

Optimize MicroBlaze Processors for Consumer Electronics Products

An RTOS can be critical for getting the most out of the MicroBlaze processor.

by John Carbone
VP, Marketing
Express Logic, Inc.
jcarbone@expresslogic.com

A fast processor is essential for today's demanding electronic products. Efficient application software is equally essential, as it enables the processor to keep up with the real-world demands of networking and consumer products.

But what about the real-time operating system (RTOS) that handles your system interrupts and schedules your application software's multiple threads?

If it's not optimized for the processor you're using, and isn't efficient in its handling of external events, you'll end up with only half a processor to do useful work.

An efficient RTOS should do more than just handle multiple threads and service interrupts. It must also have a small footprint and be able to deliver the full power of the processor used by your application software. Express Logic's ThreadX® real-time operating system is just such an RTOS.

The ThreadX RTOS has now been optimized to support the Xilinx MicroBlaze™ soft processor, and is available off-the-shelf for your next development project.

A Good Fit

Let's look at some of the reasons why the ThreadX RTOS is a good fit with the MicroBlaze processor.

Size

The ThreadX RTOS is only about 6 KB - 12 KB in total size for a complete multi-tasking system. This includes basic services, queues, event flags, semaphores, and memory allocation services.

Its small size means that more of your memory resources will be preserved for use by the application and not absorbed by the RTOS. This will result in smaller memory requirements, lower costs, and lower power consumption. Table 1 shows code sizes for various optional components of the ThreadX RTOS on the MicroBlaze processor.

Efficiency

The ThreadX RTOS can perform a full context switch between active threads in a mere 1.2 μ s on a 100 MHz Xilinx Virtex-II Pro™ FPGA. This is particularly critical in applications with high interrupt incidences, such as network packet processing.

The ability of the ThreadX RTOS to handle interrupts efficiently means that your system can handle higher rates of external events such as TCP/IP packet arrivals, delivering greater throughput.

Speed

The ThreadX RTOS responds to interrupts and schedules required processing in less than 1 microsecond. The single-level interrupt processing architecture of the ThreadX RTOS eliminates excess overhead found in many other RTOSs.

This keeps the ThreadX RTOS from gobbling up valuable processor cycles with housekeeping tasks, thus leaving more processor bandwidth for your application. Thus, you can configure a less expensive processor or add more features to your application without increasing processor speed and cost.

Table 2 shows execution times for various ThreadX services measured according to the following definitions:

- **Immediate Response (IR):** Time required to process the request immediately, with no thread suspension or thread resumption.
- **Thread Suspend (TS):** Time required to process the request when the calling thread is suspended because the resource is unavailable.
- **Thread Resumed (TR):** Time required to process the request when a previously suspended thread (of the same or lower priority) is resumed because of the request.
- **Thread Resumed and Context Switched (TRCS):** Time required to process the request when a previously suspended thread with a higher priority is resumed because of the request. As the resumed thread has a higher priority, a context switch to the resumed thread is also performed from within the request.
- **Context Switch (CS):** Time required to save the current thread's context, find the highest priority ready thread, and restore its context.
- **Interrupt Latency Range (ILR):** Time duration of disabled interrupts.

Ease of Use

All services are available as function calls, and full source code is provided so you can see exactly what the ThreadX RTOS is doing at any point.

Many other commercial RTOS products are massive collections of unintelligible system calls with non-intuitive names. The ThreadX RTOS uses service call names that are orderly, simple in structure, and ultimately easier to use. This

enables shorter development times and faster time to market.

Fast Interrupt Handling

The ThreadX RTOS is optimized for fast interrupt handling, saving only scratch registers at the beginning of an interrupt service routine (ISR), enabling use of all services from the ISR, and using the system stack in the ISR.

ThreadX Instruction Area Size (in bytes)	
Core Services (required)	4,804
Queue Services	3,200
Event Flag Services	1,720
Semaphore Services	932
Mutex Services	2,540
Block Memory Services	1,144
Byte Memory Services	1,876

Table 1 – Code size of various ThreadX services

ThreadX Service Call Execution Times				
Service	IR	TS	TR	TRCS
tx_thread_suspend	1.0 µs	2.0 µs		
tx_thread_resume			1.0 µs	2.8 µs
tx_thread_relinquish	0.5 µs			1.9 µs
tx_queue_send	0.9 µs	2.4 µs	1.6 µs	3.4 µs
tx_queue_receive	0.9 µs	2.3 µs	2.5 µs	4.2 µs
tx_semaphore_get	0.3 µs	2.3 µs		
tx_semaphore_put	0.3 µs		1.3 µs	3.0 µs
tx_mutex_get	0.4 µs	2.4 µs		
tx_mutex_put	1.0 µs		2.0 µs	3.6 µs
tx_event_flags_set	0.6 µs		1.7 µs	3.4 µs
tx_event_flags_get	0.4 µs	2.5 µs		
tx_block_allocate	0.4 µs	2.3 µs		
tx_block_release	0.4 µs		1.4 µs	3.1 µs
tx_byte_allocate	1.4 µs	2.9 µs		
tx_byte_release	0.9 µs		3.0 µs	4.7 µs
Context Switch (CS)	1.2 µs			
Interrupt Latency Range (ILR)	0.0 µs-0.8 µs			

Table 2 – Execution times of various ThreadX services

Efficient interrupt processing delivers the best performance from MicroBlaze processors, giving your product a boost over the competition.

Full Source Code

By providing full source code, you have complete visibility into all ThreadX services – no more mysteries are hidden within the RTOS “black box.”

This full view of source code enables you to avoid delays in development caused by an incomplete understanding of RTOS services. With the ThreadX RTOS, full source code is there any time you need it.

Conclusion

Express Logic's ThreadX RTOS is well matched for networking and consumer applications based on MicroBlaze processors. It's been optimized to deliver the processor's inherent performance directly through to the application.

What's next? Express Logic is working to port NetX™, its TCP/IP network stack, to MicroBlaze. With so many of today's consumer electronics devices accessing the Internet, efficient TCP/IP software is increasingly critical. The NetX TCP/IP stack will provide further support for MicroBlaze in consumer electronics products, and will enable developers to concentrate on their application code, getting their product to market faster.

To develop a successful, high-performance consumer, networking, or other electronic product around MicroBlaze processors, use an efficient RTOS like ThreadX by Express Logic. For further information on the ThreadX RTOS and other embedded software products from Express Logic, please visit www.expresslogic.com.