Upload your solution to each problem as a Logisim .circ file to CourseWorks.
1. (20 pts.) Create a circuit for a 4-to-16 decoder using AND gates and inverters only. Arrange and name the inputs and outputs as shown below. Treat $W$ as the most significant bit and let $A0$ be true when all inputs are false. Only one of the outputs should ever be true.

$W \rightarrow \quad \rightarrow A0$

$X \rightarrow \quad \rightarrow A1$

$Y \rightarrow \quad :$

$Z \rightarrow \quad \rightarrow A15$

Name your solution “hw2-1.circ” and submit it via Courseworks.
2. (10 pts.) In Logisim, implement the logical OR function using just a single two-input MUX (under “Plexers→Multiplexer”; set “include enable” to “no”) and constant “0” and “1” inputs (“Wiring→Constant”). Do not use additional gates.

Draw your inputs and outputs as shown below:

\[
\begin{align*}
A & \rightarrow \\
B & \rightarrow \\
\end{align*}
\rightarrow Y
\]

Name your solution “hw2-2.circ” and submit it via Courseworks.
3. (15 pts.) In Logisim, implement \( F = \overline{X} \overline{Y} \overline{Z} + \overline{Y}Z + X\overline{Y} \) using just constants and \( \) 

(a) a 3-to-8 decoder (under “Plexers→Decoder.” Set “include enable” to “No” and note the input wires are a bundle at the bottom) and an OR gate;

(b) an 8 input mux; and

(c) a 4 input mux whose select inputs are \( X \) and \( Y \), and an inverter.

Arrange your inputs and outputs as shown below and name your files “hw2-3a.circ,” “hw2-3b.circ,” and “hw2-3c.circ.”

\[
\begin{align*}
X \to & \quad \overline{X} \overline{Y} \overline{Z} \\
Y \to & \quad \overline{Y}Z \\
Z \to & \quad X\overline{Y} \\
\rightarrow & \quad F
\end{align*}
\]
4. (20 pts.) Implement an eight-input mux using three four-input muxes and no other gates (constants are OK).

Arrange your inputs and outputs as shown below and name your solution “hw2-4.circ”

A0 →
A1 →
... →
A7 → → F
X →
Y →
Z →

Here, A0 through A7 are the eight inputs, and X, Y, and Z are the three selects. X is the most significant bit, selecting between, e.g., A0 and A4.

Name your solution “hw2-4.circ.”
5. (35 pts.) Implement a three-bit binary carry-lookahead adder “hw2-5.circ.” A0 through A2 and B0 through B2 are the two binary inputs (A0 and B0 are the LSBs), C0 is the carry in, and Y0 through Y3 is the four-bit output. Arrange your inputs and outputs like this:

\[
\begin{align*}
A2 & \rightarrow \\
B2 & \rightarrow \\
A1 & \rightarrow \quad \rightarrow Y3 \\
B1 & \rightarrow \quad \rightarrow Y2 \\
A0 & \rightarrow \quad \rightarrow Y1 \\
B0 & \rightarrow \quad \rightarrow Y0 \\
C0 & \rightarrow \\
\end{align*}
\]

(a) As text labels in your solution, write expressions for \(G_0, \ldots, G_2\) and \(P_0, \ldots, P_2\), the carry generate and propagate functions, in terms of the inputs.

(b) Write sum-of-product expressions for \(C_1, \ldots, C_3\) in terms of the \(G’s, P’s,\) and \(C_0\). Use + for OR, & for AND, and ! for NOT.

(c) Write the equations for the \(Y’s\) in terms of these. Use ^ for XOR.

(d) Implement the carry-lookahead adder circuit corresponding to these equations using inverters, AND, NAND, OR, NOR, and
XOR gates. The critical path should be four gates.