Documentação Laboratorio de VHDL

Todos os arquivos RTL dos componentes estão em suas respectivas pastas.

**Flip Flop tipo D e JK**

**Multiplexador de quatro entradas**

Componente que possui 4 entradas para dados, uma para um seletor, e uma saída que é decidida pelo seletor. Há 2 versões de Multiplexador com 4 entradas.

A primeira foi feita usando como base 3 multiplexadores de 2 entradas. Funciona da seguinte maneira: entram 4 entradas, que são separadas para 2 multiplexadores de 2 entradas, ou seja, se tivermos A, B, C e D, A e B serão entradas de um multiplexador enquanto C e D serão a entrada para o outro multiplexador. As saídas desses multiplexadores são então usadas como entradas no terceiro e último multiplexador. Teremos então duas entradas que decidiram a saída dos multiplexadores: S0 e S1, onde S0 estará decidindo a saída dos 2 multiplexadores iniciais e S1 decidirá a saída do último multiplexador.

A segunda versão funciona através de IF-ELSE-THEN para decidir a saída, baseada em uma entrada que é um vetor de 2 bits. Então teremos as entradas A, B, C e D e o vetor de bits Selector. Dependendo do Selector, a saída S do multiplexador será uma das 4 opções de entrada.

**Porta lógica XOR utilizando portas AND, NOT e OR**

A porta XOR foi construída utilizando as portas AND, NOT e OR, tendo como entradas A e B , e saída a seguinte operação (feita com as portas mencionadas anteriormente): Saída <= [ (NOT{A} AND B) OR (A AND NOT{B} ) ].

**Somador de inteiro com +4**

O somador recebe 2 entradas inteiras, sendo que a segunda será obrigatoriamente o valor 4, e tem como saída a soma dessas entradas.

**Memória ROM de 16 bits**

Memória que guarda o programa a ser executado. Possuí como entrada vetores que seriam o programa em 0s e 1s, e um inteiro que irá dizer qual linha de vetores será enviada para saída.

**Memória RAM de 16 bits**

Memória utilizada para guardar dados temporariamente em um vetor de endereços de memória. Possui uma entrada de 16 bits para dados a serem armazenados, uma entrada que será o endereço onde os dados serão armazenados ou de onde serão recuperados, uma entrada para o clock do sistema e uma para a flag de escrita na memória, e por fim temos a saída de 16 bits, que é o dado que eestá sendo recuperado da memória.

Funciona baseado no clock no processo de escrita, onde se o clock estiver na borda alta e a flag de escrita estiver ativa, será escrita no endereço de memória os dados de entrada. E de saída, envia os dados armazenados no endereço de memória dado como uma das entradas.

**Banco de Registradores de 16 bits**

Componente em que estão localizados os registradores do sistema. Tem como entrada dois endereços a serem lidos, os quais são utilizados para recuperar os valores que serão enviados para os registradores de saída A e B; um dado de entrada de 16 bits e o endereço do registrador de destino onde este dado será escrito. O componente recebe ainda um sinal vindo da Unidade de Controle que irá decidir ser o dado será escrito ou não no registrador de destino.

**Somador de 16 bits**

Componente recebe como entrada 2 (dois) vetores lógicos e realiza a soma destes, para então enviar como saída. Há duas versões do somador de 16 bits.

A primeira utiliza 8 componentes que são somadores de 2 bits, os quais por sua vez usam 2 componentes que são somadores de 1 bit. A estrutura de cada somador possui as entradas A e B que serão somadas, e através de operações lógicas utilizando . Funciona dividindo os bits da entrada de 16 bits e encaminhando-os como entrada para estes somadores, que além de receberem os bits a serem somados, também recebem o carry-in como entrada. Na saída temos o resultado da soma de cada somador que é concatenada ao resultado final, além do carry-out.

A segunda versão é mais simples: A e B são dois vetores de entrada com 16 bits e C é a saída de 16 bits que é soma dos vetores A eB.

**Unidade de Controle do MIPS uniciclo 16 bits**

Componente que, dependendo de sua entrada, irá enviar sinais de entrada para os demais componentes. Ele recebe como entrada apenas um vetor de sinais lógicos chamado [OPCode], que no caso do nosso processador é de 4 bits. Baseado no OPCode, este componente enviará como saída os sinais que irão ser usados como entrada nos demais componentes do sistema.

**ULA de 16 bits, utilizando port map, com as seguintes operações:**

**AND, OR, NOT, NOR, NAND, XOR, SHIFT de 2 bits à esquerda, SHIFT de bits à direita, soma e subtração.**

Componente que executa as operações principais do sistema. Possuí como entrada dois vetores de 16 bits para realizar as operações, um vetor de 4 bits para decidir qual será a saída, e a saída em si.

Dentro da ULA existem os respectivos componentes para as operações: AND, OR, NOR, XOR, SHIFT de 2 bits para Esquerda ou Direita, Soma e Subtração. Para as operações lógicas foram usados como entrada os bits mais significativos dos registradores A e B de entrada, enquanto que para as operações aritméticas foram usados os vetores de 16 bits desses mesmos registradores.

Por fim, o resultado de todas essas operações entram em um multiplexador que, baseado no sinal de 4 bits de entrada vindos da Unidade de Controle, decidirá qual das operações vai para a saída do componente.

**Extensor de Sinal de 8 para 16 bits**

Componente que aumenta o número de bits do vetor de entrada, para que não haja problemas de compatibilidade com os demais componentes que utilizarão este dado. Tem como entrada um vetor de 8 bits e como saída o mesmo vetor, mas com 16 bits sem que haja qualquer alteração no seu significado.