|  |
| --- |
| 數位積體電路  Lab2  智能系統所 312581006 張宸瑋 |
| **2-1: 1-bit Full Adder standard cell** |
|  |
| 波形圖(測資: pattern\_adder\_1\_bits.vec |
|  |
| Power |
|  |
| Worst case delay: 28.6p |
| 透過worst case delay的分析，我們可以找出critical path為A或B到S的路徑 |

|  |
| --- |
| **2-2: Different logic family of 1-bit Full Adders** |
|  |
| 波形圖 |

|  |  |
| --- | --- |
| Power | |
| CMOS power |  |
| CPL-type power |  |
| DCVS-type power |  |
| PPA measurment | |
| CMOS |  |
| CPL-type |  |
| DCVS-type |  |

|  |
| --- |
| 分析: |
| 在這裡我們可以看到CMOS的電晶體數量理論上會是比較多的，而實際測試結果也是如此，而消耗功率也較少，而CPL的電晶體數量也是三者最少的，但功率消耗有點多，推測可能是因為某些情況消耗一下恆定電流，而DCVS的電晶體數量居中，消耗的功率也是。 |

|  |
| --- |
| **2-3: Design a 4-Bit Adder** |
|  |
| 一開始先透過01\_run中的指令，來驗證behavior level，成功後會跑出如上的結果圖 |
|  |
| 確定功能沒問題之後，透過dc\_shell-t -f syn.tcl | tee syn.log，這句command做合成，轉換成gate level |
|  |
| 接著進行gate level的驗證 |
|  |
|  |
| 接著透過nWave，再次確認功能無誤 |
|  |
| critical path是b[0]的rising 到Output[4]的rising，總共花費83.16ps。 |
| **2-4: CMOS Logics for 4-Bit Adder** |
|  |
| verilog gate level netlist |
|  |
| HSPICE netlist |
| 透過助教提供的檔案，將verilog gate level netlist 轉換成HSPICE netlist (這裡用助教提供的.py檔案好像不行，跟用的cdl腳位對不太起來，我是後來自己手動改才成功的。 |
|  |
| 透過waveview查看波行書出是否正確 |
|  |
| Power |

|  |
| --- |
| Worst case delay |
| 當a=0010、b=0100 -> a=1100、b=0111 會有Worst case delay: 98.43ps |
| 討論:  這裡透過窮舉所有可能pattern的方式，來找出Worst case delay，首先透過worst\_case\_delay.py檔案，生成所有可能的.vec檔案，以及對應的.sp檔案，並且全部編譯，透過python來將所有測量出來的delay值比較，找出最大值。  分析:  這裡因為我用的是窮舉法，所以生成的資料非常多，也非常耗時，但我相信一定有可以減少pattern 生成的方式，希望之後有空可以回來思考一下。 |