Design Methodology for Custom Reconfigurable Logic Architectures

Johann Glaser

Description

In current trends towards ubiquitous computing and the Internet of Things (IoT), low power consumption is of increasing concern, for example in wireless sensor network (WSN) nodes. One approach to reduce the power consumption is to off-load the CPU by autonomous modules. These relieve the CPU from simple tasks, e.g., performing periodic sensor measurements. The CPU in turn stays in an inactive low-power mode for extended periods. It is only activated if more complex tasks have to be accomplished, such as communicating a new value via the wireless network. Such autonomous CPU supplement modules are implemented as logic designs. These must be reconfigurable to maintain the flexibility for the realization of different applications, to allow adaption to new environments, and to fix design issues. In [Gla15] a new design methodology for the development of such reconfigurable CPU supplement modules is introduced.

Contrary to FPGA design, where chips with a predefined reconfigurable architecture are configured, the proposed methodology includes the development of the silicon circuitry itself. The reconfigurable modules have to support both, control-dominated tasks as well as data processing. To reduce silicon area and power consumption, the approach utilizes a mixed-granularity logic architecture. Besides fine-grained functional units and signals, this adds coarse-grained functional units with more complex functionality and operating on multi-bit vectors. This requires that heterogeneous, i.e., multiple different kinds of functional units, are integrated. This further requires, that each reconfigurable module is specifically developed for its given application domain. The design methodology proposed in [Gla15] addresses this task.

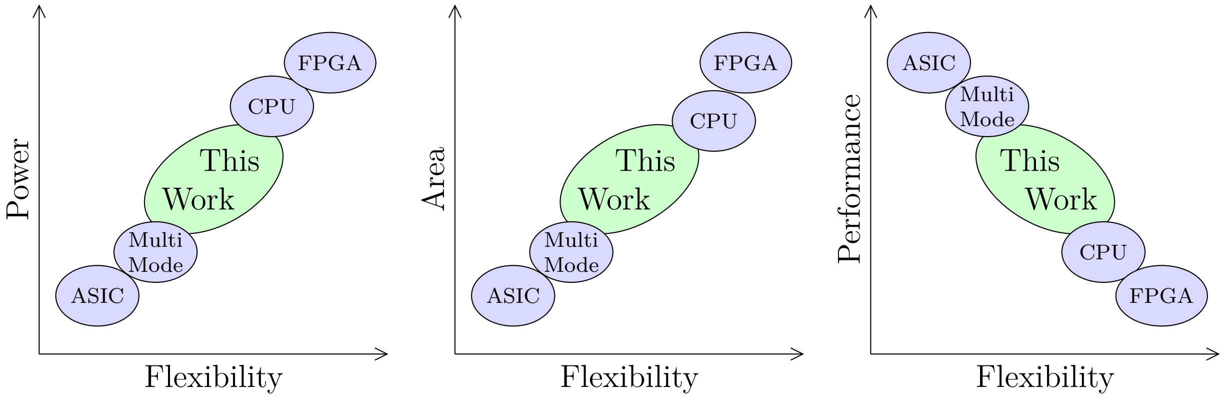
State of the art design methodologies for reconfigurable logic architectures are limited to application domains for data processing but do not support cycle-accurate control-dominated tasks. These approaches either use coarse-grained or fine-grained architectures, but do not provide mixed granularity reconfigurable logic. The functional units of the reconfigurable logic either have to be instantiated explicitly or are mapped manually.

The design methodology introduced in [Gla15] solves these issues. The underlying architecture specifically supports mixed granularity logic. The functionality of reconfigurable modules is specified with a set of example applications as VHDL or Verilog logic designs. These enable the definition of control-dominated as well as data processing tasks including all signals and cycle accurate timing. From these the functional units and the interconnect are optimized in a novel semi-automatic procedure.

The resulting reconfigurable module is delivered as an IP core for the integration in a chip design. It can implement any of the example applications and provides flexibility to implement new applications which were not anticipated during the design process. In order to further increase its flexibility, additional functional units and routing resources can be included during the optimization. The design methodology is the first to incorporate full verification of the example applications, of all intermediate steps, and of the generated reconfigurable module. Simulation and logical equivalence checking are used to ensure compliance to the specification.

The design methodology was implemented as an EDA design flow incorporating custom, open-source, and commercial tools. All tasks which are not essentially manual are automated to assist the designer and to achieve high productivity. The design flow was used to develop an exemplary WSN node SoC including a reconfigurable sensor interface module. This application domain includes control-dominated tasks as well as data processing. The WSN SoC was produced in a 350 nm CMOS process. It correctly implements all example applications and new applications. This demonstrates the feasibility of the design methodology. The reconfigurable module shows a 180-fold reduction in energy consumption for sensor measurements, compared to the integrated CPU. Its chip area is 4.0–4.3 times smaller and it requires 9.1–23.4 times less configuration data than embedded FPGAs. Additionally it provides enough flexibility to implement diverse new applications and to fix design issues of example applications.

To summarize, the following images classify the approach of [Gla15] in terms of flexibility, power consumption, chip area, and performance.



Unique Contributions

- **Design Methodology.** The main contribution of [Gla15] is a design methodology for the development of reconfigurable CPU supplement modules which are optimized for a given application domain. It is the first design methodology which is universal and independent of the application domain of the generated reconfigurable modules. This is achieved by the introduction of the concurrent support of both, control-dominated tasks and data processing. Previous design methodologies were limited to data processing and did not support cycle accurate control-dominated tasks.

The feasibility, applicability, and correctness of the design methodology were demonstrated with a manufactured reconfigurable module which implements a WSN node sensor interface. The experimental results show that the design methodology leads to a functioning reconfigurable module which correctly implements the specified functionality. Additionally, enough flexibility is provided to implement new functionality. The reconfigurable module requires less power compared to a CPU to perform the same task. It requires less chip area and less configuration data than an embedded FPGA.

Besides a sensor interface, the independence of the application domain and the universality of the design methodology enable the implementation of other tasks which are neither related to WSNs nor to low power consumption. Firstly, applications which require a given degree of flexibility benefit from the reduced power consumption, chip area, and size of the configuration data, compared to other reconfigurable or programmable architectures. Secondly, the introduced design methodology enables applications to include flexibility after production, which was previously precluded due to complexity, cost, chip area, or power consumption.

For example, the introduced design methodology enables the implementation of new WSN network protocols. These can be based on frequent operations which were previously precluded due to too high power consumption of the CPU. Additional to the use as a low-power technique in WSN nodes, the resulting reconfigurable modules enable fast responses and constant latency to react on incoming information. With this feature, rapid and ultra-low-power real-time control loops can be implemented. In the research area of software defined radio (SDR), the introduced design methodology fills the gap between software implementations using digital signal processors (DSPs) and fine-grained reconfigurable logic implementations using FPGAs.

Further applications include network protocol handlers, baseband processors, digital filters, CPU accelerators, computer vision preprocessors, gaming controllers, etc. However, an increased degree of flexibility leads to higher power consumption and chip area and lower performance. For a given application the tradeoff between these properties determines the benefit gained by the utilization of the introduced design methodology.

The design methodology is based on RTL HDL designs and therefore independent of the semiconductor process. The development of a reconfigurable module and its integration into a chip design is straightforward and uncomplex. Therefore it is also suitable for small designs, wherever post-silicon flexibility is required, for example to avoid the use of an embedded or external FPGA, or within an FPGA if no partial reconfiguration is available or desired. Further, the results achieved with the design methodology are verified, i.e., the integration of reconfigurable modules is a secure design decision which does not cause additional risks.

The design methodology effectively closes the gap between fixed function logic designs and fully flexible embedded FPGAs.

- **Transition-Based Reconfigurable FSM (TR-FSM).** The second core contribution of [Gla15] is a novel architecture for reconfigurable FSMs, termed “Transition-Based Reconfigurable FSM” (TR-FSM). The TR-FSM is used in the reconfigurable modules to efficiently implement control-dominated tasks. Its feasibility, applicability, and correctness were also demonstrated with the manufactured reconfigurable module.

The TR-FSM architecture reduces the power consumption, chip area, configuration data, and propagation delay time compared to other reconfigurable architectures for FSMs, and allows straight-forward mapping algorithms.

The TR-FSM is an independent design unit which can be integrated in SoC designs wherever an efficient reconfigurable FSM implementation is required. It is available as a VHDL RTL design with a number of generics which allow full customization of the silicon implementation to the requirements of the designer. Its actual functionality can be specified in various formats, from which the configuration data is generated by a dedicated tool. The configuration data is provided for convenient integration in design verification, firmware drivers, and in custom configuration infrastructure.

References

[Gla15] Johann Glaser. *Design Methodology for Custom Reconfigurable Logic Architectures.* PhD thesis, TU Wien, 2015.