Embedded Computer Architecture 2

Riccardo Battistig s1716239, Pietro Pennestrì s2382660

February 1, 2020

Contents

1 Parallel FIR filter with N=6

The Finite Impulse Response (FIR) filter was realized in Clash using higher order functions of zipWith and foldl. It is a direct implementation of the figure given. The longest combinatory path is the first element of the vector xs, namely the one multiplied at the beginning by h1, as it will have to pass the entire chain of adders to the output z. This can be seen in the RTL schematic shown in figure 1, where the critical path is the highest path. Additionally, the flow summary (figure 2) shows a usage of 6 DSP blocks, as each DSP block contains the possibility to compute a multiplication as well as an addition.

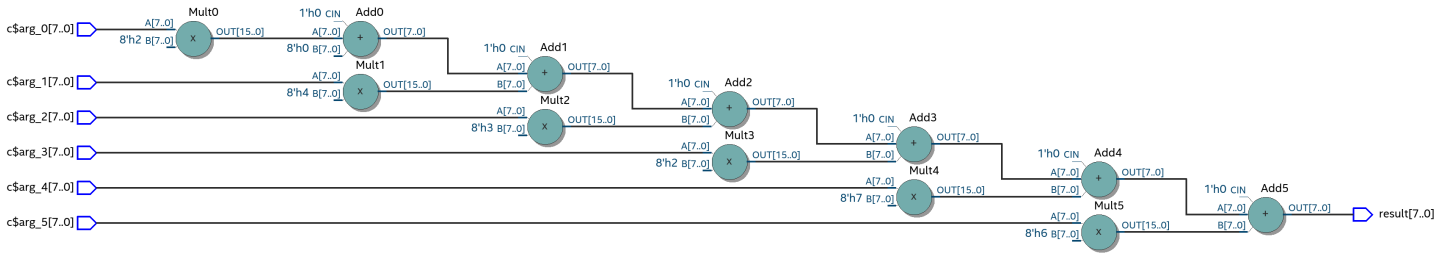

Figure 1: RTL Schematic of the Parallel FIR Filter with $N = 6$

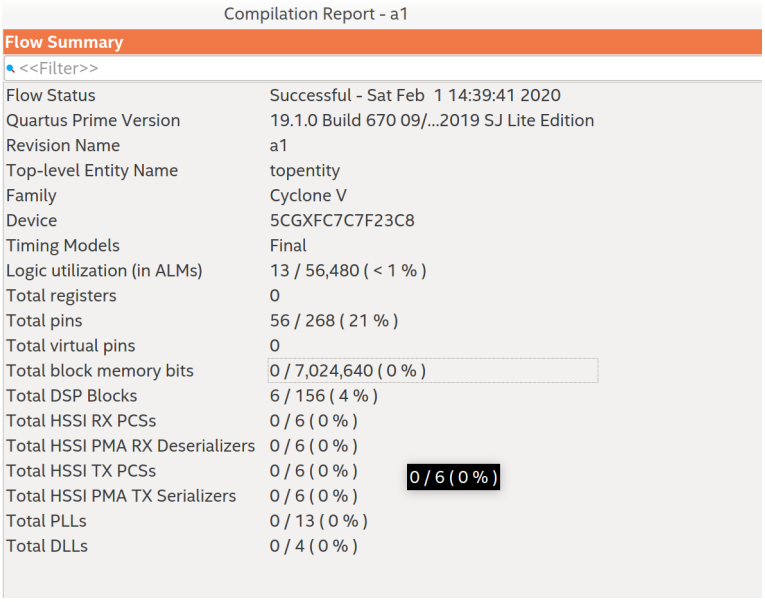

Figure 2: Flow Summary of the Parallel FIR Filter with $N = 6$

2 A Larger Parallel FIR Filter

The larger filter is in fact very similar to figure 1, but with more parallel inputs. For this assignment we refer to the figure 1 which has the same structure as the one for the larger case. The flow summary can be found in figure 3. As expected the DSP block usage are 100 blocks, which is the same ratio of DSP blocks to inputs as the smaller parallel FIR filter, one for each input. The critical path in this case is still the initial vector value multiplied by h0, and in this case passes 99 additional adders.

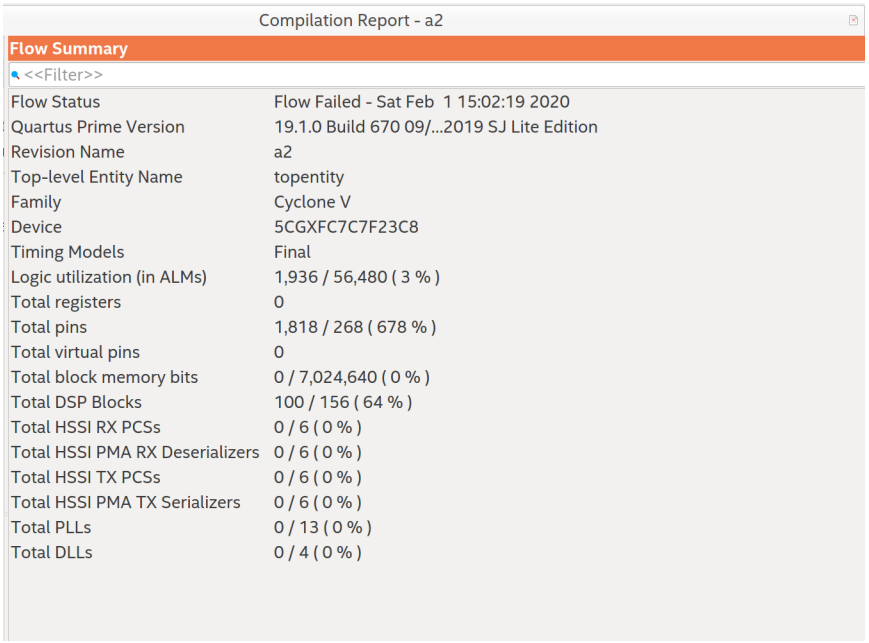

Figure 3: Flow Summary of the larger Parallel FIR Filter with a larger input array

3 FIR filter with N=6

The difference between this FIR filter and the one of assignment 1 is that this filter has a shift register input. The Clash code is nearly the same, besides the definition of the shift registers i.e. the states (memory) which is used. The RTL schematic can be seen in figure 4, as well as the flow summary in 5. Notice that the RTL schematic has the same DSP structure as the parallel filter, as well as the same critical path. By comparing the flow summary with the parallel FIR case, it can be seen that the only difference is in the amount of registers used is higher in the shift register implementation, as well as the total amount of pins used being smaller (necessarily as less inputs are used).

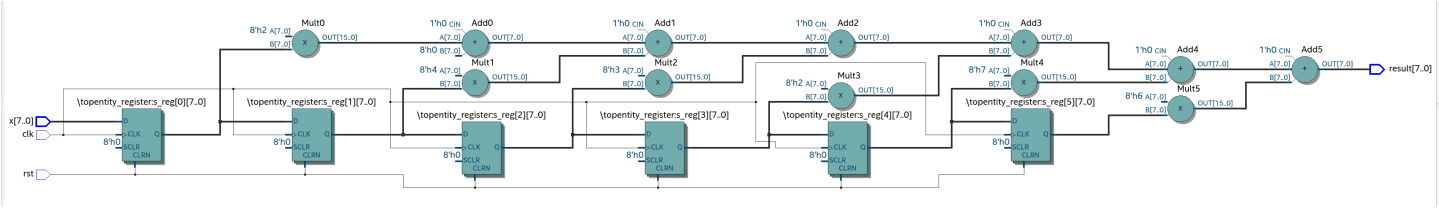

Figure 4: RTL Schematics of the FIR Filter with $N = 6$

Compilation Report - a3

Flow Summary

<<Filter>>

Flow Status	Successful - Sat Feb 1 15:14:38 2020
Quartus Prime Version	19.1.0 Build 670 09/...2019 SJ Lite Edition
Revision Name	a3
Top-level Entity Name	topentity
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	22 / 56,480 (< 1 %)
Total registers	61
Total pins	18 / 268 (7 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	6 / 156 (4 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

Figure 5: Flow Summary of the FIR Filter with $N = 6$

4 A larger FIR filter

Similar to assignment 2, a larger FIR implementation was constructed. The scaling was quite trivial from the smaller FIR implementation, as higher order functions were already used. The RTL schematic therefore has the same structure as the previous exercise and will not be shown (as it is very large). Instead the flow summary is given in figure 6. From the flow summary one can see the resemblance to figure 5. The ratio of DSP to inputs is the same, as well as the pin amounts, as the input is only one in both cases. The amount of registers however is higher in this larger FIR implementation. The critical path remains the same path of the first coefficient and a chain of adders, however is obviously longer in this case compared to the small FIR filters.

Compilation Report - a4

Flow Summary

<<Filter>>

Flow Status	Successful - Sat Feb 1 17:56:59 2020
Quartus Prime Version	19.1.0 Build 670 09/...2019 SJ Lite Edition
Revision Name	a4
Top-level Entity Name	topentity
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	2,079 / 56,480 (4 %)
Total registers	2071
Total pins	38 / 268 (14 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	100 / 156 (64 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

Figure 6: Flow Summary of the FIR Filter with a Larger Input

In order to create the VHDL the simulation the Clash code in list 1 was added. The simulation was ran in Clash using the command also shown in the bottom of the listing. The result of the filtered input is shown in the plot in figure **??**.

```
fir2_100Sim :: HiddenClockResetEnable dom  => Signal dom Value1 -> Signal dom Value1
fir2_100Sim = mealy fir2_100 init_coef100

simulate @System fir2_100Sim inputSignal
```

Listing 1: Simulation Clash Code of FIR filter

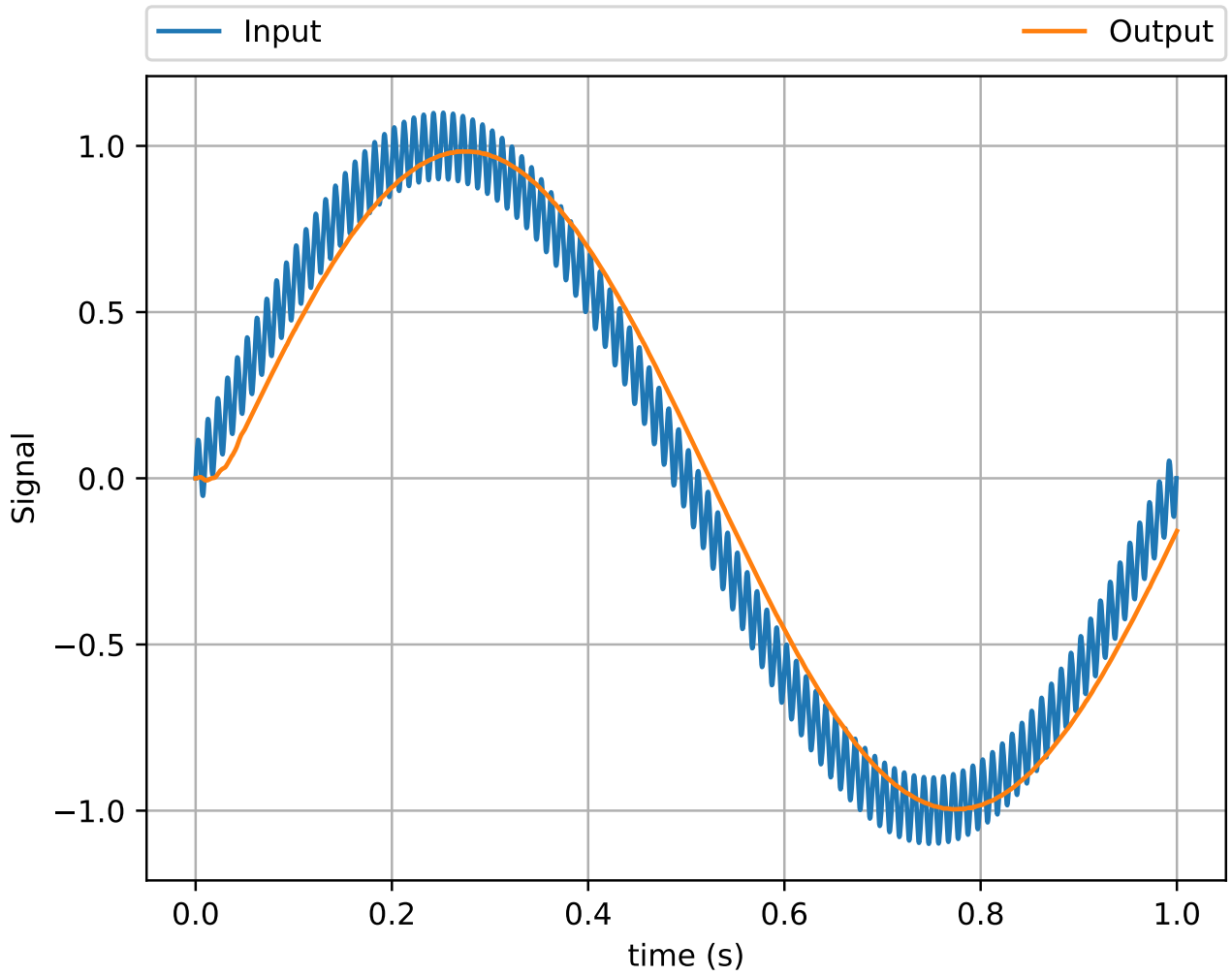

Figure 7: Plot of the Resulting Output Of the larger FIR Filter

5 Symmetric FIR filter with N = 6

In this case the equation of the symmetric FIR filter is given in: 1

$$h_0x_0 + h_1x_1 + h_2x_2 + h_2x_3 + h_1x_4 + h_0x_5 \tag{1}$$

Using factorization this reduces to the equation in equation 2.

$$h_0(x_0 + x_5) + h_1(x_1 + x_4) + h_2(x_2 + x_3) \tag{2}$$

This factorization was implemented in Clash and the following RTL schematic in figure 8 and the flow graph in 9. The flow graph shows that indeed 3 DSP blocks were used, and looking at the RTL schematic the other constraints can be validated. The system takes only 1 clock cycle as only 3 multiplications have to be done for one set of inputs. The critical path is greatly reduced as we can exploit the parallelism and get length of only 1 multiplier and 3 adders, namely the bottom path in the RTL schematic.

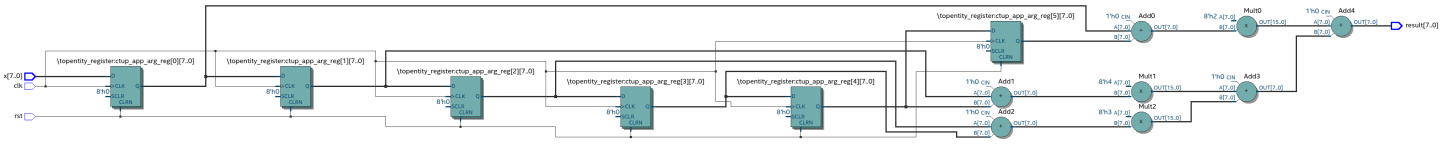

Figure 8: RTL Schematics of the Symmetric FIR Filter with $N = 6$

Compilation Report - a5

Flow Summary

<<Filter>>

Flow Status	Successful - Sat Feb 1 18:20:59 2020
Quartus Prime Version	19.1.0 Build 670 09/...2019 SJ Lite Edition
Revision Name	a5
Top-level Entity Name	topentity
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	27 / 56,480 (< 1 %)
Total registers	59
Total pins	18 / 268 (7 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	3 / 156 (2 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

Figure 9: Flow Summary of the Symmetric FIR Filter with $N = 6$

6 Symmetric FIR filter

As mentioned the coefficients and inputs were changed in order to accommodate the question requirements. The RTL schematic (as seen in figure 19 in the appendix) are once again omitted as it's simply a blown up version of the schematic in figure 8. The combinational path is thus similarly a multiplier followed by the entire chain of additions (foldl or in this case: sum). The throughput is 1 ouput every clock cycle.

Compilation Report - a6

Flow Summary

<<Filter>>

Flow Status	Successful - Sat Feb 1 18:53:17 2020
Quartus Prime Version	19.1.0 Build 670 09/...2019 SJ Lite Edition
Revision Name	a6
Top-level Entity Name	topentity
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	1,970 / 56,480 (3 %)
Total registers	1968
Total pins	38 / 268 (14 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	50 / 156 (32 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

Figure 10: Flow Summary of the Symmetric FIR Filter for a Large Input

In order to create the VHDL the simulation the Clash code in list 2 was added. The simulation was ran in Clash using the command also shown in the bottom of the listing. The result of the filtered input is shown in the plot in figure **??**.

```
1 fir3_100Sim:: HiddenClockResetEnable dom  => Signal dom Value1 -> Signal dom ( Value1 )
2 fir3_100Sim = mealy fir3_100 (replicate d100 0)
```

```
3
4 simulate @System fir3_100Sim inputSignal
```

Listing 2: Simulation Clash Code of Symmetric FIR Filter

7 Transformed symmetric FIR filter with N = 6

The transformations were applied to the FIR. The resulting signal flow graph (SFG) is shown in figure 11. From the SFG it can already be deducted that the critical path is only two operations long, where all paths are of the same length, namely an addition and multiplication. This means that the critical path is shorter than in figure 8. However in the transformed symmetric FIR there is less reusing of partial products which results in a greater amount of DSP blocks used in the transformed version. Both designs have the same throughput of one output per clock cycle. The RTL schematic can be seen in figure 12 as well as the flow summary in figure 13.

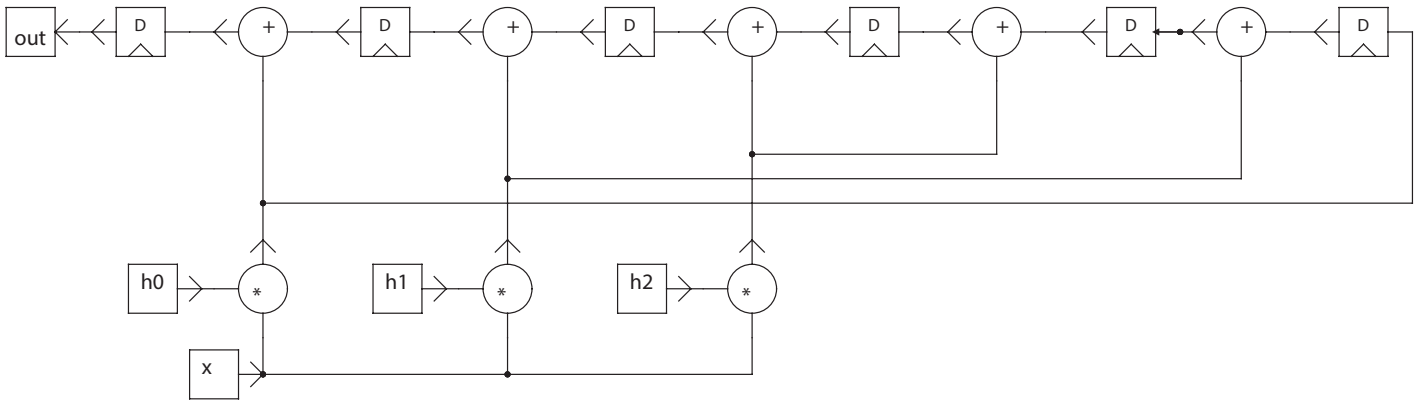

Figure 11: SFG of Transformed FIR Filter for N = 6

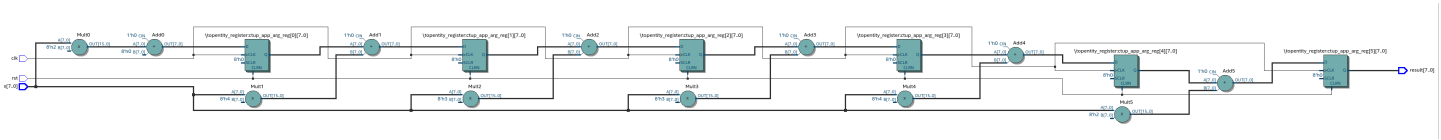

Figure 12: RTL Schematics of the Transformed Symmetric FIR Filter with $N = 6$

Compilation Report - a7

Flow Summary

<<Filter>>

Flow Status	Successful - Sat Feb 1 20:59:06 2020
Quartus Prime Version	19.1.0 Build 670 09/...2019 SJ Lite Edition
Revision Name	a7
Top-level Entity Name	topentity
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	21 / 56,480 (< 1 %)
Total registers	48
Total pins	18 / 268 (7 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	6 / 156 (4 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

Figure 13: Flow Summary of the Transformed Symmetric FIR Filter with $N = 6$

8 IIR filter

The IIR filter was synthesized in Clash and resulting in the RTL schematic shown in figure 20 in the Appendix. The picture is not shown here as it is too large to be displayed in this PDF. We refer the reader to the Clash code and compiling it in order to obtain the full image for inspection. From the Flow summary shown in figure 14 it can be confirmed that the DSP blocks used are 7, just like what can be deduced from the figure given (4). It is noticeable in both the RTL schematic and the logical utilization as seen in the flow summary that there is a lot more selection hardware needed for this implementation compared to a FIR filter. The Throughput is once again 1 out put per clock cycle. From the SFG given it can be seen more clearly (but it can also be seen in the RTL) that the critical path defining the latency is two multipliers long and 2 additions. This corresponds to the path going around clockwise on the perimeter of the SFG, passing the coefficient b_0 and z and returning to the first register on the left.

When given an impulse as an input a filter impulse response is expected. In fact this is the case, as seen in the results plotted in figure 15.

Compilation Report - a8

Flow Summary

<<Filter>>

Flow Status	Successful - Sat Feb 1 22:08:43 2020
Quartus Prime Version	19.1.0 Build 670 09/...2019 SJ Lite Edition
Revision Name	a8
Top-level Entity Name	topentity
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	132 / 56,480 (< 1 %)
Total registers	54
Total pins	38 / 268 (14 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	7 / 156 (4 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

0 / 6 (0 %)

Figure 14: Flow Summary of Synthesis of the IIR Filter

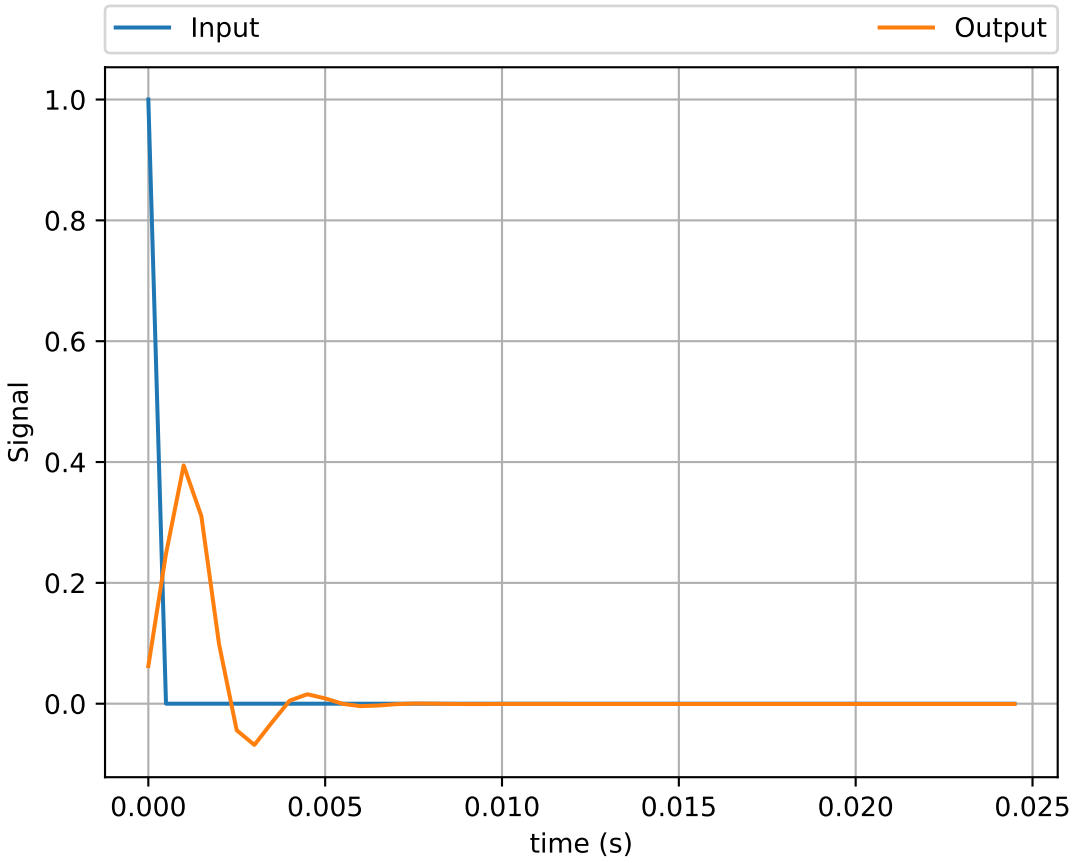

Figure 15: Plot of the Output of the IIR Filter

9 Transformed IIR filter

By applying the transformation on the IIR filter the figure 16 is obtained. Based on this SFG an RTL schematic was made as shown in the Appendix (due to size/readability issues) figure 21. For quick verification one can see the flow sumarry given in 17. In contrast to the other transformation made this one increases the critical path of the circuit, which is S0 making a clockwise round around circuit and flowing eventually to the output; this path in fact totals 2 multipliers and 6 adders. It is therefore a very big latency implementation.

It is noticeable in both the RTL schematic and the logical utilization as seen in the flow summary that there is a lot more selection hardware needed for this implementation compared to a FIR filter. The Throughput is once again 1 out put per clock cycle. From the SFG given it can be seen more clearly (but it can also be seen in the RTL) that the critical path defining the latency is two multipliers long and 2 additions. This corresponds to the path going around clockwise on the perimeter of the SFG, passing the coefficient b_0 and z and returning to the first register on the left. The resource usage of the two is quite similar, and both have the same throughput of 1 output per clock cycle.

Just like in figure 15 an impulse is given to the filter which results in the same output shown in figure18. This result is to be expected as we initially assumed that the transformation would have no effect on the behavior of the system.

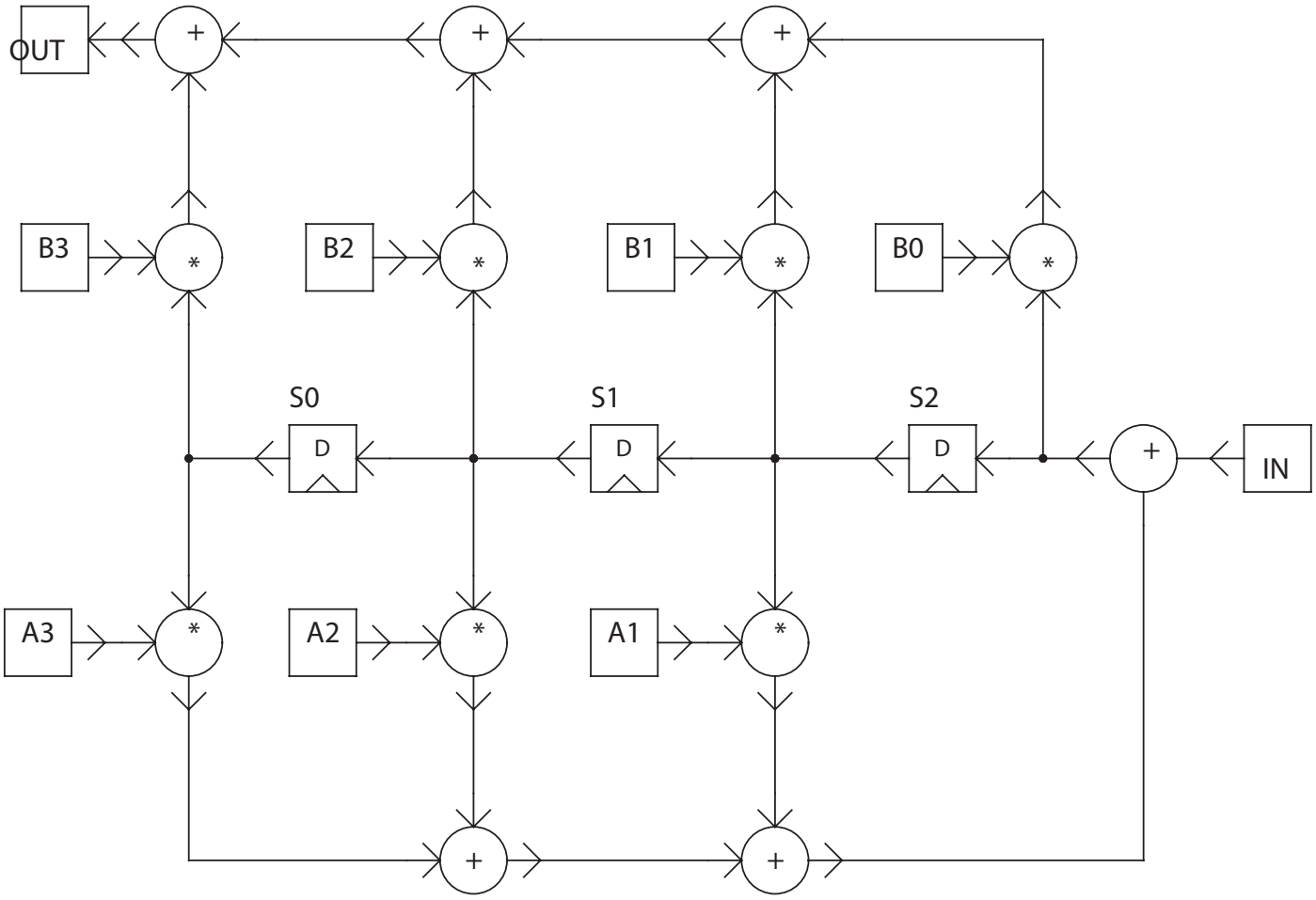

Figure 16: Schematics of the Transformed IIR Filter

Compilation Report - a9

Flow Summary

<<Filter>>

Flow Status	Successful - Sat Feb 1 22:15:10 2020
Quartus Prime Version	19.1.0 Build 670 09/...2019 SJ Lite Edition
Revision Name	a9
Top-level Entity Name	topentity
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	138 / 56,480 (< 1 %)
Total registers	37
Total pins	38 / 268 (14 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	7 / 156 (4 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

Figure 17: Flow Summary of Synthesis of the Transformed IIR Filter

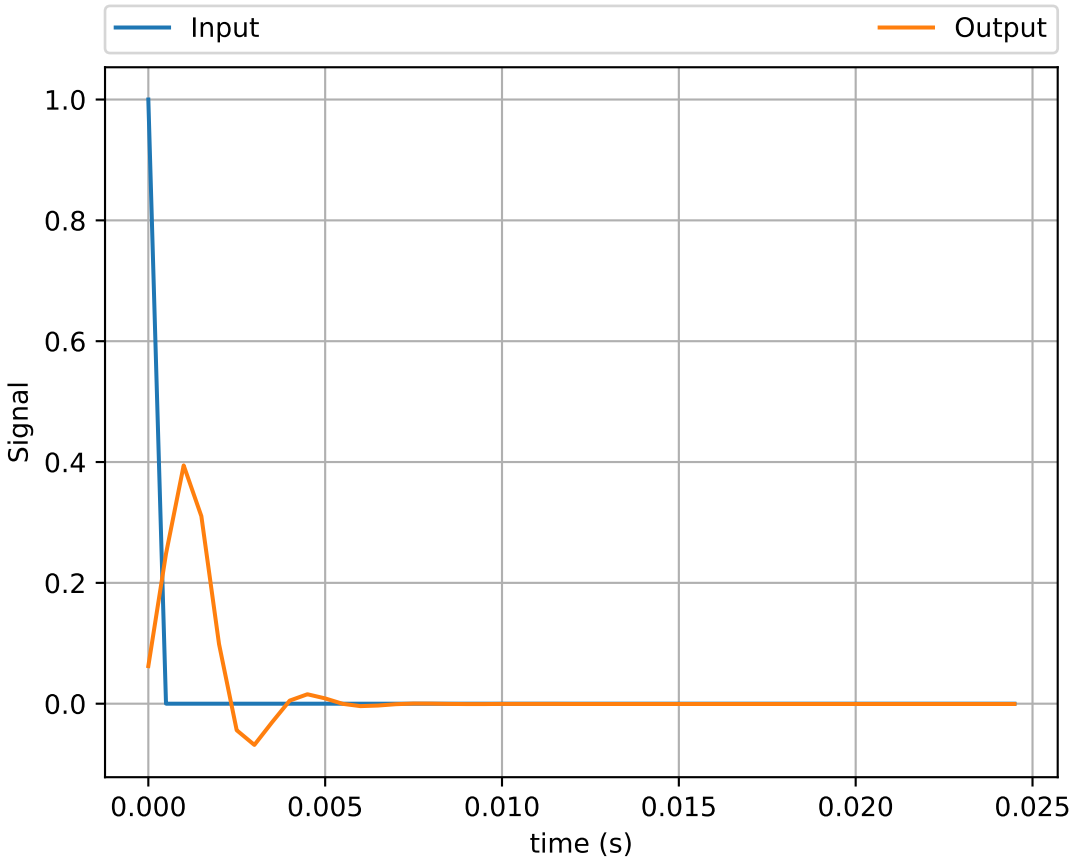

Figure 18: Plot of the Output of the Transformed IIR Filter

10 Appendix - Figures

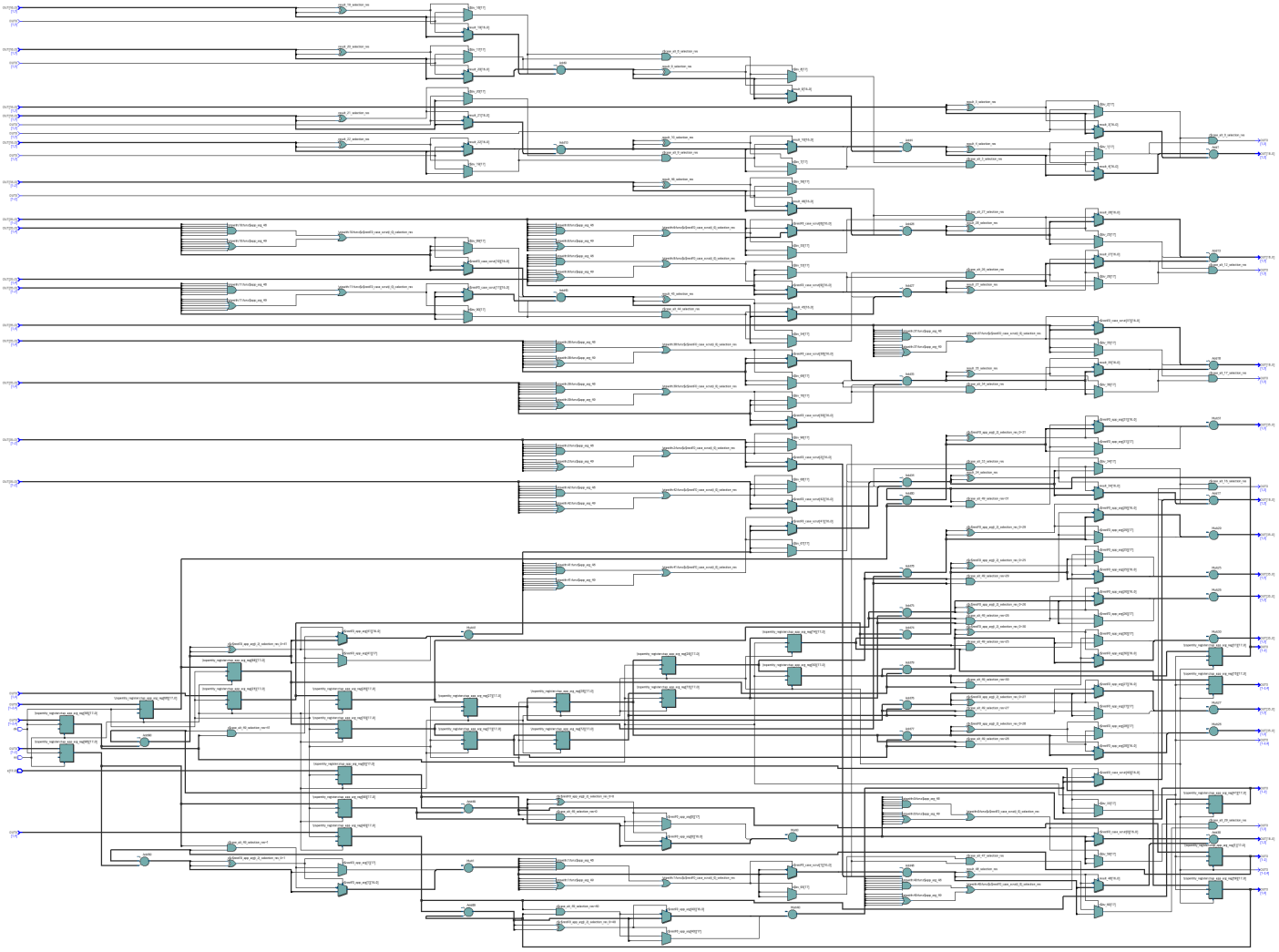

Figure 19: RTL Schedule of the Symmetric FIR Filter for a Large Input

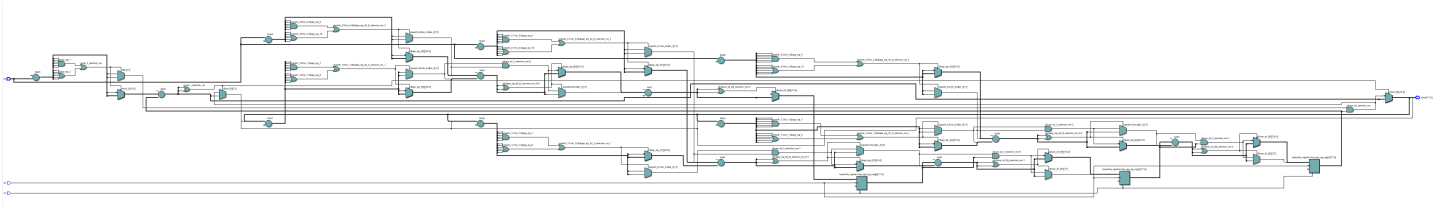

Figure 20: RTL Schematics of Synthesis of the IIR Filter

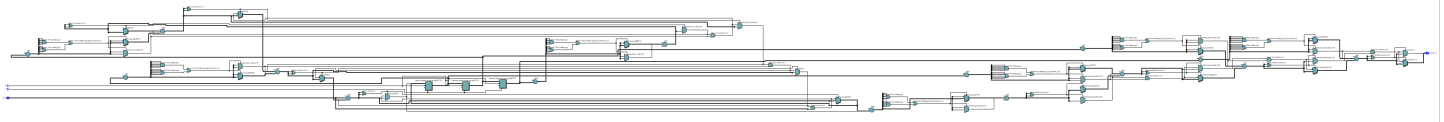

Figure 21: RTL Schematics of Synthesis of the Transformed IIR Filter