Universidade Federal de Minas Gerais

Departamento de Engenharia Eletrônica

Laboratório de Arquitetura e Organização de Computadores - 2022/1

08/04/2022

Arthur Coelho Ruback - 2019021140

Gabriel Pimentel Gomes - 2018020271

Lucas Rafael Silva Santos - 2013430595

**Decisões de projeto para o processador**

1. Instruction Set:
   1. NOP
   2. SYSCALL, para as diversas chamadas do sistema, como EXIT
   3. ADD (I), SUB(I), MUL(I), DIV(I)
   4. AND(I), OR(I)
   5. Carregar nos/recuperar dos registradores especiais: LDOP1, LDOP2, STOP3
   6. LW, SW
   7. JUMPR, JLINK, JUMP
   8. BEQ
   9. SLT(I)
   10. Possíveis instruções especiais para IO física
2. Tamanho da Instrução: **16 bits**
3. Tamanho das palavras de dado: **16 bits**
4. Memória endereçada **a cada 16 bits** (2 bytes)
5. Capacidade de memória será capaz de endereçar:

10,25 KB

1. Formas de endereçamento que a CPU será capaz de tratar:

relativo ao registrador base, pseudo-direto, relatIvo ao PC, imediato, registrador

1. Formas de entrada/saída que a CPU será capaz de tratar:

Interrupção

1. Banco de registradores com **16 registradores**
2. Priorizará uso de banco de registradores
3. Modelo RISC
4. Modelo Harvard
5. Dados armazenados no esquema **Little Endian**
6. Será feito um processador ciclo único seguido de um pipeline simples

Tabela 1. Mapa de memória:

| Faixa de endereços (hex) | Tamanho da mem. (bytes) | Utilidade da mem | Nome do componente vhd de memória | Tecnologia da mem |
| --- | --- | --- | --- | --- |
| 0x1FFF - 0x1000 | 4 KB (4096 B) | Memória de Programa | Mem\_program | FLASH |
| 0x0FFF - 0x0800 | 2 KB (1024 B) | Memória de dados dinâmica (heap/stack) | Mem\_data | RAM |
| 0x07FF - 0x0400 | 1 KB (1024 B) | Memória de dados estática | Mem\_data | RAM |
| 0x03FF - 0x0000 | 1 KB (1024 B) | Periféricos mapeados | Mem\_mapped | RAM |

**obs:** foram colocados mais bytes na memória de programa pois o tamanho reduzido das instruções nos obrigou a fazer várias “mini instruções” (instruções de deslocamento de dados, etc) para suprir esse ponto fraco. O aumento do número de instruções devido a essas “mini instruções” demandaria mais memória de programa.