CHAPTER 4

RESULTS AND DISCUSSION

As results, after completed the analysis of the Wallace Tree multiplier, it has been proven that the pipelining method could increased the speed of the multiplier. However, the area of the multiplier designed is also increased because of the 4-stages D flip-flop that has been implemented in the design. Basically, the objective of this project is mainly focus on the speed of the Wallace Tree multiplier, so, the increased area of the multiplier does not take into count. Otherwise, the high speed 8-bits x 8-bits Wallace Tree multiplier design in this project was to be implementing on the field-programmable gate array (FPGA) Altera UP2 board, not to be fabricating on a chip. Figure 4.1 shows part of the source code for the high speed 8-bits x 8-bits Wallace Tree multiplier with pipelining method.

The completed source code for both designs, conventional and pipelining high-speed 8-bits x 8-bits Wallace Tree multiplier included the sub-circuits are shown in the Appendices.

```
// Define an 8-bits x 8-bits Wallace Tree multiplier
module wallace_t_pl(z, clk, set, rst, x, y);

// I/O port declarations
output [16:0] z;
input [7:0]  x, y;
input clk, set, rst;

// Internal nets
reg [15:1] sh, ch;
reg [49:1] sf, cf;
reg [16:1] w;

// Instantiate an 8-bits x 8-bits Wallace Tree multiplier

d_ff d1 (w[1], clk, x[0], set, rst);
d_ff d2 (w[2], clk, x[1], set, rst);
d_ff d3 (w[3], clk, x[2], set, rst);
d_ff d4 (w[4], clk, x[3], set, rst);
d_ff d5 (w[5], clk, x[4], set, rst);
d_ff d6 (w[6], clk, x[5], set, rst);
d_ff d7 (w[7], clk, x[6], set, rst);
d_ff d8 (w[8], clk, x[7], set, rst);

d_ff d9 (w[9], clk, y[0], set, rst);
d_ff d10 (w[10], clk, y[1], set, rst);
d_ff d11 (w[11], clk, y[2], set, rst);
d_ff d12 (w[12], clk, y[3], set, rst);
d_ff d13 (w[13], clk, y[4], set, rst);
d_ff d14 (w[14], clk, y[5], set, rst);
d_ff d15 (w[15], clk, y[6], set, rst);
d_ff d16 (w[16], clk, y[7], set, rst);

halfadd ha1 (sh[1], ch[1], (w[1] & w[10]), (w[2] & w[9]));
fulladd fa1 (sf[1], cf[1], (w[1] & w[11]), (w[2] & w[10]), (w[3] & w[9]));
fulladd fa2 (sf[2], cf[2], (w[2] & w[11]), (w[3] & w[10]), (w[4] & w[9]));
fulladd fa3 (sf[3], cf[3], (w[3] & w[11]), (w[4] & w[10]), (w[5] & w[9]));
fulladd fa4 (sf[4], cf[4], (w[4] & w[11]), (w[5] & w[10]), (w[6] & w[9]));
fulladd fa5 (sf[5], cf[5], (w[5] & w[11]), (w[6] & w[10]), (w[7] & w[9]));
fulladd fa6 (sf[6], cf[6], (w[6] & w[11]), (w[7] & w[10]), (w[8] & w[9]));
.       .       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .       .
endmodule
```

Figure 4.1: Part of the source code for the high speed 8-bits x 8-bits Wallace Tree multiplier with pipelining method

The conventional circuit produces the maximum speed of 14.99 MHz or maximum delay of 66.7 nanoseconds to complete one process of 8-bits x 8-bits multiplication. However, after upgrading the conventional Wallace Tree into pipeline Wallace Tree, by adding D flip-flop stages at every input of the half adder and the full adder, the speed has improve much, increased to 54.05 MHz and the maximum delay has decreased to 18.3 nanoseconds. It was three times faster than the conventional design.

Table 4.1 show the multiplicands, multipliers and the results that were used to test and prove the functional of both design, the conventional and pipelining high speed 8-bits x 8-bits Wallace Tree multiplier.

Table 4.1: The test bench for the conventional and pipelining high speed 8-bits x 8-bits Wallace Tree multiplier

Multiplicand (Hexadecimal)	**Multiplier (Hexadecimal)**	**Multiplication Result (Hexadecimal)**
0F	FA	00EA6
F0	FF	0EF10
5A	F0	05460
00	F0	00000

Figure 4.2 show the result of the multiplication process done by the high speed 8-bits x 8-bits Wallace Tree multiplier with pipelining that has been design. Figure 4.3 show the result of Timing Analyzer Tool for both of the Wallace Tree multiplier design.

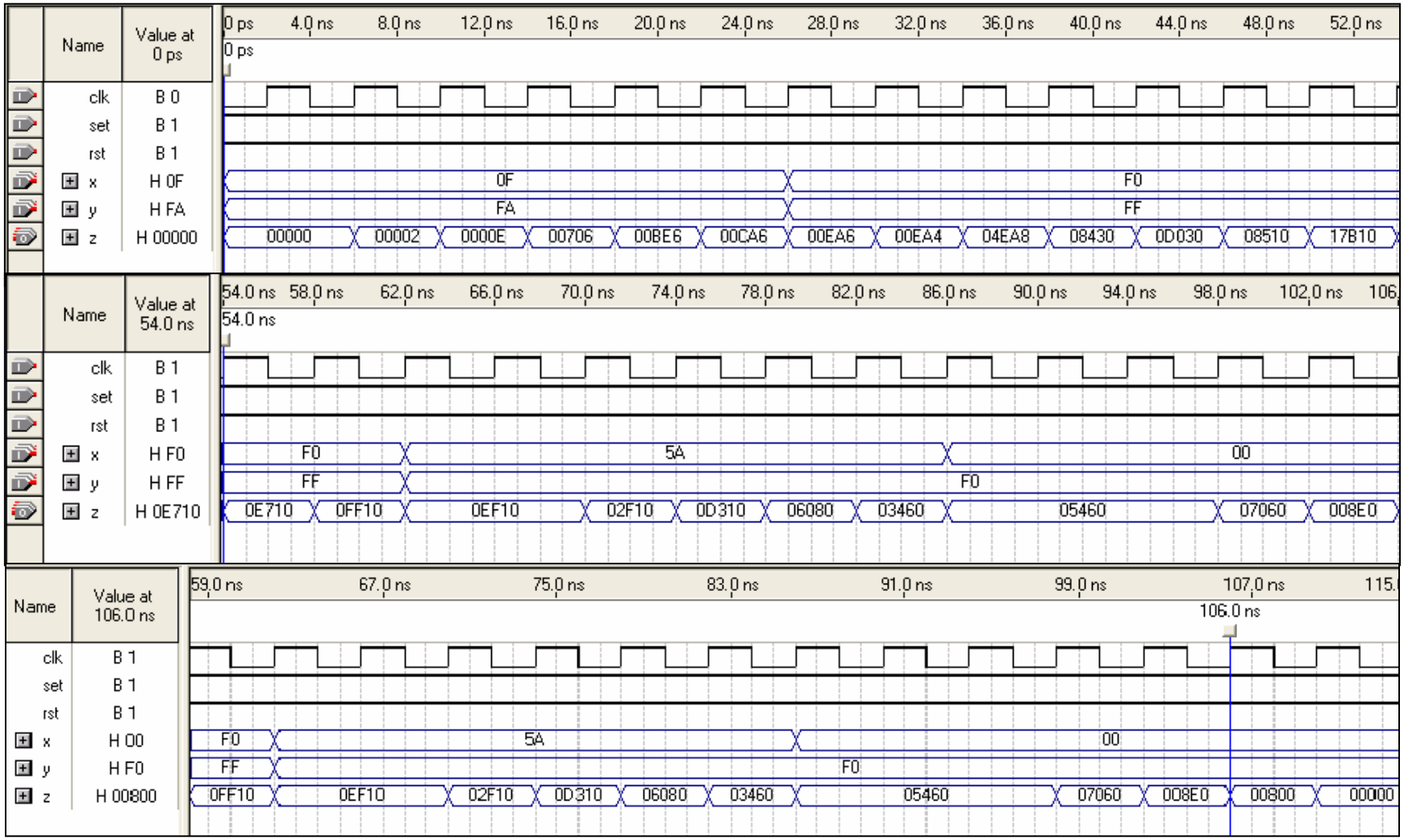
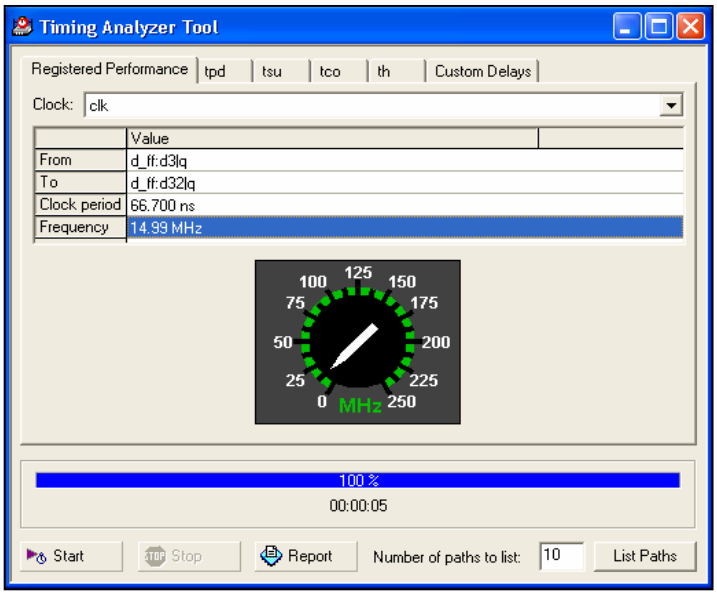
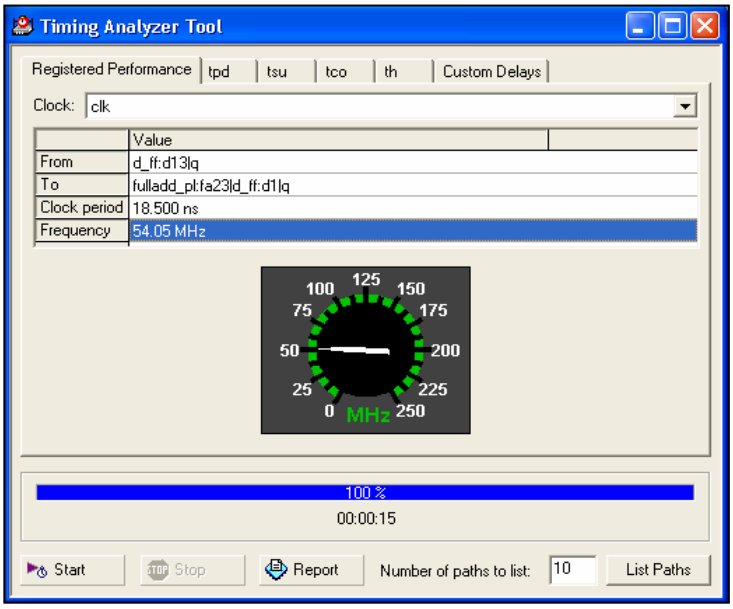


Figure 4.2: Result of the multiplication process by the high speed 8-bits x 8-bits Wallace Tree multiplier with pipelining



The conventional high speed 8-bits x 8-bits Wallace Tree multiplier design



The pipelining high speed 8-bits x 8-bits Wallace Tree multiplier design

Figure 4.3: The results of Timing Analyzer Tool

Finally, after completed the analysis of the Wallace Tree multiplier, it has been proven that the pipelining method could increased the speed of the multiplier.