# **UNIVERSIDADE DE SÃO PAULO**

ESCOLA DE ENGENHARIA DE SÃO CARLOS

# **SEL0384 - Laboratório de Sistemas Digitais 1**

Prof. Maximiliam Luppe

Bárbara Fernandes Madera - nº: 11915032

Johnny Caselato Guimarães - nº: 11915481

**Prática 3 - Relatório de Prática de Laboratório: Multiplexador**

**SÃO CARLOS**

**2023**

1. **Objetivos**

Esta atividade prática tem como principal objetivo a implementação e síntese de circuitos combinacionais, neste caso, um circuito multiplexador 2 para 1, em dispositivos reconfiguráveis através da linguagem de descrição de hardware VHDL no Kit Mercurio® IV disponibilizado. Além disso, a implementação deverá incluir o uso da estrutura de comando concorrente *WHEN-ELSE*, na qual uma série de expressões e condições são avaliadas, pela ordem em que se encontram, a fim de direcionar um sinal de entrada específico à uma saída, ou seja, realizar o processo de seleção do multiplexador.

1. **Equipamentos Necessários para Prática**

* Kit Mercurio® IV
* Software Quartus II Web Edition

1. **Procedimento Experimental**
   1. **Equações booleanas e circuitos TTL**

**Multiplexador de 2 para 1:**

O Multiplexador de 2 para 1 possui duas entradas de seleção (S0 e S1) e duas entradas de dados (I0 e I1). A saída (Z) é selecionada com base nas entradas de seleção da seguinte forma:

Z = (S1 AND I0) OR (NOT S1 AND I1).

Com isso,são duas entradas de seleção (S1 e NOT S1);duas entradas de dados (I0 e I1);duas portas lógicas AND para calcular (S1 AND I0) e (NOT S1 AND I1) e uma porta lógica OR para calcular a saída final Z.Logo, são necessárias 2 portas AND, 1 porta OR e 1 porta NOT para implementar o Multiplexador de 2 para 1 usando circuitos integrados da família TTL7400.

**Multiplexador de 3 para 1:**

O Multiplexador de 3 para 1 possui três entradas de seleção (S0, S1 e S2) e três entradas de dados (I0, I1 e I2). A saída (Z) é selecionada com base nas entradas de seleção da seguinte forma:

Z = (S2 AND S1 AND I0) OR (S2 AND NOT S1 AND I1) OR (NOT S2 AND S0 AND I2).

Dessa forma,são utilizadas três entradas de seleção (S0, S1, S2), que também requerem portas NOT; três entradas de dados (I0, I1, I2);portas lógicas AND para calcular as combinações de seleção;portas lógicas OR para calcular as saídas intermediárias e uma última porta OR para calcular a saída final Z.Portanto, são necessários 3 portas AND, 3 portas OR e 3 portas NOT para implementar o Multiplexador de 3 para 1 usando circuitos integrados da família TTL7400.

Lembrando que os circuitos integrados da família TTL7400 geralmente contêm múltiplas portas lógicas em um único chip, como as portas AND, OR e NOT. Assim, foram selecionados os componentes específicos que atenderam às suas necessidades da prática e, em seguida, conectados os pinos de entrada e saída de acordo com as equações booleanas. O número exato de chips dependeu da disponibilidade e da escolha específica dos chips TTL7400 para o projeto.

* 1. **Implementação de um multiplexador 2 para 1**

Conforme as orientações, foram criados os diretórios para o projeto denominado MercurioIV\_mux. A implementação do multiplexador foi feita com dois arquivos, sendo o primeiro *top level* onde uma instância do componente MUX era criada, a qual, por sua vez, tinha a lógica descrita no segundo arquivo “mux21”:

**--Projeto multiplexador 2 para 1**

**--SEL0384 - Atividade 4**

**--Autores:**

**--Johnny Caselato Guimaraes - N: 11915481**

**--Barbara Fernandes Madera - N: 11915032**

**--Prof.: Maxmillian Lupe**

**ENTITY mux21 is**

**PORT(**

**I0, I1 : in bit\_vector(3 downto 0);**

**s : in bit;**

**Z : out bit\_vector(3 downto 0)**

**);**

**END mux21;**

**ARCHITECTURE concorrente of mux21 is**

**BEGIN**

**Z <= I0 WHEN S = '0' ELSE**

**I1;**

**END;**

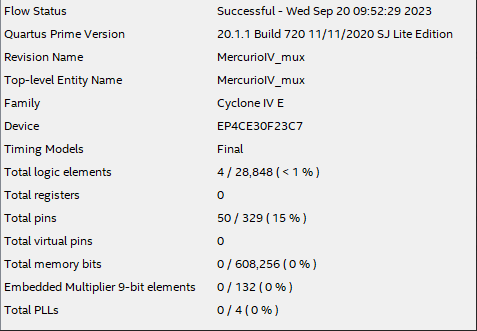
Vemos então que, na arquitetura da entidade do multiplexador, foi utilizado o comando concorrente proposto *WHEN-ELSE*, o qual direciona à saída Z (associada a 4 LEDs na matriz) uma das entradas IO ou I1, de acordo com a condicional atendida, neste caso se s = ‘0’ a saída replica o sinal presente em I0 (associada as chaves SA[3~0]), enquanto que, para os outros casos (que é apenas s = ‘1’) é replicada a entrada I1 (associada as chaves SB[3~0]).

A lógica também foi implementada utilizando outro comando concorrente *WITH-SELECT*, no qual é necessária a declaração de todas as condições possíveis, o que, para esta aplicação, não é muito diferente.

1. **Resultados obtidos**

Após a implementação do código, apresentado na seção anterior, a compilação foi bem sucedida, resultando, conforme a figura 1, no uso de apenas 4 células lógicas para a aplicação do sistema proposto.

Figura 1 - Relatório de compilação do projeto.



Adiante, através da visualização RTL exposta nas figuras 2 e 3, vemos a implementação das conexões de entrada e saída do hardware no fluxo da lógica que foi desenvolvida na arquitetura da entidade “mux21”. Especialmente na figura 3, vemos explicitamente a alocação de uma unidade multiplexadora associada às chaves para a seleção e interruptores como entradas de dados (4 bits por entrada), localizados em SW e GPIO0\_D respectivamente.

Figura 2 - Visualização RTL do sistema.

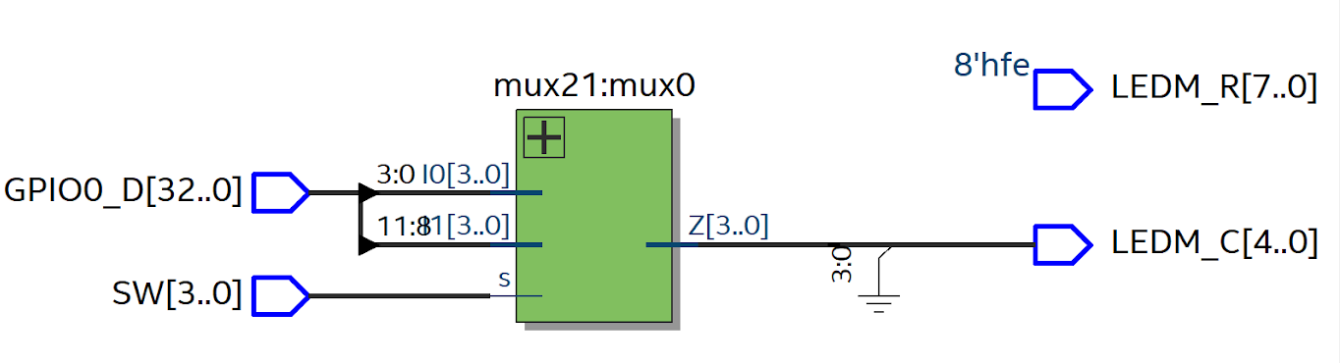
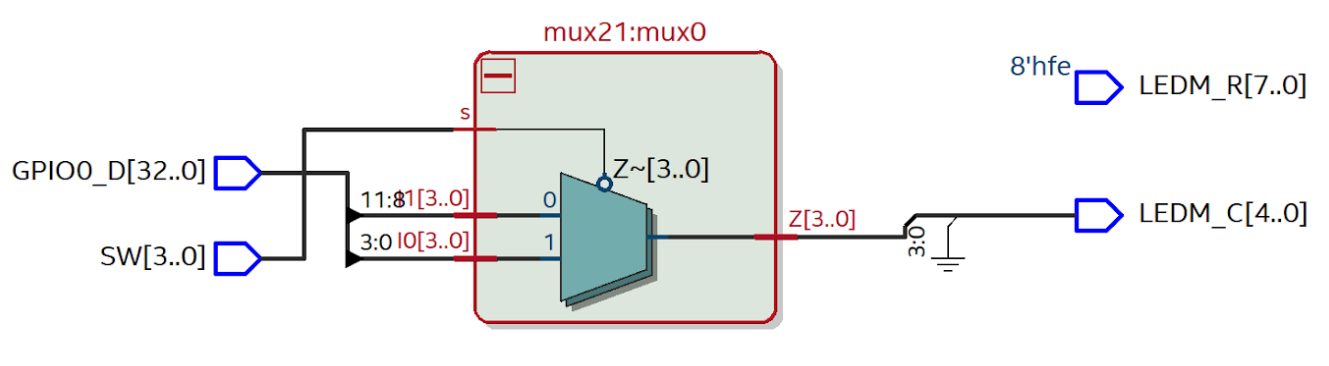
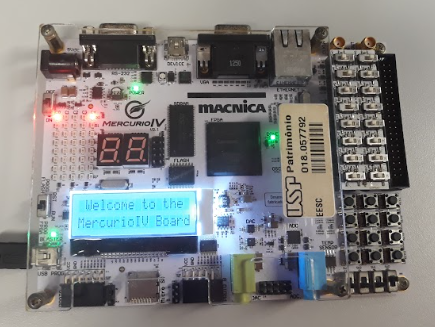


Figura 3 - Visualização explícita da implementação do MUX.



Finalmente, após a avaliação positiva da lógica implementada, o programa foi carregado à placa do kit de desenvolvimento presente na figura 4. Operante e conforme o previsto.

Figura 4 - Kit Mercurio® IV com implementação do MUX bem sucedida.



**5.Conclusão**

Por conseguinte, os resultados da implementação foram favoráveis, com apenas 4 células lógicas sendo utilizadas no FPGA, conforme mostrado no relatório de compilação. A visualização RTL revelou as conexões de entrada e saída do hardware, demonstrando a alocação das entradas de dados e seleção nas chaves e interruptores do Kit Mercurio® IV.

No final, o programa foi carregado com sucesso na placa do kit de desenvolvimento, demonstrando a operação conforme o esperado.

Portanto, esta prática de laboratório atingiu seu objetivo ao permitir que os alunos aprendessem e aplicassem conceitos de VHDL e circuitos combinacionais na implementação de um multiplexador, fornecendo uma experiência prática valiosa na programação e uso de dispositivos reconfiguráveis.