The Origins of SpaceWire

Paul Walker
4Links Limited
The Mansion, Bletchley Park
Milton Keynes, MK3 6ZP, England
Email:paul@4Links.co.uk

INTRODUCTION

SpaceWire is a new standard, having been published in January this year (2003). The technology and ideas in SpaceWire can, however, be traced back to the era of the minicomputer in the 1960s and '70s, and this paper traces the evolution since then. It can be seen that, throughout this evolution, key principles have been maintained, such as symmetry, asynchronous point-to-point links, scalability, and modularity.

THE PROBLEMS WITH THE BUS INTERFACE

Even 40 years ago, there was concern that a single bus in a system was a bottleneck. As you add nodes to the bus, you need the bus to go faster, but because of the extra loads, it actually goes slower. So the bus limits performance. It is also a prime example of a Single Point of Failure, because failure on any node can disable the entire bus, and any break in the bus splits the bus into two buses that can not communicate. Figure 1a, b, and c, show these problems.

OVERCOMING THE BUS INTERFACE PROBLEMS

One of the first computer systems to recognize the bus interface problem and to do something about it was the Modular One from CTL (Computer Technology Limited) in the UK. As its name implies, it was highly modular, with modules such as a processor, a memory (core memory in those days), disc controller, and a communications multiplexor able to connect to many comms lines and/or terminals. Each module was a box, and the boxes could be build into small and large systems, with any number of any type of module. Fig. 2 consists of a single processor (‘1.11’), single memory (‘1.21’), tape reader and console. Fig. 3 shows the point-to-point interface (‘1.01’) between the two modules. This paper will show that SpaceWire has evolved from this 1.01 point-to-point interface of Modular One.
Of course Fig. 2, with just a single processor and single memory, only needs a single 1.01 point-to-point interface. Fig. 4, with six modules, needs rather more. And Fig. 5, with up to 120 modules, would use several hundred interfaces. The modules of Fig. 5 could include, say, 16 processors, 64 memories, and a variety of different I/O controllers. Modular One was, as a single processor, significantly higher performance than competing minicomputers of the day, and the ability to use multiple processors to give even higher performance was unique in minicomputers. The key feature that made these multiple processors possible was the point-to-point interface.

Those familiar with SpaceWire will recognize the concept of point-to-point interfaces, and they will recognize the scalability made possible by such interfaces. Are there more similarities between the Modular One interface and SpaceWire?

Yes. The Modular One interface was asynchronous, it was flow-controlled, and it was symmetrical, all like SpaceWire. Being asynchronous meant that whatever the speed of the equipment, whatever the length of cable, the modules could always communicate, without the need for accurate clocks, phase locking, or global synchronization or even synchronization between the two ends of the interface. The transfers across the interface were very simple atomic transactions, but they had to follow the correct sequence, and, once an interface was in use for a transaction, it completed that transaction before starting another. Symmetry simplifies the use of the interfaces. You don’t have to worry which end you’re plugging into which end of the connection. And you don’t have to design two genders of interface both in the cable and in the electronics at each end.

There are, of course, major differences between the Modular One interface and SpaceWire, but it is remarkable that, an interface was designed nearly 40 years ago with the characteristics: point-to-point, asynchronous, flow-controlled, symmetrical, and scalable. With these interfaces, it became possible to assemble computer systems with any number of processors, memories, and peripherals, without the bandwidth and latency limitations of a single bus, and without the processor-centric view of the world presented by the mainframe or PC. The interface was the fundamental enabling technology that made these systems modular, that gave them their name of Modular One.

**MODULAR ONE IN THE SPACE INDUSTRY**

CTL successfully sold a number of Modular One systems to the European Space Agency, and Fig. 6 shows part of these systems at ESTEC. The ESA systems incorporated a number of CAMAC crates, and used a special DMA interface to CAMAC in addition to the standard interface so far described. The author had a small involvement with this contract, in helping to commission the interface to the CAMAC crates.
THE ARCHITECT OF MODULAR ONE

The architect of Modular One was Iann M Barron who wrote in 1977, after having left CTL:

There is an even less favourable relationship between processing and interface capability [in microprocessors], measured in silicon area. Because of the space consumed by the drive transistors and the bonding pads, 16 bits of interface occupies a similar area to a simple processor, so that again, there is a strong rationale for trading interface for processing wherever possible. This will lead to a predominance of serial interfaces and information compression techniques.

... The pinouts of the transputer may be regarded as forming an interface. At present, the definition of these pinouts [in microprocessors] is far too irregular and uncontrolled to be regarded as a satisfactory interface. The definition of a proper interface for the transputer would greatly facilitate its use both singly and in assemblies, as well as reducing the pin count. Pinouts represent a major cost in chip area and packaging, so that the development of improved transputer interfaces will lead to useful cost reductions.

Iann Barron, with partners from the US, formed the company INMOS in the following year, 1978, to develop these ideas into the transputer. Serial interfaces were clearly to be a major factor in the transputer's development, greatly facilitating the transputer's use both singly and in assemblies, and giving useful cost benefits.

THE TRANSPUTER AND ITS LINKS

It became possible to put processor, memory, and asynchronous symmetrical interfaces onto a single chip, and this opportunity was taken with the transputer. The block diagram in Fig. 7a is taken from early publicity material that INMOS produced for the transputer (the IMS T424), clearly showing the significance of the four serial links. Fig. 7b shows a packaged die of the later T800 floating-point transputer, with the four links on the left towards the top.

The cost benefits are clearly visible from Fig. 7b. Overall, the four links, including the physical layer interface, all the serializing and de-serializing (SERDES) and DMA logic for each direction for each link, take up about the same space as the fixed-point processor. By comparison the on-chip RAM, the floating-point processor and the memory interface (including all its pins) each take up significantly more chip area.

At the time the transputer was introduced, Ethernet was a shared bus (yellow Coax) running 10Mbits/s half duplex, shared between everything connected to the yellow Coax bus. Each Ethernet interface needed a chip-set of three chips, compared with a link being around 2% of a single chip on the transputer and its DMA engine another 2%.

Performance of the early transputer links was modest, but at 20Mbits/s in each direction (full-duplex) a single link was well over twice the performance of an Ethernet connection. The transputers had four of these full-duplex link interfaces running at 20Mbits/s, to give a total of 160Mbits/s per transputer.

The 160Mbits/s provided what, at the time, was outstanding performance between chips, yet the transputer links retained the characteristics that had been built into the Modular One interface. So the transputer links were also
symmetrical, they were asynchronous, flow-controlled, and they made it possible to build highly modular systems up to any size. The most obvious new feature of the transputer links was that they were serial, and this attribute has clearly been followed through into SpaceWire. The embedded DMA engines for the transputer links enabled data transfer to occur concurrently with processing, and access to the links was built into the instruction set to minimize the processing overhead of setting up a transfer. We have yet to see these fully to be carried through into SpaceWire.

TRANSPUTERS IN SPACE

Transputers have been used in space on several missions, including those from Surrey Satellite Technology Limited (UK), from SunSpace (South Africa), and the SOHO mission, which was a collaboration between ESA and NASA. We believe there were other missions which were classified. The transputer links were particularly used for connections between camera and processor, perhaps a pointer to one of the principal initial applications of SpaceWire.

An interesting aside, which the European space community might observe, is that the transputers that were flown were not Rad-Hard. They were selected from batches which performed better than others in the presence of radiation. But the key property of the transputer that made the use of non-rad-hard components possible was the fault-tolerance enabled by the links. Fig. 8a shows a fully connected network of transputers (actually the 4LLinks Limited company logo), with possible five-fold redundancy. Fig. 8b shows three transputers connected in a Triple-Modular-Redundancy (TMR) arrangement, with each transputer having two spare links for connecting to other devices. The practical effect of this, on the SOHO mission, for example, is that the cameras and transputers have continued to function even while a number of other failures have caused at least temporary malfunction.

Table 1: SSTL Satellites which used transputers
(provided by Alex da Silva Curiel, SSTL)

- UoSAT-4 (1990) (ESA funded experiment)
- UoSAT-5 (1991) (SSTL design from this flight onwards)
- KITSAT-1 (1992)
- PoSAT-1 (1993)
- FASAT-Alfa (1995)
- FASAT-Bravo (1997)
- TMSAT (1997)
- UoSAT-12 (1999)
- Tsinghua-1 (2000)
- TiunoSat (2001)

Fig. 9a shows the SOHO satellite, which used transputers and was an ESA/NASA collaboration. Fig. 9b shows a photo of the Sun taken by the EIT camera on SOHO. Fig. 10 shows TiunSat, one of the several SSTL satellites which used transputers.
MODULARITY WITH THE TRANSPUTER

With so many common characteristics between the transputer’s links and the Modular One’s interface, it comes as no surprise that the transputer was used to build modular systems. The internal memory of the transputer was enough for many tasks, but the trend over the years has been to use more and more memory, and so the transputer needed a memory interface which used lots of pins and lots of silicon area. The memory interface was particularly easy to use, and incorporated a highly flexible memory controller, but the overall package of 84 pins required specific printed circuit board design. A desire to make prototyping transputer systems even easier led to the development of TRAnputer Modules (TRAMs). These were conveniently packaged onto small PCBs and the obvious interface to use between the PCBs was the transputer links. The TRAM standard provided powerful building block computers, all of which fitted in a (rather wide) 16-pin Dual-in-Line package (DIP). Fig. 11 is one of the publicity shots for the transputer, using one of these 16-pin TRAMs as an example of how much can be squeezed into a sardine can! Many of the TRAMs were the minimum size with just 16 pins, referred to as “Size1”. If that size of PCB was insufficient, multiple 16-pin footprints could be used for a single TRAM, and there were many produced as “Size2, Size4” or larger. With these larger sizes, only 16 pins were used per TRAM, and so additional TRAMs could be stacked above, much as Lego™ bricks are assembled on top of each other.

As well as the simplest combinations of transputer and memory, many TRAMs were produced which had interfaces to other devices and standards. Fig. 12 shows an Ethernet interface TRAM. Other interface TRAMs included RS232, Disc drives, graphics devices including camera and displays, and indeed many of the interfaces that SpaceWire is currently being interfaced to. TRAMs were also produced for DSP and for extended links such as RS422 and optical fibre.

The TRAMs were sufficiently simple that assemblies of them could be put together on matrix boards, just using 16-pin DIP sockets. It was also easy to build carrier boards, and these were built for many form factors such as the PC, Mac, the NEC PC, VME, and a variety of other form-factors. Fig. 13 shows a carrier board for the Mac, populated with four Size2 TRAMs. Although the TRAM standard was published by INMOS and used internally for demonstrations, the first company to commercialize TRAMs was Levco, whose Mac board is shown in Fig. 13. INMOS and many other companies followed Levco and TRAMs became a highly successful open standard.
THE T9000, DS-LINKS, AND ROUTING SWITCHES

The INMOS T9000 built on the serial links of the original transputers, by making them much faster (100Mbits/s), by significantly improving the flow-control mechanism, and by embedding a packet protocol so that virtual channels could be multiplexed onto a single link.

The new links used a new coding scheme that embedded clock with data by using two gray-coded signals, Data, and Strobe, such that the Strobe signal made a transition on every bit where the Data signal did not change. The Data and Strobe signals resulted in the links being described as DS-Links, a name that was trademarked by STMicroelectronics, the company that had bought INMOS.

While introducing new capabilities, the DS-Links retained the characteristics we have been following, the symmetry, the scalability, the modularity, and flow-controlled asynchronous links.

The packet protocol made it possible to build simple hardware-based routing switches (with the earlier transputers, any routing had to be done by the transputers' processors, which reduced processing performance and increased latency). By contrast, with hardware routing switches, packets could be very quickly routed through even large networks with very short delay. Fig. 15 shows the C104, the first packet routing switch for these links. It had 32 ports, plus a control port, and at its time was probably the highest throughput single-chip packet switch in existence.

FROM DS-LINKS TO IEEE 1355

Packet switching has been increasingly used for communications, for both short distance Local Area Networks (LANs) and for Wide Area Networks (WANs). The T9000 made packet switching possible between collections of transputers and link-interfaced peripherals on a printed circuit board (PCB). Restricting the distance to within a PCB or box seemed to be an unnecessary restriction on the distance of connections and there was an obvious opportunity to extend the advantages of the new technology beyond the distances achievable on a PCB. The author therefore consulted with colleagues and customers to propose a standard that would carry DS-Link signals through cables, that would allow connections between boxes and equipments that were interfaced by these links.

The proposed standard was initially for internal use within INMOS/ST. As well as the version for copper cables, up to 10 metres, a fibre version was proposed with a much longer reach. Interestingly, the fibre version of 1355 included a mechanism for distributing time, so was in some ways a precursor to the time-codes of SpaceWire. The copper version of 1355 was used by INMOS and a number of customers, particularly including CERN, who built a large network of up to 1024 nodes, all connected to C104 routing switches. CERN also built the fibre version of the draft standard, to find that it gave excellent performance.

STMicroelectronics, with CERN and Bull (who introduced Gigabit/s links), were the foundation for an EU project to turn the internal INMOS standard into an IEEE standard, IEEE 1355. There were a number of Working Group meetings to discuss the emerging IEEE 1355 standard, and an association formed to promote it, with members including STMicroelectronics (ST), Bull, CERN, Dornier SatellitenSysteme (DSS, subsequently Astrium Munich), and the author, who had formed 4Links to serve and promote the new standard.
SETBACKS FOLLOWED BY THE BIRTH OF SPACEWIRE

The EU project successfully generated the IEEE 1355 standard. Soon after its completion, ST decided to abandon the T9000, the C104, and everything to do with the 1355 standard. This was the correct decision at the time for the company but with no long-term component availability, most of the contributors to the standard then abandoned it also.

Although they did not take part in the standardization or promotion activities, British Aerospace at Filton had developed an FPGA implementation of the DS-Link. Just as ST abandoned the standard and components for it, British Aerospace abandoned this project, which led to the project leader leaving to join the University of Dundee.

ST abandoning the technology created an opportunity for 4Links, and British Aerospace abandoning their projects created an opportunity for the University of Dundee. There were just a few companies and organizations who saw the value in the new standard and would enable us to carry it forward. These included Canon from outside the space industry. They also included ESA, and DSS/Astrium, who licensed the DS-Link from ST to build the SMCS chip.

4Links developed a PCI-1355 board with the limited chips that were available and found a small number of customers in the space industry, particularly including ESA. 4Links also collaborated from 1995 with Dr B M (Barry) Cook, working in spare time from Keele University, to produce a CPLD implementation of the link. This was used by Satellite Services for their own design of PCI board, and gave outstanding results when tested at DSS/Astrium. Barry Cook was later seconded to 4Links from Keele and has since joined 4Links full time as our CTO.

A key supporter of the standard in the early days was Eonic Systems. They negotiated a distribution agreement with DSS/Astrium to sell the Mosaic-020 board which used the SMCS chip. To help promote the Mosaic board, Eonic organized a “Space Day” with presentations from many different companies, all related somehow to the 1355 standard and its use in space. In retrospect, this meeting at Brussels airport was effectively the “zeroth” meeting of the SpaceWire Working Group. It demonstrated that there was the critical mass required, and gave the participants the confidence to move forward.

THE SPACEWIRE STANDARD

The SpaceWire standard corrected errors such as the initialization malfunction, which unfortunately had made the 1355 standard and the SMCS chip no longer fully symmetrical. The standard introduced a Network Layer, with a simple definition of basic routing switch protocols. Late on in the standards process, it introduced Time-Codes, a very elegant way of distributing time throughout a SpaceWire network. Even with these new additions, however, many of the characteristics and principles that we’ve been following through this paper have carried through into SpaceWire, as shown in Table 2. One or two characteristics have not yet worked their way through to SpaceWire, and it will be interesting to see if they become part of SpaceWire at a later date.

<table>
<thead>
<tr>
<th>Table 2: Characteristics and principles that have evolved into SpaceWire</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Point-to-point</strong></td>
</tr>
<tr>
<td><strong>Symmetrical</strong></td>
</tr>
<tr>
<td><strong>Asynchronous</strong></td>
</tr>
<tr>
<td><strong>Flow-controlled</strong></td>
</tr>
<tr>
<td><strong>Modular</strong></td>
</tr>
<tr>
<td><strong>Scalable</strong></td>
</tr>
<tr>
<td><strong>Fault-tolerant</strong></td>
</tr>
<tr>
<td><strong>Serial</strong></td>
</tr>
<tr>
<td><strong>DMA engine</strong></td>
</tr>
<tr>
<td><strong>Packet protocol</strong></td>
</tr>
<tr>
<td><strong>Virtual Channels</strong></td>
</tr>
<tr>
<td><strong>Network protocol</strong></td>
</tr>
<tr>
<td><strong>Time distribution</strong></td>
</tr>
<tr>
<td><strong>Comms instructions</strong></td>
</tr>
</tbody>
</table>

There is much more that could be said to bring this history to the present day, but I’ll leave that to the other papers in this seminar. I hope that this paper shows the generic principles that are the foundation which makes SpaceWire so great. I’ve been privileged to have been involved with these principles now for thirty years, although they start even earlier and it took a long time for me to realize both the principles and their benefits. They now provide a bedrock of understanding that is available to any project that is willing to benefit from the experience.
ACKNOWLEDGEMENTS OF THE VISION

Perhaps the best summary of what SpaceWire owes to its history is this comment at an early SpaceWire meeting:

“If it had not existed, we’d have had to invent it!”

So many people have been involved in the history that it is impossible to record and thank them all. The key person without whom SpaceWire would not exist is Iann Barron, whose vision has inspired all that has followed. The original transputer links owe much to David May, Miles Chesney, Gerry Talbot, and Graham Stewart. The T9000 links owe much to Peter Thompson, Robert Simpson, Bob Krysiak and Jim Nicholas. The IEEE 1355 standard would not have existed without Colin Whitby-Streven, nor without the system work carried out by Bob Dobinson, Brian Martin and their team at CERN. Patrick Planeke was the first contact that 4Links had with ESA, and Tim Pike and Anja Christen from DSS/Astrium were in touch long before the SpaceWire Working Group was formed. Eric Verhulst was the drive behind the key meeting at Brussels airport. And of course the SpaceWire standard would not have been created without the dedication of Philippe Armbruster and Josep Rosello of ESA and Steve Parkes from Dundee, nor without the contributions and support of the rest of the Working Group. And 4Links would not have been able to support SpaceWire without Barry Cook and our colleagues, and without our customers — who include many of the authors of the papers you are about to hear.

ACKNOWLEDGEMENTS FOR THIS PAPER

The author also acknowledges and thanks the following sources of photos and other information used in this paper (where relevant, URLs are given adjacent to the photos):

Figs. 2, 3, 5, the 1977 quotation, and information about Modular One and the transputer, supplied by Iann Barron
Fig. 4: Birmingham University Computer Science Department web site, referred to by Dave Miles
Fig. 6: supplied by Anneke van der Geest and Leny Ouweland of ESTEC, contacted by Dave Miles
Fig. 7: From INMOS T424 Brochure, in author’s possession
Fig. 8: From INMOS.com web site, supplied to the site by Mike Thomas
Fig. 9a: SOHO satellite from ESA web site, referred to by James Barrington Brown
Fig. 9b: Photo of Sun, Naval Research Laboratory (NRL) web site
Fig. 10: Photo of SSTL transputer board, and table 1: supplied by Alex da Silva Curiciel of SSTL
Fig. 11: Transputer Module (TRAM) in sardine can: Copy for INMOS advertisement, in author’s possession
Fig. 12: Ethernet TRAM: Sundance product from Ram Meenakshisundaram’s web site
Fig. 13: Mac Carrier board for TRAMs: Levcor product, photo from Ram Meenakshisundaram’s web site
Fig. 14: T9000 from INMOS photo of T9000, in author’s possession
Fig. 15: From INMOS.com web site, supplied to the site by Peter Thompson