

Processadores Assíncronos

Cássia Nunes
Helen Peters
Iúri Chaer

Arquitetura de Computadores
Programa de Residência em Tecnologia Google
{cassiasn, helenpeters, iuri.chaer}@gmail.com

1 Introdução

Circuitos assíncronos são sistemas cujo funcionamento é orientado a eventos, fazendo com que diferentes *bits* de um *chip* sejam processados com velocidades diversas, enviando e recebendo dados conforme a necessidade.

A maioria dos sistemas hoje em dia é síncrona, e devem a sua popularidade à simplicidade de projeto e implementação. No entanto, eles são extremamente ineficientes quando comparados aos seus contrapartes.

Vários estudos sobre processadores assíncronos foram realizados nas últimas décadas, a seguir alguns trabalhos são apresentados.

2. Revisão Bibliográfica

Em 1989, Martin, Burns e Lee [5] apresentaram o primeiro projeto completo de um microprocessador assíncrono. Controle e *data path* foram projetados separadamente e em seguida combinados de maneira mecânica. O *pipeline* foi considerado como um problema de programação concorrente. As fases do *pipeline* eram: *fetch*, *decode* e *execute*. Começando com um programa seqüencial para o processador, concorrência era introduzida através de séries de transformações de programa. As transformações eram definidas para sobrepor as fases de execução da instrução, mas elas não eram mecânicas nem únicas. O projetista decidia como decompor um programa em vários programas concorrentes. Apesar da simplicidade da arquitetura projetada e dos resultados modestos, este primeiro projeto foi significativo por introduzir interessantes concepções de operação, utilizar idéias de processamento tipo *pipeline* e pelo reduzido grupo de trabalho envolvido.

Paver investiga, em sua tese de doutorado [7], a pos-

sibilidade da construção comercial de circuitos complexos utilizando projeto assíncrono. Projetos assíncronos oferecem soluções mais econômicas para vários problemas que ocorrem em projetos síncronos. As principais áreas beneficiadas são: sincronização global, performance e consumo de energia. Os benefícios estão diretamente ligados a ausência de um *clock* global que coordene o funcionamento de todas as unidades do circuito. O processador AMULET1 foi projetado e implementado utilizando técnicas de projeto assíncronas. Ele foi o primeiro a ser compatível em código com outro já existente, o microprocessador ARM6. AMULET1 demonstrou que é possível implementar uma arquitetura RISC comercial complexa. O projeto resultante apresenta soluções para muitos problemas associados a processadores RISC modernos, como: suporte para exceções exatas, compatibilidade entre conjunto de instruções invertidas e operações em *pipeline*. O processador assíncrono AMULET1 não apresentou nenhuma vantagem significativa quando comparado com o processador síncrono ARM6. Contudo, o AMULET1 foi o primeiro processador assíncrono projetado utilizando a técnica de *Micropipeline*. Enquanto, o seu contraparte ARM6 faz parte da quarta geração de processadores síncronos e era um líder mundial em eficiência de energia e área do *die*. O AMULET1 é muito flexível, ajusta automaticamente sua performance a mudanças de temperatura e tensão, utilizando energia apenas quando existe algum trabalho útil a ser realizado. Existe um escopo considerável para melhoria da eficiência de energia e performance. Contudo, a penalidade de área para o controle lógico assíncrono é difícil de ser superada.

O trabalho proposto por Werner e Akella [10] apresenta aspectos da arquitetura de processadores assíncronos tais como organização do *pipeline*, *issue* de instruções, *branching* e tratamento de exceções e interrupções. Os autores apontam algumas desvantagens

gens de se utilizar sistemas síncronos, tais como: *clock skews* (em um sistema complexo o *clock* pode alcançar diferentes pontos de um circuito em instantes de tempo diferentes devido ao atraso resultante do aumento do tamanho dos fios), distribuição do *clock* (é necessário uma quantidade substancial de área de silício para distribuir o *clock* pelo sistema) e consumo de energia. Sistemas assíncronos são uma alternativa para contornar estas limitações. Estes sistemas não apresentam um sinal de *clock* que sincroniza todas as operações realizadas pelo sistema. Tipicamente, estes sistemas são compostos por unidades que se comunicam entre si através de protocolos *handshake*. A ausência de *clock* elimina o problema de *clock skew* e distribuição do *clock*, e o consumo de energia, em geral, é diminuído, já que porções do sistema não utilizadas podem ser “desligadas”. Sistemas síncronos permitem que cada bloco funcional possa ser otimizado sem a preocupação de sincronismo com um *clock* global. Por este motivo a performance do sistema reflete a performance média dos seus componentes individuais, ao contrário dos sistemas síncronos em que a performance é limitada pelo pior caso da performance de um determinado componente. Por outro lado, no caso de sistemas assíncronos a complexidade do projeto aumenta significativamente e por isso pode ser necessário utilizar mais hardware para implementar o sistema. Por estes motivos, existem poucos processadores assíncronos implementados. Os autores apresentam uma revisão de alguns processadores assíncronos [2, 8, 4, 3, 9, 5] e uma comparação qualitativa destes processadores.

Arvind e Mullins [1] propuseram uma arquitetura superescalar assíncrona baseada na técnica de composição de instruções (*instruction compounding*). Esta técnica define grupos de instruções dependentes, selecionando grupos (*compounds*) cujo grafo resultante seja um grafo dirigido acíclico (DAG). Dentro destes grupos resultados podem ser adiantados entre instruções sucessivas, permitindo um adiantamento dinâmico de dados baseados em informações locais, enquanto mantém as vantagens de se utilizar uma arquitetura assíncrona. Foram realizados experimentos que comparam a arquitetura proposta (utilizando *compounds* dinâmicos, definidos em tempo de execução, e estáticos, definidos em tempo de compilação) com uma arquitetura superescalar síncrona. Os resultados mostram que a performance pode ser melhorada no caso assíncrono por reduzir tempo gasto com sincronização em tempo de execução e por explorar paralelismo de instruções de forma mais eficiente.

Em [6] os autores propõem um processador assíncrono para redes de sensores sem fio (RSSF). Nestas redes a energia é a um fator crítico, já que a os ele-

mentos da rede apresentam uma fonte de alimentação limitada cuja recarga em geral é impraticável. As tarefas de sensoriamento, comunicação e processamento de dados são as responsáveis pelo consumo de energia nos nós, sendo que tipicamente a energia consumida com tarefas de comunicação (transmissão e recepção de dados) é maior do que a consumida pelas outras duas tarefas. Grande parte dos trabalhos propostos para RSSFs tentam amenizar o consumo de energia através da diminuição do tráfego de informações na rede através de um aumento da computação realizada localmente em cada nó. Por este motivo, diminuir a quantidade de energia consumida com processamento passa a ser uma tarefa cada vez mais importante. Neste trabalho os autores alcançam uma redução de consumo de energia significativa utilizando um processador assíncrono, comparado com o caso de um processador síncrono. O processador assíncrono projetado minimiza o consumo de energia por realizar gerenciamento de energia através de mudança dinâmica de tensão e frequência e trabalhar com uma tensão muito próxima do limite do processador.

Referências

- [1] D. K. Arvind and R. D. Mullins. A fully asynchronous superscalar architecture. In *IEEE PACT*, pages 17–22, 1999.
- [2] E. Brunvand. The nsr processor. *Proceeding of the Twenty-Sixth Hawaii International Conference on System Sciences*, January 1993.
- [3] K.-R. Cho, K. Okura, and K. Asada. Design of a 32-bit fully asynchronous microprocessor (FAM). *Proceedings of the 35th Midwest Symposium on Circuits and Systems*, IEEE Press, August 1992.
- [4] M. Dean. Strip: A self-timed risc processor. Technical Report CSL-TR-92-543, Stanford University, Stanford, California, 1992.
- [5] A. J. Martin, S. M. Burns, T. K. Lee, D. Borkovic, and P. J. Hazewindus. The design of an asynchronous microprocessor. *SIGARCH Comput. Archit. News*, 17(4):99–110, 1989.
- [6] L. Necchi, L. Lavagno, D. Pandini, and L. Vanzago. An ultra-low energy asynchronous processor for wireless sensor networks. *Proceedings of the 12th IEEE International Symposium on Asynchronous Circuits and Systems*, March 2006.
- [7] N. C. Paver. *The Design and Implementation of an Asynchronous Microprocessor*. PhD thesis, Department of Computer Science, University of Manchester, Manchester, UK, 1994.
- [8] R. Sproull, I. Sutherland, and C. Molnar. Counterflow pipeline processor architecture. Technical Report SMLI TR-94-25, Sun Microsystems Laboratories, Mountain View, California, 1994.

- [9] J. Tierno, A. Martin, D. Borkovic, and T. K. Lee. A 100-mips gaas asynchronous microprocessor. *IEEE Design and Test of Computers*, 11(2):43–49, 1997.
- [10] T. Werner and V. Akella. Asynchronous processor survey. *Computer*, 30(11):67–76, November 1997.