Comparative Analysis of HPC and Accelerator Devices: Computation, Memory, I/O, and Power

Justin Richardson, Steven Fingulin, Diwakar Raghunathan, Chris Massie, Alan George, Herman Lam

NSF Center for High-Performance Reconfigurable Computing (CHREC)
ECE Department, University of Florida
Gainesville, FL 32611
Email: {richardson,fingulin,raghunathan,massie,george,hlam}@chrec.org

***Abstract*—The computing market constantly experiences the introduction of new devices, architectures, and enhancements to existing ones. Due to the number and diversity of processor and accelerator devices available, it is important to be able to objectively compare them based upon their capabilities regarding computation, I/O, power, and memory interfacing. This paper presents an extension to our existing suite of metrics to quantify additional characteristics of devices and highlight tradeoffs that exist between architectures and specific products. These metrics are applied to a large group of modern devices to evaluate their computational density, power consumption, I/O bandwidth, internal memory bandwidth, and external memory bandwidth.**

I. Introduction

RECENT developments in computing device technologies are pushing devices to multi- and many-core architectures to exploit explicit parallelism rather than instruction-level parallelism. It has become more important to be able to fairly compare the disparate devices that enter the market, which incorporate both fixed and reconfigurable logic, both between and within respective architectural classifications. Performance and power metrics are needed to objectively compare devices in both high-performance computing (HPC) and embedded computing to understand device tradeoffs. These metrics will assist application scientists with algorithm-guided device selection early in the development cycle.

A modern computational device, regardless of whether or not it is an HPC or accelerator device, has several key components that can be used to characterize the device. Fig. 1 shows how four key metrics are related in the characterization of a device for performance comparison.

The first common component in this framework is the computational cores on the device. With the modern multi- and many-core push, we are seeing an increasing number of cores on a device. These cores can take many shapes and sizes but in the end all work to perform the computations that the device uses to complete its tasks. The computational capacity of these cores can be used to compare differing devices if they are computing the same task. *Computational density* (CD) and its power-aware version, *computational density per watt* (CD/W) are metrics for comparing devices based on these computational units.

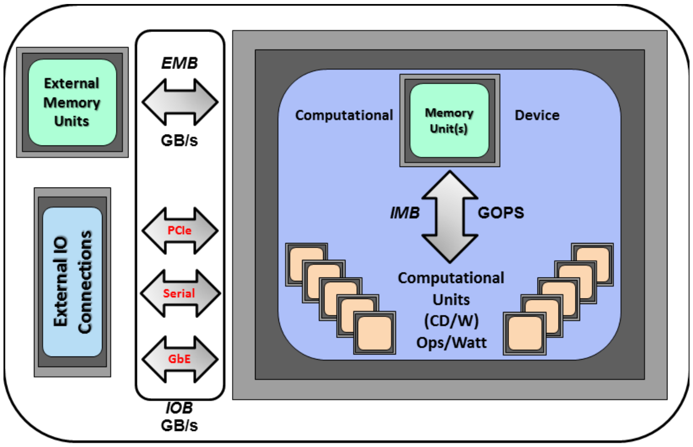


Fig. 1. Device Characterization Framework and Metrics

Another major type of component typically found within a device is internal memory units. These can be distinguished because they are very close to the computational cores and are the lowest in the memory hierarchy. The bandwidth between these memory units and the computational cores will determine how many operations can be fed into the computational units and thus can be a limiting factor in performance. The *internal memory bandwidth* (IMB) metric is used to compare various devices based on how many operations can be sustained by the computational cores for a given application's needs.

The third major component found in modern computational devices starts to move data outside the device. Memory units inside the device are supplemented with external memory units. These units are not as close to the computational cores as the internal memory units, but they are typically much larger and hold application instructions and data ready to be quickly passed into the device for use. The bandwidth of these external memory units is a key metric for application performance. With an increasing number of cores being placed on a device, it has become far more important to keep them fed with data

978-1-4244-9517-7/10/$26.00 © 2010 IEEE

as fast as possible. *External memory bandwidth* (EMB) is a metric that is very useful for quantifying this trait.

The final major component that this paper addresses is the farthest line of communication from the computational cores, the input-output (I/O) ports. These ports allow information, usually data, to be passed from sensors, controls, or any other of a variety of sources into and out of the computational cores. These connections range from slow serial connections for debugging to high-speed serial and parallel communication standards. The ability of a device to keep acquiring and processing meaningful data is key to its ability to perform well in today's environment. The *input-output bandwidth* (IOB) metric is a key metric for comparing devices in environments where large amounts of continuous data processing is required. IOB and the other metrics work together to help compare a set of disparate devices and allow device users to consider their needs in a device before spending significant resources into development.

This paper extends our previous work [1] by introducing two new device metrics. In addition to the original set of metrics defined in [1] for computational density (CD), CD per watt (CD/W) and internal memory bandwidth (IMB), new metrics for off-chip bandwidth are introduced in this paper: external memory bandwidth (EMB) and I/O bandwidth (IOB). These metrics are used to characterize how well each device can interact with the rest of the system. In Section III, the original set of metrics is briefly reviewed and the new metrics are defined in more detail. This paper also extends our previous work by evaluating a set of new devices and new device categories (such as digital signal processors (DSPs) and NVIDIA's Ion platform) based on the complete suite of device metrics (the previous metrics along with EMB and IOB). Results described in Section IV will show interesting outcomes in terms of computational density per watt and a trend of rapidly increasing I/O capabilities of devices. Section V provides a summary and conclusions.

II. Related Work

This work evaluates the computational performance of the two major categories of processing devices: Fixed (FMC) and Reconfigurable (RMC) Multi- or Many-Core devices. Dehon [2] relates the number of processing elements, their width, and clock frequency to performance, normalized by die area and process technology. Dehon's paper proposed a metric similar to the bit-level computational density we use in our work. Our previous work in Williams et al. [1] introduced CD, a metric which is used to determine the computational capability of a device and compare it to other devices, both within and between architectural categories at varying levels of precision such as floating-point, integer, and bit-level. Our work is a direct extension of these metrics by introducing newer bandwidth metrics and adding newer devices to perform a comprehensive evaluation.

Bandwidth metrics form an important part of the device metrics which we introduce. Sohi and Franklin [3] illustrate how low cache bandwidth hampers system performance especially when instruction issuing or parallel processing capabilities increase. Saulsbury et al. [4] suggest that integrating simple single-scalar processors tightly with memory can outperform high-end superscalar processors with traditional memory hierarchies and bandwidth limitations. Burger et al. [5] shows that memory bandwidth is a major performance bottleneck on many benchmarks due to latency-hiding techniques. The operand transfer rate from external memory and the effectiveness of on-chip memory in reusing operands are increasingly having a bigger impact on system performance.

These works have helped to lay the groundwork for this expansion and extension of device metrics and have given a glimpse into the vast landscape of fair device characterization and comparison. In the next section, this paper reviews the specific methodologies used in the CD, CD/W, and IMB metrics and introduces new methodologies for our external extensions of these metrics.

III. Methodology

A. Review of CD and CD/W

The CD metric [1] is used to determine the computational capability of a device and compare it to other devices, both within and between architectural categories at varying levels of precision. A device's capability for computation is characterized by its integer CD, floating-point CD, and bit-level CD at various sizes.

As in [1], we use MC to collectively refer to multi-core and many-core devices, which have at least two major computational components in a single package; Fixed MC (FMC) and Reconfigurable MC (RMC) are the two primary classes. FMC devices have a fixed hardware structure that cannot be changed after fabrication. RMC devices can change their logical hardware structure after fabrication to adapt to changing problem requirements. The reader should refer to [1] for a more detailed discussion on the reconfigurability factors that are used to classify a device as either FMC or RMC.

To determine the integer CD for FMC and coarse-grained RMC devices, Eq. 1 is used, where N_i is the number of integer execution units or the number of integer instructions that can be issued simultaneously of element type i, CPI_i is the average number of clock cycles per integer instruction for element type i (such as DSP, ALU, or LUT resources), and f is the operating frequency of the device. The subscript i represents the type of computational element within the device that is under analysis. The summation over i, in this equation, takes into account architectures that support vector/SIMD integer instructions by including different types of computational components. We assume that only addition and multiplication operations are considered, and the number of parallel operations is maximized while keeping the number of additions and multiplications equal. When calculating the number of parallel operations supported by a device, we consider a hardware-supported, multiply-accumulate operation as only one operation.

$$CD_{int} = f \times \sum_{i} \frac{N_i}{CPI_i} \tag{1}$$

For FPGAs, integer CD is determined using achievable frequency and the number of parallel operations of a fully utilized logic fabric and DSP resources. A single integer core for both addition and multiplication is instantiated on an FPGA using vendor IP cores. For each core, the resource utilization along with the maximum achievable frequency is determined from the vendor tools. This information allows the number of simultaneous cores that can be instantiated on a device to be determined, utilizing all available DSP and logic resources and assuming 15% logic overhead for steering logic and I/O interfacing. Again, only addition and multiplication operations are considered and balanced. The number of parallel operations is multiplied by the maximum achievable frequency, limited by the lowest between multiplication and addition. Based on the amount of available on-chip memory resources, the number of parallel operations is limited in order to incorporate memory bandwidth or on-chip RAM resources for data buffering, which can have a limiting effect on the peak CD. The on-chip memory needs to allocate two operands per operation for memory-sustainable CD, which is the CD used throughout this paper. This provision ensures that the number of parallel operations a device can support is limited by the realistic ability of the internal memory structure to provide data for each parallel operation.

To illustrate the process in which CD is calculated for a device, a Virtex-6 SX475T FPGA can be analyzed for integer performance. For example, when calculating the 32-bit integer (i.e. Int32) CD for the Virtex-6 SX475T, the Int32 IP cores of adders and multipliers are first generated with tools supplied by the vendor. One of each design is synthesized and simulated on the FPGA. Using the utilization report, it can be determined that the fabric could support 1937 operations in parallel, half multiplies and half additions. From [6] we see that the block RAMs of the fabric can only supply 1064 pairs of operands each clock cycle. Since this amount is less than the maximum number of parallel operations, 1064 is the maximum amount of memory-sustainable operations that can be computed in parallel. Using the timing report generated from the design of multipliers and adders, the operating frequency is determined to be 296 MHz. Since each operation takes one cycle, CPI is 1, and we are computing 1064 operations in parallel. Using Eq. 1, the memory-sustainable Int32 CD for the device is calculated as:

$$CD_{int} = 296\ \textit{MHz} \times 1064\ \textit{ops} = 314.944\ \textit{GOPS} \tag{2}$$

The CD per watt (CD/W) metric is calculated by taking the CD for each level of parallelism and dividing by the power consumption at that level of parallelism. For FMC, the maximum power is used. For RMC, power is assumed to scale linearly with resource utilization and achievable frequency. Floating-point CD and CD/W are determined using a similar procedure as integer CD and CD/W. For a more detailed methodology of CD and CD/W, please refer to [1] [7] [8].

B. Review of IMB

IMB is used to measure the on-chip memory interface capabilities. It quantifies the memory performance of a system by measuring the rate at which data can be transferred from on-chip memories to the processing elements. IMB is important because memory often becomes a bottleneck, limiting the amount of operands supplied to the processing elements of the system. It is defined separately for cache-based systems (CBS) and block-based systems (BBS). Eq. 3 from [1] is used to calculate the IMB for BBS.

$$IMB_{block} = \sum_{i} \frac{N_i \times P_i \times W_i \times f_i}{8 \times CPA_i} \tag{3}$$

In Eq. 3, N_i is the amount of block memories of type i, W_i is the data width, P_i is the amount of ports for memory of type i, CPA_i is the number of cycles per access to memory, division by 8 is to convert from bits per second to bytes per second, and f_i is the frequency of the device, or if the frequency is not constant, as in FPGAs, then it is variable up to the operating frequency of the design. Once again, the subscript i denotes memory type to support devices that have more than one type of internal memory.

In CBS, multiple separate levels of cache may exist, and IMB is calculated separately for each, so that the hit-rate consideration is only included per level of cache. They also feature hardware to determine associativity, line size, coherency protocol, replacement algorithms, etc. These parameters affect the access times of the cache, and are addressed by the frequency and cycles per access variables. The equation used to calculate IMB for CBS is given from [1] as follows:

$$IMB_{cache} = \%\ \textit{hitrate} \times \sum_{i} \frac{N_i \times P_i \times W_i \times f_i}{8 \times CPA_i} \tag{4}$$

In Eq. 4, N_i is the number of block memories of type i, W_i is the data width, P_i is the number of ports for memory of type i, CPA_i is the number of cycles per access to memory, division by 8 is to convert from bits to bytes per second, and f_i is the frequency of the device.

IMB does not account for register access times; instead it is assumed that these are internal components separate from block or cache memory. Registers are usually quickly accessed and do not hinder performance. IMB only seeks to evaluate the rate at which processing elements have all the necessary operands, and it is assumed that registers do not limit this rate.

To aid the understanding of this calculation, an example using the Virtex-6 SX475T is presented. On the Virtex-6 SX475T, there are 1064 36Kb block RAMs. Each of these has a 72-bit port width, and simple dual-port functionality. The maximum operating frequency is used, which is 600 MHz. Since all of these BRAMs are the same, the summation is over this single set. Using Eq. 3, the IMB for the device is calculated as:

$$IMB_{block} = \frac{1064 \times 2 \times 72 \times 600\ \textit{MHz}}{8 \times 1} = 7776\ \textit{GB/s} \tag{5}$$

C. EMB

EMB is proposed and introduced in this paper to describe the total bandwidth achievable to external memory from a device or vice-versa. EMB only includes the bandwidth of usable data, so extra bits used for error-correction coding (ECC) are not included. EMB and IOB (detailed in Section III-D), are introduced to characterize the capability of a device to interface with the rest of the system. EMB does not include I/O bandwidth or network-controller bandwidth as these are typically at the cost of a user-defined interfacing implementation for an application. EMB is defined only for directly attached memory. Although a device could access another device's memory through an I/O port, this is not considered in the calculation of EMB.

For FMC and coarse-grained RMC devices with built-in memory controllers, EMB is the sum of the concurrent EMB provided by all memory controllers. For devices that use a front-side bus (FSB), the entire bus is allocated for memory bandwidth. Otherwise, the external datapath width and the switching frequency are used to determine EMB. To determine EMB for FPGAs, the methodology employed is similar to determining CD for FPGAs. A single memory controller is instantiated on the FPGA using a vendor IP core. The resource utilization is determined along with the maximum-achievable memory interface frequency. The number of simultaneous cores that can be instantiated utilizing all available resources are determined, assuming 15% logic overhead for steering logic and I/O interfacing. Limiting factors include the number of LUTs, ALMs/Slices, and the number of bonded IOBs.

To get a better understanding of how to calculate EMB, a step by step calculation for Virtex-6 SX475T is as follows. A single DDR2 memory-controller IP core is instantiated on the chip and the resource utilization is obtained. The maximum number of DDR2 controllers that can be instantiated simultaneously is calculated by dividing the available number of bonded IOBs (840) by the number of IOBs used (121) by a single memory controller. The memory-interface frequency (533 MHz) is multiplied with the memory-interface width (64 bits) using the appropriate units, to get the EMB of one DDR2 controller as 8.528 GB/s. This rate is in-turn multiplied by 6, the maximum number of DDR2 controllers instantiated to calculate a maximum EMB of 51.168 GB/s.

D. IOB

IOB is proposed here and used to describe the total I/O capabilities of a device, not just the external memory bandwidth. Devices with dedicated ports for interfacing with memory often also have additional ports for data input/output, which are not considered in the EMB calculation. Devices may also have higher bandwidth capabilities on a port that shares all or some pins with ones used for a memory interface, such as is the case for FPGAs. An I/O bandwidth metric describes the maximum data throughput of a device that EMB omits or a higher total bandwidth that is possible on a device.

IOB is calculated as the total aggregate sum of the bandwidth provided by all inputs and outputs that can operate concurrently. The highest bandwidth ports are used when there is overlap or non-concurrency. Eq. 6 shows the aggregation of I/O ports based on i, where i represents the different types of I/O ports that can be used concurrently. Line encoders can be used to encode data into a different format which benefits transmission for reasons other than data throughput. Various schemes such as 8b/10b or 64b/66b, can be employed that have varying overheads on the line rate. If an encoding scheme is used, such as 8b/10b, then α_i represents the fraction of IOB that is available for data. For the case of 8b/10b encoding, we assume that fraction is 0.80. The aggregate sum is then added to the input/output bandwidth of any dedicated external memory controllers available on the device (denoted as IOB_{mem}).

$$IOB = IOB_{mem} + \sum_i \alpha_i \times IOB_i \tag{6}$$

There are numerous ways to characterize the I/O of a device. In single-ended I/O, one signal is made between two ICs and compared to a specified voltage range or to a reference voltage. In differential signaling, two signals are made between two ICs and the signals are compared to each other to determine the logic value [9]. These two signaling methods can have differing bandwidths, even when comparing two single-ended signals to a single differential signaling pair. When studying devices' IOB, it is important to use fair comparisons and keep all parameters equal when direct comparisons are desired.

For an example, consider the Nvidia Tesla C1060 GPU. There are two interfaces for IO on this processor, the memory interface and the PCIe bus. To compute IOB, the aggregate is taken of both interfaces and the calculations are shown in Eqs. 7-9. It has 4 GB of dedicated GDDR3 memory clocked at 800 MHz on a 512-bit interface. The PCIe interface has a 500 MB/s transfer rate for each lane in each direction.

$$IOB_{mem} = 800 \textit{ MHz} \times (512 \textit{ bits}/8) \times 2 = 102.4 \textit{ GB/s} \tag{7}$$

$$IOB_{PCIe} = 500 \textit{ MB/s} \times 16 \textit{ Lanes} \times 2 = 16 \textit{ GB/s} \tag{8}$$

$$IOB = IOB_{mem} + 1 \times IOB_{PCIe} = 118.4 \textit{ GB/s} \tag{9}$$

IV. Results and Analysis

In this section we focus on reporting trends observed amongst various devices when the suite of metrics is applied to them. For our study, a full range of results were collected, including Bit, Int16, Int32, Single-Precision Floating Point (SPFP) and Double-Precision Floating Point (DPFP) forms of CD and CD/W, as well as IMB, EMB, and IOB for the devices shown in the following figures. Due to the large number of devices which have been included in our study, we illustrate the more interesting cases through graphs in this section and have reported all the other metrics in expansive tables in the Appendix. Many of these devices are new to this paper and were not included in [1]. Only selected metrics are presented in detail: Int16 CD/W for RMC and FMC devices; SPFP CD/W for RMC and FMC devices; EMB for FMC and RMC devices; and IOB for FMC and RMC devices.

The FMC devices highlighted in this study, listed in Fig. 3, include a range of CPUs, DSPs, and GPUs. The RMC devices, listed in Fig. 2, include FPGAs of varying types, the Tilera TILE64 processor, and the PACT XPP-3c. In this study the maximum level of exploitable parallelism of devices was calculated and the performance of devices is compared by varying the level of parallelism. This means that the clock frequency is recalculated for each metric (i.e. Int16 vs Int32) and varies based on which metric is being calculated.

A. CD and CD/W Metrics

Fig. 2 shows Int16 and SPFP forms of CD/W for RMC devices. The bars are grouped by device, one representing Int16 precision and the other SPFP. The data labels above each bar denote the maximum number of parallel operations that each device is capable of sustaining at the given precision. Some interesting results can be observed from the figure. The EP4SE530 has the highest memory-sustainable Int16 CD/W (54.07 GOPS/Watt) due to the large logic fabric and the high amount of on-chip memory. The Virtex-6 LX760 has the highest memory-sustainable SPFP CD/W (10.27 GOPS/Watt). The PACT XPP-3c has the largest number of parallel operations (348) of non-FPGA devices studied and has higher Int16 CD/W (16.24 GOPS/Watt) than all the other coarse-grained RMC devices such as the TILE64 even though it has no SPFP support.

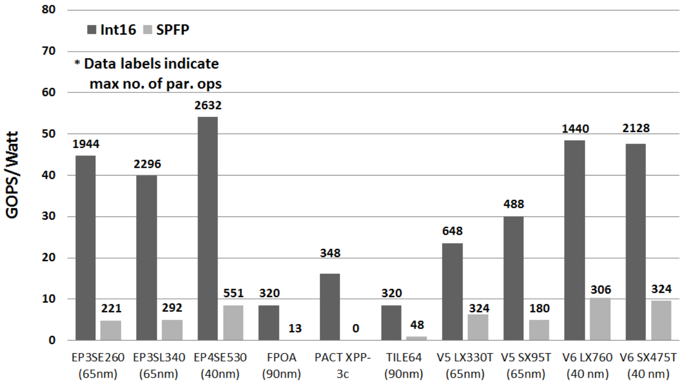


Fig. 2. CD/W for RMC devices

Fig. 3 shows Int16 and SPFP forms of CD/W for FMC devices in the same format as Fig. 2. The TI-OMAP device has the highest Int16 CD/W ratio (7.72 GOPS/Watt). Even though this device does not have a large number of computational resources, it achieves high CD/W due to its low power consumption. Another trend visible from this figure is significantly lower CD/W numbers for the FMC devices as compared to RMC which can be directly attributed to the presence of a vast number of computational resources on a FPGA and the higher power consumption of FMC devices. The Virtex-6 LX760 has the greatest unconstrained Int16 CD (3443.2 GOPS), but it is limited by its IMB, which significantly lowers the memory-sustainable CD/W to 48.37 GOPS/Watt.

For SPFP CD/W, the results show that most RMC devices perform better than GPU devices; this is because GPU devices have high power consumption which offsets the benefits of its superior SPFP performance. For high levels of parallelism, the GeForce GTX480 has the highest SPFP CD/W (4.12 GOPS/Watt) of the FMC devices studied. Although high-end FPGAs in the Stratix IV and Virtex-6 families have a much larger number of parallel operations than the GeForce GTX480, the achievable frequency is low compared to the operating frequency of the GTX480. However, the FPGAs mentioned have a CD/W that is 3x larger than the GTX480. Interestingly, the GTX285 does not perform nearly as well as the GTX480 even though they are of the same family due to the high power consumption of the device and lower number of processors. Similarly, the Core i7-980X (1.84 GOPS/Watt) and Itanium 9350 (1.18 GOPS/Watt) have high CD performance, but are not power-efficient. The TI OMAP-L137 DSP uses very low power which allows it to perform well in CD/W, despite it having the lowest CD of the devices studied.

For a complete list of devices with their respective CD values, see Table V in the Appendix. Table II in the Appendix, reports CD for FPGA devices for Int16 and Int32 precisions. This table is included to point out the difference in frequencies and power usage over various precisions for FPGAs.

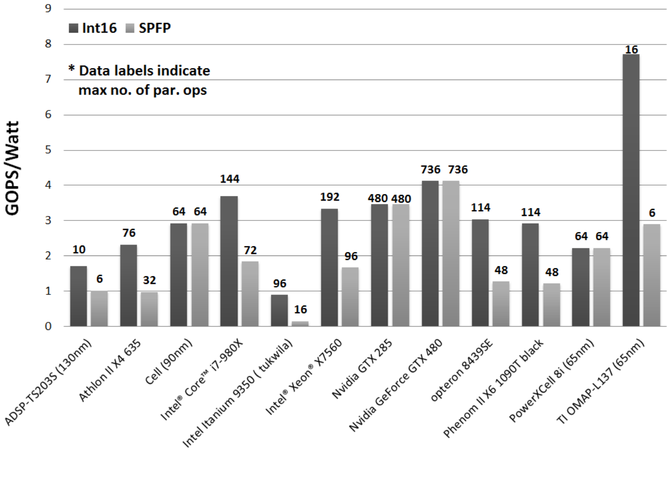


Fig. 3. CD/W for FMC devices

B. EMB Results

Fig. 4 shows EMB for key RMC devices in GB/s. The Virtex-6 LX760 has the highest EMB (68.2 GB/s) of the devices studied due to the fact that the LX760 has a higher number of bonded IOBs than corresponding devices. These bonded IOBs allow the simultaneous instantiation of a higher number of DDR2 controllers resulting in a higher EMB. The numbers above each bar in the graph show the remaining logic utilization after the instantiation of memory controllers required to attain maximum EMB. These percentages illustrate the fact that a very low logic overhead is required to have multiple memory controllers and hence attain a higher EMB. Interestingly it is seen that most of the devices require almost no logic utilization after instantiating their memory controllers whereas the Virtex-4 SX55 occupies almost 40 % of the chip for its two memory controllers.

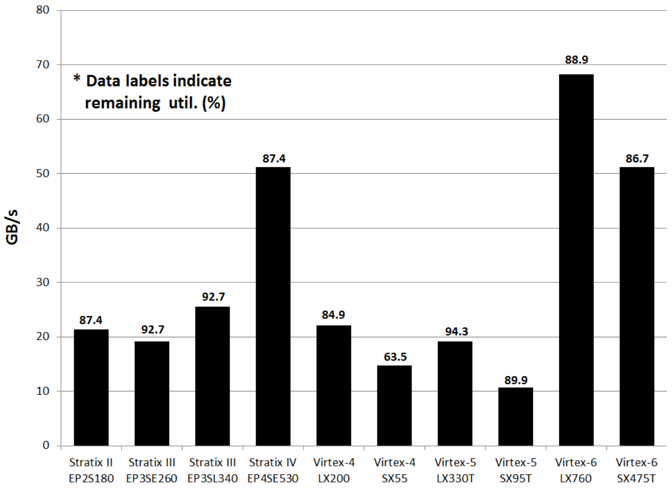


Fig. 4. EMB for FPGA Devices

Fig. 5 shows EMB for FMC devices in GB/s. As expected, GPU devices perform the best amongst all categories of devices. GPUs are designed to handle highly parallel applications which require large sets of streaming data. This design makes it necessary to have fast and wide memory buses that result in high EMB. The Nvidia GeForce GTX480 has the highest EMB (177.4 GB/s) amongst all devices. CPU devices typically handle smaller applications using smaller sets of data, hence they have a lesser EMB. The Intel Xeon X7560 has the highest EMB (34.11 GB/s) amongst the non-GPU FMC devices, which is due to the addition of the high-bandwidth Intel Quick Path Interconnect (QPI).

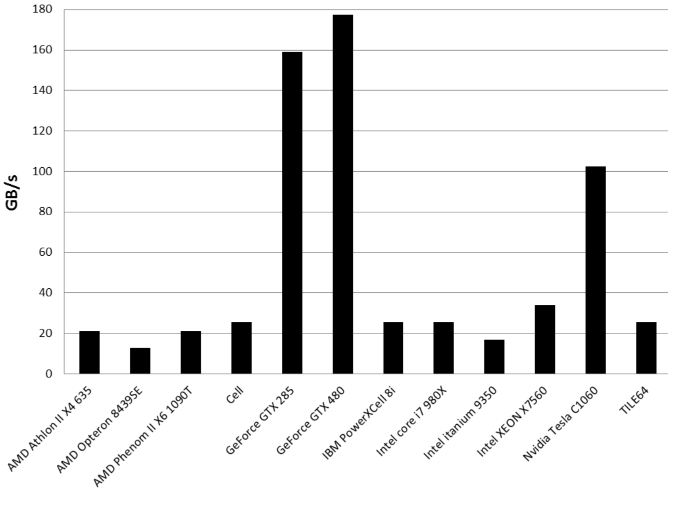


Fig. 5. EMB for FMC Devices

C. IOB Results

Fig. 6 shows IOB for both FMC and RMC devices in GB/s. The package of an FPGA is shown in parenthesis. The FPGAs assume differential signaling and an 8b/10b encoding scheme, which has an overhead of 20 percent, in order to compare the I/O data rates of FMC devices to the I/O line rates of the FPGAs. The I/O bandwidth shown consists of equal parts input and output. It should be noted that an unbalanced I/O can have an effect on the total I/O achievable by a device since not all channels are bidirectional and there may be an unequal number of input and output ports.

The GeForce GTX 480 has the highest IOB (193.4 GB/s) of the devices studied, followed by the GTX 285 (175 GB/s). GPUs are optimized for 3D rendering, which requires processing on large working sets of data. This form of data processing makes having a wide and fast memory bus a necessity to achieve high performance. Comparing against microprocessors, the amount of data that needs to be processed is too large to fit in the cache of a CPU. The working set of applications that typically run on CPUs have random memory access patterns and are smaller than those that run on a GPU, requiring many frequent fetches from off-chip memory. CPU memory interfaces are shifting from buses to a very fast group of serial data lines communicating via packets with much lower latency, such as HyperTransport or Intel's QPI. As CPUs have been increasing in the number of cores and IOB, streaming applications may be more effectively parallelized on them.

V. Conclusions

We have enhanced our existing methodology for device metrics to assess the off-chip memory bandwidth of devices, using external memory bandwidth (EMB) and I/O bandwidth (IOB). Developers can use the device metrics described in this paper and in [1] to assist in algorithm-guided device selection early in the development cycle and to understand device tradeoffs. We have also presented a study of a new and diverse set of devices to determine their computational capabilities and off-chip bandwidths. There is a large variation in the resulting data that arises when these metrics are used to study disparate accelerator technologies.

A few interesting trends observed by evaluating the data include FPGA devices showing the highest CD and CD/W for bit and integer operations. This trend can be attributed to the large fabrics and amount of LUTs enabling FPGAs to achieve massive amounts of parallelism. As observed in Section IV, GPUs tend to perform well in most categories; however, they stand out in floating-point calculations due to the high clock rates of their shader units and the sheer number of them (GTX480 has the highest SPFP CD: 1031.136 GOPS). CPUs also perform well in floating-point, especially double-precision. Many of the other devices have to expend extra resources or clock cycles for DPFP calculations, however modern CPUs have dedicated functional units for that purpose. Combined with the highest clock speeds of any of the studied devices, they perform DPFP operations well.

The EMB and IOB results show that GPUs have the highest external bandwidth, with very wide memory controllers working at very high frequencies. FPGAs also perform well,

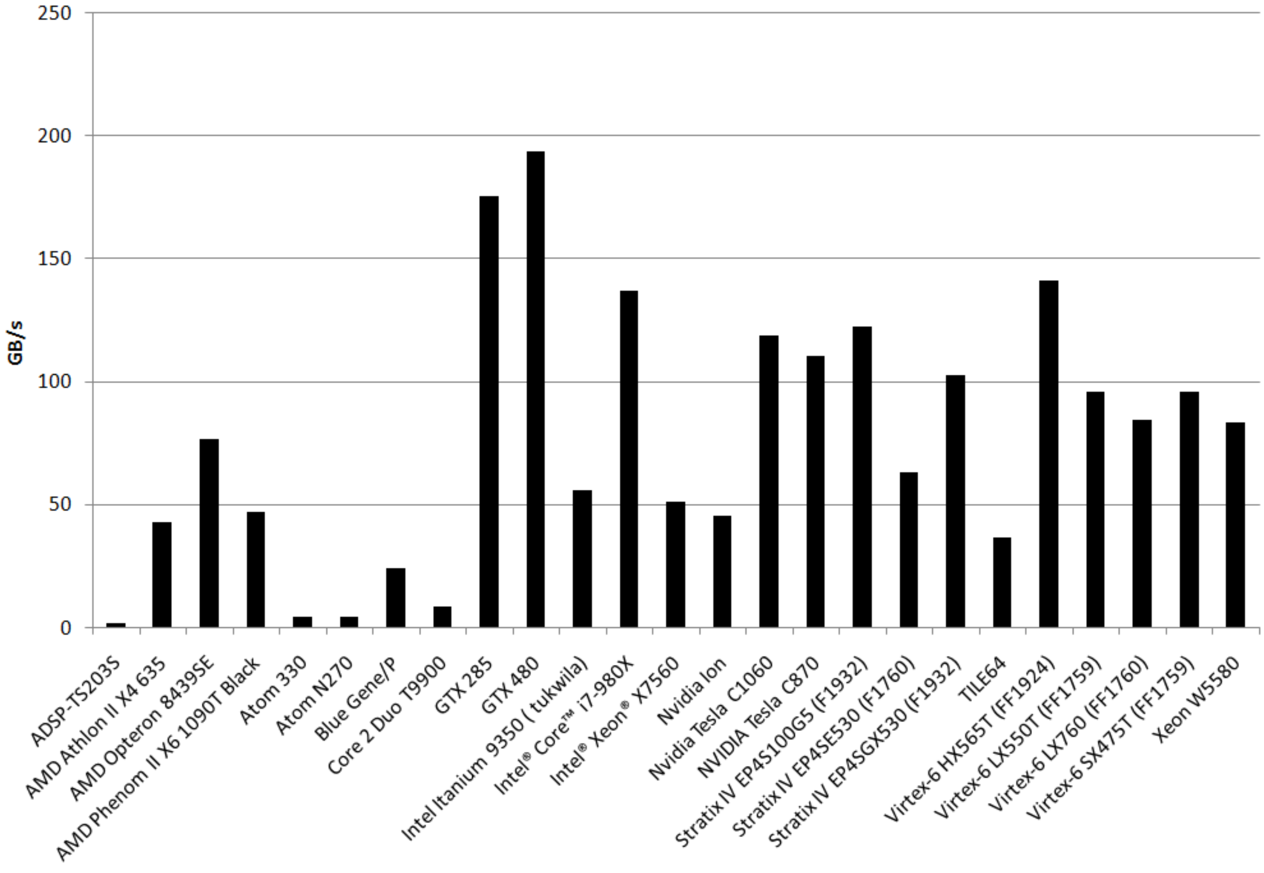


Fig. 6. IOB data for selected devices

but their operating frequency keeps them from matching GPU memory bandwidth. Some of the newer CPUs, particularly the Intel Core i7-980x, achieve very high IOB scores as compared to other CPUs. This is due to the shift from the FSB connection to QPI. Using a point-to-point interconnect allows a much higher clock rate than a shared bus, providing much higher data bandwidth.

Future work is planned to allow for more user defined parameters when calculating certain metrics. These parameters include the addition of more avalible operations and varying the ratio of operations when determining CD. This expansion will allow users to more closely determine which device would fit their algorithm based upon the required calculations. Another planned extension of these metrics includes the parameterization of IOB and EMB metrics. Our goal is to allow users to define and customize the individual attributes used to calculate each metric to best suit their application.

VI. Acknowledgments

This work was supported in part by the I/UCRC Program of the National Science Foundation under Grant No. EEC-0642422. The authors gratefully acknowledge vendor equipment and/or tools provided by various vendors that helped make this work possible.

References

[1] J. Williams, A. George, J. Richardson, K. Gosrani, C. Massie, and H. Lam, "Characterization of fixed and reconfigurable multi-core devices for application acceleration," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 3, no. 4, 2011.

[2] A. DeHon, "Reconfigurable architectures for general-purpose computing," Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 1996.

[3] G. S. Sohi and M. Franklin, "High-bandwidth data memory systems for superscalar processors," *SIGOPS Operating Systems Review*, vol. 25, no. Special Issue, pp. 53–62, 1991.

[4] A. Saulsbury, F. Pong, and A. Nowatzyk, "Missing the memory wall: the case for processor/memory integration," in *ISCA '96: Proceedings of the 23rd Annual International Symposium on Computer Architecture*. New York, NY, USA: ACM, 1996, pp. 90–101.

[5] D. Burger, J. R. Goodman, and A. Kägi, "Memory bandwidth limitations of future microprocessors," in *ISCA '96: Proceedings of the 23rd Annual International Symposium on Computer Architecture*. New York, NY, USA: ACM, 1996, pp. 78–89.

[6] *Virtex-6 Family Overview*, Xilinx, Inc., 2008.

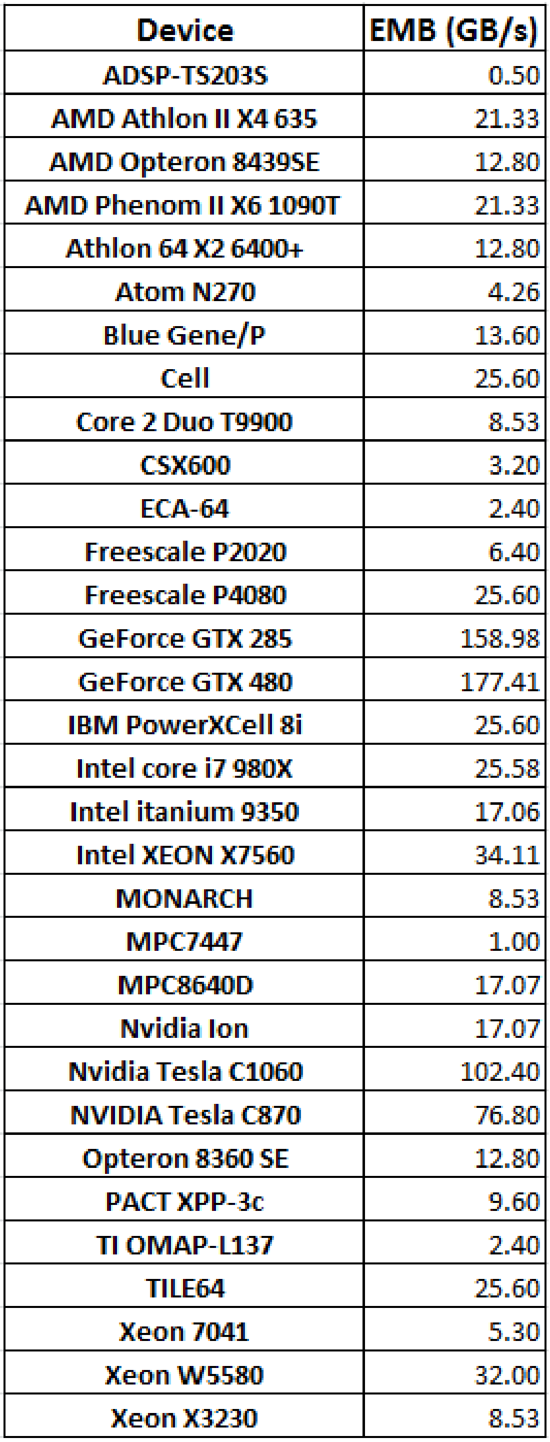
[7] J. Williams, A. George, J. Richardson, K. Gosrani, and S. Suresh, "Computational density of fixed and reconfigurable multi-core devices for application acceleration," *Proc. of Reconfigurable Systems Summer Institute 2008 (RSSI)*, July 7-10, 2008.

[8] ——, "Fixed and reconfigurable multi-core device characterization for hpec," *Proc. of High-Performance Embedded Computing Workshop (HPEC)*, Sep. 23-25, 2008.

[9] A. Athavale and C. Christensen, *High-Speed Serial I/O Made Simple, A Designers' Guide, with FPGA Applications*, Xilinx Connectivity Solutions, April 2005.

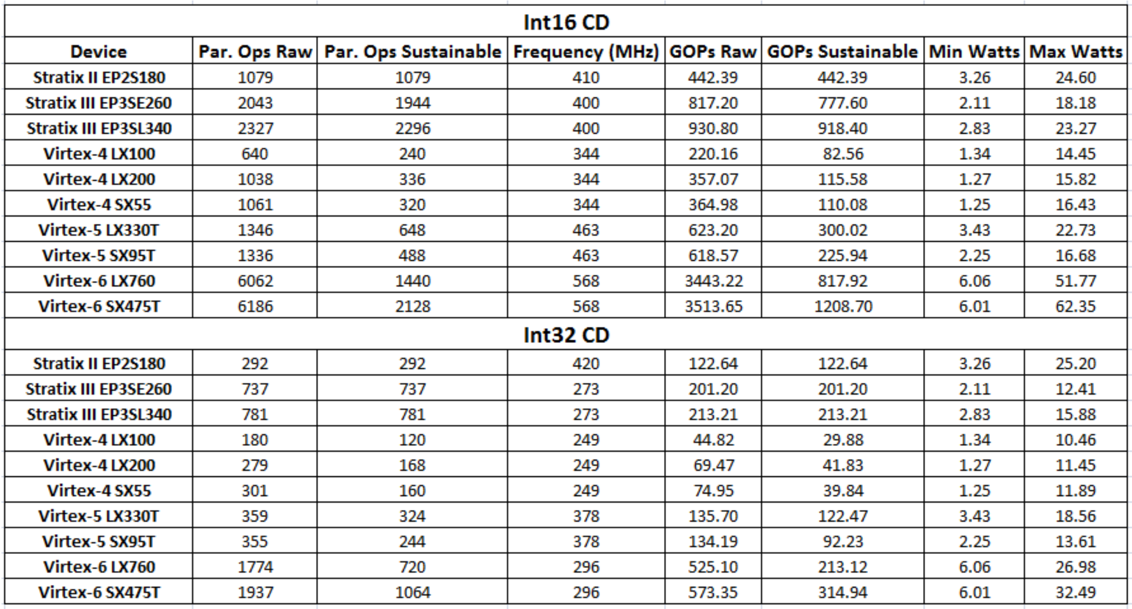
VII. Appendix: Additional Data

TABLE I
EMB of non-FPGA devices



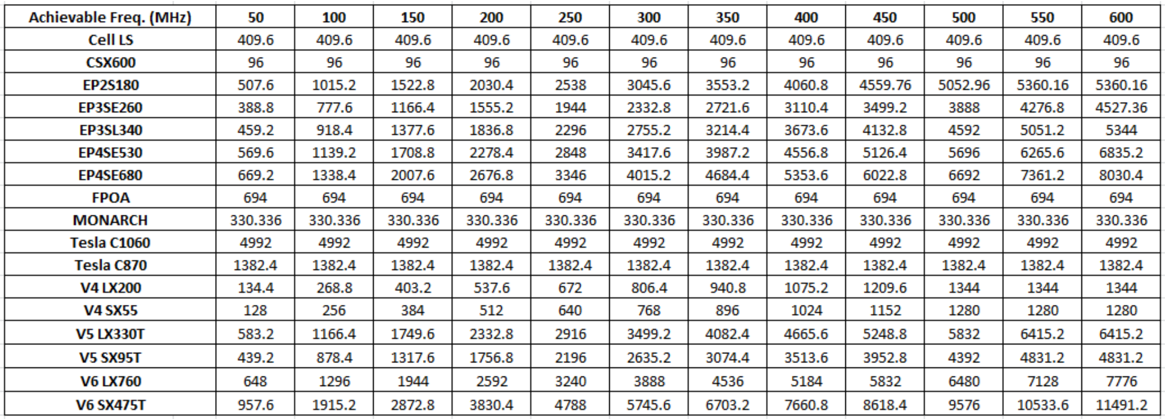
Device	EMB (GB/s)
ADSP-TS203S	0.50
AMD Athlon II X4 635	21.33
AMD Opteron 8439SE	12.80
AMD Phenom II X6 1090T	21.33
Athlon 64 X2 6400+	12.80
Atom N270	4.26
Blue Gene/P	13.60
Cell	25.60
Core 2 Duo T9900	8.53
CSX600	3.20
ECA-64	2.40
Freescale P2020	6.40
Freescale P4080	25.60
GeForce GTX 285	158.98
GeForce GTX 480	177.41
IBM PowerXCell 8i	25.60
Intel core i7 980X	25.58
Intel itanium 9350	17.06
Intel XEON X7560	34.11
MONARCH	8.53
MPC7447	1.00
MPC8640D	17.07
Nvidia Ion	17.07
Nvidia Tesla C1060	102.40
NVIDIA Tesla C870	76.80
Opteron 8360 SE	12.80
PACT XPP-3c	9.60
TI OMAP-L137	2.40
TILE64	25.60
Xeon 7041	5.30
Xeon W5580	32.00
Xeon X3230	8.53

TABLE II
CD of FPGA devices showing frequency and power variations



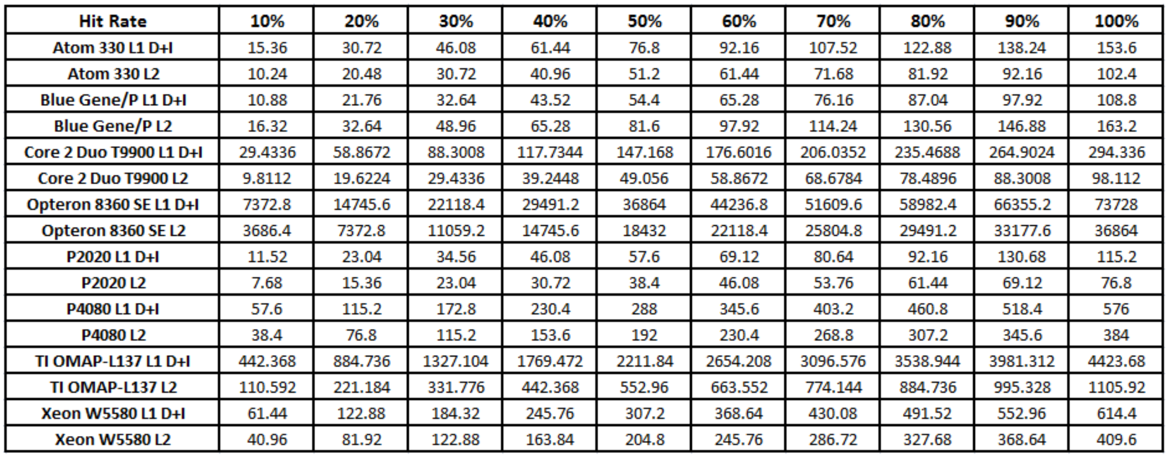
Int16 CD							
Device	**Par. Ops Raw**	**Par. Ops Sustainable**	**Frequency (MHz)**	**GOPs Raw**	**GOPs Sustainable**	**Min Watts**	**Max Watts**
Stratix II EP2S180	1079	1079	410	442.39	442.39	3.26	24.60
Stratix III EP3SE260	2043	1944	400	817.20	777.60	2.11	18.18
Stratix III EP3SL340	2327	2296	400	930.80	918.40	2.83	23.27
Virtex-4 LX100	640	240	344	220.16	82.56	1.34	14.45
Virtex-4 LX200	1038	336	344	357.07	115.58	1.27	15.82
Virtex-4 SX55	1061	320	344	364.98	110.08	1.25	16.43
Virtex-5 LX330T	1346	648	463	623.20	300.02	3.43	22.73
Virtex-5 SX95T	1336	488	463	618.57	225.94	2.25	16.68
Virtex-6 LX760	6062	1440	568	3443.22	817.92	6.06	51.77
Virtex-6 SX475T	6186	2128	568	3513.65	1208.70	6.01	62.35
Int32 CD							
Stratix II EP2S180	292	292	420	122.64	122.64	3.26	25.20
Stratix III EP3SE260	737	737	273	201.20	201.20	2.11	12.41
Stratix III EP3SL340	781	781	273	213.21	213.21	2.83	15.88
Virtex-4 LX100	180	120	249	44.82	29.88	1.34	10.46
Virtex-4 LX200	279	168	249	69.47	41.83	1.27	11.45
Virtex-4 SX55	301	160	249	74.95	39.84	1.25	11.89
Virtex-5 LX330T	359	324	378	135.70	122.47	3.43	18.56
Virtex-5 SX95T	355	244	378	134.19	92.23	2.25	13.61
Virtex-6 LX760	1774	720	296	525.10	213.12	6.06	26.98
Virtex-6 SX475T	1937	1064	296	573.35	314.94	6.01	32.49

TABLE III
IMB for BBS over a range of achievable frequencies (GB/s)



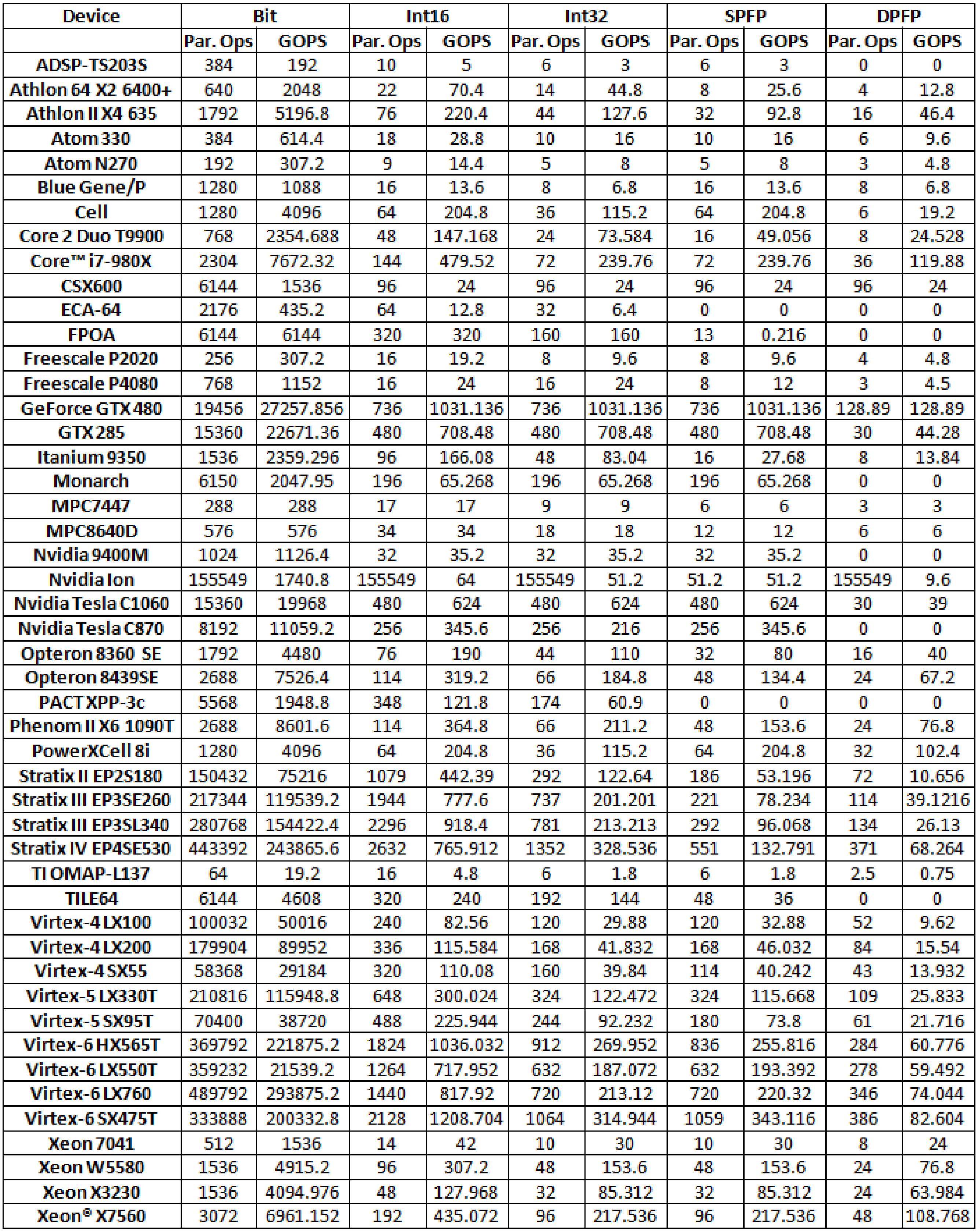
Achievable Freq. (MHz)	**50**	**100**	**150**	**200**	**250**	**300**	**350**	**400**	**450**	**500**	**550**	**600**
Cell LS	409.6	409.6	409.6	409.6	409.6	409.6	409.6	409.6	409.6	409.6	409.6	409.6
CSX600	96	96	96	96	96	96	96	96	96	96	96	96
EP2S180	507.6	1015.2	1522.8	2030.4	2538	3045.6	3553.2	4060.8	4559.76	5052.96	5360.16	5360.16
EP3SE260	388.8	777.6	1166.4	1555.2	1944	2332.8	2721.6	3110.4	3499.2	3888	4276.8	4527.36
EP3SL340	459.2	918.4	1377.6	1836.8	2296	2755.2	3214.4	3673.6	4132.8	4592	5051.2	5344
EP4SE530	569.6	1139.2	1708.8	2278.4	2848	3417.6	3987.2	4556.8	5126.4	5696	6265.6	6835.2
EP4SE680	669.2	1338.4	2007.6	2676.8	3346	4015.2	4684.4	5353.6	6022.8	6692	7361.2	8030.4
FPOA	694	694	694	694	694	694	694	694	694	694	694	694
MONARCH	330.336	330.336	330.336	330.336	330.336	330.336	330.336	330.336	330.336	330.336	330.336	330.336
Tesla C1060	4992	4992	4992	4992	4992	4992	4992	4992	4992	4992	4992	4992
Tesla C870	1382.4	1382.4	1382.4	1382.4	1382.4	1382.4	1382.4	1382.4	1382.4	1382.4	1382.4	1382.4
V4 LX200	134.4	268.8	403.2	537.6	672	806.4	940.8	1075.2	1209.6	1344	1344	1344
V4 SX55	128	256	384	512	640	768	896	1024	1152	1280	1280	1280
V5 LX330T	583.2	1166.4	1749.6	2332.8	2916	3499.2	4082.4	4665.6	5248.8	5832	6415.2	6415.2
V5 SX95T	439.2	878.4	1317.6	1756.8	2196	2635.2	3074.4	3513.6	3952.8	4392	4831.2	4831.2
V6 LX760	648	1296	1944	2592	3240	3888	4536	5184	5832	6480	7128	7776
V6 SX475T	957.6	1915.2	2872.8	3830.4	4788	5745.6	6703.2	7660.8	8618.4	9576	10533.6	11491.2

TABLE IV
IMB for CBS for various hit rates (GB/s)



Hit Rate	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Atom 330 L1 D+I	15.36	30.72	46.08	61.44	76.8	92.16	107.52	122.88	138.24	153.6
Atom 330 L2	10.24	20.48	30.72	40.96	51.2	61.44	71.68	81.92	92.16	102.4
Blue Gene/P L1 D+I	10.88	21.76	32.64	43.52	54.4	65.28	76.16	87.04	97.92	108.8
Blue Gene/P L2	16.32	32.64	48.96	65.28	81.6	97.92	114.24	130.56	146.88	163.2
Core 2 Duo T9900 L1 D+I	29.4336	58.8672	88.3008	117.7344	147.168	176.6016	206.0352	235.4688	264.9024	294.336
Core 2 Duo T9900 L2	9.8112	19.6224	29.4336	39.2448	49.056	58.8672	68.6784	78.4896	88.3008	98.112
Opteron 8360 SE L1 D+I	7372.8	14745.6	22118.4	29491.2	36864	44236.8	51609.6	58982.4	66355.2	73728
Opteron 8360 SE L2	3686.4	7372.8	11059.2	14745.6	18432	22118.4	25804.8	29491.2	33177.6	36864
P2020 L1 D+I	11.52	23.04	34.56	46.08	57.6	69.12	80.64	92.16	130.68	115.2
P2020 L2	7.68	15.36	23.04	30.72	38.4	46.08	53.76	61.44	69.12	76.8
P4080 L1 D+I	57.6	115.2	172.8	230.4	288	345.6	403.2	460.8	518.4	576
P4080 L2	38.4	76.8	115.2	153.6	192	230.4	268.8	307.2	345.6	384
TI OMAP-L137 L1 D+I	442.368	884.736	1327.104	1769.472	2211.84	2654.208	3096.576	3538.944	3981.312	4423.68
TI OMAP-L137 L2	110.592	221.184	331.776	442.368	552.96	663.552	774.144	884.736	995.328	1105.92
Xeon W5580 L1 D+I	61.44	122.88	184.32	245.76	307.2	368.64	430.08	491.52	552.96	614.4
Xeon W5580 L2	40.96	81.92	122.88	163.84	204.8	245.76	286.72	327.68	368.64	409.6

TABLE V
MEMORY-SUSTAINABLE CD ACROSS PRECISIONS



Device	Bit		Int16		Int32		SPFP		DPFP	
	Par. Ops	GOPS	Par. Ops	GOPS	Par. Ops	GOPS	Par. Ops	GOPS	Par. Ops	GOPS
ADSP-TS203S	384	192	10	5	6	3	6	3	0	0
Athlon 64 X2 6400+	640	2048	22	70.4	14	44.8	8	25.6	4	12.8
Athlon II X4 635	1792	5196.8	76	220.4	44	127.6	32	92.8	16	46.4
Atom 330	384	614.4	18	28.8	10	16	10	16	6	9.6
Atom N270	192	307.2	9	14.4	5	8	5	8	3	4.8
Blue Gene/P	1280	1088	16	13.6	8	6.8	16	13.6	8	6.8
Cell	1280	4096	64	204.8	36	115.2	64	204.8	6	19.2
Core 2 Duo T9900	768	2354.688	48	147.168	24	73.584	16	49.056	8	24.528
Core™ i7-980X	2304	7672.32	144	479.52	72	239.76	72	239.76	36	119.88
CSX600	6144	1536	96	24	96	24	96	24	96	24
ECA-64	2176	435.2	64	12.8	32	6.4	0	0	0	0
FPOA	6144	6144	320	320	160	160	13	0.216	0	0
Freescale P2020	256	307.2	16	19.2	8	9.6	8	9.6	4	4.8
Freescale P4080	768	1152	16	24	16	24	8	12	3	4.5
GeForce GTX 480	19456	27257.856	736	1031.136	736	1031.136	736	1031.136	128.89	128.89
GTX 285	15360	22671.36	480	708.48	480	708.48	480	708.48	30	44.28
Itanium 9350	1536	2359.296	96	166.08	48	83.04	16	27.68	8	13.84
Monarch	6150	2047.95	196	65.268	196	65.268	196	65.268	0	0
MPC7447	288	288	17	17	9	9	6	6	3	3
MPC8640D	576	576	34	34	18	18	12	12	6	6
Nvidia 9400M	1024	1126.4	32	35.2	32	35.2	32	35.2	0	0
Nvidia Ion	155549	1740.8	155549	64	155549	51.2	51.2	51.2	155549	9.6
Nvidia Tesla C1060	15360	19968	480	624	480	624	480	624	30	39
Nvidia Tesla C870	8192	11059.2	256	345.6	256	216	256	345.6	0	0
Opteron 8360 SE	1792	4480	76	190	44	110	32	80	16	40
Opteron 8439SE	2688	7526.4	114	319.2	66	184.8	48	134.4	24	67.2
PACT XPP-3c	5568	1948.8	348	121.8	174	60.9	0	0	0	0
Phenom II X6 1090T	2688	8601.6	114	364.8	66	211.2	48	153.6	24	76.8
PowerXCell 8i	1280	4096	64	204.8	36	115.2	64	204.8	32	102.4
Stratix II EP2S180	150432	75216	1079	442.39	292	122.64	186	53.196	72	10.656
Stratix III EP3SE260	217344	119539.2	1944	777.6	737	201.201	221	78.234	114	39.1216
Stratix III EP3SL340	280768	154422.4	2296	918.4	781	213.213	292	96.068	134	26.13
Stratix IV EP4SE530	443392	243865.6	2632	765.912	1352	328.536	551	132.791	371	68.264
TI OMAP-L137	64	19.2	16	4.8	6	1.8	6	1.8	2.5	0.75
TILE64	6144	4608	320	240	192	144	48	36	0	0
Virtex-4 LX100	100032	50016	240	82.56	120	29.88	120	32.88	52	9.62
Virtex-4 LX200	179904	89952	336	115.584	168	41.832	168	46.032	84	15.54
Virtex-4 SX55	58368	29184	320	110.08	160	39.84	114	40.242	43	13.932
Virtex-5 LX330T	210816	115948.8	648	300.024	324	122.472	324	115.668	109	25.833
Virtex-5 SX95T	70400	38720	488	225.944	244	92.232	180	73.8	61	21.716
Virtex-6 HX565T	369792	221875.2	1824	1036.032	912	269.952	836	255.816	284	60.776
Virtex-6 LX550T	359232	21539.2	1264	717.952	632	187.072	632	193.392	278	59.492
Virtex-6 LX760	489792	293875.2	1440	817.92	720	213.12	720	220.32	346	74.044
Virtex-6 SX475T	333888	200332.8	2128	1208.704	1064	314.944	1059	343.116	386	82.604
Xeon 7041	512	1536	14	42	10	30	10	30	8	24
Xeon W5580	1536	4915.2	96	307.2	48	153.6	48	153.6	24	76.8
Xeon X3230	1536	4094.976	48	127.968	32	85.312	32	85.312	24	63.984
Xeon® X7560	3072	6961.152	192	435.072	96	217.536	96	217.536	48	108.768