M32R/D
32-bit RISC Microprocessor with On-Chip DRAM

Eric Nguyen
DRAM Integration

- 128b internal bus at 66MHz for CPU-memory data transfer
- 16b external bus at 16MHz for power reduction
- 80 pin plastic QFP package
- Integration DRAM for main memory, page frame and data buffer
Small RISC CPU Core

• 52.4 MIPS @ 66MHz (DRAM ver.),
  80 MIPS @ 100MHz (Logic ver.)
• CPU core size: Smaller than existing
  RISC MPUs
• Code size: Smaller than existing
  RISCs, nearly traditional CISC level
• Full synthesizable HDL description
  is available
DSP Function

- All 32x16b and 16x16b DSP instructions in 1 cycle
- 16b and 32b fixed-point arithmetic supported (rounding & saturation)
- For FIR and IIR filters in MODEM, data compression/decompression, etc.
On-Chip Memory Support

- Two caching modes: On-chip DRAM caching and Off-chip ROM caching
- Wait-cycle control for the on-chip DRAM
- Refresh control of the on-chip DRAM both for normal operation and sleep mode
M32R/D Features

• CPU core:
  – Small 32-bit RISC Architecture
• VAX MIPS:
  – 52.4 MIPS @ 66.6MHz (Dhrystone V2.1)
• Memory:
  – 2 Mbyte DRAM with 2 Kbyte Cache
• Peripheral Logic:
  – 32b x 16b DSP-like Multiply and Accumulator Memory Controller, etc.
M32R/D Features Continued

- **External Bus:**
  - 24-bit address, 16-bit data
- **Clock:**
  - 66.6MHz (internal) / 16.6MHz (external)
- **Supply Voltage:**
  - 3.3V
- **Power:**
  - 275mW (typ.) / 700mW (max.) / ≤ 2mW (stand-by)
- **CPU Size:**
  - 4.0mm² (2nd gen.)
M32R/D Features Continued

- Process Technology:
  - 0.4µm CMOS, 2 metal layers (2nd gen.)
- Package:
  - 80-pin plastic QFP
- Schedule:
  - Engineering Samples: Q2/96
  - Production: Q1/97
- Options:
  - DRAM size (512 Kbyte and 1 Mbyte)
Performance Advantage

- Cache Miss and Line Fill (at 66MHz)
- External DRAM
  - 14 cycles, 4-2-2-2 wait cycles
    [Standard DRAM]
  - 8 cycles, 4-0-0-0 wait cycles
    [Synchronous DRAM]
- Integrated DRAM
  - 5 cycles
Power Dissipation

• External DRAM:

\[
PT_{\text{Total}} = \text{PCPU} + \text{PDRAM} \times 2 + \text{PBUS}
\]
\[
= 360\text{mW} + 280\text{mW} \times 2
\]
\[
+ 80\text{pF} \times 3.3\text{V}^2 \times 66\text{MHz} \times 70 \times 0.1
\]
\[
= 1322\text{mW}
\]

(PDRAM is doubled, because 2-16 x 1 Mbit DRAMs are required for a 32-bit data bus)

• Integrated DRAM:

\[
PT_{\text{Total}} = \text{PCPU} + \text{PDRAM} + \text{PBUS}
\]
\[
= 360\text{mW} + 280\text{mW}
\]
\[
+ 80\text{pF} \times 3.3\text{V}^2 \times 16\text{MHz} \times 30 \times 0.1
\]
\[
= 682\text{mW}
\]
System Cost

- **Number of Chips**
  - External DRAM: CPU with Digital-ASIC, DRAM x 2, Analog-ASIC
  - Integrated DRAM: CPU with DRAM, Digital/Analog-ASIC

- **Package**
  - External DRAM: 160 - 200 pin PQFP
  - Integrated DRAM: 80 pin PQFP

- **MCU Die Size with Integrated DRAM**
  - CPU 2x2 mm²: 16% overall die size increase relative to the DRAM
## M32R/D CPU Core Features

<table>
<thead>
<tr>
<th>Performance</th>
<th>Operating Frequency</th>
<th>Average MIPS</th>
<th>CPI</th>
<th>Multiply and Accumulate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100MHz (Bus Clock 16.6MHz)</td>
<td>80 MIPS</td>
<td>1.25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 cycle</td>
<td>1 cycle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instructions</td>
<td>Instructions</td>
<td>83</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Registers</td>
<td>16 x 32 bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus Interface</td>
<td>Address Space</td>
<td>4 Gbytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Bus</td>
<td>32 bit</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bus Cycle</td>
<td>1 cycle (min)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core</td>
<td>Supply Voltage</td>
<td>3.3V</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Process Technology</td>
<td>0.5μm, 3 metal layers</td>
<td>For synthesis and P&amp;R</td>
<td></td>
</tr>
</tbody>
</table>
M32R/D Development Tools

M32R/D Tool Chain

C Program Source
Assembler Source
Compiler source
User library
Target module
Target library
Library module
ANSI-C library
UNIX based host WS
Debugger
On board monitor
Target board
Evaluation module
Emulator
Target board
Emulator 1
Simulation
Debugger
Emulator 2
Simulator
Debugger
M32R/D Software Tools

- ANSI-C Cross Compiler
  - Mitsubishi (Q2/96)
  - GNU (Q4/96)
- Real Time OS
  - Mitsubishi ITRON (Q2/96)
- Evaluation Board
  - Mitsubishi (Q3/96)
- ICE
  - Mitsubishi (Q4/96)
  - 3rd party (Q1/97)
- Verilog-HDL Description
  - Mitsubishi (Q2/96)
System Configuration Concept

= M32R/D + I/O ASIC + ROM
Application Examples

• Personal Digital Assistant (PDA)
• Intelligent Data Pager
• Printer System Configuration based on M32R/D
• Telephony Internet Terminal
• CATV (Digital) Internet Terminal
• CATV (Analog) Internet Terminal
• Digital Camera
• Future Camera Application
• Set-top Box
• Multimedia Processing System
M32R/D Advantages

- DRAM Integration
- Small RISC CPU Core
- DSP Function
- On-Chip Memory Support
Java™ on M32R/D

Yuichi Nakao
Java™ for Embedded System: Java Advantages

- Portable, Multi-platform
  - Interpreted, Architecture Independent Byte Code

- Robust
  - No Direct Memory Access, Garbage Collection

- High Security
  - Limitation on Applet, Byte Code Verifier
For two years Mitsubishi has been implementing Java onto our MPU
We have ported
- Java™ Runtime, HotJava™, Core Library, Network Class Library, Display Class Library
Demonstration at the Mitsubishi booth
Our goal is to provide a portable and embedded Java system using the M32R/D RISC processor
M32R/D: Ideal Platform for Compact Data Processing

- High Performance Compact for RISC Core
  - 52.4 MIPS @ 66.7MHz
  - 2mm x 2mm
- On-Chip DRAM
  - High performance with wide bandwidth
    - 128-bit internal bus
  - Low power consumption
    - 275mW with 2 Mbyte DRAM on-chip
  - Small footprint for MPU-memory system
M32R/D Target Application

- Portable or handheld system with low power consumption
  - PDA, PIC
  - Data Pager, Wireless Smart Phone
- Communication or graphic controller with high performance and high bandwidth
  - Network Protocol Controller
  - Digital Still Camera
  - Navigation System with Graphic Display
  - TV/VCR Graphic User Interface
  - Video Games
Java on M32R/D Application: Concept

- Single chip Java engine for portable and embedded system
- Basic S/W in on-chip DRAM
  - Java Runtime + Real time kernel + library
Java on M32R/D Application: Target Applications

- Portable Internet Browser
  - PIC
  - PDA
  - Wireless Phone
- Java Data Pager
- Video Games, Navigation System, VCR/TV with HotJava
Application Block Diagram: Java PIC

- Java PIC
- Graphic
- Micro Kernel
- Java Runtime
- Graphic
- Micro Kernel
- Baseband LSI
- Synthesizer Modulator
- RF
- LPF BPF
- M32R/D
- Applet Viewer
- Flash Memory
- Li Battery
- LI Battery
- LCD
- Speaker
- Microphone
- IrDA
- Speaker
- RF
- Synthesizer Modulator
- Baseband LSI
- Flash Memory
- M32R/D
- Applet Viewer
- Flash Memory
- M32R/D
- Applet Viewer
Application Block Diagram: Interactive TV with HotJava
Demonstration

• Executing PDA Applet Demonstration
• Browsing Web through Ethernet
• Downloading and Executing Simple Applet from PC
Demonstration: Hardware Configuration

M32R/D Demo System

Touch Panel → Color LCD → Display Driver Board

M32R/D Board

DRAM

CPU

DRAM

Super I/O FPGA

Interrupt Controller

Timer

PWM

LCD Controller

A/D Converter

DMAC

PCMCIA I/F

UART

RTC

Two PCMCIA Card Slots
1 for Program
1 for LAN Connection
Demonstration: Software Configuration

- PC
- M32R/D Demo System
- Applet
- HotJava Classes
- Java Foundation Classes
- AWT Classes
- Net Classes
- Java Runtime Virtual Machine
- Libawt
- Libnet
- Micro Kernel
- M32R/D
- Program in PCMCIA
- Super I/O Chip
- M32R/D Program in PCMCIA Super I/O Chip
Software

• Java and M32R/D: an ideal platform for portable and embedded applications
• Potential Applications
  – Portable Internet Browser
  – Java Data Pager
  – Video Games, Navigation System, VCR/TV with Internet Access
• Demonstration at Mitsubishi Booth
M32R/D: Java™ in Silicon

Mamoru Sakamoto
Requirements for Embedded Java

• Vendor Requirements
  – Maintainability
  – Reusability
  – Internet Product

• Customer Requirements
  – Easy Operation
  – Serviceability
  – Customization
  – Low Cost

Java

No OS
Small Memory

Embedded Java
Embedded Java System Structure

Create Applet ➔
Application ➔
Java Foundation Classes ➔
Java Applet

Create Application ➔
Applet Viewer ➔
HotJava

Integrate Native Method ➔
AWT Classes ➔
Net Classes

Driver I/F ➔
Java Runtime Virtual Machine ➔
POSIX Library (sub set)

Libawt ➔
Libnet

Program Memory
WWW Server
On-Chip DRAM

Device Driver

LCD Panel
Touch Screen
Sound
Network

Micro Kernel
Java Implementation

- Java Application Development
  - Applet: Download from WWW Server
  - Application: Embedded in Java

<table>
<thead>
<tr>
<th>Access Capability</th>
<th>File</th>
<th>Memory</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No</td>
<td>No</td>
<td>Restricted to Host</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>No Restriction</td>
</tr>
</tbody>
</table>

- Customize Java Class File to Various Applications
  - Native Method

- Apply Java to Embedded Products
  - Run Java on Micro Kernel
Programming and Running Java Applets
Programming and Running Java Application

Java Language

Authoring Tool → Image Data

Sound Editor → Sound Data

Java Compiler → Byte Code

Java M32R/D

Memory File System

Java Including Application

Insert Card → Power On

Java Booted from SRAM Card

Application Start

Download and Store SRAM Card

M32R/D Evaluation Board
Programming Native Method

- Adapt Java Class to Application Requirement
  - PDA, STB, Navigation, TV, PCS

1. Native Class
   - Javac
2. Class Header
   - Javah
3. C Stub Functions
   - CC32R
4. Stub Table
   - Stublictable.o
5. Method.lib
6. Stubtable.o
7. Java.o
8. Linker
9. PCMCIA Card
10. ROM

Java Core Library
Network Library
Windows Library

Application
Specific Java
Incorporating Device Driver in Java

- Multimedia Application Products
  - To Develop Various Drivers for STB, PDA, Navigation, TV
    - Common Platform
  - To Respond Quickly to Demand from Devices
    - Asynchronous I/O Multi Task Feature
  - To Achieve Low Cost
    - Small Memory Size
Notify Event from Driver to Java

- Event Flag: To Designate Device Type
- Forced Handler: To Notify Event to Java

Applet
Send Event to Component
Forced Handler

Java Runtime Task

Interrupt Handler

Storage | Network | Pointing Devices

set_flg()
ras_fex()

xxx_open()
xxx_close()
xxx_read()
xxx_write()

Device Driver
Future Plan

- Reduce Memory Size for Consumer Electronics
  - Resident Class File in Card, ROM
- Improve Performance
  - Performance Evaluation with Conventional Benchmark
  - Porting Just-in-Time Compiler
  - Realize Multi Thread with Micro Kernel Task Facility