# Increased Scalability and Power Efficiency by Using Multiple Speed Pipelines

**Refêrencia:** Emil Talpes, Diana Marculescu. *Increased Scalability and Power Efficiency by Using Multiple Speed Pipelines;* ISCA 2005
**Autor do resumo:** Douglas Gielo Quinellato (RA: 057615)

## Resumo

Um dois grandes problemas enfrentados pelos projetistas de arquiteturas é a baixa escalabilidade de alguns dos elementos do pipeline quando se aumenta a freqüência do clock e o tamanho dos pipelines. Vários estudos mostram que é grande a diferença entre a escalabilidade do front-end e o back-end do pipeline, sendo o primeiro (Issue Window) mais devagar e menos escalável. Em face disso, este artigo propõe dois mecanismos que permite que estas duas partes sejam separadas, funcionando em sua velocidade ótima.

O primeiro desses mecanismos é o **Dual clock issue window**, que permite que o front-end trabalhe em uma freqüência diferente do back-end; o segundo e o **Pré-Scheduled execution**, que tenta manter a execução das instruções o máximo de tempo na parte mais rápida do pipeline. Para isso é proposto o uso de uma **execution cache**(EC) logo apos a *issue window*. Este cache armazena as instruções já decodificadas e com registradores renomeados.

Estes mecanismos funcionam utilizando-se de dois modos de operação: **trace creation mode** e **trace execution mode**. No primeiro, as instruções são lidas do I-Cache, e entram no pipeline pelo *front-end*. As instruções decodificadas são executadas e armazenadas no execution cache. Quando ocorre um *branch misprediction a* próxima instrução é procurada no EC; se for achado, o front-end do pipeline é desligado e as instruções são executadas diretamente do back-end, sendo buscadas no EC. Instruções no EC são armazenadas em *issue order*. Para o primeiro caso, o pipeline trabalha numa freqüência menor (*baseline)*, ditada pelo Issue Window; no segundo caso, trabalha-se numa freqüência maior, devido ao desligamento do front-end.

Para a verificação do desempenho, foi utilizado um processador de 9 estágios, superescalar (4-way), com I*ssue Window* de 128 entradas, suportando *issue* de 6 instruções por ciclo. Caches L1 de 64K com tempo de acesso de dois ciclos foi considerado um design bem balanceado para o *baseline*. Foram utilizados os SPEC95 e SPEC 2000 para a medição do desempenho.

O uso de Dual clock issue window, junto com a limitação de renomeamento de registradores levou a necessidade de adicionar 3 estágios no pipeline, causando uma queda de desempenho de 10% em alguns *brenchmarks*. Porém, a arquitetura apresentada sobrepõe essa penalidade, conseguindo uma melhoria de 5%, operando no modo *baseline.* Esta arquitetura pode utilizar o EC em 88% do tempo, o que dá uma boa oportunidade de melhoria de velocidade utilizando-se um clock mais rápido nessa situação.

Utilizando clocks diferentes observamos um aumento no desempenho maior, variante de acordo com o aumento das freqüências do front e do back end. Com um aumento de 50% na freqüência, temos um aumento de 56% do desempenho, por exemplo.

O consumo de energia diminuiu também com esta arquitetura Com uma implementação em 130nm, temos uma diminuição de 30%, ficando em 20% a redução em 65nm.