Exercises 3
Logic Design

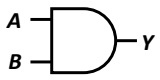
Computer Organization and Components / Datorteknik och komponenter (IS1500), 9 hp
(not part of IS1200, but can be used for review)

KTH Royal Institute of Technology
Friday 18th December, 2020

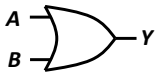
Suggested Solutions

Gates and Boolean Algebra

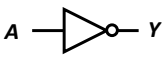
1.

AND 

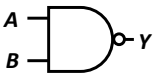
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR 

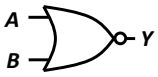
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT 

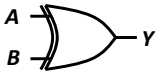
A	Y
0	1
1	0

NAND 

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR 

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

XOR 

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

2. (a) The truth table is

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

(b) The boolean algebra expression is: $\overline{\overline{A}\,\overline{B}}$.

(c) By applying De Morgan's Theorem, we get $\overline{\overline{A + B}}$, which with the law of involution can be reduced to $A + B$.

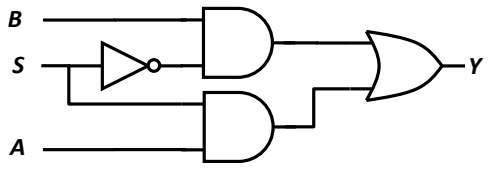
3. (a) The truth table is

A	B	C	Y_1	Y_2
0	0	0	1	0
0	0	1	1	1
0	1	0	0	Z
0	1	1	0	Z
1	0	0	0	Z
1	0	1	0	Z
1	1	0	1	0
1	1	1	1	1

(b) The tristate buffer passes through the input data if the enable signal is true, else the output signal is said to be *floating*. This is indicated in the truth table with character Z. When a signal is floating, it means that the signal has high impedance, that is, giving similar behavior as if the wire was disconnected.

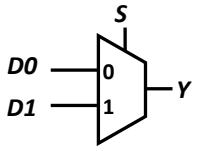
4. (a) $\overline{S}\,\overline{A}\,B \;+\; \overline{S}\,A\,B \;+\; S\,A\,\overline{B} \;+\; S\,A\,B$
By commutativity, change the order in the first two terms
$\overline{S}\,B\,\overline{A} \;+\; \overline{S}\,B\,A \;+\; S\,A\,\overline{B} \;+\; S\,A\,B$
By using the law of distributivity, we can break out the terms
$\overline{S}\,B(\overline{A} + A) \;+\; S\,A(\overline{B} + B)$
Not that we can do this since we can interpret $\overline{S}\;B$ as one symbol. Using the dual theorem for complements we get
$\overline{S}\,B \cdot 1 \;+\; S\,A \cdot 1$
Finally, using identity, we get
$\overline{S}\,B \;+\; S\,A$

(b) A possible circuit is as follows



Multiplexers, Decoders, and Adders

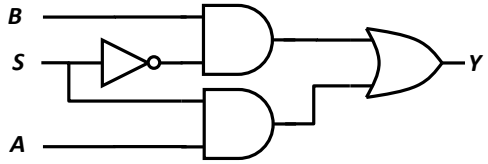
5. (a) A 2:1 multiplexer looks as follows



(b) A truth table can be written as follows. Note how the symbol ? for "don't care" means that it does not matter if this value is 1 or 0: it gives the same output anyway.

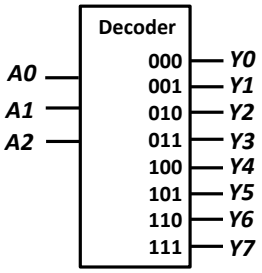
S	A	B	Y
0	?	0	0
0	?	1	1
1	0	?	0
1	1	?	1

(c) A possible circuit is as follows



(d) It is exactly the same circuit! Note also how the truth table in exercise 4 compares with the one above.

6. (a) A 3:8 decoder looks as follows



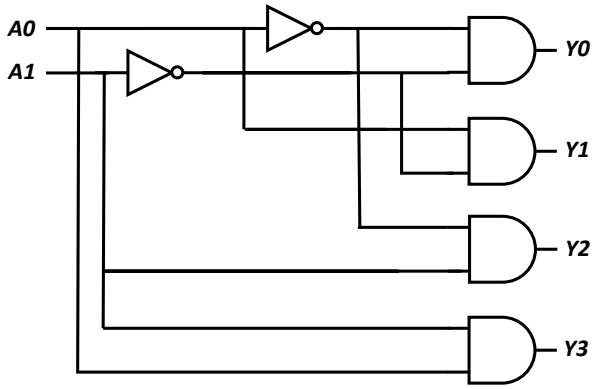
(b) Truth table

A1	A0	Y3	Y2	Y1	Y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

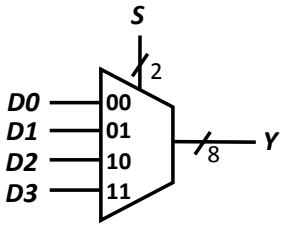
Note that only one output signal is true at the same time.

(c) For the expression $X{:}Y$ (for instance $X = 2$ and $Y = 4$ in a 2:4 decoder), symbol X means the number of input bits, and Y the number of output bits. Note also the relationship $Y = 2^X$.

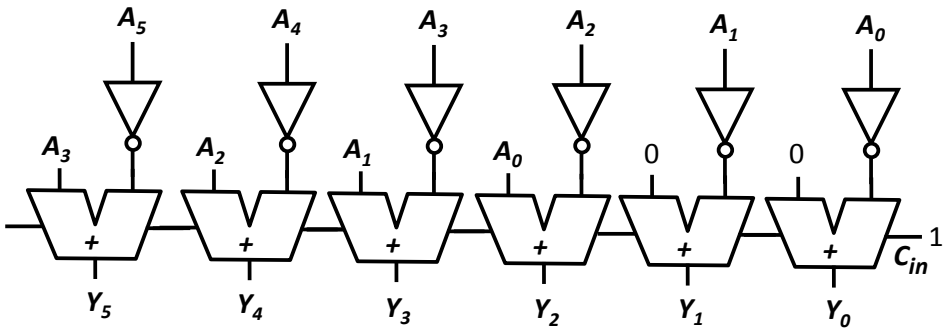
(d) A possible solution:



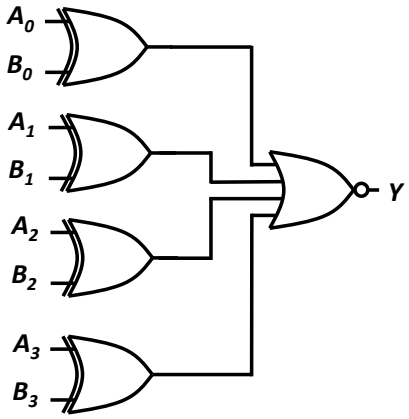
7. The following multiplexor shows that the output data signal has a 8-bit bus width. As a consequence, the input data bus width must also be 8-bit. Hence, we only need to specify one of the signals, either the input or the output.



8. The following circuit implements a 6-bit ripple-carry adder. Multiplication by 3 is done by shifting the input to the left by 2, and then subtracting the input value.



9. One possible solution for a 4-bit equality comparator.

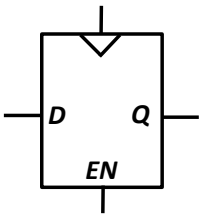


Latches, Flip-Flops, Registers, and Register Files

10. An SR latch can be described with the following truth table.

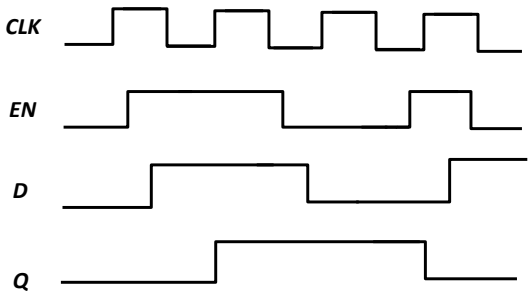
S	R	Q	$\overline{Q}$
0	0	Q_{pre}	$\overline{Q}_{pre}$
0	1	0	1
1	0	1	0
1	1	0	0

11. (a)



(b) A D Flip-Flop is edge triggered and not level-triggered, as in the case of the D latch. The SR latch does not have a data input, as both D Flip-Flops and D latches have.

(c)



12. (a) At the start, $Y_1 = \texttt{0x3}$ and $Y_2 = \texttt{0x5}$, that is, the initial values of the registers. Note that each of the registers hold a 8-bit value. After the first rising edge, the registers read the value that is available on the left hand side. R1 then receives value $\texttt{0x8}$ because it receives the sum of value Y_2 and $A = \texttt{0x3}$. R2 just receives the value from Y_1. Hence, after the first rising edge $Y_1 = \texttt{0x8}$ and $Y_2 = \texttt{0x3}$. After the second clock cycle, we have $Y_1 = \texttt{0x6}$ and $Y_2 = \texttt{0x8}$, and after the third $Y_1 = \texttt{0xB}$ and $Y_2 = \texttt{0x6}$.

    (b) In this exercise, it is important to notice that the XOR gate works as an inverter, since $B = 255$ means that all bits are set to one. Convince yourself that XORing with value one inverts a signal. Since the carry in signal to the adder is 1, the circuit will perform a subtraction of value 2 in each iteration. At start, $Y_1 = \texttt{0x10}$ and $Y_2 = \texttt{0x8}$. After the first rising edge $Y_1 = \texttt{0x6}$ and $Y_2 = \texttt{0x10}$. After the second clock cycle, we have $Y_1 = \texttt{0xE}$ and $Y_2 = \texttt{0x6}$, and after the third $Y_1 = \texttt{0x4}$ and $Y_2 = \texttt{0xE}$.

13. (a) A register is a memory that is updated at a clock edge. A register also has a memory, and it is also updated at a clock edge. However, a register file is a simple addressable memory, where you specify an address for where you should read and write data. A register has no address, and the whole register is updated at the clock edge. Both a register file and a register can either be triggered on the raising or falling edge.

    (b) The register file has two read ports and one write port.

    (c) Each data item is 16 bits large and the register file can address $2^4 = 16$ data items. Hence, the total number of bits that can be stored in this register file is $16 * 16 = 256$ bits.

    (d) $RD_1 = \texttt{0x22}$ and $RD_2 = \texttt{0xe7}$. The new known state is $\{\texttt{0x3} \mapsto \texttt{0x53}, \texttt{0xa} \mapsto \texttt{0x22}, \texttt{0x9} \mapsto \texttt{0xe7}\}$