Research Article

Concurrent Driving Method with Fast Scan Rate for Large Mutual Capacitance Touch Screens

Mohamed Gamal Ahmed Mohamed and HyungWon Kim

*Electronic Engineering Department, School of Electrical Engineering and Computer Science,
Chungbuk National University, 1, Chungdae-ro, Seowon-gu, Cheongju-si, Chungcheongbuk-do 362-763, Republic of Korea*

Correspondence should be addressed to HyungWon Kim; hwkim@cbnu.ac.kr

Received 10 April 2015; Accepted 21 June 2015

Academic Editor: Chi Chiu Chan

Copyright © 2015 M. G. A. Mohamed and H. Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel touch screen control technique is introduced, which scans each frame in two steps of concurrent multichannel driving and differential sensing. The proposed technique substantially increases the scan rate and reduces the ambient noise effectively. It is also extended to a multichip architecture to support excessively large touch screens with great scan rate improvement. The proposed method has been implemented using 0.18 μm CMOS TowerJazz process and tested with FPGA and AFE board connecting a 23-inch touch screen. Experimental results show a scan rate improvement of up to 23.8 times and an SNR improvement of 24.6 dB over the conventional method.

1. Introduction

Touch screen panels (TSPs) are very convenient computer human interface. They are used to detect human touch or a special pen by changing one of their physical properties. For medium- and small-size TSPs for mobile products, projected mutual capacitive-type TSPs have superior visibility and durability and also provide multitouch functions [1]. They are also increasingly adopted for large PC monitors and TV screens [2, 3]. In mutual capacitance TSPs, touch detection is determined by measuring the change in the mutual capacitance.

Figure 1 shows the conventional touch screen controller (TSC) system scanning a projected mutual capacitance TSP. It works to drive a sequence of pulses to the driving electrodes (TX) which results in a staircase signal after integration of the output signal coming out through the sensing electrodes (RX) [2, 4, 5]. The touch screen panel structure used in Figure 1 uses a crossline structure with large plates to magnify the value of the mutual capacitance to get better touch detection results. There are other different electrode patterns invented to control the values of mutual capacitances, self-capacitances, and electrode resistance and so control the touch detection sensitivity and accuracy [6, 7].

Single-line driving single-line sensing (SDSS) is considered the conventional scan method. It has a frame scan rate [3] given by the following formula:

$$\text{Scan rate}_{(\text{Conv.Sch.})} = \frac{1}{T_{\text{drive}} \times N_{\text{integration}} \times N_{\text{TX}} \times N_{\text{RX}}} \quad (1)$$

Here, N_{TX} is the number of TX lines, N_{RX} is the number of RX lines, and T_{drive} is a period of one TX pulse. $N_{\text{integration}}$ is the number of integration steps (or the number of TX pulses per sensing point). As indicated by (1), conventional TSC systems sequentially drive each TX line and sequentially sense each RX line, resulting in an unacceptable low frame scan rate for large touch screen panels.

Different research efforts have been done to improve the frame scan rate and signal to noise ratio as they are the most important issues in controller design [8–10]. Mutual capacitance touch screen panels are susceptible to noise, so much of research has been focused on how to reduce the noise and increase the touch signal strength [11, 12] by single-line driving differential sensing (SDDS). While there are software algorithm approaches [13–15], many papers proposed circuit-level approaches to address the noise reduction problem [16, 17]. Frame scan rate has been solved for small- and

FIGURE 1: Conventional touch screen controller system.

medium-size screens by different driving methods [12]. However it is still not efficient for large-size TSPs.

In this paper, we propose a new TSC architecture. This architecture can be used in multichip architecture to share scanning of a large TSP [18]. As the size of TSPs grows, in general, the number of driving (TX) lines and sensing (RX) lines tends to increase to keep their high resolution of touch detection. It becomes therefore difficult to control a large TSP because frame scan rate is much lower than required. Our touch screen controller used the proposed concurrent driving technique with differential sensing in order to improve frame scan rate and eliminate noise effects.

2. Concurrent Driving Method

Large touch screen panels usually require a large number of driving (TX) and sensing (RX) lines in order to satisfy their touch detection resolution requirements. This imposes high burden on TSC design. The time required to scan one frame grows along with the number of TX and RX lines in most touch screen control methods based on (1). However most TSC systems have a target frame scan rate that they must satisfy to ensure touch detection quality which is usually in the range of 100 Hz~200 Hz [8, 9].

Our TSC provides an effective solution with a very high scan rate for large touch screen panels. It consists of three components driving a specified number of TX lines, sensing a specified number of RX lines, and running detection algorithm software on the sensed data and then it sends touch position data to the next chip in multichip architecture via serial link. A novel driving and scanning technique is introduced, which reduces the frame scan time and ambient noise effects. The controller can be used for touch screen panels of different sizes, fabricated with various materials and various patterns.

Our touch screen scanning method consists of two steps to scan one frame of touch screen panel. In the first step (concurrent driving differential sensing (CDDS)), all TX lines (rows) are concurrently driven by opposite polarity pulses,

which is described by Figure 2(a). This step determines the touched RX lines by differentially sensing the output of each two adjacent RX lines. The idea behind this technique is to cancel the common mode signal while leaving only the differential signal by applying opposite excitation pulses to every adjacent cell in the same RX line. Since the major portion of ambient noise often appears as common mode signals in the adjacent RX lines, such noise can be canceled effectively by differential sensing leading to an increased signal to noise ratio (SNR). This step results in zero output in case of no touch while giving nonzero output if any cell on the two RX lines is touched. The second step (single-line driving differential sensing (SDDS)) is to sequentially scan each pair of two adjacent cells in only the touched RX lines as shown in Figure 2(b).

The differential integrator input integrates the difference between two adjacent lines. Through concurrent driving, each TX signal propagates through one TX line and two RX lines connected by two mutual capacitances as shown in Figure 3. Hence the differential RX output, in case of single-line driving, can be expressed by

$$V_{RX_m} - V_{RX_{m+1}} = (F_{TX}(S) F_{C_{M1}}(S) F_{RX}(S) - F_{TX}(S) F_{RC}(S) F_{C_{M2}}(S) F_{RX}(S)) V_{in} = F_{TX}(S) \cdot F_{RX}(S) (F_{C_{M1}}(S) - F_{RC}(S) F_{C_{M2}}(S)) V_{in}. \quad (2)$$

Here $F_{TX}(S)$ and $F_{RX}(S)$ are the transfer functions (TFs) for TX line and RX line, respectively. $F_{C_{M1}}(S)$ and $F_{C_{M2}}(S)$ are TFs for the mutual capacitance of the first and the second cells, respectively. $F_{RC}(S)$ is TF for the RC circuit connecting the first and the second cells. Considering that the resistance and self-capacitance connecting two adjacent cells are very small, we can ignore their transfer function. Therefore the differential output of two RX lines equals

$$V_{RX_m} - V_{RX_{m+1}} = (F_{C_{M1}}(S) - F_{C_{M2}}(S)) F_{TX}(S) F_{RX}(S) V_{in}. \quad (3)$$

FIGURE 2: Proposed concurrent driving method with differential sensing. (a) Step 1: concurrent driving with differential sensing. (b) Step 2: single-line driving with differential sensing. The output of integrator is positive if one of the even cells marked in purple color is touched, negative if one of the odd cells marked in green color is touched, and zero if there is no touch.

FIGURE 3: Schematic for one TX line connected with two RX lines in mutual capacitance touch screen showing the signal propagation path for differential sensing.

In case of no touch, the output is zero since the two mutual capacitances are equal and so are their transfer functions. On the contrary when there is a touch point affecting the two cells in different amount, the output is not zero. The output has positive or negative value based on which cell is touched and the polarity of the input signal as shown in Figure 2.

Applying the superposition theorem on the system of concurrent differential driving case, the system output is

the summation of the effect of all supplies which results in zero output in case of no touch, positive in case of touch affecting the even ordered cells (marked in purple in Figure 2), and negative in case of touch affecting the odd ordered cells (marked in green in Figure 2).

The timing diagram of Figure 4 illustrates the two steps. The first step (CDDS) drives all TX lines concurrently with opposite polarity pulses and sequentially reads out every pair

FIGURE 4: Timing flow diagram for concurrent driving method. $(m, m + 1)$ and $(n, n + 1)$ are indexes for touched RX pairs.

of two RX lines. During this step, each pair of RX lines is marked as touched or untouched. Once all the touched pairs are determined, the second step (SDDS) starts by selecting the first touched pair of RX lines and sequentially drives the TX lines starting with the first TX line. Each cell on the selected RX line is marked based on the differential integrator output. A positive integrator output indicates that the first cell of the RX pair is touched, while a negative integrator output indicates that the second cell is touched. The above process is repeated for all TX lines to scan all touched RX lines. Once all the touched RX lines are scanned, all touched cells (touch positions) can be determined.

Our method eliminates the needs for scanning all cells every frame which is the major drawback of conventional methods, and so it can improve the frame scan rate. For an example touch screen of $N_{TX} \times N_{RX} = 44 \times 78$ lines, a conventional sensing method requires $3432 \times \Delta t$ seconds where Δt is the time for scanning one cell. The proposed method, on the other hand, needs only $78 \times \Delta t + n \times 44 \times \Delta t$ seconds, where n is the number of RX pairs that are determined as touched pairs. Here, the first term ($78 \times \Delta t$) is the time consumed by the concurrent scan step which scans each of the RX lines and finds touched lines. The second term ($n \times 44 \times \Delta t$) is the time consumed by the selective scan step which scans only the touched RX lines.

Next, consider an example where five fingers are touching different RX lines. Assuming that one touch point affects three RX lines on average [5], the number of affected RX lines is 15 and the number of RX pairs is 10. Then the proposed method has a total scanning time of $78 \times \Delta t + 10 \times 44 \times \Delta t = 518 \times \Delta t$. This is an improvement of 6.63 times compared with the conventional method. However in case of no touch event for long time, our scanning method keeps repeating the first step until touch event occurs, so that our TSC keeps scanning the touch screen panel every $T_{drive} \times N_{integration} \times N_{RX}$ seconds which equals 0.5 milliseconds for 1.25 MHz excitation signal and eight integration steps. Therefore it increases the touch screen response because the first step takes very short time to finish.

For our proposed method, a generalized form of the frame scan rate is given by

$$\text{Scan rate}_{(\text{Prop.Sch.})} = \frac{1}{T_{drive} \times N_{integration} \times (N_{RX} + n \times N_{TX})}. \quad (4)$$

Our proposed method employs a differential sensing circuit to cancel ambient noise, which is the major cause of poor touch detection performance in most cases.

It can substantially speed up the overall sensing process allowing a very fast frame scan rate. It, therefore, is well suited for large touch screens.

3. Multichip Architecture

To support large touch screen panels, we may use one of the two conventional methods. The first is fabrication of another TSC that has a larger number of TX and RX lines. This, however, is time consuming and costly and cannot achieve a high frame scan rate. The second is to use multichip architecture with each chip employing a conventional scanning technique. For the first method, the scan time is proportional to the square of the screen size. The second method can allow sequential TX driving and simultaneous RX sensing. Thus its scan time linearly increases in proportion to the number of TX and RX lines.

In this section, we propose a novel scanning method which can efficiently reduce the overall scan time in multichip architecture. Figure 5 shows a TSC designed for this multichip architecture. This TSC conducts three main functions: drive a set of TX lines, sense a set of RX lines, and run detection algorithm. The chip consists of two main blocks: CPU and TX/RX controller. The CPU is connected with other functional blocks through an internal system bus called AHB bus. The CPU and TX/RX controller communicate through a set of control registers. While the TX/RX controller controls the TX and RX switches, the CPU reads sensing data from

FIGURE 5: Touch screen controller architecture.

FIGURE 6: The proposed touch screen controller (TSC) in multichip architecture scanning large touch screen panel.

its RX memories and runs an algorithm for touch position calculation. Table 1 summarizes the function of major blocks in the proposed TSC chip of Figure 5.

Even for extremely large touch screens, only a small number of TSCs are needed as shown in Figure 6. The number of TSCs (N_{TSC}) needed for a given touch screen size is given by

$$N_{TSC}^2 = \text{ceil} \left(\frac{\text{touch screen size}}{\text{supported size by one TSC}} \right), \quad (5)$$

where $\text{ceil}(x)$ is a function that rounds x up to the next integer.

Our chip supports a commercial 23-inch touch screen of projected mutual capacitance type, which has TX $\times$ RX electrode lines of 44×78 channels. The proposed multi-TSC scanning technique can support larger touch screens by using multiple chips. Let N_{TSC} be the number of TSCs. We partition the TX and RX lines of a touch screen into N_{TSC} segments of TX and N_{TSC} segments of RX lines, respectively. Then the N_{TSC} TSCs cooperate in 2 steps as given in Figure 6.

TABLE 1: Proposed touch screen controller block function.

Block	Functions
CPU	(i) Run algorithm which is installed on the external flash memory or data memory (ii) Feed control parameters to control registers block (iii) Start TX/RX controller (iv) Fill TX memory with TX signal sample values (v) Read RX signal sample values from RX memory (vi) Control UART and SPI blocks for data exchange with other chips or a host system
Host system interface	(i) Write the internal memories or control registers (ii) Test and debug other blocks (iii) Use the touch positions for host system applications
AHB bus	Connect CPU to other functional blocks
External memory controller	Connect external flash memory to work as program memory and SRAM memory to work as data memory
Data memory	Used as internal data memory. It can also be used as a program memory and data memory at the same time
RX memory (2 modules)	(i) Two-port memory operating as a FIFO with one port connected to the AHB bus and the other to an ADC (ii) Store TX output values sampled from the ADC, which is to be analyzed by the CPU (iii) If one of the two memory blocks is storing RX samples, CPU is reading the other one. Once the memory is filled up, the two memories swap their roles
TX memory	(i) Two-port memory operating as a FIFO with one port connected to the AHB bus and the other to signal generator to feed signal values to DAC (ii) CPU precalculates and stores signal samples into this memory block before initiating the TX/RX controller
Control registers	Register storing control parameters for TX/RX controller
ADC	Convert the analog RX signals sampled from sensing circuit
DAC	Convert digital TX values stored in DAC memory to analog values to be applied to TSP
TX/RX controller	Control all TX and RX switches, ADC, DAC, and TX driver. It sends synchronous control signals to all blocks based on the value of control registers
TX driver	Amplify the signal coming out from DAC to be applied to TSP. It also can set different gains using a control signal coming from TX/RX controller
Sensing circuit	Amplify the output signal coming from TSP and cancel the noise
Set of TX switches	Used to select active TX line/lines to apply TX signals
Set of RX switches	Used to select a RX pair to read out RX signal

Each TSC generates its TX signals in parallel and receives RX signals directly to/from TSP, and it also calculates the touch positions individually. For example, suppose we control a very large touch screen (132×234 lines), nine times larger in size than the touch screen size supported by one TSC introduced in Section 2. The proposed architecture uses three TSCs (TSC0~TSC2) as illustrated by Figure 7. Step 1 (CDDS): each of the N_{TSC} TSCs drives its segment of TX lines concurrently, while each of the N_{TSC} TSCs determines which RX lines are touched (stored in L_{RX}); Step 2 (SDDS): each of the N_{TSC} TSCs drives entire TX lines sequentially for only the RX lines that have been selected as touched lines in Step 1. The reading operation for RX lines is always conducted sequentially. In Step 1, the reading is done once per every RX pair. On the other hand, in Step 2, the reading is repeated as many times as the number of TX lines per each RX pair that was selected in Step 1.

Our proposed architecture, therefore, can further reduce the latency of detection. It also can reduce power consumption and SoC design costs. The following equation gives

the frame scan rate of the multichip architecture for large TSP supported:

$$\text{Scan rate}_{(\text{Prop.Sch.}+\text{MultiChip})} = \frac{1}{T_{\text{drive}} \times N_{\text{integration}} \times (N_{\text{RX(chip)}} + n_{\text{TSC}} \times N_{\text{TX}})} \quad (6)$$

Here n_{TSC} is the number of RX pairs determined as touched pairs by every TSC, $N_{\text{RX(chip)}}$ is the number of RX lines supported by one chip, and N_{TX} is the total number of TX lines.

4. Experimental Results

4.1. System Implementation. Our proposed method has been implemented with FPGA board connected with analog front end (AFE) board. The AFE board has our chip fabricated in $0.18 \mu\text{m}$ CMOS TowerJazz process. Figure 8 shows our system implementation and our chip layout. Fabricated chip has all analog switches, sensing circuits, and digital logic for

FIGURE 7: Operation of concurrent driving method in multichip architecture. (a) Pseudocode. (b) Timing flow diagram.

controlling switches. However FPGA board has ADC, DAC, and all control and processing logic. In this system, embedded CPU on FPGA is used to run touch detection algorithm and extract touch points in order to send them to PC through UART cable.

4.2. Scan Rate Improvement. As described in the previous section, our scanning technique (CDDS + SDDS) uses two steps to scan one frame. Conventional touch screen detection methods (SDSS or SDDS) need a total scan time of $N_{TX} \times N_{RX} \times \Delta t$ seconds to scan one frame. Here Δt represents the time required to scan one cell. Δt is determined by the frequency of driving signal and the number of charge integrations required to meet the target performance. On the other hand, for our proposed technique, it takes $N_{RX(chip)} \times \Delta t$ seconds to finish the first step and $n_{TSC} \times N_{TX} \times \Delta t$ seconds to finish the second step. Therefore, the scan complexity

of our proposed method is $O(2N)$ when a single touch is considered, while it slightly increases for multitouch. In contrast, the scan complexity of conventional methods is $O(N^2)$ regardless of the number of touches.

For the multichip TSC architecture, the required scan time does not increase with the number of TSCs employed, since all TSCs operate in parallel. In particular, the parallel processing keeps the time taken by the first step always the same, no matter how many TSCs are used. On the other hand, the required time to finish the second step of our multichip scheme is given by $n_{TSC} \times N_{TX} \times \Delta t$. For example, consider a multichip scheme with 3 TSCs. If the number of touched RX pairs for each TSC is (3, 5, 1), then n_{TSC} is 5, which is the largest number of RX pairs of all TSCs. Thus for the above example TSP of 132×234 lines, our method gives a scan time of $78 \times 1 \text{ us} + 5 \times 132 \times 1 \text{ us} = 738 \text{ us}$ (assuming $\Delta t = 1 \text{ us}$). In contrast, the conventional method requires

FIGURE 8: TSC system implementation using FPGA board and AFE board connected with a commercial 23-inch touch screen panel. Touch points are transferred to PC using UART connection in order to display them using graphical user interface software. The chip used in AFE board for testing is fabricated on $0.18\ \mu\text{m}$ CMOS TowerJazz process. It has all analog switches, sensing circuits, and digital logic for switches control. However FPGA board has ADC, DAC, and all control and processing logic.

$123 \times 234 \times 1\ \text{us} = 30888\ \text{us}$. This example thus shows that our multichip architecture can substantially improve the frame scan rate compared to conventional schemes.

Figure 9 compares the frame scan time for 3 different schemes simulated with touch screens of various sizes: (1) the proposed multichip architecture with concurrent scan, (2) multichip architecture with a conventional scan, and (3) conventional single TSC. In this comparison, the numbers of TX and RX lines of each touch screen size are scaled in a way that makes the pitch between the lines the same as the pitch of the 23-inch touch screen (a commercial touch screen used for experiment). This comparison is done with the following parameters: 1.25 MHz excitation TX pulses and 8 TX pulses for each charge integration. We also assume that every TSC has disjoint five touch points which leads to ten pairs of RX lines.

As shown in Figure 8, the frame scan rate for our technique is 22 times higher than the conventional scheme for the 70-inch touch screen and 14.3 times higher for the 46-inch touch screen.

4.3. Noise Cancellation. We have conducted experiments for a single TSC case with a 23-inch commercial touch screen with the system shown in Figure 8. We implemented the digital controller in an FPGA and the differential driving and sensing circuit in an AFE board. The differential integrator of the sensing circuit has been implemented in cascaded stages of subtraction and integration circuits.

Figure 10 compares experimental results of our TSC with differential sensing and the results of the single-line sensing method. For both cases, the output of the integrator is measured using concurrent TX driving with differential TX pulses for a 23-inch TSP. In Figures 10(a) and 10(b), the signal in blue color represents the difference between two RX lines in case of differential sensing. In Figures 10(c) and 10(d), the blue signal is the RX line signal for the single-line

FIGURE 9: Frame scan rate used with different scanning techniques for different touch screen sizes.

sensing method. The signal in green color represents one TX line input, and signals in sky blue and purple represent the integrator output.

The proposed CDDS scheme can cancel common mode signals, which can in effect remove most of ambient noise as shown in Figures 10(a) and 10(b). As described in Section 2, the integrator output of the proposed scheme becomes zero in case of no touch (Figure 10(a)) and nonzero in case of touch applied (Figure 10(b)). On the other hand, concurrent driving with single-line sensing (CDSS) scheme results in nonzero output in both untouched (Figure 10(c)) and touched case (Figure 10(d)). In this scheme, the integrator output gives larger values for touched case. Nonzero output of the untouched case is due to the accumulation of the signal difference between every two adjacent cells in the same RX line.

Therefore, the proposed CDDS method enables efficient noise reduction by cancelling ambient noise. This effect can

FIGURE 10: Measurement results of a 23-inch touch screen panel with concurrent differential driving. (a) With differential sensing in case of no touch. (b) With differential sensing in case of touch applied. (c) With single-line sensing in case of no touch. (d) With single-line sensing in case of touch applied.

FIGURE 11: Experimental results for a 23-inch touch screen panel. (a) SNR of four different driving and sensing schemes. (b) Scan rate of the proposed method CDDs compared with conventional method SDSS. This experiment was conducted with 1.25 MHz TX pulse signal with 10 V swing. Note: SNR is calculated using the equations in [19].

be observed by inspecting that the peak-to-peak noise values in Figure 10(a) are much lower than that of Figure 10(c). Such lower noise values allow smaller deviation of the signal samples at the ADC from their average values and so lead to more accurate decision of touch detection. Therefore, the CDDs scheme gives a higher SNR, and then it can eliminate

the need for additional noise filters. Figure 11(a) compares the SNR gain of four different driving and sensing schemes. First, we compare the two schemes of concurrent driving. While the SDDS scheme gives 69.13 dB, the proposed scheme CDDs provides an SNR of 67.82 dB showing negligible SNR degradation. Then we compare the 2 schemes of single-line driving.

While the SDSS scheme gives 43.2 dB, the CDSS scheme gives 40.11 dB. These results show that the concurrent driving has little degradation in SNR compared to the single-line driving scheme. The differential sensing, however, provides a large SNR improvement, 27 dB, over the single-line sensing. Figure 11(b) shows that the proposed scheme CDDS provides 6.63 times higher scan rate than the conventional scheme SDSS for 5-finger touch case and 23.83 times higher for one-finger touch case. These experimental results, therefore, prove that the concurrent driving with differential sensing scheme provides substantial improvement in frame scan rate at little sacrifice in SNR.

5. Conclusions

We have presented a new architecture for touch screen controller which can be extended to a multichip structure for parallel processing. We have also developed a concurrent driving and high SNR sensing method for projected mutual capacitance touch screens. A two-step algorithm is proposed to reduce the time required for scanning large touch screens. With the concurrent driving and multichip architecture, our method can provide substantially faster frame scan rate than conventional methods, a crucial requirement for large touch screen panels (TSPs). Experimental results show that our proposed method can improve the frame scan rate by 6.6 times for a 5-finger touch and 23.8 times for a single-finger touch for a 23-inch TSP compared to a conventional method. They also show that its touch detection SNR gain is 67.82 dB, a 24.6 dB improvement compared with the conventional scheme.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the Center for Integrated Smart Sensors funded by the Ministry of Science, ICT & Future Planning as Global Frontier Project (CISS-2014011066).

References

- [1] G. Barrett and R. Omote, "Projected-capacitive touch technology," *Information Display*, vol. 26, no. 3, pp. 16–21, 2010.
- [2] M. G. Mohamed, T.-W. Cho, and H. Kim, "Efficient multi-touch detection algorithm for large touch screen panels," *IEEE Transactions on Smart Processing and Computing*, vol. 3, no. 4, pp. 246–250, 2014.
- [3] I. Seo, T.-W. Cho, H. Kim, H. G. Jang, and S. Lee, "Frequency domain concurrent sensing technique for large touch screen panels," in *Proceedings of the IEEE Fall Conference*, Seoul, Republic of Korea, November 2013.
- [4] J.-S. Lee, D.-H. Yeo, J.-Y. Um et al., "37.3: a 10-touch capacitive-touch sensor circuit with the time-domain input-node isolation," *SID Symposium Digest of Technical Papers*, vol. 43, no. 1, pp. 493–496, 2012.
- [5] I. Seo, U. Jang, M. Mohamed et al., "Voltage shifting double integration circuit for high sensing resolution of large capacitive touch screen panels," in *Proceedings of the 18th IEEE International Symposium on Consumer Electronics (ISCE '14)*, Jeju, Republic of Korea, 2014.
- [6] J. Lee, M. T. Cole, J. C. S. Lai, and A. Nathan, "An analysis of electrode patterns in capacitive touch screen panels," *Journal of Display Technology*, vol. 10, no. 5, pp. 362–366, 2014.
- [7] H. Akhtar and R. Kakarala, "A methodology for evaluating accuracy of capacitive touch sensing grid patterns," *Journal of Display Technology*, vol. 10, no. 8, pp. 672–682, 2014.
- [8] A. Ng, J. Lepinski, D. Wigdor, S. Sanders, and P. Dietz, "Designing for low-latency direct-touch input," in *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*, pp. 453–464, Cambridge, Mass, USA, October 2012.
- [9] A. Ng and P. H. Dietz, "The need for speed in touch systems," *SID Symposium Digest of Technical Papers*, vol. 44, no. 1, pp. 547–550, 2013.
- [10] J. H. Yang, S. H. Park, J. Y. Jeon et al., "A high-SNR area-efficient readout circuit using a delta-integration method for capacitive touch screen panels," *SID Symposium Digest of Technical Papers*, vol. 43, no. 1, pp. 1570–1573, 2012.
- [11] I.-S. Yang and O.-K. Kwon, "A touch controller using differential sensing method for on-cell capacitive touch screen panel systems," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1027–1032, 2011.
- [12] H. Ma, S. Heo, J. J. Kim, and F. Bien, "Algorithm for improving snr using high voltage and differential manchester code for capacitive touch screen panel," *Electronics Letters*, vol. 50, no. 24, pp. 1813–1815, 2014.
- [13] M. G. A. Mohamed, U. Jang, I. Seo et al., "Efficient algorithm for accurate touch detection of large touch screen panels," in *Proceedings of the 18th IEEE International Symposium on Consumer Electronics (ISCE '14)*, Jeju, Republic of Korea, June 2014.
- [14] K. Lim, K.-S. Jung, C.-S. Jang, J.-S. Baek, and I.-B. Kang, "A fast and energy efficient single-chip touch controller for tablet touch applications," *IEEE/OSA Journal of Display Technology*, vol. 9, no. 7, pp. 520–526, 2013.
- [15] Y. Kim and A. H. Tewfik, "Low power detection on capacitive touch screens," in *Proceedings of the 1st IEEE Global Conference on Signal and Information Processing (GlobalSIP '13)*, pp. 638–641, December 2013.
- [16] H. Shin, S. Ko, H. Jang, I. Yun, and K. Lee, "A 55 dB SNR with 240 Hz frame scan rate mutual capacitor 30×24 touch-screen panel read-out IC using code-division multiple sensing technique," in *Proceedings of the 60th IEEE International Solid-State Circuits Conference (ISSCC '13)*, pp. 388–389, February 2013.
- [17] I. Seo and H. Kim, "Dual sensing with voltage shifting scheme for high sensitivity touch screen detection," *Journal of the Institute of Electronics and Information Engineers*, vol. 52, no. 4, pp. 71–79, 2015.
- [18] G. Choi, M. G. A. Mohamed, and H. Kim, "Distributed architecture of touch screen controller SoC for large touch screen panels," in *Proceedings of the International SoC Design Conference (ISOCC '14)*, pp. 22–23, IEEE, Jeju, Republic of Korea, November 2014.
- [19] ATMEL, *Buttons, Sliders and Wheels: Sensor Design Guide*, ATMEL, 2011.