

Deliverable 1/2:

Testbench, constraint, and testvector files for all three cases can be found at
https://github.com/Spoogius/gmu-ece-545/tree/main/HW_4

Deliverable 3:

Case i - Behavioral Simulation

For the behavioral simulation a clock period of 10 ns was used.

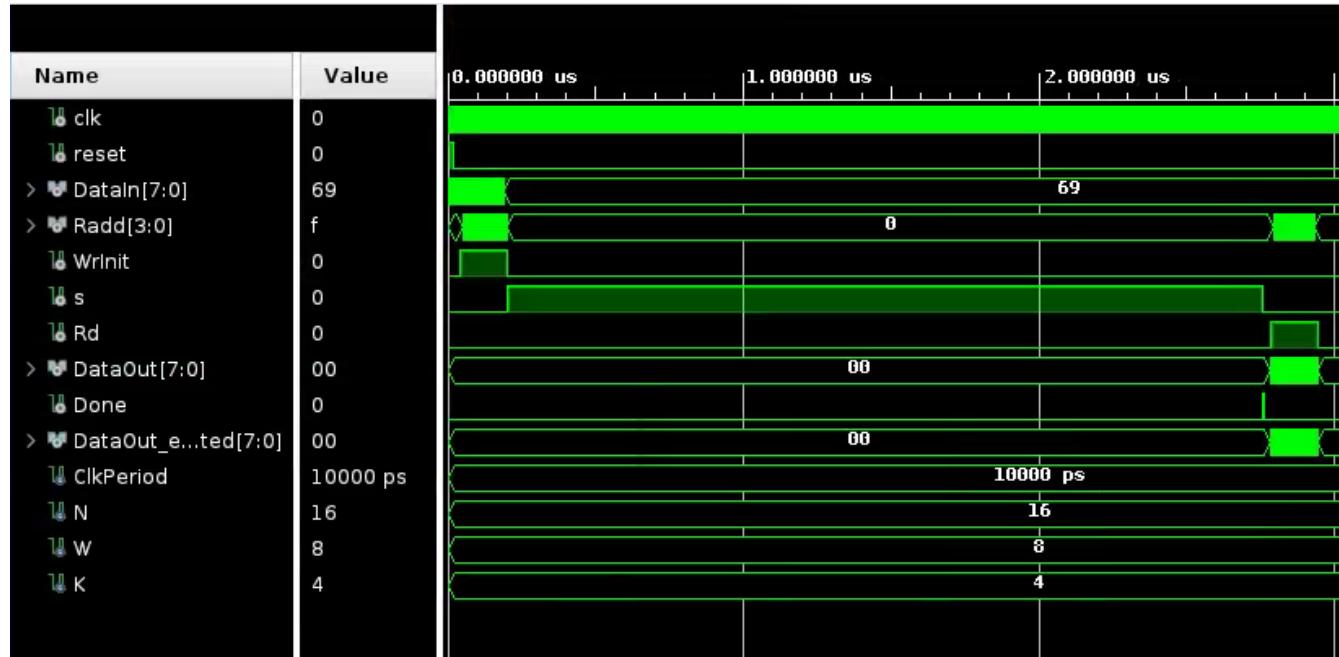


Figure 1. Case i - full behavioral simulation

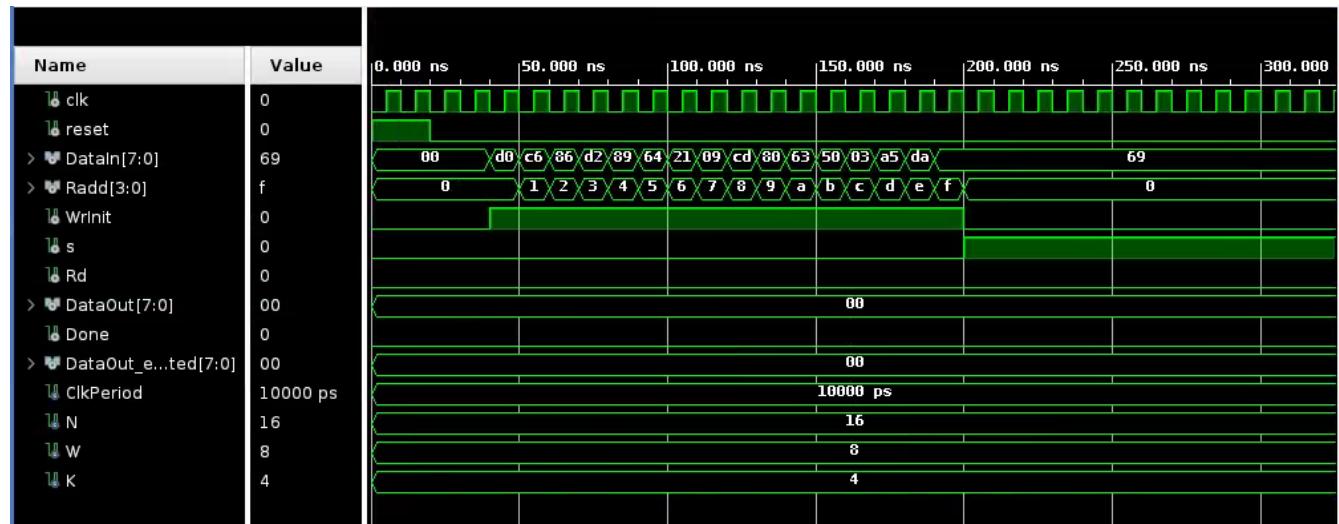


Figure 2. Case i - Loading behavioral simulation

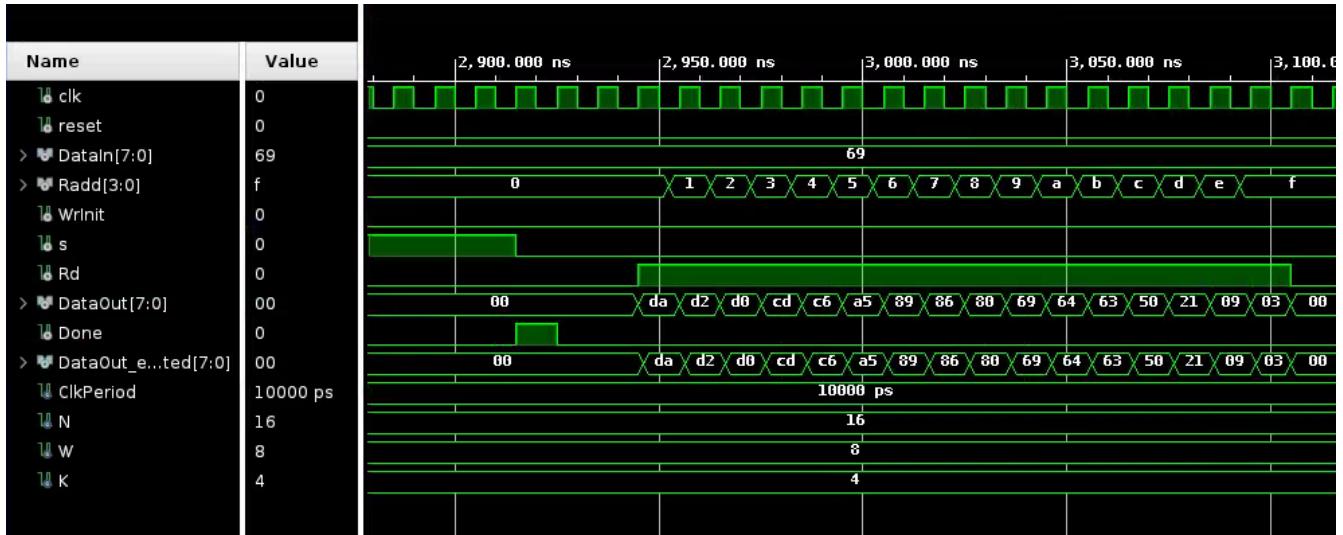


Figure 3. Case i - Reading behavioral simulation

Case i - Post-synthesis Simulation

For the post synthesis simulation, a clock period of 29ns was used. 29 ns comes from the result in *Deliverable 4.a*.

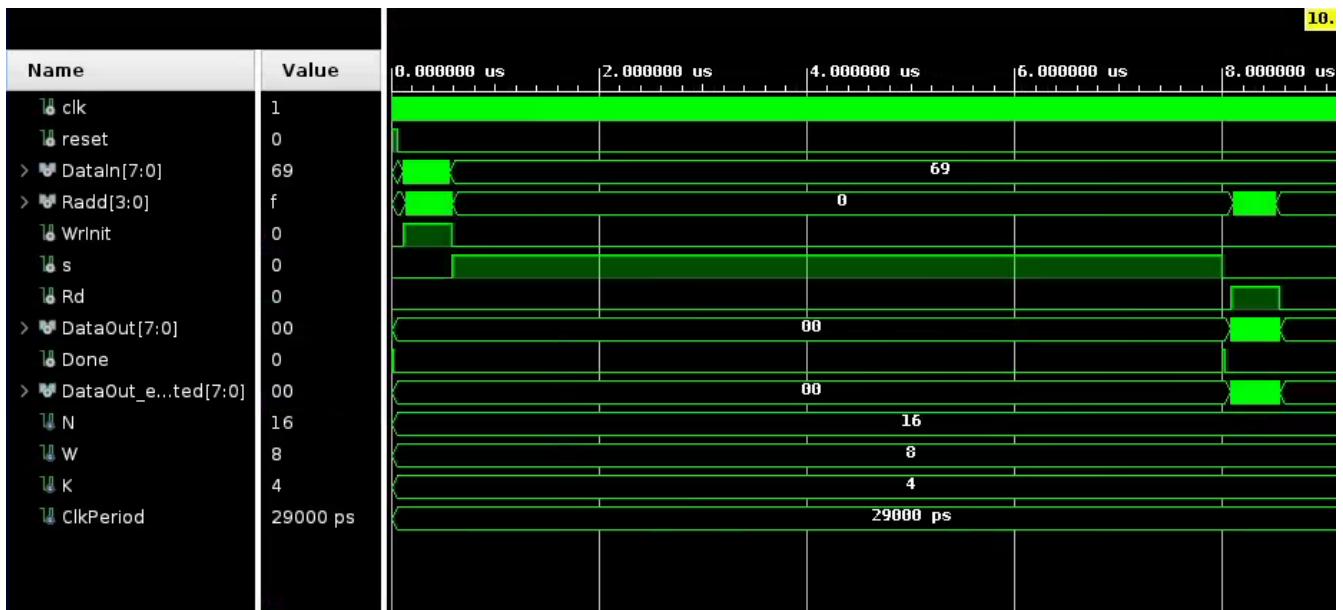


Figure 4. Case i - full synthesis timing simulation

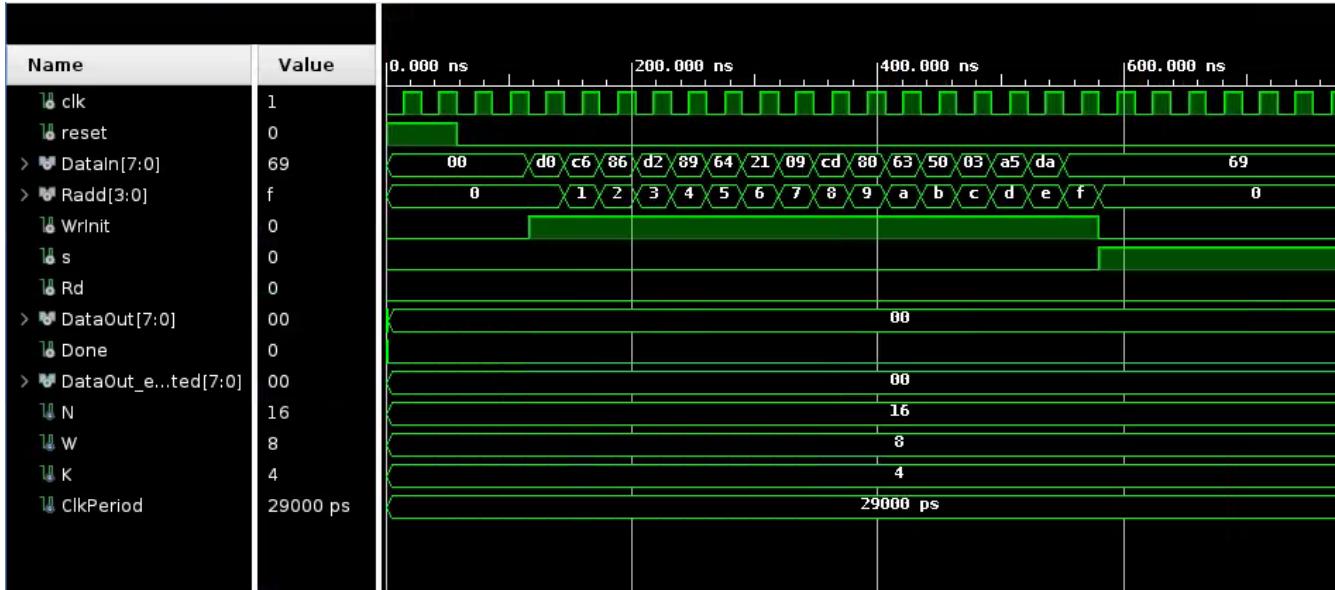


Figure 5. Case i - Loading synthesis timing simulation

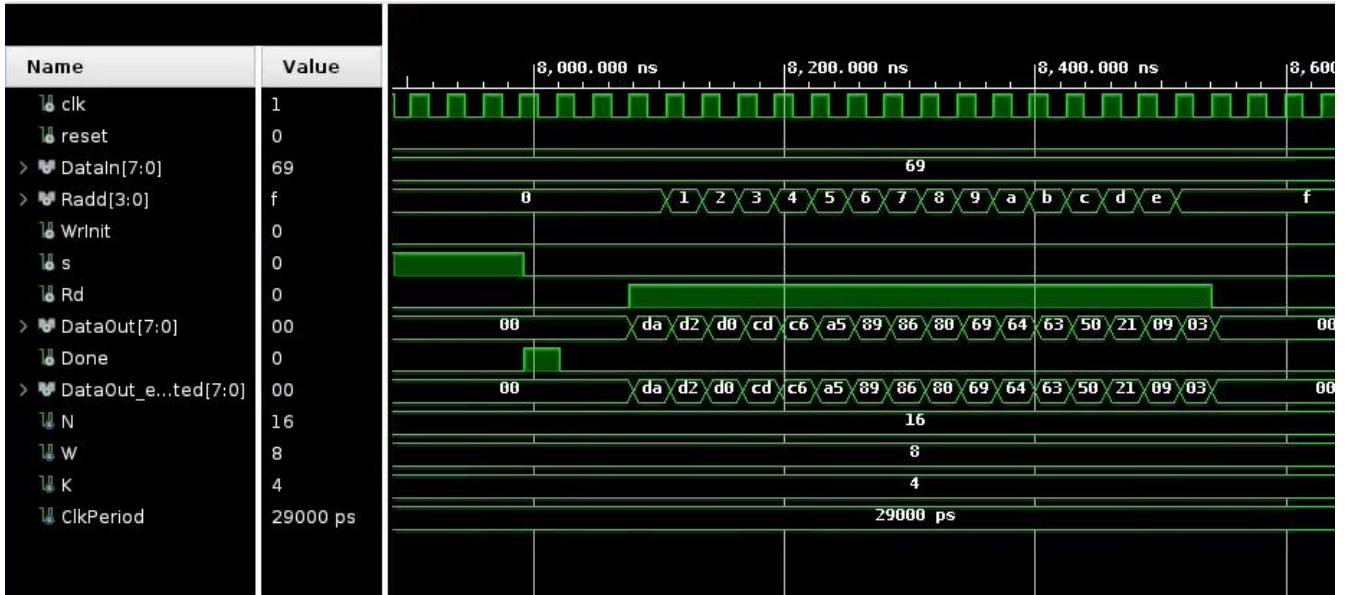


Figure 6. Case i - Reading synthesis timing simulation

Case i - Post-implementation Simulation

For the post implementation simulation, a clock period of 15 ns was used. 15 ns comes from the result in *Deliverable 4.b*.

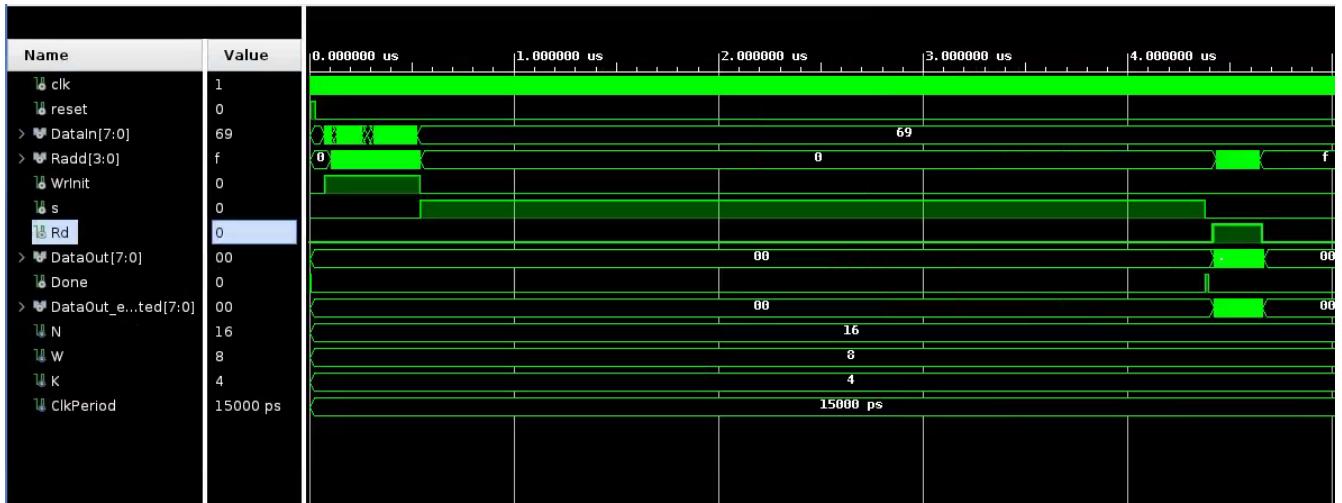


Figure 7. Case i - full implementation timing simulation

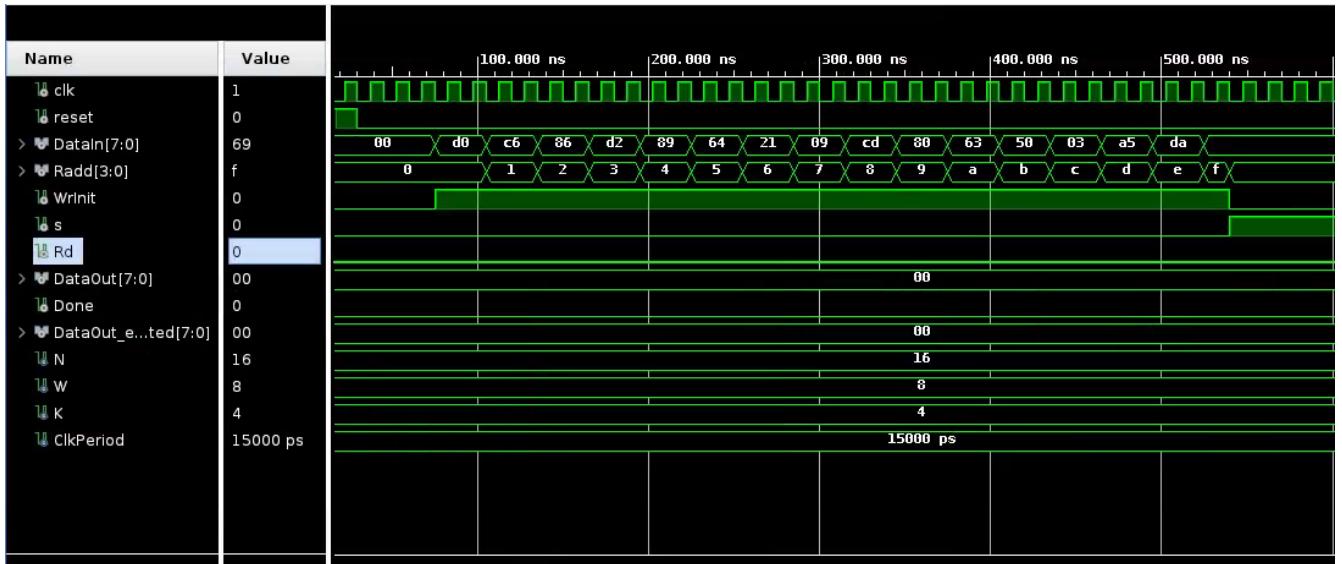


Figure 8. Case i - Loading implementation timing simulation

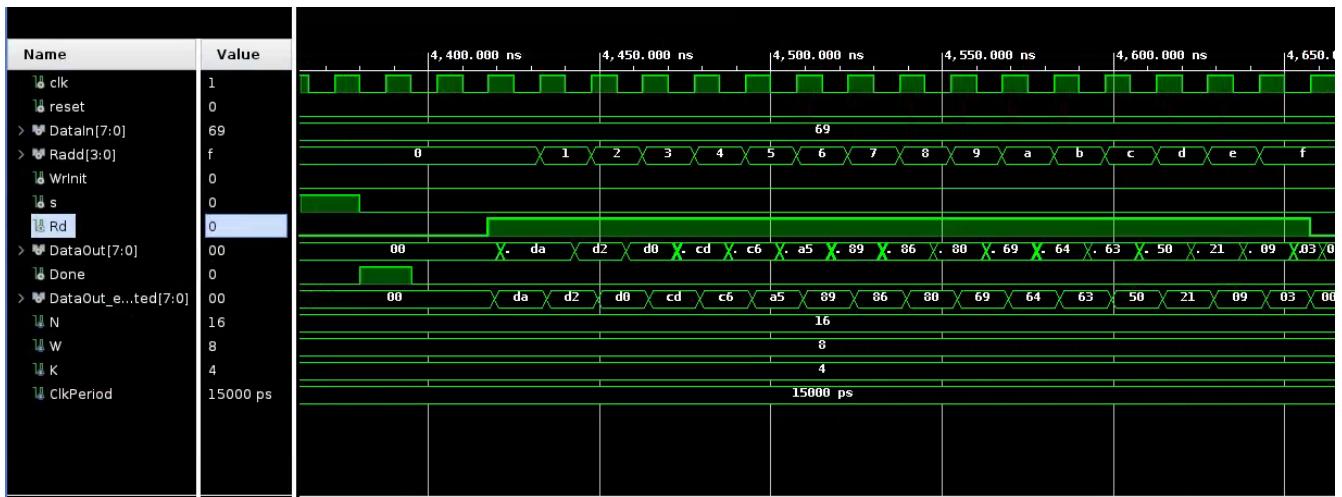


Figure 9. Case i - Reading implementation timing simulation

Case ii - Behavioral Simulation

For the behavioral simulation a clock period of 10 ns was used.

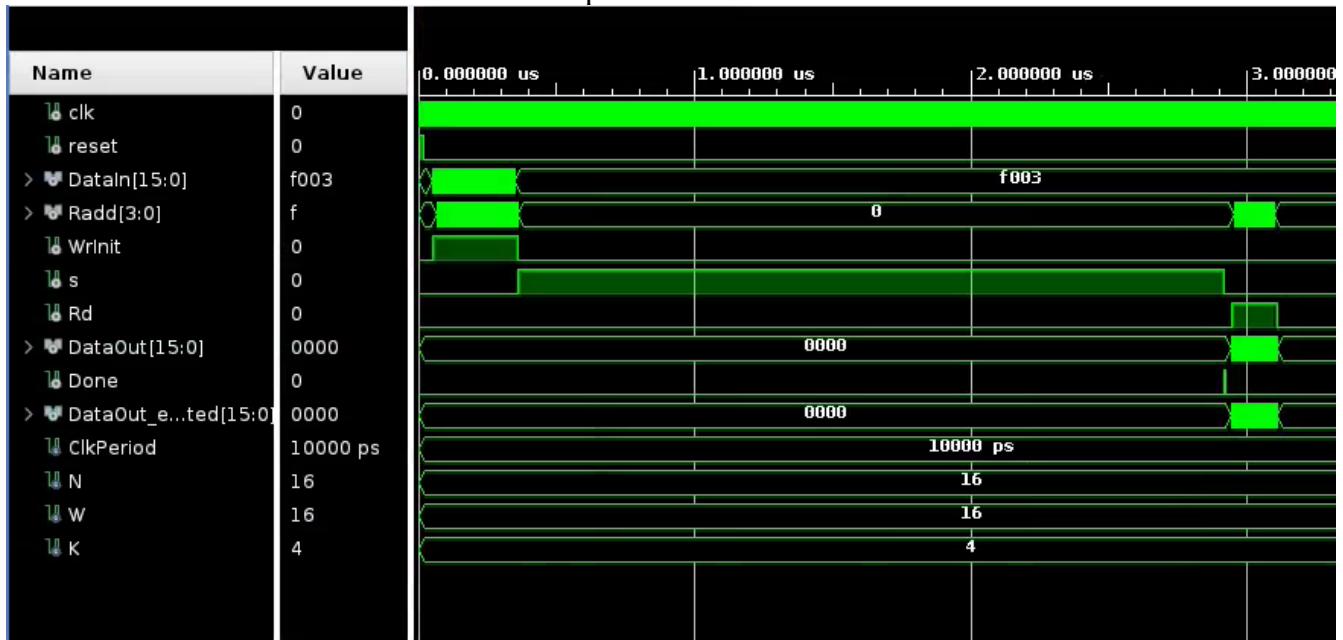


Figure 10. Case ii - full behavioral simulation

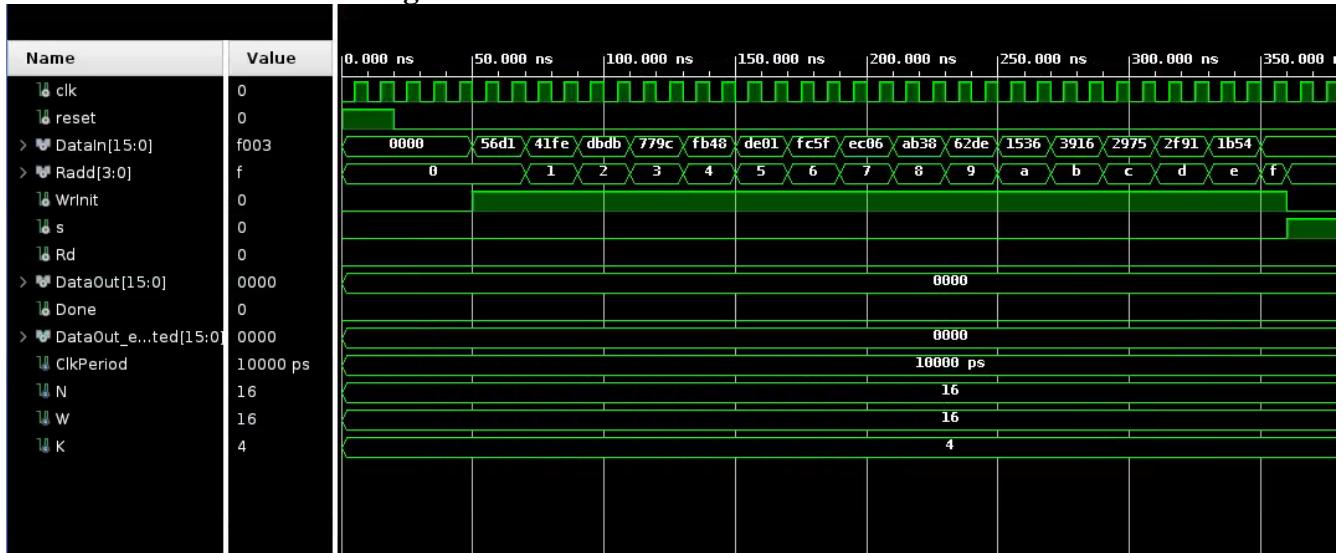


Figure 11. Case ii - Loading behavioral simulation

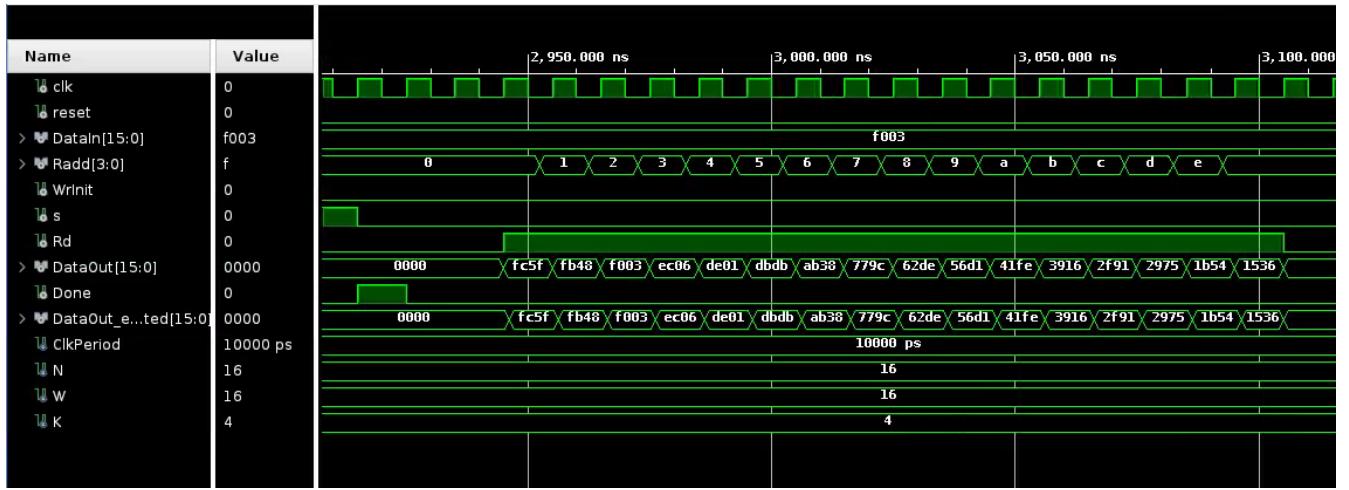


Figure 12. Case ii - Reading behavioral simulation

Case ii - Post-synthesis Simulation

For the post synthesis simulation, a clock period of 36 ns was used. 36 ns comes from the result in *Deliverable 4.a.*

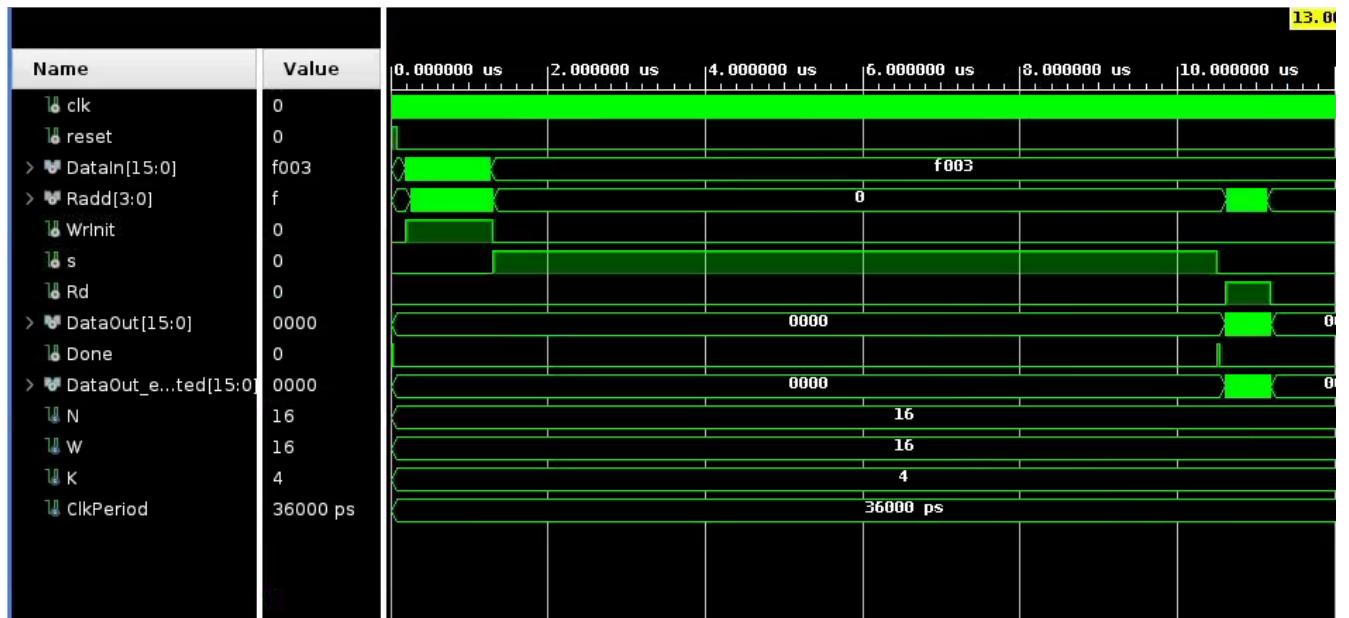


Figure 13. Case ii - full synthesis timing simulation

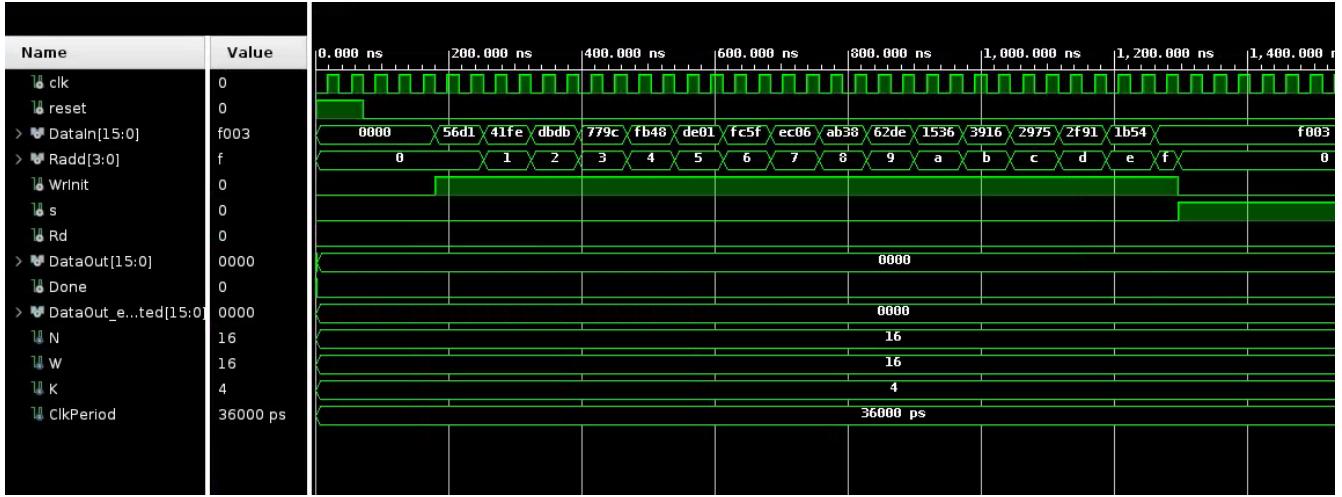


Figure 14. Case ii - Loading synthesis timing simulation

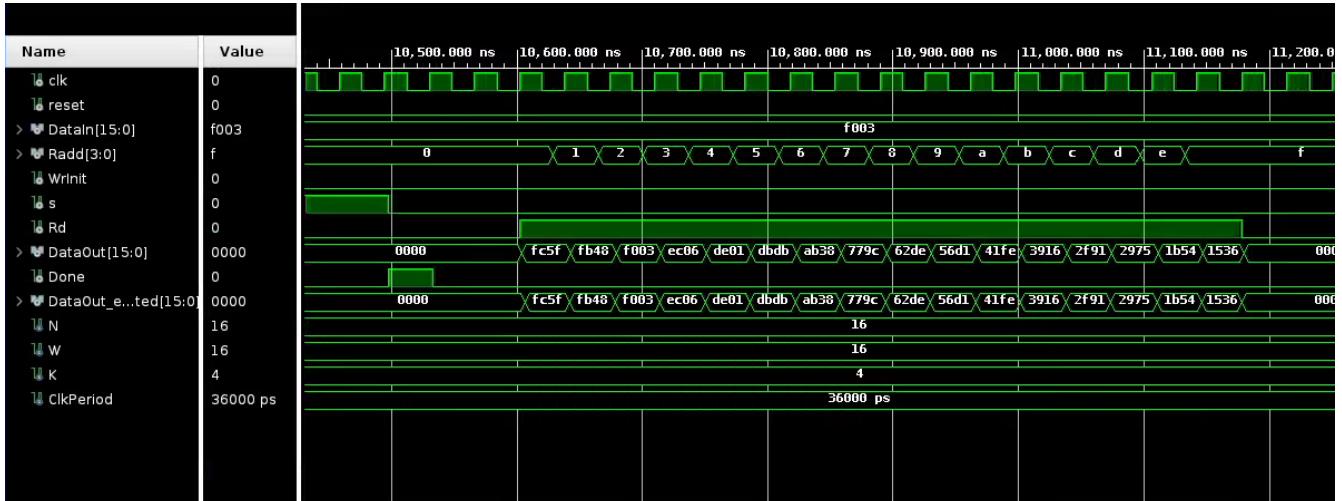


Figure 15. Case ii - Reading synthesis timing simulation

Case ii - Post-implementation Simulation

For the post implementation simulation, a clock period of 20 ns was used. 20 ns comes from the result in *Deliverable 4.b*.



Figure 16. Case ii - full implementation timing simulation

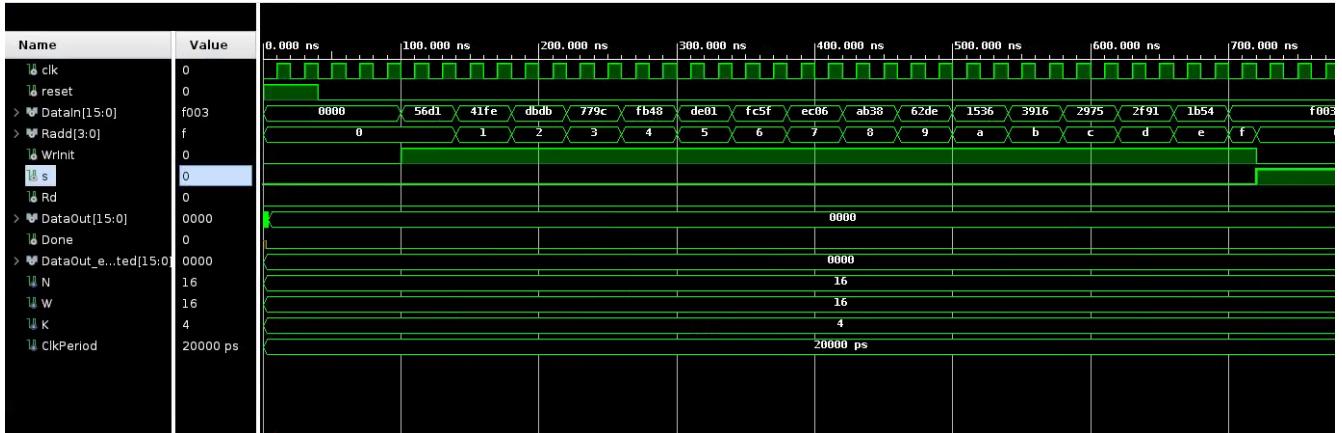


Figure 17. Case ii - Loading implementation timing simulation

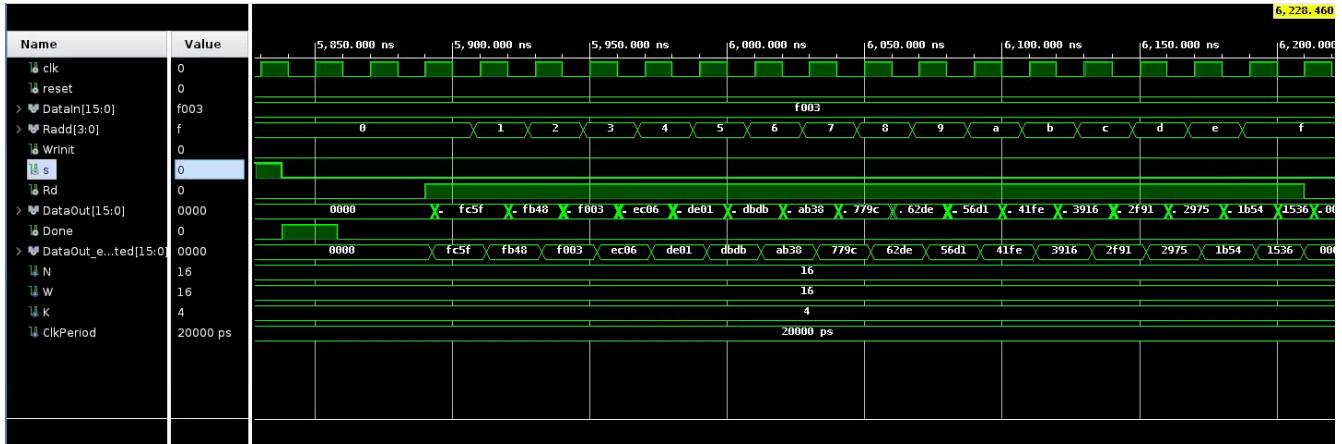


Figure 18. Case ii - Reading implementation timing simulation

Case iii - Behavioral Simulation

For the behavioral simulation a clock period of 10 ns was used.

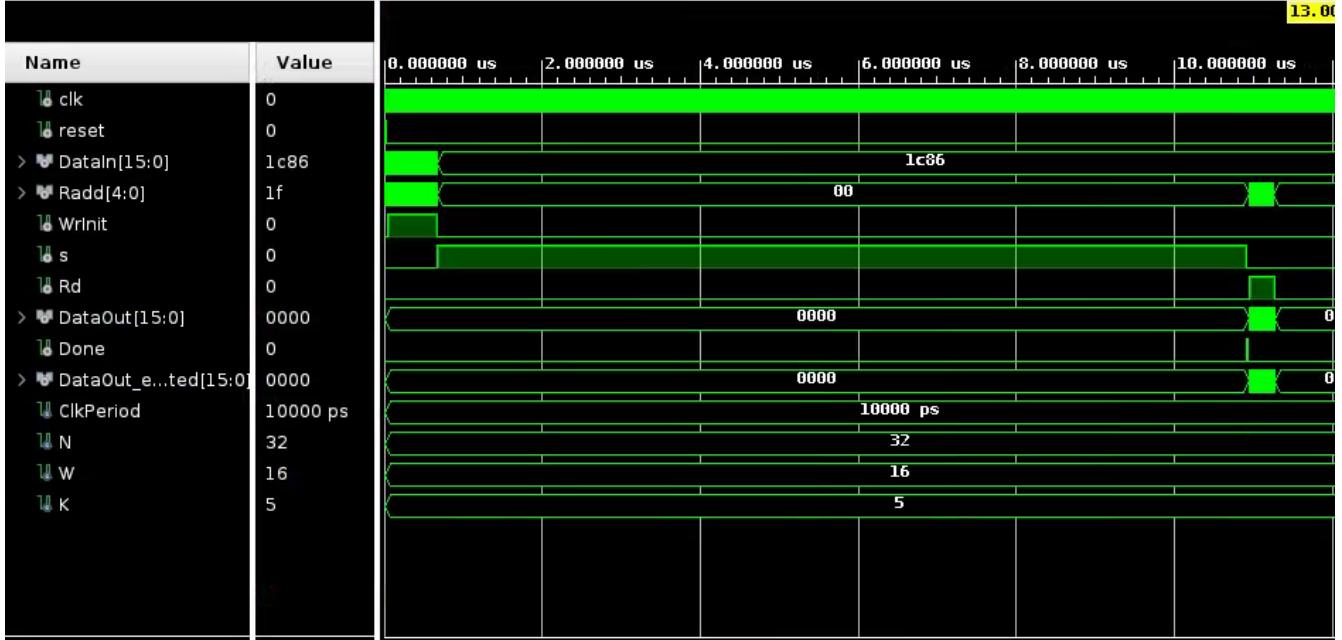


Figure 19. Case iii - full behavioral simulation

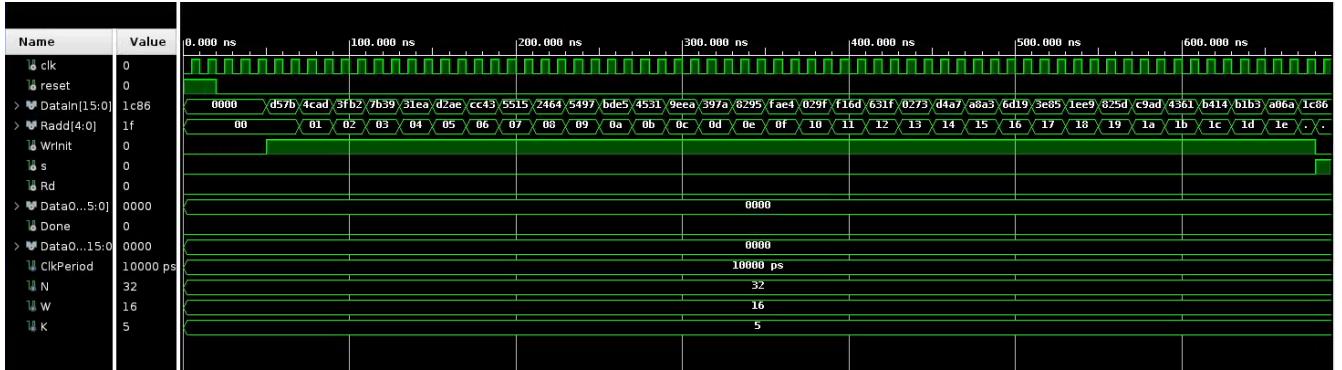


Figure 20. Case iii - Loading behavioral simulation



Figure 21. Case iii - Reading behavioral simulation

Case ii - Post-synthesis Simulation

For the post synthesis simulation, a clock period of 36 ns was used. 36 ns comes from the result in *Deliverable 4.a*.

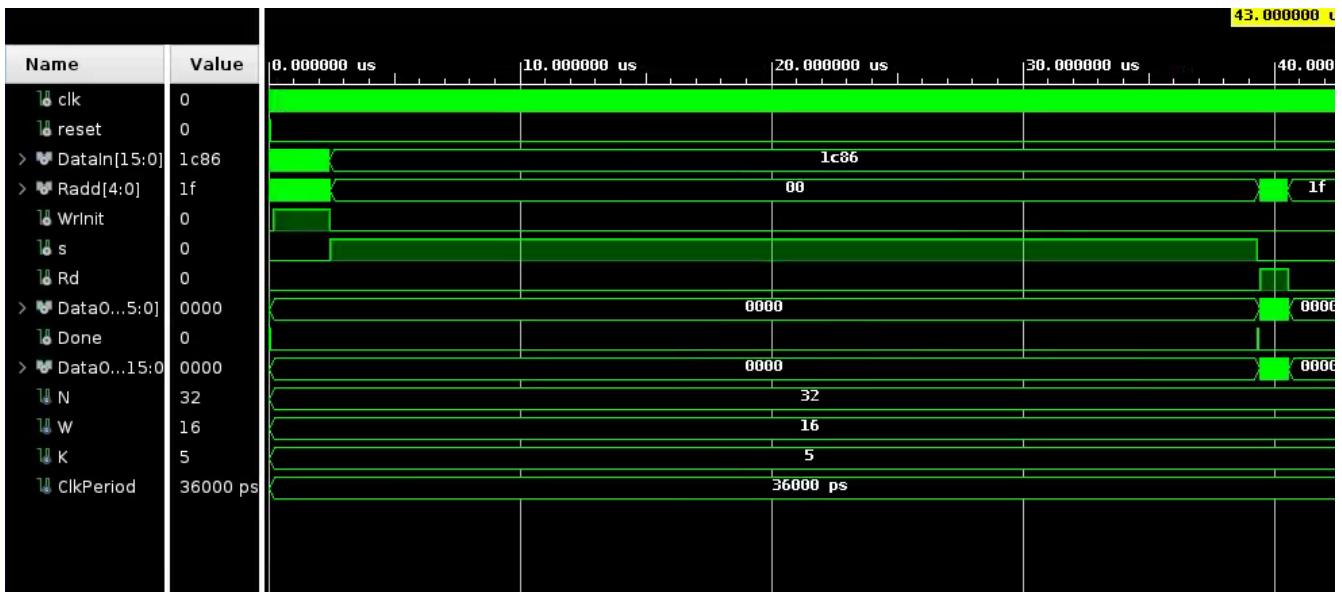


Figure 22. Case iii - full synthesis timing simulation

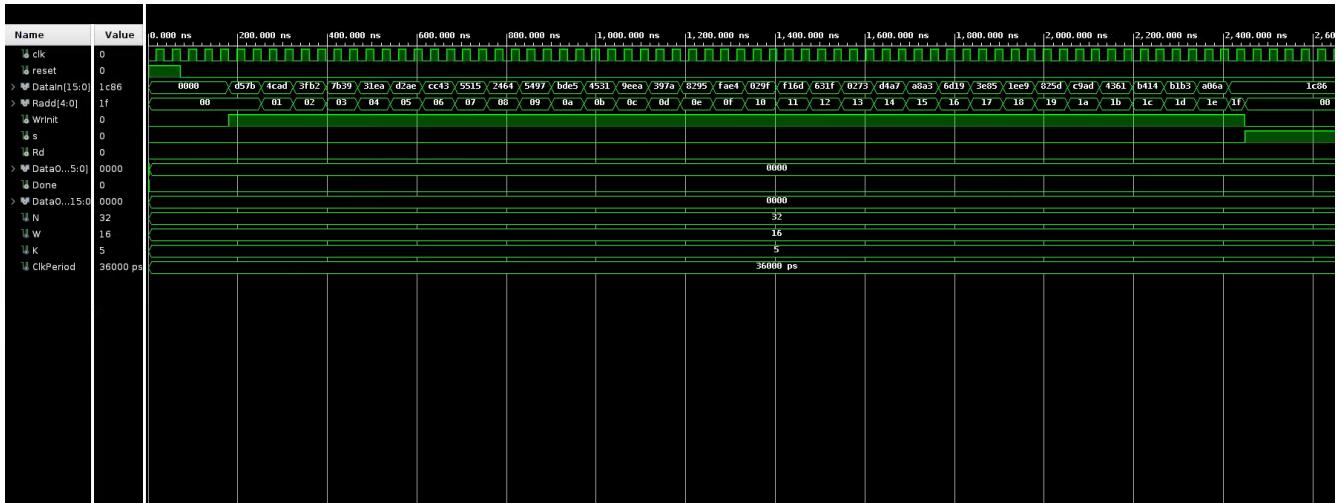


Figure 23. Case iii - Loading synthesis timing simulation



Figure 24. Case iii - Reading synthesis timing simulation

Case iii - Post-implementation Simulation

For the post implementation simulation, a clock period of 20 ns was used. 20 ns comes from the result in *Deliverable 4.b*.

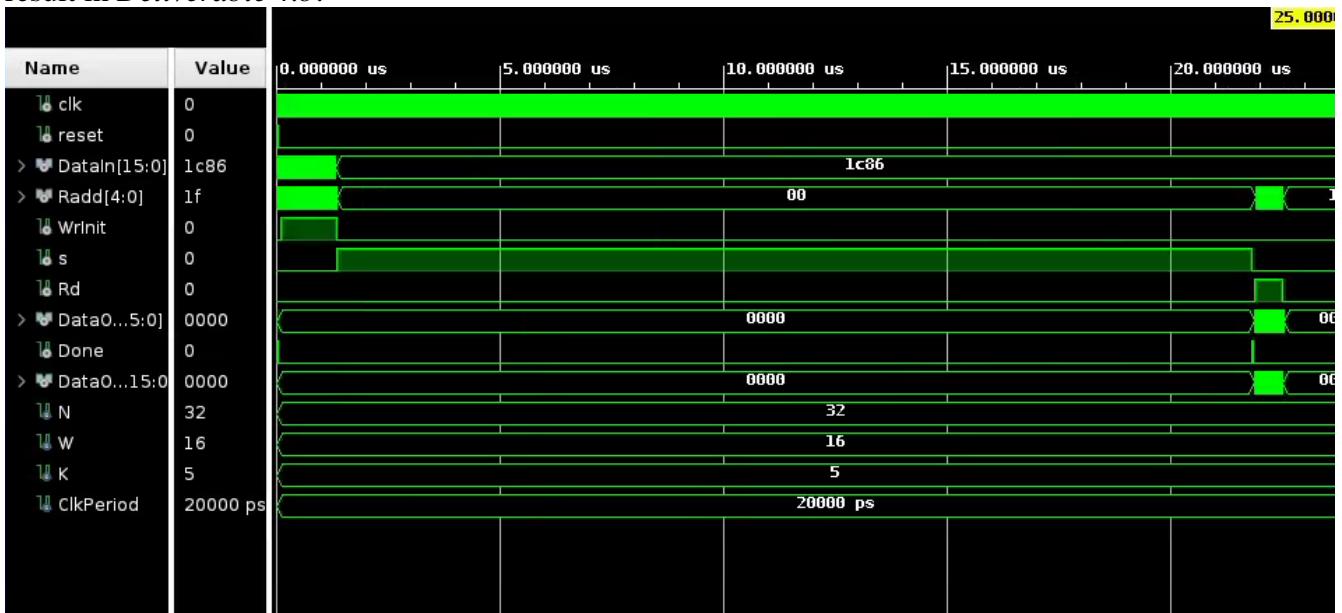


Figure 25. Case iii - full implementation timing simulation

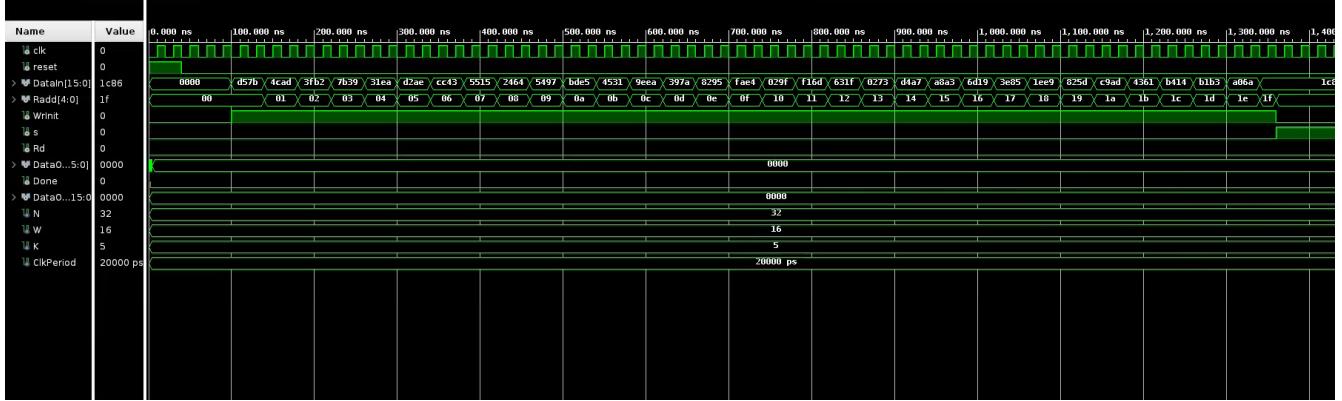


Figure 26. Case iii - Loading implementation timing simulation



Figure 27. Case iii - Reading implementation timing simulation

To demonstrate error reporting, for case i I overwrote the upper 4 bits of the expected values with 0x6. The expected result demonstrating that error reporting is working would be for all *dataout* values that don't start with a 0x6 should be printed. Figure 29 shows the simulated output, and Figure 30 shows the print statements. Clearly, the print statements demonstrate error reporting is working as intended for case i.

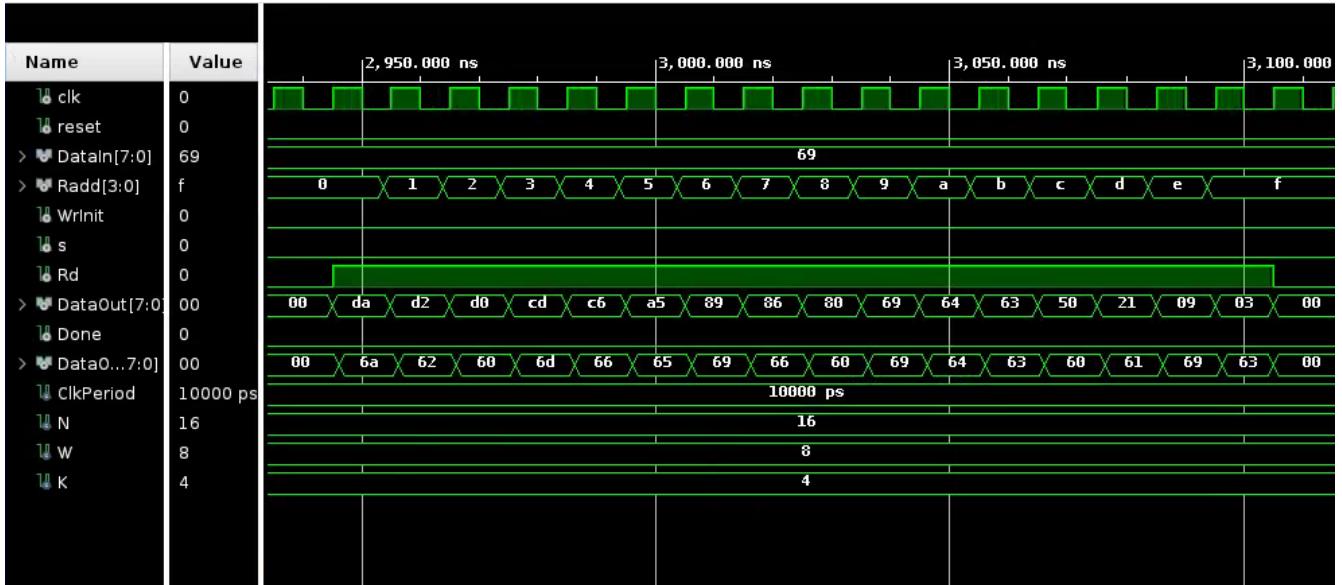


Figure 28. Error reporting test for case i

```
# run 4000ns
Time: 2952ns, Actual Output: 0xDA Expected Output: 0x6A
Time: 2962ns, Actual Output: 0xD2 Expected Output: 0x62
Time: 2972ns, Actual Output: 0xD0 Expected Output: 0x60
Time: 2982ns, Actual Output: 0xCD Expected Output: 0x6D
Time: 2992ns, Actual Output: 0xC6 Expected Output: 0x66
Time: 3002ns, Actual Output: 0xA5 Expected Output: 0x65
Time: 3012ns, Actual Output: 0x89 Expected Output: 0x69
Time: 3022ns, Actual Output: 0x86 Expected Output: 0x66
Time: 3032ns, Actual Output: 0x80 Expected Output: 0x60
Time: 3072ns, Actual Output: 0x50 Expected Output: 0x60
Time: 3082ns, Actual Output: 0x21 Expected Output: 0x61
Time: 3092ns, Actual Output: 0x09 Expected Output: 0x69
Time: 3102ns, Actual Output: 0x03 Expected Output: 0x63
```

Figure 29. Error reporting test print statements for case i

The same error checking test is then ran again for case iii, this time the upper 4 bits are overwritten with 0x8. Again, as can be seen in Figure 30 and 31, error reporting is working as intended.



Figure 30. Error reporting test for case iii

```

Time: 10952ns, Actual Output: 0xFAE4 Expected Output: 0x8AE4
Time: 10962ns, Actual Output: 0xF16D Expected Output: 0x816D
Time: 10972ns, Actual Output: 0xD57B Expected Output: 0x857B
Time: 10982ns, Actual Output: 0xD4A7 Expected Output: 0x84A7
Time: 10992ns, Actual Output: 0xD2AE Expected Output: 0x82AE
Time: 11002ns, Actual Output: 0xCC43 Expected Output: 0x8C43
Time: 11012ns, Actual Output: 0xC9AD Expected Output: 0x89AD
Time: 11022ns, Actual Output: 0xBDE5 Expected Output: 0x8DE5
Time: 11032ns, Actual Output: 0xB414 Expected Output: 0x8414
Time: 11042ns, Actual Output: 0xB1B3 Expected Output: 0x81B3
Time: 11052ns, Actual Output: 0xA8A3 Expected Output: 0x88A3
Time: 11062ns, Actual Output: 0xA06A Expected Output: 0x806A
Time: 11072ns, Actual Output: 0x9EEA Expected Output: 0x8EEA
Time: 11102ns, Actual Output: 0x7B39 Expected Output: 0x8B39
Time: 11112ns, Actual Output: 0x6D19 Expected Output: 0x8D19
Time: 11122ns, Actual Output: 0x631F Expected Output: 0x831F
Time: 11132ns, Actual Output: 0x5515 Expected Output: 0x8515
Time: 11142ns, Actual Output: 0x5497 Expected Output: 0x8497
Time: 11152ns, Actual Output: 0x4CAD Expected Output: 0x8CAD
Time: 11162ns, Actual Output: 0x4531 Expected Output: 0x8531
Time: 11172ns, Actual Output: 0x4361 Expected Output: 0x8361
Time: 11182ns, Actual Output: 0x3FB2 Expected Output: 0x8FB2
Time: 11192ns, Actual Output: 0x3E85 Expected Output: 0x8E85
Time: 11202ns, Actual Output: 0x397A Expected Output: 0x897A
Time: 11212ns, Actual Output: 0x31EA Expected Output: 0x81EA
Time: 11222ns, Actual Output: 0x2464 Expected Output: 0x8464
Time: 11232ns, Actual Output: 0x1EE9 Expected Output: 0x8EE9
Time: 11242ns, Actual Output: 0x1C86 Expected Output: 0x8C86
Time: 11252ns, Actual Output: 0x029F Expected Output: 0x829F
Time: 11262ns, Actual Output: 0x0273 Expected Output: 0x8273

```

Figure 31. Error reporting test print statements for case iii

Deliverable 4:

Case i:

The post synthesis WNS from the timing report is 4.247 ns for a clock period of 10 ns. The minimum clock period to pass timing would be $10\text{ns} - (-17.399) = 27.4\text{ns}$. Where 17.399 is the most negative timing violation in the timing report.

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 4.247 ns	Worst Hold Slack (WHS): -1.070 ns	Worst Pulse Width Slack (WPWS): 4.600 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): -17.399 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 26	Number of Failing Endpoints: 0
Total Number of Endpoints: 81	Total Number of Endpoints: 81	Total Number of Endpoints: 23

Timing constraints are not met.

Figure 32. Case i – Post Synthesis timing report

The post implementation WNS from the timing report is -0.241 ns for a clock period of 10 ns. The minimum clock period to pass timing would be $10\text{ns} - (-0.464) = 10.5\text{ ns}$. Where 0.464 is the most negative timing violation in the timing report.

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): -0.241 ns	Worst Hold Slack (WHS): 0.084 ns	Worst Pulse Width Slack (WPWS): 4.600 ns
Total Negative Slack (TNS): -0.464 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 2	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 81	Total Number of Endpoints: 81	Total Number of Endpoints: 23

Timing constraints are not met.

Figure 33. Case i – Post Implementation timing report

The time required for sorting can be measured as the number of clock cycles between the rising edges of the two signals *s* and *done*. As shown in Figure 34, simulated with a 10 ns clock it takes 2550 clock cycles, or 255 ns.

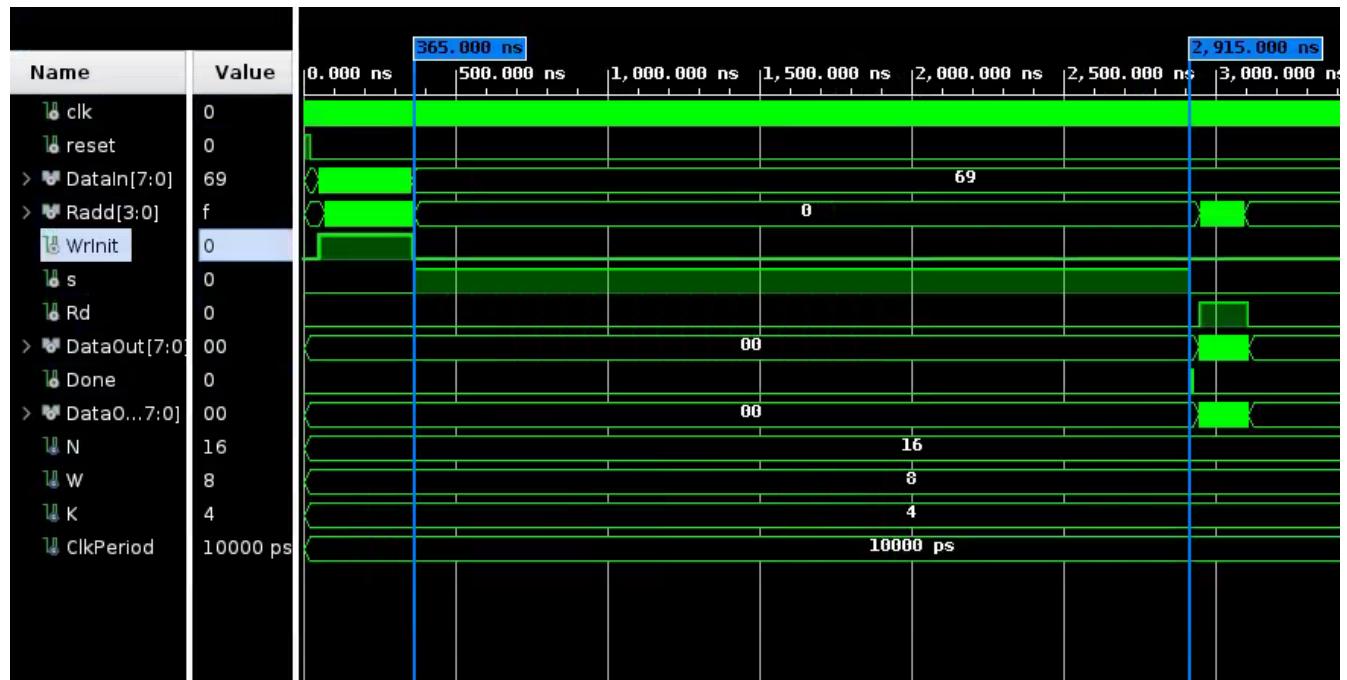


Figure 34. Case i – Markers showing the simulated time required for sorting

The delay between data out and the rising clock edge can be measured by running a post synthesis timing simulation. It can be seen in Figure 35 a delay of 10.034 ns, this is roughly consistent with the expected delay reported by the post implementation timing simulation which estimated that the circuit would slightly fail timing when using a 10 ns clock.

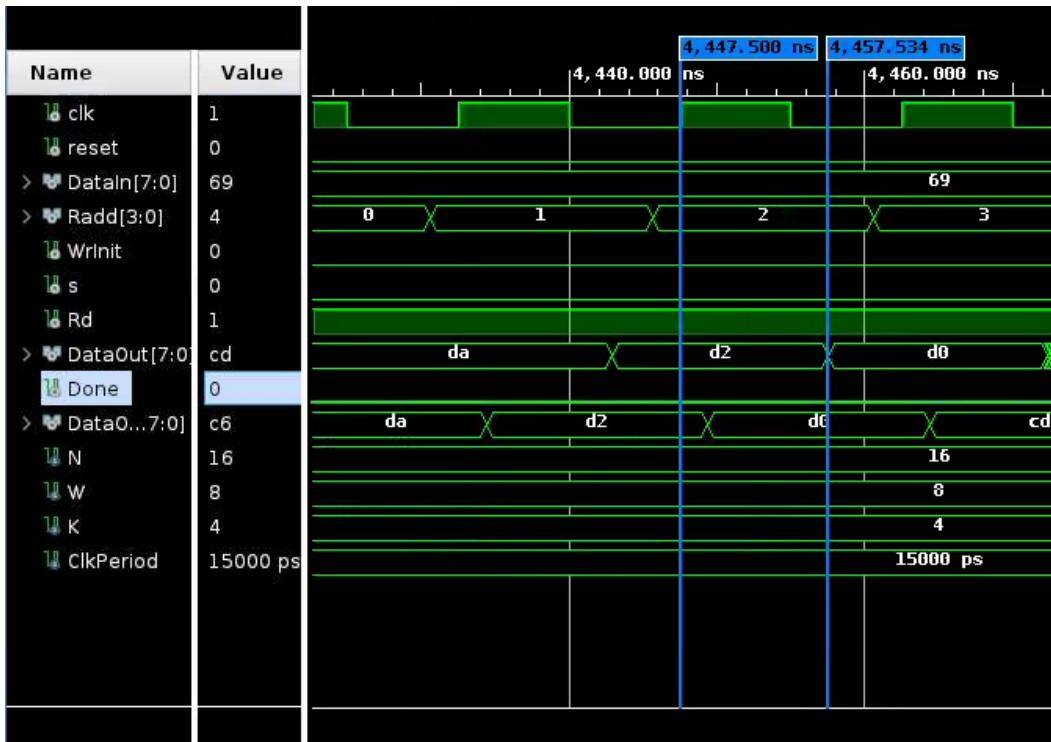


Figure 35. Case i – Markers showing data out delay

Figure 35 provides a breakdown of post implementation resource utilization for the circuit.

Name	1	Slice LUTs (303600)	Slice Registers (607200)	Slice (75900)	LUT as Logic (303600)	Block RAM Tile (1030)	Bonded IOB (600)	BUFGCTRL (32)
Sorting		33	13	11	33	0.5	26	1
Sorting_controller (Controller)		3	5	3	3	0	0	0
Sorting_datapath (Datapath)		30	8	10	30	0.5	0	0

Figure 36. Case i – Post Implementation Resource Utilization

Case ii:

The post synthesis WNS from the timing report is 4.247 ns for a clock period of 10 ns. The minimum clock period to pass timing would be $10\text{ns} - (-16.396) = 16.8\text{ns}$. Where 16.396 is the most negative timing violation in the timing report.

Design Timing Summary								
Setup		Hold		Pulse Width				
Worst Negative Slack (WNS):	4.247 ns	Worst Hold Slack (WHS):	-1.070 ns	Worst Pulse Width Slack (WPWS):	4.600 ns			
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	-16.796 ns	Total Pulse Width Negative Slack (TPWNS):	0.000 ns			
Number of Failing Endpoints:	0	Number of Failing Endpoints:	25	Number of Failing Endpoints:	0			
Total Number of Endpoints:	74	Total Number of Endpoints:	74	Total Number of Endpoints:	21			

Timing constraints are not met.

Figure 37. Case ii – Post Synthesis timing report

The post implementation WNS from the timing report is -0.241 ns for a clock period of 10 ns. The minimum clock period to pass timing would be $10\text{ns} - (-0.516) = 10.5\text{ ns}$. Where 0.516 is the most negative timing violation in the timing report.

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): -0.251 ns	Worst Hold Slack (WHS): 0.061 ns	Worst Pulse Width Slack (WPWS): 4.600 ns
Total Negative Slack (TNS): -0.516 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 4	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 74	Total Number of Endpoints: 74	Total Number of Endpoints: 21

Timing constraints are not met.

Figure 38. Case ii – Post Implementation timing report

The time required for sorting can be measured as the number of clock cycles between the rising edges of the two signals s and $done$. As shown in Figure 39, simulated with a 10 ns clock it takes 2550 clock cycles, or 255 ns. Since N is the same as in case i the runtime as a function of clock cycles doesn't change.

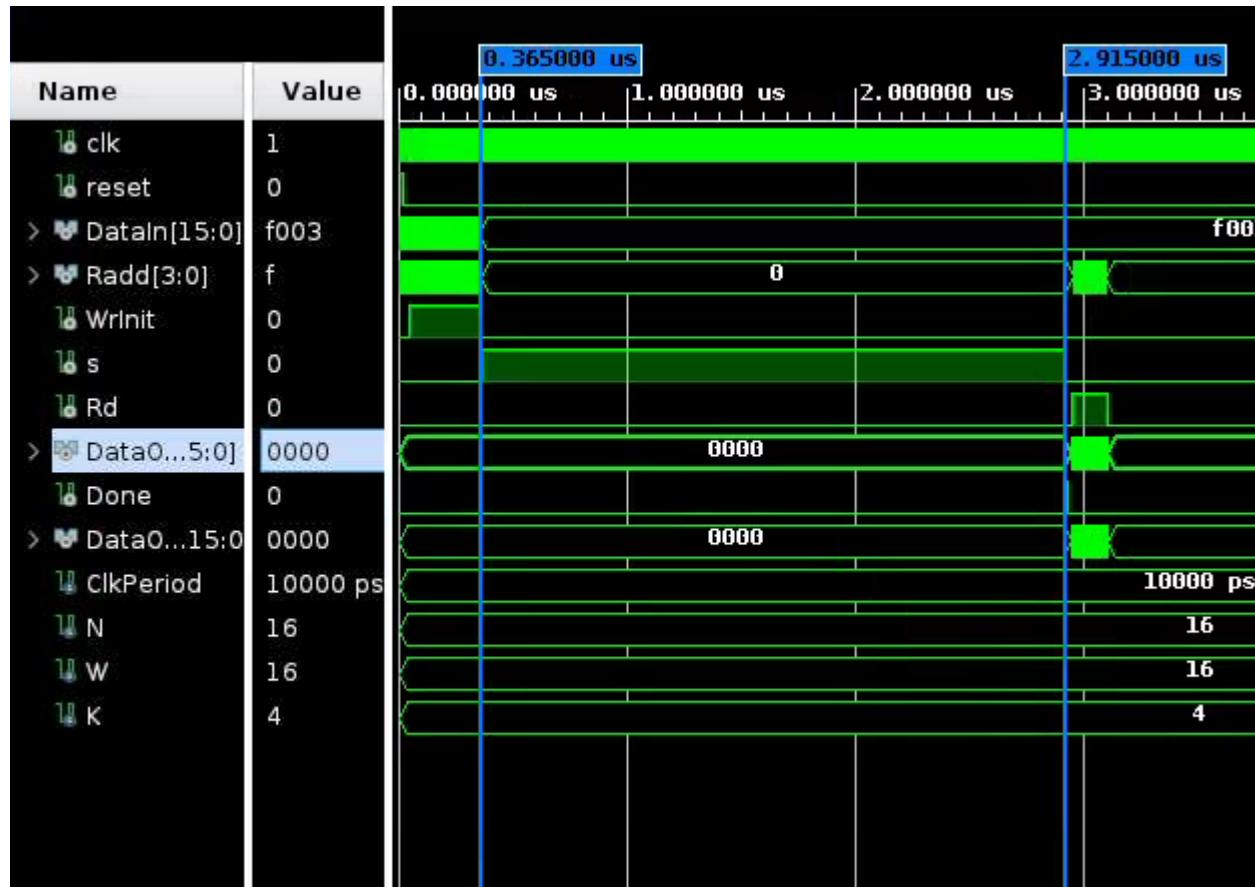


Figure 39. Case ii – Markers showing the simulated time required for sorting

The delay between data out and the rising clock edge can be measured by running a post synthesis timing simulation. It can be seen in Figure 40 a delay of 10.901 ns, this is roughly consistent with the expected delay reported by the post implementation timing simulation which estimated that the circuit would slightly fail timing when using a 10 ns clock.

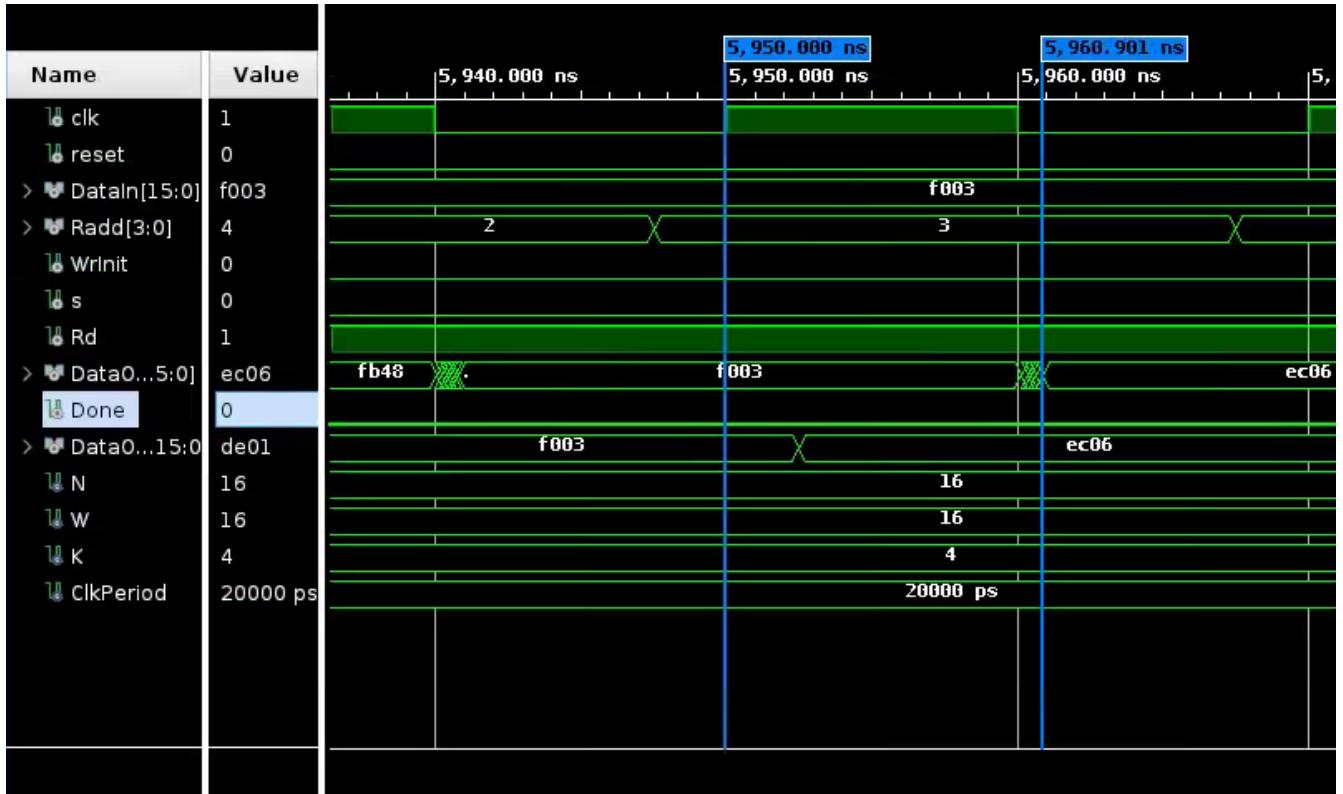


Figure 40. Case ii – Markers showing data out delay

Figure 41 provides a breakdown of post implementation resource utilization for the circuit.

Name	1	Slice LUTs (303600)	Slice Registers (607200)	Slice (75900)	LUT as Logic (303600)	Block RAM Tile (1030)	Bonded IOB (600)	BUFGCTRL (32)
Sorting		49		13	15		49	0.5
Sorting_controller (Controller)		3		5	4		3	0
Sorting_datapath (Datapath)		46		8	15		46	0

Figure 41. Case ii – Post Implementation Resource Utilization

Case iii:

The post synthesis WNS from the timing report is 4.247 ns for a clock period of 10 ns. The minimum clock period to pass timing would be 10ns – (-17.399) = 17.4ns. Where 17.399 is the most negative timing violation in the timing report.

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 4.247 ns	Worst Hold Slack (WHS): -1.070 ns	Worst Pulse Width Slack (WPWS): 4.600 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): -17.399 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 26	Number of Failing Endpoints: 0
Total Number of Endpoints: 81	Total Number of Endpoints: 81	Total Number of Endpoints: 23

Timing constraints are not met.

Figure 42. Case iii – Post Synthesis timing report

The post implementation WNS from the timing report is -0.241 ns for a clock period of 10 ns. The minimum clock period to pass timing would be $10\text{ns} - (-0.464) = 10.5$ ns. Where 0.464 is the most negative timing violation in the timing report.

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): -0.241 ns	Worst Hold Slack (WHS): 0.084 ns	Worst Pulse Width Slack (WPWS): 4.600 ns
Total Negative Slack (TNS): -0.464 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 2	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 81	Total Number of Endpoints: 81	Total Number of Endpoints: 23

Timing constraints are not met.

Figure 43. Case iii – Post Implementation timing report

The time required for sorting can be measured as the number of clock cycles between the rising edges of the two signals *s* and *done*. As shown in Figure 44, simulated with a 10 ns clock it takes 1023 clock cycles, or 10.23 us. Since *N* is now bigger than in cases i and ii the runtime as a function of clock cycles increases as expected.

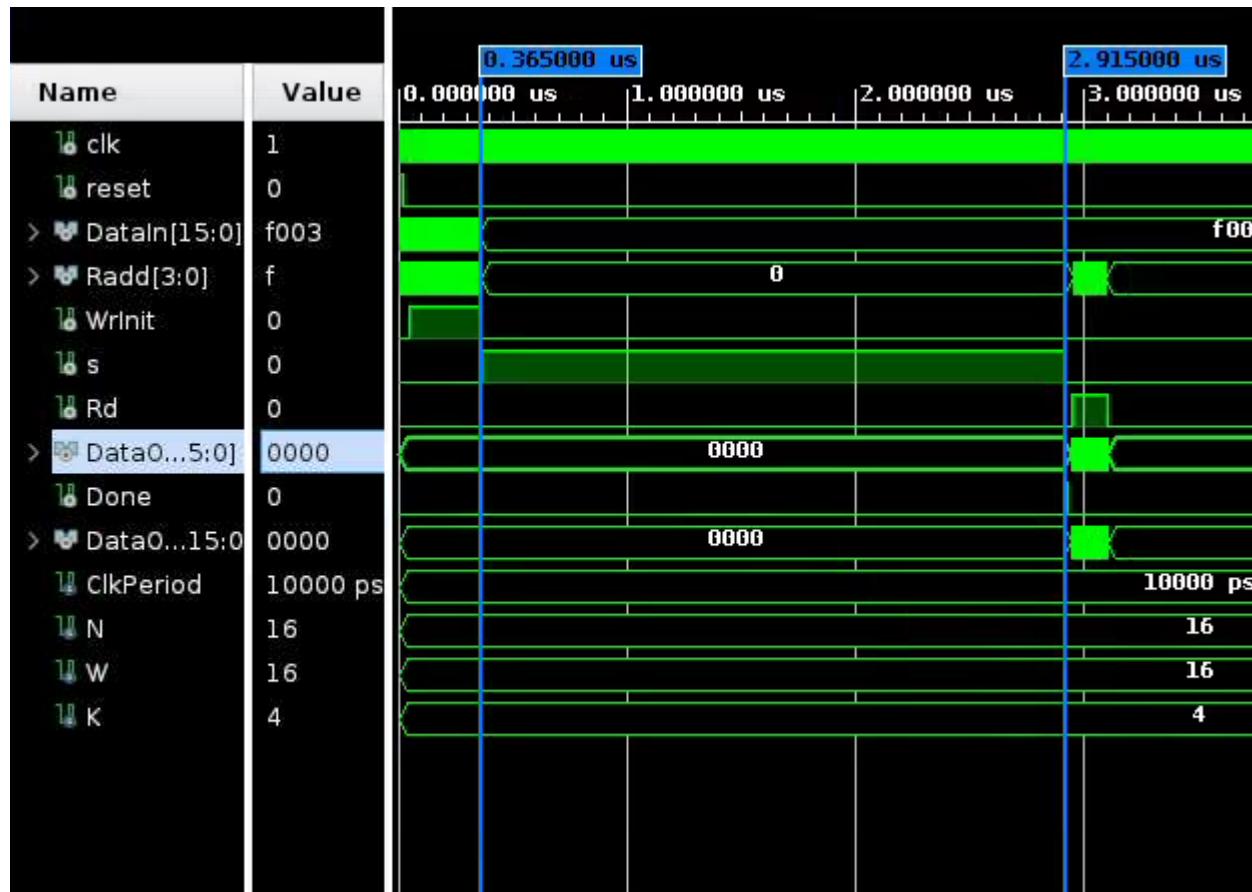


Figure 44. Case iii – Markers showing the simulated time required for sorting

The delay between data out and the rising clock edge can be measured by running a post synthesis timing simulation. It can be seen in Figure 45 a delay of 10.584 ns, this is roughly consistent with the expected delay reported by the post implementation timing simulation which estimated that the circuit would slightly fail timing when using a 10 ns clock.

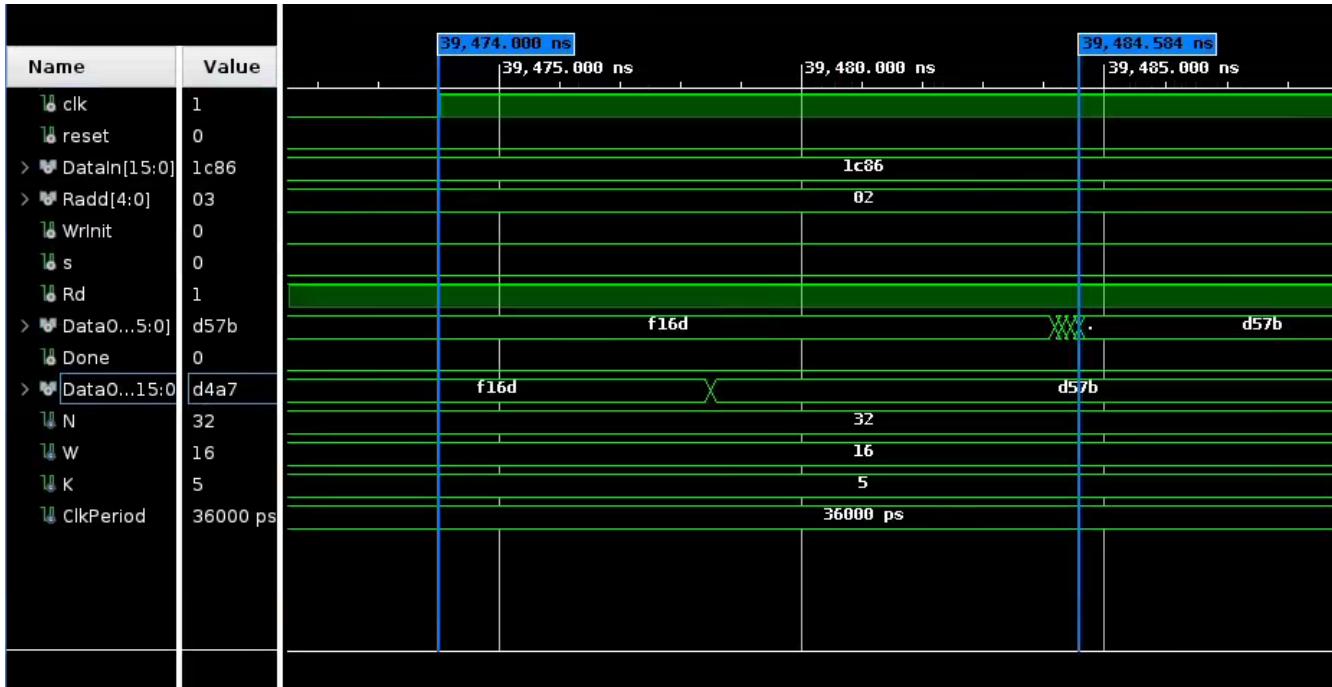


Figure 45. Case iii – Markers showing data out delay

Figure 42 provides a breakdown of post implementation resource utilization for the circuit.

Name	1	Slice LUTs (303600)	Slice Registers (607200)	Slice (75900)	LUT as Logic (303600)	Block RAM Tile (1030)	Bonded IOB (600)	BUFGCTRL (32)
Sorting		52		15	14	52	0.5	43
Sorting_controller (Controller)		2		5	3	2	0	0
Sorting_datapath (Datapath)		51		10	14	51	0.5	0

Figure 46. Case iii – Post Implementation Resource Utilization

Deliverables 4.e-h:

Define N as the number of data points to be sorted, and w as the bit width of the datapoints. The following hardware resource utilizations are dependents on N and w because...

LUTs: The number of look up tables used by the design to implement the comparator functions. The width of the comparisons is dependent on w when comparing data values to each other, as the width of the data values increase so must the size of the LUTs, therefore increasing utilization. It is also dependent on N because N effect counter sizes, and all comparison functions that use the counter indexes.

FFs: The number of flip-flops, or registers changes only with w since the sorting algorithm only works with comparing and swapping two values at a time, regardless of N . The actively used values are registered, the size of the registers is dependent on how big a single data value is, therefore it changes with w .

BRAMs: RAM utilization is dependent on both w and N . It is dependent on N because the algorithm requires all N values of *datain* to be stored in memory before sorting starts, clearly to store N values, more utilization is required to store *datain* for larger values of N . For w , like with N if w is larger the value the total number of bits needing to be stored increases therefore as w grows so does RAM utilization.

IOBs: Since IOBs designate Input output blocks, they are dependent on w because *datain* and *dataout* are reported as parallel buses, if w is larger the buses must be wider, therefore using more I/O. N effects the number of I/O needed because one of the circuit inputs is an address for the internal RAM to store all values of *datain* if N grows more unique address are needed, and the address bus must have more bits to represent more addresses. Therefore, the I/O usage as a function of N changes at a rate of $\log_2(N)$.

Changing the value of N will only have an effect on the minimum clock period if N is large enough the also increase the number of bits needed to represent the address. If both N and $N+1$ can be represented with k bits than the two cases will have the same minimum clock periods. However, if $N+1$ requires $k+1$ bits to represent all memory addresses the design for the $N+1$ case will have a larger minimum clock period as the design will require more resources. This can be see in the timing differences between cases i and ii since N changes from 16 to 32, requiring 4 and 5 address bits respectively.

Increasing w will always increase the minimum clock period since w reading and writing w bits to the RAM and performing comparison functions on the data values will always be slower when using data of a wider width. Likewise, decreasing the w decreases the minimum clock period.

The major observable difference between the post-synthesis and post-implementation timing simulations is that during the synthesis simulation bit bus changes are happening instantaneously. The simulator accounts for delays between stages of the circuit so values change after some delay from the clock, unlike in the behavioral simulator. In the post implementation timing simulation, we can observe bits on a bus changing more naturally, that is they don't all change at the same time. And during this period of change the bus value can take unexpected values for short periods of time while individual bits are changing. This is demonstrated Figure 47 and Figure 48 showing the same transition of *dataout* for the same case i circuit, for the post-synthesis and post-implementation simulations respectively.

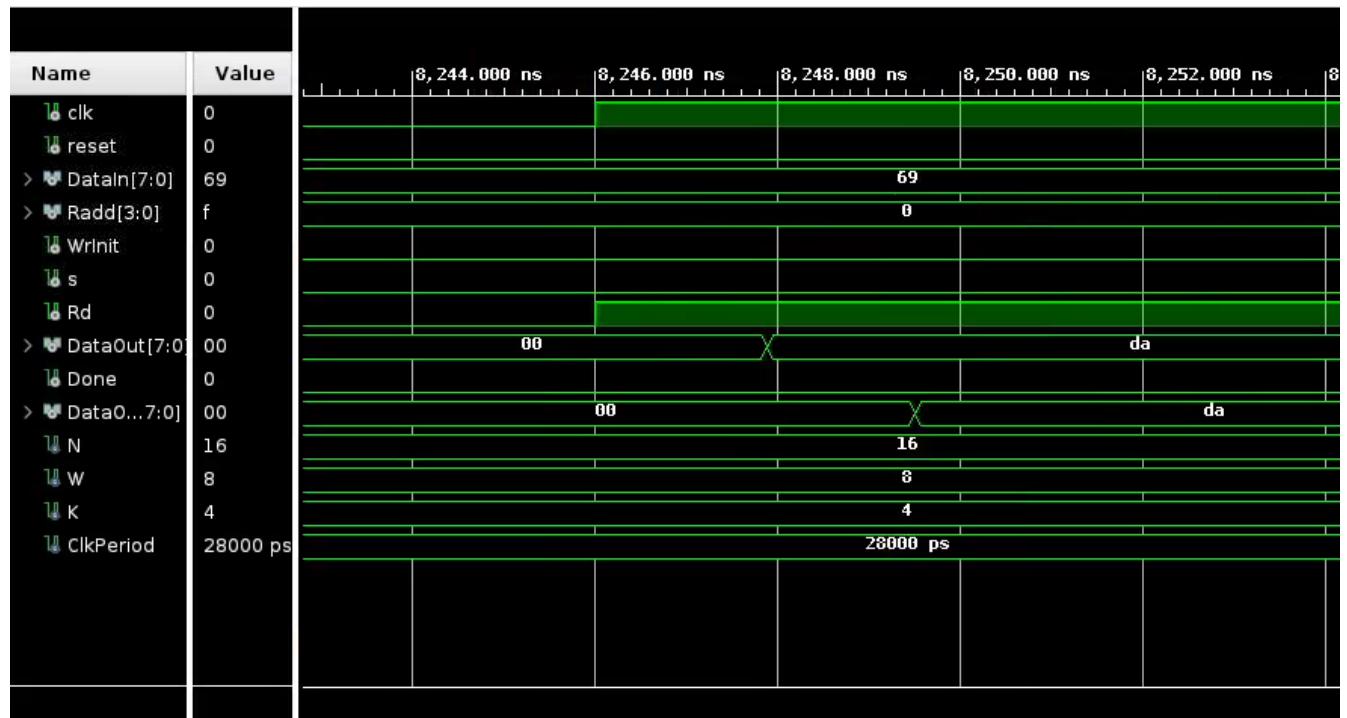


Figure 47. Post Synthesis Instantaneous bit change

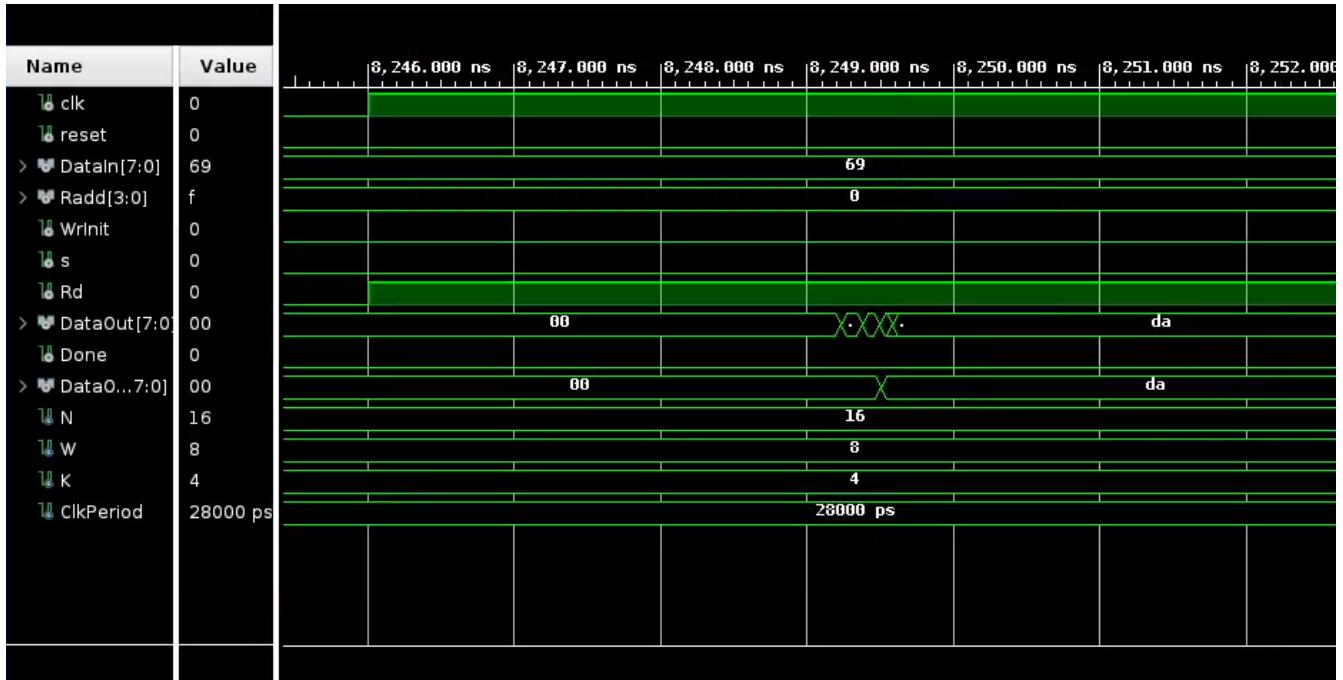


Figure 48. Post Implementation bit change