01 Apr 2016

Comparison of Binary and Multi-Level Logic Electronics for Embedded Systems

Shirly M. Damti

Steve Eugene Watkins
Missouri University of Science and Technology, watkins@mst.edu

R. Joe Stanley
Missouri University of Science and Technology, stanleyj@mst.edu

Recommended Citation

S. M. Damti et al., "Comparison of Binary and Multi-Level Logic Electronics for Embedded Systems," *Proceedings of SPIE Smart Structures and Materials + Nondestructive Evaluation and Health Monitoring (2016, Las Vegas, NV)*, vol. 9801, SPIE, Apr 2016.
The definitive version is available at https://doi.org/10.1117/12.2219374

Comparison of Binary and Multi-level Logic Electronics for Embedded Systems

Shirly M. Damti, Steve E. Watkins, *Senior Member SPIE*, and R. Joe Stanley[*]
Department of Electrical and Computer Engineering
Missouri University of Science & Technology, Rolla, Missouri 65409, USA

ABSTRACT

Embedded systems are dependent on low-power, miniaturized instrumentation. Comparator circuits are common elements in applications for digital threshold detection. A multi-level, memory-based logic approach is in development that offers potential benefits in power usage and size with respect to traditional binary logic systems. Basic 4-bit operations with CMOS gates and comparators are chosen to compare circuit implementations of binary structures and quaternary equivalents. Circuit layouts and functional operation are presented. In particular, power characteristics and transistor count are examined. The potential for improved embedded systems based on the multi-level, memory-based logic is discussed.

Index Terms – multi-level circuits, embedded systems, comparator

1. INTRODUCTION

Many applications exist in which minimal electronics hardware design is important. In particular, embedded systems for control devices, sensors nodes, etc. and components within micro-processor-based systems often have constraints regarding power and size. Binary digital technology has experienced decades of steady improvements in device density, speed, and cost. However, further miniaturization of transistors has physical limitations and other approaches are being actively considered for computing systems [1]. Various approaches with multi-level electronics have been explored as a means to reduce interconnect lines and processing components [2-4]. Multi-level FPGA systems and CMOS voltage-mode gates are examples [5,6].

This work explores a multi-level, memory-based approach that uses CMOS technology in unconventional ways. Computation is accomplished with multi-level signals and through memory manipulation [7-10]. Quaternary or four-level designs are described [7,10], although the approach can be extended to other multi-level options. Quaternary designs offer a reduced number of interconnect lines with modest discrimination needs among signal levels. For instance, 2-bit binary designs require two signal lines while equivalent quaternary designs only require one signal line. Logic gate implementations in this technology compare favorably with equivalent circuits using binary logic [10].

Comparator circuits are selected to compare binary and multi-level logic implementations. A quaternary implementation is shown for the functional equivalent of a basic 4-bit comparator that is based on CMOS binary gate structures, i.e. AND, OR, NOT (inverter), and NAND. Truth tables and circuit layouts are presented for each approach. Power characteristics and transistor count are examined as comparison measures. Analog-to-binary and analog-to-quaternary converters are also introduced as the front-end for the embedded quaternary system. The use of the multi-level, memory-based logic technology for embedded systems is discussed.

[*] Author E-mail Addresses: smdr86@mst.edu, steve.e.watkins@ieee.org, and stanleyj@mst.edu
Contact Author: S. E. Watkins. 121 EECH, Missouri University of Science and Technology (formerly University of Missouri-Rolla), 301 W. 16th St., Rolla 65409-0040 573-341-6321, Fax 573-341-4532

2. EMBEDDED SYSTEM INSTRUMENTATION

2.1 Hardware Constraints in Embedded Systems

Embedded systems are computer systems, typically microprocessor-based, that perform a dedicated operation by interacting with the surrounding environment using sensors or other mechanisms within the context of overall electrical or mechanical systems [11]. Embedded systems have hardware and software components and often have real-time operating constraints [12]. Embedded systems are prevalent in our everyday lives, with applications in control, instrumentation, home, automotive, and office. Embedded systems are conventionally used in applications that are smaller in size, lower unit cost, lower power consumption, and more autonomous than general purpose computer system applications [11]. Embedded systems have limited programming and support resources compared to general purpose computer systems, making programming and interfacing with other devices more difficult. However, embedded systems are dedicated to perform targeted tasks or operations. Thus, embedded system designers can optimize the circuit to minimize its size including circuit components, power consumption, upgrade capability, and performance.

2.2 Comparator Circuits for Digital Threshold Detection

Comparators are digital components in computer-based systems and in devices interfaces that perform the operation of comparing the magnitude of two input binary words, particularly equality. Common comparator circuit applications include dedicated control devices, such as device selection and identification, and sensor nodes, among others. Several studies have explored alternative comparator designs, including serial architecture [13], parallel architectures [14], and modified parallel architecture [15,16] to enhance speed, power consumption, and size of design. In these existing approaches, conventional digital logic components and Complementary Metal on Oxide on Semiconductor (CMOS) transistor configurations were examined. In the context of this work, a comparator implementation is considered that is targeted toward applications such as digital threshold detection. Example applications include detection of air pollution in vehicles [17].

2.3 Conventional Binary Comparator vs. Equivalent Quaternary Comparator

The complexity of a binary logic gate configuration for a comparator grows as the number of bits increase. The three status outputs from a magnitude comparison of binary words A and B are

$$A < B\ (L = 1),\ A = B\ (E = 1), \text{ and } A > B\ (G = 1)$$

to represent less than (L), equal to (E), and greater than (G) conditions, respectively. From an implementation perspective, multi-bit magnitude comparators may be implemented by cascading 1-bit comparators. Let the binary words for the n-bit case be

$$A = a_{n-1}\ldots a_3a_2a_1a_0 \text{ and } B = b_{n-1}\ldots b_3b_2b_1b_0.$$

The 1-bit comparators use input bits a_i, b_i, L_i, E_i, and G_i, where $L_i = 1$ if $a_{i-1}\ldots a0 < b_{i-1}\ldots b_0$, $E_i = 1$ if $a_{i-1}\ldots a_0 = b_{i-1}\ldots b_0$, and $G_i = 1$ if $a_{i-1}\ldots a_0 > b_{i-1}\ldots b_0$, and outputs L_{i+1}, E_{i+1} and G_{i+1} to provide the status of the n-bit word magnitude comparison at the ith bit position. Specific examples of logic and implementations are described later.

The equivalent quaternary comparator takes the quaternary words A and B and provides a single output in which three of the levels are correlated with the A < B, A = B, and A > B conditions, i.e. L, E, and G. (The remaining level would be unused.) A single quaternary signal is equivalent to a 2-bit binary signal. The quaternary signal has level for a 0 value (equivalent to a binary 00), a 1 value (equivalent to a binary 01), a 2 value (equivalent to a binary 10), and a 3 value (equivalent to a binary 11).

Figure 1a shows a block diagram for a 2-bit binary comparator. The inputs are the 2-bit binary words $A = a_1a_0$ and $B = b_1b_0$ and the three prior 2-bit values for L, E, and G. The three 2-bit binary outputs are the next values for L, E, and G. The equivalent quaternary block diagram is shown in Figure 1b. The inputs are the quaternary words A and B and the single prior L/E/G value. The output is a single quaternary L/E/G value. The reduction in signal lines is obvious. For both binary and quaternary cases, Figure 2 shows the signal levels for these inputs and the outputs.

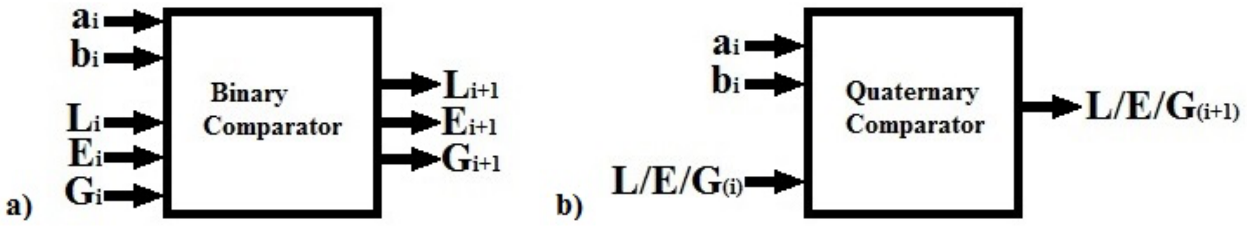

Figure 1. Comparator Functional Block Diagrams for a) Binary Signals and b) Quaternary Signals.

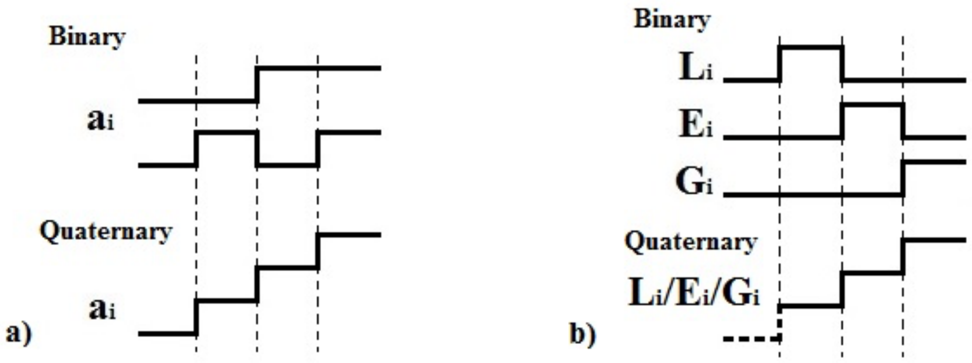

Figure 2. a) Input Words for Binary and Quaternary and b) Comparator Signals for Binary and Quaternary.

3. MULTI-VALUE, MEMORY-BASED LOGIC

3.1 Quaternary, Memory-based Processing

An optimized embedded system will provide the desired logic functionality while minimizing the size and number of circuit components and the power consumption. A quaternary, memory-based processor will replace the conventional binary processor, e.g. a comparator. Since an embedded system may have other functions that use binary digital electronics or may need to interface with a binary digital world, an application may require hybrid features. Hence, the overall replacement electronics consist of a quaternary processor and converters (both binary-to-quaternary B/Q and quaternary-to-binary Q/B). The converters can be implemented with the same technology as the processor [18]. Figure 3 shows a quaternary equivalent for a 2-bit processor.

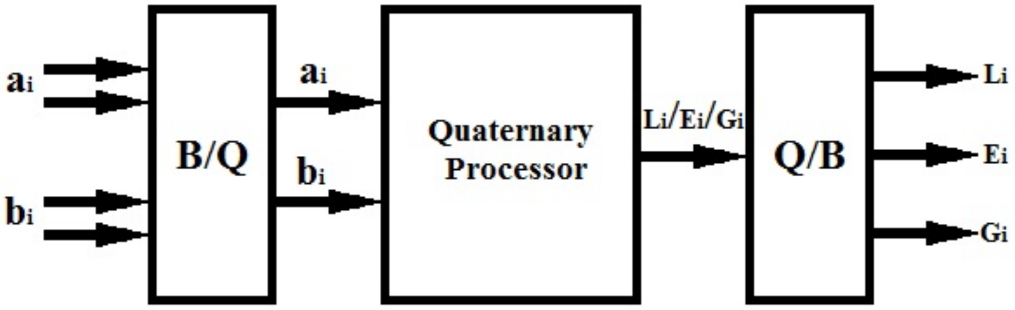

Figure 3. Proposed Hybrid System.

This work explores a multi-level, memory-based approach that uses CMOS technology in unconventional ways. Computation is accomplished with multi-level signals and through memory manipulation [7-10]. Quaternary or four-level designs are described [7,10], although the approach can be extended to other multi-level options. Quaternary designs offer a reduced number of interconnect lines with modest discrimination needs among signal levels. For instance, 2-bit binary designs require two signal lines while equivalent quaternary designs only requires one signal line. Logic gate implementations in this technology compare favorably with equivalent circuits using binary logic [10].

3.2 Quaternary, Memory-based Logic Approach

The memory-based approach to computing consists of CMOS architectures of arrays and drivers [10,18]. For a quaternary processor, the array consists of sixteen cells that are arranged in a 4x4 matrix of cells. Each cell in the array contains a set value structure of 0, 1, 2, or 3 signal levels that are determined by internal voltage reference connections. The drivers decode quaternary inputs and produce control signals for the array. One driver signal selects a row and the other selects a column. The quaternary value in a selected cell is the output. Hence, any truth-table relationship can be implemented by setting the correct cell values (i.e. by internal connections). The block diagram for the processor is shown in Figure 4. This example takes two quaternary words A and B and produces a single quaternary output. If bigger words are desired, additional array processors can be added (not shown). Prior literature describes the implementation of the equivalent of binary logic gate functions in both simulation and hardware [10]. This prior work simply illustrates the approach with logic gates – an efficient memory-based system would be designed to produce the desired logic system function rather that directly implementing the equivalent arrangement of logic gates.

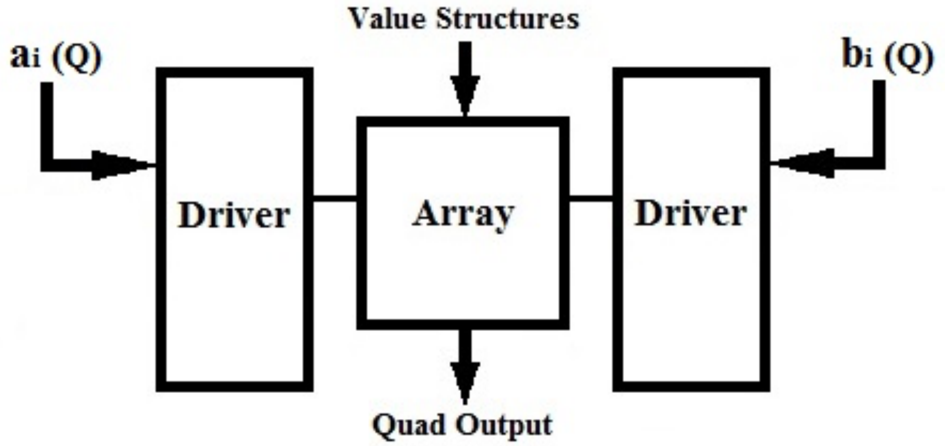

Figure 4. Example of a Quaternary, Memory-based Processor using Input Drivers and an Array.

4. SIGNALS AND ARCHITECTURE

4.1 Binary Logic Operations

The primary binary logic operations used in this research correspond to the AND, OR, and NOT (inverter) gates. Each logic gate accepts one or more input values and outputs one value with each input and output representing one of two logic levels. From a hardware perspective, the low and high logic levels are determined by specific voltage values (the values depend on the logic family of the gates). In general, the lower voltage value in a specific logic family corresponds to "low" or "zero" level, and the higher voltage value corresponds to "high" or "one" level. Each gate carries out a specific operation independently of the logic family. For instance, the AND gate accepts two or more values, and outputs a "high" value only if all inputs are "high" and a "low" value otherwise. The OR gate accepts two or more values and outputs their sum, and therefore outputs a "high" if any input value is "high", and a "low" value is all inputs are "low". The inverter accepts one input value and negates it, so a "low" input produces a "high" output, and vice-versa. In this research, Fairchild Semiconductor's TinyLogic binary logic gates [19] were used in analyzing a 4-bit binary comparator for comparison with the proposed quaternary comparator. The number of transistor, power consumption, and worst propagation delay values are determined using manufacturers specifications.

Perhaps the most basic comparator is the 1-bit binary comparator which is used in this report. A 1-bit binary comparator accepts the five inputs a_i, b_i, L_i, E_i, and G_i and it produces the three outputs L_{i+1}, E_{i+1}, and G_{i+1} [20]. Two inputs to the 1-bit binary comparator inputs are binary values that are put for comparison, namely $A = a_i$ and $B = b_i$. The L_i, E_i, and G_i inputs indicate of the previous relationship between a_{i-1} and b_{i-1}; these additional inputs make comparator cascades possible. Table 1 illustrates the truth table for various input and output combinations. If no prior relationship is considered for a single 1-bit comparison, the L_{i+1}, E_{i+1}, and G_{i+1} outputs represent less than, equal to, and greater than conditions between a_i and b_i, respectively, i.e.
if $a_i > b_i$, then $G_{i+1} = 1$ ($E_{i+1} = L_{i+1} = 0$);
if $a_i = b_i$, then $E_{i+1} = 1$ ($G_{i+1} = L_{i+1} = 0$); and
if $a_i < b_i$, then $L_{i+1} = 1$, ($G_{i+1} = E_{i+1} = 0$).
In this context, only one L/E/G value can be high for a meaningful result.

Table 1. Truth Table for 1-Bit Binary Comparator (Binary Values $A = a_i$ and $B = b_i$).

L_i *	**E_i ***	**G_i ***	**a_i**	**b_i**	**L_{i+1} ***	**E_{i+1} ***	**G_{i+1} ***
0	0	0	0	0	0	1	0
0	0	0	0	1	1	0	0
0	0	0	1	0	0	0	1
0	0	0	1	1	0	1	0
1	0	0	0	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	0	0	0	1
1	0	0	1	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	1	1	0	0
0	1	0	1	0	0	0	1
0	1	0	1	1	0	1	0
0	0	1	0	0	0	0	1
0	0	1	0	1	1	0	0
0	0	1	1	0	0	0	1
0	0	1	1	1	0	0	1

* Status Condition: If two or three of the L/E/G signals are high, then the condition is an error or don't care.

The 4-bit binary comparator can be constructed by simply using four 1-bit comparators in cascade. In this case, two binary nibbles, $A=a_3a_2a_1a_0$ and $B=b_3b_2b_1b_0$, are compared in a series of operations beginning with their least significant bits. The initial values of L_i, E_i, and G_i of the first 1-bit comparator are 0, 1, and 0, respectively, since at this point no bits of A and B have been compared, and the two nibbles are considered equal. (These initial states may also be all low as well.) After the comparison operation of each pair of bits a_i and b_i, the outputs L_{i+1}, E_{i+1}, and G_{i+1} become the initial state of the next 1-bit comparator in the cascade where the next pair of bits a_{i+1} and b_{i+1} are compared [20]. As listed in Table 1, for any pair of bits where $a_i > b_i$, the outputs are G_{i+1} is "high", and the others are "low", regardless of previous states (independently of the values of L_i, E_i, and G_i). Similarly, for any pairs of bits where $a_i < b_i$, the output L_{i+1} is "high", and the others are "low". On the other hand, when $a_i = b_i$, the outputs L_{i+1}, E_{i+1}, and G_{i+1} retain the input values of L_i, E_i, and G_i, respectively.

4.2 Quaternary Logic Operations

In multi-level logic, more than two states exist per each input or output. In quaternary-level logic, four different states exist that correspond to four different voltage levels for each bit in the quaternary word. The quaternary values 0, 1, 2, 3 are equivalent to binary values 00, 01, 10, and 11, respectively. The results of the logic operations on two 1-bit quaternary values are paralleled to the results that would have been produced by two 2-bit binary values. For example, a binary AND operation on input quaternary values 1 and 3 (binary values 01 and 11) produces a quaternary output of value of 1 (binary value of 01). Other non-binary logic operations may be defined. A basic digital comparator or magnitude comparator operation has the quaternary words (i.e. $A = a_i$ and $B = b_i$) to be compared as inputs and the desired output representing less than, equal to, and greater than conditions. Note that the output can be a single quaternary value, cf. Figure 2b. Table 2 shows the proposed truth table for this comparator operation in quaternary.

Table 2. Truth Table for Quaternary Comparator (Quaternary Values $A = a_i$ and $B = b_i$).

a_i	b_i	L_i / E_i / G_i *	a_i	b_i	L_i / E_i / G_i *
0	0	2	2	0	3
0	1	1	2	1	3
0	2	1	2	2	2
0	3	1	2	3	1
1	0	3	3	0	3
1	1	2	3	1	3
1	2	1	3	2	3
1	3	1	3	3	2
* Status Condition: 0 (error or don't care), 1 (L, A < B), 2 (E, A = B), and 3 (G, A > B)					

Comparator designs for longer quaternary words require additional relationships. One approach is to process each bit separately using the relationships in Table 2. These resulting L/E/G quaternary values can be processed through additional array processing. For instance, a comparison of 2-bit quaternary words (i.e. $A = a_ia_0$ and $B = b_1b_0$) can be performed by processing the least significant digits (LSD) a_0 and b_0 and the most significant digits (MSD) a_1 and b_1. The respective intermediate processing outputs (i.e. L_0, E_0, and G_0 and L_1, E_1, and G_1) can be processed as shown in Table 3. The output is the L/E/G value for the 2-bit quaternary word.

Table 3. Truth Table for Quaternary Comparator Cascade (Quaternary Values A = a_ia_0 and B = b_1b_0).

L_0 / E_0 / G_0 * From LSD (a_0 to b_0)	**L_1 / E_1 / G_1 * From MSD (a_1 to b_1)**	**L / E / G ***	**L_0 / E_0 / G_0 * From LSD (a_0 to b_0)**	**L_1 / E_1 / G_1 * From MSD (a_1 to b_1)**	**L / E / G ***
0	0	0 (error)	2	0	0 (error)
0	1	0 (error)	2	1	1
0	2	0 (error)	2	2	2
0	3	0 (error)	2	3	3
1	0	0 (error)	3	0	0 (error)
1	1	1	3	1	1
1	2	1	3	2	3
1	3	3	3	3	3
* Status Condition: 0 (error or don't care), 1 (L, A < B), 2 (E, A = B), and 3 (G, A > B)					

4.3 CMOS Architectures

The circuit architecture for a quaternary, memory-based circuit is described in prior work [10]. The driver and array structures can be implemented with standard CMOS fabrication technology. The basic components for the drivers and arrays are inverters, sense amplifiers, and pass amplifiers. These basic components have a higher transistor count than that required to implement individual binary logic gates. However, more complex circuitry functions can be implemented and optimized with fewer transistors than in the equivalent binary circuits that use complex assemblies of logic gates [18].

An example of the CMOS architecture for an array is shown in Figure 5. The value structures, e.g. the reference quaternary voltage levels, are the V_0, V_1, V_2, and V_3 connections at the top. The sixteen cells in the array are hardwired to a set value in the array. The control lines L_0, L_{0_N}, L_1, L_{1_N}, L_2, L_{2_N}, L_3, and L_{3_N} serve to select a particular row in the array. These lines are produced by the driver (not shown). The cell values in the selected row are passed to the intermediate output lines Out_0, Out_1, Out_2, and Out_3. Similar circuitry uses control lines from the second driver to select the desired column. The value of the selected cell, Out_i, is passed as the final output of the array. In terms of the inputs, the quaternary A = a_i selects the row and the quaternary B = b_i selects the column.

If a conversion from binary to quaternary (B/Q) or quaternary to binary (Q/B) is desired, similar circuitry is used. For instance, a binary-to-quaternary circuit feeds the LSD and the MSD values in the binary word A = a_1a_0 to four sets of pass amplifier structures. Each set is tied to the reference quaternary voltage levels (just as the cells in the array). The equivalent quaternary value to the input is selected and produced as the converter output. Details of the B/Q and Q/B converters will be discussed in future work.

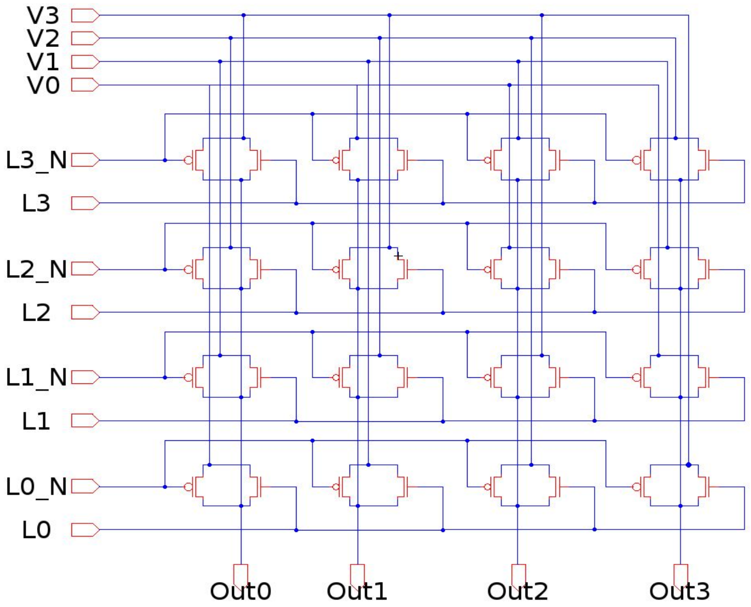

Figure 5. Quaternary Array Architecture in CMOS [10].

5. COMPARATOR IMPLEMENTATIONS

5.1 Conventional Binary (4-bit) Comparator

A conventional 4-bit binary comparator can be designed in various ways. Figure 6 shows a design utilizing AND, OR, and NOT logic gates [20]. This design is a cascade of four 1-bit identical comparators. The binary input bits are shown at the top and the initial L/E/G inputs are shown at the left. CMOS transistors are assumed and representative specifications were taken from Fairchild Semiconductor datasheets [19]. Transistor count, propagation delay, and power consumption were determined using typical gate configurations (assumed to be 4 for a 2-input NAND gate, 2 for a 2-input NOT gate, 6 for both a 2-input AND gate and a 2-input OR gate, and 2 additional transistor per each additional input). The propagation delay values were taken for the worst case. The power consumption values were determined on a per gate basis for the maximum estimate. Figure 7 shows a different design in which the NAND-gate equivalent of the cascade circuit in Figure 6 is utilized. Again, Fairchild CMOS specifications are assumed.

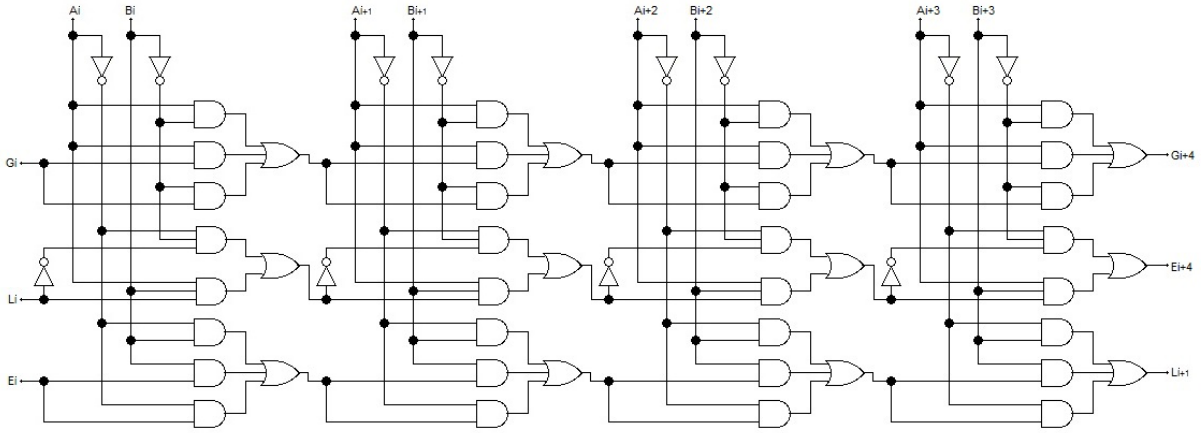

Figure 6. Binary Comparator Implementation Utilizing AND. OR, and NOT Gates.

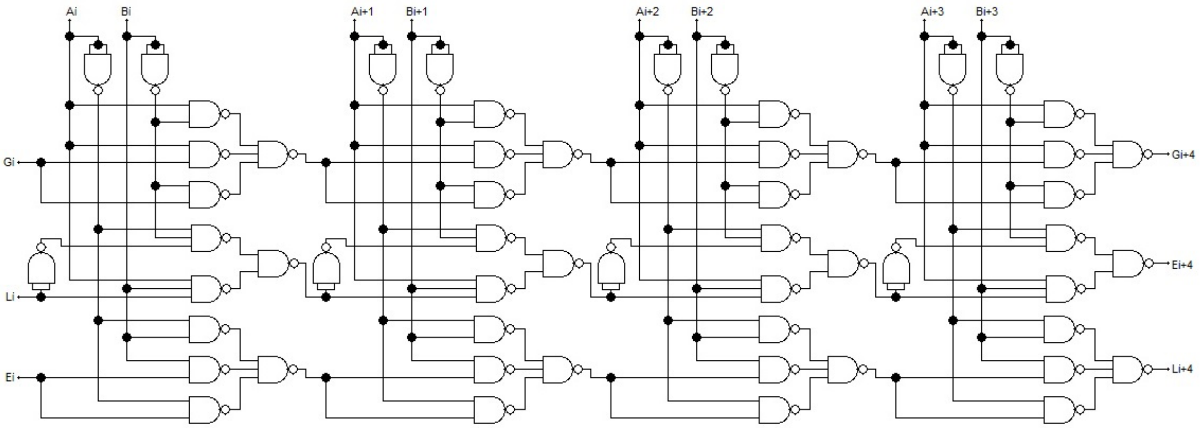

Figure 7. Binary Comparator Implementation Utilizing NAND Gates.

As an additional comparison, a commercial CMOS design is considered. The 4-Bit Magnitude Comparator CD4063B from Texas Instruments is used [21]. The comparator gate layout is shown in Figure 8. Transistor count, propagation delay, and power consumption were estimated based on available datasheet information. (The delay values, which are shown in the next section, are larger than those calculated for the implementations in Figures 6 and 7. This commercial comparator may have additional features that are not reflected on the listed layout.) This commercial comparator is chosen as a medium-speed device.

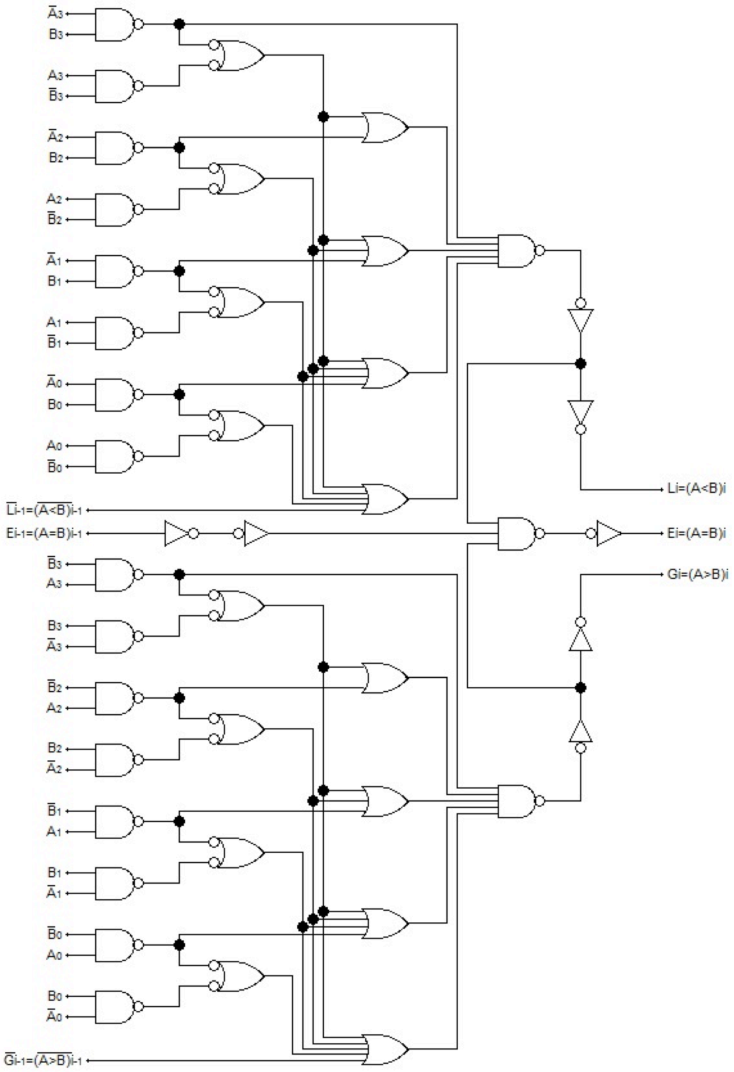

Figure 8. Binary 4-bit Commercial TI Comparator [21].

5.2 Equivalent Quaternary Comparator

The quaternary equivalent of a 4-bit binary operation uses 2-bit quaternary words A = a_1a_0 and B = b_1b_0. Figure 9 shows one design using the memory-based approach. Three driver-array circuits are used for the overall comparator. The first driver-array circuit handles the least significant digits (LSD) a_0 and b_0 and produces an intermediate output L_0, E_0, and G_0, cf. Table 2. The second driver-array circuit handles the most significant digits (MSD) a_1 and b_1 and produces an intermediate output L_1, E_1, and G_1, cf. Table 2. The intermediate outputs are inputs into the third driver-array circuit. The overall L/E/G output is produced using the truth table in Table 3. (The convention is that the array cell for row 0 and column 0 is the lower left-hand corner.) Transistor count, propagation delay, and power consumption were estimated based on design calculations [22]. A 65-nm CMOS technology with 1-GHz switching operation was assumed. As with many other memory-based systems, the propagation delay through the system is essentially uniform for all input cases. The power consumption was calculated assuming that the overall output from an input case is obtained before the next input is applied.

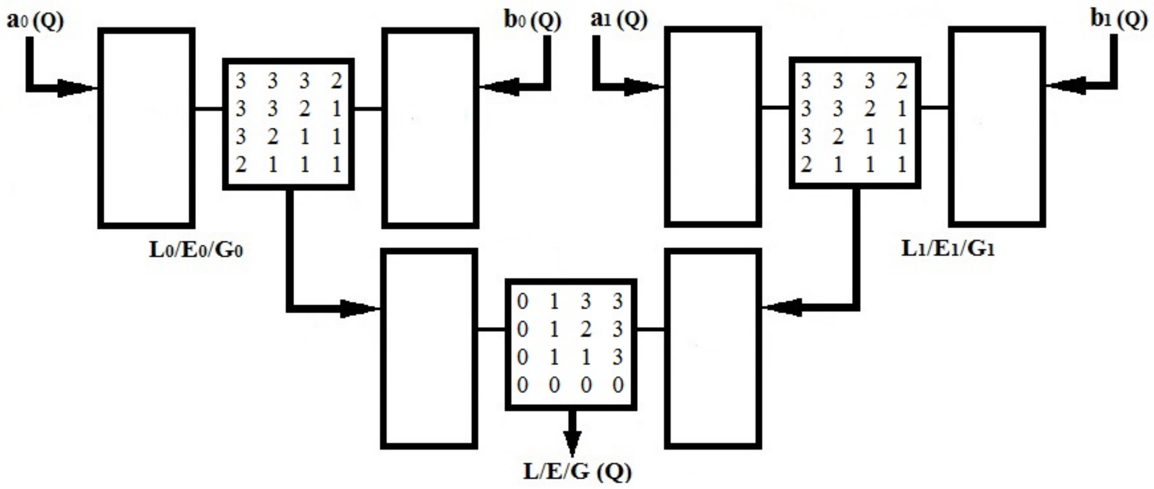

Figure 9. Quaternary Comparator Implementation.

6. COMPARATOR PERFORMANCE

Three binary comparators are considered. One utilized AND, OR, and NOT gates. Another utilized NAND gates. The third is a medium-speed commercial comparator. One quaternary, memory-based design is considered. The respective transistor counts, propagation delays, and power consumption values are listed in Table 4. For the binary AND/OR/NOT and NAND comparators, the initial entries reflect values for a single 1-bit comparator and the totals show the values time four for the four 1-bit comparators in the 4-bit cascade. For the quaternary comparator, values are shown for both the comparator with B/Q and Q/B converters (binary input to binary output) and the comparator alone (quaternary input to quaternary output).

Since the CMOS technologies for the four comparators are not identical, the comparisons show general trends. The total numbers of transistors for the binary comparators are similar; the number of transistors for the quaternary devices is greater especially with the converter overhead. While this trend is a disadvantage with regard to area needs, the reduction in interconnect lines for quaternary gives a compensating advantage. The memory-based processor design with its parallel processing does much better with propagation delay and switching power. Even with a non-optimized design, the quaternary version would be more comparable with high-speed, low-power binary comparators. For embedded applications in which component density, power, and performance issues are critical, the quaternary approach shows considerable promise even in a hybrid (mixed binary and quaternary) system. Also, optimizations of the design may provide enhancements in component count, speed, power consumption, and performance; more complex functions are expected to require fewer transistors than equivalent binary gate-based devices [18].

Table 4. Comparison of Circuit Implementations.

Features	Binary Comparators			Quaternary Comparator†	
	AND/OR/NOT	NAND Only	Commercial*	With B/Q & Q/B	W/o B/Q & Q/B
Number of Gates or Components	2-input AND: 6 3-input AND: 2 2-input OR: 1 3-input OR: 2 NOT: 3	2-input: 10 3-input: 4	-----	Drivers: 6 Arrays: 3 B/Q: 2 Q/B: 1	Drivers: 6 Arrays: 3
Transistors Needed	2-input AND/OR: 6 3-input AND/OR: 8 NOT: 2	2-input: 4 3-input: 6	-----	Drivers: 42 Arrays: 32 B/Q: 38 Q/B: 20	Drivers: 42 Arrays: 32
Total Transistors	80 x 4 = **320**	64 x 4 = **256**	**292**	**426**	**348**
Worst Propagation Delay per Gate	2-input AND/OR: 12.7 ns 3-input AND/OR: 19.0 ns NOT: 12.0 ns	2-input: 12.0 ns 3-input: 18.0 ns	-----	Drivers: 1 ns Arrays: 1 ns B/Q: 1 ns Q/B: 1 ns	Drivers: 1 ns Arrays: 1 ns
Worst Propagation Delay	12 +19 +12.7 = 43.7 ns 43.7 x 4 = **174.8 ns**	12 +18 +12 = 42 ns 42 x 4 = **168 ns**	**250 ns (typical)**	6 x 1 ns = **6 ns**	4 x 1 ns = **4 ns**
Power Consumption per Gate	AND/OR/NOT: 0.1344 W	2-/3-input: 0.1344 W	-----	Drivers: 1.40 µW Arrays: 1.06 µW B/Q: 1.25 µW Q/B: 0.65 µW	Drivers: 1.40 µW Arrays: 1.06 µW
Total Power Consumption	4 x 14 x 0.1344 = **7.5264 W**	4 x 14 x 0.1344 = **7.5264 W**	**0.500 W (typical)**	**14.73 µW**	**11.58 µW**
* CMOS (Internally Optimized for TTL)				† 0.65-nm CMOS	

7. SUMMARY AND DISCUSSION

The memory-based quaternary comparator seems to be a good candidate for applications with rigorous size, speed, and power requirements. It provides the magnitude comparator function with high speed and low power. Other truth-table operations can be designed with the same approach. The work demonstrates the modularity of the design. An important characteristic is the use of mature CMOS fabrication technology for the CMOS driver-array structures.

Future reports on converter design and other functional systems are underway and future work is planned for the design and implementation for a quaternary comparator. Other approaches are possible in which a driver-array for the quaternary words are combined with an initial L/E/G input would allow modular cascade comparators to be assembled. These memory-based approaches for the comparator are not optimized. By reusing circuitry and adding advanced features, the transistor count can be reduced [18, 22]. Other functions could be combined with the comparator operation. Complete quaternary systems, as well as hybrid systems, can be developed for embedded applications.

ACKNOWLEDGEMENTS

The authors thank Benjamin Cooper of Core Memory Circuits and Indira P. Dugganapally of Intel for discussions.

REFERENCES

[1] Cavin, R. K., "Science and Engineering Beyond Moore's Law," *Proceedings of the IEEE,* 100 (Special Centennial), 1720-1749 (2012).

[2] Smith, K. C., "Multiple-Valued Logic: A Tutorial and Appreciation," *Computer*, **21**(4), 17-27 (1988).

[3] Smith, K. C., "The Prospects for Multivalued Logic: A Technology and Applications View," *IEEE Transactions on Computers*, **30**(9), 619-634 (1981).

[4] Hurst, S. L., "Two Decades of Multiple-valued Logic-An Invited Tutorial," 18th International Symposium on Multiple-Valued Logic, Palma de Mallorca, Spain, 24-26 May 1988.

[5] Kelly, P. M., McGinnity, T. M., Maguire, L. P., and McDaid, L., "Exploiting Binary Functionality in Quaternary Look-up Tables for increased Functional Density in Multiple-valued Logic FPGAs," *Electronics Letters*, **41**(6), 300-302, (2005).

[6] da Silva, R. C. G., Boudinov, H., and Carro, L., "A Novel Voltage-Mode CMOS Quaternary Logic Design," *IEEE Transaction on Electron Devices*, **53**(6), 1480-1483 (2006).

[7] Dornajafi, M., Watkins, S. E., Cooper, B., and Bales, M. R., "Performance of a Quaternary Logic Design," IEEE Region 5 Technical Conference, Kansas City, Missouri, 18-19 April 2008.

[8] Cooper, B. J., [*Device and Method for Enabling Multi-valued Digital Computation and Control*], U.S. Patent No. 8,513,975 (Issued 20 August 2013).

[9] Cooper, B. J., [*Device and Method for Enabling Multi-valued Digital Computation*], U.S. Patent No. 8,593,875 (Issued 26 November 2013).

[10] Dugganapally, I. P., Watkins, S. E., and Cooper, B., "Multi-level, Memory-based Logic using CMOS Technology," IEEE Computer Society Annual Symposium on VLSI - ISVLSI, Tampa, Florida, 9-11 July 2014.

[11] Mazidi, M. A., Mazidi, J. G., and McKinlay, R. D., [The 8051 Microcontroller and Embedded Systems Using Assembly and C, Second Ed.], Prentice Hall, Upper Saddle River NJ, (2005).

[12] Heath, S., [Embedded Systems Design. EDN series for design engineers, Second Ed.], Newnes, Oxford and Boston, 2 (2002).

[13] Cheng, S. W., "A High-speed Magnitude Comparator with Small Transistor Count," Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 1168- 1171, (2003).

[14] Stine, J. E. and Schulte, M. J., "A Combined Two's Complement and Floating-point Comparator," IEEE International Symposium on Circuits and Systems (ISCAS), 89-92, (2005).

[15] Chandrakasan, A. P. and Brodersen, R. W., [Low Power Digital CMOS Design], Kluwer, Norwell MA, (1995)

[16] Veeramachaneni, S.,Krishna, M. K., Avinash, L., Reddy, P. S., and Srinivas, M. B., "Efficient Design of 32-bit Comparator using Carry Look-ahead Logic," IEEE Northeast Workshop on Circuits and Systems (NEWCAS), 867-870, Aug. 5-8, 2007.

[17] Krishna, V. S., Arun, S., and Mazher Iqbal, J. L., "Embedded System based Air Pollution Detection in Vehicles**,"** International Journal of Emerging Technologies in Computational and Applied Sciences, 12(3), 286-289, (2015).

[18] Dugganapally, I. P., [Cormem Digital Reasoning Architecture using CMOS] Ph.D. Dissertation, Missouri University of Science and Technology, (2015).

[19] Fairchild Semiconductor, "TinyLogic UHS NC7SZ04, NC7SZ08, NC7SZ11, NC7SZ32, NC7SZ332, NC7SZ00, NC7SZ0410," (2016). Available: http://www.fairchildsemi.com/products/logic/gates/.

[20] Givone, D. G., "Logic Design with MSI Components and Programmable Logic Devices," [Digital Principles and Design, 1st ed.], McGraw-Hill, New York, Chap. 5 246-248, (2003).

[21] Texas Instruments, "CD4063B 4-Bit Magnitude Comparators," (2016). Available: http://www.ti.com/product/CD4063B/technicaldocuments/.

[22] Cooper, B., [Personal Communication], Core Memory Circuits, (2016).