UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

SEL0384 – Laboratório de Sistemas Digitais I

Prof. Dr. Maximiliam Luppe

Bárbara Fernandes Madera - nº: 11915032

Johnny Caselato Guimarães - nº: 11915481

PRÁTICA Nº6
Aprendizado baseado em Projeto (PBL) PBL 01
Circuitos Combinacionais - ALU

SÃO CARLOS

2023

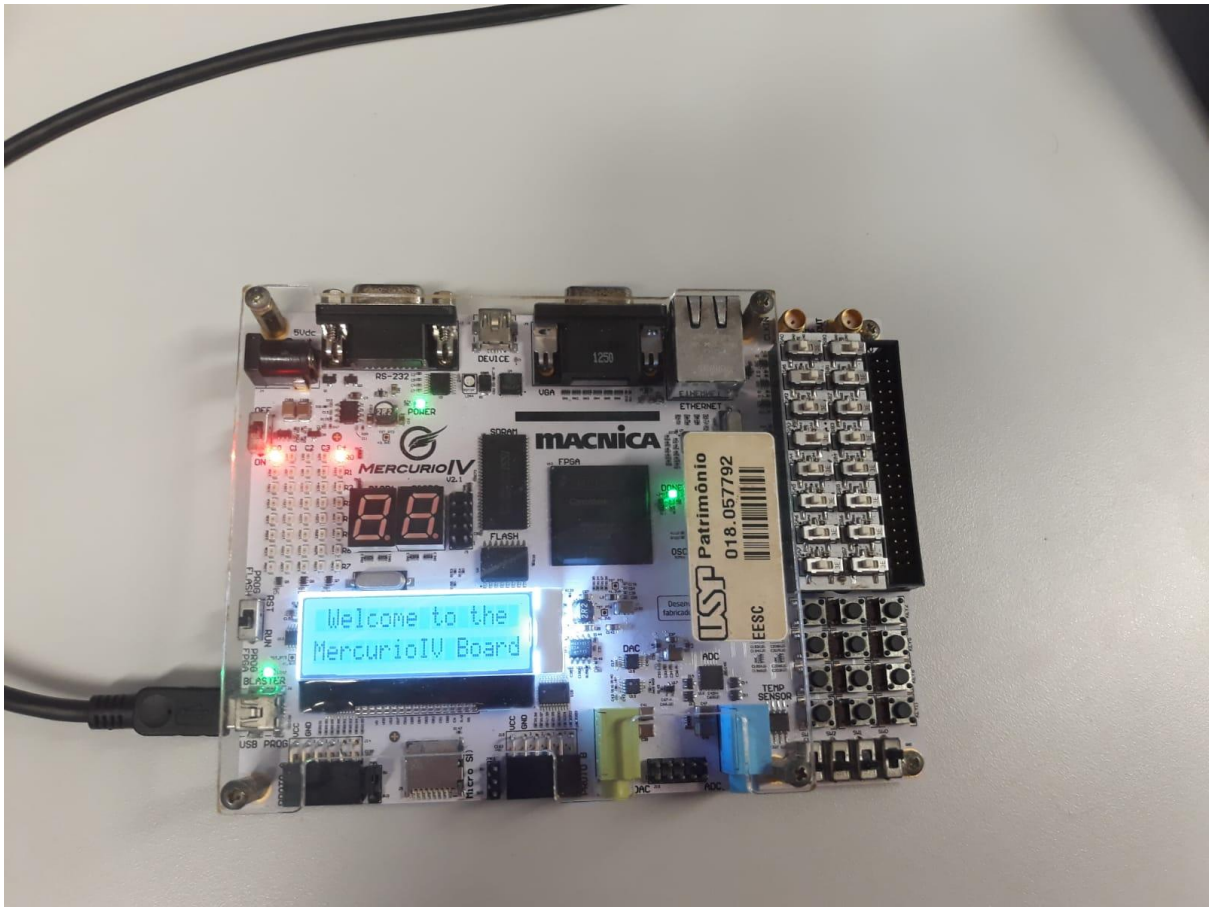
1.Objetivos:

O objetivo deste projeto é criar uma ALU parametrizável que seja capaz de realizar operações aritméticas (soma, subtração) e operações lógicas (AND, OR) entre dois operandos de N bits. O barramento de dados da ALU é parametrizável para diferentes larguras (N bits). Além disso, a ALU deve ser capaz de gerar uma flag de Zero.

2.Equipamentos Necessários para Prática

- Kit Mercurio® IV
- Software Quartus II Web Edition

Figura 1- Kit Mercurio® IV com implementação do ALU bem sucedida.



3.Introdução:

Este relatório descreve a implementação de uma Arithmetic Logic Unit (ALU) de N bits usando a linguagem de descrição de hardware VHDL. Este projeto faz parte da disciplina SEL0384 - Laboratório de Sistemas Digitais I e segue a metodologia de aprendizado ativo baseado em desenvolvimento de projeto (PBL - Project-Based Learning). Dessa forma, para implementar a ALU, foram utilizados componentes lógicos básicos, como portas lógicas, somadores completos e multiplexadores. A ALU foi dividida em blocos hierárquicos, incluindo um circuito somador de N bits, um comparador de N bits, um multiplexador de N bits e outros componentes necessários para suportar as operações listadas na Tabela 1.

O Somador de N bits, por sua vez, é um componente central da ALU, responsável por realizar operações de adição e subtração. Ele opera em complemento de 2, garantindo que as operações de subtração sejam tratadas corretamente. O somador de N bits é implementado utilizando somadores completos de 1 bit e é expandido para N bits. A adição binária é feita considerando os bits individuais dos operandos e um bit de carry-in. O complemento de 2 é usado para representar números negativos, facilitando a subtração. O somador,então, utiliza equações booleanas para calcular as somas e os carry-outs em cada posição binária. Geralmente, um somador é construído usando portas lógicas (como XOR, AND, OR) para calcular somas e carry-outs em cada posição binária.Somadores completos podem ser construídos hierarquicamente a partir de somadores menores, como somadores de meio e somadores completos, para adicionar números de várias palavras.

O multiplexador, por sua vez, é um dispositivo que seleciona uma das várias entradas com base em um sinal de controle. Na ALU, os multiplexadores são usados para escolher entre diferentes operações ou resultados. Um multiplexador tem várias entradas e uma única saída. O sinal de controle determina qual entrada é passada para a saída. As equações booleanas do multiplexador, por conseguinte, são simples e dependem do número de entradas. Dessa forma,um multiplexador pode ser construído com portas lógicas, mas é frequentemente implementado como uma única unidade lógica tal que os multiplexadores podem ser agrupados em cascata para criar seleções mais complexas.

Com isso, a ALU foi implementada em VHDL, seguindo a estrutura hierárquica definida. O código VHDL está organizado em vários módulos, incluindo o somador de N bits, o multiplexador de N bits e outros componentes auxiliares. A saída da ALU é gerada de acordo com as equações booleanas e os sinais de controle.

4.Resultados

A implementação da ALU foi testada no Kit Mercurio® IV. O arquivo MercurioIV_ALU.vhd foi configurado como a entidade principal (Top-Level Entity) do projeto e incorporou o código da ALU (alu.vhd). As chaves SW[2:0] foram usadas para controlar as operações da ALU, a saída Result foi exibida no display de 7 segmentos por meio do decodificador binário para 7 segmentos, e a saída Zero foi exibida no LED_R. No total, para a implementação do sistema, foram utilizadas 38 células lógicas após a compilação.

Figura 2 - Resumo dos processos e recursos utilizados após a compilação.

Flow Summary	
Flow Status	Successful - Tue Oct 31 23:14:51 2023
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Web Edition
Revision Name	MercurioIV_ALU
Top-level Entity Name	MercurioIV_Golden_Top
Family	Cyclone IV E
Device	EP4CE30F23C7
Timing Models	Final
Total logic elements	38 / 28,848 (< 1 %)
Total combinational functions	38 / 28,848 (< 1 %)
Dedicated logic registers	0 / 28,848 (0 %)
Total registers	0
Total pins	29 / 329 (9 %)
Total virtual pins	0
Total memory bits	0 / 608,256 (0 %)
Embedded Multiplier 9-bit elements	0 / 132 (0 %)
Total PLLs	0 / 4 (0 %)

Neste sentido, os códigos implementados em VHDL descrevem entidades e arquiteturas para implementar uma lógica digital em um ambiente de design de hardware de formas distintas. São eles:

1. **'adder':** Este código define uma entidade 'adder' que representa um somador de 4 bits. A arquitetura 'hierarquia' instancia quatro somadores completos

('fadd') para realizar a adição bit a bit dos vetores de entrada 'A' e 'B'. O resultado é armazenado no vetor 'RESULT'.

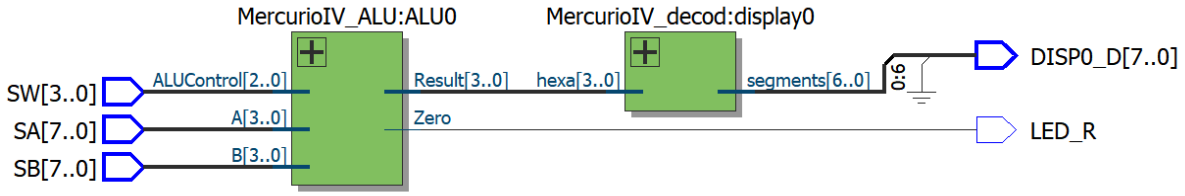
2. **'subtractor':** Este código define um subtrator, o qual é estruturado da mesma forma que um somador, possuindo 4 componentes ‘fadd’ mas, operando com a lógica de complemento de 1 para realização de operações de subtração.

3. **'fadd':** Esse código define um somador completo de um bit ('full adder'). A arquitetura 'concorrente' implementa as saídas 's' (soma) e 'co' (carry out) com base nas entradas 'a', 'b' e 'ci' usando operações lógicas.

4. **'MercurioIV_Golden_Top':** Este componente de nível superior instancia subcomponentes, como 'MercurioIV_ALU' e 'MercurioIV_decod'. A ALU recebe operandos de 8 bits, tal que realiza operações baseadas em um controle de 3 bits e produz um resultado de 4 bits. O decodificador converte esse resultado em segmentos para exibição em um display de sete segmentos. A interconexão de sinais permite a comunicação entre esses componentes para operações mais complexas e exibição de resultados.

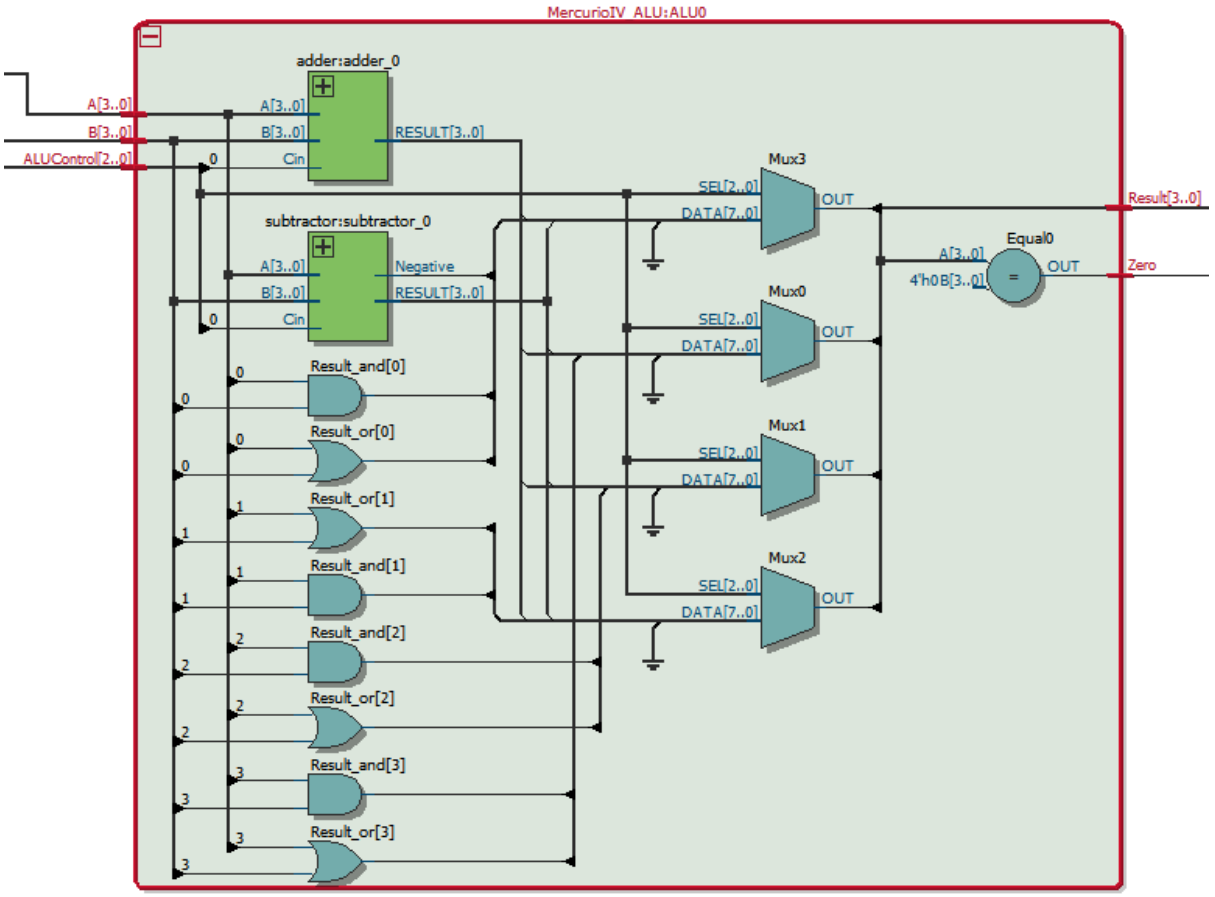
A composição destes elementos é representada na figura 3 com a visualização em RTL.

Figura 3 -Visualização RTL geral.



E, mais a fundo, podemos expandir a visualização para compreender o interior da ALU implementada, onde são observados os componentes e operações lógicas, elementos de seleção, comparação, soma e subtração.

Figura 4 - Visualização RTL interna da ALU implementada.



Em resumo, os códigos apresentam diferentes níveis de abstração em design de hardware, desde somadores de bits individuais até um componente de maior complexidade- o qual instancia componentes menores para realizar operações lógico-aritméticas e exibir resultados em displays. Cada código representa um elemento importante na construção e funcionamento da Unidade Lógica Aritmética(ALU).

5.Conclusão:

Neste projeto, foi desenvolvida uma ALU de N bits parametrizável em VHDL, capaz de realizar uma variedade de operações aritméticas e lógicas. A implementação hierárquica permitiu uma organização eficiente do código e facilitou a compreensão do funcionamento da ALU. A ALU foi testada com sucesso no Kit Mercurio® IV, demonstrando seu funcionamento conforme as especificações.