Responsive Processor
for Parallel/Distributed Real-Time Control

Nobuyuki Yamasaki
Keio University / Electrotechnical Laboratory
E-mail: yamasaki@{ics.keio.ac.jp, etl.go.jp}
http://www.ny.ics.keio.ac.jp
Abstract

Responsive Processor for Parallel/Distributed Real-Time Control integrates:

- Processing Core (SPARC)
- Responsive Link (Real-Time Communication)
- Computer I/O (SDRAM I/F, DMAC, PCI, USB, PIO, SIO, Timers, Counters, …)
- Control I/O (ADC, DAC, PWM Generators, Pulse Counters, …)

System-on-a-chip for Parallel/Distributed Real-Time Control

Combination with some processors

Realization of various kinds of control systems
Components of Responsive Processor

- **Responsive Link** (Real-time communication)
- **Control I/O** (ADC, DAC, PWM Generators, Counters, ...)
- **Processing Core** (SPARC)
- **Computer I/O**
- **Responsive Processor**
- **Motor Sensor** ...
- **SDRAM**
- **PCI**
- **USB...**
Responsive Processor
Applications

- Robots
- Office Automation
- Factory Automation
- Home Automation
- Intelligent Buildings
- Amusement Systems, ...

Functionally distributed control

Original embedded chip with standard interface
Event Processing for Embedded Control

- Embedded control systems must respond to external inputs (sensors, user inputs, etc.) and operate according to the inputs.
- External inputs (interrupts) may occur frequently and continuously.
- Short response time
- Short processing time
- Control of devices (motors, sensors, etc.)
- Time limitation.
Responsive System

Responsive = Reactive + Real-time

- **Reactive**: A characteristic in which a reflexive action occurs according to an external input

- **Real-Time**: A characteristic in which the exactness of the system depends on not only the result but also the time it took to achieve the result
Design Policy of Responsive System

Basic Strategy by Hardware

- Shorter response time to external events
- Shorter processing time for the events

Basic Tactics by Hardware

- Immediate transmission of events
- Parallel processing with respect to different kinds of events
- Concurrent processing in regard to the same kind of events
How to Realize a Responsive System

- Generation of an event from a sensor
  - Immediate transmission to the proper module to process the event
- Generation of several events simultaneously or continuously
  - Different kind of several events
    - Parallel processing by function-specific modules
    - Reduction of both processing and response time
  - The same kind of several events
    - Concurrent processing
    - Reduction of response time
Function-specific Parallel Computer

- Each module controls different I/O functions
- Coarse grain function-specific tasks
- Each module can be treated as an agent
- MIMD

Parallel computer architecture for embedded control systems
Current Real-Time Communication

- IEEE-1394 (FireWire, iLink, DV)
- USB
  - Soft Real-Time Communication: Isochronous transfer
  - Error correction is not supported in case of the isochronous mode.
  - Central Control: Low robustness
  - Maximum connection number is limited:
    (IEEE-1394: 63, USB: 127)
  - Topology is fixed. (Tree structure)
**Trade-off on Real-Time Communication**

- **Soft real-time**: Guarantee of bandwidth
  - Maximization of Throughput
- **Hard real-time**: Guarantee of latency
  - Minimization of Latency

<table>
<thead>
<tr>
<th>PACKET SIZE</th>
<th>LARGE</th>
<th>SMALL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Throughput</strong></td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Latency</strong></td>
<td>Long</td>
<td>Short</td>
</tr>
</tbody>
</table>
Responsive Link for Real-Time Communication

- **Split transmission of data and events**
- **Fixed packet size**: 64B data, 16B event
- **Full-duplex and differential I/F**
  - Hardware routing
    - Independent routing of data and events
    - **Cut-through switch with overtaking function** (The packet with higher priority can overtake other packets at each node.)
    - **Priority replacement** (Packet priority can be replaced with new priority at each node.)
    - When the network address is same but priority is different, the different route can be reset to realize exclusive lines or detours.
  - **Automatic error correction**
- **Flexible link speed** (12.5 to 100M bps)
- **Point-to-point link configurable for any topology**
Split Lines for Event and Data

- **Shared traffic**
  - indefinite latency and throughput

- **Event link**
  - Low Latency

- **Data link**
  - High Throughput
**Data Link**

Soft real-time communication for bulky data

- Multimedia data transmission (images, voice, etc.)
- Relatively large fixed packet size (64B)
- Total throughput is important

**Event Link**

Hard real-time communication for control

- Inter-processor interrupt and synchronization
- Relatively small fixed packet size (16B)
- Low latency for relatively few packets is important
Packet Format

Data Packet Format (64B)

<table>
<thead>
<tr>
<th>Source Addr.</th>
<th>Destination Addr.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Control &amp; Status</td>
<td></td>
</tr>
</tbody>
</table>

Event Packet Format (16B)

<table>
<thead>
<tr>
<th>Source Addr.</th>
<th>Destination Addr.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Control & Status Format (32bits)

<table>
<thead>
<tr>
<th>16 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>Full</td>
</tr>
<tr>
<td>Data Length</td>
</tr>
</tbody>
</table>

Frame Format (16bits)

<table>
<thead>
<tr>
<th>Data Bits</th>
<th>Redundancy Bits</th>
</tr>
</thead>
</table>
Cut-through Switch with Overtaking Function
Control of Overtaking Buffers
It is possible to set different routes in case of the same network address when the priorities are different. (Default route is priority 0.)

It is possible to replace a packet priority with a new priority at each node.
Routing according to Priority

Source

Data (Priority0)
Data (Priority1)

Event (Priority2)
Event (Priority0)

Destination
Low Level Communication

- Forward Error Correction (FEC)
  - Cyclic hamming code
  - 8bit data + 4bit redundant code
- Bit stuffing
- NRZI (Non Return to Zero Inverted)
- DPLL (Digital Phase-Lock-Loop)
- Synchronous frame (Setup Pattern)
- Flexible link speed (100Mbaud, 50Mbaud, 25Mbaud, 12.5Mbaud)
Responsive Link I/F

Responsive Link Connector

Tx Data+
Tx Data-
Rx Data+
Rx Data-

Data Link

Rx Event+
Rx Event-

Event Link

Responsive Link Cable
Required Functions

A questionnaire survey by using robotics and control mailing lists

Required functions, developing environments, operating systems, etc.
Functions of Responsive Processor

- Processing Core (SPARC 100MHz)
- Power Management Unit (100, 80, 60, 40, 20 [MHz], Sleep)
- MMU (64way)
- Responsive Links (4 links, 200, 100, 50, 25 [MHz])
- DMAC (4channels, Bus swapping, Bus sizing)
- SDRAM I/F (2channels, 100MHz)
- PCI I/F (Master/Target)
- USB I/F (Function, Hub)
- PWM Generators (50MHz, 9channels)
- Pulse Counters (24bit, 9channels)
- Timers/Counters (16bit, 4channels)
- Real-Time Clock
- A/D Converters (10bit, 8channels)
- D/A Converters (8bit, 2channels)
- Interrupt Controllers (43channels)
- SIO (RS-232C, 2channels)
- PIO (16bit), ...
Diagram of Responsive Processor
Hardware Design Rules

- Process: 0.35 μm, CMOS, 4 layered metal
- Usable gates: 2,378 k gates
- Die size: 14.5 mm x 14.5 mm = 210 mm²
- Package: 416pin BGA (40 mm x 40 mm)
- Voltage: 3.3 V
- Max. power: 2 W
Responsive Link

Routing Table

Overtaking Buffer for Data

Overtaking Buffer for Event

Communication Buffer (DPM)

MMU

DMAC

SDRAM I/F

PCI

USB

ADC,DAC

SPARC lite

I/O Controller
(IFC, SIO, PIO, Timer, Counter, PWM, etc.)

(I R C , S I O , P I O , T i m e r , C o u n t e r , P W M , e t c . )
Responsive Processor
Performance (Speed v.s. Power)

Performance of MPU

<table>
<thead>
<tr>
<th>Clock (MHz)</th>
<th>100</th>
<th>80</th>
<th>60</th>
<th>40</th>
<th>20</th>
<th>Sleep</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed (MIPS)</td>
<td>121</td>
<td>97</td>
<td>73</td>
<td>48</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>Power (W)</td>
<td>1.0</td>
<td>0.8</td>
<td>0.6</td>
<td>0.4</td>
<td>0.2</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Performance of Responsive Link

<table>
<thead>
<tr>
<th>Clock (MHz)</th>
<th>200</th>
<th>100</th>
<th>50</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Speed (Mbaud)</td>
<td>100</td>
<td>50</td>
<td>25</td>
<td>12.5</td>
</tr>
<tr>
<td>Speed of Data (Mbps)</td>
<td>67</td>
<td>33</td>
<td>17</td>
<td>8</td>
</tr>
<tr>
<td>Latency of Event (sec)</td>
<td>3.1</td>
<td>6.2</td>
<td>12.5</td>
<td>25</td>
</tr>
<tr>
<td>Power (W)</td>
<td>0.2</td>
<td>0.1</td>
<td>0.05</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Latency of Event (Worst) = 1 (sec) + 2 (sec/hop) x n (hop)
ControlBoards

- PCI card (PCI half size)
- CardBus card (PCMCIA size)
- Embedded board (Credit card size)
Developing Environment

- Cross development on Windows PC via PCI, USB, or RS-232C
- GNU tools (gcc, as, ld, make, etc.)
- Source level debugging by GDB
- Host OS : Linux, FreeBSD, Solaris, Windows
volatile unsigned long *ptr;
int i, j;
char buf[BUFSIZE];
char *token;

ASI4put(0x03010000, 0xffffffff);

while( 1 ) {
    printf( "MON>" );
    if( !fgets( buf, BUFSIZE, stdin ) )
        break;
    edit( buf );
    if( !(token = strtok( buf, " \
        usage( "mon" );
        continue;
    } type = *token;
    if( !(token = strtok( NULL, " \
        usage( "mon" );
        continue;
    
printf( "MON>" );
    if( !fgets( buf, BUFSIZE, stdin ) )
        break;
    edit( buf );
    if( !(token = strtok( buf, " \
        usage( "mon" );
        continue;
    
(gdb) break main
(gdb) break main
(gdb) break main
Note: breakpoint 1 also set at pc 0x2010280.
(gdb) j main
Continuing at 0x2010280.
(gdb) step
ASI4put(0x03010000, 0xffffffff);
(gdb) while( 1 ) {
    printf( "MON>" );
    (gdb) |
Operating Systems

Commercial

- VxWorks
- pSOSSystem
- iTRON
- OS-9

Research

- RT-Mach
- PULSER
- RT-Linux
Standardization

ISO/IEC JTC1 SC25 WG4 Responsive Link SG

Responsive Link SG: Matsushita Electric Industrial Co., Mitsubishi Electric Corporation, Fujitsu Limited, Hitachi, STARC, Electrotechnical Laboratory, Keio University, Kyusyu University

Conclusions

Responsive Processor for Parallel/Distributed Real-Time Control integrates:

- Responsive Link
- Processing Core (SPARC)
- Computer I/Os
- Control I/Os

Easy processor connection
Flexible configuration