



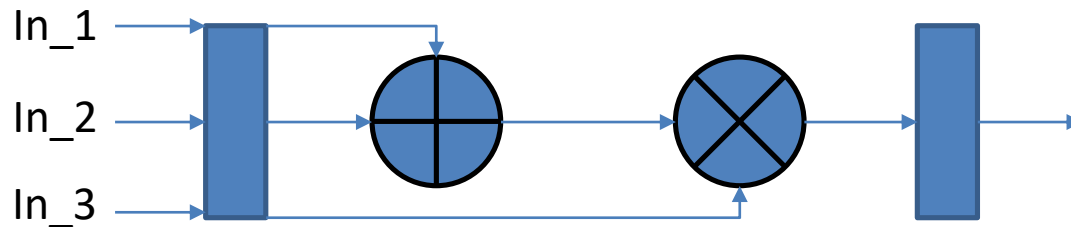
# 2019 DCS Lab 9

---

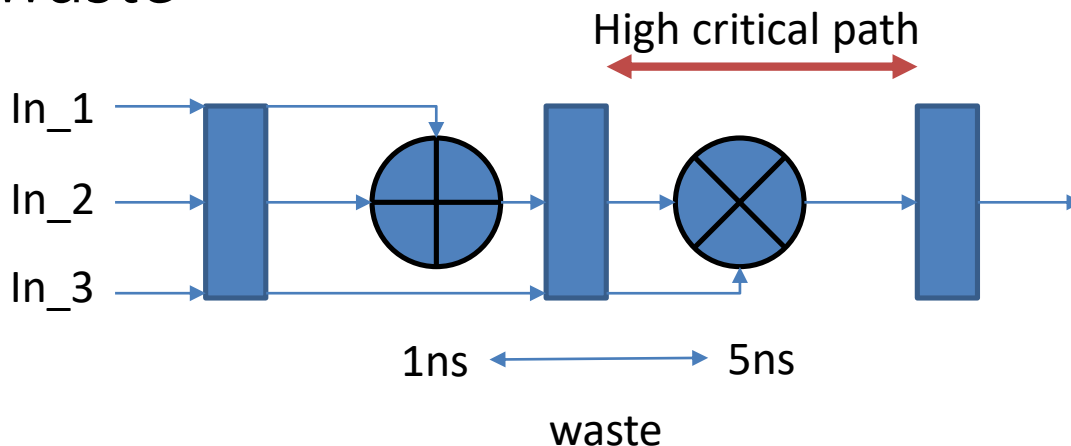
Unbalanced pipeline

# Pipeline

- No pipeline, long critical path

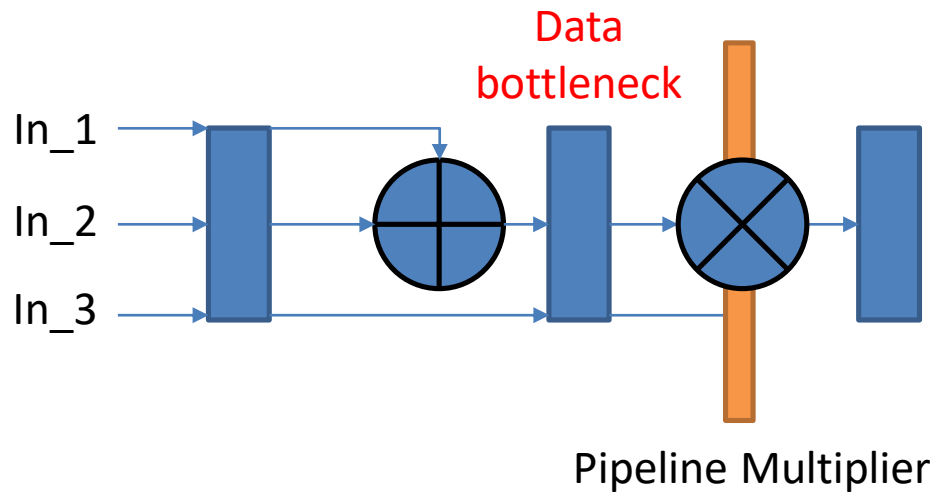


- Balanced pipeline, need higher clock period and waste



# Unbalanced pipeline

- Unbalanced pipeline, more complex, higher area, lower clock period.



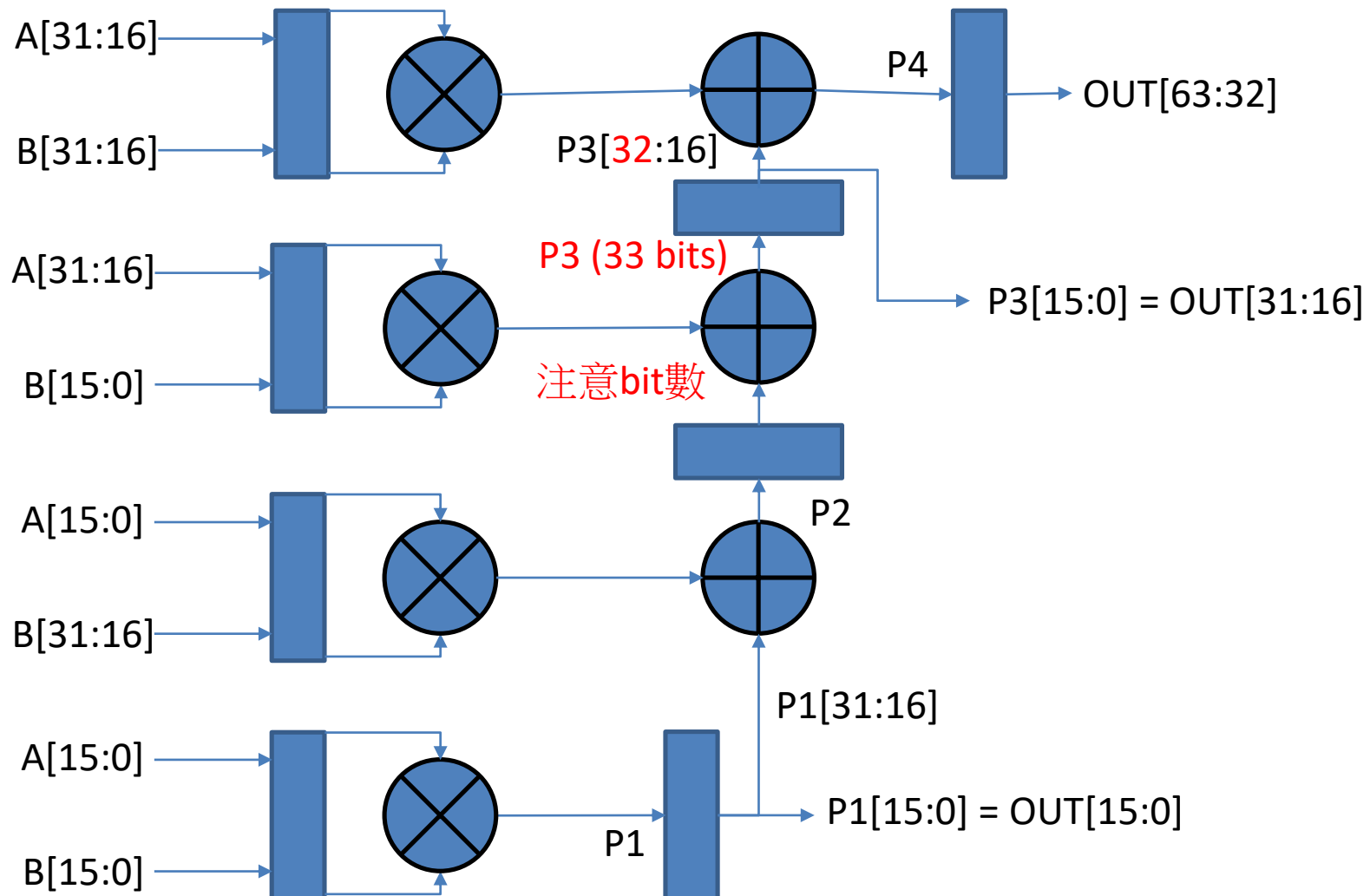
Fast to slow, solution (Reference: Ch 10 p23)

1. Stall
2. Split pipeline
3. Parallel copies
4. FIFO

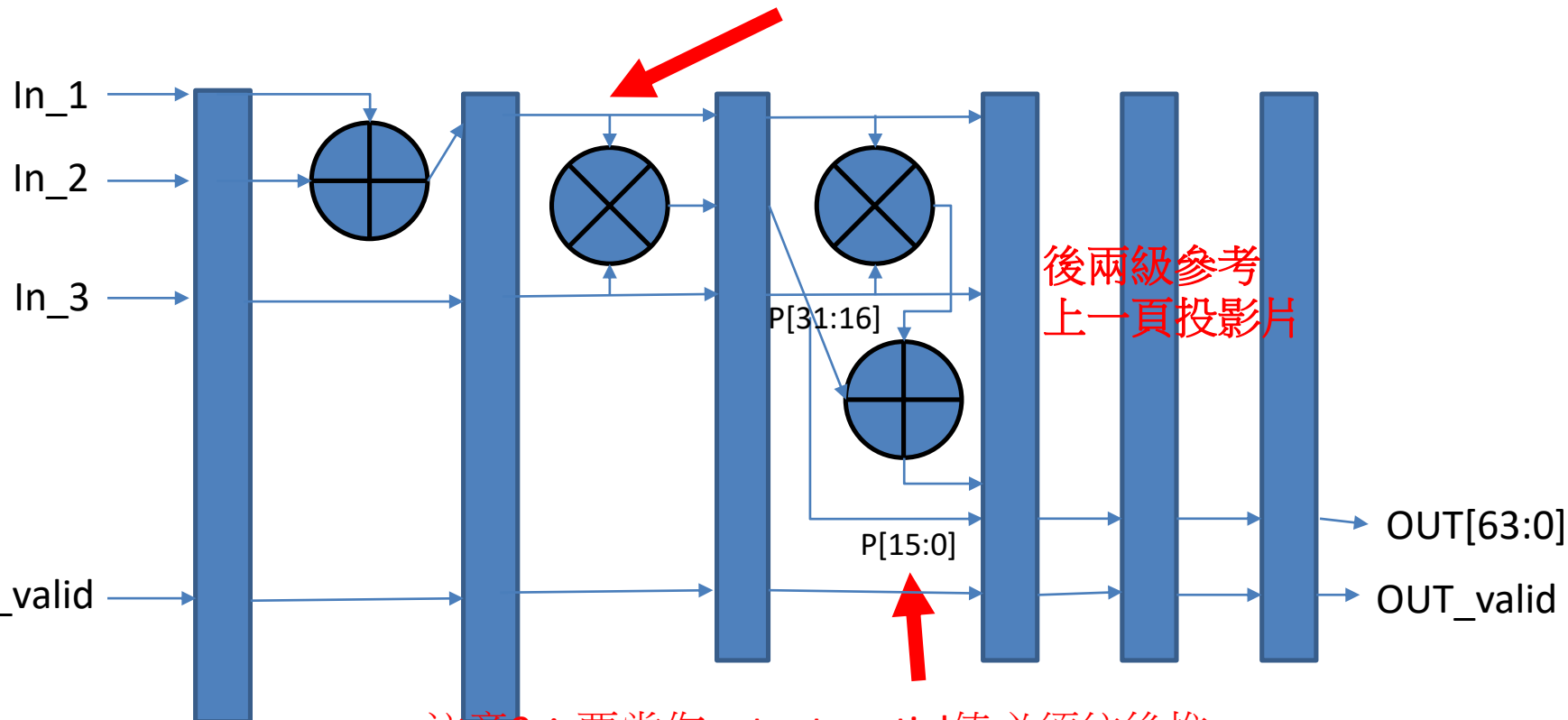
# You need to design

- $Out = (in\_1 + in\_2) \times in\_3$ 
  - in\_1, in\_2: 31 bits
  - in\_3: 32 bits
  - Out: 64 bits
- Pipeline Multiplier
  - $A[31:0] \times B[31:0]$ 
    - $= (A[31:16] \times 2^{16} + A[15:0]) \times (B[31:16] \times 2^{16} + B[15:0])$
    - $= A[31:16] \times B[31:16] \times 2^{32} + (A[31:16] \times B[15:0] + A[15:0] \times B[31:16]) \times 2^{16} + A[15:0] \times B[15:0]$

# Pipeline Multiplier



注意1：因為4級pipe乘法用同樣的input，所以要把FIFO值繼續往後推



注意2：要當作output partial值必須往後推，否則output會抓到錯誤的partial

# adv\_pipeline.sv

Input Signal	Bit Width	Definition
clk	1	Clock
rst_n	1	Asynchronous active-low reset
In_1	31	為連續資料，in_valid = 1時，同時給予三筆data，注意bit數
In_2	31	
In_3	32	
in_valid	1	當此訊號拉起時給in資料，連續20cycle。

Output Signal	Bit Width	Definition
out_valid	1	Out_valid可為不連續資料，當此訊號拉起時，testbench依照input data順序檢查。
Out	64	Out_valid=1時，輸出一筆64bits資料

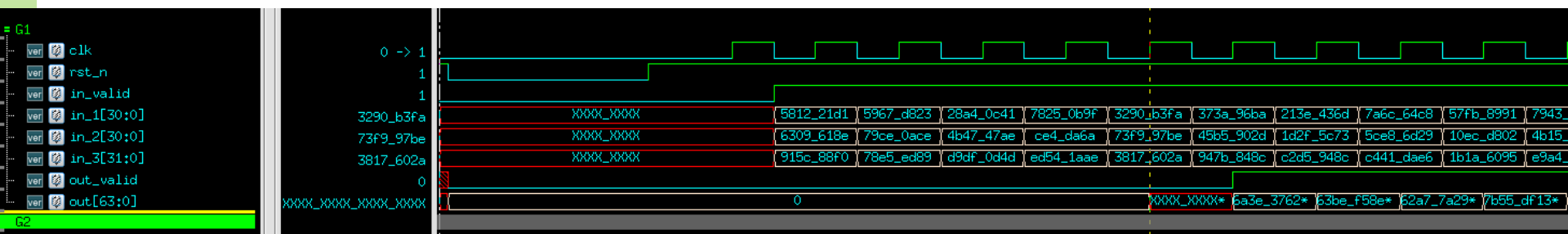
# Spec

- 可以使用各種方式balance pipeline，不需照建議架構寫design。
- output檢查順序不得更改。
- 所有output必須非同步負準位reset。
- 01\_RTL 需要PASS。
- 02\_SYN不能有error跟latches。
- 02\_SYN，clock period = 4.5ns，timing slack必須為MET。
- 注意1：使用建議架構較為輕鬆，請注意設計中的bit數。
- 注意2：乘法器critical path非常長，如不使用pipeline timing可能不會met。
- 注意3：數字非常大，不容易靠肉眼debug，請想清楚撰寫。



# Output & Waveform

- Waveform



# Command

- `tar -xvf ~dcsta01/Lab09.tar`
- Upload
  - `cd 09_upload`
  - `./01_upload`
  - `./02_download demoX`