# IDT Peripheral Bus (IPBus™)
# Intermodule Connection Technology Enables Broad Range Of System-Level Integration

**An IDT White Paper**

**Introduction**

Different types of products require different solutions. Some are better served by a discrete CPU solution that enables rapid processor upgrade as well as the spawning of multiple products from one basic design. Other products require system-level, integrated-processor solutions. These designs need the small size, reduced cost, and reduced power consumption afforded by system-level integration. Consequently, IDT supports both choices with a strategy that encompasses both discrete- and integrated-processor designs.

The success of this strategy depends upon a design methodology and organization optimized for two different design challenges. Our microprocessor design team builds stand-alone CPUs and CPU cores. Our system-level design group builds System Controllers that can support discrete CPUs and also serve as controllers in highly integrated, system-level solutions. This group is also responsible for external IP inclusion. The key to making these system-level designs efficient is the IDT Peripheral Bus (IPBus). It is the interface that ties functional blocks together in System Controllers and system-level integration solutions. The IPBus also makes it much easier for IDT to include third-party IP functions in a design.

**The Discrete Solution Strategy**

The discrete CPU solution strategy is aimed at customers building edge-of-the-art systems, and who plan to build multiple end products from one design. From their perspective, it is far more important to be able to quickly take advantage of a faster microprocessor than to limit themselves to a core processor integrated onto a system-level IC. For that reason, IDT continues to invest heavily in CPU R&D, and to offer System Controller adjunct ICs that can support a wide range of CPUs and CPU speeds.

IDT has been designing and manufacturing RISC CPUs for more than a decade. Our CPUs are based on the MIPS RISC CPU architecture, but we have freely innovated within the bounds of that standard. Our CPU families include both 32- and 64-bit RISC CPUs, and a new class of advanced 32-bit CPUs that implement the new MIPS-32 ISA. The latter offer a convenient bridge between 64- and 32-bit implementations. For example, the existing code from a 64-bit design can be easily moved to our new RC32364™ CPU.

From the outset, IDT believed that our RISC CPUs could provide more than just general-purpose processing performance. We encouraged designers to consider using the CPU's power as a way to emulate fixed-function hardware, too. As a result, the designer can eliminate extra hardware components without sacrificing some performance gain, and the end design can be completed faster because there's no need to "tune" the hardware. In general, the CPU will need to interface with memory, add-ons and application-specific functions. To that end, IDT has integrated the interface and control functions into what it calls "System Controllers."
(see figure 1)

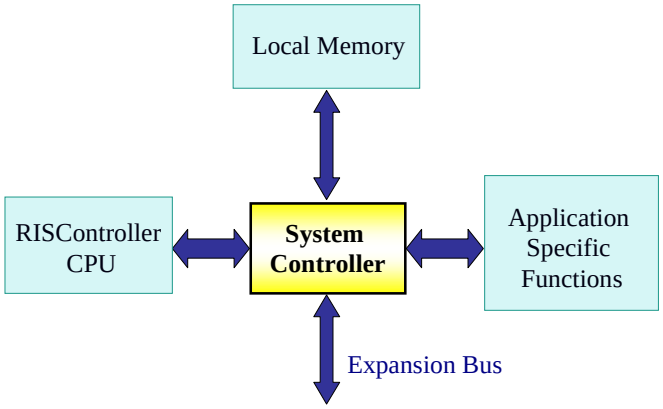


Figure 1. In most systems, the RISController CPU will interface with memory, expanded system functions and application specific functions. The System Controller IC is designed to facilitate these and other capabilities within a single chip.

In the first such controller, the RC64145™, the IC uses a 64-bit implementation of the IPBus to tie the separate functions together (see figure 2). Here, the System Controller is the path through which the CPU engages the memory subsystem. It also supports the PCI bus for expansion, has timers to support real-time operating system needs and a UART for asynchronous I/O to support application-specific functions, such as remote control and the like.

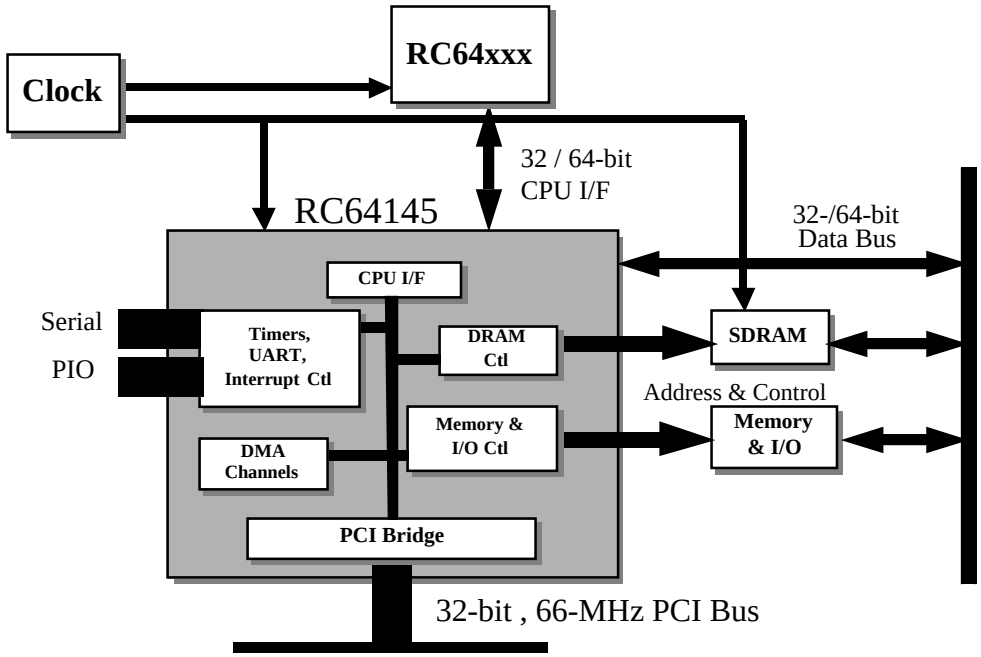


Figure 2. The RC64145 System Controller offers an interface to the CPU and a bridge to other IP. The various functions are tied together with a 64-bit IPBus, and expansion capabilities are supported through the PCI interface.

The same System Controller is designed to be used with a range of CPUs that span clock speeds from 133 to 250 MHz (see figure 3). As a result, a designer can put together a single design that can provide end products to cover a broad price/performance range.

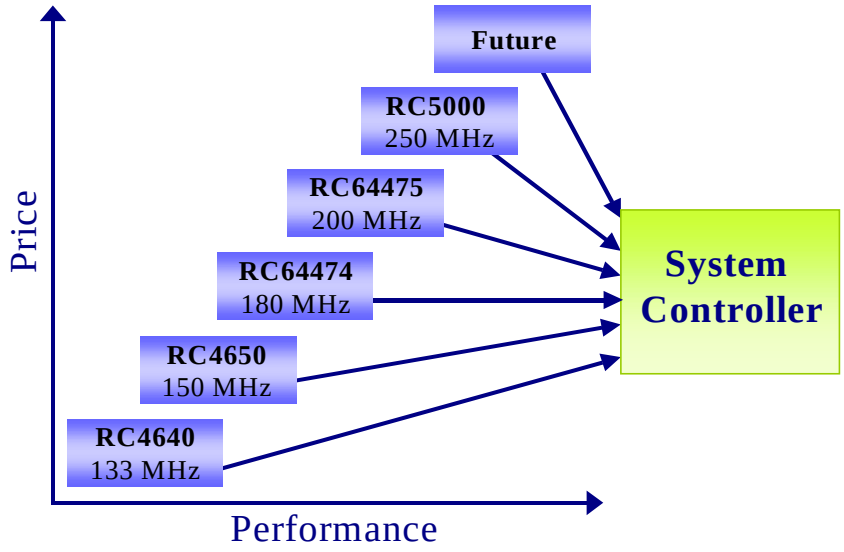


Figure 3. The RC64145 System Controller can be used with many different 64-bit CPUs, covering a two-to-one speed range, and supporting a broad range of price/performance end products.

The second System Controller is the RC32134™. It is designed for use with the advanced 32-bit CPU, the RC32364. Note the similarity in functions between the RC64145 and this System Controller. Again, the tie that binds is the IPBus, but here we used a 32-bit implementation. In this case, the CPU interfaces directly with memory rather than working through the System Controller's internal bus (see figure 4). The IPBus can accommodate either data flow.

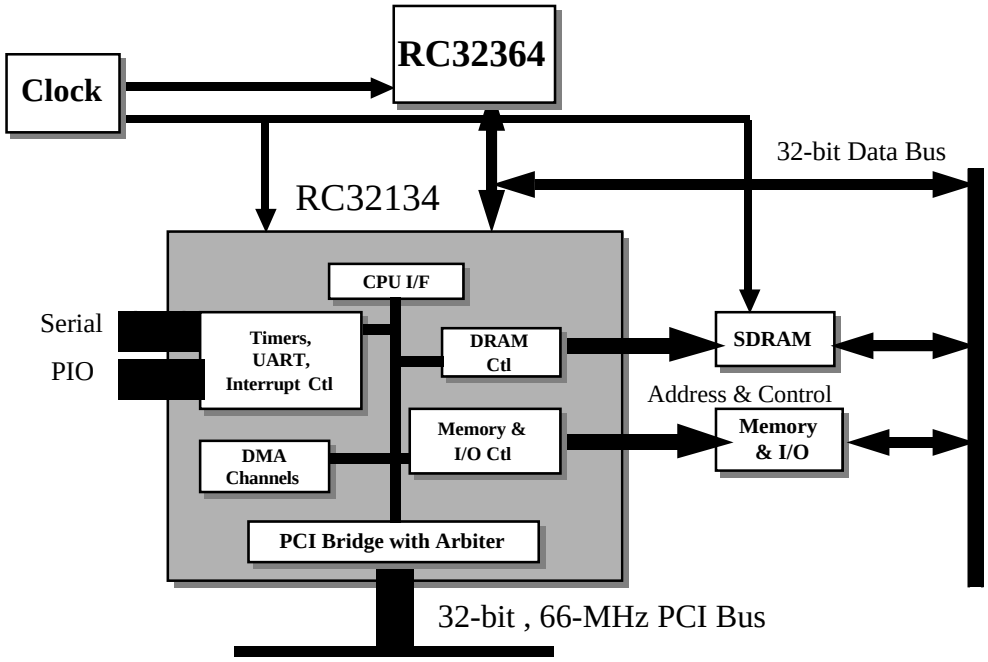


Figure 4. The RC32134 System Controller is designed specifically for the RC32364 CPU. The controller offers support for both EDO and SDRAM memories.

**IP Bus – The Tie That Binds**

As shown in the previous section, the IPBus is the essential tie that binds the various functions together in the System Controller (see figure 5). It is also a key part of the system-level integration design. The IPBus is a synchronous high-speed (>100 MHz) bus that is processor independent. It features demultiplexed, pipelined address and data as well as burst data capability – both cache-line size and indeterminate length. It is a multi-master type bus that also provides "fly-by" DMA support. The IPBus interface is a small piece of code that is part of all the functional modules designed by IDT's system-level design group. It is also added to any third-party function that is to be integrated into a System Controller or system-level integration IC. A slave module requires about 500 gates, and a master/slave module can be implemented using under 1000 extra gates. The IPBus is designed to work well with standard IC design tools, making it very easy to incorporate it into a design.

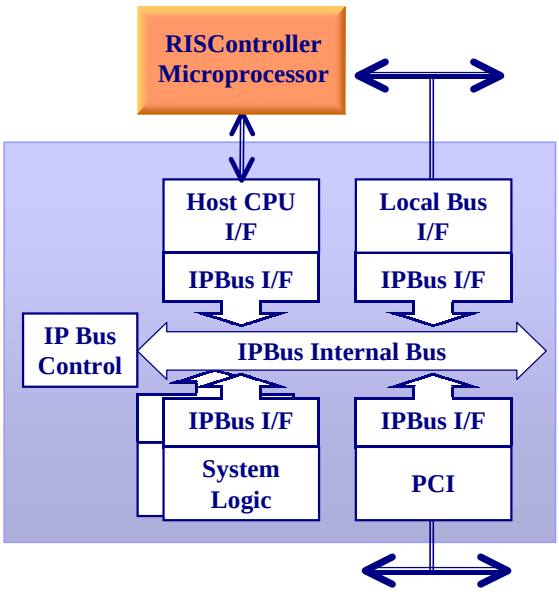


Figure 5. The IPBus is an integral part of any System Controller or system-level integration design. It makes the whole design structure modular allowing the incorporation of a variety of functions, including external IP.

**System-Level Integration Strategy**

The system-level integration strategy is aimed at those who need high-volume solutions where small size, low price, and low-power consumption are paramount. At any point in time, the system-level integration capability is a snapshot of the CPU and controller capabilities developed to serve the discrete solution strategy (see figure 6). The Koa project, shown here, features the RC32300 core, a 133-MHz/175 MIPS processor. As shown, there is extensive re-use of modules from the RC32134 System Controller, and the use of the IPBus to tie it all together. Here, the tradeoff is the CPU and ancillary functions are integrated onto the same IC. There is no elegant way, therefore, to simply upgrade processor speed.

IDT's system-level integration strategy includes integrating high-performance CPU cores. Software emulation of common hardware functions can then be used to save gate count. The CPU can be used to perform a number of value-added protocols in software. For example, one could emulate a V.90 modem, voice-over-IP functionality and SAR simulation for a design to be used in an ATM network environment. Overall, the benefit using an integrated processor design is lower cost, smaller size and lower power consumption.

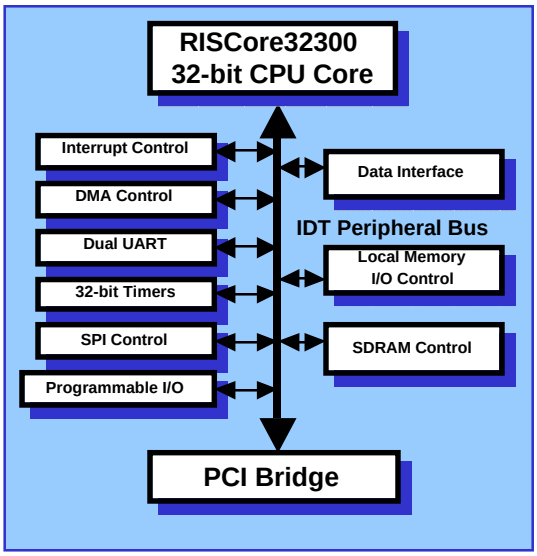


Figure 6. The System Level Integration is a snapshot of the capabilities developed in the discrete solution strategy, then cast onto a single piece of silicon Note the re-use of functions from the RC32134 controller, such as DMA control, timers, UARTs, and so on. The CPU core is the 133 MHz/175 MIPS 32300, a powerful RISC processor that can support gate-count reducing hardware emulation of V.90 modems and the like.

**Proofs**

Strategies are one thing; proven strategies are another. The first two System Controllers attest to the efficacy of IDT's IPBus technology. It takes no great leap of faith to see how that same technology will lend itself to application specific standard products (ASSPs) that incorporate IDT's CPU and RAM cell technologies, its System Controller function expertise and its ability to quickly incorporate any third-party or customer's IP functions into the mix.