

CI210 - Projetos Digitais e Microprocessadores

- Trabalho Prático -

Profa. Michele Nogueira - michele@inf.ufpr.br
Monitora: Juliana Franchi - jufranchi5@gmail.com

¹Departamento de Informática - NR2/UFPR

Entrega: As aulas reservadas para apresentação do projeto são as dos dias: 12, 14, 19 e 21 de Junho. A definição da ordem de apresentação das duplas será realizada em breve
(Não serão aceitos trabalhos fora deste prazo!)

Apresentação

Utilizando como base no picoProcessor (veja código fonte no site web da disciplina, seção 'Material de Referência'), você deverá implementar, dentre as opções de projetos abaixo, pelo menos **duas**. Este trabalho prático deverá ser feito em DUPLAS e exclusivamente em VHDL.

Dica: Antes de tentar implementar alguma nova instrução, procure entender o funcionamento completo do picoProcessor e identificar as fases de *Busca de instrução*, *Decodificação de instrução*, *Leitura de memória (quando aplicável)* e *Execução da instrução*. É importante também **projetar** antes de **implementar** :-).

Na seção 'Material de Referência' do site Web da disciplina, você encontrará a documentação do picoProcessor.

Modificações Propostas:

1. Soma em ponto flutuante

Implementar as operações de soma e subtração em ponto flutuante (*add.s* e *sub.s*) e as duas conversões (*cvt.s.w* e *cvt.w.s* que usam muito da mesma lógica);

2. Multiplicação em ponto flutuante

Implementar a instrução para multiplicação em ponto flutuante (*mul.s*)

3. Memória Cache

Construir uma memória cache associativa, que retém os dados e o endereço recentemente lidos da memória RAM, eliminando possíveis leituras redundantes. **Nota:** Assegure-se de, ao fazer operações com memória, buscar primeiro na cache e de escrever os dados da cache na memória principal.

4. Cache de instruções

Construir uma memória cache que lê algumas palavras a frente da memória ROM, de forma que a próxima instrução a ser executada provavelmente já esteja na cache, quando preciso. **Nota:** Assegure-se de manter algumas palavras já executadas, no caso de pequenos laços. **Extra:** Decodifique saltos incondicionais e continue a busca de palavras a partir do endereço de salto.

5. Byte e Half-word Load/Store

O picoProcessor atualmente somente faz *load* e *store* para palavras inteiras. Implementar a lógica extra para realização de operações de memória com palavras parciais.