

Departamento de Engenharia Elétrica e de Computação

SEL0384 – Laboratório de Sistemas Digitais I

Prof. Dr. Maximilian Luppe

PRÁTICA N°10

Aprendizado baseado em problema (PBL)

PBL02 - Circuitos Sequenciais - Banco de Registradores

Problema:

Implementar um Banco de SIZE Registradores de WIDTH bits utilizando linguagem de descrição de hardware VHDL em projeto parametrizável

Equipamentos necessários:

- Kit Mercurio® IV

Introdução:

Dando continuidade à construção do processador RISC-V, com a ULA já implementada, o problema seguinte é voltado agora ao Contador de Programa, aos Registradores Auxiliares e ao Banco de Registradores

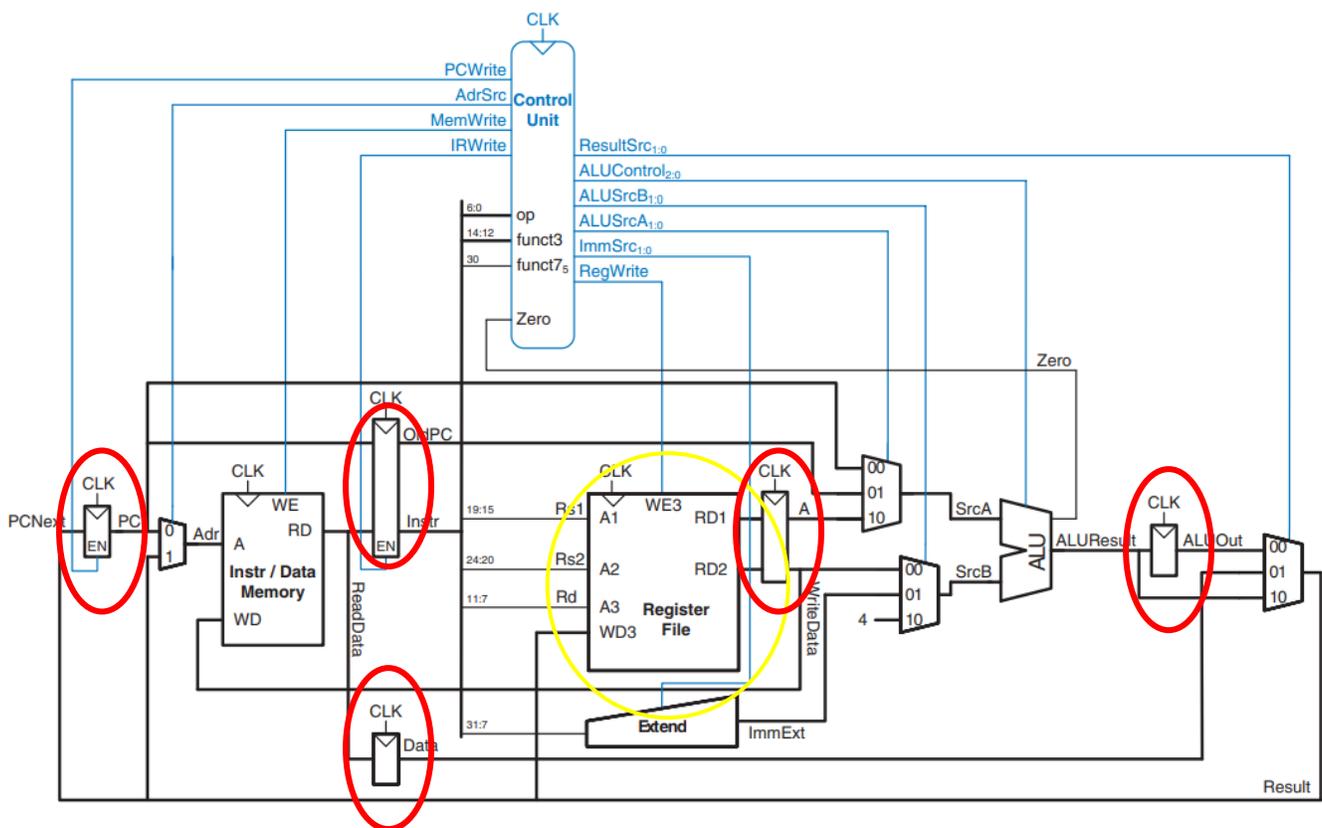
Sob o ponto de vista de Sistema Digitais, os Registradores são circuitos sequenciais formados por Flip-Flops (FF) que, sob o ponto de vista de Organização de Computadores, compõem o nível mais alto da hierarquia de memória. São memórias voláteis (ou seja, seus dados são perdidos quando o circuito é desligado), normalmente compostos por FF Tipo-D ligados em paralelo. Sua função é bastante simples: armazenar um dado de N bits por certo período de tempo. Os registradores possuem os seguintes terminais:

- Entrada de dados (N bits);
- Saída de dados (N bits);

- Clock;
- Set e/ou Reset, síncrono ou assíncrono;
- Habilitação para o clock.

Numa arquitetura podemos classificar os registradores em duas classes distintas: auxiliares (indicados em vermelho, na figura 1) e de dados (indicados em amarelo, na figura 1). Os registradores auxiliares são utilizados para o perfeito funcionamento da arquitetura, mas não são visíveis ao programador. Como exemplo temos os registradores A e B que são utilizados temporariamente pela ULA. Além destes temos o IR (*Instruction Register*), que armazena o código da instrução que está sendo executada. Os registradores de dados são aqueles utilizados pelo conjunto de instruções da arquitetura. No caso de uma arquitetura RISC-V, eles formam o Banco de Registradores. Nosso Banco de Registradores será formado pelos registradores de R_0 a R_{M-1} , sendo $M = 2^{\text{SIZE}}$.

Figura 1 - Processador RISC-V multiciclo



Fonte: Digital Design and Computer Architecture - RISC-V Edition <https://doi.org/10.1016/C2019-0-00213-0>

Como primeiro problema, o aluno deverá desenvolver um registrador de WIDTH bits contendo os sinais de controle EN (Clock Enable) e CLK. O componente irá compor uma entidade independente e será utilizado como todos os registradores independentes na organização da arquitetura.

Em seguida, utilizando o registrador construído na etapa anterior, será construído um banco de registradores com 2^{SIZE} registradores. O banco de registradores deverá ter:

- 1 Entrada de dados (WD3);
- 2 Saídas de dados (RD1, RD2);
- Seletor de registrador para escrita (A3);
- 2 Seletores de registrador para leitura (A1, A2);
- Clock (CLK);
- Habilitação de escrita (WE3).

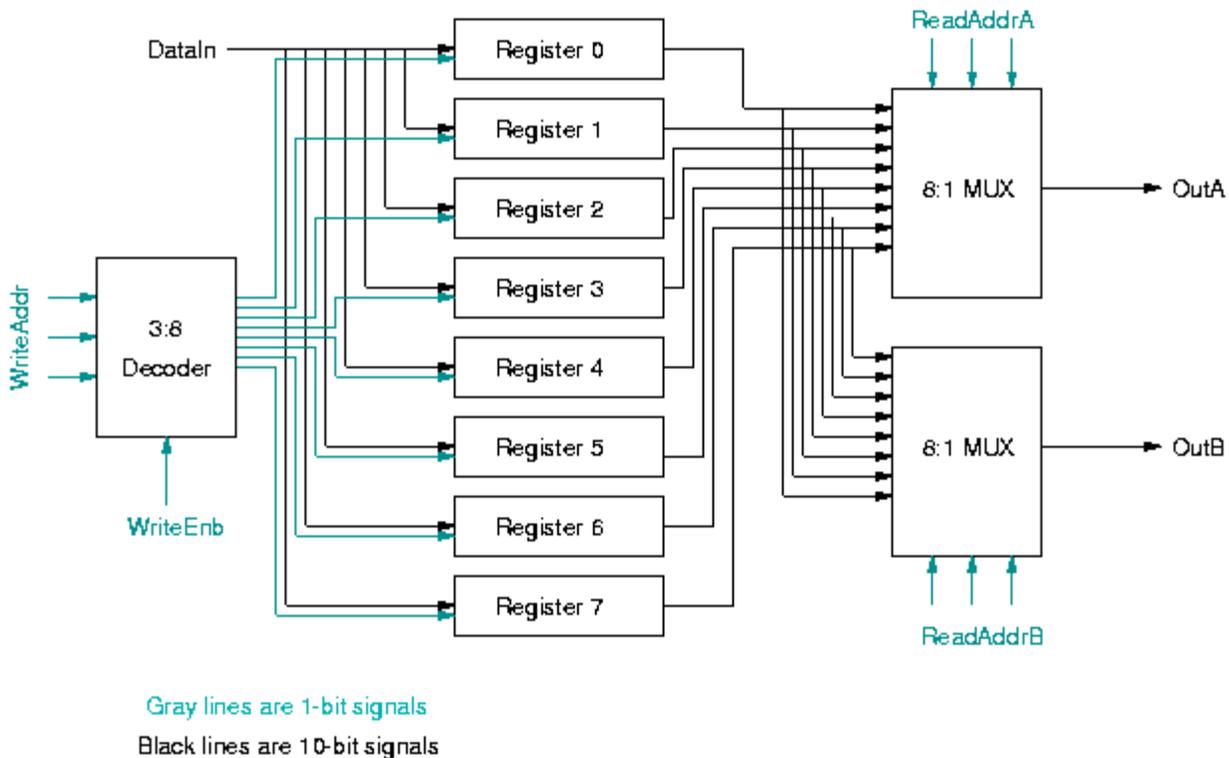
Utilize Multiplexadores e decodificadores para fazer a seleção dos registradores. Quando WE3 é ativado (igual a '1'), o valor presente em WD3 é armazenado no registrador indicado por A3, após a transição do sinal de CLK. O valor dos registradores indicados por A1 e A2 são apresentados nas saídas RD1 e RD2, após a transição do sinal de CLK.

O relatório deve descrever, de forma sucinta, conceitos de banco de registradores em arquiteturas de 32 bits.

Procedimento Experimental:

Apresentar a implementação de um registrador genérico de WIDTH bits, com entradas de CLK e EN, e como este pode ser utilizado para implementar um Banco de Registradores, baseado na arquitetura RISC-V (figura 1), com barramento de dados de tamanho parametrizável, utilizando a linguagem de descrição de hardware VHDL.

Figura 2 - Configuração de um banco de 8 registradores



Fonte: [site](#)

Criar uma pasta denominada `MercurioIV_RegBank`, com as subpastas `docs`, `modelsim`, `quartus` e `src`, e criar um projeto na pasta `quartus`, também denominado `MercurioIV_RegBank`.

Abrir o projeto no Quartus Lite e implementar o Banco de Registradores utilizando VHDL, denominado `RegBank.vhd`, armazenando o código na pasta `src`. Incorporar o código do Banco de Registradores ao projeto principal (`MercurioIV_RegBank`), com `WIDTH = 4` e `SIZE = 2`, ligando as chaves `SB[3:0]`, `SB[7:4]` e `SW[3:0]` às entradas `A1`, `A2` e `A3`, os push-buttons `KEY[8]` e `KEY[5]` às entradas `CLK` e `WE3`, a chave `SA[3:0]` à entrada `WD3`, as saídas `RD1` e `RD2` aos display de 7 segmentos `DISP0` e `DISP1` por meio de decodificadores binário para 7 segmentos, e executar o projeto no kit Mercurio® IV.

Apresentar código VHDL, circuito RTL, número de células lógicas utilizadas e foto do kit com o circuito funcionando.