



Lab 9 pipeline

陳柏瑤

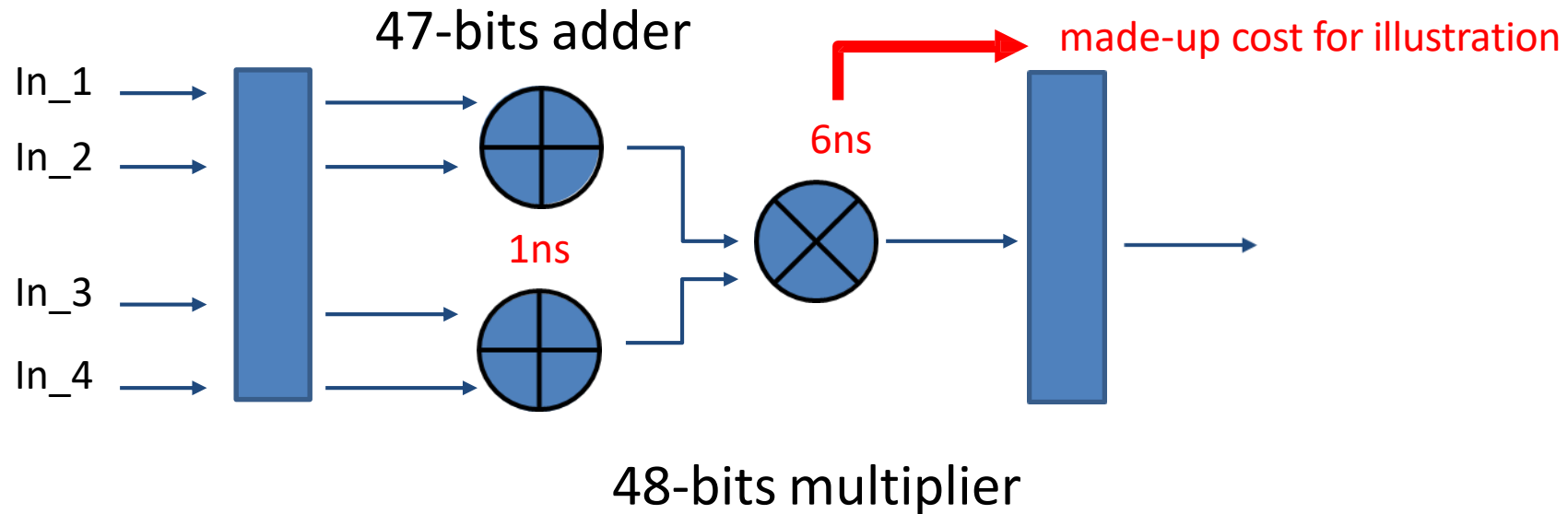
Design a simple operation

- ***Out*** = (***in_1*** + ***in_2***) × (***in_3*** + ***in_4***)
 - **in_1, in_2, in_3, in_4**: 47 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

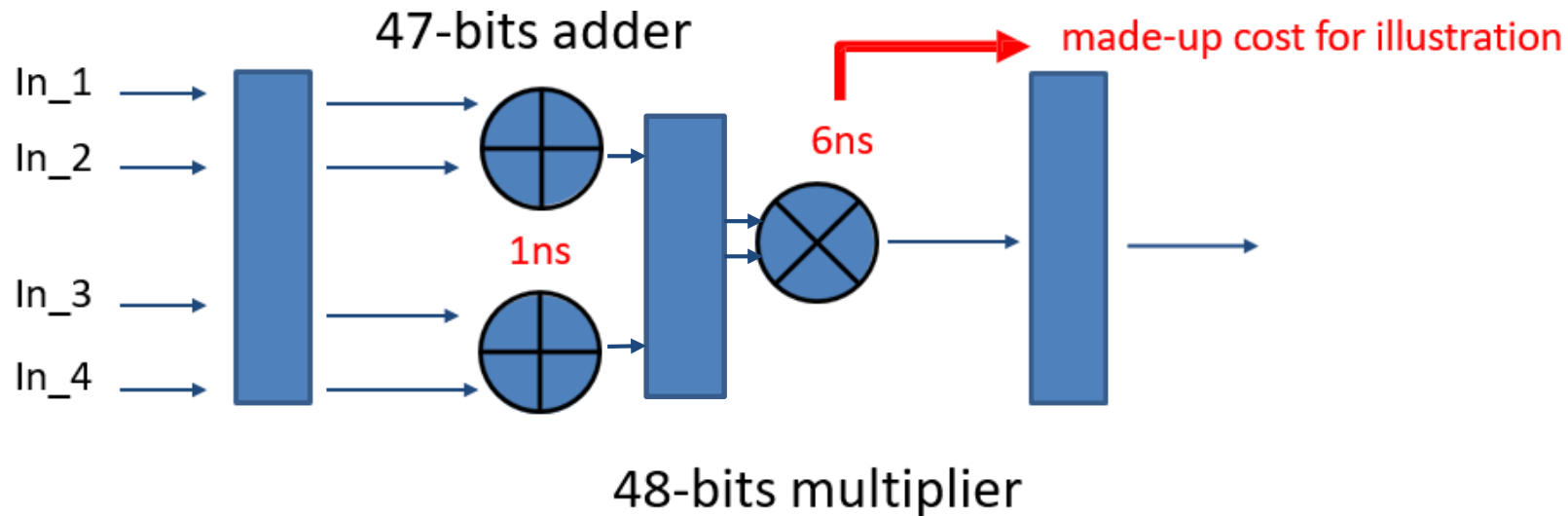


critical path = 47-bits adder + 48-bits multiplier = 7ns > cycle time
you will fail !!

Simple pipeline design

assume cycle time: 5ns

- Simply separate adder and multiplier



critical path = $\max(47\text{-bits adder}, 48\text{-bits multiplier}) = 6\text{ns} > \text{cycle time}$

- Beats naïve design, but still faces negative slack

For our binary case

- Example: Partition operands A and B into two parts

$$- A[47:0] * B[47:0]$$

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

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

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

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

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

P1 =

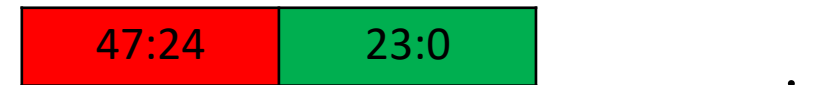
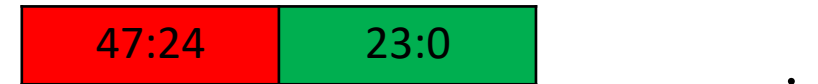
P2 +

P3 +

P4 +

out

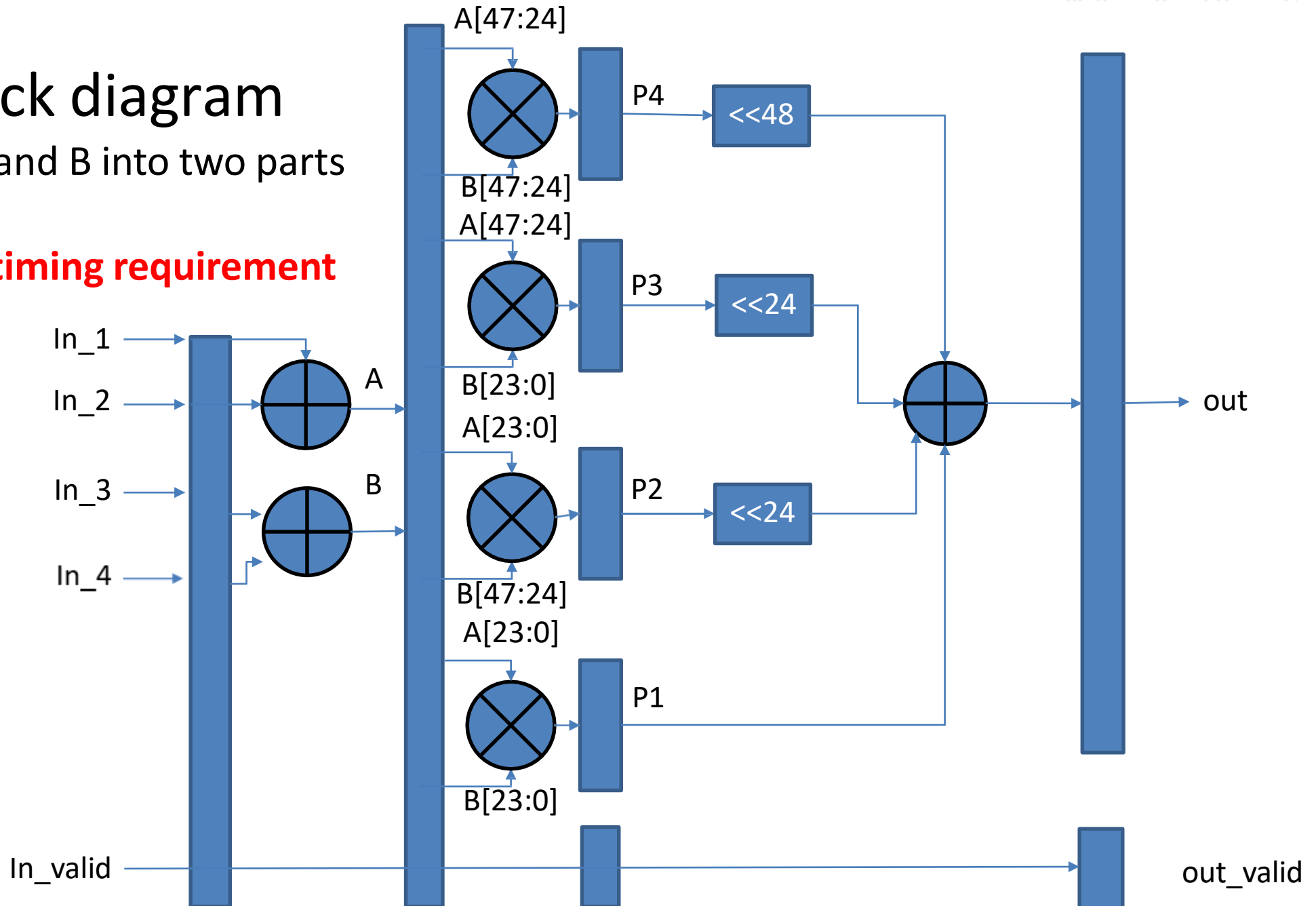
$$\begin{array}{r} A[47:24] \quad A[23:0] \\ * \quad B[47:24] \quad B[23:0] \\ \hline \end{array}$$



Example block diagram

For partitioning A and B into two parts

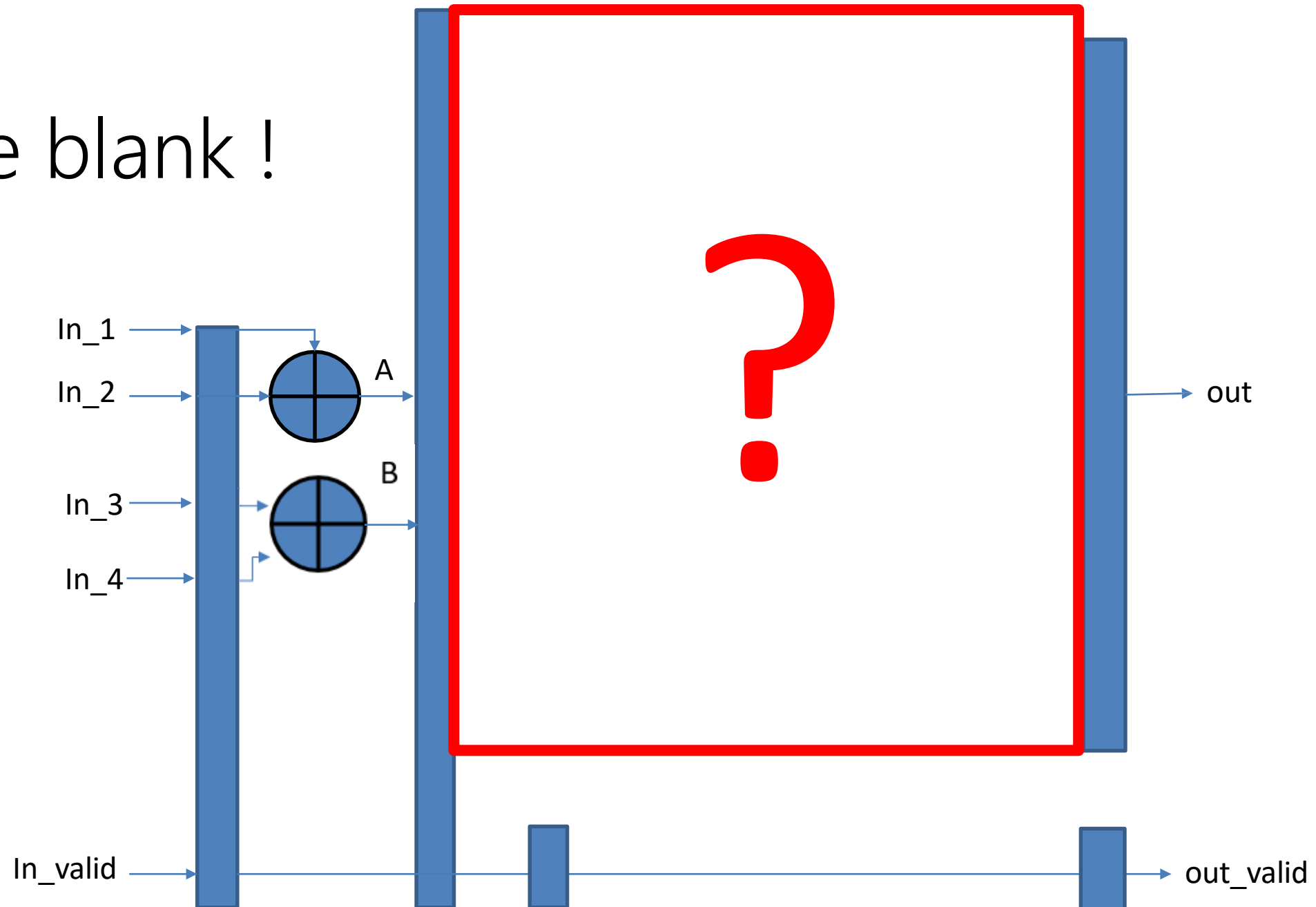
✗ You won't pass timing requirement using this design



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	47	unsigned inputs
in_2	47	unsigned inputs
in_3	47	unsigned inputs
in_4	47	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,
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

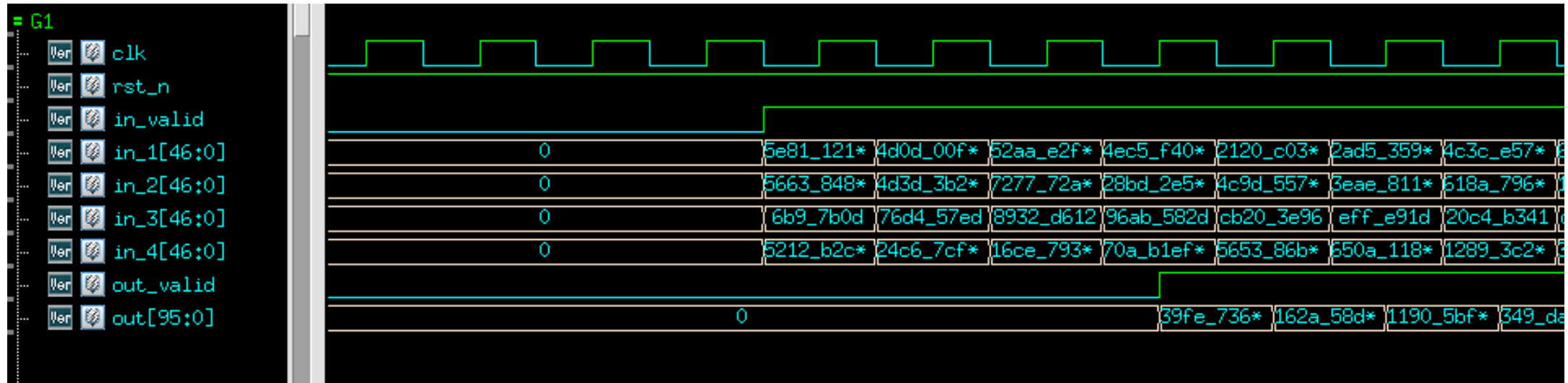
- Input and output signals are **unsigned**
- All output ports have to be reset to **0**
- 01_RTL **PASS**
- 02_SYN clock period = **4.5ns**, timing slack must be **MET**, no **error** and **latch**
- 03_GATE **PASS**, no **error** and **timing violation**

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/Lab09.tar`
- Upload
 - `cd 09_upload`
 - `./01_upload`
 - `./02_download demoX`
- Separate combinational and sequential blocks

DEMO1: 5/11 16:20:00

DEMO2: 5/11 23:59:59