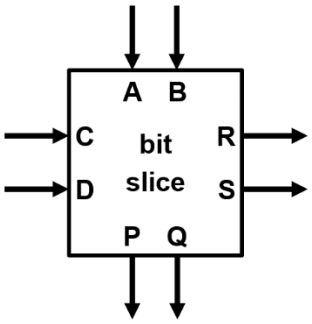
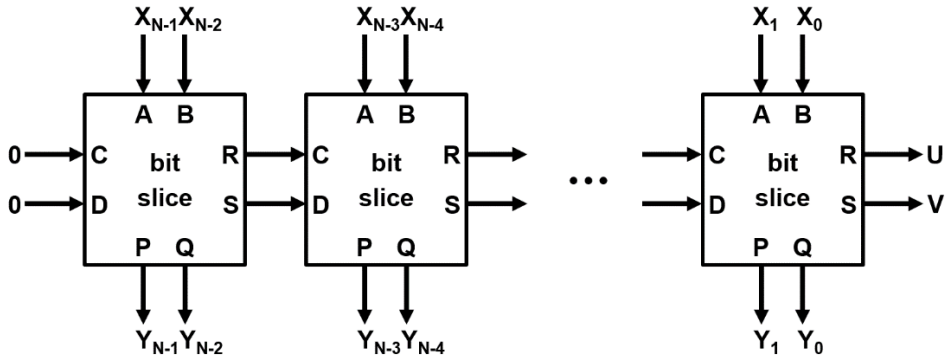
1. **Bit Slice Analysis**

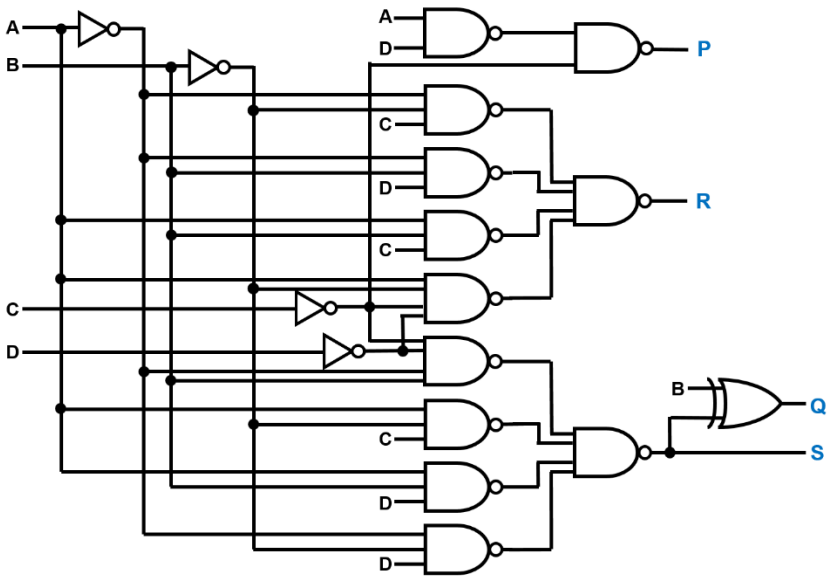
Prof. Lumetta has done it again. He designed a bit-sliced circuit for ECE120, then forgot what the circuit does! Please help him to figure it out. The interface to a single bit slice is shown to the right. Each bit slice consumes two bits of an unsigned number (into inputs A and B) and produces two bits of an unsigned number (from outputs P and Q). Two bits are also passed in from the previous bit slice (more significant bits) as inputs C and D, and two bits are passed out to the next bit slice (less significant bits) as inputs R and S.



For an N-bit unsigned number $X=X_{N-1}X_{N-2}\ldots X_1X_0$, N/2 bit slices are hooked together as shown below to produce unsigned number $Y=Y_{N-1}Y_{N-2}\ldots Y_1Y_0$ and outputs U and V.



A circuit diagram for the bit slice appears to the right.



a. Start by writing a truth table for outputs P, Q, R, and S in terms of inputs A, B, C, and D. Be sure to get the right answer—double-check your work! If you make mistakes here, you will have difficulty understanding the purpose of the circuit.

b. Use your truth table from **part (a)** to fill in the table of all bit slice outputs when four copies of the bit slice are connected together and X=01000011.

	A	B	C	D	P	Q	R	S
bit slice for X_7X_6	**0**	**1**	**0**	**0**				
bit slice for X_5X_4	**0**	**0**						
bit slice for X_3X_2	**0**	**0**						
bit slice for X_1X_0	**1**	**1**						

From the table, write the bits corresponding to outputs Y (8 bits), U (1 bit), and V (1 bit). Finally, write the decimal values represented by both X and Y in the 8-bit unsigned representation.

c. Repeat **part (b)** with input X=01111010.

d. Repeat **part (b)** with input X=0101110111 and five copies of the bit slice.

e. Based on your results from **parts (b)**, **(c)**, and **(d)**, explain how output Y relates to input X.

f. Based on your results from **parts (b)**, **(c)**, and **(d)**, explain how the outputs UV (two bits) relate to input X.

g. If we instead choose to think of the bits of X as representing a number in 2's complement, does the design still operate as intended (your explanation in **parts (e)** and **(f)**)? Explain your answer.

h. Let's analyze the delay of the bit slice. Fill in the table below with the number of gate delays from each input to each output. Remember that we use the longest path (the one with the most gates) for such measurements. If an input has no affect on a particular output, write "N/A" in the corresponding box in the table. Refer to the bottom of p. 55 of the notes for an example of analysis on the comparator developed in Section 2.4.

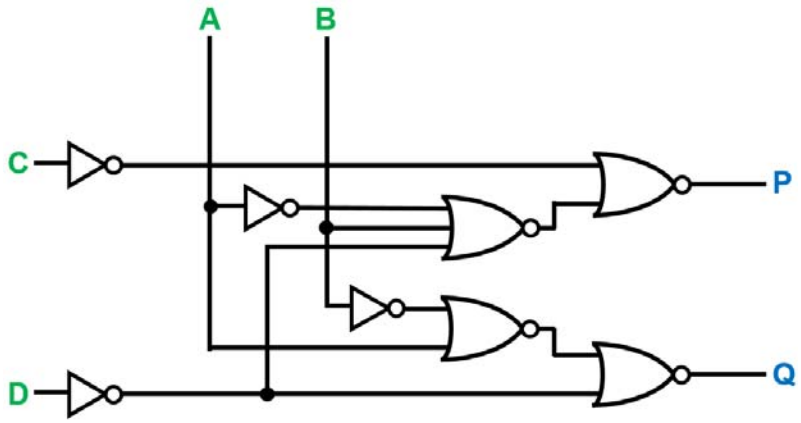
		outputs			
		P	**Q**	**R**	**S**
inputs	**A**				
	B				
	C				
	D				

i. Now calculate the number of gate delays needed to perform a computation using N/2 copies of the bit slice on an N-bit unsigned number. Assume that N is even. Assuming that all bits of A and B arrive together, how many gate delays are needed before the last bit slice produces its last output?

2. **Debugging a Bit-Sliced Comparator**

A friend in ECE120 has designed a bit-sliced comparator for unsigned numbers. The circuit below illustrates a single bit slice. The bits of the numbers to be compared are fed into the A and B inputs. Two additional input bits, C and D, are provided by the bit slice that handles the next most significant bits. In other words, information flows from the most significant to the least significant bits, in the opposite direction as the design in Section 2.4 of the notes. Each bit slice produces outputs P and Q, which are passed to the next bit slice (as inputs C and D, respectively). The representation used for the bits passed between bit slices is as follows: 00 means that A > B; 10 means that A < B; and 11 means that A = B. The CD/PQ=01 pattern is undefined, and should not appear. The first bit slice, corresponding to the most significant bit of the numbers being compared, has inputs CD=11 (which means A=B).

Unfortunately, your friend has made a mistake and needs your help to correct the design.



a. Begin by analyzing the circuit to produce a truth table for outputs P and Q as a function of A, B, C, and D.
b. Copy the bits from your truth table in **part (a)** into two K-maps, one for P, and a second for Q.
c. Using the representation for CD and either your truth table for **part (a)** or your K-maps from **part (b)**, identify the input combination that leads to an incorrect output pattern. There is only one such input combination.
d. Unfortunately, your friend doesn't believe you. Consider the example A=001000 and B=000100. Using these values as inputs, use your truth table to calculate the value of PQ produced by each of the six bit slices. The final output should be PQ=00, as expected, because A > B.
e. Explain why the example in **part (d)** produces a correct answer even though the circuit exercised your so-called "error" From **part (c)**.
f. Give a counterexample to convince your friend that the design is wrong. In other words, give values of A and B such that the final bit slice's PQ pattern is incorrect, and show the PQ bits produced by each of the bit slices when processing your example.
g. Explain how to correct the design by giving a correct equation for the incorrect output(s) and indicating which term or factor is missing or has been implemented incorrectly in your friend's design.