Analyzing the Impact of Local and Global Indication on a Self-Timed System

PADMANABHAN BALASUBRAMANIAN*, NIKOS E. MASTORAKIS§
* School of Computer Science
The University of Manchester
Oxford Road, Manchester M13 9PL
UNITED KINGDOM
padmanab@cs.man.ac.uk
§ Division of Electrical Engineering and Computer Science
Military Institutions of University Education, Hellenic Naval Academy
Piraeus 18539
GREECE
mastor@hna.gr

Abstract: - The aim of this paper is to analyze the impact of local and global indication on a self-timed system architecture. Towards this end, we investigate the effect of these two indicating frameworks with respect to a 32-bit self-timed carry-ripple adder, constructed using a cascade of self-timed dual-bit adder modules. In both cases, the self-timed ripple carry adder corresponds to weak-indication. We consider the analysis from the perspective of both homogeneous and heterogeneous encoding of data, employing unordered delay-insensitive codes. The simulation results reveal that global indication benefits in terms of minimizing the power and area parameters while reporting a performance advantage over local indication.

Key-Words: - Self-timed design, Adder, Function block, Standard cells, Indication, Input/output mode

1 Introduction
‘Reliability’ is labelled as one of the five crosscutting design challenges in the Semiconductor Industry Association’s 2008 design report [1], which drives home the point that ‘robustness’ is assuming greater significance for digital logic design in ultra deep submicron technologies. In this scenario, self-timed design attracts attention – thanks to its inherent ability to tolerate supply voltage, process parameter and temperature variations [2]. With the absence of a global clock, self-timed circuits tend to exhibit better noise and EMI properties compared to their synchronous counterparts [3]. Also, they feature greater modularity permitting convenient design reuse [4], which is of interest and importance since design reuse as a percentage of all logic is expected to be 55% by 2020 [1].

Conventionally, self-timed systems implicitly support local indication. This paper deals with introducing the phenomenon of global indication and compares these two indicating models. The self-timed ripple carry adder (RCA), constructed using dual-bit adder modules, is considered as a case study and the effects of local and global indication are analyzed.

The rest of this paper is organized as follows. Preliminaries relating to self-timed logic circuits and their predominant indicating styles are described in section 2. The self-timed dual-bit adder modules, synthesized on the basis of homogeneous and heterogeneous delay-insensitive (DI) encoding of data are presented in section 3. The results corresponding to the simulation studies of 32-bit self-timed RCAs, featuring local/global indication, are given in section 4 and the inferences are drawn. The conclusions are arrived at in the final section.

2 Fundamentals of Self-Timed Logic Circuits
A self-timed logic circuit (also called as ‘function block’) must indicate (acknowledge) the arrival of all its primary inputs and the attainment of steady state on all its intermediate nodes through its primary outputs. Seitz, in his seminal work [5], classified a self-timed logic block into two robust categories as ‘strongly indicating’ or ‘weakly indicating’.

A strong-indication logic block waits for all of its inputs (valid/spacer) to arrive before it starts to
compute and produce any output (valid/spacer). On the other hand, a weak-indication logic block starts to compute and produce outputs (valid/spacer), even with a subset of the inputs (valid/spacer). However, it is imperative that at least one output (valid/spacer) should not have been produced until after all inputs (valid/spacer) have arrived. The signalling scheme for strong and weak-indication timing scenarios is depicted in figure 1.

![Fig. 1. Portraying strong and weak-indication circuit input-output behaviour](image1)

A majority of the self-timed designs adhere to a 4-phase handshaking convention for simplicity of implementation and employ a DI data-encoding scheme, with dual-rail data encoding widely preferred [6]. With dual-rail encoding, each data wire $d$ is represented using two data wires, $d^0$ and $d^1$; with the request signal embedded within the data wires. A low-to-high transition on the $d^0$ wire indicates that a zero has been transmitted, while a low-to-high transition on the $d^1$ wire indicates that a one has been transmitted. Since the request is embedded within the data wires, a transition on either $d^0$ or $d^1$ informs the receiver about the validity of the data. The condition of both $d^0$ and $d^1$ being a 0 at the same instant is referred to as the spacer (empty data). It should be noted that $d^0$ and $d^1$ are not allowed to be 1 simultaneously as it is illegal and invalid, since the coding scheme is unordered [7], i.e. no code word forms a subset of any other code word.

![Fig. 2. Dual-rail encoding and 4-phase handshaking](image2)

Referring to figure 2, the 4-phase return-to-zero handshake protocol is explained as follows:

- The dual-rail data bus is initially in the spacer state. The sender transmits the codeword (valid data). This results in 'low' to 'high' transitions on the bus wires (i.e. any one of the rails of all the dual-rail signals is assigned a logic 'high' state), which correspond to non-zero bits of the codeword.
- After the receiver receives the codeword, it drives the ackout (ackin) wire 'high' ('low').
- The sender waits for the ackin to go 'low' and then resets the data bus (i.e. it is driven to the spacer state).
- After an unbounded, but finite (positive) amount of time, the receiver drives the ackout (ackin) wire 'low' ('high'). A single transaction is now said to be complete and the system is ready to proceed with the next transaction.

3 Self-Timed Dual-Bit Adder Modules

A dual-bit adder is used to perform simultaneous addition of two bits of data, inclusive of an input carry, and can be thought of as an embedded 2-bit carry lookahead adder [12]. Let the five single-rail inputs of the dual-bit adder be named as $a_{msb}$, $a_{lsb}$, $b_{msb}$, $b_{lsb}$ and $cin$, where ($a_{msb}$, $a_{lsb}$) and ($b_{msb}$, $b_{lsb}$) represent the augend and addend inputs and $cin$, the input carry. The dual-bit adder would produce three single-rail outputs, namely $Cout$, $Sum_{msb}$ and $Sum_{lsb}$, where $Cout$ is the overflow carry and $Sum_{msb}$ and $Sum_{lsb}$ represent the most significant and least significant sum output bits respectively.

---

1 The explanation remains valid for data representation using any DI data-encoding scheme.
Herein, we shall consider dual-bit adder realizations synthesized based on ‘homogeneous’ and ‘heterogeneous’ DI data-encoding schemes. While the homogeneous data-encoding scheme implies a similar type of DI code employed for encoding all the data inputs, the heterogeneous data-encoding scheme refers to a combination of DI codes used for encoding the input data. In this work, we shall utilize dual-rail encoding for the former scheme and a combination of dual-rail and 1-of-4 encoding schemes for the latter.

3.1 Homogeneously Encoded Self-Timed Dual-Bit Adders

Let \((a_{11}, a_{10}), (a_{01}, a_{00}), (b_{11}, b_{10}), (b_{01}, b_{00}), \) and \((\text{cin}1, \text{cin}0)\) be the dual-rail encoded inputs, and \((\text{Cout}1, \text{Cout}0), (\text{Sum}11, \text{Sum}10) \) and \((\text{Sum}01, \text{Sum}00)\) be the dual-rail encoded outputs of the homogeneously encoded self-timed dual-bit adder respectively. The synthesized weakly indicating dual-bit adder \([8]\) is shown below.

In all the diagrams, the AND gate symbol with the marking ‘C’ on its periphery represents the Muller C-element, which governs the rendezvous of the input signals. The C-element outputs a 1 (0) when its inputs are 1 (0); otherwise it retains its existing state. The dual-bit adder comprises C-gates, OR gates and complex gates, as can be seen in figure 3. The adder block is weakly indicating as the job of indicating the primary inputs is distributed between its primary outputs. This adder module is suitable for placement in the conventional self-timed system topology that prescribes strong or weak-indication property for its function block.

The dual-bit adder portrayed in figure 4 is neither strongly indicating nor weakly indicating but corresponds to ‘early output logic’ in that the primary outputs may be produced based on even a subset of the inputs, thereby further relaxing the weak-indication timing constraints. However, it is to be noted that the early reset of this dual-bit adder is possible while early evaluation does not occur. It can be noticed that all the input-complete C-gates in the first logic level of the adder shown in figure 3 have been replaced by input-incomplete AND gates in figure 4, which renders it early propagative.
3.2 Heterogeneously Encoded Self-Timed Dual-Bit Adders

The dual-bit adder module based on heterogeneous data encoding can utilize the 1-of-4 codes for encoding the augend and addend inputs and the sum outputs, while the input and output carry signals can be encoded using the dual-rail code. The 1-of-4 encoded values of single-rail inputs given in Table 1 represent only one of various possible encoding formats and an arbitrary case is considered here for an illustration. As shown in Table 1, two non-redundant bits of information are represented at a time by asserting only half of the physical lines as logic ‘high’ in a 1-of-4 code in comparison with a double-rail code, though both require the same number of physical lines. As a result, the 1-of-4 encoding approach would experience only half the number of transitions compared to the dual-rail encoding approach. Consequently, the dynamic power component of the former approach is likely to be less than the latter due to reduced switching activity [9]. Nevertheless, considering the extra encoding and decoding circuitry required for 1-of-4 encoded datapaths in comparison with dual-rail datapaths the power savings gained are likely to diminish.

Table 1. Data representation based on dual-rail and 1-of-4 DI data encoding schemes

<table>
<thead>
<tr>
<th>Single-rail inputs</th>
<th>Dual-rail encoded data</th>
<th>1-of-4 encoded data</th>
</tr>
</thead>
<tbody>
<tr>
<td>a b</td>
<td>(a1 a0) (b1 b0)</td>
<td>e0 e1 e2 e3</td>
</tr>
<tr>
<td>0 0</td>
<td>(0 1) (0 1)</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>0 1</td>
<td>(0 1) (1 0)</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>1 0</td>
<td>(1 0) (0 1)</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>1 1</td>
<td>(1 0) (1 0)</td>
<td>0 0 0 1</td>
</tr>
</tbody>
</table>

Let \((a0, a1, a2, a3)\) and \((b0, b1, b2, b3)\) be the 1-of-4 encoded equivalents of the pairs of augend and addend inputs of the dual-bit adder respectively. \((\text{Sum0}, \text{Sum1}, \text{Sum2}, \text{Sum3})\) represents the 1-of-4 encoded equivalent of the pair of single-rail sum output bits. Dual-rail code is used for representing the input and output carries. The synthesized heterogeneously encoded dual-bit adder module is portrayed in figure 5.

The heterogeneously encoded dual-bit adder block shown in figure 5 corresponds to the weak-indication criteria as its primary outputs assume the collective responsibility of indicating all the primary inputs. Similar to the previous case, the input-complete gates in the first logic level can be replaced by non-input-complete gates to yield an early output dual-bit adder version that incorporates heterogeneous data encoding.

4 System Architectures, Simulation Results and Inferences

The conventional self-timed system topology is depicted in figure 6, which basically employs dual-rail encoding.

The registers consist of a bank of 2-input C-elements with a C-element dedicated for each function block input. The completion detection
(CD) logic [10] is composed of a bank of 2-input OR gates, where the inputs to each 2-input OR gate are the true and false bits of the dual-rail encoded input signal, and a C-element tree that is used to ensure synchronized data arrival from the bank of 2-input OR gates.

The basic operation of the self-timed system is explained as follows. When the system is reset, the acknowledge input to the current stage register (ackin signal) is driven to logic high. According to the handshake protocol described in section 2, valid data is fed through the current stage register into the current stage function block for requisite processing of data. After data processing, the outputs of the function block are fed to the next stage register. In the meantime, the CD logic of the current stage register asserts its ackout to high, which subsequently resets the ackin input to the preceding stage register bank, thus paving the way for flushing the current stage function block to facilitate future data processing. However, this is not achieved until the CD logic of the next stage register asserts its ackout to high, which drives the ackin signal input to the current stage register bank as low. The subsequent transactions are carried out likewise.

The system topology shown in figure 6 is said to be ‘locally indicating’, since the embedded function block is itself properly indicating. The function block that signifies the robust asynchronous combinational logic realization is of strongly or weakly indicating type. The function block can either be an individual module or can represent a cascade of smaller function blocks. It was proved in [7] that a legal interconnection of strongly or weakly indicating function blocks is permissible and gives rise to a larger strong or weak-indication function block. Sixteen stages of the weakly indicating homogeneously encoded dual-bit adder module, shown in figure 3, can be cascaded to form a 32-bit self-timed RCA that signifies the larger arithmetic function block.

The modified self-timed system architecture that is of ‘globally indicating’ type is shown in figure 7. This system architecture can comprise a linear cascade of early output homogeneously encoded dual-bit adder modules portrayed by figure 4. The main point of distinction between the ‘locally’ and ‘globally’ indicating self-timed system architectures lies in the fact that the entire responsibility of indicating all the primary inputs is confined to the primary outputs of the function block in the former case, while the responsibility is shared between the function block and the system topology (CD logic) in the latter case. This is feasible by means of the ‘synchronizer’ that is used to synchronize the arrival of the non-indicated primary inputs with a pair of primary outputs before allowing those primary outputs to be fed to the next stage register bank. Thus, this paves the way for relaxing the strict indication criteria imposed on the function block and thereby the function block could make use of more input-incomplete gates leading to possible advantages in terms of the design metrics. This is because the C-element, which is an input-complete conjunction operator, is not available as a standard element in commercial cell libraries and this necessitates its design in a full-custom or semi-custom fashion. Herein, we have adopted a semi-custom approach to realize C-gates using complex gates of the cell library that supports granularities of the range of 2 to 4 inputs.

The self-timed system configuration that could handle heterogeneously encoded inputs and outputs is shown in figure 8.

In essence, it is similar to the topology portrayed by figure 6, but with the additional provision of appropriate encoding and decoding circuitry. In cognizance of the dual-bit adder encoding mechanism discussed in section 3.2, a subset of the dual-rail inputs (augends and addends) is 1-of-4 encoded before being fed to the function block while the remaining inputs (input carry) are fed as
such. The non-dual-rail sum outputs produced by the logic block are decoded before being passed onto the next stage, while the dual-rail carry outputs are fed to the next stage as such. The encoding cost is 28 transistors per bit, while the cost of decoding is 12 transistors per bit.

In view of the replacement of C-gates in the first logic level of the heterogeneously encoded dual-bit adder module shown in figure 5, a modification is necessary to the system configuration depicted in figure 8 to permit global indication. The alternate datapath for that highlighted using dotted lines in figure 8 is shown in figure 9. All the distinct outputs of the function block need not be synchronized with the logical sum of the homogeneously encoded input signals (OR-logic block output), but only an encoded dual-bit adder sum output would suffice. The OR-logic block basically consists of a tree of OR-gates. The least significant 1-of-4 encoded dual-bit adder’s sum output is alone synchronized with the output of the OR-logic block, while the other adders sum outputs (including the intermediate and output carries) could be relaxed. Thus the overall system configuration would be ‘globally indicating’.

Before presenting the simulation results for the 32-bit self-timed RCAs, which are synthesized on the basis of different indicating models and DI data-encoding schemes, the simulation mechanism is elucidated. The delay parameter refers to the maximum propagation delay encountered in the datapath (stage-to-stage delay). The critical path delay metric was estimated using PrimeTime. To avoid the notion of a clock source, the option of a virtual clock was used that only acted as a remote reference to restrain the input and output ports of the design [13]. The area and power metrics correspond to the input registers, CD logic and function block.

The delay and power metrics consider estimated parasitics in addition to the component parameters. The area metric gives a combined account of the area of all the logic cells and has been obtained as a part of the PrimeTime framework. The total power dissipation represents the sum of dynamic and static power components, where dynamic power is the gross of switching and internal power values. Cadence NC-Sim has been used for functional simulation and also to obtain the switching activity files corresponding to the gate level simulations of Verilog circuit descriptions. Random input data were supplied to the function blocks at intervals of 15ns through a test bench, which models the environment. The switching activity files derived were subsequently used for power estimation using PrimeTime PX.

The simulations targeted a PVT corner (supply voltage of 1.32V and junction temperature of -40°C) of a 130nm bulk CMOS standard cell library. All the circuit inputs were configured to possess the driving strength of the minimum sized inverter of the cell library, while the outputs were associated with fanout-of-4 drive strength. Suitable buffering for the acknowledge input was provided in order to eliminate timing violations. Further, the combinational and the CD logic forming a part of the self-timed system were optimized for minimum latency. Since identical registers and a similar CD circuit were used for all the realizations, the area and power metrics can be deemed to reflect that of the self-timed combinational logic, thus enabling a legitimate comparison.

Table 2. Delay, area and power parameters of various homogeneously and heterogeneously encoded 32-bit self-timed RCAs

<table>
<thead>
<tr>
<th>Adder realization style</th>
<th>Delay (ns)</th>
<th>Area (μm²)</th>
<th>Power (µW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local homogeneous dual-bit adder</td>
<td>5.9</td>
<td>14921</td>
<td>871.9</td>
</tr>
<tr>
<td>Global homogeneous dual-bit adder</td>
<td>5.6</td>
<td>8833</td>
<td>648.3</td>
</tr>
<tr>
<td>Local heterogeneous dual-bit adder</td>
<td>5.8</td>
<td>10889</td>
<td>688.4</td>
</tr>
<tr>
<td>Global heterogeneous dual-bit adder</td>
<td>5.7</td>
<td>9594</td>
<td>685.5</td>
</tr>
</tbody>
</table>

Table 2 summarizes the delay, area and power metrics of the various 32-bit self-timed RCAs. It could be seen that the global self-timed system configuration betters its locally indicating counterpart in terms of delay, area and power for both homogeneous and heterogeneous DI data encoding schemes. The globally indicating homogeneously encoded 32-bit RCA reports less delay, area and power than the locally indicating version by 5%, 41% and 26% respectively, while the globally indicating heterogeneously encoded 32-
bit RCA exhibits reduced delay, area and power in comparison to the locally indicating system version by 2%, 12% and 0.4% respectively. It is also evident that the homogeneously encoded globally indicating RCA is preferable compared to the heterogeneously encoded globally indicating RCA. This is due to the overhead incurred in the design metrics as a result of the extra encoding, decoding and synchronization circuitry associated with the latter.

5 Conclusion

The impact of local and global indication on a self-timed system architecture has been analyzed in this work on the basis of a dual-bit adder functionality. Though the case study considered here is purely representative, it can be intuitively observed that global indication is preferable when compared to local indication for self-timed system realizations. This is because a globally indicating system framework introduces a greater degree of parallelism with respect to detecting proper arrival of input data into a stage whilst permitting data processing in the current stage function block. Also, the indication constraints associated with the function block implementation are allowed to be relaxed. However, a formal proof to substantiate the generalization of this hypothesis is necessary and this constitutes further work.

Nevertheless, the important observation made being that global system indication encourages a greater usage of input-incomplete gates for the function block realization, facilitating optimization of the design metrics, in comparison with local system indication that prescribes relatively more input-complete gates. It is to be noted that the global indication model can be extended to any self-timed design method and would suit any arbitrary circuit specification. The other added benefit associated with global system indication is that there exists an opportunity for faster system reset (return-to-zero) on account of the early propagative nature of the function block, i.e. early reset could happen whereas early set does not occur. With respect to the dual-bit adder functionality, this becomes feasible as the output carry logic is relaxed owing to weak-indication. This characteristic is similar to that described by means of a directed graph for the case of Martin’s adder [11]. However, our discussion here is rather more generic and addresses the entire system configuration.

References: