Towards 100G Packet Processing: Challenges and Technologies
Christian Hermsmeyer, Haoyu Song, Ralph Schlenk, Riccardo Gemelli, and Stephan Bunse

Driven by media-rich and bandwidth-intensive Internet applications, 100 Gigabit Ethernet embodies the next logical and necessary line speed following 10G/40G, although enormous challenges exist. Can today’s device technologies scale, or will new, disruptive approaches be required to overcome throughput and power density limitations? The evolution of field programmable gate array (FPGA) and application-specific integrated circuit (ASIC) technology was analyzed concerning speed, density, power, and pin interfacing. While clock frequencies cannot keep up with the higher interface speeds, massive parallelism and deep pipelining are needed to scale the throughput. The state-of-the-art architectures and algorithms for every aspect of packet processing are described. In addition, we look at alternative memory concepts and cover some emerging technologies: asynchronous FPGAs as a means for boosting the system clock and serial interfaces reducing the pin count between devices. Thereby, economic considerations limit the choice between the options. We conclude that although significant effort is necessary in terms of device and board technology, economic 100G networking is viable. © 2009 Alcatel-Lucent.

Introduction

The traffic that fills future networks is estimated to grow roughly along the lines of Moore’s law [12, 53] in the coming years. This traffic growth is fueled by faster access technologies, a growing number of consumers, the use of versatile, high-quality Internet Protocol (IP) applications, and improved personal content generation capabilities, such as high-resolution digital cameras and streaming devices. As a matter of fact, peer-to-peer applications like file sharing are dominating today’s Internet traffic. Already, a trend can be seen with some of the key service and application providers like Google*/YouTube*, Yahoo!*, Amazon*, and others towards using 10 Gigabit Ethernet (GbE) interfaces on their server stacks, asking for 100 GbE uplinks towards central switches and for multiple 100 GbE interfaces towards the point of presence. Similar trends exist in the supercomputing and medical engineering space, and in storage area networks (SANs).

While in the server and computing space the debate is still on whether to deploy upcoming 40 GbE technology in network interface card (NIC) functions, providers of transport networks are expected to be much more reluctant to perform a costly knight’s move from current 10G technology to 40G as an intermediate step, and finally to 100G. On top of the
**Panel 1. Abbreviations, Acronyms, and Terms**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIC</td>
<td>Application-specific integrated circuit</td>
</tr>
<tr>
<td>ASSP</td>
<td>Application-specific standard product</td>
</tr>
<tr>
<td>ATCA</td>
<td>Advanced telecommunications computing architecture</td>
</tr>
<tr>
<td>CAM</td>
<td>Content addressable memory</td>
</tr>
<tr>
<td>CAUI</td>
<td>100G attachment unit interface</td>
</tr>
<tr>
<td>CICQ</td>
<td>Combined input crosspoint queuing</td>
</tr>
<tr>
<td>CIOQ</td>
<td>Combined input-output queuing</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary metal-oxide semiconductor</td>
</tr>
<tr>
<td>CoS</td>
<td>Class of service</td>
</tr>
<tr>
<td>CP</td>
<td>Control plane</td>
</tr>
<tr>
<td>DDR3</td>
<td>Double data rate SDRAM 3</td>
</tr>
<tr>
<td>DPI</td>
<td>Deep packet inspection</td>
</tr>
<tr>
<td>DP-QPSK</td>
<td>Dual polarization-QPSK</td>
</tr>
<tr>
<td>DPSK</td>
<td>Differential phase shift keying</td>
</tr>
<tr>
<td>DRAM</td>
<td>Dynamic random access memory</td>
</tr>
<tr>
<td>EIR</td>
<td>Excess information rate</td>
</tr>
<tr>
<td>FA</td>
<td>Fabric adapter</td>
</tr>
<tr>
<td>FB-DIMM</td>
<td>Fully-buffered dual inline memory module</td>
</tr>
<tr>
<td>FCRAM</td>
<td>Fast cycle RAM</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field programmable gate array</td>
</tr>
<tr>
<td>FR-4</td>
<td>Flame resistant 4</td>
</tr>
<tr>
<td>FSB</td>
<td>Front side bus</td>
</tr>
<tr>
<td>GbE</td>
<td>Gigabit Ethernet</td>
</tr>
<tr>
<td>GPP</td>
<td>General purpose processor</td>
</tr>
<tr>
<td>HOL</td>
<td>Head of line (blocking)</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated circuit</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute for Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/output</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPv4</td>
<td>Internet Protocol version 4</td>
</tr>
<tr>
<td>IPv6</td>
<td>Internet Protocol version 6</td>
</tr>
<tr>
<td>IQ</td>
<td>Input queuing</td>
</tr>
<tr>
<td>ITU-T</td>
<td>International Telecommunication Union, Telecommunication Standardization Sector</td>
</tr>
<tr>
<td>ITRS</td>
<td>International technology roadmap for semiconductors</td>
</tr>
<tr>
<td>LA-1</td>
<td>Look-aside-1 (interface)</td>
</tr>
<tr>
<td>LAG</td>
<td>Link Aggregation Protocol</td>
</tr>
<tr>
<td>LAN</td>
<td>Local area network</td>
</tr>
<tr>
<td>MAC</td>
<td>Media access control</td>
</tr>
<tr>
<td>MIPS</td>
<td>Microprocessor without interlocked pipeline stages</td>
</tr>
<tr>
<td>MLD</td>
<td>Multi-Lane Distribution</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multiprotocol label switching</td>
</tr>
<tr>
<td>MPLS-TP</td>
<td>MPLS-transport profile</td>
</tr>
<tr>
<td>Mpps</td>
<td>Millions of packets per second</td>
</tr>
<tr>
<td>NIC</td>
<td>Network interface card</td>
</tr>
<tr>
<td>NP</td>
<td>Network processor</td>
</tr>
<tr>
<td>NPU</td>
<td>Network processing unit</td>
</tr>
<tr>
<td>NRE</td>
<td>Non-recurring engineering (costs)</td>
</tr>
<tr>
<td>NSE</td>
<td>Network search engine</td>
</tr>
<tr>
<td>ODU4</td>
<td>Optical data unit level 4 (100G line rate)</td>
</tr>
<tr>
<td>OIF</td>
<td>Optical Internetworking Forum</td>
</tr>
<tr>
<td>OQ</td>
<td>Output queuing</td>
</tr>
<tr>
<td>OTN</td>
<td>Optical transport network</td>
</tr>
<tr>
<td>OTU4</td>
<td>Optical transport unit level 4 (100G line rate)</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed circuit board</td>
</tr>
<tr>
<td>PCS</td>
<td>Physical coding sublayer</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical (layer)</td>
</tr>
<tr>
<td>QDR</td>
<td>Quad data rate</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of service</td>
</tr>
<tr>
<td>QPSK</td>
<td>Quadrature phase-shift keying</td>
</tr>
<tr>
<td>RAM</td>
<td>Random access memory</td>
</tr>
<tr>
<td>RED</td>
<td>Random early detection</td>
</tr>
<tr>
<td>RISC</td>
<td>Reduced instruction set computing</td>
</tr>
<tr>
<td>RLDRAM</td>
<td>Reduced latency DRAM</td>
</tr>
<tr>
<td>RS</td>
<td>Reconciliation sublayer</td>
</tr>
<tr>
<td>SAN</td>
<td>Storage area network</td>
</tr>
<tr>
<td>SERDES</td>
<td>Serializer/deserializer</td>
</tr>
<tr>
<td>SDH</td>
<td>Synchronous digital hierarchy</td>
</tr>
<tr>
<td>SDRAM</td>
<td>Synchronous DRAM</td>
</tr>
<tr>
<td>SF</td>
<td>Switch fabric</td>
</tr>
<tr>
<td>SFI-S</td>
<td>Scalable SERDES framer interface</td>
</tr>
<tr>
<td>SPI-S</td>
<td>Scalable system packet interface</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static random access memory</td>
</tr>
<tr>
<td>TCAM</td>
<td>Ternary CAM</td>
</tr>
<tr>
<td>TDM</td>
<td>Time division multiplexing</td>
</tr>
<tr>
<td>TM</td>
<td>Traffic manager</td>
</tr>
<tr>
<td>T-MPLS</td>
<td>Transport MPLS</td>
</tr>
<tr>
<td>TSI</td>
<td>Time slot interchanger</td>
</tr>
<tr>
<td>VLAN</td>
<td>Virtual LAN</td>
</tr>
<tr>
<td>VOQ</td>
<td>Virtual output queuing</td>
</tr>
<tr>
<td>XDR</td>
<td>Extreme data rate</td>
</tr>
</tbody>
</table>
financial challenges implied by a two-stage investment strategy, the present view on modulation formats indicates that advantages exist in the “slower” symbol transmission rate chosen for 100 Gb/s serial optical transmission (e.g., ~28 Gbaud for dual polarization-quadrature phase-shift keying; DP-QPSK) over 40 Gb/s, e.g., ~43 Gbaud differential phase shift keying (DPSK), when looking at spectral efficiency, and compatibility with the existing fiber topologies.

Consequently, network operators have begun to push for availability of 100 GbE technology right away. Some have expressed their intentions to scale their transport networks over the next few years by using a 100 GbE interconnect between transport equipment (shown at right in Figure 1). For long haul transmission, operators are using the optical transport network (OTN) [21] technology, and these signals will be carried inside an optical data unit level 4 (ODU4) frame.

We believe, however, that the first implementations of 100 GbE technology will be within the data interfaces of large routers, in the form of an interconnect with central offices and/or specialized data centers. Another example, shown in the left portion of Figure 1, is the Internet Protocol/Multiprotocol Label Switching (IP/MPLS) backbone, connected over MPLS transport profile (MPLS-TP) [4, 20] infrastructure, in combination with OTN and/or the synchronous digital hierarchy (SDH) switching and grooming technology. This is mainly driven by capital and operational spending reductions on the port cards, compared with multiple, lower-speed interfaces, and by gains from statistical multiplexing. Core transport switches and large multi-service switches can separate the IP/MPLS traffic based on a tunnel identifier or on VLAN tags and distribute suitably sized traffic streams to appropriate destinations across lower-speed interconnect networks. When routers and transport network elements are co-located, native 100 GbE LAN physical (PHY) technology is an option for inexpensive interconnection.

Figure 1.
Applications for 100 GbE: Router interface optimization and backbone dimensioning.
As a mediation strategy until 100 GbE technology becomes available, aggregation of multiple 10 GbE interfaces with the Link Aggregation Protocol (LAG) [13] can solve the capacity demand to some extent, but issues exist with operational complexity and transport efficiency, as well as the inefficient use of limited port space in systems and hence cost problems.

The demand for 100 GbE technology has spawned a number of standardization activities, research efforts, and industry initiatives since 2005. It is not only the optical transmission interface that requires a 10-fold increase in capacity, compared with current 10 GbE interfaces. In fact, most of the parts that construct today’s systems suffer from scalability limitations. In switching and routing systems, backplane capacity and the slot access capacity are easily exceeded, and so are the board space of the input/output (I/O) units and the budget for thermal power dissipation and power supply. This leads to higher integration of functions in silicon, and eventually also to novel cooling concepts. The speed over backplane traces and between individual components needs to be increased, creating additional challenges for connectors and printed-wire board material. The data path is loaded in each direction with packet rates of up to 150 million packets per second (Mpps) on a 100 GbE port. Therefore, packet processing components and traffic management components will have to provide increased interconnect speed and will be charged with higher memory throughput, wider internal parallel structures, higher clock rates, and more logic gates to provide sustained packet processing at wire speed.

Addressing those challenges at an early stage will help in preparing the data and transport systems to leverage 40 GbE and 100 GbE interfaces as clients to their networks. Standardization is playing a major role in assuring commonality of standard components like framers and optical modules, and of compatibility of signal rates between, e.g., the Ethernet signals, and the OTN hierarchy.

We will not discuss changes of the required packet processing and traffic management functionality or changes in the control plane. Notably, early implementations might be purpose-built and provide for a limited subset that is feasible within the boundaries of available technology.

In the first section, we will walk through the functions that make up a 100 GbE packet switching subsystem, and we identify the prevalent limitations and architectural challenges and discuss means to resolve those issues.

In the technology section, we will provide insights into the major technical trends for 100G packet processing and switching, and considerations on how new and disruptive technologies could help to overcome the challenges of early implementations.

The goals of this paper are to identify the need for advances in many aspects of system design and to demonstrate ways to fill those needs with new methods and designs that either are already available or will be available soon.

**Challenges in 100 GbE Subsystems**

Subsystems for 100 GbE may be utilized in large routers or switches, core transport systems, and, later on, in high-capacity servers. Though many differences will exist between them, like the amount of memory used for various purposes, there are also many commonalities in the construction of the data path and the side band processing and interfaces, and hence a generic architecture can be established in order to assess scalability limits and bottlenecks for the throughput.

**Subsystem Taxonomy**

Figure 2 shows the generic architecture of a 100 GbE packet input/output subsystem as it is attached to a central switch fabric. The paper focuses on the components that make up the data path (packet processing, scheduling, and switching functions; drawn in color), with transport layer functions shown for completeness only (optics, PHY, media access control (MAC); drawn in shaded gray), and discusses interfaces between them.

The network processor (NP) performs packet classification, label generation/swapping, address learning/mapping, and policing. To this end, specific table entries are retrieved from external memory and network search engines. The NP interfaces with the control plane (CP) for the purpose of forwarding table management and control protocol processing. Based on the NP’s forwarding and marking decision, a traffic manager (TM) performs packet queuing and
de-queuing, shaping, and scheduling and resides on ingress and/or on egress, depending on the chosen switch architecture. The fabric adapter (FA) schedules the packets across the central switch fabric. Interconnection between FA and switch fabric is realized via a backplane using high-speed serializer/deserializer (SERDES) circuits.

Common interface standards greatly reduce cost and development time for the components of a 100 GbE ecosystem. Equipment manufacturers and component vendors are therefore obliged to coordinate the definition of line-side and fabric-side system interfaces, as well as memory interfaces, and the packet-based onboard transmission interfaces.

Currently, different organizations focus on particular aspects of the standardization: The Institute of Electrical and Electronics Engineers (IEEE) defines the 100G attachment unit interface (CAUI) and the multi-lane distribution (MLD) specifications for media access control devices [14]. At the same time, the Optical Internetworking Forum (OIF) pushes implementation agreements for framers based on the International Telecommunication Union, Telecommunication Standardization Sector (ITU-T) optical transport unit level 4 standard [21], such as the scalable SERDES framer interface (SFI-S). Although the comparatively lean SFI-S can be used both for OTN framers and LAN PHYs, the authors see a clear preference to use MLD for 100 GbE LAN applications instead: MLD’s virtual lane concept allows the de-skewing of several optical channels, a prerequisite for cost-effective optical 100 GbE modules. Both bit-stream based standards are used in functional layers below the reconciliation sublayer (RS), depicted in

---

**Figure 2.** High-level architecture of a 100 GbE switching subsystem.

---

**CAUI—100G attachment unit interface**
**CP—Control plane**
**EFEC—Enhanced forward error correction**
**FA—Fabric adapter**
**GbE—Gigabit Ethernet**
**MAC—Media access control**
**MLD—Multi-lane distribution**
**NP—Network processor**
**ODU4—Optical data unit level 4 (100G line rate)**
**ORX—Optical receiver**
**OTX—Optical transmitter**

**PCS—Physical coding sublayer**
**PHY—Physical layer**
**PMA—Physical medium attachment**
**PMD—Physical medium dependent**
**RS—Reconciliation sublayer**
**SERDES—Serializer/deserializer**
**SFI-S—Scalable SERDES framer interface**
**SPI-S—Scalable system packet interface**
**TM—Traffic manager**
**VO—Virtual output**
**VOQ—Virtual output queue**
the left portion of Figure 2. In upper layers, packet-based interfaces support multiple channels and flow-control mechanisms, predominant examples being Interlaken [17] and the scalable system packet interface (SPI-S) [36]. While Interlaken was announced early and is already supported by many devices, SPI-S is approved by the OIF, which includes more than 100 manufacturers. We expect that these two competing standards will continue to co-exist, which complicates interworking between components in the 100 GbE ecosystem.

**Medium Access Controller**

Strictly speaking, the MAC functionality is part of the Ethernet transport layer (as opposed to Ethernet switching layer this paper is dealing with). Also, we do not expect that the functional definition of the MAC will differ greatly from Gigabit and 10 Gigabit MAC specifications, as its main task is Ethernet frame checksum validation and generation. Sometimes integrated with other packet-aware functions, the physical allocation of the MAC and its associated physical coding sublayer (PCS) function also affect the definition of the network processor and/or the optical front end/framer functions. Since it is critical for the subsystem architecture, the cost structure, and the feasibility of individual components, this physical allocation is critical in the context of 100 GbE.

While 10 GbE implementations integrate the physical coding sub-layer with the optical module, the functional split for 100 GbE will most probably be different: shifting the PCS function from the optical module towards the MAC allows for simpler optical modules. For 100 GbE LAN PHY implementations, the Multi-Lane Distribution Protocol is intended to connect MAC/PCS with the optical module. In turn, 10 Gb/s transceiver technology is required even in first generation 100 GbE MAC/PCS implementations, because MLD is typically realized with 10×10 Gb/s physical links. The same holds for 100 GbE line cards with an OTN front end, which depend on an OTU4 framer instead of 100 GbE LAN PHY (a possible functional layering is depicted in the inset of Figure 2): If the OTU4 framer is integrated with the MAC/PCS, the interface towards the optical module could be SFI-S, which also requires 10 Gb/s links.

**Network Processing**

Network processing comprises the functions of packet parsing (extracting bits from the incoming packet to generate a key), classification (determining a flow out of the key), editing (adding or changing bits of the packet), and metering/marking of packets for subsequent quality of service (QoS) handling in a traffic management function. Considering 100 GbE interfaces with a packet rate of up to 150 Mpps, wire speed processing results in a packet arrival event every 6.7 ns.

Design of specialized code blocks for these functions, e.g., in a hard-coded application-specific integrated circuit (ASIC), provides for a relatively fast implementation, though this strategy cannot deliver the flexibility needed whenever protocol changes, bug fixes, or new operator demands have to be adopted quickly. For 100G packet processing, this approach will not be discussed further. Instead, packet processing is usually performed with processor kernels (that may again be implemented in field programmable gate arrays (FPGAs) or ASICs). A number of alternative approaches are shown in Figure 3.

The upper part of the figure shows the different architectures. In a first step, one can distinguish general-purpose processors (GPPs) from network processors. GPPs are not tailored to any specific computational task and thus are widely used in personal computers and servers. NPs, on the other side, are especially designed for packet processing; they provide for more I/O bandwidth and avoid the enormous power dissipation of GPPs.

Before discussing network processors, we will have a closer look at GPPs. With their unmatched flexibility, they can profit from an industry-wide deployment and from many established vendors. GPPs focus on high processing power, while high I/O-throughput, e.g., for massive data transmission, is of lower priority. Another drawback is the demand for code compatibility over decades. This is achieved by separating the top architecture (which is the architecture visible and used by the assembler programmer) from the micro architecture, which may evolve from one processor generation to another. All top-level assembler code needs to be interpreted in real time and is translated into code for the micro architecture.
This results in high power consumption and high transistor count.

In an evaluation, a GPP implementation that comes close to simplex 100G wire-speed packet processing and traffic management can be realized with four of today’s Intel® Xeon® processors (e.g., X5482) [16] and a commercially available memory hub interconnecting the Xeons’ front side buses (FSBs) with the memory. Each processor can reach 150 watts of power dissipation, with additional 47 watts from the memory hub, not accounting for the memory. The number of devices and their power consumption are completely beyond all reasonable limits. The advanced telecommunications computing architecture (ATCA) shelf specification, as a benchmark, limits power dissipation per slot to 200 watts [40]. It is also evident that an ATCA blade with the dimensions of 322 mm × 280 mm cannot even host a simplex processing direction that follows this architecture. The authors evaluate an approach using GPPs to be reasonable only in applications with moderate bit rates, and when the GPP architecture provides additional benefit for other computational tasks.

In contrast to GPPs, network processors are especially tailored to the needs of packet processing. As detailed in Figure 3, they have dedicated packet transmission interfaces with high bandwidth like SPI or Interlaken, and power dissipation is considered as a limiting factor, clearly higher weighted than a concept to reuse older program code. In general, three types of network processor architectures can be distinguished: parallel, partly pipelined, and fully pipelined.

The fully parallel architecture, sometimes referred to as a “sea of RISCs,” consists of several processor kernels, each able to perform all functions, such as parsing, classification, and packet update. This architecture is fairly flexible and easily programmable. As an example, the Broadcom® BCM1480 device [5] uses such an architecture. In order to leverage a broad software base, the kernels can be standard reduced instruction set computing (RISC) kernels, such as the microprocessor without interlocked pipeline stages (MIPS) architecture. A disadvantage to this approach is that all processors require access to shared resources such as memory and I/O interfaces. Furthermore, a re-sequencing of data is necessary to avoid misordering due to varying processing times. The complete program code also needs to be accessible for all kernels. To mitigate the latter issue, a software designer may decide to operate the NP in a pipelined fashion, e.g., to generate a key in kernel 1 and then to send the
packet to kernel 2. As this would also limit the contention for external resources, the number of threads running in each kernel can be lowered. In general, such a network processor structure is more appropriate if functions beyond layer 2 and layer 3 handling are required, and if bit rates are moderate. Such a structure is not well suited for 100 GbE.

The second type of NP architecture is partly pipelined. Dedicated kernels exist for each major functional block of the network processor, and packets are shifted from one processor to the other. For example, there are specific kernels for parsing, classification, metering, and marking. The advantage is that external resources only need to be connected to those kernels that need them to perform their functions. A classification kernel requires access to the classification memory only. Additionally, a subset of the execution code needs to be stored at each kernel, so the classification cores require the classification program only, but no editing and no parsing instructions. It is also possible to optimize the instruction sets of the kernels.

Usually, simple pipelining does not provide for sufficient resources to achieve wire speed processing. In turn, each pipeline stage consists of several processors operating in parallel. As a consequence, the problem of resequencing remains. An issue with pipelining is that programming becomes harder. As the kernels have no common memory, heap and stack operations cannot cross the kernel boundaries. Information that is added to each packet must be kept at a bare minimum to avoid a blocking of the communication between the kernels. This all requires that software be carefully partitioned.

The third architectural option for network processors is the fully pipelined processor. While partly pipelined processors use only few pipelining stages, normally less than 10, these processors consist of a huge pipeline of kernels, sometimes more than 1,000. Each kernel processes only a few program instructions and then shifts the packet to the next kernel. This allows for a very effective hardware implementation and provides a processing power big enough to avoid further parallel operation. As re-sequencing is avoided, programming becomes more difficult: As discussed before, no heap or stack operations can cross kernel boundaries. With the constant shifting of data from one kernel to the next, the use of subroutines is rendered impossible.

At the bottom of Figure 3 it is indicated that each of the discussed architectures for network processors can be implemented using either FPGAs or ASICs. FPGAs allow for easy reconfigurability, while ASICs provide higher speed and consume less power. (See the section on technology for details.)

At present, the partly or fully pipelined network processor architectures are the only reasonable choices for handling 100 GbE. With further progress in silicon technology, fully parallel processors may also pass this hurdle, though it seems highly unlikely that GPPs will be used for 100 GbE anytime soon. Specialized hardware based on hard-coded functional blocks instead of processor kernels offers little flexibility and most likely will fall out of use.

Memory

Memory devices are used to store the lookup tables, the statistical counters, the queue lists, and the packets themselves (i.e., those used as the packet buffer). While the memory speed is always a concern, the relatively small storage requirements of lookup tables and the statistical counters allows the use of fast but low-density static random access memory (SRAM) devices (288 Mb 500 MHz QDR-III SRAM is on the roadmap of major SRAM vendors). On the other hand, the packet buffer requires a significant amount of storage, so it has to be carried in slow but high-density dynamic random access memory (DRAM) devices in the hope of using more pins to gain enough aggregated bandwidth. The gap between the memory speed and the packet processing throughput requirement continues to widen. While the network line speed doubles every 22 months [33, 43], during the same period the memory clock rate improves less than 20 percent [19]). In 100 GbE, packets need to be processed at a rate of potentially more than 150 Mpps and millions of flows can be present at the same time. Memory access poses as the major performance bottleneck of the whole system.

For instance, although the I/O clock of the double data rate SDRAM 3 (DDR3) synchronous DRAM (SDRAM) can be 800 MHz, the fundamental random access time (t\text{RC}) will still be 45 ns. However, in 100
GbE, packets can arrive every 6.7 ns, which is about seven times faster than the random access rate. Moreover, the size mismatch between the memory transaction cell and the packet can significantly impact the memory bandwidth efficiency (e.g., if a packet is just one byte longer than the cell size, it needs to be transferred in two cells, which halves the effective bandwidth). A simple analysis indicates that the worst-case effective bandwidth of a DDR3 SDRAM can be less than 40 percent of the raw bandwidth. With substantial optimizations of the memory controller, which significantly increase the access latency and control complexity, the effective bandwidth can be brought up to 80 percent. Taking the queue management memories into consideration, the total pin count of the packet buffer subsystem can easily exceed 1,000. The board real estate and the signal integrity both face challenges that have never been seen before.

Reduced latency DRAM (RLDRAM) can provide SRAM-like performance with DRAM density and cost. Unlike DDR3 SDRAM, RLDRAM is conceived and optimized for a data networking application where small data chunks are randomly accessed throughout the memory. It has short burst size, fast row cycle time, and low bus turnaround time. With eight memory banks and a simplified command set, an RLDRAM-II chip can roughly achieve 90 percent memory bandwidth efficiency with a much simpler controller (i.e., each bank can be accessed in one clock cycle and in a round-robin fashion, plus one bus turnaround cycle to switch read and write operations). The drawback is that its projected I/O clock rate (533 MHz or 600 MHz) is lower than that of the fastest DDR3 SDRAM (800 MHz), which means more RLDRAM-II chips might be needed to implement a 100G packet buffer. RLDRAM is also more costly than DDR3 SDRAM.

There are some other types of advanced memory devices available in market. The Rambus extreme data rate (XDR*) DRAM is not supported by majority NP/FPGA vendors yet because of licensing issues. It exhibits good performance only when the data chunk size is large, which is not the preferred method for networking applications. Fast-cycle random access memory (FCRAM*), which is similar in design to RLDRAM, cannot beat RLDRAM in most application scenarios [39].

Table I summarizes some key characteristics and typical values of the next-generation memory devices now on the market or scheduled to ship no later than 2009. They are listed in a roughly increasing order of cost and power consumption per memory cell.

In 100 GbE subsystems, we need to apply two techniques to close the gap between the memory speed and the packet processing speed: (1) use multiple memory chips to gain higher aggregated bandwidth; (2) use algorithm innovations to reduce the required memory accesses. Along these lines, there are two trends that are worthy of mention. First, more and more SRAMs are embedded on chip to provide extremely low access latency and high memory bandwidth. A few tens to a hundred megabits of SRAM can be integrated in an FPGA or ASIC (e.g., the Altera* Stratix* IV FPGA [2] contains 23 megabytes of embedded memory cells). Although still scarce,

<table>
<thead>
<tr>
<th>Memory type</th>
<th>Voltage</th>
<th>Data rate</th>
<th>Density</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR3 SDRAM</td>
<td>1.5 V</td>
<td>2.4 Gb/s/pin</td>
<td>2 Gb+</td>
<td>36 ns</td>
</tr>
<tr>
<td>RLDRAM II</td>
<td>1.5 V</td>
<td>1.066 Gb/s/pin</td>
<td>576 Mb+</td>
<td>15 ns</td>
</tr>
<tr>
<td>XDR2 DRAM</td>
<td>1.5 V</td>
<td>8 Gb/s/pin</td>
<td>512 Mb+</td>
<td>35 ns</td>
</tr>
<tr>
<td>QDR-III SRAM</td>
<td>1.2 V</td>
<td>1 Gb/s/pin</td>
<td>288 Mb+</td>
<td>&lt;5 ns</td>
</tr>
</tbody>
</table>

Table I. DRAM and SRAM comparison.

DDR3—Double data rate SDRAM
DRAM—Dynamic random access memory
QDR—Quad data rate
RLDRAM—Reduced latency DRAM

SDRAM—Synchronous DRAM
SRAM—Static random access memory
XDR†—Extreme data rate

†Trademark of Rambus Inc.
on-chip SRAM is increasingly being used for high-speed packet processing, and its efficient use is the subject of active research. An example is shown in the next section. Second, serial transceivers will be used to reduce the memory I/O pin count, allowing a higher degree of chip aggregation. This point is further discussed in the section on high-speed I/O circuits.

**Search Engines**

Network search engines (NSEs) offload the NP from complex search tasks such as address lookups, access control, and QoS classification. While for IP-routed networks this can result in technological challenges for a foreseeable time (due to the need for IPv4/IPv6 address look-ups in significantly large tables), as well as in severe feasibility problems for many commercial 100G packet products (due to the power, cost, and board space budget restrictions), those problems are less severe, or can be widely avoided for applications that operate on the tunnel identifier only (e.g., Transport-MPLS (T-MPLS)/MPLS-TP) in support of core packet transport networks, as described in the introduction and depicted in Figure 1.

The ternary content addressable memory (TCAM) is by far the most widely used type of NSE. While the search performance of TCAMs is growing from generation to generation and already supports one to two searches at 100 Gb/s, the interface towards the NP is becoming more and more of a bottleneck.

State-of-the-art TCAMs primarily use a parallel look-aside-1 (LA-1) interface [34], which offers limited performance [35]. IPv6 lookups, for example, require long search keys and cannot be supported at 100 Gb/s. Extending the width of the parallel interface is not a solution to increase performance, because the high pin count on NPs is already a huge problem. Therefore, serial interfaces for TCAMs are considered to be a viable path to 100 GbE networking. Standards for search co-processor serial interfaces have recently been approved [18, 37].

TCAMs are considered power-hungry devices because of their brute-force way to activate all the cells for each search. In turn, cooling effectiveness requires a luxury expenditure of costly board space. This makes them by no means a match for DRAM and even SRAM in terms of power consumption. In order to stay within the tight power budget of a 100 GbE system, TCAM alternatives could be used to conduct some search tasks, if viable.

For example, some NSE vendors have developed algorithm-based NSEs to support fast address lookups [15, 32]. With TCAM-like flexibility and performance, this also achieves significant cost and power reduction. Another approach is to employ on-chip memory-efficient Bloom filters to achieve 100 Gb/s IPv6 address lookups without using a TCAM [47]. Hardware-based hash tables with deterministic lookup performance can also efficiently solve a wide range of lookup problems currently handled by TCAMs, such as MAC table lookups [46]. As these techniques become more mature, they will play a more important role to reduce the overall power consumption by replacing TCAMs in many applications.

Wherever possible, reducing the breadth and the depth of the search directly results in less board space consumption, power dissipation, and less critical signal routing issues.

**Traffic Manager**

Traffic managers are responsible for managing the bandwidth of packet flows based on criteria such as guaranteed bandwidth, packet loss rate, latency, and jitter, in order to meet the quality of service requirements. The management mechanisms include metering, policing, shaping, queuing, and scheduling. These are the same for 100 GbE as for its low-speed predecessors, except for the fact that the packets need to be handled much faster and the number of flows can be much larger. As a result, the two major challenges the 100 GbE traffic manager faces are packet buffering and queue maintenance.

It has been evident from the previous discussion that the high throughput of the packet buffer can only be achieved by aggregating the bandwidth of multiple DRAM devices. However, the method in which these memory devices are organized has a significant impact on the packet buffer performance.

**Figure 4** offers a graphical representation of packet buffer architectures. The most straightforward method is to arrange memory devices in parallel to form a wide data bus, as shown in Figure 4a. Apart
from the signal integrity issue due to the high pin count, this method would generate unreasonably large cell sizes. A cell is defined as the minimum memory transaction unit (burst size $\times$ bus width). Usually the packets are segmented into cells first, and then these cells are stored in free buffer slots through a sequence of memory transactions. A partially filled cell wastes bandwidth as well as storage capacity. The waste is exaggerated by a cell size that is significantly larger than the minimum packet size. Unfortunately, the DDR3 SDRAM requires a large burst size ($\geq 4$) to maximize the bandwidth utilization. In combination with the wide data bus, a large cell size is set to lower the achievable buffer throughput. To compensate the bandwidth loss (which can be easily 50 percent), our only leverage is to use an even wider data bus, which in turn magnifies the memory inefficiency.

Another method is to group the memory devices to form multiple independent packet buffers, each with a narrower interface to allow small cell sizes, as shown in Figure 4b. Though it offers the best memory efficiency, this method has its own deficiencies: e.g., the need to balance the load of these packet buffers to get the desired throughput. This seemingly easy task is actually very difficult to achieve. While one can have some control on scheduling the packet writes, the packet reads are determined by the traffic manager and totally unpredictable from the packet buffer point of view. For an extended period of time, all memory read requests may be concentrated to only one packet buffer and thus overload it. Intelligent write-dispatch, which adapts to the read pattern, may help balance the load. For example, one can choose to write a packet into a buffer for which the fewest read requests are backlogged. In addition to the load balancing issue, temporary load fluctuation causes the variation of access latency. Hence a re-sequencing buffer is needed to handle the out-of-order read data. Packet-drop is also possible, due to cell loss when the buffer is full. Moreover, packets suffer longer access

Figure 4. Packet buffer architectures.
latency that must be bounded. All of these are open issues that demand further investigation. In general, the load balancing model for the packet buffer scales better than the parallel model in terms of bandwidth efficiency. We have designed a 100G packet buffer based on the load balancing model. RLDRAM II is used since it allows a very simple memory controller. While the memory accesses are slotted, the memory controller can schedule a slot to either write or read the memory. This enables the write bandwidth to be dynamically shared between reads and writes so the high-level load balancing is achieved.

Packets in the buffer are organized as logical queues. The queue descriptors are typically implemented as linked lists in off-chip SRAM devices. In 100 GbE, a packet can arrive or leave in less than 7 ns. A 500 MHz QDR-II SRAM allows memory access every 2 ns. Hence one en-queue operation plus one de-queue operation must finish within three memory accesses. This is impossible with the conventional linked list maintenance algorithms. If higher memory speed is not available, novel algorithms must be implemented. In [24], a buffer aggregation scheme with a coarse-grained scheduling algorithm is used to reduce the SRAM size and bandwidth needed, at the cost of some wasted DRAM space.

**Fabric Adapter and Switch Fabric**

100G data signals do not impose specific requirements on switch architectures, however, since systems with 100 GbE and OTU4 signals are generally large in total capacity, the selected switching architecture needs to offer appropriate scalability.

**Multi-service switch architecture.** Large core nodes and aggregation transport nodes are required to provide multi-service functionality as well as flexible switching granularity in support of time division multiplexing (TDM), packet, and OTN services [27]. Such hybrid systems are referred to as traffic-type agnostic. In order to support TDM and OTN services, small and predictable node transition latencies and virtually zero internal data loss are necessary; the cost for fulfillment of these requirements, in terms of speedup, is sensitive to the selected switching architecture paradigm. Potential architectural build-outs of such traffic-type agnostic systems are dual star/dual fabric, single star/dual fabric, packet-based agnostic fabric, and TDM-based agnostic fabric. Agnostic fabrics are fabrics where a single switching infrastructure comprising the fabric itself and the fabric access devices and their interconnections is able to switch both packets and TDM signals [27].

**Figure 5** provides details on both the dual star/dual fabric architecture and the single star/dual fabric architecture. In the dual star/dual fabric switch arrangement shown in Figure 5a, dedicated forwarding infrastructure is available for TDM and packet switching. Its main advantage is design simplicity, along with a clear boundary between the packet and TDM-processing entities, which allows for independent scalability. However, the duplication of switch cards and of the number of backplane traces is costly. In addition, the design complexity of the interconnecting backplane can turn out to be beyond feasibility for large systems. The single star/dual fabric switch arrangement solves the backplane complexity issue by sharing the traces and selecting the signal’s appropriate switch destination by means of an additional analog crossbar. In fact, the inefficient use of fabric resources common to these two architectures is being criticized because of the associated cost. For the reasons listed above, the use of dual star/dual fabric and single star/dual fabric architectures is restricted to low-capacity systems.

In traffic-type agnostic switch fabrics, a single and smaller switching facility, based on either TDM switching or packet/cell switching, is used to handle both types of client signals.

**Figure 6** compares packet-based and TDM-based agnostic switching infrastructure. TDM-based agnostic switch fabrics, shown in Figure 6a, are generally built upon a memory-based crossbar (time slot interchanger), and thus latency requirements (e.g., around 30 μs) and bit errors are adequately fulfilled, and TDM services and multicast are naturally supported. Packet transport requires an adaptation layer that bridges the granularity gap between payload datagrams (i.e., packets) and the timeslots that are cross-connected in the matrix. With appropriate matrix scheduling, the low demand for speedup makes this architecture very suitable for high-capacity systems.
Packet-based agnostic switch fabrics, shown in Figure 6b, are mostly based on self-routing fabrics [23] or buffered crossbars [44]. As in many current high-end routers, ingress traffic can be balanced across several parallel switching planes (the spatial equivalent of speedup). This concept leverages “graceful degradation” as an efficient equipment protection strategy.

Non-deterministic arrival times of the data stream on the egress side require re-ordering at egress, however. When switching asynchronous packet flows, packet-based agnostic fabrics behave like traditional packet switches. Conversely, TDM switching is obtained by performing circuit emulation at the equipment level: the TDM signal is chopped into fragments, then those fragments are switched and re-ordered on egress. Low cell loss probability requires a sensitive design of the buffered crossbar, as well as internal forwarding priority schemes. The speedup needed to emulate a TDM switch can become an issue; conversely, an arbitrary switching granularity can be easily realized without impacting switch fabric complexity.

The choice of packet-based versus TDM-based agnostic architectures depends on several factors, like the share between native services and emulated services.

For TDM-based agnostic systems, scaling is hampered from granularity: the finer the switching

---

**Figure 5.** 

PKT—Packet  
PRC—Primary reference clock  
TDM—Time division multiplexing
granularity selected, the lower the resulting single stage capacity of a switch entity will be. For packet-based agnostic systems, the emulation of zero-loss, as present in traditional TDM crossbars, is reflected in speedup demand, buffering cost, and latency.

**Buffering architecture and matrix scheduling for packet applications.** When implementing packet applications with an agnostic fabric, the buffering architecture chosen can neutralize the various bottlenecks that limit scalability: required matrix and port speedup, number of queues, amount of meta-information for matrix scheduling, and complexity of the centralized matrix scheduler (if present). An output queuing (OQ) switch, where cells are first switched and then buffered at egress, is the closest approximation of an ideal switch architecture. As outlined in [6] and [48], it provides minimum latency and maximizes throughput since contention cannot take place inside the switch. On the other hand, it requires the costly speedup of $N$ for a system with $N$ ports. In order to reduce the requirements for speedup, other approaches have been analyzed. Input queuing (IQ), which trades low speedup against high complexity ($O(N^2)$ for a system with $N$ ports) of centralized matrix scheduling, suffers from high latency and head of line (HOL) blocking, as described in the introduction of [6] and paragraph 3.1 of [30]. Virtual output queuing (VOQ) has been introduced to solve HOL blocking [50].

---

**Figure 6.**
Packet-based, agnostic and TDM-based, agnostic switching infrastructure.
Combined input-output queuing (CIOQ), together with VOQ (VOQ-CIOQ), has improved performance with output buffers and a modest fabric speedup (with respect to OQ) [6, 48]. Combined input crosspoint queuing (CICQ) with VOQ [44] leverages the buffering of crosspoints; this approach enables the use of distributed and hence more scalable scheduling. Reference to matrix scheduling algorithms can be found in [3, 30, 42, 49].

A possible implementation of a packet application with a TDM-based agnostic fabric uses an unbuffered crossbar matrix, combined with IQ and VOQ buffering schemes. A centralized scheduler, together with output arbiters, increases the throughput and limits the required speedup to a very modest amount. Without central scheduling, full system throughput cannot be achieved. Centralized scheduling, however, introduces serious scalability and complexity problems due to the amount of meta-information exchanged. Using VOQ for both matrix scheduling and traffic scheduling is not viable due to the huge number of queues. For instance, given eight classes of service (CoS) and 4,000 VLANs per port on a 100G card with 10×10 G ports, 32 K × 10 VOQs per card would have to be scheduled.

A preferred implementation will use a VOQ scheme where the centralized scheduling is applied only to a limited set of queue aggregates (coarse granularity). Fine-grained output queues for service scheduling are transparently regulated by means of a distributed weighted random early detection (RED) algorithm. (See [11] for details on RED.) In contrast to a case where central scheduling only arbitrates the matrix traffic and in which data have to wait for transmission before being sent (to avoid contention), in the preferred implementation case, the excess information rate (EIR) portion of traffic can be “pushed” across the matrix to egress, and suitable backpressure signals are generated at egress towards ingress, where weighted traffic discard takes place according to a plurality of policies. This allows for sophisticated QoS management, while minimizing the central scheduler complexity. Finally, reduced latency and a fair allocation of excess information rate for best effort and regulated traffic can be achieved.

Combining fabric infrastructure with NP/TM packet functions—issues. Some application-specific standard product (ASSP) vendors offer switch fabric elements, fabric access devices, and fabric access/traffic management integrated devices. Network processing units (NPUs) exist as standalone devices or integrate a traffic manager as well. Merchant NPUs with integrated TM can greatly simplify board design complexity and reduce power consumption. They require, however, careful integration of the TM portion with matrix scheduling mechanisms. In general, the merchant TM holds the service output queues and has to be carefully married with the switch fabric (SF) or fabric adapter (FA), e.g., by use of selective backpressure. It is very difficult to fairly control EIR distribution beneath different services/sources, and full control of EIR can only be obtained by providing speedup between the egress TM and the fabric chipset, which usually turns out to be quite costly.

Conversely, in the case of proprietary designs, the TM and the NP function are hosted in different devices, because at present (for memory bandwidth limitations at 100 Gb/s) no integrated ingress/egress TM can be mapped on existing FPGAs. FPGAs are flexible in following an ever-evolving feature set. Integration of the TM with a (proprietary) fabric adapter is cost optimized for packet applications, while a separate FA and TM is cost optimized for TDM applications.

Technology

When increasing the rate of the fastest interface by a factor of 10, as is the case when going from 10 GbE based systems to 100 GbE, almost all of the parts of such a system suffer scalability limitations. Within the constraints of a typical transmission system with respect to power, heat dissipation budget, and board space, the design and the architecture of a 100 GbE subsystem can only be accomplished along the evolution of its constituent components. We will discuss the relevant ones below, such as silicon performance, high-speed interfaces in the data path and in the side band, as well as evolution of ASIC and FPGA technology.

Chip Technology

The biggest obstacle to 100 GbE packet processing is the availability of fast, complementary metal-oxide
semiconductor (CMOS) technology, supporting high logic density, extended chip-internal clock speed, and integrated memory technology.

Increased system bandwidth and limited power dissipation budgets require at least the 65 nm, if not the upcoming 45/40 nm, chip technology generation. However, 100G designs, which are at the technological edge concerning throughput and I/O performance, still cannot fully exploit the improvement in logic circuit density, which roughly doubles per CMOS process generation (“Moore’s law”) [41]. This is because chip-internal clock speeds ($f_i$), which are critical in 100 GbE NPs and TMs, generally scale only linearly with the reciprocal ratio of logic gate sizes ($1/a$), as compared to logic density, which scales with the square of the reciprocal ratio of logic gate sizes ($1/a^2$) [19].

Integrated circuit (IC) history as well as the international technology roadmap for semiconductors (ITRS) [19] show approximately a 40 percent increase in $f_i$ per technology generation. Unfortunately, the global clock speed $f_g$ (i.e., the speed of the clock that is applied externally to a functional block or the entire chip) only increases by about 15 percent in the same interval, because on-chip wire delays do not scale with the ratio of logic gate sizes, but stay rather constant [29]. Interconnect wire delays have become the predominant limitation of frequency improvements, as can already be observed in present 65 nm designs [31]. In particular, 100G relevant architectural elements, like crossbar switches that require long chip-internal wires with rapid interaction, do not scale well. One approach to hide the interconnect delays in 100G designs is a further parallelization of the data path. Relaxing the clock frequency can be achieved through a widening of the internal bus structure, e.g., from 256 bit wide (running at 400 MHz) towards 320 bit wide (running at 320 MHz), at the expense of real estate. A more radical approach is the elimination of global interconnects (and thus global clocking), which ultimately leads to asynchronous designs, a topic that is discussed in more detail below.

With non-recurring engineering (NRE) costs reaching several million U.S. dollars for complex application-specific integrated circuits, provident 100 GbE equipment vendors look for alternatives especially for first-generation and low-volume implementations. Field programmable gate arrays are an interesting option as they offer many advantages, including rapid prototyping, shorter time-to-market, and programming flexibility. Within the boundaries of internal resources, flexible design updates can reflect emerging features in a dynamic services landscape. Despite the leading-edge chip process technology [2], the major drawbacks of FPGAs, when compared to ASICs, are higher power consumption, lower density, and mediocre clock rate. Recent empirical measurements show that FPGAs consume more than five times the power of ASICs and are slower by a factor of more than 3 [25].

ASICs, known for their comparably low power consumption, high density, and high internal speed, have to be designed “right the first time.” Since 100G packet transport and switching standards are not yet settled, vendors are reluctant to invest in ASICs in order to avoid costly redesigns for potential changes. Therefore, we believe that ASICs will initially be deployed in proprietary functions, such as switch fabrics, where the routine processing rarely needs updates, and where processing speed is the first order consideration. As soon as the market picks up and volumes of 100 GbE-capable systems start to rise, standards will mature, and the economics of ASICs will dominate the packet processing platforms of the second generation of components.

Another aspect for which FPGAs are outperformed by ASIC technology is the available amount of internal memory. An ASIC in 65 nm technology can embed about four times the memory of a corresponding FPGA. In general, large packet buffers for packet processing and traffic management functions are a serious issue in terms of floor space, power consumption, and pin count. Recently, the stacked-die technology has been proposed for both ASICs and FPGAs; this technology allows for multiple dies to be stacked in the same package. At present no devices are available, but the technology is actively tested by silicon and IC vendors, and availability is a matter of a few years. This technology could have many beneficial effects on communication designs: integration of control memories—mainly static random access memories
(RAMs)—and thus availability of pin count for I/O and packet buffers, and integration of data arrays and packet buffers. This depends on integration of dynamic RAM, and overall power reduction.

Application-specific standard products, such as programmable NPs or highly configurable TMs, combine advantages from both the ASIC and FPGA worlds: Their re-configurability allows them to adapt to a certain set of application scenarios, providing high throughput at the same time. As opposed to ASICs, which are usually designed for one customer, they appeal to a wider market and can therefore benefit from lower pricing. Customers of ASSPs, however, face the problem of differentiating themselves from their competition.

**High-Speed I/O Circuits**

The need to transfer 100 Gb/s of data between components imposes enormous requirements on high-speed I/O circuits. Additional internal headers, which are used to convey meta-information between devices, require an overspeed of up to 50 percent on certain links, increasing the peak bandwidth to 150 Gb/s. Using state-of-the-art 6.25 Gb/s differential signaling, as many as 24 serial transceivers are needed per (150 Gb/s) interface. Because per-pin bandwidth scales like $f_i$ linearly with the reciprocal ratio of logic gate sizes, 10 Gb/s transceivers have become feasible for the latest ASIC technologies. Relative to 6.25 Gb/s transceivers, this will reduce I/O related power dissipation, which is a major contributor to the overall power consumption in today’s systems [8]. Calculations show that the ratio of I/O power to corepower will increase per chip technology generation with the reciprocal ratio of logic gate sizes [26].

With the introduction of 3.125 Gb/s serial transceivers in 2002 [55], FPGAs have principally picked up on ASIC I/O technology. However, a delay of two years between the availability of a SERDES macro for an ASIC and the support in FPGAs can be observed to date, as shown in **Figure 7**. This figure, which is based on data from previous Alcatel-Lucent printed circuit board designs, suggests that the widespread availability of 10 Gb/s transceivers in FPGAs—a crucial prerequisite for flexible 100 GbE line cards—can be expected soon.

![Figure 7. Evolution of serial transceiver speeds in ASICs and FPGAs.](image)

While high-speed data path interfaces have employed serial transceivers for almost a decade, memory interfaces still rely on parallel I/O technology. For 100G capable memory subsystems, this presents a real issue, as hundreds of pins have to be allocated to parallel memory interfaces on devices like NP and TM, leading to difficulties with timing closure in layouts that run above 600 MHz.

Using high-speed serial links on next-generation memory interfaces leads to an increase in bandwidth per pin. This reduces the number of pins and associated traces on the printed circuit board (PCB) and in turn results in smaller PCBs with fewer layers. Because both data and clock are embedded in the same differential signal pair, the routing on the PCB is further simplified (no setup/hold requirements). Naturally, there are also disadvantages, like the complexity due to the higher bit rate on the serial links and the associated difficulties in debugging them. Oscilloscopes are required instead of standard logic analyzers. Additionally, the SERDES transceivers cause higher latencies in the communication between the NP/TM and the memory. This is because of additional serializer/de-serializer and frame-alignment circuitry. However, this can be countered by burst data transfer, where the serial memory interface is at an advantage with its inherently faster speed and its bidirectional nature (i.e., no bus turnarounds).
Recent developments such as (serial) fully buffered dual inline memory modules (FB-DIMMs), serial high-speed SRAMs [28], and initiatives like the Serial Port Memory Technology (SPMT) Working Group [45] underline the need for serial I/Os. Serial interfaces for high-speed, low-latency memories used in the telecommunications industry will open the door to new board layout possibilities: Since those memories could be placed anywhere on the board and do not have to be in proximity to the ASIC/FPGA, memory daughterboards would be possible. This would allow for simpler memory upgrade and better thermal management.

**Board Architecture**

The increasing speed of serial transceivers also demands higher bit rates across PCBs. Regular board material like the widely used FR-4 does not offer the required loss factor (tan δ) and has too high a relative static permittivity (εᵣ) to be used at these rates. Hence, board traces at 10 Gb/s require careful simulations to ensure correct operation. Unfortunately, material designed to better support higher frequencies (like the Rogers-4350 or FR-35) is expensive and more difficult to handle. In order to reduce cost, a combination of low-speed (e.g., FR-4) board kernels is used to transport the power supply and low frequency layers, while the single- or double-side high frequency layers are used for signal transmission.

In order to limit signal distortions and crosstalk at such frequencies, all lines need to be operated differentially. Furthermore, pre-filtering on the transmit side and post-equalization on the receive side, which is available in most of today’s FPGAs and ASICs operating at these frequencies, can improve the situation. While current equalizers offer only limited achievable gain, applying digital signal processing will become feasible for next-generation technology.

**High-Performance Programmable Devices**

Bridging the performance gap between ASICs and FPGAs has been a goal since the introduction of FPGA-to-ASIC conversion technologies in the mid-1990s [38]. It has become even more relevant since the widespread availability of structured ASICs [56] in 2002. Structured ASICs combine a set of predefined metal layers (comprising “tiles” of logical functions such as gates and multiplexers) with a few specific, customizable metal layers. As a result, the performance and gate count of structured ASICs are—depending on the internal architecture—close to those of standard ASICs, with the advantage of greatly reduced NRE cost and design time.

In the last few years, we have observed a shift of focus in the industry from structured ASICs to reconfigurable architectures customized for specific applications, such as communication and networking [1, 7]. Common to these telecom-specific devices are high device speeds of more than 1 GHz, and a large number of high-speed interfaces, which makes them potential candidates for 100G packet processing. Some achieve ASIC-like performance with an architecture of pre-fabricated packet engines that can be flexibly connected via high-speed point-to-point interconnects [7]. These packet engines are able to extract data from packet headers and to perform packet editing. On-chip SRAM and CAM suggest the further integration of look-up and control tasks required in packet applications.

Given the multi-year head start of established NP vendors that recently announced 100G capable ASSPs [9, 54], these technologies will have to prove themselves in a very competitive market. Considering the currently limited feature set of high-performance programmable FPGAs, we expect that those will first be restricted to tasks that require comparably low complexity, such as a 100 GbE MAC.

**Asynchronous Processing**

Clock frequencies of current FPGAs are not following the increase in the baseband bit rate of optical line interfaces, because FPGA vendors usually optimize for complexity rather than speed. As a result, the internal data path needs to be widened (e.g., >300 bit for 100G) and more resources are consumed, leading to placement and routing problems. New approaches currently under investigation deviate from the former guiding principle of completely synchronous circuits by featuring asynchronous logic and pipelined interconnects.

Asynchronous data communication is already used in ASICs, but when looking at today’s FPGAs, they still rely on synchronous operation and a
timing-consistent clock distribution. Specialized 100G-capable FPGAs with asynchronous operation are currently under development [56]. Since they implement handshake mechanisms and do not rely on a global clock, the kernel design is far simpler and offers better margins for increasing internal throughput. Thus, faster operational speed (>1 GHz) can be expected [51]. As the logically correct operation of an asynchronous circuit does not depend on a clock net, the circuitry can be operated within a wide range of supply voltages and temperatures. In [10], the operation was verified for temperatures from –196 °C to 127 °C and supply voltages from 0.13 to 2.3 V. Although the wide operating range will hardly be applicable in practice (since the achievable throughput of the implemented test circuit also varies from 1.7 kHz to 1.15 GHz over power and temperature), this property, in effect, makes the design process simpler. In [52], it is reported that when implementing benchmark circuits, most run within 75 percent of the FPGA’s maximum throughput. Achieving this with synchronous FPGAs requires design effort significantly beyond the capabilities of generic place and route.

On the other hand, asynchronous designs are usually larger (factors of 2 . . . 6 are given in literature [51]). This is partly compensated, as there is no need for a complex clock infrastructure and the resulting smaller data buses can be routed with fewer resources. In total, the asynchronous technique shows a promising speed advantage, but it must prove its usability in practice. The availability of IP cores and the design tool support are decisive parameters for the success of this technology.

Conclusion and Outlook

Fueled by the explosive growth of the Internet driven by media-rich applications, 100 GbE has recently gained tremendous momentum as a key enabling technology to address the network bandwidth shortage issue. However, given the limitations of current technologies, 100 GbE packet processing and switching still cannot be efficiently implemented for power, board space, and complexity over the next few years. As of today, demonstrating 100 GbE as a proof of concept is possible, but further architecture optimizations and technology advances are necessary for 100 GbE to economically compete against 40 GbE and n × 10 GbE implementations. Forwarding traffic at layer 2 with technologies such as T-MPLS/MPLS-TP can alleviate packet processing, e.g., by avoiding IP address lookups, deep packet inspection (DPI), and session-based QoS, thus improving 100G product realization feasibility.

In this paper, we scrutinize every aspect of 100 Gb/s packet processing and switching, trying to identify the main technical challenges that hamper the development of 100 GbE systems and to provide our insights about the corresponding technical trends that can help remove the obstacles. The key observations are summarized as follows:

- The use of serial memory I/Os can efficiently reduce power consumption and real estate requirements, allowing a higher degree of integration (i.e., stacked die).
- Advanced algorithmic search engines using Bloom filters or other data structures can be deployed to replace TCAMs with commodity memories as a way to reduce power consumption without sacrificing performance.
- Load balanced memory controllers can achieve the high aggregated memory bandwidth required by the packet buffer with the minimum power consumption and pin count.
- 40 nm ASIC/FPGA technology is needed to integrate the duplex processing path in a single chip.
- Higher transceiver speeds (10 Gb/s for FPGAs, 20 Gb/s for ASICs) are needed for efficient chip and backplane interconnections.
- Asynchronous devices may prove to be critical to boost 100 GbE system throughput, and to achieve significant power reduction.

Although the 100 GbE standard will not be finalized before 2010, packet processing and switching systems with 100 Gb/s or higher line card capacity are already on the horizon. Being the first generation of such systems, they are often purpose-built with simplified architectures and a reduced feature set, targeting some specific applications such as the provider backbone, data center aggregation, and media distribution. The pre-standard status also gives FPGAs more competitive advantages over ASICs. As the technologies
become more mature and the standardization process is completed, full-fledged 100 GbE multi-service switching systems will soon be a reality. We conclude that although significant efforts are necessary in terms of device and board technologies, economic 100G networking is viable.

Acknowledgements

We acknowledge gratefully that part of this work has been supported by the German Ministry for Research and Education (BMBF) under the EUREKA project “100GET—100Gbit/s Carrier-Grade Ethernet Transport Technologies” (grant 01BP0720).

*Trademarks

Altera and Stratix are registered trademarks of Altera Corporation.
Amazon is a registered trademark of Amazon Technologies, Inc.
Broadcom is a registered trademark of Broadcom Corporation.
FRAM is a registered trademark of Fujitsu Microelectronics Limited Corporation.
Google and YouTube are registered trademarks of Google, Inc.
Intel and Xeon are registered trademarks of Intel Corporation.
RLDRAM is a registered trademark of Qimonda AG.
XDR is a trademark of Rambus Inc.
Yahoo! is a registered trademark of Yahoo! Inc.

References


(Manuscript approved March 2009)

CHRISTIAN HERMSMEYER is a distinguished member of technical staff in the Chief Technical Office of the Optics Division at Alcatel-Lucent in Nuremberg, Germany. Mr. Hermsmeyer started his career in hardware development for private ISDN PBX at Philips Kommunikations Industrie AG in Nuremberg, Germany, and Glasgow, Scotland. He continued in the ASIC department of Lucent Technologies, where he developed SONET/SDH and data devices. In the System Engineering department his work has spanned the areas of system architecture and feature definition for packet and transport integrating systems. Within CTO he currently leads the initiative on 100 Gb/s packet processing. He received his M.S. degree in electrical engineering from the University of Dortmund, Germany, and holds several patents for transmission and data networks. He is member of the Alcatel-Lucent Technical Academy.

HAOYU SONG is a member of technical staff in the Network Protocols and Systems Department of Alcatel-Lucent Bell Labs in Holmdel, New Jersey, in the United States. He received his D.Sc. and M.S. degrees in computer engineering from Washington University in St. Louis, Missouri, and B.E. degree in electronics engineering from Tsinghua University, Beijing, China. His research interests include algorithms and architectures for high-performance routers switches and network security systems.

RALPH SCHLENK is a member of technical staff in the Hardware Engineering Department of the Optics Division at Alcatel-Lucent in Nuremberg, Germany. He received his Dipl.-Ing. (M.Sc.) degree in electrical engineering from the University of Erlangen, Germany. His work at Alcatel-Lucent has spanned the areas of optical transmission simulation, prototyping of electronic and optical equalizers for long-haul transmission networks, embedded software development for packet switching, and system architecture definition. Mr. Schlenk has authored coauthored various technical conference papers and holds patents for optical transmission networks. He is a member of the Alcatel-Lucent Technical Academy.
RICCARDO GEMELLI is a member of technical staff in the Chief Technical Office of the Alcatel-Lucent Optics division in Vimercate (Milan), Italy. He holds an M.S. degree in communications and electrical engineering from Milan Polytechnic. Mr. Gemelli joined Italtel in Milan as an ASIC designer developing devices for ATM applications. Later, in the Italtel System Engineering Lab (Central R&D) he was involved with various levels of responsibility in MEDEA and IST European funded research projects. In the Italtel Switching division, he developed a broadband-remote access server (B-RAS) and a border gateway for IP telephony; both applications based on a proprietary NPU. His current work at Alcatel-Lucent is focused on the switching architecture of multi-service nodes and on the development of network optimization algorithms. Mr. Gemelli is author/co-author of patents and papers in communication, switching architectures, and performance evaluation areas.

STEPHAN BUNSE is a research engineer in the Packet Transport Networking Technologies Department at Alcatel-Lucent Bell Labs in Stuttgart, Germany. He received its Dipl. Ing. degree in electrical engineering from the University of Dortmund, Germany. His research interests include communication network architecture and transport network hardware.