

HyperRAM™ as a low pin-count expansion memory for embedded systems

Bobby John, Principal Product Marketing Engineer,
RAM Memory Solutions

Rapid advances in microelectronics are driving mega trends across industries, creating a need for new technologies and optimized devices with better performance. With large volumes of data being made available due to the increasing content of electronics in automotive, industrial, smart home and IoT devices, there is a requirement to seamlessly process and render information. Application platforms are also expected to be scalable to cope with the burgeoning demand for better user experience and functionality. In view of this, MCU vendors are developing new-generation MCUs with higher performance and lower power consumption to meet system requirements.

While the MCU/FPGA plays a pivotal role in many embedded sub-system designs, additional capabilities are often required to implement the necessary system functions to optimize performance. One of the most constrained elements in MCU-based designs is the on-chip memory. The memory choice the designer makes for an embedded project drives the overall system capability and performance. Most MCUs are designed with internal memory optimized for certain applications, but this will not support all requirements. Many applications, especially those performing RAM-intensive functions like math algorithms, data buffering, or audio/video applications require large amounts of temporary storage beyond what the on-chip memory of an MCU has to offer.

Need for expansion memory

A trend predicted by Gordon Moore in 1965 was that semiconductor circuits would double the number of transistors on a chip every two years. This made it possible to shrink microelectronic components while maintaining or improving on performance every generation. Although this rate has slowed recently, the processing power combining digital logic and support peripherals possible in today's MCUs and SoCs is staggering. Multiple processing cores, analog circuits, I/Os and on-chip memory create a powerful stand-alone processor capable of supporting many applications. There is a balance, however, on dedicating too much chip area to memory, when other on-chip functions are needed and external expansion memory is available to complete the system.

Typically, MCUs are designed to have just enough on-chip memory to meet the target application's needs. Larger MCUs might have more processing power and correspondingly larger on-chip SRAM or Video RAM to run more powerful algorithms and process larger amounts of data. Conversely, smaller MCUs will come with smaller on-chip memory. If there is a requirement for additional RAM, the designer will supplement the system using external RAM which acts as expansion memory.

Human Machine Interfaces (HMIs), for example, may require a significant amount of buffer memory for rendering graphics. Compression techniques are sometime used to overcome this problem during data transmission to reduce local storage requirements or system bandwidth requirements. This means that a significant amount of scratchpad memory may be needed to decompress these files. Rendering these high-resolution images on displays also requires additional memory for buffering the images. Large

working memories are also needed in communications applications and in machine vision applications as buffer storage.

Expansion memory options

If an application has a requirement for additional RAM, or if the system design is frozen, but needs to implement additional functionality, what are the available options? A larger MCU might be an option. But, since most MCUs are RAM constrained, this would probably mean a more expensive device. And if a larger MCU is being considered just for extra RAM, you are probably paying for extra features and peripherals that are not needed. A larger MCU would also require more board space. Hence, for an existing product, this would mean redesigning the entire system to accommodate the new MCU with additional on-chip memory.

Another solution would be to add additional external RAM. External memory devices are available in various sizes ranging from kilobits to gigabits, so it is easy to add memory to accommodate the increased memory requirements. This would mean replacing an existing memory with a higher density device in the same package, or adding additional memory devices to the bus. By keeping the MCU and Memory separate, you get the flexibility to optimize the memory while keeping the design scalable for future requirements.

Depending on the density and performance requirements of the target application, traditional external expansion memory options are Static Random Access Memory (SRAM), Pseudo-Static Random Access Memory (pSRAM) and Dynamic Random Access Memory (DRAM). Each family uses a different technology and architecture which allows it to target certain density and performance specifications that a designer can choose to meet his design requirements. The figure below shows the density range which each of the technologies support.

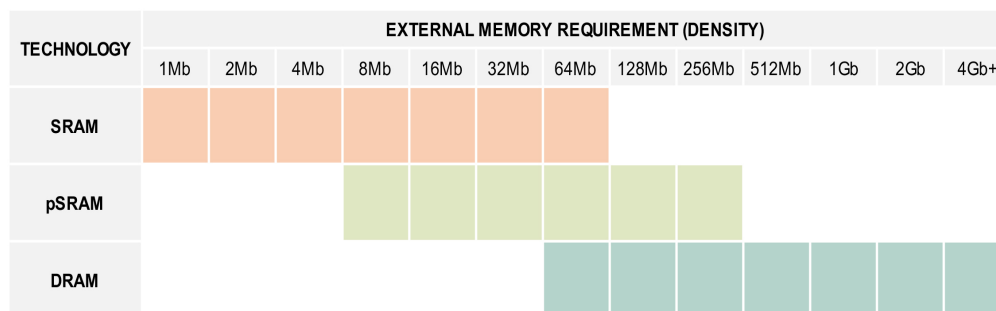


Figure 1: Density range for expansion memories

Traditionally, external expansion memories feature high pin-count parallel interfaces to support the requirements of most high-performance applications. However, parallel interfaces, while providing superior performance compared to serial interfaces, have their own trade-offs. The most significant is that parts with parallel interfaces have larger printed circuit board (PCB) footprints and higher pin count requirements. These interfaces require external address and data bus pins to facilitate the data I/Os and the MCU many not have sufficient pins available to support this. High pin-count memories typically also have fairly high standby current consumption.

Yet another alternative is to choose external memory with a standard serial interface such as I2C or SPI. These protocols use one/two/four data lines which are accessed serially (usually 1 to 4 bits at a time). The serial interface is simpler and smaller than parallel, but many times slower in terms of throughput. Due to fewer pins and lower speeds, serial interface memories typically consume less power when

compared to parallel interface memories. However, the greatest gain is in footprint – both in terms of device size and pin count. One of the expansion memory options created to meet this is the Serial SRAM. Serial SRAMs, however, are only available in densities up to 32Mb and speeds up to 52 MBps. Due to their low throughput, serial memories are usually limited to use in portable devices where size and power consumption matter more than access time, such as for handheld devices and wearables.

However, recent enhancements in SPI protocol, like introduction of Dual, Quad and Octal mode have made SPI a preferable option to be used with higher throughput. SPI was initially a three-wire protocol consisting of two data lines and one clock line, but due to the higher demands for throughputs in embedded devices new multi-lane protocols have been introduced to improve performance. These innovations have significantly closed the performance gap with parallel memory devices, providing a low-pin count solution for systems requiring higher throughput, however this still does not meet the throughput requirements of many high-performance applications.

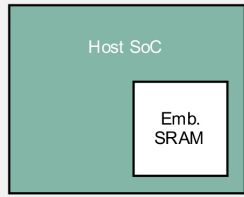
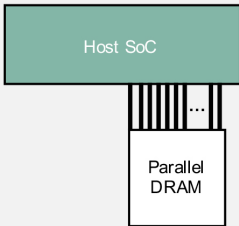
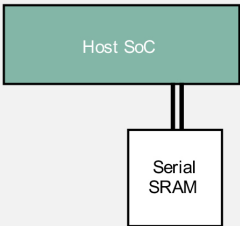
	Embedded RAM	External – Parallel RAM	External – Serial RAM
			
RAM Density	Limited ($\leq 64\text{Mb}$)	Flexible (32Mb to 1Gb+)	Limited (64Kb to 32Mb)
Throughput/Performance	Very High	High (100 Mbps to 1.6 Gbps+)	Low (Upto 52 Mbps)
Power Consumption	N/A (Embedded)	High	Low
SoC Package Size/Pins	Small	High	Small
RAM Package Size/Pins	N/A (Embedded)	High	Small
PCB Cost (4/6 Layer)	1x	2x	1x

Figure 2: Options for expanding RAM

The table above shows the available options for a designer to add on additional memory. The options for Internal vs External RAM are compared across the different parameters that affect design choice. Each of the available options for external memory have their own pros and cons. While a DRAM based memory may be a fit for certain high-performance applications, a Serial SRAM would be a better choice for power or board space constrained applications. In the case of applications that need a balance of throughput, footprint and power consumption, neither approach seems to be a good choice without any trade-offs.

For example, let us consider a Human Machine Interface (HMI) application with a 60Hz 720p resolution (1080x720) and 32 bpp depth where there is a requirement to deliver crisp images with fast refresh rates (link). Here, 128Mb of external memory would be required to buffer 4 frames with a throughput of >200 MBps. Serial SRAMs cannot be considered for this application as they are neither available in the required density nor do they meet the performance requirements. For other available options, the number of pins required for transmitting data which includes the address pins, the data I/O pins and the function pins (such as Chip Select, Output Enable, Write Enable, etc.) and the maximum throughput possible is provided in the table below.

128Mb: Total # of pins required for Data Tx	Data Bus Width		
	x8	x16	x32
pSRAM	-	32 pins, 266 MBps	-
SDR DRAM	29 pins, 200 MBps	38 pins, 400 MBps	56 pins, 800 MBps
DDR1 DRAM	-	41 pins, 800 MBps	61 pins, 1600 MBps

From the specifications, it would appear that using a x16 SDR DRAM memory with 38 pins is the option to be considered.

Traditionally, given the performance requirement and lack of alternatives, the “too-many-pins” limitation was not a design constraint. External expansion memories were used in systems where the controller performed extremely complicated functions and required large caches. Controllers for these applications have typically been large and had sufficient pins to interface with the high pin-count external memories. High-end applications could justify interfaces like DDR3 and DDR4. Unfortunately, this was often overkill for many applications where the pin count, speed, or power requirements are too high. What is needed is an interface that tries to strike a balance between pin count and performance.

Finding a niche: HyperBus™ – solving the “too-many-pins” problem

The HyperBus™ interface was introduced to enable a new memory architecture with a massively reduced pin count that could provide bandwidth required for high performance systems. Benefiting from the development of special protocol and clocking schemes and dedicated techniques for enhancing data integrity, the HyperBus interface is able to offer higher data throughput rates than legacy, high-pin count parallel interfaces using a low pin-count, small footprint package. Self-refreshed DRAM (HyperRAM™) and Flash (HyperFlash™) devices are available with the HyperBus interface. With HyperBus, microcontrollers can support multiple external flash and RAM devices on the same 8-bit data bus, versus the 16-bit data and 16-bit address bus and associated control pins of a typical parallel interface.

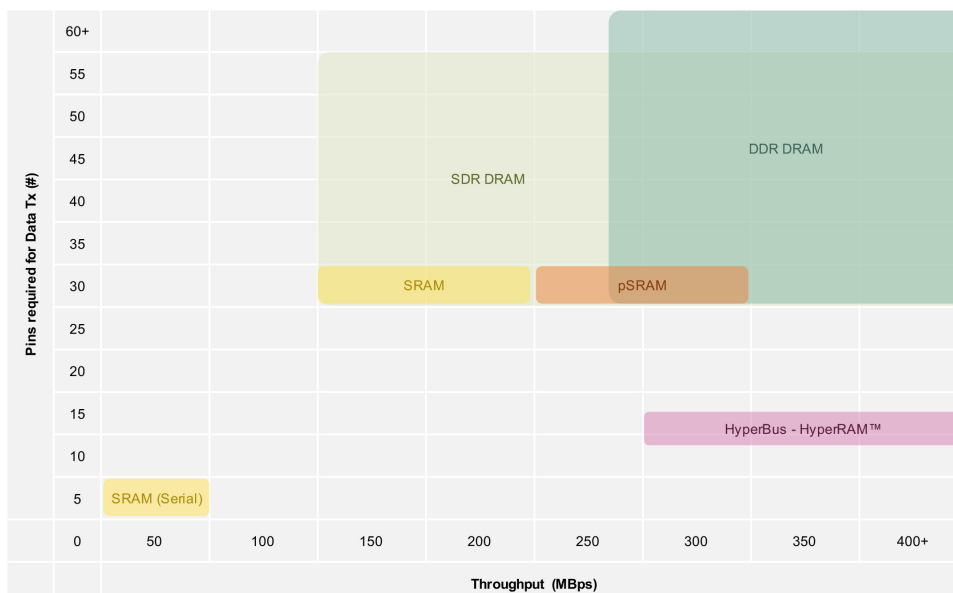


Figure 3: Map of expansion memory technologies

HyperBus fills the expansion memory bandwidth requirements between very high-speed DDR and slower interfaces such as SPI, QSPI, and I2C as can be seen in the Throughput vs Pin count map provided above. High speed serial interfaces exist for specific applications like MIPI for camera

interfacing, CAN for automotive signaling and PCI Express for interconnecting components to the computer motherboard. However, these serial interfaces operate at significantly higher clock rates and use more sophisticated protocols and are much more complex to implement on most hosts. They have their place in target applications but are overkill for many midrange embedded applications. HyperBus-based expansion RAM fills a large performance gap with an interface that is standardized and easy to implement.

	Embedded RAM	External – Parallel RAM	External – Serial RAM	External - HyperRAM
RAM Density	Limited (≤ 64Mb)	Flexible (32Mb to 1Gb+)	Limited (64Kb to 32Mb)	Flexible (64Mb to 512Mb)
Throughput/Performance	Very High	High (100 Mbps to 1.6 Gbps+)	Low (Upto 52 Mbps)	High (400 Mbps to 800 Mbps)
Power Consumption	N/A (Embedded)	High	Low	Low
SoC Package Size/Pins	Small	High	Small	Small
RAM Package Size/Pins	N/A (Embedded)	High	Small	Small
PCB Cost (4/6 Layer)	1x	2x	1x	1x

Figure 4: HyperRAM expansion memory strikes a balance between density, performance and pin-count

HyperBus is based on the JEDEC compliant Octal Extended SPI (xSPI) interface. The JEDEC xSPI standard defines requirements for the compatibility of high-performance x8 serial interfaces, including read and write commands, electrical characteristics, signalling protocols for command and data transfers, and a standard pin-out in a Ball Grid Array (BGA) footprint. The inclusion of the HyperBus interface in the JEDEC xSPI standard simplifies support of HyperBus-based memories for SoC providers and system designers developing high-performance applications by ensuring compatibility with xSPI compliant memories.

HyperBus utilizes a high-speed 8-bit DDR interface for both address and data. In addition, it has an optional differential clock, a read/write latch signal, and a chip select for each memory device. HyperBus allows to connect an external flash and RAM on the same bus, and works with any microcontroller with a HyperBus compatible peripheral interface.

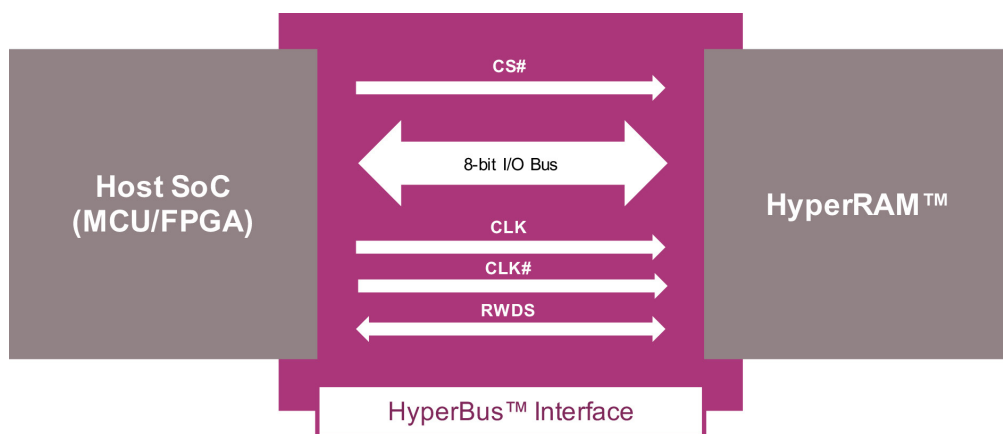


Figure 5: HyperBus can deliver upto 400 Mbytes/s using on 12 signal lines including an 8-bit data/command bus

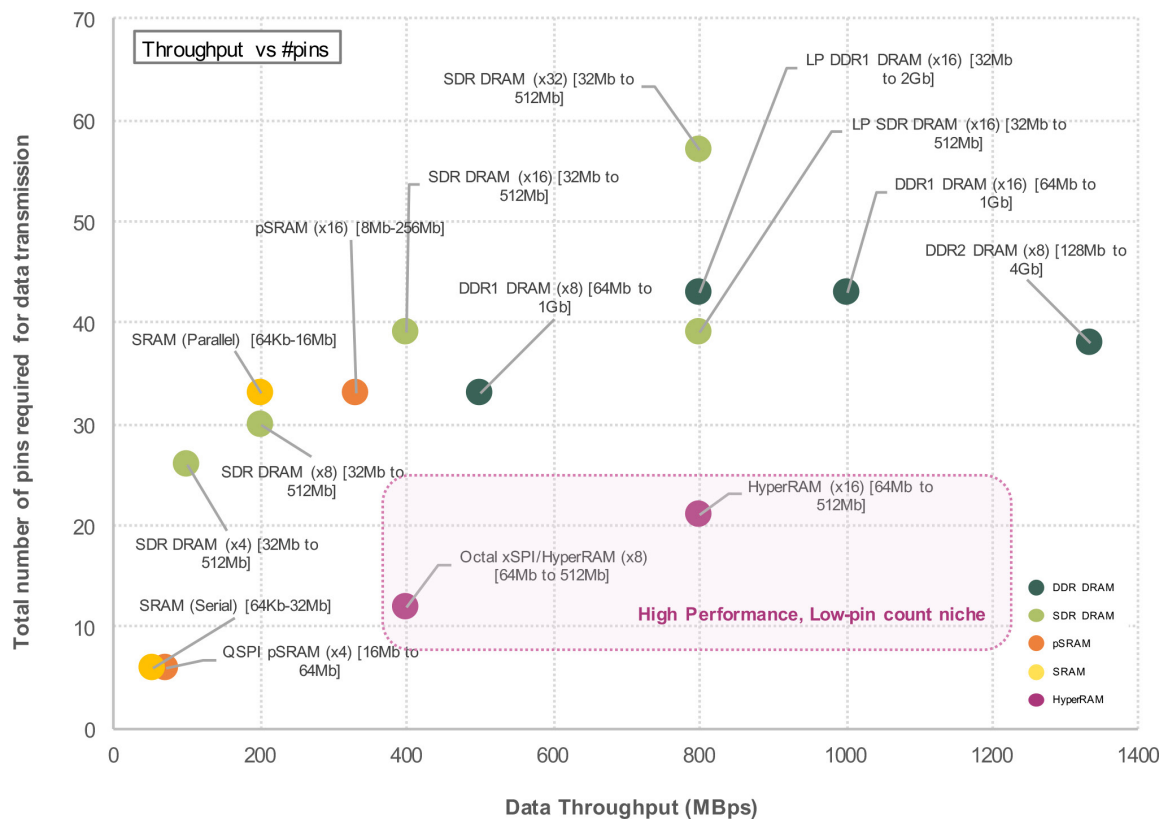
In total, 12 control and data signals are utilized to transfer data:

- CS#: Chip Select. Bus transactions are initiated with a High-to-Low transition. Transactions are terminated with a Low-to-High transition.
- CK (Clock Signal) and CK# (Clock signal on opposite phase): A command, address, and data output with respect to the crossing of the CK and CK# (optional) signals.
- DQ [7:0]: Data Input/Output: Command, address, and data are transferred on these signals during read and write transactions. As HyperBus is a DDR interface, data is transferred on both rising and falling clock edges.
- Read Write Data Strobe (RWDS): A bidirectional read/write data strobe signal, designated RWDS, is used to latch data. This enables the host to calibrate the read data accurately.

HyperBus is set up as a master/slave interface, where one host master interfaces to one or more slave memory devices on the bus. Each slave device is selected using an active low chip select, designated CS0#, CS1#, CS2#, etc. Only one chip select may be active at any time.

HyperRAM™: HyperBus (xSPI) based expansion ram

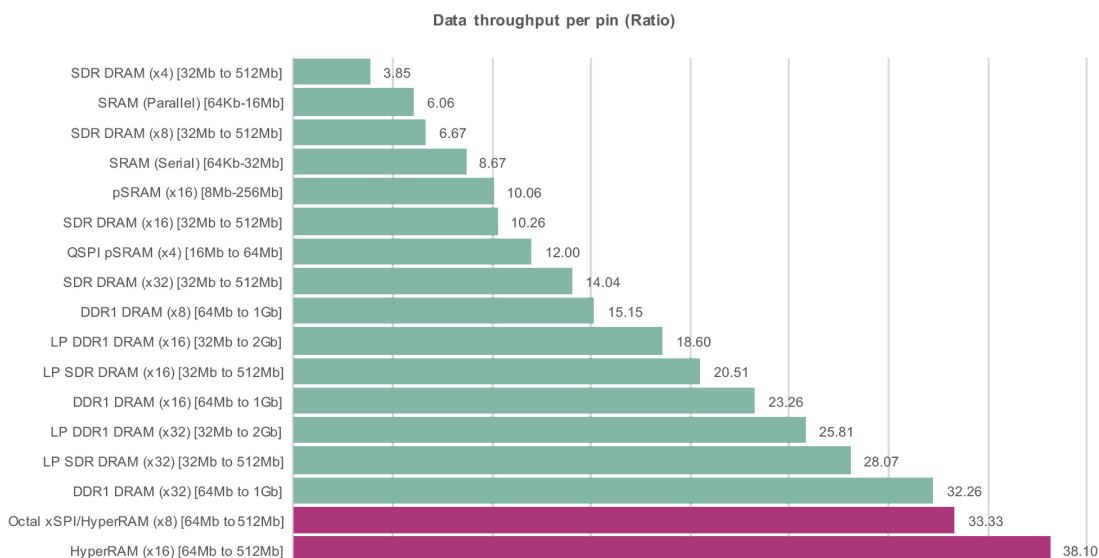
While advances in semiconductor technology continually increase the logic gate count in today's devices, I/O count generally remains constant. As a result, more data traffic has to be transferred over the same number of pins. The graph below lists the available expansion memory options for applications requiring densities up to 1Gb and a mapping of the throughput delivered by them vs the number of pins required to achieve the data transmission (including Address, Data and Function pins).



For lack of a standardized metric, we take the ratio of throughput to # pins (data throughput per pin) to compare the different options available for comparing Throughput vs Pin results across the technologies.

Memory Technology (Interface)	Data Bus Width	Available Density Range	Data Throughput (MBps)	Total Pins Required for Data Transfer (#)	Ratio of Throughput to # pins required
SRAM (Serial/QSPI)	x2/x4	64Kb-32Mb	52	6	8.67
SRAM (Parallel)	x8/x16	64Kb-16Mb	200	33	6.06
pSRAM (A/D MUX)	x16	8Mb-256Mb	332	33	10.06
pSRAM (QSPI)	x4	16Mb to 64Mb	72	6	12.00
HyperRAM (xSPI/HyperBus)	x8	64Mb to 512Mb	400	12	33.33
HyperRAM (HyperBus Ex. I/O)	x16	64Mb to 512Mb	800	21	38.10
DRAM (SDR)	x4	32Mb to 512Mb	100	26	3.85
DRAM (SDR)	x8	32Mb to 512Mb	200	30	6.67
DRAM (SDR)	x16	32Mb to 512Mb	400	39	10.26
DRAM (SDR)	x32	32Mb to 512Mb	800	57	14.04
DRAM (DDR1)	x8	64Mb to 1Gb	500	33	15.15
DRAM (DDR1)	x16	64Mb to 1Gb	1000	43	23.26
DRAM (DDR1)	x32	64Mb to 1Gb	2000	62	32.26
DRAM (DDR2)	x8	128Mb to 4Gb	1334	38	35.11
DRAM (DDR2)	x16	128Mb to 4Gb	2668	47	56.77
DRAM (DDR2)	x32	128Mb to 4Gb	5336	64	83.38
LP DRAM (SDR)	x16	32Mb to 512Mb	800	39	20.51
LP DRAM (SDR)	x32	32Mb to 512Mb	1600	57	28.07
LP DRAM (DDR1)	x16	32Mb to 2Gb	800	43	18.60
LP DRAM (DDR1)	x32	32Mb to 2Gb	1600	62	25.81

The table demonstrates that HyperRAM achieves the performance levels of DRAM memories by utilizing less than half the number of pins.



In the HMI example cited earlier, HyperRAM would emerge as the perfect memory of choice for implementing the design in an optimal and cost-efficient manner. Operating at 200MHz with DDR, and requiring only 12 pins for data transmission, HyperRAM memories can achieve 400MBps throughput.

128Mb: Total # pins required for Data Tx	Data Bus Width		
	8-bit (x8)	16-bit (x16)	32-bit (x32)
pSRAM	-	32 pins, 266 MBps	-
SDR DRAM	29 pins, 200 MBps	38 pins, 400 MBps	56 pins, 800 MBps
DDR1 DRAM	-	41 pins, 800 MBps	61 pins, 1600 MBps
HyperRAM/xSPI Devices	12 pins, 400 MBps	-	-

Available in a broad range of densities starting from 64Mb, HyperRAM ensures that the HMI design is scalable to meet the ever-increasing demand of larger displays and richer graphics. The low-pin count implementation ensures that you use an MCU with just the requirements that you need. In addition to saving on the MCU cost, you also save on board space and PCB layers due to the requirement for routing lesser number of signals. Additionally, HyperRAM requires 83% lower active current than the alternative SDR DRAM option. This effectively optimizes both board space and BOM cost and makes the system power efficient.

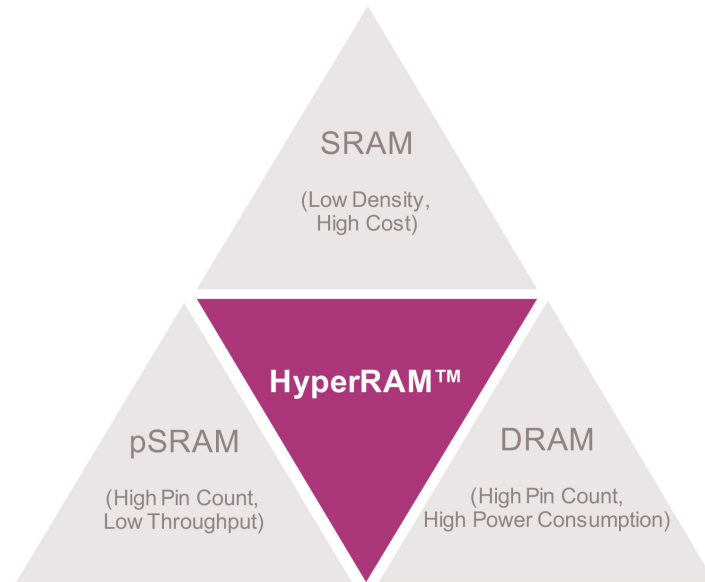


Figure 6: HyperRAM overcomes the shortcomings of traditional expansion memories

Using HyperRAM memory in MCU/FPGA applications provides embedded developers with the opportunity to create unique solutions. Capturing camera or microphone data as well as intermediate layer storage in the HyperRAM memory is ideal for Artificial Intelligence and Machine Learning applications. As discussed earlier, the memory can also be used as a frame buffer for displays, retaining the last video frame while a system goes to sleep or is rendering new graphics. An FPGA can be used to bridge between other processor interfaces such as QSPI or traditional parallel SRAM, where there may be optimization in PCB footprint, layout, or cost.

HyperRAM + HyperFlash: Reducing complexity and simplifying system design

Flash is an essential component for all embedded systems for code storage. Using both the HyperFlash and HyperRAM, the MCU can easily extend both its Flash & RAM internal memory with minimal impact on PCB size and design complexity. HyperRAM™ supports a common footprint with HyperFlash, using only 12 pins for data transactions as shown in the image below.

HyperRAM and HyperFlash share a common footprint and requires only 1 additional pin to communicate on the same HyperBus.

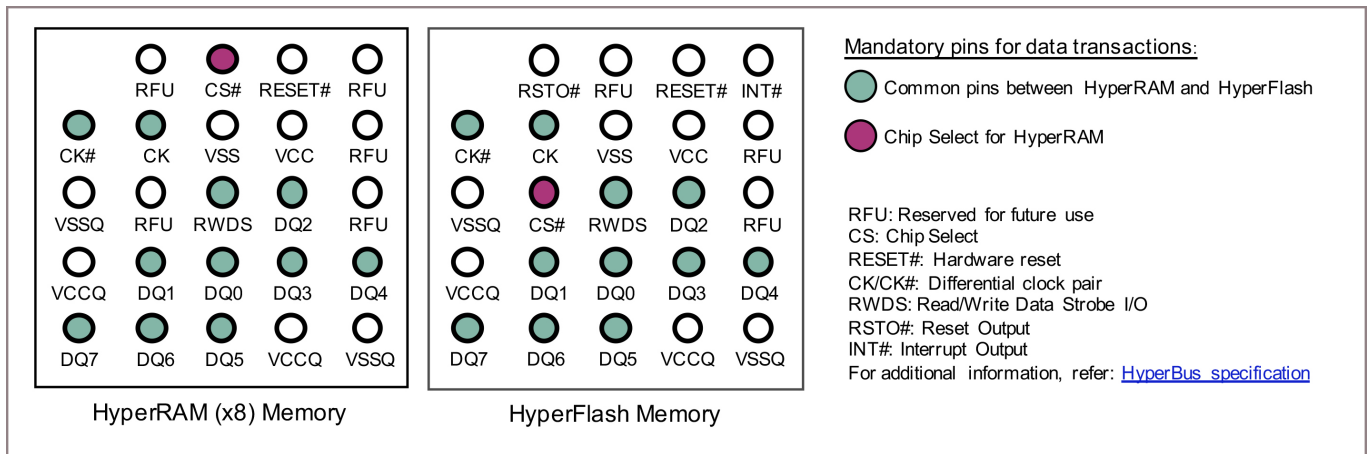


Figure 7: HyperRAM and HyperFlash share a common footprint

The increased data throughput-per-pin that HyperBus offers as mentioned earlier enables the use of an MCU with fewer pins (smaller, therefore cheaper), and a PCB with fewer layers (easier to design and cheaper to manufacture and assemble). In many cases, systems using HyperBus technology will benefit from both this cost saving and enhanced performance when replacing legacy SDRAM + NOR Flash systems. The SDRAM + NOR Flash solutions would require more than 6 PCB layers to implement the complex signal routings that arises from the high pin count requirement of the two memories. In the case of the HyperBus solutions, the connection between the MCU and HyperFlash and HyperRAM in separate packages require only four layers including Vcc and GND. An MCP device, combining the RAM and Flash in a single package, goes further and requires one less layer. This HyperBus solution is ideal for mid-range graphics applications such as the instrument cluster described earlier.

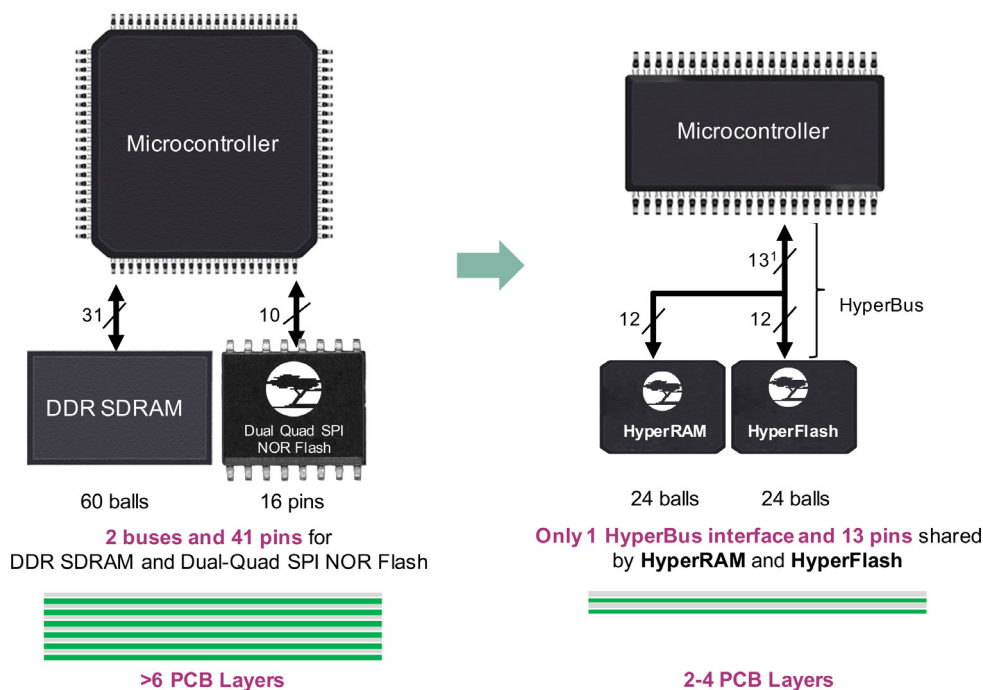


Figure 7: HyperRAM and HyperFlash share a common footprint

Given the flexibility that is being offered to the designer for both NOR & RAM, a combination of HyperBus based memories offer compelling value without compromising on cost and will suit the needs of many applications.

Conclusion:

For designers wishing to scale up a family of products based on a uniform board layout, the HyperBus interface offers a simple way to migrate to higher speeds and densities while being power efficient and optimizing on overall system cost. Memories based on the HyperBus interface enable faster systems with quicker response times and rich user experiences. This high throughput, low-pin count interface delivers tremendous possibilities enabling applications that meet specifications previously not achievable.

Infineon memory solutions products:

HyperRAM™ and HyperFlash™ memory ICs are available from Infineon Technologies (formerly Cypress Semiconductor) in a wide range of density options, each housed in a 6mm x 8mm, 24-ball BGA package. Infineon also offers a HyperFlash + HyperRAM MCP solution: a system-in-package combining a NOR Flash die and a self-refresh DRAM die in a single 24-ball BGA package. All products are available in both Industrial and Automotive grades.

www.cypress.com/HyperRAM

Published by
Infineon Technologies AG
81726 Munich, Germany

© 2020 Infineon Technologies AG.
All Rights Reserved.

Please note!

This Document is for information purposes only and any information given herein shall in no event be regarded as a warranty, guarantee or description of any functionality, conditions and/or quality of our products or any suitability for a particular purpose. With regard to the technical specifications of our products, we kindly ask you to refer to the relevant product data sheets provided by us. Our customers and their technical departments are required to evaluate the suitability of our products for the intended application.

We reserve the right to change this document and/or the information given herein at any time.

Additional information

For further information on technologies, our products, the application of our products, delivery terms and conditions and/or prices, please contact your nearest Infineon Technologies office (www.infineon.com).

Warnings

Due to technical requirements, our products may contain dangerous substances. For information on the types in question, please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by us in a written document signed by authorized representatives of Infineon Technologies, our products may not be used in any life-endangering applications, including but not limited to medical, nuclear, military, life-critical or any other applications where a failure of the product or any consequences of the use thereof can result in personal injury.