CELL LIBRARY FOR AUTOMATIC SYNTHESIS OF ANALOG ERROR CONTROL DECODERS

Jie Dai, Chris J. Winstead, Chris J. Myers, Reid R. Harrison, Christian Schlegel

Department of Electrical and Computer Engineering
University of Utah, Salt Lake City, UT
{jack,myers}@vlsigroup.ece.utah.edu

ABSTRACT

This paper presents a cell library for automatic synthesis of analog error control decoders. By using some basic cells, analog error control decoders can be automatically synthesized. Also, using automatic synthesis based on this cell library, the circuit performance is not degraded and the circuit is smaller and lower power compared with corresponding canonical designs.

1. INTRODUCTION

It has recently been observed that a number of important algorithms in error-control coding can be interpreted as operations of the *sum-product algorithm* on *probability propagation networks* which are a kind of *factor graph* [1] [2] [3]. Researchers have also noticed that the sum-product algorithm on probability propagation networks may be well suited for analog VLSI [4] [5]. As a result, analog error control decoders have been built by using BiCMOS [6] [7] and recently by using CMOS [8]. Meanwhile, automatic synthesis for large analog circuits is still a problem. One of the main reasons for this lack of automation is that analog design in general is perceived as less systematic and more heuristic than digital design. In designing analog error control decoders, researchers have found that analog error control decoders have a regular structure and the entire circuit can be built using a few basic cells. As a result, a few basic cells can be built and then a tool can automate the construction of the analog error control decoder using these cells [7]. However, using their method, different decoders may need different basic cells. As a result, it is still not a general method that can be used for different analog error control decoders. This paper shows that all analog error control decoders can be built by using a small number of basic cells in a cell library, facilitating automatic synthesis. Morever, our analysis shows that analog error control decoders built using this method are comparable in performance, smaller, and lower power than the corresponding *canonical designs*.

Supported by NSF grant CCR-9971168.

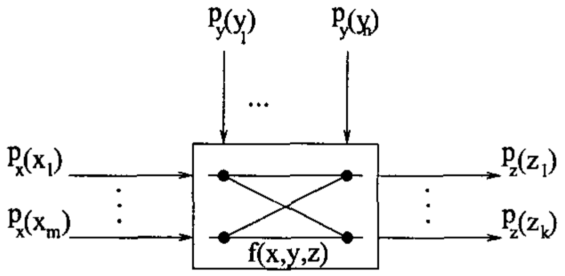

Fig. 1. Probability propagation network building block.

2. BASIC CELLS

Figure 1 is the canonical basic building block of the sum-product algorithm and Figure 2 is the corresponding circuit. The values of m, n, k are allowed to vary. There are also many possible functions $f(x, y, z)$, making the number of such building blocks nearly infinite. Suppose that each z term is the sum of $l = xy/z$ product terms, there are

$$\binom{l}{xy}\binom{l}{xy-l}\binom{l}{xy-2l}\cdots\binom{l}{l}$$

different combinations, each requires a unique cell. For example, if $m = n = k = 4$, and each z term is the sum of 4 xy product terms, there are 63,063,000 different cells. Of course, we must find some small basic cells so that the number of basic cells can be kept under control.

In Figure 2, the circuit below the wire network is used to generate product terms and the circuit above the wire network is used to do normalization so that the current of all the outputs added together equal I_u where I_u is the current designated as representation of probability 1. Breaking the circuit shown in Figure 2 into two cells results in a product cell and a normalization cell.

In communication, the signal used always belongs to a signal set which have signal number of 2's power such as

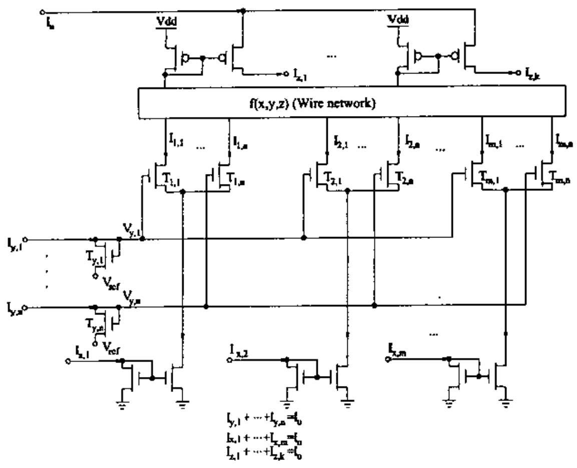
f(x,y,z) (Wire network)
Vdd
Vref
$I_{y,1}+\cdots+I_{y,n}=I_u$
$I_{x,1}+\cdots+I_{x,m}=I_u$
$I_{z,1}+\cdots+I_{z,k}=I_u$

Fig. 2. Typical cell used in canonical design.

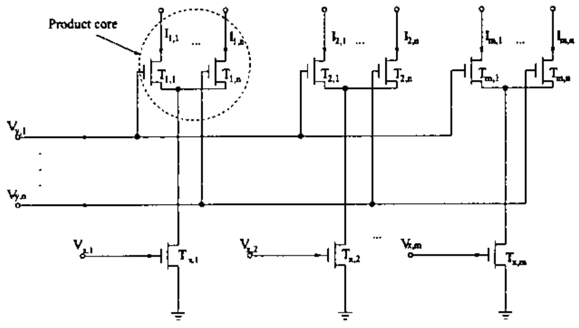

Fig. 3. Product cell.

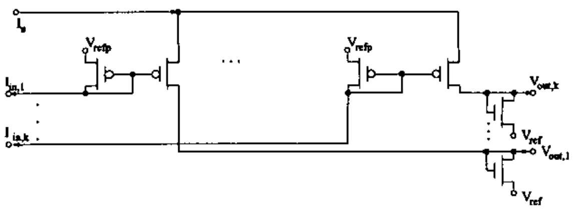

Fig. 4. Normalization cell.

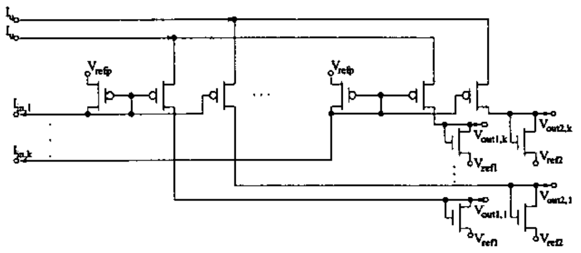

Fig. 5. Double normalization cell.

BPSK, QPSK, 8-PSK, 16-QAM. Thus, m, n, k in Figure 1 are all powers of 2. Now, it seems that we could build a limited number of basic cells. However, if the output of a normalization cell is needed to be distributed to the input of a lot of product cells, then the current output of the normalization cell needs to be duplicated. Because the number of product cells to which the normalization cell's output must be distributed varies, it is difficult for us to build a limited number of normalization cells. Therefore, it is better for us to move the diode connected transistors from the input of product cell to the output of normalization cell. As a result, the product cell accepts voltage input and provides current output where the normalization cell accepts current input and provides voltage output. Figure 3 shows a product-m-n cell and Figure 4 shows a normalize-k cell.

Note that V_{ref} in the normalization cell must be different based on whether its output is connected to the x input or y input of a product cell. As a result, the output of a normalization cell must only be connected to either x inputs or y inputs of product cells. In the case when the output of a normalization cell is needed to be connected to both the y inputs of product cells and x inputs of product cells, normalization cells which we call dnormalize-k cells shown in Figure 5 are needed.

From the above analysis, if $m \leq 2^r$, $n \leq 2^s$, and $k \leq 2^t$, then a cell library needs rs product-m-n cells, t normalize-k cells and t dnormalize-k cells. Our current cell library has 80 cells: product-2-2, ..., product-2-256, ..., product-256-256, normalize-2, normalize-4, ..., normalize-256, dnormalize-2, dnormalize-4, ..., dnormalize-256, and it enables construction of current analog error control decoders. This more than sufficient for current analog error control decoders in which typical values of m, n, k are not larger than 16. As a result, a cell library with 24 cells is enough for current analog error control decoders. Figure 6 and 7 are two examples of how to build larger blocks using these cells. The circuit shown in Figure 6 calculates the joint probabilities of two independent variables. The circuit shown in Figure 7 is the butterfly trellis (also called soft XOR). It performs the following operation and it is often used when Z is a parity check variable of X and Y.

$$\begin{bmatrix} pZ(0) \\ pZ(1) \end{bmatrix} = \begin{bmatrix} pX(0)pY(0) + pX(1)pY(1) \\ pX(0)pY(1) + pX(1)pY(0) \end{bmatrix}$$

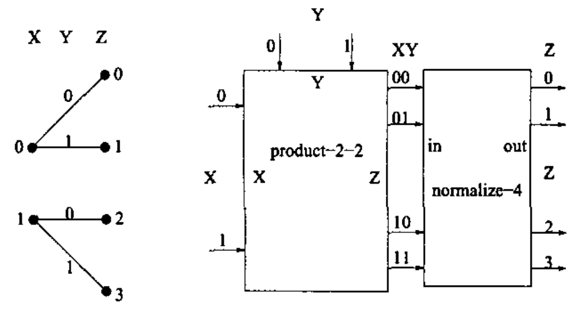

Fig. 6. Example 1.

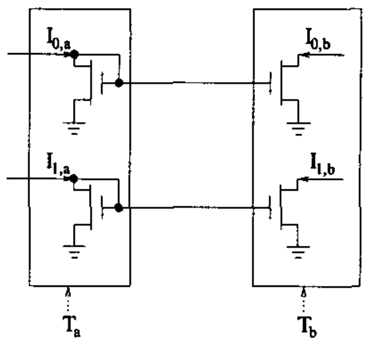

Fig. 8. Current mirrors

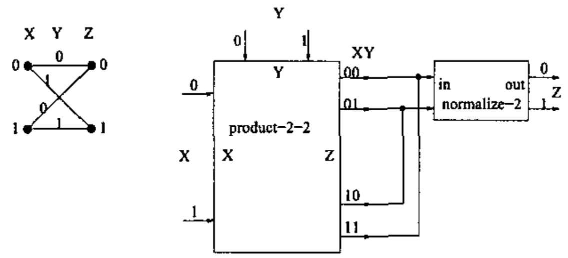

Fig. 7. Example 2.

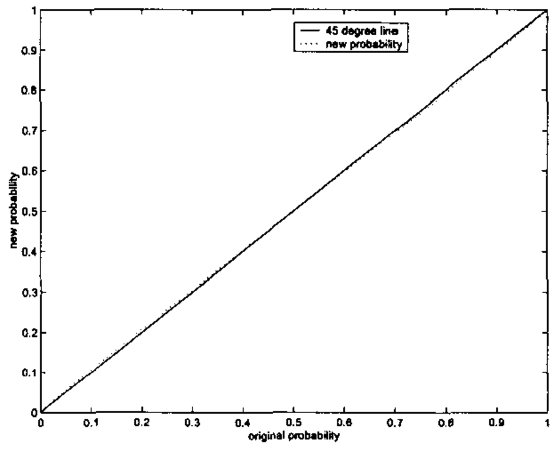

Fig. 9. Thermal effect for the current mirrors

3. THERMAL EFFECT

A potential problem of using these basic cells to build a circuit is cross-chip temperature difference. The voltage-current relation has dependence on temperature. As a result, using this approach, this thermal effect may be greater than the canonical design because the temperature between different blocks may be different. As a result, this thermal effect is analyzed by using the equations described in [9] and choosing typical values for $k_3 = 1.5$ and $k_4 = 2mV/K$. For the circuit shown in Figure 8, Matlab simulation shows that even if $I_{0,b}$ and $I_{1,b}$ are changed significantly compared with $I_{0,a}$ and $I_{1,a}$, the probability $p_{0,b} = I_{0,b}/(I_{0,b} + I_{1,b})$ does not change significantly compared with $p_{0,a} = I_{0,a}/(I_{0,a} + I_{1,a})$ because the currents are magnified by nearly the same scale. Figure 9 shows the relationship between $p_{0,b}$ and $p_{0,a}$ if $T_a = 300K, T_b = 310K$, and $I_{0,a} + I_{1,a} = 1nA$. It is clear that even $10K$ variation doesn't change the probability much. For analog error control decoders, the circuit blocks are uniformly distributed in the chip so that the power consumed is also nearly uniformly distributed. Also, for a CMOS circuit working under weak inversion, the consumed power is very small so that the temperature difference between different blocks should be likely too small to cause any problems.

4. DECREASING THE CIRCUIT COMPLEXITY

Note from Figure 3 that in the product core, there are always n transistors even if only k products are useful and the other $n-k$ products are discarded by shorting their output to Vdd. These dummy transistors are needed to make the following equation hold for the current shown in Figure 2.

$$I_{i,j} = I_{x,i} * I_{y,j}/(I_{y,1} + ... + I_{y,n}) = I_{x,i} * I_{y,j}/I_u$$

We can minimize the transistor count by substituting the $n-k$ transistors with 1 transistor, if the circuit shown in Figure 10 is used to generate the gate voltage of this transistor. Figure 10 contains $k+2$ transistors (1 PMOS transistor and $k+1$ NMOS transistors). As a result, we can replace the original $n-k$ transistors with $k+3$ transistors. In cases that n is much larger than k, using this technique can save a lot of transistors. The disadvantage of using this technique is that the circuit may be a little slow compared with

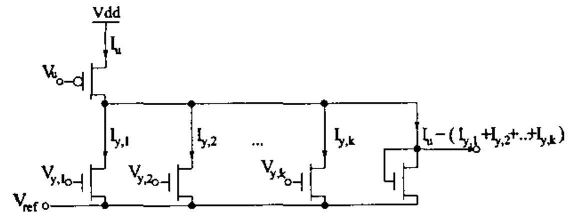

Fig. 10. Reference cell.

the original design. Also, notice that if this technique is used, then the gate voltages of the transistors in the product core-$V_{y,1}, V_{y,2}, ...V_{y,k}$ and the voltage corresponding to $I_u - (I_{y,1} + ... + I_{y,k})$ are provided by different cells, so the thermal effect may cause problems in this case.

In order to use this technique, we need to build another family of basic cells which we call reference cells. This kind of cell always contains $k+1$ NMOS transistors and 1 PMOS transistor. Because k is always a power of 2, building ref-2, ref-3, ref-5, ..., ref-129 are enough for most analog error control decoders.[1] We also need to build corresponding product cells product-2-3, product-2-5, ..., product-2-129.

5. COMPARISON WITH CANONICAL DESIGN

The circuit cells shown in Figure 2 are used in canonical design. Notice that the product cell and normalization cell are combined together, and the cell uses current input and current output. The advantages are that the thermal problem is reduced and the wires between the product cell and the normalization cell are short because the two cells are combined in one cell. However, the thermal problem is likely not a serious problem for our approach from the previous analysis. Also, the wire network does not significantly affect the performance. As a result, the circuit performance using our approach should still be comparable with the canonical design. We have verified the performance to be nearly the same with Spice simulation of the extended Hamming(8,4) decoder [8] using this approach and the canonical design.

For the circuit complexity, when the outputs of a cell shown in Figure 2 are needed to be given to several cell's input, then the output current needs to be duplicated by using more transistors. For the extended Hamming(8,4) decoder [8], the core of the decoder uses 292 transistors using this approach. If the canonical design shown in Figure 2 is used, then the core of the decoder requires 36 more transistors.

This approach is also lower power because current duplication is saved in some cases. For the extended Hamming(8,4) decoder[8], the power used by the core of the decoder using this approach is $32 * I_u * Vdd$ while the power used by the corresponding canonical design is $40 * I_u * Vdd$ because 8 current duplications are saved while each current duplication consumes $I_u * Vdd$.

[1]We use the number of NMOS transistors in the cell to discriminate different cells.

6. CONCLUSIONS AND FUTURE WORK

We have developed a cell library to support a general automatic synthesis method which is adaptable for different analog error control decoders. Using this library, the circuit performance is nearly the same and the circuit is smaller and lower power compared with the corresponding canonical designs. Currently, an automatic synthesis tool is under construction and we would like to use this tool and our new cell library to design some analog error control decoder chips and test them in the future.

7. REFERENCES

[1] B. J. Frey, F. R. Kschischang, H. A. Loeliger, and N. Wiberg, "Factor graphs and algorithms," *Proc. 35th Allerton Conf. on Communications, Control, and Computing*, pp. 666–680, Sept. 1997.

[2] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, pp. 498–519, Feb. 2001.

[3] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inform. Theory*, pp. 325–343, Mar. 2000.

[4] N. Wiberg, H. A. Loeliger, and R. Kotter, "Codes and iterative decoding on general graphs," *European Transactions on Telecommunications*, pp. 513–525, Sept./Oct. 1995.

[5] J. Hagenauer and H. A. Loeliger, *The analog decoder*, Nov. 1997.

[6] M. Moerz, T. Gabara, R. Yan, and J. Hagenauer, "An analog .25um bicmos tailbiting map decoder," *IEEE Proc. International Solid-State Circuits Conference*, pp. 356–357, Feb. 2000.

[7] Felix Lustenberger, *On the Design of Analog VLSI Iterative Decoders*, Ph.D. thesis, Swiss Federal Institute of Technology, 2000.

[8] C. Winstead, J. Dai, W. J. Kim, S. Little, Y. B. Kim, C. Myers, and C. Schlegel, "Analog map decoder for (8,4) hamming code in subthreshold cmos," *Advanced Research in VLSI*, pp. 132–147, Mar. 2001.

[9] Yannis Tsividis, *Operation and modeling of the MOS transistor*, 1999.