Desynchonization Tool for High-Level Synthesis of Asynchronous Circuits

Jean Simatic, Rodrigo Possamai Bastos, and Laurent Fesquet
Univ. Grenoble Alpes, TIMA Laboratory, F-38031 Grenoble
CNRS, TIMA Laboratory, F-38031 Grenoble
Email: {jean.simatic, rodrigo.bastos, laurent.fesquet}@imag.fr

***Abstract*—Event-triggered systems are suitable to address the power issues raised by the Internet of Things. We present a tool for the high-level synthesis (HLS) of event-driven (asynchronous) circuits.**

I. Context and design flow

The power consumption of a signal processing chain (analog-to-digital converter (ADC) + digital processing) is driven by both the number of samples and the complexity of the processing. Non-uniform sampling techiques (such as level-crossing) can drastically reduce the number of samples with a reasonnable complexity cost. Moreover, by using event-driven (asynchronous) logic, the processing part naturally stands by while waiting for the next sample.

The target architecture is composed of an ADC, and a digital processing unit. Figure 1 describes the design flow starting from the application and down to the netlists of those two components. The rest of the flow is standard.

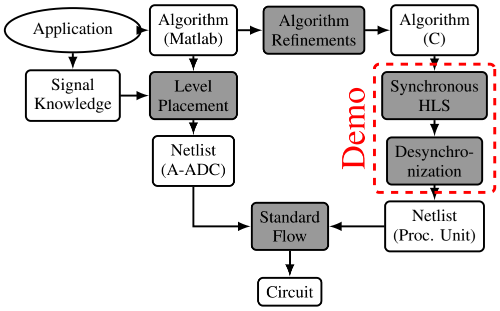


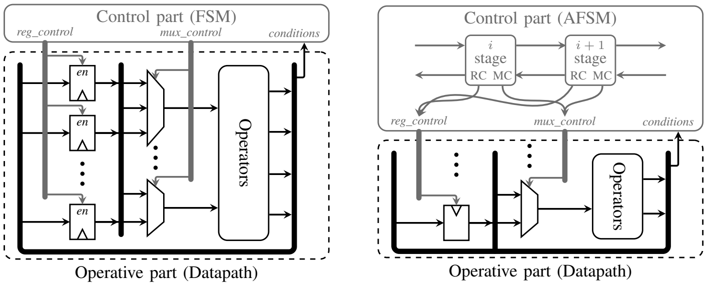
Figure 1. Design flow for ultra-low power applications.

For the ADC, the flow targets an extension of Allier's level-crossing ADC [1] in order to benefit from the sparse sampling. Some methods exist for choosing the levels [2], [3]. They depend on both the type of algorithm and a *a priori* knowledge about the signals (*e.g.* signal database).

For the processing part, the tool firstly translates the Matlab algorithm into C. The C code takes into account the hardware-related limitations (static memory allocation and finite operator precision). Then, the HLS tool AUGH [4] derives a RTL description of a synchronous implementation that is then desynchronized by our tool.

II. Desynchronization principle

AUGH generates a FSM (control part) and a mux-based data-path (operative part) as shown in Figure 2a. To desynchronize the FSM, the presented tool generates a distributed asynchronous controller that duplicates the behaviour of the state machine. Indeed, the FSM controls the multiplexers and enables the registers in the data-path. Thus, even if the data-path components do not change, its processing has an asynchronous behaviour.



(a) Generated by synchronous HLS

(b) After desynchronization

Figure 2. Model of circuits

III. Tool anatomy

The desynchronization performs the following operations:

a) VHDL parsing: The tool analyzes the top entity to determine the role of each FSM output (mux or register control) and input. By parsing the FSM entity, the tool builds the state graph corresponding to the FSM.

b) Graph transformation: The tool transforms the graph into a form closer to the actual implementation. For example, loops where a state is its own successor require a special treatment.

c) FSM netlist generation: For each state, the tool generates an asynchronous controller. If a state has several successors (resp. predecessors), additional controllers implementing a split (resp. a merge) are generated.

IV. Conclusion

Tested on a GCD algorithm and a digital (FIR) filter, the desynchronized versions proved small (<5%) area overhead and similar computing times compared to the synchronous one. Also, longer critical paths have less impact thus allowing to optimize the data-path area.

The presented desynchronization tool is a key element in a design flow for ultra-low power applications. This flow follows an event-driven paradigm in both the sampling (level-crossing scheme) and the digital processing (asynchronous circuit).

References

[1] E. Allier, "Asynchronous analog to digital interface: a new class of converters based on time quantization," Ph.D. dissertation, Institut National Polytechnique de Grenoble - INPG, Nov 2003, iSBN 2-84813-016-4.

[2] G. Roa, T. Le Pelleter, A. Bonvilain, A. Chagoya, and L. Fesquet, "Designing ultra-low power systems with non-uniform sampling and event-driven logic," in *Integrated Circuits and Systems Design (SBCCI), 2014 27th Symposium on*, Sept 2014, pp. 1–6.

[3] C. Arslan, J. Poujaud, and L. Fesquet, "A method to automatically determine the level-crossing thresholds in non-uniform sampling and processing," in *Event-based Control, Communication, and Signal Processing (EBCCSP), 2015 International Conference on*, June 2015, pp. 1–4.

[4] A. Prost-Boucle, O. Muller, and F. Rousseau, "Fast and standalone design space exploration for high-level synthesis under resource constraints," *Journal of Systems Architecture*, vol. 60, no. 1, pp. 79–93, 2014.