Microprocessor Research at Peking University

Microprocessor R&D Center, Peking University
Content

- JBCORE32 Microprocessor
- Retargetable Compiler JBRC-2
- Assembler generator JBASM-2
- Microprocessor Simulators on structure, organization and signal level
- Program development and debug environment
- Operating system prototype
- Information electronic prototyping system based on the JBCore32 microprocessor
Main features of JBCore32

- Supports 32-bit and 16-bit instruction sets, and can switch between them dynamically; offers both high code density and high performance for the demand of embedded system; easy for low-power implementation.

- Owns complete instruction set, and have rich addressing modes, flexible in organizing instruction fields; can support DSP applications.

- Harvard architecture, 5-stage pipeline; hardware interlock; supports precise interruption.

- Has 7 kinds of execution modes—user, system, supervisor, IRQ, fast IRQ, memory management and user extended instruction—and adopts the register window technology to support fast switching between the modes, facilitating real-time processing and Operating System implementation.

- Provides high-speed I/O control, memory management control and testing control mechanism.

- Designed with the standard VHDL, having high IP (Intellectual Property) value.
### JBCORE32 Instruction Set (32bits)

| Bit 31  | Bit 30  | Bit 29  | Bit 28  | Bit 27  | Bit 26  | Bit 25  | Bit 24  | Bit 23  | Bit 22  | Bit 21  | Bit 20  | Bit 19  | Bit 18  | Bit 17  | Bit 16  | Bit 15  | Bit 14  | Bit 12  | Bit 11  | Bit 10  | Bit 9   | Bit 8   | Bit 7   | Bit 6   | Bit 5   | Bit 4   | Bit 3   | Bit 2   | Bit 1   | Bit 0   |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| D_Irm_Shift | 0  | 0  | 0  | Op | codes | S  | Rn  | Rd  | Shift | Immediate | 0  | Shift | 0  | Rm  |
| D_Reg_shift  | 0  | 0  | 0  | Op | codes | S  | Rn  | Rd  | Rs   | 0  | Shift | 1  | Rm  |
| Multiply     | 0  | 0  | 0  | 0  | 0  | A  | S  | Rn  | Rd   | Rs   | 1  | 0  | 0  | 1  | Rm  |
| Multiply long | 0  | 0  | 0  | 0  | 1  | U  | 0  | S  | Rn   | RdLo | RdHi | 1  | 0  | 0  | 1  | Rm  |
| MvFromStatus | 0  | 0  | 0  | 1  | 0  | R  | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | Rm  |
| MvToStatus   | 0  | 0  | 0  | 1  | 0  | R  | 1  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | Rm  |
| MvToDisp     | 0  | 0  | 0  | 1  | 0  | R  | 1  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | Rm  |
| BranchEx    | 0  | 0  | 0  | 1  | 0  | 0  | 1  | L  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | Rm  |
| MvFromSpec   | 0  | 0  | 0  | 1  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | Rm  |
| CLZ         | 0  | 0  | 0  | 1  | 0  | Z  | 1  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 1  | 0  | 0  | 1  | Rm  |
| D_Irm_Imple | 0  | 0  | 0  | 1  | 0  | Op | codes | S  | Rn  | Rd  | Rotate | Immediate |
| MvToSpec     | 0  | 0  | 0  | 1  | 0  | R  | 1  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | Rm  |
| LS_R_offset | 0  | 1  | 0  | P  | U  | B  | W  | L  | Rd  | Shift | Immediate | 0  | Shift | 0  | Rm  |
| LS_SkipStep  | 0  | 1  | 0  | P  | U  | 0  | W  | L  | Rd  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | Rm  |
| LS_HEADER    | 0  | 1  | 0  | P  | U  | 1  | W  | L  | Rd  | High Offset | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | Rm  |
| Swap/SwapByte | 0  | 1  | 0  | 0  | 0  | B  | 0  | 0  | Rd  | Rotate | Immediate |
| LS_I_effect  | 0  | 1  | 1  | P  | U  | B  | W  | L  | Rd  | 0  | 0  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | Rm  |
| LS_Multiply  | 0  | 1  | 1  | P  | U  | S  | W  | L  | Rd  | Sbit_high_RegList | 0  | 1  | H  | 11bit_low_RegList |
| Extended Instruction | 0  | 1  | 1  | X  | 1  | 23bit_offset_high | 23bit_offset_low |
| Mov LongForm | 1  | 0  | 0  | 0  | 23bit_offset | high | 23bit_offset | low |
| BranchLink  | 1  | 0  | 1  | Cond | L  | 24bit_offset |
| Co_Load/Store | 1  | 1  | 0  | P  | U  | N  | W  | L  | Rd  | CRd | Cp_num | 9  | bit | offset |
| Co_DataProc  | 1  | 1  | 1  | 0  | Op1 | CRn | CRd | Cp_num | Op2 | 0  | CRn |
| Co_Reg Trans  | 1  | 1  | 1  | 0  | Op1 | L  | Rd  | CRd | Cp_num | Op2 | 1  | CRn |
| SWI          | 1  | 1  | 1  | 1  | 28bit_sw1_number |

Microprocessor R&D Center, PKI
| Instruction                      | 15 | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
|---------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Move shifted register           | 0  | 0  | 0  | Op | Rd |    |    |    |    |    |    |    |    |    |    |    |
| Add/subtract                    | 0  | 0  | 0  | 1  | 1  |    | Rs |    |    |    |    |    |    | Rd | I  | Op |
| Move/Compare/add/subtract Immediate | 0  | 0  | 1  | Op | Rd |    |    |    |    |    |    |    |    |    |    |    |
| ALU operation                   | 0  | 1  | 0  | 0  | 0  | Rs |    |    |    |    |    | Op |    |    |    |    |
| Hi register operation/branch exchange | 0  | 1  | 0  | 0  | 0  | Rs/Hs | 1 | H1 | H2 | Op |    | Rd/Hd |    |    |    |    |
| PSR operation                   | 0  | 1  | 0  | 0  | 0  |    | L  | S  | D  | 1  | 1  | 1  | 1  | 1  |    | Rds/Hds |
| PC-relative load                 | 0  | 1  | 0  | 0  | 1  |    | Rd |    |    |    |    |    |    |    |    | Word8 |
| Load/store with register offset  | 0  | 1  | 0  | 1  | 0  |    | Rb |    |    |    |    |    | Rd | L  | B  | Ro |
| Load/store sign-extended byte/halfword | 0  | 1  | 0  | 1  | 1  |    | Rb |    |    |    |    |    | Rd | H  | S  | Ro |
| Load/store with immediate offset | 0  | 1  | 1  | B  | L  |    | Rb |    |    |    |    |    | Rd |    |    | Word5 |
| Load/Store half/word            | 1  | 0  | 0  | 0  | L  |    | Rb |    |    |    |    |    | Rd |    |    | Word5 |
| SP-relative load/store          | 1  | 0  | 0  | 1  | L  |    | Rb |    |    |    |    |    |    |    |    | Word8 |
| Load address                    | 1  | 0  | 1  | 0  | Sp |    | Rd |    |    |    |    |    |    |    |    | Word8 |
| Add offset to stack pointer     | 1  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | S  |    |    |    |    |    | Word7 |
| Push/Pop register               | 1  | 0  | 1  | 1  | 1  | 1  | L  | R  |    |    |    |    |    |    |    |    |
| Multiple load/store             | 1  | 1  | 0  | 0  | L  |    |    |    |    |    |    |    |    |    |    |    |
| Conditional branch              | 1  | 1  | 0  | 1  |    |    |    |    |    |    |    |    |    |    |    | Sword8 |
| Software interrupt              | 1  | 1  | 0  | 1  | 1  | 1  |    |    |    |    |    |    |    |    |    | Word8 |
| Unconditional branch            | 1  | 1  | 1  | 0  | 0  |    |    |    |    |    |    |    |    |    |    | Soffset11 |
| Long branch with link           | 1  | 1  | 1  | 1  | H  |    |    |    |    |    |    |    |    |    |    | Offset11 |
JBCORE32 Microprocessor Controller

Instruction Pipeline

Instruction

IF/ID

16To32

RegFile Address

Port Address

State Machine

Non Taken

Exception Handler

Signal & Imm Generator

Branch Handler

Write Control

EX/MEM

MEM/WB

I-BusCtrl D-BusCtrl

Instruction

FIQ IRQ I-Abort D-Abort

Branch Handler

EX Sig

Interlock

Bypassing

Hazard Handler

MEM Sig WB Sig

EX/MEM
The Layout of JBCore16-II

- Technology: Alcatel 0.35 μm
- Area: 5.0 mm²
- Core Power consumption: 12 mW
- Place and route: 5 levels
- Frequency: 30 MHZ
- Packaging: 68-PIN CQFP
- The date of wafer fabrication: 11/2000
The Layout of JBCore32

- Current technology: Alcatel 0.35 µm
- Area: 20 mm2
- Power consumption: < 300 mW
- Place and route: 5 levels
- Packaging: 208-PIN PGA
- Wafer fabrication technology: UMC 0.25 µm
- Expected date of wafer fabrication: 7/2001
- Expected Frequency: 100-200 MHz

Microprocessor R&D Center, PKL
JBCODES provides a complete set of application program development and debugging platform supporting Instruction architecture performance evaluating, cache performance evaluating. It can support the research and development of Microprocessor efficiently.

New achievement of year 2000

- Retargetable compiler JBRC-2  Support 32/16bits instruction system code generating  and can embed assembly language in C, and can execute SPEC95

- Assembler generator JBASM-2  Hard-coding technology  support complex addressing modes, variable instruction instruction description, and support the separating of assembling and linking procedure

- 32bits operating system prototype  Virtual Memory Management  Multiple tasks scheduling

- Simulators on structure, organization and signal level  High-speed, accurate, nice series of simulator

- Program development and debugging environment  support project management and operating of multiple files, complete debugging functions  and friendly interface
Hardware/Software Co-design environment for Microprocessor

Program Development and debugging environment

- Compiler
  - *c
- Assembler
  - *asm
- Linker
  - *bin

Operating System Prototype

- Signal level simulator
- Structure level simulator
- Organization level simulator

Program Development and debugging environment

- Instruction set architecture (ISA)
- Evaluating Platform
- Cache Performance Evaluating Tools
- Microprocessor Design

Tradeoff of microprocessor design

- CPI
- Instruction number
- Cycle time

Microprocessor R&D Center, PKU
Microprocessor R&D Center, PK

Jade Bird Retargetable Compiler • JBRC-1

MIPS Description file
X86 Description file
BCORE Description file

Compiler Research and Development Infrastructure

Code Generator
Generator

Code Selection

Compiler Back end

Low level Optimizing
Register allocation

Assembly Language

Microprocessor R&D Center, PKU
Assembler Generator • JBASM-2

Preprocessing Module
- Preprocessor
  - Expression evaluating instruction mode processing
- First Pass assembling
  - Process directives, Creating symbol form
- Second Pass assembling
  - Code translating, pseudo instruction expansion
- Target file generating
  - Select to generate related files for simulating and debug

Instruction system Related modules
- Description file analyzing
- Lexical analyzing syntax rules inspection
- Intermediate data structure representation And reduction
- Code emitting

Basic assembly modules
- Target code, Symbol table combination
- Linker
- Target file generating
  - Select to generate related files for simulating and debug

I/O Module
- Linker
- Symbol table
- Symbol form

Other modules
- Microprocessor R&D Center, PKL
Architecture level simulator ArchSim

Simulating environment

1. Format preprocessor
2. Reset virtual machine status
3. Pre-analyzing instruction and machine status
4. Interpreting instruction
5. 16 bit instruction converting
6. Executing instruction

Simulating instruction fetch

- Exception detecting
- System calling
- Error
- Processing error

I/O operation

- Simulating operation
- Simulating I/O operation
- Information feedback
- Result comparison and analyzing

Processing error

Simulating result

Pre-analyzing instruction and machine status
Organization level simulation OrgSim

Loader

Assembler

*.exe *.jbc

Simulating result display

Instruction pipeline

Control signal

Simulating execution

Simulating result display

Register

Memory

Output Display

Component port Display

Control signal table

Instruction pipeline

Control signal
Signal level Simulator SigSim

Preprocessor

Instruction stream

SIMULATING datapass

Data flow

Control signal generating module

C source file

*.*.cc

PureCC compiler

Assembler

Assembly source program

*.asm

Object file

*.obj

Linker

Interface

Display driver module

Simulating Controller

Instruction analysis

Data dependence analysis

Control Dependence analysis

Signal flow
Program Development integrated environment and tools

System Demand → Project Files

→ Project Management

→ File editing

→ Assembler

→ Standard head file

→ Compiler

→ Standard Lib

→ Assemble

→ *.as

→ Linker

→ *.obj

→ Register Display

→ File editing

→ File editing

→ Memory Display

→ File editing

→ I/O Display

→ Microprocessor R&D Center, PKU
Prototyping system based on the JBCore32 microprocessor

The hard disk of PC is used for the system virtual hard disk.

Support parallel Multimedia

MPEGII Chip
Decode the audio and video stream, and send to the TV

The hard disk of PC is used for the system virtual hard disk.