WHITE PAPER

Smart Sensors

Why sensors need intelligence?

Joseph Yiu May 2014

Introduction

Smart sensors have become a popular topic in the last couple of years. It is often related as a part of the “Internet of Things (IoT)” evolution. However, even without the IoT aspect, the growth of smart sensor market is still happening.

In general, a sensor is considered to be smart, when it has built-in data processing functionalities, communication interface. In many cases a smart sensor also includes a processor system, making it almost like a microcontroller. For example, as shown in figure 1:

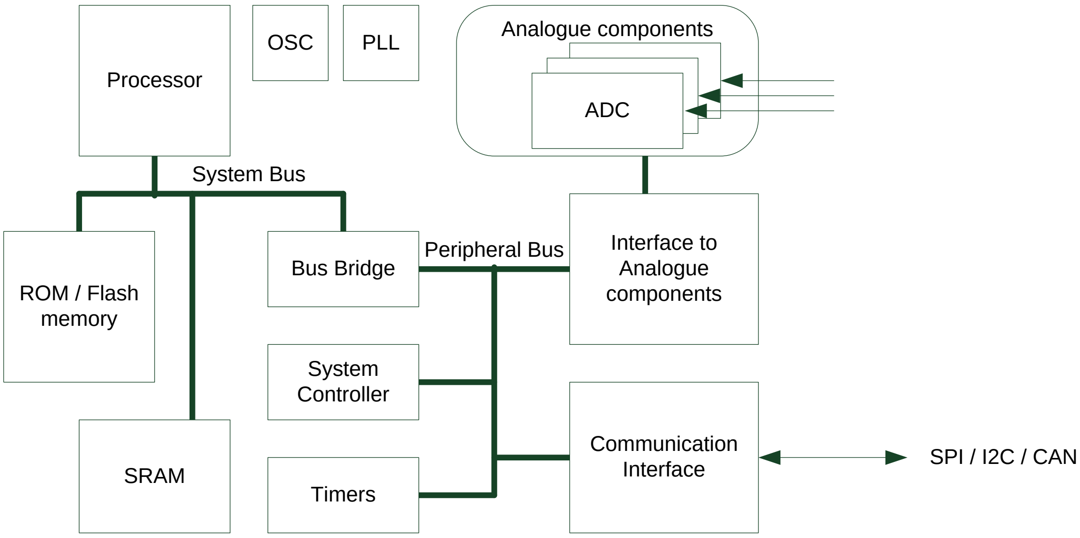

Figure 1 - A simple processor based system for a smart sensor

In fact, with the right peripheral interface (e.g. ADC) many microcontrollers can be used as intelligent sensors. However, there are also many custom designed smart sensor chips which provide special interfaces, such as touch screen controllers.

Reasons for smart sensors

Traditionally, sensors might only contain some analogue circuits, and sometimes additional digital interfaces. The digital interface enables the sensors to be connected to other components (e.g. microcontrollers) easily.

For some of the more advanced sensors, additional digital hardware are added to carry out various forms of digital signal processing such as filtering and power management.

In recent years, instead of adding more and more digital circuitry into the sensors to provide additional functions, it is getting more common and efficient to put a processor system into the design. There are a number of reasons for this:

Better accuracy and reliability

Adding a processor in a sensor design makes it easy to add additional features to improve accuracy and reliability. For example, self calibration and built-in self test (BIST) are common requirements for many smart sensors. Also, run time self checking functions can also be done. For example, in many safety critical applications such as automotive, it is crucial for sensors to report if there is a potential fault in the sensors, rather than reporting information that could be incorrect.

A processor system with programmable non-volatile memory (NVM) can also take advantage of the NVM to store and update calibration data, and use software to handle additional compensation or fine tuning of sensor data, and report issues such as signal out-of-range or supply voltage issues. While these can be handled with custom digital design, it can be complex to handle and update, should one want to add new functionalities.

Better energy efficiency

In most cases sensors need to connect to other processor systems via a communication interface. Typically the main processor (e.g. an application processor in a mobile computing device) can consume much more energy than the sensors, so it is best to allow the main processor to stay in sleep mode as much as possible.

The communication interface between the application processor and the sensors can also consume quite a lot of power (e.g. due to I/O pad driving current and the associated digital logic).

For better energy efficiency, a smart sensor can help by:

- Automatically sample data, and only wake up the main processor if useful data is captured.

- Apply Intelligence data sampling management to reduce data sampling activity when not required.
- Buffer up data in SRAM and only wake up the main processor if a certain number of data samples are available.

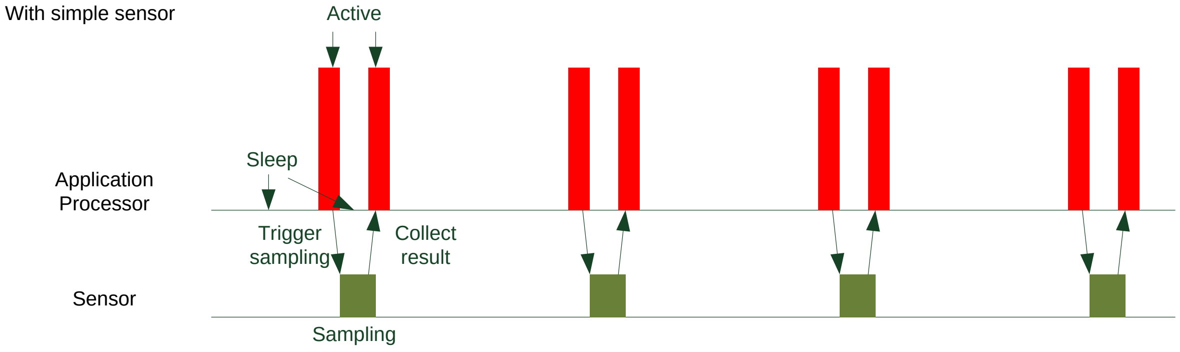

Figure 2 - Simple sensors require more involvement from the application processor and hence consume more power.

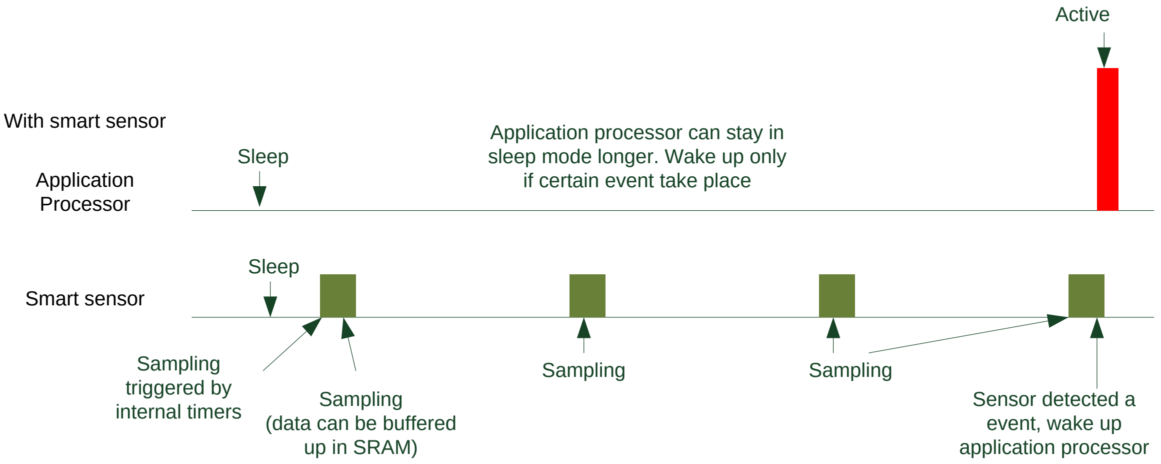

Figure 3 - Smart sensors can manage data sampling schedule and wake up application processor only when necessary.

It can also reduce the amount of data passing on the communication interface by:

- Filtering or pre-processing data, or consolidating information from multiple sensing elements before reporting results.
- Apply data compression to reduce data size if possible.

Furthermore, a smart sensor can also reduce power consumption by:

- Applying sleep mode features on the processor and utilize low power design techniques in microcontrollers to get lower power consumption.

All these techniques enable a system-designer to reduce the processing needed on the main application processor, as well as maintaining a low power profile on the sensors.

Additional features

Having a processor inside the sensor opens up new opportunities for additional features. For example, a smart sensor can have built-in diagnosis mode, or even test mode for corner case test stimulus generation (you might not want to really apply high temperature to your smart phone prototype when developing the software for the temperature sensors).

You can also consider representing the captured data in other ways to reduce the data processing needed on the main processor.

Flexibility

A programmable sensor allows designers to customize the data sampling sequence, data processing and other behaviours easily. If the whole system is designed with hardware logic, the design can be very complex to handle, and can be difficult to verify. In the worst case, if a bug is found in the hardware design, the chip might become useless, whereas software based systems can be patched by reprogramming the NVM (e.g. flash memory) on the devices, for bug fixing and features updates.

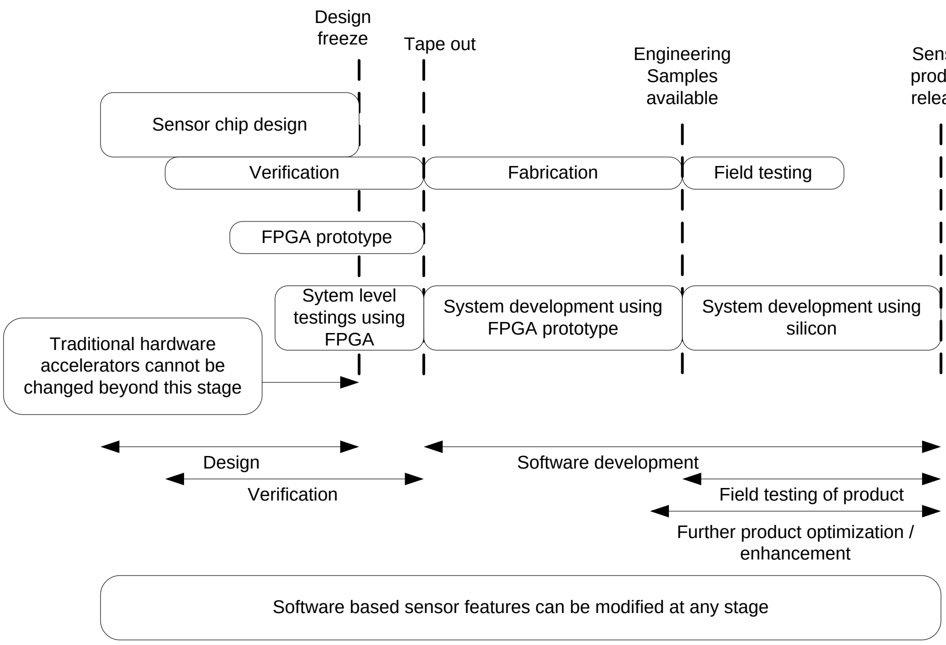

Figure 4 - Smart Sensors with processor system have a much more flexible design cycle.

Another potential advantage of using a software based approach is that you can create multiple profiles of the sensor operations for different operation conditions.

Ok, but what's the catch?

Many readers might wonder, is there any drawback of using a processor system? Potentially yes. For example, the data processing speed of a processor system is not going to be as good as dedicated hardware processing units. However, given that most data captured by sensors are not timing critical, a small delay of the processing results usually doesn't matter.

Another area that may worry some designers is that the silicon area may increase. In reality, often the main portion of the silicon area on a sensor is occupied by the large analogue circuits, so adding a microprocessor will only be, if at all, a small increase in the silicon area. In fact, if you want to handle lots of data processing only using hardware accelerator logic on the sensors, you might actually end up with a large digital design which is much larger than an ARM® Cortex®-M0 processor.

In some cases, if you need to store calibration data and buffer up sampled data on the sensor, you will possibly need an on chip flash and an on chip SRAM macro anyway, which are significant in size

Typically in sensors applications, low power, accuracy and reliability are critical factors. The total silicon area could be slightly larger with a processor system on chip, but you can still get lower power, better accuracy and reliability using a processor.

Design Considerations

For designers of smart sensors, the correct choice of the processor is a key part for the success of their sensor product. Since most sensors are built on mixed-signal semiconductor processes (e.g. 180 Ultra Low Leakage standard cell library), the transistor geometry can be relative large and therefore small gate count is often a critical requirement, as it translates in smaller silicon area. In addition, the energy efficiency of the processor, low power features, configurability can all be very important.

Furthermore, let's not forget to consider the code density aspect. Indeed, since a fair amount of the silicon area will be picked up by the NVM (e.g. on chip flash memory), a processor with high code density will allow a smaller NVM size and hence can contribute to reduce the silicon cost.

Why use ARM Cortex-M processors in smart sensors?

Today, many companies are already using ARM® Cortex®-M processors to power their smart sensor products. For example, ST's LIS331EB smart motion sensor is a 3-axis sensor-fusion accelerometer based on the ARM Cortex-M0 Processor. The Cypress TrueTouch Noise immune multi-touch sensor is also based on ARM Cortex-M processor.

Figure 5 - Examples of sensors based on ARM Cortex-M processors – ST LIS331EB smart motion sensor, Cypress TrueTouch sensors.

There are many technical advantages of using the ARM Cortex-M processors. First of all, they are very small (minimum gate count is only about 12K gates) and are very energy efficient. They also provide excellent code density, allowing the sensors to be implemented with a small amount of NVM, thus reducing silicon size requirements.

Although the overall silicon size might increase by adding a processor system (mostly due to memory size), a chip designer can utilize various low power design techniques for microcontrollers and sleep mode features on the processor to reduce power consumption, whereas implementing similar low power optimization in hardware processing blocks can be much more difficult and time consuming.

The Cortex-M processors also have very flexible interrupt management by having a built-in interrupt controller called NVIC (Nested Vectored Interrupt Controller), which allows for automatic handling of interrupt prioritization, nesting of interrupting, exception vector lookup, etc.

High configurability of the processor design is very useful in sensor applications. Unlike off-the-shelf microcontroller products, a sensor device might only require a small number of IRQ inputs and does not need some of the advanced debug features. The Cortex-M processors have a number of configuration parameters to allow some of the unused features to be removed to reduce gate count and power.

A wide range of processor choices and scalability of the architecture is another important advantage. For example, if a designer is using a Cortex-M0+ processor for a sensor design, and in the next product, it needs additional processing power for an extra data processing function, the design can be upgraded up to a Cortex-M3 or Cortex-M4 processors, with very little change in the system level design and software (e.g. recompile the code to take advantages if the extra instructions).

Summary

Having a processor system in a smart sensors makes the sensor devices much more capable of handling new challenges in today's electronic applications, while still possible to reduce overall system level power.

The low power and efficient nature of the ARM Cortex-M processors make them very suitable for wide range of smart sensors application, and are already used in a number of smart sensor products on the market.