gates with very little time delay. This is essential because logic decisions sometimes have to be made before completing an instruction. For instance, functions energized after gate G11 may have return signals that turn on gate G12 or G22, depending on previous results from G11. This form of decision making sometimes requires more time than allowed on the cycler state and a lock signal is set on the logical oscillator by G11. After the signal to G12 or G22 is received, the logical oscillator is released and the process continues. Cycler jumps are handled in this same manner.

Control of overlapped operations between two control systems is also done in the micrologic. One system or the other becomes a master controller by using logical oscillator lock-unlock controls.
Inputs From Other Logic Control Systems, Special Toggles, Decode etc.

Figure 4. Sample Micrologic Function Block Diagram
Micrologic output signals are used to control the proper sequencing of data processing in the computer. The thousands of function bus signals trigger (through driver circuits at times) all forms of digital circuits. The true itemized operation of each function bus would have to be traced by using the logic design manual for a particular computer.
SYSTEM MODULES

Introduction

The control-system circuits are direct-current coupled and use saturated transistors for level-logic designation. A direct-coupled system has some advantages over the pulsed-circuit schemes. With the direct-coupled circuits, there is always a real connection that maintains the logic implications until the signal is released. This form of logic also can be operated at low or high frequencies without signal deterioration and timing difficulties. As indicated previously, the Galaxy Computer circuits can be one-stepped through each operation so that circuit debugging and program check out can be facilitated.

Minus two volts and ground have level logic significance in the control system. The logical one level for the digital registers is ground and minus two for the control units, i.e., oscillator, cycler, micrologic and decode. These two voltage levels have to be firmly maintained, with respect to each other, over the full dynamic switching range of the transistor circuits. A significant noise signal on either voltage bus would imply a logic signal. If time is allowed for the transient signal to dissipate itself, it would not interfere with the logic state of the machine. Control signals are between minus two and plus one volts.
All of the control units act as current sinks for the function bus signals; i.e., each digital system unit that is triggered is done so by a low-impedance source to the minus two volt bus.

Timing signals provided by the logical oscillator do not pass through an inverter amplifier until they are used in the digital system. This helps account for the control system circuit speed. It does, however, force the control circuits to switch large currents and this is the reason for the current drivers.

Most units in the Galaxy Computer have similar digital circuits. The binary storage unit is a toggle which is gated in several ways. The toggle has been thoroughly evaluated with a computer program (2) for wide static parameter variations in worst case design. Small changes occur in the resistor values for toggles in different parts of the computer, but they are still within design limits.

**Logical Oscillator Circuit**

The logical oscillator is similar to an RC-timed multivibrator circuit. However, innovations are included in the circuit design which make it a good logical timing clock for the Galaxy Computer. The timing unit must have the following control functions:

(1) Direct-current coupled circuit

(2) 0 to 10 megacycles frequency range
(3) Lock-delayed-unlock with near instantaneous restarting

(4) Remote control

(5) Race-free output signals

The circuits in Figures 5 and 7 fulfill the stated requirements.

It is easiest to explain the circuit by starting with the timing circuits. The bases of inverters Q5 and Q14 are controlled by the diversion of base currents to discharge capacitors C1 and C2. Transistors Q9 and Q10 serve as current sources for the base currents of Q5 and Q14. Emitter followers Q4 and Q15 serve to recharge C1 and C2 for the timing period.

A timing cycle would be started with Q14 saturated and Q5 off. Emitter follower Q4 has then forced, with very small time delay, C1 to charge through CR2, and CR3 to the minus two volt bus. The voltage across C1 is about 4 volts positive at emitter of C4. When the computer run signal is initiated, the lock circuit input to Q7 is released allowing current from Q9 to saturate Q5. When the collector of Q5 goes down, diode CR1 forward biases forcing C1 to shift its plus potential to about 0.7 volts above minus two volts bus. The negative side of C1 has reverse biased CR2 and pulled the emitter of Q13 to about 3.3 volts below the minus 2 volts bus. This has caused base current normally entering Q14, by the forward biased base-collector diode of Q13, to be shunted to discharge C1.
Figure 5. Logical Oscillator Circuit Diagram
Immediately Q14 goes off and, by C15, capacitor C2 is charged through CR4 and CR3 to previous voltage levels similar to C1 in eight nanoseconds. The recharge time of C1 or C2 is governed by the collector resistor of Q4 and Q14 and the equivalent of twelve picofarads capacitance to the minus 2 volts bus.

The voltage on C1 is linearly being reduced by the constant current from Q10. After some fixed time period, the emitter of Q13 will rise, due to the elimination of restraining voltage of C1 and divert current into the base of Q14. One oscillator time period is a linear relationship as shown:

\[
T = \frac{(V_{C1}) (C_1) (R_s)}{V_s} \text{ seconds}
\]

where
\[
V = \text{volts} \\
C = \text{farads} \\
R = \text{ohms}
\]

Frequency of oscillation is then \( f(\text{cps}) = \frac{1}{2T} \).

Voltage \( V_s \) is under the control of two potentiometers and one variable resistor. The two potentiometers adjust the time periods independently for transistors Q5 and Q14. At times, unsymmetrical waveform can be used, as in the case of the need for a longer transmitting red state and less recovery time green state. The variable resistor is used to control the base frequency. Figure 6a is an oscillograph of the red state with the symmetry controls set for two illustrations of duty cycle control.
(a) Two Illustrations of Symmetry Control On Red State Waveform

(b) Transition of Output Red-Green States

Figure 6. Logical Oscillator Output Waveforms

Figure 7. Lock-Delayed-Unlock Circuit
Transistors Q3 and Q16 are merely signal-coupling devices from the output of the timing stage to a toggle. The toggle acts as a timing-circuit synchronizer for the red-green states.

Signals from the toggle are race-free in that during the transition between states, the two states are never on at the same time. If this did happen, red and green-state logic from the cycler would try to activate function buses at the same time. There are no inverter amplifiers beyond the toggle to the output of the function buses to deteriorate red-green state time isolation. The toggle output waveform rise and fall times are each ten nanoseconds at 10/90 percent points for a four-volt signal. There is no rise and fall time deterioration over the d.c. to ten megacycle frequency range. Figure 6b is an oscillograph made with a Tektronix type 585 oscilloscope with a P80 plugin.

For remote one-step control, switches can be connected to the emitters of Q8 and Q11. When the one-step switch is activated, both emitters are at minus two volts and base currents for Q5 and Q14 go to zero. Pushing either normally closed red or green one-step switches allows Q5 or Q14, with released base current from Q9 or Q10, to trigger the toggle.

The lock-delay-unlock circuits are used for logic function control of the logical oscillator (Fig. 7). The circuit is triggered on
either a red or green state micrologic signal. For a lock signal, the emitter of Q1 is pulled to a ground potential. This signal is level-shifted by CR2 to saturate Q2 at minus two volts. Transistor Q7 or Q12 then clamps the logical oscillator much the same way the one-step function does. Whichever state the lock circuit clamps, the inverter Q5 or Q14 on that state immediately recharges the timing capacitor. Diodes CR2 and CR4 and their connection with Q6, Q13 make this possible by isolating the bases of Q5 and Q14. This feature enables the oscillator to immediately restart when unlocked.

For a delayed-unlock, a small capacitor Cn is added to the standard lock-unlock circuit as shown in Figure 7.

Several delayed unlock circuits can be set up with different values of Cn to have a selection of delayed-unlock periods. Multiple diode fan-in is permissible on both lock and unlock inputs.

**Driver Circuits**

Different types of driver circuits are necessary in a computer system. The purpose of each circuit is to make, while preserving the high speed characteristics of the original signal, some form of impedance match for larger current or voltage switching. Decay of the transmitted wave shapes, due to stray capacitance and
slow switching circuits, impair the operational speed of the computer. Drivers are necessary to transmit function bus signals through sections of coaxial cable and to switch many register gates concurrently. The driver circuits provide high-current cycler and logical oscillator timing signals, just to state a few uses. The circuits shown in Figure 8, with low-capacitance voltage level-shift and emitter follower amplifier circuits, have high-current gain with minimum time delay.

The low-capacitance level-shift circuit uses any voltage reference diodes, up to 12 volts, supplied by an isolated 90-volt power supply through two 10k ohm resistors.

The isolated power supply and resistors maintain 4.5 millamperes through the diode to simulate a battery of Vfd volts with about 15 ohms internal resistance.

The reference diode CR1 must have a voltage drop higher than the sum of the VCE saturation voltage of Q1 and VBE saturation voltage of Q2. This is to insure Q2 is switched to saturation. It appears as though Q1 or Q2 could not saturate due to the standoff voltage of the diode.

The current in the diode is not large enough to make it a lower impedance than either the saturated collector-emitter of Q1 or the base-emitter saturation of Q2. Because of the diode's higher
Figure 8. Several Configurations of Level-Shift and Driver Circuits
impedance, both transistors will saturate and the diode will have a resultant voltage drop equal to $Q_1V_{ce} + Q_2V_{be} - Q_2V_{ce}$. The diode's apparent resistance has increased when forced out of saturation by $Q_1$ and $Q_2$. However, the resistance is not high enough to influence the isolated current source, so the base current of $Q_2$ will remain at 4.5 millamperes.

This circuit effectively shifts the collector voltage of $Q_1$ to drive $Q_2$ without increasing the normal collector current in $Q_1$ beyond the base current of emitter follower $Q_2$.

Figure 8b illustrates the fan-in possibilities, where the resistor on the base of $Q_2$ is necessary to return the level-shift circuit to a positive voltage level. To prevent current hogging between $Q_2$, $Q_3$ in Figure 8c, two level-shift circuits are employed. In this way, each transistor is assured of 4.5 millamperes base current. One level-shift circuit could be used by connecting both bases, each in series with 20 ohms, together. The resistors reduce the current hogging problems, but also half the normal 4.5 millampere individual base current.

The circuits in Figures 8a, b, c offer current gain only. The dynamic voltage levels of the inverters that drive these circuits are adequate to make them compatible with the logic circuits.
Contrary to many of the driver circuits are those needed to drive the register gates. These gates demand a fast positive going signal that can deliver currents for 20 to 120 transistors. Figure 9 shows a driver circuit that can transmit control signals through coaxial cable or twisted-pair wire some distance to the register gates. When the base of Q1 is pulled down by the function bus, emitter followers Q2 and Q3 are driven into saturation at six volts and four volts respectively. A 130-ohm termination on 93-ohm coaxial cable and twisted-pair wire worked quite satisfactorily.

To prevent reverse base-emitter voltage breakdown on the register gate transistor, diode CR3 is used to clamp the output of Q3 from going below 0.8 volts.

The switching speed of this circuit is dependent upon the inverter transistor Q1 and length of cable. Six to ten nanoseconds is the inverter delay and about twelve nanoseconds for eight to ten feet of cable.

Other special driver circuits are often needed in a computer system. Most of them can be assembled from the schemes just described to fit the application.
Figure 9. Coaxial Cable - Gate Driver Circuit
Cycler Circuits

The cycler is a system of gates and toggles that selectively gate the logical oscillator red-green states to the micrologic circuits. As seen in Figure 10, the toggles have standard component values. The toggles have a voltage swing of 3.7 volts which is compatible with the system logic levels. Gates between the toggles and the series red-green gates are transistorized so that all signals in the cycler use the same minus two volt bus. This common voltage bus is a good method of maintaining minimum ground noise.

The earlier discussion of how the cycler logically functions for Figure 3 applies to Figure 10. The only exception, Figure 3, included four cycler states where Figure 10 has enough circuit for only two.

To start the cycler, a negative going signal pulls collector of Q2 to minus two volts. Emitter-base of Q3 is then back biased allowing base current into Q4 so that when the red state is turned on, output Rk, of Q4, follows with approximately one nanosecond delay. Saturated Q4 will force Q10 to pull Q8 into saturation. This turns off Q9 and Q7, thus pre-setting Q13 against a clamp diode. The logical oscillator flips to the green state and output Gk is delayed about one nanosecond.
Set-to-one Signal From Gk-1

Logical Oscillator

Red Bus

Green Bus

Figure 10. Circuit Diagram For Two Logic States of the Cycler
Saturation of Q13 turns on Q11 and off Q2, thus reverse biasing emitter-base of Q5. Transistor Q6 is then prepared to turn off Q8 on the next red state and turn off Q13. These two cycler states are then inoperative until a new set-to-one signal is transmitted to the Rk toggle.

Transistor Q12 is the connective gate for the normal sequencing of the cycler. If an abnormal sequence is necessary, a circuit, as shown in Figure 11, is used. The circuit is turned on by the micrologic and performs the stated functions.

An oscillograph in Figure 12 shows the combined performance of the logical oscillator, delayed-unlock circuit and cycler. The cycler is connected for a continual sequence of four states. The first three states are with a normal logical oscillator period. On the Gk + 1 state, a delay was triggered that extended the period.

Micrologic Circuits

Effectively interconnected gates, which logically determine functions to be completed for digital processing, can substantially increase a computer's capabilities. Because of the thousands of functions performed during a program, it is important that each function is electrically switched as fast as possible.
Figure 11. Abnormal Sequence Circuit

Micrologic Abnormal Sequence Signal

6v
3.3k
820
1v
Q1
CR1

45v
10k
820
Q2
CR2

-45k
10k
CR3

Inhibit To the Zero Side of the Rk+1 Toggle
To the One Side of the Rk+(n+1) Toggle

Figure 12. Cycler Logic State Output Waveforms

Trace
Rk
Gk
Rk+1
Gk+1 with a delay period

(Vertical, Horizontal - No Scale)
Gating techniques shown in Figure 13 are almost as fast as the basic timing unit. Each gate adds but a few nanoseconds delay to the propagated signal. The chief reason for this is that each gate is normally pre-set.

Clamp diodes are provided on the emitter of each transistor gate level so that when the decode releases the base diodes, the transistor will partially saturate. If a gate has not been pre-set, the prime delay in turning the transistor on is a function of the charge time for the wire lead capacitance on the base. For long (2 to 8 ft) signal transmissions, it is necessary to use coaxial cable.

Each transistor base has an independent current supply through a 4.7k ohm resistor to the ten volt bus. The base current is used as a standard load to compute, for a section of logic with unused gates, the decode current sink requirements.

The voltage level-shift circuit shown in Figure 13 is the same as the one described in the section on driver circuits. A level-shift circuit can be connected at any gate level. Adjustment of the voltage shift can be made by adding or removing diodes.

Only two function bus decoupling diodes are shown at each gate level. The number can be increased to nine without interference with logic switching speed. Too many diodes forward biased do cause a delay because of individual storage characteristics. It may
Figure 13. Circuit Diagram For Micrologic In Figure 4.
be necessary to use a resistor, connected as Rx to gate Q13, to aid the reverse biasing of diodes on turn off.

Care has to be taken to minimize pull-up resistor applications because they aid in taxing the current-driver circuits. The number of series gates is limited to three without a level-shift circuit and amplifier. The limitation on series gates is due to the transistor saturation voltage variance for a 10 to 100 millampere range in function bus currents. Micrologic section with light function bus currents can have a fourth gate added.

Cycler output voltage swings are clamped to adequately allow the function bus diodes on the emitter level of Q10 to be fully reverse biased. This reduces the capacitive load, at the function bus connection, on the digital system circuits.

Circuit Components

Components for the Galaxy Computer were selected for their quality and reliability.

The selection of Fairchild planar epitaxial 2N2368 transistor for the basic digital transistor was based upon extensive tests of NPN and PNP switching transistors. A large sample of transistors were tested and documented for a broad saturation-beta range. Two specified characteristics on the 2N2368 were maximum saturation
voltages at $I_c = 30$ and 100 milliamperes. With a beta of 10, the maximum saturation voltages were 0.25 and 0.5 volts, respectively.

The complementary PNP transistor for most of the driver circuits was the Motorola 2N965. This is a germanium mesa unit with very good switching characteristics. Epitaxial PNP silicon transistors are now on the market with equivalent switching speeds at high power dissipation that would be a good substitute.

The logic application and clamp diodes were of the Fairchild FD100 ultrafast planar type. Reverse recovery of these diodes was specified at 2 nanoseconds. Very satisfactory gold bonded germanium diodes were also used. They were the Clevite CGD 1030. Voltage level-shift circuits used two types of diodes, Clevite IN816 reference diode with a tightly controlled forward voltage specification. The Zeners were of the Motorola alloy junction one-fourth watt type and were used because they have lower dynamic resistance at currents from one to ten milliamperes at five volts.

Allen Bradley hot molded carbon five percent one-fourth watt resistors were used in most applications. Other applications in the computer required different types of transistors, diodes, etc., but wherever possible, the previously described components were used.
CONCLUSION

The Galaxy Computer logic-control system is a nanosecond class of electronic circuits that offer a very modular assembly scheme. Interlock controls between control systems allow the development of concurrent logic operations. The flexibility of the logical oscillator and micrologic circuits, plus the logic jump feature of the cycler, contribute to the efficiency of the computer system. Each circuit described in this thesis has been assembled and operated under conditions for which it was intended. Satisfactory results were obtained and documented (2).
SUGGESTED THESIS TOPICS

A controversy exists as to which type of timing techniques are best. Asynchronous, two phase clocking and four-phase clocking schemes have their individual advantages for certain machine applications. A very useful thesis to the computer designers is a method where the dynamic characteristics of the machine could be evaluated for particular algorithms using different timing techniques. This would have to include some form of time description for the logic and control circuitry. For an introduction to this problem, see the paper presented by D. J. Chesarek at the 1963 Pacific Computer Conference, titled "Logic Limitations of Proposed Gigahertz Circuits."

A program written by Lynn Quam for the simulation of computer logic could be adapted to approach this type of problem. His paper is described in the Galaxy (2) Computer publications under the title, "A Formal Language for Computer Design Automation."

Although the toggle is a specialized digital circuit, it has characteristics common to large-signal, inverters, amplifiers, logic gates, etc. A rather challenging thesis problem would be the inclusion of both dynamic and static operating characteristics into one analysis procedure for this kind of circuit.
On a smaller scale, a very necessary circuit in a computer system is one that has a low impedance drive for both positive and negative signals. Often, complementary transistors are coupled in a class B connection to perform this function. There is a cross-over period in that configuration that offers a mis-match to the circuits being driven.

The development of the circuit should be direct connected, high (40) current gain, with no input signal loss or deterioration.
BIBLIOGRAPHY


GLOSSARY

ALGORITHM ----- A fixed step-by-step procedure for accomplishing results, usually a simplified procedure for accomplishing a complex result.

ASYNCHRONOUS ----- A term applied to a computer in which the execution of one operation is dependent on a signal that the previous operation is completed, rather than a fixed cycle time.

DECODE ----- To determine the meaning of coded characters.

INSTRUCTION ----- A set of identifying characters designed to cause a computer to perform certain operations. A machine instruction consists of one or more operations.

PROGRAM ----- A number of grouped machine instructions which carry out a well-defined function for the solution of a given problem.

RACE FREE ----- The condition of no concurrent performance by associated circuits that have logic conflicts.

TOGGLE ----- One of several terms used to describe a binary storage circuit. Other names are bistable multivibrator and flip-flop circuits.