ITERATIVE REALIZATION OF MULTIVALUED LOGIC SYSTEMS

Vason P. Srini

Department of Computer Science, Tennessee Technological University
Cookeville, Tennessee 38501

ABSTRACT

The realization of multivalued combinational functions and sequential machines by using arrays of one type of cells is considered. The algebra used for the multivalued logic system has two binary operations and a set of unary operations. Each of these operations is realized by a cellular array. The cells are combinational and implemented by using binary logic gates. The cells are also designed so that all unrestricted multiple faults in arrays of these cells are detectable. Multivalued combinational functions, storage elements and sequential machines are realized by interconnecting the arrays realizing the operations.

Index Terms

Multivalued logic, +gate, ·gate, U gates, cell, cellular array, general fault, multiple faults, Fault detection.

I Motivation

The recent developments [1,2] in the processing technology of large scale integrated circuit (LSI) chips such as 16K and 64 K RAM chips, 64K and 256K CCD chips, and microprocessors have created a flurry of activities in realizing multivalued logic systems and microprocessors based on multivalued logic. Some of the major factors contributing to the above activities are the high circuit density per unit area of silicon, limitation on the number of pinouts, and the need for faster, powerful and economical computers. The success of RAM chips in memory systems have also shown that regularity in circuit structure and interconnection increases the yield of the fabrication process and also improves the diagnosability of the RAM chips [3]. It has also been observed that LSI and very large scale integrated circuits (VLSI) are almost impossible to test economically unless diagnostics is considered during the design phases of the chips [4].

The wide spread use of RAM chips, which consist of identical cells, has aroused interest in the design and fabrication of microprocessors based on cellular structures. If the individual cells are designed to be diagnosable [5], then testing an array of these cells and consequently testing the entire microprocessor might be economically feasible.

An iterative realization of multivalued logic combinational functions and storage elements using cells of just one type, which are implemented by binary logic circuits based on existing and proven technologies such as TTL, I^2L, or ECL, is shown. The functionally complete multivalued algebra of Allen [6] is used for realizing the combinational functions and storage elements. Each operation of the algebra is realized by a cellular array. By suitably interconnecting the arrays realizing the operations the multivalued logic functions are realized.

If the number of logic levels, N, in the algebra is restricted to be a power of 2, then the N-valued clocked storage element of Sintonen [7] can be realized in a straightforward manner. This restriction also allows straightforward encoding to pinouts and decoding from pins.

Several algebras have been proposed for representing multivalued logic functions [8-13] and several realizations for combinational functions and N-valued storage elements using binary logic circuits [8-9] and non-binary logic circuits [10-12, 14] have been developed. Three of these approaches [9-11] are interesting. The vector Boolean algebra approach of Lee [9] is an extension of two valued Boolean algebra into a functionally complete multivalued algebra. The operations n-vector AND, n-vector OR, generalized complement, and rotation form a functionally complete set and each is realized by using binary logic circuits. Three canononical forms for representing multivalued combinational functions have been developed and the forms are realized by using gates corresponding to the operations. But the test generation for multiple faults is quite involved. The N-valued Boolean algebra of Dussault [10] with additional operations for functional completeness has two forms for representation. The realizations of the forms utilizes a decoder, binary circuits for realizing tables of subfunctions and an encoder to pinouts. The realizations are non-iterative, the encoder is quite complex, and there is no provision in the design for

diagnosability. Cellular realization of multivalued logic functions using multivalued q-functions is discussed by Kodandapani [11]. Three different types of cells are used in the realization and the cell design does ot include features which might be helpful in diagnosing.

The algebra to be used in this work is described in Section II. The basic cell to be used for realizing the operations is shown in Section III. The realization of combinational functions, N-valued storage elements and sequential machines are described in Section IV. The testability of the cell is considered. Multiple fault detection in an array of these cells under general fault assumption is discussed in Section V.

II Algebraic System

The multivalued algebraic system considered in this discussion is the one developed by Allen [6]. Let $L = \{0,1,2,3...(N-1)\}$ be the set of N logic values and k is the smallest integer such that $N \leq 2^k$. Each element, x, of L is represented by a k-tuple of zeros and ones such that the decimal value is x. Let + be a binary operator such that $x + y$ = maximum (x,y) and $\cdot$ be a binary operator such that $x \cdot y$ (or xy) = minimum (x,y), where $x,y \in L$.

Let U be a set of unary operators ${}^{a}X^{b}$ such that

$${}^{a}X^{b} = \begin{cases} N-1 & \text{when } a \leq \text{logical value of } X \leq b \\ 0 & \text{when logical value of } X < a \text{ or logical value of } X > b, \end{cases}$$

$a,b \in L$ and $a \leq b$.

The system (L, +, $\cdot$, U) is a functionally complete multivalued algebra. The binary operations are associative and commutative.

III Realizing the Operations

The two binary operations and the set of unary operations are realized by using cellular arrays. The basic cell [15] is combinational and implemented by using binary logic circuits. The description of the cell is included for the sake of completeness.

The cell has three kinds of inputs. One input is from the external world, called primary input ($x_{i,j}$). The second input is from the left, called left pinout (S_1S_2) and the third input is from the top, called top input ($y_{i,j}$). The cell has two outputs: one to the right, called right output (P_1P_2) and the second to the bottom, called bottom output ($z_{i,j}$). All input and output lines are binary. We assume that the complement of $z_{i,j}$ ($\bar{z}_{i,j}$) is also available as bottom output. A typical cell along with the truth table realized by it is shown in Figure 1.

+ gate: This gate implements the + operation. It consists of a one-dimensional array of k cells with the interconnetion pattern shown in Figure 2. If $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,k})$ and $y_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,k})$ are two multivalued logic variables, then $x_i + y_i$ can be computed by using y_i as the top input, x_i as the primary input and 00 as the left input to the array of k cells.

The array operates in the following manner. Starting from the leftmost cell, the values of $x_{i,j}$ and $y_{i,j}$, $1 \leq j \leq k$ are compared. If $x_{i,j} = 1$ and $y_{i,j} = 0$ then $x_i=(x_{i,1}, x_{i,2}, \ldots, x_{i,k})$ is the maximum and the value of x_i is given as output $(z_{i,1}, z_{i,2}, \ldots, z_{i,k})$. If $x_{i,j} = 0$ and $y_{i,j} = 1$ then $y_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,k})$ is the maximum and the value of y_i is given as output $(z_{i,1}, z_{i,2}, \ldots, z_{i,k})$. If $x_{i,j} = 1$ (0) and $y_{i,j} = 1$ (0) then it is not possible to decide which is maximum from the first j cells in the array. The outputs $z_{i,k}$, $1 \leq k \leq j$ are 1's (0's). Subsequent cells are to be compared until either one of the two cases mentioned above happens or all cells are compared.

$\cdot$ gate: This gate implements $\cdot$ operation. It is similar to + gate except that the left input is 1 1.

Since the binary operations are associative, $(x_1 + x_2 + \ldots + x_n)$ can be computed by a two-dimensional array of (n-1) cells and k columns with the left boundary set to a sequence of 0's and $(x_1\ x_2 \ldots x_n)$ can be realized by the above array if the left boundary is set to a sequence of 1's.

Ugates: Each elements, ${}^{a}X^{b}$, in the set U of unary operations is realized by a two-dimensional array of Figure 3. The left half of the first row compares the logic value of the variable X and a. If a is less than or equal to the logic value of X then the output of the right half of the row is (N-1). Otherwise the output is 0. The output of the left half of the row is the maximum of a and X and this is compared to b in the left half of the second row. If the logic value of X is less than or equal to b then the output of the right half of the second row is the output of the right half of the first row. Otherwise the output is 0. The output of the left half of the second row is ignored.

IV Realizing Logic Systems

The realization of combinational functions of multivalued logic is first discussed. Let $x_1, x_2, \ldots, x_n$ be the input variables that can take values from the set L and let F be the

output variable. Let α denot the n-tuple of logic values associated with input variables and the output be $F(\alpha) \in L$. Then, a table with N^n rows and (n+1) columns can be used to specify the combinational function. The function can be represented by the form:

$$F = F(\alpha_1)\ {}^{\alpha_{1,1}}{}_{x_1}\ {}^{\alpha_{1,1}}\ {}^{\alpha_{1,2}}{}_{x_2}\ {}^{\alpha_{1,2}}\ \ldots\ {}^{\alpha_{1,n}}{}_{x_n}\ {}^{\alpha_{1,n}} + F(\alpha_2)\ {}^{\alpha_{2,1}}{}_{x_1}\ {}^{\alpha_{2,1}}\ {}^{\alpha_{2,2}}{}_{x_2}\ {}^{\alpha_{2,2}}\ \ldots\ {}^{\alpha_{2,n}}{}_{x_n}\ {}^{\alpha_{2,n}} + \ldots + F(\alpha_i)\ {}^{\alpha_{i,1}}{}_{x_1}\ {}^{\alpha_{i,1}}\ {}^{\alpha_{i,2}}{}_{x_2}\ {}^{\alpha_{i,2}}\ \ldots\ {}^{\alpha_{i,n}}{}_{x_n}\ {}^{\alpha_{i,n}} + \ldots + F(\alpha_{N^n})\ {}^{\alpha_{N^n,1}}{}_{x_1}\ {}^{\alpha_{N^n,1}}\ {}^{\alpha_{N^n,2}}{}_{x_2}\ {}^{\alpha_{N^n,2}}\ \ldots\ {}^{\alpha_{N^n,n}}{}_{x_n}\ {}^{\alpha_{N^n,n}}$$

where $\alpha_i = (\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n})$,

$$\alpha_{i,j} \in L,\ 1 \le i \le N^n,\ 1 \le j \le n,$$

such that the N-ary value of the n-tuple is (i-1). The number of terms in the above form can be reduced by using the minimization algorithm of Allen [6].

The above form can be realized by using +gates, · gates, U gates, and the logical constants of L.

A combinational function for a 3 valued system is shown in Table 1 [6].

Table 1

x_1	x_2	F
0	0	2
0	1	2
0	2	0
1	0	1
1	1	2
1	2	2
2	0	0
2	1	2
2	2	2

$$F = 2\ {}^{0}{}_{x_1}{}^{0}\ {}^{0}{}_{x_2}{}^{0} + 2\ {}^{0}{}_{x_1}{}^{0}\ {}^{1}{}_{x_2}{}^{1} + 1\ {}^{1}{}_{x_1}{}^{1}\ {}^{0}{}_{x_2}{}^{0} + 2\ {}^{1}{}_{x_1}{}^{1}\ {}^{1}{}_{x_2}{}^{1} + 2\ {}^{1}{}_{x_1}{}^{1}\ {}^{2}{}_{x_2}{}^{2} + 2\ {}^{2}{}_{x_1}{}^{2}\ {}^{1}{}_{x_2}{}^{1} + 2\ {}^{2}{}_{x_1}{}^{2}\ {}^{2}{}_{x_2}{}^{2}.$$

After minimizing,

$$F = 2\ {}^{0}{}_{x_1}{}^{0}\ {}^{0}{}_{x_2}{}^{1} + 1\ {}^{1}{}_{x_1}{}^{1} + 2\ {}^{1}{}_{x_1}{}^{2}\ {}^{1}{}_{x_2}{}^{2}.$$

A realization of the above form is shown in Figure 4.

Realizing multivalued logic sequential machines is now considered. This requires the availability of an N-state storage element in addition to the combinational circuits for realizing next state and output functions. Let the complement of a multivalued logic variable X be denoted by $\bar{X}$ whose value is equal to (N-1) minus the logic value of X. Let $N=2^k$. Then, the clocked N-state storage element of Sintonen [7] can be realized by using two · gates with two inputs, one ${}^{(N-1)}{}_{x}{}^{(N-1)}$ gate, one $\overline{{}^{(N-1)}{}_{x}{}^{(N-1)}}$ gate and one + gate with two inputs. The realization of $\overline{{}^{(N-1)}{}_{x}{}^{(N-1)}}$ is similar to the realization of the gate ${}^{(N-1)}{}_{x}{}^{(N-1)}$ which is in the set of U gates, except that the output is from the complement side of $z_{i,j}$ (i.e. $\bar{z}_{i,j}$). The interconnection of gates realizing the N-valued storage element is shown in Figure 5.

The next state function of the storage element is

$$Q(t+1) = {}^{(N-1)}{}_{x_1}{}^{(N-1)}\ x_2 + \overline{{}^{(N-1)}{}_{x_1}{}^{(N-1)}}\ Q(t),$$

where $x_1, x_2 \in L$.

The clocking state is (N-1) and x_1 is the clocking input.

The realization of sequential machines is illustrated by a synchronous mod N^2 counter, where N = 4. The state table is shown in [7] and excitation functions are:

$\underline{Q_1}$: $x_1 = 3$

$$x_2 = 1^0 Q_1{}^0 + 2\ {}^1Q_1{}^1 + 3\ {}^2Q_1{}^2$$

$\underline{Q_2}$: $x_1 = Q_1$

$$x_2 = 1\ {}^0Q_2{}^0 + 2\ {}^1Q_2{}^1 + 3\ {}^2Q_2{}^2.$$

The block diagram for realizing the synchronous mod N^2 counter is shown in Figure 6.

V Fault Detection

A cell is considered faulty if the behavior of the cell changes to any combination function other than the one shown in Figure 1. This assumption is known as general fault assumption [16, 17]. The detection of faults in one and two-dimensional arrays when one or more cells have failed is considered.

The following conditions are necessary and sufficient for an array to be testable [16, 17]:

a) It should be possible to apply to the input terminals of any cell, a complete set of tests for detecting all faults in the cell, independent of the position of the cell in the array.

b) For each such test, it should be possible to propagate the effect of the fault to an observable output.

Based on the above conditions Prasad [5] had developed a simple design criterion for multiple fault detection under general fault assumption. Let a cell be characterized by a function g: X x Y → X, where X is the set of left inputs and Y is the set of top inputs. Let there exist an element "b" belonging to Y such that g(s,b) = μ(s) for some permutation μ on X, where s ε X. Then the test for one-dimensional array is [5],

$$T^k = \{(s, y_{1,1}, y_{1,2} \cdots y_{1,j}, \cdots y_{1,k}) \mid$$

$$s \in X,\ y_{1,1} = y_{1,2} = \cdots = y_{1,(j-1)} = y_{1,(j+1)}$$

$$= \cdots = y_{1,k} = b,\ y_{1,j} \in Y \text{ for some } j\}\ .$$

We note from Figure 1 that for b = 0, g(s,0) = (s,0) when $x_{i,j} = 0$, and for b = 1, g(s,1) = (s,1) when $x_{i,j} = 1$, with μ the identity permutations on X.

Therefore one-dimensional array of cells shown in Figure 2 is testable.

The test for $x_{i,j} = 0$ when k = 2 is

$$T^2 = \begin{matrix} s \\ y_{1,1} \\ y_{1,2} \end{matrix} \begin{bmatrix} 0&0&1&1&2&2&3&3&0&1&2&3 \\ 0&0&0&0&0&0&0&0&1&1&1&1 \\ 0&1&0&1&0&1&0&1&0&0&0&0 \end{bmatrix},$$

and the test for $x_{i,j} = 1$ is

$$T^2 = \begin{matrix} s \\ y_{1,1} \\ y_{1,2} \end{matrix} \begin{bmatrix} 0&0&1&1&2&2&3&3&0&1&2&3 \\ 1&1&1&1&1&1&1&1&0&0&0&0 \\ 0&1&0&1&0&1&0&1&1&1&1&1 \end{bmatrix}.$$

Let there exist a ε X, b ε Y such that g(s,b) = (μ(s),b) and g(a,c) = (a, ν(c)), where c ε Y and for some arbitrary permutations ν on X and ν on Y. Then the test for a two-dimensional array is [5].

$$T^{p,k} = \{(s_{1,1}, s_{2,1}, \cdots s_{i,1}, \cdots s_{p,1},$$

$$y_{1,1}, y_{1,2}, \cdots y_{1,j}, \cdots y_{1,k}) \mid$$

$$\text{for some } i \text{ and } j,\ s_{q,1} = a,\ q \neq i,\ y_{1,m}$$

$$= b,\ m \neq j,\ s_{i,1} \in X,\ y_{1,j} \in Y\}$$

We note from Figure 1 that for a = 2, b = 0, g(s,0) = (s,0) and g(2,c) = (2,c) when $x_{i,j} = 0$, and for a = 2, b = 1, g(s,1) = (s,1) and g(2,c) = (2,c) when $x_{i,j} = 1$ with μ the identity permutation on X and ν the identity permutation on Y. Therefore two-dimensional arrays of cells are testable. The test set when p = 2 and k = 2 is shown below.

VI Remarks

Realizing multivalued logic systems by using gates constructed from arrays of cells has been discussed. A cellular array can be tested for multiple unrestriced faults by a test set consisting of 2 [8pk - 6(pk - p-k) - 4k - 2p + (p-1)(k-1)] elements, calculated by using [5], where p is the number of rows and k the number of columns in the array. This non-exhaustive testing of arrays of cells reduces the test time for LSI chips using lasers [18] at the production stage and thus removing one of the major obstructions to the production of low cost LSI.

The test set for $x_{i,j} = 0$ when p = 2 and k = 2 is,

$$T^{2,2} = \begin{matrix} s_{1,1} \\ s_{2,1} \\ y_{1,1} \\ y_{1,2} \end{matrix} \begin{bmatrix} 2&2&2&2&2&2&2&2&2&2&2&2&0&0&1&1&3&3&0&1&3 \\ 0&0&1&1&2&2&3&3&0&1&2&3&2&2&2&2&2&2&2&2&2 \\ 0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&1&1&1 \\ 0&1&0&1&0&1&0&1&0&0&0&0&0&1&0&1&0&1&0&0&0 \end{bmatrix}$$

and the test set for $x_{i,j} = 1$ is,

$$T^{2,2} = \begin{matrix} s_{1,1} \\ s_{2,1} \\ y_{1,1} \\ y_{1,2} \end{matrix} \begin{bmatrix} 2&2&2&2&2&2&2&2&2&2&2&2&0&0&1&1&3&3&0&1&3 \\ 0&0&1&1&2&2&3&3&0&1&2&3&2&2&2&2&2&2&2&2&2 \\ 1&1&1&1&1&1&1&1&0&0&0&0&1&1&1&1&1&1&0&0&0 \\ 0&1&0&1&0&1&0&1&1&1&1&1&0&1&0&1&0&1&1&1&1 \end{bmatrix}$$

References

1. T.H.P. Chang, et al, "Electron-beam lithography draws a finer line", Electronics, May 12, 1977, Vol. 50, No. 10, pp. 89-98.

2. L. Altman and C. L. Cohen, "Japan Presses innovations to reach VLSI goal", Electronics, June 9, 1977, Vol. 50, NO. 12, pp. 99-122.

3. V. P. Srini, "Fault location in a semiconductor RAM unit", to appear in IEEE Transactions on Computers, April 1978.

4. R. Spillman, "Single stuck type fault detection in multi-valued combinational circuits", Proc. of 1976 Symposium on Multiplevalued logic design, pp. 97-101.

5. B. A. Prasad and F. G. Gray, "Multiple fault detection in arrays of combinational cells", IEEE Transactions on Computers, Vol. c-24, No. 8, Aug. 1975, pp. 794-802.

6. C. M. Allen and D. D. Givone, "A minimization technique for multiple-valued logic systems", IEEE Transactions on Computers, Vol. c-17, No. 2, Feb. 1968, pp. 182-184.

7. L. Sintonen, "A clocked multivalued flip-flop", IEEE Transactions on Computers, Vol. c-26, No. 3, March 1977, pp. 292-294.

8. B. Benjauthrit and I. S. Reed, "Galois switching functions and their applications". IEEE Transctions on Computers, Vol, c-25, No. 1 Jan. 1976, pp. 78-86.

9. S. C. Lee, "Vector Boolean algebra and calculus", IEEE Transactions on Computers, Vol. c-25, No. 9, Sept. 1976, pp. 865-874.

10. J. Dussault, G. Metze, and M. Kriegar, "A multivalued switching algebra with Boolean properties", Proc. of 1976 Symposium on Multiplevalued logic design, pp. 68-73.

11. K. L. Kodandapani and R. V. Setlur, "A cellular array for multivalued logic functions", Proc. of 1974 Symposium on Multiplevalued logic design, pp. 529-544.

12. R. C. Braddock, G. Epstein, and H. Yamanaka, "Multiplevalued logic design and applications in binary computers", Proc. of 1971 Symposium on Multivalued logic design, Buffalo NY, pp. 13-25.

13. Z. G. Vranesic, E. S. Lee, and K. C. Smith, "A manyvalued algebra for switching systems", IEEE Transaction on Computers, Vol. c-19, No. 10, Oct. 1970, pp. 964-971.

14. D. M. Miller and J. C. Muzio, "A ternary cellular array", Proc. of 1974 Symposium on Multivalued logic design, pp. 469-482.

15. V. P. Srini, "Realization of fuzzy forms", IEEE Transaction on Computers, Vol. c-24, No. 9, Sept. 1975, pp. 941-943.

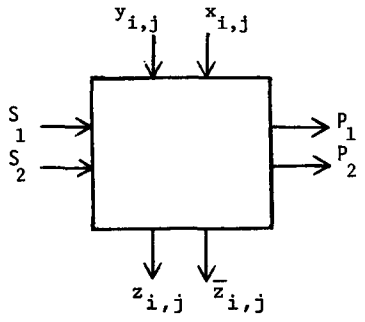
16. A. D. Friedman and P. R. Menon, Fault detection in digital circuits, Prentice-Hall Inc., Englewood Cliffs, NJ 1971.

17. W. H. Kautz, "Testing for faults in cellular logic arrays," Proc. 8-th annual Symp. on Switching and automata theory, 1967, pp. 161-174.

18. J. G. Smith and H. E. Oldham, "Laser testing of integrated circuits", IEEE Journal of Solid State Circuits, Vol. sc-12, No. 3, June 1977, pp. 247-252.

Acknowledgement

The author is thankful to Sue for her encouragement and sustenance.



$x_{i,j}$ - Primary input

$y_{i,j}$ - Top input

S_1S_2 - Left input

P_1P_2 - Right output

$z_{i,j}$ - Bottom output

$\bar{z}_{i,j}$ - Complement of

$x_{i,j} = 0$

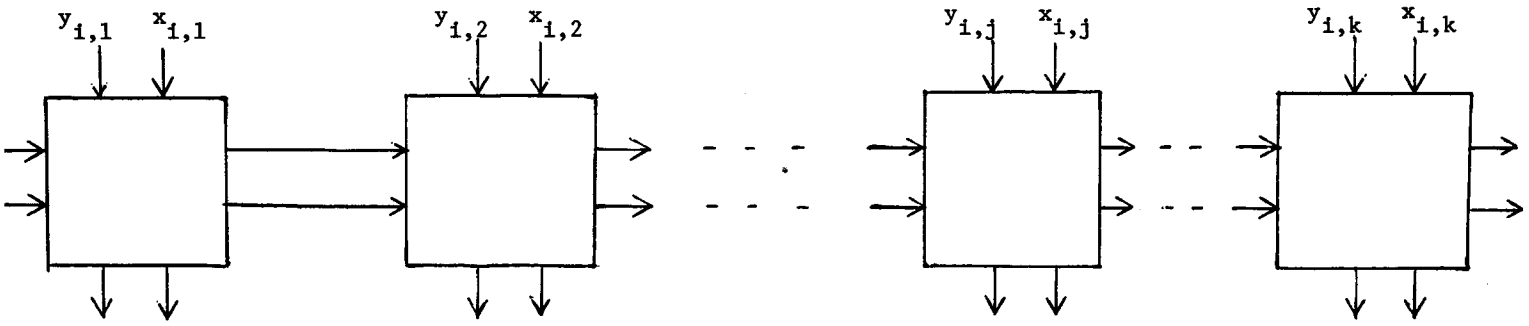
S_1S_2	$y_{i,j}$	0	1
(00)	0	(0,0)	(2,1)
(01)	1	(1,0)	(1,0)
(10)	2	(2,0)	(2,1)
(11)	3	(3,0)	(1,0)

Encoded P_1P_2 $z_{i,j}$

$x_{i,j} = 1$

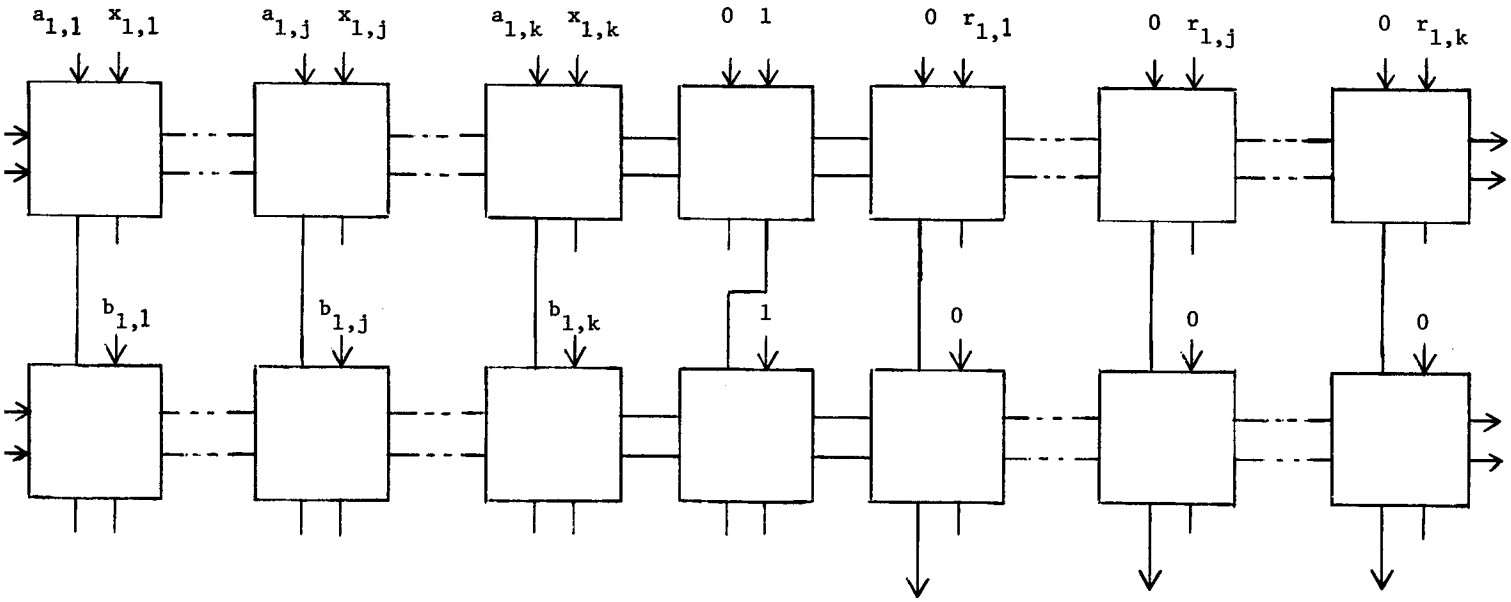
S_1S_2	$y_{i,j}$	0	1
(00)	0	(1,1)	(0,1)
(01)	1	(1,1)	(1,1)
(10)	2	(2,0)	(2,1)
(11)	3	(2,0)	(3,1)

Figure 1. A typical cell and its truth table



$$x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,j}, \ldots, x_{i,k})$$
$$y_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,j}, \ldots, y_{i,k})$$

Figure 2. Realizing + operation



$$a = (a_{1,1}, a_{1,2}, \ldots, a_{r,j}, \ldots, a_{1,k})$$
$$b = (b_{1,1}, b_{1,2}, \ldots, b_{1,j}, \ldots, b_{1,k})$$
$$N\text{-}1 = (r_{1,1}, r_{1,2}, \ldots, r_{1,j}, \ldots, r_{1,k})$$
$$x = (x_{1,1}, x_{1,2}, \ldots, x_{1,j}, \ldots, x_{1,k})$$

Figure 3. Realizing $^{a}X^{b}$

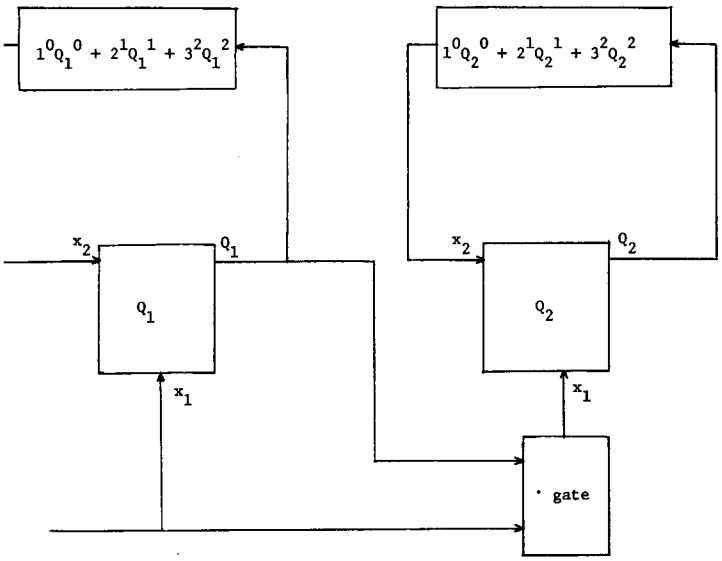


Figure 6. Block diagram for synchronous mod N^2 counter

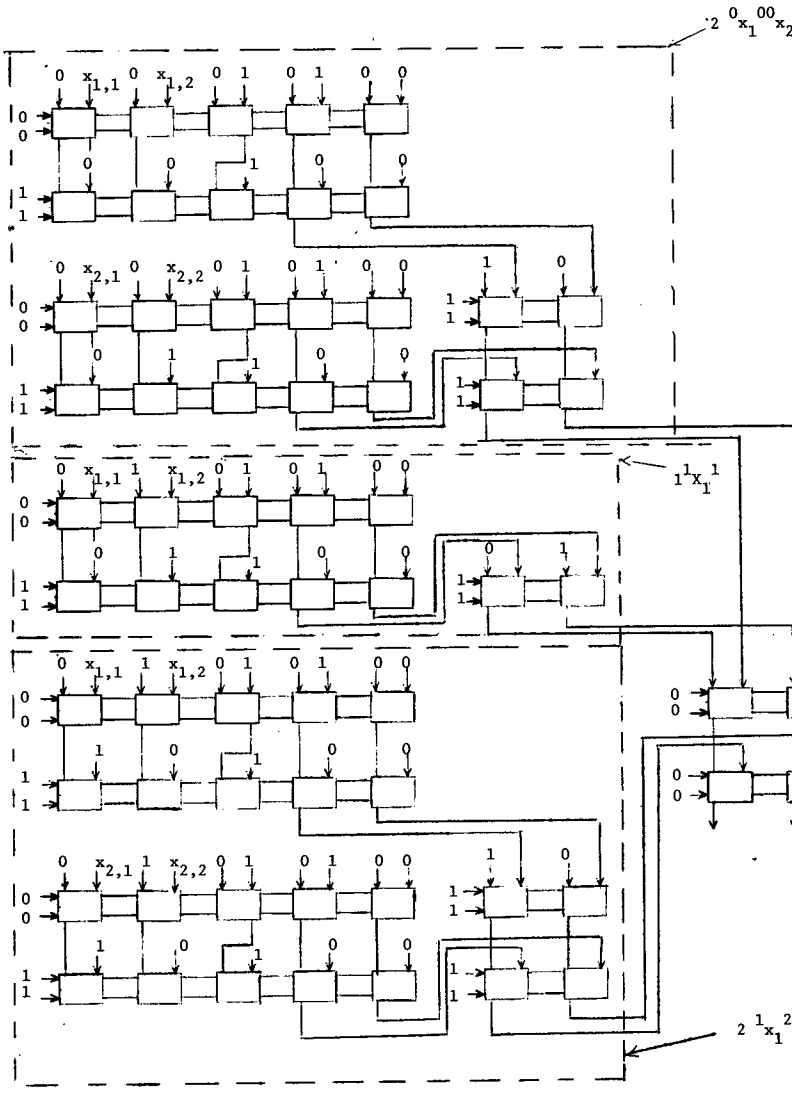


Figure 4. Realizing $2^0x_1^{00}x_2^1 + 1^1x_1^1 + 2^1x_1^{21}x_2^2$

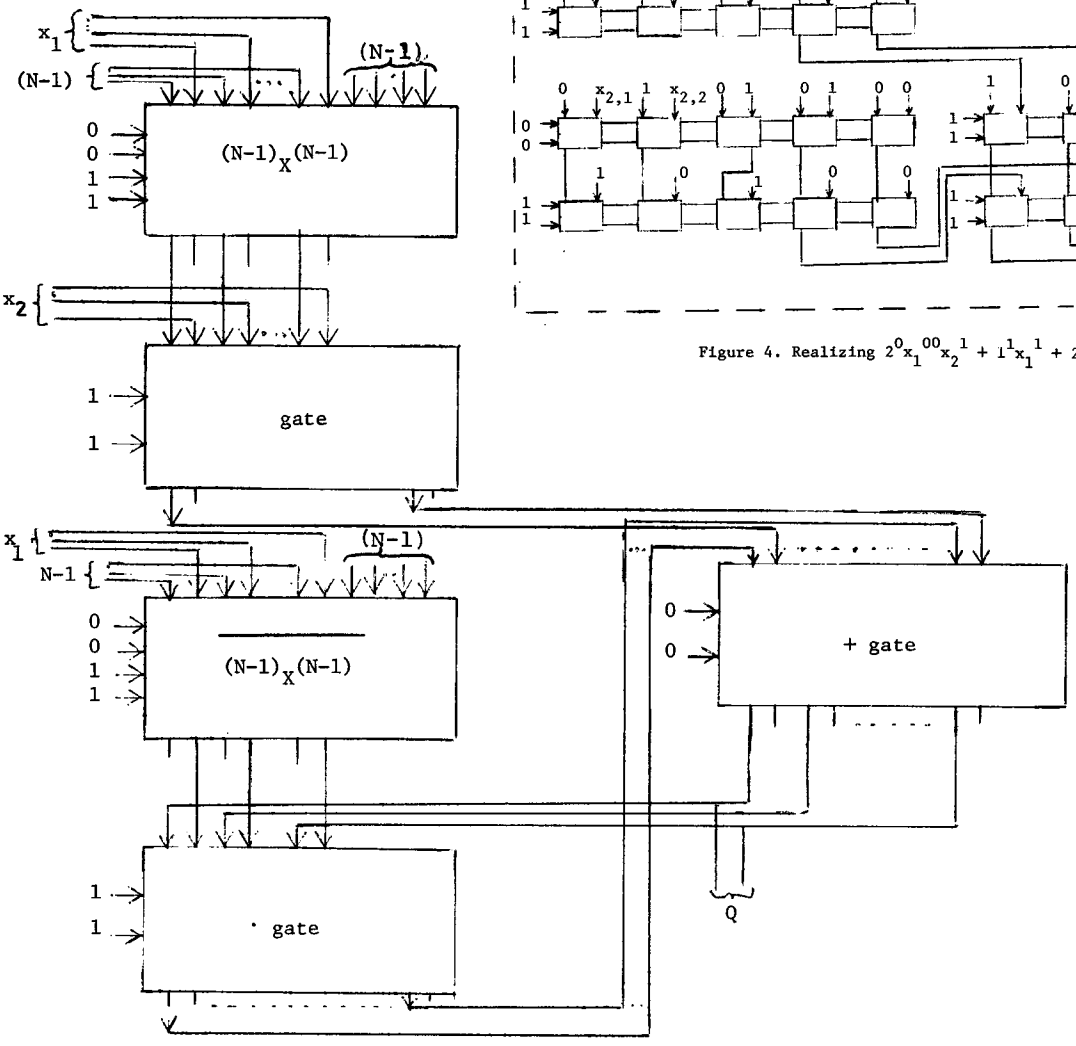


Figure 5. Realizing N-valued storage element