

Heuristiken für das TechnologyMapping

Alexander Zorn

Geboren am 26. Mai 1996 in Bonn

2. April 2018

Bachelorarbeit Mathematik

Betreuer: Prof. Dr. Stephan Held

Zweitgutachter: YYYY YYYY

FORSCHUNGSINSTITUT FÜR DISKRETE MATHEMATIK

MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT DER
RHEINISCHEN FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

Inhaltsverzeichnis

1	Einleitung	2
2	Terminologie & grundlegender Algorithmus	2

1 Einleitung

Das Chipdesign ist ein Forschungsgebiet, welches in den letzten Jahrzehnten eine immer bedeutendere Rolle eingenommen hat. Es ist ein zu einem Projekt imenser Wichtigkeit und Beteiligung verschiedenster wissenschaftlicher Zweige (Mathematik, Physik, Informatik, Chemie etc.) geworden. Professor Korte/Vygen sagte einmal HIER ZITAT EINFUEGEN.

Die schwierige Aufgabe hierbei besteht darin einen booleschen Schaltplan von atemberaubender Größe auf einem wenige Quadratzentimeter großen Chip unterzubringen.

Ein Schaltplan (später als Circuit definiert) beschreibt hierbei eine Implementierung einer Booleschen Funktion mithilfe kleiner Bauteile (später Gates). Eine solche lässt sich mit mehreren unterschiedlichen Bauplänen (Kandidaten) von Gates realisieren wobei jede Realisation Eigenschaften an Größe und Schnelligkeit (Delay) besitzt.

Die Aufgabe des TehnologyMapping ist es nun den Bauplan zu finden, welcher eine Kostenfunktion (bestehen aus Größe und Delay) optimiert.

In der Vorliegenden Arbeit wird ein PTAS (polynomial time approximation algorithm) für kleine Circuits vorgestellt und aus diesem eine Heuristik für die Anwendung auf dem gesamten Netz des Chips entwickelt.

WAS NOCH FEHLT : KLEINES BEISPIEL DER IMPLEMENTIERUNG GEBEN TERMINOLOGIE VERBESSERN AUF BENUTZTE ARBEITEN VERWEISEN

2 Terminologie & grundlegender Algorithmus

Definition 2.1. Boolesche Variable und Funktion:

Eine boolsche Variable ist eine Variable mit Werten in $\{0, 1\}$.

Sei $n, m \in \mathbb{N}$. Eine boolsche Funktion ist eine Funktion $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ mit n inputs und m output.

AB HIER LUCAS VORLAGE

Definition 2.2. Library: Eine Library ist eine Menge L von Gates (boolesche Funktionen) mit zwei Abbildungen $d, \text{area} : L \rightarrow \mathcal{R}_{\geq 0}$, die jedem Gate sowohl eine Verzögerung d_l , als auch eine Fläche area_l zuordnen.

Definition 2.3. Circuit: Ein Circuit C auf der Library L ist ein zusammenhängender gerichteter azyklischer Graph (DAG), bei dem jeder Knoten einer dieser 3 Arten entspricht:

- einem Inputknoten ohne eingehende Kanten
- einem Gate aus L mit ≥ 1 eingehenden und ≥ 1 ausgehenden Kanten

- einem Outputknoten ohne ausgehende Kanten

Jeder Gateknoten kann an jeder seiner eingehenden Kanten einen Inverter vorschalten. Für einen Knoten v sei $\text{fanin}(v)$ die Zahl seiner eingehenden, $\text{fanout}(v)$ die Zahl seiner ausgehenden Kanten. Knoten mit $\text{fanout}(v) > 1$ heißen Highfanoutknoten. Wir betrachten vorerst nur Circuits mit exakt einem Outputknoten.

Definition 2.4. *cone*: Für einen Knoten g aus einem DAG S bezeichne

$$\text{cone}(g) := S[V \cup \{g\}], V = \{v \in V(S) : \exists v\text{-}g\text{-Weg in } S\}$$

Sowie für eine Knotenmenge G sei $\text{cone}(G) := S[\cup_{g \in G} V(\text{cone}(g))]$

Algorithmus : TechnologyMapping auf einer Arboreszens

Input : Circuit C kreisfrei mit finalem Output o , Library L
verfügbarer Gates

```

1 bester_kandidat[]  $\leftarrow \emptyset$ 
2 bester_inv_kandidat[]  $\leftarrow \emptyset$ 
3 foreach Knoten  $n \in V(G)$  in topologischer Ordnung do
4   berechne alle (invertierte) Matches auf  $n$ 
5   foreach Match  $m$  auf  $n$  do
6     Berechne besten Kandidaten mit  $m$  auf  $n$ 
7     Update best_(inv)_kandidaten
8  $best\_final \leftarrow \text{best\_kandidat}[o]$ 
9 Implementiere  $C$  entsprechend  $best\_final$ 
```
