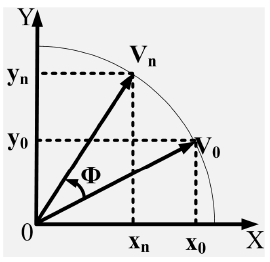
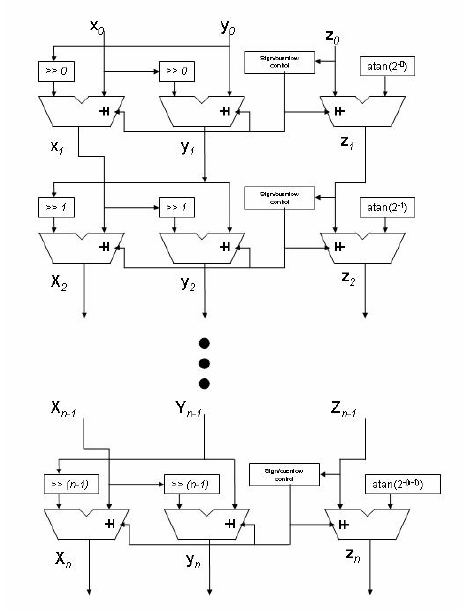
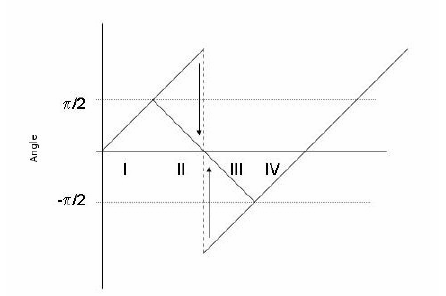
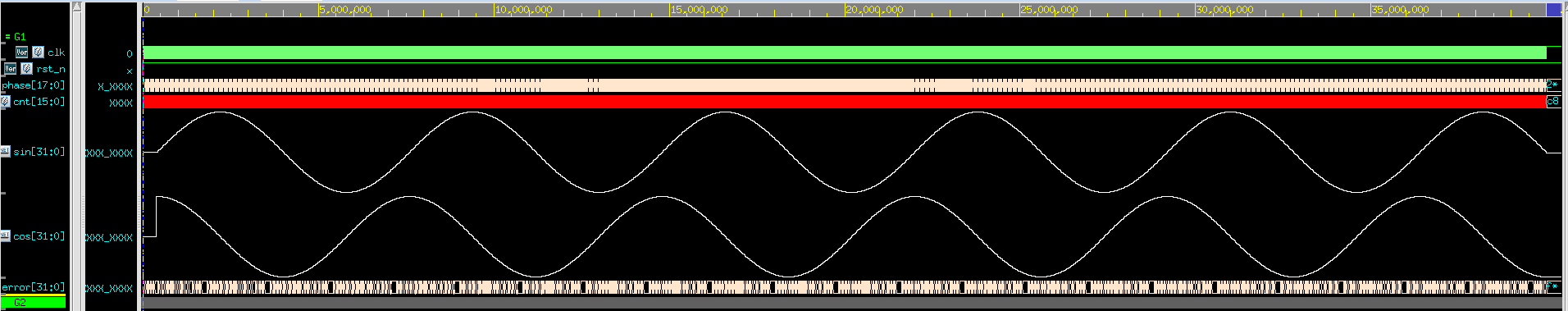
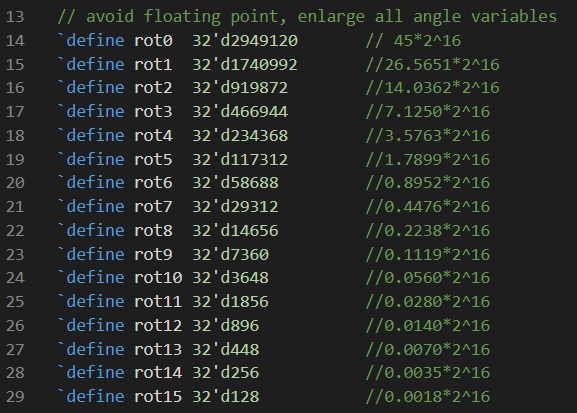
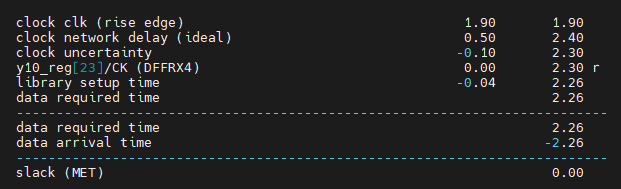
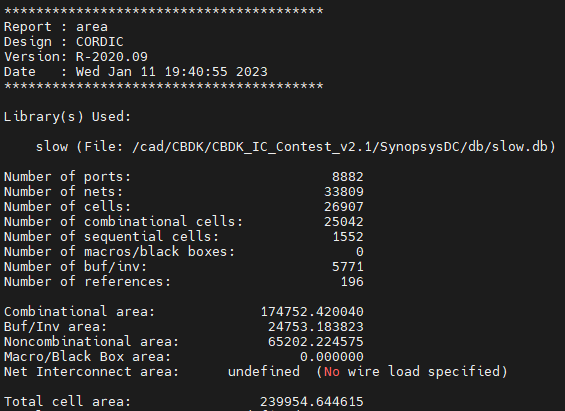
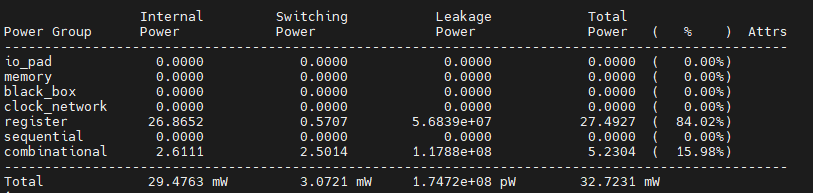
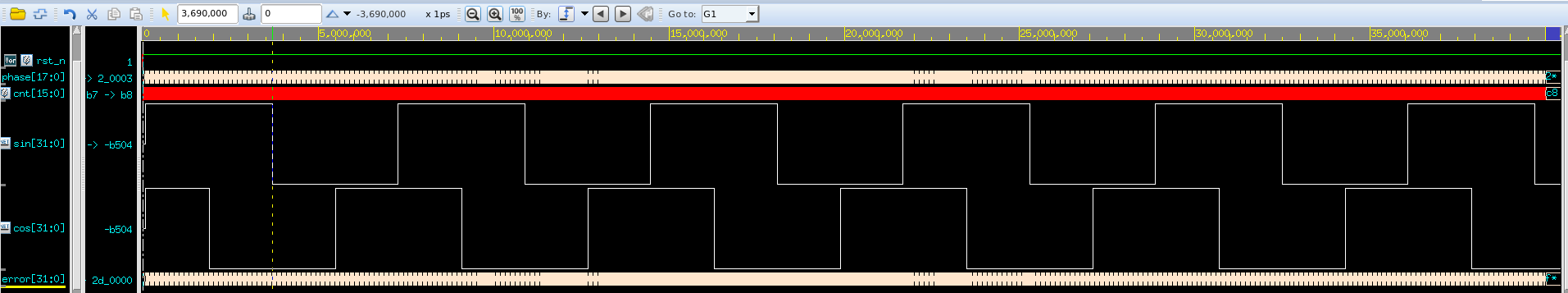
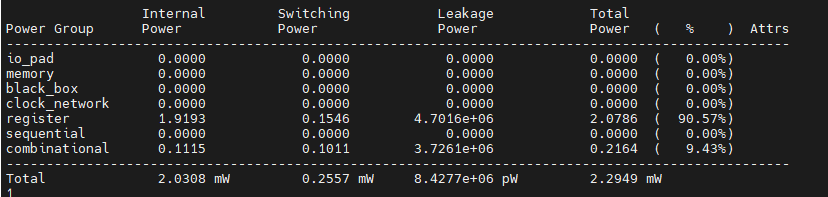
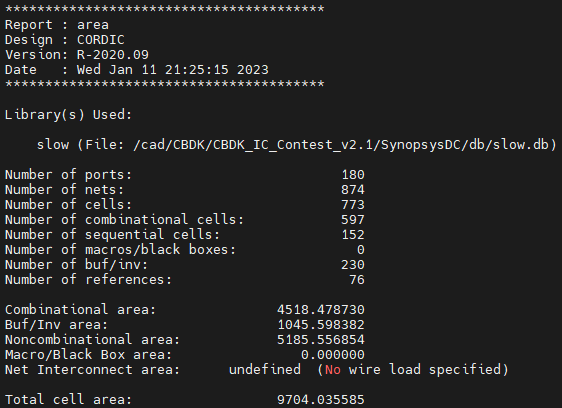
**DSP VLSI Architecture Design**

Final Project

Sine-Cosine Computation Using CORDIC Algorithm

學號：110521168

姓名：陳旻盛  
授課老師：蔡宗漢 教授

1. **CORDIC Algorithm原理** Coordinate Rotation Digital Computer，又稱CORDIC，是一種座標旋轉的計算方法，於1959年由J.D.Volder提出，主要用於三角函數、雙曲線、指數、對數的運算。此算法通過基本的加法和移位運算代替乘法運算，使得向量的旋轉和定向計算不再需要三角函數、乘法、開平方、反三角等函數。  
   以右圖的二維平面為例，當點V0逆時   
   鐘旋轉到Vn時，可以根據旋轉矩陣的運算得知：  
    xn = x0cosθ – y0sinθ = cosθ(x0 – y0tanθ)  
    yn = y0cosθ + x0sinθ = cosθ(y0 + x0tanθ)  
    此時，我們將旋轉角θ細分為θi，且滿足tanθi = 2-I，i為非負整數。由於每次的旋轉角度θi是固定不變的，如果一直朝著同一方向旋轉一定會超過θ，所以另兩方向d = 1代表逆時針，d = -1代表順時針，可以得到每次旋轉的角度為dθi，剩餘的角度為zi+1，則zi+1 = zi -dθi。  
   且因為θmax = Σtan-12-I = 99.7。  
    θmin = Σtan-12-I = -99.7。  
    所以我們需要確保在每一次計算的過程中，輸入向量都在第一、四象限。  
   在計算xn和yn的時候，我們可以發現透過不斷提出cosθ以及用2-I來取代tanθ，即可做到在運算過程中都只有加法以及移位操作。  
   可得結論：xn =   
    yn =   
   所以當我們給定初始值x0 = i, y0 = 0，代入上式可得xn = cos   
   yn = sin，即可實現sine和cosine的計算。
2. **架構設計**  
     
   此次以16級pipeline來設計架構，在每一次迭代的過程中，將迭代後更新的值放入下一級pipeline中，讓每一級pipeline stage都只要經過一個clock cycle就能完成運算，並且每個clockcycle都能得到輸出。  
     
   因為要確保維持在第一、第四象限，所以需要調整在第二、三象限時的角度。若位於第二象限則須順時針旋轉90度到第一象限，第三象限則需逆時針旋轉90度到第四象限。
3. **Verilog實作  
   **因為在testbench中，角度會一直在0~359循環，所以sin和cos的波型也一直循環下去。  
   且因迭代16次使得誤差非常小，所以整個訊號曲線會非常平滑。  
   在過程中也透過將角度的變量放大，避免了浮點數的運算，用以滿足精度的要求：  
   ****  
     
   report\_timing  
   ****  
     
     
     
   report\_area  
   ****  
     
   report\_power  
   ****  
     
   可以看到因為迭代了16，所以需要很多額外的硬體來儲存變數，導致面積以及功耗較大，但是可以得到非常精準的三角函數運算值。  
   若大幅減少迭代次數到1次，可以節省大量的面積以及功耗，但由於迭代次數少，角度的修正十分有限，所以得到的誤差值會非常大，以下為只跌代1次的結果：  
   ****  
   ****  
   可以看到雖然面積和功耗小了很多，但因為只迭代一次，角度只能有正負45度的修正，所以得到的值只會有兩種，sin和cos的波型就會變成方波，使其誤差非常大。
4. **參考資料**
5. [1] E. O. Garcia, R. Cumplido, and M. Arias, "Pipelined CORDIC design on FPGA for a digital sine and cosine waves generator," in *2006 3rd international Conference on Electrical and Electronics Engineering*, 2006: IEEE, pp. 1-4.