**PROJECT**
# WHIRLWIND
(DEVICE 24-X-3)

REPORT R-127

# WHIRLWIND I COMPUTER BLOCK DIAGRAMS

Submitted to the
SPECIAL DEVICES CENTER
OFFICE OF NAVAL RESEARCH
Under Contract N5ori60

Report by
R. R. Everett & F. E. Swain
SERVOMECHANISMS LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge 39, Massachusetts

Project DIC 6345

September 4, 1947

# FOREWORD

This report is a description of the Whirlwind electronic digital computers under development at the Servomechanisms Laboratory of M. I. T. for the Office of Naval Research. The arithmetic nature and the physical nature of the computers are covered briefly, and the block diagrams for the prototype computer WWI are discussed in some detail.

# TABLE OF CONTENTS


Page No.

1. INTRODUCTION . . . . . 1
   - 1.1 The Whirlwind Computers . . . . . 1
     - 1.11 Whirlwind I Computer
     - 1.12 Whirlwind II Computer
   - 1.2 The Present Set of Block Diagrams . . . . . 2
   - 1.3 The Organization of this Report . . . . . 2
     - 1.31 Breakdown into Sections of Increasing Complexity
     - 1.32 Section 2 — Preliminary Considerations
     - 1.33 Section 3 — A General Outline of the System
     - 1.34 Section 4 — The Arithmetic Element
     - 1.35 Section 5 — The Block Diagrams
     - 1.36 Section 6 — The Timing Diagrams

2. PRELIMINARY CONSIDERATIONS . . . . . 4
   - 2.1 Storage . . . . . 4
   - 2.2 Parallel Transmission of Digits . . . . . 4
   - 2.3 Arithmetic Considerations . . . . . 5
     - 2.31 The Binary Number Base
     - 2.32 Fixed Point
     - 2.33 Signs and Negative Numbers
   - 2.4 Switching Considerations . . . . . 6
   - 2.5 An Order Code . . . . . 6
     - 2.51 The Standard Order
     - 2.52 Desired Operations
     - 2.53 An Order Code
   - 2.6 Word Length . . . . . 11
     - 2.61 WWI
     - 2.62 Computing with Multiple-Length numbers
     - 2.63 WWII
   - 2.7 Input and Output . . . . . 12
   - 2.8 Checking . . . . . 12
     - 2.81 General Considerations
     - 2.82 Arithmetic Check
     - 2.83 Transfer Check
     - 2.84 Other Checks
   - 2.9 Required Computing Speed . . . . . 13
     - 2.91 For WWII
     - 2.92 For WWI
     - 2.93 Storage Read Time
     - 2.94 Computer PRF

3. A GENERAL OUTLINE OF THE SYSTEM . . . . . 14
   - 3.1 System Build Up . . . . . 14
   - 3.2 Storage Characteristics . . . . . 15
   - 3.3 Control Characteristics . . . . . 15
     - 3.31 Clock
     - 3.32 The Program Counter
     - 3.33 Time Pulse Distributor
     - 3.34 Operation Control

## TABLE OF CONTENTS


Page No.

3. A GENERAL OUTLINE OF THE SYSTEM — Continued

3.4 Program Register Characteristics . . . . . 18
3.5 Arithmetic Element Characteristics . . . . . 18
3.6 Input and Output Characteristics . . . . . 18
3.7 Readback Checking . . . . . 19
3.8 Test Storage . . . . . 19
3.81 Toggle-Switch Storage
3.82 Flip-flop Storage
3.83 Setup
3.84 Counter Modifications
3.85 Starting the Computer Using Test Storage

4. THE ARITHMETIC ELEMENT . . . . . 21

4.1 Binary Arithmetic . . . . . 21
4.11 Addition
4.12 Subtraction
4.13 Multiplication
4.14 Shifting
4.15 Division
4.2 The Arithmetic Registers . . . . . 25
4.3 The A-Register . . . . . 25
4.4 The Accumulator . . . . . 25
4.41 Simple Adder
4.42 High-Speed-Carry
4.43 Shift and Carry
4.44 Shift Left
4.45 Other Requirements
4.5 The B-Register . . . . . 27
4.6 The Operations . . . . . 27
4.601 Clear and Add — *ca*
4.602 Add — *ad*
4.603 Clear and Subtract — *cs*
4.604 Subtract — *su*
4.605 Multiply and Roundoff — *mr*
4.606 Multiply and Hold Full Product — *mh*
4.607 Divide — *dv*
4.608 Transfer to Storage — *ts*
4.609 Shift Right — *sr*
4.610 Shift Left — *sl*
4.611 Subprogram — *sp*
4.612 Conditional Program — *cp*
4.613 Transfer Digits — *td*
4.614 Special Add — *sa*
4.615 Store and Display — *sd*

5. THE BLOCK DIAGRAMS . . . . . 36

5.1 The Complete System . . . . . 36

# TABLE OF CONTENTS


Page No.

5. THE BLOCK DIAGRAMS — Continued
   - 5.11 The System
   - 5.12 Bus Connections
   - 5.2 The Control . . . . . 36
     - 5.21 The Complete Control
     - 5.22 The Master Clock
     - 5.23 The Program Counter
     - 5.24 The Program Register
     - 5.25 32-Position Matrix Switch
     - 5.26 Operation Control
     - 5.27 Time Pulse Distributor Control
     - 5.28 Pulse Distributor .
     - 5.29 Program Timing Matrix
   - 5.3 Test Storage . . . . . 38
     - 5.31 Storage Chassis Arrangement
     - 5.32 Flip-Flop Storage
     - 5.33 Storage Output Section
     - 5.34 Flip-Flop Storage Control
     - 5.35 Toggle-Switch Storage
   - 5.4 The Arithmetic Element . . . . . 39
     - 5.41 The Complete Element
     - 5.42 Section of A-Register
     - 5.43 Accumulator Sections
     - 5.44 Accumulator Sections AC0–AC15
     - 5.45 B-Register Section
     - 5.46 B-Register Sections BR0, 1, 14, 15
     - 5.47 Step Counter
   - 5.5 Check Register . . . . . 41

6. THE TIMING DIAGRAMS . . . . . 42
   - 6.1 Program Timing . . . . . 42
   - 6.2 Operation Timing . . . . . 43
     - 6.201 Timing for Add — *ad*
     - 6.202 Timing for Clear and Add — *ca*
     - 6.203 Timing for Subtract — *su*
     - 6.204 Timing for Clear and Subtract — *cs*
     - 6.205 Timing for Multiply and Roundoff — *mr*
     - 6.206 Timing for Multiply and Hold Full Product — *mh*
     - 6.207 Timing for Divide — *dv*
     - 6.208 Timing for Transfer to Storage — *ts*
     - 6.209 Timing for Shift Right — *sr*
     - 6.210 Timing for Shift Left — *sl*
     - 6.211 Timing for Subprogram — *sp*
     - 6.212 Timing for Conditional Program — *cp*
     - 6.213 Timing for Transfer Digits — *td*
     - 6.214 Timing for Special Add — *sa*
     - 6.215 Timing for Store and Display — *sd.*

# LIST OF DRAWINGS

| Figure Number | Drawing Number | Title |
|---|---|---|
| 1 | A–30339 | General Block Diagram |
| 2 | A–30445 | Store Result |
| 3 | A–30440 | Origin of Orders |
| 4 | A–30441 | Setup Order |
| 5 | A–30442 | Read Out Order |
| 6 | A–30443 | Setup Operation |
| 7 | A–30444 | Perform Operation |
| 8 | A–30446 | Subprogram |
| 9 | A–30454 | Electronic Switch |
| 10 | A–30452 | Time Pulse Distributor |
| 11 | A–30453 | Control |
| 12 | A–30400 | Decimal Addition |
| 13 | A–30401 | Binary Addition |
| 14 | A–30402 | A Binary Adder |
| 15 | A–30417 | Negative Numbers |
| 16 | A–30410 | Subtraction Using 9's Complements |
| 17 | A–30403 | Decimal Multiplication |
| 18 | A–30355 | Binary Notation |
| 19 | A–30409 | Binary Multiplication |
| 20 | A–30404 | Modified Binary Multiplication |
| 21 | A–30447 | Multiplication I |
| 22 | A–30448 | Multiplication II |
| 23 | A–30449 | Multiplication III |
| 24 | A–30450 | Multiplication IV |
| 25 | A–30451 | Multiplication V |
| 26 | A–30406 | Rounding Off |
| 27 | A–30683 | Decimal Division |
| 28 | A–30686 | Binary Division |
| 29 | C–37072 | Arithmetic Element |
| 30 | A–30433 | Accumulator |
| 31 | A–30432 | Accumulator |
| 32 | A–30685 | Shift and Carry |
| 33 | A–30684 | Shift Left |
| 34 | C–37099 | Clear and Add |
| 35 | C–37100 | Add |
| 36 | A–30682 | Arithmetic Check |
| 37 | C–37101 | Clear and Subtract |
| 38 | C–37102 | Subtract |
| 39 | C–37103 | Multiply and Roundoff |
| 40 | C–37104 | Divide |
| 41 | C–37105 | Transfer to Storage |
| 42 | C–37106 | Shift Right |
| 43 | C–37107 | Shift Left |
| 44 | C–37108 | Conditional Program |
| 45 | C–37109 | Special Add |
| 46 | C–37071 | System Block Diagram |
| 47 | B–37073 | Control Functions |
| 48 | B–37070 | Bus Connections |

## LIST OF DRAWINGS

| Figure Number | Drawing Number | Title |
|---|---|---|
| 49 | B–37098 | Control |
| 50 | B–37058 | Master Clock |
| 51 | B–37062 | Program Counter |
| 52 | B–37067 | Program Register |
| 53 | B–37066 | Control Switch |
| 54 | C–37077 | Operation Matrix I |
| 55 | C–37078 | Operation Matrix II |
| 56 | B–37076 | Time Pulse Distributor Control |
| 57 | B–37068 | Pulse Distributor |
| 58 | B–37075 | Program Timing Matrix |
| 59 | C–37064 | Storage Chassis Arrangement |
| 60 | B–37057 | Flip-flop Storage Section |
| 61 | B–37060 | Storage Output Section |
| 62 | B–37061 | Flip-flop Storage Control |
| 63 | C–37072 | Arithmetic Element |
| 64 | B–37056 | Section of A-Register |
| 65 | C–37063 | Accumulator Sections |
| 66 | C–37096 | Accumulator Sections |
| 67 | B–37069 | B-Register Sections |
| 68 | B–37097 | B-Register Sections |
| 69 | B–37074 | Step Counter |
| 70 | B–37065 | Check Register |
| 71 | A–30874 | Program Timing |
| 72 | B–37080 | Timing for Add |
| 73 | B–37081 | Timing for Clear and Add |
| 74 | B–37082 | Timing for Subtract |
| 75 | B–37083 | Timing for Clear and Subtract |
| 76 | B–37084 | Timing for Multiply and Roundoff |
| 77 | B–37085 | Timing for Multiply and Hold Full Product |
| 78 | B–37094 | Timing for Divide |
| 79 | B–37086 | Timing for Transfer to Storage |
| 80 | B–37088 | Timing for Shift Right |
| 81 | B–37089 | Timing for Shift Left |
| 82 | B–37090 | Timing for Subprogram |
| 83 | B–37091 | Timing for Conditional Program |
| 84 | B–37092 | Timing for Transfer Digits |
| 85 | B–37093 | Timing for Special Add |
| 86 | B–37087 | Timing for Store & Display |
| 87 | B–37001 | Parallel Digit Computer Codes |

# 1. INTRODUCTION

## 1.1 The Whirlwind Computers

The Whirlwind Computers will be of the high-speed electronic type in which numbers will be handled as video signals and computation will be carried out in digital form. Digit transmission will be of the parallel type. The basic principle of the machines is to carry out one arithmetic operation at a time and to depend upon high operating speed rather than simultaneous computing operations to shorten the solution time of problems. The performance of a single arithmetic operation at a time will greatly simplify the setup of problems, and it is considered more feasible to obtain high operating speed than to complicate the equipment with simultaneous operations.

**Numbers will be handled in the binary system of notation.** This system, which makes use of only the digits zero and one, is especially well suited for use in electronic video circuits. The operators and those using the Whirlwind Computers for the solution of problems will, of course, work in the decimal system. Suitable conversion equipment will be provided at the input and output to transfer between the decimal and binary number bases.

The computers will have large internal storage of the high-speed type which can be used interchangeably for either control orders or numerical values. Sufficient internal storage will be provided to permit solution of most of the problems now contemplated without the use of an external slower-speed storage. Sufficient flexibility in control will, however, be provided to make possible the addition of external storage for supplementing internal storage when this becomes necessary and when satisfactory external storage media have been developed.

The Whirlwind Computer will use parallel digit transmission over a bus system providing one channel for each binary digit of the numerical values.

Figure 1 shows a simplified block diagram of the Whirlwind type computer. Figure 46 shows a more complete block diagram which will be discussed in later sections. Each major unit of the computer can transmit digits to the parallel digit transfer bus and can receive digits from this bus. The operation of the various output and input gates between the bus and units of the computer is controlled in the proper sequence by the central control unit.

The basic impulse rate for operation of the computer will be one megacycle. Pulses at a repetition rate of one megacycle will be transmitted from the central control to the other elements of the Whirlwind Computer. Within the Arithmetic Element itself, however, this basic impulse rate of one megacycle will be increased to two or possibly four megacycles in order to obtain higher multiplying rates. The Arithmetic Element will operate at this higher frequency with pulses of 1/10 microsecond duration or less. For a computer carrying numbers forty binary digits in length, these operating frequencies make possible a multiplication interval of forty to fifty microseconds. This interval is sufficient for all operations necessary in a multiplication of two forty-digit binary numbers, including transfers to and from storage.

Control orders, initial data, and partial and intermediate results will be stored in electrostatic storage tubes. These tubes will be of the beam deflection type with storage of ones and zeros as positive and negative charges on a dielectric surface. Appropriate circuits will be provided with the storage system so that stored signals will be self-maintaining and so that repeated extraction of a single number from storage is possible. Erasing of a stored signal will not be necessary prior to storage of a new value. Storage will therefore present no restrictions to the manner in which a problem may be programmed or solved.

It is anticipated that except for the electrostatic storage tube, most vacuum tubes in the computer will be of standard types.

Orders and numbers may be stored interchangeably in the computer storage system. Greater flexibility is thereby achieved, and any arbitrary division may be made between orders and numerical values in the computer storage. Problems with a small controlling program and a large amount of initial data, or problems with a large controlling program and small initial data and partial results, may be handled with equal flexibility.

### 1.11 Whirlwind I Computer

The Whirlwind I Computer is being designed as a prototype for the Whirlwind II system. The Whirlwind I system will initially be used to study and test the circuits and operating principles which will go into the Whirlwind II Computer. The Whirlwind I system, though intended principally as a prototype mechanism, will nevertheless have greater speed, flexibility, and storage capacity than digital computers in existence at the present time. Following preliminary performance tests, it will be used for research into the mathematics of numerical analysis and digital computation. It will provide a computing system for studying digital computer applications and for laying the foundation necessary for effective use of the Whirlwind II system.

The Whirlwind I Computer is being planned for a storage capacity of 2,048 numbers of 16 binary digits each. Sixteen binary digits are sufficient for the storage of control orders and are satisfactory for a very few mathematical problems. In order that the Whirlwind I Computer may be effective for general mathematical work, it will be arranged to compute with numbers having a length which is any required multiple of 16 binary digits. Setting up problems for computation of multiple-length numbers will be no more difficult than the programming of solutions in the basic number length of 16 digits.

### 1.12 Whirlwind II Computer

Detailed design of the Whirlwind II Computer has not been started. In general, it will be an extension of the Whirlwind I system to a larger number of digits and a greater storage capacity. Basic operating speeds will remain the same as in Whirlwind I. It is expected that 640,000 binary digits of high-speed internal storage will be provided in the Whirlwind II system. Number length in the Whirlwind II Computer will lie between 40 and 46 binary digits.

It is expected that the Whirlwind II Computer will be very similar to the Whirlwind I and will incorporate only those changes found desirable after observation of performance of the Whirlwind I system. Photographic input and output will be provided. Some form of magnetic storage may be used with the Whirlwind II system for those applications where an erasable medium is necessary.

## 1.2 The Present Set of Block Diagrams

In the present set of block diagrams the basic nature of the Whirlwind Computers is described. The more fundamental arithmetic operations of addition, subtraction, multiplication, and division are discussed in detail as are the control operations of subprogramming, digit extraction, shifting, and certain special operations.

Operation of the Arithmetic Element and the central control is explained with timing diagrams showing time relationships of pulses for executing the required computer operations.

The control orders discussed are only a partial list of those to be available in the Whirlwind I Computer. Control orders still to be developed include operations with absolute numbers, input and output, start and stop signals, and computation with multiple-length numbers.

The present block diagrams do not treat electrostatic storage, decimal-binary converters, or input and output devices, because studies are not yet completed. To make the block diagrams workable and to permit testing of computer components, as they are completed, a small amount of "test storage" is illustrated. This test storage provides 32 words of 16 binary digits each including 5 words of high-speed vacuum-tube flip-flop storage, and 27 words of manually set toggle-switch storage. The basic computer system has been so designed that it is largely independent of the final high-capacity storage characteristics, and the electrostatic storage can be added to the system as it becomes available. It is anticipated that the test storage will be retained in the final system for checking purposes.

An incomplete treatment of checking is presented. Checking of transfers is incorporated, but automatic checking, programmed check problems, mathematical checks, checking by duplicate operation, and trouble-shooting facilities have not been sufficiently studied and will be reported at a later date.

## 1.3 Organization of this Report

### 1.31 Breakdown into Sections of Increasing Complexity

In order to make this set of diagrams more readily understandable, it is advisable to discuss them on several levels of detail. An attempt will be made to give an overall picture of the system and how it works before going into the detailed characteristics of the several components. In ad-

dition, this method of presentation will allow the reader to choose the detail in which he desires to study the system.

The following paragraphs discuss briefly the contents and purposes of the different sections of this report.

### 1.32 Section 2 — Preliminary Considerations

The section on Preliminary Considerations contains a brief discussion of the basic problems involved in designing a computer, as well as substantiation of the various solutions arrived at in the design of Whirlwind I.

### 1.33 Section 3 — A General Outline of the System

The section on general system outline discusses the entire computer in general terms. The number and nature of the components are discussed briefly. An outline of the manner in which the computer operates is given.

### 1.34 Section 4 — The Arithmetic Element

The methods of binary arithmetic are outlined. Simplified block diagrams are given showing how these elementary methods may be mechanized using basic electronic components.

### 1.35 Section 5 — The Block Diagrams

Section 5 contains a detailed discussion of the block diagrams of the complete system and components.

### 1.36. Section 6 — Timing Diagrams

The timing diagrams required for controlling the operation of the entire computer are discussed.

## 2. PRELIMINARY CONSIDERATIONS

### 2.1 Storage

The characteristics of the storage to be used have more effect upon the design of a computer than any other consideration. The evolution of the Whirlwind Computers has been determined by the results of project work on electrostatic storage.

**There are two basic types of storage.** Serial storage contains elements which store entire numbers in space or time order. This type is illustrated by the mercury delay line, in which a group of pulses is stored in time sequence in a mercury column. Storage may be maintained indefinitely in a column of finite length by recirculating. The original electrostatic storage tube proposed for Project Whirlwind was also of this type. The sequence of pulses was stored spacially on a dielectric in a cathode-ray tube but, since the tube electron beam swept a complete line at a time, the information was available in time sequence only.

The second type is parallel storage, in which **the basic element stores a single pulse.** An array of elements is needed to store a complete number, but these elements may be read simultaneously, resulting in a complete number in space rather than in time sequence. The RCA Selectron and the position-type electrostatic storage tubes being developed by this project are of the parallel type, as are also the more familiar storage elements such as flip-flop registers, mechanical counters, and relay banks.

As will be discussed below, the serial or parallel nature of the storage will determine the general nature of the computer, the kinds of components, the general speed and complexity, and the nature of the orders and switching.

Other storage characteristics, not in order of importance, are:

1) Size and cost in the light of desired computer capacity.
2) Switching complexity.
3) Erasing and restoring methods.
4) Time required for storing and reading.
5) Reliability and ease of checking.
6) Most suitable number base.

According to present plans, the Whirlwind electrostatic storage will have the following characteristics:

1) It will be parallel in nature.
2) Storage will be represented by one of two possible charge conditions on a dielectric plate. The base 2, or binary system, is therefore most suitable.
3) Storage capacity will be 1024 binary digits per tube. Thirty-two tubes will suffice for WWI and 640 will suffice for WWII.
4) The tubes will be arranged in banks, each bank containing as many tubes as there are digits in the number length of the machine. The number of banks is determined by the number capacity of the machine. Two 32-way switches are needed to select the deflection voltages in the two co-ordinates of the tubes, plus a switch to select the desired storage bank.
5) Approximately five microseconds will be required for setting up and reading or storing, according to the best present estimates.
6) It will not be necessary to order the erasing or restoring of stored information. Information will be automatically maintained until replaced by new information. When replaced, old information will be automatically erased.
7) The storage is susceptible of complete checking methods.

The storage will not be discussed in detail in this report. The above characteristics are given because they affect computer design.

### 2.2 Parallel Transmission of Digits

Parallel digit transmission has been selected for the Whirlwind Computer.

Since the storage proposed for the Whirlwind Computers is of the parallel type, it is natural to propose parallel handling of digits throughout the machine. Parallel digit transmission has other advantages where high computing speeds are desired.

Maximum operating speeds of basic computer components are established by video bandwidth and time delays which are measured in fractions of a microsecond. When computing speed requirements are low, the serial digit transmission system requires a smaller ratio of control-to-computing components and requires few types of circuits. As speed is increased, the serial digit transmission computer must make use of simultaneous number transfers and arithmetic operations. For high speed the resulting serial system is much more complicated and harder to program than the parallel system selected for the Whirlwind Computers.

Block diagrams for both serial and parallel machines have been developed for comparison purposes.

## 2.3 Arithmetic Considerations

While the most important choices to be made are type of storage and serial or parallel transmission of digits, there are several arithmetic considerations of importance. These considerations include the choice of a number base, means for handling decimal points, and means for handling negative numbers.

### 2.31 The Binary Number Base

The Whirlwind Computers will use the binary base for computing and storing numbers.

A computer can be built to use numbers written in any base. The base 10 is convenient because it is the customary notation in which people are taught to compute and for which most existing computing machines are built.

The base 10 is not particularly satisfactory for electronic computing machines, however, because the storage of numbers in this base requires a storage element capable of 10 separate states or conditions. In the interest of reliability, it is preferable to use electronic equipment as though it had only two stable states, such as conducting or cutoff for a tube and charged or discharged for a condenser. The base 10 requires several such elements for storing each digit, and the complexity of the computing equipment is much increased.

The base 2 is well suited to electronic computers because the storage elements require only two separate states, which are available in conservatively operated electronic equipment. The difficulties arising because the machine and its operators use different bases are overcome by providing automatic conversion devices which allow the machine to receive decimal data and to yield decimal results.

### 2.32 Fixed Point

The fixed point system will be used in the Whirlwind Computers.

Any digital computing machine is faced with a scale factor problem. The computer registers have a finite capacity, while the numbers they may be called on to hold have, at least theoretically, an infinite range.

There are in general two methods for solving this problem:

1) The floating-point system. Associated with each number in the machine is a scale factor. This scale factor is stored in the machine and is changed during the computation so as to keep the significant figures of the number within the register. Since the storage space which can be allocated to storing the scale factor is limited, the range of numbers is also limited, but the range can be wide enough to cover almost any expected case. It is necessary for the operator to give only secondary attention to scale factors.

2) The fixed-point system. A scale factor is still associated with each number in the machine, but this factor is assigned by the operator setting up the problem and is not changed during the course of a problem. It is not necessary to store the scale factor or to provide the equipment for computing its changes. The range of numbers which can be handled is dependent on the register length and is limited for any given problem. Special steps can be taken to care for those cases where the normal range is inadequate.

Although it is not in line with the general principles of high-speed computers to reduce the work the machine must do at the expense of greater trouble to the operator, the advantages of the floating-point system are outweighed by the extra complexity of the computing equipment and the reduced computing speed.

### 2.33 Signs and Negative Numbers

The Whirlwind Computers will use the 9's complement system for representing negative numbers.

An arithmetic computer must be able to subtract and to use the algebraic signs of numbers. There are many ways of accomplishing these ends,

but the three most common are:

1) Absolute value and sign
2) 10's complements
3) 9's complements

The absolute value and sign method is that customarily used in hand computation. The human computer adds or subtracts according to the signs of the numbers under consideration or appends the proper sign to products and quotients. The computer must be able to subtract as well as add.

The 10's complement method makes it unnecessary to build into the equipment the ability to subtract. A negative number is represented by the difference between the absolute value of the number and some higher power of the base. The base in any system is commonly written as one-zero, or 10 (called ten), hence the name 10's (or tens) complement. Cologarithms are an example of a 10's complement notation in the decimal system.

Subtraction in the 10's complement system becomes the addition of a complement, the actual subtraction having taken place when the complement was formed. The advantage of the system is that the subtraction process required for forming a complement is essentially simple and easily mechanized. For example, in the binary system a complement may be obtained by inverting the original number, that is, interchanging 0's and 1's, and adding a 1 in the right-most place.

The 9's complement system is identical with the 10's complement system except that the 1 mentioned above is not added to the right-most place of the inverted number. Complementing becomes an extremely fast and simple operation, since the addition operation has been avoided and it is necessary only to invert 0's and 1's. The use of 9's complements is most satisfactory when a parallel digit transmission system is used. The use of 9's complements has some numerical disadvantages which, however, seem to be outweighed by physical advantages.

## 2.4 Switching Considerations

With a parallel type system such as the one used in the Whirlwind Computers, a single bus capable of transmitting a complete number becomes a group of coaxial cables, 16 in WWI and approximately 40 in WWII. The switching of a number becomes, then, the switching of these multiple lines. Provision of more than one multiple bus requires the addition of considerable equipment.

For these reasons, only one main bus system is to be provided for the Whirlwind Computers, and switching is kept to a minimum; namely, connecting or disconnecting each element to the bus. As discussed above, it is preferable to use parallel elements, one element per digit, and to perform computing steps in sequence rather than to use serial elements and perform many computing steps in parallel. In line with this thought, the switching will be done sequentially. Only one number will be transmitted at any one time.

The basic system is shown in Figure 1. A single bus of parallel conductors is provided to which are connected the several elements of the computer; namely, the Storage, the Control, the Arithmetic Element, the Input, and the Output. These elements will be discussed in greater detail in later sections.

## 2.5 An Order Code

The orders to be provided cannot be completely defined and discussed until the entire computer system has been described in some detail. General considerations will be given here.

It is of first importance to realize the nature of the orders within the computer. The characteristics of these orders are:

1) They are binary coded numbers which do not differ in appearance in any way from the binary numbers used by the computer.

2) These orders are stored in the computer storage in exactly the same manner as numbers.

3) There are no restrictions on which of the storage registers may be used for orders and which for numbers. All registers in storage are controlled by a single switch and transmit to and receive from a single place, the main bus.

4) Since the contents of any register may be sent to the bus, the word held by any register may be received by the control and treated as an order. The operator in setting up the program determines which words are to be used as orders. The arrangement of storage is completely flexible. The amount allocated to orders and the amount allocated to numbers may be adjusted to suit the requirements of any particular problem.

5) An order may be sent to the Arithmetic Element for actual arithmetic modification. By this means a sequence of similar operations may be carried out by a single set of orders continually modified by the addition or subtraction of suitable constants. This ability of the computer is very useful for generalized order sequences where a standard set of orders is to be used in many similar problems.

### 2.51 The Standard Order

When switching was considered in Section 2.4 above, it was stated that the switching of only one number at a time was to be ordered. It is furthermore true that, as far as numbers are concerned, they are always transferred to or from storage. Each order then should describe the storage register to which or from which the number is to be transferred, plus a description of the element which is to be the source or destination for the number, as well as a description of any further operations to be carried out, particularly in the Arithmetic Element.

Each Whirlwind order is to be broken into two parts:

1) The number of the storage register which is to be connected to the bus.

2) A code number designating one of a group of computing operations. This code number can be decoded by the control, which will then carry out the corresponding operation.

The number of digits required in an order is an important consideration in choosing the register length to be provided in the computer. The register number section of the order must be large enough to describe the maximum storage capacity of the computer. WWI has a capacity of 2,048 or $2^{11}$ registers. Eleven binary digits must be provided in the register number section of the order. Thirty-two ($2^5$) possible operations will be provided, requiring 5 binary digits for the operation code section. The total order length is 16 binary digits, which has been chosen as the register length for WWI.

WWII will have a storage capacity of 16,384 or $2^{14}$ words. Sixty-four ($2^6$) operations will be provided. The total order length will be 20 binary digits. Since this length is too short for numerical values in a computer of WWII capacity, two orders will be stored in a single register. The resulting register length of 40 binary digits will probably be increased to 44 or 46 to allow for order expansion at some later date.

### 2.52 Desired Operations

The number and kind of operations to be used in the operation code section of the order are determined by what the computer is to do. It is possible to build up all the desired operations out of some very elementary operations. These elementary operations may be described in the terms of the algebra of logic, but consist essentially of sensing the presence or absence of single digits or the coincidence of two digits. If the simplest imaginable computer were to be built, it might contain only these few operations. Unfortunately, several hundred elementary operations would be required to build up even a single addition of two complete binary numbers, since the addition of each digit and the addition of each carry would require separate ordering. The total number of operations for a sizable problem would be enormous. The time consumed would be so great as to make such a computer useless.

It is desirable to build into the computer the ability to perform automatically the elementary arithmetic operations in order to speed up the computing process. The extent and complexity of the operations to be built in are determined by the price to be paid in equipment for each new operation and the saving in computing speed effected. The Whirlwind Computers will have the ability to add, subtract (using complements), and multiply automatically.

The Whirlwind Computers will also be able to divide and to shift the position of a number in a register to either the right or the left. The amount of additional equipment necessary for these latter operations is small for computers of the parallel type. It is possible to omit these operations, substituting for them procedures using only addition, subtraction, and multiplication.

No further arithmetic operations will be built in. Such processes as square-rooting will be built up out of the more elementary operations.

Beyond the elementary arithmetic operations, certain control operations are of extreme importance. These operations allow great savings in the number of orders required for a problem, allow the machine to exercise a certain amount of program generation in that it can make changes

in its computing procedure according to computed results, and allow the computer to perform such operations as sorting and extraction of data from tables.

These control operations are:

1) Subprogramming. In this operation the control is ordered to stop the sequence of orders it is performing and to change to some other sequence stored within the machine.

2) Conditional Subprogramming. Here the change in the order sequence is made to depend on some previously computed result, most easily on the sign of a number. Any decision the machine may be called on to make may be reduced to the question of whether or not one or more of a sequence of numbers is positive or negative.

3) Digit Transfers. This operation allows the insertion of computed data in actual orders. Usually the insertion is restricted to insertion of a computed storage register number, for example, where an entry is to be removed from a table. The register number of the desired entry is computed and inserted in the order for removing the entry.

Before the computer can be used there must be some means of inserting data and extracting results. Film readers and recorders will be used for this purpose, the film serving as an intermediate and easily handled storage medium that is intelligible to both the computer and to external printers. Such a storage medium is often called an external memory. These readers and recorders must be under the control of the computer so that new information may be ordered in when desired and results recorded when available. A series of order operations must be available for this purpose.

At present the input and output devices and their controlling orders are still under consideration. The accompanying block diagrams include no input mechanisms and only a single very simple output, for test purposes, the control thus far worked out being limited to a single order to send a result to this output.

The complete input and output mechanism for WWI will be reported later. The mechanism for WWII will probably be somewhat more complicated because of the large number of connections to the controls and instruments required in simulation problems.

Certain additional operations will be desirable for the Whirlwind Computers. These operations will render more efficient processes such as interpolation and multiple-length number operations. Of these, only a single order, used for addition of multiple-length numbers, is included in this report. The total number of these operations will be small, probably not over six.

### 2.53 An Order Code

The following paragraphs define those control orders for WWI which have been established thus far. All operations are represented by two lower-case letters of the English alphabet chosen to be descriptive of the process. The following definitions are used:

**Symbols**

| | |
|---|---|
| AC | "Accumulator" register of Arithmetic Element |
| AR | "A-Register" of Arithmetic Element |
| BR | "B-Register" of Arithmetic Element |
| S(x) | The contents of Storage Register x, where x is the identifying number of the storage register to be used in the computing operation. |

| Operation Code Designation | Description | Meaning |
|---|---|---|
| *ca* | Clear and add | Clear AC and add the contents of register x into it. |

This operation is used for clearing the Arithmetic Element and inserting a new number. The order designating the operation to be performed, addition, multiplication, etc., is usually the one that brings in the second of the pair of numbers to be used. In multiplication, a *ca* order will bring in the multiplier, while a second order will bring in the multiplicand and order the multiplication operation. A *ca* order is also needed for the transfer of a word from one storage register to another, since the only means provided for this operation is to send the information first to the Arithmetic Element and to restore it from there.

| Operation Code Designation | Description | Meaning |
|---|---|---|
| *ad* | Add | Add the contents of register x to whatever is already in AC. |

This operation is used for simple addition. The contents of AC may be the result of a previous calculation or may have been brought into AC by a *ca* order.

| | | |
|---|---|---|
| *cs* | Clear and Subtract | Clear AC and subtract the contents of register x into it. |

This operation puts the negative of a number into AC. It is of value when the negative of a sum or a product is desired.

| | | |
|---|---|---|
| *su* | Subtract | Subtract the contents of register x from whatever is already in AC. |

This operation is used for simple subtraction. It is identical with order *ad* except that the complement of the number in register S(x) is added to AC.

| | | |
|---|---|---|
| *mr* | Multiply and round off | Multiply the contents of register x by whatever is in AC and round off the result to one register length. |

The contents of AC are automatically transferred to BR by this order. There they form the multiplier. The double-length product which results in AC and BR is rounded off to the last digit in AC. The contents of BR are not significant and are discarded.

| | | |
|---|---|---|
| *mh* | Multiply and hold full product. | Multiply the contents of register x by whatever is in AC but do not round off. |

This operation is identical with *mr* except that the full double-length product is retained in AC and BR. It is necessary to retain the less significant half when computing with numbers which are twice a register length or when some of the digits in BR are significant and are to be saved.

| | | |
|---|---|---|
| *dv* | Divide | Divide the contents of AC by whatever is in register x. |

This order performs division. The quotient will appear in BR, from which it must be extracted by shifting left into AC. The shifting left will require a separate order. Roundoff will be performed in the Shift-left operation.

| | | |
|---|---|---|
| *ts* | Transfer to Storage | Transfer the contents of AC to register x. |

This operation is used for storing computed results which are to be used in the future. In conjunction with a *ca* order, a *ts* order may be used for transferring information from one storage register to another.

A simple arithmetic operation such as addition may require three orders, a *ca* order to bring in one of the numbers to be added, an *ad* order for the second, and a *ts* order to store the result. In many instances, however, a computed result may be used immediately, eliminating the two orders required for storing it and for bringing it forth at a later time.

| Operation Code Designation | Description | Meaning |
|---|---|---|
| *sr* | Shift right | Shift the contents of AC and BR to the right the number of digits designated by the number in the register number section of the order. |

The shift operations are the only ones in which the first 11 digits of the order are not used to designate a register position. No register position is needed since the numbers concerned are already in AC and BR. Only four of the register position digits are needed to describe the number of places to be shifted. Shifting right is equivalent to dividing by powers of 2.

| | | |
|---|---|---|
| *sl* | Shift left | Shift the contents of AC and BR to the left the number of digits designated by the number in the register number section of the order. |

*sl* is the same as *sr* except that the direction of shift is to the left. Shifting left is equivalent to multiplying by powers of 2. *sl* is also used for extracting information from BR.

| | | |
|---|---|---|
| *sp* | Subprogram | Transfer the register number x to the program counter. |

The *sp* or subprogram order is used to transfer the place in storage where the control is extracting orders. Its uses are in calling up smaller sequences which are used more than once in a longer sequence, in calling in special sequences stored in standard sections of the Storage, and in returning to the beginning of subsequences previously used. The mechanization of the *sp* and *cp* orders is given in Section 3.1.

| | | |
|---|---|---|
| *cp* | Conditional Program | Transfer the register number x to the program counter if the number in AC is greater than 0. |

The change in program is made to depend on the sign of the number in AC. If this number is greater than zero, the control proceeds as for an *sp* order. If the number in AC is equal to or less than zero, the control will ignore the order and proceed in the previous sequence.

| | | |
|---|---|---|
| *td* | Transfer digits | Transfer the left-hand 11 digits in AC to the register position section of the order in x. |

This order allows insertion of a computed register position in an order in Storage. *td* is valuable for modifying programs to fit special problems. A standard program may thus be modified to fit the special requirements of each problem that arises, or a program for interpolation may be modified to extract the desired information from Storage according to a computed argument.

| Operation Code Designation | Description | Meaning |
|---|---|---|
| *sa* | Special add | Add as in *ad* but retain any overflow for use in double-length number addition. |

This order allows easy addition of mutiple-length numbers. Any carry to be added from one section to another is stored by this operation and added in correctly on the next *ca* order. See Section 4.614 for a complete description of Special add.

In addition to the orders described above, the following will also be needed for the WWI computers:

1) Absolute value operations in which arithmetic operations are carried out on the absolute values of the numbers concerned.

2) Special orders to facilitate computing with numbers longer than one register length.

3) Input and output orders for controlling the supply of data to the machine and the extraction of results.

A simple output order for test purposes follows:

| | | |
|---|---|---|
| *sd* | Store and Display | Transfer the contents of AC to register x and also to the output element. |

This order is to be used in WWI with test storage only for the display of results on a cathode-ray tube. A more flexible set of output orders will be provided for WWI with electrostatic storage.

## 2.6 Word Length

### 2.61 WWI

It is difficult to decide the number of digits to be provided in a general-purpose computer. Some problems require large number lengths to get a relatively inaccurate result. Other problems require but a few digits. For problems of the latter type a large register capacity would be very inefficiently used.

Estimates of desirable register lengths vary from a minimum of about 10 decimal digits to a maximum of 20, 30, or more decimal digits. The decision must be made on the basis of the type of machine, effect of register length on computing speed, size and cost of equipment (particularly storage), types of problems to be considered, necessary order length, and possibilities for computing with double and half-length numbers.

For WWI the shortest reasonable word length was chosen to reduce machine size. This length was determined by the minimum order length, and turned out to be 16 binary digits or about 5 decimal digits. A shorter register would require shortening of the order either at the expense of the number of possible operations or the number of possible storage registers. Since these numbers were selected as the minimum desirable for the purposes of the computer, the minimum word length was immediately fixed.

### 2.62 Computing with Multiple-Length Numbers

Five decimal digits are inadequate for many problems, although they will suffice for a large number of the investigations and problems for which WWI will be used. Provision will be made for computing with numbers of 10, 15, or more decimal digits stored and operated on in 5-digit sections. Coding will be arranged so that the operator setting up the problem, having once decided on the number of digits to be used, will not need to consider the sub-operations needed for each arithmetic step.

The time required for handling numbers in small sections is comparatively great. Since WWI will have a basic operating speed considerably in excess of what might be considered necessary except for test purposes, the resulting speed is still high.

When multiple-length numbers are used, the storage capacity of the machine is reduced accordingly. The use of double-length numbers does not cut the capacity in half, however, because by the

use of subprograms to order the more complicated arithmetic processes, the number of orders required is not appreciably increased.

### 2.63 WWII

As stated above, an order will occupy a complete register in WWI. The desire for greater effective speed and capacity in WWII requires a register with more digits. The orders will be slightly longer for WWII than for WWI, probably 20 to 22 binary digits instead of 16. The added capacity if one order still occupies a register is inadequate. If the register length is increased, the extra length is wasted as far as order capacity is concerned unless the increase is great enough to allow the storing of more than one order in a single register. If, then, it is desired to utilize the register capacity fully in storing orders, the register length must be a multiple of the order length, or a multiple of approximately 20 binary digits.

Forty digits have been decided upon as adequate for WWII purposes. Forty binary digits are equivalent to about 12 decimal digits. The jump to 60 binary digits does not seem of sufficient value to make up for a 50% increase in equipment and a small reduction in computing speed.

The storage of even two orders in one register for WWII will increase the complexity of the control because the machine will have to differentiate between the two halves.

## 2.7 Input and Output

The input and output devices were mentioned in Section 2.52. Their detailed discussion will be postponed until a later report.

## 2.8 Checking

### 2.81 General Considerations

In any sort of calculations the need for checking is immediately evident. A body of results is worthless unless there is some guarantee as to freedom from errors. This is particularly true for computing machines where the amount of calculation may be enormous and the slightest mistake anywhere may destroy the answer.

Furthermore, the machine must do the checking by itself. First, the body of computation will, in all likelihood, be too large to be checked efficiently by any means other than the machine itself. Second, the purpose of the machine would be defeated to a large extent if any appreciable computing labor which could be performed by the machine were forced upon the operator. Third, the machine must be able to make checks during the course of a calculation which will not only discover an error when made and prevent other results from being disturbed, but which will also aid greatly in trouble-shooting and repair.

The checking methods to be applied by the machine fall into three types:

1) Running checks.—These checks are made at each step of each operation. Wherever possible, every transfer of a digit and every switching or sequencing operation is to be checked at the time the transfer, switching, or sequencing is accomplished. In many places it is not possible to check every step. In these places, the steps will be checked in groups of operations, either with the original equipment or with duplicate equipment provided for the purpose. These checks will find intermittent as well as steady errors.

2) Spot checks.—These checks will be of value only for finding steady failures. They will consist largely of previously set up check sequences and complete check problems giving known results. Some of these checks may be performed automatically by the machine while others may be programmed by the operator.

3) Mathematical checks.—These checks will be smoothness checks on results, checks by repeating problems using different mathematical methods, and such other mathematical checks as are suited to the particular problem being considered. The machine will carry out the calculations necessary to the check according to a program prepared by the operator.

Only a few of the above checks have been worked out; these are shown in the set of block diagrams. The problem of checking will be considered in more detail before construction of WWI.

### 2.82 Arithmetic Check

It is possible for the operator as well as the machine to make mistakes. The major part of the operator's errors will be checked at the input devices and by the use of sample problems. One error that the operator may make is to select improper scale factors. As discussed in Section 2.32 above, the scale factors for the problem must be set by the operator so as to be consistent and so as to keep every number the machine must consider within

the bounds imposed by the finite register length of the machine. Since it may be difficult to place a reasonable upper bound on some of the partial results of a complicated problem, the operator may make the error of allowing some number to exceed or "overflow" a register. Under these circumstances, the machine must realize the overflow and stop the computation so that a correction can be made; otherwise, the value of the computation will be destroyed. Such a check will be part of WWI. It is called the arithmetic check and is described in Section 4.62.

### 2.83 Transfer Check

The computer contains a single bus system which is used for transmitting all numbers and orders from one element of the computer to another. The elements are connected to the bus by sets of gate tubes which are opened or closed according to whether the element is sending, receiving, or not involved in the operation. Ever present sources of error are the failure of one of these gate tubes to transmit a digit, the loss of a digit due to an intermittent or permanent failure of the bus, or the generation of a spurious pulse at some point.

For these reasons every transfer on the bus is to be checked. The check is accomplished by reading the word from the receiving element back into the bus to a special check register. This register has already received the original word from the sending element via an entirely different bus and gate tube system. The two words are compared. An error made in transmitting either word to the check register will stop further operations.

### 2.84 Other Checks

Work is proceeding on the other types of checking briefly described in Section 2.81.

## 2.9 Required Computing Speed

### 2.91 WWII

Since the computing speed of WWI will be made comparable to that needed in WWII, the required speed of WWII will be considered first.

Consideration of the speed requirements for some of the simulator problems proposed for WWII leads to the following performance specifications.

1) A storage read time (i.e., the time required to obtain a number or order from storage) of approximately six microseconds.

2) A complete multiplication time including all ordering, extracting of numbers from storage, and storing of results of about forty microseconds. Such a multiplication speed requires a 4-megacycle Arithmetic Element operating speed.

3) Addition and subtraction times of not much less than multiplication, probably about thirty microseconds.

4) The requirements are such that computing speed is actually at a premium. It is worth while to provide a considerable amount of increased equipment and complexity in order to increase operating speed.

5) For speeds such as those above, the storage read time is the controlling factor. There is a definite limit to the increase in computing speed that can be obtained by speeding up the arithmetic part of the computation.

The system to be described in the next sections has been designed with these considerations in mind.

### 2.92 WWI

WWI is a prototype for WWII, and the speed requirements are therefore those discussed in Section 2.91. In order to reduce development time, the Arithmetic Element may be run at 2 megacycles instead of the 4 megacycles needed for WWII.

### 2.93 Storage Read Time

As discussed in 2.91, the read time for the electrostatic storage is really the controlling factor in the speed of both WWI and WWII.

The present estimates of storage read time have not been completely substantiated as yet. Any required reading time can be provided in WWI by means of the Delay Counter in the Control. See Section 5.27.

### 2.94 Computer PRF

The computer pulse repetition frequency, PRF, should be as high as is reasonably possible in view of the need for maximum computing speed. The meaning of *reasonably possible* is a problem of electronic research, and is determined by the operating speeds of presently available flip-flops and switches and the problems of transmitting short pulses and ensuring their coincidence at widely separated parts of the equipment.

At present the computer will use two PRF's:

1) In the Arithmetic Elements, 4 mc. (possibly 2 mc. in WWI).

2) Elsewhere in the computer, 1 mc

# 3. A GENERAL OUTLINE OF THE SYSTEM

## 3.1 System Build Up

The most general form of the Whirlwind Computers is shown in Figure 1. The purposes and characteristics of the various elements have been discussed in Section 2.

Figure 2 shows WWI in more complete detail. The explanation of each element can best be approached by building up the system unit by unit. Later, the various elements will be discussed in detail.

Consider Figure 3. Every operation must be ordered. It is necessary, therefore, to have a supply of orders within the machine. These orders are kept in the element marked Storage, which consists of a number of storage places called registers. In order to facilitate obtaining the contents of these registers, they are numbered consecutively from one up to the limit of storage capacity. Thus, each order will be put in some numbered register and can be obtained by having the Storage read out the contents of the register having this number.

Some means is needed for remembering in what sequence these orders are to be carried out. There are several methods available. It is possible to have each order contain information as to where in Storage the next order is to be found. This method has certain advantages but is wasteful of storage space. Instead, consider the fact that orders are usually used in consecutive sequence. This is not always so; in fact, flexibility in the choice of what order is to be used is a very important quality of a computer, but usually orders are used one after the other. If, therefore, the orders are stored in the desired sequence in adjacent storage registers, they can be obtained by reading out of these registers in consecutive order.

A counter is provided which is used to store the register number of the next order to be used. Normally, after each order is extracted and carried out, a 1 can be added to the counter. The next order to be extracted will then be the one which follows in the storage sequence. It must be possible to change this sequence when desired. Changing the number in the counter will be considered later.

Figure 4 shows how an order is set up. The counter contains the number of the register which holds the desired order. This register number is read out onto the bus through a set of gate tubes *GT*. A switch attached to the storage is used to pick the desired storage register. The order number travels down the bus and through another set of gate tubes to set this switch. The desired storage register has now been chosen.

Figure 5 shows the next step, extracting the order from Storage. The gate tubes on the Storage are opened and the order read out onto the bus. A problem then arises as to what to do with the order. As discussed in Section 2.51, the orders consist of two sections. One section is the operation order, which could now be sent directly to the Operation Control. The other section is the storage register number which, since its purpose is to choose a number from Storage, should be sent to the Storage Switch. It is possible to keep orders and numbers in two different storages, but to do so would reduce the flexibility of the machine, with few compensating advantages. It is preferable to have one storage which holds both orders and numbers and to allocate such sections of that storage as are desirable to each use. The Storage Switch has already been set to choose the register holding the order. The order cannot be sent directly to this switch, since no time is available for clearing and preparing it for the reception of a new switch order.

A new register, the PR or Program Register, is provided for storing the order until the Storage Switch can be cleared. The same bus is used for the transfer. The gate tubes on the switch are not opened.

Figure 6 shows how the operation is set up. The Storage Switch has been cleared. The order is read from PR onto the bus and from the bus into the Storage Switch and Operation Control Switch. The Operation Control Switch decodes the operation order and sets the Operation Control for the desired operation. The Storage is set up to connect a particular register to the bus either for sending or receiving.

A complete arithmetic operation such as an addition requires three complete orders: two to send the numbers to be added to the Arithmetic Element, and one to store the result. It is possible to reduce this number if a result is to be used im-

mediately, thus saving storing the result and then bringing it back to the Arithmetic Element for the next operation. In general, the first order will transmit one number to be operated on to the Arithmetic Element. Figure 7 shows such an operation. The storage gate tubes are opened and the desired number read out onto the bus. The input gate tubes of the Arithmetic Element are also opened, allowing the number to go into the Arithmetic Element. Some additional operations such as clearing may be needed within the Arithmetic Element. The Operation Control will supply the necessary order pulses.

The next order will put the second number into the Arithmetic Element and order the operation to be performed. The entire sequence so far described must be gone through for each order:

a. One is added to the counter.
b. The counter contents are read into the Storage Switch via the bus.
c. The order is read out into the Program Register, and
d. from there to the Storage and Operation Control Switches.
e. The new number is read from Storage to the Arithmetic Element, where the desired operation is carried out under the control of the Operation Control.

Figure 2 shows the storage of an arithmetic result. The output gate tubes of the Arithmetic Element are opened, and the result transmitted to the bus and into storage via the storage input gate tubes.

The subprogram operations are shown in Figure 8. The order has been extracted from Storage and stored in PR. From PR the order has been transferred to the Storage and Operation Control Switches. As with the shift operations, the Storage is set up but not read. The Operation Control transfers the register position section of the order in PR to the counter. The next order to be taken will be the one residing in the register designated by the subprogram order.

The conditional subprogram order is shown in the same Figure 8. The Operation Control sends a pulse to the Arithmetic Element. If the sign of the number in the Accumulator is negative as evidenced by a one in the sign digit position, the pulse is stopped and nothing further occurs. The next order in sequence as given by the counter will be taken, and the computation proceeds. If the number in the Accumulator is positive, the pulse will be returned to the Operation Control Switch, where it will change the switch setting from conditional subprogram to subprogram. The register position section of the order in PR will be transferred to the counter as with the subprogram order, and the order sequence will be changed.

The basic elements of the system will now be discussed.

## 3.2 Storage Characteristics

No electrostatic storage is shown in the accompanying block diagrams. Additions to these diagrams will be released as additional details are established.

Provision is being made in WWI for 2,048 electrostatic storage registers of 16 binary digits each. Each storage tube is expected to store a 32x32 array of digits, a total of 1,024. Sixteen such tubes in parallel will store 1,024 16-digit words. Provision is being made for two such banks, a total of 32 storage tubes, 2,048 words, or 32,768 binary digits.

The addition of electrostatic storage should not significantly change the present block diagrams for the remainder of the computer. A separate control will probably be provided for the electrostatic storage for the following reasons:

1) Such a separate control for electrostatic storage is likely to prove saving in both equipment and complication as compared to combined controls.

2) The separate control can be designed later to fit the as-yet-undefined characteristics of the storage without requiring rebuilding of the presently proposed control. The separate control may also be more readily changed in the event of changes in the storage than would be possible with a single control.

## 3.3 Control Characteristics

### 3.31 Clock

A primary requirement for the computer is a master clock or source of timing pulses. These pulses are used for keeping the computer in synchronism, for timing operations, as a pulse source for originating control pulses, and for step-by-step operation in trouble shooting.

The operation of a computer of this type is

a step-by-step process rather than a continuous one. This concept holds from the simplest part of an operation through to the computation of complete problems. Each operation is made up of a number of sub-steps, each sub-step consisting of a single transfer, shift, or check. A number of these sub-steps may be performed by different parts of the equipment at the same instant of time, but each part is subject to only one such operation for a given sub-step.

The clock puts out a pulse which, in essence, orders the computer to perform a single sub-step. The computer will do so and then await the order to perform the next sub-step. The clock is used to order these sub-steps at the highest rate possible compatible with the operating speed of the equipment. A lower speed would not affect the computer operation or results but only the time required for a computation. A manual control is provided which causes the clock to put out a single clock pulse. By this means, the computer may be operated at low speed for trouble shooting.

The clock provides pulses at two different frequencies. The higher frequency is supplied to the multiply and shift controls in the Arithmetic Element while the lower frequency is supplied to the rest of the computer controls. The higher frequency is derived from the lower by frequency multiplication. On single-pulse operation the clock will put out a pulse on both the high- and low-frequency lines. The additional pulses which result on the low-frequency line will have no effect.

Provision will be made for starting or stopping the flow of pulses from the clock. This is useful in starting and stopping the computer, and in stopping a computation when an error has been sensed or some external device has fallen behind in time.

### 3.32 The Program Counter

The counter should have the following characteristics:

1) Range—The counter should be able to count as high as the total register capacity of the machine. For WWI with electrostatic storage this capacity is 2,048 registers. Eleven binary digits are needed to describe 2,048 registers. An 11-section scale-of-two counter will be used.

2) Counting Rate—The counting rate is relatively unimportant since the counter need add only once for each operation completed. The time per operation will be of the order of 10 μs. The net counting can be as low as 100 kc. Actually, the counting rate will be high to give more leeway in positioning the adding pulse in the operation cycle.

3) Reading and Changing—The counter must be able to read onto the bus. Provision must also be made for clearing the counter and reading in a new number from the bus, as described in Section 3.1.

4) End Pulse—When the counter is full, the next pulse will return it to zero. When returning to zero, the counter will put out an end pulse which will signify the completion of the sequence of orders. This pulse can be used for test purposes.

### 3.33 Time Pulse Distributor

Thus far the mechanism by which the Control orders the opening and closing of gates and the transferring of numbers and information in the machine has not been discussed. This ordering can be accomplished by supplying pulses to the equipment at the right places and at the right times. The pulses supplied for setting up the order and operation are independent of the operation which is to follow. The pulses supplied in ordering the actual operation are dependent upon the Operation Control Switch setting. The Operation Control Switch determines distribution of pulses to different lines at different specified times during the course of a computation period for control of computer operation. A different sequence of pulses must be available for each operation to be performed.

The basic device needed for computer control is an electronic switch. Such a switch is shown in Fig. 9. This switch, which is a simplified version of the types to be used in the actual computer, consists of a number of flip-flop circuits whose plates are connected to B+ through a diode matrix. The switch shown contains only two flip-flops. As each of these flip-flops has two stable states, the switch has four possible positions. By adding more flip-flops, more positions may be obtained. The number of positions=$2^n$, where n=number of flip-flops.

The switch operates as follows. Consider the conditions under which only the right-hand tubes of the flip-flops are on. These tubes will be

drawing plate current, while the left-hand tubes of the flip-flops will be cut off and will be drawing no plate current. The tubes draw current through the diodes from B+ via the load resistors at the left-hand side of the diagram. Examination of the diagram shows that plate current is being drawn through only the three lower resistors. The top line of the switch is connected only to the left-hand tubes of the flip-flop. No current is drawn through the upper resistor, and therefore the upper line will remain at B+. The lower three lines will be at a potential less than B+ by the drop in the load resistors. The upper line, therefore, is at a positive potential with respect to the lower three, and can be used for controlling a gate tube. Suppose the condition of one of the flip-flops is changed. Let the left-hand tube of the left-hand flip-flop draw current while the right-hand tube of the left-hand flip-flop is cut off. Plate current will be drawn through the upper resistor by the left-hand tube, lowering the voltage of the upper line. It will be seen, however, that the third line is not connected to any of the "on" tubes. No current will be drawn through the corresponding resistor, and line $C_3$ will be at B+. Each of the four possible positions of the tube flip-flops results in a different output line being at B+. Diodes are used to prevent voltage from feeding back to the "on" line and to keep all the "off" lines at the same potential.

Figure 10 shows how a matrix switch can be used to make a time pulse distributor which will produce consecutive time pulses on the output lines. The distributor consists of an eight-way diode matrix switch, such as has just been described. The eight outputs are supplied to two-grid gate tubes, while the flip-flops driving the switch are connected together in a counter circuit. Suppose now that all the flip-flops are in their off condition and that the corresponding condition of the switch is such that the uppermost gate tube is on. A clock pulse comes in from the left, goes down the connecting line, and up to the grids of all the gate tubes. Since the upper gate tube is on, a pulse which is inverted by the transformer coupling will appear at the output and will go out as a positive pulse on the first line. All the other gate tubes will be off and no pulses will appear on the remaining seven lines. After the pulse has come out of the upper gate tube, it will finish its travel through the delay line at the lower left and will go in and change the condition of the first flip-flop. This will change the switch position so that the upper gate tube will go off and the second gate tube will go on. The appearance of the next clock pulse therefore will result in a pulse on the second line and no pulses on any other lines. The second pulse will change the first flip-flop back to a zero, the overflow from this flip-flop tripping the second flip-flop. In like manner each gate tube in turn is opened so that the corresponding clock pulses will come out in consecutive order, starting with the top line and finishing with the bottom line. After the last pulse appears, the switch will be reset, and the next pulse will appear at the top line.

The distributor is receiving a continuous string of clock pulses and dividing them up so that they appear in consecutive order on the output lines. A complete operation is considered to be started when the first clock pulse appears out of the time pulse distributor.

### 3.34 Operation Control

Figure 11 shows how a time pulse distributor can be used in conjunction with the Operation Control Switch to order the different operations. The Time Pulse Distributor is the same as that just described, the pulses appearing in consecutive order on lines one through eight. The 32-position switch is the Operation Control Switch which is supplied with the operation control part of the order. Only one of the thirty-two possible outputs is selected for any given control order.

The row of gate tubes is used to connect the time pulses to the various parts of the equipment. These tubes are selected according to the position of the Operation Control Switch.

Consider the left-most gate tube. It is connected to the *ca* or Clear-and-Add order line of the Operation Switch, and to this line only. Therefore, whenever a *ca* operation is to be carried out, this gate tube will be on and the No. 1 time pulse from the Time Pulse Distributor will go out the control line to the Arithmetic Element. In any other operation, this gate tube will be off and no time pulse will go out. The next pair of gate tubes have their outputs connected together and to one particular part of the equipment. However, they supply time pulses at different times according to the operation to be carried out. The left-hand gate tube is connected to the *ca* line, while the right-hand gate tube is connected to the *cs* line. Therefore, when a *ca* operation is to be carried out, a time pulse will appear at time 2 on this line, while when the *cs* op-

eration is being carried out, a time pulse will appear at time 3. For all other operations, no time pulse will appear. The fourth gate tube is connected to two of the other operation lines, the *ad* and *su* operations. Whenever an *ad* or *su* order has been selected, a time pulse will appear at time 4 on the output of this gate, while on all the other operations no time pulse will appear. The last gate tube is like the first except that it is connected to the *ad* order and time pulse No. 2. Actually, for the machine there will be some forty or fifty of these gate tubes, and the connection matrices will be quite large. However, one 8-position time pulse distributor and a 32-position switch will be adequate for Whirlwind I control.

## 3.4 Program Register Characteristics

The sole purpose of the Program Register in WWI is to serve as a temporary storage for orders while the Storage Switch is being cleared.

In character, the Program Register will be a single flip-flop storage register provided with reading and receiving connections to the bus.

## 3.5 Arithmetic Element Characteristics

The Arithmetic Element should be capable of the following operations:

1) Addition
2) Subtraction
3) Multiplication
4) Division
5) Shifting
6) Transmitting and receiving from the bus
7) Detecting the sign of the number in the Accumulator

In operation, the Arithmetic Element obtains one number at a time from the bus and performs such operations on it as are requested. Where two numbers are required, as in addition, the first number is assumed to be in the Accumulator of the Arithmetic Element and the second is brought in by the operation order. The number in the Accumulator may be the result of a previous operation or may have been brought in by the previous transfer order.

The result of an arithmetic operation remains in the Accumulator. It may be placed in storage by an additional order, but such a transfer does not affect the contents of the Accumulator. A number in the Accumulator remains there until cleared by a new order which requires an empty Accumulator for its completion.

The characteristics of the Arithmetic Element will be discussed in greater detail in Section 4.

## 3.6 Input and Output Characteristics

The output devices needed for WWI and WWII must be quite versatile. It will be necessary to read out large quantities of numerical data, probably onto film, and to provide for converting these data from binary to decimal base and for printing the results. Visual displays will be needed as well as film recorders for making permanent records of graphs of functions, both against time and against other functions. For certain simulator applications, a large number of electrical magnitude signals must be converted from binary code and sent to other equipment to actuate meters and servos.

Film readers will be used for input mechanisms. Provision must be made for separate preparation of input films from keyboards where setup is convenient and does not require machine time. The contents of such films may be read into the computer at high rates of speed and there stored. It is also desirable to have a keyboard available which will allow rapid insertion of particular data into the machine during the course of a calculation. For simulator use, means must be provided for converting from mechanical and electrical signals to binary code and for inserting such data in the machine. These various devices will be discussed in detail in later reports.

An extremely desirable additional function of the input and output devices is the use of the input and output medium as an intermediate-speed storage of large capacity. The film equipment proposed is only partly satisfactory for this purpose. Large quantities of previously prepared data may be placed on the films of the input readers and called up as needed by the machine. However, if computed results are of such bulk as to require storage on the film, the time involved before such data may be received by the input is considerable. The film must be developed and sent to a reader. Even with automatic equipment, the remaining storage time may be excessive for some problems. This difficulty will not be serious, however, for large classes of

problems of immediate interest. A fluorescent film storage with no developing time should be available at some time in the future. Magnetic tape may also be used.

## 3.7 Readback Checking

Readback checking has been discussed previously in Section 2.8. The general purpose and procedure of the check is given there. The detailed timing will be discussed later.

One item that should be mentioned is the omission of readback checking on transfers to and from the Program Register. Such transfers are actually checked by a roundabout method. When the order is removed from Storage, it is also sent to the Check Register by way of the check bus. The order is sent to the Program Register and then to the Operation and Storage Switches. It is then read back into the main bus and sent to the Check Register for comparison. The check circuit includes the transfers to and from the Program Register as part of the checking loop.

## 3.8 Test Storage

For testing and checking purposes a small amount of storage in the form of vacuum tube flip-flops and manual switches is to be provided in WWI. This storage, limited to a total of 32 registers, will be valuable during installation and preliminary testing of the computer and probably will be retained in the complete system for the storage and control of a programmed check problem. Such a check might be run by the computer every few seconds to inspect components for proper operation. Operation of the computer is explained in this report using test storage as an example where necessary. The diagrams include connections and circuits for the test storage but this report has not yet been extended to include electrostatic storage.

### 3.81 Toggle-Switch Storage

The Toggle-Switch Storage is part of the test storage being built for WWI, and will consist of 27 banks of switches. Each bank contains 16 toggle switches, and can, therefore, store one 16-binary-digit word. The contents of each bank or register may be readily set or changed by manually setting the switches.

In combination with the five flip-flop storage registers to be described below, the 27 toggle-switch registers form a total storage capacity of 32 words which may be readily controlled by a 32-way electronic switch.

The toggle-switch registers may be used for either orders or constants.

During reading, the switches of the selected row will all be pulsed at once. The outputs of all switches will be mixed in a crystal diode matrix and sent to one set of line drivers for sending pulses to the bus. There is no question of receiving data from the bus for Toggle-Switch Storage.

### 3.82 Flip-Flop Storage

The Flip-Flop Storage is also part of the test storage. Five flip-flop registers of 16 binary digits each will be provided. These registers can receive as well as transmit information rapidly. In fact, these registers, in sufficient quantity, would do everything electrostatic storage would do. The expense and complication of such an arrangement would be prohibitive if the storage capacity were more than a few registers.

The registers will be connected to the bus through a single set of line drivers. The desired register will be selected by the Storage Switch, and the proper gate tubes pulsed to cause it to read out to the bus or receive from the bus. In general, these registers will be used for holding partial and completed results during tests but they may also be used for holding orders which are to be modified.

Each digit of each register will be provided with a toggle switch connected to a reset line. Whenever the reset line is pulsed, all registers are returned to some initially preset value.

### 3.83 Setup

The switches on the flip-flop registers and the toggle-switch registers are used for inserting data in the machine with test storage. No input readers or external storage is used. In general, constants and orders will be put in the Toggle-Switch Storage and initial conditions in the Flip-Flop Storage reset switches. The number of order sequences to be set up will be small, and manual setting will not be a handicap.

Remote setting of the Flip-Flop Storage resets will be provided in order to bring all the setup switches into one array.

### 3.84 Program Counter Modifications

Toggle switches will be provided on the sec-

tions of the Program Counter. When a calculation is begun, a reset pulse will set the counter to the register position of the first order to be used. The rest of the orders will be taken in turn. When the last order is taken the counter will be full. The next add pulse will cause an end pulse which will reset the counter to the first order of the test sequence or return it to the main program in the electrostatic storage as required.

### 3.85 Starting the Computer Using Test Storage

The sequence of operations for operation with test storage will be:

1) The Toggle-Switch Storage will be set to the desired constants and orders. The orders will all be in one sequence and will be stored in higher-numbered registers, the last order being stored in the highest-numbered register. The position of the first order depends on the total number of orders. Any of the unused registers may be used for constants.

2) The Flip-Flop Storage reset switches will be set to the desired initial conditions of the variables.

3) The reset switches on the Program Counter will be set to the register number of the first order in the order sequence.

If now the computer is turned on, and the clock is started, all the reset lines can be pulsed so that the initial conditions are inserted in the flip-flop registers and the register number of the first order is in the Program Counter. Then, the computer can begin at the desired control order.

The computer will run through the desired order sequence. When the sequence is complete, the end pulse will reset the counter to the first order position. The process will continue until the computer has been stopped manually, or until an error is detected by the checking and alarm circuits.

For test purposes, results appearing from the computer as a result of control orders in test storage can be displayed on the face of a cathode-ray tube as an amplitude plot against time. The tube will start to sweep when the reset lines to the computer are all pulsed. The results from the computer will be converted from binary code to electrical magnitude and used to set the vertical deflection on the tube. The beam will actually trace a series of steps which may either be smoothed electrically or disregarded if close enough together.

The beam will eventually sweep off the face of the tube. The computer will continue calculating steps but the results will not be displayed. A new reset pulse may now be supplied to the computer, reinserting all initial conditions and also restarting the sweep on the cathode-ray display tube. The computer will recompute the previous trace and continue to do so at a repetition rate determined by the sweep interval. The display will remain fixed on the face of the tube and may be examined at leisure.

In addition to this visual display, it will be possible to stop the computer and read numerical magnitudes from neon bulbs on the registers.

# 4. THE ARITHMETIC ELEMENT

## 4.1 Binary Arithmetic

### 4.11 Addition

It is necessary to consider the methods of binary arithmetic to understand the processes which WWI and WWII will use for computing.

Consider the familiar problem of decimal addition. Figure 12 shows a simple decimal problem. The digits are added in pairs, it being possible, however, to obtain a sum in any column which is greater than 9. The extra digit needed to describe this sum is known as a carry. This carry must be added in to the left-hand adjacent place. Ordinarily, this carry operation is accomplished without writing it out specifically as shown in the example. A decimal adding machine must be able to perform carry additions of this type.

Figure 13 shows a binary addition. The two binary numbers shown are summed up and it is once again possible to obtain a sum in any column which requires two digits. Thus, the sum of a 1 and a 1 in the binary system comes out 10, the 1 being a carry which once again must be added into the left-hand adjacent place. It is possible for these carries to produce other carries which must again be added in. In fact, it is possible, using 40-digit binary numbers, for a carry to be propagated the full length of the number, requiring 39 carry additions in the process. It can be shown statistically, however, that the expected number of carries in a 40-digit addition will be about five.* It is also possible to build binary adders which are able to sum the carry simultaneously in a single operation.

Figure 14 shows a simple binary adder. The lower flip-flops have their input signals supplied to their cathodes. They form simple scale-of-two counters. The addition of two ones to a flip-flop will restore it to zero, the overflow setting one of the upper row of flip-flops. These upper flip-flops store the carries. Following the first operation of addition, the sum will appear in the lower flip-flops and the carries in the upper flip-flops. Gate tubes are connected to these upper flip-flops in such a manner that the gate is open if the flip-flop holds a 1. If, then, the other grids of the gate are pulsed, a carry pulse will be transmitted to the left-hand adjacent adder flip-flop. At each pulsing of the carry line, one carry addition will be performed.

### 4.12 Subtraction; End-Around-Carry

Negative numbers must be handled. Several possibilities are shown in Figure 15. The first is to handle the number as its absolute magnitude with an associated sign. This method is the customary notation. From the point of view of the computer, however, such a notation is not too satisfactory since it requires a special subtracting unit and the ability to discriminate signs. Methods more convenient from the point of view of the computer are those using complements. In the 10's complement system, the positive number is subtracted from some higher power of the base than can be held in a standard register. The sum of a number and its complement is, therefore, this power of the base. The digit that signifies this power will lie off the left-hand end of the register, the sum of a number and its complement then appearing as a zero. Since, therefore, we have obtained a number which when added to the original number equals zero, we have satisfied the requirements for a negative number designation. A similar method but more convenient is the use of 9's complements. In this system the positive number is subtracted from a power of the base less 1. The convenience may be noted from the illustration: the 9's complement is obtained by subtracting each digit of the positive number from 9. In binary notation each digit of a positive number is subtracted from one. This is equivalent to replacing all zeros by ones and all ones by zeros. The conversion from 9's complement to 10's complement can be effected by adding a one to the 9's complement. Unfortunately, this addition may produce carries requiring an adder and an addition time for changing the sign of a number. The 9's complement system avoids propagation of carries.

By using the complement system, an adder may be made to subtract. Subtraction processes are shown in Figure 16. The first example shows a subtraction in which the result is negative. The subtrahend is complemented and added to the minuend. The result is the difference in its complemented form. The second example shows the subtraction of two numbers whose difference is positive. Once again the subtrahend is complemented and added to the minuend. In this particular case, however, the

* Burke, A. W., Goldstine, H. H. and von Neumann, J., *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*, The Institute for Advanced Study, 1946.

difference is not in the desired form. It consists of the desired difference $N_1-N_2$ plus the terms $2^n-1$. Fortunately, the correction for these terms is easy. The $2^n$ represents a digit off the left-hand end of the register. If, therefore, this digit is carried around and added into the right-most place, both the $2^n$ and the $-1$ will be corrected. This process is known as "end-around-carry", and is easily executed physically by connecting the carry section of the left digit of the Accumulator to the right digit counter.

### 4.13 Multiplication

Multiplication is nothing more than the process of successive additions, the multiplicand being added to the partial product according to the digits in the multiplier. Figure 17 shows a decimal multiplication example. The rightmost digit of the multiplier is multiplied by the digits of the multiplicand in turn. It will be noted that the results of each of these small multiplications may be a two-digit number requiring, therefore, an addition process to obtain the partial product. Following the use of the right-most digit of the multiplier, the next digit to the left is used. The partial products formed are added together. Addition of partial products is continued until all digits of the multiplier have been used.

In multiplication, binary notation presents many advantages. As shown in Figure 18, the binary multiplication table is reduced to 1x1=1. Since the largest product of any two binary digits is the single binary digit one, there can be no carry between sections in forming the partial product.

Figure 19 shows a binary multiplication example. It is carried out in the same fashion as the decimal example previously discussed except that each step is now simply a choice of whether or not to add in the multiplicand. This choice is governed by whether or not the multiplier digit is a one.

A modification of the binary multiplication process is shown in Figure 20. This modification is helpful for the mechanization of the process. The multiplication begins as before with the addition of the multiplicand depending on the right-most digit of the multiplier. Now, however, consider shifting both the multiplier and the partial product to the right one digit. The second step of the multiplication now considers the right-most digit of the remaining part of the multiplier, the multiplicand being added in directly. It is not necessary to shift the multiplicand since the partial product has already been shifted. This process of shifting multiplier and partial product to the right and then adding in the multiplicand, if the right-most digit of the multiplier is a one, continues until the entire multiplier has been used.

The basic essentials for this multiplication are shown in Figure 21. The multiplicand is stored in a binary register, the A-Register, while the multiplier is stored in a second binary register, the B-Register. The product is built up in the adding unit of the Accumulator or AC. Gate tubes are provided for adding the contents of the A-Register into the Accumulator.

Figure 22 shows the addition of a sensing device for examining the right-most digit of a multiplier. A gate tube is connected to the right-most digit flip-flop of the B-Register in such a fashion that this gate tube is on when the digit of the right-most column of the B-Register is a one. A pulse is now supplied to this gate tube, and if the multiplier digit is a one, the gate tube between AR and AC will be opened and the contents of AR will be added to AC.

Figure 23 shows the means provided for shifting the multiplier and partial product by one digit. A second gate tube is provided which is also connected to the right-most digit of the B-Register. This gate tube is arranged so that it is on when the right-most digit is a zero. When a pulse is supplied to this gate tube the contents of the Accumulator and B-Register are both shifted to the right one digit. The digit which would ordinarily be shifted off the Accumulator is put into the now empty space in the left-hand end of the B-Register.

Figure 24 shows a complete system for carrying out the successive additions of a multiplication. The clock pulses are supplied at the bottom line. If the right-most digit of the multiplier is a zero, gate tube No. 2 will be on and gate tube No. 1 will be off. The clock pulse will pass through gate tube No. 2 and cause the partial product in AC and the multiplier in BR to be shifted to the right one digit. A new multiplier digit will be put in the sensing position.

If the right-most digit of the multiplier is a one, gate tube No. 1 will be on and gate tube

No. 2 will be off. The clock pulse will pass through gate tube No. 1 and add the contents of AR into AC. This pulse after passing gate tube No. 1 will also return to the input of the right-most digit of BR, changing it from a one to a zero. The next pulse to come along will find gate tube No. 2 on and will shift instead of add. At each step of the process, partial product and multiplier are shifted. The multiplicand is added to the partial product prior to the shift if the multiplier digit is a one. If the multiplier digit is zero, the shift only is performed.

Figure 25 shows the complete multiplication control. A flip-flop is used to control a gate tube on the clock pulse input. When a multiplication is desired, a pulse is sent into this flip-flop, turning on the gate tube and sending clock pulses to the gate tube control previously mentioned for forming the successive additions of the multiplication. A counter is supplied for stopping the additions when the multiplication is complete. At each shift this counter is indexed one point. When the counter is full, it will send an overflow pulse down to the input flip-flop, turning off the clock pulse gate tube and stopping the multiplication. The counter is set so that this multiplication operation will be stopped after all digits of the multiplier have been sensed.

If would be possible to order the multiplication operation using the control distributor described previously. The main control is used for ordering some of the simpler operations such as addition and subtraction. More complicated operations such as multiplication and division may be more simply and efficiently accomplished by using separate controls as shown in Figure 25.

Nothing has been said about effecting the carry for each of the additions in the multiplication. The carries cannot be allowed to accumulate since it is not possible to store more than one per digit. One possibility is to effect a full carry following each addition, thus clearing all the carry flip-flops. This method would be satisfactory mathematically but would be wasteful of computing time. As will be shown below, it is not necessary to perform more than one step of the carry for each addition in the multiplication.

The only requirement on the carry at each addition is that it be possible to make another addition to the partial product without overflowing the capacity of the register. It is not necessary to clear the carry flip-flops entirely until the final product is obtained. The total capacity of each digit section of the Accumulator is 11 in binary notation or 3 in decimal notation. The 2 can be stored in the carry flip-flop; the 1 can be stored in the partial-sum flip-flop. The multiplicand to be added in cannot exceed 1 in each digit section. (The mutiplicand has no unadded carries.) If it can be assured that no section of the Accumulator holds more than 2 following an addition, then it can also be assured that the next addition cannot require any section to hold more than its capacity, 3.

This assurance can be obtained by performing only a single step of the carry. A section can hold a 3 only if its carry flip-flop holds a 1 while its partial-sum flip-flop also holds a 1, either there originally or added in by a carry from the right. But a single carry step clears all carry flip-flops initially, leaving in each section a 2 if the partial-sum flip-flop held a 1 and a 1 was added in from the right; a 1 if there was originally a 1 and no carry or a 0 and a carry, or a 0 if the partial-sum flip-flop held a 0 and there was no carry from the right. Under no circumstances will the section hold 3. The next addition and as many more as are required may, therefore, be safely made as long as a single carry is performed after each one.

It will be necessary to perform a full carry on the final product before it is in standard form (no unadded carries). It is also necessary to shift the contents of the carry flip-flops to the right as well as the contents of the partial-sum flip-flops. This shifting may be accomplished in several ways. One is to perform the single carry and then shift both rows of flip-flops. Another is to shift the partial-sum flip-flops only, performing the single carry afterward, taking account of the fact that the sections into which they are to be added have been shifted. The new carries result in the proper sections. A method is described in Section 4.43 for combining both shift and carry in a single operation. Using this method, the system described in Figure 25 is still appropriate.

Since the product of 2n-digit numbers will result in a 2n-digit product, some method must be provided for handling these extra digits. In general, they are insignificant and can be thrown away. This process requires the proper rounding off of the number. In rounding off, if the last n

digits are merely thrown away, the remainder will always be less than the complete number. In the course of a problem requiring a very large number of multiplications, the bias due to this procedure will become excessive. It is desirable to increase by one the right-most digit that is kept if the number thrown away is greater than a half of this digit. A convenient way of accomplishing this in the decimal system is to add five to the first digit to be thrown away. If this digit is five or greater, a one will be carried to the right-most digit of the number saved; if this digit is less than five, the right-hand digits will simply be discarded. This process is illustrated in Figure 26. In binary notation an entirely similar process can be used in which a one is added to the first digit to be discarded, resulting in a carry if this digit is a one and no carry if it is zero. No provision is made for properly caring for the ambiguous case in which the discarded part of the number is exactly one-half. However, in 40-digit multiplication, 40 digits will be thrown away, and the chances of this 40–digit number being exactly one-half are very small. The bias due to this process is also very small and can be neglected in most problems.

### 4.14 Shifting

The process of transferring a number to right or left within a register is called shifting. If a number is shifted one digit to the right, each digit of the number is transferred to the next digit space on the right. The right-most digit is transferred off the register and lost. Zeros or ones are inserted in the left-hand end depending on whether the number being shifted is positive or negative. The process may be extended any number of digit spaces, and is similar for shifting left.

Shifting is equivalent to pointing off. Shifting right is equivalent to moving the point to the left and vice versa. Thus, shifting right is equivalent to dividing by the base, in this case 2. Shifting left is equivalent to multiplying by 2.

### 4.15 Division

Division is carried out by a process of successive subtractions in the same manner as decimal division. Once again the binary system is considerably more convenient than the decimal.

1) There is no question as to how many times the divisor will go into the remainder. In decimal division there are ten possibilities. In binary division there are only two possibilities, 1 and 0, corresponding to success and failure. The divisor either goes in or it does not.

2) In the normal course of mechanized division, the process is to subtract the divisor from the dividend as many times as possible until an overcast occurs. By overcast is meant the appearance of a negative remainder. There are now two possibilities.

First, the divisor may be added once, clearing the overcast. The net number of subtractions is the appropriate digit of the quotient. The remainder may now be shifted left one digit and the process repeated. See the illustrative example, Figure 27.

The second possibility is to shift left immediately after the overcast occurs and then to add the divisor to the negative remainder until a positive result is obtained. The remainder is again shifted, the divisor subtracted, and the process proceeds alternately adding and subtracting the divisor. The quotient will be in rather peculiar form. The additions must be considered as negative digits in the quotient. Corrections must be made before the result is in standard form.

When binary numbers are used no such correction is necessary. Suppose that in the course of a division an overcast occurs. Now, in the first method above the overcast would be followed by an addition, a shift, and a subtraction. The shift and subtraction is equivalent to subtracting one-half the divisor, and the net result, including the preceding addition, is the addition of one-half the divisor. The same result may be obtained by shifting and adding. Only one addition or subtraction is needed at each step.

The above considerations lead to the following procedure for binary division.

1) Subtract the divisor.

2) If the remainder is negative, put down a zero in quotient and shift the remainder left one digit. Next, add the divisor.

3) If the remainder is positive, put a one in the quotient and shift the remainder left one digit. Then again subtract the divisor.

4) Continue the above process for the required number of steps.

Since the division process requires subtrac-

tion and the shifting of negative numbers, care must be taken to perform the necessary sign corrections and end-around-carries.

Figure 28 illustrates a binary division carried out by this method.

Since the computer registers are incapable of holding a number greater than one, the divisor must always exceed the dividend for a successful division. A convenient check on this requirement is to notice the result of the first subtraction of the division process. If the remainder is positive, then the dividend exceeds the divisor, a setup error has been made, and the machine should stop.

A second, relatively minor, problem arises in the representation of zero in the 9's complement system. If a number is added to its negative the result should be zero. In the 9's complement system the result is all 9's, which must represent zero. In the binary system zero is all 1's. All 0's also represent zero, since the negative of zero is also zero.

The means which the computer has available for checking the sign of the remainder is to examine the sign digit of the Accumulator. If this digit is a one, the remainder is negative, representing an unsuccessful subtraction. If the digit is a zero, the remainder is positive, representing a successful subtraction. But a zero remainder, which represents the most successful subtraction possible, results in a one in the sign digit place. The quotient which should end in a one followed by all zeros will instead end in a zero followed by all ones. This difficulty is corrected by the rounding procedure, as will be discussed later.

## 4.2 The Arithmetic Registers

The Arithmetic Element is shown schematically in Figure 29. It consists of three major parts and a number of smaller control elements. The major parts are:

1) The A-Register—The A-Register or AR is a simple flip-flop storage register, the purpose of which is to receive all numbers coming from storage to the Arithmetic Element. It holds incoming numbers prior to their addition to or subtraction from the contents of the Accumulator. The A-Register is also used as a storage place for the multiplicand in multiplication and the dividend in division.

2) The Accumulator—The Accumulator or AC is the adder element which is used for all the arithmetic operations. It forms the sum or difference in addition and subtraction and the product in multiplication, and holds the remainder in division.

3) The B-Register—The B-Register or BR is used to hold the multiplier during multiplication and the quotient during division.

The three registers have other capabilities, which will be discussed in the appropriate sections.

## 4.3 The A-Register

The functions of the A-Register are:

1) To receive a number coming into the Arithmetic Element from storage. The A-Register must be connected to the bus by a set of gate tubes.

2) To transmit numbers to AC for addition or subtraction. Gate tubes must be provided for reading either the number or its complement into AC.

3) To hold the multiplicand and divisor. No additional equipment is required for these services. The contents of AR are simply added or subtracted into AC in a normal fashion, but under the control of the process in question.

4) To sense the sign of the number in the A-Register for use in determining the signs of products and quotients.

5) To change the sign of the number in the A-Register by forming its complement.

## 4.4 The Accumulator

### 4.41 Simple Adder

The Accumulator must be able to add two binary numbers. One of these numbers is presumed to be in the Accumulator flip-flops; the second is sent in from the A-Register through a set of gate tubes.

Section 4.11 describes briefly the processes of binary addition. Figure 30 shows a simple binary adder. One number is held in the lower flip-flops. The digits of the second number are supplied on the lines coming into the trigger inputs of the flip-flops. The sum appears in these flip-flops with the carries, if any going into the upper or carry flip-flops. A carry signal is supplied only when a partial-sum flip-flop goes from a 1 back to a 0, a situation that occurs when a 1 is added to a 1.

The carries are added into the adjacent partial-sum flip-flops when the carry line is pulsed. These carry additions may produce new carries requiring more carry additions. There may be as many carry additions as there are digits in a number, although the average number of carries will be much smaller. If a sensing device is added which will determine when all the carry flip-flops are empty, advantage can be taken of the small average number of carries.

### 4.42 High-Speed-Carry

It is possible to perform all the carries in one step. Referring again to the low-speed carry, it will be seen that a new carry will be created only if a carry being added to a partial-sum flip-flop finds this flip-flop already holding a 1. The resulting carry must be added to the adjacent partial-sum flip-flop. If, therefore, the original carry digit coming in is allowed to go past the flip-flop holding a 1 and add into the next in line, the time-consuming side trip through the carry flip-flops will be avoided.

Such a carry system is shown in Figure 31. The addition is the same except that the carry signal to the carry flip-flops is sent through gate tubes connected to the 1 sides of the partial-sum flip-flops. This is done so that the carry may be conveniently avoided later in the operation.

The carry flip-flop gate tubes are pulsed as before but only one pulse is necessary. If the adjacent partial-sum flip-flop holds a 1, the carry finds the high-speed carry gate tube on, and passes through to add to the next section. If this section also holds a 1, the carry will pass on until it finds a 0. The carry digit will also add into the flip-flops on the way, but diodes are provided to prevent the carry digits from resetting the carry flip-flops. The addition procedure is then:

1) Add the new number into the Accumulator. The sum will appear partly in the partial-sum flip-flops and partly in the carry flip-flops.

2) Pulse the carry line once. All the carries will be added in and the complete sum will appear in the partial-sum flip-flops. The time required will be the transit time through the high-speed carry gate tubes. There may be as many of these tubes in line as there are digits in the Accumulator.

Also shown in Figure 31 are the read-out gate tubes, which allow the contents of the Accumulator to be transmitted to the bus and from there to storage.

### 4.43 Shift and Carry

The Accumulator must be able to shift right, not only because of the general value of this operation, but also because it is a basic part of the multiplication procedure. It turns out that a special modification of shift right is very valuable in reducing multiplication time. This modification consists of performing the single carry and shift-right operations simultaneously with only one setting of the flip-flops required.

The method is as follows:

1) The contents of the partial-sum and carry flip-flops of any digit column following a single carry are determined by the prior contents of the same partial-sum flip-flop and the carry flip-flop belonging to the next digit to the right. This is so because the contents of the carry flip-flop are added to the partial-sum flip-flop, the sum of the two determining the contents of the digit following the carry.

Following both the shift and the carry, the contents of any digit column are completely determined by the original contents of the same carry flip-flop and the partial-sum flip-flop in the next digit column on the left.

There are two possible positions for each of the two flip-flops in question. If, then, the two flip-flops are made part of a four-way switch, the outputs of the switch may be used to select the new contents of a digit column. The equivalents are shown in the following table.

| FF | FF2 | CARRY FF |
|---|---|---|
| FF1 | FF3 | PARTIAL-SUM FF |

CONTENTS

| Before | | After | |
|---|---|---|---|
| FF1 | FF2 | FF3 | FF2 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |

If, then, both FF1 and FF2 are either 0 or 1 before shift and carry, then FF3 should be a 0 after-

wards. If either FF1 or FF2 is a 1 and the other 0 beforehand, then FF3 should be a 0 after the operation. The easiest way to set FF2 when it contains a 1 initially is to clear it if FF1 is 0 and not clear it if FF1 is a 1. The case where FF1 is 1 and FF2 is 0 is still satisfactory, since FF2 will remain 0 even without a clear pulse.

The system is shown in abbreviated form in Figure 32. The method is also satisfactory for simple shift right.

#### 4.44 Shift Left

There is no need for a special method such as the carry added to shift right. Shift left is described in Figure 33.

The gate tubes GT0 and GT1 are connected to the 0 and 1 sides of FF2 respectively. One gate tube or the other will be on depending upon the condition of FF2.

A pulse on the shift-left line will proceed through the open gate tube and set FF1 to the same state as FF2. An equivalent system will set FF2 to the same state as FF3 and so on. The delays are provided to allow time for the shift-left pulse to proceed through the gate tubes before the FF's are set.

During the division operation it is desirable to shift left on certain digits where the shift is not ordinarily required. These circumstances will be clarified when the division and shift operations are described in Section 4.6.

#### 4.45 Other Requirements

In addition, the Accumulator must have the following abilities:

1) It must be able to read out to the bus. All information extracted from the Arithmetic Element for storage or display must come from the Accumulator.

2) It must be able to read into the B-Register. The multiplier is originally in AC and must be transferred to the B-Register before a multiplication can start. It would be too slow to transfer by shifting; therefore, a set of transferring gates are provided.

### 4.5 The B-Register

The B-Register is used for holding the multiplier during multiplications and the quotient during division. The B-Register must have the following abilities:

1) It must be capable of receiving numbers transferred to it from the Accumulator. It is not necessary for the B-Register to read out except by shifting left.

2) It must be able to shift its contents to the left or right, thereby either inserting digits in AC or removing them from AC as the case may be.

3) It must be able to sense the condition of both the right and left digits for use in multiplication and correction for sign.

### 4.6 The Operations

The operations given in the order code of Section 2.53 will now be discussed in detail. The Arithmetic Element drawing, Figure 29, will be used in these discussions with those parts not pertinent to the particular operation deleted in each case.

#### 4.601 Clear and Add—*ca*. Figure 34

The purpose of the clear-and-add order is to insert a number coming along the bus from Storage into the Accumulator. All numbers going into the Arithmetic Element go first into the A-Register. From there the number will go to AC, which has been cleared previously. The order of operations is:

1) Clear A-Register.
2) Read in from bus.
3) Clear Accumulator. (This may be done at the same time as either 1 or 2.)
4) Add (transferring contents of AR into AC). No further operations are necessary.

#### 4.602 Add—*ad*. Figure 35

The purpose of the add order is to add a number coming along the bus from Storage to a number already in the Accumulator. The order of operations is:

1) Clear A-Register.
2) Read in from bus.
3) Add (transferring contents of AR into AC).
4) Carry (pulse the high-speed carry line once). The sum will then be in the AC partial-sum flip-flops. The Accumulator itself will take care of any end-around-carries.

One problem, that of the arithmetic check, still remains. See Section 2.82. If the sum of the two numbers added exceeds unity, the register capacity of the computer will have been exceeded and the computer should recognize the error.

This check is carried out in the following manner:

1) If the numbers added together are both positive, the sum will be positive. An overflow will result in the appearance of a 1 in the sign-digit space after the carry.

2) If one number is positive and the other negative or if positive numbers are subtracted, then the sum can never exceed in magnitude the larger of the two numbers. These numbers are each presumed to be less than unity; therefore, the sum can never be greater than unity and an overflow cannot occur.

3) If both numbers are negative, the sum should be negative. The sign digits when summed result in a 0 in the sign digit of AC plus a carry in this column. The sign digit is replaced by a carry from the right. If no such carry appears, the sign digit will remain 0, representing an overflow.

To recapitulate, if the numbers added are positive and a carry is added into the sign-digit place of AC, an overflow has occurred. If the numbers added are negative and no carry is added into the sign-digit place of AC, an overflow has occurred. No overflow is possible if the numbers added are of different sign.

Consider Figure 36, which is a sketch of the AC0 or sign-digit section of the Accumulator showing the connections required for arithmetic check. The details of the section have been left out. The AC0 carry FF is shown external to the Accumulator in Figure 35.

The only differences between this section and a normal Accumulator section are:

1) The signals to the carry flip-flop are sent to the trigger input instead of to the 0 or 1 inputs.

2) The dotted line connecting the carry input from AC1 to the AC0 carry trigger input has been added.

3) A diode has been added to keep the carry pulse from clearing the AC0 carry FF.

The operation now is:

1) Addition of two positive numbers.

Before the carry, both AC0 FF's will hold 0's. If a carry comes in from AC1 signifying an overflow, the carry pulse will travel up through the dotted line, setting FF 309.01 to a 1 and opening GT 600.04. The check pulse, supplied at a later time, will go out on the alarm line. If no carry comes in, FF 309.01 will remain set to 0, GT 600.04 will be off, and no alarm pulse will result.

2) Addition of a positive number to a smaller negative number.

Before the carry, AC0 partial-sum FF will hold a 1 and the AC0 carry FF a 0; GT 05 will be on. A carry will come in from the right (see Section 4.12, Fig. 16) and change AC0 partial-sum FF back to 0. The pulse will travel through the dotted line setting FF 309.01 to a 1. The pulse will also pass through GT 05, providing an end-around-carry which will travel through the delay back to FF 309.01, resetting it to 0. There can be no overflow and there will be no alarm.

3) Addition of a positive number to a larger negative number.

Before the carry, the AC0 partial-sum FF will hold a 1 and the AC0 carry FF a 0. GT 05 will be on. However, no carry will come in from the right. (See Section 4.12, Figure 16.) FF 309.01 will remain off. There can be no overflow and there will be no alarm.

4) Addition of two negative numbers.

Before the carry, the AC0 partial-sum FF will hold a 0 and the AC0 carry FF a 1. When the carry line is pulsed, an end-around-carry will result but FF 309.01 will not be cleared. If there is an overflow, no carry will come in from the right, the FF will remain set, GT 600.04 will be on, and the check pulse will produce an alarm.

If a carry comes in from the right, it will reset FF 309.01 by way of the dotted line. There will then be no alarm.

**4.603 Clear and Subtract–*cs*. Figure 37**

The clear-and-subtract order is similar to the clear-and-add order except that the negative or complement of the number is put into AC. The order of operations is:

1) Clear A-Register.

2) Read in from bus.

3) Clear Accumulator. (May be done at the same time as either 1 or 2.)
4) Subtract (transferring the negative of the number in AR into AC).

No further operations are necessary.

### 4.604 Subtract—*su*. Figure 38

The subtract operation is the same as add except that the complement of the number in AR is added to the number in AC. The arithmetic check is the same as for add.

The order of operations is:

1) Clear A-Register.
2) Read in from bus.
3) Subtract (transferring negative of contents of AR to AC).
4) Carry. (Pulse the high-speed carry line once.)
5) Arithmetic check

No further operations are necessary.

### 4.605 Multiply and Roundoff—*mr*. Figure 39

In the multiply-and-roundoff operation, the number coming along the bus from storage is to be multiplied by the number in the Accumulator. The product is to be rounded off to a single register length, the insignificant digits being discarded.

The multiplication operation is outlined in Section 4.13. The multiplicand should be in AR as it will be when received from the bus. The multiplier should be in BR and must be transferred there from the Accumulator.

The initial operations are:

1) Clear AR and BR (to prepare them for receiving multiplicand and multiplier).
2) Read in from bus (putting multiplicand in AR).

Before transferring the multiplier from AC to BR, consider the problem of the multiplication of negative numbers. The multiplication process that has been described is only for positive numbers. The product of negative numbers represented in the 9's complement becomes very complicated.

$$\begin{array}{r} 2 - N_1 - 2^{-n} \\ \times\ 2 - N_2 - 2^{-n} \\ \hline 4 - 2(N_1 + N_2) + N_1N_2 - 2 \times 2 \times 2^{-n} \\ \quad + 2^{-n}(N_1 + N_2) + 2^{-2n} \end{array}$$

$$= N_1N_2 + 4 - 2^{-n+2} + 2^{-2n} - (2 - 2^{-n})(N_1 + N_2)$$

Correction must be made for all the superfluous terms.

For this reason the Arithmetic Element is designed to multiply positive numbers only. Negative numbers are changed to positive numbers prior to a multiplication and the sign of the product is changed after the operation if necessary.

The Sign-Control FF 304.01 is used for remembering the proper sign for the product. GT 304.05 is connected to the sign digit FF of AC in such a manner as to be on if this digit is 1 (corresponding to a negative multiplier). A pulse coming in the accumulator sign line will pass GT 304.05 if the multiplier is negative. The pulse then goes in the complement line to AC, changing the multiplier to a positive number. It also goes to the Sign-Control FF 304.01, setting it to a 1. If the multiplier is positive, the pulse goes no further than GT 304.05. The accumulator sign pulse thus insures that the multiplier is positive. If the sign of the multiplier has been changed, this fact is remembered in FF 304.01.

The same procedure is used for checking the sign of the multiplicand in the A-Register. GT 304.07 is on if the multiplicand is negative. The A-Register sign pulse will then complement the A-Register, changing the multiplicand to a positive number. The sign pulse will also change the condition of Sign-Control FF 304.01.

The result of these operations is:

a) Both multiplier and multiplicand are positive.
b) If the sign of neither multiplier nor multiplicand was changed, FF 304.01 will remain set to 0.
c) If the sign of either the multiplier or the multiplicand (but not both) was changed, FF 304.01 will be set to a 1, representing the fact that the sign of the product will be wrong and must be changed.
d) If the signs of both multiplier and multiplicand were changed, FF 304.01 will be set to a 0, which is correct since the product of two negative numbers is positive.

The further operations in preparation are:

3) Accumulator Sign (to check sign of multiplier)

4) To B-Register (transferring positive multiplier from AC to BR)
5) Clear AC (ready for product)
6) A-Register Sign (to check sign of multiplicand)
7) Multiply

The multiply pulse changes FF 306.01 to a 1, opening GT 306.04 and allowing high-frequency time pulses to reach the B-Register. The multiply pulse also resets the Step Counter 305 so that an end-carry will be obtained after the proper number of steps have been completed.

The high-frequency time pulses are supplied to the right-most digit of BR.

If BR15 holds a 1, the pulse will return on the from-BR15(1) line and go up to the add input of the A-Register. The multiplicand will be added into AC. A connection is provided within BR for resetting BR15 to 0.

If BR15 is a 0 or has been reset to 0 by a BR15(1) pulse, the next time pulse will come out the from-BR15(0) line and will cause a shift-and-carry in AC, a shift-right in BR, and an addition into the Step Counter.

The system shown is thus equivalent to Figure 25, Section 4.13. When the proper number of steps have been performed (15 for the 16-digit WWI), the step counter will put out an end-carry pulse which will return FF 306.01 to 0, closing GT 306.04 and stopping the operation.

The original multiply pulse stopped the flow of time pulses to the distributor in the main control. The rest of the computer has been waiting during the high-frequency part of the operation. The end-carry pulse from the Step Counter not only stops the high-frequency time pulses but also turns on the main control Time-Pulse Distributor, restarting the main part of the computer.

The contents of the Accumulator do not constitute the final product as yet. There may be some carries left in the carry flip-flops, since only single carries have been performed. The next steps are then:

8) Carry (high-speed)
9) Roundoff
10) Clear BR

The roundoff pulse is sent to BR at the same time that the carry line is pulsed. If BR0, the left-most digit of BR and correspondingly the next digit of the product beyond the last one to be kept, is a 1, the roundoff pulse will be sent into the end-around-carry line and be added into AC15. If BR0 is a 0, nothing will be added in. The contents of BR following roundoff are of no value and may be discarded.

It may be necessary to change the sign of the rounded product which is now in the Accumulator. If the sign is to be changed, FF 304.01 will be set to a 1 and GT 304.04 will be on. A pulse on the product-sign line will pass through the gate tube and complement the product in the Accumulator. The pulse will also reset the Sign-Control FF to 0, ready for the next operation.

11) Product-Sign

The rounded product with the proper sign is now in AC and the multiplication operation is complete.

### 4.606 Multiply and Hold Full Product—*mh*. Figure 39

The *mh* operation is identical with the *mr* operation except that the product is kept at the full two-register length instead of being rounded off at one register length.

The steps are the same as those in *mr*, except that the roundoff pulse is omitted at the final carry and the B-Register is not cleared. The product-sign pulse complements AC if necessary but not BR. The only way that the contents of BR can be used is to put them in AC by shifting left. The sign of BR will be taken care of during this shift, which is a separate operation.

### 4.607 Divide—*dv*. Figure 40

In the divide operation, the number held in the Accumulator of the Arithmetic Element is to be divided by the number coming along the bus from Storage. The quotient will be left in the B-Register. An additional shift-left operation will be required to get the quotient into AC where it can be used.

The division operation is outlined in Section 4.15. The dividend will be in AC as the result of some earlier operation. The divisor will be brought into AR from the bus. The necessary equipment is shown in figure 48.

The initial operations are:

1) Clear AR (prior to receiving the divisor from the bus).

2) Read in from bus (putting divisor in AR).

Once again it is convenient to operate with **positive numbers only.** The divisor and dividend are made positive, and the desired sign of the quotient stored in the Sign-Control FF 304.01, as in multiply-and-roundoff. See Section 4.65.

3) Accumulator Sign (to check sign of dividend).
4) A-Register Sign (to check sign of divisor).

The divisor and dividend are now ready for the process of successive subtractions by which the division is accomplished. One difficulty still remains: the capacity of a register is less than unity. The divisor must therefore be greater than the dividend for the quotient to be less than unity. Once again, the burden of assuring this condition lies on the person assigning scale factors as part of the setup procedure. It would still be desirable for the machine to be able to detect such a setup error.

A method of determining such a division overflow is to examine the result of the first subtraction. If the remainder is positive, the dividend was greater than the divisor and an overflow has occurred.

It is therefore desirable to subtract the divisor from the dividend before any shifting is done. The division control to be described below provides for subtraction only following a shift. The initial subtraction is therefore ordered by the main control.

5) Subtract (forms first subtraction of divisor from dividend)
6) Divide

The divide pulse sets FF 308.01 to a 1, opening GT 308.04, allowing low-frequency clock pulses to reach the Divide Time-Pulse Distributor. The Step Counter is also reset to give an end-carry after the proper number of counts.

The Divide Time-Pulse Distributor FF 308.02 is originally set to 0. GT 308.05 is on, and GT 308.06 is off. The first time pulse from GT 308.04 will pass through GT 308.05 to carry line and effect the carry which will add one to the Step Counter.

FF 308.02 will be set to a 1 by the time pulse after it progresses through the delay. The next pulse will pass through GT 308.06, which is now open, and will go to the divide-shift-left lines of both AC and BR.

Divide-shift-left differs from standard shift-left in the following particulars:

a) Every digit of AC is shifted, including both the sign digit AC0 and AC1, which are not shifted ordinarily.
b) The left-most digit of BR, BR0, is not shifted left although it is shifted ordinarily. The digits of BR are, therefore, not shifted into AC.

The reasons for these changes are as follows:

The entire contents of AC are to be shifted; therefore, AC1 must be shifted. It is now necessary to shift negative numbers. A negative number is assumed to have 1's extending beyond the last digit instead of 0's as for positive numbers.

| | |
|---|---|
| Positive number | 0.10101 |
| means | .....0000.1010100000..... |
| Negative number | 1.01010 |
| means | .....1111.0101011111111..... |

When a negative number is shifted to either right or left, it is necessary to insert 1's instead of 0's in the vacated spaces. The insertion may be conveniently accomplished in shift-left by taking the 1 shifted off the left end and inserting it in the right end in a sort of end-around-carry. The contents of BR bear no relation to the contents of AC and should not be shifted into AC.

The sign of the remainder must be sensed before the decision to add or subtract can be made. When the divide shift-left is made, the sign digit is shifted off into the lines leading around to AC15. The sign digit is thus readily available for initiating addition and subtraction. The shifted sign digit is also used for building up the quotient.

The following operations result from the divide shift-left pulse.

a) The entire contents of AC are shifted 1 space to the left. This shift includes the insertion of a 1 in the right-most place of AC, if necessary.
b) The contents of BR are also shifted left except for the left-most digit.
c) A 1 shifted left from AC0 represents a negative remainder. The 1 is therefore used to initiate an addition and to insert a 0 in the right end of BR.

d) A 0 shifted left from AC0 represents a positive remainder. The 0 is therefore used to initiate a subtraction and to insert a 1 in the right end of BR. Note that a shift produces a positive pulse on either the 0 or the 1 line. An actual 0 pulse is thus available for producing the above operations.

The necessary additions, subtractions, shifts, and insertion of digits into the quotient are all accomplished by the divide shift-left pulse. Following each divide shift-left is a full high-speed carry supplied from the Divide Time-Pulse Distributor. A full carry is needed to guarantee obtaining the true sign of the remainder. Low-frequency clock pulses are used to allow time for the carry and for the sequence of operations initiated by divide shift-left.

The divide-error FF 308.03 is used to determine if the result of the first subtraction is positive or negative and to give a warning if an overflow has occurred. The Accumulator sign pulse sets FF 308.03 to 1, opening GT 308.07. The first subtraction is then ordered by the main control. The Divide Time-Pulse Distributor takes over control and performs the carry and the first divide shift-left. If a 1 is shifted off, signifying a negative remainder, FF 308.03 will be reset to 0 and the division will proceed. If a 0 is shifted off, signifying a positive remainder and an overflow, the 0 pulse will pass through GT 308.07 and out the alarm line, stopping the clock.

A pulse is added into the Step Counter for each divide shift-left. When the desired number of subtractions have been performed, the Step Counter will put out an end-carry pulse which will reset FF 308.01 to 0, turning off GT 308.04, and shutting off the supply of low-frequency clock pulses.

The process will be stopped after a divide shift-left, which will result in an addition or subtraction. There will be no carry to complete this addition, but no harm will be done since the remainder is not to be retained. The final divide shift-left is necessary to provide the quotient digit corresponding to the next-to-last addition or subtraction.

The Step-Counter end-carry pulse will restart the main control. The next step is to clear AC. AC holds a remainder including possible carries. The quotient digit associated with this remainder has not been inserted in BR. The contents of AC are worthless and may be cleared.

7) Clear AC
8) Product sign

The product sign pulse now gives the proper sign to AC but not BR. It was necessary to clear AC to insure the AC sign digit being 0 prior to product sign. When the quotient is shifted from BR into AC it will be given the proper sign automatically. See Section 4.610.

The division process is now complete except for shifting the quotient from BR into AC, which is accomplished by a separate operation.

Several comments are necessary on the number of steps required. It is desirable to compute an extra digit of the quotient in order to provide for rounding off. Seventeen digits, one of them sign, must therefore be computed. Since the sign digit is always 0, the 16 significant digits may all be stored in BR.

The division process thus requires 17 subtractions, 17 carries, and 17 shifts. The first subtraction is ordered by the main Control. The remaining 16 are ordered by the divide shift-left pulse. The 17th shift produces the unnecessary subtraction mentioned above.

The shift operation used following *dv* to shift the quotient into AC should have 15 steps. The 15 significant digits of the quotient are shifted into AC, the 16th digit is used for rounding off, and the sign digit will be the sign of AC as determined by the product sign system.

Computing an extra digit and using it for rounding off is of particular value when the divisor is exact. The division process is determined by the sign of the remainder at each step: a negative remainder represents an unsuccessful subtraction. But if the divisor is exact, some remainder will be 0. Zero is represented by all 1's in the 9's complement system. Its sign digit is therefore a 1, which makes the 0 remainder appear to the machine as negative. The operation appears unsuccessful, while actually the subtraction was as successful as possible. The machine should put a 1 in the quotient followed by all 0's. Instead, the machine will put a 0 in the quotient followed by all 1's. This difficulty will be corrected by the round-off, which will clear the 1's and insert a 1 in the 0 place.

### 4.608 Transfer to Storage–*ts*. Figure 41

The *ts* operation sends the contents of AC out onto the bus, from which they are sent to some storage register designated by the register number section of the *ts* order. The Accumulator is not cleared, because its contents may be of immediate as well as of future use.

The sequence of operations is:

1) Pulse the to-bus line on AC;

no further operations are necessary.

### 4.609 Shift-Right–*sr*. Figure 42

In Shift-right, the contents of AC and BR are shifted to the right. The digits shifted off AC are put into the left end of BR. The digits shifted off BR are simply lost. Zeros are inserted in the left end of AC. Following the shift the number is rounded off and BR is cleared.

Information as to the number of places to be shifted is received from the bus and sent to the Step Counter 305. The information is given in the register number section of the *sr* order. The register number section of the order is always sent to both the Storage Switch and the Step Counter. When the Step Counter is to be used for shifting, it is already properly set according to the order. The Storage Switch is set to some random register, but no harm is done since the Storage is not read. When the Step Counter is to be used for some operation other than shifting, it is reset to the desired quantity. When the step counter is not used its contents are unimportant. For test storage, sufficient digits are available in the order to allow separate sections for the Step Counter and Storage switch codes. The control changes required to allow use of the register number section for both purposes are not discussed in this report.

Once again, in order to simplify the corrections necessary for the shifting of negative numbers, the numbers will be made positive prior to shifting and corrected after shifting.

Assuming that the Step Counter has been previously set by the order from the bus, the sequence of operations is:

1) Accumulator-Sign (changing sign of number in AC if necessary)

It is not necessary to change the sign of BR, since the number in BR is always positive. The number in BR is always obtained as the result of some operation involving only numbers whose signs have been made positive. In correcting for sign following these operations, the contents of AC only are changed, not those of BR. The sign of the number in AC is assumed to belong to the quantity in BR. In BR, therefore, a negative number is represented by its absolute magnitude plus sign.

2) Shift-right

FF 307.02 will be switched to a 1, turning on GT 307.05, allowing high frequency clock pulses to reach the shift-and-carry line. Shifting to the right is accomplished by the same shifting means used in multiplication. The single carry built into this shift does no harm. The contents of BR are shifted right at the same time so as to keep the digits necessary for rounding off.

The Step Counter will count the number of places shifted. When the desired number have been counted, the Step Counter will produce an end-carry, which will reset FF 307.02, closing GT 307.05 and shutting off the supply of clock pulses. The end-carry will also restart the main control as in the multiplication operation.

3) Roundoff

The Roundoff pulse will add 1 to AC15 if BR0 is a 1. The high-speed-carry system will take care of any carries resulting from this addition.

4) Product-Sign (corrects sign of AC if necessary).

5) Clear BR. (The contents of BR are worthless following the roundoff.)

No further operations are necessary.

### 4.610 Shift Left—*sl*. Figure 43

There is no difference between *sl* and *sr* except that high-frequency clock pulses are supplied to the shift-left line in *sl* and to shift-and-carry in *sr*.

The Step Counter will have been set previously by the order from the bus. The sequence of operations is then:

1) Accumulator-Sign. (The shifting is done using positive numbers.)

2) Shift-left.

FF 307.01 will be switched to a 1, turning on GT 307.04, allowing high-frequency clock pulses to reach the shift-left line. Both AC and BR are shifted to the left. All digits are shifted except the left two in AC. The sign digit is left as it was in order to keep the sign of the shifted quantity the same as before. The digit in AC1 is therefore not

shifted into AC0. There is no value to shifting AC0 to the left, since it is always zero and has no place to go anyway. The digits from the left end of BR are shifted into the right-most place of AC.

When the desired number of places have been shifted, the end-carry from the Step Counter will stop the high-frequency clock pulses and restart the main control.

3) Roundoff. (The shifted number is rounded to the nearest digit in the right-most place of AC.)
4) Product sign. (Corrects sign of AC if necessary.)
5) Clear BR. (The contents of BR are worthless following the roundoff.)

No further operations are necessary.

**4.611 Subprogram—*sp*.**

The Arithmetic Element takes no part in the subprogram operation. See Section 2.5.

**4.612 Conditional Program—*cp*. Figure 44**

The *cp* order makes the operation of subprogramming dependent on the sign of the number in AC.

As shown in Figure 48, the only elements needed are AC, where the number in question is held, and GT 304.08. If AC holds a positive number, the left digit of AC will be 0 and GT 304.08 will be on. A compare pulse will pass through the gate tube and return on the change-control line to the Operation Control, where it will initiate the subprogram operation. If the number in AC is non-positive (negative or zero), GT 304.08 will be off, the compare pulse will not get through, and no subprogram will result.

The sequence of operations is:

1) Compare. (Pulse GT 304.08.)

No further operations are necessary.

**4.613 Transfer Digits—*td*. Figure 41**

As far as the Arithmetic Element is concerned, there is no difference between *ts* and *td*. In both orders, the entire contents of AC are read out into the bus. The storage receives all or part of the word depending on whether the order is *ts* or *td*.

The sequence of operations is the same as for *ts*.

**4.614 Special Add–*sa*. Figure 45**

The purpose of *sa* is to facilitate the addition of double-length numbers.

In the addition of double-length numbers, there is the possibility of a carry from the left-most digit of the less significant section to the right-most digit of the more significant section. This carry will appear as an overflow in the addition of the less significant sections. This overflow must be removed and added to the sum of the more significant sections. Essentially, then, the following steps are required:

1) Sensing of an overflow after the addition of the small sections. This ability is already available in the arithmetic check.
2) Determination of the sign of the overflow so that the carry may be a plus 1 or a minus 1, as required.
3) Removal of the overflow by either adding or subtracting 1 depending on the sign of the carry.
4) Memory of the existence and sign of the carry and its addition to some later sum.

The following points should be clarified. Sign digits are carried with both sections of a double-length number. This is necessary for manipulations with numbers of unlike sign and for end-around-carries in summing negative numbers. The sign of the carry from small section to large section does not, therefore, necessarily bear any relation to the sign of the large section.

The sequence of operations is:

1) Perform a normal addition *ad*, omitting only the arithmetic-check pulse.

Following this addition FF 309.01 will be set to a 1 if there has been an overflow.

2) Pulse the special-add line.

If FF 309.01 is 0, GT 309.06 will be off and nothing will occur. If there has been an overflow, the pulse will pass through GT 309.06 and add into AC0. If the overflow is a –1, resulting from the addition of two negative sections, the AC0 partial-sum FF will hold a 0 and will be reset to 1, clearing the overflow. The carry GT of AC0 will be shut.

Just after passing through GT 309.06, the

special-add pulse will also proceed to FF 309.02, setting it to a 1, and via the delay to FF 309.01, resetting it to 0.

The result will be FF 309.01 at 0 and FF 309.02 at 1 for a –1 carry.

If the overflow is a +1, resulting from the addition of two positive numbers, the AC0 partial-sum FF will hold a 1 and will be reset to 0 by the special-add pulse from GT 309.06. The overflow will thus be cleared.

The special-add pulse will also proceed to FF 309.02, setting it to a 1, and to FF 309.01, resetting it to 0, as with a –1 overflow. The pulse added into the AC0 partial-sum FF will, however, find the carry GT open, come out the left digit carry line, set FF 309.01 back to a 1, and reset FF 309.02 to 0. Corrective delays are provided to allow setting and resetting of the flip-flops in sequence.

The result will be FF 309.01 at 1 and FF 309.02 at 0 for a +1 carry. If there has been no carry, both FF 309.01 and FF 309.02 will be 0.

The result has been to perform the first three and part of the fourth of the four required steps listed above. The correct sum has been formed and is in AC. FF 309.01 remembers the existence of a positive carry, and FF 309.02 remembers the existence of a negative carry.

The sum in AC may now be transferred to Storage by a standard *ts* order. The settings of FF's 309.01–2 are not changed by such an operation.

Following the storage of the sum which had been in AC, the first of the large sections to be added is put into the Arithmetic Element with a *ca* order. As described above, this operation consists of clearing AC and adding into it the contents of AR, which have been obtained from the bus.

The following modifications are made in *ca* in order to add in possible carries from special-add.

1) Following the clearing of AC but before the number in AR is put in AC, the special-carry line is pulsed. (See Figure 47.)

If there has been no carry, the pulse will find GT's 309.04 and 309.07 closed because FF's 309.01–2 will both be set to 0. The *ca* operation can then proceed in a normal fashion.

If there has been a +1 carry, the special carry pulse will find GT 309.04 open and will go into the end-around-carry line. Since AC has been cleared to all 0's, the result will be to insert a 1 in the right-most place.

If there has been a –1 carry, the special carry pulse will find GT 309.07 open and will go in and complement all digits except AC15. Since AC has been cleared, the result will be to have all digits 1's except the right-most. The complement representation for -1 will thus have been inserted in AC.

In every instance the special carry pulse resets FF's 309.01–2 to 0.

2) Following the addition of the contents of AR into AC there may be carries if a 1 or -1 had been inserted due to special-add. It is necessary to perform a high-speed carry to obtain the proper sum.

The *sa* operation is now complete.

When the second number is added into AC with an *ad* order, the true sum including the carry from the smaller section will be obtained. The addition of the special carry pulse and a standard carry to the *ca* operation will have no effect during normal additions. FF's 309.01–2 can be on prior to a *ca* operation only if there has been a special-add operation immediately preceding.

### 4.615 Store and Display—*sd*. Figure 41

As far as the Arithmetic Element is concerned, there is no difference between *ts* and *sd*. In both operations the contents of AC are simply read out into the bus. The difference is that the Output display unit as well as the Storage receives the number from the bus in the *sd* order.

The sequence of operations is:

1) Pulse the to-bus line on AC.

No further operations are necessary.

# 5. THE BLOCK DIAGRAMS

## 5.1 The Complete System

### 5.11 The System—Figure 46

The complete system for WWI except for input and output devices is shown in Figure 46. A general discussion of the various elements has been given above. No details are shown; the purpose of the diagram is to show the major units and their interconnections. Often the number of control cables going from one element to another is too large to be shown easily on one small drawing. These multiple lines have been represented by a double line; the actual cables are listed in Figure 47.

The controls going into the computer from some external control desk are:

1) On-Off (controlling the output from the Master Clock).
2) Single-Pulse (provides a single time pulse for step-by-step operation with the Master Clock off).
3) Restart (for initially clearing the computing registers).

### 5.12 Bus Connections—Figure 48

Figure 48 shows the elements connected to the main bus or check bus and the gate tubes which connect them.

The bus system is complicated by the read-back checking. All elements receiving data from the bus must also be able to transmit to the bus for checking reasons. Such elements as the switches are therefore forced to have sets of read-out gate tubes, although these are not needed for the basic operations.

All elements transmitting to the bus for reasons other than checking must also transmit to the check bus. Certain elements, such as the Program Counter, must have three sets of gates, one for receiving from the main bus and one each for transmitting to the main bus and the check bus.

The principle of read-back checking has been described briefly in Section 2.82 and will be considered in more detail in Section 6.1 on Timing.

## 5.2 The Control

### 5.21 The Complete Control—Figure 49

Figure 49 shows the elements of the complete Control. The purposes of the different elements have been described above in Section 3:

The Program Counter holds the register number of the next order to be taken from Storage.

The Program Register serves as a temporary location for an order while the Storage switch is being cleared.

The Master Clock supplies the timing pulses which operate the entire machine.

The Control switch, Time-Pulse Distributor, Operation Matrix, and Operation Timing Matrix form the Operation Control, the functioning of which is described in Section 3.33. The purpose of the Operation Control is to order the different operations by supplying timed pulses to the proper elements.

The Program Timing Matrix supplies those control pulses which are the same for every operation. The Operation Timing Matrix and the Program Timing Matrix use the same time pulses from the Time-Pulse Distributor.

### 5.22 The Master Clock—Figure 50

The Master Clock is expected to consist of a one-megacycle oscillator for supplying the low-frequency clock pulses and a frequency multiplier for increasing the basic frequency to four megacycles for the high-frequency clock pulses. Separate pulse shapers will be provided for high- and low-frequency pulses.

The manual on-off control simply starts or stops the one-megacycle oscillator. With the oscillator off, single clock pulses for step-by-step operation can be obtained from the single-pulse source. The clock will put out a single pulse on both high- and low-frequency clock pulse lines for each operation of the manual single-pulse control. It is not necessary to provide four high-frequency pulses for each low-frequency pulse under manual control, since the computer uses one or the other frequency but never both at the same time.

### 5.23 The Program Counter—Figure 51

Four sections of a possible form of counter are shown. The counting is done by the FF's 01. The add pulse comes in from the left. If the first FF is at 0, it will be switched to a 1, increasing the binary number contained in PC by 1. Note that

in PC as drawn, a number is read in the reverse of the usual order; that is, the smaller end is at the left. If the first FF is a 1, GT 05 will be open and the add pulse can pass through to add into the second FF. The first FF will be reset to 0, after a delay, by the same add pulse. The system is very similar to that used for the Accumulator high-speed carry.

The GT's 01 are used for reading in a new number from the bus in a subprogram operation. Prior to the reading in, the counter must be cleared or reset to 0 by a pulse on the clear line.

It is also possible to change the contents of PC by pulsing the reset line. Toggle switches are provided so that the counter may be set to any desired first order prior to a calculation. The reset ability is part of the test storage setup described in Section 3.35. It will not be needed with electrostatic storage.

The Program Counter must also be able to read out into the bus via GT 02 and into the check bus via GT 03. Buffer amplifiers are provided for the signals going out to either bus. A gate generator FF 02 is used to provide a gate longer than a pulse length for the signals coming in from the bus.

When the counter is full, it will put out an end-carry pulse which may be used for resetting the counter when test storage is used.

The buffer amplifiers on the control lines extending to the right are for driving possible adjacent sections. The general arrangement of buffers, etc., shown may be changed drastically when the final layouts are made, but the general functions of the counter will remain.

### 5.24 The Program Register—Figure 52

Four sections of a possible form of register are shown. The digits are stored in flip-flops provided with gate tubes for reading in and out to the bus. The clear line is pulsed prior to each read-in, in order to clear the register.

### 5.25 32-Position Matrix Switch—Figure 53

This same switch, with minor modifications as noted, is used in both the Operation Control and the test storage.

The switch is shown schematically. Five flip-flops are used to store the five-digit binary number describing the desired output position. GT's 01 are used for reading in the operation part of the order from the bus. GT's 02 and BA's 02 are used for sending the checking read-back signal back to the bus.

The FF's through the BA's 04 drive the crystal matrix. One of the BA's 05 will be selected, and a positive signal will appear on one of the output lines.

The purpose of the change to *sp* (subprogram) line has been described in Section 3.1. The binary codes for the different orders have not yet been established. The *cp* (conditional program) order will, however, have the same code as *sp* except that the fifth digit will be a 0 instead of a 1. The change-to-*sp* pulse will change FF 05 to a 1. The switch will then reset to *sp* from *cp*.

### 5.26 Operation Control—Figures 54 and 55

The two drawings show the Operation Matrix and Operation Timing Matrix of the Operation Control with the Control switch shown as a block. The purpose and use of these matrices are described in Section 3.34.

The connections in the operation matrix are shown made by crystal diodes to avoid feeding signals from one operation line to another. Diodes are not needed in every connection, but are so shown for the sake of consistency.

The connections in the Operation Timing Matrix are direct. There can be no feedback because in no case is there more than one connection from a time-pulse line to a gate tube.

The destinations of all control pulses coming from the matrices are given on the drawing. The connections shown were determined from the timing diagrams to be described in Section 6.0.

### 5.27 Time Pulse Distributor Control—Figure 56

The Time-Pulse Distributor supplies the timing pulses for the Control. This distributor provides only eight pulses, and is not used for controlling the multiple operations such as multiplication, division, and shifting. During the multiple parts of such operations, the Time-Pulse Distributor is off. It is possible to stop the distributor and, upon restarting it, to have it continue to produce pulses in the previous order.

FF 04 is used for starting and stopping the distributor. When FF 04 is set to 0, GT 18 will be

on and low-frequency clock pulses will be supplied to the distributor. When FF 04 is set to 1 by a stop-clock, alarm, or start-delay pulse, GT 18 will be shut. No further time pulses will be produced by the distributor. It will, however, remain at its setting at the time it was stopped. When GT 18 is reopened, the next clock pulse supplied will come out the same line as though there had been no delay. The pulse distributor operation is independent of pulse repetition frequency (PRF). This independence of PRF also results in proper operation during step-by-step or single-pulse operation.

A multiply, divide, or shift pulse sent to the Arithmetic Element, as described in Section 4.65, is also sent in on the stop-clock line. When the Arithmetic Element has finished its part of the operation, the Step Counter will produce an end-carry which not only shuts off the supply of clock pulses to AE but resets FF 04 to 0, restarting the Time-Pulse Distributor.

It is not now known how much time will be required to set the switches on the test storage. Several microseconds may be required. The time schedule for electrostatic storage is also unkown at this time. FF 05, GT 19, and the delay counter provide the necessary flexibility to allow the storage setup time to vary within expected limits without affecting either the Control or the control timing. Whenever the Storage Switch is set (as shown here, this will be on TP's 2 and 5), a start-delay pulse will be sent to FF 04, shutting off the supply of pulses to the distributor. The start-delay pulse will also set FF 05 to 0, opening GT 19, allowing clock pulses to reach the Delay Counter. This counter may be set to produce an end-carry after any desired number of pulses. The end-carry will reset FF 04 to 0 and FF 05 to 1, restoring the original condition. The effect of the Delay Counter is simply to introduce an adjustable delay each time the Storage Switch is set up.

Whenever the checking circuits of the computer detect an error, an alarm pulse will be sent to the Control. This alarm pulse will set FF 04 to 1, stopping the Time-Pulse Distributor, and keeping the situation as it was at the time of the error. The alarm pulse will also set FF 06 to 1, putting out a gate to the warning signal circuit which will inform the operator of the error.

The manual-restart line permits the distributor to be restarted after a fault is corrected.

### 5.28 Pulse Distributor — Figure 57

One possible form of the Pulse Distributor is shown in this drawing. It consists of an 8-way diode matrix switch with the three driving FF's connected in a counter circuit. Each time pulse, as it appears, will change the contents of the counter register to a different number greater by 1. The switch output will thus change for each time pulse.

A gate tube and buffer amplifier are connected to each of the switch outputs. The time pulses coming in are supplied to the grids of all these tubes. The switch will supply a gate to only one tube at a time; the time pulse will thus appear on the selected output only. A delay is provided in the input to the counter to delay the change in the switch position until after the time pulse has gone out the output line.

### 5.29 Program Timing Matrix — Figure 58

This matrix supplies those pulses which are identical for all operations. The time pulses from the Pulse Distributor are supplied at the left to the matrix. The Program Timing Matrix may be thought of as an extension of the Operation Timing Matrix, with the gate tubes omitted since they would be gated all the time.

## 5.3 Test Storage

### 5.31 Storage Chassis Arrangement — Figure 59

This drawing shows a possible arrangement of the elements of test storage. Five flip-flop registers are provided, each divided into four sections of four digits each. Twenty-seven toggle-switch registers are provided, the desired one of the 32 registers being selected by the 32-position storage switch.

A single output member, also divided into four sections, is provided. The outputs of flip-flop and toggle-switch registers are mixed in this member and put on the bus by a single set of drivers.

Provision is also made for reading out to the check bus and for reading into flip-flop storage from the main bus.

### 5.32 Flip-Flop Storage — Figure 60

One 4-digit section is shown. The number is stored in the FF's 01.

When this particular register is selected by the Storage Switch, a selector gate from the switch output will be supplied to GT 04 and GT 05. If a

FF is set to a 1, a gate will be supplied to the Storage output member through the corresponding BA 04. If a FF is set to 0, no gate will be supplied.

When a new number is to be stored, a clear pulse is sent to the selected register. The selector gate opens GT's 05 as well as GT's 04. A number coming in the in line will then be stored in the FF's 01.

The initial-value line allows insertion of initial values in the flip-flop registers prior to a test problem. The single-pole double-throw toggle-switches are set to the desired number for each register. When the initial-value line is pulsed, this number will be inserted in the register. The initial-value line is common to all flip-flop registers. Thus, all registers are set at once independent of the Storage Switch setting.

The in and out lines are common to all registers, the connections to the lines being determined by the selector gate from the storage switch. The BA's 04 are needed on the botton section only, the GT's 04 presumably being able to drive the out line.

### 5.33 Storage Output Section — Figure 61

When a number is to be stored in the Flip-Flop Storage, a gate is supplied to GT's 01 on the storage-read-in line. Pulses proceeding along the bus are thus sent through GT 01, BA 01, and on the in line to the Flip-Flop Storage.

For reading out, a gate is supplied from the selected register in either flip-flop or toggle-switch storage to GT's 02. When the storage-read-out line is pulsed, the contents of the selected register will be transmitted onto the bus. The gate is supplied to GT's 03, as well as to GT's 02, so that the number may be read out onto the check bus by pulsing the storage-check line.

### 5.34 Flip-Flop Storage Control — Figure 62

The Storage Control takes the switch gate selecting one of the Flip-Flop Storage registers and generates the selector gate mentioned in Section 5.32. The switch gate is also supplied to GT's 07, which send the clear pulse to the proper register before a number is stored in that register.

The storage-reset pulse is sent to all flip-flop registers through the buffer amplifiers BA 08.

### 5.35 Toggle-Switch Storage

No drawing is given for the switch array.

## 5.4 The Arithmetic Element

### 5.41 The Complete Element — Figure 63

This drawing and the purposes of the elements shown have been discussed in detail in Section 4.6.

### 5.42 Section of A-Register — Figure 64

The number in AR is stored in the FF's 01.

A number is sent into AR through the GT's 01. The FF's must first be cleared.

A number in AR may be read back onto the bus for checking by means of GT's 02 and BA's 02.

The number in AR can be read out to AC through GT's 05 and BA's 05 or its complement can be read out to AC through GT's 04 and BA's 04. The gate tubes GT 04 are connected to the other sides of the flip-flops from the gate tubes GT 05.

A complement line is provided which will switch all the FF's, essentially changing the sign of the number in AR when required for sign handling.

### 5.43 Accumulator Sections — Figure 65

This drawing shows the center sections of AC. The end sections are somewhat different and are described separately in Section 5.44.

The partial sum or actual digits of the number in AC are stored in the FF's 01. The carries are stored in the FF's 02. A single digit section includes one partial-sum and one carry flip-flop. For construction reasons, the carry flip-flops are associated physically with the partial-sum flip-flops one space to the left, and are so shown in the drawing. The dashed lines enclose a complete digit section.

For addition, the incoming pulse from AR is supplied to DE 01 and GT 06. If the FF 01 holds a 1, the incoming pulse will proceed along the carry digit line to add 1 into the corresponding carry flip-flop. The pulse will also proceed through the delay to switch the partial-sum flip-flop. This method is described in Section 4.41. The delay DE 01 is made just long enough to allow the carry pulse to pass through GT 06 before FF 01 is changed to a 0. A high-speed-carry system using GT 12 and GT 05 is included.

The contents of AC may be read out to the main bus through GT 02, to BR through GT 04, and to the check bus through GT 03.

Shifting left is accomplished by means of the GT's 13 and 14. GT 14 will be on if the FF 01 holds a 0, while GT 13 will be on if the FF 01 holds a 1. A shift-left pulse will pass through the on tubes and set the adjacent flip-flop on the left to agree with the setting of the original flip-flop. The delays DE 02 and DE 03 are used to delay the shifting pulses until the shift-left pulse has gone completely through all GT's 13 and 14.

A clear line is provided for clearing AC prior to inserting a number. A complement line is provided for correcting the sign of the number in AC. Crystal diodes are provided to keep control pulses from feeding back into other control lines. .

The diode matrix is used for the shift-right-and-carry operation. The general process is described in Section 4.43. A 4-way diode-matrix switch is used to convert the original settings of the partial-sum and carry digit flip-flops into control pulses for properly setting the flip-flops during shift-and-carry. The switch is driven from the partial sum and carry flip-flops. These flip-flops, FF's 01 and 02, belong to different digit sections, but have been shown placed on one chassis in order to ease the connection problem.

One of the GT's 08, 09, 10, 11, is selected by the switch. The shift-and-carry pulse, which is supplied to all four tubes, proceeds through the selected one to set the partial-sum flip-flop. GT 07 is used for resetting the carry flip-flop when necessary.

### 5.44 Accumulator Sections AC0–AC15 — Figure 66

This drawing shows the end sections of AC, which differ somewhat from the middle sections. The additions to the AC0 section include a connection between the high-speed-carry line (carry-digit-from-right) and the left-digit-carry line described in Section 4.62.

A to-AC0 line is provided for subtracting the carry in special-add. See Section 4.614. The only other difference is a special divide-shift-left line which shifts AC0 and AC1 left during division. These digits are not ordinarily shifted.

The right-most digit, AC15, differs largely in the omission of the carry flip-flop. The carry flip-flop in the same digit section as AC15 is actually part of the AC14 chassis. Since there is no carry flip-flop, the shift-and-carry is reduced to a simple shift-right to BR0 involving only GT15 and GT16.

A –1 special-add line is added to complement all sections except AC15 if a –1 carry results from special add.

The from-divide-shift-left line is used to obtain a full shift as well as the shift of the left two digits only.

### 5.45 B–Register Section — Figure 67

The B–Register stores a number in the FF's 01. A clear line is provided, and also a means for reading in from AC. The BR cannot read out except by shifting left into AC.

Two pairs of gate tubes with delays are used for shifting right and left from each digit to its neighbor.

### 5.46 B–Register Sections BR0, 1, 14, 15 — Figure 68

An extra gate tube GT 08 is added to BR0 for providing roundoff. A divide-shift-left line and a diode are used to provide for shifting all BR sections except BR0 on divide.

The only other difference is the addition of the line connecting from-BR15(+) to the 0 side of BR15 through the delay DE 05. The line is used for resetting BR15 during multiplication. See Section 4.65. The regular shift-right gate tubes GT 07 and GT 06 are used to get the from–BR15 signals necessary for multiplication.

### 5.47 Step Counter — Figure 69

The Step Counter is used in the Arithmetic Element to count the number of steps performed in shifting, multiplying, and dividing. Five sections are shown, allowing the counter to count to 32. Since there are 17 steps to the division process, the fifth section is required.

In divide and multiply, the counter is reset by a pulse on the corresponding line. The toggle switches are set to give the desired initial setting of the counter. The switches are used to allow convenient changing of counter settings for changed procedures or testing of arithmetic elements with other than 16 digits. After the proper number of add pulses have been supplied, the counter flip-flops will all have returned to 0. The 0 sides of the flip-flops are mixed through diodes and supplied to the end-carry GT 04. The next add pulse will pass through this gate tube and out the end-carry line.

For the shift operation, the counter is set

according to the number of digits to be shifted by a code order coming in from the bus. See Section 4.69. The counter is first cleared. The order is then read into the counter from the bus through GT's 01. GT's 02 are used for reading the counter contents back onto the bus for checking if the operation is a shift.

## 5.5 Check Register — Figure 70

The check register is made up of FF's 01 which can read in from both main bus and check bus. A number on the main bus may be read into the register through GT's 01, while a number coming down the check bus is always added into it.

If the numbers coming in from the main bus and the check bus are identical, all FF's 01 will end up at 0. If there is any discrepancy, one of the flip-flops will hold a 1, GT 04 will be on, and a check-CR pulse will produce an alarm.

The register is self-clearing in the case of no error. A manual-clear line is provided for clearing the register after correction of a fault.

# 6. TIMING DIAGRAMS

## 6.1 Program Timing

The timing diagrams are given in Figures 72 through 86. Each diagram lists all elements of the computer which receive control pulses, plotted against the time pulses from the Time Pulse Distributor. The required control pulses are shown at the proper time for each operation. The sequence of all the operations may be checked on these drawings. They may also be used for choosing the connections on the Timing and Control Matrices.

The upper section of each diagram, labelled Program Timing, is the same for each operation. This section includes the timing pulses necessary for order manipulation and setup of each operation. The Operation Timing section of each diagram lists those control pulses which are required for the particular operation under consideration.

The sequence of operations carried out by the Program Timing is described below. Figure 71 shows these pulses only.

**TP No. 1** The Storage Switch is cleared.

**TP No. 2** The contents of the Program Counter are sent onto the bus. The gates on the input to the Storage Switch are opened and the switch receives the order register number stored in the counter.

The contents of the Program Counter are also sent onto the check bus and are received by the Check Register.

A pulse is also produced which turns off the main Time-Pulse Distributor and starts the Delay Counter. The operation of the computer will cease until the storage selection has had time to set up.

When the Delay Counter has counted the preset number of clock pulses, it will produce an end-carry which will restart the main Time-Pulse Distributor.

**TP No. 3** The contents of the Storage Switch are sent on to the main bus. The gates on the Check Register are opened, and the switch contents added in on top of the number received from the Program Counter via the check bus. If the numbers are identical, the register will end up holding all 0's. The transfer-check-pulse, which is delayed slightly to allow completion of the register operation, then checks to see if the register does hold all 0's.

The Program Register is cleared to make it ready to receive the order about to be read out from Storage.

An add pulse is sent to the Program Counter.

**TP No. 4** The storage is read out onto the bus. The gates on PR are opened and the order from Storage read into it.

The Storage is also read out onto the check bus and into the Check Register.

The Control Switch and Storage Switch are cleared prior to receiving the order from PR. The clearing of each switch is delayed one-half a pulse to allow completion of reading from Storage and generation of a control pulse on TP No. 4 by the Operation Control.

The Step Counter is cleared and made ready to receive order information.

**TP No. 5** The contents of PR are read out onto the bus. The gate tubes on the Control Switch, Storage Switch, and Step Counter are opened, and the appropriate parts of the order read into these elements.

The A-Register is cleared and made ready to receive a number from Storage, if desired.

The Delay Counter is again started to give the storage sufficient time to set up. After the desired delay, the main Time-Pulse Distributor will again start.

**TP No. 6** The contents of the Control Switch, Storage Switch, and Step Counter are read onto the bus and received by the Check Register. A transfer check examines the correctness of the transfer of the order from Storage to the Control Switch, Storage Switch, and Step Counter via the Program Register.

**TP No. 7** The Operation Control takes over and supplies the necessary pulses for storage manipulation for the particular operation being carried out.

**TP No. 8** Same comments as TP No. 7— The transfer check is used to check any transfer ordered by the Operation Control.

The distributor will next put out TP No. 1 and start the cycle over again.

## 6.2 Operation Timing

### 6.201 Timing for Add–*ad*. Figure 72

A description of this operation will be found in Section 4.602.

The program timing has been described in Section 6.1. The Control Switch was set by the order on TP No. 5 and the setting checked on TP No. 6. The Operation Control then takes over.

**TP No. 6** No action.

**TP No. 7** The Storage is read out onto the bus and the number received by the A-Register. The Storage is also read onto the check bus to the Check Register for the transfer check.

**TP No. 8** The number in AR is added into the Accumulator. The number in AR is also sent onto the bus and received by the Check Register where it is added to the original number sent from Storage. The result is checked for correct transmission.

All eight time pulses from the main Distributor have now been used. The Control Switch is not reset, however, until the next TP No. 5, and the Arithmetic Element is not needed for the next operation until TP No. 7. TP's 1 through 4 are thus available for the present operation.

**TP No. 1** No action. The carry could be accomplished with this pulse but is deferred until TP No. 2. When a given control pulse is required in many different orders the same time pulse should be used for the operation, if possible, in order to simplify the control matrices.

**TP No. 2** High-speed carry.

**TP No. 3** Arithmetic check to check for overflow.

**TP No. 4** No action.

### 6.202 Timing for Clear and Add–*ca*. Figure 73

A description of this operation will be found in Section 4.601.

**TP No. 6** Clear AC.

**TP No. 7** Read out from Storage into AR. The Storage is also read onto the check bus for transfer to CR. Special carry for adding in any carry resulting from special add.

**TP No. 8** The contents of AR are sent into AC and also back onto the bus for checking.

**TP No. 1** No action.

**TP No. 2** Carry. A high-speed carry is required because a carry may result from a special-add.

**TP No. 3** Arithmetic check.

### 6.203 Timing for Subtract–*su*. Figure 74

A description of this operation will be found in Section 4.604.

**TP No. 6** No action.

**TP No. 7** Read-out from Storage into AR and the Check Register.

**TP No. 8** Send the complement of the number in AR into AC. Send the number in AR back onto the bus for checking.

**TP No. 1** No action.

**TP No. 2** High-speed carry.

**TP No. 3** Arithmetic check.

### 6.204 Timing for Clear and Subtract–*cs*. Figure 75

A description of this operation will be found in Section 4.603.

**TP No. 6** Clear AC.

**TP No. 7** Read-out from Storage into AR. The Storage is also read out into the check bus.

**TP No. 8** The complement of the number in AR is sent into AC. The number in AR is also sent back into the bus for checking.

**TP No. 1** No action.

**TP No. 2** No action.

**TP No. 3** Arithmetic check.

### 6.205 Timing for Multiply and Roundoff–*mr*. Figure 76

A description of this operation will be found in Section 4.605.

**TP No. 6** The sign of AC is checked and BR is cleared.

**TP No. 7** Read-out from Storage into AR. The Storage is also read out onto the check bus.

The multiplier in AC is transferred to BR.

**TP No. 8** The contents of AR are sent back to the Check Register for checking.

**The sign of AR is checked.** The check is delayed one-half pulse to allow reading AR back to bus for checking.

**TP No. 1** AC is cleared. The multiply line is pulsed, turning control of the Arithmetic Element over to the multiply control and stopping the main Time-Pulse Distributor. When the successive addition part of the multiplication is complete, the Step Counter will put out an end-carry pulse which will restart the main Distributor.

**TP No. 2** A high-speed carry and roundoff are needed to finish the product in AC.

**TP No. 3** The B-Register is cleared.

**TP No. 4** The product-sign pulse corrects the sign of AC.

### 6.206 Timing for Multiply and Hold Full Product–*mh* — Figure 77

A description of this operation will be found in Section 4.606.

**TP No. 6** The sign of AC is checked, and BR is cleared.

**TP No. 7** Read-out from Storage into AR. The Storage is also read out into the check bus.

The multiplier in AC is transferred to BR.

**TP No. 8** The contents of AR are sent back to the Check Register for checking.

**The sign of AR is checked.** The check is delayed one-half pulse to allow reading AR back to bus for checking.

AC is cleared.

**TP No. 1** Multiply.

**TP No. 2** High-speed carry but no roundoff.

**TP No. 3** The B-Register is not cleared, since the second half of the product is in the register.

**TP No. 4** Product-sign pulse corrects the sign of AC.

### 6.207 Timing for Divide–*dv*. Figure 78

A description of this operation will be found in Section 4.607.

**TP No. 6** Check sign of dividend in AC.

**TP No. 7** The Storage is read out onto the bus and the number received by AR. The Storage is also read out into the Check Register via the check bus.

**TP No. 8** The number in AR is sent back onto the bus and to the Check Register for checking. After a delay of one-half pulse the sign of AR is checked.

**TP No. 1** The divisor in AR is subtracted once from the dividend in AC.

**TP No. 2** The divide pulse stops the main Time-Pulse Distributor and turns control over to the divide Time-Pulse Distributor in the Arithmetic Element. When the successive subtraction part of the division operation is complete, the Step Counter will put out an end-carry pulse which will restart the main Distributor.

**TP No. 3** Clear AC.

**TP No. 4** Correct the sign digit of AC, if the quotient should be negative.

### 6.208 Timing for Transfer to Storage–*ts*. Figure 79

A description of this operation will be found in Section 4.608.

**TP No. 6** Clear storage. The Flip-flop Storage control clears only the register selected by the Storage Switch. This pulse is necessary for test storage only.

**TP No. 7** The contents of AC are sent onto the bus and received by the Storage. The contents of AC are also sent to the Check Register.

**TP No. 8** The Storage is read back onto the bus and sent to the Check Register for transfer check.

**TP No. 1** No action.

**TP No. 2** No action.

**TP No. 3** No action.

### 6.209 Timing for Shift Right–*sr*. Figure 80

A description of this operation will be found in Section 4.609.

**TP No. 6** The sign of AC is checked.

**TP No. 7** No action.

**TP No. 8** No action. The Transfer-check pulse in the program timing does no harm since the Check Register remains clear when there are no transfers.

**TP No. 1** A shift-right pulse is sent to the Arithmetic Element. The main Time-Pulse Distributor is shut off until the shift is completed. The end-carry from the Step Counter restarts the main Distributor.

**TP No. 2** Roundoff the shifted number.

**TP No. 3** Clear BR.

TP No. 4 The sign of AC is corrected, if necessary.

### 6.210 Timing for Shift Left–*sl.* Figure 81

A description of this operation will be found in Section 4.610.

TP No. 6 The sign of AC is checked.

TP No. 7 No action.

TP No. 8 No action.

TP No. 1 Shift left.

TP No. 2 Roundoff the shifted number.

TP No. 3 Clear BR. Arithmetic check.

TP No. 4 The sign of AC is corrected, if necessary.

### 6.211 Timing for Subprogram–*sp.* Figure 82

A description of this operation will be found in Section 4.611.

TP No. 6 Clear Program Counter.

TP No. 7 Send the contents of PR onto the bus again. Open the Program Counter gates to receive the order from the bus. This transfer is not checked.

TP No. 8 through TP No. 4 No action.

### 6.212 Timing for Conditional Program–*cp.* Figure 83

A description of this operation will be found in Section 4.612.

TP No. 6 Compare pulse to the Arithmetic Element.

If the number in AC is positive, the compare pulse will return as a change-control pulse which will clear the Program Counter and change the Operation Switch to *sp.* The Control will then proceed as in *sp.* If the number in AC is negative, there is no further action.

### 6.213 Timing for Transfer Digits–*td.* Figure 84

A description of this operation will be found in Section 4.613.

TP No. 6 No action.

TP No. 7 The contents of the Accumulator are sent onto the bus and received by the Storage. The contents of AC are also sent to the Check Register.

TP No. 8 The Storage is read back onto the bus and sent to the Check Register for transfer check.

TP No. 1 through TP No. 4 No action.

### 6.214 Timing for Special Add–*sa.* Figure 85

A description of this operation will be found in Section 4.614.

TP No. 6 No action.

TP No. 7 The Storage is read out onto the bus and the number received by AR. The Storage is also read out into the Check Register via the check bus.

TP No. 8 The number in AR is added into AC. The number in AR is also sent back onto the bus for the transfer check.

TP No. 1 No action.

TP No. 2 High-speed carry

TP No. 3 The special-add pulse corrects the sum for overflow and sets the flip-flops for storing the overflow.

TP No. 4 No action.

### 6.215 Timing for Store and Display–*sd.* Figure 86

A description of this operation will be found in Section 4.615.

TP No. 6 Clear Storage.

TP No. 7 The contents of AC are sent onto the bus and received by the Storage. The contents of AC are also sent to the Check Register.

The gate tubes leading to the display unit are also opened to allow reception of the number on the bus.

TP No. 8 The Storage is read back onto the bus and sent to the Check Register for transfer check. The transfer to display is not checked.

TP No. 1 No action.

TP No. 2 No action.

TP No. 3 No action.

# GLOSSARY

| Term | Abbreviation | Meaning |
|---|---|---|
| Accumulator | AC | The adding unit of the Arithmetic Element. |
| Arithmetic Element | AE | The part of the computer that performs the actual arithmetic operations. |
| A-Register | AR | The register in the Arithmetic Element used to hold numbers coming into the Arithmetic Element from the bus. |
| Binary | | Using the base 2. |
| Buffer Amplifier | BA | An electronic amplifier used to amplify a pulse to a level high enough to drive a coaxial line or a large number of gate tubes. |
| Bus | | A group of conductors used for transmitting a complete number or order. |
| Bus Driver | BD | An amplifier for sending a pulse onto a bus line. Its characteristics are low impedance for driving the line and high impedance when the line is being used for other purposes. |
| B-Register | BR | The register in the Arithmetic Element used for holding the multiplier, etc., during arithmetic operations. |
| Check Bus | | A second bus used for checking purposes only. |
| Check Register | CR | The special register provided for the transfer check. |
| Control | | That part of the computer which controls the operation of the Storage and the Arithmetic Element. |
| Delay Counter | | A binary counter used in the control to stop the operation of the computer for a time sufficient to allow the storage switch to set. |
| Delay Element | DE | A delay line used for delaying pulses for corrective purposes or to avoid conflicts. |
| Flip-Flop | FF | A two-tube electronic device of which either one tube or the other is conducting but not both. According to which of the tubes is conducting, the flip-flop is said to be storing a 0 or a 1. |
| Flip-flop Storage | | The 5 flip-flop registers used in test storage as high-speed memory elements. |

## GLOSSARY

| Term | Abbreviation | Meaning |
|---|---|---|
| Gate Tube | GT | A multigrid tube which will conduct only if positive voltages are supplied to all grids. Only two-grid tubes are considered in this report. |
| Input | | The equipment used for supplying information to the computer. |
| Main Bus | | The bus composed of many parallel cables used for transmitting the numbers and orders actually used by the computer. |
| Master Clock | | The primary source of the pulses which are used to operate the computer. |
| Operation Control | | The part of the complete Control which directs the arithmetic part of each operation. |
| Operation Control Switch | | The switch which selects the operation to be performed. |
| Operation Matrix | | The array of connections which gates the control gate tubes according to the selected position of the Operation Control Switch. |
| Operation Timing Matrix | | The array of connections which supplies the selected control gate tubes with the proper time pulses. |
| Output | | The equipment used for extracting information from the computer. |
| Parallel Transmission | | The system of data transmission in which the digits of a number are transmitted simultaneously over separate lines. |
| Program Control | | The part of the Control which sets up each operation prior to its arithmetic part. |
| Program Counter | PC | The part of the computer that selects the next order to be performed. |
| Program Register | PR | The part of the computer used for holding orders after they are extracted from Storage but before they are carried out. |
| Pulse Repetition Frequency | PRF | Number of pulses generated per second. |
| Program Timing Matrix | | The array of connections used to supply the correct timing pulses for setting up an operation. |

# GLOSSARY

| Term | Abbreviation | Meaning |
|---|---|---|
| Register | | A group of elements used in the machine to store a single number or order. |
| Scale Factor | | The multiplier associated with each number in the machine used to force these numbers to occupy the limited range of a machine register. |
| Serial Transmission | | The system of data transmission in which the digits of a number are transmitted in a sequence over a single line. |
| Sign-Control Flip-flop | | The flip-flop used in the Arithmetic Element for storing the proper sign of the result of an operation. |
| Step Counter | | The binary counter used in the Arithmetic Element to count the steps in multiplication, division, and shift operations. |
| Storage | | That part of the computer that holds the numbers and orders used by the computer. It is made up of a number of storage registers. |
| Storage Switch | | The switch or group of switches used for selecting the desired register in the storage. |
| Subprogram | | A subsidiary sequence of orders which may be interpolated in the main sequence of orders whenever desired. |
| Time Pulse Distributor; Time Pulse | TP | A device for distributing clock pulses on to a set of lines in some fixed sequence; the resulting pulses. |
| Toggle-switch Storage | | The 27 toggle-switch registers making up part of the test storage. |
| Whirlwind I | WWI | The prototype electronic computer now being designed. |
| Whirlwind II | WWII | The final electronic computer to be designed later. |
| Word | | The several digits making up a number or order. |

# LIST OF DRAWINGS

| Figure Number | Drawing Number | Title |
|---|---|---|
| 1 | A–30339 | General Block Diagram |
| 2 | A–30445 | Store Result |
| 3 | A–30440 | Origin of Orders |
| 4 | A–30441 | Setup Order |
| 5 | A–30442 | Read Out Order |
| 6 | A–30443 | Setup Operation |
| 7 | A–30444 | Perform Operation |
| 8 | A–30446 | Subprogram |
| 9 | A–30454 | Electronic Switch |
| 10 | A–30452 | Time Pulse Distributor |
| 11 | A–30453 | Control |
| 12 | A–30400 | Decimal Addition |
| 13 | A–30401 | Binary Addition |
| 14 | A–30402 | A Binary Adder |
| 15 | A–30417 | Negative Numbers |
| 16 | A–30410 | Subtraction Using 9's Complements |
| 17 | A–30403 | Decimal Multiplication |
| 18 | A–30355 | Binary Notation |
| 19 | A–30409 | Binary Multiplication |
| 20 | A–30404 | Modified Binary Multiplication |
| 21 | A–30447 | Multiplication I |
| 22 | A–30448 | Multiplication II |
| 23 | A–30449 | Multiplication III |
| 24 | A–30450 | Multiplication IV |
| 25 | A–30451 | Multiplication V |
| 26 | A–30406 | Rounding Off |
| 27 | A–30683 | Decimal Division |
| 28 | A–30686 | Binary Division |
| 29 | C–37072 | Arithmetic Element |
| 30 | A–30433 | Accumulator |
| 31 | A–30432 | Accumulator |
| 32 | A–30685 | Shift and Carry |
| 33 | A–30684 | Shift Left |
| 34 | C–37099 | Clear and Add |
| 35 | C–37100 | Add |
| 36 | A–30682 | Arithmetic Check |
| 37 | C–37101 | Clear and Subtract |
| 38 | C–37102 | Subtract |
| 39 | C–37103 | Multiply and Roundoff |
| 40 | C–37104 | Divide |
| 41 | C–37105 | Transfer to Storage |
| 42 | C–37106 | Shift Right |
| 43 | C–37107 | Shift Left |
| 44 | C–37108 | Conditional Program |
| 45 | C–37109 | Special Add |
| 46 | C–37071 | System Block Diagram |
| 47 | B–37073 | Control Functions |
| 48 | B–37070 | Bus Connections |

# LIST OF DRAWINGS

| Figure Number | Drawing Number | Title |
|---|---|---|
| 49 | B-37098 | Control |
| 50 | B-37058 | Master Clock |
| 51 | B-37062 | Program Counter |
| 52 | B-37067 | Program Register |
| 53 | B-37066 | Control Switch |
| 54 | C-37077 | Operation Matrix I |
| 55 | C-37078 | Operation Matrix II |
| 56 | B-37076 | Time Pulse Distributor Control |
| 57 | B-37068 | Pulse Distributor |
| 58 | B-37075 | Program Timing Matrix |
| 59 | C-37064 | Storage Chassis Arrangement |
| 60 | B-37057 | Flip-flop Storage Section |
| 61 | B-37060 | Storage Output Section |
| 62 | B-37061 | Flip-flop Storage Control |
| 63 | C-37072 | Arithmetic Element |
| 64 | B-37056 | Section of A-Register |
| 65 | C-37063 | Accumulator Sections |
| 66 | C-37096 | Accumulator Sections |
| 67 | B-37069 | B-Register Sections |
| 68 | B-37097 | B-Register Sections |
| 69 | B-37074 | Step Counter |
| 70 | B-37065 | Check Register |
| 71 | A-30874 | Program Timing |
| 72 | B-37080 | Timing for Add |
| 73 | B-37081 | Timing for Clear and Add |
| 74 | B-37082 | Timing for Subtract |
| 75 | B-37083 | Timing for Clear and Subtract |
| 76 | B-37084 | Timing for Multiply and Roundoff |
| 77 | B-37085 | Timing for Multiply and Hold Full Product |
| 78 | B-37094 | Timing for Divide |
| 79 | B-37086 | Timing for Transfer to Storage |
| 80 | B-37088 | Timing for Shift Right |
| 81 | B-37089 | Timing for Shift Left |
| 82 | B-37090 | Timing for Subprogram |
| 83 | B-37091 | Timing for Conditional Program |
| 84 | B-37092 | Timing for Transfer Digits |
| 85 | B-37093 | Timing for Special Add |
| 86 | B-37087 | Timing for Store & Display |
| 87 | B-37001 | Parallel Digit Computer Codes |

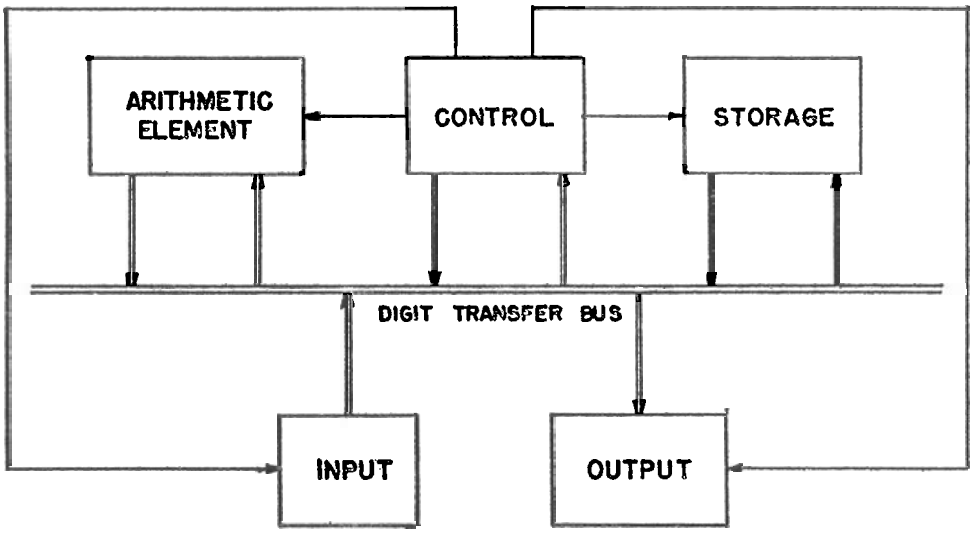


Figure 1

GENERAL BLOCK DIAGRAM

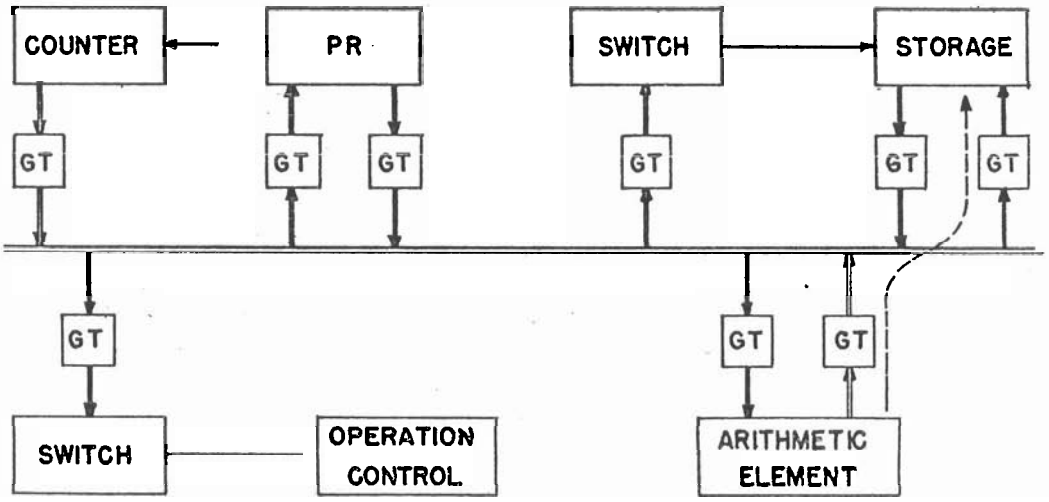


Figure 2

STORE RESULT

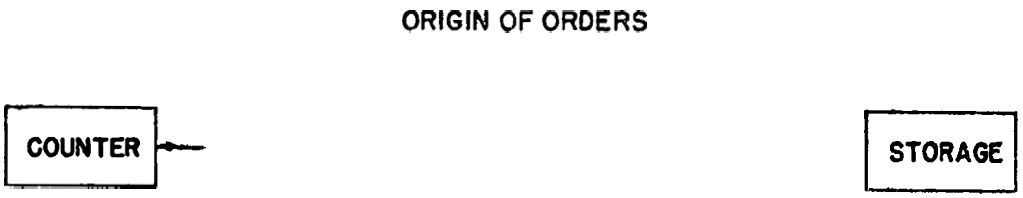


Figure 3
ORIGIN OF ORDERS

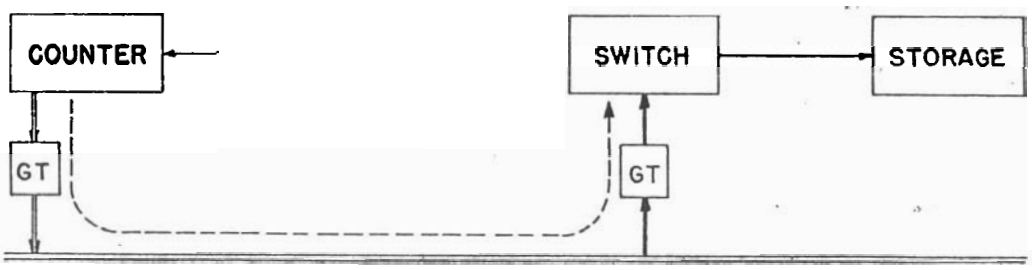


Figure 4
SETUP ORDER

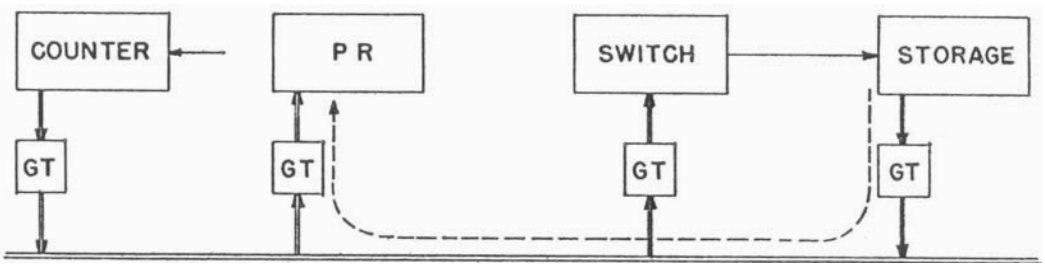


Figure 5
READ OUT ORDER

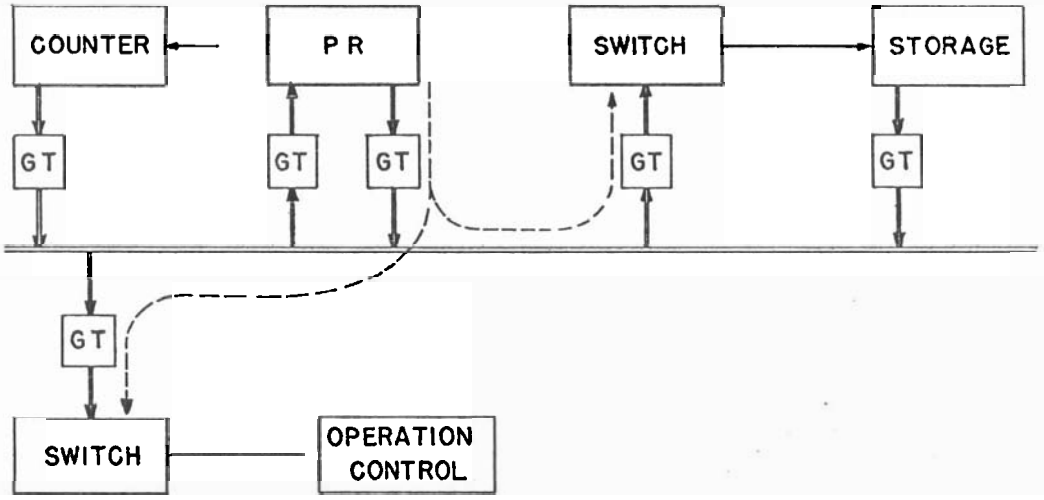


Figure 6
SETUP OPERATION

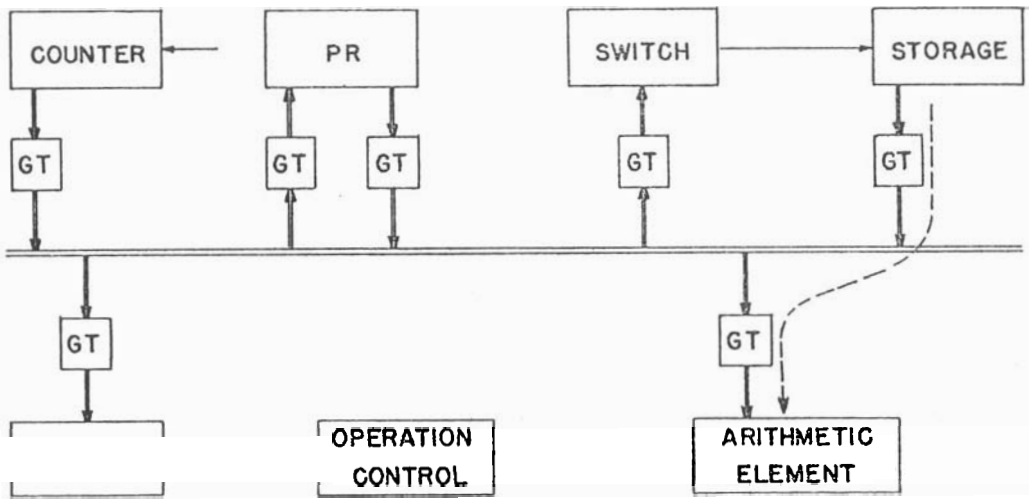


Figure 7

PERFORM OPERATION

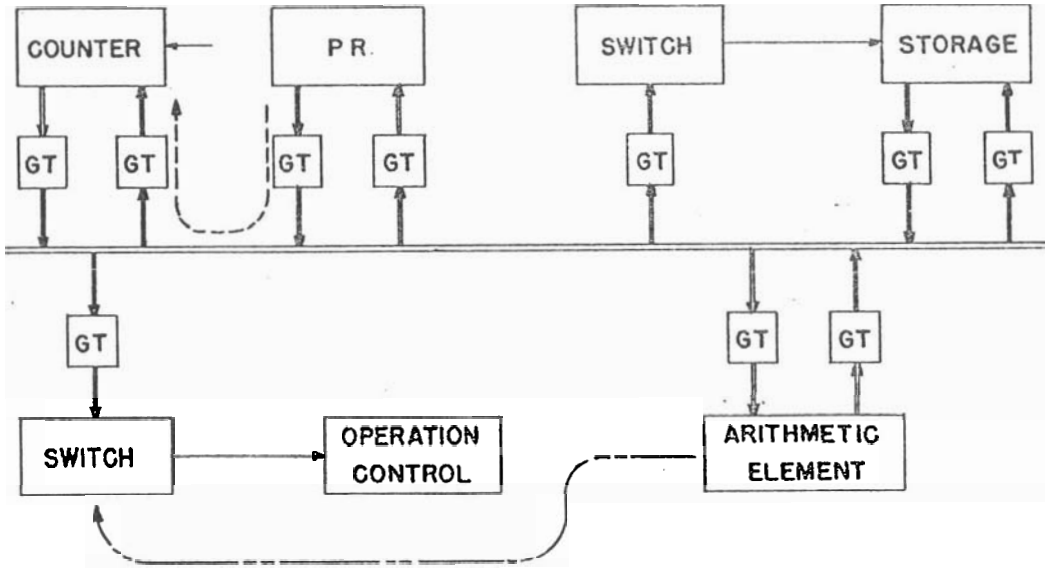


Figure 8

SUBPROGRAM

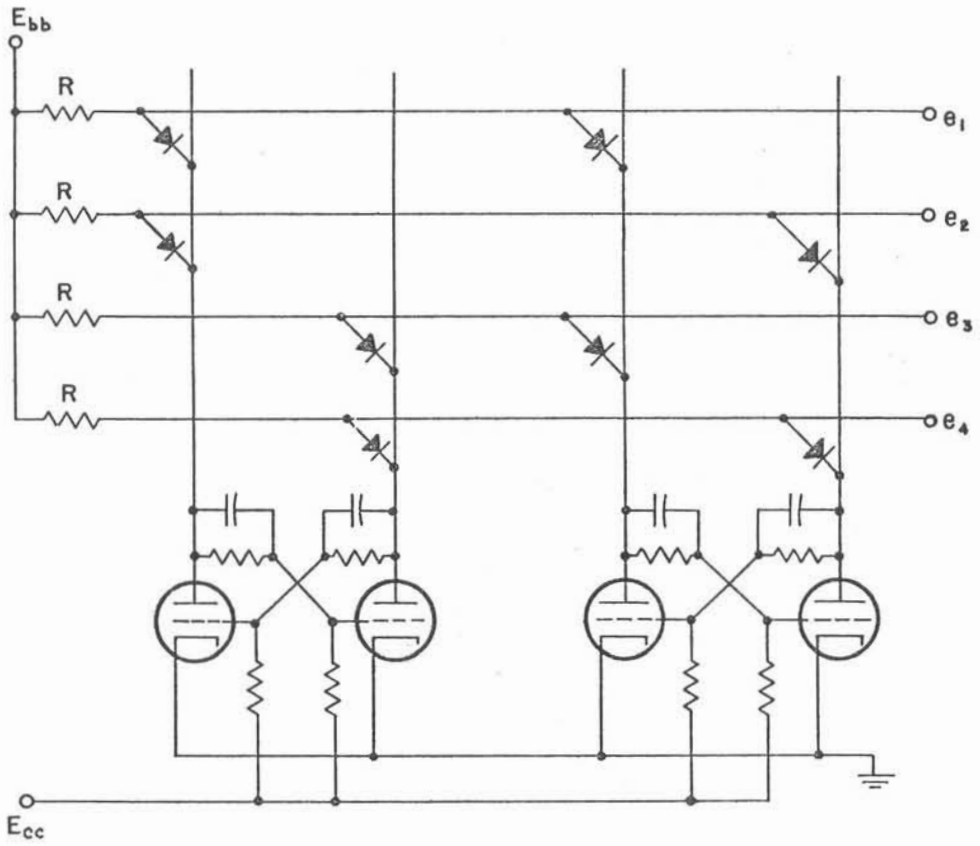


Figure 9
ELECTRONIC SWITCH

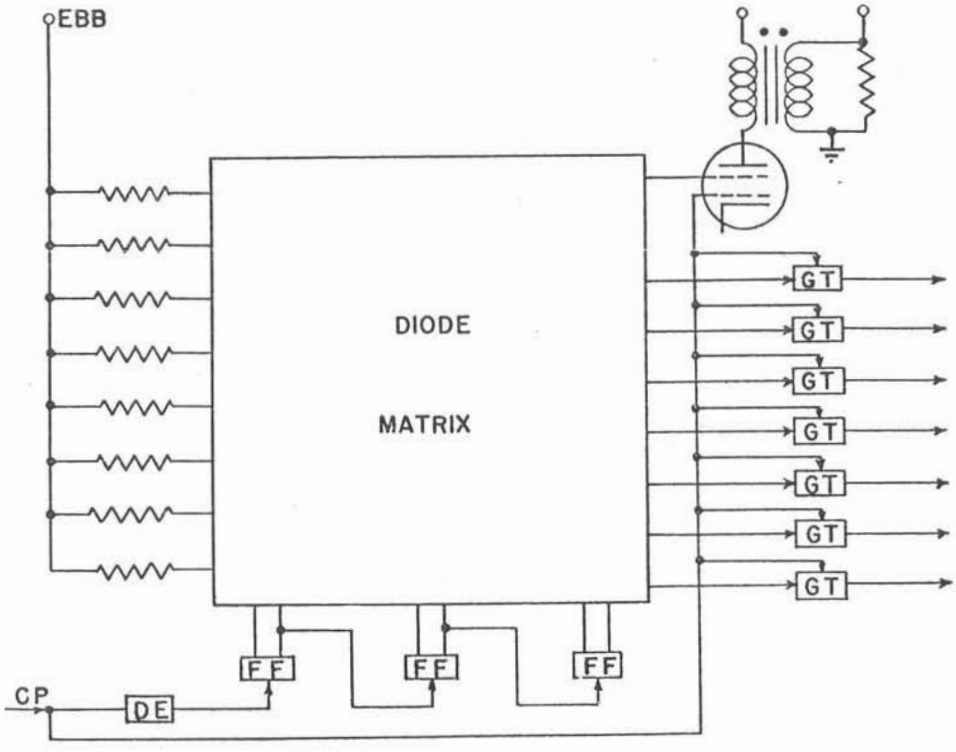


Figure 10
TIME PULSE DISTRIBUTOR

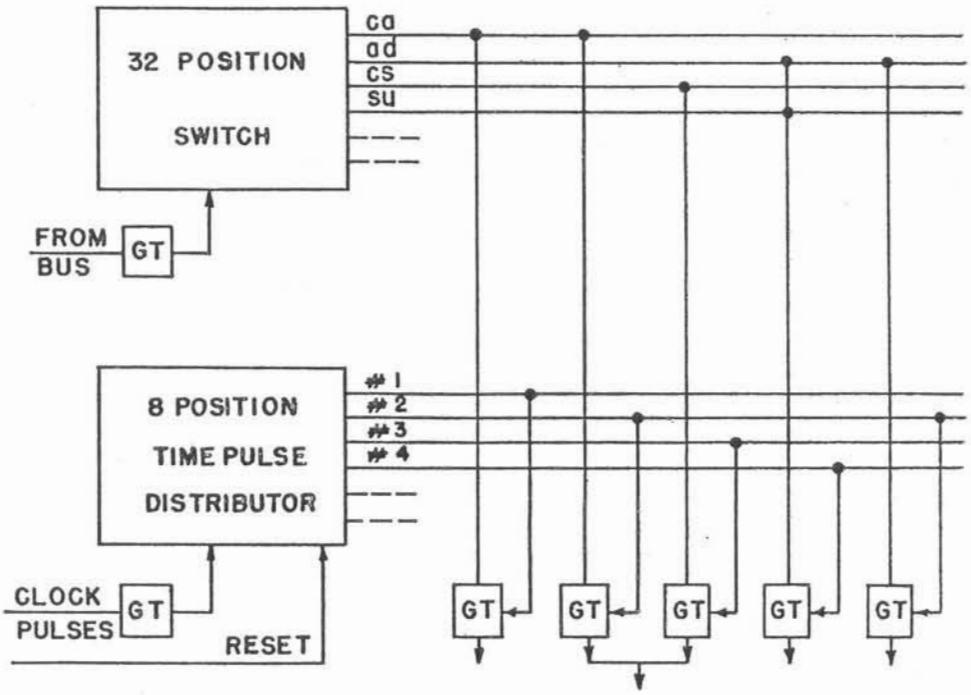


Figure 11

CONTROL

```
 3546
 1371
 ----
   1
 4817
   1
 ----
 4917
```

Figure 12

DECIMAL ADDITION

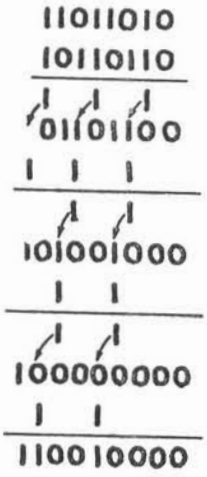


Figure 13
BINARY ADDITION

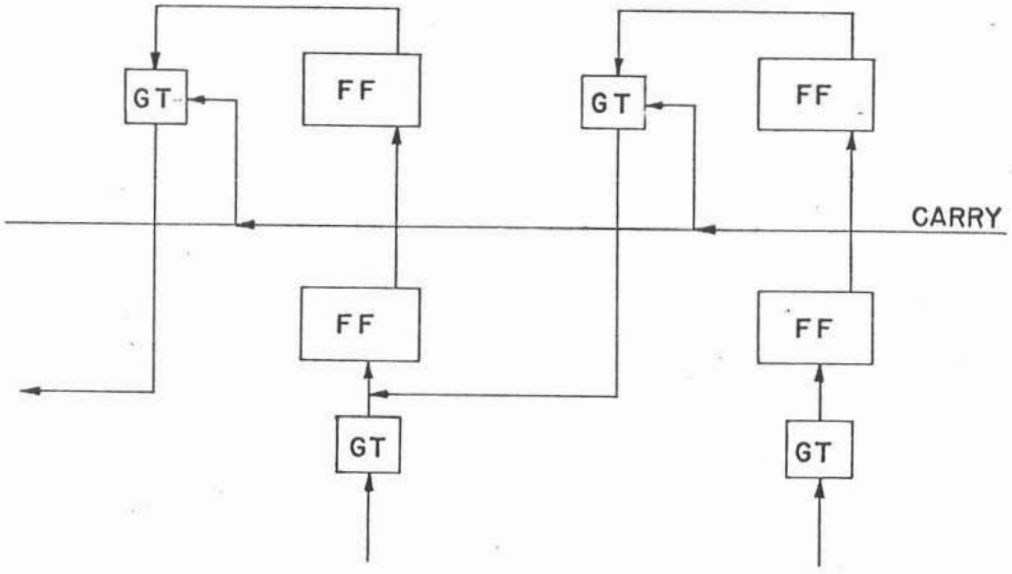


Figure 14
A BINARY ADDER

| | DECIMAL | BINARY |
|---|---|---|
| POSITIVE NUMBER | 223 | 11011111 |
| NEGATIVE " | - 223 | - 11011111 |
| 10'S COMPLEMENT | 1\|777 | 1\|00100001 |
| 9'S " | 1\|776 | 1\|00100000 |

Figure 15

NEGATIVE NUMBERS

| BINARY | | DECIMAL |
|---|---|---|
| +010110110 | $= N_1$ | = 182 |
| -011010010 | $= -N_2$ | = -210 |
| | | -28 |
| 010110110 | $= N_1$ | |
| 100101101 | $= 2^n - N_2 - 1$ | |
| 111100011 | $= 2^n - (N_2 - N_1) - 1$ | |
| -000011100 | $= N_1 - N_2$ | = -28 |
| 010110110 | $= N_1$ | = 182 |
| -010010010 | $= N_2$ | = -146 |
| | | +36 |
| 010110110 | $= N_1$ | |
| 101101101 | $= 2^n - N_2 - 1$ | |
| 1000100011 | $= N_1 - N_2 + 2^n - 1$ | |
| END AROUND CARRY +1 | | |
| 000100100 | $= N_1 - N_2$ | = +36 |

Figure 16

SUBTRACTION USING 9'S COMPLEMENTS

```
    356   MULTIPLICAND
    627   MULTIPLIER
     42
    35
   21
   2492   PARTIAL PRODUCT
    12
   10
   6
   9612   PARTIAL PRODUCT
   36
  30
 18
 223212   PRODUCT
```

Figure 17
DECIMAL MULTIPLICATION

REPRESENTS POWERS OF 2

MULTIPLICATION TABLE:

1 × 1 = 1
1 × 0 = 0
0 × 0 = 0

ADDITION:

1 + 1 = 10
1 + 0 = 1
0 + 0 = 0

BINARY COLUMNS $\approx 3\frac{1}{3}$ × DECIMAL COLUMNS

ONLY DIGITS 1 AND 0 REQUIRED IN EQUIPMENT

Figure 18
BINARY NOTATION

```
  10110   MULTIPLICAND      22
  10011   MULTIPLIER        19
 -----                     ----
  10110                     198
 10110                      22
-------                   ----
 1000010  PARTIAL PRODUCT   418
 00000
-------
 1000010  PARTIAL PRODUCT
00000
-------
 01000010 PARTIAL PRODUCT
10110
--------
110100010  = 418 PRODUCT
```

Figure- 19

BINARY MULTIPLICATION

```
                10110   MULTIPLICAND
STEP 1          10011   MULTIPLIER
                10110   PARTIAL PRODUCT
              ---------
                10110   MULTIPLICAND
                 1001   SHIFTED MULTIPLIER
STEP 2          10110   SHIFTED PARTIAL PRODUCT
               10110
              -------
              1000010   PARTIAL PRODUCT
            ----------
                10110   MULTIPLICAND
                  100   SHIFTED MULTIPLIER
STEP 3          1000010 SHIFTED PARTIAL PRODUCT
               00000
              -------
              1000010   PARTIAL PRODUCT
            ----------
                10110   MULTIPLICAND
                   10   SHIFTED MULTIPLIER
STEP 4           1000010 SHIFTED PARTIAL PRODUCT
               00000
              -------
              01000010  PARTIAL PRODUCT
            ----------
                10110   MULTIPLICAND
                    1   SHIFTED MULTIPLIER
STEP 5          01000010 SHIFTED PARTIAL
                         PRODUCT
               10110
              -------
              110100010 PRODUCT
```

Figure 20

MODIFIED BINARY MULTIPLICATION

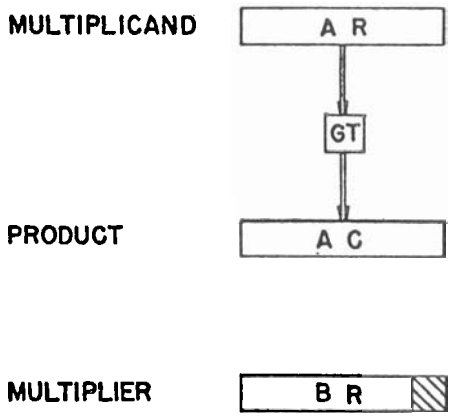


Figure 21
MULTIPLICATION I

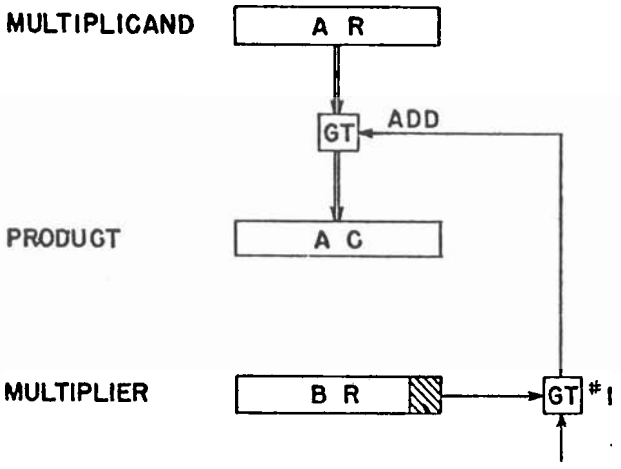


Figure 22
MULTIPLICATION II

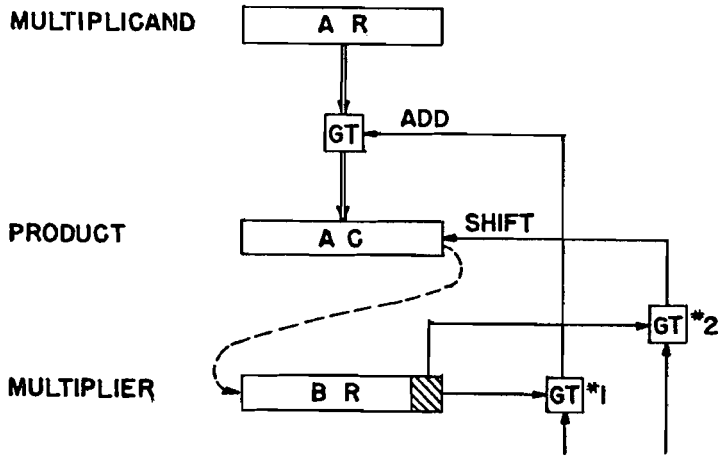


Figure 23
MULTIPLICATION III

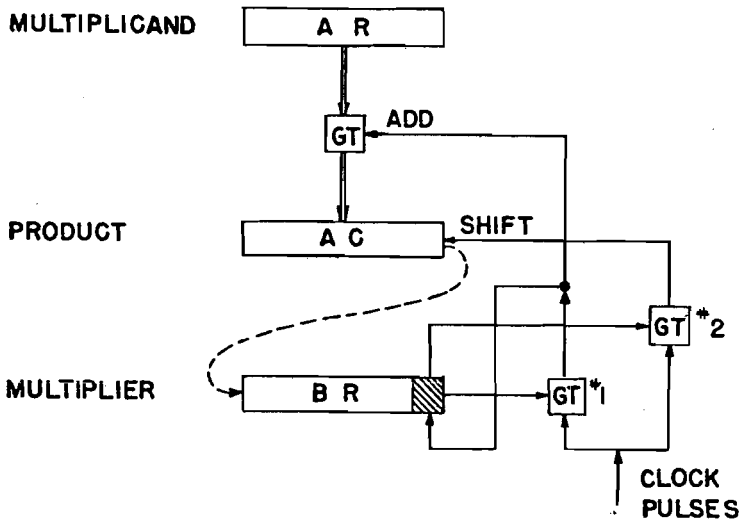


Figure 24
MULTIPLICATION IV

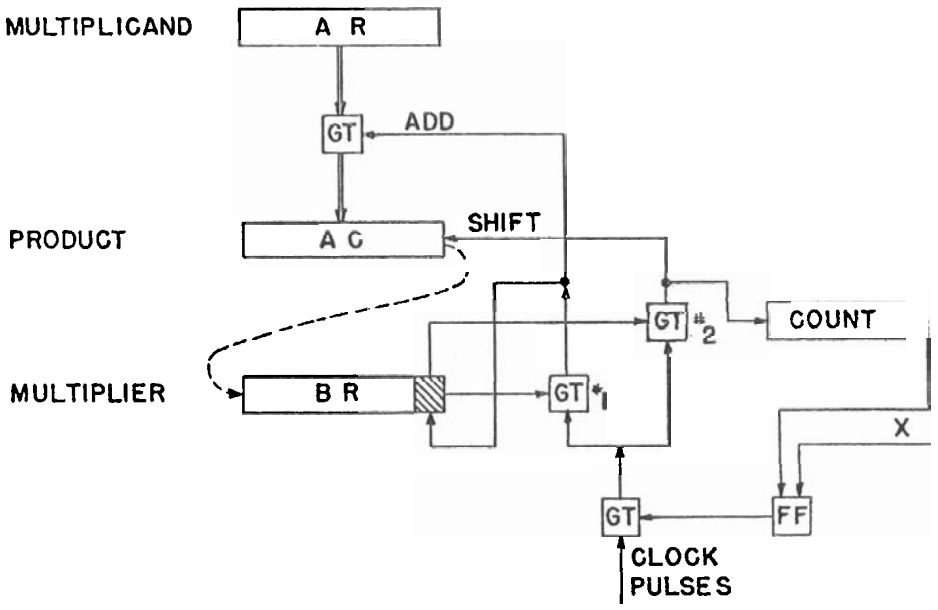


Figure 25
MULTIPLICATION V

DECIMAL

| | |
|---|---|
| 1734 | 6148 |
| | 5 |
| 1735 | 1 |

| | |
|---|---|
| 173 | 46148 |
| | 5 |
| 173 | 9 |

BINARY

| | |
|---|---|
| 11010 | 1010 |
| | 1 |
| 11011 | |

| | |
|---|---|
| 1101 | 01010 |
| | 1 |
| 1101 | 1 |

Figure 26
ROUNDING OFF

| | | |
|---|---|---|
| | 1 4 2 | |
| 28 | 3 9 7 6 | |
| | −2 8 | SUBTRACT 1 |
| | 1 1 7 6 | POS. REMAINDER |
| | −2 8 | SUBTRACT 2 |
| | −1 6 2 4 | OVERCAST |
| | +2 8 | RESTORE −1 |
| | 1 1 7 6 | SHIFT 1 NET SUBTRACTION |
| | − 2 8 | SUBTRACT 1 |
| | 8 9 6 | POS. REMAINDER |
| | − 2 8 | SUBTRACT 2 |
| | 6 1 6 | POS. REM. |
| | −2 8 | SUBTRACT 3 |
| | 3 3 6 | POS. REM. |
| | −2 8 | SUBTRACT 4 |
| | 5 6 | POS. REM. |
| | − 2 8 | SUBTRACT 5 |
| | −2 2 4 | OVERCAST |
| | +2 8 | RESTORE −1 |
| | 5 6 | SHIFT 4 NET SUBTRACTION |
| | −2 8 | SUBTRACT 1 |
| | 2 8 | POS. REM. |
| | −2 8 | SUBTRACT 2 |
| | 0 | POS. REM. |
| | −2 8 | SUBTRACT 3 |
| | −2 8 | OVERCAST |
| | +2 8 | RESTORE −1 |
| | 0 | REMAINDER 0 2 NET SUB. |

Figure 27

DECIMAL DIVISION

| | | |
|---|---|---|
| | 0.1 0 1 1 0 | |
| 0.101 | 0.0 1 1 1 0 | |
| | 1.0 1 0 1 1 | SUBTRACT |
| | 1.1 1 0 0 1 | NEG. REMAINDER 0 IN QUOTIENT |
| | 1.1 0 0 1 1 | SHIFT LEFT |
| | 0.1 0 1 0 0 | ADD |
| | 1 0.0 0 1 1 1 | |
| | ↳ 1 | END AROUND CARRY |
| | 0.0 1 0 0 0 | POS. REMAINDER 1 IN QUOTIENT |
| | 0.1 0 0 0 0 | SHIFT LEFT |
| | 1.0 1 0 1 1 | SUBTRACT |
| | 1.1 1 0 1 1 | NEG. REMAINDER 0 IN QUOTIENT |
| | 1.1 0 1 1 1 | SHIFT LEFT |
| | 0.1 0 1 0 0 | ADD |
| | 1 0.0 1 0 1 1 | |
| | ↳ 1 | END AROUND CARRY |
| | 0.0 1 1 0 0 | POS. REMAINDER 1 IN QUOTIENT |
| | 0.1 1 0 0 0 | SHIFT LEFT |
| | 1.0 1 0 1 1 | SUBTRACT |
| | 1 0.0 0 0 1 1 | |
| | ↳ 1 | END AROUND CARRY |
| | 0.0 0 1 0 0 | POS. REMAINDER 1 IN QUOTIENT |
| | 0.0 1 0 0 0 | SHIFT LEFT |
| | 1.0 1 0 1 1 | SUBTRACT |
| | 1.1 0 0 1 1 | NEG. REMAINDER 0 IN QUOTIENT |

Figure 28

BINARY DIVISION

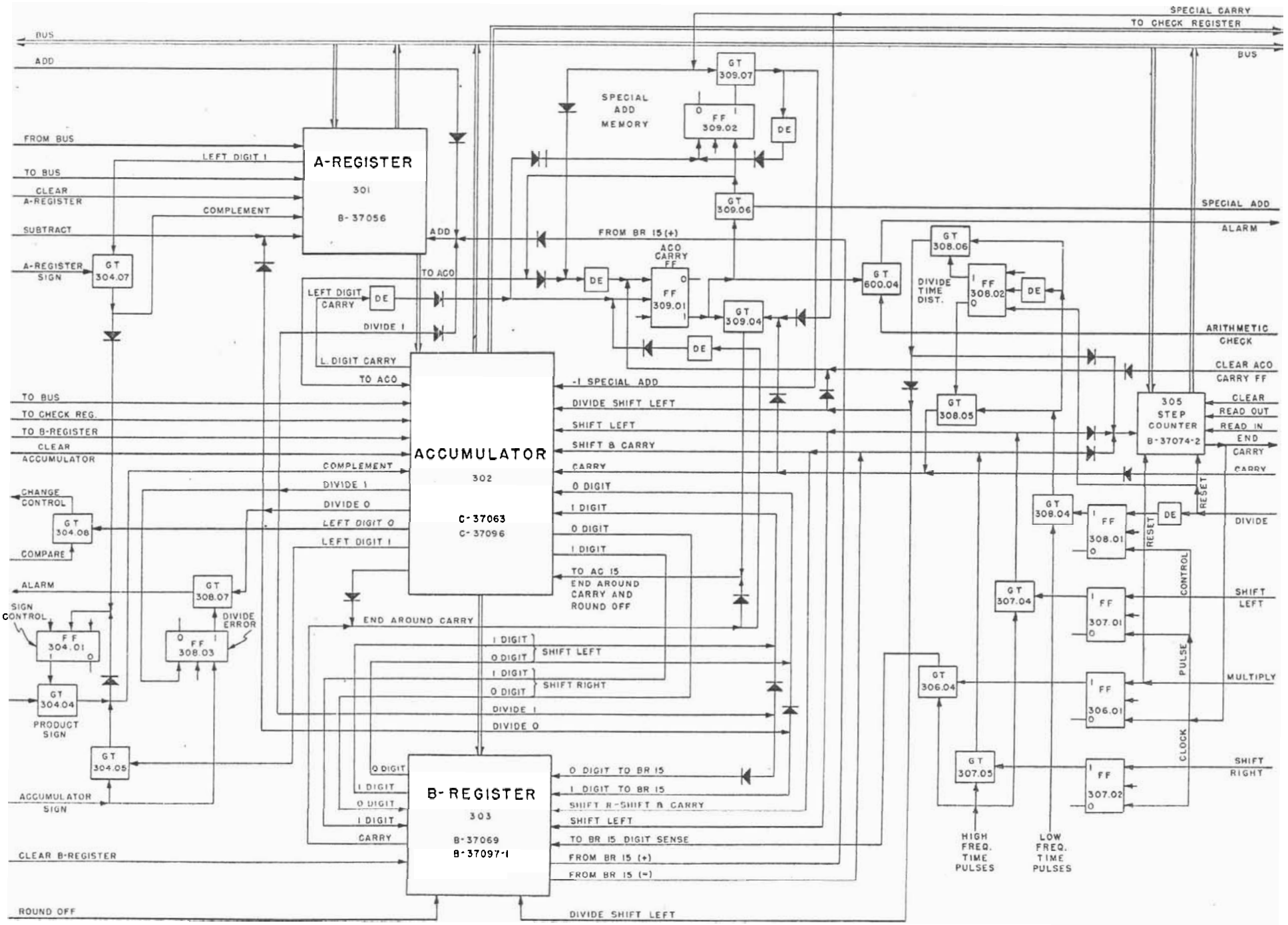


Figure 29

ARITHMETIC ELEMENT

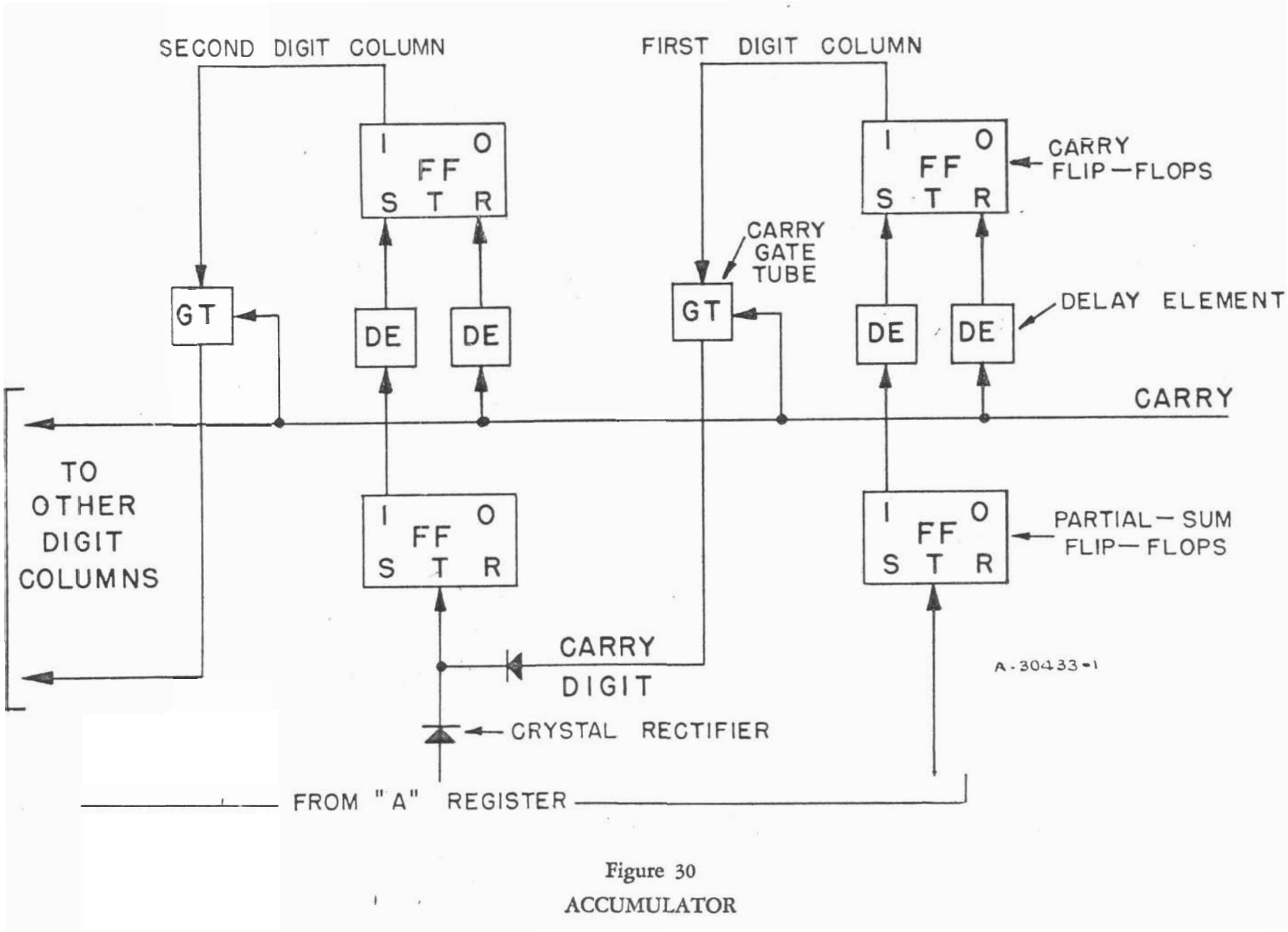


Figure 30
ACCUMULATOR

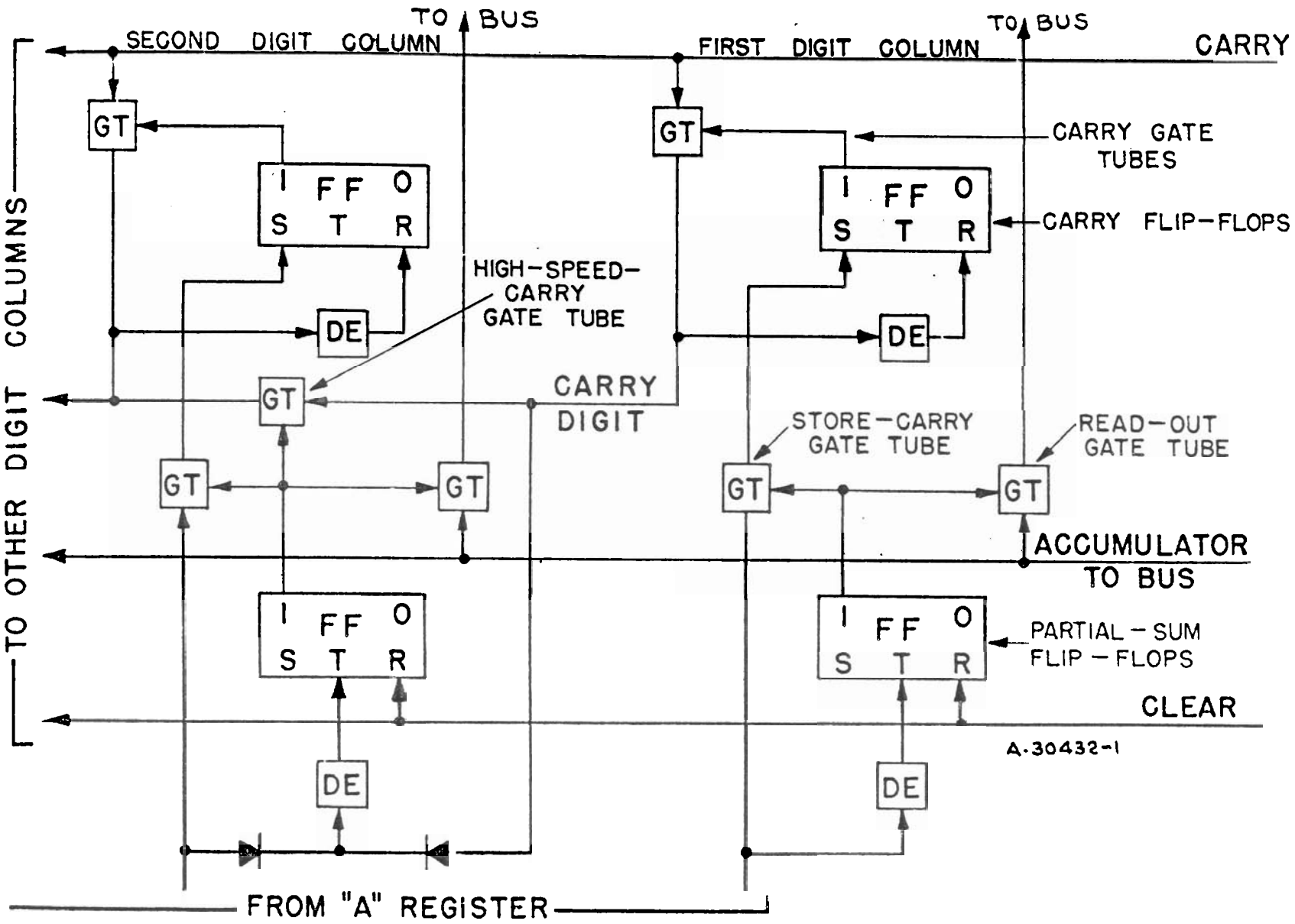


Figure 31
ACCUMULATOR

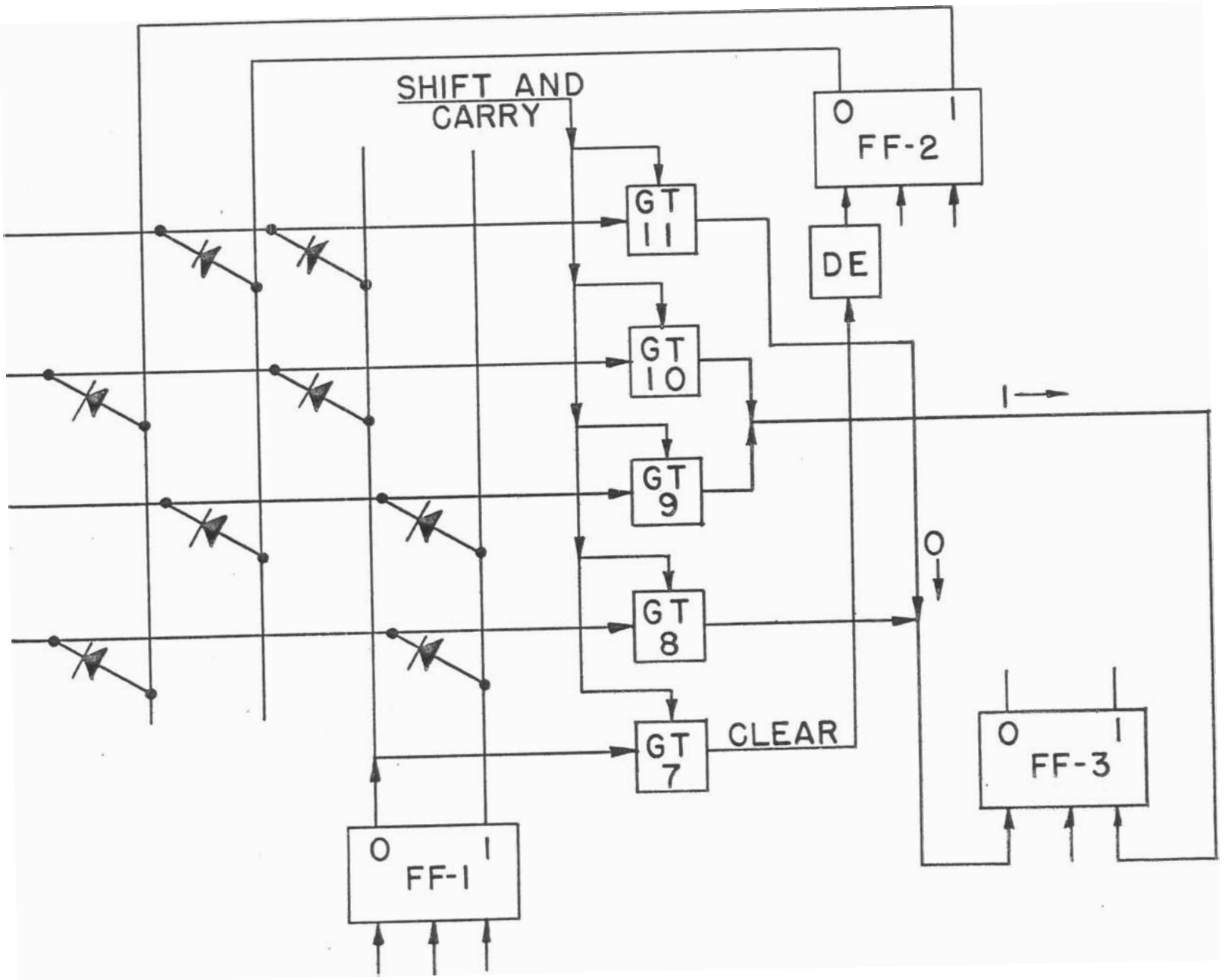


Figure 32
SHIFT AND CARRY

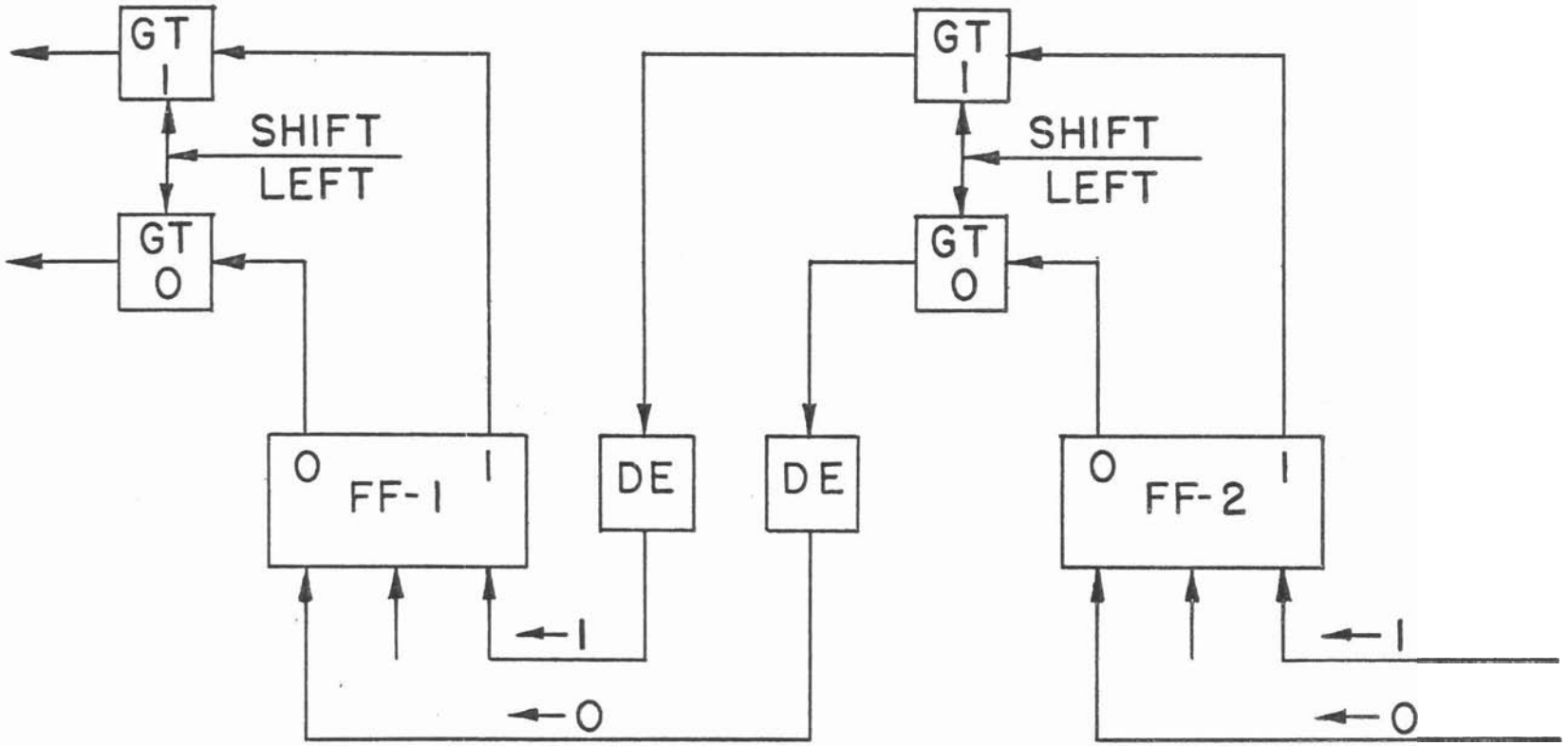


Figure 33
SHIFT LEFT

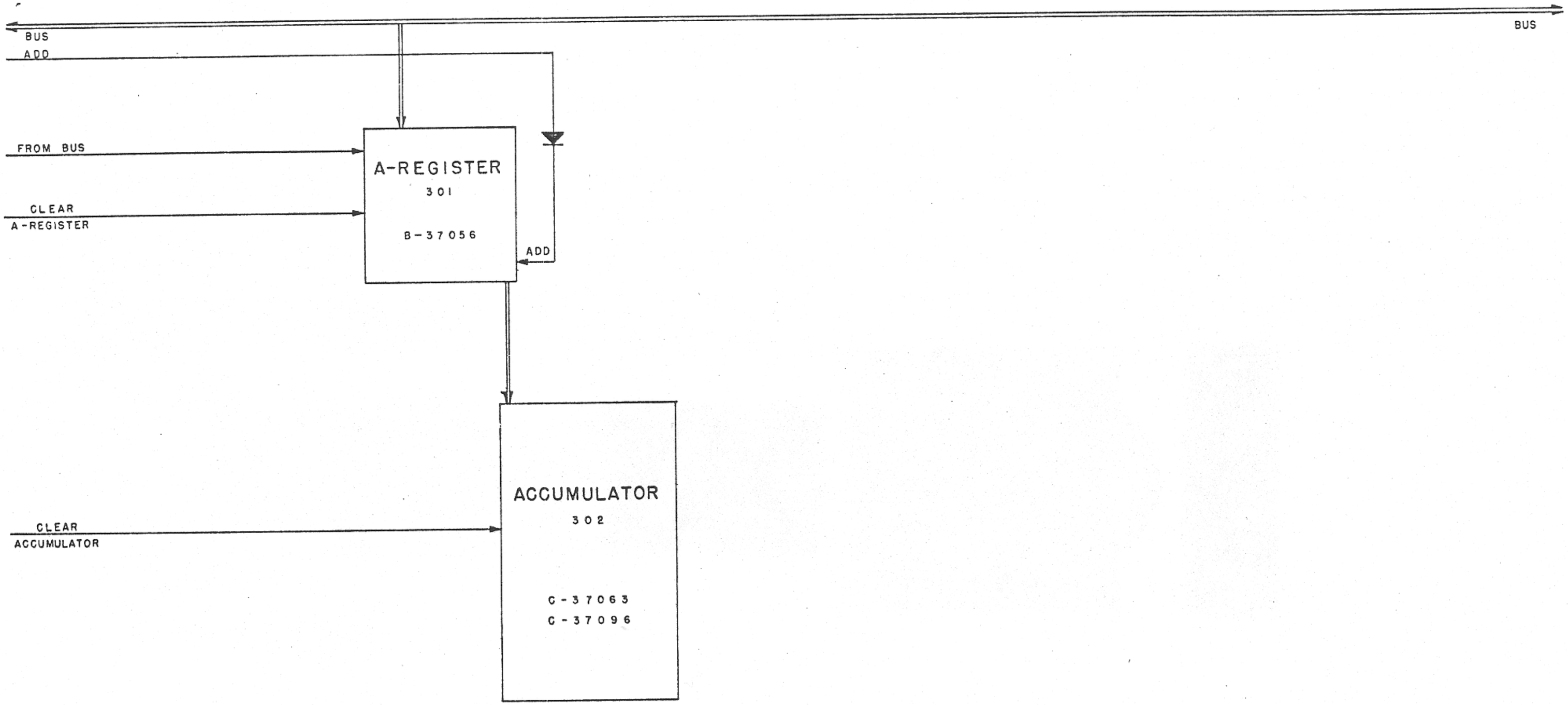


Figure 34
CLEAR AND ADD

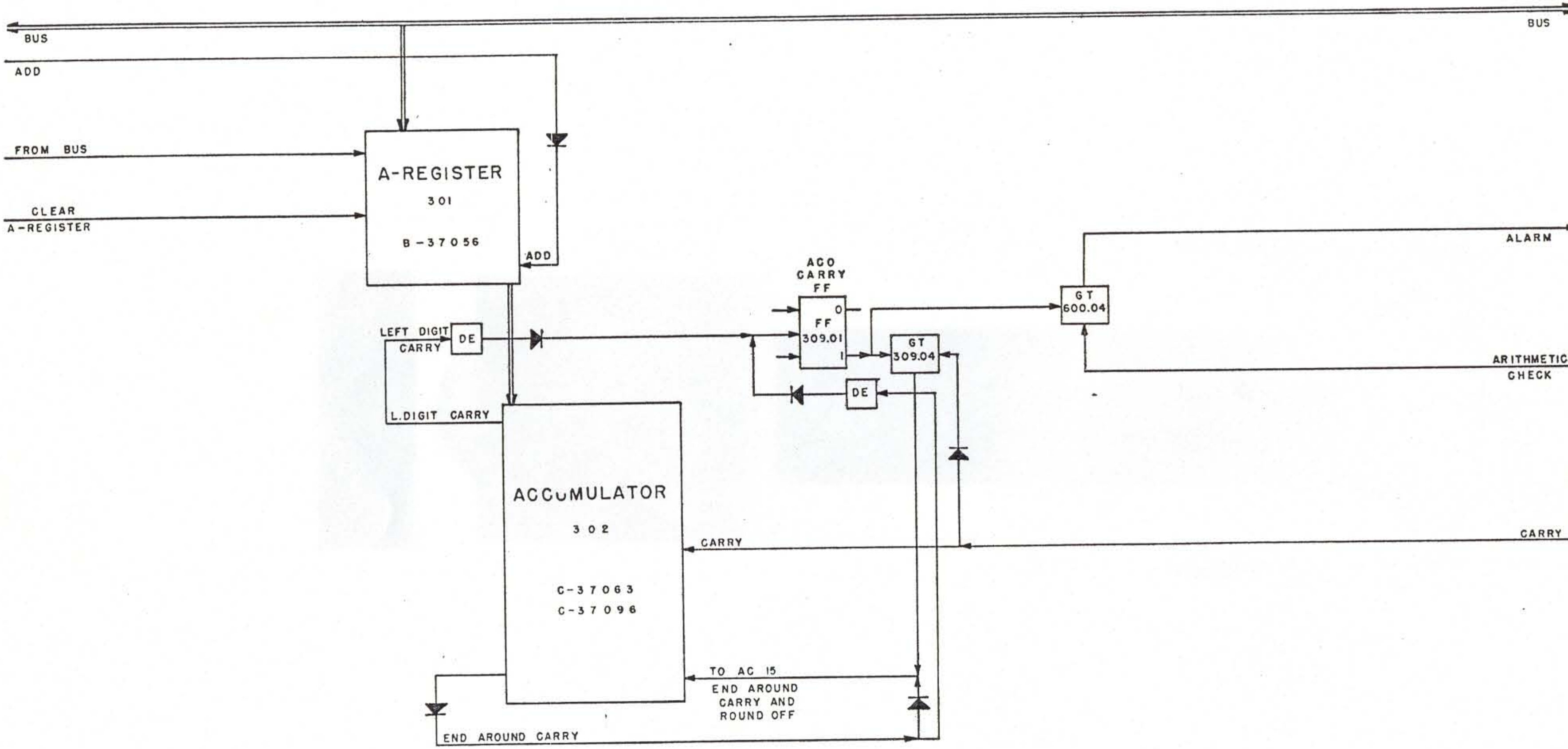


Figure 35

ADD

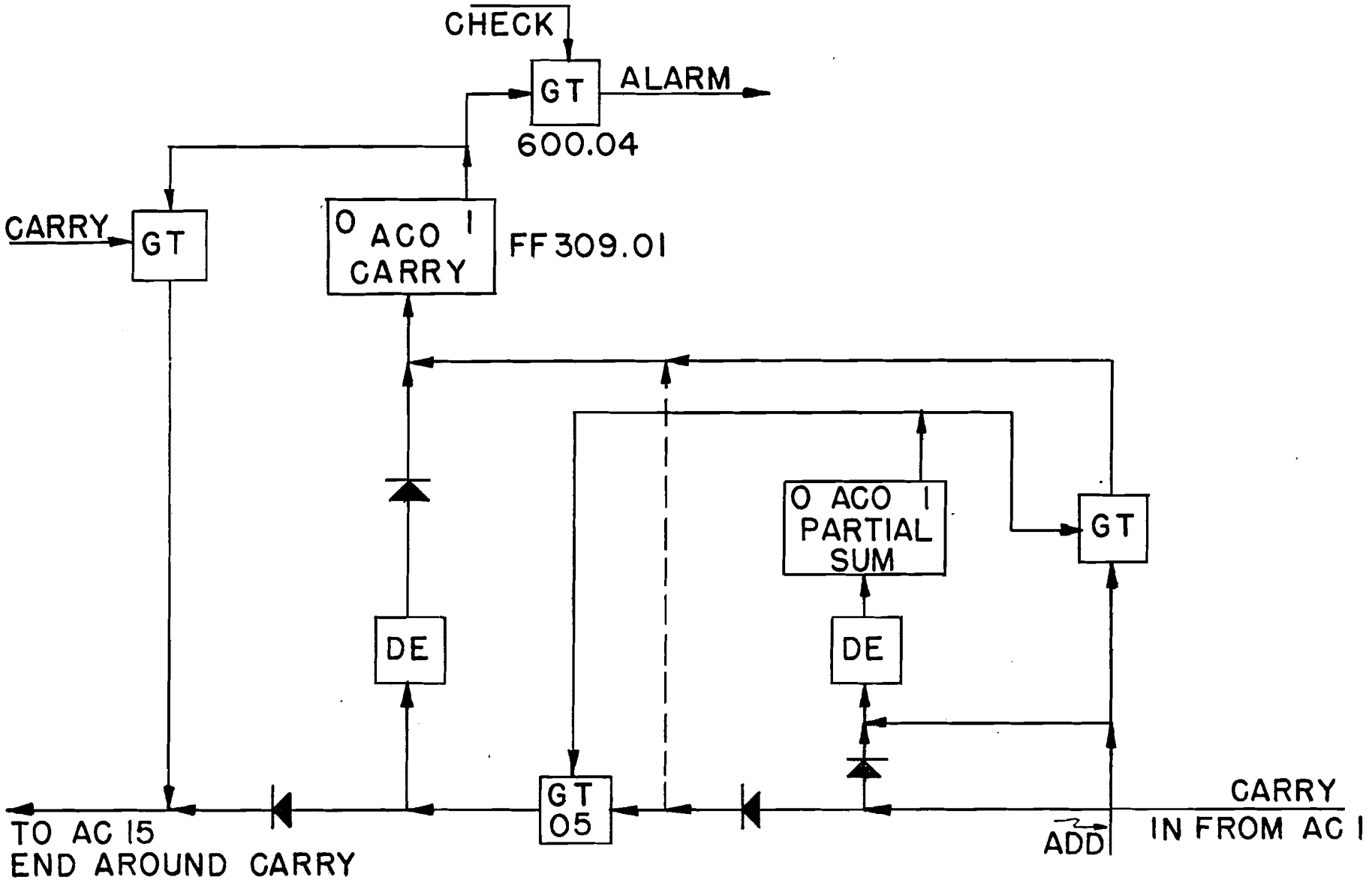


Figure 36
ARITHMETIC CHECK

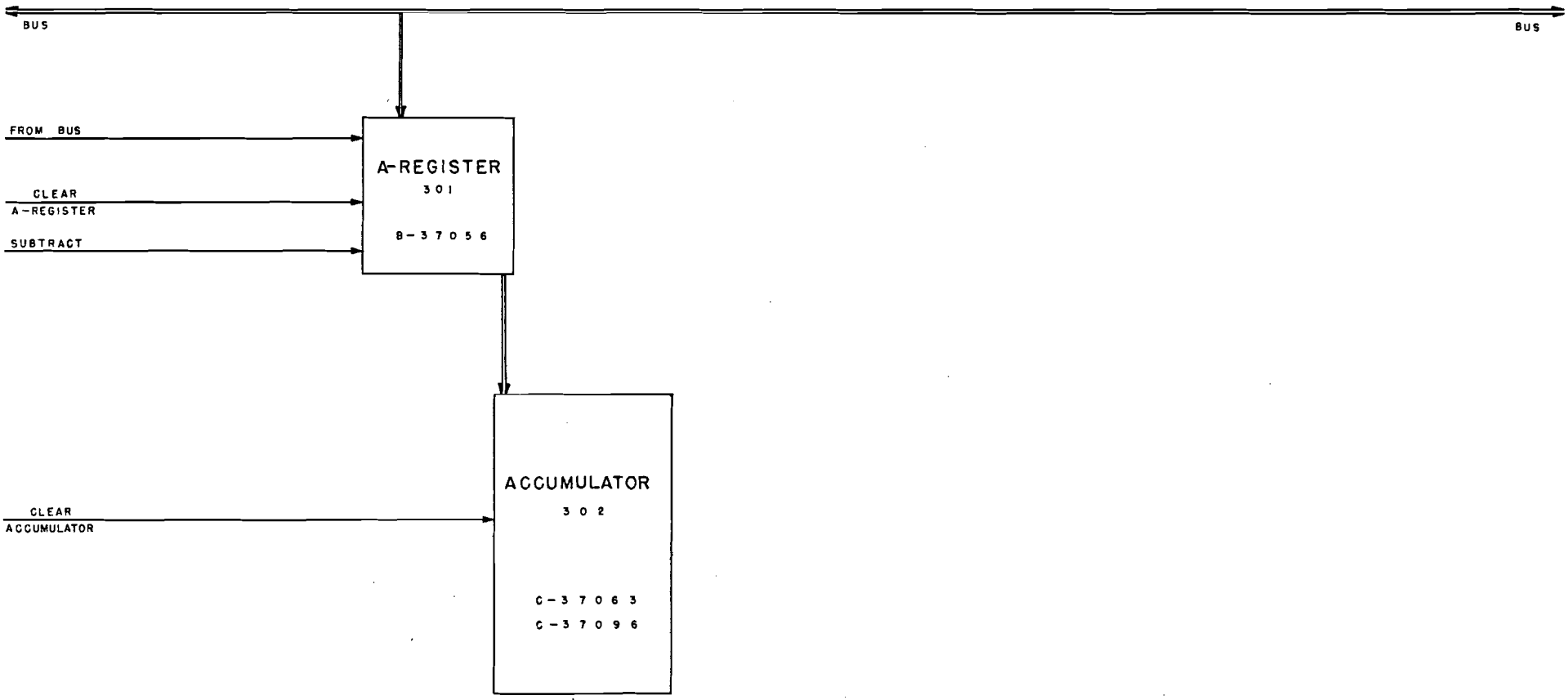


Figure 37
CLEAR AND SUBTRACT

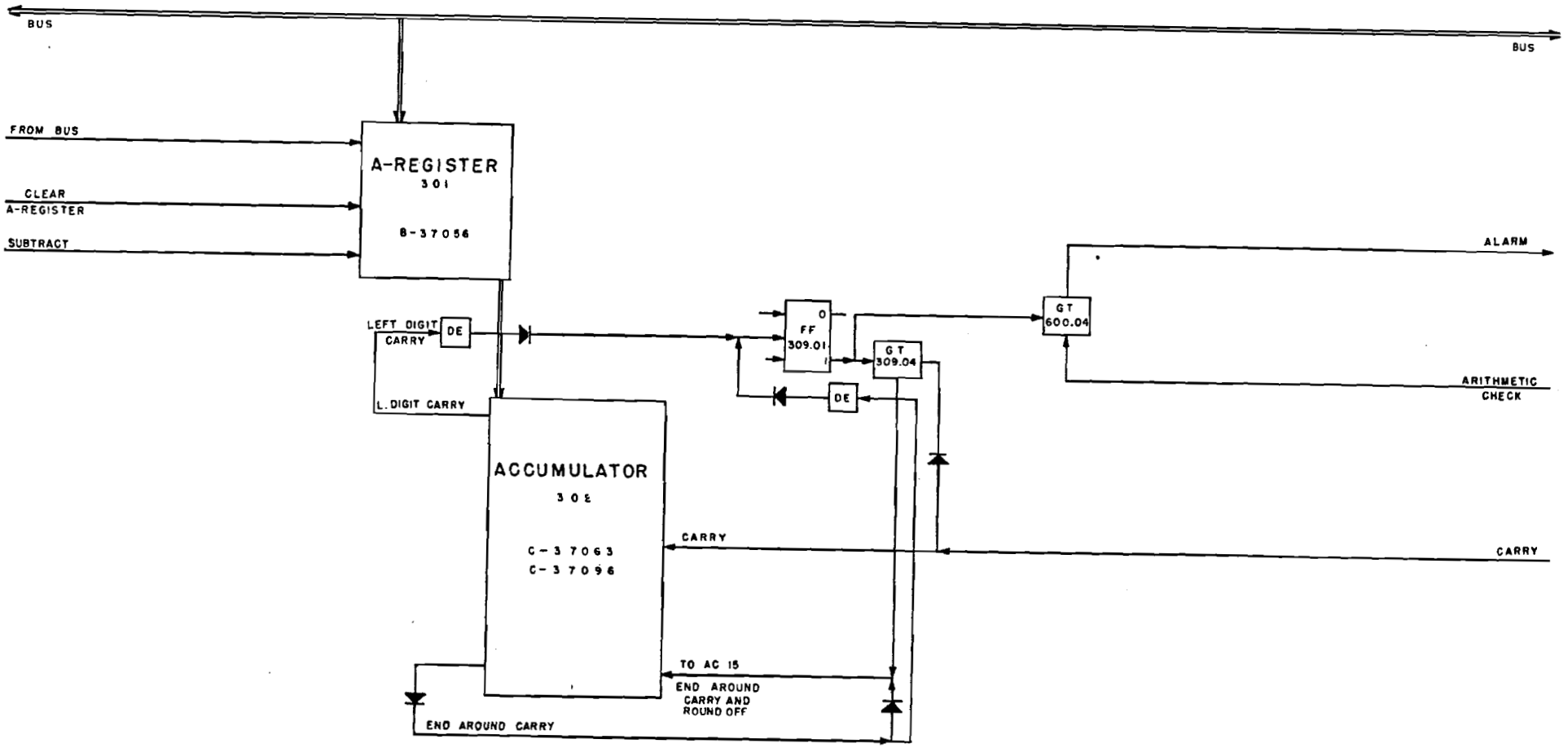


**Figure 38**
**SUBTRACT**

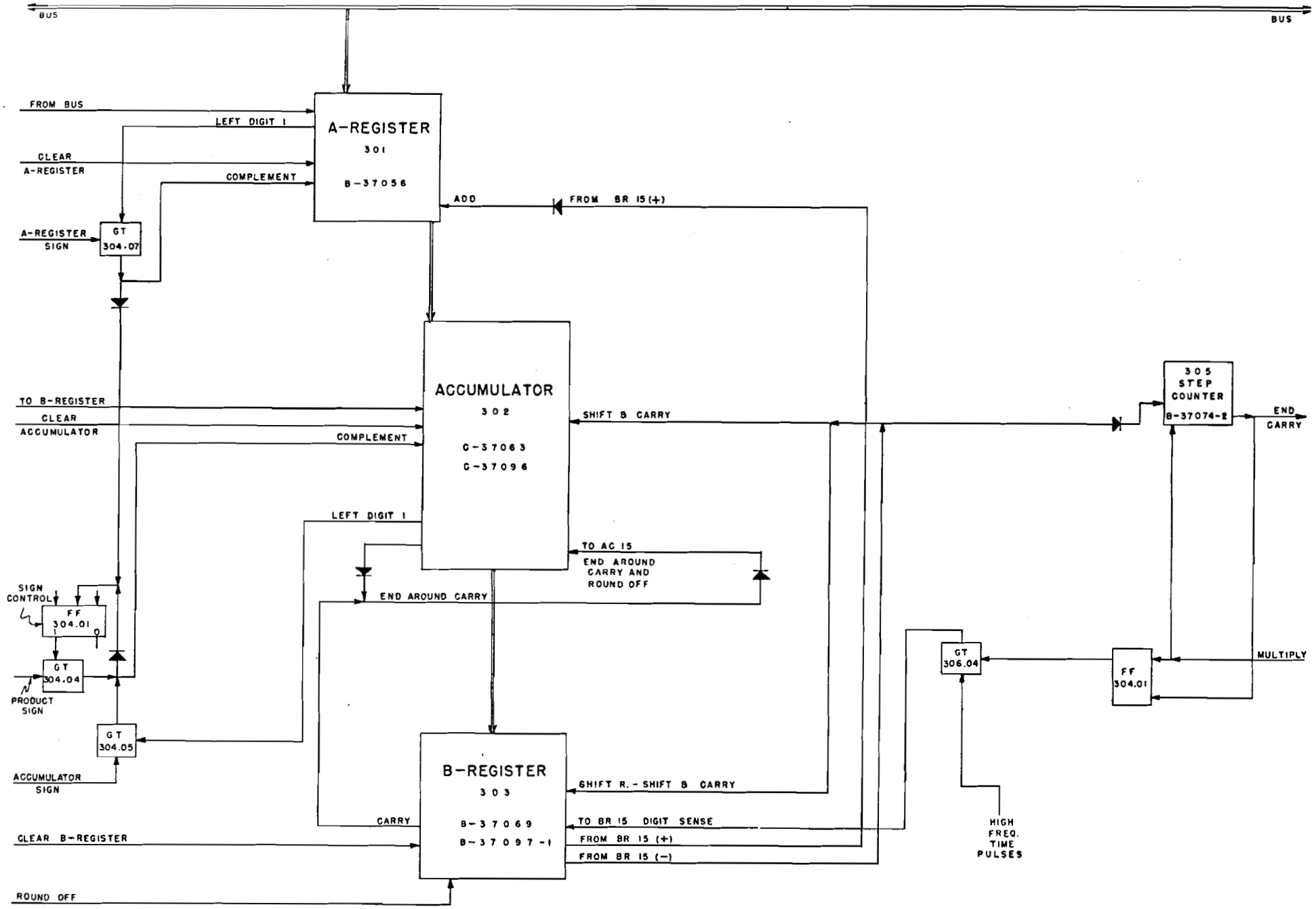


Figure 39

MULTIPLY AND ROUNDOFF

MULTIPLY AND HOLD FULL PRODUCT

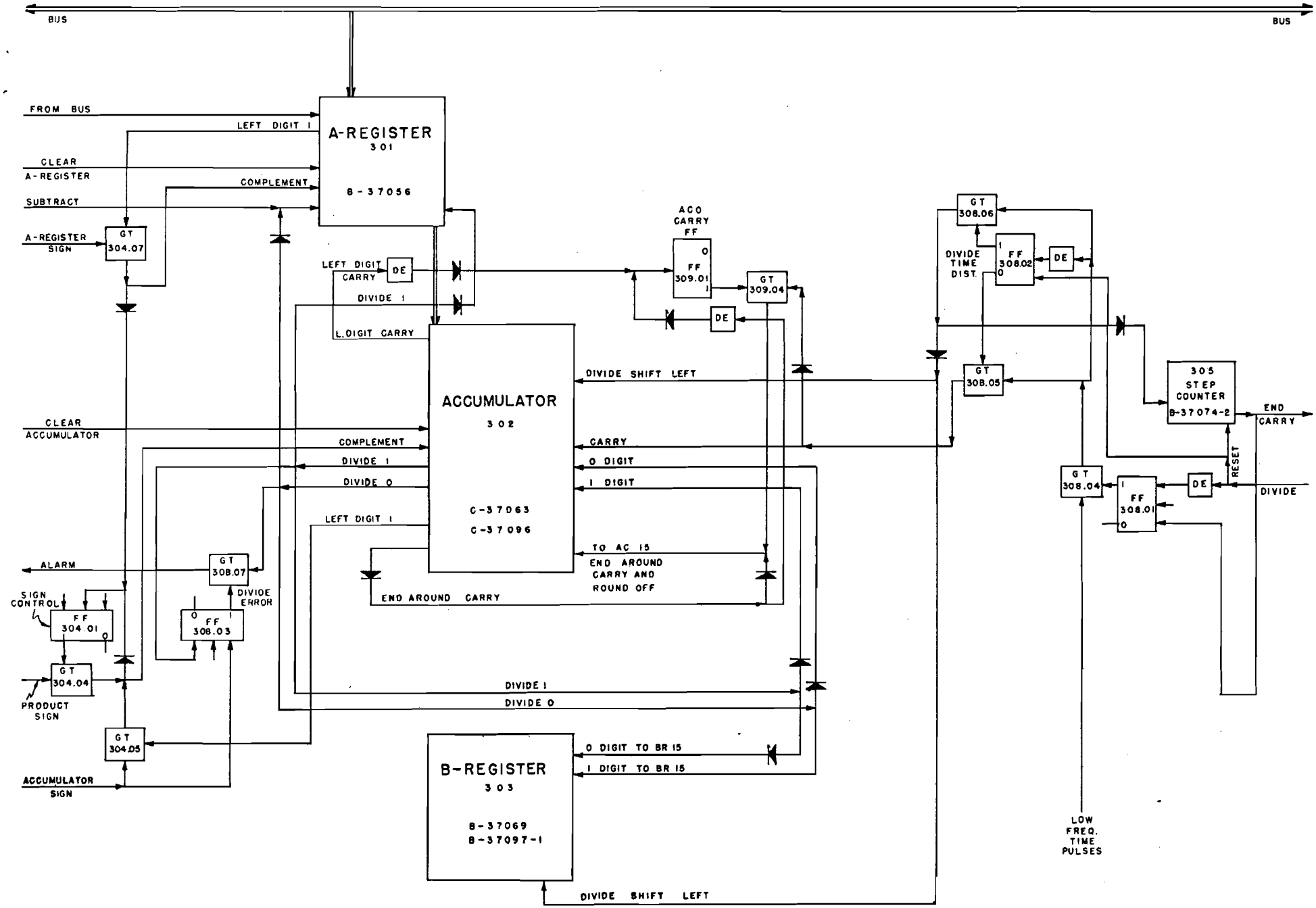


Figure 40
DIVIDE

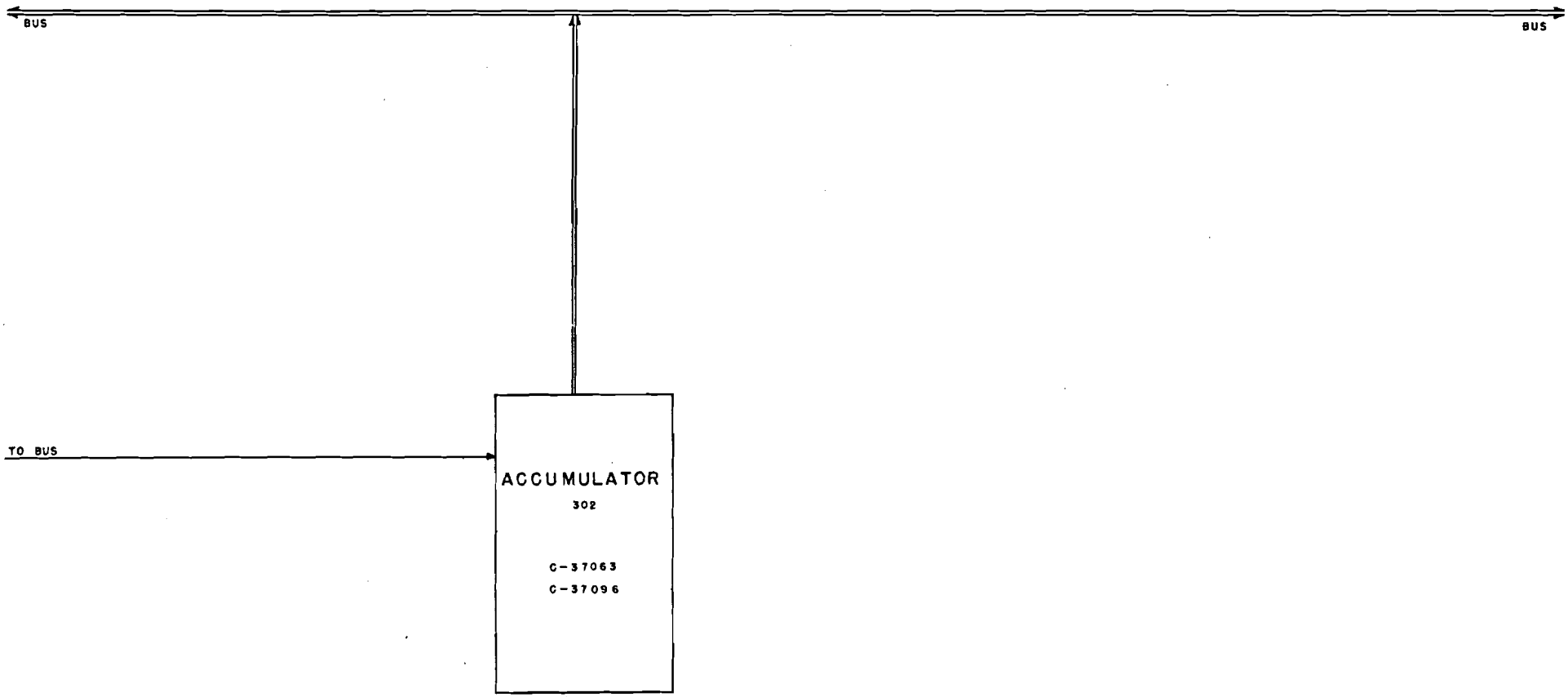


Figure 41

TRANSFER TO STORAGE

TRANSFER DIGITS

STORE AND DISPLAY

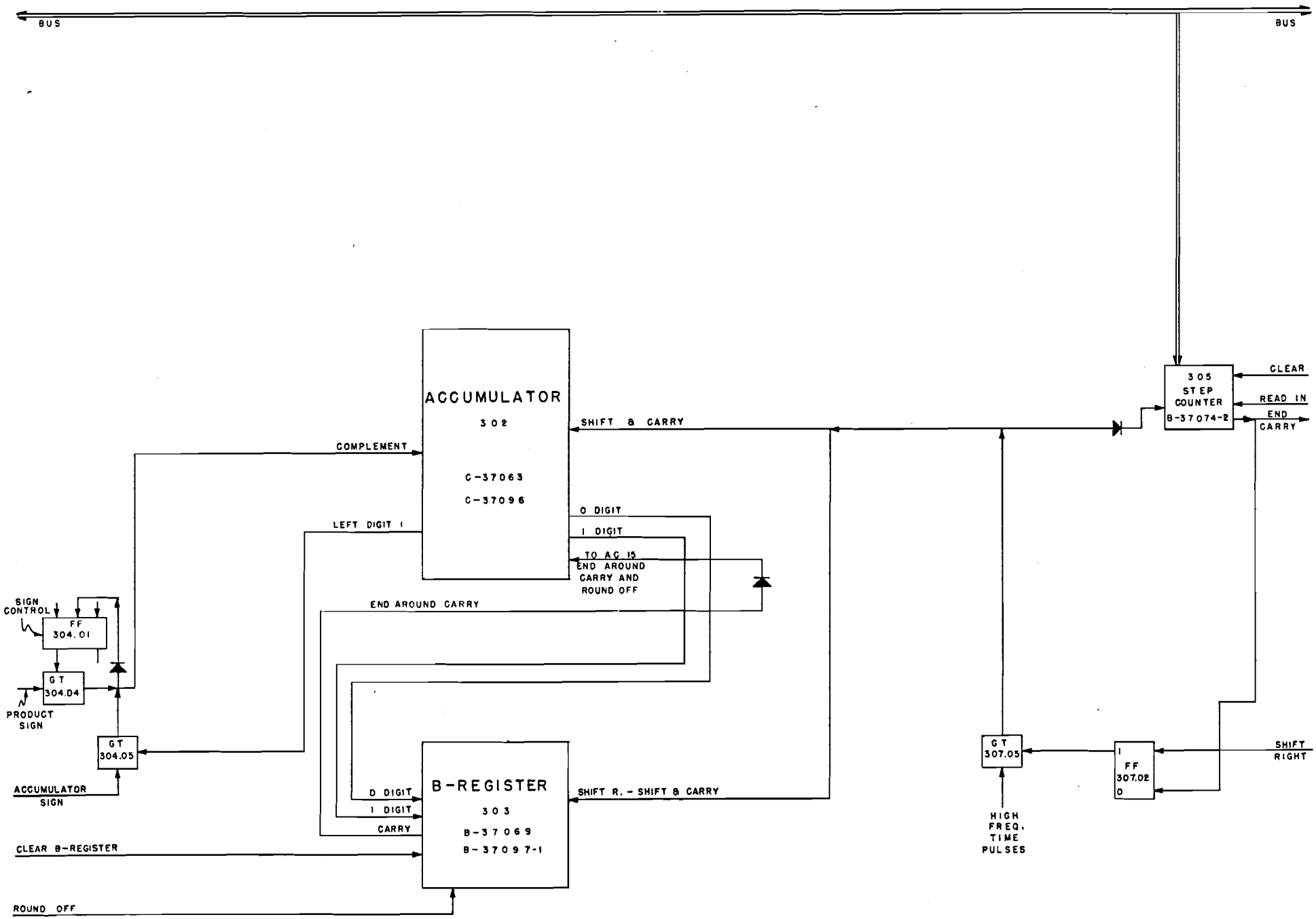


**Figure 42**
**SHIFT RIGHT**

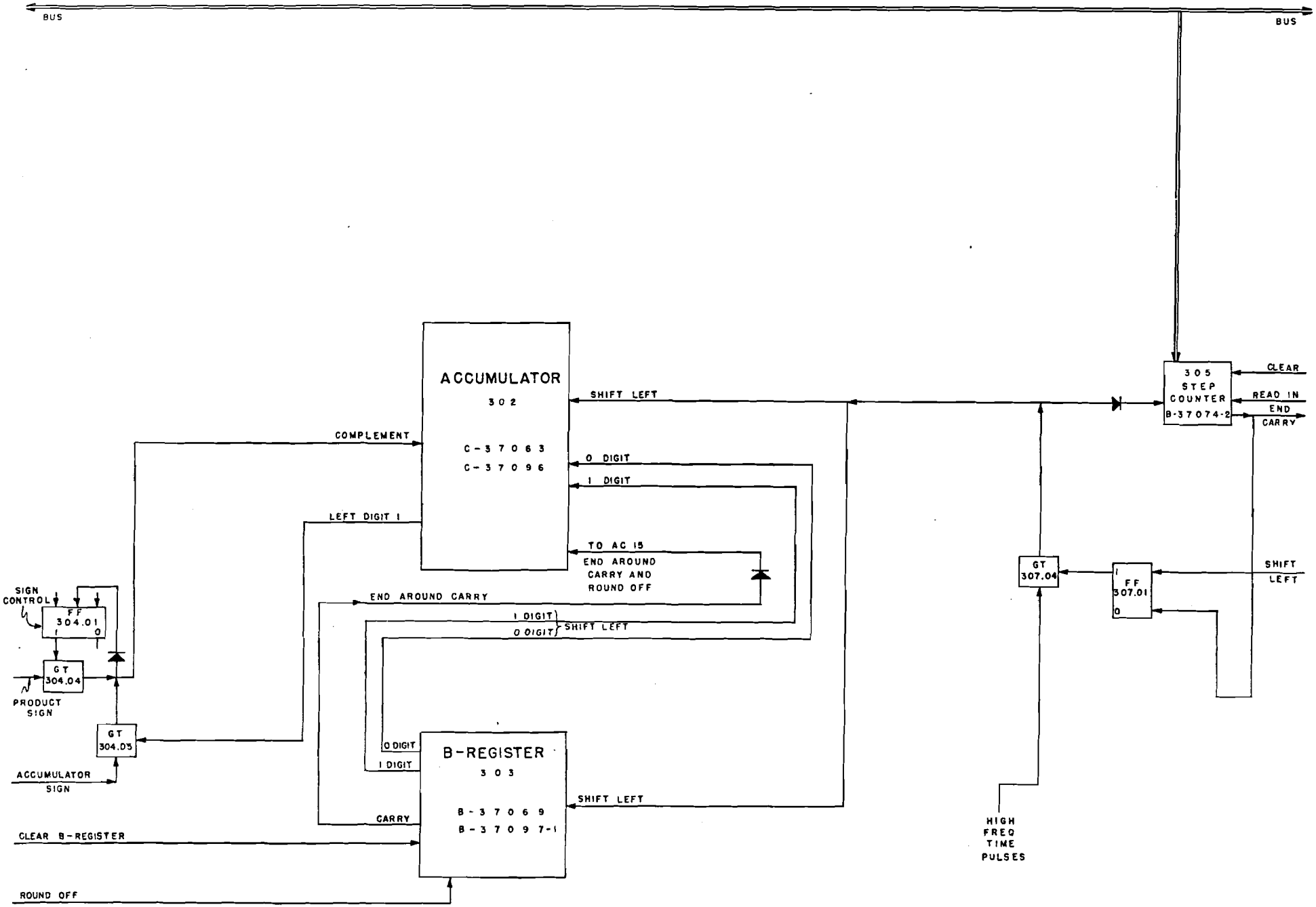


Figure 43

SHIFT LEFT

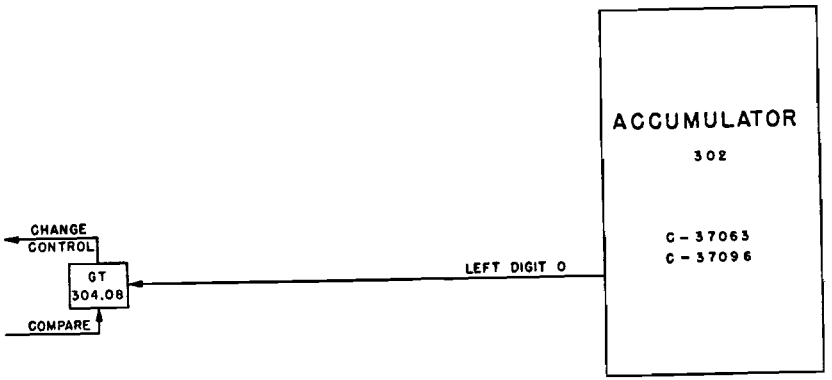


Figure 44
CONDITIONAL PROGRAM

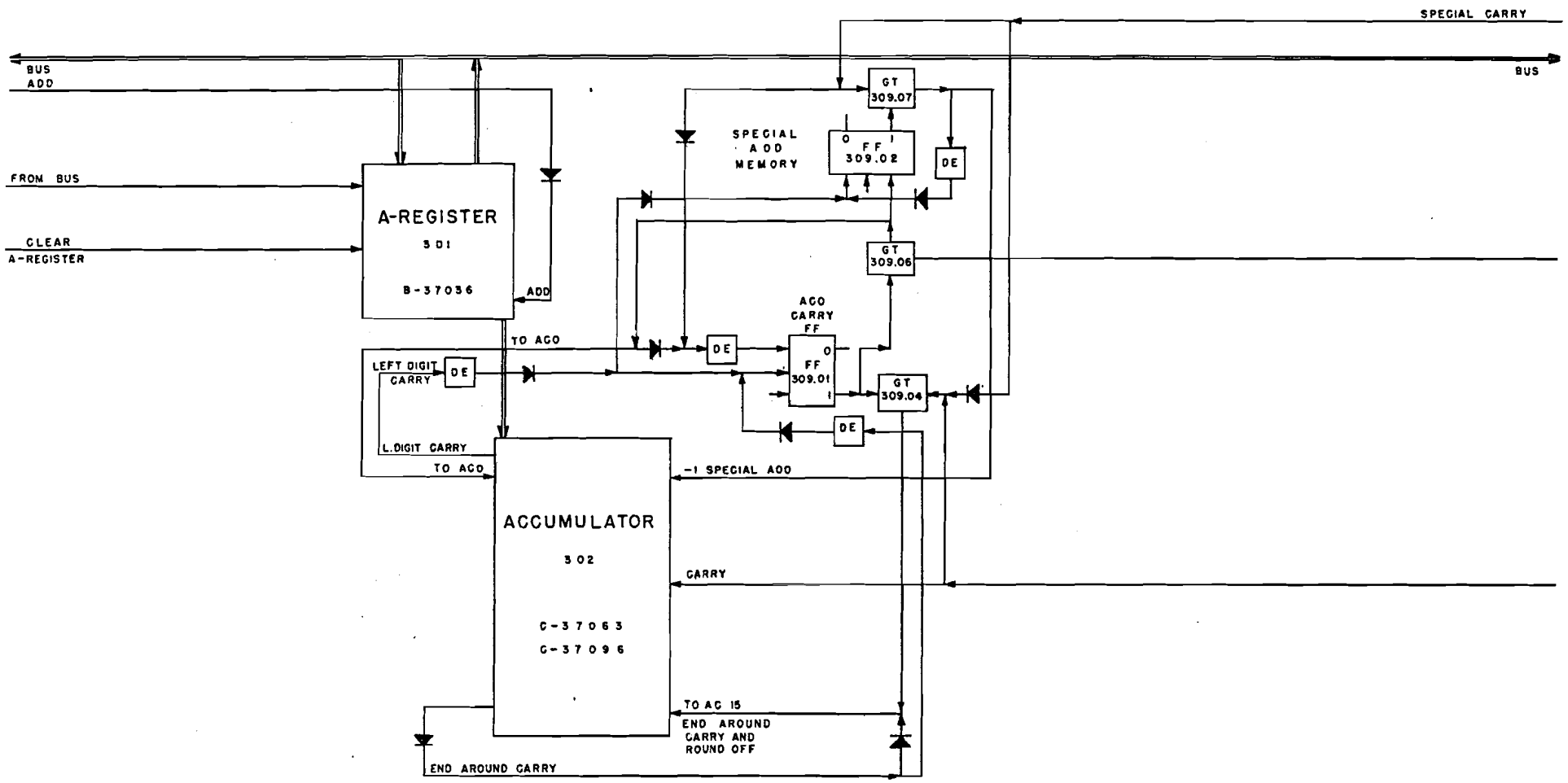


Figure 45

SPECIAL ADD

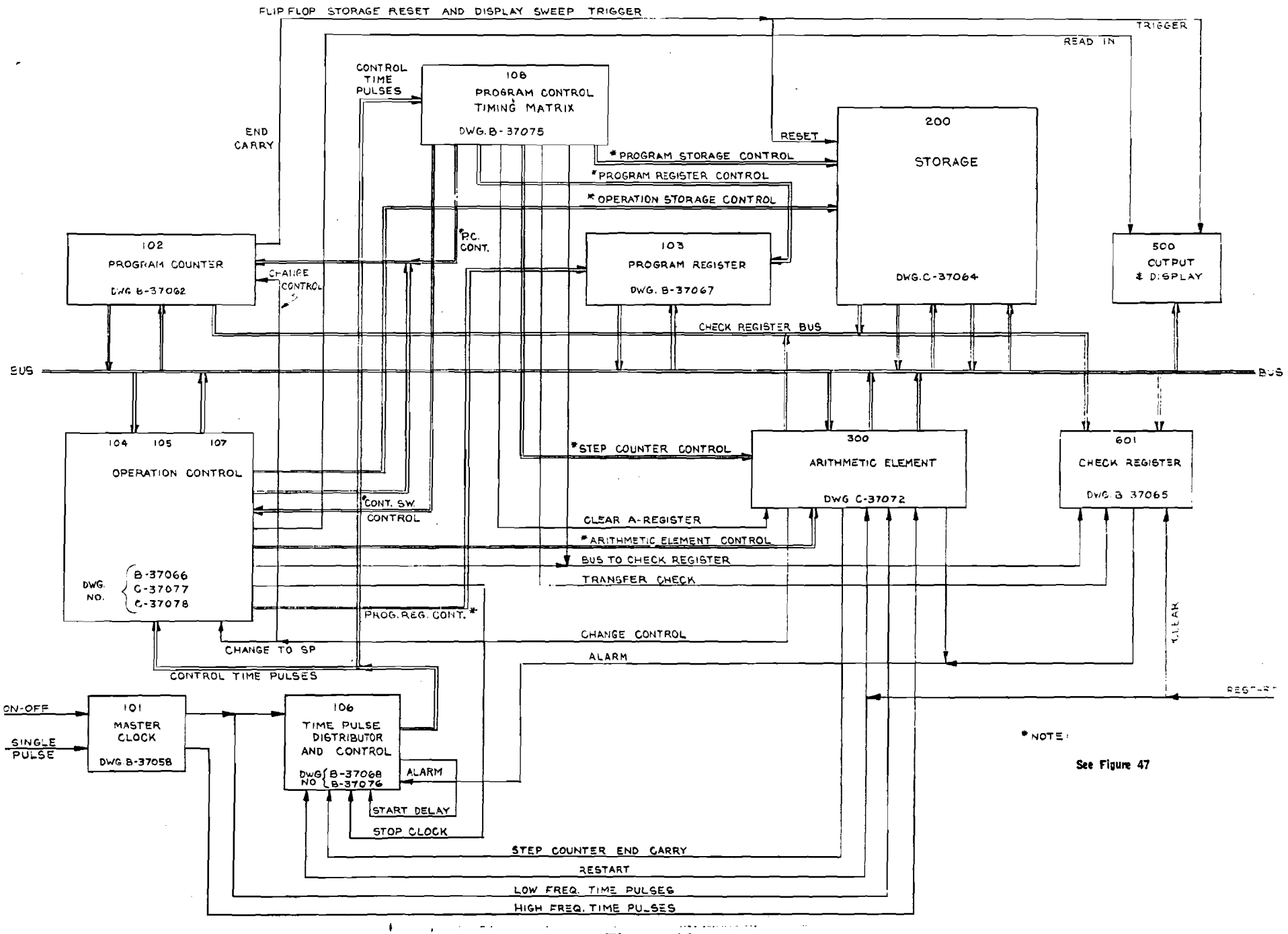


**Figure 46**

**SYSTEM BLOCK DIAGRAM**

**ARITHMETIC ELEMENT CONTROL**

| | | |
|---|---|---|
| 1. | Bus to A-Register | GT 301.01 |
| 2. | A-Register to Bus | GT 301.02 |
| 3. | Subtract: A-Register to Accum. | GT 301.04 |
| 4. | Add: A-Register to Accumulator | GT 301.05 |
| 5. | Accumulator to Bus | GT 302.02 |
| 6. | Accumulator to Check Register | GT 302.03 |
| 7. | Accumulator to B-Register | GT 302.04 |
| 8. | Carry | GT 302.20 |
| 9. | Roundoff | GT 303.08 |
| 10. | Product Sign | GT 304.04 |
| 11. | Accumulator Sign | GT 304.05 |
| 12. | A-Register Sign | GT 304.07 |
| 13. | Compare | GT 304.08 |
| 14. | Multiply | FF 306.01 |
| 15. | Shift Left | FF 307.01 |
| 16. | Shift Right | FF 307.02 |
| 17. | Divide | FF 308.01 |
| 18. | Special Add | GT 309.06 |
| 19. | Arithmetic Check | GT 600.04 |
| 20. | Clear Accumulator | |
| 21. | Clear B-Register | |
| 22. | Special Carry | |

**PROGRAM COUNTER CONTROL**

| | | |
|---|---|---|
| 1. | In from Bus | GT 102.01 |
| 2. | Out to Bus | GT 102.02 |
| 3. | Out to Check Register | GT 102.03 |
| 4. | Add Pulse | |
| 5. | Clear | |

**PROGRAM REGISTER CONTROL**

| | | |
|---|---|---|
| 1. | In from Bus | GT 103.01 |
| 2. | Out to Bus | GT 103.02 |
| 3. | Clear | |

**CONTROL SWITCH CONTROL**

| | | |
|---|---|---|
| 1. | In from Bus | GT 104.01 |
| 2. | Out to Bus | GT 104.02 |
| 3. | Clear | |

**STEP COUNTER CONTROL**

| | | |
|---|---|---|
| 1. | In from Bus | GT 305.01 |
| 2. | Out to Bus | GT 305.02 |
| 3. | Clear | |

**PROGRAM STORAGE CONTROL**

| | | |
|---|---|---|
| 1. | Bus to Storage Switch | GT 201.01 |
| 2. | Storage Switch to Bus | GT 201.02 |
| 3. | Storage Readout | GT 203.02 |
| 4. | Storage Switch Clear | |
| 5. | Storage to Check Register | GT 203.03 |

**OPERATION STORAGE CONTROL**

| | | |
|---|---|---|
| 1. | In from Bus | GT 203.01 |
| 2. | Storage Readout | GT 203.02 |
| 3. | Out to Check Register | GT 203.03 |
| 4. | Storage Clear | |

NOTE: See Figure 46

Figure 47

CONTROL FUNCTIONS

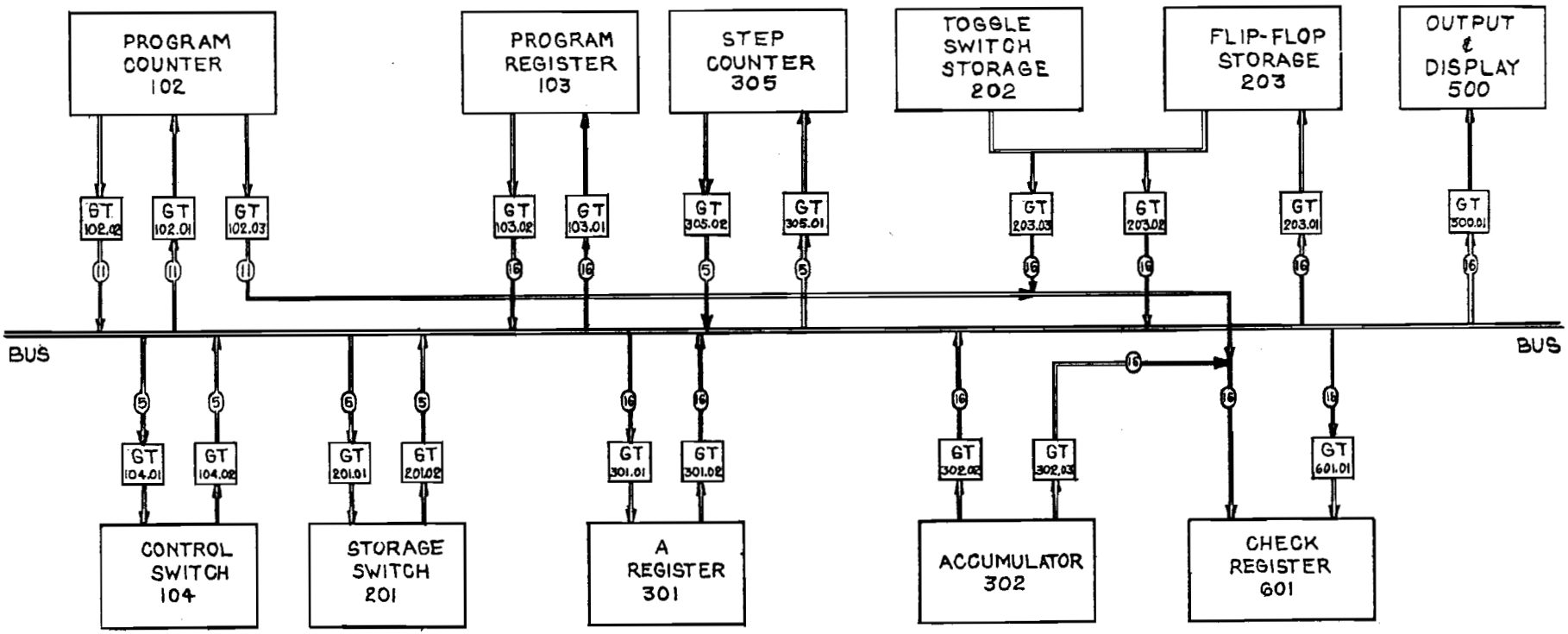


Figure 48

BUS CONNECTIONS

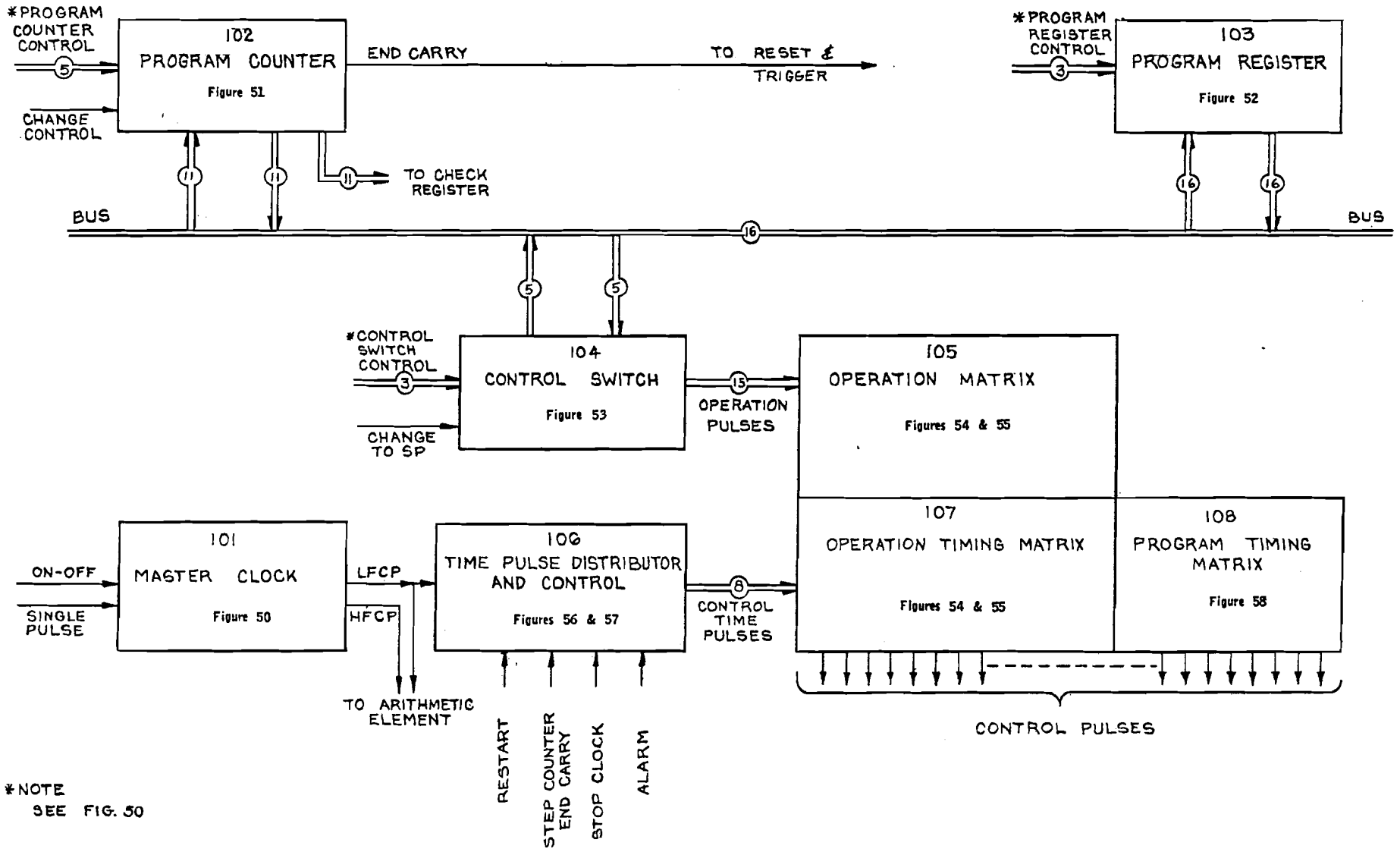


Figure 49

CONTROL

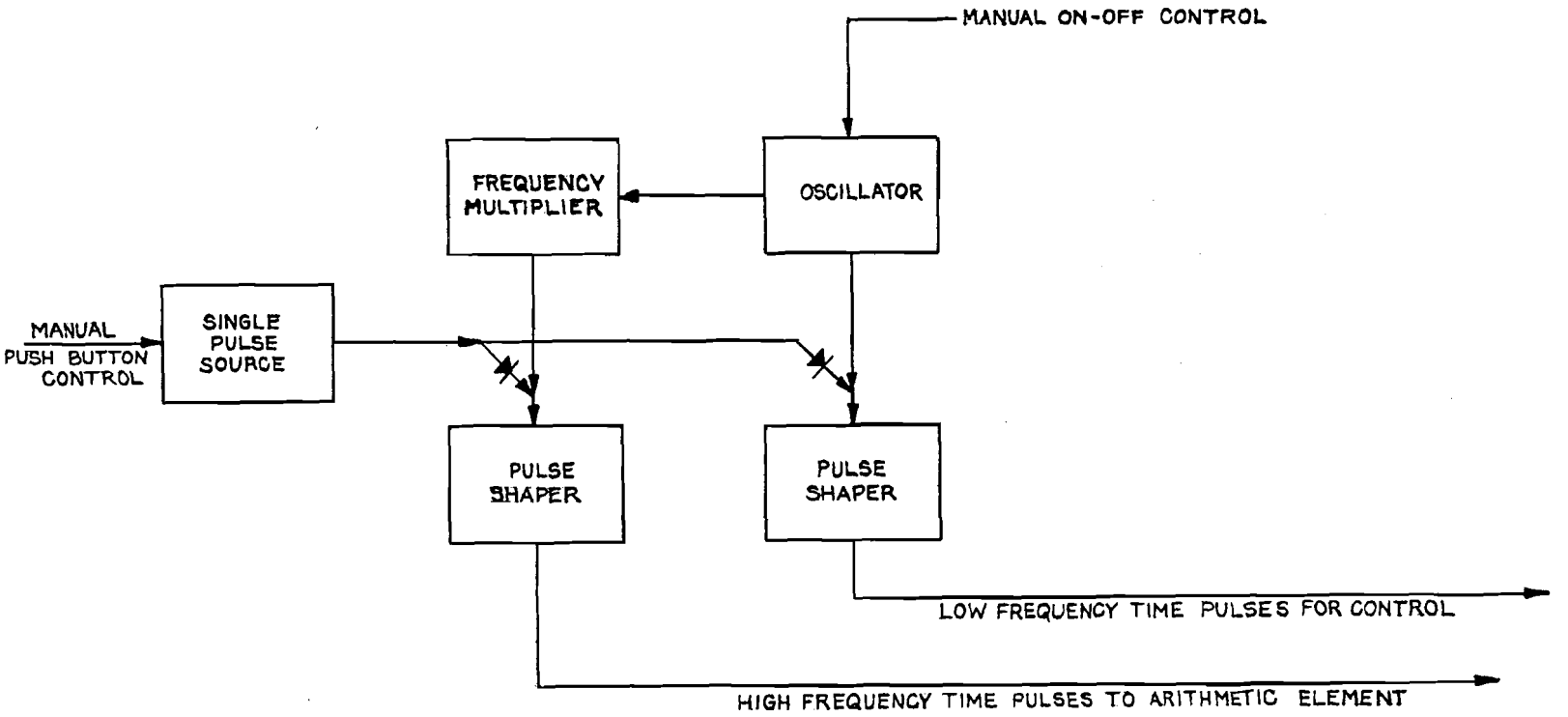


Figure 50

MASTER CLOCK

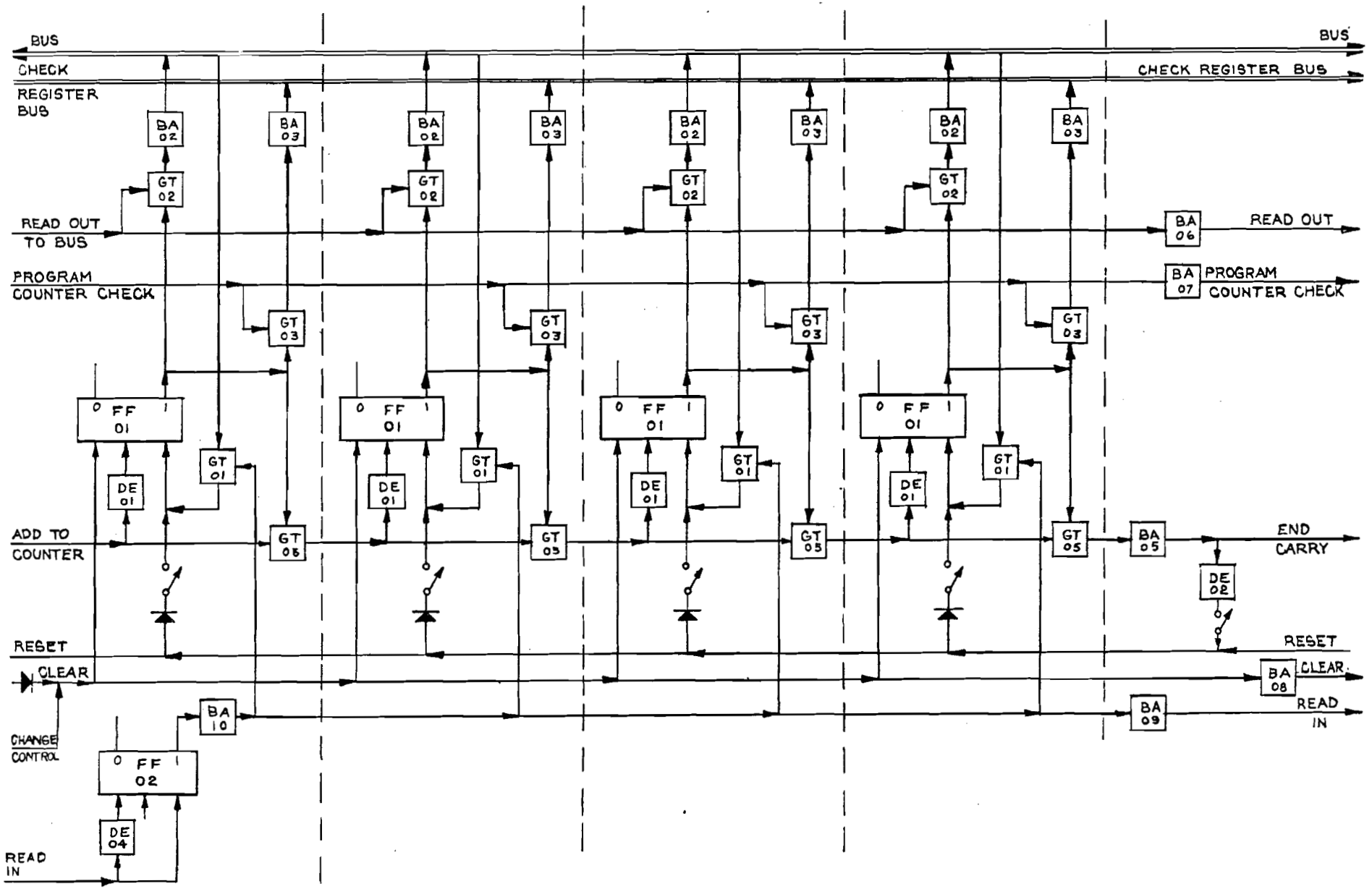


Figure 51

PROGRAM COUNTER

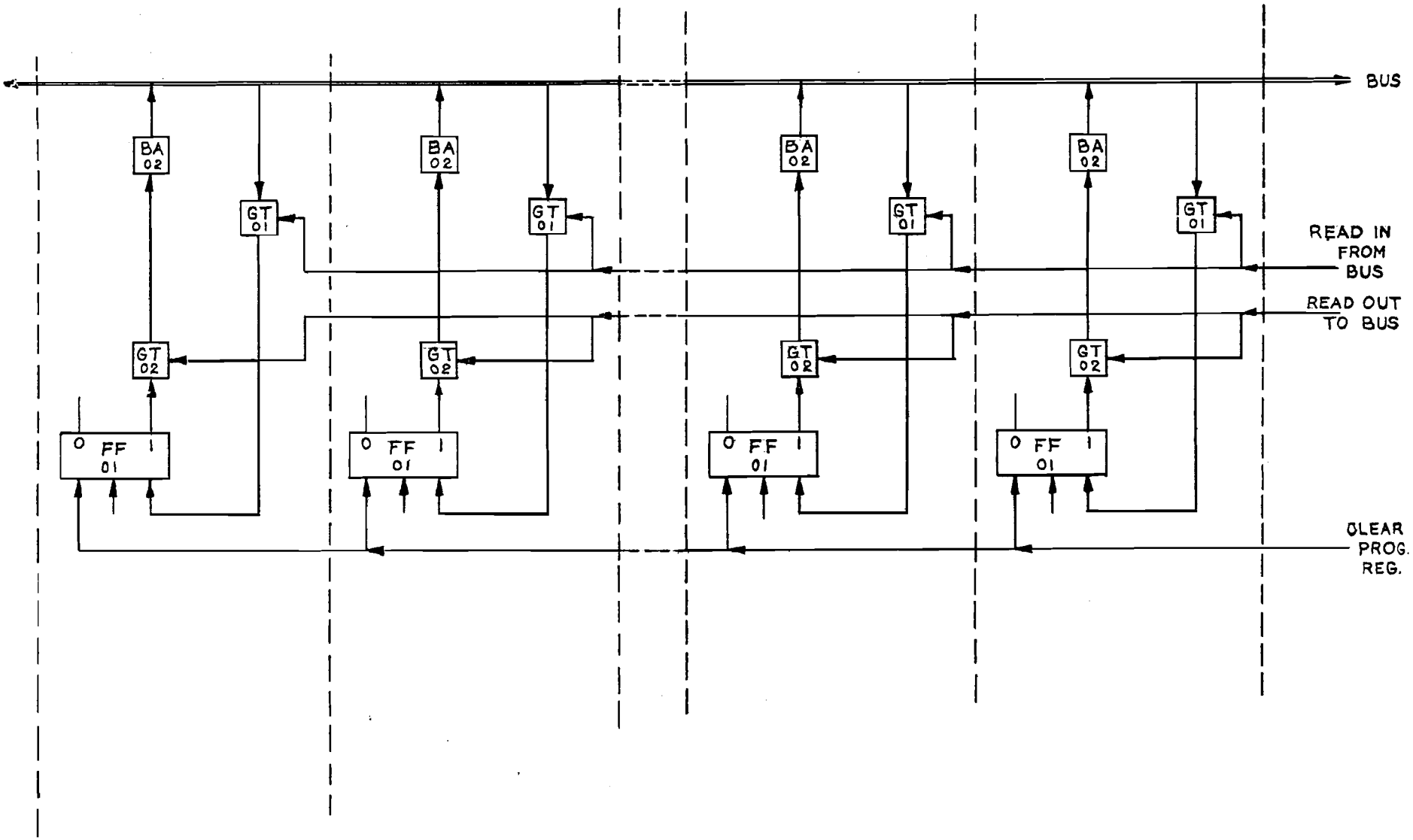


Figure 52

PROGRAM REGISTER

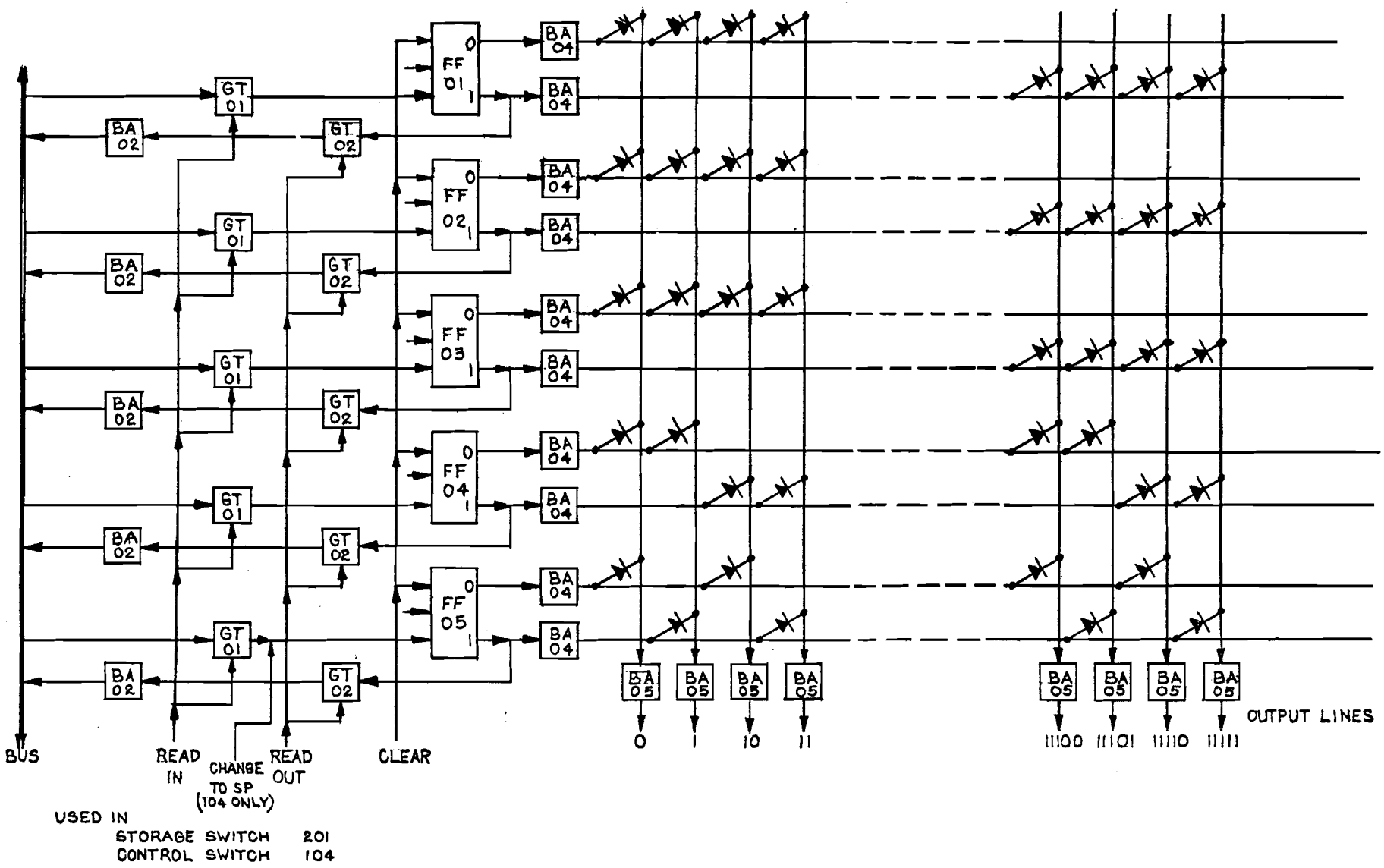


Figure 53

CONTROL SWITCH

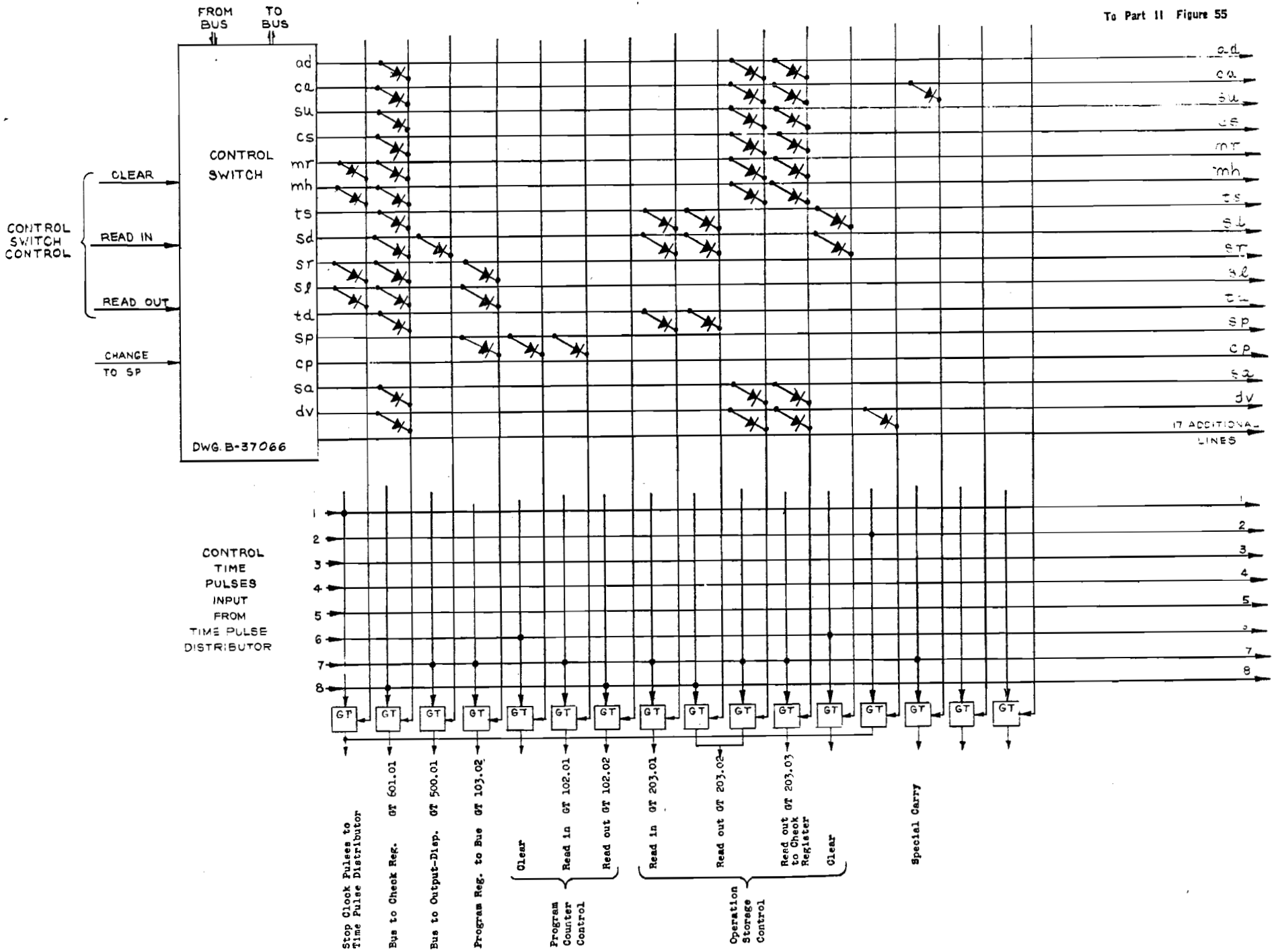


**Figure 54**

**OPERATION MATRIX I**

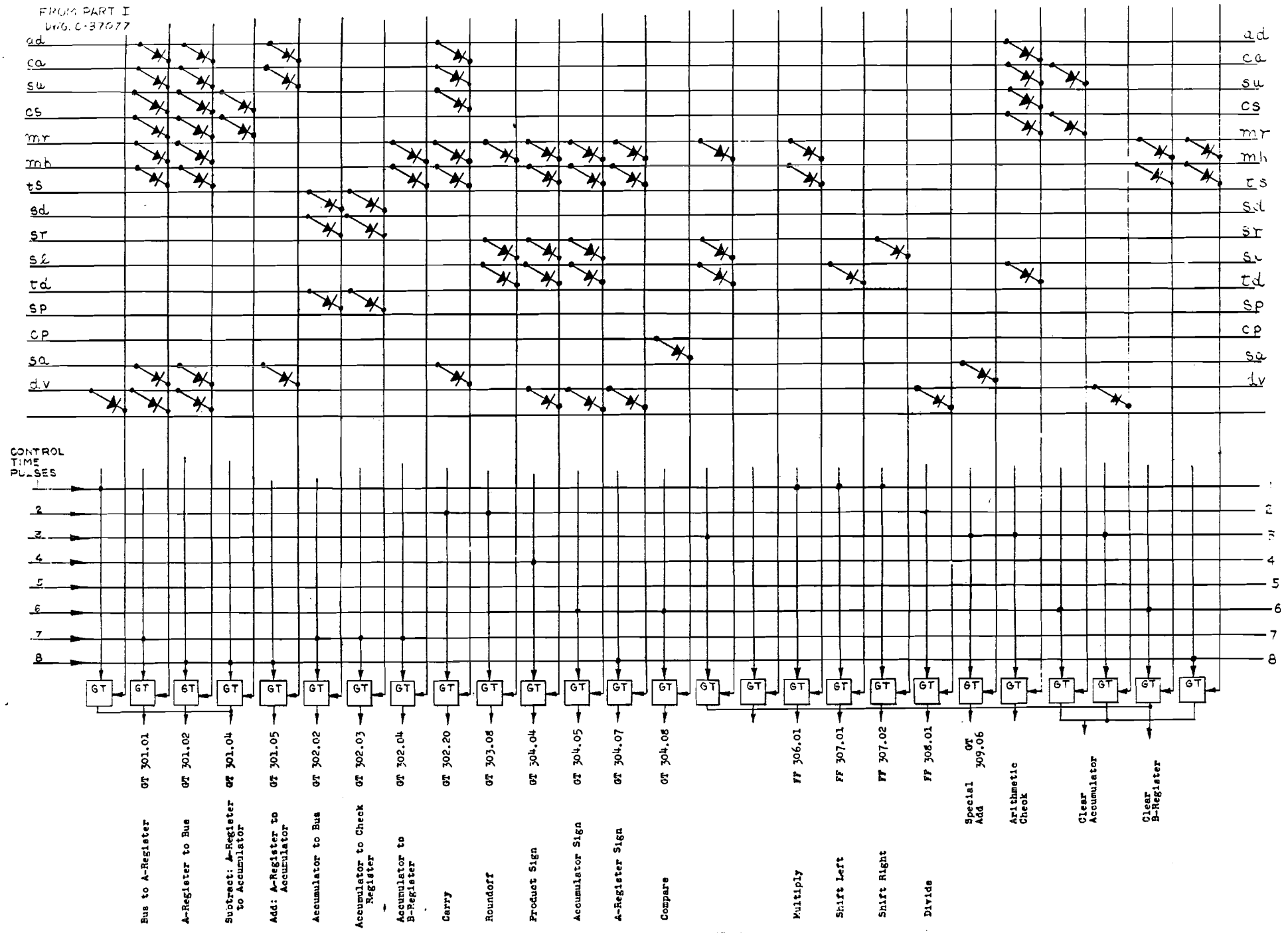


Figure 55

OPERATION MATRIX II

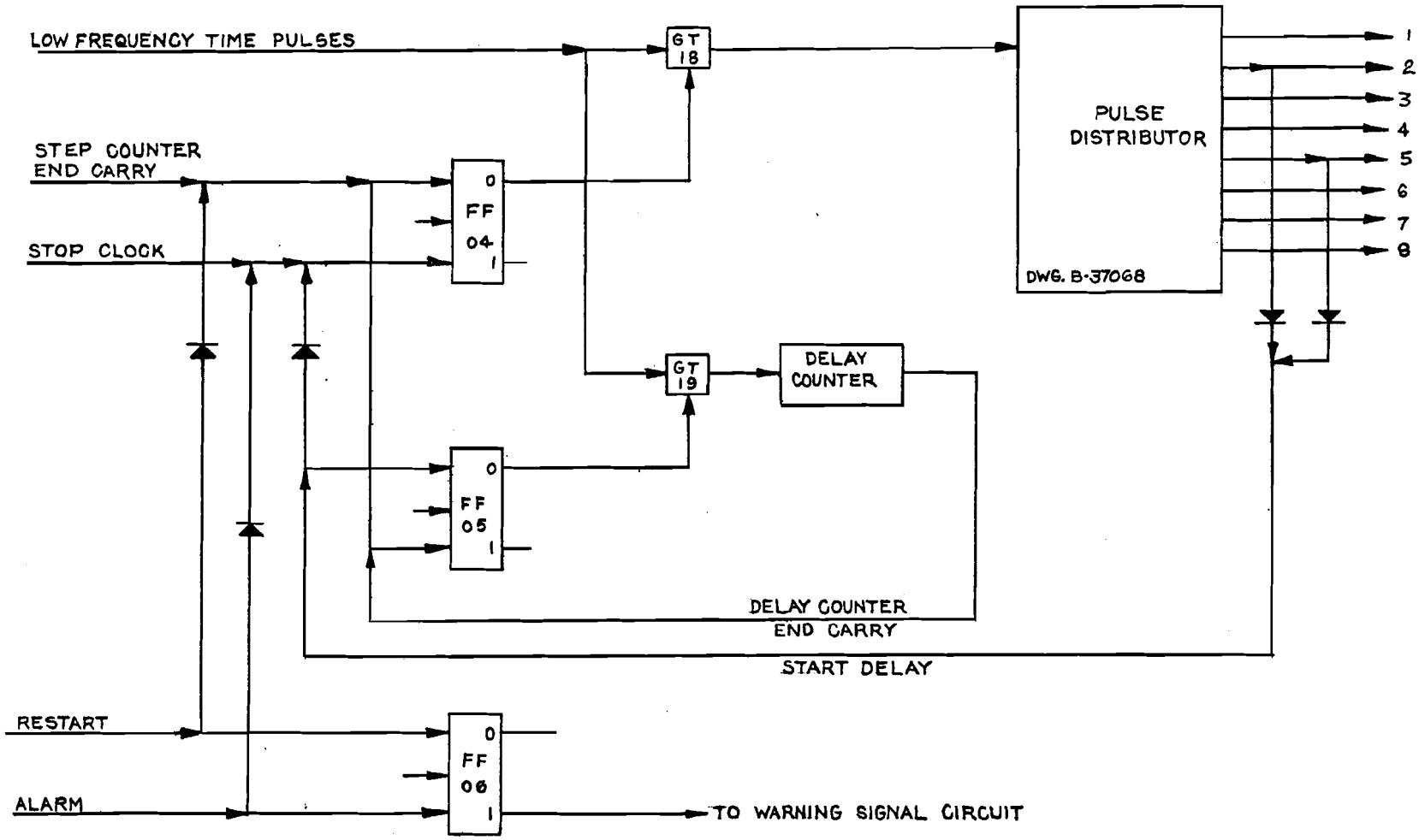


Figure 56

TIME PULSE DISTRIBUTOR CONTROL

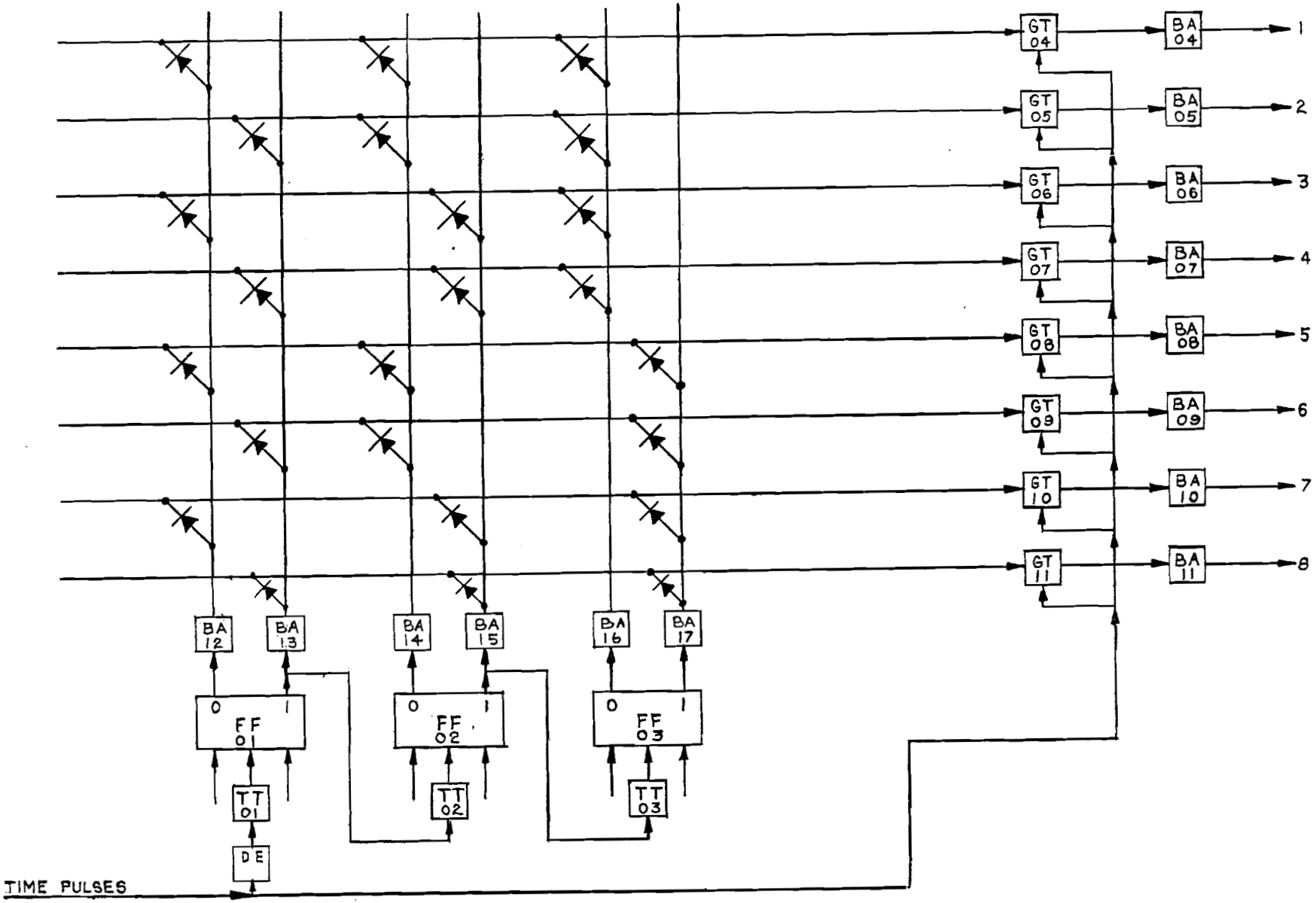


Figure 57

PULSE DISTRIBUTOR

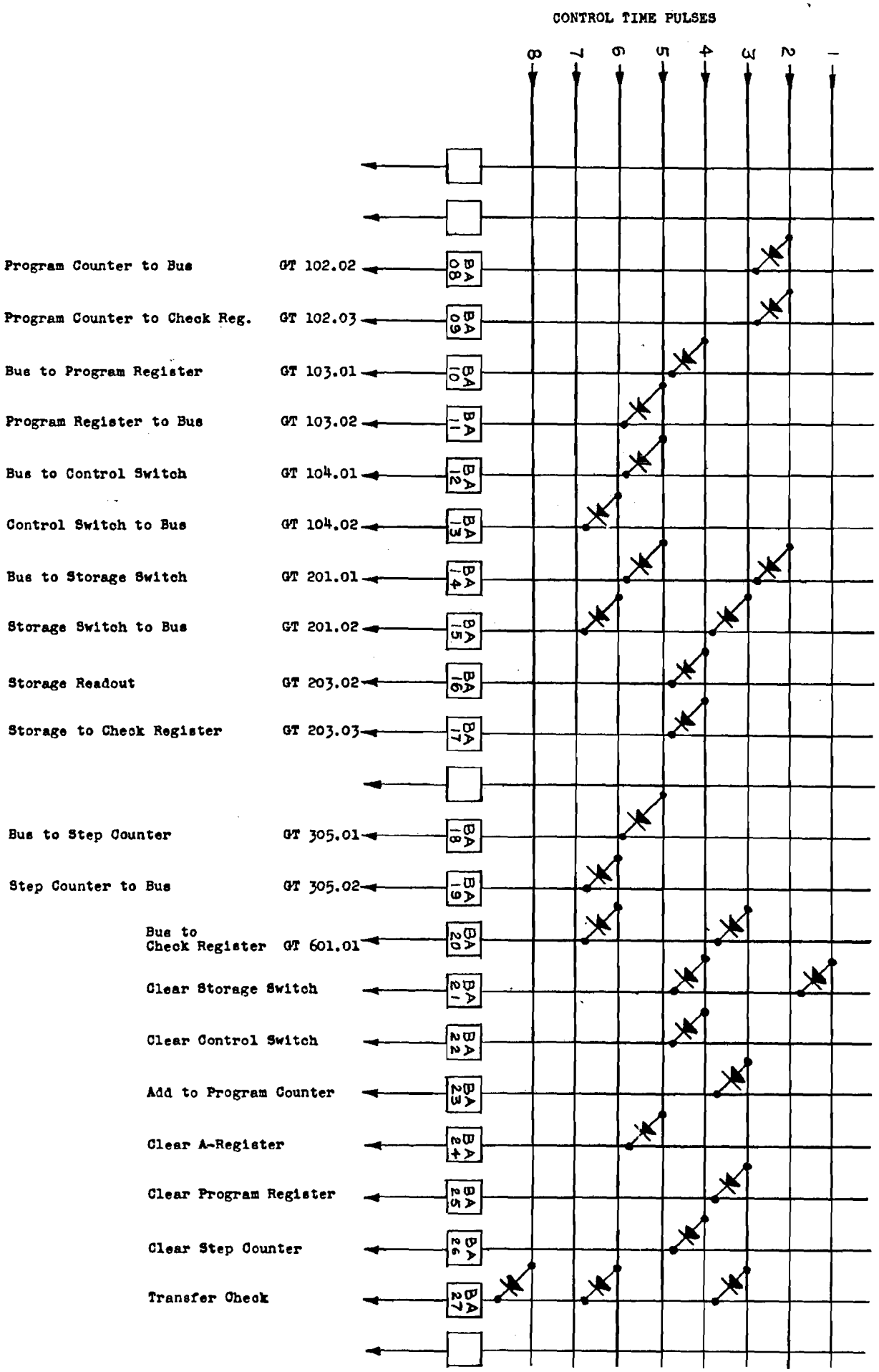


Figure 58

PROGRAM TIMING MATRIX

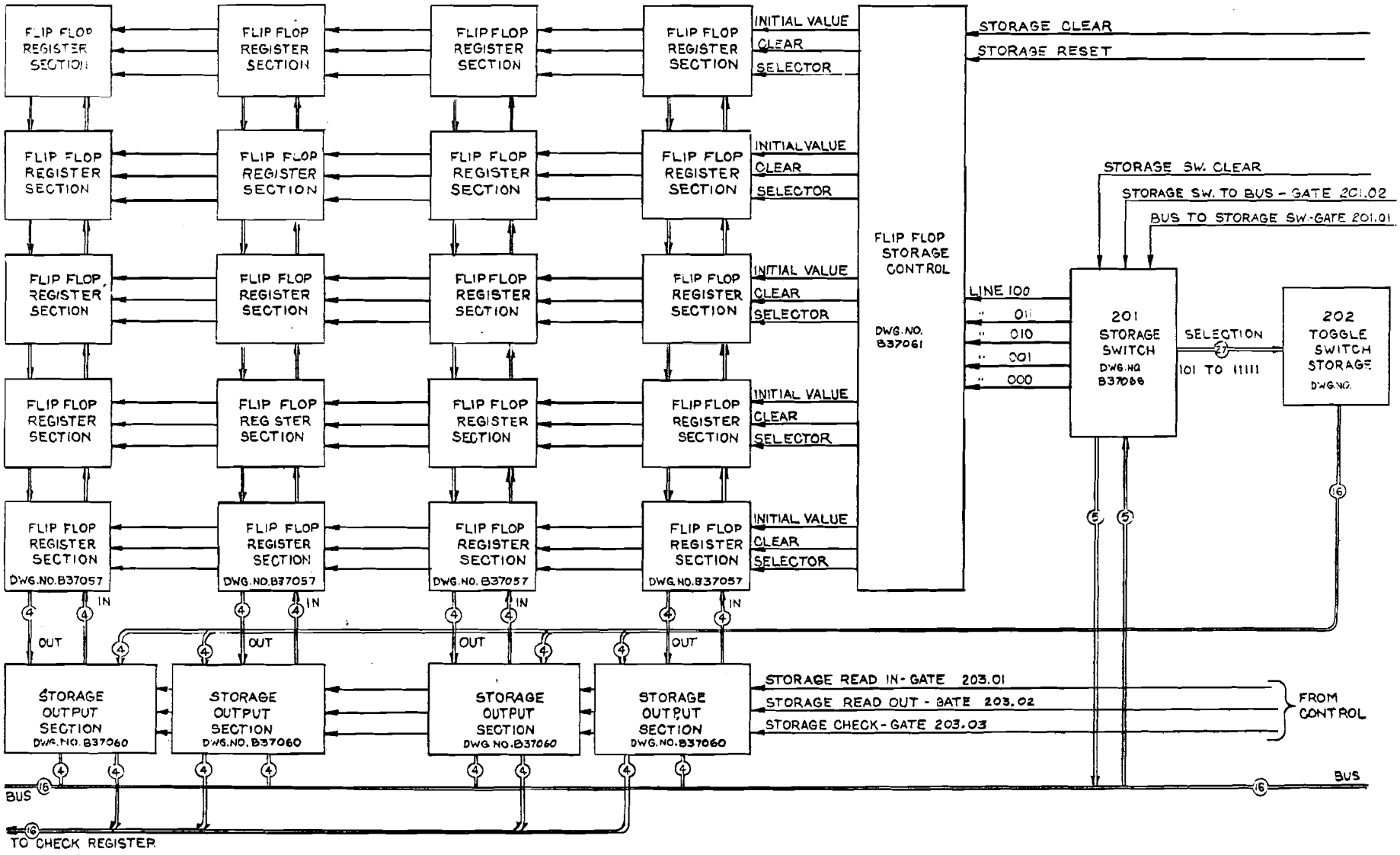


**Figure 59**

**STORAGE CHASSIS ARRANGEMENT**

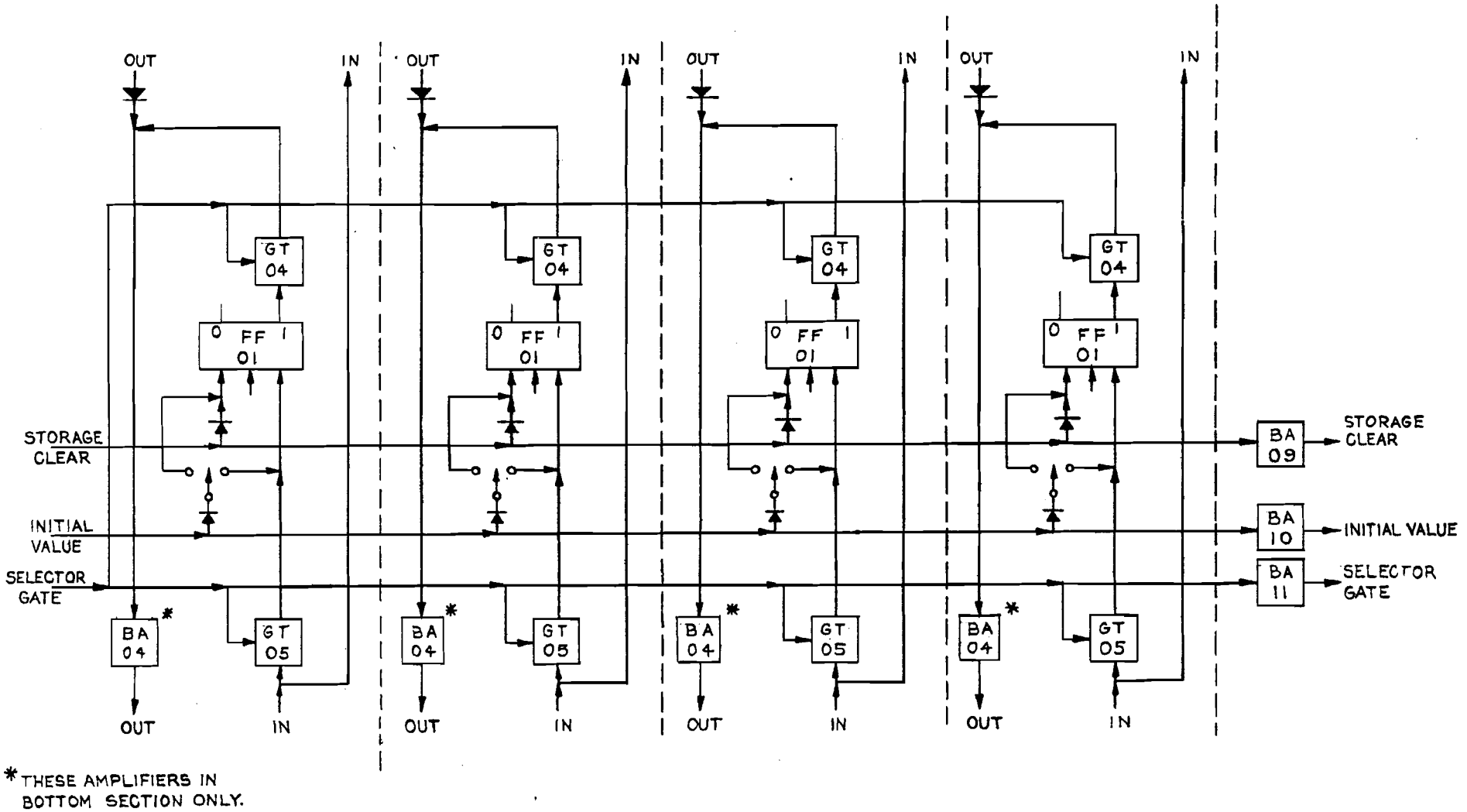


Figure 60

FLIP-FLOP STORAGE SECTION

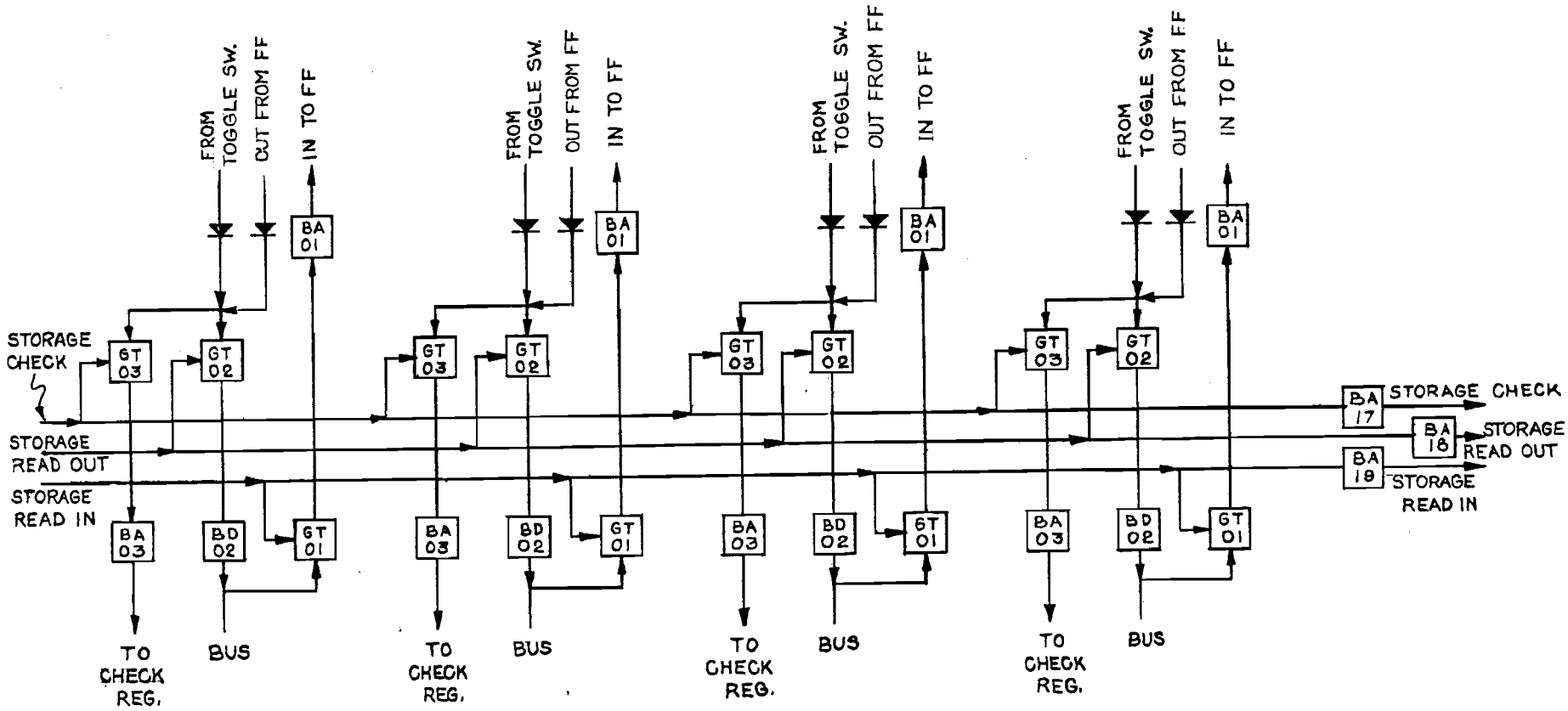


Figure 61

STORAGE OUTPUT SECTION

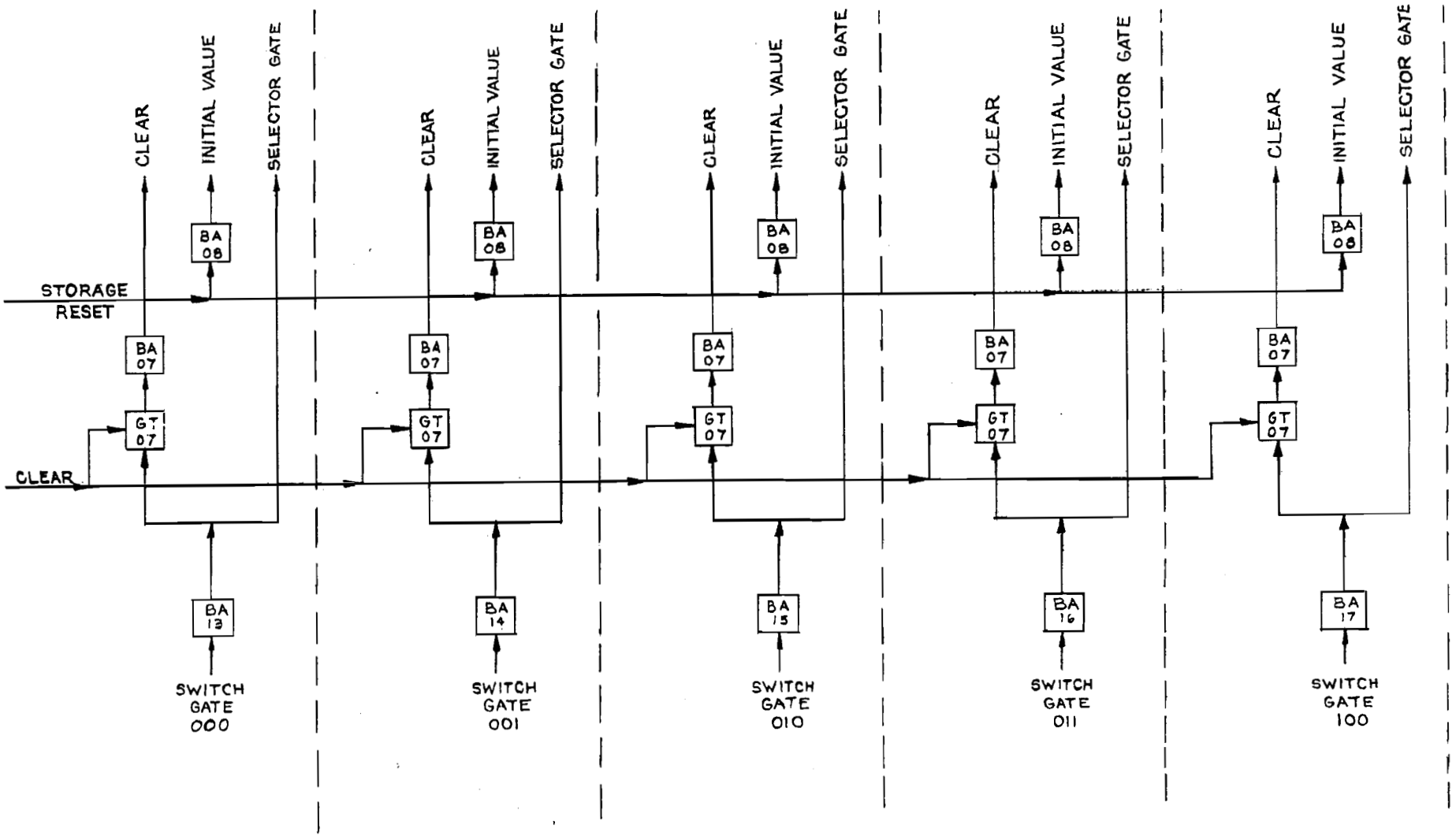


Figure 62

FLIP-FLOP STORAGE CONTROL

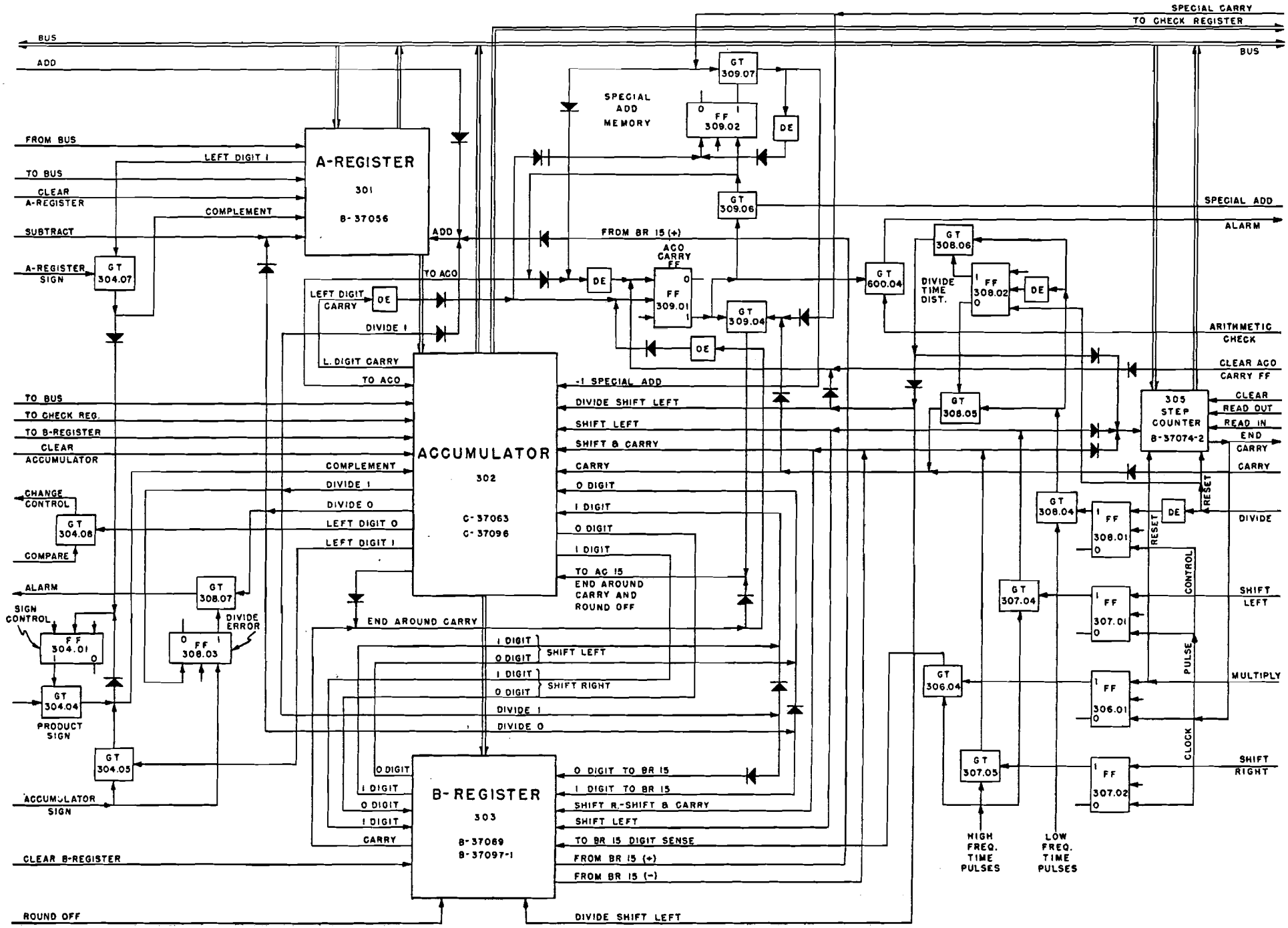


Figure 63

ARITHMETIC ELEMENT

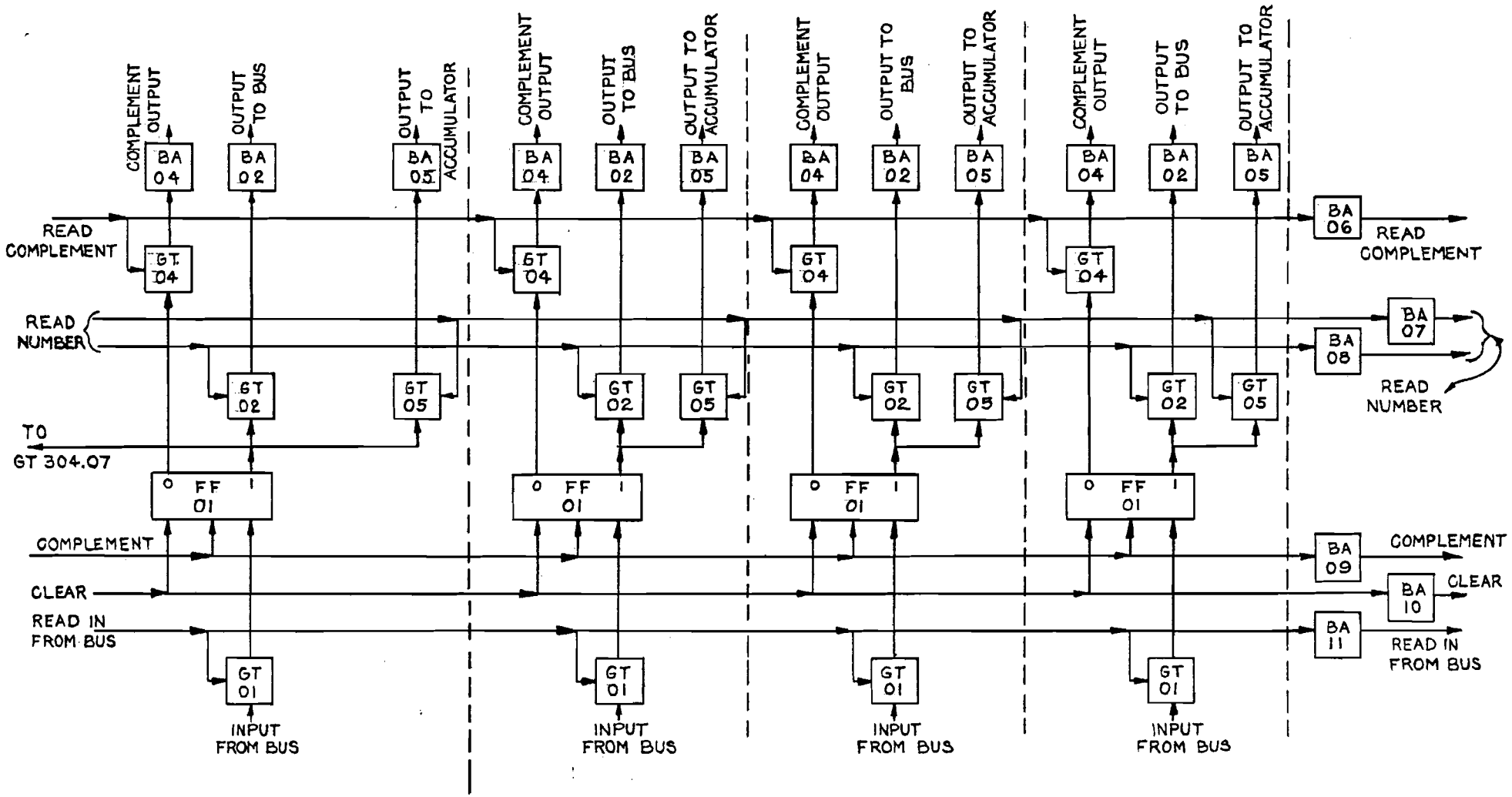


Figure 64

SECTION OF A-REGISTER

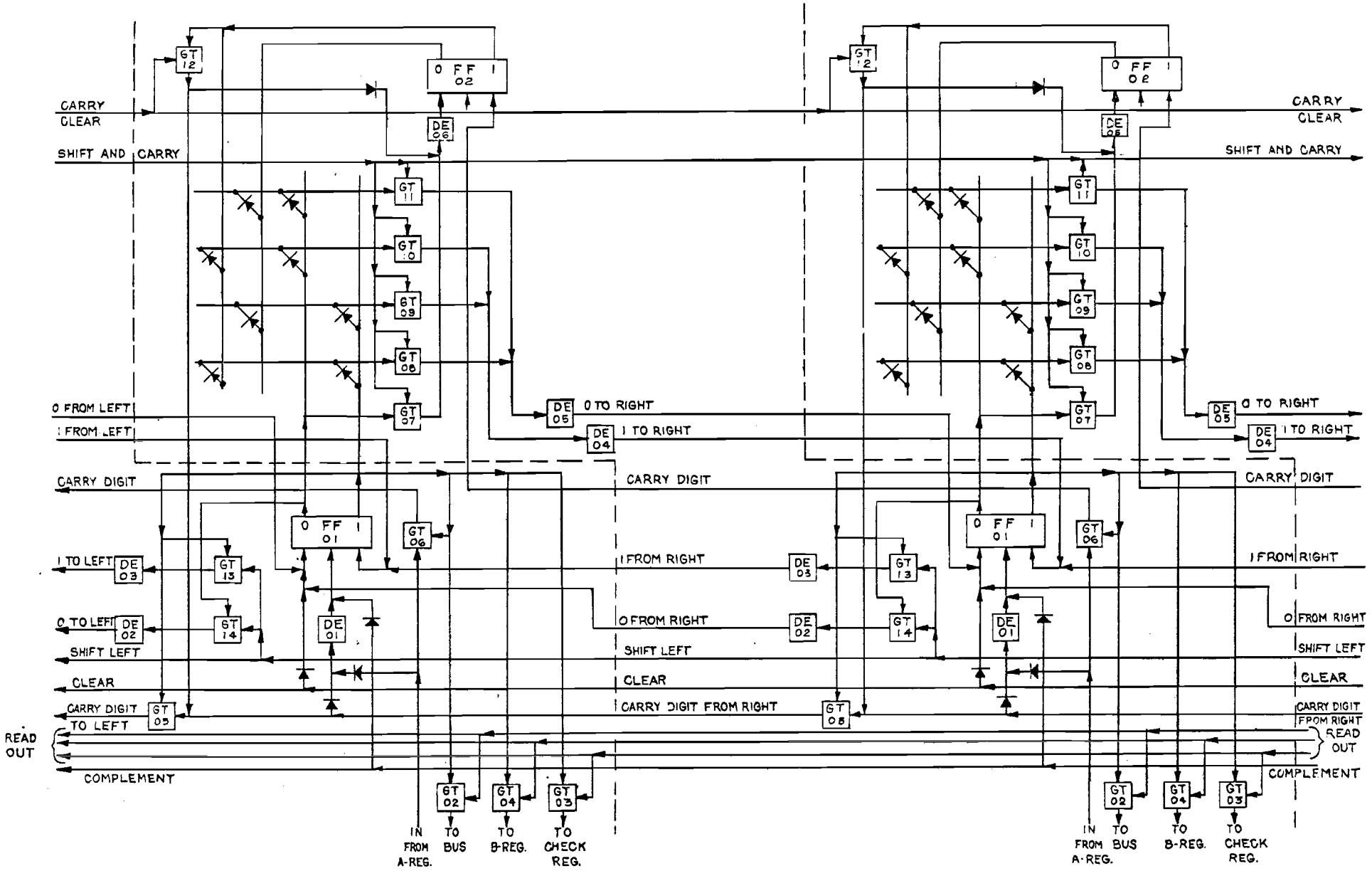


**Figure 65**

**ACCUMULATOR SECTIONS**

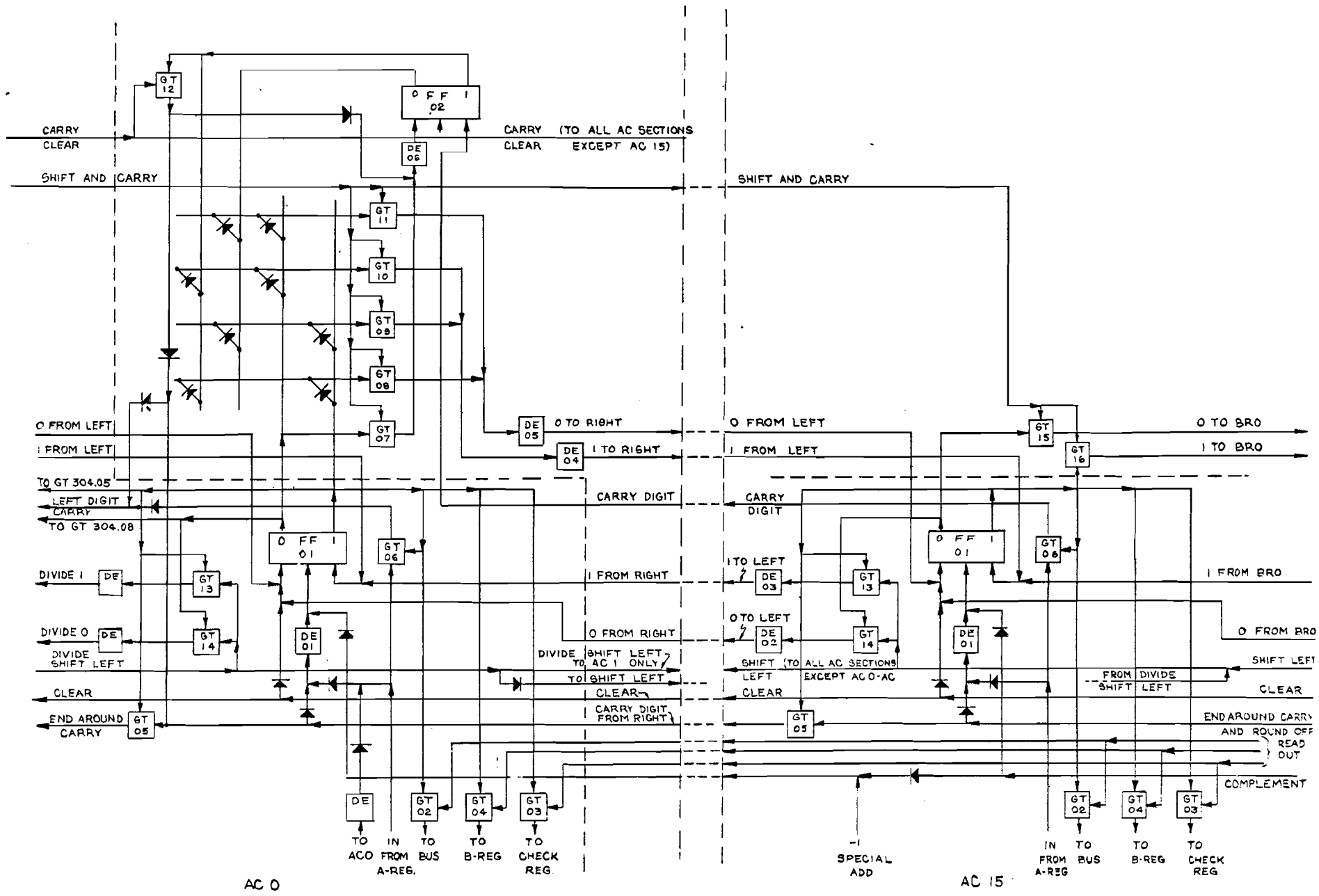


**Figure 66**

**ACCUMULATOR SECTIONS AC0-AC15**

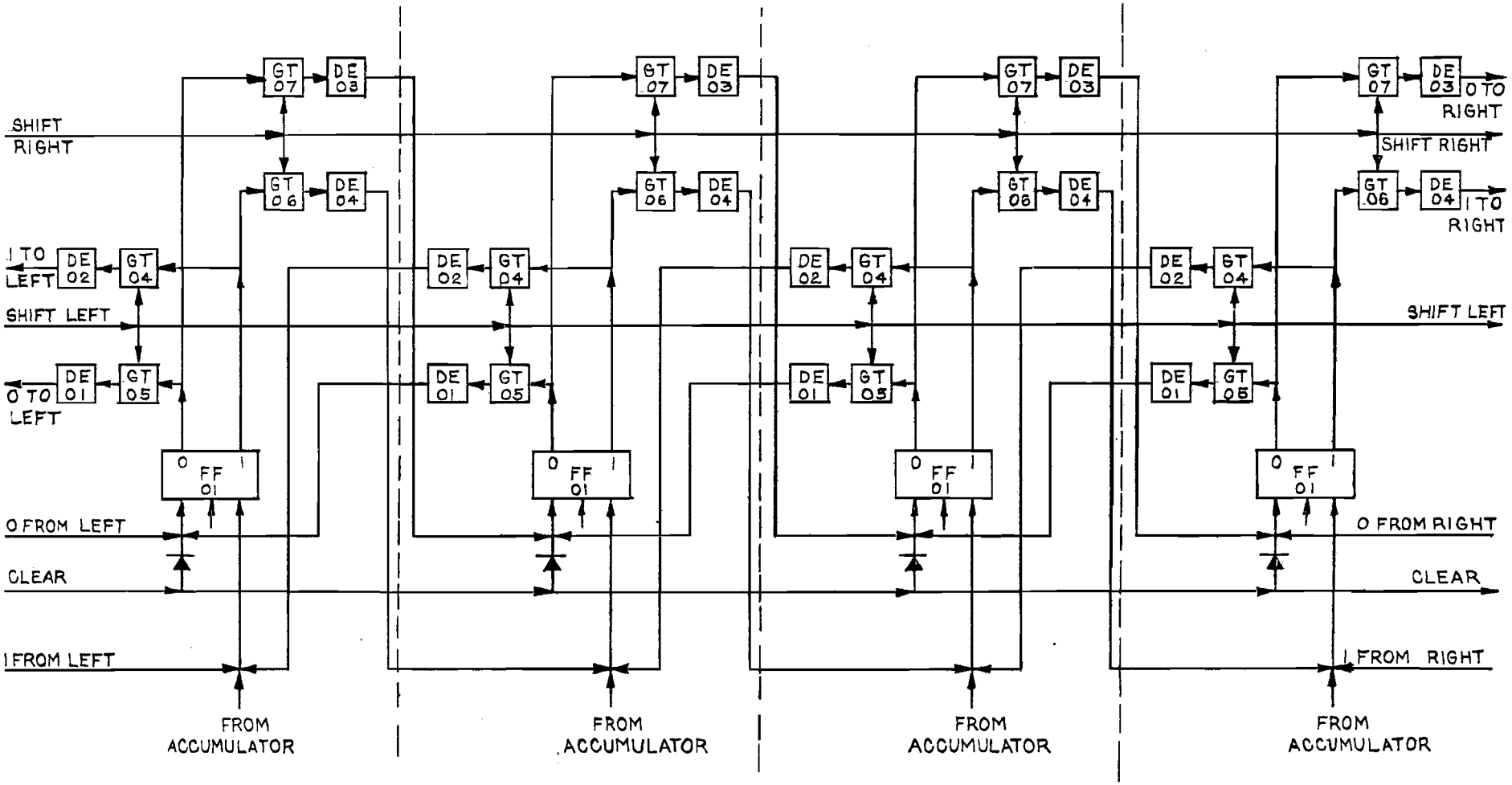


**Figure 67**

**B-REGISTER SECTIONS**

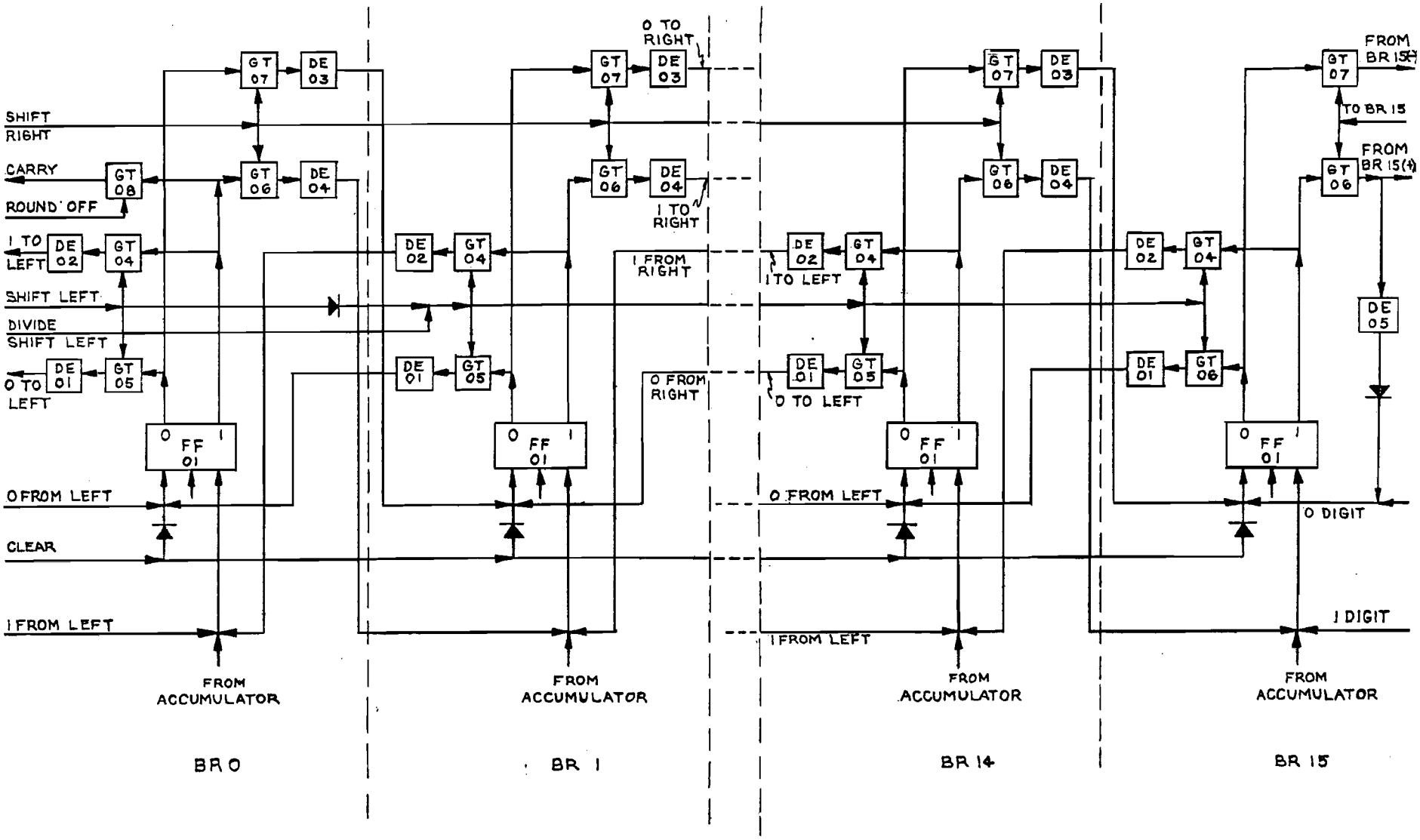


Figure 68

B-REGISTER SECTIONS BR 0, 1, 14, 15

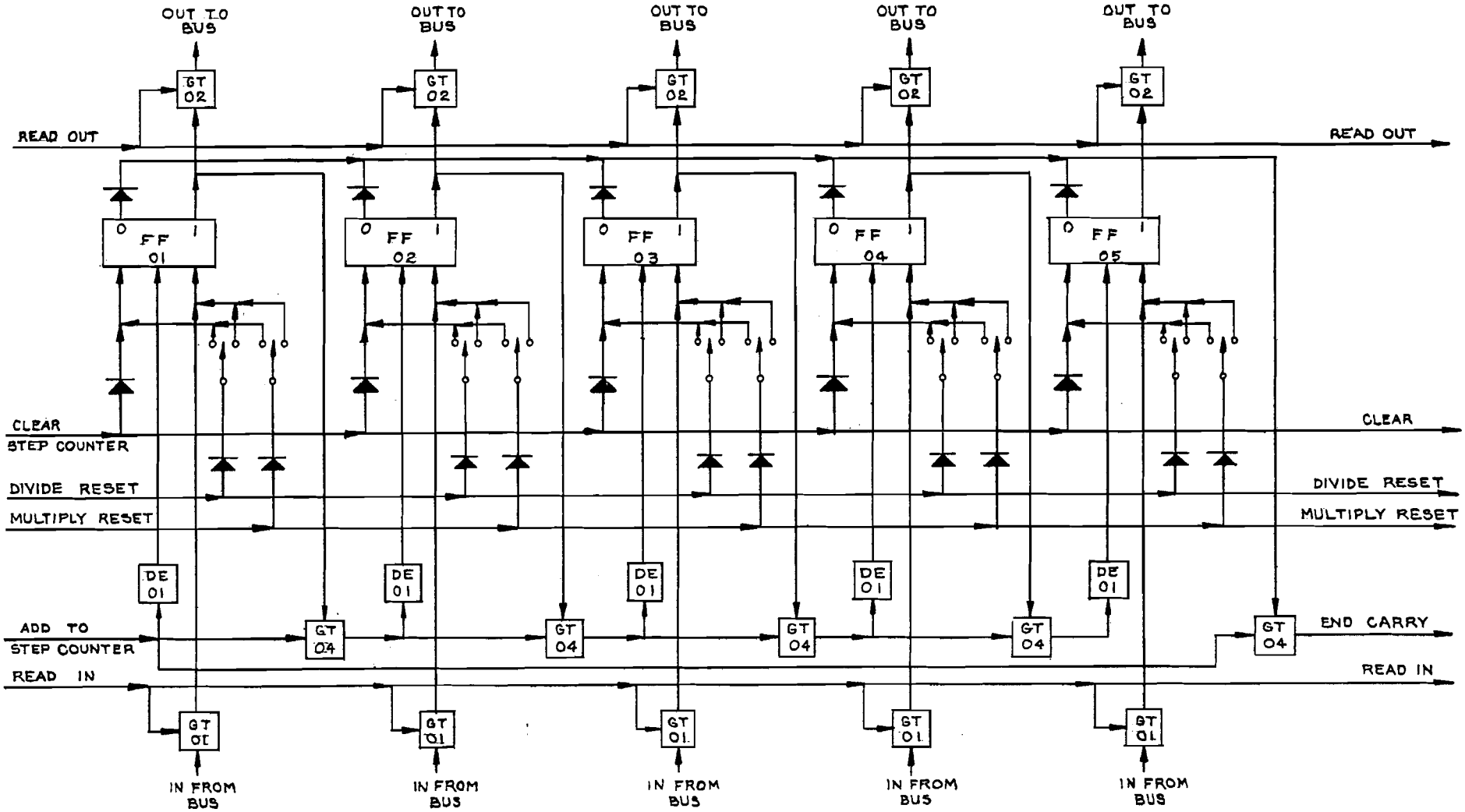


Figure 69

STEP COUNTER

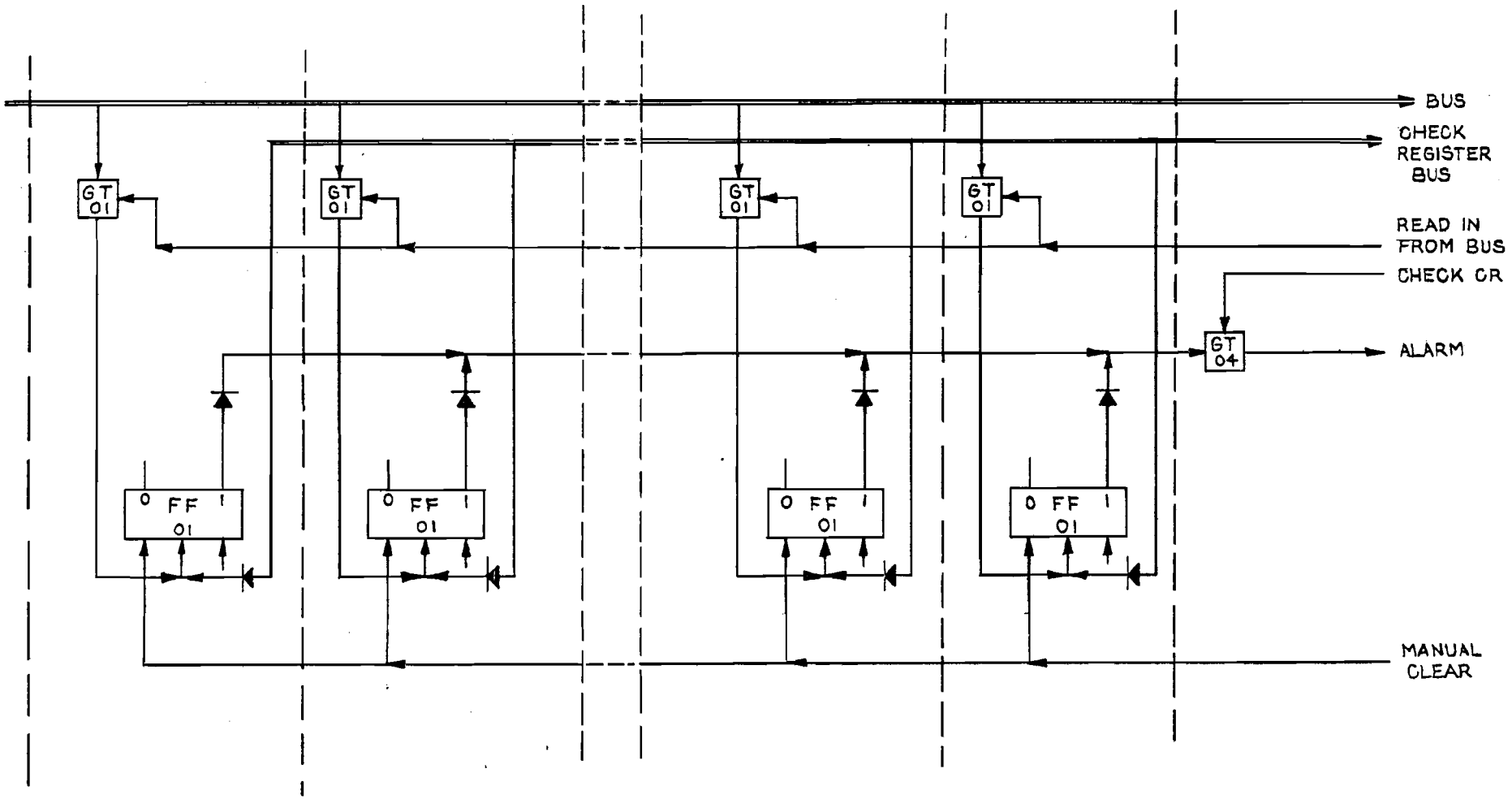


Figure 70
CHECK REGISTER

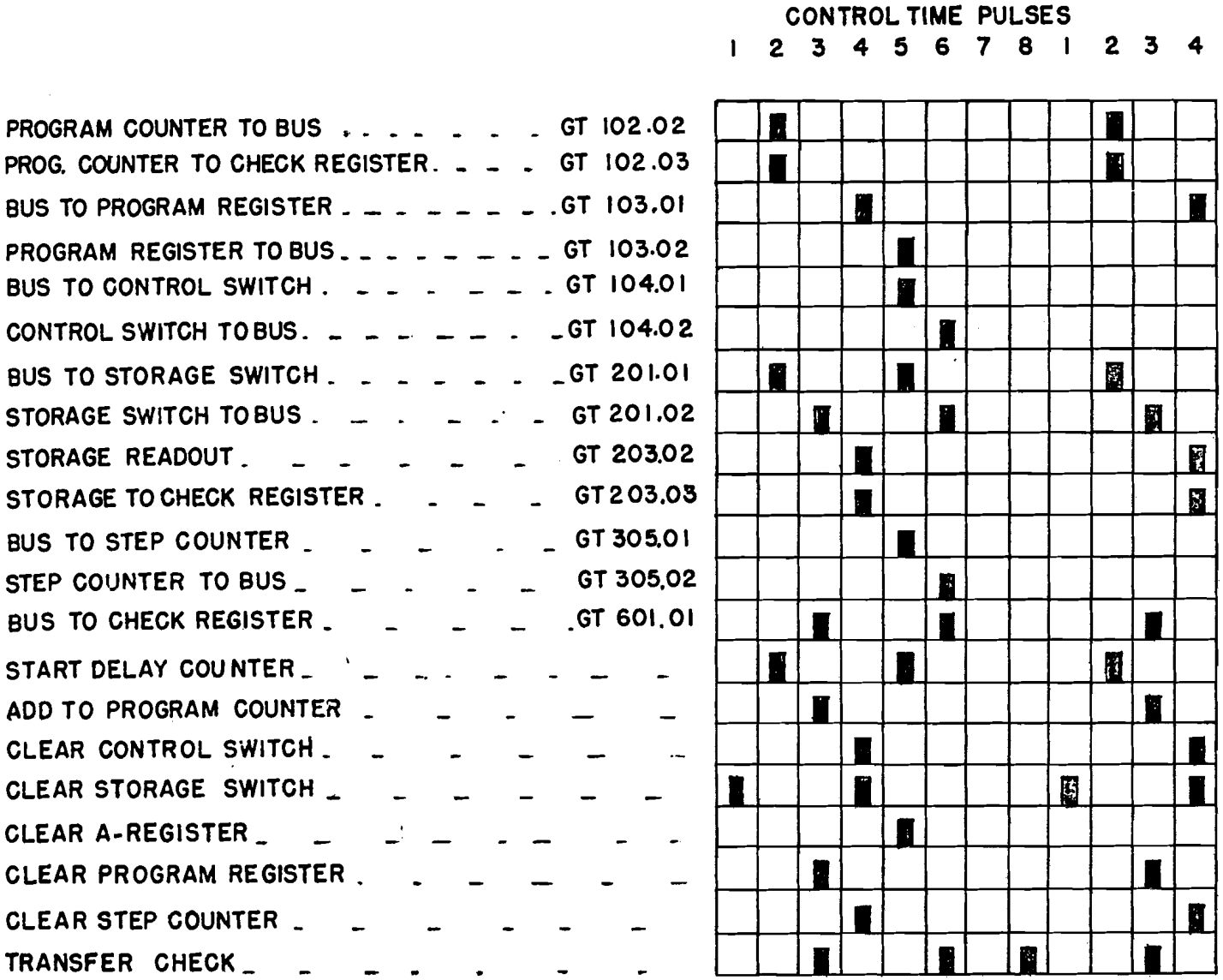


Figure 71

PROGRAM TIMING

OPERATION: ADD ad

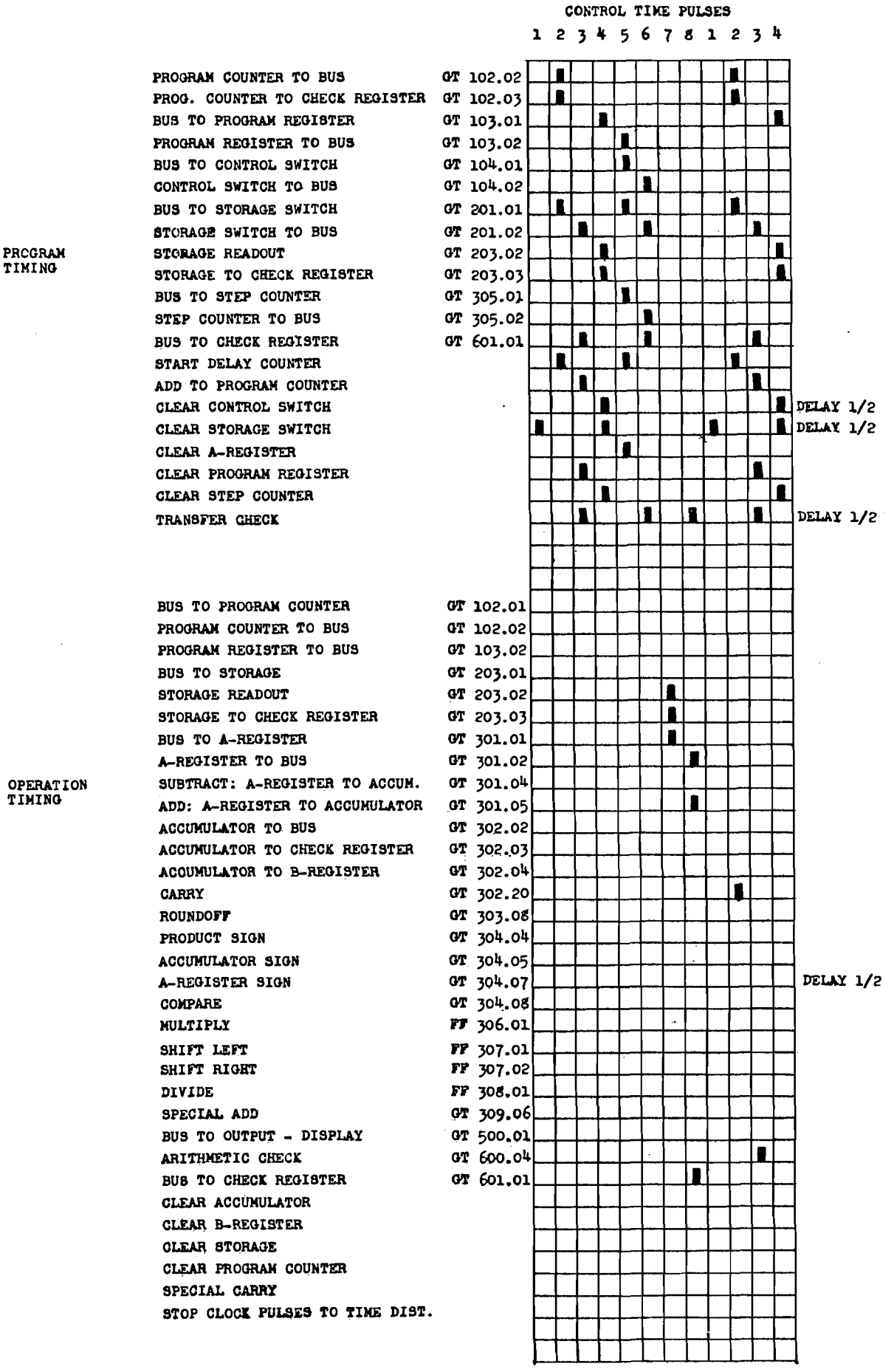


Figure 72

TIMING FOR ADD

OPERATION: CLEAR AND ADD ca

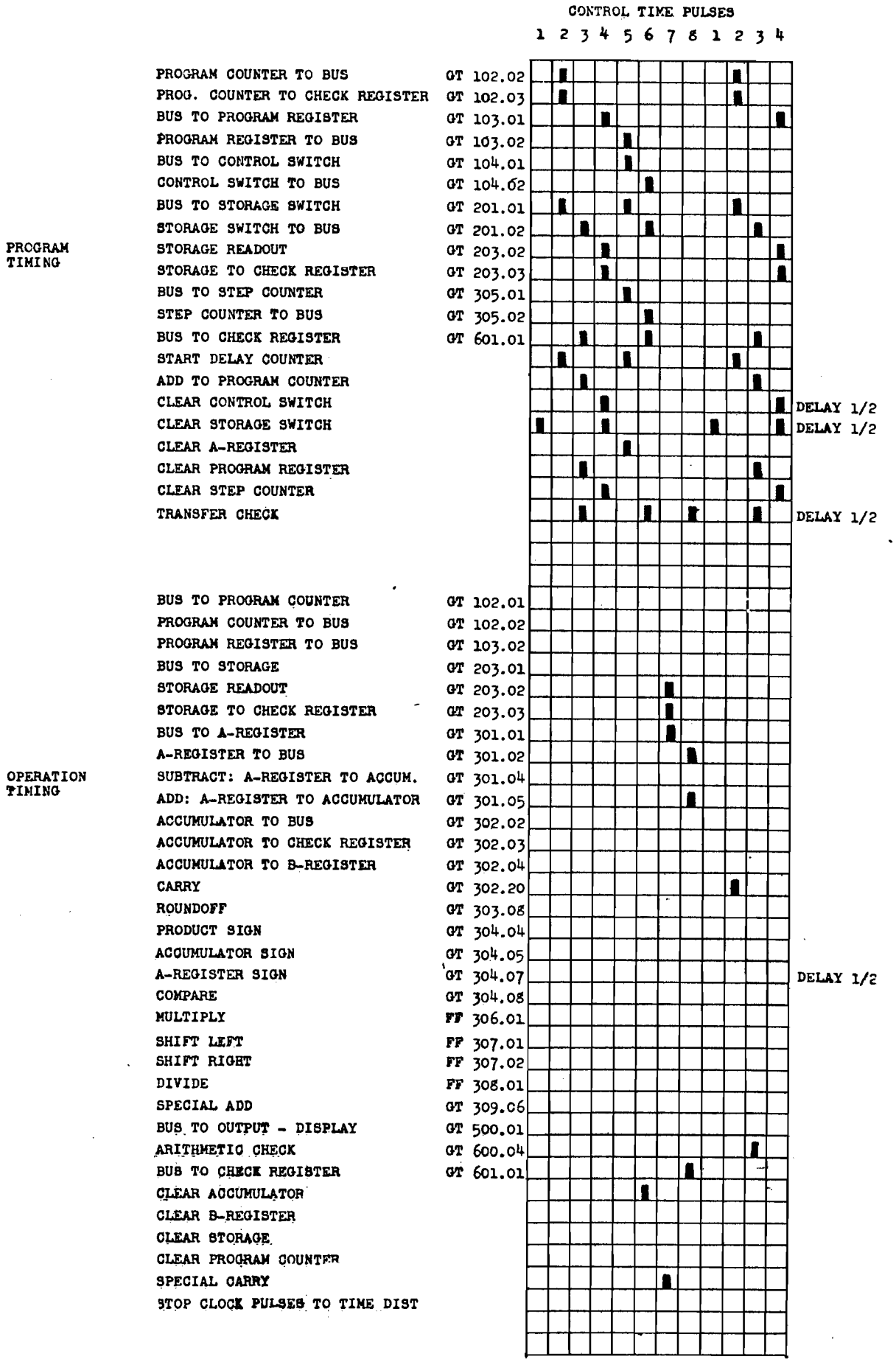


Figure 73

TIMING FOR CLEAR AND ADD

OPERATION: SUBTRACT su

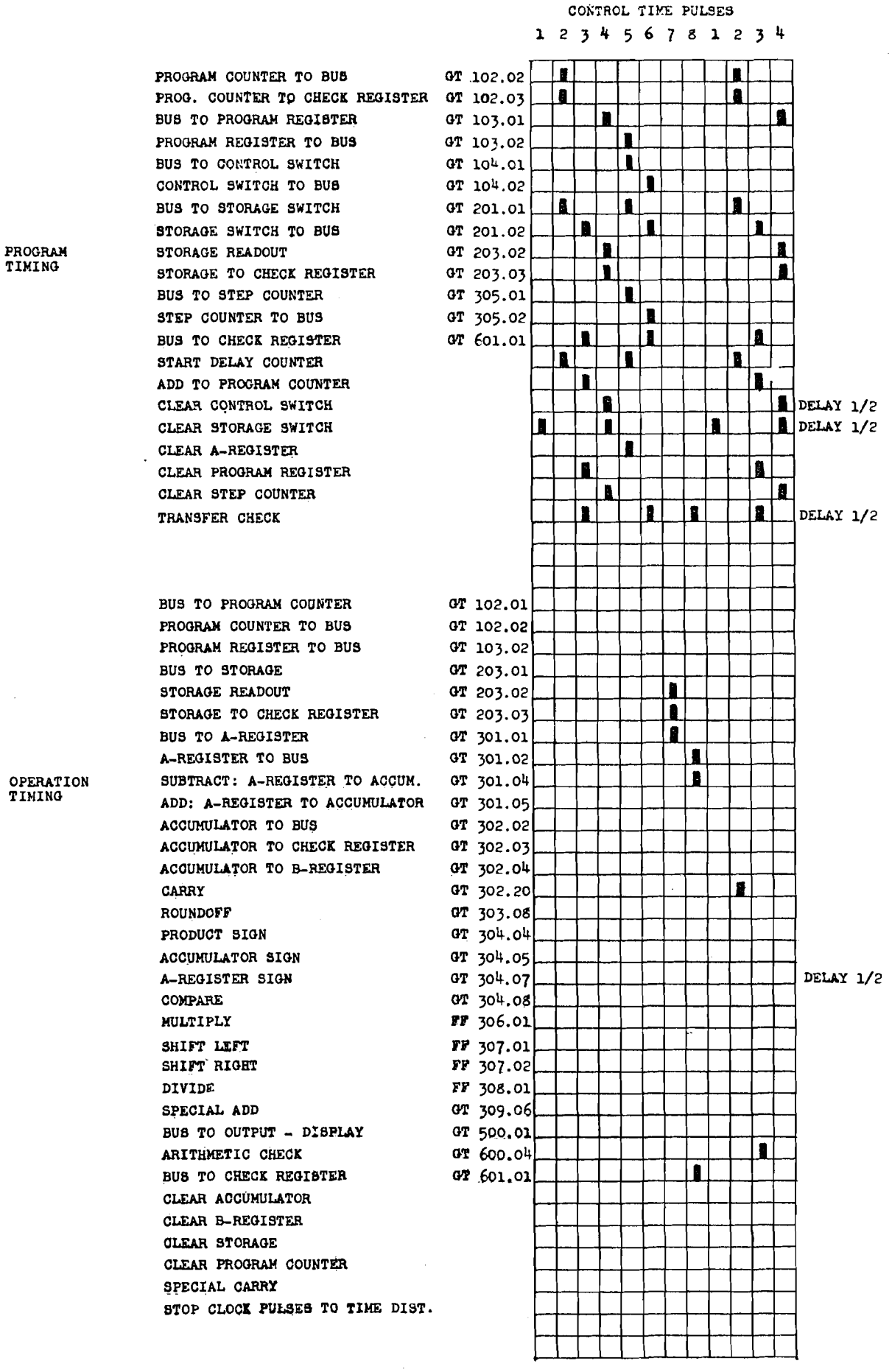


Figure 74

TIMING FOR SUBTRACT

OPERATION: CLEAR AND SUBTRACT 08

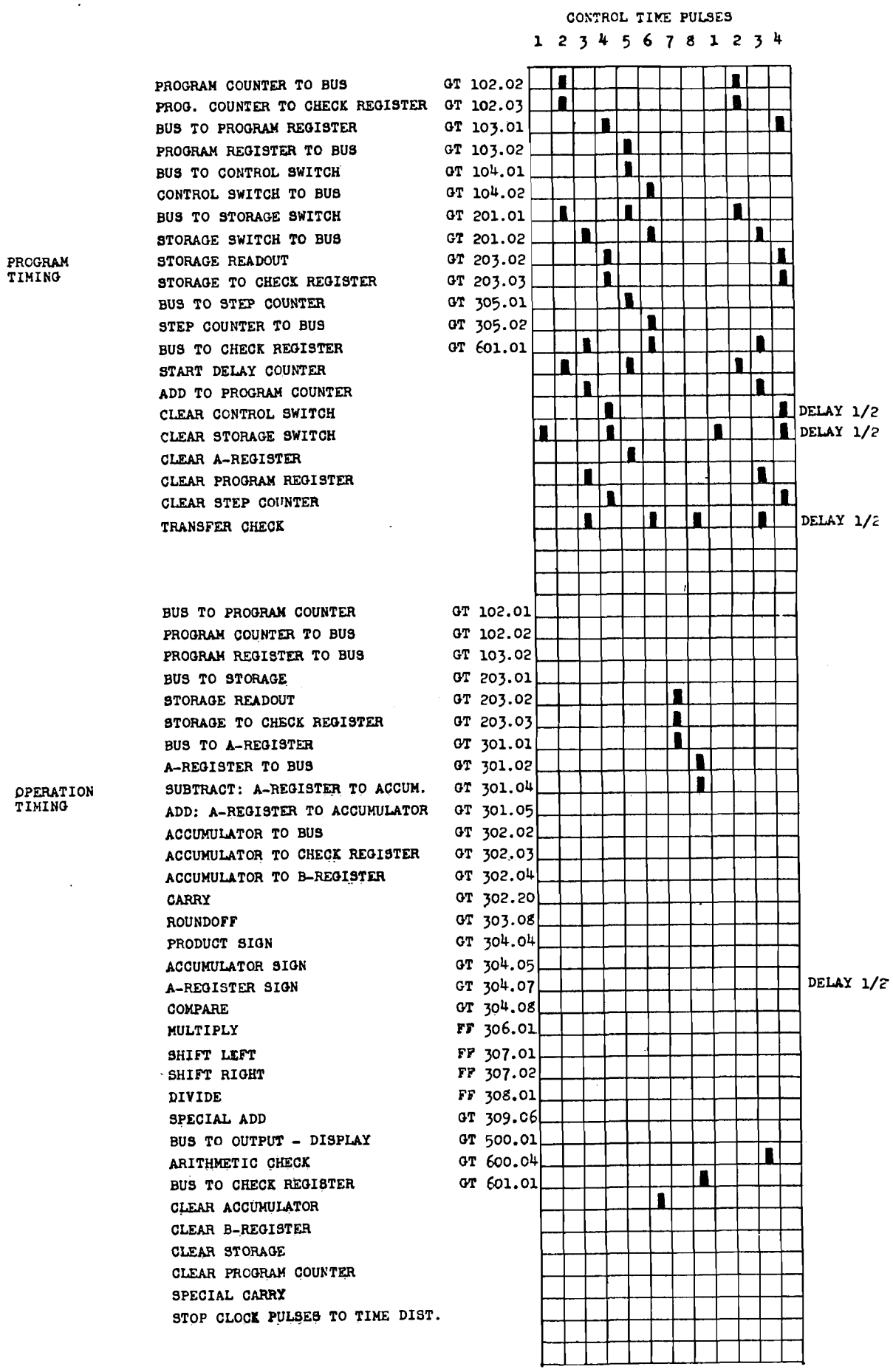


Figure 75

TIMING FOR CLEAR AND SUBTRACT

OPERATION: MULTIPLY AND ROUNDOFF mr

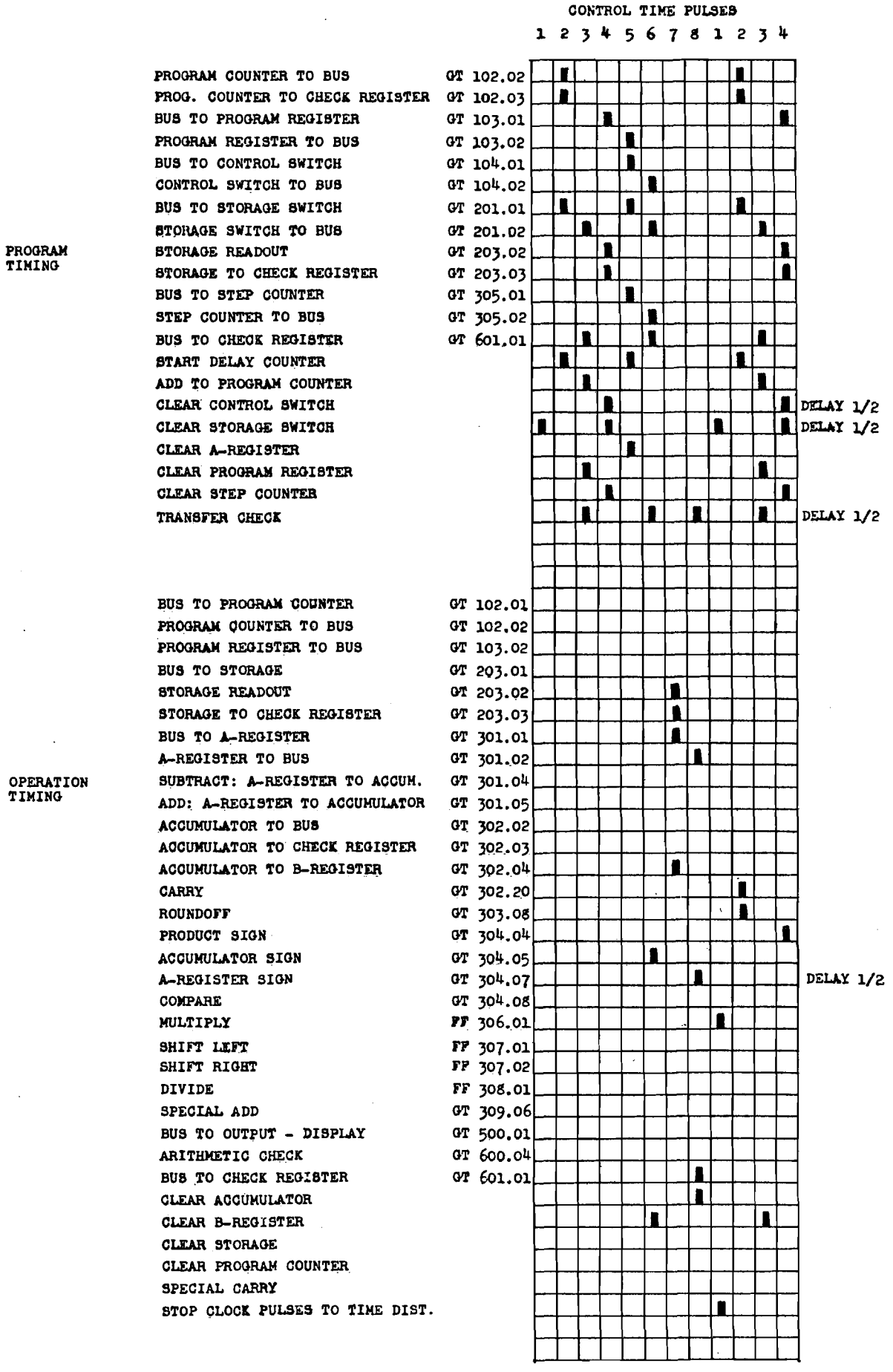


Figure 76

TIMING FOR MULTIPLY AND ROUNDOFF

OPERATION: MULTIPLY AND HOLD FULL PRODUCT mh

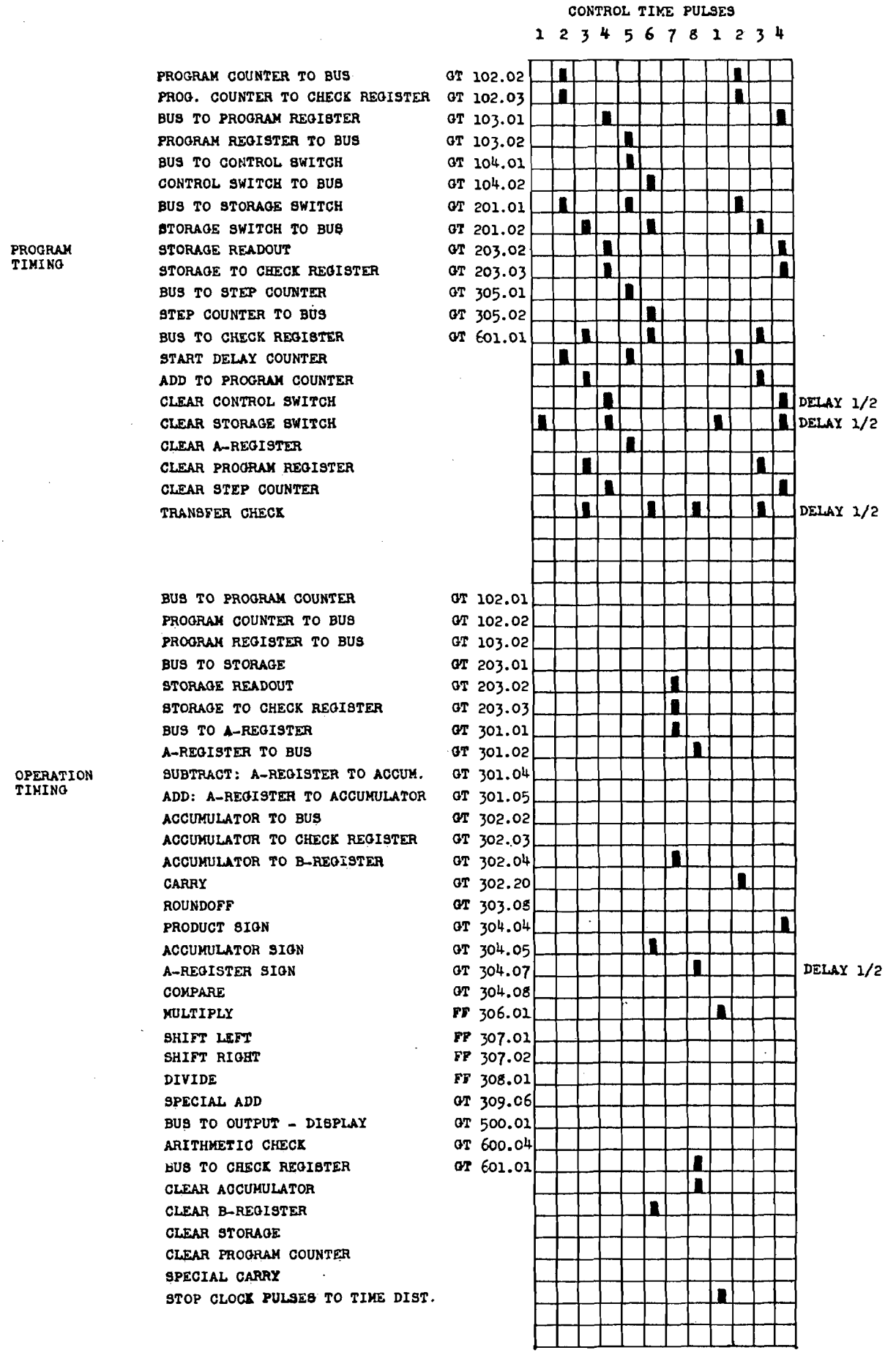


Figure 77

TIMING FOR MULTIPLY AND HOLD FULL PRODUCT

OPERATION: DIVIDE dv

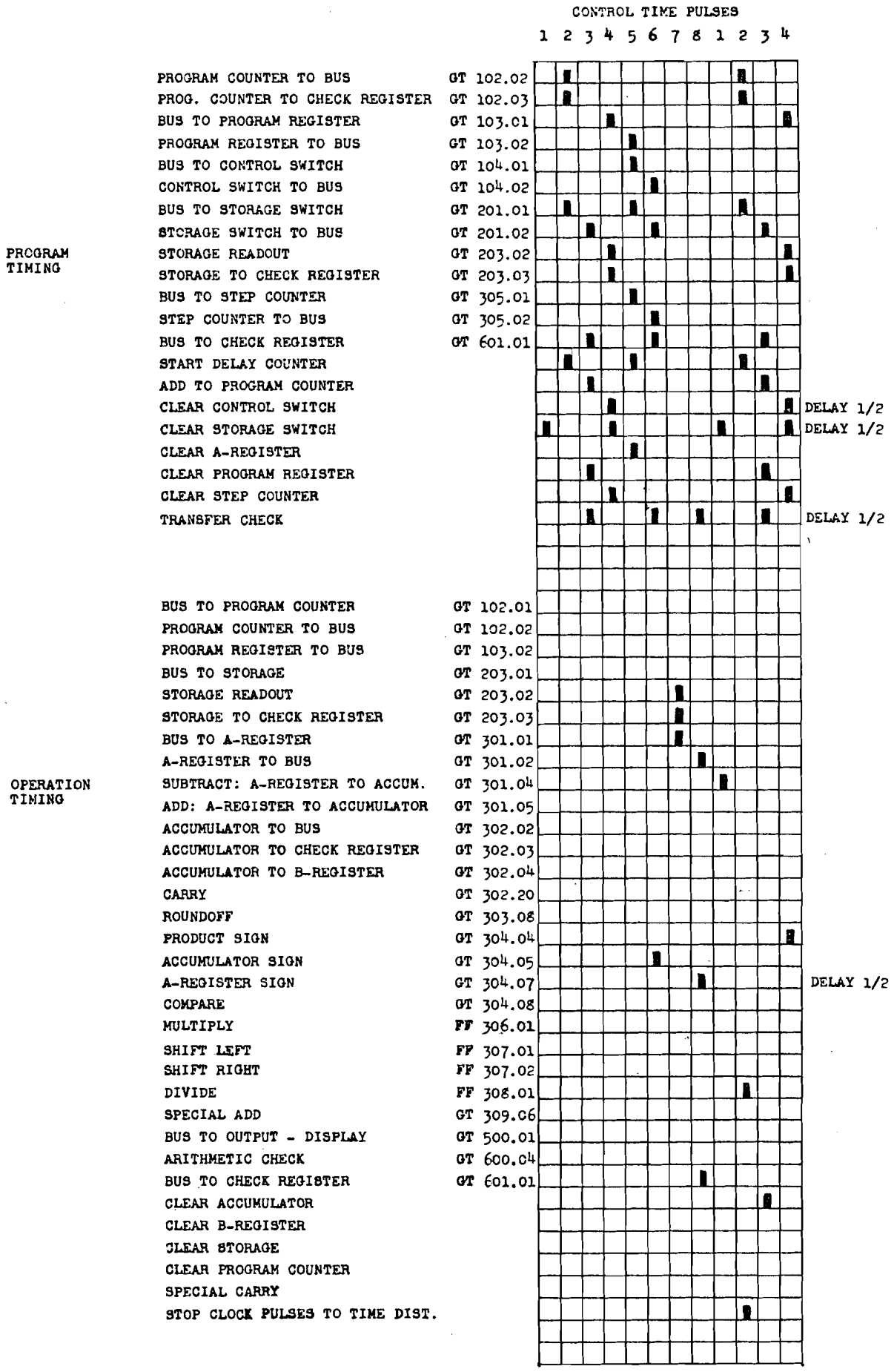


Figure 78

TIMING FOR DIVIDE

OPERATION: TRANSFER TO STORAGE ts

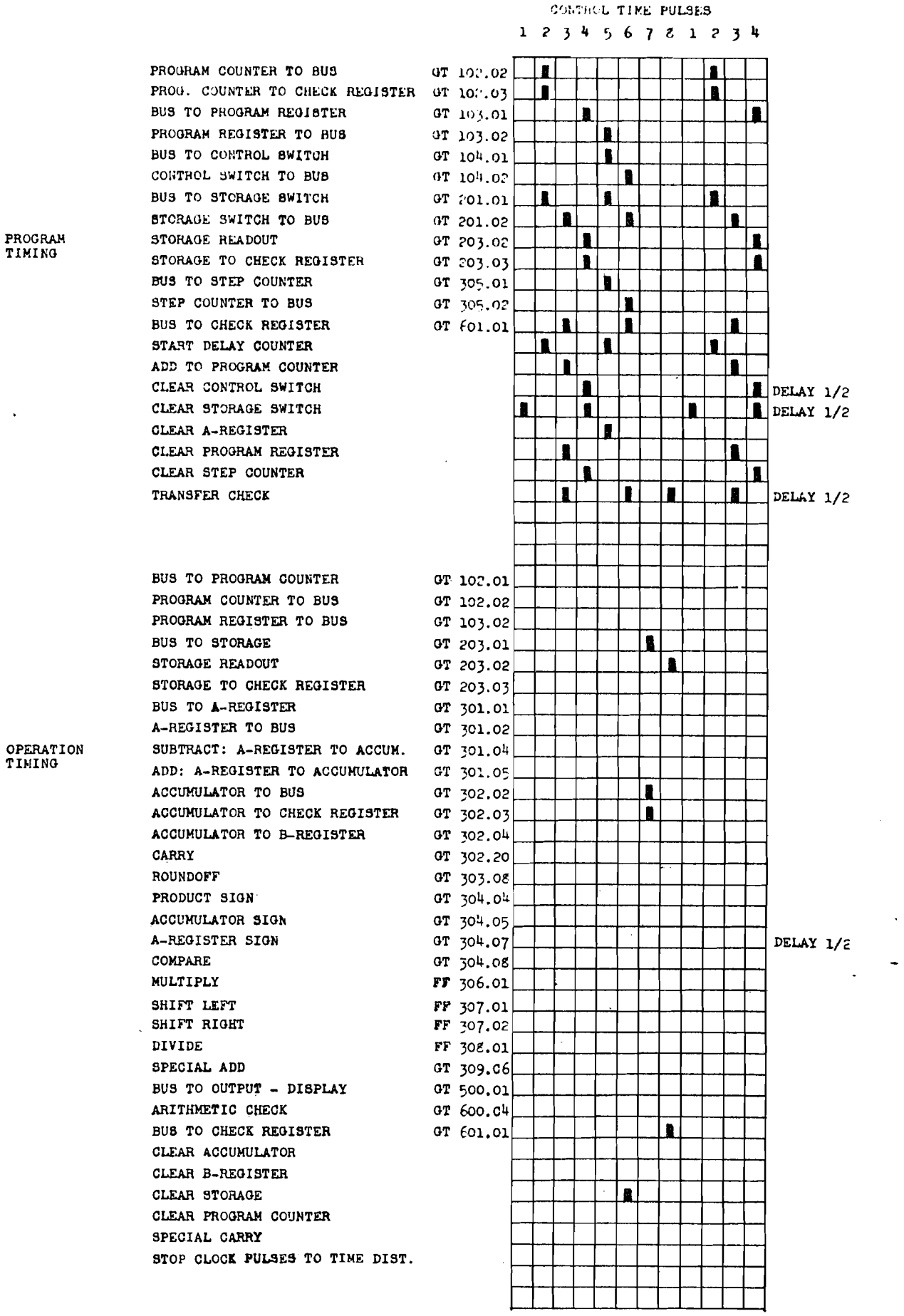


Figure 79

TIMING FOR TRANSFER TO STORAGE

OPERATION: SHIFT RIGHT sr

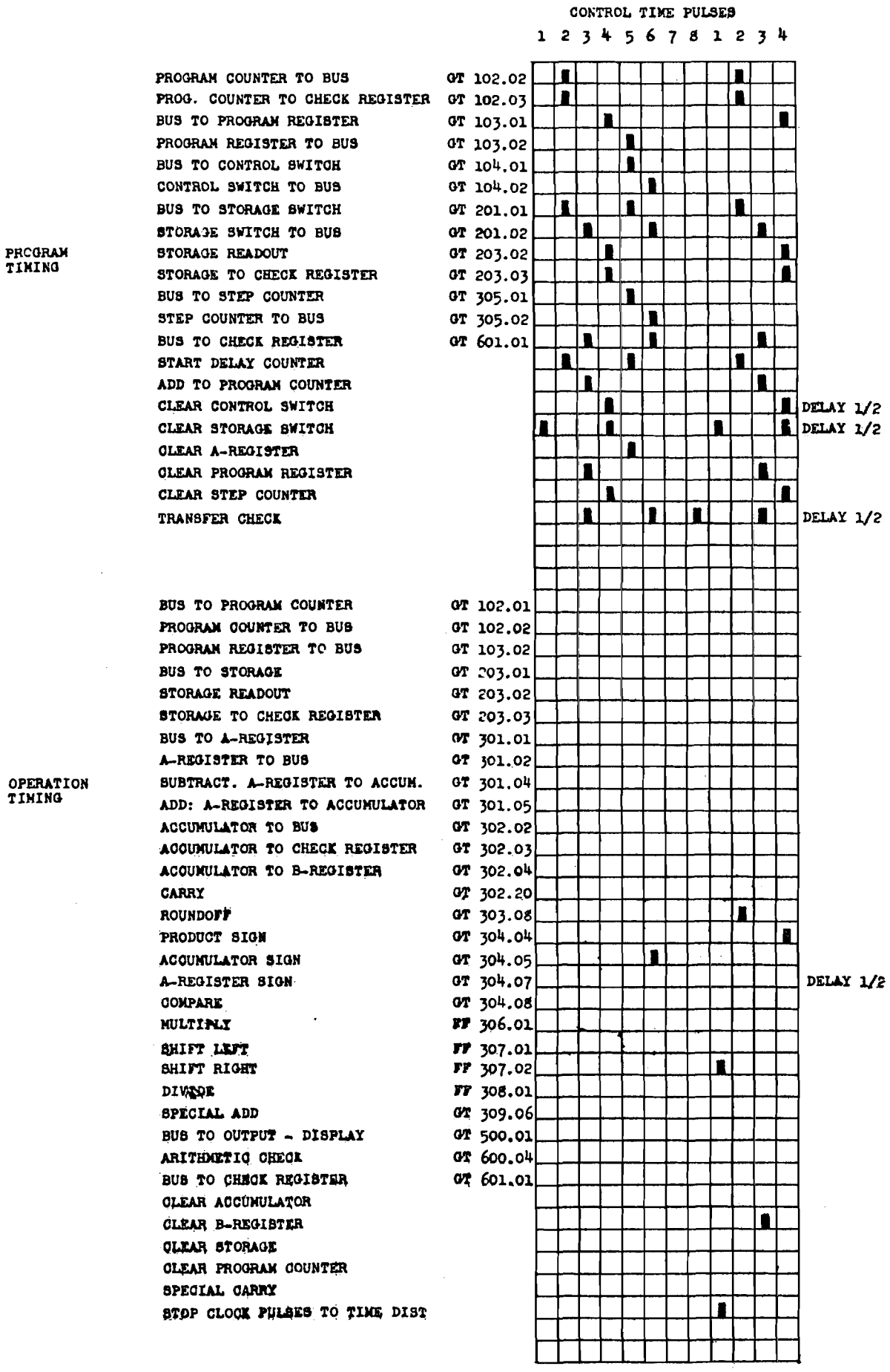


Figure 80

TIMING FOR SHIFT RIGHT

OPERATION: SHIFT LEFT s1

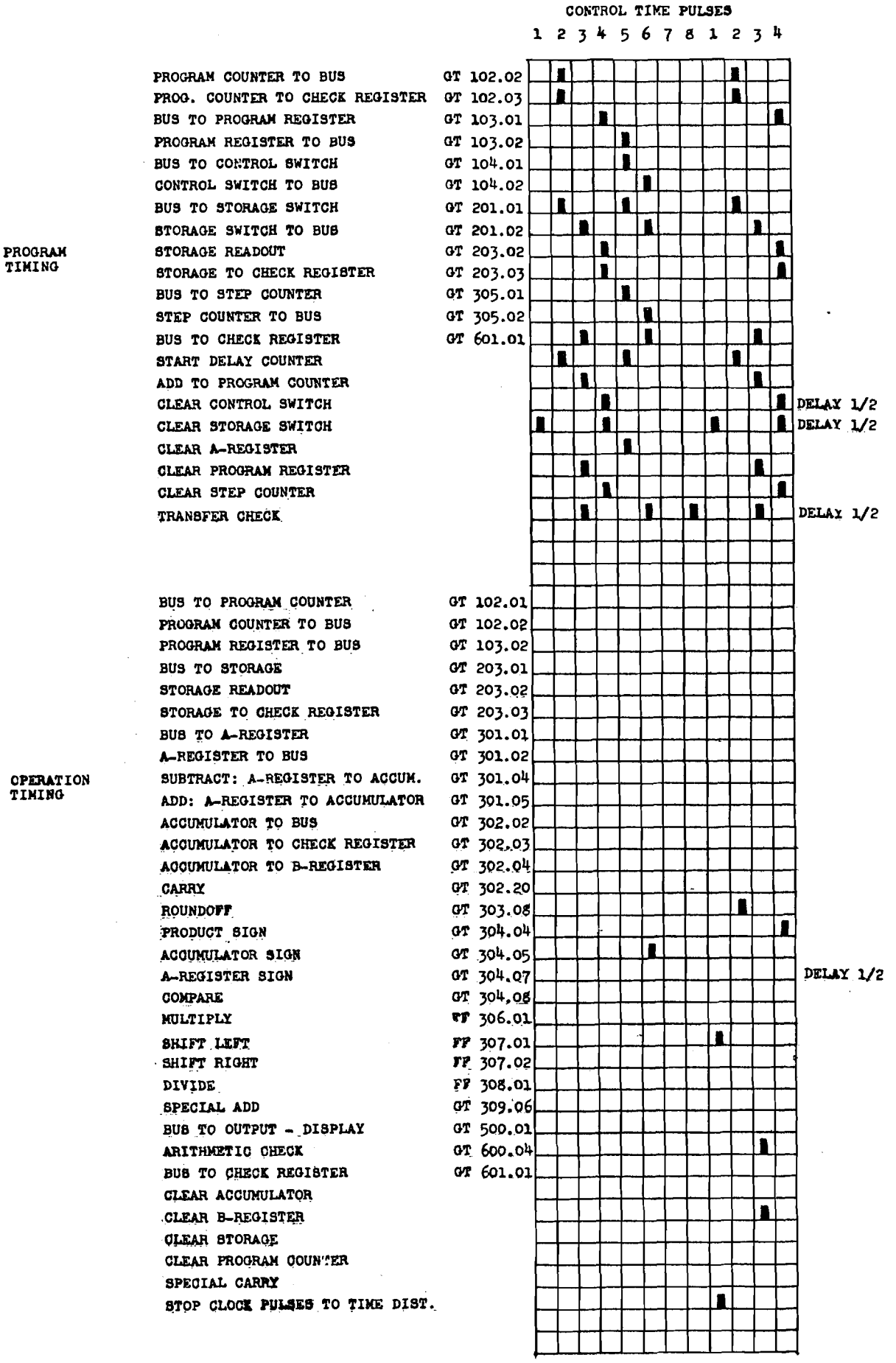


Figure 81

TIMING FOR SHIFT LEFT

OPERATION: SUBPROGRAM sp

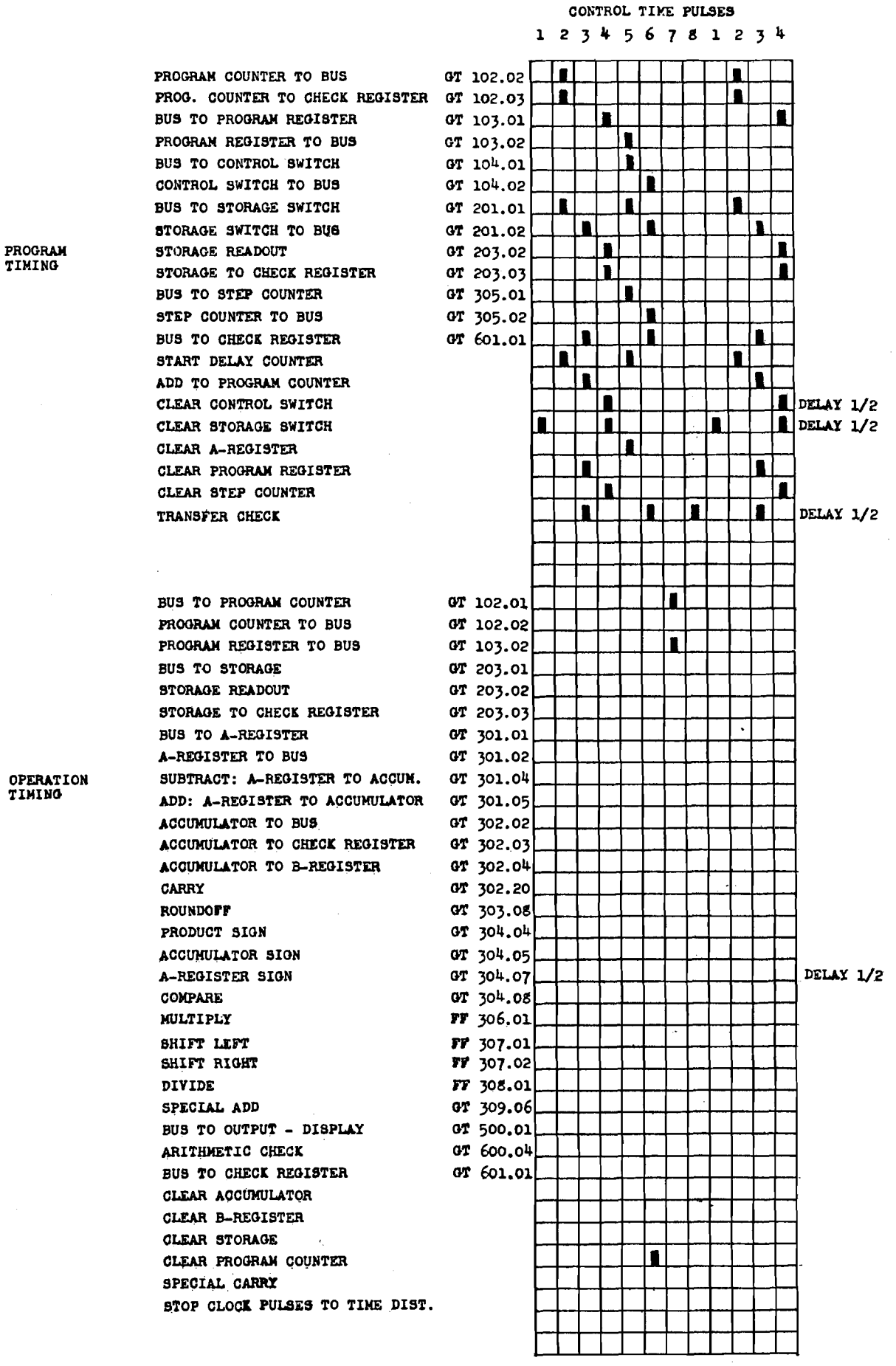


Figure 82

TIMING FOR SUBPROGRAM

OPERATION: COMPARE op

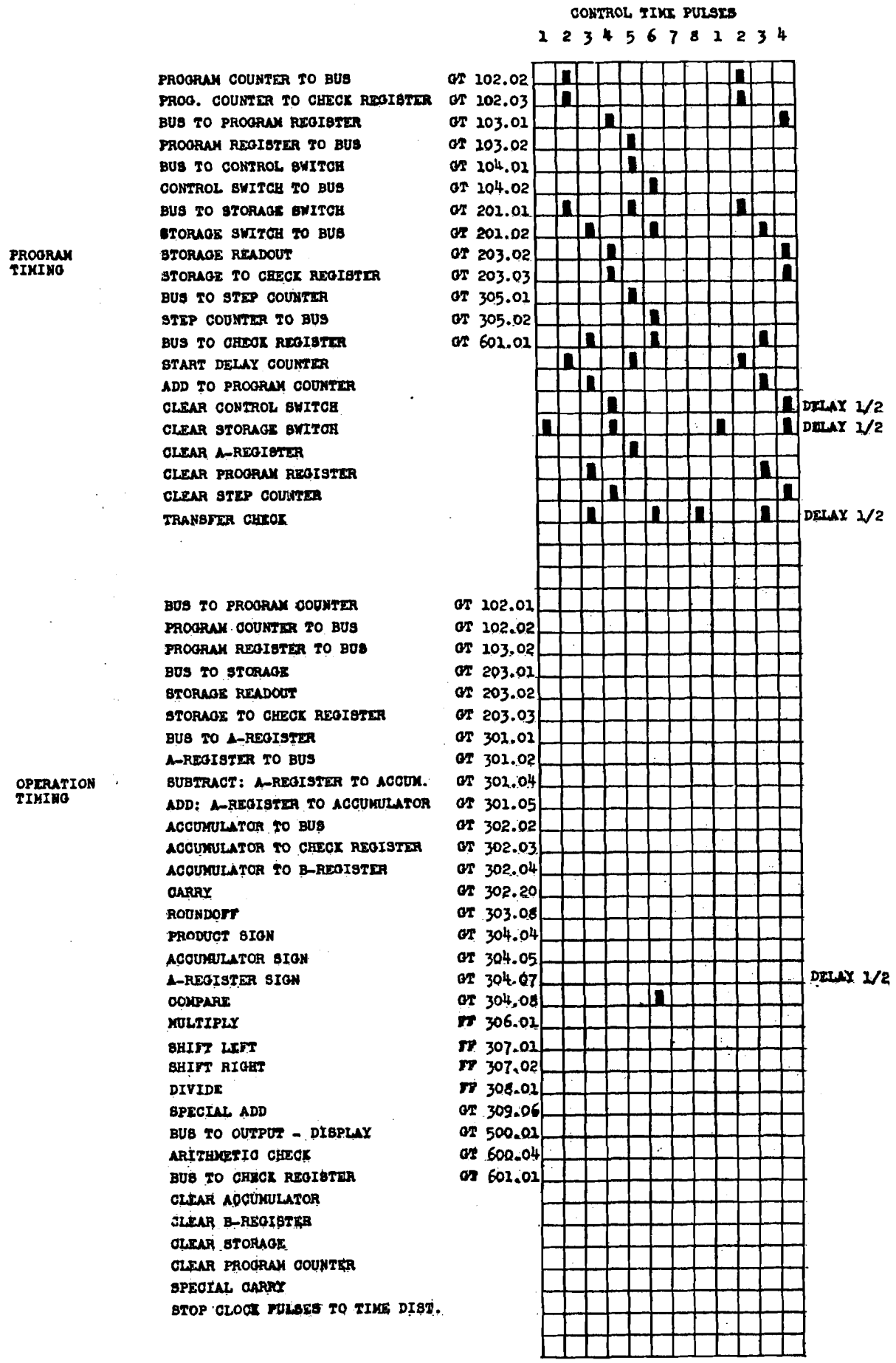


Figure 83

TIMING FOR CONDITIONAL PROGRAM

OPERATION: TRANSFER DIGITS td

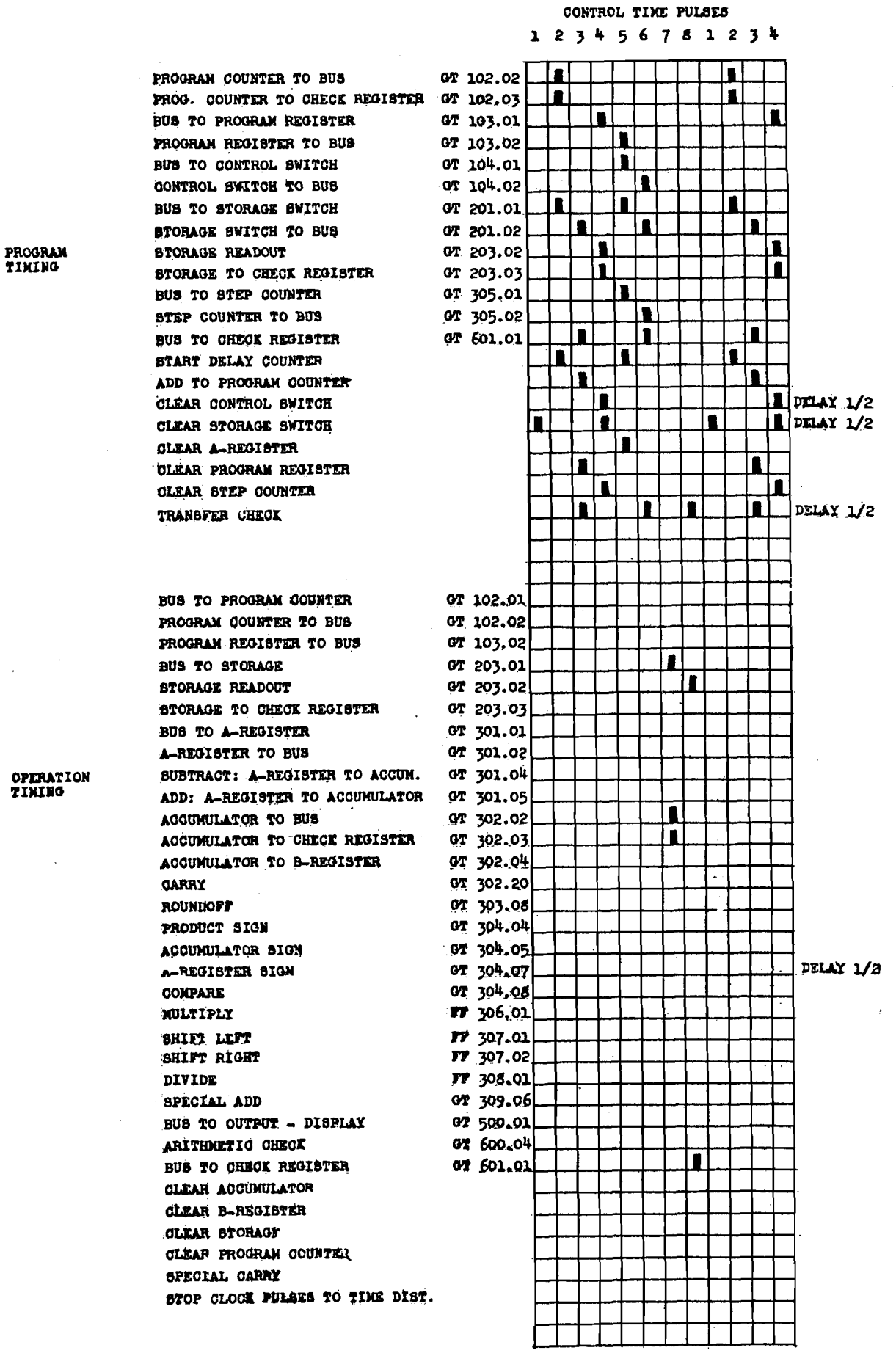


Figure 84

TIMING FOR TRANSFER DIGITS

OPERATION: SPECIAL ADD sa

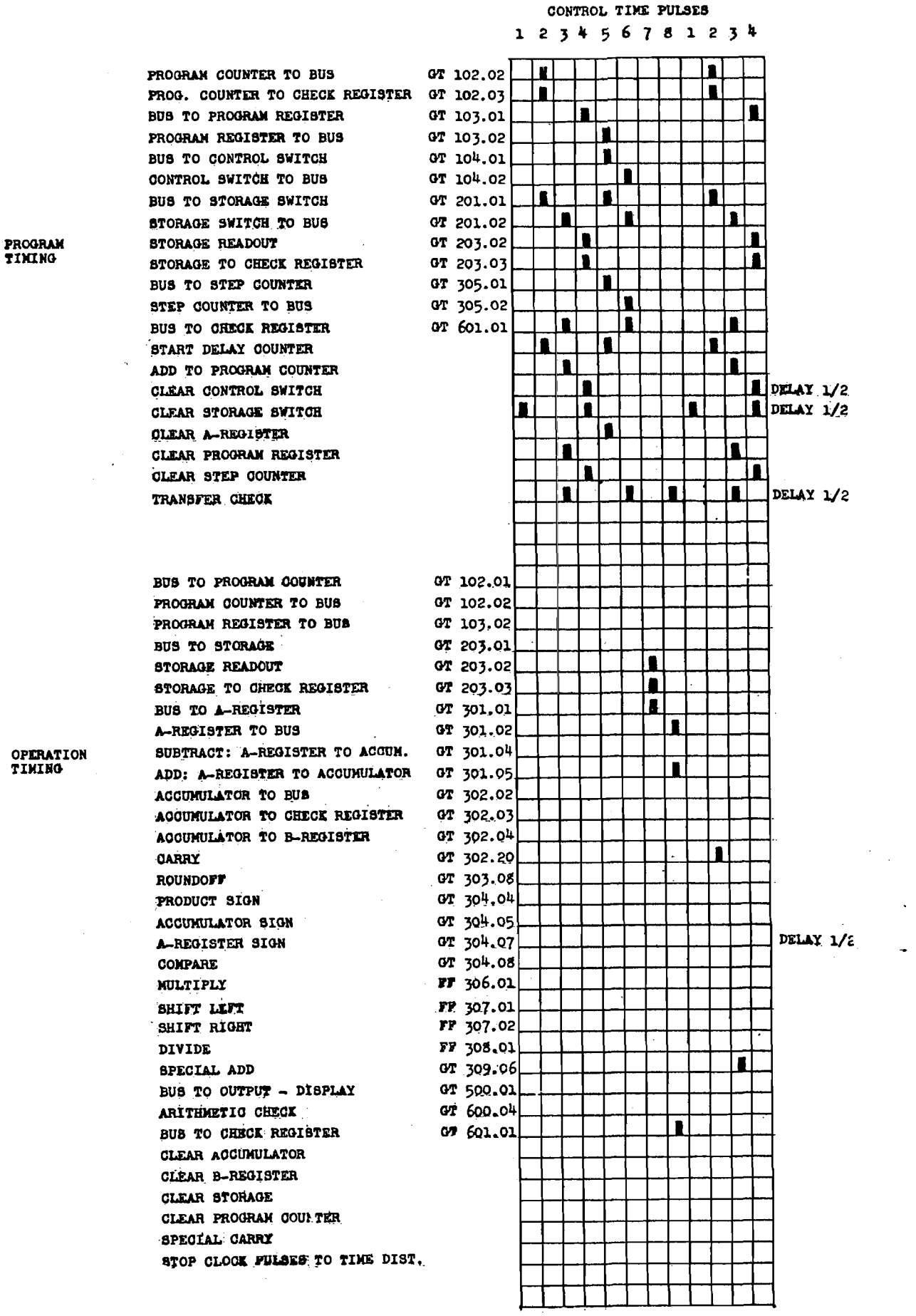


Figure 85

TIMING FOR SPECIAL ADD

OPERATION: STORE AND DISPLAY sd

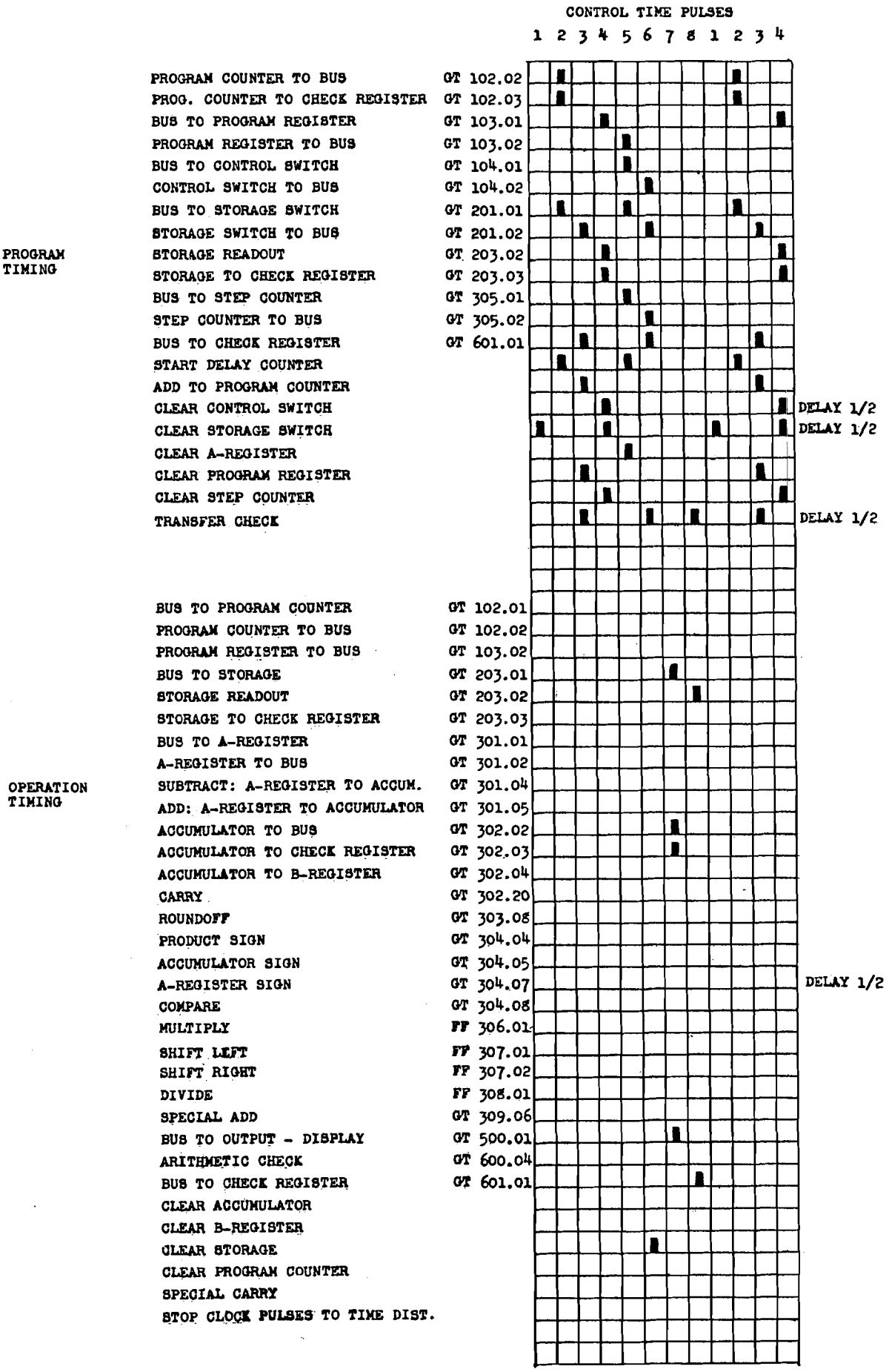


Figure 86

TIMING FOR STORE AND DISPLAY

UNIT CODE NUMBERS

100 Control

- 101 Master Clock
- 102 Program Counter
- 103 Program Register
- 104 Control Switch
- 105 Operation Matrix
- 106 Time Pulse Distributor
- 107 Operation Timing Matrix
- 108 Program Timing Matrix
- 109 Repeat Switch - (Removed from System)

200 Storage

- 201 Switch
- 202 Toggle Switch Storage
- 203 Flip Flop Stroage

300 Arithmetic Element

- 301 A-Register
- 302 Accumulator
- 303 B-Register
- 304 Sign Control
- 305 Step Counter
- 306 Multiply
- 307 Shift
- 308 Divide

400 Input

500 Output

600 Checking

- 601 Check Register

LETTER SYMBOLS

| | |
|---|---|
| AC | Accumulator |
| BA | Buffer Amplifier |
| BC | Binary Coder |
| BD | Bus Driver |
| DE | Delay Element |
| ES | Electronic Switch |
| FD | Frequency Divider |
| FF | Flip Flop |
| GT | Gate Tube Circuit |
| IC | Intensity Control Circuit |
| PF | Pulse Forming Circuit |
| ST | Storage Tube |
| TG | Trigger Pulse Generator |
| TP | Timing Pulse Generator |
| TT | Trigger Tube |

DRAWING SYMBOLS

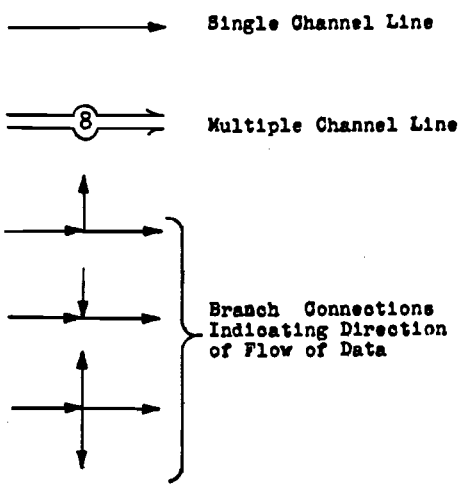


Figure 87

PARALLEL DIGIT COMPUTER CODES