

## **Avaliação da Técnica de Dessincronização ao projeto de circuitos digitais: um estudo de caso com o Neander**

**CARVALHO, Tainã R.; JOÃO, Leonardo; SOARES, Rafael I.**

Universidade Federal de Pelotas – CDTec/Computação;  
{trcarvalho,ldrsjoao,rafael.soares}@inf.ufpel.edu.br

### **1. INTRODUÇÃO**

Nas últimas décadas a tecnologia CMOS (do inglês, Complementary Metal-Oxide-Silicon) desenvolveu-se vertiginosamente, impulsionada pela evolução da tecnologia submicrônica, que reduz consideravelmente as dimensões de transistores, componente elementar desta tecnologia. O avanço na tecnologia de fabricação de circuitos integrados em alta escala de integração (do inglês, Very Large Scale Integration - VLSI) proporciona o advento de SoCs (do inglês, Systems on Chip) (GAJSKI, 1997).

O desenvolvimento de SoCs em sua quase totalidade parte do pressuposto da discretização de tempo. Este pressuposto submete todas as entradas do sistema à temporização de um único sinal de controle, gerado externamente e denominado relógio (em inglês, clock). O pressuposto da discretização de tempo pelo uso de um sinal de relógio global no projeto de sistemas digitais é a característica principal do estilo síncrono de projeto. O tempo transcorrido entre transições do sinal de relógio permite que os valores de entrada amostrados sejam usados na computação de novos resultados. Este tempo garante a estabilização dos valores nas entradas e saídas de elementos de armazenamento, bem como a ocorrência e o descarte adequado de valores transitórios no interior do circuito.

O uso intensivo deste estilo de projeto na concepção de circuitos integrados VLSI bem como o uso da tecnologia de fabricação CMOS não traz apenas benefícios. A adoção de um único relógio para controlar todo o sistema digital traz alguns problemas, tais como a distribuição global do relógio, o escorregamento e o consumo de potência deste sinal (MOCHO, 2006). Estes problemas eram desprezíveis até meados da década de 80 ou facilmente tratáveis, porém começam a se tornar difíceis de resolver com a acentuada miniaturização da tecnologia VLSI.

A alternativa ao sistema síncrono são os sistemas assíncronos, onde o relógio é substituído por um conjunto de sinais para comunicação (handshake, em inglês), indicando quando um novo dado está disponível e quando este pode ser armazenado. Como vantagens é possível citar a eliminação do escorregamento de relógio, potencial para menor consumo de potência devido a taxa de atividade no circuito ocorrer apenas quando o circuito for exigido e desempenho no caso médio, não dependendo de um atraso crítico global. Por outro lado, sistemas assíncronos são complexos de projetar, pois o tempo é contínuo e não discreto e eventos podem ocorrer em instantes aleatórios o que torna o paradigma pouco usado na prática e com poucas ferramentas disponíveis ao apoio automatizado do projeto de circuitos. Neste contexto, dessincronização é uma técnica para automatizar o projeto de circuitos assíncronos a partir das especificações e ferramentas de projeto de circuitos síncronos. Desta forma, este trabalho apresenta um estudo de caso de aplicação de dessincronização a fim de avaliar suas vantagens e desvantagens sobre o processador didático Neader.

## 2. METODOLOGIA

O processo de dessincronização de um sistema é feito em 3 etapas de acordo com o (MADSEN, 2011).

- (i) Substituir registradores (FFs) por latches mestre/escravo;
- (ii) Medir o tempo de atraso de todos os caminhos dos circuitos combinacionais do sistema para avaliar os atrasos;
- (iii) Implementar controladores para os latches e elementos de atraso adequados a cada parte combinacional do circuito.

Neste sentido, a primeira etapa deste trabalho foi revisar o funcionamento da parte operativa e parte de controle da arquitetura do processador Neander. Um diagrama em blocos mostra o modelo da arquitetura do processador Neander na Figura 1.

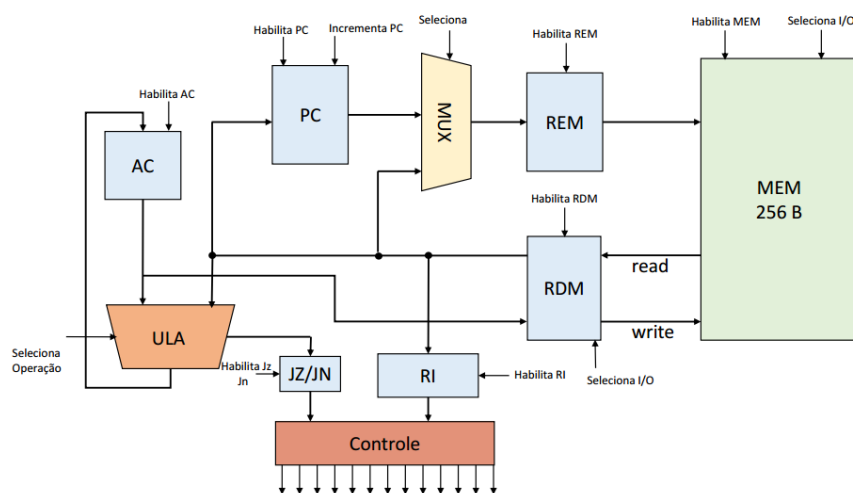


Figura 1. Arquitetura do processador Neander.

O Neander é uma arquitetura baseada em acumulador, com uma memória física de 256B compartilhada entre dados e instruções com largura de dados e endereços de 8 bits. Os dados são representados em complemento de 2. O conjunto de instruções baseado no acumulador disponibiliza as seguintes instruções lógicas e aritméticas ADD, OR, AND, NOT, além das operações de carga LD e STO e operações de salto e desvio JMP, JN, JZ e instruções de controle HLT e NOP. As instruções que manipulam dados em memória possuem 2 bytes, onde existe o *opcode* nos 4 bits mais significativos do primeiro byte e o byte de endereço.

A arquitetura é descrita em VHDL e prototipada em um dispositivo FPGA Cyclone IV da Altera. O projeto e validação da arquitetura utiliza o ambiente Quartus II da Altera. Com esta ferramenta é possível analisar o caminho crítico da arquitetura para estimar seus atrasos. Com base nestas informações, todo o projeto de dessincronização será baseado no maior atraso inicialmente, para que toda a arquitetura possa funcionar sem problemas de temporização.

A seguir é analisada a máquina de estados finitos (MEF) que controla toda a parte operativa do Neander, onde revisa-se todas as sequências de sinais que devem ser geradas para a execução de todas as instruções. Com base nestas informações, deu-se início a etapa de substituição dos registradores por *latches*

controlados por uma lógica baseada em protocolo de handshake. De acordo com (MADSEN, 2011) um protocolo de 4 fases baseado nos sinais de requisição (do inglês, Request - REQ) e reconhecimento (do inglês, Acknowledge - ACK) é suficiente para garantir a validade dos dados e tempo de propagação dos mesmos entre dois módulos funcionais de um circuito assíncrono.

Cada estado da MEF tem como objetivo controlar o funcionamento de determinados componentes da parte operativa. Para que um estado  $N$  libere os sinais para a parte do controle, ele espera uma resposta de um estado  $N-1$  através do sinal (REQ). Quando o estado  $N$  recebe em REQ (nível alto), ele envia um (ACK), que serve para desativar o estado anterior. E isso ocorre sequencialmente no circuito até que a operação seja executada.

Deste modo, com a substituição de todos os registradores e o sinal global de relógio por latches e lógica adicional de controle para o protocolo ACK-REQ temos a MEF assíncrona com treze estados apresentada na Figura 2.

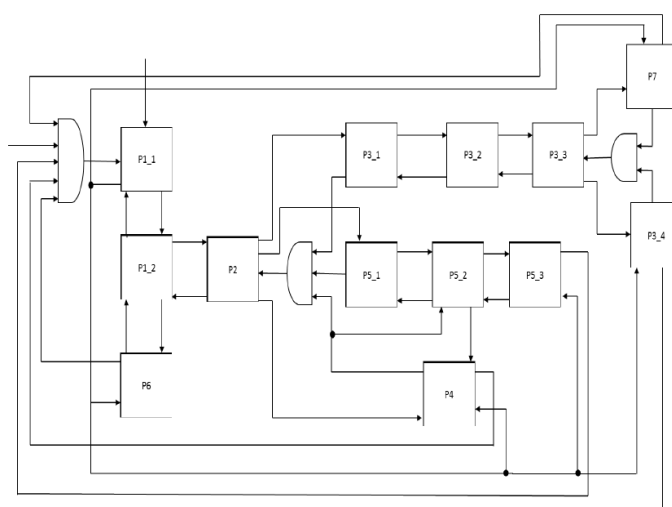


Figura 2. Diagrama em blocos do controle assíncrono após a dessincronização da MEF.

Analisando o diagrama na Figura 2, nota-se que em certos estados existem mais de uma entrada do sinal ACK, exigindo uma lógica OR. Isto ocorre devido ao compartilhamento do estado para mais de uma instrução onde existem mais de um sinal REQ. Os REQs de cada estado possuem diferentes atrasos, que estão relacionados aos caminhos críticos de cada componente e também dos próprios estados.

A análise do caminho crítico de cada estado e componentes da parte operativa é realizado com o Quartus II. No entanto, como a ferramenta faz otimizações no projeto durante a síntese do hardware ocorrem modificações nas avaliações dos atrasos de cada etapa. Por questões práticas, os atrasos obtiveram valores superestimados. Para simular esses atrasos, foi utilizado um método abstrato que a ferramenta disponibiliza chamado *Transport*. Este método faz com que a variação do sinal tenha um atraso definido pelo projetista, porém este artifício não permite a prototipação da arquitetura.

### 3. RESULTADOS E DISCUSSÃO

Após a implementação da arquitetura Neander dessincronizada são avaliados alguns parâmetros tais como área, caminho crítico, número de registradores e memória. Uma comparação entre as versões síncrona e

assíncrona é apresentada a fim de se estabelecer parâmetros de referência para esta nova arquitetura. A Tabela 1 resume os resultados obtidos.

Tabela 1. Comparação de parâmetros de síntese entre projetos do Neander.

Projeto	Nº de ALUTs	Nº de registradores	Memória (B)	Caminho Crítico
Assíncrono	211	24	256	9,357 ns
Síncrono	122	48	256	4 ns

Nota-se claramente observando a Tabela 1 que o projeto dessincronizado possui maior área. Isto se justifica pela implementação da lógica de controle de handshake existente entre os estados da MEF de controle. Com a substituição dos registradores da arquitetura por latches nota-se claramente que houve uma redução na versão assíncrona. Isto tem um potencial para redução de energia, pois o chaveamento ocorre apenas quando houver uma demanda de dados. Além disso, o tempo de propagação de dados aumenta devido principalmente a arquitetura ter sido projetada por uma superestimação dos atrasos entre os estados de controle. Este parâmetro tem um forte potencial de otimização na arquitetura a fim de reduzir o tempo de execução das instruções e com isso obter um melhor desempenho.

#### 4. CONCLUSÕES

Este trabalho permitiu avaliar a técnica de dessincronização de sistemas digitais síncronos a fim de obter um sistema digital assíncrono. Esta técnica é uma alternativa de projeto usada para projetar circuitos assíncronos com o suporte de ferramentas automatizadas de projeto de circuitos síncronos. O paradigma assíncrono é uma opção para solucionar problemas tais como consumo de potência, escorregamento de relógio e distribuição do sinal de relógio em sistemas síncronos. Um estudo de caso é realizado com a arquitetura do processador didático Neander. Os resultados obtidos demonstram a viabilidade da técnica avaliada, embora para isso muitos desafios são necessários, a começar pela ferramenta Quartus II que causa entraves na criação de elementos de atraso, por exemplo. Os resultados obtidos indicam que a arquitetura síncrona possui melhor desempenho que a arquitetura projetada. Isto se deve principalmente ao fato da técnica aplicada não estar adequadamente otimizada aos atrasos da parte operativa da arquitetura. Em trabalhos futuros está previsto avaliar o fluxo de projeto ASIC e também o ambiente de desenvolvimento da Xilinx para este propósito, além de se realizar as otimizações necessárias e a definição de um cenário para comparação mais justa entre as arquiteturas.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

GAJSKI, Daniel D. Principles of Digital Design, New Jersey: Prentice Hall, 1997

MADSEN, R. Desynchronization of Digital Circuits, Kongens Lyngby, Denmark, 2011.

MOCHO, R. Circuitos Assíncronos na Plataforma FPGA, Dissertação - Universidade Federal do Rio Grande do Sul, Porto Alegre, p.132, 2006.