White Paper

Using Memory Buses to Build Modular Portable Device Architectures

The convergence of multiple functions into a single portable device is creating new challenges. Product lifetimes are becoming shorter, and embedded applications are evolving quickly. This means that OEMs cannot afford to design new platforms for each new generation. More flexible, modular platforms, that can easily be upgraded, or to which a new functionality can be added with minimum redesign effort, offer a solution to this problem.

The best example of this situation is the mobile phone. Camera functions, video, games, 802.11 connectivity, MP3 capability and computer synchronization are being added to basic voice functionality. PDAs or portable gaming consoles are facing the same challenges, with the merging of similar functionalities.

Defining a Modular Architecture

A flexible architecture includes multiple subsystems that can be upgraded independently with minimum impact on the other blocks. Each subsystem supports a specific function that evolves independently from the others (for instance wireless technology evolving independently from video performance).

Figure 1 shows an example of an architecture using four subsystems. One is dedicated to communication and is typically centered around a baseband modem. Another is dedicated to applications, perhaps an application processor running 3D games or supporting PDA functionality. A third subsystem provides connectivity to Video Broadcast (DMB or DVB-H), and a fourth block is dedicated to audio processing, supporting for example MP3 player functionality.

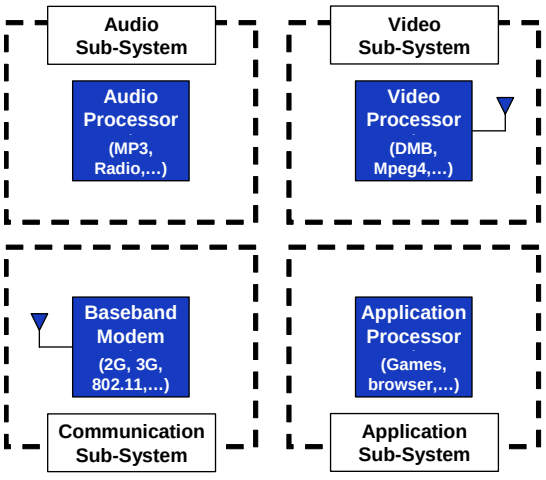

Figure 1. Example of a Modular Architecture with 3 Subsystems

These subsystems can be used to build specific platforms, and left untouched for others. An entry-level platform would probably use only one of these. A more advanced platform could use two, three or four subsystems, depending on the targeted features and market.

In a modular architecture, each of these blocks can be upgraded independently. For instance, the communication block can be upgraded to support a different wireless standard (moving from GPRS to W-CDMA for instance) or to upgrade the existing protocol (adding HSDPA to an existing W-CDMA baseband). Similarly, the application sub-system can be upgraded to support new capabilities (e.g., games, OS upgrade) independently of the wireless standard supported.

Connecting Subsystems with a Future-Proof Interface

To minimize the impact of each upgrade, the different subsystems have to connect to each other through an interface that will be available on the next-generation chipset, and will support the high-throughput needs of the next generation.

Each subsystem is typically centered around a processing element (ASIC, DSP, or GPP). Most of these processors today lack such an interface: serial standard interfaces are too slow to support new throughput needs such as 3G, video or 802.11; and high-speed interfaces most often involve proprietary protocols that are not compatible across chip vendors, and involve software changes and validation each time one of the blocks is updated.

The memory bus provides a good candidate for an interconnect—all processors support and will continue to support a memory interface, and the bandwidth achieved is high above current requirements (290 Mbps for a 55-ns access time on an x16 bus). Two memory buses can be connected together using a dual-port interconnect.

A dual-port interconnect is a memory-mapped device that allows two processing elements to access a shared memory space independently. It allows these processing elements to exchange data simply through their memory bus, using standard write and read operations. *Figure 2* shows an example of a baseband modem and a video processor using a dual-port to connect to each other. The video-phone built with this architecture has two subsystems that can evolve independently.

The dual-port further allows the upgrade of an existing system by adding a processing element to the existing architecture. In the example shown in *Figure 2*, the communication subsystem might have been an existing system. The addition of the new processor causes limited changes to the existing system since the existing processor only perceives additional

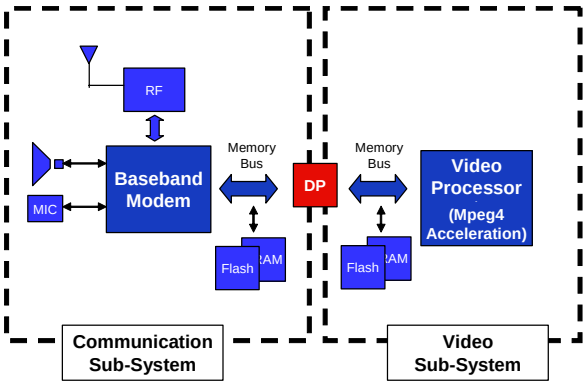

Figure 2. Example of a Handset Block Diagram, using a Dual-Port to Connect Two Subsystems

memory on its bus. One port of the dual-port is connected to the existing processor's memory bus and the other port could be connected to any type of device with an SRAM interface. These devices could include any type of processor (adding gaming, MP3, video, or even PDA functionality), or modem (adding a new communication band, 3G acceleration, 802.11 connectivity or video broadcast).

Modular Portfolio Strategy

More than just enabling upgrades with short development time, modular architectures maximize the re-use of IP developed for specific subsystems. A whole portfolio can be built out of the different blocks available, with the use of similar subsystems across multiple platforms.

Figure 3 describes a handset portfolio based on this modular approach. A single baseband chip is used in the simplest designs. It is then enhanced multiple times to create several new products by the addition of different application processors via a dual-port interconnect. The dual-port interconnect enables two completely different applications processors to be added to the baseband processor with minimal changes. This allows the market and cost differentiation of the new platforms, supporting additional applications while using the same communication block.

When the next-generation baseband is developed, the previously developed application IP can be re-used without changes while upgrading the communication subsystem. With this process, the existing architecture can even be exported easily from one band to another (GPRS to CDMA, for instance) with minimal changes to software and system architecture.

Cypress MoBL® Dual-Port

A dual-port is the most flexible interprocessor interconnect available today. By offering a standard SRAM interface that connects with the existing memory bus, it allows an existing processor to interface with almost any other processing element with high bandwidth. It also simplifies and reduces the communication software overhead.

Cypress Semiconductor combined its expertise in dual-port architecture and low-power technology to build the low cost MoBL® (More Battery Life™) Dual-Port, designed and priced specifically for mobile platforms. The Cypress MoBL Dual-Port also solves several other needs of new portable device architectures.

In terms of power, the Cypress MoBL low-leakage technology allows the dual-port's standby power consumption to be below 3.6 µW at 1.8V. This makes the Cypress MoBL Dual-Port the lowest-power dual-port available on the market today. It also supports multiple I/O voltage (1.8V, 2.5V and 3V) for compatibility with a wide variety of baseband and other processors.

Further, the bandwidth requirements of wireless systems continue to increase. Wireless systems are now required to support non-cellular technologies such as 802.11b, with data rates up to 11 Mbps. Standard low-speed serial interconnects, when available on some processors, cannot usually provide the required bandwidth. The Cypress MoBL Dual-Port offers fast access, 35 ns, over an x16 bus, allowing over 400 Mbps of throughput, which exceeds the data rate requirements of 3G, WiFi or video broadcast.

With a small footprint being another requirement, the Cypress MoBL Dual-Port is available in die form or packaged in a small 6x6 mm 0.5-mm-pitch BGA.

In terms of functionality, the Cypress MoBL Dual-Port behaves as any other Cypress asynchronous dual-port. A

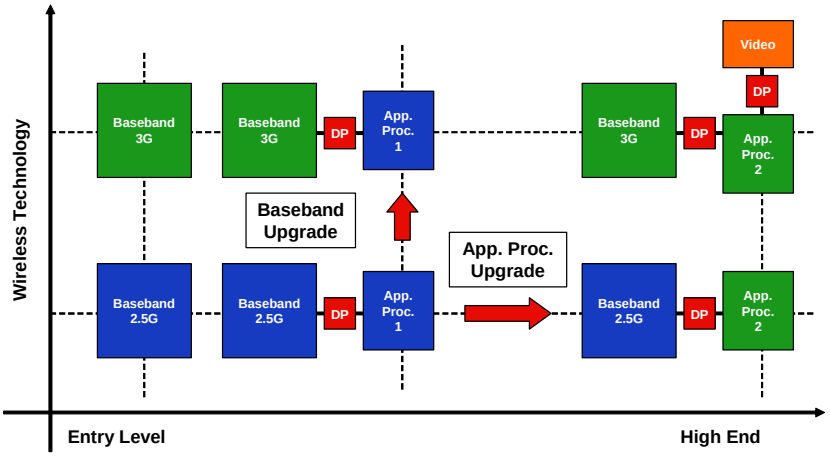

Figure 3. Example of a Handset Portfolio using a Modular Architecture Strategy

busy signal is asserted when both ports try to access the same memory space at the same time, providing built-in arbitration. A mailbox functionality allows the two processors to send each other interrupts simply by writing to a specific location. This can be used as a way to signal to the other processor that data is available for download.

Handset system designers want to minimize the number of GPIO pins that drive very simple functionality, such as reading external DIP switches or lighting LEDs. The Cypress MoBL Dual-Port allows some of these signals to be external to the processor by incorporating input read registers and output drive registers. The Input Read Register (IRR) captures the status of two external binary devices such as DIP switches in a specific memory space. By simply reading from the dual-port, either processor can monitor the status of these two devices. The Output Drive Registers (ODR) can drive up to five interrupts signals. This allows either processor to control up to five external devices by just writing to a specific location in the dual-port.

Conclusion

To stay competitive, portable device OEMs now adopt a modular strategy for their designs. This allows them to quickly upgrade a platform, and release several new handsets in a short window. Dual-ports enable this strategy by providing a high-throughput, memory-mapped interconnect. They allow the creation of multiple sub-systems that can evolve independently and communicate through their memory interface. To answer this emerging need, Cypress has designed the new MoBL Dual-Port specifically for this type of application, making it the lowest power dual-port available.