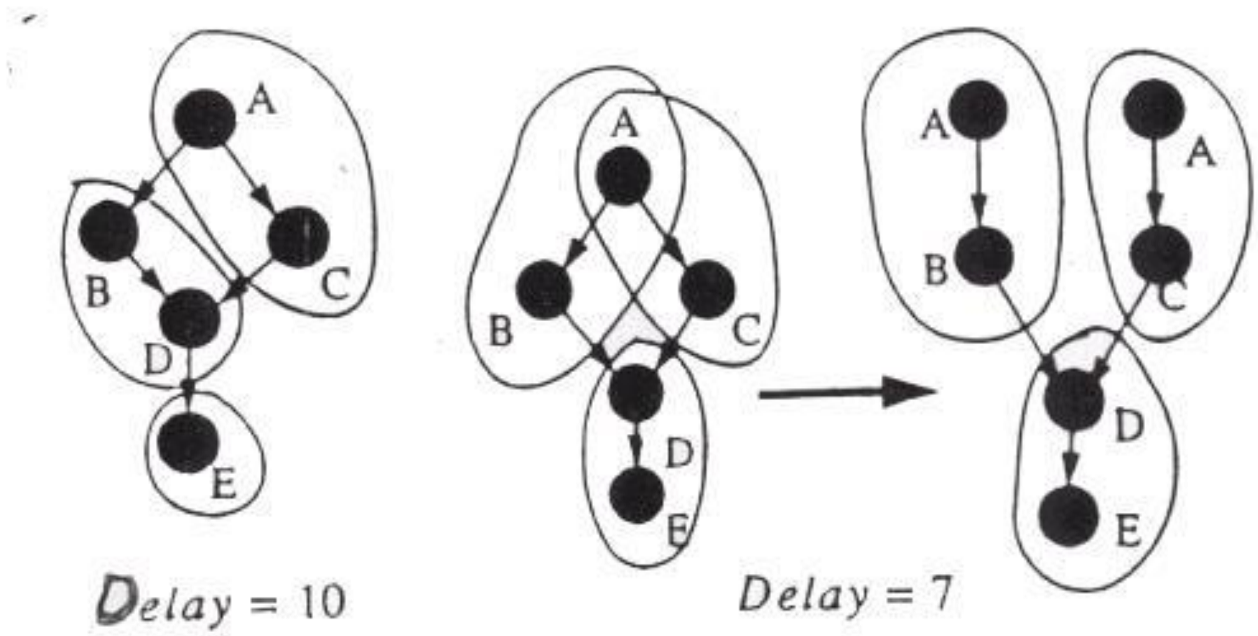
Clustering for Delay Minimization

- Allow gate duplication.
- Gate duplication may help reduce delay.

D=3; M=2; $\delta(v)$=1, $w(v)$=1, for each v.

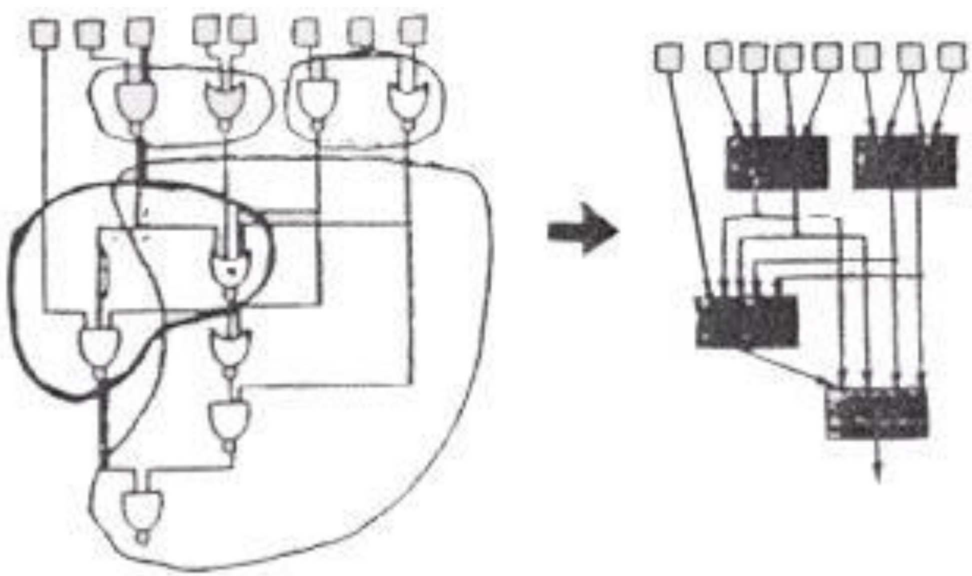
Without gate duplication   With gate duplication



Unit Delay Model

- No gate delay.
- No interconnection delay within a cluster.
- Delay of 1 unit for an interconnection between 2 clusters.
- An optimal algorithm for area constraint only (Lawler, Levitt and Turner, IEEE TC, 1966).
- An optimal algorithm for pin constraint only (Cong and Ding, ICCAD, 1992).

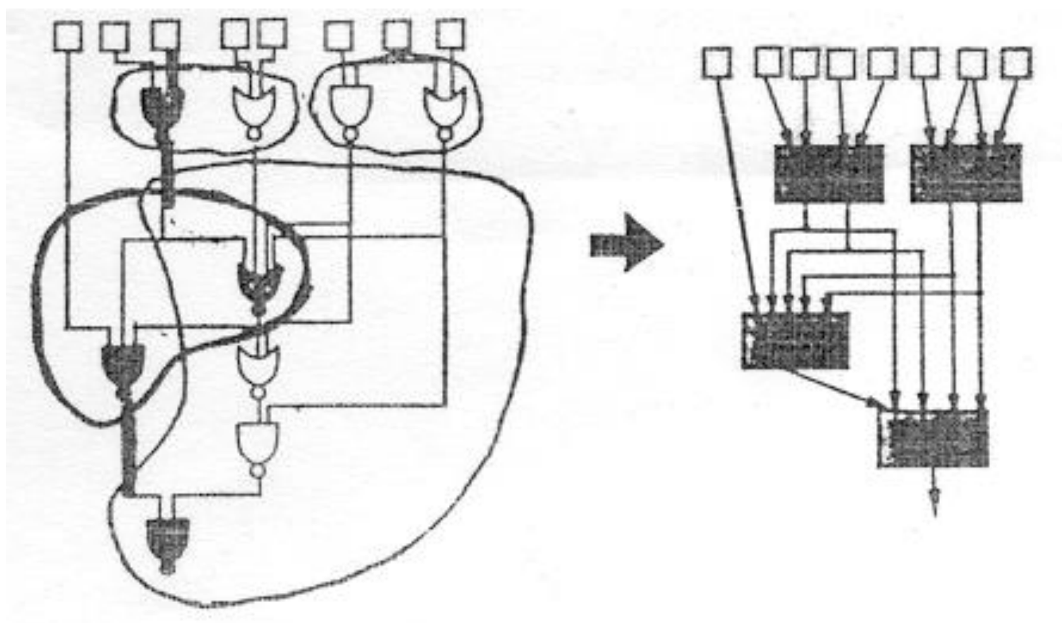
Circuit delay = 3.



General Delay Model

- Each gate has a delay.
- No interconnection delay within a cluster.
- Delay of D units for an interconnection between 2 clusters.
- A heuristic algorithm for area constraint only (Murgai, Brayton and Sangivanni-Vincentelli, ICCAD, 1991).

$$D = 2,\ \delta(v) = 1,\ \text{circuit delay} = 6+4 = 10.$$



General Delay Model (Cont'd)

- Rajaraman and Wong, "Optimal clustering for delay minimization," DAC, 1993.
- Optimal algorithm: $O(n^2 \log n + nm)$, where n is # of gates, m is # of interconnections.
- Definitions:

  - M: the area constraint on a cluster.

  - $W(C)$: the total area of the gates in cluster C.

  - N: a given combinational circuit.

  - N_v: v and all its *predecessors* in N.

  - $\delta(v)$: the delay of v.

  - $\Delta(u,v)$: maximum delay along any path from the output of u to the output of v, ignoring delays on interconnections.

  - $w(v)$: the area of v.

  - $l(v)$: the delay at v in an optimal clustering of N_v.

    For each *primary input* v, $l(v)=\delta(v)$.

  - $l'(u)=l(u)+\Delta(u,v)$, for each u in $N_v-\{v\}$.

General Delay Model (Cont'd)

- Algorithm: labeling phase + clustering phase.
- **Labeling phase:** compute $l(v)$ for each v in a topological order.

  - P: the set of nodes in N_v-$\{v\}$ sorted in non-increasing order in the value of l'.

```
Algorithm Labeling(v);
begin
done ← false;
cluster(v) ← {v};
while (not done)
        Remove the first node u in P;
        if (W(cluster(v)) + w(u)) ≤ M)
                cluster(v) ← cluster(v) ∪ {u};
                if P is empty
                        done ← true;
                endif
        else
                done ← true;
        endif
endwhile
l1(v) ← max{l'(x) | x ∈ cluster(v) ∩ PI};
l2(v) ← ~~l'(u) + D;~~ max{l'(u)+D | u ∈ Nv - cluster(v)}
l(v) ← max{l1(v) , l2(v)};
end
```

General Delay Model (Cont'd)

- **Clustering phase:** generate the clusters based on the information obtained in the labeling phase.
- Overall algorithm:

begin
Compute the maximum delay matrix Δ. $\Delta(i,j)$ is the maximum delay along any path from the output of i to the output of j;
for each PI i, **do** $l(i) \leftarrow \delta(i)$;
Sort the non-PI nodes of N in topological order to obtain list T;
while T is non-empty
  Remove the first node v from T;
  Compute N_v;
  for each node $u \in N_v \backslash \{v\}$ **do**
    $l'(u) \leftarrow l(u) + \Delta(u,v)$;
  Sort the nodes in $N_v \backslash \{v\}$ in order of decreasing value of l' to form list P;
  Call $Labeling(v)$;
endwhile
$L \leftarrow \mathcal{PO}$;
$S \leftarrow \phi$;
while L is not empty
  Remove a node v from L; $N - cluster(v)$
  $S \leftarrow S \cup \{cluster(v)\}$;
  for all nodes x in N, such that x is adjacent to y, for some $y \in cluster(v)$, $L \leftarrow L \cup \{x\}$;
endwhile
end

General Delay Model (Cont'd)

- An example: M=3, D=3