

Questão 1

a entrada `data_in`, e o clock como entrada, a saída (`data_out`) só vai receber quando o ciclo de clock mudar, na transição do clock, quando o clock mudar de '0' para '1'.

Questão 2

o multiplexador tem 4 entradas A, B, C, D um vetor SEL e a saída S, ele funcionará baseado neste vetor, com negado e não negado, por exemplo, quando o `sel(0)` for negado (`NOT sel(0)`) e o `sel(1)` não for, ou seja, `sel = 01`, o resultado será 'B' e assim por diante

Questão 3

Duas variáveis de entrada, uma de saída, a de saída recebe `A XOR B`, que é justamente a porta XOR, apenas isso

Questão 4

o somador terá dois vetores de 4 posições, um de entrada e outro de saída, o processo de soma acontece da seguinte forma, o número 4 em binário é "0100", cada posição do vetor de entrada somará com um bit desse binário, o `A(0) + 0` e assim por diante, o resultado será armazenado em uma variável criada dentro da ARCHITECTURE, assim como um vetor criado dentro dela, depois será jogada para o vetor e a variável principal criada dentro da ENTITY.

Questão 5

Um vetor de entrada de 4 bits (`address`) e um vetor de saída de 8 bits (`data`) declaradas na entidade, o armazenamento nesta memória funciona na base do 'case', um vetor 'men' declarado dentro da arquitetura será os espaços para o armazenamento (espaços de 8 bits), e com o case será selecionado qual desses espaços deverá ser utilizado. Utilizando os seus multiplexadores.

Questão 6

duas "variáveis" para armazenar a quantidade de bits (bits e words) e o tamanho das palavras na memória, `data_in` e `data_out` para os vetores de entrada e saída direta do

código, e as estruturas de condição que só entram nelas se forem '1' a informação vinda, se forem TRUE.

Questão 8

É um somador completo,, pois ele soma 3 varia veis e a saída são duas. "cin", "a" e "b" são as entradas e "cout" e "s" são as saídas. "a", "b" e "s" como vetores para poder somar todos os 16 bits usando um range e a variável "i" para tal. Por isso precisa da variável "cout", porque como um somador de 16 bits é composto por 16 somadores de 1 bit, quando faz a soma de um e o bit mais significativo "vai" (é armazenado no "cout") ele vai para o proximo somador de 1 bit e assim sucessivamente.

Questão 9

Trabalha com dois vetores de 16 bits para a entrada, do tipo unsigned 'A' e 'B' e um para a saída 'SUM', o resultado simplesmente é a subtração de 'A' e 'B' com o resultado armazenado em 'SUM'.

Questão 10

Usa 3 variáveis de entrada, variável "selection" será o valor do case, ou seja, o valor desta variável escolherá o case e por consequência qual operação da ULA será executada sendo ela um vetor de 3 posições para fazer as 8 combinações 000,001....pois a ULA possui 8 operações (ou seja, completar todos os casos do "case"). As variáveis "A" e "B" serão as variáveis das operações. E a variável "O" receberá o valor desta operação.

Abaixo a Tabela Verdade deste funcionamento:

```
-- sel2 sel1 sel0 | O
-- -----
-- 0 0 0 | A + B
-- 0 0 1 | A - B
-- 0 1 0 | A and B
-- 0 1 1 | A or B
-- 1 0 0 | A xor B
-- 1 0 1 | not A
```

-- 1 1 0 | not B

-- 1 1 1 | Z

Questão 11

Ele declara dois vetores, "slv_8" e "slv_16_OUT", depois apenas faz a conversão de um para outro, usando uma "função" do Quartus que recebe respectivamente o tipo e o vetor e o tamanho do espaço alocado, muito parecido com a função Malloc presente no C.

Questão 13

Usa um vetor, do tipo INOUT para ser o contador, ele terá um espaço para 2 bits, sendo assim o contador irá 0 a 2, ele nunca chegará a 3 por causa das condições colocadas dentro dos comandos de decisão implementados no corpo da 'arquitetura'.

Referencias

<https://www.youtube.com/watch?v=XOuGgW-Qqpk>

<https://www.youtube.com/watch?v=wyiAtMWpCX8>

https://pt.wikibooks.org/wiki/VHDL/Flip-flop_d

<https://www.youtube.com/watch?v=qdxQv3barIE>

<http://docslide.com.br/documents/somador-1-4-8-16-32-e-64-bits-vhdl.html>

<https://www.youtube.com/watch?v=Zkkck2MovCc>

<https://www.youtube.com/watch?v=hMy3vJ8MPVQ&t=806s>

<http://stackoverflow.com/questions/17451492/how-to-convert-8-bits-to-16-bits-in-vhdl>

<http://www.forosdelweb.com/f14/vhdl-contador-sincrono-8-bits-886991/>

http://www.ic.unicamp.br/~cortes/mc602/slides/VHDL/VHDL_5_MC_Contadores_Regs_v1.pdf

file:///C:/Users/ufrr/Downloads/4-BookMIT_code_from_examples_Jul08425.pdf

<http://www.edaboard.com/thread38052.html>