COMS12200
Introduction to
Computer Architecture

Simon Hollis (simon@cs.bris.ac.uk)

COMS12200 Part 5 – Simon Hollis

CLOCKED EXECUTION

Review of control

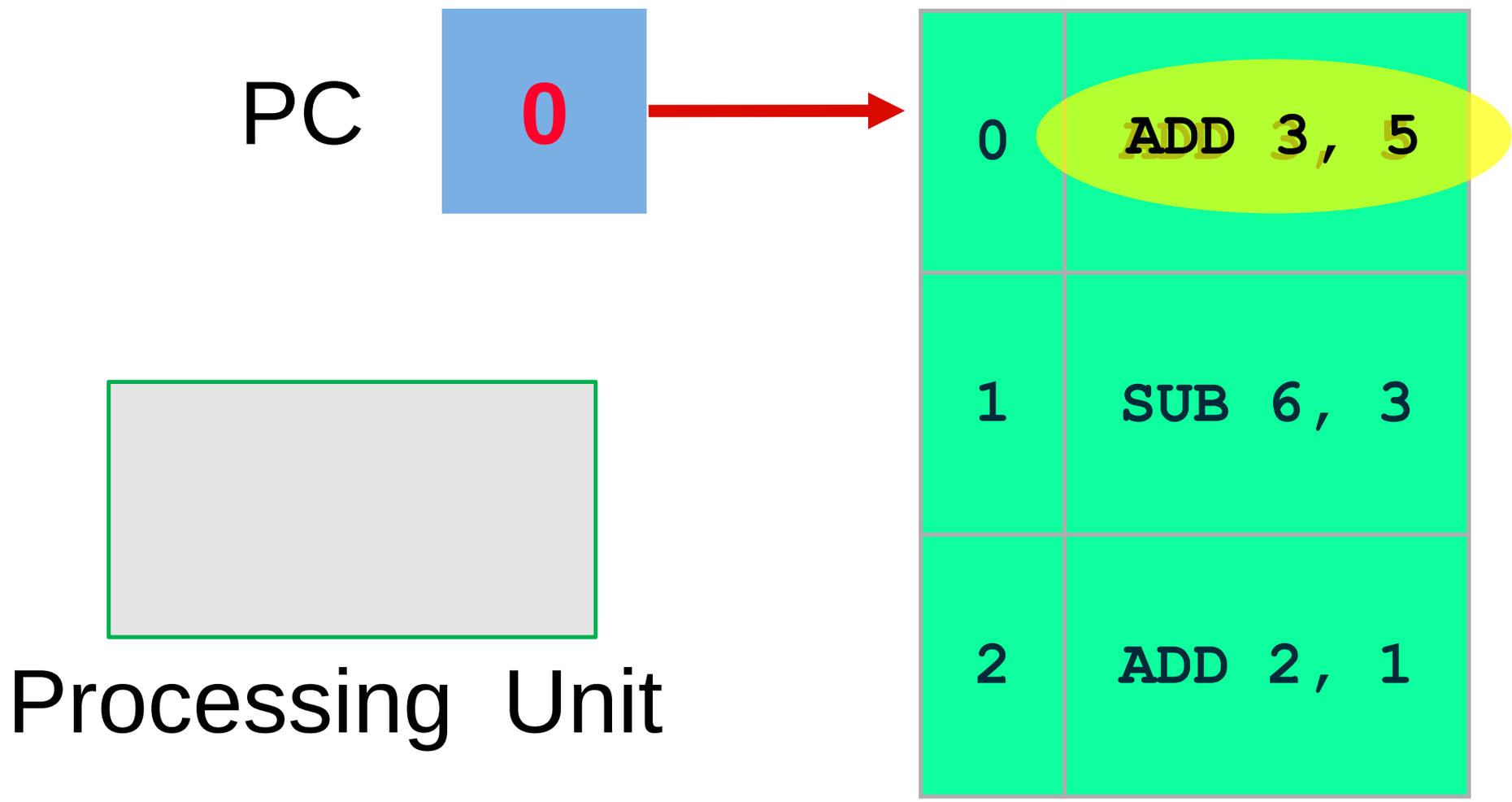
- We’ve seen in labs the need for a control path to orchestrate the flow of data through (simple) processing systems.
- e.g. we’ve hand-set the enable inputs of Register components and set the ALU functionalities.
- You may by now be just a little tired of flicking switches to make things happen in the correct sequence.

Time as a concept

- If you think about it, you have been using **time** to cause what you want to happen, when you want it to happen.
- e.g. one computation *after* another.

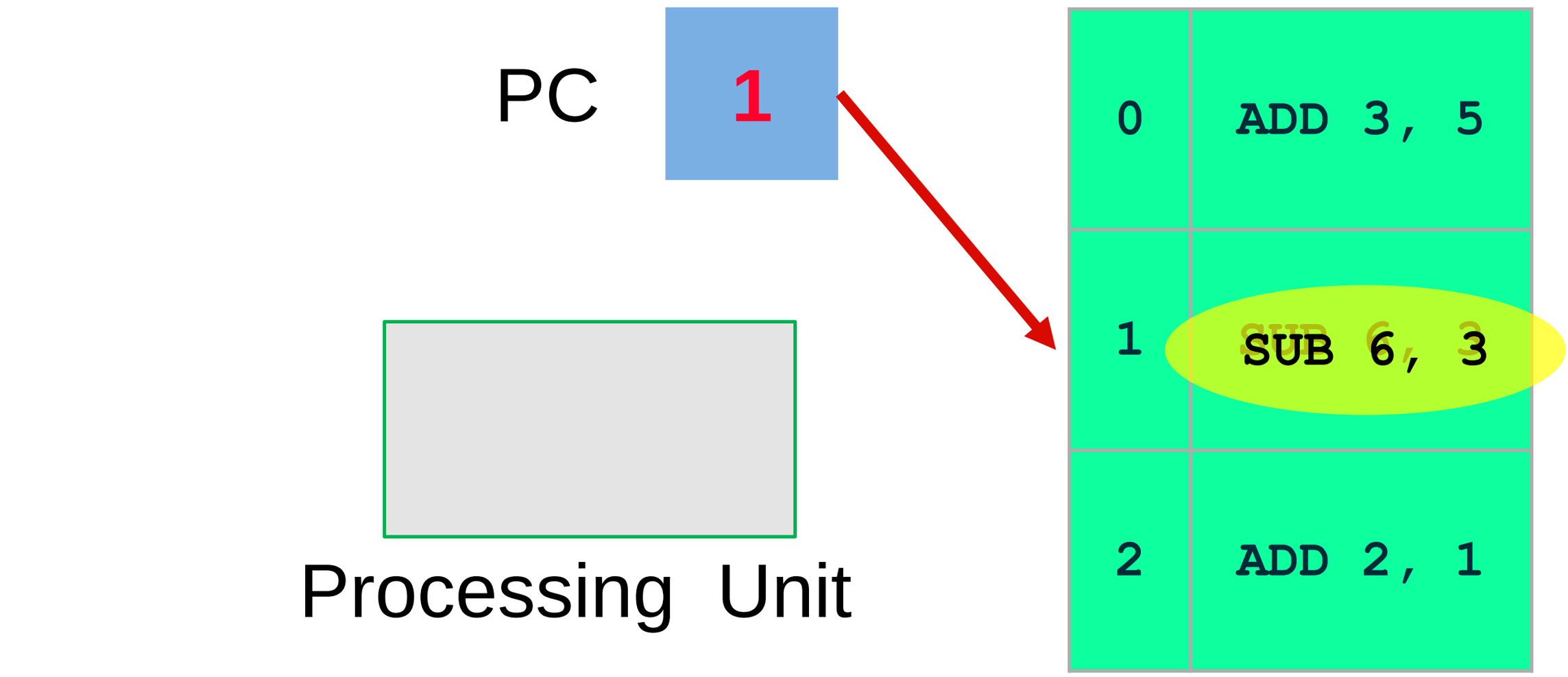
Recall PC Operation

- Under normal program flow, the PC increments by one instruction address per instruction fetched.



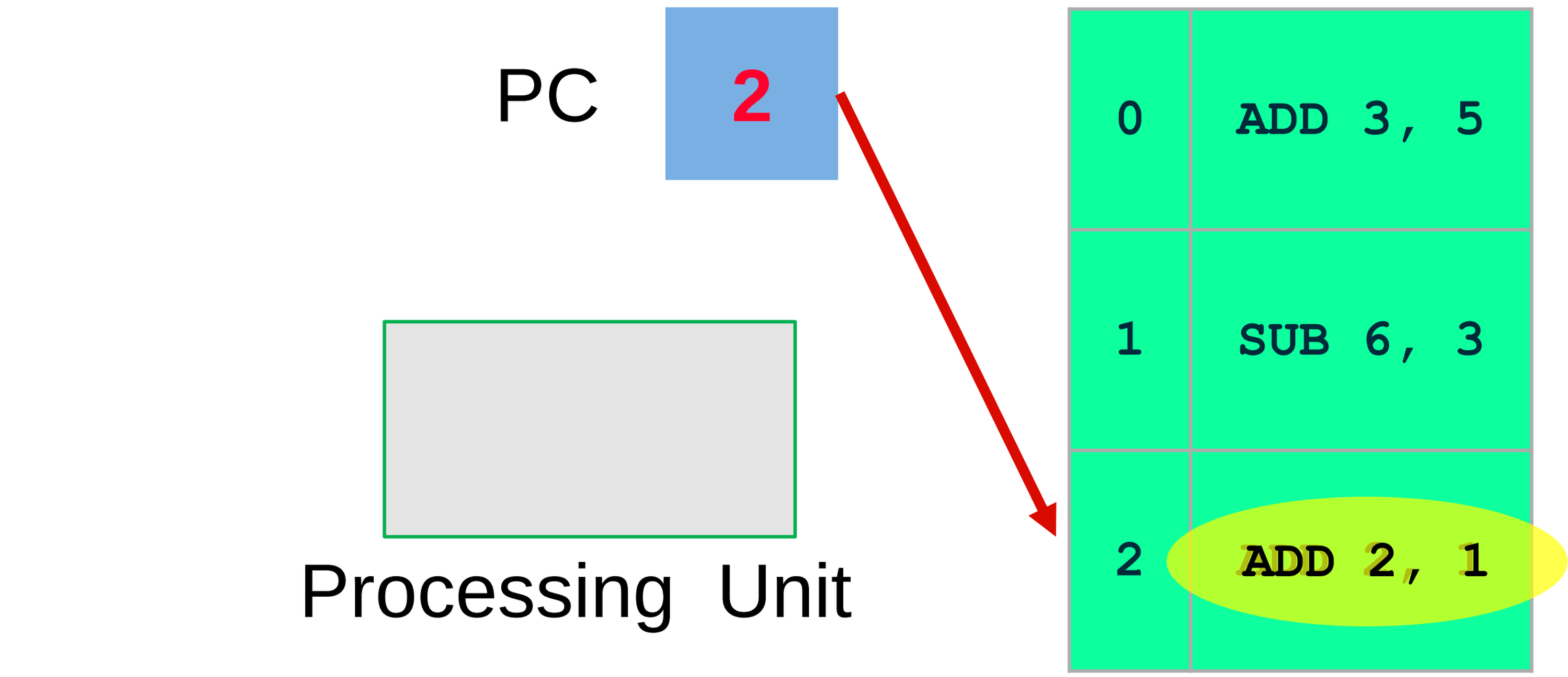
Recall PC Operation

- Under normal program flow, the PC increments by one instruction address per instruction fetched.



Recall PC Operation

- Under normal program flow, the PC increments by one instruction address per instruction fetched.



Timeline example

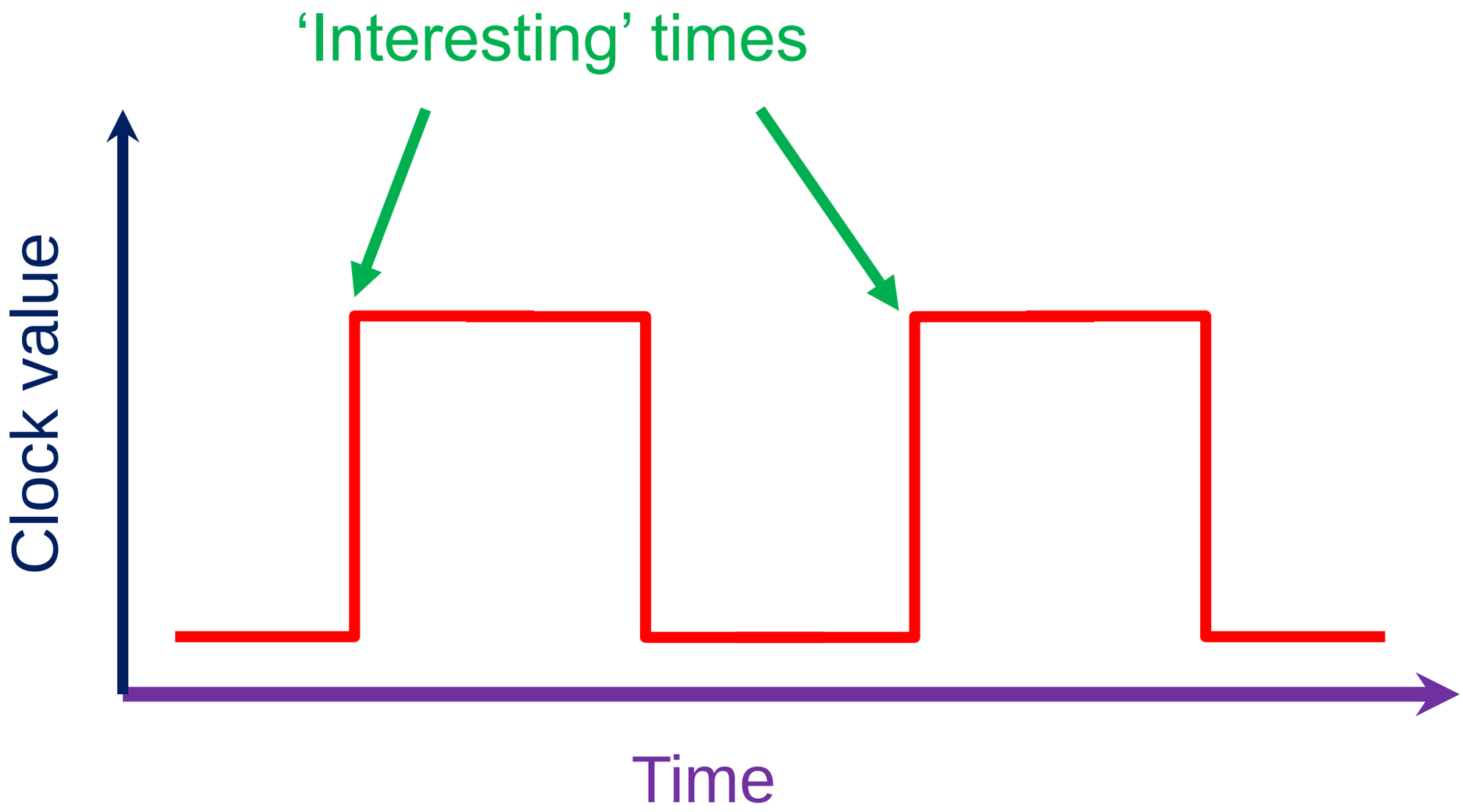
We could think about this another way:

Time	PC value	Operation
0s	0	ADD 3, 5
1s	1	SUB 6, 3
2s	2	ADD 2, 1

So, if we could track time, we could make the right things happen in the right order

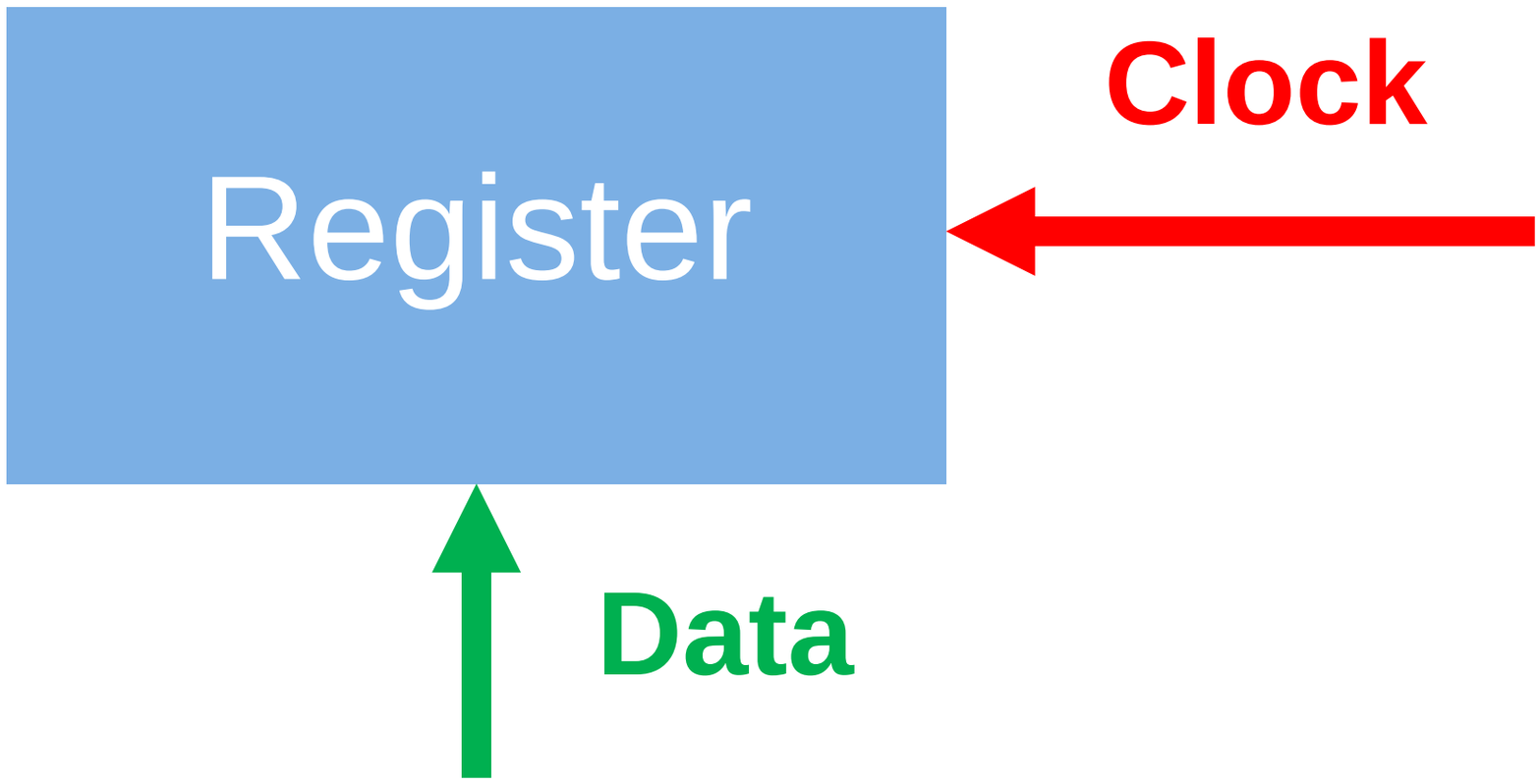
A *clock*

Fundamentally, a clock delineates line.



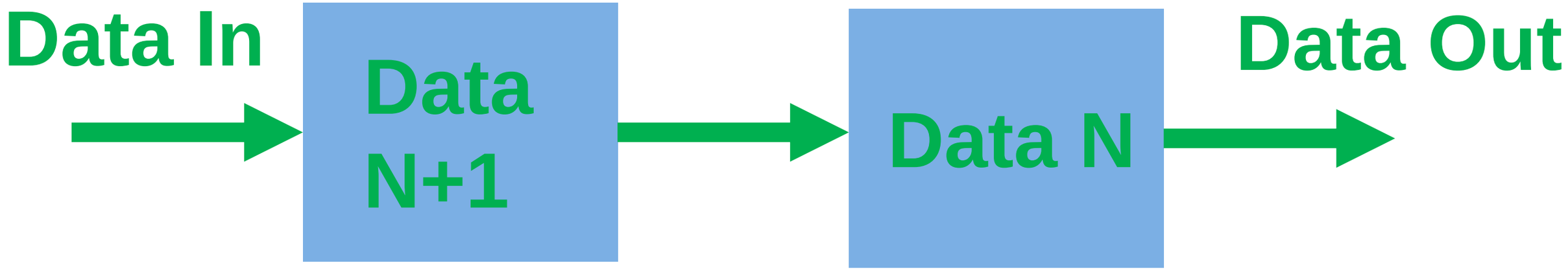
A clock can help

We can use the clock to automate control of our data path.



A FIFO

Imagine that we want to build a FIFO data structure (First-In, First-Out).



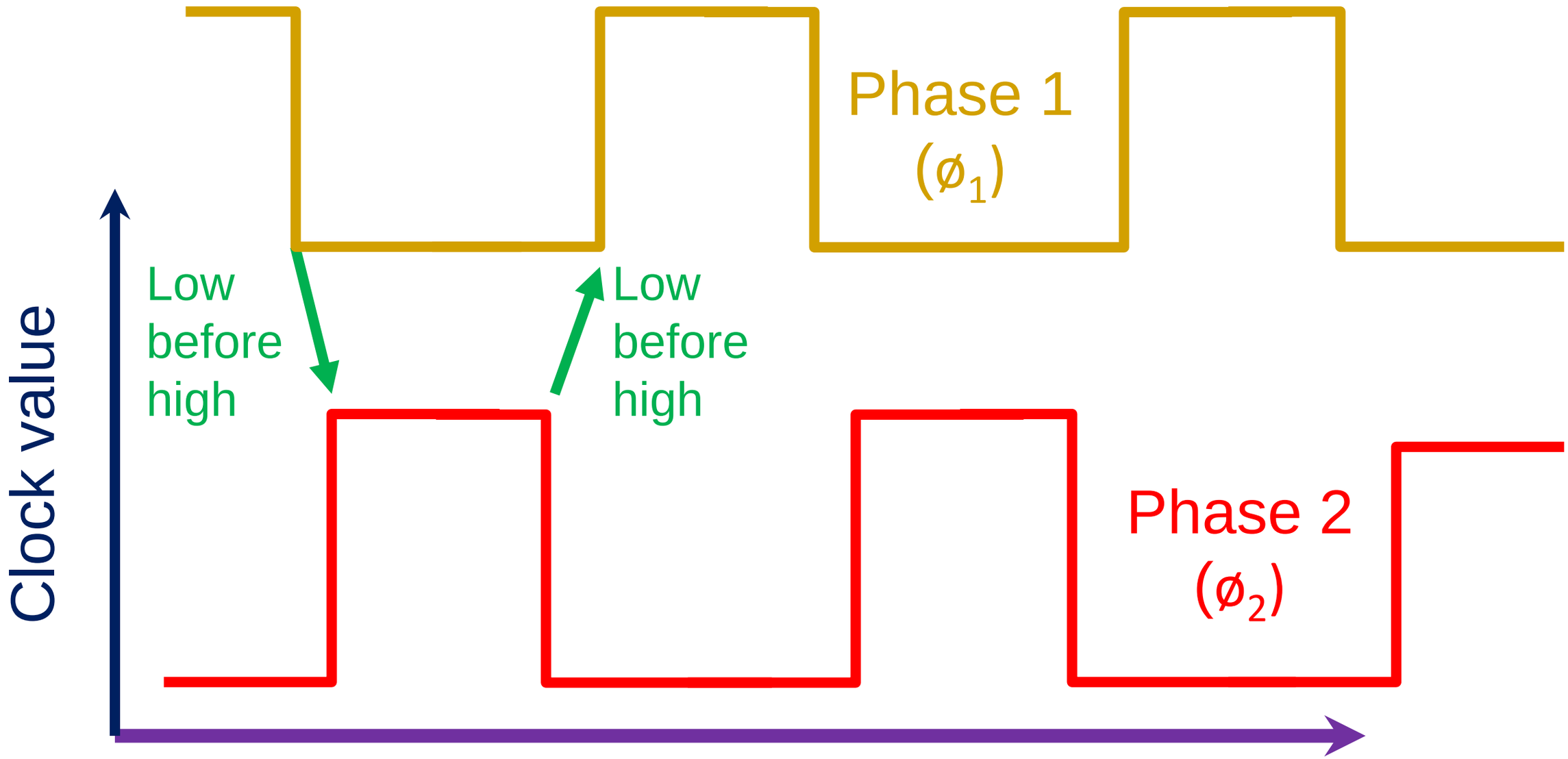
1. How would you build this by hand?
2. How could a clock help build one?

The *two-phase* clock

- Consider the case where we want two things to happen at two different times.
- To do this easily, we need more than just our single clock signal.
- An easy-to-implement approach uses two clocks, running at the same speed, but out of phase to give two disjoint clock pulses.

The two-phase clock

A non-overlapping clock works best



Use of two phases

If we treat the two phases as two separate enable signals for some logic, we can see how they can allow for lock-step sequencing.

1. FIFO and two-phase clocks
2. ALU paths

Build-a-comp clocks

- We have a Clock module as part of the Build-a-comp kit.
- It produces a two-phase clock just as shown on a previous slide.
- The frequency is tuneable, but the phase relationship is fixed.

Using the Build-a-comp clock

- Build-a-comp clock demonstration
- Building a FIFO with build-a-comp kit

Summary

- Clocks allow for automation of time-based control signals.
- Two common implementations are single phase and two-phase-non-overlapping.
- The second type allows easy building of computational data flows.
- We’ve seen the Build-a-comp implementation.