μSystems Research Group

School of Electrical and Electronic Engineering

Newcastle University

Design Methods for Minimum Energy Point Asynchronous Processors in the Internet of Things

A. Wheeldon

Technical Report Series

NCL-EEE-MICRO-TR-2017-204

January 2017

Contact: a.r.wheeldon2@ncl.ac.uk

Supported by EPSRC grant GR/XXXX

NCL-EEE-MICRO-TR-2017-204

μSystems Research Group

School of Electrical and Electronic Engineering

Merz Court

Newcastle University

Newcastle upon Tyne, NE1 7RU, UK

http://async.org.uk/

Design Methods for Minimum Energy Point Asynchronous Processors in the Internet of Things

A. Wheeldon

January 2017

Abstract

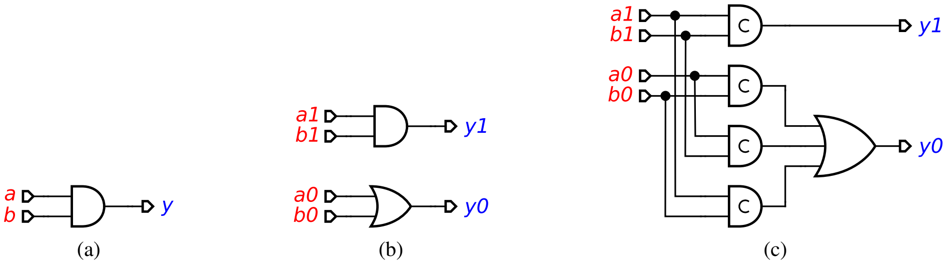
As internet of things devices become increasingly abundant, the need to minimise their energy consumption becomes evermore important. This research will concentrate on processor design for the internet of things, specifically focused on the use of asynchronous (clockless) digital logic. While asynchronous design has been around for decades and can offer many advantages, it remains underutilised in industry, presumably due to its more formal design process and underdeveloped design tools when compared with synchronous design. The research covers four areas of processor design using asynchronous design methodologies, and works towards wider adoption of asynchronous design. The areas include design partitioning, memory architecture, parameterised circuits, and instruction set architecture; all of which will enable improved energy efficiency. Throughout, special consideration will be given to the design process, and effort made to improve existing (or create new) design flows in an attempt to push asynchronous design into wider use.

1 Introduction

Many internet of things (IoT) devices rely on batteries or energy harvesting for power. The low power density and sparse energy natures of these power sources puts increasing pressure on circuits to be as energy efficient as possible. Additionally, with the always-on nature of many of these devices, energy efficiency becomes arguably more important. This energy efficient requirement has lead to a quest to find the minimum energy point (MEP) of these systems. It has been shown that the MEP of many systems lies in the sub-threshold region of operation where the supply voltage is below the transistor threshold voltage (some examples are [1, 2]).

Unfortunately, operating in the sub-threshold region introduces severe uncertainty in gate delays, forcing traditional synchronous circuits to incorporate large timing overheads to increase reliability. By using very conservative timing, synchronous circuits can achieve reliable operation under most operating conditions but at a vastly reduced speed.

Asynchronous circuits are sequential circuits which operate without a clock. Speed-independent (SI) circuits are a subset of asynchronous circuits which take as long as necessary to compute under the current runtime



(a) (b) (c)

Figure 1: Gate-level implementations of AND gates for (a) single-rail, (b) dual-rail NCL-X, and (c) dual-rail NCL-D asynchronous design styles. AND gates denoted with the letter ‘C’ represent the C-element [3] – a frequently used memory element in asynchronous design.

conditions. This helps to alleviate the reliability shortcomings of synchronous circuits. In addition, SI circuits inherently adapt their speed depending on the runtime gate delays and therefore automatically increase their speed as the operating conditions allow.

Bounded delay (BD) asynchronous circuits are less reliable than SI circuits, but more-so than synchronous circuits under sub-threshold conditions. Their advantages are reduced power and area overheads compared to SI circuits.

There are a few different asynchronous design styles which fall into the SI and BD categories. *Single-rail* is a BD design style which uses the same combinational logic as in a synchronous system, but introduces a delay line to perform handshaking in place of the clock. It achieves this by modifying an existing synchronous design. This design style requires minimal power and area overhead versus a synchronous design.

The *dual-rail* design style belongs in the SI category[1]. Dual-rail circuits generate a negated output as well as the normal output signal making them very reliable. However, this can require many times the number of gates compared to the synchronous implementation, causing large power and area overheads. Figure 1 compares implementations of an AND gate. NCL-D[2] is the most robust, but also requires the most logic. NCL-X is a compromise between single-rail and NCL-D at the expense of reduced robustness.

A few IoT projects have recently addressed the issue of energy efficiency for specific applications (ReISC [4], PULP [5], Bellevue [6], among others). The author offers a different approach through the use of SI and BD asynchronous circuits in the design of an MEP processor suitable for *general purpose* IoT applications. The processor will implement the ARM Cortex-M0+ instruction set architecture (ISA) which is widely used in the IoT space, allowing easy adaptation of existing products to the new processor. Designed with low power and energy efficiency in mind [7], it makes the best ISA candidate for an MEP design[3].

The proposal is arranged as follows: Section 2 reviews previous work in relevant areas; Section 3 summarises the research that will be undertaken throughout the project; and finally Section 4 presents a planned timeline of events.

[1] Dual-rail is the simplest implementation of the ‘m-of-n’ encoding technique. It is so-called 1-of-2 since only one rail may be high out of the two rails per bit.

[2] Null convention logic (NCL)

[3] It should also be noted that the author has previous experience [8] with the ARM architecture.

2 Background

This section discusses previous works in the areas of asynchronous processors, asynchronous systems synthesis and system modelling. Each work's relation to the proposed project is given.

Asynchronous Processors

Although several papers on asynchronous processor design exist, there is evidently a lack of coverage of operation in sub-threshold.

Furber et al. [9] have proven a commercially viable asynchronous processor is possible during the AMULET projects. However, these implementations are almost two decades old, and the synchronous counterparts on which they are based have long been superseded. In addition, the entire AMULET3 processor core was designed by hand without the aid of automation tools. The proposed project can reflect upon the downfalls of the AMULET projects and improve the design process by introducing automation.

Research has shown that traditional static RAM (SRAM) is a limitation for decreasing the operating voltage of a processor [10, p. 142]. There is a need to overcome this limitation to allow MEP operation of a processor. The memory subsystem is one of the most energy hungry in a processor, it is therefore an important subject for study. [11] demonstrates SRAM operating down to 0.36 V. The memory incorporates read/write completion signals which are used to self-time the otherwise-synchronous processor. It is possible that this SRAM architecture could be used in a fully asynchronous processor design.

As discussed, the processor in [11] is mainly synchronous with clocks generated from memory completion signals. While this may be a good architecture for systems limited by memory speed, it does not help for systems operating in sub-threshold with highly variable gate-delays where the processor core may be the speed bottleneck. A fully asynchronous system could adapt to both of these scenarios at runtime.

Asynchronous Synthesis

Desynchronisation [12] is a method of asynchronous logic synthesis. This method of synthesis follows very closely with the industry standard synchronous design flow. It allows designers trained in synchronous synthesis to rapidly develop asynchronous systems without the need to drastically change the design flow. Desynchronisation is a promising synthesis method for bringing asynchronous design to wider adoption, making it an important subject for investigation in this research. Tools exist for performing automatic desynchronisation of a behavioural design (eg. Pipefitter [13]). These tools will be investigated as part of the research.

Mokhov et al. [14] show a method for optimising the amount of completion detection circuitry for a given design. Contrastingly, *De Genarro* [15, p. 101] uses a mixture of NCL-X and NCL-D gates to design a minimal yet reliable circuit; using less area than a full NCL-D implementation. These design techniques can be compared for use in designs which benefit from dual-rail logic and will be important when designing for MEP and wide

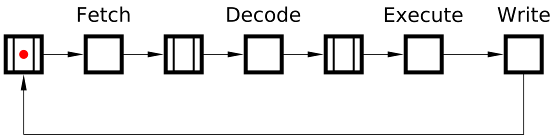


Figure 2: A dataflow structure for a simple RISC processor. The classic fetch, decode and execute stages can be seen. Here, execute and write operations happen in the same functional unit.

operating voltage (covering both super-threshold and sub-threshold operation).

Logic parameterisation [16] is a method of selecting different circuit implementations at runtime. This technique can allow a system to choose between a robust circuit implementation for sub-threshold operation, and a low energy implementation when the supply voltage is sufficient enough. This technique may help to achieve higher energy efficiency over a wide operating voltage when compared with designs using only one circuit implementation.

Modelling

Dataflow structures [17, p. 6] are graph models which allow easy modelling of large asynchronous systems. They use the notion of tokens to pass data from one functional block to the next. These models allow easy separation of systems into *datapath* and *control* [18] which is required for asynchronous logic synthesis. Such models will be invaluable when designing a complex processor. Figure 2 shows an example dataflow structure for a RISC processor. WORKCRAFT [19] can be used to draw and simulate such models.

Mokhov has created the *conditional partial order graph* model [20] aimed at modelling of microcontroller ISAs. This work can formalise the specification of an ISA, and additionally, allow direct synthesis of this specification to an area efficient instruction decoder. This work will be paramount in the design of an efficient ISA for asynchronous processors.

3 Research

This section poses some research questions to be investigated during the project and how they will be answered. The section is divided into the four main areas identified for research, namely; *design partitioning*, *memory architecture*, *parameterised circuits* and *ISA*. From the research in these areas, it is expected that tools and methods will be produced, enabling lower energy processor designs than those which exist today for the IoT market.

Design Partitioning

As pointed out in Section 1, several design styles for asynchronous circuits exist, each with different merits in terms of energy consumption and reliability. The research will investigate whether it is worthwhile partitioning an asynchronous processor design in order to utilise multiple design styles. This question will be answered in the context of minimising energy per instruction over a wide operating voltage. Such partitioning could be performed by processor subsystem (registers, instruction decoder, etc.), or on a more fine-grained level. Partitions may need additional interfacing logic due to their differing design styles. The interfacing logic overhead will be taken into account when choosing potential partitioning schemes. The chosen design style for each partition may depend on the processor's overall critical path, the switching activity of the partition[4], or a number of other factors. Dataflow structures will be used to model a processor in order to facilitate such partitioning.

Memory Architecture

Following on from the discussion of memory for MEP in Section 2, the research will investigate possible solutions such as a tiered memory architecture. Such an architecture would disable access to SRAM at supply voltages below its functional limit, resulting in a reduced instruction set when the processor operates close to its MEP. The restricted SRAM access could be of detriment to applications where the processor spends most of its time operating near the MEP. It is also likely that existing code would need to be modified to detect the availability of SRAM at runtime.

A different approach to solving the SRAM operational voltage limitation is to use a multiple supply voltage architecture. The SRAM would operate at the minimum voltage required for full functionality, whereas the processor core would operate at a separate and lower voltage. This approach would likely require level shifting logic between the two subsystems, therefore increasing power, energy and area overhead. However, by using this approach, modification to existing application code could be avoided.

Logic Parameterisation

As discussed in Section 2, *logic parameterisation* allows different circuit implementations to be utilised at runtime. The research will find whether this technique provides advantages in a processor design for a wide operating voltage. It is expected that a mixture of BD and SI implementations will provide the optimal solution, with parameterised circuits allowing a hybrid design to be produced.

As an extension to parameterised circuits, the research may look to repurposing some of the 'switched off' circuitry in a parameterised circuit for other operations. As an example, this could allow for increased concurrency when operating at a higher supply voltage, allowing for greater performance.

[4] The switching activity of a digital circuit is the probability of an energy consuming transition occurring within the circuit for any given input. It is a function of all the gates in the circuit.

ISA

The suitability of the Cortex-M0+ ISA will be analysed with reference to sub-threshold and asynchronous operation. There may be opportunities for power, energy, and/or area optimisations for the asynchronous implementation since the ISA is designed with synchronous operation in mind. Care must be taken however, to preserve backward compatibility with existing code to ensure a smooth transition to the new processor design for existing IoT products. *De Gennaro's* previous work on the area-efficient synthesis of a Cortex-M0+ instruction decoder using conditional partial order graphs [21] will be used as a starting point for this investigation.

4 Strategy

It is expected that a complete processor design will be produced and manufactured as a result of this project in order to prove the design methodologies developed. The target process is likely to be 65nm TSMC for which sub-threshold cells are currently in development as part of a separate project.

The *Verilog* hardware description language (HDL) will be used as a starting point to describe a synchronous processor. *Cadence Incisive Simulator* will be used to verify the behavioural design. *Cadence RTL Compiler* will be used to synthesise a gate-level design from the behavioural HDL which can be used for analogue simulation. The EEE department currently holds licenses for all aforementioned software and intends to do so for the foreseeable future. Following on from the synchronous design, a basic asynchronous processor will be produced allowing single- and dual-rail workflows to be investigated. This design will then be adapted for compatibility with the Cortex-M0+ ISA. This will allow the research to contrast current asynchronous design styles for the application of MEP processor design. Synthesis of the asynchronous control logic will be performed using *Workcraft* [19] – an open source tool actively developed by the µSystems Research Group.

A brief summary of work over a three year period follows.

Year 1

- Evaluate asynchronous workflows through the design of a simple processor.
- Investigate memory architecture for MEP operation.
- Tape out an asynchronous memory subsystem in the sub-threshold process.
- Prepare a test methodology for the memory subsystem.
- Explore design partitioning techniques for an asynchronous Cortex-M0+ design.

Year 2

- Receive the manufactured memory and evaluate its operating voltage range.
- Begin adaptation of the asynchronous processor to suit the Cortex-M0+ ISA.

- Investigate possible advantages of logic parameterisation in the design.
- Explore potential ISA improvements for the asynchronous Cortex-M0+ design.

Year 3

- Tape out the asynchronous Cortex-M0+ design in the sub-threshold process.
- Prepare test methodology for Cortex-M0+ design.
- Receive manufactured Cortex-M0+ design and test over a wide operating voltage.
- Collect empirical results and write thesis.

Should the work be completed ahead of schedule, the project could be expanded by investigating asynchronous bus architectures and microcontroller peripherals. This would work towards a full microcontroller implementation for the target applications.

References

[1] A. Wang and A. Chandrakasan, “A 180-mV subthreshold FFT processor using a minimum energy design methodology,” *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Jan. 2005.

[2] J. Myers, A. Savanth, R. Gaddh, D. Howard, P. Prabhat, and D. Flynn, “A Subthreshold ARM Cortex-M0+ Subsystem in 65 nm CMOS for WSN Applications with 14 Power Domains, 10T SRAM, and Integrated Voltage Regulator,” *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 31–44, Jan. 2016.

[3] D. Muller, “Theory of Asynchronous Circuits,” Univeristy of Illinois, Tech. Rep., 1955.

[4] N. Ickes, Y. Sinangil, F. Pappalardo, E. Guidetti, and A. P. Chandrakasan, “A 10 pJ/cycle ultra-low-voltage 32-bit microprocessor system-on-chip,” in *2011 Proc. ESSCIRC*. IEEE, Sep. 2011, pp. 159–162.

[5] F. Conti, D. Rossi, A. Pullini, I. Loi, and L. Benini, “Energy-efficient vision on the PULP platform for ultra-low power parallel computing,” in *2014 IEEE Work. Signal Process. Syst.*, Oct. 2014, pp. 1–6.

[6] F. Botman, J. de Vos, S. Bernard, F. Stas, J. D. Legat, and D. Bol, “Bellevue: A 50MHz variable-width SIMD 32bit microcontroller at 0.37V for processing-intensive wireless sensor nodes,” in *2014 IEEE Int. Symp. Circuits Syst.*, Jun. 2014, pp. 1207–1210.

[7] (2016, December) Cortex-M0+ Processor – ARM. [Online]. Available: https://www.arm.com/products/processors/cortex-m/cortex-m0plus.php

[8] A. Wheeldon, “Design of an ARM Cortex-M0 DesignStart Compatible CPU Core,” Master’s thesis, University of Southampton, 2015.

[9] S. Furber, D. Edwards, and J. Garside, “AMULET3: a 100 MIPS asynchronous embedded processor,” in *Computer Design, 2000. Proceedings. 2000 International Conference on*, 2000, pp. 329–334.

[10] M. Rykunov, “Design of asynchronous microprocessor for power proportionality,” Ph.D. dissertation, Newcastle University, 2013.

[11] H. Fuketa, D. Kuroda, M. Hashimoto, and T. Onoye, "An Average-Performance-Oriented Subthreshold Processor Self-Timed by Memory Read Completion," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 58, no. 5, pp. 299–303, May 2011.

[12] J. Cortadella, A. Kondratyev, L. Lavagno, and C. Sotiriou, "Desynchronization: Synthesis of Asynchronous Circuits From Synchronous Specifications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 10, pp. 1904–1921, Oct. 2006.

[13] I. Blunno and L. Lavagno, "Automated synthesis of micro-pipelines from behavioral Verilog HDL," in *Proc. - Int. Symp. Asynchronous Circuits Syst.* IEEE Comput. Soc, 2000, pp. 84–92.

[14] A. Mokhov, D. Sokolov, and A. Yakovlev, "Completion Detection Optimisation," Newcastle University, Tech. Rep., 2005.

[15] A. De Gennaro, "Design of Reconfigurable Dataflow Processors," Master's thesis, Politecnico di Torino, 2014.

[16] A. Mokhov, D. Sokolov, and A. Yakovlev, "Adapting Asynchronous Circuits to Operating Conditions by Logic Parametrisation," in *2012 IEEE 18th Int. Symp. Asynchronous Circuits Syst.* IEEE, May 2012, pp. 17–24.

[17] J. Sparsø and S. Furber, *Principles of Asynchronous Circuit Design: A Systems Perspective*. Kluwer Academic Publishers, 2001.

[18] D. Sokolov, I. Poliakov, and A. Yakovlev, "Analysis of static data flow structures," *Fundam. Informaticae*, vol. 88(4), pp. 581–610.

[19] (2016, December) Workcraft. [Online]. Available: http://www.workcraft.org/

[20] A. Mokhov, "Conditional Partial Order Graphs," Ph.D. dissertation, Newcastle University, 2009.

[21] A. De Gennaro, P. Stankaitis, and A. Mokhov, "A heuristic algorithm for deriving compact models of processor instruction sets," in *Application of Concurrency to System Design (ACSD), 2015 15th International Conference on*, June 2015, pp. 100–109.