Mobile Applications with Reconfigurable Hardware

Michael Coughlin (student), Ali Ismail (student), Eric Keller
University of Colorado
{michael.coughlin,ali.ismail,eric.keller}@colorado.edu

While there's a great deal of innovation in the software space for smart phones, there's relatively limited ability to innovate in the architecture space – only the large vendors and a very few semiconductor companies are able to introduce new advances, whereas anyone can write software. There are many different examples of new technologies that greatly benefit from hardware support, including new wireless spectrum protocols, hardware-accelerated encryption and transparent scanning of memory or network packets. Many of these technologies can be demonstrated in laboratory environments using powerful PC or FPGA-based platforms, or even custom silicon, but face significant hurdles to general deployment.

We present our early exploration of an alternative architecture to today's processor-centric mobile devices that introduces reconfigurable logic to smart phones, exposes the hardware to applications, and extends a mobile operating system to allow for software control of the current hardware configuration. A processor coupled with programmable logic (PL), such as an FPGA, along with operating system and development support, would open possibilities for developers to introduce new hardware as easily as they are able to introduce new software.

Challenges: Several new technologies have recently matured to a level that they enable FPGAs to be used on mobile platforms. These new technologies are High-level Synthesis, which allows for the creation of FPGA hardware from high-level code (C, C++) [2], Partial Reconfiguration, which allows for FPGA resources to be shared between different applications [3], and modern programmable SOCs, that allow for CPUs and FPGAs to share a single chip [5]. However, these technologies have only been used in traditional architectures in a very limited sense, which presents two primary challenges when applying these technologies to mobile platforms.

The first challenge is to combine these new technologies into a single system that is accessible to application developers. This is difficult to achieve due to the lack of interoperability in the design of these technologies. To address this challenge, a system must be created that allows for application developers to be able to program an FPGA without acquiring a hardware engineer's knowledge and experience, which is what would be required today.

The second challenge is to extend a mobile operating system to provide support for FPGA resources, as most operating systems do not have support for programming FPGAs, and to provide guarantees of operating system primitives, like security and isolation, to FPGA resources, as hardware devices often have the capability to circumvent standard security mechanisms, *e.g.,* by using direct memory access. To address this challenge, a mobile operating system must be extended to allow for the programming of FPGAs and new security and isolation mechanisms should be considered for FPGA devices to provide operating system guarantees.

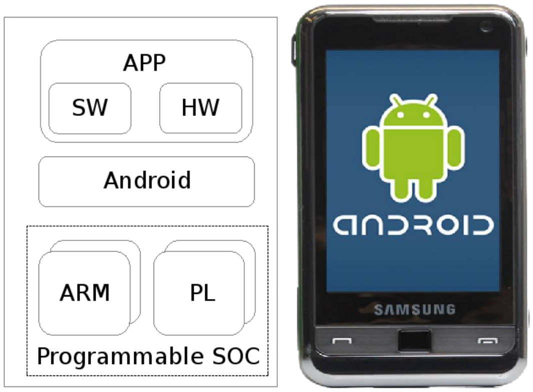

Figure 1: Apps with hardware concept.

Providing OS and Development Support: We are building an architecture that would allow for application developers to specify custom hardware devices along with their software applications and operating system capabilities that allow for applications to write custom hardware for the attached FPGAs. Our architecture is illustrated in Figure 1, specifically targets the Android operating system, as the tools used to interface with FPGAs are already included in certain Linux kernels, on which Android is run [1].

Currently, we have been able to run Android on our target platform, the Zedboard development board from Avnet [4], and integrated both device drivers to interact with custom logic in the FPGA and drivers that allow for software control of the FPGA's partial reconfiguration capabilities. We have also developed tools to completely automate the HLS and FPGA design flow. We are currently working on combining these systems and the creation of appropriate demonstration applications.

References

[1] Android. http://www.android.com/.

[2] Vivado High-Level Synthesis. http://www.xilinx.com/products/design-tools/vivado/integration/esl-design/.

[3] Xilinx Partial Reconfiguration. http://www.xilinx.com/tools/partial-reconfiguration.htm.

[4] Zedboard. http://www.em.avnet.com/en-us/design/drc/Pages/Zedboard.aspx.

[5] Zynq-7000 All Programmable SoC. http://www.xilinx.com/products/silicon-devices/soc/zynq-7000/.