Computer Aided Design Optimization

Virendra Singh

Computer Architecture and Dependable Systems Lab

Department of Electrical Engineering

Indian Institute of Technology Bombay

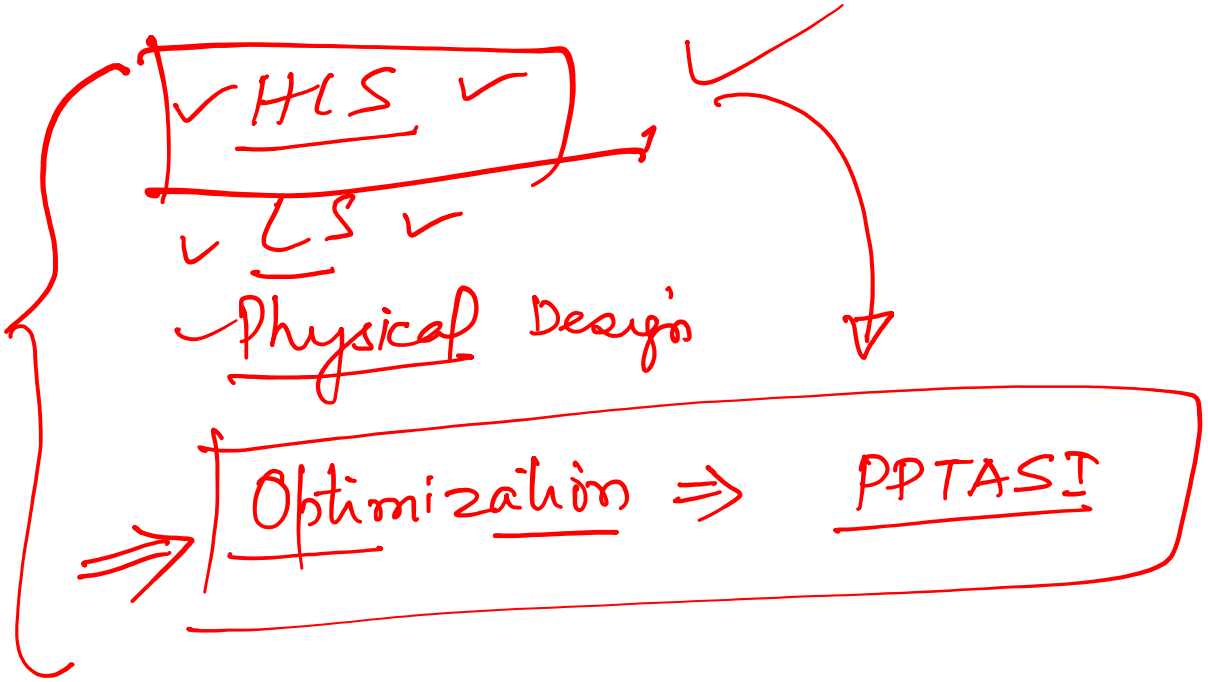
http://www.ee.iitb.ac.in/~viren/

E-mail: viren@ee.iitb.ac.in

EE-677: Foundations of VLSI CAD

Lecture 5 on 05 August 2021

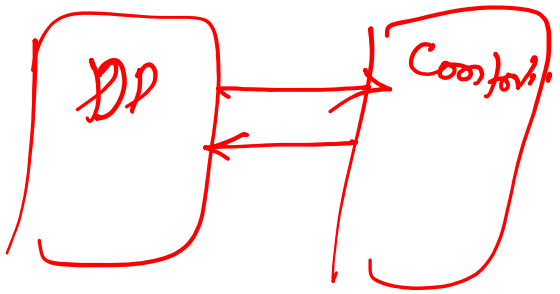
CADSL


HLS
LS
Physical Design
Optimization ⇒ PPTASI

HIGH LEVEL Architectural Synthesis

Architectural Level Abstraction

- Datapath
- Controller



Architectural Synthesis

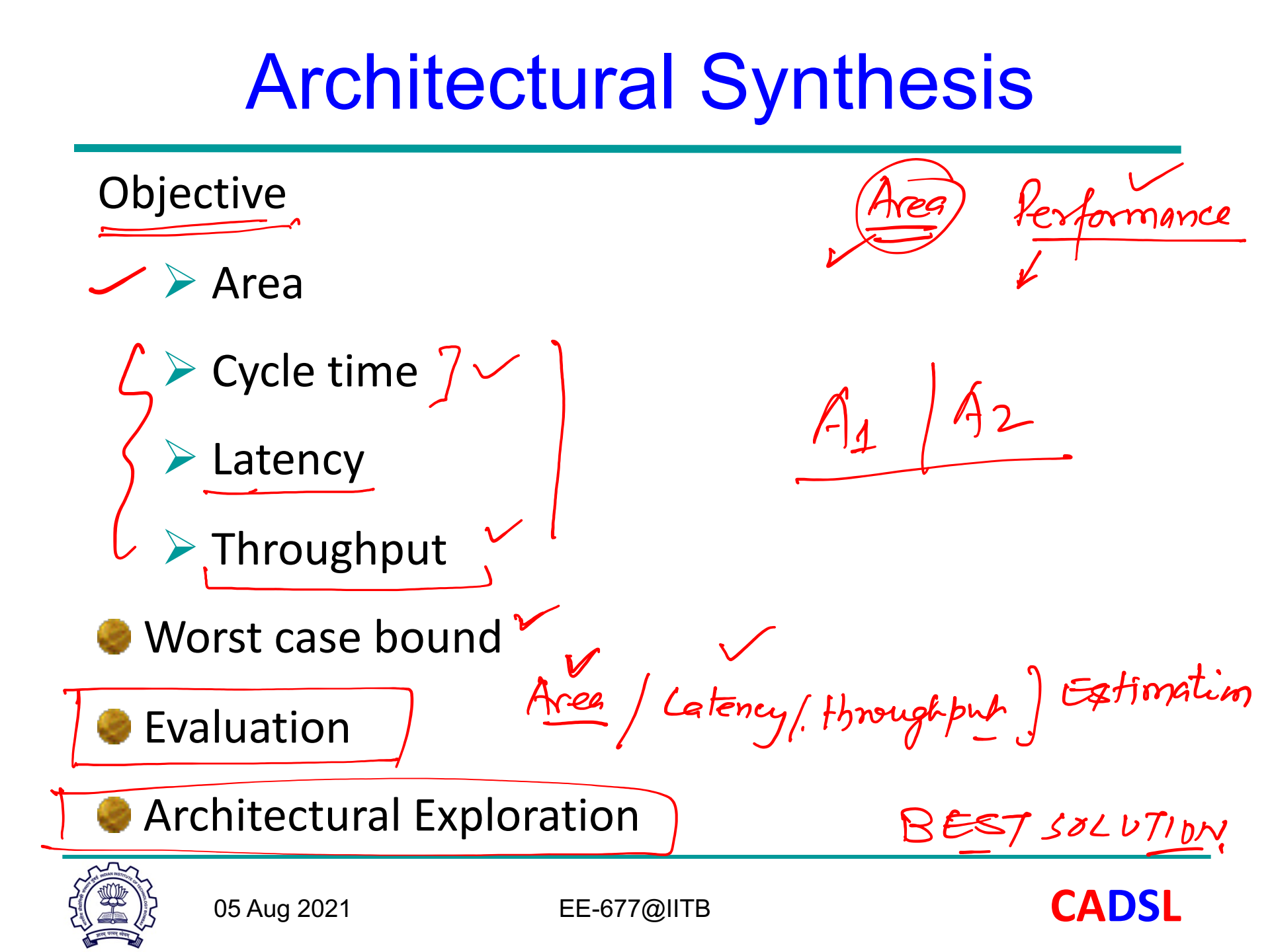
- Constructing the macroscopic structure of a digital circuit starting from behavioural models that can be captured from Data flow or Sequencing Graph] Program

Architectural Synthesis

Objective

- Area
- Cycle time
- Latency
- Throughput

- Worst case bound
- Evaluation
- Architectural Exploration



Architectural Synthesis

Architectural synthesis tool can select an appropriate design point according to some user specific criterion and construct corresponding user specific Datapath and Controller

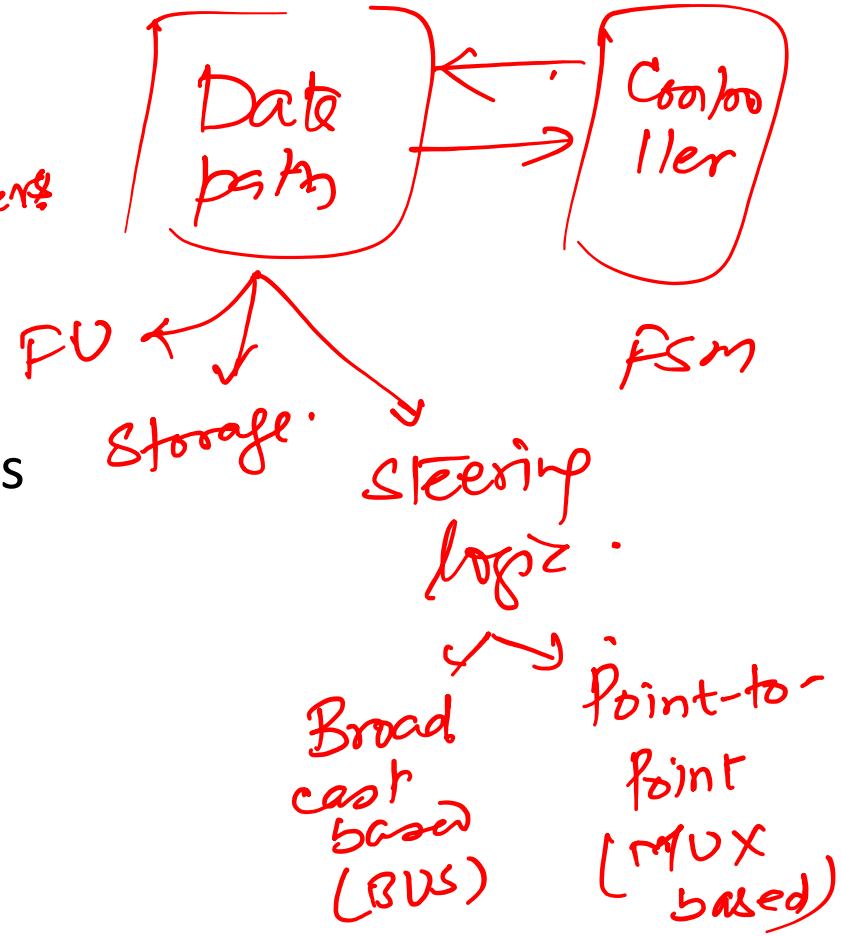
Circuit Specification for Architectural Synthesis

- Behavioural circuit model
- Details about resources being used and constraints
- Capture by Sequencing Graph

Architectural Synthesis

Resources

- Functional Resources
  - Primitive Resources
  - Application Specific Resources
- Memory Resources
- Interface Resources



Architectural Synthesis

Circuit Specification

- Sequencing Graph
- A set of functional resources, fully characterized in terms of area and execution delay
- A set of constraints

Architectural Synthesis

Computation: Differential Equation Solver

$Y'' + 3 x y' + 3y = 0$

$X(0) = 0$

$y(0) = y$

$y'(0) = u$

```
Diffeq{
    read (x, y, u, dx, a)
    repeat{
        xl = x + dx
        ul = u – (3*x*u*dx) –
                (3*y*dx)
        yl = y + (u*dx);
        c = xl < a
        x = xl; u = ul; y = yl;
    until (c);
    write (y)
}
```

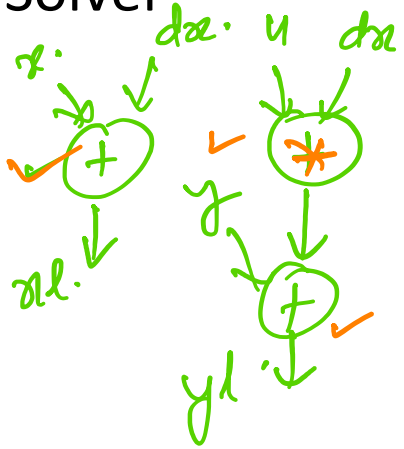
Architectural Synthesis

```
architecture BEHAVIOUR of DIFFEQ is
begin
process
        variable x, y, u, dx, a, xl, ul, yl: bit8;
begin
    wait until start'event and start = '1';
        x := x_port; y := y_port; a := a_port; u := u_port; dx := dx_port;
        DIFFEQ_LOOP:
        while ( x < a) loop
                wait until clk'event and clk = '1';
                xl = x + dx;
                ul = u – (3*x*u*dx) – (3*y*dx);
                yl = y + (u*dx);
                x = xl; u = ul; y = yl;
        end loop DIFFEQ_LOOP;
        y_port := y;
end process
end BEHAVIOUR;
```

Architectural Synthesis

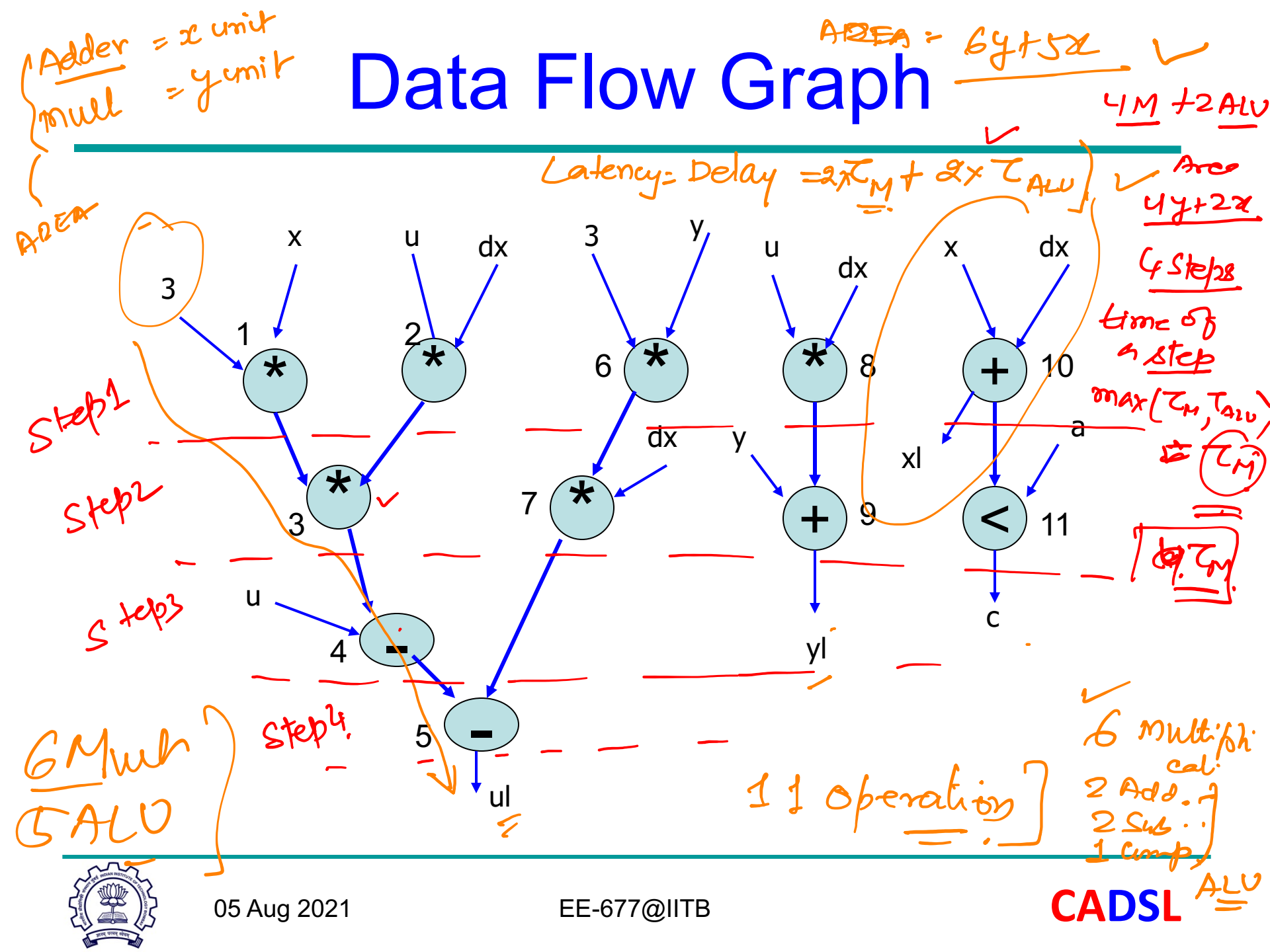
Computation: Differential Equation Solver

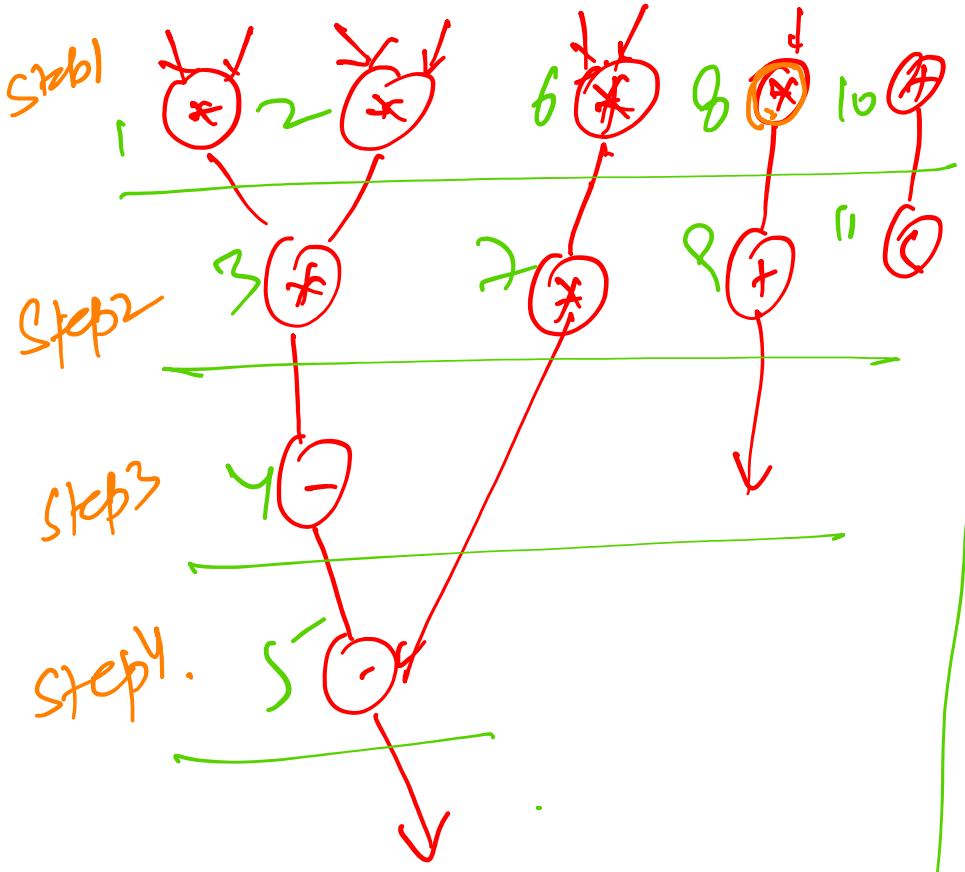
```
xl = x + dx
ul = u – (3*x*u*dx) – (3*y*dx)
yl = y + (u*dx);
c = xl < a
```

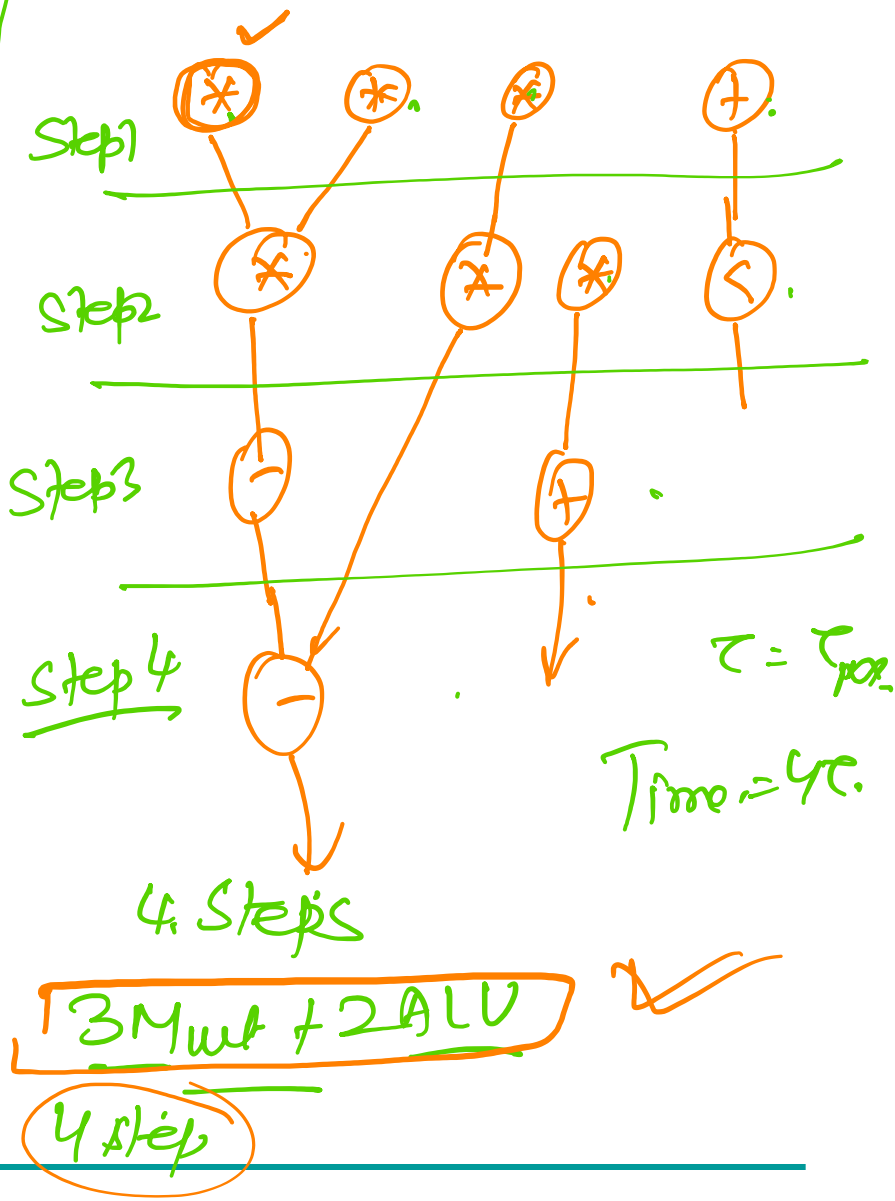


Data Flow Graph (DFG): represent operation and data dependencies

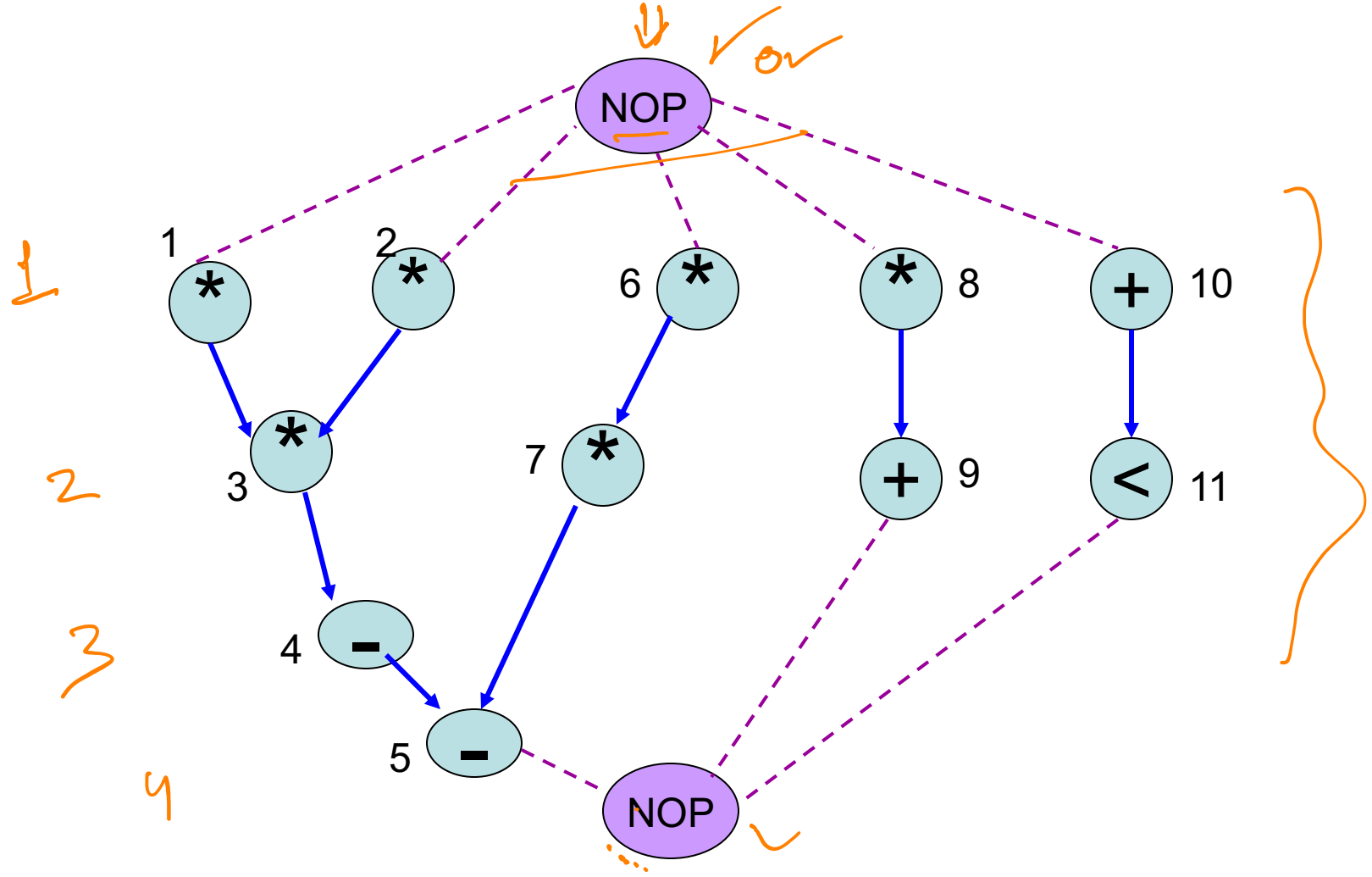
Data Flow Graph



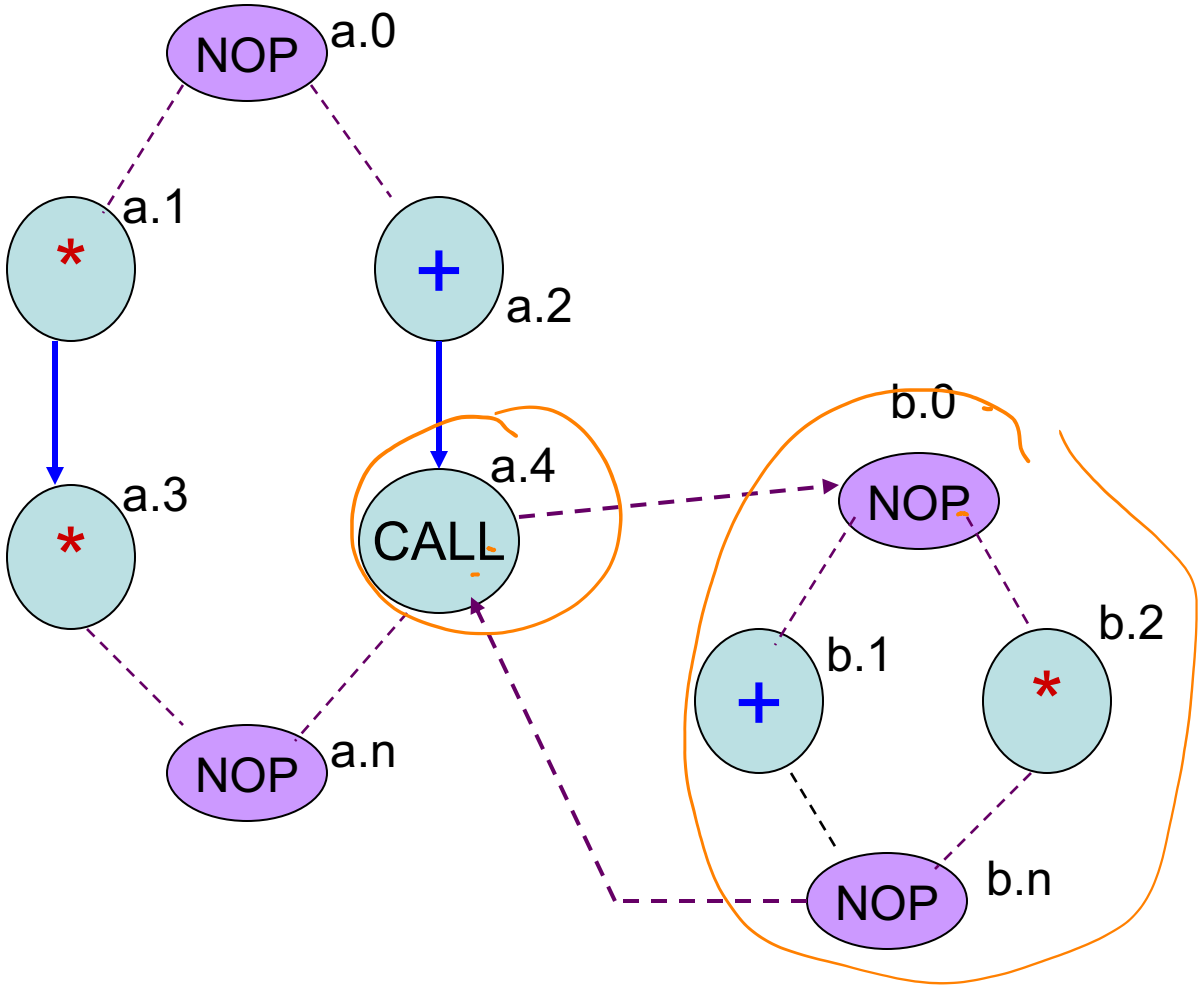

Step1
Step2
Step3
Step4
4 M + 2 ALU
4 step


Step1
Step2
Step3
Step4
4 Steps
3 Mult + 2 ALU
4 step
Time = 4τ.

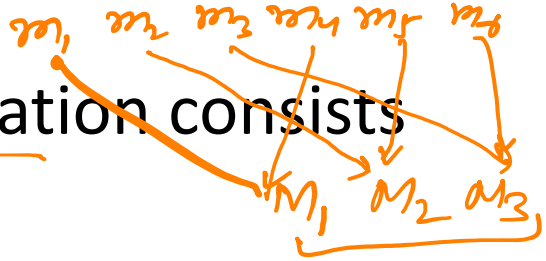
Sequencing Graph



Hierarchical Sequencing Graph



Architectural Synthesis



Architectural Synthesis and optimization consists of two stages

1. Placing the operation in time and in space, i.e., determining their time interval of execution and binding to resources
2. Determining detailed interconnection of the datapath and the logic-level specifications of the control unit

Temporal Domain: Scheduling

Delay **D** = $\{d_i; i = 0,1, 2, \ldots\ldots n\}$

Start time T = $\{t_i; i= 0, 1, \ldots., n)$

Scheduling: Task of determining the start timing, subject to preceding constraints specified by sequencing graph

Latency $\lambda = t_n - t_0$

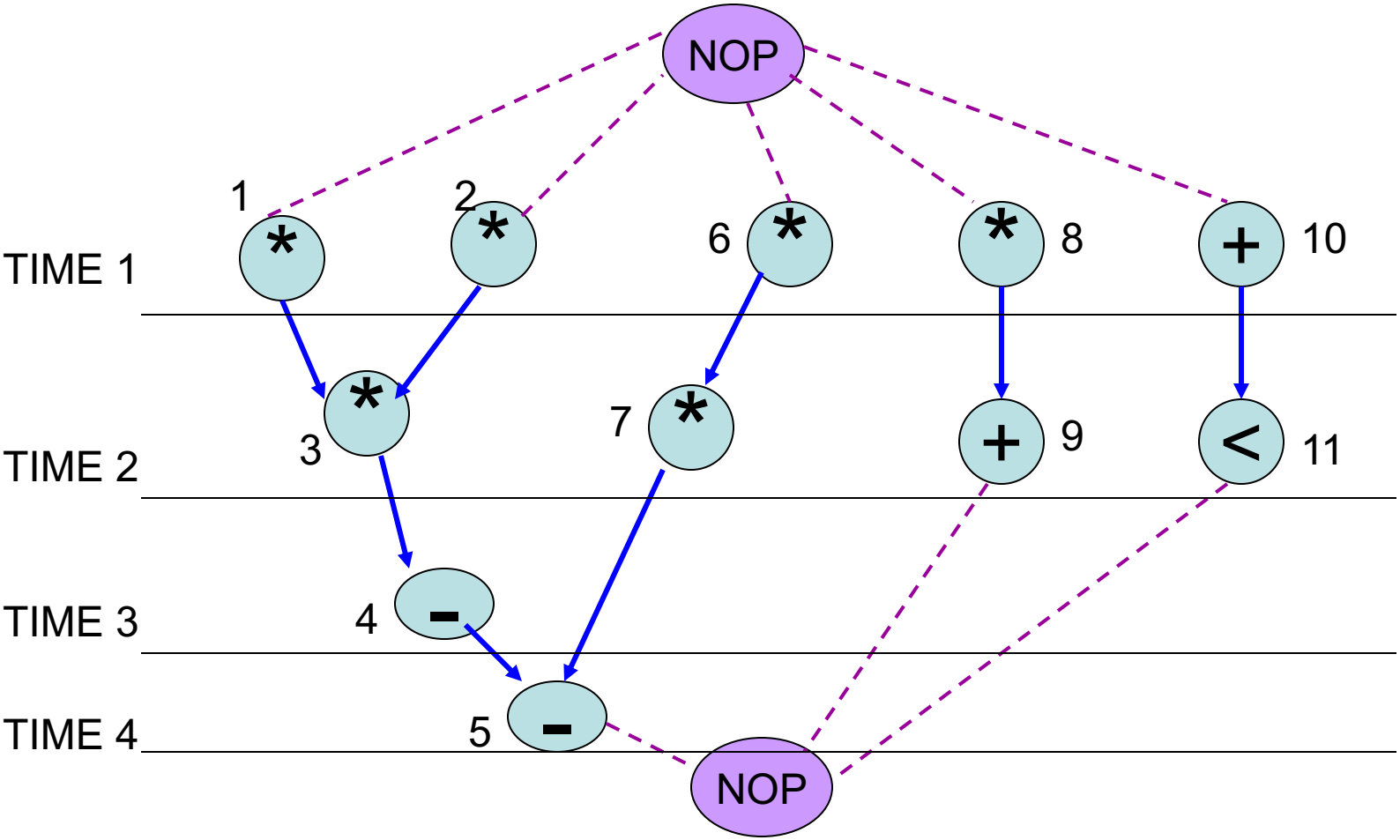
Temporal Domain: Scheduling

A scheduled sequencing graph is a vertex-weighted sequencing graph, where each vertex is labeled by its start time

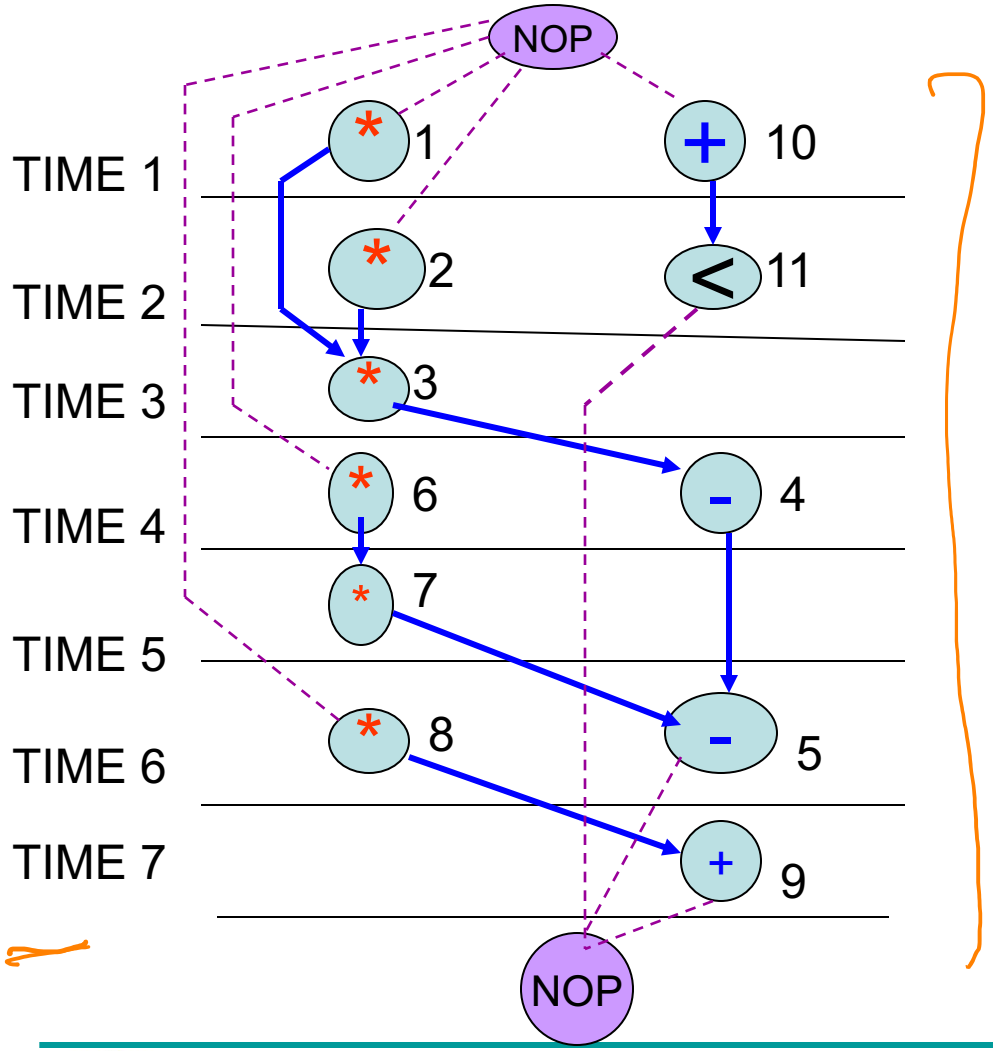
Operation	Start time
V1,V2, v6, v8, v10	1
V3, v7, v9, v11	2
V4	3
V5	4

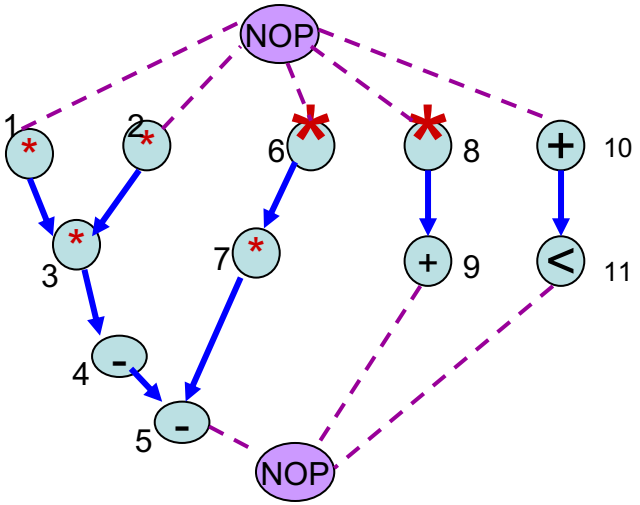
- Chaining

Temporal Domain: Scheduling



Temporal Domain: Scheduling





1M + 1ALU ⇐

7 steps ⇒ 4.

Spatial Domain: Binding

A fundamental concept that relates operation to resources is binding

- Resource types
- Resource sharing

Simple case of binding is a dedicated resources

Spatial Domain: Binding

$\beta(v1) = (1,1)$

$\beta(v2) = (1,2)$

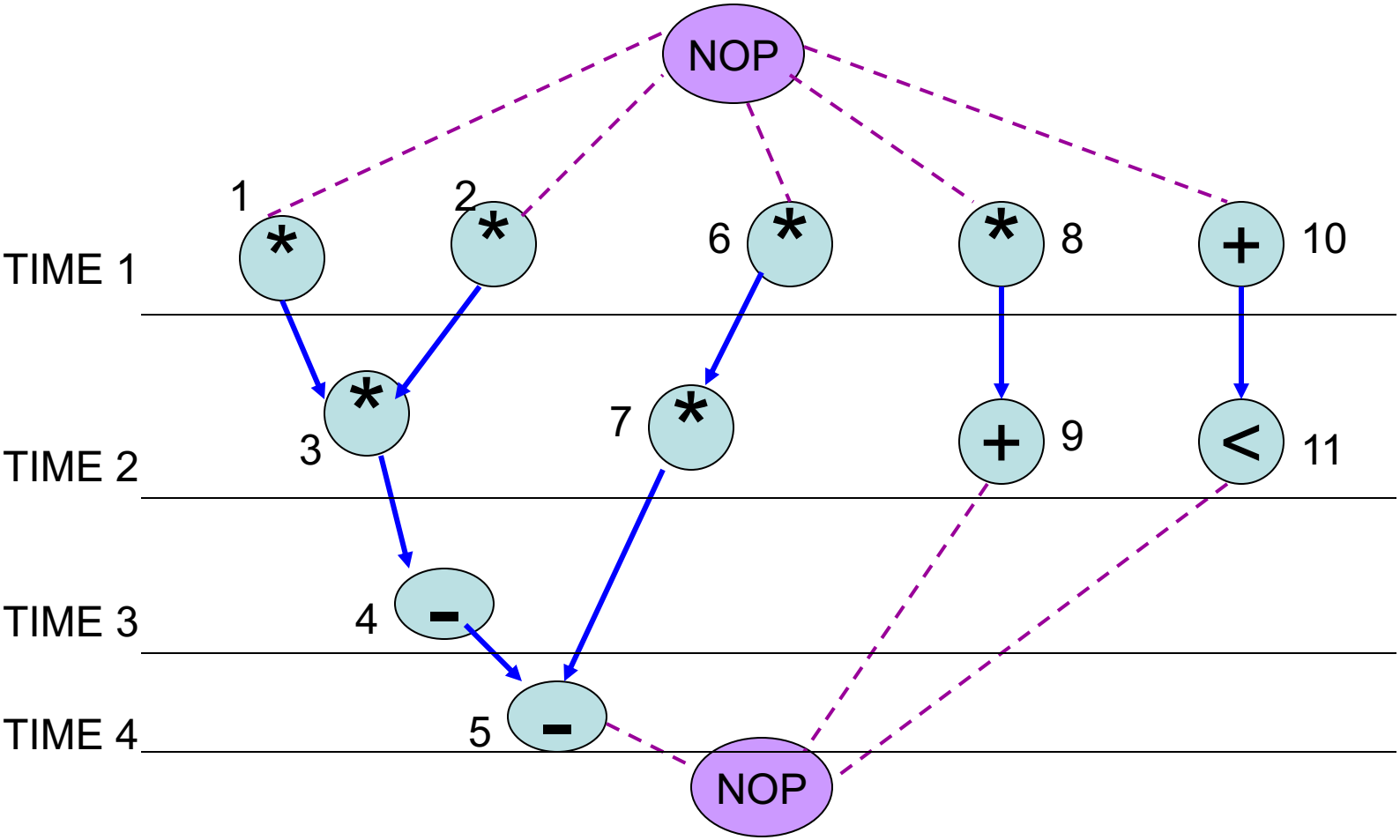
$\beta(v3) = (1,3)$

$\beta(v4) = (2,1)$

$\beta(v5) = (2,2)$

..

Spatial Domain: Binding

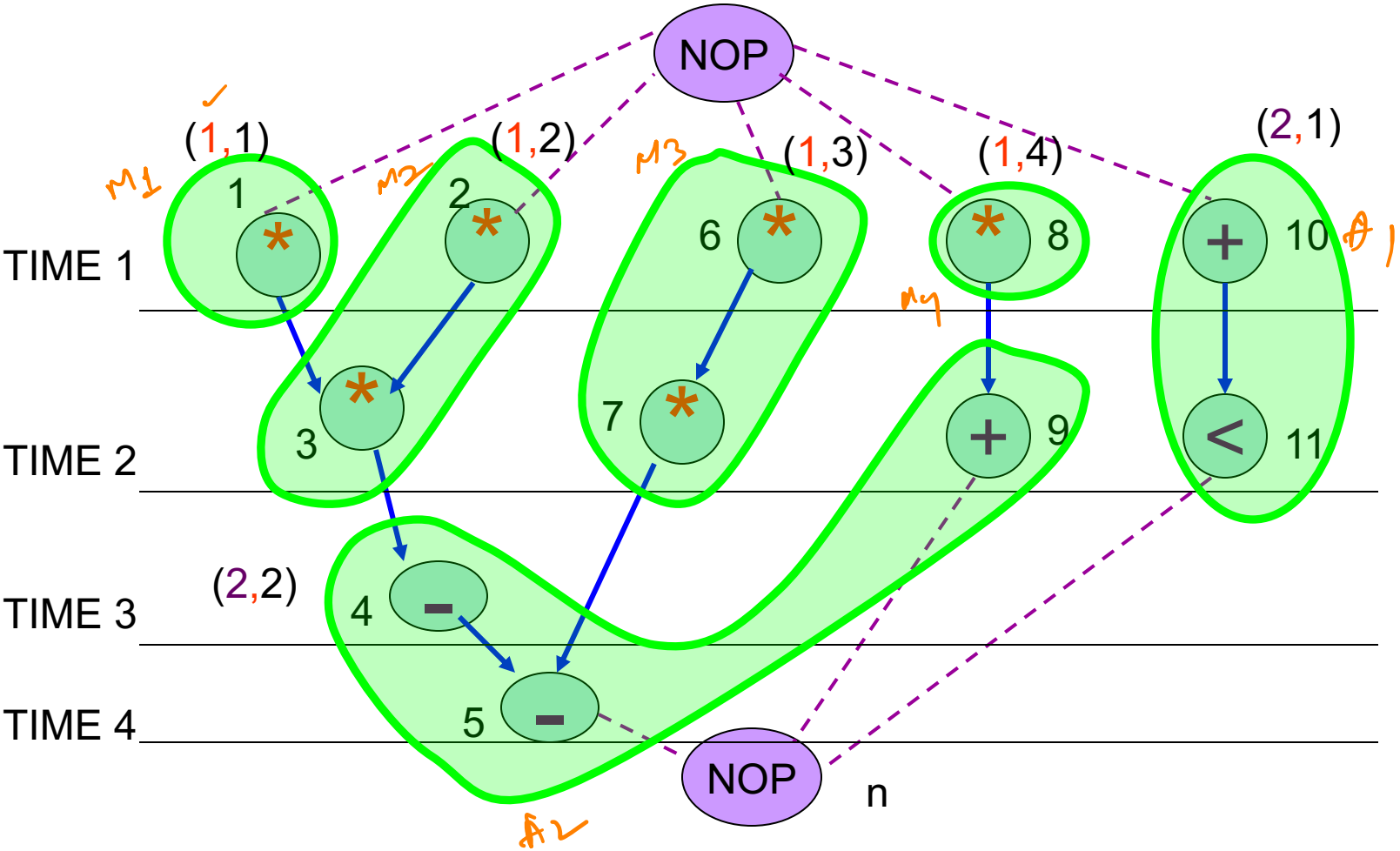


Spatial Domain: Binding

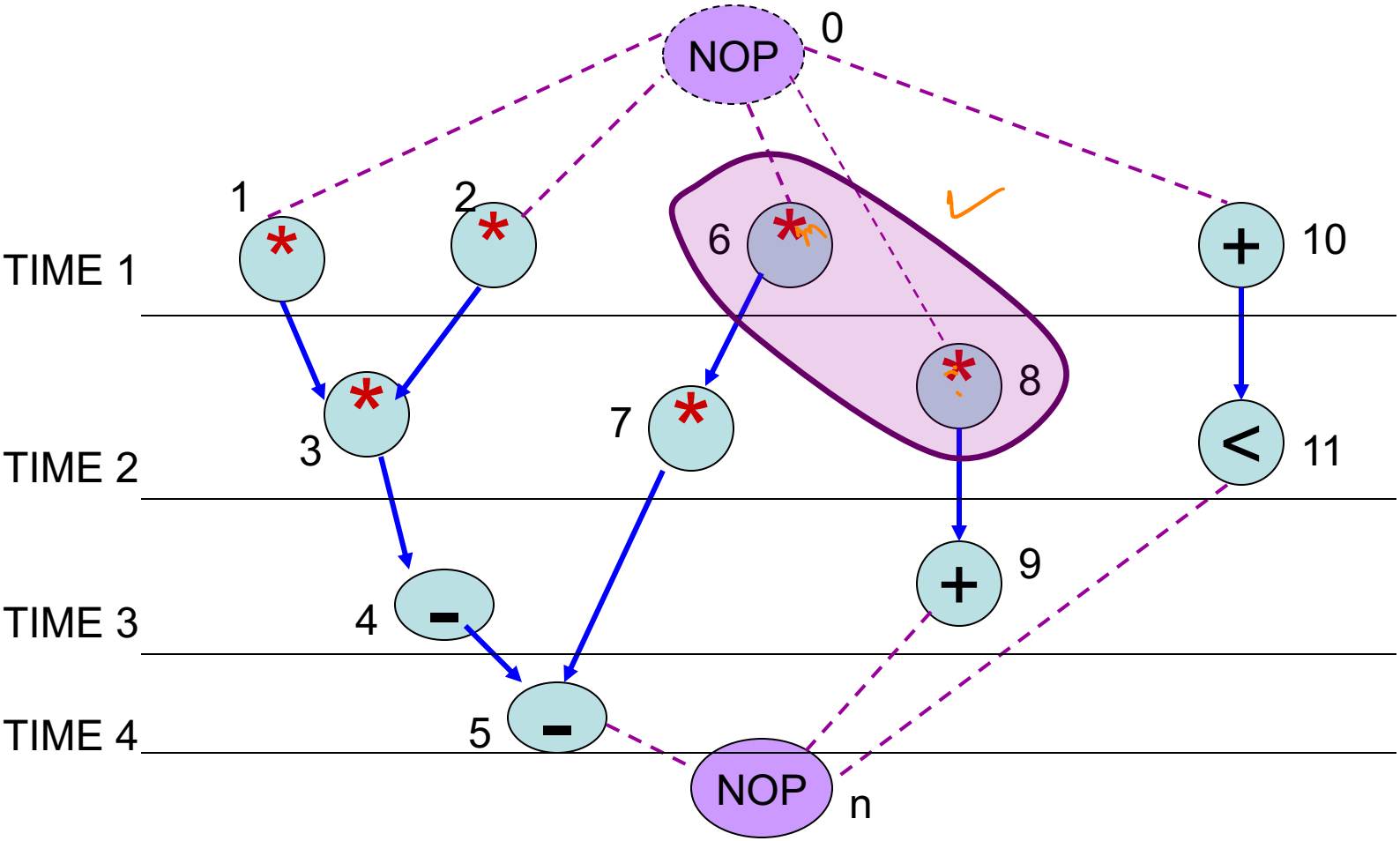
- A necessary condition for resource binding to produce a valid circuit implementation is that operation corresponding to the shared resource do not execute concurrently

- A resource binding can be represented by a labeled hyper-graph, where the vertex set V represents operations and the edge set E_β represents the binding of the operation to the resources

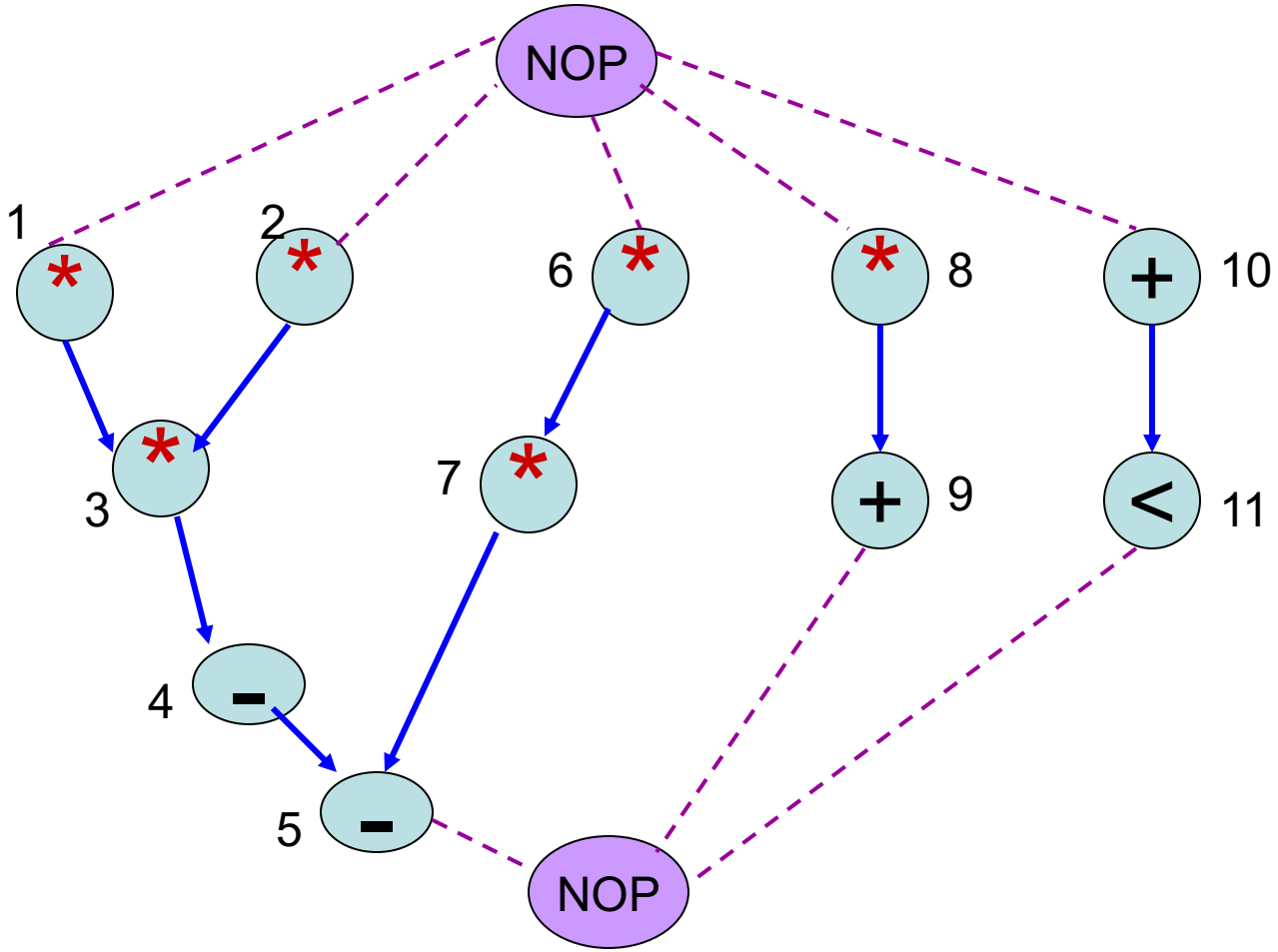
Spatial Domain: Binding



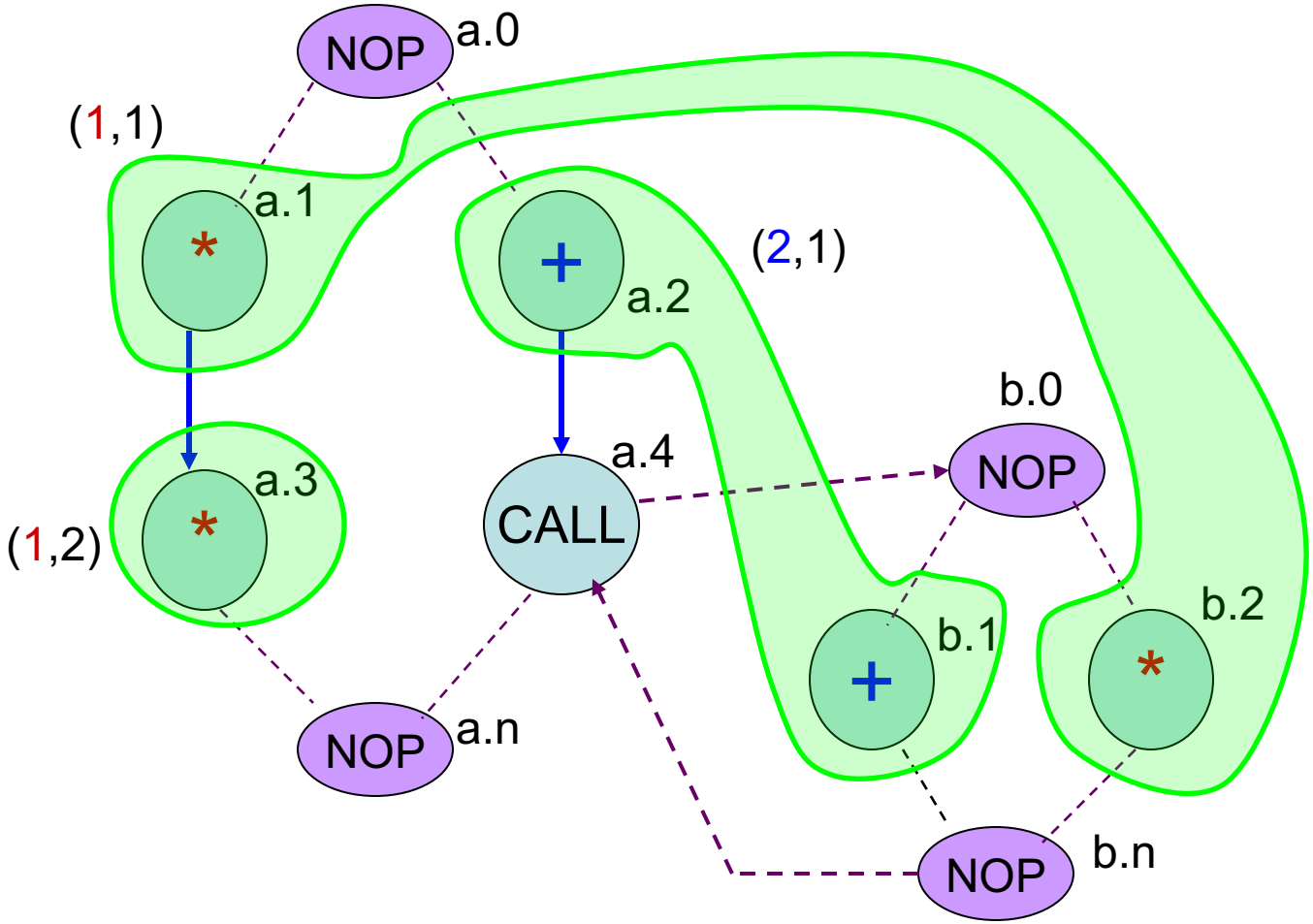
Spatial Domain: Binding



Sequencing Graph



Hierarchical Sequencing Graph



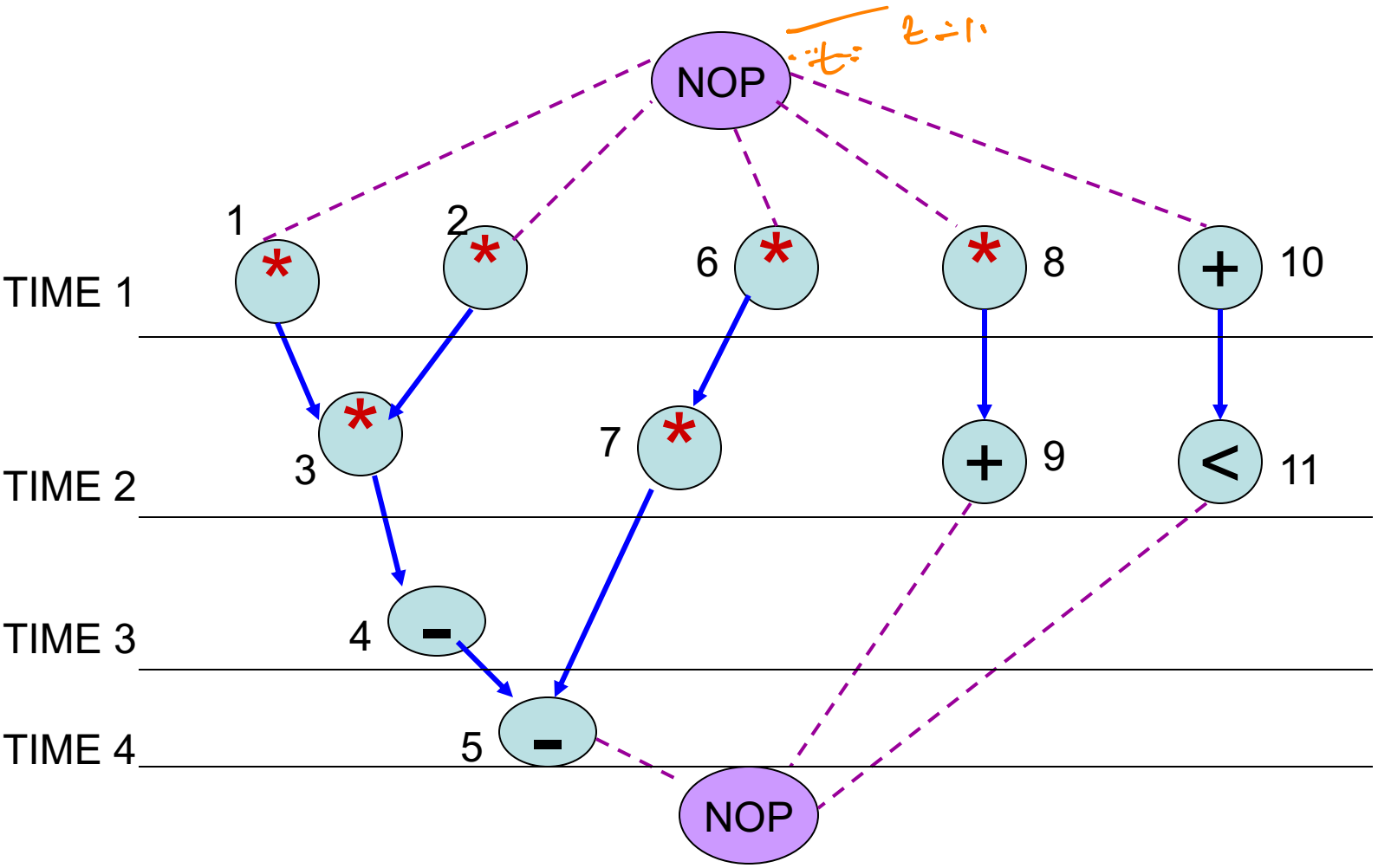
Area/Performance Estimation

Accurate area and performance estimation is not an easy task

Schedule: provides latency

Binding: provides information about the area

ASAP Scheduling



⇒ ASAP Scheduling

ASAP($G_s(V,E)$){

Schedule v_0 by setting $t_0^s = 1$;

repeat{

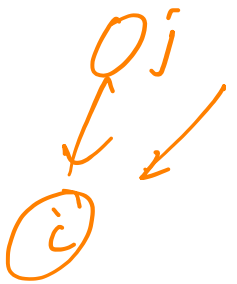
select vertex v_i whose predecessors are all scheduled;

schedule v_i by setting $t_i^s = \max\{t_j^s + d_j\}$

} untill (v_n is scheduled)

return (t^s);

}



1

2+1

3+1

λ best

Thank You