



A high-speed interconnect network using ternary logic

Madsen, Jens Kargaard; Long, S. I.

Published in:
Proceedings of the 25th International Symposium on Multiple-Valued Logic

Link to article, DOI:
[10.1109/ISMVL.1995.513502](https://doi.org/10.1109/ISMVL.1995.513502)

Publication date:
1995

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Madsen, J. K., & Long, S. I. (1995). A high-speed interconnect network using ternary logic. In *Proceedings of the 25th International Symposium on Multiple-Valued Logic* (pp. 2-7). IEEE.
<https://doi.org/10.1109/ISMVL.1995.513502>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A High-Speed Interconnect Network using Ternary Logic

Jens Kargaard Madsen

Center for Broadband Telecommunications
Electromagnetics Institute
Technical University of Denmark
Email: jkm@emi.dtu.dk

Stephen I. Long

Department of Electrical and Computer Engineering
University of California, Santa Barbara
Email: long@ece.ucsb.edu

Abstract

This paper describes the design and implementation of a high-speed Interconnect Network (ICN) for a Multiprocessor System using ternary logic. By using ternary logic and a fast point-to-point communication technique called STARI (Self-Timed At Receiver's Input), the communication between the processors is free of clock skew and insensitive to any delay differences in buffers and wires. In addition, the number of signal wires and pins are reduced by 50 percent in comparison with a similar binary implementation. The ICN architecture is based on a crossbar topology and the high-speed part consists of two LSI GaAs chips, Interface and Crossbar, which were implemented in a 0.8 μm MESFET process. In a 4x4 ICN, communication at 300 Mbit/s per wire was demonstrated, which is twice as fast as pure synchronous and four times faster than pure asynchronous communication in the specific test set-up.

1 Introduction

The basic philosophy of a multiprocessor system is to have several processors working in parallel on the same task with the overall goal of increasing the performance or in some cases to obtain fault tolerance. The interconnection network (ICN) has a significant impact on the performance of the overall system. It is crucial that the increased performance, available by utilising processors in parallel, not be lost in transporting data through an inadequate ICN. Factors that influence this are many: Contention, throughput, latency, reconfiguration time etc.

With increasing data rates and clock frequencies in digital systems, problems as clock-skew and signal propagation delays become more severe. The latter means that signal propagation delays between chips in the network becomes comparable or even larger than the clock cycle. To overcome this, either a lower network

clock frequency can be used or an asynchronous communication technique. Both techniques can result in a slow physical communication across the network, creating a bottleneck for the entire system. An example of a slow asynchronous communication is the two-cycle handshake protocol. Here each data transfer is characterised by a request phase and an acknowledge phase corresponding to a transfer cycle time equal to the round trip delay between the transmitting and receiving processor. Other asynchronous techniques exist for optimising the communication between synchronous subsystems [1]. A relatively new interprocessor communication technique called STARI (Self-Timed At Receiver's Input) [2] provides communication bandwidth comparable with techniques using phase lock loops, but uses only digital circuit techniques. This is in contrast with phase lock loop techniques, where both digital and analogue circuits are required.

This paper describes the design and implementation of a high-speed ICN based on a crossbar topology and the STARI communication scheme using ternary logic and signalling. By using ternary logic, the communication between the processors is free of clock-skew and insensitive to any delay differences in buffers and wires. In addition, the number of signal wires and pins are reduced by 50 percent in comparison with a similar binary implementation. Two LSI GaAs chips, Interface and Crossbar have been designed and implemented in the Vitesse H-GaAs II MESFET process. In a 4x4 ICN, communication at 300 Mbit/s was demonstrated, which is twice as fast as pure synchronous and four times faster than pure asynchronous communication in the specific test set-up. These results are the first to demonstrate STARI in operation and at high speed.

2 System design

This section describes the system design of the implemented ICN. First a brief description of the overall

system architecture, which is based on a crossbar topology, is given. Next, a subsection describing the STARI communication scheme follows. The remaining subsections describe the two chips, Interface and Crossbar, that together constitute the high-speed part of the ICN.

2.1 System Architecture

The system architecture of the implemented ICN is based on a crossbar topology as illustrated in figure 1. The ICN consists of four nodes all assumed to be placed in a mesochronous region. This means that the clock frequency is the same for all processor interfaces, but the clock phases might be different and independent in time.

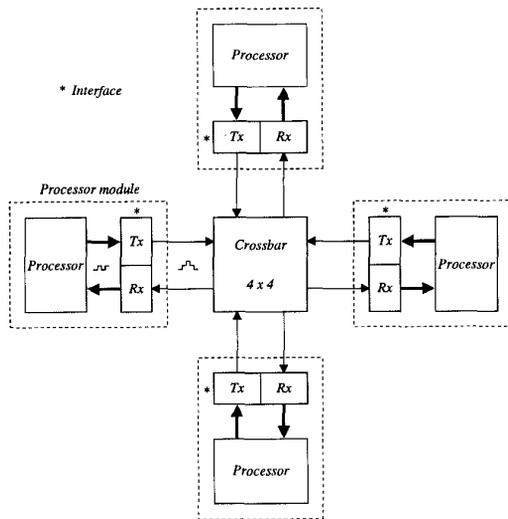


Figure 1: System architecture of the ICN.

The network requires the design and implementation of three chips: The control chip (not shown), the network interface chip, and finally the crossbar switch chip.

The control chip is responsible for configuring the network when there are requests corresponding to a set-up or removal of a connection between two processors. An important task is a fair arbitration for servicing the requests. The operation speed of this chip, compared with the operation speed of the physical links between the processors, is less important. This is so because the procedure involved in setting up a connection is only active at the beginning of a transfer, thus it does not degrade the performance of the communication in steady state. The high-speed parts of the system has been the

primary objective in this work, and the control chip will not be discussed any further in this paper.

The interface chip, which is responsible for the communication between the local processors through the ICN, implements the STARI interface. Basically, the interface chip eliminates phase misalignment between two processor clocks and encodes the binary data into ternary data values (and reverse), thus allowing maximum transfer rate at the global system clock frequency, independent of the physical distance and logic (buffer and cross points) in between. It means that several data values can be travelling on the same link at the same time in case the delay from one processor to another processor is larger than the cycle time of the system clock. All data buses are one bit wide.

The last chip is the crossbar chip which performs the actual switching. It establishes connection between two data lines (input port and output port) to each interface chip, both carrying ternary signals. It also receives a number of control signals from the control chip, used when connections are set-up and removed.

2.2 STARI communication

The STARI [2] communication is a point-to-point communication technique as illustrated in figure 2. STARI is a hybrid of self-timed and synchronous techniques. This makes it possible to eliminate clock-skew related problems and still obtain maximum transmission transfer rate. STARI assumes that the transmitter and receiver operate in a mesochronous region, thus assuring that the receiver accepts data at the same frequency as the transmitter sends data. However, the interface is insensitive to arbitrary phase differences as indicated by the delays T_{tx} and T_{rx} .

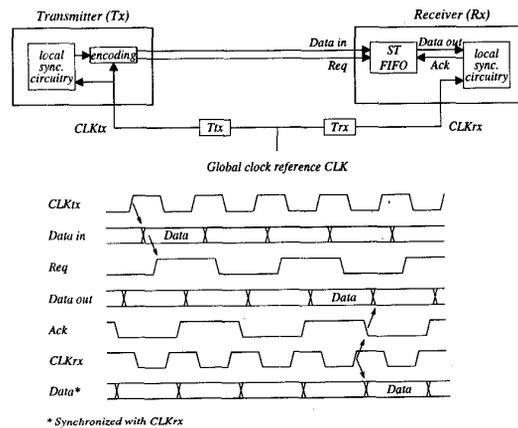


Figure 2: STARI interface.

The key component of STARI is a self-timed FIFO (First In First Out) queue placed at the input of the receiver. The transmitter sends a new datum on its data line(s) *Data* and toggles its request signal *Req* in each clock cycle. The FIFO reads the data line(s) by using the request information and due to its self-timed nature, the inserted data ripples to the end of the structure asynchronously. At the same time, the receiver reads one datum from the FIFO in each clock cycle by toggling its acknowledge signal *Ack*. Thus in steady state, one datum is inserted and one datum is removed from the FIFO in each cycle assuming that the FIFO can complete each insert and remove operation within one clock cycle. Once properly initialised, this prevents the FIFO from overflowing or under flowing.

Ternary encoding of the request signal into the data signal(s) by separating each datum (low or high) with a spacer (middle) as illustrated in figure 3 has the following advantages. First, it reduces the number of signal wires from $n + 1$ to n which in case of a single data signal corresponds to a 50 percent reduction in signal wires and pins. Second, it eliminates the requirement of matched delays between the request signal and the data signal(s), thus making the transmission insensitive to any spread of delays in buffers or signal wires. For high-speed designs where issues as interconnect and delay spread are important design factors, these properties are especially beneficial. Less attractive is the utilisation of only half of the available communication bandwidth, but due to the encoded request no dummy data values are required before the data can be assumed to be valid after initialisation. This property should be compared to "self-clocked" binary coding techniques (e.g. the Manchester code) which require some kind of phase lock oscillator circuitry and unique start sequences in order to recover the transmitted data values as well as detection of the arrival of real data values.

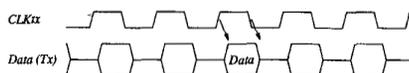


Figure 3: Ternary encoding.

2.3 Interface chip

A block diagram of the interface chip is shown in figure 4. The interface chip consists of a transmitter part and a receiver part. Two binary control signals (*Tx*, *Rx*) are received from the control chip. These control signals are used for activating transmission and reception of data. This means, the control chip responds to a connection set-up request by activating the *Tx* signal of

the transmitting processor and the *Rx* signal of the receiving processor, and the transmission can start.

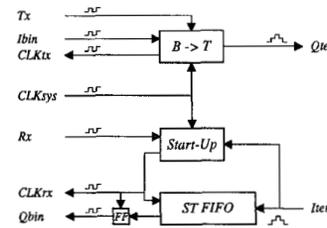


Figure 4: Interface block diagram.

The transmitter part implements the encoding of the binary data signal *Ibin* into the ternary signal *Qter*. When *Tx* is high, thus releasing *CLKtx*, the output is either high or low determined by the binary data signal when *CLKtx* is high, and middle when *CLKtx* is low. Otherwise, the output is kept at the middle level. Thus, it is possible to distinguish successive data values when the transmission is active, and also detection of the first data value arrival at the receiver input, because the output is kept at the middle level when the transmission is inactive.

The receiver part includes the self-timed FIFO, a start-up circuit and a retiming flip-flop. The self-timed FIFO operates with ternary values (false (*f*), spacer (*s*), true (*t*)) where each FIFO element works as shown in figure 5.

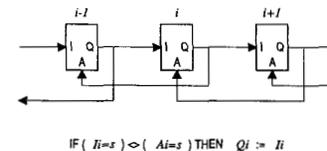


Figure 5: Self-Timed FIFO queue.

Here each FIFO element compares the status (*f/t* or *s*) of the succeeding stage with the status of its own input. If these are different, the input is passed to the output. Otherwise, the output remains unchanged. This ensures that each data variable alternate between false or true and spacer without overwriting any data, regardless of delays in wires and FIFO elements [3].

After the network is reconfigured by the control chip, the start-up circuit (see figure 6) recognises the arrival of the first valid data by monitoring the ternary input signal *Iiter*. A change from *s* to *f/t* signifies new data. The *CLKrx* is started when the FIFO is approximately half filled, which allows absorption of possible variations in global clock frequency due to clock

drifting. CLK_{rx} is used as acknowledge signal (i.e. the acknowledge signal for the last FIFO element in the FIFO queue, see figure 4) and a positive edge triggered flip-flop is used for reading the valid data output of the FIFO, thus decoding the ternary data to binary data values again. The relationship between the number of synchroniser flip-flops and FIFO elements is chosen so that the FIFO is approximately half-filled, and with a start-up circuit including two flip-flops it can be seen from figure 6 that approximately 2-3 valid data values are written into the FIFO. Thus a FIFO capacity of nine elements (i.e. 4-5 valid data values) is chosen.

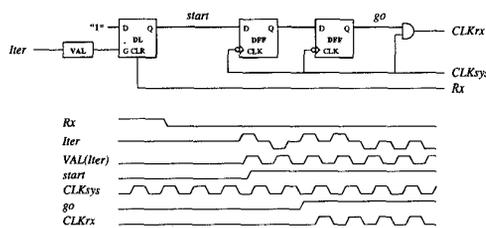


Figure 6: Start-up circuit.

The arrival of the first data value is asynchronous with respect to the clock signal at the receiver. Thus, synchronisation failure is possible as in any non-synchronised system, but the metastability problem can be made arbitrarily small by increasing the number of synchroniser flip-flops and FIFO elements. In a given implementation it is a trade-off between reliability and chip area when choosing a specific number of flip-flops.

2.4 Crossbar chip

A block diagram of the 4x4 crossbar chip is shown in figure 7. The crossbar chip is based on a multiplexer architecture, where each switching cell is a multiplexer

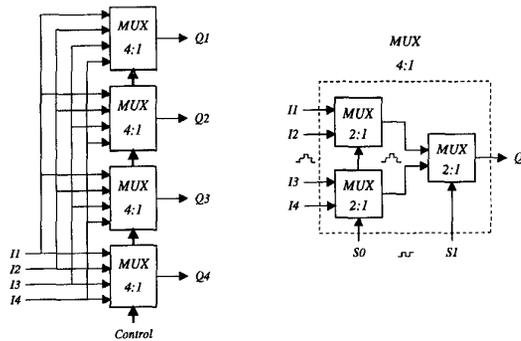


Figure 7: Crossbar block diagram.

logic circuit as opposed to a transmission gate [4]. This has the advantage of keeping the loads small and constant for each internal node, making it feasible for high-speed operation.

The four input signals, I_1, \dots, I_4 (ternary signals), are distributed to each of the four 4-to-1 multiplexers corresponding to the four ternary outputs, Q_1, \dots, Q_4 . The control lines (S_0, S_1) to each 4-to-1 multiplexer determine which input is connected to the output of the multiplexer. Broadcasting is possible, and a connection is stable until a new configuration of the particular output occurs. Finally, the three 2-to-1 multiplexer gates in each 4-to-1 multiplexer are ternary logic gates.

3 Implementation

In the implementation of the interface chip as well as the crossbar chip both binary and ternary logic gates are used. To obtain high speed and compatibility between the two logic structures, the circuit topologies chosen are based on a current mode logic structure and the GaAs MESFET process technology described in the following subsections.

3.1 Circuit Topology - TSCFL and SCFL

Figure 8 shows the basic structures (inverters) of the two logic families. The binary logic family is SCFL [5] (Source Coupled FET Logic), which is similar in structure to its binary counterpart in silicon ECL (Emitter Coupled Logic). The basic principle is steering of the current path from ground to V_{SS} in the differential stage either through the left branch or the right branch corresponding to two logic states.

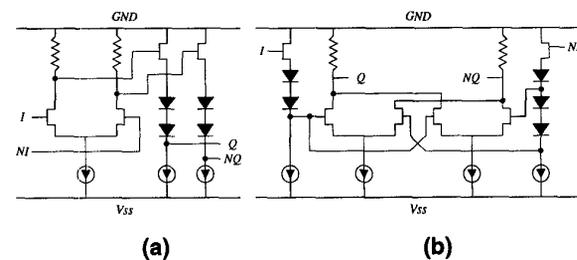


Figure 8: Basic structures of (a) SCFL and (b) TSCFL.

The ternary logic family is TSCFL [6] (Ternary Source Coupled FET Logic). TSCFL has two differential amplifiers in the differential stage, and by using a more sophisticated level shifting, three logic states can be obtained. When valid data (*low* or *high*) are applied to the input terminals, the function of the ternary inverter is

the same as that of its binary counterpart. When the input data (*middle*) is invalid, the output also produces invalid data.

The more complex gates as AND, OR and flip-flops are made by stacking differential pair of transistors in the differential stages in both families. Because of their differential nature, high speed and good noise immunity are characteristics of SCFL and TSCFL, and thus suitable for high-speed design. In the former GigaBit Logic GaAs MESFET process, the basic structure in TSCFL demonstrated speed of operation above 1.5 Gbit/s which was slightly lower (25 %) than the maximum speed of operation of a similar binary SCFL implementation [6,7]. For the more complex gates the gap in operation speed between the two families was more severe (25-50 %). The relative lower speed of operation is mainly due to a larger peak-to-peak voltage swing required in TSCFL.

3.2 Fabrication and Key Figures

The interface and crossbar chips are implemented and fabricated in the HGaAs-II process from Vitesse Semiconductor Corporation [8], which is a 0.8 μm self-aligned MESFET process. It has three metal layers and features both E- and D-FETs.

Both chips are ECL compatible. The logic levels for the external ternary signals are -1.4 V (low), -0.7 V (middle) and 0.0 V (high). With a typical power consumption of 5 mW for the internal gates and 200 mW for the 50 Ohm output buffers, the total power dissipation for each chip is 1.3 W. Finally, both chips occupy 2.15x2.45 mm² and are packaged in 28 pin packages.

4 Performance

Evaluation of the chips is carried out in two scenarios. The first scenario is the system test, where two interface chips are connected to the crossbar chips. Due to test limitations of the system test set-up only a '...010101...' or a constant data pattern could be transmitted across the network. The second scenario is a test of the interface chip alone in order to demonstrate the STARI communication with a pseudo random bit pattern by using a high-speed pattern generator.

4.1 System test

Figure 9 shows the test set-up, where two interface chips are connected to the crossbar chip. The remaining two output ports are connected to the oscilloscope and used as monitors for the two other ports (connections) by using the multicast function of the crossbar chip. The crossbar chip is configured as follows: $Q_1=I_1$, $Q_2=I_1$, $Q_3=I_1$ and $Q_4=I_2$. As binary stimulus to the two

interface chips, representing data from the local processors, both chips receive the binary pattern '...010101...', which is represented by the ternary pattern '...fstfsfstfst...'.

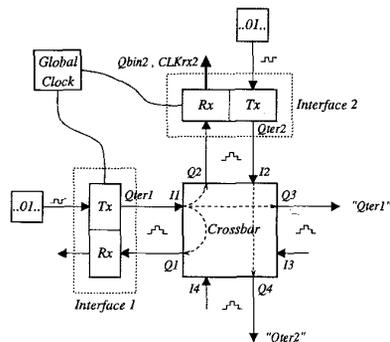


Figure 9: System test set-up.

The global clock frequency was varied continuously from 10 MHz to 300 MHz and figure 10 shows the situation at 300 MHz. The 300 Mbit/s is measured to be the system maximum communication speed which corresponds to the maximum speed of operation of the self-timed ternary FIFO. The transmitting part of the interface chip and the crossbar chip are tested successfully up to 500 MHz in the test set-up.

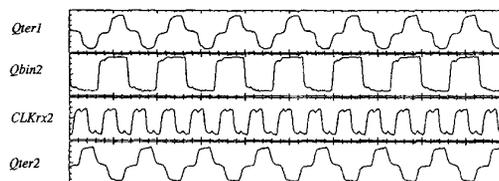


Figure 10: System test at 300 MHz (5 ns/div, 800 mV/div).

The STARI technique is robust against variation in the global clock reference as expected (demonstrated by the continual change from 10 MHz to 300 MHz). The propagation delay of the crossbar chip is measured to 1 ns and with approximately 1 m of transmission length corresponding to 5 ns between interface 1 and interface 2 a total transmission delay of 6 ns between the two interface chips is found. At 300 MHz (3.3 ns) this corresponds to two data values travelling between the two interface chips at the same time. This is almost twice as fast as pure synchronous (6 ns delay corresponds to a maximum data rate of 167 Mbit/s) and four times faster than two-cycle self-timed communication (twice the time of a one way transfer corresponding to a round trip delay

of 12 ns and a data transfer rate of 83 Mbit/s) in the specific test set-up. These transfer rate estimates (167 Mbit/s and 83 Mbit/s) are optimistic, because they do not include delays in output and input circuitry of the transmitter and receiver, which will lower the data transfer rates further in comparison with STARI.

4.2 STARI test

In order to verify STARI communication with an arbitrary bit pattern, a test of the interface chip alone using a high-speed pseudo random bit generator as stimuli is made. By connecting the ternary output to the ternary input of the same interface chip via a 50 Ohm cable (length 0.5 m) and the pseudo random generator to the binary input, a pseudo random bit pattern is applied. Figure 11 shows measurement results at 300 MHz where the curves from the top are: The clock signal of the transmitting part, the ternary output (i.e. the ternary input also), the clock signal of the receiving part and finally the decoded binary output. The 300 MHz is again the maximum transmission rate of the interface chip corresponding to the maximum speed of operation of the self-timed ternary FIFO.

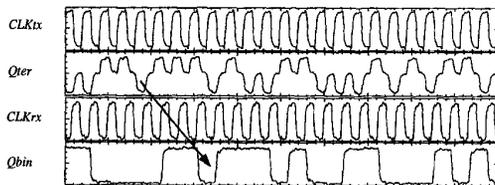


Figure 11: Test of interface chip alone with a pseudo random bit pattern at 300 MHz (5 ns/div, 800 mV/div).

5 Conclusions

In summary, this work has demonstrated the use of ternary logic/signalling in the design of a high-speed interconnect network based on a crossbar topology and the STARI communication technique, that include both synchronous and asynchronous circuitry. By using ternary logic/signalling, the number of signal wires and pins were reduced by 50 percent compared with a similar binary implementation. Another important result of the

work is the verification of STARI communication. Using this technique communication speed at 300 Mbit/s in a 4x4 ICN was demonstrated. The results obtained are the first to demonstrate STARI in operation and at high speed.

Acknowledgement

The authors would like to thank the Danish Technical Research Council for funding the project under grant No. 16-4484-3.

References

- [1] D.G. Messerschmitt, "Isochronous Interconnect," Chapter 6 in *Synchronization Design for Digital Systems*, by T.H. Meng, Kluwer Academic Publishers, 1991.
- [2] Mark R. Greenstreet and Kai Li, "Simple hardware for fast interprocessor communication," Technical Report CS-TR-242-90, Department of Computer Science, Princeton University, Princeton, NJ, January 1990.
- [3] J. Staunstrup and M.R. Greenstreet, "Designing Delay Insensitive Circuits using Synchronized Transitions," Technical Report 1989-60, Department of Computer Science, Technical University of Denmark, 1989.
- [4] S.E. Butner et al., "A Fault-Tolerant GaAs/CMOS Interconnection Network for Scaleable Multiprocessors," *IEEE Journal of Solid-State Circuits*, Vol. 26, No. 5, pp. 692-705, May 1991.
- [5] S. Katsu et al., "A GaAs Monolithic Frequency Divider using Source Coupled FET Logic," *IEEE Electron Device Letters*, Vol. 3, No. 8, pp. 197-199, Aug. 1982.
- [6] J.K. Madsen, "TSCFL - A New GaAs High-Speed Family for Delay Insensitive Circuit using Ternary Logic Elements," In proceedings of the European Gallium Arsenide Applications Symposium, ESTEC, Noordwijk, The Netherlands, 1992.
- [7] J.K. Madsen, "Ternary High-Speed Logic for Self-Timed Systems," Ph.D. Thesis ID-TR:1993-115 (in Danish), Department of Computer Science, Technical University of Denmark, 1993.
- [8] *Foundry Design Manual*, Vitesse Semiconductor Corporation, 741 Calle Plano, Camarillo, CA93012, US.