IBM

# Customer Engineering Manuals Supplement

System/Unit 7094

Re: Form No. R23-2550-1

This Supplement No. S23-4001

Date October 11, 1962

Previous Supplement Nos. S23-4000

IBM 7094 DATA PROCESSING SYSTEM
CUSTOMER ENGINEERING INSTRUCTION - REFERENCE

This publication is a supplement to the 7094 Customer Engineering Instruction-Reference Manual, Form R23-2550-1. The enclosed section replaces corresponding material in the present manual.



*International Business Machines Corp., Product Publications Dept., Poughkeepsie, N. Y.*

PRINTED IN U.S.A

# INDEXING

## Table of Contents

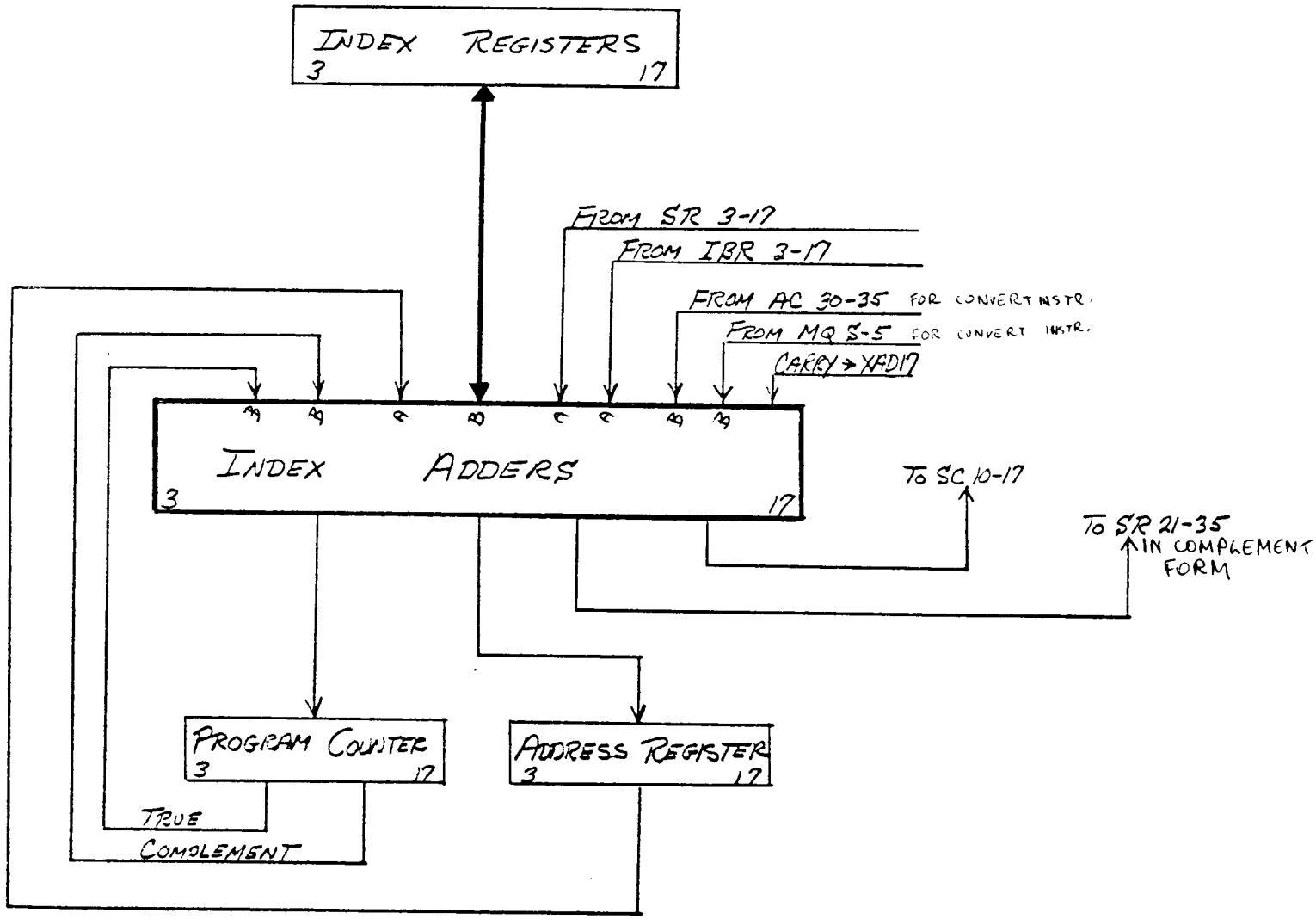
INDEX ADDERS (Figure )

The index adders (XAD) have been placed in the 7094 to: perform functions connected with indexing instructions and tagging operations; provide a means for incrementing the program counter and address register; and provide system data flow paths to or from the program counter, address register, shift counter, index registers, storage register, IBR, accumulator, and MQ.

This 15 position (3-17) index adder is similar in makeup and logic to that of the main adder. It has been separated into four groups, each having lookahead capabilities within itself as well as to the adjoining groups. Incrementing a value in the index adders is accomplished by injecting a "hot 1" or carry into XAD 17. Overflow conditions are recognized and remembered by means of an XAD 3 carry trigger.

Design of the circuitry is effected by two major factors: speed and economy of components. Because of this, variations occur in the layout and design of each group; the logical purpose of each group, however, remains the same.

XR 2




INDEX ADDER ROUTING

FIGURE _____

Bo 3/62

Index Adder Position (Figure )

Each adder position consists of basic elements performing specific functions: +AO blocks to test the two input values to be added; a minus cascode OR (-OE) circuit to determine a sum condition; a distributor element to route the sum output to various designated places under circuit routing control; and carry circuitry to the next adder position.

Input +AO Blocks

The +AO blocks, tied together as they are, test the in-phase and out-of-phase conditions of the two input values to be added and provide three important indications to the adder circuitry:

1. The two in-phase outputs tied together perform an exclusive OR function. A plus output indicates a neither/both input condition which is a complement exclusive OR ($\bar{E}$). A minus output, on the other hand, indicates an either input condition which is an exclusive OR (E).

2. In adder lookahead it is necessary to know if an adder position can generate a carry from outside inputs. The out-of-phase output of the upper +AO provides this generate (G) function. The output, when plus, indicates a both input condition and that the adder has the ability to generate (G) a carry from within itself. The output, when minus, indicates a neither/either condition which is the complement of generate ($\bar{G}$).

3. In adder lookahead it is also necessary to know if an adder position has the ability to propagate a carry from a next lower position through itself and to the next higher position. A moments thought shows that this ability exists if there is at least one input to the adder position. The out-of-phase output of the lower +AO, when plus, indicates an

either/both input condition; the adder position can propagate (P). The out-of-phase output, when minus, indicates a neither input condition and the adder position cannot propagate ($\overline{P}$).

The above symbols and definitions can be summarized as follows:

| FUNCTION | ADDER INPUT REQUIREMENTS |
|---|---|
| E - Exclusive OR | - Either input only |
| $\overline{E}$ - Complement exclusive OR | - Neither/both inputs |
| G - Generate | - Both inputs |
| $\overline{G}$ - Complement generate | - Neither/either inputs |
| P - Propagate | - Either/both inputs |
| $\overline{P}$ - Complement propagate | - Neither inputs |

Minus Cascode Exclusive OR Circuit (Figures & )

The adder sum output is determined by the minus cascode exclusive OR circuit. This circuit produces a minus (no sum) output for exclusive OR input conditions. Putting it another way, a sum output is produced when both inputs are alike; both plus or both minus. The two cases causing a sum will be:

1. An adder E (exclusive OR) and no input carry.
2. An adder E (comp exclusive OR) and an input carry.

The XAD sum line signal is sent to a distributor (DSU) block where it is, in turn, gated to appropriate computer circuitry.

Index Adder Carry (Figure )

Carrying within a particular group is accomplished by testing the input conditions to the various adder positions; it is not necessary to look at the sum output of a position to

determine if a carry will result. This lookahead technique reduces the amount of time necessary for the adder group to obtain a final value.

The adder circuitry, because of its design for speed and economy, is not always straightforward or consistant in the logic used to develope individual carries. Knowing what conditions are needed, however, can be a great help in following the actual systems ALD pages and circuitry. These conditions can be summarized as follows:

1. An output carry is produced if the adder position can generate (G) a carry from its two inputs. Only one condition produces this; both inputs being active.

2. An output carry is also produced if an input carry is received from the preceeding position and this adder position has the ability to propagate (P) a carry through itself. A moments thought shows that this propagate ability is present if either one or both of the inputs are active.

Knowing these two conditions for carry generation, refer to Figure showing positions 17-14 (group 1) of the index adders. An adder 17 carry ($+K_{17}$) results if:

1. There is a "hot carry" into adder 17 ($K_{HC}$) and adder 17 has the ability to propagate (P).

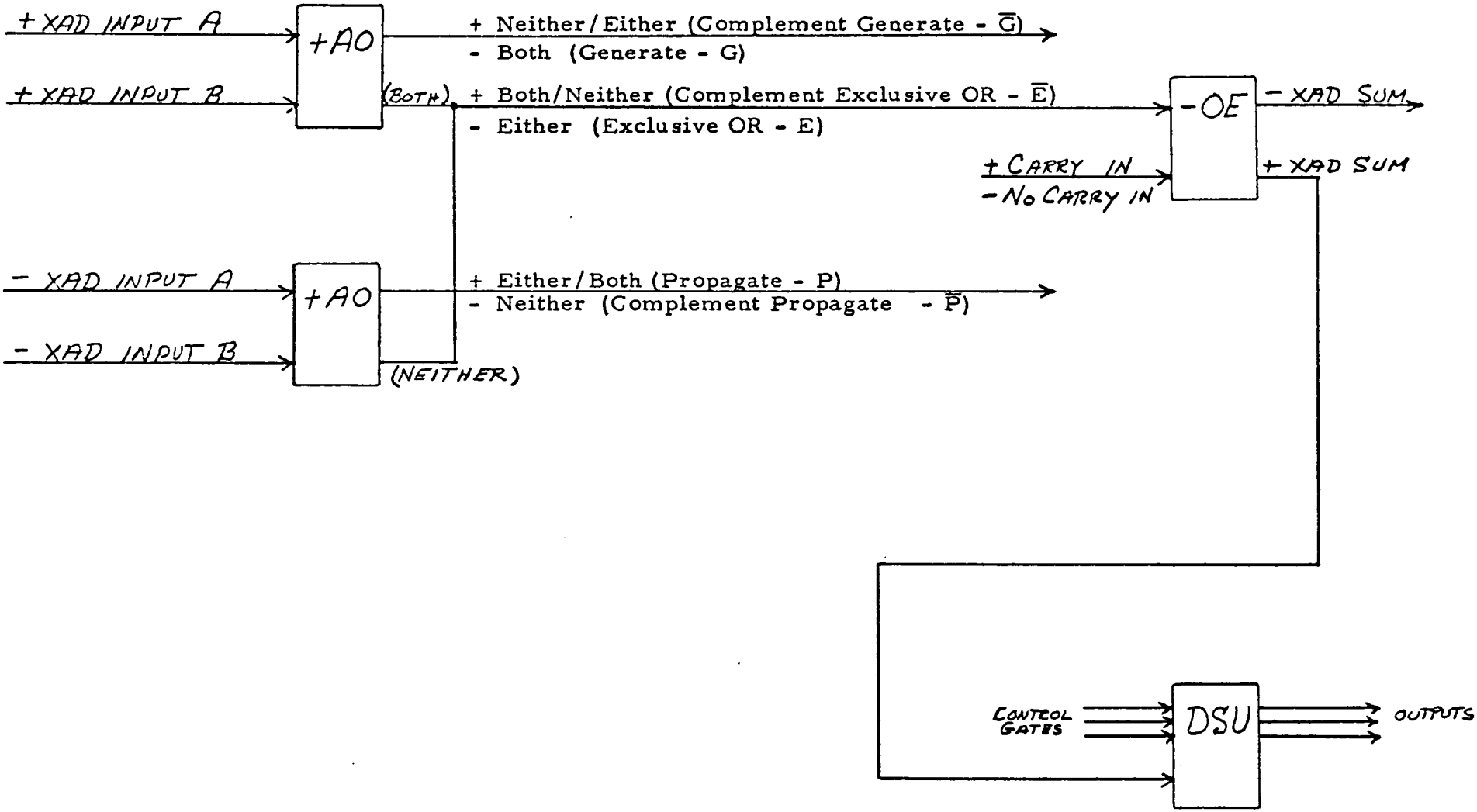
2. The two adder inputs produce complement exclusive OR ($\overline{E}$) and propagate (P) conditions. Both of these conditions occurring at the same time is another way of saying that both inputs were active and the adder position has the ability to generate (G) a carry.

Adder 16 carry takes on a reversal in logic. To obtain an adder 16 carry either or both inputs to the +A must be minus.

1. The lower leg is minus for a generate (G) condition.

2. The upper leg is minus when both inputs to the +O (16 Carry Generate) circuit are minus. These represent an exclusive OR adder input condition and no input carry from the preceeding position.

By taking positions 16 and 17 as typical examples and knowing the two basic rules for carry generation, there should be no trouble in following through the remaining positions of the index adder. Notice that there is no 14 carry generated as in the above cases; this is accounted for in the group carry lookahead circuitry.

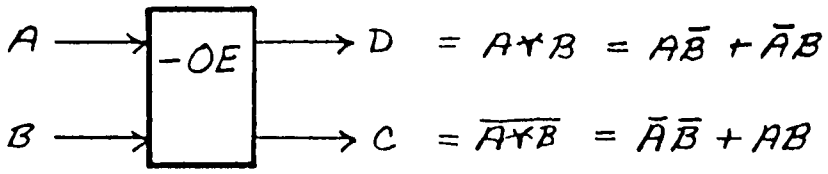



INDEX ADDER POSITION

FIGURE _____

XR 7

Bo 9/62




| INPUTS | | OUTPUTS | |
|---|---|---|---|
| A | B | C | D |
| + | + | + | − |
| − | + | − | + |
| + | − | − | + |
| − | − | + | − |

MINUS CASCODE EXCLUSIVE OR

FIGURE _____

BO 9/62

XC 8

Index Adder Lookahead (Figures & )

The speed with which the index adders can produce a final answer is dependent on its ability to signal carries from the lowest to the highest positions (i.e. from XAD 17 to XAD 3). This lookahead capacity is accomplished by determining each groups ability to either generate a carry from within itself or to propagate a carry from a lower group through itself and on to the next higher group. This procedure is similar to that previously explained for carry generation within group 1.

Using group 1 (Figure ) as an example, we see that at the same time that individual carries are being generated from position to position, additional circuitry is also looking at the individual generate (G), propagate (P) or exclusive OR (E) conditions to determine carry capabilities of the group as a whole. Three outputs are possible:

1. PG - Indicates that the group has the ability to propagate a carry through itself.
2. GG - Indicates that the group has the ability to generate a carry from its individual adder input conditions.
3. KG - Indicates a carry to be sent to the next higher group.

Group 1 being the lowest group produces two outputs. Both of these arise from the same internal conditions and indicate group generate (GG) and group carry (KG) conditions.

Group 2 can produce all three possible outputs: PG2, GG2, and KG2. KG2 occurs because of either of two conditions:

1. group 2 can generate a carry; or
2. group 1 has generated a carry and group 2 has the ability to propagate the carry through itself.

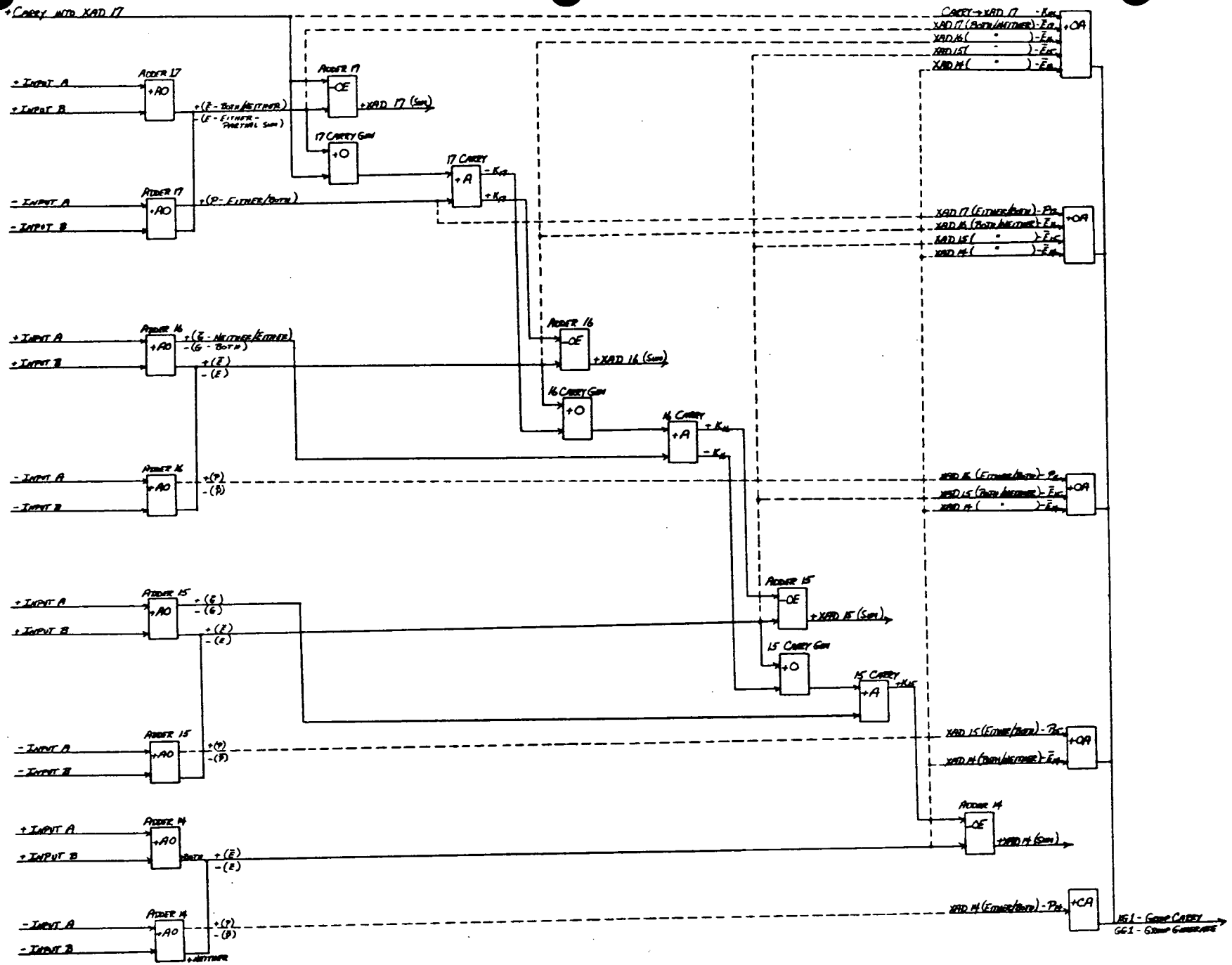
Group 3 can also produce PG3, GG3, and KG3 but notice that the group carry generation (KG3) can now occur as the result of three possibilities:

1. group 3 can generate a carry (GG3);

2. group 2 has generated a carry (GG2) and group 3 can propagate (PG3);

3. group 1 has generated a carry (GG1) and groups 2 and 3 can propagate (PG2 and PG3).

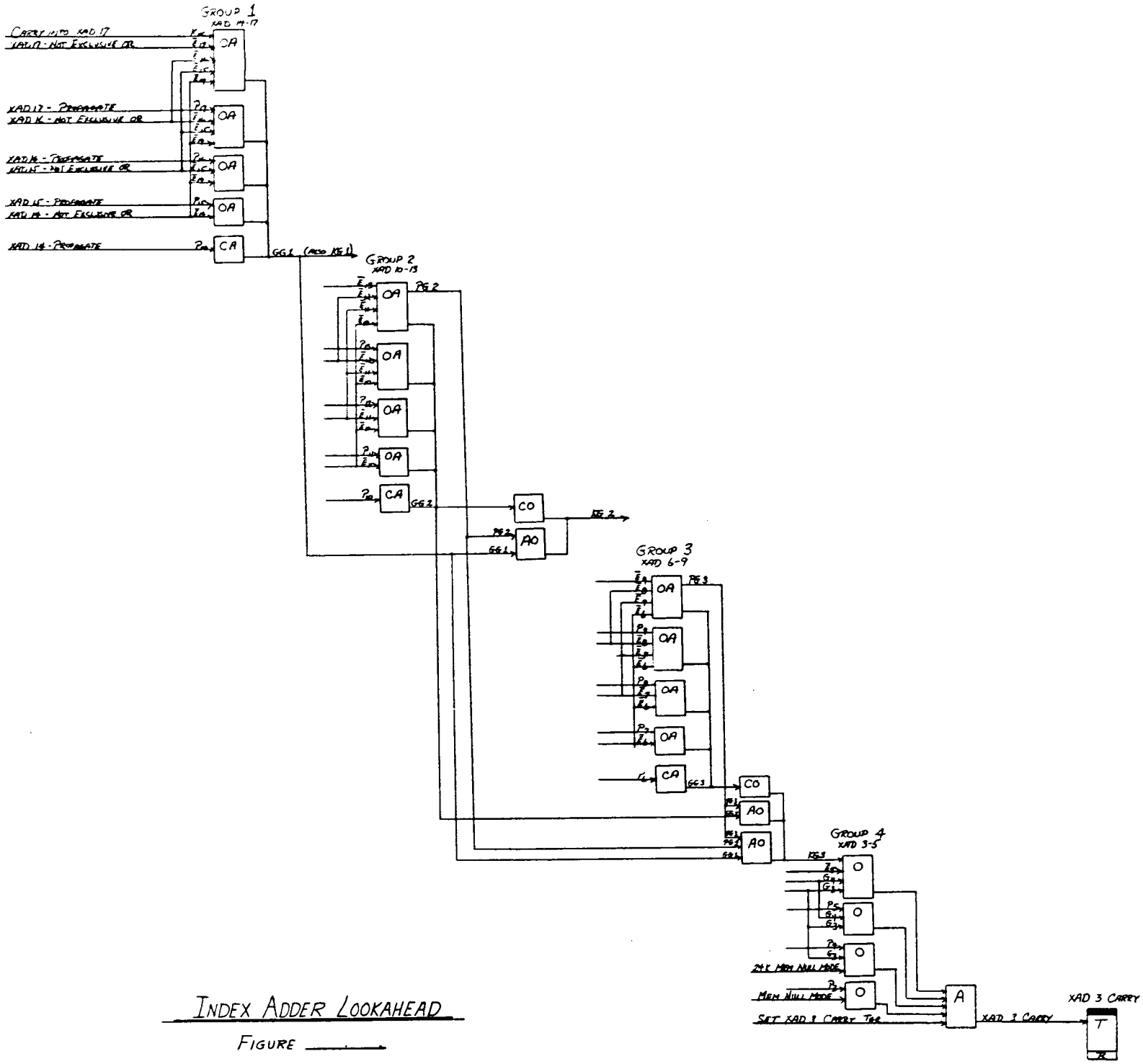
Group 4 receives the carry from group 3 (if there was one) and includes it with its own lookahead test circuitry. The logic is similar to that of the preceeding groups with exception of the two memory nullification conditions. Memory nullify mode is a 704 compatibility function and effectively halves memory to 16K by disregarding the XAD 3 outputs. 24K memory nullify mode effectively reduces the memory size to 8K by disregarding the outputs from both XAD positions 3 and 4.




INDEX ADDERS - GROUP 1

FIGURE ____

XR 11




INDEX ADDER LOOKAHEAD

FIGURE ——

INDEX INSTRUCTIONS

The 7094 has seven 15-position index registers and can operate in one of two index modes. Multiple tag mode (set by the EMTM instruction) causes the computer to work on a 3 index register basis and provides compatibility with 7090 programs. Instruction positions 18, 19 and 20 indicate any one or any combination of the three index registers. Multiple tagging in these positions defines the mode of operation and causes OR'ing of the index registers.

Use of all seven index registers can be accomplished by leaving the multiple tag mode (LMTM instruction). Instruction positions 18, 19 and 20 are decoded to indicate a single specific index register; index OR'ing is not possible.

A 3-position tag register accepts and retains the tag bits from either the storage bus or IBR depending on the various conditions of overlap. The output decoding of this register is dependent on whether the machine is in 3 or 7 index register mode.

A separate set of 15-position adders (index adders) have been placed in the 7094; they are similar in function but completely isolated from the normal 39-position main adder circuitry.

The output of the index registers is always gated to the index adders in 1's complement form. The 1's complement output is converted to 2's complement by a carry into index adder position 17. An indexable instruction need not be tagged; however, its address is always gated to the index adders and operated on by having an all 1's output from the index registers sent to the adders together with a carry to position 17. The net effect is to add 0's to the instruction address.

Six index instructions are provided to test the various registers and transfer if the specified conditions are met. In addition, 16 instructions are provided to transfer data

in true or complement form to or from the various index registers and the accumulator or core storage. In each case, either the address or decrement poriton can be specified. Two instructions also load the index register in true or complement form from its own address positions.

A summary of this second group of 18 index instructions is as follows:

A - To Index Register

1 - From core storage

| | | | |
|---|---|---|---|
| LXA | +0534 | - | Load Index from Address |
| LXD | -0534 | - | Load Index from Decrement |
| LAC | +0535 | - | Load Index from Address Complemented |
| LDC | -0535 | - | Load Index from Decrement Complemented |
| AXT | +0774 | - | Address to Index True |
| AXC | -0774 | - | Address to Index Complemented |

2 - From Accumulator

| | | | |
|---|---|---|---|
| PAX | +0734 | - | Place Address in Index |
| PDX | -0734 | - | Place Decrement in Index |
| PAC | +0737 | - | Place Address in Index Complemented |
| PDC | -0737 | - | Place Decrement in Index Complemented |

B - From Index Register

1 - To storage

| | | | | |
|---|---|---|---|---|
| | SXA | +0634 | - | Store Index in Address |
| | SXD | -0634 | - | Store Index in Decrement |
| * | SCA | +0636 | - | Store Index in Address Complemented |
| * | SCD | -0636 | - | Store Index in Decrement Complemented |

2 - To accumulator

| | | | | |
|---|---|---|---|---|
| | PXA | +0754 | - | Place Index in Address |
| | PXD | -0754 | - | Place Index in Decrement |
| * | PCA | +0756 | - | Place Index in Address Complemented |
| * | PCD | -0754 | - | Place Index in Decrement Complemented |

* New 7094 Instructions

To Index Register From Core Storage

The next six instructions make reference to data in a core storage location and load it into a specific index register/registers in either true or complement form.

Overlap is not allowed by an instruction in the next higher odd location. LXA and LXC require use of the address register at a time when it would conflict with the overlap operation; AXT and AXC are one cycle instructions.

Load Index From Address - LXA +0534 I, E (Figure )

This instruction makes reference to a core storage location and loads the specified index register with the contents of positions 21-35.

During I6 time, the storage bus is gated into the storage register and positions 21-35 routed immediately into the address register. At the beginning of the next E cycle, this address is sent to MAR. Address modification is not possible; gating circuitry which normally takes the address register to the index adders and back again is blocked. Decoding from the tag register does gate index register outputs into the index adders together with a carry to position 17; no logic is performed, however, because the index adder outputs are not gated back again to the address register.

During the E cycle, the storage bus is set into the storage register at 6-time and positions 21-35 are routed to the address register at 11-time. At the beginning of the next I cycle, the address register contents are routed to the index adders during I0 (D3) time and from there, set into the index register by an I2 CP set pulse. The set pulse resets the index register to clear out old information; the same pulse, delayed by circuitry (03.05.33.1), then overides the reset pulse and causes the index register to be set to the new value.

At the same time that the I0 (D3) pulse is gating the address register to the index adders, an A2 (D2) pulse is also bringing up "gate AR - XAD" on 03.06.06.1. This second line however, performs no logic in this operation.

Load Index From Decrement - LXD -0534 I-E (Figure )

This instruction makes reference to a core storage location and loads the specified index register with the contents of positions 3-17. The initial phase of the operation is similar to LXA; positions 21-35 of the LXD instruction are gated to the address register and out to MAR for the E cycle core storage reference. Address modification is blocked as previously explained in LXA.

During the E cycle, the storage bus is set into the storage register at 6-time and positions 21-35 routed to the address register at 11-time. The routing to the address register at this time, however, performs no logic during this operation.

At the beginning of the next I cycle, storage register positions 3-17 (decrement) are routed directly to the index adders by an I0 (D3) pulse; circuitry from the address register to the index adders is blocked. An I2 CP set pulse resets the selected index register and samples in the new value from the index adders to complete the operation.

Load Index From Address Complemented - LAC +0535 I, E (Figure )

This instruction makes reference to a core storage location and loads the specified index register with the complemented contents of positions 21-35.

The initial phase of the operation is similar to LXA; positions 21-35 of the LAC instruction are gated to the address register and out to MAR for the E cycle core storage reference. Address modification is blocked as previously explained in LXA.

During the E cycle, the storage bus is set into the storage register at 6-time and positions 21-35 are routed to the address register at 11-time. At the beginning of the next I-cycle, the address register contents are routed to the index adders by an A0 (D3) pulse and from there, set into the index register by an I2 CP set pulse. This pulse accomplishes both the resetting and setting of the selected index register.

At the same time that the I0 (D3) pulse is gating the address register to the index adders, an A2 (D2) pulse is also bringing up "gate AR - XAD" on 03.06.06.1. This second line, however, performs no logic in the operation.

At I2 time the index register contains the true value. In order to replace this with the 2's complement, the index register (which always comes out in 1's complement form) is returned to the index adders with a carry to position 17 at I4 (D2). An I5 CP set pulse places the complemented value back into the index register and the operation is completed.

Load Index From Decrement Complemented - LDC -0535 I, E (Figure )

This instruction makes reference to a core storage location and loads the specified index register with the complemented contents of positions 3-17.

The initial phase of the operation is similar to LXA; positions 21-35 of the LDC instruction are gated to the address register and out to MAR for the E cycle core storage regerence. Address modification is blocked as previously explained in LXA.

During the E cycle, the storage bus is set into the storage register at 6-time and positions 21-35 routed to the address register at 11-time. The routing to the address register at this time, however, performs no logic during this operation.

At the beginning of the next I cycle, storage register positions 3-17 (decrement) are routed directly to the index adders by an I0 (D3) pulse; circuitry from the address register to the index adders is blocked. An I2 CP set pulse resets the selected index register and samples in the new value from the index adders.

At I2 time, the index register contains the true value. In order to replace this with the 2's complement, the index register (which always comes out in 1's complement form) is returned to the index adders with a carry to position 17 at I4 (D2). An I5 CP set pulse places the complemented value back into the index register and the operation is completed.

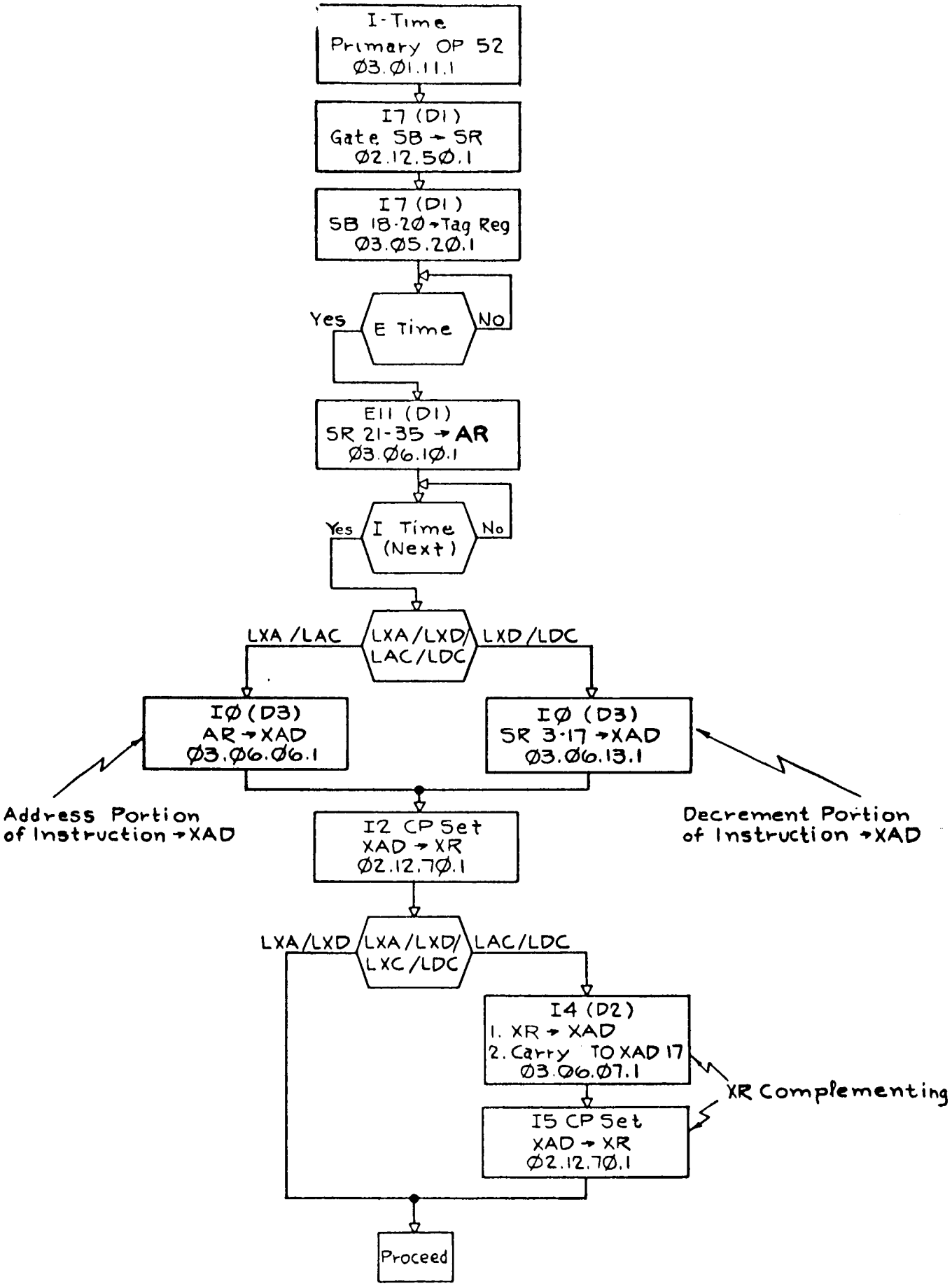



FIGURE LXA/LXD/LAC/LDC FLOW CHART

Address to Index True - AXT +0774 I (Figure )

This instruction loads positions 21-35 of the AXT into the specified index register in true form.

During I6 time, the storage bus is gated into the storage register and positions 21-35 immediately routed into the address register. Address modification is not possible (due to bits in PR 6 and 7); no gating is generated to take the address register to the index adders and back again (03.06.06.1). Decoding fromthe tag register, however, does gate the index register to the index adders with a carry to position 17, but no logic is performed.

AXT/AXC decoding forces an "end opn XR cntls" (02.15.64.1). The incremented program counter is sent to MAR and the computer takes a next I time. During the initial portion of this cycle an I0 (D3) pulse gates the address register to the index adders and from there into the index register at I2 CP set pulse time.

Primary Op 76 instructions normally take XAD 10-17 to the shift counter at I10 time (03.06.11.1). AXT decoding blocks this gating, however, and the shift counter remains reset.

Address to Index Complemented - AXC -0774 I (Figure )

This instruction loads positions 21-35 of the AXC into the specified index register in 2's complement form.

The first I cycle is identical to AXT -- the storage bus is gated into the storage register at 6 time and positions 21-35 immediately routed to the address register. The incremented program counter is gated to MAR and the instruction ends operation and proceeds to the next I cycle. Address modification is not possible.

During the next I cycle, an I0 (D3) pulse gates the address register to the index adders. An I2 CP set pulse places the value into the index register in true form. Complementing is accomplished by routing the index register (which always comes out in 1's complement form) to the index adders with a carry to position 17 at I4 (D2) time. An I5 CP set pulse gates the complemented value back into the index register and the operation is complete.

Primary Op 76 instructions normally take XAD 10-17 to the shift counter at I10 time (03.06.11.1). AXC decoding blocks this gating, however, and the shift counter remains reset.

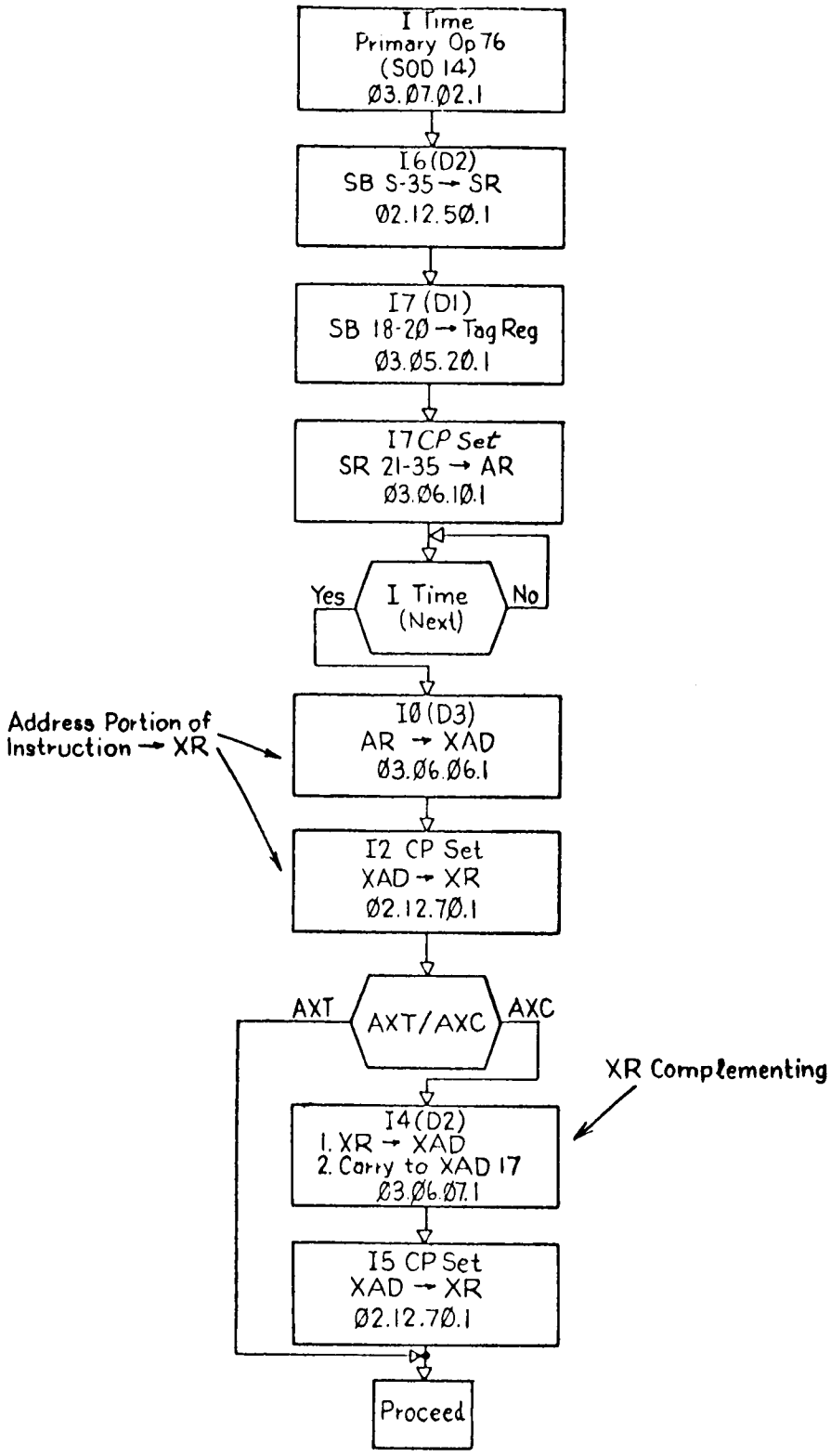



FIGURE AXT/AXC FLOW CHART

To Index Register From Accumulator

These next four instructions are concerned with placing information from the accumulator into a specified index register/registers. They require only one (I) cycle and, therefore, cannot be overlapped by an instruction in a next higher odd address.

Place Address In Index - PAX +0734 I (Figure )

The object of PAX is to place positions 21-35 of the accumulator (the address portion) into the specified index register. At I6 (D2) and I7 (D1) respectively the storage register and tag register are set from the storage bus, and SR positions 21-35 immediately routed to the address register. Setting the storage register and gating 21-35 to the address register has no logic at this time but functions because of normal I-time circuitry.

Address modification is not possible; gating circuitry which normally takes the address register to the index adders and back again is blocked. Tag register decoding, however, does gate the specified index register to the index adders along with a carry to XAD 17. This occurs as a normal I-time function but performs no logic because the XAD outputs are not gated back again to the address register.

The accumulator contents must be placed in the storage register in order to obtain routing paths for data-flow to the index register. To accomplish this, the accumulator is routed to the storage register at I10 time; at I11 time SR positions 21-35 are routed to the address register. These preliminary routings and functions during the first I cycle are common to all four POD 72 instructions (PAX, PDX, PAC, PDC).

During the next (I) cycle, the address register contents are routed through the index adders by an I0 (D3) pulse and set into the specified index register at I2 CP set time to complete the operation.

Place Decrement In Index - PDX -0734 I (Figure )

The PDX instruction places positions 3-17 of the accumulator (the decrement portion) into the specified index register. All of the initial I-time functions occur as explained for PAX; the accumulator contents are routed to the storage register at I10 time. Positions 21-35 of the SR are routed to the address register as a normal POD 72 function but performs no logic during the PDX instruction.

At I0 (D3) time of the next (I) cycle, positions 3-17 of the storage register are routed directly to the index adders and set in the specified index register at I2 CP set pulse time. I0 (D3), "gate AR-XAD", circuitry (03.06.06.1) is blocked by a PDX condition.

Place Address In Index Complemented - PAC +0737 I (Figure )

The PAC instruction places the complemented contents of accumulator positions 21-35 (address portion) into the specified index register. All of the initial I-time functions occur as explained for PAX. The accumulator contents are routed to the storage register at I10 time and SR 21-35, then, immediately routed to the address register at I11 time.

During the initial portion of the next (I) cycle, the address register contents are routed through the index adders by an I0 (D3) pulse and set into the specified index register at I2 CP set time. This places a true value in the index register.

Complementing is accomplished by routing the index register (which always comes out in 1's complement form) to the index adders with a carry to XAD 17 at I4 (D2) time. An I5 CP set pulse gates the complemented value back into the index register to complete the operation.

Place Decrement In Index Complemented - PDC -0737 I (Figure )

The PDC instruction places the complemented contents of accumulator positions 3-17 (decrement portion) into the specified index register. All of the initial I-time functions occur as explained for PAX; the accumulator contents are routed to the storage register at I10 time. At I11 time, SR positions 21-35 are routed to the address register as a normal POD 72 function but performs no logic during the PDC instruction.

At I0 (D3) time of the next (I) cycle, positions 3-17 of the storage register are routed directly to the index adders and set into the specified index register at I2 CP set pulse time. This places a true value in the index register.

Complementing is accomplished by routing the index register (which always comes out in 1's complement form) to the index adders with a carry to XAD 17 at I4 (D2) time. An I5 CP set pulse gates the complemented value back to the index register to complete the operation.

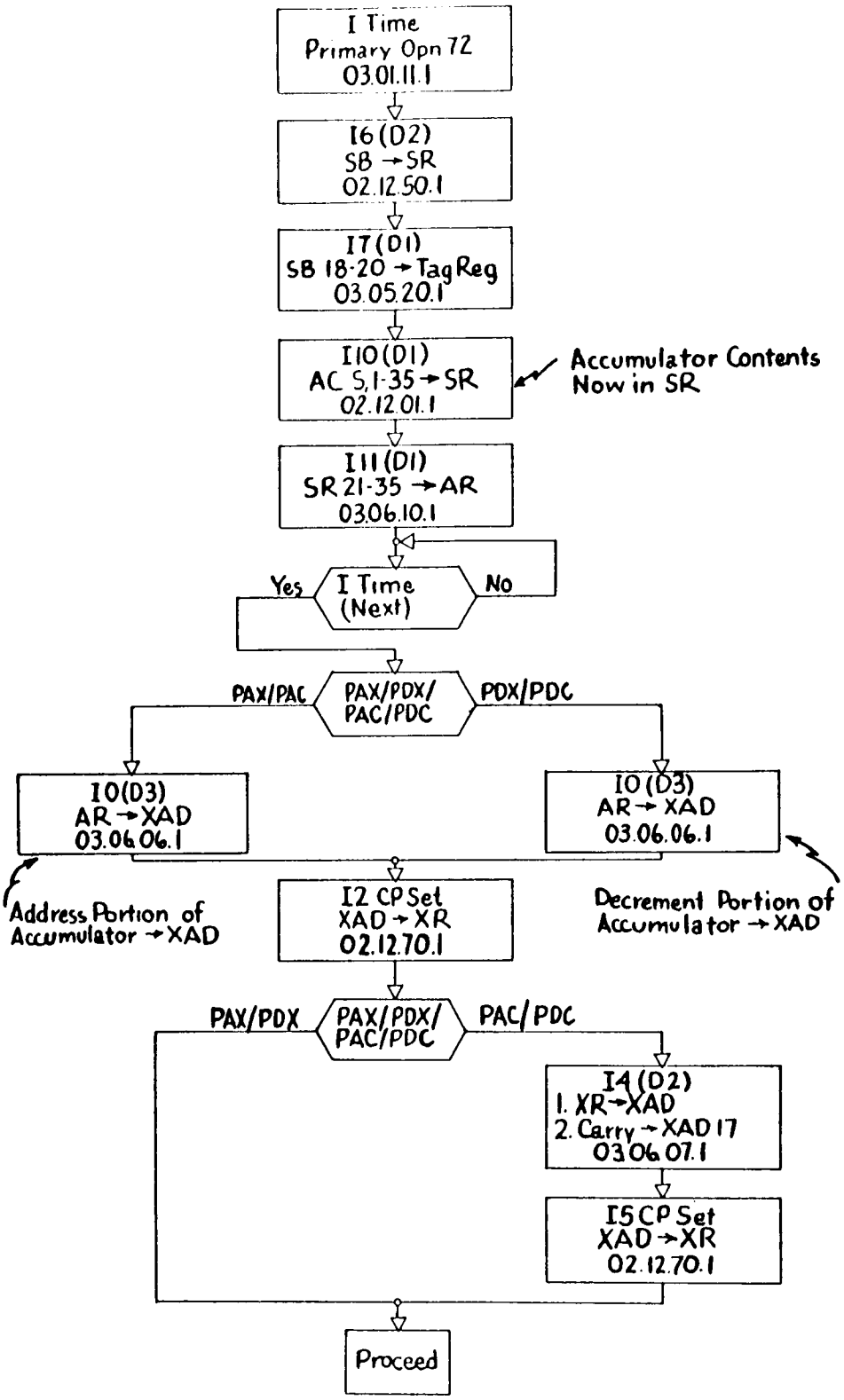



FIGURE PAX/PDX/PAC/PDC FLOW CHART

From Index Register To Core Storage

These next four instructions are concerned with taking information from the specified index register and storing it in true or complemented form in either the address or decrement portion of the indicated core storage location. Two cycles are required (I and E) for their execution. Overlapping is not permitted by an instruction in the next higher odd location.

Store Index In Address - SXA +0634 I, E (Figure )

The object of the SXA is to store the true value of the index register into the address portion of the specified core storage location. At I6 (D2) and I7 (D1) time respectively, the storage register and tag register are set from the storage bus and SR positions 21-35 immediately routed to the address register. At the beginning of the next E cycle, this value is sent to MAR for the E cycle core storage reference. Address modification is not possible; gating circuitry is blocked which normally takes the AR to the index adders.

Also, during the first I time, tag register decoding gates the specified index register contents to the index adders with a carry to XAD 17. This occurs at I9 (D2) as a normal I-time function; however, no logic is performed during the SXA instruction because of the adders are not gated beyond this point.

The index register contents must be routed to the storage register in order to obtain a data-flow path to the storage bus and core storage. A routing path is available from the index adders to the storage register but it must be remembered that this path has the capability of routing only the complement of the index adders.

During the early portion of the E cycle at E0 (D3) time, the index register is again gated to the index adders. Because the contents of the index registers are always brought out

in 1's complement form, the adders now contain the 1's complement of the value desired. An E1 (D1) pulse gates the complement of the index adders to positions 21-35 of the storage register; the true index register value is now in the SR and ready to be sent to core storage via the storage bus. "Store address" controls are active because of SXA operation decoding, and the address portion of core storage is modified without distrubing the remainder of that data word.

Multiple tagging is possible with this group of instructions. In order to maintain compatibility with previous systems, the specified index registers must be ORed together. Therefore, (for the SXA instruction) at the same time that the index adders are being gated to the SR, the ORed output is set into the specified index registers. The index registers now contain the 1's complement of the ORed values. (If multiple tagging is not specified, the index register contains the 1's complement of the original value.) The index register (or registers) is returned to a true value by gating the output to the index adders and back again during the following E4 (D2) time.

Store Index In Decrement - SXD -0634 I, E (Figure )

The SXD instruction stores the true value of the index register into the decrement portion of a specified core storage location. All of the initial I-time functions occur as explained for SXA. Positions 21-35 of the storage register are routed to the address register for core storage reference during the next E cycle.

During the early portion of the next E cycle at E0 (D3) time, the index register contents (which always come out in 1's complement form) are gated to the index adders where the complement of the index adders is routed and set into positions 21-35 of the storage register. Double complementing produces the true index register value in the SR.

The ORed complement of the index register (or registers) also replaces the original contents at E1 time; the true ORed value is restored during the following E4 (D2) time.

Positions 21-35 of the storage register must be moved into positions 3-17 in order to accomplish storing the desired value in the decrement of a core storage location. This swapping of the address to the decrement is accomplished by routing 18-35 of the SR to P-17 of the adders; the adders are then routed back to and set into the storage register. Positions 18-35 of the storage register are reset. The storage register is gated to core storage to complete the operation.

"Store decrement" controls are active because of SXD operation decoding, and the decrement portion of core storage is modified without distrubing the remainder of the data word.

Store Complement Of Index In Address - SCA +0636 I, E (Figure )

The SCA instruction stores the 2's complement of the index register value into the address portion of a specified core storage location. All of the initial I-time functions occur as explained for SXA but with one addition; the 2's complement is gated from the index adders and set into the index register at I10 time. The address register contains the core storage reference for the next E cycle.

During the early portion of the next E cycle at E0 (D3) time the index register contents are gated to the index adders. The index register contains the 2's complement of the original value; the output, however, is the 1's complement of that. The complement of the index adders is routed and set into positions 21-35 of the storage register. This double complementing effectively places the 2's complement of the original index register contents into the SR where it is placed on the storage bus and sent to core storage. "Store Address" controls allow modification of the address portion without disturbing the remainder of the data word.

At this point the index register contains a complemented value. Restoring the original contents is accomplished during the next I-time when the index register (or registers) is gated to the index adders with a carry to XAD 17. Taking this output back to the index register completes the operation.

Store Complement Of Index In Decrement - SCD -0636 I, E (Figure )

The SCD instruction stores the 2's complement of the index register value into the address portion of a specified core storage location. All of the initial I-time functions occur as explained for SXA but with one exception; the 2's complement is gated from the index adders and set into the index register at I10 time. The address register contains the core storage reference for the next E cycle.

During the early portion of the next E cycle at E0 (D3) time the index register contents are gated to the index adders. The index register contains the 2's complement of the original value; the output, however, is the 1's complement of that. The complement of the index adders is routed and set into positions 21-35 of the storage register. This double complementing effectively places the 2's complement of the original index register contents into the SR. Positions 21-35 of the storage register must be moved into positions 3-17 in order to accomplish storing the desired value in the decrement of a core storage location. This swapping of the address to the decrement is accomplished by routing 18-35 of the SR to P-17 of the adders; the adders are then routed back to and set into the storage register, where the contents are placed on the storage bus and set into the MDR. Positions 18-35 of the storage register are reset. "Store decrement" controls are active because of the SCD operation decoding and the decrement portion of core storage is modified without distrubing the remainder of the data word.

At this point the index register contains the complemented value. Restoring the original contents is accomplished during the next I time when the index register is gated to the index adders with a carry to XAD 17. Taking this value back to the index register completes the operation.

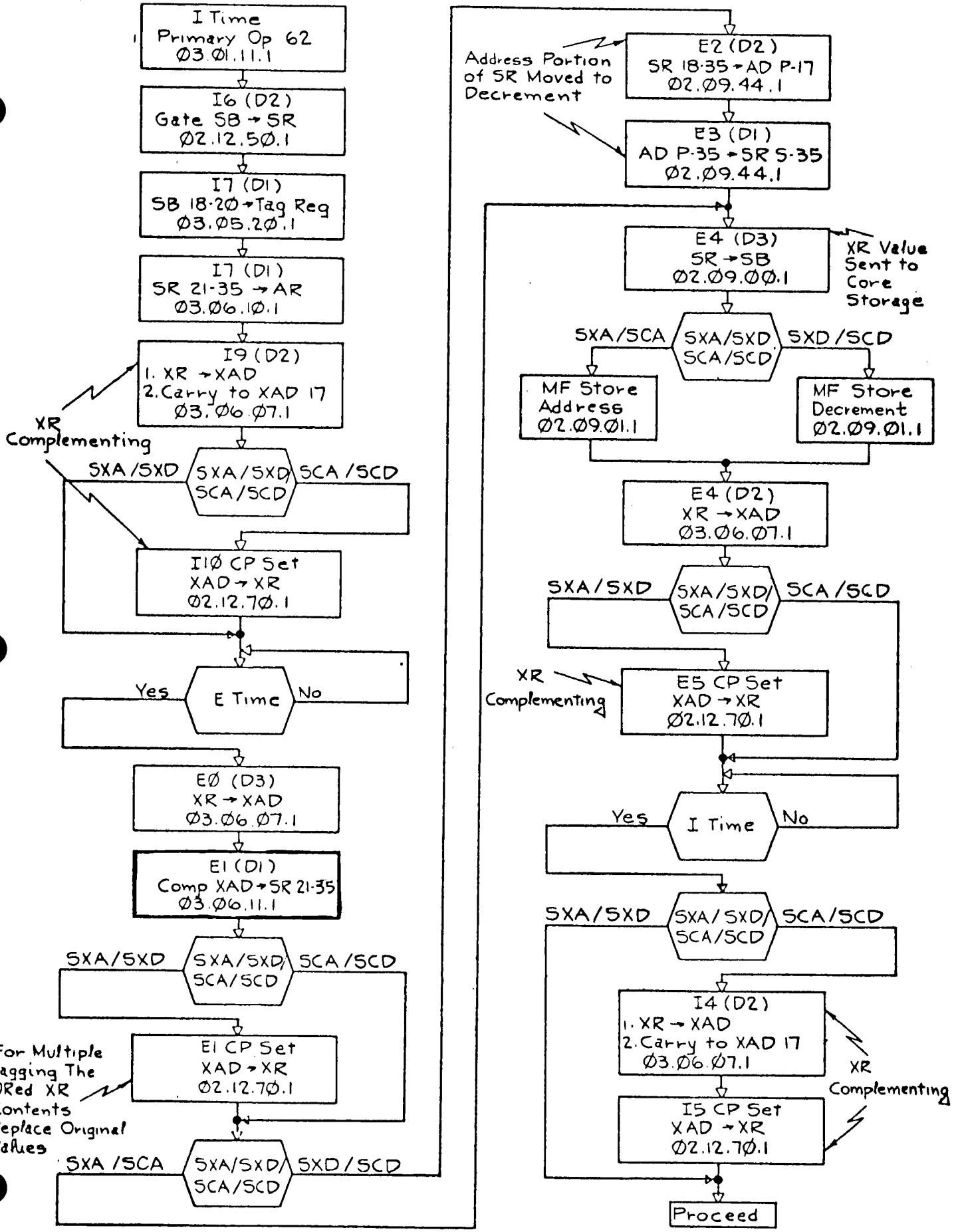



FIGURE SXA/SXD/SCA/SCD FLOW CHART

From Index Register To Accumulator

These next four instructions are concerned with taking information from the specified index register and placing it in true or 2's complement form in either the address or decrement portion of the accumulator. These are one (I) cycle instructions; therefore, overlapping is not permitted by an instruction in the next higher odd core storage location.

Place Index In Address PXA +0754 I (Figure )

The object of the PXA instruction is to place the true value of the index register into the address portion of the accumulator. The data flow path to the accumulator is: from the index register, through the index adders, to the storage register, and through the main adders into the accumulator.

At I6 (D2) and I7 (D1) time respectively, the storage register and tag register are set from the storage bus; SR positions 21-35 are immediately routed to the address register as a normal I-time function, but performs no logic in this operation.

During the first I-time, the tag register decoding gates the specified index register contents to the index adders with a carry to XAD I7 at I9 (D2) as a normal I-time function, however, no logic is performed during the PXA instruction because the adders are not gated beyond this point. The normal I9 (D2) gating of the address register to the index adders at this time is belocked because this is a non-indexible instruction.

During the early portion of the next (I) cycle at I0 (D3) time, the index register is again gated to the index adders. Because the contents of the index registers are always brought out in 1's complement form, the adders now contain the 1's complement of the value desired. An I1 (D1) pulse gates the complement of the index adders to positions 21-35 of the storage register; the true index register value is now in the storage register and ready to be sent to the accumulator via the main adders.

Multiple tagging is possible with this group of instructions. In order to maintain compatibility with previous systems, the specified index registers must be ORed together. Therefore, (for the PXA instruction) at the same time that the index adders are being gated the SR, the ORed output is set into the specified index registers. The index registers now contain the 1's complement of the ORed values. If multiple tagging is not specified, the index register contains the 1's complement of the original value. The index register (or registers) is returned to a true value by gating the output to the index adders and back again during I4 (D2) time.

With the index register value in the storage register, the operation is completed by routing the SR through the main adders and setting it into the accumulator during I4 (D3) time. The accumulator is cleared prior to the setting; therefore, if no tag is specified, the accumulator contains all 0's, at the end of the operation.

Place Index In Decrement - PXD -0754 I (Figure )

The PXD instruction places the true value of the index register into the decrement portion of the accumulator. The data flow path is similar to the PXA instruction; the initial I-time functions are identical.

During the early portion of the next (I) cycle, the index register is again routed to the index adders and from there to positions 21-35 of the storage register. The ORed 1's complement of the index registers (in the case of multiple tagging) also replaces their original contents at I1 (D1) time. The true value is restored again during the following I4 (D2) time.

With the index register value in the storage register, the operation is completed by routing positions 18-35 to positions P-17 of the main adders and fromthere into the decrement portion of the accumulator. The remaining positions of the accumulator are cleared.

Place Complement Of Index In Address PCA +0756 I (Figure )

The PCA instruction places the 2's complement of the index register into the address portion of the accumulator. The data-flow path is identical to PXA; the initial I-time functions are identical with one exception. When the contents of the index register (or registers) is gated out at I9 (D2) time with a carry to XAD I7, an I10 CP set pulse sets this 2's complement value back into the index register. This is necessary so that the complemented value will be available to be sent to the storage register.

During the early portion of the next (I) cycle the index register is again routed to the index adders and from there to the storage register. The 1's complement of the index register coming into the adders and the complement of the index adders being sent to the storage register effectively places the present index register value in the SR. The present value, however, is the 2's complement of the original index register contents. The true value is restored again during the following I4 (D2) time when the index register and a carry are routed to the index adders and back again.

With the 2's complement index register value in the storage register, the operation is completed by routing the contents of the SR through the main adders and into the accumulator.

Place Complement Of Index In Decrement PCD -0756 I (Figure )

The PCD instruction places the 2's complement of the index register into the decrement portion of the accumulator. The data-flow path is similar to the PXA Instruction; initial I-time functions are also the same except that the 2's complement is generated and set into the index register during I9 (D2) time.

During I0 (D3) time of the next cycle, the index register is routed through the index adders and set into posttions 21-35 of the storage register. The storage register now contains the 2's complement of the original index register contents. The true index register value is restored again during the following I4 (D2) time when the index register and a carry are routed to the index adders and back again to the index register.

With the 2's complement index register value in the storage register, the operation is completed by routing positions 18-35 to positions P-17 of the main adders and from there into the decrement portion of the accumulator.

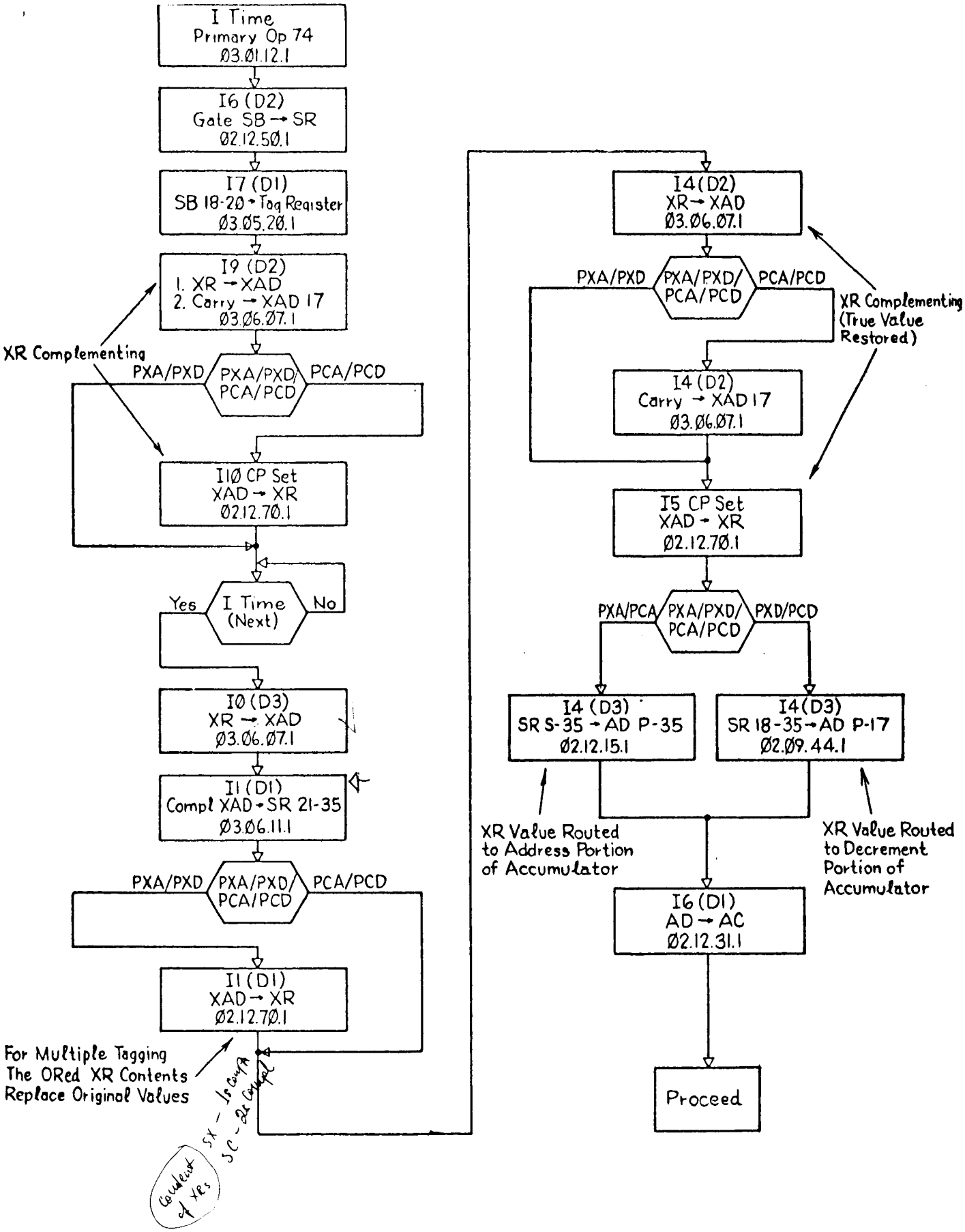



FIGURE PXA/PXD/PCA/PCD FLOW CHART

Conditional And Unconditional Index Transfers

These first four instructions (TIX, TNX, TXH and TXL) are conditional transfers concerned with comparing their decrement portion with the contents of a specified index register/registers. High, equal, or low indications cause the computer to either proceed to the next sequential instruction or transfer to the core storage location specified in the address portion of the instruction being executed. These are one (I) cycle instructions; therefore, overlapping is not permitted by an instruction in the next higher odd core storage location.

Bit recognition in positions 1 or 2 of the storage bus (02. 11. 40. 1) blocks sending positions 3-11 to the program register; instead, positions 1 and 2 are gated to PR positions 8 and 9. Decoding at this point recognizes these as non-indexable instructions (02. 12. 76. 1).

The last two instructions (TXI and TSX) are unconditional transfers. They perform the required functions and transfer without regard to index register contents. Being one cycle instructions, they cannot be overlapped.

Transfer On Index TIX +2000 1 (Figure )

The TIX instruction compares its decrement with the contents of the specified index register. If the number in the specified index register is greater than the decrement, the contents of the index register are reduced by the amount of the decrement and the computer transfers to location Y. When the number in the decrement is equal to or less than the decrement, no reduction is made and the computer takes the next instruction in sequence.

At I6 (D2) and I7 (D1) time respectively, the storage register and tag register are set from the storage bus; SR positions 21-35 are immediately sent to the address register to provide a transfer address if the decrement value is not greater than the index register.

At I8 (D2) time delayed, positions 3-17 of the storage register are routed to the index adders. At the same time, the index register and a carry to XAD 17 are also gated to the adders. (Because this is a non-indexable instruction, circuitry is blocked that would normally take the address register to the index adders.)

A ripple carry out of XAD3 sets an XAD3 carry trigger and indicates that the decrement value is equal to or greater than the index register. Under these conditions the program counter is gated to MAR and the computer continues with the next sequential instruction. No carry from XAD3 indicates that a successful reduction is possible, and the adders are gated back to the index register. The result from the adders is the 2's complement of the true value; this is corrected during the next I4 (D2) time when the index register and a carry to XAD 17 are gated to the index adders and back again.

With no XAD3 carry, a "set condition met" trigger is turned ON which causes address register gating to MAR, and the computer proceeds at the transfer address. During the following I2 (D2) time the address register is gated to the index adders as a flow-path to the program counter.

If multiple tagging is specified with this instruction, a successful reduction replaces the index registers with their ORed values minus the decrement. If a successful reduction is not possible, the index registers retain their original contents.

Transfer On No Index TNX -2000 I (Figure )

The TNX instruction compares its dec rement with the contents of the specified index register. If the number in the index register is greater than the decrement, the contents of the index register are reduced by the amount of the decrement and the computer proceeds to the next instruction in sequence. When the number in the index register is equal to or less than the decrement, no reduction is made and the computer transfers to location Y.

The sequence of operation for this instruction is similar to TIX except that the conditional transfer circuits are activated when there is a carry from XAD3.

Transfer On Index High TXH +3000 I (Figure )

The TXH instruction compares its decrement with the contents of the specified index register. If the number in the specified index register is greater than the decrement, the computer transfers to location Y. If the number in the index register is less than or equal to the decrement, the computer takes the next instruction in sequence. Index register contents are not altered.

Execution of TXH is similar to TIX; at I8 (D2) time delayed, SR positions 3-17 are added to the 2's complement from the index register to determine if there will be a ripple carry from XAD3. This carry is the only indication needed; the index adders are not returned to the index register. A carry causes the program counter to be gated to MAR and the computer proceeds in sequence; no carry (index register high condition) causes the address register to be gated to MAR and the computer transfers to location Y.

If a transfer is successful, the address register is routed through the index adders and set into the program counter during the next I time.

Transfer On Index Low Or Equal TXL -0300 I (Figure )

The TXL instruction compares its decrement with the contents of a specified index register. If the number in the index register is greater than the decrement, the computer takes the next instruction in sequence. If the number in the index register is equal to or less than the decrement, the computer transfers to location Y.

Execution of this instruction is similar to TNX except that the index adders are not returned to the index registers. An XAD3 carry (index register equal or low condition) indicates a successful transfer and causes the address register to be gated to MAR. No carry causes the program register to be gated to MAR nad the computer proceeds to the next instruction is sequence.

A successful transfer causes the address register to be routed through the index adders and to the program counter during the next I time.

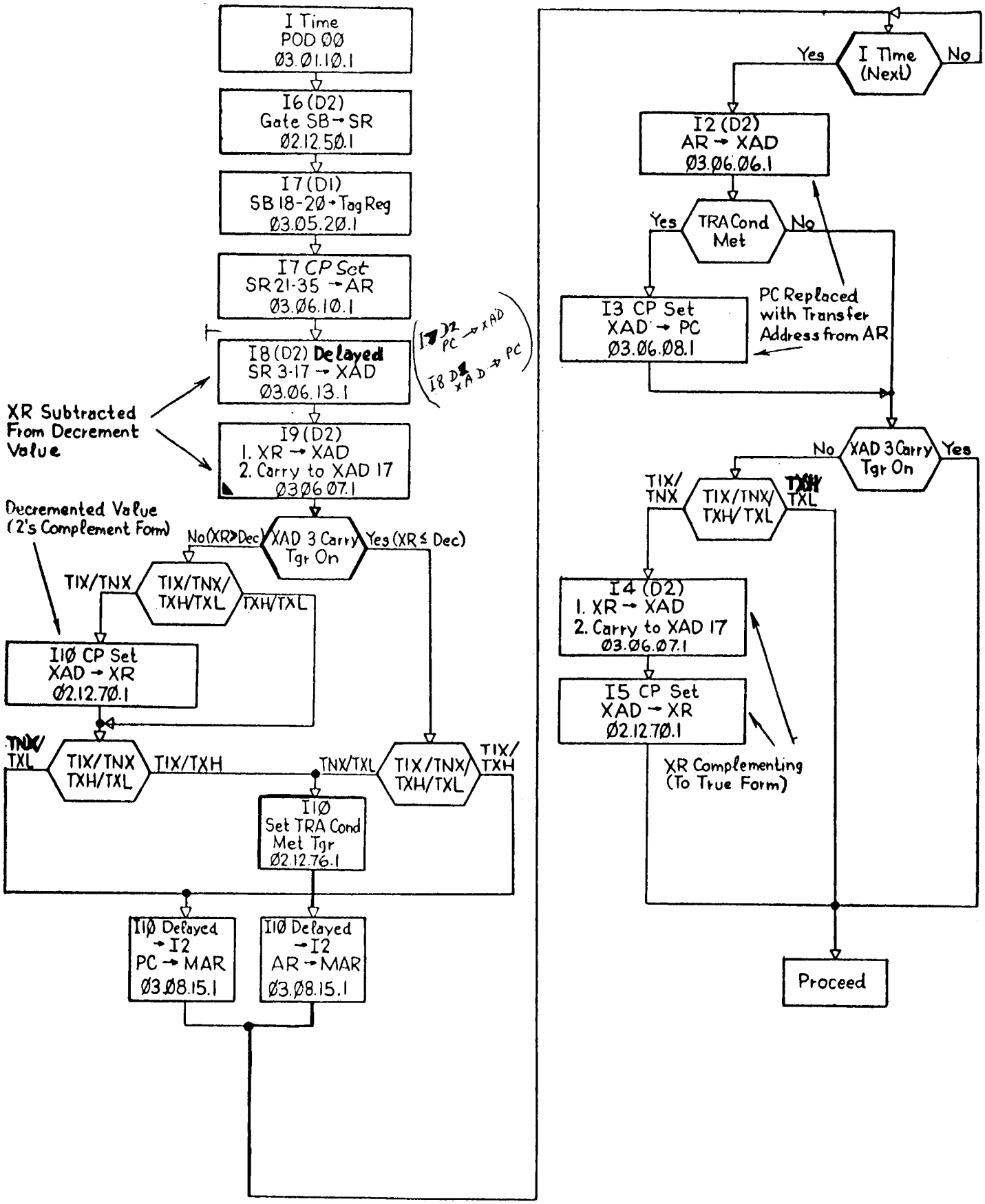



FIGURE TIX/TNX/TXH/TXL FLOW CHART

XR45

Transfer With Index Incremented TXI +1000 I (Figure )

TXI is an unconditional transfer to the location specified in positions 21-35 of its address field. In addition, the decrement portion is added to the specified index register. This is a one (I) cycle instruction; therefore, overlapping is not permitted by an instruction in the next higher odd core storage location. Bit recognition in positions 1 or 2 of the storage bus (02.11.40.1) blocks sending positions 3-11 to the program register; instead, positions 1 and 2 are gated to PR positions 8 and 9. Decoding (02.12,76.1) recognizes this as an non-indexable instruction.

At I6 (D2) and I7 (D1) times respectively, the storage register and tag register are set from the storage bus; SR positions 21-35 are immediately sent to the address register to provide the transfer address when the address register is gated to MAR at I10 time. Because this is a non-indexable instruction, normal AR to XAD gating is blocked.

Index register contents always come out in 1's complement form and if added to a number at this time would effectively accomplish subtraction. Because of this, it is necessary to cycle the index register through the index adders with a carry to XAD 17. This is accomplished at I9 (D2) time and places the 2's complement of the original value into the index register. The final addition of the decrement to the index register is performed during the next I cycle.

Because this is an unconditional type transfer instruction, the "transfer conditions met" trigger is turned ON at I10 time and the contents of the address register are gated to MAR. During the following I2 (D2) time the address register is gated to the index adders and from there into the program counter.

During the next (I) cycle incrementing of the index register is accomplished by gating positions 3-17 (decrement portion) of the storage register to the index adders together with the index register and a carry to XAD 17. Setting the index adders back into the index register completes the operation.

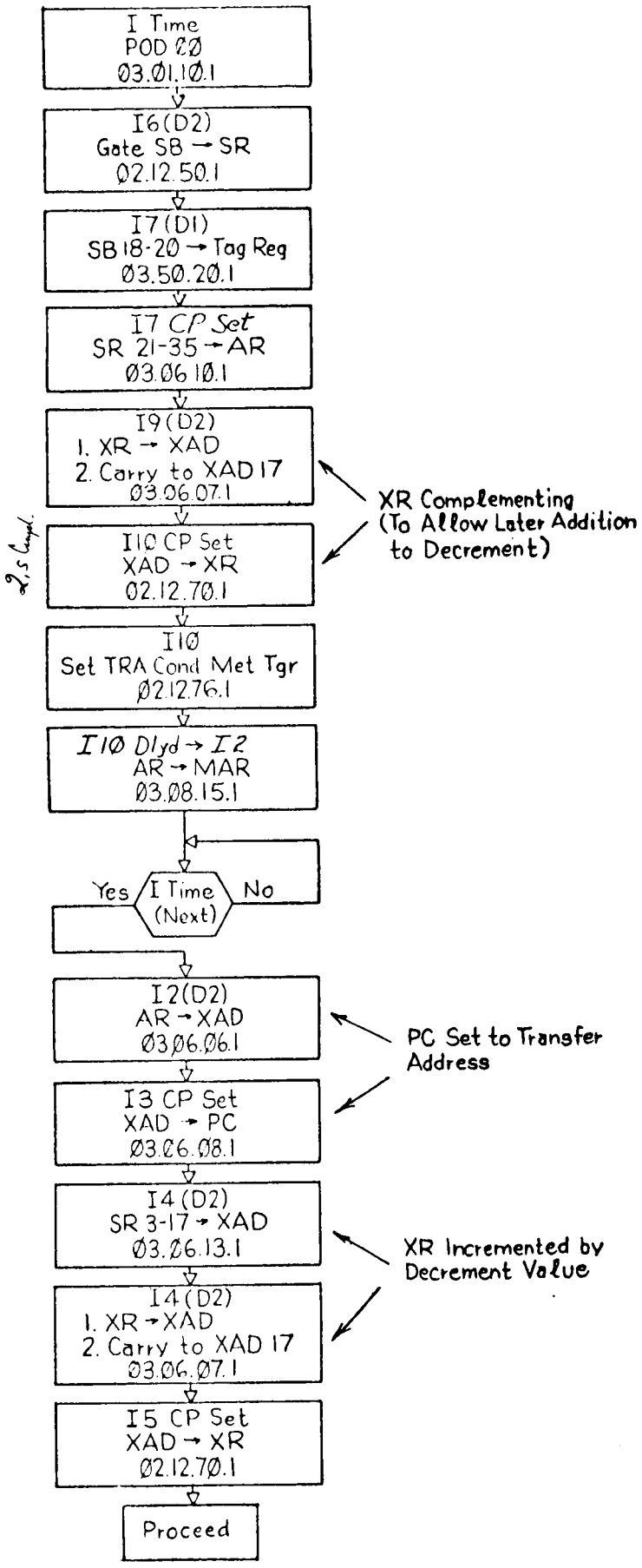



FIGURE TX1 FLOW CHART

Transfer And Set Index TSX +0074 I (Figure )

The TSX instruction places the 2's complement of the program counter (the location of the TSX instruction) into the specified index register and transfers to the location specified in positions 21-35 of the TSX instruction. This is a one (I) cycle instruction; therefore, overlapping is not possible by an instruction in the next higher odd core storage location. Decoding of PR 6 and 7 (02.12.76.1) defines this as a non-indexable instruction.

At I6 (D2) and I7 (D1) times respectively, the storage register and tag register are set from the storage bus, and SR positions 21-35 immediately sent to the address register. This is the transfer address which is used as the core storage reference for the next cycle.

To store the present value of the program counter, the normal stepping circuitry which occurs during I7 (D2) time must be blocked. The program counter is gated to the index adders in the usual manner but the index adders are blocked from being routed back again to the program counter.

The program counter can be gated to the index adders in either true or complement form. The 2's complement is made available to the index register by gating the complement of the program counter to the index adders with a carry to XAD 17. Normal gating of the address register to the index adders at I9 (D2) time is blocked by TSX (03.06.07.1); normal gating of the address register to the index adders is also blocked because this is a non-indexable instruction (03.06.06.1).

At I10 time the "transfer conditions met" trigger is uncoditionally set ON and the address register is gated to MAR. During the early portion of the next (I) cycle, the address register value is gated to the index adders and set into the program counter to complete the operation.

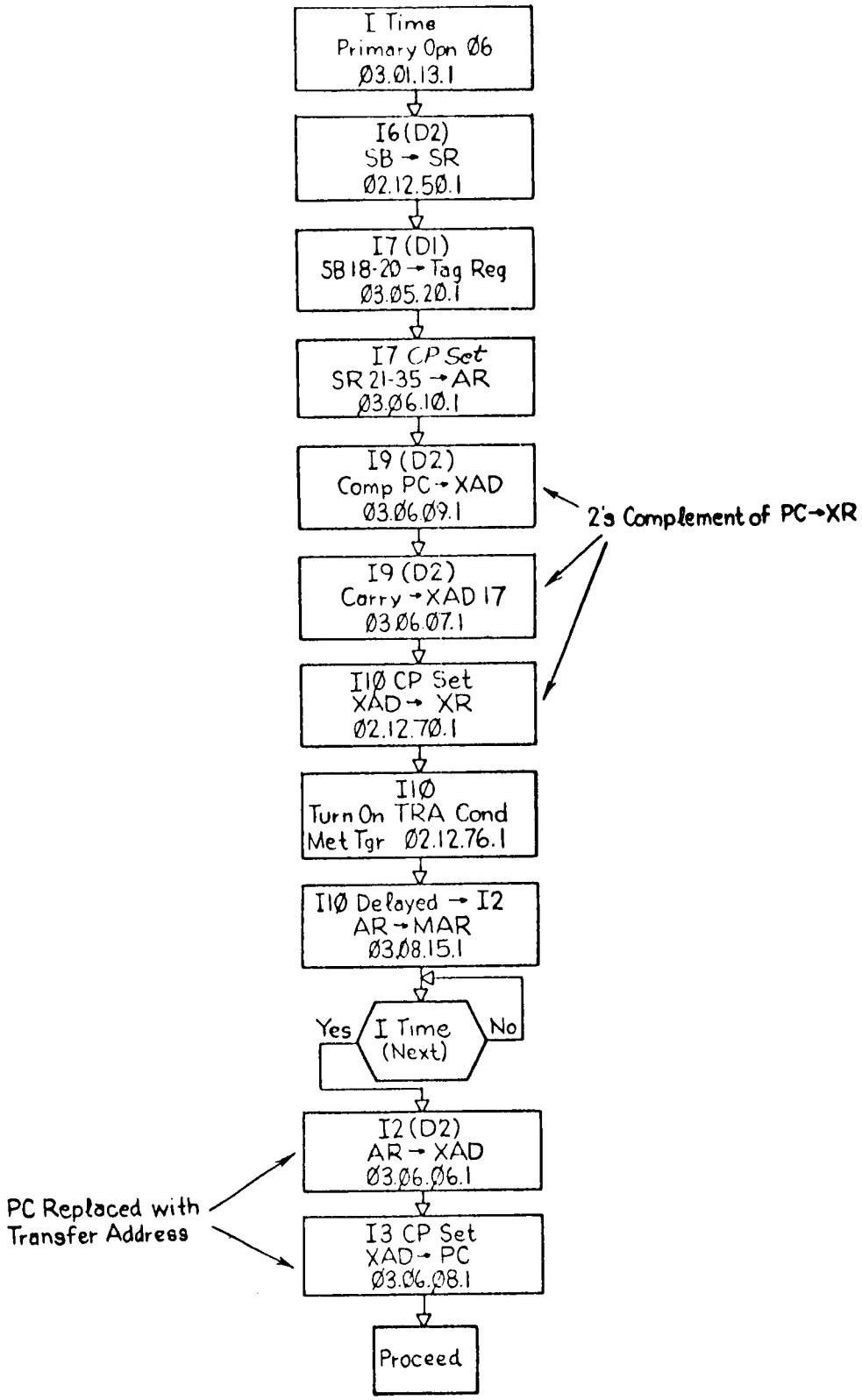



FIGURE TSX FLOW CHART

COMMENT SHEET

IBM 7094 DATA PROCESSING SYSTEM

CUSTOMER ENGINEERING INSTRUCTION-REFERENCE, FORM S23-4001

FROM

NAME _____

OFFICE NO. _____

FOLD

CHECK ONE OF THE COMMENTS AND EXPLAIN IN THE SPACE PROVIDED

FOLD

☐ SUGGESTED ADDITION (PAGE , TIMING CHART, DRAWING, PROCEDURE, ETC.)

☐ SUGGESTED DELETION (PAGE )

☐ ERROR (PAGE )

EXPLANATION

CUT ALONG LINE

FOLD

FOLD

NO POSTAGE NECESSARY IF MAILED IN U.S.A.

FOLD ON TWO LINES, STAPLE, AND MAIL

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 81

POUGHKEEPSIE, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
P.O. BOX 390
POUGHKEEPSIE, N.Y.

ATTN: CE MANUALS, DEPARTMENT 296

CUT ALONG LINE

FOLD

FOLD

10/62:200-56

STAPLE

STAPLE