FACULTY OF ENGINEERING TECHNOLOGY
GROUP T
ELECTRONICS AND ICT
ANDREAS VESALIUSSTRAAT 13 B-3000 LEUVEN

Final Project

Complex Digital Design

Belgin Ayvat
Wiktoria Radecka

06.05.2024

Contents

1 Summary

The main objective of this project is to boost the performance of a multi-precision adder by implementing a faster adder circuit. We have opted for the carry select adder circuit as the main building block. In this way, we can significantly improve the operation latency and hence, adhere to the latency requirement of at most 18 clock cycles.

The maximum `ADDER_WIDTH` that the design can tolerate is 64 bits[1] and in that case the worst negative slack is 1.27 s. This leads to the conclusion that the maximum number of cycles required to complete 512-bit addition is $\frac{\texttt{OPERAND\_WIDTH}}{\texttt{ADDER\_WIDTH}} + 2 = \frac{512}{64} + = 8 + 2 = 10$ clock cycles.

Furthermore, we have implemented the subtraction operation using the fundamental carry select adder block. Since subtraction is essentially an addition of the augend and the inverted addend, the implemented operation required merely inverting each bit of the later one and adding 1 to it.

2 Technical Description

For this project, we have chosen a uniform-sized carry select adder (CSelA). The idea behind this structure is to pre-compute the result of Full Adders (FA) without waiting for the carry in bit. It works by instantiating two FAs, one that has $C_{\text{in}} = 0$ and one for $C_{\text{in}} = 1$. Once the the actual carry in bit is available, CSelA selects the correct output with multiplexers.

A high-level block diagram of the carry select adder that is implemented can be seen in Figure 1. The finite state diagram of `uart_top.v` can be seen in Figure 2.

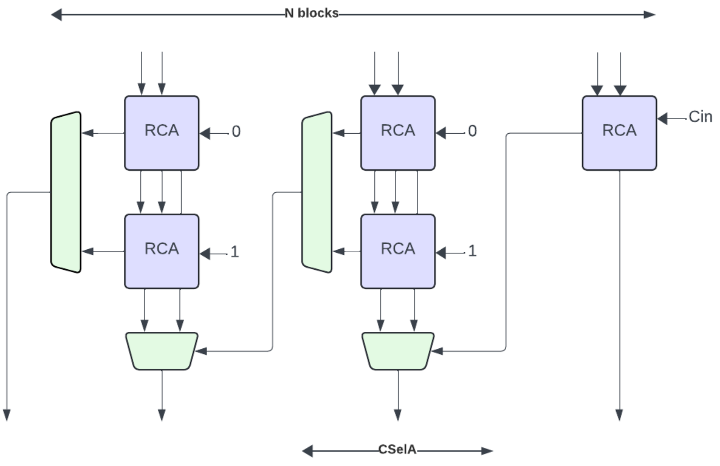


Figure 1: High-level overview of the adder

The overall structure we implemented entails $N-1$ blocks of CSelA and 1 ripple carry adder (RCA) at the beginning. Inside each CSelA, we have two blocks of 8 RCAs, one per each carry in bit possibility. We notice that the number of RCAs inside one CSelA should be a divisor of the number of bytes we plan to add.

The subtraction was implemented using the structure with a multiplexer. We decided to use a register `rOperation` to indicate whether or not we should flip the addend bits. This register is going to input a bit into 2-to-1 multiplexer to choose which augend (inverted or not) to use for the operation.

[1] however, at the last minute suddenly 128 bits and 256 bits

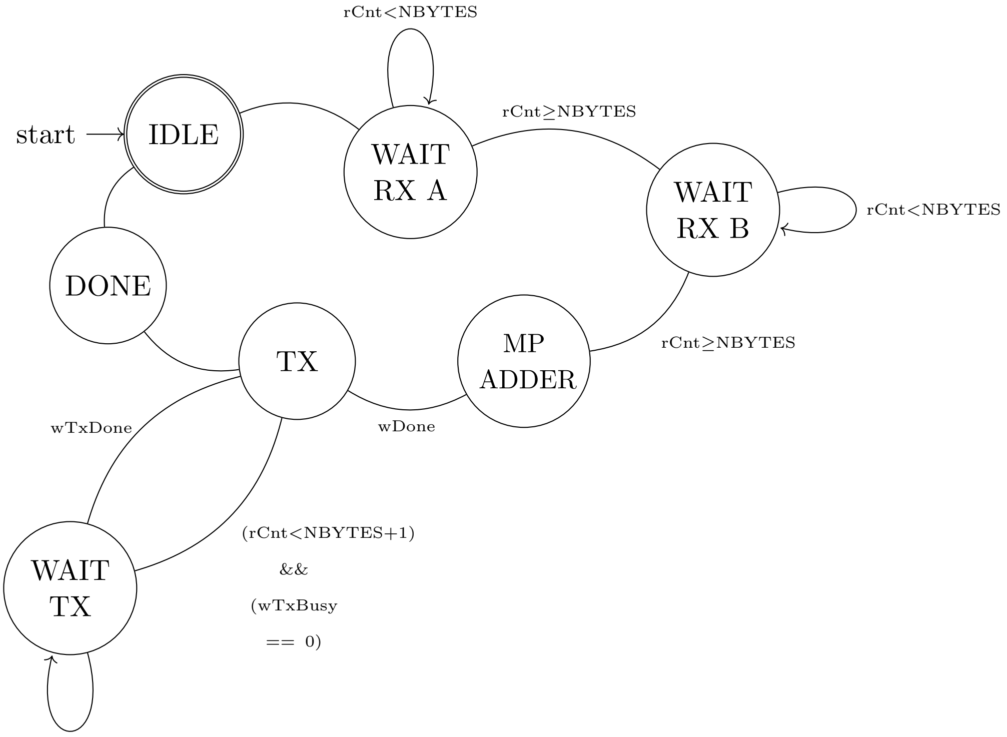


Figure 2: State Transition diagram of `top_module`

3 Performance Evaluation

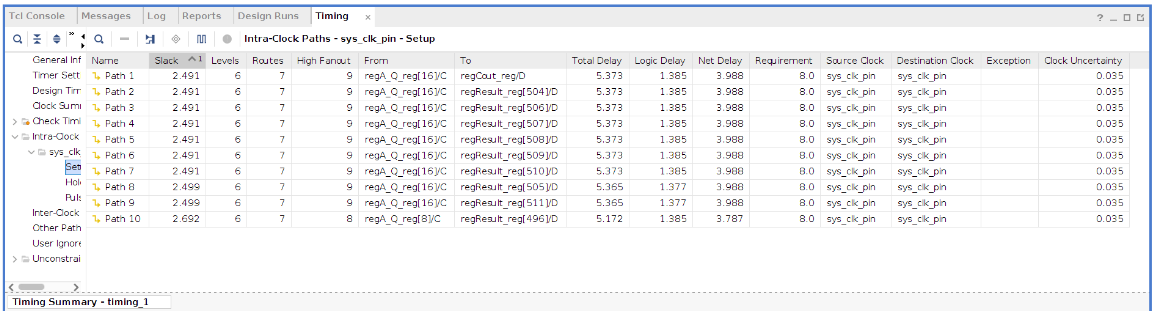
The worst-case delays of our design for different adder widths are displayed in Figure 3. For all cases, the longest path occurs between bit 16 of operand A and the carry at the output. As the adder width increases, the delay increases. This occurs because with increasing adder width, the number of CSelA blocks in the adder increases hence the carry at the output needs to wait longer at the end for the earlier carry bits to arrive.

When an adder width of 128 is used, the design does not meet the timing requirements and provides a worse negative slack of -1.205 nanoseconds.

The design can be improved against this negative slack problem by modifying the adder architecture to contain carry lookahead logic inside the individual CSelA blocks with M bit carry lookahead adders in cascade. This would allow the carry computation to be faster than the multiplexers propagating.

Another solution around the negative slack problem is to use a larger block size for the individual CSelA blocks inside the adder with the RCA's cascading. As can be seen from the lecture slides in Figure 4 the logic gate delay provided with a block size of M = 16 gives a lower amount of delay at the logic gates.

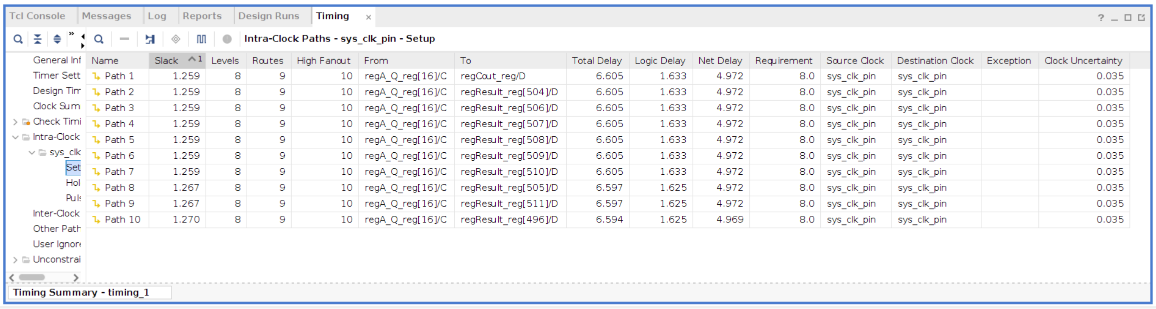
As can be seen in the utilization reports for different adder widths in Figure 5, the area increases when the adder becomes larger. This is expected because the number of carry select adder blocks instantiated grows with the adder width.



Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack ^1	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	2.491	6	7	9	regA_Q_reg[16]/C	regCout_reg/D	5.373	1.385	3.988	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 2	2.491	6	7	9	regA_Q_reg[16]/C	regResult_reg[504]/D	5.373	1.385	3.988	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 3	2.491	6	7	9	regA_Q_reg[16]/C	regResult_reg[506]/D	5.373	1.385	3.988	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 4	2.491	6	7	9	regA_Q_reg[16]/C	regResult_reg[507]/D	5.373	1.385	3.988	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 5	2.491	6	7	9	regA_Q_reg[16]/C	regResult_reg[508]/D	5.373	1.385	3.988	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 6	2.491	6	7	9	regA_Q_reg[16]/C	regResult_reg[509]/D	5.373	1.385	3.988	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 7	2.491	6	7	9	regA_Q_reg[16]/C	regResult_reg[510]/D	5.373	1.385	3.988	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 8	2.499	6	7	9	regA_Q_reg[16]/C	regResult_reg[505]/D	5.365	1.377	3.988	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 9	2.499	6	7	9	regA_Q_reg[16]/C	regResult_reg[511]/D	5.365	1.377	3.988	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 10	2.692	6	7	8	regA_Q_reg[8]/C	regResult_reg[496]/D	5.172	1.385	3.787	8.0	sys_clk_pin	sys_clk_pin		0.035

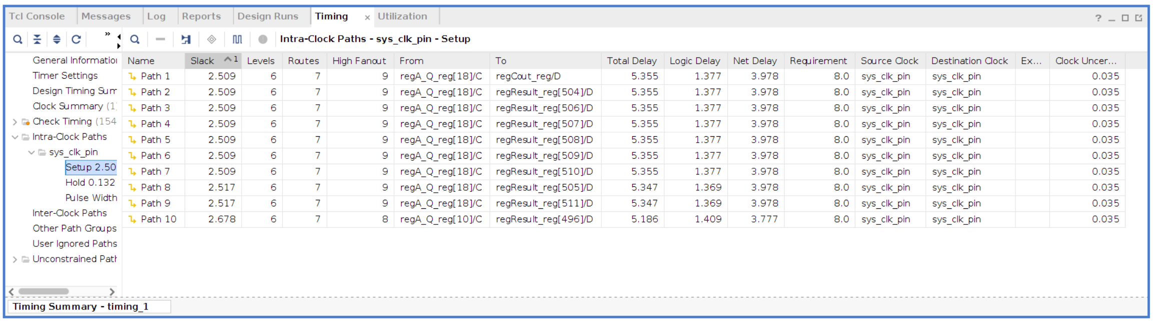
(a) Worst Case Delay for `ADDER_WIDTH` = 32



Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack ^1	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	1.259	8	9	10	regA_Q_reg[16]/C	regCout_reg/D	6.605	1.633	4.972	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 2	1.259	8	9	10	regA_Q_reg[16]/C	regResult_reg[504]/D	6.605	1.633	4.972	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 3	1.259	8	9	10	regA_Q_reg[16]/C	regResult_reg[506]/D	6.605	1.633	4.972	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 4	1.259	8	9	10	regA_Q_reg[16]/C	regResult_reg[507]/D	6.605	1.633	4.972	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 5	1.259	8	9	10	regA_Q_reg[16]/C	regResult_reg[508]/D	6.605	1.633	4.972	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 6	1.259	8	9	10	regA_Q_reg[16]/C	regResult_reg[509]/D	6.605	1.633	4.972	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 7	1.259	8	9	10	regA_Q_reg[16]/C	regResult_reg[510]/D	6.605	1.633	4.972	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 8	1.267	8	9	10	regA_Q_reg[16]/C	regResult_reg[505]/D	6.597	1.625	4.972	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 9	1.267	8	9	10	regA_Q_reg[16]/C	regResult_reg[511]/D	6.597	1.625	4.972	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 10	1.270	8	9	10	regA_Q_reg[16]/C	regResult_reg[496]/D	6.594	1.625	4.969	8.0	sys_clk_pin	sys_clk_pin		0.035

(b) Worst Case Delay for `ADDER_WIDTH` = 64



Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack ^1	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Ex...	Clock Uncer...
Path 1	2.509	6	7	9	regA_Q_reg[18]/C	regCout_reg/D	5.355	1.377	3.978	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 2	2.509	6	7	9	regA_Q_reg[18]/C	regResult_reg[504]/D	5.355	1.377	3.978	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 3	2.509	6	7	9	regA_Q_reg[18]/C	regResult_reg[506]/D	5.355	1.377	3.978	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 4	2.509	6	7	9	regA_Q_reg[18]/C	regResult_reg[507]/D	5.355	1.377	3.978	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 5	2.509	6	7	9	regA_Q_reg[18]/C	regResult_reg[508]/D	5.355	1.377	3.978	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 6	2.509	6	7	9	regA_Q_reg[18]/C	regResult_reg[509]/D	5.355	1.377	3.978	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 7	2.509	6	7	9	regA_Q_reg[18]/C	regResult_reg[510]/D	5.355	1.377	3.978	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 8	2.517	6	7	9	regA_Q_reg[18]/C	regResult_reg[505]/D	5.347	1.369	3.978	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 9	2.517	6	7	9	regA_Q_reg[18]/C	regResult_reg[511]/D	5.347	1.369	3.978	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 10	2.678	6	7	8	regA_Q_reg[10]/C	regResult_reg[496]/D	5.186	1.409	3.777	8.0	sys_clk_pin	sys_clk_pin		0.035

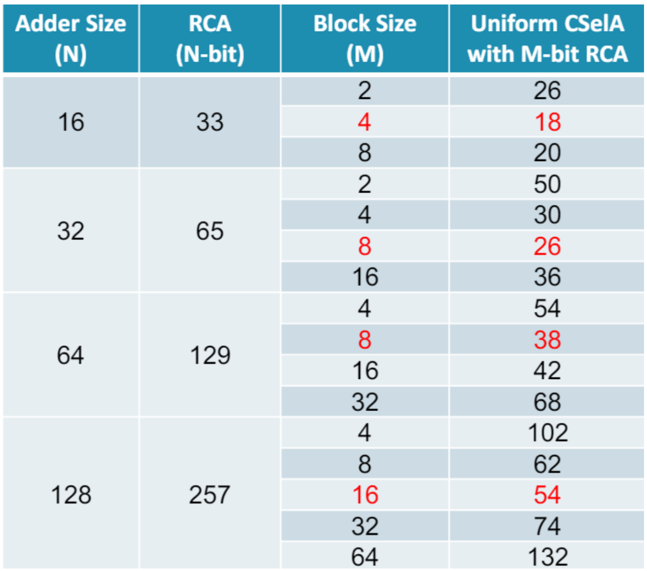
(c) Worst Case Delay for `ADDER_WIDTH` = 128

Figure 3: Synthesis Timing Reports

4 Comparison

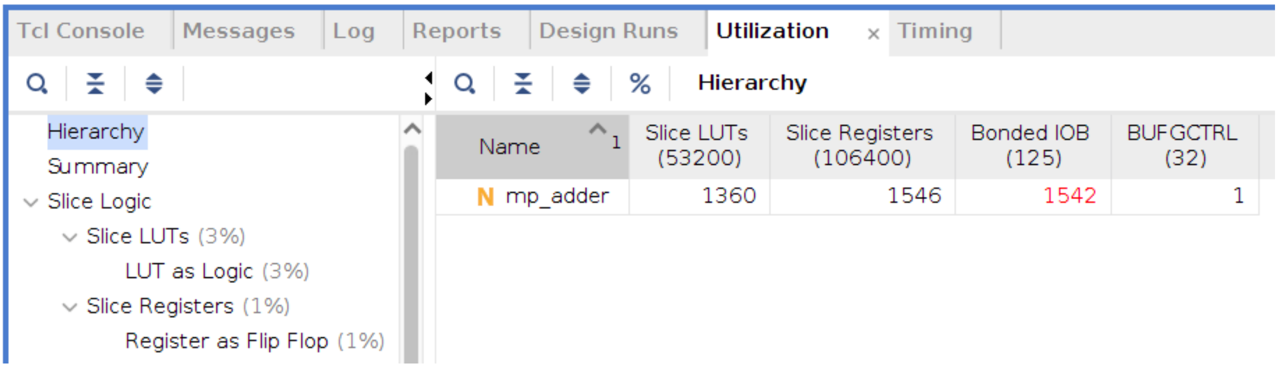
The worst-case delay of the CSelA for `ADDER_WIDTH` = 16 is higher than the one occurring with the RCA with the same adder width (Figure 6). This signifies that the longest path in the circuit is overcome faster in the improved circuit than in the initial one. In other words, the improved circuit propagates the carry bit faster.

However, this comes with a trade-off in area (Figure 7). The CSelA uses more area as it requires additional building blocks such as multiplexers and an additional RCA in comparison with the basic adder.



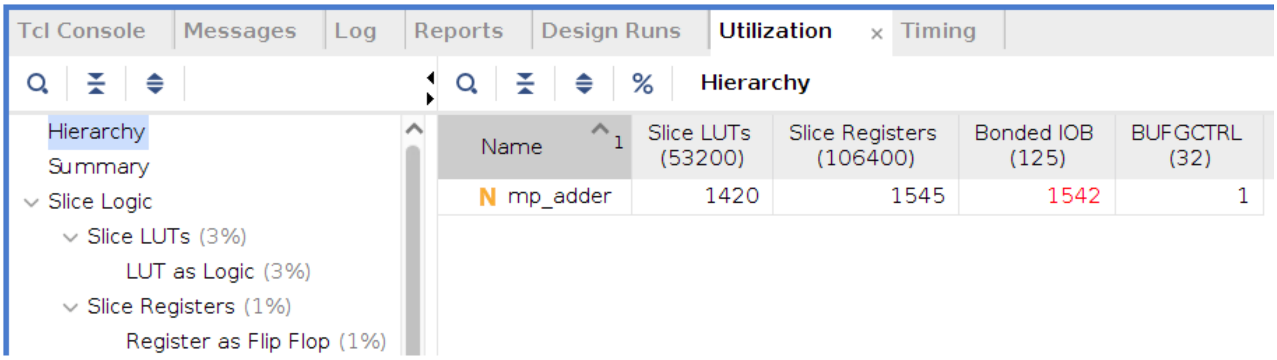
Adder Size (N)	RCA (N-bit)	Block Size (M)	Uniform CSelA with M-bit RCA
16	33	2	26
		4	18
		8	20
32	65	2	50
		4	30
		8	26
		16	36
64	129	4	54
		8	38
		16	42
		32	68
128	257	4	102
		8	62
		16	54
		32	74
		64	132

Figure 4: Logic gate delay for different block sizes and adder widths in a carry select adder



Name	Slice LUTs (53200)	Slice Registers (106400)	Bonded IOB (125)	BUFGCTRL (32)
N mp_adder	1360	1546	1542	1

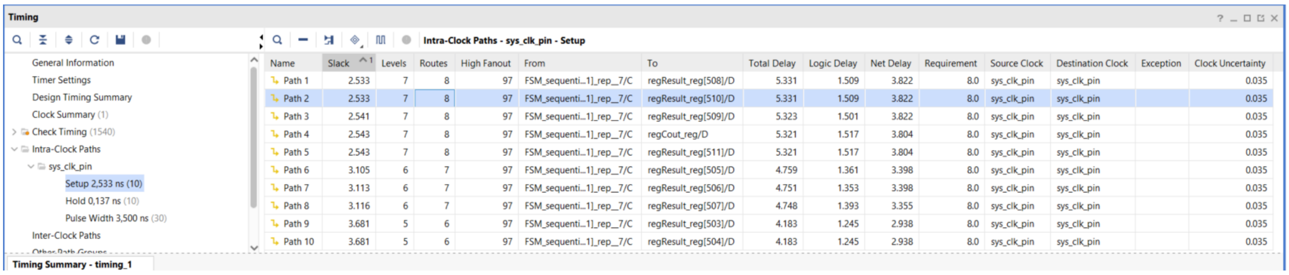
(a) Area cost for ADDER_WIDTH = 32



Name	Slice LUTs (53200)	Slice Registers (106400)	Bonded IOB (125)	BUFGCTRL (32)
N mp_adder	1420	1545	1542	1

(b) Area cost for ADDER_WIDTH = 64

Figure 5: Area costs for different adder widths



Timing

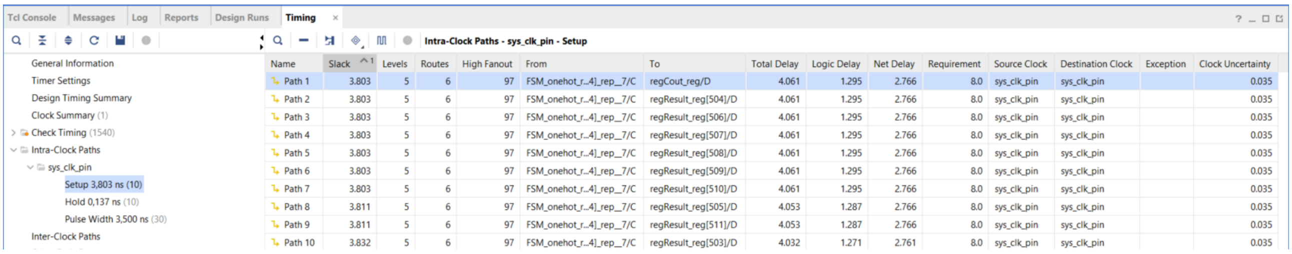
Intra-Clock Paths - sys_clk_pin - Setup

General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (1540)
Intra-Clock Paths
sys_clk_pin
Setup 2,533 ns (10)
Hold 0,137 ns (10)
Pulse Width 3,500 ns (30)
Inter-Clock Paths

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	2.533	7	8	97	FSM_sequenti...1]_rep__7/C	regResult_reg[508]/D	5.331	1.509	3.822	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 2	2.533	7	8	97	FSM_sequenti...1]_rep__7/C	regResult_reg[510]/D	5.331	1.509	3.822	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 3	2.541	7	8	97	FSM_sequenti...1]_rep__7/C	regResult_reg[509]/D	5.323	1.501	3.822	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 4	2.543	7	8	97	FSM_sequenti...1]_rep__7/C	regCout_reg/D	5.321	1.517	3.804	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 5	2.543	7	8	97	FSM_sequenti...1]_rep__7/C	regResult_reg[511]/D	5.321	1.517	3.804	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 6	3.105	6	7	97	FSM_sequenti...1]_rep__7/C	regResult_reg[505]/D	4.759	1.361	3.398	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 7	3.113	6	7	97	FSM_sequenti...1]_rep__7/C	regResult_reg[506]/D	4.751	1.353	3.398	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 8	3.116	6	7	97	FSM_sequenti...1]_rep__7/C	regResult_reg[507]/D	4.748	1.393	3.355	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 9	3.681	5	6	97	FSM_sequenti...1]_rep__7/C	regResult_reg[503]/D	4.183	1.245	2.938	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 10	3.681	5	6	97	FSM_sequenti...1]_rep__7/C	regResult_reg[504]/D	4.183	1.245	2.938	8.0	sys_clk_pin	sys_clk_pin		0.035

Timing Summary - timing_1

(a) Worst Case Delay of RCA for `ADDER_WIDTH` = 16



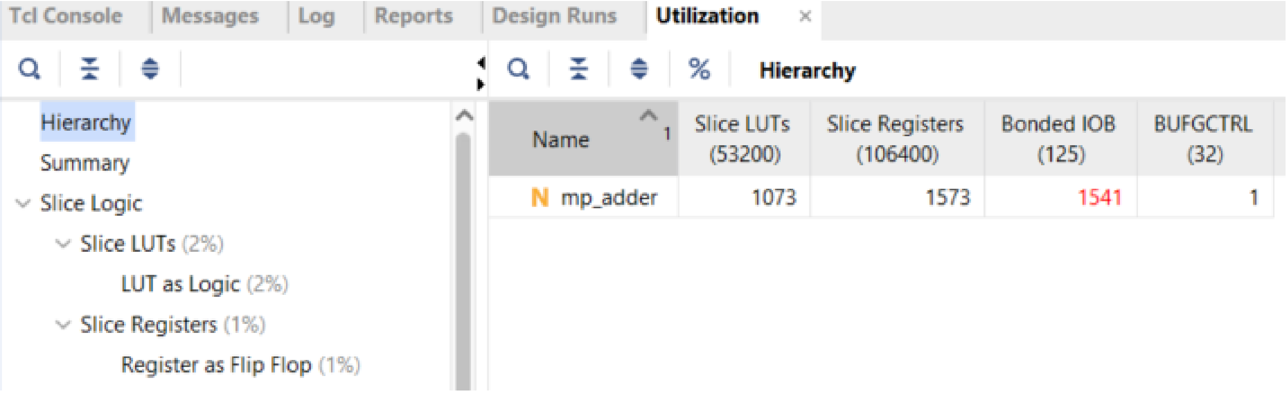
Intra-Clock Paths - sys_clk_pin - Setup

General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (1540)
Intra-Clock Paths
sys_clk_pin
Setup 3,803 ns (10)
Hold 0,137 ns (10)
Pulse Width 3,500 ns (30)
Inter-Clock Paths

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	3.803	5	6	97	FSM_onehot_r...4]_rep__7/C	regCout_reg/D	4.061	1.295	2.766	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 2	3.803	5	6	97	FSM_onehot_r...4]_rep__7/C	regResult_reg[504]/D	4.061	1.295	2.766	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 3	3.803	5	6	97	FSM_onehot_r...4]_rep__7/C	regResult_reg[506]/D	4.061	1.295	2.766	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 4	3.803	5	6	97	FSM_onehot_r...4]_rep__7/C	regResult_reg[507]/D	4.061	1.295	2.766	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 5	3.803	5	6	97	FSM_onehot_r...4]_rep__7/C	regResult_reg[508]/D	4.061	1.295	2.766	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 6	3.803	5	6	97	FSM_onehot_r...4]_rep__7/C	regResult_reg[509]/D	4.061	1.295	2.766	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 7	3.803	5	6	97	FSM_onehot_r...4]_rep__7/C	regResult_reg[510]/D	4.061	1.295	2.766	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 8	3.811	5	6	97	FSM_onehot_r...4]_rep__7/C	regResult_reg[505]/D	4.053	1.287	2.766	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 9	3.811	5	6	97	FSM_onehot_r...4]_rep__7/C	regResult_reg[511]/D	4.053	1.287	2.766	8.0	sys_clk_pin	sys_clk_pin		0.035
Path 10	3.832	5	6	97	FSM_onehot_r...4]_rep__7/C	regResult_reg[503]/D	4.032	1.271	2.761	8.0	sys_clk_pin	sys_clk_pin		0.035

(b) Worst Case Delay of CSelA `ADDER_WIDTH` = 16

Figure 6: Worst case delays for `ADDER_WIDTH` = 16 for RCA and CSelA

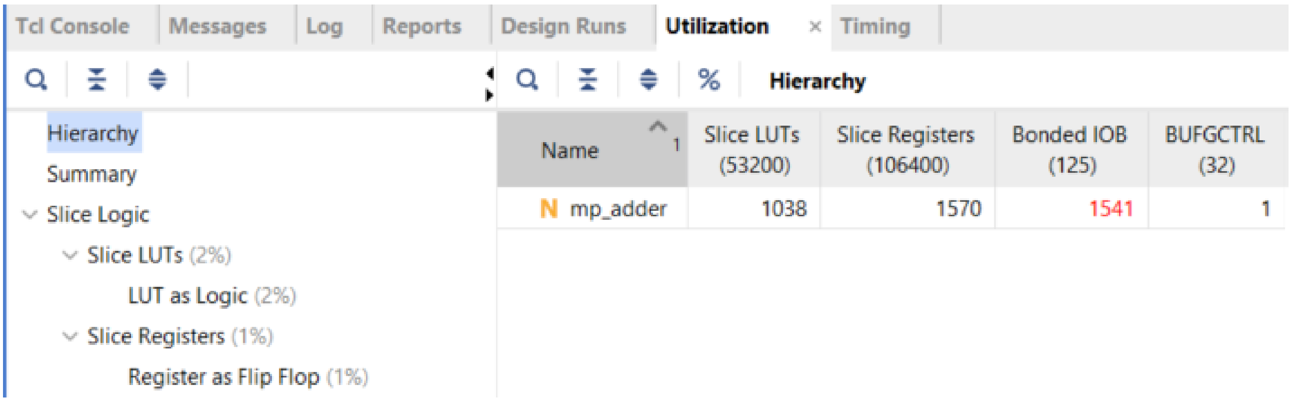


Hierarchy

Hierarchy
Summary
Slice Logic
Slice LUTs (2%)
LUT as Logic (2%)
Slice Registers (1%)
Register as Flip Flop (1%)

Name	Slice LUTs (53200)	Slice Registers (106400)	Bonded IOB (125)	BUFGCTRL (32)
N mp_adder	1073	1573	1541	1

(a) Area utilization of RCA for `ADDER_WIDTH` = 16



Hierarchy

Hierarchy
Summary
Slice Logic
Slice LUTs (2%)
LUT as Logic (2%)
Slice Registers (1%)
Register as Flip Flop (1%)

Name	Slice LUTs (53200)	Slice Registers (106400)	Bonded IOB (125)	BUFGCTRL (32)
N mp_adder	1038	1570	1541	1

(b) Area utilization of CSelA `ADDER_WIDTH` = 16

Figure 7: Area utilization for `ADDER_WIDTH` = 16 for RCA and CSelA