Altreonic's formalized technology gives more for less.

3rd August 2011. Altreonic is now announcing a port of OpenComRTOS to the high performance PowerPC processors of Freescale and integrating it in the OpenComRTOS Designer environment.

Full OpenComRTOS fits comfortably in L1 cache

A full kernel with all services only requires between 7.1 to 9.8 KBytes for program memory and less than 6 KBytes of data memory, depending on the compile time options and services used. This was measured by compiling a minimal application for an e600 target with Altivec support and comparing the results using a mapfile analyser. Nevertheless, this is still a complete priority based preemptive scheduling RTOS with support for distributed priority inheritance. Besides task scheduling, services provided are: events, semaphores, resources, port hubs, fifos, packet and memory pools in blocking, non blocking, blocking with timeout and asynchronous semantics. OpenComRTOS transparently supports single as well as large multiprocessor systems. Porting to the PPC has been swift and efficient.

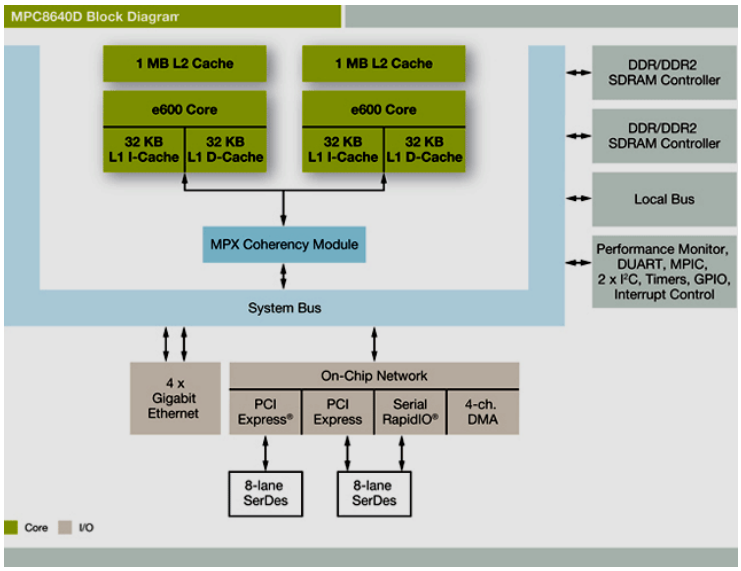

Figure. Block diagram of the Freescale MPC8640D 2-core processor chip.

For applications where performance is paramount, using less memory means higher performance and less energy consumption. The small code size of OpenComRTOS frees a lot more of the L1 cache for the user applications than traditional monolithic RTOS. If this allows reducing the clock frequency, even more power can be saved.

Formally developed and Transparent multi-processor support

What differentiates OpenComRTOS as well is that it was formally developed. Besides the inherent safety and security benefits of the code, this has also resulted in a very clean and unique scalable architecture, typically resulting in 10 times less code than a traditionally designed and functionally equivalent RTOS. In addition, with OpenComRTOS one can transparently program a multiprocessor architecture allowing to distribute the application over multiple cores, multiple

multi-core chips on a single board and over multiple boards even when geographically distributed. Small code size matters here as well, as it results in lower communication latencies.

Simultaneously, one of the nodes can be running a classical host-OS like Linux or Vx-Works transparently integrating with it. The latter allows any node full access in a transparent way to all host services like file I/O, Human Interface devices and network I/O.

OpenComRTOS Designer

OpenComRTOS Designer consist of a high level visual development environment (OpenVE) in which the user specifies application and target topology in an independent way allowing to simulate the application on his development PC. Code generators then generate most of the target specific C code and the build system. A new task level debugger and the visual OpenTracer allow examining and profiling the application at runtime. OpenComRTOS Designer also imports platform descriptions for complex boards, e.g. using the e500 based P4080 with all its I/O and communication options. The designer can opt either for a transparent use of the underlying communication network, either to dedicate some of the communication paths as direct point-to-point connections. The latter is sometimes beneficial for very high speed signal processing applications.

Other performance data:

Interrupt latency:

- IRQ (using a software interrupt) to ISR: 70 cycles
- IRQ to Task: 917 cycles.

Semaphore loop:

This is a small test program that consists of two tasks signaling and waiting on 2 semaphores. This is a stress test as it contains 4 context switches and 4 semaphore services: 1682 cycles.

More information can be found at following links: www.altreonic.com

Contact data:

Eric Verhulst,

eric.verhulst @ altreonic.com,

T.+32 477 608 339

Note: figures updated 10th August 2011