

Near to Optimal Size Selection in Combinatorial Circuits

Nodari Vakhania* Frank Werner**

*Science Faculty, UAEM, Mexico

** Faculty of Mathematics, Otto-von-Guericke Universität Magdeburg, Germany

INCOM 2012, Bucharest/Romania, May 23-25, 2012

Outline of the Talk

- 1 Introduction
- 2 Problem
- 3 Basic Concepts and the Algorithm
- 4 Concluding Results

1. Introduction

– design of VLSI circuits is a *multi-stage process*

– **3 phases:**

- **partitioning:** chip is split into smaller pieces
- **placement:** locations of all circuit gates (or blocks) are fixed
- **routing:** a realization of the connections settled in the placement phase is determined (i.e., find the layouts for the wires connecting the terminals on the gates)

– even the determination of an optimal layout (e.g. with minimum wire length) for a single net is *NP*-hard

⇒ *often:* heuristic algorithms

1. Introduction

2 routing phases:

Global routing: exact geometric details are ignored (i.e., only 'loose' routes for the wires are determined)

Detailed routing: determination of the exact geometric layout of the wires connecting the gate

⇒ each routing phase remains *NP*-hard

The **greedy algorithm** is the most widely used approach for gate sizing (it iteratively resizes the nodes on or near the critical path by means of particular heuristics)

1. Introduction

Problem

given: a combinatorial circuit with a number of gates, where for each gate a set of possible integer *size values* and a set of integer *delay values* are known

goal: determination of a *feasible size selection* for the gates such that in the corresponding oriented acyclic graph with the gates as nodes, the resulting critical path has the minimal length

⇒ derivation of an **approximation algorithm** with a maximum absolute error equal to the maximum difference between the largest and smallest overall delays taken over all gates

1. Introduction

Some references on VLSI design:

Geiger et al. (1984): early introduction into the VLSI design

Sherwani (1999): basic concepts and algorithms for VLSI design

Behjat et al. (2005): several ILP models for global routing

Szeszler (2005): combinatorial algorithms for detailed routing

Dickson (2007): overview on global routing dealing with algorithms, theory and computation

Deza et al. (2011): algorithms for global routing including an approximation algorithm based on IP models

1. Introduction

Some references on gate sizing:

Chuang et al. (1995): optimal gate size selection from a library to minimize the total circuit area subject to timing constraints

Coudert et al. (1996): comparative study of five algorithms for gate sizing on constraint free delay optimization and delay constraint power optimization

Beefink et al. (1998): selection of a good set of gate sizes to minimize the delay (or size) error of a prescribed measurement

Joshi and Boyd (2008): application of geometric programming to gate size selection in a circuit to minimize the total area subject to timing constraints

2. Problem

$G = (V, E)$ – acyclic graph

$V = \{0, 1, \dots, n+1\}$ – set of $n+2$ gates (0 and $n+1$ are fictitious gates)

gate i : one of the integer sizes $\alpha_{i1} \leq \alpha_{i2} \leq \dots \leq \alpha_{im}$ can be assigned

δ'_{ij} – delay of gate i imposed by α_{ij}
($\delta'_{i1} \geq \delta'_{i2} \geq \dots \geq \delta'_{im}$)

δ''_{ij} – additional delay which will imply gate i of size α_{ij} for its direct predecessor gates ($\delta''_{i1} \leq \delta''_{i2} \leq \dots \leq \delta''_{im}$)

overall delay time $\delta_{ij} = \delta'_{ij} + \delta''_{ij}$

if gate l has the size α_{lj} , then to any arc $(k, l) \in E$, the delay δ_{lj} of gate l is assigned

2. Problem

- **solution** defined by

$$\sigma : i \longrightarrow \alpha_{ij(i)}, \quad i \in V, 1 \leq j(i) \leq m$$

- solution $\sigma = \{\sigma_{ij(i)}\}$ is **feasible** if

$$\sum_{i=0}^{n+1} \sigma_{ij(i)} \leq B_{max},$$

where B_{max} is a given integer bound

- **optimal solution:** a feasible solution σ , which yields the minimal length of a critical path in G_σ

3. Main Concepts and the Algorithm

- **initial solution** σ_0 : assign to each gate its smallest size

Iterative procedure:

- determine the critical path P_0 in G_{σ_0}
- choose one gate on P_0 and assign a new (greater than the current) size to it
 - ⇒ this decreases the delay of this gate and hence the length of P_0
- proceed with the next iteration (i.e., determine again a critical path in the new graph and the vertex on it whose size is increased at that iteration)
 - ⇒ continue until the algorithm cannot increase further the size of the selected gate because of the overall limit B_{max}

3. Main Concepts and the Algorithm

substitution gate i : gate whose size selection is increased at the current iteration

overall size: sum of the sizes of all gates at a particular iteration h

regular size selection $\gamma(i, h)$: size selection of gate i at the beginning of iteration h

estimating size selection $\gamma^*(i, h)$ of gate i at iteration h :

$$\gamma^*(i, h) = \max \left\{ \frac{\delta_{i\gamma(i,h)} - \delta_{ik}}{\alpha_{ik} - \alpha_{i\gamma(i,h)}} \mid k > \gamma(i, h) \right\}$$

3. Main Concepts and the Algorithm

Global balancing procedure:

whenever a gate is substituted, we **rebalance** the **size assignments** of the gates involved in this substitution

- gate i is **revised** at iteration h' , $h' > h$, if $\gamma(i, h') = \gamma(i, h)$
(i.e., to gate i , its regular size selection corresponding to the beginning of the iteration of its last substitution is reassigned)
- gate i is **restored** at iteration h^* , $h^* > h'$, if $\gamma(i, h^*) = \gamma^*(i, h)$
(i.e., the estimating size selection of gate i at iteration h is reassigned to gate i at iteration h^*)

3. Main Concepts and the Algorithm

If gate i is **substituted** at iteration h , we may revise some gates substituted earlier and related to gate i . The revision of these gates may cause the restoration of some other related gates, and so on.

- gate i is **declared active** (for a specific set of gates):
it is possible that the size selection of this gate is revised later
- gate i is **not declared active**: its size selection will not be revised later

3. Main Concepts and the Algorithm

Observation

Restoration (revision, respectively) of any gate in the global balancing procedure increases (decreases) the overall size.

Strategy: Since we have an overall size bound B_{max} , we are interested in the maximum 'relative decrease' of the delay on one unit of increased size which is measured by the rate function $r_{ih}, i \in P_h$

\Rightarrow we do not look for the substitution which maximally decreases the length of P_h but we substitute the gate which will decrease the delay possibly less but the 'overall decrease' of the delay on one unit of increased size will be the best

3. Main Concepts and the Algorithm

Algorithm APPROX

Step 0

Construct the initial solution σ_0 ; $h := 1$.

Step 1

Determine the critical path P_h and $i \in P_h$ with

$$r_{ih} = \max\{r_{kh} \mid k \in P_h\};$$

Substitute gate i ;

Apply the global balancing procedure for gate i .

Step 2

If $B_h < B_{max}$, then $h := h + 1$ and go to Step 1
else stop $\{ \sigma_h \text{ is a } \Delta\text{-approximate solution} \}$.

4. Concluding Remarks

Major Results:

1. Let

σ_{opt} – **optimal solution**

σ – **approximate solution** by Algorithm APPROX

$\Delta = \max\{\delta_{ik} - \delta_{il} \mid i \in V, k, l \in \{1, \dots, m\}, k \neq l\}$

$\tau(\sigma) =$ length of a critical path in G_σ

$$\Rightarrow \tau(\sigma) \leq \tau(\sigma_{opt}) + \Delta$$

4. Concluding Remarks

2. Let

$$c = \left[\frac{\max\{\delta_{ik} - \delta_{il} \mid i \in V, k, l \in \{1, \dots, m\}, k \neq l\}}{\min\{\alpha_{ik} - \alpha_{il} \mid i \in V, k, l \in \{1, \dots, m\}, k \neq l\}} / \frac{\min\{\delta_{ik} - \delta_{il} \mid i \in V, k, l \in \{1, \dots, m\}, k \neq l\}}{\max\{\alpha_{ik} - \alpha_{il} \mid i \in V, k, l \in \{1, \dots, m\}, k \neq l\}} \right] + 1.$$

\Rightarrow Algorithm APPROX has a running time of $O(c^2 B_{\max} n^3 \log n)$