AB23

Manual Q.151

FP 6000 COMPUTER SYSTEM

PRELIMINARY TECHNICAL DATA

FERRANTI ELECTRONICS
A Division of Ferranti-Packard Electric Limited
Industry Street
TORONTO 15
Ontario, Canada

PROPRIETARY RIGHTS

PRINTED IN CANADA

FUNCTION CODE SUMMARY

ADDITION AND SUBTRACTION

000	n + c → x	010	x + c → n	100	N + c → x			
001	x + n + c → x	011	n + x + c → n	101	x + N + c → x			Overflow may be set on exit but Carry is cleared
002	- n - c → x	012	- x - c → n	102	- N - c → x			
003	x - n - c → x	013	n - x - c → n	103	x - N - c → x			
004	n + c → x	014	x + c → n	104	N + c → x			
005	x + n + c → x	015	n + x + c → n	105	x + N + c → x			Overflow cannot be set. Sign of result always positive. Carry set if appropriate.
006	- n - c → x	016	- x - c → n	106	- N - c → x			
007	x - n - c → x	017	n - x - c → n	107	x - N - c → x			

LOGICAL OPERATIONS

020	x & n → x	030	n & x → n	120	x & N → x			Logical AND
021	x ∨ n → x	031	n ∨ x → n	121	x ∨ N → x			Logical INCLUSIVE OR
022	x ≠ n → x	032	n ≠ x → n	122	x ≠ N → x			Logical EXCLUSIVE OR

MISCELLANEOUS

023	Obey n as an instruction		033	0 → n	Clear n
024	n_j → x	Extract character	034	x_j → n_j	Insert character
025	n_e → x	Extract exponent	035	x_e → n_e	Insert exponent
026	Set C if n ≠ x or c = 1	Test equality	036	x_s → n_s	Part word
027	Set C if n + c > x	Test relative size	037	x_m → n_m	conversion

123	No operation	Dummy instruction
124	N → x_c, 0 → x_m	Set counter
125	Set Mode N	
126	Block Transfer	N words from address x to address x*
127	Check Sum → x	Sum N words from address x* ignoring overflow

MULTIPLICATION AND DIVISION

040	n.x → x:	Unrounded multiply
041	n.x - 2^{-24} → x:	Rounded multiply
042	n.x + x* → x:	Semi-cumulative multiply
043	10.x: + n_j → x:	Decimal-binary conversion
044	x:/n → x*	Unrounded double length division (remainder to X)
045	x:/n + 2^{-24} → x*	Rounded double length division (remainder to X)
046	x*/n → x*	Unrounded single length division (remainder to X)
047	10.x: → x, Char → n_j	Binary-decimal conversion

INDEXING AND BRANCH INSTRUCTIONS

050	Branch to N if x = 0	060	Single word modify	Add 1 to modifier
052	Branch to N if x ≠ 0	062	Alternate word modify	Add 2 to modifier
054	Branch to N if x > 0	064	Character modify	Add 1/4 to modifier
056	Branch to N if x < 0	In each case, subtract 1 from counter, branch if counter is non-zero.		

070	Subroutine entry	Store ONR and V in x, clear V, branch to N
072	Subroutine exit	Branch to N + x, V reset as at entry unless set by subroutine
074	Branch on Condition X:	
	X = 0 unconditionally	X = 4 if V clear, clear V
	X = 1 if V set	X = 5 if C set
	X = 2 if V set, clear V	X = 6 if C clear
	X = 3 if V clear	X = 7 if V clear, invert V

SHIFTING

110	Left Shift x,	N_s places, type N_t where:	N_t = 0 Cyclic Shift
111	Left Shift x:		N_t = 1 Logic Shift
112	Right Shift x,		N_t = 2 Arithmetic Shift
113	Right Shift x:		N_t = 3 Special Shift
114	Normalize x		
115	Normalize x:		

FLOATING POINT

130	Convert Fixed to Floating
131	Convert Floating to Fixed
132	x: + n: → x:
133	x: - n: → x:
134	x: . n: → x:
135	x: / n: → x:
136	√n: → x:

CONTROL AND INPUT/OUTPUT INSTRUCTIONS

150	Suspend me if my unit X of type N is busy
151	Release my unit X of type N
152	Disengage my unit X of type N
153	Place in my register 9 control registers of my unit X of type N
154	Read more program from my unit X of type N
155	Suspend and dump on my unit X of type N
156	Abollah this program
157	Suspend this program awaiting operator message to EXECUTIVE

160	Suspend this program awaiting operator's message to EXECUTIVE and print on the console typewriter any message from control word n.	170	Paper Tape Read	from unit X with control word n.
161	Spare	171	Paper Tape Punch	
162	Suspend me if my subprogram X is active.	172	Print one line	
163	Activate my subprogram X, entering at instruction N	173	Card read	
164	Suspend this subprogram awaiting activation by the master, and, if necessary release suspension on the master due to this subprogram.	174	Card punch	
		175	Magnetic tape operation	
		176	Drum operation	
		177	Peripheral Typewriter operation	

TIMING

The times shown are for unmodified instructions. Indexing increases these times by 6 or 2 usecs respectively.

	6 usecs Core	2 usecs Core
Groups 00, 01, 02 and 03 except function 023	18	7
Function 023	7	3
Function 040-041	67	40
042	72	41
043	58	27
044	76	45
045	79	48
046	71	44
047	58	27
Groups 05 and 06	13	5
Function 070	15	8
072	12	4
074	7	3
Group 10	12	5
Functions 110 & 112	18 + N_s	6 + N_s
111 & 113	42 + N_s	15 + N_s
114	37 + Sh	15 + Sh
115	42 + Sh	16 + Sh
Function 120-122	12	5
123	7	3
124	13	5
125	8	4
126	32 + 12N	13 + 4N
127	32 + 7N	14 + 3N

LEGEND

Note: In general, all subscripts apply to x or n.

N is a core store address or a number.

X is an accumulator (registers 0 - 7).

M is a modifier register (registers 1 - 3).

n, x, m refer to the contents of N, X, M respectively.

n + x → n means that the result of adding n and x replaces n, only the quantity following the arrow has been altered.

C is the carry digit register; c is used in orders 000-017 and 100-107. C is always left clear by any order unless that order sets C.

V is the arithmetic overflow indicator. V remains set until cleared.

ONR is the Order Number Register.

x* refers to the contents of X + 1

x: indicates a double length number

s is the sign bit or the most significant bit

N_t These comprise the most significant 2 bits and least
N_s significant 10 bits respectively, of the 12-bit address

n_j is any one of the four 6-bit characters of n.

x_e (the signed floating point exponent) is the least significant 9 bits of the contents of X + 1.

x_s is the least significant 12 bits of x (the N address).

NORMAL ORDERS

X	F	M	N
3	7	2	12

BRANCH ORDERS

X	F	N
3	6	15

DEFINITION OF TERMS

1. Executive Program

This is a program, retained in the core store, which is automatically entered whenever required for the purpose of:-

(a) Initiating a peripheral transfer.

(b) Control of program time-sharing.

(c) Monitoring abnormal or illegal operations.

(d) Control of special subroutines.

Communication between the programmer and the Executive program is carried out via the console typewriter. Executive has four special instructions, not available to normal programs, which are used to communicate with peripheral controls and for entering normal programs. FP6000 could operate as a single program machine without having an Executive but Executive is required if program time-sharing is to be carried out.

2. Datum and Limit Addresses

Each operating program is assigned a core store area on input. The Datum and Limit addresses specify the program area such that

$$D \leqslant \text{PROGRAM AREA ADDRESSES} < L$$

Thus, Limit is outside the program area.

A constraint on the choice of D and L is that they must each be a multiple of 64 and therefore the minimum area that can be occupied by an operating program is 64 words.

When a program (other than Executive) is running, each N address is automatically checked against D and L. Note that X and M do not require checking since they always lie in $D \leqslant X,M < D+8$.

3. Monitor Mode

Under control of the operator, Executive may enter a program in one of four modes as follows:-

(a) No monitoring - run

(b) Do one instruction, then interrupt

(c) Interrupt at the next 123 order

(d) Interrupt at the next 123 order or successful jump (and do not jump)

4. Peripheral Incident

A Peripheral Incident is any event in a peripheral which requires action by the Executive program, for example:-

(a) End of a transfer

(b) Parity failure

(c) Data overdue

Peripheral incidents cause interrupts.

5. Interrupt

An Interrupt is an automatic jump to a fixed location (16) in the Executive program. Interrupts occur only on one or more of the following events:-

(a) Illegal operation (D and L failure, unassigned order).

(b) Monitor mode.

(c) Timer (optional)

(d) Peripheral incidents

6. Group 13 - 17 Orders

These are orders which can be used to call in subroutines, such as initiating peripheral transfers, which are under the control of the Executive program. The actions of these instructions are all identical:

(a) Store instruction in Executive program's working area

(b) Store Order Number

(c) Enter Executive program at a fixed location(32)

7. Hesitation

A Hesitation is a computer operation which transfers data between (ie. either to or from) the core store and a peripheral control unit. A hesitation can occur between any two instructions or in some cases in the middle of an instruction.

8. Hesitation Request

Hesitation Request is a signal generated by the peripheral control and passed on to the computer by the Hesitation Control requesting a data transfer to or from that peripheral at the first opportunity.

9. Order Number

The Order Number is the address of the instruction which is presently being obeyed. At the end of each instruction it is incremented by 1 and the next instruction in sequence is obeyed (unless the present instruction is a successful transfer). When a program is running, its order number is held in one of the working registers in the Arithmetic Unit. When a program is not running, its Order Number is stored in Datum + 8, and is retrieved from there by the Executive Program whenever it desires to activate that program.

10. Control Word

Each peripheral has a Control Word in core store in a fixed location which can be addressed by Hesitation Control. Before a transfer is initiated, the Executive program sets up the control word to contain:

(a) The core store address of the first data transfer

(b) The count of the number of words or characters to be transferred

This word is read out of store during a hesitation and is used by the computer to carry out the transfer.

Preliminary Description of

FP6000 Logic

Definition of Terms
Function Code Summary

FERRANTI ELECTRONICS
A Division of Ferranti-Packard Electric Limited
Industry Street
TORONTO 15
Ontario, Canada

FUNCTION CODE SUMMARY

ADDITION AND SUBTRACTION

000	$n + c \rightarrow x$	010	$x + c \rightarrow n$	100	$N + c \rightarrow x$	Overflow may be set on exit but Carry is cleared
001	$x + n + c \rightarrow x$	011	$n + x + c \rightarrow n$	101	$x + N + c \rightarrow x$	
002	$-n - c \rightarrow x$	012	$-x - c \rightarrow n$	102	$-N - c \rightarrow x$	
003	$x - n - c \rightarrow x$	013	$n - x - c \rightarrow n$	103	$x - N - c \rightarrow x$	
004	$n + c \rightarrow x$	014	$x + c \rightarrow n$	104	$N + c \rightarrow x$	Overflow cannot be set. Sign of result always positive. Carry set if appropriate.
005	$x + n + c \rightarrow x$	015	$n + x + c \rightarrow n$	105	$x + N + c \rightarrow x$	
006	$-n - c \rightarrow x$	016	$-x - c \rightarrow n$	106	$-N - c \rightarrow x$	
007	$x - n - c \rightarrow x$	017	$n - x - c \rightarrow n$	107	$x - N - c \rightarrow x$	

LOGICAL OPERATIONS

020	$x \,\&\, n \rightarrow x$	030	$n \,\&\, x \rightarrow n$	120	$x \,\&\, N \rightarrow x$	Logical AND
021	$x \vee n \rightarrow x$	031	$n \vee x \rightarrow n$	121	$x \vee N \rightarrow x$	Logical INCLUSIVE OR
022	$x \not\equiv n \rightarrow x$	032	$n \not\equiv x \rightarrow n$	122	$x \not\equiv N \rightarrow x$	Logical EXCLUSIVE OR

MISCELLANEOUS

023	Obey n as an instruction		033	$0 \rightarrow n$	Clear n
024	$n_j \rightarrow x$	Extract character	034	$x_3 \rightarrow n_j$	Insert character
025	$n_e \rightarrow x$	Extract exponent	035	$x_e \rightarrow n_e$	Insert exponent
026	Set C if $n \neq x$ or $c = 1$	Test equality	036	$x_a \rightarrow n_a$	Part word
027	Set C if $n + c > x$	Test relative size	037	$x_m \rightarrow n_m$	conversion

123	No operation	Dummy instruction
124	$N \rightarrow x_c$, $0 \rightarrow x_m$	Set counter
125	Set Mode N	
126	Block Transfer	N words from address x to address x*
127	Check Sum $\rightarrow$ x	Sum N words from address x* ignoring overflow

MULTIPLICATION AND DIVISION

040	$n.x \rightarrow x:$	Unrounded multiply
041	$n.x + 2^{-24} \rightarrow x:$	Rounded multiply
042	$n.x + x^* \rightarrow x:$	Semi-cumulative multiply
043	$10.x: + n_j \rightarrow x:$	Decimal-binary conversion
044	$x:/n \rightarrow x^*$	Unrounded double length division (remainder to X)
045	$x:/n + 2^{-24} \rightarrow x^*$	Rounded double length division (remainder to X)
046	$x^*/n \rightarrow x^*$	Unrounded single length division (remainder to X)
047	$10.x: \rightarrow x:$ Char $\rightarrow n_j$	Binary-decimal conversion

INDEXING AND BRANCH INSTRUCTIONS

050	Branch to N if $x = 0$	060	Single word modify	Add 1 to modifier
052	Branch to N if $x \neq 0$	062	Alternate word modify	Add 2 to modifier
054	Branch to N if $x \geqslant 0$	064	Character modify	Add 1/4 to modifier
056	Branch to N if $x < 0$	In each case, subtract 1 from counter, branch if counter is non-zero.		

070	Subroutine entry	Store ONR and V in x, clear V, branch to N
072	Subroutine exit	Branch to N + x, V reset as at entry unless set by subroutine
074	Branch on Condition X:	
	X = 0 unconditionally	X = 4 if V clear, clear V
	X = 1 if V set	X = 5 if C set
	X = 2 if V set, clear V	X = 6 if C clear
	X = 3 if V clear	X = 7 if V clear, invert V

SHIFTING

110	Left Shift x,	N_s places, type N_t where:	$N_t = 0$ Cyclic Shift
111	Left Shift x:		$N_t = 1$ Logic Shift
112	Right Shift x,		$N_t = 2$ Arithmetic Shift
113	Right Shift x: ,		$N_t = 3$ Special Shift
114	Normalize x		
115	Normalize x:		

FLOATING POINT

130	Convert Fixed to Floating
131	Convert Floating to Fixed
132	$x: + n: \rightarrow x:$
133	$x: - n: \rightarrow x:$
134	$x: . n: \rightarrow x:$
135	$x: / n: \rightarrow x:$
136	$\sqrt{n:} \rightarrow x:$

150	Suspend me if my unit X of type N is busy
151	Release my unit X of type N
152	Disengage my unit X of type N
153	Place in my register 9 control registers of my unit X of type N
154	Read more program from my unit X of type N
155	Suspend and dump on my unit X of type N
156	Abolish this program
157	Suspend this program awaiting operator message to EXECUTIVE

160	Suspend this program awaiting operator's message to EXECUTIVE and print on the console typewriter any message from control word n.
161	Spare
162	Suspend me if my subprogram X is active.
163	Activate my subprogram X, entering at instruction N
164	Suspend this subprogram awaiting activation by the master, and, if necessary release suspension on the master due to this subprogram.

170	Paper Tape Read
171	Paper Tape Punch
172	Print one line
173	Card read
174	Card punch
175	Magnetic tape operation
176	Drum operation
177	Peripheral Typewriter operation

from unit X with control word n.

TIMING

The times shown are for unmodified instructions. Indexing increases these times by 6 or 2 usecs respectively.

	6 usecs Core	2 usecs Core
Groups 00, 01, 02 and 03 except function 023	18	7
Function 023	7	3
Function 040-041	67	40
042	72	41
043	58	27
044	76	45
045	79	48
046	71	44
047	58	27
Groups 05 and 06	13	5
Function 070	15	8
072	12	4
074	7	3
Group 10	12	5
Functions 110 & 112	$18 + N_s$	$6 + N_s$
111 & 113	$42 + N_s$	$15 + N_s$
114	37 + Sh	15 + Sh
115	42 + Sh	16 + Sh
Function 120-122	12	5
123	7	3
124	13	5
125	8	4
126	32 + 12N	13 + 4N
127	32 + 7N	14 + 3N

LEGEND

Note: In general, all subscripts apply to x or n.

N is a core store address or a number.

X is an accumulator (registers 0 - 7).

M is a modifier register (registers 1 - 3).

n, x, m refer to the contents of N, X, M respectively.

$n + x \rightarrow n$ means that the result of adding n and x replaces n, only the quantity following the arrow has been altered.

C is the carry digit register; c is used in orders 000-017 and 100-107. C is always left clear by any order unless that order sets C.

V is the arithmetic overflow indicator. V remains set until cleared.

ONR is the Order Number Register.

x* refers to the contents of X + 1

x: indicates a double length number

s is the sign bit or the most significant bit

N_t These comprise the most significant 2 bits and least
N_s significant 10 bits respectively. of the 12-bit address

n_j is any one of the four 6-bit characters of n.

x_e (the signed floating point exponent) is the least significant 9 bits of the contents of X + 1.

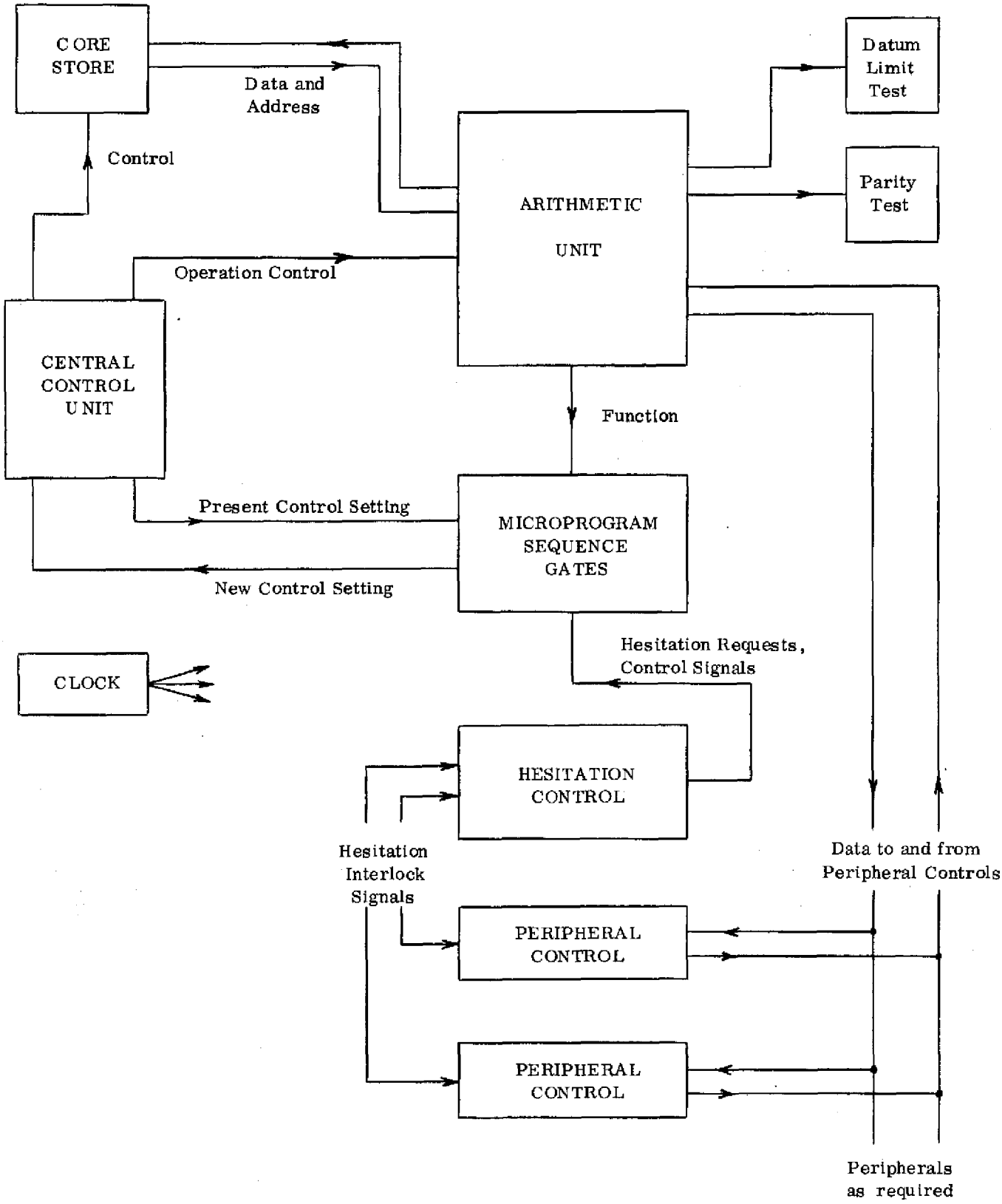
x_a is the least significant 12 bits of x (the N address).

NORMAL ORDERS

X	F	M	N
3	7	2	12

BRANCH ORDERS

X	F	N
3	6	15



FP6000 - SYSTEM BLOCK DIAGRAM

SECTION 1

CENTRAL COMPUTER

1.1 SYSTEM BLOCK DIAGRAM

This diagram shows the main sections of the central computer in block form together with the main control and data transfer paths. The interaction of these blocks is described below in general terms and the detailed operation of each block is covered later in a separate section.

The Arithmetic Unit contains three full-length working registers, several shorter registers (for staticizing the instruction, datum and limit), and a parallel adder which can carry out arithmetic and logical operations on the contents of these registers. One register can send data to or receive data from the core store, core store addresses can be sent from several other registers. Data can be sent to or received from peripheral control units. Many special facilities are provided within the arithmetic unit to achieve the efficient realization of the Instruction Code.

All operations and facilities in the Arithmetic unit are controlled by signals originating in the Central Control unit. Central Control consists mainly of a set of control flip-flops, each one of which in general, controls a specific sub-operation in the Arithmetic Unit. Usually, several of these flip-flops will be set to define and control a complete Arithmetic unit operation. Central control is also used to control core store operations and, on some occasions, the clock rate.

The Microprogram Gates are responsible for sequencing the control flip-flops in Central Control. The Microprogram is informed of the sequence required by either the Function register (in the Arithmetic unit) or by Hesitation control. The Microprogram then can be broken down into sub-units, each one of which is responsible for sequencing a particular order (or set of similar orders) or hesitation. The microprogram defines the sequence of actions necessary in the Arithmetic unit (via Central Control) to carry out each instruction in turn, as written by the programmer. Within each instruction (or group of similar instructions) or hesitation, the present setting of the Central Control flip-flops can be used by the microprogram gates to determine what has to be done next. When a sequence contains two or more identical operations, additional flip-flops (in Central Control) or counters (in Arithmetic Unit) are available to allow the microprogram to make a decision.

Hesitation Control provides an interlock between all peripheral controls so that the computer can deal with them one at a time if several request data transfers at the same time. Hesitation Control also informs the microprogram regarding the type of peripheral being dealt with (ie. input, output, word-at-a-time, character-at-a-time) so that the Hesitation sequencing can be defined. Hesitation control also generates a fixed core store address for each peripheral, this is known as Control Word Address.

The Clock produces timed pulses which set control and data flip-flops in all parts of the machine. The clock normally runs at a fixed rate (1 Mc/s) but can be slowed down for special operations (such as core store cycles) by the central control. Considerable manual control of the clock is also available for maintenance procedure.

1.2 DESCRIPTION OF SEQUENCING

The entire sequencing is summarised on Drawing E.13322 which should be examined while reading the following.

1.2.1 Overall Sequencing Diagram

Many blocks of sequences (shown in square brackets on the sequence diagram) are available to the microprogram. Once one of these blocks is entered, it is always continued to the end. However, in between blocks, the microprogram must decide which new sequence block to enter. These decisions are described on the Overall Sequence diagram.

Considering the top of the diagram as a starting point, the first decision to be made is whether or not to enter the Hesitation sequence. If there is a Hesitation Request (HR) signal from the Hesitation control, this sequence is entered and provides data to a peripheral. If there is no Hesitation Request signal, the next consideration is whether or not there is any reason to Interrupt, ie. Enter Executive program. This decision is based on an OR signal from all possible reasons for Interrupt (see Definitions). Of course, if the machine is in Executive already, no Interrupt can occur.

1.2.1 continued

If neither Interrupt nor Hesitation Request signals are present, the next sequence to be entered is "Preparation of Order". Thus, the computer continues with the current operating program. Preparation of Order reads the next instruction in the program from the store and sets it up in the Instruction registers. When this step is finished, the microprogram examines the Function register and, if this particular instruction requires three working registers, the Order Number is written away into Core Store so that the Order Number Register can be used during the instruction.

1.2.2 Hesitation Sequence Selection

Clearly, different types of peripherals have different data-transfer rates and the maximum permissible waiting-time, between requesting data and actually receiving data, will vary according to the type. For convenience, peripherals are divided into two groups, according to waiting-time, and the relevant Hesitation Requests are approximately designated:-

Fast ($\leq$ < 100 microseconds)

Slow (> 100 microseconds)

Hesitation Requests may be initiated at any random time and may therefore occur immediately after a sequence block is entered. Since any such sequence must be completed, it follows that its duration must be less than the relevant maximum waiting-time if the peripheral's request for data is not to become "Overdue". To accomplish this, the sequencing is arranged as described below and the Orders are divided into three groups, namely; Short, Long and Very Long.

Short orders are those that can be completed quickly enough to accommodate Fast peripherals. Long orders can accommodate Slow peripherals but not Fast peripherals and Very Long orders are those which are too lengthy for even the Slow peripherals. This is summarised as follows:-

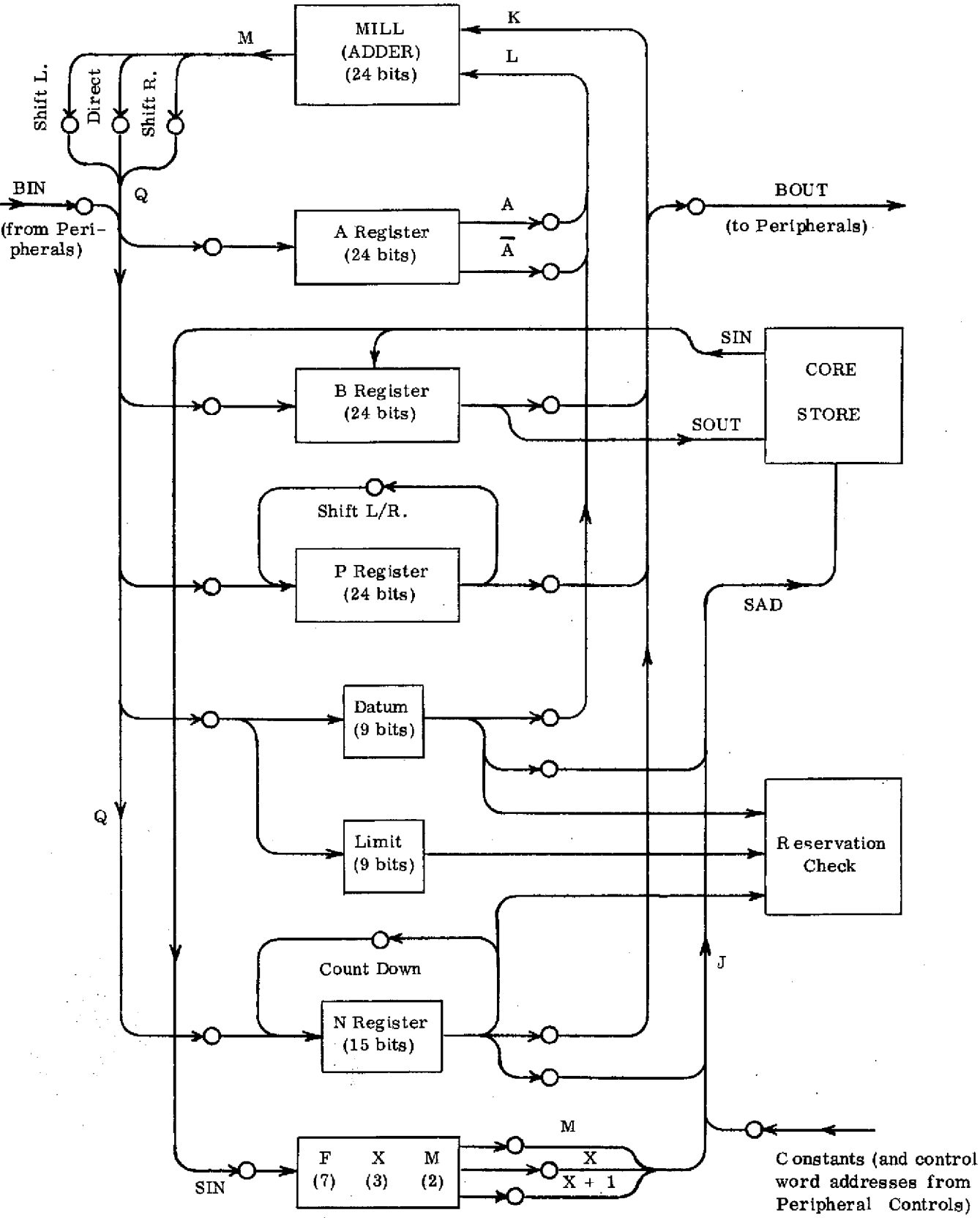
Orders	Peripherals	
	Fast	Slow
Short	✓	✓
Long	X	✓
Very Long	X	X

Both the Long and Very Long order sequences are organised so that the sequence may be broken at one or more places to permit the insertion of a Hesitation sequence. No such provision is required for Short Orders.

Thus, if a Fast Hesitation Request (FHR) occurs immediately after a Long order sequence is entered, the operation continues to a convenient break-point and prepares to enter the Hesitation sequence. Since the Hesitation sequence requires the use of some of the working registers of the Arithmetic Unit, the data in these registers must be stored before entering the Hesitation sequence and must be retrieved after the Hesitation ready for re-entering the order sequence. This sequence is determined by the "Intra-Order Hesitation" (IOH) flip-flop which is set when a Long order is left to enter a hesitation and is reset when the order is re-entered. If two Fast Hesitation requests are to be serviced, the second is also dealt with before the order is re-entered.

Very Long Orders can be left in the middle in the same manner as described above, and the computer will do so if any "Hesitation Request" is present at that time.

At the end of the order, or of the Interrupt sequence, or of a hesitation (other than an IOH) the sequence arrives back at the original starting point and the whole process is repeated.



ARITHMETIC UNIT

SECTION 2

ARITHMETIC UNIT

2.1 GENERAL DESCRIPTION

The main functions of the Arithmetic Unit are as follows:-

(a) Storage of data while in use.

(b) Storage of the instruction while in use.

(c) Carrying out arithmetic or logical operations.

(d) Storage of location register.

(e) Storage of data and addresses for the core store.

The arithmetic unit block diagram shows the necessary sub-units. Registers A and B are 24 bit flip-flop registers used mainly for holding data during arithmetic or logical operations. Register P is a 24 bit flip-flop register used primarily for holding the program location. If required, the program location can be stored and P can also be used as a working register. The N register is a 15 bit flip-flop register which is used mainly for holding the N address of the instruction currently being obeyed. The D and LIM registers are both 9 bit flip-flop registers and are used for holding the datum and limit addresses of the program which is currently being processed. The F, X and M registers hold the corresponding part of the instruction presently being obeyed. The K and L highways are 24 bit parallel data transmission paths which form the two inputs to the Mill. The Mill processes the 24 bits in parallel and can produce on the Q highway the sum, the AND function or the EQUIVALENT function of the data presented to it on the K and L highways. The result on the Q highway can then be routed into one or more of the working registers as shown on the Block Diagram. All of these arithmetic unit operations are controlled by signals originating in Central Control.

Addresses for the core store are held in the D, N, X and M registers and the contents of these registers can be routed to the core store address highway (SAD). Data from the core store can be read into the B, F, X and M registers but data to be written into the core store always originates in the B register. Again these operations are controlled from Central Control.

Many special facilities required in the arithmetic unit, such as partial routing of registers to highways, internal shifting of registers, and internal counting of registers, are also provided and these are described in the detailed analysis of the sub units which follows.

The logic drawings which show the arithmetic unit are numbered from 0 to 30 in the lower right hand corner. Drawings 0 to 23 each show one bit of the A, B and P registers, L and K highways, Mill and Q highway. All of these drawings are therefore very similar and a detailed description of each is not necessary. Drawings 25 to 30 each contain one sub unit and are described in detail under the sub unit heading. Drawing 24 forms an exception and is reserved for other use.

2.2 HIGHWAYS

The term Highway refers to important data or address transmission wires and is used in preference to the term Busbar to imply parallel transmission of data. Within the arithmetic unit the main highways are K, L, M and Q.

The K highway is one of the data inputs to the Mill. It is a 24 bit parallel highway onto which data can be routed from the B, P, or N registers. One bit of the K highway is shown on each of the logic drawings from 0 to 23.

The L highway forms the other data input to the Mill. Register A (either normal or inverted) and register D can be routed to the L highway. One bit of the L highway is shown on each of drawings 0 to 23.

The M highway is the output of the adder part of the Mill and also appears on drawings 0 to 23.

Highway Q is the final Mill output and also the input of registers A, B, P, D, LIM and N. The information on the Q highway may therefore be inserted into any of these registers by applying a strobe pulse to the desired flip-flops. One bit of the Q highway is shown on each of drawings 0 to 23.

2.2 continued

Other highways which enter or originate in the arithmetic unit are SAD, SIN, SOUT, BIN and BOUT. The S prefix designates core store highways and the B prefix designates peripheral highways.

For the sake of brevity, the term SAD highway is reduced to J when applied to waveform names. Thus, both the terms SAD and J refer to the same highway but the more descriptive term SAD is generally used in written text.

The SAD highway is a 15 bit core store address highway and appears on drawings 26, 27, and 30. Addresses can be routed to SAD from registers N, D (most significant 9 bits), X (least significant 3 bits), X + 1 (least significant 3 bits), and M (least significant 2 bits).

The fact that the routing of D and any one of X, X + 1 or M do not overlap on the SAD highway is used to avoid the necessity of having to add D and X, for example, before sending the address to the store. The addition effectively takes place on the SAD highway since the datum address must always be a multiple of 64 and always has its least significant 6 bits equal to zero. The address X + 1 is provided by an adder on one output of the X register which also can be routed to the SAD highway and which will be described in more detail in the section on the F, X, and M registers. The SIN and SOUT highways carry data from and to the core store. Data on the SIN highway occurs as a pulse during read operations in the core store. This pulse is used to set information onto the B register flip-flops on their collectors. The SOUT highway presents the information in the B register to the core store for use during write operations.

The BIN and BOUT highways carry data from and to peripheral controls during hesitations. Data arriving on BIN is gated onto the Q highway at the proper time and from there into the B register to be written away in core store. Similarly, data is read from the core store into the B register, routed onto the K highway, and from there routed onto the BOUT highway and sent to a peripheral control during an output peripheral transfer.

2.3 A REGISTER

The A Register consists of 24 single entry flip-flops, one of which is shown on each of Logic Diagrams 0 to 23. The input of each bit of the A register is directly connected to the corresponding bit of the Q highway, thus the clock signal XQA, transfers data from the Q highway into the A register. This clock waveform is gated by signals from Central Control (see drawing S).

The output of each bit of the A register goes to two Nor gates each of which has as its output the corresponding bit of the L highway. The gating waveforms -CAL and -CIAL can therefore be used to route the normal or inverted contents of the A register onto the L highway. The three waveforms +MSP6, +MSP7 and +MSP8, which appear on some of these gates on drawings 0 to 23, are all normally negative and therefore do not effect the gating of the A register to the L highway.

It will be noted that if -CAL and -CIAL are negative simultaneously, that all bits of the L highway will be positive regardless of the contents of the A register. This property is utilised in providing constants on the L highway whenever they are required. The waveforms +MSP6, +MSP7 and +MSP8 are used to provide the required constant by keeping some bits of the L highway negative. The three constants required and the waveform configurations which provide them are shown in the table below.

Note that the L highway is + for 1 and is - for 0.

TABLE 1

CONSTANT									+MSP6	+MSP7	+MSP8
1)	111	111	111	000	000	000	000	000	-	+	+
2)	111	111	111	000	000	000	000	010	-	-	+
3)	001	111	111	000	000	000	000	000	+	-	+

2.4 B REGISTER

The B Register consists of 24 single entry flip-flops, one of which is shown on each of drawings 0 to 23. The input to each B register flip-flop is the corresponding bit of the Q highway. The clock input to the B register is broken up into six separate waveforms to provide the facility of inserting a section of the Q highway into the corresponding section of the B register without disturbing the remainder of the B register. These sections and waveforms are as follows:-

XQB0	Bits 0 to 5
XQB1	Bits 6 to 8
XQB2	Bits 9 to 11
XQB3	Bits 12 to 14
XQB4	Bits 15 to 17
XQB5	Bits 18 to 23

These gated clock signals can be used separately or in combination by Central Control. They are generated on drawing S and are shown on drawings 0 to 23 in the proper bit positions.

The B register may be routed onto the K highway by a Nor gate on the zero output of each bit of the B register (for example, 1D10/3 on drawing 0). At times, it is required to route only a part of the B register onto the K highway and the required facility is provided by two gating waveforms -CBK6 and -CBK18 and two inhibiting waveforms +MSP1 and +MSP23.

Table 2 shows the bit positions in which each of these waveforms is used and Table 3 shows the routing function and the waveform configurations required to produce them.

TABLE 2

Waveform	Used on Bits
-CBK6	0 - 5
-CBK18	6 - 23
+MSP1	12 - 14
+MSP23	15 - 23

The facility is also provided of routing each character in B (each of the four 6-bit blocks) onto the least significant character position of the K highway (K0 - K5). Three gates on each of drawings 0 to 5 are shown (for example, 1F28/1, 2F29/1 and 2E11/1 on drawing 0) with the main gating waveform -CBCHK and the character selection gating waveforms -CH0, -CH1 and -CH2. The selection of character 3 is determined in Central Control by using the waveform -CBK6.

The zero output of each B register flip-flop is connected to a Nor gate whose output is the data highway to the core store (for example, 1D10/4 on drawing 0). This waveform is not gated and therefore the contents of the B register are continually supplied to the core store and can be written away in the core store whenever Central Control orders a write operation.

Data may be entered in the B register from either the core store or the Engineer's Handswitches, as follows. First, the zero output of each B register flip-flop is connected via a diode (for example, 1D10/B on drawing 0) to the line carrying signal +CBC. Note that this is actually divided into three waveforms (+CBC1, +CBC2 and +CBC3) to satisfy the loading requirements of the Pulse Gate; each Pulse Gate is only capable of driving 10 flip-flops.

TABLE 3

ROUTING	-CBK6	-CBK18	+MSP1	+MSP23
B0 - B5 → K0 - K5	-	+	-	-
B0 - B11 → K0 - K11	-	-	+	+
B0 - B14 → K0 - K14	-	-	-	+
B0 - B23 → K0 - K23	-	-	-	-

2.4 continued

These waveforms produce short pulses (approximately 0.3 usecs) which are applied to the B register at the beginning of each core store read cycle to clear the B register. Second, two Nor gates drive the one output of each B register flip-flop (for example, 1D10/2 and 1D10/5 on drawing 0). These two gates apply a positive pulse to set data into the B register from either the core store or the Handswitches as determined by the control waveforms -CSB and -CHSB respectively.

2.5 P REGISTER

The P register consists of 24 triple entry flip-flops, one of which is shown on each of drawings 0 to 23. The #2 input to each P register flip-flop is the corresponding bit of the Q highway, thus the #2 clock input inserts data from the Q highway into the P register. The P register is divided into groups in a manner similar to the B register, ie. as shown below in Table 4.

TABLE 4

Bits	Clock Waveforms
Q0 - Q14 → P0 - P14	XQP0 and XQP2
Q15 - Q21 → P15 - P21	XQP1
Q22 - Q23 → P22 - P23	XQP3

These clock signals are used singly or together as determined by Central Control.

The other two inputs to each P register flip-flop are used to shift the P register internally, one place left or right. Therefore the #1 input to each bit of the P register is the output of the preceding bit (ie. next lower significant bit) and the #3 input is the output of the succeeding bit of the P register (ie. next most significant bit). Central Control can shift the contents of the P register one place left by energizing the #1 clock waveform XPL0, XPL1, and XPL2 or it can shift the contents of the P register one place right by energizing the #3 clock input XPR0, XPR1, and XPR2. These clock waveforms are divided into groups for loading purposes. There is no requirement for shifting only part of the P register at a time.

There are three exceptions to the P register inputs mentioned above, these are as follows:-

(1) The #1 input of bit 0 of register P is determined by Microprogram gates 1E10/1, 1E10/6, 1F11/6, 1E12/6 and 1E8/6 on drawing 0. Thus, the value of the least significant bit of P, when shifting P left internally, is decided by Microprogram. The function of each of these gates is described later in the section on Microprogram.

(2) The #3 input of bit 23 of the P register is similarly governed by a set of Microprogram gates on drawing 23.

(3) The #3 input of bit 22 of the P register can be either:-

(a) Bit 23 or

(b) Other data determined by Microprogram; this is governed by the set of Microprogram gates shown on drawing 22.

The zero output of each bit of the P register goes to a Nor gate (for example, 1E11/2 on drawing 0) whose output is the corresponding bit of the K highway. The gating waveforms -CPK15 and -CPK9 are used to route either the least significant 15 bits or the most significant 9 bits or all (if both are negative) of the contents of register P onto the K highway. These signals are generated by Central Control.

The gates routing bits 0 - 14 inclusive onto the K highway are inhibited by the waveform +CNK15. Thus, if register P and register N are both routed to the K highway simultaneously, the least significant 15 bits of P will be replaced by the contents of N.

A special routing facility is provided which sets bits 4 to 7 of the P register onto bits 0 to 3 of the K highway. The gates which provide this facility are shown on drawings 0 to 3. The gating waveform is -CSPK (for example, 1F16/6 on drawing 0).

2.6 N REGISTER

The N register consists of 15 triple-entry flip-flops as shown on drawing 27. The #1 input to each bit of the N register is the corresponding bit of the Q highway. The #1 clock input to each bit of the N register is gated by Central Control and provides the facility of inserting the contents of the Q highway into the N register when required. This clock waveform is divided into two parts, XQN0 and XQN1 for Pulse Gate loading purposes only.

2.6 continued

The #2 input to each of the least significant 10 bits of the N register provides the facility of subtracting one from the number stored in the N register. The quantity N-1 is provided by a half-subtractor circuit whose input is the contents of the N register. The half-subtractor Nor gates are shown directly below each bit of the N register on drawing 27. This logic produces the quantity N-1 by inverting the contents of each bit of the N register, starting at the least significant end, up to and including the first bit which contains a 1; the remainder of the N register is left unchanged. Therefore, the input to the least significant bit N0 is simply its own inverted output. The input to N1 is its own inverted output providing that N0 contains a zero (Nor gates 1E2/5 and 1E2/2). The input to N2 will be its own inverted output providing that the output of gate 1E2/5 is positive. This will be true only if N1 is being switched to a 1. N1 will be switched to a 1 only if both N0 and N1 contain zeros.

The logic required for each additional stage is identical to that for bit N2 with the exception of gates 1F8/1 and 1F14/6. These gates decode the fact that the first 4 bits and the first 8 bits respectively of N all contain zeros and are required to prevent the build-up of a long series of logic. When the first bit which contains a 1 is reached, the output of the gate driving its 1 input will be negative (for example, 1E2/5 for N1). This will cause all gates driving inputs on more significant bits to be turned off. The contents of the N register in all more significant bits will therefore remain unchanged because of that property of the flip-flops which states that if both inputs to a flip-flop are negative the contents of the flip-flop will not change if a clock pulse is applied. The #2 clock input XCN is gated by Central Control which therefore controls the facility of counting down by one in the N register.

The #3 input to each bit of the N register is used for inserting the constant 24 into the N register. Therefore, the '0' side of each bit is grounded with the exception of bits N3 and N4 on which the '1' side of the #3 input is grounded. The #3 clock input XNC0 and XNC1 can therefore be used by Central Control to insert the constant 24 into the N register and clear its original contents.

The zero output of each bit of the N register is connected to a Nor gate whose output is the corresponding bit of the K highway. The Central Control routing waveform -CNK15 can therefore be used to route each bit of the N register onto the corresponding bit of the K highway. In shift and normalize instructions, it is desired to use only the least significant 9 bits of the N register and therefore the gates routing bits N9 to N15 onto the K highway are inhibited by waveform +CG11. These gates are shown on drawing 27 (for example, gate 1E10/2 for bit 0).

The '0' side of each bit of the N register is also connected to a gate whose output is the corresponding bit of the SAD highway. The Central Control routing waveform -CNJ can therefore be used to route the contents of the N register to the SAD highway. Each bit is then inverted so that both phases are available for addressing the Core Store. Thus, for example, gate 1E31/3 produces -SAD2 from the signal +SAD2.

This inverter can be either a standard Nor type FP-01 or a Power Nor type FP-83 depending on the core store requirements. If the core store requires a Power Nor driving the address bus, a further inversion of the SAD highway can be provided (for example, gate 1E33/4 for bit 0) so that both phases of the SAD highway will be driven by Power Nor's. Packages 1E33, 1E37 and 1F37 will not be inserted unless this capability is required. If it is required then the core store address cable should be plugged into 1FJ1 instead of 1FJ2.

The one output of each of the least significant 6 bits of the N register is inverted twice (for example, by 1D12/3 and 1F20/6 for bit 0) to provide both phases of the peripheral address highway BAD. The logic for bit 5 is slightly different to allow the facility of making both phases of BAD5 positive. This is accomplished by Central Control waveform -CNB which is normally positive and is interpreted by the Peripheral Control as meaning that no peripheral address is being sent. When -CNB is negative, BAD5 is allowed to take up its proper values and the peripheral control being addressed can take the proper action.

Bits 10 and 11 of the N register are used to specify the mode of shifting in group 11 instructions and therefore these signals are used extensively in the Microprogram logic. To provide the required driving capability, each output of these two flip-flops is inverted. The special input to 2F25/4 normally has no effect. Its purpose will be described under the Microprogram section on the normalize instruction.

2.7 REGISTERS F, X, and M

Register F consists of seven single entry flip-flops and is shown on drawing 29. The input to each bit of the F register consists only of Ground on the zero side. Thus the clock waveform XCF, which is gated by Central Control, can be used to clear the F register. Data is set into the F register by gates and diodes driving the collectors of the flip-flops. Data from the core store is set into the F register during a core store read cycle by Central Control gating waveform -MSP2 (for example, gate 1C13/5 for bit 0). The core store data lines and Handswitch lines, which drive the F register, supply bits 14 to 20 which specify the position of the seven function bits in an instruction word. Similarly, data can be set into the F register from the Engineer's Handswitches by the strobing waveform -CHSF (for example, by 1C13/2 for bit 0). The waveform +CFF is used by Central Control to force the Function register to contain a 123 instruction. This facility is useful in start-up procedures.

Three sets of waveforms from the F register are required by Microprogram. The first set is simply the contents of each bit of the F register. To provide the required driving capability each flip-flop output is inverted and waveforms RF0 to RF6 are formed thereby. The second set of waveforms is a decoding of the binary contents of the least significant 3 bits of the F register. Both phases of each of these eight waveforms named CF0 to CF7 are provided as shown on drawing 29. The third set of waveforms consists of a decoding of the most significant 4 bits of the F register and are named CG0 to CG17 (octal). These waveforms define the various groups of Functions in which the F register contents is to be found. These three sets of waveforms are used extensively in the Microprogram to define the required sequence for the instruction currently being obeyed. The special inputs +CHM and +CIOH1, which are shown on some of the gates producing CG waveforms, are used to inhibit the F register decoding during Hesitations. Their purpose will be fully described under the section on Hesitation microprogram.

The X register consists of three single entry flip-flops shown on drawing 30. The zero input to each bit of this register is grounded and the clock input XCX can therefore be used by Central Control to clear the X register when required. Information is set into the X register on the collectors of the flip-flops by the same waveforms that are used for the F register. The core store and Handswitch data lines which provide data to the X register are bits 21 to 23 which is the position of the three X bits in an instruction word.

The zero output of each flip-flop in the X register is connected to a Nor gate whose output is the corresponding bit of the SAD highway (gates 1C5/5, 1C6/5 and 1C9/5 on drawing 30). This routing is controlled by Central Control waveform -CXDJ10. The routing waveform -CXDJ11 can route the quantity X + 1 onto the SAD highway. The six gates on which this waveform is used produces the quantity X + 1 on the SAD highway by a rule which states that the inverse of the contents of each flip-flop will be routed to SAD unless at least one flip-flop of lower significance contains a zero, in which case the normal content of the flip-flop will be routed to SAD. Each output of each bit of the X register is inverted on the lower half of drawing 30 to provide the driving requirements of a set of decoding gates which provide waveforms +CTX0 to +CTX7. These waveforms are a decoding of the binary contents of the X register and are used in the Microprogram, (74 instruction).

The M register consists of two single entry flip-flops shown on drawing 30. The zero input of each flip-flop is grounded so that Central Control can use the clock input XCX to clear the M register when required. Data is set into the M register by the waveforms -MSP2 and -CHSF, as are used on the X and F registers. The core store and Handswitch lines which drive the M register are bits 12 and 13 which are the positions of the modifier (or Index) bits in an instruction word. The zero output of each bit of the M register is connected to a gate whose output is the corresponding bit of the SAD highway. The routing waveform -CMDJ routes M to SAD. The two gates 1C9/3 and 1C9/4 provide a waveform -CM0 which is negative only when both flip-flops of the M register contain zeros. This waveform is used by the Microprogram.

2.8 MILL

The Mill consists of 24 binary adder sections, a carry system, and a set of shift gates. All of this logic is shown on drawings 0 - 23 incl. with the Carry System on drawing 25.

The adder (excluding the carry system) consists of eight Nor gates for each binary bit to be added. These eight gates are shown in identical positions on each of drawings 0 to 23. On drawing 0 they are 1E11/1, 1E11/3, 1E11/4, 1E12/4, 1E12/3, 1E12/2, 1E12/1, and 1E12/5. The data inputs to each stage of the Mill are the L highway, the K highway, and the carry signal (shown on drawing 0 as the output of 1E10/4). Data output from the Mill consists of the sum digit (+RMS), the carry generate signal (-RMG), and the carry transmit signal (-RMT). There are also two control signal inputs +CAND and +CLOG. These control inputs are used to cause the Mill to produce the AND function or the EQUIVALENT function when required. When +CAND and +CLOG are both negative, the Mill is set up for adding and the relationship between data inputs and outputs is as shown in Table 5 below.

TABLE 5

Inputs			Outputs		
RK	RL	CARRY	RMS	RMG	RMT
0	0	0	0	0	0
0	1	0	1	0	1
1	0	0	1	0	1
1	1	0	0	1	1
0	0	1	1	0	0
0	1	1	0	0	1
1	0	1	0	0	1
1	1	1	1	1	1

The carry generate waveform -RMG (produced by 1E11/3 and 1E11/4 on drawing 0) is negative only when that particular digit would generate a carry and is independent of the carry into that digit. Similarly, the waveform -RMT (produced by 1E11/1 on drawing 0) is negative only when that digit would transmit a carry signal and is also independent of carry into that digit. These two signals are the inverses of the AND function and the OR function respectively of the K and L highways as shown by equations 1 and 2.

$$-RMG = \overline{K.L} = \overline{K} \vee \overline{L} \qquad \text{Equation 1}$$

$$-RMT = \overline{K \vee L} \qquad \text{Equation 2}$$

Table 5 shows that the sum digit RMS is equal to the exclusive OR function of K and L if there is no carry into this digit, and the inverse of the exclusive OR function of K and L if there is carry. The exclusive OR function is provided (by 1E12/1 on drawing 0) by performing the AND function on -RMT and the inverse of -RMG as shown in equation 3.

Providing C = 0,

$$+RMS = \overline{(K.L)\ \overline{(K \vee L)}} \qquad \text{Equation 3}$$

$$= K.\overline{L} \vee \overline{K}.L$$

If there is carry into this digit the exclusive OR function is produced (by 1E12/3 on drawing 0) and then inverted to form RMS (by 1E12/5 on drawing 0). The proper signal is selected by the carry signal or its inverse.

The control signal +CLOG has the effect of forcing each digit of the Mill to select the inverse of the exclusive OR as its output. This is the EQUIVALENT function and can be produced whenever required by Central Control. If the control waveform +CAND is positive at the same time as +CLOG, the output of the Mill will be the AND function of K and L. Therefore, by controlling these two inputs, Central Control can cause the Mill to produce the sum, the EQUIVALENT function, or the AND function on the M highway, +RMS.

The carry system divides the Mill up into six blocks, each containing four digits. The logic for carry within each block is shown on drawings 0 to 23 and the interblock carry logic is shown on drawing 25. Considering Block 0 (digits 0 to 3) as a typical block the logic operates as follows. The only requirements for carry into digit 0 is that there be carry into block 0 (1E10/4 on drawing 0). On all higher significant blocks the carry into block signal CYB will be produced by considering carry out of lower significant blocks. On block 0 this signal is a waveform used by Central Control to force carry into the bottom of the Mill in order to add one. Digit 1 will have carry into it if either of two conditions apply, namely:-

(a) Digit 0 generated a carry, or

(b) There was carry into block 0 and digit 0 transmitted that carry

2.8 continued

This is produced by gates 1E4/4 and 1E4/2 on drawing 1. Therefore, in general, it can be said that there will be carry into any digit if all digits between it and the last digit to generate a carry are transmitting that carry. The interblock carry signals are generated on drawing 25. The five sets of gates at the top of drawing 25 derive block generate signals which are equivalent to the digit generate signals already described. The five gates at the bottom of drawing 25 produce block transmit signals which are identical to the digit transmit signals already described. These block generate and block transmit signals are used to generate the carry into block signals -CYB1 to -CYB5 shown on drawing 25. For example, -CYB1 will be negative if either:

(a) +CYB0 is positive and the block 0 transmit signal (the output of 1F5/6) is positive, or

(b) If block 0 generate is positive (output of 1F7/1, 1F7/2, 1F7/4 and 1F7/5). This inter-block carry logic is identical to the inter-digit carry logic in that there will be carry into a block if all blocks between it and the last block which generate a carry signal are transmitting that carry.

This rather elaborate carry system reduces the number of series gates through which logic must propagate. With the system used in FP6000 the longest possible logic path between the input to the Mill (K & L highways) and the output of the Mill (M highways) is six gates, whereas the simplest form of carry propagation would require a series path of 48 gates.

The remaining section of the Mill is the shift gates which are shown on drawings 0 to 23 (for example, 1D12/2, 1D12/6, and 1D12/1 on drawing 0). Normally, each output of the Mill (the M highway) is routed directly to the corresponding bit of the Q highway by Central Control waveform -CMQ. This waveform is broken into two parts, -CMQ6 and -CMQ18, to allow Central Control the facility of routing out either the least significant six bits or all 24 bits of the M highway onto the Q highway. Central Control waveform -CSHR1 routes each bit of the M highway to the next lower significant bit of the Q highway (1D12/6 on drawing 0). Similarly, waveform -CSHL1 routes each bit of the M highway to the next higher significant bit of the Q highway. The contents of Q23 when shifting right, and of Q0 when shifting left, are determined by the Microprogram since they depend on the operation being performed (gates 1D12/1 on drawing 0 and 1D19/6 on drawing 23).

2.9 MISCELLANEOUS ARITHMETIC UNIT FACILITIES

One additional facility provided is that of repeating the least significant six bits of the Q highway on each of the other three blocks of six bits of the Q highway. The operation is likened to the spreading of a fan and is used to take a character in position 3 and insert it into any other character position in the B register. This character can arrive on the Q highway either through the Mill by using the waveform -CMQ6 or from the peripheral input highway BIN. The gates providing this facility are shown on drawings 6 to 23 and the gating waveform is -MSP29.

One gate on each of drawings 0 to 23 provides the facility of routing the peripheral data input highway +BIN to the Q highway (for example, 1D12/5 on drawing 0). Central Control may route only the least significant six bits to Q using waveform -MSP26 or it may route all of BIN to Q using both -MSP26 and -MSP17.

Each bit of the K highway is inverted by a Nor to form the peripheral data output bus -BOUT (for example, 1E13/4 on drawing 0). This waveform is not gated and therefore the contents of -BOUT is always the inverted contents of the K highway.

Another facility required is that of placing the least significant nine bits of the K highway onto the most significant nine bits of the Q highway. This is done by one gate on each of drawings 15 to 23 by the gating waveform -CSHL15 (for example, 2E26/6 on drawing 15). During this operation the least significant fifteen bits of the Q highway will contain all 1's since every gate driving -RQ0 to RQ14 is turned off. Microprogram normally makes use of the special strobing facilities into registers to remove these 1's if they are not desired.

Microprogram often has special requirements to insert data onto bits 0, 22, or 23 of the Q highway. Several Microprogram gates are shown on each of the corresponding drawings. The purpose of each of these Microprogram gates will be described in the section on Microprogram.

2.10 D, LIM and G REGISTERS

These three registers are all used to hold information pertaining to the program currently operating in the machine in applications where program time sharing is being used.

2.10.1 G Register

The G register consists of four single entry flip-flops which are shown on drawing Q. The input of each bit of the G register is the corresponding bit of the Q highway. Information can be set into the G register by Central Control by energizing the clock waveform XQG. The two least significant bits of the G register contain the program number and are used to light one of four lamps on the console typewriter which indicate the program currently running in the computer. The most significant two bits of the G register hold the monitor mode information on the program currently running. The outputs of these two flip-flops are used by the Microprogram gates shown on drawing Q to determine when a monitor mode interrupt should be originated.

2.10.2 D Register

The D register consists of nine single entry flip-flops shown on drawing 28. The inputs to the D register flip-flops are bits 6 to 14 of the Q highway. This corresponds to the position of the datum address held by the Executive program in the program index word. Central Control can insert data into the D register from the Q highway be energizing the clock waveform XQD. The zero output of each bit of the D register is connected to a gate whose output is one bit of the SAD highway (for example, 1G32/5 on drawing 28). Central Control waveform -CDJ is used to route the contents of the D register onto the core store address highway. The zero side of each bit of the D register is also connected to a gate whose output is one of bits 6 to 14 of the L highway (for example, 1G32/3 on drawing 28). Central Control routing waveform -CDL can therefore be used to route D onto the L highway. Each output of each bit of the D register is also connected to the datum and limit tester gates shown across the centre of drawing 28. The operation of these gates will be described under the section on the datum and limit tester.

2.10.3 LIM Register

The LIM register consists of nine single entry flip-flops shown at the top of drawing 28. The input to each bit of the LIM register consists of one of bits 15 to 23 of the Q highway. This defines the position of the limit address held by Executive in the program index word. The outputs of the limit register flip-flops are used only in the datum and limit tester.

Data is loaded into the D, LIM and G registers only by the special executive orders functions 172 and 173.

2.11 SUMMARY OF ARITHMETIC UNIT FACILITIES

(a) Q Highway Strobing

FROM Q BITS	TO	WAVEFORM
0 - 23	A Register (all)	XQA0, XQA1, XQA2
0 - 23	B Register (all)	XQB0 - 5
0 - 5	B Register (0 - 5)	XQB0)
6 - 8	B Register (6 - 8)	XQB1)
9 - 11	B Register (9 - 11)	XQB2)
12 - 14	B Register (12 - 14)	XQB3)
15 - 17	B Register (15 - 17)	XQB4)
18 - 23	B Register (18 - 23	XQB5)
0 - 23	P Register (all)	XQP0 - 3
15 - 21	P Register (15 - 21)	XQP2
22 - 23	P Register (22 - 23)	XQP3
0 - 14	N Register (all)	XQN0, XQN1
0, 1, 4, 5	G Register	XQG)
6 - 14	D Register (all)	XQD)
15 - 23	LIM Register (all)	XQL)

NOTE: These waveforms can be used together to give any desired combination (eg. Q6 - 11 & B6 - 11, XQB1 & XQB2).

NOTE: Always used simultaneously.

(b) Other Strobing

FUNCTION:	WAVEFORMS:
Clear F Register	XCF)
Clear X and M Registers	XCX)
Shift P Register right 1 place	XPR0 - 2
Shift P Register left 1 place	XPL0 - 2
Count N Register down by 1	XCN
Set N Register contents = 24	XNC0 and XNC1

NOTE: Always used simultaneously.

2.11 continued

(c) Routing to K Highway

FROM	TO K BITS	WAVEFORMS:
B Register (all)	0 - 23	-CBK6 & -CBK18
B Register (0 - 5)	0 - 5	-CBK6
B Register (0 - 11)	0 - 11	-CBK6, -CBK18, +MSP1, +MSP23
B Register (0 - 14)	0 - 14	-CBK6, -CBK18, +MSP23
B Register (0 - 5)	0 - 5	-CBK6 forced by -CBCHK and -CH3
B Register (6 - 11)	0 - 5	-CBCHK, -CH2
B Register (12 - 17)	0 - 5	-CBCHK, -CH1
B Register (18 - 23)	0 - 5	-CBCHK, -CH0
P Register (all)	0 - 23	-CPK15, -CPK9
P Register (0 - 14)	0 - 14	-CPK15
P Register (15 - 23)	15 - 23	-CPK9
P Register (0 - 8)	0 - 8	-CPK15, +MSP24
N Register (all)	0 - 14	-CNK15
P Register (4 - 7)	0 - 3	-CSPK

(d) Routing to L Highway

FROM	TO L BITS	WAVEFORMS
A Register (all)	0 - 23	-CAL
Inverse A Register (all)	0 - 23	-CIAL
Special Constants	0 - 23	-CAL, -CIAL, -MSP6 - 8
D Register (all)	6 - 14	-CDL

2.11 continued

(e) Mill Functions

FUNCTION:	WAVEFORMS
Add	none
Add 1	+CYB0
Equivalent	+CLOG
And	+CLOG, +CAND
M to Q (all)	-CMQ6, -CMQ18
M to Q (0 - 5, 6 bits only)	-CMQ6
M Shift left to Q	-CSHL1
M Shift right to Q	-CSHR1

(f) Core Store and Handswitch Routing

FUNCTION:	WAVEFORMS
Clear B Register	-CBC (0.3 usec -ve pulse)
Core Store data to B	-CSB
Handswitch Data to B	-CHSB (0.5 usec -ve pulse)
Core Store data to F, X, M	-MSP2
Handswitch data to F, X, M	-CHSF (0.5 usec -ve pulse)

(g) Miscellaneous Facilities

FUNCTION	WAVEFORMS
Fan Q (Repeat 0 - 5 on each character)	-MSP29
BIN to Q (all)	-MSP26 & -MSP17
BIN to Q (0 - 6)	-MSP26
K0 to K8 - Q15 to Q23	-CSHL15

SECTION 3

CENTRAL CONTROL UNIT

3.1 General Description

The Central Control unit consists basically of a set of flip-flops each representing one or more control functions. The main purpose of the Central Control unit is to provide:-

(a) Control signals to the arithmetic unit to define the operation to be carried out.

(b) Control signals to the core store.

(c) Present control setting signals to the Microprogram to be used in sequence decoding.

(d) Access by Microprogram output signals which will provide the next control setting.

The Central Control logic is shown on drawings M to T. In general, each control flip-flop on these drawings corresponds to a particular control function in the arithmetic unit or core store. In most cases a variety of control functions must be available to the Microprogram. However, in other cases, control functions are mutually exclusive and therefore a separate flip-flop for each is not required (for example, only one of the three possible Mill operations can be carried out at any time). In such cases a set of flip-flops is used and the outputs are decoded to provide the correct control signals.

It is often important that the amount of logic between the control flip-flop and its final control signal is kept to a minimum. For example, all logic between a routing flip-flop (which routes a register onto the K or L highways) and the control signal itself must be added to the total length of logic through which signals must be able to propagate in one digit time. One gate is normally required because of the limited driving capability and restrictive lead length requirements of the flip-flops. This time requirement is not important for signals which control operations late in the arithmetic unit data path, such as -CMQ. This restriction also does not apply during any core store READ cycle since, in this case, the time between the read command to the core store and the arrival of the data in the DATUM register is available for the propagation of control logic. This time is occasionally used to force control signals directly from microprogram output signals without setting the control flip-flop.

3.2 Register Routing - Drawing M

Drawing M shows the five main register routing control flip-flops. Underneath each flip-flop is a designation of its control function. For example, the flip-flop 2C10/1 in the upper left corner of drawing M controls the routing of the A register to the L highway. The "one" output of each of the flip-flops is connected to gates which produce the control signals for use by the arithmetic unit. Both outputs of each flip-flop are connected to gates which produce both phases of signals specifying the state of that flip-flop to be used by the Microprogram decoding. Examples of such waveforms are -CAL24, and +CAL24.

There are several instances on drawing M of Microprogram waveforms which force control waveforms directly without setting the flip-flop. For example, the waveforms +MAL1 and +MAL2 force the control waveform -CAL. This is allowable since these two Microprogram waveforms are generated during READ/PAUSE cycles of the core store.

The input to each of these control flip-flops consists of a mix of many Microprogram waveforms. If any one of the Microprogram waveforms on the input is positive then the flip-flop will be set (by the next clock pulse) and the operation it specifies will take place during the following digit time. If none of the Microprogram waveform inputs are positive, the flip-flop will not be set or will be reset. The clock input XB2 to all of these control flip-flops is not gated (see description of drawing S) and therefore they receive every clock pulse which is generated.

Gate 2C3/1 in the upper right hand corner of drawing M forces the control waveform -CBK6 when it is desired to route one of the characters in the B register onto the K highway and character 3 is the selected character (as described in the section on the Arithmetic Unit). Gate 2C3/2 in the lower right hand corner of drawing M is a Microprogram gate which drives +SAD0. The function of this gate will be described in the section on Microprogram.

3.3 Core Store Address Routing - Drawing N

Drawing N shows four flip-flops which control the routing of registers to the SAD highway and two flip-flops which control the routing of the constants 9 and 10 to the SAD highway. Since only one of register N, X, X + 1 or M or the peripheral CONTROL WORD can be routed to the SAD highway at one time, three flip-flops are sufficient for specifying all the required conditions. Flip-flops 2B8/3, 2B6/1 and 2B6/3 are used as shown in the table on drawing N to specify which routing is desired. If none of these flip-flops are set then no address routing takes place. Note that the SAD highway is abbreviated J for the purpose of control function designations and waveform names.

The input to each routing flip-flop consists of a mix of a large number of Microprogram waveforms. The flip-flops are set by the output of each mix according to the table on drawing N. The output of each mix that establishes the settings for X → J, X + 1 → J, or M → J is also taken to gates which will set D → J. Thus the normal condition is that the content of the D register is added to the content of the X, X + 1, or M registers whenever one of these three is routed to the SAD highway.

When the machine is operating in Executive Mode the setting of D → SAD is normally inhibited since the datum for the Executive program is normally zero yet the DATUM register may contain the datum for one of the operating programs. Mode bits 20 and 21 of the P register specify whether or not the Executive program wishes its X and M addresses respectively to have datum added. The additional restriction is applied that X addresses cannot have datum added in functions of group 4, group 11 or group 12 while in Executive mode since the location register, ie. the contents of P, is stored during these instructions. Therefore, the gates in the upper left corner of drawing N inhibit the setting of the D → SAD flip-flops as follows:-

(a) The common output of 2B10/2 and 2B10/5 is positive when in Executive Mode and either the contents of P20 are zero or the function being obeyed is in one of groups 4, 11, or 12. This waveform inhibits gates 2B10/4, 2B11/3 and 2B11/4 which normally cause the D → SAD flip-flop to be set when either X → SAD or X + 1 → SAD is being set.

(b) The output of 2B10/3 is positive when in Executive Mode and the contents of P21 is zero. This signal inhibits gate 2B11/5 which normally causes D → SAD to be set when M → SAD is being set.

The clock pulse input to each of these flip-flops is not gated so that each one receives every clock pulse that is generated.

The outputs of the X → J, N → J, and X + 1 → J flip-flops are decoded according to the table on drawing N to provide the required control waveforms. These decodings are done on the collectors of inverters connected to the flip-flop outputs so that the negative going control signals required can be generated with only one gate delay between it and the flip-flop. The signals -CXDJ10, +CNJ15 and -CNJ15 are generated for use in the Microprogram decoding.

The CONTROL WORD address routing waveform -CCWJ inserts a one onto bit 6 of the SAD highway (via gate 2B7/2) and enables gates 2B3/1 to /6 which drive the least significant six bits of the SAD highway. The other input to each one of these gates is a signal which is generated in the HESITATION CONTROL and which sends to the SAD highway the CONTROL WORD address of the particular peripheral being processed. Thus the table of peripheral control words starts at base address 64 and extends upwards to a maximum of address 127.

The "one" input of the D → J flip-flop is connected to gate 2B5/1 which provides the control signal -CDJ. Certain microprogram operations shown as a mix on gate 2B9/1 require the setting of the D → J flip-flop by itself. The actual address required under these circumstances is D + 8. Gate 2B7/1 decodes the fact that the D → J flip-flop is set by itself and sets a 1 onto bit 3 of the SAD highway if these conditions are satisfied. When this facility is being used in the Executive program the address required is 8. Since the D register may contain the datum for some operational program the waveform -CDJ is inhibited by gate 2C9/6 when the machine is operating in Executive Mode.

The flip-flops 2B14/3 and 2B14/1 force the constants 9 and 10 respectively onto the SAD highway. These addresses are required for storing the contents of the B and N registers before carrying out a HESITATION in the middle of a long order.

3.4 Mill Control and Miscellaneous - Drawing O

Drawing O shows 9 control flip-flops which are used to generate the control signals for the arithmetic unit and the Microprogram.

The two flip-flops SHR and SHL control the gates routing the output of the Mill (the M highway) onto the Q highway. If the SHR flip-flop is set by any one of the microprogram waveforms shown to its left, the control waveform -CSHR1 will enable all of the gates which shift the output of the Mill right one bit and route it onto the Q highway. In this case -CMQ6 and -CMQ18 will be inhibited.

If the SHL flip-flop is set, a similar function will be performed on gates which shift the output of the Mill left one bit before routing it onto the Q highway. If neither of these flip-flops are set, -CMQ6 and -CMQ18 will be active unless inhibited by +MSP17 and +MSP26 which do a BIN → Q operation or by +CSHL15 which shifts the contents of the K highway left 15 places and inserts it on the Q highway. -CMQ18 alone will be inhibited by -MSP29 which does a FAN Q operation. -CSHR and -CSHL are used by the Microprogram.

The control waveform -CN1 is used by the microprogram for decoding purposes and the control waveform +CN1 is used on drawing F on the input of pulse gate 1G26/1 to enable a clock pulse to be sent to the N register to count down the contents of the N register by one whenever the CN1 flip-flop is set. The control waveform -CDL obtained from the flip-flop 2B36/3 is used on the slice to route the DATUM register onto the L highway and also by the Microprogram for decoding purposes. It should be noted here that the input -CEXM to gate 2A37/4 inhibits the three microprogram waveforms at the left from setting the D → L flip-flop while in Executive Mode. The two flip-flops CYB0 and CLOG are used to specify the three types of operations possible within the Mill and also the conditions for CYB0. If the CYB0 flip-flop is set and the CLOG flip-flop is not set, -CYB0 will become active. If the CLOG flip-flop is set and the CYB0 flip-flop is not set, the two waveforms +CLOG0 and +CLOG1 will become active and will cause the Mill to do an EQUIVALENT operation. If both of the flip-flops are set +CAND will become active and will cause the Mill to perform an AND operation. If neither of the flip-flops are set (or if only CYB0 is set) the Mill will perform a normal sum operation. The gate 2C12/6 in the upper right corner which generates +SAD1 will be described in the section on Microprogram.

The NIL flip-flop does not itself control a specific operation within the machine. It is used to specify micro-operations which cannot be specified in any other manner. Its operation will be described in more detail in the section on Microprogram. The function of the three gates 2A25/3, 2A26/5 and 2A25/4 which generate waveform -CSPK will also be described in more detail in the section on Microprogram.

The PH flip-flop is set by the Microprogram waveforms as shown but is not reset until end of order (note that this differs from all other control flip-flops on this drawing). It is used by the microprogram to distinguish between two common microfunctions within a single instruction. The Microprogram uses the two waveforms -CPH1 and -CPH0 to accomplish this.

The BCH → K flip-flop is used in conjunction with the two gates 2C19/2 and 2C17/6 to generate the waveform -CBCHK which controls the gates routing BCH to K (see section on arithmetic unit). The operation of these two gates will be described in more detail in the section on Microprogram.

Mill Functions

Flip-Flop Status		Function	Active Waveforms
CYB0	CLOG		
0	0	Add	None
1	0	Add 1	+ and - CYB0
0	1	EQUIVALENT	+ CLOG0 - 1
1	1	AND	+ CLOG and + CAND

3.5 READ/WRITE CONTROL - Drawing P

The flip-flops 2B20/1 and 2B20/3 which are designated READ R/P and WRITE R/P respectively as shown in the centre of drawing P, generate the control waveforms which select the required core store cycle as shown by the table directly below.

Flip-flop Status		Control Waveform	Core Store Cycle
READ	WRITE		
1	0	CR	Read Regenerate
0	1	CW	Clear Write or Write after Read Pause
1	1	CRP	Read Pause

3.5.1 Normal Operation

The many input signals which control these flip-flops are applied to FP-84 Diode OR cards (as shown at the left of Drawing P) and thence to five separate NOR's namely, 2B25/1, 2B22/6, 2B22/1, 2B21/6, and 2B21/1. In addition to producing the flip-flop control signals, these gates also generate Slow Clock signals as shown in the table below and which are used as described in section 6.5. Note that the designation -Slow Clock #1 does not appear on Drawing P and is only included to clarify this description.

With the exception of 2B21/6 (-Slow Clock #3), all these input gates are connected to gates which drive Pulse Gate 2B17/1. Thus, providing the circuitry shown in the upper LH corner of Drawing P does not generate any active inhibiting waveforms, the input of Pulse Gate 2B17/1 will go positive at the onset of all types of core store cycle except Write following Read/Pause and, at the next clock pulse, +CCI (Cycle Initiate Pulse) will be transmitted to the core store.

Provided the waveform +MIBC is not active (described more fully in the section on Microprogram), the waveform -CBC will become active at the onset of each Read Regenerate and Read Pause cycle. As noted in the section on the Arithmetic Unit, -CBC clears the B register at the start of these two types of cycle.

READ/WRITE CONTROL CONDITIONS

Input Gate	-Slow Clock #	Flip-flop Setting		Core Store Cycle	Control Waveforms Generated
		READ (2B20/1)	WRITE (2B20/3)		
2B25/1	1 (a)	1	0	Read Regenerate	CR +CCI -CBC -CCC -
2B22/6	1 (b)	1	0		
2B21/1	1 (c)	0	1	Clear Write	CW +CCI - -CCC -
2B21/6	3	0	1	Write (following Read Pause)	CW - - -CCC -
2B22/1	2	1	1	Read Pause	CRP +CCI -CBC - +CSW

3.5.1 continued

The waveform -CCC (Continue Core Store Cycle) becomes active for both Read and Write conditions but not during a Read-Pause cycle. This is accomplished as follows, gate 2C19/5 is enabled during a Read-Regenerate cycle and gate 2B18/5 is enabled during a Clear Write or Write (following Read-Pause) cycle. Note that -CCC is determined only by the status of the READ and WRITE flip-flops whereas +CCI can be inhibited by other factors. Thus, a Read or Write cycle having started, -CCC ensures that such a cycle is completed.

The exception to the foregoing, namely, the Read-Pause cycle requires that the operation be completed by initiating a Write (following Read-Pause) cycle. The latter is started by the waveform +CSW (Write Cycle Initiate) which becomes active at the next clock pulse following the enabling of gate 2B18/4 by -CRP (Read-Pause).

The actions initiated by the five input gates are clearly shown in the table opposite. Note that gates 2B25/1 and 2B22/6 produce identical results during normal operation; however there is a distinct difference between them when operating under manual conditions as described later.

3.5.2 Inhibiting Factors

If a Datum/Limit failure occurs the top input to gate 2B33/5 in the upper left corner of drawing P will become active and the output of this gate will become negative. This waveform +CDLF will also set the ILLEGAL OPERATION flip-flop (see description of drawing Q) and this in turn will make the waveform +CDLF1 active (+CDLF could disappear at this time). Therefore, ignoring all other inputs to gate 2B26/6 except the output of 2B33/5, the output of gate 2B26/6 will become positive. Since the output of this gate is an input to 2B26/1, 2B25/4, 2B25/3 and 2B25/2 it will inhibit these gates from making the input to pulse gate 2B17/1 positive. This in turn will inhibit the CYCLE INITIATE pulse to the core store from becoming active at the next clock pulse. The function of gates 2B25/5 and 2B26/6 is more fully described in the section on Reservations Check.

3.5.3 Manual Operation

The foregoing description presumes that the Handswitch Mode control (shown in the upper LH corner) is set to NORMAL. However, for test purposes, the Handswitches can be used to enter data directly into the B register and the Clock can be placed under manual control as described later in section 6. Under these circumstances the Core Store operation is partially or entirely inhibited as described below.

The Handswitch Mode control switch has 3 positions, as follows:-

Mode 1 Normal operation

Mode 2 ORDER - The Order Number is read from the Handswitches and the operands are read from the Core Store.

Mode 3 ALL - all the required data is derived from the Handswitches.

When the Handswitch Mode control is turned to Mode 3, the series of gates 2B26/1, 2B25/4, 2B25/3, etc. is inhibited. Therefore, regardless of the conditions established by the microprogram waveforms at the left of drawing P, the Cycle Initiate pulse (+CCI) is inhibited and no action takes place in the core store.

When operating in Handswitch Mode 2, the significant point concerns waveform +MRN1 which only becomes active when the Order is to be extracted from the Core Store as the first phase in the operation of an instruction. +MRN1 forms an input to gate 2B25/1 which sets up a Read Regenerate cycle and Cycle Initiate Pulse as shown by the table. However, +MRN1 is also applied to gates 2B25/5 and 2B25/3 so that the HS Mode 2 setting inhibits 2B25/6 and thereby inhibits the Cycle Initiate Pulse when +MRN1 is also active.

3.5.4 Data Transfer

The two waveforms -CSB and -MSP2, shown on the RH side of drawing P, are gating waveforms used to gate the core store data into the B register and F, X, M registers respectively during Read-Regenerate and Read-Pause cycles (see table of waveform uses in section 2 - Arithmetic Unit).

The output of gate 2B18/6 will become positive thus making the waveform -CSB active if:-

(1) The flip-flop 2B20/1 is set (that is the core store is in a Read Regenerate or Read Pause cycle).

(2) The data is not being read from the Handswitches (ie. not in Mode 3 nor reading the order while in Mode 2).

(3) If +MAL2 is not active (the function of this waveform will be described more fully in the section on Microprogram).

3.5.4 continued

The output of gate 2B29/6 will be positive thus making the waveform -MSP2 active, if:-

(1) The machine is not in Hand Switch Mode 3.

(2) The machine is not in Hand Switch Mode 2.

(3) The Core Store is in a READ cycle.

(4) Special Mode is set.

(5) N is routed to J.

That is, we are reading the order in preparation of order and are not in Hand Switch Mode.

The two waveforms -CHSB and -CHSF are the two corresponding gating waveforms used during Read Regenerate or Read Pause operation while in Hand Switch Mode.

The output of gate 2B18/1 will be positive if:

(1) The machine is in Hand Switch Mode 2.

(2) Reading.

(3) In Special Mode.

(4) Routing N to SAD.

This in turn will make the outputs of gate 2B27/1 negative if the waveform +CHSBT is positive and make the output of gate 2B27/3 negative. This will make the waveform -CHSB active. The waveform +CHSBT is a special timing waveform generated by the clock (see section on clock).

The output of gate 2B18/2 will become positive if:

(1) The Hand Switch Mode switch is in Mode 3.

(2) +MAL2 is not active.

(3) The flip-flop 2B20/1 is set (that is the core store is doing a Read Regenerate or Read Pause cycle).

This in turn will try to make the output of 2B27/1 negative and if the special timing waveform +CSHBT is positive, the output of gate 2B27/3 will also be negative and the waveform -CHSB will become active.

The special timing waveform +CHSBT is necessary since, unlike the core store, the information on the hand switches is always present and not in a pulse form. Assume that the waveform -CHSB becomes active during a certain digit time. If the next clock pulse resets the state of the control flip-flops there will be certain delay (the time required for this new setting to propagate through the decoding logic) before the waveform -CHSB will be made inactive. This would therefore inhibit putting any new information into the B register during this time since the information from the hand switches like the information from the core store sets the B register via the flip-flop collectors. The special timing waveform +CHSBT is therefore used to gate the gating waveform -CHSB. +CHSBT is a positive pulse which will be positive between clock pulses, thus the waveform -CHSB will not become active only between clock pulses.

The output of gate 2B29/1 will be positive if:

(1) The Hand Switch Mode switch is in either the Mode 2 or Mode 3.

(2) The machine is in a READ cycle.

(3) Special Mode is set.

(4) N is routed to J.

This in turn will try to make the output of gate 2B27/4 negative and if at the same time the special timing waveform +CHSBT is active the output of gate 2B27/5 will also be going negative and therefore, the waveform -CHSF will become active. The special timing waveform +CHSBT is again necessary here for the reason described above.

The logic in the upper right corner of drawing P is described in detail in the section on Parity Check.

The function of gate 2A20/6 which generates the waveform +SAD0 will be described in the section on Microprogram.

3.6 DRAWING Q

3.6.1 Mill 24

The flip-flop 1C31/1 in the upper left corner of drawing Q is used to store the parity bit when receiving information from a peripheral or from the core store. It also stores the output of bit 23 of the Mill when shifting the Mill output left one position before putting it on the Q highway (see section on Arithmetic Unit). The waveform -CBC on the input to gate 1C32/4 will clear the flip-flop 1C31/1 so that information from the core store can be gated through 1C32/6 by the waveform -CSB and set this flip-flop on its collector (see description of B Register in section on Arithmetic Unit).

When receiving information from peripherals the parity bit arrives on +BIN 24 and the waveform -MSP 26, when active (see section on Microprogram) will gate +BIN 24 onto the input of the flip-flop 1C31/1 in order that it may be set by the next clock pulse if there is a "one" on +BIN 24. The gate 1C32/2 routes the output of bit 23 of the Mill onto the input of this flip-flop when shifting left. The two gates 1C33/3 and 1C33/4 generate both phases of the waveform RB24 which is used by the Microprogram. The operation of the two gates 1C30/1 and 1C30/5 is described in the section on Parity Test.

3.6.2 Waveform +RMS24

The logic in the centre of the page generates the waveform +RMS24 used in testing for overflow. Ignoring for the present the function of gate 1C30/6 and the +CG11 input on gates 1C29/1, 1C29/2 and 1C29/6 this logic will perform the following operation:-

RMS24 = K23 + L23 + Carry from slice 23

A more detailed description of how the waveform +RMS24 is used will be given in the section on Microprogram.

3.6.3 G Register and Monitor Mode

The two flip-flops 1D9/1 and 1D9/3 constitute the two least significant bits of the G Register. The sole function of these two bits is to illuminate a light on the typewriter console. The four FP-38 packages decode the four possible states of these two flip-flops and provide the driving power for the four lights located on the typewriter. These lights indicate which program the machine is presently obeying. The two flip-flops can be loaded from the Q highway (bits 0 and 1).

The two flip-flops 1G21/3 and 1G21/1 store the 2 bits which specify Monitor Mode. These two flip-flops can also be loaded from the Q highway (bits 4 and 5). The meanings of the four possible states of these two flip-flops are given in the table at the bottom of drawing Q.

Gate 1G20/3 decodes the fact that the machine is in Monitor Mode 3 and not in Executive Mode to give the waveform +CMM3. 1G20/5 gives the other phase of this waveform. The use of both phases of this waveform is described in the section on Microprogram.

The Monitor Mode flip-flop 2B32/3 can be set when not in Executive Mode (+CEXM input on gate 2B31/3) under the following conditions:-

(1) The output of gate 1G20/1 will become active setting the flip-flop 2B32/3 if:-

(a) The machine is at End of Preparation (-MEP1).

(b) In Monitor Mode 1.

(c) Not in groups 13 to 17 (+CG20).

(The MNIL1 input on gate 2B31/3 inhibits the gate 1G20/1 from setting the Monitor Mode flip-flop when the machine is doing a 23 order, ie. it is not possible to monitor at the end of a 23 order. This condition is described in more detail in the section on Microprogram).

(2) If the waveform +MPK8 becomes active this will set the Monitor Mode flip-flop. The generation of this waveform will be discussed in the section on Microprogram.

(3) The output of gate 1G20/6 will become active setting the Monitor Mode flip-flop if the machine is in Monitor Mode 2 or 3 and in a 123 order with X = 7.

If the Monitor Mode flip-flop is set the output of gate 2B31/4 will become positive thus making both phases of the waveform CINT active providing the INHIBIT INTERRUPTS switch S8 is not closed. The INHIBIT INTERRUPT switch, located on the Engineer's Control Panel and constituting one input into 2B31/2, will inhibit all Interrupts caused within the central machine. It will not inhibit Interrupts caused by peripherals. The waveform CINT will, when active, cause the machine to enter the Interrupt Microprogram upon reaching the end of the next order.

3.6.3 continued

The function of gate 2A29/1, which can also cause an Interrupt when its output is positive, is discussed in the section on Microprogram.

3.6.4 Illegal Orders

Whenever the ILLEGAL OPERATION flip-flop 2B32/1 is set the waveform +CDFL1 will become active and since this waveform is an input to gate 2B31/1, it will also cause the interrupt waveform CINT to become active, with the same result described above.

Whenever the flip-flop 2B2/3 is set CINT will again become active.

The output of gate 2B34/6 in the upper right corner of drawing Q will be positive whenever the function register contains the hypothetical functions 66 - 67, 76 - 77, or 116 - 117.

The input to gate 2B34/3 will become positive due to gate 2A28/1 whenever the machine is in Executive Mode and at End of Preparation and the Function register contains a function in groups 13 to 17 with the exception of functions 160, 171, 172 and 173. The output of gate 2B34/3 will therefore become negative whenever an illegal order is specified in the function register. Gates 2B34/5 and 2B34/2 control the microprogram for illegal orders and will be discussed in the section on Microprogram.

Gate 2B29/2 will make the waveform +CPF active causing a parity failure and stopping the clock, whenever an illegal order is specified while in Executive Mode (-CEXM) and parity is not trying to be cleared (+CCP).

The input to gate 2B33/2 will be negative whenever an illegal order is specified. Its output will therefore go positive setting the ILLEGAL OPERATION flip-flop. Gate 2B33/1 will set the ILLEGAL OPERATION flip-flop whenever there is a datum or limit failure (-CDLF) and the SUSPEND DATUM and LIMIT CHECK flip-flop is not set (+CSDLT) and the Function register is not being forced to function 123 (+CFF).

The contents of the MONITOR MODE flip-flop and the ILLEGAL OPERATION flip-flop will be routed onto +BIN21 and 22 respectively whenever the waveform -CRSR64 becomes active. This waveform will also reset both of these flip-flops on the clock pulse following. Its use will be more fully described in the section on Microprogram.

SECTION 5

CORE STORE OPERATION

5.1 GENERAL

The FP6000 computer is designed to work with core stores of any speed up to a cycle time of 2 microseconds and up to a maximum size of 32,768 words, each word containing 24 binary digits plus parity.

5.2 INTERCONNECTIONS

The central computer is interconnected with the core store by the following wires:-

(2) 24 data lines plus 1 parity line in each direction, ie. input and output.

(b) 15 address lines.

(c) Read Pulse line.

(d) Write Enable line.

(e) Write Pulse line.

The core store requires no internal register, this facility being provided by the B register in the computer. Parity generation and checking is also carried out within the central computer.

5.3 MODES OF OPERATION

There are three different modes of operation, as follows:-

5.3.1 Read/Regenerate Mode

This is used when it is desired to use the contents of a core store location within the central computer. The sequence of events for a cycle of this type is as follows:-

(1) At the clock pulse time initiating the cycle both the Read flip-flop and the Address Routing flip-flop are set.

(2) The Read flip-flop energizes a pulse which clears the B register on its collectors (ie. sets B to all zeros).

(3) The Read flip-flop is also used to generate a READ pulse on the Read Pulse line to the store.

(4) The Read flip-flop opens the gates from the data input lines to the B register collectors (ie. opens the SIN highway).

(5) During the data strobe time in the core store, data is present on the information lines and is able to set the B register flip-flops on their collectors.

(6) The Read flip-flop energizes the WRITE ENABLE line which causes the core store to continue the cycle, regenerating the data which it has just read.

(7) At the end of the core store cycle, a computer clock pulse is generated (this timing can be controlled either by adjustable internal computer timing or by an "End-of-cycle" signal from the store) which sets up a new control state in the computer.

Note that the data will be available in the B register for the write half of the core store cycle (at least 1.2 usec.) so that a logical or arithmetic operation can be carried out on the data which is read during the same computer digit time in which the reading takes place.

5.3.2 Clear/Write Cycle

This type of cycle is originated when the B Register contains information which is required to be written into a core store location. The sequence is as follows:-

(1) At the clock pulse time originating the cycle, the Write flip-flop and the Address Routing flip-flop are set.

(2) The Write flip-flop generates a READ pulse on the Read Pulse line to the store.

(3) The read cycle of the store is carried out as in 5.3.1 except that no data is placed in B because the input gates are closed. The purpose of this read cycle is merely to clear the store location required.

(4) The Write flip-flop energizes the Write Enable line which allows the write cycle to occur immediately after the read operation is finished.

5. 3. 2 continued

Since the data in the B register is unchanged, the information that was in B at the beginning of the cycle will be written into the core store.

(5) End of cycle; same as (7) for Read/Regenerate cycle.

Note that in both types of cycle described so far, the action of the core store is identical. The control lines to the store are energized in an identical way and the only differentiation between the two types of cycle is within the computer. (ie., the input gates from SIN to the B register are not opened for a Clear/Write cycle).

5. 3. 3 Read/Pause/Regenerate Cycle

The two types of cycle described previously both take place in one computer digit time. That is, there is a clock pulse at the beginning and end of the cycle, but none during the cycle (This digit time will be longer than a normal one). Under certain conditions a clock pulse is required in the middle of the store cycle in order to generate data to be written away during the write half of the cycle. Examples of these conditions are as follows:-

(i) If the "write-data" is located in registers A or P, it must first be transferred to the B register before writing into the store.

(ii) If a single logical or arithmetic operation is to be performed on the contents of a store location and the result is to be returned to the same location in store.

The requirement for this type of operation is that there must be a clock pulse in the computer after the read half cycle is completed and before the write half cycle begins. If the write half cycle normally begins less than 1 usec. after the data from the read half cycle is available, then the write half cycle must be delayed to allow the logic operation to take place (ie. whether or not it is delayed depends on the core store speed). Thus, the sequence of events is as follows:-

(1) At the clock pulse time initiating the cycle the Read/Pause flip-flop and Address routing flip-flops are set.

(2) A READ pulse is sent to the core store but the Write Enable is NOT sent.

(3) The gates routing SIN to the B Register are opened and the Clear B pulse is generated as in 5. 3. 1 - (2) on the preceding page.

(4) The data from the store arrives in the B register. The core store then ceases operations.

(5) 1 usec. later, a computer clock pulse is generated (by internal or core store timing).

(6) This clock pulse sends a WRITE pulse to the core store which causes the store to continue the cycle.

(7) This clock pulse also sets the same Address Routing flip-flop as was set during the Read portion so that the address never changes during the Read/Pause/Write cycle.

(8) At the end of the write cycle (again timed either internally or from the store) another clock pulse is generated which resets the control flip-flops and sets those which specify the next operation.

SECTION 7

PARITY CHECK

7.1 PARITY ADDERS

The Parity Adders on drawing 26 generate 5 waveforms (both phases) which describe the parity of the contents of the B register. Waveforms CEP0 to CEP3 describe the parity of characters 0 to 3 respectively of the word held in the B Register. They become active when their respective character has even parity. CEP will become active when the word held in the B register has even parity. These waveforms are used by the Microprogram to make a decision while checking or generating parity.

First, let us look at the least significant character of the word contained in the B Register, character 3 (6 bits, 0 to 5). The character is divided into 3 equal parts (starting from one end of the character) and each part is then examined separately and a decision made regarding its parity.

The common output of gates 1E11/5 and 1E10/5 will be negative only if the parity of bits 0 and 1 is even (all 0's is considered even parity). The common output of gates 1E5/5 and 1E4/5 will be negative only if the parity of bits 0 and 1 is odd. Similar decisions are then made for bits 2 and 3 and bits 4 and 5 by the next 8 gates along the bottom of the drawing.

The common output of gates 1G14/1, 1G14/2, 1G14/4 and 1G14/6 will be positive (hence +CEP3 will be active) if any one of the 4 gates tries to make that point positive. The output of 1G14/1 will be positive if all 3 parts of character 3 have even parity. The output of gate 1G14/2 will be positive if bits 0 and 1 have even parity and the other 2 parts of character 3 both have odd parity. The output of gate 1G14/4 will be positive if bits 2 and 3 have even parity and the other two parts of character 3 both have odd parity. The output of gate 1G14/6 will be positive if bits 4 and 5 have even parity and the other two parts of character 3 both have odd parity. Therefore +CEP3 will become active for all possible conditions in which character 3 has even parity. In a similar manner the common output of gates 1G15/1, 1G15/2, 1G15/4 and 1G15/6 will be active (negative) if the parity of character 3 is even.

Both phases of CEP0, CEP1 and CEP2 are obtained in the same manner as described above for CEP3.

The 8 gates at the right hand side of the page look at each character within the word to decide whether or not the whole word has even parity. Again if any one of these gates is trying to make +CEP active (positive) it will be active regardless of the fact that any one or all of the other gates may be trying to make +CEP inactive (negative).

+CEP will be made active by:-

(1) 2E17/6 if all characters have even parity.

(2) 2E16/6 if characters 0 and 1 have odd parity. and characters 2 and 3 have even parity.

(3) 2E16/1 if characters 0 and 2 have odd parity and characters 1 and 3 have even parity.

(4) 2E15/6 if characters 0 and 3 have even parity and characters 1 and 2 have odd parity.

(5) 2E17/3 if characters 0 and 3 have odd parity and characters 1 and 2 have even parity.

(6) 2E14/1 if characters 0 and 3 have even parity and characters 2 and 3 have odd parity.

(7) 2E14/6 if characters 0 and 2 have even parity and characters 1 and 3 have odd parity.

(8) 2E14/2 if all characters have odd parity.

+CEP is inverted by 2E14/3 to give the opposite phase -CEP.

7.2 PARITY CHECK

On the uppermost portion of drawing Q the parity of the word held in the B Register (bits 0 to 23) is checked against the parity bit held in the Parity Bit flip-flop. If the Parity Bit flip-flop contains a 1 and the parity of the word held in the B Register is even, the output of gate 1C30/5 will be positive. Thus, the waveform -CPTF will be positive (inactive) indicating that there has not been a parity test failure. If the Parity Bit flip-flop contains a zero and the parity of the word held in the B Register is odd, the output of gate 1C30/1 will be positive. If the Parity Bit flip-flop contains a 1 and the parity of the word held in the B Register is odd, both gates 1C30/1 and 1C30/5 will have one positive input and therefore -CPTF will be negative (active) indicating a parity test failure. If the parity bit flip-flop contains a zero and the parity of the word held in the B Register is even, both gates will again have one positive input making -CPTF active.

This waveform -CPTF is then used on the upper right corner of drawing P on the input of gate 2B26/2. The output of this gate will be positive (making +CPF active and causing a parity failure) if:-

(1) Parity checking is not inhibited (+MIPC).

and (2) The Clear Parity flip-flop is not set (+CCP).

and (3) The core store is being routed to the B Register (-CSB).

and (4) The Handswitch Mode switch is not in the Mode 3 position (read everything from the Hand-switches).

and (5) There is a parity test failure (-CPTF).

and (6) The computer is in a read or read/pause cycle.

and (7) The computer is not reading the order with the Handswitch Mode switch not in the Mode 2 position (read Instruction from the Hand-switches).

The waveform +CPF will therefore only become active for parity failures in operations within the central computer. There are however two reasons other than parity failure when +CPF will become active.

Both these are concerned with stopping the clock and are as follows:-

(1) Gate 2C36/5 on the upper right corner of drawing P will make +CPF active when the OFF button on the AC Control Unit is operated. -CSD is a waveform generated by the AC control unit, denoting that the dc power for the central machine and the ac power for the core store are about to be turned off. The function of +CPF is to stop the clock and this is the reason for its use under these conditions. +CRP on the second input to 2C36/5 inhibits the clock being turned off in the middle of a Read/Pause cycle in the core store and thereby prevents loss of information since the Write cycle is not carried out.

(2) On the right hand side of drawing Q gate 2B29/2 will make +CPF active (again to stop the clock) if an illegal order arises while in executive mode and the Clear Parity flip-flop is not set (+CCP).

Parity can be cleared by closing the switch on the engineer's control panel called Clear Parity Fail and setting the flip-flop 2B28/1 on the upper right corner of drawing P. This will make the waveform +CCP active (positive) and since this waveform is an input to the two previously mentioned gates 2B26/2 and 2C29/2 it will inhibit the waveform +CPF thus allowing the clock to run again. The Clock Mode switch should be turned to "One Shot" before operating this switch. When the Clear Mode switch is then turned back to RUN the clock will run and the first clock pulse will reset this flip-flop.

The waveform -CEP generated by the parity adder is also used on the right hand side of drawing T in the parity checker for peripheral transfers.

The output of gates 2A14/2 will be positive if the computer is in a word hesitation (-CWH) and if the parity of the word contained in the B Register is even (-CEP). If under these conditions the contents of the Parity Bit flip-flop RB24 is a zero the common output of gates 2A15/5 and 2A15/2 will try to go negative. This, however, will not be possible until +MQN2 (which is the input of gate 2A15/4) goes positive, since the output of this gate is connected in common with the outputs of the two gates above. +MQN2 will only become positive (active) at the end of the Hesitation microprogram, that is after the information has been received from the peripheral. At this point -BPF can assume a polarity governed by gates 2A15/5 and 2A15/2. The peripheral control which was selected during the hesitation will only look at the -BPF bus at this point in the sequencing.

SECTION 8

RESERVATION CHECK

8.1 GENERAL

The logic for the Reservation Check is located on drawing 28 along with Datum and Limit registers and the associated gating for the Datum register. The function of the Reservation Check is to compare the contents of the upper 9 bits of the N register with the contents of the Datum and Limit registers and produce a signal in accordance with the following law:-

$$L > N \geqslant D$$

where:- L is the contents of the Limit register (9 bits)
N is the contents of the 9 most significant bits of the N register
D is the contents of the Datum register (9 bits)

The result of these two separate comparisons is to make the output of gate 2F19/5 (shown at the centre RH side of drawing 28) positive when the above conditions are satisfied and to make -CDLF active when either comparison fails.

8.2 LIMIT CHECK

The Limit register is considered to contain the upper 9 bits of a 15-bit address, the lower 6 bits of this address are always zero since the Limit must always be a multiple of 64. Thus the contents of the Limit register are compared with the upper 9 bits of the N register to determine if and when a Limit Failure occurs.

The relevant circuitry is shown in the upper half of drawing 28 and is arranged so that the common output of gates 2E34/1, 2F34/6, 2F31/6, etc. becomes positive when the condition L > N is satisfied. This common output line is known as the - LIMIT FAILURE line.

Each Limit bit, with the exception of the least significant, is compared with its N counterpart in two gates which cater for all possible conditions. Consider the most significant comparison, ie. between L8 and N14, and refer to the LH side of drawing 28. Normal signals are applied to gate 2E34/1 and inverse signals are applied to 2F32/1 producing results as listed in the table below.

Note that the first two possibilities define the status of the Limit check with no ambiguity and regardless of the remaining bits of L and N. On the other hand, when L8 = N14 the comparison must be extended to the next significant bits (L7 and N13) and if these are also equal then the comparison is further extended - through to L0 and N6 if necessary.

Reverting to the first condition, when L8 is greater than N13, the basic law is satisfied and gate 2E34/1 drives the - LIMIT FAILURE line positive. When N13 exceeds L8, gate 2F32/1 is enabled and applies a positive input to 2F34/6, 2F31/6, 2F25/6, etc. to override comparisons between less significant bits and ensure that the - LIMIT FAILURE line is made active.

L8	N14	2E34/1		2F32/1		Remarks:
		Inputs	Output	Inputs	Output	
1	0	- -	+	+ +		L greater than N, as required
0	1	+ +		- -	+	Limit Failure, N exceeds L
0	0	+ -		- +		L = N, indeterminate condition which requires reference to next significant bit
1	1	- +		+ -		

8.2 continued

When L8 = N13, both 2E34/1 and 2F32/1 are inhibited and the next comparison assumes primary importance. Thus, if L7 is greater than N13, gate 2F34/6 is enabled and if N13 exceeds L7, gate 2F32/2 is enabled to override less significant comparisons and make - LIMIT FAILURE active. Similar pairs of gates are provided for each subsequent comparison down to L1 and N7. The least significant comparison, between L0 and N6, requires only one gate (1F28/6) since the only logical decision required is whether or not L0 is greater than N6. In other words, the condition L0 = N6 is treated the same as when N6 exceeds L0, ie. a Limit check failure.

For convenience, the more significant "fail" gates are connected in pairs. Thus, the outputs of 2F32/1 and 2F32/2 are combined, similarly 2F31/1 and 2F25/1 are combined - the overall effect is to reduce the number of inputs on the less significant "go" gates.

One further condition requires an explanation, as follows. The maximum Limit, which is 32,768, may be specified and in this case the Limit register will contain all zeros. Since it is impossible to have a Limit failure when this maximum value is specified, additional circuitry is included to ensure that -CDLF remains positive (inactive) if L0 to L8 are all zero. This is effected by the combination of 2C13/3 and 2C14/2 which inhibits 2F19/2 when an all-zero condition occurs.

Finally, if a Limit failure occurs, gate 2F19/2 will be enabled and force the output of gate 2F19/5 negative thus making -CDLF active.

8.3 DATUM CHECK

The logic for the Datum check is similar to the logic for the Limit check. Hence, the common output of gates 2E35/6, 2F35/6, 1F32/6 and 1F26/6 will be negative if the content of the Datum register is greater than the content of the upper 9 bits of the N register. Conversely, this line will go positive if N is greater than or equal to D, (ie. a Datum failure). In this case the input to gate 2F19/5 will be positive and again the output will be forced negative making -CDLF active.

SECTION 9

HESITATION CONTROL

9.1 GENERAL

When a Peripheral Control wishes to access the working store to extract or insert a single word or character, it sends a Hesitation Request to the central computer. When this occurs the computer is made to hesitate at the next convenient point in the sequencing. The Hesitation Microprogram is then entered, and Hesitation Control processes the Request, sends back a Hesitation Select to the Peripheral Control, and the appropriate transfer of information takes place. The logic for Hesitation control is shown on drawings V, W, X and Y.

9.2 HESITATION REQUESTS

The Hesitation Request lines (+PHR..) from all Peripheral Controls are divided into blocks of 10 in Hesitation Control. Each drawing Y constituting one block (eg. Y0 is block 0 and Y1 is block 1, etc.). Within each block all Hesitation Request signals from slow Peripheral Controls are mixed to give a waveform -CHRB. The -CHRB waveform from each block is then mixed on drawing V to give a signal CSLHR. The Hesitation Request lines from all fast Peripheral Controls are mixed on drawing V to give a waveform CFHR. These two waveforms CFHR and CSLHR are used to tell the Microprogram that a fast or slow Hesitation Request is waiting to be processed. The computer will then hesitate at the next convenient point and enter the Hesitation Microprogram (see section on Microprogram) at which time the +CHM waveform (in the upper left corner of drawing V) will become active.

9.3 BLOCK RACE

Separate Hesitation Requests may be originated by several different Peripheral Controls during the interval preceding the computer's hesitation. However, only one such request can be accommodated at any time therefore the several Hesitation Request signals compete with each other to obtain priority. First, the various peripheral blocks compete in the "block race" and then the individual peripherals of the selected block compete with each other. The selection of any given block inhibits all other blocks and, similarly, the selection of any individual peripheral inhibits all other competing waveforms.

The Block Race is started when +CHM becomes active and generates the signals -CESF and -CESS shown in the upper LH corner of drawing V. Note that -CESF precedes -CESS in time owing to the inclusion of the two extra gates in the latter circuit. Hence, slow Hesitation Requests are handicapped in the Block Race and fast peripherals receive priority.

Thus, the Block Race is started by -CESF, followed almost immediately by -CESS. These signals, together with the various -CHRB waveforms, are applied to gates, as shown on drawing W, to select one of the blocks. However, the combination -CESF and -CFRB which occurs when a fast Hesitation Request is active, wins the Block Race and makes +CSF active. This waveform inhibits all other blocks from being selected and starts the race between the various fast peripherals which results in the selection of one +BHS output.

In the event that no fast Hesitation Request is received but two or more slow Hesitation Requests are active, the Block Race will be confined to selecting the first +CSB waveform which becomes active. This waveform inhibits all the gates producing other +CSB waveforms and also +CSF. The selected +CSB waveform is then applied together with -CESS to start the race between individual slow peripherals within the relevant block - see drawing Y. Again, one (and only one) +BHS waveform becomes active and the resultant signal (Hesitation Select) is sent to the relevant Peripheral Control. Note that there is a +BHS.. waveform corresponding to each +PHR.. waveform.

9.4 HESITATION SELECT

The single +BHS.. waveform which will be selected during any one Hesitation is used on drawing X to generate the address of the control word associated with that particular Peripheral Control. This address is then used by the Hesitation Microprogram (see section on Microprogram).

All +BHS.. waveforms associated with word-at-a-time peripherals are mixed on drawing V (under Word Mix) to give a waveform CWH. All +BHS waveforms associated with input type peripherals are also mixed on drawing V (under Input Mix) to give a waveform CINH.

CWH is used by the Hesitation Microprogram to determine whether the Hesitation it is going to perform is a word or a character type Hesitation (if it is not a word type it must be a character type). CINH is similarly used by the Hesitation Microprogram to determine whether it is an input or an output type Hesitation (if it is not an input type it must be an output type).

9.5 PARITY

During output hesitations the logic on the right hand side of drawing T also generates the odd parity bit (-BOUT 24) which is sent to the Peripheral Control along with the information being transferred.

If the computer is in a word hesitation and the parity of the word contained in the B Register is even, the output of gate 2A14/2 will go positive, putting a positive input on gate 2A14/5 and making -BOUT 24 negative (active).

If the computer is not in a word hesitation (ie. is in a character hesitation), the character flip-flops specify character 0 (that is the most significant character of the word contained in the B Register is being sent out to a Peripheral Control) and the parity of this character is even, the output of 2A16/2 will go positive and once again -BOUT 24 will be negative (inactive). A parity bit for characters 1, 2 and 3 will be generated in a like manner when applicable.