QUESTÕES COLOCADAS PELOS ALUNOS ENTRE 28 DE NOVEMBRO E 5 DE DEZEMBRO RELATIVAMENTE AO TRABALHO 1

Questão 1

Nós já conseguimos uma simplificação que use 37 portas xor, ao invés das 64 do ponto de partida. Qual seria um número razoável de xors de uma boa simplificação para termo de comparação com a nossa?

Desculpem só agora lhes responder, mas tenho andado muito ocupado.
Uma simplificação que use apenas 37 portas xor para a implementação paralela do encoder é muito bom.
Qual é o máximo atraso de propagação, em termos de gates xor, obtido?

Questão 2

Tenho uma duvida, o encoder é suposto ter só 16 bits de entrada e o checker 24 bits de entrada certo?

Exactamente, se está a falar na versão paralela.
Na versão bit serial, só há uma entrada de dados (os bits são multiplexados no tempo).
Leia com atenção as notas que coloquei na sessão tutorial sobre o trabalho.

Questão 3

Para o encoder na entrada o professor prefere que usemos

d_in: std_logic_vector(15 downto 0);
ou

dIN_0,dIN_1,dIN_2,dIN_3 : IN STD_LOGIC;
dIN_4,dIN_5,dIN_6,dIN_7 : IN STD_LOGIC;
dIN_8,dIN_9,dIN_10,dIN_11 : IN STD_LOGIC;
dIN_12,dIN_13,dIN_14,dIN_15 : IN STD_LOGIC; *?*

Tanto faz. Escolham aquela que lhes seja mais conveniente.

Questão 4

Visto que segunda-feira não irá haver aulas e a entrega do Projeto é na quarta, quando serão as apresentações dos Projetos?

Estou a pensar calendarizá-las para as próximas 5.ª e 6.ª feira, dias 10 e 11 de Dezembro, mas ainda não é certo. Cada grupo terá 15m para apresentar e defender o seu trabalho.

Questão 5

Eu e o meu colega temos uma pequena duvida.
Nós ao tentar implementar uma solução mais eficiente para o projeto, deparamo-nos que usar um circuito com Xor's e usar o mesmo circuito mas com desmultiplexadores 1:2, temos o mesmo resultado, porém como o desmultiplexador tem menos gates lógicas do que o Xor, achamos que poderá ser uma das soluções para tornar mais eficiente o projeto, posteriormente também teremos que reduzir o número de desmultiplexadores.
Gostaríamos de saber se esta abordagem é plausível.

A função booleana de uma gate X-OR 2 é "`y = x1 AND NOT x2 OR NOT x1 AND x2`".
A função booleana de um desmultiplexador 1:2 é "`y1 = NOT s AND x` e `y2 = s AND x`".
Não percebo, assim, como diz que obtemos o mesmo resultado.
O pretendido para a versão paralela dos dispositivos é minimizar o número de gates X-OR 2 usadas na implementação, bem como o atraso de propagação global no pior caso.

Questão 6

Em anexo encontra-se uma imagem de uma possível implementação de uma simplificação para o trabalho prático. Gostaria de ter feedback para saber se posso seguir com esta implementação ou se está longe de uma versão mais otimizada, ou até mesmo se está errado?

O pretendido para a versão paralela dos dispositivos é minimizar o número de gates X-OR 2 usadas na implementação, bem como o atraso de propagação global no pior caso.
O ponto de partida para uma implementação "força bruta" da abordagem baseada na aplicação directa do

algorítmo da divisão é, como mostrei nas notas da sessão tutorial relativa ao trabalho, 72 gates X-OR 2 e um atraso de propagação global no pior caso de 10 atrasos de propagação de gates X-OR 2.
O ponto de partida para uma implementação "força bruta" da abordagem baseada na aplicação directa das propriedades do módulo (resto da divisão inteira) é, como mostrei nas notas da sessão tutorial relativa ao trabalho, 64 gates X-OR 2 e um atraso de propagação global no pior caso de 11 atrasos de propagação de gates X-OR 2.
O que posso dizer é o seguinte: é possível, a partir de uma das abordagens, baixar substancialmente quer o número de gates X-OR 2 usadas, quer o número de atrasos de propagação de gates X-OR 2 no pior caso. Para tal, tem que descobrir um meio de partilhar o cálculo de resultados parcelares e de realizar as operações em paralelo.