

# ReFreeMIPS: uma arquitetura MIPS

## baseada em CGRA

Tiago Trevisan Jost

Universidade Federal do Rio Grande do Sul

Instituto de Informática

Orientado por Profa. Dra. Erika Cota



### MOTIVAÇÃO

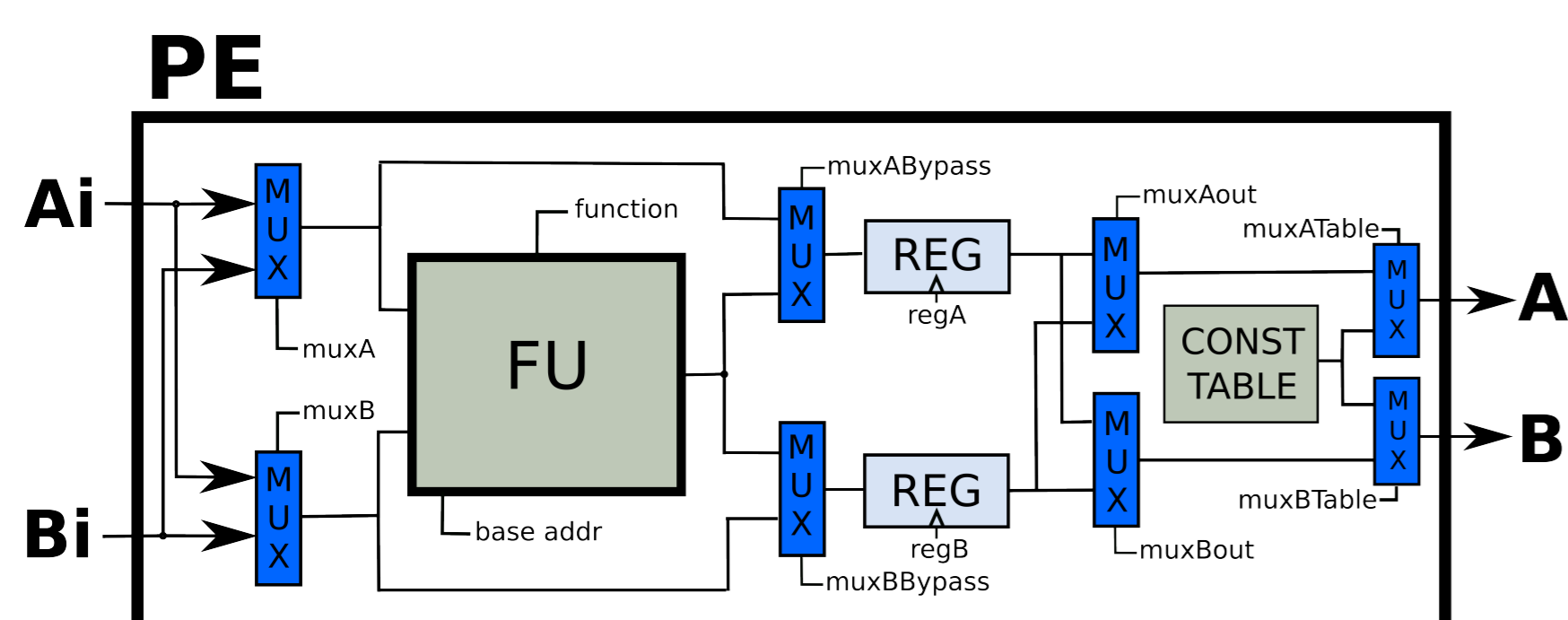
Apesar da grande utilização de microprocessadores em projetos de sistemas embarcados, outras arquiteturas foram imergindo neste cenário. Arquiteturas reconfiguráveis tem sido amplamente utilizadas por oferecerem grande flexibilidade e bom desempenho.

Muitas dessas arquiteturas foram propostas nos últimos anos. A maioria delas, porém, executa uma única ISA, normalmente exclusiva da arquitetura. A falta de compatibilidade com outras ISAs, como MIPS, limita sua utilização e dificulta a execução de código legado. Este trabalho propõe, portanto:

- A implementação da ISA do MIPS-I em uma arquitetura reconfigurável. Esta arquitetura do tipo CGRA (Coarse-grained Reconfigurable Architecture) denominada ReFree, possui como principal característica a ausência de um banco de registradores.
- Manter a compatibilidade com a ISA previamente proposta para o ReFree, portanto, não comprometendo o desempenho original da arquitetura.

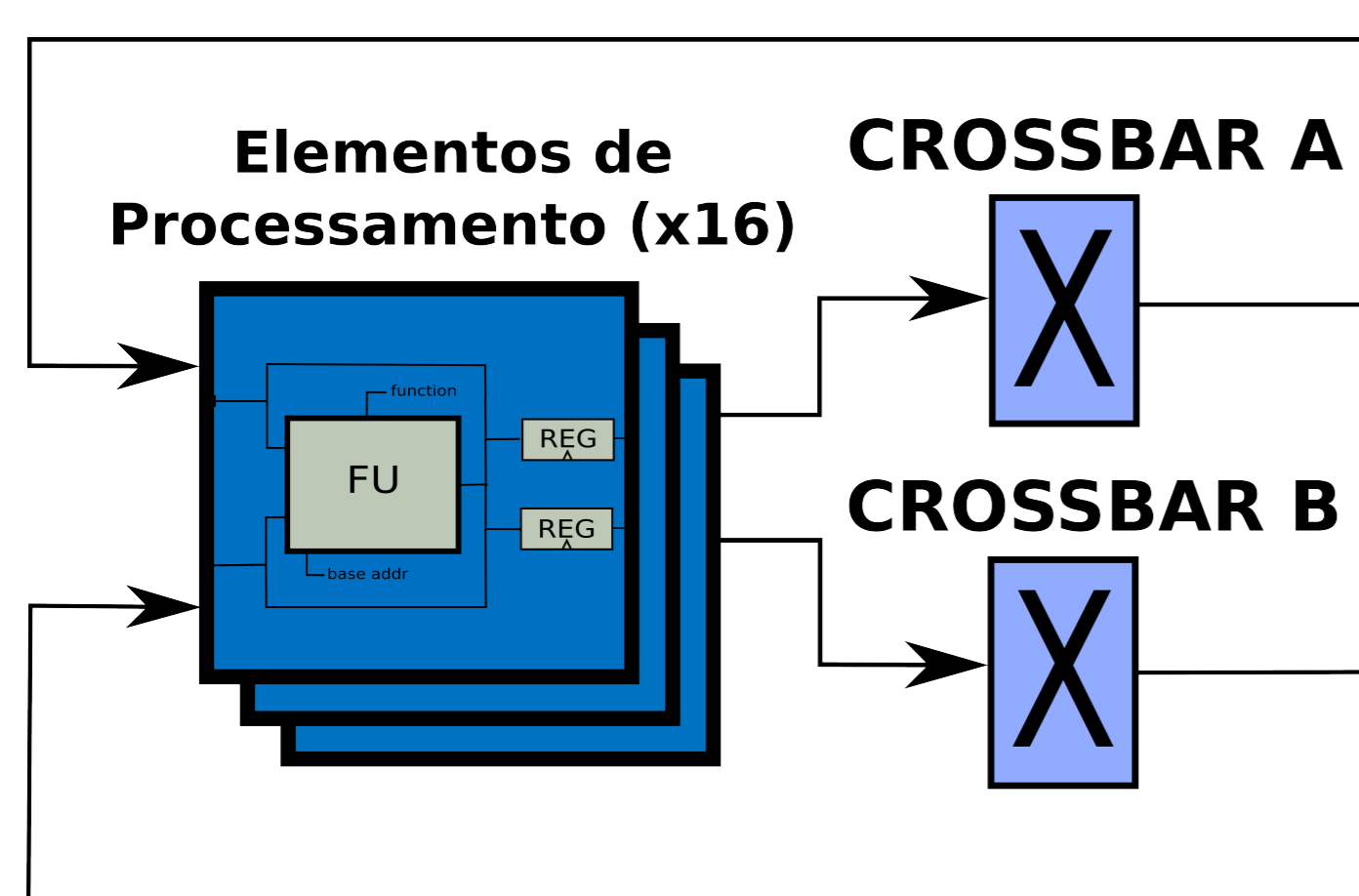
### IMPLEMENTAÇÃO

O ReFree é composto por 16 Elementos de Processamentos (PE) heterogêneos que podem operar como ULA (unidade lógica e aritmética), Multiplicador e interface de memória. Cada unidade possui 2 registradores internos, ou seja, a arquitetura possui um total de 32 registradores, mesma quantidade de uma arquitetura MIPS padrão.



Estrutura Interna das Unidades de Processamento.

Todas unidades são conectadas entre si através de duas redes crossbar, o que possibilita que a saída de uma PE possa ser utilizada como entrada de qualquer PE.



Topologia de Interconexão do ReFree.

Para a execução de instruções MIPS, somente 3 dessas Unidades são utilizadas: uma

ULA, um Multiplicador e uma unidade de interface de memória. Outras unidades somente serão utilizadas para acesso aos registradores internos, ou quando o modo de execução ReFree é utilizado.

A arquitetura é composta de um pipeline com os seguintes estágios: Busca de Instruções, Decodificação, Execução e Escrita no Registrador Destino.

- **Busca de instrução (IF):** realiza a busca da Instrução no endereço apontado pelo PC (Program Counter)
- **Decodificação (ID):** estágio responsável por decodificar a instrução e gerar os sinais para os estágios 3 e 4.
- **Execução (EX):** estágio no qual a instrução é efetivamente executada. Instruções do tipo **LOAD** e **STORE** realizam acesso à memória e cálculo de endereço neste estágio.
- **Escrita no Registrador Destino (WB):** responsável pela escrita no registrador destino (quando necessário).

### Pipeline Hazards

- **Hazard de Dados:** todas dependências de dados podem ser resolvidas através da técnica de *forwarding*, não necessitando de *stalls*.
- **Hazard de Controle:** assume-se que todo *branch* é *not taken*. Caso contrário, instruções são removidas do pipeline e busca-se a instrução no endereço correto.

### Funcionamento básico

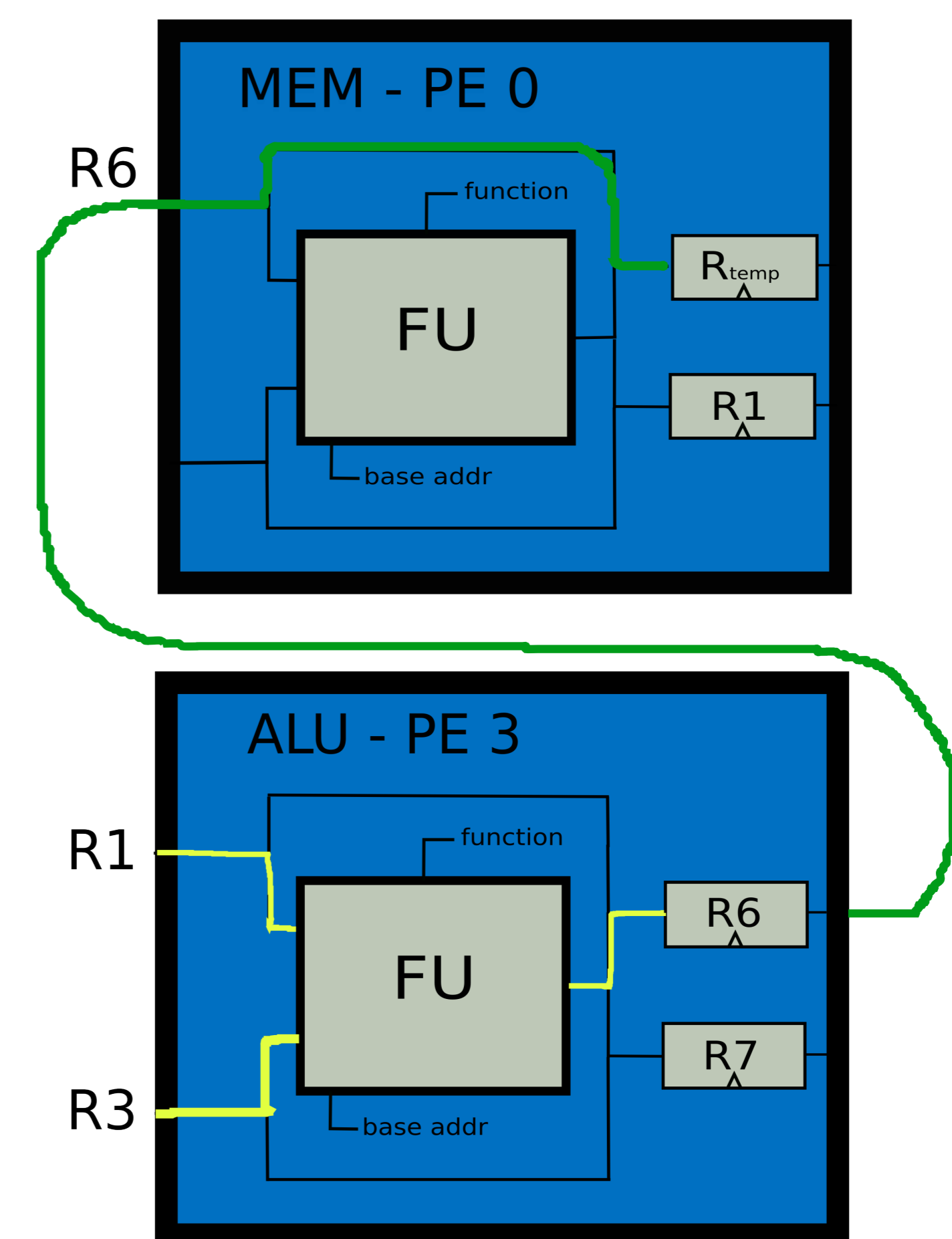
Os estágios de EX e WB são realizados no ReFree e grande parte das instruções necessitam de pequenas alterações para seu funcionamento. Valores dos registradores das PEs utilizadas nas instruções devem ser armazenados em um outro registrador temporário (tipicamente R0) e poderão ser reescritos quando o PE não for mais utilizado.

Exemplificando, a ULA está no PE 3, memória no PE 0 e multiplicador no PE 1. R0 representa um registrador temporário e não a constante de valor 0 (que é obtido da Tabela de Constantes).

Instrução Original	No estágio EX	No estágio WB
ADD R1, R1, R3	ADD R6, R1, R3 e MOVE R0, R6	MOVE R1, R6 e MOVE R6, R0
LOAD R7, 0(R3)	LOAD R0, 0(R3)	MOVE R7, R0
MFLO R6	MOVE R2, LO e MOVE R0, R2	MOVE R6, R2 MOVE R2, R0

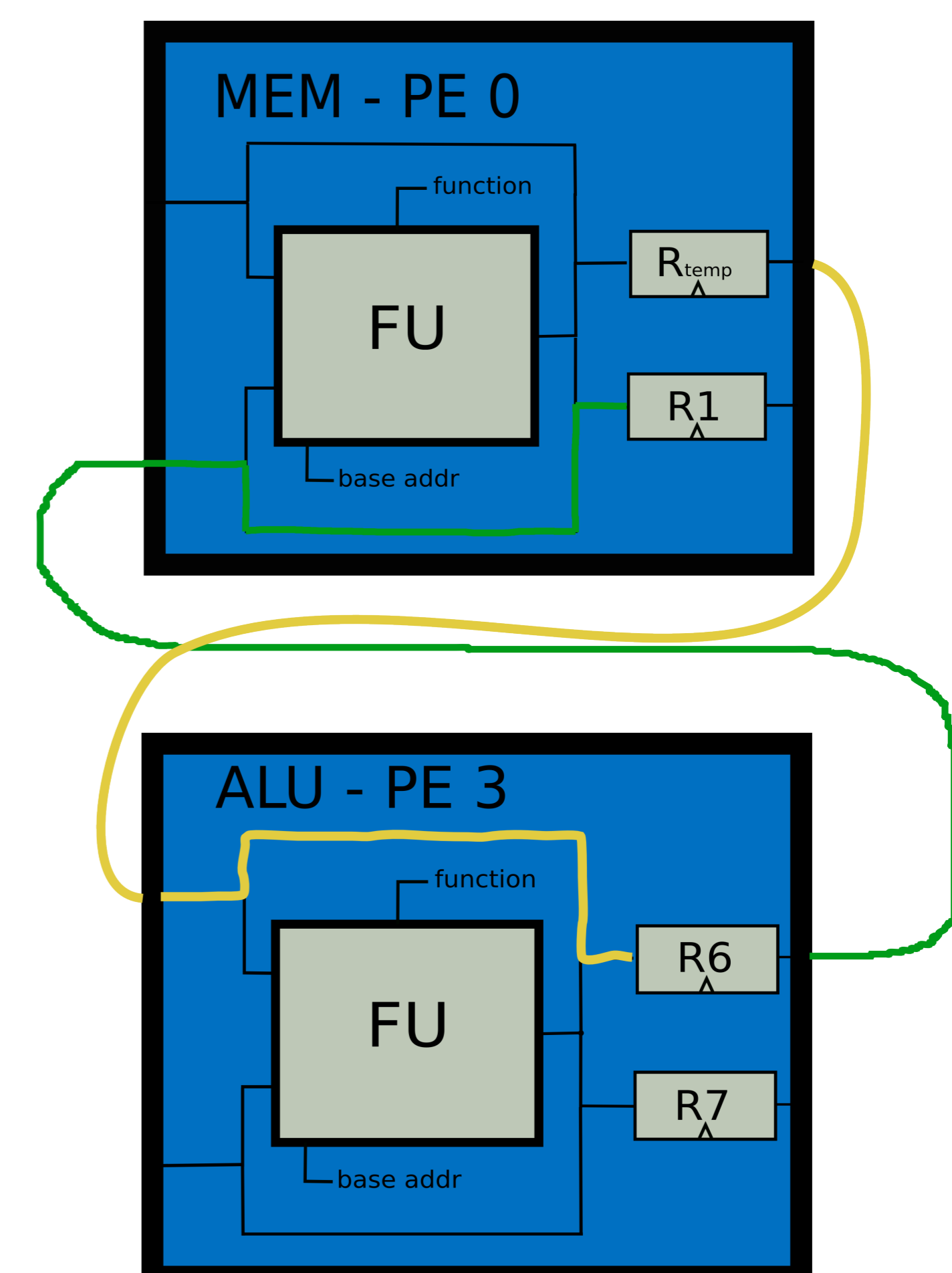
Exemplo da execução de Instruções de ULA, Memória e Multiplicação.

No estágio EX de uma instrução que utiliza ULA, deve-se armazenar o valor atual de R6 em R0 (registrador temporário) e realizar a operação definida na instrução.



Instrução ADD R1, R1, R3 no Estágio EX.

Já no estágio WB, é feita a escrita no registrador correto e o valor de R6 pode voltar para seu lugar original, caso este não seja utilizado novamente.



Instrução ADD R1, R1, R3 no Estágio WB.

### BENCHMARKS

Utilizou-se 5 benchmarks para verificação do funcionamento da solução proposta: ADPCM, DFT, CJPEG, Matrix e x264. Todos benchmarks foram executados com sucesso, não apresentado nenhum problema de incoerência.

### TRABALHOS FUTUROS

- Realizar a integração da ISA MIPS com a ISA ReFree, de modo a encontrar e explorar o maior paralelismo possível.
- Realizar a análise de consumo de potência e desempenho da solução e compará-la com diferentes arquiteturas.