Paul Zuber

TopCool

Dissertation

Publication Date
August 2007

Contact
pauleontologe@mytum.de

Lehrstuhl für Integrierte Systeme
Technische Universität München

Wire topology optimisation for low power CMOS

Paul Zuber

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzende: Univ.-Prof. Dr. rer. nat. Doris Schmitt-Landsiedel

Prüfer der Dissertation:

1. Priv.-Doz. Dr.-Ing. Walter Stechele
2. Univ.-Prof. Dr.-Ing. Ulf Schlichtmann

Die Dissertation wurde am 16.04.2007 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 20.07.2007 angenommen.

Preface

I wish to express my utmost and sincere thank to PD Dr.-Ing. Walter Stechele. Never, from the acceptance of the topic at the Institute for Integrated Circuits (later Systems) in 2002, to the perfect timing to say, “That is it!,” never I had the feeling of doing the wrong thing. Prof. Dr.-Ing. Frank Johannes is gratefully acknowledged for inspiration and fruitful discussions and Prof. Dr.-Ing. Ulf Schlichtmann for taking the role of the second examiner. I also appreciate the efforts and time of Prof. Dr. rer. nat. Doris Schmitt-Landsiedel for chairing.

Prof. sc.techn. Andreas Herkersdorf and Prof. Dr.-Ing. Ingolf Ruge, the head and the founder of the institute, respectively, deserve many thanks. The working conditions were what I consider ideal—generous, international, well-equipped and fairly balanced between research, teaching and other tasks. I will always remember the conference and teaching stays in remote locations.

Words cannot express my sincere thankfulness to my family and my friends, and especially to Mariana Burghiu, who accepted my “social down times” that were necessary once in a while. The love and care given by her and by my parents Ulrike and Alois and my grandmother Elisabeth is probably the sweetest experience that man can conceive.

A special thanks goes to Armin Windschiegl. Without him, I would have never been introduced to the subject. I feel indebted to Prof. Dr. Peter Gritzmann and Michael Ritter of the Institute for Combinatorial Geometry, TU München, who reacted to my “distress call” regarding “Theorem Eins” spontaneously and provided unsolicited help in mathematical proofreading.

It matters a great deal to me to thank my fellow friends, the “integrated circuits people,” Winthir Brunnbauer, Michael Eiermann, Wolfgang Embacher, Jürgen Foag, Sven Haar, Stephan Herrmann, Kilian Jacon, Thorsten Mahnke, Hubert Mooshofer, Norma Constanza Müller, Ulrich Niedermeier, حسین پاکار (Hossein Pakar), Jürgen Regner, Fabian Vogelbruch, Thomas Wild, Thomas Winkovich and Roland Zukunft, the “Spanish connection,” Lucía Alvado Carcel, Julio Lidón Simón, José Manuel Martínez Ibáñez, Raúl Medina Beltrán de Otálora y Nuria Pazos Escudero, the “integrated systems people,” عبد المجيد بوعجيلة (Abdelmajid

Bouajila), Christopher Claus, Mathias Ihmig, Κίμων Καρράς (Kimon Karras), हर्षद कस्तुरे (Harshad Kasture), Andreas Laika, Prof. 이규영 (Gyoo-Yeong Lee), Daniel Llorente, Michael Meitinger, Rainer Ohlendorf, Holm Rauchfuß, 王忠磊 (Zhonglei Wang), 王文杰 (Wenjie Wong), Johannes Zeppenfeld and 袁征 (Yuan Zheng), and the institute staff, Verena Draga, Wolfgang Kohtz, Martin Kottermeier, Gabi Spörle and Doris Zeller for pleasing collaboration, proof-reading, and countless hours of joy.

عثمّان بحلوس (Othman Bahlous), Robert Hartl, Florian Helmut Müller and Thomas Ilnseher who were directly involved in the *TopCool* project, are specially acknowledged. Thanks also to अमित चौधरी (Amit Chaudhari) in India for surrendering his copy of [Eck01].

Last but not least, my industry contacts, especially Dr. Michel Berkelaar (Magma Design Automation Eindhoven BV) and Dr. Gerard Allan (Predictions Ltd) are acknowledged for their help in obtaining university licenses of their respective products.

Without political or cultural connotation, this work is written in British English.

Brunnthal, April 2007 Paul Zuber

Für

Hanni und Simon

Abstract

Power optimisation has become one of the most important goals when designing integrated systems. This dissertation develops a method that optimises the capacitive wire power consumption of digital integrated CMOS circuits.

Wire capacitances account for the largest share of capacitances in today's and future integrated systems. The energy required to toggle the logic value of a signal is directly proportional to its net capacitance. As coupling capacitances between adjacent wires on the same metal layer exceed inter layer coupling, the work focuses on distancing neighbouring wires.

A methodology is proposed that takes a detail-routed layout from a commercial router, the activity factors of the nets, and a set of technology-specific rules as input. The power reduction process can be split into three parts. First, all groups of parallel wire segments that can be shifted laterally are extracted from the layout. Second, for every such group, a mathematical programming problem is formulated and solved. Its solution re-distributes the local whitespace between the wires inside every group, taking the switching activities of the wires into account. The more active a wire the more space it will acquire and thus the less toggle energy is required. Finally, a heuristic selects a subset of non-intersecting groups with a high total power saving. After optimisation, the new layout is returned to the commercial design flow for further processing such as parasitic extraction and power calculation.

As opposed to prior methods, switching activities are included, and the circuit area is preserved during optimisation. In addition, timing is not adversely affected. Emphasis is placed on seamless fit into commercial design flows and speed. Experimental results have shown that the proposed methods save more than 8% wire switching power of an eight-core processor system containing more than one million wires.

Kurzfassung

Die Optimierung des Leistungsverbrauchs hat sich zu einem der wichtigsten Entwurfsziele für integrierte Systeme entwickelt. Die vorliegende Dissertation erarbeitet eine Methode, welche die kapazitive Verlustleistung digitaler integrierter CMOS-Schaltungen verringert.

Der höchste Anteil der in heutigen Systemen auftretenden Kapazitäten entfällt auf die Verdrahtungslastkapazitäten, vor allem auf die Kopplung zwischen parallelen Leitungen. Die benötigte Energie um den Logikzustand auf einer Leitung zu ändern, ist direkt proportional zur Leitungskapazität. Das Ziel der Arbeit ist das optimale Einstellen der Distanzen zwischen den Leitungen in einer integrierten Schaltung.

Es wird ein Verfahren vorgeschlagen, das die Leistung einer fertig verdrahteten Schaltung aus einem kommerziellen Verdrahter optimiert. Dazu liest es neben der Schaltung auch die Schaltaktivitäten ein. Die Leistungsoptimierung kann in drei wesentliche Schritte eingeteilt werden. Zunächst werden alle Gruppen von parallelen Leitungen, die sich seitlich verschieben lassen, gesucht. Im zweiten Schritt wird für jede dieser Gruppen ein mathematisches Optimierungsproblem erstellt und gelöst. Das Ergebnis ist eine Neuverteilung der nicht verwendeten Verdrahtungsresourcen, wobei die Schaltaktivitäten berücksichtigt werden. Je aktiver ein Draht ist, desto mehr Abstand wird diesem zugewiesen. Damit verringert sich dessen Schaltenergie. Am Ende entscheidet eine Heuristik, welche der Gruppen in der endgültigen Schaltung berücksichtigt werden. Nach der Optimierung wird die neue Schaltung in den kommerziellen Entwurfsablauf zurück exportiert. Dort können unter anderem die Kapazitäten und die Leistung neu berechnet werden.

Im Vergleich zu früheren Methoden werden die Schaltaktivitäten berücksichtigt. Die Schaltungsfläche bleibt erhalten. Darüber hinaus wird das Zeitverhalten nicht negativ beeinflusst. Schwerpunkte wurden auf die Kompatibilität mit Standardentwurfsabläufen und Geschwindigkeit gesetzt.

Experimentelle Ergebnisse zeigen, dass mit dieser Methode deutlich mehr Leistung eingespart werden kann, als mit anderen. Außerdem ist die vorgeschlagene Methode um Größenordnungen schneller. Es wird erwartet, dass das Einsparpotential dieser Methode mit neuen Technologiegenerationen weiter zunimmt.

Contents

Abstract I

Kurzfassung II

1 Introduction 1
- 1.1 Power trends . . . 1
- 1.2 Wire crisis . . . 2
- 1.3 Goal of this dissertation . . . 3
- 1.4 The proposed solution TopCool . . . 4
- 1.5 Outline . . . 6

2 Basics 7
- 2.1 On-chip interconnect . . . 7
  - 2.1.1 Wire physics . . . 9
  - 2.1.2 Routing basics . . . 10
  - 2.1.3 Capacitance determination . . . 11
- 2.2 CMOS Power basics . . . 15
  - 2.2.1 Energy . . . 15
  - 2.2.2 Switching activity . . . 15
- 2.3 Conclusions . . . 17

3 Prior art 20
- 3.1 Entire circuits . . . 21
  - 3.1.1 Interconnect-power dissipation in a microprocessor . . . 21
  - 3.1.2 Exploiting metal layer characteristics for low power routing 23
  - 3.1.3 Routing methodology for minimising interconnect energy dissipation . . . 23
- 3.2 Specialised wire topology optimisations . . . 25
  - 3.2.1 Spacing parallel bus wires . . . 25
  - 3.2.2 Permuting parallel bus wires . . . 28
  - 3.2.3 Combined spacing and permuting parallel bus wires . . . 30
- 3.3 Related techniques and objectives . . . 31
  - 3.3.1 Wire spreading . . . 31

3.3.2 Placement . . . 31
3.3.3 Power driven routing using a genetic algorithm . . . 31
3.3.4 Equivalent pin assignment . . . 32
3.3.5 Bus coding . . . 32
3.3.6 X-Initiative . . . 32
3.4 Present approach in context of prior art . . . 33
3.5 Summary . . . 35

4 Parallel wire optimisation 37
4.1 Parallel wire spacing . . . 37
4.1.1 Fast wire spacing . . . 39
4.1.2 Non-uniform parallel wire spacing considering detours . . . 43
4.2 Parallel wire permutation for low power CMOS . . . 51
4.3 Conclusions . . . 53

5 Systematic optimisation of entire detail-routed layouts 55
5.1 Searching parallel wires . . . 55
5.1.1 Problem . . . 55
5.1.2 Approach . . . 56
5.1.3 Complexity . . . 59
5.2 Re-spacing parallel wires . . . 61
5.3 Selecting the best groups of parallel wires . . . 62
5.3.1 Connection graph . . . 63
5.3.2 Selection heuristic . . . 63
5.3.3 On rectangle intersection . . . 68
5.4 Conclusions . . . 68

6 Experimental results 69
6.1 Setup . . . 69
6.1.1 Benchmark circuits . . . 69
6.1.2 Synthesis flow . . . 71
6.1.3 Constraints . . . 72
6.1.4 Library and operating conditions . . . 74
6.1.5 Activities . . . 75
6.1.6 Server . . . 75
6.1.7 Implementation . . . 75
6.1.8 Processing layouts . . . 76
6.2 Optimisation results . . . 77
6.2.1 Notes on other approaches . . . 77
6.2.2 Power saving results . . . 79
6.2.3 Runtime . . . 82

6.2.4 Manufacturing grid . . . 84
6.2.5 Timing . . . 85
6.2.6 Yield . . . 85
6.3 Effect of dropping correlations . . . 88
6.4 Effect of wire ordering . . . 91

7 Limitations and future work **93**
7.1 Multidimensional objectives . . . 93
7.1.1 Timing . . . 93
7.1.2 Yield and manufacturability . . . 95
7.1.3 Signal integrity . . . 96
7.2 Improving power savings . . . 97
7.2.1 Simultaneous switching power . . . 97
7.2.2 Capacitance model . . . 98
7.2.3 Short-circuit power . . . 99
7.2.4 Crosstalk noise power . . . 100
7.2.5 Cross-box optimisation . . . 101
7.2.6 Rectangle splitting . . . 102
7.2.7 Wire permutation . . . 103
7.2.8 Vias . . . 104
7.2.9 Post routing gate sizing . . . 107
7.3 Impact of technology scaling . . . 108
7.3.1 Wire switching power fraction . . . 108
7.3.2 Wire aspect ratio . . . 108
7.3.3 Dielectric . . . 109
7.3.4 Process variations . . . 109
7.3.5 X-Architecture . . . 109

8 Conclusions **110**

A Proof of Theorem 1 **112**
A.1 Known results from convex programming . . . 112
A.2 Characterisation of the optimal distances . . . 113
A.3 Power-optimal wire ordering . . . 114

References **118**

List of Figures

1.1 Energy labels. . . . 2
1.2 TopCool external flow. . . . 5

2.1 CMOS circuit layers. . . . 8
2.2 Wire coupling past and today. . . . 9
2.3 State of the art routing screenshot. . . . 11
2.4 Capacitance extraction experiment. . . . 13
2.5 Capacitance fitting. . . . 14
2.6 Energy drawn by coupling nodes. . . . 19

3.1 Activity driven wire spacing. . . . 25

4.1 Bus wire spacing including correlations. . . . 41
4.2 In-place wire spacing considering wire detours. . . . 44
4.3 Discrete solution space. . . . 47
4.4 River routing problem. . . . 49
4.5 Wire ordering. . . . 52
4.6 The optimal wire order for low power CMOS. . . . 53

5.1 Finding parallel wires in a layout. . . . 57
5.2 Multiple corridors. . . . 60
5.3 Stabbing queries. . . . 60
5.4 Illustrative run of high weight independent set heuristic 67

6.1 Comparison of the CPU runtimes. . . . 83
6.2 CPU runtime relative to synthesis. . . . 83
6.3 Influence of the manufacturing grid on power saving. . . . 84
6.4 Defect data. . . . 89
6.5 Prediction error. . . . 90
6.6 Prediction error. . . . 91

7.1 Capacitance between two wires for lonely and populated cases. . . . 99
7.2 The wire topology between two selected boxes can be optimised. . . . 101
7.3 Rectangle splitting. . . . 102
7.4 Illustration of field forces. . . . 105
7.5 Illustration to clarify the spring force. . . . 106
7.6 Example for the spring force. . . . 106

List of Tables

2.1 Typical wire properties. 8

3.1 Comparison between techniques for entire circuits. 34
3.2 Power-driven parallel wire spacing techniques. 34
3.3 Power-driven wire permutation techniques. 35

6.1 Benchmark suites. 71
6.2 Benchmark circuits 72
6.3 Constraints used to synthesise the benchmarks. 73
6.4 Wire dimensions 74
6.5 TopCool parameters used to optimise the benchmark circuits. . . . 76
6.6 Interconnect power reduction methods. 77
6.7 Interconnect power of layouts with relaxed constraints. 79
6.8 Interconnect power of layouts with medium constraints. 80
6.9 Interconnect power of layouts with hard constraints. 81
6.10 Critical path delays. 86
6.11 Change in critical path delays, best case. 87
6.12 Yield-limits. 87
6.13 Reduction potential of wire ordering on M tracks (%). 92

List of Algorithms

1 Power-driven wire spacing considering detours. 46
2 River routing for wire endings. 50
3 The optimal wire order for low power CMOS. 54

4 Searching for groups of parallel wires. 58
5 High weight independent set heuristic. 66

1 Introduction

Low power dissipation is undoubtedly one of the most important goals when designing integrated systems. The power consumption of a chip largely determines the material type of its housing and the cooling measures required to maintain a tolerable substrate temperature. Erratic system behaviour can emerge from increased transistor temperature, and sustained high current densities can even cause irreversible damages to on-chip power supply wires. A maximised time between two battery charges is not only a decisive selling point of mobile wireless devices in the increasingly popular multimedia segment, but also an indispensable requirement for medical or space applications. All of these factors directly affect the system cost, apart from the price for electricity alone.

In addition to these short-term costs, there is also the aspect of environmental protection. Less power consumption of integrated systems helps to reduce the emissions of greenhouse gases and thus to comply with international agreements such as the Kyoto protocol. The second *European Climate Change Programme* progress report [ECC03] amounts the fuel-based CO_2 emissions caused by electronics in the residential sector to 16 million tons in 1990.[1] The estimated values for 2010 before and after applying all programme measures increase to 64 and 34 million tons, respectively. Without taking action, the possible economical and the ecological damage is known to exceed the expected programme costs by far. Several political programmes already support low power devices and label complying products, cf. Figure 1.1.

1.1 Power trends

Unfortunately, power reduction competes fiercely with performance. The demand for higher frequencies and integration densities also takes its toll on power consumption. Assuming constant circuit area, few power reduction recipes exist that are neutral on or even increase the performance. As a result, the power

[1] One million ton of CO_2 corresponds to about two billion kilowatt-hours electricity.

Figure 1.1: Energy labels. Left: Energy star used for low power office equipment, right: Ecolabel used for low power electronics and environmentally sound goods.

consumption of an average integrated system has continuously increased during the last decades and will most probably continue to do so.

Complementary metal oxide semiconductor (CMOS) has established as the logic family of choice for super large scale integration (SLSI) digital circuits. CMOS had been introduced for the mass market in the 1980ies because of its low static power consumption, i.e. its low permanent loss of energy through leakage currents. Therefore, power reduction techniques for CMOS have traditionally addressed the dynamic power consumption that only occurs during signal transitions. Although the significance of static power has increased with the decreasing transistor sizes during the last years, dynamic power still constitutes the major fraction, particularly in low leakage technologies. As of the time of this thesis, Hafnium-based transistors [PCA07] were introduced independently by IBM and Intel, reducing the static leakage of next-generation chips by a factor of five.

The present thesis focuses digital circuits fabricated in CMOS technology, and addresses the capacitive power consumption caused by wires, the most significant part of the dynamic power consumption.

1.2 Wire crisis

The capacitive power consumption caused by wires has not always played a prominent role, in past days it was the transistor capacitance and not the wire capacitance that outweighed. The quotation below lively describes the consequences

of increasing wire capacitances with increasing integration densities, also known as the wire crisis:

> "In today's billion-plus transistor chips, which have multiple layers of wires connecting transistors and many kilometres of interconnects per square centimetre, the wires cost more than the transistors." [Per06]

Capacitances emerge if conducting objects are placed in proximity. Routing wires in an integrated system determines the position and shape (topology) of the transistor interconnects and thus their capacitances. When more and smaller transistors are required, the wire capacitances do not decrease as fast, stay constant or even increase. Three major reasons for the wire crisis are the largely inverse proportional relationship between distance and capacitance, the increased resistivity of copper at very small wire dimensions and the increasing average length and number of wires on a chip.

Wire capacitances are charged and discharged as signals change their voltage. This requires energy, and with continuous switching, capacitive power consumption occurs. Recently, Intel published a power break-down of a mobile mass-market CPU [MKWS04]. Its wire power consumption is amounted to 51 % of the total dynamic power loss with an estimated increase to 65 % – 80 % within five years.

1.3 Goal of this dissertation

So far, few works have investigated the link between low power design and the routing of the signal wires. Related works are discussed in Section 3. The goal of this dissertation can be defined to fill this gap:

> *Find an integrated circuit design methodology to optimise the wire topology in order to achieve circuits with less power consumption.*

To achieve this goal, this design methodology must adhere to the following basic conditions:

1. The number of available transistors that can be manufactured on one chip grows faster than the engineers' ability to meaningfully connect them to a higher level system. This is also known as the designer productivity gap. Thus, the optimisation must be automated as far as possible and should not require tuning or interception. Additionally, algorithm complexities must

be low. With huge input data cardinalities, quasilinear complexity in the problem size can be considered as the maximum tolerable limit.

2. The reduction in power consumption must be obtained without major drawbacks on other design goals. Methods that inherently demand for compromises in timing or area are not acceptable.

3. To assess the usefulness of the technique, an implementation is required that seamlessly taps into an existing design flow. This enables to optimise commercial circuits and to measure the power reduction with industry-proven tools, two requirements that assure significance and credibility of the work.

4. A design methodology should be universally applicable. It should optimise any circuit regardless of the function implemented by the circuit.

1.4 The proposed solution TopCool

This thesis introduces *TopCool*[2], a methodology and its implementation to reduce the wire power of detail-routed circuits. Motivated by the general trends observed above, it follows the following principles:

1. Power optimisation is performed post-routing. Existing sophisticated routers are used to obtain an initial routing result which complies with design rules and timing constraints. The method then improves upon the routing results to obtain a modified version with reduced power consumption.

2. The prior art has shown that considering switching activities during low power optimisations delivers much better reduction results rather than assuming constant activities. The present work includes switching activities—for the first time during wire topology optimisation for low power.

3. Capacitances are balanced against each other by individually adjusting the distances between adjacent wires—an effective measure as the major fraction of a wire capacitance can be attributed to the lateral coupling component. At the same time, the major drawback of current routers, the dependence on a routing grid that prevents distances to accept arbitrary values, is circumvented. Wire lengths are not optimised in this work.

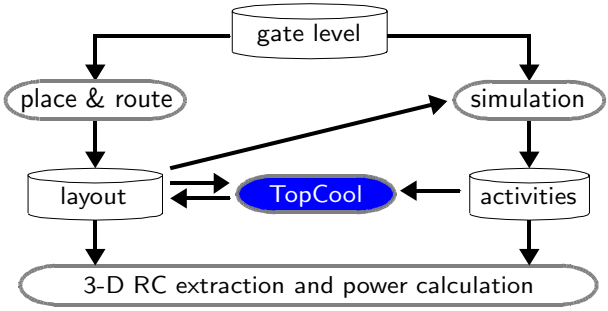


Figure 1.2: Illustration of how the proposed method taps into a commercial design flow. Existing detail-routed layouts are modified based on switching activity information for the nets.

A solution is proposed that complies with the requirements and principles described above. Figure 1.2 shows the intended external flow of the method. Apart from reading and writing the design layout and reading the switching activity data, the internal flow can be split into three major steps:

Search A detail-routed circuit is searched layer by layer for all areas of parallel wires.

Space Considering the switching activities, the wires inside every such area are laterally displaced to minimise the total switched capacitance caused by the wires in the area. This is the actual power reduction principle.

Select Not all such areas can be optimised at the same time since they generally intersect. A selecting step solves this non-trivial problem, and delivers the best subset for actual modification.

By using this approach, the actual power optimisation problem is reduced to one physical dimension, the spacing step. This simplifies the problem and thus ensures fast completion rates.

After optimisation, the new layout is returned to the normal design flow for further processing such as verifying the success of the operation with capacitance and power measurement tools. Figure 1.2 illustrates this step.

[2]Wire *top*ology optimisation in order to obtain *cool*er chips.

1.5 Outline

Including this introduction, the entire thesis is split into eight chapters:

Chapter 2 provides basic knowledge in the fields covered by the thesis. This includes wire physics and CMOS power consumption mechanisms. It can be safely skipped without loss of continuity by readers familiar with these topics.

In Chapter 3, an overview of related work in power reduction through routing modification is given. It is sectioned into a part that covers specialised techniques that apply to parallel wires and a part on approaches that address the wire power consumption of entire circuits.

A new approach to activity-driven parallel wire spacing is presented in Chapter 4. It differs from previously published wire spacing techniques in speed and auxiliary constraints, which allows it to be applied in-place in already detail-routed layers. The observation of the optimal spatial wire order is described in the second section of Chapter 4. This order maximises the effect of spacing.

Based on these results, Chapter 5 develops a new methodology to decrease the wire power consumption of entire routed designs. It is split into sections according to the internal flow of the proposed method: searching regions of parallel wires, applying the spacing technique of the previous chapter in these regions, and finally selecting the best subset of regions to be actually committed in the final layout.

The methodology developed in the previously described chapters was implemented as tool, and a series of experiments was conducted. The results are listed and discussed in Chapter 6.

There are several opportunities to further improve the methodology developed. A disquisition on the current limitations and suggestions on possible fields of future research can be found in Chapter 7.

Chapter 8 provides a summary of the main statements and their conclusions.

Appendix A proofs the optimality of the wire order observed in Chapter 4.

2 Basics

Today, static CMOS logic is the first choice to implement digital circuits. Production is mature and thus cheap in comparison to other techniques. Furthermore, its design principle is well understood and circuit models are simple to use. Static CMOS logic was introduced because it did not dissipate significant leakage power. This thesis deals with the dynamic power consumption, more specifically, the capacitive power.

In digital circuits, there only exist two nominal states for any signal, zero and one. CMOS uses a high voltage and a low voltage level to signal a logic one and a logic zero, respectively. The high voltage is usually referred to as V_{DD}, a technology-specific value, today in the order of magnitude of 1 V, and the low level as V_{SS}, "ground" or "gnd", i.e. 0 V.

Unfortunately, wires and logic gates exhibit parasitic capacitances. Capacitances store charge carriers and thus energy. Whenever a logic value changes its state, charge carriers must be added or removed from these capacitances. Continued signal changes are responsible for the dynamic power consumption.

This chapter provides the necessary background required in the sequel of the thesis. It is outlined in two sections. The first section describes how wires are modelled and laid out in today's chips. In addition, it derives the capacitances between these wires, and draws implications on the underlying work. Section 2.2 presents the theory on the power consumption in general and links it to the power consumption caused by wire capacitances. For deeper knowledge, however, standard text books such as [WE93, Yea98, Rab03, HSWZ05] are recommended.

2.1 On-chip interconnect

Manufactured CMOS circuits exhibit a planar assembly in several layers, cf. Figure 2.1. Semiconductor regions (silicon) that implement the transistors reside at the bottom. Above, layers follow for the wires that interconnect the transistors and primary input and output pins of the circuit Single wires run within one layer. Via layers provide electrical connection between two wires of adjacent layers. Today, several such layers are required to route all connections.

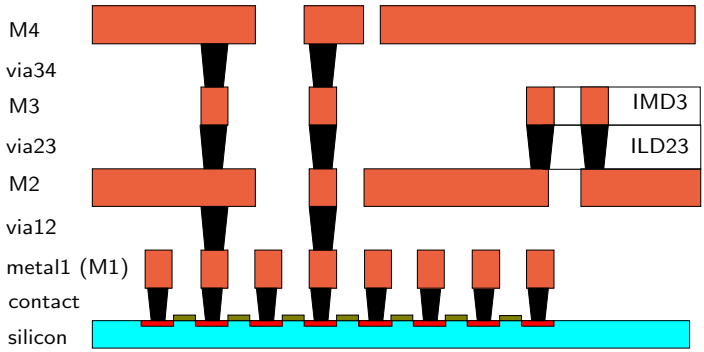
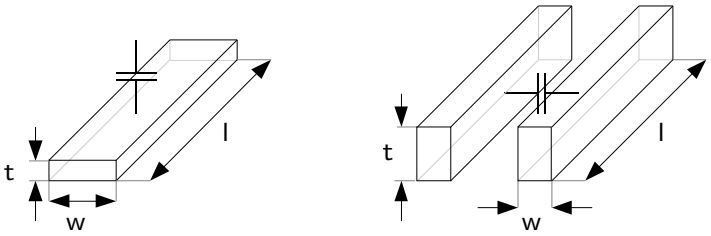


Figure 2.1: Schematic CMOS circuit layers (not to scale). Metal layers are commonly denominated by "metal*x*" or "M*x*". The numbering starts with one for the bottom-most metal layer, in between are the via layers. "IMD" and "ILD" refer to the inter metal dielectric and the inter layer dielectric, respectively.

Table 2.1 lists some important interconnect properties. Chapter 7.3 provides more details on the impact of "technology scaling", a term used for the continuous decrease of the minimum feature size to cope with the demand for ever higher integration densities.

Table 2.1: Typical wire properties are listed to illustrate the increasing importance of wire capacitances. The wire length is given under the assumption of 30% utilisation of routing resources [HSWZ05].

year	2004	2009	2018
minimum feature size	90 nm	60 nm	18 nm
number of metal layers	10	12	14
on-chip wire length	700 m	1.6 km	5 km



(a) Past. (b) Today.

Figure 2.2: In the past, wires mainly coupled to other layers. Today, wires mainly couple to neighbours on the same layer.

2.1.1 Wire physics

Single wires are commonly modelled as cuboids.[1] It is important to know that the technology specifies a nominal thickness for every layer. Integrated circuit (IC) designers can not influence this constant. Furthermore, there are limits on the minimum possible wire width and the minimum possible distance between any two wires.

Wires and logic gates capacitively couple to other wires, diffusion areas, and the substrate, to any conducting object in general. These capacitances store charge carriers and thus energy proportional to their capacitance value. The capacitance value is influenced by the physical wire geometry.

A capacitance of a metallic object decreases with increasing distance to other metallic objects and it increases with surface area. As the technological progress demands ever higher integration densities, the space between wires decreases. Figures 2.2(a) and 2.2(b) sketch this technological change. Fully compensating for the increasing capacitance with a smaller wire surface area is impossible as the wire resistance – the resistivity of copper grows at very small dimensions – sets a lower limit on the wire cross section $t \cdot w$.

As a result, the wire aspect ratio t/w is kept high, i.e. the thickness t does not scale as fast as the width w. Consequently, the wire capacitances are dominated by the lateral capacitances between wires on the same layer. This dominance is additionally fuelled by the parallel routing direction within a layer and the increasing average wirelength l.

[1] Physically, they exhibit a trapezium-shaped cross section, a deviation from the nominal intended shape caused by limitations in the manufacturing process.

2.1.2 Routing basics

Routing finalises the design of an integrated system. The task is to find a physical representation of the nets, the electrical connections between the components of the circuit. The component positions and their pin positions are known. With today's complexities of many millions of nets, this process must be automated.

This thesis proposes a method that utilises existing routing algorithms and improves the routing results by locally re-distributing the space between the wires. No matter which router is used, one property of routing output helps significantly in this respect: wires on the same metal layer are preferably routed in the same routing direction. Perpendicular directions on the same metal layer may occur but are exceptions.

Today, the most often deployed grid-based routers place wires next to each other at a uniform distance. This grid is called "routing pitch" and is normally the sum of the minimum possible wire spacing and width. Grid routers ensure completion of complex designs in acceptable runtimes, but naturally do not support individual space allocation. In contrast, gridless routers provide a much higher solution space and thus quality of the routing solution. But these types of routers are still academic or used for special purposes at most, due to runtime problems.

A compromise was introduced by subgrid routers [LN03]. These allow for wire positions that are integer multiples of half or even quarter of the routing pitch, about 50 nm for a 130 nm technology. However, a much finer, nearly continuous resolution is possible.[2] Figure 2.3 shows the routing output of a subgrid-based router.

Wire length minimisation has always been the main objective during routing. On the one hand, this avoids congestions and on the other hand, it minimises the wire resistance and capacitance and thus delay and power consumption. However, for capacitance-influenced objectives, wire length reduction alone is not adequate any more, because of the increasing sidewall couplings. A sound router must also use the existing space to distance the wires appropriately.

In addition to housekeeping the routing area and the high number of objects to deal with, routers have to consider numerous other constraints: circuit timing, complex manufacturing design rules, manufacturing yield, and signal integrity are only few of those as listed for instance in [Kah03, Leu03]. Whether all constraints are coped with, can only be ascertained after the layout has been completed. In anticipation to Chapter 3, no power driven routers are known.

[2] Values in the order of 10 nm are common for 180 nm technologies and 5 nm for 130 nm technologies.

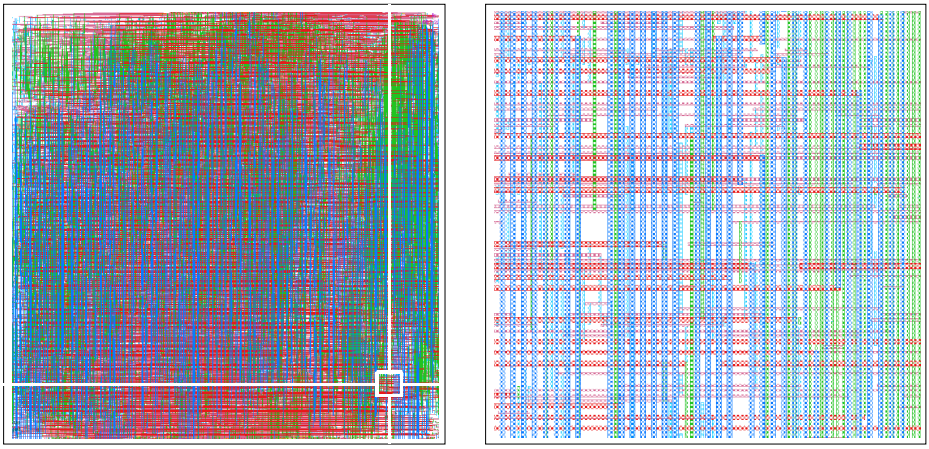


Figure 2.3: Screenshot of the routing result of a state of the art integrated circuit router. The example shows a microprocessor integer unit. Left: entire circuit, (0.55 mm × 0.55 mm). Right: zoomed view, the broadest layers wires measure about 400 nm in width. More than ninety percent of the wires run in the preferred routing direction of the respective layer. The preferred routing directions between adjacent layers alter perpendicularly.

2.1.3 Capacitance determination

Capacitance influences power linearly, cf. next section. The capacitance of the switched wires must be evaluated in order to obtain reasonable estimations for the power. In the following, established capacitance estimation methods are presented, and the capacitance model used throughout the thesis is developed.

3D-Extraction

After routing, cf. eg. Figure 2.3, a three dimensional representation of the layout describes the location and size of all physical objects. Two general methods have been developed in order to obtain accurate figures for the capacitances between these objects.

Deterministic methods, such as FastCap [NW92], divide the whole model into small elements. Then, electric flux lines are approximated numerically for every

element. A general disadvantage of such techniques is the high runtime. The accuracy of the extraction result is high, but is inherently limited by the floating point resolution and the element size. Hierarchical extraction methods [SLKY98, SLKY02] promise to speed up extraction by at least one order of magnitude.

Monte Carlo techniques are faster than deterministic methods. Using a random walk technique, [CI92][3] delivers a value that a capacitance is likely to have, together with a standard deviation. This is very useful since it is not important to know every capacitance in a circuit with the same accuracy. For example, in the case of power estimation, the capacitance of a reset line is irrelevant as it does not switch, cf. next section. The extractor focuses the CPU on critical nets. Furthermore, the accuracy of individual nets can be much lower – such as $\pm 10\,\%$ – than that of the objective function – such as $\pm 0.1\,\%$ – according to the general law of error propagation.

2.5D-Extraction

Especially during re-design iterations, 3D-extraction is too slow. A reduction in complexity is achieved by exploiting the layered assembly of CMOS. The set of different layers is limited and thus the number of significantly different capacitive coupling patterns between the layers. Several basic constellations of wires are extracted with the help of a 3D-extractor. The values are stored in tables. During 2.5D-extraction, the full chip is modelled as a superposition of these basic constellations and the capacitance is derived by table look-ups. Most design tools' built in capacitance extractors work this way.

Formulae used in this thesis

Final layouts generated in this thesis are checked with independent 2.5D, time-allowing even with 3D, full-chip capacitance extractors. During the optimisation process, simplified 2D capacitance formulae are used. For incremental local optimisations, these provide sufficient accuracy and allow to maintain reasonable runtimes. The model is developed below. See also Section 7.2.2 for an opportunity to optimise this model.

Following [MMP02], the capacitance for the middle wire in an arrangement of three parallel wires is evaluated. Figure 2.4 displays the three parallel wires which are assumed to run on a certain metal layer m. Throughout the experiment, the arrangement is symmetric, that is, the wire distance is d for both pairs. Their length

[3] Now QuickCap by Magma Design Automation.

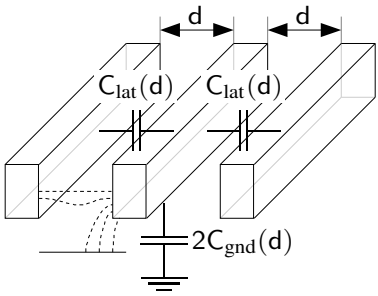


Figure 2.4: Capacitance extraction experiment.

is chosen relatively high so that the front panel couplings do not significantly influence the results. To account for the wiring in the layers above and below, the metal layers $m-1$ and $m+1$ are assumed to be empty, $m-2$ and $m+2$ are modelled as ground planes[4]. The full reason for combining all capacitances to other layers and the capacitance to infinity into one variable is provided in the next section.

Experiments are set up for every layer m in the given technology. The distance d is continuously increased from the technology-related minimum possible distance to a value where significant changes in capacitance do not occur any more, about ten times the minimum distance. At each distance, a new model is created and passed to a 3D capacitance extractor. The capacitance extractor measures the capacitance of the middle wire with respect to its two lateral neighbours C_{lateral} and with respect to ground C_{ground}. The values from 3D extraction are divided by the wire length l to obtain a per-unit-length capacitance as a function of wire distance. The results are plotted in Figure 2.5

To be able to evaluate the capacitance at arbitrary distances, a fit algorithm is used. Fitting provides an analytic representation of a curve drawn through a set of data samples. In this case, first order polynomials in d suit as fit functions for both capacitance parts:

$$C_{\text{ground},m}(d) = t_{\text{ground},m} + |s_{\text{ground},m}| d^{e_{\text{ground},m}} \quad (2.1)$$

$$C_{\text{lateral},m}(d) = t_{\text{lateral},m} + |s_{\text{lateral},m}| d^{e_{\text{lateral},m}} \quad (2.2)$$

[4]Where applicable. Modelling empty layers above and below layer m represents an optimistic case, i.e. underestimates the capacitances. The pessimistic corner is discussed in the Outlook, Section 7.2.2.

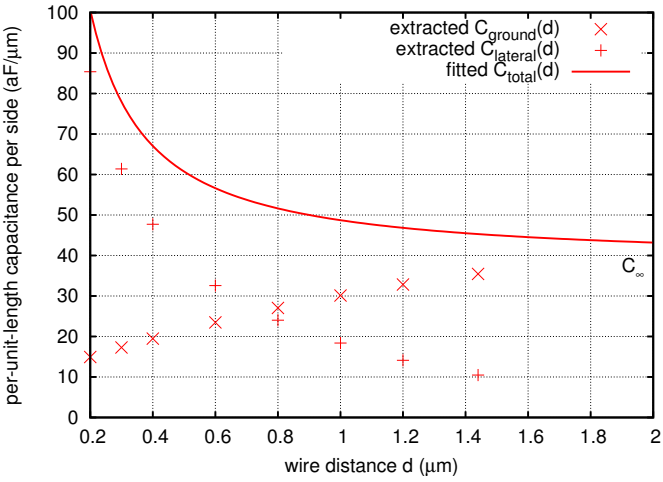


Figure 2.5: Capacitance fitting.

The constants t, s, and e are derived by the fitting process. For convenience, the metal layer index m is dropped and "lateral" and "ground" are abbreviated from now on. Obviously, $e_{\text{lat}} < 0$, reflecting the nearly inverse-proportional behaviour of a parallel plate capacitor. Less intuitive is the fact that $C_{\text{gnd}}(d)$ increases with increasing d. This can be explained by the decreasing shield effect for the coupling between the wire sidewalls and the ground planes, also known as fringe capacitances. The fringe capacitances converge at infinity. Thus, $0 < e_{\text{gnd}} < 1$. Figure 2.4 indicates the fringing field schematically.

Notice that the total ground capacitance of the middle wire is termed $2C_{\text{gnd}}(d)$ in the experiment and is valid if the left and the right neighbour are located at the same distance d. In doing so, also asymmetric constellations ($d_{\text{left}} \neq d_{\text{right}}$) can be evaluated:

$$C^{\text{wire}}_{\text{gnd}}(d_{\text{left}}, d_{\text{right}}) = C_{\text{gnd}}(d_{\text{left}}) + C_{\text{gnd}}(d_{\text{right}}) \tag{2.3}$$

$$C^{\text{wire}}_{\text{lat}}(d_{\text{left}}, d_{\text{right}}) = C_{\text{lat}}(d_{\text{left}}) + C_{\text{lat}}(d_{\text{right}}) \tag{2.4}$$

For a wire-by-wire evaluation, two distances are required. Sometimes it is practical to evaluate capacitances gap-by-gap. To evaluate gap capacitances of two wires at distance d, $2 \cdot C_{\text{gnd}}(d)$ and $C_{\text{lat}}(d)$ are used.

2.2 CMOS Power basics

2.2.1 Energy

The energy required to fully charge a capacitance C to a voltage V is widely known and given for example by [KSW95]:

$$W = C \cdot V^2. \tag{2.5}$$

Wires exhibit parasitic capacitances. As a result, energy is required to cycle the logical signal state of a bit. In order to obtain the wire power consumption of a gate that is loaded with a wire-related C, the signal event cycle rate $\alpha \cdot f$ must be known.

$$P_{\text{cap}} = \alpha \cdot f \cdot C \cdot V_{\text{DD}}^2 \tag{2.6}$$

This formula is valid for a physical capacitance C which couples to a fixed potential, such as V_{DD} or ground supply or a signal that is static. Considering that a wire generally couples to several other wires, cf. previous section, the following consequences apply. Lumping all parallel connected capacitances into a total capacitance is allowed if the adjacent wires do not toggle. Even if the particular capacitance components couple to different voltage levels at the time of the toggle event, Equation (2.6) remains correct [Yea98].

If two adjacent wires toggle simultaneously and in opposite directions, however, the effective voltage difference across the terminals of the coupling capacitance is $2 \cdot V_{\text{DD}}$. This results in a different effective capacitance value, a phenomenon referred to as Miller effect. From Q equals $C \cdot V$, and W equals $Q \cdot V$, the energy W needed to charge one wire is now double that of the case where the other one is static. Likewise, if both wires toggle into the same direction, their effective coupling capacitance becomes zero. Figure 2.6 summarises the possible transition cases for two wires that toggle individually or simultaneously. Mutual or effective switching activities provide a convenient way to model these effects on power.

2.2.2 Switching activity

The switching activity α_i of a particular net i may be treated as a probability for the net to cycle within a time unit. For the time unit, the clock period $1/f$ is chosen. This normalises α of the clock net to 1.0. In synchronous circuits, any signal net can change its state once per clock cycle. A full zero - one - zero cycle of the net

requires at least two clock cycles. In other words, α for signal nets is nominally limited[5] to 0–0.5.

Few nodes immediately imply an α value. The clock signal, if not temporarily gated, exhibits a value of 1.0. Power and ground signals do not toggle, $\alpha_{V_{DD}/gnd} = 0$. Also reset and scan signals are inactive during normal circuit operation. For most other signals, determining an accurate value for α is not trivial. To obtain a first system level figure, the easiest way is to assume a constant value for all nodes by averaging over the α values of a similar design.

Assuming a constant value is not always a good practise [Rab03], though, especially for optimisation purposes [SK96]. Switching activities may differ from node to node by many orders of magnitudes. For example, a least significant bit toggles more often than a most significant bit in a binary counter. For power optimisation, it makes more sense to minimise the capacitance of the least significant bit signal than for a quiet signal if there is the choice to minimise one of the two.

The switching activity is data dependant. For example, a general purpose microprocessor exhibits different switching activities for different programs it executes or for different input data of a program. The most accurate way to derive the switching activity is by netlist simulation with extracted gate and interconnect delays. A testbench applies activity patterns to the circuit model's inputs and traces all internal signals over a specified simulation time. The switching activity for a wire with respect to a constant voltage can by calculated by dividing the signal cycle count by the number of clock cycles elapsed. The effective switching activity c between two wires that switch simultaneously can be defined as

$$c := \frac{1}{N}\Big(1 \cdot (N_{RL} + N_{FH}) + 2 \cdot N_{RF}\Big)$$

with N simulation cycles in which N_{RL} rising transitions of one signal while the other signal is low, N_{FH} falling transitions while the other is high, and N_{RF} opposite transitions occur. These are the only events in which energy is drawn from the supply, cf. Figure 2.6.

The effective switching activity between two wires is compared to the switching activities of the single wires. Assume the two switching activities to be α_1 and α_2. Then, the effective activity equals $\alpha_1 + \alpha_2$ unless both of the following conditions are true:

1. The two nodes toggle simultaneously within the clock period. In that case, the "X-transitions" of Figure 2.6 can occur.

[5] Exceptions exist for glitches, cf. [Yea98].

2. The signals on the two nodes are correlated. Without correlation, double- and zero-energy transitions – these "X-transitions" – eliminate each other.

Simultaneous toggling is often avoided by design tools for signal integrity reasons, but can occur. The second condition is commonly true. Thus, the effective coupling activity differs from the sum of the individual switching activities in general. Deriving c is difficult, and no practical solutions are known, cf. Sections 7.2.2.

A lower bound and an upper bound of c as a function of α_1 and α_2 can be given as follows. Without loss of generality, $\alpha_1 < \alpha_2$ is assumed. If all common transitions occur simultaneously and in the same direction, c accepts its minimal value: $c_{\min} = \alpha_2 - \alpha_1$. Only those transitions of wire two are counted in which wire one is quiet (i.e. $\alpha_2 - \alpha_1$). On the other hand, if all common transitions occur simultaneously and in opposite directions, they have to be counted twice (i.e. $4 \cdot \alpha_1$), in addition to those transitions of wire two when wire one is quiet, $c_{\max} = 3 \cdot \alpha_1 + \alpha_2$. This results in maximal absolute and relative estimation errors of $\pm 2 \cdot \alpha_1$ and $-\frac{2}{\alpha_2/\alpha_1 - 1} .. + \frac{2}{\alpha_2/\alpha_1 + 3}$, respectively.

Unfortunately, gate level simulation is slow, but it is required to take all signal correlations into account. Some temporal and spatial correlations do not stem from the data applied to the circuit alone. Even for totally uncorrelated input vectors, such as random binary data on a bus, the logic gates down the path introduce correlation. For example, an AND-gate will not toggle if one of its inputs is low, and likewise, an OR gate does not toggle if one of its inputs is high.

Propagation of few given switching activities and signal probabilities through the logic is commonly used to estimate the correlation between the inputs and the outputs of every logic gate. It provides a popular compromise between accuracy and speed, although the resulting estimated activities can still be far from reality. Typically, the switching activities of the primary inputs are given through the stimuli. Spatial and temporal correlation between the signals, however, are dropped during propagation. Although there have been studies on taking correlations during propagation into account [MMP94, BLR04], no commercial propagation algorithm is known that handles mutual (effective) switching activities.

2.3 Conclusions

This section reassembles the formulae and observations mentioned earlier to the power consumption model used in this thesis.

The following equation assumes two wires of length l at a distance d. The power consumption caused by the switched capacitances is

$$P(d) = f \cdot V_{\mathrm{DD}}^2 \cdot l \cdot \Big((\alpha_1 + \alpha_2) \cdot C_{\mathrm{gnd}}(d) + c \cdot C_{\mathrm{lat}}(d) \Big). \tag{2.7}$$

The problem of the difficult to attain accuracy of the individual value of the effective switching activity c is addressed by putting it into the perspective of the high number of couplings in a design. Although approximating c by $\alpha_1 + \alpha_2$ sometimes overestimates and sometimes underestimates the power, the balance of thousands to millions power contributions cancels out the major part of the error [SK96, MKWS04][6] during power estimation. The effect on power optimisation will be quantified in the Experimental Results.

This step simplifies Equation (2.7) significantly as it eliminates c, and distinguishing between two capacitive components becomes no longer necessary. It is possible to use the same fit function pattern as in Section 2.1.3 for the total capacitance instead of using the weighted sum of lateral and ground part,

$$C(d) := t_{\mathrm{tot}} + |s_{\mathrm{tot}}| d^{e_{\mathrm{tot}}}, \tag{2.8}$$

with fitted constants $t_{\mathrm{tot}} = C(\infty)$, $e_{\mathrm{tot}} < 0$ and s_{tot} for every layer. Thus, the most often deployed formula in this thesis to express the power consumption caused by the coupling of two wires becomes:

$$P(d) = f \cdot V_{\mathrm{DD}}^2 \cdot l \cdot (\alpha_1 + \alpha_2) \cdot C(d). \tag{2.9}$$

[6] The same effect rectifies lumping all cross-over and cross-under capacitances into C_{gnd} as done in Section 2.1.3.

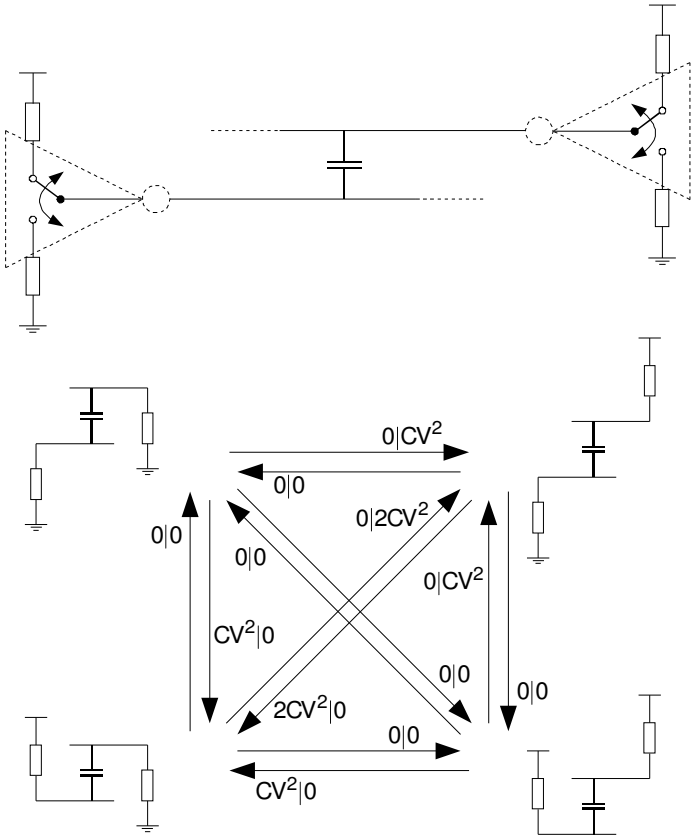


Figure 2.6: Top: two gates' output nodes couple capacitively. Bottom: energy drawn from the power supply to change the logic states at both terminals of the parasitic capacitance. The gates driving the two nodes are modelled as resistors, which is where all energy dissipates. The first and second values annotated to the transition arrows indicate the energy drawn by the left and the right gate, respectively.

3 Prior art

The problem of increased wire capacitance, and especially the lateral coupling component, has been raising the awareness of designers and researchers for a long time. Nevertheless, optimisation methods on low power routing are sparsely covered in the literature.

> "Like floorplanning, commercial place and route software is not yet power-driven and efficient algorithms are just emerging."

The above sentence is quoted from [LB97]. Almost a decade later, it still holds. For commercial software, there is the exception of low power placement and low power clock tree synthesis. But according to their data sheets, the layout flows of the three major physical design vendors, Synopsys, Cadence, and Magma, do not perform any other power-driven steps. In particular, the routers do not explicitly consider switching activities during global- or detailed routing.

Also the number of academic and industrial publications regarding routing or interconnect design with respect to low power is fairly low. This is reflected in the respective reference sections of the literature cited in these works. In major parts, these consist of links to standard literature, such as very general textbooks or to techniques with different objectives. For example, due to the lack of related work, the authors of [SYMY03c] describe a list of techniques that modify the coupling distance of wires for crosstalk noise and delay objectives, and evaluate their applicability to low power.

An inevitable common goal for any power-driven physical design step, is to minimise the switched capacitance,

$$\Sigma_i \alpha_i C_i. \tag{3.1}$$

Certain techniques [DDM96, SYMY03c] use existing routers and modify their input parameters. Existing physical design steps, however, reduce $\Sigma_i C_i$ instead. In particular, placement and routing target minimal wire length, irregardless of the switching activities of particular wires. Unfortunately, the total capacitance does not correlate very well with (3.1) [LB97]. Therefore, methods that minimise

the $\Sigma_i C_i$ objective do not minimise the power but rather by-produce a decreased power consumption.

The remainder of this chapter provides three sections to classify the prior art. Section 3.1 describes known low power routing techniques that apply to entire circuits. Specialised techniques that tend to exhibit more academic character, appear in the second section. To fully define the scope of the underlying work, the last of these sections provides some examples of related techniques and objectives not covered by this work. The final sections, Section 3.4 and 3.5, link the prior art to the present approach and summarise this chapter, respectively.

3.1 Entire circuits

All publications cited in this section share one property. They use an existing commercial router and modify its input parameters. This strengthens the supposition that dedicated low power algorithms for modern processes are not known. Following an early work of Marc Laurent and Michel Briet [LB97], guidelines for low power during routing can be:

1. Order the routing by net activity.
2. Preferred routing levels must be the upper metal levels.
3. The routing pitch may be relaxed.
4. If the previous item is not possible, local optimisations must be performed.

Interestingly, there appears one work on each of the points above, with the exception of the last one. The subsequent sections resemble this enumeration. They carry the name the work that represents the respective item.

3.1.1 Interconnect-power dissipation in a microprocessor

The *Mobile Platform Group* of *Intel Haifa, Israel* published a paper on interconnect statistics and the benefits of routing for low power by driving the order of routing by activity. The authors Nir Magen, Avinoam Kolodny *(Electrical Engineering Dept., Technion, Haifa)*, Uri Weiser, and Nachum Shamir *(Intel Israel (74) Ltd., Haifa, Israel)* include the switching activities of the nets during routing to a certain extent. Only a top fraction of the power critical nets are

handled in a special way, while the remainder of the nets undergoes the normal routing process.

Wire statistics in seven figures build the major part of the paper. These provide valuable information about net length distributions of several Intel Pentium processors. One "Low Power Processor 130nm", presumably one of the Pentium family, serves as a basis for the rest of the investigations. Power and capacitance breakdowns and other statistics for this processor are reported.

One general conclusion of the first part is that in a net length histogram of an Intel processor, "the number of nets decreases exponentially with net length," which is displayed for seven different processors of five different technology generations. A second important conclusion regards the almost even distribution of switching activities among "all the net lengths." Finally, as an important result also for this work, the contribution of the capacitive net switching power to the total dynamic power of the net is amounted to about 50 % on average. This value can range to as high as over 90 % for individual global interconnects. Unfortunately, the static power consumption of the design is not reported. However, even with conservatively estimated leakage currents, the optimisation potential becomes evident. A suggestion to exploit the statistics follows.

It might have been corporation politics or an early development stage that only little over two columns have been devoted to the presentation of the optimisation idea, including figures and results. Two possible methods for switched capacitance reduction are suggested. The wire length of a net measures about 30 % above its theoretically possible minimum after routing, on average. Therefore, the first strategy comprises a length reduction for high activity nets in exchange for a length increase of low activity nets. Second, double wire spacing is suggested to expose the wires to less lateral capacitances.

Since both methods are impossible to do for all nets, the authors propose to use only a fraction of the higher-activity nets for optimisation. This is justified since "90 % of the total dynamic power is consumed by 10 % of the wires." The authors modified the input parameters of a not further specified commercial router.

The methodology consists of configuring an existing router to use twice the minimum spacing during clock tree synthesis. Then, the top n % of the most active nets – the exact number is derived empirically to below 5 % but not published – are routed before all others. Also the spacing rule is doubled for these nets. In doing so, lengths of the most active nets are minimised and their distances to others are kept high.

Afterwards, the remainder of the nets is globally and locally routed in a normal way. As pre-routing in a user-specified order causes congestion problems, routing

of the non-critical nets may fail. In such a case, routers normally resort to "rip up and re-route". Since ripping up and re-routing a power-critical net renders the previously gained saving invalid, "the rip-up mechanism was enhanced, allowing it to select a power-critical net only if no other blocking candidates exists."

Experimental results show dynamic power savings of four to 18 percent for different processor blocks. Since the capacitance is reduced for $n\%$ of the nets, positive timing slack is created. The authors could reduce the driver sizes on the according paths to save additional power, in total up to 50% dynamic power. These numbers appear attractively high, but a closer look is not provided. It is pointed out that the numbers are valid for a single case study. "Further studies of other processors should take place to verify the results." Runtime numbers are missing in the report.

3.1.2 Exploiting metal layer characteristics for low power routing

Between 2002 and 2004, Armin Windschiegl, my predecessor in the Institute, published [WZS02, Win04]. He observed higher metal layers to exhibit up to 42% less typical per-unit-length wire capacitances in a given $0.25\,\mu$m process. Thus, assigning the more active wires to the upper layers and the less active nets to the lower layers was expected to reduce the switched capacitances. An additional aspect was the distribution of the total wirelength to all layers to increase the average wire distance. The following recipe was developped:

To avoid congestion problems, timing critical nets are routed first. Then, the remaining nets are sorted by activity, and an algorithm estimates the wire lengths for every net. Given an m metal layer process and an estimated total wire length l, the most active wires that, combined, measure $2l/m$ are assigned to the top metal layer-pair, the next $2l/m$ to layers $m-2$ and $m-3$, and so on. If local congestions cause a wire to be exposed to higher capacitances than supposed to, a net may be routed on any layer.

The work does not contain an application example. The reason is that the commercial router used in that time, Cadence Silicon Ensemble, did not provide enough configuration options to fully implement the idea.

3.1.3 Routing methodology for minimising interconnect energy dissipation

In 2003, researchers Atsushi Sakai, Takashi Yamada, and Yoshifumi Matsushita at *Sanyo Electric Co., Ltd., Gifu, Japan* and Hiroto Yasuura of *Kyusyu University,*

Fukuoka, Japan published studies [SYMY03a, SYMY03c] on addressing general capacitive coupling effects of entire circuits caused by uniform routing grids. Sakai's low power method is actually a modification of an earlier work [SYMY03b] that discussed a similar technique targeted to crosstalk noise and delay. As that work proved to be successful, given a 15 % delay reduction for the example circuit used, porting it for low power seems rectified.

The application of this method for low power consists in increasing the routing pitch for the routing layers. Routing at increased pitches under a timing constraint, however, quickly congests dense areas. The commercial router used in that work did not succeed in completing the design with increased M1 or M2 pitches. Therefore, the low metal layers were excluded from widening the pitch.

The standard wire pitch was multiplied by 1.25, 1.5, ..., in 0.25 increments. Different sets of routing layers to which the new pitch applies were defined. Layers M3–M6 and M4–M6 were considered for a six metal layer 130nm technology and layers M3–M5 for a five metal layer 180nm technology. A new routing process was started for every combination of wire pitch factor and set of layers.

Upon completion, power is estimated, and the method selects the best of all routing solutions. In general, this multiplies the routing time. One point in the two dimensional solution space (wire pitch, layer set) that optimises one circuit, however, can be used for similar circuits and with similar constraints as well.

The example under consideration was a ULSI[1] image processor with 100,000 instances. Power measurements were done post layout but unfortunately with constant activities of 0.2 rather than with simulated or propagated activities.[2] The reported power savings reach up to 9 %. Unfortunately, it is not specified if this number refers to the capacitive switching energy or to the total power reduction.

The maximum in power saving occurs at a pitch of 1.5 times the minimum pitch applied to metal layers M3–M6. A further increase in the wire pitch did not pay off. Costly wire detours emerge at higher wire pitches as routing resources diminish. Area constraints are reported in [SYMY03b] only. In that work, the standard cells utilise 40 % – 60 % of the available core area.

The paper reports the runtime increase for the optimal configuration. For the given instance, the runtime of the router nearly four-folds.

Advantages of this method are its simplicity and the independence of switching activities which can be difficult to obtain. Disadvantages of this work include

[1] ultra large scale integration

[2] Constant activity factors can be assumed during high level power estimation. For optimisation, however, they do not provide enough accuracy. [SK96, Yea98]

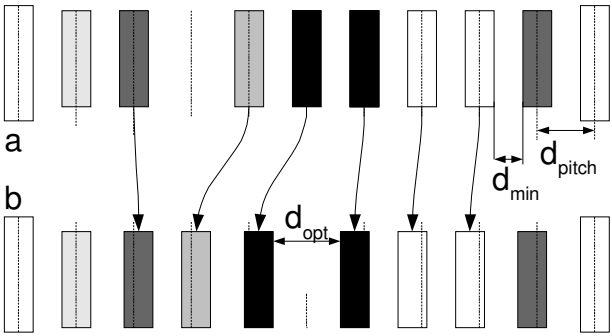


Figure 3.1: This figure illustrates activity-driven wire spacing for low power. The darker the wire, the more active, and the more isolated it gets placed.

the limitation to layouts with low area constraints due to the inefficient usage of space for low activity wires. This will be discussed in Section 6.2.2 in more detail. Also the runtime of a router configured for different spacings at different layers increases significantly.

3.2 Specialised wire topology optimisations

All low power interconnect design methodologies in this section deal with the special case of parallel wires. Optimisation options for this case consist of spacing, ordering, and combinations thereof.

3.2.1 Spacing parallel bus wires

Increasing the space between parallel wires probably constitutes the most intuitive method for low power interconnects. The principle is displayed in Figure 3.1.

Given is a bus (a), represented by a set of n parallel wires, and there exist unused routing resources. Two extra shield wires enclose the bus. The minimum possible physical width of the bus is $(n+1) \cdot d_{\min}$ plus the wire widths, where $d_{\min}$ denotes[3] the minimum allowable distance between two lines. The idea is to place the wires off-grid (b), so that to exploit the unused space. An individual distance is assigned to each wire pair.

[3] All variables and symbols follow the convention of this document. Original work may use different symbols for the same meaning or same symbols with different meaning.

Two methods [MMP02, RNS05] appear in the literature to solve the wire spacing problem, not including the publications emerged from this dissertation [ZMS05, ZGRS05, ZWdO+05, ZBI+07]. They differ in the assumptions about the capacitance model used, the amount of extra space, border conditions, and the allowed value range for the resulting wire positions. These differences lead to very distinct solution approaches for basically the same problem.

Wire placement for crosstalk energy minimisation in address buses

Politecnico di Torino was the first to publish a paper [MMP02] on wire spacing for address buses in the year 2002. The authors, Luca Macchiarulo, Enrico Macii and Massimo Poncino, respect not only the decrease of lateral capacitance but also the rise in ground capacitance at increased spacings. This is necessary because for the bus assumed, all wires are driven by locally close flip-flops and thus toggle simultaneously within one clock period. The ground capacitances $C_{\text{gnd},i}$ exhibit a log(d)-shaped dependence on the distances d_i. The characteristic of the wire-to-wire or lateral capacitances $C_{\text{lat},i}$ exhibits the well-known inverse proportional picture. Both capacitance types contribute to the total energy with different weighting factors, so that the task becomes to minimise

$$\Sigma_i \alpha_i \cdot C_{\text{gnd},i} + c_i \cdot C_{\text{lat},i}, \tag{3.2}$$

where α_i denotes the switching activity of wire i, and c_i combines the cross-coupling activities of wire i to wires $i-1$ and $i+1$, cf. Section 2.2.2. A heuristic is presented. An outer loop examines every wire. An inner loop moves the wire to the left and then to the right until it violates the minimum distance constraint. After every move, the power is evaluated. The program records the current wire constellation if it is the best one that occurred so far. The best solution is accepted after executing the method N_{runs} times with random starting positions. The heuristic exhibits the runtime complexity $O(N^2 \cdot N_{\text{runs}})$ with N representing the number of wires.

ARM and MIPS[4] instruction set emulators are used to generate several instruction address bus traces for different programs. The traces are necessary to count signal transitions and thus to calculate the activities α_i and c_i, cf. Chapter 2.

It is evident that distributing more space around highly active wires than around less active wires saves more power than distributing an assumed amount of space uniformly. The effect was quantified in the paper by comparing a uniformly spaced

[4] ARM and MIPS are popular microprocessor families.

bus with an activity-driven, non-uniformly spaced bus. Four times as much power reduction potential was shown for the non-uniformly spaced bus compared to the uniformly spaced bus at an assumed extra area of 10 %. The power reduction is measured with respect to an area-minimal bus, i.e. parallel wires at the minimum distance. Differences diminish with increasing amount of extra area.

Power savings lie between about 15 % and 40 %, depending on the extra space added (up to 30 %) and the assumed initial wire spacing. The authors also asses the generality of a spacing solution. Once a bus is laid out and fabricated, it should also comply with modified activities stemming from a different program. The power saving results do not drop considerably when executing a program on a bus that was spaced with activities stemming from a program mix. This is an important result.

The only weak point in the method is the runtime. It lies in the order of minutes for a 32-bit bus. This problem is solved with an improved approach in Section 4.1.

An efficient algorithm for wire spacing

In [RNS05], Edwin Naroska, Uwe Schwiegelshohn (both *University Dortmund, Computer Engineering Institute*) and Shanq-Jang Ruan (*National Taiwan University of Science and Technology, Dept. of Electronic Engineering*) provide an exact algorithm for solving a simplified wire spacing problem, compared to the previous approach.

Several assumptions are made. First, the used capacitance model includes the lateral capacitances between two wires but not the ground capacitances. Not modelling the ground capacitance neglects the effect of increased fringing and thus increased power dissipation for increased distances. The ground capacitance, however, typically exceeds the lateral capacitance at distances around $2d_{\min}$, cf. [MMP02]. Mathematically, the first term in Equation (3.2) on page 26 is dropped. Second, the resulting distances are generally not integer multiples of the manufacturing grid. One can overcome the violation of the integrality with a rounding technique after calculating the optimal distances [ZBI+07].

The resulting algorithm assigns an optimal distance to every wire pair in quasilinear time. Experimental results in this work are provided, but the power saving opportunities for wire spacing alone are not listed since the method is chained with bus inversion and permutation techniques, cf. Section 3.2.3.

3.2.2 Permuting parallel bus wires

Assume a permutation π of the numbers $\{1\ldots,N\}$. Then,

$$(\text{wire}_0, \text{wire}_{\pi(1)}, \ldots, \text{wire}_{\pi(N)}, \text{wire}_{N+1}) \tag{3.3}$$

denotes a *wire ordering* and $\mathscr{W}$ the set of all wire orderings.

There exists a power-optimal wire ordering over $\mathscr{W}$. Two situations may be distinguished.

1. Simultaneous wire switching is assumed. In this case, the coupling activities c_i of (3.2) are not constant but depend on the order. No additional wire space is required to achieve a power reduction.

2. Wire ordering is combined with wire spacing. The order of the wires influences the efficiency of optimal capacitance sharing through wire spacing, even if no simultaneous switching of neighbouring lines occurs.

Item 1 is described in the following. The first work discusses wire ordering for the specialised case of a multiplexer, while the subsequent three treat more general cases. Section 3.2.3 provides case 2. The optimal wire order for uncorrelated or non-simultaneous switching is derived in Chapter 4.2.

Reducing cross-coupling among interconnect wires in deep-submicron datapath design

Joon-Seo Yim (*DSP Group, LG Korea* and Chong-Min Kyung (*Department of Electrical Engineering, KAIST, Taejon, Korea*) received the 1999 Design Automation Conference (DAC) best paper award in the category *Technology driven design methodologies*.

[YK99] provides a control signal ordering scheme for full-custom data path designs. The authors ordered the control signals of a multiplexer such that no couplings in opposite directions can occur. This saves 10 % wire power of the multiplexer.

Coupling-driven bus design for low-power application-specific systems

Youngsoo Shin and Takayasu Sakurai, University of Tokyo, contributed to the topic with a study on bus optimisation by shuffling the bus lines in July 2001. In their work [SS01], the solution is approached with two heuristic algorithms.

The first one searches the bus wires for shields. Shields are wires that exhibit a switching activity below a certain threshold value. The remaining wires then build cliques such that the wires of each clique show switching correlations close to one, "that means that they have high possibility to have transitions in the same direction." Finally, an alternating arrangement of the cliques and the shields resemble a solution to the problem.

A simulated annealing approach (SA) constitutes the second heuristic. SA allows for a move in the solution space towards a worse than the current solution with a certain probability. As time advances, this probability decreases and the heuristic turns into a steepest gradient descent. In the given case, a move is a "one-to-one exchange between randomly selected bus lines (...) or a group-to-group exchange between randomly selected two groups of bus lines. The move itself is chosen randomly."

The methods are tested on instruction and data address bus traces of different processors and different programs. Both approaches are reported to reach power savings of up to 60 % of the bus wire power in a 0.35 μm technology. SA has a slight advantage on power savings but needs two orders of magnitude more CPU time to get there.

Low-energy encoding for deep-submicron address buses

Also the optimisation presented in [MMP01][5] targets the cross coupling energy of the bus by permuting the bus wires. It appeared in August 2001. Simultaneous adjacent transitions of different polarities require about twice the cross coupling energy of two serial transitions, and uniform transitions require almost no cross coupling energy.

It seems lucrative to build pairs with uniform transition polarities and avoid those with opposite directions. The predictability of the average transition polarities is assured by limiting the studies to address buses. The authors reduced the problem to a special instance of the travelling salesman problem (TSP) which is known to be $\mathcal{NP}$-hard.[6]

A special heuristic is leveraged to find a solution near the power optimum that is reported to lie 26 % below the power of a non-permuted bus on average. A great advantage of this method is that there is no need for encoding or decoding hardware.

[5] The work appeared without the word "Encoding" in the ACM digital library.

[6] In simple terms, it is unlikely that a solution can be verified in less than combinatorial runtime.

However, if the endings of the bus wires are fixed, a special permutation network has to be routed in order to accomplish the desired wire order. A slight modification of a left edge channel router is proposed for this task. The power, area, and timing overheads of this network are reported to be quite low, less than 4 % each for a 1 mm bus. It requires to be routed on at least one additional routing layer.

Value-based bit ordering for energy optimisation of on-chip global signal buses

The same problem as above is described by Krishnan Sundaresan and Nahir Mahapatra (*Dept. of Elect. & Computer Engineering, Michigan State University, East Lansing*) in [SM06]. The authors contribute by using a publicly available travelling salesman solver. In addition, it is no longer assumed that the bus must be enclosed by two additional shield wires. The bus energy reduced by 30 % and 17 % for instruction and data buses of Spec CPU2000[7] benchmarks, respectively.

3.2.3 Combined spacing and permuting parallel bus wires

Combining wire swapping and spacing for low-power deep-submicron buses

The third paper of *Politecnico di Torino* on the topic is [MPS03]. It provides a combination of the optimisation principles of changing the relative positions and inter-wire spacing. Again, a heuristic algorithm is developed. It starts with a certain wire as current configuration. Then the least coupling wire is added to the system at a certain distance that depends on its activity and its power contribution in a reference solution. For the same examples as above, the power savings now reach impressive 50 % on average.

An efficient algorithm for simultaneous wire permutation, inversion, and spacing

Also [RNS05], cf. "An efficient algorithm for wire spacing" in Section 3.2.1, combines wire spacing with permutation. The authors develop a genetic algorithm and a fast algorithm to find a new wire order and then apply wire spacing described in Section 3.2.1. The wire permutation step does not target a minimal power consumption through a minimised number of opposite transitions of neighbouring

[7] http://www.spec.org/cpu/, March 2007, last update March 2007

bus wires. It rather "binds the major coupling effects to a set of small wires". Increasing the space for the small set of wires drastically reduces the power. Reported values are 50 % average capacitive bus power reduction for a PISA[8] processor benchmark.

3.3 Related techniques and objectives

3.3.1 Wire spreading

Wire spreading is not a power optimisation technique per se. Its objective is to minimise the number of particle-related defects on a waver in order to increase the chip yield. The wires are locally re-distributed to decrease the critical area, i.e. the area on the chip in which a particle creates a short. Decreased coupling capacitances and thus a decreased power dissipation are by-products.

Wire spreading can be done during or after routing [Leu03]. As early as 1997, Jeffrey Z. Su and Wayne Wei-Ming Dai, *Dept. of Computer Engineering, University of California, Santa Cruz* presented "Post-route optimisation for improved yield using a rubber-band wiring model" [SD97]. This technique is based on an academic router of 1992, "SURF", which models the interconnects as rubber-bands. Today, all of the big design tool companies as well as small companies such as ICYield[9] offer wire spreading for yield [Leu03, LC04].

3.3.2 Placement

The output of the placer influences the routing lengths. Keeping those cells closely together that are connected by active nets seems lucrative. There have been early suggestions [Ped96] as well as recent implementations [OJ04, CHK+05] of weighted wire length minimisation through partitioning, floor planning, and placement. Reported power savings are around ten percent. The method in [OJ04] also leads to an even temperature distribution over the die.

3.3.3 Power driven routing using a genetic algorithm

B. M. Goni and T. Arslan of *Department of Electronic and Electrical Engineering,* University of Edinburgh, and B. Turton of *Cardiff School of Engineering,* Cardiff

[8] a MIPS derivative

[9] http://www.icyield.com/index.html, November 2006

University, published a channel routing method using a genetic algorithm[10]. Unfortunately, channel routing has almost completely lost popularity since more than two layers of metal and over-the-cell routing have become state of the art. Therefore, it was not further investigated in this work.

3.3.4 Equivalent pin assignment

Equivalent pin assignment has now been a standard technique in Synopsys products for years. Functional symmetric gates may expose different pin capacitances. Thus, power can be reduced by controlling how nets with different switching activities are connected to equivalent pins.

3.3.5 Bus coding

Much more publications discuss saving energy through bus encoding than through spacing. The common idea is to reduce the switching factors α_i and c_i by adding a coder-decoder pair at the bus ends. Published power savings reach several ten percent. Claudia Kretschmar et al., however, noticed that none of the bus coding papers published the energy overhead caused by the codec logic. In her work [KNM04], she concludes that bus lengths would have to measure in the order of 10cm to break even, assuming that the codec logic is synthesised in a 130nm standard cell library.

3.3.6 X-Initiative

The first microprocessor, the Intel 4004, used diagonal ("X") routing instead of manhattan-routing[11]. The advantage on power is obvious: In the best case, the wire length and thus its contribution to the power dissipation, reduces by a factor of $\sqrt{2}$.

For simplicity reasons, current state of the art manufacturing technologies and physical design algorithms do not support diagonal routing. "The X Initiative was created to advance the usage of the X Architecture by ensuring support for the X Architecture throughout the design and manufacturing cycle.[12]" The major EDA

[10] http://www.see.ed.ac.uk/~SLIg/papers/isas_sci.pdf, November 2006, last modified July 1999

[11] http://download.intel.com/museum/research/arc_collect/history_docs/pix/4004.jpg, November 2006

[12] http://www.xinitiative.org/Default.aspx?tabid=31, November 2006

suppliers are currently re-writing their algorithms to support 45 degree routing. First tape-outs occurred.

3.4 Present approach in context of prior art

This work aims to reduce the power of entire circuits. Since a routing solution must be competitive in completion rate, design rule cleanness, timing-, signal integrity- and manufacturability closure, a method is desired that improves upon existing routers, which already respect all these constraints [Leu03].

Dedicated low power routing algorithms for modern multi-layer technologies are not known. Commercial EDA vendors would never provide access to their source codes to incorporate changes. The prior art has shown varying success on just modifying routing parameters. Especially area constraints limit their general applicability.

Thus, a post-routing optimisation technique is favoured for this work. The present thesis fills the last item of Section 3.1 on page 21, that is, a method is provided that performs local optimisations that do not cost area. All optimisations described originate from modifications of the wire distances, not their lengths. No logic is added or altered. Weighted capacitances, Formula (3.1), as in the contributions on bus power optimisation (Section 3.2) are optimised, not total capacitances. "Power driven wire spacing" may circumscribe it aptly. Table 3.1 summarises the afore-mentioned points. Notice that the respective works base on too different assumptions about the underlying technology so that comparisons in power saving numbers are not summarised at this point.

The presented optimisation method searches for sets of parallel wires in the entire design, and applies a specially tailored power-driven wire spacing algorithm. Compared to the already published low power spacing algorithms, there are important differences. These are listed in Table 3.2.

Also the existence of an optimal wire order for low power CMOS was observed (Section 4.2) and proved (Appendix A) during the work on this thesis. The general difference to other ordering problems, as described in Section 3.2.2, lies in the assumption that no simultaneous switching of adjacent wires occurs. Wire ordering is difficult to implement. It will therefore rest in the theory state in this thesis. Table 3.3 summarises the contributions to wire ordering for low power.

Table 3.1: Comparison between techniques for entire circuits.

Name / Reference	Activities included / Distances modified	Remarks
Intel [MKWS04]	two classes / two classes	Also adjusts net lengths by changing the routing order. Area limits applicability.
Windschiegl [Win04]	yes / indirectly	Assigns high-activity nets to layers with low specific capacitance and vice versa.
Sanyo [SYMY03c]	no / globally increased	May increase area or fails under tight area constraints. Activity-independent.
Spreading e.g. [SD97]	no / uniformly	Actually a yield improvement technique. Low power is always a by-product
TopCool Chapter 5	yes / individually	Finds and spaces parallel wires. Exploits whitespace locally. No area overhead.

Table 3.2: Comparison between power-driven parallel wire spacing techniques.

Group / Reference	Detours / Complexity	Application / Exact	Remarks
Torino [MMP02]	no / $O(N^2 \cdot N_{\text{runs}})$	address bus / no	The heuristic algorithm considers simultaneous switching power.
Dortmund, Taipei [RNS05]	no / quasilinear	instruction, address bus / yes	Is actually part of swapping and inverting. Does not consider manufacturing grid.
TopCool Section 4.1.2	yes / $O(N^2 \cdot \log N)$	entire circuits / quasi	Building block of low power approach in present thesis.

Table 3.3: Comparison between power-driven wire permutation techniques.

Group Reference	X-transitions Complexity	Remarks
Tokyo [SS01]	yes low	No spacing required.
Torino [MMP01]	yes (unknown)	No spacing required. Formulation as travelling salesman problem (TSP).
Michigan [SM06]	yes NP-hard	Uses exact solver for TSP. Considers edge effects.
Torino [MPS03]	yes (unknown)	Integrated ordering and spacing solution.
Dortmund, Taipei [RNS05]	yes low	Ordering used to prepare for spacing.
TopCool Section 4.2	no quasilinear	Proof of optimality. Increases the effect of wire spacing alone.

3.5 Summary

There is very little literature on power-driven layout and particularly on routing. Earlier works such as [LB97] or [Ped96] restrict themselves to general guidelines only. While the major EDA vendors provide a great variety on low power methods on higher levels in the design flow, still virtually no low power support exists during routing.

Intel and Sanyo published papers [MKWS04, SYMY03c] in which the routing pitch of existing design flows is modified. In addition, [MKWS04] uses a routing order driven by node activity. The main advantage of relaxing the routing pitch for less lateral capacitance is simplicity as the existing router does all the work. There are limitations, however, as already pointed out in [LB97]. Increasing the pitch also increases the wirelength and therefore the power consumption. If overdone, a power minimum is crossed. Converging slows down the design flow since a new routing process must be started for every pitch considered. An area constraint can forbid a globally relaxed pitch at all. In that case it depends largely upon the capability of the underlying commercial EDA software to use a relaxed pitch locally. There are spreading techniques [SOC06] for yield which reduce the total capacitance as a side effect. Power, i.e. switched capacitance, does not correlate well with total capacitance, and switching activities are not considered during wire spreading.

Academic work on low power routing, such as the work by *Politecnico di Torino* [MMP01, MMP02, MPS03] focuses very specialised areas. Wire spacing and permutation for address buses is an attractive optimisation problem, but it depends on the correlation of the bus signals. This correlation is given if the optimised bus is used for one specific purpose only, such as instruction or data words or addresses. Moreover, it must be a bus that is physically reflected by parallel wires in the detail-routed design. On bus-centric system-on-chips, however, there exists one central multiplexed bus, such as AMBA or PLB, that is shared by the components over time. As a result, data words, instructions and addresses of different sources may traffic the bus. The correlation is lost. In addition, if hierarchical designs are synthesised flatly, buses are not likely to be routed as parallel wires next to each other.

This work, *TopCool*, also develops a power-driven parallel wire spacing technique. Its use, however, is not intended to exclusively space buses. It is rather a building block for the optimisation of entire circuits.

4 Parallel wire optimisation

This chapter contributes to two theoretical wire optimisation concepts, activity-driven wire spacing and wire ordering to decrease the power consumption of parallel wires. Contributions to wire spacing allow for its application in arbitrary detail-routed circuits. All required algorithms that directly affect parallel wires are detailed in Section 4.1 while the practical means required to apply spacing to optimise given entire circuits is the content of Chapter 5. During the research work carried out, effects of the wire order on the benefits of spacing were observed and documented in Section 4.2. Possible practical applications for ordering are described in the Outlook, Section 7.2.7.

4.1 Parallel wire spacing

Wire spacing is a method for the derivation of particular distances between wires on a chip. For simplicity reasons, it is defined for the manageable case where few long wires run in parallel. Wire bends, layer changes or wire splits are not considered.

Re-distancing wires directly translates into new capacitance values which influence timing, crosstalk, and power. Furthermore, wire distance relates to the probability of a particle-related short. Distancing wires can be accompanied with an increase in circuit area if the placement is wire-limited.

Since non-uniform spacing has shown high power saving potential for buses, cf. the work of Luca Macchiarulo [MMP02], described Section 3.2.1, the principle deserves a closer look. The goal of this thesis, however, is not to route optimised buses. The goal is to optimise the power consumption of entire circuits to which buses contribute only a certain fraction. In many cases, all nets – including bus wires – are routed in the style of the “rat’s nest”, a colloquial expression for wires routed at random positions and with random topologies.

A major difference between bus routing and signal net routing regards the number of nets for which a suitable route through the circuit has to be found. Bus routing generates one to few connections between circuit blocks in hierarchically synthesised designs. The bus wires are usually kept together and follow the same

physical path. Signal net routing is carried out for all other nets in a design. Each net obtains a dedicated topology. The much higher difficulty results from the high number of nets to be considered.

Another aspect that contrasts routing straight buses to optimising already-routed wire topologies is circuit area. Bus routing assumes enough free routing resources before constructively laying out the parallel running wires. Post-routing optimisation has to find suitable parallel wire segments in a given layout after routing. Conditions to qualify for wire spacing include sufficiently long, movable, parallel running wire pieces and a certain amount of whitespace. Consequently, bus wire spacing is superseded by post-routing wire spacing.

The spacing process itself must be considered as an in-place modification. It is required to optimise the wire topology locally without disturbing the electrical behaviour of the circuit. In particular, wire pieces that are moved laterally must not touch other nets or lose connection to other wires of the same net. Further, in-place wire spacing relies on the existence of whitespace after detailed routing. This is a strength rather than a weakness, since most detail-routed layouts can not use every possible routing track. Intractable congestion problems would prevent routing completion when constraining the number of available layers and standard cell row utilisation too tightly.

A method of finding qualifying parallel wire groups is described in Chapter 5. In the present section, parallel wires qualifying for wire spacing are assumed, and in-place wire spacing is developed. This requires three major properties of a wire spacing method that have not been covered in literature so far.

Speed First, the wire spacing technique deployed must be fast, considering that the number of wires in a design is roughly one order of magnitude higher than its number of nets. The number of nets, in turn, can be amounted to about the number of standard cells which may reach hundreds of millions as of today. While the runtime to space few buses with in the order of magnitude of ten to one hundred wires is harmless, it clearly becomes an issue when compared to the number of wires in an entire design which can reach billions.

Connectivity Second, wire spacing techniques in the literature assumed physically dangling wire endings. This means that the terminal pins of the object that the wires are connected to are assumed to move as well. This is generally not possible for a set of parallel wires in a detail-routed layout. A mechanism is required to keep the wire endings connected to their

original positions. This detouring between the old and new wire positions introduces extra wires that exhibit parasitic resistance and capacitance and may therefore cause a delay and a power overhead. This effect must be considered during wire spacing.

Area Third, the production cost depends over-linearly on area. Bus wire spacing assumed an area investment for a power return. However, an increase in cost is often not acceptable. Methods that globally increase the routing pitch such as [SYMY03a] are ineligible if the area has to be increased prior to routing to alleviate congestion problems introduced by relaxing the pitch. If the whitespace remaining after detailed routing is exploited and relocated from quiet to active wires, then no area investment is required while effectively reducing the power.

In the following, these three requirements are discussed in more detail. Fast wire spacing is achieved by formulating the problem as a convex program for which efficient solution methods exist. In a next step, signal correlations are neglected. This greatly accelerates the design flow since correlation coefficients can be expressed by switching activities, and sophisticated methods for approximating switching activities exist, cf. Section 2.2.2.

Then, the convex program is extended in order to consider new connection wires required for the new and old wire endings. Area preservation is ensured by proper constraints in the optimisation problem. A solution strategy is presented that solves the problem quasi-exactly and in low runtime.

4.1.1 Fast wire spacing

Fast wire spacing techniques consider the convex property of the problem. A second speed-up is a by-product of dropping the consideration of correlation of coupling wires. This is done because of the difficulty to obtain proper values for c. For justification of this step, the loss of power optimisation potential and the error in power estimation when ignoring correlation coefficients of simultaneously switching wires is shown in the Results, Section 6.3.

Formulation of bus wire spacing as integer-convex optimisation

Figure 4.1 displays the general wire spacing problem for buses as found in the literature [MMP02]. Wire endings are assumed to be dangling and the wire length

is assumed to be long so that inhomogeneities in the electric field at the wire endings can be neglected.

The power dissipation caused by the wires depicted is derived by summing up N instances of Formula (2.7). This results in Formula (4.1), the objective formula of a minimisation problem. The technology-specific minimum distance[1] $d_{\min}$ needs to be maintained, cf. Constraint (4.3). Thus, in order to legally route N wires, at least $(N+1)\cdot d_{\min}$ of total space must be present. Additional space can be used to distribute it among the wires. Following the convention of [MMP02], the total available space can be defined as $W := (1+\sigma)\cdot(N+1)\cdot d_{\min}$ where σ is a factor that represents an assumed amount of additional space, such as 10%, cf. Constraint (4.2). An alternative way to express the excess space is to distribute the N wires on $M > N$ tracks.

$$\sum_{i=1}^{N} \alpha_i \left(C_{\text{gnd}}(d_i) + C_{\text{gnd}}(d_{i+1}) \right) + c_{\text{l},i} C_{\text{lat}}(d_i) + c_{\text{r},i} C_{\text{lat}}(d_{i+1}) \quad = \quad \min! \tag{4.1}$$

$$\sum_{n=1}^{N+1} d_n \quad \leq \quad W \tag{4.2}$$

$$\forall n \in \{1 \ldots N+1\}: \quad d_n \quad \geq \quad d_{\min} \tag{4.3}$$

$$\mathbf{d} \quad \in \quad m \cdot \mathbb{N}^{N+1} \tag{4.4}$$

Chapter 2.2 introduced all required variables. Note that the correlation factors c_i of a wire i are indexed with respect to its left and right neighbour.

The distances between a wire i to its left and right neighbour are d_i and d_{i+1}, respectively. These distances may sum up to at most the total whitespace W, see Formula (4.2). Furthermore, the wire positions and thus the distances, have to be integer multiples of a manufacturing grid m, see Constraint (4.4)

Due to the concavity of the ground capacitances $C_{\text{gnd}}(d_i)$, problem (4.1–4.4) is not a strictly convex optimisation problem in the sense that the objective is convex. If the case "$<$" in (4.2) is allowed – which makes sense since there is no reason not to save area – then a convex solution strategy can be applied. Alternatively, it is possible to reduce the problem to a geometric programming problem with proper expressions for the capacitance functions.

Both solution strategies have been pre-published in [ZBI+07]. Due to the rather theoretic aspect of bus spacing, see introduction of this chapter, the algorithms and results are not reproduced in this thesis. In brief, both solution strategies find the

[1] There may also exist crosstalk-related minimum distances.

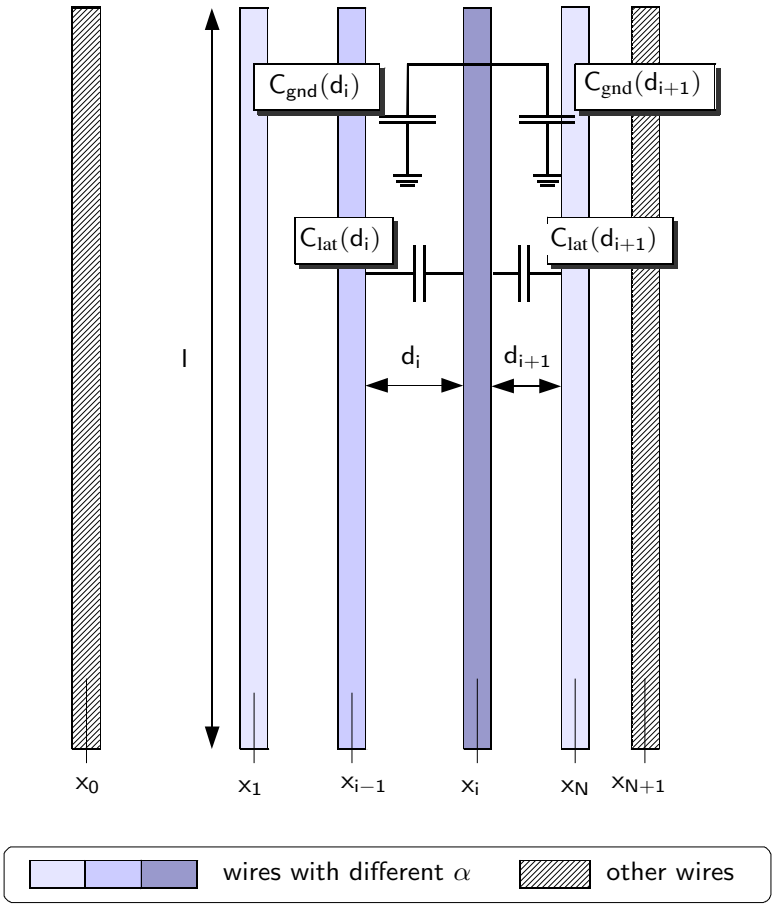


Figure 4.1: Bus wire spacing assumes a certain amount of invested area which is σ times larger than the minimum area required to comply with the design rules. It is distributed individually (non-uniformly) between the wires to balance the switched lateral and ground capacitances.

optimal distance vector. Compared to the state of the art, more than ten percent less wire power and several orders of magnitude less runtime are achieved.

Dropping the assumption of simultaneous transitions

In the following, the correlation between signals is neglected, which is equivalent to dropping the assumption of simultaneous transitions. That means that c_i are assumed to accept the values of $\alpha_{i-1} + \alpha_i$. A total per-unit-length capacitance function between two wires depending on their distance can be used:

$$C(d) := C_{\text{lat}}(d) + C_{\text{gnd}}(d),$$

since $C_{\text{lat}}(d)$ and $C_{\text{gnd}}(d)$ are now weighted with the same factor. Note that this factor is $\alpha_{i-1} + \alpha_i$ as the power caused by the common capacitance of the two wires is evaluated. Problem (4.1–4.4) reduces to (4.5–4.8):

$$\sum_{i=1}^{N} \alpha_i \big(C(d_i) + C(d_{i+1})\big) = \min! \tag{4.5}$$

$$\sum_{n=1}^{N+1} d_n \leq W \tag{4.6}$$

$$\forall n \in \{1 \ldots N+1\}: \quad d_n \geq d_{\min} \tag{4.7}$$

$$\mathbf{d} \in m \cdot \mathbb{N}^{N+1} \tag{4.8}$$

Approximating the crosstalk activities with the switching activities trades quality versus simulation effort. Unlike timing, however, power is an "average" value. Over a large number of coupling pairs, positive and negative errors cancel out each other [MKWS04] during analysis.

It is nevertheless interesting to evaluate the loss in optimisation potential and the accuracy of predicting power values. For this purpose, Section 6.3 is provided in the results-chapter. In brief, neglecting correlations during optimisation is safe in terms of maintaining optimisation potential and estimation accuracy with high probability.

Now, wire spacing can be done very quickly, but maintaining connectivity of the wire endings has not yet been investigated. This is done in the following subsection, the main building block for the overall method.

4.1.2 Non-uniform parallel wire spacing considering detours

Preliminaries

According to Figure 4.2, again a routing layer with vertical preferred routing direction is assumed. Most wires run in y-direction and wire spacing affects their x-position.

Parallel wires in a detail-routed layout can be re-spaced if no obstacles such as vias or perpendicular wires in x-direction prevent a lateral movement. Displacing only a fraction of a wire, termed "segment" from now, and maintaining the electrical connection, breaks up the parent wire into five shorter wires, cf. Figure 4.2

Assumed is a group of N such parallel wire segments (E.g. $N = 4$ in Figure 4.2) at given locations $(x_i^0, i \in \{1 \dots N\}) =: \mathbf{x^0} \in g \cdot \mathbb{N}^N$. Existing routers adhere to a routing grid g which is common practise in industry to maintain acceptable runtime and memory consumption during routing, cf. Chapter 2.

Further, x_0 and x_{N+1} are defined as the fixed positions of the two wires enclosing the system to the left and the right. These contain bends or vias, and thus cannot be moved. Notice that, as x_0 and x_{N+1} cannot be modified, they force an area constraint to the local topology. This is in contrast to the previously described problems where an extra amount of space was invested and expressed by W. Furthermore, the objects at the borders can belong to any circuit net and exhibit a switching activity that must be included.

Problem Formulation

While seeking the power-optimal position vector $(x_i^*, i \in \{1 \dots N\}) = \mathbf{x}^*$ for the vertical wire segments, the following integer-convex optimisation problem for the total switched capacitance is faced:

$$\sum_{i=1}^{N+1} (\alpha_{i-1} + \alpha_i) \cdot C(d_i) \cdot l + 2C_{\text{detour}} \cdot \sum_{i=1}^{N} \alpha_i \cdot |x_i^0 - x_i| \quad = \quad \min! \tag{4.9}$$

$$\forall i \in \{1 \dots N+1\}: \quad d_i := x_i - x_{i-1} - \frac{w_i + w_{i-1}}{2} \quad \geq \quad d_{\min} \tag{4.10}$$

$$\mathbf{x} \quad \in \quad m \cdot \mathbb{N}^N, \tag{4.11}$$

with the constants α_i (switching activity of wire i), w_i (width of wire i), l (wire segment length in y-direction), C_{detour} (per-unit-length capacitance of detour wire), $d_{\min}$ (minimum tolerable distance), and m (manufacturing grid). The constants

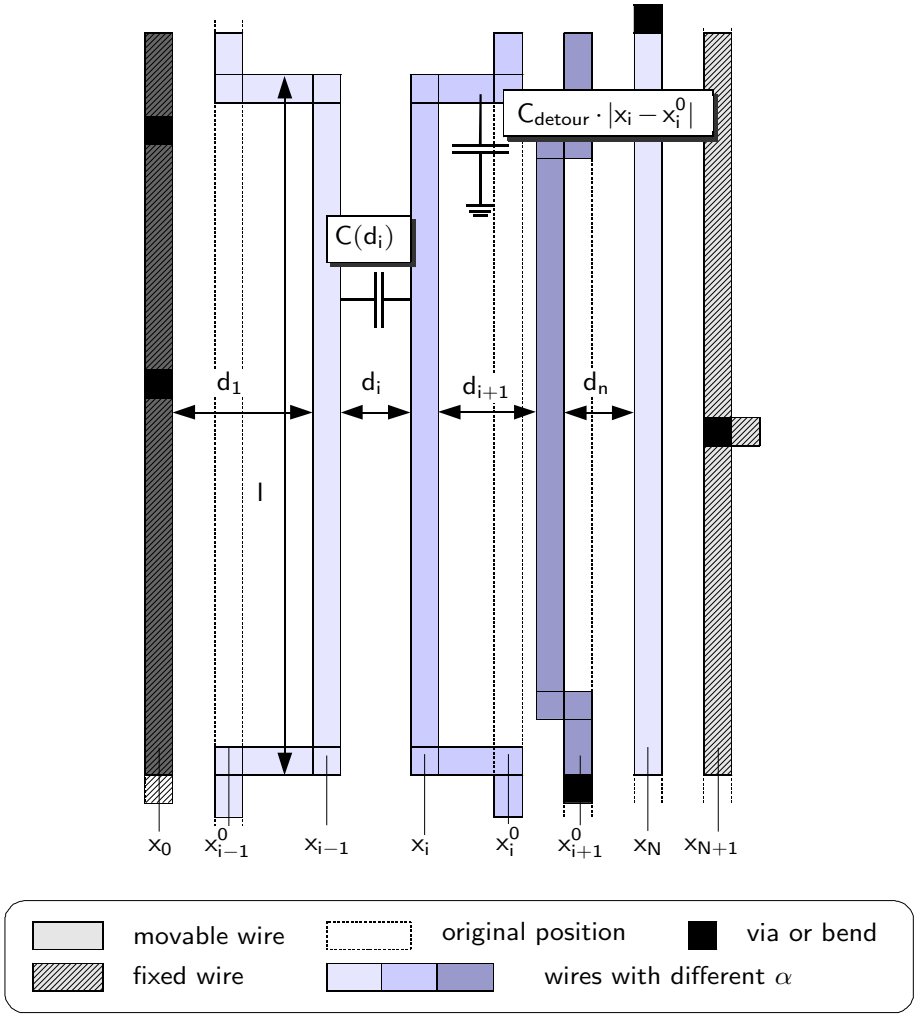


Figure 4.2: In-place power-driven wire spacing. The cost for detours, cf. horizontal stubs of length $|x_i - x_i^0|$ is considered. Resulting distances depend on the switching activities.

α_0 and α_{N+1} denote the effective switching activities of the wires enclosing the system. The constants f (frequency), and V_{DD} (supply voltage) are not displayed. According to common, lateral wire positions are defined by its centre. A vertical wire extends $w/2$ to the left and to the right of its x-position.

Problem Characterisation

One of the main differences of (4.9–4.10) to the previous problem (4.5–4.6) is that now the position vector $\mathbf{x}$ is the independent variable, and not the distance vector $\mathbf{d}$. This stems from the wire length of the detouring-stubs which influence the capacitance and cannot be expressed as a function of $\mathbf{d}$ conveniently. All capacitance functions are per-unit-length capacitances. As opposed to previously described problems, these have to be weighted with different lengths, l for the coupling between the wires, and $|x_i^0 - x_i|$ for the detouring-wires.

Multiplied by the respective switching activities, the first term in the new objective function (4.9) measures the switched capacitances between wires i and $i-1$, and the second term reflects the contribution of the detouring-wires of length $|x_i^0 - x_i|$.

Again, minimum distance (4.10) and integrality (4.11) constraints ensure compliance with technology design rules.

All functions in (4.9) are convex, and the problem is linearly constrained by (4.10). Any local minimum of (4.9–4.10) in $\mathbf{x}$ is thus its global minimum. The integrality constraint (4.11) renders this problem integer-convex, a problem class that is more difficult to solve than (linearly constrained) convex programming problems in general. Furthermore, the objective contains non-differentiable functions. Building the negative power gradient $-\nabla P(\mathbf{x})$ as required by some convex programming approaches is not possible, aside from the fact that following gradients generally violates integrality constraints such as (4.11). Instead, a local search of the discrete solution space is proposed as follows.

Approach

The minimum distance constraints and integrality constraints are never violated when displacing wires by small, discrete step sizes that are integer multiples of the manufacturing grid m.

Finding the wire with the currently best power derivative, and moving that wire at a time approaches the minimum – since there is only one minimum[2] –

[2]More precisely, due to the integrality constraint, there are at most $2N$, but these must have the

without ever violating any of the constraints.

It remains to show that all possible grid points (solutions) are accessible. The minimum is reached unless the only way descending to the minimum leads across an invalid-distance solution. In that and only in that case, the movement of a whole subgroup of minimum-spaced wires by the step size must be considered as well. Each distance constraint (4.10) combines the values of at most two independent variables. That means that a single-wire move could violate only one constraint—when it gets to close to an adjacent neighbour. To avoid this, that neighbour is moved as well, possibly pushing its own neighbour in that direction as well, and so on. This is shown in Figure 4.3 for two wires. More complex move patterns (such as moving some wire i to the left and simultaneously some other wire j to the right, or moving more wires at a time) are not considered. Most of them are not needed to reach all possible grid points because all such moves could be sequenced by single wire moves and by single wire-group moves. For more details, the reader is referred to [ZBIS07].

Algorithm 1 is proposed to solve (4.9–4.11) quasi-exactly and quickly.

Algorithm 1 Power-driven wire spacing considering detours.

01: For ($\Delta x := a \cdot m$; $\Delta x \geq m$; $\Delta x /= 2$)

02: evaluate ΔP of every possible move and build heap

03: while (ΔP_{best} at top of heap < 0)

04: commit best move at top of heap

05: evaluate ΔP for new moves and update heap

06: end while 03

07: end for 01

In order to converge quickly, a relatively large step size Δx of a times the manufacturing grid m (line 01) is used. The value should be a power of two and can be selected empirically through runtime measurements.

A move (lines 02 and 04 in the algorithm) is defined in this context as a lateral displacement of a wire or wire group by $\pm\Delta x$ as described above. If the wire cannot move, because the resulting distance to the neighbour wire would be

same objective value and must be located at adjacent grid points.

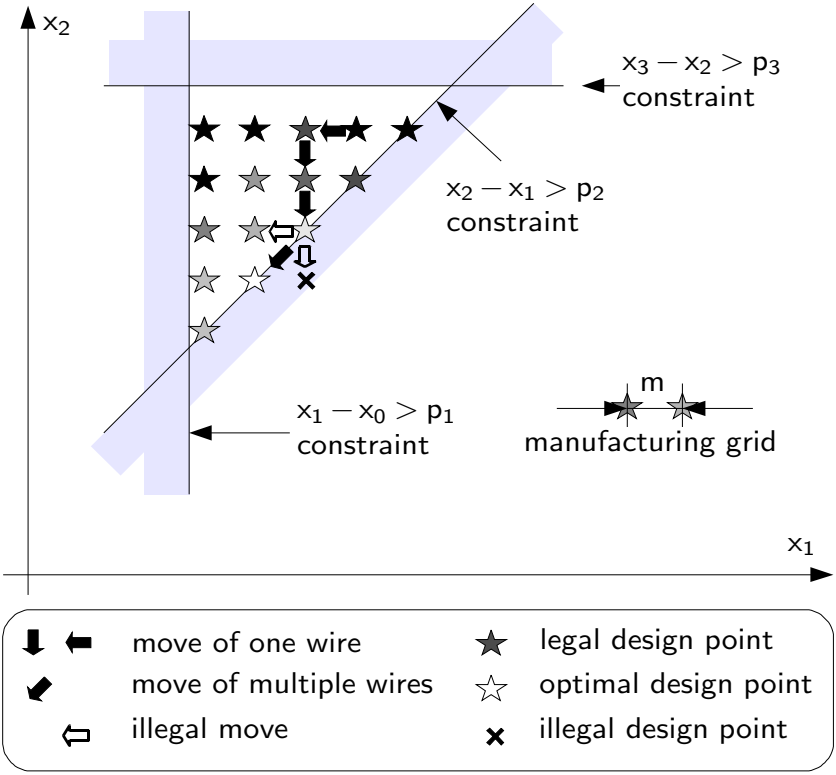


Figure 4.3: Illustration of how the discrete solution space is searched. The example shows a two-dimensional problem, i.e. two wires can be moved. Their coordinates span the solution plane (x_1, x_2). The wires at x_0 and x_3 are fixed. No two wires must get closer than a minimum distance, this results in the three pitch constraints shown (pitch is distance plus width). Stars indicate valid positions on integer multiples of the manufacturing grid. Dark stars indicate bad solutions, i.e. wire constellations that cause high power consumption. To reach the optimal solution, multiple (two) wires may have to be moved into the same direction simultaneously.

smaller than $d_{\min}$, then also that neighbour is moved. This mechanism extends to any number of wires in a group of wires of distances smaller than $|\Delta x| + d_{\min}$. If one of the fixed wires at x_0 and x_N prevents the movement of the outermost wire, then the move is not possible.

Such a move will change the distances (4.10) and the detour lengths $|x_i^0 - x_i|$ of the involved wires and thus the capacitances between the vertical wires $l \cdot C(d_i)$, and the horizontal stub capacitances $C_{\text{detour}} \cdot |x_i^0 - x_i|$, respectively. Multiplied by the corresponding activity factors, V_{DD} and f, this leads to a power difference ΔP inside the box before and after the particular move, evaluated in lines 02 and 05.

Only the best of all (at most $2N$) currently possible moves (line 04) is committed. Storing the moves in a heap keeps the runtimes low for larger values of N because the best of all moves is always at the top of the array and can be retrieved in constant time. Updating the heap can be done in $O(\log N)$ time and is required after every move for those wire(s) that have been moved and their neighbours or neighbouring groups of minimum-spaced wires.

A non-negative value of ΔP_{best} at the top of the heap indicates that there are no more moves that save power for the current step size (ll. 03,06). The wires are then moved with successively halved step sizes until Δx falls below the manufacturing grid m (line 01).

The for-loop is independent of N. It is assumed that the total space, $x_{N+1} - x_N$, is proportional to N so that the while-loop is run through with at most $O(N)$ complexity. In the worst case, the spacing method requires $O(N^2 \cdot \log N)$ steps. This is the case if the heap needs to get updated for all N wires in line 05. That, however, only occurs if every move is a group-move of all wires. The average case is much less complex. Since this spacing routine is called with relatively small numbers of N, such as typically less than twenty, but a high number of times, cf. next chapter, it is more important to optimise the actual execution time than the asymptotic complexity. This can be done by minimising the number of memory accesses and by using fixed point arithmetic. Again, more details on the algorithm, its convergence and quasi-exactness can be found in [ZBIS07].

Post processing

After spacing, the optimised and initial wire endings need to be connected at both sides of the wires. This is an instance of the river routing problem [She93] as these connections are to be laid out without having to change the layer with costly vias.

There exist very specialised and very general problem variants with different goals and side constraints of single layer routing and respective solution meth-

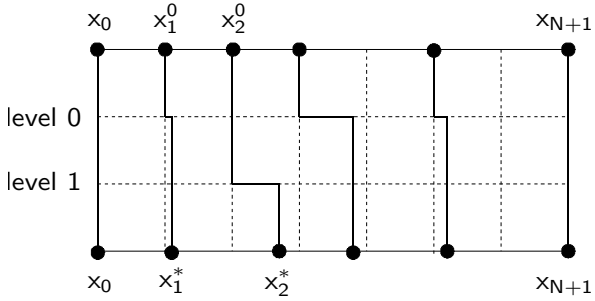


Figure 4.4: Simple river routing problem appearing after wire spacing. The upper and lower row represent the old and new wire positions, respectively. The task is to find legal connections between the respective positions on the same layer without touching or crossing other wires.

ods [She93]. The problem appearing in this particular context is easily described with the help of Figure 4.4.

One property of the problem follows directly from the allowed solution space for initial and final wire positions. Initial wire positions are commonly bound to a routing grid while the final positions are bound to a manufacturing grid. Another property of the problem regards the boundary condition: the problem is area-constrained by the objects at positions x_0 and x_{N+1}.

The simplest form of river routing requires two wire bends for every wire, one bend for turning into the perpendicular connecting stub, and one turn back to the parallel wire piece at the new position. That also produces the minimum possible number of bends.

The y-position at which turning occurs is an integer multiple of the routing pitch, in the following referred to as "level". The level depends on the old and new positions of the neighbouring wires. Wires that can turn to run perpendicular immediately (because no neighbouring wires are closer to the new position plus the minimum distance) are associated with a level of zero. Wires that cannot turn because another wire bend at level h is blocking the way, are associated with a level of $h+1$, and so on. A simple river router was implemented that minimises the maximal level and requires minimal number of bends, cf. Algorithm 2.

A flag for every wire indicates if a level has been determined for it yet or not (level of ∞). All wires that did not move are excluded from consideration by assigning a level of zero (lines 01-07).

Algorithm 2 River routing for wire endings.

01: forall wires i *// process unmoved wires*

02: if $(x_i^* == x_i^0)$

03: $h_i := 0$

04: else

05: $h_i := \infty$

06: end if

07: end forall

08: actlev := 0

09: while (∃ some wires not processed) *// process moved wires*

10: forall wires i that are not yet processed

11: if $(x_i^* > x_i^0)$ and $\big((x_i^* \leq x_{i+1}^0 - d_{\min} - \frac{w_i + w_{i+1}}{2})$ or $h_{i+1} <$ actlev $\big)$

12: h_i := actlev *// moved right*

13: else if $(x_i^* < x_i^0)$ and $\big((x_i^* \geq x_{i-1}^0 - d_{\min} - \frac{w_{i-1} + w_i}{2})$ or $h_{i-1} <$ actlev $\big)$

14: h_i := actlev *// moved left*

15: end if

16: end forall 10

17: actlev ++

18: end while 09

A loop starts (line 09) with a current level zero (line 08) and tries to apply the current level to those wires which have not been processed yet. If wires cannot be processed at the current level, that value is successively increased by one (line 17) until all wires are processed. A wire i qualifies for turning at the current level if the neighbouring wire is far enough away or has already turned (line 11 if wire i moved to the right and line 13 if wire i moved to left).

As a result of river routing, the length l of a wire coupling at the optimised distance in Objective (4.9) is actually not a constant, see for instance wires $i-1$ and i (level 0) in comparison to wire $i+1$ (level 1) in Figure 4.2. For simplicity, this influence was neglected during optimisation. Once the river routing results are known, however, the power saving of the spacing result can be adjusted accordingly.

River routing has to be run once for every set of parallel wires. Its results are applied twice, at both wire ends and mirrored. The maximal reduction of l is therefore $2 \cdot (N-1) \cdot (w + d_{\min})$ if all wires need to be routed at a new level. Another simplification regards the maximal level. If the length reduces to below a predefined threshold due to the maximal turning level, such as five track widths, then the entire spacing solution is discarded and spacing for the current instance considered infeasible, although a river routing solution may exist that reduces the maximum level for the price of an increased number of bends.

This spacing technique including river routing achieves power saving results comparable to previously published techniques for buses [MMP02], up to 49 %[3] in some circuit regions. At the same time, it runs orders of magnitudes faster, and it routes connections between initial and new wire endings, cf. [ZBI+07, ZBIS07].

4.2 Parallel wire permutation for low power CMOS

The flow of the overall proposed optimisation method is continued in Chapter 5. In the present section, wire permutation for low power, formally to be defined below, is investigated. Permuting the wires prior to the previously described wire spacing affects the efficiency of wire spacing. Unfortunately, a practical application has not been found yet.

A deeper study on the problem showed that a power-optimal wire ordering exists that maximises the effect of wire spacing and thereby only depends on the switching activities of the wires, confer Figure 4.5.

The following assumptions are made in this section:

[3] That value depends on the capacitance model used, cf. Figure 7.1 on page 99.

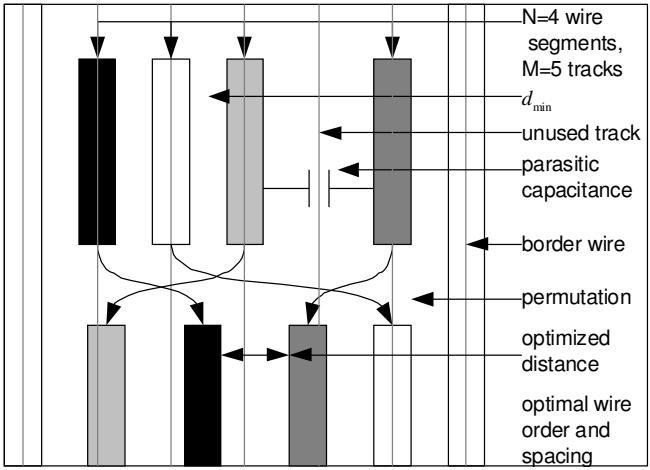


Figure 4.5: Wire ordering for low power CMOS. The order of the switching activities (shaded) influences the effect of wire spacing.

1. No simultaneous switching is present. This is the main difference to wire ordering techniques in the literature, cf. Section 3.2.2, which aim to minimise the number of wire pairs that switch simultaneously or their correlation coefficient.

2. Wire endings are physically dangling. No detour wires or permutation overhead is included.

3. Wires to be ordered are enclosed by zero-activity wires such as power or ground lines, i.e. $\alpha_0 = \alpha_{N+1} = 0$.

4. The best way how to achieve this order physically during or after routing is unclear, although the outlook in Chapter 7 provides some ideas.

The following considerations apply to the wire spacing problem (4.5–4.8).

The power-optimal wire order

Definition 1 *Given is a set* $(\alpha_0, \alpha_1, \ldots, \alpha_N, \alpha_{N+1})$ *of* α*-factors of the* N *nets to be routed. For a permutation* π *of the numbers* $\{1, 2, \ldots, N\}$,

$$(\alpha_0, \alpha_{\pi(1)}, \ldots, \alpha_{\pi(N)}, \alpha_{N+1})$$

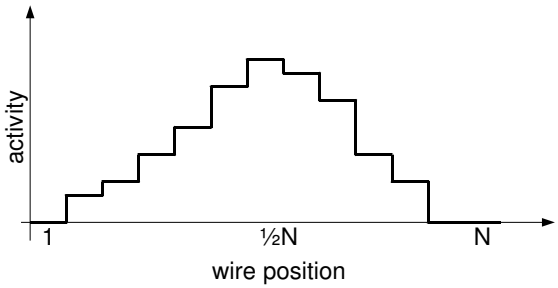


Figure 4.6: Illustration of the optimal wire order for low power CMOS. In order to increase the effect of activity-driven wire spacing, high-activity wires should be placed next to each other.

is called a wire ordering[4]*. The set of all conceivable wire orderings is denoted by* $\mathbb{A}$*. A particular wire ordering is called a power-optimal wire ordering if the solution for (4.5–4.8) is minimal over all wire orderings in* $\mathbb{A}$*.*

An investigation of the problem with different wire orderings revealed the following observation, cf. Figure 4.6.

Theorem 1 *A wire ordering* $(q_n) \in \mathbb{A}$ *is a power-optimal wire ordering if it is constructed by Algorithm 3:*

The proof of this theorem is the result of the cooperation with Peter Gritzmann and Michael Ritter, *Institute for Combinatorial Geometry, Centre for Mathematical Sciences, Technische Universität München*, and is provided in Appendix A.

Without proof, also the existence of a worst wire order is proposed. The worst wire order is used to find out the potential of wire ordering. The observation of this section, an early version of the proof, and experimental results have been prepublished in [ZGRS05]. An excerpt of the results is reproduced in the Experimental Results, Section 6.4.

4.3 Conclusions

An activity-driven wire spacing technique has been developed. It differs from previously published techniques in objective and implementation. While earlier

[4]Note that this definition is different from Section 3.2.2 to better serve the needs of this section.

Algorithm 3 The optimal wire order for low power CMOS.

01: Start with $q_0^{(0)} := q_{N+1}^{(0)} := 0$, $K^{(0)} := \bigcup_{i=1}^{N} \{\alpha_i\}$.

02: For $s = 1, \ldots, N$:

03: Let $q_a^{(s-1)} \leq q_{a+1}^{(s-1)}$ be the two greatest elements of $(q_i^{(s-1)})$
(Theorem 4 of Appendix A will prove that these have to be adjacent).

04: Let $c := \min K^{(s-1)}$ and define $K^{(s)} := K^{(s-1)} \setminus \{c\}$.

05: Define $(q_i^{(s)})$ by inserting c between $q_a^{(s-1)}$ and $q_{a+1}^{(s-1)}$, i.e.

$$q_i^{(s)} := \begin{cases} q_i^{(s-1)}, & \text{for } i \leq a \\ c, & \text{for } i = a+1 \\ q_{i-1}^{(s-1)}, & \text{for } i \geq a+2. \end{cases}$$

06: end for 02

works developed techniques that allow to route optimised buses with dangling wire endings, the new technique allows for an in-place modification of already-routed wires. For this purpose, correlation considerations are dropped, and integer-convex programming is used to provide an exact solution in very low runtime. It includes a mechanism to connect the old and new wire endings with the help of a river router.

An observation on the theoretically optimal wire order was made when carrying out the research for this thesis, and as these observations also apply to parallel wires, they have been provided in the second section of this chapter. Its results show promising increases of the effect of wire spacing but are rather theoretical as a practical application method has yet to be found and validated. Suggestions on how to utilise this effect are noticed in the Outlook.

5 Systematic optimisation of entire detail-routed layouts

Using the results of the previous chapter, a methodology is developed to reduce the power consumption in detail-routed layouts. The general idea is to exploit that wires on the same layer are routed preferably in parallel, see also Chapter 2.

This method takes three major steps. Finding all locations in a given detail-routed circuit where wire spacing of Section 4.1 can be applied is the first step and described in Section 5.1. Second, the application of wire spacing itself is recapitulated in the current context in Section 5.2. Section 5.3 provides the third step, selecting the best wire spacing instances to be considered in the final layout.

Both the methodology and its implementation as tool are referred to as *TopCool*. The name stems from wire *top*ology and *cool*er chips through power reduction.

5.1 Searching parallel wires

The goal can be described by finding suited and lucrative locations in the design where parallel wire spacing can be applied. In the present suggestion, all local modifications made in the layout affect the current layer only. This greatly simplifies the problem. Moving vias with the wires would require to include a possible change in wire lengths on adjacent layers. Also perpendicular wires would require special treatment. Additional terms in the objective function as well as more constraints could render the problem intractable.

The present method thus requires that vias and wire endings are left at their fixed positions, although the Outlook 7.2.8 presents ideas for moving the vias as well. Since any wire in a design ends in a via or a bend and both prevent a lateral displacement of an entire wire, the following problem is formulated.

5.1.1 Problem

> *Find rectangular regions in a given layout that are enclosed by, but do not contain wire endings.*

This problem description captures four basic conditions:

1. The regions should be as large as possible,
2. the regions can only contain parallel wires,
3. but must not contain wire endings, and
4. all such regions must be investigated.

The more wires that run in parallel, the more local whitespace can be expected, and thus more freedom exists during its re-distribution among wires of different activity. The longer the parallel wires on the other hand, the less the impact of the overhead of detouring. Consequently, the regions should be as large as possible. A region that is not enclosed by a wire ending on either side could be enlarged until it is enclosed by a wire ending (or the chip boundary) on that side, and thus save more power.

Regions fulfilling this boundary condition can only contain parallel wires or nothing. In the former case, instances of the wire spacing problem of Section 4.1.2 are created and solved. In the latter case, the region is discarded.

Since the regions are subject to a later spacing process, they must not contain wire endings which cannot be moved with the current wire spacing concept.

A single wire can generally be contained in more than one such region. Since it is unknown a priori from which of multiple overlapping regions the power reduces most, the strategy is to find and investigate all regions and then to select the best subset.

Regions are referred to as “rectangular areas” or “boxes” in the sequel.

5.1.2 Approach

The suggested method operates layer by layer. In a routed layout, wires preferably run in parallel within one layer. A layer is scanned in the preferred direction. Whenever a via or wire ending is encountered, it is considered as delimiter of a rectangular area.

Without loss of generality, a layer with horizontal preferred routing direction is assumed, i.e. the wires run in x-direction. For vertical layers, all values of x and y have to be interchanged.

Given is a non-redundant list V of all vias and wire bends, sorted in ascending x-direction. For any $i \in V$, Algorithm 4 on page 58 builds a list L_i of all boxes that are enclosed by i to the left and any further elements in V to the right, top

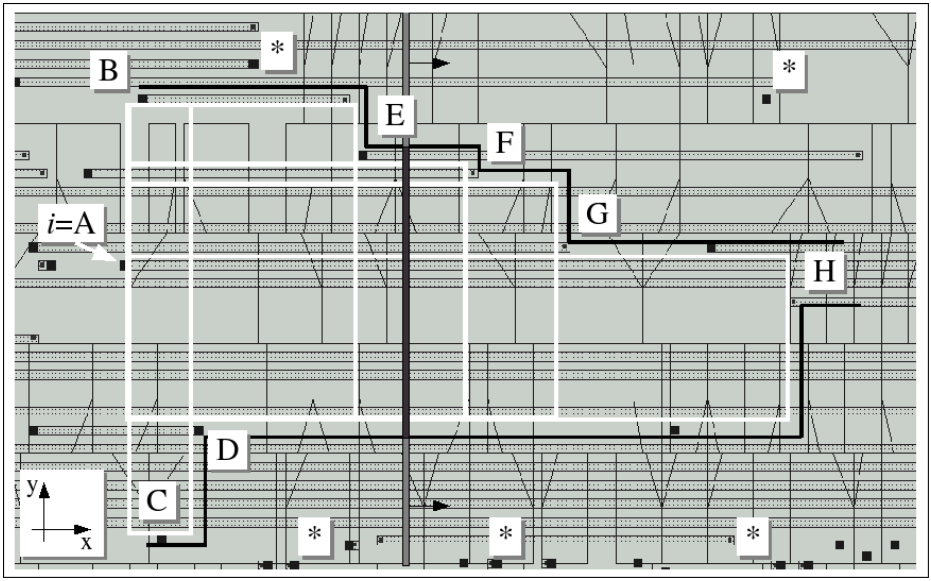


Figure 5.1: Layout photo of a horizontal routing layer to illustrate the scanline algorithm. It finds all boxes (white) that group parallel wire segments that are subject to a later spacing process. As the algorithm continues, the search corridor (black) is narrowed. Vias outside the corridor (*) do not need to be considered.

and bottom. Figure 5.1 on page 57 shows a layout photo of a routing layer in x-direction. See A in that figure for an instance of i.

The y-locations of the next two elements in V located above and below i define a search corridor. Note that the chip boundaries can serve as these two elements as implemented in line 01 of Algorithm 4. For illustration reasons, $(C.y - B.y)$ is used as initial corridor in Figure 5.1, and not the chip boundaries.[1]

An imaginary scanline is traversed in x-direction starting from i, cf. the grey vertical line in the figure. In the algorithm, this corresponds to processing the elements of V in sorted x-direction (lines 05–06), starting at i (line 03). Whenever the scanline encounters a further element $j \in V$ (eg. D…H) that is within the

[1] [ZBIS07] optimises the same algorithm with limited starting corridor height, such as twenty tracks. The principle is the same but that algorithm is faster. Larger boxes are seldom and thus do not contribute to the power saving significantly.

Algorithm 4 Principle of searching all rectangles with common left delimiter i.

```
01: cor.low := chip.min_y,    cor.high := chip.max_y
02: L_i := {}    // the list of boxes with i as left delimiter
03: j := i
04: while ( cor.high − cor.low > one wire pitch )
05:     // take next j within corridor:
06:     j := min_x{o ∈ V : o.x > j.x ∧ cor.low < o.y < cor.high}
07:     new_box.(left, right, top, bottom) := ( i.x, j.x, cor.high, cor.low )
08:     L_i.push_back ( new_box )
09:     if ( j.y > i.y ) then
10:         cor.high := j.y
11:     else
12:         cor.low := j.y
13:     end if 09
14: end while 04
```

corridor, a new box is found and added to L_i. The new box is limited by i to the left, j to the right and by the current corridor to the top and bottom. In the algorithm, line 07 sets the dimensions of the new box found, and line 08 stores it in the list L_i.

An update of the corridor boundaries takes place immediately afterwards: $j.y$ is assigned to the upper corridor coordinate if greater than $i.y$ (eg. E, F, G; lines 09–10) or to the lower border coordinate otherwise (eg. D, H; lines 11–12). This excludes vias and wires that are outside the y range of the corridor. These cannot build valid rectangles with i anymore—such rectangles would contain j and violate the third condition on page 56. As a consequence, the corridor narrows down throughout the execution of the algorithm.

The scanline continues to traverse until the corridor width $(\text{cor.high} - \text{cor.low})$

falls below a predefined threshold, such as one routing track pitch (line 04). In that case, only one wire is left for wire spacing which is still a valid case[2] as the single wire contained in such boxes can be placed in optimal distance to the box delimiters.

Every box can only contain parallel wire segments and no vias or perpendicular wires. The algorithm is exact in that all such boxes with i to the left are found on a metal layer.

Algorithm 4 should be considered illustrative because of technical details such as doubly detected or empty boxes.More important is the runtime complexity. The application of Algorithm 4 to all instances $i \in V$ to detect all boxes in the circuit can result in impractical quadratic runtime complexity if the range query in line 06 is not carefully implemented. Therefore, the following modification is proposed to achieve efficient quasilinear runtime.

The scanline needs to traverse the layer only once as it is possible to keep track of multiple active corridors simultaneously, see Figure 5.2 for an example. All active corridors $\{ \mathrm{cor}_1, \mathrm{cor}_2, \mathrm{cor}_k, \ldots, \mathrm{cor}_q \}$ extend from some element $\mathrm{cor}_k.i \in V$ in the left to the scanline in the right.

Any element $j \in V$ hit by the scanline not only triggers a stabbing query for all active corridors at a time, but also serves as a new starting point that defines a new left border.

The stabbing query is equivalent to the question: which out of a given set of intervals are stabbed by a certain point? In the given case, which of the q currently existing corridors contain j? All corridors that contain j narrow down as described above—simultaneously. The stabbing process is illustrated in Figure 5.3.

At the same time, a new corridor is generated with j as left border. The following section shows that this actually guarantees quasilinear complexity.

5.1.3 Complexity

Notice that the number of boxes $|L_i|$ per start point i depends on the via density but is largely independent of the circuit size, expressed as $|V|$. A proof has been sketched in [Iln06]. In brief, the number of boxes local to i is not modified if circuit area is added and $|V|$ increased. By symmetry, the same is true for the number of boxes per end point j. This is easy to see if the circuit is mirrored, that is, x is replaced by $-x$. The independence of the number of boxes per start point or end point is an important result.

[2] It is even a very common case as pointed out in [ZBIS07].

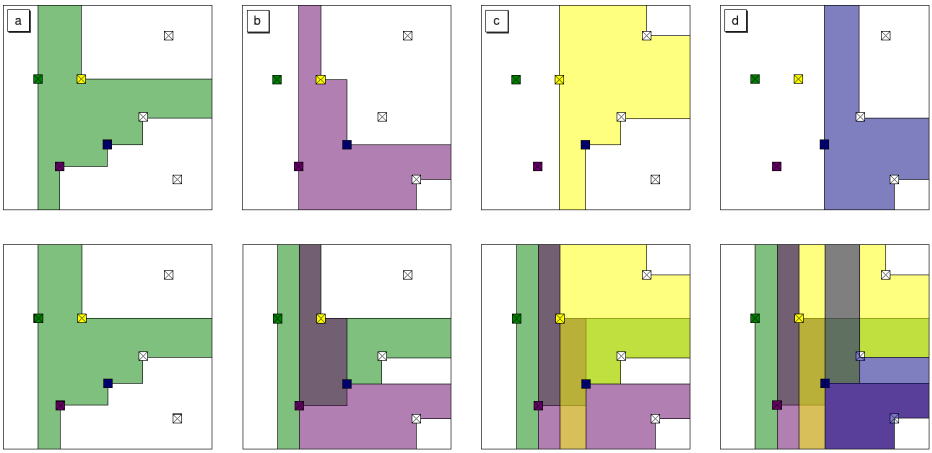


Figure 5.2: Top row: Corridor shapes of the left four vias when the scanline has reached the right box border. Bottom row: Keeping track of multiple active corridors at the same time. The number of intersections at a point, i.e. the number of different colours in any vertical intersecting line depends at most linearly on the number of corridors.

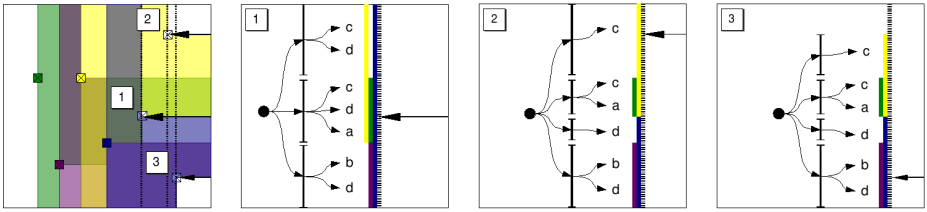


Figure 5.3: Illustration of stabbing queries. The answer to a stabbing query is a report of which of the active corridors contain (are stabbed by) a point. The first query delivers the three corridors {a,c,d}, the second {c} and the third {b,d}. The corridor names correspond to Figure 5.2. Dotted vertical is the scanline.

The y-dimensions of a corridor at a certain x-point are treated as an y-interval in the following. The implementation uses a segment tree. This binary tree is said to "live" on the scanline. Its leaves correspond to the elementary intervals defined by the vertical intersections of the corridor intervals. Trees can be abstracted by more intuitive ordered sets (which in turn can be implemented with trees). Each elementary interval in the set contains a set of corridor identifiers of those corridors which have contributed to the very elementary interval.

Stabbing is known to report which of the corridors contain j in $O(\log M + R)$ runtime complexity [SDK96]. This assumes M intersections of currently active intervals, R of which are stabbed by j. R is actually the number of boxes for the end point j since a new box is returned for every corridor that contains j. This number does not grow with respect to $|V|$.

There exists one corridor for every element in V, the theoretic maximum for the number of active corridors q. $M/2 = q$ intervals can have at most M intersections (elementary intervals). The worst case dependence of M on $|V|$ is therefore linear. The practical dependence is much weaker, though, since the number of currently active intervals $q \ll |V|$ is limited to the boxes about the scanline. In any case, stabbing with one via or wire ending is a function of at most logarithmic runtime complexity in $|V|$.

The other operations required, namely adding and removing intervals from the tree (which corresponds to updating a corridor), are logarithmic in the tree size M as well [SDK96]. Stabbing is executed $|V|$ times. Adding and removing a corridor is executed $|L|$ times, where L is the total number of boxes in the design, $L := \cup_{i \in V} L_i$. $|L_i|$ is independent of $|V|$ so that $O(|L|) = O(|V|)$. Thus, the entire proposed scanline algorithm exhibits

$$O(N \cdot \log N), N := |V|$$

runtime complexity. [ZBIS07] presents a detailed implementation of an alternative approach using the interval tree concept [Ede83a] and the suggestions of [SDK96]. As opposed to the segment tree, the interval tree (skeleton) is static, and this seems to be more runtime-efficient.

5.2 Re-spacing parallel wires

In the second step, activity-driven wire spacing as developed in detail in Section 4.1 is applied to all instances of groups of parallel wires found.

Note that Section 4.1 modelled the N parallel wires to be enclosed by fixed wires that run in parallel at the sides of the region. In the given case, this is not necessarily true. The N parallel wires can be limited by vias, perpendicular wire endings, or parallel wires that do not cross the full length of the rectangular box and thus may not couple with the wire segments inside the box along the full length. To accommodate for that fact, effective switching activities α_0 and α_{N+1} are derived from the actual activities and the fraction of the coupling lengths. If the box border along the preferred routing direction is not fully shielded, then coupling with adjacent wires outside the box is ignored. Section 7.2.5 briefly covers a resulting issue.

Important to note is that the wire spacing and river routing results and in particular the possible power reductions $\Delta P_{\text{box},i}$ are recorded for every box $i \in L$, but no modification to the layout is performed at this point. Boxes with very low relative power consumption $\Delta P_{\text{box}}/P_{\text{box}}$ are discarded.

5.3 Selecting the best groups of parallel wires

Since wire segments can be contained in multiple boxes in L, that is, the boxes intersect, cf. Figure 5.1, a selection procedure for the boxes is required. The same segment of a wire can only be re-spaced within one context. If the set of boxes L is mapped to a graph $\mathscr{L}$, the problem can be reduced to a maximum weight independent set problem (MWIS) [NO03].

The nodes of the graph correspond to the boxes containing the optimised wire topologies. Each node carries a weight which represents the possible absolute power saving ΔP_{box} as recorded in Section 5.2. An edge between two nodes means that the two boxes overlap. At most one of two overlapping boxes can be selected for optimisation because each box entails a different wire topology optimal only for the region covered by the box.

WWIS seeks an independent set I of nodes, i.e. the set of nodes of a subgraph of $\mathscr{L}$ without edges while maximising the sum of the node weights in I. In the given instance, a non-intersecting subset of boxes with maximal total power saving

$$\Sigma_{i \in I} \Delta P_{\text{box},i} = \max!$$

is required.

A complementary problem to MWIS is the "maximum weight clique (MWC)" problem. The goal is to find the clique (subset of nodes where every node pair shares an edge) with the highest total node weight. It can be generated from MWIS

by replacing every edge in the graph by a non-edge and every non-edge by an edge.

5.3.1 Connection graph

Building the connection graph means mapping rectangles to nodes and connecting two nodes with an edge if the two corresponding rectangles overlap.

Determining which out of a given set of n iso-oriented rectangles in a two-dimensional plane physically intersect is well studied. It can be implemented with a scanline algorithm and the use of stabbing (interval query) and reverse stabbing (range query), cf. [SDK96, Ede83a, Ede83b] or almost any textbook on geometric algorithms. Then, a runtime complexity of at most

$$O(n \cdot \log n + k),$$

where n is the number rectangles, is achieved. The number of overlaps k is at most $1/2n^2$ so that the worst-case runtime for building the connection graph becomes $O(n^2)$ in general.

In the given case, however, k depends linearly[3] on N. This is easy to reason. The same proof as given before for the number of boxes as function of $|V|$ can be used [Iln06]: increasing the existing number $|V|$ by adding more circuit area containing wires and vias does not change the number of rectangles and overlaps in the existing circuit area if constant wire and via density is assumed. Therefore, also the sub-problem of building the connection graph is quasilinear, that is

$$O(N \cdot \log N).$$

5.3.2 Selection heuristic

Wwis itself, however, is known to be $\mathcal{NP}$-hard [GJ79] and, in the general case, even impossible to approximate to a constant factor in polynomial time [PY91]. The cardinality of the set of boxes $|L|$ prohibits exhaustive search and heuristics with high polynomial runtime complexities.

Three standard methods were applied on the global problem, two exact methods, and a heuristic. All of them approved that the problem can not be solved practically as a whole.

[3] Notice that $O(n) = O(N)$ since $O(|L|) = O(|V|)$.

MWC-solver [NO03] uses a branch-and-bound algorithm to solve the maximum weight clique (MWC) problem exactly. Therefore, its worst-case runtime complexity can be expected to become exponential. Empirically, while problems with graphs with up to some thirty nodes could be solved virtually instantaneously, eighty to one hundred nodes could already cause runtimes in the order of hours. The author of this software, "Cliquer" is Patric Östergård of *Teknillinen Korkeakoulu (Helsinki University of Technology)*[4].

ILP-solver An integer linear programming solver rather than an explicit MWIS solver can be used since MWIS can be mapped to the following integer linear program in $\mathbf{u}$:

$$\Sigma_{i\in L}\Delta P_{\mathrm{box},i}\cdot u_i = \max! \tag{5.1}$$

$$\forall i\in L: \quad u_i \in \{0,1\} \tag{5.2}$$

$$\forall \text{ edges } \{j,k\}: \quad u_j+u_k \leq 1 \tag{5.3}$$

and passed to an appropriate solver. In (5.1), $\Delta P_{\mathrm{box},i}$ denotes the power saving of the box i. A box i is selected if the independent variable u_i, that is forced to a binary value (5.2), accepts one, and not selected otherwise. Neighbouring box pairs j and k are excluded from the solution since their respective binary values u_j and u_k may sum up to at most one, cf. (5.3).

Of course this method can not decrease the problem complexity itself. Experiments with "lp_solve" by Michel Berkelaar[5], however, showed that this method exhibits a much less dramatic runtime explosion than Cliquer from problem sizes larger than 50 on. Unfortunately, some exceptions at smaller problem sizes occurred where lp_solve took much longer than Cliquer.

Qualex [Bus06] is a heuristic (based on quadratic programming) and was published by Stanislav Busygin of *University of Florida*.[6] The reported runtime complexity is $O(N^3)$. Tests showed that this cannot be tolerated for problem sizes in the order of thousands, let alone orders of many millions.

Thus, the problem demands to resort to an own heuristic. A property of the given graphs that could be used to reduce the problem to a simpler complexity class

[4] `http://users.tkk.fi/~pat/cliquer.html`, March 2007.
[5] `http://sourceforge.net/projects/lpsolve`, March 2007
[6] `http://www.busygin.dp.ua/npc.html`, March 2007.

directly follows from Section 5.3.1. The number of edges grows linearly with the number of nodes. This is low compared to the worst-case dependency of the edges on the nodes, $O(N^2)$. Thus, it can be assumed that relatively simple heuristics already perform well. Several heuristics were implemented and experimented with. Among those were:

Random Continuously pick a random rectangle r, and mark r and all its neighbours as taken until no more rectangles are left.

Greedy Continuously pick the currently best rectangle r and mark r and all its neighbours as taken until no more rectangles are left. "Best" is defined in this context as the rectangle with the highest power saving.

Greedy 2 Like the previous item but with a different definition of "best": The best rectangle is the one with the best power saving minus the power savings of its neighbours that have not been taken so far.

Block Cliquer This method solves the MWIS problem for the b highest nodes exactly, then for the next b nodes and so on. The intuition is to concentrate on the boxes with the highest savings. The name stems from the fact that Cliquer was preferred as exact solver because the linear programming method – sometimes – caused very high runtimes. Despite the extra effort to convert MWIS to MWC, this method was the method with the best worst-case execution time over all instances of problem sizes smaller b that occurred.

Experiments for problem sizes that can be verified exactly – less than 200 nodes – showed that all of the above result in at least 90% of the optimum with advantages for the methods increasing in the given order.

Block Cliquer proved efficient, in the order of 99% of what can be achieved by the exact solution for less than 200 nodes. It suffers only from the fact that b automatically decreases during the flow of the algorithm because the more rectangles that are taken, the more neighbours emerge that have been marked as overlapping. As a result, the exact solver operates on much less than b nodes in later phases. This is inefficient as the largest value of b must be adapted to the maximal tolerable runtime but is used only for the first instance. This problem was circumvented by the introduction of:

Block Cliquer 2 (Algorithm 5) which solves the MWIS problem for the b highest nodes exactly, then for the next b *remaining* nodes and so on.

The difference to Block Cliquer is that the exact solver now always operates on b nodes (except for the last run). See Figure 5.4 for an example. Let I denote the resulting independent set. Further, let $\mathscr{G}$ be defined as the subgraph of $\mathscr{L}$ with those nodes which correspond to the remaining boxes in L with the b highest savings or all remaining nodes if less than b are left. Any edge between $\mathscr{G}$ and nodes in $\mathscr{L}$ with less weight is voided.

Algorithm 5 High weight independent set heuristic.

```
01: I := {}
02: while (L not empty)
03:     G := subgraph ( L, b )
04:     MWIS(G) := solve MWIS for G exactly
05:     I ∪= MWIS(G)
06:     L\ = MWIS(G) ∪ neighbours of nodes in MWIS(G)
07: end while 02
```

MWIS($\mathscr{G}$) denotes the exact MWIS solution for the subgraph $\mathscr{G}$ (line 03). Cliquer [NO03] was used as the exact solver for this step, cf. line 04. Then, I is replaced by $I \cup \text{MWIS}(\mathscr{G})$ (line 05), and L is reduced by MWIS($\mathscr{G}$) and the neighbours of MWIS($\mathscr{G}$) in $\mathscr{L}$ (line 06).

If L is not empty, start again (line 07). When L is empty, I contains those non intersecting boxes whose modifications will actually be committed in the layout.

Complexity

The runtime can be balanced against the quality by selecting b. If set to one, Block Cliquer 2 is equivalent to the simple greedy heuristic described above, if set to $|L|$, an exact solver is obtained. Constant values between 10 and 50 are practical on modern-day computers and the given graph properties.

Regarding the number of boxes $|L|$, the loop adheres to linear runtime. Since $|L|$ grows linearly with $N := |V|$, the subset selection complexity is only theoretically determined by sorting the boxes by their power saving, that is $O(N \cdot \log N)$ asymptotic. The practical cost of sorting, however, is negligible compared to calling an MWIS solver on a problem size of b several times.

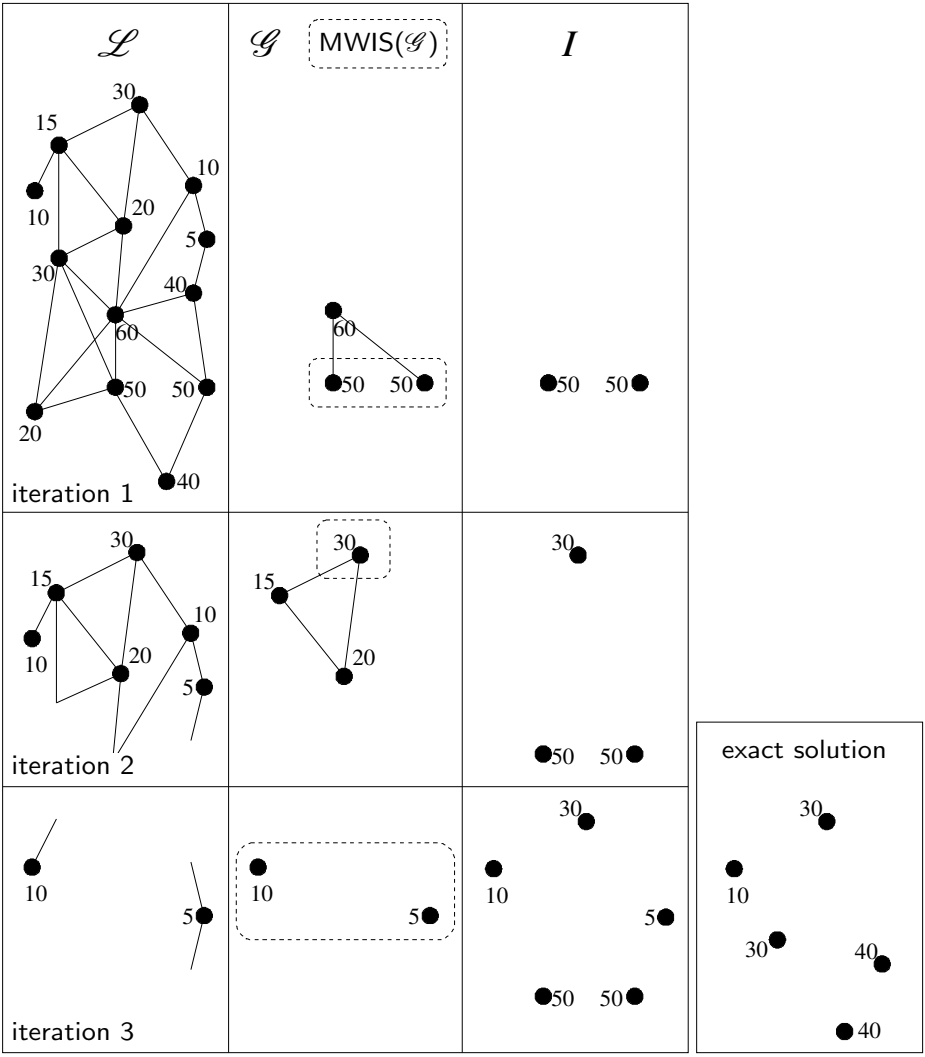


Figure 5.4: Illustrative run of the proposed maximum weight independent set (MWIS) heuristic for selecting the best boxes that do not overlap. The vertex weights quantify the power savings (e.g. nW). In this example, b, the size of the subgraph $\mathscr{G}$ for which MWIS is solved exactly, is set to three. The resulting independent set I is not optimal as the exact solution shows (right).

5.3.3 On rectangle intersection

If only one of two intersecting rectangles is selected, then it seems evident that optimisation potential of the rectangle that was not selected lies partly idle. This is discussed in more detail in Section 7.2.6.

5.4 Conclusions

This chapter described the proposed methodology to reduce the power consumption of integrated systems caused by wire capacitances. The methodology uses three major steps.

First, in Section 5.1, existing detail-routed layouts are searched for all instances of parallel running wire segments. The search goal is to deliver input data suited for subsequent wire spacing. This comprises determining layout regions limited by but not containing wire endings or vias. Only parallel wire segments are allowed to fully cross the length of the regions.

Second, wire spacing is executed on the instances found in the first step. In Section 5.2, wire spacing was briefly recapitulated to illustrate the entire flow of the methodology. Due to its importance, spacing had been described in detail in Chapter 4 already.

Third, as soon as the possible power saving results for the individual wire groups are known from the previous steps, a selection process takes place. With the help of a maximum weight independent set heuristic, only those wire groups are selected that maximise the total power saving and do not overlap. See Section 5.3 for details.

It was shown that the entire method can be implemented with quasilinear asymptotic runtime complexity in N, an important fact considering that the number of wire endings and vias $N = |V|$ can grow into billions today.

6 Experimental results

6.1 Setup

6.1.1 Benchmark circuits

Circuits suitable for serious benchmarking of routing and power consumption must fulfil the following requirements:

Size There must be a certain amount logic on the chip. Too few logic causes too sparse wire distributions with amounts of whitespace unrealistic for commercially competitive circuits. Too small circuits also do not reflect the asymptotic runtime behaviour of the algorithms developed.

Industry-related The industrial significance of claims stated increases if the benchmark circuit is commercially used or close to commercially used circuits. Purely academic examples are often not taken seriously.

Technology and tool independent Their use is limited if circuits depend on libraries of a specific manufacturer, libraries that not everyone is granted access to, or are provided in a computer-readable format that can be processed by a specific tool only.

Up-to-date Although old benchmark circuits can be synthesised in modern technology nodes – given they are provided in technology independent format – the results do not represent recent designs very well. For example, an increased number of routing layers was required to be able to implement certain arithmetic building blocks within reasonable amounts of area.

Functional layer Some physical benchmark suites contain anonymised cells and their pin locations only. For power estimation, and to measure the effect on other design parameters such as timing, however, a functional representation of the circuit is required. This can be in form of source code in RT-level or gate-level.

Testbench A testbench that simulates the circuits under typical and worst case conditions (in terms of power) is required to derive switching activities. While this causes high runtimes on gate level, at least the primary inputs and outputs of RT-blocks can be simulated.

Variety A benchmark suite, also known as training set, should comprise several different testbench circuits in order to cover as many input constellations to the algorithms as possible. With a large training set, the universal applicability can be assured with higher confidence.

Acceptance A good benchmark suite should be frequently used among researchers to increase confidence and significance of the results and possibly to provide direct comparisons with other works.

Open Benchmark circuits should be publicly available to ease replication of results claimed.

These requirements contain several contradictions. For example, commercial circuits are very unlikely to be published, unless the functional layer is removed or the circuits have reasonably aged. Large circuits are difficult to simulate on gate level. Another difficulty is comparing among several works, since benchmark circuits synthesised from RT-level may produce significantly different results depending on synthesis software, technology, and constraints set.

Some popular benchmark suites are summarised in Table 6.1. None of them fulfil all requirements of above. The closest match is provided by the IWLS-05[1] benchmark suite. It was compiled by Cadence with the purpose of providing common examples to show the usefulness of new design methodologies. IWLS-05 did not introduce new circuits. It rather comprises the ISCAS-85 [HYH99] suite[2], the ISCAS-89 extensions, and the ITC-99 Torino benchmarks as well as open source designs from opencores.org[3], Faraday[4], and Gaisler Research[5].

For this thesis, an own benchmark suite is compiled, cf. Table 6.2. It shares many of the circuits of IWLS-05 but leaves out the small circuits such as all ISCAS circuits, most Torino circuits and the Faraday circuits. Experiments have shown that these do not create meaningful amounts of wiring harness to assess the usefulness of the proposed methodologies.

[1] http://iwls.org/iwls2005/benchmarks.html, February 2007
[2] http://www.fm.vslib.cz/~kes/asic/iscas/, February 2007
[3] http://www.opencores.org/, February 2007, last update January 2007
[4] http://www.faraday-tech.com/, February 2007
[5] http://www.gaisler.com/, February 2007

Only modern designs, mainly (multi-)processor systems were selected. “Mlite” is a MIPS-compatible CPU, “eth” a 10/100 Mbit ethernet medium access controller, “dct” a single-cycle discrete cosine transformation and “3des” a triple block-cipher encryption circuit. These circuits were taken from `opencores.org`. “Leon 2” and “Leon 3” are real processor designs from the European Space Agency and Gaisler Research offering a SPARC[6]-compatible instruction sets. The Leon 2 integer unit and designs consisting of one, two, four and eight Leon 3 cores were deployed. The circuits “b14”, “b15” and “b17” to “b19” refer to two medium-sized and the three largest ITC-99 Torino benchmark circuits [CRS00], respectively.

In order to achieve a meaningful implementation of the Leon processors, tri-port memory modules for the register files are required. These were not available in the present technology library. Empty boxes were instantiated instead.

6.1.2 Synthesis flow

Figure 1.2 on page 5 illustrated how the proposed method taps into a commercial design flow externally. To synthesise the benchmark designs, a commercial state-of-the-art synthesis tool, “BlastChip APX” version 2005.03.114-linux24_x86_64 by Magma Design Automation[7] was used. This tool unifies the entire synthesis process from RT-level over gate-level and placed gates down to clock- and detail-routed layouts. Default parameters of the respective commands were used in all synthesis steps.

[6] A popular RISC microprocessor family from Sun Microsystems.

[7] Michel Berkelaar of Magma Eindhoven BV is gratefully acknowledged for his help in obtaining a university license of this tool.

Table 6.1: Benchmark suites available today with year of publication, number of circuits contained and the size of the largest circuit in (rough) number of cells.

Name	Year	Circuits	Largest	Pro	Contra
ISCAS	1985	11	6k	widespread	old, small
MCNC	1989	-	-	widespread	old, small
ITC	1999	21	100k	well-maintained	synthetic
ISPD	2002	10	1M	large, commercial	no functional layer
IWLS	2005	84	300k	well-maintained	not widespread

Table 6.2: Benchmark circuits used and statistics for hard synthesis constraints.

Model	Nets	Wires	Purpose
mlite	18k	.10M	MIPS compatible CPU
eth	27k	.10M	10/100 Mbit/s ethernet controller
dct	49k	.27M	discrete cosine transformation
3des	52k	.30M	triple des encryption
iuL2	18k	.15M	Integer unit of SPARC compatible CPU Leon 2
1cL3	38k	.22M	Instance of SPARC compatible CPU Leon 3
2cL3	58k	.36M	multi-CPU design with two instances of Leon 3
4cL3	.10M	.65M	multi-CPU design with four instances of Leon 3
8cL3	.18M	1.2M	multi-CPU design with eight instances of Leon 3
b14	5.0k	20k	Viper processor (subset)
b15	7.1k	40k	80386 processor (subset)
b17	18k	.11M	Three copies of b15
b18	46k	.27M	Two copies of b14 and two of b17
b19	.10M	.54M	Two copies of b18

6.1.3 Constraints

Apart from numerous synthesis parameters, such as the CPU-time effort vs. accuracy, or the desired process, voltage, and temperature analysis corner, there are three major design parameters which influence the routing results:

Timing The maximum allowable clock speed of a design depends on the technology used and the operating conditions applied. Many applications require a fixed clock speed for the current design in the context of the entire system. Others are to be synthesised with as fast a clock speed as possible to increase their market value.

Area Silicon cost depends on area. Also the availability of packages forces limits on the area.

Chip aspect ratio The chip or the core is not necessarily square, any aspect ratio between width and height can be chosen.

Further degrees of freedom including the placement of the pins or clock gating were not explored. With the exception of the ethernet controller, none of the

Table 6.3: Constraints used to synthesise the benchmarks.

Case	Timing	Area
relaxed	10.0 times minimum delay	60 % cell row utilisation
medium	2.0 times minimum delay	70 % cell row utilisation
hard	1.25 times minimum delay	$\geq$ 80 % cell row utilisation

present benchmarks circuits defines a target clock speed. Also an area constraint is not given, as this would make only sense in the specific instance with particular economic and electric constraints. The same applies to the aspect ratio. Thus, artificial – but reasonable – constraints have to be applied to the circuits.

One way to achieve reasonable numbers for target frequencies is to refer to comparable implementations of the same circuit. Every circuit was synthesised with an unrealistically high frequency of 1GHz from RT to gate level in the first instance. Then, the worst case path was determined with a static timing analyser and considered as the minimum possible clock period. Working at that edge of the technology is uncommon. Backing off from that value by reasonable percentages finally delivered the target clock period.

Area can either be expressed in terms of absolute chip dimensions or in target utilisation of cell area. In this work, the second possibility was made use of, for reasons indicated above.

Notice that there is one more vertical layer that is going to be optimised in processes with even number of layers, since the horizontal M1 layer can not be optimised due to too many special objects such as cell pins and power rails.

Investigations showed that the influence of the chip aspect ratio has no significant influence on the optimisation potential. The intuition for testing this was that with a larger chip dimension in vertical preferred routing direction, longer wires would emerge that might deliver better optimisation opportunities. The aspect ratio was arbitrarily set to 1.6 for all circuits, cf. [Bah06].

Table 6.3 summarises the three sets of constraints that were applied—"relaxed", "medium" and "hard". Area constraints for the three cases were set to 60 %, 70 %, and to as high a cell row utilisation as possible before normal routing would fail, resulting in around 80 %–95 %. Timing goals were 10-fold, 2-fold, and 1.25-fold minimum delay, respectively. The ethernet controller was area-constrained only, its timing was set to 25MHz fixed.

Table 6.4: Wire dimensions (nm). The values might be modified by the author to protect the intellectual properties of ST-Microelectronics.

Layer	Direction	Width	Spacing	Thickness
M1	horizontal	210	180	320
M2	vertical	200	210	320
M3	horizontal	200	210	320
M4	vertical	200	210	320
M5	horizontal	400	420	800
M6	vertical	400	420	800

6.1.4 Library and operating conditions

Up-to-date technology libraries are disclosed to research facilities only under strict agreements[8]. The information published here must adhere to these agreements. For all experiments, a 130 nm process by ST-Microelectronics was used, although 90 nm and 65 nm processes have been made available in the meantime. This technology exhibits a choice of two threshold voltages to target high speed or low leakage circuits. For the underlying work, the latter was chosen. Keeping leakage low is common in low-power circuits. Another reason is that the fraction of leakage power is small and thus the upper bound of the total power reduction by reducing the dynamic power can be explored.

Of the wide range of conditions (process derating coefficients, supply voltage and temperature) that could be applied, a timing-worst condition was selected. The nominal supply-voltage was derated to 1.08 V and the temperature was assumed to be 85 °C.

Routing was performed on six metal layers. Minimum distances of the wires and widths were set to the values listed in Table 6.4. The thickness is a technology-dependent constant and cannot be modified by the designer, but it significantly affects the capacitance curves. There are more complex design rules, all in all a 100-page document. Most of the rules affect the active silicon areas and are not applicable during routing. However, there also exist additional design rules for routing. For example, very broad wires require higher minimum spacings. Broad wires were turned off as these are required only in special cases.

[8] CMP Tima in France is gratefully acknowledged for its engagement.

6.1.5 Activities

Deriving switching activities is difficult. Large companies with huge amounts of CPU power may be able to derive exact values for α by simulating the gate level netlist after layout. For this purpose, a testbench must be available that applies stimuli that reflect the intended use of the circuit. For a processor, these stimuli are delivered by the programs it executes. A test for the smallest processor considered, mlite, had shown that execution times are in the order of one hundred instructions per second when simulated on gate level.

Instead, switching activity propagation was performed as offered by BlastChip, cf. Chapter 2. Toggle activities of 0.5 at the primary inputs were assumed. Propagation respected gate and wire delays and was run with sixteen iterations. That parameter is an optional number that increases the approximation quality at the cost of runtime. The default value is two.

Power estimation based on activity propagation is known to give good relative results [Emb04] but the absolute power number can be significantly off. The same must be assumed for individual activity values. Thus, the optimisation with propagated activities should be considered as proof of concept. Industrial applications of this work should be at least partly based on simulated activities.

6.1.6 Server

Experiments were conducted on an IBM x-series server[9] with two dual-core Xeon processors with 64-bit registers, 36 bit physical and 48 bit virtual address widths, and 3.2GHz clock frequency[10]. The random access memory of four gibibyte[11] was sufficient to synthesise and optimise all circuits considered but limited the 3D capacitance extraction to the smaller circuits. The operating system was Linux, 64-bit kernel, version 2.6.13-15.7-smp of the openSuSE (version 10.0) distribution.

6.1.7 Implementation

By implementing appropriate interfaces, layout information and switching activity could be accessed in *TopCool*. The entire proposed methodology was implemented

[9] IBM is acknowledged for the server.

[10] CPU family 15, model 4.

[11] http://de.wikipedia.org/wiki/Gibibyte: A gibibyte (a contraction of giga binary bytes), abbreviated GiB, is a unit of 2^{30} byte. This notation was introduced by the International Electrotechnical Commission in 1999 to eliminate the 2^{30} vs. 10^9 ambiguity of one gigabyte.

Table 6.5: TopCool parameters used to optimise the benchmark circuits.

Parameter	Symbol	Value	Reference
manufacturing grid	m	8 nm	Section 4.1.2.
start step size factor	a	16	Section 4.1.2.
minimum box length	–	11 pitches	Section 5.1
minimum box width	–	1 pitch	Section 5.1
minimum power saving per box	$\Delta P_{\text{box}}/P_{\text{box}}$	4 %	Section 5.2
subgraph size	b	32	Section 5.3.2.

in C and C++. As compiler, `gcc`, version 4.0.2 20050901 (prerelease) was deployed. Compiling included all optimisations with switch `-O3` and generated 64-bit code.

Apart from standard libraries including the standard template library (STL), the only external source was the open source MWIS solver "Cliquer" [NO03] which was used to solve instances of the rectangle selection problem exactly.

Parallel programming techniques were not deployed, as experiments showed that using all four processor cores to optimise layers in parallel gained only a speed-up of a factor of about two. Difficulties in communication between and debugging of parallel threads lead to abandonment of threaded programming.

Table 6.5 summarises the parameters used for the experiments. These were derived manually to achieve a reasonable trade-off between speed and quality. The manufacturing grid was not specified by the library vendor. A value of 8 nm seemed appropriate. Smaller values did not increase the power savings significantly.

6.1.8 Processing layouts

After synthesis as well as after optimisation, the layouts were processed further. An industry-standard full-chip 3D capacitance extractor, "QuickCap" by Magma Design Automation delivered more accurate values for the capacitances than the 2.5D extractor integrated into BlastChip could deliver. In particular, the differences were significant for wires at off-grid positions.

QuickCap uses a Monte Carlo based extraction method. With default parameters it extracts every net with the same accuracy and the accuracy increases as more runtime is spent. It is possible to specify the driving resistances R of every net to be extracted to quickly determine the capacitances of the wires in the critical paths. QuickCap then automatically focuses only on the accuracy of ΣRC. This

Table 6.6: Interconnect power reduction methods used for comparison.

Name	Reference	Principle
Spread	[SD97], implementation e.g. by Magma	Commercial, post-routing non-activity-driven wire spreading. Implementation not documented.
Sanyo	[SYMY03a]	Routing with widened routing pitch.
Intel	[MKWS04]	Routing with widened routing pitch for active wires.
TopCool	this / [ZBIS07]	Post routing activity-driven re-distribution of local whitespace.
Sanyo + *TC*	[SYMY03a] + this / [ZBIS07]	Routing with method of Sanyo chained by application of *TopCool*.
Intel + *TC*	[MKWS04] + this / [ZBIS07]	Routing with method of Intel chained by application of *TopCool*.

accelerates extraction a lot since Cs associated to very smalls Rs can be of low accuracy. This principle was exploited in this work by specifying α values instead of R values to quickly get accurate numbers for $\Sigma\alpha C$ which is proportional to the power consumption of the wires[12].

Gate level power analysis to measure the power consumption of the circuits and design rule checking (DRC) were performed with BlastChip. No design rules were violated by applying the activity-driven wire optimisation.

6.2 Optimisation results

6.2.1 Notes on other approaches

To fully validate the usefulness of *TopCool*, not only its power savings are reported but also those achieved by implementations of other techniques. Table 6.6 summarises the methods used. While these have been described in Chapter 3 in detail, some comments have to be made on their implementations.

Spreading is a post-routing operation. It is applied after detailed routing and returns a new layout. It widens wire distances locally, targeting a uniform wire distance. A parameter exists that balances the jog length against the spreading

[12]This idea could make up a decent application note for QuickCap.

opportunity. It was set to full spreading opportunity. Spreading is developed for yield so for the sake of a fair comparison, also the yield limits are compared.

The method of Sanyo is a routing method. The entire design flow including placement is identical to the reference design flow. The placement output is saved and re-loaded every time a new routing attempt is started. While the original paper suggested to route at 25 %, 50 %, 75 % and 100 % increased pitch for layers three to six and layers four to six, and picking the best solution, only the option four to six was selected in this work. Increasing the pitch also for layer three caused routing to fail immediately in all instances. Loading the placement, setting the pitch, and starting clock-, global- and final routing was done with a TCL[13]-script which calls the appropriate commands inside the Magma software.

Also the method of Intel [MKWS04] is a routing method. It routes the 5 % most active nets first and with twice the normal spacing, and then all other wires. It also comprises a power-driven rip up and re-route scheme, which is, however, not further specified. For this reason, a simplified version was implemented. It forces double spacing for the 4 % most power consuming nets in layers four to six. This comprises two important differences.

First, the percentage was selected intuitively as the original paper only specified an empirical value smaller than 5 %. A global optimisation to find the best values for the percentage and the space widening for every circuit was not performed. Second, due to superior results, the percentage was applied to the most power consuming nets as opposed to the paper which suggests to use the highest activity nets.

A TCL script was used to estimate the power for every net by multiplying the activity with the estimated capacitance value extracted from the placement information. It then applies the double spacing rule to the most power consuming nets and starts the router for all nets. Routing the active nets first and then the remaining nets as described in the original work was implemented but the implementation was discarded as the router used could not complete the designs that way.

Despite the simplifications compared to the original work, the method is referred to [MKWS04].

Table 6.7: Interconnect power of reference layouts with relaxed constraints and optimisation results of commercial wire spreading, Sanyo [SYMY03a], Intel [MKWS04], the presented method TopCool, and combinations of the latter.

Model	Wire power consumption (mW)						
	ref	spread	Sanyo	Intel	*TopCool*	Sanyo+*TC*	Intel+*TC*
mlite	.2361	.2357	.2255	.2134	.2240	.2155	**.2110**
eth	1.516	1.485	1.542	1.393	1.415	1.462	**1.361**
dct	1.988	2.004	1.974	1.868	1.909	1.908	**1.840**
3des	4.331	4.278	4.164	4.031	4.126	4.048	**3.951**
iuL2	1.838	1.761	*	*	**1.704**	*	*
1cL3	1.617	1.561	1.507	*	1.511	**1.451**	*
2cL3	2.910	2.770	*2.495*	*	2.666	***2.390***	*
4cL3	6.152	5.855	6.220	*	**5.656**	5.925	*
8cL3	11.09	*10.49*	*	10.96	**10.12**	*	10.43
b14[1]	17.45	16.70	16.30	16.34	16.15	**15.32**	15.91
b15[1]	45.18	43.83	45.14	43.19	**41.84**	43.50	41.94
b17[1]	99.94	98.14	92.34	90.82	92.71	**88.06**	88.77
b18	.4415	.4278	.4224	*.3757*	.4050	.4016	**.3677**
b19	1.256	1.218	1.168	1.075	*1.136*	1.115	***1.032***
max (-%)		*5.41*	*14.3*	*14.9*	*9.55*	***17.9***	*17.8*
avg (-%)		2.90	3.78	5.79	7.10	7.34	**7.48**

[1] μW **bold: best in row** *slant: best in column* * over-congested

6.2.2 Power saving results

Tables 6.7–6.9 list the wire power consumption of the benchmark circuits before and after applying the described power saving methods according to the formula $P = V_{DD}^2 \cdot f \cdot \Sigma_i \alpha_i \cdot C_{net,i}$ for the cases of relaxed, medium and hard synthesis constraints, respectively. An asterisk (*) is inserted in a table cell if routing was not possible with the respective method. In that case, the result of the reference solution is used to build the average which is displayed in the bottom row.
The power saving results attribute the following properties to *TopCool*.

[13] Tool Command Language, quasi standard interface language of design tools.

Table 6.8: Interconnect power of reference layouts with medium constraints and optimisation results of commercial wire spreading, Sanyo [SYMY03a], Intel [MKWS04], the presented method TopCool, and combinations of the latter.

Model	Wire power consumption (mW)						
	ref	spread	Sanyo	Intel	*TopCool*	Sanyo+*TC*	Intel+*TC*
mlite	2.366	2.343	1.940	2.054	2.233	***1.836***	2.032
eth	1.420	1.391	1.422	1.316	1.325	1.354	**1.283**
dct	12.82	12.74	12.43	11.91	12.16	12.03	**11.68**
3des	20.61	20.40	19.77	18.96	19.72	19.25	**18.59**
iuL2	12.90	12.43	*	*	**12.07**	*	*
1cL3	11.22	10.80	*	10.63	10.38	*	**10.23**
2cL3	20.59	19.71	*	19.00	18.96	*	**18.22**
4cL3	28.57	27.28	*	29.04	**26.33**	*	27.96
8cL3	55.49	*52.78*	*	*	**50.91**	*	*
b14[1]	92.46	88.83	83.90	*77.69*	*83.76*	79.89	***74.84***
b15	.2297	.2244	.2121	.2002	.2120	.2031	**.1937**
b17	.4856	.4752	.4399	.4370	.4501	**.4227**	.4275
b18	1.842	1.791	*1.667*	1.581	1.698	1.590	**1.542**
b19	5.152	5.010	4.973	4.436	4.724	4.762	**4.382**
max (-%)		*4.88*	*18.0*	*16.0*	*9.41*	***22.4***	*19.1*
avg (-%)		2.83	4.59	8.13	7.16	7.08	**10.2**

[1] μW **bold: best in row** *slant: best in column* * over-congested

1. Both wire spreading and *TopCool* are post routing operations, so they can be applied to any design, no matter how congested the routing area is. This is not true for [MKWS04] and [SYMY03a] which modify the routing constraints of an existing router. The commercial router used honours the applied rules very well for small designs with relaxed constraints, and is able to save remarkable amounts of power. It fails, however, on larger or more tightly constrained designs due to congestion problems, limiting the universal applicability of [MKWS04] and [SYMY03a].

   Some of the circuits investigated, for example b14 and b15, could even have been routed with less than six layers, even if the hardest timing constraints

Table 6.9: Interconnect power of reference layouts with hard constraints and optimisation results of commercial wire spreading, Sanyo [SYMY03a], Intel [MKWS04], the presented method TopCool, and combinations of the latter.

Model	Wire power consumption (mW)						
	ref	spread	Sanyo	Intel	*TopCool*	Sanyo+*TC*	Intel+*TC*
mlite	4.396	4.396	*	*	**4.065**	*	*
eth	1.388	1.361	1.403	1.263	1.296	1.342	**1.232**
dct	39.20	38.67	37.13	36.07	37.03	35.87	**35.23**
3des	35.71	35.28	33.86	32.42	34.16	33.04	**31.80**
iuL2	16.80	16.24	*	*	**15.88**	*	*
1cL3	12.71	12.23	*	*	**11.82**	*	*
2cL3	22.06	21.15	*	*	**20.48**	*	*
4cL3	38.04	*36.25*	*	*	**35.20**	*	*
8cL3	69.81	66.61	*	*	**64.68**	*	*
b14	.2394	.2312	*.1991*	.2041	*.2184*	***.1882***	.1966
b15	.4365	.4248	.3953	*.3477*	.4017	.3761	***.3366***
b17	.7442	.7258	.7350	*	**.6918**	.7092	*
b18	3.939	3.826	3.657	3.430	3.656	3.513	**3.360**
b19	10.91	10.62	10.45	*	10.16	**10.10**	*
max (-%)		*4.71*	*16.8*	*20.3*	*8.77*	*21.4*	***22.9***
avg (-%)		2.80	3.45	5.30	**6.88**	5.53	6.27

[1] μW **bold: best in row** *slant: best in column* * over-congested

were applied. It is evident that these circuits provide a high optimisation potential for all methods. It was found that designs with arithmetic modules create higher amounts of wiring, especially the Leon 2 integer unit. These cannot be routed with increased pitches or reduced numbers of layers, but can be optimised by *TopCool* very well.

2. *TopCool* saves about twice the amount of power as spreading that does not take activities into account. This fully agrees with numbers published for buses [MMP02].

3. *TopCool* achieves the highest average power savings of all methods with

hard constraints. Furthermore, the quality of the optimisation is largely independent of the circuit size and the constraints set. The largest circuit considered, eight Leon 3 cores with 1.1 million wires, shows a power reduction of 8.25 % in the medium constrained case, where the average reaches its best value of 7.16 %.

4. *TopCool* cooperates well with [MKWS04] and [SYMY03a]. All of the highest power savings are achieved with *TopCool* or one of the combinations, see bold entries in the tables.

5. As expected, the additional power reductions by *TopCool* on top of the methods [MKWS04] and [SYMY03a] are less than the power reductions of *TopCool* alone. The differences of the maximum reduction percentage scores between Sanyo and Sanyo plus *TopCool* are 3.6, 4.4 and 4.6 for the three constraint cases and 2.9, 3.1 and 2.6 between Intel and Intel plus *TopCool*. The less additional potential for Intel seems plausible as that method already provides more space for the active wires.

6.2.3 Runtime

The CPU runtimes as a function of the number of wires are displayed for the medium constraints in Figure 6.1. Curves for other constraints behave similarly. Unlike suggested in Chapter 5, the current implementation uses the $O(N^{1.5})$ worst case search algorithm described in [Iln06], if N is the number of wire endings and stacked vias. This results in a measured $O(n^{1.19})$ complexity, n representing the number of wires over the considered benchmark circuits, cf. thin plot in Figure 6.1. Nevertheless, the current implementation operates about five times faster than commercial spreading on average, and more than one, and up to two orders of magnitude faster than [MKWS04], and [SYMY03a], respectively.[14]

The runtimes are given as additional runtime compared to the standard design flow. The high values for [MKWS04] and [SYMY03a] stem from the additional constraints applied to the commercial router and the increased number and difficulty of routing attempts. The break-in of the runtimes of these methods for the Leon 2 integer unit (124k wires, cf. Figure 6.1) is plausible, as the router fails very early due to the very high routing resource utilisation of the arithmetic building blocks. In all other failing cases, these methods fail later.

[14] In the meantime, I implemented the quasilinear version and submitted the results to IEEE [ZBIS07]. In brief, the runtime decreased by about 40,%, compared to the numbers published in this document.

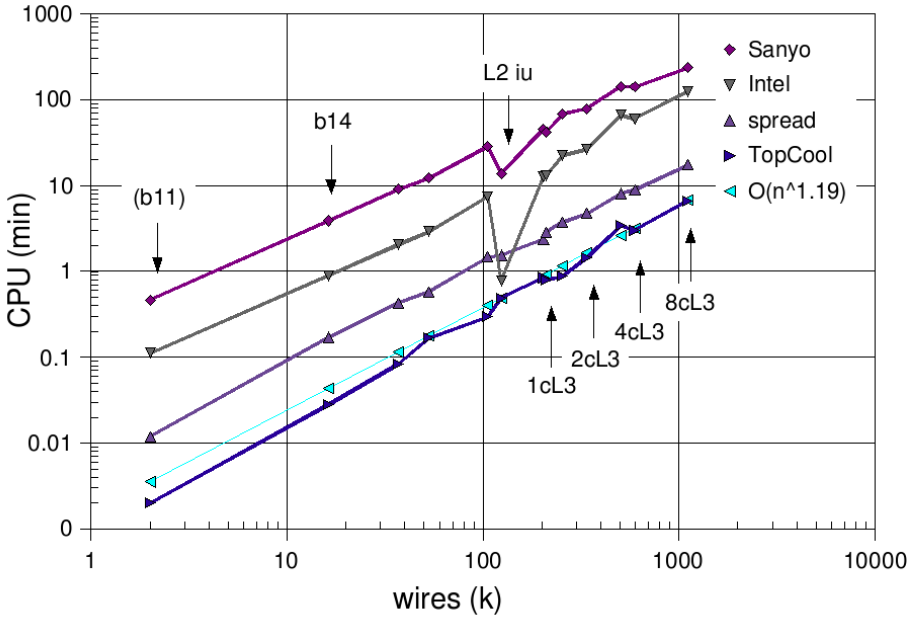
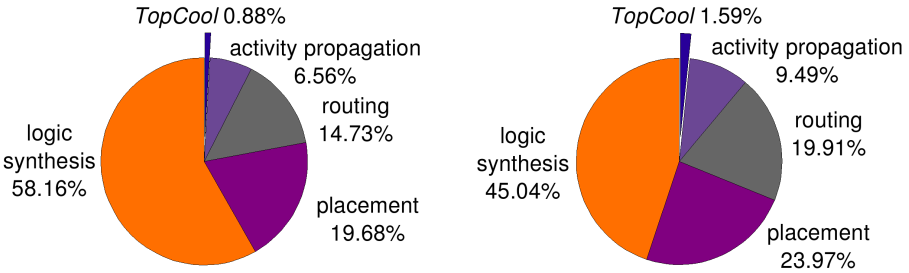


Figure 6.1: Comparison of the CPU runtimes of the considered low power routing techniques. The x-axis represents the number of wires as a measure of design complexity.



(a) 1-core Leon3

(b) 8-core Leon3

Figure 6.2: Comparison of the CPU runtime relative to synthesis time (medium constraints).

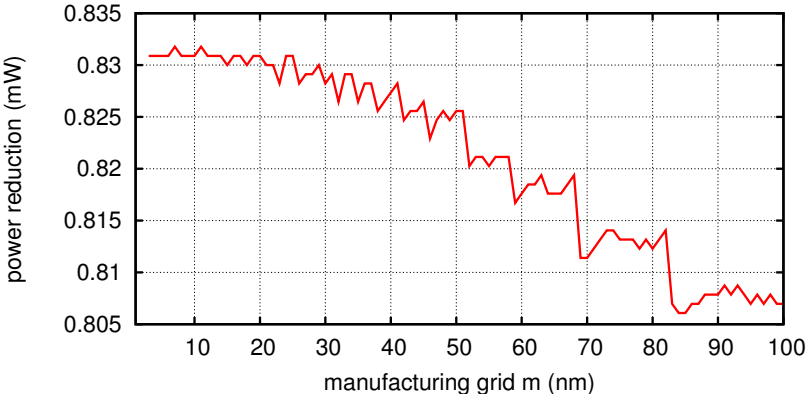


Figure 6.3: Influence of the manufacturing grid on power saving.

TopCool is actually executed twice, the first time with the original layout as input, and the second time on its own output of the first run, saving about 20 % more power than in one run alone. The time for the two runs is already considered above. More than one additional run did not show significant advantages.

Compared to the entire design flow, cf. Figure 6.2, *TopCool* makes for circa 1 % of the CPU time. The method depends on activities whose propagation takes about 6 %–10 % of the total time at the choosen accuracy. In serious low power projects, however, these are calculated anyway. Routing alone accounts for about 15 %–20 %. The relative increase of the runtime of *TopCool* and the design steps other than logic synthesis in the large design can be attributed to the relative decrease of logic synthesis time (hierarchical logic synthesis of eight similar single-processor cores).

6.2.4 Manufacturing grid

The manufacturing grid is limited by the resolution of the lithography tools used. This value is not specified for the given technology. Apart from that, process variations make it difficult to adhere to such a nominal grid exactly. Therefore, experiments were set up that show the influence of the assumed manufacturing grid on the power saving results. The outcome for the Leon 2 integer unit example is shown in Figure 6.3. Power saving results are virtually constant below about 25 nm, and at 100 nm, only about 3 % of the optimisation potential is lost.

6.2.5 Timing

In general, timing is the most important design goal. Especially at very late phases of the design flow, timing must comply with its nominal value specified. Therefore, the effect on timing of applying the presented methods was measured after RC extraction. While a hold time violation was not reported in any instance, the worst change in critical path delay was less than +1%, and -1.47% on average, cf. Table 6.10. Only in three of the 42 circuits considered, the critical path delay increased. All three occurances had hard constraints, though. Skew did not change by more than -18ps...+14ps in any design.

Timing is not significantly worsened. This can be attributed to the exploitation of local whitespace to primarily reduce capacitances and the cancellation of delay increases and decreases of critical path wires. On the contrary, the delays on most paths, especially those containing high activity nets, are even improved. Up to 9.45% were reported, see Table 6.11.

The router deployed also modifies the placement and to a certain extent also the logic. This explains that a 44% shorter critical path was found for the circuit dct after applying Sanyo [SYMY03a] in the best case.

The Outlook provides a more detailed discussion about the circuit timing in Section 7.1.1. Section 7.2.9 recapitulates post routing gate sizing that can be used to trade the new slack for smaller gates and thus further power optimisations, as suggested in [MKWS04].

6.2.6 Yield

A sound physical design methodology must observe effects on yield. Table 6.12 lists two types of yield-limits altered by the presented method. Particle-related shorts are reduced as a by-product of whitespace reallocation as the critical area is decreased. Detouring wires, on the other hand, increases the wire length and thus the probability of opens. The numbers are normalised to one cm^2 of silicon covered with the respective circuit, and were extracted with the help of the "Edinburgh Yield Estimation System[15] (EYES)" [All06].

Commercial wire spreading is optimised to reduce probabilities of shorts. As expected, it achieves higher yield-limits, 0.37% better than the reference design, while *TopCool* improves yield by 0.12%. Section 7.1.2 contains suggestions on how to improve the yield-limit of *TopCool*.

[15] Gerard Allan of Predictions Software Ltd. is acknowledged for a university license of this software.

Table 6.10: Critical path delay relative to reference design, worst case over all constraint sets. The critical path delay after applying TopCool increased only in three of $14 \times 3 = 42$ (14 circuits, three constraints) cases considered.

Model	Critical path delay (%)					
	spread	Sanyo	Intel	*TopCool*	Sanyo+*TC*	Intel+*TC*
mlite	0.37	1.47	0.76	0.37	1.33	0.51
eth	-0.24	0.15	-0.22	0.96	0.18	0.68
dct	-0.32	1.82	5.54	-1.37	0.46	4.78
3des	-0.17	0.41	1.74	0.00	0.60	0.56
iuL2	-0.44	0.00	0.00	-1.17	0.00	0.00
1cL3	-0.28	0.49	0.01	-0.78	0.00	0.00
2cL3	-0.69	0.98	0.00	-0.53	0.13	0.00
4cL3	-0.66	2.65	2.34	-0.24	2.31	0.02
8cL3	-0.58	0.00	0.39	-0.43	0.00	0.00
b14	0.22	0.26	1.35	-0.37	-0.32	0.27
b15	-0.50	-0.12	0.34	-1.35	-0.84	-0.34
b17	-0.21	0.65	-0.34	-1.02	0.75	-0.80
b18	-0.43	0.12	0.09	-0.75	-0.40	-0.86
b19	1.50	1.11	0.00	0.16	-0.17	0.00
avg	-0.61	-1.27	0.01	-1.47	-1.47	-0.35

Table 6.11: Critical path delay relative to reference design, best case over all constraint sets.

Model	Critical path delay (%)					
	spread	Sanyo	Intel	*TopCool*	Sanyo+*TC*	Intel+*TC*
mlite	-0.85	-0.24	0.00	-1.89	-0.33	-0.57
eth	-1.79	-2.32	-0.43	-2.06	-1.93	-1.66
dct	-0.52	-44.35	-1.74	-1.71	-44.82	-2.50
3des	-0.55	-1.65	-1.56	-1.81	-1.68	-0.80
l2	-1.12	0.00	0.00	-1.55	0.00	0.00
1l3	-1.23	0.00	0.00	-0.97	-0.35	-0.16
2l3	-1.11	0.00	-0.17	-1.57	0.00	-0.36
4l3	-1.26	0.00	0.00	-9.45	0.00	0.00
8l3	-1.16	0.00	0.00	-9.19	0.00	-0.32
b14	-2.44	-2.47	-0.41	-1.60	-2.93	-1.10
b15	-0.86	-2.39	-0.83	-2.08	-2.08	-1.26
b17	-0.76	-0.51	-0.40	-1.53	-1.18	-1.21
b18	-0.86	-0.53	-1.29	-1.29	-1.37	-1.82
b19	-0.28	-6.50	-1.80	-1.80	-3.05	-2.76

Table 6.12: Yield-limits improve compared to the reference design, but the potential of commercial spreading is not reached.

Case	Average yield-limit M2…M6 shorts / opens (%)		
	ref.	spread	*TopCool*
relaxed	97.449 / 98.891	97.841 / 98.842	97.615 / 98.850
medium	97.050 / 98.790	97.446 / 98.740	97.209 / 98.747
hard	96.306 / 98.611	96.749 / 98.557	96.465 / 98.562
total	95.742	96.097	95.859

In the following, metal layer M4 of the Leon 2 integer unit (hard constraints) is used as example. Figure 6.4 (top) depicts the defect size distributions assumed by EYES. The tool extracts the critical area, i.e. that area on which a particle causes harm, cf. Figure 6.4 (middle). The fault probabilities of Figure 6.4 (bottom) are products of the above.

Spreading and *TopCool* reduce the critical area for frequent, small particles and increase it for less frequent, large particles. This can be explained by unlocking large areas of whitespace and dividing it into smaller amounts in order to increase wire distances. It seems that spreading reduces the fault probabilities mainly due to particle sizes in the order of two wire widths.

6.3 Effect of dropping correlations

The effect of replacing c between two wires by the sum of their α-values is quantified empirically for bus wires which are assumed to be driven by unskewed flip-flops. Only if signals do not toggle at the same time within a clock period or if the signals are not correlated, then c can be replaced by $\alpha_{\text{left}} + \alpha_{\text{right}}$ without error.

A Leon 3 system running the stack and integer benchmark program "stanford" is used. Address and instruction buses were traced. In addition, a data bus that transports video data pixel by pixel on 24 lines (eight bit for red, green and blue each), and a binary 32-bit counter are considered. More details about the benchmarks used for this experiment can be found in [ZBI+07].

If the replacement of c is done despite simultaneous switching and correlation, errors in the power formulae occur. These errors are illustrated for these four typical-case examples in Figures 6.5 and 6.6.

Each figure shows three curves. A curve marked with "optimise" displays the case where crosscoupling activities were used for optimisation and for measuring the reduction. This is the optimal case. The curve marked with "analyse" represents the power saving results achieved by neglecting the correlations during optimisation, i.e. correlations are used for final analysis but not during optimisation. This reflects the real physical power savings of achieved by a simplified optimisation. The curve "none" shows the results when also predicting the savings with wrong correlation coefficients. This represents what the optimisation "thinks" it achieves.

It can be concluded that crosscoupling activities can be neglected during optimisation. The loss in optimisation potential is low. The misprediction when dropping

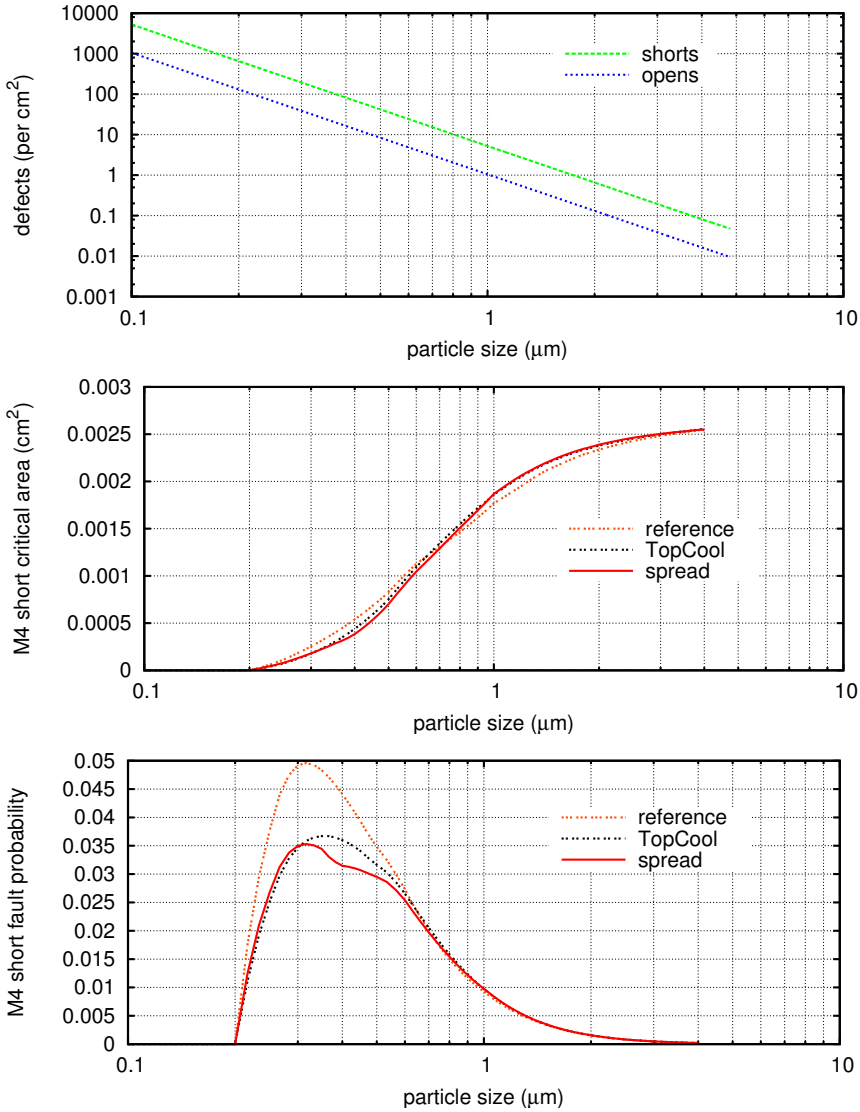


Figure 6.4: Top: the defect size distributions assumed for all layers follow a $1/x^3$ law, according to [Sta83, All06]. Middle: critical area with respect to particle-related shorts on layer M4 for the Leon 2 integer unit. Spreading and TopCool reduce the critical area for small particles and increase it for large particles. Bottom: respective fault probability distributions. The integral under the distribution is the fault probability.

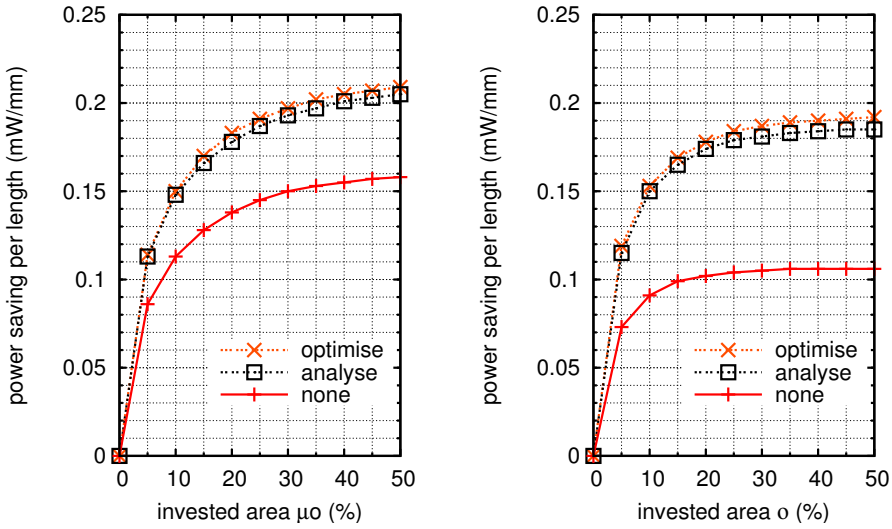


Figure 6.5: Positive error in power saving prediction when crosscoupling activities are neither used for optimisation nor for analysis. Left: Stanford address trace, right: 32-bit counter.

crosscoupling activities, however, is higher. Savings can be over- or underestimated for buses or groups of parallel wires in large designs. This means that the box selection heuristic described in Chapter 5 can be guided by poorly estimated node weights. It might select a box with supposedly higher power saving and that may eliminate the opportunity for supposedly unattractive, overlapping boxes to contribute to the overall power reduction. The the resulting independent set might not be of highest possible weight.

For three major reasons, however, it is expected that these errors diminish when optimising at multiple locations in large circuits. First, adjacent wires in large designs may but do not necessarily exhibit the same electrical distance from the driving flip-flops. The probability that any two wires toggle at the same time within one clock period is roughly one half to the number of logic stages.[16] Therefore, the majority of adjacent wires can be assumed not to toggle at the same time. The number of these problem cases is commonly further reduced by wire swapping or clock skewing techniques in physical design tools to avoid timing and

[16] Neglecting spatial correlations and glitching.

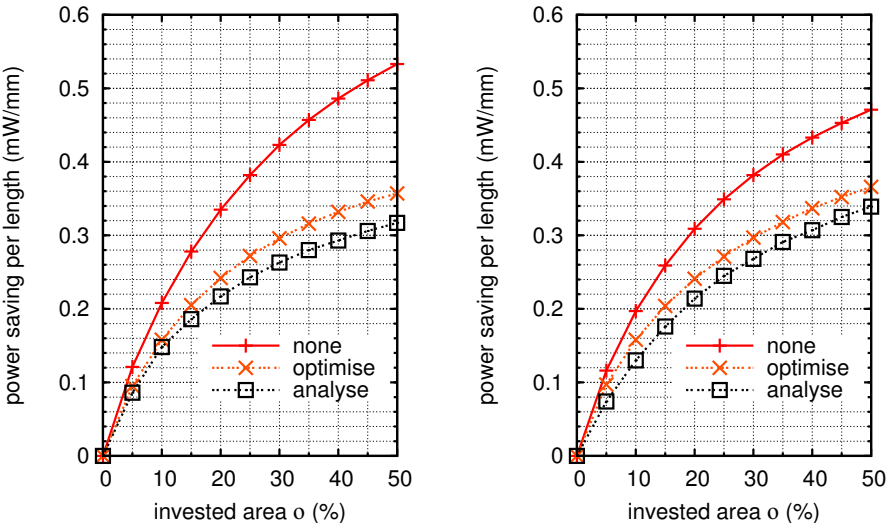


Figure 6.6: Negative error in power saving prediction. Left: Stanford instruction trace, right: 24-bit video data.

crosstalk problems. The three curves in Figures 6.5 and 6.6 approach each other if drawn for wire groups in real layouts. Second, over- and under-estimations cancel out each other. The independent set may contain boxes with supposedly higher and supposedly lower power consumptions. Third, the presented in-place spacing technique operates mostly with low quantities of remaining whitespace where the prediction error is relatively small. See also [ZBIS07] for a quantitative justification.

Nevertheless, an exact error bound can not be provided. Unfortunately, this problem has no known solution. See also Sections 2.2 and 7.2.1 in the Outlook.

6.4 Effect of wire ordering

In the experiments for wire ordering, N wires were distributed on $M > N$ possible tracks thrice; the power-optimal order, the power-worst order, and an order sorted by activities to represent a typical case, were used. The activities of the wires were randomly chosen from a histogram that is similar to commercial circuits [Emb04]. A $1/x$ shape was used because it reflects very well the fact that few wires toggle very often but many wires are quiet.

Table 6.13: Reduction potential of wire ordering on M tracks (%).

(a) N random wires.

N	Number of tracks M			
	N+1	1.25N	1.5N	1.75N
8	3.5	5.8	8.3	9.6
16	3.6	9.3	12.4	13.7
64	3.4	16.5	19.2	19.9
256	3.0	20.3	22.5	22.7

(b) N pre-sorted wires.

N	Number of tracks M			
	N+1	1.25N	1.5N	1.75N
8	2.6	4.3	6.2	7.2
16	2.7	7.4	9.9	10.9
64	2.6	14.4	17.0	17.6
256	2.3	19.0	21.1	21.3

Tables 6.13(a) and 6.13(b) show the difference of the wire power consumption before and after re-ordering and subsequent spacing. Note that the values do not directly reflect the effect of spacing but rather the difference between spacing with the one order and spacing with the best order:

$$\frac{P_{\text{some order and optimal spacing}} - P_{\text{best order and optimal spacing}}}{P_{\text{best order and optimal spacing}}}.$$

In other words, a power reduction potential of up to the values in the tables could be wasted if wire ordering was not cared for. The values are averaged over some 300,000 instances.

As Section 4.2 and Appendix A point out, the order is easily achieved. More details on the wire ordering experiments have been pre-published in [ZGRS05]. This work contains an additional interesting observation for the system designer. She can make out an upper optimisation limit of parallel wires by looking at just two parameters, the highest and the lowest switching activity. The closer the ratio of these two is to one, the less optimisation potential exists.

The Outlook offers future research suggestions on wire ordering in Section 7.2.7. Results and further developments are to appear in an applied mathematics journal [GRSZ07] and in the doctoral thesis of Michael Ritter.

7 Limitations and future work

This chapter is divided into three sections. The first section provides additional investigations on the influence of the suggested method on timing, yield, and crosstalk. Section 7.2 elaborates on further power saving improvements and points out guidelines to extend the technique accordingly. The last section sheds light on the robustness of the method. Robustness is used in this context as the applicability of the method when moving to a different fabrication technology [Len90].

7.1 Multidimensional objectives

TopCool reduces the switched wire capacitance. Besides area, no other design properties are considered. The interconnect capacitance, however, influences not only capacitive power but also short-circuit power and – more importantly – timing and signal integrity. In addition, the manufacturing yield is altered by moving wires.

This section briefly summarises the observations on the before-mentioned objectives and investigates the abilities of *TopCool* to optimise for those. Multidimensional objectives are used for instance in [SYMY03b]. A weighted product for several objectives, P_i is minimised instead of a single objective:

$$\Pi P_i^{e_i} = \min!$$

An individual exponent e_i emphasises the importance of particular objectives, such as power, yield, delay, or crosstalk noise. A popular example is the power-delay product. Multidimensional objectives must be used if the resource "whitespace" is intended to be traded for concurring objectives. Currently, the improvement of other variables than power – if at all – are by-products.

7.1.1 Timing

Delay represents a special case. Timing closure is an instance of a hard, i.e. constrained objective. While optimising for some objective such as power, no

path delay must exceed a certain value. This is a different problem type than minimising one quantity as far as possible. Handling timing constraints exactly during power optimisation would require a different approach.

Timing, however, is the most important circuit variable, even with increasing importance of power, yield and crosstalk. If a timing specification of a model is not met, the circuit will most likely not work. In the best case, it can still be rated for a lower clock speed and sold at a lower price. But in general, ASICs must operate in a specified environment that imposes one particular clock speed on the circuit. Apart from that, hold time violations always render a chip invalid independent of the clock period.

TopCool does not inherently affect timing, as opposed to many other low power techniques, such as supply voltage reduction or gate down-sizing. In other words, no power-delay trade-off is required. As listed in Chapter 6, the current method shows marginal influence on the critical path delay of numerous benchmark circuits. Significant timing violations do not occur in the circuit examples considered. However, that does not guarantee that timing is not destroyed for other examples.

Timing is influenced in four ways. First, detouring a timing critical wire is likely if it is not power critical. The added wire length contributes to resistance and capacitance and thus delay. Second, detouring a wire towards a timing critical neighbour (because both are not power critical) will increase the shared capacitance in between and cause a delay increase as well, possibly subject to a Miller factor. Both situations may lead to a setup time violation. A hold time violation might occur if, third, space is added around a power critical wire of a short path. Care must be taken, fourth, when dealing with the clock tree. Advancing or retarding the delay on a clock tree branch by adding or removing capacitance modifies the clock skew. This can cause further setup and hold time violations [Sau99].

It would require a somewhat pathological sample circuit to create a severe setup time violation. Its critical path would have to consist mainly of quiet signal nets whose wires would already have to be surrounded by considerable amounts of space.

Wire spreading for yield as reported in [SOC06] has much higher reported impact on both types of timing. Unfortunately, the used spreading algorithm and example circuit is not documented. Significant success on reducing the number of timing violations was met with by introducing a double spacing rule for setup critical nets, and minimum spacing rules for hold critical nets during spreading. The critical area quality was thereby not severely affected. The same method can readily be used for *TopCool* if significant timing violations should occur.

Rather than preserving timing when optimising for power, the *TopCool* method can also be modified to fix timing violations late in the traditional flow. Timing can not be met after detailed routing exactly because wire lengths and capacitances are hard to predict. As a result, timing violations occur late in the flow. By replacing the switching activity in the *TopCool* method by a measure for timing criticality, the application of *TopCool* reduces the overall extent of timing violations, eliminating the need for costly re-design cycles at higher levels.

Additional design possibilities exist by modifying the wire widths [CKP01]. Wire width is proportional to wire conductivity in the first order. The objective function of *TopCool* and the maximum area constraint must be altered to include the wire widths as variables. Furthermore, new constraint functions for the widths are introduced to reflect the timing requirements. It must be verified, however, if the optimisation problem remains convex.

7.1.2 Yield and manufacturability

Section 6.2.6 shows that the critical area for a particle related short is decreased. This is easy to understand as wires get separated from each other by using remaining whitespace after detailed routing. In comparison to a dedicated wire spreading tool, the reduction of critical area in *TopCool* is a by-product. It is expected that the present method can be tuned to work significantly better in optimising for yield, for power, and for a combination thereof than current spreaders.

The change to be made is a modification of the spacing goal. As smaller particles occur more frequently than larger particles [SD97], the goal should be to distribute the present whitespace evenly. This is already done in current commercial spreading tools, but based on the routing pitch, or possibly a half or a quarter routing pitch. Finer resolutions are not supported. The result delivered by *TopCool* is quasi-gridless. Therefore, free space caused by for instance only one unallocated routing track can be used to spread multiple surrounding wires evenly. The quantitative advantage for yield must be verified, though, because Figure 6.3 in the previous Chapter has shown a negligible influence of the assumed grid on the power saving. Currently, regions of low power savings are discarded and not modified in the layout, cf. Section 5.2. An extension towards hybrid optimisation of yield and power could include those regions and optimise them for yield.

Manufacturability of jogs is a concern. Therefore, [Leu03] suggests not to perform post routing methodologies that introduce jogs. A router that is power-aware and adjusts spacing on the run is clearly preferred. About 10 to 30% of

the jogs generated by *TopCool* can be removed in a post processing step, cf. Section 7.2.5. It is also possible to provide a higher thickness for the jogs to avoid manufacturing-related opens.

The current implementation requires much less jogs and thus much less wire-length than the commercial spreader. This is reflected by the higher yield-limit for missing material related opens, cf. Section 6.2.6. In addition, jog lengths are currently limited to a certain amount of routing tracks. Shorter rectangles are discarded. This does not harm power saving opportunities since too short boxes introduce too high extra wire overhead anyway.

For technical reasons, antenna rules and via overhangs have been turned off during all investigations in this work. The influence on power savings is irrelevant. For manufacturability and yield during fabrication, however, these must be considered. The consequences for the presented method are more rule consideration routines and handling of extra information.

7.1.3 Signal integrity

This subsection discusses crosstalk noise related signal integrity issues. Capacitive crosstalk originates on a victim wire when the signal value on a neighbouring aggressor wire transitions. Crosstalk noise is the superposition of the capacitive crosstalk of all aggressor wires that are coupling with the victim. Noise adds to the output signal voltage level. Finite output conductances of the driving gate renders the signal quality low even without noise. As a result, signal voltage levels deviating from the nominal values V_{DD} and gnd emerge and cause two undesired effects.

First, there is the effect on subthreshold current of the driven gates which is discussed in more detail in Section 7.2.4. Second, the probability of an unwanted logic transition increases if the static noise margin of a driven gate is violated. The result is additional glitch power in the best case. In the worst case, registering of false values or even false registering can occur.

The *TopCool* method generally places more wires away from each other than it places wires closer to each other. The signal integrity (SI) due to capacitive crosstalk noise is therefore expected to improve significantly. Quantitatively, the improvement can be seen in the first order expression for noise,

$$V_{\mathrm{noise}} \propto R_{\mathrm{drive}} \frac{C_C}{C_{\mathrm{net}}}, \tag{7.1}$$

with some technology and circuit specific proportionality constant. The relation

reflects the driving gate's limited ability to quickly countervail a cross coupled voltage pulse due to its on-resistance. At constant gate sizes, the noise can be decreased by reducing the coupling capacitance to total net capacitance ratio.

Not all nets require minimal coupling capacitances for safe noise levels. *TopCool* can be modified for minimising the probability of signal integrity related problems by taking the noise margins of the driven gates into account. The principle of optimising all boxes of parallel wires and selecting a maximum weight stable subset of boxes remains the same.

For this purpose, a proportionality constant that replaces the switching activity α can be introduced for every net. It combines R_{drive}, the driven gates' noise sensitivities, and technology-specific factors. The wire capacitance $C(d)$ becomes $C_{\text{lat}}(d)/C_{\text{net}}(d)$, the coupling capacitance over the total net capacitance as a function of distance. With these modifications, *TopCool* now reduces the risk of unwanted signal toggles per box.

More sophisticated noise models take the wire resistance, the net branching topology and the possible timing windows of the aggressors into account [CMS02]. An additional source of error is that the fraction of C_{net} that is present within the box is altered. The exact error value is unknown since the actual selection of boxes is not determined before the MWIS step, which in turn depends on the output of the spacing process. More research is required for finding the solution with minimal crosstalk noise related risk. Nevertheless, a power minimising method that is aware of the most crosstalk-critical nets can be implemented this way.

7.2 Improving power savings

The present work shows wire power savings of up to 9.55 %. There are several opportunities to further improve the method in this respect. Two general optimisation directions are pointed to by improvements to the method itself and by expansions of the method. This section provides subsections on modelling-related opportunities for low power (7.2.1 – 7.2.4), on problems (and possible solutions) inherent to the box method (7.2.5 – 7.2.6), and on structural modifications of the present approach needed to open up higher power savings (7.2.7 – 7.2.8). Section 7.2.9 briefly describes gate sizing in combination with *TopCool*.

7.2.1 Simultaneous switching power

Chapter 4 shows that the Miller effect on simultaneous switching wires during low power spacing can readily be respected. Capacitance between wires is split

into a lateral and a ground component with different weighting factors, c and α, respectively. The problem was found to be convex if no detouring wires are included in the objective function.

If the introduction of the c-factors into the equations results in tractable optimisation problems, has yet to be studied. The derivation of c is difficult. It can be surmised that simultaneous switching power is not reported in current gate level power analysis tools for this reason. In any case, the influence on accuracy of the power report is understood to be low, assuming that the design has not been optimised for low simultaneous switching power, cf. Section 6.3.

From switching activity propagation alone, it is not possible to tell how many transitions of two nets are uniform, opposed, or stand-alone. Time-allowing, a gate level simulation of the design can be performed. Tracing all signals, however, entails a data volume problem, considering for example hundreds of thousands of signal values at millions of clock cycles. In order to save space and processing time, the following recipe can be used:

1. Find all capacitive coupling pairs in the layout.
2. Perform static timing analysis for all coupling nets. Sort out all coupling pairs without overlapping timing transition windows. Current logic synthesis tools already support items one and two.
3. Simulate the design on gate level. Only trace value changes on the net pairs with overlapping timing windows and evaluate signal traces to derive c.

For the cases where two wires that switch simultaneously within one clock period and are correlated, c is now derived exactly—and not over- or underestimated by $\alpha_1 + \alpha_2$. The spacing routine is now guided by an accurate value for c, and this leads to superior results. The improved spacing results, however, are not expected to pay off directly. Important is the increase in prediction accuracy so that the selection heuristic does not select boxes that are just apparently better.

7.2.2 Capacitance model

The capacitance model used, $C(d) = C_\infty + s \cdot d^e$, is accurate if the routing densities on the adjacent layers are constant. This is a simplification. Figure 7.1 depicts how the capacitance reduction through spacing diminishes more quickly with more populated layers. Fringing capacitances exhibit a higher fraction of the total capacitance in that case.

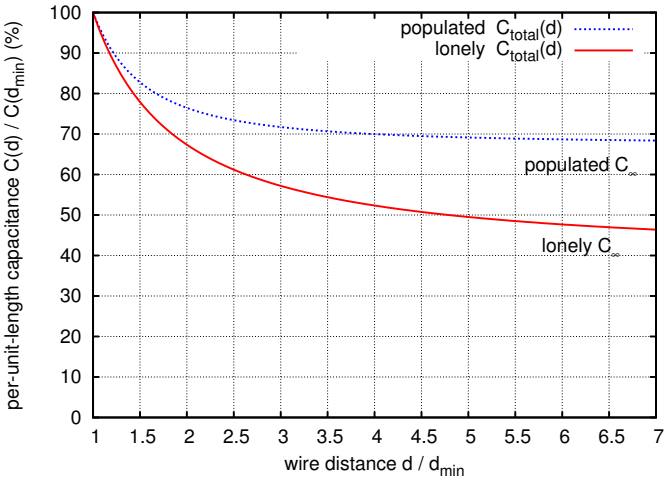


Figure 7.1: Capacitance between two wires on layer M3 of a 130 nm process. M1 and M5 are ground plates. In the case "lonely", M2 and M4 are empty, in case "populated" the routing density is 100 %.

Improving the power saving results is possible through a more accurate capacitance model. It requires the equation to take into account the metal fill rates in the areas above and below the current box. The model is best created with a series of experiments and interpolation or fitting.

7.2.3 Short-circuit power

It is widely known [WE93, Yea98, Rab03] that decreasing the load capacitance increases the short circuit power of the driving gate. The less capacitive a node gets, the less it attracts the current from V_{DD} during a transition of the input voltage of a driving gate. Excess current flows straight through the NMOS transistor to ground.

Yeap notes that the capacitive power saving always exceeds the increase of the short circuit power [Yea98] when reducing the capacitive load. A secondary effect, however, is often overlooked. At constant gate size, reducing the capacitance improves the slope of the signal. The time frame decreases in which both PMOS and NMOS transistors of the *driven* gates are in their on-regions. As a result, the short circuit currents decrease linearly with transition time in the first order [WE93].

Decreasing C is therefore always good practise in terms of power consumption.

First investigations show that the internal energy of the cells of the entire chip always decreases after applying the present method as is. Depending on circuit, one to ten percent internal energy is saved. The method may be tuned to optimise for the total dynamic power consumption as opposed to the capacitive power consumption of the wires alone.

A model of the influence of the capacitance on power consumption is required to consider the effects described. Taking the first derivatives of the short-circuit power types with respect to C models a first order sensitivity s of P to C.

$$\begin{aligned} s_1 &:= \frac{\partial P_{\text{cap}}}{\partial C} = \alpha f V_{\text{DD}}^2 \\ s_2 &:= \frac{\partial P_{\text{int, driver}}}{\partial C} \propto \alpha \\ s_3 &:= \frac{\partial P_{\text{int, driven}}}{\partial C} \propto \alpha \end{aligned}$$

Note that s_2 is negative, $\Sigma s_i > 0$, and s_i are proportional to α. In order to optimise the total dynamic power objective, α in the objective function is replaced by a linear combination of s_i. In particular, the constants γ_2 and γ_3 in $\alpha \leftarrow \alpha + \gamma_2 \cdot s_2 + \gamma_3 \cdot s_3$ are required to account for the fraction of a node capacitance that is present inside the current box.

The sensitivity s_1 can be expressed in closed form (7.2) while s_2 and s_3 of a node can be measured. To do so, an incremental capacitive load ∂C is subtracted from the load of the node. Then, a power analysis for the driven gates of the node measures $\partial P_{\text{int, driven}}$. An additional power analysis of the driving gate measures $\partial P_{\text{int, driver}}$. This usually consists of a simple table look-up in the characterisation library.

7.2.4 Crosstalk noise power

Leakage power accounts for an ever higher fraction of the total power consumption. Crosstalk noise voltage, cf. Section 7.1.3, influences the voltage level of a node and thus the static power consumption of the driven gate, even if the node exhibits no nominal activity itself. Subthreshold currents depend exponentially on gate–source voltages below the threshold voltage. Thus, low noise is highly desirable for low static power.

The present method can be extended to integrate crosstalk noise power by minimising Equation (7.1). As in the previous subsection a sensitivity measure,

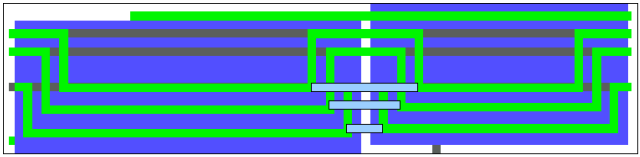


Figure 7.2: The wire topology between two selected boxes (dark) can be optimised as shown. The intended positions of the indicated bridges must be verified for enough whitespace. About ten to thirty percent of all jogs introduced are candidates for removal.

$$s_4 \quad := \quad \frac{\partial P_{\text{sub, driven}}\left(V_{\text{noise}}(C)\right)}{\partial C},$$

is required that measures the first order dependence of the static power consumption of the driven gates on the capacitance in a given box.

In order to account for this effect, the cells must be properly characterised. The current industry-standard to model leakage power in standard cell designs provides default values for every gate and every logic state, rather than quasi-continuous look-up table techniques. At time of this writing, new standards have yet to emerge [Syn06].

7.2.5 Cross-box optimisation

The MWIS heuristic selects a subset of boxes. However, the wire topology is optimised within each box only but not with respect to the surrounding of the box. Two factors are distinguished: firstly, there is coupling of the outermost wires of two boxes adjacent on the preferred routing direction side. Secondly, unnecessary clips may emerge if two boxes are adjacent by the side perpendicular to the preferred routing direction. In other words, one and the same wire that leaves a box and enters the next box is detoured twice, cf. Figure 7.2.

The first issue can be addressed by a post-processing step after rectangle selection. The spacing step is repeated for the selected rectangles with known border conditions. This is an iterative process, as these border conditions are altered again during re-spacing. Assuming only small adaptions to be made, the quality of the selected subset will not decrease and their spacing solutions will improve slightly. Currently, *TopCool* is executed twice.

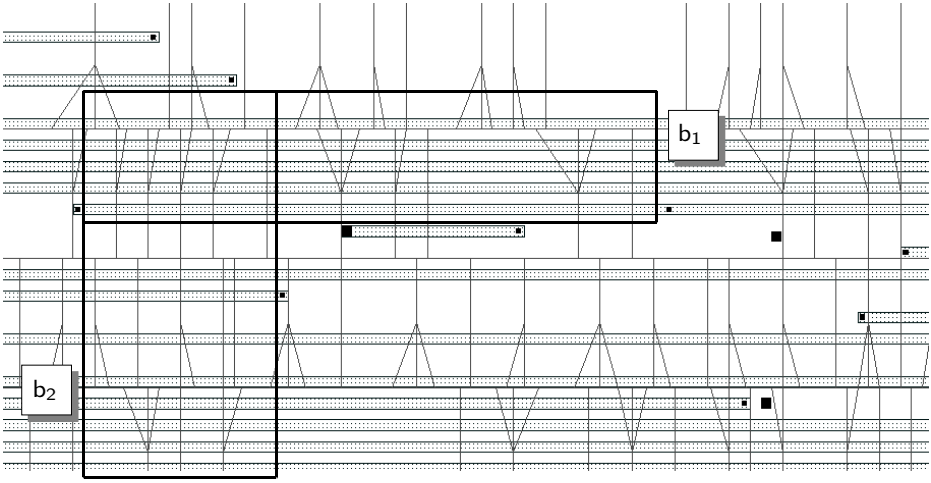


Figure 7.3: Selecting a box may let valuable optimisation potential lie idle. Splitting the boxes into their elementary areas captures the whole potential.

Removal of excessive clips (jogs) is technically more difficult than it may seem. It can also be addressed in a post processing step. One difficulty is that the layout of the circuit changes as jogs are removed. Sophisticated data structures are required to handle that task in tolerable runtimes. Guidelines are pointed out in [Iln06]. The main benefit of jog removal is manufacturability improvement. Power and timing benefits are expected to be of secondary importance.

7.2.6 Rectangle splitting

Figure 7.3 shows one inherent limitation of the box approach. Only one of two overlapping boxes b_i and b_j can be selected. The area $(b_i \cup b_j) \backslash b_i$ will remain unoptimised if b_i is selected and no other boxes covering that area exist.

It may be favourable to add the areas $(b_i \cup b_j) \backslash b_i, \forall i, j$ as individual boxes, perform the spacing operations therein, and fully consider them during the MWIS step. However, the number of boxes will increase by the order of their overlaps. This is directly reflected in memory consumption and runtime. The area graph increases significantly. Further research is required if this increase can be adequately handled by the MWIS heuristic.

An indication of the possible additional savings is given by running the method

on its own output. This also addresses the inter-box problem of Section 7.2.5. About 20% additional savings are achieved.

7.2.7 Wire permutation

Wire permutation in combination with wire spacing saves more energy than wire spacing alone, as quantified in Section 6.4 and in [ZGRS05], and therefore deserves a closer look. Unfortunately, wire permutation cannot be executed on detail-routed layouts, as opposed to wire spacing alone. It would require the routing layers above or below a box of parallel wires to exhibit enough whitespace to route a permutation network [MMP01]. One could think of two ways to solve this problem. However, both need more research.

Global router takes wire permutation into account

Integrated circuit routing generally happens on two or more coarseness levels. During global routing, net segments are assigned to buckets that span a coarse grid over the whole chip. Bucket sizes vary among implementations and chips; a typical value is 100×100 buckets for a medium to large chip.

The idea is to build buckets that contain wires with similar activities. Buckets with low activity wires should be densely filled, and buckets with high activity wires should be sparsely filled. This approximates the unimodal optimal wire order on a coarse level. Then, the buckets are detail-routed. As a result, subsequent wire spacing in boxes contained in high-activity buckets takes full advantage of optimal capacitance sharing.

Change only few wires

Detouring a wire across another layer costs four vias, two on every side of the box to change the layer for the detouring wire, in addition to the detouring wire itself. One idea is to route the permutation network only for very few, possibly only one wire. The resulting wire order is better than the original order. The selection of which wire to change the position for should follow the criteria below:

1. It must be verified that only non-power critical wires are detoured. Otherwise, the parasitic capacitances of the permutation network may reduce or void the wire spacing returns.

2. Timing critical wires may not be detoured because four extra vias are added. Resistance and capacitance overhead cannot be tolerated on timing critical nets.

3. There must be enough routing space above or below. This is not trivial to ensure. Even if there is enough whitespace to place a new wire, it must also be checked against sufficient capacitive coupling headroom.

A quick experiment can evaluate the pay-off of this method. The first step is finding a candidate wire $i \in \{1..N\}$ that is neither power- nor timing critical and that can be detoured on a layer above or below. The switching activity of wire i is removed and inserted at position j to build a new order, for instance for $j = 2$: $(\alpha_1, \alpha_i, \alpha_2, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_N)$. Wire-spacing as usual now delivers a different power saving for every j. The solutions should be ordered by power saving to then check for space for the best solution as possible.

7.2.8 Vias

The present low power post routing design methodology allows for lateral movements of wire pieces that are not connected to vias. Moving vias, however, enables more ample modifications to wire topologies. Thus, the solution space increases and less power consumption can be expected. Instead of creating jogs to route a wire around an obstacle, the obstacle can be pushed to create space for the wire.

A more complex optimisation model than that of boxes of parallel wires is developed from this conclusion. It uses fields and forces[1] to model a power gradient in whose direction to move wires and vias. However, the possibility to split a wire into wire pieces at non-movable obstacles as inherently supported by the box method, is still missing. In the following, two kinds of forces are identified. Then, implementation issues are discussed.

Field force

Assumed is a system of parallel wires, like a layer with a preferred routing direction. The first task at hand is to minimise the switched lateral coupling capacitances of the wires. As opposed to the box method, there is no limitation that wire segments must be of equal length. In particular, entire wires are allowed to move.

[1] Modelling optimisation criteria as fields and forces is inspired by placement algorithms developed at the *Institute for Electronic Design Automation, TU München* [EJ98, OJ04, ORJ05].

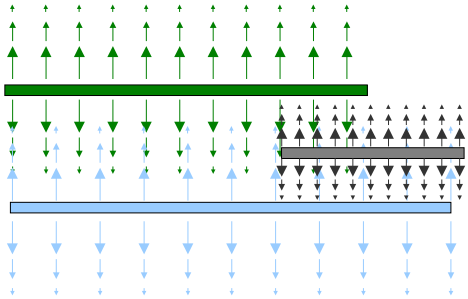


Figure 7.4: Illustration of field forces. Every wire is assumed to carry a hypothetical load proportional to its switching activity. The wires repel each other in the resulting field.

Each wire is assumed to carry a hypothetical electric charge q proportional to its switching activity α. The polarity of all charges is the same. The sum of the charges determine an electric field. Parallel wires repel each other in this field. By letting the wires move into the direction of the repelling force, the whole system converges to a power-minimal state, cf. Figure 7.4.

Spring force

A wire movement, however, is only possible if the wire endings maintain electrical connection. This means that either connecting branch lines are routed or that the vias attached to the wire move as well. Vias connect perpendicular running branch lines on adjacent layers. If a via that connects a wire end to a branch line on a layer below or above moves as well, either slides on the line below or above, prolongs it, or shortens it.

Contraction spring forces are introduced in order to account for a change in the lengths of the branch lines. See Figure 7.5 for an illustration. Length influences coupling and ground capacitance and thus power linearly. The higher the switching activity of a wire, the shorter it should get. Furthermore, branch lines couple to neighbouring wires themselves. Thus, the contraction constant of the spring correlates both with the switching activity and with the hypothetical electrical field (see above) in the layer below or above.

Figure 7.6 provides an example of how the topology can be changed by introducing the spring force. The optimised position for the vertical segment of the net connecting the two end nodes is above the other vertical wire. Starting from the

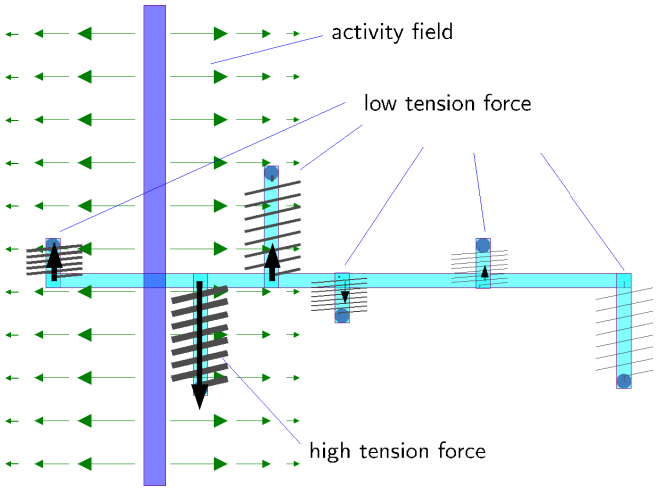


Figure 7.5: Illustration to clarify the spring force. Considered is a movement of the long horizontal line. The vertical branch wire close to the long vertical wire gets more contracted than the other branch wires. The horizontal segment moves downwards to decrease the total switched capacitance.

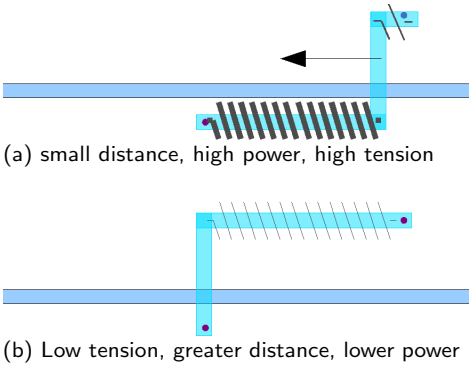


Figure 7.6: Simple example to show the benefits of the additional spring force.

original position (a) and using only the field forces, the wire would move further towards south. With the second force, however, the lower horizontal segment receives a high contraction which is transmitted via the vertical segment to the upper horizontal segment. The spring force of the latter is less since it is farther away from the long horizontal wire segment. The layout topology converges to its power-optimised state (b).

Implementation

All forces are normalised to the dimension power. A wire displacement not only causes a change in coupling capacitance but also a change in the lengths of perpendicular branch wires. The gradient or difference quotient measures the gain or cost of a wire displacement.

Electronic design automation generally demands $O(N \cdot \log N)$ runtimes considering object counts that reach billions. It is currently unclear if simply descending the gradient by selecting the best of all moves leads to a global minimum. In addition, a good strategy to break the wires must be found.

An implementation of the present concept is therefore best approached by starting from the existing *TopCool* method. Allowing the boxes to contain movable vias combines the advantages of both approaches.

7.2.9 Post routing gate sizing

Slack, a measure for how much breathing room exists for timing, can be traded for smaller and less powerful driving transistors. This process is called "gate down-sizing." Standard cell designs allow for a change in drive strengths of individual gates before and after layout. Intel suggested to combine their method with gate sizing in [MKWS04]. The goal is not to save area. Area is fixed after placement. The goal is to save gate-internal energy that generally decreases with size.

TopCool modifies the wire capacitances and thus also the path delays. In most cases, especially for active nodes, path delays are decreased. This creates positive slack. As especially the active gates can be down-sized, the effect on total internal energy consumption is expected to be high.

After routing, the circuit timing is met exactly only in seldom cases. A gate sizing step tries to meet the original constraints and the results can hardly be attributed to the delay reductions by *TopCool* alone. Experiments on gate sizing should therefore comprise gate sizing before and after the application of *TopCool*, cf. [Bah06].

7.3 Impact of technology scaling

The pace of technological development deserves a closer look. Wires scale differently than transistors from technology generation to technology generation. Only the width and minimum distance of the wires scale at the same rate as the transistors in order to take full advantage from increased integration densities. In the following, the influence of technology scaling on the presented method is examined.

7.3.1 Wire switching power fraction

The wire length cannot scale according to the same rule as transistors. Following Rabaey [Rab03], one can distinguish local, constant-length and global wires. Only local interconnect length decreases over the years, while the length and number of global wires increases with chip size. More and more capacitive load moves from the transistors to the wires. As the present method minimises wire switching power, optimisation potential will be increasingly present in the future.

The fraction of total power consumption caused by static power dissipation has been growing during the last decades, though. This means that the method is more effective with low-leakage processes and not during standby times of the circuit. In general, the static leakage current does not exceed one third of the dynamic power consumption during active operation [Rab05]. Static and dynamic power consumption can always be balanced against each other by the selection of V_{DD} and the threshold voltage $|V_{\mathrm{t}}|$ [Yea98]. Thus, a method that reduces any of static or dynamic power consumption can be used to optimally decrease the total power. In particular, *TopCool* can be used even if the static power consumption should dominate in the future.

7.3.2 Wire aspect ratio

Wire resistances increase as a result of decreased wire widths. To compensate for the consequent rise in *RC*-delay, wire thicknesses are not scaled at the same rate as wire widths. The industry thereby accepts the increased relative lateral coupling.

Also from the viewpoint of ever increasing wire aspect ratios, one can argument well for the effectiveness of the presented method. High aspect ratios cause high lateral coupling at minimum distance. Due to the roughly inverse proportionality of the coupling capacitances on the distance, spacing starting at d_{min} will reduce

the lateral capacitance more quickly than it will increase the ground capacitance in the future.

7.3.3 Dielectric

In order to compensate for increased wire delays over the technology generations, the industry endeavours research for new low-permittivity materials between the wires. Success was met with [ITR07] by introducing new inter layer dielectric (ILD) materials with $\varepsilon_r = 3.9$, as opposed to silicon dioxide ($\varepsilon_r = 4.1$) which is still used for the inter-metal dielectric (IMD) for processing reasons. It is expected that materials with decreasing ε_r will continue to establish themselves for the ILD first. As a result, distancing wires as proposed in this thesis will continue to reduce the total wire capacitance effectively, because the influence of the ground capacitance decreases.

7.3.4 Process variations

Process variations are a serious concern. Physical wire properties deviate from their nominal values and therefore their electrical properties as well. Lithography, etching, polishing and alignment inaccuracies mainly cause chip-to-chip, wafer-to-wafer or lot-to-lot variations. But with ever shrinking geometries, also on chip variations get hard to control.

A variation-sensitive approach highly depends on the availability of statistical data. Even provided that the industry measures, publishes and keeps the information up to date, still investigations must be done on how to deal with it.

7.3.5 X-Architecture

The presented method is readily applicable to technologies with 45 degree routing capabilities. The routing concept of a preferred routing direction on every layer is still present. In addition, diagonal routes can be used for the detouring branch lines.

8 Conclusions

Although the literature has been aware of the increasing influence of wire capacitances on power dissipation of digital CMOS circuits, the number of publications dedicated to optimising the wire topology for less power is low. Yet, commercial power aware routers do not exist.

In this thesis, a method named *TopCool* was introduced that reduces the wire power consumption of detail-routed circuits and seamlessly taps into an industry-strength design flow. It is split into three major parts—searching, spacing, and selecting groups of parallel wires. The actual power reduction is performed in the spacing step. Integer-convex optimisation is leveraged to carry out activity-driven wire spacing in every group of parallel wires in the design. The entire method is shown to exhibit quasilinear runtime complexity. No area investment is required, since remaining whitespace after detailed routing is exploited.

The results show several advantages of this method over others. Methods that globally increase the routing pitch can fail due to area limitations. Other approaches that do not include the switching activities achieve only half of the power savings.

TopCool reduces the interconnect switching power by 5 %–9 %. The presented method achieves the highest average power reductions of all known approaches over a set of benchmark circuits such as multiprocessor systems if the circuits are reasonably – that is, tightly – constrained.

The only currently known limitation of the method itself is the required trade-off between yield and low power. *TopCool* improves the yield-limit of particle-related shorts but does not achieve the same value as commercial wire spreading that is specially optimised for this task, and that cannot be executed in combination with *TopCool*. Timing is not adversely affected. *TopCool* requires switching activities which are difficult to approximate. This is a known problem in literature and not of *TopCool* itself.

CPU runtimes are several times smaller than those of other approaches. About three minutes are required to reduce the wire power consumption of a one million wire design by five milliwatts (8 %) on a current server. A positive energy balance is obtained in this example if one optimised circuit is operated for 1,000 hours or

1,000 such circuits for one hour each, assuming 100 W power consumption of the design server during optimisation.

Fractions of the wire power to the total power were determined to up to 33 %, depending on benchmark circuit. This translates into an overall chip-wide power saving of up to about 2.5 % only because of the reduced wire power. This value can be expected to increase as the ratio of the wire power to the total power increases with circuit size, and with technology generation. Both the circuit sizes (up to one million wires) and the technology node (130 nm) used in the experiments range at the low end of today's possibilities. In addition, the internal energy and the leakage current are reduced as well, and this effect has not been included, yet. With further optimisations, larger circuits, and more modern technologies, 4 %–5 % power reduction seem realistic.

Yet the dissertation points out further power saving opportunities. A theoretical part proves the existence of an optimal wire order for low power CMOS, leveraging significantly more optimisation potential than wire spacing alone. In order to take full advantage of this observation, however, optimisations should be performed already during routing.

Generally, the method of power saving by distance optimisation should be an integral part of physical EDA tools to optimally interact with tasks like routing, design rule checking or timing verification. Nonetheless, a standalone version was shown that works very well in cooperation with existing commercial physical design automation tools.

A Proof of Theorem 1

This proof is the result of the cooperation with Peter Gritzmann and Michael Ritter and is stated from [ZGRS05]. An even more elegant version is in preparation [GRSZ07]. The capacitance model (2.8) on page (18) used $d^{e_{\text{tot}}}$ and a constant $e_{\text{tot}} < 0$ to model the convexity of the total capacitance per unit length of two wires at distance d. In this proof, a slightly less general case, $e_{\text{tot}} = -1$ is assumed.

In what is to follow, $\mathbf{u_n}$ denotes the n-th unit vector and $\mathbf{e}$ the all-ones vector $(1,\ldots,1)^T$.

A.1 Known results from convex programming

In the first instance, a basic result from convex programming is summarised.

Theorem 2 *Let $P \subset \mathbb{R}^n$ be a closed convex subset of the open convex set S and let $f : S \to \mathbb{R}$ be a convex differentiable function. Then $\mathbf{x}^* \in P$ minimises f over P if and only if*

$$-\nabla f(\mathbf{x}^*) \in N_P(\mathbf{x}^*),$$

where $N_P(\mathbf{x}^*)$ is the cone of the outer normals in $\mathbf{x}^*$, defined by

$$N_P(\mathbf{x}^*) := \{\mathbf{c} \in \mathbb{R}^n : \max_{\mathbf{x}\in P} \mathbf{c}^T\mathbf{x} = \mathbf{c}^T\mathbf{x}^*\}.$$

For the proof of this theorem and some background on convex programming, confer [Roc72], theorem 27.4. The next lemma is a simple inequality of concave functions, which will be essential for the proof of the claim.

Lemma 1 *Let $a \geq b \geq 0$ and let $c > 0$. Then $\sqrt{a} - \sqrt{b} \geq \sqrt{a+c} - \sqrt{b+c}$ with equality if and only if $a = b$.*

Proof 1 *This follows straightforward from the fact that the function $x \mapsto \sqrt{x}$ is differentiable in $(0,\infty)$ and has strictly decreasing slope.* □

A.2 Characterisation of the optimal distances

In the following, the optimal distance vector $\mathbf{d}$ of (4.5–4.6) for a given wire ordering is characterised. For ease of notation, γ_n shall denote $\alpha_n + \alpha_{n-1}$ in the sequel.

Theorem 3 *Let $P := \{\mathbf{d} \in \mathbb{R}^{N+1} : \mathbf{d} \geq d_{\min}\mathbf{e} \wedge \sum_{n=1}^{N+1} d_n \leq \beta\}$. Then $\mathbf{d}^* \in P$ is optimal for (4.5–4.6) if and only if there exists $\tilde{\gamma} > 0$ that satisfies both*

$$d_n^* = \max\{\tilde{\gamma}\sqrt{\gamma_n}, d_{\min}\} \quad \forall n = 1, \ldots N+1 \tag{A.1}$$

$$\text{and} \sum_{n=1}^{N+1} d_n^* = \beta. \tag{A.2}$$

Proof 2 *Let $\mathbf{d}^* \in P$ and $\tilde{\gamma}$ as required by the theorem. It has to be shown that $-\nabla f(\mathbf{d}^*) \in N_P(\mathbf{d}^*)$. For this purpose let I denote the index set $I := \{n : d_n^* = d_{\min}\}$. Required are $\lambda_0, \lambda_1, \ldots, \lambda_{N+1} \geq 0$ such that*

$$-\nabla f(\mathbf{d}^*) = \begin{pmatrix} \frac{\gamma_1}{{d_1^*}^2} \\ \vdots \\ \frac{\gamma_{N+1}}{{d_{N+1}^*}^2} \end{pmatrix} = \sum_{n \in I} \lambda_n(-\mathbf{u_n}) + \lambda_0 \mathbf{e}.$$

One can easily calculate $\lambda_0 = \frac{1}{\tilde{\gamma}^2}$ and $\lambda_n = \frac{1}{\tilde{\gamma}^2} - \frac{\gamma_n}{d_{\min}^2}$ for $n \in I$ by inserting the expression for d_n of (A.1) into the gradient components. As $\tilde{\gamma} \leq \frac{d_{\min}}{\sqrt{\gamma_n}}$ for $n \in I$, cf. (A.1), the λ_n are all nonnegative, so $-\nabla f(\mathbf{d}^*) \in N_P(\mathbf{d}^*)$ and $\mathbf{d}^*$ is optimal by Theorem 2.

For the converse, suppose $\mathbf{d}^* \in P$ is an optimum for (4.5–4.6). Let $m := \max\{\frac{\gamma_n}{{d_n^*}^2} : n = 1, \ldots, N+1\}$ and $G := \{n : \frac{\gamma_n}{{d_n^*}^2} = m\}$. According to Theorem 2, $-\nabla f(\mathbf{d}^*) \in N_P(\mathbf{d}^*)$, hence there are $\lambda_0, \lambda_1, \ldots, \lambda_{N+1} \geq 0$ such that $-\nabla f(\mathbf{d}^*) = \sum_{n=1}^{N+1} \lambda_n(-u_n) + \lambda_0 e$. Of course $\nabla f(\mathbf{d}^*) \neq \mathbf{0}$ (because $\exists i : \gamma_i \neq 0$), so λ_0 must attain some value $\geq m$ (note this implies that $\mathbf{d}^*$ is on the hyperplane $\mathbf{e}^T\mathbf{d} = \beta$, hence $\sum_{n=1}^{N+1} d_n^* = \beta$, cf. (A.2)) and $\lambda_n = \lambda_0 - \frac{\gamma_n}{{d_n^*}^2}$ for $n = 1, \ldots, N+1$. In case $\lambda_n > 0$ for all $n = 1, \ldots, N+1$, the vector $\mathbf{d}^*$ would be determined by the intersection of $N+2$ hyperplanes with normal vectors $-\mathbf{u_1}, \ldots, -\mathbf{u_{N+1}}$ and $\mathbf{e}$, which is clearly impossible as $M > N$. So at least one λ_n must be 0, and this can only be the case if $\lambda_0 = m$, which means $\lambda_n = 0 \iff n \in G$. So for $n \in G$ there are $d_n^* = \frac{\sqrt{\gamma_n}}{\sqrt{m}} = \tilde{\gamma}\sqrt{\gamma_n}$ with $\tilde{\gamma} := \frac{1}{\sqrt{m}}$, whereas for $n \notin G$ the value of d_n^* is determined by the intersection of the hyperplanes $-{\mathbf{u_n}}^T\mathbf{d} = d_{\min}$ with

$\mathbf{e}^T\mathbf{d} = \beta$, *therefore* $d_n^* = d_{\min}$ *for all* $n \notin G$. *For* $i \notin G$ *and* $j \in G$ *the inequality* $\frac{\gamma_i}{d_{\min}^2} < \frac{\gamma_j}{(d_j^*)^2} = \frac{1}{\tilde{\gamma}^2}$ *holds, so* $\mathbf{d}^*$ *is of the form stated above.* □

The following considerations are valid if the optimal wire spacing $\mathbf{d}^*$ is of the form $d_n^* = \tilde{\gamma}\sqrt{\alpha_n + \alpha_{n-1}}$. If one or more distances are at their lower bound, further steps are required, but the result is basically the same. A complete proof will be published by Michael Ritter. So the objective function (4.5) is reduced to

$$\tilde{\gamma} \sum_{n=1}^{N+1} \sqrt{\alpha_n + \alpha_{n-1}} = \min!$$

A.3 Power-optimal wire ordering

This section provides several "building blocks" and the proof of the overall claim. The basic idea is to make use of the inductive nature of the proposed algorithm. For this purpose, the notion of a *unimodal wire ordering* is formalised.

Definition 2 *Let* $(q_n)_{n=0,\ldots,N+1} \in \mathbb{A}$ *be a wire ordering of the* (α_i). *If there exists an index* $t, 1 < t < N+1$, *such that* $q_{n-1} \leq q_n \;\forall\; n \leq t$ *and* $q_n \geq q_{n+1} \;\forall\; n \geq t$, *the wire ordering* (q_n) *is called a unimodal wire ordering with mode* t.

The next two theorems provide the key ideas for the proof of Theorem 1. To avoid some technical details, only the case is proved, when all elements of the set of activities are pairwise distinct. Similar arguments can be applied for the general case, but some special instances must be taken care of.

Theorem 4 *A power-optimal wire ordering* $(q_n)_{n=0,\ldots,N+1} \in \mathbb{A}$ *is unimodal.*

Proof 3 *To see the unimodality, assume the existence of a wire ordering* (q_n) *minimising (4.5–4.6) that is not unimodal. Then there exists an index* $1 < t < n$ *such that* $q_{t-1} > q_t < q_{t+1}$. *The smallest possible* t *with that property is examined. Without loss of generality,* $q_{t-1} \leq q_{t+1}$ *is assumed. Let* (p_n) *be the sequence defined by*

$$p_n := \begin{cases} q_n, & \text{for } n \neq t-1, t, t+1 \\ q_t, & \text{for } n = t-1 \\ q_{t-1}, & \text{for } n = t \\ q_{t+1}, & \text{for } n = t+1. \end{cases}$$

Then the objective function for (p_n) differs from that of (q_n) by

$$\begin{aligned}
&\sqrt{q_t+q_{t-2}}+\sqrt{q_{t-1}+q_t}+\sqrt{q_{t+1}+q_{t-1}}\\
&-\sqrt{q_{t-1}+q_{t-2}}-\sqrt{q_t+q_{t-1}}-\sqrt{q_{t+1}+q_t}\\
=&\sqrt{q_{t-1}+q_{t-2}-(q_{t-1}-q_t)}-\sqrt{q_{t+1}+q_{t-1}-(q_{t-1}-q_t)}\\
&-(\sqrt{q_{t-1}+q_{t-2}}-\sqrt{q_{t+1}+q_{t-1}})
\end{aligned}$$

Applying Lemma 1 to $q_{t+1}+q_{t-1}\geq q_{t-1}+q_{t-2}>0$ and $q_{t-1}-q_t>0$ shows that the objective for (p_n) is less than for (q_n), an obvious contradiction.

Theorem 5 *Again, let $(q_n)_{n=0,\dots,N+1}\in\mathbb{A}$ be a power-optimal wire ordering. Furthermore, let $q_t\geq q_s\geq q_r$ denote the three greatest elements of (q_n). These elements must be chosen such that one of them is adjacent to both of the others; if $q_t>q_s,q_r$, then q_t is located between q_r and q_s, i.e. either $r=t-1\wedge s=t+1$ or $r=t+1\wedge s=t-1$.*

Proof 4 *For this claim, again assume that (q_i) is optimal with maximal element q_t and q_s, q_r as defined in the theorem. Suppose $q_t>q_r,q_s$ is not located between q_s and q_r. Then, due to unimodality both q_s and q_r have to be on the same side of q_t. Assuming w.l.o.g. that $s,r>t$, therefore $q_{t-1}\leq q_r\leq q_s\leq q_t$. Also due to unimodality, $s=t+1$, $r=t+2$ (there can be no smaller element between them, because q_t is the unique mode of the sequence). Now the sequence can be reordered by changing the places of q_t and q_s without destroying unimodality, hence (p_n) is defined to*

$$p_n:=\begin{cases}q_n, & \text{for } n\neq t,t+1\\ q_{t+1}, & \text{for } n=t\\ q_t, & \text{for } n=t+1.\end{cases}$$

Then

$$\begin{aligned}
&\sum_{n=1}^{N+1}\sqrt{q_n+q_{n-1}}\leq\sum_{n=1}^{N+1}\sqrt{p_n+p_{n-1}}\\
\iff&\sqrt{q_t+q_{t-1}}+\sqrt{q_{t+1}+q_t}+\sqrt{q_{t+2}+q_{t+1}}\\
&\leq\sqrt{p_t+p_{t-1}}+\sqrt{p_{t+1}+p_t}+\sqrt{p_{t+2}+p_{t+1}}\\
\iff&\sqrt{q_{t+2}+q_t-(q_t-q_{t+1})}-\sqrt{q_t+q_{t-1}-(q_t-q_{t+1})}\\
&\leq\sqrt{q_{t+2}+q_t}-\sqrt{q_t+q_{t-1}},
\end{aligned}$$

and the same argument as above contradicts the assumption. Consequently, q_s and q_r both have to be adjacent to the maximal element q_t. □

The following theorem proofs that the wires with the three highest activities must always settle in the middle. It will be used as induction step to proof the overall claim of Theorem 6.

Theorem 6 *A wire ordering (q_n) for a problem of size $N+1$ is optimal if and only if the sequence (q'_n) defined by removing a maximal element from (q_n) is an optimal wire ordering for the reduced problem of size N.*

Proof 5 *First, let (q_n) minimise the sum $v := \sum_{n=1}^{N+1} \sqrt{q_n + q_{n-1}}$ and let c be the maximal element of the wire ordering, a and b the two elements next in size which are both adjacent to c by Theorems 4–5. Then removing c defines a wire ordering (q'_n) with objective value $v' = v - \sqrt{a+c} - \sqrt{c+b} + \sqrt{a+b}$. Suppose there is a wire ordering (p'_n) with objective value $w' < v'$. The elements a and b are the two greatest elements of (p'_n), therefore they have to be adjacent and a sequence (p_i) can be defined by inserting c between a and b. The objective value of (p_n) is $w = w' - \sqrt{a+b} + \sqrt{a+c} + \sqrt{c+b}$, so $w < v$, contradicting the optimality of (q_n).*

To see the other direction, let (q_n) be some wire ordering of length $N+1$ with objective value v, greatest element c and adjacent elements a and b, such that (q'_n) defined by removing c from (q_n) minimises $v' = \sum_{n=1}^{N} \sqrt{q'_n + q'_{n-1}}$. Suppose (q_n) is not the optimal wire ordering for length $N+1$, then there exists a sequence (p_n) with objective value $w < v$, and (p'_n) can be defined with objective value w' by removing c from (p_i). The optimality of (q'_n) is contradicted, considering that $w = w' + \sqrt{a+c} + \sqrt{c+b} - \sqrt{a+b}$ and $v = v' + \sqrt{a+c} + \sqrt{c+b} - \sqrt{a+b}$, so $w' < v'$. □

The construction provided in Theorem 1 simply formalises the induction step given in Theorem 6. It can now be proofed by induction. The induction step has already been proofed above, and the induction basis assumes to begin with a set of three wires:

Proof 6 (of Theorem 1) *The overall proof proceeds by induction. The algorithm given in Theorem 1 mimics exactly the statement of Theorem 6, so the induction step is clear. For the induction basis, the case of $N = 3$ wires is investigated. Assuming activities $0 < e \leq f \leq g$, the ordering arising from the construction is either $(0,e,g,f,0)$ or $(0,f,g,e,0)$, depending on whether f is inserted to the*

right or to the left of e. Theorem 4 has shown that the optimal solution has to be unimodal with mode g, and that e and f have to be adjacent to g. Thus there are no other solutions than the constructed sequences. In addition, the objective values of the two possible solutions are equal, so both are optimal. □

Bibliography

[AAN03] Ravishankar Arunachalam, Emrah Acar, and Sani R. Nassif. Optimal shielding/spacing metrics for low power design. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, page 167. IEEE Computer Society, 2003.

[All06] Gerard A. Allan. *EYES: User Manual*. Predictions Software Ltd., http://www.icyield.com, 2006.

[Bah06] Othman Bahlous. *Evaluierung von Verlustleistungsoptimierenden Verdrahtungsverfahren*. LIS Diploma Thesis, TU München, Germany, August 2006.

[BLR04] Sanjukta Bhanja, Karthikeyan Lingasubramanian, and N. Ranganathan. A stimulus-free graphical probabilistic switching model for sequential circuits using dynamic bayesian networks. In *Proceedings of the 41st annual conference on Design automation (DAC)*, pages 773–796. ACM Press, 2004.

[Bus06] Stanislav Busygin. A new trust region technique for the maximum weight clique problem. *Discrete Applied Mathematics*, 154(15):2080–2096, 2006.

[CHK+05] Yongseok Cheon, Pei-Hsin Ho, Andrew B. Kahng, Sherief Reda, and Qinke Wang. Power-aware placement. In *Proceedings of the 42nd annual conference on Design automation (DAC)*, pages 795–800. ACM Press, 2005.

[CI92] Y. L. Le Coz and R. B. Iverson. A stochastic algorithm for high speed capacitance extraction in integrated circuits. *Solid-State Electronics*, 35:1005–1012, July 1992.

[CKP01] Jasong Cong, Cheng-Kok Koh, and Zhigang Pan. Interconnect sizing and spacing with consideration of coupling capacitance. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 6:1164–1169, 2001.

[CMS02] L. Chen and M. Marek-Sadowska. Closed-form crosstalk noise metrics for physical design applications. In *Proceedings of the conference on Design, automation and test in Europe (DATE)*, page 812. IEEE Computer Society, 2002.

[CRS00] Fulvio Corno, Matteo Sonza Reorda, and Giovanni Squillero. RT-level ITC'99 benchmarks and first ATPG results. *Design & Test of Computers, IEEE*, 17(3):44–53, 2000.

[DDM96] Jeffrey A. Davis, Vivek De, and James Meindl. Optimal low power interconnect networks. In *Symposium on VLSI Technology: Digest of Technical Papers*, pages 1002–1008, June 1996.

[ECC03] European Commission. *Second ECCP Progress Report - Can we meet our Kyoto targets?*, April 2003.

[Eck01] Bruce Eckel. *Thinking in C++*, volume 1. Pearson Education, 2nd edition, 2001.

[Ede83a] H. Edelsbrunner. A new approach to rectangle intersections, Part I. *International Journal of Computer Mathematics*, 13:209–219, 1983.

[Ede83b] H. Edelsbrunner. A new approach to rectangle intersections, Part II. *International Journal of Computer Mathematics*, 13:221–229, 1983.

[EJ98] Hans Eisenmann and Frank M. Johannes. Generic global placement and floorplanning. In *Proceedings of the 35th annual conference on Design automation (DAC)*, pages 269–274. ACM Press, 1998.

[Emb04] Wolfgang Embacher. *Analysis of Automated Power Saving Techniques using Power Compiler (TM).* LIS Diploma Thesis, TU München, Germany, May 2004.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability. A guide to the theory of NP-completeness.* W. H. Freeman and Co., San Francisco, Calif., 1979. A Series of Books in the Mathematical Sciences.

[GRSZ07] Peter Gritzmann, Michael Ritter, Walter Stechele, and Paul Zuber. Optimal wire ordering and spacing in low power semiconductor design. In *(in preparation for) SIAM Journal on Applied Mathematics (or) Journal on Optimization*, 2007.

[Har07] Robert Hartl. *TopCoolViewer – OpenGL-Viewer für Layout-Daten.* LIS Bachelor Thesis, TU München, Germany, February 2007.

[HMH01] Ron Ho, Kenneth W. Mai, and Mark A. Horowitz. The future of wires. *Proceedings Of The IEEE*, 89(4):490–504, April 2001.

[HSWZ05] Andreas Herkersdorf, Walter Stechele, Thomas Wild, and Paul Zuber. *Lecture notes integrated systems technology and solutions in networking / communication (ISNC).* Lehrstuhl für Integrierte Systeme, Technische Universität München, May 2005.

[HYH99] Mark C. Hansen, Hakan Yalcin, and John P. Hayes. Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering. *IEEE Design and Test of Computers*, 16(3):72–80, 1999.

[Iln06] Thomas Ilnseher. *Implementing a Power Saving Post Routing Engine for Evaluation Purposes.* LIS Diploma Thesis, TU München, Germany, August 2006.

[ITR07] *International Technology Roadmap for Semiconductors.* Internet: http://public.itrs.net, last update March 2007, 2007.

[Kah03] Andrew B. Kahng. Research directions for coevolution of rules and routers. In *Proceedings of the 2003 international symposium on Physical design (ISPD)*, pages 122–125. ACM Press, 2003.

[KNM04] Claudia Kretzschmar, André K. Nieuwland, and Dietmar Müller. Why transition coding for power minimization of on-chip buses does not work. In *Proceedings of the conference on Design, automation and test in Europe (DATE)*, pages 512–517. IEEE Computer Society, 2004.

[KSW95] Ralf Kories and Heinz Schmidt-Walter. *Taschenbuch der Elektrotechnik*. Verlag Harri Deutsch, 2nd edition, 1995.

[LB97] Marc Laurent and Michel Briet. Low power design flow and libraries. In Wolfgang Nebel and Jean Mermet, editors, *Low power design in deep submicron electronics*, pages 45–77. Kluwer Academic Publishers, 1997.

[LC04] Lavi Lev and Ping Chao. Down to the wire - routing requirements for the nanometer era. *Cadence Design Systems White Paper*, 2004.

[Len90] Thomas Lengauer. *Combinatorial algorithms for integrated circuit layout*. John Wiley & Sons, Inc., New York, NY, USA, 1990.

[Leu03] Hardy Kwok-Shing Leung. Advanced routing in changing technology landscape. In *Proceedings of the 2003 international symposium on Physical design (ISPD)*, pages 118–121. ACM Press, 2003.

[LN03] Hardy Kwok-Shing Leung and Raymond X. Nijssen. Subgrid detailed routing, January 2003. United States Patent US 6,507,941.

[MKWS04] Nir Magen, Avinoam Kolodny, Uri Weiser, and Nachum Shamir. Interconnect-power dissipation in a microprocessor. In *SLIP '04: Proceedings of the 2004 international workshop on System level interconnect prediction*, pages 7–13, 2004.

[MMP94] Radu Marculescu, Diana Marculescu, and Massoud Pedram. Switching activity analysis considering spatiotemporal correlations. In *Proceedings of the 1994 IEEE/ACM international conference on Computer-aided design (ICCAD)*, pages 294–299. IEEE Computer Society Press, 1994.

[MMP01] Luca Macchiarulo, Enrico Macii, and Massimo Poncino. Low-energy for deep-submicron address buses. In *Proceedings of the 2001 international symposium on Low power electronics and design (ISLPED)*, pages 176–181. ACM Press, 2001.

[MMP02] Luca Macchiarulo, Enrico Macii, and Massimo Poncino. Wire placement for crosstalk energy minimization in address buses. In *Proceedings of the conference on Design, automation and test in Europe (DATE)*, page 158. IEEE Computer Society, 2002.

[MPS03] Enrico Macii, Massimo Poncino, and Sabino Salerno. Combining wire swapping and spacing for low-power deep-submicron buses. In *Proceedings of the 13th ACM Great Lakes symposium on VLSI (GLSVLSI)*, pages 198–202. ACM Press, 2003.

[NO03] Sampo Niskanen and Patric R. J. Östergård. *Cliquer User's Guide, Version 1.0*. Tech. Rep. T48, Communications Laboratory, Helsinki University of Technology, Espoo, Finland, 2003.

[NW92] Keith Nabors and J. White. Multipole-accelerated 3-d capacitance extraction algorithms for structures with conformal dielectrics. In *Proceedings of the 29th ACM/IEEE conference on Design automation (DAC)*, pages 710–715. IEEE Computer Society Press, 1992.

[OJ04] Bernd Obermeier and Frank M. Johannes. Temperature-aware global placement. In *Proceedings of the 2004 conference on Asia South Pacific design automation (ASP-DAC)*, pages 143–148. IEEE Press, 2004.

[ORJ05] Bernd Obermeier, Hans Ranke, and Frank M. Johannes. Kraftwerk: a versatile placement approach. In *Proceedings of the 2005 international symposium on Physical design (ISPD)*, pages 242–244. ACM Press, 2005.

[PCA07] Carlo A. Pignedoli, Alessandro Curioni, and Wanda Andreoni. Anomalous behavior of the dielectric constant of hafnium silicates: A first principles study. *Physical Review Letters*, 98(3):037602, 2007.

[Ped96] Massoud Pedram. Power minimization in ic design: principles and applications. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 1(1):3–56, 1996.

[Per06] Tekla S. Perry. Wizard of Watts. *Spectrum, IEEE*, 43 INT(6):26 – 31, June 2006.

[PVV05] Vassilis Paliouras, Johan Vounckx, and Diederik Verkest, editors. *Integrated Circuit and System Design, Power and Timing Modeling, Optimization and Simulation, 15th International Workshop, (PATMOS)*, volume 3728 of *Lecture Notes in Computer Science*. Springer, 2005.

[PY91] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[Rab03] Jan M. Rabaey. *Digital integrated circuits: a design perspective*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.

[Rab05] Jan M. Rabaey. Traveling the wild frontier of ultra low-power design. In Paliouras et al. [PVV05], page 747.

[RNS05] Shanq-Jang Ruan, Edwin Naroska, and Uwe Schwiegelshohn. Simultaneous wire permutation, inversion, and spacing with genetic algorithm for energy-efficient bus design. In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, pages 4–8, Apr 2005.

[Roc72] Ralph Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1972.

[RSG03] Vijay Raghunathan, Mani B. Srivastava, and Rajesh K. Gupta. A survey of techniques for energy efficient on-chip communication. In *Proceedings of the 40th conference on Design automation (DAC)*, pages 900–905. ACM Press, 2003.

[Sau99] Stephan Sauter. *Entwurfskriterien für Taktnetz-Architekturen zur Berücksichtigung der Parametervariationen auf Chip- und Waferebene.* PhD thesis, Technische Universität München, June 1999.

[SD97] Jeffrey Z. Su and Wayne W. Dai. Post-route optimization for improved yield using a rubber-band wiring model. In *Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design (ICCAD)*, pages 700–706. IEEE Computer Society, 1997.

[SDK96] Alfred Schmitt, Oliver Deussen, and Marion Kreeb. *Einführung in graphisch-geometrische Algorithmen.* Teubner, Stuttgart, 1996.

[She93] Naveed Sherwani. *Algorithms for VLSI Physical Design Automation.* Kluwer Academic Publishers, 1993.

[SK96] P. Schneider and S. Krishnamoorthy. Effects of correlations on accuracy of power analysis - an experimental study. In *Proceedings of the 1996 international symposium on Low power electronics and design (ISLPED)*, pages 113–116. IEEE Press, 1996.

[SLKY98] Weiping Shi, Jianguo Liu, Naveen Kakani, and Tiejun Yu. A fast hierarchical algorithm for 3-d capacitance extraction. In *Proceedings of the 35th annual conference on Design automation (DAC)*, pages 212–217. ACM Press, 1998.

[SLKY02] Weiping Shi, Jianguo Liu, Naveen Kakani, and Tiejun Yu. A fast hierarchical algorithm for three-dimensional capacitance extraction. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, volume 21, pages 330–336, March 2002.

[SM06] Krishnan Sundaresan and Nihar R. Mahapatra. Value-based bit ordering for energy optimization of on-chip global signal buses. In *Proceedings of the conference on Design, automation and test in Europe (DATE)*, pages 624–625. European Design and Automation Association, 2006.

[SOC06] T. Serdar, O. Omedes, and B. Carpentier. Timing preservation in wire spreading utilized for yield improvement. In *IEEE International Conference on Integrated Circuit Design and Technology (ICICDT)*, pages 24–26, May 2006.

[SS01] Youngsoo Shin and Takayasu Sakurai. Coupling-driven bus design for low-power application-specific systems. In *Proceedings of the 38th conference on Design automation (DAC)*, pages 750–753. ACM Press, 2001.

[Sta83] Charles H. Stapper. Modeling of integrated circuit defect sensitivities. *IBM Journal of Research and Development*, 27(6):549–557, 1983.

[Str00] Chris W. H. Strolenberg. Stay away from minimum design-rule values. In *Proceedings of the conference on Design, automation and test in Europe (DATE)*, pages 71–73. ACM Press, 2000.

[SYMY03a] Atsushi Sakai, Takashi Yamada, Yoshifumi Matsushita, and Hiroto Yasuura. Reduction of coupling effects by optimizing the 3-d configuration of the routing grid. In *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, volume 11, pages 951–954, October 2003.

[SYMY03b] Atsushi Sakai, Takashi Yamada, Yoshifumi Matsushita, and Hiroto Yasuura. Reduction of crosstalk noise by optimizing 3-d configuration of the routing grid. In *Proceedings of the 2003 conference on Asia South Pacific design automation (ASP-DAC)*, pages 49–52. ACM Press, January 2003.

[SYMY03c] Atsushi Sakai, Takashi Yamada, Yoshifumi Matsushita, and Hiroto Yasuura. Routing methodology for minimizing interconnect energy dissipation. In *Proceedings of the 13th ACM Great Lakes symposium on VLSI (GLSVLSI)*, pages 120–123. ACM Press, April 2003.

[Syn06] Synopsys. Liberty reference manual (version 2006.06). 2006.

[WE93] Neil H. E. Weste and Kamran Eshraghian. *Principle of CMOS VLSI Design*. Addison-Wesley, 2nd edition, 1993.

[Win04] Armin Windschiegl. *Prognose von Leitungskapazitäten zur Verlustleistungsanalyse auf Logikebene*. PhD thesis, Technische Universität München, November 2004.

[WZS02] Armin Windschiegl, Paul Zuber, and Walter Stechele. Exploiting metal layer characteristics for low-power routing. In *Proceedings of the 12th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pages 55–64. Springer-Verlag, 2002.

[Yea98] Gary K. Yeap. *Practical low power digital VLSI design*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[YK99] Joon-Seo Yim and Chong-Min Kyung. Reducing cross-coupling among interconnect wires in deep-submicron datapath design. In *DAC '99: Proceedings of the 36th ACM/IEEE conference on Design automation*, pages 485–490, New York, NY, USA, 1999. ACM Press.

[ZBI+07] Paul Zuber, Othman Bahlous, Thomas Ilnseher, Florian Helmut Müller, and Walter Stechele. Crosscoupling power optimal wire spacing in quasilinear runtime. *SPIE International Symposium on Microtechnologies for the New Millennium*, May 2007.

[ZBIS07] Paul Zuber, Othman Bahlous, Thomas Ilnseher, and Walter Stechele. Wire topology optimization for low power cmos. In *(submitted to) IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2007.

[ZGRS05] Paul Zuber, Peter Gritzmann, Michael Ritter, and Walter Stechele. The optimal wire order for low power CMOS. In Paliouras et al. [PVV05], pages 674–683.

[ZMS05] Paul Zuber, Florian Helmut Müller, and Walter Stechele. Optimization potential of CMOS power by wire spacing. volume 67 of *Lecture Notes in Informatics*, pages 344–348. Springer, 2005.

[ZWdO+05] Paul Zuber, Armin Windschieg, Raúl Medina Beltrán de Otálora, Walter Stechele, and Andreas Herkersdorf. Reduction of cmos power consumption and signal integrity issues by routing optimization. In *Proceedings of the conference on Design, Automation and Test in Europe (DATE)*, pages 986–987. IEEE Computer Society, 2005.

Colophon

This dissertation was written on a Lenovo IBM Thinkpad running under Mandriva Linux, version 2007 free. It uses 11 pt Libertine Libertine (Philipp H. Poll) and Helvetica fonts and the report style of the KOMA-Script project (Frank Neukam, Markus Kohm and Axel Kielhorn) and pdf-LaTeX. Arabic and Farsi names in the Preface were set with ArabTeX (Klaus Lagally). Devanagari glyphs (JanaSanskrit) were provided by *Centre for Development of Advanced Computing, Nodia.* The name of the font for Hangul and Chinese is "DejaVu Sans". The cover photo "TopCool" shows the Zugspitze in Bayern.

Below: routing layers of a circuit as seen by *TopCool.* Different colours in the "rat's nest" correspond to different switching activities. Traditional physical design tools did not include the activities. This graphic was generated by *TopCoolViewer* [Har07].