

**Departamento de Engenharia Elétrica e de Computação**

**SEL0606 – Laboratório de Sistemas Digitais**

**Prof. Dr. Maximilian Luppe**

**PRÁTICA Nº7**

**Aprendizado baseado em problema (PBL)**

**PBL02 - Circuitos Sequenciais - Banco de Registradores**

**Problema:**

Implementar um Banco de M Registradores de N bits utilizando linguagem de descrição de hardware Verilog em projeto parametrizável

**Equipamentos necessários:**

- Kit DE10-Lite

**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

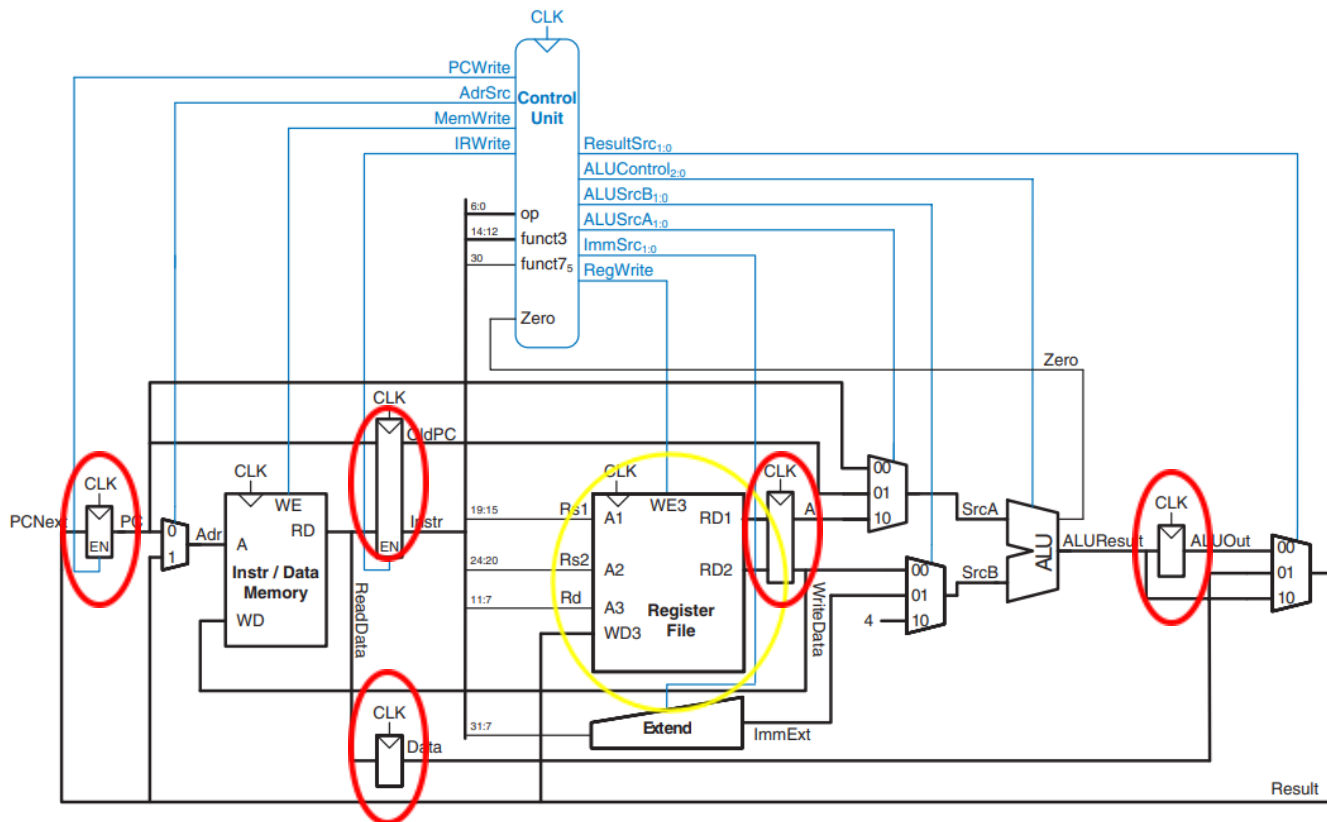
Sob o ponto de vista de Sistema Digitais, os Registradores são circuitos sequenciais formados por Flip-Flops (FF) que, sob o ponto de vista de Organização de Computadores, compõem o nível mais alto da hierarquia de memória. São memórias voláteis (ou seja, seus dados são perdidos quando o circuito é desligado), 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  $R_{M-1}$ .

Figura 1 - Processador RISC-V multiciclo



Fonte: Digital Design and Computer Architecture - RISC-V Edition

<https://doi.org/10.1016/C2019-0-00213-0>

Outro registrador especial é o PC (*Program Counter*), que é utilizado para indicar, na memória, o endereço da próxima instrução a ser executada. A cada instrução executada, ele é atualizado para apontar para o endereço da próxima instrução a ser executada, que pode estar na próxima posição de memória, ou em outra posição de memória gerada por uma instrução de um desvio de fluxo. Dependendo da arquitetura, o PC também pode ser implementado por meio de um contador com carga paralela de dados. Contadores são circuitos sequenciais que geram uma sequência pré-definida de valores e, no caso de contadores com carga paralela, eles podem ser atualizados com um valor diferente do próximo valor da sequência. Os contadores podem ser síncronos ou assíncronos.

Desta forma, o PC pode ser implementado tanto como um registrador, acrescido de um circuito dedicado formado por um circuito somador (para incremento de 1) e por um multiplexador (para o desvio de fluxo), ou com um contador (síncrono ou assíncrono) com carga paralela. Neste caso, cada grupo apresentará as duas implementações, indicando vantagens e desvantagens de cada implementação, justificando a escolha entre as duas possibilidades.

Existe ainda um banco de registradores que reunirá M registradores de uso geral. Estes registradores armazenam dados de N bits. Como primeiro problema, o aluno deverá desenvolver um registrador de N 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 M 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).

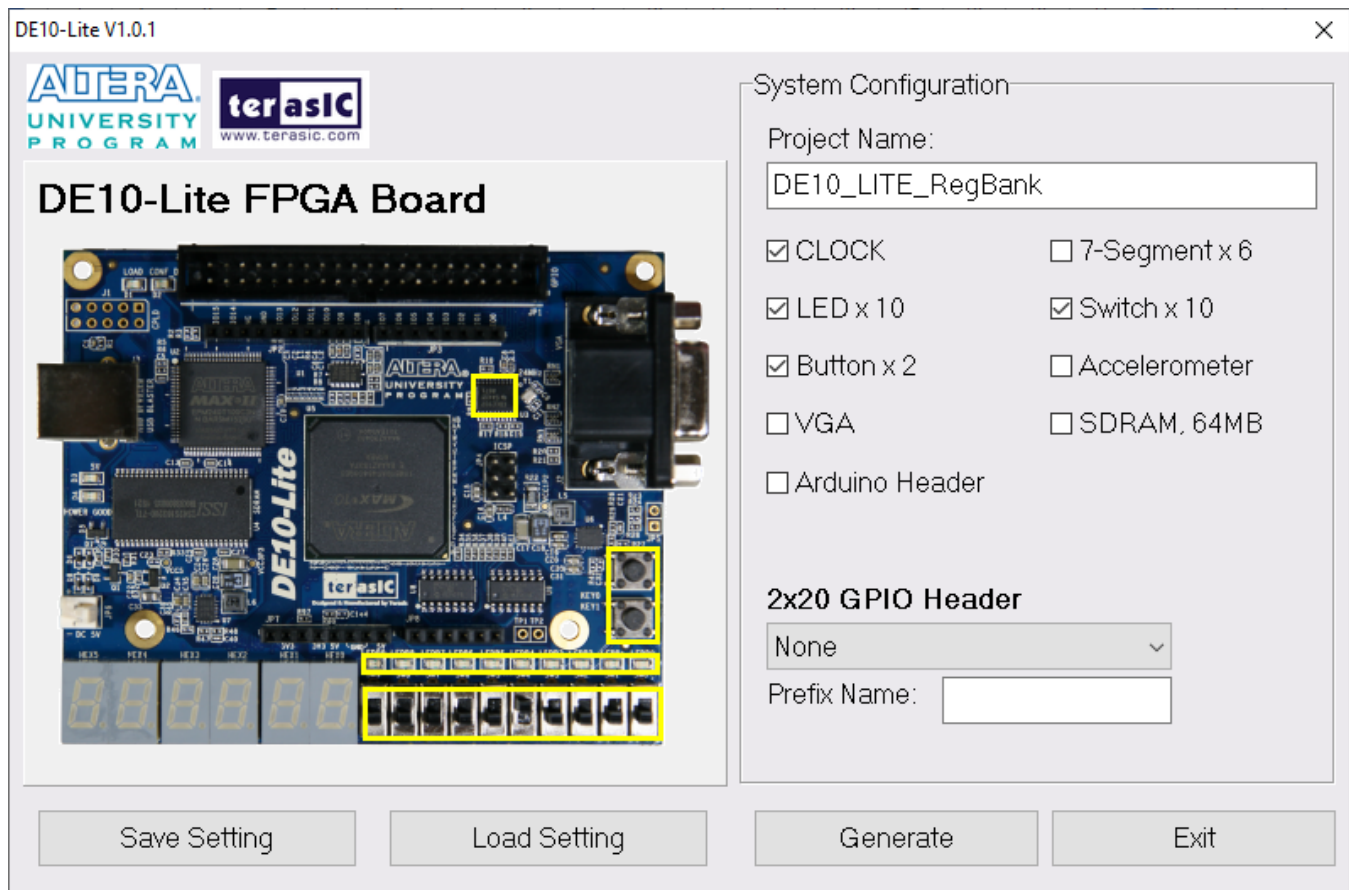
Utilize Multiplexadores e decodificadores para fazer a seleção dos registradores. Quando WE3 é ativado (igual a '1'), o valor presente em WD3 é armazenado no registrador indicado por A3, após a transição do sinal de CLK. O valor dos registradores indicados por A1 e A2 são apresentados nas saídas RD1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

O relatório deve descrever, de forma sucinta, conceitos de registradores, incluindo paralelos e seriais, implementação de circuitos registradores paralelos com Clock Enable, conceitos contadores síncronos e assíncronos, com Clock Enable e Parallel Load, esquemático dos registradores, individualmente, esquemático final do PC implementado.

### Procedimento Experimental:

Apresentar a implementação de um registrador genérico de N 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 Verilog.

Figura 2 - Configuração do DE10\_Lite\_SystemBuilder.exe



Fonte: o autor

Criar uma pasta denominada `DE10_LITE_RegBank`, com as subpastas `docs`, `modelsim`, `quartus` e `src`, e, utilizando o `DE10_Lite_SystemBuilder.exe`, criar um projeto na pasta `quartus`, também denominado `DE10_LITE_RegBank`, ativando apenas o clock, os chaves, os push-buttons e os LEDs, conforme apresentado na figura 2.

Abrir o projeto no Quartus Lite e implementar o Banco de Registradores utilizando Verilog, denominado `RegBank.v`, armazenando o código na pasta `src`. Incorporar o código do Banco de Registradores ao projeto principal (`DE10_LITE_RegBank`), com `WIDTH = 4` e `SIZE = 2`, ligando as chaves `SW[1:0]`, `SW[3:2]` e `SW[5:4]` às entradas `A1`, `A2` e `A3`, os push-buttons `KEY[0]` e `KEY[1]` às entradas `CLK` e `WE3`, as chaves `SW[9:6]` às entradas `WD3`, as saídas `RD1` e `RD2` aos `LEDR[3:0]` e `LEDR[7:4]`, e executar o projeto no kit `DE10_LITE`.

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

Exemplo de projeto parametrizável:

```
module RegBank
#(
    parameter WIDTH = 4,
               SIZE = 3)
(
    input CLK,
    input WE3,
    input [SIZE-1:0] A1,
    input [SIZE-1:0] A2,
    input [SIZE-1:0] A3,
    input [WIDTH-1:0] WD3,
    output [WIDTH-1:0] RD1,
    output [WIDTH-1:0] RD2
);
    // your code goes here...
endmodule
```

Exemplo de solicitação de projeto parametrizável:

```
// Instantiate RegBank design and connect with Testbench variables
RegBank #(
    .WIDTH (4), .SIZE(2))
RB0 ( .CLK (KEY[0]),
      .WE3 (KEY[1]),
      .A1 (SW[1:0]),
      .A2 (SW[3:2]),
      .A3 (SW[5:4]),
      .WD3 (SW[9:6]),
      .RD1 (LEDR[3:0])
      .RD2 (LEDR[7:4]));
```