Porting VME-Based Optical-Link Remote I/O Module to a PLC Platform
- an Approach to Maximize Cross-Platform Portability Using SoC

T. Masuda, A. Kiyomichi
Japan Synchrotron Radiation Research Institute (JASRI)
Outline

• Background
  – Optical-linked remote I/O system
  – Platform consideration

• Development of module
  – Design policy
  – Hardware implementation
  – Software implementation
  – Implementation of FPGA logic

• Summary
Background

• VME
  – employed at SPring-8 as FE computers.
• Optical-linked Remote I/O systems
  – utilized to cover widely distributed accelerator equipment.
  – consist of VME-based master boards and several kinds of slave boards.
  – two types of optical-linked remote I/O system
    • RIO system
    • OPT-VME system
Background

• **RIO system**
  – developed by Mitsubishi Electric Co.
  – used since 1997, already **discontinued**.
  – employ **over 1,400 slave boards** in SPring-8.
    • mainly for SR magnet power supplies control.
    • many of them can be replaced with **OPT-VME system**.
      – developed the compatible slave boards.
Background

- **OPT-VME system**
  - developed by SPring-8 at 2001.
  - two types of **VME-based** master boards.
    - OPT-VME
    - OPT-CC
Background

- **OPT-VME system**
  - developed by SPring-8 at 2001.
  - two types of **VME-based** master boards.
    - OPT-VME
    - OPT-CC
  - employ **over 400 slave boards** in SPring-8.
    - 10 types of slave boards are available.
Background

- **OPT-VME system**
  - developed by SPring-8 at 2001.
  - two types of **VME-based** master boards.
    - OPT-VME
    - OPT-CC
  - employ **over 400 slave boards** in SPring-8.
    - 10 types of slave boards are available.
  - original communication protocol (OPT-Protocol 2006)
    - Only support point-to-point connection.
Background

- **OPT-VME system**
  - OPT-CC
    - also available in the relay-mode.
      - max. 132 slave boards can be controlled from a master.
Background

- **OPT-VME system**
  - OPT-CC
    - also available in the relay-mode.
      - max. 132 slave boards can be controlled.
    - realized by switching control I/F.
Background

- **OPT-VME system**
  - OPT-CC
    - also available in the relay-mode.
      - max. 132 slave boards can be controlled.
    - realized by switching control I/F.
Background

- **OPT-VME system**
  - OPT-CC
    - also available in the relay-mode.
    - max. 132 slave boards can be controlled.
    - realized by switching control I/F.

In a relay mode
Background

• **OPT-VME system**
  – OPT-CC
    • also available in the relay-mode.
      – max. 132 slave boards can be controlled.
    • realized by switching control I/F.
    • communication procedure with the remote slave board is *implemented in the device driver* for the master board.
Background

- **OPT-VME system**
  - OPT-CC
    - also available in the relay-mode.
      - max. 132 slave boards can be controlled.
    - realized by switching control I/F.
    - communication procedure with the remote slave board is **implemented in the device driver** for the master board.

The device driver is responsible for the high-level communication procedures including the remote slave control.
Platform Consideration

- **VME**
  - Passed over 30 years, become out-of-date.
  - Two major issues;
    - Lack of bandwidth.
    - Discontinued the de-fact standard bus-bridge chip *Tsi148*.
      - *This has been a big problem for VME users.*
  - considering the next-generation alternative platform.
Platform Consideration

• **MTCA.4**
  
  – Decided to introduce MTCA.4 as a high-end platform.
  
  – Analog-based old SR LLRF system controlled by VME is planned to be replaced with MTCA.4-based digital LLRF system.
Platform Consideration

• **MTCA.4**
  – Decided to introduce MTCA.4 as a high-end platform.
  – Analog-based old SR LLRF system controlled by VME is planned to be replaced with MTCA.4-based digital LLRF system.

• **Linux PLC (Programmable Logic Controller)**
  – one of the candidate to cover a low-end side.
  – e.g. e-RT3 (FA-M3) by Yokogawa Electric Co.
    • already applied as front-end computers in both SPring-8 and SACLA.
Background

• Developed the new master module of the OPT-VME system based on the e-RT3 platform.
  – To effectively utilize the resources of large amount of OPT-VME slave board (~400).
    • RIO slave boards (~1,400) are also integrated by replacing OPT-VME based compatible slave boards.
  – Considering alternative platform portability such as a PCI Express (MTCA.4)
Development of the new master module

• **OPT-PLC**
  – e-RT3-based new master module for the *OPT-VME system.*

<table>
<thead>
<tr>
<th>SoC</th>
<th>Xilinx Zynq 7015 : XC7Z015-1CLG485C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>1GB DDR3-SDRAM</td>
</tr>
<tr>
<td></td>
<td>128MB QSPI Flash</td>
</tr>
<tr>
<td>LAN</td>
<td>1 port (RJ-45 Connector)</td>
</tr>
<tr>
<td>MicroSD</td>
<td>1 port (Micro-SD socket)</td>
</tr>
<tr>
<td>UART</td>
<td>1 port (Micro-USB connector)</td>
</tr>
<tr>
<td>High-Speed Serial I/F</td>
<td>4 pairs x 6.25GBps in a 70pins stacking connector (Molex 53625-0774)</td>
</tr>
<tr>
<td>JTAG</td>
<td>1 port</td>
</tr>
<tr>
<td>Power</td>
<td>+5V±5%</td>
</tr>
</tbody>
</table>
OPT-PLC module

• Design Policies

1. Equip with as many optical channels as possible.
2. Separate an I/O unit from a logic control unit.
3. Control the module using the e-RT3 general-purpose device driver.
4. Control the module from a sequence CPU in addition to a Linux CPU.
OPT-PLC module

• Design Policies

1. Equip with as many optical channels as possible.
2. Separate an I/O unit from a logic control unit.
3. Control the module using the e-RT3 general-purpose device driver.
4. Control the module from a sequence CPU in addition to a Linux CPU.

→ Hardware Implementation
Hardware Implementation

- Consists of three PCBs.
  - Separate two I/O boards from the control logic board.
  - Connected using 70 pins stacking connector each other.
    - PCB is a little small to mount the FMC.
- Equipped with 5 optical channels.
OPT-PLC module

• Design Policies

1. Equip with as many optical channels as possible.
2. Separate an I/O unit from a logic control unit.
3. Control the module using the e-RT3 general-purpose device driver.
4. Control the module from a sequence CPU.

Implementation of Software & FPGA logic
OPT-PLC module

• Design Policies

1. Equip with as many optical channels as possible.
2. Separate an I/O unit from a logic control unit.
3. **Control the module using the e-RT3 general-purpose device driver.**
4. Control the module from a sequence CPU in addition to a Linux CPU.

→ Implementation of Software & FPGA logic

**Keyword : SoC**
e-RT3 General-Purpose Device Driver

- supplied and supported by Yokogawa Electric Co.
- primitive device driver to handle memory access and interrupt.
e-RT3 General-Purpose Device Driver

- supplied and supported by Yokogawa Electric Co.
- primitive device driver to handle memory access and interrupt.

Background

- **OPT-VME system**
  - OPT-CC
    - also available in the relay-mode.
      - max. 132 slave boards can be controlled.
    - realized by switching control I/F.
    - communication procedure with the remote slave board is implemented in the device driver for the master board.

The device driver is responsible for the high-level communication procedures including the remote slave control.
e-RT3 General-Purpose Device Driver

- supplied and supported by Yokogawa Electric Co.
- primitive device driver to handle memory access and interrupt.

How do we implement this high-level communication procedures?

The device driver is responsible for the high-level communication procedures including the remote slave control.

Background

- OPT-VME system
  - OPT-CC
    - also available in the relay-mode.
    - max. 132 slave boards can be controlled.

OPT-CC: master mode
OPT-CC: relay mode
slave board
Software Implementation

- Adopt SoC (Xilinx Zynq 7000)
- Implement the high-level communication procedures as application software running on ARM Linux in SoC.

API functions for OPT-VMEs

\[
\text{ioctl(fd, request, args)}
\]

Device driver for OPT-VME master

Communication Procedures

VME CPU Board (Solaris)

Linux CPU module

ARM processor in Zynq SoC

Communication procedure process

device driver

DPRAM

Descriptor Area

request

args

....
Software Implementation

- Adopt SoC (Xilinx Zynq 7000)
- Implement the high-level communication procedures as application software running on ARM Linux in SoC.

VME CPU Board (Solaris)

API functions for OPT-VMEs

ioctl(fd, request, args)

device driver for OPT-VME master

Communication Procedures

Linux CPU module

API functions for OPT-VMEs
OPT-PLC interface functions

Device driver

Communication procedure process x5

ARM processor in Zynq SoC

Software Implementation

- Adopt SoC (Xilinx Zynq 7000)
- Implement the high-level communication procedures as application software running on ARM Linux in SoC.
Software Implementation

- Adopt SoC (Xilinx Zynq 7000)
- Implement the high-level communication procedures as application software running on ARM Linux in SoC.
Software Implementation

- Adopt SoC (Xilinx Zynq 7000)
- Implement the high-level communication procedures as application software running on ARM Linux in SoC.

![Software Implementation Diagram]

- API functions for OPT-VMEs
  - `ioctl(fd, request, args)`

- Device driver for OPT-VME master

- Communication Procedures

- Linux CPU module
  - API functions for OPT-VMEs
  - OPT-PLC interface functions
  - e-RT3 general-purpose device driver

- ARM processor in Zynq SoC
  - Communication processor process
  - Entire system control process
  - DPRAM
    - Descriptor Area
    - `request`
    - `args`
    - `...`
Software Implementation

- Adopt SoC (Xilinx Zynq 7000)
- Implement the high level communication procedures as application software running on ARM Linux in SoC.

As a result the device driver of platform side is simplified, the module portability to other platform is enhanced.

API functions for OPT-VMEs

```c
ioctl(fd, request, args)
```

Device driver for OPT-VME master

Communication Procedures

API functions for OPT-VMEs

OPT-PLC interface functions

e-RT3 general-purpose device driver

Descriptor Area

- `request`
- `args`
- ...
Porting the VME-Based Optical-Link Remote I/O Module to a PLC Platform
- an Approach to Maximize Cross-Platform Portability Using SoC
Implementation of FPGA Logic
Implementation of FPGA Logic

Porting the VME-Based Optical-Link Remote I/O Module to a PLC Platform
- an Approach to Maximize Cross-Platform Portability Using SoC
Implementation of FPGA Logic

We can port the FPGA logic to an alternative bus by replacing this part.
Summary

- We have successfully ported the VME-based optical-link remote I/O module to the e-RT3 platform.
- The developed module OPT-PLC is equipped with Zynq 7000 SoC to build the communication procedures as the application S/W on the ARM Linux.
  - the interface with the PLC bus is simplified and the e-RT3 general-purpose device driver is available.
- We can port the developed FPGA logic to an alternative bus e.g. the PCI express by replacing the PLC bus interface block in the PL part.
- The interface simplification enhances portability.
Thank you for your attention.