Intelligent Sensor Management

Managing Sensors Using the iCE40 Ultra-Low Density FPGA

Sensors in Smartphones

Sensors are changing the landscape of today's smartphones. According to IMS Research, "Sensors are poised to enable the next stage of mobile handset evolution. Various different sensor types are now showing the potential to alter user input methods, user interfaces, and to enable whole new genres of use cases for mobile handsets." Sensors not only enhance the user interface to smartphones but also are used to monitor environmental conditions such as battery life, temperature, and ambient light. In fact, today's high-end smartphones can contain anywhere from 10 to 20 different sensors.

The Need for Sensor Management

Managing sensors in a mobile system can be challenging because different sensors have different behaviors and operate differently with the processors. Some sensors might need frequent and closer attention from the processor (e.g. gyroscope sensors). The processor has to repeatedly read the new orientation data from the gyroscope sensor even though no movement has been detected. Figure 1 shows the repeated data pattern the processor reads from the gyroscope sensor. This adds extra overhead to the processor and prevents it from entering into the sleep mode. Some users call the gyroscope a "chatty sensor" because it needs constant attention from the processor. Another example is a multi-touch screen in a handset using interrupt mode. When multiple fingers swipe the screen at the same time, it can generate overloaded interrupts to the processors. This can significantly affect processor performance and increase system power consumption.

> " Sensors are poised to enable the next stage of mobile handset evolution. Various different sensor types are now showing the potential to alter user input methods, user interfaces, and to enable whole new genres of use cases for mobile handsets. "
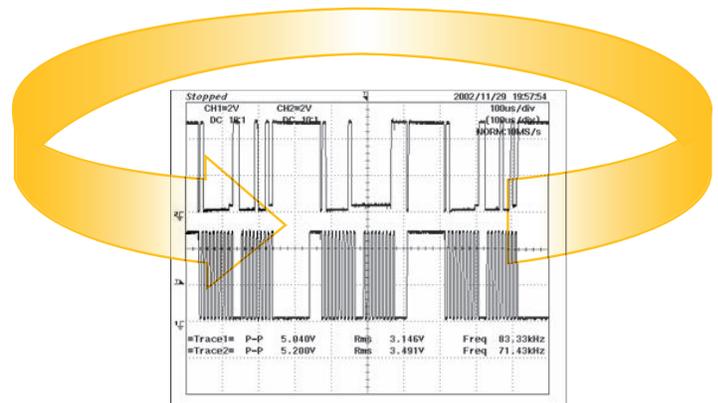>
> IMS Research

Figure 1: Auto-polling in a gyroscope sensor. Reading the same I^2C transaction over time increases processor overhead.
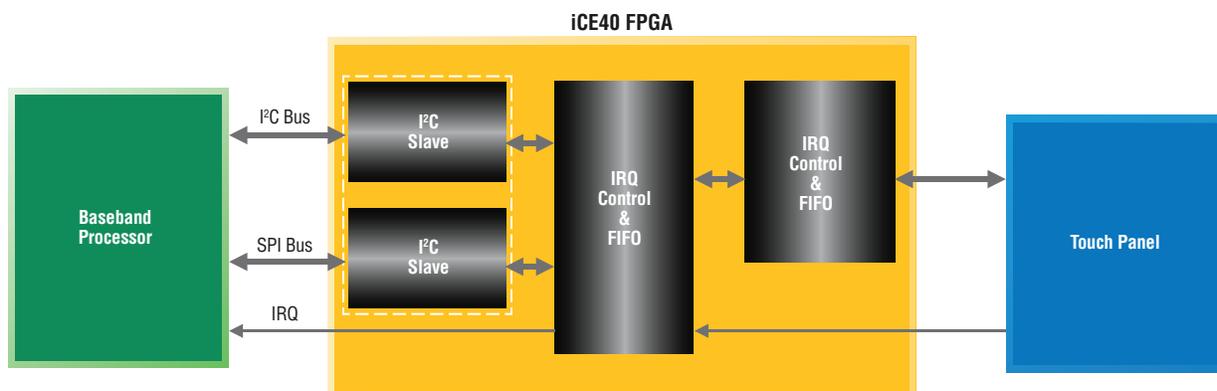

Figure 2: Interrupt Aggregator Design using the iCE40 FPGA

Lattice Solutions

Because different sensors work with processors differently and can potentially affect a processor's overall performance and power consumption, carefully-designed sensor management is critical to solve this problem. By adding some intelligent sensor management logic in an iCE40 FPGA, the designer can offload the processor, improve performance and lower power consumption. Various functions can be incorporated into an iCE40 FPGA, such as:

- Interrupt aggregator (Figure 2)
- Data filtering
- Data pre-processing
- Auto-polling (Figure 4)

These are just a few examples of functions that can be implemented. Software engineers should be consulted before implementing these functions since data/interrupt handling can affect the overall software scheme. Software engineers can also locate the processor bottleneck and pinpoint the exact functions to be implemented.

Auto-polling is very troublesome for the processor and often prevents it from entering into the sleep state. The iCE40 FPGA can be used for the auto-polling to offload the processor. It can further be used to filter out unnecessary data to make it more efficient. Once changed data is detected, iCE40 devices can then send the interrupt to the processor to read back only the data that is required.

Multi-touch screens can generate thousands of interrupts at a given time. This tremendously increases processor overhead and can slow down other tasks if not managed properly. The iCE40 FPGA can be used to aggregate these interrupts into efficient summaries to the processor. Furthermore, the interface to the processor can be customized to any available interface such as a SPI or processor local bus offering extra flexibility in terms of design integration.

Lattice has demonstrated this capability by integrating this solution in a Lattice device interfacing to a Texas Instruments Beagle Board (see Figure 3).
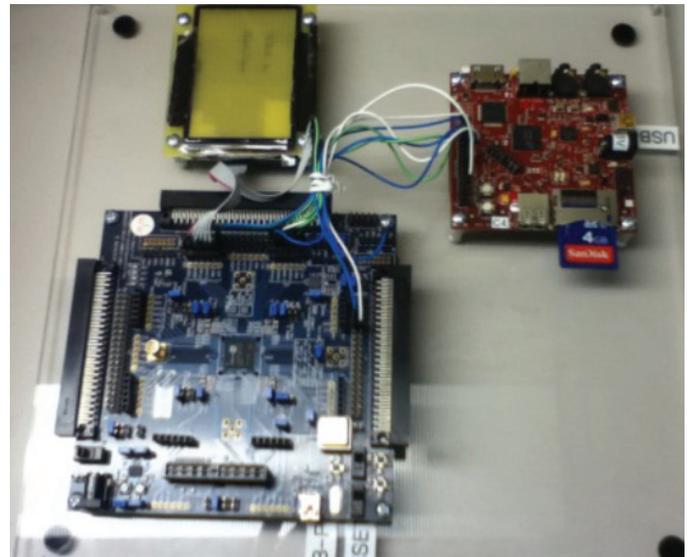

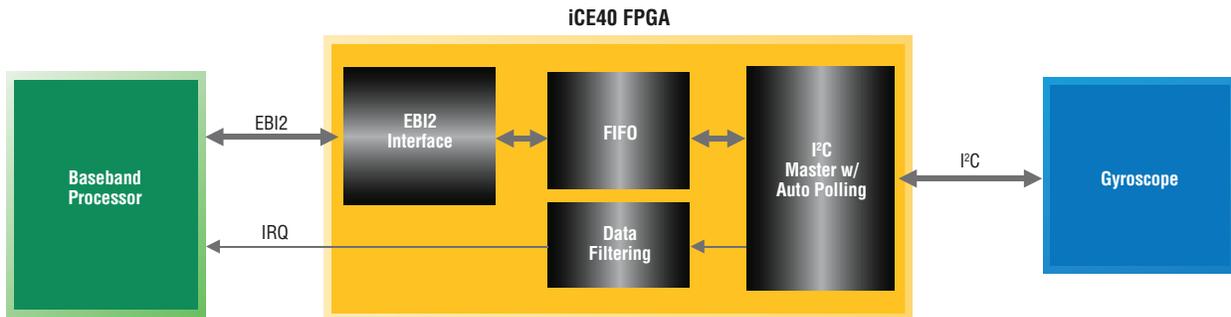*Figure 3: Interrupt Aggregator Demo using TI Beagle Board*

Figure 4: Auto-Polling implemented along with I²C Master to offload processor

October 2012
Order #: SB003