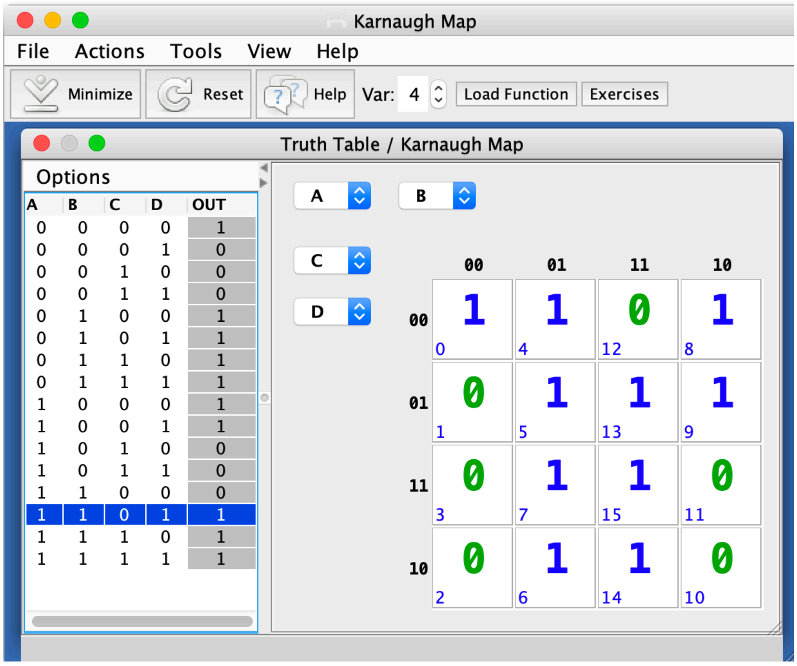
UFRGS - Instituto de Informática, Circuitos Digitais, ERE - 2020-2

Respostas dos primeiros exercícios da Área 2, arquivo ex3.pdf, exercícios 1 a 13

Questão 1)

(B*C)+(!B*!C*!D)+(A*!C*D)+(!A*B)



Questão 2)

A função de um mux2x1 com entradas a e b e controle s é = !s*a + s*b
O sinal do centro que seleciona os dois outros mux é !b*a+b*!a
z1= !(!b*a+b*!a)*c + (!b*a+b*!a)*!c = a xor b xor c
z2= !(!b*a+b*!a)*c + (!b*a+b*!a)*b = (!A*B)+(!A*C)+(B*C) ou !A(B+C) + B*C

Questão 3)

Implementar equações com multiplexadores. A forma mais básica e sistemática de realizar isso é separando a função uma variável por vez, o que pode ser feito algebricamente (mais difícil) ou separando a tabela verdade. Assim, por exemplo, tomamos a variável A, e separamos os casos onde A=0 e A=1, analisando a função em relação a B,C,D.

Neste caso, para a função do item a: F = !A*B + !A*!B*!C + B*!C*!D + A*B*D + !B*C*D
Separando a variável A algebricamente, temos:
F_A0 = B + !B*!C + B*!C*!D + !B*C*D
F_A1 = B*D + B*!C*!D + !B*C*D

Agora para implementar estas duas funções, separamos a variável B, e temos:
F_A0B0 = !C + C*D
F_A0B1 = 1
F_A1B0 = C*D
F_A1B1 = D+!C*!D

Repetindo isso mais um nível com a variável C teremos 8 funções cujas expressões somente podem ser 0 1 D ou !D, que são entradas a serem colocadas em 4 mux 2x1, em uma árvore de 4,2,1 muxes 2x1, controlados pelas variáveis C, B e A, respectivamente.

Questão 4)

Esse exercício é muito simples, e apenas serve para lembrar que uma porta XOR já faz essa operação naturalmente, de selecionar um sinal direto ou invertido.

Z = C xor X

Questão 5)
Multiplicador de 2 bits: a1a0 multiplicado por b1b0

O restultado tem por default 4 bits, r3r2r1r0, e consideraremos b como o multiplicador.
r0 = a0b0
r1 = a1b0 + a0b1 (gerando carry c1)
r2 = a1b1 + c0 (gerando carry c2)
r3 = c2

Questão 6)

Um mux com TGs deve ter duas TGs conectadas na saída e controladas por sinais invertidos. Note que uma inversão do sinal já é necessária para operar uma única TG, mas pode ser a mesma necessária para a segunda, com os controles ligados ao contrário. Assim, cada TG tem um par de transistores, e precisamos de um inversor, totalizando 6 transistores.
Para a implementação com portas lógicas, precisamos de 3 portas de duas entradas e mais um inversor, totalizando 14 transistores.

Questão 7)

Usa o mesmo comportamento já citado de que a porta XOR funciona como um seletor de inversor. Para o item 7.a), Y = 10001100, com a saída normal do registrador, e para o caso do item7.b), Y=00001100, invertendo os sinais.

Questão 8)

Esta questão explora a outra grande aplicação das portas XOR, pois elas separam os casos onde as entradas são diferentes, dos iguais (não importando se/qual estão em 0 ou 1).

Então, um comparados é apenas uma função not XOR, ou XNOR. Para comparar operandos inteiros, como os dois bytes da questão, entretanto, precisamos garantir que todos os bites (um e outro e outro) sejam iguais, fazendo uma AND de todas as XNORs.

Questão 9)

Para implementar um circuito com portas NOR, parece natural minimizar como produto de somas, isto é, agrupando os 0s da função no mapa de KArnaugh, e fazendo o produtório dos maxtermos. Mas não necessáriamente. Como não há garantia de qual expressão ficará menor, se a que representa o on-set (uns) ou o off-set (zeros) da função, podemos minimizar como de costume para soma de produtos, identificando os cubos implicantes primos em 1, e depois usar a equivalente de que NOT NOR = OR e de que NOR (NOT, NOT, …) = AND para implementar a soma de produtos apenas com NORs e INVs.

Questão 10)

Vide slides da disciplina.

Questão 11)

F = [00110101], ou seja, F = !Sel*A + Sel*B, e o circuito deriva da expressão dessa estrutura, uma porta OR, duas portas AND e um inversor.

Questão 12)

Pode ser feita algebricamente, considerando que Y = !!Y, e as negações pode ser passadas para dentro da expressão, ou então pode ser feito novamente por circuitos equivalentes para OR e AND, implementados com NORs e Invs.

Questão 13)

Para implementar uma função de 3 entradas com um multiplexador 4x1, com dois sinais de controle, podemos selecionar duas das variáveis para controlar o mux, e conectar a terceira ou constantes 0 e 1 nas suas entradas. Por simplicidade, selecionamos as variáveis A e B, as primeiras, mais significativas, e avaliamos então funções de S em relação a C nos quatro quartos da tabela, que nesse caso ficam sendo S00 = 0, S01 = !C, S10 = C e S11 = 1.