A Very-Low-Latency Superscalar Microcontroller for Automotive, Industrial, and PC-Peripheral Applications

Y. Sugure, S. Takeuchi¹, Y. Abe, H. Yamada, K. Hirayanagi¹, A. Tomita¹, K. Hagiwara¹, T. Kataoka¹, T. Yamazaki¹ and T. Shimura

Hitachi, Ltd., Japan
¹Renesas Technology, Corp., Japan
Outline

• Background
• Architecture of next-generation microcontroller
  - Smaller code size
  - Low-latency instruction processing
  - Low-latency interrupt processing
• Summary
Background

Microcontroller (MCU) concept

MCU controlled by regular programs on-chip memory

- More complex control
- Faster processing
- Improved performance for real-time tasks

- Increased ROM capacity
- Higher cycle performance
- Improved interrupt processing
Requirements for Next-generation MCU

- Increased ROM capacity
  - Smaller code size
    - Reduced program code size

- Higher cycle performance
  - Low-latency instruction processing without increasing pipeline stages
    - Shorter execution cycles

- Improved interrupt processing
  - Low-latency interrupt processing
    - Shorter interrupt response time
SH-2A Core Features

SH-2A operations at 200 MHz

• Reduced program code size
  - New instructions
  - Improved C compiler

• Shorter execution cycles
  - Harvard architecture
  - Superscalar architecture:
    Parallel execution of two instructions
  - New 32-bit instructions

• Improved interrupt response time
  - Register banks
Block Diagram

For automotive, industry (with ROM):

- SH-2A core
  - CPU
  - FPU
  - Register banks
- FLASH
- Memory-bus (for data)
- 32bit 200MHz
- Internal bus
- DMAC
- Bridge
- External-bus controller
- Timer
- A/D
- SCI
- PORT

For PC-peripheral (without ROM):

- SH-2A core
  - CPU
  - FPU
  - Register banks
- Cache
- RAM
- External bus
- DMAC
- Bridge
- External-bus controller
- Timer
- A/D
- SCI
- PORT
Smaller Code Size : New Instruction

- Conventional compatible instruction set

- Add new 16-bit and 32-bit instructions
  - 16-bit-fixed-length instruction
    - Dynamic-shift
    - Division (32bit/32bit)
    - Storage of multiplication result in general register
    - Saved/restored multiple register

  - 32-bit-fixed-length instruction
    - 20-bit immediate or 12-bit displacement
    - Bit-manipulation
Smaller Code Size : Effect

- **SH-2**
- **SH-2A** (optimizing C compiler + enhanced instruction set)

<table>
<thead>
<tr>
<th>Code size [%]</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>For automobiles</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>For industry</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For automobiles and for industry.
Two new architectures for SH-2A

**Harvard Architecture**
- Separated Instruction-Bus and Data-Access-Bus

**Superscalar**
- Issues 2 instructions in parallel
- Double integer pipeline
- Max. 5-stage pipeline
  3-stage: integer, shift, and branch operations
Pipeline Structure

IF  Instruction fetch,
ID  Instruction decode,
EX  Execution,
MA  Memory access,
WB  Write back,
E1  Execution-1,
E2  Execution-2,
F1  Floating execution-1,
F2  Floating execution-2

Dual issue

EX  FPU-load/store*
WB  Floating-point*

EX  Integer (\[2\])

EX  Shift

EX  Branch

EX  Load/store

*1) single precision
New 32-bit Instructions

<table>
<thead>
<tr>
<th>SH-2A</th>
<th>15</th>
<th>8-bit</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOV #imm,Rn</td>
<td>Op</td>
<td>Rn</td>
<td>imm</td>
</tr>
<tr>
<td>MOV.L @(disp,PC),Rn</td>
<td>Op</td>
<td>Rn</td>
<td>disp</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>4-bit</th>
<th>16-bit</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV.I20 #imm20,Rn</td>
<td>Op</td>
<td>Rn</td>
<td>imm</td>
<td>Op</td>
</tr>
<tr>
<td>MOV.L @(disp12,Rm),Rn</td>
<td>Op</td>
<td>Rn</td>
<td>Rm</td>
<td>Op</td>
</tr>
<tr>
<td>MOV.L Rm,@(disp12,Rn)</td>
<td>Op</td>
<td>Rn</td>
<td>Rm</td>
<td>Op</td>
</tr>
</tbody>
</table>

Conventional:
- Max. 8-bit immediate or displacement field
  - Larger data located in program area of Flash memory

SH-2A:
- Max. 20-bit immediate or 12-bit displacement field
  - Prevent wastage of cycles in constant memory access
  - Improve cycle performance
Execution of Both 16/32-bit Instruction

Superscalar structure by 16-bit decoder

- Easy execution of both 16- and 32-bit instructions
- Minimization of decoder logic depth

16-bit instruction:

```
IF -> 16-bit decoder
```

32-bit instruction:

```
IF -> 16-bit decoder -> 32-bit decoder
```

- Minimization of decoder logic depth
Improvement of Cycle Performance

SH-2A: 1.8 times faster speed (avg.)
Interrupt Mechanism

SH-2A: Provides a register bank for each interrupt level
- Automatically saves register contents when interrupted

Conventional:
Save registers to memory by sequential instructions

SH-2A:
Save 19 CPU registers to register bank

Save/restore
R0-R14, GBR, PR
MACH, MACL
SR, PC

Save/restore
R0-R14, GBR, PR
MACH, MACL

Save/restore SR, PC
Implementation of Interrupt Processing

Instruction processing

CPU registers

<table>
<thead>
<tr>
<th>R0</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>MACH</th>
<th>MACL</th>
<th>GBR</th>
<th>PR</th>
<th>SR</th>
<th>PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>R4</td>
<td>R5</td>
<td>R6</td>
<td>R7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R8</td>
<td>R9</td>
<td>R10</td>
<td>R11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R12</td>
<td>R13</td>
<td>R14</td>
<td>R15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

selectors

BUS-A

B

C

BUS-D

E

F

Execution (for decoder 1)

Execution (for decoder 2)

32bit 6
Implementation of Interrupt Processing

Interrupt processing

CPU registers

- R0, R4, R8, R12
- R1, R5, R9, R13
- R2, R6, R10, R14
- R3, R7, R11, R15
- MACH, MACL, GBR, PR
- SR, PC

Register bank

- R0, R1, ..., R14
- GBR, MACH, MACL, PR

MUX

BUS-A
- B, C or E, D, F

32bit × 4

for saving to register bank

for saving to memory
Improvement of Interrupt Response Time

**Conventional:**

<table>
<thead>
<tr>
<th>Interrupt processing time</th>
<th>37 cycles (462.5ns at 80MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register save time</td>
<td>9</td>
</tr>
</tbody>
</table>

**SH-2A:**

<table>
<thead>
<tr>
<th>Interrupt processing time</th>
<th>6 cycles (30nsec at 200MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register save time</td>
<td>1 6</td>
</tr>
</tbody>
</table>

- Save 19 registers in 5 cycles
- Parallel with interrupt processing

Low-latency interrupt
Summary

• SH-2A core in microcontrollers for automotive, industrial, and PC-peripheral applications
  - 200MHz, 360MIPS, 400MFLOPS

• Smaller-code size
  - About 25% smaller by new instructions and enhanced C compiler

• Low-latency instruction execution
  - Dual-issue superscalar with 3- or 5-stage pipeline
  - 1.8-faster cycle performance (avg.)

• Low-latency interrupt execution
  - Saves registers to register bank in parallel with interrupt processing
  - Interrupt response time: 37 cycles ÷ 6 cycles
    (463nsec ÷ 30nsec)