# Efficient Motion Capturing and Idling Stop Display system utilizing CAAC – IGZO semiconductor FETs

†‡Ioannis Papakostas, †‡Georgios Dikaros, †Shuhei Maeda, †Takuro Ohmaru,
†Takayuki Ikeda, ‡Atsuto Maki and †Shunpei Yamazaki

Abstract

This paper focuses on the integration of two applications of crystalline oxide semiconductor FETs, developed by SEL, the Motion capturing system and the Idling Stop (IDS) Display, in order to create a new efficient motion capturing and display system. For the idling stop feature, refreshing is performed only if there are differences between the current displayed frame and the one that is input to the device. The new system ensures efficiency by eliminating the need for an image processor and extra memory on the display side, since comparison has already been performed by an analog processor on the Motion capturing system and images are fed to the display through an interface which supports SVGA transmission.

## 1. Introduction

In today's world, efficiency and sustainability are notions which constitute the gist of engineering, thus, it is of utter importance to create technology which combines high quality of services and low power consumption. Towards this goal, various types of applications have been successfully developed by Semiconductor Energy Laboratory, Co., Ltd. (SEL), utilizing the crystalline oxide semiconductor (typically, c-axis-aligned a-b-plane-anchored crystal In-Ga-Zn-O, CAAC-IGZO), which realizes FETs showing extremely low off-state current [1] and easily constructing non-volatile memory with a gain cell type structure [2]. Among the most notable applications, there were two systems of particular interest for their low power consumption characteristics, the techniques used in their design and the possibility of their integration in order to create a new system. The Motion Capturing System and the Idling Stop (IDS) Display showed particular promise independently, however, it was clear that the connection of the two devices into one should be the next step. In this paper, we will describe the two pre-existing applications, analyze the process of creating an interface for data transferring, using DVI wired communication and finally propose future work which can potentially achieve even better results.

† Semiconductor Energy Laboratory Co., Ltd.
‡ KTH Royal Institute of Technology

## 2. Systems

The two pre-existing systems were created independently and each one of them was designed to utilize to its best the advantages offered by the oxide semiconductor FETs. In this section both devices are described for a better understanding of the reader.

### 2.1. Idling Stop Display

Low off-state current contributes to decrease in power consumption, a fact that makes the IGZO transistor, an excellent component for displays of personal computers and personal digital assistants and also devices that are specialized to store data, such as memory. Among the applications created, is a liquid crystal display (LCD) [3] and an organic light emitting display [4] utilizing a CAAC-IGZO FET as a pixel transistor, to achieve ultra-low refresh rate of image data because of their improved data retention. These displays, which are called "Idling Stop (IDS) Displays", can be used like e-paper requiring no data refresh while image data are not updated. However, in order to apply the IDS display approach to the images that are used as input, it is needed to detect differences between images in each frame with an image processor. Power consumption of this image processing and power reduction of the idling stop driving, are in the relationship of a trade-off.

Taking all the aforementioned information, regarding the IDS Display, into account, we can conclude that the materials, components and techniques used for this display had a high potential of reducing the overall power consumption if we could lower the power consumption of the image processing.

### 2.2. Motion Capturing System

The heart of our motion capturing system is a vision sensor which features in-pixel non-volatile analog memory utilizing a c-axis-aligned a-b-plane-anchored crystal In-Ga-Zn-O (CAAC-IGZO) [5]. This crystal constitutes the base for a FET with extremely low off-state current, making it ideal for applications where storing data is essential. Our sensor consists of four basic components: the analog processor, the pixel array, the row/column drivers and the Analog-to-Digital Converter (ADC). The major requirement regarding the efficiency of this motion capturing system has always been low power consumption, thus there has to be a component that is responsible for activating/deactivating only the components needed on every stage of the process, from capturing an image to digitizing it. This task is assigned to the controller, which provides the activation/deactivation signals and guides the whole process through three operating modes, namely the motion capturing, the standby and the imaging mode. In ***figure 1***, all the components constituting the motion capturing system as well as the three operating modes are shown, while the procedures taking place in each mode are described below.

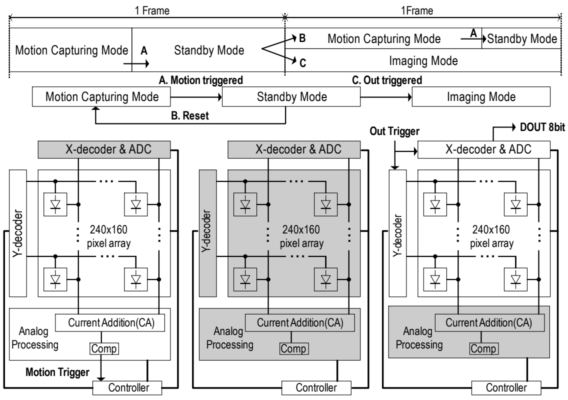


**Fig. 1** *Operating modes of motion capturing system*

Our system is configured to work with a frequency of 60 frames per second which means that it is allocated a time period of 16.67ms in order to capture and process a new frame. For Motion Capturing mode this process can take less time (depending on how soon motion detection will happen) while for the imaging mode it takes exactly 16.67ms. The initial mode, where motion is detected, is the motion capturing mode. All components are active except for the Column driver and the ADC. Motion detection is performed by the analog processor, which is responsible for evaluating differences between the captured and a reference frame. Once motion is detected, the system immediately switches to the standby mode, where all components are deactivated, dropping in this way power needs and remains in this state until the next frame. In the next frame we have a transition to the imaging mode, where all the components but the analog processor, are reactivated, including the Column Driver and the ADC. In this mode our sensor captures a new frame and digitizes it after it has first stored it as the new reference frame. Image Sensor part of ***figure 2*** depicts what is described in this section.

## 3. Connection of the two applications - Interface

Given the logical correlation between the applications - one captures images that have to be displayed, the other displays images that it receives as input – and the fact that there are many advantages in creating a connection between them, we proceeded with creating an interface for the transmission of image data from the Motion capturing system to the IDS Display. This way we would combine two devices with very low power consumption to create a complete solution with potential impact in fields like surveillance and at the same time we would solve the biggest issue of the IDS Display which was the comparison of images. Data comparison is already carried out at the motion-capturing side, where it is both essential and more efficient power-wise, thus, there is no need for an image processing system on the display side.

### 3.1. Requirements

For the implementation of the interface, various factors had to be considered. Compatibility, flexibility, image resolution, data rate and of course power consumption were among the most important parameters that influenced the design decisions.

In order to connect systems for the transmission of data, system architects often have to deal with data conversion for compatibility. Starting from the Motion capturing system side, data is captured in 240x160 resolution, while the IDS, displays images in Full HD quality (1920x1080). The interface we would create should be able to convert an image to the desired resolution and also transmit data at a frequency high enough for the transmission to be completed before the time when the next image should be transmitted. This way we would ensure the smooth functionality of the system and the absence of problems related to inconsistent data.

Moreover, we would like the frame/refresh rate to be modified in a way to reflect motion detection. Thus, the rate at which images are displayed, should be the rate at which new images are input from the Motion capturing system.

Last but not least, the desired behavior of our system would be to display an image at all times. Practically this means that we would have to account for the data retention time of the display. Even if no motion is detected, we have to refresh the image displayed before data is lost.

### 3.2. Design Decisions

Taking all the aforementioned problems into consideration, the process of designing the interface became a challenging and demanding task. The system we propose is structured in a way that it performs as desired and allows for careful monitoring of the different stages of the process. Therefore, the role of bridging the gap between the Motion capturing system and the IDS Display, was decided to be performed by software. The software used acts like the intermediate between the two devices, allowing for images to be transmitted, after being processed and also stored in memory. For this purpose, software, which served as complement to the two devices, was modified to meet the new needs. The new versions of this software allow for an image to be received, stored, processed and displayed in the monitor. Moreover, it controls the refresh rate and other details, such as the height and width of the image to be displayed.

#### *3.2.1. Image resolution*

Trying to tackle the problems analyzed in the previous section, we started from the issue of different image resolution. We decided to deal with this issue in software where we can have better flexibility on the conversion of resolution and adapt to different resolution needs.

#### *3.2.2. Refresh rate dependent on motion capturing*

The second issue we resolved had to do with the rate of image refreshing on the IDS side. The solution we chose for this problem was based on making the refresh dependent on the rate at which different images are received. This way, it is ensured that the display would work in an idling stop manner as long as it would not receive a different image. After a new image is captured, it is output to the FPGA responsible for converting the raw data, to data that can be transmitted to the IDS display. The output of the FPGA is then provided as input to a DVI transmitter after the image resolution has been converted, from 240x160 to 800x600. However, this increase in the number of pixels does not represent increase in quality or size of the image, as the real data remain at 240x160. The major benefit of using the existing DVI format is that we could use the excess of pixels to control the refresh rate of the IDS display by providing information acquired at the motion capturing side. To achieve our purpose we set one of the remaining pixels to carry the information provided by the motion trigger signal. If the image was different than the one captured, the value of the pixel would be set to *250*. If it was the same, then its value would be equal to *0*. Software decides on refreshing the image or not by checking the value of the pixel against the previously mentioned values. See ***figure 3*** – Interface Part).

#### *3.2.3. Data retention time*

Next issue that had to be taken care of was data retention time. We can control the maximum time without refresh of the display by setting an initial value to the variable ***Reg_Frame***. The internal counter is refreshed every time a different image is received. (See ***figure 3*** – Software part). Finally, ***figure 2*** depicts the complete functionality of the application through a block diagram.

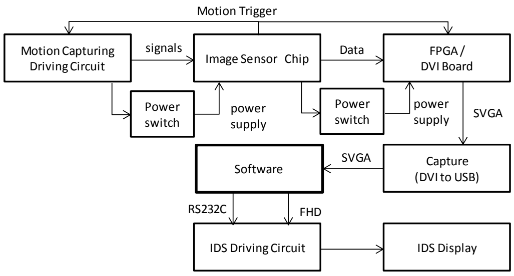


**Fig. 2** *Block Diagram of the connection between the devices.*

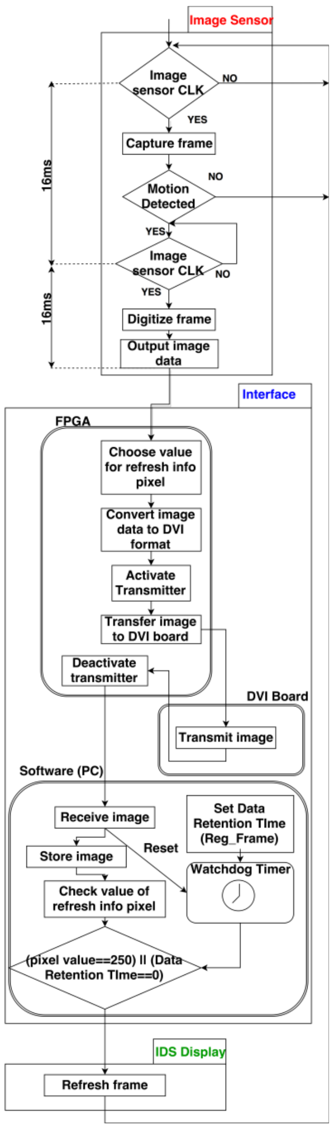


**Fig. 3** *Flow chart of the new system.*

### *3.2.4. Operation Algorithm*

The final design decision that was taken was related to the operation of the interface. One of the major requirements that have to be met, is not to lose any frames. This means that all images produced by the motion capturing system have to be displayed on the IDS. The most demanding scenario is that of continuous motion, which means that an image is produced every two frames–approximately **32ms**. If we can ensure proper functionality of the system for this scenario, we can guarantee that no frames will be lost for any other scenario as well. The key to such an achievement is selecting an operating frequency for the different parts of the interface that adds no overhead. Motion can be detected once every **32ms**, therefore, this time period of **32ms** determines how often transmission of data to the IDS can happen. Our circuit has to be configured in such a way that it finishes its task and consumes the least energy during this time window. This leaves much space for experimentation and investigation in order to find the best configuration for efficiency. Frequency of operation, for image transmission to the IDS, does not change the total energy consumption for the duration of active operation but determines how much time of the **32ms** will be left after that. For the time that is left, a decision has to be made to either shut the component down or keep it powered on (in standby mode). The following investigation concerns only the cases for which motion is detected in consecutive turns. It is obvious that if no motion is detected, the best configuration is to keep all the components powered off. As we can see in ***figure 4***, we can identify three different values of frequency which affect this decision.

Frequency $f_1$ is the highest frequency at which the operation can be executed. This depends on the maximum supported frequency. Frequency $f_2 < f_1$ is a frequency for which no matter which value we choose in the **[**$f_2$, $f_1$**]**, it is always better to power off and power on components. This holds because we can take advantage of the time we keep the components powered off. As we lower the frequency, the time required for the task to be executed is longer. This means that the total energy consumption for the components in standby is getting lower and lower as it is proportional to time. On the other hand, energy consumption of turning the components off and then on again remains the same. This continues until we reach the frequency $f_2$. Specifically for this frequency it makes no difference if we choose to shut down components or not because the same amount of energy is consumed. We say that $f_2$ is the **break-even point**. Frequency $f_3 < f_2$ is the lowest frequency at which the task can be performed in order to stay within the **32ms** limit. No matter which value we choose in the **[$f_3$, $f_2$]** it is always better to keep the components powered on (***figure 4***). This holds because for any frequency in this range the energy consumed in standby mode is less than the aggregated energy consumed by powering off and powering on the transmission circuit. Frequency $f_3$ leaves no time for powering off and then powering on again or being in a standby mode because the circuit is supposed to be constantly operating.

In the case of keeping the components powered on (Standby mode) some further investigation has to be conducted. For how much time should the components remain powered on? For this paper we assume that the system will be used in an environment of human movement or similar, relatively slow motions. According to figure 6 in reference [6], even when conditions impose movements of small duration, those movements are above a threshold of **225ms**. Given that this is the least time period in which motion can be detected we can easily derive that if we detect motion once, it is very likely that we will detect motion again several more times. In this case, we would prefer the components to stay active for a time as close as it gets to **225ms.** Since operation is divided in cycles of **32ms** as explained before, we have to find a number of **M** cycles during which the components must stay powered on. Then the total time for which we keep the components active before powering them off again will be:

$$Tac = M * 32ms \quad \textbf{(2)}$$

When a new image is received from the motion capturing system, the FPGA checks the status of the transmission components. If they are powered off, they are activated and their activation timer is set to the desired value ($Tac$). Converted data are then transferred to the PC and finally the image is transmitted to the IDS display. Transmission components remain powered on until the transmission activation timer reaches 0.

To further elaborate on this basis, we can consider the case for which a motion which lasts **225ms** starts right after the motion capturing system has captured a frame to check for motion. The next time the system will check for motion again is after approximately **16ms –** the time required to be spent in the motion capturing mode without detecting motion. This means that motion can be detected **16ms** after it has started which leaves us with **225ms – 16ms = 209ms** of motion. Then, equation **(2)** can become:

$$209 = M * 32$$

But the number **M** represents the number of operating cycles and can only be an integer value, so we choose **M = 6** full cycles or in other words:

$$Tac = 6 * 32 = 192ms.$$

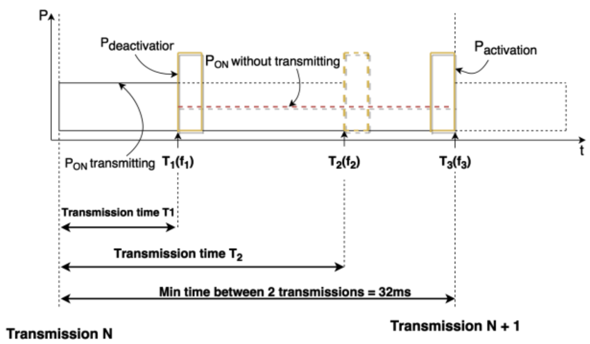


**Fig. 4** *Power consumption related to the different frequencies. The integrals for each figure with respect to time give the Energy consumption for that amount of time.*

## 4. Discussion - Future Work

In this paper, we made an investigation for finding different solutions for the connection of two very energy efficient devices in order to create an integrated system for motion capturing and image display.

There are many limitations to the problem we tried to address, due to the nature of such system and also the versatility in its implementation. This unified product can be customized, based on the restrictions and constraints of its application in order to become much more efficient. In our model, we make some assumptions to assist our investigation and we report potential solutions for specific conditions and needs.

As future work, there is much interest in conducting experiments for measuring the different power consumptions and make the required comparisons in order to draw conclusions regarding the optimum solution. If keeping the components powered on proves more efficient than powering them off and on again, research can be made regarding the optimum number of cycles, **M**, for which the components should remain powered on. For this paper we limited our work to the least duration of movement according to [6], but further optimization could be made with different data, like the average duration of a movement. After that, the system can be adjusted and implemented depending on the specific application in which it is going to be used.

Moreover, our system should become much more independent. The existence of software facilitates our needs for the moment, because it offers the option of constant monitoring, but makes our system so dependent on PC software in order to function properly. An automatic process with a different interface can be implemented in the future, which can feature all functionality offered by the PC. In surveillance environments, keeping records of motion detection could be as important as detecting it, thus the new interface could be enhanced with an extra circuit for encoding data and a device for storing data which can be either integrated or external.

Another approach we have to consider would be to enhance our system with more flexibility and practicality. Towards this direction we can use wireless communication for the connection of the two devices. Even though wireless protocols produce very much consumption overhead [7], this can be mitigated to a good extent by the high efficiency of the rest of the components of the system. Such an implementation, although significantly less efficient than a wired one, could boast notably lower power consumption than similar implementations.

## 5. Conclusion

In conclusion, our main proposal is very promising in creating an interface for the unification and integration, of an efficient motion capturing system along with an efficient display. The extra consumption overhead of the interface, is limited due to the proposed algorithms for its operation and the whole system becomes more efficient, by removing the need for an image processor on the IDS display side while preserving its idling stop feature. The evaluation of the proposed solution will be dependent first on its functionality, as it should perform according to the requirements, and also on the total power consumption compared to the two independent systems. Meeting the requirements and achieving lower or equal power consumption will constitute a success for our project.

## References

[1] Y. Sekine et al., “Success in Measurement the Lowest Off-state Current of Transistor in the World”, Electrochemical Society Transactions, 37(1) pp. 77-88 (2011).

[2] H. Inoue et al., “Nonvolatile Memory With Extremely Low-Leakage Indium-Gallium-Zinc-Oxide Thin-Film Transistor”, Journal of Solid-State Circuits, 47(9) pp. 2258-2265 (2012).

[3] S. Amano et al., “Low Power LC Display Using In – Ga –Zn – Oxide TFTs Based On Variable Frame Frequency” , Society of Information Display, 41 pp. 626-629 (2010).

[4] T. Tanabe et al., “New Threshold Voltage Compensation Pixel Circuits in 13.5-inch Quad Full High Definition OLED Display of Crystalline In-Ga-Zn-Oxide FETs”, Society of Information Display, 43 pp. 88-91 (2012).

[5] T. Ohmaru et. al, “25.3μW at 60fps 240x160 Pixel Vision Sensor for Motion Capturing With In-Pixel Non-Volatile Analog Memory Using Crystalline Oxide Semiconductor FET”, IEEE International Solid State Circuits Conference, Session 6, Image Sensors and Displays, 6.5 (2015)

[6] H. Tanaka et.al, “An Optimization Principle For Determining Movement Duration”, Journal of Neurophysiology, vol. 95, no. 6 (1 June 2006)

[7] M. Siekkinen et al., “How low energy is Bluetooth low energy? Comparative Measurements with ZigBee/802.15.4”, WCNC 2012 Workshop on Internet of Things Enabling Technologies, Embracing Machine-to-Machine Communications and Beyond (2012).