8 Bit Kogge-Stone Adder with Optimized XOR Gate

Dong W Kim & Yoonsoo Shin (A09593727 & A10164206)

Adders in electronics is a digital circuit that adds binary numbers.. Many processors use adders not only in arithmetic, but also in calculating addresses within cache and RAM, as well as incrementing and decrementing operators within programs. Because of its basic functions, having a fast, energy efficient, and robust adder is crucial to processors, and computers as whole.

Designing such adder brings many challenges to the table. Assume a simple full adder that adds two one bit inputs. If the inputs are 0 and 0, the output is 0. If the inputs are 0 and 1 or 1 and 0, the output is 1. However, the last logic brings a problem, as 1 and 1 is equal to 0, and there is a carry out number to a higher binary digit. The sum can be calculated XOR gate, as they have the same logic. The carry out operation must also be taken to account by XOR gate, should the carry out value be added to the next adder. As such, one of the biggest challenge is keeping track of carryouts as well as sums, and making the process speedy and efficient.

In the previous architecture, the speed is not optimal because carry ripples through every adder cell causing the critical path to be longer. Each adder cell has to wait until the previous adder has calculated the carry out.

Kogge-stone adder is one of the fastest adder designs commonly used in the industry. It is a parallel version of carry-lookahead adder. Its parallel structure minimizes logic depth and bounded fan-out and computes carries fast at the cost of increased area.

Also, it is fairly easy to implement considering it only consists of 2-bit XOR, AND, and OR gates.

The most innovative part of our adder design is within the simplified design of the 2-bit XOR gate. Any adder, whether it is ripple carry or any form of carry look-ahead, requires multiple XOR gates to function, as it is used to sum the inputs and/or carries; our Kogge-Stone adder utilizes 15 of them. However, most common 2-bit XOR design uses combination of 4 NANDs, or total of 16 transistors. Such quantity of transistors, as well as the fan in and out multiple gates. cause delay, and ultimately, throttle the speed of the adder. To avoid this problem, our team implemented a streamlined version of the 2-bit XOR gate which only requires 8 transistors. Due to its simplicity, we can save space, which is notoriously at a premium in Kogge-Stone adders, as well as power consumption, while increasing its performance.

There were a few mishaps during the design. The first was within our design of the XOR gate. While the XOR gate was optimized for speed, space, and power, certain output was not as our team expected. When the input of the XOR gate is 1 and 1, the returning logic should be 0. However, our output voltage of the said logic came out to be 0.1~0.2V when our VDD was 1. We assume that this problem was derived from the transmission gate used in our design, which are known to be not as robust as other more commonplace gate designs (NAND, NOR, inverters, etc…). While the problem was notable, it did not cause any severe problems; it seemed that many of other robust gates and cells stifled the low noise, and therefore, not affect the overall result of the Kogge-Stone adder.

Another problem our team faced was finding the optimal VDD for the Kogge-Stone design. Our team originally planned to increase the VDD from the original 1V to 2V for both the synthesized ripple carry adder and our Kogge-Stone design in order to increase the maximum frequency. For the synthesized, we saw increase in its performance without any problems. However, Kogge-Stone adder had problems; it seemed that the performance seemed to decrease and the results seemed to lose its robustness. Our team observed many floating outputs when the output should be at 0. Even when our frequency was decreased to half of that of 1V max frequency, the problem persisted. Our team hypothesized that this problem was due to the custom 2-bit XOR gate and its 0.1~0.2V output when its inputs are 1 and 1. However, when we replaced the custom XOR gate with the library-given freepdk45_cells XOR2X1 cell, the problem persisted. Because we did not see any improvements, our team decided to continue using the custom XOR gate. In order to increase the performance, but to avoid the standing problem, we have decreased the new VDD to 1.3 V, which seemed to show increase in performance, but not create the said problem. The cause of the problem seems to be inherent with the Kogge-Stone adder design. For more concrete and detailed explanation, more time and study is needed.

Common enhancements to the origial Kogge-Stone adder involves increasing the radix and sparsity. Increasing the radix from 2 to 4 or more increases power and delay of each cell, but reduces the number of stages needed. Increasing the sparsity means more carry bits generated, reducing the total computation needed.

Overall, our Kogge-Stone adder design was faster and more power efficient than the synthesized ripple-carry counterpart. However, this does not mean that our team's design was perfect; we have learned many lessons that could improve the design of the adder if we were to implement it again in the future. The first of which is the use of dynamic logic. In contrast to the static CMOS design, which our team has implemented, dynamic logic design is known to be faster and less area consuming than its static counterpart, because dynamic gates and cells usually use less transistors. Furthermore, the power consumption tradeoff may be more apparent due to its dynamic nature. The only downside is that it is more difficult to design than the static. However, for better performance, it would be a wise move to implement dynamic logic despite its difficulty. Another improvement our team can make in the future is decreasing the size of our full adder. While our design is faster and less power consuming than the synthesized adder, it is still bigger than synthesized. The size could be reduced by using newly fabricated transistors in the future, which would be smaller than the ones available today. Another way to decrease the size would be to use dynamic logic, for the reasons stated before.

Acknowledgments:

Haopei Deng

Le Wang

Yuan Cao

References:

[1] P. Chakali, M.K. Patnala, “Design of High Speed Kogge-Stone Based Carry Select Adder”, IJESE, vol.1, no.4, Feb 2013.

[2] J. M. Rabaey, “Digital Integrated Circuits – A Design Perspective”, New Jersey, Prentice-Hall, 2001.

[3] D. W. Parent, “Kogge Stone Adder Logic Verification”, SJSU.

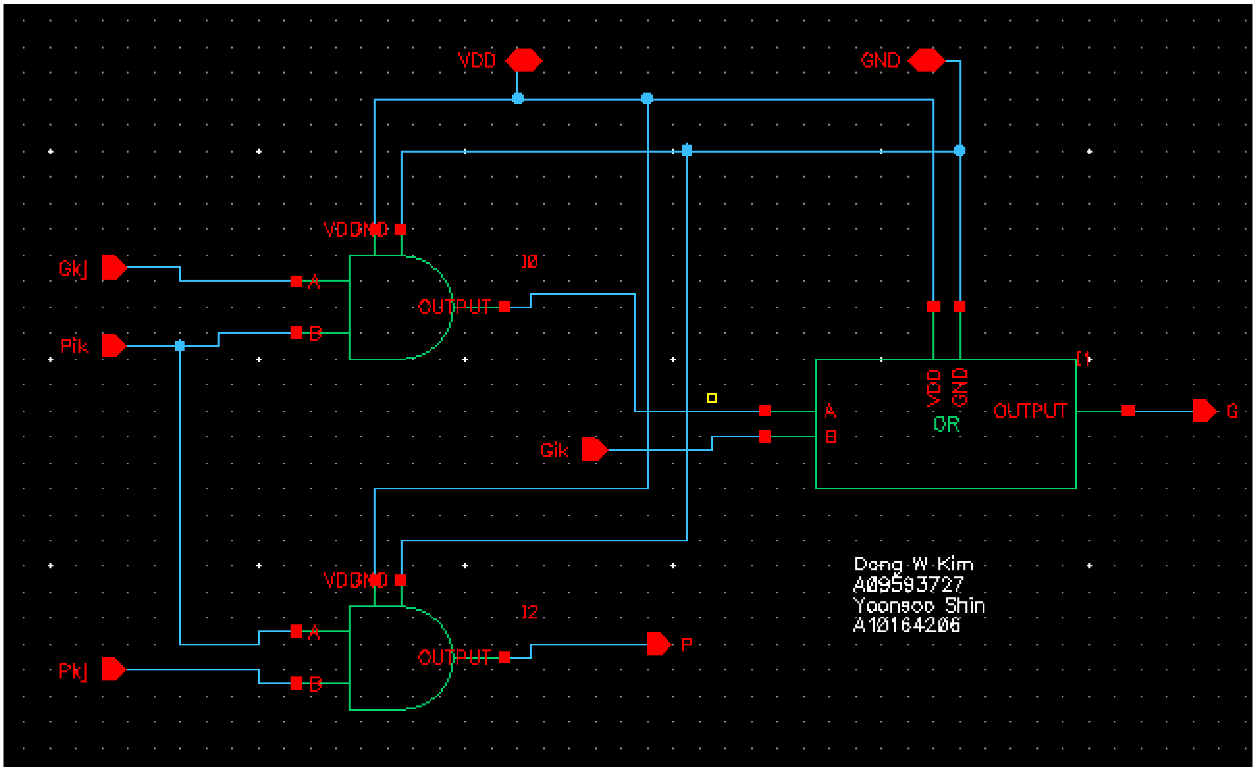
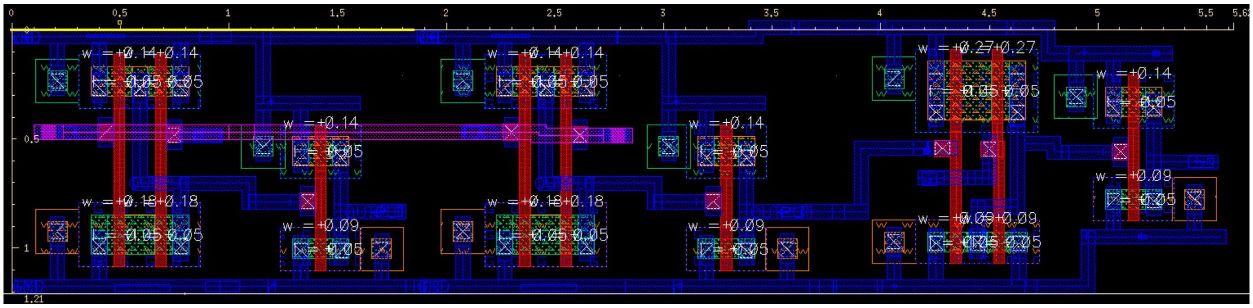
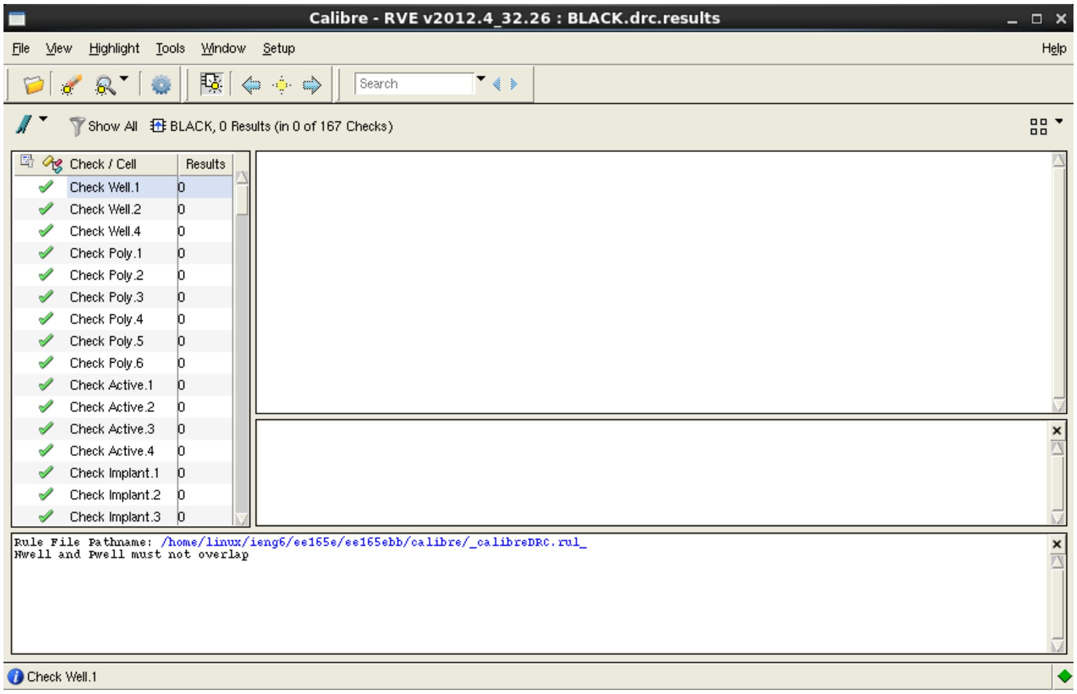


Figure 1 | Black Cell Schematic

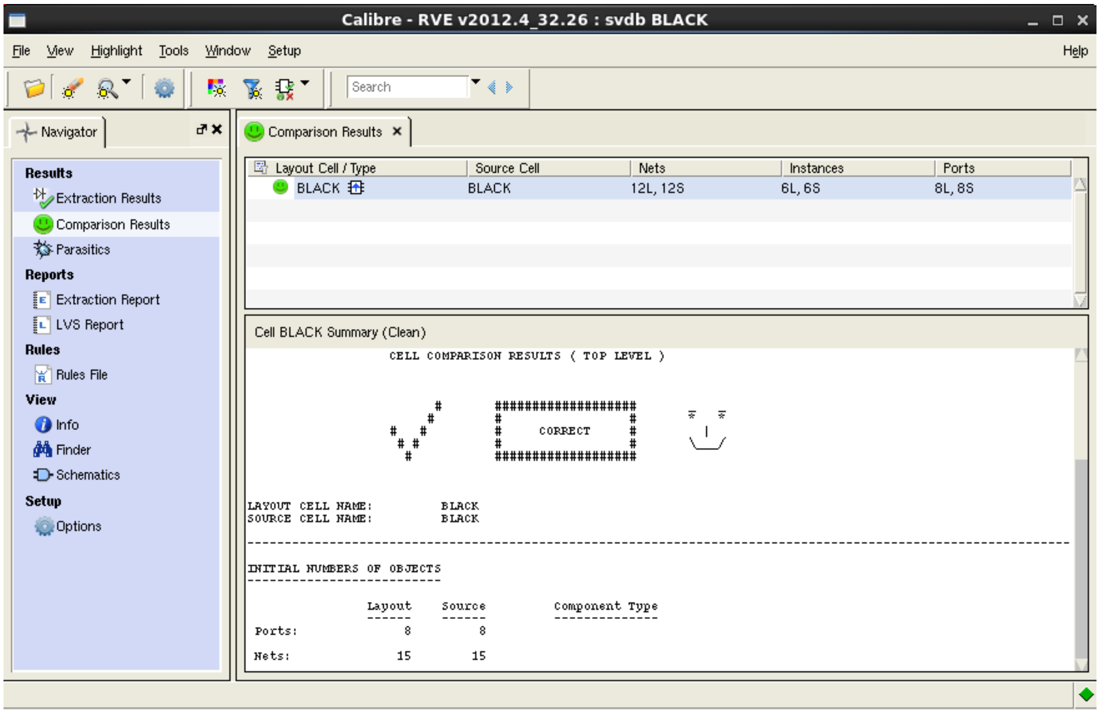



Calibre - RVE v2012.4_32.26 : BLACK.drc.results
File View Highlight Tools Window Setup Help
Search
Show All
BLACK, 0 Results (in 0 of 167 Checks)

Check / Cell	Results
✓ Check Well.1	0
✓ Check Well.2	0
✓ Check Well.4	0
✓ Check Poly.1	0
✓ Check Poly.2	0
✓ Check Poly.3	0
✓ Check Poly.4	0
✓ Check Poly.5	0
✓ Check Poly.6	0
✓ Check Active.1	0
✓ Check Active.2	0
✓ Check Active.3	0
✓ Check Active.4	0
✓ Check Implant.1	0
✓ Check Implant.2	0
✓ Check Implant.3	0

```
Rule File Pathname: /home/linux/ieng6/ee165e/ee165ebb/calibre/_calibreDRC.rul_
Nwell and Pwell must not overlap
```

Check Well.1



Layout Cell / Type	Source Cell	Nets	Instances	Ports
BLACK	BLACK	12L, 12S	6L, 6S	8L, 8S

Cell BLACK Summary (Clean)

```
                    CELL COMPARISON RESULTS ( TOP LEVEL )


                        #       ###################
                       #        #                 #
                  #   #         #     CORRECT     #        \ /
                   # #          #                 #         |
                    #           ###################       \___/


LAYOUT CELL NAME:          BLACK
SOURCE CELL NAME:          BLACK

--------------------------------------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
--------------------------

                 Layout     Source          Component Type
                 ------     ------          --------------
 Ports:               8          8

 Nets:               15         15
```

Figure 2 | Black Cell Layout, Black Cell DRC All Clear Screenshot, Black Cell LVS All Clear Screenshot

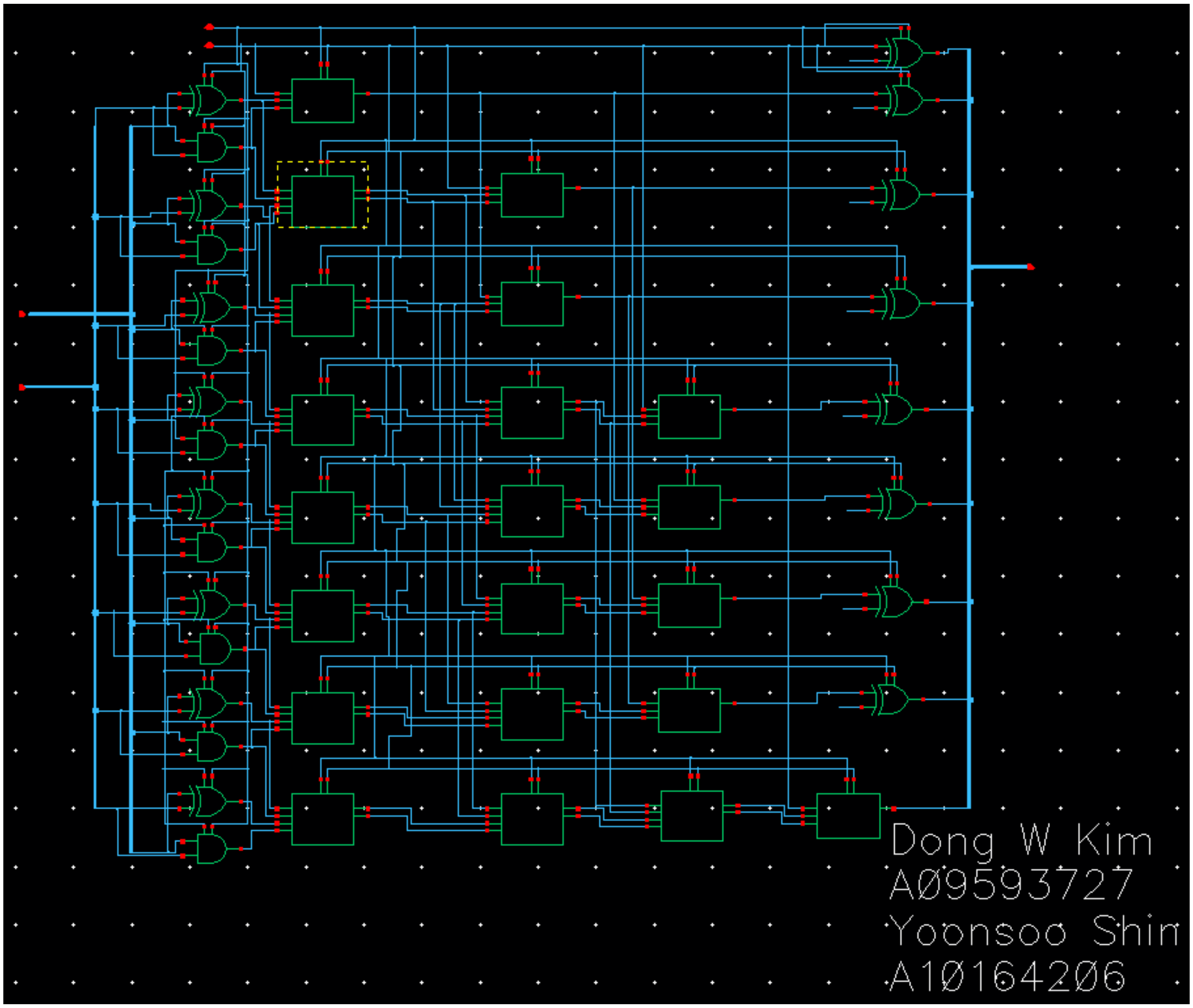
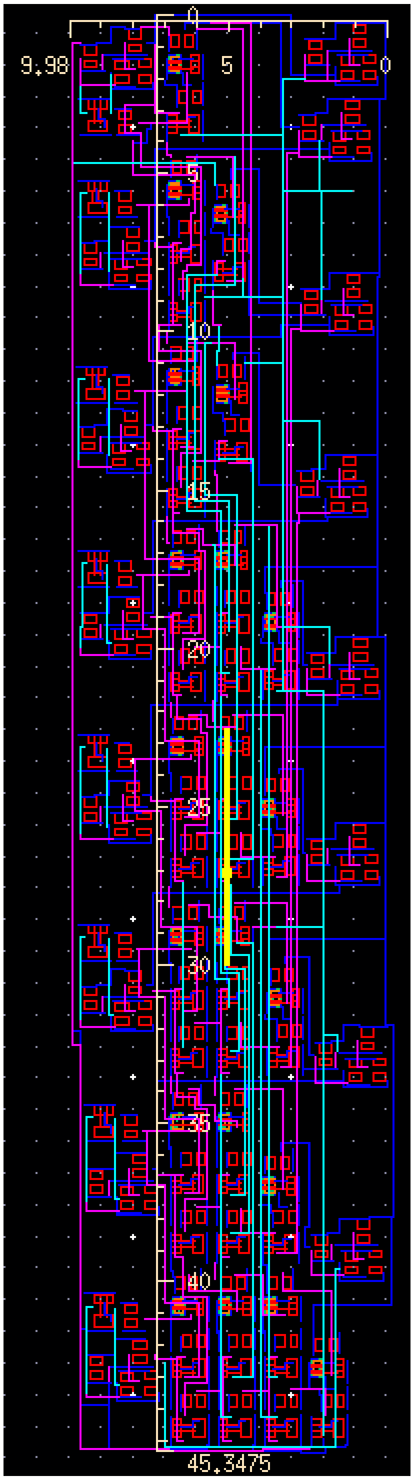
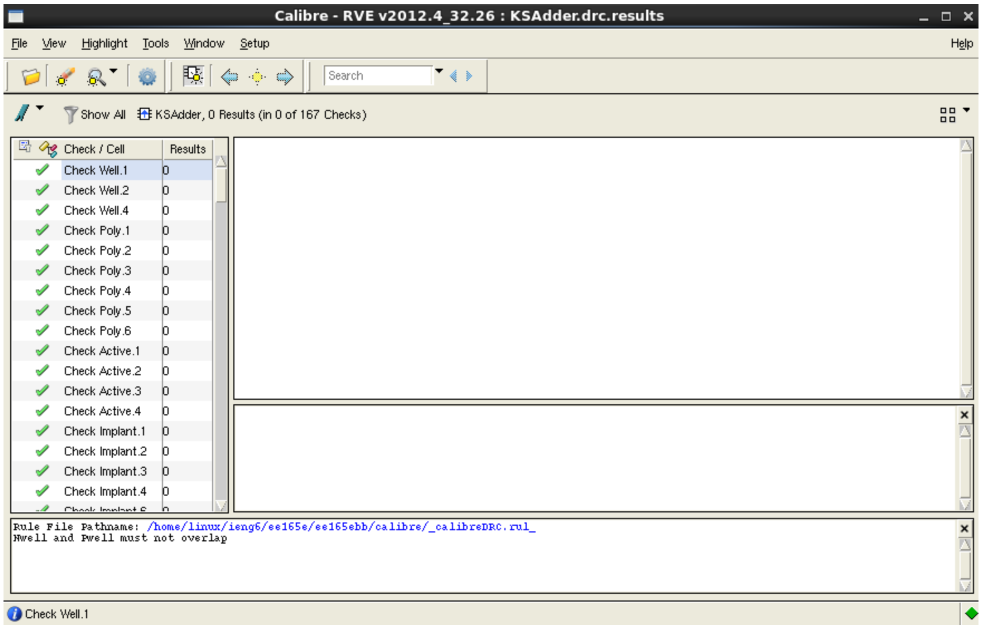


Figure 3 | Schematic of the 8-bit Kogge-Stone Adder



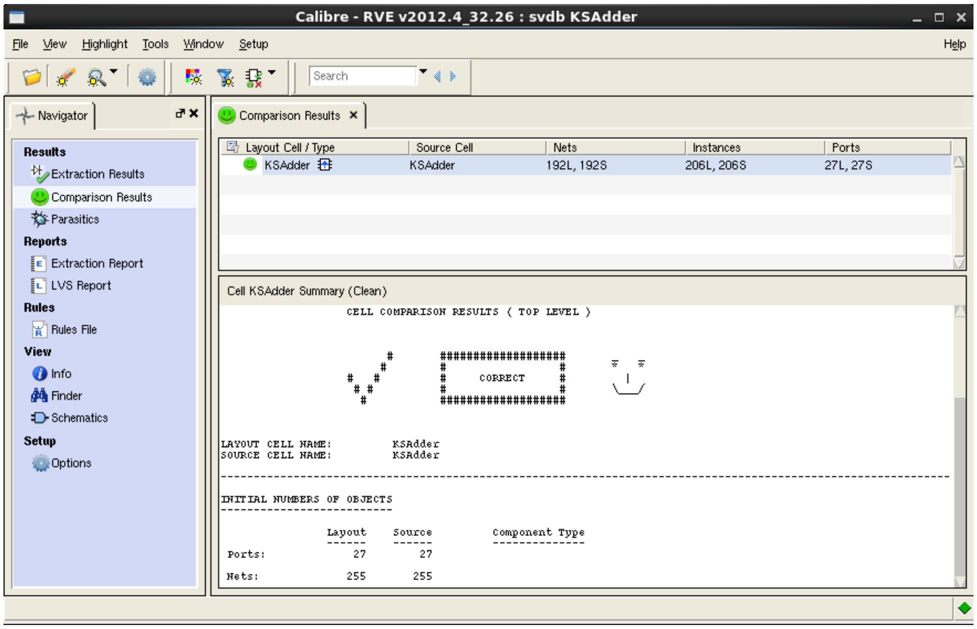

Calibre - RVE v2012.4_32.26 : KSAdder.drc.results
File View Highlight Tools Window Setup Help
Search
Show All

KSAdder, 0 Results (in 0 of 167 Checks)

Check / Cell	Results
✓ Check Well.1	0
✓ Check Well.2	0
✓ Check Well.4	0
✓ Check Poly.1	0
✓ Check Poly.2	0
✓ Check Poly.3	0
✓ Check Poly.4	0
✓ Check Poly.5	0
✓ Check Poly.6	0
✓ Check Active.1	0
✓ Check Active.2	0
✓ Check Active.3	0
✓ Check Active.4	0
✓ Check Implant.1	0
✓ Check Implant.2	0
✓ Check Implant.3	0
✓ Check Implant.4	0

```
Rule File Pathname: /home/linux/ieng6/ee165e/ee165ebb/calibre/_calibreDRC.rul_
Nwell and Pwell must not overlap
```

Check Well.1



Layout Cell / Type	Source Cell	Nets	Instances	Ports
KSAdder	KSAdder	192L, 192S	206L, 206S	27L, 27S

Cell KSAdder Summary (Clean)

```
                    CELL COMPARISON RESULTS ( TOP LEVEL )



                       #        ###################
                      #         #                 #
                    #   #       #     CORRECT     #
                     # #        #                 #
                      #         ###################


LAYOUT CELL NAME:         KSAdder
SOURCE CELL NAME:         KSAdder

--------------------------------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
--------------------------

                 Layout    Source         Component Type
                 ------    ------         --------------
 Ports:              27        27

 Nets:              255       255
```

Figure 4 | 8 bit Kogge-Stone Layout, DRC and LVS All Clear Screenshot

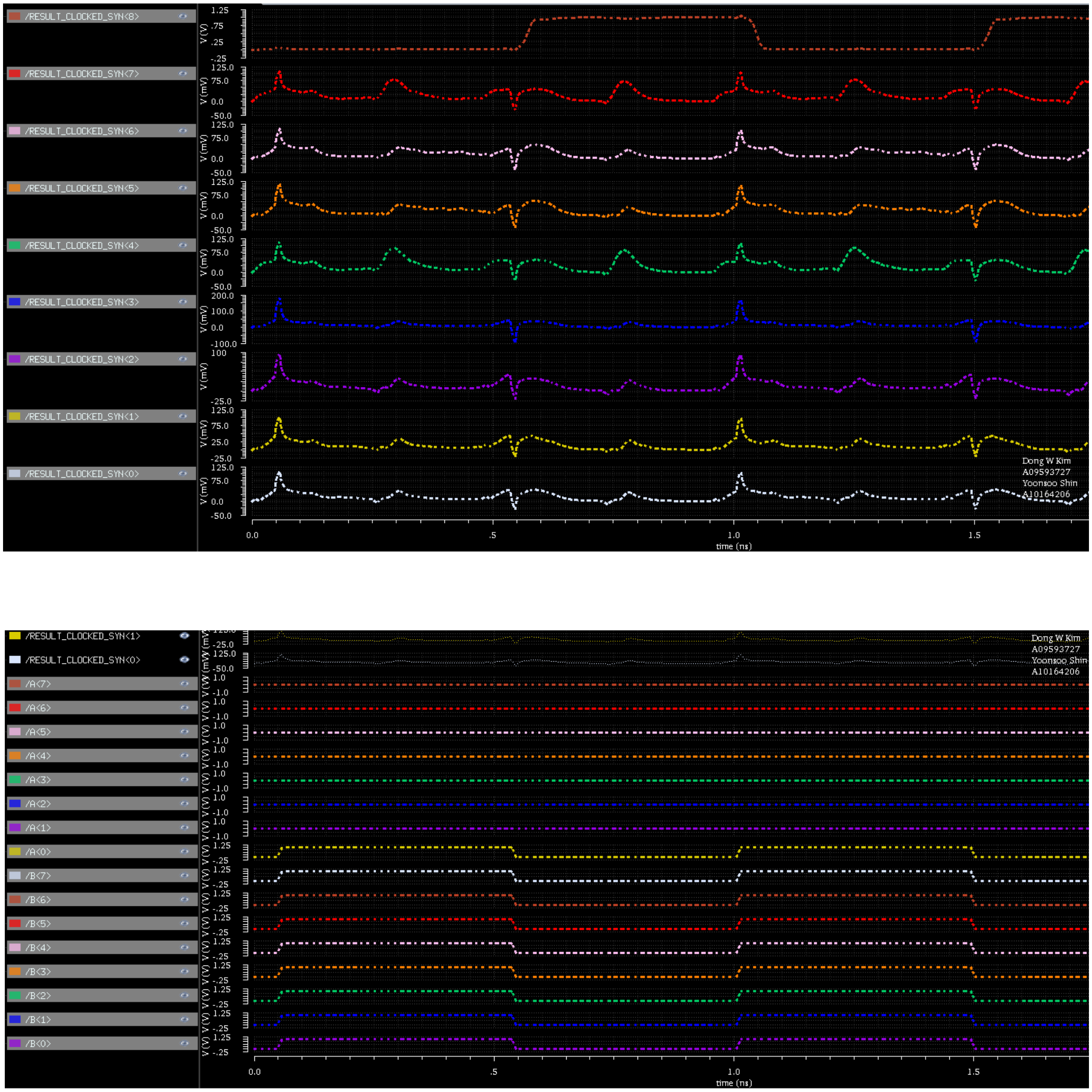


Figure 5 | Transient Simulation of Post-extracted Synthesized Adder (1V @ 2.09GHz), Input A and B

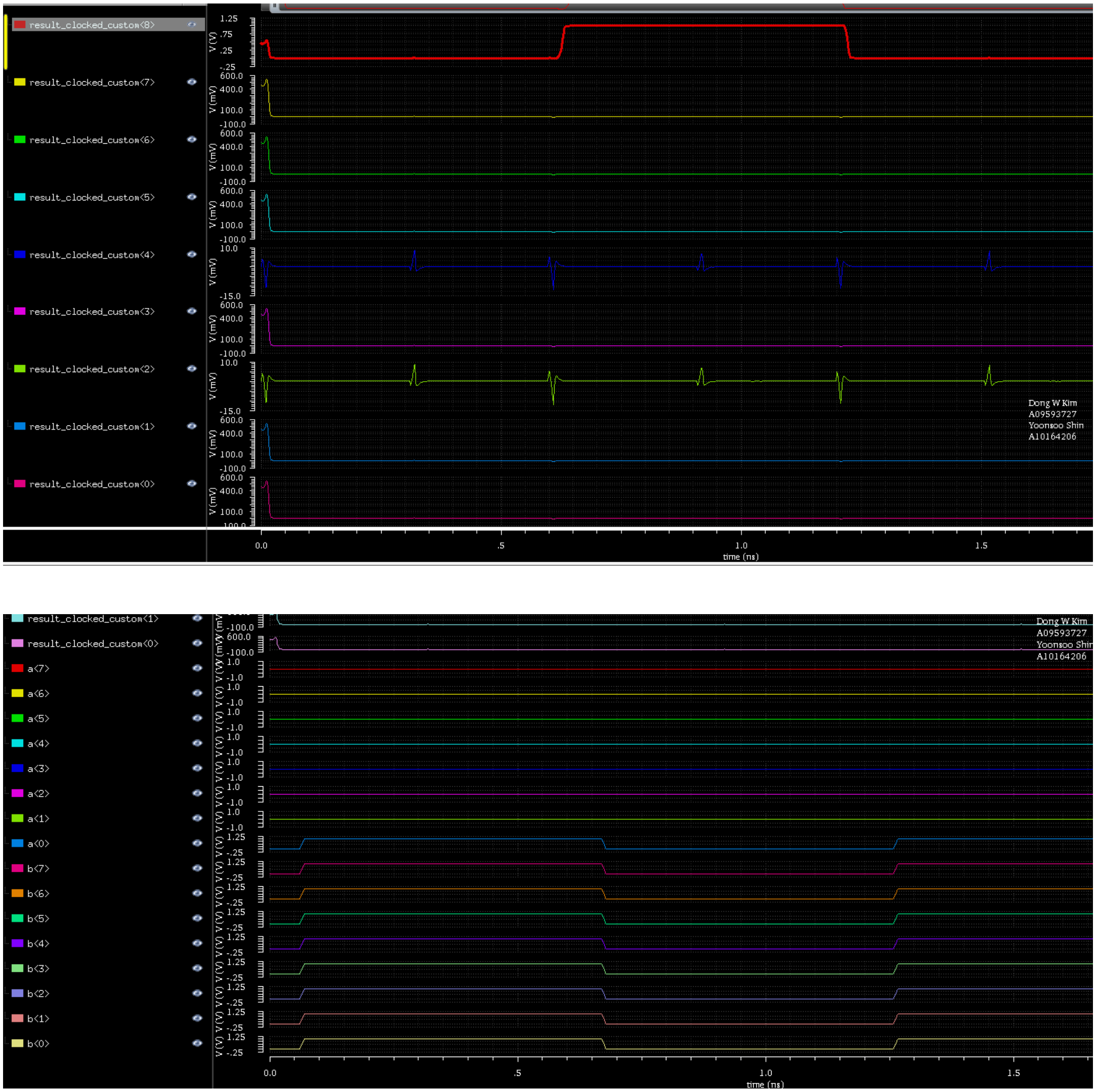
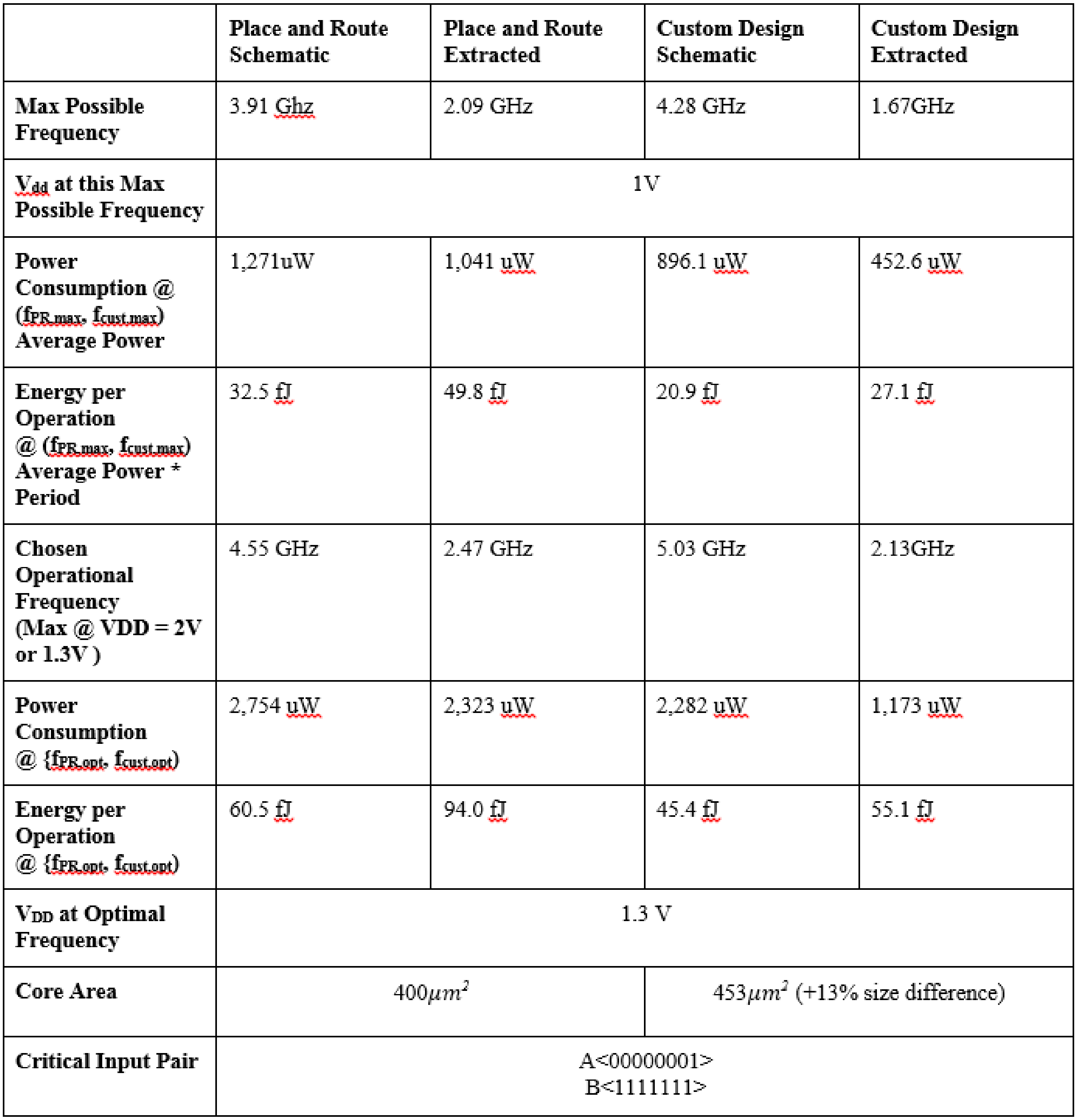


Figure 5 | Transient Simulation of Post-extracted Kogge-Stone Adder (1V @ 1.69GHz), Input A and B



	Place and Route Schematic	Place and Route Extracted	Custom Design Schematic	Custom Design Extracted
Max Possible Frequency	3.91 Ghz	2.09 GHz	4.28 GHz	1.67GHz
V_{dd} at this Max Possible Frequency	1V			
Power Consumption @ ($f_{PR,max}$, $f_{cust,max}$) Average Power	1,271uW	1,041 uW	896.1 uW	452.6 uW
Energy per Operation @ ($f_{PR,max}$, $f_{cust,max}$) Average Power * Period	32.5 fJ	49.8 fJ	20.9 fJ	27.1 fJ
Chosen Operational Frequency (Max @ VDD = 2V or 1.3V)	4.55 GHz	2.47 GHz	5.03 GHz	2.13GHz
Power Consumption @ {$f_{PR,opt}$, $f_{cust,opt}$)	2,754 uW	2,323 uW	2,282 uW	1,173 uW
Energy per Operation @ {$f_{PR,opt}$, $f_{cust,opt}$)	60.5 fJ	94.0 fJ	45.4 fJ	55.1 fJ
V_{DD} at Optimal Frequency	1.3 V			
Core Area	$400\mu m^2$		$453\mu m^2$ (+13% size difference)	
Critical Input Pair	A<00000001> B<1111111>			

Figure 6 | Table of Data

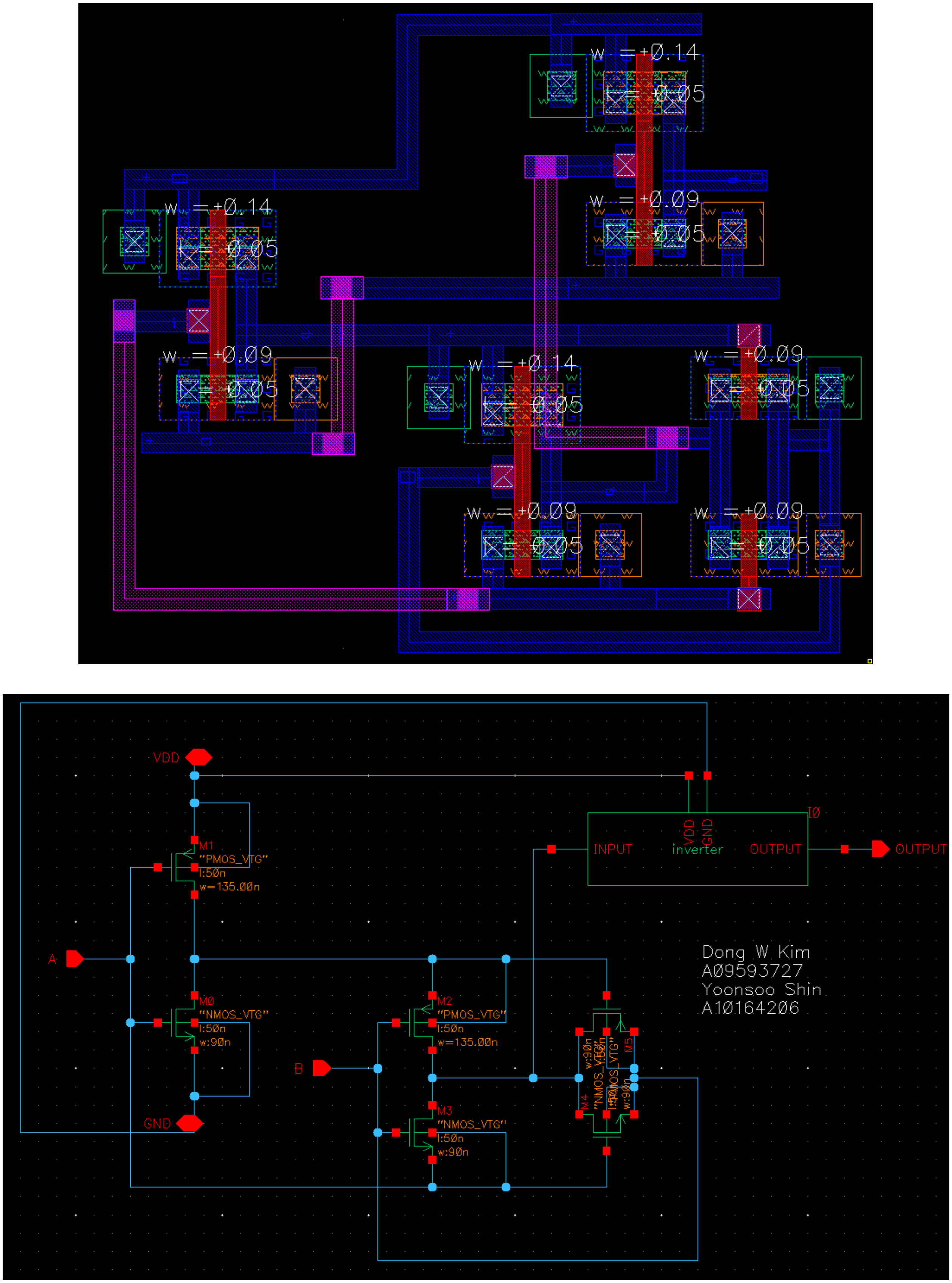


Figure 7 | Optimized XOR Gate Schematics and Layout