Multiple-Valued Caches for Power-Efficient Embedded Systems

Emre Özer[†], Resit Sendag[‡] and David Gregg[*]

[†] *ARM Ltd. Cambridge, UK*
[‡] *Department of Electrical and Computer Engineering, University of Rhode Island, USA*
[*] *Department of Computer Science, Trinity College Dublin, IRELAND*
E-mail: *emre.ozer@arm.com, sendag@ele.uri.edu and david.gregg@cs.tcd.ie*

Abstract

In this paper, we propose three novel cache models using Multiple-Valued Logic (MVL) paradigm to reduce the cache data storage area and cache energy consumption for embedded systems. Multiple-valued caches have significant potential for compact and power-efficient cache array design. The cache models differ from each other depending on whether they store tag and data in binary, radix-r or a mix of both. Our analytical study of cache silicon area shows that an embedded System-on-a-chip (SoC) equipped with a multiple-valued cache model can reduce the cache data storage area up to 6% regardless of cache parameters. Also, our experiments on several embedded benchmarks demonstrate that dynamic cache energy consumption can be reduced up to 62% in a multiple-valued instruction cache in an embedded SoC.

1. Introduction

Multiple-valued logic (MVL) circuits have been designed over the last 20 years as an alternative to binary circuits. The advantages of the MVL are the use of fewer operations, potentially fewer gates and the reduction in the number of interconnections. The disadvantages are that it has worse noise margin levels than the binary, and the circuit complexity may increase with multilevel signals.

Designers using the MVL paradigm have focused on ternary (i.e. radix 3) and quaternary (i.e. radix-4) number systems [3] [4] [5] [6] [7]. Ternary system consists of three digits, namely {0, 1, 2} and quaternary system consists of four digits {0, 1, 2, 3}. Also, balanced ternary number system {-1, 0, 1} is radix-3 number system, which has the property of representing both sign and unsigned numbers without any explicit sign bit. The negative numbers are naturally represented since the number system includes -1. Thus, the balanced ternary has an advantage over the ternary system because it does not have a sign bit representation: it needs one *trit* fewer than ternary. A digit is called *bit*, *trit* and *quatrit* for binary, ternary and quaternary number systems. The study in [1] reports that the most efficient base in computations in terms of cost and complexity is *e* (i.e. 2.7…) and 3 is the nearest integer to *e*. Quaternary or radix-4 is also drawn special attention due to the fact that its conversion from/to binary is simpler than the ternary/balanced ternary system. There have been also several attempts to design mixed binary and MVL circuits that require the use of binary-MVL encoder and MVL-to-binary decoder circuits [8] [9] [10].

We focus on architectural issues of caches that are designed in ternary, balanced ternary and quaternary number systems. Our target system is an embedded SoC platform with a processor, instruction and data caches, and our goal is to explore the potential of using multiple-valued caches in embedded systems for reducing cache sizes as well as dynamic cache energy consumption. Multiple-valued caches can be designed with a mix of binary and *radix-r* addressing and tag/data store. We believe that the design of multiple-valued caches can reduce the cache data storage area, and allows reduction in the number of cache digit line transitions. This may therefore reduce dynamic cache energy consumption. In commercial processors such as *Pentium Pro* [16], *Alpha 21264* [17] and *StrongARM SA-110* [18], caches alone consume 33%, 16% and 43% of the total chip power. The energy consumption is particularly critical for embedded processors such as *StrongARM SA-110* whose caches consume almost half of its chip power.

The organization of the paper is as follows: *Section 2* introduces the novel multiple-valued cache models. Next, *Section 3* analyzes cache storage area requirements followed by analytical dynamic cache energy consumption analysis in *Section 4*. Later, *Section 5* presents the experimental framework and the results of dynamic cache energy consumption. *Section 6* discusses the related work. Finally, *Section 7* concludes the paper with a discussion of the future work.

Proceedings of the 35th International Symposium on Multiple-Valued Logic (ISMVL'05)
0195-623X/05 $ 20.00 **IEEE**

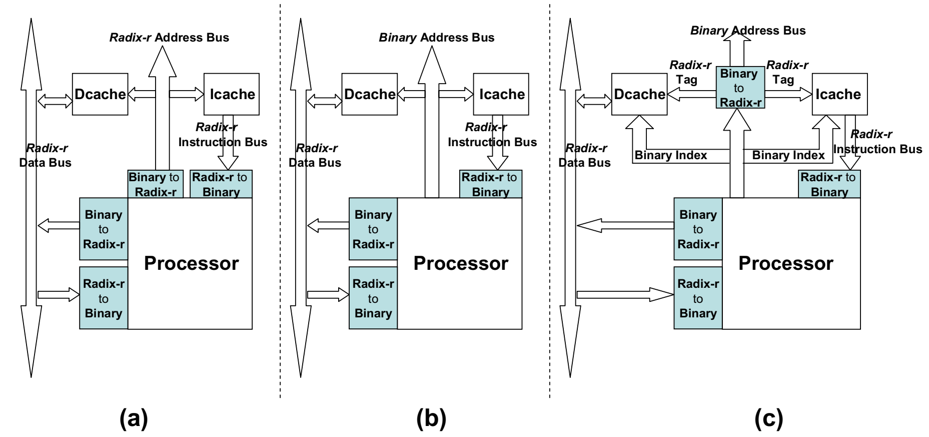


Figure 1 Three multiple-valued cache model in a SoC system, (a) RIRT, (b) BIBT, (c) BIRT *radix-r* caches

2. Multiple-valued Cache Models

We propose several multiple-valued caches and their advantages and disadvantages. Multiple-values caches can be addressed in binary or *radix-r* form, and can store tags in binary or in *radix-r*. Using these combinations, we can categorize three different multiple-valued caches: 1) *radix-r-indexed radix-r-tagged (RIRT)*, 2) *binary-indexed binary-tagged (BIBT)* and 3) *binary-indexed radix-r-tagged (BIRT)*. The common characteristic of all three cache models is that data in the data array cells are stored in *radix-r* format.

2.1. Radix-r-Indexed Radix-r-Tagged (RIRT) Radix-r Cache

This is the most extreme model of all three, which requires that all data and addresses leaving off the processor core are converted from binary to *radix-r* form. The SoC chip using this cache model is shown in **Figure 1a**. Here, the processor performs computations in binary but everything else around it works in *radix-r* number system. For this SoC system, there needs to be four conversion (i.e. encoder/decoder) circuits: two for bi-directional data bus, one for address bus and one for instruction bus.

Tag arrays in the fully binary caches are organized as content-addressable memories (CAM). Similarly, the tag arrays in the RIRT cache should be designed with *radix-r* CAM cells. Also, the tag comparison must be performed in *radix-r* form. This may further complicate the tag array design of this cache model in terms of area and power consumption.

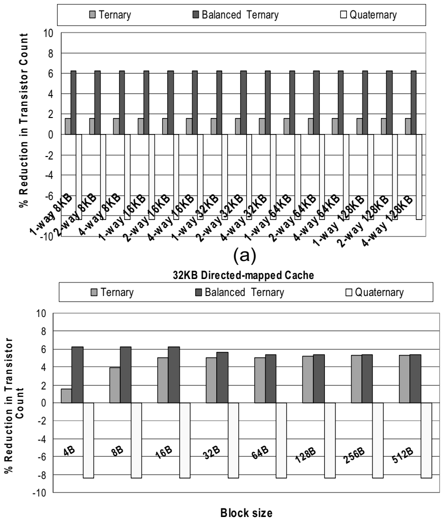
2.2. Binary-indexed Radix-r-tagged (BIRT) Cache

The SoC model with BIRT caches as shown in **Figure 1c** has *radix-r* data and instruction buses with a binary address bus. Indexing to the caches is performed in binary but tag bits are converted into *radix-r* format. In a BIRT cache, tags and data are stored in *radix-r* form in the tag and data array cells. This model reduces the tag storage area in the cache as in the RIRT cache model. However, it has an advantage over the RIRT, which is that it does not convert index bits into *radix-r* form since the index bits are used only to index to a cache line and are not stored in the caches. Similar to the RIRT cache, tag comparison has to be done in *radix-r*, this adds extra complexity to *radix-r* comparator logic design.

3. Cache Area Requirements

The RIRT and BIRT cache models need to store *radix-r* tags in CAM, and associative tag comparison has to be performed in *radix-r*. The storage area requirements for these two cache models may be prohibitively high. Besides, accessing the RIRT cache using *radix-r* index does not save space at all because the index field is not stored in the cache. Therefore, the conversion of all address bits on the bus is wasteful in the RIRT cache. For these reasons, we will not further analyze the RIRT and BIRT cache models in this paper.

In this section, we compare the data array storage area requirements of a fully binary cache with that of a BIBT cache in terms of transistor counts. We propose three BIBT *radix-r* caches: 1) BIBT ternary, 2) BIBT balanced ternary and 3) BIBT quaternary caches. A data array in a cache consists of words (i.e. rows) and digit lines (i.e. columns) and there is a memory cell that keeps 1-digit of data at each intersection of a row and a column. In general, the data array memory cells of caches are designed with CMOS SRAM technology in order to increase performance. A basic CMOS SRAM memory cell contains 6 transistors [15]. Recently, a CMOS SRAM multiple-valued memory cell has been designed and presented that its layout and performance characteristics approach to the binary CMOS SRAM memory cell [2]. Based on this work, a ternary (also balanced ternary) memory cell contains 9 transistors and a quaternary cell has 13 transistors.



(a)

(b)

Figure 2 Percentage reductions in the number of transistors in the BIBT *radix-r* cache: **(a)** various cache configurations of fixed 4-byte block size, **(b)** 32KB directed-mapped cache with variable block size

Figure 2a shows the percentage reduction in the number of transistor counts in data array memory cells of the BIBT ternary/balanced ternary and quaternary caches with respect to the fully-binary cache with different cache sizes of fixed 4-byte block size. We assume that the address and data buses are 32-bit long. As seen from the graph, the data array storage savings do not depend on the cache parameters such as the size and associativity. The change in the number of cache size and/or associativity has no effect on the size of data arrays. Thus, the percentage reduction of the number of transistors for all BIBT *radix-r* caches stays constant over various cache parameters. In summary, storing data in ternary and balanced ternary formats can save at about 1.5% and 6.2% in cache storage area, respectively. Note that balanced ternary uses only one *trit* fewer than ternary but its saving in the data storage area is about 5% percentage points better than ternary. On the other hand, the quaternary BIBT cache increases cache storage area about 8.3% due to its large number of transistor usage.

We also vary the block size to observe the trend in the transistor savings in memory cells as shown in **Figure 2b**. We show the results using a 32KB direct-mapped cache since the size and associativity do not change the percentage reduction or increase in the data storage area. The percentage reduction rate in the ternary cache slightly increases as the block size is increased and becomes very close to the balanced ternary. This is due to fact that the advantage of 1 fewer *trit* in ternary number system becomes negligible with wider cache block sizes. Similar to **Figure 2a**, the percentage increase rate in the quaternary cache does not vary with cache block size. This is because the number of cache lines in the cache is halved as the quaternary cache block size is doubled.

In summary, the BIBT ternary and balanced ternary caches can reduce the data storage area while the quaternary cache model increases the cache area. When projecting these results to energy consumption domain, the ternary and balanced ternary caches reduce the cache leakage energy consumption, whereas the quaternary cache increases it. The cache leakage energy, which depends on the number of transistors, is the energy consumed when transistors in the cache are not switching, i.e. in standby mode. However, most energy consumed in the caches is due to dynamic energy consumption that depends on the number of the switching activities made by the transistors in the cache [11].

4. Dynamic Cache Energy Models

The energy consumption in caches consists of three components: 1) the address decoding circuitry, 2) the cell array (i.e. tag and data) and 3) the peripheral circuitry (i.e. due to driving the external buses). The energy dissipated by the cell arrays designed with CMOS SRAM technology is caused by the switching activities of the bit lines.

We formulate the dynamic cache energy consumption for the fully binary and BIBT *radix-r* cache models. We focus on the cell array energy consumption since the energy consumed by the address decoding and peripheral circuitries is the same for both the binary and BIBT *radix-r* caches. The energy consumption of the cell arrays in a binary cache is defined as formulated in [11]:

$$\begin{aligned} E_{cell\_array} &= E_{tag} + E_{data} \\ E_{tag} &= TWS * TBLS * TBL_{trans} \qquad \textbf{(1)} \\ E_{data} &= DWS * DBLS * DBL_{trans} \end{aligned}$$

Here, *TWS* and *DWS* are tag and data word sizes that denote the number of memory cells in a word. *TBLS* and *DBLS* are tag and data bit line sizes, which are the number of memory cells per bit line (i.e. the number of cache lines). TBL_{trans} and DBL_{trans} are the number of bit line transitions for tag and data cells, respectively. Bit line transitions occur due to cache read and writes. For tag arrays, the BIBT *radix-r* cache has the same word and bit line sizes as the fully binary cache because it stores tags in binary and has the same number of cache lines as the binary cache. Hence, only the energy consumption of the data arrays will be considered from this on when comparing the energy consumption of the binary cache with the BIBT *radix-r* cache. The energy consumptions in the data arrays of the fully binary and BIBT *radix-r* caches are shown below:

$$\begin{aligned} E_{data}^{bin} &= WS_{bin} * BLS_{bin} * BL_{trans}^{bin} \\ E_{data}^{r} &= WS_{r} * DLS_{r} * DL_{trans}^{r} \end{aligned} \quad \textbf{(2)}$$

DLS_r and DL^r_{trans} stand for digit line size and the number of transition in digit lines in *radix-r* format. The BIBT *radix-r* cache has the same number of digit lines as the bit lines in the binary cache because the number of cache lines is the same in both caches. Hence, the cache energy model of the BIBT *radix-r* cache in **Equation 2** can be given as follows:

$$E_{data}^{r} = WS_{r} * BLS_{bin} * DL_{trans}^{r} \quad \textbf{(3)}$$

Also, a word line size in *radix-r* is reduced by a factor of *$log_2(r)$* where *r* represents the radix number in the BIBT *radix-r* cache with the following equation:

$$WS_r = \left\lceil \frac{WS_{bin}}{\log_2 r} \right\rceil \quad \textbf{(4)}$$

Now, WS_r in **Equation 3** is replaced by **Equation 4** and we get the following the energy consumption of the data arrays for the BIBT *radix-r* cache model.

$$E_{data}^{r} = \left\lceil \frac{WS_{bin}}{\log_2 r} \right\rceil * BLS_{bin} * DL_{trans}^{r} \quad \textbf{(5)}$$

The percentage of energy reduction in the data arrays of the BIBT *radix-r* cache with respect to the binary cache can be derived as follows:

$$\%\,\text{Reduction in Energy} = 100 * \left(1 - \frac{\left\lceil \frac{WS_{bin}}{\log_2 r} \right\rceil * DL_{trans}^{r}}{WS_{bin} * BL_{trans}^{bin}} \right) \quad \textbf{(6)}$$

Equation 6 tells us that the dynamic cache energy reduction in the data arrays depends on the bit line transition activities of the binary cache and the digit line transition activities of the BIBT *radix-r* cache, the word line size of the binary cache and *r* value. In the next section, we present an empirical study to contrast the dynamic energy consumed in the data arrays of these caches by measuring the transition activities.

Table 1 Benchmarks

MediaBench [13]	*adpcmencoder, unepic, mpeg2decoder, cjpeg, g721encoder, gsm encoder*
MiBench [14]	*sha, fft, rijndael, susan, crc32*
Other	*fir, matmul (matrix multiplication), k-means clustering, viterbi decoder*

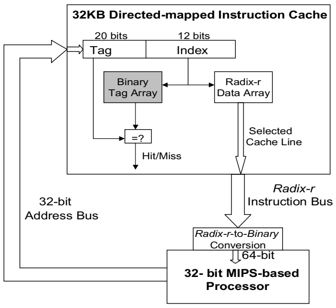


Figure 3 32-bit *MIPS*-based processor with 32KB direct-mapped instruction cache in a SoC system

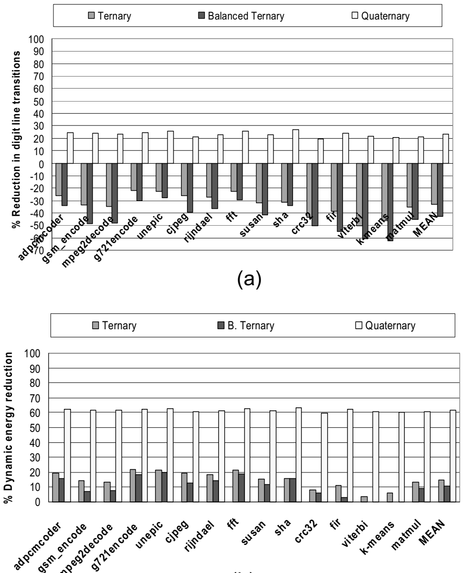
5. Experimental Results

We run 11 embedded benchmarks from two benchmark suites and 4 other benchmarks as shown in **Table 1**. In this paper, we focus on an energy model of instruction cache. We use the *SimpleScalar* [12] simulator to simulate a *MIPS*-based processor with a 64-bit ISA and a 32KB directed-mapped instruction cache with a block size of 8-bytes. Each benchmark is executed by the simulator to observe instruction cache transactions, and then estimate the dynamic energy consumed in the data arrays of the BIBT binary, ternary, balanced ternary and quaternary caches by counting cache bit and digit line transitions. Each *SimpleScalar* instruction has 64 bits that forms an 8-byte block size, which is fetched from the instruction cache at each cycle. In case of a cache miss, the missed word line is brought in from the memory and written into the instruction cache. The block sizes for the ternary, balanced ternary and quaternary caches are 42 trits, 40 trits and 32 quatrits, respectively. The simulated SoC system is shown in **Figure 3**.

5.1. Digit line Transitions

Figure 4a shows the percentage reduction in the number of digit line transitions in the data arrays of the BIBT ternary, balanced ternary and quaternary instruction caches. The percentage reduction in the figure is compared to the bit line transitions of the fully binary instruction cache. The last column in the figure shows the arithmetic mean of all benchmarks.

The figure shows that the digit line transitions actually increase if the BIBT ternary and balanced ternary instruction caches are used. The balanced ternary number system is worse than the ternary because using -1 as a *trit* value causes more *trit* changes in an instruction word than its ternary counterpart. Both number systems have more *trit* changes in an instruction word than the bit changes in the binary one, and therefore have more digit line transitions than the binary. The ternary and balanced ternary caches increase the digit line transition activity by 33% and 43%, respectively across all benchmarks. In contrast, the quaternary number system decreases the digit line transition activity by 23% and has less digit changes in an instruction word than the other number systems including the binary.



(a)

(b)

Figure 4 Percentage reductions in **(a)** the number of digit line transitions **(b)** dynamic cache energy consumption

5.2. Dynamic Cache Energy Consumption

Figure 4b shows the percentage reduction in dynamic cache energy consumption estimated by **Equation 6**. Although the ternary and balanced ternary caches have more digit line transitions than the binary, they reduce dynamic energy consumption for almost all benchmarks. This is caused by the reduction in the number of instruction word size. The dynamic cache energy consumption of the ternary cache is slightly better than the balanced ternary except in *sha* where having 1 fewer trit pays off in terms of more reduction in energy for the balanced ternary cache. On the other hand, the balanced ternary cache increases the energy consumption by 0.5% and 1.5% in *viterbi* and *k-means* benchmarks due to the excessive number of digit line transitions.

The quaternary cache, on the other hand, reduces the dynamic energy by significant amounts for all benchmarks. This is due to both savings in the instruction word size and digit line transitions. Overall, it can reduce the dynamic energy consumption by about 62% across all benchmarks. This is only about 15% and 11% for the ternary and balanced ternary caches.

5.3. Discussion

In terms of cache storage area usage, the ternary and balanced ternary caches are more advantageous than the quaternary cache. The quaternary cache increases chip area about 8%, this has an effect on the leakage or standby (i.e. when the cache has no switching activity) cache energy consumption since the leakage energy on caches relies on the number of transistors. Thus, we expect an increase in the leakage energy consumption at about 8%. On the other hand, using the quaternary cache reduces dynamic cache energy consumption up to 62%. If power consumption is of major concern and the leakage current of the cache is negligible, then the quaternary cache is the most power-efficient option. Besides, quaternary logic can be easily interfaced with binary logic using fast conversion circuits [9] [10].

We believe that the conversion circuitry between the processor and instruction cache can be designed in such a way that it does not add an excessive number of penalty cycles on the cache hit path. This argument is supported by the following facts that it takes about 8ns to convert from quaternary to binary in [9]. Similarly, conversion from ternary to binary takes about 9ns according to [10].

6. Related Work

There have been some attempts for designing multiple-valued memory units such as ROM, DRAM and CAM in the MVL literature [19] [20] [21] [22] [23] to increase memory density. We, in this paper, propose architectural cache models that are designed with multiple-valued static CMOS memory cells. Our goal is to show the viability of using multiple-valued CMOS SRAM-based caches to improve the area and energy consumption. This is the first study that investigates the architectural issues of multiple-valued caches, and compares their area and energy to the binary ones.

7. Conclusion and Future Work

We have proposed novel multiple-valued cache models to reduce the cache storage area and cache dynamic energy consumption for embedded systems. Three multiple-valued cache architectural models have been discussed, which are *radix-r*-indexed *radix-r*-tagged (RIRT), binary-indexed binary-tagged (BIBT) and binary-indexed *radix-r*-tagged (BIRT). We have found out that the BIBT cache model was the most cost-effective model among all. Then, we have focused on designing the data array cells of the BIBT cache model using ternary, balanced ternary and quaternary number systems to analyze the effects of using different *radix-r* number in storing data. Our analytical study for cache silicon area has shown that an embedded System-on-a-chip (SoC) equipped with a BIBT cache model can reduce the cache data storage area up to 6% regardless of cache parameters. We have also simulated several embedded benchmarks using the *SimpleScalar* simulator to demonstrate that dynamic cache energy consumption can be reduced up to 62% in a multiple-valued instruction cache.

We are planning to perform similar analysis and experiments for data caches. Further, we are investigating the viability of using the BIRT cache model in the context of storing *radix-r* tags in the content-addressable memory cells and *radix-r* associative tag comparison.

References

[1] S. L. Hurst, “Multiple-valued logic – Its Status and its future”, *IEEE Transactions on Computers*, Vol. C-33, 1984.

[2] U. Cilingiroglu and Y. Ozelci, “Multiple-Valued Static CMOS Memory Cell”, *IEEE Transactions on Circuits and Systems-II, Analog and Digital Signal Processing*, Vol. 48, No. 3, Mar. 2001.

[3] K. W. Current, “Current-Mode CMOS Multiple-Valued Logic Circuits”, *IEEE Journal of Solid-state Circuits*, Vol. 29, No. 2, Feb. 1994.

[4] X. W. Wu, “CMOS ternary logic circuits”, *IEE Proceedings*, Vol. 137, No. 1, Feb. 1990.

[5] D. Etiemble and M. Israël, “Comparison of Binary and Multivalued ICs According to VLSI Criteria”, *IEEE Computer*, Apr., 1988.

[6] A. Srivastava, “Back gate bias method of threshold voltage control for the design of low voltage CMOS ternary logic circuits”, *Microelectronics Reliability*, 2000.

[7] C. Y. Wu, and H. Y. Hang, “Design and Application of Pipelined Dynamic CMOS Ternary Logic and Simple Ternary Differential Logic”, *IEEE Journal of Solid-state Circuits*, Vol. 28, No. 8, Aug. 1993.

[8] F. Q. Li, M. Morisue and T. Ogata, “A Proposal of Josephson Binary-to-Ternary Converter”, *IEEE Transactions on Applied Superconductivity*, Vol. 5, No. 2, June, 1995.

[9] I. M. Thoidis, D. Soudris, I. Karafyllidis and A. Thanailakis, “The Design of Low Power Multiple-valued Logic Encoder and Decoder Circuits”, *Proceedings of the Sixth IEEE International Conference on Electronics, Circuits and Systems*, Vol. 3, Sep., 1999.

[10] H.N. Venkata, “Ternary and Quaternary Logic to Binary Bit Conversion CMOS Integrated Circuit Design Using Multiple Input Floating Gate MOSFETs”, *MS Thesis*, Louisiana State University, Baton Rouge, Dec. 2002.

[11] C-L. Su and A. M. Depain, “Cache Design Trade-offs for Power and Performance Optimization: A Case Study”, *the Proceedings of the International Symposium on Low-Power Electronics and Design (ISLPED’95)*, 1995.

[12] D. Burger and T. Austin, “The SimpleScalar Tool Set, Version 2.0”, *Technical Report #1342*, Computer Sciences Department, University of Wisconsin-Madison, June 1997.

[13] C. Lee, M. Potkonjak and W. H. Mangione-Smith, “MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems”, *Proceedings of the 30th Annual IEEE/ACM International Conference on Microarchitecture (Micro-30)*, Raleigh, N.C., Dec. 1997.

[14] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge and R. B. Brown, “MiBench: A Free, Commercially Representative Embedded Benchmark Suite*”, the IEEE 4th Annual Workshop on Workload Characterization*, Austin, TX, Dec. 2001.

[15] J. M. Rabaey, A. Chandrakasan and B. Nikolić, “Digital Integrated Circuits”, *Prentice Hall Electronics and VLSI Series*, 2003.

[16] S. Manne, A. Klauser and D. Grunwald, “Pipeline Gating: Speculation Control for Energy Reduction”, *Proceedings of the 25th Annual International Symposium on Computer Architecture, June 1998.*

[17] M. Gowan, L. Biro, and D. Jackson, “Power Considerations in the Design of the Alpha 21264 Microprocessor”, *the 35th Design Automation Conference*, 1998.

[18] J. Montanaro et al., “A 160 MHZ 32-b 0.5 W CMOS RISC Microprocessor”, *Digital Technical Journal, Vol. 9, No. 1*, 1997.

[19] M. Stark, “Two bits per cell ROM”, *Proc. COMPCON*, 1981.

[20] D. A. Rich, K. L. C. Naiff, K. G. Smalley, “A Four-State ROM Using Multilevel Process Technology”, *IEEE Journal of Solid-State Circuits*, Vol. SC-19, No. 2, Apr. 1984.

[21] B. Donoghue, P. Holly and K. Ilgenstein, “A 256K HCMOS ROM Using a Four-State Cell Approach”, *IEEE Journal of Solid-State Circuits*, Vol. SC-20, No. 2, Apr. 1985.

[22] T. Okuda and T. Murotani, “A Four-Level Storage 4-Gb DRAM”, *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 11, Nov. 1997.

[23] T. Hanyu, S. Aragaki and T. Higuchi, “Functionally separated, multiple-valued content-addressable memory and its applications”, *IEE Proc. Circuits Devices Systems*, Vol. 142, No. 3, June 1995.