

Análise Física do Processador MIPS_Robot

Vicente S. Cruz¹, Henrique C. Freitas², Philippe O. A. Navaux¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

²Instituto de Informática – Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte – MG – Brazil

{vscruz,navaux}@inf.ufrgs.br, cota@pucminas.br

Resumo. Com a crescente evolução dos processadores, hoje é possível efetuar processamento de dados complexos em segundos. A robótica também compartilha essa evolução, permitindo que os controladores centrais possam gerenciar, de forma mais eficiente, o comportamento dos robôs. O objetivo desse trabalho é verificar o custo de área acarretado pela inclusão de instruções de auxílio ao movimento robótico no MIPS_Robot [Cruz, 2007]. Através da síntetização em FPGA, notou-se que a área sofreu um aumento aproximado de 2.5x em relação ao processador original. Um aumento aceitável dado o grande desempenho obtido com as novas instruções.

1. Introdução

A necessidade de desempenho computacional sempre foi um dos principais responsáveis pelos avanços tecnológicos. Graças a isso, hoje é possível realizar, em poucos segundos, processamentos que antes levavam horas ou dias. Os robôs compartilham esse desenvolvimento, pois os recentes processadores de controle central permitem que as máquinas executem tarefas consideradas impossíveis anteriormente. Uma vez que o desenvolvimento de um *chip* robótico é bastante complexo, seu desenvolvimento precisa ser efetuado em etapas. Em [Cruz, 2007] e [Cruz, 2008] é apresentado o MIPS_Robot, uma modificação do processador MIPS que teve o seu conjunto de instruções (*Instruction Set Architecture – ISA*) estendido com a inclusão de instruções de auxílio à movimentação de robôs. São elas: **Translação, Rotação em X, Rotação em Y, Rotação em Z, Seno e Cosseno**. Por fim, através da simulação de uma aplicação, mostrou-se que o *ISA* do MIPS_Robot possui um desempenho de quase 150 vezes superior ao MIPS original. Uma vez que os trabalhos anteriores avaliaram o desempenho do novo conjunto de instruções, este trabalho tem o objetivo de analisar os custos físicos do MIPS_Robot. Dessa forma, através de modificações do PLASMA, um processador que possui o mesmo *ISA* do MIPS original, foi possível obter os resultados de aumento de área que as novas instruções acarretam.

2. O Processador PLASMA

O PLASMA [Rhoads, 2001] é um processador de código fonte aberto, desenvolvido em VHDL, possui os cinco estágios de *pipeline* e implementa o mesmo conjunto de instruções do MIPS R3000. Assim, como o objetivo é verificar o custo de área das novas instruções, tal processador pode ser empregado a esse propósito. No entanto, a

descrição do PLASMA não inclui instruções de ponto flutuante (*Floating Point – FP*) no seu *ISA*, de modo que foi preciso incluí-las também. Para efetuar as modificações, foi utilizada a ferramenta *ISE Webpack 11.3*, da Xilinx® [Xilinx, 2009]. A figura 1 mostra a topologia lógica da organização do PLASMA.

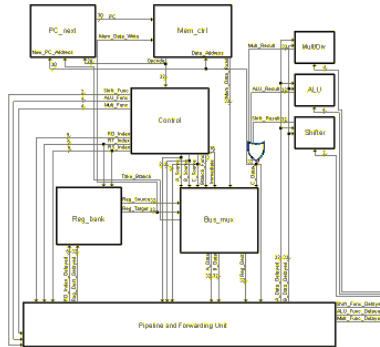


Figura 1. Topologia lógica da organização do PLASMA.

3. Modificações do PLASMA

A inclusão das unidades de operação responsáveis pela execução das novas instruções foi feita através da inclusão de *IP Cores*. Um *IP Core* é uma unidade de execução responsável por efetuar, dentre diversas tarefas, uma determinada operação aritmética ou lógica. Para implementar a unidade de *FP*, incluíram-se os *IP Cores* responsáveis pelas operações de **soma e subtração**, **multiplicação**, e **divisão em FP**. Para inserir a unidade de cálculo de seno e cosseno, utilizou-se o *IP Core* que implementa o algoritmo de CORDIC [Andraka, 1998]. Para que fosse possível efetuar a operação de translação, foram inseridos mais dois *IP Cores* de soma e subtração, de modo que fossem feitos três operações em paralelo para gerar a resposta. Por fim, a operação de rotação foi desenvolvida através da inclusão de quatro unidades de multiplicação, duas unidades de soma e subtração, e um conjunto de multiplexadores que encaminham os valores de entrada da unidade para serem processados, dependendo da operação de rotação a ser realizada. A topologia lógica da unidade de rotação, e a nova organização do PLASMA são mostradas nas figuras 2 e 3, respectivamente.

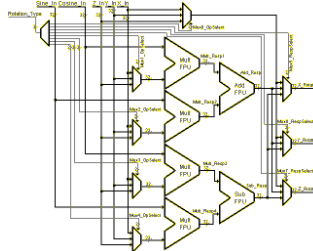


Figura 2. Unidade de rotação

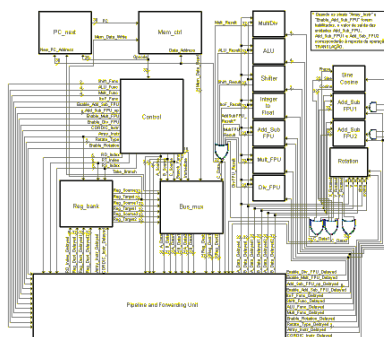


Figura 3. Nova organização do PLASMA.

4. Resultados

Os resultados de aumento de área foram obtidos através da sintetização da descrição do PLASMA para um dispositivo FPGA modelo XC5VLX50 da família Virtex5. A quantidade de área ocupada é medida pela quantidade de *Full Used LUT-FF Pairs*, *Slice LUTs*, *IO Buffers* e *Slice Registers* usadas. Abaixo segue as figuras 4, 5, 6 e 7 mostram a quantidade de elementos usadas pelo PLASMA original e pelo novo PLASMA.

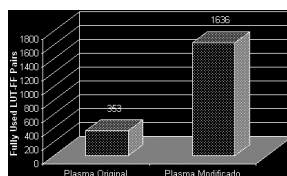


Figure 4. Quantidade de *Full Used LUT-FF Pairs*.

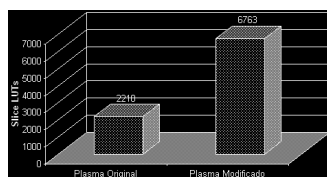


Figure 5. Quantidade de *Slice LUTs*.

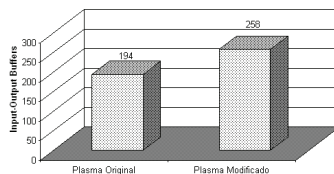


Figure 6. Quantidade de *IO Buffers*.

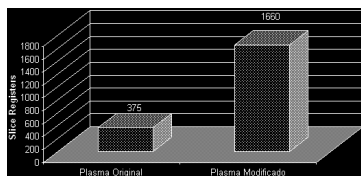


Figure 7. Quantidade de *Slice Registers*.

Nota-se que a quantidade de *Fully Used LUTs – FF pairs* usadas pelo processador modificado é 3,6 vezes maior que o original, seguido de um aumento de 2,06 vezes na quantidade de *Slice LUTs*. A quantidade de *IO Buffers* usada foi aproximadamente 0,33 vezes maior, enquanto que o total de *Slice Registers* aumentou 3,4 vezes.

5. Conclusões

O incremento do tamanho da área ocupada pelo PLASMA modificado foi mais do que o dobro que o PLASMA original porque foram incluídas diversas unidades repetidas. Dessa forma, é possível obter um processamento paralelo dos dados e maior aumento da vazão. Esse é o caso da instrução de translação, onde se utilizaram três *IP Cores* de soma e subtração em FP para gerar as novas coordenadas do ponto translado de forma paralela.

Assim, mesmo com o aumento de área, o desempenho do processador modificado em relação ao original é excepcionalmente maior, o que viabiliza a implementação do novo conjunto de instruções. Como trabalhos futuros, serão obtidos os valores de frequência e consumo de potência do novo processador.

6. Referências Bibliográficas

- Andraka, R. A. (1998) “A Survey of CORDIC Algorithms for FPGA Based Computers”, Sixth International Symposium on Field Programmable Gate Arrays. February 1998.
- Cruz, V. S., Freitas, H. C., Navaux, P. O. A. (2007) “Proposta de Expansão do Conjunto de Instruções do MIPS para Robótica”, WSCAD – CTIC, Outubro 2007.
- Cruz, V. S., Freitas, H. C., Navaux, P. O. A. (2008) “Avaliação do Pipeline no Processador MIPS_Robot”, 8ª Escola Regional de Alto Desempenho, Março 2008.
- Rhoads, S. (2001), “PLASMA – Most MIPS I(TM) Opcodes”, Disponível em <<http://www.opencores.org>>. Acessado em: Janeiro, 2008.
- Xilinx®, ISE Webpack, Última versão em 2009. Disponível em <<http://toolbox.xilinx.com>>. Acessado em: Janeiro, 2008.