Introduction of Processor Design for AI Applications

L08 – Scheduling and Resource Allocation

Pengju Ren
Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University

http://gr.xjtu.edu.cn/web/pengjuren

Scheduling and Resource Allocation

Scheduling and **resource allocation** are two important tasks in hardware or software synthesis of DSP systems

■ Scheduling – **when** to do the process?

Assign every node of the DFG to a *control time step*, the fundamental sequencing units in synchronous systems and correspond to clock cycles

■ Resource allocation – **who** to execute the process?

Assign operations to hardware with the goal of minimizing the amount of hardware required to implement the desired behavior

Scheduling

- **Static scheduling**
  - If all processes are known in advance
  - Perform scheduling before run time
  - Most data-stream algorithms are amenable to static scheduling
- **Dynamic scheduling**
  - When the process characteristics are not completely known
  - Decide dynamically at run time by scheduler that runs concurrently with the program

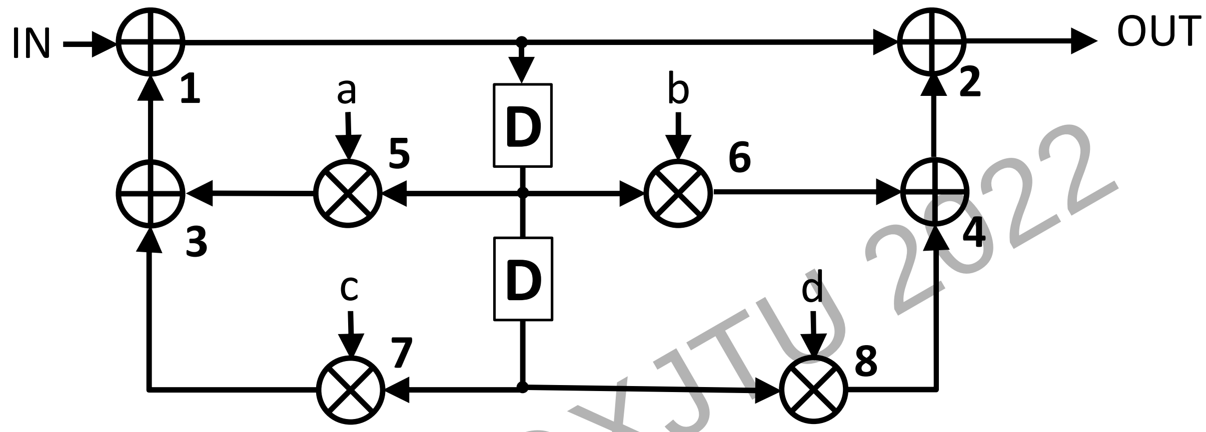
Criteria of Scheduling Optimization

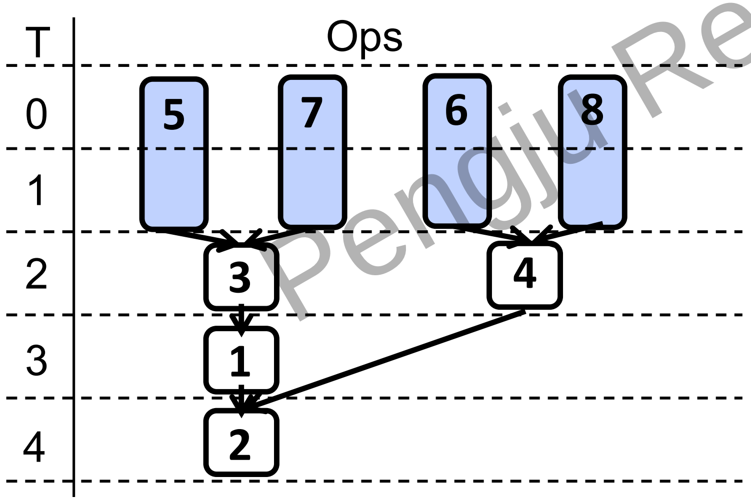
- **Sample period optimal**
  - Optimal if *sample period* = *iteration bound*
- **Delay optimal**
  - Optimal if delay = *delay bound*, the shortest possible delay from input to output of the algorithm
- **Resource optimal**
  - Minimum amount of resource
  - Processor based system corresponds to a resource limited scheduling problem

ASAP and ALAP Scheduling

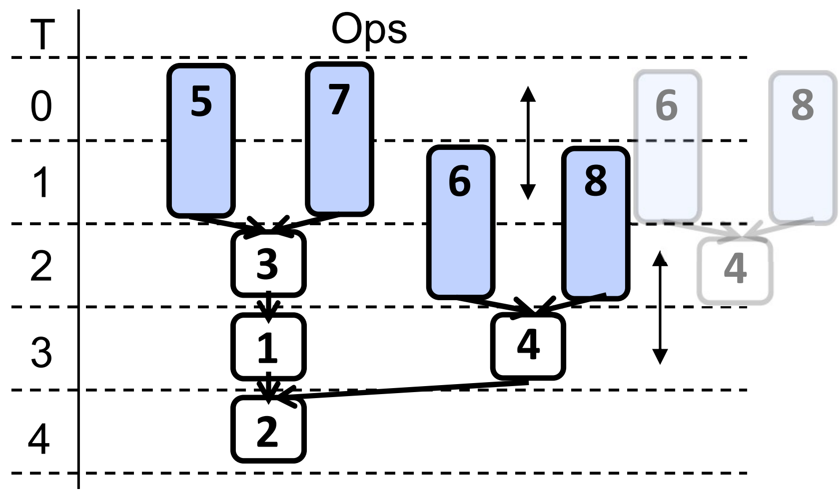
- **ASAP**
  - A computation can be performed *as soon as all of its inputs are available*
  - The aim is to obtain the shortest possible execution time without considering resource requirements
- **ALAP**
  - Similar to ASAP, the operations are scheduled *as late as possible*
  - Can be used to determine the time range in which the operations can be scheduled

Example： ASAP and ALAP Scheduling





ASAP Scheduling

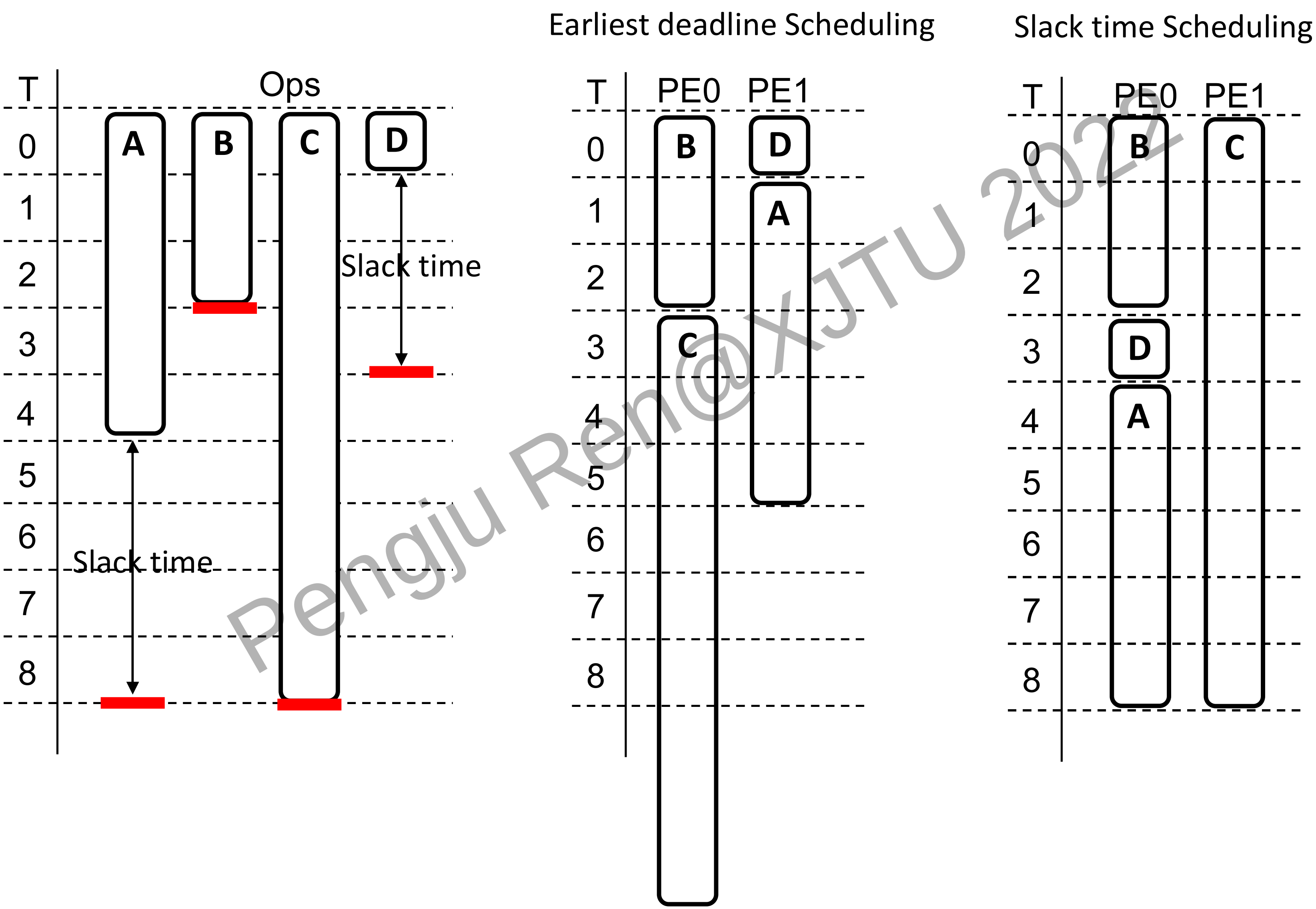


ALAP Scheduling

Earliest Deadline and Slack Time Scheduling

- **Earliest deadline scheduling**
  - In each time step, the processes whose deadline is closest will be done
  - Proven to be execution-time-optimal for single processor
- **Slack time algorithm**
  - Schedule the process whose slack time is least
  - **Slack time:** the time from present to the deadline minus the remaining processing time of a process
  - Better than earliest deadline scheduling when more than one processor is used

Example：Earliest Deadline and Slack Time Scheduling



Critical Path List Scheduling and Example

- One of the list scheduling method
- Form an ordered list of operations to be performed according to critical paths
- The operations are picked one by one from this list and assigned to a free resource (PE)

T	Ready List				Mul-A	Mul-B	Add-A
0	5	6	7	8	5	7	
1		6		8	6	8	
2		3					3
3		1	4				1
4			4				4
5		2					2

Force-Directed Scheduling

Target: to distribute the computation on time axis

While (Unscheduled nodes exist)

{

① Compute the time frames (life-span) for each node

② Build the *distribution graph*

③ Compute the *total force* = (*self force*) + (*indirect force*)

④ Schedule the node into the time step that minimizes the total force

}

Example：Force-Directed Scheduling (1)

$$y'' + 3xy' + y = 0$$

$$let\ \ u = y' = \frac{dy}{dx}$$

$$\frac{du}{dx} = y'' = \frac{d^2y}{dx^2} = -3xy' - 3y = -3xu - 3y$$

$while(x < a)\{$

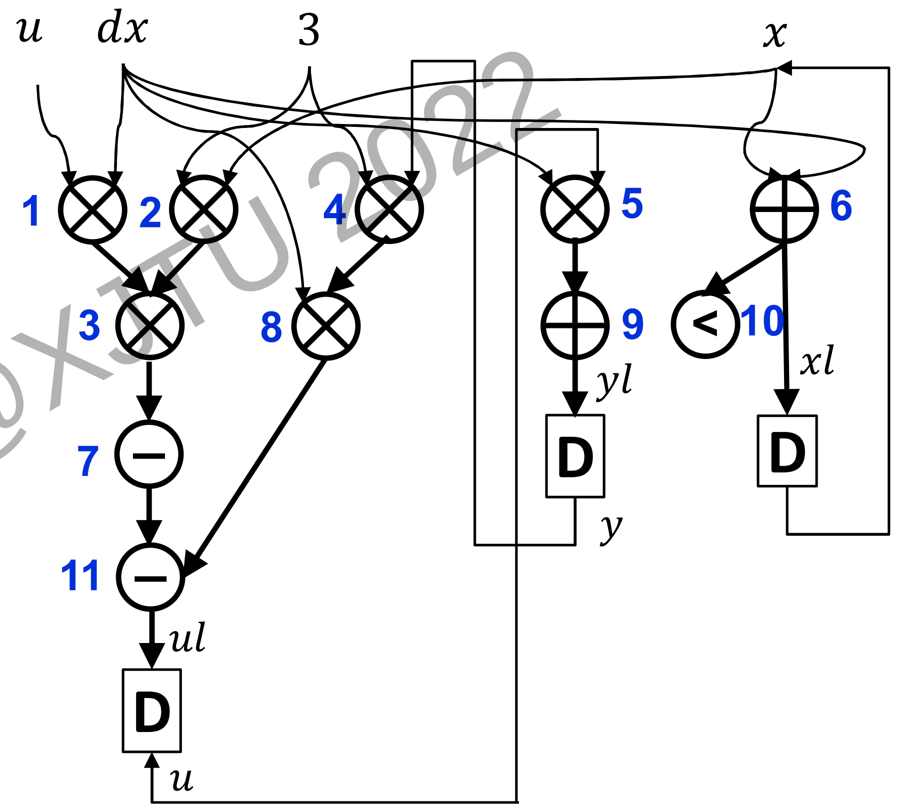
$\quad xl = x + dx;$

$\quad ul = u - (-3xu - 3y)dx = u - 3xudx + 3ydx;$

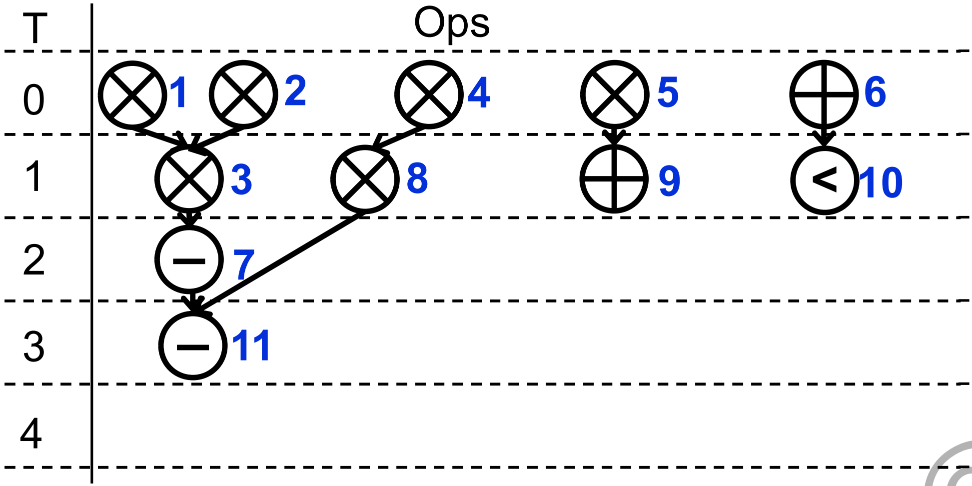
$\quad yl = y + udx;$

$\quad x = xl; y = yl; u = ul$

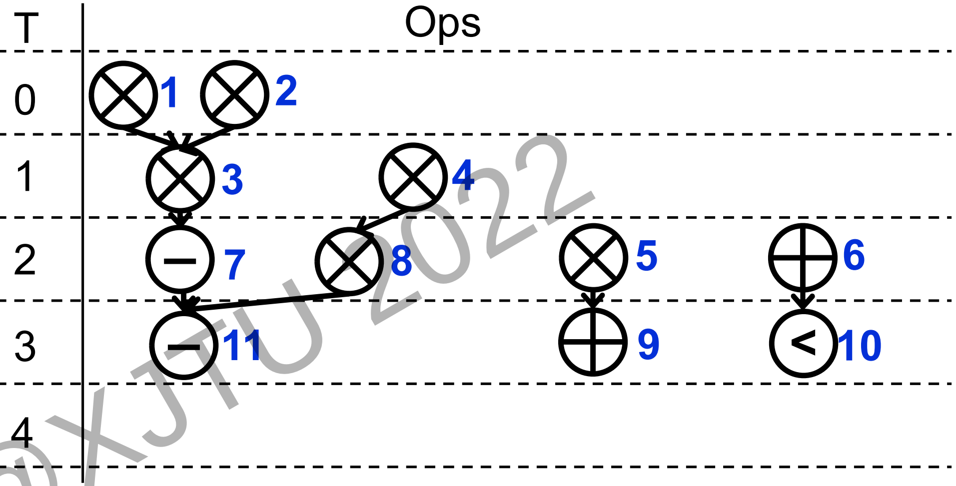
$\}$



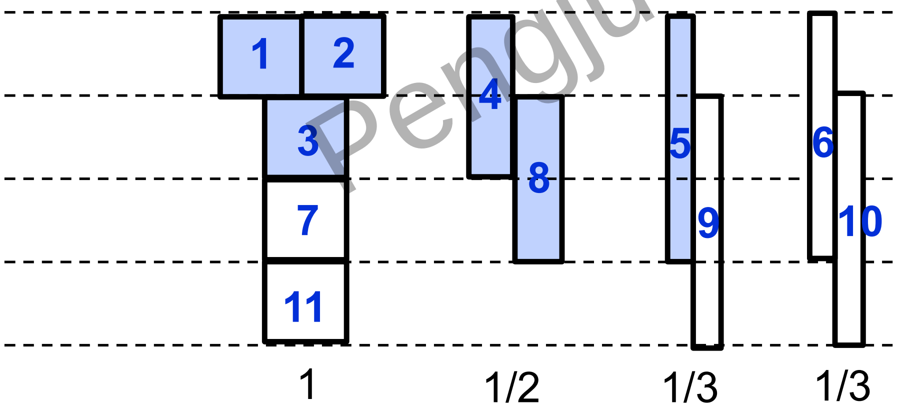
Example： Force-Directed Scheduling (2)



ASAP Scheduling



ALAP Scheduling



Distribution Graph

Multiplication	Addition
$2+\frac{1}{2}+\frac{1}{3}=2.833$	$\frac{1}{3}=0.333$
$1+\frac{1}{2}+\frac{1}{2}+\frac{1}{3}=2.333$	$\frac{1}{3}+\frac{1}{3}+\frac{1}{3}=1$
$\frac{1}{2}+\frac{1}{3}=0.833$	$1+\frac{1}{3}+\frac{1}{3}+\frac{1}{3}=2$
0	$1+\frac{1}{3}+\frac{1}{3}=1.666$

Execution time frame and associated probabilities

Example： Force-Directed Scheduling (3)

For each possible assignment of a node, n_j,to a time step TS_j, within the node's time frame ($S_j \le TS_j \le L_j$), a self force is calculated as:

$$self\_force\ \ (\mathrm{j}) = \sum_{i=S_j}^{L_j} DG(i) \times x(i)$$

Where $DG(i)$ is the distribution value at time step i, and $x(i)$ is the change in the probability associated with time step i. The predecessor and successor forces are the self forces of predecessor and /or successor nodes that will be affected by the assignment of a node to a time step.

The **total force** is the summation of the self, the predecessor and the successor forces.

Example：Force-Directed Scheduling (4)

Taking node 4 as an example:

$$self\_force_4(1) = force_4(1) + force_4(2)$$
$$= (DG_M(1)^* \, x_4(1)) + (DG_M(2)^* \, x_4(2))$$
$$= (2.833 * (1 - 0.5)) + (2.333 * (0 - 0.5))$$
$$= 0.25$$

$$self\_force_4(2) = force_4(1) + force_4(2)$$
$$= (DG_M(1)^* \, x_4(1)) + (DG_M(2)^* \, x_4(2))$$
$$= (2.833 * (-0.5)) + (2.333 * (0.5))$$
$$= -0.25$$

Node 4 scheduled into time step 1,will not affect its successor(node 8) therefore, the Total $force_4(1) = +0.25$

Example： Force-Directed Scheduling (4)

If node 4 scheduled into time step 2,will not affect its successor (node 8) to be scheduled into time step 3.

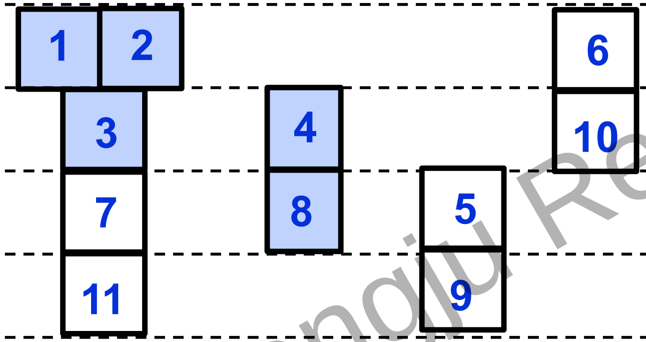
$$
\begin{aligned}
Succ\_force_4(2) &= force_8(2) + force_8(3) \\
&= (DG_M(2)^* \, x_8(2)) + (DG_M(3)^* \, x_8(3)) \\
&= \big(2.333 * (0 - 0.5)\big) + (0.833 * (1 - 0.5)) \\
&= -0.75
\end{aligned}
$$

$$
\begin{aligned}
force_4(2) &= self\_force_4(2) + Succ_{force_4}(2) \\
&= -0.25 - 0.75 = -1.0
\end{aligned}
$$

Therefor the lowest force is chosen and node 4 is scheduled to time step 2.

Example：Force-Directed Scheduling (5)

Repeat the process until all the other nodes are scheduled.



Distribution Graph

Multiplication	Addition
2	1
2	1
1	2
0	2

Execution time frame and associated probabilities

Next Lecture： Systolic Array