|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **UNIVERSIDADE FEDERAL DE UBERLÂNDIA** | | | | | | | |
| **Faculdade de Computação - Campus Monte Carmelo** | | | | | | | |
| **Curso** | Sistemas de Informação | | **Período** | | 3º | **ano** | **Semestre** |
| 2013 | 01 |
| **Disciplina** | GSI013 Arquitetura e Organização de Computadores | | | **Avaliação** | | sub | **Valor** | 30,00 |
| **Professor** | Dr. rer. nat. Daniel Duarte Abdala | | | | | | **Data** | 23/09/2013 |
| **Aluno (a)** |  | | | | | | **Nota** |  |
| **Matrícula** |  | | **Vista em** | | \_\_\_ /\_\_\_ / \_\_\_\_\_\_ | | **Nota**  **Vista** |  |
|  |  | | **Assinatura** | |  | |  |  |

Observações:

* Prova individual e sem consulta;
* Valor total da prova: 20 pontos;
* A prova terá duração de 120 minutos a contar da leitura da prova (19:00 ~ 20:40);
* A prova deve ser respondida a caneta azul ou preta;
* Respostas ilegíveis não poderão ser corrigidas, e consequentemente, serão julgadas como incorretas;
* Não é permitido o uso de calculadoras;
* Desligue o celular;
* A interpretação faz parte da prova.

1. (5.0) Sabemos que a instrução **j** utiliza o tipo de instrução **Type-J**, que reserva 6 bits para o opcode e os 26 bits restantes para o endereço de memória para o qual deve ser executado o salto. Qual a faixa de endereços endereçáveis utilizando a instrução **j**? Caso seja necessário saltar para uma posição de memória que esteja fora do limite imposto pelos 26 bits de **j,** o que poderíamos fazer? Considere que desejamos saltar para uma posição na faixa 0x04000000 – 0xFFFFFFFF.
2. (10.0) Dado o programa abaixo, execute a conversão para assembly. Utilize a tabela 1 para converter o programa em assembly para código de máquina. Lembre-se de identificar qual o tipo de instrução está sendo codificada.

|  |  |
| --- | --- |
| Programa em C | Contraparte em Assembly |
| int v[10];  int x = 42;  int y = 21;  int z = x + y;  v[3] = z;  v[5] = x-y;  y = x;  v[5] = v[3] – x; |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

O programa acima requererá que dados sejam alocados no banco de memória de dados (Data Memory). Decida em que posições de memória alocar as informações necessárias.

Tabela – Lista de opcodes da implementação MIPS32-Mono

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instrução | opcode | | | | | |
|  | **5** | **4** | **3** | **2** | **1** | **0** |
| add | 0 | 0 | 0 | 0 | 0 | 0 |
| sub | 0 | 0 | 0 | 0 | 0 | 1 |
| and | 0 | 0 | 0 | 0 | 1 | 0 |
| or | 0 | 0 | 0 | 0 | 1 | 1 |
| slt | 0 | 0 | 0 | 1 | 0 | 0 |
| beq | 1 | 0 | 0 | 0 | 0 | 0 |
| lw | 0 | 1 | 0 | 0 | 0 | 0 |
| sw | 0 | 0 | 1 | 0 | 0 | 0 |
| addi | 1 | 0 | 0 | 0 | 0 | 1 |
| subi | 1 | 0 | 0 | 0 | 1 | 0 |
| j | 1 | 1 | 1 | 0 | 0 | 0 |

Em nossa implementação do MIPS32-Monociclo os registradores presentes no register-file são nomeados de $S1-$S31 (sequencialmente de 1 a 31). O registrador $ZERO possui valor constante definido em “0” e possui endereço 0.

1. (5.0) Considerando a Figura 1 que apresenta o diagrama esquemático do MIPS32-Monociclo descreva os valores dos sinais de controle necessários para a execução de uma instrução do tipo **SW**.
2. (5.0) Porque a implementação monociclo é raramente adotada em microprocessadores modernos? Justifique sua resposta utilizando exemplos.
3. (5.0) Crie uma descrição exemplo em VHDL que exemplifique a utilização das estruturas **entity**, **architecture**, **port**, **signal**, **when**. Descreva para que servem testbenches e em seu contexto, como funciona a diretiva **assert**.

**“I have a theory about the human mind. A brain is a lot like a computer. It will only take so many facts, and then it will go on overload and blow up.”**

**- Erma Bombeck**

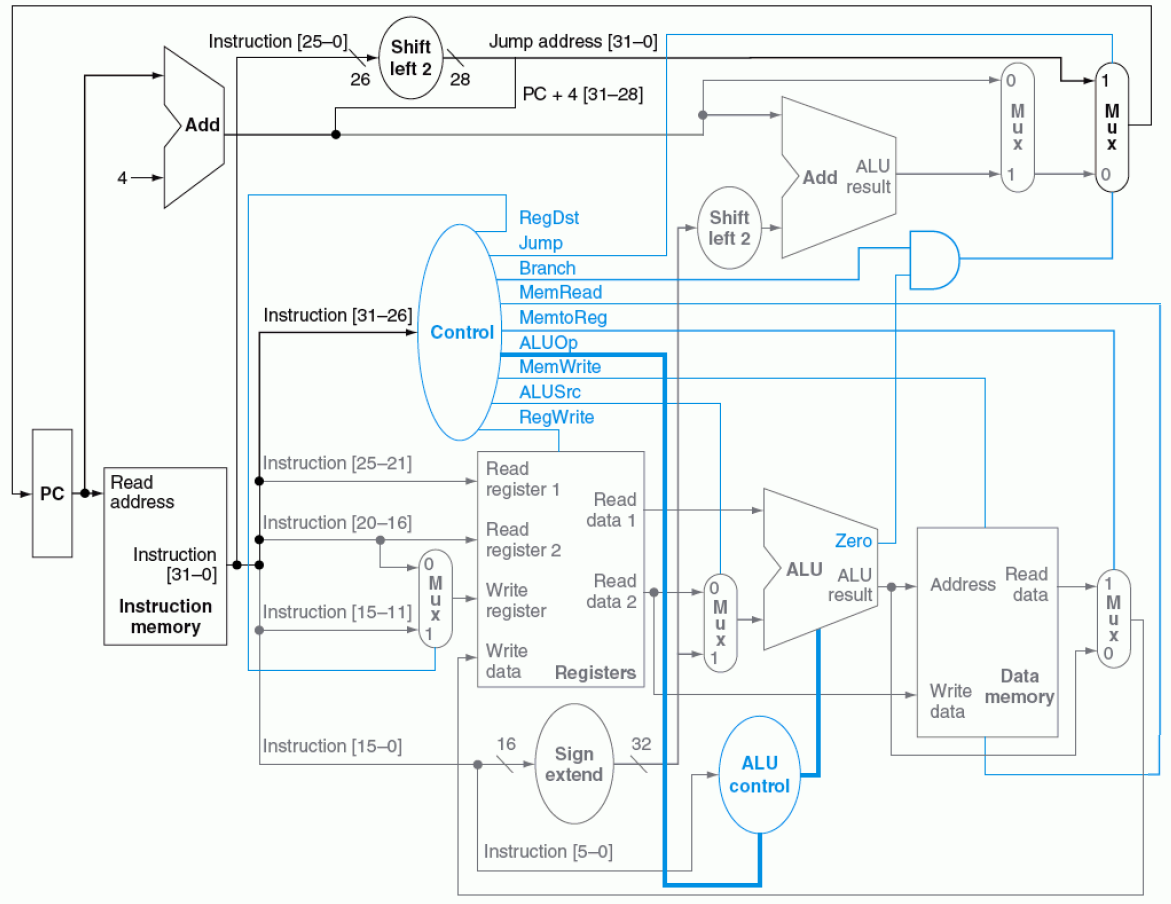


Figura – (Patterson, Arq. e Organização de Computadores, pg. 314) Diagrama do processador MIPS32-Monociclo

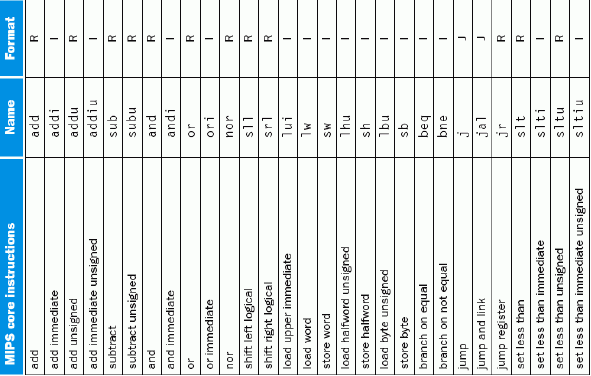


Figura – Lista de Instruções MIPS32