

Computer-Aided VLSI System Design

Final Project: Gauss-Seidel Iteration Machine

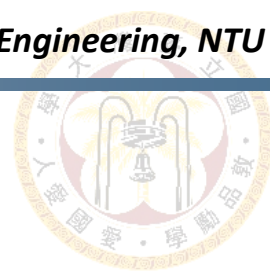
Graduate Institute of Electronics Engineering, National Taiwan University

MediaTek



NTU GIEE

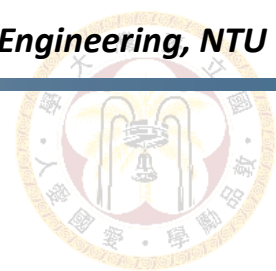




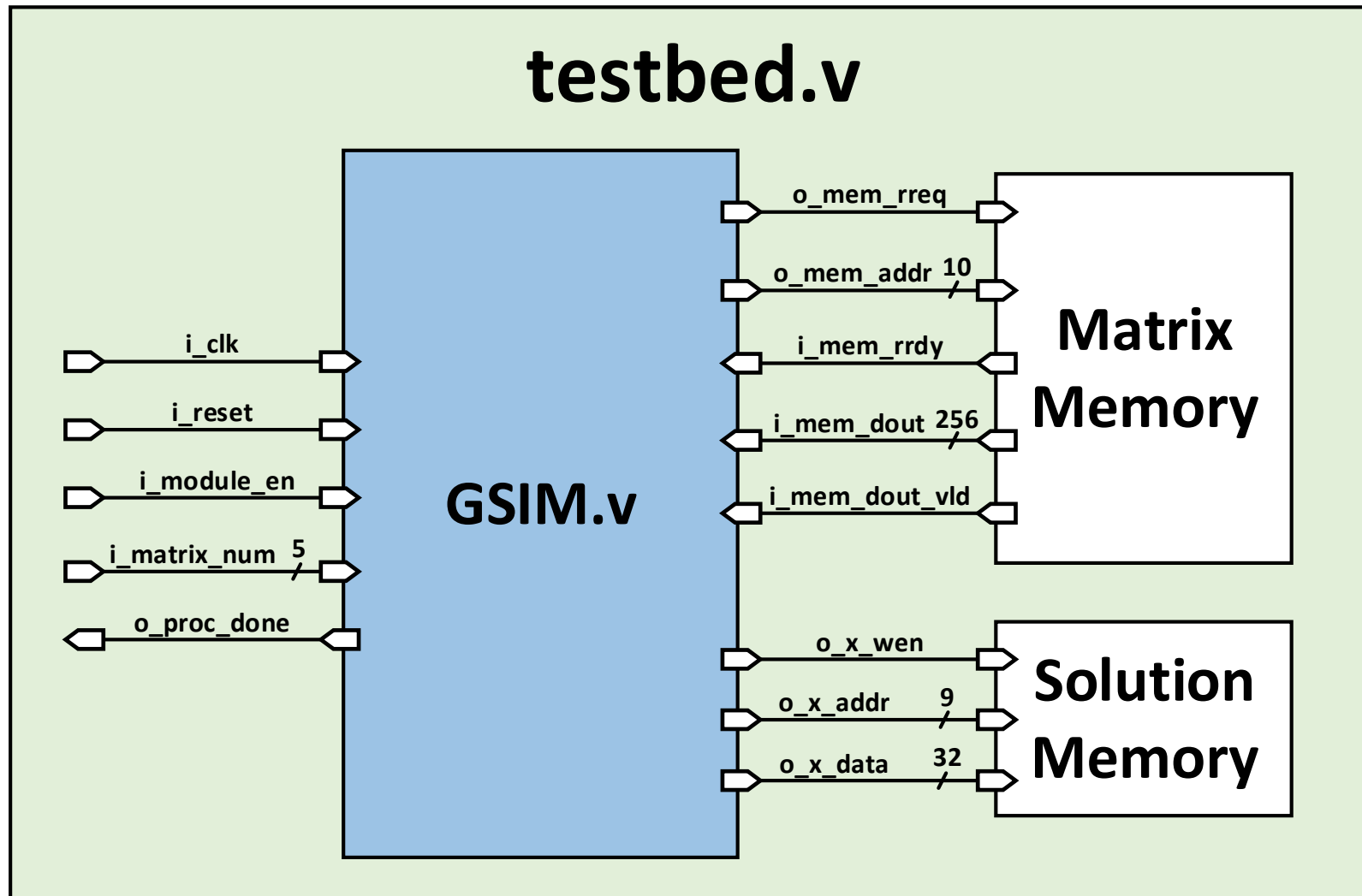
Introduction

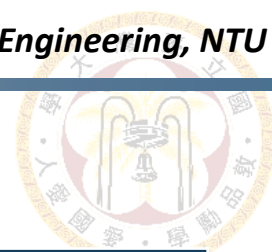
- 請完成一Gauss-Seidel Iteration Machine(GSIM)的電路設計來求出多元線性聯立方程式(Linear Equation)之解
- 如下圖所示，矩陣A、B為已知之整數值，待求矩陣X之解
 - 在此專題中， N 固定為16

$$AX=B \longrightarrow \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_N \end{bmatrix}$$



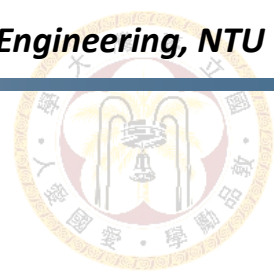
Block Diagram





Input/Output

Signal Name	I/O	Width	Simple Description
i_clk	I	1	本系統為同步於時脈正緣之同步設計。 (註: Host端採clk正緣時送資料。)
i_reset	I	1	高位準"非"同步(active high asynchronous)之系統重置信號。
i_module_en	I	1	模組控制訊號。當為high時模組操作有效。
i_matrix_num	I	5	要計算矩陣數量。
o_proc_done	O	1	運算完成訊號。當將所有要求的解輸出後，須將此訊號設為high代表運算完成，並在i_module_en為0時再設為low。
o_mem_rreq	O	1	要讀取matrix memory時須設為high。
o_mem_addr	O	10	要讀取matrix memory之位址。
i_mem_rrdy	I	1	要讀取matrix memory之ready訊號。為high時代表此時可讀取memory data。
i_mem_dout	I	256	Matrix memory data。共有16筆16-bit資料，採用2's complement表示。細節請參考。
i_mem_dout_vld	I	1	Matrix memory data有效訊號。為high時代表此時i_mem_dout有效。
o_x_wen	O	1	輸出資料有效之控制訊號。當為High時，表示目前輸出的資料為有效的。
o_x_addr	O	9	輸出矩陣解之位址。
o_x_data	O	32	要輸出之矩陣解。採用2's complement表示(16-bit整數+16-bit小數)。



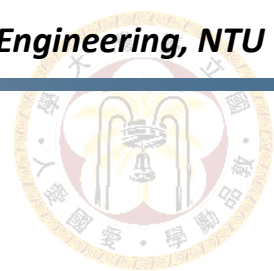
Gauss-Seidel Iteration Machine

- 所求多元線性聯立方程式如下式所示

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \cdots + a_{1N}x_N &= b_1 \\a_{21}x_1 + a_{22}x_2 + \cdots + a_{2N}x_N &= b_2 \\&\vdots \\a_{N1}x_1 + a_{N2}x_2 + \cdots + a_{NN}x_N &= b_N\end{aligned}\tag{1}$$

- 欲求 x_1, x_2, \dots, x_N 的值，可以將上式整理成底下式子

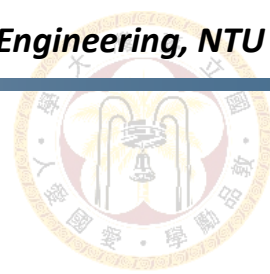
$$\begin{aligned}x_1^1 &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^0 - \cdots - a_{1N}x_N^0) \\x_2^1 &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^1 - a_{23}x_3^0 - \cdots - a_{2N}x_N^0) \\&\vdots \\x_N^1 &= \frac{1}{a_{NN}}(b_N - a_{N1}x_1^1 - a_{N2}x_2^1 - \cdots - a_{NN-1}x_{N-1}^1)\end{aligned}\tag{2}$$



Gauss-Seidel Iteration Machine

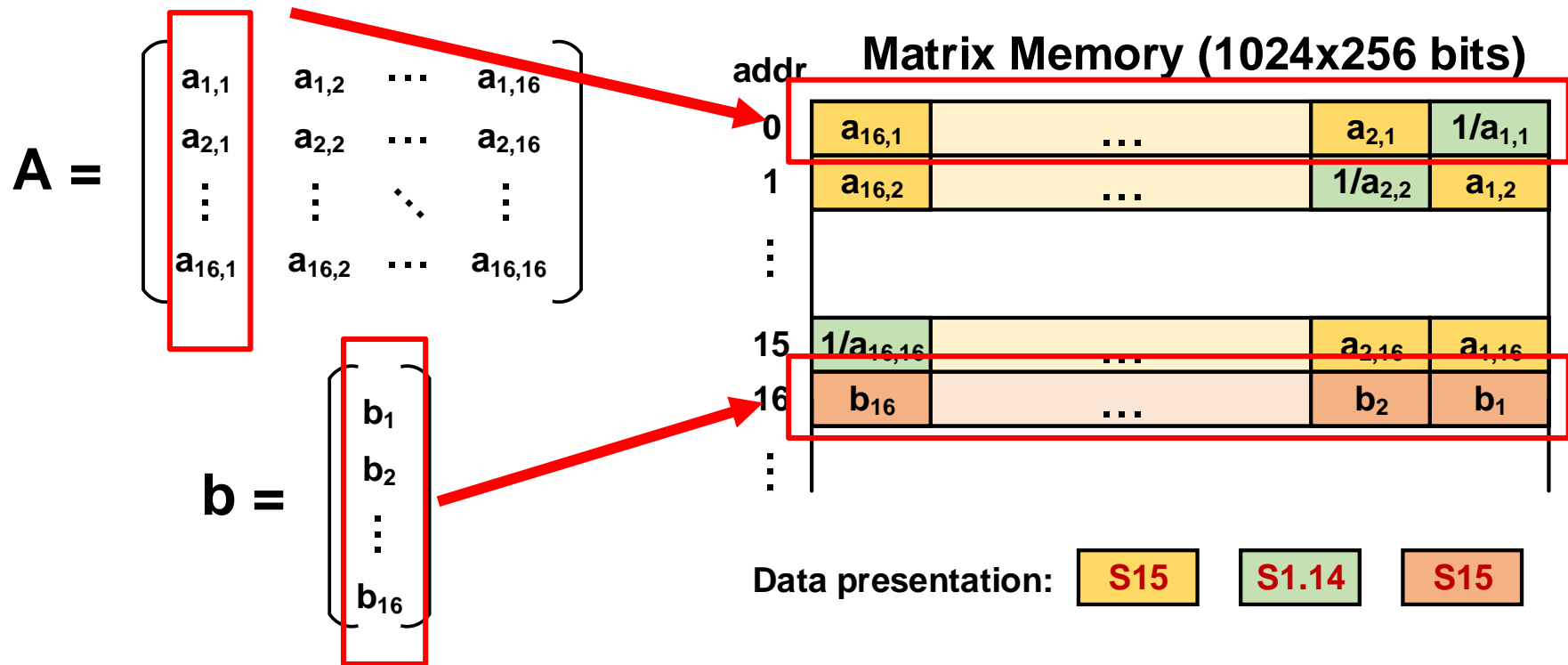
- Gauss-Seidel Iteration就是將(2)式作相同的動作數次的疊代，其行為如下式所示，反覆地疊代數次後，即可將所有待求的x值收斂在某一個值，該x值即為所求，
 - 在此專題中，疊代次數固定為16。

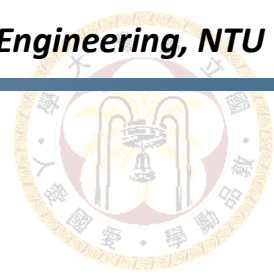
$$x_i^{k+1} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^N a_{ij} x_j^k \right] \quad (3)$$



Data Presentation

- 2's complement
 - Ex. 16-bit(2-bit整數+14-bit小數) -> **S1.14**
- 矩陣A及b都是放在外面的memory中，由設計者決定怎麼讀取





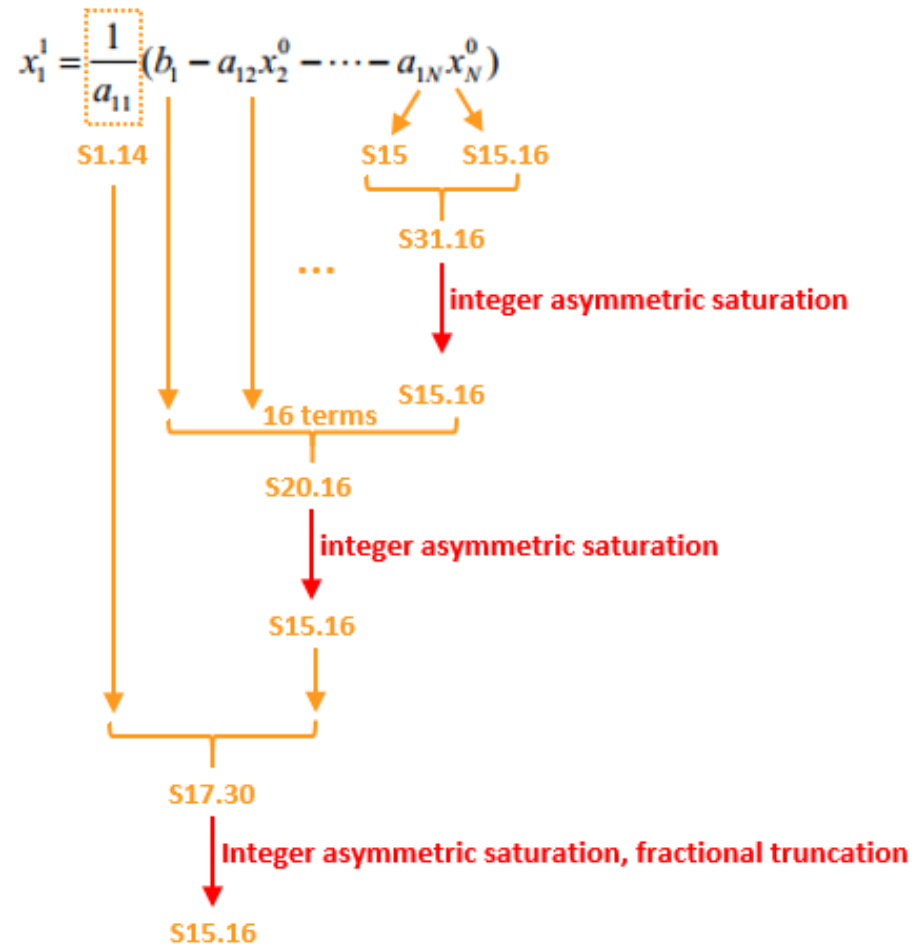
Order for Computation

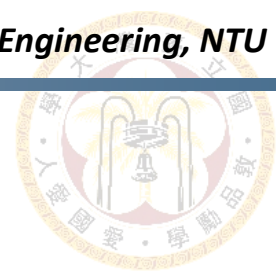
Integer asymmetric saturation

- 若發生overflow，則取其能表示的最大值/最小值來做為結果

Fractional truncation

- 小數部分當bit-width變小時直接truncate即可(不用四捨五入)





Initialization

- X 初始化方式如下

$$X^0 = \begin{bmatrix} x_1^0 \\ x_2^0 \\ \vdots \\ x_N^0 \end{bmatrix} = \begin{bmatrix} b_1/a_{11} \\ b_2/a_{22} \\ \dots \\ b_N/a_{NN} \end{bmatrix}$$

b_N : S15

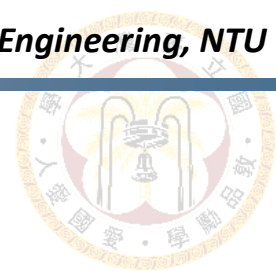
$1/a_{NN}$: S1.14

S15.16

S17.14



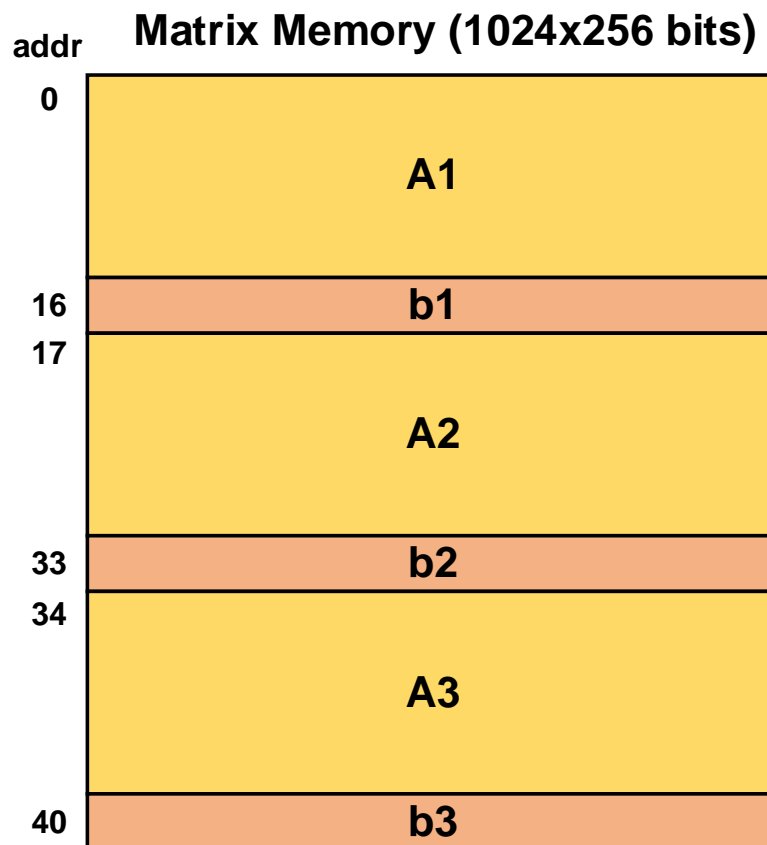
Integer asymmetric saturation

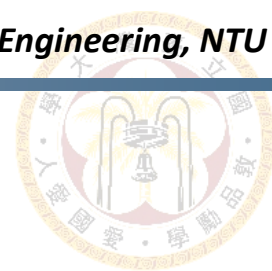


Matrix Storage

- 假設要處理3個矩陣，則其儲存在memory的順序如下

A1 -> b1 -> A2 -> b2 -> A3 -> b3



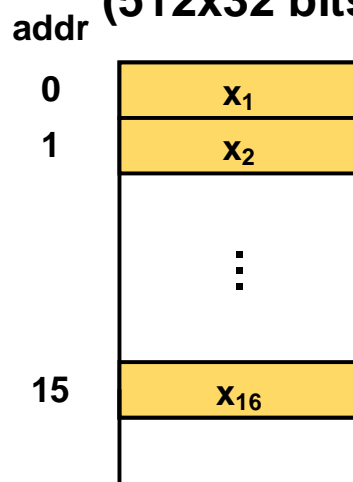


Result Output

- 將矩陣解輸出儲存至solution memory
 - 一次只輸出32-bit答案

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{16} \end{bmatrix}$$

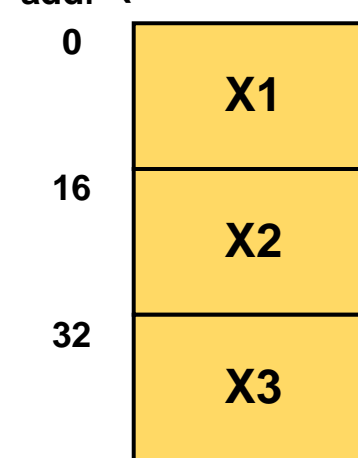
**Solution Memory
(512x32 bits)**



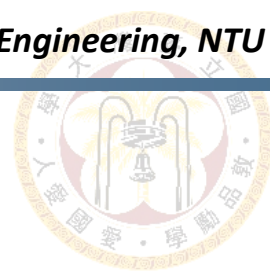
(a)

- 如果要解的矩陣有三個

**Solution Memory
(512x32 bits)**

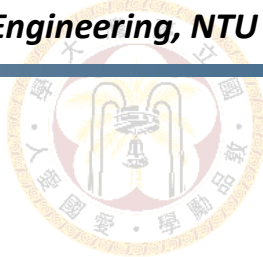


(b)

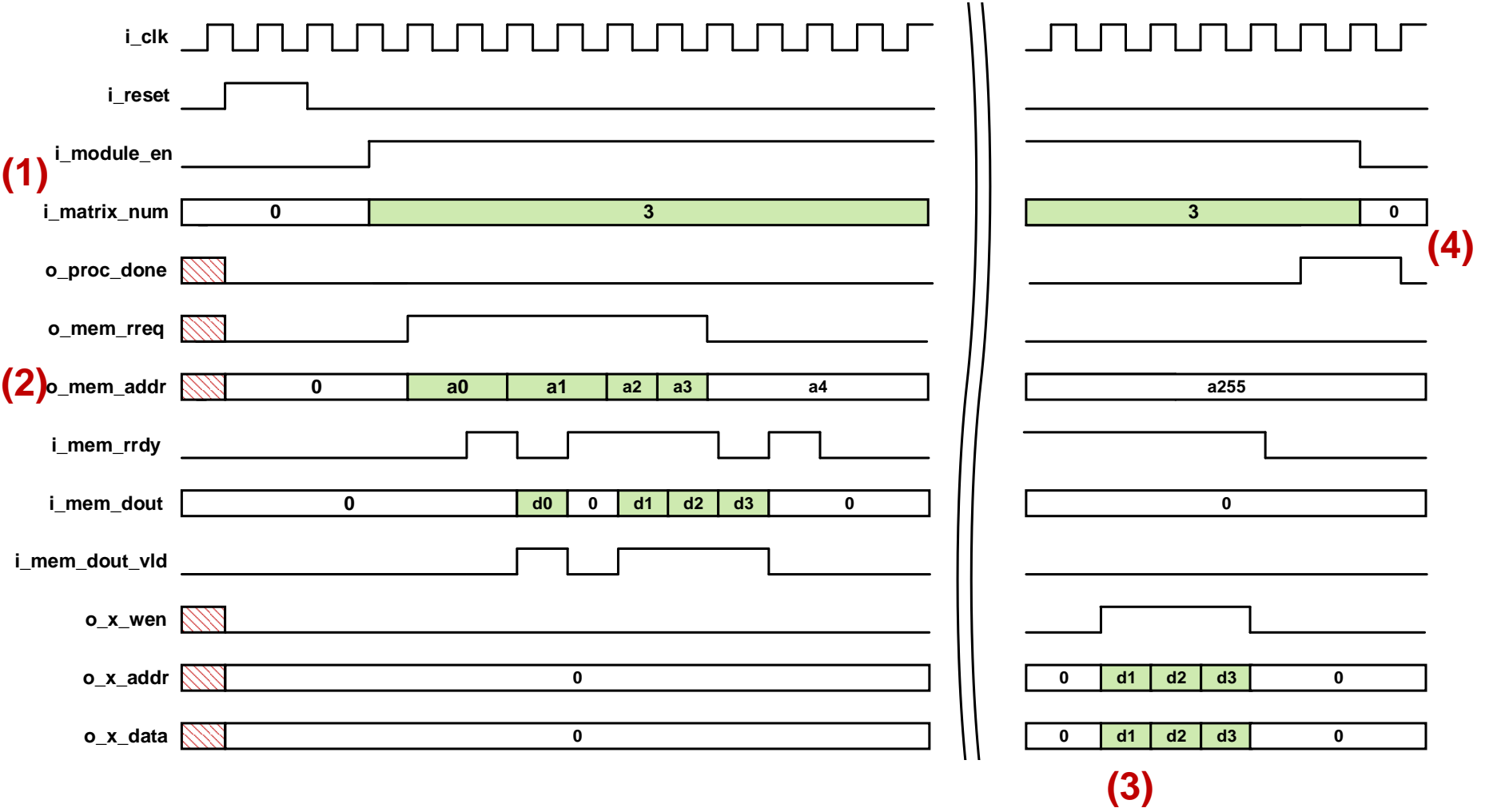


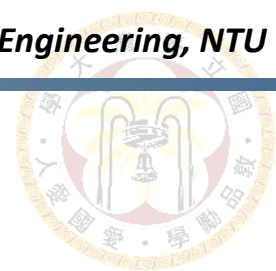
Specification

- Only worst-case library is used for synthesis.
- The synthesis result of data type should **NOT** include any **Latch**.
- The slack for setup-time should be **non-negative**.
- **No any timing violation and glitches** for the gate level simulation and post-layout simulation.

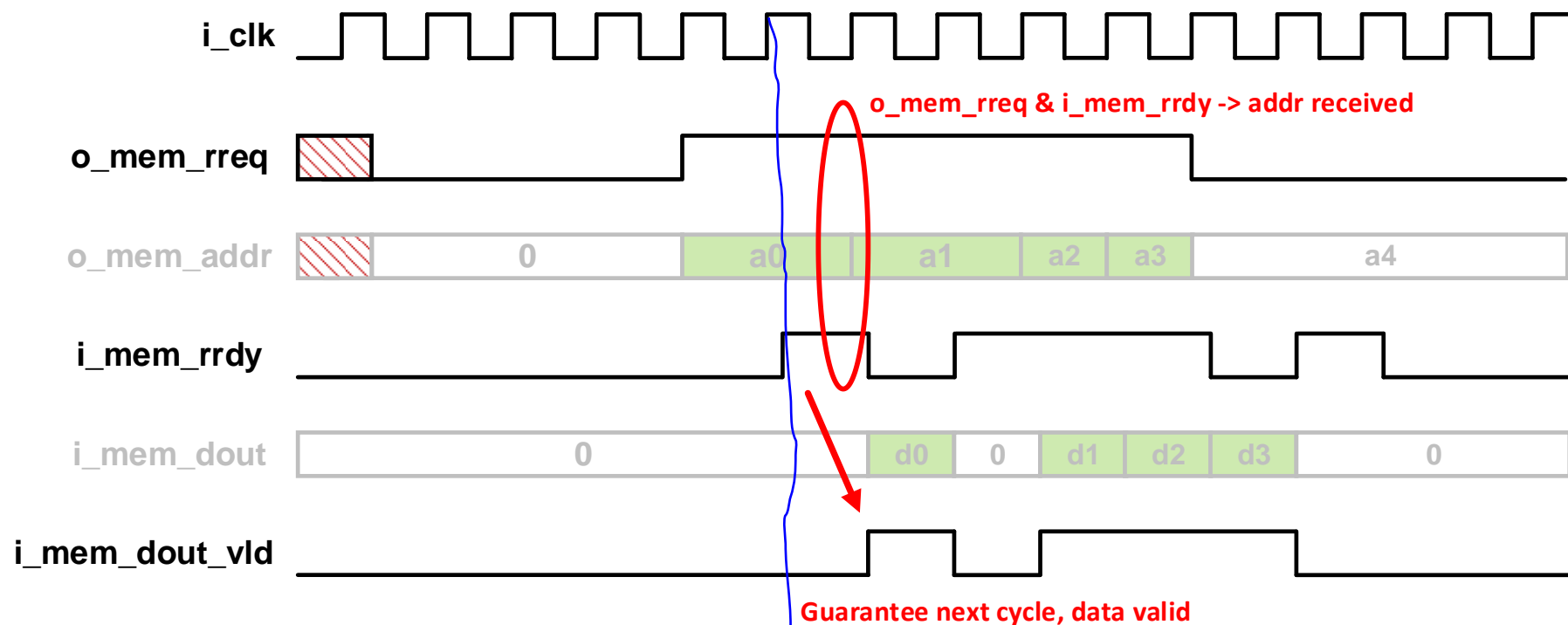


Waveform

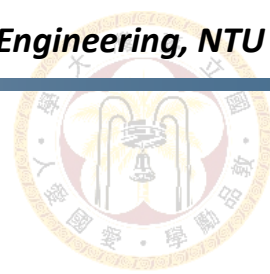




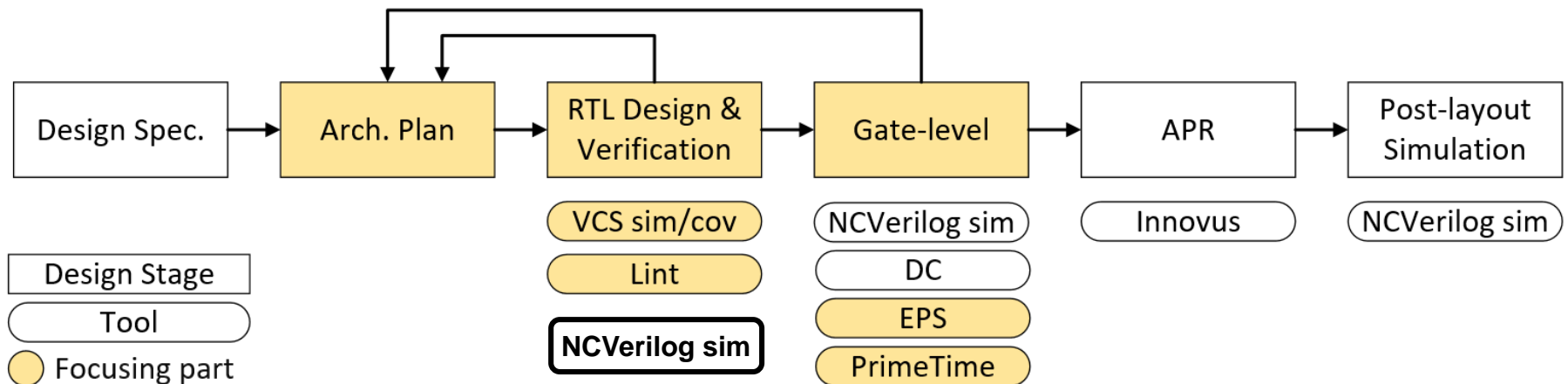
Handshake

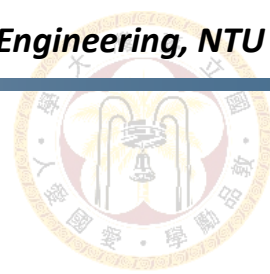


i_mem_rrdy在每個cycle為1的機率為0.5



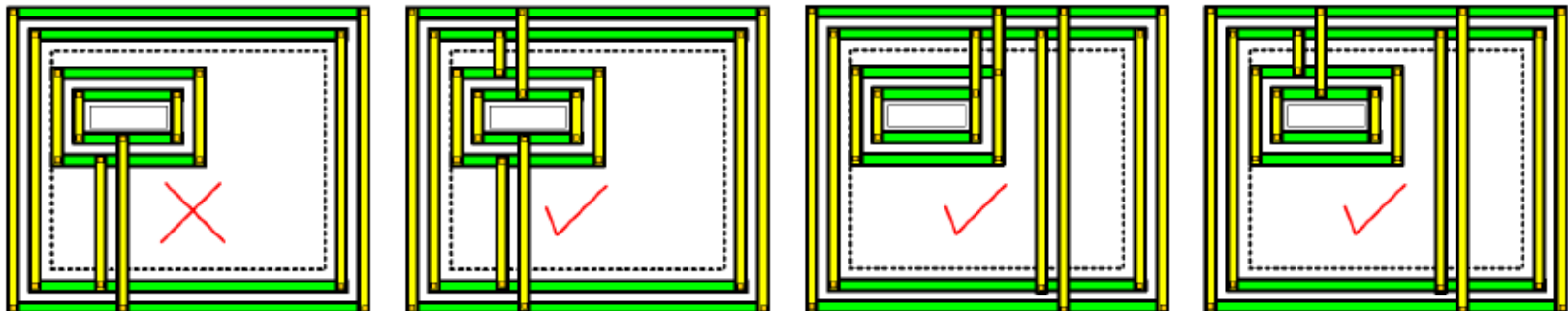
Design Flow

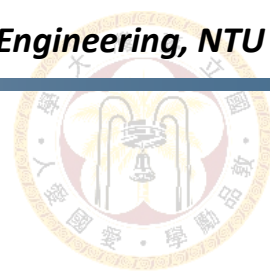




Specifications for APR (1)

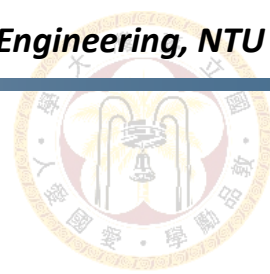
- 只需做 Marco layout 即不用包含 IO Pad 、 Bonding Pad)
- **set_aspect_ratio 0.6**
- VDD 與 VSS Power Ring 寬度請各設定為 **2um** 只須做一組
- 不需加 Dummy Metal
- **Power Stripe 務必至少加一組**，其 VDD 、 VSS 寬度各設定為 **2um**
 - **Power Stripe 垂直方向至少一組，水平方向可不加**





Specifications for APR (2)

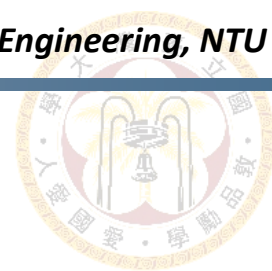
- 務必要加 Power Rail (follow pin)
- Core Filler 務必要加
- APR 後之 GDSII 檔案務必產生
- 完成 APR DRC/LVS 完全無誤
- 記得先產生GSIM.ioc，再重新讀取該檔來設定 pin position



Grading Policy (1)

- **Baseline** 50% + **Performance** 35% + **Report** 15%

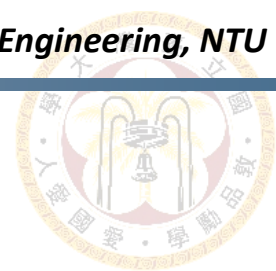
Item		%	Description
RTL Simulation		10	通過提供的5個pattern
Verification		10	Coverage (line 100%), nLint no Error
Synthesis		15	EPS, Pass gate-level sim
APR		15	Finish APR with no DRC/LVS errors Pass post-layout simulation
Performance	Area, time	20	Area x Time
	Power	15	10: Compare active window, total energy 5: idle, idle_after_active
Report		15	



Grading Policy (2)

- **Baseline** 50% + **Performance** 35% + **Report** 15%

Violation	Penalty
不符合 design specification	Performance*0.5
無法通過hidden pattern	Performance*0.5
沒有考慮random i_mem_rrdy	Performance不評分
違反繳交格式與規則	總分-3

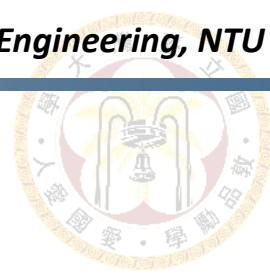


Grading Policy (3)

- RTL Simulation 注意事項
 - 沒有考慮random i_mem_rrdy但能通過tb0~tb4也可以拿到所有分數
- Coverage 注意事項
 - Coverage只針對執行**tb3**的**line coverage**的結果來進行評分()

Name ▾	Score	Line	Toggle	FSM	Condition	Branch
testbed	82.96%	95.45%	90.04%	50.00%	83.02%	96.30%
u_GSIM	80.28%	96.00%	73.41%	50.00%	83.33%	95.83%
u_matrix_mem	99.58%	100.00%	98.73%			100.00%

- Synthesis 注意事項
 - 沒有做EPS但能過gate-level simulation還是能拿到所有分數
 - 有做EPS除了能比較有機會做出比較小的面積以外，Report分數也會看有沒有使用EPS進行評分
- Power注意事項
 - 只需跑**gate-level tb4**即可
 - 評分會用Energy來進行評分



Performance (1)

■ Score = Area × Time

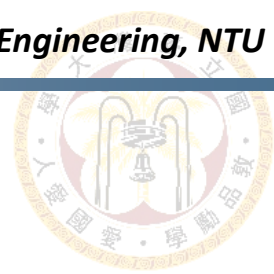
– Area

```
innovus #> analyzeFloorplan
```

```
***** Analyze Floorplan *****
Die Area(um^2)      : 1147937.22
Core Area(um^2)     : 174367.11
Chip Density (Counting Std Cells and MACROs and IOs): 83.061%
Core Density (Counting Std Cells and MACROs): 80.961%
Average utilization  : 100.000%
Number of instance(s) : 9110
Number of Macro(s)    : 3
Number of IO Pin(s)   : 57
Number of Power Domain(s) : 0
***** Estimation Results *****
```

– Time

```
----- Congratulation! You have pass all the pattern! -----
Simulation complete via $finish(1) at time 404572700 PS + 0
../00_TESTBED/testbench.v:171 $finish;
```



Performance (2)

■ Power

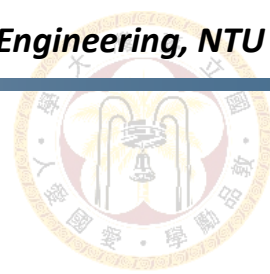
- idle_power和idle_after_active_power請根據模擬情形進行調整 (參考MTK講義p12)
- 若idle_power和idle_after_active_power的difference超過5%，此部分扣3分

```
## ===== idle window ===== TA modify
read_vcd -strip_path testbed/u_GSIM ./gsim.fsdb \
| | | -time {10.5 1010.5}
update_power
report_power
report_power > try_idle.power

## ===== active window ===== TA modify
read_vcd -strip_path testbed/u_GSIM ./gsim.fsdb \
| | | -when {i_module_en}
#report_switching_activity -list_not_annotated -show_pin

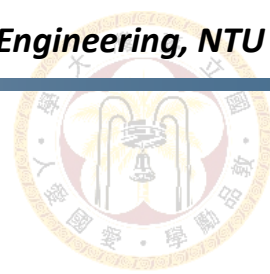
update_power
report_power
report_power > try_active.power

## ===== idle_after_active window ===== TA modify
read_vcd -strip_path testbed/u_GSIM ./gsim.fsdb \
| | | -time {98913.5 99913.5}
update_power
report_power
report_power > try_idle_after_active.power
```



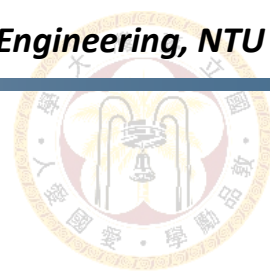
Report

- 需要包含底下幾個項目
 - 架構設計
 - 硬體優化方法 (latency, area, power...)
 - nLint report with 0 errors
 - Coverage result
 - Congestion map (如果有跑EPS流程)
 - Primetime power report (Gate-level)
 - Layout
 - Area result
 - Performance 表格



Submission (1)

- **GSIM.v**
- **GSIM_syn.v**
- **GSIM_syn.sdf**
- **GSIM_pr.v**
- **GSIM_pr.sdf**
- **GSIM.gds**
- **GSIM_final.tar** (archive of the design database directory)
- **report.pdf**
- **all other design files** included in your design for rtl simulation (optional)



Submission (2)

- **Due Friday, Jan. 14, 23:59**
- **Final project presentation (MTK experience sharing)**
 - **Date: January 18, 2021**