# First Set of Exercises

**Exercise 1**
Is the following equation correct?

$$(w\,[m:n])\,[p:q] \;=\; (p{+}n \leq m \rightarrow w\,[p{+}n : q{+}n] \mid w\,[m : q{+}n])$$

If so, show why it holds.

If not, explain why not and write down *and justify* a corrected version. □

**Exercise 2**
Are all the three equations and three conditional equations given below true? Justify the ones you think are true and give counterexamples to the ones you think are false.

$$\begin{array}{rcl}
 & & (w_1 \frown w_2)\,[|w_2|{-}1 : 0] \;=\; w_2 \\
 & & (w_1 \frown w_2)\,[|w_1|{+}|w_2|{-}1 : |w_2|] \;=\; w_1 \\
 & & \mathsf{V}(w_1 \frown w_2) \;=\; 2^{|w_2|}\mathsf{V}(w_1) + \mathsf{V}(w_2) \\
p < |w_2| & \Rightarrow & (w_1 \frown w_2)\,[p : q] \;=\; w_2\,[p : q] \\
q \geq |w_2| & \Rightarrow & (w_1 \frown w_2)\,[p : q] \;=\; w_1\,[p{-}|w_2| : q{-}|w_2|] \\
p \geq |w_2| \;\wedge\; q < |w_2| & \Rightarrow & (w_1 \frown w_2)\,[p : q] \;=\; w_1\,[p{-}|w_2| : 0] \frown w_2\,[|w_2|{-}1 : q]
\end{array}$$

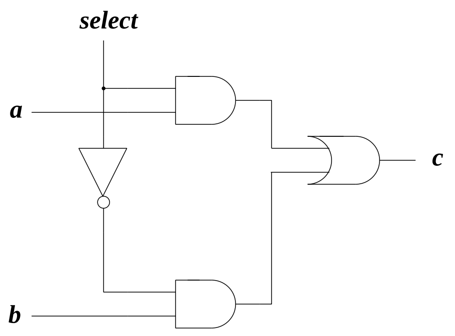
[**Hint:** Mike Gordon hopes they are all true!] □

**Exercise 3**
If $m > n$, why is $w\{m{\leftarrow}b\}\,[n:0] \;=\; w\,[n:0]$? □

**Exercise 4**
What is $\mathsf{V}(w\{m{\leftarrow}b\})$ in general? Justify your answer. □

**Exercise 5**

Consider the following design for a multiplexer:

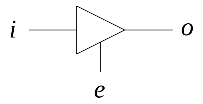



Write down the behaviour of the components, compose them together and then show that the resulting term is equivalent to:

$c \;=\; (select \;\rightarrow\; a \;\mid\; b)$

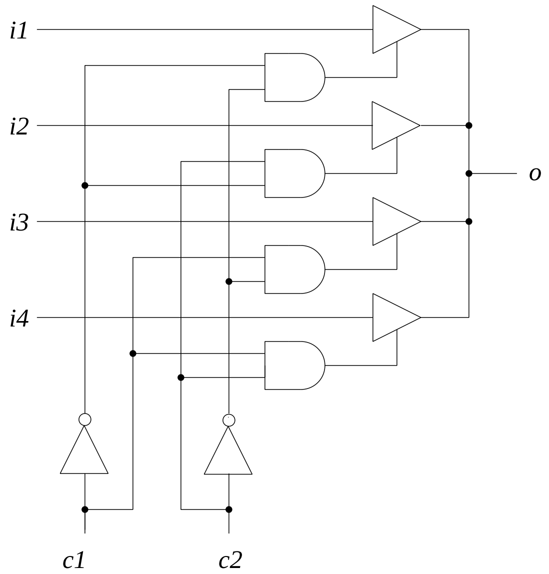
□

**Exercise 6**
A tri-state buffer connects its input $i$ to its output $o$ when the enable line $e$ is driven with Hi. When $e$ is Lo the output $o$ is in a high impedance state $Z$.




Write down and explain a logical predicate modelling such a buffer.

The circuit shown below is a four-way multiplexer designed with tri-state logic. Assume that the control inputs $c1$, $c2$ and the data inputs $i1$, $i2$, $i3$, $i4$ are either Hi or Lo.

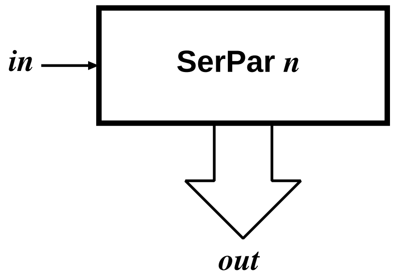



1. Write down a specification of the multiplexer.

2. Write down predicates modeling an inverter and an and-gate.
3. Explain how the multiple driving of the output $o$ by the four buffers can be modelled.
4. Explain how to compose together the predicates modelling the components of the multiplexer into a predicate representing the complete circuit and write down the result.

□

**Exercise 7**
The serial-to-parallel converter SerPar has a bit input $in$ and an $n$+1-bit word output $out$:

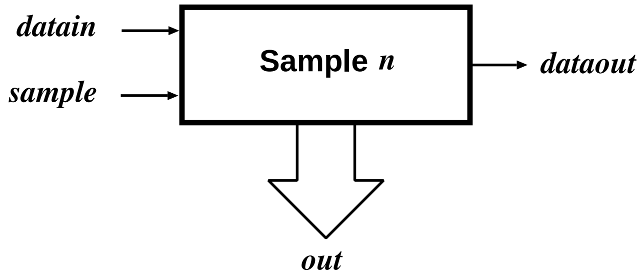



SerPar is specified by:

$$\mathsf{SerPar}\ n\ (in, out)\ \equiv\ (\forall t.\ |out(t)| = n{+}1)\ \wedge\ \forall m.\ m \leq n\ \Rightarrow\ \forall t.\ out(t{+}n{+}1)\,[m] = in(t{+}m)$$

Design an implementation of SerPar using unit-delays in series and show that your design meets the above specification. □

**Exercise 8**
Write a formal specification of a device Sample:




The informal specification of Sample $n$ is that the output at $dataout$ at the $t$+$n$+1 is the input at $datain$ at time t, and the output at $out$ at time $t$+$n$+1 is W ($n$+1) 0 if $sample$ is T (i.e. 1) at time $t$+$n$+1, otherwise the output at time $t$+$n$+1 is the $n$+1-bit word consisting of the last $n$+1 bits input at $datain$.

Design an implementation of Sample $n$ with only unit-delays and combinational logic.

Verify that your design meets its specification. [**Hint:** use SerPar of the previous exercise.]

□

**Exercise 9**
You are given several registers $\mathsf{RegT}$ specified by:

$$\mathsf{RegT}(in, out) \ \equiv \ (out\ 0 = \mathsf{T}) \ \wedge \ \forall t.\ out(t{+}1) = in\ t$$

and several combinational NOR gates, each specified by:

$$\mathsf{Nor}(i_1, i_2, o) \ \equiv \ \forall t.\ o\ t = \neg(i_1\ t \ \vee \ i_2\ t)$$

Design a device with one input $in$ and one output $out$ that only uses these components and satisfies the specification $out = \mathsf{Parity}\ in$, where:

$$\begin{array}{lcl} \mathsf{Parity}\ in\ 0 & = & \mathsf{T} \\ \mathsf{Parity}\ in\ (t{+}1) & = & (in\ t \ \rightarrow \ \neg(\mathsf{Parity}\ in\ t) \ \mid \ \mathsf{Parity}\ in\ t) \end{array}$$

Verify that your design meets its specification.

[**Hint:** first do a design using whatever combinational devices you find convenient, then implement these in terms of NORs] □

**Exercise 10**
You are given devices $\mathsf{One}$, $\mathsf{Not}$, $\mathsf{Mux}$ and $\mathsf{Del}$ specified by:

$$\begin{array}{lcl} \mathsf{One}\ out & \equiv & \forall t.\ out\ t = \mathsf{T} \\ \mathsf{Not}(in, out) & \equiv & \forall t.\ out\ t = \neg(in\ t) \\ \mathsf{Mux}(sw, in_1, in_2, out) & \equiv & \forall t.\ out\ t = (sw\ t \ \rightarrow \ in_1\ t \ \mid \ in_2\ t) \\ \mathsf{Del}(in, out) & \equiv & \forall t.\ out(t{+}1) = in\ t \end{array}$$

Use these components to design a resetable register $\mathsf{ResetReg}$ specified by:

$$\begin{array}{lcl} \mathsf{ResetReg}(reset, in, out) & \equiv & (\forall t.\ reset\ t \ \Rightarrow \ (out\ t = \mathsf{T})) \ \wedge \\ & & (\forall t.\ out(t{+}1) = ((reset\ t \ \vee \ reset(t{+}1) \ \rightarrow \ \mathsf{T} \ \mid \ in\ t)) \end{array}$$

Verify that your design meets its specification. □