PRELIMINARY DESCRIPTION

Of the

UNIVAC

PRELIMINARY DESCRIPTION OF THE UNIVAC

for EMCC personnel only

1. Introduction

A thorough grasp of the detailed functioning of the UNIVAC depends upon understanding the interrelationships of its various components. For this purpose, a short demonstration routine embodying all instructions will be introduced into the computer, and its execution will be traced, step by step. The level of the treatment is indicated by the detail of Figure U-100, Simplified Block Diagram of UNIVAC.

The reader should be acquainted with the instruction code, and have a general knowledge of the primary purposes of each of the principal units of the UNIVAC System: the UNIVAC (Central Computer), Supervisory Control, UNISERVO, UNIPRINTER, and UNITYPER.

The System is assumed to be energized and ready to receive a problem. The demonstration routine has been placed on a magnetic tape and this tape is mounted on UNISERVO No. 1. The Clock and Cycling Unit (CU) provide the timing signals required to control the sequencing of the computer.

2. Initial Read Operation

The problem is started from the Supervisory Control by setting, the Initial Tape Selector Switch (not shown) to "1", and depressing the Initial Read Start Button. The next appropriate signal from the Cycling Unit activates the Initial Read Circuits. The Initial Read Timer steps the Initial Read Switch (IRS) to its first position. This prepares the computer to receive data from the UNISERVO. When the IRS is stepped by the IR Timer to the next position, the first 60 words from UNISERVO No. 1 are transferred over the Read Bus, through the Input Synchronizer, to Register I. The Input Synchronizer accumulates the words at the comparatively slow rate of the tape and transfers them to Register I at the much faster speed of the computer. When the sixtieth word has been transferred to Register I, the IRS is stepped to its next position. Next the block of 60 words is transferred from Register I over High Speed Bus No. 1 (Input) (HSB1 I), through the High Speed Bus Amplifier, over High Speed Bus No. 2 (memory) (HSB2M), to the first 60 locations in the memory. The Odd-even Checker counts the pulses of each digit as it passes, and stops the computer if an even count is registered for any digit. The IRS is now stepped to its final position, which starts the computer on its normal cycle of operations.

3. The Four-Stage Cycle of Operation

Each instruction word contains a pair of instructions, called the left and right instructions. The computer executes first the left, and then the right. Prior to executing the instruction-pair, the memory location containing the word must be determined, the word extracted from the memory, and placed in a special register, the Control Register (CR).

Thus, four events must occur in sequence: (α) determination of the memory location of the instruction word, (β) extracting this word from the memory and transferring it to CR, (γ) executing the left instruction, and (δ) executing the right instruction. This sequence is followed until the computer is stopped by an error signal or by a stop instruction.

The Greek letters α, β, γ, δ indicate the time intervals or cycles during which the operations listed above take place. A special counter called the Cycle Counter (CY) keeps track of the currect cycle. CY has two binary stages, and thus can count to four. The start circuit provides a signal which clears CY to α. The termination of each operation produces an ending pulse (EP) which steps CY to the next state.

When CY is in the α-state, signals are sent to the Function Table (FT) which cause the memory location number contained in the Control Counter (C) to be transferred over HSB1A to the HSB amplifier and thence over HSB2A to CR and through a delay line to the Static Register (SR). CY is advanced to the β-state by an ending pulse.

The three decimal digits of the desired memory location are set up in the right part of SR, early in β-time, and control the Memory Switch. The tens and hundreds digits select the desired ten-word memory channel. To obtain the desired word in the group of ten, use is made of the Time Selection Counter (TSC). This counter runs continuously, counting from zero to nine and repeating. Its reading corresponds to the number of the word next to appear at the output of the mercury delay line. This is the only word in the group which may be extracted at that particular instant. The units digit in SR is compared with TSC, and when coincidence occurs, a special Time Selection (TS) pulse permits the word to be transferred from the memory over HSB1M to the HSB amplifier and thence over HSB2A to CR.

The Control Counter reading is routed to the adder, while the instruction word is being extracted from the memory. A unit is added to the previous memory location number, providing the number of the next memory location in the normal sequence, which is returned to C. An ending pulse then advances CY to the γ -state.

The left instruction of the pair circulating in CR was sent to SR at the end of β-time. During γ-time, the two instruction digits control the Function Table to initiate the desired operation. The three memory location digits control the Memory Switch, to extract the data word which is to be operated upon. At the conclusion of the operation, an ending pulse advances CY to the δ -state.

The right instruction passes to SR at the end of γ -time. During δ-time, the instruction digits control the Function Table and the memory location digits control the Memory Switch as described for the left instruction. When the operation prescribed by the right instruction has been executed, an ending pulse steps CY to the α -state and the four-cycle sequence is repeated with the next pair of instructions. Instructions may be provided, which alter the normal sequence of memory locations containing instructions, thus permitting choices, iterative routines, etc.

The four-cycle operation of the UNIVAC is summarized in the following table:

Cycle	CY	Description
	00	Read out of C to CR. Transfer CR to SR. Give ending pulse.
	01	Read out of memory to CR. Transfer left instruction to SR. Increase C by unity. Ending pulse associated with TS signal.
	10	Execute (left) instruction set up to SR. Transfer right instruction to SR. Ending pulse associated with instruction executed.
	11	Execute (right) instruction set up in SR. Ending pulse associated with instruction executed.

4. Data Transfers

The first twelve instructions are contained in memory locations 000 through 005. These illustrate the various ways of transferring information from one memory area to another. The following quantities have been introduced into the memory and with the instructions constitute the block of 60 words now in the first 60 memory locations:

Memory Location	Quantity	Memory Location	Quantity
041	x_0	048	z_0
042	y_0	049	z_1
043	y_1	050-059	c_0---c_9

The twelve instructions will (a) transfer x_0 to memory locations 201 and 301, (b) interchange y_0 and y_1 with z_0 and z_1, and (c) transfer c_0---c_9 from 050-059 to 210-219.

Memory Location	Instruction			
000	B 041		$x_o \rightarrow$ rA and rX	
		L 042	$y_o \rightarrow$ rL and rX	
001	F 043		$y_1 \rightarrow$ rF	rA unchanged
		V 048	z_o, $z_1 \rightarrow$ rV	y_o, y_1 inter-
002	W 042		z_o, $z_1 \rightarrow$ 042, 043	changed with
		J 048	$y_o \rightarrow$ 048	z_o, z_1
003	G 049		$y_1 \rightarrow$ 049	
		H 201	$x_o \rightarrow$ 201	
004	C 301		$x_o \rightarrow$ 301; 0 $\rightarrow$ rA	
		K 000	0 $\rightarrow$ rL	
005	Y 050		$c_o \ldots c_9 \rightarrow$ rY	
		Z 210	$c_o \ldots c_9 \rightarrow$ 210...219	

Returning to the Control Circuits, the stimulation of the start circuit by the Initial Read Switch clears CY to α. The Control Counter is registering 000, the first memory location. During α-time, this memory location is transferred through CR to SR. During β-time, the instruction pair B 041 - L 042, located at 000 in the memory (word 0 of channel 00), is transferred to CR and the left instruction (B 041) to SR. A unit is also added to the number in C, so that C contains 001 at the end of β-time. During γ-time, the memory location digits "04" select channel "04' at the Memory Switch, and the digit "1" is compared with TSC until TSC reads "1". The TS signal, in conjunction with the FT signals called for by the instruction digit B, now close the clear gate of A, open the memory output gate and the input gate of A. The previous contents of A are cleared out and simultaneously x_o is transferred from memory channel "04" to A. As soon as the transfer has been completed, the Control Circuits emit a signal called "Time Out" (TO), which closes the input gate to A, and opens the clear gate in A to permit x_o to circulate in A until needed. The TO signal lasts for one minor cycle, to allow time for the FT signals of the next instruction to build up and the old signals to die out. (A minor cycle is the time required for a word to pass a given point in the computer.)

The ending pulse from the control circuits steps CY to δ, and the right instruction (L 042) is executed in a similar manner. This time the TS signal, which initiates the transfer of y_0, is given when TSC reads "2". The clear and input gates of Register L are operated by TS and the FT signals are produced by the instruction digit "L" in SR.

In addition to the operations described, the instruction digits B and L also cause a quantity on the HSB2A to be read into Register X, after its previous contents have been cleared. Thus, at the end of γ -time, the quantity x_0 is circulating in both Registers A and X. At the end of δ -time, the quantity y_0 is circulating in both Registers L and X, whereas x_0 is now circulating in A only, X having been cleared prior to the read-in of y_0. CY is now in the α-state.

The four-cycle sequence is now repeated with the second instruction pair (F 043 - V 048) from memory location 001. At the end of β -time, this instruction word is circulating in CR, the left instruction (F 043) has been sent to SR, and C reads 002. The quantity y_1 is read out of the memory into Register F during γ -time, as soon as TSC reads "3". The F instruction does not provide a read-in to X.

The right instruction (V 048) is executed during δ-time. This instruction reads two successive words from the memory into Register V. Since a basic operation is limited to a single word transfer, the Program Counter (PC) is required when more than one basic operation is needed in the execution of an instruction. The "V" instruction digit in SR in conjunction with PC controls the Function Table to permit two words to be transferred. In this case, the TO signal, which terminates the operation, is delayed by FT until the words in positions "8" and "9" of channel "04" have passed onto HSB1M through the memory output gate. Likewise the input gate of V is held open for two minor cycles.

As the result of the execution of the first four instructions, x_0 is circulating in Register A, y_0 in L and X, y_1 in F, and the pair of words z_0, z_1 is circulating in V. All of these words are also circulating in their original places in channel "04" of the memory.

The instruction word (W 042 - J 048) from memory location 002 is placed in CR and the two instructions executed during the next four cycles α, β, γ, and δ. C is augmented to 003. The instruction, W 042, in conjunction with PC, allows the two words, z_0 and z_1, to be transferred into channel "04" of the memory as soon as the TS signal is given. (TSC reading "2"). "J 048" causes y_0 in X to be placed in memory channel "04" as word "8". As any word is read into a memory location, the previous contents are cleared.

The instruction pair (G 049 - H 201) from memory location 003 brings about the transfer of y_1 from Register F to memory channel "04" as word "9", and the transfer of x_0 from A to memory channel "20" as word "1". This completes the inter change of y_0, y_1 with z_0, z_1 and the transfer of x_0 from 041 to 201.

Memory location 004 yields C 301 - K 000, as the instruction pair next to be executed. The "C" instruction digit operates exactly as the "H" instruction digit, except that it introduces decimal zeros (symbol "z") from the Cycling Unit into Register A as z_0 is passed onto HSBIA. When the "K 000" instruction is carried out, the memory gates are kept closed, so that word "0" in channel "00" is not passed to HSBIM. The "K" instruction digit opens the output gate of Register A, the input gate of Register L, thus permitting the contents of A to pass into L, and also introduces z into A from CU to replace the former contents of A. Thus, registers A and L now both contain decimal zeros, and x_0 is circulating as word "1" in both channels "20" and "30" of the memory.

The final instruction pair (Y 050 - Z 210) of this group carries out a ten-word transfer. "Y 050" causes the entire channel "05" to be transferred to Register Y. The Program Counter is again used to provide the TO signal at the end of the ten-word transfer. Again each word continues to circulate in memory channel "05" in its former place in addition to entering Y. "Z 210" causes the ten words in Y to be transferred to memory channel "21". The Control Counter now contains 006, and next group of instructions is ready to be carried out.

5. Arithmetic Operations

The second group of 22 instructions illustrates the execution of the several types of algebraic addition and the four shift instructions. Data words are now located as follows:

Memory Location	Quantity
041	x_0
042	z_0
043	z_1
048	y_0
049	y_1

The addition instructions are to solve the equation:

$$\left| x_0 + y_0 - 2z_0 \right| = w_0$$

(It is assumed that x_0, y_0 and z_0 are all positive quantities.) The four shift instructions are introduced to form the absolute value of the sum and to round-off the sum to eight significant digits.

Memory Location	Instructions		Remarks
006	B 039		Preparation for possible overflow in the
		C 000	addition operation
007	B 041		$x_o \longrightarrow rA$
		*A 048	$rA = x_o + y_o$; if overflow go to 000
008	S 042		$rA = x_o + y_o - z_o$
		X 000	$rA = x_o + y_o - 2z_o$
009	;1 000		Shift (rA) one place left, dropping sign
		.3 000	Shift (rA) three places right
010	A 038		Add round-off
		.1 000	Shift (rA) one more place right
011	;3 000		Shift (rA) three places left to reposition after round-off
		C 044	$(044) = w_o = \lvert x_o + y_o - 2z_o \rvert$ with round-off
000	-1 000		Overflow subroutine: Shift (rA) one-place right
		U 033	Transfer control to 033
033	A 037		Add one unit in MSD position
		C 044	$(044) = (x_o + y_o)/10$
034	B 042		$(rA) = z_o$
		-1 000	Shift (rA) one place right
035	C 042		$(042) = z_o/10$
		B 044	$(rA) = (x_o + y_o)/10$
036	00 000		
		U 008	Transfer control to 008
037	010000	000000	Unit in MSD position
038	000000	000005	Round-off correction
039	-1 000	U 033	Overflow instruction

6. Overflow

Should overflow occur in the summing operation, the computer will automatically insert the pair of instructions located in the first memory location (000). Consequently, an instruction pair to initiate the desired corrective action must be placed in 000, prior to performing an addition.

The instruction pair (B 039 - C 000) from 006 transfers storage (039) to 000 the instructions to initiate corrective action in case of overflow. The left instruction (B 041) from 007 brings x_0 into Register A. The right instruction (*A 048) directs the addition of y_0 to x_0, the sum being placed in A. The asterisk indicates the possibility of overflow.

7. Addition

The add instruction (A) is executed in several steps. First y_0 is transferred from 048 to X. The TO signal from the Control Circuits, which terminates the transfer, steps the Program Counter rather than the Cycle Counter. While the next set of control signals from the Function Table (initiated jointly by PC-2 and "A") are building up, the quantities x_0 in A and y_0 in X are transferred simultaneously to the Comparator (CP) over special paths. Here the relative magnitudes and the signs of x_0 and y_0 are compared. The Comparator sends control signals to the Adder which appends the sign of the larger magnitude to the sum, and actuates the complementer if the smaller magnitude has a sign opposite to that of the larger. During the second step, x_0 from A and y_0 from X are sent via special paths to the Adder, and the sum is returned to A, again by a special path. The ending pulse then advances CY to begin the next cycle.

Should a carry occur when the most significant decimal digits are added, a special overflow signal (OF) operates during the next succeeding α-time to admit decimal zeros to HSBIA, instead of the memory location contained in C. Thus, during β-time, the instruction pair from 000 (now -1 000 U 033) is routed to CR, rather than that from 008 (the present reading of C). Further, the quantity in C is not augmented by a unit, as would normally occur.

8. Overflow Routine

The instruction "-1 000" will cause the decimal digits in A, except the sign digit, to be shifted one place right. This is accomplished by routing the word through a delay path which takes one digit-time less than the normal re-circulation path. The least significant digit is dropped, and a decimal zero is supplied from the Cycling Unit for the most significant digit of the shifted word. The sign is routed over the normal path, and thus remains in its proper position.

The instruction "U 033" interrupts the normal sequence of instructions in the following manner: During δ-time, the Function Table signals set up by "U" transfer the three memory location digits (here "033") from CR to C, after clearing C. Thus, the next instruction pair is taken from 033, rather than from 008, the previous number in C.

The instruction pair (A 037 - C 044), from 033, causes a "one" in the MSD position to be added to the sum in A, thus supplying the digit lost in the addition producing the overflow. The instruction C 044 then transfers the corrected sum to 044 for temporary storage.

In order that the decimal points may be aligned, the quantity, z_0, which is to be subtracted from the sum $x_0 + y_0$ must be shifted right one place. This is accomplished by the next three instructions, "B 042", "-1 000" and "C 042". Finally, the sum, $x_0 + y_0$, must be returned to Register A, duplicating the conditions existing if no overflow had occurred. The instruction "00 000" is called the "skip" instruction, as its only effect is to supply the ending pulse which advances CY to the next state thus "skipping" to the next instruction. "U 008" transfers control back into the main routine, at the point at which the overflow occurred. Since control transfers are effective only at α-time, all control transfer instructions must be righthand instructions. Thus, the "skip" was required to fill in the lefthand instruction. Register A now contains the sum $x_0 + y_0$, and if overflow occurred the original sum was shifted one place right, with a "one" inserted in the new MSD position.

9. Subtraction

The instruction pair (S 042 - X 000), from 008, is now executed. The "S" instruction is carried out in exactly the same manner as the "A" instruction, with the single exception that the sign of the quantity is reversed as it enters register X on the first step. The "X" instruction utilizes the same procedure as the "A" instruction, except that the first step is omitted. Nothing is read from the memory, but the comparison of magnitudes and signs of X and A by CP, and the subsequent addition of the quantities in X and A, are carried out. Register A now contains the sum: $x_0 + y_0 - 2z_0$, since "$-z_0$" was twice "added" from X to A.

10. Shifts

The absolute value of the quantity in A is obtained by the simple expedient of shifting <u>all</u> digits of A one place left. Thus, the sign is "dropped off" to the left. This is accomplished by the ";1 000" instruction, which introduces an extra delay of one digit-time in the circulation path of A, and inserts a decimal zero into the LSD position. The ".3 000" instruction shifts <u>all</u> digits of A three places to the right and inserts decimal zeros in the sign position and the two following positions. This is accomplished by routing the entire word in A over the one-digit-time-shorter path for three successive circulations. The Program Counter is used to control this shift, which is terminated when the reading of PC agrees with the second instruction digit (here "3").

The word from 010 (A 038 .1 000) adds a "5" to the LSD position of the sum in A, producing a carry to the next decimal position if the LSD of the sum was "5" or greater (round-off). The second instruction then shifts the result one more place right. Thus, the result of the round-off addition, except for the possible carry, is dropped.

The final instruction pair (03 000 - C 044), brings the sum, w_0, back to the original decimal point position (if overflow did not occur), and transfers w_0 to memory location 044. The shift instruction ";3 000" might just as well have been used here, since the left-most four digits, prior to the left shift, were the decimal zeros inserted by the two previous right shifts. The "On 000" instruction shifts all digits, except sign digit, "n" places to the left.

11. Multiplication

There are three instructions for accomplishing a multiplication: M, N and P. The N instruction differs from the M instruction only in that the sign of the multiplier is reversed at the beginning of the operation. The P instruction is unique in that it omits the round-off correction, which is supplied by the M and N instructions and retains the entire 22-digit product. In the following example, the M instruction only will be illustrated, and the differences between it and N and P will be pointed out at the conclusion.

A complete multiplication involves three instructions: (1) transferring the multiplicand from the memory to L, (2) transferring the multiplier to X and initiating the multiply operation which delivers the product to A, and (3) transferring the product from A to memory storage. In the example chosen, x_0 will be the multiplicand (abbreviated "icand"), y_0 will be the multiplier ("ier") and the product will be delivered to 045. The three instructions, therefore, become:

Memory Location	Instruction		Remarks
012	L 041		$x_0 \Rightarrow rL$ (x_0 = 0.11111 111111)
		M 048	$y_0 \cdot x_0 \rightarrow$ (y_0 = -.12345 678901)
013	C 045		(rA) $\rightarrow$ 045 ((rA) = -.01371 742100)

Table 2 shows, at the end of each minor cycle, the contents of each register or counter concerned with the multiply operation. The Program Counter (PC) is used to count the steps of the process. F contains the absolute value of three times the multiplicand (on and after step 4). L contains the multiplicand. The product $3 \cdot |\text{icand}|$ is built up in A during the first three steps for transfer to F. The round-off correction is placed in A on step 4. Thereafter A contains the partial product as it is built up on steps 5 through 15. X initially receives the multiplier, and on steps 5 through 15 transmits the new multiplier digit to the Multiplier-Quotient Counter (MQC), while receiving the LSD of the old partial product from A. MQC receives the complement of the new multiplier digit at the end of each step. This digit is increased to zero each step by successive additions as the multiplies of the "icand" are added. CP receives the signs of "ican" and "ier" on step 3, stores the appropriate sign of the product from steps 4 through 15, and supplies this sign to A and X on step 15.

Step 1 requires a variable number of minor cycles, depending upon the position of the multiplier word in the memory channel, and is terminated at the end of the TS minor cycle, leaving "ier" in X. Also on this step, A is cleared to decimal zeros, which, along with the absolute value of the "icand" from L, are sent over special paths to the Adder, the sum being returned to A. Steps 2 and 3 repeat the addition of $|\text{icand}|$ to the contents of A, thus building up $3 \cdot |\text{icand}|$ in A.

The contents of A are transferred over the HSB to F on step 4. Register A is then cleared to decimal zeros, which, along with the round-off correction (0.50000 000000), are sent over special paths to the Adder. The sum is returned to A. The "ier" is sent to MQC, where the nine's complement of the LSD of "ier" (here "-1") is set up. All digits of the "ier", in X, are shifted one place right, a decimal zero being supplied to the sign position.

Table 2 - Multiplication Example

PC	Min Cyc	F	L	A	X	MQC	CP
		xxxxxx xxxxxx	011111 111111	xxxxxx xxxxxx	xxxxxx xxxxxx	x	x
1	1 TS			000000 000000 011111 111111	 -12345 678901		
2	TO 1			 022222 222222			
3	TO 1			 033333 333333			
4	TO 1	 033333 333333		 000000 000000	 -1234 567890	 0	 –
5	TO 1 IER		 (<3)	050000 000000 061111 111111 006111 111111	 1-123 456789	-1 0 0	
6	IER			000611 111111	11-12 345678	-9	
7	1 2 3 IER	(>3) (>3) (>3)		033944 444444 067277 777777 100611 111110 010061 111111	 011-1 234567	-6 -3 0 -8	
8	1 2 3 4 IER	(≥3) (≥3)	 (<3) (<3)	043394 444444 076727 777777 087838 888888 098949 999999 009894 999999	 9011- 123456	-5 -2 -1 0 -7	
9	1 2 3 IER	(≥3) (≥3)	 (<3)	043228 333332 076561 666665 087672 777776 008767 277777	 69011 -12345	-4 -1 0 -6	
10	1 2 IER	(≥3) (≥3)		042100 611110 075433 944443 007543 394444	 36901 1-1234	-3 0 -5	
11	1 2 3 IER	(≥3)	 (<3) (<3)	040876 727777 051987 838888 063098 949999 006309 894999	 93690 11-123	-2 -1 0 -4	
12	1 2 IER	(≥3)	 (<3)	039643 228332 050754 339443 005075 433944	 39369 011-12	-1 0 -3	
13	1 IER	(≥3)		038408 767277 003840 876727	 73936 9011-1	0 -2	
14	1 2 IER		(<3) (<3)	014951 987838 026063 098949 002606 309894	 97393 69011-	-1 0 -1	
15	TO 1 IER		 (<3)	 013717 421005 -01371 742100	 -59739 369011	 0 (-)	

MQC produces one of three signals, depending on the value of the digit complement it contains. If the digit is the complement of 3 or greater, the "≥3" signal is produced; if the digit is the complement of 1 or 2, the "<3" signal is generated; and if the digit is decimal 0, the "IER" signal is produced. The "≥3" signal, for each minor cycle during which it is present, causes the contents of F (3· |icand|) to be added to the partial product in A, and also admits three pulses to step MQC three times, thus increasing by three the digit complement stored in MQC. The "<3" signal, for each minor cycle during which it is present, causes the contents of L, less sign (|icand|), to be added to the partial product in A, and admits one pulse to step MQC. The "IER" signal, present for one minor cycle only, causes the contents of both A and X to be shifted one place right, the LSD from A becoming the MSD of X, and the LSD from X being sent to MQC.

The example should be traced, minor cycle by minor cycle, until the process controlled by the three signals: "≥3", "<3" and "IER" is understood. It should be noted that the "0" multiplier digit requires one minor cycle; a "1" or "3", two minor cycles; a "2", "4" or "6", three; a "5", "7" or "9", four; and an "8", five minor cycles per step. Multiplication by a null quantity thus requires a minimum of 21 minor cycles.

On the final step (15), the sign of the product, which has been stored since step 4 in CP, is affixed to the rounded product in A and to the modified low-order product digits in X. The TO signal precedes this step in order to allow time for the FT signals, causing the sign transfer, to rise to their proper value. Should the final multiplier digit be "0", the IER minor cycle of step 15 would follow immediately upon the corresponding cycle of step 14, and the sign transfer might not take place.

The N instruction differs from the M instruction of the above example only in that it causes the sign of the multiplier to be reversed upon the read-in to X on step 1. The P instruction differs only in that the addition of the round-off correction in step 4 is omitted. Thus, a 22-digit product is correctly registered in A and X, containing respectively the high-order and lower-order portions of the product.

12. Division

A complete division involves three instructions: (1) transferring the divisor from the memory to L, (2) transferring the dividend to A and initiating the divide operation which delivers the quotient to A, and (3) transferring the quotient from A to memory storage. The example chosen is that of dividing the product obtained in the multiplication example by x_0, yielding y_0 as the quotient. The three instructions are:

Memory Location	Instruction		Remarks
013		L 041	$x_0 \rightarrow rL$ (x_0 = 0.11111 111111)
014	D 045		$(045)/x_0 \rightarrow rA$ ((045) = -.01371 742100)
		C 046	$rA \rightarrow 046$ ((rA) = -.12345 678900)

It should be noted that x_0 was left in L at the conclusion of the multiplication, and hence it was not necessary to use the "L 041" instruction of line 013.

The computer employs the non-restoring method of division. In this method, the divisor is first subtracted from the shifted dividend until an overdraft occurs. The number of subtractions, not counting the one which produces overdraft, becomes the first (MSD) quotient digit. The dividend is again shifted left, and the divisor is added back until overdraft again occurs. In this case, the nine's complement of the number of additions, less the one producing the overdraft, becomes the quotient digit. The sequence of alternate subtractions and additions is continued until 12 quotient digits have been determined. A round-off correction is added to the quotient, which is then shifted right to become the rounded 11-digit quotient.

Table 3 shows, at the end of each minor cycle, the contents of each register or counter concerned with the divide operation. The steps are counted by PC. A Binary Counter is used to keep track of the alternations of subtraction and addition. L contains the divisor. Register A initially contains the dividend, and thereafter registers the result of each subtraction or addition as the operation proceeds. Register A also receives the 12-digit quotient on step 15, and the rounded 11-digit quotient on step 16. Register X receives a quotient digit from MQC, in the LSD position, at the end of each step (3 through 14), after the previous contents of X have been shifted one position left. MQC is cleared to decimal zero at the start of each step of the division. It is then stepped once for each subtraction or addition except the one which produces overdraft. The Binary Counter so controls the read-out of the quotient digit from MQC to X, that the digit is read out directly following subtraction steps, whereas the nine's complement is read out following addition steps. The Comparator serves the same function as in multiplication—it determines and stores the sign of the quotient until needed at step 16 for A and X.

The Binary Counter is initially set to 1, which produces a signal causing the Adder to subtract the absolute value of the divisor (from L) from the dividend (from A). Since the Adder "subtracts" by complementing the subtrahend and adding, a carry from the twelfth digit position is obtained whenever the difference has the sign of the minuend. Hence, the absence of this carry indicates an overdraft. The remainder will be in complement form, indicating a negative quantity. When the divisor is added back to the shifted remainder, a carry will eventually be produced, which then becomes the overdraft signal on addition. The overdraft signal from the Adder is used to produce the "OR" signal in the Multiply-Divide Circuits. The "OR" signal serves the same purpose in the division as does the "IER" signal in the multiplication—that of terminating the step and preparing for the next step. The "OR" signal thus reverses the output of the Binary Counter, shifts the contents of both A and X one place left, causes the quotient digit to be inserted into the LSD position in X, and clears MQC to decimal zero.

The 12-digit quotient is registered in X at the end of step 14. On step 15, this quotient is transferred from X to A. After clearing A, the round-off correction (000000 000005) is added, and the sum returned to A. On step 16, the rounded quotient in A and the unrounded quotient in X are both shifted right one place, and the sign held in CP is inserted into both quotients. An ending pulse is produced on step 16 which terminates the operation.

Table 3 - Division Example

PC	Min Cyc	Bin Ctr	L	A	X	MQC	CP
			011111 111111	xxxxxx xxxxxx	xxxxxx xxxxxx		
1	1	1				0	
	TS			-01371 742100			–
2	TO						
	1			013717 421000			
3	TO						
	1			002606 309889			
	2			*991495 198778		1	
	OR			914951 987780	xxxxxx xxxxx1		
4	1	0		926063 098891		-9	
	2			937174 210002		-8	
	3			948285 321113		-7	
	4			959396 432224		-6	
	5			970507 543335		-5	
	6			981618 654446		-4	
	7			992729 765557		-3	
	8			*003840 876668		-2	
	OR			038408 766680	xxxxxx xxxx12		
5	1	1		027297 655569		0	
	2			016186 544458		1	
	3			005075 433347		2	
	4			*993964 322236		3	
	OR			939643 222360	xxxxxx xxx123		
6	1	0		950754 333471		-9	
	2			961865 444582		-8	
	3			972976 555693		-7	
	4			984087 666804		-6	
	5			995198 777915		-5	
	6			*006309 889026		-4	
	OR			063098 890260	xxxxxx xx1234		
7	1	1		051987 779149		0	
	2			040876 668038		1	
	3			029765 556927		2	
	4			018654 445816		3	
	5			007543 334705		4	
	6			*996432 223594		5	
	OR			964322 235940	xxxxxx x12345		
8	1	0		975433 347051		-9	
	2			986544 458162		-8	
	3			997655 569273		-7	
	4			*008766 680384		-6	
	OR			087666 803840	xxxxxx 123456		

* Indicates overdraft

Table 3 - Division Example (Cont.)

PC	Min Cyc	Bin Ctr	L	A	X	MQC	CP
9	1	1	(011111 111111)	076555 692729	(xxxxxx 123456)	0	(-)
	2			065444 581618		1	
	3			054333 470507		2	
	4			043222 359396		3	
	5			032111 248285		4	
	6			021000 137174		5	
	7			009889 026063		6	
	8			*998777 914952		7	
	OR			987779 149520	xxxxx1 234567		
10	1	0		998890 260631		-9	
	2			*010001 371742		-8	
	OR			100013 717420	xxxx12 345678		
11	1	1		088902 606309		0	
	2			077791 495198		1	
	3			066680 384087		2	
	4			055569 272976		3	
	5			044458 161865		4	
	6			033347 050754		5	
	7			022235 939643		6	
	8			011124 828532		7	
	9			000013 717421		8	
	10			*988902 606310		9	
	OR			889026 063100	xxx123 456789		
12	1	0		900137 174211		-9	
	2			911248 285322		-8	
	3			922359 396433		-7	
	4			933470 507544		-6	
	5			944581 618655		-5	
	6			955692 729766		-4	
	7			966803 840877		-3	
	8			977914 951988		-2	
	9			989026 063099		-1	
	10			*000137 174210		0	
	OR			001371 742100	xx1234 567890		
13	1	1		*990260 630989		0	
	OR			902606 309890	x12345 678900		
14	1	0	(011111 111111)	913717 421001	(x12345 678900)	-9	(-)
	2			924828 532112		-8	
	3			935939 643223		-7	
	4			947050 754334		-6	
	5			958161 865445		-5	
	6			969272 976556		-4	
	7			980384 087667		-3	
	8			991495 198778		-2	
	9			*002606 309889		-1	
	OR			026063 098890	123456 789001		
15	TO	1				0	
	1			123456 789001			
16	TO			123456 789006			
	1			-12345 678900	-12345 678900		

* Indicates overdraft

There is a special circuit in MQC which responds to a carry from the units position in that counter. This carry produces the Overflow signal which operates as previously described under the A instruction. Thus, if ten subtractions of the divisor from the shifted dividend are not enough to produce overdraft, carry occurs at MQC which results in the Overflow response-transfer of control to 000. Such a result will be produced if the divisor is zero, or is numerically less than or equal to the dividend. In effect, this requires the correct quotient to be a quantity less than one.

13. Extraction

The E instruction permits the replacement of one or more arbitrarily selected digits in one word, with the correspondingly located digit of another word. The complete extraction operation requires four instructions: (1) transferring into F the word which controls the extraction, (2) bringing into A the word whose digits are to be replaced, (3) transferring from the memory to HSB2A the word which is to furnish the replacement digits, and (4) storing the modified word in the memory.

The four instructions are:

Memory Location	Instructions		Remarks
015	F 042		$z_0 \rightarrow rF$ (z_0 = 000011 000000)
		B 043	$z_1 \rightarrow rA$ (z_1 = B 200 H 400)
016	E 049		$E_{5,6}(y_1)$ (y_1 = 123417 032657)
		C 017	$(rA) \rightarrow 017$ ((017) = B 217 H 400)
017	[00 000		
		00 000]	
017	B 2(00)	H 400	

The operation to be performed is to select one of the ten words $c_0 \ldots c_9$, depending on the key digits (5th and 6th) of y_1, and place this word in 400.

The E instruction operates in the following manner: a signal from FT, produced by the presence of "E" in SR in conjunction with TS causes the word in F to be scanned. Each digit which has a binary "one" as the least significant pulse (all <u>even</u> decimal digits, B,D,F, etc.) produces no effect. Each digit which has a binary "zero" as the least significant pulse (all <u>odd</u> decimal digits, A,C,E, etc.) will cause a special "extract signal", E, to be present for seven pulse times. The "E" signal will clear the corresponding seven pulses (one computer digit) from the word circulating in A, and admit in its place a computer digit from HSB2A. Thus, on the minor cycle during which the "TS" signal admits a word from the memory to HSB2A, the "E" signal will replace certain computer digits in A with corresponding digits from HSB2A, as determined by the "extractor" word contained in F. Table 4 illustrates this action for the present example:

Table 4 - Extract Example

Digit Position:	12	11	10	9	8	7	6	5	4	3	2	1
Extractor (in F)	0	0	0	0	1	1	0	0	0	0	0	0
Old word in A	B	0	0	2	0	0	H	0	0	4	0	0
Word on HSB2A	1	2	3	4	1	7	0	3	2	6	5	7
New word in A	B	0	0	2	1	7	H	0	0	4	0	0

14. Control Transfers

The next group of instructions to be considered are the control-transfer instructions U. Q, and T. The U instruction has already been described, under the A instruction (overflow). The Q and T instruction accomplish the transfer of control in exactly the same manner as the U instruction, but the execution of the transfer is subject to a condition. The condition necessary for the Q instruction is that all 12 digits in A be identical with the 12 digits contained in L. The condition required for the T instruction is that the algebraic magnitude of the quantity registered in A be greater than that of the quantity held in L. If the quantities in A and L are not numerical, they are compared on the basis of their binary representation in the C-10 code; e.g., G < K, S < Y.

The example chosen to illustrate the Q and T instructions also involves the remaining instructions to which the computer is responsive. These will be discussed as they arise in the description of the example.

It is assumed that a block of data is to be processed, and the results recorded, both for future use in the computer and for typing out on a UNIPRINTER. One of three input tapes is to be chosen to supply the raw data, the criterion being the relative magnitude of the selected c_1 (line 017) and w_0. The coding of the example follows:

Memory Location	Instructions		Remarks
018	L 044		$w_0 \rightarrow L$
		T 027	If $c_1 > w_0$, go to 027
019	00 000		
		Q 029	If $c_1 = w_0$, go to 029
020	12 000		$\underline{c_1 < w_0}$: $T_2 \rightarrow rI$
		31 120	$rI \rightarrow$ 120 ...179; $T_1 \rightarrow rI$
021	30 060		$rI \rightarrow$ 060...119
		50 400	$c_1 \rightarrow$ S.C.
022	R 119		$U(c+1) \rightarrow 119$
		U 060	Go to 060
023	55 480		(480...539) $\rightarrow T_5$ (100 pulses/inch)
		76 000	(480...539) $\rightarrow T_6$ (20 pulses/inch)
024	61 000		Rewind T_1
		86 000	Rewind T_6 and set interlock
025	,0 000		Breakpoint
		10 043	S.C. $\rightarrow$ 043
026	10 026		S.C. $\rightarrow$ 026
		90 000	Stop
027	23 000		$\underline{c_1 > w_0}$: $T_3 \rightarrow rI$ (backward)
		40 120	$rI \rightarrow$ 120...179
028	11 000		$T_1 \rightarrow rI$
		U 021	Go to 021
029	14 000		$\underline{c_1 = w_0}$: $T_4 \rightarrow rI$
		31 120	$rI \rightarrow$ 120...179; $T_1 \rightarrow rI$
030	00 000		
		U 021	Go to 021

The right instruction from line 017 was "H 400", which placed c_i ($i = 0...9$ as selected by the key digits of y_1) in 400 but also left c_i in A. Line 018 causes w_0 from 044 to be placed in L and then "tests" to see if $c_i > w_0$. The T instruction causes the quantities from A and L to be sent to the comparator, where (A) is subtracted from (L) in the Binary Subtractor. If A > L, carry will occur and the Conditional Transfer (CT) signal will be sent to the Control Counter to permit it to receive the memory location digits which have been sent to HSB by CR. Thus, if $c_i > w_0$, the digits "027" will replace "019" in C, and the next instruction word executed will be line 027. However, if $c_i \leq w_0$, the CT signal is absent, and C remains unchanged at 019.

Assuming that the test of line 018 fails, the next line introduces first the skip and then the Q instructions. The Q instruction likewise causes the quantities from A and L (unchanged by the T instruction) to be sent to CP, where they become inputs to a Half Adder. If all twelve digits are identical, no output will be received from the Half Adder. If any two corresponding digits are dissimilar, the resulting output from the Half Adder will inhibit setting the flip-flop which produces the CT signal. Thus, the transfer of control takes place (as described for the T instruction) if and only if the two quantities are identical.

Assuming $c_i \neq w_0$, then $c_i < w_0$, and the Q instruction will have no effect on C. For this alternative, a block of raw data is to be read from the tape on UNISERVO No. 2 (abbreviated T_2), into memory locations 120...179; a second block of instructions from T_1 is to be read into 060...119; c_i is to be typed out on the Supervisory Control printer; and the control is then to be transferred to the first word (060) of the new instruction block. At the conclusion of the processing of the block of data by the instructions in 060...119 (whatever they may be), the control is to be returned to line 023.

15. Tape Instructions

The eight tape instructions are executed by the computer under the control of the Central Input-Output and Interlock Circuits, and either the Input Synchronizer or the Output Synchronizer. Simplified block diagrams of these circuits are shown as Figs. 1, 2, and 3, respectively.

Referring to Fig. 1, the tape instruction is assumed to be set up in the Static Register. The first instruction digit is passed to the Main Function Table and also to the First Instruction Digit Auxiliary Function Table. The latter table is designed to produce two of four possible signals. If the digit is "4" or less, the "R" signal (for "read") is produced. The digit "5" or greater produces the "W" signal (for "write"). If the digit is odd, the "F" signal (for tape to move forward) is emitted; an even digit produces the "B" signal (for tape to move backward). To execute the given tape instruction, the first instruction digit, in conjunction with PC, produces the FT signals required.

The second instruction digit passes from SR to the Second Instruction Digit Auxiliary Function Table, from which a selector signal, "nS", is sent to UNISERVO "n" (where "n" is 1, 2, ...9, -). If the selected UNISERVO is free (not engaged in a read, write, or rewind operation), the selector signal is returned as an "FIR" signal, a "BIR" signal, or both. The FIR signal is returned if the last instruction to that UNISERVO involved forward tape movement.

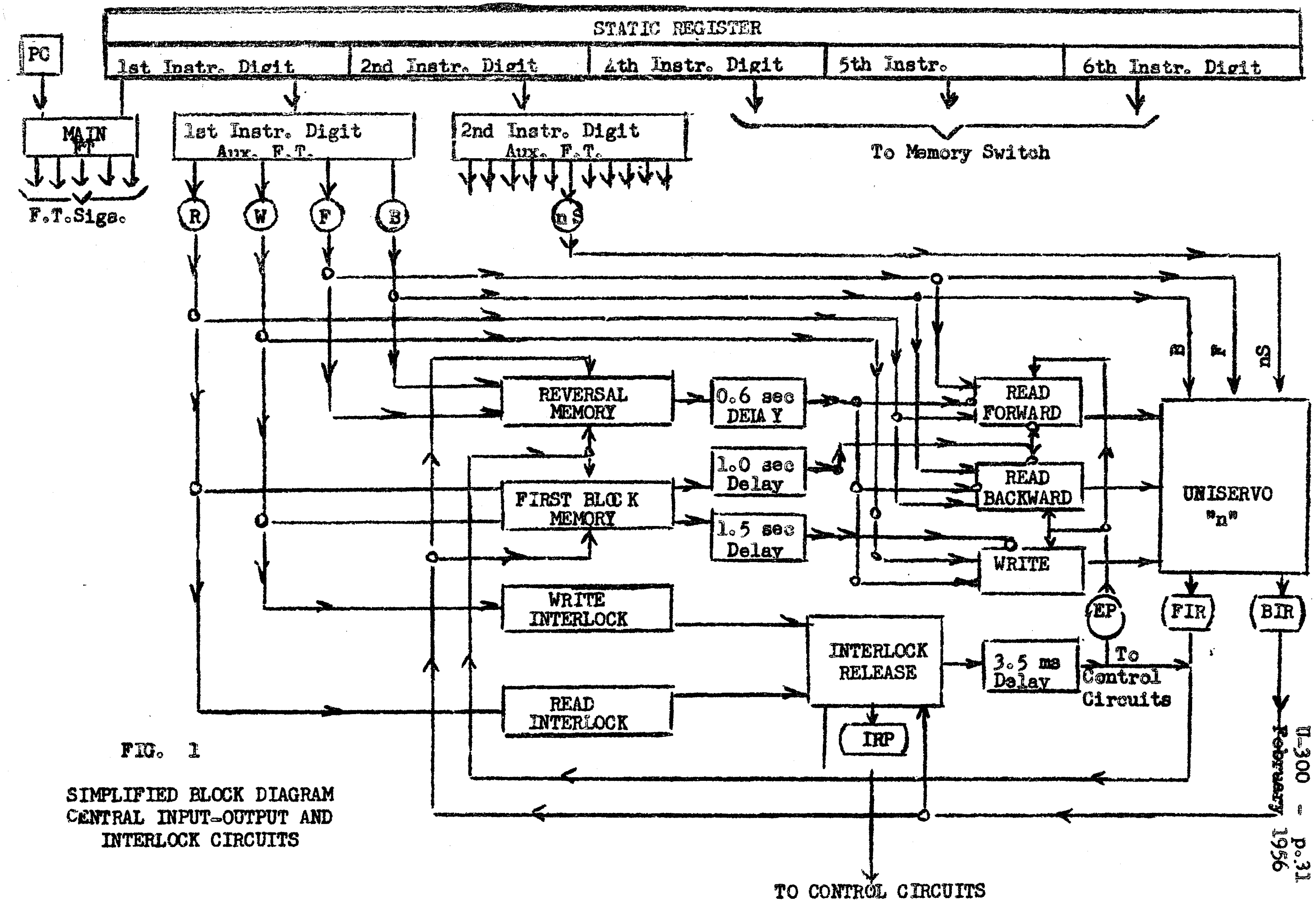

FIG. 1

SIMPLIFIED BLOCK DIAGRAM
CENTRAL INPUT-OUTPUT AND
INTERLOCK CIRCUITS

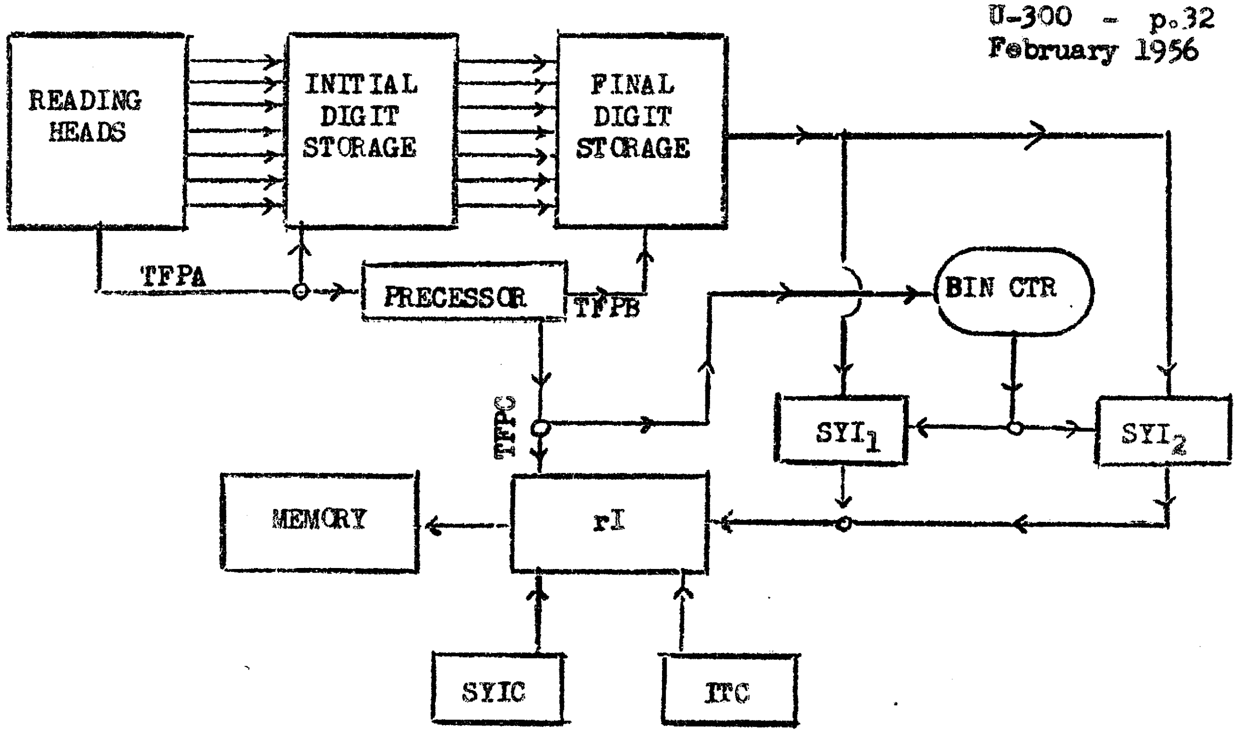

FIG. 2

SIMPLIFIED BLOCK DIAGRAM
OF INPUT CIRCUITS

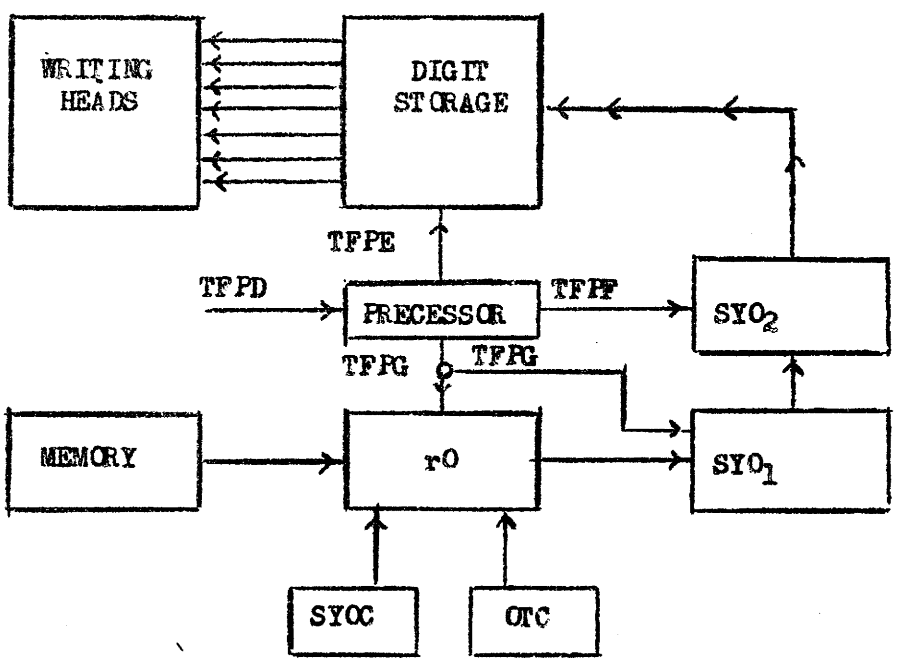

FIG. 3

SIMPLIFIED BLOCK DIAGRAM
OF OUTPUT CIRCUITS

The BIR signal denotes prior backward tape movement. The return of both signals indicates that the tape is in the rewound condition.

The FIR, BIR, F and B signals are sent to the "Reversal Memory". If the combination of signals indicates that the tape is to move in the opposite direction from its last movement, a delay of 0.6 second is interposed to permit realignment of the tape-tensioning controls for the new direction of movement.

The FIR, BIR, R and W signals are sent to the "First Block Memory". Presence of both FIR and BIR indicates that the block to be read is the first block on the tape. In this case, a delay of 1.0 second is interposed preceding the execution of a (forward) read instruction. The delay is 1.5 seconds if the instruction is "write". In either case, the tape is started at the regular time, but the reading or writing is held up until the end of the delay period.

The R signal is sent to the "Read Interlock". If a previous read instruction is still in progress, action is delayed until its completion. The W signal is sent to the "Write Interlock", which delays execution of the instruction, if necessary, until a previous write operation is completed.

The "Interlock Release" produces the "IRP" signal as soon as the Read Interlock or the Write Interlock permits. This signal is sent to the Control Circuits to set Time Out and step PC, thus initiating the execution of the tape instruction. On the last stage of PC for the particular tape instruction, an ending pulse is produced, which clears PC, steps CY, sets Time Out, and prepares SR for the next instruction. Although not accomplished in this manner, Interlock Release may be considered to produce the ending pulse 3.5 milliseconds later than IRP.

"Read Forward" is activated by the R and F signals, and the ending pulse as soon as permitted by the 0.6-sec. or the 1.0-sec. delay, if either is present. "Read Backward" is similarly controlled by the R and B signals, and the ending pulse and either delay. "Write" is controlled by the W and F signals, and the ending pulse subject to the 0.6-sec. or the 1.5-sec. delays.

The two rewind instructions are subject to the Write Interlock and Reversal Memory. The other controls for these instructions have been omitted from Fig. 1.

16. The Read Instructions

Referring to Fig. 2, a "read" instruction energizes the Reading Heads at the earliest time permitted by the interlock and delay controls. The pulse combination (1-7 pulses) constituting each computer digit is transferred to "Initial Digit Storage" as each group of magnetic "spots" constituting the digit passes under the Reading Heads. An eighth pulse, called the "sprocket channel pulse" is present in each digit pulse group on the tape. This pulse is delayed approximately one minor cycle after being read from the tape, and is then used as a control signal, TFPA. TFPA causes the transfer of the incoming digit from Initial Digit Storage to Final Digit Storage, and activates the Precessor. The Precessor is effectively a one-word register containing a single pulse. At the output of the Precessor, this pulse appears as the TFPB signal at the proper instant to cause the transfer of each incoming digit from Final Digit Storage to one of the two Synchronizer Input Registers (SYI_1 or SYI_2). These registers are used alternately, the alternation being produced by a Binary Counter. The first incoming digit is the "sign" digit, and this is placed in the last digit position of the circulating "word" in SYI. For each succeeding digit, the Precessor pulse is advanced seven pulse times, so that the new digit is placed in SYI immediately ahead of its predecessor. When all twelve digits have been accumulated, the Precessor pulse has been advanced sufficiently to appear at a second output of the Precessor to be released as the TFPC signal. TFPC initiates the transfer of the completely assembled word from the appropriate SYI to Register I, and steps the Binary Counter to alternate the SYI registers. (One SYI register is filling from the tape while the other transfers its word to I.)

The channel and word position in I, which receives the incoming word from SYI_1 or SYI_2, is determined by the reading of the Synchronizer Input Counter (SYIC) and the Input Tank Counter (ITC). Both counters are cleared to decimal zero at the beginning of each read instruction. The TFPC signal, indicating a word completely assembled in SYI, steps SYIC. The reading of SYIC is compared with that of TSC (in the Memory Switch). On coincidence, the word is transferred to that channel of I determined by the reading of ITC. When ten words have been transferred, SYIC emits a carry pulse which steps ITC, and then clears SYIC itself. When the 59th word has been transferred to I, a special circuit is activated which terminates the read operation (normally after the 60th word has been read) as soon as a period of one millisecond elapses without a new digit being read. This circuit produces a signal, "RE", which stops the tape, de-energizes the Reading Heads, and, after a delay of 7 milliseconds, resets the Read Interlock. The special circuit is also designed to produce an error indication and prevent the resetting of the Read Interlock if the 60th word is not completed or if a 721st digit is read before the end of the block is reached.

If a "read backward" instruction is being executed, the Precessor timing is altered to reverse the order in which incoming digits are accumulated in SYI. The readings of SYIC and ITC are complemented on read-out, and thus cause the incoming words to be assembled in I in the reverse order.

If a "3n" or "4n" instruction is being executed, the previous contents of I are emptied into the memory. This process is controlled by SYIC and ITC for Register I and by the Control Register, Static Register and Memory Switch for the Memory. Associated with CR is a timing circuit which causes "10" to be added to the memory location number held by CR every eleventh minor cycle. If CY is on γ, the left instruction is augmented; if CY is on δ, the right instruction is augmented. Actually, the word in CR is sent to the adder and the sum returned to CR on every minor cycle. The timing circuit provides decimal zeros as the second adder input on the first nine cycles of each ten, and "000010 000000" or "000000 000010" (CY on γ or δ, respectively) as the second adder input for every tenth minor cycle. Each time SYIC reads "9", a signal is sent to the Control Circuits to step PC, set TO, clear SR and set up the modified tape instrudtion in SR, thus selecting the next memory channel to receive the ten words from the next I channel. When PC reads"8", the FT signals calling for the transfer from I to memory are terminated, and the tape reading begins. For a "30" or "40" instruction, the ending pulse terminates the entire operation, as no tape reading is called for.

17. The Write Instruction

The write instructions "5n" and "7n" bring about the recording of a block of data on UNISERVO "n" at 100 or 20 pulses per inch, respectively. The six channels of memory are selected by modification of the instruction in CR as described for the read instructions 3n and 4n. The channels in Register 0 are selected by the Output Tank Selector OTC. When the Interlock Release (Fig. 1) produces the IRG2 signal, the block of data is transferred from the memory to 0. After all six channels of 0 have been filled, the first word of the first channel is transferred to SYO_1. This word is then transferred to SYO_2 and the second word to SYO_1. (See Fig. 3).

A "Start Writing" signal is sent to the Output Synchronizer when the tape has come up to its proper speed and the Writing Heads have been energized. This signal causes a series of pulses, called TFPD to be produced. For the 5n instruction, the TFPD pulses are produced once every two minor cycles (less 7 pulse-times), which produces a density of 105 pulses to an inch of tape. The 7n instruction records at a rate one-fifth as fast.

The "Start Writing" signal starts the Precessor so that the Precessor pulse will appear at one output (as a TFPF pulse) at the correct instant to transfer the "sign" digit from SYO_2 to Digit Storage. The TFPF pulse re-enters the Precessor so as to appear at this same output seven pulse-times earlier on succeeding minor cycles. The Precessor pulse arrives at a second output (as a TFPE pulse) after the completion of the writing of the digit (initiated as soon as the digit was set up in Digit Storage), and resets the flip-flop in Digit Storage preparatory to the receipt of the next digit. Thus, the twelve digits of the word in SYO_2 are successively recorded by the Writing Heads. The 12th TFPF pulse (called TFPG) restarts the Precessor for the next word, clears SYO_2, and initiates the transfer of the word in SYO_1 to SYO_2 and also the next word from 0 to SYO_1. SYOC and OTC control the output of 0 to SYO_2 in a manner similar to the control of I by SYIC and ITC. The transfer of the last word from

O to SYO_2 initiates the generation of a "Write Ending" (WE) pulse, which is sent to the Central Input-Output Circuits (Fig. 1) to reset the Write Interlock and provide the ending pulse to the Control Circuits to terminate the write operation.

18. The Read Tape Operations

Returning to the example and the coding on page 28, the left instruction of line 020 is "12 000". The first instruction digit, "1", causes the R and F signals to be emitted from the First Instruction Digit Auxiliary Function Table (see Fig. 1). In conjunction with PC-1, this digit also stimulates the FT signals required to initiate the interlock tests prior to a read forward operation. The second instruction digit, "2", causes the "2S" signal to be sent to UNISERVO No. 2, which we will assume is in its rewound condition. Hence, both signals FIR and BIR will be returned to the Central Input-Output Circuits, and, together with R, will activate the First Block Memory. The R signal finds the Read Interlock reset, as no read operation is in progress, and hence, the IRP will be produced and sent to the Control Circuits to set TO and advance PC. Although not accomplished in this manner, the First Block Memory may be considered to delay the production of the IRG2 signal for 1.0 second.

On PC-2, further FT signals are produced to initiate the read forward operation (as soon as permitted by the 1.0 sec. delay) and to provide an ending pulse to permit the computer to proceed to the next instruction. Since UNISERVO No. 2 is free, the F signal starts the tape in motion. However, the Reading Heads are not energized until the end of the first-block delay. When this delay has expired, the IRG2 signal, in conjunction with the R and F signals, cause the Reading Heads to be energized and the read operation to be started.

The block of data from T_2 is now read into I, as already described.

Meanwhile, the right instruction "31 120" has been set up in SR at the beginning of the 6 cycle. The first digit, "3", together with PC-1, produces the FT signals to initiate the interlock tests and the R and F signals. The Read Interlock has not been reset, as the previous read instruction is still in progress. The second digit, "1", causes the 1S signal to be sent to UNISERVO No. 1, which returns the FIR signal only, since the first block of T_1 has been read. The Reversal Memory is not set, since the combination F and FIR indicates no reversal is necessary for UNISERVO No. 1.

The Read Interlock prevents <u>any</u> further operation of the computer until the completion of the previous read instruction. Eleven milliseconds after the RE signal is emitted by the Input Synchronizer (denoting the completion of the block), the Read Interlock is reset, and permits the present instruction to be carried out.

The IRP now advances PC, whereupon new FT signals cause the contents of I to be transferred to 120...179 in the Memory. PC is advanced each time a channel has been transferred, so that the augmented instruction in CR will be set up in SR to select the next Memory channel. This transfer requires 66 minor cycles or approximately 2.4 milliseconds. PC is now reading "8", and a third set of FT signals clear SYIC and ITC. The ending pulse initiates the execution of the next instruction. A 3.5-ms delay, initiated on PC-2, is inserted prior to starting tape movements, to allow time for the relays in the selected UNISERVO to pick up. Thus, the transfer from I to the Memory is completed before the tape starts to move. A second delay of 6-ms is now inserted to allow the tape to come up to speed and then the Reading Heads are energized and the Input Synchronizer is stimulated to read the block into I.

The chosen block of raw data is now in the memory, and the new block of instructions is in I. The left instruction in line 021 (30 060) transfers the block of instructions from I to 060...119 in the Memory. Again, the Read Interlock will delay the execution of this instruction until the completion of the previous read. When "0" is the second instruction digit, no "nS" selector signal is present. Instead, a signal is sent from the "0" line of the 2nd Instruction Digit Auxiliary FT to the Interlock Release, which replaces the FIR or BIR signal, and thus actuates the Interlock Release as soon as the Read Interlock is reset.

19. Supervisory Control Operation

The right instruction of line 021 (50 400) causes the contents of 400 to be printed on the Supervisory Control. The "5" causes the 1st Instruction Digit Auxiliary FT to generate the W and F signals, and the "0" replaces the "nS" signal as previously described. As this is the first write operation, the Write Interlock is free and the IRP and IRG2 signals are produced by the Interlock Release without delay.

The "0" line of the Auxiliary FT also controls the "X00" line of the Main FT, which, in conjunction with "5", produces FT signals which alter the normal output sequence. HSB2 is connected directly to SYO_2, and "c_1" from channel 40 is read to SYO_2 at TSC "0". The "Start Writing" signal produces the initial TFPD, and starts the Output Synchronizer Processor, which, in turn, sets up the "sign" digit of c_1 in the Digit Storage flip-flops. The TFPE signal from the Processor clears these flip-flops, which have been supplying the digit pulse to the printer decoding table, from which the solenoid which actuates the selected type bar is controlled. Just before the type bar strikes the platen on the printer, a contact is closed which produces a "Go-ahead" signal, which becomes the next TFPD to stimulate the Processor to produce a TFPF. This TFPF causes the next digit from SYO_2 to be set up in Digit Storage. The printing proceeds through the first 12 digits. When the "Go ahead" signal following the 12th digit is returned to the Synchronizer it causes a 13th digit (an "ignore") to be set up in Digit Storage. This is printed as an "x", if the switch on the printer is set to "Check Print"; otherwise, the character has no effect. After a 40 μs delay, a WE (Write Ending) signal is given which terminates the operation.

Before the printing of c_1 had begun, an ending pulse was obtained from a FT signal set up by PC-2 and "5", so that the computer could proceed with the next instruction pair without waiting for the completion of the print operation.

20. Record Transfer of Control

Line 022 (R 119 - U 060) is a combination of particular usefulness in programming. The R instruction causes the reading of C (now 023) to be placed in memory location 119 as a U instruction (000000 U00023). This is accomplished by reading out of C from two gates. The first, open during the first three digit-times of the TS minor cycle, allows the three digits circulating in C to reach HSB1A. The second, open for the entire minor cycle except the first three and sixth digit-times, provides decimal zeros plus "U" in the sixth digit position to HSB2A from CU. The U instruction then transfers the digits "060" to C, so that the next instruction selected will come from 060.

It is assumed that the lines 060-118 contain a routine which processes the data contained in 120...179, placing the results in 480...539. When line 119 is reached, the transfer instruction U will return control to line 023.

21. The Write Operations

The left instruction of line 023 (55 480) causes the block of data beginning with 480 to be recorded on T_5. If the Supervisory Control printer has not completed the printing of c_1, the Write Interlock will delay the IRP from the Interlock Release. Assuming UNISERVO No. 5 is rewound, the 5S signal will be returned as both FIR and BIR, so that the First Block Memory will impose a 1.5-second delay before the "Start Writing" signal is produced. The tape on UNISERVO No. 5 will be started 3.5 milliseconds after the IRG2 signal is generated. During this 3.5 milliseconds period, the six channels of memory, 48-53, will be transferred to Register 0, controlled by SYOC and OTC. After stimulation by the "Start Writing" signal, the Precessor will regulate the successive word transfers from 0 to SYO_1 and from SYO_1 to SYO_2, as well as the digit transfers from SYO_2 to Digit Storage and from the latter to the Reading Heads. An odd-even check is made during the final transfer. An even pulse count in any recorded digit will cause an error neon on the Supervisory Control to be lighted, and will also prevent the resetting of the Write Interlock.

Should a "bad spot" occur on the tape, its presence is indicated by a punched hole, which is detected by a photo-cell. This photocell is located, with respect to the Writing Heads, so that the first digit and last words may be recorded without interference. The recording of any other word is delayed 0.1 second following the last hole detected.

When recording for tapes to be used in the tape-to-card device, a 10 millisecond delay is interposed after every ten recorded words. This delay is controlled by a selector button on the Supervisory Control.

As previously explained, the 5n instruction produces density of approximately 100 pulses to an inch of tape. The 7n instruction is executed in the same manner as the 5n instruction, but TFPD pulses are produced at a rate only one-fifth as fast so that the tape may be used with a UNIPRINTER to print out its data. Hence, the right instruction of line 023 (76 480) records on T_6 the same block just recorded on T_5, except for the lower pulse density. The execution of this instruction is held up by the Write Interlock until the previous write instruction has been completed.

22. The Rewind Instruction

Line 024 (61 000 - 86 000) produces the rewinding of T_1 and T_6. The execution of the rewind instruction requires either the FIR or the BIR signal from the selected UNISERVO. (If both are received, the tape is considered already rewound.) In the present case T_1 is free, but T_6 is still receiving data from the 76 instruction of line 023.

23. The Breakpoint Instruction

The instruction ",0 000" (breakpoint) is decoded as a "skip" if the Breakpoint Switch on the Supervisory Control is set to "Normal". When this switch is set to "Breakpoint", the computer will stop. The stop is effected by setting the Stop flip-flop, which prevents the TO flip-flop from resetting. Thus, the computer remains in Time Out until the "Start" bar on the Supervisory Control keyboard is depressed.

In the present example, the breakpoint was supplied to halt the computer if no more data was to be processed. Assuming that such is not the case, the Breakpoint Switch would be set at normal and the instruction is then interpreted as a skip.

24. Supervisory Control Input

The right instruction "10 043" calls for a word typed on the Supervisory Control, to be placed in memory location 043. The Supervisory Control Input (SCI) Switch is in its normal position. The Input Synchronizer serves the same function, i.e., to assemble the digits of the typed word as it does for words read from a tape. The "go ahead" signal from the keyboard becomes the TFPA to initiate the reception of a digit. After twelve digits have been typed, the Word Release key is depressed. This causes the printer to print a "period", produces the RE signal, steps PC, and terminates the typing operation. On PC-3, the word is transferred from SYI to the Memory (043), by-passing I, on the following TS cycle. The RE signal initiates the same ending delays (4 ms and 7 ms) used in tape reading.

The left instruction of line 026 (10 026) is executed in a similar manner, after the Read Interlock is reset at the conclusion of the previous 10 instruction. A new value of y_1 would be typed into 043, and a new instruction pair (or the same one) may be typed into 026. The new instruction pair will not be carried out at this time, since the current pair is in CR, and hence the stop instruction, "90 000", will next be carried out. This instruction, like ",0", sets the Stop flip-flop and prevents Time Out from ending until the Stop flip-flop is reset by the Start bar on S.C.

25. Tape Reversal

Returning to the test of line 018: If $c_i > w_0$, control is transferred to line 027. The instruction pair "23 000 - 40 120" causes T_3 to read (the tape running backward) into I, and then the contents of I are read into the Memory. Assuming UNISERVO No. 3 had last operated in the forward direction, the combination of B (from "2") and FIR signals at Reversal Memory will interpose a 0.6 second delay before the tape motor is started. This time is required by the balancing circuits which control the tensioning loops, in order that the correct tension be applied to the tape for the new direction of movement. The "40" instruction operates exactly as the "30" instruction, and is interchangeable with it.

Line 028 (11 000 U 021) causes the new block of instructions from T_1 to be read into I and then transfers control to 021 where these instructions are read into the Memory. The main sequence of instructions is now carried out, as already described. It would have been proper to use the instruction "31 120", in place of the two instructions, "40 120" and "11 000". The same results would have been achieved, and in a shorter time interval.

Returning to the test of line 019: If $c_i = w_o$, control is transferred to line 029, where the next block of T_4 is read to 120...179, the new block of instructions is read to I, and control is then transferred to the main routine at line 021.

26. Checks

The computer is provided with several types of checks to detect and locate computer errors. The basic check is that of the oddness of the pulse count on each computer digit. Eight binary counters are employed for this purpose: one on the minuend input of one of the duplicated Adders, one on the subtrahend input of the other duplicated Adder, one on each of the duplicated High Speed Busses, two on the input flip-flops of the Input Synchronizer, and two on the output flip-flops of the Output Synchronizer. Each counter is designed to energize an error circuit should any computer digit sampled be found to contain an even number of pulses.

There are a number of duplicated units in the computer, the pulse trains of each of which are continuously compared in a like-unlike checking circuit. The A,F,L and X Registers, the Adder, the comparator, HSB2A, and a portion of the Cycling Unit are duplicated. Many elements in the remaining units are duplicated and error circuits are associated with them to stop the computer and indicate the source of the error by lighting a neon on the Supervisory Control.

The Function Tables are designed to decode the checking pulse, and hence each output line requires an odd number of input lines to excite it. A even pulse combination, therefore, results in no excited output line, and the computer stalls. The stall is detected by a delay flop associated with CY, which recovers if no pulse is sent to advance CY during a 2-second interval. The stall is indicated by a neon on the Supervisory Control.

The Memory contents are subjected to a Periodic Memory Check every three seconds. Each tank is read into the HSB, the HSB checker sampling each digit to insure its oddness. The PMC occurs on α-time, thus interrupting the normal sequence of instructions until completed. The MQC is used to count the minor cycles. By clearing MQC to binary zeros, the carry is deferred until the 13th minor cycle, thus insuring that every digit in a channel is subjected to the check.