# Redundant Adders Consume Less Energy

Smitha.K.G
School of Computer Engineering
Nanyang Technological University
Singapore
kava0002@ntu.edu.sg

Hossam.A.H.Fahmy
School of Computer Engineering
Nanyang Technological University
Singapore
hossamfahmy@ntu.edu.sg

A.P.Vinod
School of Computer Engineering
Nanyang Technological University
Singapore
asvinod@ntu.edu.sg

***Abstract*— We conduct a complete analysis of the effect of digit redundancy in adders on their delay, power, energy, and energy-delay product. To our knowledge, this is the first such detailed analysis.**

**We discuss the hybrid signed digit representations that offer a continuum of choices from two's complement representation on one extreme, all the way to a fully signed digit representation on the other extreme. Power and time delay reductions are achieved as a result of algorithmic level changes. Our analysis using TSMC 1.8$\mu$m technology indicates that the increment in power over the whole range from two's complement to fully signed representation is relatively small ($52.174\%$), while the reduction in speed is much larger ($95.455\%$). The best designs from the energy and energy-delay product points of view are the most redundant.**

**We also present a new Modified Hybrid Signed Digit(MHSD) adder that leads to greater improvements. Compared to the Hybrid Signed Digit(HSD) adder [1], MHSD adder shows power decrement of $1.653\%$ and speed increment of $17.716\%$.**

***Keywords*—Redundant representations, Signed Digit Numbers, Hybrid Signed Digit Adder, Energy Delay Product.**

## I. INTRODUCTION

Redundant representations play an important role in high speed computer arithmetic since they provide what is known as carry free addition. The redundant representations limit the carry propagation to a rather short range between adjacent digits and hence decrease the number of inputs on which any output digit depends. With no long carry propagation, the redundant addition is much faster since the operations are performed in parallel for all digits of a number simultaneously.

In the pure binary signed digit (SD) number system, each digit assumes one of three values $\{-1, 0, 1\}$. As a result, a redundancy is introduced in the number system, i.e. it is possible to represent the same number in more than one way. For example, with the notation $\bar{1} = -1$, the number 3 is represented as either $011$ or $10\bar{1}$. Circuits exploit this redundancy to limit the length of carry propagation chains to only one digit position. An adder thus adds two numbers in a fixed time irrespective of the word length.

To have a redundant representation, it is not necessary to make each digit signed. It is possible to combine both conventional binary and signed digits to yield a hybrid form of a redundant representation. Adders with such hybrid inputs propagate the carry in the unsigned (conventional) binary portion till the next higher signed (redundant) digit location. Depending on the choice of the redundant digits locations within the number, a designer is able to tune the speed performance of the adder.

However, the speedup in addition time with a redundant number representation comes with its own cost. The circuit area of redundant adders is generally larger since two bits are needed to represent a single binary signed digit. Also, the basic adder cell that adds two signed digits and an input signed carry to produce a signed digit and a signed carry is more complex than a full adder for unsigned digits. The comparison of power consumption in redundant and conventional adders is slightly more complicated. Due to the increased number of inputs and outputs in redundant adders, the number of nodes that might switch and consume power increases. On the other hand, the long carry chains that exist in conventional binary adders and where a high switching activity occurs do not exist in redundant adders. Actually, the detailed analysis of the use of signed digits presented here reveals a continuum of possible realizations that trade off area and power for a faster addition.

Redundant representations are useful in practice when power consumption is a major concern and when the worst case acceptable delay is predetermined. In this study, we analyze the performance of hybrid adders and study the effect of changing the amount of redundancy in the number representation. The amount of redundancy is identified by the distance between two signed digits. As the distance the distance decreases, a higher redundancy exists and we get a faster adder.

The evolution of mobile computing favors low power and low energy consumption. The exploding market of portable electronic appliances fuels the demand for complex integrated circuits powered by light weight batteries. To compare different designs on their power consumption performance, we need a metric that allows us to determine the most efficient one. The first choice for such a metric is obviously to measure the consumed power in each design and choose the one with the least power. However, the use of power as the metric is problematic. The largest contribution to power consumption in CMOS circuits is due to the nodes that change their values every clock cycle (known as dynamic power consumption). We can always reduce the power consumed by a certain design by reducing the operating frequency. This results in a low power system but

1–4244–0387–1/06/$20.00 © 2006 IEEE 

with an immense delay which is not very useful.

The second natural choice is to use the total energy consumed in the operation as a metric. This is an improvement over power because running the part slower does not directly change the energy used in an operation, it simply spreads the same energy use over a longer time. This metric has its own problem however. The energy of an operation decreases by the reduction of the supply voltage since the energy is roughly $nCV^2$, where $nC$ is the total circuit capacitance multiplied by the number of transitions needed to complete the operation. However, a lower supply voltage affects the switching speed of transistors and dramatically increases the delay of the operation. Thus, the lowest energy solution also runs very slowly.

To avoid these problems, we prefer the use of the energy-delay product (EDP) proposed by Horowitz *et al.* [2]. A smaller EDP value implies a lower energy solution at the same level of performance, i.e. a more energy efficient design. If the area of the circuit is of large importance then the EDP multiplied by the area would be an even better metric.

For completeness, we present in this study a combined analysis of speed, power, area, energy and EDP trade offs with the change of the amount of redundancy in the adder.

We start first by explaining the functionality of a general hybrid signed digit (HSD) adder in section II. We then propose a modified hybrid signed digit (MHSD) adder in section III with a higher performance. Section IV provides the detailed results of our analysis while section V presents our conclusions. The results indicate that our proposed MHSD adder has an improved performance compared to the traditional HSD adder.

## II. Hybrid Signed Digit Adders

The fully SD and the regular two's complement number representations are at two extremes of the spectrum of the hybrid signed digits. In an intermediate HSD, the maximum carry propagation length can be set to any desired value between one and the full word length. Phatak *et al.* proposed such a hybrid system adder [1] and later analyzed its area and power consumption [3].

The area required decreases as the length of the carry propagation chain increases. Phatak *et al.* [3] simulated layouts of basic adder cells to obtain the average number of transitions and power consumption per addition for a word length of 2. Those adder cells were named as HSD-1 adder (HSD adder with distance between two consecutive signed digits (d), where $d = 1$). In the analysis [3], they assumed that for word lengths beyond 2 digits, the power consumption is estimated by extrapolating the results of two cascaded HSD-1 adders. To a first degree, this is a valid assumption but one cannot rely on it to get robust results or to compare various designs. In our analysis, we use state of the art CAD tools (synopsis) with TSMC 1.8$\mu$m technology to estimate the delay, area, and power consumption of each design. Our input to the synopsis tool for each design is a circuit description at the gate level to achieve a higher degree of fidelity in our simulations. Average power dissipation is obtained by calculating the number of transitions occurred for random input test vectors.

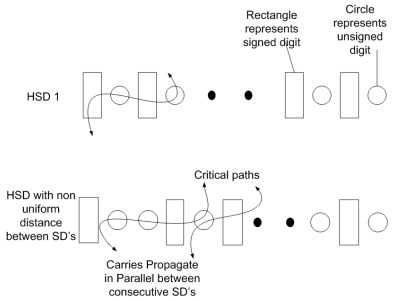


Fig. 1. HSD formats illustration

To introduce the discussion of our modified adder, we present here the radix-2 HSD-1 adder originally proposed by Phatak [1]. For radix-2 the signed digits take any value in the set $\{-1, 0, 1\}$. The unsigned digits are normal bits and assume values in $\{0, 1\}$. The addition procedure presented in [4] enables a signed digit to stop an incoming carry from propagating further. Consequently, the carries propagate between the signed digits and the maximum length of carry propagation equals the distance between the signed digits.

For instance, let $x_i$ and $y_i$ be radix-2 signed digits to be added at the $i^{th}$digit position, and $c_{i-1}$ be the carry into the $i^{th}$ digit position. Each of these variables takes any of the three values $\{-1, 0, 1\}$. Hence this sum is represented in terms of a signed output $z_i$ and a signed carry $c_i$ as $x_i + y_i + c_{i-1} = 2 * c_i + z_i$ where $c_i, z_i \in \{-1, 0, 1\}$. For the unsigned $(i+1)^{th}$ digit position, $a_{i+1}$ and $b_{i+1}$ are the bits to be added and $a_{i+1}, b_{i+1} \in \{0, 1\}$. The carry into the $(i+1)^{th}$ position originates in the signed $i^{th}$ digit position and is signed itself ($c_i \in \{-1, 0, 1\}$). The output digit $s_{i+1}$ is restricted to be unsigned, *i.e.* $s_{i+1} \in \{0, 1\}$. Hence the carry out of the $(i+1)^{th}$ position must be allowed to assume the value -1 as well. In particular, if $a_{i+1} = b_{i+1} = 0$ and $c_i = -1$ then $c_{i+1} = -1$ and $s_{i+1} = 1$ else $a_{i+1} + b_{i+1} + c_i = 2 * c_{i+1} + s_{i+1}$ where $c_{i+1}, s_{i+1} \geq 0$. All the carry propagation chains between the signed digit positions are executed simultaneously. The HSD formats with limited carry propagation chains are illustrated in Figure 1.

## III. MHSD adder Implementation

We now present our modification of the original HSD adder. The SD adder used is based on the work of Kuninobu *et al.* [4]. The unsigned adder is a ripple carry adder (RCA). Interface circuits are used to have proper carry propagation through the signed and unsigned adders.

### A. Additions of bits at unsigned position

A radix-2 ripple carry adder (RCA) is used in the unsigned digit position. For radix-2, the unsigned digits take

any value in $\{0,1\}$. Inputs to the $(i-1)^{th}$ cell are the bits $a_{i-1}$, $b_{i-1}$, and the incoming carry $c_{i-2}$. The outputs are the carry out $c_{i-1}$ and the final sum $s_{i-1}$. The output digit $s_{i-1}$ and carries $c_{i-1}$ and $c_{i-2}$ are unsigned and take one of the two values 0 or 1. The unsigned carry and sum are obtained as $c_{i-1} = a_{i-1}b_{i-1} + c_{i-2}(a_{i-1} + b_{i-1})$ and $s_{i-1} = a_{i-1} \oplus b_{i-1} \oplus c_{i-2}$ respectively.

### B. Additions of bits at signed position

Inputs to the signed adder cell are the bits $x_i$, $y_i$ and the incoming carry signal $c_{i-1}$. The incoming carry $c_{i-1} \in \{0,1\}$. A signed digit $x_i$ is encoded in two bits $x_i^s x_i^a$, with, $-1, 0, 1$ encoded by 11, 00, and 01, respectively. The output of the cell is a signed output represented as $z_i^s, z_i^a$ and signed carry digit $c_i^s$.

We derive two conditions for the carry depending on the values of $x_i$ and $y_i$:

1) if both $x_i$ and $y_i$ are non negative, the carry $c_i^s \in \{0,1\}$
2) at least $x_i$ or $y_i$ is negative, the carry $c_i^s \in \{-1,0\}$

Let $w_i = 0$ when the first condition is satisfied and $w_i = 1$ when the second condition is satisfied. Then, the variable $v_i$ defined by $v_i = w_i + c_i^s$ is always non-negative. In effect, the carry $c_i^s$ is expressed as the difference of two bits $v_i$ and $w_i$. From the above assumptions we formulate the equations of a signed adder.

$$w_i = x_i^s + y_i^s \quad (1)$$

$$s_i^a = x_i^a \oplus y_i^a \quad (2)$$

$$z_i^s = \overline{\overline{s_i^a} + v_{i-1}} \quad (3)$$

$$z_i^a = s_i^a \oplus v_{i-1} \quad (4)$$

$$v_i = \overline{x_i^s y_i^s + \overline{x_i^a y_i^a}} \quad (5)$$

### C. MHSD adder

The RCA's carry-out, $c_{i-1} \in \{0,1\}$ forces $w_{i-1}$ of SD adder to be always zero. The carry-in to the SD cell can be directly connected to $v_{i-1}$ as $w_{i-1} = 0$. The SD adder produces a signed carry of $\bar{1}$ when any of its inputs is $\bar{1}$. Instead of passing this signed carry to the next unsigned adder, we take out that $\bar{1}$ to the output. Nonnegative values of $c_i^s$ are passed to the next unsigned adder's carry-in $c_i$ through an interface logic, where $c_i = v_i\overline{w_i}$

When a carry of $\bar{1}$ occurs ($v_i = 0$ and $w_i = 1$) then that value is taken to the output as a negatively valued bit. The resultant negative bit $wout(NL) = w_i\overline{v_i}$ has the mathematical weight of a carry. This operation of taking the negatively valued bit to the output, results in a variable bit output with respect to the number of SD adders in the design. The net result remain unchanged but transforms from one redundant output to another redundant form. According to Kornerup [5], conversion from one redundant representation to another redundant form takes place in constant time. Such conversions take place in true parallel, but in general through more than one level because conversion in one position will influence neighbour positions. Hence the above conversion can be minimized by exploiting its redundancy.

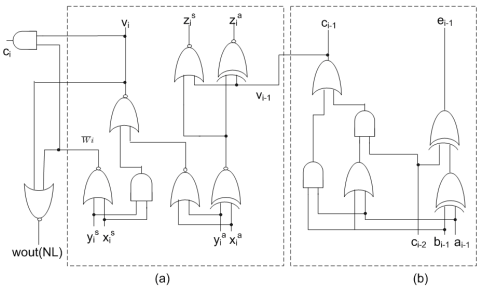


Fig. 2. MHSD-1 adder. Redundant binary adder cell (a) signed digit position (b) unsigned digit position

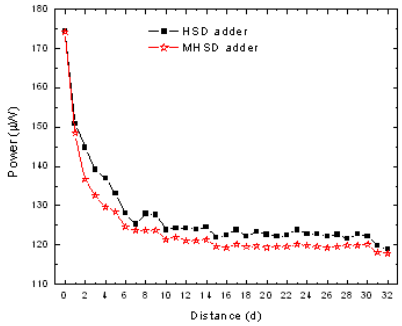


Fig. 3. Power dissipation as a function of the distance $d$ between two consecutive SD's for 32 bit adder

Figure 2 shows the complete representation of HSD-1 adder in radix-2 with interface circuitry. The logic diagram of two adjacent digit positions requires only 62 transistors in MHSD adder design, instead of 74 that are needed for cells in [1].

## IV. Analysis

The delay, power, energy, EDP and area*EDP consumed by the MHSD adder is compared with the HSD adder. Results are obtained for different fixed distance between signed digits 'd'.

An adder with $d = 0$ is a fully signed adder, while $d = 1$ implies that there is a SD adder at every odd place. Similarly, $d = 31$ indicates that the SD adder is in the $32^{nd}$ place. An adder with $d = 32$ is a fully unsigned RCA. Figure 3 shows that there is a reduction in power for the MHSD adder design in comparison with HSD adder. This is mainly due to the fact that the switching activity is reduced as the carry in to the signed digit is made unsigned. As $d$ is increased from 16 to 31, the position of the signed digit shifts from the $17^{th}$ to the $32^{nd}$ place. However, the total number of signed digits and the total number of unsigned digits, remain 1 and 31, respectively, for $d$ in this range. Hence,the power consumption and area is nearly constant for all $d$ values in the range from 16 to 31. The critical path, on the other hand, increases linearly with the longest distance between signed digits as illustrated in Figure 4. The delay of the MHSD adder is less than that of the HSD adder as seen in Figure 4.

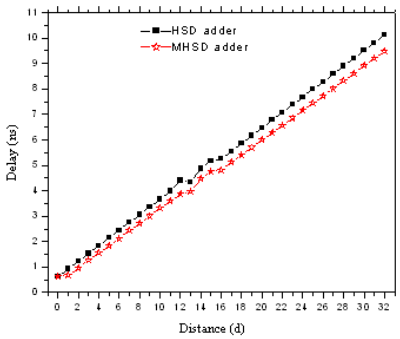


Fig. 4. Critical path delay versus the maximum distance $d$ between two consecutive SD's for 32 bit adder

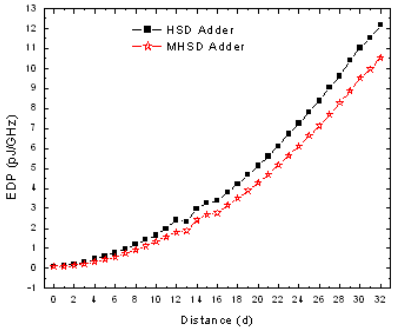


Fig. 6. EDP as a function of the distance $d$ between two consecutive SD's for 32 bit adder

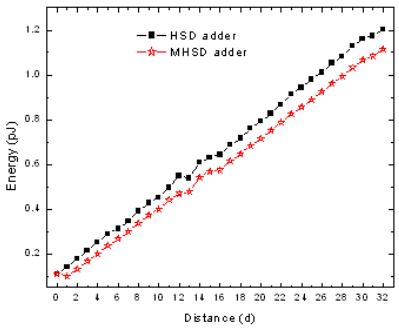


Fig. 5. Energy as a function of the distance $d$ between two consecutive SD's for 32 bit adder

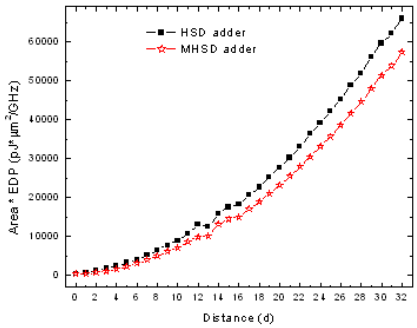


Fig. 7. Area*EDP as a function of the distance $d$ between two consecutive SD's for 32 bit adder

It is clear from Figure 3 that, compared to HSD, our proposed MHSD adder has a lower power consumption. In general, the delay analysis indicates that a fully redundant representation has the minimum delay due to the carry free addition property.

Figure 5 shows the energy as a function of $d$ while Fig. 6 shows the EDP as a function of $d$. The MHSD adder has a better performance in both the energy and EDP metrics.

If the area of the circuit is of large importance then the EDP multiplied by the area would be an even better metric. Area*EDP analysis shown in Figure 7 indicates that for an area constraint hardware. We can have HSD adders with lower EDP compared to unsigned adder and lesser area compared to fully SD adder.

## V. Conclusion

We have presented a complete analysis of the consumed power, delay, energy, EDP, and area*EDP for redundant adders as function of the distance $d$ between signed digits. We note that the change in power over the whole range of $d$ is relatively small ($115 \rightarrow 175\mu W$) while the change in delay is much larger ($11 \rightarrow 0.5ns$). Hence, in general, the more redundant the representation the better it is from the energy and EDP point of view.

Compared to the HSD-1 adder, MHSD-1 adder shows power decrement of $1.653\%$ and speed increment of $17.716\%$. The power reduction is achieved as a result of algorithm level change, which restricts the carry to be unsigned while passing through unsigned adder. This leaves room for additional power reduction by employing standard techniques at circuit and layout levels of the design.

Possible future work can be done for converting the variable redundant output to a fixed redundant output depending on the adder size.

## References

[1] D. S. Phatak and I. Koren, "Hybrid signed-digit number systems: A unified framework for redundant number representations with bounded carry propagation chains," *IEEE Transactions on Computers*, vol. 43, pp. 880–891, Aug. 1994.

[2] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design," in *Symposium on Low Power Electronics*, pp. 8–11, Oct. 1994.

[3] D. S. Phatak, S. Kahle, H. Kim, and J.-J. Lue, "Hybrid signed-digit representation for low power arithmetic circuits," in *Proceedings of the Power-Driven Micro-architecture Workshop In Conjunction With ISCA, 98 in Barcelona*, June 1998.

[4] S. Kuninobu, T. Nishiyama, H. Edamatsu, T. Taniguchi, and N. Takagi, "Design of high speed mos multiplier and divider using redundant binary representation," in *Proceedings of the 8th IEEE Symposium on Computer Arithmetic*, pp. 80–86, July 1987.

[5] P. Kornerup, "Digit-set conversions: Generalizations and applications," *IEEE Transactions on Computers*, vol. 43, pp. 622–629, May 1994.