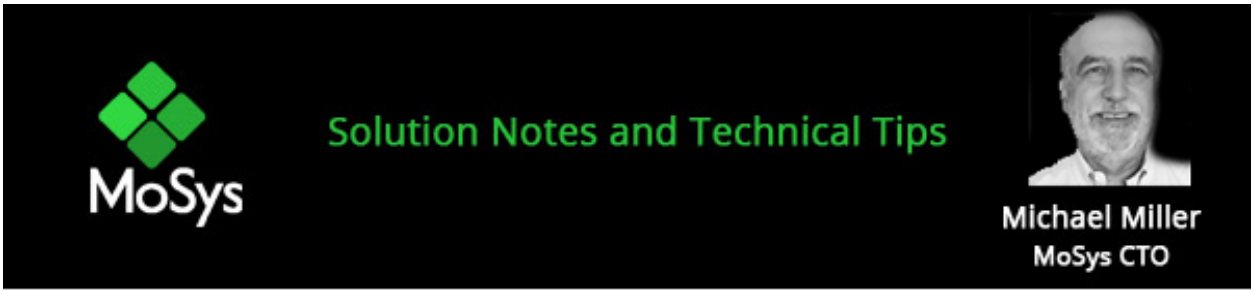

SOLUTION NOTE #1004 How Bandwidth Engines (Accelerator Engines) Complement FPGA BRAM, uRAM, or M20K

TOPIC SUMMARY:

Overview

As system performance demands continue to increase, memory architecture has become a more critical element in the design to support today's needs and tomorrow's demands.

- How much available internal FPGA memory?
- With a single chip high-speed low latency 1Gb serial attached SRAM, what more can be done with external SRAM attached to the FPGA?
- What is the tradeoff between using ***both*** Internal FPGA ram and External SRAM?
- When to use a DRAM based memory?

The location of the memory, the costs, performance tradeoffs and ease of design all must be considered. In this note we look at three basic types of memory to consider.

- Internal FPGA memory
- Attached DRAM memory
- External SRAM memory

Situational Analysis

Internal FPGA Memory

It is easily understood that it is advantageous to use resources (in this case memory) that is close and localized to a source of a data request. This has been a strong forcing function for memory resources to increase in both ASICs and FPGAs. Even with the on die real estate becoming increasingly valuable, chip manufacturers are dedicating significant area resources to memory.

- Xilinx
  - BRAM - a set of smaller blocks of ram which are approximately 36Kb each and are distributed across the die. It's total resource of approximately 95Mb in a larger device.
  - uRAM - larger blocks of RAM (approx. 288Kb each) that are in columns of the die. Its total resource being approx. 360Mb in the larger FPGA devices
- Intel (Altera)
  - Utilizes M20K blocks which are, as the name implies, blocks of 20Kb of ram which are distributed across the die with a total resource of approx. 230Mb in a larger device.

Each of these have some restrictions with regards to routing, timing and speed when the desire is a unified large block of memory. This approach may increase the design effort, but it certainly does benefit in the area of latency and convenience if no external memory resource is needed.

When more memory than is available on the FPGA is required, the alternatives are external SRAM or attached DRAM (like DDR or HBM). A fundamental tradeoff between external SRAM and DRAM is that SRAM has lower latency but is limited in capacity while DRAM has much higher storage capacity but much longer latency.
There are many applications that would benefit by utilizing a complementary memory architecture with the available FPGA memory. Additional SRAM and DRAM can achieve a higher performance target than internal resources alone could achieve.

HBM Memory

The FPGAs vendors have been spending the last few years incorporating HBM (High Bandwidth Memory) into their offerings using some form of interposer to the die of the FPGA. The HBM, although not a part of the FPGA die, is connected internally in the package with the FPGA die. These are large DRAM chip stacks that provide large density and high bandwidth memory resources to the FPGA without impacting the valuable FPGA die resource. However, there can be potential drawbacks including:

- Cost
- Thermal considerations

- DRAM as Base Technology
    - Base technology is still DRAM so longer latency
    - Marginalized performance when accessing the memory in a truly random pattern
    - Refresh issues – can impact access rate
- Manufacturability
    - The added steps of designing, assembling, testing and yielding multiple complex die in one package
- Product Longevity

What we can do next is address each of these potential drawbacks.

Cost

As with any new or extremely complex technology there tends to be a cost associated with its use. The same seems to be true of HBM. The development of the product is predicated on the stacking of multiple DRAM die to help keep the overall footprint of the memory as small as possible and fitting it in a package with other devices. In order to accomplish this, the vendors such as Samsung and SK Hynix utilize TSV (through silicon vias) to stack die on top of each other. At the same time high density DRAM products also push to use leading edge process technologies to achieve the density their customers are requesting. The combination of using the most advanced technology available on both these efforts in a single product leads to additional costs.

Thermal Considerations

In looking at thermal considerations it is important to realize that DRAM itself has the issue of the storage mechanism being a capacitive storage. This allows for the small size but results in the need to REFRESH the cell due to leakage of the charge off the capacitive storage element. A second order issue is that heat has an additional negative effect and causes the charge on the cell to leak even faster resulting in a condition that requires more frequent refreshing of the cell as the junction temperature increases. Many DRAM devices can tolerate a junction temperature of approximately 85C and will function at 95C to as high as 100C with additional refresh.

When a device that requires additional support at higher temperature is placed next to a device (such as a high end FPGA) that can dissipate 100W or multiple 100s of watts, it can require additional measures to insure that the HBM stacks are properly cooled to maintain the integrity of the device. The result may be a larger heatsink, additional airflow or in an extreme case some form of liquid cooling. Each of these measures add cost and complexity to the system design.

DRAM as Base Technology

When looking at the design criteria of an HBM stack, the goal is to provide superior memory density with significant access bandwidth for large bulk storage application needs. The result is stacks of DRAM devices. This allows for large density per mm^2 of die area.

Routing to the stacks is addressed by making the connection using die layout criteria over an interposer structure - which is essentially an additional piece of silicon that supports multiple extremely wide buses - to provide the increased bandwidth. The base technology remains a DRAM with good burst performance, but unimpressive random-access performance when compared to SRAM technologies. There is also the need for the DRAM cell to be refreshed to ensure data integrity. This impacts the ability to randomly access locations that may be in a ROW that is being refreshed.

Manufacturing Complexity

Assembly of the newer MCM (Multi-Chip-Modules) is a technically challenging process. However, with industry drivers like IBM, Intel, Samsung, Amkor, TSMC, etc. the technical feasibility of this process has been steadily improving from straight interposers to EMIB (Embedded Multi-Die Interconnect Bridge). The feasibility hurtle has been quickly addressed. The remaining issue, at least as of this writing, is the real cost of assembling and yielding multiple very complex die. These are not insurmountable issues but they are complex enough that they will always add cost due to issues of die yield, die process and die power which only get multiplied as many die are assembled in a single package.

Product Longevity

The last issue we will discuss with regards to HBM is longevity of supply.
Since the base structure is a DRAM and it has become known that the driving force behind DRAM is to continually move into the latest process technology in order to accommodate the ever-increasing desire for more and more storage. This has historically also involved some changes in device pinout, timing, power requirements, and die size, each of which could impact the use of a present day design a couple of years down the line. This can cause concern for systems that have historically required system longevity in the 10-year expected lifetime.

FPGA Memory Architecture Approaches

The hope of vendors is that the combination of on die SRAM-like memory (uRAM, BRAM, M20K RAM) along with the appendage of HBM will address the bulk of customer needs. Certainly, the benefits can be significant when you have localized memory resources and if the available resources can meet density, speed, and access requirements of the system. This holds if the size of the memory does not interfere with other resource requirements of the FPGA, and the size of the memory needed allows the RAM resource to route cleanly.

FPGA resources tend to be spread across the die, which is ideal for allowing multiple logic functions to easily access the resources without having to route traces across the die, but this often can cause routing issues for other functional blocks in the FPGA. What MoSys has experienced, both by developing its own code that utilizes the FPGA memory resource and from feedback from customers doing FPGA designs is that there is a real measurable impact in performance when attempting to utilize the FPGA resources as a large unified memory block. It is still possible to develop larger blocks of memories but there are potential performance tradeoffs.

In fact, the latest product offering from MoSys is IP called the GME (Graph Memory Engine) which is structured to be extremely efficient in walking through graph structures that reside in memory. This is extremely useful in applications such as LPM, Regular Expression, DDoS, and other algorithm structures such as TCAM (the GME IP supports Millions of Rules without causing memory explosion). In each of the above applications there is significant benefit to having large memory structures that are accessible in a

random pattern. This can cause resource and performance tradeoffs within the FPGA by forcing a tradeoff of blocks of logic that benefit from the availability of as much SRAM (Random access) memory as possible to increase performance and throughput.

MoSys SRAM … Results of Collaboration

It is certainly understandable that an FPGA can't provide all the resources that every application is looking for. Because of this, MoSys has partnered with Intel and Xilinx to use MoSys Accelerator Engine devices as a complementing device to their FPGAs. As partners, MoSys has been working with all the FPGA vendors to ensure interoperability and achieve SRAM type performance.

- GCI serial Protocol – A Bandwidth breakthrough
    - MoSys has developed a unique high speed, light weight serial protocol called GCI (GigaChip Interface), that ensures that the two devices can properly communicate over the SerDes links using the available SerDes links. In FPGA applications that are resource constrained, customers can offload the FPGA by taking advantage of the high reliability and low latency that the GCI protocol offers while utilizing a minimum amount of FPGA (LUTs or ALMs) and a small footprint of routing resources on the PCB.
    - Devices are available that can take advantage of links that run from 10Gbps up to 25Gbps.
    - Interface resources can require as few as 16 pins, typical systems use only 32 Pins, with a maximum number of pins per MoSys device utilizing 64 pins
    - GCI enables an easy way to expand much needed high-speed SRAM while not burdening the board designer with the typical issues encountered when designing in external traditional parallel IO type memory like QDR. By taking advantage of the serial GCI interface, PCB board layouts are much less complex, routing and signal integrity are improved, and board space is reduced.
    - MoSys provide RTL Memory Controller reference design resided in the FPGA and controls the SerDes signals to the memory so they are transparent to the user application. This provide benefits of high

bandwidth SerDes to achieve high data bandwidth over only 32 FPGA pins with a common RTL memory interface.

- MoSys Bandwidth Engine single chip devices are 4-8x the capacity of the latest QDR devices:
    - 576Mb single chip
    - 1.1Gb single chip
- High speed of tRC 1.2-3.2ns
- Signal Integrity with onboard Auto-Adaptation

If your design would benefit by having resources beyond what is available on the desired FPGA device to implement functions like oversubscription buffers, statistics, fast lookup tables, flow caches, or any number of logical blocks that require high speed and low latency, the MoSys accelerators can be an easy way to extend and complement the FPGA's (or ASIC's) resource. MoSys devices directly address all the above-mentioned issues that localized resources address:

- Low Latency – Initial latency as low as 18 FPGA clock cycles with follow-on results available every clock cycle after the initial pipeline.
- Utilizing SerDes allows the freedom to place the memory devices anywhere on the PCB enabling an easier thermal board profiling
- Easy to design with low signal pin count
- Signal integrity - by utilizing a CEI compliant SerDes with auto adaptation, the need to be concerned with extra signal tuning components is eliminated
- Manufacturing/Assembly - The device uses only standard single die processing and assembly
    - Traditional full functional testing of a monolithic device
- Low cost
- Low power
- Product Longevity – MoSys has a commitment to providing devices for the lifetime of systems for all industries including markets like Telco that need 10year product lifetime support

Summary

Memory architectures are more important than ever, and there are options today that were not available in the past. These tradeoffs affect not only the hardware architecture but allows software partitioning to take advantage of different memories.

FPGAs support a variety of memory technologies. HBM provides huge amounts of capacity and Bandwidth but is expensive and can have marginal performance in random access applications. SRAM are the desired choice when low latency and high random access is needed and MoSys provides a single chip SRAM device with over 1Gb of capacity. MoSys devices have been shown to provide a highly beneficial complement to the resources found on FPGA (or ASIC) silicon.

The MoSys QPR4/8 memories and the Accelerator Engine family of products are in full production and they have been reliably shipping for close to 10 years with excellent reliability. MoSys would like to explore with you how these products can Accelerate your application.

MoSys is always interested if you found the ideas presented in this solution note were helpful, so any feedback would be greatly appreciated and will support us in what future topics and data will be addressed in future solution notes.

If you need to free up resources on your FPGA or would like more flexibility in your FPGA part selection options, please contact MoSys and we can do a memory architecture design tradeoff review with you. Contact: AppContact@Mosys.com for a memory architecture discussion! Email us and we will arrange to have one of our technical specialists speak with you. You can also sign up for our Newsletter. Already convinced? You can request a quote from sales. Finally, please follow us on social media so we can keep in touch.