Lab 6: Xilinx System Generator Designing a DSP Block

University of Guelph

ENGG*3050 Winter 2016

Professor Shawki Areibi

Graham Thoms 0795111

Vithursan Thangarasa 0821795

Group 6

1 Introduction

The objective of this lab was to learn about the Xilinx System Generator software through implementing a specified design and using the automatic synthesis and simulation tools. A DSP was designed using Simulink and the Xilinx Block-set. The implemented design was verified through simulation and through co-simulation using the FPGA board.

2 Issues with Tutorial

When performing the tutorial "Using Xilinx System Generator 14.6 for Co-Simulation (SOE)" there were no issues that were encountered. The given steps were clear and the correct results were obtained as expected, which helped carry out Lab 6.

3 Design Simulation Using Simulink

3.1 Design 1: Implementation and Simulation

The DSP of design 1 has the following function: $out = (4 * x) + (3.2 * y) - (2.1 * z)$

The Xilinx System Generator block diagram for Design 1 is presented in **Figure 1**, and a simulation of it is presented in **Figure 2**.

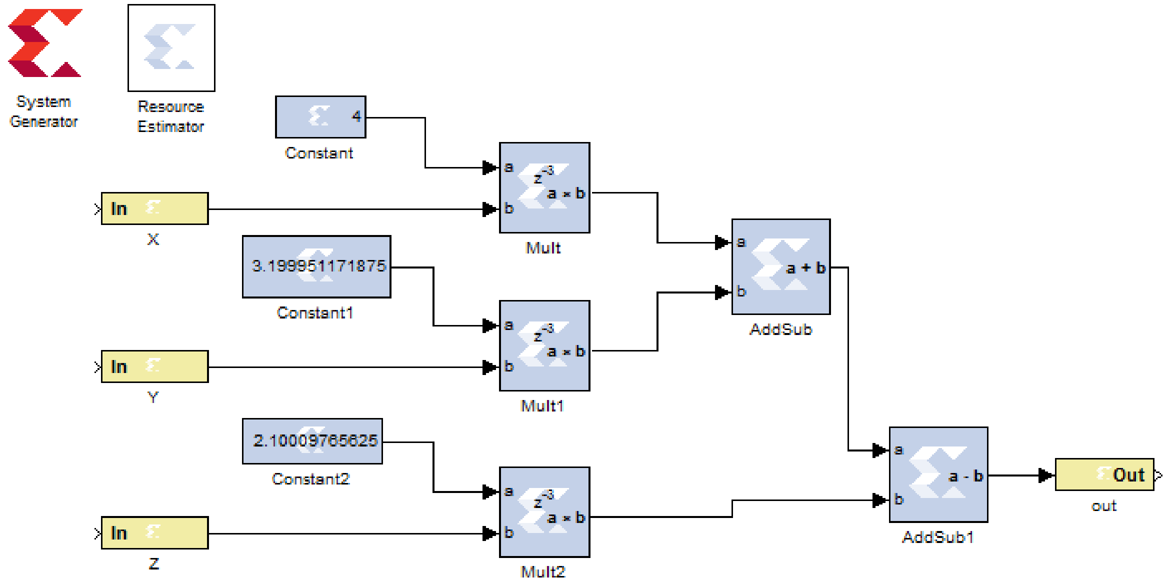


Figure 1 Shows the hardware block implementation of Design 1.

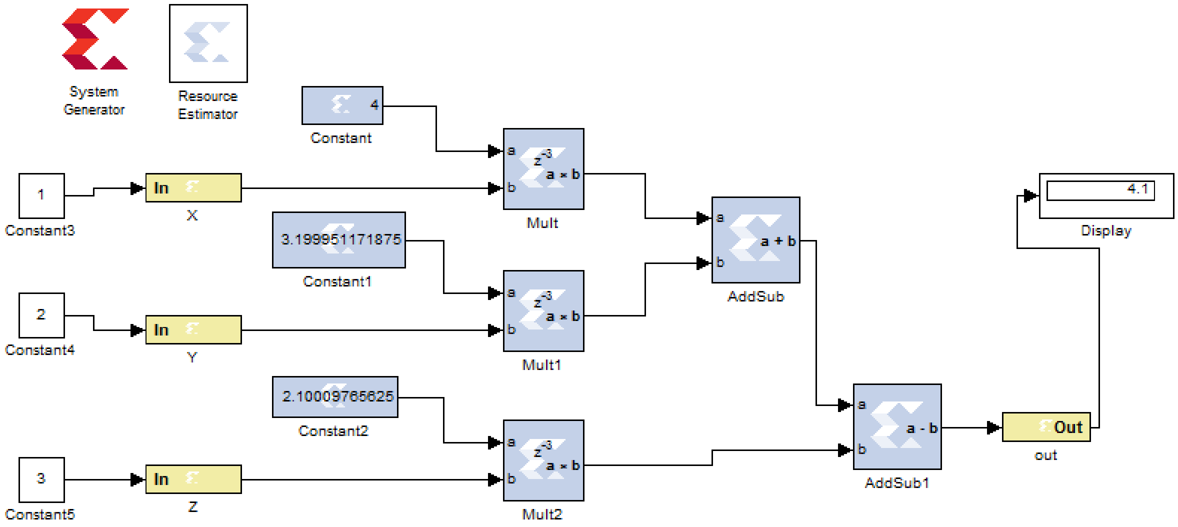


Figure 2 Shows the simulation of Design 1 using simulink; correct result of out=4.1 when x=1, y=2, z=3.

3.2 Design 2: Implementation & Simulation

The DSP of design 2 has the following function:

$$f(x) = \begin{cases} x^3 + 0.5x & |x| \leq 1 \\ 2x - 0.5sgn(x) & |x| \geq 1 \end{cases}$$

The Xilinx System Generator block diagram for Design 1 is presented in **Figure 3**, and a simulation of it is presented in **Figure 4**.

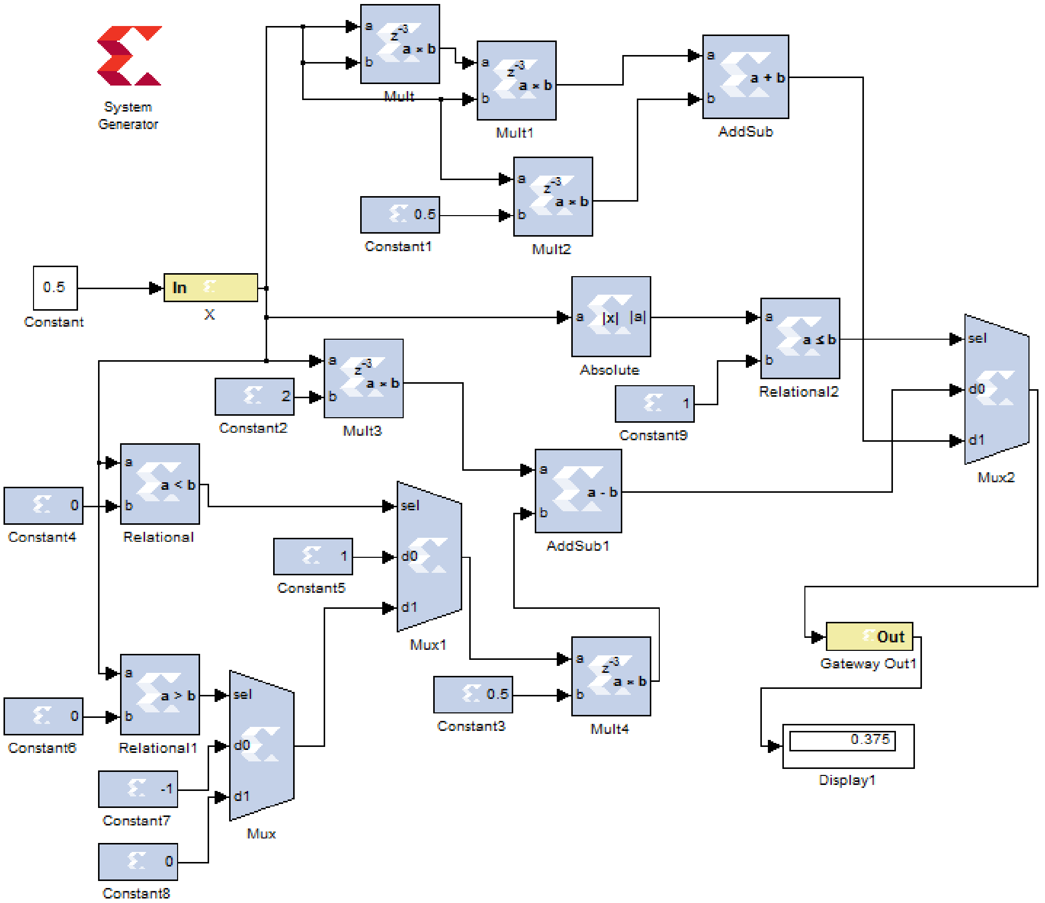


Figure 3 Shows Design 2 implementation with correct results when input abs(x) <= 1 x= 0.5, out = 0.375.

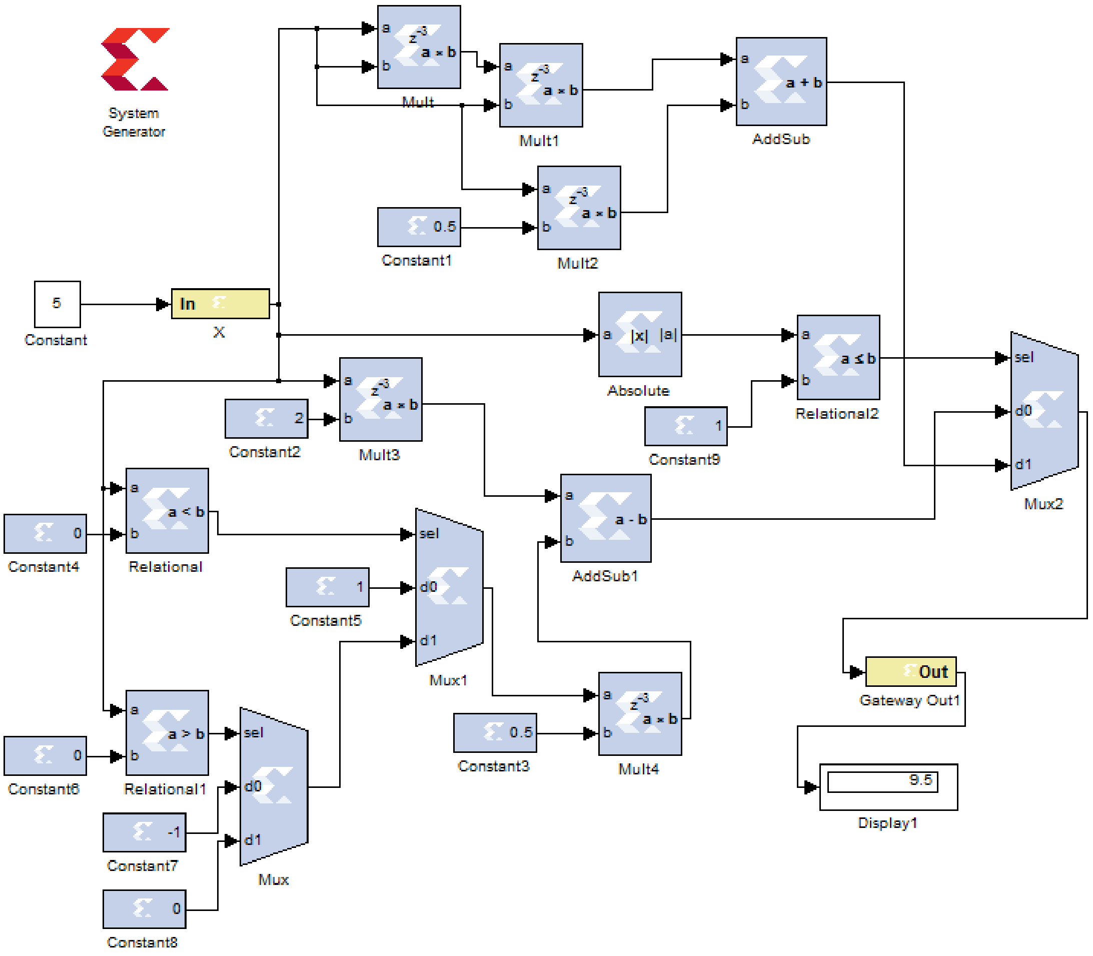


Figure 4 Shows Design 2 implementation with the correct result when input abs(x) > 1, x=5, out=9.5.

4 Co-Simulation Using Spartan 6 FPGA Board

4.1 Design 1: Co-Simulation

The Xilinx System Generator Hardware/Software Co-Simulation block diagram for Design 1 is presented in **Figure 5**.

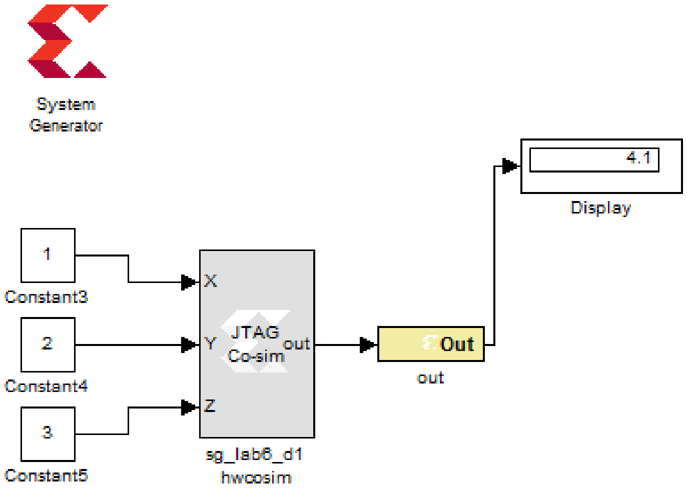


Figure 5 Shows the co-simulation of design 1.

4.2 Design 2: Co-Simulation

The Xilinx System Generator Hardware/Software Co-Simulation block diagram for Design 2 is presented in **Figure 6 and 7**.

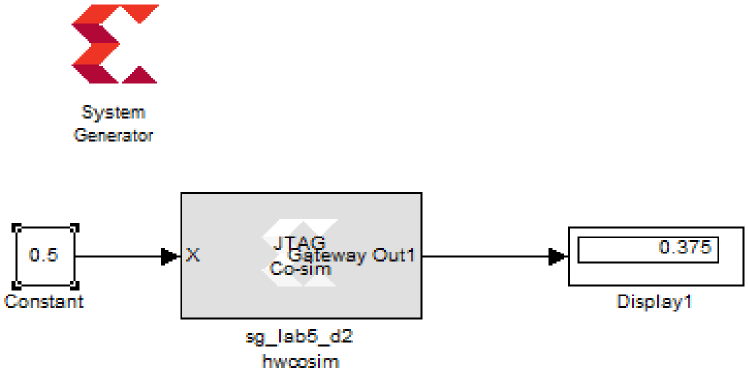


Figure 6 Shows the co-simulation results for design 2 when input abs(x) <= 1.

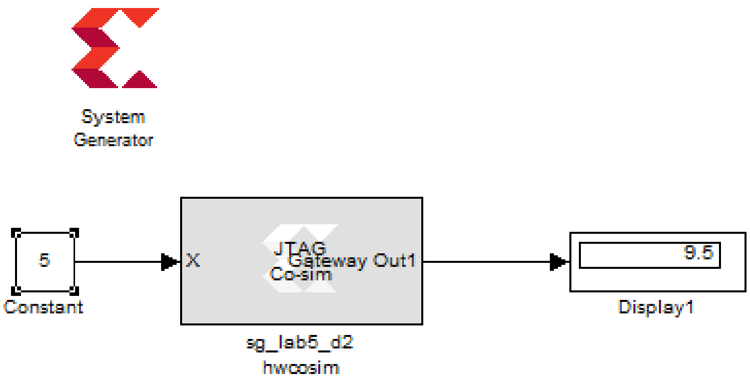
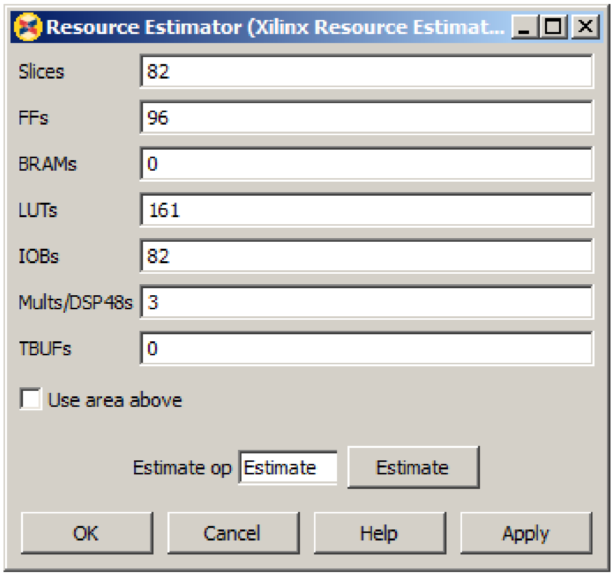


Figure 7 Shows the co-simulation results for design 2 (abs(x) > 1).

4 Simulink Simulation and Co-Simulation Comparisons

4.1 Design 1: Resource Estimation & Comparison

It is observed that the Behavioural HDL implementation uses 82 slices and 161 LUTs (**Figure 8**). The Embedded Multipliers implementation uses 130 slices and 161 LUTs (**Figure 9**). It is to be noted that optimizing Embedded Multiplier implementation for speed or area, the number resources do not change (**Figures 10 and 11**). This is primarily due the size of the design, as it is small, therefore this indicates the tool has already optimized the solution. Also, the optimum pipelining option also uses the same resources 130 slices and 161 LUTs (**Figure 12)**. In essence, optimizing the design for speed/area or optimum pipelining did not have an effect on the number of resources used. However, the generated HDL file uses significantly less slices and LUTs, as it uses 18 Slices and 67 LUTs.



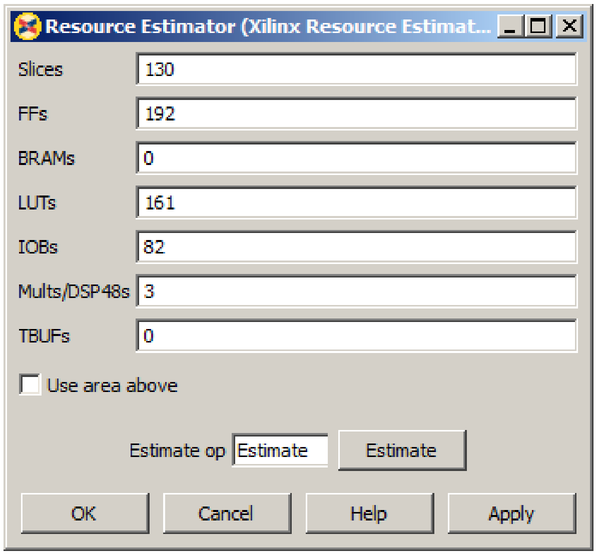
Resource Estimator (Xilinx Resource Estimat...

Slices	82
FFs	96
BRAMs	0
LUTs	161
IOBs	82
Mults/DSP48s	3
TBUFs	0

□ Use area above

Estimate op Estimate

Figure 8 Shows resources using behavioural HDL for multipliers and adders/subtractors - Design 1.

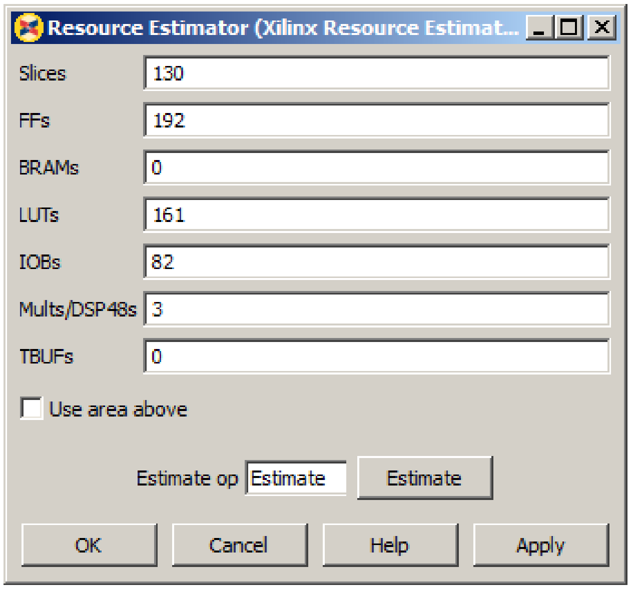


Slices	130
FFs	192
BRAMs	0
LUTs	161
IOBs	82
Mults/DSP48s	3
TBUFs	0

□ Use area above

Estimate op Estimate

Figure 9 Shows the resources used when using Embedded multipliers for Design 1.

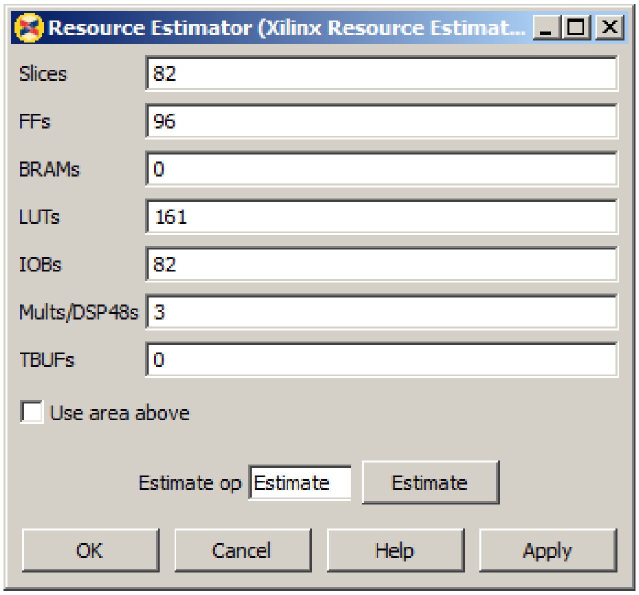


Slices	130
FFs	192
BRAMs	0
LUTs	161
IOBs	82
Mults/DSP48s	3
TBUFs	0

□ Use area above

Estimate op Estimate

Figure 10 Shows resources used in Design 1 when optimizing for speed.

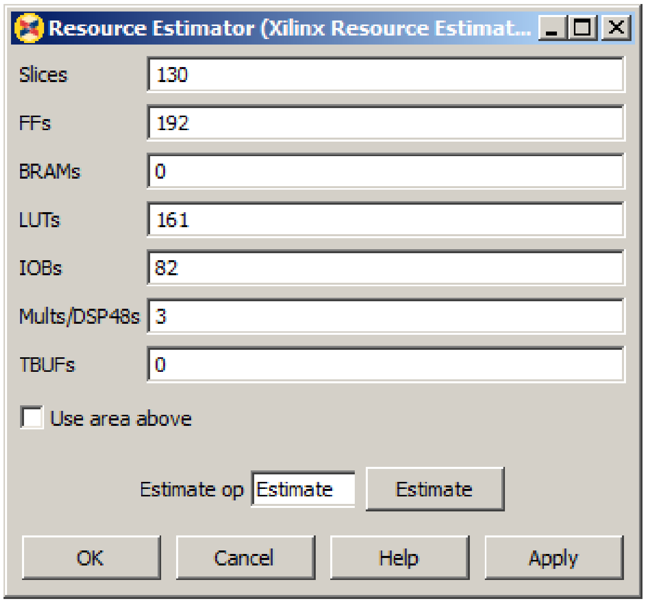


Slices	82
FFs	96
BRAMs	0
LUTs	161
IOBs	82
Mults/DSP48s	3
TBUFs	0

□ Use area above

Estimate op Estimate

Figure 11 Shows resources used in Design 1 when optimizing for area.

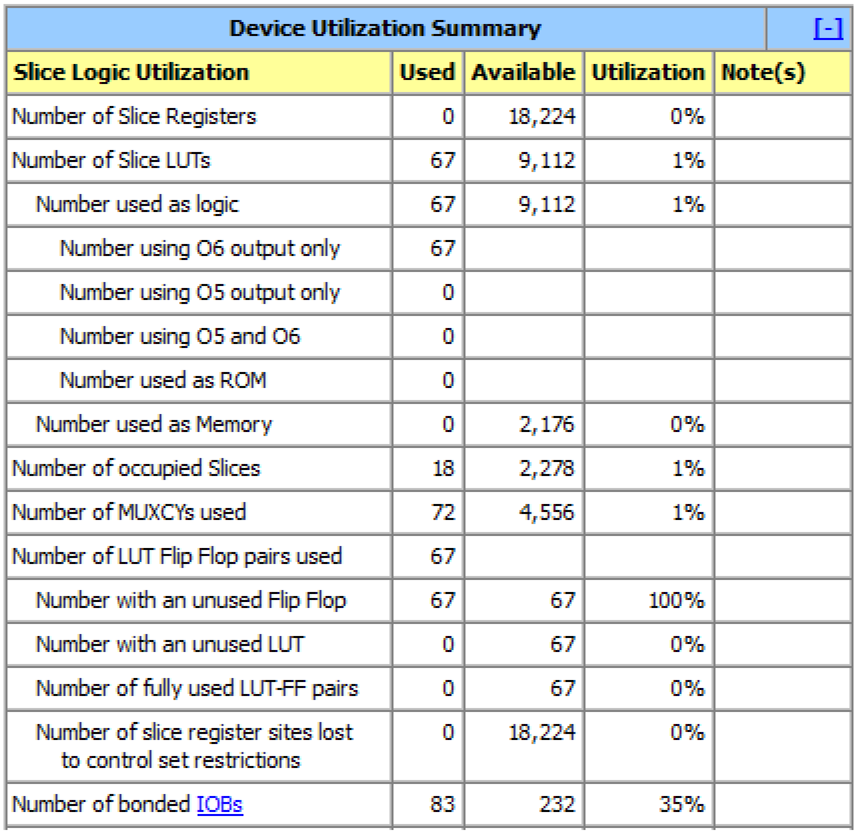


Slices	130
FFs	192
BRAMs	0
LUTs	161
IOBs	82
Mults/DSP48s	3
TBUFs	0

□ Use area above

Estimate op Estimate

Figure 12 Shows resources used in Design 1 when testing for optimal pipelining.

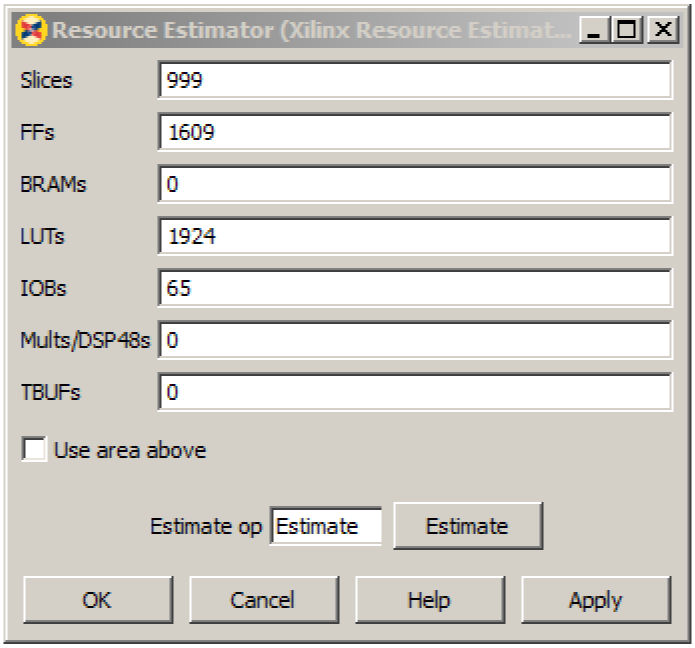


Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	0	18,224	0%	
Number of Slice LUTs	67	9,112	1%	
Number used as logic	67	9,112	1%	
Number using O6 output only	67			
Number using O5 output only	0			
Number using O5 and O6	0			
Number used as ROM	0			
Number used as Memory	0	2,176	0%	
Number of occupied Slices	18	2,278	1%	
Number of MUXCYs used	72	4,556	1%	
Number of LUT Flip Flop pairs used	67			
Number with an unused Flip Flop	67	67	100%	
Number with an unused LUT	0	67	0%	
Number of fully used LUT-FF pairs	0	67	0%	
Number of slice register sites lost to control set restrictions	0	18,224	0%	
Number of bonded IOBs	83	232	35%	

Figure 12 Shows the resources used from the generated Design 1 HDL file.

4.2 Design 2: Resource Estimation

It is observed that the Behavioural HDL implementation uses 999 slices and 1924 LUTs (**Figure 13**). The Embedded Multipliers implementation uses 305 slices and 461 LUTs (**Figure 14**). It is to be noted that optimizing Embedded Multiplier implementation for speed or area, the number resources do not change (**Figures 15 and 16**). This is primarily due the size of the design, as it is small, therefore this indicates the tool has already optimized the solution. It does indeed use more resources than an unoptimized design, with 1005 slices and 1937 LUTs. Also, the optimum pipelining option also uses the same resources 1005 slices and 1937 LUTs (**Figure 17)**. In essence, optimizing the design for speed/area or optimum pipelining increased the number of resources used and is more than the Behaviour HDL implementation. However, as expected, the generated HDL file uses significantly less slices and LUTs, since it uses 53 Slices and 130 LUTs (**Figure 18**).

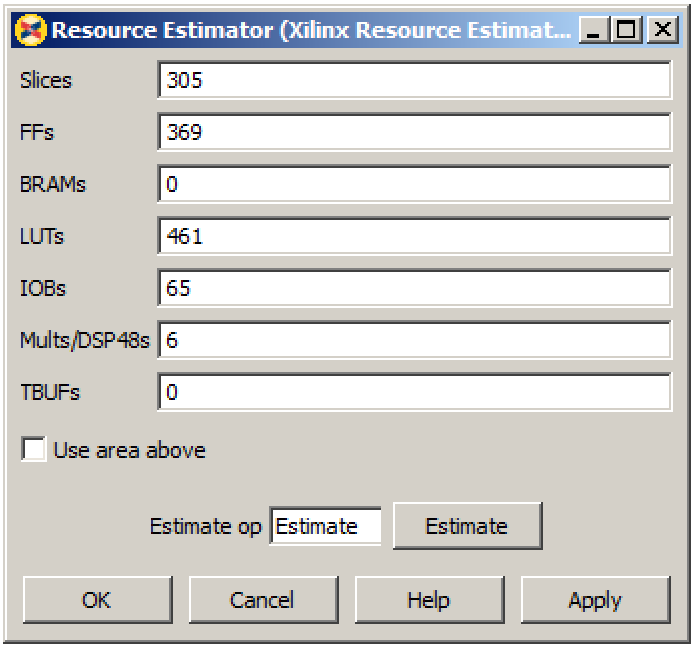


Slices	999
FFs	1609
BRAMs	0
LUTs	1924
IOBs	65
Mults/DSP48s	0
TBUFs	0

□ Use area above

Estimate op Estimate

Figure 13 Shows resources using behavioural HDL for multipliers and adders/subtractors - Design 2.

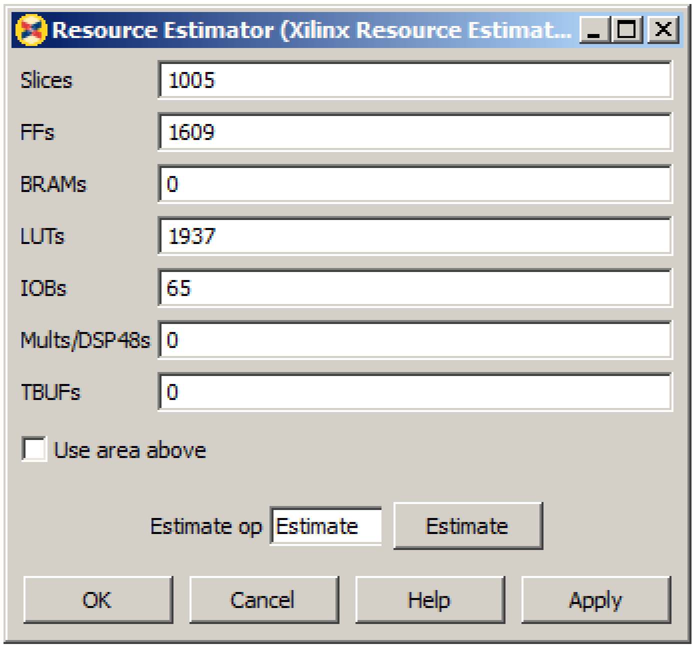


Slices	305
FFs	369
BRAMs	0
LUTs	461
IOBs	65
Mults/DSP48s	6
TBUFs	0

□ Use area above

Estimate op Estimate

Figure 14 Shows the resources used when using Embedded multipliers for Design 2.



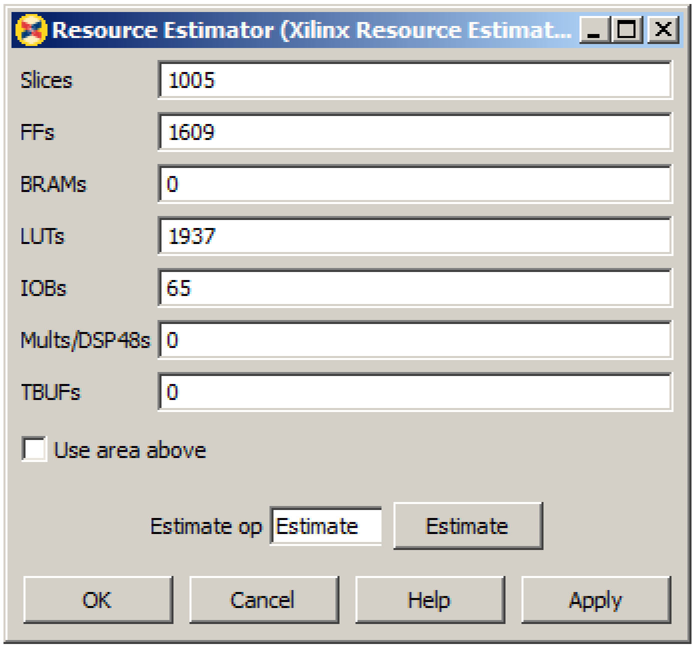
Resource Estimator (Xilinx Resource Estimat...

Slices	1005
FFs	1609
BRAMs	0
LUTs	1937
IOBs	65
Mults/DSP48s	0
TBUFs	0

□ Use area above

Estimate op Estimate

Figure 15 Shows resources used in design 2 when optimizing for speed.



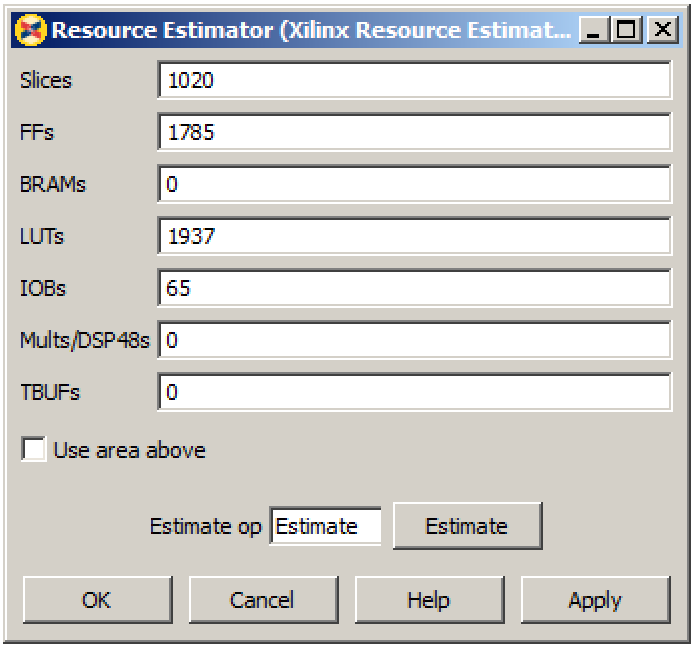
Resource Estimator (Xilinx Resource Estimat...

Slices	1005
FFs	1609
BRAMs	0
LUTs	1937
IOBs	65
Mults/DSP48s	0
TBUFs	0

□ Use area above

Estimate op Estimate

Figure 16 Shows resources used in design 2 when optimizing for area.

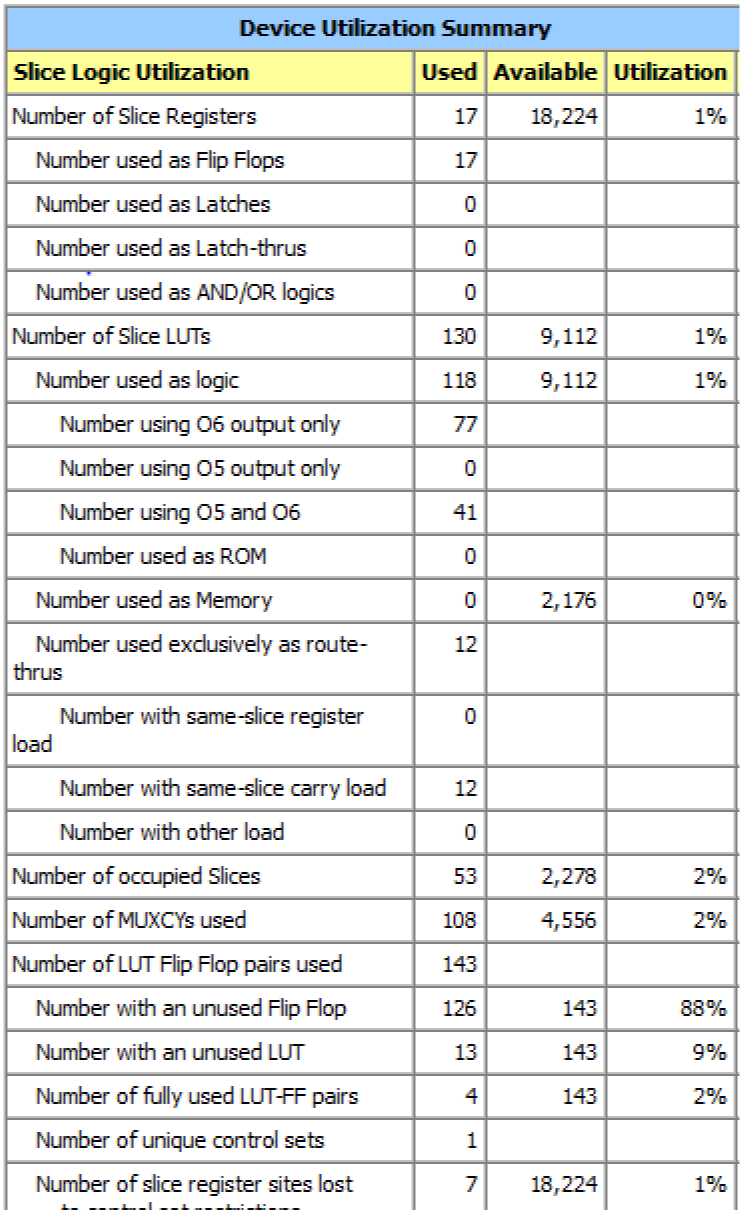


Slices	1020
FFs	1785
BRAMs	0
LUTs	1937
IOBs	65
Mults/DSP48s	0
TBUFs	0

□ Use area above

Estimate op: Estimate

Figure 17 Shows resources used in design 2 when testing for optimal pipelining.



Device Utilization Summary			
Slice Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	17	18,224	1%
Number used as Flip Flops	17		
Number used as Latches	0		
Number used as Latch-thrus	0		
Number used as AND/OR logics	0		
Number of Slice LUTs	130	9,112	1%
Number used as logic	118	9,112	1%
Number using O6 output only	77		
Number using O5 output only	0		
Number using O5 and O6	41		
Number used as ROM	0		
Number used as Memory	0	2,176	0%
Number used exclusively as route-thrus	12		
Number with same-slice register load	0		
Number with same-slice carry load	12		
Number with other load	0		
Number of occupied Slices	53	2,278	2%
Number of MUXCYs used	108	4,556	2%
Number of LUT Flip Flop pairs used	143		
Number with an unused Flip Flop	126	143	88%
Number with an unused LUT	13	143	9%
Number of fully used LUT-FF pairs	4	143	2%
Number of unique control sets	1		
Number of slice register sites lost	7	18,224	1%

Figure 18 Shows the resources used from the generated design 2 HDL file.

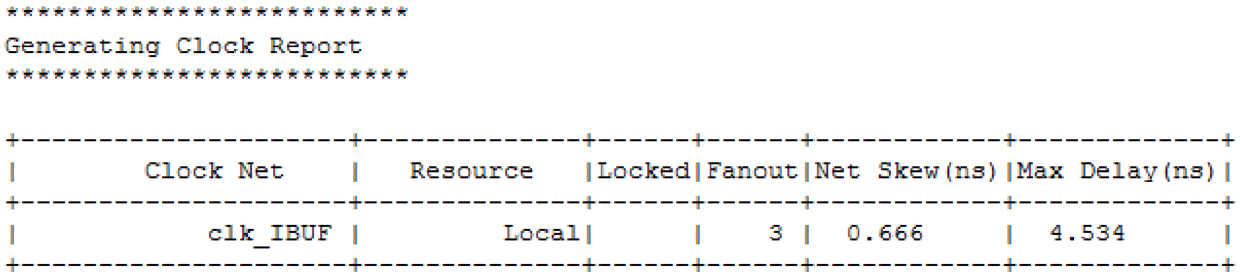
5 Maximum Frequency

The maximum frequencies of the Embedded Multiplier and HDL implementations were calculated using the equation below.

$$Maximum\ Frequency = \frac{1}{(Net\ Skew + Max\ Delay)}\ GHz$$

5.1 Design 1: Maximum Frequency

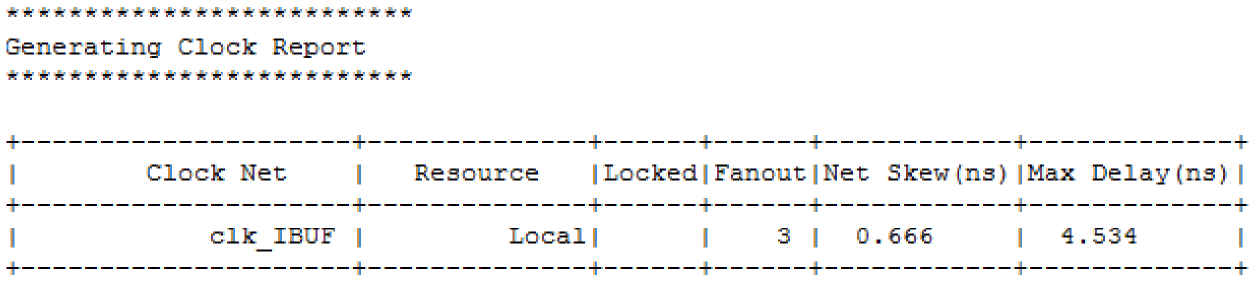
The Net Skews and Max Delays for the respective implementations were determined from the Place and Route Statistics Reports (**Figures 19, 20 and 21**). The Maximum Frequencies of the two different implementations are summarized in the **Table 1**. Here, it can be seen that the Behavioural HDL Option produces a design with a higher frequency (0.889680 GHz) than the Embedded Multiplier Optimized for Speed (0.192307 GHz) and Embedded Multiplier Optimized for Area (0.192307 GHz). It can be noted that the Embedded Multiplier Optimized for Speed and Embedded Multiplier Optimized for Area both have the same maximum frequency. This is because the overall design is not very complex, so the most optimized solution is generated in both scenarios (speed and area).



```
**************************
Generating Clock Report
**************************

+---------------------+--------------+------+------+------------+-------------+
|      Clock Net      |   Resource   |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+---------------------+--------------+------+------+------------+-------------+
|            clk_IBUF |         Local|      |    3 |  0.666     |  4.534      |
+---------------------+--------------+------+------+------------+-------------+
```

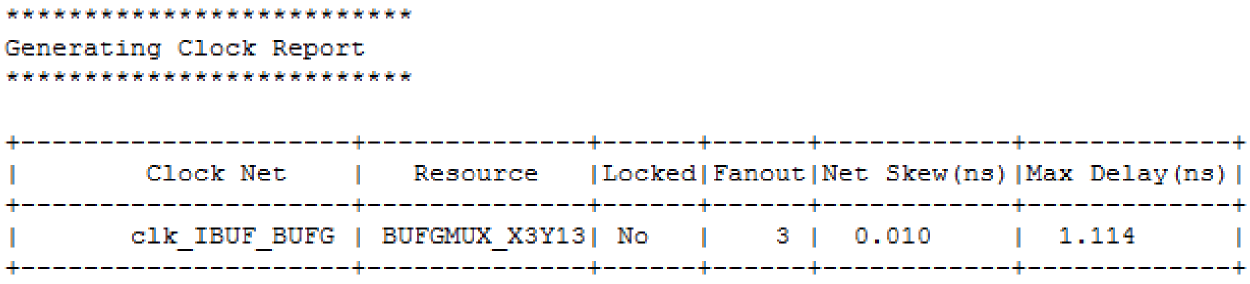
Figure 19 Shows the Place and Route Statistics for Embedded Multiplier (optimized for speed) - Design 1.



```
**************************
Generating Clock Report
**************************

+---------------------+--------------+------+------+------------+-------------+
|      Clock Net      |   Resource   |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+---------------------+--------------+------+------+------------+-------------+
|            clk_IBUF |         Local|      |    3 |  0.666     |  4.534      |
+---------------------+--------------+------+------+------------+-------------+
```

Figure 20 Shows the Place and Route Statistics for Embedded Multiplier (optimized for area) - Design 1.



```
**************************
Generating Clock Report
**************************

+---------------------+--------------+------+------+------------+-------------+
|      Clock Net      |   Resource   |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+---------------------+--------------+------+------+------------+-------------+
|       clk_IBUF_BUFG | BUFGMUX_X3Y13| No   |    3 |  0.010     |  1.114      |
+---------------------+--------------+------+------+------------+-------------+
```

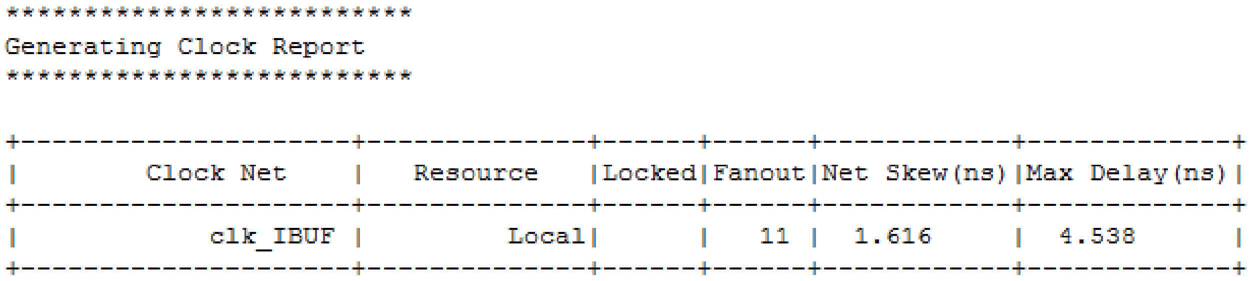
Figure 21 Shows the Place and Route Statistics for Behavioural HDL - Design 1.

Table 1 Shows a summary of the maximum frequency results for implementations - Design 1.

Summary of Maximum Frequencies for Different Multiplier Implementations	
Implementation Option	**Maximum Frequency**
Embedded Multiplier (optimized for speed)	0.192307 GHz
Embedded Multiplier (optimized for area)	0.192307 GHz
Behavioural HDL Option	0.889680 GHz

5.2 Design 2: Maximum Frequency

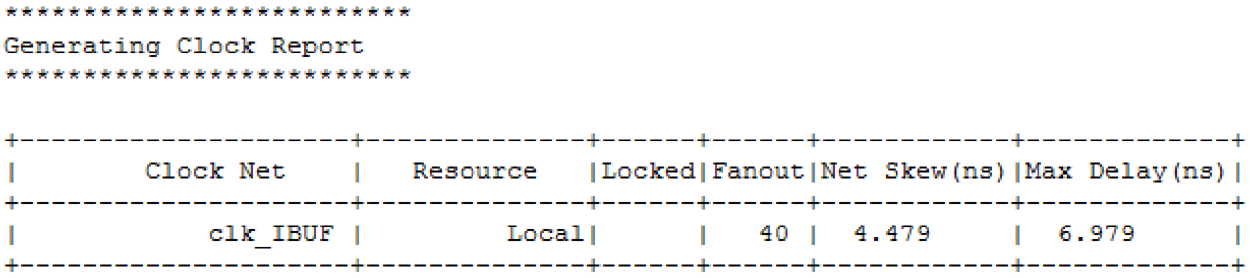
The Net Skew and Max Delay for the respective implementations were determined from the Place and Route Statistics Reports (**Figure 22, 23 and 24**). The Maximum Frequencies of the two different implementations are summarized in the **Table 2**. Similar to Design 1 (Section 6.1), it can be seen that the Behavioural HDL Option produces a design with a higher frequency (0.866551 GHz) than the Embedded Multiplier Optimized for Speed (0.162496 GHz) and Embedded Multiplier Optimized for Area (0.087275 GHz). It can also be observed that the Embedded Multiplier Optimized for Speed generated almost double the maximum frequency than the Embedded Multiplier Optimized for Area.



```
**************************
Generating Clock Report
**************************

+---------------------+---------------+------+------+------------+-------------+
|        Clock Net    |   Resource    |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+---------------------+---------------+------+------+------------+-------------+
|            clk_IBUF |          Local|      |   11 |  1.616     |  4.538      |
+---------------------+---------------+------+------+------------+-------------+
```

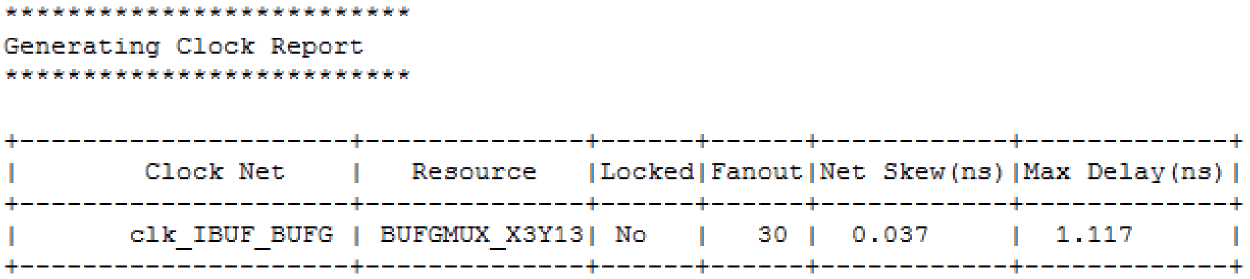
Figure 22 Shows the Place and Route Statistics for Embedded Multiplier (optimized for speed) - Design 2.



```
**************************
Generating Clock Report
**************************

+---------------------+--------------+------+------+------------+-------------+
|      Clock Net      |   Resource   |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+---------------------+--------------+------+------+------------+-------------+
|             clk_IBUF |         Local|      |  40  |  4.479     |  6.979      |
+---------------------+--------------+------+------+------------+-------------+
```

Figure 23 Shows the Place and Route Statistics for Embedded Multiplier (optimized for area) - Design 2.



```
**************************
Generating Clock Report
**************************

+---------------------+--------------+------+------+------------+-------------+
|      Clock Net      |   Resource   |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+---------------------+--------------+------+------+------------+-------------+
|        clk_IBUF_BUFG | BUFGMUX_X3Y13| No   |  30  |  0.037     |  1.117      |
+---------------------+--------------+------+------+------------+-------------+
```

Figure 24 Shows the Place and Route Statistics for behavioural HDL - Design 2.

Table 2 Shows a summary of the maximum frequency results for implementations - Design 2.

Summary of Maximum Frequencies for Different Multiplier Implementations	
Implementation Option	**Maximum Frequency**
Embedded Multiplier (optimized for speed)	0.162496 GHz
Embedded Multiplier (optimized for area)	0.087275 GHz
Behavioural HDL Option	0.866551 GHz

6 Issues with lab

The Embedded Multiplier implementations for Design 1 that were optimized for speed and optimized for area respectively, did not have a change in maximum frequency after Place and Route. In reality, when dealing with more complex systems, a designed expects to see a difference in the performance. Especially, as an entry-level designer learning the tools for Hardware/Software Co-Simulation it is a good learning experience to see the change in different designs. However, this aspect seemed to have defeated the purpose of the lab.

7 Conclusion and Future Considerations

In this lab the Xilinx System Generator software was explored and two DSP designs were implemented on Simulink and simulated both in software and on hardware. The System Generator for DSP allowed for quick creation and implementation of DSP algorithms for FPGAs, because it allowed design entry using hardware blocks and verification via simulation. In addition, this made way for hardware/software co-design of embedded systems by building and debugging DSP co-processors for the Xilinx MicroBlaze soft processor core. In the future, it may be suggested that students be required to implement a single more complex DSP system and compare the different implementations, in order to better observe the differences in performance among different designs. This will provide students with an even more clear understanding of optimizing for speed, area using embedded multipliers and behavioural HDL solutions.