M2 Master E3A Integration Circuits Systems

Design of generic square root computing architectures on FPGA using VHDL

IFTAKHER Mohammed Akib – No. 22211204

Instructor: Professor Hervé MATHIAS

Contents

List of Figures

List of Tables

Introduction

The project entails developing multiple architectures for computing the square root of a given number. Two distinct square root computing methods with various design restrictions have been implemented. VHDL has been used to create and test these designs.

Newton Method: The Newton-Raphson technique (also known as Newton's method) is a method for quickly calculating the root of a real-valued function f(x) = 0. It is based on the premise that a straight-line tangent to a continuous and differentiable function can approximate it (*Newton Raphson Method | Brilliant Math & Science Wiki*, n.d.).

VHDL: The Very High-Speed Integrated Circuit Hardware Description Language (VHDL) is a hardware description language. It is used in electrical design automation to define mixed-signal and digital systems such as ICs (integrated circuits) and FPGAs (field-programmable gate arrays) (Solutions, 2022).

FPGA: FPGAs (Field Programmable Gate Arrays) are integrated circuits that are frequently marketed off-the-shelf. They are referred to as 'field programmable' because they allow customers to alter the hardware after the production process to satisfy specific use case requirements. This enables in-place feature updates and bug fixes, which is very important for remote deployments.

FPGAs are made up of customizable logic blocks (CLBs) and a set of programmable interconnects that allow the designer to link and modify the blocks to do anything from simple logic gates to complicated operations. Full SoC architectures with many processes may be implemented on a single FPGA chip (Arm Ltd., n.d.).

Objective

The objective of this project is to develop multiple architectures for computing the square root of a given number. Two different algorithms have been utilized to design 5 different architectures. The first architecture solely based on the Newton's method and the rest four architectures based on a distinctive algorithm called square root iterative algorithm. While completing this project, the algorithm and its output has been analysed and all the necessary data has been documented.

1) Behavioural approach:

Architecture 1:

This design is based on the Newton method. In this specific architectural design, the function given to compute the root is the f(x) = x^2 - N function.

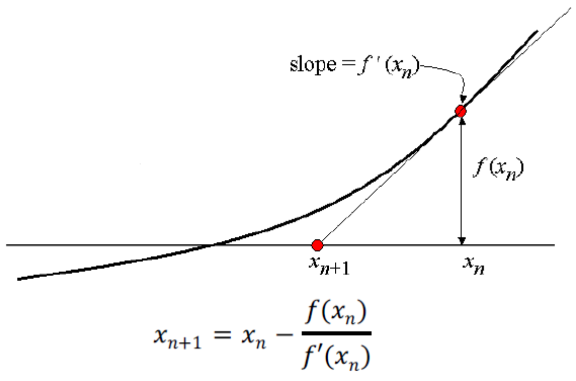


Figure 1: Newton's Method principle

It will calculate the square root of a 2n-bit unsigned input N_in. In this case, an arbitrary starting guess x0 is used, and each iteration yields a new result that is closer to the answer using Newton's formula. When two successive iterations yield the same result, which is the integer square root of N in, the algorithm terminates. To complete the algorithm, the Newton's method has been simplified.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_{n+1} = x_n - \frac{{x_n}^2 - N}{2x_n}$$

$$x_{n+1} = x_n - \left(\frac{x_n}{2} - \frac{N}{2x_n}\right)$$

$$x_{n+1} = \frac{x_n}{2} + \frac{N}{2x_n}$$

$$x_{n+1} = \left(x_n + \frac{N}{x_n}\right)/2$$

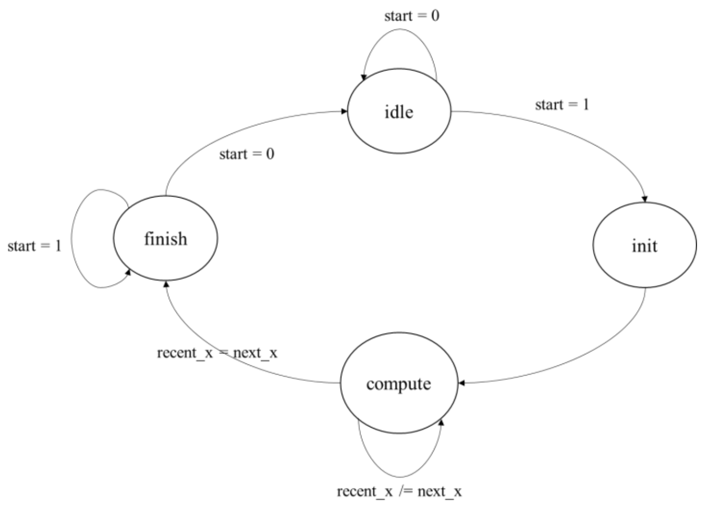


Figure 2: Final State Machine for Architecture 1.

The states are explained below,

1. **idle**: The procedure can begin once start is set to '1'. Depending on where the program starts, it can either go to compute or remain idle. Going to init states is required to initialize the arbitrary values.
2. **init**: Recent x and next x have been defined for the intention of the computation in this state.
3. **compute**: next_x has been determined in this stage based on recent_x and the specified initial value (N_in) through testbench. In the following phase, it is compared against recent_x to ensure equivalence. The state either moves to the finish state or remains in the compute state based on equality.
4. **finish**: The program either switches to idle states and waits for additional input, or it stays in the finish state and generates the results that are initialized by next_x. The results only accept Last 32 bits which remarks as a right shifted values of next_x.

In the **figure 3**, it can be noted that it takes longer time or in another word cycles to generate output when the input is bigger. For example, when the input is 64 the cycle number is 20. Again, the cycle number is 11 when the input is 36. But in both cases, the correct output has been generated.

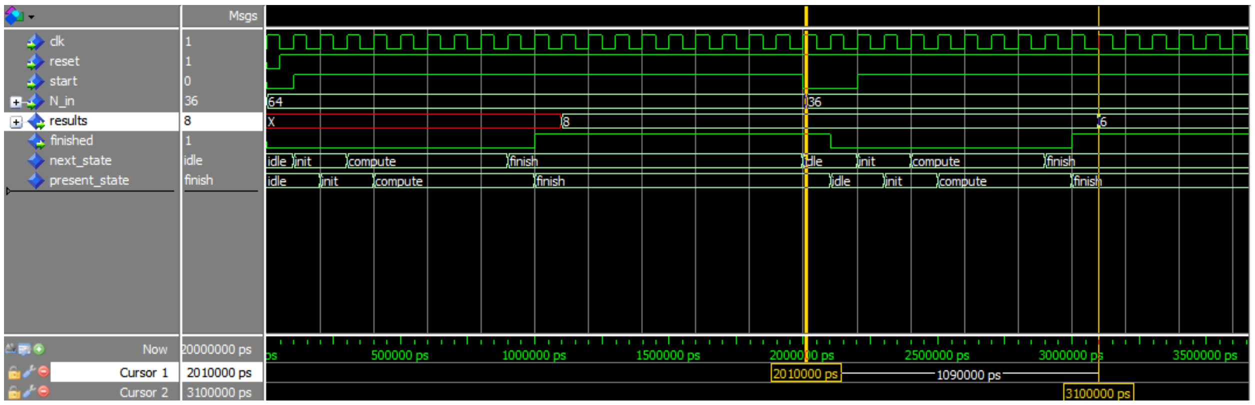
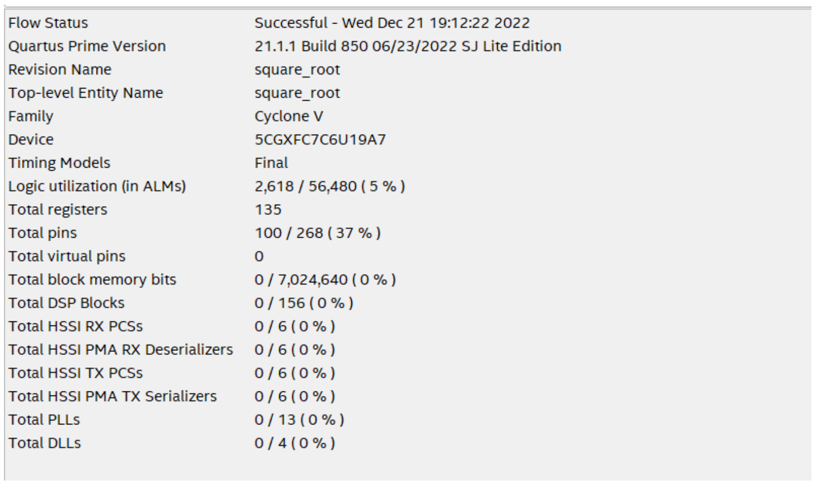


Figure 3: ModelSim waveform generated for architecture 1.



Flow Status	Successful - Wed Dec 21 19:12:22 2022
Quartus Prime Version	21.1.1 Build 850 06/23/2022 SJ Lite Edition
Revision Name	square_root
Top-level Entity Name	square_root
Family	Cyclone V
Device	5CGXFC7C6U19A7
Timing Models	Final
Logic utilization (in ALMs)	2,618 / 56,480 (5 %)
Total registers	135
Total pins	100 / 268 (37 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
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 4: Flow summary of architecture 1.

Table 1: Tables representing the frequency, clock cycles and resources for Architecture 1.

Maximum working frequency	Worst-case required number of clock cycles to perform the computations	The resources used in the FPGA
Fmax 4.28 MHz	20 for the input 64	Total registers: 135 Logic utilization (in ALMs): 2,611 / 56,480 (5 %) Total Pins: 100/268 (37%)

The computation time to get the result of N_in=64 is:

$$\text{Computation time} = \text{number of clock cycles} \times \frac{1}{F_{max}} = 20 \times \frac{1}{4.28 \times 10^6} = 4672.89719626 \text{ ns}$$

Architecture 2:

This structure is sequential, with the goal of completing the second method while minimizing the amount of clock cycles necessary to complete the computation. It is created with modest modifications to the FSM shown in Figure 2.

```
Load N ;
$V = 2^{2n-2}$ ;
$Z = 0$ ;
For $i = n-1$ downto 0 do
    $Z = Z+V$ ;
    If $(N-Z)>=0$ then
        $N=N-Z$ ;
        $Z = Z+V$ ;
    Else
        $Z = Z-V$ ;
    End If ;
    $Z = Z/2$ ;
    $V = V/4$ ;
End For ;
Result = Z;
```

Figure 5: Considered Square root iterative algorithm.

Similar to the previous architecture, this architecture has four states.

1. **idle**: Similar to architecture 1.
2. **init**: Based on the start, when the operation comes to init state, N_dup, Z, V, i have been initialized. Then the program goes to compute state.
3. **compute**: In this state, a conditional statement compares "i" with 0. Based on the feedback, the state either goes to finish state or remains in compute state. If it is in the compute states, Z is updated with V and compared with N_dup. Then, either N_dup is subtracted using Z and once again Z is updated with the addition of V. If it is not this case, Z will be updated by subtracting V. Later, Z, V and i are being divided or reduced and the whole operation stays in compute state.
4. **finish**: The finish state replicates the similar operation observed in the architecture 1.

In **figure 5**, it can be seen that two N_in values are being passed to the architecture from the testbench where it is successfully computed to provide the square root values of those inputs. Another thing here can be observed that in both times, the computation requires similar cycles numbers.

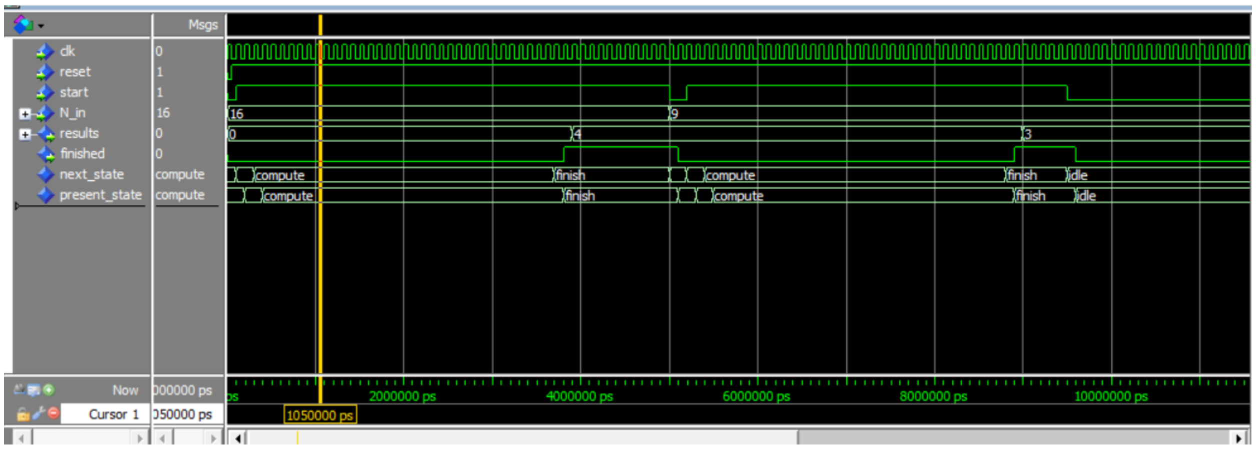
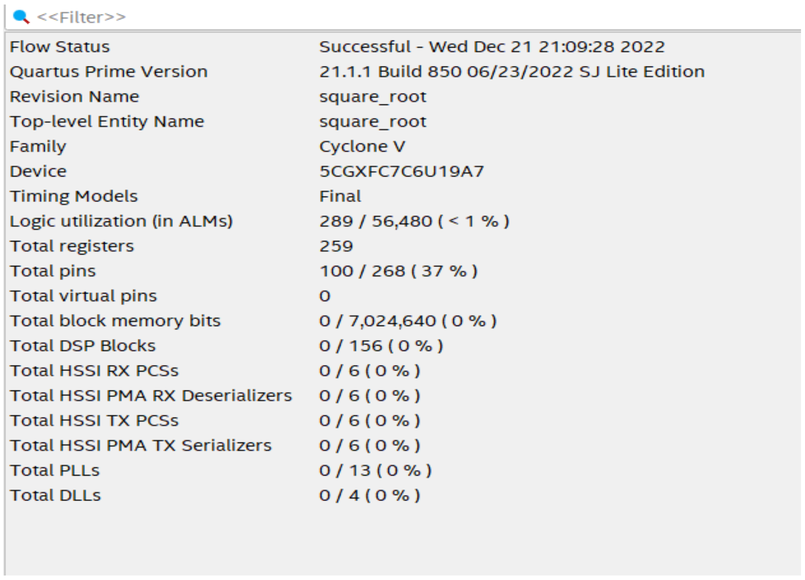


Figure 6: ModelSim waveform generated for architecture 2.



<<Filter>>

Flow Status	Successful - Wed Dec 21 21:09:28 2022
Quartus Prime Version	21.1.1 Build 850 06/23/2022 SJ Lite Edition
Revision Name	square_root
Top-level Entity Name	square_root
Family	Cyclone V
Device	5CGXFC7C6U19A7
Timing Models	Final
Logic utilization (in ALMs)	289 / 56,480 (< 1 %)
Total registers	259
Total pins	100 / 268 (37 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
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 7: Flow summary of architecture 2.

Table 2: Tables representing the frequency, clock cycles and resources for Architecture 2.

Maximum working frequency	Worst-case required number of clock cycles to perform the computations	The resources used in the FPGA
Fmax 125.44 MHz	37.5-37.9	Total registers 259 Logic utilization (in ALMs) 289 / 56,480 (< 1 %) Total Pins: 100/268 (37%)

The computation time to get the result of N_in is:

$$\text{Computation time} = \text{number of clock cycles} \times \frac{1}{F_{max}} = 37.5 \times \frac{1}{125.44 \times 10^{6}} = 298.947704082 \text{ ns}$$

Architecture 3:

This architecture is based on a combinatorial architecture for square root. The significant difference here is that the use of clock

Here, V register has been initialized in a unique way so that each of the value of V register has been shifted 2 bits right compared to its previous values.

Again, in the second for loop it can be noted that Z_temp register has been initialized with previous Z register value and previous V register value. For example, 30th Z_ temp register has been initialized with 31st V register.

In the next line, Z_div_2 has been initialized to half of previous Z register value. There is a formula optimization done here which is related to the initialization of Z register part at the end of the square root iterative algorithm.

Afterwards, previous input register X has been compared with the Z_temp register and a new register named compare has been initialized using when statement since it is a combinational design. Based on this compare register, current input register has been defined either by subtracting Z_temp from previous input register or keeping the previous input register value.

Again, based on compare register, current Z register has been initialized by adding the previously declared Z_div_2 with previous V register which is already 2 bits right shifted. In another case, if the compare register is not respecting the condition, current Z register will be initialized to just Z_temp.

In another word, the square root iterative algorithm has been modified to design combinatorial architecture. The whole algorithm has been explained below,

if N_in > Z:

$$Z_i = (Z_{i+1}+V_{i+1}) + V_{i+1}$$

$$Z_i = \frac{Z_i}{2} = \frac{(Z_{i+1}+V_{i+1})+V_{i+1}}{2}$$

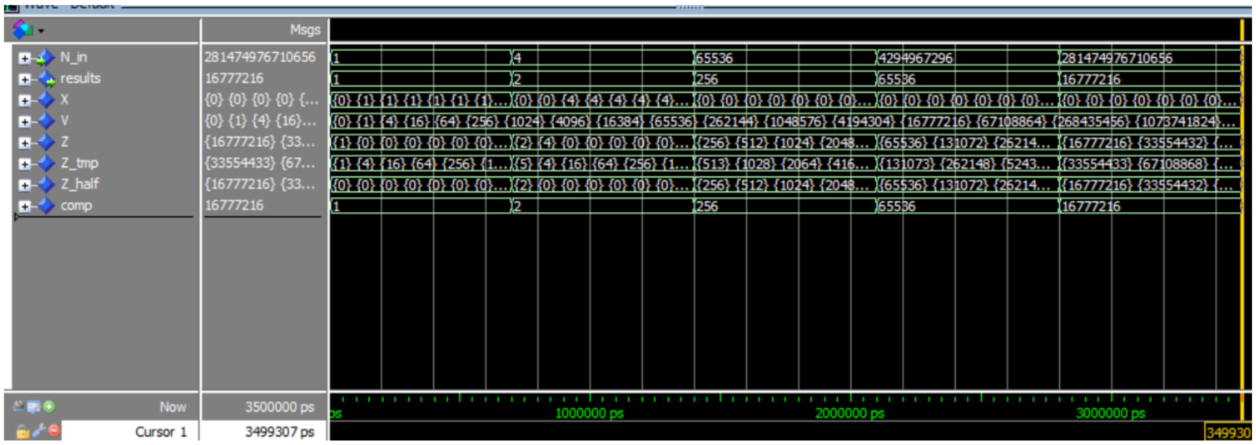
or if N_in < Z:

$$Z_i = (Z_{i+1}+V_{i+1}) - V_{i+1} = Z_{i+1}$$

$$Z_i = \frac{Z_i}{2} = \frac{Z_{i+1}}{2}$$

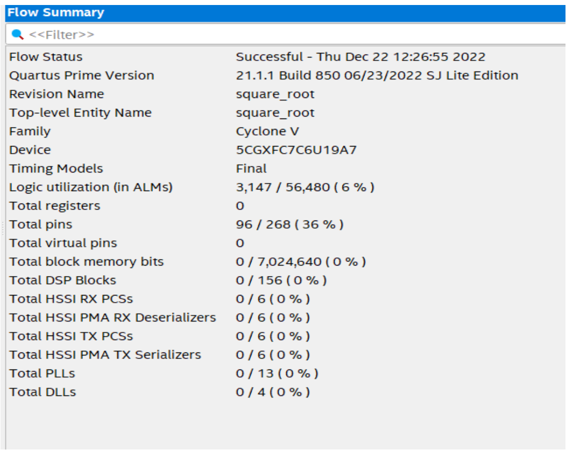
In the **figure 9**, it can be seen that the clock is absent. Since it is combinational design, the output is instant. As a result, register and maximum frequency is missing from the waveform.

But with the addition of the clock signal, both register and maximum frequency can be recovered as it has been noted in **figure 11**.



	Msgs
N_in	281474976710656
results	16777216
X	{0} {0} {0} {0} {…
V	{0} {1} {4} {16}…
Z	{16777216} {33…
Z_tmp	{33554433} {67…
Z_half	{16777216} {33…
comp	16777216
Now	3500000 ps
Cursor 1	3499307 ps

Figure 8: ModelSim waveform generated for architecture 3.

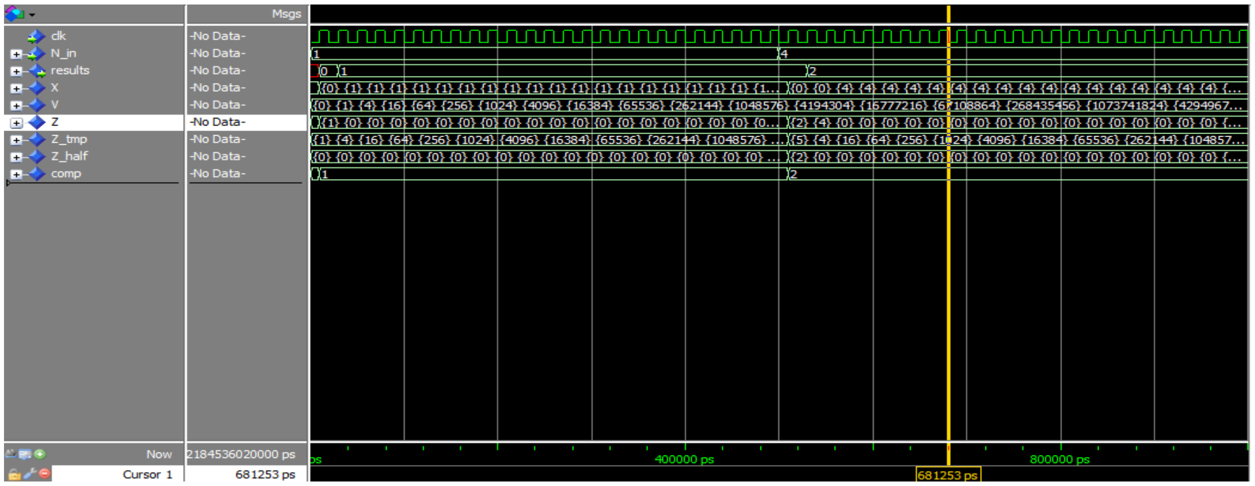


Flow Summary

Flow Status	Successful - Thu Dec 22 12:26:55 2022
Quartus Prime Version	21.1.1 Build 850 06/23/2022 SJ Lite Edition
Revision Name	square_root
Top-level Entity Name	square_root
Family	Cyclone V
Device	5CGXFC7C6U19A7
Timing Models	Final
Logic utilization (in ALMs)	3,147 / 56,480 (6 %)
Total registers	0
Total pins	96 / 268 (36 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
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 %)

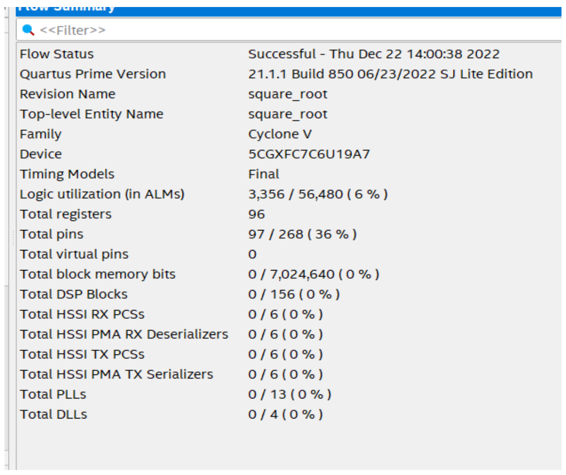
Figure 9: Flow summary of architecture 3.

With the addition of the clock in the architecture:



	Msgs
clk	-No Data-
N_in	-No Data-
results	-No Data-
X	-No Data-
V	-No Data-
Z	-No Data-
Z_tmp	-No Data-
Z_half	-No Data-
comp	-No Data-
Now	2184536020000 ps
Cursor 1	681253 ps

Figure 10: ModelSim waveform generated for architecture 3 with clock signal.



Flow Status	Successful - Thu Dec 22 14:00:38 2022
Quartus Prime Version	21.1.1 Build 850 06/23/2022 SJ Lite Edition
Revision Name	square_root
Top-level Entity Name	square_root
Family	Cyclone V
Device	5CGXFC7C6U19A7
Timing Models	Final
Logic utilization (in ALMs)	3,356 / 56,480 (6 %)
Total registers	96
Total pins	97 / 268 (36 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
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 11: Flow summary of architecture 3 with clock.

Table 3: Tables representing the frequency, clock cycles and resources for Architecture 3.

Maximum working frequency	Worst-case required number of clock cycles to perform the computations	The resources used in the FPGA
With clock signal: Fmax 7.12 MHz	With clock signal: 35 cycles	Total registers 0 Logic utilization (in ALMs): 3,147 / 56,480 (6 %) Total Pins: 96/268 (36%) With clock signal: Total registers 96 Logic utilization (in ALMs): 3,356 / 56,480 (6 %) Total Pins: 97/268 (36%)

The computation time to get the result of N_in is:

$$\text{Computation time} = \text{number of clock cycles} \times \frac{1}{F_{max}} = 35 \times \frac{1}{7.12 \times 10^6} = 4915.73033708 \text{ ns}$$

Architecture 4:

The architecture is based on architecture 3 which has been modified to implement a pipelined architecture

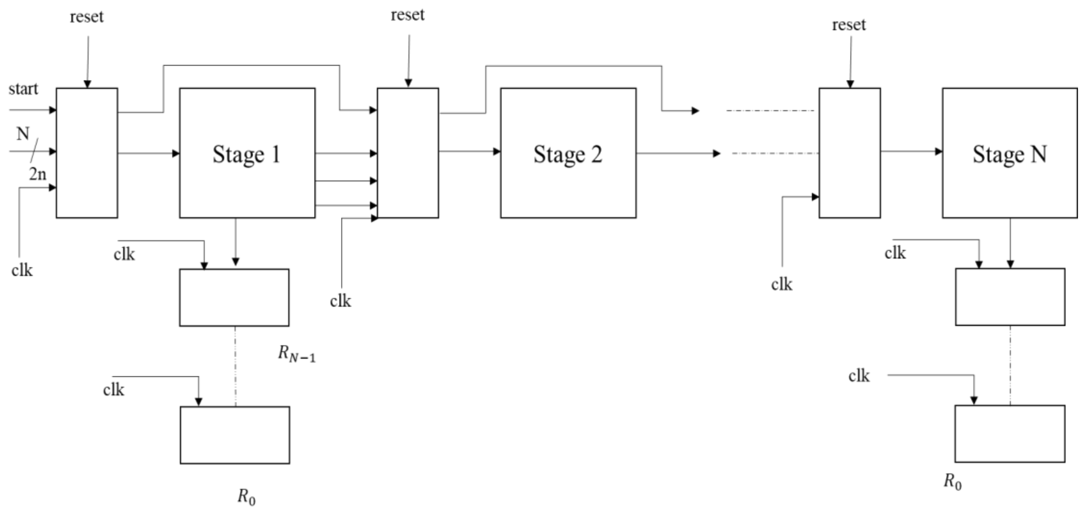


Figure 12: Datapath based architecture 4.

This architecture is combination of both combinatorial and sequential design. This architecture is implemented using register which is based on arrays. In total 7 signals have been declared based on this register type.

Similar to the architecture 3, a generate loop has been used to generate 32 V register value starting from V (1) to V (32). All the values are 2 bits right shifted from the MSB of its previous value except V (32). Moreover, V (0) has been initialized to 0.

Then the combinatorial part has been implemented which functions exactly as mentioned in architecture 3. Then actual pipeline architecture which is based on sequential architecture has been developed. Inside this architecture, 3 signal such as clock, reset and start controls the whole process.

In the beginning, reset is activated where register such as X, Z and start_shift signal has been set to 0 along with the results and finished.

Then in the next step when reset is deactivated and clock signal is initiated, all the start_shift signal has been initialized in a way that first MSB of start_shift receive the start signal from the testbench and pass its old value to the 2nd highest MSB and 2nd highest MSB of start_shift pass its old value to its previous MSB (3rd highest MSB) and so on.

In the next step, 32nd X register accepts input from the testbench N_in and 32nd Z register has been assigned to 0 when start_shift is 1.

Later in the actual pipelining stage, a loop has been implemented in a way that 31st down to 0 X register is initialized with similar N_out when respective start_shift is activated. Same case happens for Z who accepts value from Z_out.

Finally, when LSB of start_shift is activated, results are being produced which is a copy of initial Z_out register.

Since it is a pipeline structure, the computation theoretically takes 1 cycles.

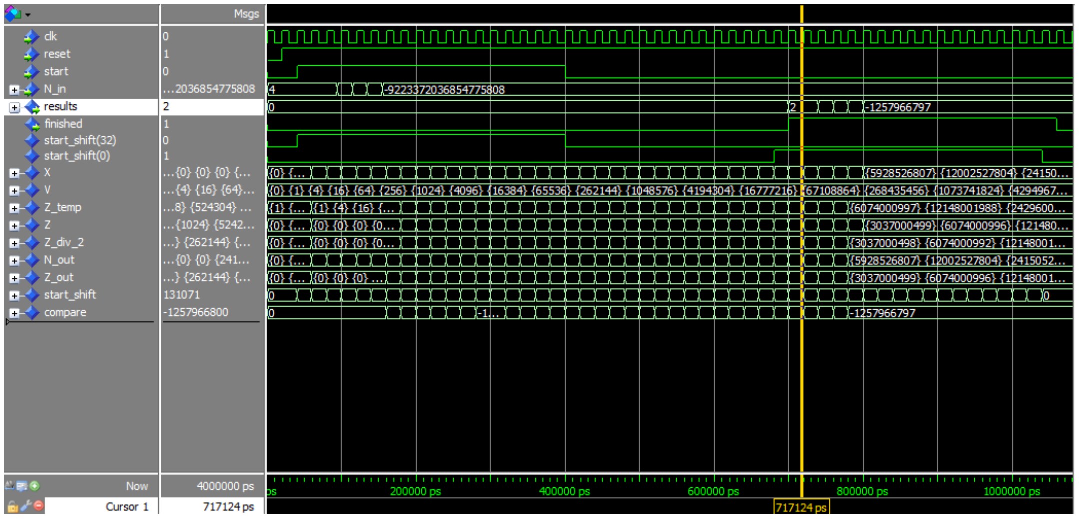
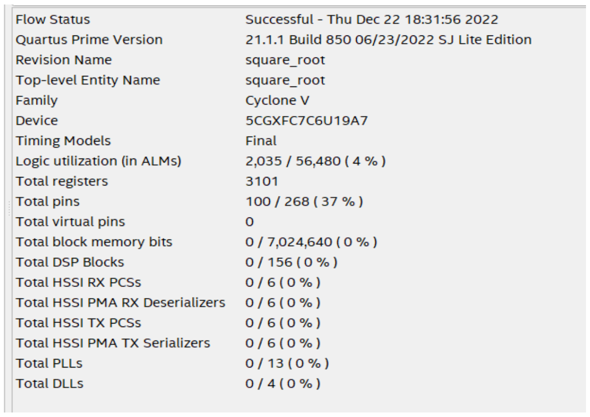


Figure 13: ModelSim waveform generated for architecture 4.



Flow Status	Successful - Thu Dec 22 18:31:56 2022
Quartus Prime Version	21.1.1 Build 850 06/23/2022 SJ Lite Edition
Revision Name	square_root
Top-level Entity Name	square_root
Family	Cyclone V
Device	5CGXFC7C6U19A7
Timing Models	Final
Logic utilization (in ALMs)	2,035 / 56,480 (4 %)
Total registers	3101
Total pins	100 / 268 (37 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
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 14: Flow summary of architecture 4.

Table 4: Tables representing the frequency, clock cycles and resources for Architecture 4.

Maximum working frequency	Worst-case required number of clock cycles to perform the computations	The resources used in the FPGA
Fmax 136.63 MHz	1 cycle	Logic utilization (in ALMs) 2,035 / 56,480 (4 %) Total registers 3101 Total pins 100 / 268 (37 %)

The computation time to get the result of N_in is:

$$\text{Computation time} = \text{number of clock cycles} \times \frac{1}{F_{max}} = 1 \times \frac{1}{136.63 \times 10^{6}} = 7.31903681476 \text{ ns}$$

2) Structural approach:

Architecture 5:

For computing the square root of the input, Architecture 5 employs a structural technique. It is made up of a Datapath and various blocks (multiplexer, register, and adder/subtractor).

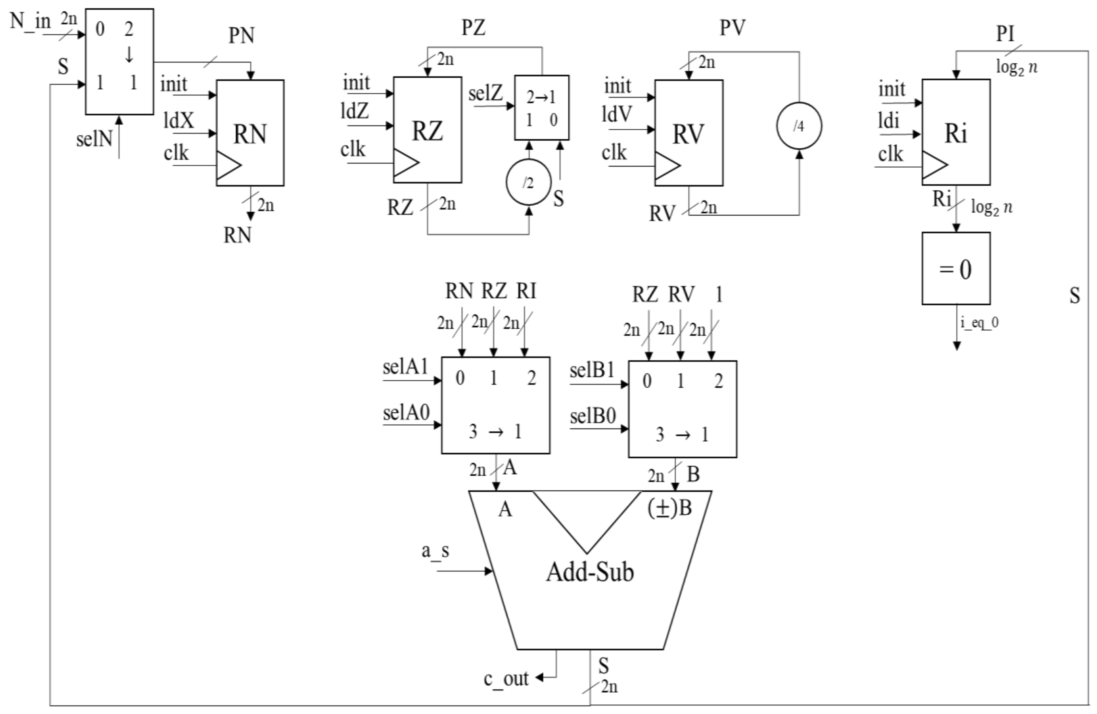


Figure 15: Datapath for architecture 5.

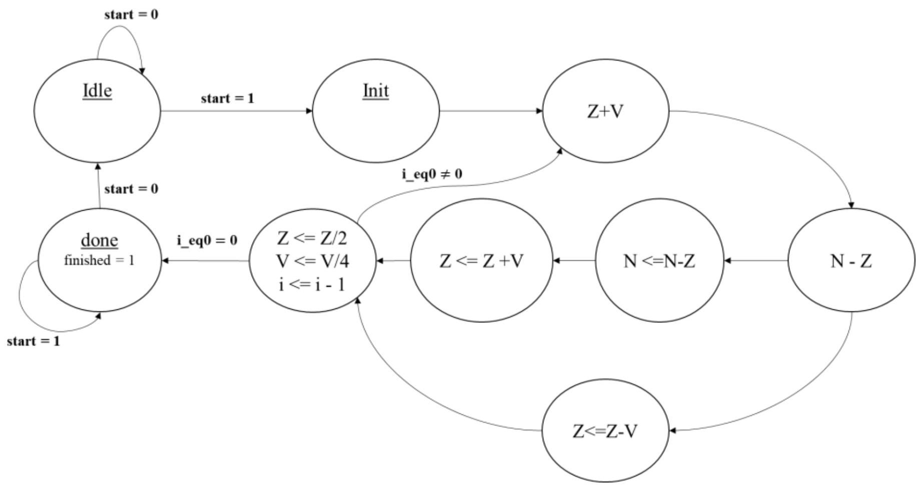


Figure 16: Final State Machine for Architecture 5.

Moreover, in figure 15, an arithmetic block can be noticed which is implemented to reduce the entire the circuit size.

Here, three registers have been implemented to utilise the output. These registers correspond to N, Z and V. It has total 5 inputs and one output.

In another block, adder and subtractor has been implemented where it receives three inputs consisting of two operand and one operator) and generates one output denoted as S.

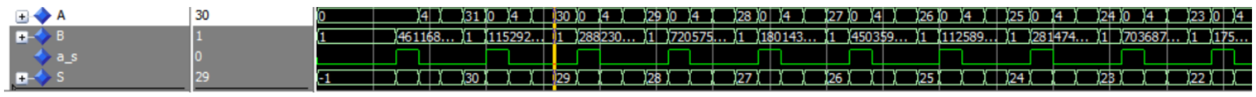


Figure 17: ModelSim zoomed image of the adder-subtractor blocks operation,

Another important component or block has been implemented here is multiplexers who accepts four inputs and produces one outputs. Based on the mux_select signal, out of three one input is chosen and produced as an output.

By observing the figure 18, it can be seen that for each provided input, a correct square root has been produced

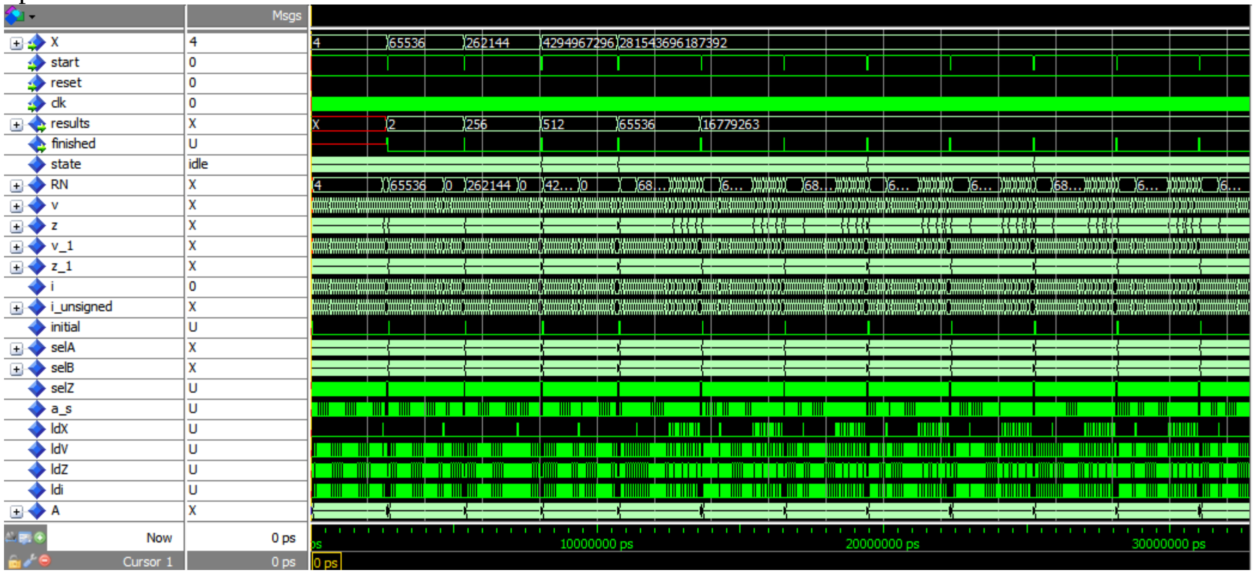
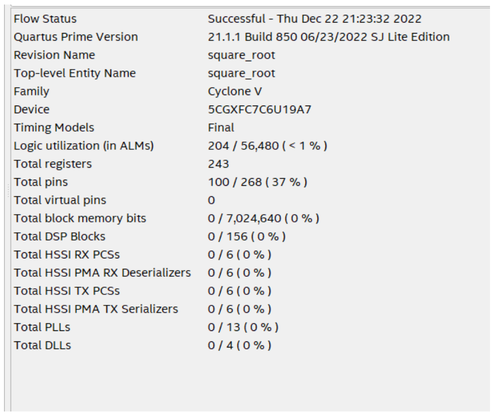


Figure 18: ModelSim waveform generated for architecture 5.



Flow Status	Successful - Thu Dec 22 21:23:32 2022
Quartus Prime Version	21.1.1 Build 850 06/23/2022 SJ Lite Edition
Revision Name	square_root
Top-level Entity Name	square_root
Family	Cyclone V
Device	5CGXFC7C6U19A7
Timing Models	Final
Logic utilization (in ALMs)	204 / 56,480 (< 1 %)
Total registers	243
Total pins	100 / 268 (37 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
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 19: Flow summary of architecture 5.

Table 5: Tables representing the frequency, clock cycles and resources for Architecture 5.

Maximum working frequency	Worst-case required number of clock cycles to perform the computations		The resources used in the FPGA
Fmax 130.07 MHz	For input	cycle	Logic utilization (in ALMs) 204 / 56,480 (< 1 %) Total registers 243 Total pins 100 / 268 (37 %)
	4	133	
	65536	132	
	4294967296	132	

The computation time to get the result of N_in is:

$$\text{Computation time = number of clock cycles} \times \frac{1}{F_{max}} = 133 \times \frac{1}{130.07 \times 10^6} = 1022.52633198\text{ns}$$

Comparison

Table 6: Table representing performance, cost, design complexity.

Architecture	ALM	Register	Pins	Computational time (ns)
1	2,611 / 56,480 (5 %)	135	100/268 (37%)	4672.89719626
2	289 / 56,480 (< 1 %)	259	100/268 (37%)	298.947704082
3	3,147 / 56,480 (6 %)	0	96/268 (36%)	-
3 (with clock)	3,356 / 56,480 (6 %)	96	97/268 (36%)	4915.73033708 ns
4	2,035 / 56,480 (4 %)	3101	100 / 268 (37 %)	7.31903681476
5	204 / 56,480 (< 1 %)	243	100 / 268 (37 %)	1022.52633198

When it comes to the performance of the architecture, it can be concluded that architecture 3 is the fastest one. But with the addition of clock, it can be seen that it is the slowest one. Beside the architecture 3, the fastest architecture is architecture number 4 and then 2,5 and finally 1. When it comes to cost, the architecture 3 doesn't have any register due to its combinatorial design where the other architectures have significant number of registers. So, from the point of view of register, architecture 4 has highest cost because of its pipeline structure. Where other architecture has around 100 to 300 registers. Once again, architecture 3 has lowest number of pins where other architecture has slightly a greater number of pins. So, when it comes to pin, all the registers have almost similar numbers. Again, ALM is designed to maximize performance and resource usage. From the table above, it can clearly state that, architecture 3 exhibits maximum performance. The other architecture followed in this sequence 1, 4, 2, 5. Architecture 5 is the most complicated in terms of complexity since it necessitates the design and usage of multiple blocks, as well as a control unit relying on an FSM with nine states to govern these blocks.

Conclusion

Through this project, two different algorithms have been implemented. One is solely based on the newton method and another method is called iterative method. Based on the iterative method further modification has been carried out to observe the combinatorial and sequential behaviour of the circuit. Moreover, unique designed such as pipelining has been implemented. At the end, a structural approach has been approached to limit the size of the architecture. Furthermore, a comparison of all architectures has been drawn to check the performance, resource being used and so on.

Reference

[1] *Newton Raphson Method | Brilliant Math & Science Wiki*. (n.d.). https://brilliant.org/wiki/newton-raphson-method/

[2] Solutions, C. P. (2022, October 13). *Hardware Description Languages: VHDL vs Verilog, and Their Functional Uses*. https://resources.pcb.cadence.com/blog/2020-hardware-description-languages-vhdl-vs-verilog-and-their-functional-uses

[3] Arm Ltd. (n.d.). *What is FPGA?* Arm | the Architecture for the Digital World. https://www.arm.com/glossary/fpga