MO 401 – Trabalho 01
Aluno: Márcio Paixão Dantas (049174)
Artigo: [1] MISHRA, P., DUTT, N., *Modeling and validation of pipeline specifications*, ACM Transactions on Embedded Computing Systems (TECS),Vol. 3 , pp. 114-139.

O fluxo tradicional de projeto *hardware*/*software* para sistemas embutidos assume o uso de processadores já em uso no mercado e que possuem compilador e simulador (*software toolkit*) disponíveis. Se o processador com as características desejadas não existe ou não possui *software toolkit*, é preciso desenvolver um. Este é um processo que demanda bastante tempo. Um fator que contribui para esta demora é que os projetistas costumam fazer modificações na arquitetura programável para cumprir requisitos como: baixo consumo elétrico, melhor desempenho, menor área e maior densidade de código.

Implementar o *software toolkit* manualmente é impraticável devido à natureza competitiva do mercado moderno. Usar linguagens de descrição de arquitetura (ADL) resolve bem este problema, permitindo explorar os parâmetros do espaço de projeto e, ao mesmo tempo, gerar automaticamente compilador e simulador para a arquitetura especificada.

O trabalho [1] apresenta uma técnica de modelagem em grafos e validação de arquiteturas que captura estrutura e comportamento de unidades de processamento, interfaces de entrada/saída e memórias. O objetivo é verificar se o *pipeline* modelado está bem formado analisando aspectos estruturais da especificação.

À partir de uma especificação em ADL EXPESSION, são obtidos dois grafos, G_A e G_B. O primeiro modela a estrutura do pipeline e, o segundo, o seu comportamento. Além dos grafos, uma função de mapeamento entre G_A e G_B também é gerada (mapeamento estrutura-comportamento).

$G_A=(V_A,E_A)$, onde os vértices, em V_A, representam unidades de processamento, dispositivos de memória, portas e conexões. Já as arestas, em E_A, são subdivididas em duas categorias: arestas de *pipeline* (conectam unidades) e arestas de transferências de dados (todas as outras combinações de vértices).

$G_B=(V_B,E_B)$ é um grafo composto por diversos subgrafos disjuntos, no qual os vértices representam os campos de cada instrução e, as arestas (orientadas), ordenam os campos. Cada subgrafo conexo maximal representa uma instrução ou operação. E_B também pode ser subdividido em duas categorias: arestas de operação (ordem dos campos na instrução) e arestas de execução (ordem em que os campos são usados durante a execução).

A validação usa uma abordagem descendente, ao contrário da maioria das técnicas existentes. Por enquanto, só aspectos estruturais podem ser validados. O projetista especifica quais propriedades devem ser validadas para que a arquitetura esteja bem formada. Entre as propriedades já implementadas, estão: conectividade (todas as unidades do modelo devem pertencer a algum caminho, seja de transferências de dados ou de *pipeline*), caminhos falsos (todos os caminhos devem realmente poder ser usados por instruções), completude (todas as instruções devem ser executáveis), finitude (garante o término de uma operação no *pipeline*) e outras propriedades específicas de arquiteturas.

A técnica de validação em [1] foi capaz de identificar problemas em diversas arquiteturas diferentes (ARM, DLX, MIPS R10K, TI C6x e Power PC). Segundo os autores, novas propriedades podem facilmente ser adicionadas. Este seria um tipo mais simples de validação e seu uso seria complementar à abordagem tradicional.