ENGIN 341 – Advanced Digital Design

Homework 4

---

1. Implement the 4-bit multiplier from Figure 1 below in VHDL. Submit the source and testbench code for the multiplier, and images of the waveform from the testbench results.

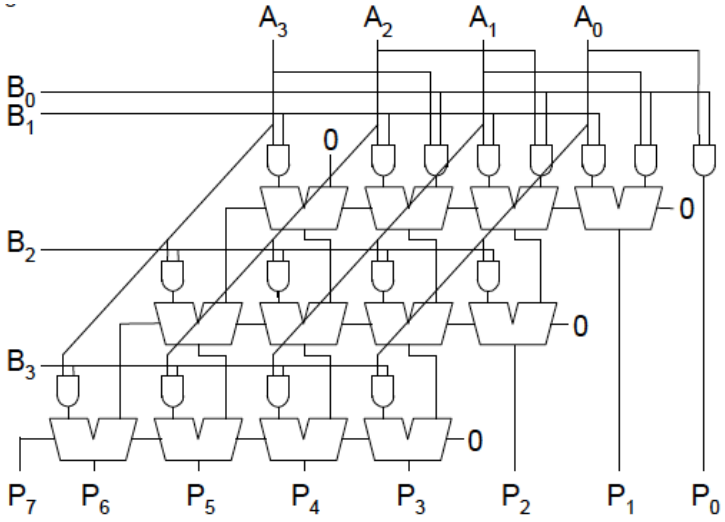

Figure 1 - 4-bit Multiplier

$$
\begin{array}{rcccccccc}
 & & & A_3 & A_2 & A_1 & A_0 & & \\
 & & \times & B_3 & B_2 & B_1 & B_0 & & \\
\hline
 & & & A_3B_0 & A_2B_0 & A_1B_0 & A_0B_0 & & \\
 & & A_3B_1 & A_2B_1 & A_1B_1 & A_0B_1 & & & \\
 & A_3B_2 & A_2B_2 & A_1B_2 & A_0B_2 & & & & \\
+ & A_3B_3 & A_2B_3 & A_1B_3 & A_0B_3 & & & & \\
\hline
P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}
$$

Figure 2 – Partial products for the 4-bit multiplier

2. Implement an 8-bit multiplier by extending the partial products and full-adders. While implementing the 8-bit multiplier, try to find patterns that could be used to generalize the code using generate statements to create an N-bit multiplier. Submit the source and testbench code for the multiplier, and images of the waveform from the testbench results.

NOTE: You may implement the 8-bit multiplier directly or by implementing an N-bit multiplier design and specifying N=8. ***If you create the N-bit design, you may use this to create your 4 bit instance for Q1.***

HINT: Not the most elegant solution, but it may be easier to create N different (2*N - 1)-bit wide partial products and padding them with zeroes in order to align them (i.e., the rows in Fig.2), and then using N full-adders in each column.