Lab 7
pipeline

楊豐賓

Design a simple operation

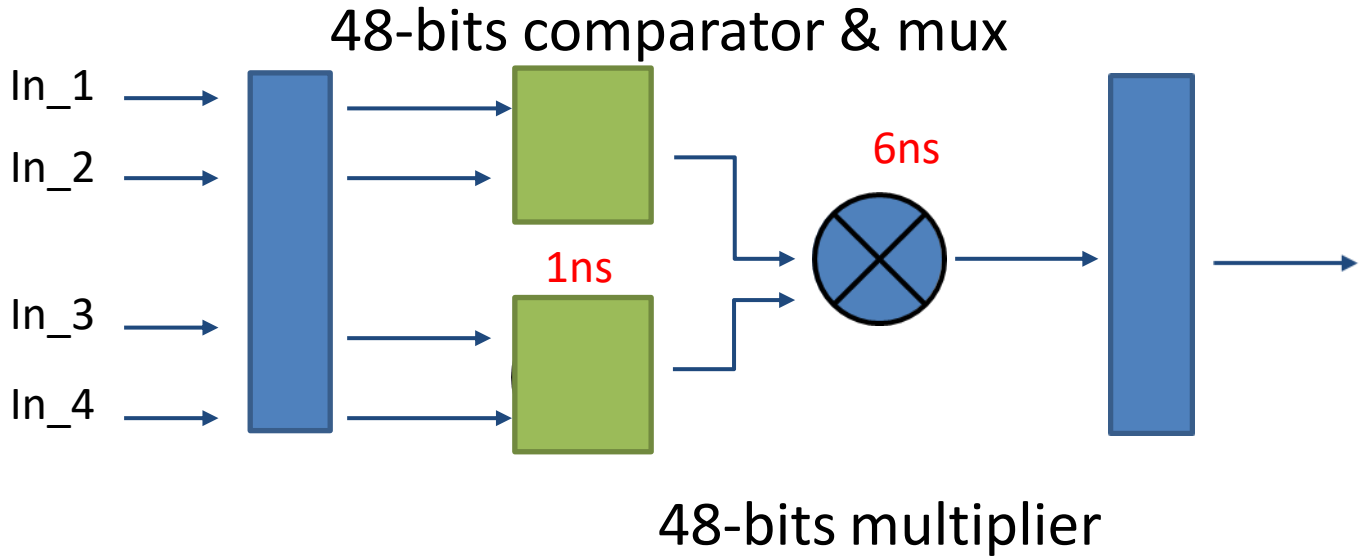
- **Out = max(in_1, in_2) * max(in_3, in_4)**
  - in_1, in_2, in_3, in_4: 48 bits unsigned number
  - Out: 96 bits unsigned number

- Large bit-width multiplier is costly

Naïve design

- No pipeline => long critical path

assume cycle time: 5ns



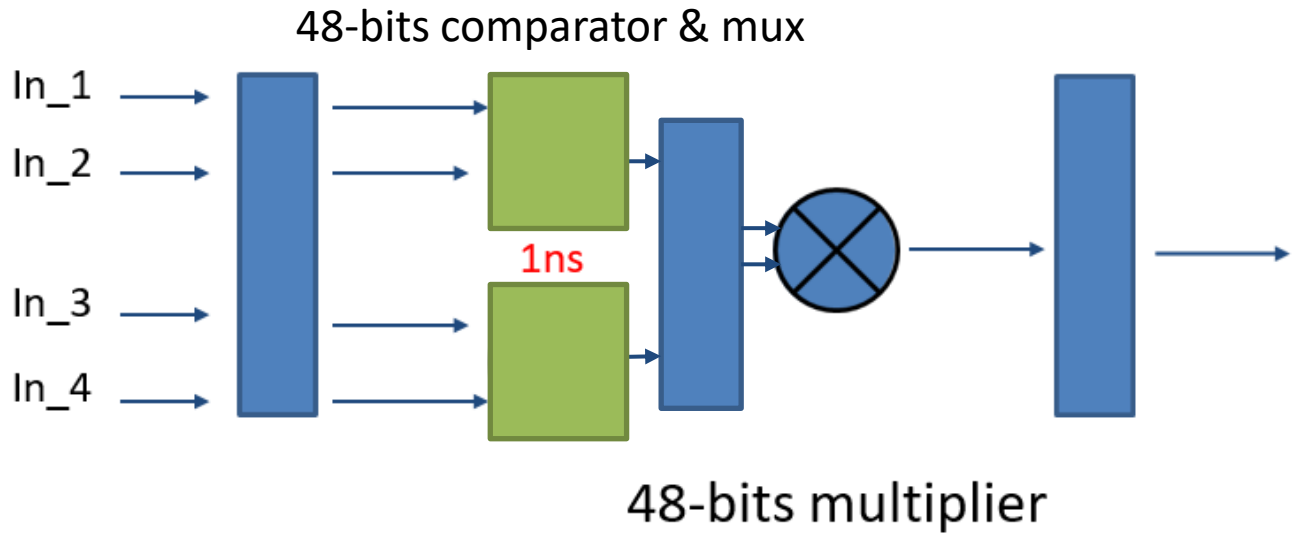
critical path = 48-bits comparator & mux + 48-bits multiplier = 7ns

> cycle time you will fail !!

Simple pipeline design

- Simply separate adder and multiplier

assume cycle time: 5ns



critical path = max(48-bits comparator & mux, 48-bits multiplier) = 6ns > cycle time

- Beats naïve design, but still faces negative slack

Can we do better?

- Let's go back to elementary school
- 25 * 68 = ?
  - P1 = 8*5=40
  - P2 = 8*2=16
  - P3 = 6*5=30
  - P4 = 6*2=12

out = P1 + P2 * 10 + P3 * 10 + P4 * 100

			2	5
	*		6	8
P1	=		4	0
P2	+	1	6	0
P3	+	3	0	0
P4	+	1	2	0	0
out		1	7	0	0

For our binary case

- Example: Partition operands A and B into two parts
  - A[47:0] * B[47:0]

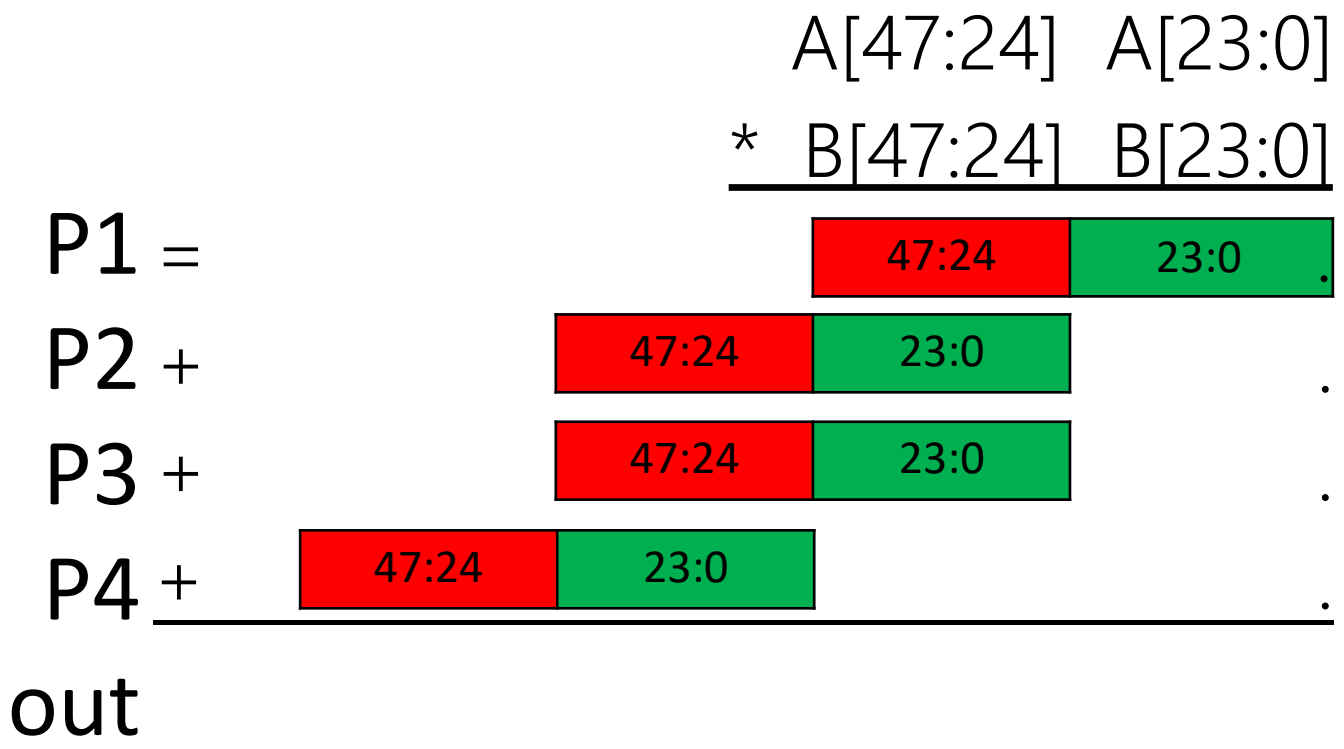
    =(A[47:24] * 2^24 + A[23:0]) * (B[47:24] * 2^24 + B[23:0])

    = A[47:24] * B[47:24] * 2^48

    + A[47:24] * B[23:0] * 2^24

    + B[47:24] * A[23:0] * 2^24

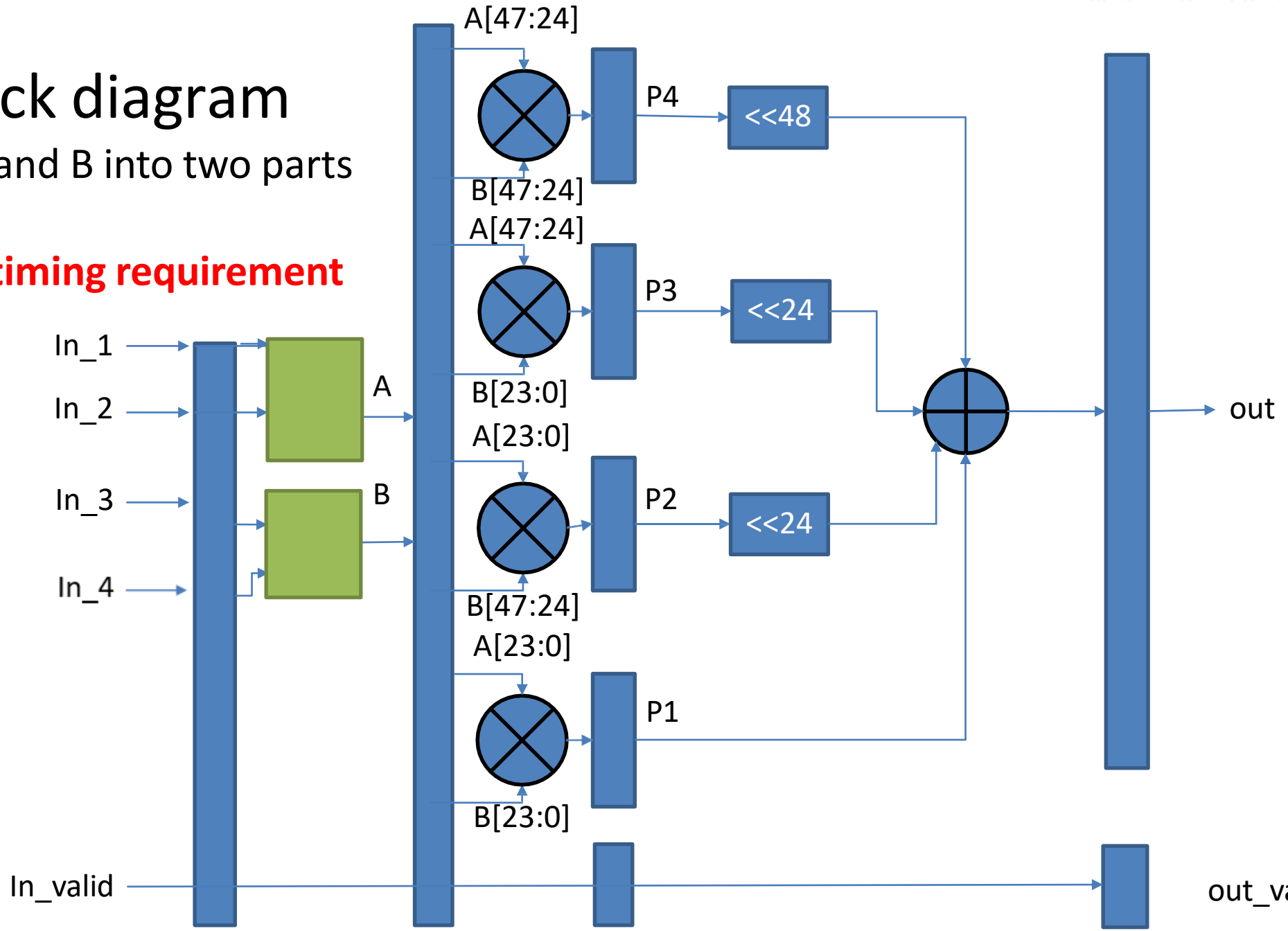
    + A[23:0] * B[23:0]



Example block diagram

For partitioning A and B into two parts

※You won't pass timing requirement using this design



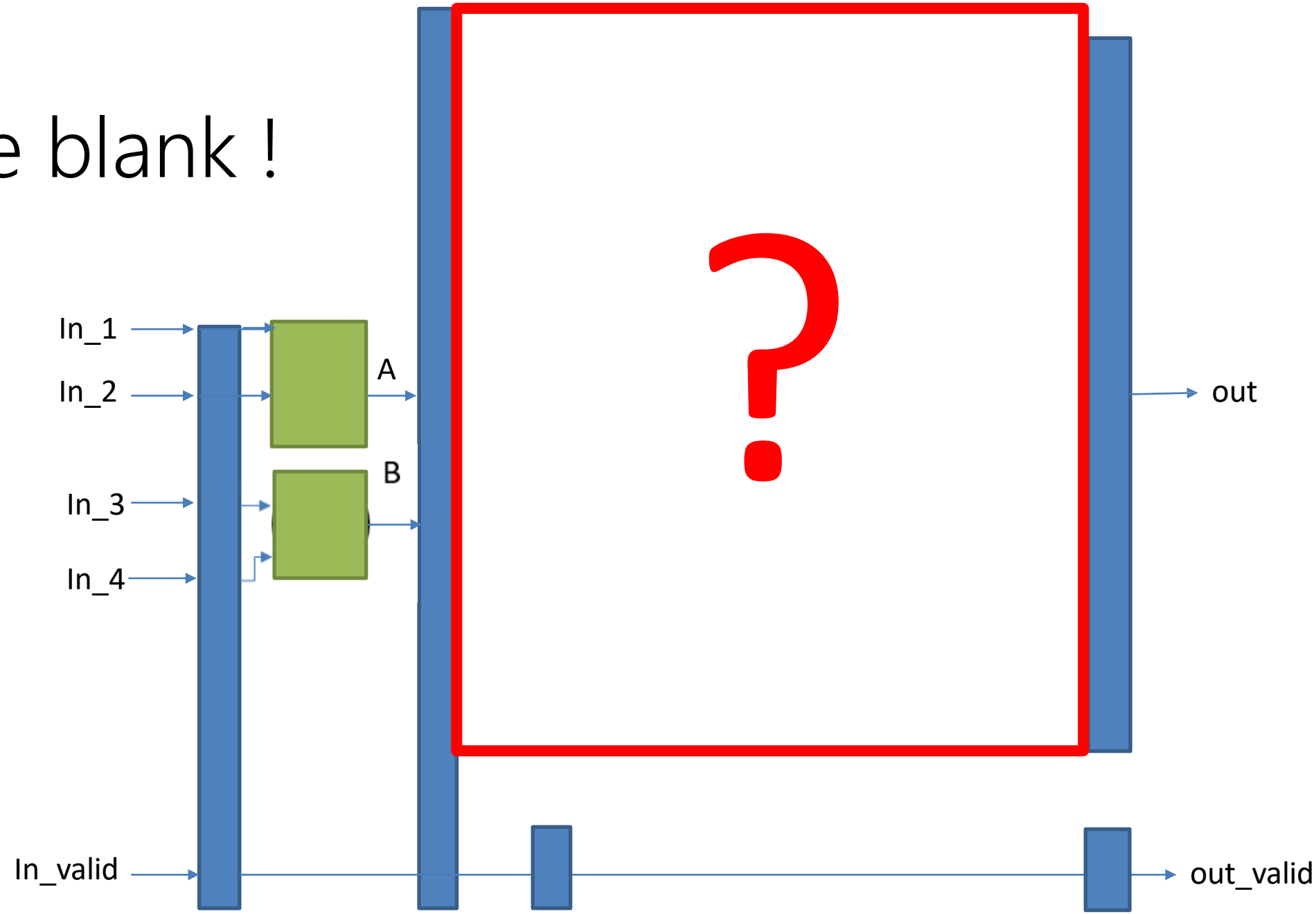
But…

- The critical path of the example block diagram is 24-bit multiplier
- What if that's still not enough ?
  - Further partition the combinational circuit on the critical path

- What you have to do in this lab
  - Refer to the example block diagram in the last page
  - Design a pipeline that partitions the multiplication of operands A and B into **three parts**

※ You won't pass timing requirement if you only follow example block diagram

Fill in the blank !



P_MUL.sv

Input Signal	Bit Width	Definition
clk	1	clock
rst_n	1	Asynchronous active-low reset
in_1	48	unsigned inputs
in_2	48	unsigned inputs
in_3	48	unsigned inputs
in_4	48	unsigned inputs
in_valid	1	in_valid = 1 indicates one valid input will be high for 1000 continuous cycles

in_1, in_2 in_3 and in_4 will be given in one cycle

Output Signal	Bit Width	Definition
out_valid	1	out_valid = 1 indicates one valid output can be discontinuous
out	96	unsigned output

Specs

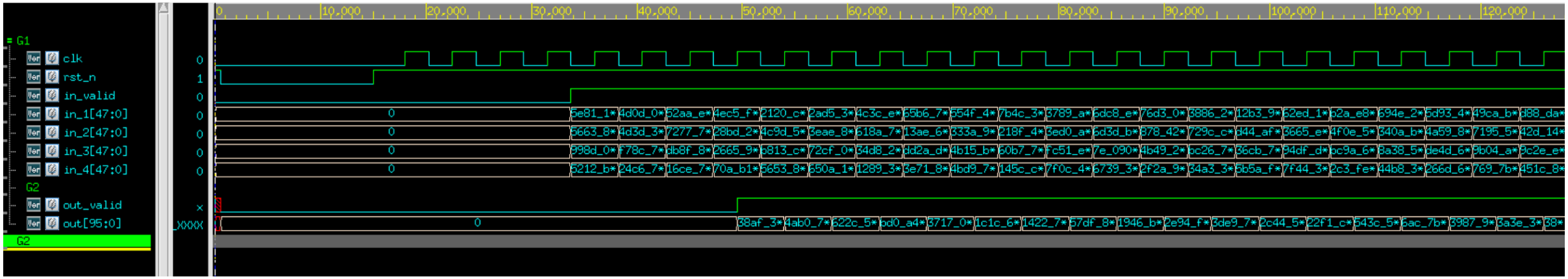
- Input and output signals are unsigned
- All output ports have to be reset to 0
- 01_RTL PASS
- 02_SYN clock period = 4.5ns, timing slack must be MET, no error and latch
- 03_GATE PASS, no error and timing violation

Something you should know

- The critical path of large bit-width multiplier is long, you must use pipeline
- Input and output numbers are extremely big, so debugging by nWave is hard
- **Longer critical path will induce longer synthesis time**
- It's simpler to refer to the provided block diagram, but you can try to design your own pipeline. Just Think twice before writing your code !

Output & Waveform

- Waveform



in_valid will be high for 1000 continuous cycles
out_valid should be high for 1000 cycles, but it can be discontinuous

Command

- tar -xvf ~dcsTA01/Lab07.tar
- Upload
  - cd 09_upload
  - ./01_upload
  - ./02_download demoX

DEMO1: 4/25 17:30:00

DEMO2: 4/26 23:59:59