Shang D, Xu Y, Gao K, Xia F, Yakovlev A. 


Copyright:

© 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

DOI link to paper:

http://dx.doi.org/10.1109/DDECS.2016.7482476

Date deposited:

30/09/2016
Low Power Voltage Sensing Through Capacitance to Digital Conversion

Delong Shang, Yuqing Xu, Kaiyuan Gao, Fei Xia, and Alex Yakovlev
LPEM Lab, uSystems Group, School of EEE
Newcastle University
Newcastle upon Tyne, NE1 7RU, UK
{delong.shang,y.xu23,k.gao,fei.xia,alex.yakovlev}@newcastle.ac.uk

Abstract—Capacitance sensors are widely used for sensing physical parameters. Conventional capacitance to digital methods use complex analog ADC techniques which are power hungry. Recently a fully digital solution was proposed with improved power consumption. This paper describes a number of problems in that solution, analyzes these problems, and proposes a new design free of these problems. A voltage sensor as an example was designed based on the proposed capacitance to digital conversion in this paper. The new method achieves the same accuracy with less than half the circuit size, and 25% and 33% savings on power and energy consumption.

Keywords—Event-Driven; Voltage Sensor; Capacitance to Digital; Wireless Sensor Application; Metastability; Error Propagation; Asynchronous; Accuracy; Low Power;

I. INTRODUCTION

Sensors are essential for many applications and have been a very popular area of research [12][13]. One of the main types of sensors is capacitance sensors, which can be used to measure various physical quantities, including position, pressure, concentration of certain chemicals, etc. This is based on the fact that these physical quantities may be made to charge a capacitor so that this charge reflects the value of the parameter. Typical capacitance to digital converters (CDCs) use charge sharing or charge transferring between capacitors to convert the sampled capacitance to voltage. This approach requires complex analogue circuits, such as amplifiers and ADCs, which increase design complexities and often increase power consumption [7]. Integrating capacitive sensors into many applications including small wireless sensor systems is therefore challenging due to the total system power/energy budget, which can be in the range of a few nW [1][2][3][4][5].

Unlike conventional methods, the converter recently reported in [7] is a fully digital solution using iterative delay chain discharge and promises reduced complexity, measurement time, power/energy, etc.

However, in this paper we will show that the solution found in [7] has potential hazards and other problems in operation. This will be discussed in more details in Section III.

The method of converting charges on a capacitor to a digital value through discharging is not new and reference-free voltage sensors have been based on this [11]. These are however irrelevant here because of the prices paid to achieve reference-freedom, which is not of interest in this paper.

The contributions of this paper are 1) presenting the general theory of this type of sensing, 2) analyzing the existing method and implementation solution in [7], 3) systematically deriving a new asynchronous event driven implementation solution with experimental and comparison results based on a voltage sensor.

The remainder of the paper is organized as follows: Section II presents the theory of capacitance to digital methods. Section III clarifies the general algorithm and explains the problems of the existing solution. Section IV presents the new asynchronous solution. Section V gives experimental results and comparative studies. Section VI concludes the paper.

II. GENERAL THEORY OF DIGITAL DISCHARGE CDCS

The iterative delay chain discharge method of [7] is illustrated in Figure 1. Conceptually, a capacitor $C_{\text{sense}}$, whose charge is related to the physical parameter being measured, is discharged to a pre-set reference voltage. And the number of iterations of discharging is related to the initial charge.

![Figure 1. Method used in [7].](image1)

![Figure 2. Discharge process.](image2)

Initially, the voltage across $C_{\text{sense}}$, $V_{\text{sense}}$, is charged to $V_{\text{high}}$ by the physical parameter. In each iteration, $C_{\text{sense}}$ is discharged through an inverter chain powered by $V_{\text{sense}}$, in which a trigger signal passes through. At the same time, the trigger signal also passes through another inverter chain powered by a reference voltage ($V_{\text{low}}$) with the same number of inverters. Signal$_{\text{H}}$ and Signal$_{\text{L}}$ are the outputs of the two
inverter chains respectively. The delays of the inverter chains are related to their Vdds. The higher the Vdd, the smaller the delay. The two delays can be compared to detect whether Vsense has been discharged to Vlow. If Signal(H) comes earlier than Signal(L), Vsense has not been discharged to Vlow. As a result, more discharging is required. So, the counter is incremented by 1, the trigger signal is toggled, and another iteration of discharge will be performed. Otherwise, the voltage of Csense has been discharged to Vlow. The iterations stop, and the value of the counter is used to represent the value of Vhigh and hence the physical parameter being measured.

Figure 2 shows the capacitance to digital discharge process. At step i, Qi will be Csense*Vi, and at step i+1 Qi+1 will be Csense*Vi+1 + Cp*Vi+1, where Cp is the capacitance of the circuit (i.e. the inverter chain) onto which the charge on Csense is discharged.

As Qi=Qi+1, Csense*Vi = (Csense+Cp)*Vi+1, we can derive the following formula:

\[ \frac{V_{i+1}}{V_i} = \frac{C_{sense}}{C_{sense}+C_p} \]

and assuming Vi+1 = Vi*(1-k), then

\[ k = \frac{C_p}{C_{sense}+C_p} \]

We can then model the discharge process by using the equation:

\[ V_{low} = V_{high}(1-k)^n \]

where n is the number of steps taken to discharge Vsense from Vhigh to Vlow, and can be used to represent Vhigh.

Based on the Taylor series,

\[ (1-k)^n = 1 - nk + \frac{n(n-1)k^2}{2!} - \frac{n(n-1)(n-2)k^3}{3!} + \ldots \]

and, if nk<<1, the above formula can be approximated as:

\[ (1-k)^n \approx 1 - nk \]

From (3), if we have V_{high} and V_{low} fixed (i.e. const) by the measurement method, we must have (1-k)^n=const. Thus, under nk<<1, we have 1-nk=const and thus nk=const, and hence

\[ \frac{n}{C_{sense}+C_p} = \text{const} \]

So, if C_{sense}>>C_p, then C_{sense}+C_p=C_{sense}. We will have

\[ \frac{n}{C_{sense}} = \text{const} \]

and thus n must be linearly proportional to C_{sense} with any fixed delay line

This means that for a fixed V_{high} and V_{low}, the value of n can be used to derive the value of a tested capacitor, and the same reason, for a fixed value of capacitor and a fixed V_{low}, the n can also be used to derive the V_{high} to form a voltage sensor. Existing methods concentrate on the detection of capacitance values. In this paper, we illustrate our method with an example of a voltage sensing solution.

III. EXISTING SOLUTION: METHODOLOGY & PROBLEMS

[7] presented a full digital implementation solution of the above algorithm. The key part of the implementation is shown in Figure 3, in which there are three delay comparators, three counters, and one clock generator, etc.

The implementation managed to achieve good power figures. However, we discovered three major problems that affect its performance and potentially even its correctness.

**Problem 1: Extra Power/Energy for correction.** The counting result has to be corrected by formula (7) in which one shifting, and two subtraction operations are needed. This means extra computation, and consumes extra power/energy.

\[ D_{OUT} = 2 \times D_{MAIN} - (D_{SUB1} + D_{SUB2}) \]  

Figure 3. Key part of the implementation proposed in [7]

**Problem 2: Accuracy.** This method may introduce errors in the final result. The dither occurs when Vsense is discharged to approach (cross) Vlow. In principle, the basic concept in Figure 1 only requires a single counter to keep track of how many discharging iterations have happened by the time V_{high} goes downto V_{low}. However, the design in [7] has an overly complicated method of detecting the point of crossing by introducing extra delays and using a delay comparator (comparator 1) as the completion detection as shown in Figure 3. The logic of the delay comparator is shown in Figure 4(a), in which Signal(H) and Signal(L) are inputs of the delay comparator. Three separate comparators are used to determine the crossing of rising and falling edges between Signal(H) and Signal(L) as well as to detect the final finishing of the conversion. And three counters maintain the data needed for the final correction based on the formula (7).

**Problem 3: Metastability.** This problem may occur in a delay comparator, where both P and O, which are internal signals of a delay comparator, may happen if Signal(H) and Signal(L) come very closely in time (Figure 5(b)), as illustrated in Figure 6. In this metastability state, both P and O, which are internal signals of a delay comparator as shown in Figure 4(a), have non-digital values. Theoretically the duration of this state is non-deterministic before the
metastability settles, and the values of P and O after settling are also non-deterministic [6]. This will affect the final result. For instance, if during one sensing of V\text{sense} or C\text{sense}, Signal(H) is close to Signal(L) twice for both falling and rising edge detection, without metastability the SUB counters could accumulate 4 more counts in total than with metastability. The final result, with and without metastability, can differ by as much as 4.

**Figure 6. Metastability.**

**Problem 3: Error Propagation.** The mechanism used in [7] forms a closed loop for detecting rising and falling edges of the Clk signal, and updates Clk working as a clock generator. Even when metastability happens, the Clk signal will be updated only after the metastability settles down so the SUB counters are safe. However the completion comparator, also subject to metastability, is not included in a closed loop. And this metastability potentially can be propagated out to damage the D\text{MAIN} as illustrated in Figure 4(b). The Finish signal is generated just after the Clk signal is set up, Clk will be withdrawn immediately based on the implementation. Since this means a non-digital value or hazard pulse on the clock of the counter, the counter’s value may become totally corrupt.

**Figure 7 Potential error propagation**

This is illustrated in Figure 7. When the detection completion signal suffers from no metastability, or if any metastability settles down before action happens in Clk, the completion is normal. However, when this metastability coincides with Clk the counter value is not trustworthy.

Neither problem 2 or 3, directly related to metastability, can be reliably discovered or verified through simulations or hardware testing, the methods relied on by [7]. But they can be reasoned about through state-space analyses, as in the discussions above [15]. More importantly, they can be eliminated through systematic asynchronous logic synthesis methods [6], as shown below.

**IV. PROPOSED NEW SOLUTION**

The solution proposed in this paper is based on the algorithm shown in Section II. We slightly simplify the algorithm in implementation. The key issues of the proposed solution are to overcome the problems listed in Section III, and to reduce power consumption and to generate the results quickly without losing accuracy.

The listed problems are caused mainly by the two reasons: 1. the mechanism to check whether the discharged V\text{sense} is approaching V\text{low}, and 2. the mechanism to terminate the CDC conversion.

To implement the algorithm, there should be the following function blocks: Signal generator, Event generator, Event comparator, and Event counting mechanism. In principle, the signal generator generates a signal from the sensed quantity, and that signal will be the input of the event generator as a trigger to produce two events. These two events are then compared in terms of occurrence orders, and based on the comparison result, either event counting is triggered or the conversion is finished. During all of this the charge on C\text{sense} is discharged.

The algorithm in Figure 1 is refined in STG [14] format as shown in Figure 8. This STG is used as a specification of our proposed solution. As shown in Figure 8, Clk is the signal coming from the signal generator; R\text{a} and R\text{b} are the two events generated by the Clk signal passing through two inverter chains (event generator) under different power domains, and they are used to replace Signal\text{H} and Signal\text{L} respectively. This means that when Clk (trigger) passes through two inverter chains powered by V\text{sense} and V\text{low}, R\text{a} and R\text{b} will be generated. V\text{low} is the pre-set reference voltage, and V\text{sense} is the voltage that we are going to measure if the C\text{sense} is fixed and/or V\text{sense} is a pre-set voltage level if measuring C\text{sense}. As discussed in Section II, each of C\text{sense} and V\text{sense} can be derived if the other one is a known constant. In this study, measuring V\text{sense} is used as an example but the method can equally be applied to capacitance measurement.

The operation of the sensor is described as follows. Firstly the measured voltage is sampled by a known capacitor (for wider applicability and faster response, a small capacitor is preferred). In other words, this capacitor is charged up to V\text{high}. After that, the charge on C\text{sense} is discharged through running an inverter chain by supplying the inverter chain’s Vdd from V\text{sense}. In general V\text{sense} is greater than V\text{low}. This means that event R\text{b} will be generated earlier than event R\text{a}, which is generated by running an inverter chain off V\text{low}. Based on the algorithm, if event R\text{b} is detected before event R\text{a} happens, event R\text{a} is recorded by the event counting mechanism, then the Clk signal is updated for the next discharging iteration. This loop is broken when event R\text{a} is detected to have come earlier than the event R\text{b}. The counting number will be proportional to the level of V\text{high} according the discussion in section II.

**Figure 8. STG specification.**

To check the order of happening of the two events, a time comparator is required. In order to avoid the problems described in Section III encountered by the solution in [7],
which uses delay comparators, here we proposed and designed a new time comparison mechanism.

As shown in the STG specification, an arbitrator is required to arbitrate \( R_a \) and \( R_b \), as shown in the dotted block in Figure 8. If \( R_a \) happens earlier, \( R_a \) will win the arbitration. \( Clk \) will be updated and another iteration of discharging will start (the top branch is fired). Otherwise if \( R_b \) wins, the bottom branch is fired, and the conversion is finished.

Here if \( R_a \) wins, the arbitration should guarantee to block \( R_b \) during each round, and vice versa as represented in the STG. This blocking mechanism is easy to implement in the specification by introducing a token. The winner will have (hold) this token, and as a result the loser cannot progress until the token is released. For example if \( R_a \) wins, the top branch takes the token. Without the token, \( R_b \) cannot be granted and the bottom branch will be blocked.

As shown in the STG specification, firstly \( Clk^+ \) triggers the two events \( R_a \) and \( R_b \), which arrive independently after their respective delays. Then, if \( R_a \) wins the arbitration, a grant signal, \( A_a \), will be generated to completely block event \( R_b \). However \( Clk^- \) (clock reset) will not happen until both \( R_a \) and \( R_b \) have been triggered. This is managed by \( Clk^- \) needing to take a token from the opposite event’s output place. \( Clk^- \), no matter which path it is, effectively withdraws both \( R_a \) and \( R_b \). This makes sure that the loser of one round of arbitration gets cancelled and does not wait for the next round. And the next round of arbitration will be between a new pair of \( R_a \) and \( R_b \) events. The arbitration is only released after both \( R_a \) and \( R_b \) have been withdrawn by one of the \( Clk^- \) signals. Then signal generator (\( Clk^+ \)) will act only after arbitration has been withdrawn. This makes sure that each round of comparison involves only \( R_a \) and \( R_b \) events generated in this round. In this way, this specification completely avoids the extra work needed for the corrections leading to three separate counters found in [7].

If \( R_b \) happens earlier than \( R_a \), \( A_b \) will be generated to grant the \( R_b \) event. The same principle to safeguard correct comparisons apply as above. The only difference is that the top branch pertains to continuing to discharge \( C_{sense} \) and the bottom branch, with the start- signal, ends the discharge and outputs the counter value, and waits for \( start^+ \) which starts a new sensing round. Such actions as ending discharge and outputting the counter value do not need to be explicitly specified and can be understood as included in start-.

From the above discussion, the \( Clk^- \) signal is to be set up, withdrawn, and then set up, withdrawn repeatedly. The process forms an oscillator. As a result, this single signal \( Clk \) can be used to control the cyclic activities needed by the voltage measurement algorithm in Figure 1 and Figure 2.

This method uses \( Clk^+ \) as the valid event trigger. This means that \( Clk^- \) is not a valid event during measurement. From this point of view, the proposed method simplifies the algorithm used in [7] in implementation. We need to test if this affects the measurement resolution. The experimental results presented in the next section will show that measurement resolutions are not affected by this.

An asynchronous implementation synthesized from the STG specification in Figure 7 is shown in Figure 9. The \( Clk \) signal passes through two independent inverter chains in the same way as the implementation in [7] to produce two events \( R_a \) and \( R_b \). These go to a MUTEX [15] to detect who wins the arbitration — this basically resolves which signal comes first, which is what we want to detect. If \( R_a \) comes first, there will be another round of oscillation to discharge \( C_{sense} \) and if \( R_b \) comes first the measurement finishes with the counter value representing \( V_{high} \) if \( C_{sense} \) is constant, or \( C_{sense} \) if \( V_{high} \) is constant.

The MUTEX used here, which is shown in the dotted box of Figure 9, is an arbiter with a metastability resolver inside. As the two events \( R_a \) and \( R_b \) are completely independent in timing, it is inevitable metastability can occur. The metastability resolver in the MUTEX ensures that if metastability happens, it will be resolved inside first before a grant signal is generated. If \( R_a \) comes earlier than \( R_b \), \( R_a \) will win the arbitration. As a result, \( A_a \) will be high. To block \( R_b \), this will update the SR latch (below the MUTEX) in Figure 9. After updating, it will wait for \( R_b \) to go up by using a C element [6] to update \( Clk \). This makes sure that both inverter chains have completed their runs before \( Clk^- \) can happen, as specified in Figure 8. In the meantime it will block the \( R_b \) signal to remove any potential glitches. When both inputs of a C element are 1s, its output is 1, when both inputs are 0s, the output is 0. Otherwise, the previous output value is kept [6]. When \( R_b \) goes up, as the SR latch has been updated, \( Clk \) will be updated to low. The low \( Clk \) signal will pass through the two inverter chains, and when both \( R_a \) and \( R_b \) are low, and \( A_b \) is withdrawn to low, \( Clk \) will be updated to high again. In the implementation, the three NOR2 gates below the MUTEX will be used to realize the control of this mechanism. The above process will be repeated, until \( R_b \) comes earlier, in which case \( A_b \) is granted. It will update the SR latch at the right hand side of Figure 9, and disable the \( Clk \) signal until the round of next sensing for a new \( V_{high} \) or \( C_{sense} \).

The specification in Figure 8 means that metastability can only happen at one point in the entire system, i.e. when trying to resolve whether \( R_a \) or \( R_b \) comes first. Elsewhere there is no contention between independently-timed signals. The design in Figure 9 ensures that when metastability does happen, it never affects the subsequent operations of the sensor.

![Figure 9. Asynchronous Implementation.](image-url)
V. EXPERIMENT, ANALYSIS, AND COMPARISON

Our design has been implemented in UMC 90nm CMOS technology using Cadence toolkits. And for fair comparisons, we ourselves re-implemented the solution proposed in [7] in the same technology, same standard cells, and using the same design flow. This ensures that simulations in the same environment given the same input do illustrate and only illustrate the differences between the two designs. Our experiments are conducted only with the methods used as voltage sensors that measure $V_{\text{high}}$. This does not affect the generality of the results.

Figure 10 shows a simulation result of our design. $V_{\text{high}}$ is 1V, and firstly it is used to charge $C_{\text{sense}}$. This means that $V_{\text{sense}}$ is 1V at the beginning. This simulation result was obtained by using Cadence Spectre analog toolkits (SPICE level).

The rising of the Start signal triggers the discharging of $V_{\text{sense}}$ and the Clk signal starts to be generated as discussed in Section IV. When $V_{\text{sense}}$ drops to $V_{\text{low}}$, the delay of the inverter chain powered by $V_{\text{sense}}$ catches up with the delay of the other inverter chain powered by $V_{\text{low}}$. This scenario may cause metastability (very hard to show in simulations) in the same way as in the solution in [7]. However, for our method this does not translate to operational problems. After that, $A_{\text{b}}$ will be generated to high and the Clk stops oscillation. Then the 20-bit counter calculates the total amount of Clk going up, when $V_{\text{sense}} > V_{\text{low}}$.

Table 1. Comparison results when $V_{\text{sense}}=1\text{V} & V_{\text{low}}=0.45\text{V}$

<table>
<thead>
<tr>
<th></th>
<th>[7]</th>
<th>This Work</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response Time</td>
<td>1.1us</td>
<td>0.9us</td>
<td>12.8%</td>
</tr>
<tr>
<td>Power</td>
<td>26.0uW</td>
<td>20.0uW</td>
<td>24.7%</td>
</tr>
<tr>
<td>Energy</td>
<td>28.7pJ</td>
<td>19.3pJ</td>
<td>32.7%</td>
</tr>
</tbody>
</table>

Figure 12(b) shows the comparison of output code between this work and the work in [7]. The higher the output code, the higher the measurement resolution in any particular case. In general, the output codes of the two methods are almost the same (with this work being slightly higher), which means that they have more or less the same resolution not considering metastability (our simulations mostly cover metastability-free cases). As discussed earlier metastability could reduce the resolution of [7] but not this work.

Table 2 shows the comparison of the number of gates and flip-flops used between this work and [7]. In total, the design in [7] used 258 gates and 52 flip-flops. In this work, the circuits contain 107 gates, which is 59% less and 20 flip-flops, which is 62% less.

Table 3 shows the comparison results of resolution in the voltage range of 0.45V to 1.0V. As can be seen from this table, the resolution of this work behaves better than [7]. In the range of 0.69V to 0.83V, the resolution of this work is 10mV while the resolution of [7] is 20mV. Moreover, the output code in the range of 0.92V to 0.94V for [7] is non-monotonic, which means the output code in this range for that sensor cannot refer to a specific voltage correctly, hence reducing the resolution around that point further. Closer inspection shows that this is because of an occurrence of problem 2 discussed in Section III.
The preliminary layout of the proposed design is shown in the right hand side of Figure 13. It was designed in UMC 90nm CMOS technology. The size of the layout is 0.0015mm² (23.2μm*65μm). And the design was packaged in a PGA package as shown in the left hand side of Figure 13.

**VI. CONCLUSIONS AND THE FUTURE WORK**

This paper presents the general theory of capacitance to digital conversion through discharging, studies the promising solution proposed in [7] and discovers three major problems. To overcome these problems, the cause of these problems, metastability, is identified, and an asynchronous capacitance to digital sensor is systematically designed, and implemented in UMC 90nm CMOS technology.

The proposed implementation solution only detects the up going edges. As a result, the solution simplifies the algorithm used in [7], and a new signal oscillating mechanism is designed by using MUTEX and smart control logic to completely overcome the identified problems by isolating metastability.

A voltage sensor was designed and implemented based on the proposed mechanism. Simulation results show that the design works as expected. For fair comparison, the solution in [7] is re-implemented using the same technology and design flow. Compared to [7], the new solution improves on measurement time (12.8% faster), and power (24.7% lower) and energy consumption (32.7% less). This is because the new solution removes the unnecessary extra corrections. It is also considerably smaller. In addition, our re-implementation of [7] does not include the sifting and additions it needs for equation (7). If those need to be considered our method is even better.

The proposed asynchronous solution makes the algorithm simple. Firstly a new detection mechanism is designed which directly detects \( V_{\text{sense}} \) discharging to \( V_{\text{low}} \) without using extra delays. This removes the need for any extra correction. As a result, in this solution, all possible metastability is restricted to one point. Secondly a MUTEX is used to filter out any propagation of possible metastability at this one point. Furthermore these allow a new mechanism to be designed to determine conversion completion rather than using extra delays, and extra result corrections.

In addition, although the algorithm is simplified, the resolution is not affected at all. And in the range from 0.69V to 0.83V, the proposed solution is better than the solution in [7] in terms of precision. In particular, the proposed solution produces monotonic code outputs, while [7] does not based on our experiments. When this happens, the resolution of our solution can be much higher than that in [7].

The preliminary layout has been done. In the future, the design will be taped out, tested, and the testing results will be compared to the obtained simulation results.

As the resolution is poorer in the higher voltage range, even though at least equal that achieved by [7], this is a region where we may achieve improvements through further investigations.

In addition, the proposed method is validated through studying a voltage sensing example. An obvious future step is to replicate this study through a capacitance sensing application.

**REFERENCES**


