Demo: Building Reliable Wireless Embedded Platforms using the Bolt Processor Interconnect

Felix Sutton, Marco Zimmerling, Reto Da Forno, Roman Lim, Tonio Gsell,
Georgia Giannopoulou, Federico Ferrari, Jan Beutel, and Lothar Thiele
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
{firstname.lastname}@tik.ee.ethz.ch

ABSTRACT

We demonstrate the capabilities of BOLT, an ultra-low-power processor interconnect for the composable construction of new multi-processor wireless embedded platforms. BOLT provides asynchronous bidirectional communication between two processors with predictable message transfer times. In this way, BOLT solves the resource interference problem inherent in today's wireless embedded platforms, enabling simpler and more robust system designs with minimal resource overhead. Using our BOLT prototype implemented on a state-of-the-art microcontroller, we demonstrate BOLT's composability and decoupling in time, power, and clock domains.

Categories and Subject Descriptors

C.1.2 [**Computer Systems Organization**]: Multiprocessors—*interconnection architectures*; C.3 [**Computer Systems Organization**]: Special-Purpose and Application-Based System—*real-time and embedded systems*

General Terms

Design, Experimentation, Performance

Keywords

Cyber-physical systems; multi-processor; processor interconnect; predictability; composability; resource interference

1. MOTIVATION AND DESIGN

Today's wireless embedded applications are characterized by ever-increasing resource demands, the need for run-time adaptability and energy-proportional system operation. At the same time, the employed devices must simultaneously handle several tasks ranging from sensing, actuation, data processing, wireless communications, control law computation, user interaction to power management. The prevailing solution to satisfy these conflicting design goals is the adoption of heterogeneous multi-processor architectures, such as
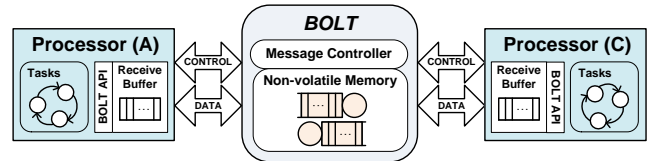
Figure 1: Bolt decouples processors A and C in time, power, and clock domains through predictable bidirectional asynchronous message passing.

the LPC4300 [2] from NXP and VF3xxR and MKW2xDx [1] from Freescale. These heterogeneous architectures facilitate the mapping of a single task to a single processing element, for example, the mapping of sensing and communication tasks onto dedicated processors. However, it has been recognized that whenever multiple components interact, the use of shared buses and shared memory leads to *coupling of power and clock domains* and *interference in the time domain.* Such component dependencies prevent the use of modular design techniques, therefore increasing system complexity and making the design process of multi-processor platforms a labor intensive task. In addition, multi-processor systems are often over-provisioned in order to meet runtime guarantees, thereby infringing on resource and power budgets.

To fill this gap, we have introduced BOLT [3] as a new processor interconnect providing bidirectional asynchronous message passing with predictable message transfer times. BOLT is the first architectural concept that enables the effective decoupling of two processors with respect to time, power and clock domains. BOLT is therefore a key building block for the communication between components on emerging dual-processor platforms to support low-power applications with high dependability requirements and stringent timing constraints. By completely eliminating or at least limiting the interference among different components on shared resources, BOLT simplifies the design of such applications. Consequently, the interconnected processors can schedule their tasks independently and may arbitrarily switch power modes to minimize energy consumption.

As depicted in Fig. 1, BOLT consists of integrated hardware and software that sits between two arbitrary processors A and C. We design and implement BOLT in such a way that the execution time of asynchronous message exchanges can be tightly bounded. BOLT provides a well-defined, non-blocking software interface to each attached processor. The unique properties of BOLT allow for composable system designs. That is, a designer is free to choose any two commercially available processors, build on existing software components, and integrate them to create a customized multi-
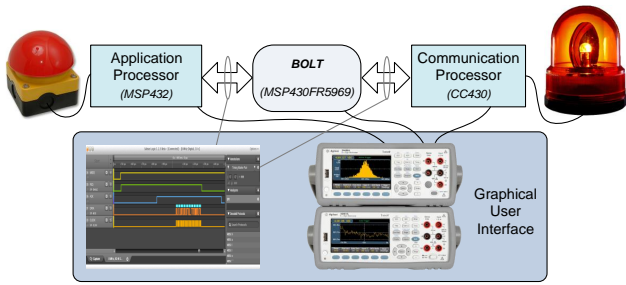
Figure 2: The demo measurement setup allows to visualize sensor events during their traversal through the Bolt architecture. Communication signals are displayed with a logic analyzer along with the power dissipation of all three processors of our prototype.

processor platform that satisfies application needs, *without changing the properties of the integrated components*. Each component can be separately designed, implemented, and validated, thus leading to modular systems with predictable timing behavior that are easier to develop, understand, and maintain.

With only 430 nA current drain during periods of inactivity, BOLT has a negligible impact on the total system power dissipation. The physical interface of BOLT consists of an SPI bus as well as additional lines for handshake signaling. A high throughput of 3.4 Mbps for messages that are 128 bytes in length can be achieved. This is only 15% less than the theoretical throughput of a standard SPI bus.

2. DEMONSTRATION

As illustrated in Fig. 2, the demonstration of BOLT leverages a multimeter and logic analyzer to visualize the power dissipation and message passing in real-time, thus giving insights into the inner workings of BOLT. Furthermore, the two main characteristics of BOLT are demonstrated: BOLT's composability and the decoupling of processors with respect to time, power and clock domains.

While any processor can be attached to BOLT, we will showcase our prototype implementation by facilitating the interconnection of two commercially available off-the-shelf evaluation platforms, namely the STM32 F3 Discovery and the Olimex MSP430-CCRF, as well as two custom development boards featuring an MSP432 application and a CC430 communication processor, as shown in Fig. 3.
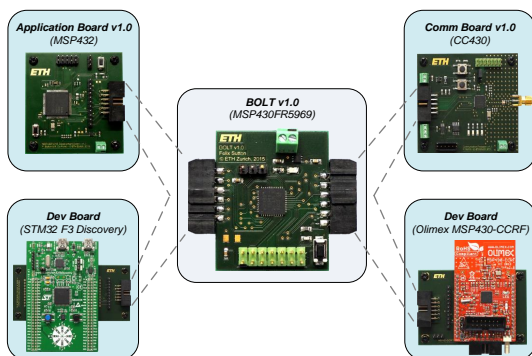

Figure 3: We demonstrate four showcase processors connectible to Bolt: MSP432, CC430, STM32 F3 Discovery and Olimex MSP430-CCRF.
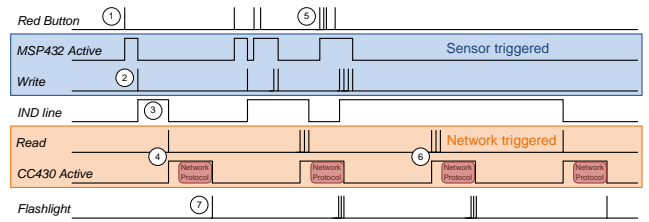

Figure 4: A signal trace from an event-triggered wireless sensing application implemented using Bolt.

In order to demonstrate BOLT's capability, we implement an event-triggered sensing application typical in many wireless sensing application scenarios. In such scenarios, resource-constrained low-power wireless sensor nodes must react to *sensor events* originating from a physical process under observation, while simultaneously reacting to *network events* originating from the wireless network. An example sequence of events and the triggered activities are shown in Fig. 4 and described in the following:

1. A sensor event is generated by pushing a button.
2. An interrupt on the application processor A triggers a wake up from deep sleep mode. After processing the sensor event, an alarm message is written to BOLT and processor A goes back to sleep mode.
3. As soon as the message has been successfully written to BOLT, the *IND* line is asserted to notify the communication processor C of the pending message.
4. Processor C reads the message queue of BOLT just before initiating the next wireless communication activity.
5. A burst of sensor events are buffered in the BOLT message queue.
6. Processor C is not impacted by the burstiness of processor A. In fact, processor C is free to choose how many messages it reads from BOLT, *i.e.*, three messages in this example. The remaining messages are read at the next scheduled communication activity on processor C.
7. Processor C reads an alarm message from BOLT, which is indicated by a flashing lamp.

While the aforementioned use case gives an intuition on what can be supported with BOLT in a very specific scenario, we believe the architectural blueprint provided by BOLT is an important building block that facilitates the construction of reliable and heterogeneous wireless embedded platforms.

Further details on BOLT are available on our website: `http://bolt.ethz.ch`.

3. REFERENCES

[1] Freescale. VF3xxR and MKW2xDx Series. `http://www.freescale.com`.
[2] NXP Semiconductors. Cortex-M4 MCUs with Cortex-M0 Co-Processors. `http://www.nxp.com/products/microcontrollers/core/cortex_m0_m4f/`.
[3] F. Sutton et al. Bolt: A stateful processor interconnect. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2015.