

IDDT TEST CALIBRATION USING A PROGRAMMABLE PROCESSING ARRAY

Mikhail Itskovich, James Plusquellic

Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Cir., Baltimore, MD 21250
email: mitskov1@umbc.edu, plusquel@umbc.edu

ABSTRACT

This paper proposes an area efficient signal processing architecture to perform Iddt test calibration through vector multiplication. The design follows the Field Programmable Array organization, and capitalizes on the unique behavior of binary encoded signals to implement compact multiply elements. Vectors with 8 bit values were multiplied at a rate of 300kHz, independently of vector size.

1. INTRODUCTION

IC testing based on quiescent current (Iddq) [1] and transient current (Iddt) [2] runs into a challenge with smaller featured processes. [3] The magnitude and variation in functional leakage current, probe card and contact impedances can mask a significant portion of the measured signal. When the variability of these noise sources gets large, it becomes difficult to distinguish functional patterns from failing patterns. [4]

Variations in power grid impedance and probe contact impedance can be measured for each circuit under test (CUT). This data can be used to cancel out the variations in the contact impedance, and improve measurement quality. This calibration function is a linear transformation, and requires an efficient means of vector and matrix multiplication. [5]

In production testing, a single chip requires multiple measurements to be calibrated simultaneously. Iddt researchers used a DSP or a PC to postprocess the test data [3],[4], but such setup can not be integrated onto a probe card. Such setup is also ill equipped to process multiple data streams. A custom hardware signal processor is desirable, but it is difficult to fit enough multiplier circuits onto a single chip. [6]

This paper proposes a one-bit processing array architecture as a solution to the calibration requirements. A key feature of the processing array is the compact multiply-and-add function that can be implemented inside each cell. Sections 2 and 3 describe the processing array architecture

and the principles of one-bit operation. Section 4 shows the application of the array elements to vector multiplication. Section 5 describes the hardware implementation of the array. Finally, section 6 summarizes the information on the processing array and the calibration circuit.

2. PRINCIPLES OF IDDT CALIBRATION

Calibrating the measurements for every CUT has been shown to greatly improve the measurement resolution by cancelling out the effects of probe card and supply parasitics. Data acquired during the calibration step generates a calibration matrix, X, which is then used to normalize the regular test data. [5]

Calibration data is acquired by forcing a defect under a supply tap and collecting current measurements from local nodes into a vector. The simulated defect is generated by a test circuit under the tap that creates a resistive power ground short. [5]

A set of 4 taps produces 4 vectors, and results in a 4x4 data matrix, TCI. The calibration matrix, X, is computed by a linear transformation in Equation 2, where RCI is a matrix of nominal probe card resistances. Nominal values are either simulated, or taken from characteristic chip measurements.

$$X = TCI^{-1} \times RCI \quad (1)$$

$$\begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} = \begin{bmatrix} t_{00} & t_{01} & t_{02} & t_{03} \\ t_{10} & t_{11} & t_{12} & t_{13} \\ t_{20} & t_{21} & t_{22} & t_{23} \\ t_{30} & t_{31} & t_{32} & t_{33} \end{bmatrix}^{-1} \times \begin{bmatrix} r_{00} & r_{01} & r_{02} & r_{03} \\ r_{10} & r_{11} & r_{12} & r_{13} \\ r_{20} & r_{21} & r_{22} & r_{23} \\ r_{30} & r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2)$$

Applying X to test data normalizes the measurements to what they would have been under the nominal, probe contract resistances, RCI. Equation 4 shows a calibration operation for a 4-vector measurement. Here, T_n is a vector of 4 Iddt test measurements, and C_n is the compensated output.

$$C_n = T_n \times X \quad (3)$$

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} t_0 & t_1 & t_2 & t_3 \end{bmatrix} \times \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \quad (4)$$

Larger vectors can provide better compensation, but the calibration becomes computationally intensive. This problem must be addressed by a capable parallel processing architecture.

3. PROCESSING ARCHITECTURE

The hardware implementation of the vector multiplication operations is based on a general purpose one-bit processing array. The array structure was originally designed for multi-channel digital filtering, but the architecture extends well to matrix operations that also require highly parallel processing resources.

The processing array is based on a dense multiply-capable Processing Unit, tiled in such a way as to allow the formation of complex functions for multiple data streams. The result is the structure in Fig. 1. The organization is similar to an FPGA, but the blocks are designed for multibit arithmetic.

Each I/O Register block provides a multibit interface for data storage and access by a digital processor. It also contains an IIR filter and dithering circuitry to decorrelate streams [2], and to interface of signals running at different clock rates.

Each cell in Fig. 1 is composed of 4 directional processing units, capable of a multiply-and-add operation. The four unit structure facilitates configuration by combining routing and function control.

Fig. 2 shows the block diagram for the processing unit. The processing unit is structured for the multiply-and-add operation. F1 and F2 perform signal selection and multiplication functions. F3 performs extended arithmetic functions as well as non-linear and feedback operations. A control loop is created by feeding the output back into F1 and F2, which enables extended functionality such as divide. A loop filter is added to reduce low order error during feedback operation.

Table 1 shows the common processing unit functions. Combining multiple functions can create complex behaviors, such as multiply-and-add or conditional arithmetic. Matrix multiplication requirements are satisfied with just the multiply-and-add and routing resources, but the extended programmable functions may benefit larger applications.

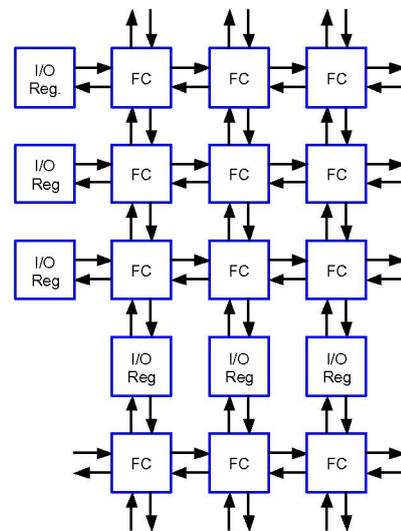


Fig. 1. Structure of the Programmable Processor Array

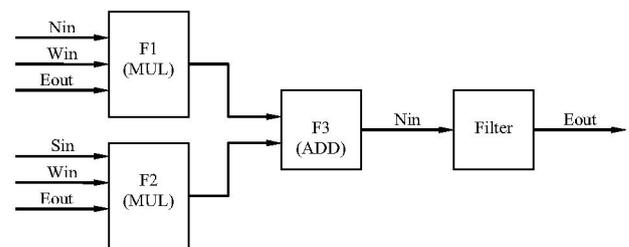


Fig. 2. Block diagram for a self-contained processing unit

Table 1. Processing Unit Functions

Function Type	Description
Fractional Arithmetic	ADD, SUB, AVE, MUL, DIV, MUL and ADD
Single Stream	SET, CLEAR, INVERT, MUL by 2, DIV by 2, SQR, SQRT
Window Logic	Any combinational logic function up to 2 inputs
Control Logic	Most logic functions up to 3 inputs

Using this architecture, a single processing unit can be implemented in 20 equivalent logic gates, and a register unit in 30 gates. Implemented in 0.13um, 512 processing units and 16 I/O units were placed inside 1mm² of the die.

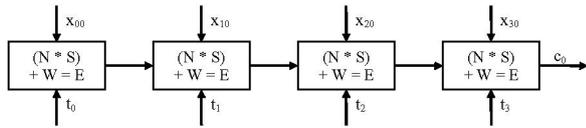


Fig. 3. Vector multiplication structure

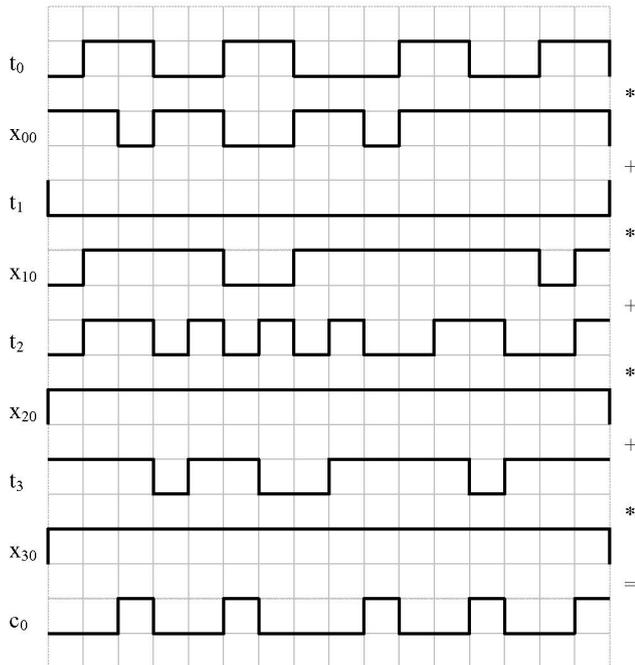


Fig. 4. Example vector multiplication data streams

4. VECTOR MULTIPLICATION

Vector multiplication using the processing unit is shown in Fig. 3. Because each processing unit can implement a multiply-and-add operation, N units are used to multiply an N -sized vector. In the example case, it takes 4 units. A full matrix multiplication requires N^2 units to complete.

Fig. 4 shows an excerpt of the one-bit stream operation, where the variables are designated on the left and operations on the right of the waveforms. To avoid overflow, an average function is used in place of add, resulting in:

$$c_0 = \frac{t_0 \times x_{00} + t_1 \times x_{10} + t_2 \times x_{20} + t_3 \times x_{30}}{4} \quad (5)$$

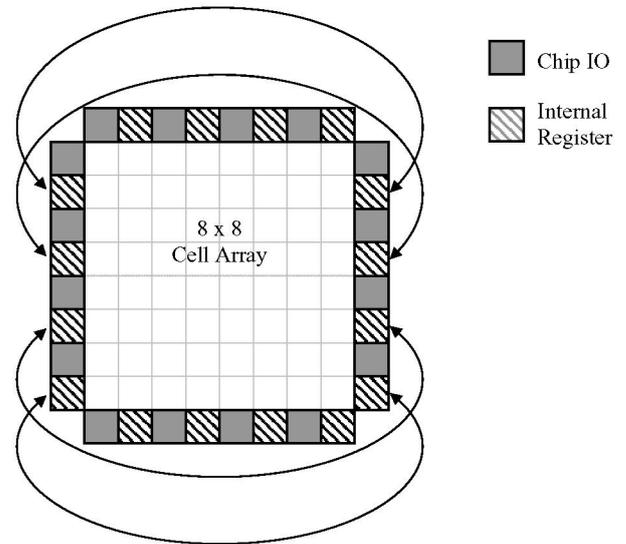


Fig. 5. Test array structure implemented in 0.13um CMOS

5. HARDWARE IMPLEMENTATION

The first processing array is implemented in 0.13um technology. The test structure is illustrated in Fig. 5, and consists of an 8x8 cell array, 64 I/O registers, and 32 IO ports. The structure has 512 available processing units.

The number of IO ports is pin limited. There are 16 dedicated input ports and 16 output ports. The remaining periphery registers are routed back into the opposite side of the array, as shown in Fig. 5, to increase routability and cell utilization in the test structure.

The system clock is limited to below 300MHz. Although the on-chip PLL and simulated circuits are capable of higher frequencies, the limit is exercised to ensure testability.

The array structure is configured by a serial data stream. The stream configures the processing cells and the I/O settings.

IO symmetry is preserved in the chip pinout. This allows multiple chips to be tiled if a larger array is required. This helps to create a larger function space and more usable IO ports. However, tiling is not a substitute for on-chip performance, because board routing is much slower than on-chip interconnect.

The processing array part of the testchip was implemented using a digital cell library and automated place and route. Automated placement and routing was performed for the expediency of the experiment, however steps were taken to mitigate area and performance losses.

Macro cells were created for the array cell and IO register circuits. Chip level placement used those macro cells within a constrained area of the chip. Since each cell has the same layout, the cell performance is similar and predictable.

Generous power ground capacitance and routing was placed in the chip to facilitate high speed operation. Clean power supply is also necessary for clean IO operation, because of the analog nature of dithering and multiple frequency interface.

Table 2. Design Summary

Technology	0.13um CMOS
Chip Dimensions	4mm X 4mm
Functional Capacity	64 cells
Cell Layout Area	25 um X 25 um
Target Clock Rate	300MHz
Target Resolution	8 bit
Target Bandwidth	4MHz
Simulated Bandwidth	100kHz
Single Function Latency	3 cycle

6. RESULTS

Limited hardware verification was performed. At this time, the extended functional information is still based on Spectre and Verilog simulations. Table 2 shows the design summary.

Vector multiplication simulated predictably as a combination of the digital functions. Fig. 4 demonstrates the behavior. To achieve the 8 bit resolution, the operation had to run for more than 1000 cycles, with the effective system bandwidth around 100kHz. This slowdown is both due to the digital filter requirements on the IO register and due to loss of significance effects for fractional multiplication.

The test remaining for the hardware is not so much the functional verification, but to measure the effectiveness of the filtering, dithering and routing. The analog and speed aspects determine the final bandwidth of the processing circuit.

7. CONCLUSION

One-bit processing architecture is a good candidate for a real-time matrix multiplication required to calibrate Iddt test measurements. This circuitry can be placed on the same die as the data acquisition devices measuring Iddt, to provide space efficiency on the probe card.

This study has pointed to some weaknesses in applying the original array architecture to matrix multiplication. Matrix multiplication relies on vector elements stored in memory. This memory comes from the I/O registers, and makes function routing inefficient. Although adding macro memory and constant storage functions to each unit will double the area, matrix multiplication will benefit from higher cell utilization and easier access to data.

Future design of the processing array should address general computing requirements of Iddt. Matrix operation is only one of arithmetic heavy operations needed for Iddt tests. The tests also benefit from on-chip filtering capabilities and cosine transformation. Since Iddt research is fairly new, there are new processing methods yet to be discovered. To be useful in the long term, the array architecture must remain flexible enough to accommodate future computing requirements without hardware change.

8. REFERENCES

- [1] R.Z.Makki, et al., "Transient power supply current testing of digital CMOS circuits," IEEE Proc. of International Test Conference, pp. 892-901, 1995
- [2] J.Plusquellic, D.M.Chiarulli, S.P.Levitan, "Digital Integrated Circuit Testing using Transient Signal Analysis," IEEE Proc. International Test Conference, pp. 481-490, 1996
- [3] A.Singh, J.Plusquellic, A.Gattiker, "Power Supply Transient Signal Analysis Under Real Process and Test Hardware Models," IEEE VLSI Test Symposium, pp. 357-362, 2002
- [4] D.Acharyya, J.Plusquellic, "Impedance Profile of a Commercial Power Grid and Test System," IEEE Proc. International Test Conference, pp. 709-718, 2003
- [5] D.Acharyya, J.Plusquellic, "Calibrating Power Supply Signal Measurements for Process and Probe Card Variations." 2007
- [6] A.J.G.Hey, "Supercomputing with Transputers - Past, Present and Future," Proc. of Int. Test Conf., pp. 129-137, 2003
- [7] J.C.Cand, G.C. Temes, "Oversampling Delta-Sigma Data Converters," (Piscataway, NJ, IEEE Press, 1992)