# CURRENT-SENSING COMPLETION DETECTION METHODOLOGY AND ITS IMPLEMENTATION IN ASYNCHRONOUS SYSTEMS

**Lukáš Nagy**
Microelectronics, $4^{th}$ class, full-time study
Supervisor: Viera Stopjaková

Faculty of Electrical Engineering and Information Technology
Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava 1, Slovakia

lukas.nagy@stuba.sk

**Abstract.** This paper addresses an alternative approach to detecting the completion of computation in asynchronous digital circuits. Presented method, as the name suggests, is based on sensing the amount of current drawn by the combinatorial logic and exploits the behavior of CMOS logic gates for separating the computation state from the idle state. The paper deals with the current sensor design, its implementation into the various asynchronous pipeline systems, and discusses the advantages over the conventional approaches.

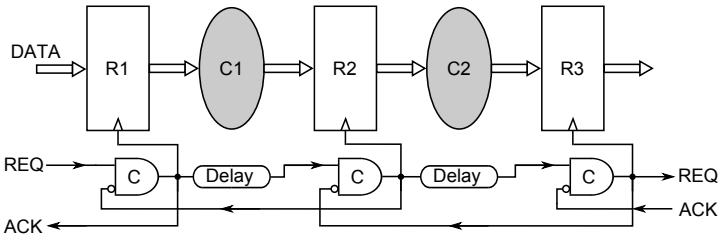
**Keywords.** Completion detection, Current sensing, Low-power, Asynchronous systems

## 1 Introduction

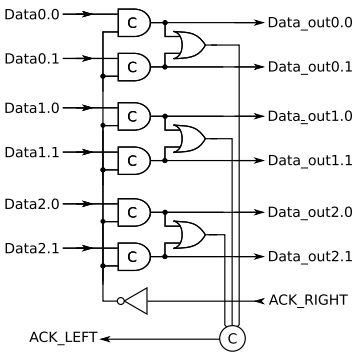
Asynchronous circuits provide several advantages compared to their synchronous counterparts. First and also most important is the complete avoidance of *clock skew*, which has become the major issue in recent years, especially in deep sub-micron technologies. Another substantial advantage is a great adaptability to process, power supply voltage and temperature (PVT) variations, since there is no strictly defined timing window in clockless systems. Significant benefits also include lower EMI radiation, average-case performance and "data-driven" operation. Known drawbacks are engineers' widespread unfamiliarity with the topic of asynchronous systems, difficulties with testing, almost no support from EDA tools developers and higher silicon area overhead, and therefore, higher price. Unlike synchronous systems, where the timing is strictly dictated by the global clock signal, asynchronous circuits relay on distributed local control provided by so-called *handshaking*. Handshaking protocol generates controlling signals according to the computation process of combinatorial logic and therefore, the controlling block needs to know when the combinatorial logic has finished its job. So far, the detection of computation completion has been performed by employing simple logic gates. However, in complex and/or wide-bit information systems, this approach could introduce significant timing margin and silicon area overhead. These are the reasons why we were motivated to develop an alternative way of handshake signals generation, and this way, reduce the timing margin or reduce the silicon area consumption.
The objectives of PhD thesis contain the design and implementation of the presented methodology into a real asynchronous system, carrying out the performance analysis and comparison to the state-of-the-art approaches. Moreover, we intend to develop a methodology for front-end as well as back-end design flow. Furthermore, the hardware testing should be developed as well.

## 2 State-of-the-art of completion detection

As we stated in previous section, the completion detection in asynchronous systems has been carried out by using simple logic cells. Figure 1 depicts the state-of-the-art completion detection circuitry in single-rail encoding as well as in the dual-rail encoding. Figure 1a shows the single-rail pipeline, where the real




(a) Completion detection in single-rail pipeline




(b) Completion detection in dual-rail pipeline

Figure 1: State-of-the-art completion detectors in single-rail and dual-rail implementation

computation time of the combinatorial part is rather simulated by a matched worst-case delay element. Advantage of this solution is in its simplicity. The delay element can be very easily implemented by using simple inverters. On the other hand, worst-case delay introduces a timing margin, which slows down the overall performance. Figure 1b depicts the dual-rail latch along with the completion detection circuitry. OR gates are used to detect the valid codeword at the block output, and their outputs are then synchronized by means of multi-input Muller C element. Again, the advantage of the presented methodology is based on its simple implementation. Basic 2-input Muller C elements can be created by a single standard cell. However, multi-input C elements are usually created by cascading 2-input C elements together. This may introduce a significant drawback when the system works with wide-bit information. Taking all this into account, the completion detection methodology proposed, developed and implemented within our work, promises to provide variable portion of delay instead of a fixed worst-case delay elements while applied in single-rail circuits, and also promises to reduce silicon area requirements in wide-bit information dual-rail systems.

## 3 Current-Sensing Completion Detection (CSCD)

The original idea of exploiting the current consumption of digital CMOS circuits for the computation completion detection was mentioned for the first time in [1]. First practical concept of current sensor along with a proposed implementation was published a few years later in [2]. Substantial portion of research has been done in [3, 4, 5, 6]. However, topologies proposed in these publications would not be applicable in (very) deep sub-micron technologies due to employment of resistors. This would require unacceptable large silicon area in modern CMOS processes. Furthermore, some of them also used bipolar transistors, which are not available in advanced CMOS technologies anymore. Another disadvantage that these sensors exhibit, is that their topology has not been verified for low-power / low-voltage applications. In [7], authors have developed and successfully implemented a CSCD into a single-rail asynchronous circuit with interesting results. However, they do not compare the system performance with matched worst-case delay but with synchronous version, which seems to be rather irrelevant. There has also been done research in the area of testing of asynchronous circuits based on CSCD [8].

## 3.1 Design of the current sensor

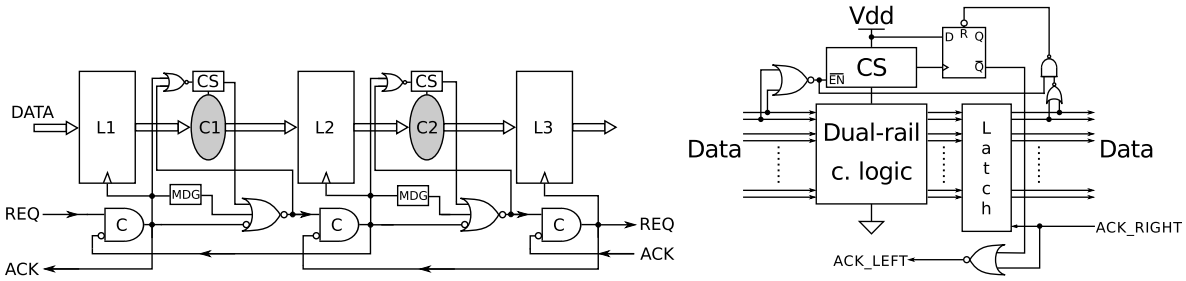
Crucial part of the CSCD methodology is the current sensor itself. Essentially, it is an analog circuit generating digital signal accordingly to the current flow from the power supply rail into the combinatorial logic. Specifications of such circuitry are focused mainly on minimum silicon area, low quiescent current consumption, sufficient slew rate and input/output delays, and most importantly on the voltage drop across the sensing element. As expected, these specifications are in contradiction with one another and a reasonable trade-off has been found.

The current sensor schematic diagram on the transistor level along with the devices' dimensions, corner analysis and Monte Carlo simulations were published in [9, 10]. We also investigated the sensor under the lowered power supply voltage in order to verify its performance in low-power / low-voltage applications. The sensor was still able to function properly with the power supply voltage of 550 mV, which represents about 45 % of the nominal value. The crucial timing parameters as well as the threshold level dependency and overall parameters were also addressed in [10].

Layout of the proposed sensor was developed as a standard cell of double height. This is fully acceptable for the automated design procedure, since advanced digital libraries contain standard cells with double height. These cells are used for switching the voltage levels for certain logic blocks. The power supply rail for the sensor as well as the ground rail are formed by the first metal layer since the design-kit standard cells use it too. The rail for the combinatorial logic is created on the second metal layer. The combinatorial logic power rail (on the first metal layer) will have to be isolated from the rest of the circuit and connected to the sensor by a via contact.

## 3.2 CSCD Implementation

Figure 2 depicts the proposed CSCD implementation into the single-rail as well as the dual-rail asynchronous circuits. In figure 2a, the current sensor is enabled by means of a simple NOR gate when the




(a) Implementation into single-rail pipeline

(b) Implementation into dual-rail pipeline

Figure 2: CSCD implementation in single-rail and dual-rail pipelines

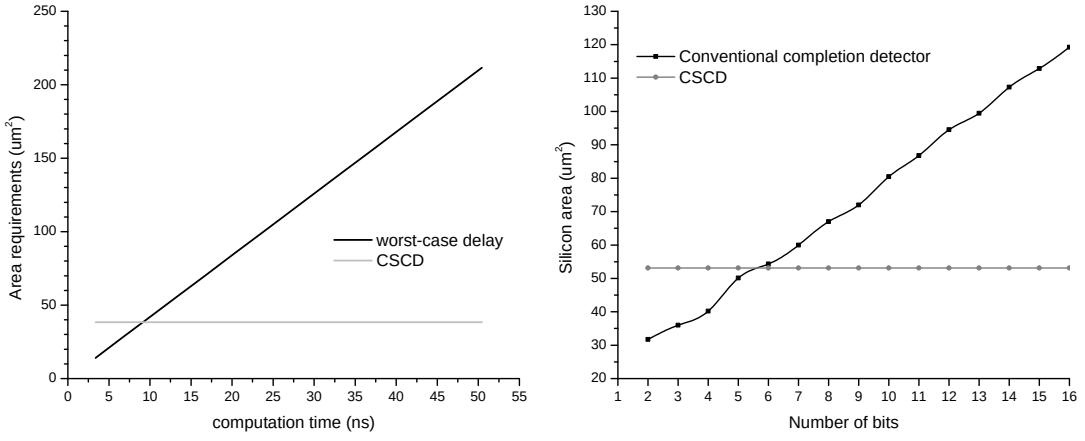
signal for enabling latch L1 is set to high state. Minimum Delay Generator (MDG) is used to generate a pulse in order to compensate the sensing delay of the current sensor [2]. Current sensor generates a pulse (active high) at its output with a duration equal to the current consumption above the threshold level of the current sensor. Finally, 3-input NOR gate synchronizes all three signals and generates a signal for the next peer.

In figure 2b, the sensor is again enabled by a simple NOR gate when it detects a valid codeword and the D flip-flop is released from the reset phase. The current sensor generates a low-state active pulse. Its rising edge flips the flip-flop's negated output to a logic zero. This signal is then synchronized with ACK_RIGHT signal from the successor by a NOR gate. Additional logic gates connected to the output pins of the latch are used to prevent undesired reset signals for the flip-flop. Latch can be formed accordingly to figure 1b, naturally, without a completion detection circuitry. The topology could be designed in

several different ways, but we believe, this is the configuration that requires the least silicon area.
For successful implementation into the real design, engineers have to be able to work with the current sensor in front-end and back-end routines. Unfortunately, there is no possibility of developing an accurate behavioral model for currently used digital simulators, since the methodology is based on an analog circuit. However, engineers will be able to use CSCD methodology in digital simulations. Synthesis tools provide the information about computation time of certain logic block. This predicted computation time can be used in testbench simulations as the pulse from current sensor generated within testbench stimuli. The rest of the CSCD is a pure logic, thus, the engineer will be able to describe such circuitry. The back-end routine is even more straightforward. For this purpose, we have developed a necessary LIB file and LEF file, which are required by place&route CAD tools. LIB file contains many important definitions, such as timing information, delay assumptions, parasitic capacitances, leakage power consumption, etc. All these information were extracted from the analog simulations during the design period for every PVT corner that design-kit contains. LEF file contains the physical dimensions of the sensor, antenna rule definitions, design rule checker information and data about every metalization level employed within the cell.
We also programed a shell script that modifies the synthesized Verilog gate-netlist and inserts the current sensor module into it. The only assumption that is required before this step is, that the engineer leaves the enabling node for the sensor floating. The input pin, which the output of the current sensor will be connected to should be left unconnected as well.

# 4 Results

The most important contribution of the presented CSCD methodology is its independence of the number of bits used by an asynchronous system. Figure 3 explains the motivation of CSCD implementation into the real circuits. Moreover, it also suggests why CSCD should undergo more intensive research in the future. Information for dependence of the conventional completion detector were taken from synthesizes VHDL source code. As we can see, the silicon area consumption for dual-rail (figure 3b) state-of-the-art completion detector increases linearly. Obviously, the dependence of the CSCD area requirements does not change with the information width. This represents great advantage. Naturally, the wider data system is working with, the greater difference in chip area demand we achieve. The point where the curves intersect, defines when the implementation of CSCD will introduce the silicon area reduction. Figure 3a shows the dependency of matched worst-case delay silicon area on the computation time. The




(a) Area consumption in single-rail pipeline (b) Area consumption in dual-rail pipeline

Figure 3: Silicon area requirements of the proposed CSCD compared to state-of-the-art

point where the lines intersect defines when it is already beneficial to implement CSCD instead of the conventionally used delay elements. Figure 4 shows the results of an analog transient simulations of the presented completion detection method in dual-rail pipeline. An 8bit ripple fulladder has been chosen for the experimental combinatorial block. Input data vectors and controlling signals were set accordingly to the handshaking protocol. The ACK_RIGHT signal from the next peer was simulated by an ideal voltage pulse source. In order to make the simulation more realistic, we also created the interconnection delay model. The values for sheet resistance of metalization layer as well as the stray capacitance were extracted from the design-kit and randomly modified (according to the corner limitations) to simulate different process variations within the die. Afterwards, we have randomly chosen the length of each interconnection from the predecessor in order to model the uneven placement of a latch from the previous peer.

In figure 4, the top waveform (solid line) represents the input data signal. This voltage was brought to selected input pins of the full-adder through the above mentioned RC elements, while the rest of the input pins were left grounded. The second waveform depicts the supply current drawn by the full-adder. One can easily observe that current peaks are strongly related to the input signal, since the signal transitions inside the full-adder occur only when the valid data appear on the input or when the reset phase is initialized. The third waveform shows ACK_RIGHT signal from the successor. Finally, the very bottom curve defines the output generated for the previous peer – ACK_LEFT. Described signal is generated correctly and precisely according to the handshake protocol rules. Hence, we can state that CSCD methodology has been successfully implemented into a dual-rail asynchronous system. Unfortunately, we could not perform post-layout simulations due to missing extraction rule files and tools in the used design-kit.

We have also performed numerous simulations with lowered power supply voltage in order to investigate the behavior, functionality and limitations of the proposed sensor. The power supply voltage of 0.8 V has been used for these simulations. This represents about 67 % of the nominal supply voltage. The results achieved by these low-power/low-voltage simulations are shown in figure 4, in the respective waveforms (dotted lines).

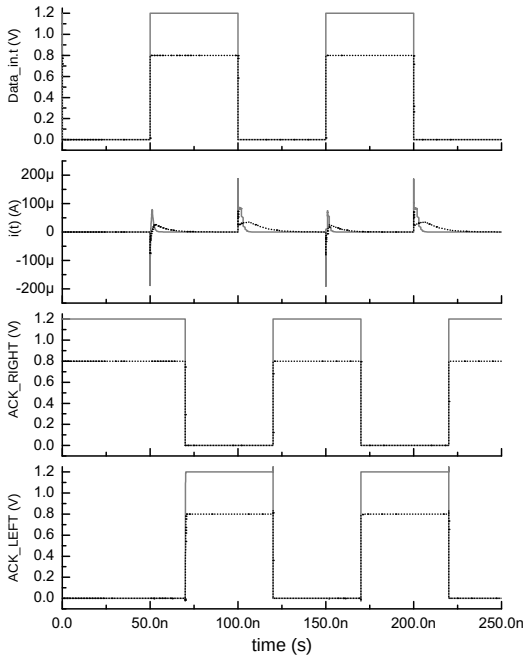



Figure 4: Handshaking signals along with input data and current transients over the time

## 5 Conclusion

We addressed an alternative approach to the state-of-the-art of completion detection in asynchronous circuits. The proposed sensor has been successfully implemented into single-rail and dual-rail pipeline circuits. Simulations results prove that the proposed sensor can be implemented in modern deep submicron technologies (90 nm was used). Furthermore, it can be employed in low-power/low-voltage applications as well. For successful down-to-silicon implementation, we have developed a shell script for automated Verilog netlist editing. We have also created necessary LIB and LEF files for straightforward and fully automated back-end routine.
Main contribution of CSCD is in lower silicon area consumption while maintaining the advantages of asynchronous circuits. The research we have carried out has defined the conditions when the usage of CSCD is beneficial (figure 3). In the near future, we intend to submit the design for fabrication and compare simulation results with the measured parameters.

## Acknowledgment

This work is resulting from the Project "Support of the Center of Excellence for Smart Technology, Systems and Services", ITMS Code 26240120005 supported by the research and development operational program funded by the ERDF, and has been done with support of the project VEGA 1/0285/09 and 1/1008/12, ENIAC-JU Project END (Agr. No. 120214) and also with R-DAS research support.

## References

[1] Izosimov, A., Shagurin, I. I., Tsylyov, V. V.: Physical Approach to CMOS Module Self-Timing, 1990, Electronic Letters

[2] Dean, M. E., Dill, D. L., Horowitz, M.: Self-Timed Logic Using Current-Sensing Completion Detection (CSCD), 1994, Journal of VLSI Signal Processing Systems

[3] H. Lampinen, O. Vainio, Dynamically biased current sensor for current-sensing completion detection, in: Circuits and Systems, 2001.

[4] H. Lampinen, P. Perala, O. Vainio, Implementation of a self-timed asynchronous parallel FIR filter using cscd, in: Norchip Conference, 2004.

[5] E. Grass, R. Morling, I. Kale, Activity-monitoring completion-detection(amcd): a new single rail approach to achieve self-timing, in: Advanced Research in Asynchronous Circuits and Systems, 1996

[6] E. Grass, S. Jones, Asynchronous circuits based on multiple localised current-sensing completion detection, in: Asynchronous Design Methodologies, 1995

[7] M. Kawokgy, C. Salama, A low-power cscd asynchronous viterbi decoder for wireless applications, in: Low Power Electronics and Design(ISLPED), 2007

[8] T. Kumaran, M. Santhi, M. Srikanth, N. Srinivasan, M. Balaji, G. Lakshminarayanan: Transient current sensing based completion detection with event separation logic for high speed asynchronous pipelines, in: TENCON 2009

[9] L. Nagy, V. Stopjakova: Completion detection in asynchronous systems by current sensing, 2010, PAD

[10] L. Nagy, V. Stopjakova: Completion detection by current-sensing in dual-rail circuits, 2011, PAD