Trend of Multi-/Many-core for Embedded Systems

Fumio ARAKAWA, Ph.D.
Chief Professional System Core Development Div.
Renesas Electronics Corp.

September 20th, 2012
Aizu, Fukushima, JAPAN

©2012 Renesas Electronics Corp., All rights reserved.
Outline

- **Markets** of Embedded Systems
- **Transition** to Multi-/Many-core
- **SH-X4**: Highly Efficient Embedded Processor Core
- **RP-X**: A Prototype SoC of Heterogeneous Multicore
- **Products**: Based on Integrated SoC Platform of Renesas
- **Summary**
Markets of Embedded Systems
Further growth of established markets

**Mobile terminals**
- Considerate interface anyone can use
- Simple functionality enabling anything
- Light weight and long battery life

**Home appliances**
- Security, Energy-saving home
- Entertainments all over the house

**Automotive**
- Eco-friendly cars with safety and relief
- Comfortable space like being at home

Expectation to newly growing markets

**Smart grid**
- Various new energy sources and eco-friendly electric power supplies
- Power grid systems enabling rapid response to various changes

**Medical and nursing care**
- Human-friendly medical and nursing care
- Rapid and wise support of medical treatments

**Cloud computing**
- Low-power and eco-friendly information services supported by COOL multicore

Issues of remote computing

- A cloud system enables **thin clients** in many cases relying on high-performance of servers remotely connected by a network.

- However, it is still desirable to accomplish a **real-time** or **dependable** operation **locally** by an embedded system.

- Some critical operation does not allow slow or unpredictable response caused by a **network delay** or **disconnection**.

- Realizing various functionalities **with/without a network**.

- **Network traffic** will become seriously enormous, and local processing can reduce the traffic.

- Disadvantage of remote computing should be **concealed from a user** by local computing.
Computing power can be like air?

- For an ideal **cloud computing**, it must be like **air**.
  - **Air** is everywhere in good condition.
  - Most of us don’t need to care how to get **air**.

- Computing loads should be partitioned and done **at optimal places**.
  - Heavy computing loads will always go to servers?
  - The partitions don’t depend on “computing loads”, but depend on “**time constants**”, “**dependability**”, or “**network loads**”.

- **High-performance embedded processors** are indispensable to solve above issues even in a cloud computing era.
Transition to Multi-/Many-core
Heterogeneous Multi-/Many-core

- **Frequency Scaling**
  - Slowing Down ↔ Power Wall
  - Pollack’s rule: $\text{Perf.} \propto \sqrt{\text{Area}}$
  - Roughly... $\text{Area} \propto \text{Power}$

- **Embedded Processors**
  - Relatively Low Performance
  - Highly Efficient

- **Accelerator IPs**
  - Highly Efficient, Highly Parallel

- **Heterogeneous Multi-/Many-core**

**High Performance under Power Limitation**
Embedded systems also face the power wall.

- High DGIPS²/W is necessary for high performance.

---

*) DGIPS: Dhrystone GIPS
Power Walls

- how to achieve both high performance and efficiency?

![Graph showing efficiency vs performance for different computing technologies.](image)
Seamless Shift to “COOL” Multicore

- Platforms & Runtime/Development Environments
- Education of Multicore Engineer
- Support of Application Programmer
- Effective Use of Huge Legacy Software for Single core
- Accumulation of New Software for Multicore

Education of Multicore Engineer

Development Environments

Runtime Environments

“COOL” Multicore Platform

New Software for Multicore

Huge Legacy Software for Single core

Software Accumulation

Multicore Engineer

Application Programmer
- Ease of ill feeling to multicore
- concealing no. of cores

Software Engineer

Hardware Structure & Issues

- Interconnection Protocols & Topologies (Bus, NoC, …)
- High-speed & Low-power Communication Circuits

Memory Architecture (Hierarchy & Topology) for Large Capacity & High Speed

High-speed & Low-power Cores (CPU, Image Processing, …)

Software Layer & Issues

- Framework best for extracting multicore potential
- Hiding issues specific to multicore from ordinary programmers
- Low-power technology further lowering COOL multicore power

Multicore ready

Multicore Virtualization & Domain Separation Layer
- Driver
- Driver
- Driver
- Driver

Multicore & Domain Interoperation Layer
- Middleware
- Middleware
- Middleware

Realtime OS
- GP OS
- Special OS

Application

Layer API

- Modularity
- Cooperation
- Productivity
- Reusability


• Ensuring realtime operations
• Environment to run legacy software effectively
• Dependability (Monitoring, Recovery, etc.)
System/Software Development Environment & Issues

**System Design**

**Issues for Multicore**
- ESL* Environment

**Upper-level Design**

**Issues for Multicore**
- Design Method
- Process/Thread Assignment
- Realtime Processing
- Performance Tuning
- Model-based Design

**System Verification**

**Issues for Multicore**
- Performance Evaluation Tools

**Functional Verification**

**Issues for Multicore**
- Verification
- Debugger/Optimizer
- Post JTAG
- Verification Coverage/Quality
- Core Assignment Optimization

**Implementation**

**Issues for Multicore**
- Parallel Programming
- Standard Parallelization API
- Automatic Parallelization Compiler
- Legacy/Open Source Treatment

*ESL: Electronic System Level

**Trend**

- Limit of single core becomes apparent to satisfy demands of higher performance, more functionality, and lower power.
- Transition to multicore is promoted by above demands.
- In automotive applications, fault tolerant systems also promote multicore, but its long lead time of designs slows down the transition.
- Demands of more than 16 cores are from high-end routers, base stations and servers.
- No company avoid multicore within this research.
- Price is not a barrier of introducing multicore.

**Issues**

- Users forecast less number of multicore than the ITRS’s for various issues.
- Major issues: 1) insufficient development/runtime environments and API, 2) ineffective utilization of legacy software, 3) lack of multicore engineers.
- Some companies forecast issues are resolved in base station and sever fields until 2015.
Survey by JEITA at some Conferences

Survey Result

- 70% of respondents consider to use Multi-core.
- 61% expect the Multi-core to improve performance and power saving, while 17% expect it to improve cost performance.
- 30% consider the Multi-core to implement and 15% won’t adopt it at all.
- Linux of 45% is the most used OS, followed by Windows of 25% and TRON of 20%.
- Ratio of AMP:SMP is 50:50.
- About API, more than 60% use OS Multi-thread/TaskAPI, 17% use OpenCL, and 11% use OpenMP.
- The highest number of cores in 5 years; 8 is 31%, 4 is 23%, 16 and 32 are 28%, respectively, and the rest is 8%.
- About software issues, 25% is execution/development environment where an application doesn’t need to be conscious of the number of cores, 13% is debugging, and 12% is guarantee of real-time processing and reuse of existing software.

Summary

- Required number of cores in 5 years will be 8, more than that of the iSuppli’s.
- Hardware and software should have scalability for the number of cores.
- Software environments should be considered debugging and parallelization supports.
**Proposal by JEITA**

- **Multi-/many-core chip/tools/OS framework** for embedded systems
  - It is necessary to build a framework of **OSes**, various development support **tools** for utilizing various many-core **chips**.
  - **Standard chip model of heterogeneous multi-/many-core chip**
  - **System Configuration Description**

![Diagram of system configuration and tools](image)

**Tools for manycore**
- Parallelizing tools
- System analysis
- Debug
- ...

**Applications**
- Medical nursing
- Printer
- Automotive
- CPS
- ...

**Manycore OS**
- Extended OS API
- Legacy API
- Application Specific API

**Manycore Chip**
- Conforming the standard model

- **OS supporting heterogeneous manycore transparently with real-time processing capability**
- **Reuse of legacy software**
- **Tools for developing highly optimized software**
- **High reliability for safety**
Why we need a standard chip model?

- Building an Eco system with an open standardization
- Software layers which require the **standard chip model**
  - Implementation architecture of manycore OSes
  - Tools
    - ex. 1) Communication/Synchronization Mechanism
    - ex. 2) Grouping of cores
- Standardization trends of chip related information
  - **IP-XACT of SPIRIT Consortium**
  - **IEEE1275 Open Firmware**
  - **It is not defined for manycore.**
- Promoting the extension of international standards by using the **System Configuration Description** for manycore.
SH-X4

Highly Efficient Embedded Processor Core

*) Some new features of SH-X4 are available only on prototype chips.
SuperH RISC engine (SH) Series Processors

- Continuous Improvement of Efficiency
- Highly Power and Area Efficient

Hierarchical Power Domains

GIPS/W

Partial Clock Activation
Core-standby
Multicore

Power down
U-standby

Clock Stop
Back Bias

R-standby

0.13\(\mu\)m

0.18\(\mu\)m

0.2\(\mu\)m

0.25\(\mu\)m

0.3

0.72

SH-MobileG1/G2/G3
(SH-X2)

SH-Mobile3
(SH-X)

SH-Mobile1

SH4-VL

SH-4

SH-3

SH-2

SH-1

1993

1994

1995

1997

1999

2002

2004

2006

2007

2008

©2012 Renesas Electronics Corp., All rights reserved.
**Special Features of SH Series Processors**

- **High Code Efficiency with 16-bit Fixed-length ISA**
  - SH-1, SH-2, SH-3, SH-4, SH-X, SH-X2, SH-X3
  - Optional 16-bit ISA: ARM Thumb, MIPS16
  - 16/32-bit Mixed-length ISA: ARM Thumb-2

- **In-order Dual Issue Superscalar Architecture**

- **Efficient Branch Acceleration**

- **Efficient FPU**

- **Architecture Extension for SH-X4**
  - ISA Extension with Prefix Code
  - 40-bit Physical Address Space
In-order Dual-Issue Superscalar Arch.

- The architecture has been blushed up continuously since 1993.
  - GMICRO/500, SH-4, SH-X, SH-X2, SH-X3 and SH-X4
- Popular Architecture for Efficiency
  - Intel: Pentium and Atom, ARM: Cortex-A8

Pipeline Structure

<table>
<thead>
<tr>
<th>I1</th>
<th>Out-of-Order Branch Issue</th>
<th>Instruction Fetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I3</td>
<td>Branch Search / Instruction Pre-decode</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Branch</td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>Instruction Decode</td>
<td>Delayed Execution</td>
</tr>
<tr>
<td>E2</td>
<td>Execution</td>
<td></td>
</tr>
<tr>
<td>E3</td>
<td>Data Load</td>
<td>FPU Data Transfer</td>
</tr>
<tr>
<td>E4</td>
<td>WB</td>
<td>FPU Arithmetic Execution</td>
</tr>
<tr>
<td>E5</td>
<td>Store Buffer</td>
<td></td>
</tr>
<tr>
<td>E6</td>
<td>Flexible Forwarding</td>
<td></td>
</tr>
<tr>
<td>E7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pipe Stages:
- BR
- EX
- LS
- FE
Efficient Branch Acceleration

- Branch Prediction, Out-of-order Issue, Fast Prediction Miss Recovery
- No Large & Complicated BHT, BTB, or Algorithms Popular for High-end
- Branch frequency is not so high in an embedded application.

<table>
<thead>
<tr>
<th>Case</th>
<th>Stall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before Accel.</td>
<td>7</td>
</tr>
<tr>
<td>Prediction Hit</td>
<td>0</td>
</tr>
<tr>
<td>Prediction Miss</td>
<td>3</td>
</tr>
</tbody>
</table>
Efficient FPU

- Highly Efficient Parallelization and Accelerations by Special Instructions
- Out-of-Order Completion of Divide and Square-root Operations
- SIMD is also applicable, but not applied to avoid large hardware

<table>
<thead>
<tr>
<th>Special Instructions</th>
<th>Equivalent Sequence</th>
<th>Pitch</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FIPR</strong></td>
<td>Inner product</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>FMUL &amp; 3 FMACs</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td><strong>FTRV</strong></td>
<td>Vector Transformation</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>4 x (FMUL &amp; 3 FMACs)</td>
<td>16</td>
<td>23</td>
</tr>
<tr>
<td><strong>FSRRA</strong></td>
<td>Square-root Reciprocal</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>FSQRT &amp; FDIV</td>
<td>4</td>
<td>34</td>
</tr>
<tr>
<td><strong>FSCA</strong></td>
<td>Sine/Cosine</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Software Library</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

- Drastic Improvements of Pitch and Latency (1/4~1/7), Power Efficiency and Area Efficiency
- SH-X4 achieved 648 MHz, 4.5 GFLOPS, 92.5 mW, 49.1 GFLOPS/W in a 45 nm Low-Power (LP) Process
**SH-X4: ISA Extension by Prefix Codes**

- **Demerit of 16-bit Fixed-length ISA**
  - Short-immediate ISA requires extra inst. of long-immediate load
  - 2-operand Insts. require extra inst. of transfer for a “c=a+b” type
  - Inefficient Register Allocations (Implicit Fixed-Register Operand)

- **Demerit of Variable-length ISA**
  - Complicated, large and slow parallel issue with serial decoding

- **ISA Extension by Prefix Codes**
  - Extra operand and longer immediate with parallel decoding

- **Instruction Decoder**
  - Wider Input from 32 to 64 bits
  - Code selection by checking Prefix
  - Prefix decoder overrides normal decoder

- **Dual Issue of Instructions with Prefixes**

*) Some new features of SH-X4 are available only on prototype chips.
Evaluation of Prefix Code Effects

- Area overhead of prefix codes is less than 2% of SH-X4
- Performance improvements are 10% ~ 34%

Performance improvement Ratio by Prefix Code

- **Dhrystone v2.1**: 2.28 → 2.65 MIPS/MHz, +16%
- **FFT**: +23%
- **FIR**: +34%
- **JPEG Encode**: +10%

*) Some new features of SH-X4 are available only on prototype chips.
SH-X4: Physical Address Space Extension

- 32-bit (4GB) logical, 40-bit (1TB) physical address space
  - Less than 15% area overhead (More than 70% if both are 40-bits)
  - Each task can still run in a 4GB address space.
  - Keeping 32-bit logical address is good for program compatibility.
  - Handling over 2GB main memory is necessary and requires more than 32-bit addressing, and 1TB will be enough for embedded use.

Application Example of Physical Address Space Extension
- Network TV, Image Processing, Motion Detection (Optical Flow)
- Total 4.4GB, but less than 2GB for each of SH-X4, FE, MX-2 or VPU

<table>
<thead>
<tr>
<th></th>
<th>0.4 GB</th>
<th>0.6 GB</th>
<th>1.6 GB</th>
<th>1.8 GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS#0</td>
<td>Video Decode</td>
<td>Detection</td>
<td>Calc. of Feature Quantity</td>
<td>Calc. of Optical Flow</td>
</tr>
<tr>
<td>OS#1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*) Some new features of SH-X4 are available only on prototype chips.
RP-X
A Prototype SoC of Heterogeneous Multicore
## Outline of RP-1, RP-2 and RP-X

<table>
<thead>
<tr>
<th></th>
<th>RP-1 (ISSCC2007 5.3)</th>
<th>RP-2 (ISSCC2008 4.5)</th>
<th>RP-X (ISSCC2010 5.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>90nm, 8-layer, triple-Vth, CMOS</td>
<td>90nm, 8-layer, triple-Vth, CMOS</td>
<td>45nm, 8-layer, triple-Vth, CMOS</td>
</tr>
<tr>
<td>Area</td>
<td>97.6 mm² (9.88 x 9.88 mm)</td>
<td>104.8 mm² (10.61 x 9.88 mm)</td>
<td>153.8 mm² (12.4 x 12.4 mm)</td>
</tr>
<tr>
<td>Voltage</td>
<td>1.0V (internal), 1.8/3.3V (I/O)</td>
<td>1.0-1.4V (internal), 1.8/3.3V (I/O)</td>
<td>1.0-1.2V (internal), 1.2-3.3V (I/O)</td>
</tr>
<tr>
<td>Frequency</td>
<td>600MHz, 4.32 GIPS, 16.8 GFLOPS</td>
<td>600MHz, 8.64 GIPS, 33.6 GFLOPS</td>
<td>648MHz, 13.7GIPS, 115GOPS, 36.3GFLOPS</td>
</tr>
<tr>
<td>Power</td>
<td>11.4 GOPS/W (for 32b)</td>
<td>18.3 GOPS/W (for 32b)</td>
<td>37.3 GOPS/W (for 32b)</td>
</tr>
</tbody>
</table>
Various Processors and Accelerators

- Trade off: Generality & Flexibility vs. Efficiency & Parallelism
- Various Sweet Spots for Various Applications

---

**Trade off: Generality & Flexibility vs. Efficiency & Parallelism**

**Various Sweet Spots for Various Applications**

- Custom Accelerator
- Programmable Accelerator
- Highly Parallel Engine
- Reconfigurable Processor
- Media Processor (VLIW)
- DSP (multi-MAC)
- DSP
- SIMD
- Multiple CPU
- Superscalar CPU
- CPU
- VPU
- MX-Core
- FE-GA
- SH-X4 Cluster

---

Generality, Flexibility
**Block Diagram of RP-X**

- **SH-X4** x8, **FE-GA** x4, **MX-2** x2, **VPU**
- Independent Frequency Control of **SH-X4**
- A **DTU** transfers data on **LMs**, **CSMs** or main memory having global addresses while an CPU processes other tasks.
- Compatibility by 32-bit Logical Address
  Extendibility by 40-bit Physical Address

*) Some peripheral buses and modules are not shown.*
**FE-GA (Flexible Engine/Generic ALU Array)**

- Dynamic Reconfigurable Processor
- Suitable for signal processing or recognition including image and sound data.
- Effective for processes with middle grain parallelism

---

**Diagram:**

- Sequence Manager (SEQM)
- Internal Bus
- System Bus
- Crossbar Network (XB)
- Configuration Manager (CFGM)
- ALU Cells
- MAC Cells
- Load/Store Cells
- Local Memories

---

©2012 Renesas Electronics Corp., All rights reserved.
MX-2 (1024-way-SIMD 4-bit PEs & SRAMs)

- Matrix Structure of Closely Coupled Processing Elements (PEs) and SRAMs
- 1024-way-SIMD 4-bit PEs with ALU, Booth Encoder, etc.
- Efficient Massively-Parallel Arithmetic Processing
- Good for Multiple-of-4-bit Wide Data
VPU (Video Processing Unit)

- Programmable Video Processing by PIPE (Programmable Image Processing Element)
- For Various Formats (MPEG-1/2/4, H.263, H.264, ...)
- For Various Resolutions (QCIF ~ full HD)
- Applicable to New Algorism and Algorism Updates

VPU5

Codec Element 2
Codec Element 1
VLCS Codec
PIPE Transform Prediction
PIPE Motion Comp.
PIPE De-block Filter
DMA
Shift-register-based bus

PIPE micro-program
Load Module
2D ALU
Store Module
Data I/O

VLSI Circuits 2008
ISSCC 2009 paper 8.7
Automatic Parallelized Code Generation

- Cooperation of Global and Local Compilers via Multicore API
- Automatic Parallelized-Object-Code Generation from Source Codes

**Processor Information**

**Global Compiler**

**Program Source**

**SH-4A Thread**

**FE-GA Thread**

**SH-4A Compiler**

**FE-GA Compiler/Library**

**SH-4A Object**

**FE-GA Object**

- **Thread Creation**
- **Macro Task partitioning**
- **Parallelism Analysis**
- **Macro Task Scheduling**

- **Parallelized Source with Multicore API**
  - Frequency control (per core)
  - Data transfer mode (by DTU)
  - Memories (LM/CSM/...)

- **Local Compilers**
- **Object Code Generation**

API: Application Programming Interface
Nearly scalable performance enhancements of SH-X4 up to 4 cores.

“2:1” is the good ratio of the number of cores of SH-X4 and FE-GA.

High improvement is observed when using DTU together.

*) Evaluated by AAC Encoder using Automatic Parallelizing Compiler
### Chip Evaluation Results

- Total Performance per watt reaches 37.3 GOPS/W

<table>
<thead>
<tr>
<th></th>
<th>Operating Frequency</th>
<th>Performance *1</th>
<th>Power *2</th>
<th>Power Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH-X4</td>
<td>648 MHz *3</td>
<td>36.3 GFLOPS</td>
<td>0.74 W</td>
<td>49.1 GFLOPS/W</td>
</tr>
<tr>
<td>MX-2</td>
<td>324 MHz</td>
<td>36.9 GOPS</td>
<td>0.81 W</td>
<td>45.6 GOPS/W</td>
</tr>
<tr>
<td>FE</td>
<td>324 MHz</td>
<td>41.5 GOPS</td>
<td>1.12 W</td>
<td>37.1 GOPS/W</td>
</tr>
<tr>
<td>Others</td>
<td>324/162/81 MHz</td>
<td>-</td>
<td>0.40 W</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>-</td>
<td>114.7 GOPS</td>
<td>3.07 W</td>
<td>37.3 GOPS/W</td>
</tr>
</tbody>
</table>

*1) GOPS is converted to 32-bit value (ex. 8-bit 4GFLOPS = 1GOPS.)
   “1GFLOPS = 1GOPS” for the total performance calculation.

*2) Measured at 1.15 V

*3) The core designed for products achieved 800 MHz later.
Coexistence of SMP Linux and μITRON

| CPU0-3 | - **Linux** is running on the four cores  
| | - Realtime Display of **input image** from a USB Camera  
| | - Display of 3 processed images (**smoothing**, **edge**, **corner**)  
| | - “fps” value increases by increasing number of threads  
| CPU4 | - Display of decoded movie of H.264 **by software** on μITRON  
| CPU5 & VPU | - Display of decoded movie of H.264 **by VPU** on μITRON  
| | - Reduction of CPU load by VPU, a Media Processing IP  

![Image of Coexistence of SMP Linux and μITRON](image_url)
Products
Based on Integrated SoC Platform of Renesas
**Integrated SoC Platform Architecture**

- Divided into System Application and Real-time Processing Domains
- System Application Domain supports open OS.
- Real-time Processing Domain achieves flexibility and performance & power balance through H/W and S/W harmonization.
- Supporting selective usage of functional IP through BUS connectivity IP.
R-Car H1

- Microphone
- Speaker
- Audio Codec
- Cameras
- HDD
- DVD
- GPS R/F
- CAN Transceiver
- MOST PHY
- VICS R/F

**R-Car H1**

- CortexA9 1GHz (CPU, NEON)
- CortexA9 1GHz (CPU, NEON)
- SH-4A (CPU, FPU)
- DDR3 I/F (500MHz, 32bit bus, 2ch)
- Display Unit (2ch)
- PowerVR SGX543MP2
- Renesas Graphics Processor
- Distortion Compensation Module
- Video in
- Serial ATA
- GPS B/B
- CAN (2ch)
- MOST I/F
- DARC
- SPI
- GPIO
- MOS-FET
- MOS-FET
- DDR3-SDRAM
- Front Monitor
- Rear Monitor
- Tuner Module
- USB 2.0 Host (3ch)
- MMC I/F
- SD Card Host I/F (4ch)
- HSCIF (2ch)
- Ethernet I/F
- PCI-Express

**R2A11301FT**

- ADC (8ch) input
- SW
- AVS I/F

©2012 Renesas Electronics Corp., All rights reserved.
R-Mobile A1

R-Mobile A1 System

Application (Linux Solution)

Partner's Application
Customer's Application
OpenMAX
OpenGL ES2.0
OpenVG
Linux Kernel
Device Driver

Media Interface

OpenMAX IL Control
Video Engine
Audio Engine

R-Mobile A1 (SoC)

ARM Cortex-A9 with NEON
3DG
SGX540
SH-4A
VCP1
VPU5F
SPU
JPEG
2DG
Camera I/F

BUS System

Peripherals

Media Engine

Media Interface

32bit

NOR Flash

DDR3

Digital RGB 24bpp

HDMI / NTSC/PAL

W-LAN

eMMC

SD Memory

SDIO

Gigabit Ethernet

Internet

Gigabit Ethernet

PHY

USB

USB

ATAPI

SDIO

HDD

TS I/F

IIS

Sound CODEC

IIC

YCbCr

Camera Module

Camera Module

IIC

YCbCr

DTV Module

SIM Card

SD

Memory

32bit

Memory

ATAPI

HDD

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module

32bit

32bit

Camera Module

Camera Module
Summary

**Market of Embedded Systems**
- Disadvantage of networks should be concealed from a user.
- High-performance embedded processors are indispensable.

**Transition to Multi-/Many-core**
- Polack’s Rule, Power Wall, Inefficient Single Core
- **Heterogeneous Multicore of Highly Efficient Cores**

**SH-X4: Highly Efficient Embedded Processor Core**
- 16-bit fixed-length ISA with **Prefix Extension**
- 32-bit (4GB) Logical and **40-bit (1TB) Physical** Address Space
- **In-order Dual-Issue Superscalar** Architecture
- Efficient Branch Acceleration and FPU Architecture
- 648MHz, 4.5GFLOPS, 92.5mW, **49.1GFLOPS/W → 800MHz** for products

**RP-X: A Prototype SoC of Heterogeneous Multicore**
- SH-X4 x8: 4-core Cluster x2, **FE-GA x4**: for Middle Grain, **MX-2 x2**: for Fine Grain, **VPU**: Programmable

**Products: Based on Integrated SoC Platform of Renesas**