



UNIVERSIDADE FEDERAL DA BAHIA
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
ENGG57 – LABORATÓRIO INTEGRADO IV-A

Relatório de mesa 09/11/2017

Secretário de mesa: Everton Geovane de Cerqueira Quadros

Resumo

No encontro da última quinta-feira (09/11/2017) iniciamos a discussão acerca do código verilog implementado até então.

Pipeline 4

- O professor recomendou que utilizássemos os IPs de memória disponíveis.
- Além disso, notamos que o clock assíncrono no pipeline 4 funcionará desde que consiga estabilizar na metade do pulso do clock.

Ajustes no clock

- Deverão ser feitos ajustes na implementação do clock, já que da forma que se encontra atualmente, estamos serializando o mesmo.
- Todos os estágios do pipeline devem ser alimentados pelo mesmo clock. Para isso, precisamos estimar o tempo para estabilizar cada estágio. Podemos fazer isso compilando os pipelines separadamente e então verificando a frequência máxima de operação para cada um deles.
- Após o ajuste do clock, devemos verificar se o forwarding continua funcionando corretamente.
- É possível que haja necessidade de utilizarmos PLL.

Pilhas e Interrupções

- Criação de pilha (usando banco de registradores ou memória), e os controles da pilha (ponteiro, etc).
- Ao identificar erro do programador (ex.: pilha cheia, ou leitura de pilha vazia), sinalizar esse erro e paralisar o processador.
- Identificar as possíveis interrupções e seus níveis (ex.: interrupção do processador, SO)

Reset, eof, bolha

- Devemos substituir o RST assíncrono por um RST síncrono (estágio 1) e evitar identificar bordas de sinais não-periódicos, como do *pc_chg*.

- Observamos também que não há necessidade de usar o sinal *eof*, que indicaria que todo o programa já foi lido.
- Como será a implementação da “Bolha”?