

---

## THE ARCHITECTURE OF A VIETNAMESE 32-BIT RISC MICROPROCESSOR, THE VN1632

**Ngo Duc Hoang, Hau Nguyen Thanh Hoang, Nguyen Phu Quoc, Do Ngoc Quynh**

IC Design Research and Education Center (ICDREC)

*(Manuscript Received on April 08<sup>th</sup>, 2010, Manuscript Revised November 25<sup>th</sup>, 2010)*

**ABSTRACT:** VN1632 is the first 32-bit Vietnamese-designed microprocessor. Its design is based on the Harvard 32-bit RISC architecture but with a five-stage pipeline. This article presents the architecture overview and the implementation of the microprocessor. The overview shows main features, the block diagram and descriptions of most salient blocks, namely registers and pipeline. The implementation describes the design detail of each block. A detailed simulation was carried out to check the overall performance of the design which was then entrusted to an American fab for fabrication using the 0.13um IBM process. Testing results of VN1632 proved that the architecture works correctly with desired performance.

**Keywords:** microprocessor, RISC, computer architecture, pipeline

### 1. INTRODUCTION

A microprocessor is a computer itself. It is, so to say, a conglomeration of all necessary functional parts for processing information data. RISC, or Reduced Instruction Set Computer, is an architecture that uses a small, highly-optimized set of instructions.

The 32-bit microprocessor VN1632 was designed and developed based on the experiences accumulated by the success of other 8-bit microcontrollers [1][2][3]. The challenge of this new task was not only the complexity, the larger scale of the 32-bit microprocessor, but to ensure the design originality, many new and hard issues have been studied and implemented: cache memory, prefetch buffer, write buffer, store buffer, bus interface, co-processor, etc...

In the present paper, we introduce the characteristics of the microprocessor. The main characteristics are the architecture of Harvard 32-bit RISC but with a five-stage pipeline and on-chip cache memory, in which instruction cache and data cache are separate. The present paper also describes the architectural implementation of the microprocessor. This implementation describes general and detailed specification. The general specification shows the modules from the top view and the connection among the modules. The detailed specification shows the detailed implementation inside each module.

The main architectural difference between the VN1632 and others is the architecture of five-stage pipeline, in which five successive instructions are loaded simultaneously in five different pipeline stages. As a result, five

instructions are executed at the same time. This effectively improves the performance of the microprocessor.

The design has been synthesized, simulated and fabricated using 0.13um IBM process. The result shows that this architecture works correctly with desired performance.

## 2. ARCHITECTURE OVERVIEW

### 2.1 Features

The VN1632 has the following main features

- Harvard RISC architecture
- Five-stage pipeline architecture
- Separate instruction cache and data cache
- Built-in cache memory
- 65 instructions
- 32-bit instruction width
- Multiply in only 2 clock cycles
- Debug support with breakpoint
- Synchronous design

### 2.2 Block diagram

Block diagram of a design give us the top view of the design. The VN1632 comprises the following blocks

- CPU registers: Registers used inside the CPU
- CP0 registers: Registers configure system operations inside and outside the CPU
- ALU/Shifter: Computational unit
- MAC: Computational unit for multiply/ add
- Instruction Cache: A cache memory for instruction fetch
- Data Cache: A cache memory for load/store data
- Bus interface unit: Controlling bus interface between the CPU and external circuit

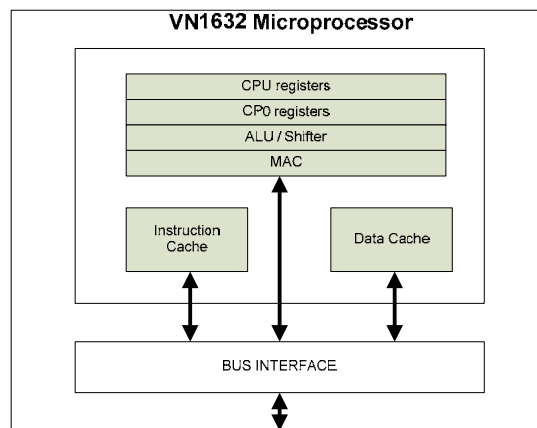


Figure 1. Block diagram

### 2.3 Registers description

There are two kinds of 32-bit register in the microprocessor, CP0 registers and CPU registers. CP0 registers contain informations to config system operations inside and outside the CPU. Meanwhile, CPU registers are used for

operation of the CPU. They are shown as follows

- 32 general purpose registers
- A program counter (PC)
- A Branch Target Address (BTA)
- HI/LO registers for storing the result of multiply operation

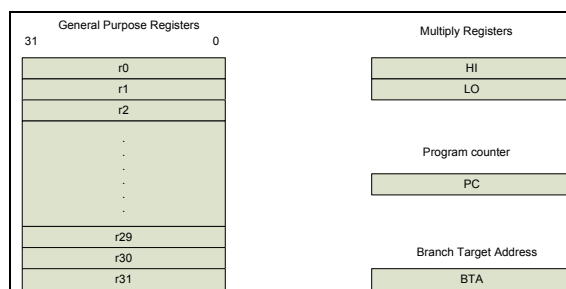


Figure 2. CPU registers

### 2.4 Pipeline description

The VN1632 uses a architecture of five-stage pipeline. Each stage performs its own task which interacts other stages. When the pipeline is fully utilized, five successive instructions are simultaneously in five different pipeline stages. Five instructions are executed at the same time resulting in execution rate of one instruction per cycle. This effectively improves the performance of the microprocessor.

The five pipeline stages are Instruction Fetch (F), Instruction Decode (D), Execute (E), Memory Access (M), Write-Back (W). Each stage is executed in one clock cycle. They are divided into 5 individual modules which are described later in this paper.

The 5-stage pipeline architecture are shown in the Figure 3

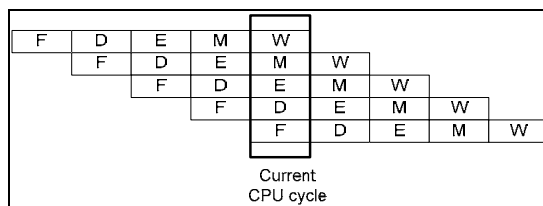


Figure 3. Five-stage pipeline architecture

### 3. IMPLEMENTATION

#### 3.1 General Specification

General Specification shows the framework of a design. It is the first task to

implement the microprocessor VN1632. The general implementation block diagram in the Figure 6 shows main blocks and main connections among the blocks of the microprocessor.

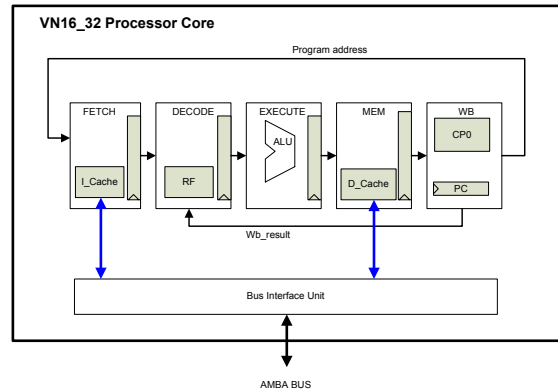


Figure 4. General implementation block diagram

The microprocessor is divided into 6 modules: FETCH (F), DECODE (D), EXECUTE (EX), MEMORY (MEM), WRITE BACK (WB), BUS INTERFACE UNIT (BIU). The first 5 modules above correspond to the 5 stages of the pipeline. Respectively, they are: Instruction Fetch (F), Instruction Decode (D), Execute (E), Memory Access (M), and Write-Back (W).

- **FETCH:** This module gets instructions from slow external memories and store in fast internal memory (I\_Cache). Then the instruction can be fetched quickly from I\_Cache instead of slow external memory.
- **DECODE:** This module decodes instructions that are fetched from IC,

then generating signals to control the following stages. Besides, it holds the 32 general purpose registers of the CPU.

- **EXECUTE:** The main part of this module is an Arithmetic Logical Unit (ALU). The mission of the ALU is to calculate from operands provided by DECODE and to feed results to the next stage.
- **MEMORY:** This module gets data from slow external memories and stores in fast internal memory (D\_Cache). Then the data can be read/written quickly from/to the internal memory.
- **WRITE BACK:** The purposes of this modules are to generate final results,

to control branching (performed via PC), and to do co-processing operations (performed by CP0).

- BUS INTERFACE UNIT: The purpose of this module is to transmit

data from the CPU to external bus system and to receive the data from bus system to the CPU.

### 3.2 Module FETCH

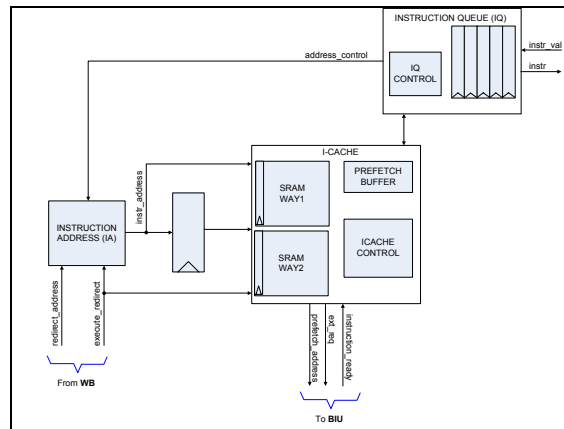


Figure 5. Block diagram of module FETCH

Figure 5 shows the block diagram of module FETCH. The module consists of 5 main blocks: INSTRUCTION ADDRESS (IA), SRAMs (SRAM stands for Synchronous Random Access Memory), PREFETCH BUFFER, ICACHE CONTROL, INSTRUCTION QUEUE (IQ)

- IA: This block generates 32-bit address pointing to the next instructions. The output address is controlled by signals from IQ and WB. Signals from IQ control the increase of the output address, and signals from WB provide an immediate address to IA.
- SRAMs: These are internal memory that is much faster than external memory. They are also called cache

memory. They are used to temporarily store the instructions that are fetched from external memory. The instructions are read from SRAMs, instead of external memory.

- PB: This block fetches instructions from external memory and write to SRAMs. It sends handshaking signals to BIU and then get data there.
- ICACHE CONTROL: This block is a state machine (SM) that controls all the operations of module FETCH. It gets signals from IQ and PREFETCH BUFFER, then send back control signals to them. It also determines the time to write data to SRAMs.
- IQ: Instructions are queued in IQ, go in turn to the following stage. The

mechanism of operation is First In First Out (FIFO). When IQ is “empty” (“empty” means less than one instruction in IQ), it sends request to

SRAMs and PB. Then, instructions from SRAMs or PB will fill up IQ.

### 3.3 Module DECODE

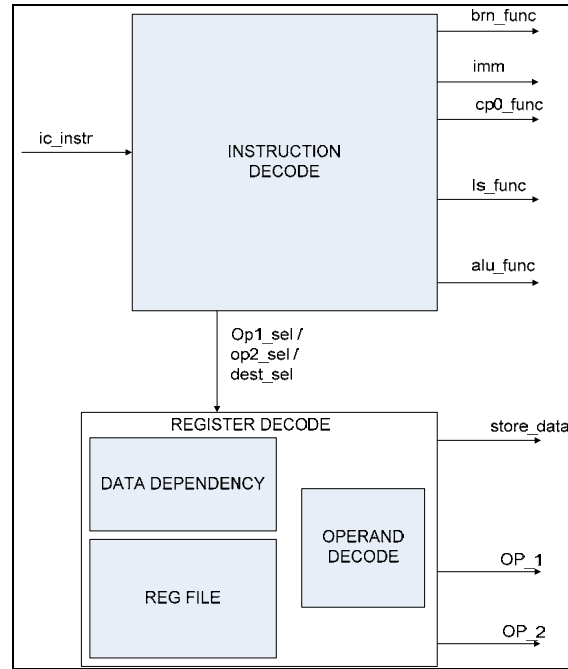


Figure 6. Block diagram of module DECODE

Figure 6 shows the block diagram of module DECODE. The module consists of 4 main blocks: INSTRUCTION DECODE, REG FILE, OPERAND DECODE, DATA DEPENDENCY

- INSTRUCTION DECODE: This block decodes the instruction supplied by module FETCH, then generates the following control signals:
  - Branch function (brn\_func)
  - Immediate value (imm)
  - CP0 function (cp0\_func)

- Load / Store function (ls\_func)
- ALU function (alu\_func)
- Operand select (op1\_sel, op2\_sel)
- Destination select (dest\_sel)

- REG FILE: This block contains 32 general purpose registers, and HI/LO registers.
- OPERAND DECODE: The purpose of this block is to choose operands. Two operands will be selected. The selection depends on control signals from INSTRUCTION DECODE and DATA DEPENDENCY.

- DATA DEPENDENCY: This block determines the dependency of data among the three following stages, then

sends selection signals to OPERAND DECODE.

### 3.4 Module EXECUTE

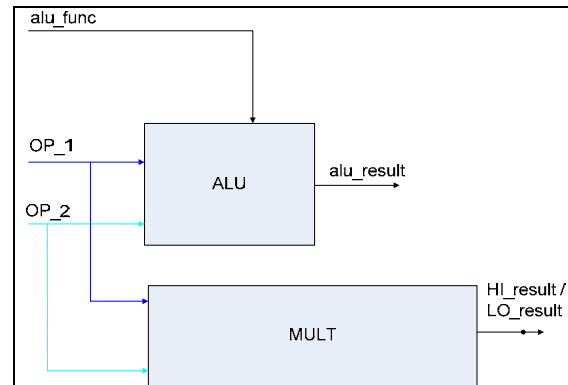


Figure 7. Block diagram of module EXECUTE

Figure 7 shows the block diagram of module EXECUTE. The module consists of 2 main blocks: ALU, MULT

- ALU (ALU stands for Arithmetic Logic Unit): This block calculates and generates results base on the two 32-bit operands OP\_1 and OP\_2, and control signal alu\_func. It is performed in one clock cycle. It do these following operation: add,

subtract, shift, and, or, xor, not, compare, etc...

- MULT (multiplier): This block multiplies the two 32-bit operands OP\_1 and OP\_2, and then generates 64-bit product. This product will be stored in HI/LO registers. MULT is performed in 2 clock cycles.

### 3.5 Module MEMORY

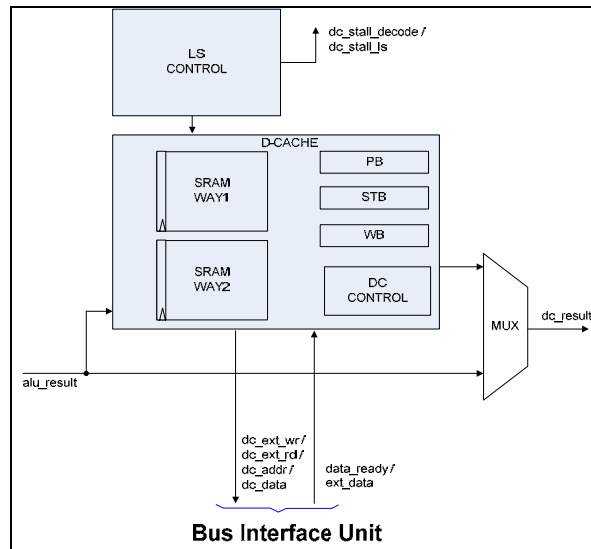


Figure 8. Block diagram of module MEMORY

Figure 8 shows the block diagram of module MEMORY. The module consists of 6 main blocks: LS CONTROL, STB, PB, WB, SRAMs, and MUX.

- **LS CONTROL (Load/Store Control):** This is a state machine that controls all the operations of module MEMORY. It gets signals from other block, then send back control signals to them. It also determines the time to write data to SRAMs.
- **STB (Store Buffer):** Data is temporarily stored in STB before being stored in SRAMs and WB.
- **PB (Prefetch Buffer):** This block fetches instructions from external memory and writes to SRAMs. It

sends handshaking signals to BIU and then gets data there.

- **WB (Write Buffer):** Data is pended in WB before being written to external memory.
- **SRAMs:** These are internal memory that is much faster than external memory. They are also called cache memory. They are used to temporarily store data that are fetched from external memory. Then data are read from SRAMs, instead of external memory.
- **MUX (multiplexer):** This multiplexer is used to select result from ALU or result from D-Cache.

### 3.6 Module WRITE BACK



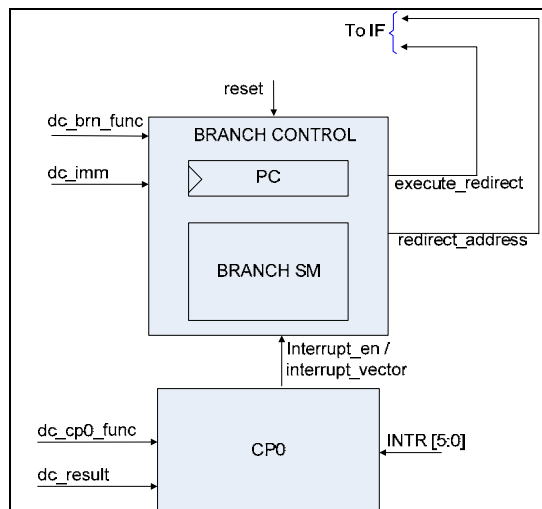


Figure 9. Block diagram of module WRITE BACK

Figure 9 shows the block diagram of module WRITE BACK. The module consists of 2 main blocks: BRANCH CONTROL and CPO.

- BRANCH CONTROL: This block controls the branching issue of the CPU. It contains register PC and BRANCH SM. The register PC holds the address of the current instruction.

BRANCH SM determines when performing a branch, and where the branch go to.

- CPO (Co-processor 0): This block contains CPO registers that hold configuration of the whole CPU system. It also controls the operation of interrupt and software trap.

### 3.7 Module BUS INTERFACE UNIT

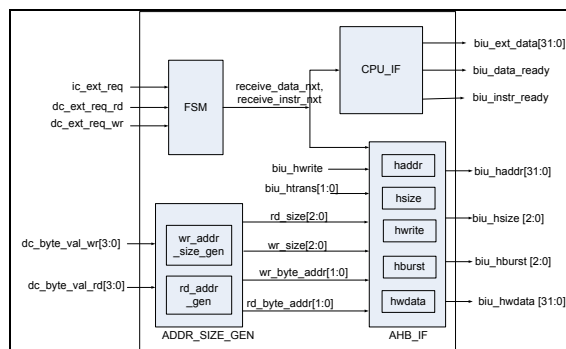


Figure 10. Block diagram of module BUS INTERFACE UNIT

Figure 10 shows the block diagram of module BUS INTERFACE UNIT. The module consists of 4 main blocks: FSM, ADDR\_SIZE\_GEN, CPU\_IF and AHB\_IF.

- FSM: This block is a state machine that controls the operation of 3 other blocks.
- ADDR\_SIZE\_GEN: This block generates addresses that are used to determines which byte/word is written or read. Besides, it also generates size of read/write data. The addresses and size will be used in AHB\_IF block.
- CPU\_IF (CPU interface): This block is used to communicate with CPU.
- AHB\_IF (AHB interface): This block is used to communicate with external bus.

#### 4. RESULTS

The VN1632 has been designed and fabricated using 0.13um IBM process. The prototype chips have been done with many applications. The results show that our chip worked corrently with desired performance. The characteristics of our design are as follows:

Process	IBM 130nm
Frequency	104 MHz
Power	30.6 (mW)
Resource	249606 Gates
Width	1144 (um)
Height	1138 (um)
Voltage	1.08 – 1.65 (V)
Temperature	-55 – +127 (C)
I/O Pad	284

#### 5. CONCLUSION

We have reported the architecture of the VN1632 which employs a five-stage pipeline. We observed that this pipeline architecture highly improves the microprocessor's performance. Furthermore, we also found that this architecture has many good features to work effectively. Therefore, this should be inherited in the next generation of Vietnamese 32-bit microprocessor.

KIẾN TRÚC VI XỬ LÝ 32-BIT KIỂU RISC CỦA VIỆT NAM, CHIP VN1632

Ngô Đức Hoàng, Hầu Nguyên Thanh Hoàng, Nguyễn Phú Quốc, Đỗ Ngọc Quỳnh

Trung tâm Nghiên cứu và Đào tạo Thiết kế Vi mạch

**TÓM TẮT:** VN1632 là vi xử lý đầu tiên do Việt Nam thiết kế. Thiết kế này dựa trên kiến trúc RISC Harvard 32-bit với kiểu đường ống 5 tầng (five-stage pipeline). Bài báo sẽ giới thiệu tổng quát về thiết kế, đồng thời trình bày phần thực hiện phần cứng của nó. Phần giới thiệu tổng quát trình bày và mô tả những đặc điểm chính của thiết kế, đó là: sơ đồ khối, tập thanh ghi, cấu trúc đường ống. Phần thực hiện phần cứng mô tả những chi tiết bên trong của từng khối. Một trình mô phỏng chi tiết được xây dựng để kiểm tra toàn bộ hoạt động của thiết kế. Sau khi hoàn thành, bản thiết kế được gửi đi chế tạo với công nghệ IBM 0.13um ở một nhà máy sản xuất chip của Mỹ. Chip VN1632 đã được kiểm tra thực tế và kết quả cho thấy rằng kiến trúc này đã hoạt động đúng với hiệu suất đã đề ra.

REFERENCES

- [1]. *The first made-in-Viet Nam 8-bit chip named RISC SigmaK3*, (2008): <http://forum.eetasia.com>
- [2]. *Vietnam - The Rising Tiger in the Semiconductor Industry*, (2008): <http://www.frost.com/prod/servlet/market-insight-top.pag?docid=125651805>
- [3]. *HN-07 microprocessor – the second Vietnamese microprocessor*, Science & Technology Development, Vol 12, No.16, (2009).