




## 2023 Digital IC Design Homework 3

NAME	林柏戎		
Student ID	Q36114239		
Simulation Result			
Functional simulation	100	Gate-level simulation	100
<pre># Pattern 55 : a^a+a+b-(a^b)= # Expected answer: 11   get: 11 --&gt; Pass # Pattern 56 : f^5-5^5+(1-1)= # Expected answer: 50   get: 50 --&gt; Pass # Pattern 57 : (1+2)= # Expected answer: 3   get: 3 --&gt; Pass # Pattern 58 : (1+2)+(2+3)= # Expected answer: 9   get: 9 --&gt; Pass # Pattern 59 : (1-1)^(c-a)+3-1= # Expected answer: 2   get: 2 --&gt; Pass # Pattern 60 : ((2-1)^(c-a))= # Expected answer: 2   get: 2 --&gt; Pass</pre>  <p>Congratulations!!! You past all patterns! Your score is 100. Total use 1586 cycles to complete simulation.</p> <pre>** Note: \$finish : C:/Users/MediaCore/Desktop/HW3/file/testfixture.sv(191) Time: 25376 ns Iteration: 1 Instance: /testfixture</pre>		<pre># Pattern 56 : f^5-5^5+(1-1)= # Expected answer: 50   get: 50 --&gt; Pass # Pattern 57 : (1+2)= # Expected answer: 3   get: 3 --&gt; Pass # Pattern 58 : (1+2)+(2+3)= # Expected answer: 9   get: 9 --&gt; Pass # Pattern 59 : (1-1)^(c-a)+3-1= # Expected answer: 2   get: 2 --&gt; Pass # Pattern 60 : ((2-1)^(c-a))= # Expected answer: 2   get: 2 --&gt; Pass</pre>  <p>Congratulations!!! You past all patterns! Your score is 100. Total use 1586 cycles to complete simulation.</p> <pre>** Note: \$finish : C:/Users/MediaCore/Desktop/HW3/file/testfixture.sv(191) Time: 25376 ns Iteration: 1 Instance: /testfixture</pre>	
Synthesis Result			
Total logic elements	361		
Total memory bits	0		
Embedded multiplier 9-bit elements	1		
Total cycle used	1586		
Clock width	16		
Flow Summary			
 <<Filter>>			
Flow Status	Successful - Fri Apr 21 14:37:32 2023		
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition		
Revision Name	AEC		
Top-level Entity Name	AEC		
Family	Cyclone IV E		
Device	EP4CE55F23A7		
Timing Models	Final		
Total logic elements	361 / 55,856 ( < 1 % )		
Total registers	155		
Total pins	19 / 325 ( 6 % )		
Total virtual pins	0		
Total memory bits	0 / 2,396,160 ( 0 % )		
Embedded Multiplier 9-bit elements	1 / 308 ( < 1 % )		
Total PLLs	0 / 4 ( 0 % )		

Description of your design
<p>主要使用的演算法：Infix Method，一開始花蠻多時間在理解 Infix to Postfix 演算法，完成 Infix to Postfix 後，發現使用 Infix 可以用更少的 Total logic elements 且 Total cycle used 也更少。</p> <p>主要分四個狀態：DATA_IN、CAL、DONE、RESET。</p> <p>DATA_IN: 將題目給的 data 丟給 data_stack，在 CAL 時可利用 data_count 控制時序上的問題。</p> <p>CAL: 將 number 丟到 value_stack，運算子丟到 ops_stack，再作演算法，亦會判斷乘法與加減的層級關係。</p> <p>DONE: 將 valid 拉起來，result 即為 value_stack [0]。</p> <p>RESET: 將各參數作 reset，便進下一筆測資。</p>

*Scoring = Area cost \* Timing cost*

*Area cost = Total logic elements + Total memory bits + 9\*Embedded multipliers 9-bit elements*

*Timing cost = Total cycle used \* Clock width*

**\* Total logic elements must not exceed 1500.**