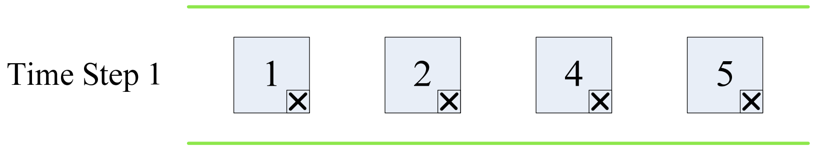
VLSI Signal Processing

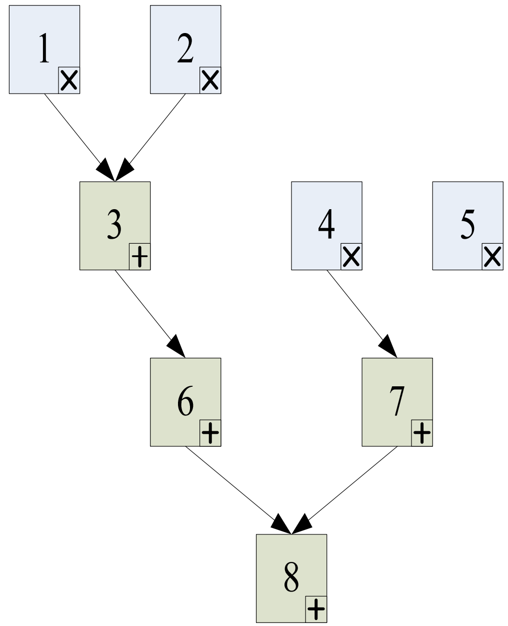
Lecture 3 Scheduling Techniques

Appendix B: Scheduling and Allocation Techniques

- ASAP(as soon as possible) Scheduling
- ALAP (as late as possible) Scheduling
- List Scheduling
- Force-directed Scheduling
- ILP (Integer Linear Programming) Scheduling

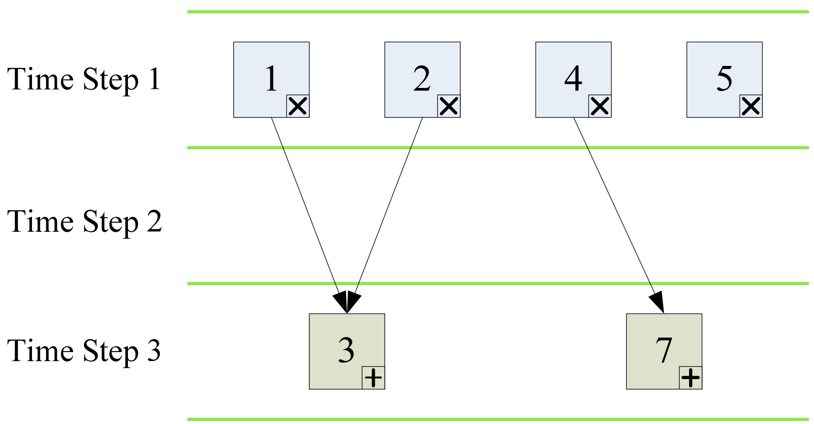
ASAP(as soon as possible) Scheduling

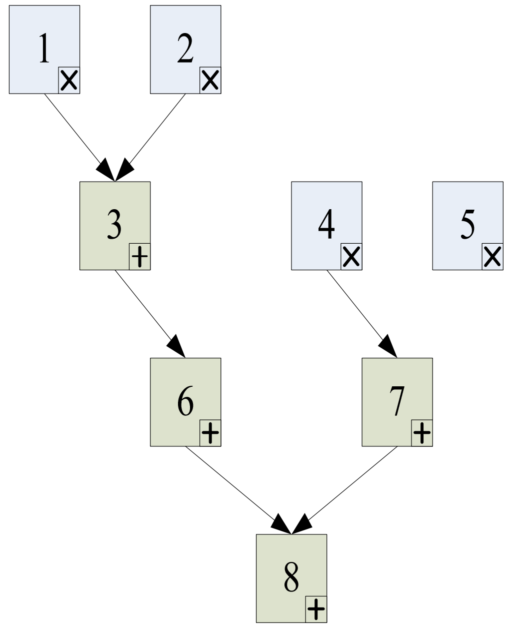




Assumption : unlimited resource

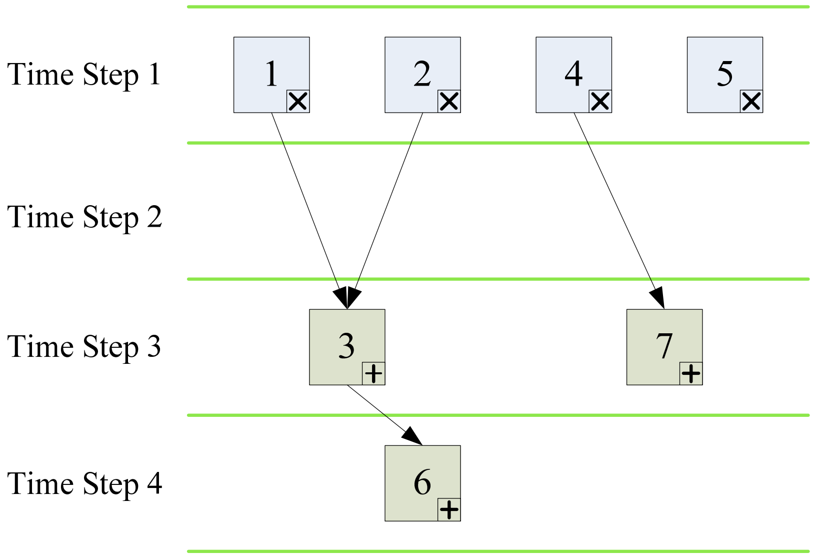
ASAP(as soon as possible) Scheduling

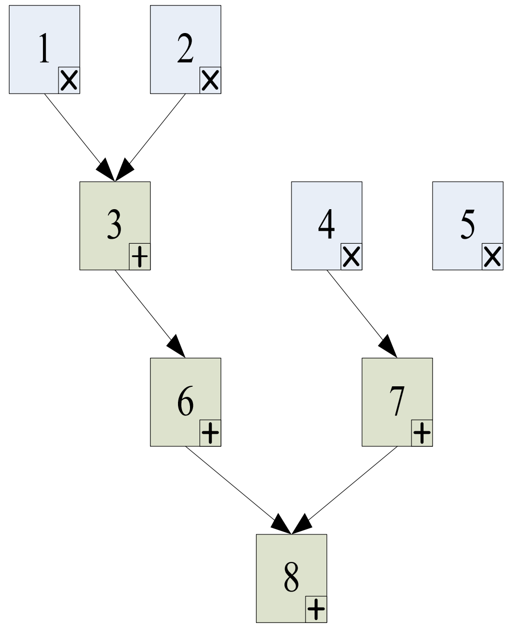




Assumption : unlimited resource

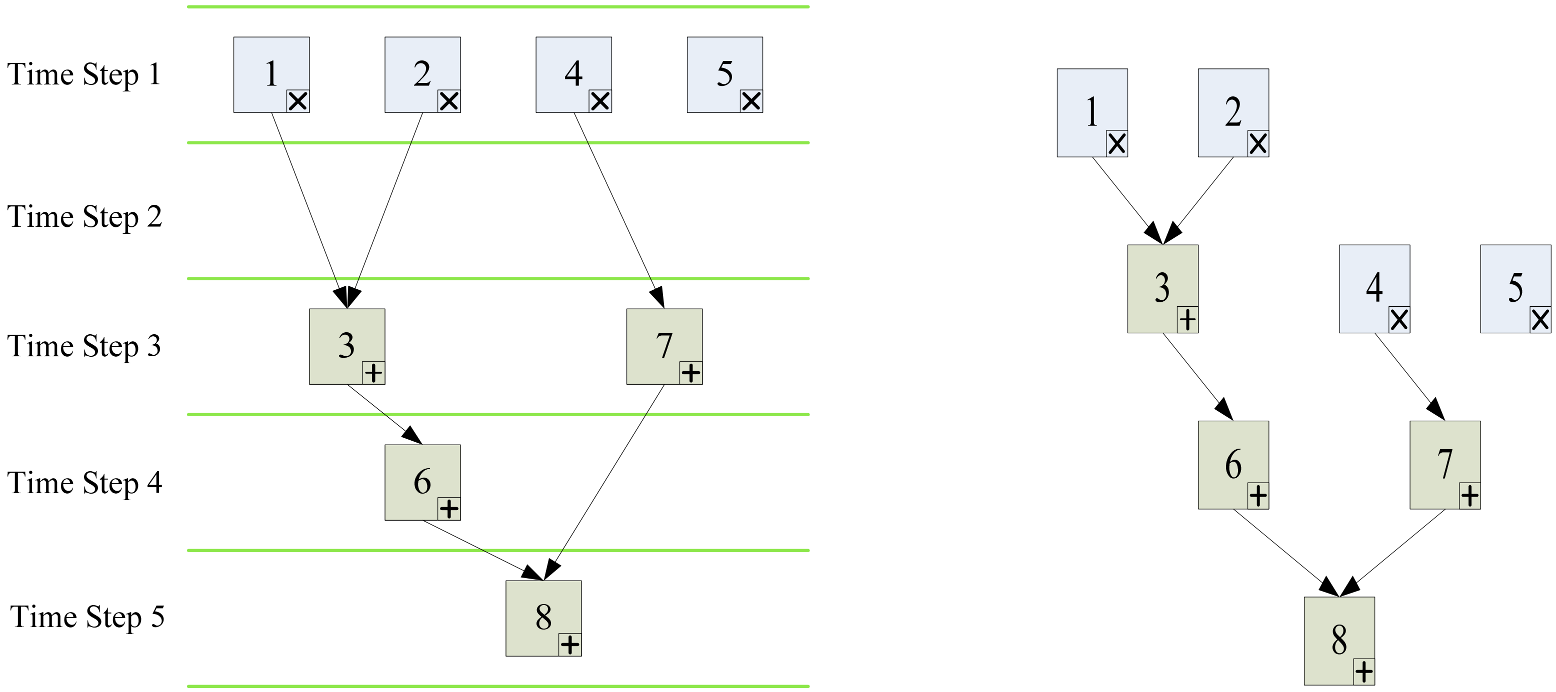
ASAP(as soon as possible) Scheduling





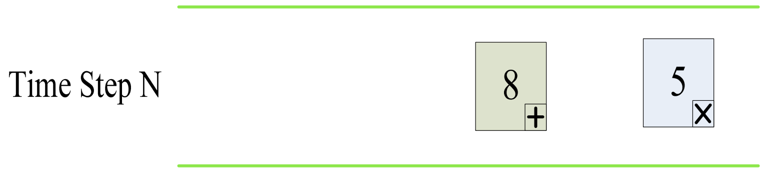
Assumption : unlimited resource

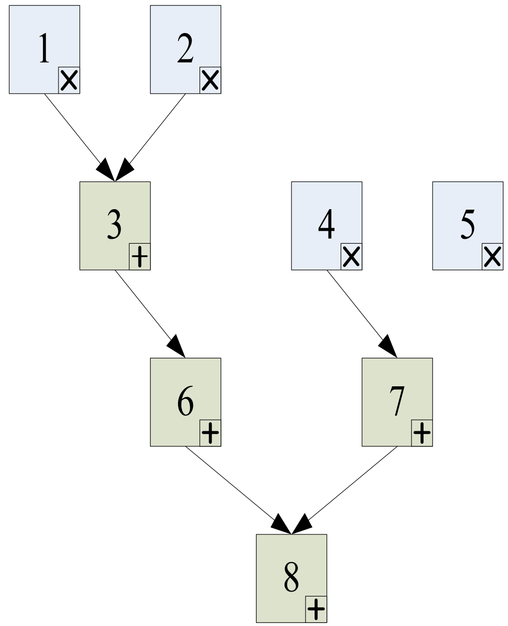
ASAP(as soon as possible) Scheduling



Assumption : unlimited resource

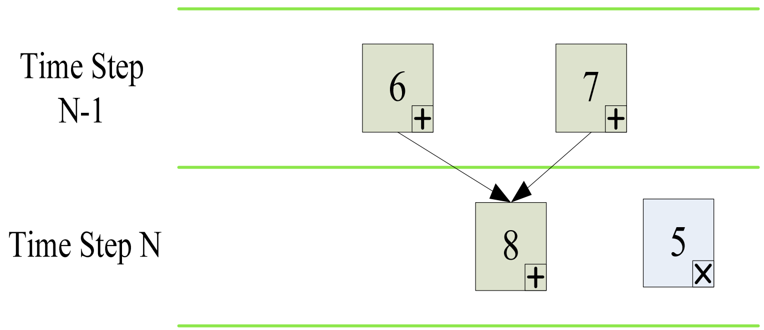
ALAP (as late as possible) Scheduling

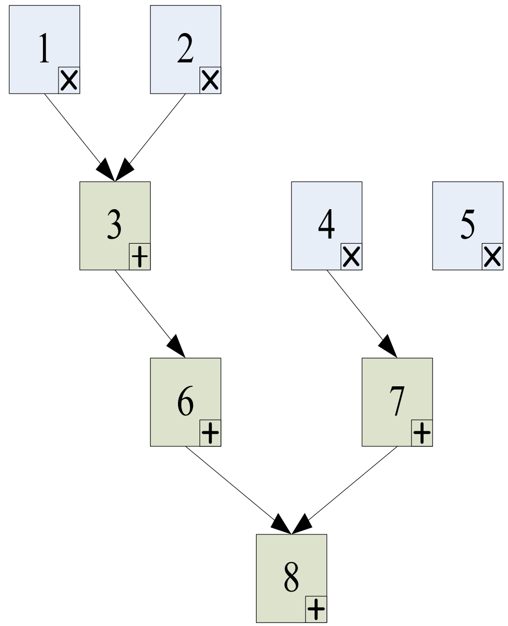




Assumption : unlimited resource

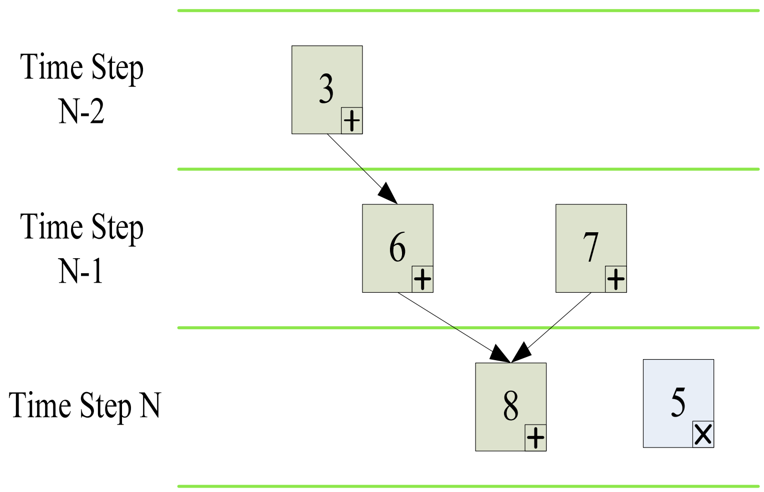
ALAP (as late as possible) Scheduling

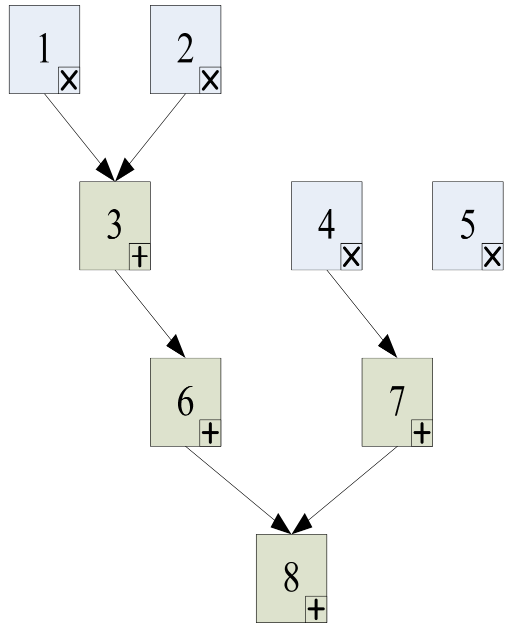




Assumption : unlimited resource

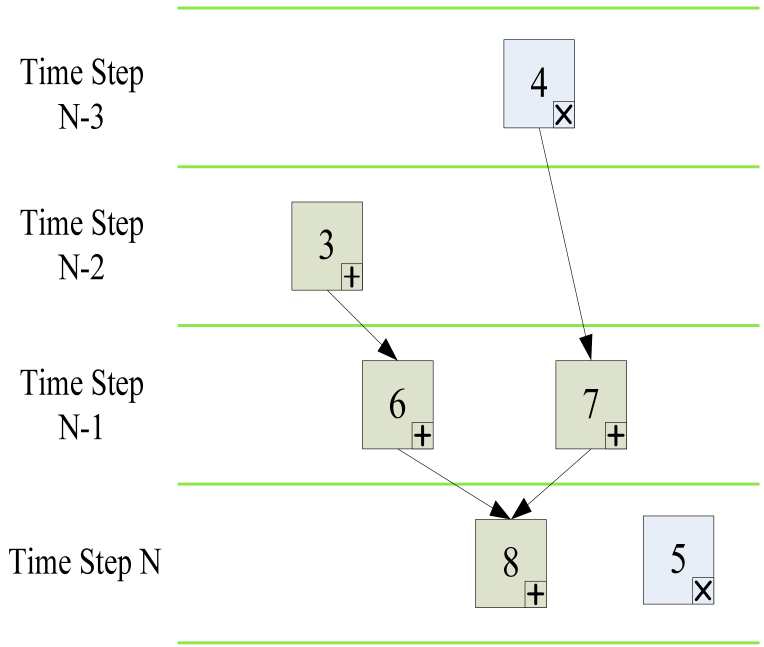
ALAP (as late as possible) Scheduling

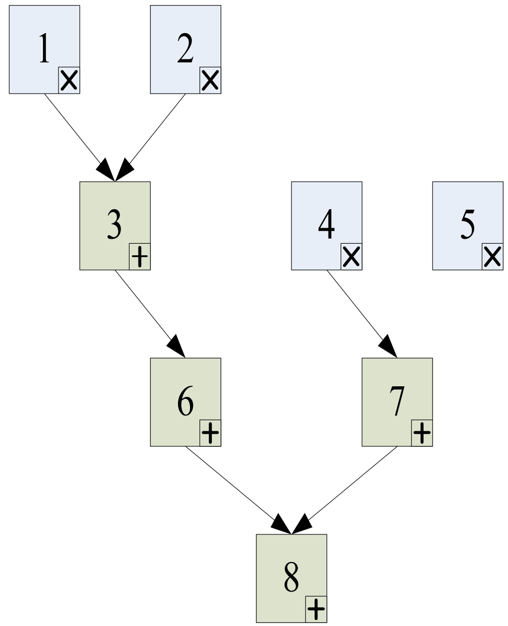




Assumption : unlimited resource

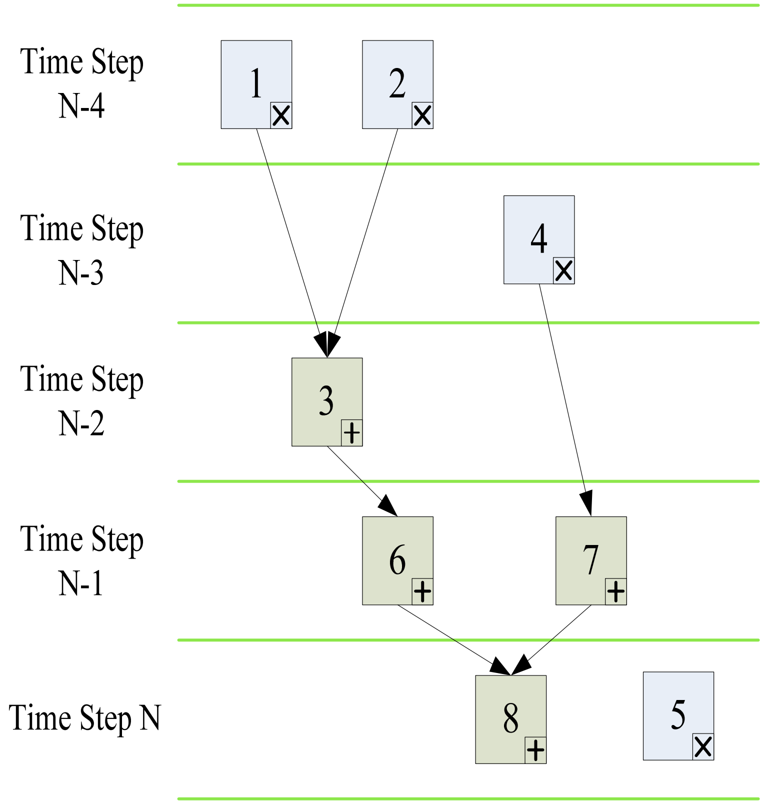
ALAP (as late as possible) Scheduling

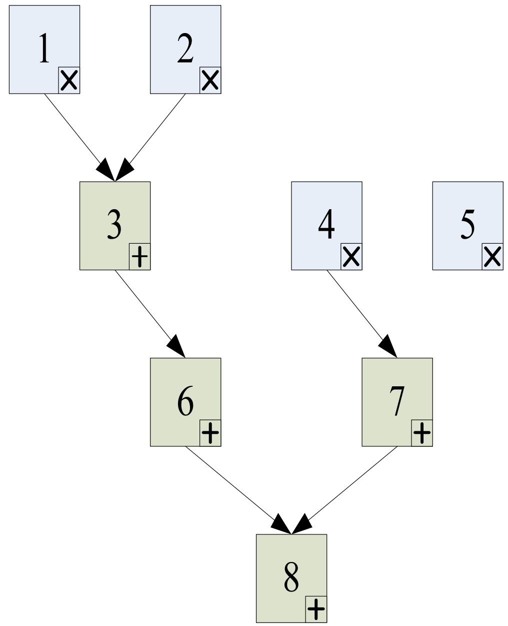




Assumption : unlimited resource

ALAP (as late as possible) Scheduling





Assumption : unlimited resource

Several Issues About ASAP and ALAP

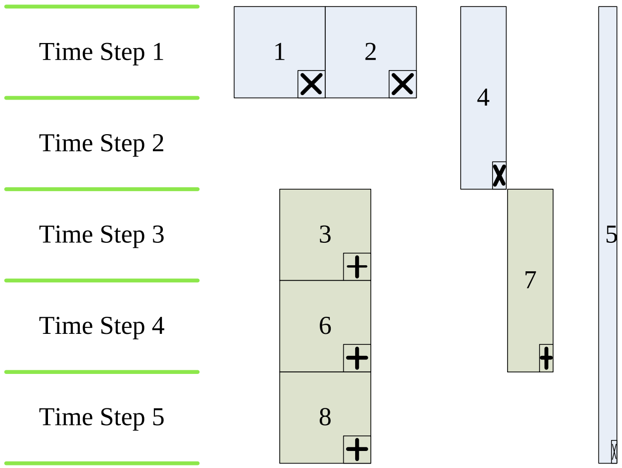
- Evaluation
  - Hardware resource
  - Feasible iteration period
- Shrink solution space

List Scheduling

- Heuristic
- Modified ASAP
- Order of unscheduling node determined by a priority function

Force-directed Scheduling (1/2)

- Time frame (the likely scheduling positions got from ASAP and ALAP)
- Distributed graph (the sum of different type operation probability in each time Step)



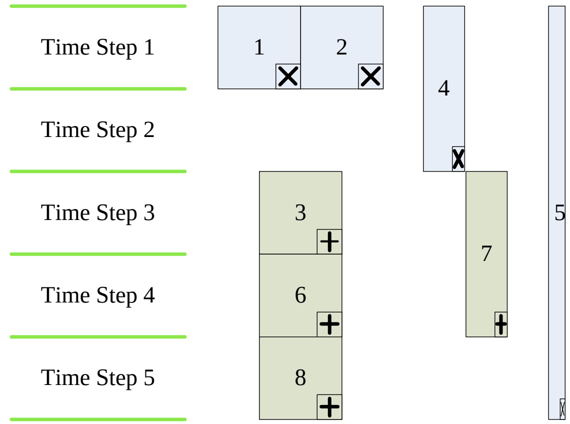
(a) time frame

ADD	MPY
0	2.7
0	0.7
1.5	0.2
1.5	0.2
1	0.2

(b) distributed graph

Force-directed Scheduling (2/2)

- Self force (the effect trying to schedule a node to the time step)
- Predecessor/Successor force



(a) time frame

ADD	MPY
0	2.7
0	0.7
1.5	0.2
1.5	0.2
1	0.2

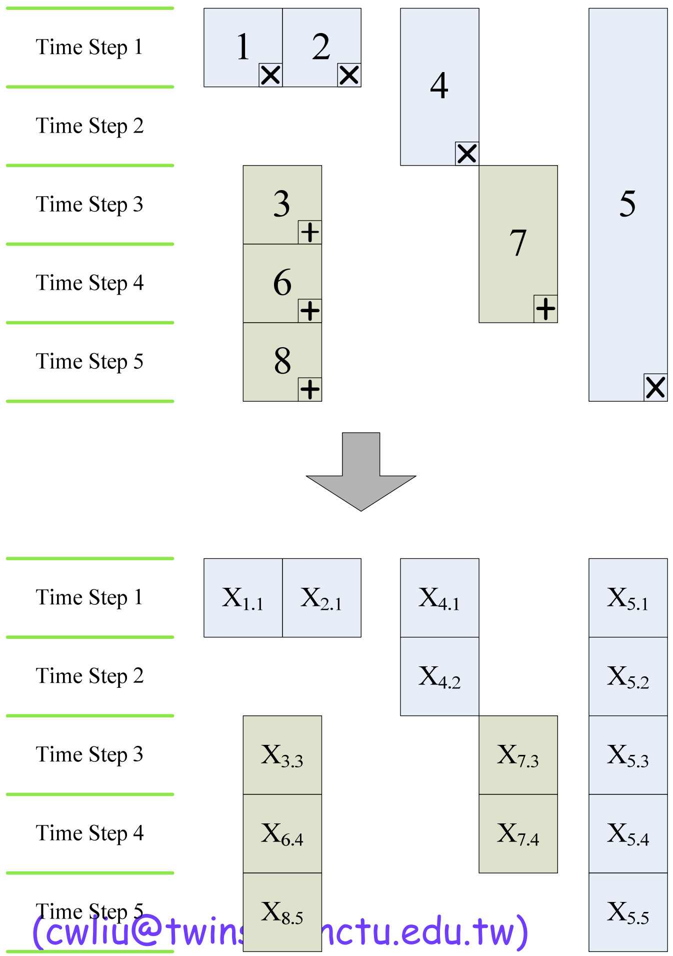
(b) distributed graph

```
Self_force4(1) = 2.7 * (1 – 0.5) + 0.7 * (0.5 - 1) = 1
Self_force4(2) = 2.7 * (0.5 - 1) + 0.7 * (1 – 0.5) = -1
```

```
Succ_force4(1) = 0
Succ_force4(2) = 1.5 * (0 – 0.5) + 1.5 * (1 – 0.5) = 0
```

ILP (Integer Linear Programming) Scheduling (1/2)

- Constrain based
- Model in binary decision variable
- High computation complexity



ILP (Integer Linear Programming) Scheduling (2/2)

- Position constrains

$X_{1.1} = 1$
$X_{4.1} + X_{4.2} = 1$
$X_{5.1} + X_{5.2} + X_{5.3} + X_{5.4} + X_{5.5} = 1$

- Resource constrains

For multipliers

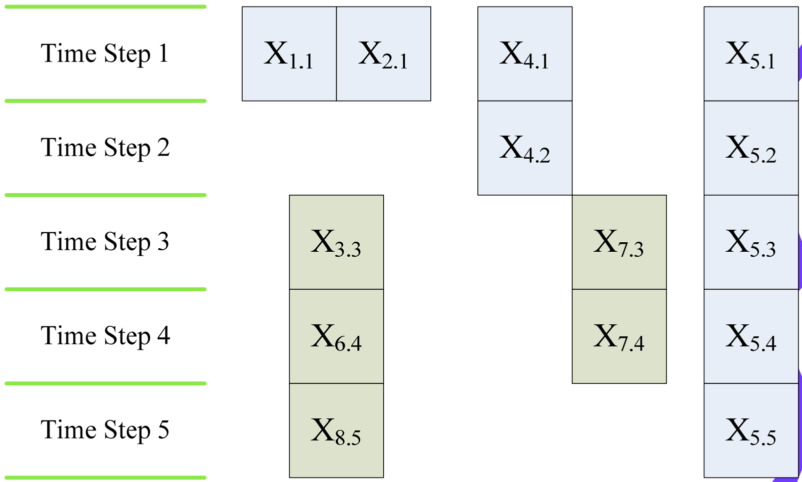
X1.1 + X2.1 + X4.1 + X5.1 <= Nmpy
X4.2 + X5.2 <= Nmpy

For adders

X3.3 + X7.3 <= Nadd
X6.4 + X7.4 <= Nadd

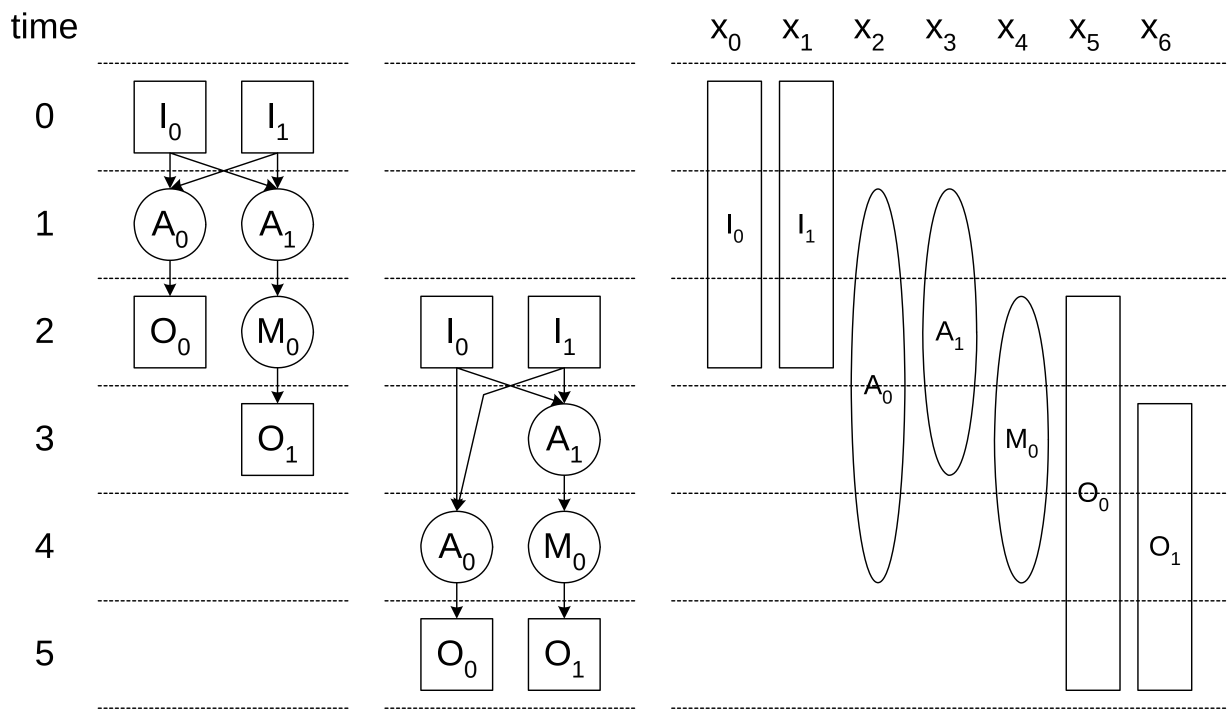
- Dependency constrains

$3X_{3.3} - X_{2.1} >= 2$
$3X_{3.3} - X_{1.1} >=2$
$3X_{7.3} + 4X_{7.4} - 2X_{4.2} - X_{4.1} >= 2$



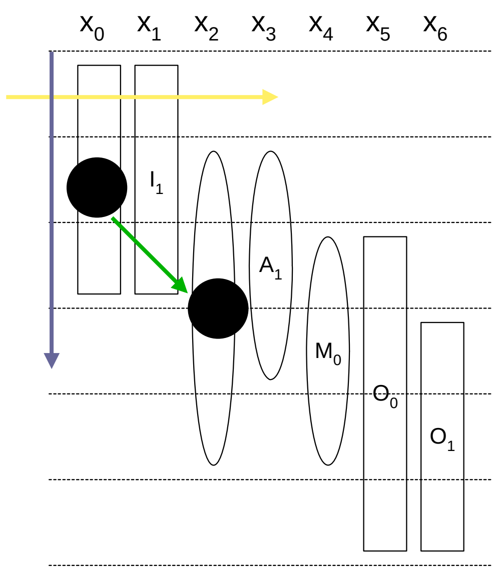
Basic Scheduling Algorithms

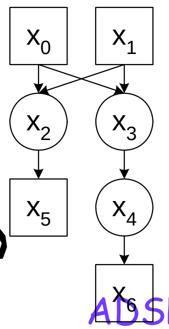
- As soon as possible (ASAP) & as late as possible (ALAP)
- Scheduling ranges



(a) (b) (c)

ILP-based Scheduling





The Boolean variable $x_{i,j}$ indicates if the vertex i is scheduled at time j

- Resource constraints (operations cannot exceed the resources)
  - $x_{0.0}+x_{1.0} \le 1;\ x_{0.1}+x_{1.1} \le 1;\ x_{0.2}+x_{1.2} \le 1$ (for input)
  - $x_{2.1}+x_{3.1} \le 1;\ x_{2.2}+x_{3.2} \le 1;\ x_{2.3}+x_{3.3} \le 1$ (for adder)
  - $x_{5.3}+x_{6.3} \le 1;\ x_{5.4}+x_{6.4} \le 1;\ x_{5.5}+x_{6.5} \le 1$ (for output)
- Allocation constraints (each node executes only once)
  - $x_{0.0}+x_{0.1}+x_{0.2} = 1$
  - $x_{1.0}+x_{1.1}+x_{1.2} = 1$
  - $x_{2.1}+x_{2.2}+x_{2.3}+x_{2.4} = 1$
  - $x_{3.1}+x_{3.2}+x_{3.3} = 1$
  - $x_{4.2}+x_{4.3}+x_{4.4} = 1$
  - $x_{5.2}+x_{5.3}+x_{5.4}+x_{5.5} = 1$
  - $x_{6.3}+x_{6.4}+x_{6.5} = 1$
- Dependency constraints (for each edge)
  - $x_{0.0} + 2\,x_{0.1} + 3\,x_{0.2} - 2\,x_{2.1} - 3\,x_{2.2} - 4\,x_{2.3} - 5\,x_{2.4} \le -1$
  - $x_{1.0} + 2\,x_{1.1} + 3\,x_{1.2} - 2\,x_{2.1} - 3\,x_{2.2} - 4\,x_{2.3} - 5\,x_{2.4} \le -1$
  - $x_{0.0} + 2\,x_{0.1} + 3\,x_{0.2} - 2\,x_{3.1} - 3\,x_{3.2} - 4\,x_{3.3} \le -1$
  - $x_{1.0} + 2\,x_{1.1} + 3\,x_{1.2} - 2\,x_{3.1} - 3\,x_{3.2} - 4\,x_{3.3} \le -1$
  - $2\,x_{2.1} + 3\,x_{2.2} + 4\,x_{2.3} + 5\,x_{2.4} - 3\,x_{5.2} - 4\,x_{5.3} - 5\,x_{5.4} - 6\,x_{5.5} \le -1$
  - $2\,x_{3.1} + 3\,x_{3.2} + 4\,x_{3.3} - 3\,x_{4.2} - 4\,x_{4.3} - 5\,x_{4.4} \le -1$
  - $3\,x_{4.2} + 4\,x_{4.3} + 5\,x_{4.4} - 4\,x_{6.3} - 5\,x_{6.4} - 6\,x_{6.5} \le -1$