

# Boynton Power Multiplier

Dylan Shadoan and Edgar Viveros

ECEN 4303: Digital Integrated Circuit Design  
Oklahoma State University - Fall 2018



**Abstract—**The Boynton Power Multiplier (BPM) is a digital integrated SCMS circuit whose function is to multiply two 8-bit numbers and output their 16-bit product. The design of the BPM includes a bitsliced custom-cell portion as well as an HDL-based standard-cell portion. The custom-cell slices contain several basic components such as adders, multiplexors, and registers. The standard cell portion is primarily a top-level design utilizing an 8-state finite state machine. The BPM is designed on a channel length of 600 nm and is meant to run at 75 MHz.

# INTRODUCTION

The goal of this project is to gain a deeper understanding of bitslice-based design and to apply various VLSI concepts from lectures such as circuit design, layout, verification, testing, and more.

The design process begins with floorplanning and then continues to implementation of gates and registers. After both 1-bit “bitslices” are completed, expansion to 8 bits is as simple as arraying out eight of each. This custom layout was verified by simulation against a “golden file” of known product vectors. The design is then analyzed for timing and size restrictions.

The standard-cell design is implemented via Hardware Descriptive Language and then simulated with ModelSim®.

The final stage of the project includes a chip finishing process in which the final synthesis and simulation takes place.

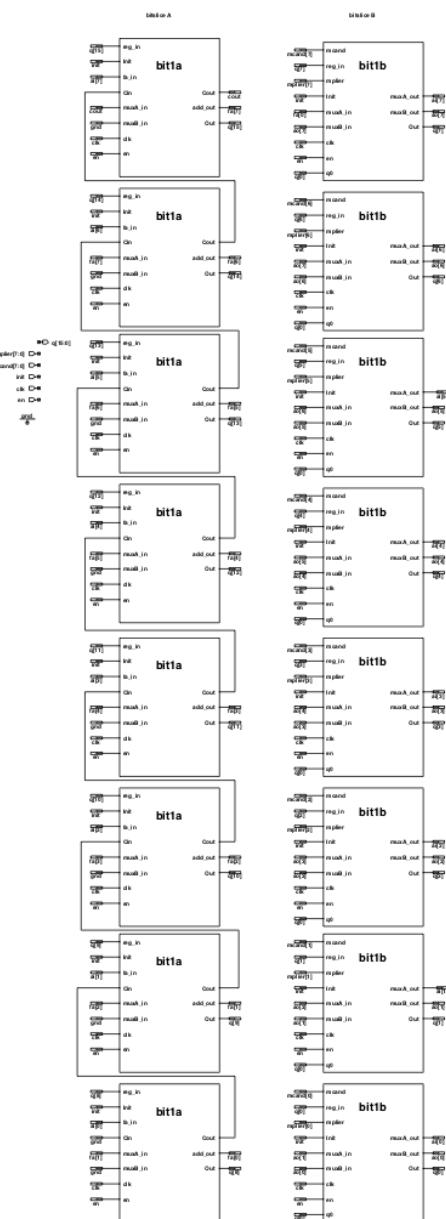
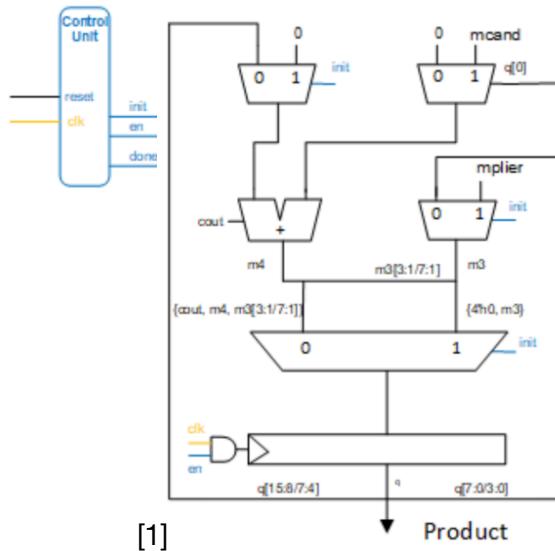
# DESIGN OVERVIEW

8-bit multiplication is achieved in the BPM through a series of 8 additions. This algorithm is similar to the common method for multiplication which involves a series of one-digit multiplications and then a final sum at the end.

$$\begin{array}{r}
 5127 \\
 \times 4265 \\
 \hline
 25635 \\
 307620 \\
 1025400 \\
 \hline
 20508000 \\
 \hline
 21866655
 \end{array}$$

The adder chosen for this design is a carry-propagate ripple-carry adder with a mirror-circuit implementation. A few multiplexors are used to organize the bit information, and D Flip-Flops are used to store the information during and after multiplication. A NAND gate and a few inverters fill out the remainder of the custom logic in the multiplier.

Once the individual elements are laid out, they are combined into two bitslice designs which can be arrayed into the final custom component.



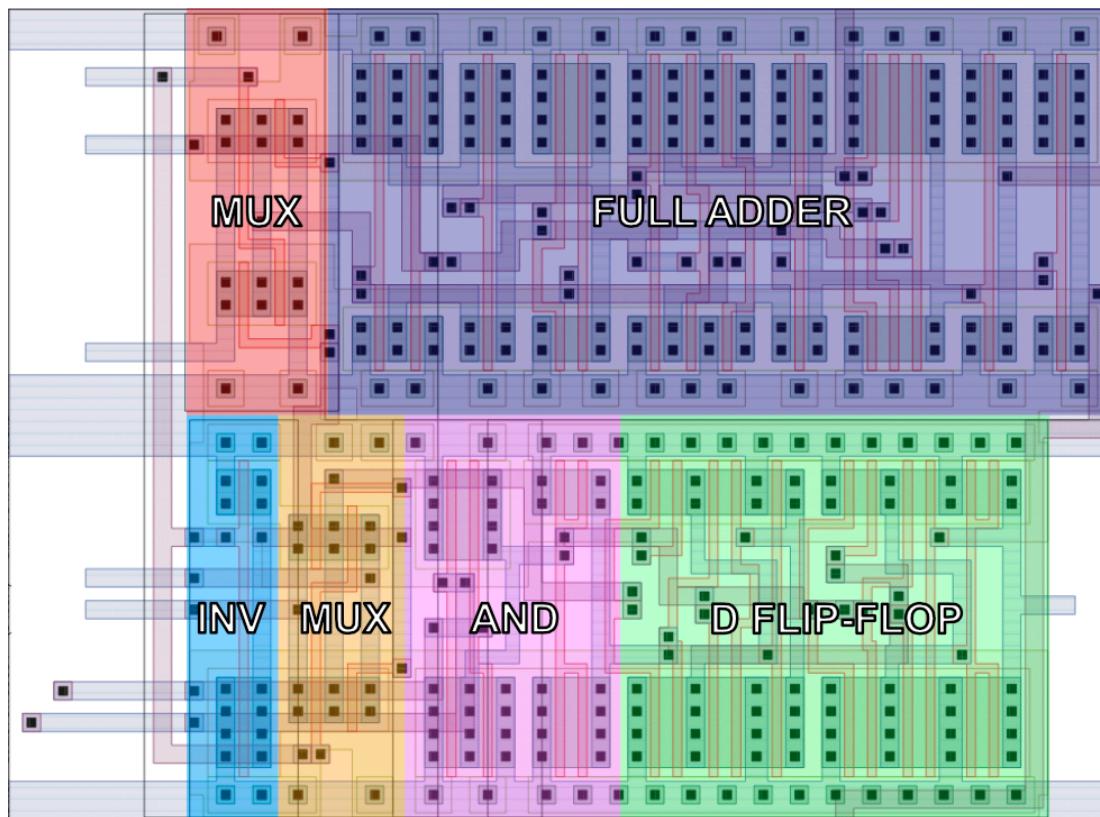
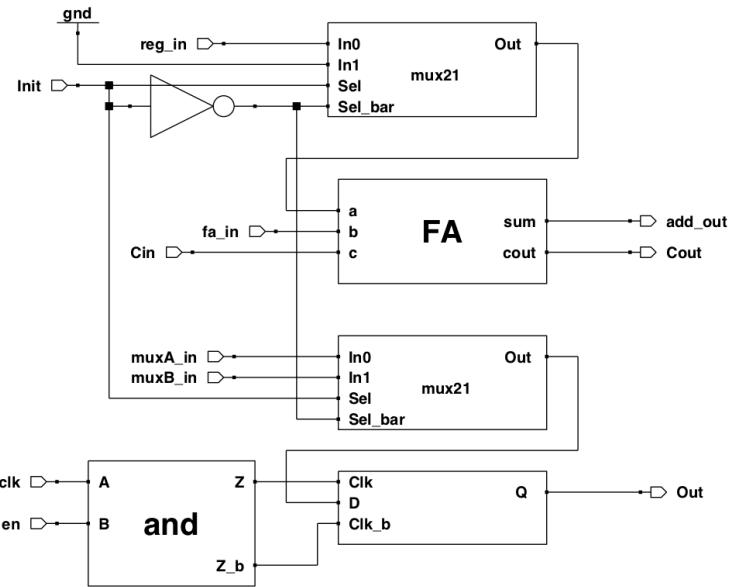
# BITSLICES

There are some minor changes to the given design that were implemented in the spirit of reducing design area (and possibly improving delay). These changes only affect the bitslices, and only by removing some unnecessary inverters. In some applications, this decision could cause some driving issues in the circuit, but for this situation, the extra driving power can be foregone safely.

The first removed inverter was dedicated to the bottom multiplexor's  $\overline{SELECT}$  input. Stealing this signal from the other multiplexor eliminates its necessity.

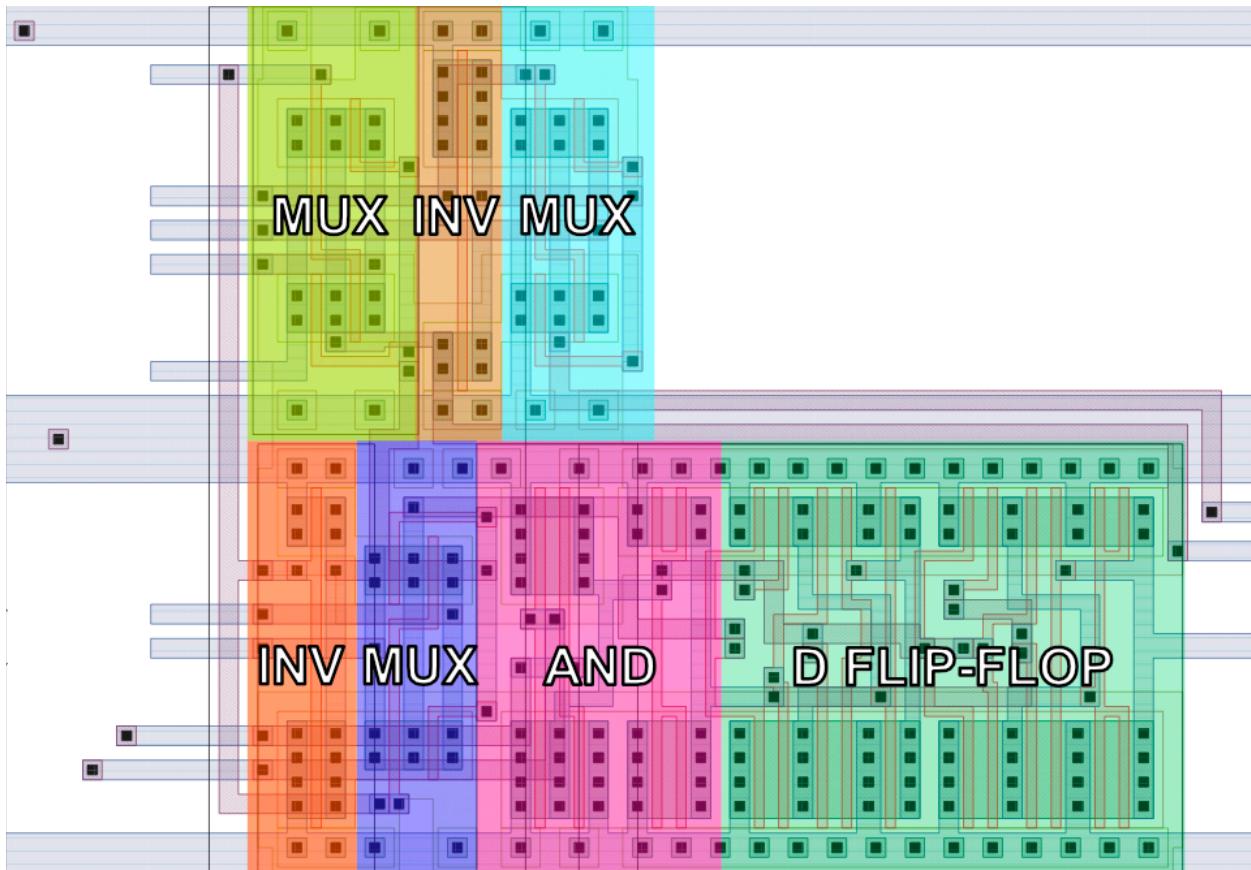
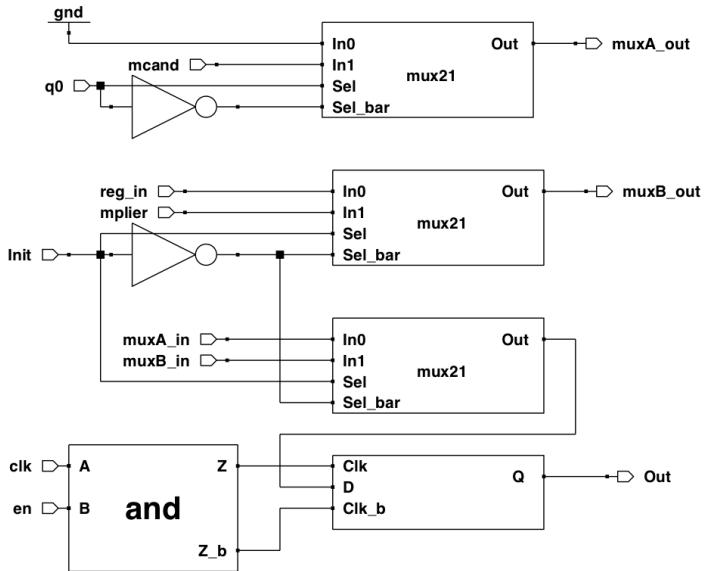
The other removed inverter is a result of the AND gate circuit terminating with an inverter and the D Flip-Flop circuit beginning with one. It turns out these components can simply share one inverter instead of putting two up against each other. In the diagram to the right, the shared inverter lies in the AND gate; this is why there is a  $Z$  and a  $\bar{Z}$  output to the DFF.

## BITSLICE A



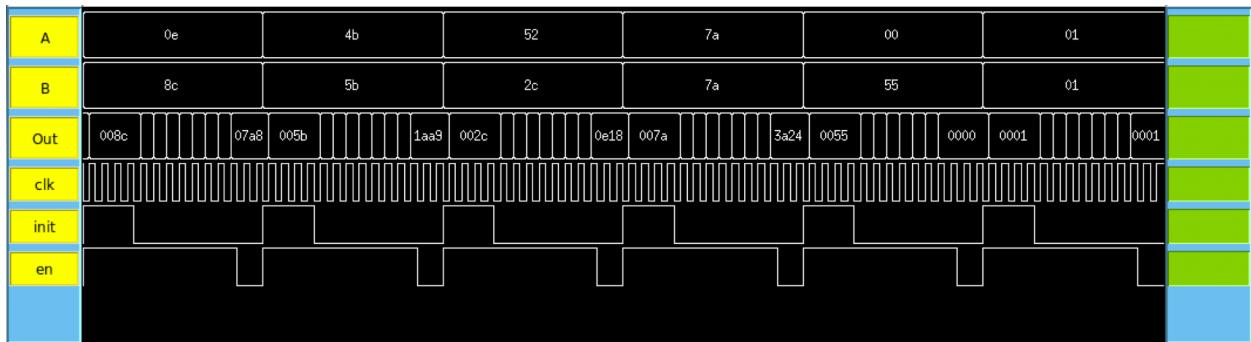
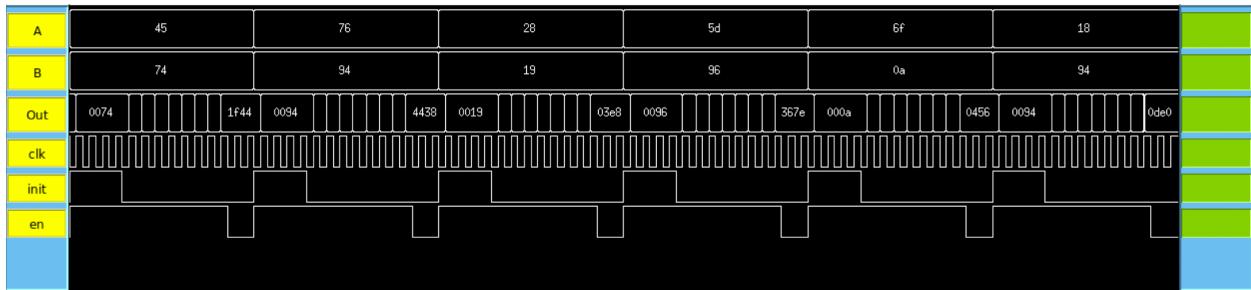
The same alteration was made to Bitslice B.

## BITSLICE B



# SIMULATION

Simulated IRSIM® output from the custom design is compared to the output from the "golden file," file which contains test vectors and asserts their correct products. With the use of assertions, the observed output may verify whether or not the operation of the design is correct. Various vectors are tested to ensure functionality.

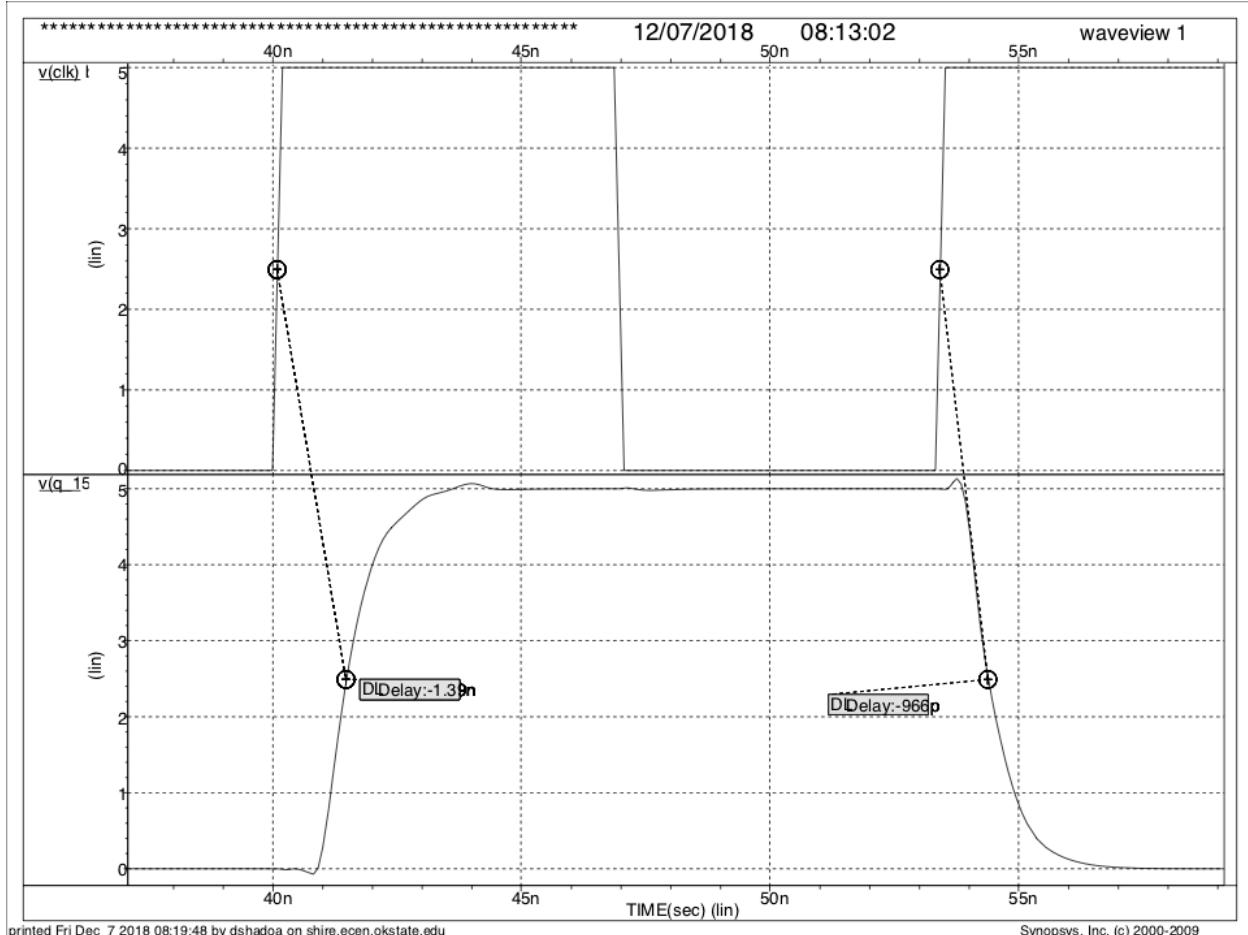


# STATISTICS

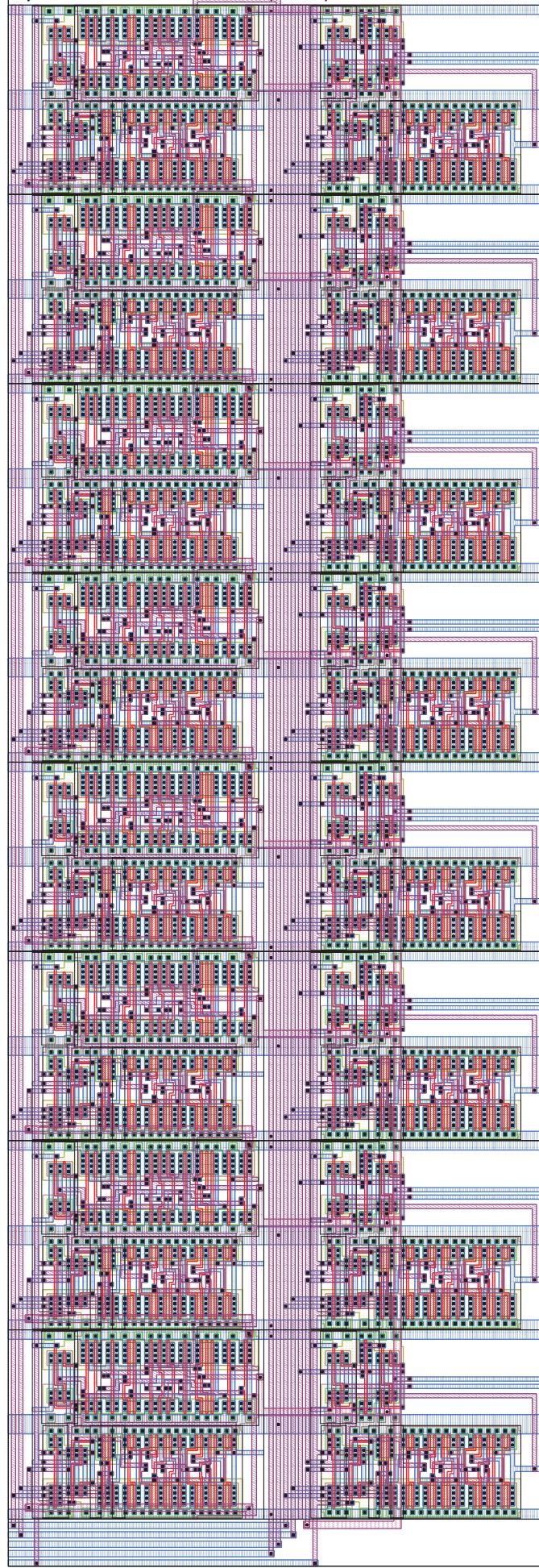
The critical path of the system occurs down the chain of carry operations through the full adder. In order to find the worst-case delay, 0xFF must be multiplied by 0x03 to cause the most carries. Using the HSPICE® waveform output, the propagation delays for the design can be found. The area can be measured using the layout design software.

Custom Cell Properties	
<b>Area</b>	$153.00 \times 447.60 = 68,482.8 \mu m^2$
<b>TPLH</b>	1.39 ns
<b>TPHL</b>	966 ps
<b>TP</b>	1.18 ns
<b>Clock Freq.</b>	75 MHz ( $T = 13.33$ ns)
<b>Channel Length</b>	600 nm

## HSPICE® OUTPUT WAVEFORM

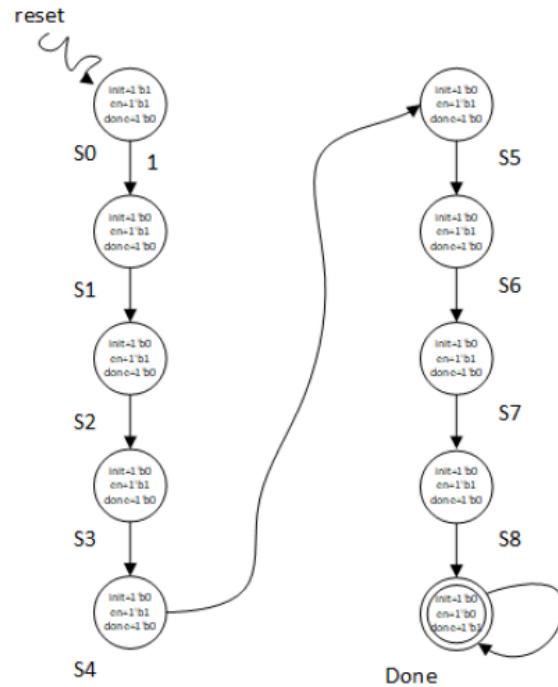


top.cif Scale: 0.022341 (567X) Size: 153 x 447 micr

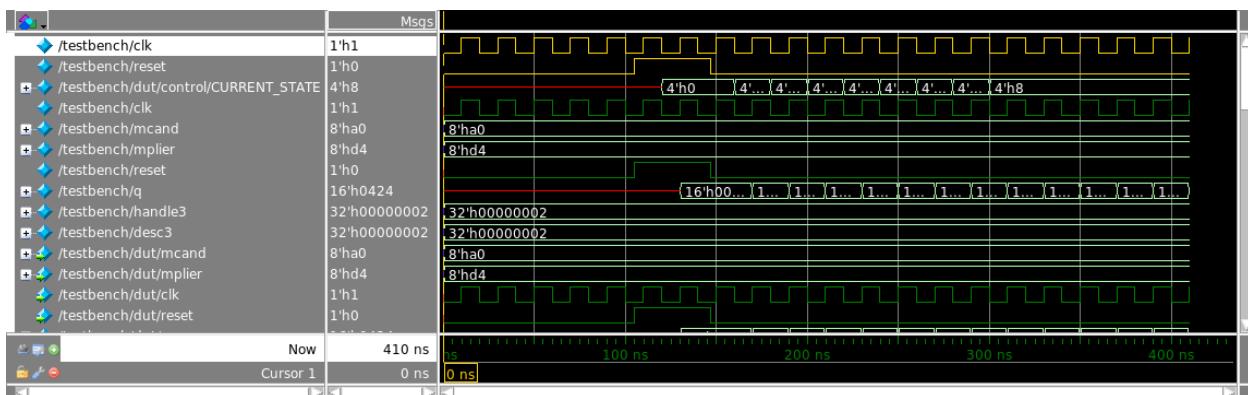


# FINITE STATE MACHINE

The standard-cell portion of the project is implemented via HDL in the form of an 8-state finite state machine. After implementing the FSM using a test bench file, the 8-bit standard design is simulated in ModelSim®. After 8 state transitions, the simulation will output the 16-bit product.



[1]



# **FINAL CHIP DESIGN**

## **REFERENCES**

- [1] J. Stine, "Boynton Power Multiplier (BPM) Project v1.0.1\* ECEN 4303: Digital Integrated Circuit Design", Oklahoma State University, Fall 2018.