

Thermal aware floorplanning incorporating temperature dependent wire delay estimation



Andreas Thor Winther^a, Wei Liu^{b,*}, Alberto Nannarelli^c, Sarma Vrudhula^d

^a Knowles Electronics, Roskilde, Denmark

^b Oticon A/S, Smørum, Denmark

^c DTU Compute, Technical University of Denmark, Kongens Lyngby, Denmark

^d Computer Systems Engineering, Arizona State University, Tempe, USA

ARTICLE INFO

Keywords:

Temperature
Wire delay modeling
Floorplanning
Congestion
Reliability
Thermal analysis

ABSTRACT

Temperature has a negative impact on metal resistance and thus wire delay. In state-of-the-art VLSI circuits, large thermal gradients usually exist due to the uneven distribution of heat sources. The difference in wire temperature can lead to performance mismatch because wires of the same length can have different delay.

Traditional floorplanning algorithms use wirelength to estimate wire performance. In this work, we show that this does not always produce a design with the shortest delay and we propose a floorplanning algorithm taking into account temperature dependent wire delay as one metric in the evaluation of a floorplan. In addition, we consider other temperature dependent factors such as congestion and interconnect reliability.

The experiment results show that a shorter delay can be achieved using the proposed method.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

With technology scaling, the feature sizes of both CMOS devices and wires shrink and designers are able to integrate more and more functionalities into a single chip. The delay in CMOS transistors decreases as the channel length is reduced in each new process. The delay in metal wires, on the other hand, shows different behaviors. For local wires, the delay decreases as the distance between the end points becomes smaller with scaling. For global wires, which have to span across the chip, the delay increases due to the fact that the die size does not shrink but slightly increases in each new process [1]. In fact, the delay in global wires has increased steadily with technology scaling over the years and already dominates path delays [2].

In addition to technology scaling, the modeling of global wires is further complicated by thermal effects. Due to the high degree of integration (processing elements and memory blocks on the same die) and various aggressive power management techniques (such as clock gating, power gating, voltage islands, etc.), the power consumption in different regions of the chip (i.e. the power density) can vary significantly [3]. The spatially non uniform power consumption within the chip exhibits as thermal gradients, which are temperature differences between different regions.

The heat generated inside the chip has to be transferred to the ambient environment mainly through the heat sink attached to the silicon substrate. However, a secondary heat conduction path also exists from the substrate towards the packaging through the metal layers [4]. In nanometer technologies, in spite of an increase in the number of available metal layers, the top metal layers may still get closer to the substrate which results in a stronger thermal coupling between the substrate and the wires [5].

The high temperature and large thermal gradient in the metal layers can affect many aspects of interconnect design, including signal delay, routing congestion and reliability. The propagation delay in metal wires is severely degraded by high temperature as the electrical resistivity in metal increases linearly with temperature. The large within-die thermal gradients result in performance mismatch between wires of the same length but subject to different temperature. Traditional physical design algorithms, such as floorplanning and routing, assume resistivity in interconnects is uniform and constant. Consequently, wirelength is used as a metric to estimate signal delay and congestion of interconnects [6–8]. However, in designs where the substrate has a nonuniform thermal profile, the traditional way of estimating wire delay can lead to large errors. This is because wire performance decreases with an increase in temperature and the delay of a hot wire and a cool wire are no longer equal even though their lengths are the same.

Furthermore, the thermal effect is more significant in global wires than in local wires because global wires are routed in layers that are

* Corresponding author. Tel.: +45 60753957.

E-mail address: wli@oticon.dk, liuweizju@hotmail.com (W. Liu).

far away from the heat sink, and global wires span long distance thus possibly developing a larger thermal gradient.

Clock networks, which contain many global wires, are very sensitive to thermal variations. In recent years, temperature variation induced clock skew in clock distribution networks has received a lot of attention. In [9,10], the authors described design time clock tree synthesis algorithms to modify merging locations against nonuniform substrate thermal profile. In [11], optimal insertion of tunable delay buffers into clock trees is discussed to adjust at run time the delay of clock distribution paths that are more susceptible to temperature variations. Thermal aware global routing algorithms for improving reliability are also discussed in [12,13].

As for global signal wires, although extensive work has been done on thermal aware floorplanning, all of these works assume electrical resistivity in wires is constant and thermal gradients in the substrate have no impact on wire delay. These assumptions are in general invalid and increasingly inaccurate in nanometer high performance designs where large temperature gradients already exist in the substrate.

In this paper, we study the problem of estimating the temperature dependent wire delay during the floorplanning stage. We first illustrate the impact of nonuniform thermal profile on the delay in wires. Then we propose a new way to estimate the wire delay in thermal aware floorplanning algorithms. The proposed algorithm takes delay, instead of wirelength, as one of the optimization goals, in this way, mitigating the excessive delay overhead caused by high temperature. In addition, we also consider the impact of routing congestion and the reliability of wires, which are important metrics in evaluating floorplans in a realistic scenario.

2. Thermal aware floorplanning

Floorplanning is the initial stage of physical implementation of VLSI circuits, that determines, to a large extent, the quality of the final design. Floorplanning transforms the functional description of a circuit, in the form of a netlist of gates and macros, into a physical description, in the form of dimensions and location coordinates.

During the floorplanning stage, the main design tasks include macro block placement, global wire planning and Power/Ground network design. Traditional floorplanning algorithms only optimize the total area and wirelength. A smaller area reduces the cost since more circuits can be produced on the same wafer. Shorter wirelength makes routing easier and usually results in better performance as well.

In recent years, as thermal issues become prominent, the maximum temperature is also added to the cost functions in so called thermal aware floorplanning algorithms [8,14,15]. Placing a hot block in the middle of cool blocks can effectively bring down the peak temperature due to better heat spreading. Since the thermal coupling between high power consumption blocks can significantly affect the temperature distribution in the whole chip, thermal aware floorplanning algorithms consider peak temperature in addition to area and wirelength in the evaluation of a floorplan. The estimation of peak temperature usually requires the use of compact thermal models that can compute the temperature profile in a very efficient way [16].

The floorplanning tool proposed in [8], HotFloorplan, is an architectural level thermal aware floorplanner. HotFloorplan represents the topology of a floorplan in *Normalized Polish Expression* [6] and the optimization algorithm is implemented as a simulated annealing process. The algorithm chooses a random generated floorplan during each iteration of the annealing and uses the maximum temperature, total area and total wirelength as evaluation metrics in the cost function.

The pseudocode for the annealing is given in Algorithm 1. In brief, the optimization process goes through a series of steps and with

Algorithm 1 Pseudocode for the HotFloorplan algorithm.

```

Set initial annealing temperature;
step = 0;
try = 0;
while steps != max steps AND probability > minimum do
  create initial floorplan;
  while try < max tries do
    try++;
    randomly perturb current floorplan;
    evaluate new floorplan;
    if new floorplan better than best floorplan then
      best floorplan = new floorplan;
    end if
    if accept(new floorplan) == true then
      current floorplan = new floorplan;
    end if
  end while
  step++;
  change simulated annealing temperature;
  calculate probability as a function of temperature;
end while
OUTPUT: Best floorplan

```

every step a synthetic *temperature*¹ is changed according to an annealing schedule. The temperature is used to define how wide the search for a better floorplan is (within the solution space) – a low temperature means a narrow search.

Within each step, HotFloorplan starts out with an initial floorplan that is simply any possible legal floorplan within the solution space. HotFloorplan then tries to optimize this floorplan by moving the blocks around. For every candidate floorplan, the algorithm invokes routines in HotSpot [17] to compute the worst case thermal profile using power dissipation values of each functional block. If the candidate has a lower cost, it is always accepted, otherwise the candidate is accepted conditionally based on a probability factor. At the end of the inner loop, we get the best known floorplan available at that temperature. The next step repeats this action only this time with another annealing temperature according to the annealing schedule. In the end, we get the best floorplan created during all steps. The algorithm either ends when the maximum number of steps has been reached, or if the probability (which is a function of the temperature) of the next step is under a given value (i.e., threshold). How HotFloorplan decides which floorplan is *the best* is vital, as it is this evaluation process that has been modified to include thermal awareness.

The connectivity information between blocks is stored in a two-dimensional connectivity matrix, and the wirelength between the two endpoints of a wire is estimated by measuring the “Manhattan distance”.

3. Delay modeling of metal wires

In this section, we describe the models used to estimate the temperature in wires and the process to calculate temperature dependent wire delay.

3.1. Temperature estimation

The temperature rise in metal wires is caused by both self-heating and heat diffusion from the substrate. During a signal transition inside a metal wire, the accelerated charge carriers (i.e., electrons) collide with other carriers and atoms in the electric field. The collisions

¹ Here, the term temperature is used as an attribute of the annealing and does not refer to the actual substrate (chip) temperature.

cause vibration in these particles and consequently result in a rise in temperature, which is often referred to as self-heating. In addition, heat generated in the active devices on the substrate (i.e., the transistors) can also diffuse towards the metal layers and cause a temperature rise in metal wires.

According to [5], the temperature within the interconnect subject to a given substrate temperature can be expressed as:

$$T(x) = T_{sub} + \frac{\theta}{\lambda^2} \left(1 - \frac{\sinh \lambda x + \sinh \lambda(L-x)}{\sinh \lambda L} \right) \quad (1)$$

$$\lambda^2 = \frac{1}{k_m} \left(\frac{k_{ins}^*}{t_m t_{ins}} - \frac{I_{rms}^2 \rho_i \beta}{w^2 t_m^2} \right) \quad (2)$$

$$\theta = \frac{I_{rms}^2 \rho_i}{w^2 t_m^2 k_m} \quad (3)$$

where T_{sub} is the substrate temperature, θ and λ are constants for a chosen metal layer in a specific technology node, k_m and k_{ins}^* are the thermal conductivity of the metal and the insulator, t_m and t_{ins} are the thickness of metal and the insulator, w is the width of the interconnect, I_{rms} is the average current density in the interconnect and ρ_i is the electrical resistivity of metal at nominal temperature. The peak temperature rise is equal to θ/λ^2 for interconnects whose lengths (L) are larger than the heat diffusion length.

Using Eq. (1), we plot the thermal profiles for a local, a semi-global and a global copper (Cu) interconnect 1000 μ m long in a 50 nm technology in Fig. 1 with parameters provided in ITRS [18]. Wire sizes in upper metal layers are larger than in lower layers. In addition, the three metal layers differ in t_{ins} which is the distance from interconnect to the substrate. The temperature in the substrate is assumed to be uniform at 100 °C. The current density is 3.0×10^6 A/cm² for all three layers.

As it can be seen in Fig. 1, the global interconnect has the highest peak temperature which is in the top most metal layer and thus the farthest away from the heat sink.

3.2. Delay calculation

The temperature of a metal wire has a negative impact on metal resistance and thus wire delay. The electrical resistance of metal has

a linear relationship with its temperature and can be expressed as:

$$R(x) = R_0(1 + \beta \cdot T(x)) \quad (4)$$

where R_0 is the resistance at the reference temperature, β is the temperature coefficient (1/°C) and $T(x)$ is the temperature profile along the length of the wire. For example, since the value of β for copper at room temperature is $3.9E-3$, a temperature increase by 10 °C will result in a resistance increase by 3.9%.

According to the distributed RC Elmore delay model [19], signal propagation delay through the interconnect of length L can be written as:

$$D = R_d \left(C_L + \int_0^L c_0(x) dx \right) + \int_0^L r_0(x) \cdot \left(\int_x^L c_0(\tau) d\tau + C_L \right) dx \quad (5)$$

where R_d is the driver cell's ON resistance, $c_0(x)$ and $r_0(x)$ are the capacitance and resistance per unit length at location x and C_L is the load capacitance.

By combining Eqs. (4) and (5), we can obtain a temperature dependent interconnect delay model:

$$D = D_0 + (c_0 L + C_L) r_0 \beta \int_0^L T(x) dx - c_0 r_0 \beta \int_0^L x \cdot T(x) dx \quad (6)$$

where

$$D_0 = R_d (c_0 L + C_L) + \left(c_0 r_0 \frac{L^2}{2} + r_0 L C_L \right) \quad (7)$$

is the Elmore delay of the interconnect corresponding to the unit length resistance at reference temperature.

Given a temperature profile and the dimensions of an interconnect, we can calculate its delay from Eq. (6). In Fig. 2, we plot the percentage of delay increase as temperature increases for wires of different lengths. It can be seen from Fig. 2 that for 100 °C rise in temperature, the delay can increase by more than 30% for all wires. In addition, the delay increase is higher for longer wires, which means the performance of global wires suffer more from high temperature.

The heat distribution within a circuit is usually non-uniform due to un-even activities and large thermal gradients typically exist on the substrate. As a specific example, we extract the substrate thermal profile from a test circuit for one segment of a wire as illustrated in Fig. 3. Based on Eq. (1), the temperature profile of this global interconnect subject to substrate thermal profile in Fig. 3 is computed

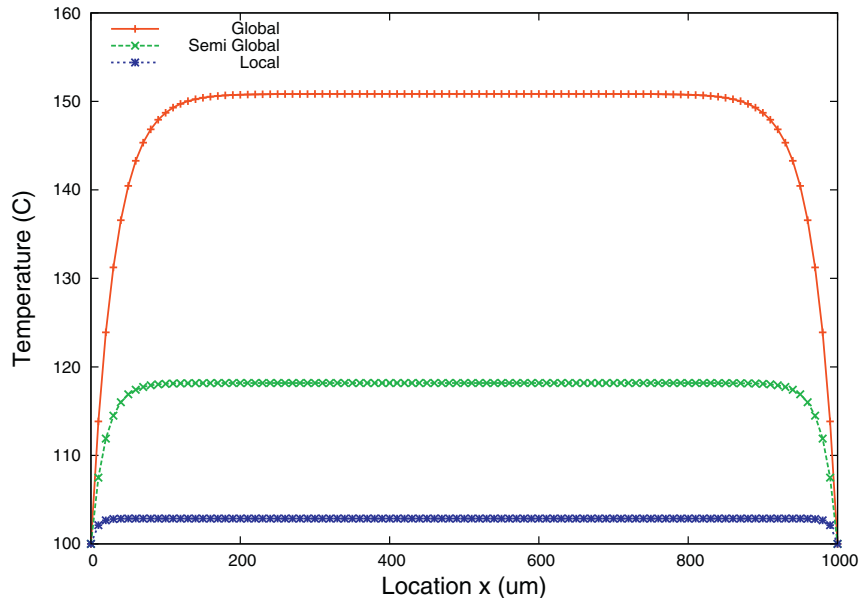


Fig. 1. Temperature profiles along the length of interconnects on different metal layers.

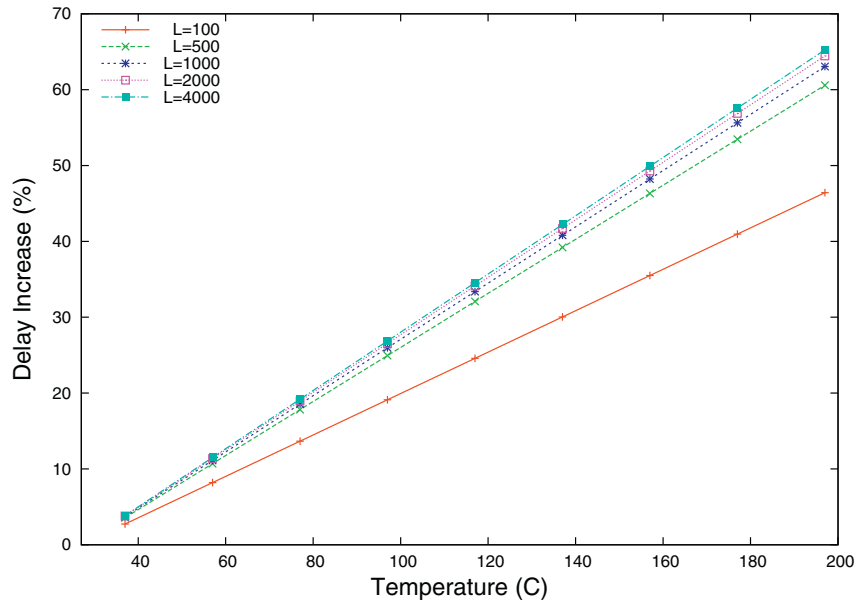


Fig. 2. Percentage increase of signal delay with respect to nominal delay at room temperature (27 °C).

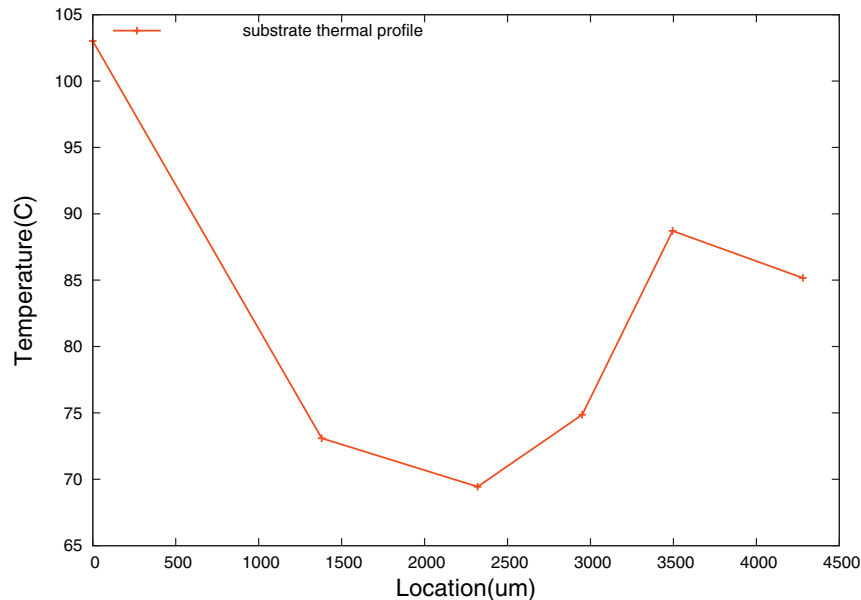


Fig. 3. Substrate temperature profile along the length of an interconnect.

and shown in Fig. 4. A thermal gradient larger than 25 °C is developed along the length of the wire. The maximum and average temperatures of the interconnect are also included for the purpose of comparison.

In Fig. 5, we plot the delay increase in this global wire. The delay at the end of the wire using the extracted thermal profile is 20 ns, while by using the maximum and average temperatures the errors in the estimation are both larger than 25%. This means that, when estimating global wire delay, ignoring the thermal gradient and using only the maximum or average temperature may result in a significant error.

In Fig. 6, we show the statistics on the average temperature of all global interconnects in the test circuit. The height of a bar represents the number of interconnects having the indicated temperature. The substrate has a peak temperature of 104 °C and a temperature gradient of around 30 °C. Obviously, the interconnects have a significant higher average temperature than the substrate and a few of them even reached above 145 °C due to self-heating.

For long wires, it is especially desirable to avoid routing above substrate regions at high temperature to reduce the performance degradation due to the temperature. However, detouring around hotspot regions may increase wirelength and cause congestion. Therefore, an accurate overall analysis is necessary to assess different routing choices.

In Fig. 7, we show the statistics on temperature gradient within the global interconnects in this test circuit. Although the temperature gradient in about half of the interconnects is less than 10 °C, more than 40% of interconnects do have a gradient larger than 20 °C. For these wires, using average or maximum temperature can introduce a large error in delay estimation and therefore an accurate analysis taking into account the thermal profile of each wire is needed.

4. Thermal aware wire planning in floorplanning

As shown in Section 3.1, the delay of global wires subject to large temperature variations is no longer linearly proportional to

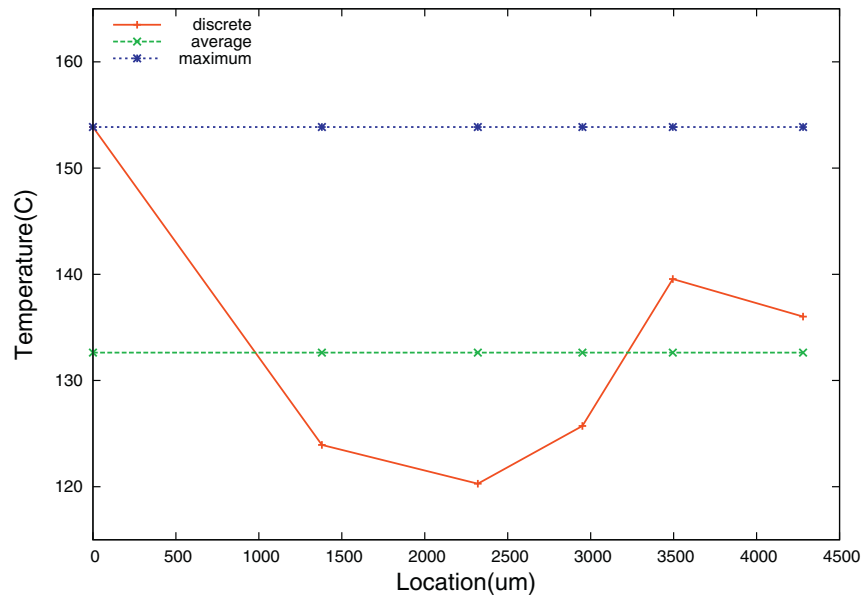


Fig. 4. Interconnect temperature profile along the length.

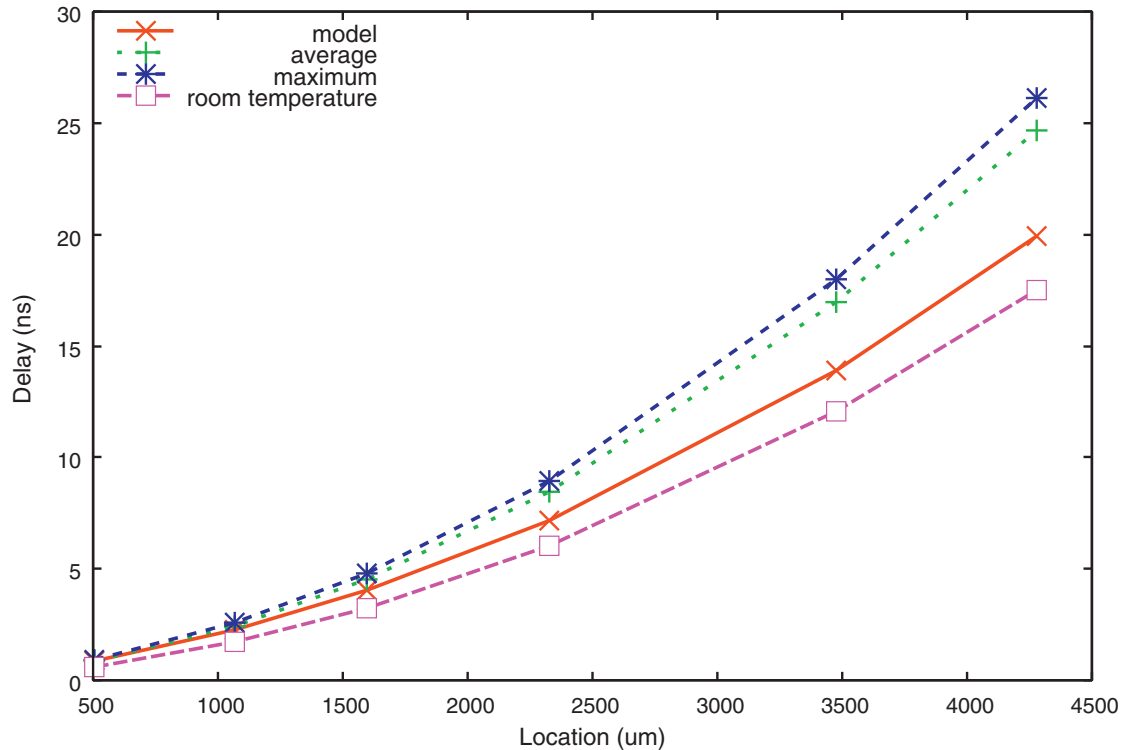


Fig. 5. Delay increase in a global wire using different thermal profiles.

wirelength. To have an accurate estimation of wire delay, the temperature effect including thermal gradients has to be considered. Since the thermal profile on the die is mainly determined by the locations of macroblocks, it is, therefore, possible to perform temperature dependent delay estimation at the floorplanning stage.

4.1. Wire planning and delay calculation

The wire planning method is described in Algorithm 2. The location of each wire is determined by performing L-shape routing between the center of the connecting blocks. Once the physical layout of the wire is known, we record the blocks over which the wire is

routed. The temperature profile along the wire and the wire delay are then calculated using Eqs. (1) and (6). With thermal effects taken in account, the two paths in the bounding box of two end points of a wire can have different delay although their lengths are the same. In our algorithm, we choose the path with a shorter delay, which is different from HotFloorplan where the two paths are considered as identical.

The temperature profile is also used to evaluate the wire reliability in terms of Mean Time To Failure (MTTF). In addition, a congestion map made up of a two dimensional matrix is updated with the route of the wire to evaluate the routability of the floorplan. The congestion map is useful because in our algorithm more wires are likely to be

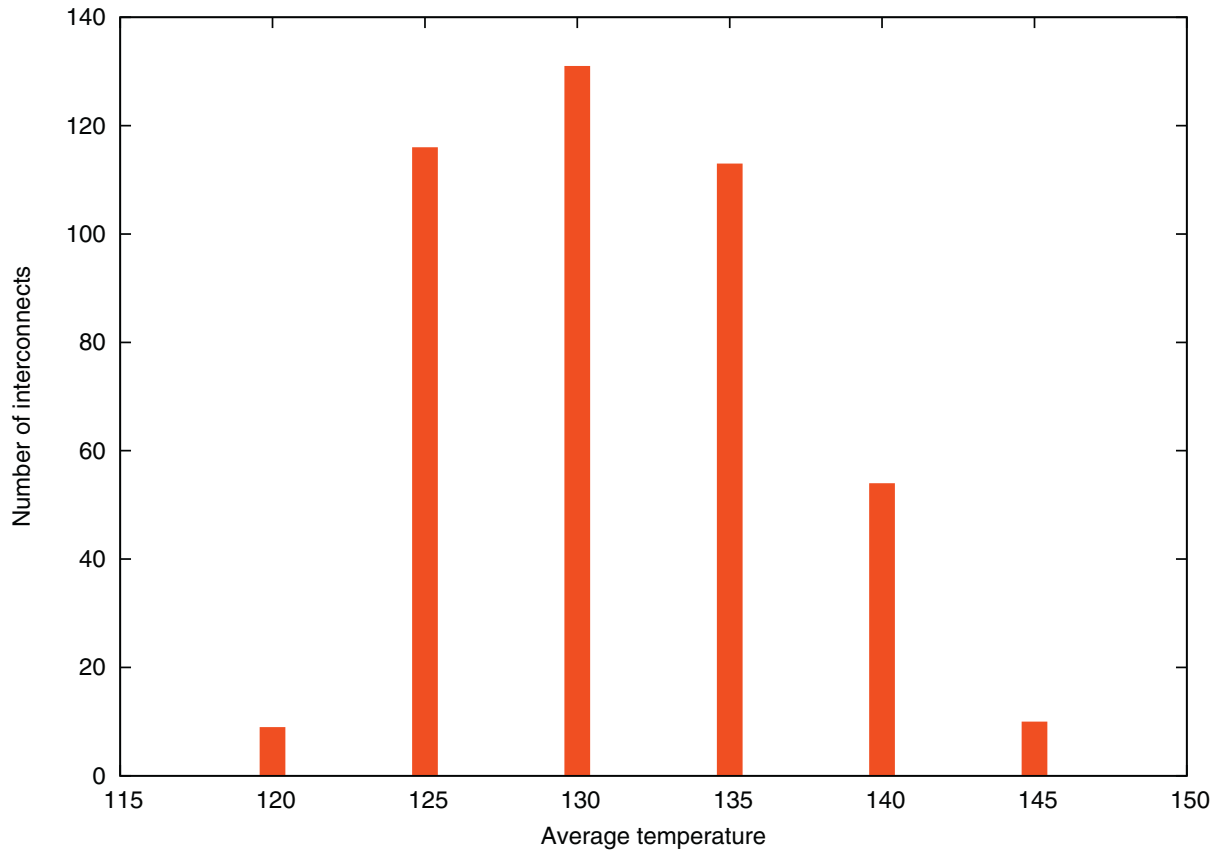


Fig. 6. Statistics on average temperature in interconnects.

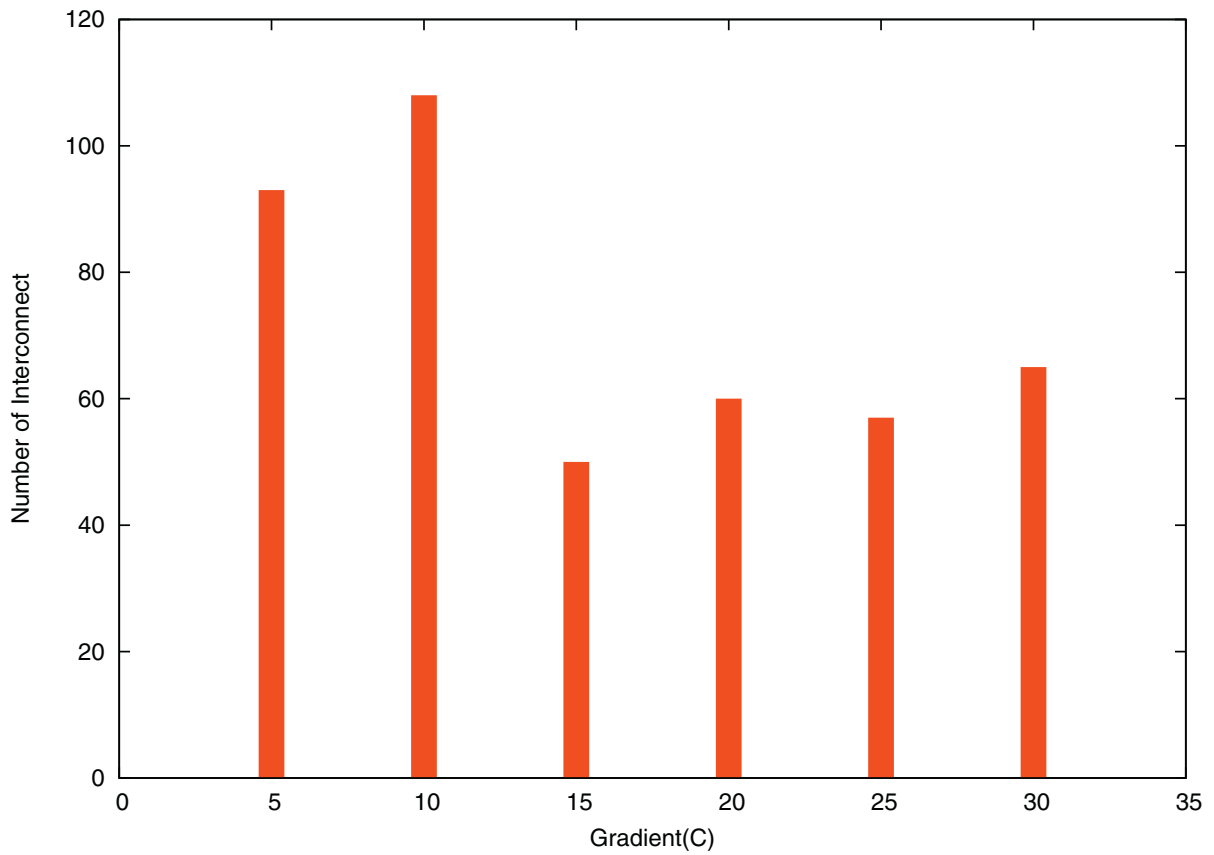


Fig. 7. Statistics on temperature gradient in interconnects.

Algorithm 2 Wire planning in floorplanning.

INPUT: Floorplan description
INPUT: Connectivity matrix
INPUT: Thermal map
for all wires in the connectivity matrix **do**
 Perform L-shape routing
 Extract the thermal profile along the wire
 Calculate wire delay using Eq. 6
 Calculate wire reliability using Eq. 11
 Update congestion map
end for
OUTPUT: Delay, Congestion and Reliability of Wires

routed in regions with a low temperature, potentially causing routing congestion. In the next two subsections we describe the congestion and reliability models used in the algorithm.

4.2. Congestion model

The floorplan is divided into a 2-dimensional grid with every cell in the grid having an initial congestion value of 0. Whenever a wire is created, we find the cells crossed by this wire and we increase by 1 the congestion value of these cells. When all the wires are routed, we have a congestion map with each cell having a value equal to the number of wires passing through it. The pseudocode for the implementation is shown in Algorithm 3.

Algorithm 3 Pseudocode for the implementation of the congestion model.

INPUT: Connectivity matrix
for all blocks, i , in the grid **do**
 congestion[i]=0
end for
for all blocks, i , in the grid **do**
 if wire runs through block i **then**
 congestion[i]=congestion[i]+1
 end if
end for
OUTPUT: Max congestion

Two examples of the same congestion map are shown in Fig. 8: in (a) the functional blocks of the benchmark circuit are outlined, and in (b) the grid map is outlined. On the map, locations with little congestion are colored in blue while congestion spots are colored in red.

4.3. Reliability model

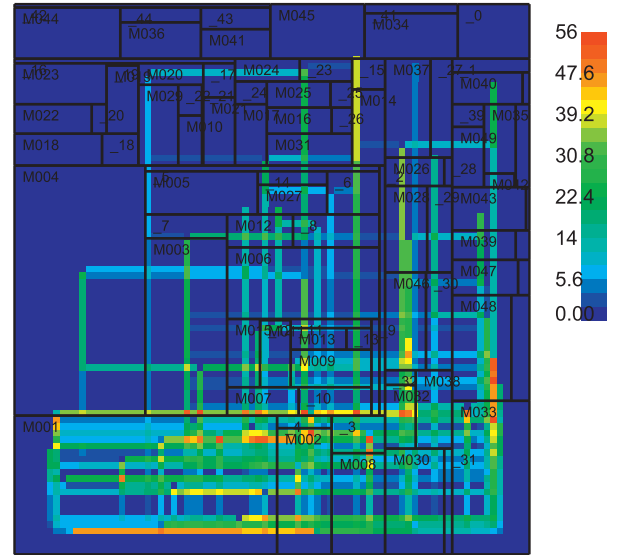
The reliability model is based on Black's equation [20]:

$$MTTF = \frac{A}{J^2} e^{\frac{E_a}{kT}} \quad (8)$$

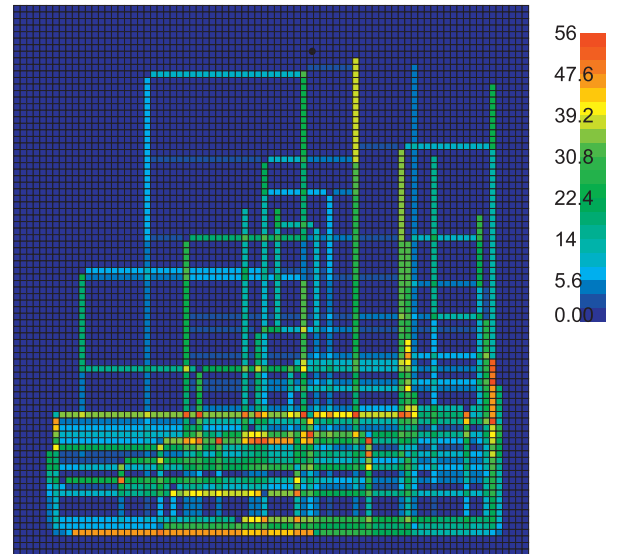
where T is the metal temperature in Kelvin, J is the current density, and A , E_a and K are constants: A is the cross-sectional area of the interconnect (technology dependent), $E_a = 0.5$ eV is the activation energy, and $K = 8.62 \cdot 10^{-5}$ eV/K is the Boltzmann constant.

The Mean Time To Failure (MTTF) is the average time to which a wire fails due to electromigration. Because of A and J , Black's equation is technology dependent.

To capture the exponential relationship between the wire reliability and temperature, we find the relative reliability, $MTTF_r$, of the wire, w , at a reference temperature (room temperature = 300 K) compared to the actual temperature. As shown in [13], using Eq. (8), we get the



(a) Functional blocks outlined



(b) Grid map outlined

Fig. 8. Example congestion maps. The functional blocks have been outlined in (a) and the grid map has been outlined in (b). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

following relation:

$$MTTF_r(w) = \frac{MTTF_{room}}{MTTF_{wire}} = \exp\left[\frac{E_a}{K} \left(\frac{1}{T_{room}} - \frac{1}{T_{wire}}\right)\right] \quad (9)$$

where T_{wire} is the temperature of the wire under investigation. The two parameters A and J cancel out in Eq. (9), leaving only well known constants (E_a , K , and T) in the equation.

To avoid that short wires with low $MTTF_r$ have a large impact on the overall reliability, the $MTTF_r$ of the wire w is multiplied by its length. This gives a cost for a wire, w , in the circuit:

$$MTTF_{cost}(w) = MTTF_r(w) \cdot l_w \quad (10)$$

Finally, the overall reliability cost metric is obtained by the average (weighted by the wirelength l_w), of all reliability costs:

$$Reliability\ cost\ metric = \frac{1}{N} \sum_{w \in C} MTTF_{cost}(w) \quad (11)$$

where N is the total number of wires in the circuit C .

Table 1
Experimental results on MCNC benchmarks using different cost functions and 50 nm process parameters.

Benchmark	No. of blocks	Cost function	T_{max} (K)	Area (mm ²)	Wirelength (m)	Tot. delay (μ s)	C_{max}	R_{avg}
ami49	49	CF1	386.7	39.9	5.518	5.55	1.00	1.00
		CF2	387.4	40.3	5.340	5.05	1.15	0.84
		CF3	389.5	43.0	5.751	5.18	0.71	0.82
ami33	33	CF1	394.0	1.3	8.438	6.02	1.00	1.00
		CF2	396.4	1.3	8.172	5.80	1.01	1.01
		CF3	385.1	1.5	9.096	6.42	0.49	0.67
apte	9	CF1	378.3	48.2	2.834	4.46	1.00	1.00
		CF2	379.4	48.2	2.723	4.20	1.00	0.95
		CF3	378.7	48.4	3.146	5.07	0.72	0.95
hp	11	CF1	360.5	9.5	1.732	2.57	1.00	1.00
		CF2	358.4	10.9	2.026	2.50	1.04	0.85
		CF3	358.0	10.9	1.786	2.01	0.67	0.69
xerox	10	CF1	367.9	20.9	20.315	25.03	1.00	1.00
		CF2	368.7	21.0	20.724	19.98	0.91	0.84
		CF3	369.5	21.3	23.612	24.98	0.82	0.88

5. Evaluation of the proposed algorithm

In this section, we describe the setup for the algorithm evaluation and present the experiment results based on a benchmark circuit set.

5.1. Experiment setup

We implemented the wire planning algorithm described in the previous section based on HotFloorplan. Algorithm 2 described in Section 4 is inserted into the simulated annealing process, which allows HotFloorplan to estimate the temperature dependent wire delay. For every candidate floorplan created during the annealing, our cost function is used to evaluate the quality of the candidate. To speed up the lengthy simulation time of HotFloorplan we adopted the fast simulated annealing (FastSA) approach proposed in [21], which reduces the number of uphill moves at the beginning of the searching and uses more time to find the local optimum.

We used the MCNC [22] macroblock benchmark circuits with parameters in a 50 nm process to evaluate our proposed wire planning methods.

To compare the proposed methods against the original HotFloorplan implementation, we run the floorplanning algorithm on each benchmark circuit using different cost functions. The three cost functions are:

- **CF1** is the original function in HotFloorplan defined as: $\lambda_A \times Area + \lambda_W \times Wirelength + \lambda_T \times T_{max}$
- **CF2** replaces Wirelength with WireDelay in CF1 defined as: $\lambda_A \times Area + \lambda_D \times WireDelay + \lambda_T \times T_{max}$
- **CF3** adds the maximum congestion and the average reliability to CF2 defined as: $\lambda_A \times Area + \lambda_D \times WireDelay + \lambda_T \times T_{max} + \lambda_C \times C_{max} + \lambda_R \times R_{avg}$

where the λ s are the weight of each metric in the cost function.

5.2. Experiment results

The experiment results are summarized in Table 1, where C_{max} and R_{avg} in the last two columns are the normalized maximum congestion value and the normalized average reliability of all wires, respectively. In the experiment results, CF1 and CF2 are of the most interest. The λ s we chose in CF1 and CF2 give order of optimization priorities from high to low as: area, peak temperature and wire performance. The parameters C_{max} and R_{avg} are normalized to CF1, which give a comparison against the results obtained from the original HotFloorplan.

The results in Table 1 show that a significant decrease in wire delay can be achieved while maintaining a minimal area overhead. For *xerox* a 20% wire delay decrease is achieved using CF2 at the cost of a 0.4% increase in area. Furthermore, the congestion, as well as reliability, has improved.

For *hp*, on the other hand, only a 3% decrease is achieved at the cost of added congestion and an area overhead of 15%. However, looking at CF3, we have the same area overhead but a 22% decrease in wire delay as well as a 33% decrease in congestion and a 31% improvement in reliability.

For *ami33* and CF2, we see a modest 4% decrease in wire delay. However, area, congestion, and reliability remain the same. For CF3, both the wire delay and area increase – this is not desirable. However, the congestion decreases substantially by over 50% and the reliability improves by 33%.

While most of the benchmarks show improvements in congestion and reliability with both CF2 (except *ami33* and *hp*) and CF3, the benchmark *ami49* has a noticeable 15% increase in congestion. Although the nature of this benchmark is not very congested with only around 5.3 m wire in a 40 mm² area, this could potentially lead to a problem with routability. Including the congestion in the evaluation (CF3) removes this problem but results in a moderate decrease in wire delay of 7% at the cost of a 8% area overhead.

In summary, explicitly integrating the wire delay in the cost function (CF2) can reduce the average wire delay at expense of larger area and longer total wirelength.

6. Conclusions

In this paper, we first characterized the temperature dependent delay in global wires. The results show that modeling wire delay without considering temperature variations (gradient) in different regions of the die crossed by global wires can lead to too pessimistic, or in general inaccurate, estimates.

Based on this characterization, we then included the temperature dependent wire model in HotFloorplan to investigate the impact of our thermal aware model on floorplanning. In addition, we modified the cost functions of HotFloorplan to estimate routing congestion and wire reliability.

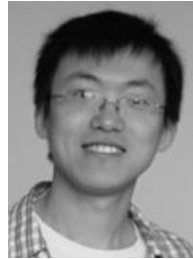
The experiments run on the MCNC benchmarks, show that in the presence of thermal gradients a shorter wirelength does not always produce a shorter delay. However, the proposed method can achieve a better average (total) delay and a better wire reliability at the cost of a moderate increase in area and wirelength.

References

- [1] R. Ho, K. Mai, M. Horowitz, The future of wires, *Proc. IEEE* 89 (4) (2001) 490–504.
- [2] V. Agarwal, S.W. Keckler, D. Burger, The Effect of Technology Scaling on Microarchitectural Structures, University of Texas at Austin, Austin, 2000.
- [3] R. Mahajan, C. pin Chiu, G. Chrysler, Cooling a microprocessor chip, *Proc. IEEE* 94 (8) (2006) 1476–1486.
- [4] Y. Zhan, S.V. Kumar, S.S. Sapatnekar, Thermally aware design, *Found. Trends Electron. Des. Automat.* (3) (2008) 255–370.
- [5] A. Ajami, K. Banerjee, M. Pedram, Modeling and analysis of nonuniform substrate temperature effects on global ULSI interconnects, *IEEE Trans. Computer-Aided Des. Integr. Circuits Syst.* 24 (6) (2005) 849–861.
- [6] D. Wong, C. Liu, A new algorithm for floorplan design, in: *Proceedings of the 23rd Conference on Design Automation*, 1986, pp. 101–107.
- [7] Y.-C. Chang, Y.-W. Chang, G.-M. Wu, S.-W. Wu, B*-trees: a new representation for non-slicing floorplans, in: *Proceedings of the 2000 Design Automation Conference*, 2000, pp. 458–463.
- [8] K. Sankaranarayanan, S. Velusamy, M. Stan, C. L. K. Skadron, A case for thermal-aware floorplanning at the microarchitectural level, *J. Instruction-Level Parallelism* 7 (2005) 1–16.
- [9] C. Liu, J. Su, Y. Shi, Temperature-aware clock tree synthesis considering spatiotemporal hot spot correlations, in: *Proceedings of the 26th IEEE International Conference on Computer Design*, 2008, pp. 107–113.
- [10] M. Cho, S. Ahmedt, D.Z. Pan, TACO: temperature aware clock-tree optimization, in: *Proceedings of International Conference on Computer-Aided Design*, 2005, pp. 582–587.
- [11] A. Chakraborty, K. Duraisami, A. Sathanur, P. Sithambaram, L. Benini, A. Macii, E. Macii, M. Poncino, Dynamic thermal clock skew compensation using tunable delay buffers, *IEEE Trans. Very Large Scale Integr. Syst.* 16 (6) (2008) 639–649.
- [12] A. Gupta, N. Dutt, F. Kurdahi, K. Khouri, M. Abadir, Thermal aware global routing of VLSI chips for enhanced reliability, in: *Proceedings of the 9th International Symposium on Quality Electronic Design*, 2008, pp. 470–475.
- [13] K. Lu, D. Pan, Reliability-aware global routing under thermal considerations, in: *Proceedings of the 1st Asia Symposium on Quality Electronic Design*, 2009, pp. 313–318.
- [14] Y. Han, I. Koren, Simulated annealing based temperature aware floorplanning, *J. Low Power Electron.* 3 (2007).
- [15] A. Gupta, N. Dutt, F. Kurdahi, K. Khouri, M. Abadir, LEAF: a system level leakage-aware floorplanner for SoCs, in: *Proceedings of the 2007 Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2007, pp. 274–279.
- [16] C.-H. Tsai, S.-M. Kang, Cell-level placement for improving substrate thermal distribution, *IEEE Trans. Computer-Aided Des. Integr. Circuits Syst.* 19 (2) (2000) 253–266.
- [17] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, M. Stan, HotSpot: a compact thermal modeling methodology for early-stage VLSI design, *IEEE Trans. Very Large Scale Integr. Syst.* 14 (5) (2006) 501–513.
- [18] ITRS, International Technology Roadmap for Semiconductors. <http://www.itrs.net/>, 2007.
- [19] W.C. Elmore, The transient response of damped linear networks with particular regard to wideband amplifiers, *J. Appl. Phys.* 19 (1) (1948) 55–63.
- [20] J. Black, Electromigration – a brief survey and some recent results, *IEEE Trans. Electron Devices* 16 (4) (1969) 338–347.
- [21] T.-C. Chen, Y.-W. Chang, Modern floorplanning based on B*-tree and fast simulated annealing, *IEEE Trans. Computer-Aided Des. Integr. Circuits Syst.* 25 (4) (2006) 637–650.
- [22] MCNC Benchmark Netlists, Available at: http://lyle.smu.edu/~manikas/benchmarks/mcnc_benchmark_netlists.html, 2012.



Andreas Thor Winther is an IC design engineer at Knowles Electronics, Denmark. He received the MS degree in electrical and computer engineering from Technical University of Denmark in 2011.



Wei Liu received the BS degree in Computer Science from Zhejiang University, China, in 2005 and the MS and PhD degrees in electrical and computer engineering from Technical University of Denmark in 2007 and 2011, respectively. From 2011 to 2012, he was a research assistant with the Department of Electrical and Computer Engineering of Politecnico di Torino, Italy. His research interests include design automation of electrical circuits, thermal analysis and optimization, design for low power and VLSI design. He is now a IC design engineer at Oticon, Denmark.



Alberto Nannarelli is an associate professor at the Technical University of Denmark. He graduated in electrical engineering from the University of Roma “La Sapienza”, Italy, in 1988 and received the M.S. and the Ph.D. in electrical and computer engineering from the University of California at Irvine in 1995 and 1999, respectively. He worked for SGS-Thomson Microelectronics and for Ericsson Telecom as design engineer and for Rockwell Semiconductor Systems as a summer intern. From 1999 to 2003 he was with the Department of Electrical Engineering, University of Roma “Tor Vergata”, Italy, as a post-doc researcher. His research interests include computer arithmetic, computer architecture, and VLSI design.



Sarma Vrudhula joined Arizona State University in 2005 as the Consortium for Embedded Systems (CES) Chair Professor in the Computer Systems Engineering. He is also the Director of the NSF Center for Low Power Electronics, which he established in 1996. CLPE is supported by the NSF, the State of Arizona and companies in the microelectronics industry. Vrudhulas research and teaching interests are in VLSI CAD for low power; energy management and energy efficient computer design; thermal management in computer systems; logic synthesis and verification; statistical performance and power optimization for VLSI; and graph theoretic techniques for VLSI layout. He has published more than 120 papers in peer-reviewed conferences and journals.