Multiple-Valued Combinational Circuits with Feedback*

Jon T. Butler† and Tsutomu Sasao‡

†Dept. of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121

‡Dept. of Electronics and Computer Science
Kyushu Institute of Technology
Iizuka 820, JAPAN

Abstract

We consider the use of feedback loops in the realization of multiple-valued combinational circuits. We show that the number of purely combinational configurations in an r-valued system is 1/r of the total number. Thus, as the radix increases, the fraction of combinational configurations decreases. We also show that, for every radix value r, there is a circuit with feedback realizing a combinational logic function that has fewer gates than any feedback-free circuit.

1: Introduction

Traditionally, researchers consider only combinational circuits without feedback loops. Indeed, acyclic combinational circuits have been so thoroughly studied compared to cyclic combinational circuits, that the term "acyclic" is, in many people's minds, a synonym of "combinational". We know of only four papers [4, 5, 6, and 8] that consider feedback in combinational logic circuits. To our knowledge, there have been no studies of *multiple-valued* combinational circuits with feedback.

Huffman [4] shows that any binary combinational logic function can be realized with just one inverter if feedback is allowed. He shows further that oscillation is needed; that is, it is necessary that some logic line toggle back and forth between 0 and 1. This is shown to be necessary only for the one inverter case. If one allows two inverters, then oscillation is not necessary.

Aoki [Fig. 7, 1] shows a set-valued implementation of a 4-bit ripple-carry adder that has feedback. Here, four binary full adders are multiplexed into one 16-valued adder. A feedback loop results because the carry output of a lower order full adder drives the carry input of the next higher order full adder.

2: Examples of Feedback in Binary Combinational Circuits

The significance of feedback in combinational logic circuits can be seen by the 1's complement adder circuit shown in Fig. 1. In the 1's complement system, negating a number $A = a_{n-1} a_{n-2} \cdots a_0$ consists of replacing each a_i by the "complement", $\overline{a_i}$. The carry-out/carry-in path throughout the circuit represents the feedback path of this circuit. If the feedback path is broken at X, then the output A can depend on input a in one of four ways; $A = a$, $A = \bar{a}$, $A = 0$, or $A = 1$. It can be seen that all ways are possible except $A = \bar{a}$. $A = a$ occurs only when all pairs (a_i, b_i) have the property $a_i = \overline{b_i}$. For this case, all full adders have the property that the carry out is identical to the carry in. In this case, the carries are all 0 or all 1, and simultaneously, the sum bits are all 0 or all 1, respectively. Although the sum takes on two values for this special case only, both values represent the same result, a 0 sum. Thus, although the circuit is sequential, our interpretation of $00 \cdots 0$ and $11 \cdots 1$ as 0, makes it reasonable to think of it as "combinational".

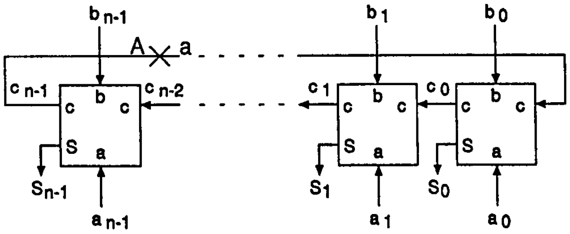


Figure 1. 1's complement adder circuit

Indeed, if one wants a true combinational circuit,

*Research supported by the Naval Research Laboratory, Washington, DC through direct funds at the Naval Postgraduate School, Monterey, CA.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **MAY 1994**	2. REPORT TYPE	3. DATES COVERED

4. TITLE AND SUBTITLE **Multiple-valued Combinational Circuits with Feedback**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Department of Electrical and Computer Engineering,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

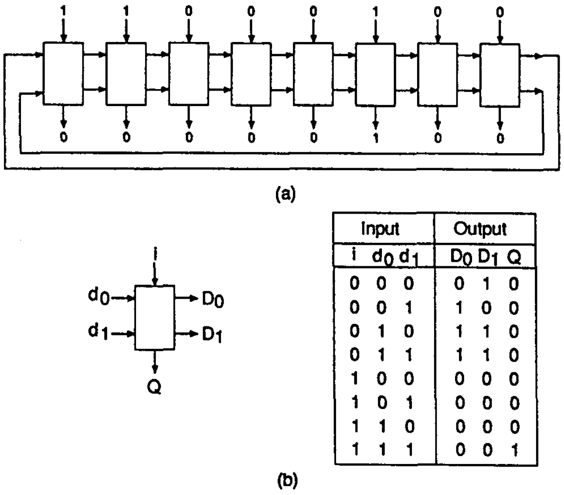
We consider the use of feedback loops in the realization of multiple-valued combinational circuits. We show that the number of purely combinational configurations in an r-valued system is llr of the total number. Thus as the radix increases, the fraction of combinational configurations decreases. We also show that, for every radix value r , there is a circuit with feedback realizing a combinational logic function that has fewer gates than any feedback-free circuit.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**6**	

then it is possible to add logic that has as inputs the sum S shown in Fig. 1, and produces, at its output S' the sum of the circuit of Fig. 1 except when S is all 1's, in which case, S' will be all 0's. The resulting circuit has feedback, but it is combinational. The additional circuit simply masks the sequential behavior of the underlying circuit. It is worth noting, however, that the additional circuit is unnecessary; our interpretation of all 0's as being equivalent to all 1's makes it so.

Our second example illustrates a circuit with feedback that does not produce sequential behavior for any input combination. Shown in Fig. 2a below is a sequence detector for a circular queue of a single binary string. That is, the input is viewed as a circular sequence of bits and the output is viewed in a similar way. A 1 on the output signifies the existence of some four bit sequence. Fig. 2a shows a circuit that recognizes the sequence 0001; that is, a 1 exists on an output if and only if the sequence 0001 precedes it.



Input			Output		
i	d_0	d_1	D_0	D_1	Q
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	1

Figure 2. Circular sequence detector

Fig. 2b shows the function of the individual units. The pair of lines d_1d_0 interconnecting the units denotes how many of the preceding bits are correct, with 00 indicating no bits correct, 01 one correct bit (0), 10 two correct bits (00), and 11 three correct bits (000).

The analysis of this circuit is straightforward. To determine the outputs, simply start at a unit that has a 1 as input. For this unit, the output $d_1d_0 = 00$, and this determines the output for all of the following units. If all inputs are 0, then assume any value for d_1d_0 of some arbitrary unit, and proceed in the same way. All d_1d_0 evaluate to 10.

In this case, the feedback path is such that if it is broken at any point, the output will not depend on the input. It will be 0 or 1 depending on the inputs applied.

3: The Feedback Path for Multiple-Valued Circuits

Before analyzing the multiple-valued case, we consider a binary example. Fig. 3 below shows a binary circuit with a single feedback loop.

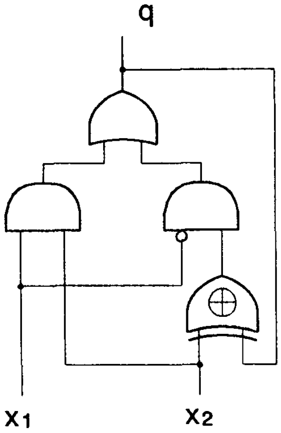


Figure 3. Example of a binary circuit with feedback

Depending on x_1 and x_2, the feedback loop will have one of four configurations, as shown in Fig. 4. Specifically, if $x_1x_2 = 00$, the feedback loop circulates the same logic value and the present value of q is its former value. If $x_1x_2 = 01$, q oscillates between 0 and 1. If $x_1x_2 = 10$ or 11, q is a constant 0 or 1, respectively, regardless of its previous value. For the first two assignments, the circuit is sequential and, for the last two, it is combinational. We say a configuration is *combinational* if its output is eventually a constant logic value, regardless of its initial logic value. Otherwise, it is *sequential.* Note that this definition applies to circuits with more than two logic values.

The directed graphs in Fig. 4 show a compact representation of circuit behavior. Here, each logic value is represented by a node from which a single directed arc emanates. The arc represents the next logic value (head), given the present logic value (tail). For example, the feedback loop configuration corresponding to a constant 1 has each arc head directed to the logic 1. Thus, if q' should ever (momentarily) be 0, it will be 1 shortly thereafter, and if it is 1, it shall remain 1.

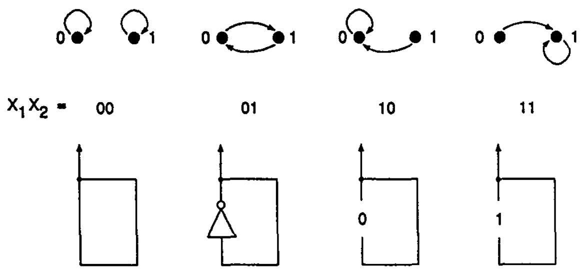


Figure 4. The four possible configurations for a feedback loop in a binary circuit

The case for multiple-valued circuits is more complicated. The four feedback loop configurations in Fig. 4 correspond to four functions on one or zero variables. Consider the circuit in Fig 5(a) having a single feedback loop. For each assignment of the inputs, x_1, x_2, ..., x_n, the circuit is reduced to Fig. 5(b), where h denotes a single input r-valued function. This is a multiple-valued extension of Fig. 4, in which the number of configuration grows with r^r. For $r = 3$ and $r = 4$, the number of feedback configuration is 27 and 256, respectively. Fig. 6 shows the directed graph representations of all configurations for $r = 3$. Any graph that contains a cycle with two or more nodes corresponds to oscillation. Since every node has an edge that exits it, there is at least one cycle in any directed graph corresponding to a cycle configuration. We have the following.

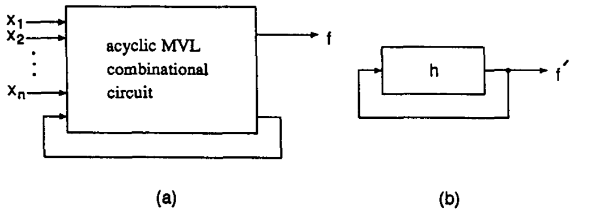


Figure 5. A feedback loop in a multiple-valued circuit

Lemma 1: Let G be the directed graph of a circuit associated with some assignment of values to variables. If G has at most one cycle and it contains one node, the configuration is combinational for that assignment.

The fact that the cycle has only one node guarantees there will be no cycling of logic values. Because there is no more than one such cycle, there can be no storage of values. In Fig. 6, the combinational configurations are shown with asterisks.

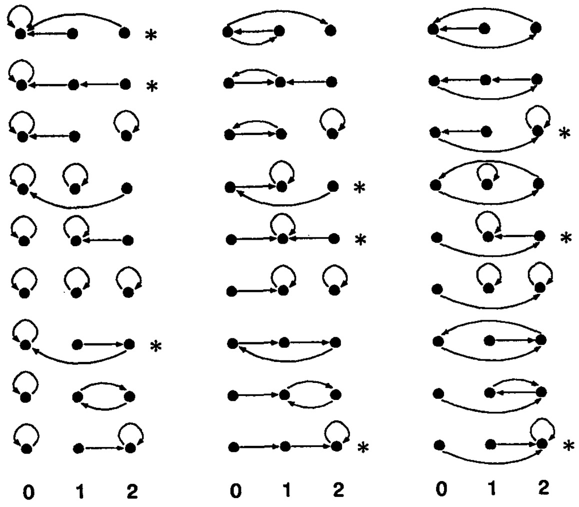


Figure 6. Directed graph representations of the cycle configuration for $r = 3$

Note that the directed graphs described in Lemma 1 have a unique node η, the one involved in the one-node cycle, while all other nodes are connected to η by a simple path. On the contrary, if there is no path from some node β to η, then a path from β is in a cycle or ends on a cycle not containing η, implying the existence of at least two cycles. This latter observation allows us to count the number of directed graphs that correspond to combinational configurations.

A graph that satisfies Lemma 1 has a single node at the root, (i.e., $n_1 = 1$), n_2 nodes with arcs to the root node, n_3 nodes with arcs to nodes with arcs to the root node, etc.. There are $n!/(n_1!n_2! \cdots n_m!)$ ways to distribute labels to the groups of nodes. Note that, between the ith and i+1th level, there are n_i arcs that can be chosen from $n_i^{n_{i+1}}$ possible arcs. Therefore, for each distribution of labels, there are $n_1^{n_2}\, n_2^{n_3}\, n_3^{n_4} \cdots n_{m-1}^{n_m}$ ways for arcs to occur between all levels. Thus, the total number $N(r)$ of combinational configurations is

$$N(r) = \sum_{n_1+n_2+\cdots+n_m=r} \frac{r!}{n_1!n_2!\cdots n_m!} n_1^{n_2} n_2^{n_3} \cdots n_{m-1}^{n_m}, \quad (1)$$

where the sum ranges over all *ordered* partitions

$(n_1, n_2, \cdots n_m)$ of r, such that $n_1 = 1$. The number of ordered partitions can be counted as follows. Align the nodes associated with logic values in a row. Between the nodes are $r-1$ spaces, of which one is already determined (that associated with $n_1 = 1$). Of the $r-2$ remaining spaces, place $m-2$ dividing lines, which are in addition to the one associated with $n_1 = 1$. There are $\begin{bmatrix} r-2 \\ m-2 \end{bmatrix}$ ways to do this. Thus, for some specified m, there are $\begin{bmatrix} r-2 \\ m-2 \end{bmatrix}$ ways to form $n_1 + n_2 + \cdots + n_m = r$ subject to $n_1 = 1$. The value of m ranges from 2 to r.

While this interpretation is readily understood, a more complex one leads to a simpler form for (1). We illustrate by an example. Consider the configuration shown in Fig. 7. We seek a representation for this that is a list of arcs. There will be r arcs, one for each of the r nodes from which one arc emanates. Order all leaf nodes in ascending order and write (in the same order) the nodes to which the arc from each leaf node goes. For the example in Fig. 6, we get 1 8 0 8, which corresponds to the arcs leaving leaf nodes 3 4 6 7. Remove the leaf nodes from the tree and repeat the process. In this case, we get 0 5, corresponding to arcs leaving 1 8. Repeat the process again. In this case, we get 2 corresponding to the arc leaving 0. Repeat again. In this case, we get 5 corresponding to the arc leaving 2. Listing this all out gives 5 5 2 0 5 1 8 0 8, where the initial 5 is in the cycle of the root node. Note that this representation is unique; a different tree will yield a different ordered tuple.

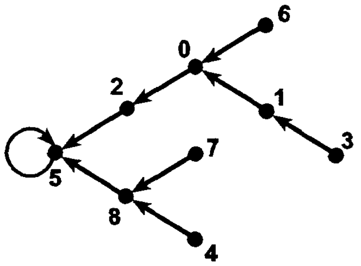


Figure 7. An example configuration

The converse operation, that of producing the tree from ordered tuple, is straightforward. Consider the example above, 5 5 2 0 5 1 8 0 8. The leaf nodes are uniquely those nodes with no incoming arcs; i.e. those nodes not in the above list. These nodes are 3 4 6 7. With the nodes listed in ascending order, we can conclude that an arc goes from 3 to 1, from 4 to 8, from 6 to 0, and from 7 to 8. With these last four nodes removed from the tree, we can repeat the process to obtain the resulting leaf nodes, etc. and the nodes for which they provide arcs. This is unique; a different ordered tuple will yield a different tree.

It follows then that the number of configurations is simply the number of tuples. That is, with r-valued systems, there are r^{r-1} ways to choose r-tuples, since each element in the r-tuple can be chosen without restriction, except the first, which must be chosen the same as the second. Thus,

$$N(r) = \sum_{n_1+n_2+\cdots+n_m = r} \frac{r!}{n_1!n_2! \cdots n_m!} n_1^{n_2} n_2^{n_3} \cdots n_{m-1}^{n_m} = r^{r-1}.$$

The significance of this result is that, as the radix r increases, the fraction of configurations that are combinational decreases. Specifically,

Lemma 2: The fraction of configurations in r-valued logic circuits that are combinational is

$$\frac{1}{r},$$

while the remaining are sequential.

If there are n r-valued inputs, there are r^n assignments of values. If we assume all configurations are equally likely, with only $1/r$-th of the configurations combinational, the chance that one of the r^n assignments produces a noncombinational configuration is close to 1 for all practical values of r and n. Indeed, this probability is

$$1 - \left[\frac{1}{r}\right]^{r^n}.$$

4: The Necessity of Feedback in Minimal Multiple-Valued Circuits

The main result of this section is an extension of a result by Rivest [8] for binary circuits. Specifically, it shows that feedback paths are necessary for the realization of *minimal* combinational multiple-valued circuits, where a minimal circuit is one requiring the fewest gates.

Theorem 1: For any radix $r \geq 2$, there exists a combinational logic circuit with feedback that requires fewer gates than any circuit realizing the same function without feedback.

Proof: It is sufficient to consider one set of sufficient operators. In particular, consider an r-valued Allen-Givone algebra [2] consisting of three operators, max of two r-valued variables, min of two

r-valued variables, and the window literal of one r-valued variable. Consider the circuit C_{cyclic} shown in Fig. 8, which is made up of the first two operators, where + represents max and • (or nothing) represents min. The circuit has three inputs, x_1, x_2, and x_3. It also has six outputs, whose expressions are given in the figure. We show the validity of function f_3; the others are similar. Consider an assignment $\alpha = \alpha_1 \alpha_2 \alpha_3$ of values to the variables $x_1 x_2\ x_3$. There are three cases. If $\alpha_3 \leq \alpha_2$, then $f_3 = \alpha_3$. If $\alpha_3 \geq \alpha_1 \geq \alpha_2$, then $f_3 = \alpha_1$. If $\alpha_3 \geq \alpha_2 \geq \alpha_1$, then $f_3 = \alpha_2$. This includes all assignments of values and corresponds to $f_3 = x_3(x_1 + x_2)$.

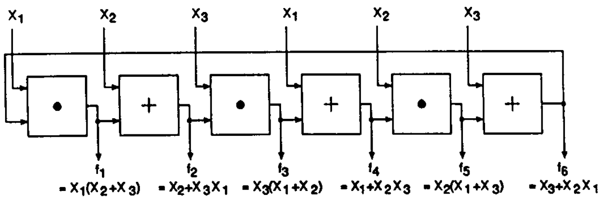


Figure 8. A circuit whose minimal form requires feedback

This circuit uses six gates. Consider a circuit $C_{acyclic}$ without feedback that realizes the same function. $C_{acyclic}$ requires at least six gates, one for each output. Since $C_{acyclic}$ is feedback-free, it has at least one output f_i whose circuit does not have, as input, an output f_j. With three inputs, the circuit producing f_i requires at least one other gate besides the output gate. Thus, $C_{acyclic}$ requires at least seven gates.

Q.E.D.

The reduction in the number of gates has been achieved by a sharing of gates among outputs. That is, among the six functions, there are twelve operations, six min and six max. With six gates in the circuit, each gate is, in effect, used twice.

Note that there is no assignment of variables for which there is a sequential configuration. This follows from the fact that any output is always equal to one of the input values. For an example, see the description of f_3 in the proof.

Rivest [8] poses the question of whether there are *single output* circuits that require feedback for their minimal realization. This is further discussed in Muroga's [7] text. As far as is known, this question has not been settled. Our next result shows, however, that for a certain single output function, there exists a realization with feedback that has the same number of gates as a minimal realization without feedback.

Consider the use of two-input NAND gates in the realization of binary functions. It is known that all binary functions can be realized exclusively with two-input NAND gates. Shown in Fig. 9a is the minimal realization of the two-input exclusive NOR function constructed from five two-input NAND gates. From Hellerman [7], we can conclude that this is minimal. Shown in Fig. 9b is a circuit with feedback realizing the same function also using five two-input NAND gates.

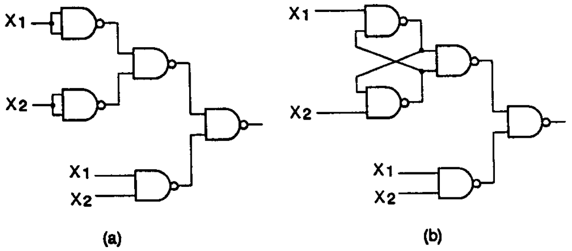


Figure 9. Minimal NAND gate realizations of the exclusive-NOR function

5: Concluding Remarks

The use of feedback in multiple-valued circuits has some interesting differences over binary circuits. First, we showed that, while two of the four configurations for feedback in binary are combinational, the fraction of combinational configurations is $1/r$, and, so, as the radix, r, increases, we find that a smaller fraction are combinational.

Our second result shows that, for any radix, there are circuits with feedback, whose minimal realization requires fewer gates than any realization without feedback.

Finally, we showed that for *single* output circuits, there exists a circuit with feedback that has the same number of gates as any feedback-free realization of that circuit. This represents a partial answer to Rivest's [8] question as to whether feedback is necessary in minimal single output functions.

References

[1] A. Aoki, "Dreams for new-device-based superchips - from transistors to enzymes," *Proc. of the Inter. Symp. on Multiple-Valued Logic*, May 1993, pp. 140-149.

[2] C. M. Allen and D. D. Givone, "A minimization technique for multiple-valued logic systems," *IEEE Trans. on Computers*, Feb. 1968, pp. 182-184.

[3] L. Hellerman, "A catalog of three-variable OR-invert and AND-invert logical circuits," *IEEE Trans. on Computers*, June 1971, pp. 198-223.

[4] D. A. Huffman, "Combinational circuits with feedback," in *Recent Developments in Switching Theory,* A. Mukhopadhyay. Academic Press, New York, 1971.

[5] W. H. Kautz, "The necessity of closed loops in minimal combinational circuits," *IEEE Trans. on Computers*, Feb. 1970, pp. 162-164.

[6] S. Malik, "Analysis of cyclic combinational circuits," *Proc. of the Inter. Conf. on CAD-93*, Nov. 1993.

[7] S. Muroga, *Logic design and switching theory*, John Wiley & Sons, New York, London, Sydney, and Toronto, 1979.

[8] R. L. Rivest, "The necessity of feedback in minimal monotone combinational circuits," *IEEE Trans. on Computers*, June 1977, pp. 606-607.