Timing Optimization

Optimization of Timing

- Three phases
  1. globally restructure to reduce the maximum level or longest path

     Ex: a ripple carry adder ==>

     a carry look-ahead adder

  2. physical design phase

     – transistor sizing

     – timing driven placement

     – buffering

  3. actual design

     – fine tune the circuit parameter

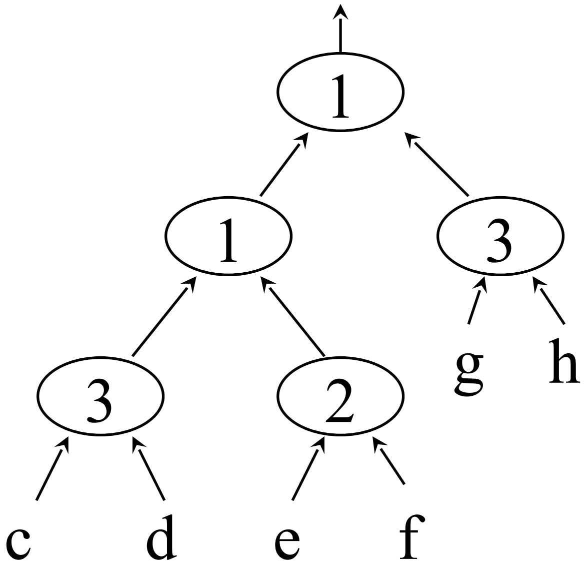
Delay Model at Logic Level

1. Unit delay model
   - assign a delay of 1 (or gate delay) to a gate

2. Unit fanout delay model
   - incorporate an additional delay for each fanout

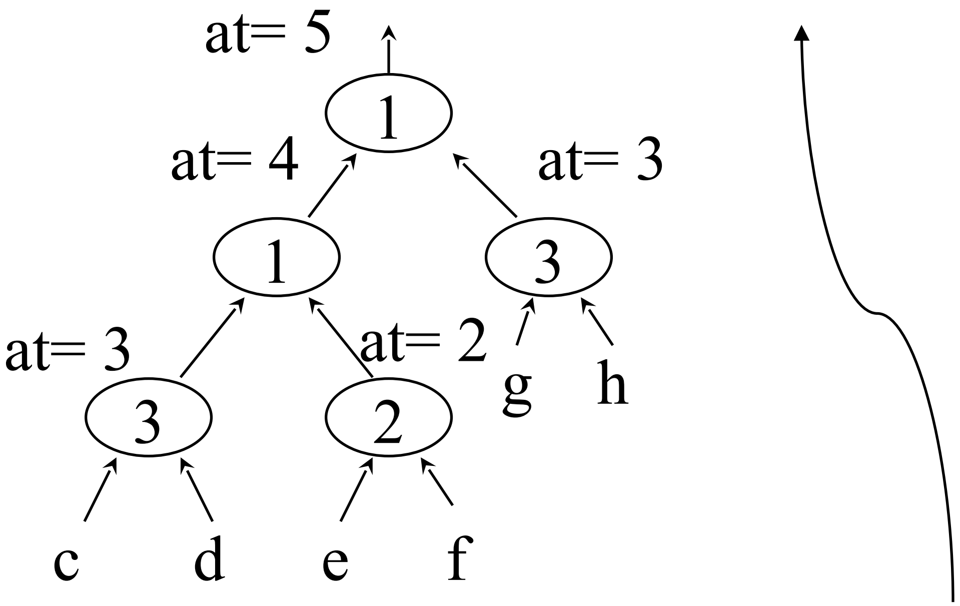
3. Library delay model
   - use delay data in the library to provide more accurate delay value

Arrival Time & Required Time



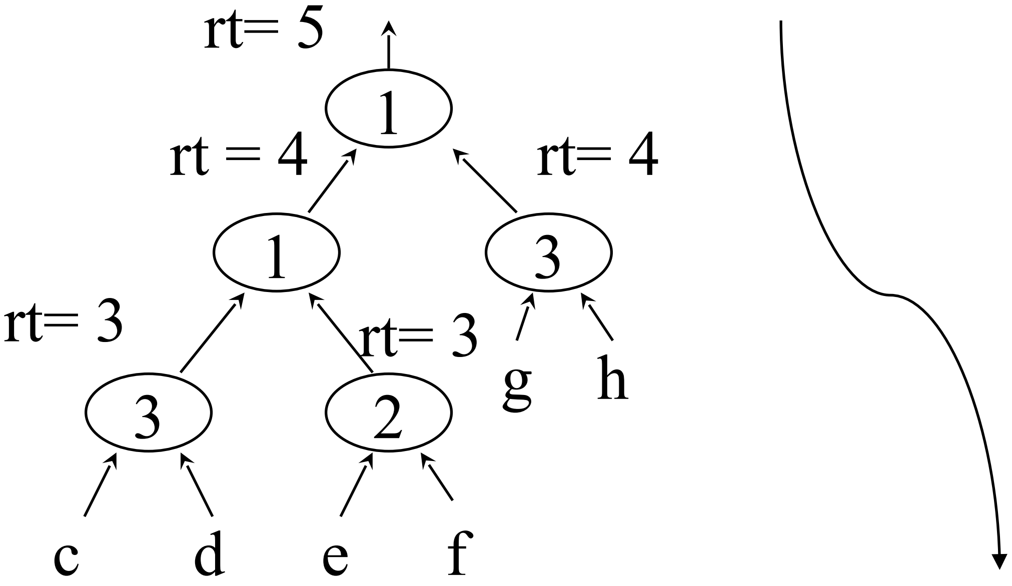
- Arrival time : from input to output
- Required time : from output to input
- Slack = required time - arrival time

Arrival Time



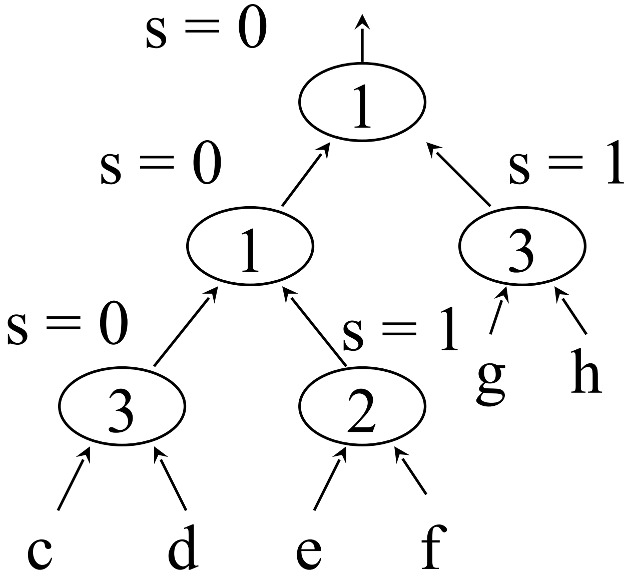
- Arrival time (at) : from input to output
- Required time (rt) : from output to input
- Slack = required time - arrival time

Required Time



- Arrival time (at) : from input to output
- Required time (rt) : from output to input
- Slack = required time - arrival time

Slack Time



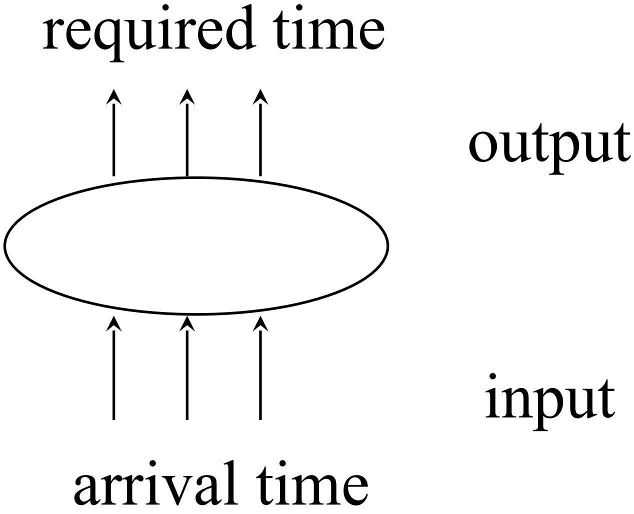
- Arrival time of output : 5
- Required time of output : 5
- Slack (s) = required time - arrival time

Timing Optimization in SIS

Restructure for Timing [SIS]

Two Steps:

- minimize area
- speed up



required time

output

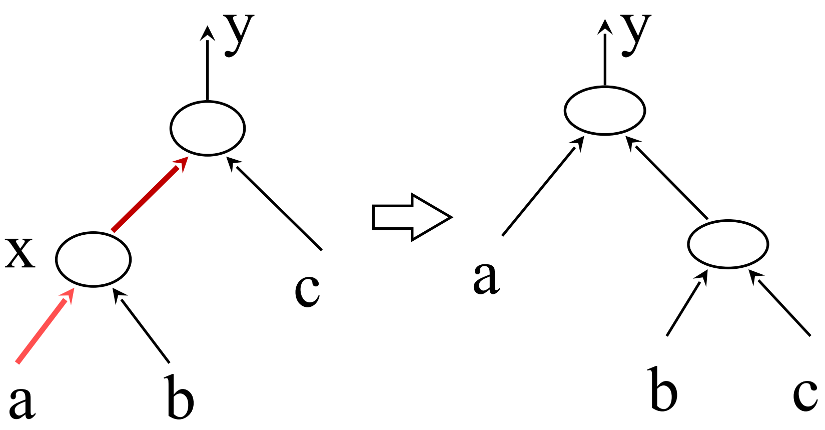
input

arrival time

critical node = with negative slack time

Basic Idea

- collapse critical nodes and re-decompose



critical path a-x-y

Speed Up Algorithm

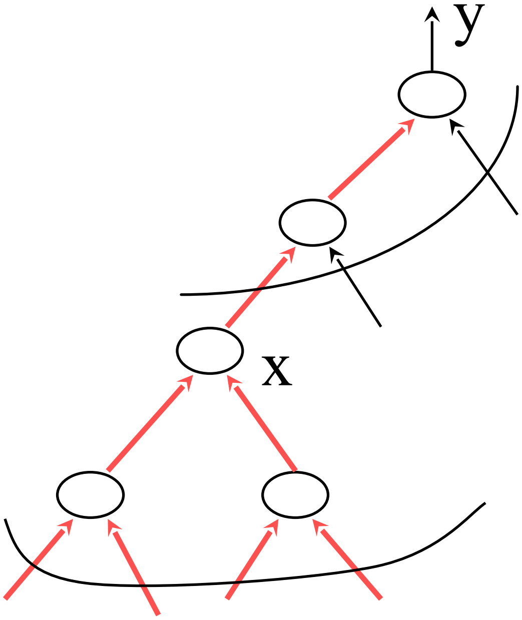
```
speed up (d)
1  compute the slack time of each node
2  find all critical nodes and compute cost for
   each critical node
3  select re-synthesis points (find minimum
   cut set of all critical node)
4  collapse and re-decompose the re-synthesis
   points
5  if timing requirement is satisfied, done.
   otherwise go to step 1
```

Speed Up (Conti.)

Step 2 :

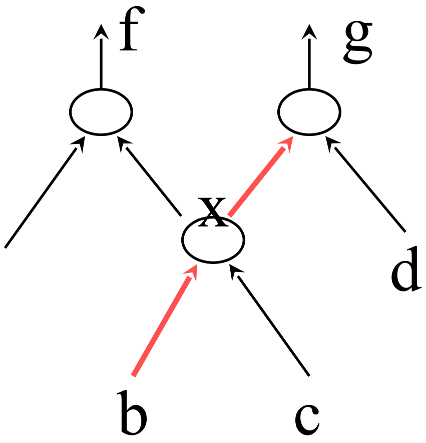
- compute cost function
  - selecting re-synthesis points has to consider
    - (1) ease for speed-up (re-synthesis)
    - (2) area overhead

Ease for Speed-Up

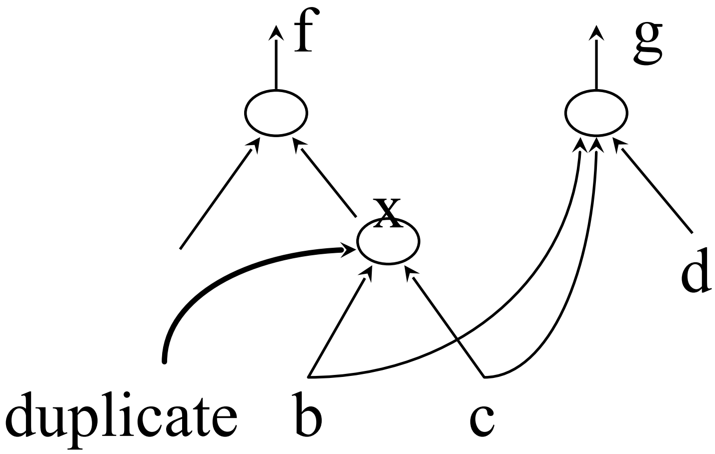


- let d = 1 (collapsing depth, given)
  y => 1 critical input
  2 non-critical inputs
  x => 4 critical inputs
- If y is chosen, it will be easier to perform re-decomposition.

Area Penalty



b-x-g critical
collapse x into g



Area Penalty (Conti.)

- define weight for critical node X

  $W_x(d) = W_x^t(d) + \alpha \star W_x^a(d)$

  - $W_x^t(d)$ reflect the ease for speed up
  - $W_x^a(d)$ reflect area increase

N(d) = signals that are input to re-synthesis region

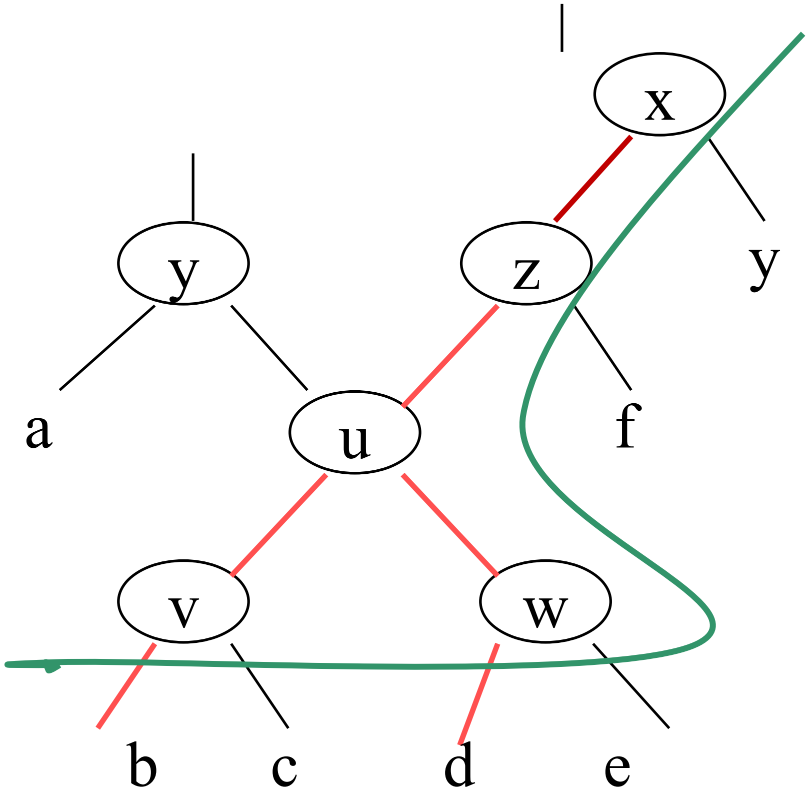
M(d) = nodes in the re-synthesis region

$$W_x^t(d) = \frac{\left| \left\{ y \in N(d) \middle| Sy \le \varepsilon \right\} \right|}{\left| N(d) \right|}$$

$$W_x^a(d) = \frac{\left| \left\{ y \in M(d) \middle| y \textit{ is shared} \right\} \right|}{M(d)}$$

Area Penalty (Conti.)

Ex:



d=3

$W_x^t(d) = 2/6$

$W_x^a(d) = 3/5$

Speed Up Algorithm

speed up(d)

1. compute the slack time of each node
2. find all critical nodes and compute cost for each critical node
3. select re-synthesis points
4. collapse and re-decompose the re-synthesis points
5. if timing requirement is satisfied, done. otherwise go to step 1

Selection of Re-synthesis Point

Now, cost function is defined,

How to select re-synthesis point ?

- Greedy algorithm?
    - Very local
- Any other more global algorithm ….

Speed Up (Conti.)

Step 3 :

Background:

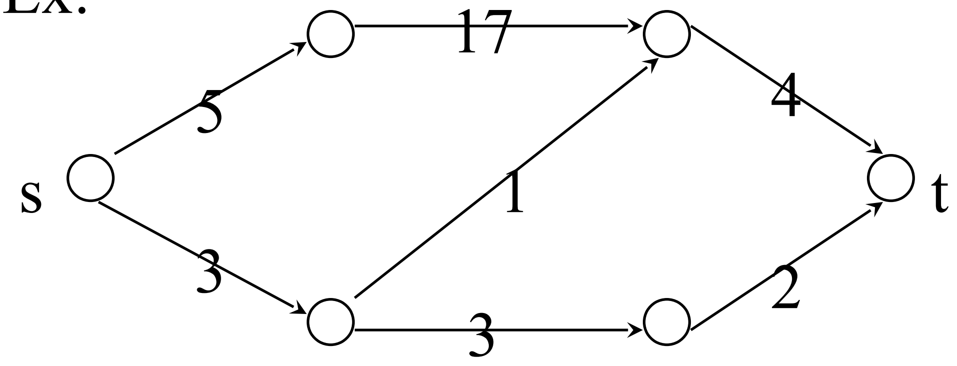
A network N=(s,t,V,E,b) is a diagram (V, E) together with a source s $\in$ V and a sink t $\in$ V with bound (capacity),

$$b(u,v) \in Z^{+} \text{ for all edges.}$$

A flow f in N is a vector in $R^{|E|}$ such that

1. $0 \leq f(u,v) \leq b(u,v)$ for all $(u,v) \in E$
2. $\sum_{(u,v)\in E} f(u,v) = \sum_{(u,w)\in E} f(v,w)$ for all $v \in V-\{s,t\}$

Ex:

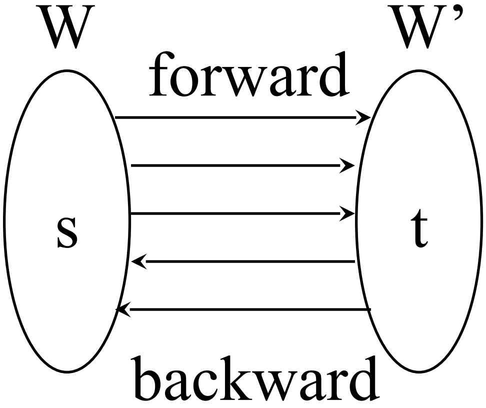


The value of the max flow |f| =6

Speed Up (Conti.)

An s-t cut is a partition (W,W’) of the nodes of V into sets W and W’ such that s ∈W and t ∈W’. The capacity of an s-t cut

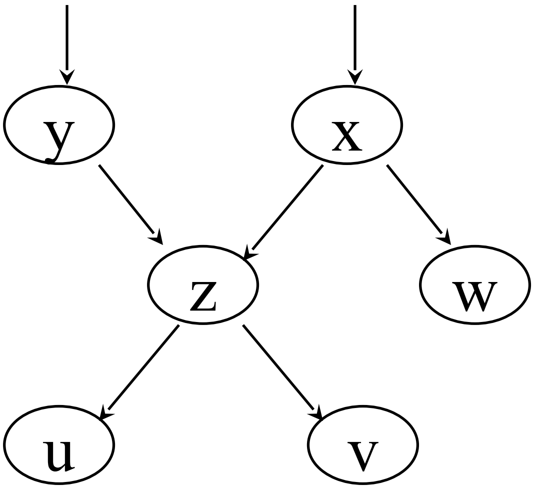
$$c(W,W') = \sum_{\substack{(i,j)\in E \\ \text{such that} \\ i\in w,\, j\in w'}} b(i,j)$$



Max-flow = min-cut

Speed Up (Conti.)

Ex:



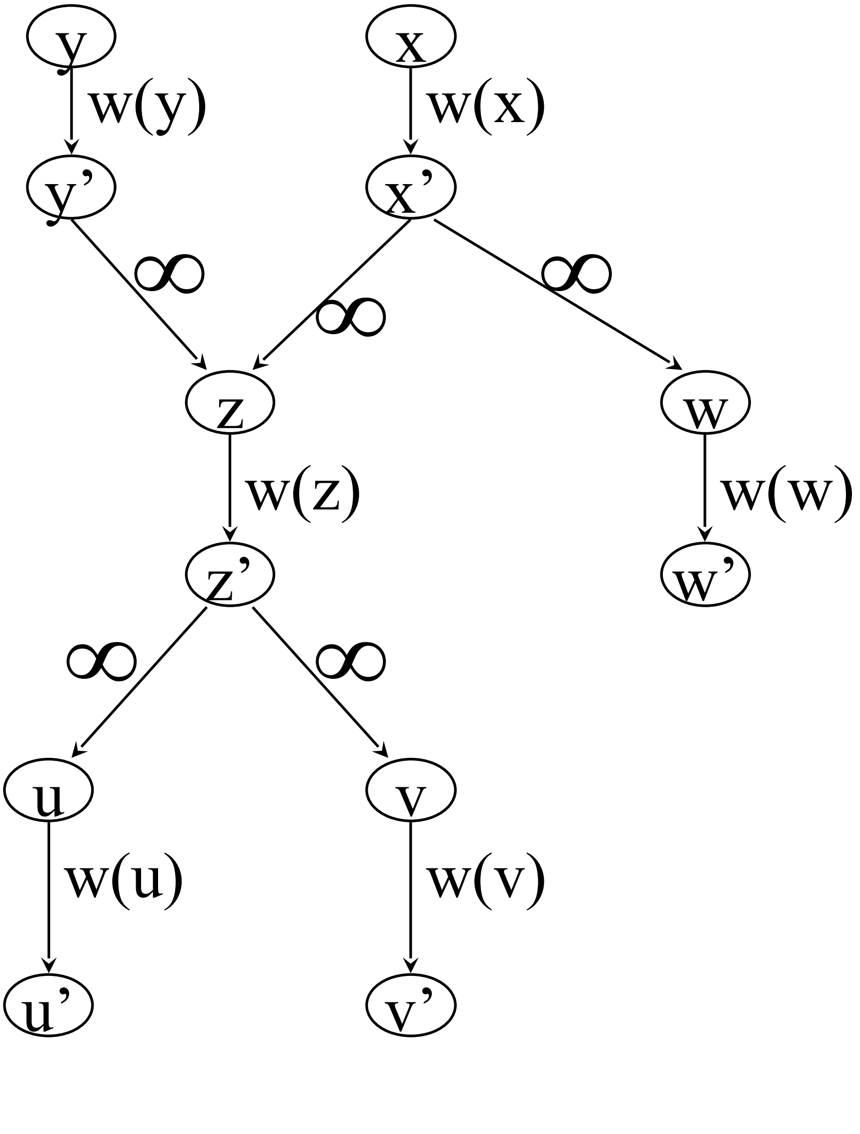
=> Network flow

But we want to have node cut not edge cut ?

Speed Up (Conti.)

Step 3:

Duplicate each node



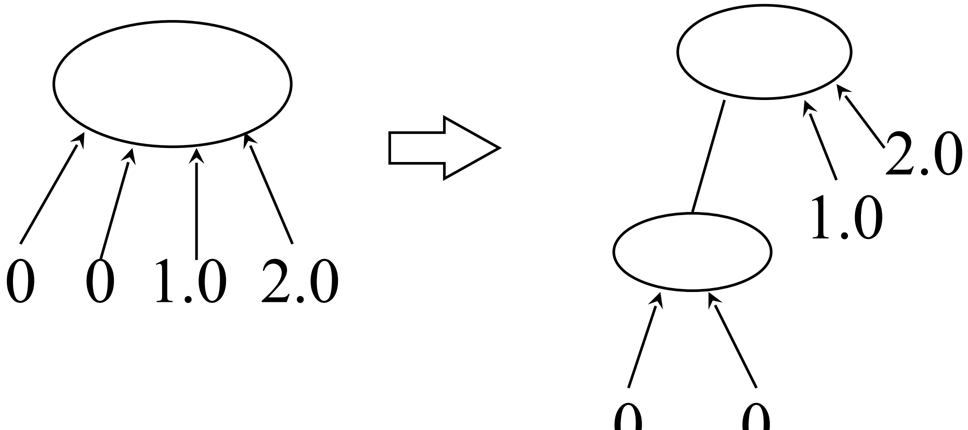
use maxflow(min-cost) algorithm to find resysthesis points

Speed Up (Conti.)

Step 4 :

Re-decompose

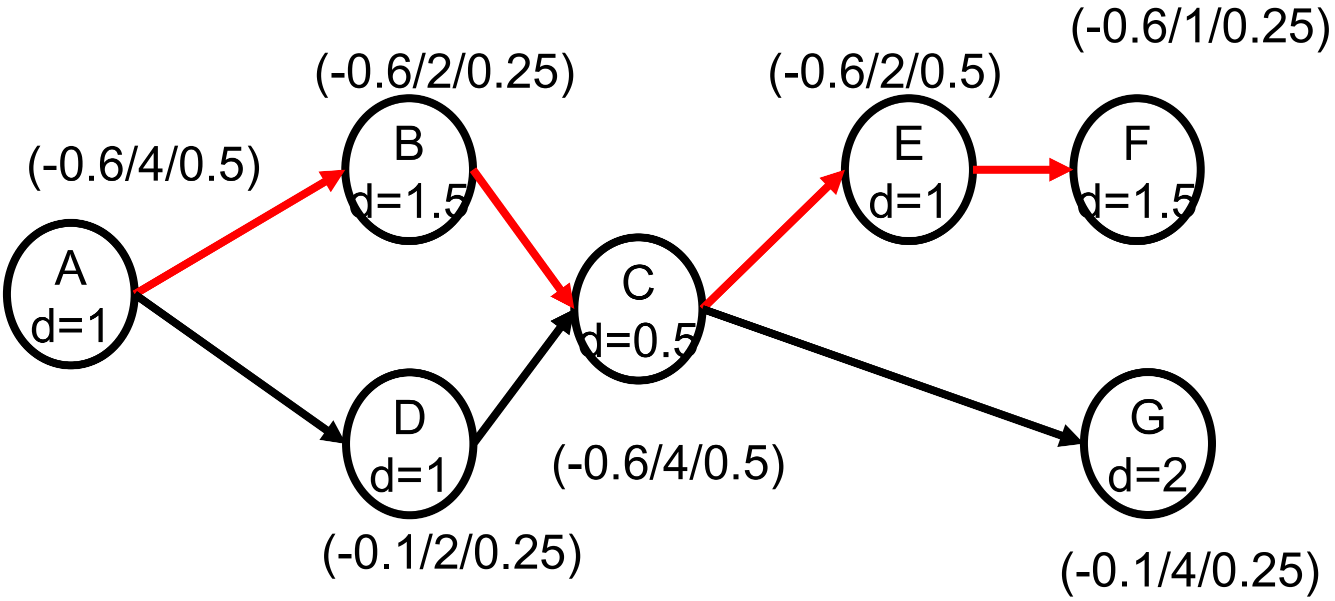
1. kernel based decomposition
   - extract divisor
   - the weight of a divisor is a linear sum of area component (literal saved) and time component (prefer the smallest arrival time)
2. and-or decomposition



Further Improvement of Cut Set (ICCAD 1999)

An Improved Cut Set (Separator Set)

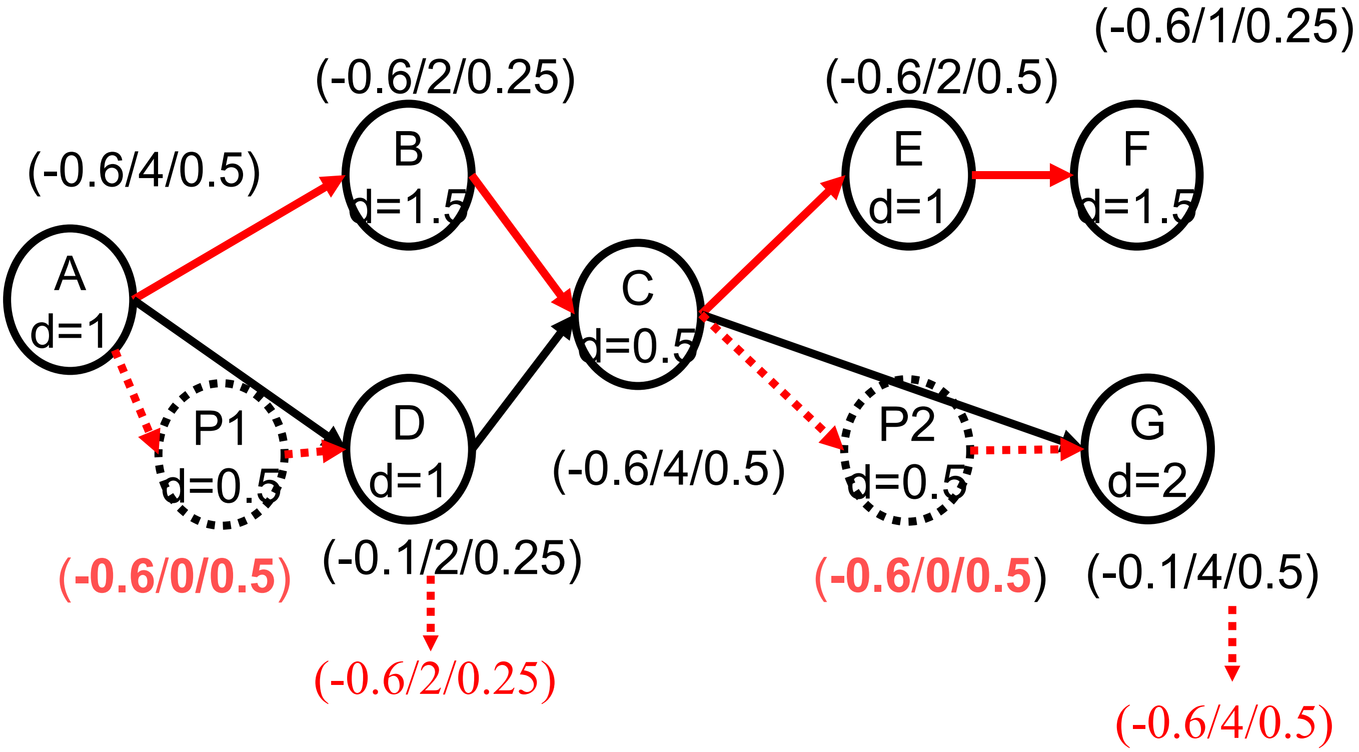
- Let required time = 4.9
- Un-balanced path delay
- Minimum cost cut set = 4 ({C})
- Delay reduction = 0.5



(x,y, z) means (slack, cost, delay reduction)

Construct a Path-balanced Graph

- *ds(e) = slack (HeadNode (e))– slack (TailNode(e))*
- If *ds(e)* > 0, insert a “padding node”
- P1 and P2 are two padding nodes
- Minimum cost cut-set = 2 ({E, P2})
- Delay reduction = 0.5



(x,y, z) means (slack, cost, delay reduction)

Techniques Used in Other Optimization Steps

- Gate sizing
- Low power design (threshold voltage assignment)
  - high threshold voltage:
    - leakage power↓
    - delay↑
  - low threshold voltage:
    - leakage power ↑
    - delay↓