

ECEN 5863: Programmable Logic Embedded System Design

FA23 Project 2 Technical Report

Matt Hartnett & Caleb Provost

Table of Contents

Executive Summary.....	2
Objectives.....	2
Module 1: SmartTime for SmartFusion2.....	2
Module 2: SF2MakerDemo.....	2
Module 3: Load-Testing FPGAs with Counters.....	3
Procedure.....	3
Module 1: SmartTime for SmartFusion2.....	3
Module 2: SF2MakerDemo.....	3
Module 3: Load-Testing FPGAs with Counters.....	3
Module Test Results.....	4
Module 1: SmartTime for SmartFusion2.....	4
Module 2: SF2MakerDemo.....	4
Module 3: Load-Testing FPGAs with Counters.....	5
Lessons Learned.....	6
Conclusions.....	7
Appendix.....	8
Module 1: SmartTime for SmartFusion2.....	8
Module 2: SF2MakerDemo.....	21
Module 3: Load-Testing FPGAs with Counters.....	32

Executive Summary

Through the implementation of Project 2 our team has successfully gained a working knowledge of Libero and the SmartFusion2 FPGA, as well as the logic capabilities of different FPGA boards.

In Module 1, we created a 32 bit shift register and created a full timing constraint set. This was an introduction to the SmartFusion2 SoC development board and the Libero software suite, including SmartTime, MicroSemi's timing constraint and analyzing tool. The result of this module was a fully constrained design that met timing closure requirements.

In Module 2, we spent more time with the SmartFusion2 SoC by creating a microcontroller and support structures on the SoC. This module also included programming the FPGA and executing C code on the microcontroller design. We were not only able to execute code, but also run a debugger that let us view register and memory values.

In the final module, we started by creating an adjustable length chain of 16 bit counters. The goal of this module was to find how many counters could fit into different FPGAs. We optimized the design for space and then determined the maximum number of counters for the following FPGA development boards: the MicroSemi SmartFusion2, the Altera DE10-Lite, and the Altera DE1-SoC. We found that the Altera devices were both more capable of fitting counters in, with the DE10 fitting 2160, and the DE1 fitting 3054. In last place was the SmartFusion2, with only 391 counter blocks. In conclusion, this project helped teach the fundamentals of Libero, the SmartFusion2, timing closure, and logic optimization.

Objectives

Our team successfully met the following objectives:

Module 1: SmartTime for SmartFusion2

The primary objective of this module is to practice timing closure using the SmartTime Static Timing Analyzer and become familiar with the Microsemi FPGA development flow. Additionally, create hardware with the SmartDesign tool and evaluate system timing with several hardware models.

Module 2: SF2MakerDemo

The objective of Module 2 is to use Libero to develop software generated from the hardware configuration. The hardware design tool SmartDesign will be first used to develop the FPGA configuration which will then be used to configure and set up the SoftConsole IDE for software development. Debugging both the hardware and software within the same Microsemi development flow is key to this module.

Module 3: Load-Testing FPGAs with Counters

This module focuses efforts into optimization of hardware configurations. Using both Libero and Quartus, a 16-bit counter will be designed and loaded on each of the three devices, DE1-SoC, SmartFusion2, and the DE10-Lite. The goal is to fill as many of the 16-bit counters into the boards as possible.

Procedure

Module 1: SmartTime for SmartFusion2

Using Libero, we had to implement several circuits and constrain them. They consisted of a 32-Bit Shift Register, a Dual Clock Edge Design, and a 16-Bit Binary Counter. The process of constraining the inputs, outputs, and clocks required compiling the circuits and synthesizing the logic for timing analysis while making iterative approaches of getting the clocks to no longer have any negative slack.

To prove our success, we additionally needed to run reports and estimate the percent utilization of logic elements for each of the FPGA boards. Which can be found in the Module Test Results sections.

Module 2: SF2MakerDemo

Using the SmartFusion2 Maker Kit, we needed to configure and constrain the board's GPIOs for LED control. This required configuring and linking the correct FPGA's peripherals and exporting the required bitstream configuration files so that SoftConsole IDE could interpret how to compile for our configured target.

Upon successful completion of this module, we needed to generate timing reports and verify hardware - namely the LEDs - were operating as expected. The results of the fMAX report and percent utilization for this project can be found in the Module Test Results section.

Module 3: Load-Testing FPGAs with Counters

The first step of this process is to create a synchronous 16 bit counter module that can be instantiated into another design as part of a larger hierarchy. A counter such as this is relatively simple, as it only requires a single block that counts up on the rising edge of the clock so long as the reset is low.

In order to create the long chain of counters that was adjustable, we utilized the “generate” capabilities of Verilog to create an n-number of counters all tied in a long chain. This allowed us to change the parameter n to find the absolute maximum value of counters that could fit in a given FPGA. In order to ensure that the synthesizer of Quartus or Libero didn't optimize away the counters, we tied the overflow output of one to the enable input of the next.

In order to determine the appropriate number of counters to begin testing, we compiled with 100 counters and estimated the logic utilization percentage. The more detailed results of those estimations are shown in the Module Test Results section below. Since some of the logic and routing did not scale with the quantity of counters and since the logic utilization approximation is not fully accurate to the limits of the chip compared to the requirements of the design, the first guess was never the final guess. However, this loose guess did help as a reasonable starting point that minimized the amount of time wasted on checking counter values.

Once the first guess was taken and a bound was established (either high if it failed to compile, or low if it succeeded), our second guess was different by a significant amount (approximately a 10% change in counter value) to establish a second bound. With the two bounds, the maximum could be determined in a logarithmic manner by testing the average of the two bounds and then using the result of that compilation as the new upper or lower bound. This method saved a significant amount of time, as guessing and checking would've taken much longer. When we were compiling the DE1-SoC with thousands of counters, each compilation would take around 10 minutes to complete, meaning that every method optimization was crucial.

Module Test Results

Module 1: SmartTime for SmartFusion2

See section Procedure: Module 1 for the details pertaining to these results:

- Shift 32:
 - Fmax - 505.051 MHz; Logic element utilization - 0.26%
- Shifter:
 - Fmax - 295.334 MHz; Logic element utilization - 0.33% with 0.31% utilization of Flip Flops and 0.07% utilization of 4LUTs
- 16-bit Counter:
 - Fmax - 377.644 MHz; Logic element utilization - 0.15% with 0.13% utilization of Flip Flops and 0.15% utilization of 4LUTs

Module 2: SF2MakerDemo

We were able to program the SmartFusion2 Maker Kit, even into the SmartFusion2's eNVM. The % utilization of logic is: 0.02% of DFFs, 9.52% of User I/O, 12.50% of chip globals.

We've determined that there must be a typo in the instructions as we were not able to find R13 and R15 in the Register's list in the debug console. The memory results for that range are provided in the rightmost column in Figure P2M2 19.

Upon programming the board (before the switches were added), the board would blink all 8 LEDs in succession before all turning off and then incrementally illuminating until all were on, then decrementally turning off. This behavior was as expected, except for using switch SW3 would reset the program. When adding switches SW1 and SW2 as the toggling direction feature, SW2 would freeze up the program and SW3 would still reset the board's program, but SW1 did behave as intended. Figure P2M2 20 and 21 demonstrate the LEDs working.

Module 3: Load-Testing FPGAs with Counters

As a summary of our results, we found that the following FPGAs allowed the following number of counters before failing the place and route step of the compilation process:

- SmartFusion2: 391
- DE10-Lite: 2160
- DE1-SoC: 3054

As mentioned above in the Procedure section, the first step of determining the maximum number of counters was a rough approximation based on the percent utilization of logic after compiling with 1 and 100 counters. The data and the conclusions of these approximations are shown below:

Microsemi SmartFusion2:

- One instance of the counter took 22 4LUTs/LEs with 16 DFFs. I expect some overhead to be added with the generative code's instantiation of more counters. So I expect 100 counters to take about 2210 4LUTs/LEs and about 1610 DFFs. As seen below, it took less than the amount I originally expected. This is likely due to optimization algorithms while compiling.
- If 100 counters takes 2195 4LUTs/LEs and 1600 DFFs, I expect to be able to fit about 550 counters in this chain of counters. However, incrementing the amount of counters above 330 (so far) keeps failing the compilation during the fitting. The below image shows the total utilization at 330 counters:

Altera MAX10 DE10-lite:

- 100 counters resulted in about 4% utilization of total logic elements:
- We're able to fit 500 with a resulting change of 8405 more logic elements used. This means there's about 22 logic elements used per addition of a counter. If the placement is consistent, we should be able to fit roughly 2260 counters onto the DE10-Lite.

Altera Cyclone V DE1-Soc:

- Starting the same code onto the DE1-SoC with 100 counters resulted in a total utilization of 1099 ALMs.
- With 500 counters added resulting in 5501 ALMs utilized, the difference of 4402 ALMs utilized for 400 more counters, we expect to use 11 ALMs per counter resulting in about 2915 counters that can fit onto the DE1-SoC. That is, if the fitting will allow it.

As discussed earlier, the next step of the process was finding an upper and lower bound of the numbers of counters and then logarithmically approaching the final answer. The results of this method are as follows:

- SmartFusion2: 391
- DE10-Lite: 2160
- DE1-SoC: 3054

In order to verify that the counter design operates as expected, RTL simulations were run to observe the behavior of the counter chain.

- RTL simulation for the SmartFusion2 Maker Kit is in Fig. P2M3 6
- RTL simulation for the DE10-Lite board is in Fig. P2M3 7
- RTL simulation for the DE1-SoC board is in Fig. P2M3 8

As part of the module, we recorded the Fmax of the boards with the maximum number of counters. This was important because it helped indicate the balance between design density and speed. The fastest frequency was on the SmartFusion2, but it was also the FPGA with the least number of counters.

- The SmartFusion2 was able to report an Fmax of 245.158 MHz.
- The DE10-Lite was able to report an Fmax of 125.75 MHz.
- The DE1-SoC was able to report an Fmax of 213.9 MHz.

The logic utilization is another key design metric that can help a designer understand the shortcomings or advantages of different designs and boards.

- The SmartFusion2 reported a total utilization of 71.14% of its 4LUTs, 51.77% of its DFFs, and 71.3% utilization of its Logic Elements.
- The DE10-Lite reported 91% utilization of its 49760 total logic elements.
- The DE1-SoC reported 100% utilization.

As stated in step 6, each board progressively had larger logic elements per configurable segment, which allowed more of our counters to fit into the “logic elements”. It got to a point where there was more available logic in the smaller boards, but the type of logic needed to add more counters was already utilized, and thus no longer able to add more counters. The DE1-SoC for instance had way more of a complete logical unit that we could more completely utilize, which led to getting 100% utilization out of it.

Conversely, the more counters we fit into the boards, the worse the timing routing became, which is the reason why the smallest amount of counters, on the SmartFusion2, was also the fastest Fmax.

Lessons Learned

Timing constraints and timing closure are both critical concepts for FPGA design. In order to ensure that your design runs reliably and efficiently, appropriate constraints are required. While the tools are not perfect, it is important to be able to read the outputs of the various timing

analyzer software. In addition, it's crucial to understand what a timing violation is and how to modify your design or your constraints to meet timing closure.

It is possible to sacrifice an FPGA's speed in order to fit more logic into it, but cascading timing constraints while doing so can make this difficult to still meet overall timing requirements. It's best to iron out and test a smaller scale of logic before trying to scale up immensely.

Test benches using HDLs are a huge time saver. They not only make the timing simulations easier to set up, but they also make it easier to ensure that all your execution scenarios are getting covered. Being able to load the file and execute it for a new board goes a long way in speeding up development.

In addition to small scale testing of logic modules, it is helpful to fully understand the tools that are used to create an FPGA design. For example, there are settings in the Quartus synthesizer that allow for a greater optimization towards logic size as opposed to logic performance. This setting alone helped us achieve the highest possible number of counters within a design.

Conclusions

Our team was successful in closing timing constraints for various circuits on the SmartFusion2 Maker Kit, the DE10-Lite, and the DE1-SoC development boards. As we use the development tools, Libero and Quartus, we've become more familiar with how to meet timing requirements in addition to how to manipulate the clocks and circuits in order to get as close to a desired result as possible.

In creating a microcontroller for the SmartFusion2, we've gained the ability to use the Libero design suite to create a complex soft processor on a MicroSemi FPGA, similar to what we've learned with Quartus and Altera. We've successfully generated a Board Support Package project for the SmartFusion2 similar to how we did with Project 1. Using Libero to generate the software development environment required more debugging configurations, but we were able to successfully use the debugger in the end.

The optimization of and fitting challenge of the 3rd module was quite fun, even though it was time consuming. Although the SmartFusion2 only fitting 391 counters is kind of a disappointment, being able to fill over 3000 of the counters into the DE1-SoC board and utilize all 100% of its ALMs.

Appendix

Figures associated with each module's work are presented below as a summary. Not all work is shown with these images. See each associated module's report for a full set of work and how these images correspond with the work done in the modules.

Module 1: SmartTime for SmartFusion2

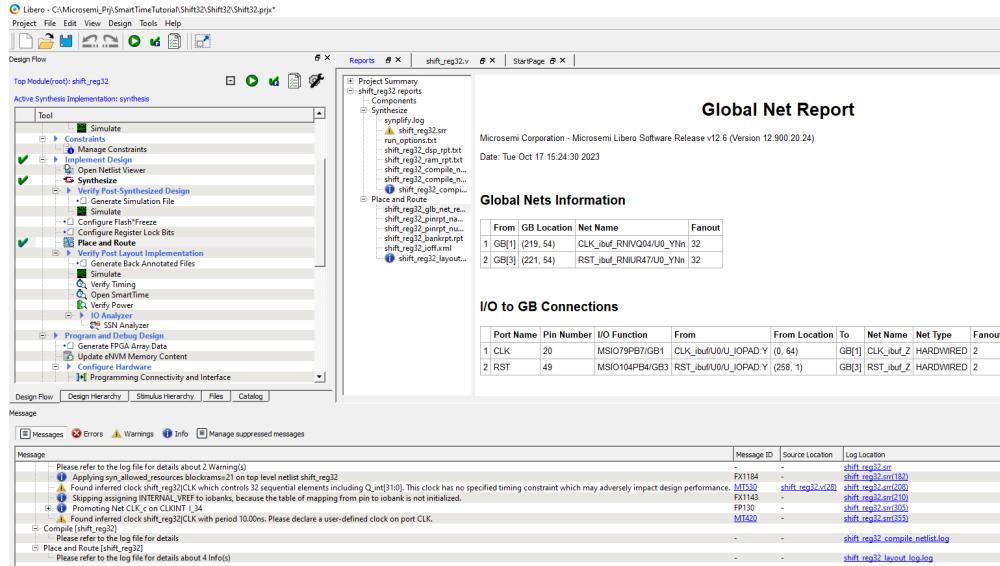


Figure P2M1 1: 32-bit Shift Register Setup Complete

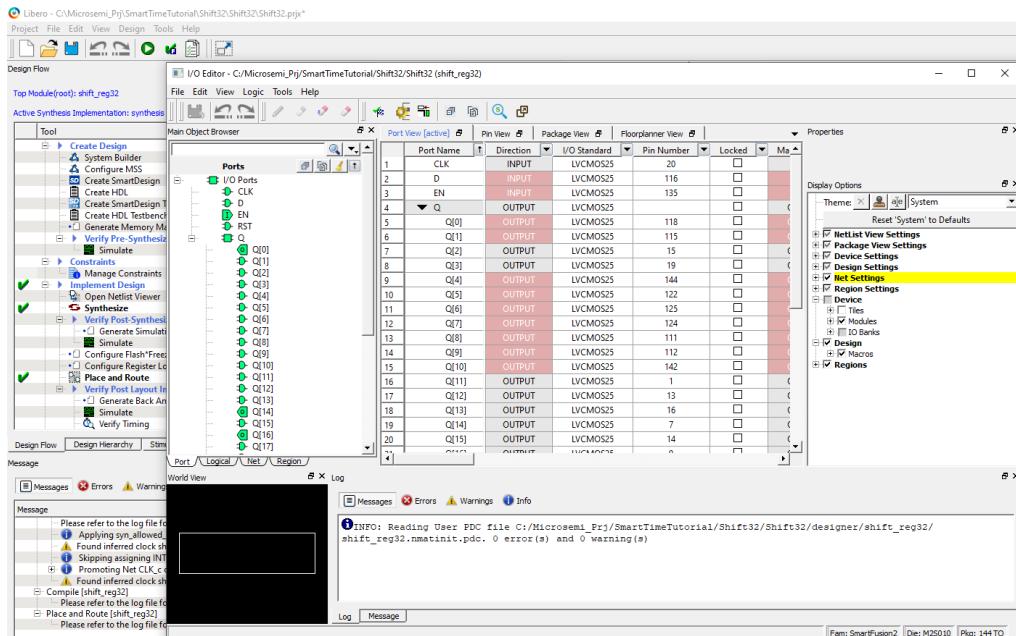


Figure P2M1 2: 32-bit Pins Assigned

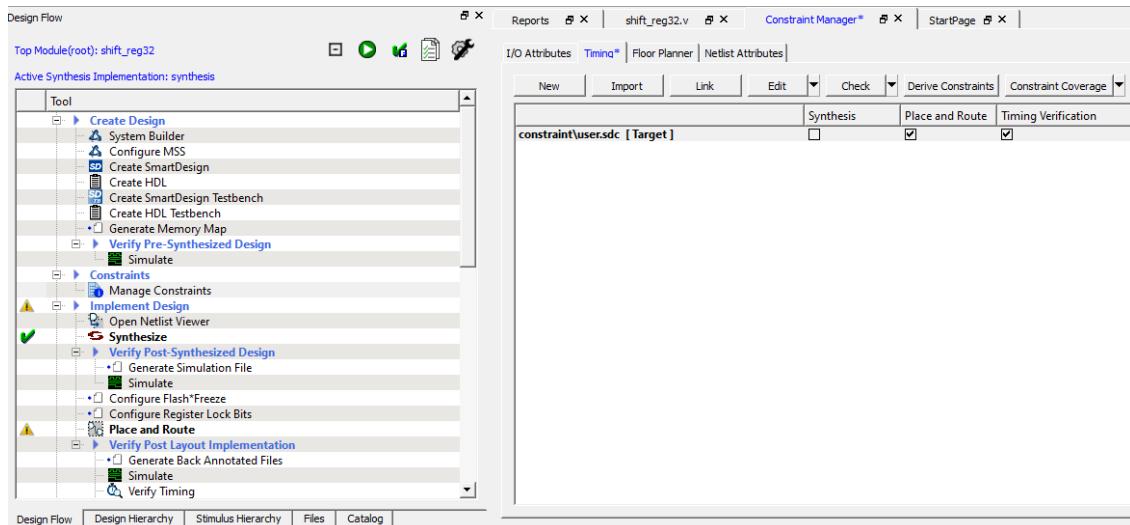


Figure P2M1 3: 32-bit Clock Constraints Added

Global Net Report

Microsemi Corporation - Microsemi Libero Software Release v12.6 (Version 12.900.20.24)
Date: Wed Oct 18 13:58:58 2023

Global Nets Information

From	GB Location	Net Name	Fanout
1 GB[1]	(219, 54)	CLK_ibuf_RNIVQ04/U0_YNn_GSouth	32
2 GB[3]	(221, 54)	RST_ibuf_RNIUR47/U0_YNn_GSouth	32

I/O to GB Connections

Port Name	Pin Number	I/O Function	From	From Location	To
1 CLK	20	MSIO79PB7/GB1	CLK_ibuf/U0/U_IOPAD/Y	(0, 64)	GB[1]
2 RST	49	MSIO104PB4/GB3	RST_ibuf/U0/U_IOPAD/Y	(258, 1)	GB[3]

Message

Message ID	Source Location	Log
FX1184	-	shift
MT330	shift.reg32.v(28)	shift
FX1143	-	shift
FP130	-	shift
MT420	-	shift
-	-	shift
-	-	shift
-	-	shift

Figure P2M1 4: 32-bit Timing Driven Place and Route Confirmation

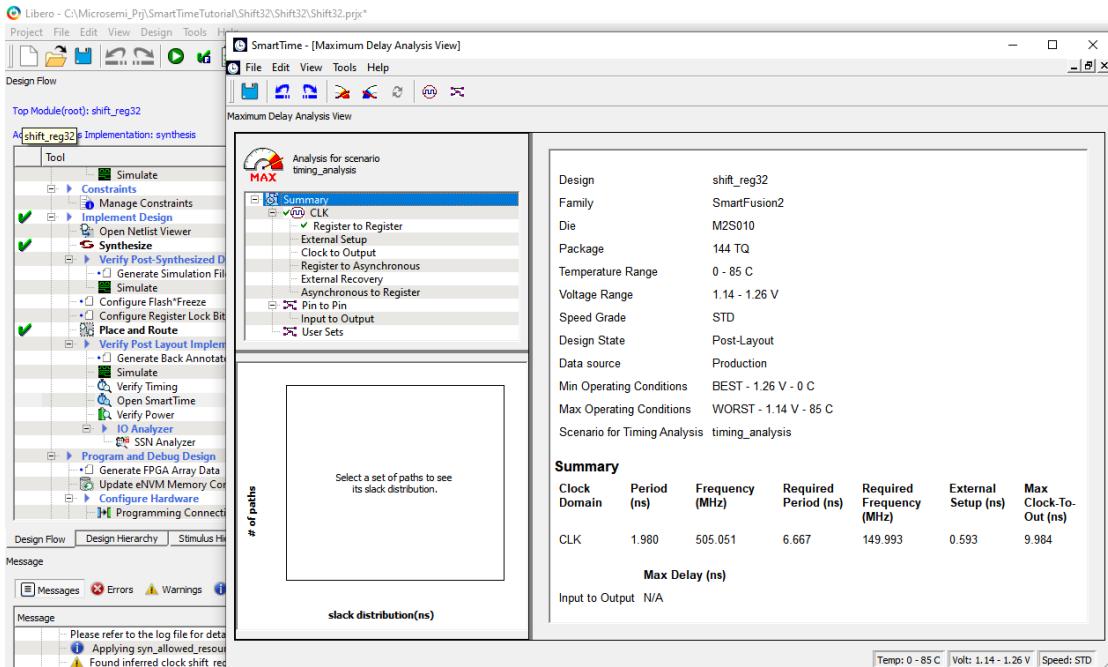


Figure P2M1 5: 32-bit Maximum Delay Analysis Summary

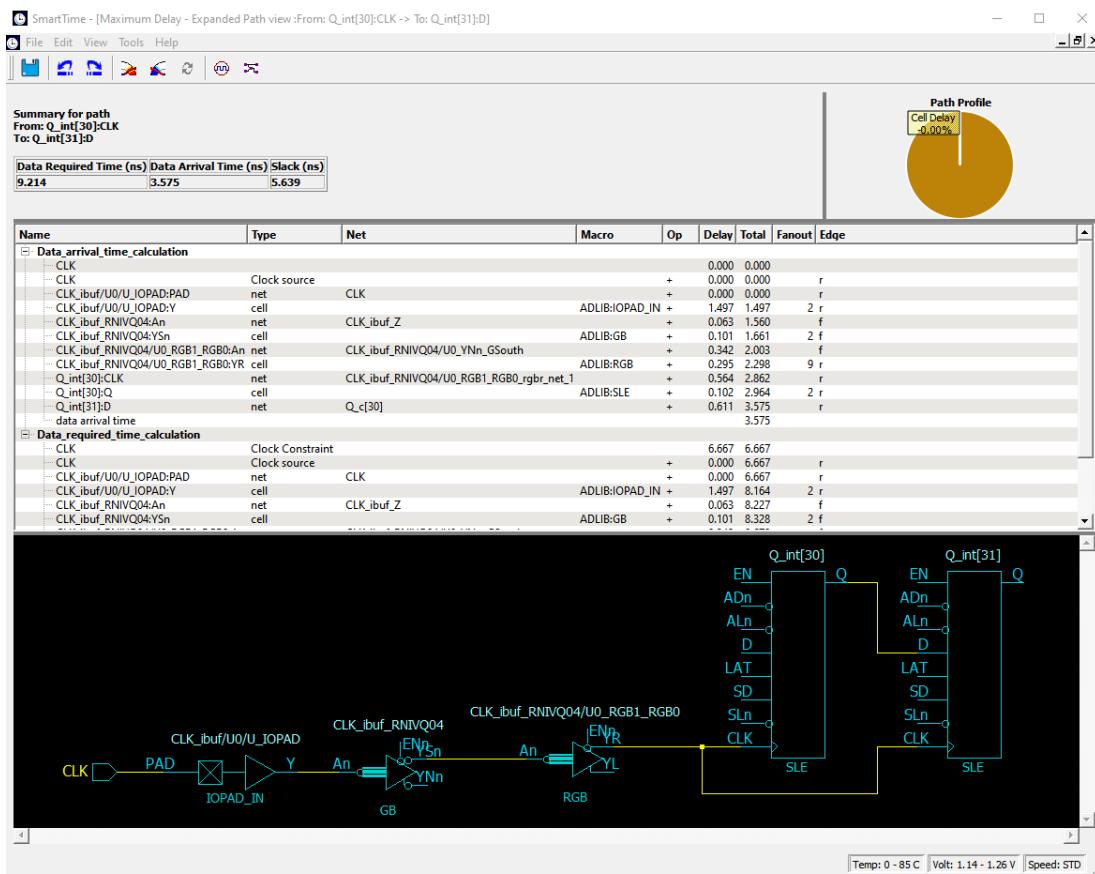


Figure P2M1 6: 32-bit Maximum's Expanded Path View Confirmation

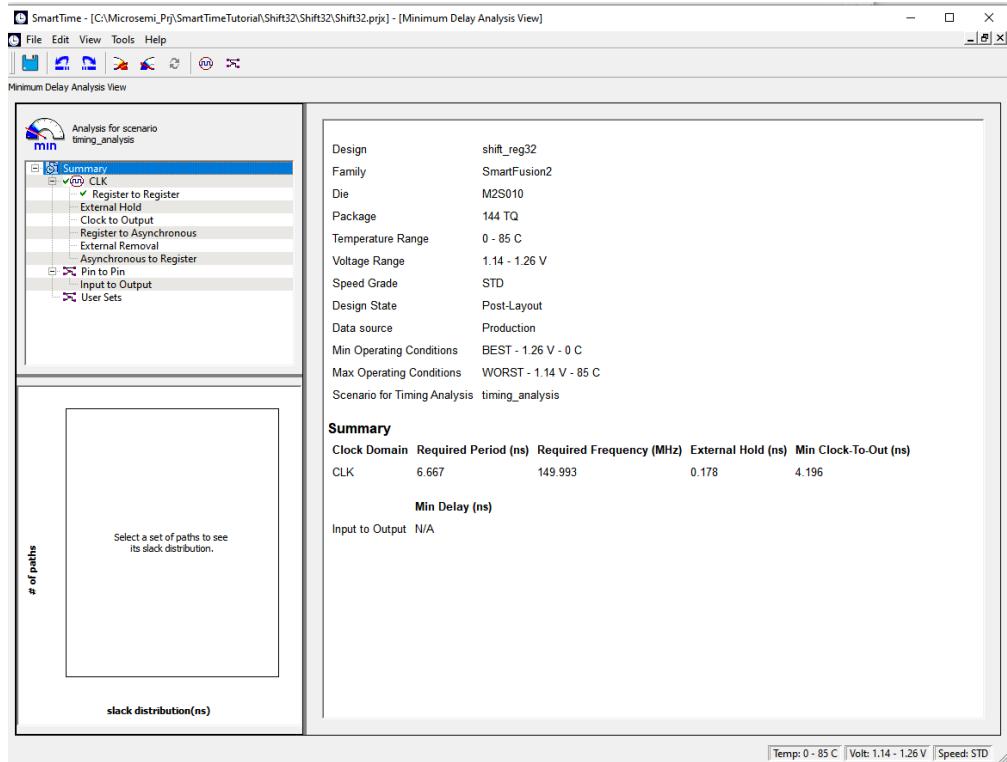


Figure P2M1 7: 32-bit Minimum Delay Analysis Summary

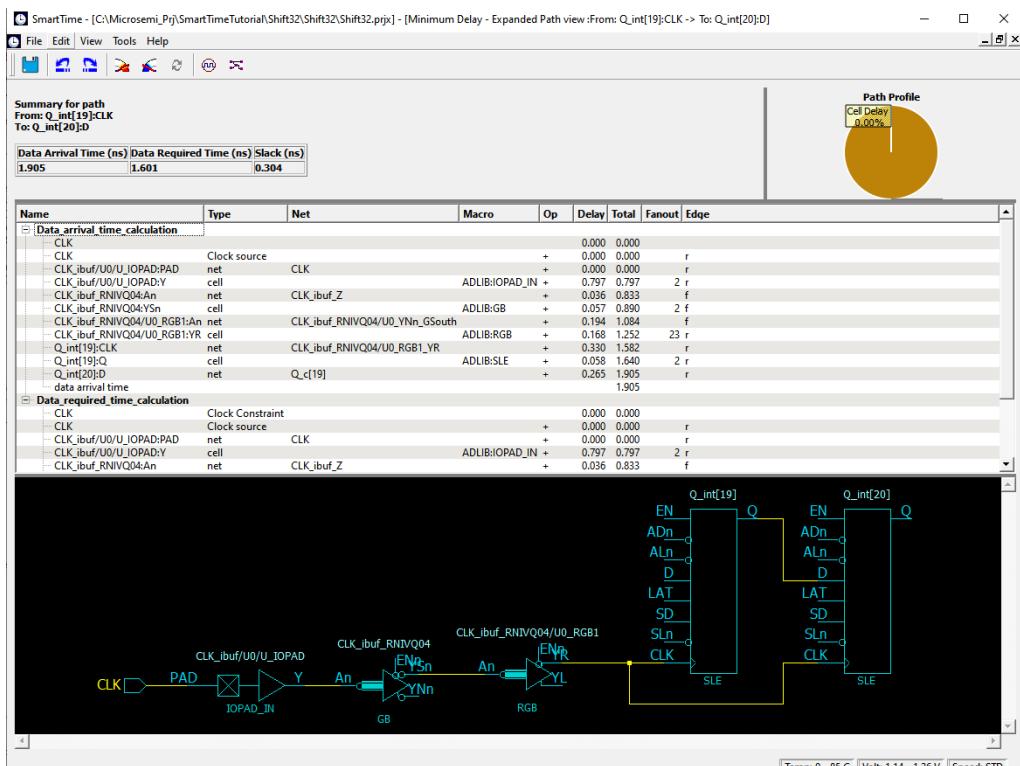


Figure P2M1 8: 32-bit Minimum's Expanded Path View Confirmation

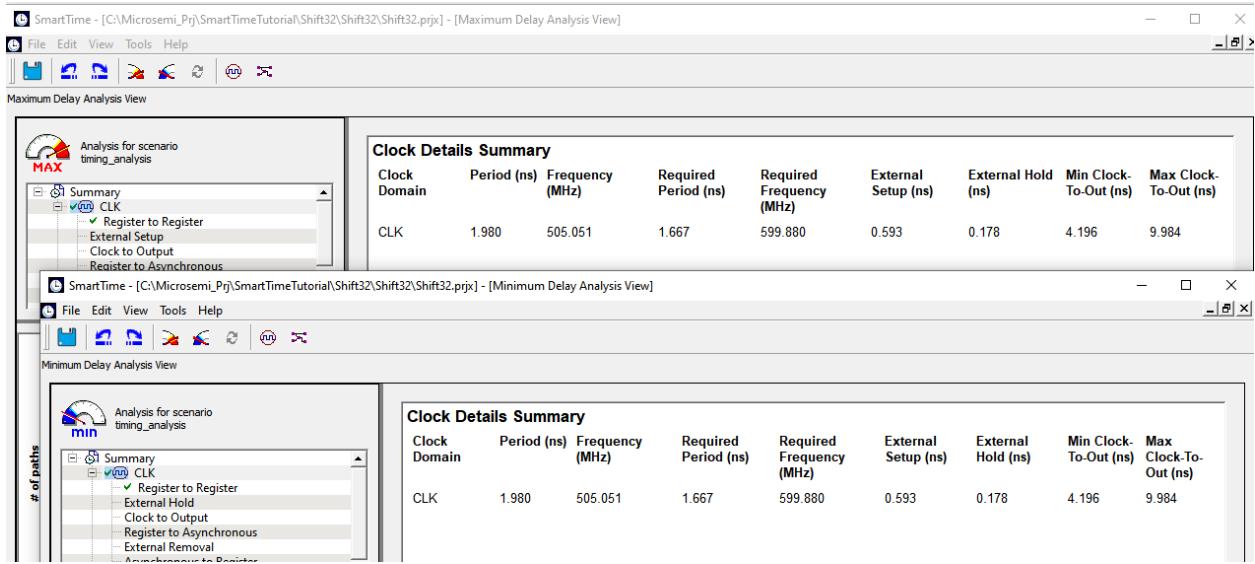


Figure P2M1 9: 32-bit Negative Slack Not Observed with 600 MHz CLK

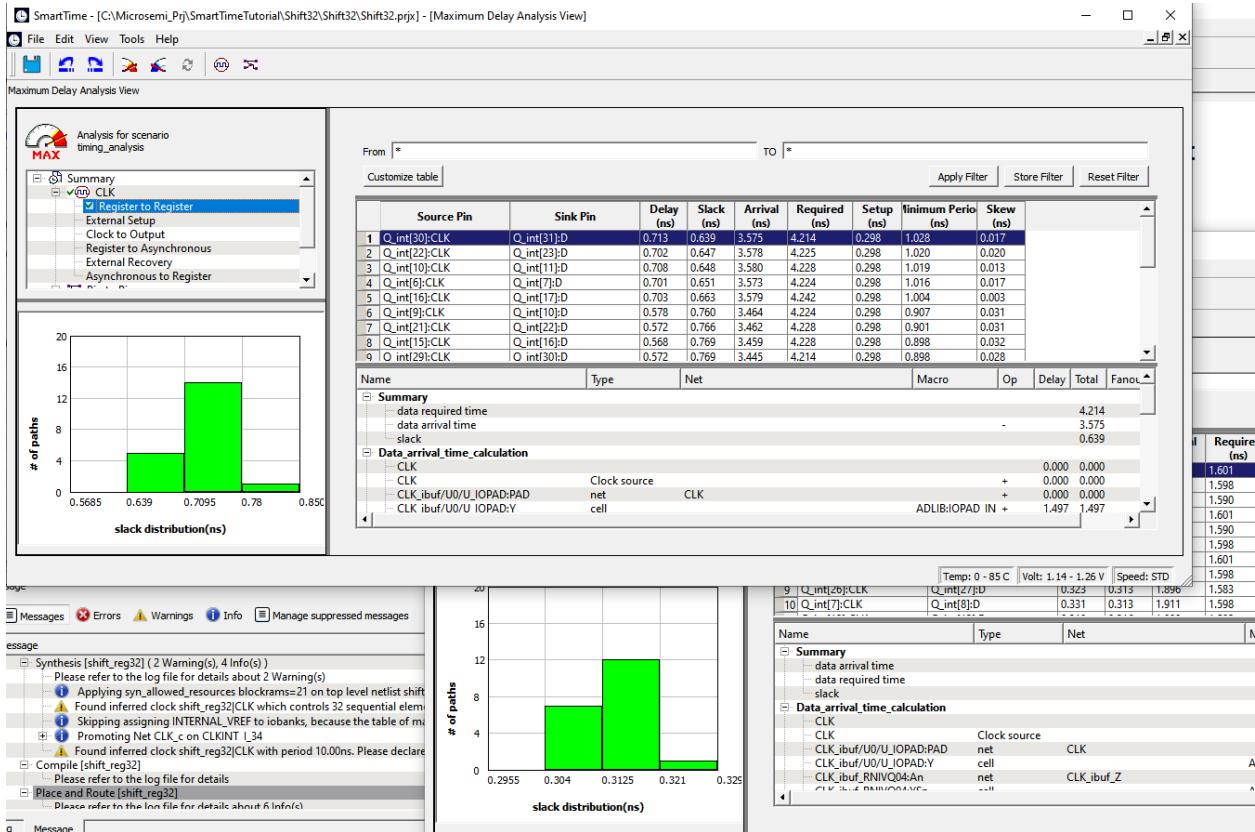


Figure P2M1 10: 32-bit Negative Slack Not Observed in Register to Register Menu

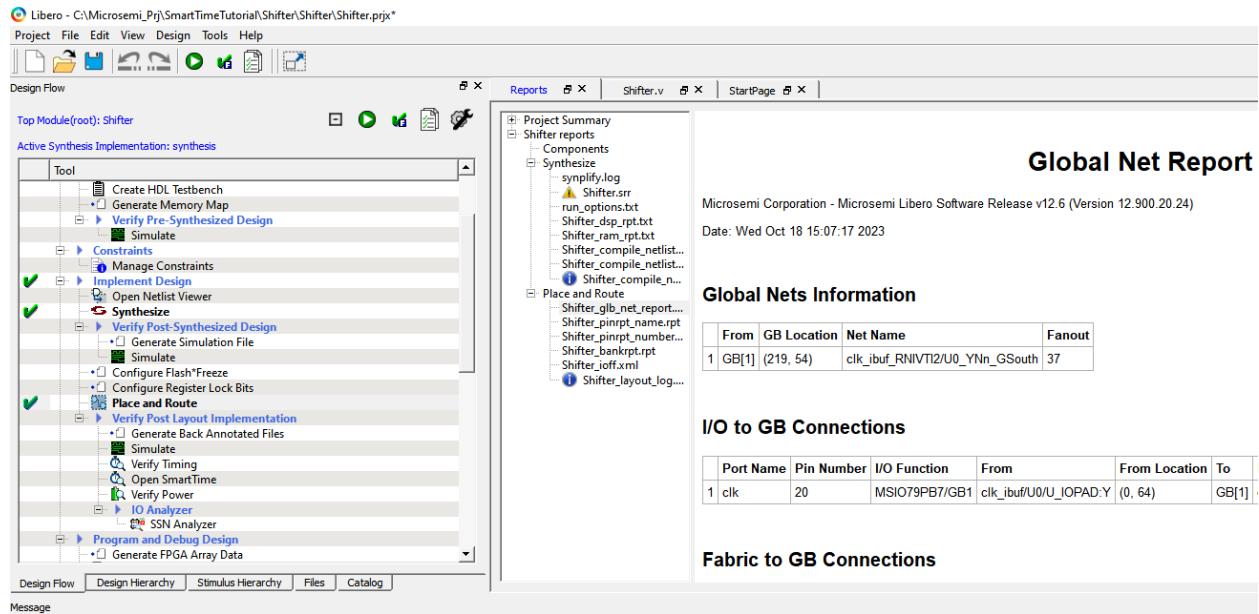


Figure P2M1 11: Shift Project Setup Confirmation

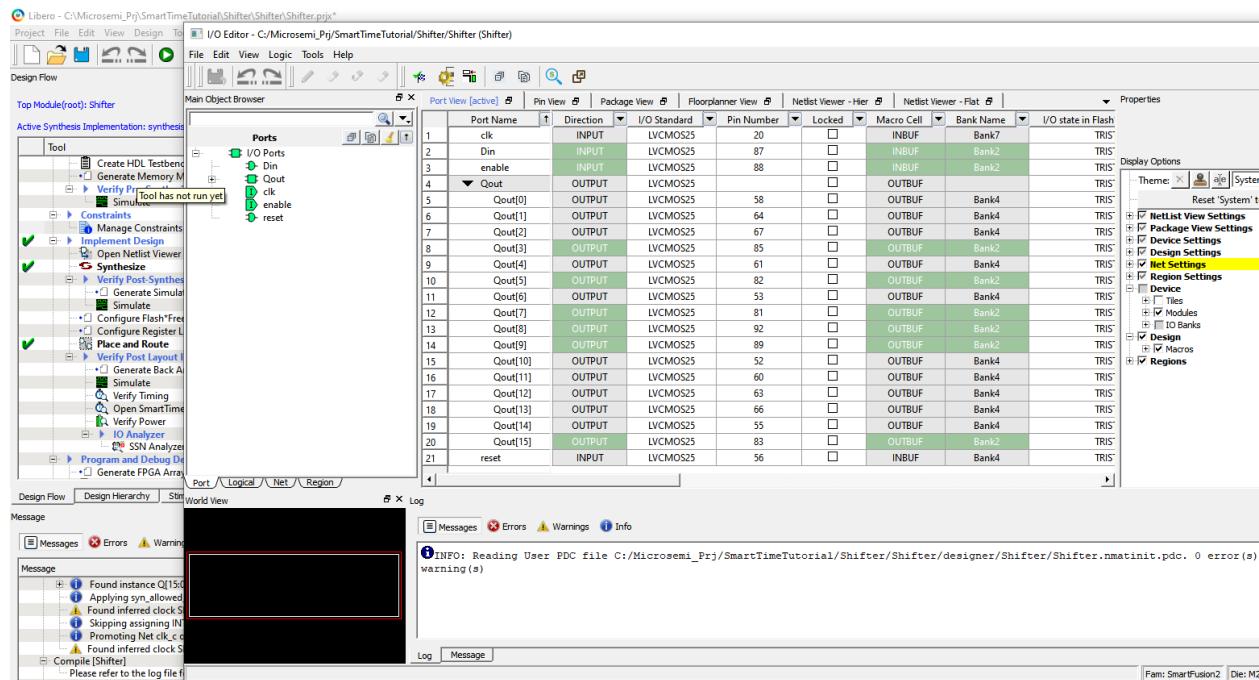


Figure P2M1 12: Shift Pins Configured

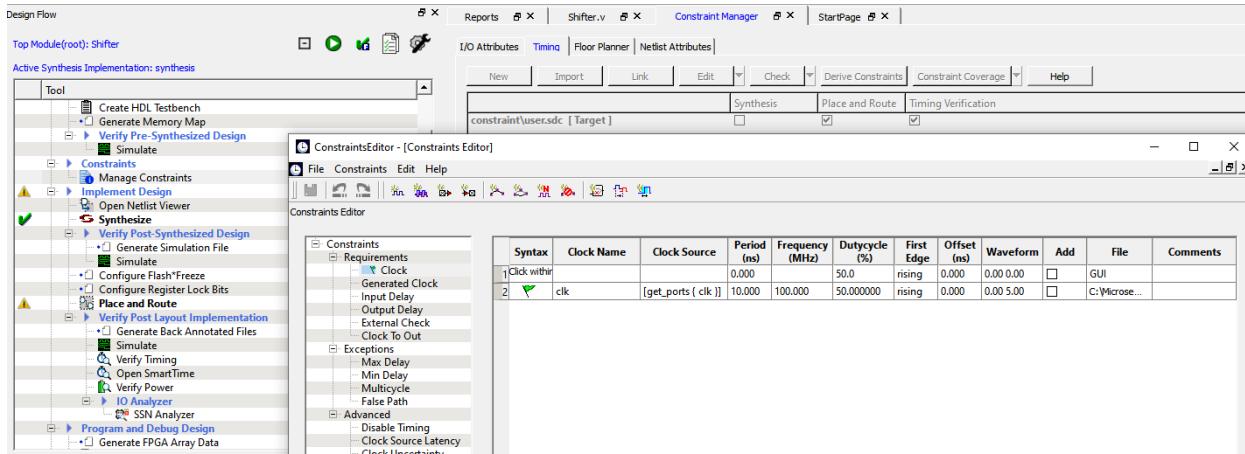


Figure P2M1 13: Shift Clock “clk” Added

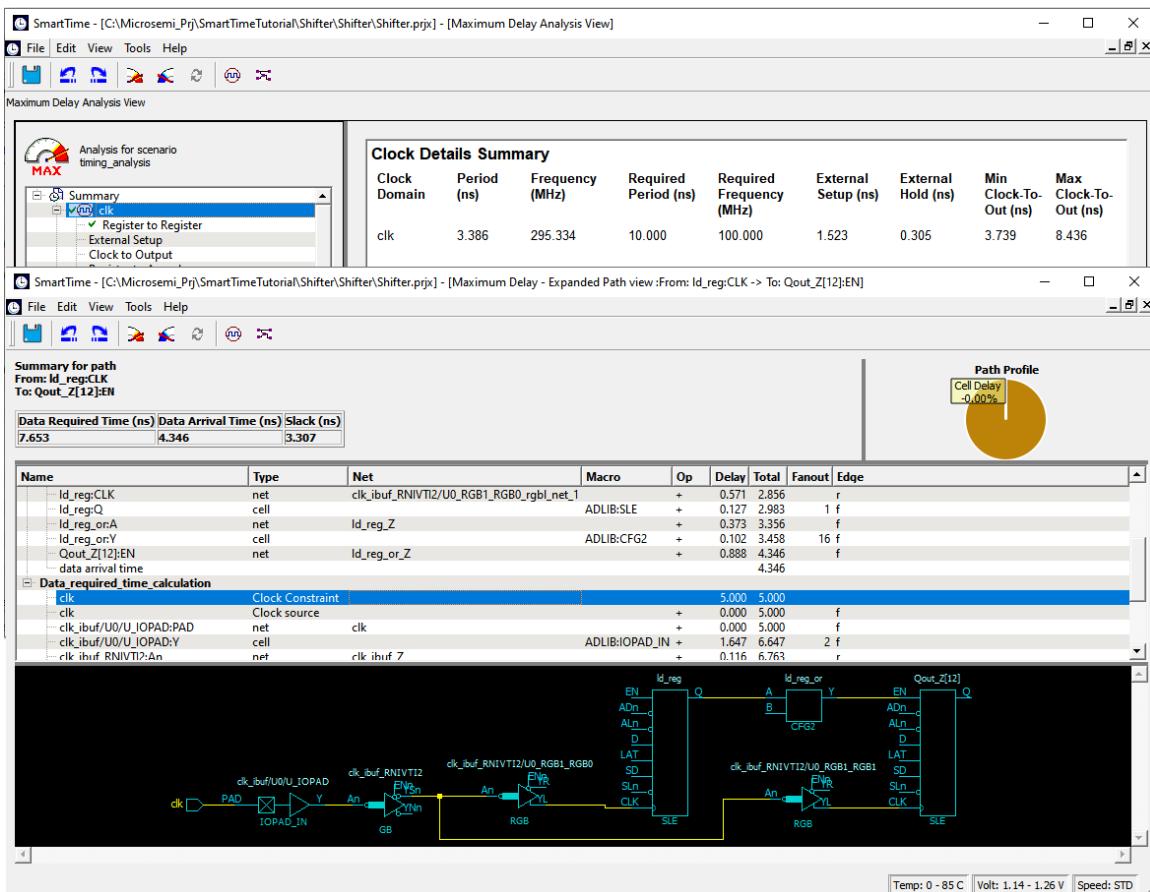


Figure P2M1 14: Shift Maximum Timing Delay Confirmed

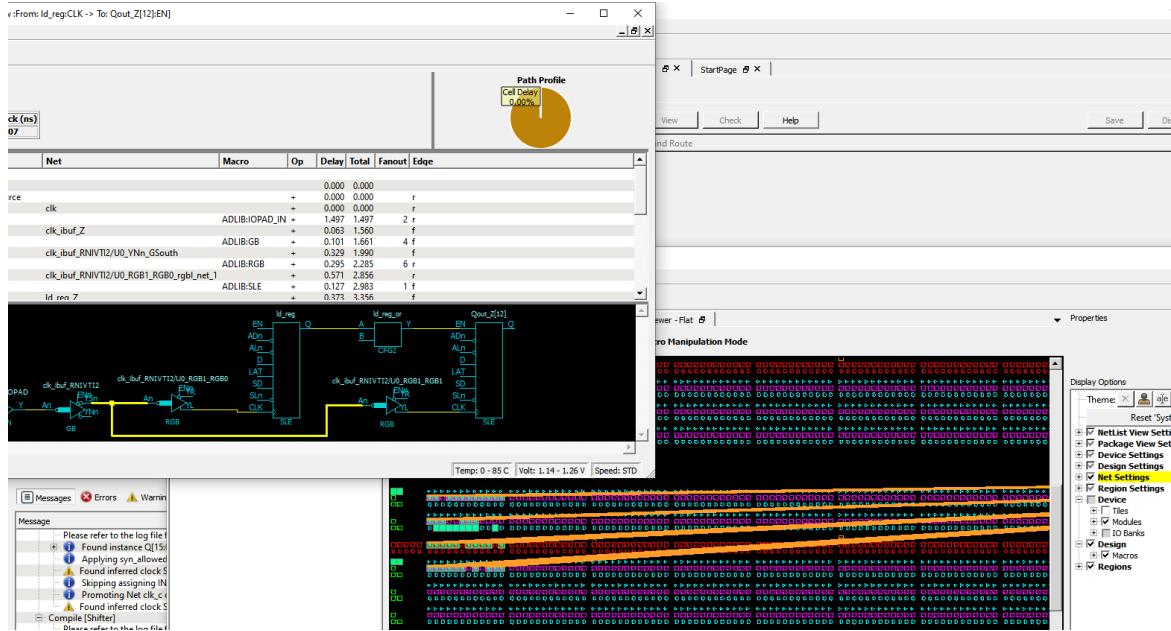


Figure P2M1 15: Shift Chip Planner Cross Probing Confirmation

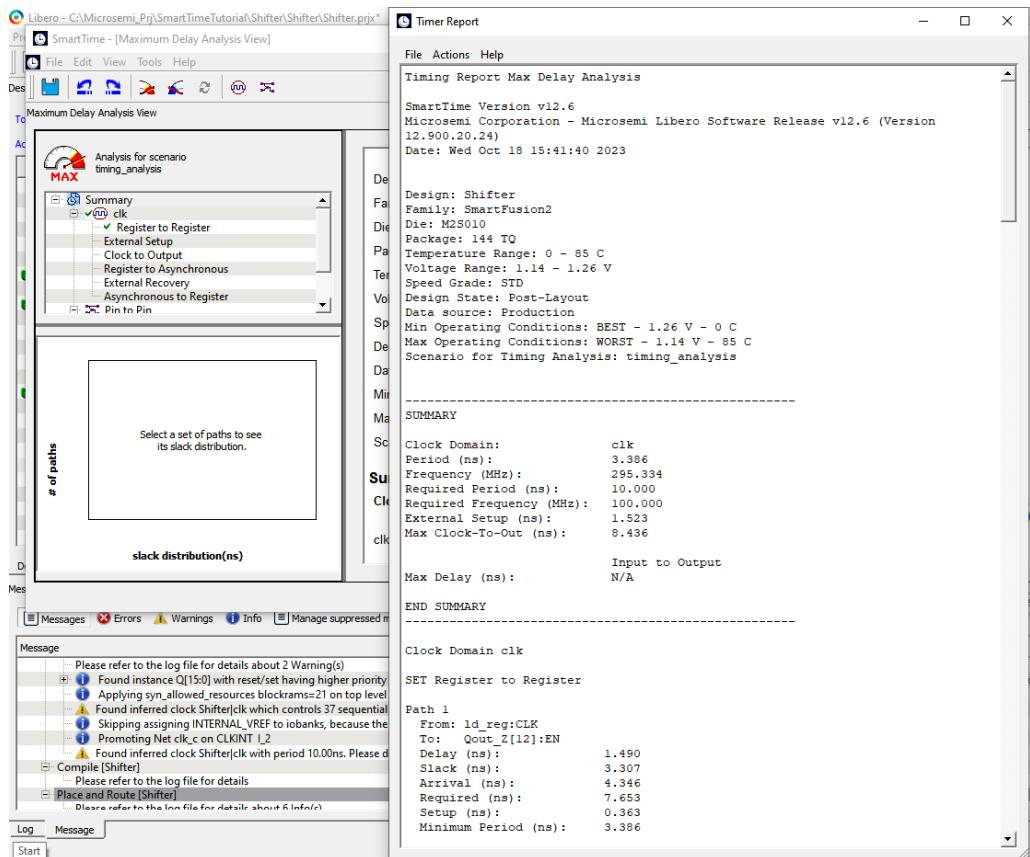


Figure P2M1 16: Shift Paths Timing Report

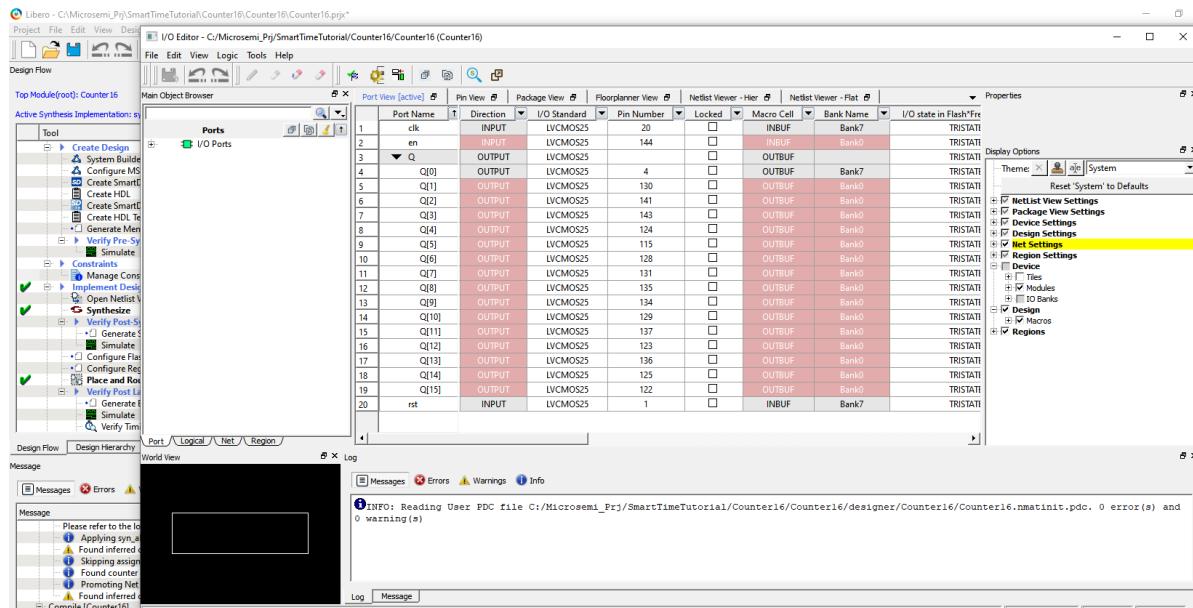


Figure P2M1 17: 16-bit Counter Setup Complete

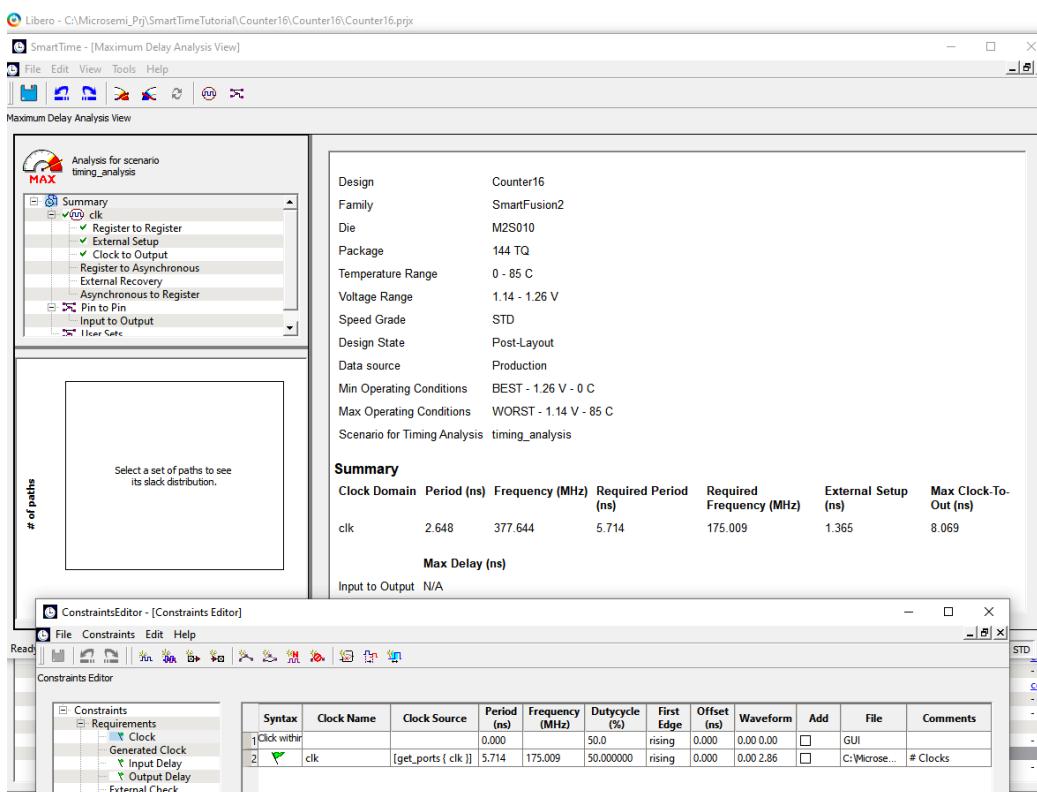


Figure P2M1 18: 16-bit Import Constraints Complete

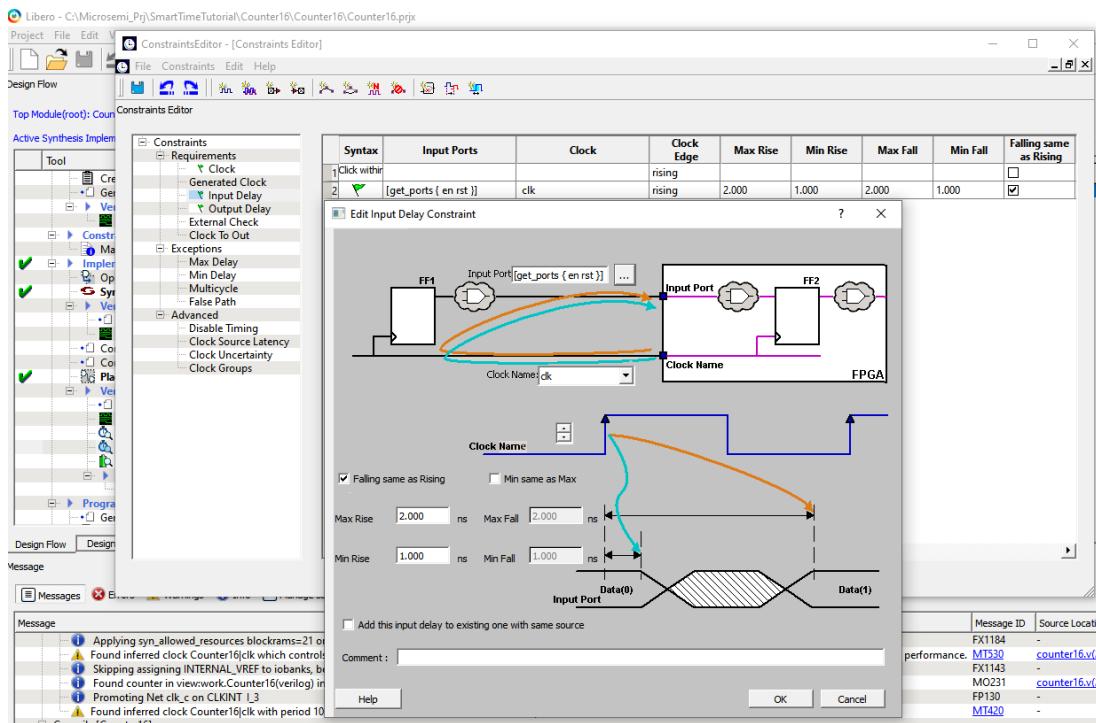


Figure P2M1 19: 16-bit Add Input Delay Constraint

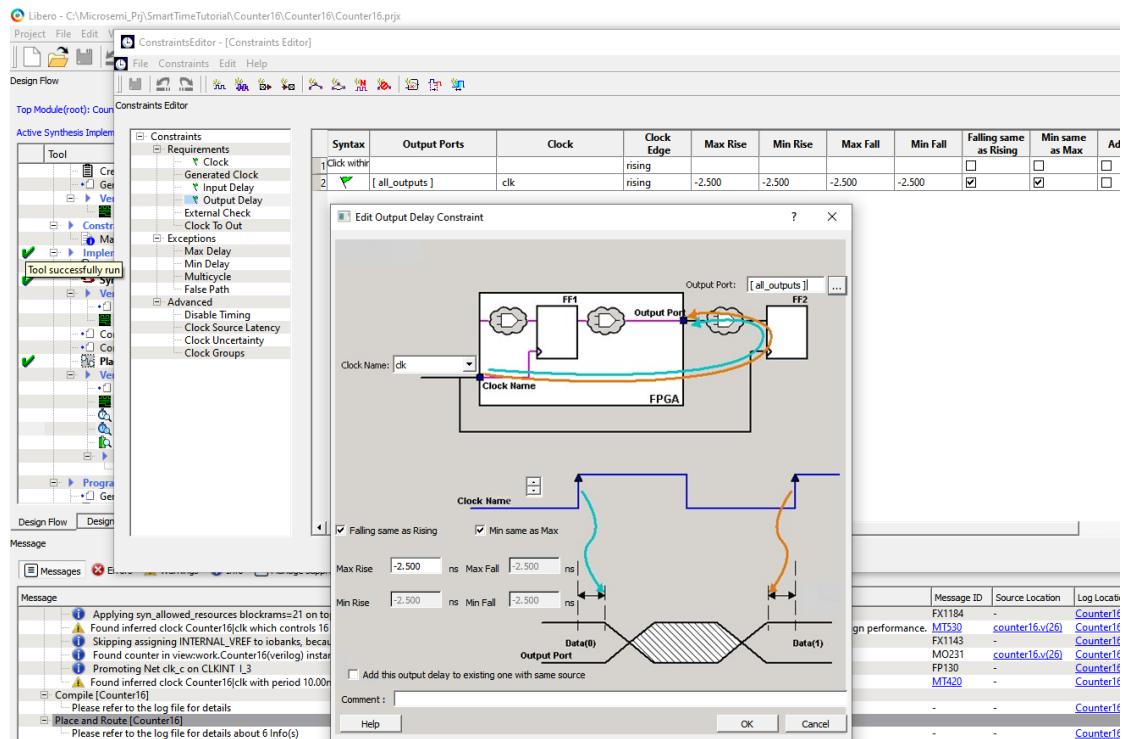


Figure P2M1 20: 16-bit Add Output Delay Constraint

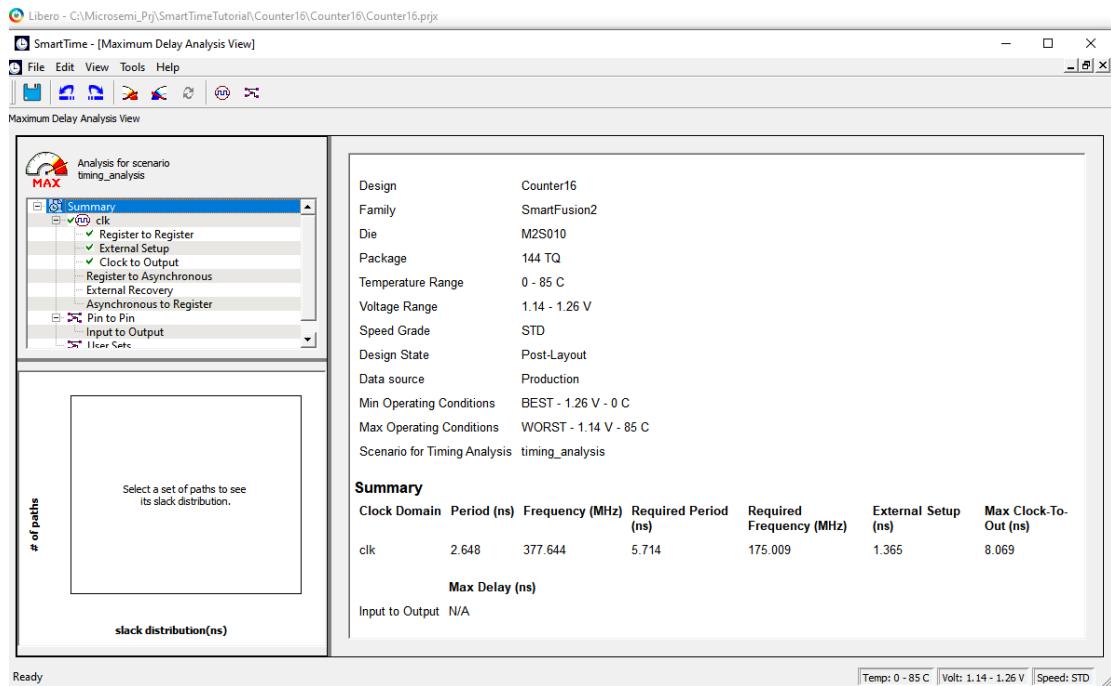


Figure P2M1 21: 16-bit Place and Route Results

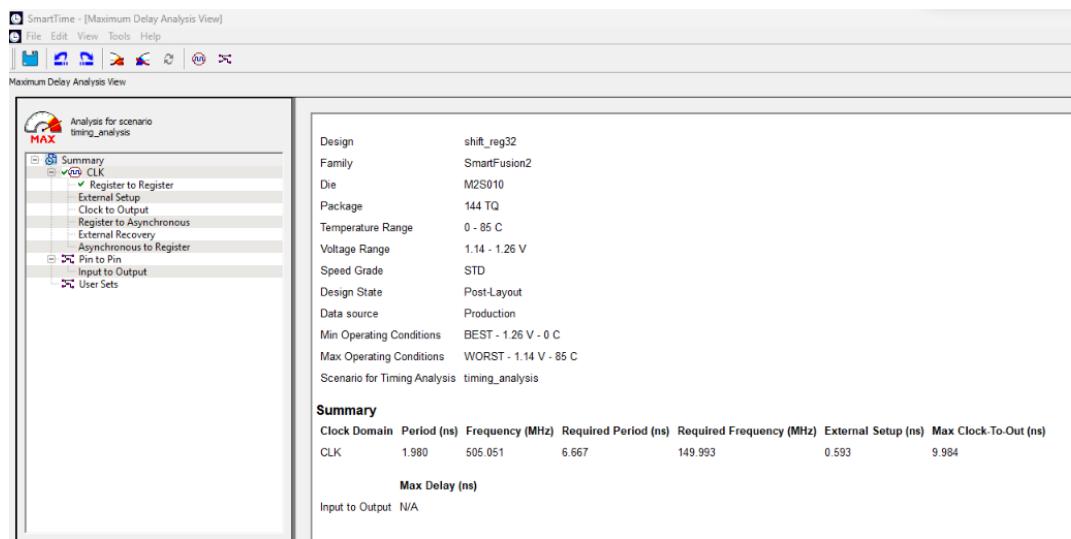


Figure P2M1 22: 32-bit Shifter Maximum Delay Analysis

shift_reg32.v | Constraint Manager | Reports | StartPage |

Project Summary

shift_reg32 reports

- Components
- synplify.log
- Synthesize
 - shift_reg32.srr
 - run_options.txt
 - shift_reg32_dsp_rpt.txt
 - shift_reg32_ram_rpt.txt
 - shift_reg32_compile_n...
 - shift_reg32_compile...
 - shift_reg32_compi...
- Place and Route
 - shift_reg32_glb_net_re...
 - shift_reg32_mindelay_r...
 - shift_reg32_layout_co...
 - shift_reg32_place_and...
 - shift_reg32_pimprt_na...
 - shift_reg32_pimprt_nu...
 - shift_reg32_bankrpt.pt
 - shift_reg32_ioff.xml
 - shift_reg32_layout...

All 0 Errors 0 Warnings 6 Info

Design: C:\Microsemi_Prj\SmartTimeTutorial\Shift32\Shift32\designer\shift_reg32\shift_reg32

Finished: Wed Oct 18 14:51:19 2023

Total CPU Time: 00:00:02 Total Elapsed Time: 00:00:02

Total Memory Usage: 445.2 Mbytes

Resource Usage

Type	Used	Total	Percentage
4LUT	0	12084	0.00
DF	32	12084	0.26
I/O Register	0	246	0.00
Logic Element	32	12084	0.26

I/O Placement

Type	Count	Percentage
Locked	0	0.00%
Placed	36	100.00%

Figure P2M1 23: 32-bit Shifter Place and Route Results

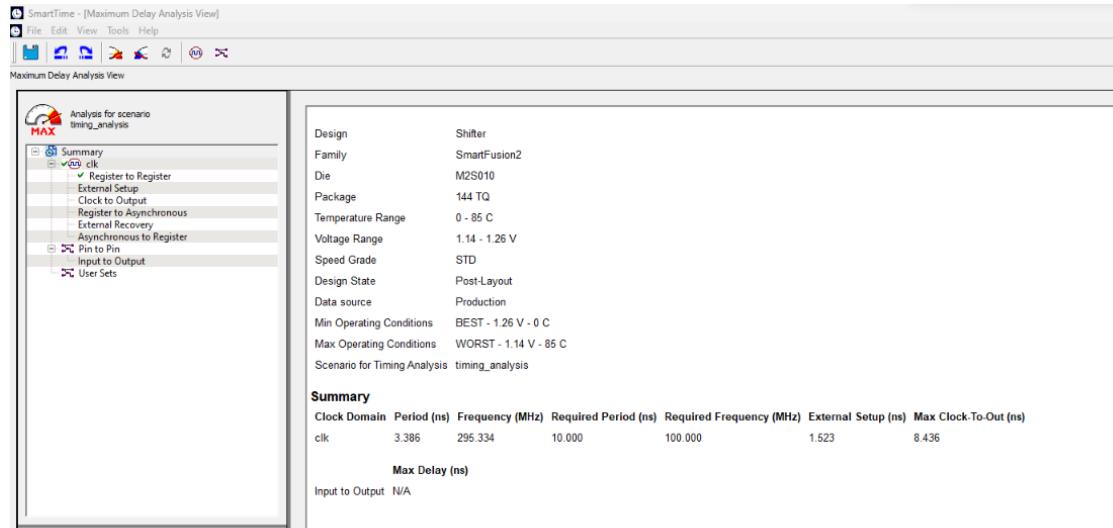


Figure P2M1 24: Shifter Maximum Delay Analysis

Shifter.v Reports StartPage

Project Summary

Shifter reports

- Components
- Synthesize
 - synplify.log
 - Shifter.srr
 - run_options.txt
 - Shifter_dsp_rpt.txt
 - Shifter_ram_rpt.txt
 - Shifter_compile_nel...
 - Shifter_compile_n...
 - Shifter_compile_n...
 - Shifter_glb_net_report...
 - Shifter_mindelay_repair...
 - Shifter_layout_combin...
 - Shifter_place_and_route...
 - Shifter_pinrpt_name.rpt
 - Shifter_pinrpt_number...
 - Shifter_bankrpt.rpt
 - Shifter_ioff.xml
 - Shifter_layout_log...
- Place and Route

Design: C:\Microsemi_Prj\SmartTimeTutorial\Shifter\Shifter\designer\Shifter\Shifter

Finished: Wed Oct 18 15:19:30 2023

Total CPU Time: 00:00:03 **Total Elapsed Time:** 00:00:04

Total Memory Usage: 445.6 Mbytes

o - o - o - o - o - o

Info: Iteration 1:

Total violating paths eligible for improvement: 0

Worst minimum delay slack:

Resource Usage

Type	Used	Total	Percentage
4LUT	8	12084	0.07
DFF	37	12084	0.31
I/O Register	0	246	0.00
Logic Element	40	12084	0.33

I/O Placement

Type	Count	Percentage
Locked	0	0.00%
Placed	20	100.00%

Figure P2M1 25: Shifter Place and Route Results

Counter16.v Reports StartPage

Project Summary

Counter16 reports

- Components
- Synthesize
 - synplify.log
 - Counter16.srr
 - run_options.txt
 - Counter16_dsp_rpt.txt
 - Counter16_ram_rpt.txt
 - Counter16_compile_n...
 - Counter16_compile_n...
 - Counter16_compile_n...
 - Counter16_compile_n...
 - Counter16_compile_n...
 - Counter16_glb_net_re...
 - Counter16_mindelay_r...
 - Counter16_layout_coo...
 - Counter16_place_and...
 - Counter16_pinrpt_na...
 - Counter16_pinrpt_nu...
 - Counter16_bankrpt.rpt
 - Counter16_ioff.xml
 - Counter16_layout_log...
- Place and Route

Design: C:\Microsemi_Prj\SmartTimeTutorial\Counter16\Counter16\designer\Counter16\Counter16

Finished: Wed Oct 18 18:16:36 2023

Total CPU Time: 00:00:02 **Total Elapsed Time:** 00:00:03

Total Memory Usage: 446.0 Mbytes

o - o - o - o - o - o

Info: Iteration 1:

Total violating paths eligible for improvement: 0

Worst minimum delay slack:

Resource Usage

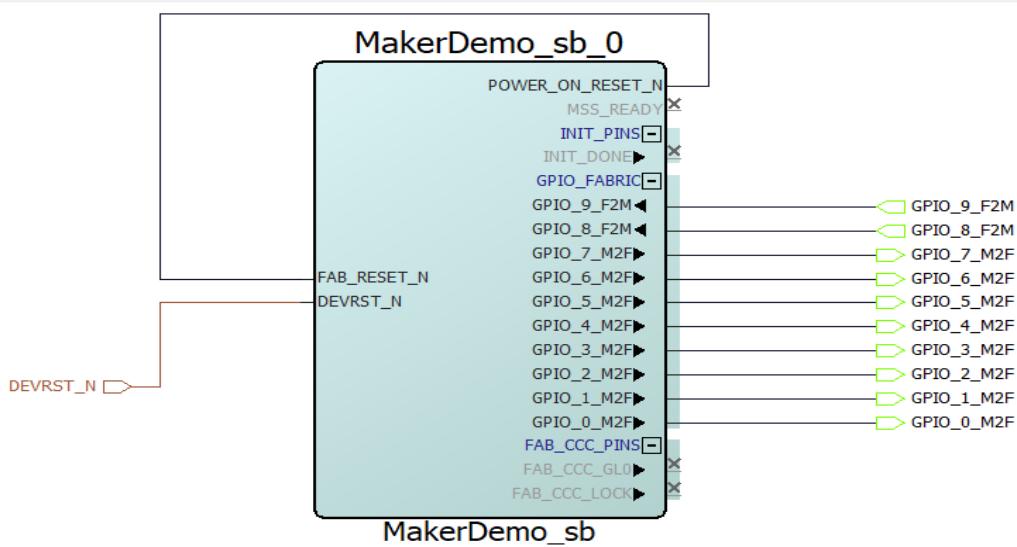
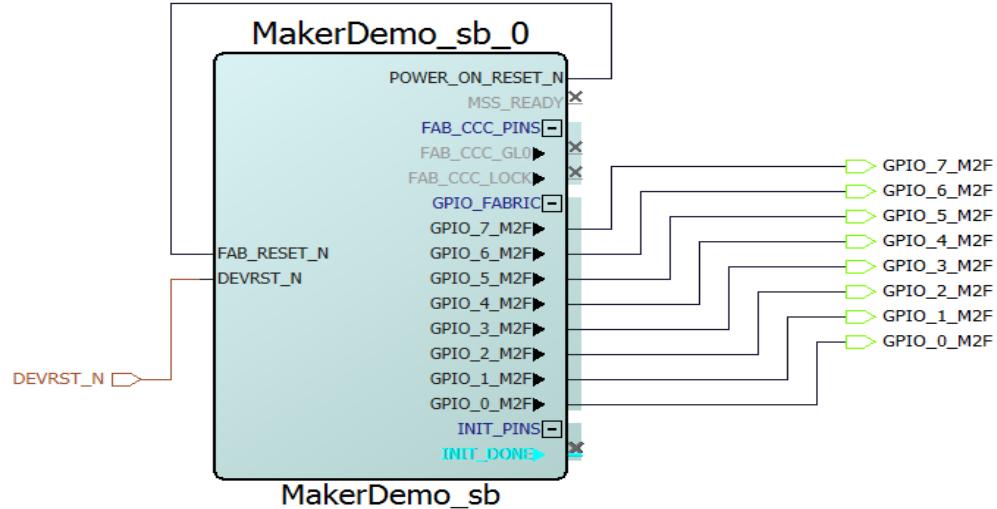
Type	Used	Total	Percentage
4LUT	18	12084	0.15
DFF	16	12084	0.13
I/O Register	0	246	0.00
Logic Element	18	12084	0.15

I/O Placement

Type	Count	Percentage
Locked	0	0.00%
Placed	19	100.00%

Figure P2M1 26: 16-bit Counter Place and Route Results

Module 2: SF2MakerDemo



```

17  # 
18  # Unlocked I/O Bank Settings
19  # The I/O Bank Settings can be locked by directly editing this file
20  # or by making changes in the I/O Attribute Editor
21 #
22
23
24  #
25  # User Locked I/O settings
26 #
27
28  set_io GPIO_0 M2F      \
29    -pinname I17          \
30    -fixed yes           \
31    -DIRECTION OUTPUT
32
33
34  set_io GPIO_1 M2F      \
35    -pinname I18          \
36    -fixed yes           \
37    -DIRECTION OUTPUT
38
39
40  set_io GPIO_2 M2F      \
41    -pinname I22          \
42    -fixed yes           \
43    -DIRECTION OUTPUT
44
45
46  set_io GPIO_3 M2F      \
47    -pinname I23          \
48    -fixed yes           \
49    -DIRECTION OUTPUT
50
51
52  set_io GPIO_4 M2F      \
53    -pinname I24          \
54    -fixed yes           \
55    -DIRECTION OUTPUT
56
57
58  set_io GPIO_5 M2F      \
59    -pinname I25          \
60    -fixed yes           \
61    -DIRECTION OUTPUT
62
63
64  set_io GPIO_6 M2F      \
65    -pinname I28          \
66    -fixed yes           \
67    -DIRECTION OUTPUT
68
69
70  set_io GPIO_7 M2F      \
71    -pinname I29          \
72    -fixed yes           \
73    -DIRECTION OUTPUT
74
75
76
77  #
78  # Dedicated Peripheral I/O Settings
79 #

```

Figure P2M2 3: Generated constraints file

```

75
76  set_io GPIO_8_F2M      \
77    -pinname I43          \
78    -fixed yes           \
79    -DIRECTION INPUT
80
81
82  set_io GPIO_9_F2M      \
83    -pinname I44          \
84    -fixed yes           \
85    -DIRECTION INPUT
86

```

Figure P2M2 4: Generated Constraint File with Bonus Implementation (applicable shown)

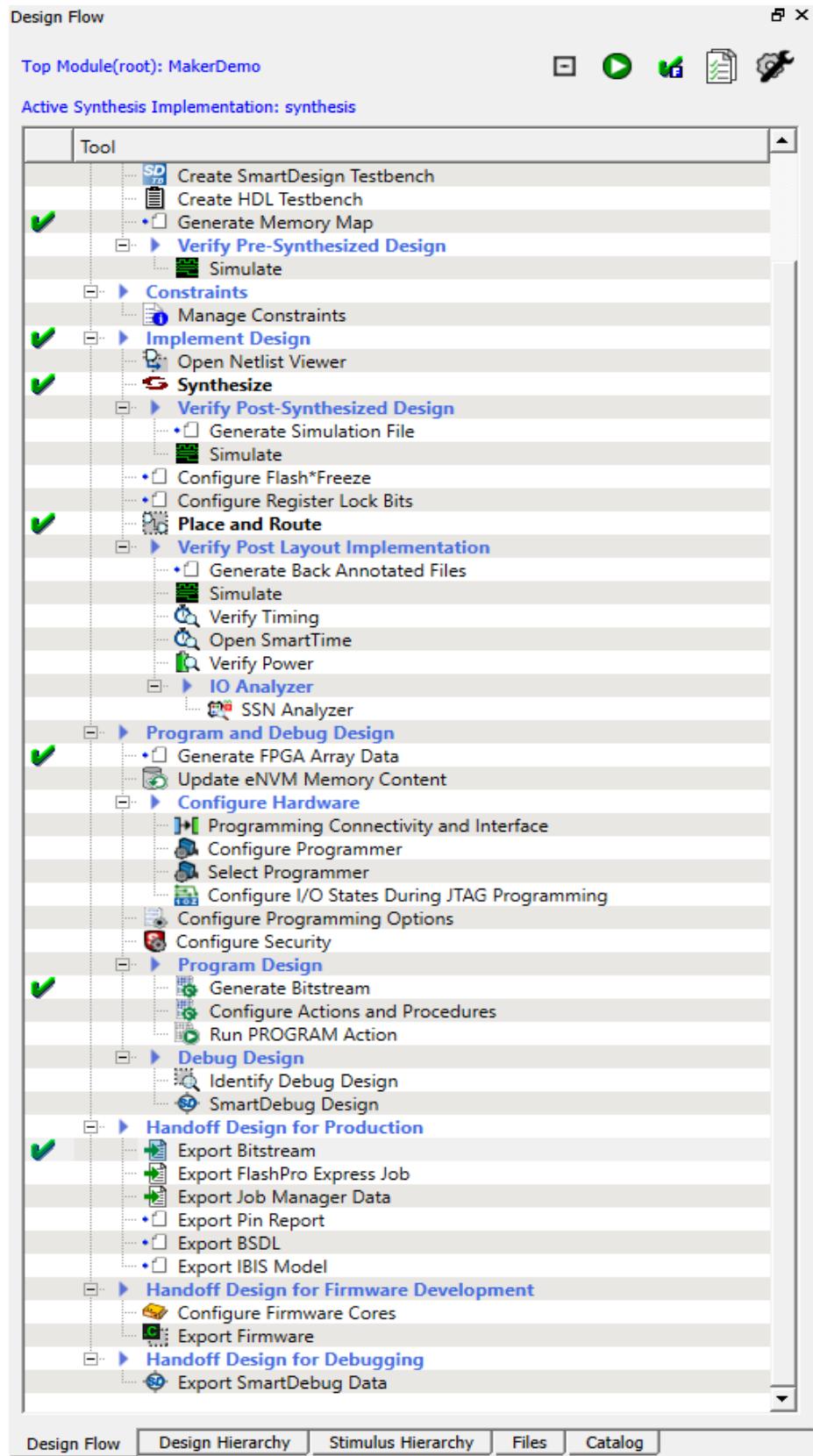


Figure P2M2 5: Fully compiled, generated, and exported project

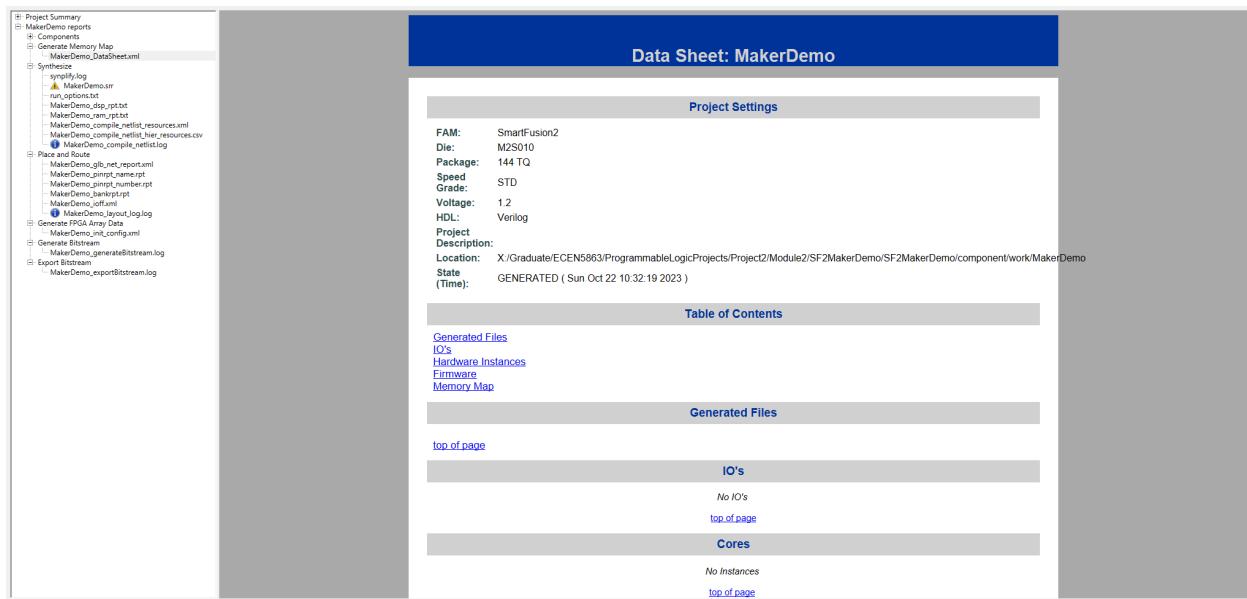


Figure P2M2 6: Report from firmware export

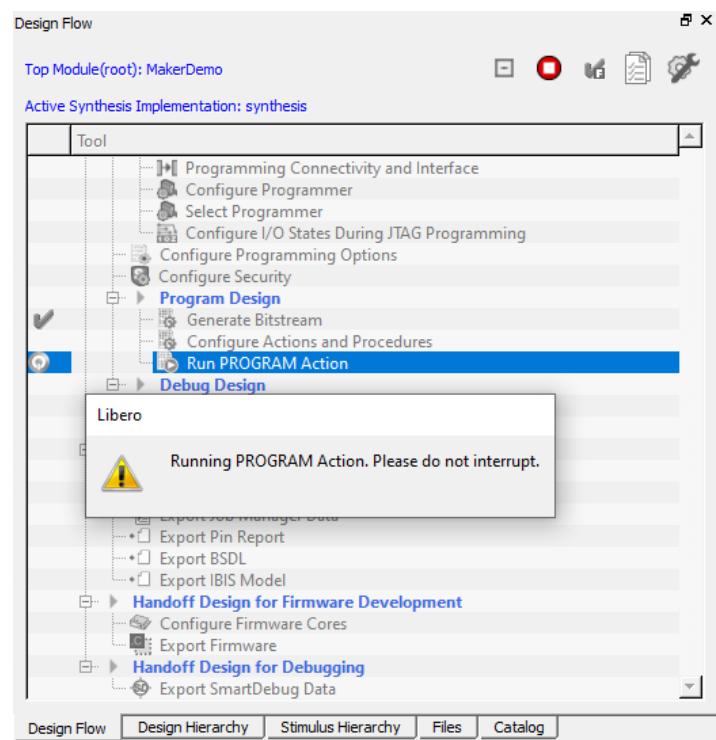


Figure P2M2 7: Programming the M2S010-MKR-KIT

All 0 Errors 1 Warning 0 Info

```

Software Version: 12.900.20.24
Programmer: 'E2001Q8RFB' : JTAG TCK / SPI SCK frequency = 1 MHz
programmer 'E2001Q8RFB' : FlashPro5
Opened 'C:\Microsemi_Prj\SF2MakerDemo\SF2MakerDemo\designer\MakerDemo\MakerDemo_fp\MakerDemo.pro'
The 'open_project' command succeeded.
Warning: Programming is already enabled for device 'M2S010'.
The 'enable_device' command succeeded.
PPD file 'C:/Microsemi_Prj/SF2MakerDemo/SF2MakerDemo/designer/MakerDemo/MakerDemo.ppd' has been loaded successfully.
DESIGN : MakerDemo; CHECKSUM : 7F3B; PDB_VERSION : 1.0
The 'set_programming_file' command succeeded.
The 'set_programming_action' command succeeded.
programmer 'E2001Q8RFB' : Scan Chain...
Programmer: 'E2001Q8RFB' : JTAG TCK / SPI SCK frequency = 1 MHz
programmer 'E2001Q8RFB' : Check Chain...
programmer 'E2001Q8RFB' : Scan and Check Chain PASSED.
programmer 'E2001Q8RFB' : device 'M2S010' : Executing action PROGRAM
Programmer: 'E2001Q8RFB' : JTAG TCK / SPI SCK frequency = 4 MHz
programmer 'E2001Q8RFB' : device 'M2S010' : Family: Igloo2
programmer 'E2001Q8RFB' : device 'M2S010' : Product: M2GL005
programmer 'E2001Q8RFB' : device 'M2S010' : Cortex-M3 Disabled (M3_ALLOWED = 0)
programmer 'E2001Q8RFB' : device 'M2S010' : EXPORT ISC_ENABLE_RESULT[32] = 007d6944
programmer 'E2001Q8RFB' : device 'M2S010' : EXPORT CRCERR[1] = 0
programmer 'E2001Q8RFB' : device 'M2S010' : EXPORT EDCERR[1] = 0
programmer 'E2001Q8RFB' : device 'M2S010' : Programming FPGA Array...
programmer 'E2001Q8RFB' : device 'M2S010' : EXPORT Fabric component digest[256] = 259b73128e0adf6d52c5bla6cc411a1256
programmer 'E2001Q8RFB' : device 'M2S010' : =====
programmer 'E2001Q8RFB' : device 'M2S010' : EXPORT DSN[128] = fed6206f61a59759db5d17a775e58fc7

```

Figure P2M2 8: Program results

```

Software Version: 12.900.20.24
Opened 'C:\Microsemi_Prj\SF2MakerDemo\SF2MakerDemo\designer\MakerDemo\MakerDemo_fp\MakerDemo.pro'
PDB file 'C:\Microsemi_Prj\SF2MakerDemo\SF2MakerDemo\designer\MakerDemo\MakerDemo.pdb' has been loaded successfully.
DESIGN : MakerDemo; CHECKSUM : 7F3B; PDB_VERSION : 1.9
Successfully exported STAPL file: 'C:\Microsemi_Prj\SF2MakerDemo\SF2MakerDemo\designer\MakerDemo\export\MakerDemo.stp'
Fabric component bitstream digest: 259b73128e0adf6d52c5bla6cc411a12566489b3d74b8071fe88d17f5dc59b58
Finished: Sun Oct 22 17:01:38 2023 (Elapsed time 00:00:02)

Successfully exported PPD file for currently secured device: 'C:\Microsemi_Prj\SF2MakerDemo\SF2MakerDemo\designer\Make
Fabric component bitstream digest: 259b73128e0adf6d52c5bla6cc411a12566489b3d74b8071fe88d17f5dc59b58
Finished: Sun Oct 22 17:01:43 2023 (Elapsed time 00:00:05)

Successfully exported DirectC file: 'C:\Microsemi_Prj\SF2MakerDemo\SF2MakerDemo\designer\MakerDemo\export\MakerDemo.da
Fabric component bitstream digest: 259b73128e0adf6d52c5bla6cc411a12566489b3d74b8071fe88d17f5dc59b58
Finished: Sun Oct 22 17:01:46 2023 (Elapsed time 00:00:03)

Project saved.
Project closed.

```

Figure P2M2 9: Bitstream generation STAPL file success

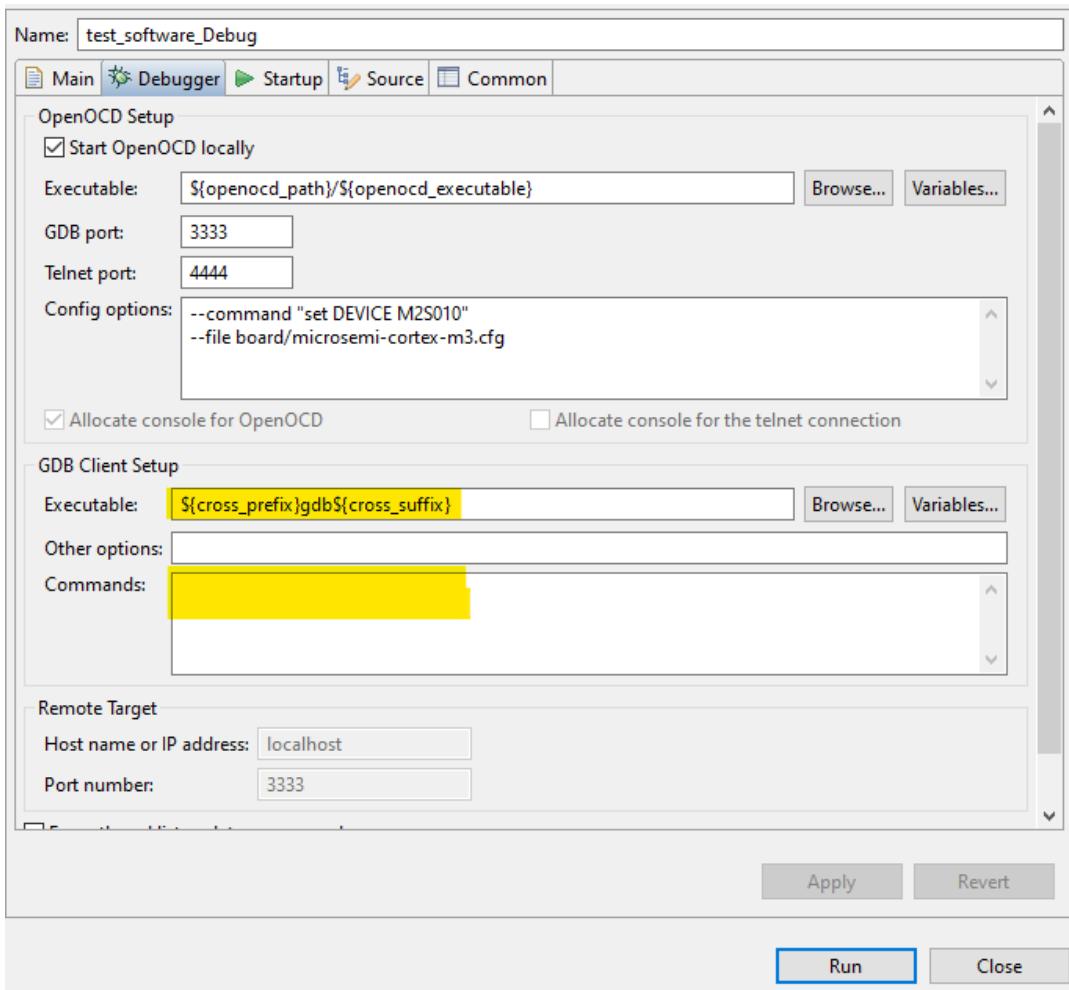


Figure P2M2 10: The configuration that was necessary in order to get it working

	Port Name	Direction	I/O Standard	Pin Number	Locked	Macro Cell	Bank Name
1	DEVRST_N	INPUT	--	72	<input checked="" type="checkbox"/>	SYSRESET	--
2	GPIO_0_M2F	OUTPUT	LVCMS25	117	<input checked="" type="checkbox"/>	OUTBUF	Bank0
3	GPIO_1_M2F	OUTPUT	LVCMS25	118	<input checked="" type="checkbox"/>	OUTBUF	Bank0
4	GPIO_2_M2F	OUTPUT	LVCMS25	122	<input checked="" type="checkbox"/>	OUTBUF	Bank0
5	GPIO_3_M2F	OUTPUT	LVCMS25	123	<input checked="" type="checkbox"/>	OUTBUF	Bank0
6	GPIO_4_M2F	OUTPUT	LVCMS25	124	<input checked="" type="checkbox"/>	OUTBUF	Bank0
7	GPIO_5_M2F	OUTPUT	LVCMS25	125	<input checked="" type="checkbox"/>	OUTBUF	Bank0
8	GPIO_6_M2F	OUTPUT	LVCMS25	128	<input checked="" type="checkbox"/>	OUTBUF	Bank0
9	GPIO_7_M2F	OUTPUT	LVCMS25	129	<input checked="" type="checkbox"/>	OUTBUF	Bank0
10	GPIO_8_F2M	INPUT	LVCMS25	143	<input checked="" type="checkbox"/>	INBUF	Bank0
11	GPIO_9_F2M	INPUT	LVCMS25	144	<input checked="" type="checkbox"/>	INBUF	Bank0

Figure P2M2 11: Hardware changes required to meet BONUS step for Module 2

Code snippet for Module 2's Bonus.

Declarations:

```
```c
/* IRQ handlers which update the switch flag */
unsigned int sw_pressed;
#define handle_sw_irq(x) do { \
 x = x ? 0 : 1; \
} while (0)
void GPIO8_IRQHandler(void) { handle_sw_irq(sw_pressed); }
void GPIO9_IRQHandler(void) { handle_sw_irq(sw_pressed); }
````
```

Definitions:

```
```c
int main(void)
{
/*...
 * Initialize input switches */
 MSS_GPIO_config(MSS_GPIO_8, MSS_GPIO_INPUT_MODE |
MSS_GPIO_IRQ_EDGE_NEGATIVE);
 MSS_GPIO_config(MSS_GPIO_9, MSS_GPIO_INPUT_MODE |
MSS_GPIO_IRQ_EDGE_POSITIVE);

/* Enable the IRQs for the switch inputs */
 MSS_GPIO_enable_irq(MSS_GPIO_8);
 MSS_GPIO_enable_irq(MSS_GPIO_9);

/*...
 * Sequence-blink */
 for (;;) {
 for (i = 0; i < 8; i++) {
 /* If the switch is pressed, the LEDs will change direction by
 * inverting the value to be assigned */
 int val = sw_pressed ? 8-i : i;
 current_val = (MSS_GPIO_get_outputs() & (1 << LED[val])) ? 1 : 0;
 MSS_GPIO_set_output(LED[val], current_val ^ 1);
 delay();
 }
 }
}
````
```

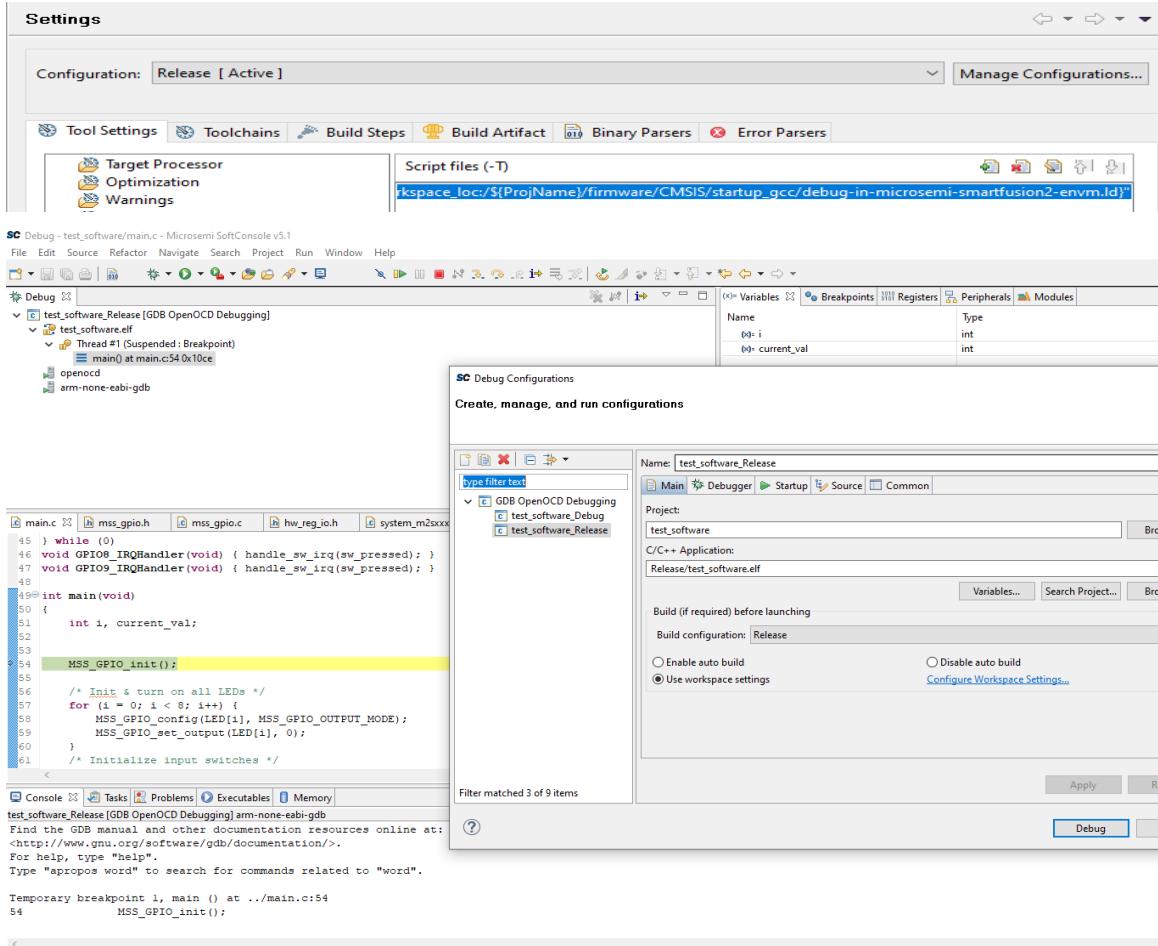


Figure P2M2 12: The Release configuration was setup to program into the SmartFusion2's eNVM

| | |
|---|---|
| Design | MakerDemo |
| Family | SmartFusion2 |
| Die | M2S010 |
| Package | 144 TQ |
| Temperature Range | 0 - 85 C |
| Voltage Range | 1.14 - 1.26 V |
| Speed Grade | STD |
| Design State | Post-Layout |
| Data source | Production |
| Min Operating Conditions | BEST - 1.26 V - 0 C |
| Max Operating Conditions | WORST - 1.14 V - 85 C |
| Scenario for Timing Analysis | timing_analysis |
| Summary | |
| Clock Domain | Period (ns) Frequency (MHz) Required Period (ns) Required Frequency (MHz) External Setup (ns) Max Clock-To-Out (ns) |
| MakerDemo_sb_0/CCC_0/CCC_INST/INST_CCC_IP:GL0 | 5.253 190.367 N/A N/A 10.494 |
| MakerDemo_sb_0/FABOSC_0/I_RCOSC_25_50MHZ:CLKOUT | N/A N/A N/A N/A N/A |
| Max Delay (ns) | |
| Input to Output | N/A |

Figure P2M2 13: fMAX results for SmartFusion2

| Design | MakerDemo | | | | | |
|---|-----------------------|-----------------|----------------------|--------------------------|---------------------|-----------------------|
| Family | SmartFusion2 | | | | | |
| Die | M2S010 | | | | | |
| Package | 144 TQ | | | | | |
| Temperature Range | 0 - 85 C | | | | | |
| Voltage Range | 1.14 - 1.26 V | | | | | |
| Speed Grade | STD | | | | | |
| Design State | Post-Layout | | | | | |
| Data source | Production | | | | | |
| Min Operating Conditions | BEST - 1.26 V - 0 C | | | | | |
| Max Operating Conditions | WORST - 1.14 V - 85 C | | | | | |
| Scenario for Timing Analysis | timing_analysis | | | | | |
| Summary | | | | | | |
| Clock Domain | Period (ns) | Frequency (MHz) | Required Period (ns) | Required Frequency (MHz) | External Setup (ns) | Max Clock-To-Out (ns) |
| MakerDemo_sb_0/CCC_0/CCC_INST/INST_CCC_IP:GL0 | 5.253 | 190.367 | N/A | N/A | N/A | 10.494 |
| MakerDemo_sb_0/FABOSC_0/RCOSC_25_50MHZ:CLKOUT | N/A | N/A | N/A | N/A | N/A | N/A |
| Max Delay (ns) | | | | | | |
| Input to Output | N/A | | | | | |

Figure P2M2 14: SmartFusion2 Compile Report

| | Source Pin | Sink Pin | Delay (ns) | Slack (ns) | Arrival (ns) | Required (ns) | Clock to Out (ns) |
|---|-----------------------|------------|------------|------------|--------------|---------------|-------------------|
| 1 | MakerDemo_sb_0/Mak... | GPIO_0_M2F | 8.490 | | 10.494 | | 10.494 |
| 2 | MakerDemo_sb_0/Mak... | GPIO_1_M2F | 8.376 | | 10.380 | | 10.380 |
| 3 | MakerDemo_sb_0/Mak... | GPIO_6_M2F | 8.208 | | 10.212 | | 10.212 |
| 4 | MakerDemo_sb_0/Mak... | GPIO_7_M2F | 8.172 | | 10.176 | | 10.176 |
| 5 | MakerDemo_sb_0/Mak... | GPIO_4_M2F | 8.159 | | 10.163 | | 10.163 |
| 6 | MakerDemo_sb_0/Mak... | GPIO_5_M2F | 8.110 | | 10.114 | | 10.114 |
| 7 | MakerDemo_sb_0/Mak... | GPIO_2_M2F | 8.096 | | 10.100 | | 10.100 |
| 8 | MakerDemo_sb_0/Mak... | GPIO_3_M2F | 8.017 | | 10.021 | | 10.021 |

Figure P2M2 15: Specific input arrive time and clock-to-out time depends on the GPIO pin

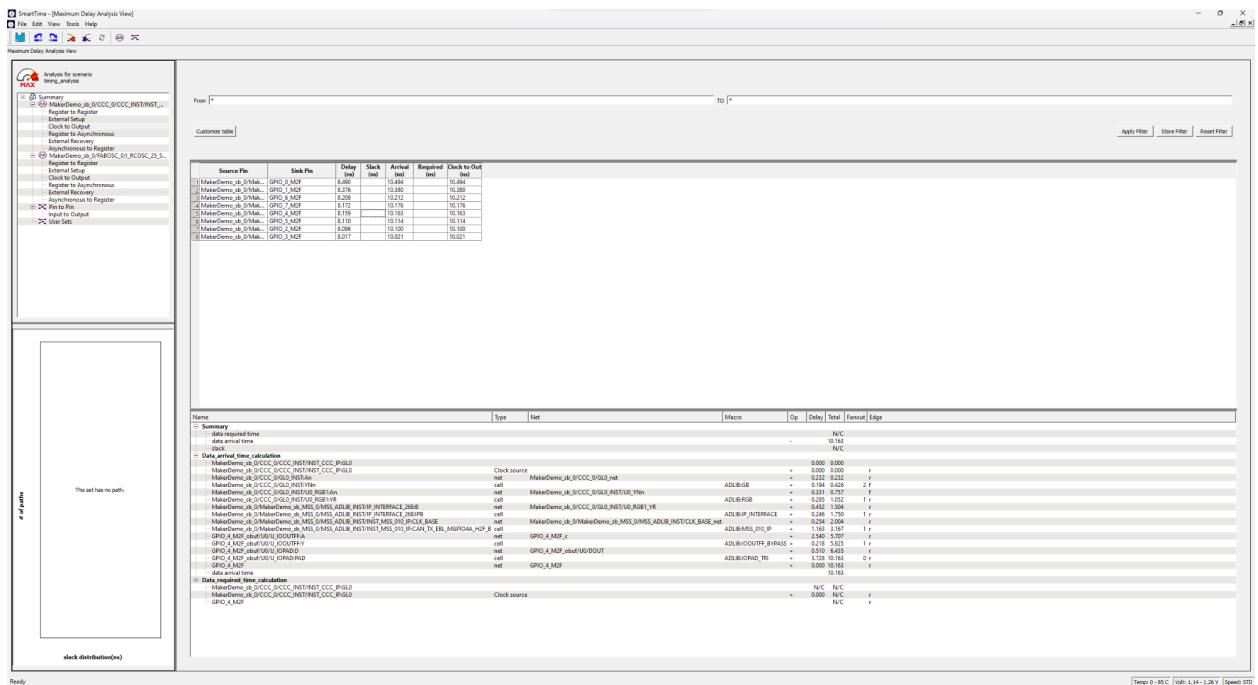


Figure P2M2 16: Pin Report

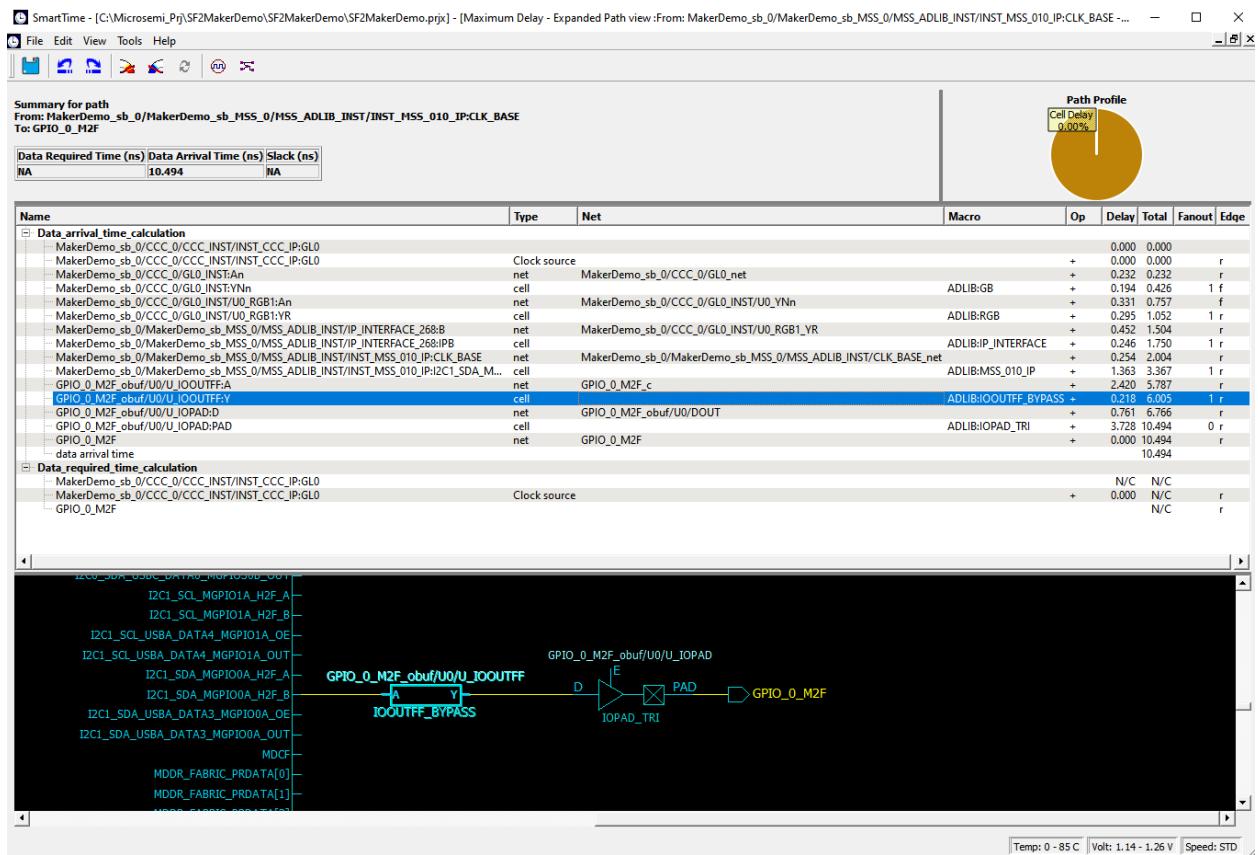
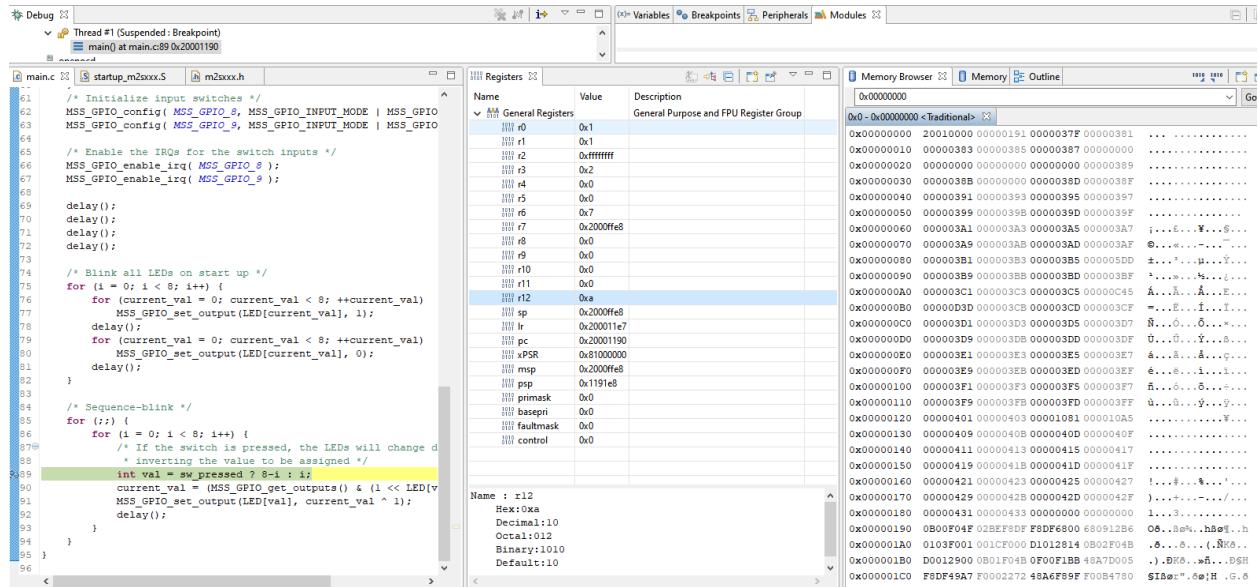


Figure P2M2 17: SmartTime Pin Report

Resource Usage

| Type | Used | Total | Percentage |
|---------------------------|------|-------|------------|
| 4LUT | 0 | 12084 | 0.00 |
| DFF | 3 | 12084 | 0.02 |
| I/O Register | 0 | 252 | 0.00 |
| User I/O | 8 | 84 | 9.52 |
| -- Single-ended I/O | 8 | 84 | 9.52 |
| -- Differential I/O Pairs | 0 | 37 | 0.00 |
| RAM64x18 | 0 | 22 | 0.00 |
| RAM1K18 | 0 | 21 | 0.00 |
| MACC | 0 | 22 | 0.00 |
| Chip Globals | 1 | 8 | 12.50 |
| CCC | 1 | 2 | 50.00 |
| RCOSC_25_50MHZ | 1 | 1 | 100.00 |
| RCOSC_1MHZ | 0 | 1 | 0.00 |
| XTLOSC | 0 | 1 | 0.00 |
| MSS | 1 | 1 | 100.00 |

Figure P2M2 18: The % utilization report



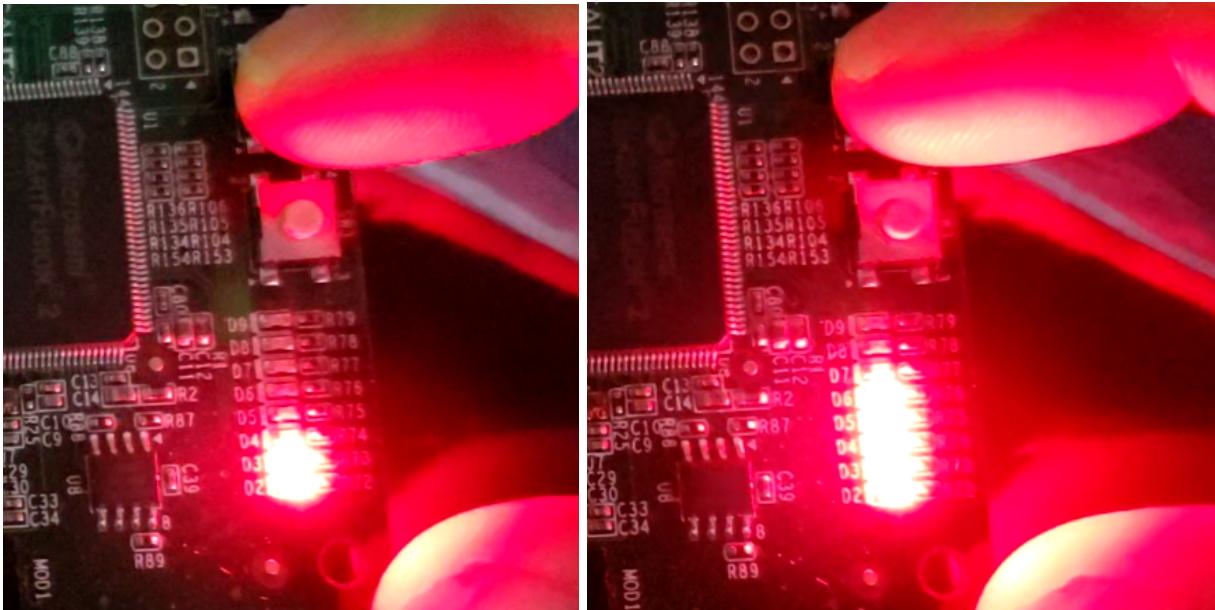
The screenshot shows a debugger interface with several windows. On the left is the code editor for 'main.c' with some code visible. In the center is the 'Registers' window showing various general-purpose registers (r0 to r12, sp, lr, pc, ssp, msp, psp) with their values and descriptions. To the right is the 'Memory Browser' window, which displays memory starting at address 0x00000000. The browser shows multiple memory blocks, each with a hex dump, ASCII representation, and a column for 'Description'. One specific row is highlighted in yellow, corresponding to the memory location shown in the registers. The memory browser also includes a search bar and navigation controls.

```

61 /* Initialize input switches */
62 MSS_GPIO_config(MSS_GPIO_8, MSS_GPIO_INPUT_MODE | MSS_GPIO
63 MSS_GPIO_config(MSS_GPIO_9, MSS_GPIO_INPUT_MODE | MSS_GPIO
64
65 /* Enable the IRQs for the switch inputs */
66 MSS_GPIO_enable_irq(MSS_GPIO_8);
67 MSS_GPIO_enable_irq(MSS_GPIO_9);
68
69 delay();
70 delay();
71 delay();
72 delay();
73
74 /* Blink all LEDs on start up */
75 for (i = 0; i < 8; i++) {
76     for (current_val = 0; current_val < 8; ++current_val)
77         MSS_GPIO_set_output(LED[current_val], 1);
78     delay();
79     for (current_val = 0; current_val < 8; ++current_val)
80         MSS_GPIO_set_output(LED[current_val], 0);
81     delay();
82 }
83
84 /* Sequence-blink */
85 for (;;) {
86     for (i = 0; i < 8; i++) {
87         /* If the switch is pressed, the LEDs will change direction
88         * inverting the value to be assigned */
89         int val = sw_pressed ? 8-i : i;
90         current_val = (MSS_GPIO_get_outputs() & (1 << LED[v
91         MSS_GPIO_set_output(LED[val], current_val ^ 1);
92         delay();
93     }
94 }
95

```

Figure P2M2 19: The memory results for that range are provided in the rightmost column though



Figures P2M2 20 & 21: Switch LED direction change results

Module 3: Load-Testing FPGAs with Counters

```

Design: C:\Microsemi_Prj\Project2\P2M3\P2M3\designer\NBitCounter\NBitCounter
Finished: Sun Oct 29 14:50:51 2023
Total CPU Time: 00:00:02           Total Elapsed Time: 00:00:03
Total Memory Usage: 445.9 Mbytes
          o - o - o - o - o - o

Resource Usage
+-----+-----+-----+
| Type      | Used | Total | Percentage |
+-----+-----+-----+
4LUT	22	12084	0.18
DFF	16	12084	0.13
I/O Register	0	246	0.00
Logic Element	22	12084	0.18
+-----+-----+-----+

I/O Placement
+-----+-----+
| Type    | Count | Percentage |
+-----+-----+
| Locked  | 0     | 0.00%   |
| Placed  | 20    | 100.00% |
+-----+-----+

```

Figure P2M3 1: Logic Utilization Report

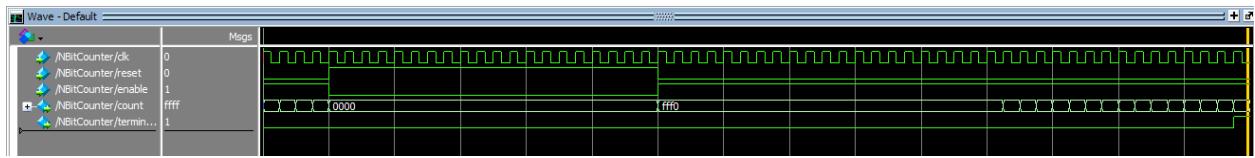


Figure P2M3 2: 16-bit counter in Verilog RTL simulation.

| Resource Usage | | | |
|----------------|------|-------|------------|
| Type | Used | Total | Percentage |
| 4LUT | 2195 | 12084 | 18.16 |
| DFF | 1600 | 12084 | 13.24 |
| I/O Register | 0 | 246 | 0.00 |
| Logic Element | 2195 | 12084 | 18.16 |

Figure P2M3 3: Microsemi SmartFusion2 10 counters logic utilization

| Resource Usage | | | |
|----------------|------|-------|------------|
| Type | Used | Total | Percentage |
| 4LUT | 7255 | 12084 | 60.04 |
| DFF | 5280 | 12084 | 43.69 |
| I/O Register | 0 | 246 | 0.00 |
| Logic Element | 7255 | 12084 | 60.04 |

Figure P2M3 4: Microsemi SmartFusion2 100 counters logic utilization

| Flow Summary | |
|------------------------------------|---|
| | <<Filter>> |
| Flow Status | Successful - Sun Oct 29 15:40:11 2023 |
| Quartus Prime Version | 18.1.0 Build 625 09/12/2018 SJ Lite Edition |
| Revision Name | P2M3 |
| Top-level Entity Name | NBitCounterChain |
| Family | MAX 10 |
| Device | 10M50DAF484C6GES |
| Timing Models | Preliminary |
| Total logic elements | 2,098 / 49,760 (4 %) |
| Total registers | 1600 |
| Total pins | 18 / 360 (5 %) |
| Total virtual pins | 0 |
| Total memory bits | 0 / 1,677,312 (0 %) |
| Embedded Multiplier 9-bit elements | 0 / 288 (0 %) |
| Total PLLs | 0 / 4 (0 %) |
| UFM blocks | 0 / 1 (0 %) |
| ADC blocks | 0 / 2 (0 %) |

Figure P2M3 5: Altera MAX10 DE10-lite 100 counters logic utilization.

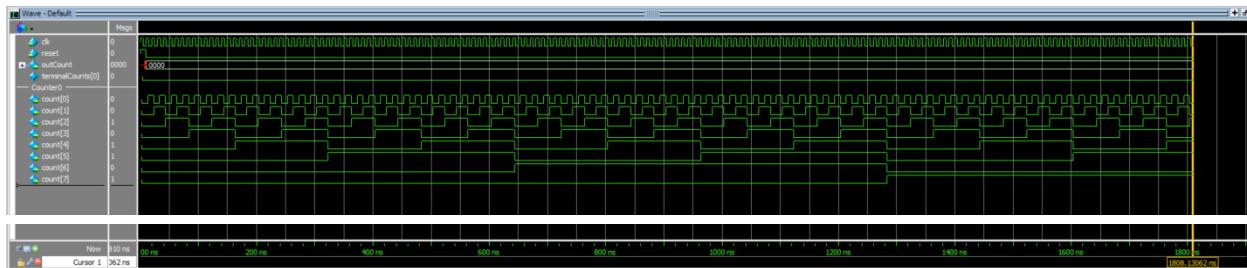


Figure P2M3 6: RTL simulation for the SmartFusion2 Maker Kit.

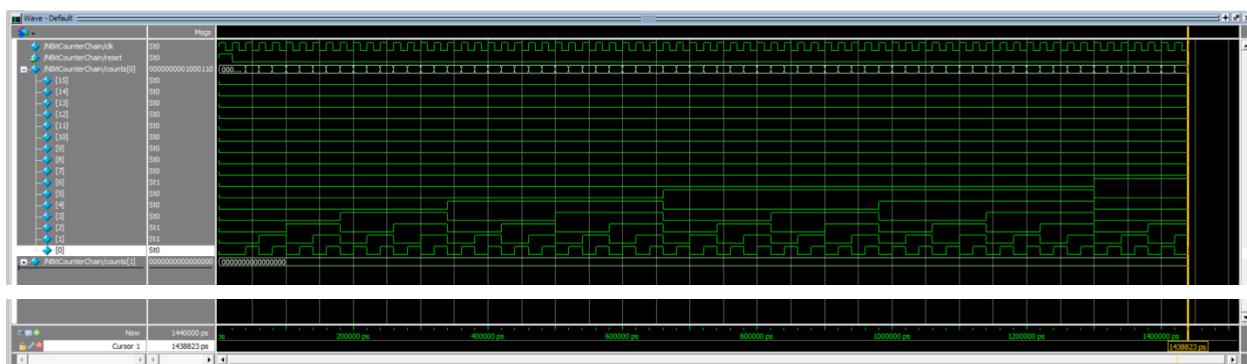


Figure P2M3 7: RTL simulation for the DE10-Lite board.

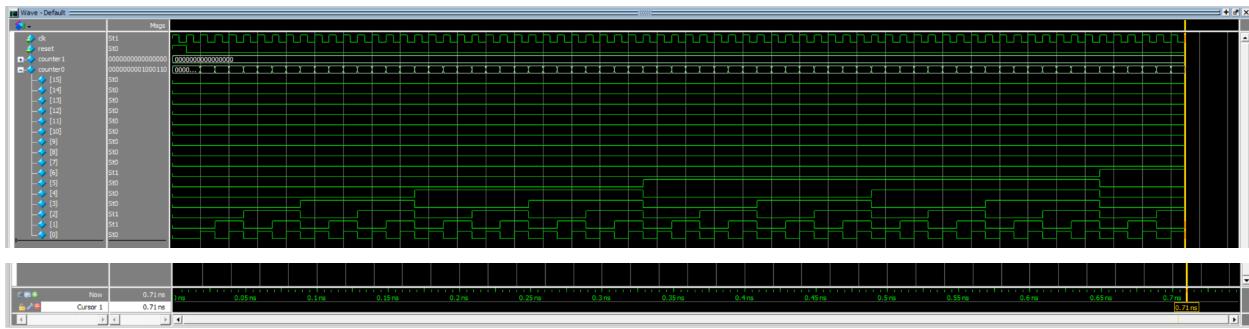


Figure P2M3 8: RTL simulation for the DE1-SoC board

Clock Details Summary

Clock Domain Period (ns) Frequency (MHz)

| | | |
|-----|-------|---------|
| clk | 4.079 | 245.158 |
|-----|-------|---------|

Figure P2M3 9: Fmax for SmartFusion2 with the full maximum of counters.

| Slow 1200mV 85C Model Fmax Summary | | | |
|------------------------------------|------------|-----------------|------------|
| <<Filter>> | | | |
| | Fmax | Restricted Fmax | Clock Name |
| 1 | 125.75 MHz | 125.75 MHz | clk |

Figure P2M3 10: Fmax for DE10-Lite with the full maximum of counters.

| Slow 1100mV 85C Model Fmax Summary | | | | |
|------------------------------------|-----------|-----------------|------------|------|
| <<Filter>> | | | | |
| | Fmax | Restricted Fmax | Clock Name | Note |
| 1 | 213.9 MHz | 213.9 MHz | clk | |

Figure P2M3 11: Fmax for DE1-SoC with the full maximum of counters.