DRI #2354

FINAL REPORT

DEVELOPMENT OF THE CIRCUIT DESIGN FROM THE LOGIC DESIGN OF A RANDOM WALK MACHINE

NsG-518

Project DRI 622

September 1966

UNIVERSITY OF DENVER

FINAL REPORT

DEVELOPMENT OF THE CIRCUIT DESIGN FROM THE LOGIC DESIGN OF A RANDOM WALK MACHINE

NsG-518

Project DRI 622

University of Denver

September 1966

SUBMITTED BY:

William D. Lansdown
Assistant Professor
Electrical Engineering

ABSTRACT

Laplace's equation, important in physical sciences, is a difficult equation to solve for arbitrary boundaries and boundary conditions. The author in his Ph. D. thesis describes the logic design of a small, high speed, random walk machine to solve Laplace's equation.

To convert the logic diagrams to circuit diagrams, a commercial circuit family is chosen to implement the logic. The basis of choice of a circuit family is a figure of merit, $f = C D$, where C is the initial cost of the logic units and D is the propagation delay through a shift register bit, a frequently used element in the machine. The minimum value of f is the most desirable value. The circuit diagrams were then developed from the logic diagrams on the basis of the circuit family chosen.

A control procedure is outlined in consideration of the computer the Department of Electrical Engineering expects to order shortly. The computer is found to be fast enough to control the random walk machine at the speed goals set for the random walk machine.

Estimated costs based upon the number and cost of the logic units required is given. Proposals submitted and to be submitted are discussed.

The appendix contains the circuit diagrams developed for the Random Walk machine along with the original logic diagrams.

ACKNOWLEDGEMENTS

The author acknowledges the work of Mr. Kenneth L. Trieu who has done a large share of the work of translating the logic diagrams to circuit diagrams.

The support of the National Aeronautical and Space Administration through Research Grant NsG-518 is gratefully acknowledged.

TABLE OF CONTENTS

1. INTRODUCTION

Laplace's equation, a well known and important equation in field theory, is the solution to the potential within a charge-free region for arbitrary boundary conditions. It describes also incompressible fluid flow and the steady state temperature distribution in solids and numerous other physical phenomena.

At present virtually the only practical methods of solution are field plotting, which has been done for years past, and iterative solutions of the analogous finite difference equation on digital computers. Other methods are known, and among these are random walk or Monte Carlo solutions, see Brown.[1] Solutions of this type have been performed on punch card equipment, see Yowell[10] and on digital computers, see Todd.[9] However, these are uneconomical even with modern high speed machines, see Forsythe[2] and Lansdown.[6] Hirai,[5] Sugiyama,[8] and their coworkers at the Osaka City University have built a random walk machine, which though slow has moderately high speed capabilities.

Lansdown[6] in his Ph. D. thesis gives the logic diagrams of a special purpose random walk machine which has a speed/cost factor of approximately 1000 times that of the IBM 7090 computer. The machine is to be controlled and have its output processed by a general purpose computer which can be used on a time shared basis if the controlling computer is sufficiently fast. This machine has speed capabilities between 10 and 20 times that of the machine of Hirai and Sugiyama because of using flip-flop register memory and counters instead of core memory. This type of design has further advantages in that the extension to three dimensions should be much more economical than a machine of the other type.

This project has had as its aim the carrying on of the development of the random walk machine. The logic diagrams of the before mentioned thesis have been transformed to circuit designs. The procedure for processing the data output of the random walk machine has been outlined for the expected Department of Electrical Engineering GE/PAC 4020 control computer. The control computer must process the output at least as fast as it is generated. This latter was not done in detail awaiting the actual order of the 4020 controller. Otherwise it will be necessary to use the University B5500, though this will not be as convenient due to the necessity of the B5500 being run by a regular operator.

One proposal has been submitted to N.S.F., though it was rejected. Further proposals will be submitted to obtain support to purchase components and support graduate students to complete the random walk machine.

2. DEVELOPMENT OF CIRCUIT DIAGRAMS

2.1 Choice of Circuit Family

Although the choice of circuit family and the transformation of logic are closely related, they will be discussed separately. In the summer of 1965 several families of commercial logic units were considered. Of these, two families had appreciably better performance from other lines considered. These were the Fairchild CTμL-952 through 957 and the Motorola MC351 through 362.

Since the RW machine is largely made up of parallel, circular, left and right shift registers, one bit of shift register was used for comparison purposes. A figure of merit was devised which was:

$$f = C\,D$$

when

c = cost of 1 bit of shift register

d = total propagation delay

The circuit family with the lowest figure of merit has the lowest cost per random step, the basic step of operation of the RW machine. The following table gives the figures of merit:

Line of Logic		C	D	f
Motorola	MC351 through 362	$13.77	22	303
Fairchild	CTμL-952 through 957	$16.07	13.9	224

These figures differ from the figures given in the progress report of 1 April 1965 through 31 September 1965 due to a recalculation of the costs and propagation delays. The Fairchild line has a better figure of merit, though it has a higher cost per bit.

Factors that will influence these figures is the price decrease as integrated circuit (IC) prices drop. Other factors that favor the Fairchild line in this application is the smaller number of IC's required: 2 1/3 Fairchild IC's per bit are required to 4 3/5 Motorola IC's. This means fewer units and fewer connections which will be an appreciable simplification. Also, manufacturers are developing several bits of shift register all in one IC, and this may cause, also a further decrease in price.

Thus the Fairchild CTμL 952-957 line was chosen to develop the circuit diagrams from the logic diagrams in the author's thesis.[3]

2.2 Transformation and Minimization

The logic diagram in the thesis were given in AND-OR-NOT logic where the Fairchild CTμL lines uses AND-NOT logic. This is sufficient in itself as the OR can be realized with AND's and NOT's. However, the Fairchild line has also the OR function by tying outputs together; this is called OR tying. Thus the replacement of AND-OR-NOT with AND-NOT-OR tie is virtually a 1-1 replacement which simplifies the transformation greatly.

The logic diagram were quite abbreviated in places, that is, a counter was shown as a single block. The circuit diagrams included here give the entire circuit (except that only 1 of the 28 identical memory rings are shown). This is particularly true not only in the case of counters, but also multiple AND's and registers and even more so in flip-flop memory.

Logic minimization programs were not listed in the B5500 program library, nor in IBM's SHARE listing, even though it is known to the writer that IBM has them for in-house use. Furthermore it did not seem likely that there would be great savings of components, so this was not considered further.

Should other families of logic become more economic at the time of construction, it will be possible to make the transition relatively easily. For example, AND's followed by OR's can be replaced directly with NAND's followed by NAND's.

3. INNERFACE AND OUTPUT DATA PROCESSING PROCEDURE FOR THE GE/PAC 4020 COMPUTER

The innerface between the RW machine and the GE/PAC 4020 will be nothing more than circuitry to change between the different levels of the two machines, and this will be very simple. Since the 4020 is a process controller it has provision for voltage level inputs.

Since the 4020 has yet to be ordered (though it seems quite likely that it will within 3 month's time), the output data processing procedure was devised only in flow diagram form. Refer to Figure 3.1 which shows an arbitrary boundary. The broken segments above AA are the upper boundary and the segments below are the lower boundary. Each of the lattice points intersected by segments representing the upper boundary have unique x axis values for the non-vertical segments U2, U3, and U5. For these points it is necessary only to add a constant to the X counter value to give a unique memory word location for each lattice point. To denote that the X value corresponds to a vertical segment, the memory word corresponding to that X value can contain a negative value. Furthermore that negative value can now be subtracted from the counter and an appropriate constant added to give a group of cells, each correpsonding to each lattice intersection with the boundary. This is shown in the flow diagram of Figure 3.2.

The numbers at the upper left hand corner of the blocks in Figure 3.2 are estimates of the memory cycles required to perform the block. The maximum number of cycles is 51 which requires 82μs of 4020 time per random walk. The average number of cycles is 41 which requires 66μs of 4020 time per random walk. The average time per random walk is 378μs. Thus 312 of the 378μs are available for other programs on a time shared basis. This shows that the GE/PAC 4020 is more than adequate as a controller for the RW machine.

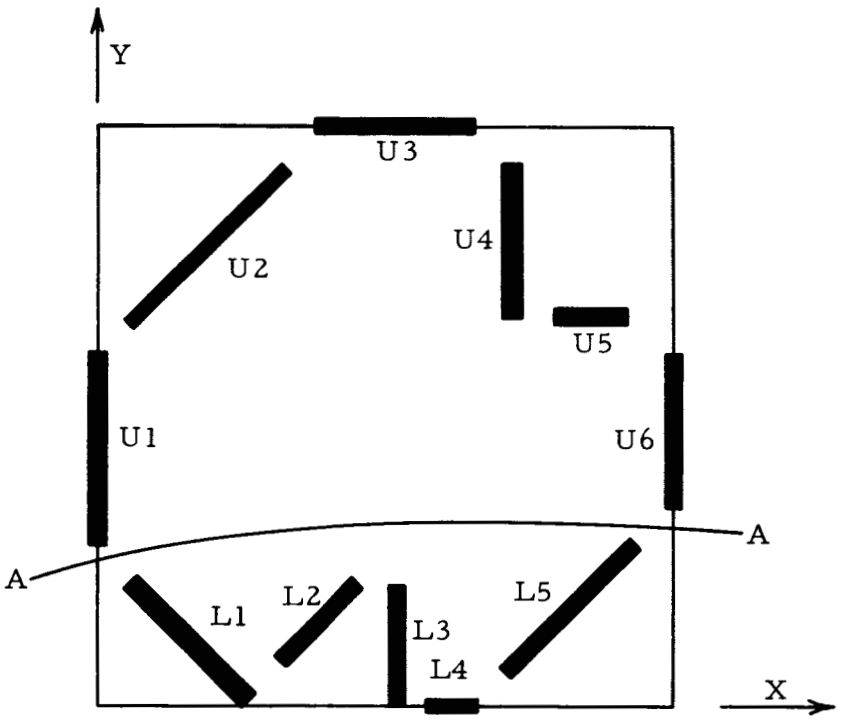


Figure 3.1. Illustrative Boundary

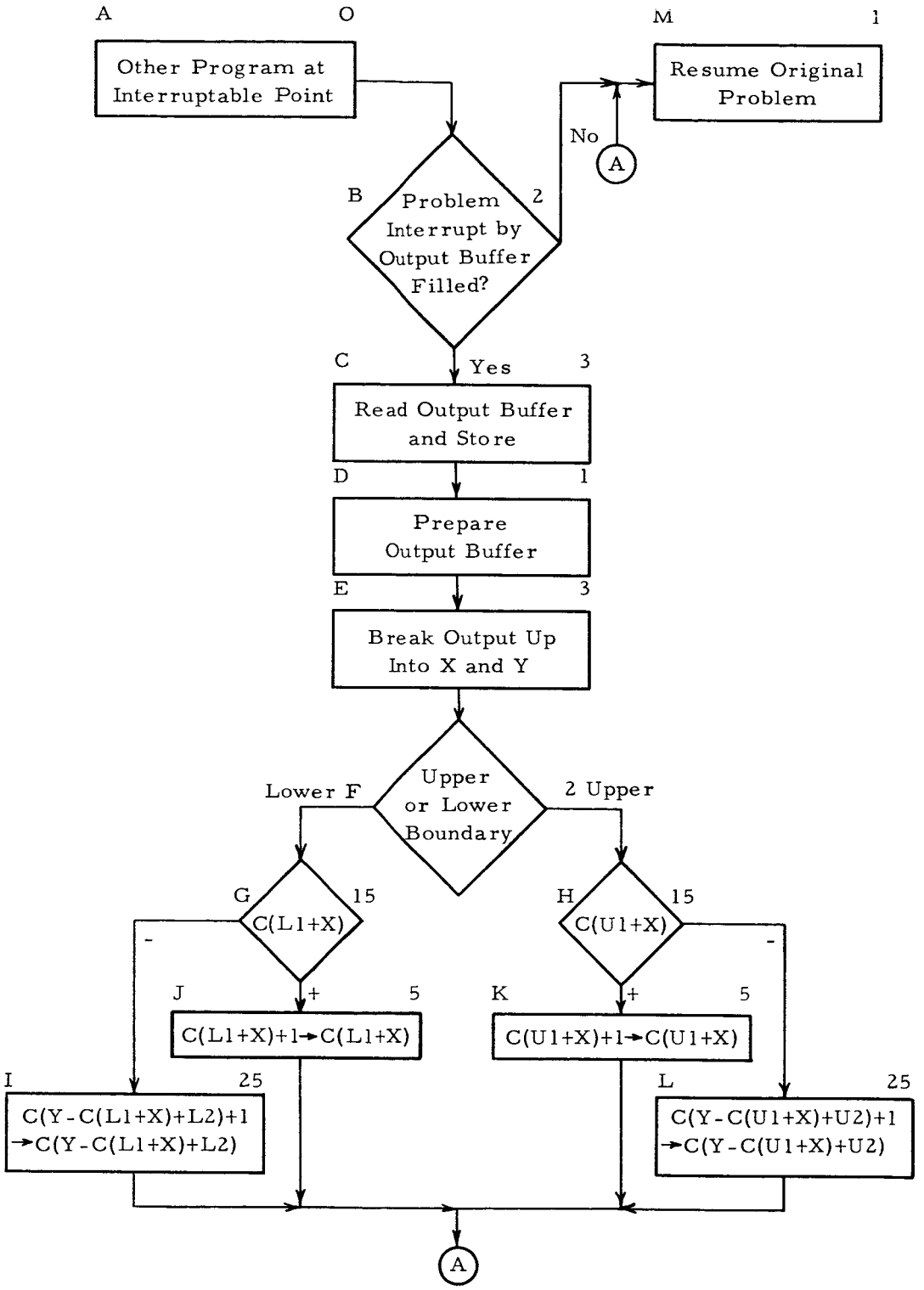


Figure 3.2. Flow Diagram of the Output Data Processing Procedure

4. ESTIMATED COSTS

To make cost estimates on the finished machine, the cost of the logic units was computed from the number of logic units used and the prices for units purchased in the 100's of units. The breakdown is as follows:

All but the flip-flop memories	$ 2,720
2 flip-flop memories, each having 14 parallel, 32 bit circular shift registers, each $7,190	14,380
Subtotal	$17,100
11% for safety factor	1,900
Total	$19,000

In addition to account for power supplies, packaging, assembly and construction with the aid of graduate assistants, add another $19,000. Estimated total cost of the RW machine is $38,000.

This figure may be appreciably reduced if fast, multi-bit shift registers become commercially available as is desirable. The savings in a multi-bit IC might justify a lower operating speed since the shift registers account for 84% of the logic circuit cost.

5. PROPOSALS

One proposal has resulted from this project. A Research Initiation Grant Proposal was submitted to the National Science Foundation. The title was "Construction of a Feasibility Model of a Random Walk Machine." The model proposed was to be built in such a way that the units built could be incorporated almost directly into the final form of the machine, thus saving duplication. This proposal was rejected.

The writer has learned that the Bureau of Reclamation* has been obtaining solutions to Laplace's equation by computer solution, though this has been relatively expensive.* This seems to be a likely prospect and will be pursued further. In addition, other governmental agencies including NASA will be considered for proposals.

---

* Conversation with Edward T. Wall, Bureau of Reclamation.

6. RESULTS

The circuit diagrams have been completed for the random walk machine utilizing the Fairchild CTμL-952 through 957 line of logic. If later developments in integrated circuitry make another line more suitable, the transformation can be made relatively easily since 1: the majority of the machine is the flip-flop circular shift registers in which almost every bit is identical. And 2: the transformation between AND's followed by OR's is made readily into NAND's followed by NAND's or NOR's followed by NOR's.

The machine cost estimate has been made on the basis of the logic units required. This has been given before as $38,000. This figure may drop as suitable multibit integrated circuits become available.

The control procedure has been outlined and is estimated at taking an average of 66 and a maximum of 82μs to process the output of the RW machine at the end of each random step. This fits very comfortably within the average time per random walk of 378μs. The GE/PAC 4020 should be able to spend 82% of its time shared with other programs.

Parallel work in the way of graduate student research is being started in the development of random bit generators and a random bit checker. This will be based on the work of Golomb,[3] Hampton[4] and Murry.[7] These units will be necessary for the random walk machine which must have a generator for its source of random bits and a checker to assure the quality of the random bits.

7. BIBLIOGRAPHY

(1) G. W. Brown, Monte-Carlo Methods, Chapter 12 of Modern Mathematics for the Engineer, Edwin F. Beckenbach, Editor, McGraw-Hill, New York, 1956

(2) George E. Forsythe and Wolfgang R. Wasow, Finite Difference Methods for Partial Differential Equations, John Wiley and Sons, New York, 1957

(3) Solomon W. Golomb, Sequences with Randomness Properties, Glenn L. Martin Co., Baltimore, Maryland, June 14, 1955 (internal report)

(4) R. L. T. Hampton, A Hybrid Analog-Digital Pseudo-Random Noise Generator; Proceedings - Spring Joint Computer Conference, 1964, p 287-301

(5) Heihachiro Hirai and Tatsuzo Mikami, Design of "Random Walker" for Monte-Carlo Method, Part II (Electronic Device), Journal of the Institute of Polytechnics, Osaka City University, Vol. 11, No. 1, Series A, 1960

(6) William D. Lansdown, The Random Walk Solution of Laplace's Equation: Method and Machine, Ph. D. Thesis, University of Kansas, Lawrence, Kansas, 1964

(7) Herschell F. Murry, Generation of Random Numbers from Natural Fluctuation Phenomena, Ph. D. Thesis, University of Kansas, Lawrence, Kansas, 1966

(8) Hiroshi Sugiyama and Osamu Miyatake, Design of "Random Walker" for Monte-Carlo Method, Part I (Theory), Journal of the Institute of Polytechnics, Osaka City University, Vol. 10, No. 1, Series A, 1959

(9) John Todd, Experiments in the Solution of Differential Equations by Monte-Carlo Methods, Journal of the Washington Academy of Sciences, Vol. 44, No. 12, 1954, p 377-381

(10) E. C. Yowell, A Monte-Carlo Method for Solving Laplace's Equation, Proceedings of the Seminar of Scientific Computation, December 1949, International Business Machines Corp., New York, 1951

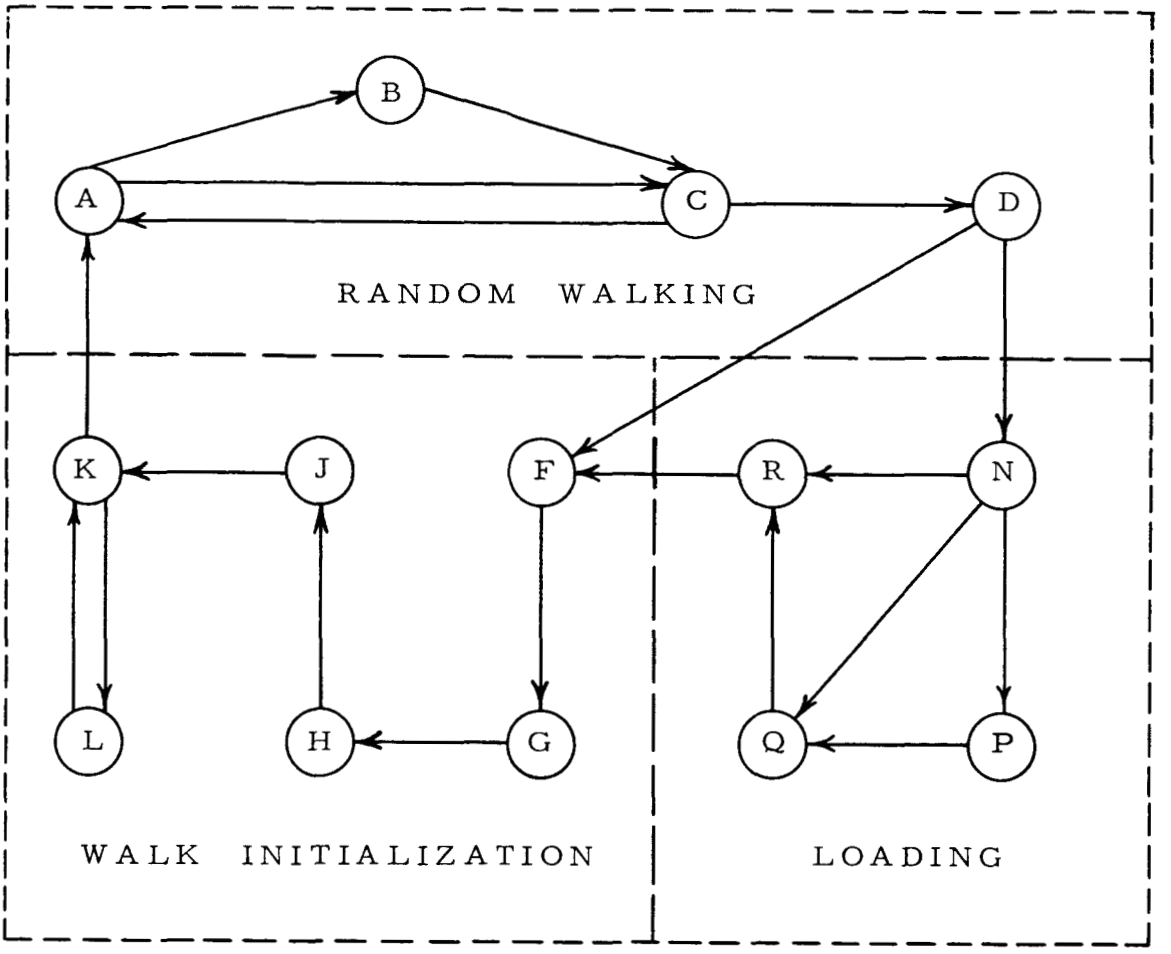
APPENDICES

APPENDIX A

CIRCUIT DIAGRAMS

The circuit diagrams are given in the same order as the logic diagrams in Appendix B. That is, Figure A. 3a is the circuit diagram corresponding to logic diagram B. 3a and both bear the same title, Clock and Phase Control. An index of the circuit diagrams follows.

INDEX OF THE CIRCUIT DIAGRAMS



Notes

1. The majority of machine operation is in the random walking section, particularly in states A, B, and C.

2. State F starts initialization for the next random walk if there are further walks to make (the usual situation).

3. State N starts the loading; external control can specify the Load Walk Counter (LWC) which is state R, Load Initial Position (LIP) which is state Q, or Load Memory (LMM) which is state P. At the conclusion of loading the process goes to walk initialization before the random walking starts.

Figure A. 1. State Diagram of Random Walk Machine Operation

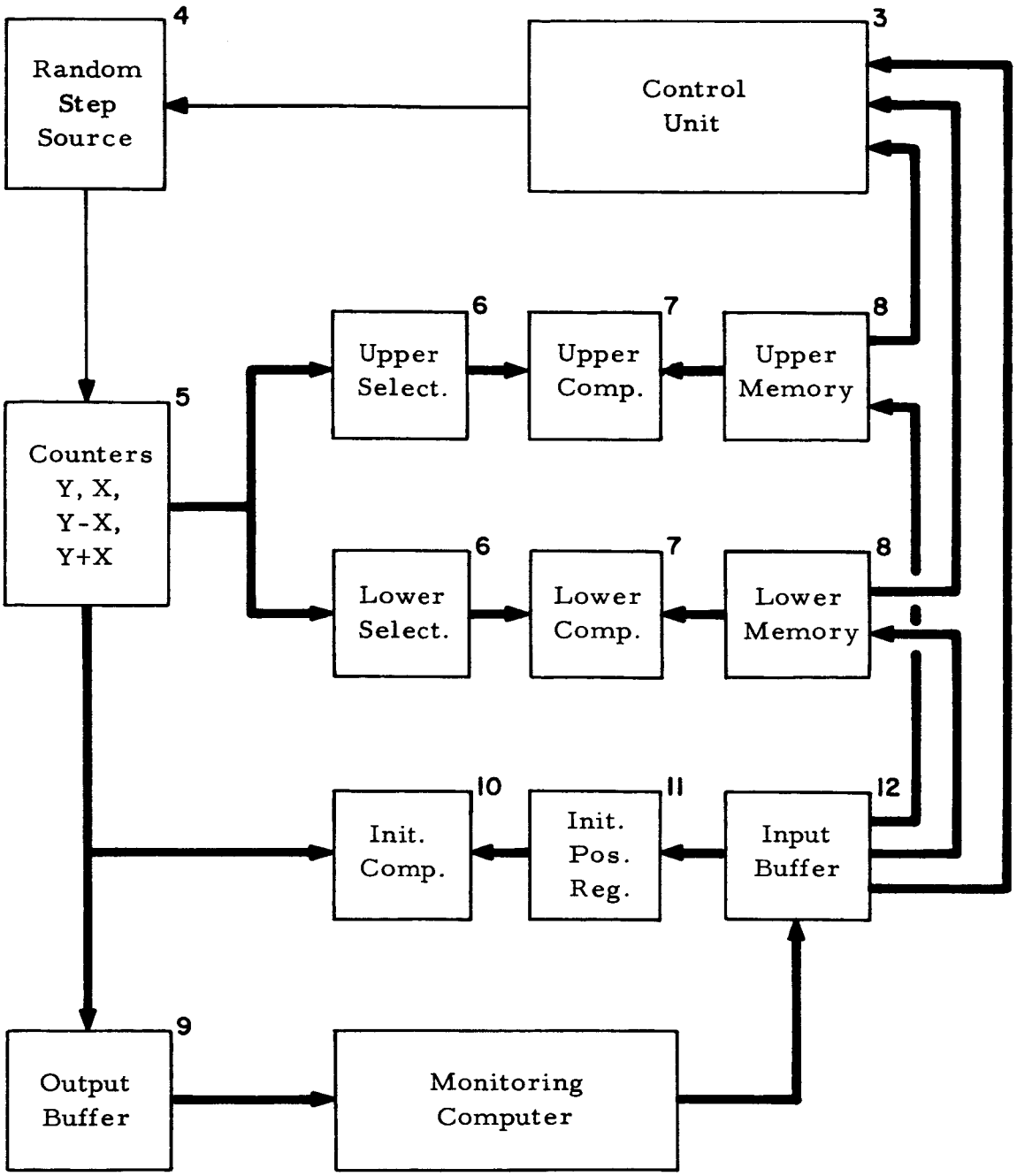


Figure A. 2. Block Diagram of Random Walk Machine with Data Paths

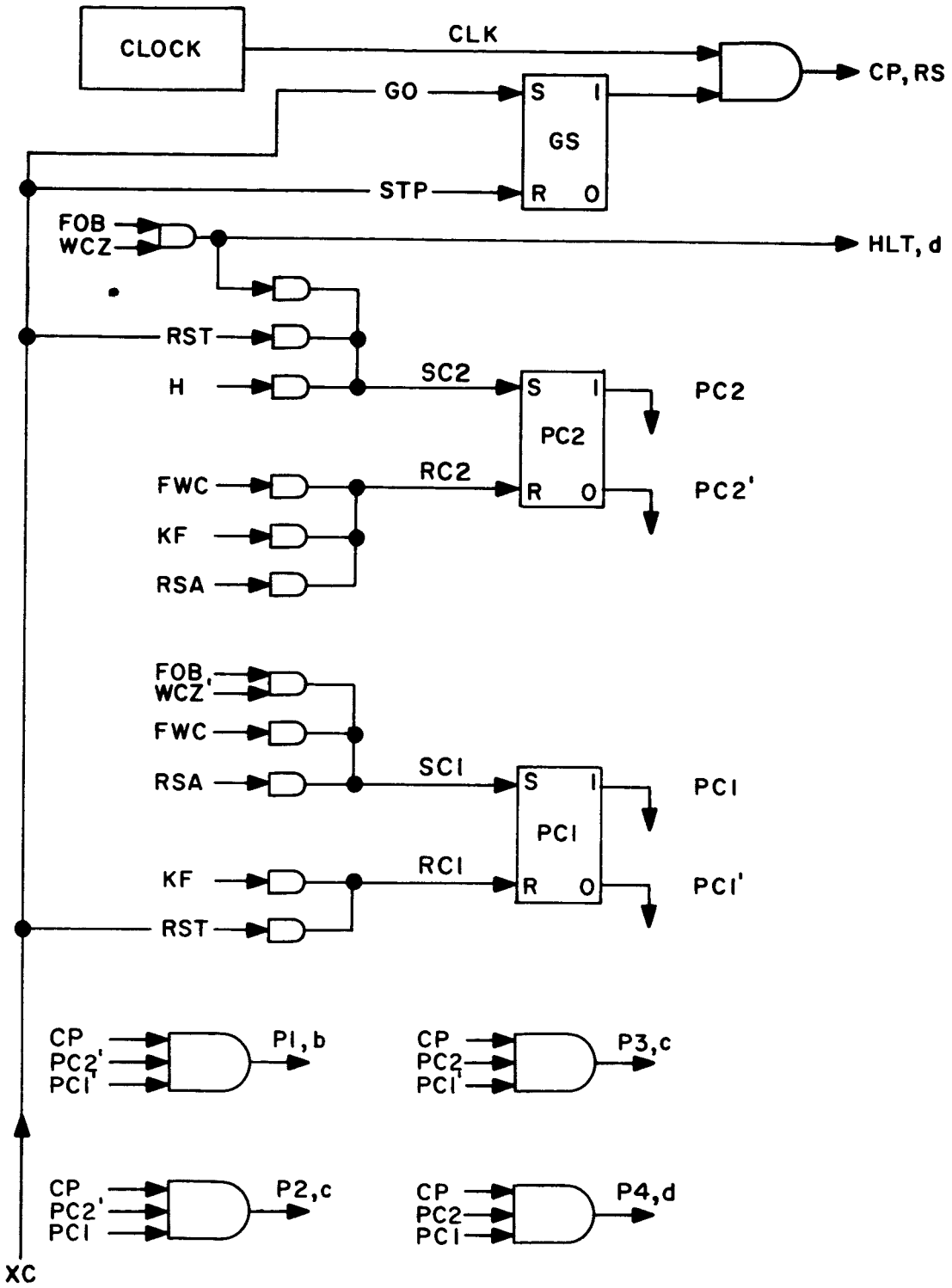
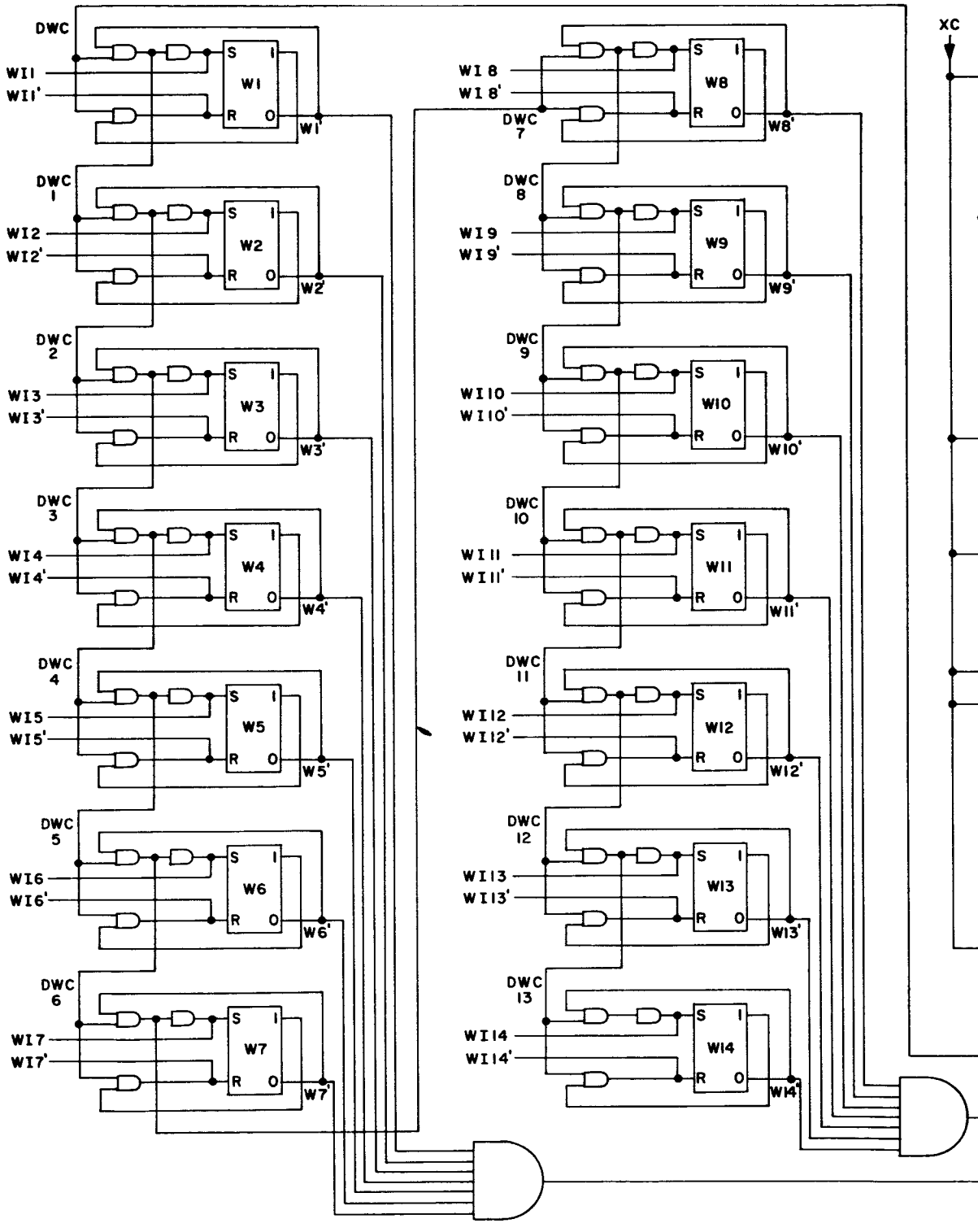
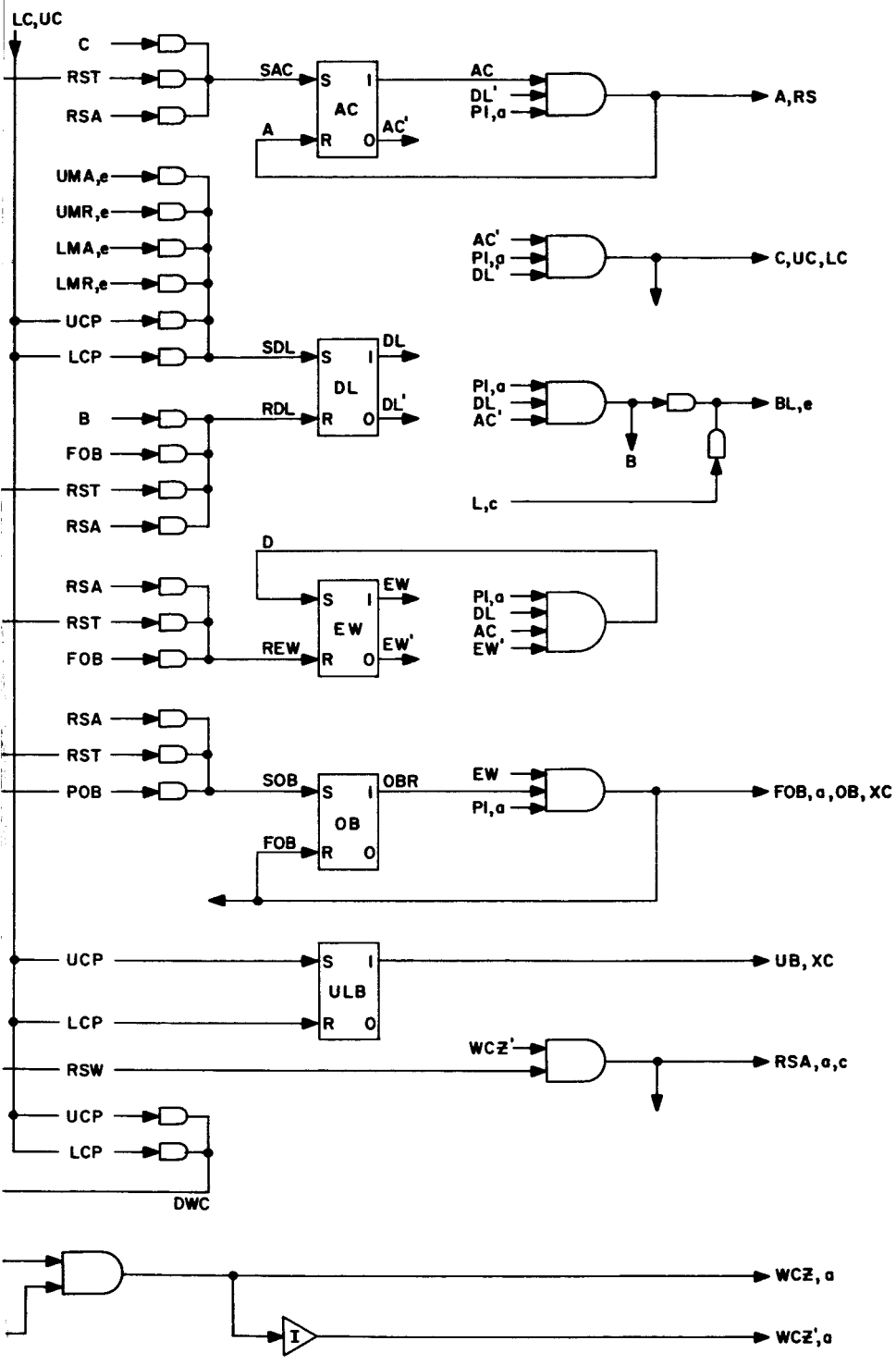


Figure A. 3a. Clock and Phase Control



NOTE: SIGNALS WI-14 AND WI'-14' FROM IB

Figure A.3b. Random St



p Control

16-2

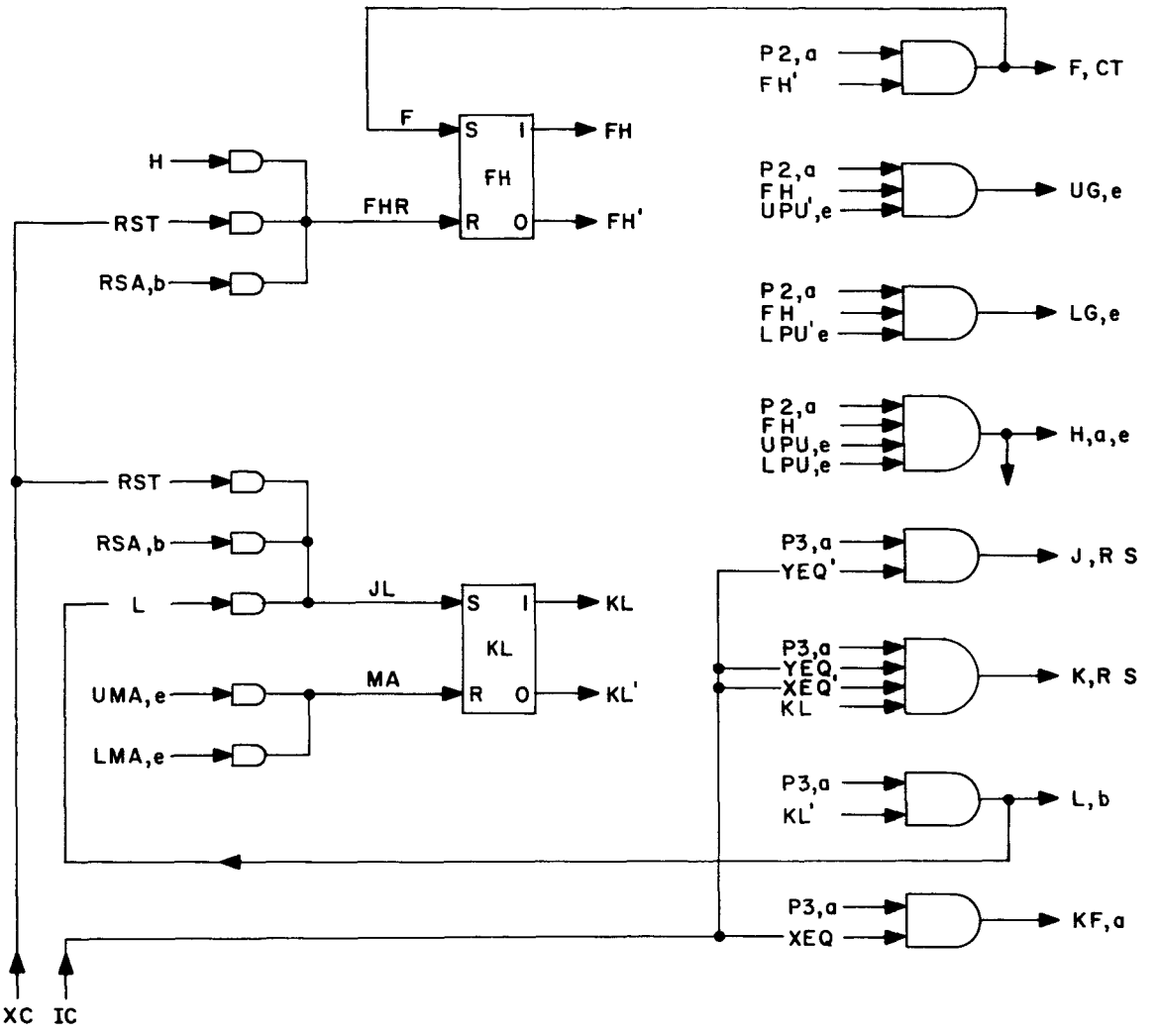


Figure A.3c. Initialization Control

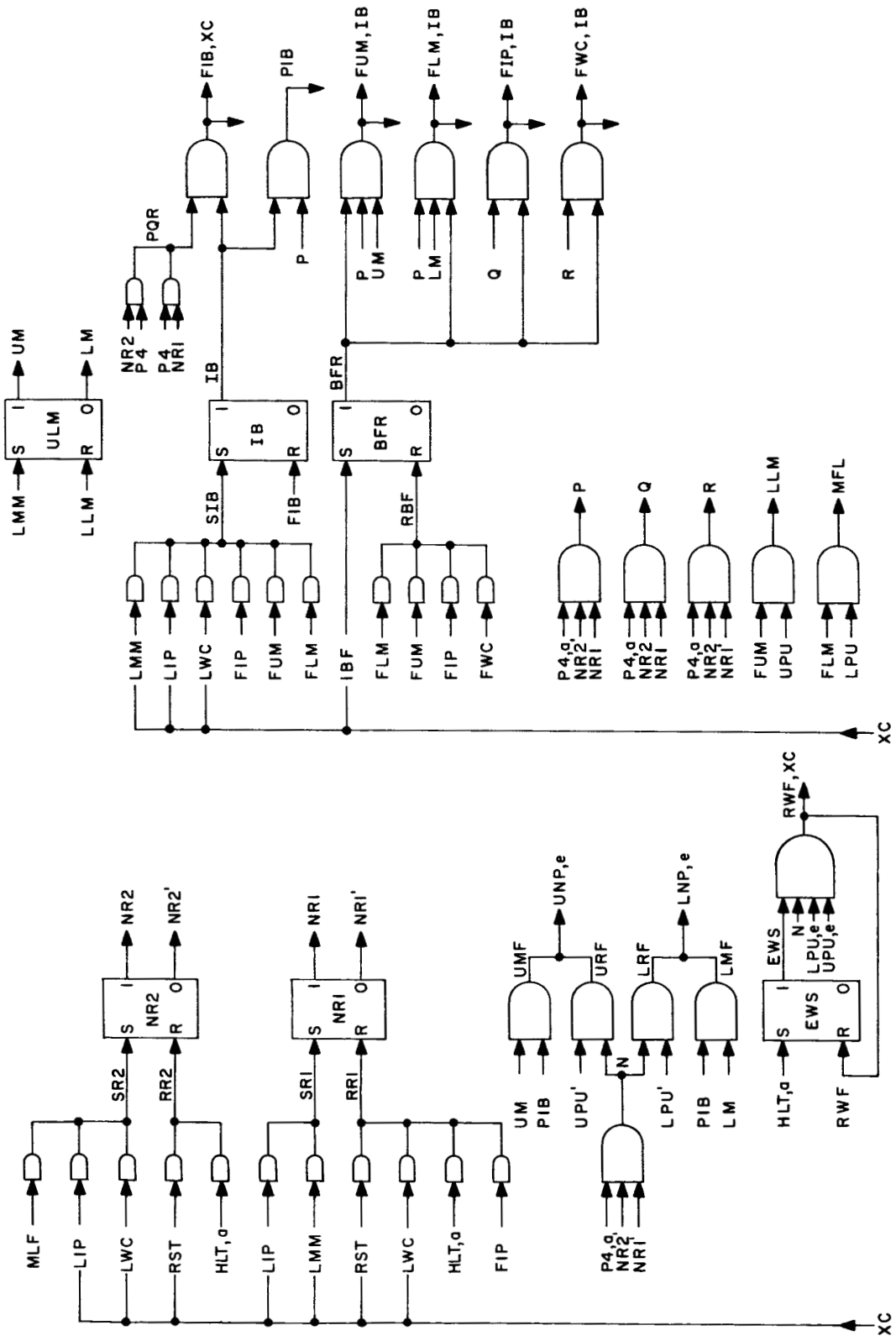
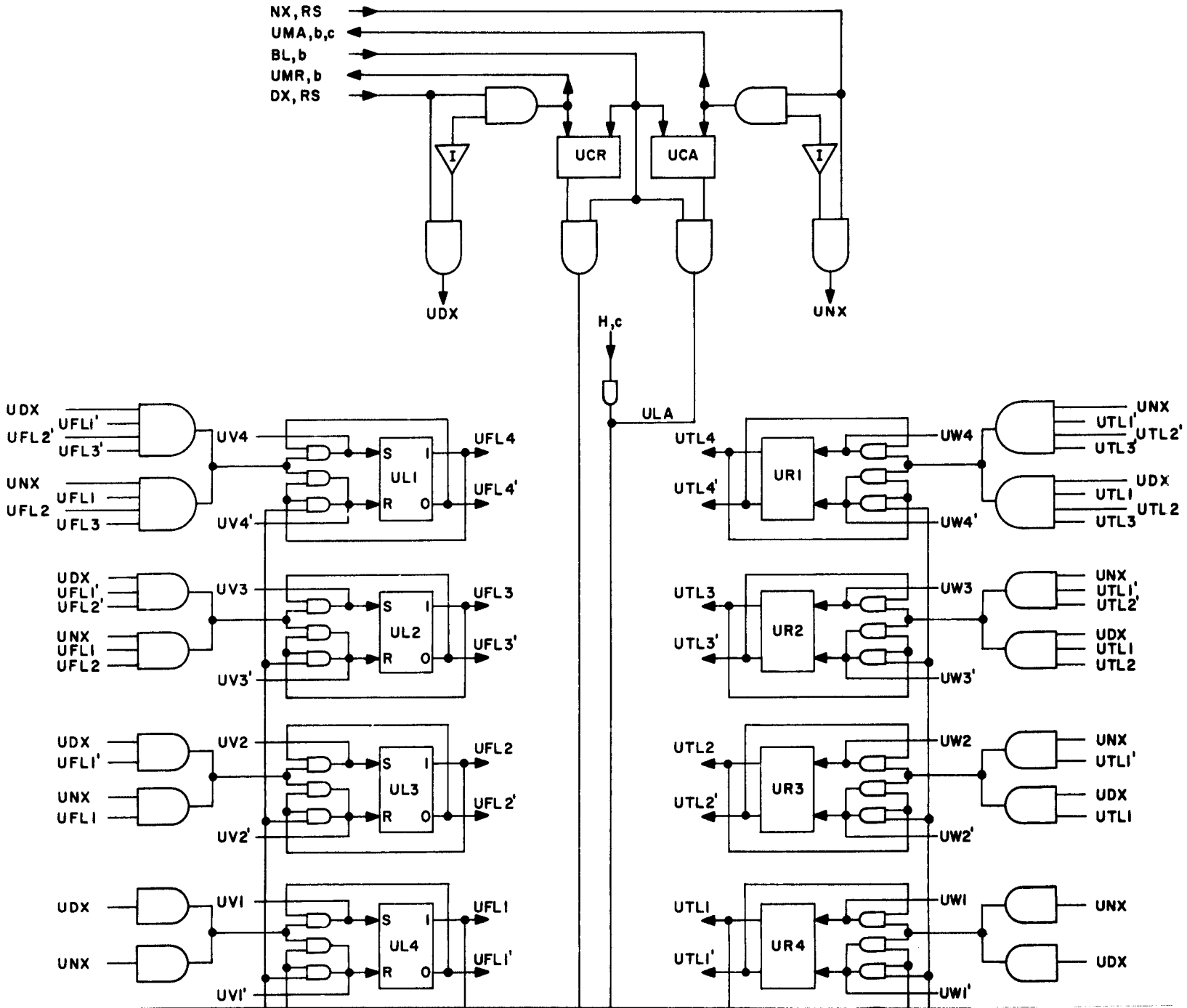


Figure A.3d. Loading Control


NX, RS
UMA, b, c
BL, b
UMR, b
DX, RS
UCR
UCA
I
I
UDX
UNX
H, c
ULA
UL1
UL2
UL3
UL4
UR1
UR2
UR3
UR4
S
R
I
O
UV4
UV4'
UV3
UV3'
UV2
UV2'
UV1
UV1'
UFL4
UFL4'
UFL3
UFL3'
UFL2
UFL2'
UFL1
UFL1'
UW4
UW4'
UW3
UW3'
UW2
UW2'
UW1
UW1'
UTL4
UTL4'
UTL3
UTL3'
UTL2
UTL2'
UTL1
UTL1'
UFL2'
UFL3'
UFL2
UFL3
UNX
UDX

1-61

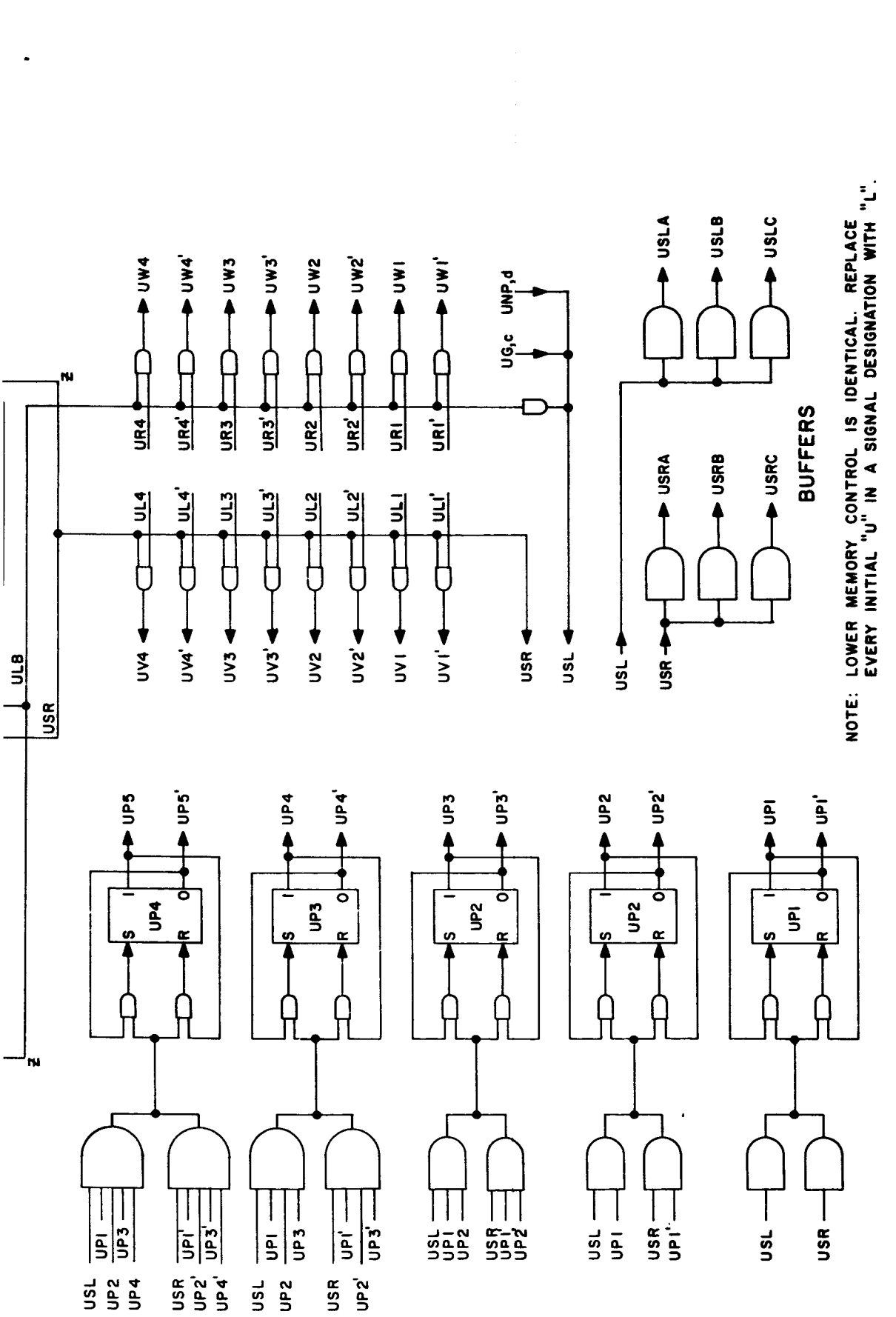


Figure A.3e. Upper Memory Control

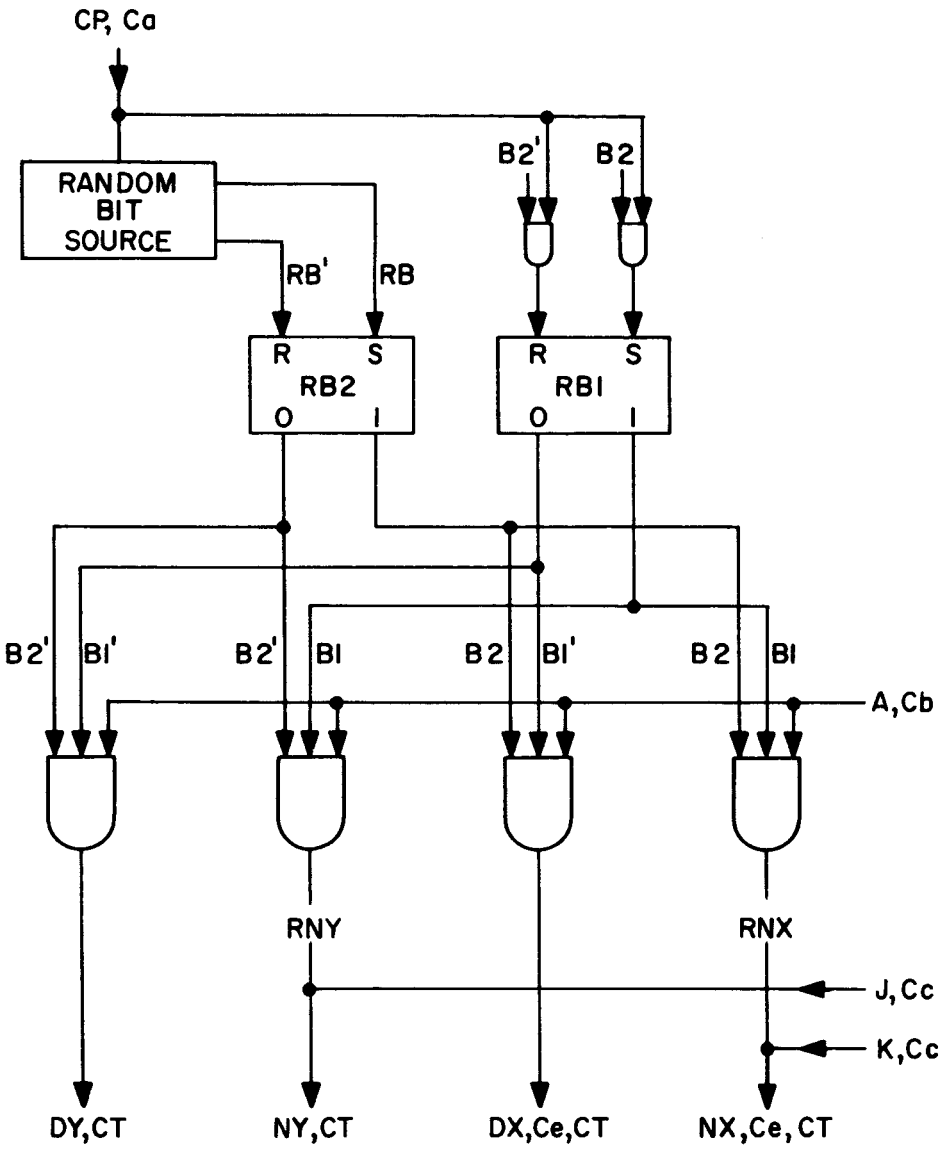
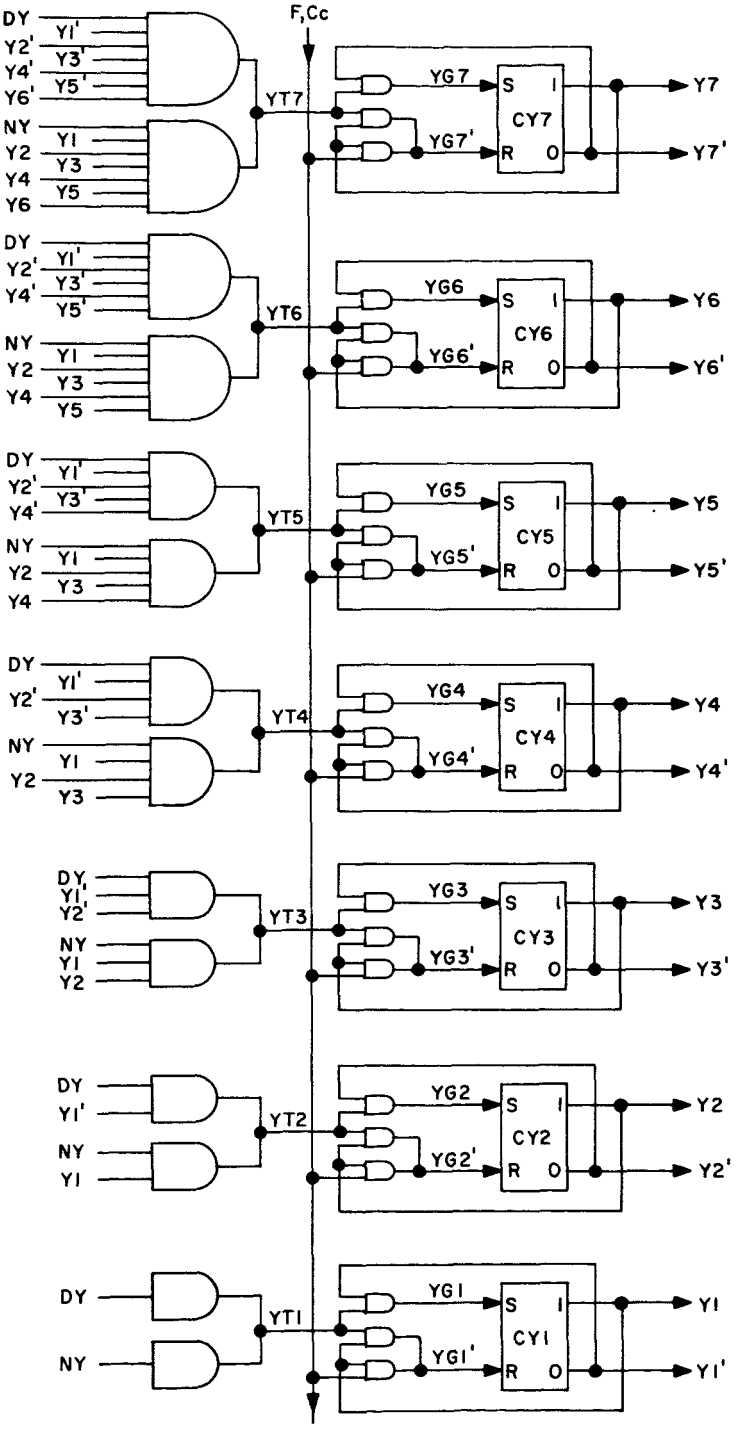


Figure A.4. Random Step Source



NOTE: ALL OUTPUT SIGNALS OF Y, X, YMX AND YPX COUNTERS GO TO UC AND LC

Figure A. 5a. Y Counter

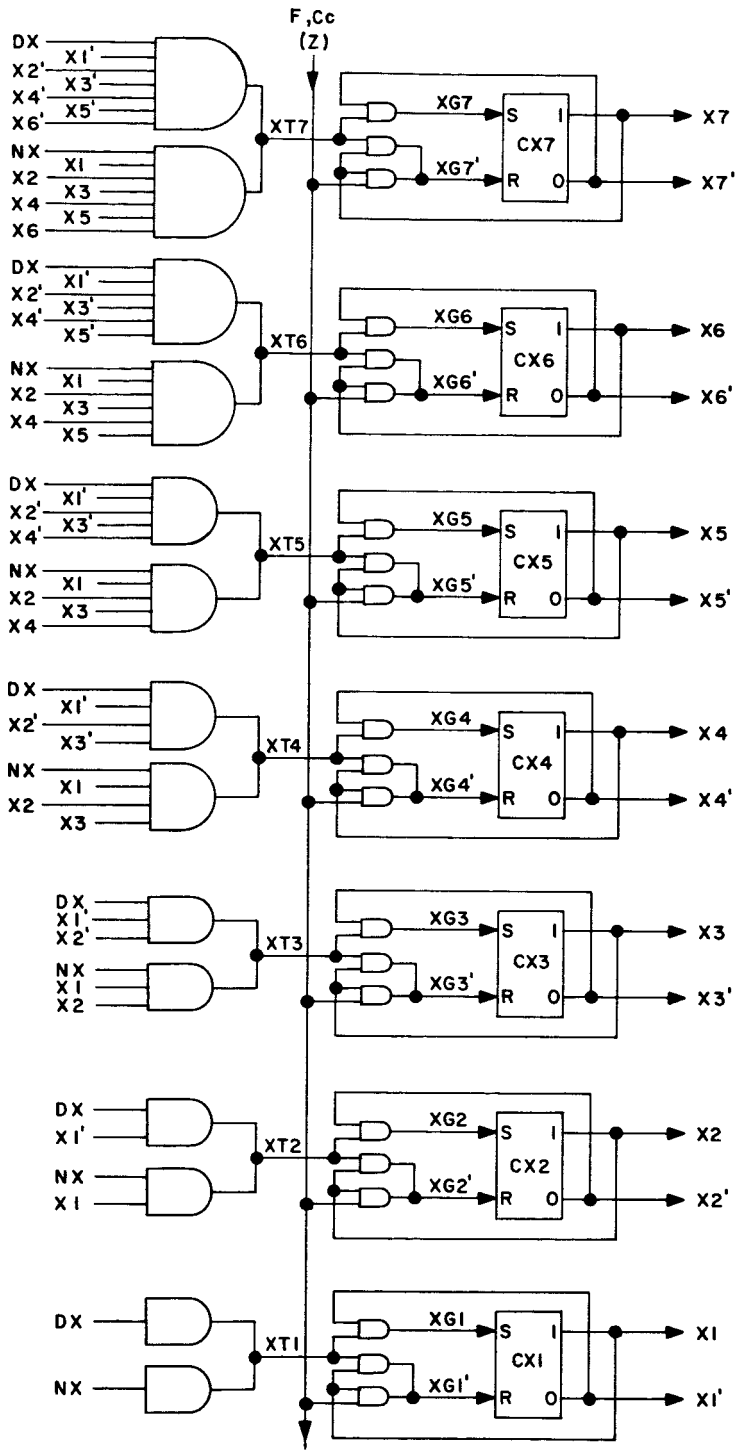


Figure A. 5b. X Counter

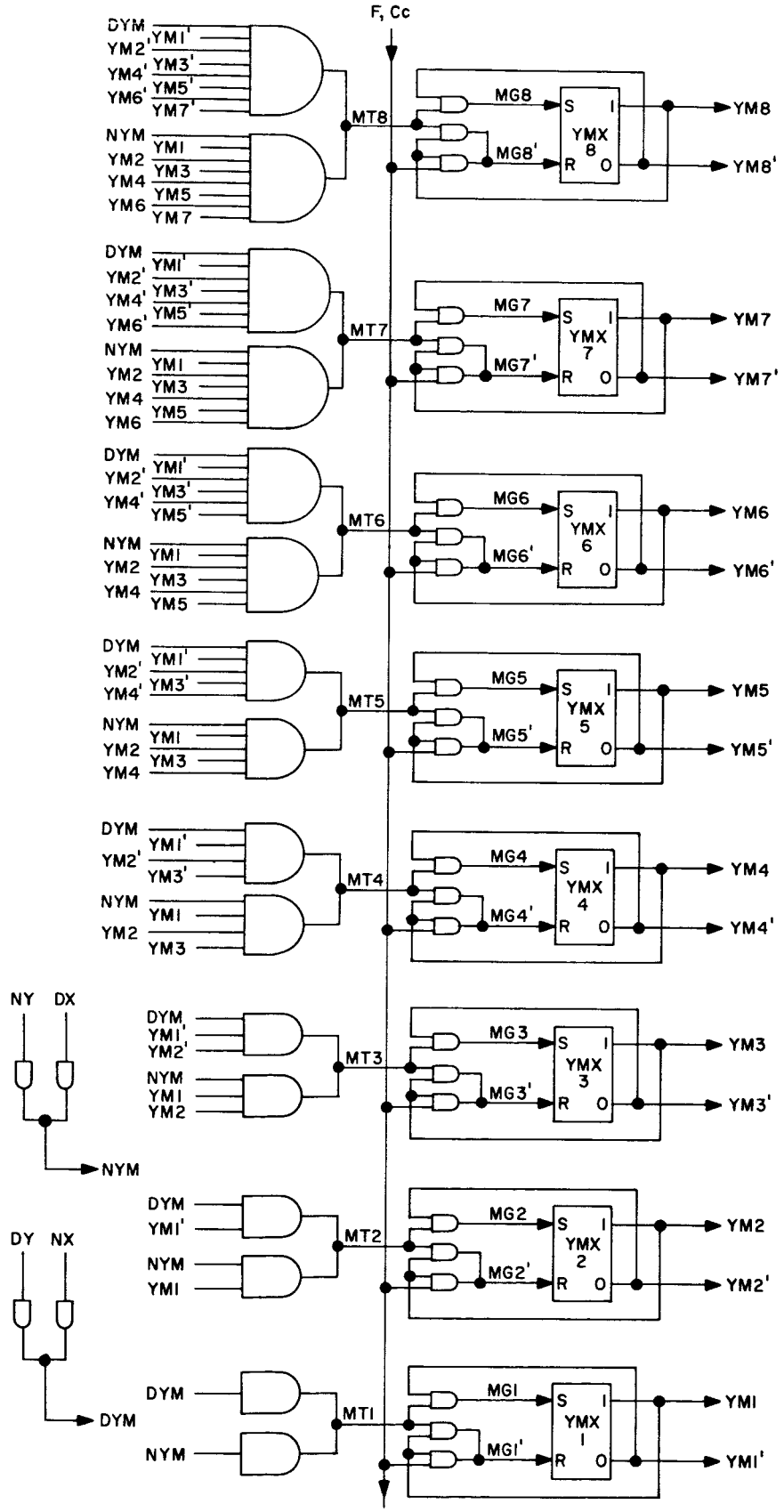


Figure A.5c. YMX Counter

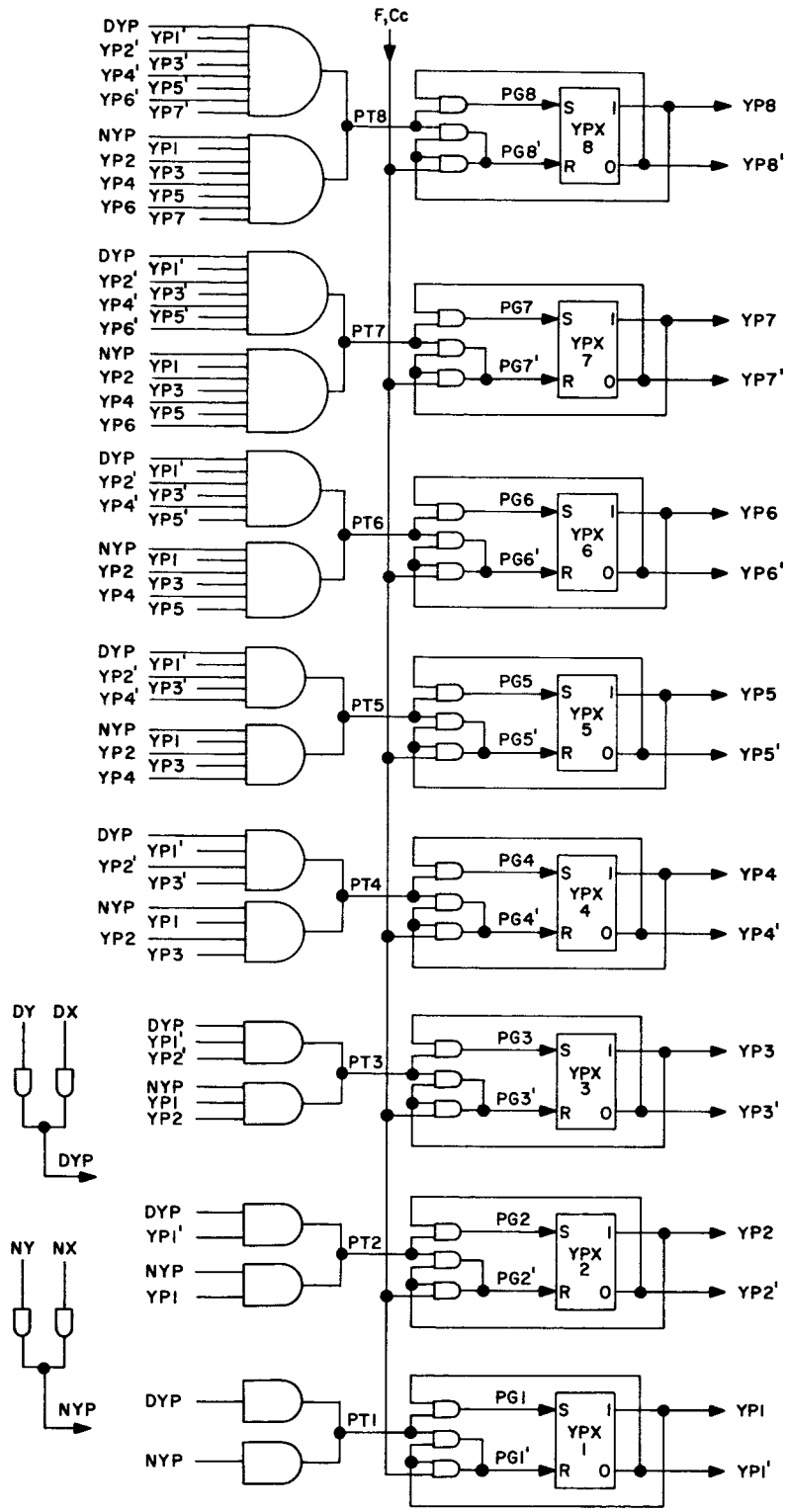
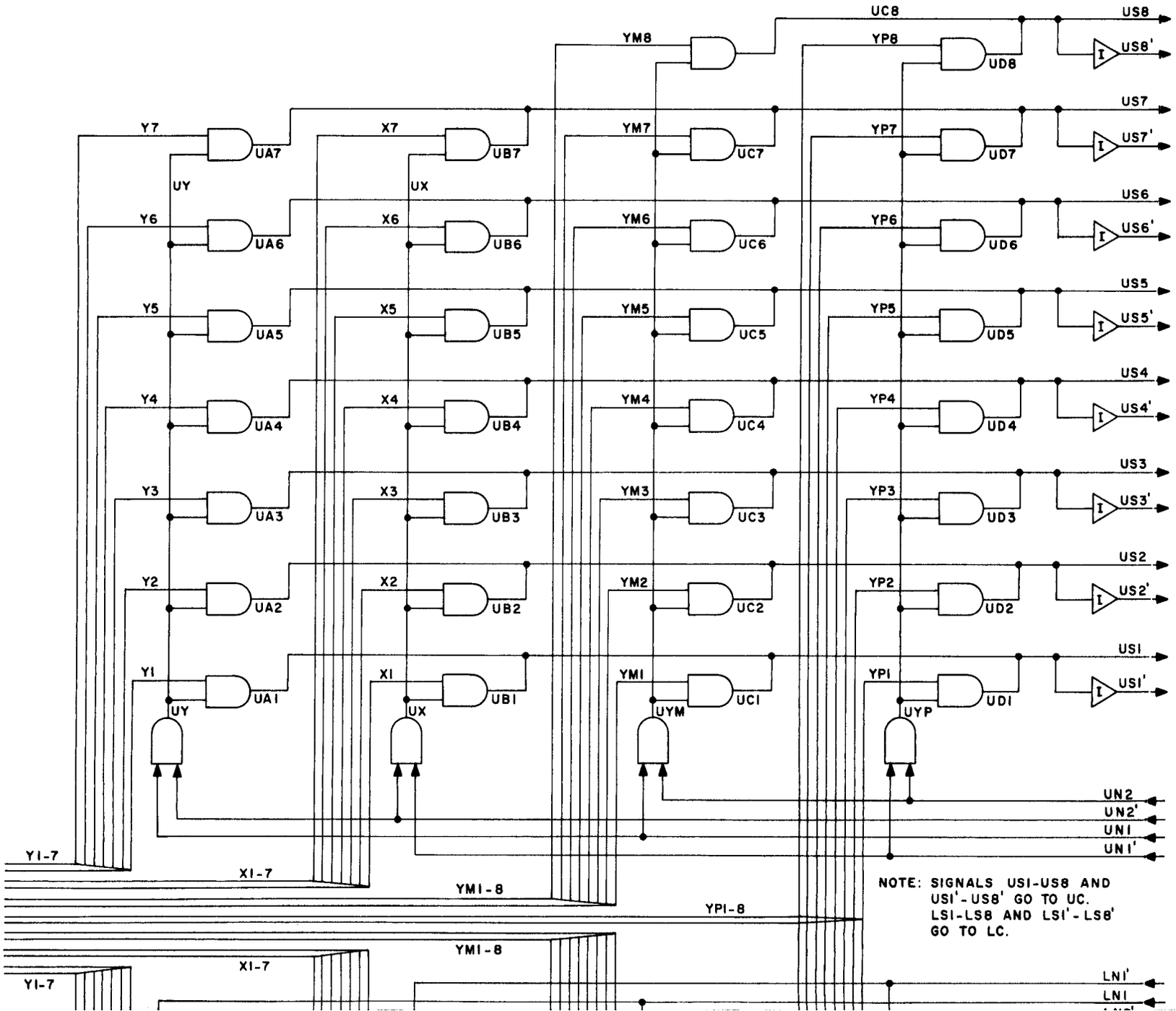


Figure A. 5d. YPX Counter


Y7
Y6
Y5
Y4
Y3
Y2
Y1
X7
X6
X5
X4
X3
X2
X1
YM8
YM7
YM6
YM5
YM4
YM3
YM2
YM1
YP8
YP7
YP6
YP5
YP4
YP3
YP2
YP1
UA7
UA6
UA5
UA4
UA3
UA2
UA1
UB7
UB6
UB5
UB4
UB3
UB2
UB1
UC8
UC7
UC6
UC5
UC4
UC3
UC2
UC1
UD8
UD7
UD6
UD5
UD4
UD3
UD2
UD1
UY
UX
UYM
UYP
US8
US8'
US7
US7'
US6
US6'
US5
US5'
US4
US4'
US3
US3'
US2
US2'
USI
USI'
UN2
UN2'
UN1
UN1'
LN1'
LN1
Y1-7
X1-7
YM1-8
YP1-8
NOTE: SIGNALS USI-US8 AND USI'-US8' GO TO UC. LSI-LS8 AND LSI'-LS8' GO TO LC.

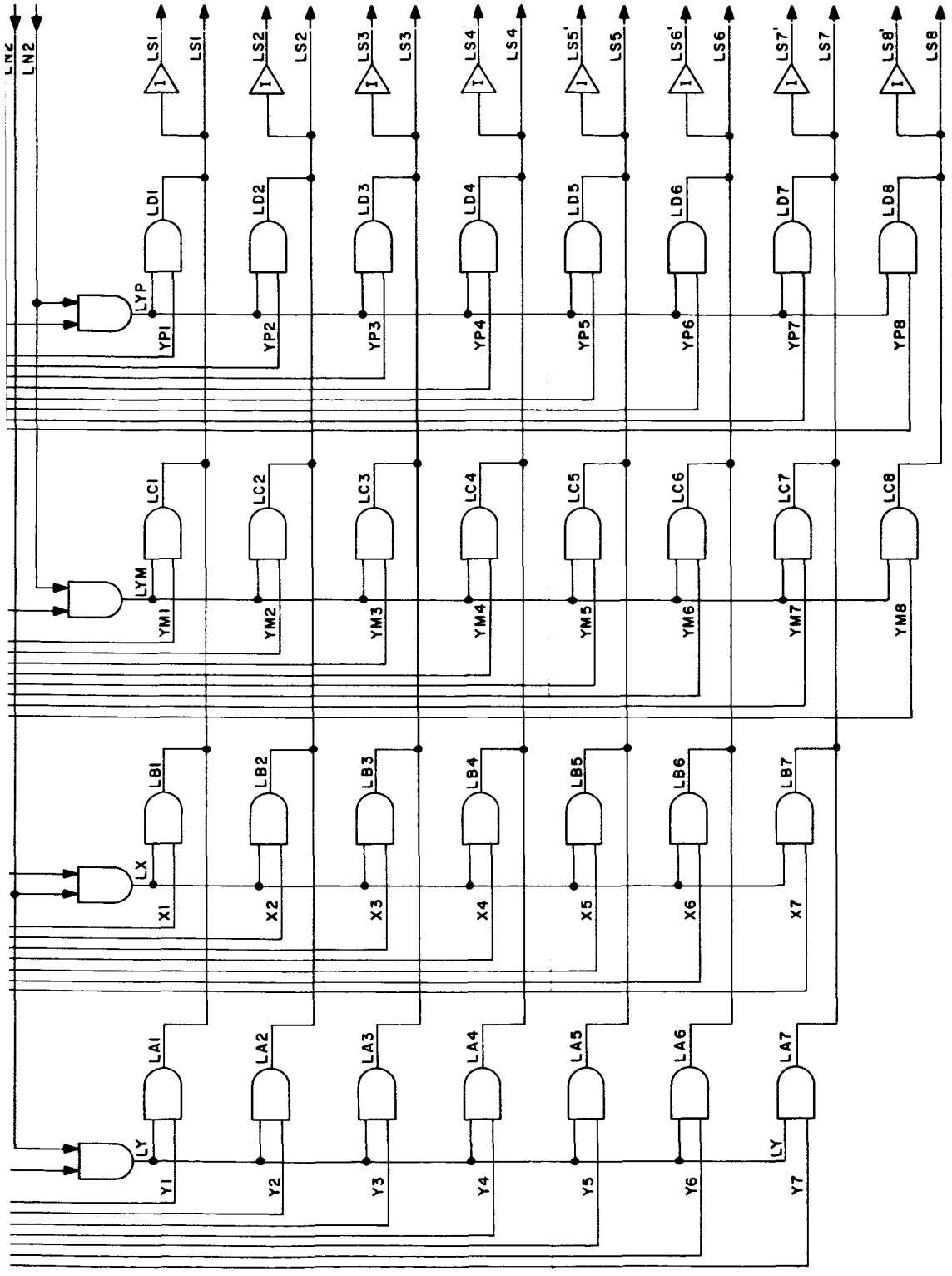


Figure A.6. Upper and Lower Selector

25-2

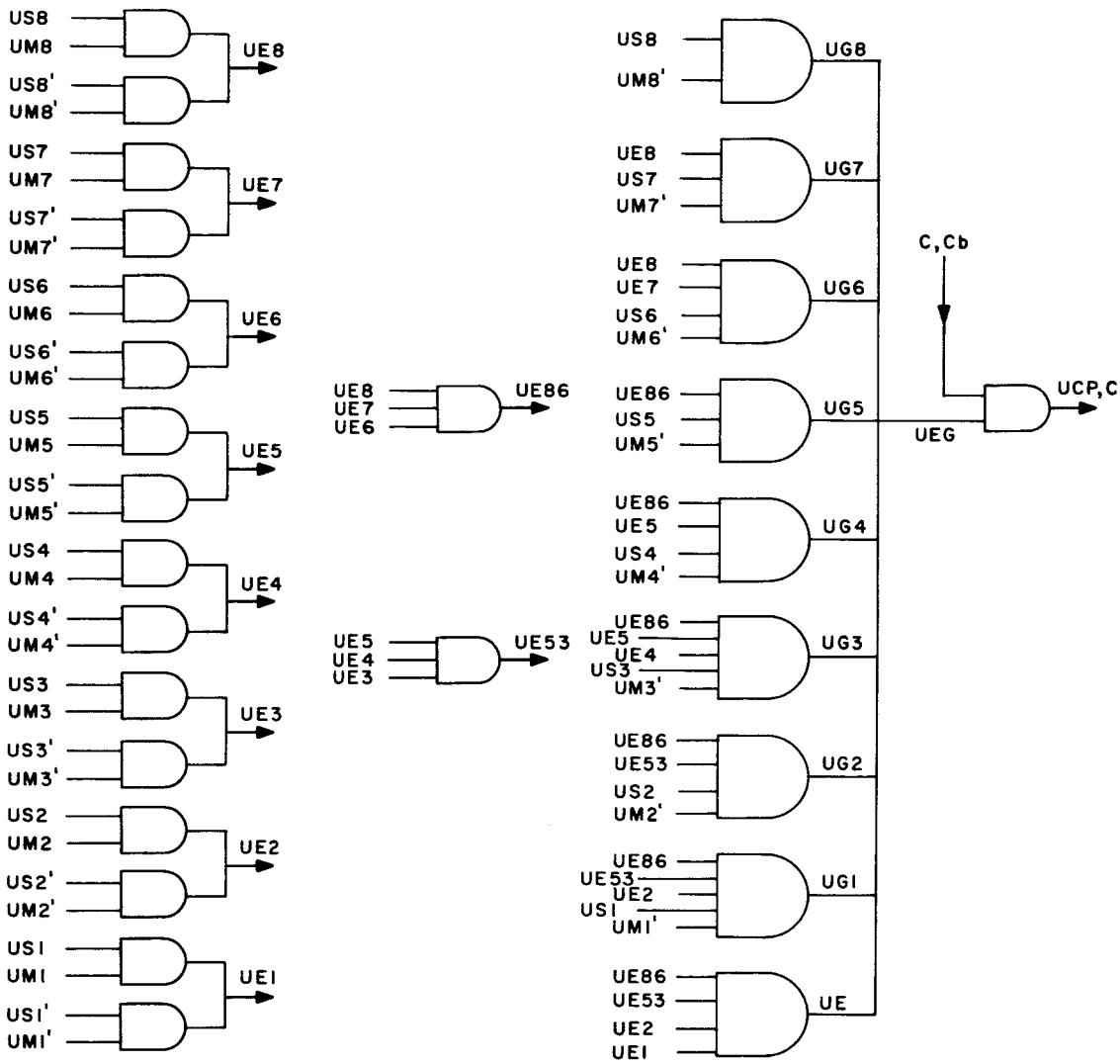


Figure A.7a. Upper Comparator

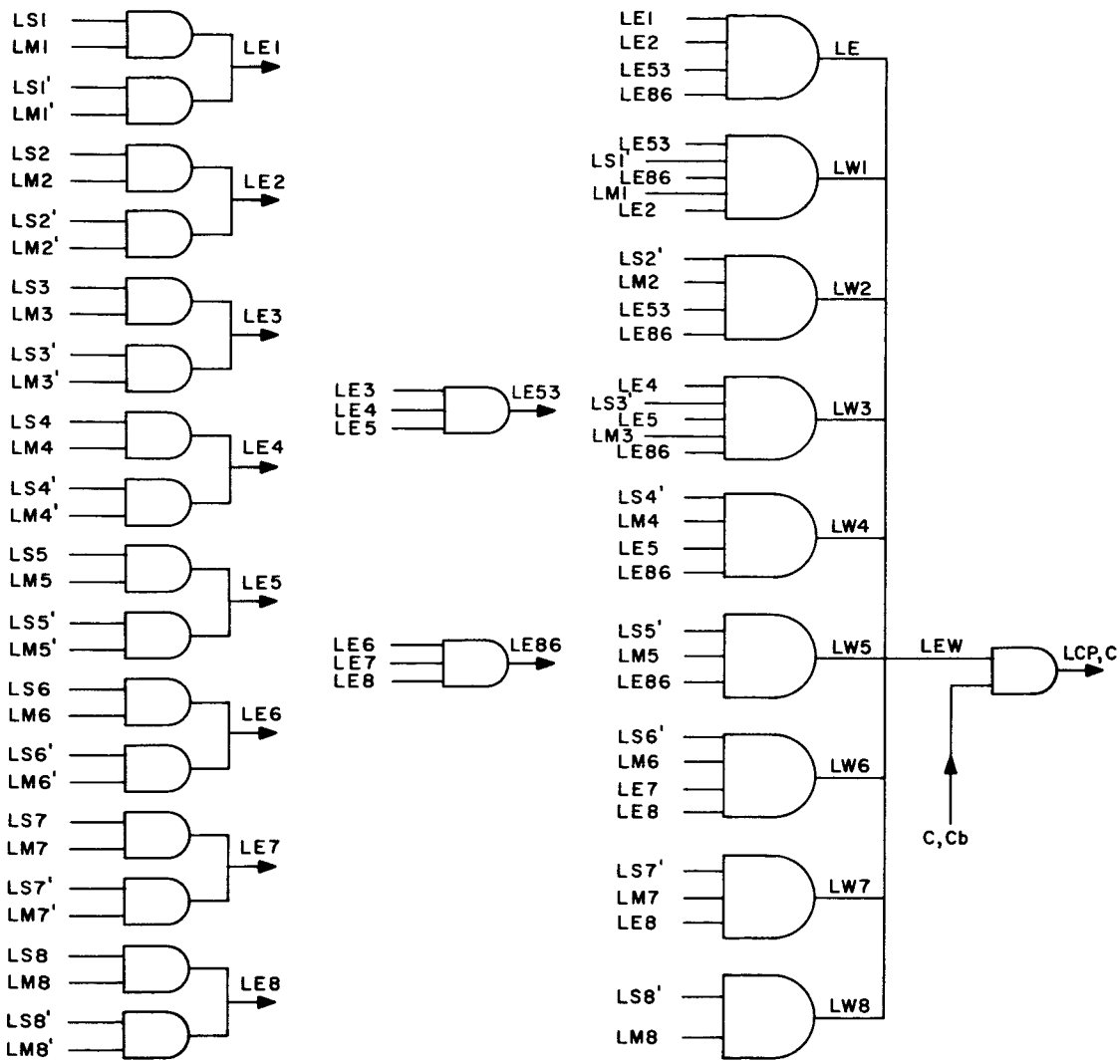
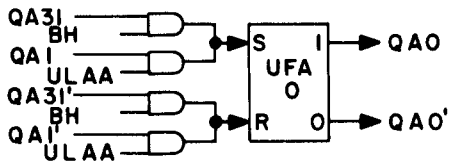
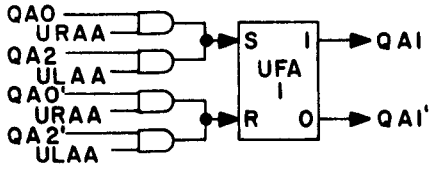
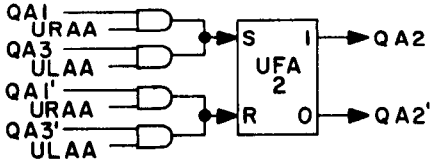
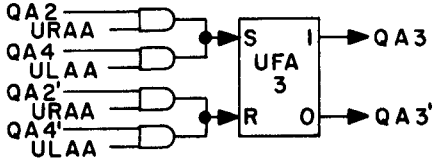


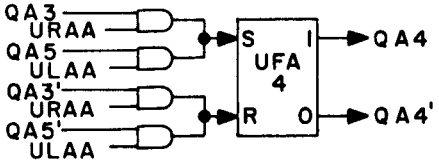
Figure A. 7b. Lower Comparator

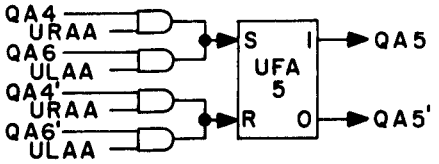

QA31 BH
QA1 ULAA
QA31' BH
QA1' ULAA
S I
UFA 0
R O
QA0
QA0'

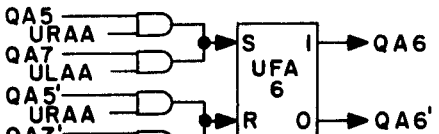

QA0 URAA
QA2 ULAA
QA0' URAA
QA2' ULAA
S I
UFA 1
R O
QA1
QA1'

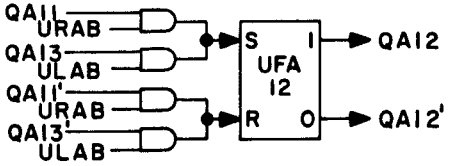

QA1 URAA
QA3 ULAA
QA1' URAA
QA3' ULAA
S I
UFA 2
R O
QA2
QA2'

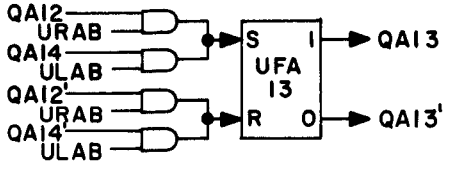

QA2 URAA
QA4 ULAA
QA2' URAA
QA4' ULAA
S I
UFA 3
R O
QA3
QA3'

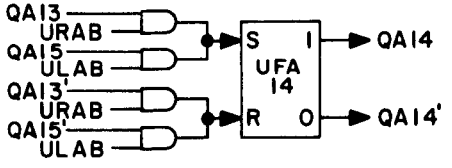

QA3 URAA
QA5 ULAA
QA3' URAA
QA5' ULAA
S I
UFA 4
R O
QA4
QA4'

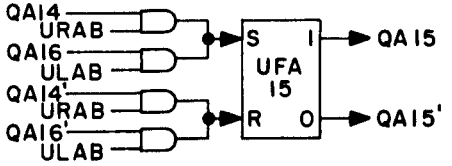

QA4 URAA
QA6 ULAA
QA4' URAA
QA6' ULAA
S I
UFA 5
R O
QA5
QA5'

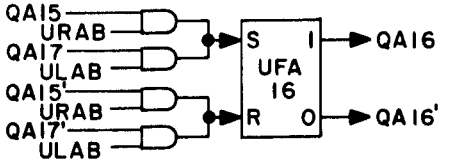

QA5 URAA
QA7 ULAA
QA5' URAA
S I
UFA 6
R O
QA6
QA6'

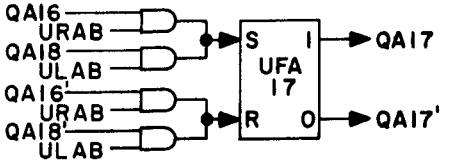

QA11 URAB
QA13 ULAB
QA11' URAB
QA13' ULAB
S I
UFA 12
R O
QA12
QA12'

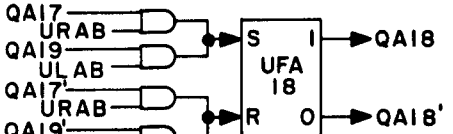

QA12 URAB
QA14 ULAB
QA12' URAB
QA14' ULAB
S I
UFA 13
R O
QA13
QA13'

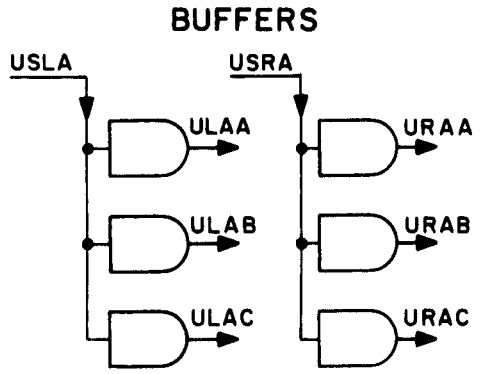

QA13 URAB
QA15 ULAB
QA13' URAB
QA15' ULAB
S I
UFA 14
R O
QA14
QA14'

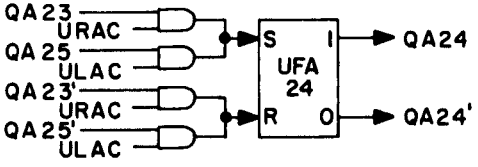

QA14 URAB
QA16 ULAB
QA14' URAB
QA16' ULAB
S I
UFA 15
R O
QA15
QA15'

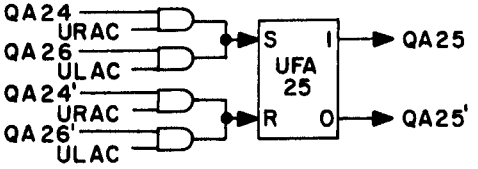

QA15 URAB
QA17 ULAB
QA15' URAB
QA17' ULAB
S I
UFA 16
R O
QA16
QA16'

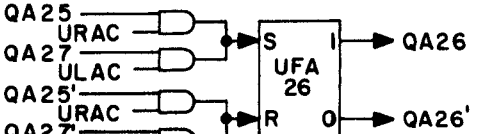

QA16 URAB
QA18 ULAB
QA16' URAB
QA18' ULAB
S I
UFA 17
R O
QA17
QA17'


QA17 URAB
QA19 ULAB
QA17' URAB
S I
UFA 18
R O
QA18
QA18'


BUFFERS
USLA
USRA
ULAA
ULAB
ULAC
URAA
URAB
URAC


QA23 URAC
QA25 ULAC
QA23' URAC
QA25' ULAC
S I
UFA 24
R O
QA24
QA24'


QA24 URAC
QA26 ULAC
QA24' URAC
QA26' ULAC
S I
UFA 25
R O
QA25
QA25'


QA25 URAC
QA27 ULAC
QA25' URAC
S I
UFA 26
R O
QA26
QA26'

18-1

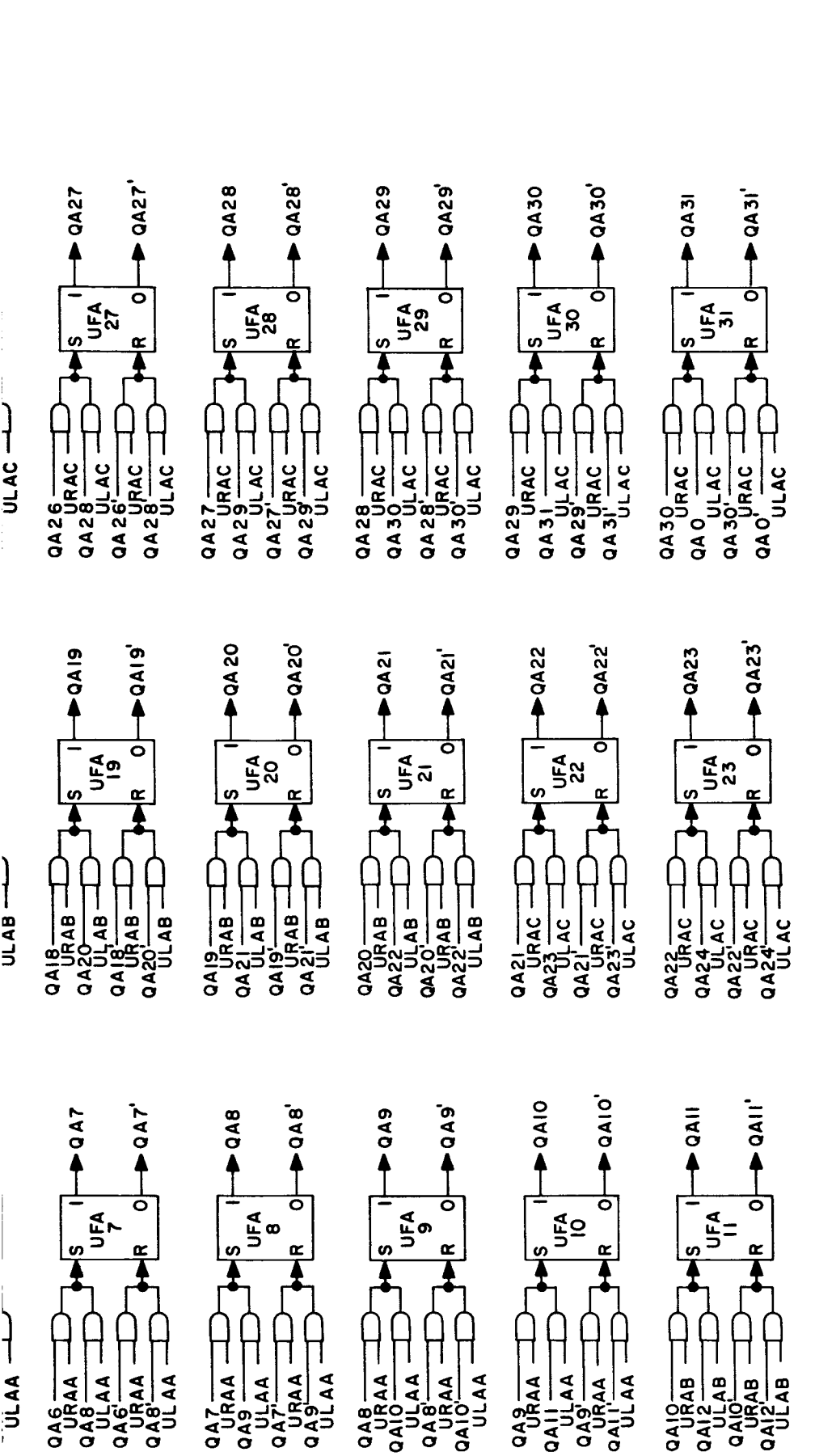


Figure A.8. Upper Memory (First Ring)

28-2

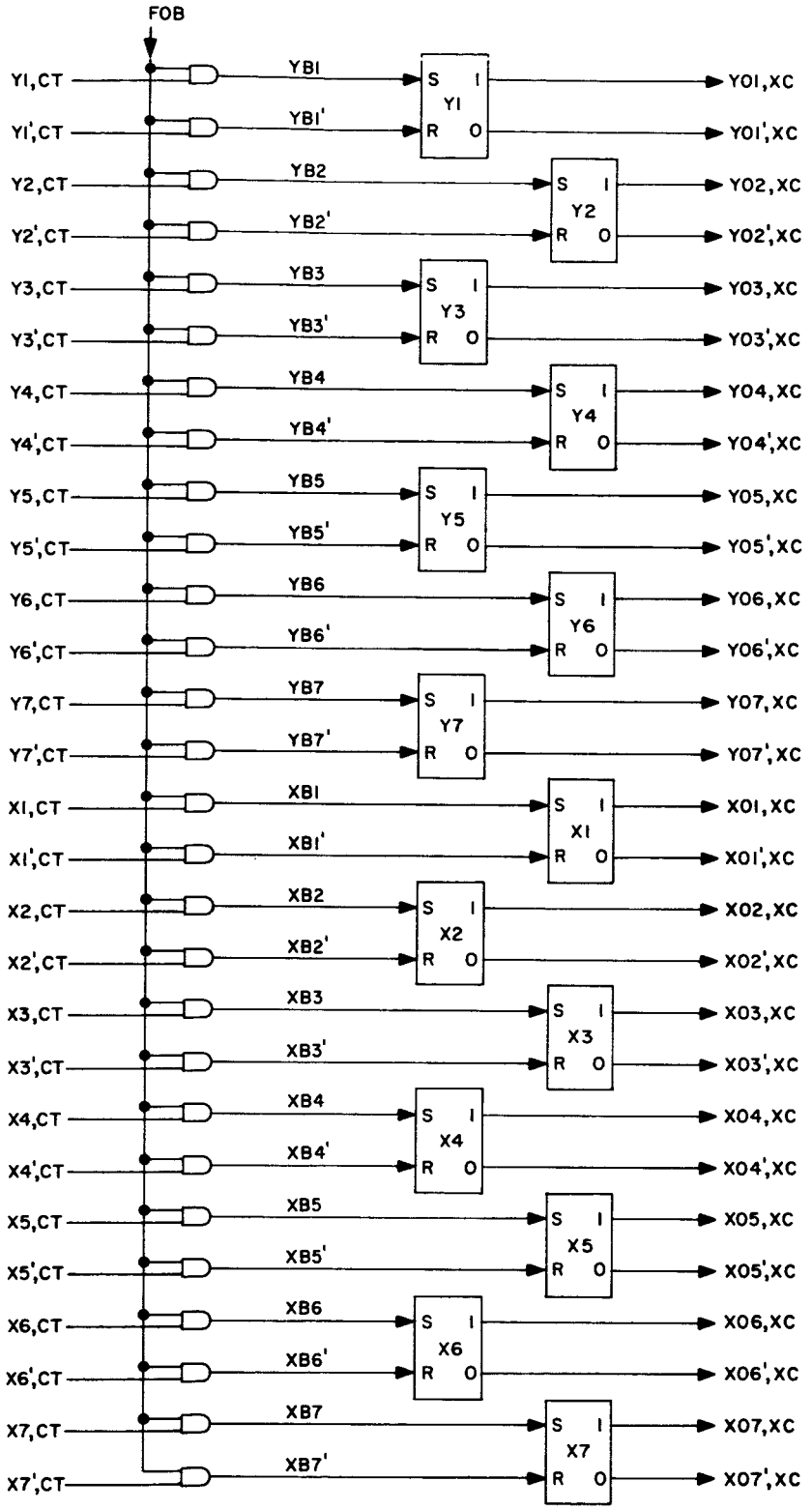


Figure A. 9. Output Buffer

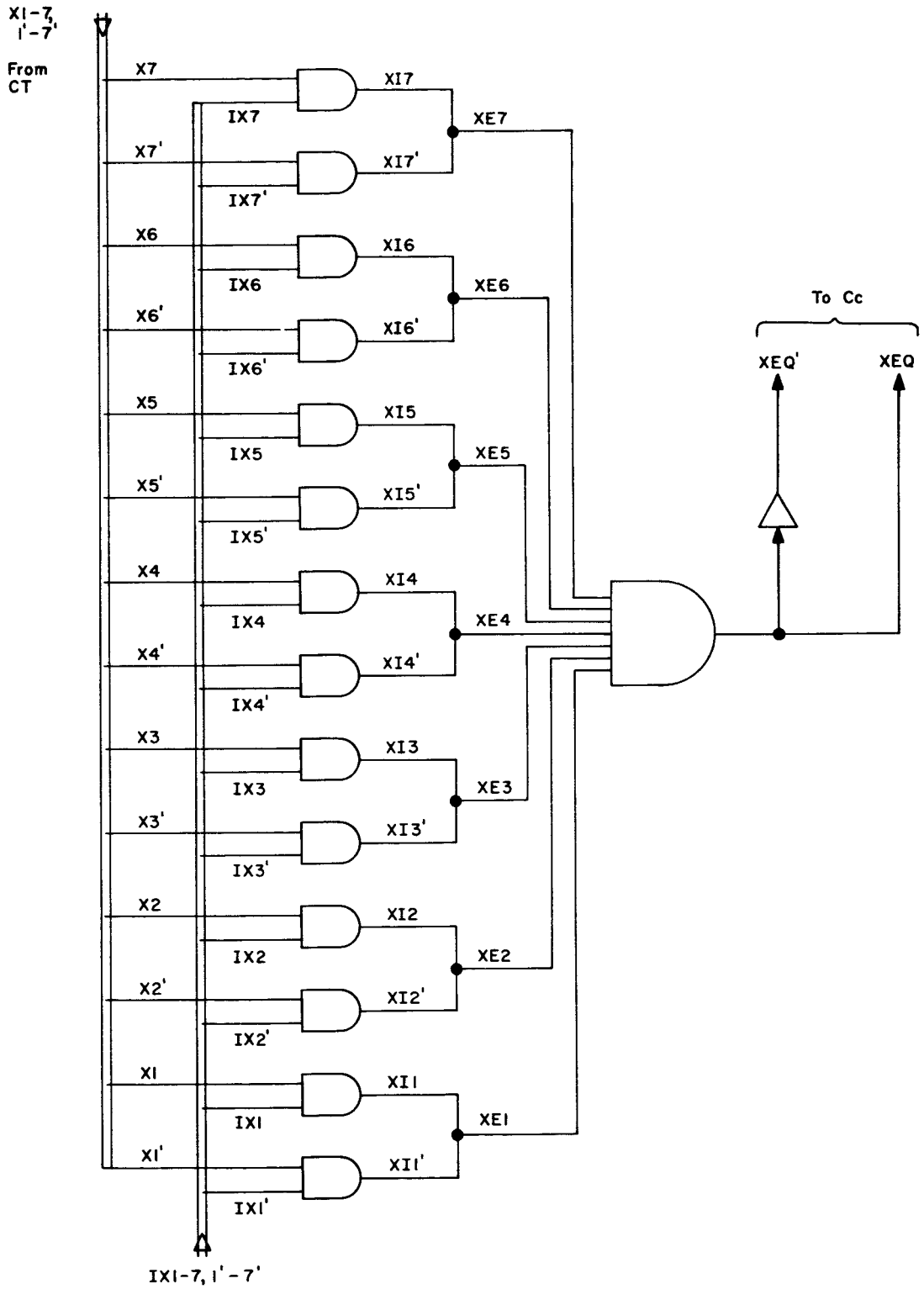
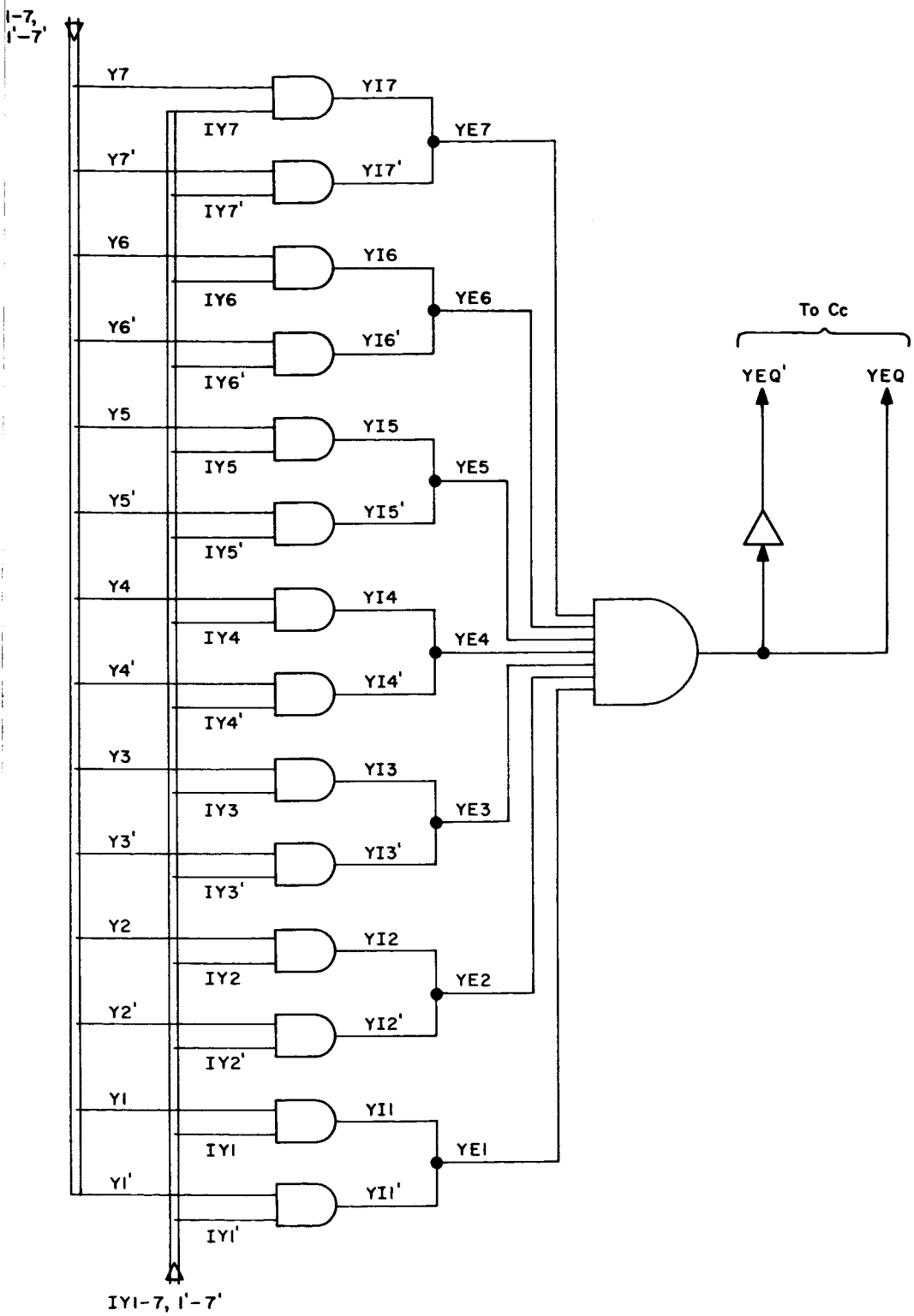


Figure A. 10. Initializatic



n Comparator

30-2

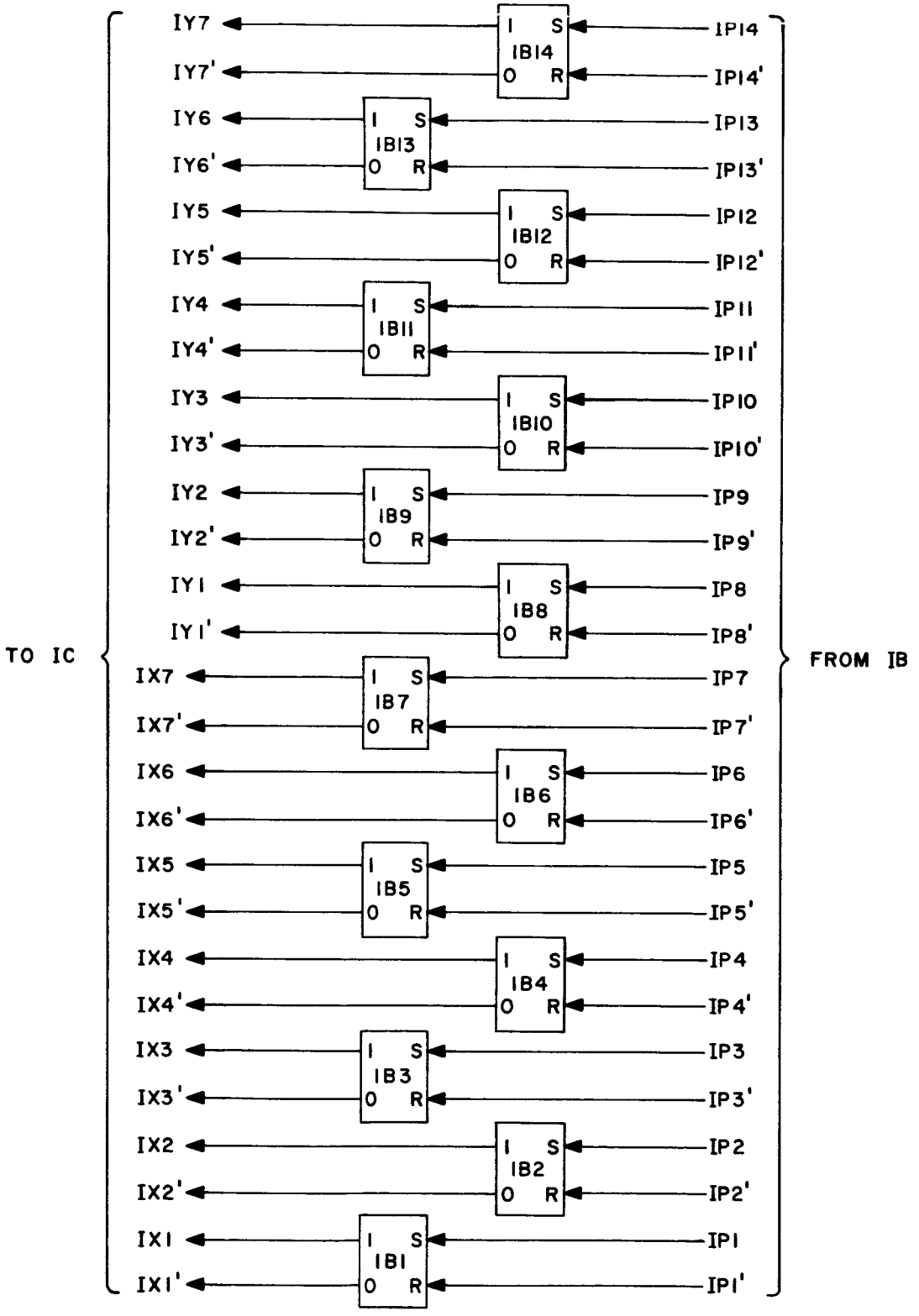
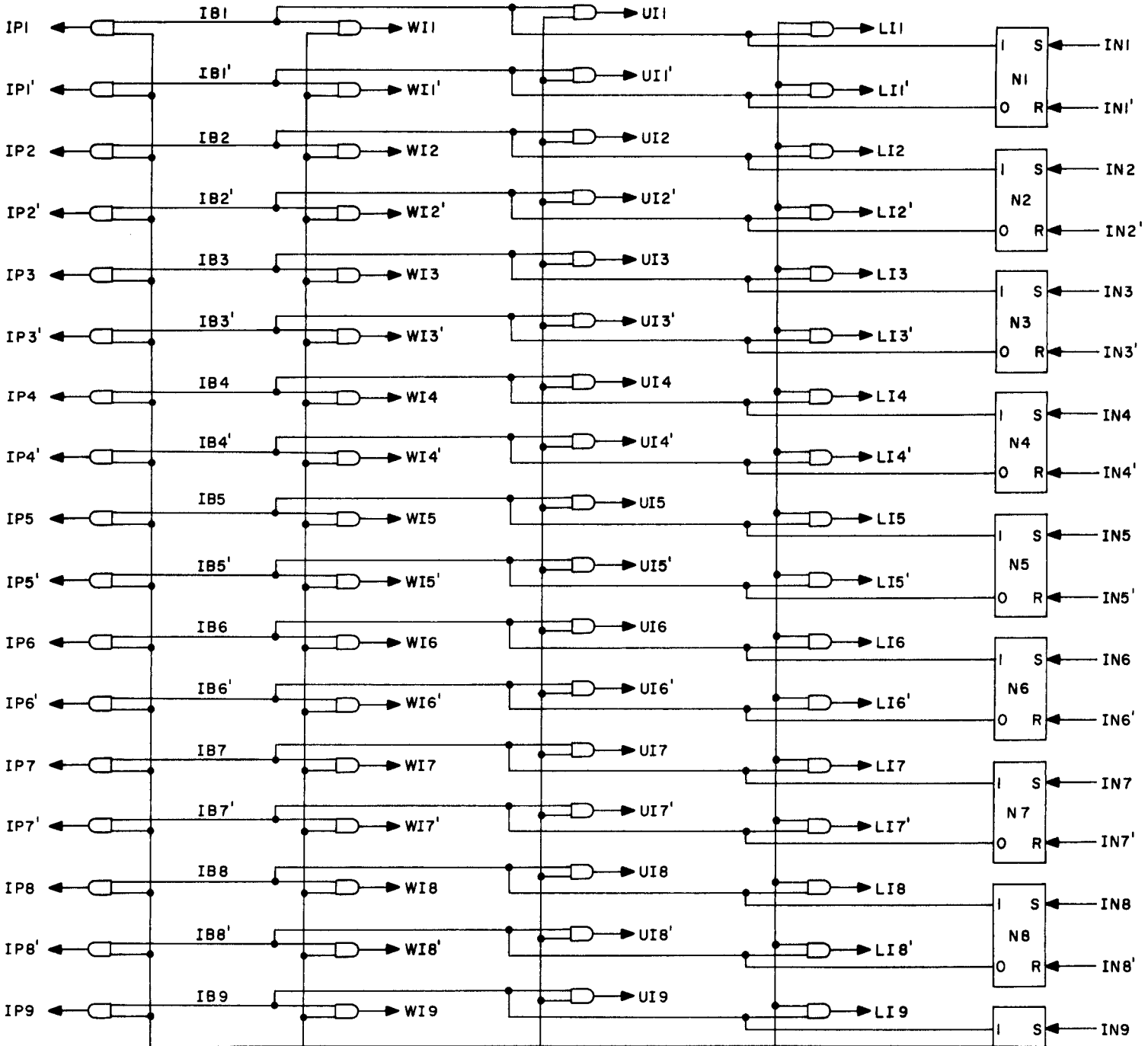


Figure A. 11. Initial Position Register


IP1
IB1
WI1
UI1
LI1
I S
N1
O R
IN1
IP1'
IB1'
WI1'
UI1'
LI1'
IN1'
IP2
IB2
WI2
UI2
LI2
N2
IN2
IP2'
IB2'
WI2'
UI2'
LI2'
IN2'
IP3
IB3
WI3
UI3
LI3
N3
IN3
IP3'
IB3'
WI3'
UI3'
LI3'
IN3'
IP4
IB4
WI4
UI4
LI4
N4
IN4
IP4'
IB4'
WI4'
UI4'
LI4'
IN4'
IP5
IB5
WI5
UI5
LI5
N5
IN5
IP5'
IB5'
WI5'
UI5'
LI5'
IN5'
IP6
IB6
WI6
UI6
LI6
N6
IN6
IP6'
IB6'
WI6'
UI6'
LI6'
IN6'
IP7
IB7
WI7
UI7
LI7
N7
IN7
IP7'
IB7'
WI7'
UI7'
LI7'
IN7'
IP8
IB8
WI8
UI8
LI8
N8
IN8
IP8'
IB8'
WI8'
UI8'
LI8'
IN8'
IP9
IB9
WI9
UI9
LI9
IN9

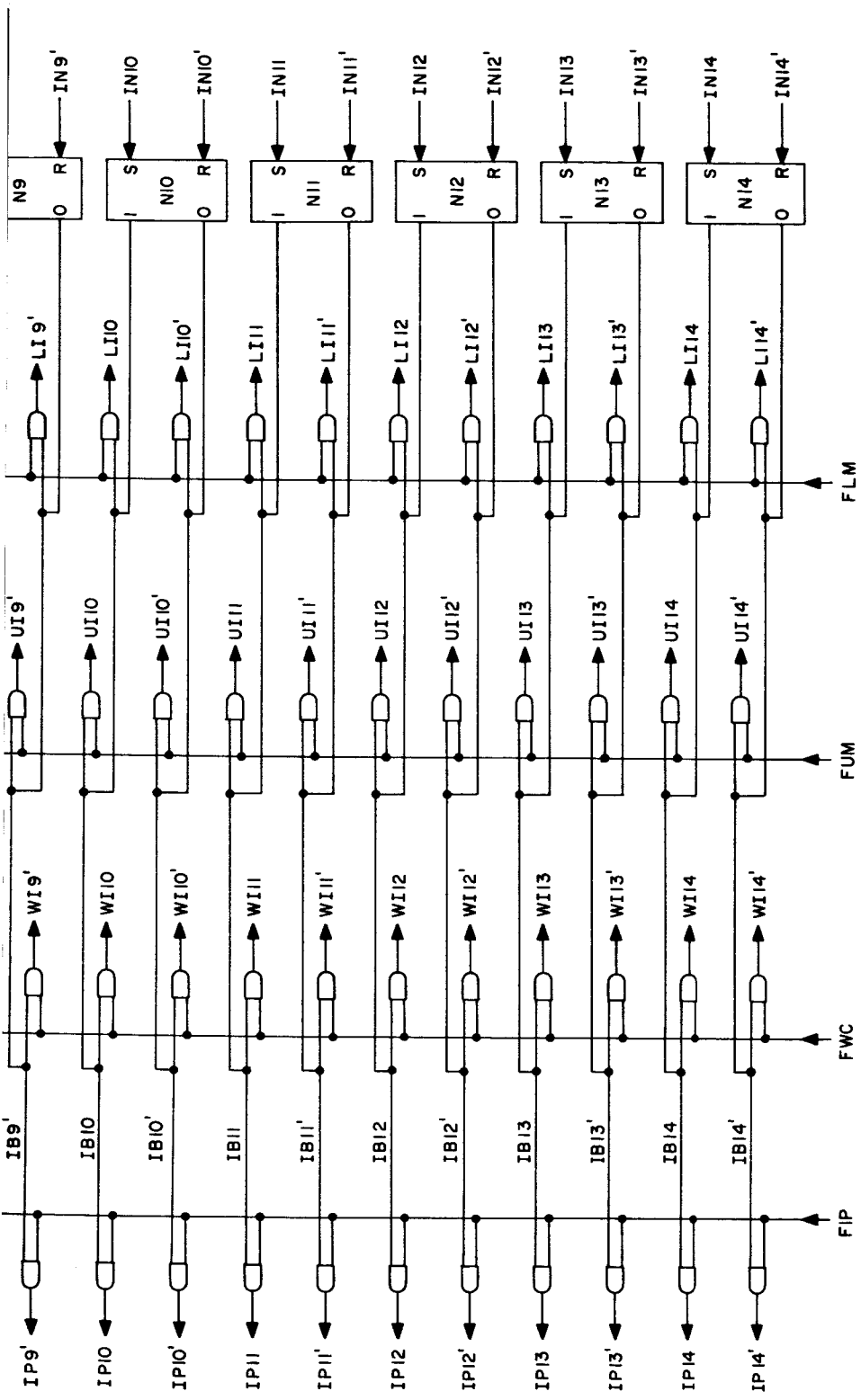


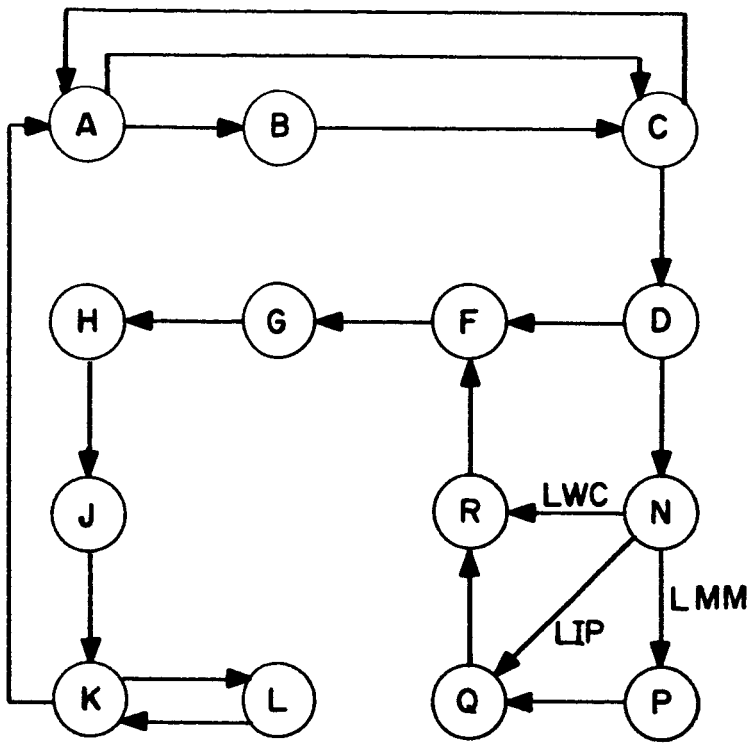
Figure A.12. Input Buffer

32-2

APPENDIX B

LOGIC DIAGRAMS

The logic diagrams from the author's thesis are here given.



Notes:

1. The exits from State N are controlled by the external control, and are labeled with the appropriate command from external control.

2. The process will halt at D if external control does not sample the contents of the output buffer and return a POB (prepare output buffer) signal.

3. The process will halt at P, Q, or R if the input buffers are not filled and the signal IBF (input buffer filled) returned by external control.

Figure B. 1. State Diagram of Control

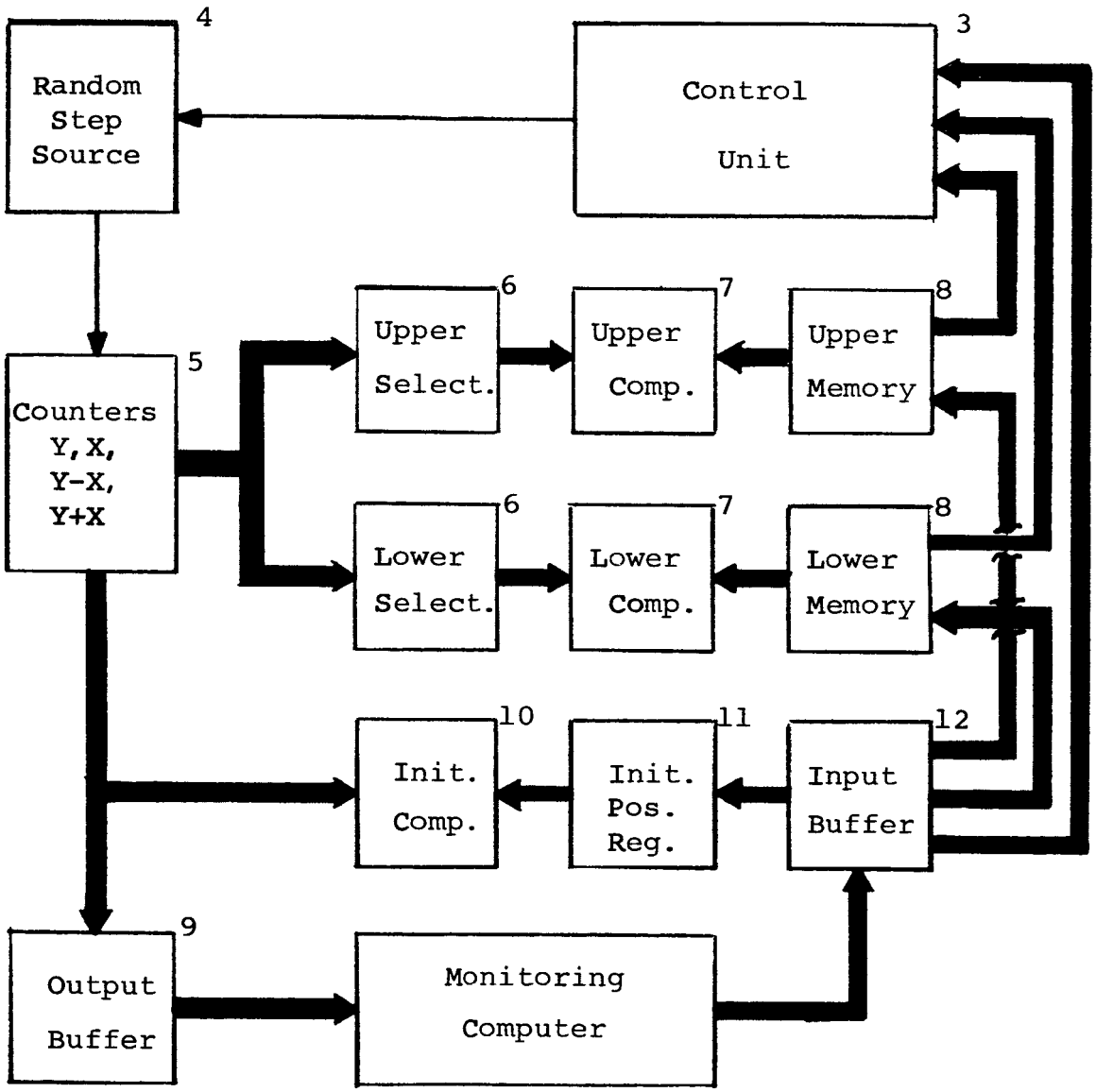


Figure B. 2. Block Diagram of Random Walk Machine with Data Paths

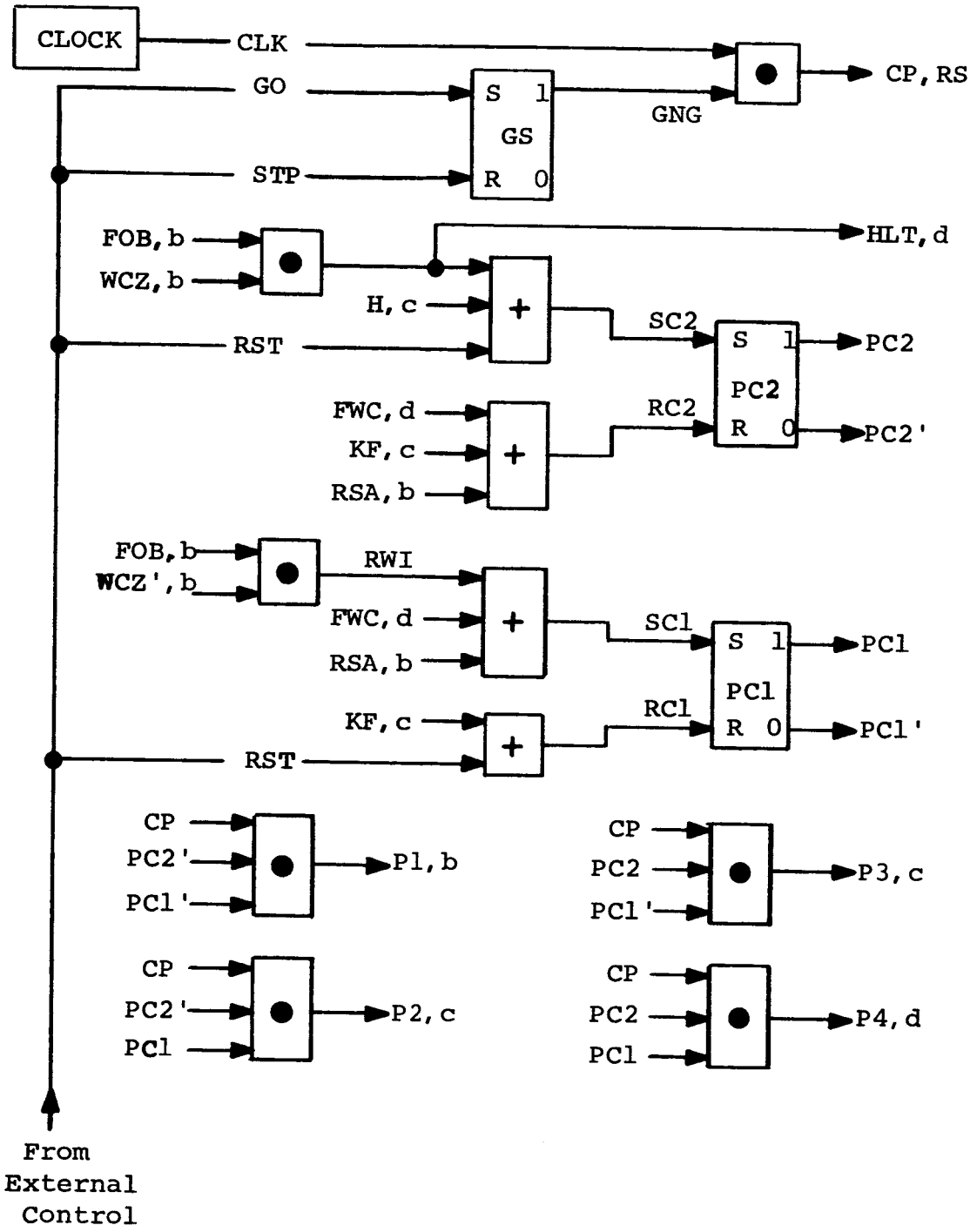


Figure B.3a. Clock and Phase Control

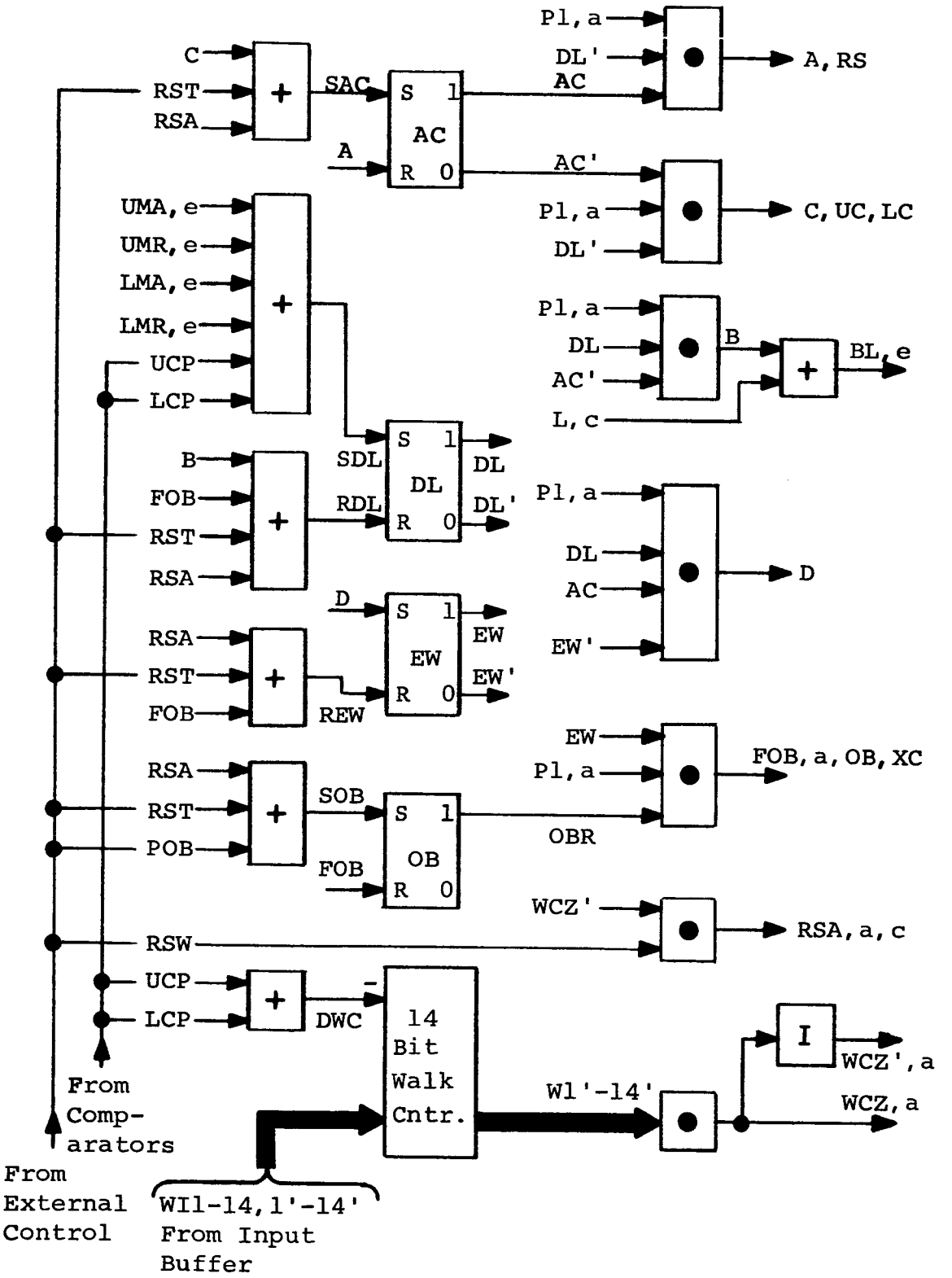


Figure B.3b. Random Step Control

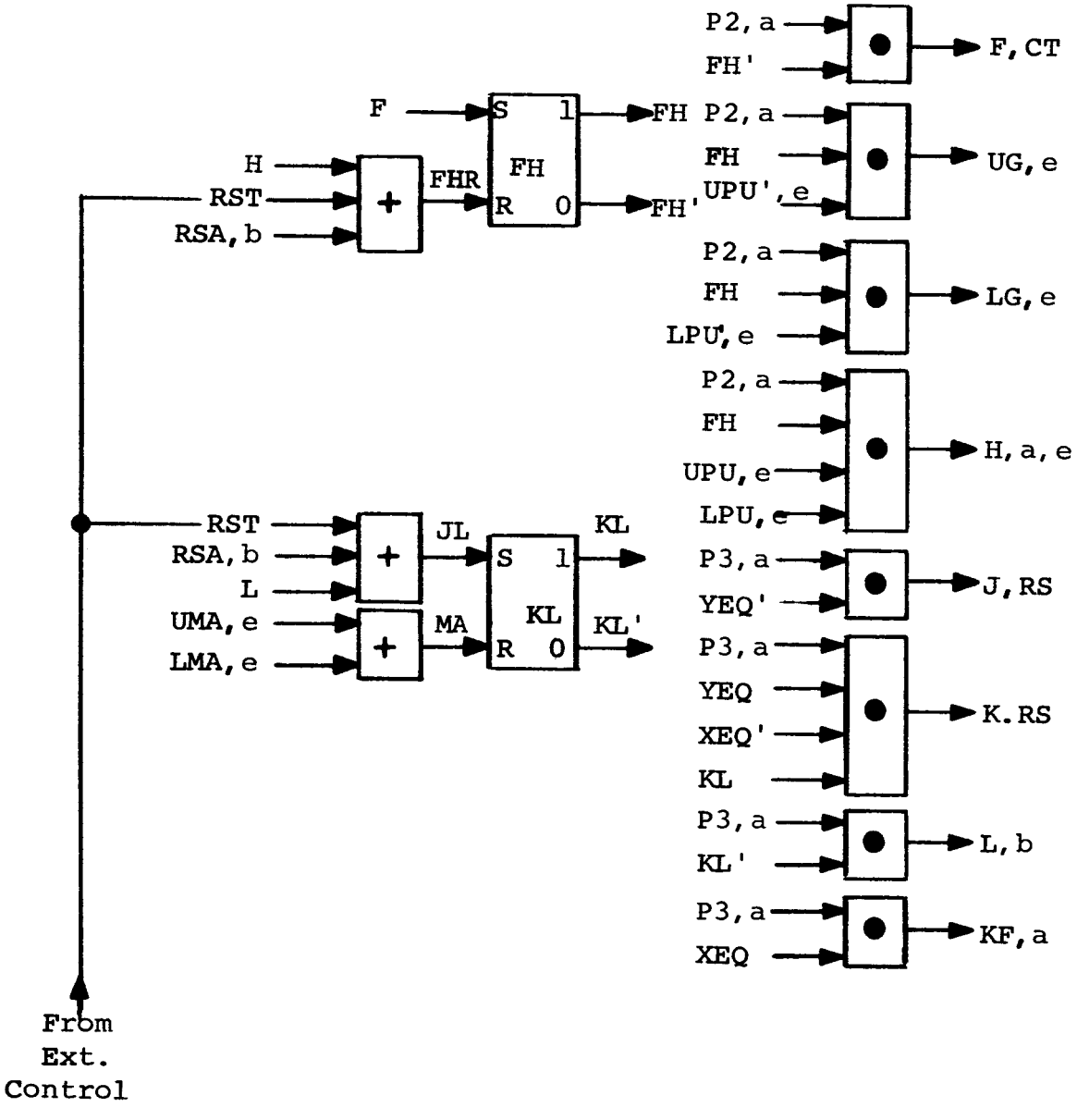


Figure B.3c. Initialization Control

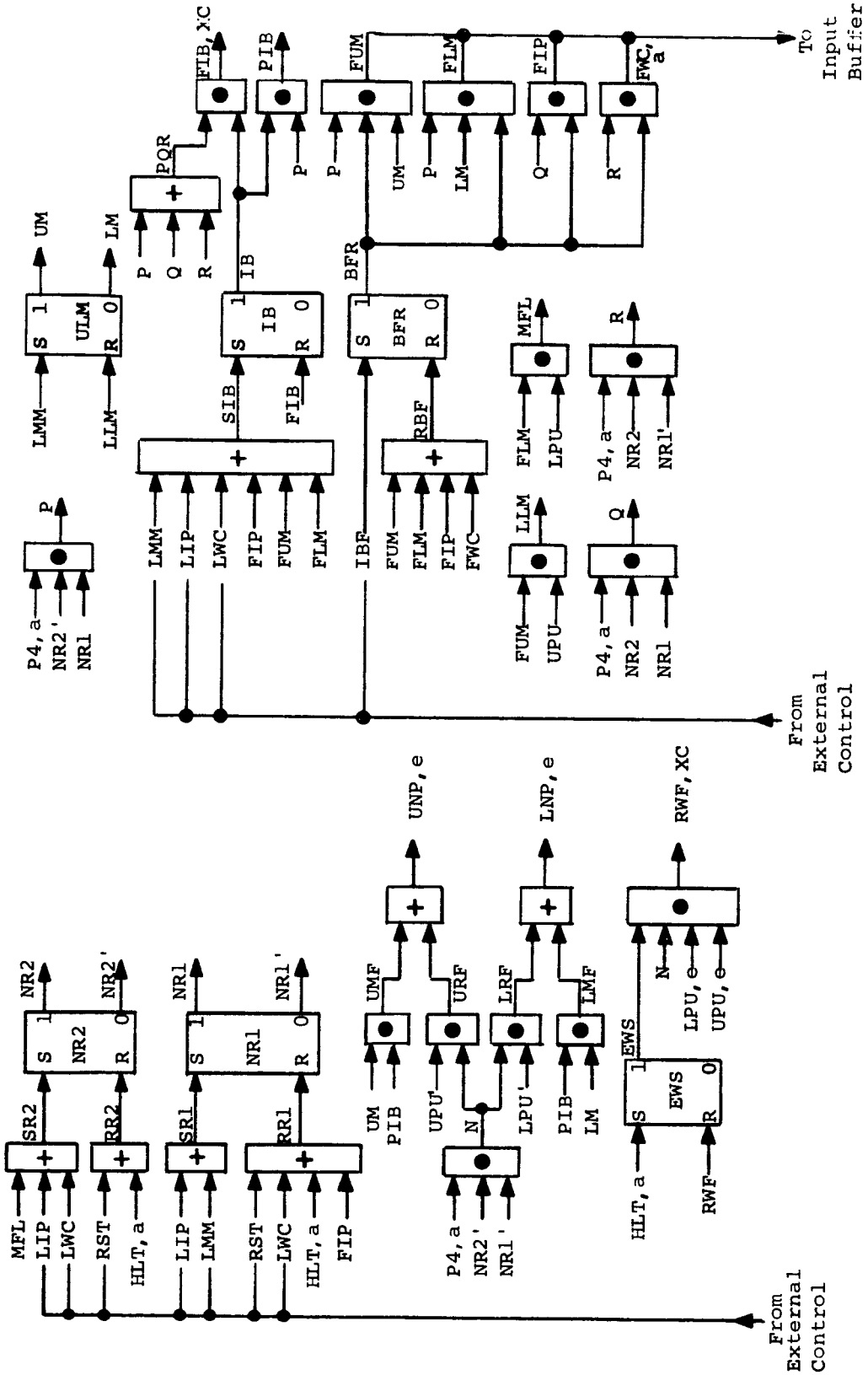
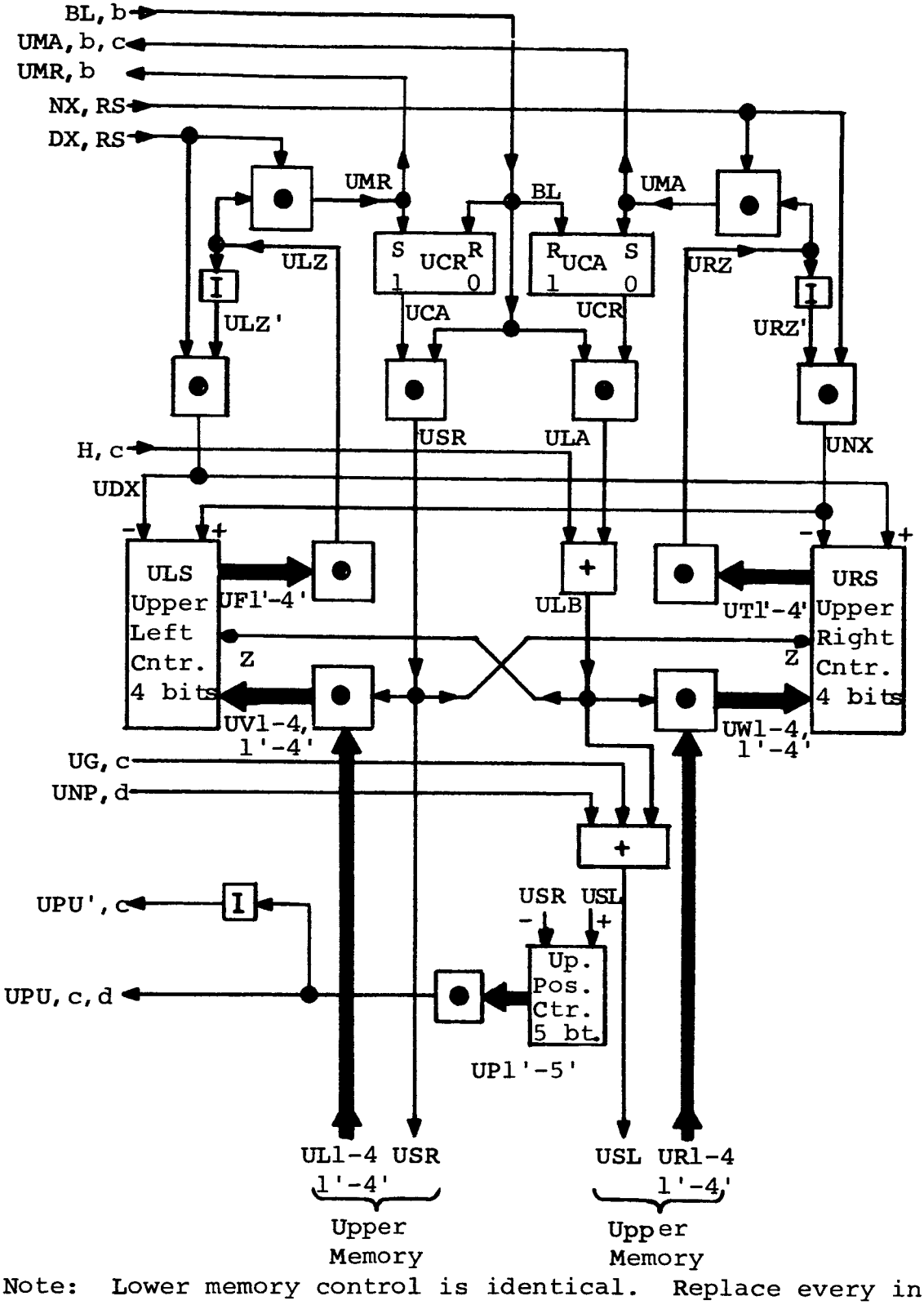


Figure B.3d. Loading Control



Note: Lower memory control is identical. Replace every initial "U" in a signal designation with "L".

Figure B.3e. Upper Memory Control

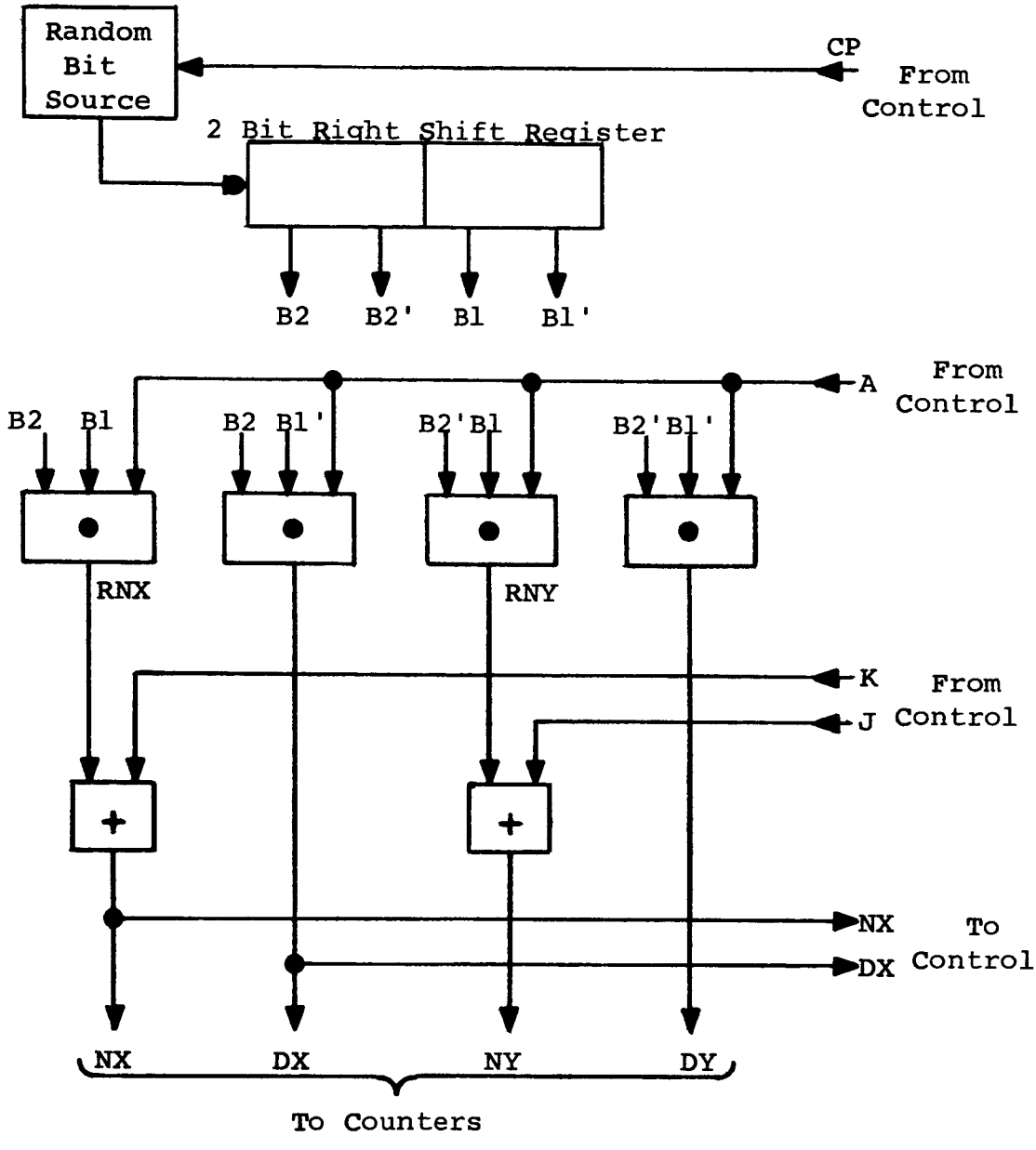


Figure B.4. Random Step Source

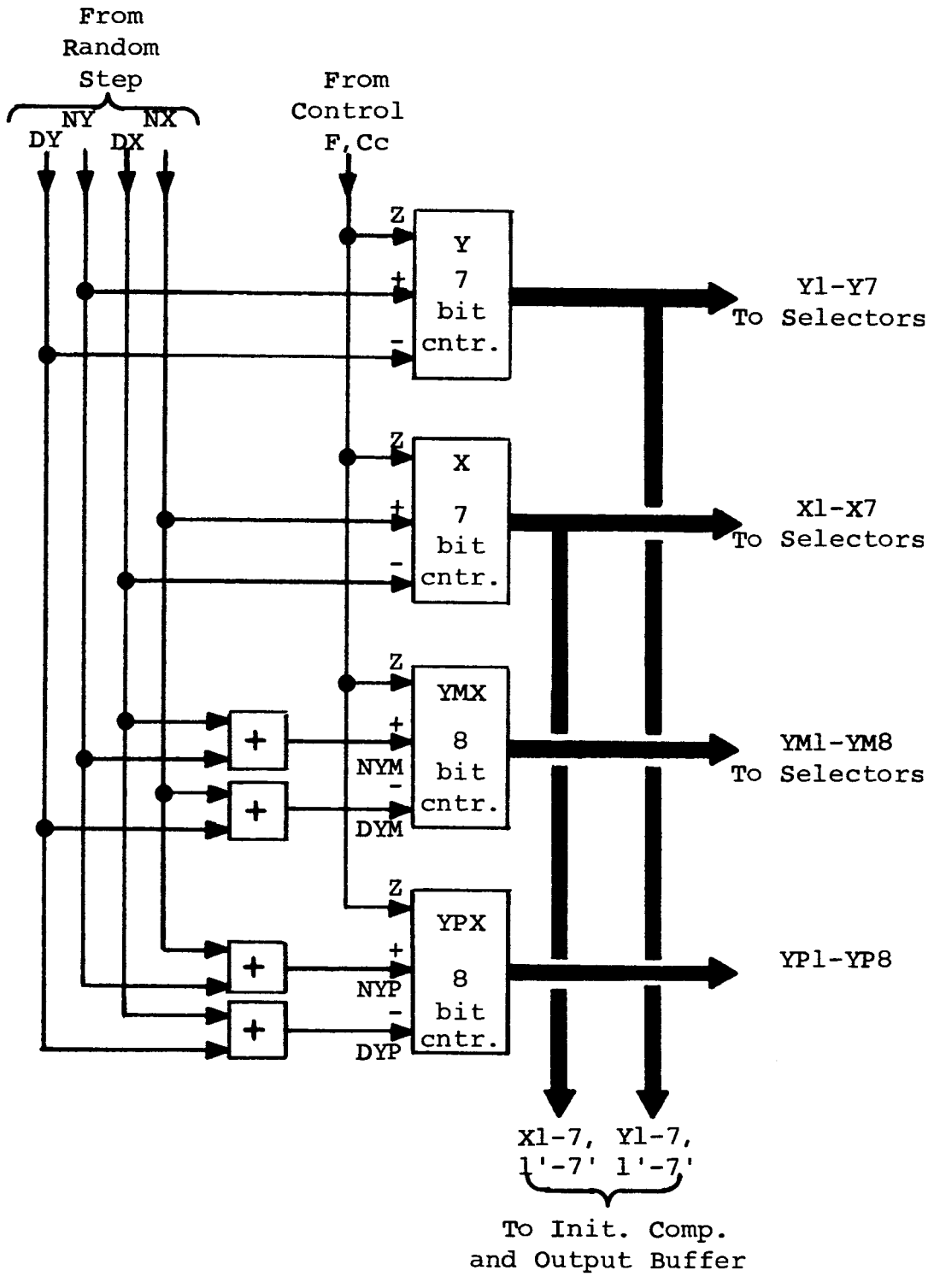


Figure B.5. Counters

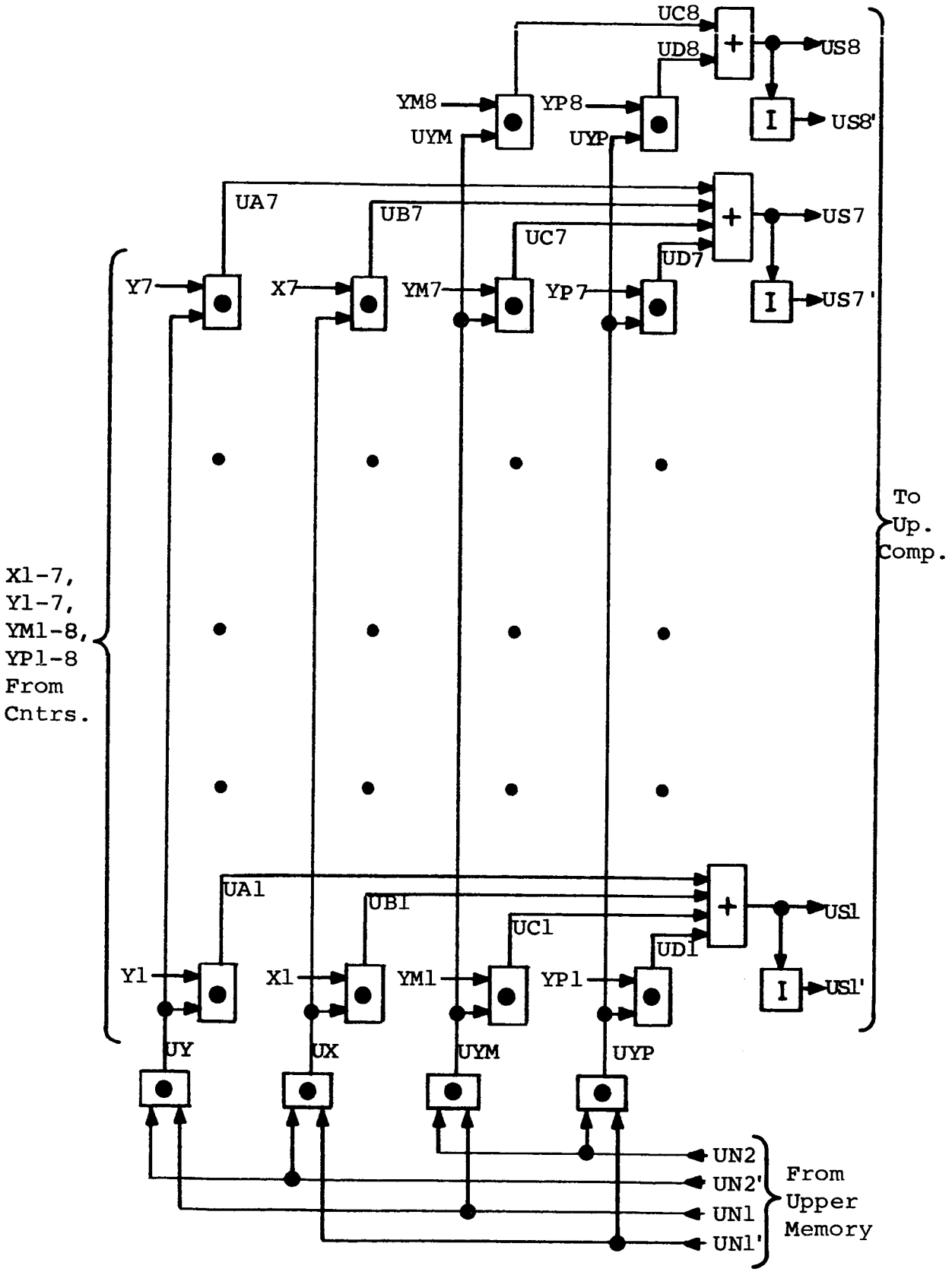


Figure B.6. Upper Selector

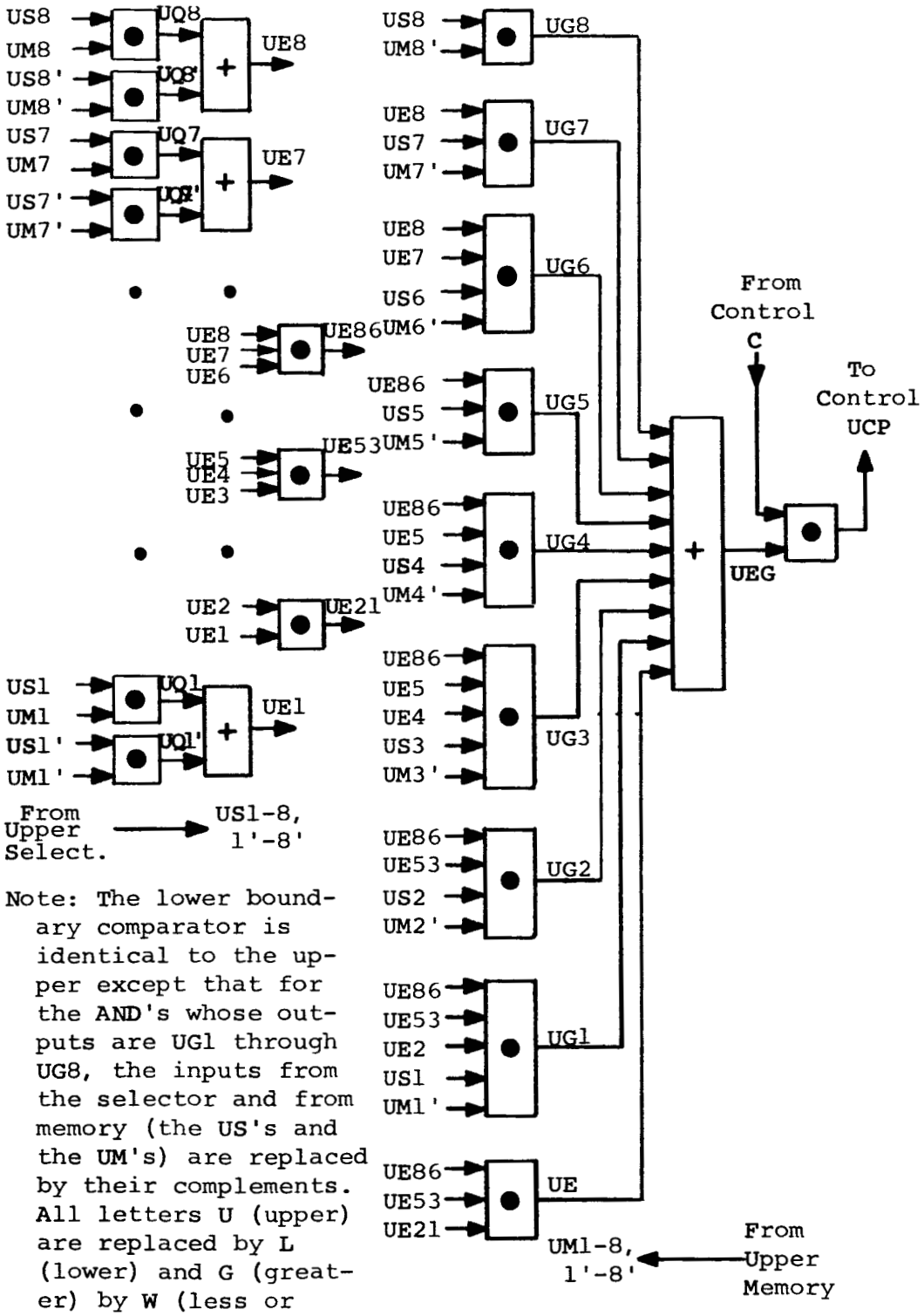
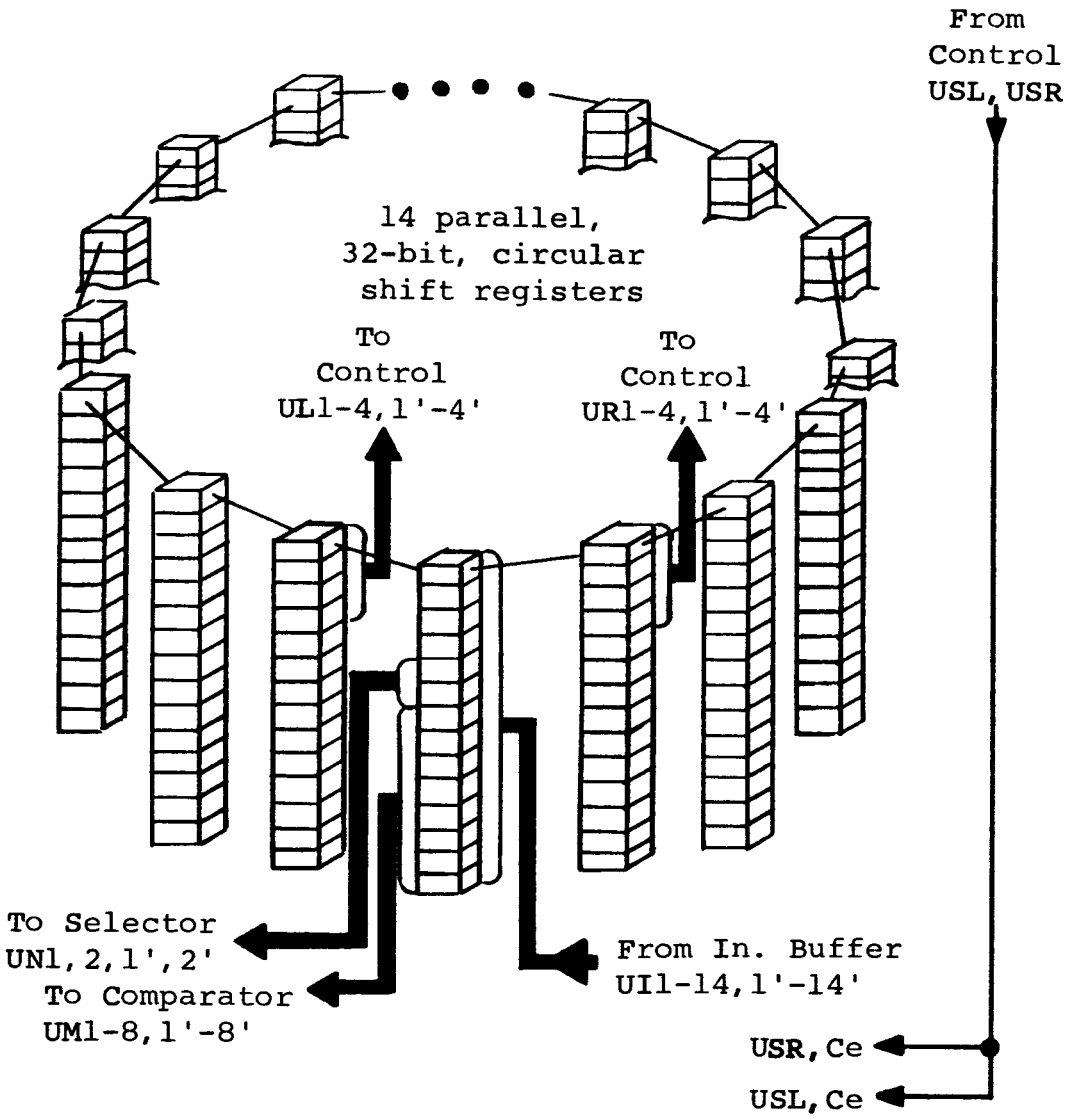


Figure B.7. Upper Comparator



Note: The lower memory is identical to the upper memory. Replace the letter "U" by "L" on signal designations.

Figure B.8. Upper Memory Block Diagram

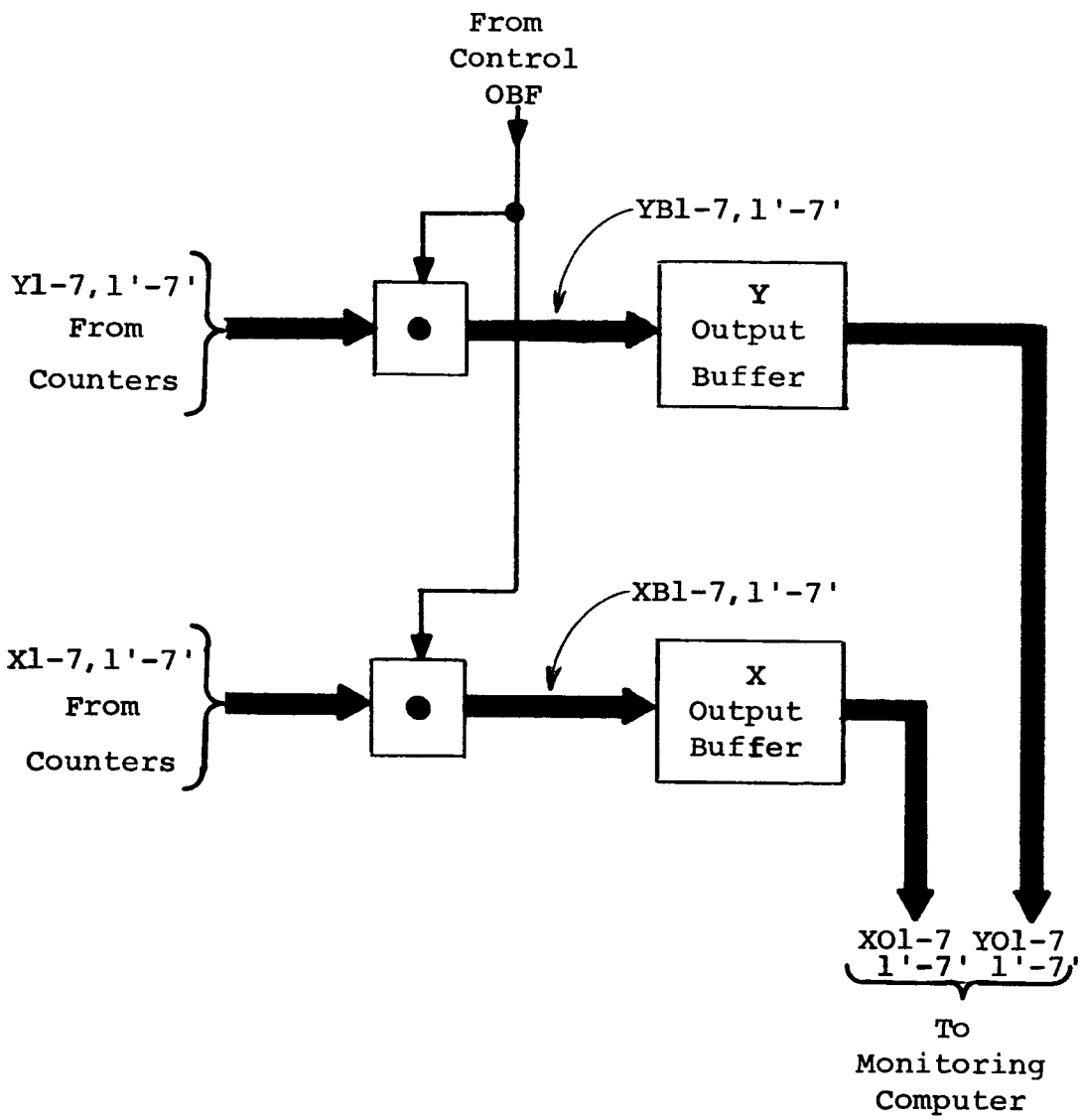
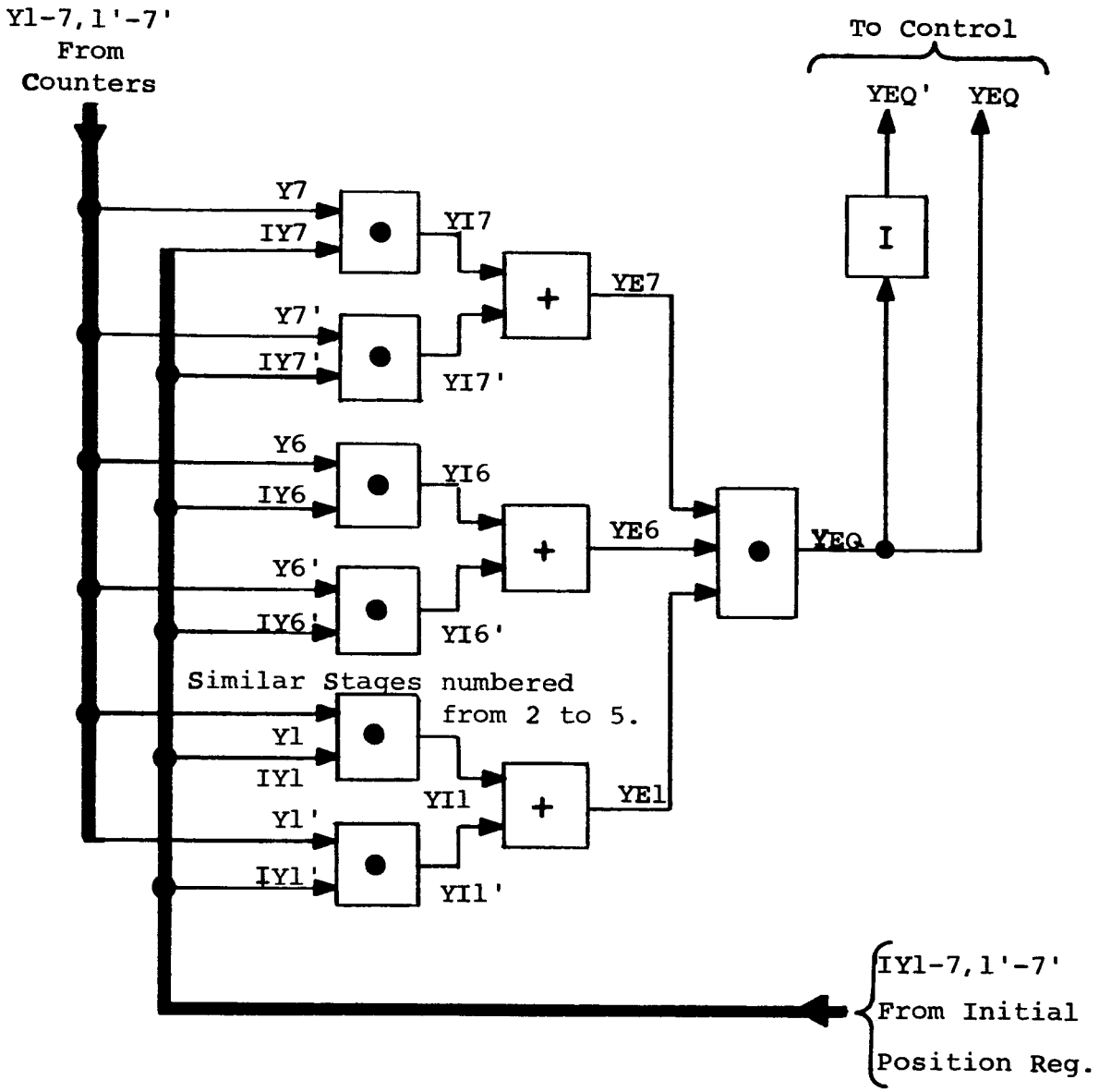


Figure B.9. Output Buffer



Note: The X initialization counter is identical. Replace Y with X.

Figure B. 10. Y Initialization Comparator

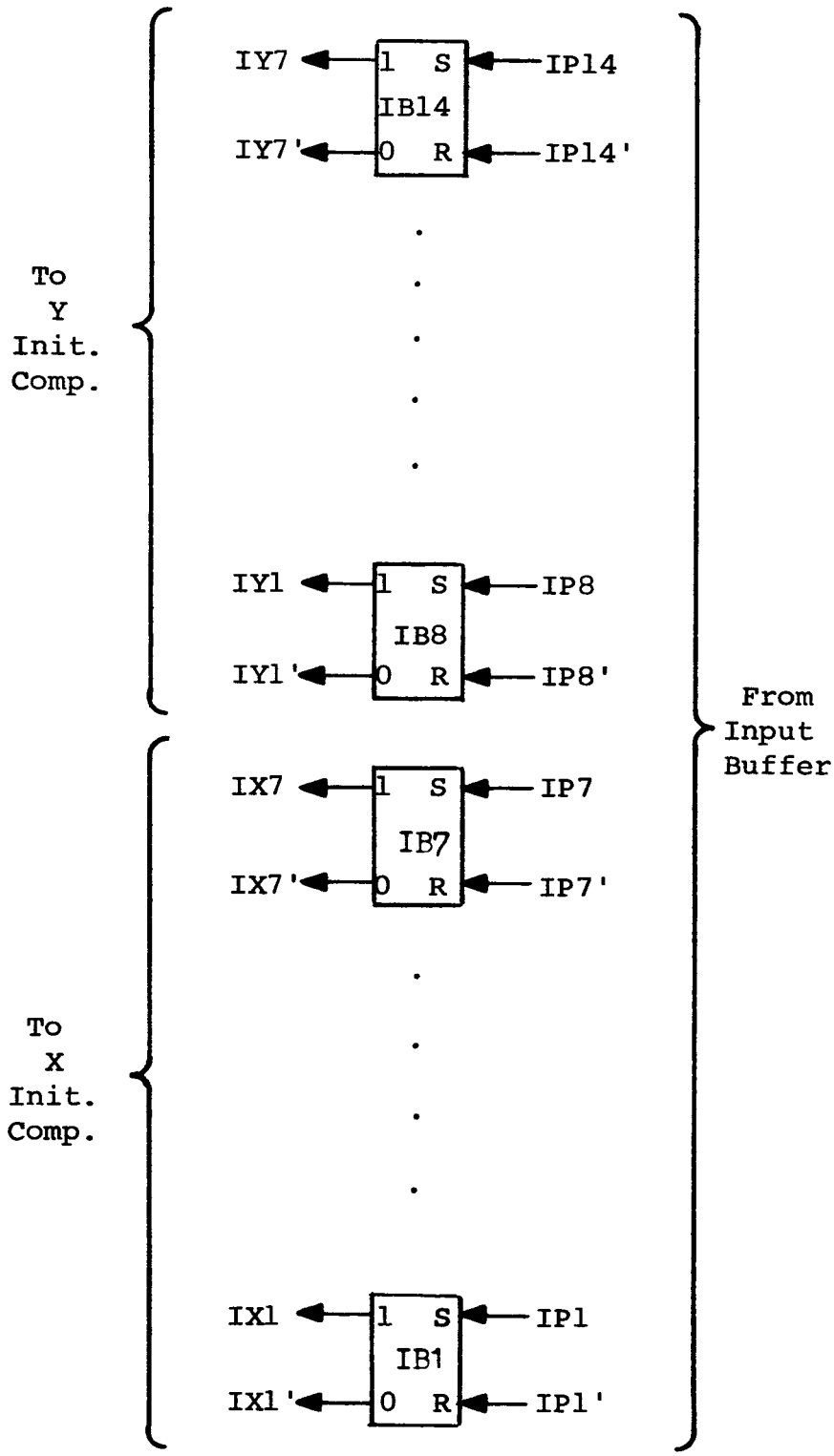


Figure B.11. Initial Position Register

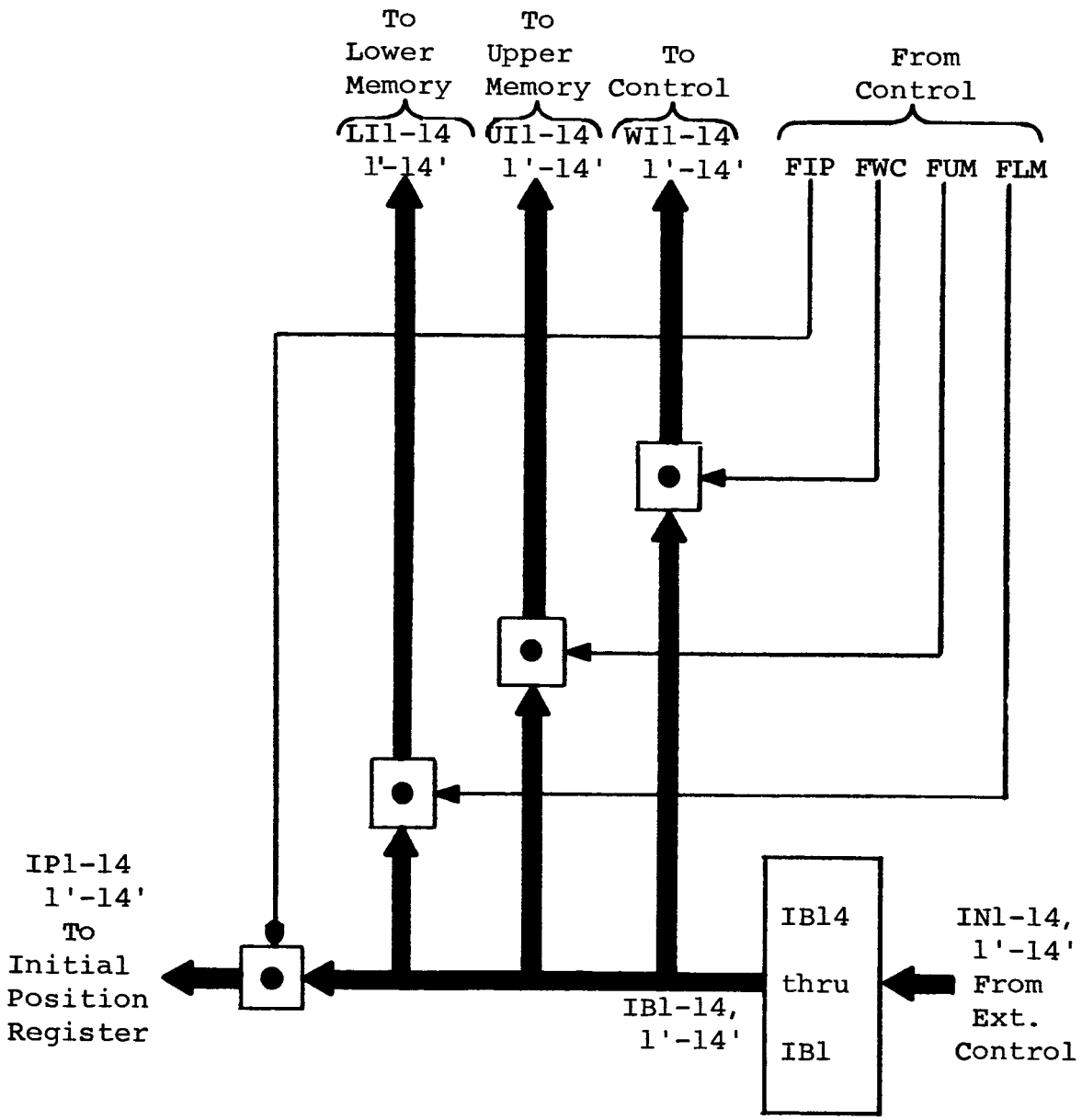


Figure B.12. Input Buffer