Outline

- The TOPPERS Project
- SystemBuilder: A System-Level Design Environment
- RTOS for MPSoCs and HW/SW Co-configuration
- Summary
The TOPPERS Project

Objectives

◆ TOPPERS = Toyohashi Open Platform for Embedded and Real-Time Systems

◆ Objectives of the Project
  - Developing various open-source software for embedded systems including RTOS and promoting their use.
  - Building an OS as predominant as Linux in the area of embedded systems!

◆ Promoting Force of the Project
  - Cooperation among industry, academia, and government, including individuals.
  - Before then, operated by voluntary organization led by ERTL.
Aims

◆ Building Definitive RTOS for the Present Generation
  ■ Based on the results of technical development achieved for the ITRON Specifications for the past 20 years

◆ Developing RTOS Technology of the Next Generation
  ■ Develop RTOS technology of the next generation that meets the requirements for embedded systems
  ■ It has little meaning to develop another OS that resembles Linux!

◆ Fostering Embedded System Engineers
  ■ Foster embedded system engineers by providing training materials using open-source software and opportunities to participate in training

ITRON Specifications

◆ What is the ITRON Specifications?
  ■ ITRON is a series of RTOS specifications and related standards for embedded systems (esp. small-scale embedded systems) developed by the ITRON project.
    • ITRON is just specification, not software.
  ■ Open specification policy
    • RTOS confirming to the ITRON specifications it not necessary open or free.

◆ Current Status of the ITRON Specifications
  ■ Most widely used RTOS specification in Japan
    • 30 – 40% of embedded systems
  ■ Widely used especially in consumer applications
History and Members

◆ Brief History
  ■ Nov. 2000
    • First version of JSP kernel released from Toyohashi Univ. of Tech.
  ■ Nov. 2001
    • Number of project member organizations becomes 4
  ■ Sep. 2003
    • Incorporated as an NPO

◆ Project Members
  ■ Total Number of Project Members: about 100
    • about 55 companies
    • 5 university or technology college labs.
    • 3 public research institutes
    • about 40 individuals

Achievements

◆ TOPPERS/JSP Kernel
  ■ Real-time kernel that conforms to the standard profile of the µITRON4.0 Specification
  ■ First product of the TOPPERS Project (released as open-source software in Nov., 2000)

◆ TOPPERS/FI4 Kernel
  ■ Real-time kernel that implements all functions defined in the µITRON4.0 Specification

◆ IIMP Kernel
  ■ An extension of JSP Kernel with protection functions conforming to the µITRON4.0/PX Specification

◆ TOPPERS/OSEK Kernel (tentative name)
  ■ OSEK-conformant kernel under development
Achievements (Cont.)

- **TINET**
  - Compact TCP/IP protocol stack running on JSP Kernel conforming to the ITRON TCP/IP API Specification
  - IPv6 version will also come soon
- **RLL (Remote Link Loader)**
  - Dynamic module loading mechanism
- **DLM (Dynamic Loading Manager)**
  - Another dynamic module loading mechanism
- **TOPPERS C++ API Template Library**
- **TOPPERS Kernel Testsuites**
- **Bridge Point for TOPEPRS/JSP**

TOPPERS/JSP Kernel

- **JSP = Just Standard Profile**
  - Real-time kernel compliant with the µITRON4.0 Specification
  - As the name shows, it has only Standard Profile functions (In strict, it has a few extensions)
  - First version was released on Nov., 2000
  - The latest version is Release 1.4
- **Original Purpose of the Development**
  - platform of education and research
  - evaluation of µITRON4.0 Specification
  - reference implementation of µITRON4.0
  - use in industries
- **The development of the kernel itself is not research, though developed by a university lab.**
TOPPERS/JSP Kernel (Cont.)

- **Target Processors** (as of the date of Release 1.4)
  - Motorola M68040, RENESAS SH1, SH3/4, H8, M32R, ARMv4 (ARM7, ARM9), MIPS3 (VR4131, VR5500), Xilinx MicroBlaze, TI TMS320C54x, SANYO XStormy16, Intel i386, NEC V850 (supported in Release 1.3), Tensilica Xtensa (supported by members)

- **Simulation Environments**
  - Simulation environments on Windows and Linux supported

- **Software Development Environments**
  - GNU software development tool is the standard
  - GHS development environment is supported
  - Can support other development environments

TOPPERS/JSP Kernel: Advantages

- **easy to understand and easy to modify**
  - very important feature when the software is used for educational and research purposes

- **easy to port to new target processors/systems**
  - Target-dependent part is clearly separated from target-independent part and is kept small (porting work as short as 3 day!).

- **low overhead and small footprint**
  - The size of task control block (TCB) is just 32 bytes (with 32-bit integer and pointer).

- **simulation environments on Windows and Linux**

- **complete free/open source solution**
  - using GNU tools as standard software tools
Application Example (1)

- Ricoh has used Windows simulation environment of JSP Kernel, and built an evaluation environment on PC to test codes created by object-oriented design tools.
  - TOPPERS Project, TOYO Corporation, Japan Rational Software (now Japan IBM) cooperated
  - Supports both Rose RealTime (IBM, formerly Rational Software) and BridgePoint (Project Technology)
  - Possible to evaluate and debug software on PC as a preliminary process, before evaluating with real target processors.

Application Example (2)

- Karaoke microphone “Do! Karaoke” made by Matsushita Electric Industrial Co., Ltd.
  - Panasonic SD Karaoke microphone “SY-MK7-S” and duet microphone “SY-DK7-S” (released by Matsushita in Feb. 2003)
Comparison with Linux

- Linux is a good success model for TOPPERS.

- Focusing on Embedded Systems
  - Linux technologies come from general-purpose computing (PC and workstations), while TOPPERS Project aims to develop solutions focusing on embedded systems.

- License Conditions
  - TOPPERS Project adopts an original license conditions (called TOPPERS License) rather than GNU GPL (General Public License).
  - GNU GPL is not designed for embedded systems and sometimes becomes an obstacle for adoption.

Basic Concept of TOPPERS License

- Considering the features of the embedded systems, the condition should be freer than GNU GPL and the BSD license.

- We want to know where and how the software is used for the accountability.
  - “reportware”

- Dual license is adopted for permitting to link the TOPPERS software with GNU software.
Summary

- The TOPPERS project is developing and distributing open-source RTOS for embedded systems based on the ITRON specification.
- The project also aims to develop next generation RTOS technologies exploiting the features of embedded systems (rather than adopting existing technologies developed for general-purpose systems).
  - raising the quality and value of the embedded system applications (electronic appliance, etc.)
- The project is now a joint project of industries and academia with focus on business promotion as well as research and education.

SystemBuilder: A System-Level Design Environment
SystemBuilder

System-level design environment from system description to FPGA implementation.

Main Features

- System description in C
- SW/HW partitioning by human designers
- Automatic SW/HW interface synthesis
- Automatic software synthesis
- Automatic behavioral synthesis with a commercial tool
- SW/RTOS/HW cosimulation at various abstraction levels
- FPGA implementation

Design Flow
System Description

Function Unit (FU)
- Software: Task
- Hardware: Module

Communication Primitives (CP)
- Non Blocking Communication (NBC)
- Blocking Communication (BC)
- Memory (MEM)

Software/Hardware Partitioning
- Specify in SDF file
- The implementation place of CP is automatically determined with tool.
  SW = FU1, FU2
  HW = FU3, FU4
  SW = FU1
  HW = FU2, FU3, FU4
## System DeFinition File (SDF)

- Specify
  - name and type of function units
  - communication channels
  - SW/HW partitioning

```plaintext
SYS_NAME = test
SW = FU1, FU4
HW = FU2, FU3

BCPRIM cp1, SIZE = 32
BCPRIM cp2, SIZE = 32
NBCPRM cp3, SIZE = 32
MEMPRIM cp4, SIZE = 32
NBCPRM cp5, SIZE = 16

BEGIN_FU
NAME   = FU1
FILE   = "fu1.c"
USE_CP = cp1(INOUT), cp3(INOUT),
        cp5(IN)
END

BEGIN_FU
NAME   = FU2
FILE   = "fu2.c"
USE_CP = cp1(IN), cp4(OUT), cp5(IN)
END

BEGIN_FU
NAME   = FU3
FILE   = "fu3.c"
USE_CP = cp2(OUT), cp4(IN), cp5(IN)
END

BEGIN_FU
NAME   = FU4
FILE   = "fu4.c"
USE_CP = cp2(INOUT), cp3(IN)
END
```

## Functional Cosimulation

- Project files for simulation are automatically generated.
- TOPPERS-Win is used as a simulation engine.
**System Synthesis**

**Software**
- Software function units are translated into tasks on ITRON RTOS.

**Interface**
- Interface specification file is generated, which will be fed by our interface synthesizer.

**Hardware**
- The followings are generated:
  - HW/HW interface (memory, handshake, etc.)
  - HW top module
  - Project files for behavioral synthesis

---

**HW/HW Interface and Top Module**
Behavioral Synthesis

- SysGen calls eXCite for synthesizing hardware function units
- eXCite: A commercial behavioral synthesizer from YXI
  - Generates RTL VHDL from C code

```
main()
int a, b, c, d;
  while(1){
    yx read(1, &c...);
    a = x + c;
    b = y + c;
    yx write clt(?, &a, ...);
    yx write clt(2, &b, ...);
  }
```

BusConnecter: Interface Synthesizer

**Software**
- Device drivers for each communication channel
- Address map of the devices and bit assignment of interrupt registers are automatically decided.

**Hardware**
- Bus interface
- Interrupt request management circuits

**Project files for cosimulation**
- RTOS Simulator on Windows (TOPPERS-Win)
- ISS (ARMulator)
- HDL Simulator (Active-HDL, ModelSim)

**Microblaze peripheral management files**
- MPD, PAO File, and template for MHS
Cosimulator Overview

- The RTOS simulator, HDL simulators, and C/C++/VB models run concurrently with communicating with each other.
- Communication is based on the Windows COM technology.
Timed Cosimulation

Cycle-Accurate Cosimulation
FPGA Platform: Xilinx Microblaze

- Softcore processor and its design environment
- Easy and flexible configuration of peripherals

**Microblaze Hardware Specification (MHS)**

- From the MHS file, an HDL description for connecting the peripherals are automatically generated.

---

Implementation

---
Summary

- SystemBuilder: A system-level design environment
- Synthesis from specification in C to FPGA implementation
  - Interface synthesis
  - Software synthesis
  - Behavioral synthesis
- Cosimulation at various abstraction levels
  - Based on the TOPPERS/JSP kernel simulator.

RTOS for MPSoCs and HW/SW Co-configuration
Backgrounds

Multiprocessor systems have become popular in embedded systems to achieve higher performance and/or lower power consumption

- Ex. Cellular Phones
  - An application processor (media processor) and a communication processor(s)

Multiprocessors for embedded systems are often heterogeneous and highly optimized for specific applications.

System Architecture Examples

[Diagrams showing different system architectures with labeled components such as PE, MEM, Sensor, LAN, DMAC, PE1, Sensor, DMAC, PE2, PE3, MEM, LAN, FIFO, PE1, Sensor, DMAC, PE2, PE3, MEM, LAN]
Traditional Programming Style

- An RTOS runs on each processor.
- Communications and synchronization between processors are often specified in application software.
  - Designing application software is time consuming and error-prone.
  - The application software is hardly reusable.

Goals

- Efficient and configurable RTOS for heterogeneous multiprocessor embedded systems.
- Co-configuration of RTOS and hardware architecture.

Assumptions as a First Step

- Homogeneous multiprocessors
- Each processor has local shared memory
- Application tasks are statically assigned to the processors.
FPGA-Based Platform

- Developed an FPGA-based board used for our platform.
  - 300 million gate FPGA
    - Up to four MicroBlaze processors
  - Four sets of processor resources
    - SRAM, RS-232C 2ch, etc.
  - Ethernet I/F, PCI I/F, etc.

RTOS Modification for MPSoC

- The TOPPERS/JSP kernel has been modified for MPSoC.
- A task runs on a specified processor, but it can be manipulated (activated, terminated, etc.) by other processors.
- Every RTOS resource (e.g., semaphore, mailbox, data queue, etc.) is owned by one processor, but can be accessed by any other processors.
  - RTOS resources have processor ID.
- Hardware is also modified to support spin lock (test & set).
RTOS Object Model

Synchronization/Communication Objects (semaphores, data queues, mailboxes, etc.)

RTOS/HW Co-configuration

Processor A {
    CRE_TASK(TASK_A, {...});
    CRE_SEM(SEM_A, {...});
    ...
} 

Processor B {...}
Processor C {...}
Processor D {...}

Co-configuration

MicroBlaze MHS File

Xilinx Tools

Binary Code
Summary

◆ The TOPPERS Project
◆ SystemBuilder: A System-Level Design Environment
◆ RTOS for MPSoCs and HW/SW Co-configuration

◆ TOPPERS software and documents are available online at
  http://www.toppers.jp/
  ■ But, most documents are in Japanese only.