

Relatório do Projeto de Modificação da Organização da Interface do Processador Plasma com a Memória

Vítor M. Da Rosa

Universidade Federal do Rio Grande do Sul

Instituto de Informática – Av. Bento Gonçalves, 9500 Campus do Vale – Porto Alegre, Brasil

vitormr@gmail.com

Resumo — *Este relatório descreve as modificações realizadas sobre o processador Plasma e os resultados obtidos. A principal alteração foi realizada na interface do processador com a memória de dados e instruções, passando o processador para uma organização Harvard. Isso permitiu aumentar o fluxo de troca de dados e instruções entre o processador e a memória, obtendo-se um aumento no desempenho de CPI (Clocks por Instrução) médio. Ademais, a implementação da organização Harvard, tornou desnecessária a utilização da lógica de controle utilizada para o compartilhamento da interface com a memória quando da busca de instruções e leitura/escrita de dados. Isso permitiu também diminuir a área do bloco operacional, ao ser retirado essa lógica de controle.*

Palavras Chave — *RISC 3000, MIPS ITM, Harvard, Plasma.*

I. INTRODUÇÃO

Dentro do escopo da disciplina de Arquitetura e Organização de Processadores 2006/1 [1], o projeto dessa disciplina resume-se em: modificar a arquitetura ou organização do processador embarcado (embutido), escolhido por cada grupo, de forma a obter um ganho significativo no seu desempenho ou no desempenho do sistema do qual esse faça parte. O projeto ainda deve ser desenvolvido de forma a apoiar ou até mesmo ser foco principal de algum tema de pesquisa sendo desenvolvido em nível de pós-graduação.

O processador escolhido para o desenvolvimento do projeto foi o processador Plasma [2]. Esse processador implementa a arquitetura MIPS ITM [3], a qual é baseada nas arquiteturas RISC 2000 e 3000 [4]. Os motivos que levaram a escolha desse core pelo grupo foram:

- a arquitetura do processador era conhecida;
- prover à referida disciplina um core de processador que implementasse a arquitetura RISC 2000/3000 em VHDL, permitindo assim, com que os próximos alunos dessa pudessem exercitar os conceitos aprendidos em aula;
- a de realizar um estudo aprofundado da arquitetura do core, MIPS ITM, para a sua utilização em projetos de pesquisa em NOC (Redes Intra-Chip) sendo atualmente desenvolvidos no grupo GAPH [5];
- a de utilizar um core que possuísse bons simuladores (ex. SPIM [7]), compilador (GCC [6]) e sistemas operacionais (Linux [8]) portados para sua arquitetura.

Para alcançar o que foi requerido para projeto da referida disciplina decidiu-se por modificar a interface do processador Plasma com a memória, passando-se para uma organização Harvard. Essa decisão é devida ao fato de que a implementação disponível desse core utiliza uma organização von Neumann. Dessa forma, o core utiliza a mesma interface para busca de instruções e leitura/escrita de dados e consequentemente a mesma memória para o armazenamento de instruções e dados. Em processadores que possuem mecanismos de pipeline para aumentar o sua média de CPI, como é o caso do processador Plasma, esse tipo de organização representa um gargalo entre o processador e a memória. Isso impede que o processador busque instruções ao mesmo tempo que acessa a memória de dados para realizar uma operação de leitura ou escrita.

Este relatório descreve as modificações realizadas no processador Plasma para solucionar os problemas mencionados acima e os resultados obtidos. A seção II a seguir apresenta uma breve descrição da arquitetura e organização do processador Plasma. Após, a seção III apresenta as alterações realizadas nesse processador. A seção IV apresenta os resultados obtidos com essas alterações. Finalmente, a seção V apresenta as conclusões obtidas com este trabalho.

II. DESCRIÇÃO DO PROCESSADOR

Como mencionado na seção I, o processador Plasma implementa a arquitetura RISC 2000/3000 de 32 bits. Essa arquitetura é do tipo load/store, ou seja, as operações lógicas e aritméticas são executadas exclusivamente sob 32 registradores internos ao processador e entre constantes imediatas e registradores. A organização do pipeline do processador está dividida em 5 estágios através de um único registrador de pipeline. Esse registrador armazena sinais dos 5 estágios dando a impressão da existência de 4 registradores ao invés de um.

Devido à interface do processador permitir acessar somente uma memória, instruções e dados são armazenados juntamente numa única memória. Ademais, o endereçamento para leitura/escrita de dados é orientado a byte, ao passo que a busca de instruções é realizada de quatro em quatro bytes. A Fig. 1 ilustra a organização da interface do processador plasma com a memória .

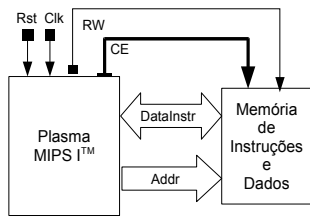


Fig. 1. Diagram Estrutural da Organização Original da Interface Processador-Memória .

Como ilustrado na Fig. 1 os sinais e os barramentos utilizados para comunicação com a memória de instruções e dados são os seguintes:

- **Addr:** barramento unidirecional de 30 bits para endereçamento da memória;
- **DataInstr:** barramento bidirecional de 32 bits para leitura/escrita de dados e leitura de instruções da memória;
- **CE:** sinal de 4 bits para controle de escrita/leitura de *byte*, *right/left half word* e *word*;
- **RW:** sinal de 1 bit para controle de operações de leitura ou escrita;

Visto que acessos a memória consomem um ciclo inteiro de clock, percebe-se que a troca de instruções e dados entre o processador e a memória é controlada por hardware adicional, interno ao processador, para permitir o compartilhamento do barramento **DataInstr** durante essa troca. Observando o bloco de controle da memória, **Mem_ctrl** e o fluxo de execução de instruções apresentados nas figuras Fig. 3, 4, 5, 6 e 7, disponíveis no Anexo B, nota-se essa limitação de acesso a memória pelo processador.

III. MODIFICAÇÕES REALIZADAS

Como mencionado no seção I, a principal modificação realizada foi na interface do processador com a memória, passando-se para uma organização Harvard. Nesse sentido, foram acrescentados dois barramentos e mais uma memória. A Fig. 2 ilustra como ficou a organização da interface processador-memória após as modificações realizadas.

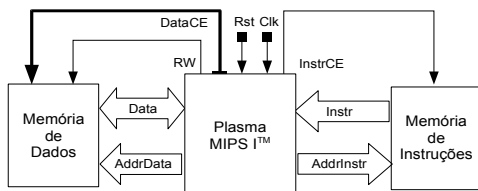


Fig. 2. Diagram Estrutural da Organização Harvard Implementada.

Assim, a interface da memória passou a conter os seguintes sinais e barramentos:

- **AddrInstr:** barramento dedicado de 30 bits para

endereçamento à memória de instruções;

- **Instr:** barramento dedicado de 32 bits para busca (leitura) de instruções;
- **AddrData:** barramento dedicado de 32 bits para o endereçamento à memória de dados;
- **Data:** barramento dedicado de 32 bits para leitura/escrita de dados;
- **DataCE:** sinal de 4 bits para controle de leitura/escrita de bytes, half words e words na memória de dados;
- **InstrCE:** sinal de 1 bit utilizado para controle da busca de instruções.

Observe que a memória de instruções é vista pelo processador como uma memória apenas de leitura. Dessa forma, foi utilizado somente o sinal **InstrCE** para permitir ao processador parar a busca de instruções, no caso de alguma instrução ocasionar uma parada (bolha) no seu pipeline.

Infelizmente o Processador Plasma utiliza uma organização extremamente complexa e fora dos padrões conhecidos para a sua arquitetura [1] [4]. Devido essa complexidade e o prazo do projeto, não se conseguiu modificar algumas partes desse processador, necessárias para o funcionamento correto de todas instruções previamente suportadas. Após as modificações, as instruções que não funcionaram corretamente foram a **sw** (store word) e **lw** (load word), devido a uma desincronização entre a escrita do dado com a escrita do endereço e sinais de controle. No entanto, a arquitetura MIPS I possui instruções de movimentação de dados, as quais também transferem dados pela porta **Data**. Como essas instruções funcionaram corretamente, para fins de comprovação do aumento de desempenho obtido utilizou-se as mesmas no lugar da instrução **st**. A seção a seguir apresenta os resultados para essa comprovação.

IV. RESULTADOS

Após implementar a organização descrita no capítulo anterior e simular a execução dos programas apresentados no Anexo A, obtiveram-se os resultados expostos na Tabela I abaixo.

Tabela I – CPIs Médios Obtidos

Programa	CPI Médio do Processador Original	CPI Médio do Novo Processador
<i>EXE.asm</i>	1,02	1,02
<i>Comum.asm</i>	2,57	2,27

V. CONCLUSÕES E TRABALHOS FUTUROS

Como foi observado na seção IV, a execução do programa **Exe.asm** não obteve nenhuma mudança de desempenho. Isso deve-se ao fato de que em programas que fazem poucos acessos à memória, mesmo executando sobre processadores com pipeline e organização Harvard,

não obtém aumento significativo de desempenho. No entanto, mesmo em programas que fazem um bom uso do conjunto de registradores, a maior parte das aplicações fazem utilizar uma quantidade razoável de dados. Nesse sentido, o programa **Comum.asm** foi utilizado para demonstrar o aumento de desempenho quando da execução desses tipos de aplicações sob o processador com a organização harvard.

Como mencionado na seção IV, devido as modificações realizadas, não se conseguiu que algumas instruções funcionassem corretamente. Em especial, o principal problema enfrentado foi o de modificar a entidade chamada **pipeline**. Essa entidade possui uma máquina de estados extremamente complexa, a qual é responsável pelo atraso de vários sinais de controle e barramentos internos ao processador. Futuramente, espera-se modificar essa entidade de forma a suportar todas as instruções da arquitetura desse processador.

ANEXO

A. Descrição dos Programas Utilizados para Testar as Modificações Realizadas

1) *EXE.asm*: Este programa assembler caracteriza-se por em sua maior parte conter instruções que realizam operações lógicas/aritméticas sob registradores, como por exemplo a instrução **add**.

2) *Comum.asm*: Este programa assembler utiliza todas as instruções da arquitetura MIPS I, exceto a instrução **st** devido ao problemas mencionados no texto.

A Tabela I abaixo apresenta o número de instruções obtido para a execução dos três programas assemblers mencionados acima.

Tabela II - CPI Médio Para a Execução de 3 Programas

Programa Assembler	Nº de Acessos a Interface de Dados	Nº Total de Instruções
EXE.asm	0	100
MixedLDEXE.asm	91	1220

B. Descrição da Operacionalização Original das Instruções pelo Processador Plasma

Quando gerado com com quatro registradores de pipeline, o processador utiliza cinco estágios:

- Primeiro estágio:

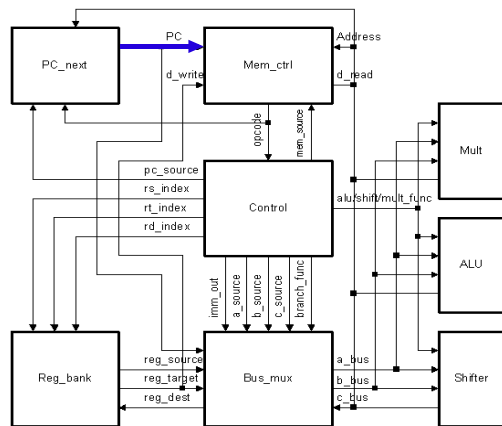


Fig. 3. IF (busca de Instrução).

- Segundo estágio:

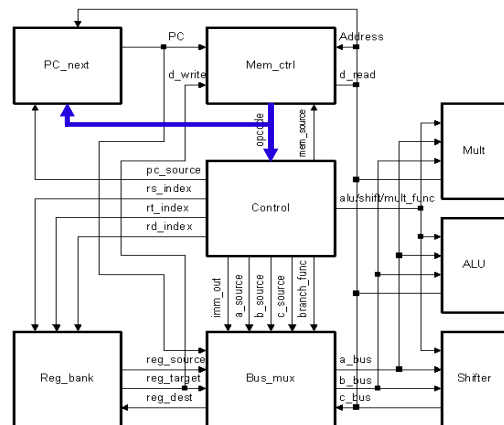


Fig. 4. ID (decodificação da instrução) .

- Terceiro estágio:

Fig. 5. EXE (execução).

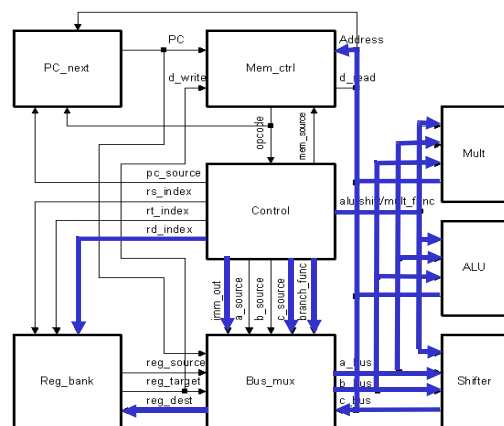


Fig. 6. WMEM (escrita de dado na memória) .

- Quinto estágio:

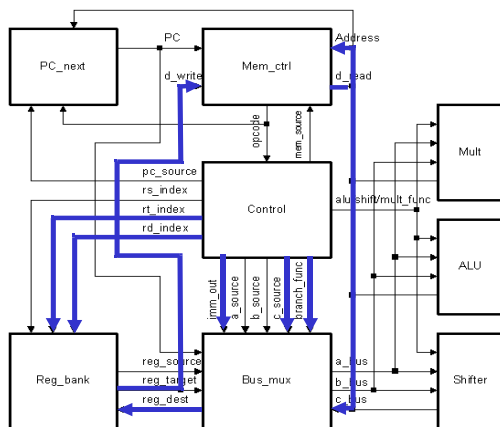


Fig. 7. RMEM (leitura de dado).

REFERÊNCIAS

- [1] F. R. Wagner. "Arquitetura e Organização de Processadores". Material disponível em: <http://www.inf.ufrgs.br/~flavio/ensino/-cmp237/cmp237.htm>. Julho de 2006.
- [2] S. Rhoads. "Plasma CPU Core". Disponível em: <http://www.open-cores.org>. Julho de 2006.
- [3] MIPS Technologies, Inc. Documentação disponível em: <http://www.mips.com>.
- [4] D. Patterson, J. Hennessy. "Organização e Projeto de Computadores: a Interface Hardware/Software". LTC, Rio de Janeiro 2000. Tradução da 2a. edição.
- [5] GAPH – Grupo de Apoio ao Projeto de Hardware. Página disponível em: <http://www.inf.pucrs.br/~gaph>.
- [6] GCC - GNU C Compiler. Documentação disponível em: www.gnu.org (linux); www.cygwin.org (windows).
- [7] SPIM. Disponível em: <http://www.cs.wisc.edu/~larus/spim.html>.
- [8] SPIM. Disponível em: <http://www.linux-mips.org>.