

Departamento de Engenharia Elétrica e de Computação

SEL0384 – Laboratório de Sistemas Digitais I

Prof. Dr. Maximilian Luppe

PRÁTICA Nº6

Aprendizado baseado em Projeto (PBL)

PBL01 - Circuitos Combinacionais - ALU

Problema:

Implementar uma ALU (*Arithmetic Logic Unit*) de N bits utilizando linguagem de descrição de hardware VHDL em projeto hierárquico e parametrizável

Equipamentos necessários:

- Kit Mercurio® IV

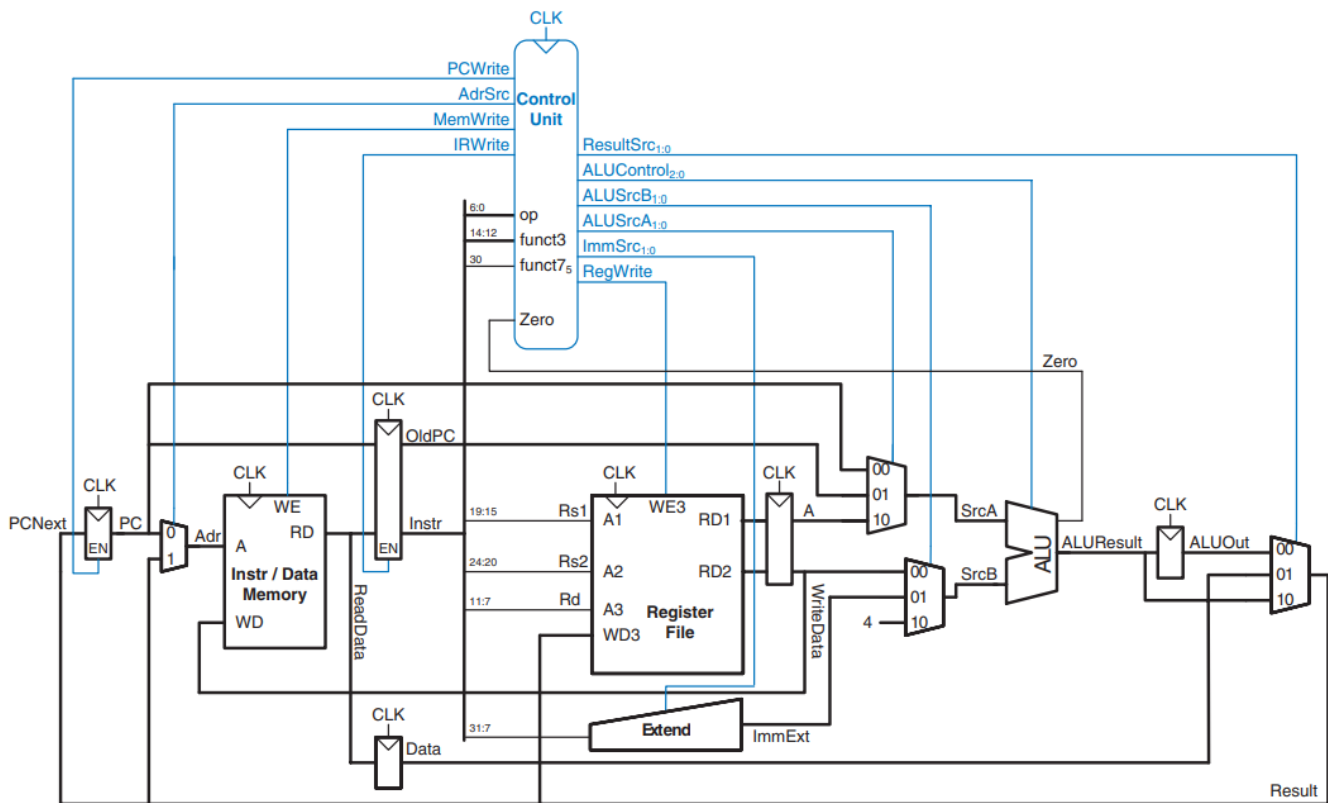
Introdução:

Valendo--se da natureza prática da disciplina SEL0384 – Laboratório de Sistemas Digitais I e visando proporcionar aos alunos uma experiência mais ativa e dinâmica nas atividades de laboratório, será adotada a metodologia de aprendizado ativo baseado em desenvolvimento de projeto (PBL - *Project-Based Learning*) ao longo das próximas práticas.

O PBL consiste em apresentar ao aluno um problema como não resolvido, de modo que este busque as informações necessárias para o desenvolvimento de um projeto que atenda à solução do problema. O aluno poderá utilizar todo o conhecimento adquirido ao longo das demais disciplinas relacionadas (Sistemas Digitais, Organização e Arquitetura de Computadores, etc.), colocando-os em prática. Para a disciplina SEL0384, o projeto central será a construção de um processador simples, baseado no RISC-V, na versão multiciclo.

O RISC-V é um conjunto de instruções (ISA) baseado em princípios RISC (Reduced Instruction Set Computing), desenvolvido desde 2010, na Universidade da Califórnia, em Berkeley. Um esquemático simples do caminho de dados para a versão multiciclo pode ser visto na figura 1. Nesta primeira parte do curso, estaremos interessados apenas na ALU - *Arithmetic-Logic Unit*. A ALU, assim como o banco de registradores e a unidade de controle, é um dos componentes centrais do processador. Sua função é realizar operações aritméticas (soma, subtração, etc.) e lógicas (AND, OR, etc.) entre dois operandos. A largura N do barramento de dados da ALU é, em algumas classificações, utilizado para discriminar o tipo de arquitetura: 8 bits, 16 bits, 32 bits, etc

Figura 1 - Processador RISC-V multiciclo



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

Sob o ponto de vista de Sistemas Digitais, a ALU é um circuito combinacional que realiza operações aritméticas (principalmente soma e subtração) e operações lógicas (AND, OR, NOT) entre dois valores de N bits, resultando em outro valor de N bits. Eventualmente a ALU pode gerar alguns sinais

adicionais, como *flags* de Zero (quando o resultado é igual a 0), de Negativo (quando o resultado é menor do que 0, considerando complemento de 2), de Carry, de Overflow etc. Operações mais complexas, como multiplicação, divisão, deslocamento, rotação e operações em ponto flutuante também são possíveis de serem implementadas.

O desenvolvimento da ALU se dará em duas partes. Na primeira parte será feito um levantamento da estrutura de uma ALU levando em consideração o uso de blocos lógicos básicos para a sua implementação (portas lógicas, somadores completos, mux21, etc.), subindo até níveis hierárquicos mais altos (circuito somador de N bits, comparador de N bits, multiplexador de N bits, etc.). Na segunda parte, a mesma será implementada em VHDL, utilizando os componentes selecionados na etapa anterior, com a apresentação do circuito implementado e seu devido funcionamento em relatório (teoria) e em fotos (funcionamento).

Para a primeira parte, deverá ser levada em consideração a seguinte tabela de funcionamento:

Tabela 1 - Conjunto de operações da ALU

ALUControl _{2:0}	Função
000	A + B
001	A - B
010	A and B
011	A or B
100	-
101	SLT
110	-
111	-

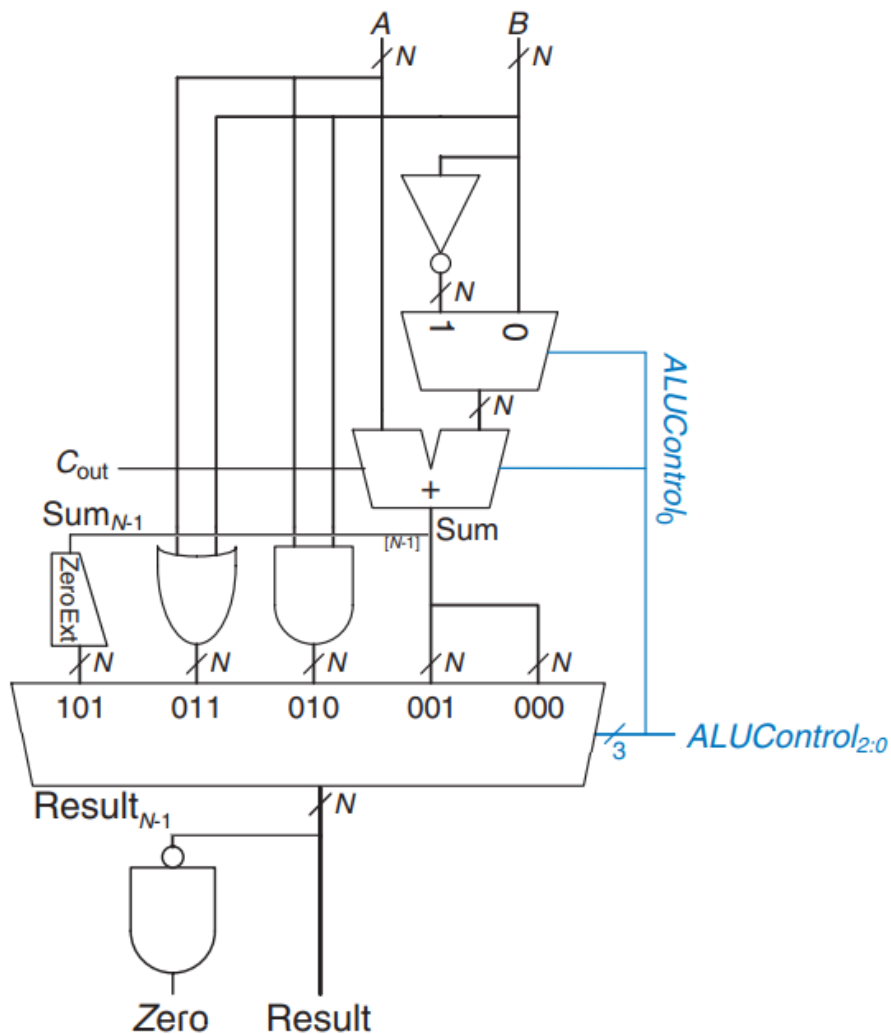
Onde ALUControl_{2:0} são três sinais de controle que selecionam a operação que a ALU realizará, A e B são os dois operando de entrada, e SLT significa *Set if Less Than*, ou seja, verifica se A é menor do que B, de acordo com a Tabela 2.

Tabela 2 - Resultado da operação SLT para $N = 4$

SLT	
$A \geq B$	$A < B$
0000	0001

Na figura 2 é apresentado o esquemático em alto nível de uma possível implementação de uma ALU de N bits. Além do resultado descrito na Tabela 1, a ALU deverá ter como saída uma *flag* de Zero.

Figura 2 - ALU de N bits com suporte para SLT e *flag* de Zero



O relatório deve descrever, de forma sucinta, o funcionamento de um somador e de um multiplexador: teoria, soma em complemento de 2, equações booleanas, estrutura lógica, implementação hierárquica, etc., e de um multiplexador: teoria, equações booleanas, estrutura lógica, implementação hierárquica, etc., e como estes componentes foram utilizados para implementar a ALU.

Procedimento Experimental:

Apresentar a implementação de uma ALU baseada na arquitetura RISC-V, com barramento de dados $N = 4$ bits, utilizando a linguagem de descrição de hardware VHDL.

Criar uma pasta denominada `MercurioIV_ALU`, com as subpastas `docs`, `modelsim`, `quartus` e `src`, e, utilizando o Quartus Lite Edition, criar um projeto a pasta `quartus`, também denominado `MercurioIV_ALU`.

Realizar a implementação de uma unidade lógico-aritmética (ULA) modular utilizando a linguagem de descrição de hardware VHDL, denominado `alu.vhd`.

Verifique o funcionamento da ULA no kit Mercurio® IV incluindo o arquivo `MercurioIV_ALU.vhd` no projeto `MercurioIV_ALU`, definindo este arquivo como “Top-Level Entity”, incorporando o código da ULA (`alu.vhd`) ao projeto principal (`MercurioIV_ALU`), as chaves `SW[2:0]` às entradas `ALUControl[2:0]`, a saída `Result` ao display de 7 segmentos por meio do decodificador binário para 7 segmentos, e a saída `Zero` ao `LED_R`, 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.