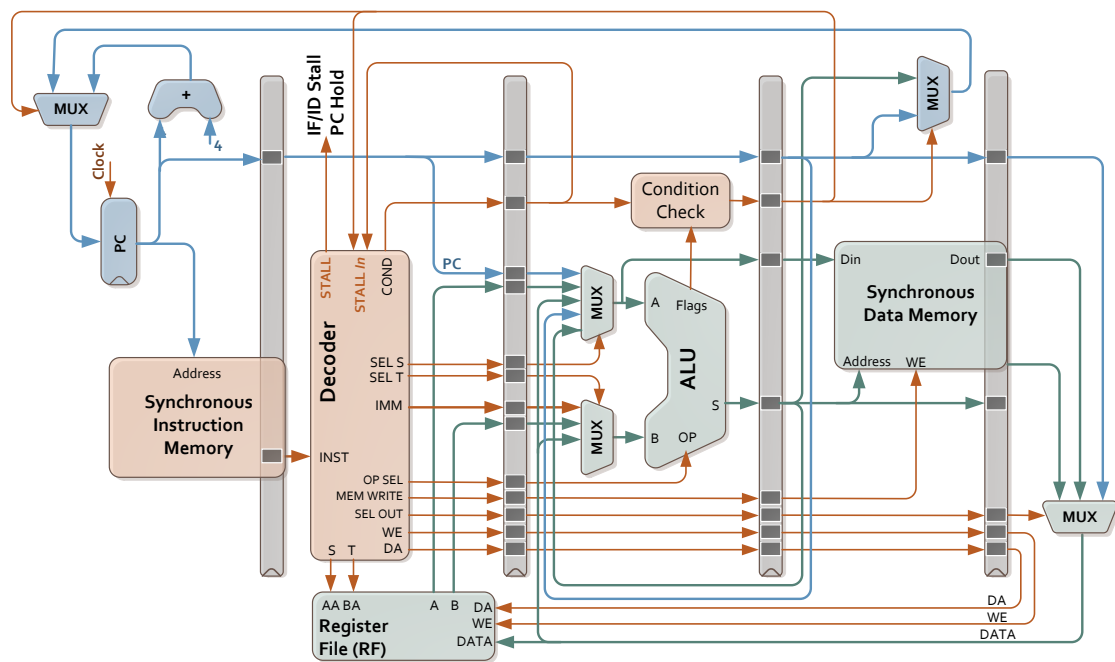


Arquitecturas Avançadas de Computadores (AAC)

2015/2016



Guia de Laboratório II

Desenvolvimento de um processador RISC *pipelined*

Demonstração a 3 de Maio nas aulas de laboratório

Entrega do relatório via fenix até às 23:59m de 8 de Maio

Departamento de Engenharia Electrotécnica e de Computadores

Instituto Superior Técnico

Abril de 2016

1. Projeto e implementação

O objetivo do segundo trabalho de laboratório consiste no projecto e implementação em VHDL de uma versão acelerada do processador RISC *pipelined* desenvolvido no contexto do 1º trabalho de laboratório.

Assim, o trabalho consiste em:

1. [Apenas para quem não terminou o trabalho de laboratório L1] Realização de todas as correções ao trabalho de laboratório 1, de forma a implementar um funcionamento *pipelined* correto.
2. Alteração da estrutura do processador de forma a acelerar a sua execução, tendo em atenção a estrutura do código exemplo (memórias de teste 1 e 2 e programa C exemplo previamente fornecido). Para tal pode, por exemplo, aplicar uma ou mais das seguintes técnicas
 - Introdução de *forwarding* de dados.
 - Introdução de um mecanismo de predição de salto (estático ou dinâmico).
 - Implementação de unidades de execução *pipelined* (ex.: multiplicador).
 - Execução de múltiplas instruções em simultâneo.

Para aumentar a performance do processador são permitidas quaisquer alterações estruturais ao processador, incluindo alterações ao número e forma dos andares de *pipeline*. No entanto, é obrigatório que o processador continue a suportar todas as instruções, mesmo que estas não sejam usadas no código de teste. De forma a minimizar os erros de projeto recomenda-se que discuta as alterações à arquitetura e os mecanismos para aumento do desempenho do processador com o docente.

Para facilitar a realização do trabalho e/ou evitar a utilização de técnicas que não produzam o efeito pretendido, poderá (opcionalmente) adicionar ao processador um conjunto de contadores, como por exemplo:

- Número de vezes que um dado caminho de *forwarding* é usado.
- Número de operações que são de fato efetuadas por uma unidade (ex: multiplicador ou barrel shifter)

Para quem não completou o 1º trabalho de laboratório, sugere-se a implementação de uma das seguintes técnicas:

- Alteração da estrutura do processador de forma a que a unidade de multiplicação seja *pipelined*
- Implementação de um preditor de saltos dinâmico.

Nota importante: o objetivo deste trabalho não é forçar o aluno a implementar todas as técnicas estudadas nas aulas teóricas. Pelo contrário, deixa-se ao critério do aluno a implementação da(s) técnica(s) que achar mais interessante(s) para os algoritmos escolhidos.

2. Entrega do trabalho

O funcionamento do trabalho de laboratório deverá ser demonstrado na aula de laboratório de 3 de Maio, de acordo com o projecto entregue via fenix até ao final do dia 8 de Maio. O ficheiro submetido no fenix (em formato ZIP) deverá conter:

- A. Relatório do trabalho seguindo o formato IEEE indicado na página da UC, contendo:
1. Identificação do trabalho e número dos alunos.
 2. Descrição da arquitectura de uma forma geral.
 3. Descrição das técnicas usadas para diminuição do CPI do processador.
 4. Relatório do processador indicando recursos utilizados (#slices, #blocos de RAM, ...) e frequência máxima de relógio. Para obter estes dados veja o relatório de *place-and-route* para uma FPGA Spartan 6 XC6SLX75. Pode ainda (opcionalmente) usar como segunda referencia o sistema embebido Zynq 7020.
 5. Uma descrição sucinta dos programas usados para avaliar o desempenho do processador.
 6. Uma análise do *speed-up* obtido com as técnicas para aumento do desempenho.
 7. Uma secção de conclusões.
- B. Ficheiros VHDL com a descrição do processador e dos *test benches*.

Nota: use sempre que possível gráficos e/ou tabelas para auxiliar a descrição dos circuitos e para indicar os resultados obtidos. Adicionalmente, não se esqueça de submeter os ficheiros do projecto (VHDL e C) no fenix juntamente com o relatório.

3. Referências

- [1] John L. Hennessy, David A. Patterson, “*Computer Architecture – A Quantitative Approach*”, 5th Edition, Morgan Kaufmann, 2011.
- [2] Xilinx, “MicroBlaze Processor Reference Guide”,
http://www.xilinx.com/support/documentation/sw_manuals/mb_ref_guide.pdf
- [3] Peter J. Ashenden, “The VHDL Cookbook”, 1st edition,
<http://tams-www.informatik.uni-hamburg.de/vhdl/doc/cookbook/VHDL-Cookbook.pdf>.
- [4] Xilinx, “XST User Guide”, v11.3, 2009,
http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_2/xst.pdf