CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will discuss more on software development and hardware design. This chapter is divided into three main parts. The first part is research materials. The second part is analysis level that focuses on software circuit design. The third part is identifying level where hardware implementation approached.

3.2 Research

This project began with researching the related materials based on the title project. It comes from many sources such as internet, journal or paper, lecturer notes and books. Those materials are about carry save architecture and what it’s needed. Supervisor then confirmed that information was right and any other needed information being asked. This process continued until the supervisor satisfied with the information. CSA mostly applied in multipliers and arithmetic computers. Since the carry save adder is very important for many application, the faster execution circuit being designed in the next level. Figure 3.1 shows the flow of researches.

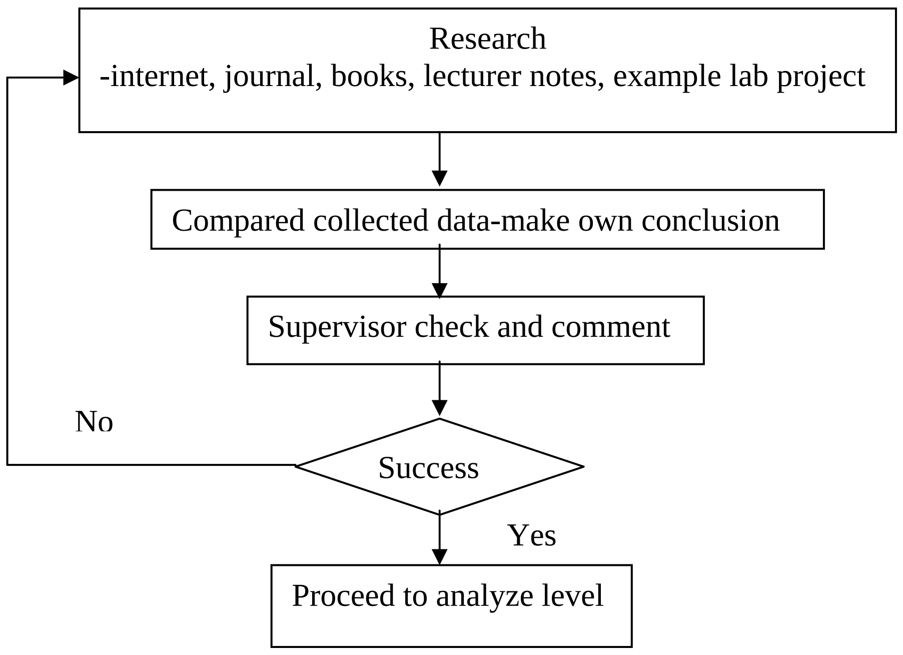

Research
-internet, journal, books, lecturer notes, example lab project
Compared collected data-make own conclusion
Supervisor check and comment
No
Success
Yes
Proceed to analyze level

Figure 3.1: Flow chart of research

3.3 Analysis

3.3.1 Software

This project analyzes the circuit by designing circuit logic using Altera Quartus II software. The setting family project is Flex10K with specific device of EPF10K70RC240-4. This setting are based on the UP2 board that will be used in the next identify level. Five different full adders were created to design carry save adder (CSA). Among these five circuits, the circuit with the smallest delay (refer figure 2.2) has been chosen in order to design the fastest CSA [27]. Following the circuit that has been designed:

- Full adders
- 4-bits CSA
- 16-bits CSA
- 4-bits RCA
- 17-bits RCA
- Six operands of 16-bits CSA
- Three level six operands of 16-bits CSA with RCA

Five different full adders being designed in three levels six operands CSA with RCA (ripple carry adder). These circuits were designed based on journals [27], [28], [33], [34] which has been shown in the Figure 2.4 until Figure 2.8. Each design used different logic gates that contribute same result as a basic full adder. These circuits were compiled using compiler tool. Through the compilation process, the delay can be verified at t_{PD} in each full adder design. The meaning of t_{PD} is the time required for signals from an input pin to propagate through combinational logic and appear at an external output pin.

The symbol block diagram of full adder from circuit designs was created to design 4-bits CSA and 4-bits RCA as shown in Figure. The block diagram symbols of these circuits have been created to design 16-bits CSA (refer to Figure 3.2) and 16-bits RCA. Six operands of 16-bits CSA used 16-bits CSA block symbol block diagram. The block diagram of six operands of 16-bits CSA and 17-bits RCA has been created to design the final stage of major design in this project; three level six operands of 16-bits CSA. The smallest delay within these compared circuit design was chosen to modify it into a faster circuit. Each circuit was created a waveform since the very basic circuit to check whether the circuit was connected precisely. Table 3.1 shows the truth table of 16-bits CSA for a random valued. The output can be calculated by using Equation 1.0 as shown in Chapter 2.

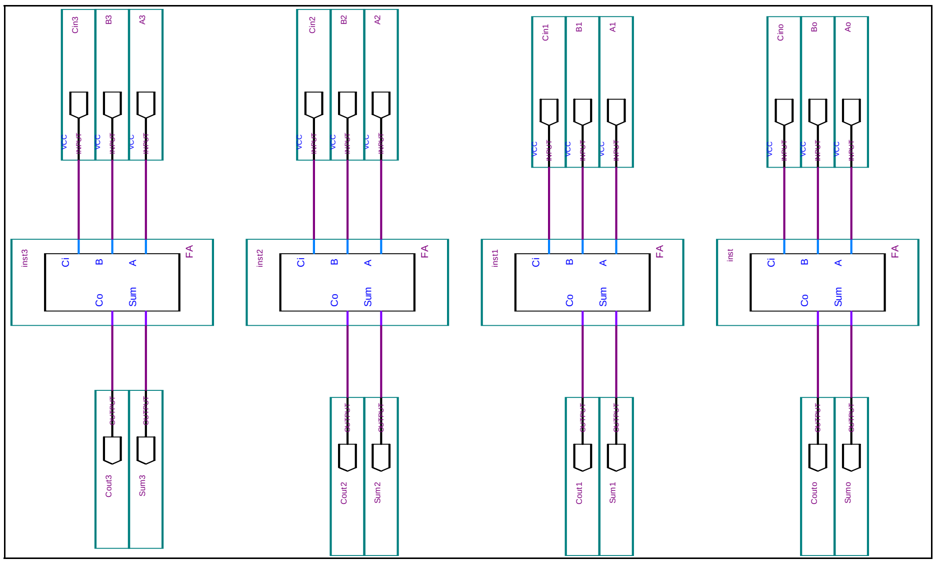


Figure 3.2: 16-bits CSA

Only then the bigger circuit designed. This project designed is six operands that contained four 16-bits CSA in three levels, which can be seen in the Figure 3.2. The goal of this design is to see how this CSA circuit functioning through many levels. The waveform of six operands of 16-bits CSA design has been built using simulation tool. While simulation process, the output has been checked with the carry save adder equation and compared it with 16-bits CSA circuit design.

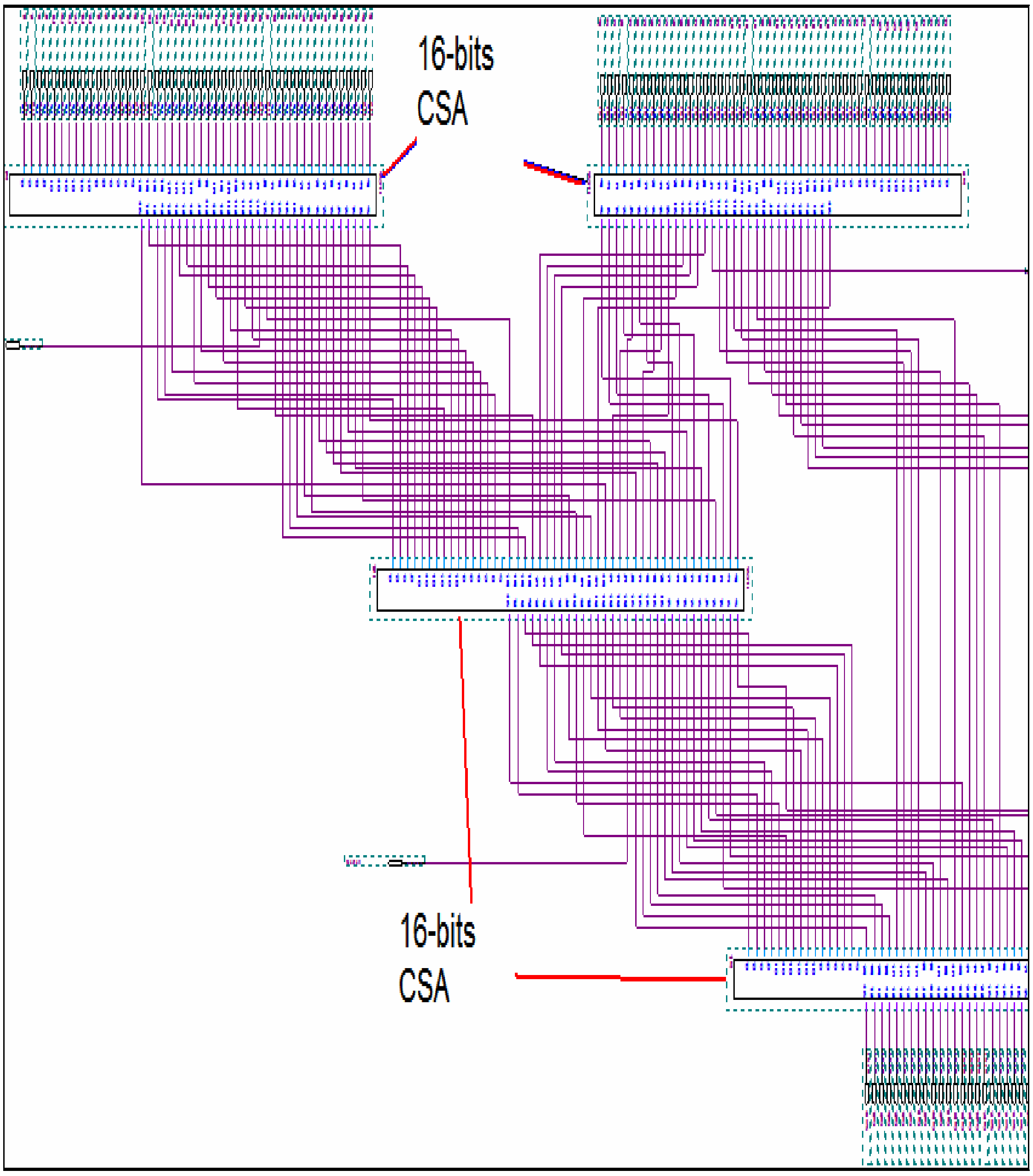


Figure 3.3: Six operands of 16-bits CSA

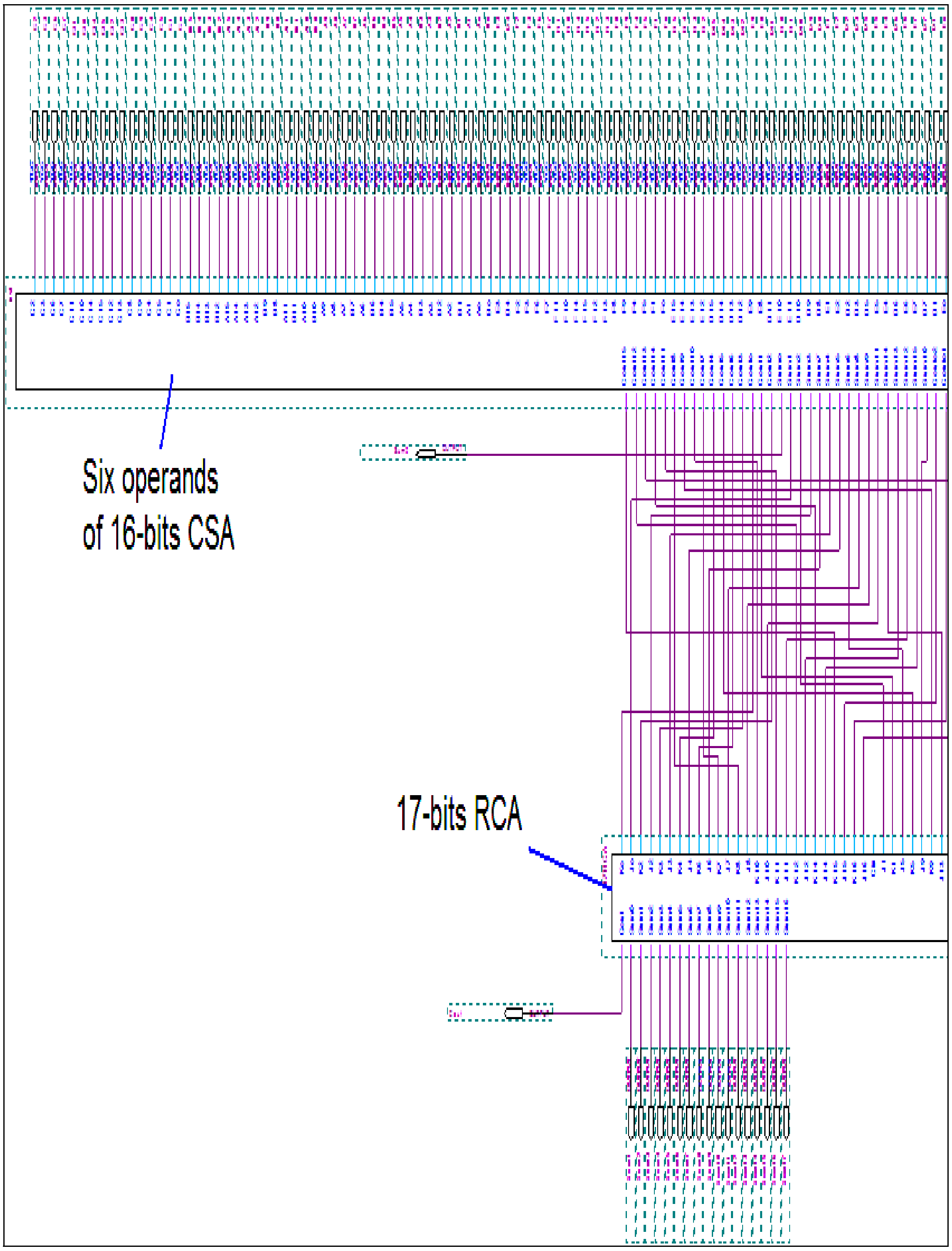


Figure 3.4: Three levels six operands of 16-bits CSA with RCA

Table 3.1: Truth table of 16-bits CSA

INPUT			OUTPUT	
A	**B**	**C**	**Sum**	**Cout**
0000	335A	8537	8667	0118
0002	335C	8539	8661	011E
0004	335E	853B	865B	0124
0006	3360	853D	8625	0142
0008	3362	853F	862F	0140
000A	3364	8541	8629	0146
000C	3366	8543	8623	014C
000E	3368	8545	862D	014A
0010	336A	8547	8637	0148
0012	336C	8549	8631	014E
0014	336E	854B	862B	0154
0016	3370	854D	8635	0152
0018	3372	854F	863F	0150
001A	3374	8551	8639	0156
A438	7BF3	42F6	9D3D	62F2
A43A	7BF5	42F8	9D37	62F8
A43C	7BF7	42FA	9D31	62FE
A43E	7BF9	42FC	9D3B	62FC
A440	7BFB	42FE	9D45	62FA
A442	7BFD	4300	9CBF	6340
A444	7BFF	4302	9CB9	6346
A446	7C01	4304	9B43	6404
A448	7C03	4306	9D4D	6402
A44A	7C05	4308	9B47	6408
A44C	7C07	430A	9B41	6408
A44E	7C09	430C	9B4B	640C
A450	7C0A	430E	9B55	640A
A452	7C0C	4310	9B4F	6410
A454	7C0E	4312	9B49	6416

Every single stage of 16-bits CSA output that has been through in six operands of 16-bits CSA compared by adding the same input at each stage at 16-bits CSA circuit design. Until the waveform has been confirmed compatible each other, the six operands of 16-bits CSA need to be constructed. The connection must be properly connected to make sure that the output is accurate.

The precise output data has been taken and recorded. The output then either compared to the truth table in the Table 3.1 or being calculated using equation 2.0 in previous chapter. This way is better to prevent any mistakes. The output waveform being analyzed based on truth table for Table 3.1. Figure 3.5 shows the flow chart of level 1 of analysis level. The design then proceeds to the three level six operands of 16-bits CSA that can be seen in the Figure 3.4 and the output waveform data has been taken. This is the final stage of circuit design.

To verify the speed of this circuit, the D flip-flop has been added at the input and output of three level 16-bits CSA with RCA circuit design. This design is called as non-pipelined design and can be seen in the Figure 3.7. The speed only can be measure with the existence of clock. This method needed to be used to generate the Timing Analyzer Tool.

The next step was designed the modified of six operands 16-bit CSA circuit by put pipeline at each stage. Pipeline applicant as speed up the circuit designs using D flip-flop. The advantage of this circuit over the D-type flip-flop is that it "captures" the signal at the moment the clock goes high, and subsequent changes of the data line do not matter, even if the signal line has not yet gone low again. The D flip-flop can be interpreted as a primitive delay line or zero-order hold, since the data is posted at the output one clock cycle after it arrives at the input.The goal is to synchronize the waveform and get better waveform clearly in terms of decreased the delay.

The modified design began with the bigger circuit which are six operands 16-bits CSA and three levels six operands 16-bits CSA with RCA. The pipeline put at the six operands of 16-bit CSA circuit design. Figure 3.8 shows the circuit design.

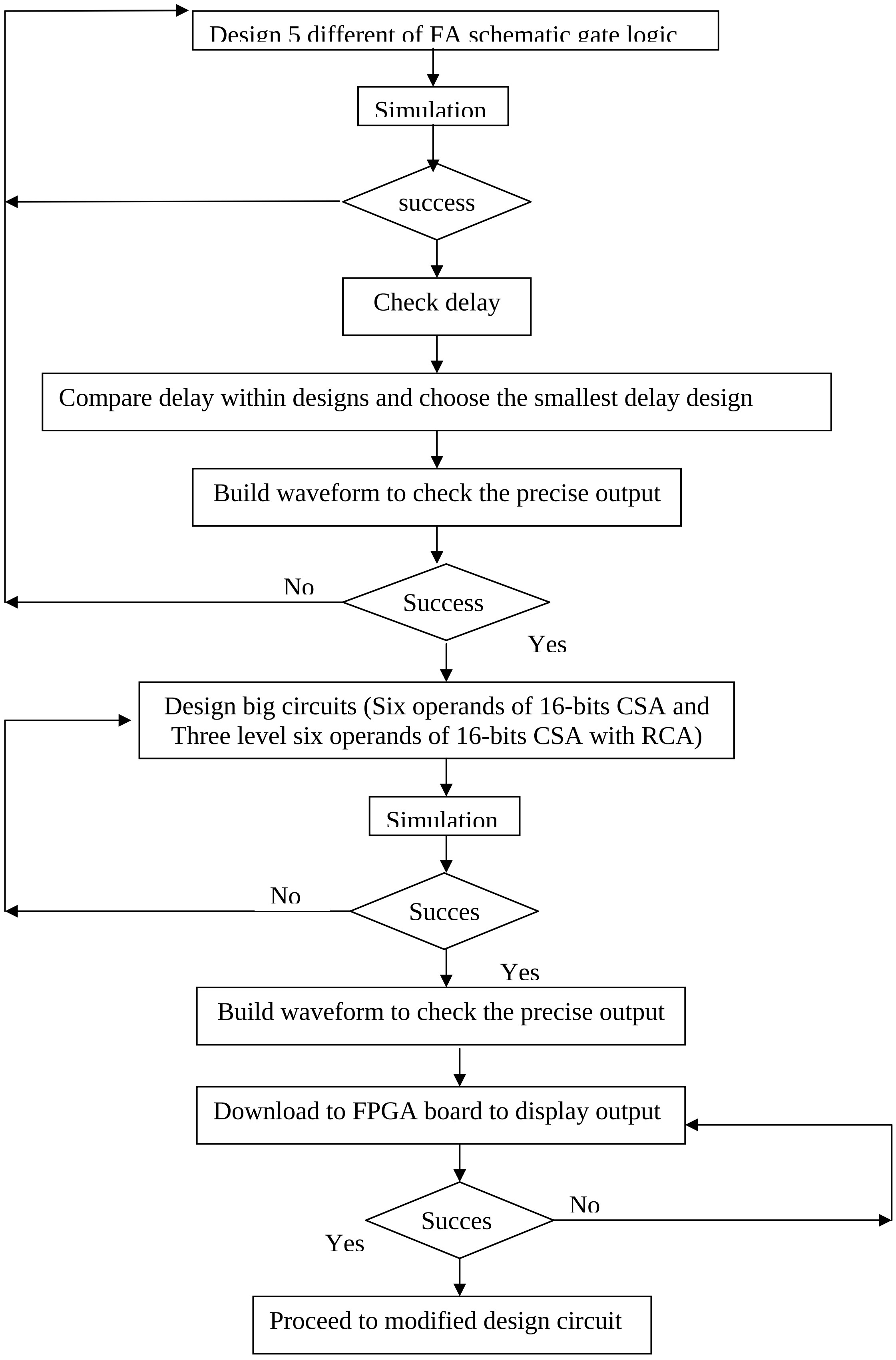
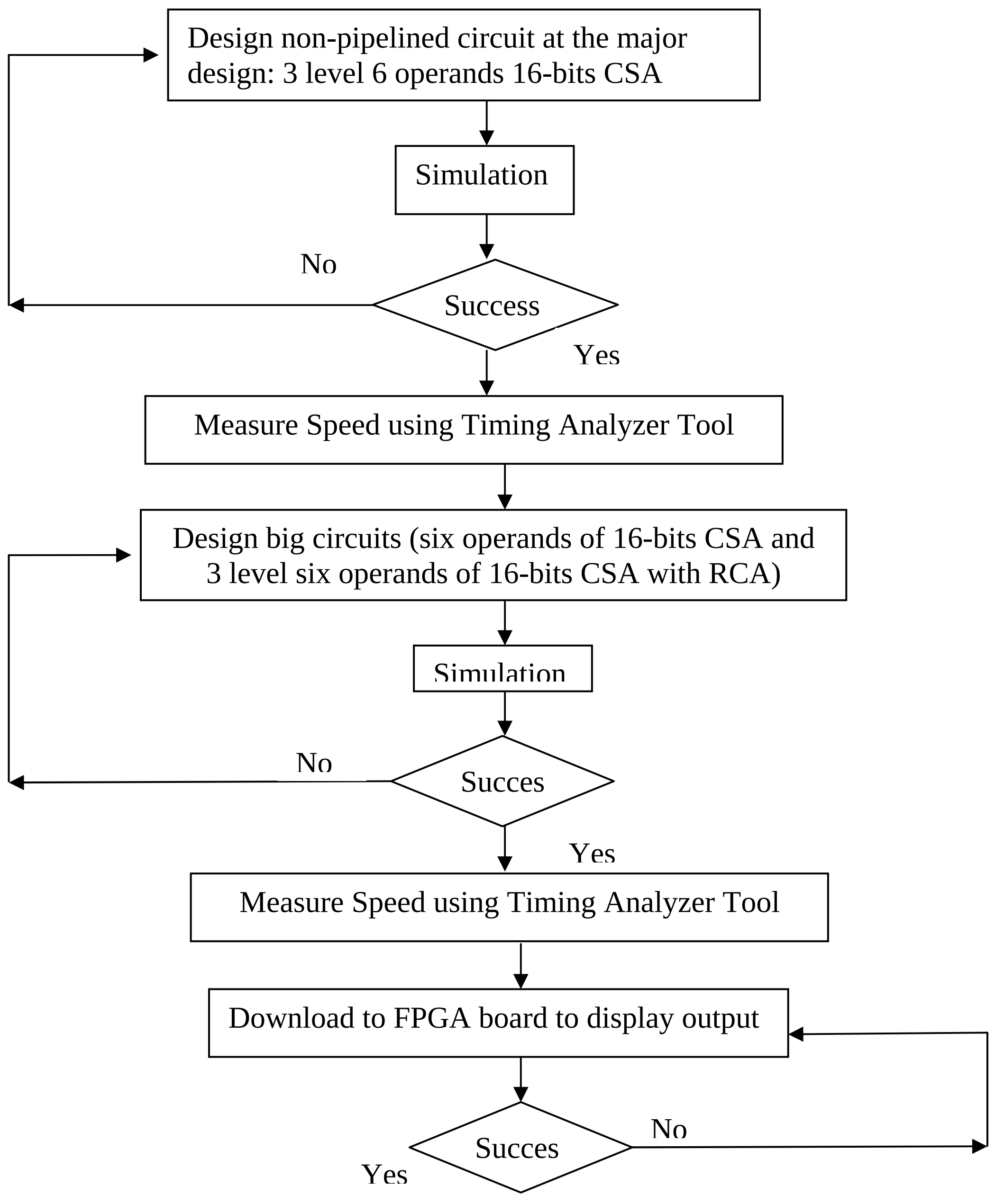

Design 5 different of FA schematic gate logic
Simulation
success
Check delay
Compare delay within designs and choose the smallest delay design
Build waveform to check the precise output
No
Success
Yes
Design big circuits (Six operands of 16-bits CSA and Three level six operands of 16-bits CSA with RCA)
Simulation
No
Succes
Yes
Build waveform to check the precise output
Download to FPGA board to display output
No
Succes
Yes
Proceed to modified design circuit

Figure 3.5: Flow Chart of level 1 analysis stage

Then, the modified six operands 16-bits CSA block symbol has been used to proceed into the three levels six operands 16-bits CSA circuit. The pipeline do not inserted anymore at this circuit design. This is because if the pipeline were put at RCA design circuit, the speed was not increased yet. There is no used of adding pipeline in order to speed up the design. The design can be seen in the Figure 3.9.



Design non-pipelined circuit at the major design: 3 level 6 operands 16-bits CSA
Simulation
No
Success
Yes
Measure Speed using Timing Analyzer Tool
Design big circuits (six operands of 16-bits CSA and 3 level six operands of 16-bits CSA with RCA)
Simulation
No
Succes
Yes
Measure Speed using Timing Analyzer Tool
Download to FPGA board to display output
No
Succes
Yes

Figure 3.6: Flow Chart of level 2 analysis stage

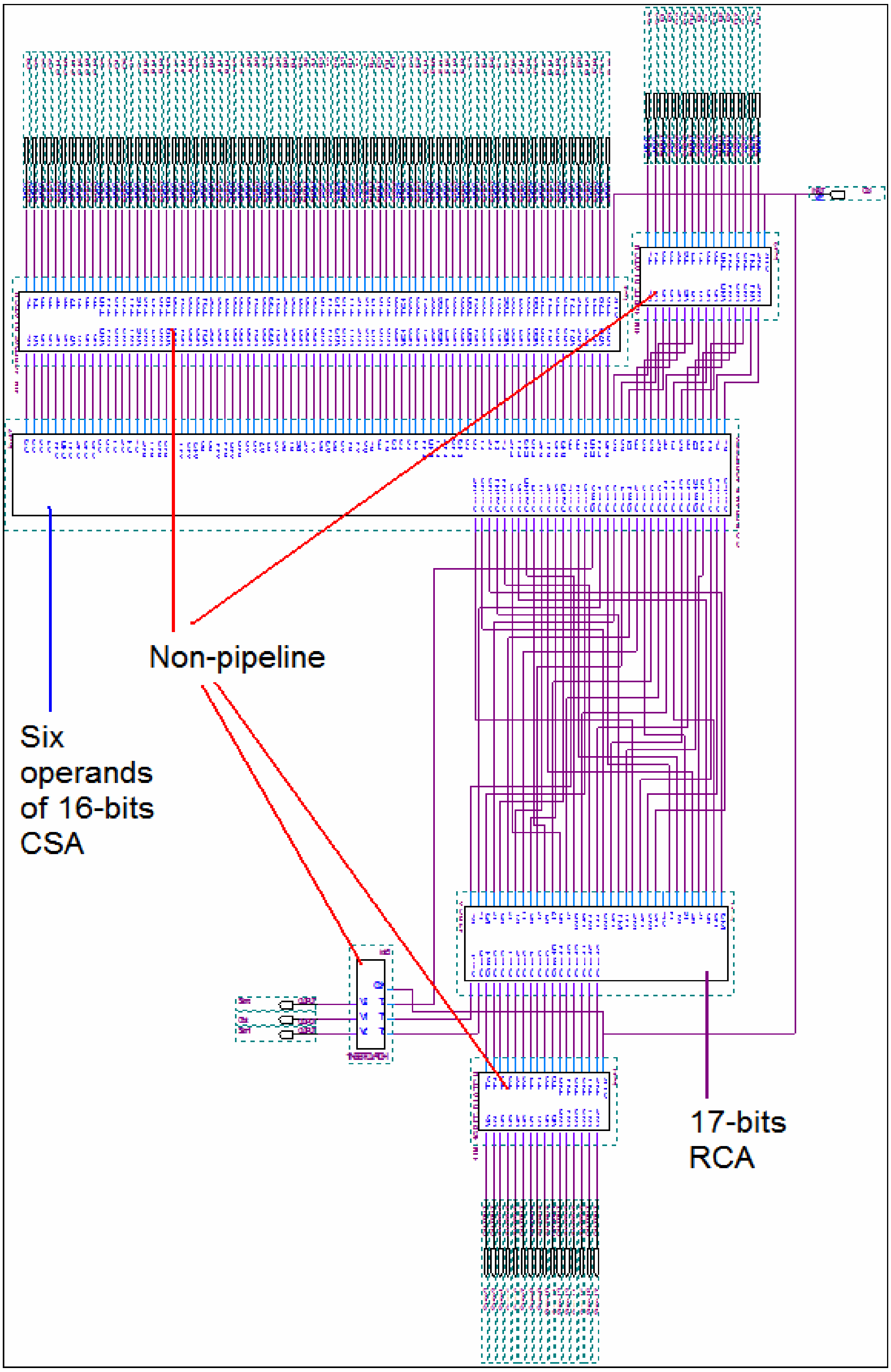


Figure 3.7: The non-pipeline three level six operands 16-bits CSA with RCA logic design

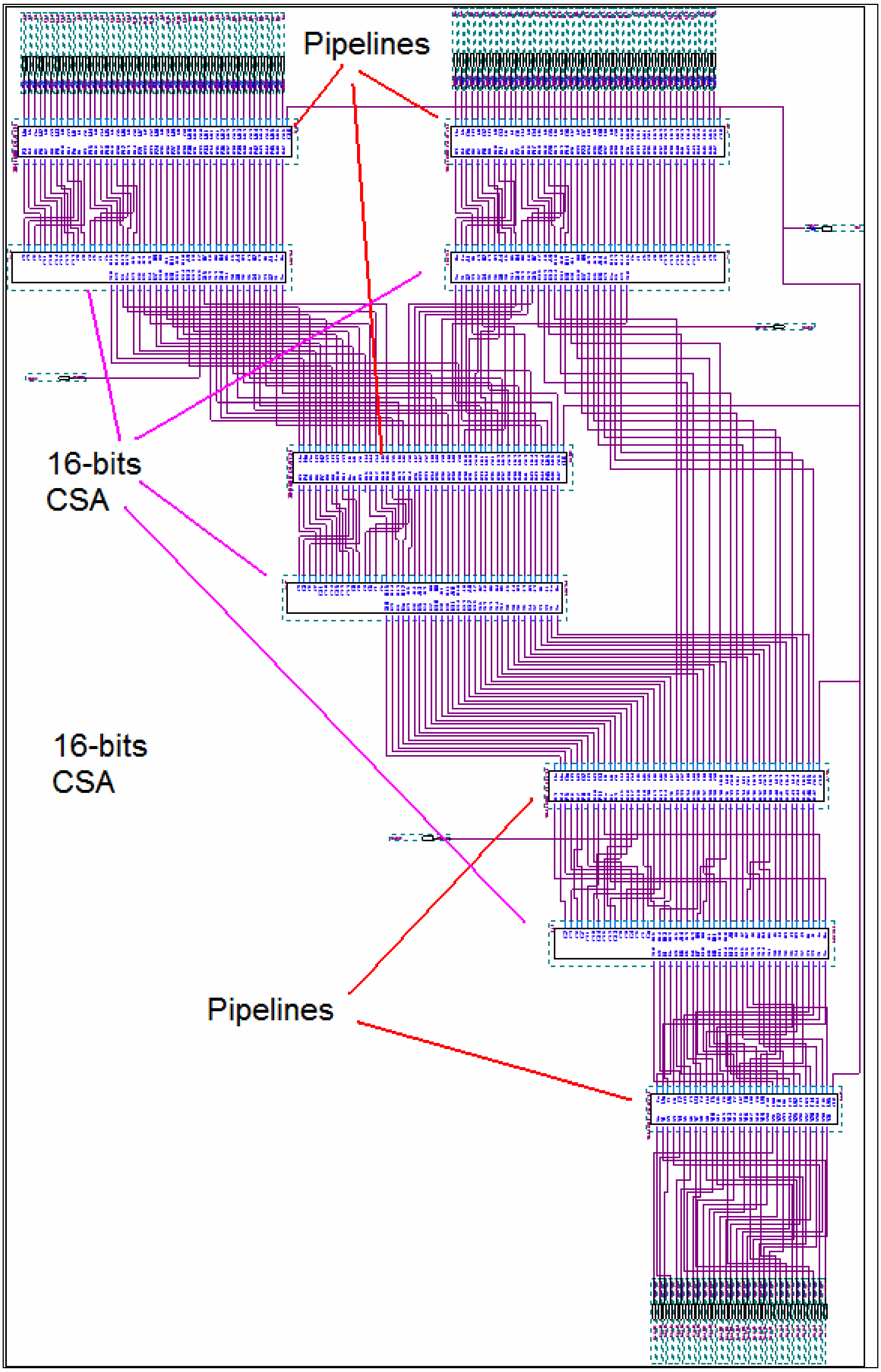


Figure 3.8: The modified six operands 16-bits CSA logic design

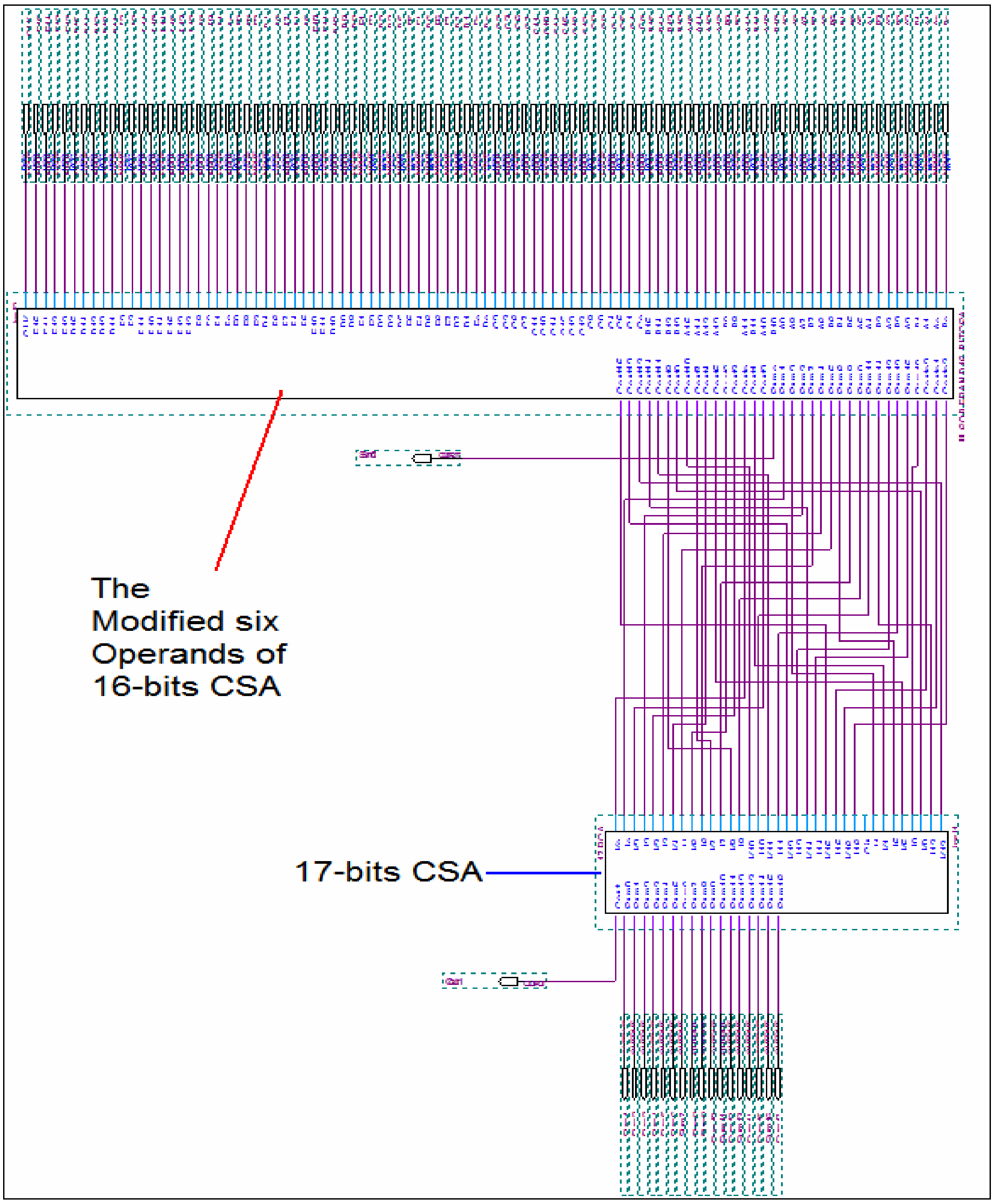


Figure 3.9: The modified three level six operands of 16-bit CSA logic design

After the compilation process, the changes of speed can be seen using Timing Analyzer Tool. It is used to measure the speed of these circuits. The frequency value can be defined as the speed of the circuits.

The output waveform of modified six operands 16-bits CSA built and the output taken after the simulation process has been done. The result then compared to the six operands 16-bits CSA circuit waveform. If the result is not compatible each other, this circuit needed to reconstruct until its output is same as compared to the waveform output of six operands 16-bits CSA. The flow chart of level 2 analysis process can be seen in the figure 3.6.

3.3.2 Hardware

The project continued with the hardware development level. After those designs are performed in the software development, the progress then proceeds to the hardware development level. The successful design circuit then downloaded into the Altera UP2 FPGA board. The UP2 Education Board is a stand-alone experiment board based on a FLEX® 10K device and includes a MAX® 7000 device. When used with the Quartus II software, the board provides a superior platform for learning digital logic design using industry-standard development tools and PLDs. The board is designed to meet the needs of instructors and students in a laboratory environment. The UP2 Education Board supports both look-up table (LUT) -based and product term-based architectures. The EPF10K70 device can be configured in-system with either the ByteBlaster II download cable or an EPC1 configuration device. Additional download cables can be purchased separately. The EPM7128S device can be programmed in-system with the ByteBlaster II download cable.

The specific device setting in the software design is EPF10K70RC240-4. On the UP2 board, the EPF10K70 device is based on SRAM technology. It is available in a 240-pin RQFP package and has 3,744 logic elements (LEs) and nine embedded array blocks (EABs). Each LE consists of a four-input LUT, a programmable flipflop, and dedicated signal paths for carry-and-cascade functions. Each EAB provides 2,048 bits of memory that can be used to create RAM, ROM, or first-in first-out (FIFO) functions. EABs can also implement logic functions, such as multipliers, microcontrollers, state machines, and digital signal processing (DSP) functions. With 70,000 typical gates, the EPF10K70 device is ideal for intermediate to advanced digital design courses, including computer architecture, communications, and DSP applications. The UP2 board is shown in figure 3.10, contains the features describe above.

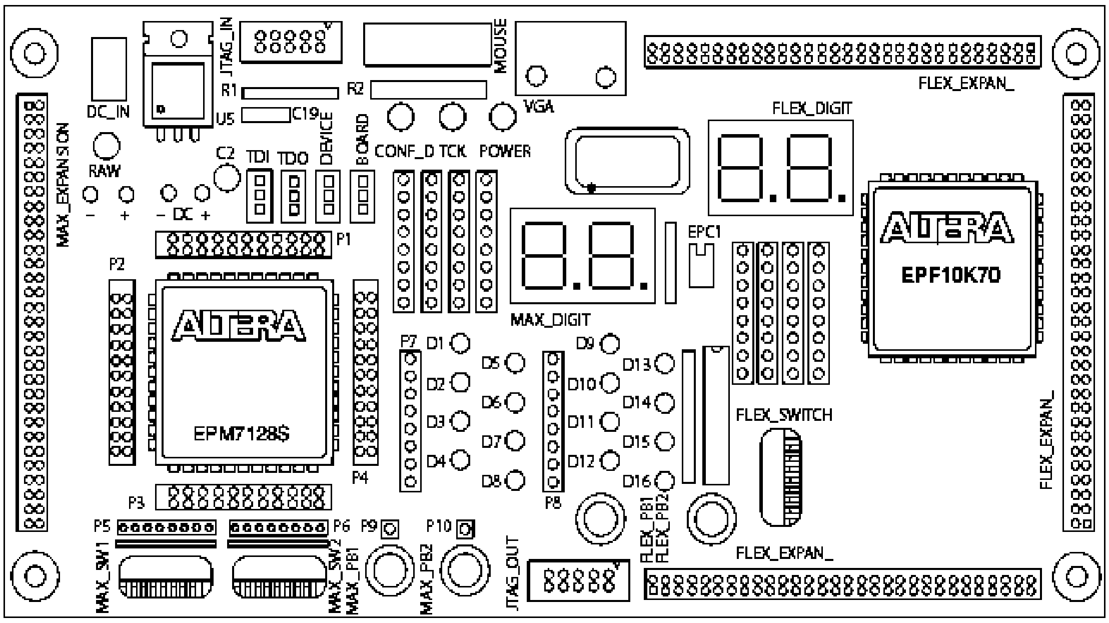


Figure 3.10: Altera UP-2 Training Board Block Diagram

The outputs of the design connected to the binary code decimal encoder [43], in the software first. Since the design project is 16-bits carry save adder, the devices needed to be use is not enough and needed to be added. There are 48 inputs and 32 outputs. The components on the UP2 FPGA board do not being used to prevent any mistakes since lots devices have been added. Thus, seven deep switch connected to the inputs of design and eight seven segments connected to the outputs of design during the uploading. These connections setting referred to the given Altera University program UP2 education kit user guide [32].

The downloading process started with the assigned pin or location chip at the Quartus II software. These pins saved and compiled. Altera UP2 board connected to the software and switched on. Hardware has been setup at the ByteBlasterMVBoard. LPT1, parallel port has been chosen automatically. Multi device JTAG chain mode selected and the hardware detected automatically or manually.

3.3.2.1 Seven Segment Display

To display the result of the design, the seven-segment was added at the output of the Altera UP2 board. Therefore, the seven-segment pin descriptions have been studied [41] [42] (refer to Appendix D). For this circuit, common anode seven-segment was used. Based on Figure 3.11, the seven-segment display pin was verified to outcome the truth table of seven-segment logic as shown in the Table 3.2. Figure 3.13 shows the hardware designed circuit test.

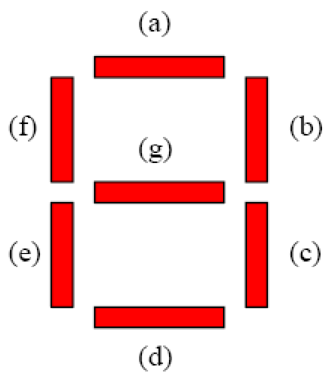


Figure 3.11: Seven segment display pins

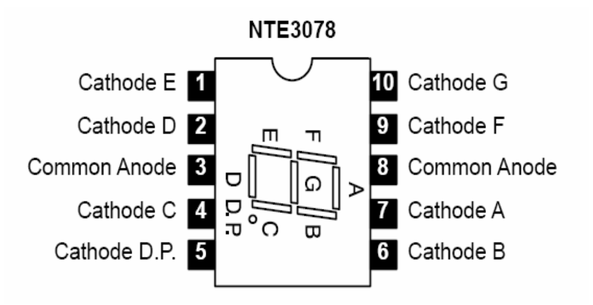


Figure 3.12: Pin connection diagram

Table 3.2: Truth table of seven segment logic

Hexadecimal Digit	INPUTS				OUTPUTS						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
a	1	0	1	0	1	1	1	1	1	0	1
b	1	0	1	1	0	0	1	1	1	1	1
C	1	1	0	0	1	0	0	1	1	1	0
d	1	1	0	1	0	1	1	1	1	0	1
E	1	1	1	0	1	1	0	1	1	1	1
F	1	1	1	1	1	0	0	0	1	1	1

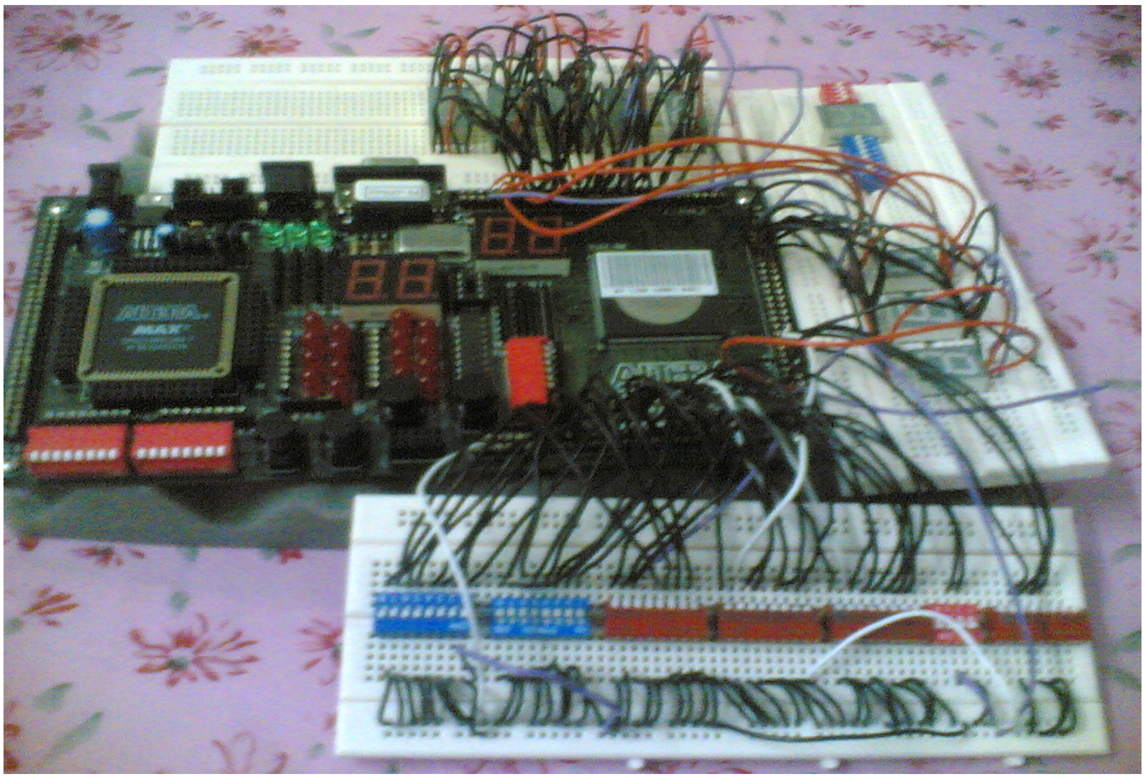


Figure 3.13: Hardware Design Circuit Test