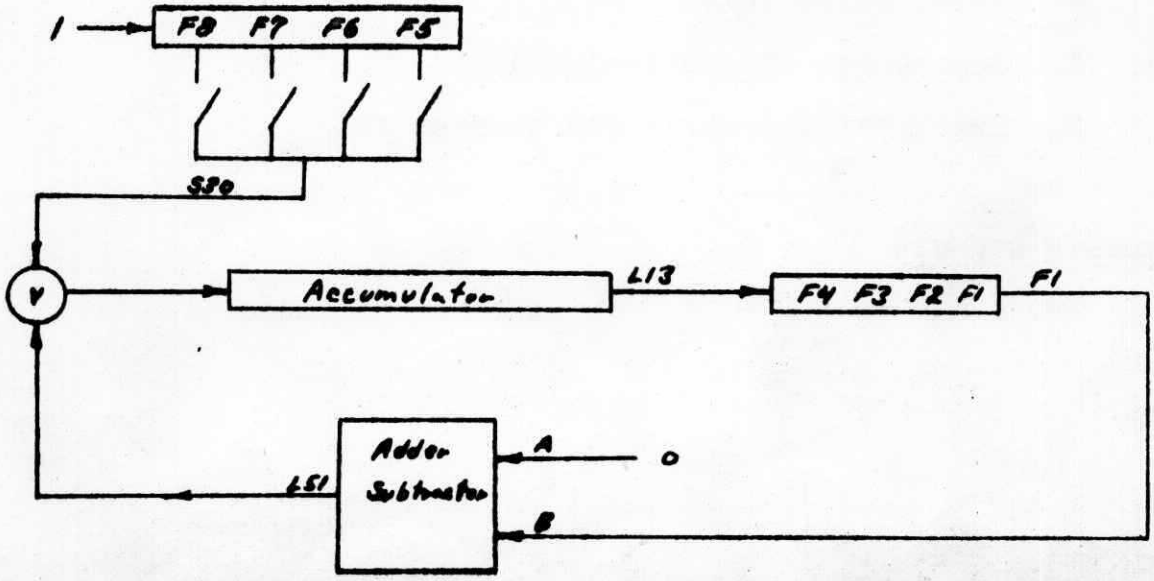
6. Record the input character (S30) in the accumulator at the fast access record net (L5).
7. Shift the accumulator (L13) into F4 and F4 through F1. - (using lack of Q2)
8. Gate F1 into adder B gate.
9. Hold adder A gate low.
10. Record adder output (L51) in the accumulator.
11. Reset F9 at T31. - (G19 x G4 x T31)
12. Set C1 at T31.

C1
1. Regenerate the control loop.
2. Reset F16 and shift F16 through F1.

Decoded Signals

K16 = G19·F9·S81	Reset entry
K33 = K2 v K16	Any input

Block Diagram



LOAD

Description

Pressing the Load button on the control panel causes the contents of the accumulator to be transferred to the control loop, but does not allow orders precession to take place. The button may be pushed at any time but will only operate if the machine is in the stop condition.

Sequence

C2
1. Set F9 if the Load and Start buttons are not pushed, if the Reset Entry device is not actuated, and if G19 is set.
2. Generate K20 if F9 is set and the Load button (S9) is pushed.
3. Set C10 high with K20.
4. Set C3 when C10 goes high.
5. Light "Load" light.

C3
1. Halt in C3 until C10 goes low (about 10 milliseconds).
2. Set C4 at T31 if C10 is low.

C4
1. Record accumulator (L13) in the control loop.
2. Reset F9 at T31.

C1
1. Regenerate the control loop.
2. Reset F16 and shift F16 through F1.

Decoded Signals

K20 = G19·F9·S9 Load

Loop cont

C4 cont.

② reset F4 at T31

$\overline{F4} = G19 \cdot C41$. (G19 remains set).

C1 ① regenerate the control loop.

$L4 = L3 \cdot C1 \cdot G19 \cdot \overline{G18}$ (G18 reset in C1, step 4. Reset)

② reset F16 and shift F16 through F1.

F16 is reset by L3 - the and gate on reset is held low by $\overline{G19}$, and the set gate is held high by G19.

$\overline{C1}$ goes low and lets F16 shift through F13.

C1 goes high and lets F13 shift through F8.

$\overline{C2}$ goes low and lets F8 shift through F1 (C18 high - F8 → F6)

set C2 and hold in C2 (with G19)

LOAD - pressing the load button transfers contents of the accumulator to control loop when the machine is in the stop condition. Orders precession is not allowed to take place (- because G1Q high - prevents L3 passing into F16)

SIGNALS

K20 (load) = G1Q · FQ · SQ G1Q = stop flip flop
SQ = load switch.

(C2) (1) set FQ if (i) load and start buttons not pushed (SQ & S7 rest).
(ii) reset entry device not actuated (this gives $\overline{S81}$)
(iii) G1Q is set
FQ = C2 · G1Q · $\overline{S81}$ · $\overline{SQ}$ · $\overline{S7}$

(2) generate K20 when load button pushed - if FQ is set
K20 = G1Q · FQ · SQ.

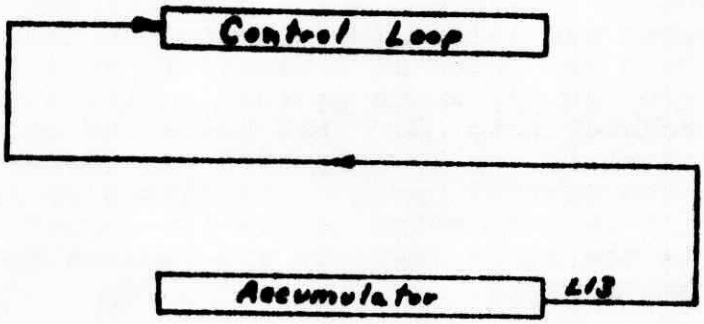
(3) set C10 high with K20. - input delay - lasts for about 10 milliseconds.
C10 = K20 · C2 · FQ · P21 ← ϕ2 ϕ3

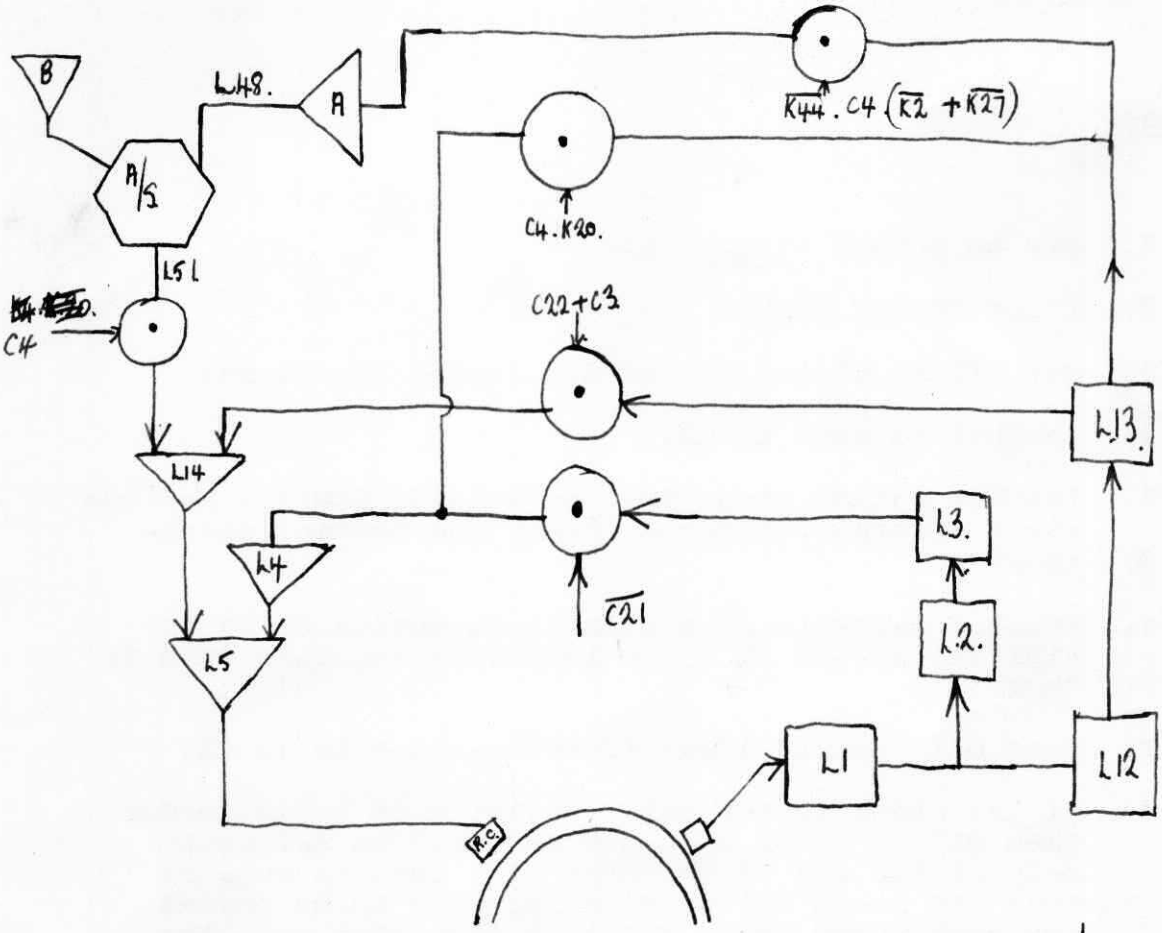
(4) set C3 when C10 goes high.
$\overline{C10}$ holds reset gate of C22 high - when C10 high, $\overline{C10}$ low - allows gate to go low for T31 to set C3.

(C3) (1) hold in C3 until C10 goes low (see C2 step 3).
(2) set C4 when C10 goes low.
C10 holds set gate of C21 high - goes low and C4 set at next T31.

(C4) (1) record accumulator in control loop. - L13 → L4.
L4 = L13 · C4 · K20.
Also. L48 = L13 · C4 · K20. , L48 → L51, L51 → L14,
L14 = L51 · C4.

Block Diagram





LOAD OPERATION.
K20. sq Switch.

HALT ON-OFF

Description

The Halt button provides a means of interrupting the course of a program and allows the operator to "single-cycle" through the program one order at a time. It does this by setting a flip-flop in C1, which in turn inhibits the advance of the C counter into C3. This holds the order in the shift register where it may be inspected via the shift register lights on the control panel. The flip-flop is then reset with the Start button, which allows the computer to perform the order in the shift register and advance to the next order where it halts again.

When the Halt button is again pushed, the computer may be started again with the Start button but will not halt on succeeding orders.

Sequence

ON

1. May be pushed at any time.
2. Light "halt" light.
3. Set G17 at C1 and T31 if G19 (stop) is not set.
4. Inhibit advance to C3.
5. Inhibit sector comparison by holding C20 low and inhibit setting F11 for multiply and detract operations.
6. Inhibit magnetic card operations during C2 by inhibiting set of F9 which indicates magnetic card is "home."
7. Turn off "Start" light if the machine is in C2.
8. If the order in the shift register is a Stop order, then G17 is reset with G19 and will be set again only at the end of the next C1. This is done so that the Start button does not need to be pushed for each reset entry and load operation when the machine is stopped.
9. Reset G17 at T31 with the one-shot circuit (S21) when "Start" button is pushed. This allows the

HALT - ON, OFF

provides a means of stepping through program - one instruction at a time - it sets a ff. in C1 which inhibits the count from moving into C3 when the start button is pushed the flip flop is reset and the order in the shift register is performed, the next order is shifted into the shift register and then halts in C2 again.

when the halt button is again pushed, the computer may be started again with the start button and does not halt in C2 now.

ON. (1) button may be pushed at any time (to give S20)
(2) light "halt" light. (supplied from hold switch direct to light amp.).
(3) set G17 at C1 and T31 - if G19 (stop ff.) is not set.

$G17 = C1 \cdot T31 \cdot \overline{G19} \cdot S20 \cdot S25$ — S25 = reset back contact.

(4) inhibit advance to C3. — G17 holds reset gate of C22 high and therefore cannot be reset.

(5) inhibit sector comparison by holding C20 low and inhibit setting F11 for multiply and detract operations.

$\overline{C20} = G17.$

$F11 = \overline{G17} \cdot C2 \cdot T31 \cdot \overline{F12} \cdot K28.$ K28 = mult. or detract.

(6) inhibit magnetic card operations during C2 by inhibiting set of F9 which indicates magnetic card is "home"

$F9 = \overline{G17} \cdot C2 \cdot K44 \cdot S73$ K44 = mag. card read or write.
S73 = mag. card home.

(7) turn off "start" light if machine is in C2.

$K51 \text{ (start light)} = G19' \cdot (G17' \vee C2').$

HALT - (ON cont.).

C2

⑧ if order is an 'stop' order in shift register. G17 is reset with G19.

$\overline{G19} = F16' \cdot F15' \cdot F14' \cdot F13' \cdot F12' \cdot C2'$

$\overline{G17} = G19 \cdot T31 \cdot C22 \cdot S25$ S25 = reset switch back contact.

this is done so that the start button does not have to be pressed for every reset entry and load operation — when the machine is in the stop condition ie. when G19 is set

⑨ reset G17 at T31 with the one-shot circuit (S21) when start button is pushed

S21 = S24 (start switch) — S21 stays high briefly for a short time and then $\overline{G17} = S21 \cdot T31$.

with G17 reset the word can move to C3 (step 4) and the command which was demand in the shift register up to this time is performed.

at next C1, G17 is again set

OFF - ① may be pushed at any time. - (S20 goes low).

② halt light goes out.

③ nothing happens to G17 until it is reset with the start button above. (see step 9 above). - (nothing high on G17 reset gate to make it reset)

④ however G17 once reset, is not set at C1 and T31 as before so that sequence continues as normal.

$G17 = \overline{G19} \cdot C1 \cdot T31 \cdot S25 \cdot S20$ ∴ as S20 low G17 cannot set.

C counter to advance and perform the operation which has been dormant in the shift register up to this time. In the C1 immediately following the operation, G17 is set again and the sequence of events starts over again at Step No. 3.

OFF

1. May be pushed at any time.
2. "Halt" light goes out.
3. Nothing happens to G17 until it is reset with the Start button in the usual manner (See No. 9 above).
4. However, G17 is not set at C1 and T31 so the computer is not brought to a halt for each operation as before.

START

Description

Pressing and releasing the Start button causes the machine to advance the cycle counter to the next C count (C3) and to reset the stop flip-flop. This allows orders precession to occur and hence puts the computer under control of the order in the control loop. The button may be pushed at any time but will only operate if the machine is in the stop condition.

Sequence

C2
1. Set F9 if the Load and Start buttons are not pushed, if the Reset Entry device is not actuated, and if G19 is set.
2. Set F10 if F9 is set and the Start button is pushed.
3. Set C10 if F10 is set and the Start button is released.
4. Set C3 when C10 goes high.

C3
1. Halt in C3 until C10 goes low (about 10 milliseconds).
2. Set C4 at T31 if C10 is low.

C4
1. Gate accumulator into the adder B gate.
2. Hold adder A gate low.
3. Record adder output (L51 = L13) in the accumulator.
4. Reset F10, F9, and G19 at T31.
5. Light "Start" light when G19 is reset.

Decoded Signals

None.

START - (BUTTON) resets stop f.f. (G19) and orders precession take place.
SIGNALS - none.

(C2) ① set F9 if. (i) load and start buttons (S9, S7) not pushed.
(ii) reset entry device not actuated
(iii) G19 (stop f.f.) is set.

$F9 = C2 \cdot G19 \cdot \overline{S81} \cdot \overline{S9} \cdot \overline{S7}$

② set F10 if F9 is set when the start button (S7) is pushed.

$F10 = F9 \cdot G19 \cdot S7$

③ set C10 if F10 is set and the start button is released

$C10 = G19 \cdot F10 \cdot \overline{S7} \cdot C2 \cdot P21 \cdot F9$.

④ set C3 when C10 goes high - lets reset C22 go low and triggered by T31.

(C3) ① halt in C3 until C10 goes low - about 10 milli-seconds.
② set C4 at T31 if C10 is low

(C4) ① gate accumulator into adder B gate

$L49 = G19 \cdot C4 \cdot L13 \cdot F10$ (F10 remains set because, although signal on set gate has gone low - $S7 \rightarrow \overline{S7}$, there is no high on reset gate.

② holder adder A gate low. - no signals.
③ record adder output ($L51 [= L13]$) in the accumulator.

$L84 = L51 \cdot C4$.

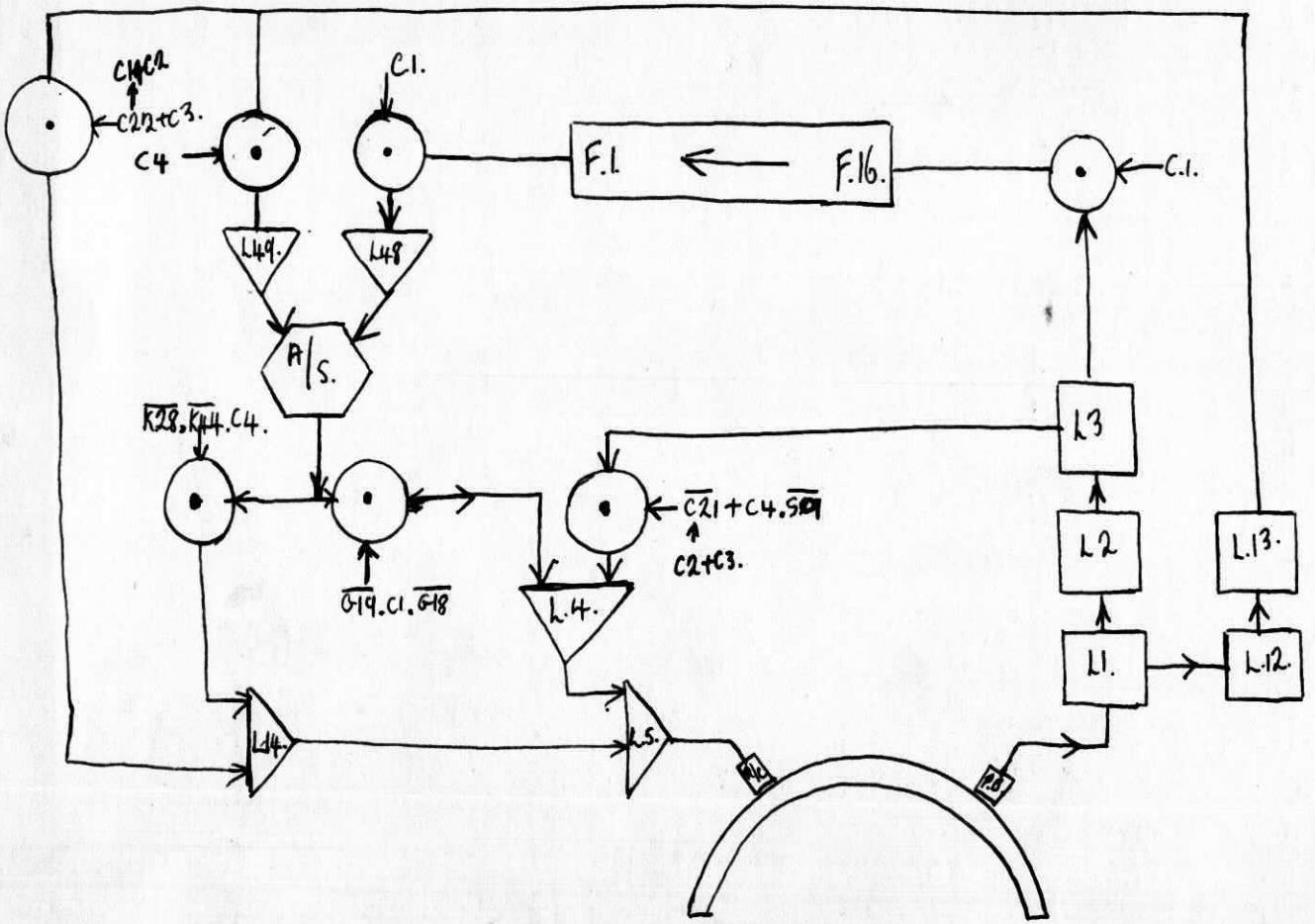
④ reset F10, F9 and G19 at T31.

$\overline{F10} = G19 \cdot C41$
$\overline{F9} = G19 \cdot C41$ } at next 13
$\overline{G19} = F10 \cdot C41$

⑤ light "start" light when G19 is reset.

"start" light = $K51 = \left(C2 + \overline{C22} + \overline{G17}\right) \cdot \overline{G19}$.

WITH G19 reset orders precession takes place in C1. - 13 can pass into F16 in C1.



START. S7

STOP

Description

The Stop command causes the machine to stop in C2 and allows the manual operations of Reset Entry, Load, and Start to be performed. Once the stop flip-flop (G19) has been set either by this operation or by Reset, the normal sequence of orders precession during C1 is prohibited. This preserves the stop command in the orders shift register.

Sequence

C2 1. Set G19. $G19 = F16' \cdot F15' \cdot F14' \cdot F13' \cdot F12' \cdot C2.$

2. Use G19 to inhibit advance to C3. (G19 holds reset gate C22 high).

3. Light Reset light with G19. $-K52(\text{reset light}) = G19$

Coding

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Operation

Lights to indicate no buttons are depressed.

Lights when Start switch is pressed. (See "Start.") $F10 = F9 \cdot G19 \cdot S7$

S7 = start.

Decoded Signals

None.

CLEAR ACCUMULATOR

Description

The Clear Accumulator command causes zeros to be recorded in the accumulator.

Sequence

C2 1. Do nothing.

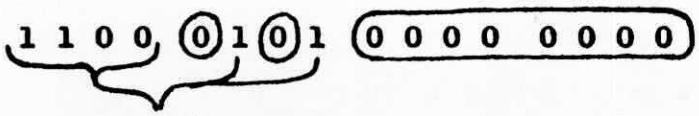
C3 1. Do nothing.

C4 1. Hold A gate low.

2. Hold B gate low by holding intervention switch net (W/29) low with F9'.

3. Record adder-subtractor (L51 = 0) in accumulator.

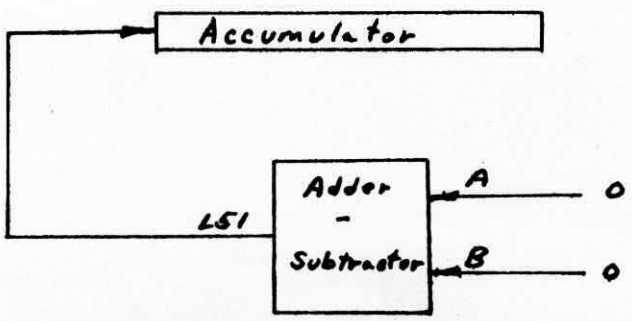
Coding



Decoded Signals

K1 = F16·F15·F14'·F13'

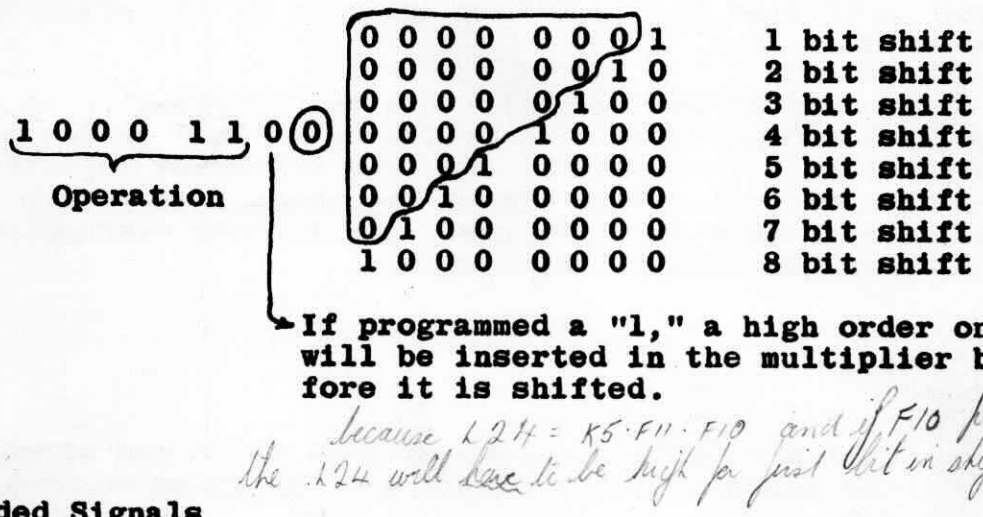
Block Diagram



6. Set C3 at T31 if F1 is low and repeat process.

7. Set C1 at T31 if F1 is high.

Coding

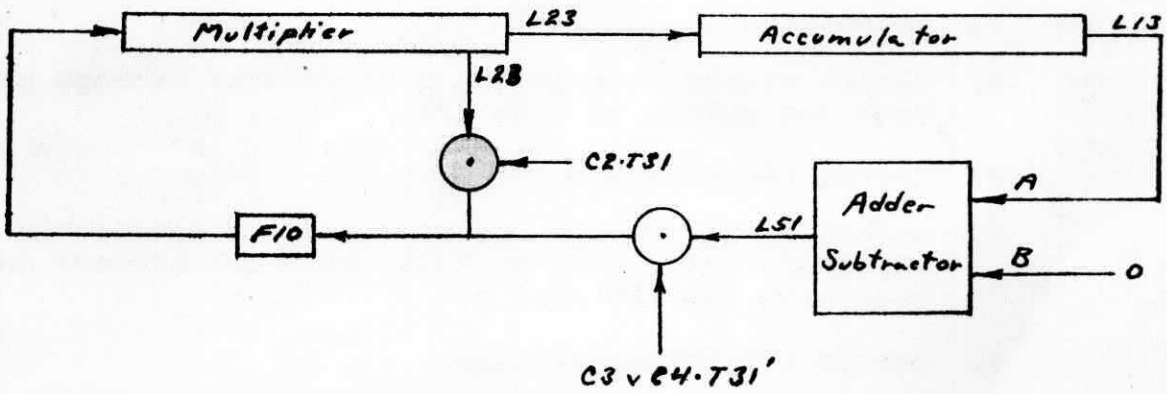


because L24 = K5·F11·F10 and if F10 programmed high the L24 will have to be high for just bit in shift.

Decoded Signals

K19 = F16·F15'·F14'	Any shift
K5 = K19·F13'·F12(C3 v C4)	Decimal shift right
K5' = F16' v F15 v F14 v F13 v F12'	(Decimal shift right)'

Block Diagram



JUMP, UNCONDITIONAL

Description

The Jump command transfers the contents of the storage register specified by the address portion of the order link to the control loop.

At the end of the operation the carry flip-flop is set so that the address specified by the order link in the shift register will be increased by one when the shift register precesses during C1. This allows the machine to process in sequence the orders in a program. For further discussion, see paragraph on "Orders Precession."

Sequence

C2
1. Form a circulating loop with F8 to F5 and circulate the sector portion of the address under control of C20.
2. Use F12 to compare sector address (F5) and sector address track (Z5). Set F12 if F5 is not equal to Z5 during C20·T31'.
3. Reset F12 at T31 if C20 is high.
4. If F12 is already reset at T31, indicating the sector address and address track are the same, set C3.
5. If F12 is set at T31, return to Step No. 2 of C2.
6. If the desired address is in fast access, C20 will be low and C3 will be set at the first T31 of C2.

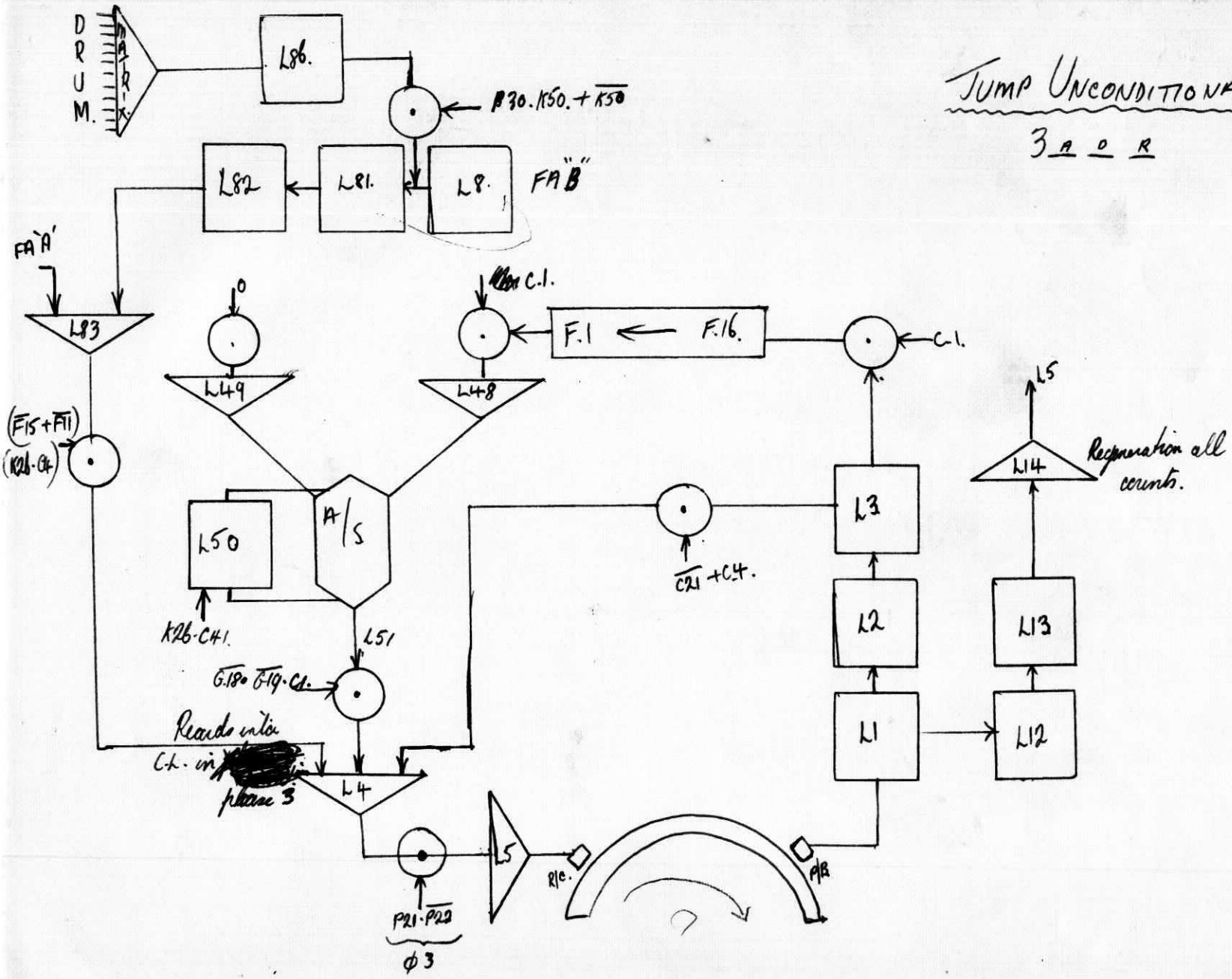
C3
1. Do nothing.

C4
1. Select either fast access B or general storage playback and unlace it with L81.
2. Retime L81 with L82.
3. Select either L82 or one of the three arithmetic registers (L13, L23, or L33) which are already on time with the L83 net.
4. Record L83 in control loop.
5. Set the carry flip-flop (L50) at T31 in order to add one to the Jump command address during C1.

Page 3-139

JUMP UNCONDITIONAL

3 A O R



JUMP UNCONDITIONAL: 3a D R

SIGNALS K26 (any jump) = F16' · F14 (F15+F13)

K9' (Jump unconditional)' = F16 + F15 + F14' + F13'

K50 (fast access) = (F1' + F2' + F3') F4' · F5' · F6' · F7' · F8' · F9' · F10'

C20 (gen. storage) = C2 · K50' · G17' (any command with a storage address)

(C2) ① form a circulating loop with F8 to F5 and circulate the sector portion of the address under control of C20.

F8 = F5 · C20

C18 = Q2 + C20 + K44 · C2 - C18 allows shift to occur F8 → F5.

② use F12 to compare sector address (F5) and sector address track (25) and F12 is set if F5 is not equal to 25 during C20 · T31'

F12 = (25 + F5)(25' + F5') · C20 · T31

∴ if 25 and F5 differ both gates will be high and F12 will set.

BUT if 25 and F5 are either both positive or both negative only one of the gates will be high the other will be closed and F12 will remain reset

③ reset F12 at T31 if C20 is high

F12' = C20 · T31

④ if F12 is already reset at T31 this indicates that the sector address and address track are the same and C3 is set. If F12 remains reset then the gate to reset C22 remains low until it is set by T31

⑤ if F12 is set during sector comparison - this indicates that the sector address and address track did not agree. and C3 is not set. Therefore the next sector address track is compared to the sector address and the cycle is repeated again at step 2. - F12 being reset at T31. (to 3)

⑥ if desired address is in fast access then C20 will be low and the

Jump unconditional

gate on the reset of C22 will be low and so C3 will be set at the first T31 in C2.

C3 - do nothing.

C4. ① select either fast access B or general storage playback and enable it. with L81. and ~~as~~ retime L81 with L82.

② select either L82 or one of the three FA"A" registers with L83:

GEN STORAGE: Drum → L86 → L81 → L82 → L83 [= L82 · (F1+F2) · (K50+F3)]

also L83 = L81 · F1 · F2 · (K50 + F3)

∴ L82 may pass into L83 if (i) (K50 + F3) = if the address is either not fast access or (if it is fast access) then it must not be above 011 = (003 = FA'B') i.e. it must not be FA'A'.

and (ii) (F1+F2) = this means that if the address has last 2 bits = 010 ending in then this gate will be low (and only if addr ends 010). - for all other addresses this gate is open. As can be seen (in the other equation for L83) this address (010 = FA.2.) is passed into L83 from L81 - this is because any address that has these last 2 bits is recorded in phase 3 and does not need to be delayed one bit time by L82 (see p.3-115).

or any address with these last 2 bits

F.A."B": Drum → L8 → L81 → L82 → L83 - (see above).

F.A."A": Drum → L1 → L12 → L13 → L83 (= L13 · F2 · F3 · K50).

" " L22 → L23 → L83 (= L23 · F1 · F3 · K50)

" " L32 → L33 → L83 (= L33 · F1 · F2 · F3 · K50).

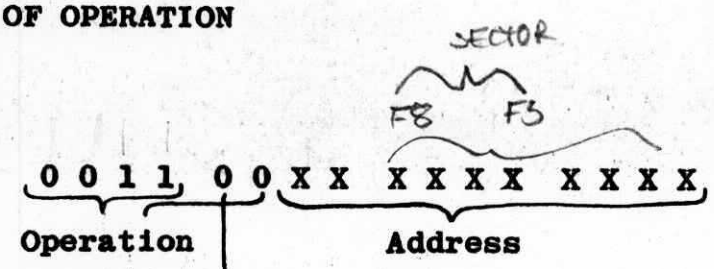
③ record L83 in control loop.

L4 = (F15 + F11) · K26 · C4 · L83 and L5 = L4 · P21 · P22

④ set the carry f.f. (L50) - so that 1 is added to the jump command address in C.I. (during order succession).

L50 = K26 · C4 · T31.

Coding

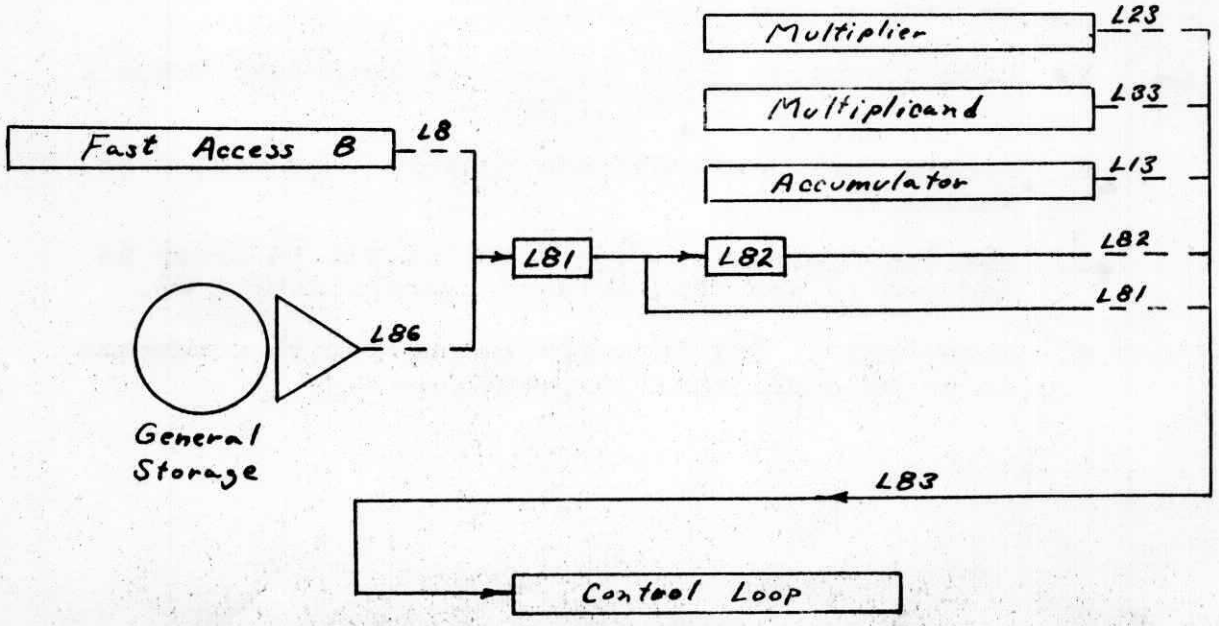


If = 1, the machine will not compare during the first word time of C2 regardless of sector address unless address is in fast access, in which case C20 will be low and C2 will only last one word time.

Decoded Signals

K26 = F16'·F14(F15 v F13)	Any jump
K9' = F16 v F15 v F14' v F13'	(Jump, Unconditional
K50 = (F1' v F2' v F3')F4'·F5'· F6'·F7'·F8'·F9'·F10'·F12'	Fast access
C20 = C2·K50'·G17'(any command with a storage address)	General storage

Block Diagram



JUMP RETURN

Description

The Jump Return command is the same as the unconditional jump command with additional features which transfer the contents of the control loop to fast access No. 2. This makes it possible for the programmer to jump out of his program into a subroutine and to return to the next order in his program by jumping to fast access No. 2 at the end of his subroutine.

Since fast access B is recorded on the drum one bit time earlier than fast access A which contains the control loop, the transfer of the control loop into fast access No. 2 must start one bit time early, taking the early control loop (L2) and recording it across the word time (i.e., from C3·T31 to C4·T30 inclusive).

The Jump Return command replaces the present jump link so it now becomes the new jump link in the orders loop. Therefore, it must be modified to an unconditional jump command before it leaves the shift register at the completion of the operation.

Sequence

C2 1. Locate proper sector (see Steps 1 to 6 of Jump, Unconditional).

C3 1. Use C5 to start recording early control loop (L2) in fast access No. 2 at T31.

C4 1. Record storage (L83) in control loop (see Steps 1 to 4 of Jump, Unconditional).

2. Record early control loop (L2) in fast access No. 2 during T31'.

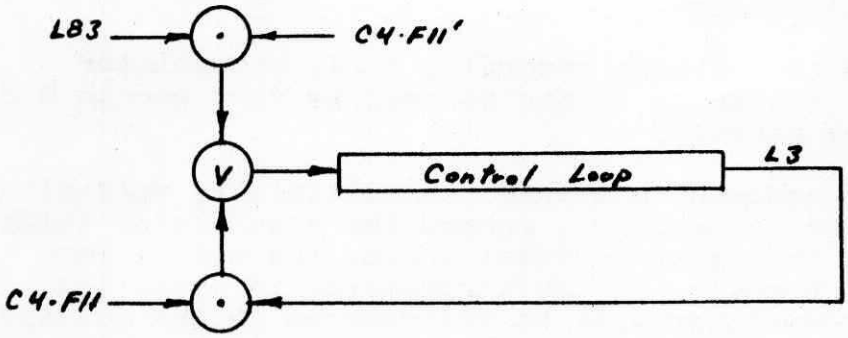
3. Set the carry flip-flop (L50) at T31 in order to add one to the jump command address during C1.

4. Reset F11 at T31 in order to convert this command to an unconditional jump command.

Decoded Signals

K26	= F16'·F14(F15 v F13)	Any jump
K11'	= F16 v F15' v F14' v F13'	(Jump if accumulator is less 0)'
K50	= } See Jump, Unconditional	Fast access
C20	= } See Jump, Unconditional	General storage

Block Diagram



TRANSFER ACCUMULATOR TO STORAGE

Description

This command transfers the contents of the accumulator to the storage register specified by the address portion of the order link. The contents of the accumulator are not destroyed in this operation.

Sequence

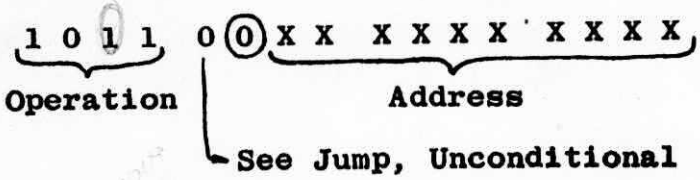
C2 1. Locate proper sector (see Steps 1 to 6 of Jump, Unconditional).

C3 1. Use C5 to start recording early accumulator (L12) in storage at T31 if the address is fast access B or general storage.

C4 1. Use C5 to continue recording early accumulator (L12) in storage if the address is fast access B or general storage.

2. If the address is either the multiplier, multiplicand, or accumulator, record the accumulator (L13) in the indicated register during the entire word time. However, if the accumulator is specified, the accumulator will be transferred to the multiplicand as well because of the minimization on the multiplicand record gate.

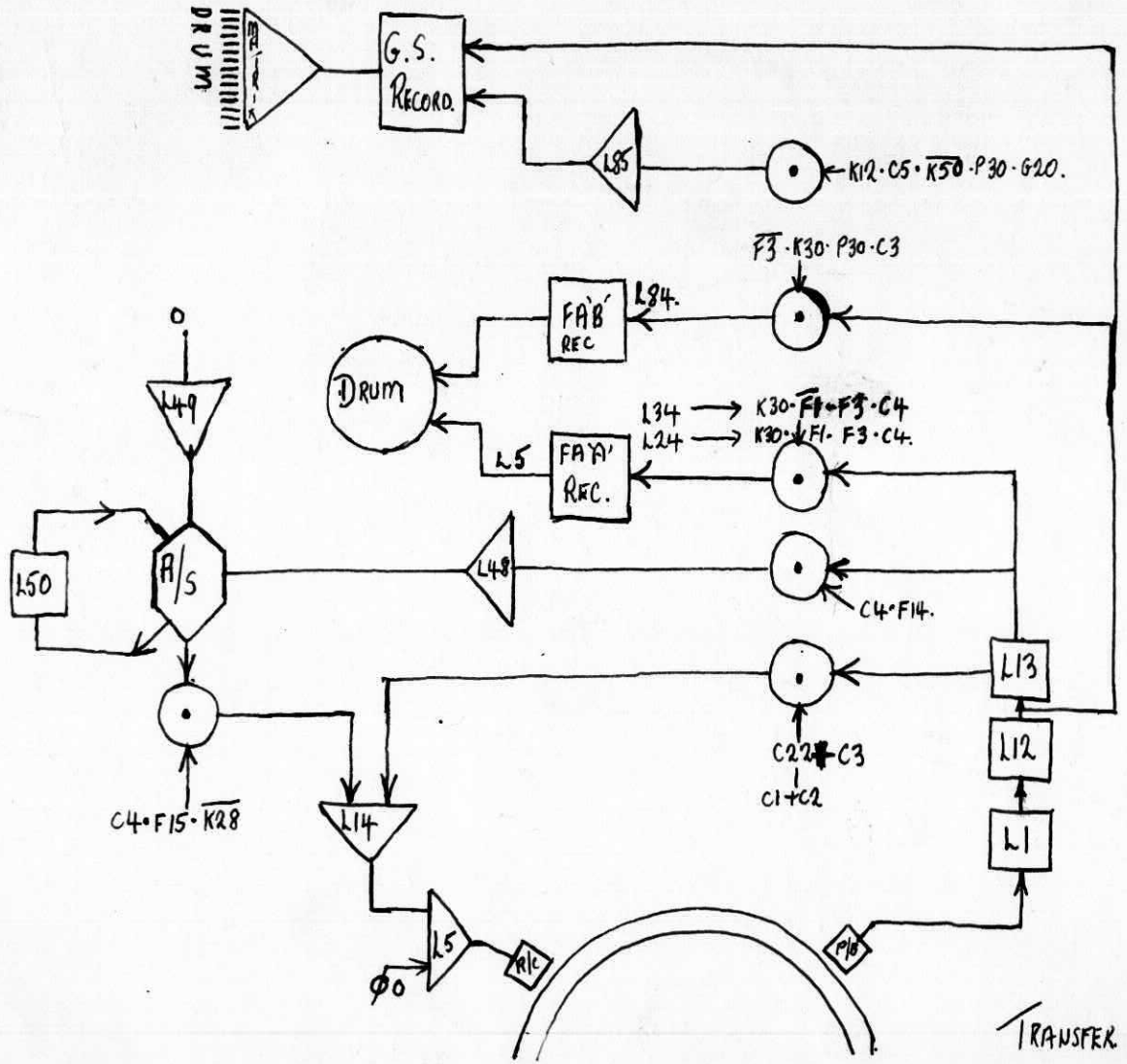
Coding



Decoded Signals

K12 = F16·F15'·F14·F13 Transfer to storage

K30 = K50·(K12 v K1·F11') Transfer to fast access



TRANSFER Acc → STORAGE

T A D R

K12 = transfer to storage (F16·F15'·F14·F13) = T

K30 = transfer to F.A.

= K50(K12 + K1·F11')

TRANSFER ACCUMULATOR TO STORAGE. T A S R. (STORE).

SIGNALS. K50 (fast access) = $(\overline{F1}+\overline{F2}+\overline{F3})F4\cdot F5'\cdot F6'\cdot F7'\cdot F8'\cdot F9'\cdot F10'$.

K20 (general storage) = $C2\cdot K50'\cdot G17'$

C5 (early record gate) = $C31+C4\cdot T31'$

K30 (storage → fast access) = $K50(K12+K1\cdot F11')$

K12 (transfer → storage) = $F16\cdot F15'\cdot F14\cdot F13$

(C2) locate proper sector - (see jump unconditional).

(C3) use C5 to start recording early accumulator (L12) into storage at T31 if the address in FA'8' or general storage

for FA'8' $L84 = L12\cdot K30\cdot C5\cdot \overline{F3}\cdot P30$.

for GEN. STORAGE. L85 (record gate) = $K12\cdot \overline{K50}\cdot P30\cdot C5\cdot G20$.

(C4) ① use C5 to continue recording early accumulator (L12) in storage if the address in FA'8' or gen. storage.

② if address is multiplier (FA.5), multiplicand (FA.4), or accumulator (FA.6.) record the accumulator in the indicated register (during the entire word time)

for acc. - $L14 = \underline{L51}\cdot \overline{K44}\cdot C4\cdot \overline{K2}\cdot \overline{K27}\cdot\cdot(\overline{K5}+F11)$

$L51 = L48$

$L48 = \underline{L13}\cdot C4\cdot F14$.

for FA.5: $L24 = \underline{L13}\cdot C4\cdot K30\cdot F3\cdot F1$.

for FA4: $L34 = \underline{L13}\cdot C4\cdot K30\cdot F3\cdot \overline{F1}$

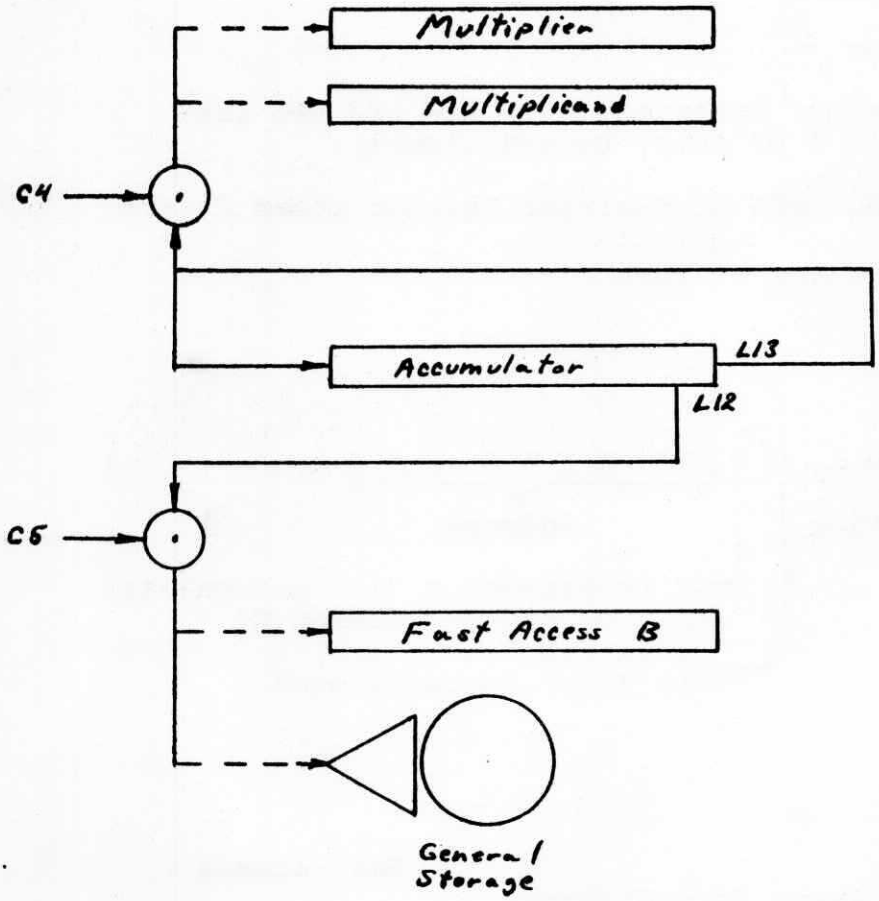
	F3.	F2.	F1
FA.4.	1	0	0
FA.6.	1	1	1

∴ also for address 006 - L13 is recorded in FA.4. as well.

K50 =
C20 = } See Jump, Unconditional — Fast Access / General Storage

C5 = C31 v C4·T31' — Early record gate
1 BIT EARLY

Block Diagram



TRANSFER STORAGE TO ACCUMULATOR

Description

This command reads the contents of the storage register specified into the accumulator.

Sequence

C2 1. Locate proper sector (see Steps 1 to 6 of Jump, Unconditional).

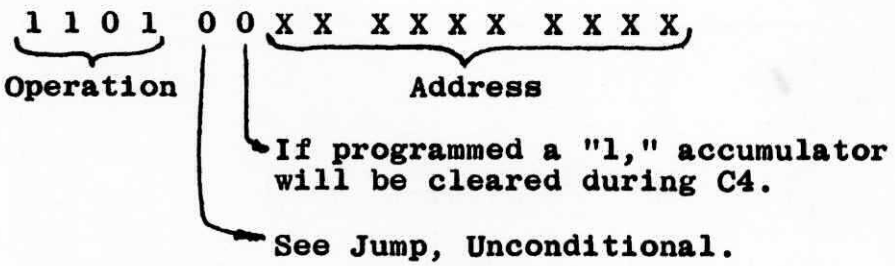
C3 1. Do nothing.

C4 1. Select proper information in the L83 net (see Steps 1 to 3 of Jump, Unconditional).

2. Record L83 into accumulator through adder B gate.

3. Hold adder A gate low.

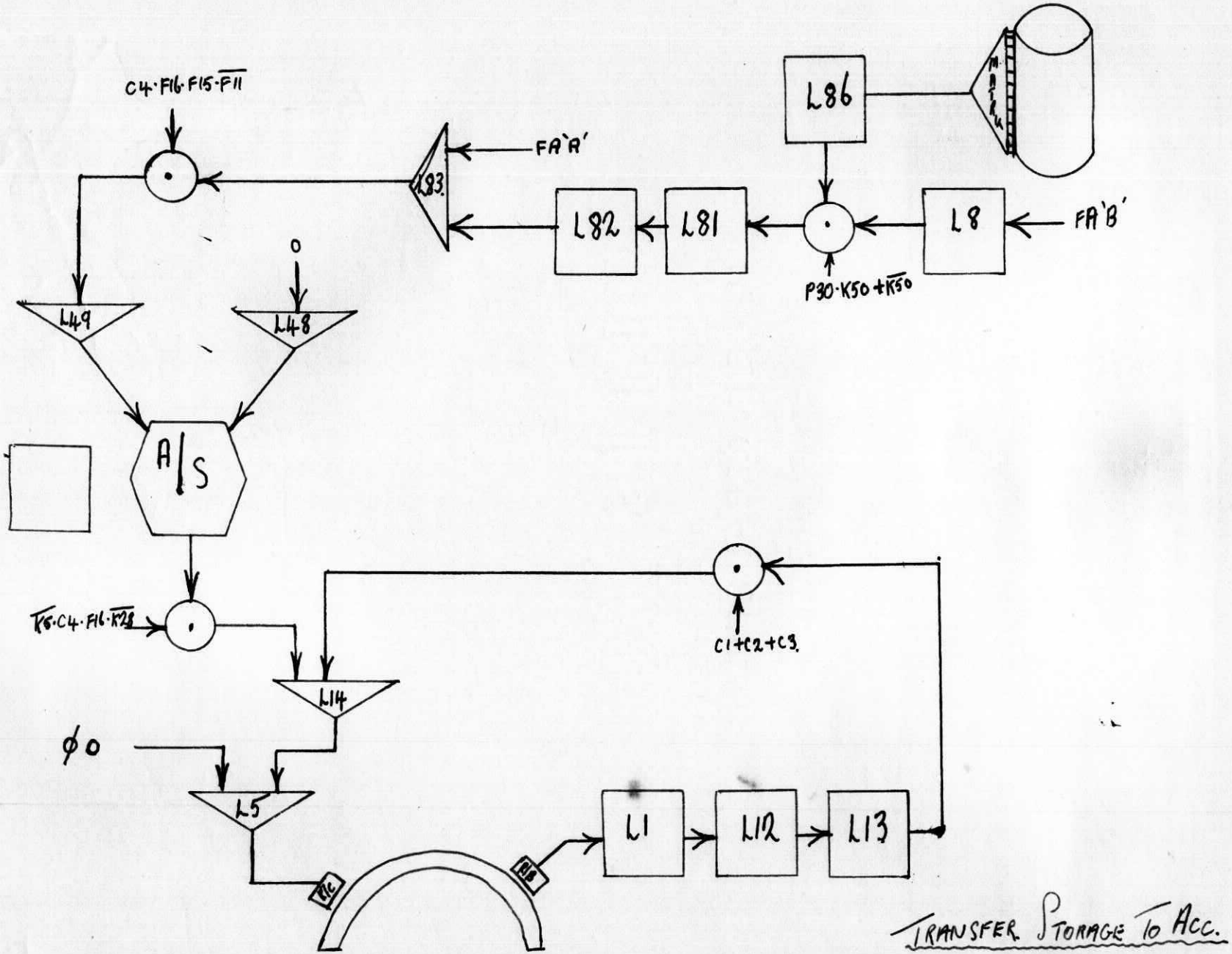
Coding



Decoded Signals

K50 = } See Jump, Unconditional — Fast access

C20 = } See Jump, Unconditional — General storage


C4·F16·F15·F11
FA`A'
L83
L86
L82
L81
L8
FA`B'
P30·K50+K50
0
L49
L48
A/S
K5·C4·F16·K28
C1+C2+C9
L14
φ0
L5
L1
L12
L13
TRANSFER STORAGE TO ACC.
VAR

TRANSFER STORAGE TO ACCUMULATOR: VIDEO

SIGNALS

K50 (fast access) = (F1'+F2+F3')·F4'·F5'·F6'·F7'·F8'·F9'·F10'

C20 (gen. storage) = K50'·G17'·C2.

SEQUENCE (under control of C20.)

(C2) (i) locate proper sector by comparing 25 (sector address flagged) with F8→F5 (sector address programmed) at F12. (ii) if F12 still reset at C2·T31 – set C3.
if F12 set at C2·T31 – continue sector comparison.

(C3) do nothing

(C4) (1) select proper information in L83 reg. (see steps 1→3 of jump unconditional.

FOR GEN STORAGE – L86→L81→L82→L83 (L83 = [L81·F1·F2·K50] + [L82(F1+F2)·K50 ↑ ϕ0→2.)

FOR FAST ACCESS "B" L7→L8→L81→L82→L83. (L83 = " ϕ3 ")

FOR FAST ACCESS "A" L1→Lx2→Lx3→L83.

(L83 = L13·K50·F3·F2
L83 = L23·K50·F3·F1
L83 = L33·K50·F3·F2'·F1'.)

(2) record L83 into accumulator through adder B gate. (L49)

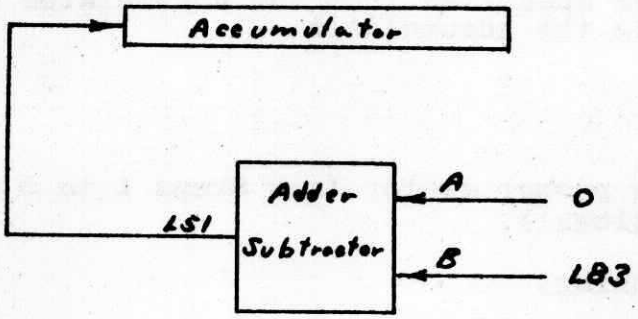
L49 = L83·C4·F16·F15·F11'

L51 = L49.

L14 = L51·C4·K2'·K27'·K5'

(3) hold adder A gate low.

Block Diagram



SUBTRACT

Description

The subtract command subtracts the contents of the storage register specified from the accumulator and stores the remainder in the accumulator.

Sequence

C2 1. Locate proper sector (see Steps 1 to 6 of Jump, Unconditional).

C3 1. Do nothing.

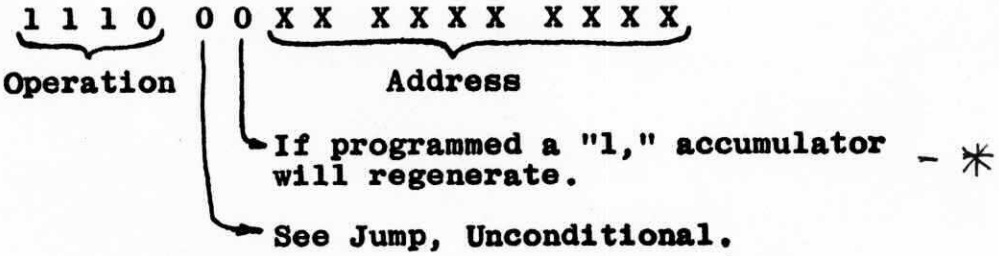
C4 1. Select proper information in the L83 net (see Steps 1 to 3 of Jump, Unconditional).

2. Subtract L83 (B gate) from accumulator (A gate).

3. Set carry flip-flop (L50) control for subtract (Q17).

4. Record result (L51) in accumulator.

Coding



Decoded Signals

K50 = } See Jump, Unconditional — Fast access

C20 = } See Jump, Unconditional — General storage

* gate on adder/subt will be closed for informat^n from L83 net. ∴ nothing will be subtracted.