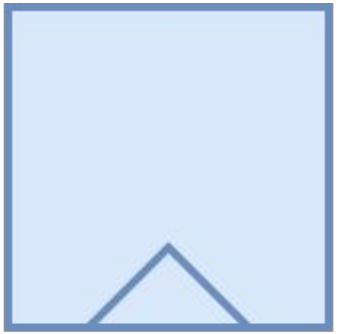
Latency Counting in the SUS Language

Lennart Van Hirtum
PC2 - Paderborn University

Prof. Christian Plessl
PC2 - Paderborn University

LATTE'24

Registers

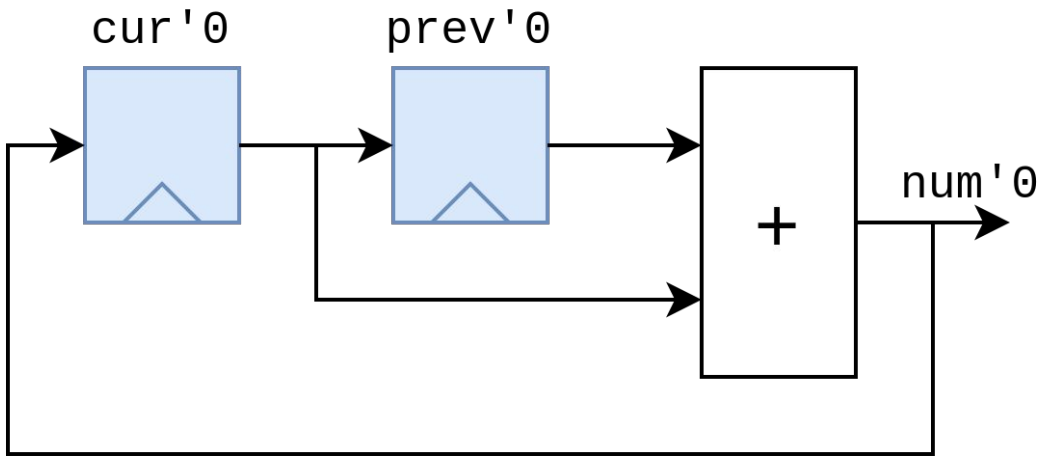


State

Latency

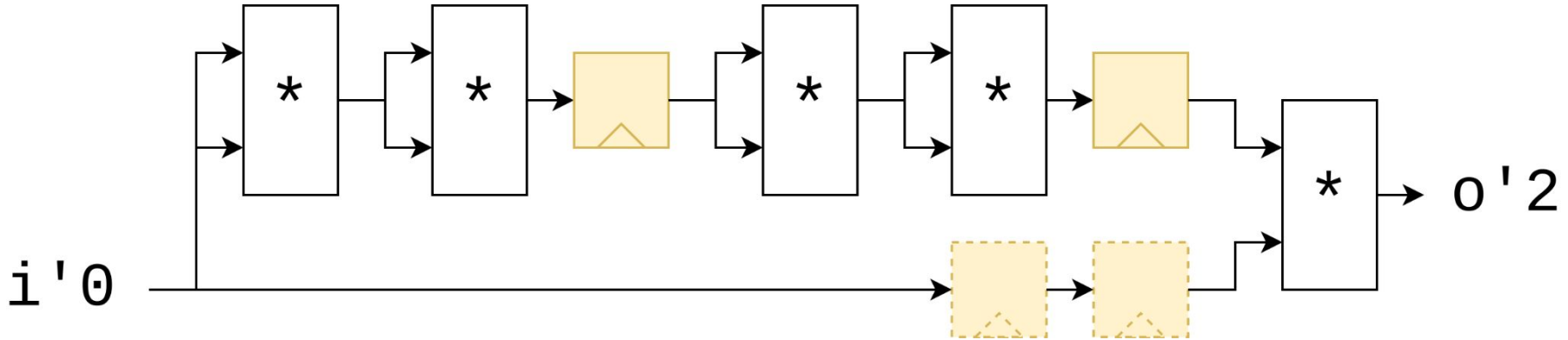
State registers

```
module fibonnaci : -> int num {
    state int cur = 1
    state int prev = 0

    num = cur + prev
    prev = cur
    cur = num
}
```



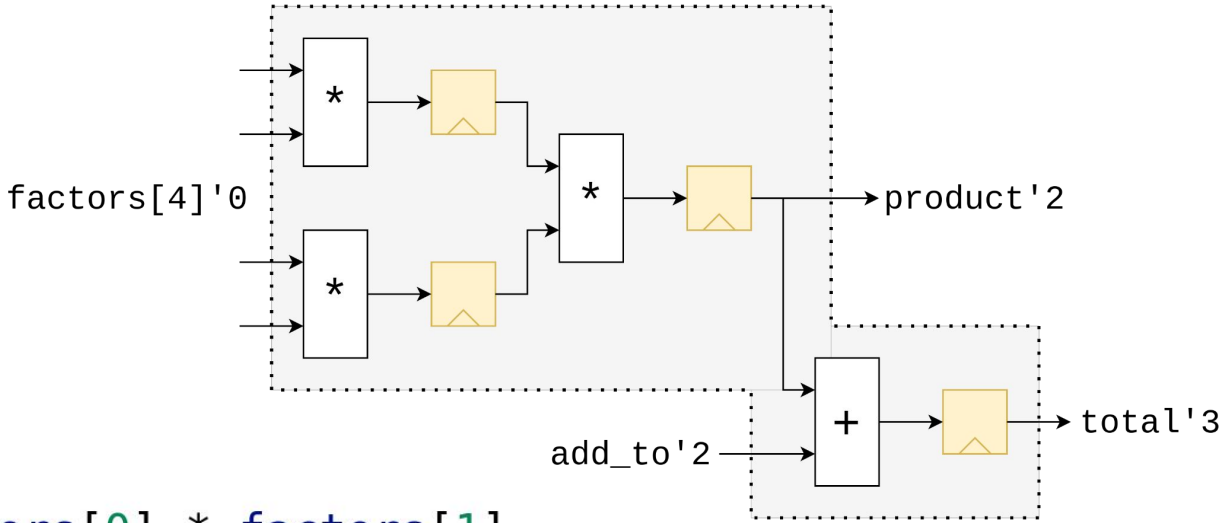
Latency Registers

```
module pow17 : int i -> int o {
        int i2  = i   * i
    reg int i4  = i2  * i2
        int i8  = i4  * i4
    reg int i16 = i8  * i8
            o   = i16 * i
}
```



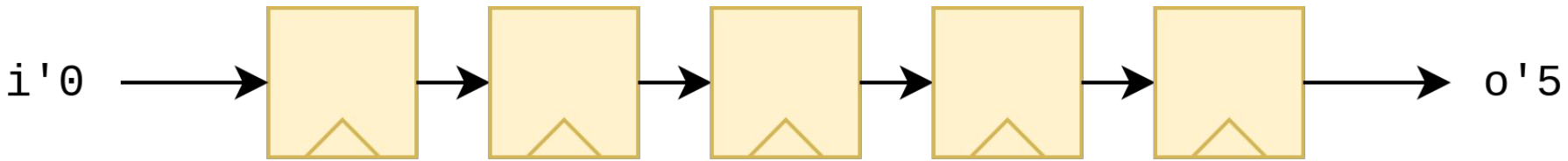
Port Latencies

```
module example_md :
    int[4] factors,
    int add_to ->
    int product,
    int total {

    reg int mul0 = factors[0] * factors[1]
    reg int mul1 = factors[2] * factors[3]

    reg product = mul0 * mul1
    reg total = product + add_to
}
```

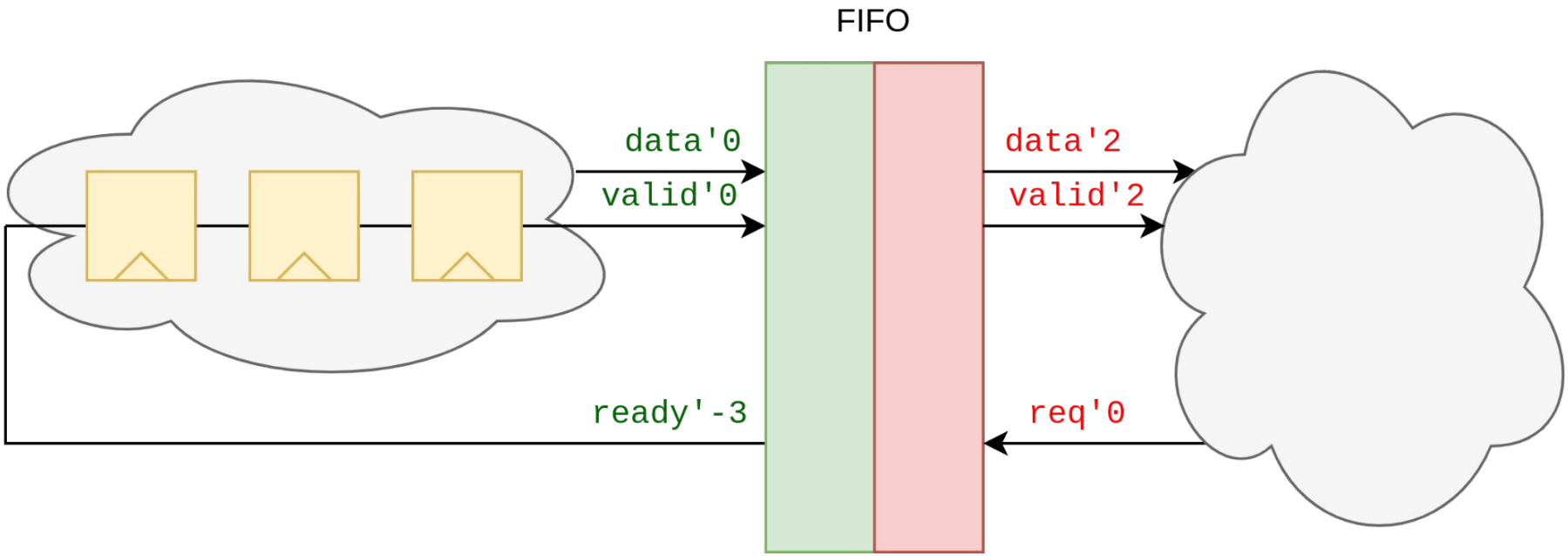


Latency Annotations

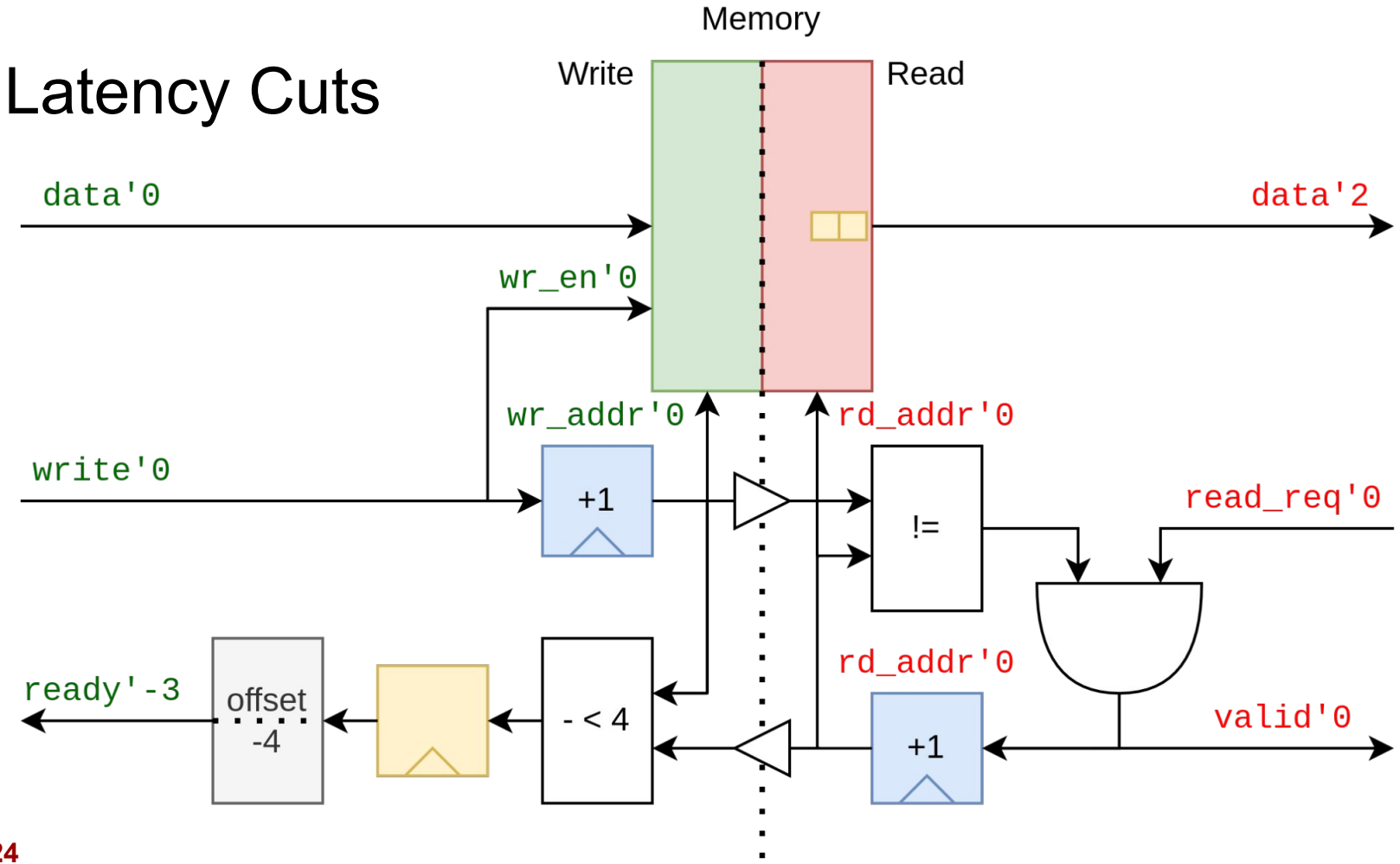
```
module module_taking_time :
    int i'0 -> int o'5 {
    o = i
}
```



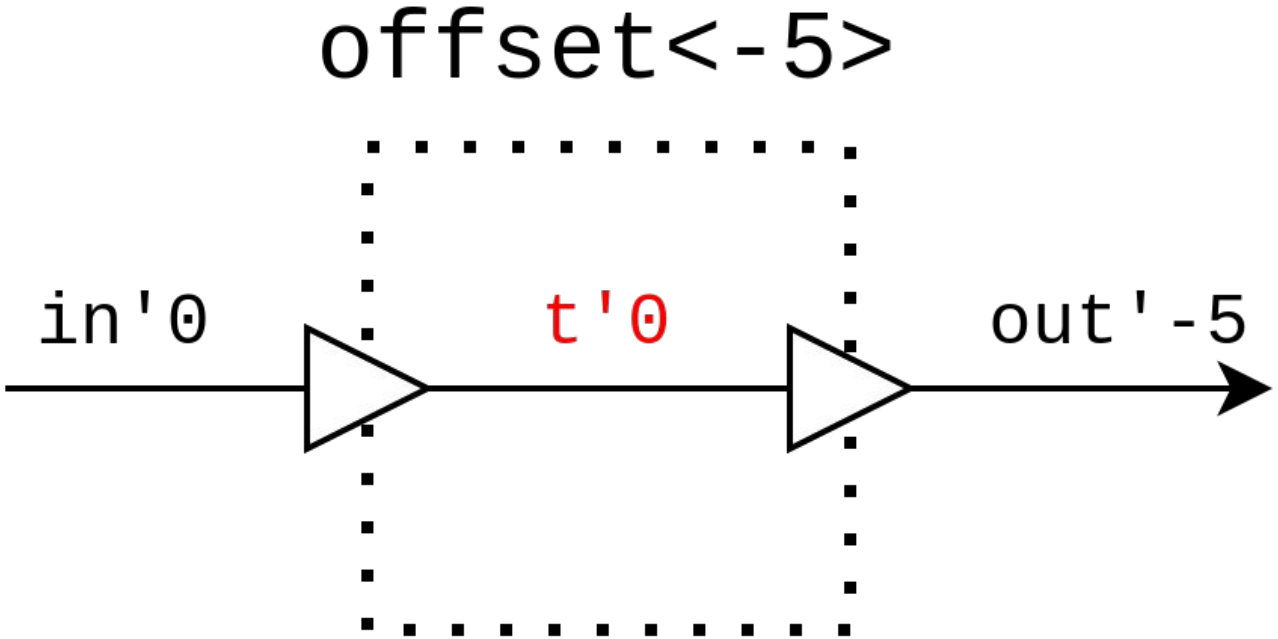
Multiple Interfaces



Latency Cuts

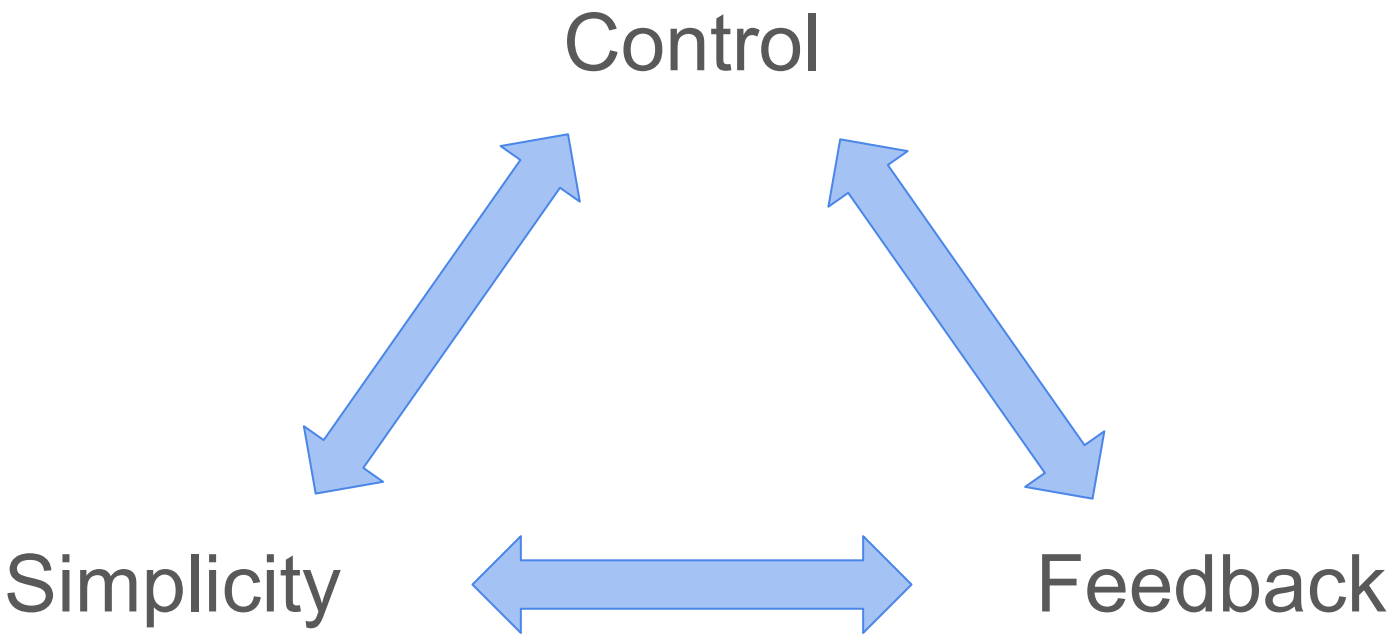


Latency offsets



github.com/pc2/sus-compiler

SUS Goals



Latency Counting

Control

Simplicity

Feedback

Control

- All register-based hardware representable
- What you write is what you get
-

Simplicity

- Conceptual
- Syntactic
- Edit Simplicity
- When to infer things?

Designer Feedback

- LSP
  - Code Navigation
  -
- Instant In-Editor Feedback
  - No compile step