

Instituto de Matemática e Estatística  
Monografia dos curso Organização de Computadores

## Arquitetura do Console Nintendo 64

**Professor:** Siang Wun Song

**Alunos:** Antônio Augusto Abello  
Gustavo Estrela de Matos  
Lucas Romão Silva

São Paulo, 23 de Novembro de 2016

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Principais Componentes do Nintendo64</b>	<b>2</b>
<b>3</b>	<b>Chip NEC VR4300</b>	<b>2</b>
3.1	Principais Componentes do Processador VR4300 . . . . .	2
3.2	Formato de instrução . . . . .	4
<b>4</b>	<b>Chip SGI RCP</b>	<b>6</b>

# 1 Introdução

## 2 Principais Componentes do Nintendo64

### 3 Chip NEC VR4300

O chip NEC VR4300 é o principal processador no Nintendo64, responsável principalmente por processar a lógica dos jogos e, também audio. Esse processador foi desenvolvido pela empresa japonesa NEC e implementa a arquitetura de conjunto de instruções MIPS, desenvolvida pela empresa de mesmo nome. A arquitetura MIPS define um conjunto de instruções do tipo RISC, *reduced instruction set computer*.

O processador VR4300 possuía uma arquitetura compatível com instruções de 64 bits, apesar de grande parte das instruções do Nintendo 64 serem de apenas 32 bits. Especificamente nesse console, o processador da NEC trabalhava a uma frequência de 93,75 MHz.

#### 3.1 Principais Componentes do Processador VR4300

A figura 3.1 mostra um diagrama com os principais componentes do processador VR4300 e como eles se comunicam. Nesta seção, discutiremos o papel de cada um desses componentes.

- **System interface** define a interface do chip com os componentes externos. Essa interface é formada por um bus de 32 bits e vários outros bits para controle de bits, interrupções, relógio, etc. No Nintendo 64, algum desses bits se conectam a memória RAM do console e, também diretamente ao cartucho com o jogo.
- **Clock generator** é responsável por determinar o clock interno do chip, que é o clock do pipeline do processador. Esse componente recebe um clock externo e define o clock de pipeline como uma fração do anterior. Um dos modos de operação do VR4300 é ter o relógio interno funcionando com 2 ciclos a cada ciclo do relógio externo.
- **Instruction cache** é um cache que guarda a instrução que está sendo executada. No contexto do pipeline, manter uma cópia da instrução lida dentro do processador é essencial para aumentar eficiência e permitir operações como desvios e interrupções.

- **Execution unit** é uma parte do hardware que é especializada em realizar operações aritméticas. Na arquitetura MIPS, a *Execution unit* tem o papel de *COP1* (coprocessador 1).
- **CP0** é o componente responsável por fazer o controle da memória (MMU), isto é, permite endereçamento virtual, controla o acesso a pedaços de memória diferentes, aloca páginas de memória e outras funções. Para ajudar na eficiência de acesso a memória, o CP0 possui uma *TLB* (*Translation Lookaside Buffer*) que é, basicamente, um cache para a tradução de endereços virtuais para físicos.
- **Data cache** é um cache para dados da memória. Os dados armazenados nesse componente são indexados pelos endereços virtuais. Podemos ver, no diagrama, uma conexão entre *Execution Unit* e esse componente, essa ligação é utilizada quando um dado necessário em uma computação já está no cache. Quando isso acontece poupamos o trabalho de buscar dados na memória ou qualquer outro componente pois ele já está disponível dentro do chip.
- **Instruction address** tem o papel de calcular o endereço da próxima instrução. Esse circuito deve ser capaz de incrementar o *Program Counter*, dando sequência as execuções de instruções, e também deve ser capaz de atualizar o *Program Counter* quando ocorrem desvios (*branch* e *jump*).
- **Pipeline control** é o circuito que controla a execução do pipeline do processador. O conceito de pipeline consiste em dividir a execução de uma tarefa de maneira que mais de uma tarefa possa ser executada simultaneamente, desde que em diferentes etapas e não dependentes. O pipeline do processador VR4300 consiste em cinco etapas diferentes:
  - IC (Instruction Cache Fetch): faz a leitura da instrução.
  - RF (Register Fetch): deixa valores de registradores prontos para serem usados em cálculos.
  - EX (Execution): executa operações aritméticas.
  - DC (Data Cache Fetch): armazena no cache os dados utilizados na instrução.

- WB (Write Back): atualiza valores na memória ou registradores.

O que ocorre em cada etapa do pipeline também depende muito da instrução que está sendo executada, portanto o que está descrito logo acima é uma visão superficial de cada uma dessas etapas.

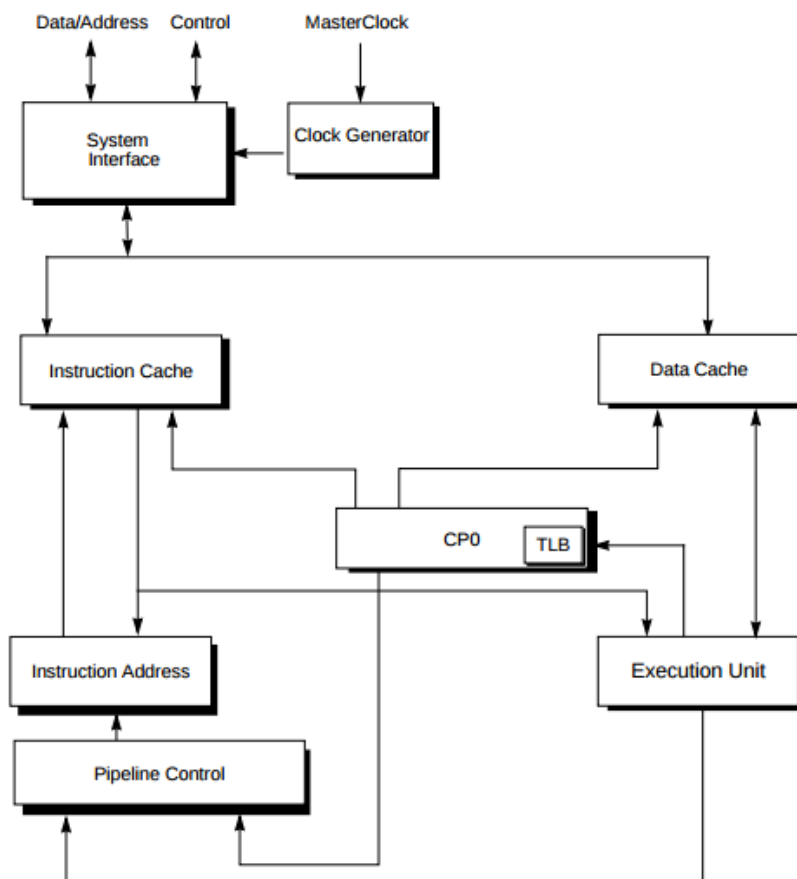


Figura 1: Um diagrama com os principais componentes do processador VR4300

## 3.2 Formato de instrução

Cada instrução do processador é formada por 32 bits, e elas podem ser separadas em três categorias: *I-type*, *J-Type* e *R-Type*.

As instruções do tipo *I-Type* são formadas por 5 bits, *op*, que determinam a operação; 5 bits em *rs* e mais 5 em *rt*, que determinam os registradores que estão sendo operados; e mais 16 bits, *immediate*, que pode representar ou um endereço ou uma constante. Exemplos de instruções desse tipo são as instruções *load* e *store*.

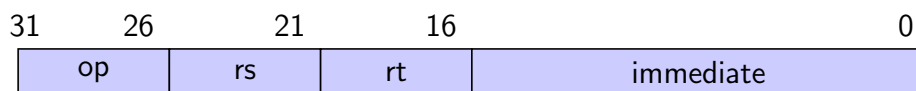


Figura 2: Formato de uma instrução *I-Type*

As instruções do tipo *J-Type* são usadas para controlar o fluxo do programa. Para isso, esse tipo de instrução pode pular para um pedaço específico do código por via de um *jump* ou um *branch*. Quando uma instrução do tipo *jump* é executada, o desvio sempre acontece, ao contrário da instrução *branch* na qual é possível determinar uma condição para o desvio. No VR4300 essas instruções são formadas por 5 bits, *op*, que determinam a operação; e mais 26 bits *target*, que determinam o endereço do possível desvio. Quando a instrução é um desvio obrigatório, os 26 bits estão todos disponíveis para determinar o endereço de destino, mas no caso de um *branch* o valor de *target* só pode determinar um *offset* de 16 bits relativo ao registrador *PC*.



Figura 3: Formato de uma instrução *J-Type*

As instruções do tipo *R-Type* envolvem apenas o uso de registradores. Exemplos de instruções desse tipo são aquelas que fazem operações aritméticas entre dois registradores e guardam o resultado em um terceiro registrador. Essas instruções são formadas por 5 bits *op*; 5 bits para cada um dos três registradores *rs*, *rt* e *rd*; 5 bits *sa* que definem um *shift* para o resultado; e mais 6 bits para *function*.

Code	Operation
0	Add
1	Subtract
2	Multiply
3	Divide
4	Square root
5	Absolute value
6	Transfer
7	Sign reverse
8	Convert to 64-bit fixed-point, rounded to nearest/even
9	Convert to 64-bit fixed-point, rounded toward zero
10	Convert to 64-bit fixed-point, rounded to $+\infty$
11	Convert to 64-bit fixed-point, rounded to $-\infty$
12	Convert to 32-bit fixed-point, rounded to nearest/even
13	Convert to 32-bit fixed-point, rounded toward zero
14	Convert to 32-bit fixed-point, rounded to $+\infty$
15	Convert to 32-bit fixed-point, rounded to $-\infty$
16–31	Reserved
32	Convert to single floating-point
33	Convert to double floating-point
34	Reserved
35	Reserved
36	Convert to 32-bit fixed-point
37	Convert to 64-bit fixed-point
38–47	Reserved
48–63	Floating-point compare

Tabela 1: Lista de todas as possíveis funções em uma instrução do tipo *R-Type* [1].

## 4 Chip SGI RCP

## Referências

- [1] NEC V<sub>R</sub>4300, V<sub>R</sub>4305, V<sub>R</sub>4310 64-bit processor User's Manual 7<sup>th</sup> edition. Japan, 2000.