

GPGPU: The Evolution of the Coprocessor

By: Scott Hoot
Managing Partner
Sage Electronic Engineering, LLC

Abstract

Coprocessing is defined as the use of a special purpose unit to supplement the central processing unit; this concept has been a part of electronic computing almost from the beginning. Coprocessors have been used to enhance the performance of core systems since the Mainframes of the 50's and 60's. While many of the traditional coprocessor functions, such as floating-point computation, have been integrated into the modern central processing unit (CPU), new and more demanding coprocessing tasks, such as graphics acceleration have grown into mainstream requirements today.

We are now at the dawn of a new era in coprocessing technology, caused by a shift to multi-core computing. The proliferation of general purpose coprocessing hardware for 3D gaming has made asymmetric multi-core hardware omnipresent. Further, the rapidly evolving standards of the personal computer (PC) industry are setting the stage to make this technology meaningful to users across a board range of computing applications including consumer, embedded, enterprise and scientific.

This article explores the history of Coprocessing and how the evolution of 3D gaming is leading to an imminent revolution in computing.

The History of Coprocessing

A coprocessor assists the main processor by performing certain special functions, usually much faster than the main processor could perform these functions itself. Typically, this involves the coprocessor decoding and executing instructions in parallel with and under the direction of the main processor.

Examples from the Past

The coprocessor has been a staple of computing from the days of the Mainframe and has continued to be an essential element for the modern computing era. Early coprocessors supported such targeted tasks as floating-point calculations (Figure 1) and IO acceleration. In time, coprocessing evolved to encompass more advanced operations such as vector processing and video acceleration. As silicon technologies advanced, the best uses for coprocessors evolved.

The first and most common coprocessor in early computing platforms was the floating-point unit (FPU). This device was closely coupled to the CPU to perform floating-point calculations many

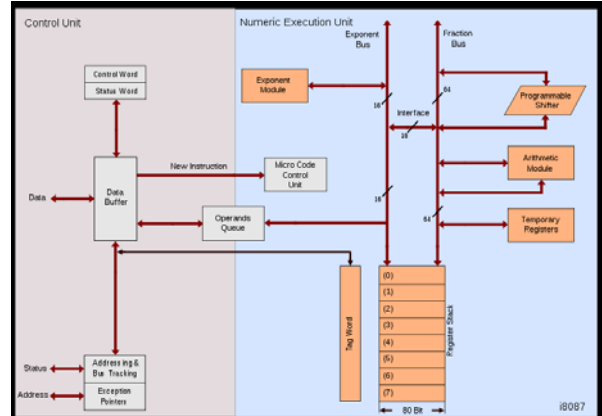


Figure 1 A – Block Diagram of an early FPU, the 8087.

times faster than the CPU could alone. The FPU was an optional component in early computers because many applications did not need floating-point capability and an FPU was a relatively expensive addition. Over time, the capabilities of the FPU grew. Eventually, the FPU was integrated with the CPU due to the decreasing cost of silicon and the performance advantages of eliminating the relatively slow external bus of previous FPUs. Today, a modern CPU is able to perform most floating-point operations in a single cycle as well as support multiple simultaneous operations though superscalar and/or SIMD operation.

Another common coprocessor in early computers enabled audio. This class of coprocessor was loosely interfaced to the CPU via a universal bus structure. It used techniques such as FM modulation and Wave Table Synthesis to minimize the data required for early computers to produce audio of acceptable quality. Overtime, the features of audio coprocessors improved. Digital Signal Processing (DSP) features for advanced audio manipulation became the mainstay. However, the audio coprocessor is less important today due to the rapid performance advances in the CPU and other coprocessors.

Processing of visual data is another task that has almost always required the use of a coprocessor to achieve acceptable performance. Graphics coprocessors were loosely coupled to the CPU in order to enable rudimentary displays.

These coprocessors were used to reduce CPU load by translating binary characters and then writing sequences of pixels representing the different characters into memory. These devices slowly evolved to support different character sets and then colors. Next, graphics coprocessors evolved to support 2D acceleration. At this stage, the video coprocessors had advanced to where they enabled specialized movement and manipulation on a per pixel basis. Finally, there was the 3D revolution. This innovation in video processing enabled detailed pixel manipulation with limited management from the CPU. Concurrently, video decode acceleration was introduced to reduce the load on the CPU when decompressing video streams. Today, the 3D revolution is ongoing as the performance of video coprocessors is advancing faster than CPU technology.

Direction for the Future

The focus of the computing industry has shifted in recent years; rather than concentrating on maximum general-purpose performance, the focus has shifted to developing greater efficiency. Ever increasing performance is still an issue, but it is no longer acceptable to achieve these goals at the expense of greater power and unit cost. This has led to a newfound industry awareness of coprocessors and how these devices may be used for general-purpose tasks.

In 1982, the DSP became the first general-purpose computational coprocessor. Over the years, the DSP has changed with the needs of the market. They are used for processing audio, communications, graphics, video and other data. These devices have evolved to encompass heterogeneous multiple cores capable of processing many gigaflops per second. As in the past, these



Figure 2 – The Toshiba Qosmio G55 Series includes a DSP like ‘Cell’ processor to give a 10-fold speed boost for video encoding tasks and enable features such as facial and gesture recognition.

devices are being employed alongside CPUs to enhance system performance. These devices enable such computationally intensive tasks as image library analysis, facial recognition, and video transcoding. However, the DSP is hampered by a lack of standards, which has limited its adoption rate. Ultimately, it is expected that this device class will be largely superseded by future coprocessor technologies.

The modern graphics processing unit (GPU) is the result of years of evolutionary development in video coprocessing. Driven by the demands of consumers for gaming and video, these devices have evolved to support ever more complex 2D and 3D graphics, as well as to accelerate decode of compressed video streams. Today, the GPU is infiltrating a host of embedded devices including cell phones and game consoles in addition to being omnipresent in PCs. The performance, standardization, extreme market penetration and general purpose compute capability of GPUs has set the stage for this device class to become the next revolution in coprocessing.

The General Purpose GPU

So, why will the GPU cause a revolution in coprocessing? There are a few simple reasons. First, graphics processing cores embrace a massively parallel computational model to achieve astonishing levels of performance. In the bargain, the GPU can attain significantly higher power for performance ratios. Next, the power and performance advantages of recent GPU generations have meshed with generic programmability. Third, usability of the GPU has drastically improved due to new tools that ease development. Finally, these devices are ubiquitous. This combination of factors has already led to the revolution that is now referred to as general purpose GPU (GPGPU) computing.

Performance

The first and most comprehensible reason GPGPUs are gaining favor is the massive performance potential of these devices. Much of this promise comes from the massively parallel nature of these machines and the incredible operation rate this permits. In addition, the GPGPU provides unrivaled memory bandwidth, which allows the compute cores to achieve their potential. However, the architecture of the GPGPU comes with tradeoffs that must be clearly understood.

Parallelism

The processing capability of the GPGPU is derived in large part from the large number of processing units it implements. Contemporary

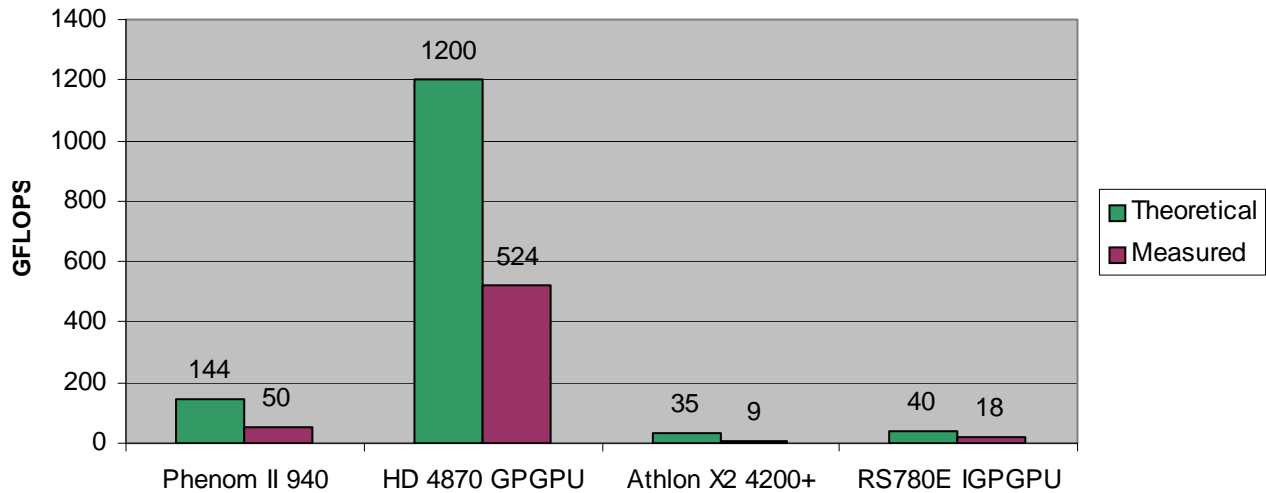


Chart A – Comparison of AMD CPU and GPU floating-point performance levels.

GPGPUs can include between 8 and 240 parallel processing cores with complexity ranging from single to 5-wide instruction pipelines. This enables a theoretical performance that can range from 40 GFLOPS to 1.2 TFLOPS on a single piece of silicon today. While achievable performance is much lower than the theoretical maximum, the potential is still significantly higher than can be achieved with modern CPUs.

Chart A depicts the relative performance capabilities of four current products available from Advanced Micro Devices (AMD). Remarkably, the RS780E integrated graphics processor has twice the performance of a moderate capability 2.2GHz dual core CPU in executing single precision floating point operations. Even more amazing is the realization that a single high performance GPU completes an order of magnitude more GFLOPS than the latest 3.0GHz quad core CPU from that same vendor. Moreover, while the methods of comparison are certainly not equivalent, the performance potential is clear. The GPGPU can enable higher system level performance than a CPU alone can offer.

Memory

Another significant factor in computational performance is memory bandwidth and latency. GPUs utilize large register sets and segmented local memory rather than caches as used in CPUs. This architecture calls for massive amounts of memory bandwidth to achieve peak performance, which GPUs provide in abundance. Chart B depicts that once again the AMD HD 4870 GPGPU is capable of streaming almost 10 times the amount of data per second than the latest generation AMD CPU. This underlying capability allows the GPGPU to live up to its potential.

Input output (I/O) bandwidth between CPU and GPGPU memory spaces is another important aspect of system performance. In current systems, a PCIe bus provides 4.8GB/s of full duplex bandwidth to exchange data between the CPU and GPGPU.

Performance Tradeoffs

Using a GPGPU to offload computations from the CPU is loaded with compromise. First, it must

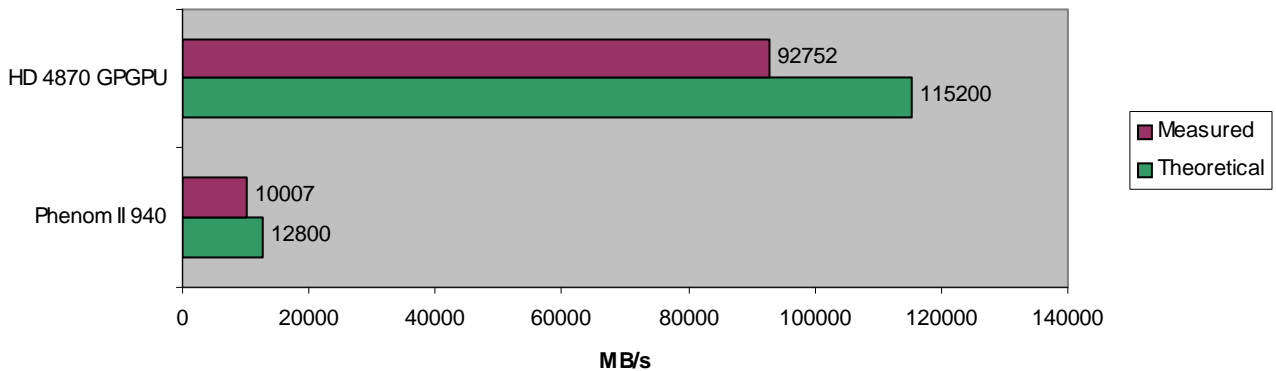


Chart B – Comparison of AMD CPU and GPU memory performance levels.

be understood that a CPU is still required for control of the GPGPU(s). However, this can become an advantage in that the CPU is available for use in parallel to the GPGPU. This can dramatically increase the potential aggregate system performance by allowing each device to handle the task it performs best. Second, the GPGPU and CPU do not share coherent access to memory. Data must be moved to GPGPU memory for calculation via the “slow” I/O bus. Then again, this also has the advantages of increasing memory bandwidth and allowing a single CPU to control multiple GPGPUs. Today, up to four GPGPUs can be operated in parallel. Besides, the I/O bandwidth is actually high in comparison to more traditional distributed computing solutions. Finally, the GPGPU was designed to support single precision floating point or 32-bit integer computation efficiently. Therefore, applications requiring greater precision may be limited.

Power

Another advantage of the GPGPU is that it offers significantly greater performance per watt than CPUs achieve. Using dramatically simpler data pipelines and simplified memory management techniques, the GPGPU can achieve many times the performance per watt of the latest generation of CPUs. Comparing the fastest CPU to the fastest GPU currently offered by AMD, there is no contest. In fact, Chart C shows that the performance per watt of RS780E IGPGPU is approximately seven times better than the best CPU offering.

Programmability

An increasingly flexible programming model is the crucial advance that enabled the general-purpose use of the GPU. Today, the GPGPU can be programmed such that virtually all of its computational resources are utilized. Under CPU control, GPGPU resources can be harnessed to

carry out many different execution requirements. In addition, memory movement between CPU and GPGPU and streaming data manipulation may be offloaded to hardware. These are the advances that transformed the graphics processor into the GPGPU.

Ubiquity

Next, there is the undisputable omnipresence of the GPU. These devices have been a mainstay of PCs for well over a decade with hundreds of millions of units shipped. Today, the GPGPU is migrating into the integrated graphics of PC and it is eventually expected that this capability will be in everyday devices such as cell phones. The sheer volume of GPUs made available by these markets virtually assures that the GPGPU will be a dominant resource for future computing challenges.

Usability

One last reason that use of the GPGPU will expand is the investment being made in its usability. Data parallel programming is a difficult task and characteristically requires customization when migrating between devices. This has traditionally been mitigated with GPU programming by graphics specific programming application programming interfaces (APIs), but these languages offered little support for other applications. Eventually, APIs were developed that can enable familiar C-like programming constructs to be applied to parallel processing. The initial attempts enabled an explosion of new development, but were tightly coupled to specific GPU architectures. All that has recently changed with the release of the OpenCL 1.0 specification. This new language bridges the gap between graphics and general purpose computing as well as unifying the industry with open standards. Thanks to all these advances, GPGPU acceptance is nearing critical mass.

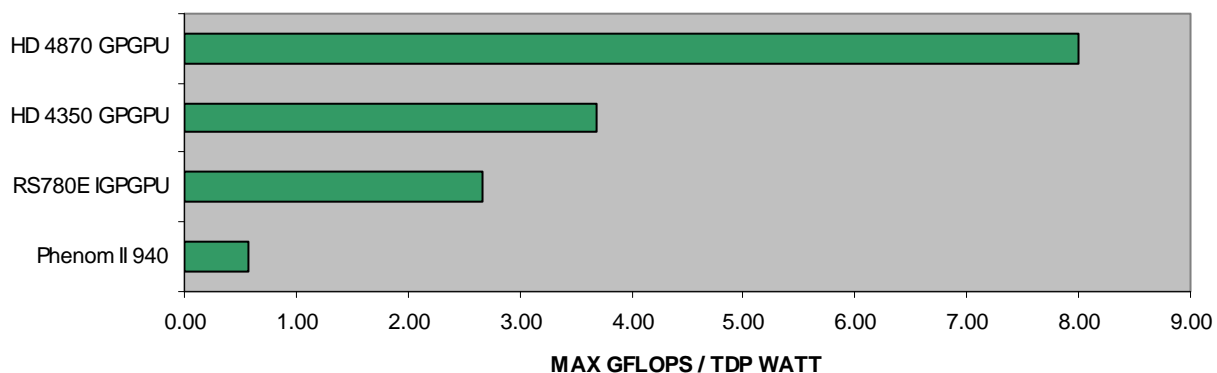


Chart C – Comparison of AMD CPU and GPU performance per watt levels.

Reasons for the GPGPU

It seems that no matter the advances in computing performance, there is always a need for more performance. Early interest in the GPGPU was mostly confined to high performance and scientific computing, which seeks performance at any cost. The efforts to support GPGPU for these applications demonstrated the potential benefits of this new coprocessor class. Today, interest has exploded to encompass most areas of computing including enterprise, embedded and consumer applications.

HPC

The high performance computing (HPC) market segment has always sought to achieve the maximum performance possible. Developers commonly expend tremendous effort to achieve moderate gains in performance. However, the GPGPU has expanded developers' horizons by enabling a new paradigm with tightly coupled massively parallel computation units. Today it is possible to have thousands, even tens of thousands, of threads scheduled simultaneously on a single system.

The immense computing capability of the GPGPU has been deployed in many HPC fields, such as bioscience, financial modeling, weather prediction, and energy exploration. The investment has allowed increasingly complex algorithms to be processed with greater immediacy and by a broader segment of scientific community.

Enterprise

The broader based enterprise computing market has also demonstrated its voracious appetite for computing performance. Further, enterprise computing is interested in reducing its performance per watt to reduce the operational costs of its server farms. The GPGPU meets both these objectives with new levels of performance per unit as well as per watt.

Some common enterprise applications demonstrate the data level parallelism that the GPGPU is so adept at accelerating. Database query and network monitoring are both based on parallel lookup and compare algorithms. Place and route as well as simulation in computer-aided design is another enterprise computing arena where the GPGPU holds promise. Processing tens of thousands of potential solutions in order to find the best fit, these applications are capable of utilizing the GPGPUs' natural strengths.

Embedded

The diverse requirements of embedded computing may not seem to be a good fit for the GPGPU, but this could not be farther from the

truth. Embedded applications typically value the ability to customize platform operation while maintaining low power and small size. The GPGPU allows the embedded developer to deploy more computational power with a smaller physical and thermal envelope than could be achieved with other solutions. Applications as diverse as digital signage, medical imaging and artificial vision are all ready to leap to GPGPU computing based on the strengths of this new coprocessor class.

Consumer

Consumer applications are perhaps the area where the GPGPU can make the biggest change in the computing status quo. The universe of consumer computing is characterized by extended periods where computing resources sit idle followed by infrequent intervals of extreme performance expenditure. In the process, tremendous amounts of resources are wasted. The GPGPU is an excellent solution to this dichotomy.

Consumer applications drove the creation of the GPU. First, the GPU was used to support graphics, and now the GPGPU is being extended to handle gaming physics calculations. In addition, it is expected that new applications, such as image recognition, picture scrapbooking and video encoding will soon experience the benefits of the GPGPU.

The Future of the GPGPU

The future of the GPGPU appears to be primed for rapid advancement. Increasing use of GPGPUs will be in large part driven by significant performance and feature improvements as well as an increased range of product offerings. Meanwhile, improved tool sets and more advanced operating environments will enable expanding levels of GPGPU adoption.

Raw GPU performance has increased at approximately twice the rate of their CPU brethren. This is attributable to three factors. First, the GPU continues to increase the speed of its execution units and memory interfaces. Second and more importantly, the number of execution units is expected to consistently grow by about 150% per year. This translates into a rate of improvement about twice that predicted by Moore's Law, which shows no signs of slowing down. As quick as the performance is increasing, the range of offerings is growing equally fast. This variety of performance levels assures that the right fit for each application will be accessible.

GPGPU features are also advancing. While these may seem unimportant, the small features being implemented are likely to pay large dividends. The first feature being touted is the shift

to IEEE 754 floating point compliance, including improved double precision support. This minor change in hardware opens the door to many more applications that require computational accuracy. The second feature is the expected integration of the CPU with the GPU, which in turn paves the way for cache coherent memory access by the GPU. This would remove many of the tradeoffs in GPGPU computing today by reducing transfer overhead from the CPU and enabling a shift to cache coherent memory access by the GPU.

Finally, the tools and operating environments for GPGPU computing are expected to see rapid near term improvements. The release of the OpenCL 1.0 specification was a first step that has already been supported by the release of the first OpenCL capable driver. However, the real advances will come with the release of OpenCL compilers from the major graphics vendors. Even more interesting is the forthcoming operating system support for OpenCL proposed for Apple's OS X release codenamed Snow Leopard and similar plans for future operating systems. These advances will ultimately make general-purpose deployment of the GPU as ubiquitous as the CPU is today.

The future is bright for the GPGPU. GPU performance and features are continuing to advance rapidly, while the barriers to deployment are falling. It is clear that interest in the GPGPU has reached critical mass and the benefits are becoming quite tangible. Soon the GPGPU will be the coprocessor used by everyone.