NYCU-ECE
DCS-2023

Exam : Online Test
Design : Systolic Array
TA : Li, Tsung-Jen

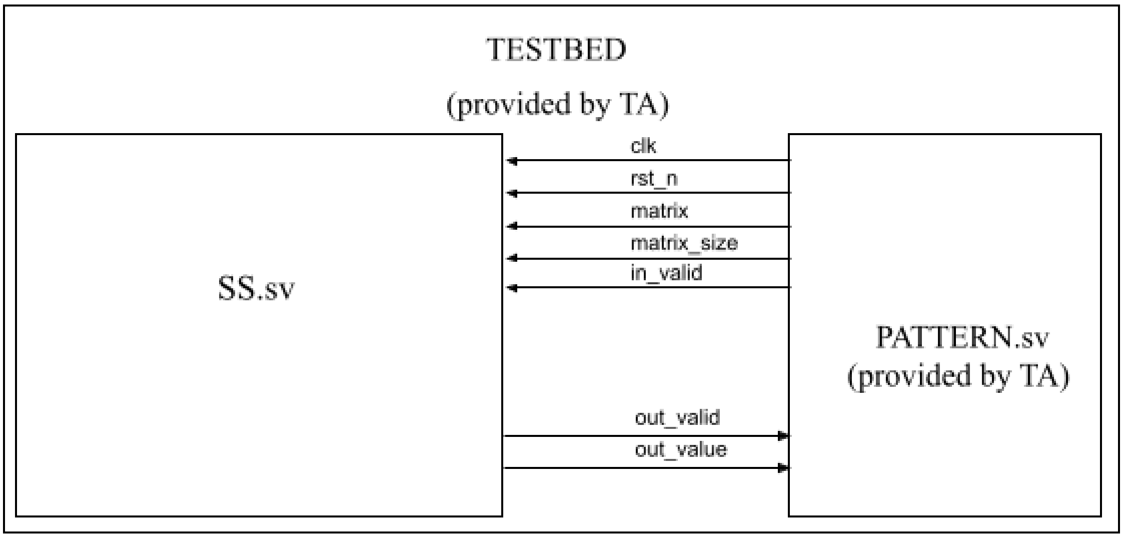
Data Preparation

1. Extract online test data from TA's directory :

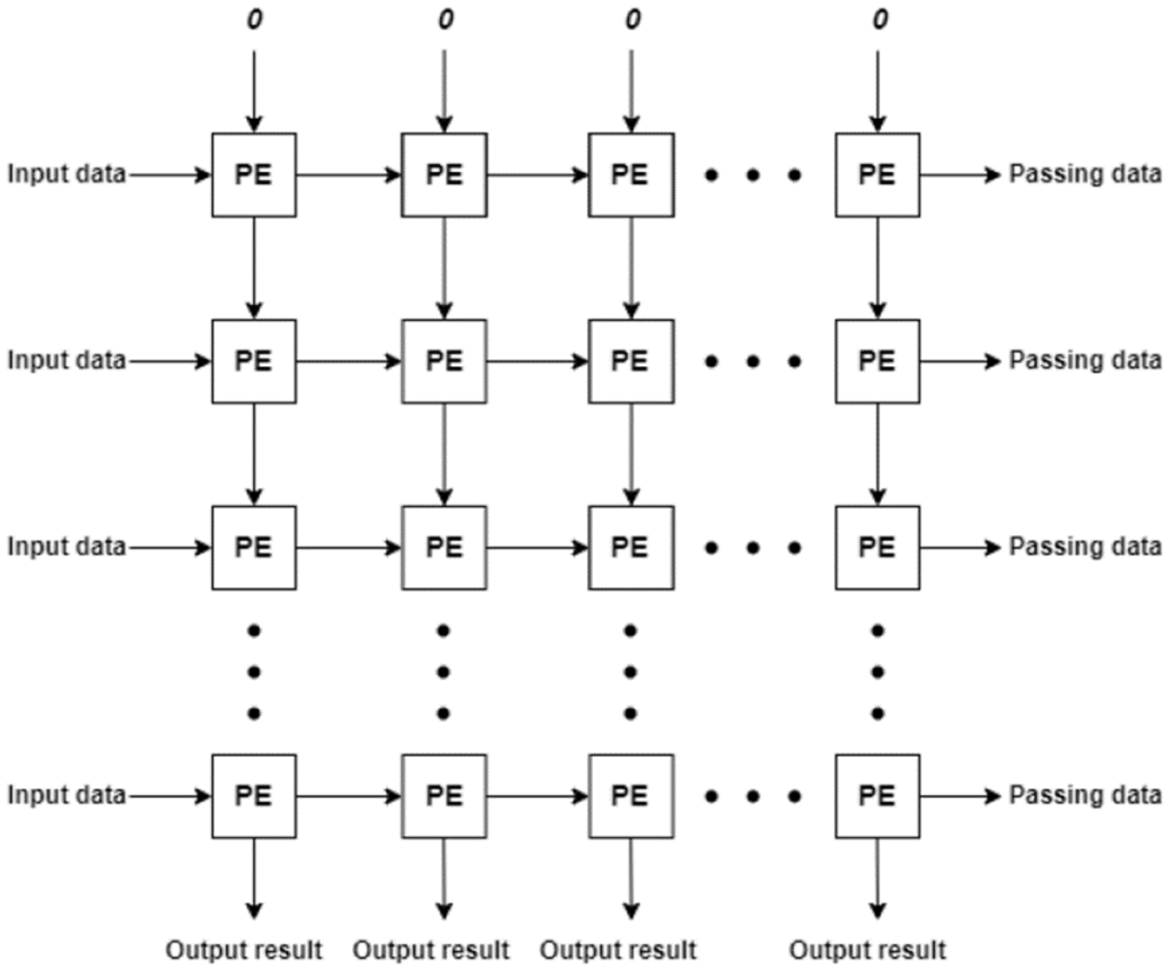
   % tar -xvf ~dcsta01/Final.tar

2. The extracted directory contains :
   a. 00_TESTBED/
   b. 01_RTL/
   c. 02_SYN/
   d. 03_GATE/

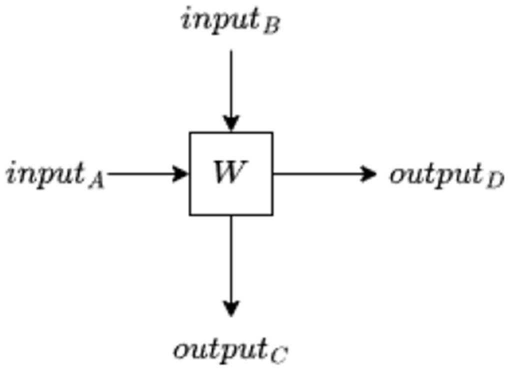
Block Diagram



Transformer based model has become the *de facto* backbone in the NLP region. Moverover, matrix-matrix multiplication and matrix-vector multiplication account for the majority of the computation. The goal of the online test is to design a matrix-matrix multiplication module which applies the method of "Systolic Array". To simplify the question, all we have to concern is the square-shaped matrix.



The behavior of each PE is Multiply-Accumulate (MAC) operation.



$$output_C = input_A * W + input_B$$

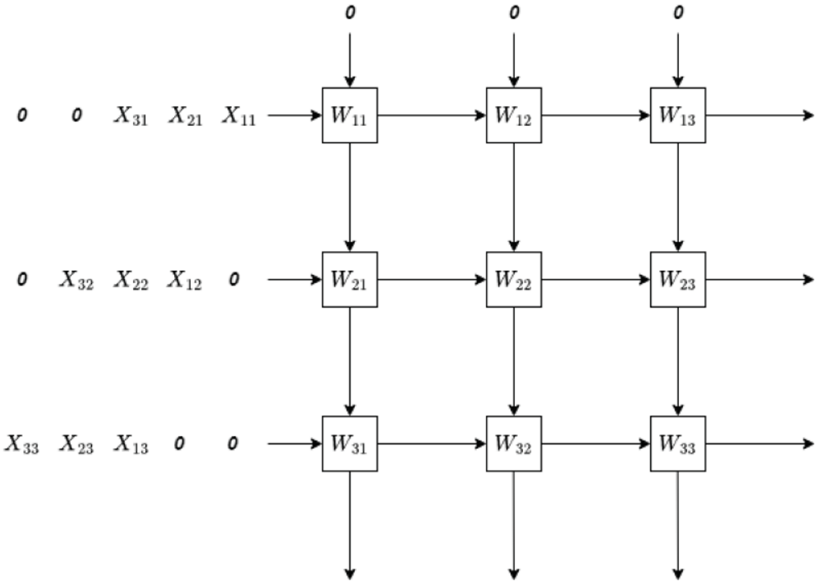
$$output_D = input_A$$

- Input

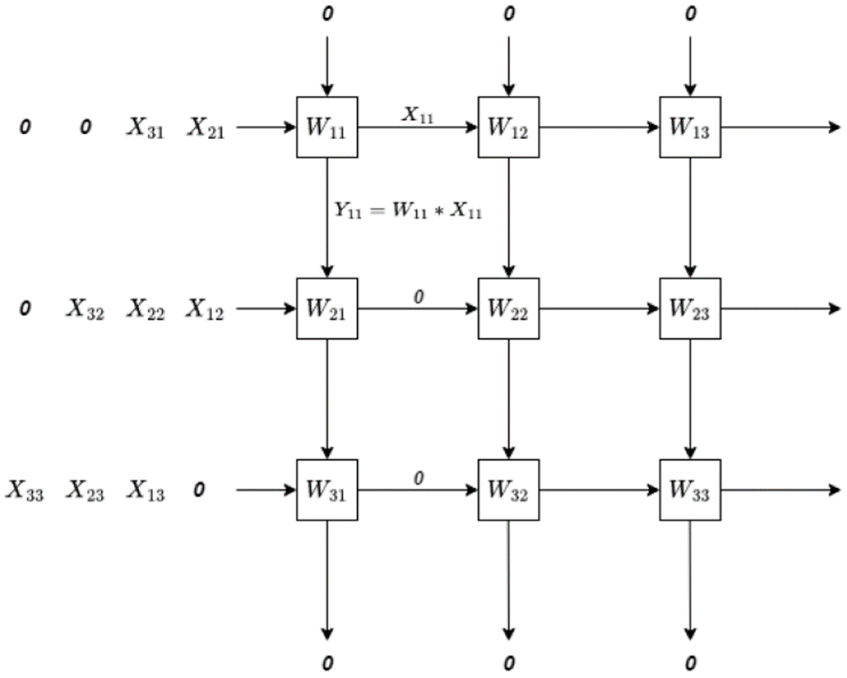
  The continuous data would be input once the ‘in_valid’ is high. It would be given in raster scan order, which is the same as the final project as well as Lab07. The first half of the cycles is the duration for the weight matrix, while the second one is for the input matrix.

- The operation of Systolic Array for each cycle

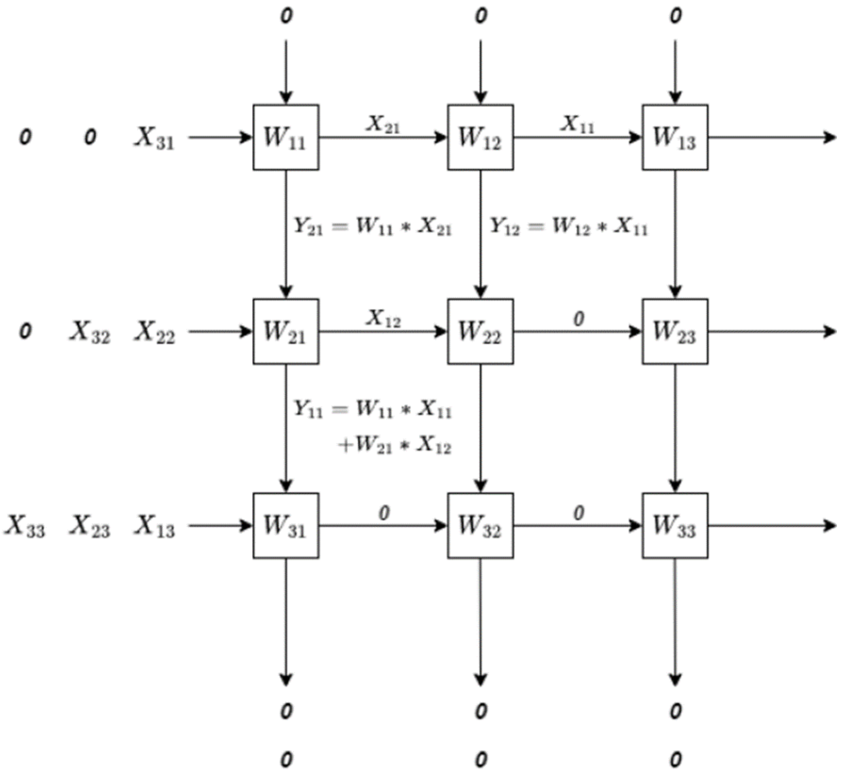
  Cycle 0


0
0
0
0 0 X_{31} X_{21} X_{11}
W_{11}
W_{12}
W_{13}
0 X_{32} X_{22} X_{12} 0
W_{21}
W_{22}
W_{23}
X_{33} X_{23} X_{13} 0 0
W_{31}
W_{32}
W_{33}

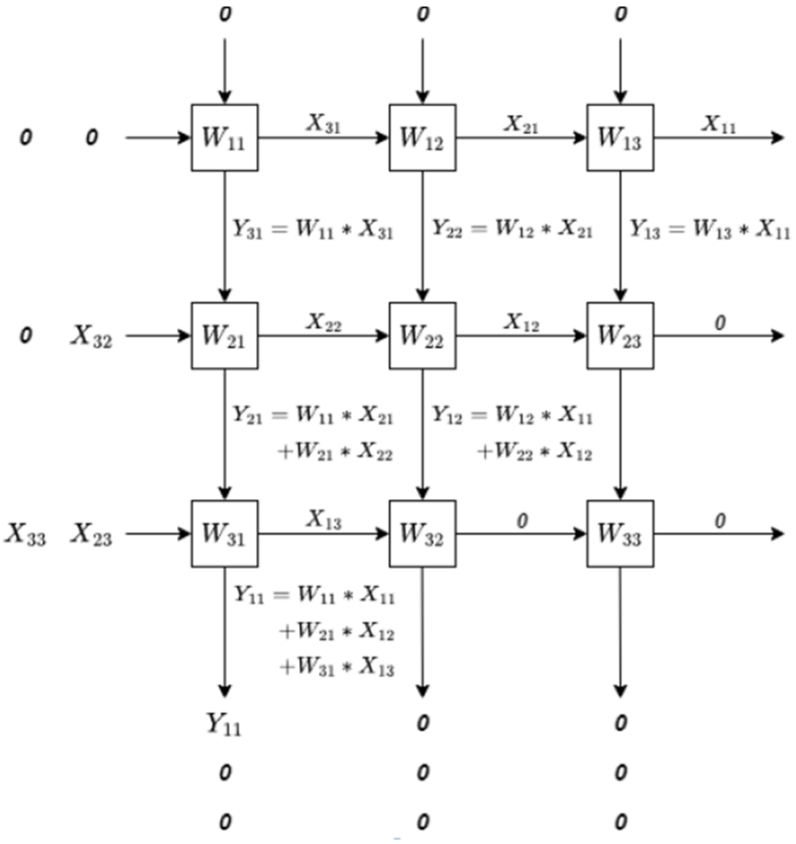
  Cycle 1



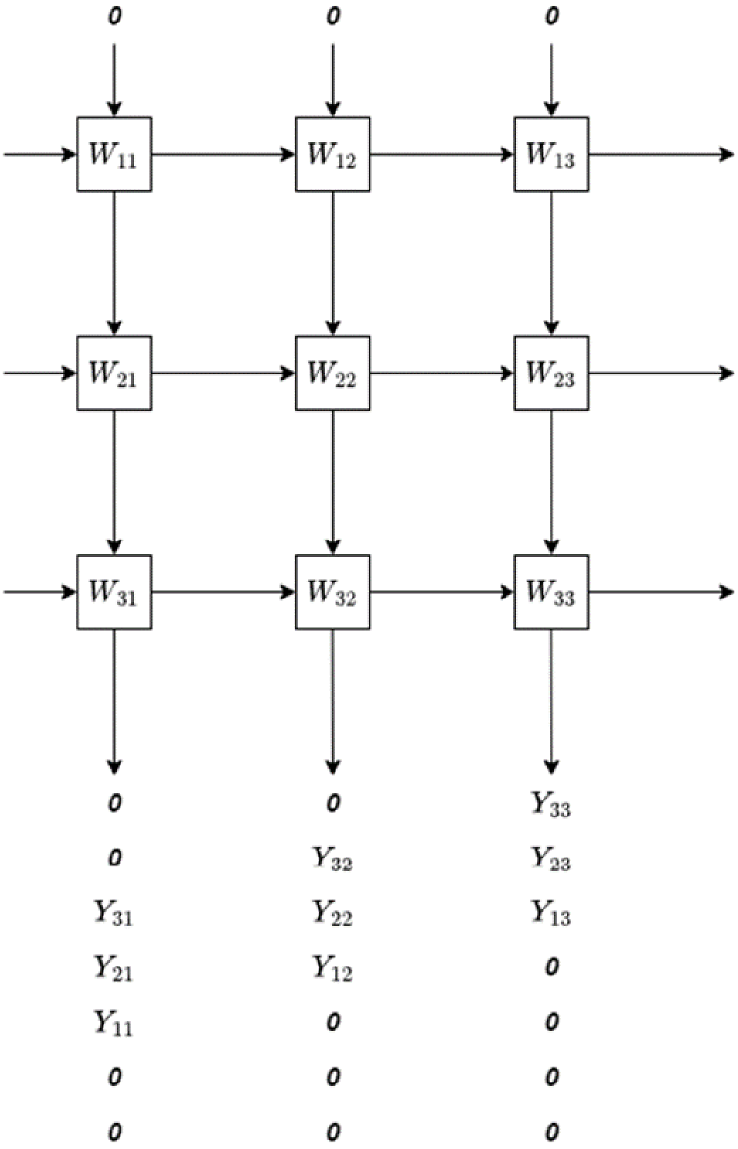
Cycle 2



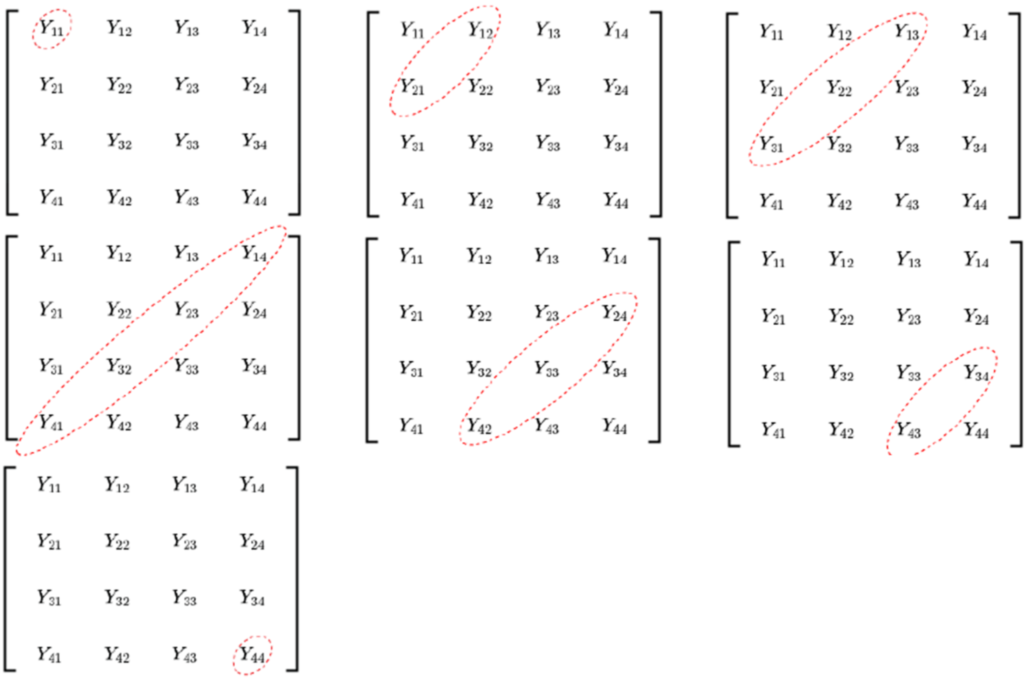
Cycle 3



Cycle 7



- Output



In this example, you will output the results with 7 cycles, and the output value will be :

Cycle0 → Y11
Cycle1 → Y12 + Y21
Cycle2 → Y13 + Y22 + Y31
Cycle3 → Y14 + Y23 + Y32 + Y44
Cycle4 → Y24 + Y33 + Y42
Cycle5 → Y34 + Y43
Cycle6 → Y44

Inputs

Signal name	Number of bit	Description
clk	1-bit	Clock Signal(clock cycle time : 5ns)
rst_n	1-bit	Asynchronous active-low reset
in_valid	1-bit	'in_valid' would be high for 8 or 32 cycles based on the 'matrix_size'.
matrix	16-bit	The continuous data would be input once the 'in_valid' is high. It would be given in raster scan order, which is the same as the final project as well as Lab07. The first half of the cycles is the duration for the weight matrix, while the second one is for the input matrix. $2 \times 2\, matrix: \ first\ 4\ cycles \ \rightarrow\ weight$ $last\ 4\ cycles \ \rightarrow\ input$ $4 \times 4\, matrix: \ first\ 16\ cycles \rightarrow\ weight$ $last\ 16\ cycles \ \rightarrow\ input$
matrix_size	1-bit	$1'b0:\ 2 \times 2\, matrix$ $1'b1:\ 4 \times 4\, matrix$

Outputs

Signal name	Number of bit	Description
out_valid	1-bit	'out_valid' should be high for **exactly** 3 or 7 cycles.
out_data	40-bit	'ont_value' outputs the correct value when 'out_valid' is 1. When 'out_valid' is 0, 'out_value' **should be set to 0**.

Specifications

1. Top module name : **SS**
2. Design file name : **SS.sv**
3. It is an **asynchronous reset** and **active-low** architecture. If you use synchronous reset (reset after clock starting) in your design, you may fail to reset signals.
4. The reset signal (***rst_n***) would be given only once at the beginning of simulation. All output signals (***out_valid***, ***out_value***) should be reset after the reset is asserted.
5. The clock period is **5 ns**.
6. Your latency should be **less than 20 cycles** for each pattern. The latency of this design is defined as the clock cycles between the falling edge of the last cycle of ***in_valid*** and the rising edge of the ***out_valid***.
7. All input signals are synchronized at **the negative edge** of the clock.
8. When ***in_valid*** is low, all other input signals (***matrix***, ***matrix_size***) are tied to the unknown state.
9. ***out_valid*** is limited to be high for only **3 or 7 cycles**, respectively.
10. ***out_valid*** should not be raised when ***in_valid*** is high.
11. The next input pattern will come in **1~2 negative edge of clock** after your ***out_valid*** falls.
12. All output signals (***out_valid***, ***out_value***) should be synchronized at **clock positive edge**.
13. The TA's pattern will capture your output for checking at **clock negative edge**.
14. The ***out_value*** should be correct when ***out_valid*** is high.
15. The ***out_value*** should be zero when ***out_valid*** is low.
16. The synthesis result (syn.log) of data type cannot include any **latches and errors**.
17. The synthesis time should be less than **1 hour**.
18. The synthesis area should be less than **1000,000**
19. After synthesis, you can check **SS.area** and **SS.timing**. The area report is valid only when the slack in the end of the timing report should be **non-negative** and the result should be **MET**.

File Uploading

1. Deadline :

   1st demo : 6/2 15:30:00

   2nd demo : 6/2 23:59:00

2. Upload :

- **Please Upload your design file to E3, and name it as "SS_dcsxxx.sv".**
- **Failure to comply with the naming rule will result in a non-grading evaluation.**

Grading policy

1. If you pass the 1st demo(pass both RTL and GATE-level simulation ; timing slack is **MET** in 02_SYN ; there's no latch in SYN.log), you can get a score of 100.
2. If you pass the 2nd demo(pass both RTL and GATE-level simulation ; timing slack is **MET** in 02_SYN ; there's no latch in SYN.log), you can get a score of 80.
3. There's **NO** 3rd demo for the online test.
4. **Failure to comply with the naming rule will result in a non-grading evaluation.**

Note

Template folders and reference commands:

1. 01_RTL/ (RTL simulation) → **./01_run**
2. 02_SYN/ (synthesis) → **./01_run_dc**
3. 03_GATE/ (gate-level simulation) → **./01_run**