**國立清華大學**

**超大型積體電路測試**

**VLSI Testing**



**Homework 2**

**系所級:電子所一年級、電機系碩士班一年級**

**學號:111063548、111061624**

**姓名:蕭方凱、尤弘瑋**

**指導老師:黃錫瑜教授**

目錄

[(a) Write the RTL code in Verilog or VHDL that takes in two 8-bit positive integers, A[7:0] and B[7:0], and produces its quotient Q[7:0] and remainder R[7:0]. 4](#_Toc134410020)

[divider.v 4](#_Toc134410021)

[Finite State Machine 6](#_Toc134410022)

[(b) Verify the correctness of your RTL code by a testbench. You should try it out by at least 3 pairs of input numbers. 7](#_Toc134410023)

[PATTERN.v 7](#_Toc134410024)

[TESTBED.v 12](#_Toc134410025)

[Display on MobaXterm 13](#_Toc134410026)

[nWave 13](#_Toc134410027)

[(c) Use a synthesis script to convert your RTL code into a gate-level netlist. Report the final gate count, the maximum operating speed (in MHz) and the estimated power dissipation in (mW) using Design Compiler. 14](#_Toc134410028)

[Area Report 14](#_Toc134410029)

[Slack Report 15](#_Toc134410030)

[Power dissipation 16](#_Toc134410031)

[(d) Add the scan chain into your gate-level netlist obtained by part(c), report the resulting gate count, the maximum operating speed (in MHz) of your circuit. Compare to the non-scan version, and report the area overhead percentage and performance penalty de to scan chain insertion. 17](#_Toc134410032)

[Area Report 18](#_Toc134410033)

[Slack Report 19](#_Toc134410034)

[Power dissipation 20](#_Toc134410035)

[Comparing before/after scan chain 20](#_Toc134410036)

[Comment: 21](#_Toc134410037)

[(e) Run ATPG using a commercial tool available and report the fault coverage. 22](#_Toc134410038)

圖目錄

[Fig 1 RTL code(divider.v) 4](#_Toc134549681)

[Fig 2 RTL code(divider.v) 5](#_Toc134549682)

[Fig 3 PATTERN.v (I/O ports and parameters) 7](#_Toc134549683)

[Fig 4 PATTERN.v (main task structure) 8](#_Toc134549684)

[Fig 5 PATTERN.v (clock generation) 8](#_Toc134549685)

[Fig 6 PATTERN.v (reset task) 9](#_Toc134549686)

[Fig 7 PATTERN.v (input task) 9](#_Toc134549687)

[Fig 8 PATTERN.v (check task) 9](#_Toc134549688)

[Fig 9 PATTERN.v (error task) 9](#_Toc134549689)

[Fig 10 PATTERN.v (fail task) 10](https://365nthu-my.sharepoint.com/personal/111063548_office365_nthu_edu_tw/Documents/碩一(下)/超大型積體電路測試/HW2/上傳報告/HW2-111063548.docx#_Toc134549690)

[Fig 11 PATTERN.v (pass task) 11](#_Toc134549691)

[Fig 12 TESTBED.v(testbench) 12](#_Toc134549692)

[Fig 13 10 pairs of inputs 13](#_Toc134549693)

[Fig 14 nWave result 13](#_Toc134549694)

[Fig 15 area report 14](#_Toc134549695)

[Fig 16 Slack report 15](#_Toc134549696)

[Fig 17 power dissipation 16](#_Toc134549697)

[Fig 18 area report(after scan chain insert) 18](#_Toc134549698)

[Fig 19 Slack report(after scan chain insert) 19](#_Toc134549699)

[Fig 20 power dissipation(after scan chain insert) 20](#_Toc134549700)

[Fig 21 Fault Coverage 22](#_Toc134549701)

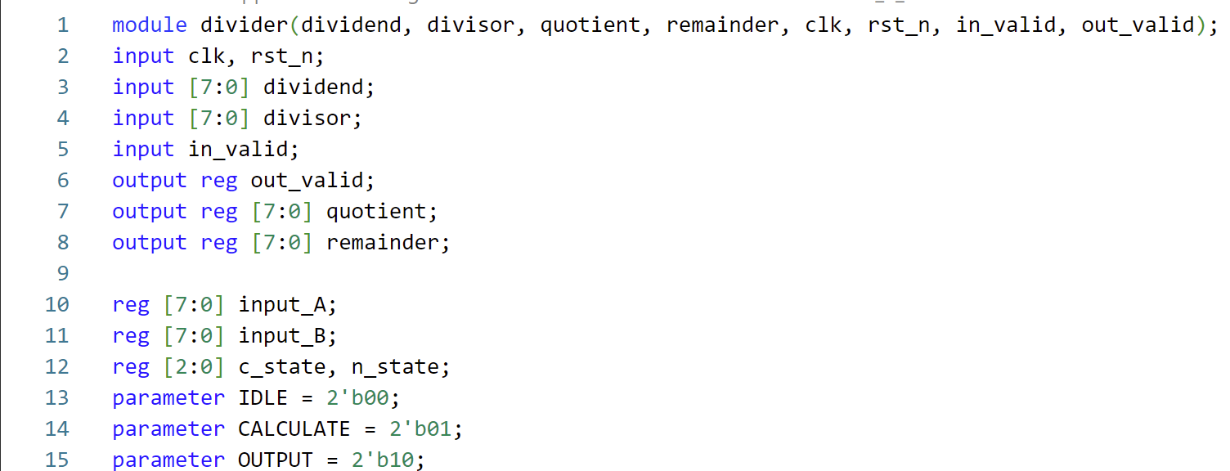
**分工:**

**蕭方凱:RTL、Testbench、Compiler、DFT、Report**

**尤弘瑋: Algorithm、RTL、Testbench、Compiler、ATPG**

# Write the RTL code in Verilog or VHDL that takes in two 8-bit positive integers, A[7:0] and B[7:0], and produces its quotient Q[7:0] and remainder R[7:0].

## divider.v



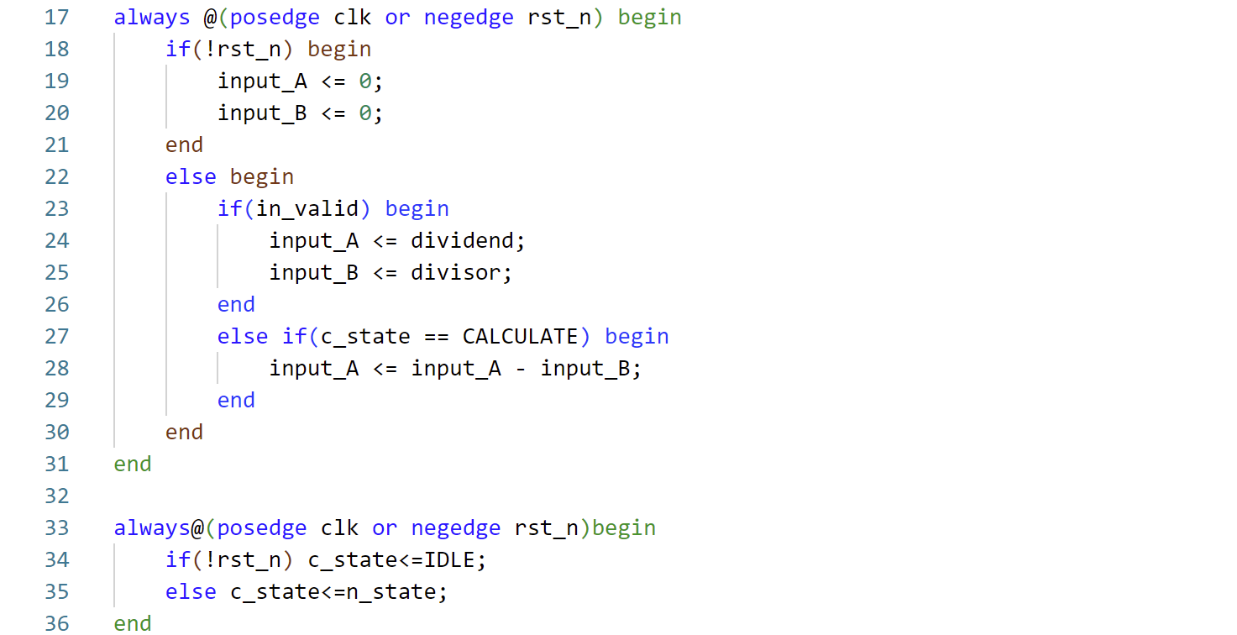
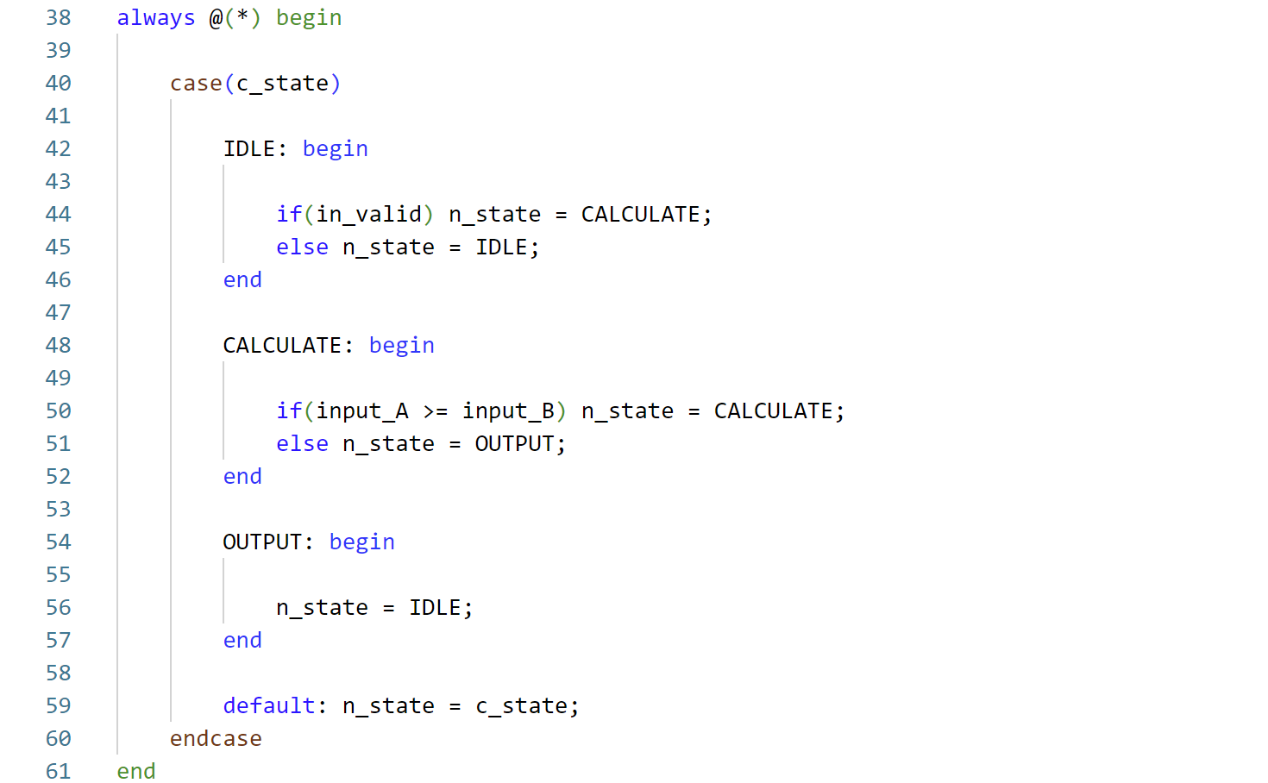


Fig 1 RTL code(divider.v)



一張含有 文字 的圖片

自動產生的描述

Fig 2 RTL code(divider.v)

## Finite State Machine

一張含有 圖表 的圖片

自動產生的描述

另外，除了FSM功能以外，RTL電路還包含以下主要功能:

1. rst\_n = 0時:

input\_A = 0; input\_B = 0; c\_state = IDLE; n\_state = IDLE; quotient = 0;

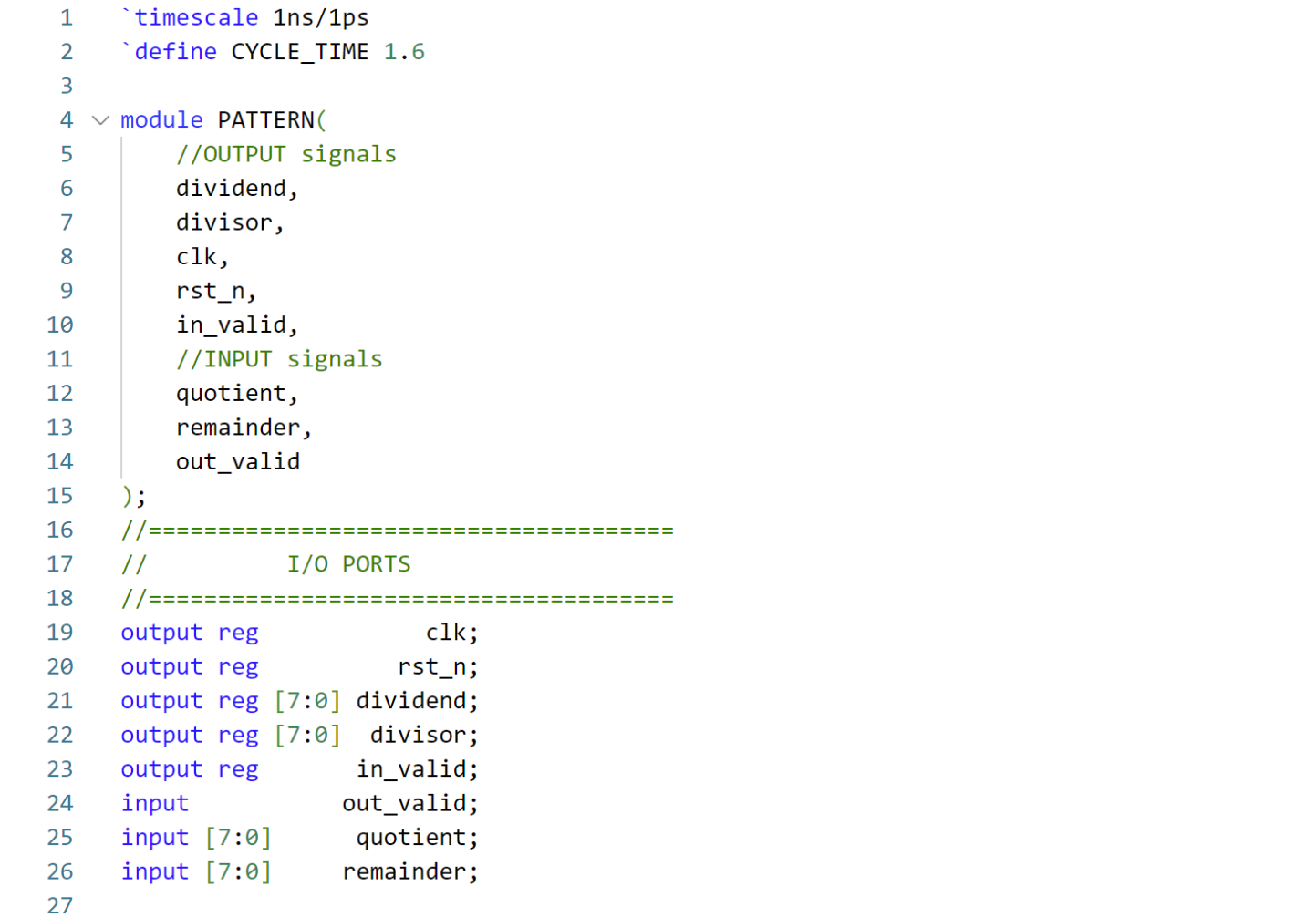
remainder = 0; out\_valid = 0

1. in\_valid = 1時:

將由pattern產生的dividend及divisor輸入給input\_A及input\_B。

# Verify the correctness of your RTL code by a testbench. You should try it out by at least 3 pairs of input numbers.

## PATTERN.v



一張含有 文字 的圖片

自動產生的描述

PATNUM: number of inputs = 10

Fig 3 PATTERN.v (I/O ports and parameters)

一張含有 文字 的圖片

自動產生的描述

Fig 4 PATTERN.v (main task structure)

Fig 5 PATTERN.v (clock generation)

此pattern分別有reset\_task、input\_task、check\_task、error\_task、fail\_task、pass\_task，及一些display以讓使用者觀察出輸入與輸出的對應結果。

以下為各個task之功能說明:

* Reset\_task: 負責將所有訊號歸零，歸零後始rst\_n常態保持在1。
* Input\_task: 負責input 被除數及除數，並算出正確的商及餘數。
* Check\_task: 驗證pattern算出的商及餘數是否與RTL算出的相同，若相同 代表RTL正確，否則RTL存在錯誤。
* Error\_task: display錯誤的input發生在第幾筆input，同時使err\_cnt加1，此err\_cnt代表錯誤筆數。
* Fail\_task: 顯示總共有幾筆錯誤資料，同時顯示出失敗小熊圖案。
* Pass\_task: 顯示PASS!! No errors were found!，同時顯示出成功小熊圖案。

一張含有 文字 的圖片

自動產生的描述

Fig 6 PATTERN.v (reset task)

Fig 7 PATTERN.v (input task)

一張含有 文字 的圖片

自動產生的描述

Fig 8 PATTERN.v (check task)

Fig 9 PATTERN.v (error task)

一張含有 文字 的圖片

自動產生的描述一張含有 資料表 的圖片

自動產生的描述

Fig 10 PATTERN.v (fail task)

一張含有 圖表 的圖片

自動產生的描述

Fig 11 PATTERN.v (pass task)

## TESTBED.v



PATTERN PART

RTL PART

Fig 12 TESTBED.v(testbench)

## Display on MobaXterm

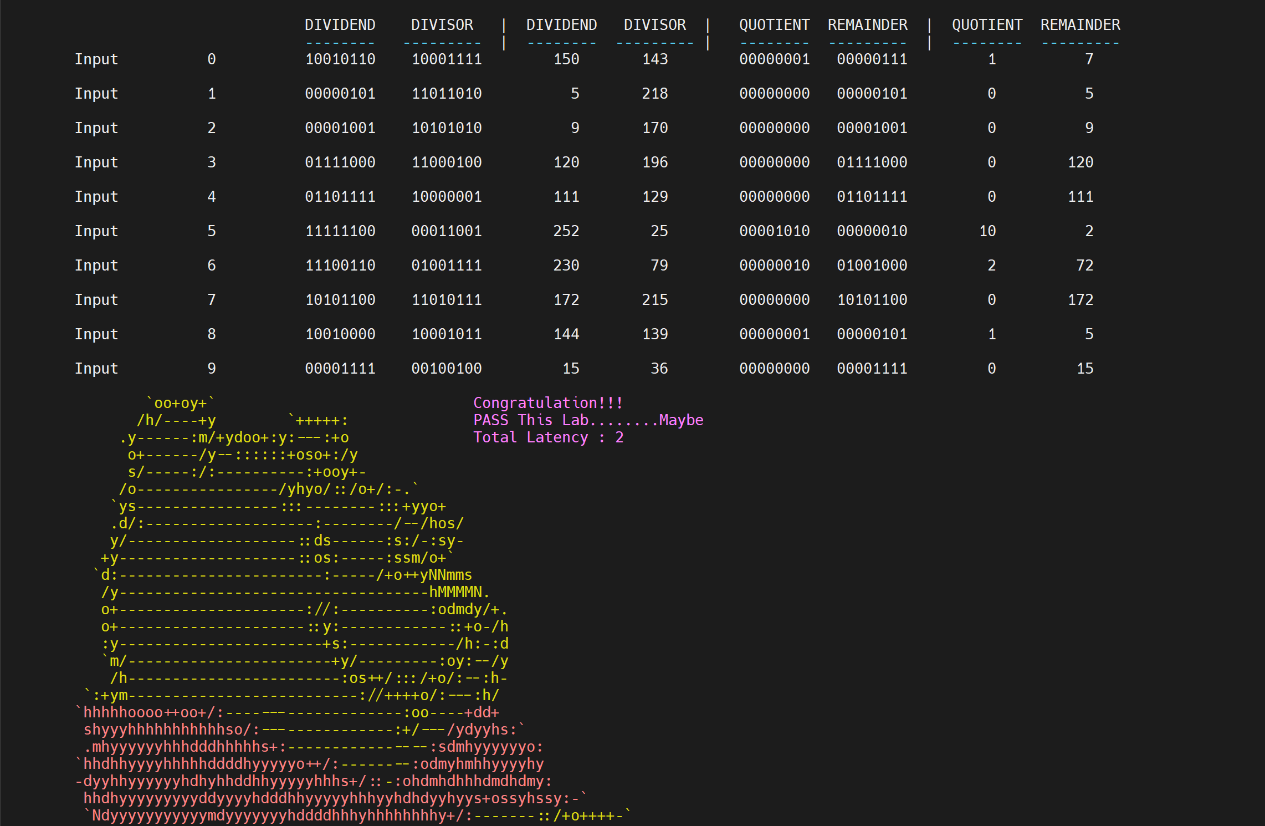
****

Fig 13 10 pairs of inputs

由左至右，分別為:

由pattern所產生的被除數及除數(binary) → 由pattern所產生的被除數及除數(decimal) → RTL的商數及餘數(binary) → RTL的商數及餘數(decimal)

## nWave

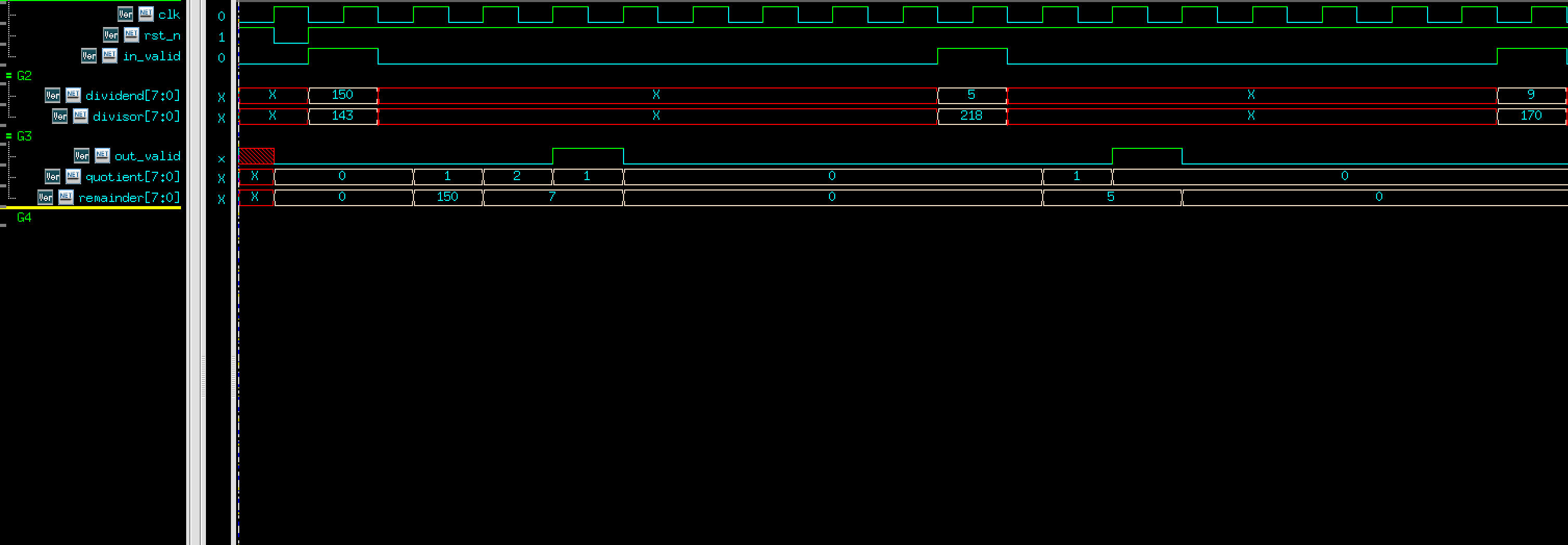


Fig 14 nWave result

一開始先將rst\_n變為0，使電路reset。Reset完後，將rst\_n拉回1，同時將in\_valid enable，使pattern提供一組被除數及除數，待RTL運算完成後，out\_valid變為1，此時的quotient及remainder即為輸出結果值。

以上面波形為例，第一筆輸入為黃色部分150/143，得到的結果為1…7; 第二筆輸入為5/218，得到的結果為0…5。

並且從波形上可看出，latency = 2 cycles。

# Use a synthesis script to convert your RTL code into a gate-level netlist. Report the final gate count, the maximum operating speed (in MHz) and the estimated power dissipation in (mW) using Design Compiler.

## Area Report

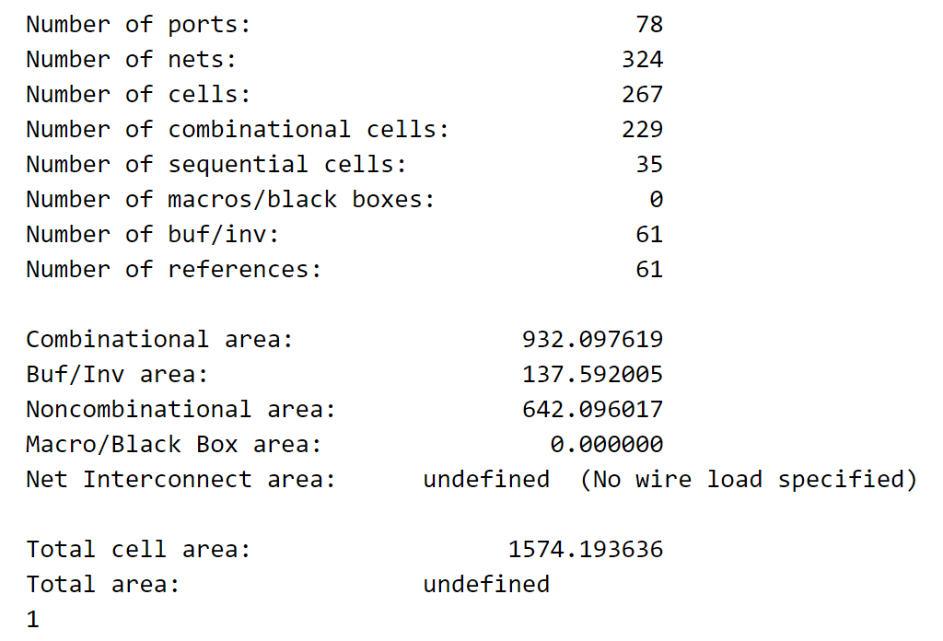


Fig 15 area report

**Total Area = 1574.194**

**Final Gate Count**

## Slack Report

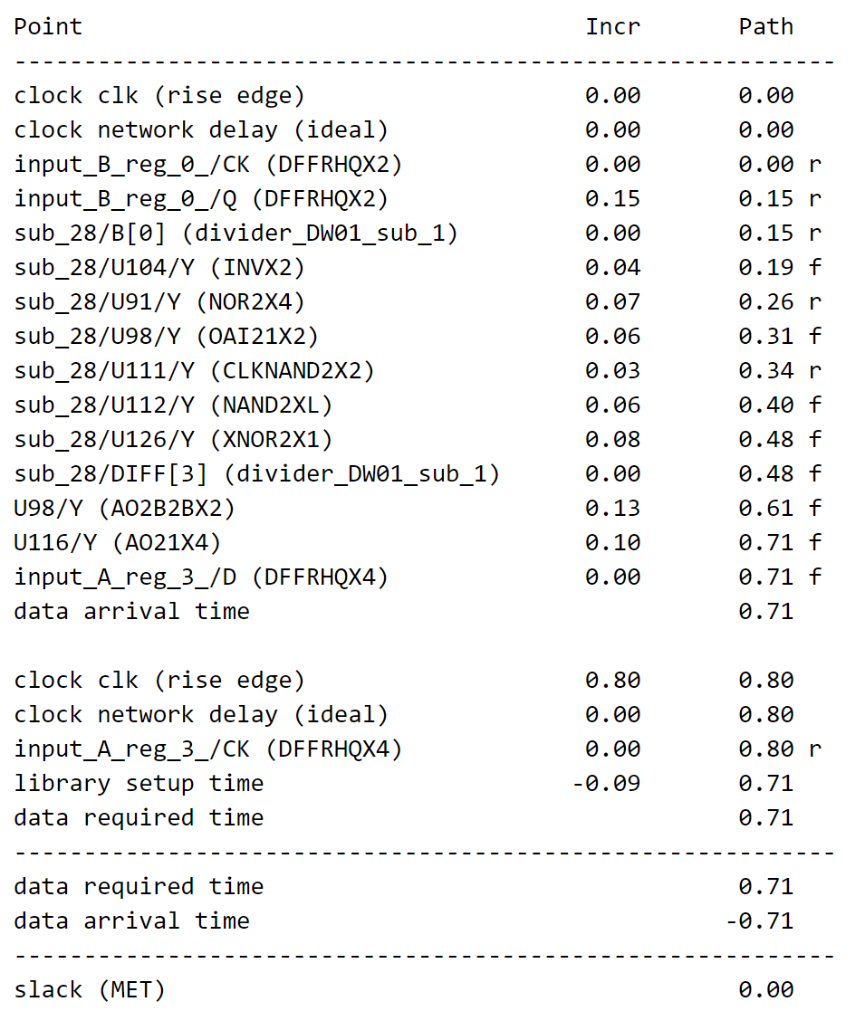


Fig 16 Slack report

**Clock period = 0.8ns**

**Maximum Operating speed**

## Power dissipation

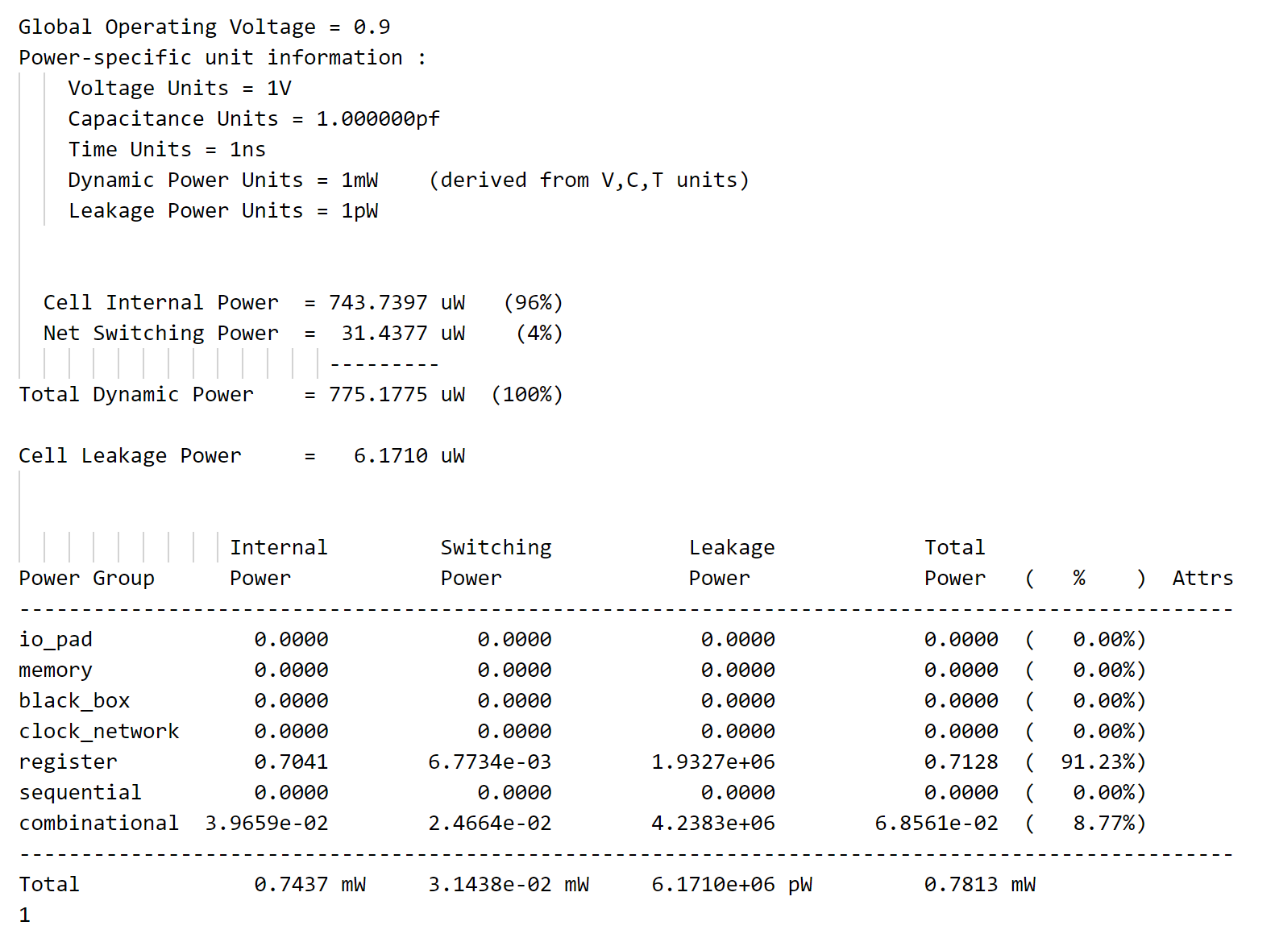


Fig 17 power dissipation

**Total Power Dissipation = 0.7813mW**

# Add the scan chain into your gate-level netlist obtained by part(c), report the resulting gate count, the maximum operating speed (in MHz) of your circuit. Compare to the non-scan version, and report the area overhead percentage and performance penalty due to scan chain insertion.

一張含有 文字 的圖片

自動產生的描述

* **DRC Report: Total violations: 0**
* **Sequential Cell Report: no violation**

## Area Report

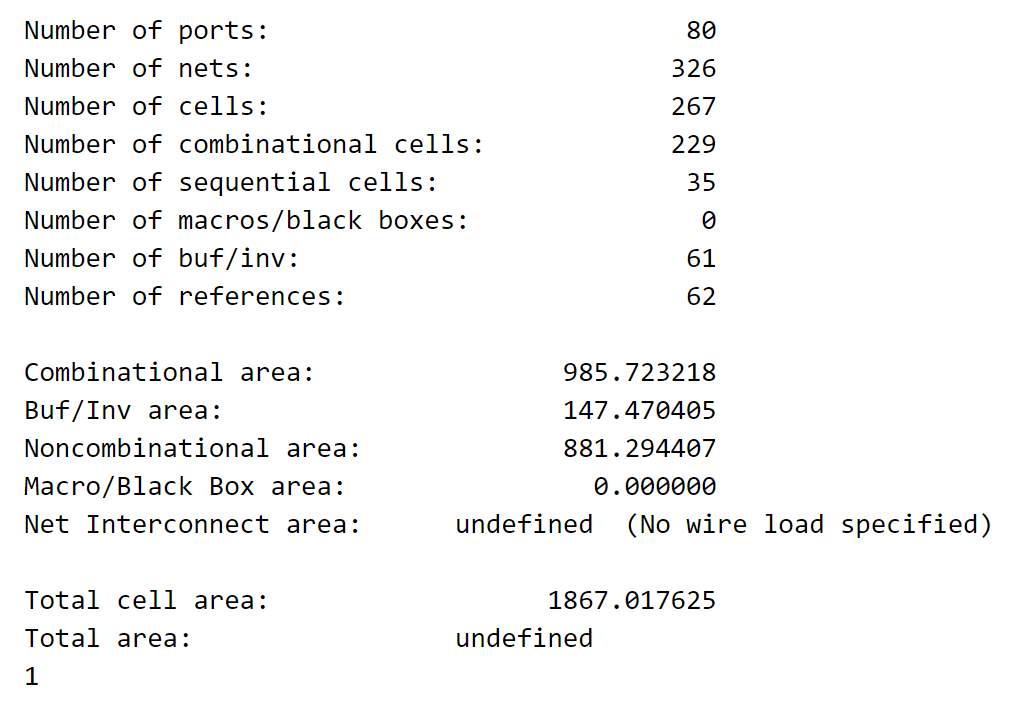


Fig 18 area report(after scan chain insert)

**Total Area = 1867.018**

**Final Gate Count**

**Overhead Percentage**

## Slack Report

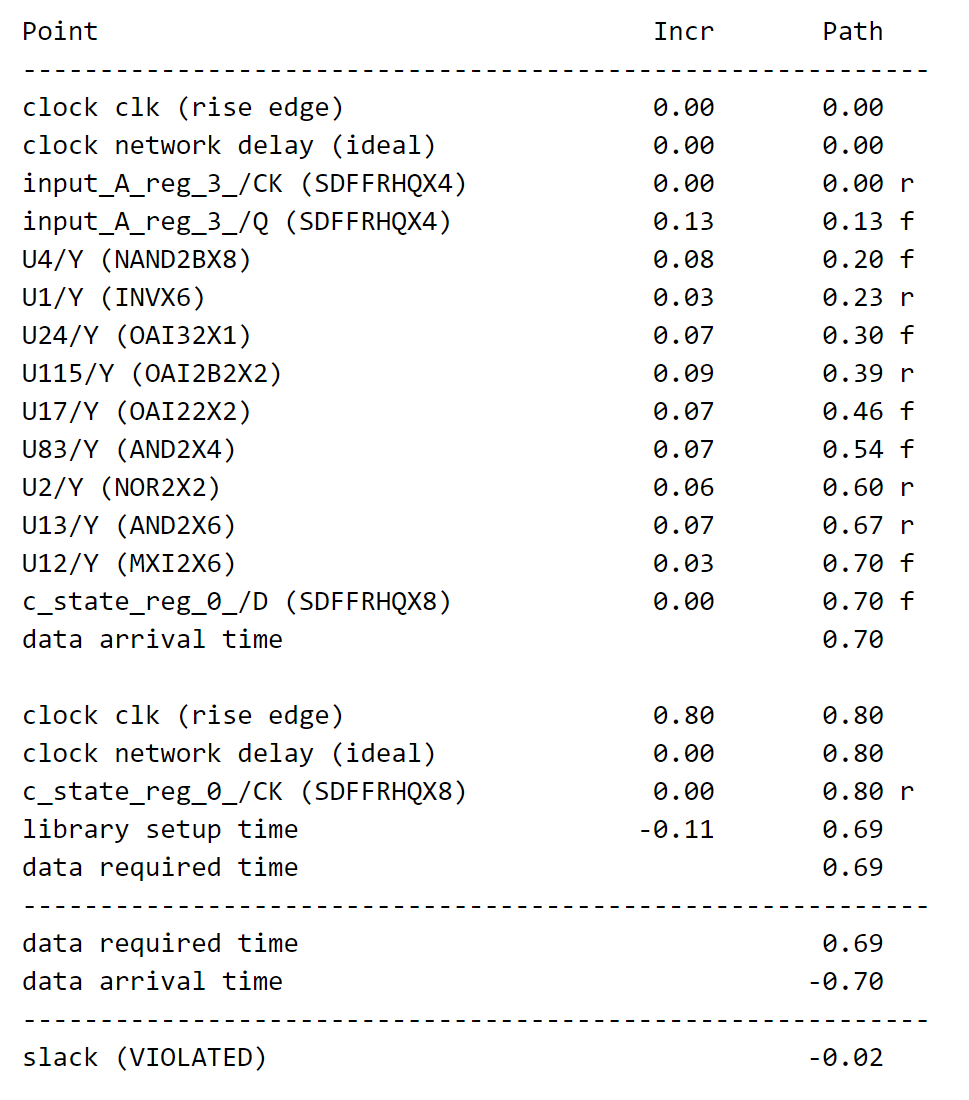


Fig 19 Slack report(after scan chain insert)

**Clock period = 0.8ns**

**Maximum Operating speed**

## Power dissipation

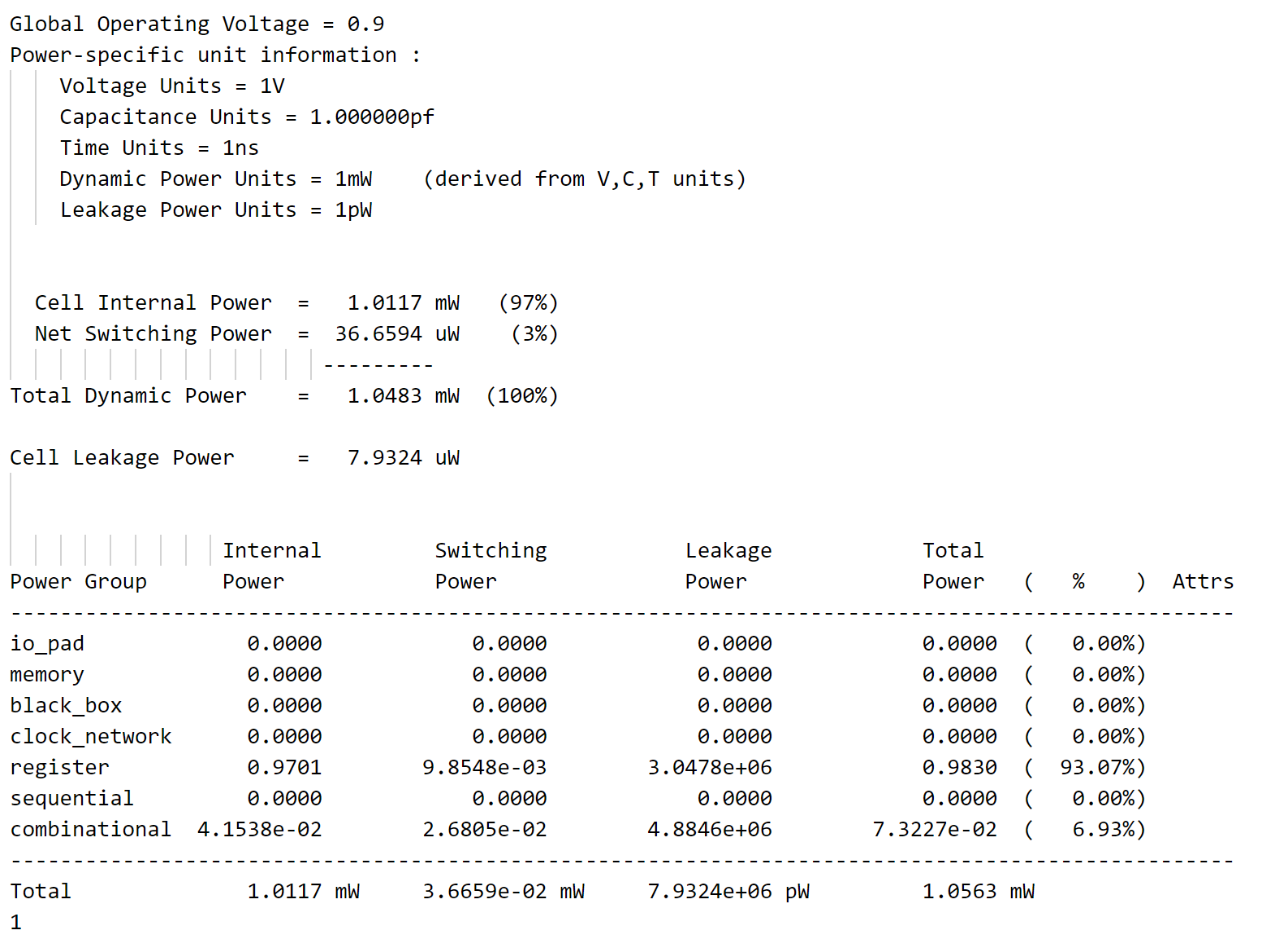


Fig 20 power dissipation(after scan chain insert)

**Total Power Dissipation = 1.0563mW**

## Comparing before/after scan chain

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Before scan chain** | **After scan chain** | **Overhead Percentage** |
| **Area** | 1574.194 | 1867.018 | 15.684% |
| **Operating speed** | 1250MHz | 1219.512MHz | 2.5% |
| **Power dissipation** | 0.7813mW | 1.0563mW | 26.03% |

## 

## Comment:

本組在clk period上試了很多組合，因operation speed及area是一個trade off，無法同時兼顧兩者。

以下為幾組不同結果(均為before insert scan chain):

1. 在clk period=2ns時(預設)，divider面積約為1167，power為0.202mW。
2. 在clk period=1.5ns時，divider面積約為1205，power為0.271mW。
3. 在clk period=1ns時，divider面積約為1394，power為0.518mW。
4. 在clk period=0.8ns時，divider面積約為1574，power為0.781mW。

**由上面幾組數據可得到在在clk period=2ns時，有最好的整體表現，雖然頻率不快，但面積及功耗都算小，是個不錯的設計。**

**但因此作業不會比較數據，且在合成RTL電路時盡可能使slack為0會比較好，故本作業還是使用clk period = 0.8ns為報告數據。**

# Run ATPG using a commercial tool available and report the fault coverage.

**Fault Coverage:**

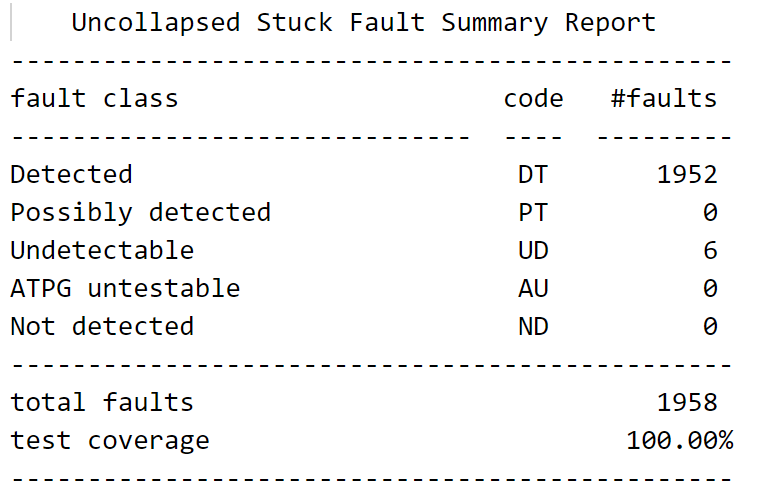


Fig 21 Fault Coverage