Improved Design of a 4-bit Absolute-Value Detector Using Simplified Chain Carry Adder

Xinhang Dong¹, *, †, Boyuan Jing², b, *, †, Xiang Yang³, c, *, †

¹ School of Information Science and Technology, Shanghai Tech University, Shanghai, Shanghai 201210, China
² School of Electronic and Control Engineering, North China Institute of Aerospace Engineering, Langfang, Hebei 065000, China
³ Henry Samueli School of Engineering, University of California, Irvine, California 92614, United States

* Corresponding author’s e-mail: adongxh@shanghaitech.edu.cn, bjingby@outlook.com, cyxiang10@uci.edu
† These authors contributed equally

Abstract. 4-bit absolute-value detector (AVD), as one of the basic implementations of bit arithmetic with logic circuits, can help grab a better understanding about digital integrated circuits. Conventional 4-bit AVDs scheme in a multi-comparator and multiplexers, or need to consider multiple situations of overflow and carry-in, both of which could make the final circuit to be complex, labyrinthine and inefficient in the meantime. In this paper, a new design of 4-bit AVD is proposed, the topology of which includes a 2’s complement calculator and a specially designed logic circuit known as chain carry adder (CCA). The whole circuit is concise and the critical path is rigorously considered to make it as short as possible. The delay is set to 1.5 times its minimum, which is positively corresponding to the length of the critical path, the energy accordingly reaches its lower limit. Gate sizing and Device Voltage (VDD) optimization are proceeded for the exact purpose of proving that the circuit energy is minimized.

1. Introduction
This paper is in purpose of showing a new design of 4-bit absolute-value detector (AVD), which has a more concise layout and minimum energy consumption.

1.1. Basic concept
Spike-sorting algorithms, as a vital progress that recently made in the neural signal acquisition systems, have gained a lot of interest in the research community. Among all, we choose one of the commonly used spike-detection algorithms known as absolute-value detection, to study and make further improvement to its design.

Due to the many uncertainties of the inputs in the actual situation, the AVDs mentioned in the following context are limited to 4-bit AVDs. In our opinion, 4-bit AVDs are more representative, which takes into account the interaction and interconnection between all inputs and presents the typical characteristics of AVDs in a relatively crisp way. In a more general case, 4-bit AVDs can be easily extended to higher-bits AVDs with more inputs [1].
1.2. Conventional designs
There are already some conventional designs of AVD [2]. In most circumstances, there exist two main types of AVD, also called magnitude comparator with signed inputs, which are known as signed magnitude comparator and bit operation based multiplexer.

1.2.1 Signed magnitude comparator
One of them is the improvement of the unsigned comparator, shown in Fig.1, see p462-463 of [3]. Basically speaking, the 4-bit unsigned magnitude comparator shown as follows is built from a carry-ripple adder and a 2’complementer. The input is A and B while the output is the sign of B-A. Since A and B are unsigned, the sign of B-A can easily show that whether A is smaller than B or not.

The biggest difference that the signed magnitude comparator holds from the circuit of the unsigned comparator is that the signed magnitude comparator introduces a one-bit signal, V, to highlight the overflow situation when subtracting the two inputs, which means the B-A in this case [4]. V is true when the inputs A and B have different signs and the output has a different sign with B. The final output is the sign of the subtraction of the two inputs, with only its most significant bit (MSB) being specially considered. The MSB of the subtraction which is known as S in this circumstance, is obtained by the following formula:

\[ S = N \oplus V. \]  

Here N is the inverse of the result’s original MSB. When V is true, we find that the overflow flips the sign of subtraction.

Same as unsigned magnitude comparator, the sign of the subtraction, also known as S, shows that whether A is smaller or not than B.

1.2.2 Bit operation based multiplexer
In another case, we put the inputs of the AVD to be a subject and a threshold. In this traditional design, the circuit includes an isolated bit operation on the MSB of the subject. MSB should be designed to be 1 or 0, so that the sign of the subject is decided. If MSB is 1, then the subject is negative; if it is 0 then the subject would be positive. In this circumstance, a multiplexer is a traditional choice. Then the absolute value of the input should be extracted, which would include at least 3 inverters and a 1-bit full adder in the process.

The second procedure is to construct a 4-bit comparator. Fig.2 shows a classic design of 1-bit comparator, using complementary pass transistor logic. The problem of this circuit, other than its complexity, is that the output may not be distinct enough to decide whether the output is 0 or 1 without...
pulling up or down. The output could be at a potential just between the high level and the low level. To avoid this flaw, a large amount of redundancy would have to be added to the circuit [5].

In this case, a new design of 4-bit AVD is made to avoid complexity and increase efficiency.

1.3. Design description
The goal of our design is to find a new layout of the 4-bit AVD with minimum energy and without any registers like pipelining. The “energy” refers to total energy that is drawn from the upper high level, which means Device Voltage (VDD) here.

Basically speaking, absolute-value detector as a special design of magnitude comparator, takes signed value as inputs and outputs 0 or 1 to illustrate whether the absolute value of one input is larger than the other. Fig.3 shows the basic diagram of a 4-bit AVD. The inputs are shown in blue as follows. The circuit of AVD, which is shown in black, including an absolute value calculator and a comparator, is to be designed. The output of the circuit is shown in red to give that whether input $A[n]$ is larger or not than the threshold value, known as Thr. If $A[n]$ is larger than Thr, the output should be high, and vice versa.
Our work of design experienced two main phases. In phase 1, we found an architecture that best optimizes the speed-energy goal. We tried several feasible options and figured out the best architecture and circuit style to meet the goal, even mixed the circuit styles. A sketch of the optimized design is made.

In phase 2, the block-level schematic of the designed AVD is obtained and an evaluation of the critical path is made. The gates along the path and the input operands that cause worst-case delay between input and output bias are determined in the process. We also consider the VDD scaling to ensure the least-energy-case is acquired.

2. Absolute-value Detector

2.1. Algorithm of Absolute-value Detector

We compare the input signals with the given threshold. When the input signal threshold is larger than or equal to the given signal threshold, the output result of absolute-value detector is 1. When the input signal threshold is less than the given signal threshold, the output result is 0.

In the process of neural signal transmission, we consider that the direction of input signal and the direction of neural signal transmission are different. Our input digital signal has 4 bits, which is the sign bit $A_7$ and the input signal bits $A_6, A_5, A_4$ from high to low. When the sign bit $A_7$ is 0, it means that the delivery direction is positive (we can understand it as a positive input signal); when it’s 1, it means the direction is opposite (we can understand it as a negative input signal). Therefore, the representation range of our input digital signal is divided into positive and negative parts, from -7 to +7. We need to calculate the absolute value of the input signals. As the absolute value result is used for making comparison in the comparison parts. In response to this problem, we propose two corresponding solutions.

(1) When the input signals are positive, we don’t process the input signals. When the input signals are negative, we convert them to the corresponding positive value by using the 2’s complement method. Then the absolute-value calculation parts output the processed signals ($A'\_n$) and compare them with the given thresholds.

(2) When the input signal is negative, we don’t calculate the absolute value of this input signal. When the input signal is positive, we use the 2’s complement methods. Then it’s compared with a given threshold. The comparison results can be obtained according to the sign of the calculation results.

With the comparative analysis, we choose the second option. Both of these schemes also use the 2’s complement operation method and the second scheme is more advantageous. The reason is that the second scheme uses logic operations to replace the function of the comparator. The second scheme fully improves the utilization of circuit devices through logic operations and reduces the circuit degree of complexity. At the same time, it’s easy to implement in engineering [6].

The algorithm principle of the absolute value detector is as follows.

(1) Determine whether the input signal is positive or negative. We collectively refer to the 4-bit input signal as A. $A'$ is the output result of the 2’s complement operation. We judge the positive or negative of the input signal according to the sign bit $A_7$. For the negative value input signal, output doesn’t make calculation. For the positive value input signal, the 2’s complement calculation is performed to obtain the corresponding negative value.

$$A' = -|A|.$$  \hspace{1cm} (2)

(2) Detect and compare $A'$ with the given thresholds. We collectively refer to the output result as B. When the input signal is a negative value, due to the direct output, a full adder is used to achieve the addition operation with the given value. When the input signal is positive, the full adder implements the subtraction operation by using the 2’s complement method. Here we simplify the circuit through logic operations. Formula (2) shows the calculations.

$$B = T+A' = T - |A|.$$  \hspace{1cm} (3)
2.2. Circuit design of taking 2’s complement

Our input signal is A. From high to low, the sign bit and the input signal bit are $A_3, A_2, A_1, A_0$. The given thresholds are collectively referred to as T. The given threshold bits are $T_3, T_2, T_1$ and $T_0$ from high to low. $A'$ is the result of 2's complement operation of the A. When $A_3$ is equal to 0, then A is a positive value. We take 2's complement operation on it. The logical operation formulas are as follows.

$$A'_0 = A_0.$$  \hspace{1cm} (4)
$$A'_1 = A_1 \oplus A_0.$$ \hspace{1cm} (5)
$$A'_2 = A_2 \oplus (A_1 + A_0).$$ \hspace{1cm} (6)
$$A'_3 = A_3 \oplus (A_2 + A_1 + A_0).$$ \hspace{1cm} (7)

Some examples of calculation results are as follows:

| $A_3$ (Sign bit) | $A_3A_2A_1A_0$ | $A'_3A'_2A'_1A'_0$ |
|-----------------|----------------|------------------|
| 1 (Negative number) | 1011 (-7) | 1011 |
| 0 (Positive number) | 0010 (+2) | 1110 |

When $A_3=1$, the input signal A is a negative value. We don’t take processing with output.

![4-bit input and 2's complement circuit schematic diagram](image-url)
2.3. Circuit design of chain carry adder

We design a chain adder to implement subtraction by using the 2's complement method. For design parameters, the chain adder input has a buffer of one unit, driven by two inverter chains of unit size. The chain adder consists of three full adders, which are connected in series with carry output. For a single full adder, the input signal \( A \), the logical operation result \( B \), \( C_{in} \) and the output is \( C_{out} \). The formula shows the results of a single full adder.

\[
C_{out} = AB + (A \oplus B) C_{in}. \quad (8)
\]

Since we have connected the carry output in series, the output of the chain carry adder is \( B_3 \). As shown in the formula.

\[
B_3 = T_2 A'_2 + (T_2 \oplus A'_2) (T_1 A'_1 + (T_1 \oplus A'_1) (T_0 A'_0)). \quad (9)
\]

The final result output is \( A'_3 + B_3 \) (in this case, \( A'_3 = 1 \)).

2.4. Schematic diagram of Absolute-value Detector

Since we have connected the carry output in series, the output of the chain carry adder is \( B_3 \). As shown in the formula.

\[
B_3 = T_2 A'_2 + (T_2 \oplus A'_2) (T_1 A'_1 + (T_1 \oplus A'_1) (T_0 A'_0)). \quad (9)
\]

The final result output is \( A'_3 + B_3 \) (in this case, \( A'_3 = 1 \)).
3. Key parameters of Absolute-value Detector Circuit

3.1. Critical path delay

The critical path is the logical path with the longest delay from input to output in the circuit diagram. In the logic circuit we designed, the critical path is marked by the red path. As shown in fig.7, the red path is the critical path.

As shown in table 3-1, several known parameters will be used in the calculations of the formulas [7].

| Gate Type               | Number of Inputs |
|------------------------|------------------|
|                        | 1                |
|                        | 2                |
|                        | 3                |
|                        | 4                |
|                        | n                |
| Inverter               | 1                |
| NAND                   | 4/3              |
|                        | 5/3              |
|                        | 6/3              |
|                        | (n+2)/3          |
| NOR                    | 5/3              |
|                        | 7/3              |
|                        | 9/3              |
|                        | (2n+1)/3         |
| Tristate, Multiplexer  | 2                |
|                        | 2                |
|                        | 2                |
|                        | 2                |
| XOR, XNOR              | 4, 4             |
|                        | 6, 12, 6         |
|                        | 8, 16, 16, 8     |

3.1.1. $V_{DD}=1v$, Critical path delay

According to the path logic effort, we calculate the logic effort in the multi-level path [8]. As the following formula shown.

\[ G = \prod g_i = \frac{4 \times 5^3}{3^3}. \] (10)

Path electrical effort is calculated:

\[ H = \frac{C_{out(path)}}{C_{in(path)}} = 32. \] (11)

Path branch effort is calculated:

\[ B = 1. \] (12)

Path effort is calculated:

\[ F = GBH. \] (13)
\[ f = \sqrt{F} = 2.69. \] (14)

Path parasitic delay is calculated:

\[ P = 22 \text{ PS}. \] (15)

Total path delay is calculated:

\[ D = Nf + P = 46.25 \text{ PS}. \] (16)
3.1.2 Critical path gate sizing approach
The $C_i$ of the output bit of the critical path is a 32-bit inverter. We calculate the size of the critical path. According to the formula calculation, the corresponding result can be obtained.

We know the below formula about the path effort $F$ how to calculate:

\[ F = \prod (g \times h) \]  \hspace{1cm} (17)

The stage effort is:

\[ f^* = g \times h \]  \hspace{1cm} (18)

We can now size the gates:

\[ C_{in} = g \times \frac{c_{out}}{f^*} \]  \hspace{1cm} (19)

We get the sizing of the critical path gates from left to right (labelled u1-u9):

- $C_{in u1} = 1.27$;
- $C_{in u2} = 2.05$;
- $C_{in u3} = 4.13$;
- $C_{in u4} = 2.78$;
- $C_{in u5} = 1.87$;
- $C_{in u6} = 3.02$;
- $C_{in u7} = 6.09$;
- $C_{in u8} = 9.83$;
- $C_{in u9} = 19.83$.

3.1.3 $V_{DD}^{opt}$ Calculation
The $V_{DD}^{opt}$ is optimal voltage for the $V_{DD}$ in the range of 0v - 1v. This voltage also satisfies the requirements that the maximum delay is 1.5 minimum delays.

As the following formula shown, we calculate the $V_{DD}^{opt}$ by using $t_p$:

\[ t_p = \frac{CV_{DD}}{K'(V_{DD} - V_T)^2} \]  \hspace{1cm} (20)

According to the situation that the maximum delay is 1.5 minimum delays, we preset $V_T = 0.2v$:

\[ D \propto \frac{V_{DD}}{(V_{DD} - V_T)^2} \]  \hspace{1cm} (21)

\[ V_{DD}^{opt} = 0.7758 \text{ V} \]  \hspace{1cm} (22)

When $V_{DD}^{opt} = 0.7758\text{ V}$, the critical path delay is calculating:

\[ D_1 = 1.5 \times D = 69.38 \text{ PS} \]  \hspace{1cm} (23)

3.2. The overall energy consumption of the circuit

3.2.1 Energy consumption when $V_{DD} = 1\text{ V}$

\[ E = C_iV_{DD}^2 = 32.00 \]  \hspace{1cm} (24)

3.2.2 The overall energy consumption of the circuit

\[ E = C_iV_{DD}^{opt^2} = 19.26 \]  \hspace{1cm} (25)

3.3. Summary of key circuit parameters
When the $V_{DD}$ changes from 0V to 1V, we consider reducing energy consumption. We plan to recalculate the critical delay and energy consumption. After optimization calculations, we find that the energy consumption when $V_{DD}$ is 1V is 39.81% lower than the energy consumption when $V_{DD}^{opt}$ is 0.7758V. At the same time, the delay is 1.5 times the previous $V_{DD} = 1\text{ V}$ delay. According to the analyses of experimental results, the energy consumption of $V_{DD}^{opt}$ at this time is the smallest energy consumption in the range of 0V to 1V.
In the calculation above we can also see that we can have a better energy performance while its delay is not being increased dramatically [9] as an individual static logic circuit [10], and this is an essential feature considering its usage in reversible neural signal processing the Arithmetic and Logic Units (ALUs) [11].

In addition to our conclusion above, since the change of dynamic energy in Complementary Metal Oxide Semiconductor (CMOS) is in proportion to its operation temperature and in inverse proportion to VDD, we conclude that for our AVD being set to a lower VDD, its energy changes more significantly with its change in operating temperature when being part of a dynamic circuit [12].

4. Conclusion

In this paper, a static-logic 4-bit AVD has been designed, implemented, and optimized with respect to the worst-case propagation delay and the total energy drawn from VDD. This new AVD is designed with the minimum energy for which the delay is 1.5 times the minimum delay, and both its delay and energy outperform traditional comparator based AVDs due to its relatively shorter critical path and replacement of comparator by the four carry bits of adders. We have only used 22 logic gates and an inverter in total, avoiding those high-fan-in gates like four-input nor gate.

Aside from the improvements we have made, there are still some flaws existing in our design. The XOR gate we used a lot in our design consumed the most energy and contributed a large part of the delay. What’s more, the performance could be improved by mirror adder if we use it at some point on the critical path.

References

[1] S. Iranmanesh, G. Raikos, Z. Jiang and E. Rodriguez-Villegas, "CMOS implementation of a low power absolute value comparator circuit," 2016 14th IEEE International New Circuits and Systems Conference (NEWCAS), 2016, pp. 1-4.

[2] S. Lubaba, K. M. Faisal, M. S. Islam and M. Hasan, "Design of a Two-Bit Magnitude Comparator Based on Pass Transistor, Transmission Gate and Conventional Static CMOS Logic," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-5.

[3] Neil H.E. Weste, David Money Harris. CMOS VLSI Design A Circuits and Systems Perspective Fourth ed.PEARSON.pp.156

[4] H. Fukuda, "Signed digit CMOS (SD-CMOS) logic circuits with static operation," Proceedings. 34th International Symposium on Multiple-Valued Logic, 2004, pp. 128-134.

[5] J. M. Rabeay, Digital Integrated Circuits: A Design Perspective, scent ed., Upper Saddle River N. J.: Prentice Hall, c 1996.

[6] X. Lin, Y. Wang, S. Nazarian and M. Pedram, "An improved logical effort model and framework applied to optimal sizing of circuits operating in multiple supply voltage regimes," Fifteenth International Symposium on Quality Electronic Design, 2014, pp. 249-256.

[7] R. M. Anacan and J. L. Bagay, "Logical Effort Analysis of various VLSI design algorithms," 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2015, pp. 19-23.

[8] Jingyue Xue, D. Al-Khalili and C. N. Rozon, "A normalized intrinsic delay model of static CMOS complex gates for deep submicron technologies," The 2nd Annual IEEE Northeast Workshop on Circuits and Systems, 2004. NEWCAS 2004., 2004, pp. 17-20.

[9] H. Q. Dao, B. R. Zeydel and V. G. Oklobdzija, "Energy minimization method for optimal energy-delay extraction," ESSCIRC 2004 - 29th European Solid-State Circuits Conference (IEEE Cat. No.03EX705), 2003, pp. 177-180.

[10] K. Subramanyam, S. Shaik and R. Vaddi, "Tunnel FET based low voltage static vs dynamic logic families for energy efficiency," 18th International Symposium on VLSI Design and Test, 2014, pp. 1-2.

[11] J. B. Chacko and P. Whig, "Low Delay Based Full Adder/Subtractor by MIG and COG Reversible
Logic Gate," 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN), 2016, pp. 585-589.

[12] P. Kocanda and A. Kos, "Static and dynamic energy losses vs. temperature in different CMOS technologies," 2015 22nd International Conference Mixed Design of Integrated Circuits & Systems (MIXDES), 2015, pp. 446-449.