MULTIPLE-VALUED COMBINATIONAL CIRCUITS SYNTHESISED USING EVOLVABLE HARDWARE APPROACH

T. KALGANOVA[1], J. F. MILLER[1], N. LIPNITSKAYA[2]

[1]Dept. of Computer Studies, Napier University, 219 Colinton Road, Edinburgh, EH14 1DJ, UK
(t.kalganova, j.miller)@dcs.napier.ac.uk
[2]Belarussian State University of Informatics and Radioelectronics, P. Brovky 6, 220600, Minsk, Belarus

ABSTRACT In this paper a gate-level evolvable hardware technique for designing multiple-valued (MV) circuits, which is easily adapted for the different types of MV gates associated with operations corresponding to different algebra types and can include other more complex logical expressions (e.g.T-gate) is proposed. The technique is based on evolving the functionality and connectivity of a rectangular array of logic cells. The evolved 3-valued 1-digit adder with carry circuit is examined as an example. The issue of choosing the optimal set of MV gates used to evolve circuit is also discussed.

KEY WORDS: MV, evolvable hardware, logic design, combinational MV circuits

1. INTRODUCTION

Evolvable Hardware extends the concepts of Genetic Algorithms to the evolution of electronic circuits. A central idea of this is that each possible electronic circuit can be represented as a chromosome in an evolutionary process in which the standard genetic operations over the circuits are carried out. The evolving circuits may be evaluated using software simulation models [7, 9, 5] and others have evolved circuits entirely in hardware [2, 8].

This paper presents a method for the synthesis of combinational MV circuits. This approach is an extension of evolvable hardware method for binary logic circuits proposed in [1, 7]. The first attempts to evolve MV arithmetical combinational circuits had been discussed in [4]. This paper presents more detailed results and further discussion about circuit structures of MV circuits evolved by the proposed method. In an experimental investigation various sets of MV gates for circuit evolution (100% functionality) it is shown that one particular set is at least in three times more effective then any other. A number of evolved circuit designs for an one-digit 3-valued adder with carry are given. These designs are evolved using functionally complete bases as well as the other sets of MV gates. This paper shows how it is possible to combine the different functionally complete groups of MV gates in circuit design.

2. THE EVOLVABLE HARDWARE METHOD

The function representation of basic building blocks is summarized in Figure 1, where NOR operator is a complement of logic level x. In this paper we have restricted all logic gates to 2 inputs although in general they can have more than two inputs. Implementation of these gates can be found in [3]. We also use the 3-valued T-gate as one of the basic component for the design.

The method proposed here is based on evolving combinational MV networks employing a rectangular array of the logic cells. The logic cells in this array are uncommitted and can be removed from the network if they prove to be redundant. The inputs that are made available are logic constants 0, 1,…,r-1, all primary inputs $x_1, x_2, \ldots, x_n$ and the unary operators acting on the primary inputs, for instance, complement of all primary inputs $\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}$.

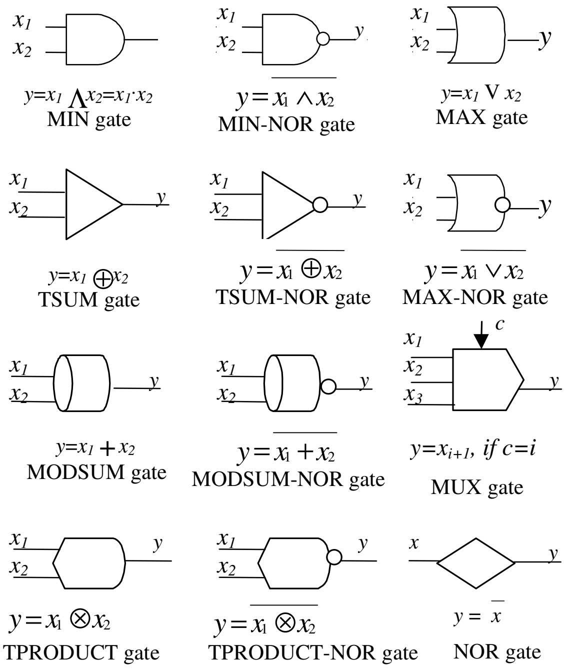

Figure 1. Symbols and analytic representation of two-input MV gates

Table 1. Cell gate functionality according to negative gene value in chromosome

Gene	*Gate function*
-1	$MAX\,(x_1, x_2) = x_1 \vee x_2$;
-2	$\overline{MAX(x_1,x_2)} = \overline{x_1 \vee x_2}$;
-3	$TSUM\,(x_1, x_2)=MIN(x_1+x_2,\ r-1)$;
-4	$\overline{TSUM\,(x_1,x_2)} = \overline{x_1 \oplus x_2}$;
-5	$TPRODUCT\,(x_1, x_2)=MAX(x_1+x_2-(r-1),\ 0)$;
-6	$\overline{TPRODUCT(x_1,x_2)} = \overline{x_1 \otimes x_2}$;
-7	$MIN\,(x_1, x_2) = x_1 \wedge x_2$;
-8	$\overline{MIN(x_1,x_2)} = \overline{x_1 \wedge x_2}$;
-9	MODSUM $(x_1, x_2) = x_1+x_2\ (mod\ r)$;
-10	$\overline{MODSUM(x_1,x_2)} = \overline{x_1 + x_2}$;
-11	MODPRODUCT $(x_1, x_2) = x_1 \cdot x_2\ (mod\ r)$;
-12	$\overline{MODPRODUCT(x_1,x_2)} = \overline{x_1 \cdot x_2}$;
-13	$\overline{x}$

Each cell in the array has a specific integer number, which is used in the network representation by chromosome. This cell is described by (n+1) integer numbers, where n is the number of inputs in the MV cell. The first n numbers describe the cell inputs and the last number defines the functional type of the cell. If this final integer is positive then the cell is assumed to be a T-gate, otherwise it refers to one of the gate types listed in Table 1. The user can choose any subset of this list of gates. For example, we can design combinational circuits using MV gates such as MIN, MAX, TSUM, MODSUM, complement of logic level x. In this case the encoded cell gate list will be given by the following:

A *chromosome* defines the connections in the network between the MV cells. Let us consider a network with two-input three-valued logic gates, or, alternatively a T-gate. A chromosome describes MV gates and m outputs of implemented function. Each MV gate is represented by triple of integer numbers $<c^1c^2c^3>$ which defines the connectivity between gates and the functional type of the gate. Note that often the chromosome contains cells that are not actually connected to any of the outputs. The process of removing these redundant cells is carried out for chromosomes with 100% functionality after the GA has completed.

3.PERFORMANCE OF GA DEPENDING ON THE SET OF MV GATES

Experimental runs of Genetic algorithm were carried out with the following parameters: mutation rate is 0.15, breeding rate is 1.00, tournament discriminator is 70%, population contains 15 chromosomes. Figure 2 shows how GA performance depends on the set of MV gates. It was found that of the gate sets used the best set was MIN-MODSUM-COMPLEMENT. This basis is startlingly better than some of the other bases. This set allows a five fold improvement in the GA performance in comparison with the MIN-MAX-TSUM-TPRODUCT-MODSUM-MODPRODUCT-COMPLEMENT set and a ten fold improvement over the MIN-MAX-TSUM-TPRODUCT-MODSUM-MODPRODUCT set. Note that the attempts to evolve circuits using only MIN-MODSUM-TSUM gates or MAX-MIN-COMPLEMENT gates didn't give any 100% circuits. The vertical axis of Figure 2 shows the average percentage of MV circuits evolved with 100% functionality for 100 iterations of the GA.

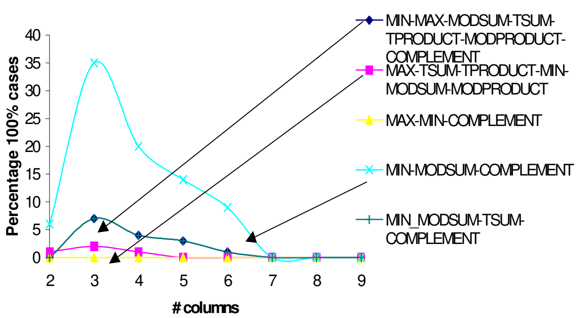

Figure 2. Dependence the GA performance on the set of MV gates

It is interesting to note that we expected to receive the highest percentage of 100% functionality cases for cases when we use all well-known MV gates, but the result contradicted out expectations as we find that a reduced set of MV gates produced the best results. Also experimental results show that the correct choose of MV gate set allows us significantly increase the GA performance without any changing in GA parameters. It is clear from these results that using one of the functionally complete bases does not one to guarantee receiving the 100% results with a small number of generations. Increasing the number of evolutionary steps must permit one to reach 100% functionality case for any functionally complete basis. Even in this case the percentage of evolved MV circuits with 100% functionality will be really very small so that we will have few examples from which we can choose efficient circuit.

4. EVOLVED CIRCUIT DESIGNS FOR 1-DIGIT 3-VALUED ADDER WITH CARRY

Now we present some of the arithmetical circuits that we have so far been able to evolve using the above method. All these circuits implement the 3-valued 1-digit adder with output carry. Some of the results are extremely interesting. The circuits have been evolved using 2-input 1-output MV gates such as MODSUM, MODPODUCT, MIN, MAX, TSUM and TPRODUCT with direct and inverted inputs. When one allowed completely free choice of gate and transformation of inputs it became easier to evolve 100% functionality immediately even though the search space had been greatly expanded.

Evolving a one-digit 3-valued adder using geometry of 3 rows and 3 columns proved to be relatively easy and the design shown in Figure 3 was obtained. This circuit has been evolved using the basic gates: MODSUM, TPRODUCT, NOR and TSUM with direct and inverted inputs and outputs.

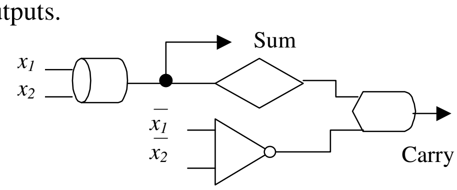

Figure 3. The evolved 1-digit 3-valued adder with carry

When a set of basic gates was changed to TSUM, MIN and MODSUM (with same geometry) the design shown in Figure 4 was evolved. The sum component in this circuit is the familiar sum-digit 3-valued circuit of conventional adders. Note that the GA has evolved sum and carry components in this design separately.

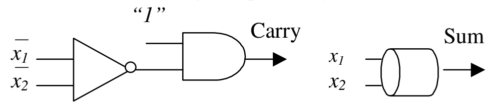

Figure 4. The one-digit 3-valued adder with carry evolved in TSUM-MIN-MODSUM gate basis.

Using only MIN-MAX-MODSUM allowed the circuit shown in Figure 5 to be evolved (with a 3 column , 2 row geometry). The sum component in this design is implemented in the optimum way: it uses only one MODSUM gate.

Attempts to reduce the number of different gates to a minimum gave the design shown in Figure 6 (2 columns by 2 rows geometry). Note that with larger geometries one can still obtain this result after simplifying the resulting circuit.

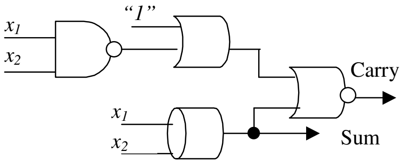

Figure 5. Evolved solution for the one-digit 3-valued adder using MIN-MAX-MODSUM-NOR gates.

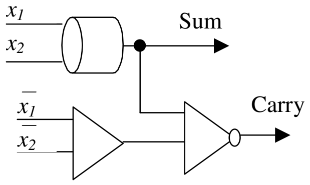

Figure 6. Using MODSUM-TSUM gates to evolve one-digit 3-valued adder design

Several experiments have been carried out which evolve MV combinational circuits using the single-control T-gate (Figure 7). The carry and sum components of these designs were evolved by GA separately.

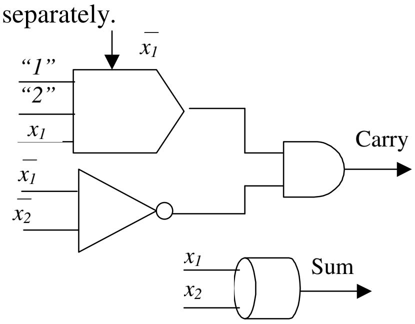

Figure 7. Evolved adder using T-gate

As we can see our chromosome representation allows us to use both 2-input 1-output MV gates and T-gate. It became easier to evolve 100% functionality immediately after having utilized this more complicated representation, despite the fact that the search space had been greatly expanded by this change. We feel that this chromosome representation allows more *routes* to the desired solution. In many occasions evolved chromosomes have achieved 100% functionality by using T-gate in place of MIN, MAX or constant gates.

When the geometry was constrained to 2x2 and the set of basis gates chosen contained only MODSUM, TSUM and MAX gates the optimum designs shown in Figure 8 was obtained. This was a gratifying result to obtain as it is clear that these designs are an optimum solution. These circuits were previously unknown.

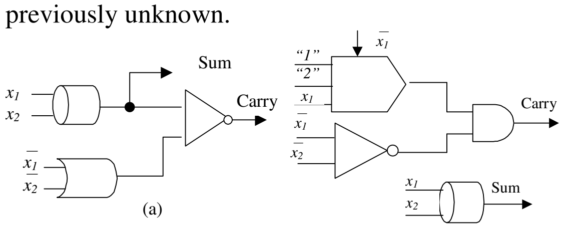

Figure 8. Two evolved optimum one-digit adder with carry

Evolving the one-digit 3-valued adder was easier to do with a larger geometry but resulted in a less efficient circuit. For example, the same results discussed above had been obtained with much larger geometries (3x4, 4x3, 4x4, 5x5, 4x6, 4x8, 4x10). Choosing too small a geometry ran the risk that no 100% solutions could be found because it was physically impossible to build the required functionality with few gates. While using too large a geometry simply gave the GA too many possibilities to work with and it struggled to find the fully functional solutions.

5. SUMMARY

In this paper it has been shown that by evolving a linear chromosome of cell functionalities and connectivities based on a rectangular array of logic cells it is possible to evolve both traditional and novel designs for arithmetical MV circuits. The method uses a GA to evolve both a netlist structure and functionality for MV cells. The fitness function tests the functionality of the circuit. This approach has allowed us for the first time allows us to synthesize combinational MV circuits in different functionally complete bases or any combinations of these. The method allows the synthesis of very novel circuit structures, which have never been seen before. We have evolved the 3-valued 2-digit adder with carry as an example. It was very interesting to find that the different sets of MV gates used when evolving the circuits radically affects the ease of implementation.

6.ACKNOWLEDGEMENT

The authors are very grateful to Prof. Vlad P. Shmerko for fruitful discussions and to prof. C. Moraga for his useful comments.

7. REFERENCES

1. Fogarty T. C., J. F. Miller, P. Thomson, Evolving Digital Logic Circuits on Xilinx 6000 Family FPGAs. *The 2nd Online Conference on Soft Computing*, 1997
2. Goeke M., Sipper M., Mange D., Stauffer S., Sanchez E., and Tomassini M., (1996) Online autonomous evolware, in *Proceedings of ICES96,* now published in *Lecture Notes in Computer Science Vol. 1259*, pp. 96-106, Springer-Verlag, Heidelberg, 1997.
3. Jain A.K., R.J. Bolton, M. H. Abd-El-Barr, CMOS Mutliple-Valued Logic Design - Part 2: Function Realization *IEEE Trans. on Circuits and Systems - I. Fundamental theory and applications.* Vol. 40. No 8. - 1993. - pp. 515-522.
4. Kalganova T., J. Miller, Evolutionary Approach to Design Multiple-valued Combinational Circuits. *Proc. of the* ACS'97. Szczecin, Poland. - 1997. - pp.333-339.
5. Koza J. R., *Genetic Programming*, The MIT Press, Cambridge, Massachusetts. 1992
6. Mariani R., R. Roncella, R. Saletti, P. Terrini A Useful Application of CMOS Ternary Logic to the Realization of Asynchronous Circuits. *Proc. of the 27th ISMVL.* - 1997. - pp.203-208.
7. Miller J. F., Thomson P., Fogarty T. C., Designing Electronic Circuits Using Evolutionary Algorithms. Arithmetic Circuits: A Case Study, chapter 6, in *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*: Recent Advancements and Industrial Applications. Wiley, 1997
8. Thompson A., An evolved Circuit, Intrisic in Silicon, Entwined with Physics. *Proc. of the ICES'96*, now published in *Lecture Notes in Computer Science*, Springler-Verlag, Heidelberg, 1996.
9. Zebulum R., M. Vellasco, M. Pacheco, Evolvable Hardware Systems: Taxonomy, Survey and Applications. *Proc. of the ICES'96*, Tsukuba, Japan, 1996.