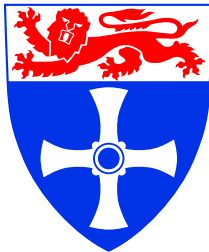

School of Electrical, Electronic & Computer Engineering

UNIVERSITY OF
NEWCASTLE UPON TYNE



Phase-encoded Transmission for NoC

C. D'Alessandro, D. Shang, A. Bystrov, A. Yakovlev, O. Maevsky

Technical Report Series

NCL-EECE-MSD-TR-2005-106

May 2005

Contact:

Crescenzo.D'Alessandro@ncl.ac.uk

Delong.Shang@ncl.ac.uk

A.Bystrov@ncl.ac.uk

Alex.Yakovlev@ncl.ac.uk

olma@yandex.ru

Supported by EPSRC grant EP/C512812/1

NCL-EECE-MSD-TR-2005-106

Copyright © 2005 University of Newcastle upon Tyne

School of Electrical, Electronic & Computer Engineering,
Merz Court,
University of Newcastle upon Tyne,
Newcastle upon Tyne, NE1 7RU, UK

<http://async.org.uk/>

Phase-encoded Transmission for NoC

C. D'Alessandro, D. Shang, A. Bystrov, A. Yakovlev, O. Maevsky

May 2005

Abstract

A novel self-timed communication protocol is based upon phase-modulation of a reference signal. The reference and the data are sent on the same transmission lines and the data can be recovered observing the sequence of events on the same lines. The sender block consists of a reference generator and variable-delay elements, while the receiver includes a delay-locked loop for synchronization and a mutual exclusion element with additional logic (validity bit and FIFO) for data recovery. This protocol exhibits high robustness with respect to transient errors caused by narrow pulse interference, usually associated with crosstalk and radiation.

1 Introduction

The issue of fast and reliable communication fabric is crucial for the successful design of systems-on-chip. An approach to design of such communication fabric is the Network-on-Chip (NoC) [1]. The synchronization of blocks is a non-trivial aspect of design, and research has delivered interfaces which are *self-timed* and *speed-independent* to address this problem. An example of self-timed interface can be found in [2], where the communication between two separate clock domains is investigated. Transient errors due to cross-talk, cross-coupling, ground bounce or environment interference become more prominent as integration increases, and this effect can be observed interconnect wires, as underlined by the work of Dupont and Nicolaidis [3, 4]. Quoting Nicolaidis: "it is predicted that single event upsets induced by alpha particles and cosmic radiation will become a cause of unacceptable error rates in future very deep submicron and nanometer technologies. This problem, concerning in the past more often parts used in space, will affect future ICs at sea level" [4]. This motivates the fault tolerance approach to design. Unfortunately, the implementation of fault tolerance leads to hardware overheads, which in its turn reduces the reliability. It may happen that large systems of the future will be spending most time recovering from transient faults. In order to alleviate this problem, a simple fault masking approach was introduced in [5]. It was applied to request-acknowledgement handshake protocols, exploiting the redundancy created by the feedback acknowledgement signal. In this paper we also exploit protocol redundancy, but feedback signals are not used. Both approaches leave a small percentage of errors unmasked. Although they significantly reduce error rate, the need in fault-tolerant design remains.

Multiple-rail encoding is based on the transmission of data on two or more lines; one or more lines go high to indicate one of the possible combinations of data. Typically, these schemes employ one state as no-data (a *spacer*, usually zero). Usual single-rail and dual-rail transmission suffers from the effects of single event upsets (SEU), as the receiver in the channel could lose data, latch wrong data or lose synchronization.

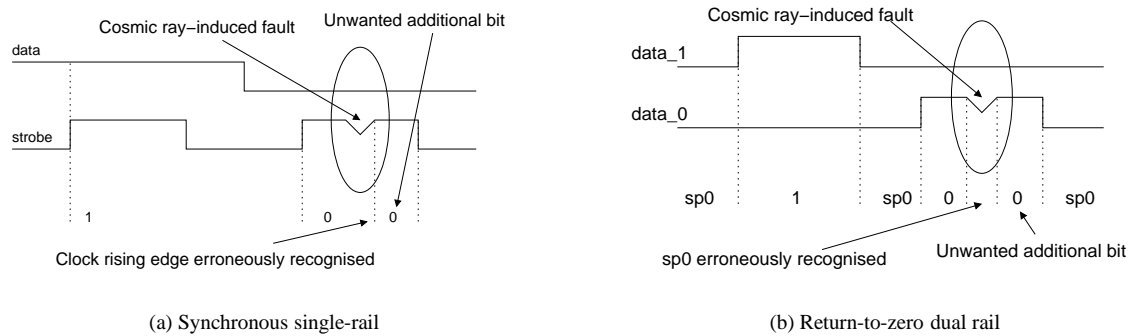


Figure 1: Protocol errors induced by pulse interference

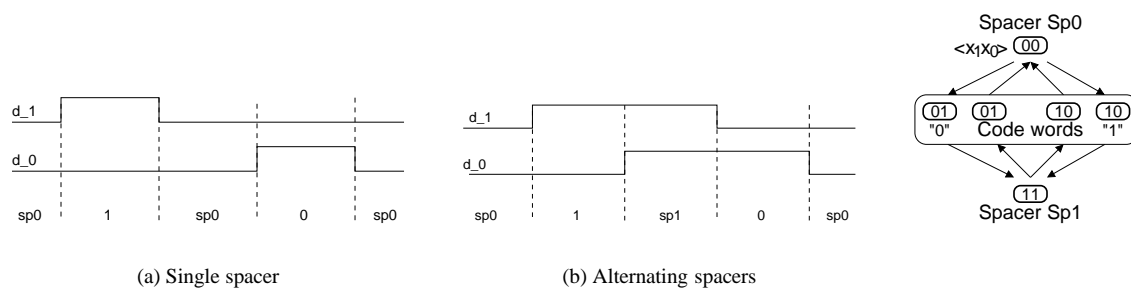


Figure 2: Dual-rail protocols

Consider, for instance, Figure 1 (a), where data is selected by a strobe. If a pulse is induced on the data line, the only way it could generate an error is if the pulse overlaps with the rising edge of a clock pulse. The occurrence of such an event has a lower probability than that of a narrow pulse upsetting the data line “far” from the strobe pulse. However, consider the case where the strobe signal is corrupted by the same narrow pulse: if the pulse is strong enough to be recognised as a valid transition, the receiver could latch additional unwanted data, possibly not only corrupting the data being sent at the time of the upset, but forcing the system to lose synchronization. In the case of dual-rail using a single spacer protocol, the upset could appear on either lines and still cause unwanted data to enter the communication channel, as in Figure 1 (b).

From the brief description given it is clear that more robust approaches are needed for these asynchronous communication channels. Dual-rail encoding using an alternating spacer protocol offers better resilience to errors; nevertheless, the general encoding is based upon the recovery of data from the value of the lines at a point in time. Our solution extends the concept of alternating spacer protocol to improve robustness on transmission lines.

Our focus is on the reliability of on-chip communication channels particularly for NoCs and we propose a methodology to improve the resistance to the special case of SEUs affecting the communication fabric.

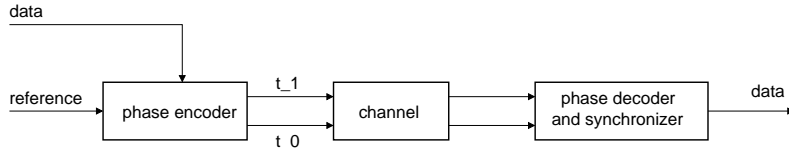


Figure 3: Block diagram of the overall system

2 Approach

2.1 Dual-rail encoding

Dual-rail codes are designed so, that data is sent across two lines rather than a single one. The data is encoded by switching high or low one of the lines; the difference in level represents an item of data. The traditional dual-rail protocol employs a single spacer, whereby after each transmission of a valid bit of data the bus returns to the zero state. In [6] the use of alternating spacers is introduced and several implications with respect to security issues (power signatures) are analysed. The paper concludes that the alternating spacer protocol (ASP) is good for security, the circuits implementing it are easy to synthesise in standard gates, and the whole approach can be integrated in the standard design flow. In this paper we use this protocol, as it allows the receiver to distinguish between SEUs and valid transitions.

2.2 PSK approach

The proposed approach builds upon the ASP. The sender and the receiver, as in Figure 3, can be synchronous systems, albeit within completely uncorrelated clock domains; alternatively they can be fully-asynchronous blocks, or combinations of the two. The reference signal is used for sampling the data by its rising and falling transitions. The data being sent modulate the phase of the clock differently on each transmission line by controlling the variable delay elements (VDE). The receiver recovers the data by comparing the signals to each other. Data values are encoded as the sign of the phase difference of the clock signal on the transmission lines. Rather than measuring the phase difference, the receiver decodes the data by observing the sequence of events on the transmission lines. The receiver records the data upon the arrival of the first transition, but the bit validity is recorded *only when the next spacer settles on the transmission lines*. Therefore, the measure of interest is the *differential delay* introduced by the VDEs on the transmission lines, which we indicate with δ .

This differential delay introduce an *event window*, where an imbalance on the lines is present. The size of this event window is determined by δ , which indicates the “nominal value” for this window, and the jitter introduced by the channel on each line γ . Provided that the system is able to reject transient faults appearing outside the event window, one such fault will generate an error only if it happens within the window. Effectively, we reduce the event window to a minimum in order to minimise the effect of transient faults, while still recognising data.

In order for the data to be correctly recovered, δ must be recognised at the receiver (but not measured); however if the channel introduces a *systematic differential delay* between the lines, transmission can be impaired. Careful design could minimize this problem, but in order to cancel out the effect of this systematic differential delay some synchronization between the two transmission lines at the receiver is needed.

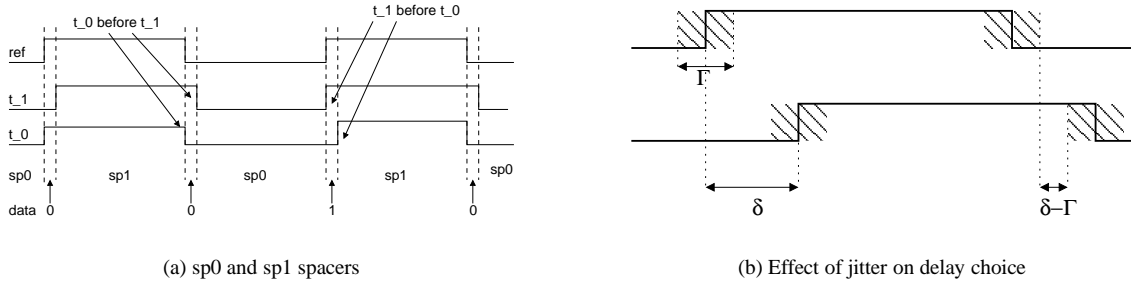


Figure 4: (a) Example of waveforms for Figure 3

Ginosar and Kol [7] describe the problem of *adaptive synchronization* and propose an adaptation protocol which employs a *training session*, where the circuit stops operating and the sender transmits dummy data to the receiver; other adaptation protocols are described. In any cases, to ensure reliability, the jitter γ introduced by the channel must be taken into account so that $\delta \gg \Gamma$ where $\Gamma = \max(|\gamma|)$ in order to guarantee deterministic behaviour of the system. However, if T is the clock period, $\Gamma \ll T$ and therefore δ can be chosen so that $T > \delta \gg \Gamma$. The value of Γ can be estimated using various techniques [8].

The fact that valid data is recorded only when the next spacer is generated (and therefore when both transmission lines have changed status) has an important property: a hazard on one of the transmission lines (generated by cross-talk, EM interference, cosmic radiation) will not be recorded as data. This is particularly important if several single-event upsets (SEU) can be generated by the environment; provided the events do not affect both lines *at the same instant*, the system will ignore the error.

An important property of the alternating-spacer protocol described in [6], from which the following definitions are reported, is the minimisation of the *exposure time* (the variation in energy consumption when processing data values). This is the time where the energy imbalance is exhibited and it is shown in the same paper that the alternating-spacer protocol has a smaller exposure time than the single-spacer protocol, in particular, the lower bound is one gate delay and the upper bound is one clock cycle. The new approach proposed in this paper, however, minimizes the exposure time of the bus, so that the bounds depend on δ and Γ and the following inequality holds: $\delta + \Gamma > \text{exposure time} > \delta - \Gamma$. Note that using a single-spacer protocol, one could minimize the exposure time by minimizing the width of the pulses representing data; however, this has the drawback of increasing spectrum occupation by the data signal.

In terms of robustness and predictability of behaviour, the system is such that a fault will become an error, if it appears through the edge of the second-arriving signal, and when a fault involves both lines at the same time. As a solution to the latter case, the designer could route the two lines so that they are physically apart if the synchronizer employed at the receiver has enough capture range. The first case has no solution; however, the probability of such an event occurring is proportional to the ratio of δ and the clock period. Therefore, the approach to avoid such errors would be to employ some high-level protocol to detect and/or correct errors.

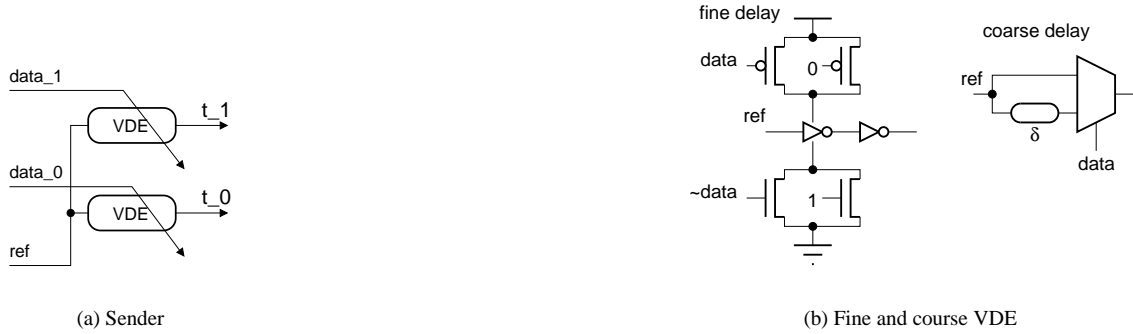


Figure 5: Block diagram of the sender and implementation examples of VDE



Figure 6: A mutual exclusion element ME

3 Operation Principles

3.1 Sender

Figure 5(a) shows a possible design of the sender having dual-rail input. The VDE implementations are shown in Figure 5(a), where the coarse VDE can be used in our sender, and the fine VDE can be used to calibrate a mesochronous system [7], which is one of application areas for our approach. For the latter, the two data lines can be set to 11 or 00.

3.2 Receiver

The receiver block has the task of recovering the data sent over the communication channel; using the approach described, this task is performed by a Phase Detector (PD). Much literature is devoted to the design and analysis of a number of different PDs. In our case, the quantity of interest is the timing relationship between two related events happening on the two transmission lines, and in particular only the sequence of events rather than their absolute distance in time. Therefore, the PD can be *binary quantized* or *lead/lag*, indicating that one event on one line leads or lags the other corresponding event on the other line.

A Mutual Exclusion element (ME), shown in Figure 6 (a) and (b) [9, 10], is essentially an S-R flip-flop followed by a *metastability filter*, which allows the two inputs to be very close to each other in terms of arrival time. If the requirement $\delta \gg \gamma$ is met, then the determinism of the system is guaranteed, and this will prevent the S-R flip-flop from entering metastability. However, as the jitter is a random variable, the

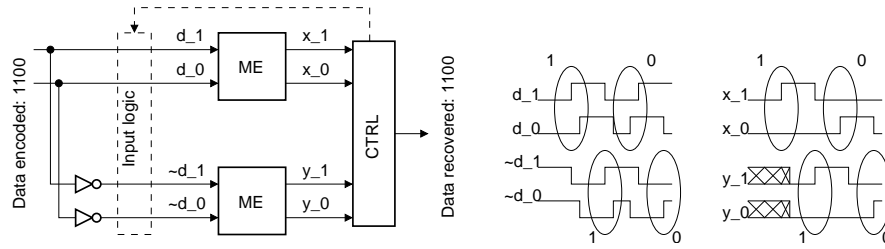


Figure 7: Double mutex receiver

time $\delta - \gamma$ could be below the expected value, presenting a metastability hazard, hence the use of an ME.

The ME will only recognize events on the rising edge of the inputs; in order to follow the protocol correctly, the falling edge of the input must also encode data. Hence the PD is built around two MEs, one of which has the inputs inverted, as shown in Figure 7. To complete the PD, additional logic is required which perform event detection and indicates the presence of a valid bit at the output. This logic is a combinational function of the input of the PD and the outputs of the MEs. It uses memory elements to ensure stable inputs to the ME and to avoid using more complex circuitry only at the outputs of the MEs.

As in any circuit decoding Phase Modulation, the receiver will require some phase alignment system, in order to make sure that the two incoming modulated clock signals are synchronized. In order to perform phase alignment, a DLL can be used and the PD can be shared between the data recovery system and the DLL, which will resemble a lead/lag-type PLL described in [11]. For this type of PLLs a class of filters called *sequential filters* is described, which has the attractive property of being governed by statistical equations and, importantly, by a set of observed values rather than a linear combination of a set of inputs. Previous work by the author [12] has illustrated a DLL based on such filters that can be employed in this case.

3.3 Repeaters, Bridges and Wire Characteristics

The system relies on the signals seen from the receiver's PD being aligned with respect to their relative phase. This can deteriorate in long wires, as the cross-talk between the two lines is dependent on the coupling capacitance of the wires, which in turns depends on their length. If however the two wires are routed differently, increasing the physical distance between them, any mismatch in length and in number of vias would result in difference of resistance of the wire; the overall capacitance between the wire and the ground planes can also be affected if the wires "hop" across layers. These two effects result in mismatch of the time constant of the wires and therefore the delay relationship will be corrupted.

As a countermeasure two options, not mutually exclusive, can be considered: clock-distribution techniques (and in general, layout techniques) and the use of *repeaters* or *bridges*. We define a *repeater* a device which simply regenerates the phase relationship between the two signals across the wires; a *bridge* is a device which receives the data and re-sends it across. Both can be implemented by putting back-to-back a receiver and a sender. A receiver, however, can be a more light-weight device, with low-latency and of simple implementation. The bridges, instead, could be used to perform additional functions: if used in NoCs, these devices can have several outputs (a combination of a receiver and several senders), thus forming a network switch. A disadvantage of bridges/switches is their latency. The latency is defined by

the data validation process, which finishes only after the second wire have switched.

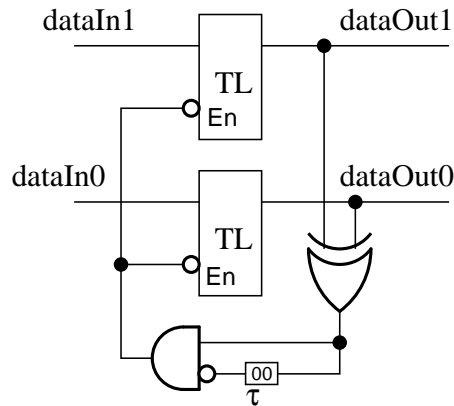


Figure 8: Repeater design

A low-latency solution is a repeater shown in Figure 8. It does not possess error masking properties, as it does not wait for the validation. The design is based upon two transparent latches, which become disabled as soon as the code word arrives. After this, they store the state for the time interval defined by the delay element. At the end of the interval they start conducting the input data to the output again. If a narrow pulse interference arrives to the input during the spacer state, it propagates to the output and gets “expanded” to the same time interval. Such an error will be corrected later at the receiver. The limitation of this circuit is the minimum switching separation interval, which should exceed the latency of two gates: the EXOR and the AND gates.

The other approach consists in actively controlling the layout of the design in order to match the characteristics of the two wires. This approach can be time-consuming, as the final routing of the wires would not be known until the layout stage; if any errors occur, the layout might have to be redone, increasing the design time unnecessarily. Therefore, the designer could make assumptions about the final physical implementation of the wires at design time using worst-case scenarios based on the process parameters for the particular process which will be employed. In this case, some heuristics can be employed and the process could be automated. Note that a mismatch can be tolerated if it is within the DLL range; to increase the range, and hence the tolerance on wire mismatches, one would then need to change δ to accommodate for the requirements.

A better solution would attempt to optimise the performance of the link by segmenting the wires and add repeaters along the line *taking into account the layout consideration*. In practice, this means to calculate the maximum allowable length of the wires to preserve phase relationship within an acceptable range and include at that point the repeater. In this case, some assumptions and calculations are required, as well as, possibly, the inclusion of layout techniques to improve reliability; however, the inclusion of repeaters will reduce the uncertainty introduced by long wires, hence simplifying the design process.

Layout techniques aiming to minimise clock skew are being investigated, as well as systems to automate the design of the wires by identifying the optimal wire length between repeaters.

4 Error Checking, Detection and Correction

An interesting aspect of this system is that error checking is simplified thanks to the inherent nature of the transmission. Error detection/correction can be achieved by the next layer up in a layered network hierarchy. The subject of error detection/correction is extensively discussed in literature, and has been for a long time with respect to networks.

The error checking mechanism can be split into two types: stuck-at faults detection and dynamic errors detection (such as errors which cause a change in the dynamic properties of the device). The stuck-at faults can be detected by observing the pattern of changes in the spacer. As the spacer alternates continuously between two possible spacers, the device can detect a stuck-at fault if the spacer alternation is not observed. Note that in that case, no data is received (as possibly one of the lines has not switched). In the case of two-wire communication, this could cause a problem as a fault can be seen as a failed data transmission. The receiver could then throw an exception requesting the sender to repeat transmission, but this, in turn, could cause the receiver to expect data when the sender has none. These issues can be resolved at design time (a time-out mechanism can be included, for instance); however, a common trait of any solution would be the inclusion of an accumulator which would “log” all the occurrences of a spacer exception and be reset at the occurrence of a correct sequence of spacers. If the sum of all exceptions exceeds a given value, the device would recognise the presence of a stuck-at fault.

The detection of “dynamic” errors can be performed employing the DLL described above. The DLL would correct any mismatch in time delay introduced by the lines if this is within the DLL’s capture range. However, if this exceeds the DLL range, the communication mechanism could be designed so that any such errors is immediately recognised. Consider, for instance, the case where one of the dual-rail line, say the one whose precedence indicates a “0”, is significantly slower than the other. In this situation, the amount of delay introduced onto the “1” line could not be sufficient to invert the edges and the systematic delay offset could be beyond the DLL range. In this case, the receiver will always receive “1”s regardless of the sender’s intention. The sender could then be designed to append a bit which is always different from the last transmitted bit: if this last bit of a transmission is equal to the penultimate, then the transmission is corrupted.

Similar approaches can be employed in the case of multiple-rail systems (see Section 6).

5 Implementation Example

5.1 Description

An implementation example has been designed and simulated as proof-of-concept. The sender is an asynchronous block assembled as shown in Figure 5(a), the coarse VDEs are built as in Figure 5(b); the delay of 500ps was chosen.

The receiver design is illustrated in Figure 9. The two ME elements and the input logic (I1-I4) implement the PD in the receiver. It also includes control logic for identifying code-words.

The receiver observes the dual-rail pair of signals (x_1 and x_0), and then based on the order of their switching, records the input code-word. In order to sample 1-to-0 switching, the inverted signals x_{1_bar} and x_{0_bar} are introduced. Under normal operation, if x_1 wins, then x_{1g} is generated, alternatively if x_0 wins, then x_{0g} signal is generated. The other ME component operates similarly for sampling the falling

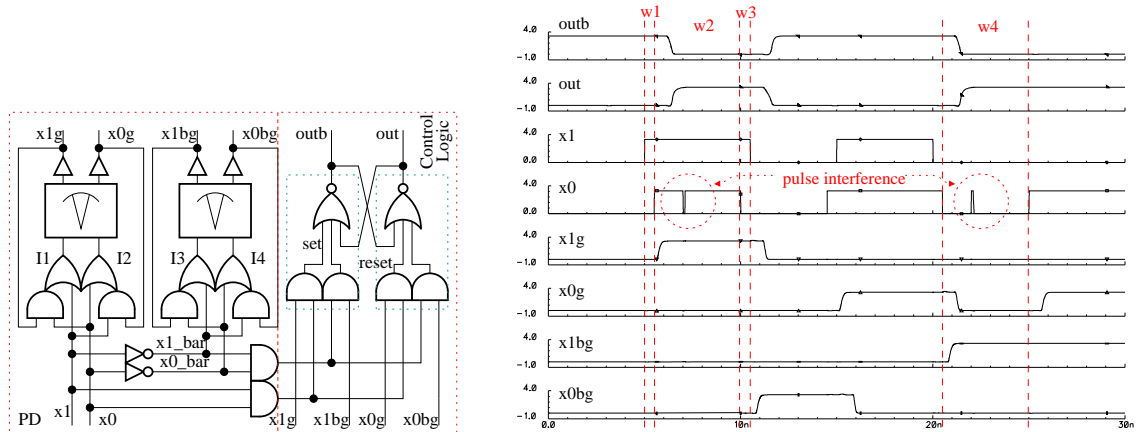


Figure 9: Implemented receiver and its waveforms

edges. The corresponding grant signals are $x1bg$ and $x0bg$.

In order to prevent pulse interference during spacer states, the input logic (gates I1-I4) is used. For example, assuming $x1$ wins the grant, during the spacer state, if the $x1$ is changed due to interference, then gate I1 will hold the grant signal until next spacer state coming ($x0$ and $x1g$ are all high).

The control logic converts dual-rail code-words to normal data. Two AN221 gates implement this mechanism. As the alternative spacer protocol is used, after a grant signal is generated, the expected spacer is known. For example, after $x1g$ is generated, the expected spacer is all-ones. When the logic receives the all-one spacer, a set signal is generated and the output ($out=1$) is produced.

The system was implemented using the AMS 0.35 μ CMOS technology (TECH-CSI) under the Cadence toolkit and simulated using the Cadence analogue simulation tool.

5.2 Results

Figure 9 shows some waveforms obtained during simulation of the system described. The value of δ was chosen to be 5% of the clock period T , which in turn was 10 ns. The maximum value of the jitter could therefore reach 500ps in the limit.

We also made an experiment using an unrealistic estimation of jitter to less than 3ps on each transmission line in order to illustrate the behaviour at “extreme” conditions. The value of δ was 15ps, five times bigger than the delay. Even at these short times, the PD does not enter metastability and exhibits correct behaviour. In Figure 9 the windows $w1$ and $w3$ indicate the distance δ while $w2$ and $w4$ are the windows where interferences are ignored. Note in fact the rejection of two faults happening at times 7ps ($w2$) and 22ps ($w3$). In the first case, the fault appear during a transmission clock cycle, but the erroneous data is not latched; in the second case, the fault appears between two clock cycles, but it still does not affect the behaviour of the transmission line. Please note that two transitions on “out” happen close to the faults, but are completely unrelated to them: in fact, they are the result of the decoding of the information appearing at times (respectively) 5ns and 20ns (the transitions on “out” start before the faulty transitions). The time between the reception of a valid data and the production of that data at the output of the receiver block is 1.68 ns. This time results from using ME, which are relatively “slow” devices. If δ was chosen to be

larger, different techniques could have been used for the design of receiver to recover data, leading to faster response at the expense of a wider event window.

The receiver is not Speed Independent and works under the Fundamental Mode without completion detection logic. Therefore, some timing assumptions are used in the design, for example the inverters used between the two MEs must have identical dynamic behaviour. Apart from this simple one, some others hold:

1. The cycle time of the sender must be greater than 1.68 ns (time to generate an output at the PD);
2. $\tau_{(x1,x0)=00/11} \geq \tau_{inv} + \tau_{and2} + 2\tau_{an221}$ after the relevant grant is generated;
3. $\delta < \tau_{grantGenerate}$, where $\tau_{grantGenerate}$ is the time between an edge arriving and the generation of the corresponding grant signal.

The first assumption is expressed in 3.2. the second is necessary to guarantee that the length of the valid spacer is long enough for the signal to propagate through the flip-flop at the right-hand side of Figure 9 and the feedback to reach the flip-flop to keep the value.

6 Multiple-rail encoding

The use of multiple-rail encoding is an attractive extension to the system. The work of John Bainbridge, summarised in his thesis [13], describes the use of a single-rail bus to avoid the overhead imposed by the use of multiple-rail bus implementations; however, he also proposes the use of 1-of-4 encoding [14] as a possible improvement in terms of power efficiency. The idea in this case consists in encoding a symbol in the sequence of arrival of a reference signal over n wires. All the issues presented in the previous sections hold, but other significant properties appear, which are worth investigating. We report a simple analysis as an example.

Assume that the delays introduced onto the n wires are all different, that is, no two wires present the same delay. Then, the order of arrival in the ideal case (no noise on the transmission lines) will correspond to one of $n!$ combinations. These states can be therefore be transmitted as a single symbol on the wires. This property can be exploited to transmit, say, control and data information in one symbol, or to multiplex several channel onto a single channel.

Note that in this case the power consumption per symbol of the proposed encoding scheme will be greater than in the case of 1-of- n encoding as more lines will switch per symbol; however, the availability of $n!$ combination per symbol reduces the overall switching activity.

Table 1 shows a summary of some encoding and a comparison with the phase encoding scheme illustrated here. Note that, even if the number of transitions per symbol appear to favour 1-of- m schemes, the overall number of transitions for an example 128 bit packet shows significantly different behaviour. This, coupled with the availability of extra states which could be used to encode control signals, indicates an attractive feature of the scheme. In general, for an n -of- m encoding, the number of states is $\binom{n}{m}$ and the transitions per symbol in the case of a NRZ protocol will be n , while for RTZ protocol it will be twice as many. In the case of the phase encoded scheme, the number of transitions per symbol is always the number of rails. The total number of transitions per packet will be:

$$\text{transitions per packet} = [(\text{packet length})/(\text{bit per symbol})](\text{transitions per symbol})$$

Type of Link	Available states	Bits per symbol	Extra states	Transitions per symbol	Transitions per packet
4-rail Phase-encoded	24	4	8	4	128
1-of-4 RTZ	4	2	0	2	128
1-of-4 NRZ	4	2	0	1	64
6-rail Phase-encoded	720	9	208	6	90
1-of-6 RTZ	6	2	2	2	128
1-of-6 NRZ	6	2	2	1	64

Table 1: Example of comparison between different encoding and relative power consumption in terms of transitions over the link. Packet length = 128 bit. RTZ = Return To Zero. NRZ = Non Return to Zero.

Given a packet of p bits, for the phase encoding scheme the equation will be:

$$\text{transitions per packet} = \frac{p}{\lfloor \log_2(n!) \rfloor} n, \text{ where } n \text{ is the number of wires}$$

For the n -of- m encoding (RTZ):

$$\text{transitions per packet} = \frac{p}{\lfloor \log_2 \binom{n}{m} \rfloor} 2n, \text{ where } m \text{ is the number of wires and } n \text{ is the number of}$$

wires switching per symbol. Table 2 illustrates some results for a 128-long packet. The n -of- m results refer to RTZ protocol, more widely used.

	Number of wires (m)								
	2	3	4	5	6	7	8	9	10
Phase Encoding	256	192	128	110	90	77	72	72	70
1-of-m	256	256	128	128	128	128	86	86	86
2-of-m	–	512	256	172	172	128	128	104	104
3-of-m	–	–	384	258	192	156	156	132	132
4-of-m	–	–	–	512	344	208	176	176	152
5-of-m	–	–	–	–	640	320	260	220	190
6-of-m	–	–	–	–	–	768	384	264	228
7-of-m	–	–	–	–	–	–	602	364	308
8-of-m	–	–	–	–	–	–	–	688	416
9-of-m	–	–	–	–	–	–	–	–	774

Table 2: Number of transitions per packet (packet length = 128 bit) for different encodings.

The table shows that for large numbers of wires (>5) the number of transitions over the links employed by the phase encoding technique is much smaller than that employed for 1-of- m encoding (which is the smallest of the n -of- m techniques shown). Note that this only applies to RTZ protocol, as the NRZ protocol reduces the switching by half at the expense of unpredictable spacers. The number of transitions can be used as a means to evaluate the power consumption of the link: therefore, we show that the phase encoding technique can deliver high performances at relatively low power, less than 1-of- m techniques, thanks to the higher number of states per symbol. Also, the system can transmit data and control information together, further improving the throughput. However, it is important to remember that a large number of wires imposes high complexity of the sender and receiver, especially the latter. If MEs are employed, banks of MEs will be needed to receive and decode the symbols and additional logic will also be necessary.

In the case of multiple-rail transmission, the cross-talk effect become more prominent and can strongly

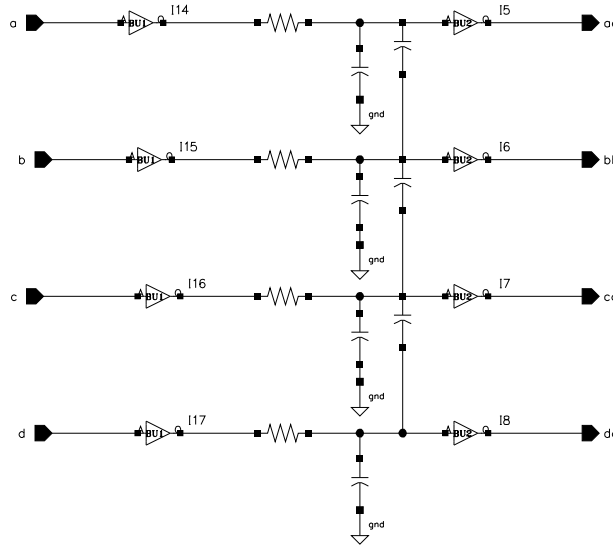


Figure 10: First-order model of four wires for 4-rail link

affect the integrity of the data being sent. Figure 10 shows a model of a 4-wire communication channel where only the RC non-idealities of the wires are taken into account. The capacitors represent the coupling to the bottom ground and the coupling across the wires involved in the communication. This model, although simplistic with respect to the real case of a physical channel, is useful to visualize the effect of capacitance and resistance across the wires. Figures 11 a) to f) illustrate these effects. We only show the result for a 4-wire channel as the results for the 2-wire channel case can be considered a subset of the results for the 4-wire case.

The images show the time relationship at the receiver between the edges. The time of arrival with respect to the first arriving edge was recorded; the relative times were sorted and plotted. The shape of the surface at the sender would be a flat surface at 45 degrees from the x - y plane: this is because the edges are sent at equal time distance from each other. First note that the corruption of phase difference across the channel is proportional to the length of the wires. This is due to the proportionality between coupling to ground and area of the wire (itself proportional to the length) and between the coupling between wires and the lengths of the wires. The figures refer to a $.35\mu\text{m}$ process and the wires are thick metal sitting on the top layer. We assume no layer change and therefore no vias across the whole length of the wires. Also note that, perhaps intuitively, the corruption of phase relationship depends on the combination. Note that the reduction in time distance is greater when two adjacent wires switch. To illustrate this phenomenon, consider Table 3, which contains the results of the 2 mm case shown in Figure 11 f).

Because the time distance between events at the sender was of 1 ns the time distance at the receiver should have been retained. Instead, this distance is greatly corrupted in several cases: although the order of arrival is preserved, the time distance is reduced in some cases by more than 80%. These corruptions always occur when the adjacent wire is being switched. Note that the total time between the first event and the last should be 3 ns, while the average is 2.5 ns.

The reason for the strong relationship between wire number, cross-talk and phase corruption is due to the intrinsic nature of the system: as all wires switch at time differences between each other approaching zero (if we assume that the time difference δ respects the inequality $T > \delta$ where T represents the minimum

Wire a	Wire b	Wire c	Wire d	Order of arrival
0.58	1.27	0.69	–	dcba
1.20	0.52	0.80	–	dcab
1.12	1.32	0.15	–	dbca
1.06	0.67	0.92	–	dacb
0.20	1.34	1.07	–	dbac
1.04	0.80	0.89	–	dabc
0.57	1.48	–	0.19	cdba
1.56	0.46	–	0.19	cdab
1.41	–	0.47	0.45	bdca
–	1.00	0.73	1.10	adcb
0.72	–	0.88	0.47	bdac
–	0.96	0.56	1.11	adbc
1.22	0.51	–	0.65	cbda
0.45	0.96	–	0.88	cadb
1.38	–	0.51	0.60	bcda
–	1.20	1.38	0.25	acdb
0.19	–	0.37	1.57	badc
–	0.85	0.50	1.17	abdc
0.61	0.51	–	1.38	cbad
0.44	0.56	–	1.48	cabd
0.65	–	0.51	1.22	bcad
–	0.19	1.37	1.18	acbd
0.19	–	1.48	0.57	bacd
–	0.82	1.19	0.64	abcd

Table 3: Results for 2 mm wires. the numbers indicate the time distance between an event on the relative wire and the previous event, as shown in the corresponding cell in the column “Order of arrival”. The times are in nanoseconds and the time difference at the sender was of 1 ns. As an example, consider row 1, where the order of arrival is “dcba”. If wire d switches at time 0, wire c will switch at time 0.69 ns, wire b at 1.27+0.69 because 1.27 is the time between events on b and c, wire a at 0.58+1.27+0.69 ns for the same reason

distance between two events happening on the reference signal) they can be considered “allies” for the purposes of cross-talk. Therefore the switching of one line will affect the switching of the adjacent line in that the latter will perform the switching faster than if it were to switch alone. For large numbers of wires, this could become a significant problem, as the cross-talk between non-adjacent lines will decrease strongly; experiments have shown that for 4 wires the effects of cross-talk, although causing corruption of phase, are not such to cause errors in data received. For large numbers of wires (experiments have shown that “large” might mean >6) the switching of a “cluster” of wires after an “isolated” wire switching can result in errors, as the sequence is corrupted. To understand this effect, consider a group of 6 wires (a-f) and a combination of switching d-a-e-f-b-c. In this case, if δ was not chosen properly, the fact that d has switched helps e to switch faster. However, a, which at the sender is before e, has no other wires to “help”. Therefore, at the receiver, e could arrive slightly before a, or cause metastability at the receiver which could resolve with a or e arriving first arbitrarily, resulting in a received combination d-e-a-f-b-c, different from what was intended.

How to choose the correct value of δ is under investigation. Another major factor which would help avoiding this effect is the physical distance between the transmission wires, which can be imposed in the

router at layout.

7 Future Work

NoCs are the obvious application domain to investigate the full potential of this novel approach. For instance, the need to insert buffers along the transmission lines to regenerate the phase relationship goes hand in hand with the presence of buffers along the transmission medium of a NoC. The capabilities of the system to support multiplexing is an additional useful property, together with the simplicity of error-checking, which can be performed at various level.

Some issues are under investigation and models and simulations are being developed, in particular in the areas of jitter estimation and selection of the optimal value of δ for reliable data recovery. The problem of jitter estimation is important: as the value of δ depends on the period of the reference signal (or in any case on the minimum time between two subsequent transitions on the reference line) and on the jitter, calculation of the jitter would lead to the identification of the minimum value for δ . However, jitter estimation is not a trivial task, as shown in relative literature. Several options are under scrutiny at the moment, with some promising result already being produced.

Another subject of investigation is the optimisation of wire length and buffer insertion. As described in Section 3.3, an algorithm, possibly to be automated, which would lead to the correct identification of optimal wire length and the inclusion of repeaters along the line can be the task of the designer.

As described in Section 3.3, the addition of control logic on a bridge would allow packet routing across the network employing this novel technique. This logic would have to read the packets, decode them and forward them appropriately.

Finally, work is under way to analyse various encoding schemes and different values for n in the case of multiple-rail encoding. Different encodings can improve throughput of the link and reduce the error rate. At the same time, increasing the value of n increases dramatically the complexity of the receiver in terms of area consumption and decoding time, and also introduces an additional overhead in terms of synchronisation of all the n lines.

8 Conclusions

A novel interconnection approach for SoCs has been presented together with some examples of implementation. The results show high robustness to transient faults of the type described (narrow-pulses) and relative simplicity of implementation. An important feature of the system described is the adaptability to a variety of environments (GALS, NoCs), achieved without the need for sophisticated circuitry. In fact, the system can almost be “plugged in” and work, as long as the synchronization protocol and the buffer stages are designed correctly.

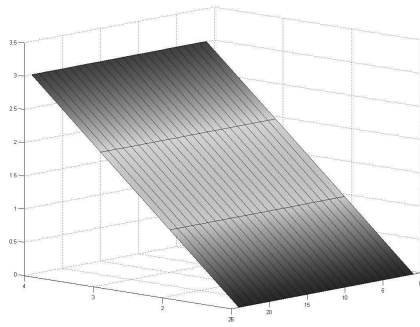
The simulation results show that the circuit works as expected and has the ability to filter out interference. More accurate evaluation of jitter and identification of minimal event windows (possibly on-line) is under consideration. Together with the jitter introduced on the transmission lines, additional sources of jitter are in fact the delay elements themselves, particularly if a delay line is employed. A more analytical description of the design requirements is therefore being carried out, together with a more accurate definition of the effects of a fault appearing through the window w_1 .

Future work aims to implement more complex protocols employing a larger number of transmission lines in order to increase throughput of the channel and increase reliability, automate the design process, identify the optimum wire length and repeaters number, and finally to devise reliable on-chip jitter estimation techniques.

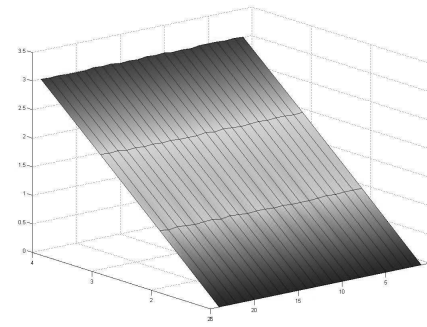
References

- [1] W. J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *Proceeding. Design Automation Conference. DAC 2001*, pages 684–689, June 2001.
- [2] Ajanta Chakraborty and Mark R. Greenstreet. Efficient self-timed interfaces for crossing clock domains. In *Proceedings. 9th International Symposium on Asynchronous Circuits and Systems*, May 2003.
- [3] Michael Nicolaidis Eric Dupont and Peter Rohr. Embedded robustness IPs. In *Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition (DATE'02)*, 2002.
- [4] Michael Nicolaidis. Time redundancy based soft-error tolerance to rescue nanometer technologies. In *17th IEEE VLSI Test Symposium*, April 1999.
- [5] A. Yakovlev. Structural technique for fault-masking in asynchronous interfaces. *IEE Proceedings E (Computers and Digital Techniques)*, 140:81–91, March 1993.
- [6] Danil Sokolov et al. Design and analysis of dual-rail circuits for security applications. *IEEE Transactions on Computers*, 54(4):449–460, April 2005.
- [7] Ran Ginosar and Rakafet Kol. Adaptive synchronization. In *Proceedings. AINT'2000*, pages 93–101, July 2000.
- [8] Marcel A. Kossel and Martin L. Schmatz. Jitter measurements of high-speed serial links. *IEEE Design and Test of Computers*, 21:536–543, November-December 2004.
- [9] C. Molnar and I. Jones. Simple circuits that work for complicated reasons. In *Proceedings. Sixth International Symposium on Asynchronous Circuits and Systems*, volume 1, pages 138–149. IEEE CS, April 2000.
- [10] Jordi Cortadella et al. Designing asynchronous circuits from behavioral specifications with internal conflicts. In *Proceedings. ASYNC'94*, pages 106–115. IEEE CS Press, November 1994.
- [11] James R. Cessna and Donald M. Levy. Phase noise and transient times for a binary quantized digital phase-locked loop in white gaussian noise. *IEEE Transaction on Communications*, COM-20(2):94–104, April 1972.
- [12] C. D'Alessandro et al. On-chip sub-picosecond phase alignment. Technical report, University of NEwcastle upon Tyne, 2005.
- [13] William John Bainbridge. *Asynchronous System-on-Chip Interconnect*. PhD thesis, University of Manchester, UK, March 2000.

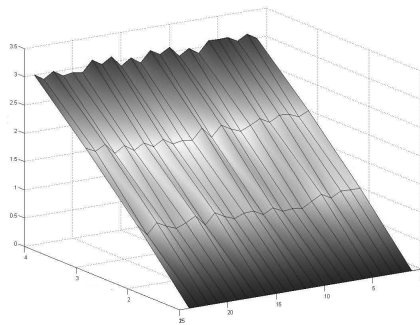
- [14] John Bainbridge and Steve Furber. Delay insensitive system-on-chip interconnect using 1-of-4 data encoding. In *Proceedings. Seventh International Symposium on Asynchronous Circuits and Systems. ASYNC 2001*, pages 118–126, 2001.



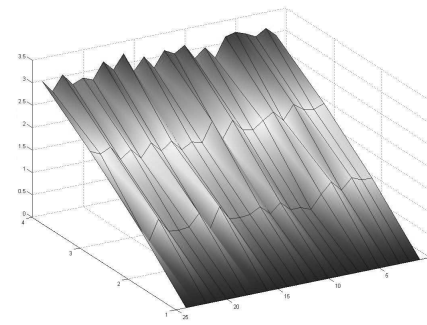
(a)



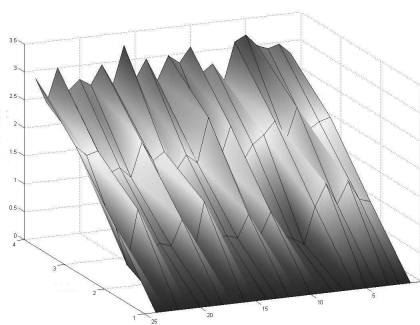
(b)



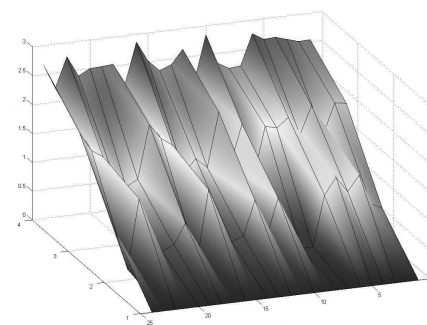
(c)



(d)



(e)



(f)

Figure 11: Results for 4-rail implementation example. The x -axis (horizontal) is the combination number of arrival ($4!=24$ possible combinations), the y -axis (front to back) represents the order of arrival and the z -axis (vertical) the time of arrival of the edge with respect to the first edge received. For correct phase relationship, the surface should be flat and at an angle of 45 degrees from the x - y plane, as every edge is sent at the same time distance from the previous. From a) to f) the wire length is increased: $2 \mu\text{m}$, $20 \mu\text{m}$, $200 \mu\text{m}$, $500 \mu\text{m}$, 1 mm , 2 mm