High Level Synthesis
Testability

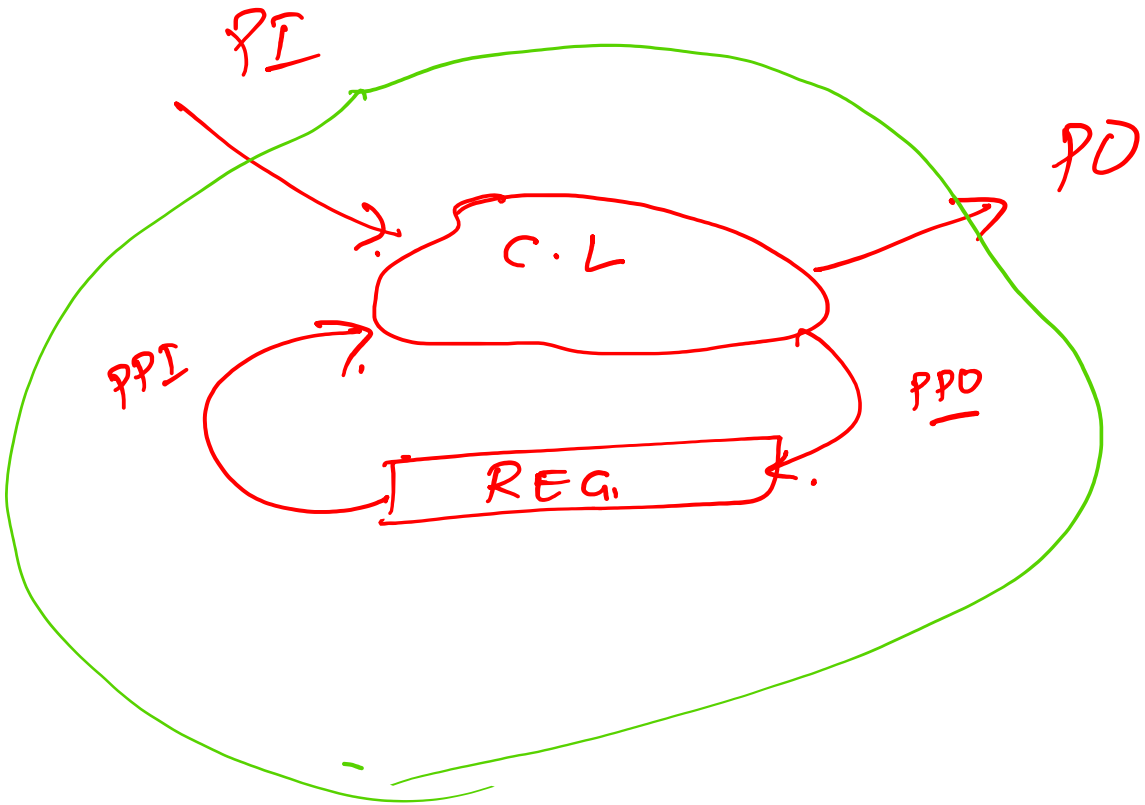
Virendra Singh
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

EE-677: Foundations of VLSI CAD

Lecture 14 on 02 Sep 2021

CADSL

Testability



PI → Controllability

Observability

REG → PO

① Whenever possible allocate a register to at least one PI or PO.

Resource Binding

Register allocation

⇓

Graph

Heuristics

(Left Edge algorithm)

~~Exercise~~

Exercise — Find a solution to allocate registers to variable to improve observability & controllability

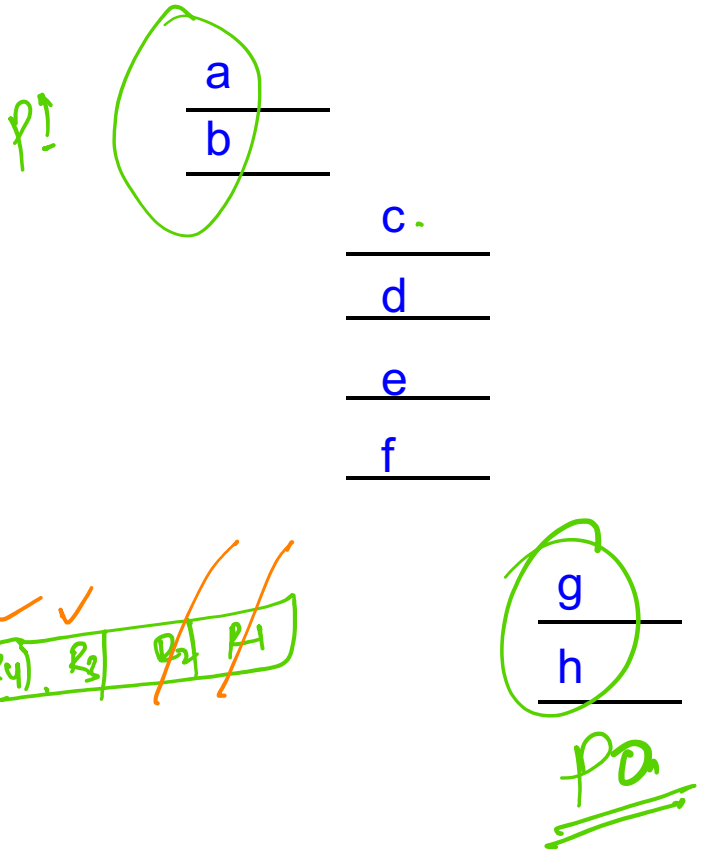
Testability

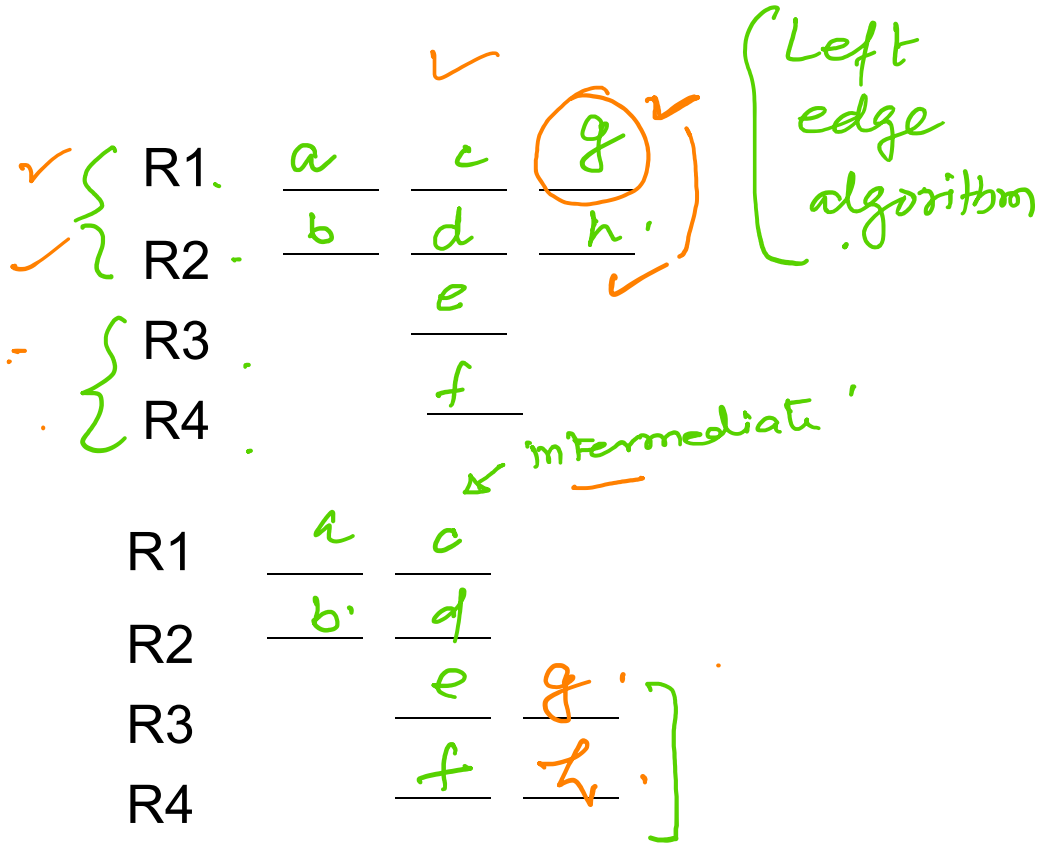
Objective

- To improve
  - Controllability
  - Observability
- Reduction in sequential depth

Controllability and Observability

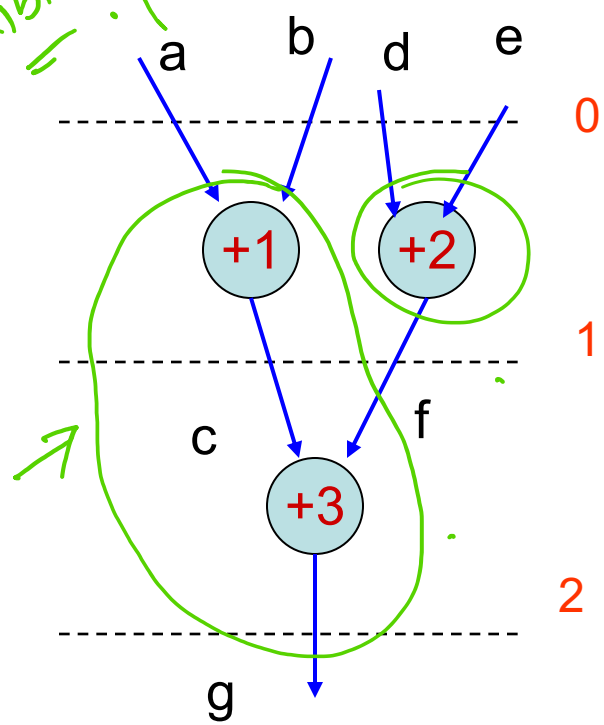
R1 R2 R3 R4

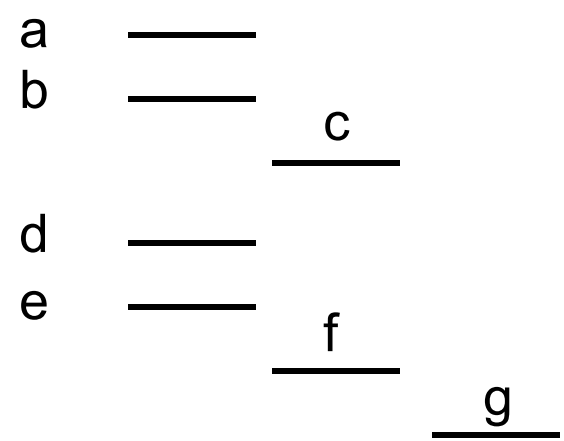




Sequential Depth

flexibility



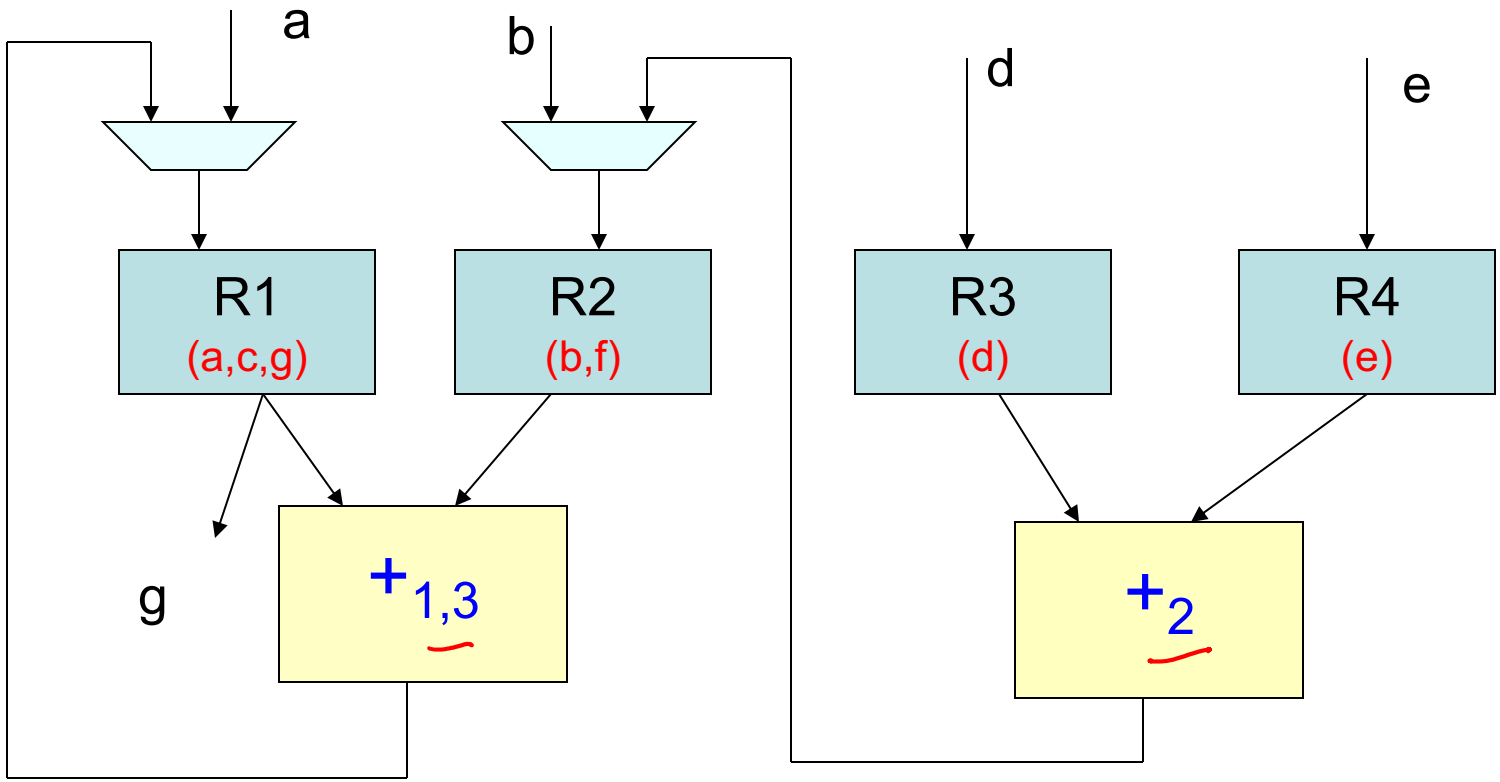


left edge.

operation to module.
var. to Reg.

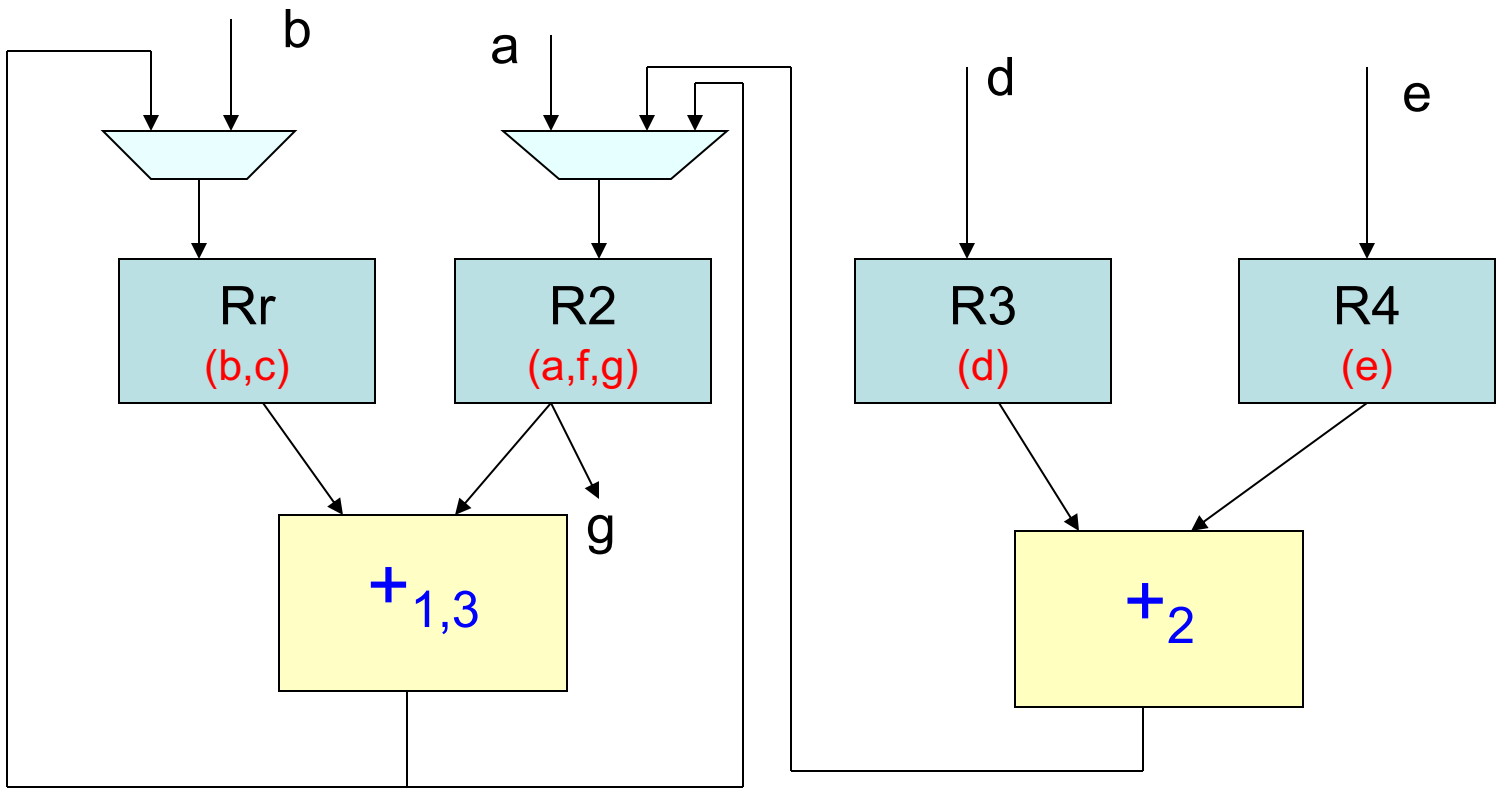
(2) Reduce sequential depth from an input register to an output register

Sequential Depth



Sequential Depth

$d_{seq} = 2$.



SCHEDULING

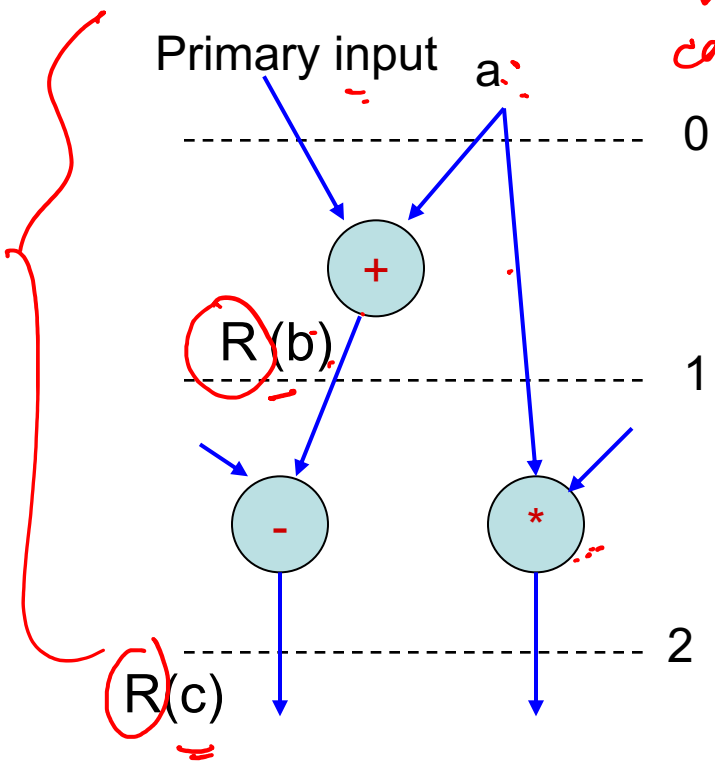
Controllability

Var → Registers

Change the life time

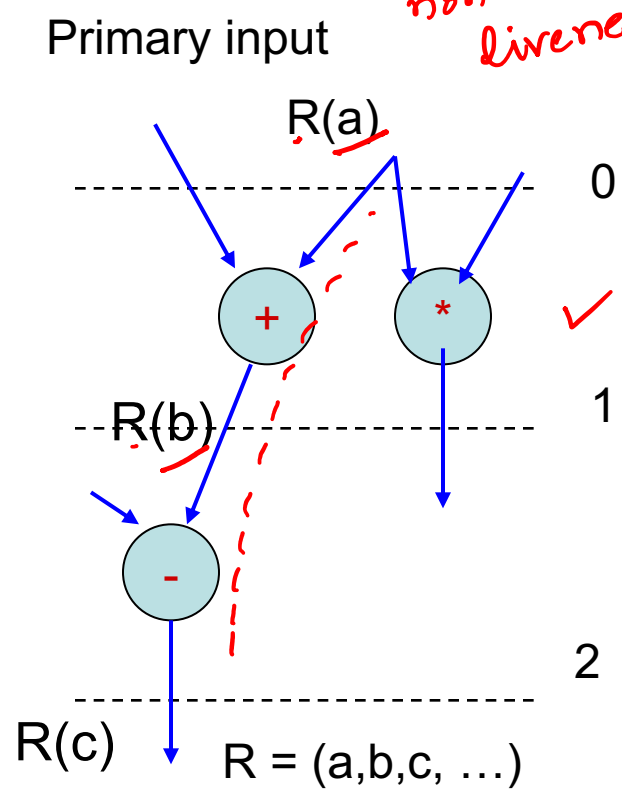
R. a, b cannot share a resource.

a & b non-overlapping liveness range.



R = (b,c, …)

Not directly controllable

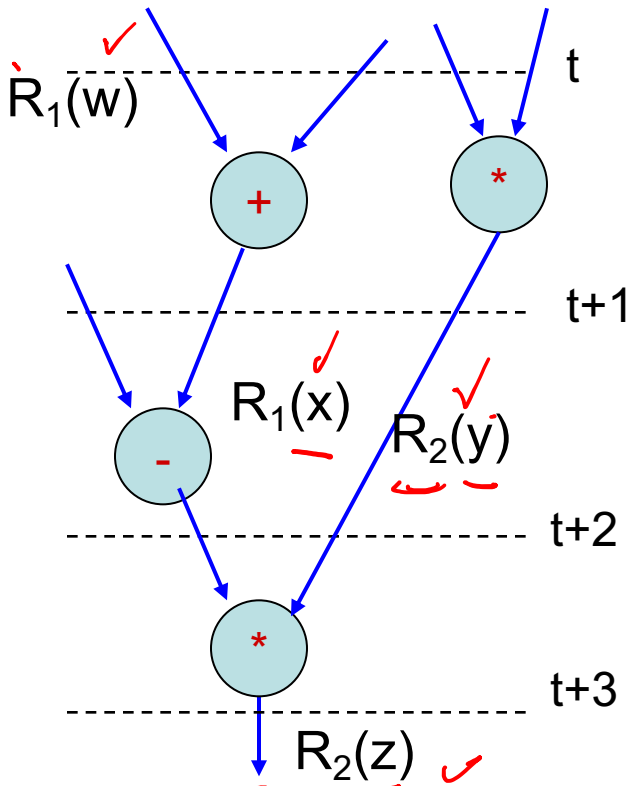


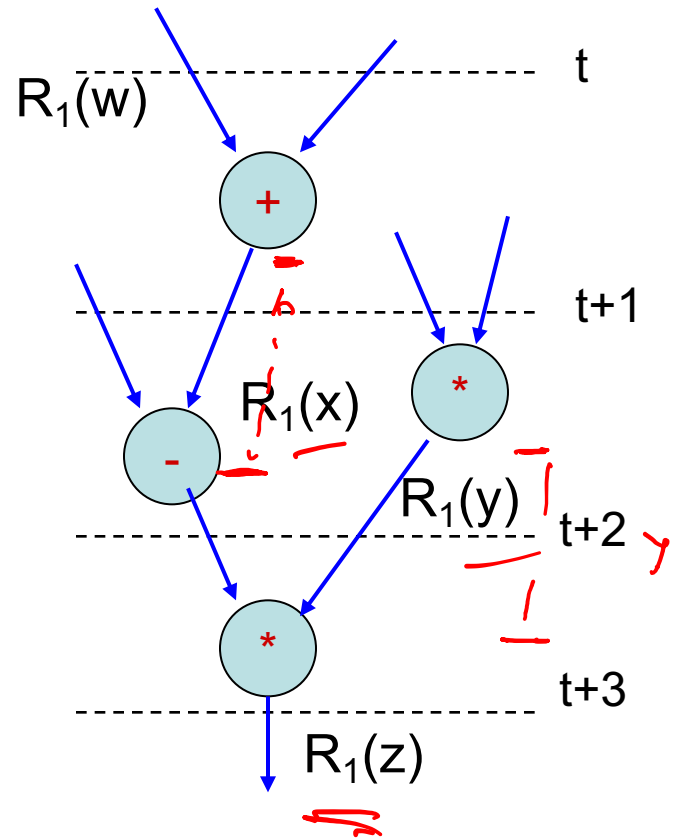
R = (a,b,c, …)

Directly controllable

Observability

Scope

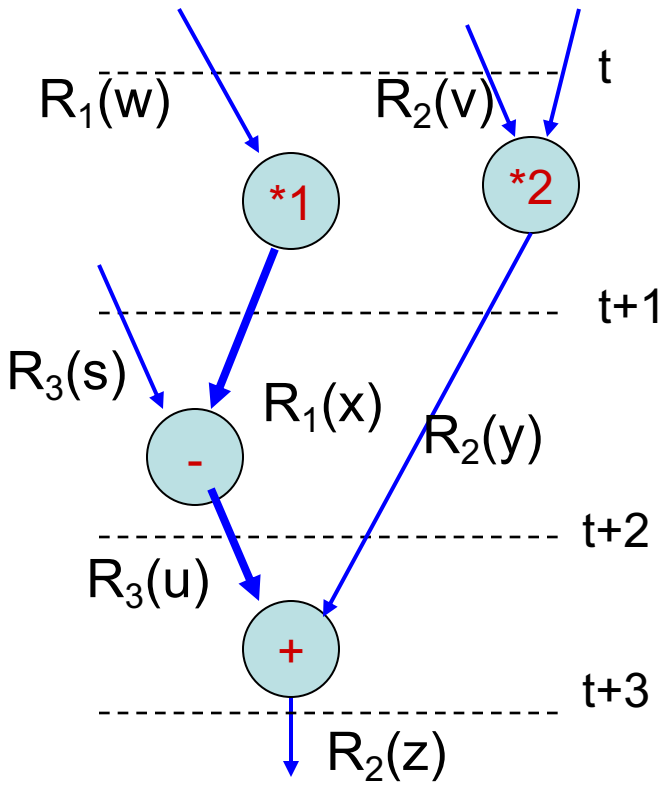


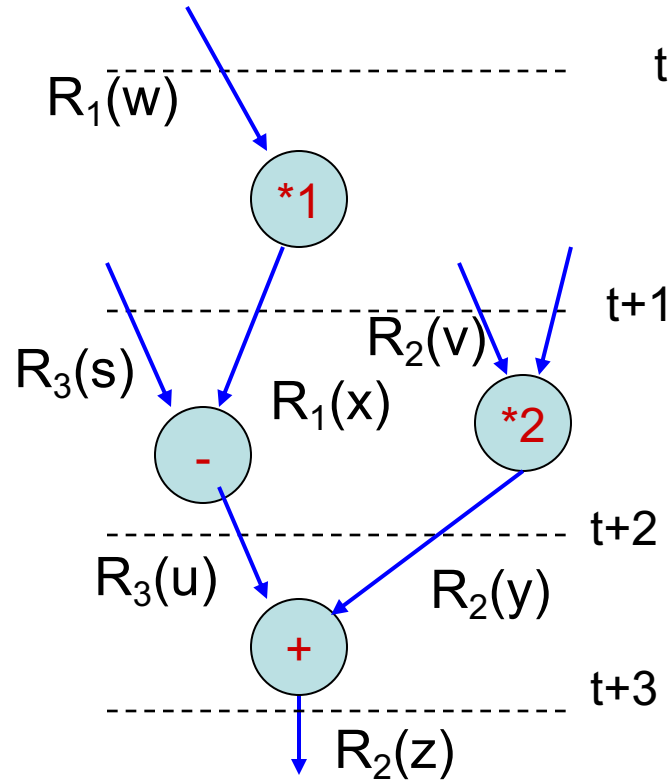


R1 = (….,w,x, …)

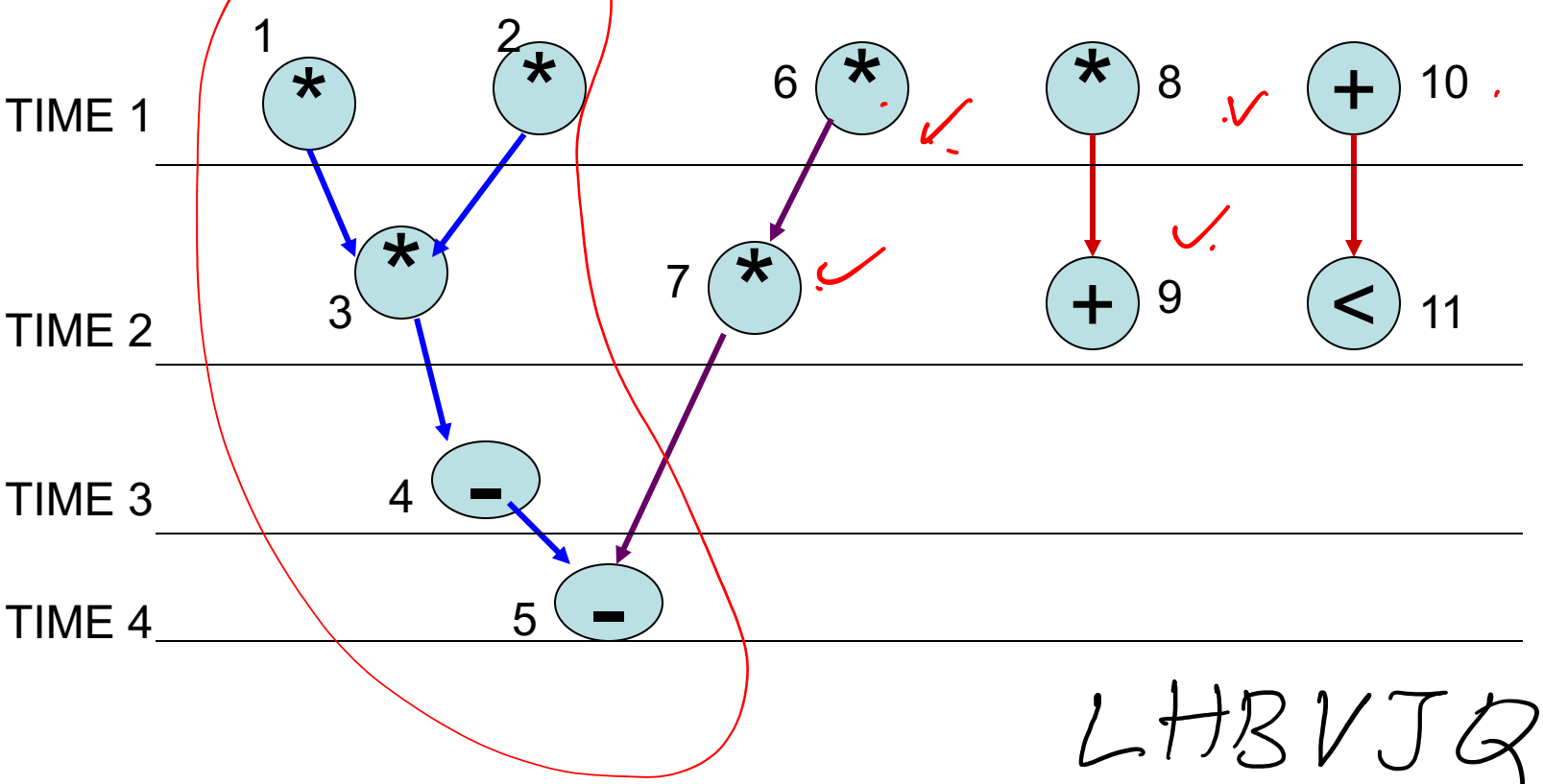
Not directly observable

Sequential Depth Reduction



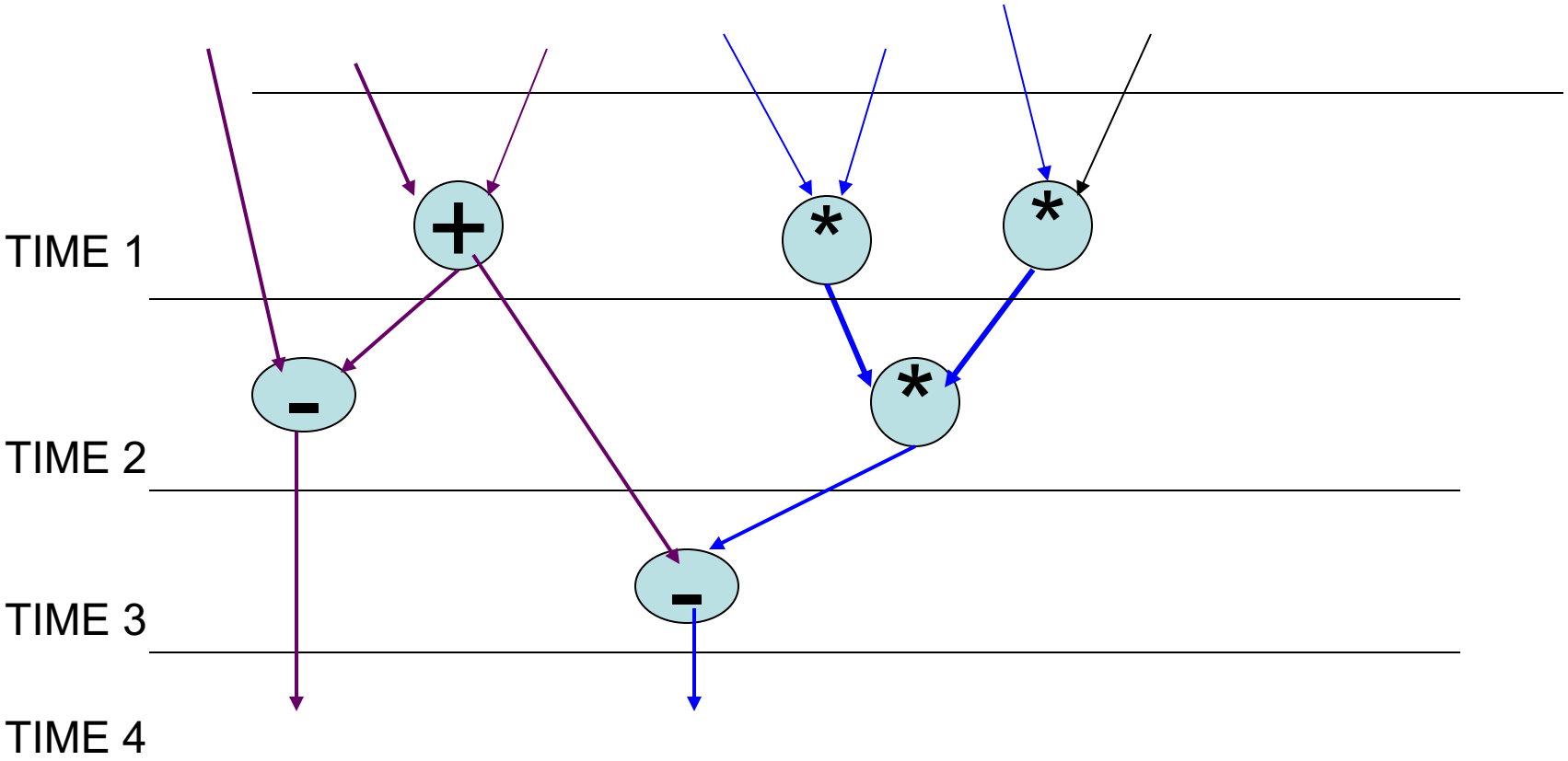


Mobility Path



LHBVJQ.

Mobility Path



Mobility Path Scheduling

```
Mobility_path_scheduling(G){
1.  ASAP_scheduling(G);
2.  ALAP_scheduling(G);
3.  Update_op_slack_and_mobility(G);
4.  While (unscheduled_op(G) ≠ 0){
5.          P_k = next_min_mobility_path(G);
6.          partial scheduling(P_k, G);
7.          testMP(P_k, G); /analyze testability on P_k
8.  }
9.  }
```

Mobility Path Scheduling

```
partial_scheduling(P_k,G){
1.  For each (operation o on Pk)
2.      if (o.earliest = o.latest) // mobility becomes 0
3.      o.active = o.earliest   // assign schedule
4.  Update_op_slack_and)mobility(G);
5.  While (unscheduled_op (P_k) ≠ 0){
6.      (o, o.ll_cycles) = next_op_with
    _least_no_light_load_cycles(Pk, G);
7.      o.active = most_preferred_cycle(o.ll_cycles, G);
```

Thank You