

# Asynchronous Transient Resilient Links for NoC

Simon Ogg  
University of Southampton  
Highfield  
Southampton  
+44-23-8059-3119  
so04r@ecs.soton.ac.uk

Bashir Al-Hashimi  
University of Southampton  
Highfield  
Southampton  
+44-23-8059-3249  
bmah@ecs.soton.ac.uk

Alex Yakovlev  
Newcastle University  
Merz Court  
Newcastle upon Tyne  
+44-191-222-8184  
alex.yakovlev@ncl.ac.uk

## ABSTRACT

This paper proposes a new link for asynchronous NoC communications that is resilient to transient faults on the wires of the link without impact on the data transfer capability. Resilience to transients is achieved by exploiting the phase relationship between data symbols and a common reference symbol where the symbols are transmitted using additional wires. Detection of transient faults is performed by comparison of the data symbol and the reference symbol. We demonstrate it is possible to achieve a similar number of transitions per bit as existing delay insensitive codes, from a power consumption point of view, but achieving resilience to transient faults. The link has been synthesized and validated using 0.12  $\mu\text{m}$  technology and power, area and performance are given. It has been shown that the link area cost is 409  $\mu\text{m}^2$  per data bit and energy per bit is 356 fJ/bit. Latency through the link is 0.8 ns and the maximum operating frequency or throughput of the link is 1.056 GHz.

## Categories and Subject Descriptors

B.4.3 [Interconnections]: Asynchronous/Synchronous Operation  
B.8.1 [Performance and Reliability]: Reliability, Testing, and Fault-Tolerance

## General Terms

Design, Reliability.

## Keywords

Asynchronous, Network-on-Chip, Point to Point Link, Transient Faults, Reliability.

## 1. INTRODUCTION

As technology scales down more IP cores are being integrated onto a single chip. Significant effort into the communication between the cores has resulted in extensive research of using Network-on-Chip (NoC) as the communication mechanism [1-3]. The NoC consists of switches and network interfaces connected together by links. Asynchronous methods of communication are finding their way into NoCs due to problems of power and clock distribution associated with synchronous circuits [4].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CODES+ISSS'08, October 19–24, 2008, Atlanta, Georgia, USA.  
Copyright 2008 ACM 978-1-60558-470-6/08/10...\$5.00.

The asynchronous link can be broadly categorized into several styles such as bundled data, quasi-delay insensitive (QDI) and delay insensitive (DI). Bundled data relies on some relative timing to be kept between the data and a reference signal. DI, or self-timed, uses data encoding so that the receiver knows when it receives valid data. There are numerous delay insensitive encodings such as Dual-Rail, 1 of 4, level-encoded dual rail (LEDR), level-encoded transition signaling (LETS) and multiple rail phase-encoding [5-9].

As circuits shrink and integration increases errors will become more prominent [10]. Errors can fall into two broad categories, permanent and transient. Permanent errors are caused by the manufacturing process. Transient errors can be caused by cross-talk, coupling or noise and particles. Up to 80% of errors can be caused by transient faults [11]. Dual rail, 1 or 4, LEDR and LETS offer little or no resilience to transient errors which could cause invalid data to be accepted at the receiver end of the link. Multiple rail phase-encoding improves on these by offering an inherent resilience to transient errors during the idle times when data is not being transmitted but at the expense of complex receivers and transmitters as the number of wires increase. Resilience to transients or soft-errors in NoC has been demonstrated in [12, 13] but these schemes use detection and correction at the router level. A link level detection scheme using Hamming codes and interleaving has been shown in [11] at the expense of including de-interleaving and Hamming distance decoding circuitry. Recently [14] has demonstrated a self correcting green joint coding scheme to tolerate transients errors and reduce crosstalk through bus encoding and triplication error correction coding. Single event upset hardened pipeline interconnect has been presented in [15] but is proposed for synchronous operation.

This paper proposes the introduction of additional wires in order to use a bit symbol to represent the data bits. Using two wires per bit allows the data symbol to have four phases. Transient resilience is achieved by exploiting the phase relationship between the data symbols and a common reference symbol. The paper is organized as follows, the motivation for this work is shown in section 2 and 3 giving examples of current asynchronous links and our proposed resilient link respectively. Section 4 describes the proposed asynchronous transient resilient link and the circuits in detail. Section 5 gives the experimental results and finally section 6 concludes the paper.

## 2. CURRENT ASYNCHRONOUS LINKS

In this section we consider existing asynchronous links and show limitations. An introduction to architectural considerations for asynchronous links is given in [16]. A single ended asynchronous link, such as bundled data uses a single reference signal to show

when the data is valid, Figure 1(a). It can be considered to have some resilience on the DATA wires as the receiver effectively ignores the data until the VALID signal is set. As technology scales down the issue of relative timing between the DATA signals and the VALID signal could diverge due to the tolerances of the wiring and the transistors in the gates. Delay insensitive codes are one of the ways to remove the dependency of timing between data and bundled reference signals since the de-coders do not care when the signals arrive [8].

Dual rail coding (Figure 1(b)) was introduced to provide a delay insensitive solution to asynchronous data transfer. This introduced extra wires, 2 per data bit, which allows a '0' or '1' to be transmitted by asserting one wire or the other. In this case asserting DATAA[x] transmits a 1 and asserting DATAB[x] transmits a 0. Figure 1(c) shows level encoded dual rail (LEDR), using the same number of wires as standard dual rail it uses less transitions per data bit. This is achieved by toggling the same wire if the data to be transmitted is the same as the previous or toggling the other wire if the data is different to the previous. Figure 1(d) shows 1 of 4 encoding, where 4 wires are used to transmit 2 data bits. A single wire is asserted and then de-asserted to represent 2 bits of data, for example asserting wire A for the receiver to obtain data '00'. Figure 1(e) shows an example of 4 wire multiple rail phase encoding. The information is contained in the arrival order of the edges rather than the logic level of the signals, care has to be taken to ensure that the arrival order of the signals remains the same as they propagate along the link.

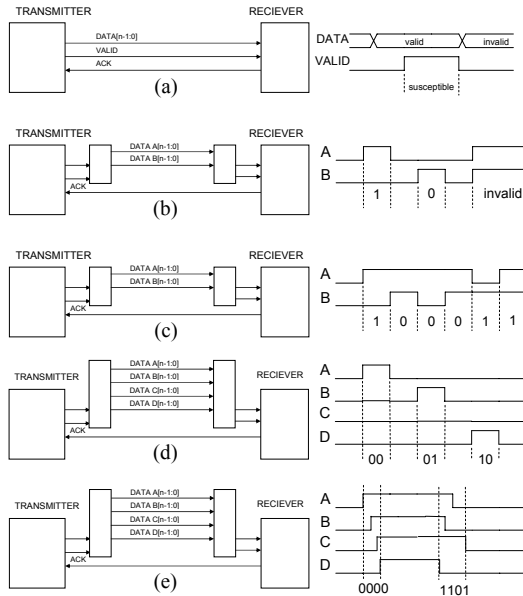


Figure 1 Current asynchronous links

More recently [14] proposed a self correcting green joint coding scheme for NoC interconnect where the data is first encoded to minimise crosstalk and then a triplication error correction code applied to allow error correction. The joint coding scheme increases the number of required wires quite dramatically, for the crosstalk minimisation the increase is 1.25x and for the triplication it is 3x, meaning that the number of wires increases by 3.75x, this however can be mitigated by serialization of the data before the triplication stage at the expense of higher link frequencies.

As discussed, a single ended asynchronous link such as bundled data does offer some resilience to transients during the time

period when VALID is not asserted, but as technology scales down the issue of keeping relative timing between the bundled VALID reference signal and data may become an issue. Delay insensitive techniques such as dual rail, LEDR and 1 of 4 were introduced to alleviate the relative timing problems and provide a solution in which delays do not matter. However, they are susceptible to transient faults which can corrupt the data.

The basic idea of introducing resilience in our work is to provide some form of matching between different parts of the transmitted symbol so that the fault affecting one part can be filtered with the help of the other parts since the validity of the overall value is a 'collective responsibility' of all parts. In the phase-encoding [6] this is achieved by mutual adjudication between the wires. In the dual-rail framework, exploited in this paper, we build on the dual rail solution by introducing a pair of reference wires which can be compared with the pair of data wires in order to obtain the original transmitted data. The phase relationship between the reference and the data symbols provides the necessary information to obtain the original data. Both the reference and data symbols use four phases.

### 3. PROPOSED RESILIENT LINK

The proposed technique uses a pair of wires per data bit plus a further pair of wires for a reference which is associated with the data bits, thus the number of wires will be  $(n*2)+2$  for  $n$  bit wide link, Figure 2. Each data bit and the reference is represented by a symbol on their pair of wires (00, 01, 11, 10). If the data symbol is in phase with the reference the data is '0'. If the data symbol is 180° out of phase with the reference the data is '1'. Should the data symbol be out of phase by  $\pm 90^\circ$  the data is considered invalid, Figure 3. It is easy to detect invalid data with this system as an error on one of the wires of the data symbol will cause the symbol to be out of phase by  $\pm 90^\circ$  which is detected by the receiver.

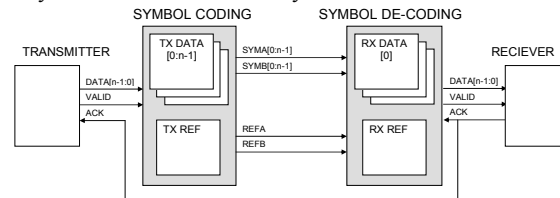


Figure 2 Overview of link

The reason of using a reference is that it allows the checking or validity of the symbol to see if it results in valid data or if there is an error present. A single reference pair of wires can be grouped with several pairs of data wires to support several bits transferred at a time. Each time a new piece of data is sent the reference increments around (moves around the quadrant by 90°) and the data moves either in-phase or 180° out of phase. By doing this there is only 1 transition on each pair of data wires and 1 transition on the pair of reference wires.

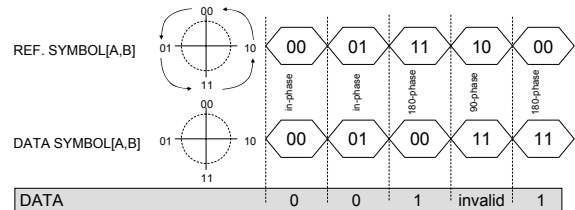


Figure 3 Symbol and reference phase relationship

The state diagram of the coding technique is shown in Figure 4. It can be considered as two cyclic planes. The lower plane which cycles round the symbol 180° out of phase to the reference while the data input is ‘1’ and the higher plane which cycles round the symbol in-phase to the reference while the data input is ‘0’. Entry and exit to the planes happens when the data input changes.

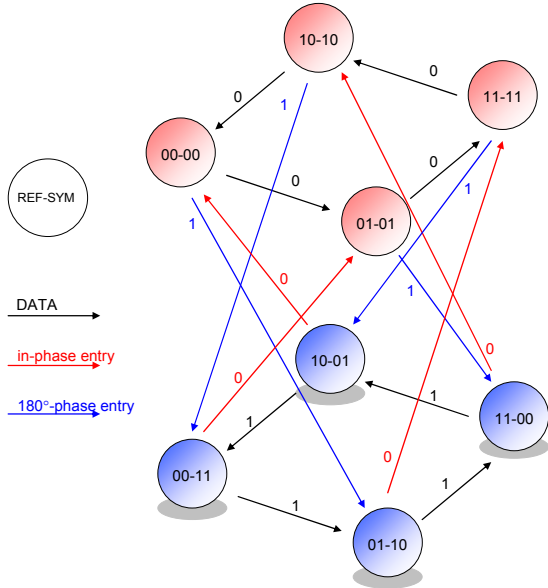


Figure 4 State Diagram of Coding

Table 1 shows comparison between a number of delay insensitive methods of data transfers that have been proposed for asynchronous links. LEDR, LETS and 1 of 4 encoding both improve on standard dual rail by reducing the transitions per bit but are still susceptible to transient faults. Multiple rail phase-encoding improves on these by offering resilience to transient faults and offering a reduced wire count when the number of bits is greater than four, but this comes at the expense of increasingly complex transmitter and receiver circuitry as the transmitter grows squarely and the receiver grows squarely or linearly dependent on the choice of decoding array in the receiver. The self correcting green coding scheme has high number of wires per bit so serialization can be employed to mitigate this. Our proposed approach offers a similar amount of wires per bit as dual rail and similar number of transitions per bit as LEDR and 1 of 4 as the number of bits increase from the point of view of power on the link and resilience to transients that multiple rail phase-encoding offers but with a linear growth in transmitter and receiver complexity.

Table 1 Comparison of proposed and existing links

Link	Wires /bit	Transitions/bit	Resilience to SEUs?	TX/RX growth
Dual-Rail [8]	2	2	N	Linear
1 of 4 [5]	2	1	N	Linear
1 of 4 LETS[9]	2	0.5	N	Linear
LEDR [7]	2	1	N	Linear
Phase-enc [6]	w/n	w/n	Y	Square
S-C Green[14]	3.75	-	Y	Linear
Proposed	2+2/n	1+1/n	Y	Linear

where  $n$  = number of bits and  $(w-1)! < 2^n < w!$

## 4. PROPOSED LINK ARCHITECTURE

The link consists of a transmitter for the reference symbol, a receiver for the reference symbol, transmitters for the data symbol and receiver for the data symbol. A single transmitter and receiver module pair for the reference is used with one or more transmitter and receiver pairs for the data as shown in Figure 5 for an  $n$  bit wide data source. For example, an 8 bit wide data source would require 1x TX REF, 1x RX REF, 8x TX DATA and 8x RX DATA modules. The SYMVALID outputs of the RX DATA modules can be ANDed together to form a single SIMVALID signal for the RX REF module. The VALIDO outputs of the RX DATA modules can also be ANDed together to provide a single VALIDO signal. We are aware that for performance, wire buffers will need to be developed to allow pipelining of long links, this work currently focuses on the coding of the data and the authors acknowledge that wire buffers need to be examined and introduced into the scheme.

### 4.1 Transmitter

The transmitter data module (TX data) is shown in Figure 6. REFA and REFB are registered into the two flip-flops to provide the SYMA and SYMB output when REFCHANGED goes high. The DATA signal will invert the symbol when high causing a 180° phase shift. REFCHANGED is generated each time REFA or REFB changes. Figure 7, the transmitter reference module (TX ref) is basically a grey code counter which increments the output REF[A,B] through the symbols 00, 01, 11, 10 each time VALID goes high.

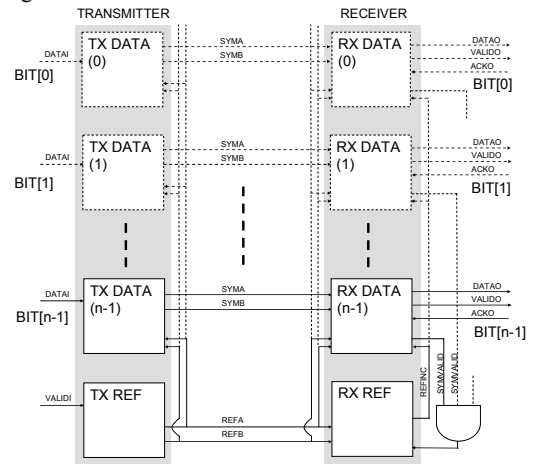


Figure 5 Circuit modules connectivity

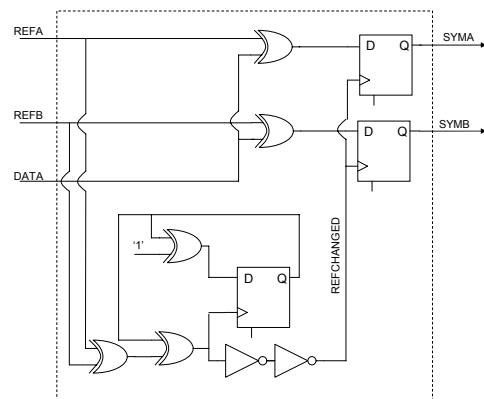


Figure 6 TX DATA Circuit

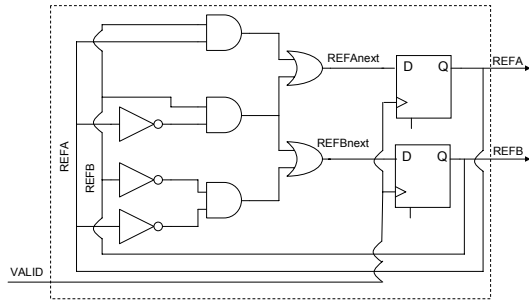


Figure 7 TX REF Circuit

## 4.2 Receiver

The receiver data module (RX Data) is shown in Figure 8. The circuit generates a SYMVALID signal if the REF[A,B] matches SYM[A,B] or the inverse, which is out of phase by 180°. When SYMVALID is high and REFINC goes high, DATA and VALID are registered into their respective flip-flops. ACK going high will clear the valid signal. The reference receiver module, Figure 9, compares the current REF[A,B] to the OLD[A,B] to see if it has incremented or remains the same. If it has incremented then REFINC goes high. When SYMVALID goes high REF[A,B] is registered into their flip-flops and REFINC goes low.

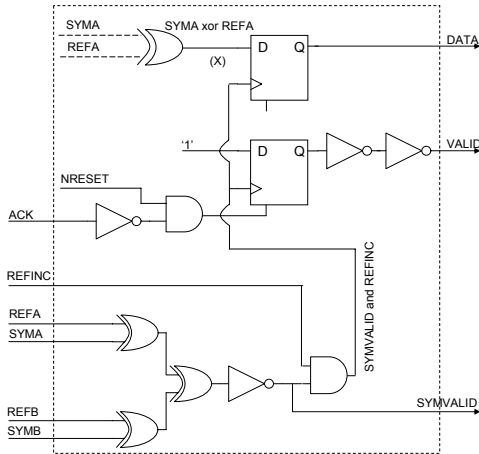


Figure 8 RX DATA Circuit

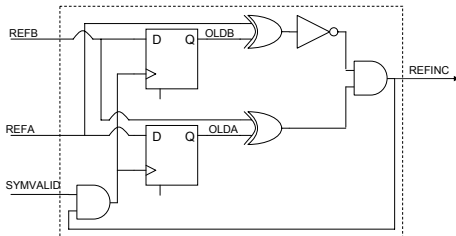


Figure 9 RX REF Circuit

## 4.3 Resilience

Although the link is resilient to transient faults it is not totally immune from them. Intuitively we can show that under certain conditions invalid data could be latched out on the receiver end. Examining Figure 8 and Figure 9 we can show the normal sequence of events when a valid reference and data symbol arrives at the receiver end, Figure 10(a). The valid data symbol

(SYM[A,B]) and reference (REF[A,B]) generates the SYMVALID which combined with REFINC through an 'and' gate generates the signal to latch SYMA xor REFA onto the DATA output. Provided that no transients affect or corrupt SYMA xor REFA before it is latched into the flip-flop then valid data is obtained. Thus the input to the flip-flop must remain stable during its setup time period. Figure 10(b) shows what happens if transients occur within the setup time of the flip-flop which latches DATA. The transient ripples through and corrupts SYMA xor REFA causing invalid data to be latched onto the DATA output.

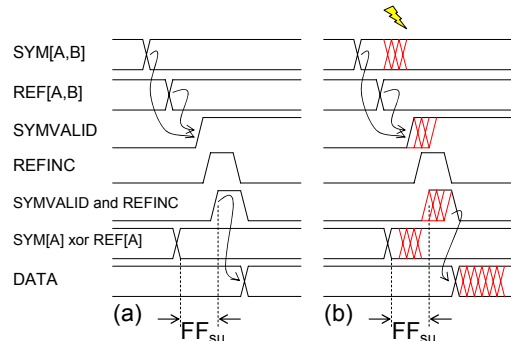


Figure 10 RX DATA timing

The setup time of the flip-flop used is approximately 125 ps nominal which can be obtained from the data sheet (ST CORE9GPHS HCMOS9 data book [17]). With this we can find out the probability if the data being corrupted if a transient fault does occur. We define  $T_w$  as the transient width ( $T_w > 0$ ),  $t_{period}$  as the inverse of the operating frequency and  $FF_{su}$  as the flip-flop setup time. We use a single event transient fault (SET) model with a fixed width. Assuming that the time it affects the signal is random within the symbol or data period  $t_{period}$  we can use the following formula to predict the probability that data will be corrupted.

$$P(\text{corruption}) = \frac{T_w + FF_{su}}{t_{period}}$$

Using a  $FF_{su}$  of 125 ps we can show the probability of the data being corrupted if a single transient occurs on one of the wire pairs and while data is being transmitted. Figure 11 shows the probability of corruption versus the operating frequency for transients widths of 100, 200, 300 and 400 ps if a transient occurs.

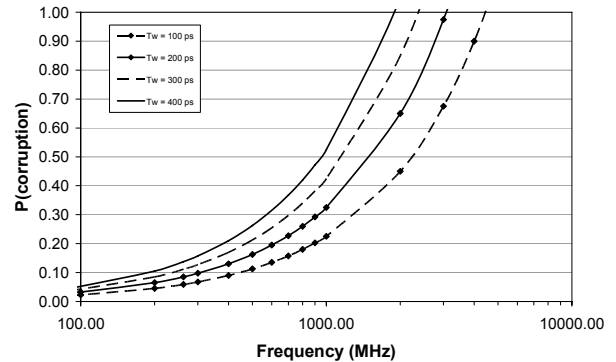


Figure 11 Probability of corruption for a single transient

## 5. RESULTS

To evaluate the resilience of the proposed link SpectreVerilog simulation were performed. The test bench scenario is shown in Figure 12. The receiver and transmitter are circuit level designs. The driver and receiver (TB driver.v and TB receiver.v) are verilog modules which generate the appropriate handshaking for the asynchronous interfaces. The pulse stream generator (TB pulses.v) uses a single bit in a 10 bit LFSR to generate several 300 ps wide pulses which are then XORed into the chosen signal. The output of the XOR will then have sporadic transients present within it. This can be spliced into any of the wires to provide a noisy or transient infected signal as required during simulation. If more than one wire is needed to have transients present then further pulse stream generators can be used with different LFSR starting seeds and frequencies as required.

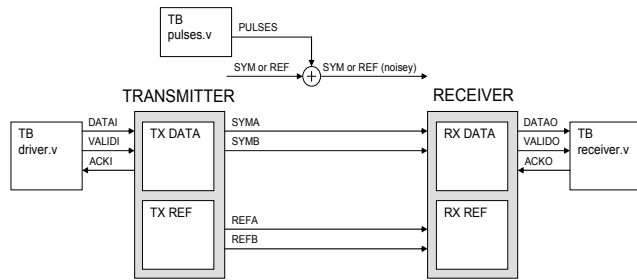


Figure 12 Test bench setup

Figure 13 shows the signal waveforms from simulation. As can be seen the transmitted data (DATAI) is the same as the received data (DATAO). Also note that REF[A,B] can clearly be seen incrementing through 00, 01, 11, 10, 00, ... SYM[A,B] can also be seen to be in-phase for DATAI=0 or 180°-phase for DATAI=1. Figure 14 shows transients on a SYM wire, note that the received data (DATAO) is received correctly even though transients are corrupting one of the symbol or reference wires. A similar result is obtained when the transients are on a single REF wire.

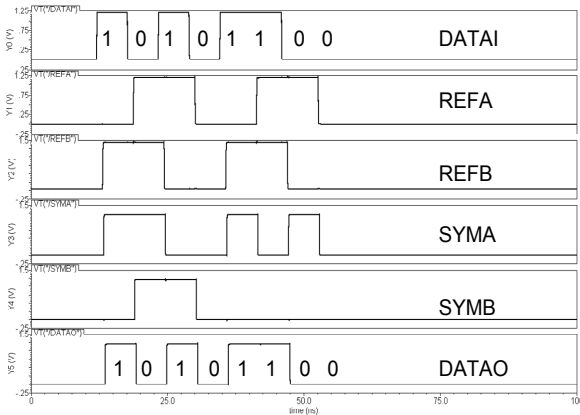


Figure 13 Reference and symbol signaling

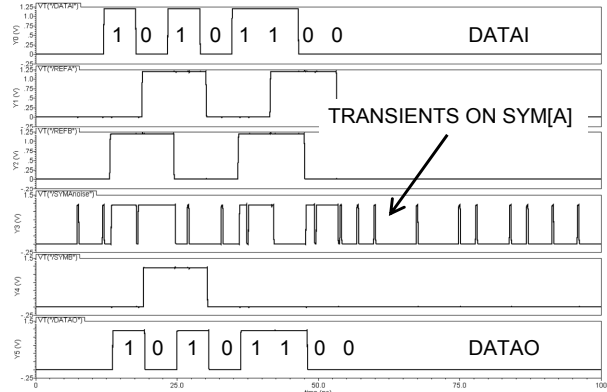


Figure 14 Transients on a symbol wire

In order to verify the probability of a single event transient corrupting the data a simulation was performed which sweeps the a transient through the symbol in order to corrupt the data. A series of 300 bits of alternating 1's and 0's were sent across the link and one of the symbol wires had a transient pulse superimposed on it by use of an xor gate to corrupt the data seen at the input to the flip-flop in the RX DATA circuit (marked X in Figure 8). The repetition rate of the transient was 555 MHz, this was chosen to be of a similar frequency to the rate at which the symbol changes (526 Msym/s) in order that the transient affects each symbol at a different position over the period it takes to transmit the 300 bits. Figure 15 shows the calculated and simulated probability of a bit error occurring with transient widths of 100 to 600 ps. The calculated values were obtained from the equation used to generate Figure 11. The simulated values were obtained from by sweeping a transient through the symbols and counting the number of bits in error. As can be seen the trend of the curve is similar to calculated, but slightly lower offset. This could be due to the fact that in simulation even if the transient affects the symbol enough to encroach into the setup time it may still not be enough to cause a violation within the analogue simulation.

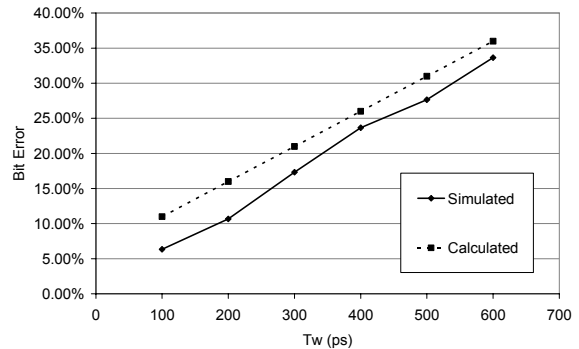


Figure 15 Simulated v Calculated Bit Error

To give an idea of the maximum operating frequency of the circuits the handshaking timings of the test bench driver and receiver were all set to 0 with the exception of DATAI<sub>VALID</sub> to VALIDI<sub>HIGH</sub> which was set to 300 ps to ensure that DATAI was valid before the VALIDI signal goes high. Running a simulation showed that 32 back to back transfers occurred in a time period of 30.3 ns. Thus a single transfer happens in 0.95 ns, giving a theoretical operating frequency of 1.056 GHz. It is important to note that the actual

operating frequency of a complete link would be lower than this as the handshaking timing used in the test bench driver and receiver would need to be based on the speed of the handshaking of the asynchronous circuitry interfacing to the link. To give insight of the extra latency introduced by the encoding and decoding the time from DATAI to DATAO in the simulated waveforms was obtained. The latency introduced by the circuitry is 0.8 ns. However, it is important to note that the latency in a physical implementation will be more than this as wire delays are introduced and need to be taken into consideration. Table 2 shows the area cost of the circuits in the 0.12 $\mu$ m technology used to simulate the circuits. The cost per bit is 409.49  $\mu$ m<sup>2</sup> and the cost of the reference is 262.24  $\mu$ m<sup>2</sup>. For example, for an 8 bit wide link the total area cost would be 3538.16  $\mu$ m<sup>2</sup> (262.24  $\mu$ m<sup>2</sup> + 8\*409.49  $\mu$ m<sup>2</sup>).

**Table 2 Area Cost of Link ( $\mu$ m<sup>2</sup>)**

Circuit	1 bit	8 bit	16 bit
TX DATA	215.84	1726.72	3453.44
RX DATA	193.65	1549.20	3098.40
TX REF	137.17	137.17	137.17
RX REF	125.07	125.07	125.07
Total	671.73	3538.16	6814.08

To give insight into the average power consumption within the various circuit modules the data pattern 0xFF00AAAAA00FF was sent bit serially from the test bench driver to the circuit. The breakdown of the power used in the four circuit modules is shown in Table 3. The total dynamic power for the DATA modules is 199.47  $\mu$ W. The simulation run time was 100 ns which gives an approximate energy usage of 0.0199 pJ to transfer the whole 56 bit data pattern meaning the bit energy is 356 fJ/bit. The REF modules energy per data transfer is 153 fJ. For example, for an 8 bit wide link the energy for each data transfer is 3001 fJ (153 fJ + 8\*356 fJ). As there is extra area and power cost associated with the coding there is a tradeoff of area and power if the scheme is used on every link within a NoC. If the cost of using the scheme on every link is too high then analysis of the NoC could be performed and the scheme used on certain links which require the added robustness of transient links.

**Table 3 Dynamic and Static Average Power ( $\mu$ W)**

Circuit	Power (dynamic)	Power (static)	Power (total)
TX DATA	91.72	0.72	92.47
RX DATA	106.26	0.74	107.00
TX REF	41.87	0.47	42.34
RX REF	43.00	0.32	43.32

## 6. CONCLUDING REMARKS

This paper has proposed and demonstrated a new asynchronous link that has resilience to transient faults. As the link is asynchronous the potential problems with synchronous design such as global clock distribution and clock skew have also been reduced. When comparing to current asynchronous links the link is shown to have a similar number of transitions per bit as level-encoded dual rail and 1 of 4 encoding but also achieves resilience to transients at the same time.

Validation of the link was carried out using synthesized gate level design and the maximum operating frequency of the circuit was

derived through simulation. It is hoped that the proposed link makes a valuable contribution to the area of efficient and soft-error resilient NoC architecture for multi-processor SoC.

## Acknowledgements

The authors would like to acknowledge the Engineering and Physical Sciences Research Council (EPSRC) for funding under grant no. EP/C512804 and EP/C512812.

## References

- [1] M. Amde, et al "Asynchronous on-chip networks," in *System-on-Chip: Next Generation Electronics*, B. M. Al-Hashimi, Ed.: IEE, 2006, pp. 625-52.
- [2] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," *Computer*, vol. 35, pp. 70-8, 2002.
- [3] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," DAC 2001, pp. 684-9.
- [4] S. Ogg, et al, "Serialized Asynchronous Links for NoC," in *DATE '08*.
- [5] J. Bainbridge and S. Furber, "Delay insensitive system-on-chip interconnect using 1-of-4 data encoding," in *7th ASYNC*, 2001, pp. 118-126.
- [6] C. D'Alessandro, et al, "Multiple-Rail Phase-Encoding for NoC," in *12th ASYNC*, 2006, pp. 107-116.
- [7] M. E. Dean, et al, "Efficient self-timing with level-encoded 2-phase dual-rail (LEDR)," *VLSI*, 1991, pp. 55-70.
- [8] T. Verhoeff, "Delay-insensitive codes — an overview " *Distributed Computing*, vol. 3, Number 1, 1998.
- [9] P. B. McGee, et al "A Level-Encoded Transition Signaling Protocol for High-Throughput Asynchronous Global Communication," *ASYNC '08*, 2008.
- [10] E. Dupont, et al, "Embedded robustness IPs for transient-error-free ICs," *Design & Test of Computers, IEEE*, vol. 19, 2002.
- [11] Teijo Lehtonen et al "Online Reconfigurable Self-Timed Links for Fault Tolerant NoC," *VLSI Design*, 2007.
- [12] Arthur Pereira, et al "Dependable Network-on-Chip Router Able to Simultaneously Tolerate Soft Errors and Crosstalk," *ITC '06*, pp. 1-9.
- [13] D. Rossi, et al, "Configurable Error Control Scheme for NoC Signal Integrity," in *IOLTS 07*, pp. 43-48.
- [14] H. Po-Tsang, et al "Low Power and Reliable Interconnection with Self-Corrected Green Coding Scheme for Network-on-Chip," in *NoCS 2008*.
- [15] A. Ejlali et al, "SEU-Hardened Energy Recovery Pipelined Interconnects for On-Chip Networks," *NoCS 2008*.
- [16] R. Dobkin et al., "Fast Asynchronous Bit-Serial Interconnects for Network-On-Chip," in *CCIT TR529 Technion: EE dept*, 2005.
- [17] ST-Microelectronics, *CORE9GPHS HCMOS9 TEC 3.2.a* vol. UNICAD2.4 / December 14, 2001, 2001.