Responsive Link
– Real-Time Communication Standard –

Japan's SC25 WG4 Responsive Link SG

Purposes:
- Real-time networks with point-to-point interfaces

Applications:
- Robots, factory automation, home automation, intelligent rooms/buildings, ITS, etc. …

Standardization status:
- IPSJ (Information Processing Society of Japan) Trial Standard IPSJ-TS 0006:2003

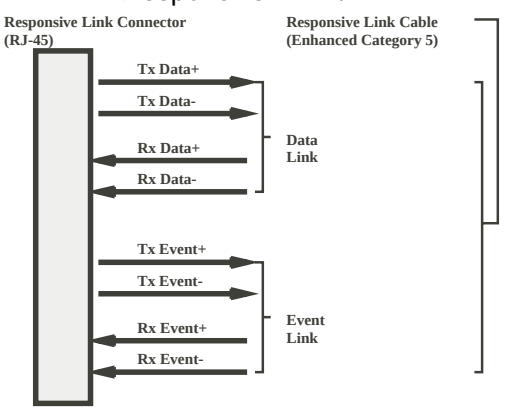
Characteristics of real-time communication:
- Soft real-time: A guarantee for the communication bandwidth
  ⇒ high throughput for bulky data
- Hard real-time: A guarantee for the communication latency
  ⇒ short latency for events

Packet Size	Larger	Smaller
Throughput	Higher	Lower
Latency	Longer	Shorter

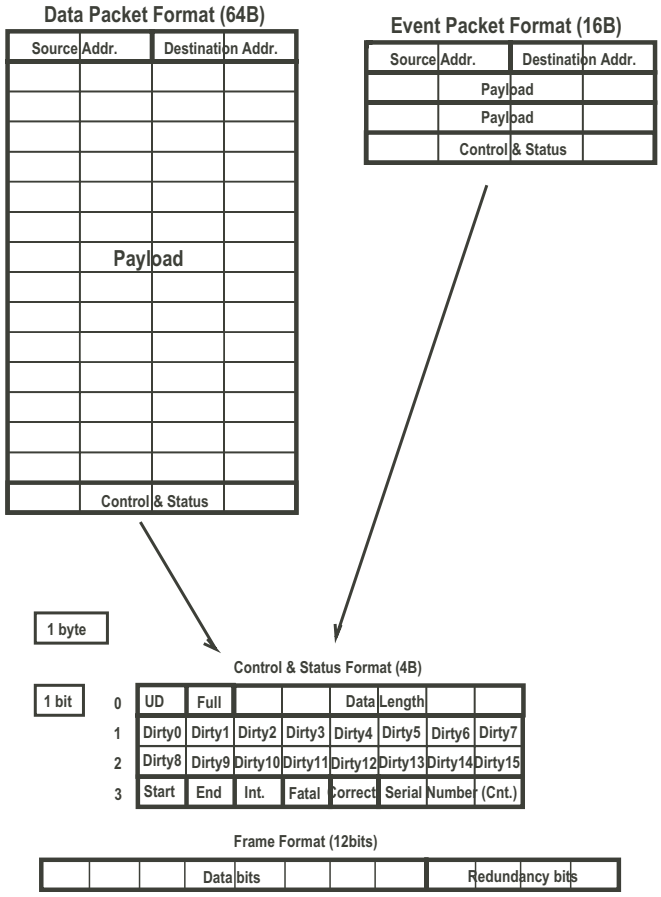
The characteristics of Responsive Link:
- Separation of data transmission for soft real-time and event transmission for hard real-time
- Fixed packet size: 64-byte data and 16-byte event
- Independent routing of the data link and the event link
- Point-to-point serial link
- 256 level priority (8-bit)
- Priority based packet overtaking (The packet with higher priority overtakes other packet at each node.)
- Packet acceleration/deceleration using priority replacement (Packet priority can be replaced with a new priority level at each node to accelerate/decelerate packets under distributed control.)
- Prioritized routing (When multiple packets with different priority levels are sent to the same destination, the different route can be set to realize exclusive communication lines or detours.)
- Plug & Play
- Topology free
- Low latency

Physical Interface:
- Electronic interface: LVDS
- Forward Error Correction(FEC)
- Bit stuffing
- NRZI(Non Return to Zero Inverted)
- DPLL(Digital Phase-Lock-Loop)
- Synchronized frame(Setup Pattern)
- Variable link speed (800, 400, 200, 100, 50, 25, 12.5[Mbaud])

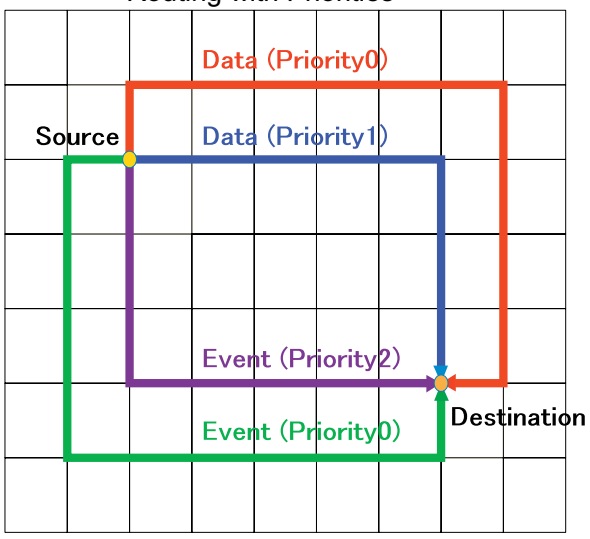
Responsive Link I/F



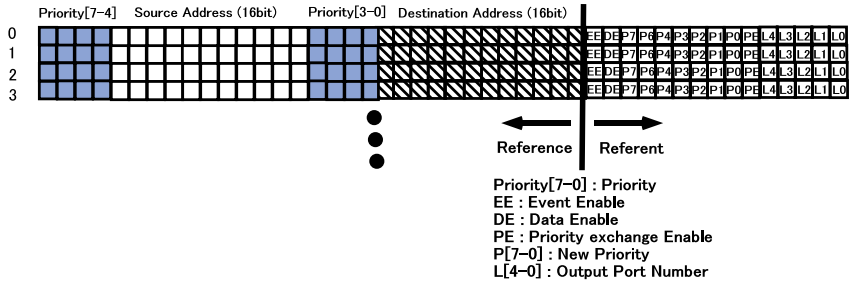
Packet Format



Routing with Priorities



Routing Table (5x5 Crossbar)



Applications using *Responsive Link* for Parallel/Distributed Control

Japan's SC25 WG4 Responsive Link SG

Needs:

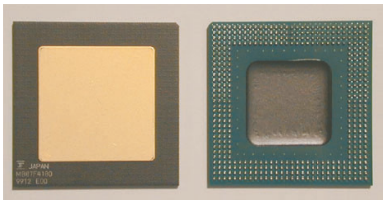
In order to realize distributed control systems, both hard real-time communication and soft real-time communication are required. Hard real-time communication should guarantee the latency of the communication to transmit control commands, synchronization signals, etc. Soft real-time communication is also required to transmit multimedia data including image, voice, etc.

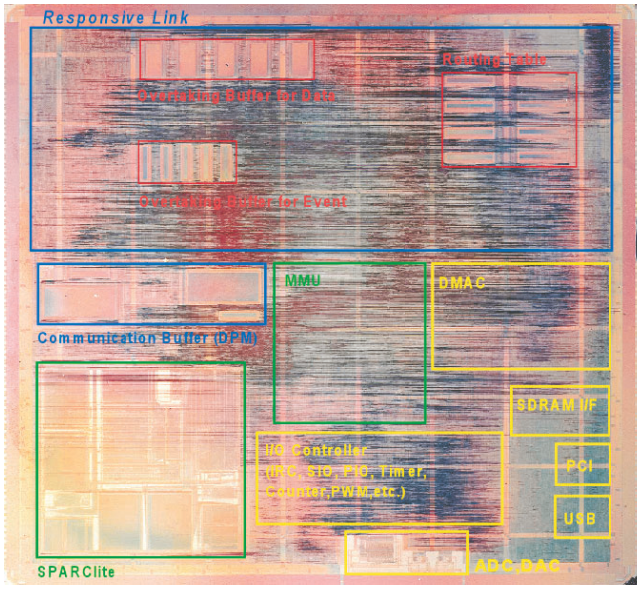
Current technologies:

IEEE1394 and USB can realize soft real-time communication by isochronous transfer. But the error correction is not supported at the isochronous transfer. They can not realize hard real-time communication. Their network topology is fixed. Therefore they can not be applied to distributed real-time control systems.
Ethernet can not support real-time communication, because it uses the CSMA/CD protocol.

Availability:

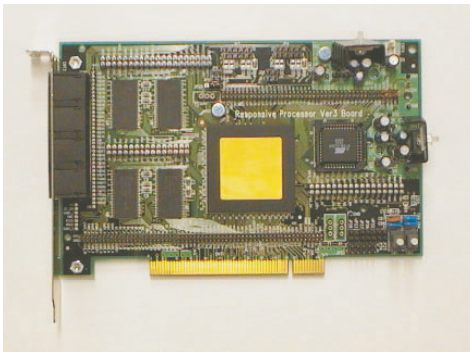
Responsive Link is integrated into several SoCs.
For example, Responsive Processor is the first VLSI chip integrating Responsive Link. Many kinds of control boards using Responsive Processors are developed. Development environments including C, C++, etc. are also available.





Responsive Processor integrates:
1. Processing Core: SPARC
2. Real-Time Communication: 4 pairs of *Responsive Links*
3. Computer I/O Peripherals:
   SDRAM I/Fs, PCI, USB, DMAC, SIO, PIO, etc.
4. Control I/O Peripherals:
   ADCs, DACs, PWM Generators, Pulse Conters, etc.

Responsive Processor including *Responsive Link* IP was designed by AIST, and fabricated by FUJITSU.



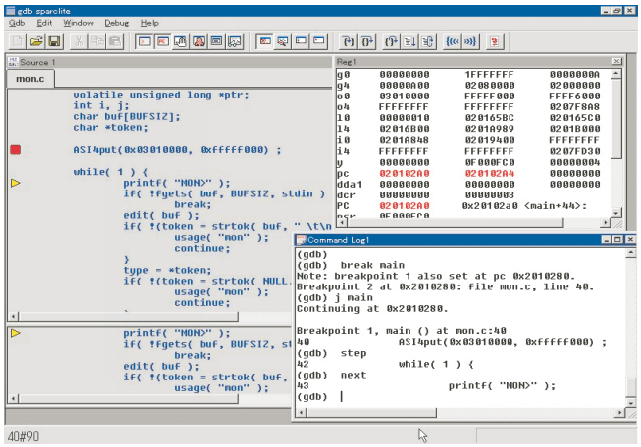
PCI Board for Responsive Processor

Performance of *Responsive Link* on Responsive Processor

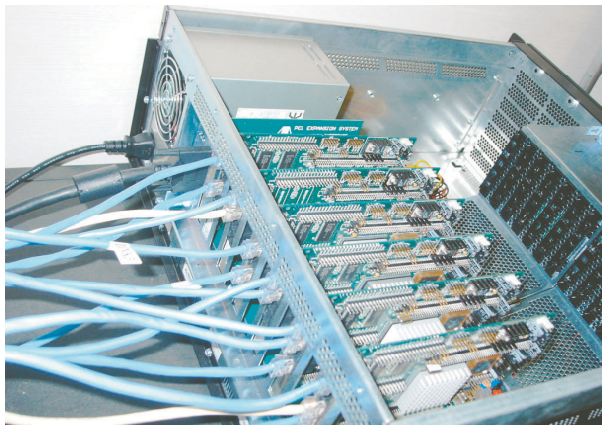
Clock (MHz)	200	100	50	25
Max. Speed (Mbaud)	100	50	25	12.5
Speed of Data (Mbps)	67	33	17	8
Latency of Event(μ sec)	3.1	6.2	12.5	25
Power (W)	0.2	0.1	0.05	0.02

Latency of Event (Worst) @ 100Mbaud =
1.0 (μ sec) + 2.1 (μ sec/hop) x n (hop)
Ex. Latency of 1000BaseT is about 10 μ sec.
Responsive Link guarantees hard/soft real-time communications.

Development Environment:
1. Cross development tools based on GNU tools (GNU C, C++, gdb, make, etc.)
2. Host machine: PC, UNIX Workstations
3. Host OS: Linux, FreeBSD, Windows, Solaris, etc.



WinGDB Debugger based on GNU gdb



Responsive Link Network Switch