Express Logic's ThreadX RTOS for RISC-V

expresslogic

Background

- ThreadX® is a commercial RTOS from Express Logic, Inc., San Diego, CA.
- In production since 1997, ThreadX is used by major IoT product manufacturers in the areas of Consumer Electronics, Industrial Controls, Medical Devices, and Transportation.
- ThreadX has been used in the development of, and is currently deployed in over 6.2 billion products.

© 2017, Express Logic, Inc.

ThreadX Overview

- ThreadX is a small footprint, priority-based, fully preemptive RTOS, with a single linear address space.
- ThreadX offers services for creation and use of Threads, Message Queues, Semaphores, Mutexes, Timers, Memory Byte and Block Pools, and Event Flags.
- ThreadX has been safety certified to IEC-61508/62304/26262, etc., to SIL-4, Class C, ASIL D.
- In Addition, ThreadX offers advanced technology, including:
  - Preemption-Threshold Scheduling
  - Real-time Event Trace
  - Memory-Protected Modules

© 2017, Express Logic, Inc.

Preemption-Threshold™ Scheduling

- A technique to reduce context switches
- Preemption-Threshold establishes a priority ceiling for disabling preemption – preemption requires a priority higher (lower number) than the ceiling

  - For example, assume a thread's priority is 20, and its Preemption-Threshold is set to 15
  - Threads with priority lower than (larger number) 14 , even if higher than (smaller number) the running thread's priority (20), will not preempt the running thread

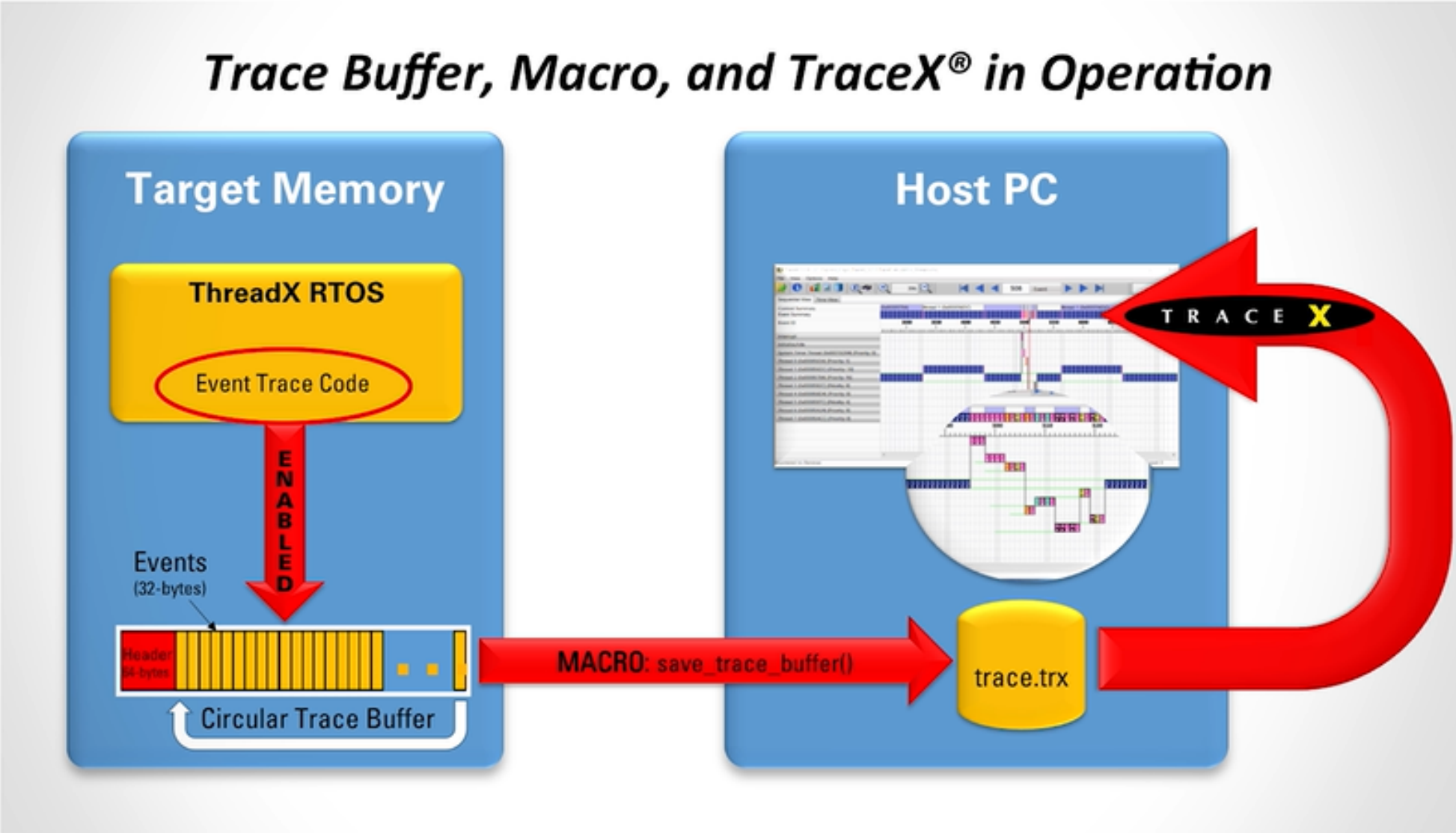
Preemption-Threshold

Priority

Priority	Comment
0	Preemption allowed for threads with priorities from 0 to 14 (inclusive)
:	
14	
15	Thread is assigned **Preemption-Threshold = 15** [this has the effect of disabling preemption for threads with priority values from 15 to 19 (inclusive)]
:	
19	
20	Thread is assigned **Priority = 20**
:	
31	

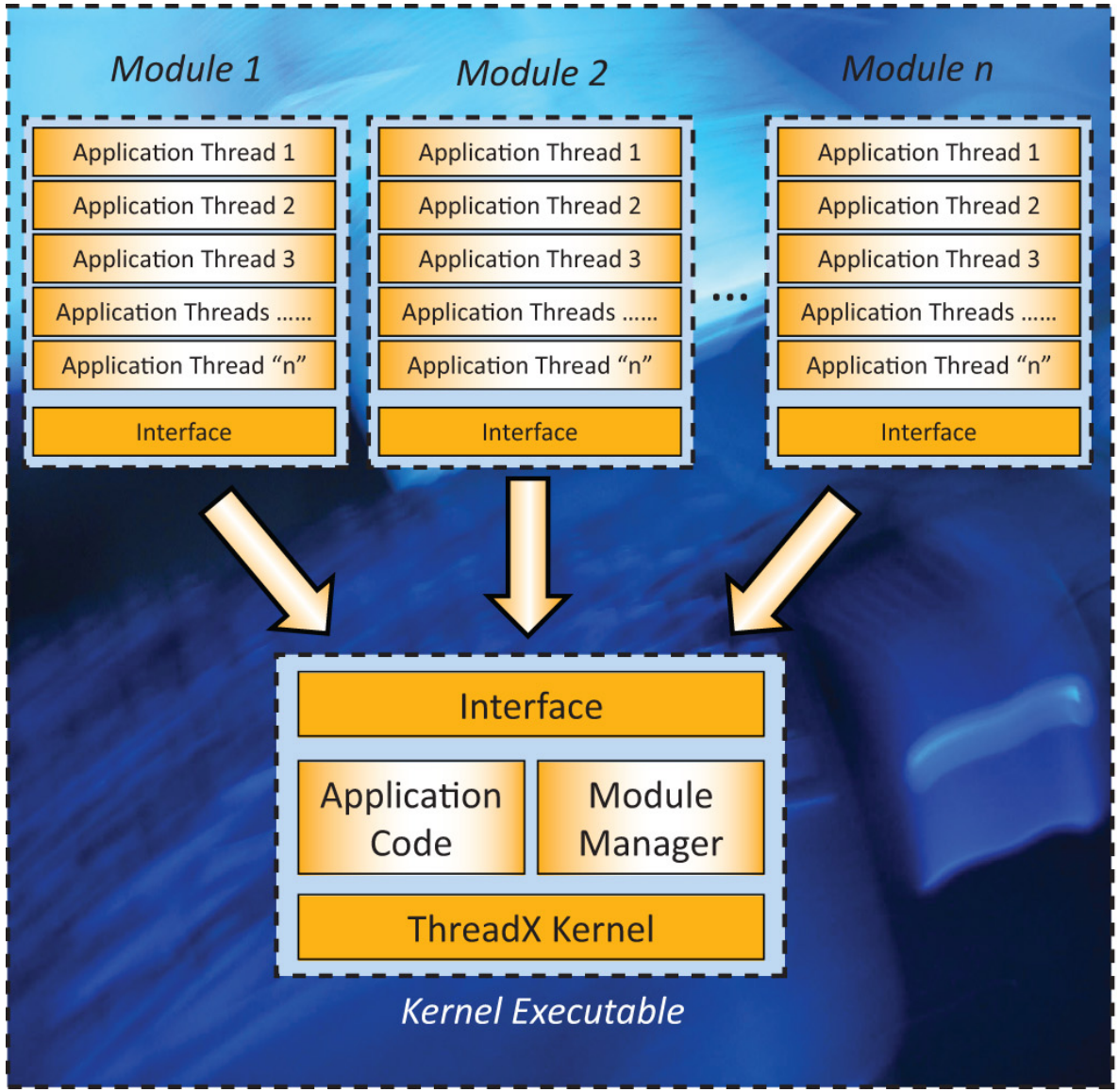
© 2017, Express Logic, Inc.

ThreadX Event Trace



© 2017, Express Logic, Inc.

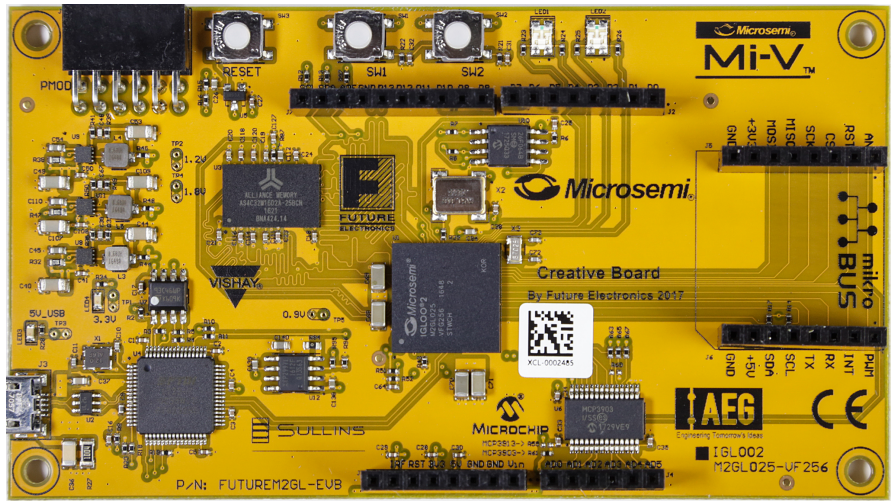
ThreadX Application Modules



© 2017, Express Logic, Inc.

Testing The Port

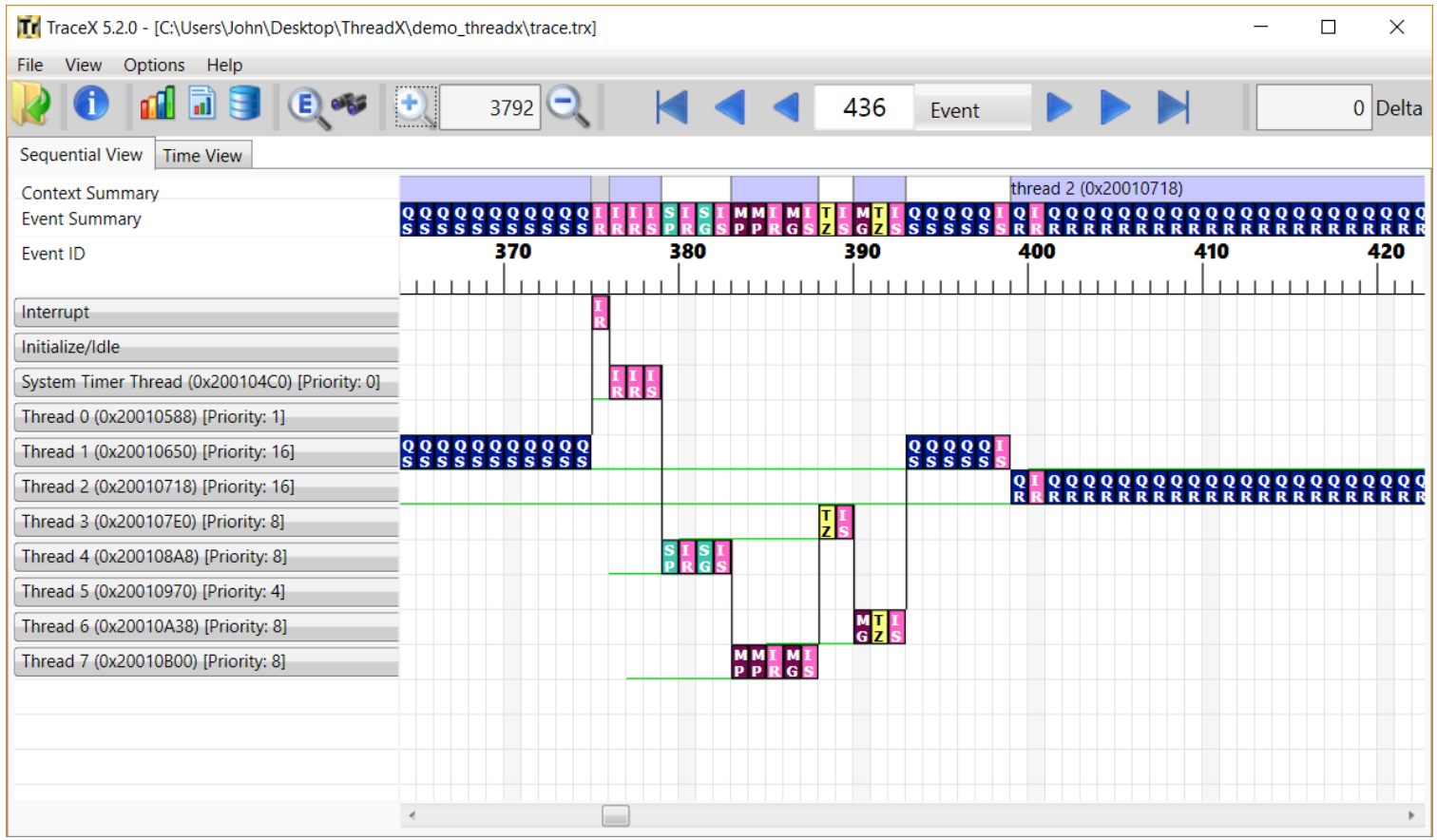
- The ThreadX / RISC-V port was developed and first tested in a software environment (SPIKE) that simulates the RISC-V.
- We then ran the port on a Microsemi Smartfusion2 Creative Development Board (FUTUREM2SF-EVB).
  - The RISC-V FPGA bit file image (*https:j/github.com/RISCV-on-Microsemi-FPGA/M2S025- Creative-Board.git*) was used for testing.
  - Hardware timers and a debug UART are provided as part of the kit.



© 2017, Express Logic, Inc.

Final Test

- Finally, we verified that the standard 8-thread ThreadX demo (demo_threadx.c) operates normally on RISC-V.



© 2017, Express Logic, Inc.

ThreadX Footprint

- **Instruction Area (ROM, Flash) Sizes**
  - Core Services: 2,082 Bytes
  - Queue Services: 858
  - Event Flag Services: 756
  - Semaphore Services 384
  - Mutex Services: 1,254
  - Block Memory Services: 472
  - Byte Memory Services: 820

- **RAM Sizes**
  - Global Data Area: 500 – 2,000 Bytes
  - Thread Control Block 176
  - Timer Control Block: 44
  - Queue Control Block: 56
  - Semaphore Control Block: 28
  - Mutex Control Block: 52
  - Event Flag Control Block: 36
  - Block Memory Control Block: 48
  - Byte Memory Control Block: 52

Note: Measurements for RISC-V were not available at press time. These measurements are typical for a 32-bit RISC processor.

© 2017, Express Logic, Inc.

ThreadX Service Times

Fast Execution (120MHz Processor)

	IR	TS	TR	TRCS
tx_thread_suspend	1.2µs	1.4µs	-	-
tx_thread_resume	-	-	1.0µs	1.9µs
tx_thread_relinquish	0.4µs	-	-	1.3µs
tx_queue_send	0.6µs	1.6µs	1.4µs	2.3µs
tx_queue_receive	0.5µs	1.6µs	1.7µs	2.6µs
tx_semaphore_get	0.3µs	1.6µs	-	-
tx_semaphore_put	0.3µs	-	1.2µs	2.0µs
tx_mutex_get	0.4µs	1.6µs	-	-
tx_mutex_put	0.5µs	-	1.8µs	2.6µs
tx_event_flags_set	0.6µs	-	1.5µs	2.4µs
tx_event_flags_get	0.6µs	1.8µs	-	-
tx_block_allocate	0.4µs	1.6µs	-	-
tx_block_release	0.4µs	-	1.1µs	2.0µs
tx_byte_allocate	1.2µs	1.6µs	-	-
tx_byte_release	0.9µs	-	2.7µs	3.4µs

Context Switch (CS):	1.0µs
Boot Time (BT):	3.3µs
Interrupt Latency Range (ILR):	0.0µs - 1.1µs

Note: Measurements for RISC-V were not available at press time. These measurements are typical for a 120MHz processor.

- **Immediate Response (IR)**: Time required to process the request immediately, i.e., no thread suspension or thread resumption.
- **Thread Suspend (TS)**: Time required to process the request when the calling thread is suspended due to unavailability of the resource.
- **Thread Resumed (TR)**: Time required to process the request when a previously suspended thread (of the same or lower priority) is resumed as a result of the request.
- **Thread Resumed and Context Switched (TRCS)**: Time required to process the request when a previously suspended higher-priority thread is resumed as a result of the request. Since the resumed thread is higher-priority, a context switch to the resumed thread is also performed from within the request.
- **Context Switch (CS)**: Time required to save current thread's context, find highest priority ready thread, and restore its context.
- **Boot Time (BT)**: Time required from tx_kernel_enter to the dispatch of the first thread.
- **Interrupt Latency Range (ILR)**: Amount of time interrupts are disabled.

© 2017, Express Logic, Inc.

Current Status and Resources

- We anticipate commercial availability of ThreadX for RISC-V before the end of 2017.
- Please contact Express Logic for further information:
  - info@expresslogic.com
  - 1-888-THREADX
  - 1-858-613-6640
  - www.rtos.com
- Thank you for your interest.

© 2017, Express Logic, Inc.