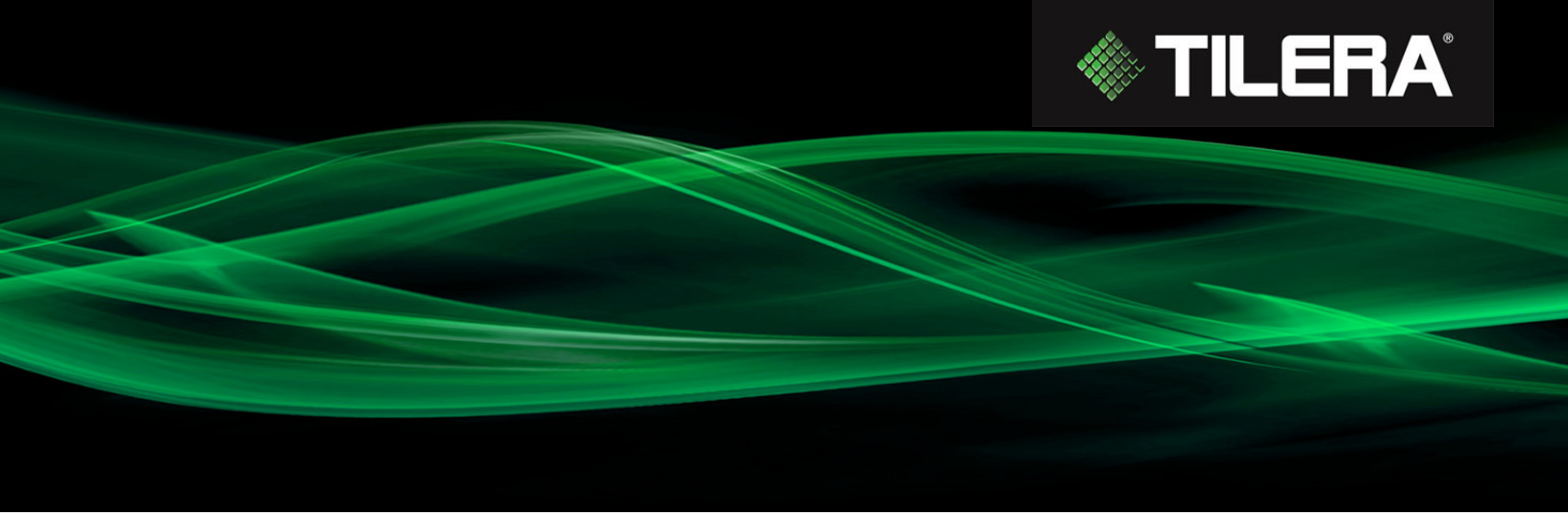

ARCHITECTURE AND PERFORMANCE OF THE TILE-GX PROCESSOR FAMILY

Matthew Mattina, CTO
Tilera Corporation

This paper describes the TILE-Gx™ processor architecture, discusses the design choices leading to performance and power-efficiency, and presents performance results on representative applications for the 72-core TILE-Gx72™, the flagship processor in Tilera's TILE-Gx family. This processor family is comprised of a series of high-performance, low-power 64-bit manycore processor SoCs. These highly integrated processors deliver industry-leading performance and performance-per-watt in the embedded networking, multimedia, and cloud datacenter markets.

1 Introduction

The past decade has seen a broad shift in focus away from single-threaded CPU performance and increasing clock frequency towards multicore architectures and throughput-oriented computing. This trend is present across the entire microprocessor industry, from quad-core smartphone system-on-chip (SoCs) to x86 multicore server processors, in order to combat increasing power density and diminishing returns from single- thread focused microarchitectural approaches. A second important trend is the emergence of high-speed packet and multimedia processing, as embodied by devices such as high-speed routers with video conferencing capabilities, next-generation firewalls, and multi-gigabit network intrusion and prevention systems. Commensurate with both of these trends, heterogeneous SoC designs containing general purpose cores and domain-specific acceleration hardware are now commonplace and provide programmable computation for a portion of the application along with hardware acceleration for specific tasks. This hardware acceleration—such as packet processing or security engines—can provide orders of magnitude speedups on specific subtasks versus software implementations, while consuming a fraction of the power [1].

Several important challenges arise from the interconnection of and programming model for multiple compute entities on the processor die. Cores and hardware accelerators need low latency, high-bandwidth communication under dynamic and unpredictable traffic patterns. The presence of on-die caches means that the same data can reside in multiple locations, resulting in data coherence issues, and on-die accelerators have given rise to inefficient, non-scalable programming models to move data between different hardware blocks (cores, accelerators, and I/O).

The TILE-Gx architecture addresses these challenges and delivers on the promise of high-performance, low-power multicore architecture by providing a large number of identical power-efficient general purpose cores running Linux and programmable in C, C++, Java, PHP, and many other standard languages. The design integrates high-speed I/O and domain specific acceleration hardware, interconnecting cores, accelerators, and I/O with high-performance cache-coherent mesh networks and a single shared-address space. The hardware accelerators and I/O are fully protected yet still directly accessible by user-space software, slashing system-induced latency.

Tilera® is currently shipping TILE-Gx processors with 9, 16, 36, and 72 cores in the embedded networking, multimedia, and cloud datacenter markets. The following sections discuss the TILE-Gx™ architecture in detail, explain several of the most important design choices, and present performance results for the 72-core TILE-Gx8072™.

2 TILE-Gx Processor Architecture

As shown in Figure 1 and Figure 2, TILE-Gx processors consist of a 2-dimensional array of identical “tiles”, multiple high-speed packet and PCIe interfaces, specialized accelerator blocks, and integrated DDR3 memory controllers. All on-chip communication occurs via multiple point-to-point intelligent Mesh (iMesh™) networks. Each tile consists of a 64-bit, 3-wide issue processor core with translation lookaside buffers (TLBs), L1 and L2 caches, and interfaces to the iMesh networks. The cores are ANSI C/C++ programmable, and run standard SMP Linux. The entire device—including the security, compression, and packet accelerator blocks—employs a flat shared address space with cache coherence maintained by hardware.

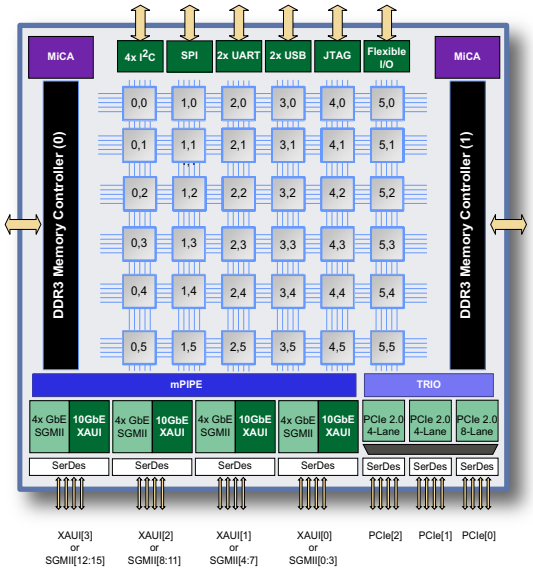

Figure 1. TILE-Gx8036 Block Diagram

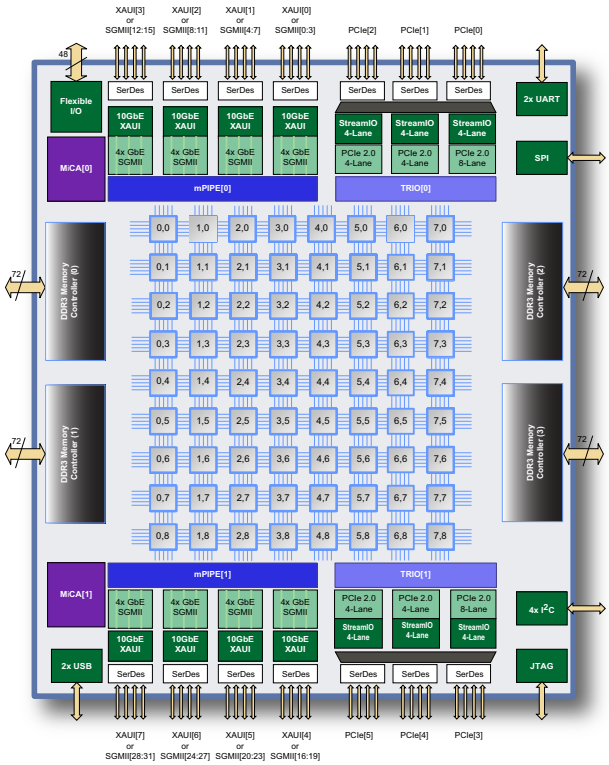

Figure 2. TILE-Gx8072 Block Diagram

The following properties are fundamental to the TILE-Gx architecture and are essential to achieving high performance and power efficiency on real-world applications:

- The TILE-Gx architecture is centered around *abundant general-purpose compute provided by the array of identical, programmable general purpose cores.* This enables standard programming models, preserving investment in software and programming skills.
- *High-performance and power efficiency comes from using a large number of efficient cores*, rather than a smaller number of larger, more complex and power-intensive cores.
- *High bandwidth iMesh interconnect and cache coherent shared memory across the whole architecture*—including I/O controllers and functional accelerators—provides both a familiar and productive programming model, and enables performance and power advantages through automatic on-chip caching. The cores' memory subsystem and the 2-D mesh interconnect work together to implement shared-memory coherence in a high-performance, power-efficient and scalable way [2].
- *Tightly integrated high-speed I/O and accelerators*, controlled via memory mapped I/O, and directly accessible from user space applications, deliver packet headers and payload directly to on-chip cache with very low latency. The accelerators include fully-programmable wire-speed packet classifiers and load balancers, cryptographic and security protocol offload, and compression/decompression engines.

The TILE-Gx architecture scales efficiently from smaller, lower-power devices to very high core-count, high-performance devices with 100's of cores. All devices in the family — from the 9-core TILE-Gx8009™ to the 72-core TILE-Gx8072 — are programmed in exactly the same fashion, enabling software reuse. The only programmer-visible variation is the size of the tile array and the specific mix of I/O and accelerators surrounding this array.

2.1 Tile Architecture

The tile array provides fully programmable general purpose processor cores. The TILE-Gx8072 device contains 72 tiles and delivers 260 billion 64-bit operations per second, while consuming less than 500 mW of power per tile. The "clean-sheet" core design is architected from the ground up with high performance and low power as guiding principles. The TILE-Gx instruction set architecture (ISA) uses a 64-bit instruction word to encode up to three RISC operations per cycle. The ISA provides a rich set of single-instruction, multiple-data (SIMD) operations that support packed 8x8-bit, 4x16-bit, and 2x32-bit operations such as multiply-and-add and sum-of-absolute-differences. At the nominal clock speed of 1.2 GHz, the device delivers over 345 billion 16-bit multiply-accumulates (MACs) per second and over 690 billion 8-bit MACs per second. Powerful atomic instructions, such as fetch-add-greater-than-zero, accelerate common shared memory constructs including producer/consumer FIFOs. The VLIW architecture and instruction encoding leads to simple, power-efficient and fast instruction decoders and an extremely short pipeline. The branch mispredict penalty is three cycles, and the load-to-use latency is two cycles. The low branch mispredict penalty obviates area and power-hungry dynamic branch predictors, and the short pipeline reduces the number of pipeline registers, further lowering area and power consumption.

The tile contains a 2-way set-associative 32 KB instruction cache and a 2-way set-associative 32 KB data cache. Both caches are backed by a unified 8-way, 256 KB second level cache with Single Error Correction Double Error Detection (SECDED) ECC protection. The second-level cache subsystem provides up to 8 outstanding cache line misses from each tile. This out-of-order memory subsys-

tem provides much of the performance benefit of out-of-order instruction execution at significantly lower power and area cost. Address translation and protection is performed by separate instruction and data fully-associative TLBs.

2.2 iMesh Interconnect Architecture

The iMesh interconnect provides low latency, high bandwidth communication to all on-chip components, including tiles and accelerators. Multiple point-to-point physical mesh networks are implemented, each dedicated to carrying a particular class of traffic and sized to provide the appropriate bandwidth for the traffic class. Table 1 lists the hardware-managed mesh networks, along with the bandwidth and class of traffic carried.

Table 1. iMesh Networks

Network Name	Traffic Class	Link Bandwidth	TILE-Gx8072 Bisection Bandwidth
SDN	Tile read and write miss requests to other tiles	19.2 GB/s	346 GB/s
QDN	Tile miss requests and write backs to main memory; cache invalidations	9.6 GB/s	173 GB/s
RDN	Cache line responses	16.8 GB/s	302 GB/s

Every tile contains an identical iMesh switch block that connects the tile to its immediate north, south, east and west neighbors. The tile's iMesh interface also connects to the L2 cache pipeline at every tile for sinking and sourcing traffic. The hardware acceleration blocks, I/O and memory controller blocks on the periphery of the tile array connect directly to the iMesh.[1]

The iMesh provides a logically all-to-all communication network, simplifying programming. Requested cache blocks are locally cached at the requesting tile and mesh latency is low, avoiding the need for the programmer to explicitly collocate process and data.

All networks are identical at the flow control level. Messages on the network are represented as packets and are divided into units the width of the network (called "flits"), plus a header flit to specify the route for the packet. The hop latency is a single cycle: packets are routed at the rate of one flit per cycle through the network. The route a packet takes is determined at the source and wormhole routing minimizes the link-level buffering requirements. iMesh packets move from source tile to the destination tile while traversing the minimum amount of wire. This leads directly to lower power and lower latency compared to ring or bus implementations.

The iMesh is a highly scalable interconnect. As shown in Figure 3, the total iMesh bandwidth *increases* as the tile array size grows, and the routing scheme eliminates any central hotspots. This enables future processors to increase the core count while maintaining the same interconnection architecture and programming model.

[1]Processors in the TILE-Gx family have different tile counts and hence different tile array dimensions.

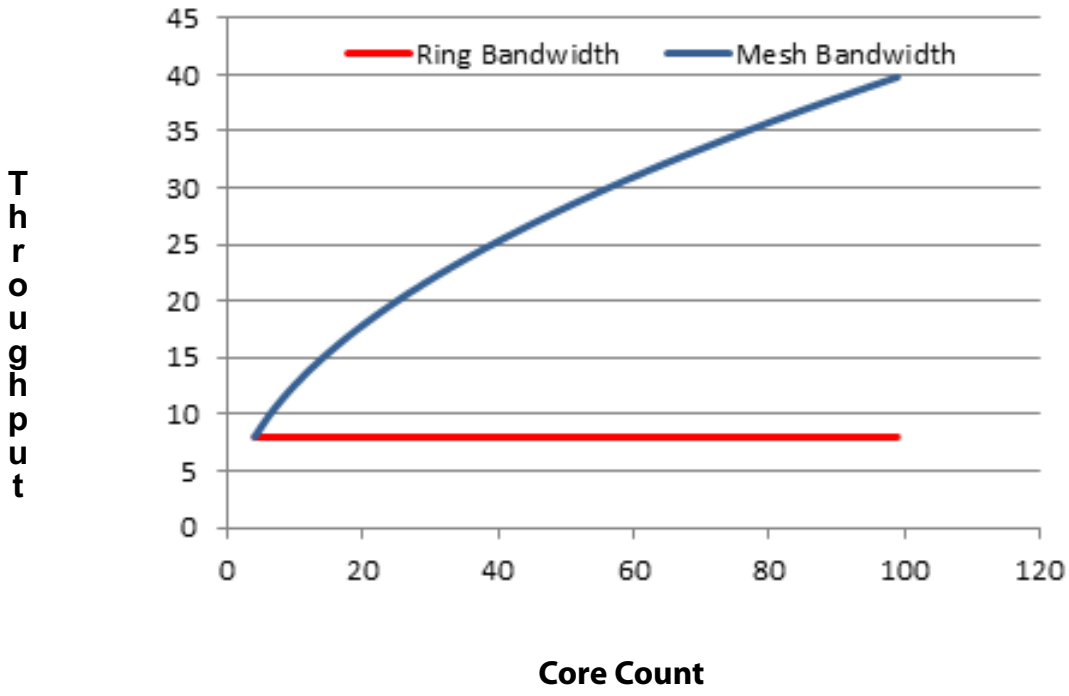

Figure 3. iMesh versus Bi-Directional Ring Bandwidth Scaling[2]

In addition to providing the overall transport mechanism for the SoC, the iMesh—together with the level-2 cache controllers—implements the global cache coherence protocol and Tilera's Dynamic Distributed Cache technology (DDC™). These two technologies enable high-performance cache coherent shared-memory, allowing any tile to request and locally cache any 64-byte memory cache line in the system. The hardware keeps track of which tiles have requested which cache blocks and ensures that the up-to-date copy is always delivered on reads.

2.3 Accelerator Architecture

The TILE-Gx processor family combines the general-purpose compute tile array with domain-specific acceleration engines for packet processing, security, and compression. All accelerators interface to the rest of the SoC via the iMesh, have direct access to the on-chip caches, share the same global address space, and participate in the cache coherence protocol. Threads running on the tiles communicate with the on-chip accelerators using memory mapped I/O. Protection and virtualization is provided in hardware by standard TLB translation mechanisms and dedicated hardware at each accelerator, allowing the accelerators to be shared by different applications (and virtual machines) and accessed from user space without requiring expensive system software intervention and locking.

2 Total available bisection bandwidth, assuming the same number of wires per link.

TILE-Gx provides dedicated "off-core" accelerator blocks, as opposed to in-core coprocessors. This architecture delivers the following advantages compared to traditional in-core acceleration architectures:

- *Improved core utilization*- By decoupling the acceleration hardware from the core, the core execution resources are freed from monitoring and managing the acceleration hardware. Off-core acceleration hardware can be configured to interrupt the core when work is completed, thereby allowing the core to do other work while the accelerator is processing the request. The TILE-Gx architecture supports the delivery of interrupts directly to user code, which further reduces interrupt latency.
- *Independent scaling of general compute performance and accelerator performance*- Devices in the TILE-Gx family can scale computing power and acceleration independently.
- *Better "fat pipe" throughput*- Significant acceleration performance on a *single flow* is needed for "fat pipe" workloads, while multiple well-balanced flows have lower per-core requirements. In-core architectures are forced into a tradeoff between ignoring the "fat-pipe" use case or over-provisioning the in-core acceleration hardware *in every core* to accommodate fat-pipe flows. In contrast, the TILE-Gx off-core architecture delivers high throughput for both "fat-pipe" AND balanced flows without burdening every core with accelerator hardware.

The following three subsections describe the packet processing, cryptographic, and compression accelerators in more detail.

2.3.1 Packet Processing Accelerator

The multicore Programmable Intelligent Packet Engine (mPIPE™) delivers wire-speed packet parsing, classification, buffer management, checksum validation/generation, and load balancing, completely off-loading these tasks from the tile array. The classification engines contained in the mPIPE are programmable, enabling flexible and custom parsing and classification of packets.

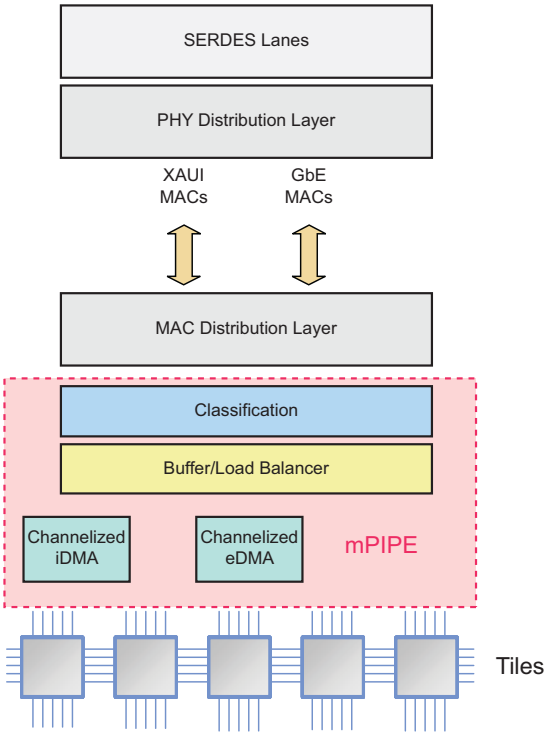

Figure 4. mPIPE Block Diagram

In the TILE-Gx8072, the 8 on-chip XAUI MACs and 32 on-chip 1 GbE MACs feed packets into 2 mPIPE subsystems where packet headers are parsed and classified at up to 120Mpps, according to a program running on the classification engines. The programmable nature of the classifier provides significant flexibility beyond simple 5-tuple hashing to assign packets to flows, buffers, and worker threads. Multiple load balancing modes including dynamic and static flow affinitization determine which of 256 ingress rings will receive a given packet. Dedicated mPIPE hardware validates IP header and TCP/IP payload checksums on packet ingress and generates checksums on packet egress. Ingress DMA engines using local I/O TLBs deliver the packet data, worker notifications, and meta-data directly to the tile array cache and to user-level code. On packet egress, 24 independent rings in each mPIPE provide QoS and flow isolation. Per-flow and global sequence numbering and egress DMA descriptor reassembly provide support for packet reordering. In addition, high-resolution ingress packet time-stamping and IEEE1588v2 precision time protocol is supported.

The combination of the mPIPE, integrated Ethernet I/Os, and programmable cores enable scale-out Ethernet fabric configurations with no additional Ethernet switch hardware. The hardware also supports IEEE 802.1Qbb priority-based flow control to ensure zero loss under congestion in datacenter networks.

2.3.2 Cryptographic Accelerators

The MiCA™ (Multicore iMesh Coprocessing Accelerator) units include acceleration for a comprehensive range of cryptographic functions, ranging from basic symmetric crypto operations (AES, 3DES, hash) to Public Key computations, as well as full security protocol packet processing. A micro-programmable sequencer allows full packet security header and trailer processing to be performed in the MiCA subsystem, thereby offloading the tile cores and improving throughput.

As shown in Figure 5, the MiCA™ (Multicore iMesh Coprocessing Accelerator) blocks connect directly to the iMesh and deliver low-latency cryptographic services to all tiles. Data is transferred directly from and to the tile's coherent memory system via hardware DMA engines and I/O TLBs provide protection and isolation. Independent crypto context structures allow for flexible key management architectures and support millions of simultaneous cryptographic flows.

Since the MiCA system supports virtualized access, each tile is afforded a private view into the accelerator and dozens of crypto operations can be in flight at a given time. Each MiCA block supports 40 contexts and a tile can request to be assigned one or more contexts from the hypervisor.

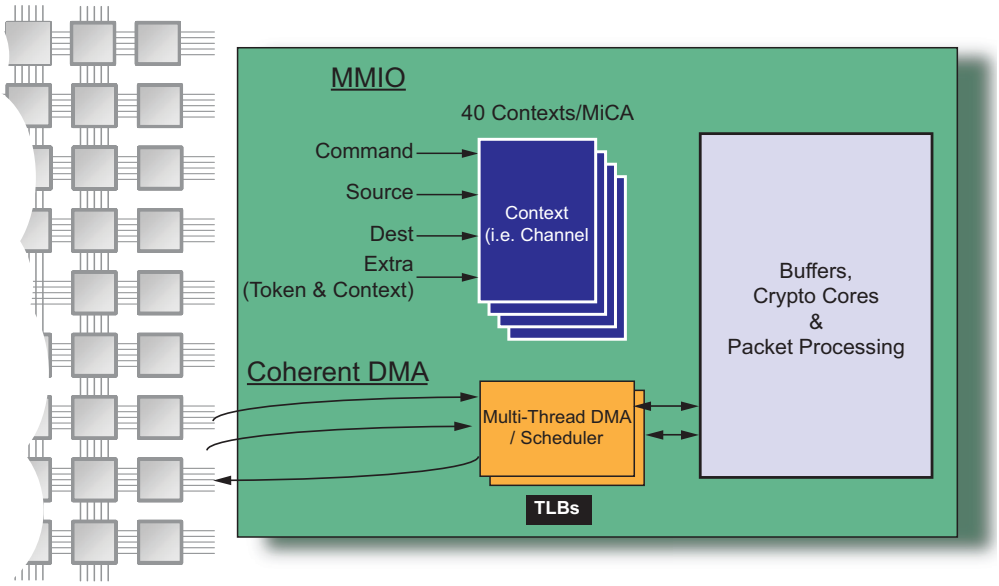

Figure 5. TILE-Gx MiCA Engine

A rich suite of cryptographic algorithms to enable many security protocols are supported, including:

- MACsec (802.1AE)
- IPsec
- SSL/TLS
- SRTP
- Wireless GSM & 3GPP

As an example, some of the IPsec ESP packet processing functions that are accomplished in the MiCA engines include:

- ESP header insertion
- Sequence Number insertion and validation
- Initialization Vector (IV) insertion
- Crypto padding
- Integrity Check Value (ICV) insertion and validation
- ESP trailer insertion

The TILE-Gx8072 MiCA system delivers over 40 Gbps of AES-GCM and supports 50,000 RSA private key transactions/second (1024-bit modulus).

The symmetric encryption algorithms supported are listed in Table 2.

Table 2. Symmetric Encryption Algorithms

Algorithms	Key Size	Modes
AES	128, 192, 256-bits	CBC, GCM, CTR, ECB
3DES	56, 168-bits	CBC, ECB
ARC4	40 – 256-bits	Stream cipher
KASUMI	128-bits	Stream cipher
SNOW 3G	128-bits	Stream cipher

The cryptographic hashing algorithms supported are listed in Table 3.

Table 3. Cryptographic Hashing Algorithms

Algorithms	Modes
MD5	Straight hash, HMAC
SHA-1	Straight hash, HMAC
SHA-256 (SHA-2)	Straight hash, HMAC
SHA-384 (SHA-2)	Straight hash, HMAC
SHA-512 (SHA-2)	Straight hash, HMAC
AES GMAC	Straight hash

The public key algorithms supported include those listed in Table 4.

Table 4. Public Key Algorithms

Algorithms	Key Sizes
RSA (with / without CRT)	Up to 4096-bit
DSA	Up to 2048-bit
Elliptic Curve	Up to 1024-bit
ECDSA	Up to 1024-bit
Diffie-Hellman	Up to 2048-bit

2.3.3 Compression/Decompression Accelerator

In addition to the packet processing and crypto accelerators, the TILE-Gx supports a compress/decompress accelerator. Lossless data compression is accelerated through a high-performance Deflate compress/decompress engine. The compression accelerator uses DMA engines to transfer data from and to coherent tile memory with no tile processor overhead. Each of the compression accelerators provide 40 contexts, allowing for virtualization similar to the cryptographic accelerators. The TILE-Gx8036™ compression system delivers up to 20 Gbps of Deflate compression performance plus up to 20 Gbps of decompression. Both compression and decompression operations can run simultaneously.

The implementation is fully GZIP compliant with support for dynamic Huffman tables and a 32 KB history depth.

2.4 I/O and Memory Interfaces

All I/Os are integrated on the device, enabling designers to create power-efficient and low-cost system designs. For example, the TILE-Gx8072 integrates 24 lanes of PCIe Gen 2 and 8x 10GbE ports. The SERDES lanes for PCIe and Ethernet are not shared, enabling simultaneous transfer at full bandwidth. 4x DDR3 memory interfaces deliver 15 GB per second of ECC-protected memory bandwidth per interface at 1866 MTps. Additional low-speed interfaces are provided, including USB 2.0 and I^2C.

2.4.1 PCIe and TRIO

TILE-Gx processors are often used in an offload capacity, paired with an x86 host processor connected via PCIe. The TILE-Gx family of processors dedicates significant hardware resources to PCIe throughput and functionality, and delivers commensurately higher performance compared to other embedded processors. Like the Ethernet I/O and mPIPE offload, PCIe TRansactional I/O (TRIO) accelerators interface directly into the iMesh, include high performance DMA engines that read and write data directly to and from tile caches, and contain dedicated I/O TLBs. The DMA engines are controlled by 32 ingress and 32 egress descriptor rings, and SR-IOV support provides 32 virtual functions. Endpoint and root complex modes are supported.

2.5 Tools and Software

One of the key benefits of the TILE-Gx architecture is its ability to enable standard programming models and runtime environments, such as ANSI C/C++, Java, and Linux. The goal is to preserve the massive existing investment in code and programming skills, to enable leveraging existing proprietary and open source code, and ultimately to provide "performance productivity"—achieving high performance in less time and development cost. With the proper architectural support, standard environments such as Linux will continue to scale [3].

In particular, Tilera has ported the GNU toolchain and Linux run-time environment to the TILE-Gx architecture. Over 4,000 standard packages are included; recompiled from the CentOS 6.3 source packages. Tilera's GNU and Linux modifications have been contributed back to the open-source community, enabling users to build their own modified environments. A typical user runs a single SMP Linux instance across all cores, but other execution models are possible, including multiple Linux instances on different sets of cores, different supervisors or operating systems, or even bare metal operation directly on the hardware.

While unmodified (architecture-independent) Linux runs well on TILE-Gx, additional software innovations were developed by Tilera that take advantage of the architecture to deliver high performance on manycore systems. In particular, we have developed the following enhancements:

- "Homecaching" control to take advantage of the cache architecture. Each OS page has controllable caching attributes, including specifications for where cache lines are "homed". Common choices are local to a tile's L2 or hardware hashed across a set of tiles' L2 caches. The former is appropriate for data structures that are typically only accessed by the local core, such as stack pages. The latter is appropriate for shared data. TILE-Gx Linux makes some reasonable default choices, allows user control, and implements features such as migrating local cache mappings when processes migrate across tiles.
- "Zero-Overhead Linux (ZOL)" mode to provide near-realtime predictable performance. We extend the "tickless kernel" idea introduced to eliminate timer interrupts when no user threads are running on a core. If only one user thread is running affinitized to a specific core, no timer interrupts are required. To eliminate further sources of OS disruption, device interrupts and background clean-up processes are directed away from the cores that are executing application code. These features leverage the large number of cores, which enables a spatial distribution model, rather than time-multiplexing application threads on a smaller number of cores.
- Multicore debugging and performance analysis tools. Tilera has extended the Eclipse Integrated Development Environment (IDE) with custom plug-ins. "Whole program" debugging is enabled via an extended debug stub that coordinates breakpoints of multiple processes. To aid in performance analysis, information from hardware performance counters is gathered via the standard `oprofile` and `perf_events` frameworks, and displayed in a graphical user interface (GUI) to navigate the rich data spanning multiple tiles and architectural levels (CPU, cache, memory, etc.).

Beyond these enhancements to Linux, Tilera's Multicore Development Environment™ (MDE) provides tuned reference applications, frameworks, and middleware that enable customers to bring high-performance products based on the Tile-Gx processor family to market rapidly. For example,

the ART (Application Run-Time) framework enables development of scalable, high-performance dataplane applications, and the GXIO library enables user-space communication with all on-chip hardware accelerators.

3 Performance Results

The TILE-Gx family of processors delivers industry-leading performance and performance-per-watt for networking, media, and cloud-based computing applications. Representative results are shown in the subsections below, along with the industry-standard CoreMark benchmark.

3.1 Networking

The integrated mPIPE packet processing accelerator combined with abundant on-chip iMesh and cache bandwidth, provides the platform for high-performance 80 Gbps networking applications on TILE-Gx. Specialized atomic instructions such as fetch-and-add-greater-than-zero enable the implementation of high performance lockless queues and FIFOs for inter-thread communication of packet descriptors. The results shown in Table 5 and Table 6, ranging in complexity from L2 forwarding to IDS/IPS (intrusion detection and prevention system), demonstrate best-in-class performance of TILE-Gx8072 on networking applications.

Table 5. TILE-Gx8072 Networking Application Performance

Benchmark Test	Performance	Utilization
Network Bridging and Routing		
Bridging with VLAN Hash Lookup (64B) (10,000 MAC Addresses)	80 Gbps	34 Cores (47%)
LPM IP Forwarding (64B) (10,000 routes)	80 Gbps	44 Cores (61%)
Forwarding with Traffic Shaping (256B)	80 Gbps	60 Cores (83%)
TCP Termination (1.5KB) Packet Throughput	80 Gbps	32 Cores (44%)
Network Packet Monitoring/Brokering		
Lossless Packet Capture to PCIe (64B)	80 Gbps	32 Cores (44%)
NetFlow and Metadata Export	80 Gbps	40 Cores (55%)
Network Security Processing		
IPsec ESP Packet Processing (AES-128 GCM Mode)	40 Gbps (1KB packets)	10 Cores (14%)

Table 5. TILE-Gx8072 Networking Application Performance (continued)

Benchmark Test	Performance	Utilization
SSL 3.0 / TLS 1.2 Record Processing (including TCP/IP)	40 Gbps (16KB records)	18 Cores (25%)
SSL Handshake Processing (RSA w/ CRT, 2048b keys)	8,500 handshakes/sec	4 Cores (5%)
Intrusion Detection System (IDS)		
Suricata IDS System, Policy-Based Ruleset (677 Rules)	13 Gbps	72 Cores (100%)

3.2 Multimedia

Key architectural features of the TILE-Gx family that enable best-in-class performance from a software codec implementation are the sum-of-absolute-differences instruction and 64b SIMD operations (including 8x8b multiplies), sophisticated hardware and software prefetch, and fast inter-thread sharing via the iMesh.

Table 6. TILE-Gx8072 Media Application Performance

Benchmark Test	Performance	Utilization
Video Codecs		
H.264 Video Encode, 1080p30, 4Mbps (BQ Terrace Image Sequence)	26 channels	70 Cores (97%)
H.264 Video Encode, 720p30, 2Mbps (Vidyo1 Image Sequence)	72 channels	70 Cores (97%)
H.264 Video Decode, 1080p, 30fps, 4Mbps	34 channels	70 Cores (97%)
MPEG2 Video Encode, 720p, 30fps, 5Mbps	82 channels	70 Cores (97%)
Audio Codecs		
AMR-NB, Encode + Decode	3000 channels	70 Cores (97%)
G.729AB, Encode + Decode	3500 channels	70 Cores (97%)

3.3 Data Center and Cloud

Table 7. TILE-Gx8072 Cloud Application Performance

Benchmark Test	Performance	Utilization
In-Memory Caching Server (Key Value Store)		
Memcached Transaction Processing (Up to 256GB of DRAM)	4M Transactions/sec <22 µSec latency	70 Cores (97%)
Virtual Switching		
Open vSwitch Port	40 Gbps	18 cores (25%)
Image Compression		
JPEG decode, Re-Size, Encode	2000 images / sec	70 Cores (97%)

3.4 Processor Benchmark

Table 8. TILE-Gx72 Benchmark Performance

Benchmark Test	Performance	Utilization
CoreMark Benchmark		
CoreMark Score (gcc4.4.6)	277,579	71 Cores (98%)

3.5 Performance Scaling

As described throughout this paper, the TILE-Gx processor architecture is designed to deliver scalable performance with increasing core count. The graphs below show TCP and H.264 encoding throughput as core count is increased. In both cases, the throughput increase is nearly linear with core count. TCP throughput performance becomes I/O bound at 80Gbps on the TILE-Gx72.

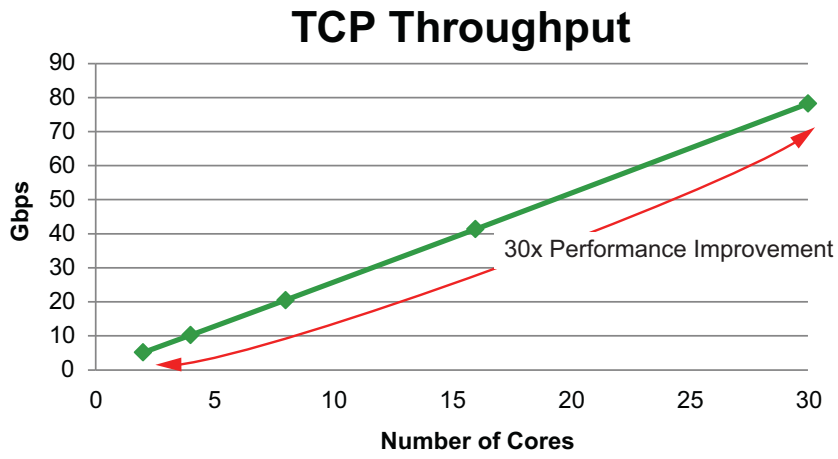

Figure 6. TCP Throughput[3]

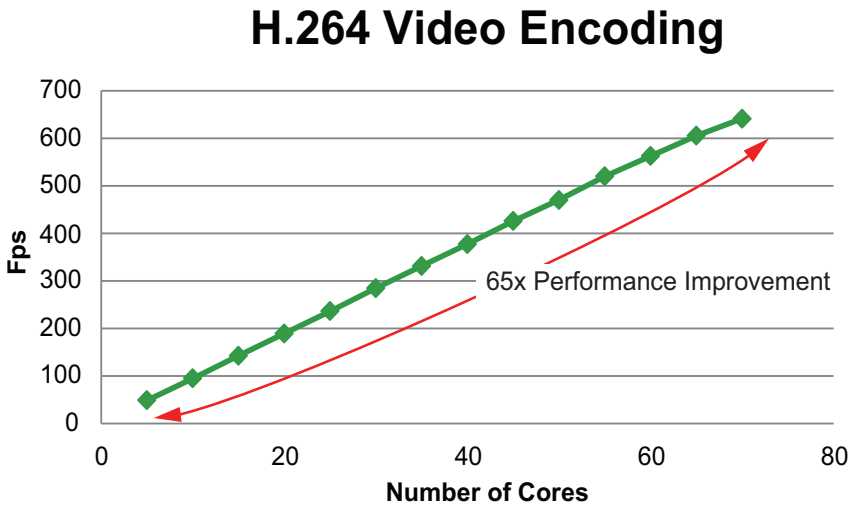

Figure 7. H.264 Video Encoding[4]

3 512 connections, 1500B packets, "echo" server, full-duplex performance.

4 "Toys_and_calendar" video sequence, 3Mbps, baseline profile.

4 Summary

Tilera's TILE-Gx family of manycore processors brings industry-leading performance to applications in the embedded multimedia, networking, and PCIe accelerator markets. This is accomplished while consuming significantly less power than comparable SoCs, and while maintaining a standard Linux, ANSI-C, shared-memory programming environment. Many design factors contribute to this result, but chief among them is the high-bandwidth, low-latency iMesh interconnect that connects large numbers of power-efficient cores, dedicated offload engines, and high-speed I/O. Tilera's devices truly deliver on the promise of multicore.

5 References

[1]

K. Fan, M. Kudlur, G. Dasika, and S. Mahlke, "Bridging the computation gap between programmable and hardwired accelerators," in Proc. Int. Symp. HPCA, 2009, pp. 313–322.

[2]

M.M.K. Martin, M.D. Hill, and D.J. Sorin, "Why On-Chip Cache Coherence is Here to Stay", in Communications of the ACM, July 2012.

[3]

S. Boyd-Wickizer, A.T. Clements, Y. Mao, A. Pesterev, M.F. Kaashoek, R. Morris and N. Zeldovich, "An Analysis of Linux Scalability to Many Cores", in Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI '10), 2010.

.