

# Technische Informatik: Abgabe 3

Michael Mardaus

Andrey Tyukin

13. November 2013

## Exercise 3.1 (MUX Wizardry)

**a)** We can construct a  $n$ -MUX from a 1-MUX and two  $(n-1)$ -MUX'es, clamping the two larger MUX'es to the inputs of the 1-MUX. A trivial induction immediately yields that we need  $2^n - 1$  1-MUX'es to construct a  $n$ -MUX in that way. For example, for a 4-MUX we need 15 1-MUX'es:

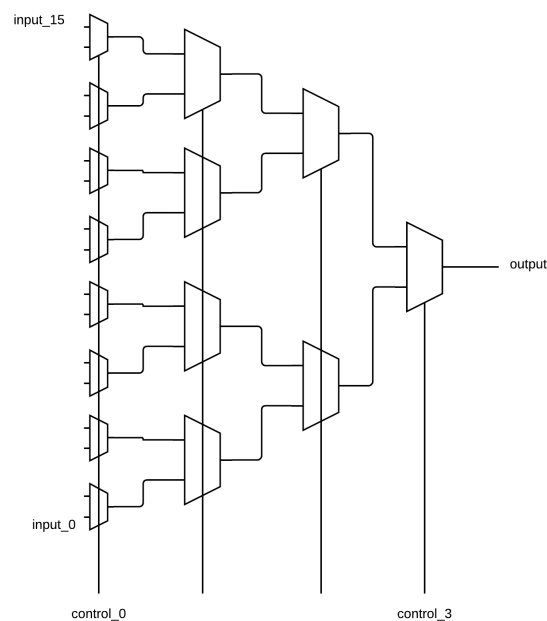


Abbildung 1: Solution to 1 (a)

**b)** We have  $f = m_0 + m_3 + m_6 + m_{13}$ . (Notice that we read the numbers from right to left, interpreting  $x_4$  as least significant bit and  $x_1$  as most significant bit).

All we have to do is to mark the corresponding inputs with ones (notice that the order of control

signals  $x_1$  to  $x_4$  corresponds to reading the numbers from right to left, the topmost input corresponds to the least significant bit).

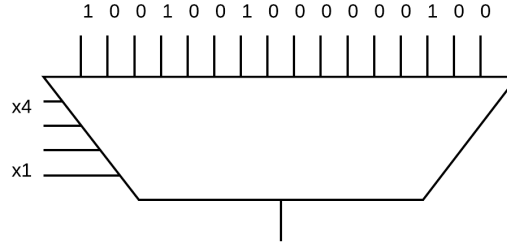


Abbildung 2: Solution to 1 (b)

c) Our goal is to implement the function from part b) with only two 2-MUX'es and one inverter. To derive the optimized circuit, we need some notation. We shall denote an N-MUX as follows:

$$MUX_{c_0, \dots, c_{N-1}}(i_0, \dots, i_{2^N-1}) := \bigvee_{k=0}^{2^N-1} i_k m_k(c_0, \dots, c_{2^N-1}).$$

where  $m_k$  is the 0-indexed minterm. More concretely, for 2-MUX we use the notation:

$$MUX_{l,h}(a, b, c, d) := a\bar{h}\bar{l} + b\bar{h}l + c\bar{h}l + dhl.$$

Notice that from this definition and idempotence it immediately follows:

$$MUX_{l,h}(a, b, c, d) = MUX_{l,h}(a, b\bar{h}, c, dh). \quad (*)$$

Using this, we calculate:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_3x_4 + \bar{x}_1x_2x_3\bar{x}_4 + x_1x_2\bar{x}_3x_4 \\ &\stackrel{\text{def MUX}}{=} MUX_{x_2, x_1}(\bar{x}_3\bar{x}_4 + x_3x_4, x_3\bar{x}_4, 0, \bar{x}_3x_4) \\ &\stackrel{\text{def XOR}}{=} MUX_{x_2, x_1}(\neg(x_3 \nrightarrow x_4), (x_3 \nrightarrow x_4)x_3, 0, (x_3 \nrightarrow x_4)x_4) \\ &\stackrel{(*), \text{ idempotence}}{=} MUX_{x_2, x_1}(\neg(x_3 \nrightarrow x_4), (x_3 \nrightarrow x_4)x_3\bar{x}_1\bar{x}_1, 0, (x_3 \nrightarrow x_4)x_4x_1x_1) \\ &\stackrel{+0}{=} MUX_{x_2, x_1}(\neg(x_3 \nrightarrow x_4), MUX_{x_1, (x_3 \nrightarrow x_4)}(0, 0, x_3, x_4)\bar{x}_1, 0, MUX_{x_1, (x_3 \nrightarrow x_4)}(0, 0, x_3, x_4)x_1) \\ &\stackrel{(*)}{=} MUX_{x_2, x_1}(\neg(x_3 \nrightarrow x_4), MUX_{x_1, (x_3 \nrightarrow x_4)}(0, 0, x_3, x_4), 0, MUX_{x_1, (x_3 \nrightarrow x_4)}(0, 0, x_3, x_4)) \end{aligned}$$

In the step marked as "+0" we used two transformations of the type

$$(x_3 \nrightarrow x_4)x_3\bar{x}_1 = (0 + 0 + x_3(x_3 \nrightarrow x_4)\bar{x}_1 + x_4(x_3 \nrightarrow x_4)x_1)\bar{x}_1 = MUX_{x_1, (x_3 \nrightarrow x_4)}(0, 0, x_3, x_4)\bar{x}_1$$

two times. Now we can replace the XOR expression by another MUX, and obtain the following circuit:

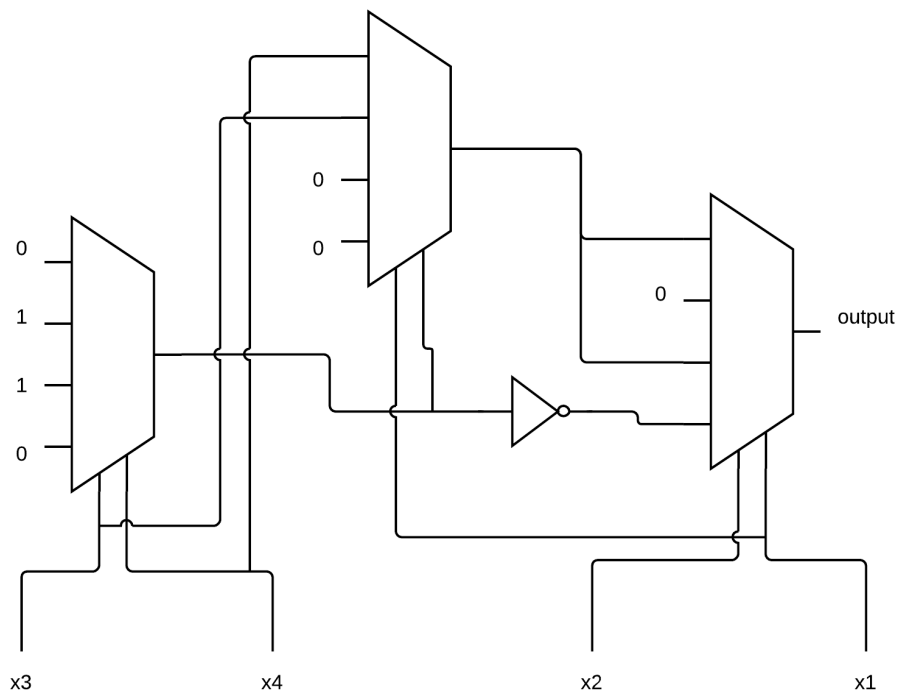
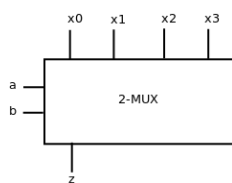


Abbildung 3: Solution to 3.1 (c)

## Exercise 2 (Air conditioner)

E-mail sent to tutors, awaiting clarification of the question.

## Exercise 3: MUX is universal



1. AND:  $x_0 = x_1 = x_2 = 0$  and  $x_3 = 1$  yields output  $z = a \wedge b$
2. OR:  $x_0 = 0$  and  $x_1 = x_2 = x_3 = 1$  yields output  $z = a \vee b$
3. NOT: since NOT is an unary operand  $b = 0$  and  $x_0 = 1, x_1 = 0, (x_2 = x_3 = 0)$  yields output  $z = \neg a$

What's all this racket? Clamp 1,1,1,0 to the inputs, yield a NAND  $\Rightarrow$  epic victory?...

## Exercise 4 (4-DeMUX from smaller parts)

If we want to construct a 4-DeMUX, but have only 4 2-DeMUX'es and infinite number of simpler gatters at our disposal, we can construct an additional 2-DeMUX from scratch from simpler AND gatters and inverters, and then clamp the four 2-DeMUX'es to it's outputs:

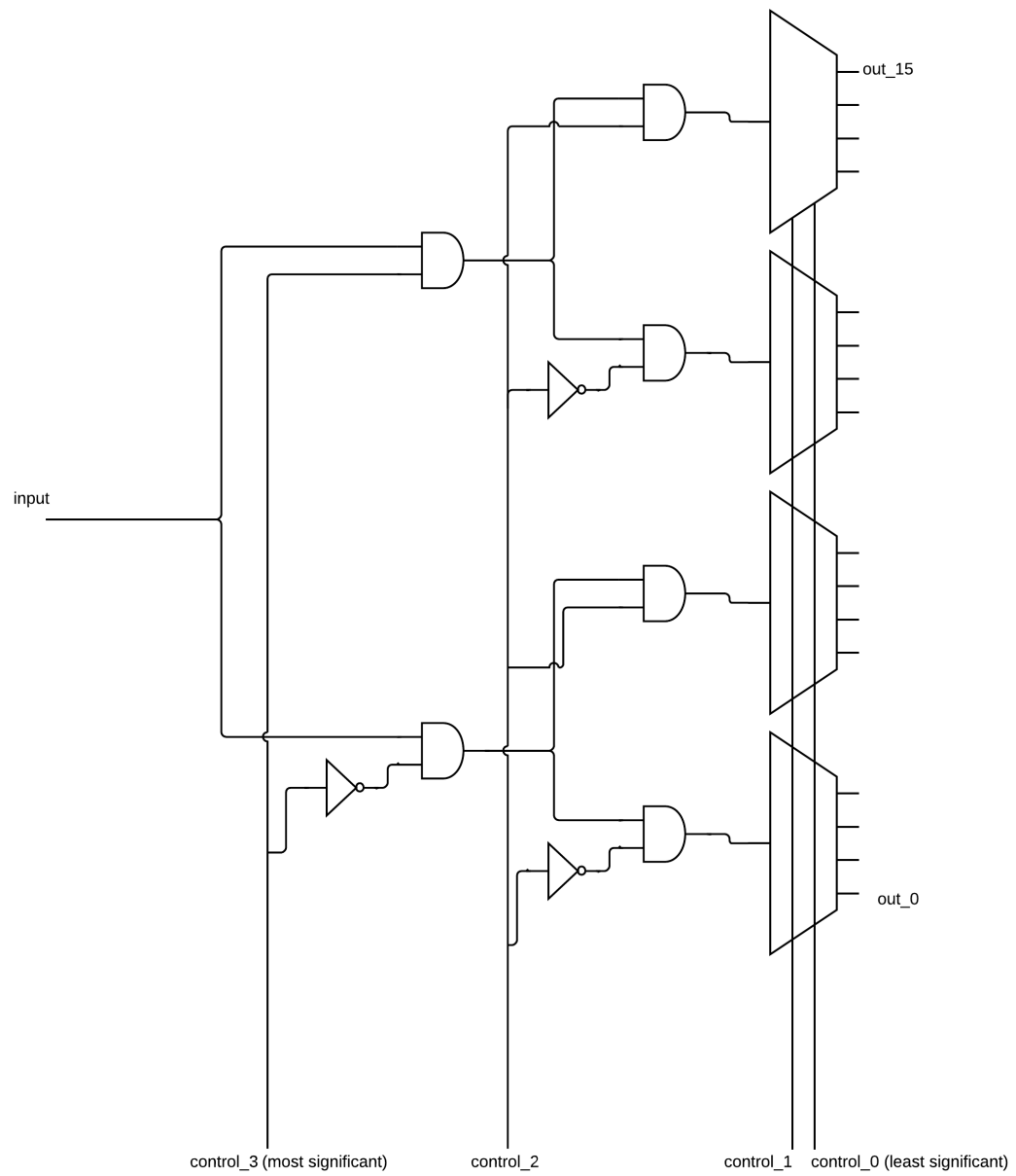


Abbildung 4: Solution to 3.4