OPEN UP A WORLD OF OPPORTUNITIES

AMD and Quixant continue to de-mystify the technology that powers the modern slot machine in the second part of our technical feature

So the saying goes, "there is no such thing as a free lunch" and in many cases accepting this clichéd assertion is both reasonable and prudent and will protect against future disappointments.

On the face of it, the electronics behind video slot machines and other gaming machines has appeared to somewhat challenge this claim – relentlessly, the latest technology has offered ever more processing power, more vibrant graphics, more immersive sound, a growing diversity of features and at seemingly lower price points.

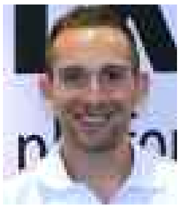
The key driver behind these radical enhancements in the price to performance/features ratio have been driven solely by improvements in the architecture around which the systems are designed, shrinkage of the components enabling greater integration, higher processing speed and growth in volume of units sold.

There are, of course, major challenges which have arisen along the way.

In the last issue, we explored the evolution in some of the technology which beats at the heart of slot machines. From the days of purely mechanical devices, through electromechanical reels with highly specialised bespoke electronics to the latest multimedia-rich video slots, the technology required to power the machines on the casino floor has risen to the challenge wholeheartedly.

One of the major dilemmas currently faced by the likes of Advanced Micro Devices (AMD), Intel and Nvidia is how to continue to evolve their technology to meet the functional demands of the latest machines when faced with growing needs for moderate power consumption, particularly in today's environment of eco-awareness and rising energy prices.

The introduction of special purpose Graphics Processing Units (GPUs) – dedicated processing resources designed specifically to accelerate computation of realistic 3D environments and high



JON JAYAL, General Manager, Quixant.

"From the days of purely mechanical devices, through electromechanical reels with highly specialised bespoke electronics to the latest multimedia-rich video slots, the technology required to power the machines on the casino floor has risen to the challenge wholeheartedly."

"OpenCL enables users to draw even more performance of out their APU-based hardware designs by leveraging the dozens and even hundreds of compute engines of the GPU, where parallel data can be processed more efficiently," adds **KEVIN TANGUAY,** Senior Manager, AMD Embedded Solutions Digital Gaming.

definition motion video – to slot machines has revolutionised the approach game developers take in crafting game graphics. GPU technology has progressed profoundly such that these high-performance, multi-core, highly parallel processing elements are being used to augment the multipurpose Central Processing Unit (CPU) for general purpose computation, beyond their traditional graphics processing role. This evolution has brought about the concept of "heterogeneous computing", embodied by AMD's Accelerated Processor Units (APUs), which combine CPU and GPU elements.

"To take maximum advantage of the APU approach, in which software code is executed via the most efficient processing element in the system, requires software engineers to revise their approach to game development. The Open Computing Language (OpenCL) was originally developed by Apple, AMD and several others and is an open and royalty-free parallel computing framework designed to enable GPUs and other coprocessors to work in tandem with the CPU without drawing boundaries between them" explains Andres Garcia, Senior Embedded Software Engineer at Quixant. "Code written in OpenCL is agnostic to the manufacturer of the processor on the target platform, how many cores it contains or what features it supports. The created code will compile and execute on any OpenCL supported hardware. An OpenCL program can run on a wide range of systems, from slot machines to cell phones to nodes in vast supercomputers. This is one of the reasons why OpenCL is so important and has the potential to transform the software industry."

"OpenCL enables users to draw even more performance of out their APU-based hardware designs by leveraging the dozens and even hundreds of compute engines of the GPU, where parallel data can be processed more efficiently," adds Kevin Tanguay, Senior Manager of AMD Embedded Solutions Digital Gaming.

The OpenCL framework defines a language specification based on C programming language, a platform Application Programming Interface (API) and a runtime API. In addition, developers have created OpenCL ports for

Java and Python, but the standard only requires that OpenCL frameworks provide libraries in C and C++.

An application running on a heterogeneous platform generally needs to perform the following basic set of steps:

1) Discover all the devices available on the platform;

2) Retrieve information about the features available for each device;

3) Create the functions (kernels) that will run in the platform;

4) Configure the memory involved for the computation;

5) Execute the kernels; and

6) Output the results.

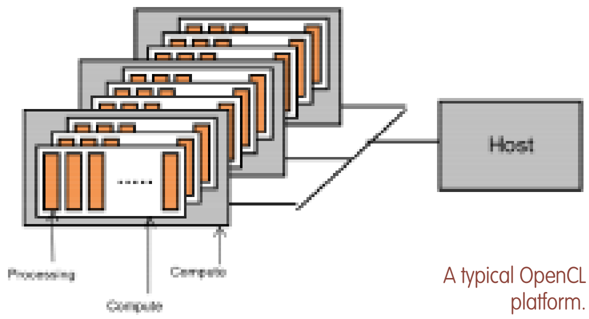
OpenCL allows developers to accomplish these steps through a series of APIs and a programming environment for the kernels.

The OpenCL platform consists of a central processor host (any computer running an

operating system) connected to one or more OpenCL Compute Devices (e.g. a GPU). Each OpenCL device is divided into several Compute Units which are further divided into a number of Processing Elements. Computations on a Compute Device occur within the processing elements, after identification of the most efficient element to execute a particular calculation. The current generation of GPUs may contain several hundred processing elements, all capable of performing operations in parallel.

Execution of an OpenCL program occurs in two parts: kernels that execute on one or more OpenCL Compute Devices and a host program that executes on the central processor host. The host program defines the context for the kernels and manages their execution.

The core of the OpenCL execution model is defined by how the kernels execute. Kernels are similar to a C language function which can be data- or task-parallel. The principle behind the execution model is to break a large computation down into several smaller, more straightforward so called work-items



A typical OpenCL platform.

which can be executed in parallel. These work-items are distributed to different processing elements in the Compute Devices under the control of the host. Such an execution model is remarkably similar to the way many CPU architectures have traditionally executed instructions, but the ability of the OpenCL architecture to access all processing elements in the system (regardless of whether they are housed in a CPU, GPU or APU) is where the power of OpenCL can be harnessed.

So what does this all mean? Well the key benefits can be adequately summarised into efficiency, flexibility and cost savings.

"To take maximum advantage of the APU approach, in which software code is executed via the most efficient processing element in the system, requires software engineers to revise their approach to game development.
ANDRES GARCIA, Senior Embedded Software Engineer, Quixant

By taking maximum advantage of all the various processing resources available in an enabled computer platform, OpenCL allows the same rich visual and audio experience to be achieved, but on a platform which uses less power, generates less heat and in some cases can be operated without the use of fans. All of these should lead to improved reliability.

OpenCL also gives developers the flexibility to write games which are portable between manufacturers of different OpenCL hardware and to the next generation of OpenCL platforms. This maximises the return on investment into game development by prolonging the useful life of the game code.

These key advantages translate into cost savings both in terms of the ability to select lower cost hardware without jeopardising the player experience and in terms of the price of on-going game code development.

To top it off, all these benefits are facilitated by an open and royalty-free development framework. So it seems you really can have your cake and eat it.