Future proofing fun

How fgpas could serve as platforms for next generation game consoles.

*By **Bart Borosky**.*

Game console designers balance many requirements in their efforts to create a fully featured, reliable low cost system. However, the console may not be a commercial success unless it offers high definition graphics, internet connectivity, motion sensing controllers and IP based streaming video content – all for an affordable price. To keep costs low, a platform based approach is common.

A typical game console platform includes the cpu, graphics processor, storage and memory, communications, I/O and user interface. At its heart are typically a few large asics or assps. For example, iSuppli claims the Sony PlayStation 3 is based on a seven core Cell based cpu (asic), an Nvidia graphics chip set (assp), XDR memory and a flexIO processor interface.

With fpgas replacing asics in other applications, perhaps now is the time to consider one in the design of next generation game consoles.

Platform based design

FPGAs help system designers maximise the savings and flexibility of a platform based approach. The following design guidelines are important for games consoles.

- ***Requirements***. Because these vary by region, programmable support for multiple standards and formats speeds time to market.
- ***Low cost, reliable design***. FPGAs can handle overall system I/O interface, control logic and bug fix integration, whilst supporting bridging functions from the asics/assps that interface to the hard disk and internal memories. Meanwhile, customised user interface logic mitigates the risk in a new product.
- ***Security***. Secure programming during final test and in the field enables standards support, incremental upgrades in coprocessing algorithms and easy testing of new functionality.

In general, three elements need to be evaluated when selecting programmable components for a platform.

- ***Embedded dsp functionality***. Performance, look up tables and block ram for coprocessing functions. These can take advantage of parallelism, tailoring functionality and performance to specific applications at optimised cost.
- ***Connectivity***. LVDS and serdes support for high speed interfaces, such as HyperTransport, PCI Express or Serial RapidIO, and
- ***Security to protect IP***.

Figure 1 shows an fpga solution that meets these requirements. The fpga is connected to the main cpu, the graphics processing chips, the user interface and other key assps such as wireless Ethernet. The fpga block performs coprocessing functions and hardware acceleration and has the custom logic necessary for future functionality – either for the user interface or elsewhere in the console.

Although low cost fpgas typically operate at less than 300MHz, high dsp throughput (63,000MMACs) can be achieved by executing serial functions in parallel. For example, an application requiring a 32 tap FIR filter with a sample rate of 100Msample/s, would require a general purpose dsp with four multipliers running at 800MHz. However, an fpga with 32

Illustration: Jürgen Ziewe

multipliers could implement this same filter at 100MHz.

The flexibility to take advantage of parallelism in fpgas is beneficial for memory access. On chip distributed memory can be used to build small high performance scratch pads, whilst on chip embedded block memories allow larger high performance memories to be created. Off chip memory, such as DDR dram, can provide large, relatively high performance memories.

"Using an fpga (might help) to 'future proof' the game console design."

***Bart Borosky**, Lattice Semiconductor*

FPGAs interface with the gpu and cpu via high speed interconnects such as HyperTransport, PCI Express or Serial RapidIO. Whilst HyperTransport is a source synchronous interface, PCI Express and Serial RapidIO use clock data recovery (cdr). For source synchronous interfaces, a differential clock is provided with the data from the transmit side, making it relatively easy to capture the data at up to 700Mbit/s. Beyond that, skew between the data channels becomes a significant part of the data valid window, especially for wide data channels.

Some fpgas have an auto alignment capability that manages data skew for high speed source synchronous interfaces, helping them to achieve LVDS data rates beyond 800Mbit/s – up to 2Gbit/s in some cases. There is no extra differential clock provided for cdr. Instead, the clock is included with the serial data packets and must be 'recovered' from the data. Consequently, clock skew is not an issue; data packets are sent serially at high speed using serdes and a cdr receiver will phase lock on the data itself to derive the clock.

For cdr, serdes jitter is the main issue at high speeds. Whilst some jitter is random, most jitter is deterministic and produced by other signals, by components switching close to the data signal, by the nature of the data pattern or by the unbalanced rise and fall time of the transitions.

Secure configuration

Game consoles are frequently targeted by hackers, so an fpga should not introduce additional vulnerability to the system. SRAM based fpgas typically draw configuration data from an external non volatile memory source and this source is protected in some fpgas with a built in 128bit AES decryption engine. The AES decryption key is typically programmed using one time programmable (OTP) fuses. Because OTP keys are non volatile and cannot be modified, hackers cannot 'reverse engineer' the functions inside the fpga.

In addition, fpgas help designers bridge assp functions to the graphics processor or system processor. Customised user interface logic helps differentiate game consoles from competitive products. Both functions can be bridged/controlled with an fpga.

Managing uncertainty

The market demands that all companies designing game consoles create manufacturable designs with more features at the lowest possible cost. Failure to embed important features or flexible interfaces can mean no demand for the product.

There is a major opportunity to place small technology 'bets' into the game console system – either to support future changes or to try out new features in hardware. A secure fpga that supports PCI Express, HyperTransport or Serial RapidIO, LVDS and dsp makes these bets possible.

The LatticeECP2M fpga family supports many of these requirements, including AES encrypted bitstream, PCI Express, LVDS and dsp, as well as programmable logic, to support multiple interface standards and custom functionality. Many dsp IP cores and MATLAB Simulink support are also available to facilitate design. ■

Author profile:

Bart Borosky is a product marketing manager with Lattice Semiconductor.

Figure 1: An fpga implementation for games consoles