Thermal-Induced Leakage Power Optimization by Redundant Resource Allocation

Min Ni and Seda Ogrenci Memik
Electrical Engineering and Computer Science
Northwestern University, Evanston, IL
{mni166, seda}@ece.northwestern.edu

ABSTRACT

Traditionally, at early design stages, leakage power is associated with the number of transistors in a design. Hence, intuitively an implementation with minimum resource usage would be best for low leakage. Such an allocation would generally be followed by switching optimal resource binding to achieve a low power design. This treatment of leakage power is unaware of operating conditions such as temperature. In this paper, we propose a technique to reduce the total leakage power of a design by identifying the optimal number of resources during allocation and binding. We demonstrate that, contrary to the general tendency to minimize the number of resources, the best solution can actually be achieved if a certain degree of redundancy is allowed. This is due to the fact that leakage is strongly dependent on the on-chip temperature profile. Distributing activity over a higher number of resources can reduce power density, remove potential hotspots and subsequently minimize thermal induced leakage. On the other hand, using an arbitrarily high number of resources will not yield the best solution. In this paper, we show that there is a power density, hence, temperature, at which the total leakage power will reach its optimal value. Such an optimal resource number can be a better starting point for the subsequent switching-driven low power binding. We also present a high-level power density-aware leakage model. Based on the estimates by this model, we optimize the total leakage power by 53.8% on average compared to the minimum resource binding, and 35.7% on average compared to a temperature-aware resource binding technique.

1. INTRODUCTION

Due to technology scaling, the share of leakage power in the total power budget is on the rise. Supply voltage levels are lowered with each technology generation, which in turn necessitates lowering of the threshold voltage levels of devices in order to maintain low delay. Leakage increases exponentially with decreasing threshold voltage levels. As a result, leakage power starts to become significant, sometimes even dominant in total power budgets, which could be up to 50% of the total power [5].

A plethora of techniques to reduce leakage power have been proposed in literature. Majority of these techniques focus on the gate or transistor-level optimizations. Assigning different threshold and/or supply voltages to transistors or gates, together with simultaneous gate sizing [6, 11, 14, 16] is one of the most popular techniques for both standby and operating mode leakage optimization. Other techniques, such as using sleep transistors to put the circuit into sleep mode whenever it idles for a certain period [5] are also used for reducing standby state leakage power. All these techniques are derived from the observation that the subthreshold leakage current, which is the most significant one among the four main sources of leakage current [12], can be expressed by the following equation [15]:

$$I_{sub} = \frac{W}{L}\mu {v_t}^2 C_{sth} e^{(V_{GS}-V_T+\eta V_{DS})/(\eta V_t)}(1-e^{-V_{DS}/V_t}) \qquad (1)$$

Therefore, subthreshold current is a function of device size, supply voltage, temperature, and other process parameters, such as threshold voltage (V_t). Most of the above techniques trade-off leakage power with the design complexity to manipulate the threshold voltage and supply voltage by adding extra power control components.

Another aspect of leakage is related to dynamic conditions such as temperature. Leakage has a superlinear dependency on temperature. Fallah et al. reported that the share of leakage power can increase from 6% at the ambient temperature to as high as 56% of total power at 110^{o}C [5]. Another study reported that the leakage power in an embedded processor can increase by about 30% due to thermal-induced leakage [8].

Temperature on a chip is itself a function of various parameters, where the foremost factors are the power density on the chip and the properties of the package. The power density, hence temperature, will continue increasing in future technologies according to α-power law [13]. The abovementioned techniques for leakage optimization generally do not address the power density on a chip. Often times, they can in fact exacerbate the effects of power density while aiming to consolidate activity on fewer localized resources (for instance in an effort to place parts of the chip in sleep mode and channel computation towards a selected subset of components).

In this work, we investigate a technique to consider the impact of resource selection on the overall power density and consequently on thermal-induced leakage in future technology nodes. Resource allocation and binding is a proper stage during high-level synthesis to consider the potential impact of area on power density. At that stage it is decided how many resources and which type of resources will be utilized in the design. More resources will result in larger area and most likely in lower power density. In this paper, we are trying to establish an effective tradeoff between the number of resources and the total leakage power. There exists an optimal point where the amount of resources used yields the most favorable power density, which in turn results in the least thermal-

ICCAD 2006, November 5–9, 2006, San Jose, California, USA.

induced leakage power. Our study reveals that often times in order to reach this point the amount of resources should be higher than the amount which would be sufficient to satisfy the same performance constraint. A judicious introduction of redundant resources when there is need to relieve power density, will ultimately help reduce thermal-induced leakage and total leakage significantly.

The major difference between our work and other hotspot-moving resource allocation techniques is that in almost all the hotspot-moving techniques [9, 10] a threshold temperature is assumed. Based on this given constraint, they are trying to make sure that there are no places on the chip where the static temperature will exceed that threshold value. However, our work is to decide what this threshold temperature is, in order to optimize performance, *e.g.*, to optimize leakage power in our work. Other low power resource binding techniques [2–4] which consider switching power can be supported by our initial allocation. In this way, the low power resource binding would address two components within two stages. The first stage is to find the optimal resource number, resulting in best power density and temperature, such that the leakage power will be minimized. The second stage is to optimize the dynamic power and maintain control over thermal behavior by existing thermal-driven techniques and switching-driven techniques based on the results of the first stage.

One reason rendering this distinction feasible is that with different starting points (different number of resources and temperature constraints), the optimal dynamic power considering switching activity does not vary significantly [2]. Experimental results reported in past work [2] show that for a given design example the optimal dynamic power for five resources is 70.882, for six resources 67.872, and for seven resources it is 65.514. Only less than 5% change is observed when adding more resources. Often times, introduction of redundancy to the resource set might in fact help reduce the impact of conflicts due to dependencies and scheduling compatibility and create more opportunities for the switching optimal binding to find a slightly lower switching assignment, which reduces the dynamic power. Therefore, we can safely conclude that the optimal dynamic power of functional units will not increase when we add resource redundancy to achieve the optimal leakage power. On the other hand, the leakage power is much more sensitive to the selection of the resource set than dynamic power. Even adding one more resource may probably reduce the leakage power by more than 50%, because leakage power is strongly coupled with power density and in turn the chip temperature. Therefore, the two-stage optimization is meaningful and effective. We will address mainly the first stage, i.e. power density and resulting thermal-induced leakage optimization during allocation.

The rest of this paper is organized as follows. Section 2 describes the leakage power estimation model we will use in this paper. Main ideas of our low power resource binding technique are discussed in Section 3. Section 4 presents our experimental flow and results. Conclusions are given in Section 5.

2. LEAKAGE ESTIMATION MODEL

Before we start to find the optimal number of resources for leakage power, it is necessary to establish first a simple model for leakage estimation. It is important to emphasize that the intention of this model is not to compute exact temperature levels. This model intends to establish the prevailing trend linking power density and temperature and subsequent expected rate of increase in leakage. Once we establish this trend it will be a reasonable tool for us to search for the best resource allocation. Most importantly, it will help us identify the point where the rate of increase in leakage power due to addition of redundant resources will finally counterbalance the decrease in thermal-induced leakage due to reduction of power density after addition of each redundant resource. Up until that point addition of redundant resources and distribution of operations onto them will be expected to progressively improve power density and hence, the total leakage.

We need to establish the following in order to achieve this goal. First, we need to have the means to compare the relative leakage of different modules at ambient temperature. For this purpose, we have used transistor-level (HSpice) simulation of simple building blocks encountered within the resources in our library to obtain leakage power values for each resource. After simulating the leakage power for a simple structure, such as a transistor or a gate, we scale it to obtain ambient leakage power for individual modules. Each module implementation in our library requires a customized scaling factor. The scaling factor not only depends on the number of transistors in the module, but also on the sizing of individual transistors and the actual threshold voltage used in the design. We used empirical data [12] to derive the leakage power scaling factors of each module type, under the basic idea that leakage power becomes a certain fraction of total power at a given temperature.

Next, we establish the trends to represent the rate of increase in leakage in response to a change in temperature analytically. Instead of using Equation 1 directly, we use *Lagrange's interpolation formula* to implement the curve fitting, as shown in Equation (2),

$$y = L_p(x) = \sum_{j=0}^{p} \frac{\prod_{i \neq j}^{p}(x - x_i)}{\prod_{i \neq j}^{p}(x_j - x_i)} y_i \tag{2}$$

where (x_i, y_i) is the leakage point obtained from the Hspice simulation. Using analytic leakage formula such as Equation 1 directly is also feasible. However, we prefer to let the simulation engine to decide the physics details and then fit the experiment data exactly by *Lagrange's* interpolation.

Having obtained the analytical form of the leakage power trend, we can use a numerical method to establish the relationship between power density and temperature. At this point, we turn our attention towards the two most important factors that affect the thermal behavior: the power density P/A and the heat transfer coefficient.

Equation 3 [7] illustrates the relationship between power density, heat transfer coefficient (i.e. thermal properties of packaging), and temperature.

$$T = T_a + h\left(\frac{P}{A}\right) \tag{3}$$

where T_a is the ambient temperature, P is the total power dissipation, A is the area of design, and h is the heat transfer coefficient as used in the heat transfer theory. The value of h represents how well the chip package can dissipate the heat. A large value of h always implies poor cooling package. An example of h value is 4.75cm$^2 \cdot {}^o$C/W, based on the operating chip temperature of 120^oC degree for the 180nm technology [7]. We will show that for every power density level, there is always a maximum package heat coefficient (thus poorest acceptable package). Using a packaging, which has an even larger heat coefficient than this will be likely to cause thermal run-away.

Figure 1 illustrates the relationship between average power density across a given chip, the heat coefficient of the package and the expected steady state temperature. In this figure, the lines starting from the origin represent the heat transfer ability of the package. It is proportional to the chip temperature. High temperature results in need for fast heat dissipation by the package. The other three curves represent the different power density levels of the chip. The bending of the curve reflects the fact that the leakage power has become a significant part of total power consumption and the leakage power has a superlinear dependency on temperature. When the heat generation equals the heat dissipation, the chip temperature will become steady. Therefore, the intersection point of both power density curve and package heat coefficient curve represents the steady state point. It can be seen from Figure 1 that for power density, the higher it is, the higher steady temperature it will reach with respect to the same packaging configuration.

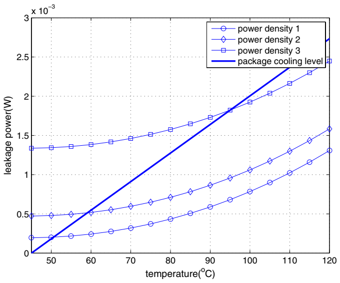

Figure 1: Establishing the relationship between temperature and power density.

This relationship between power density and package heat coefficient is the base for our leakage estimation model. The analytical formula for calculating the steady state temperature is,

$$\frac{A}{h}(T_x - T_a) = (\sum_{j=0}^{p} \frac{\prod_{i \neq j}^{p}(T_x - x_i)}{\prod_{i \neq j}^{p}(x_j - x_i)} y_i \cdot f \cdot n + P_d) \tag{4}$$

where A is the total area of resources, n is the number of resources, and f is the leakage power scaling factor. In our experiments, f is 250 for a 16-bit multiplier module and 80 for a 32-bit adder module. It is approximately proportional to the area of the module. P_d represents the dynamic power. Our purpose is to solve for the steady state temperature T_x from this equation. Before that, we first show that it is the superlinear relationship between leakage power and temperature that leads to our conclusion that there exists an optimal number of resources (corresponding to an optimal temperature).

LEMMA 1. *The steady state temperature T_x monotonically decreases with the increasing number of resources n if the Lagrange formula is linear.*

PROOF. After rearranging Equation 4, we have

$$T_x = T_a + h \cdot (\frac{P_d}{na_0} + \frac{L_1(T_x)}{a_0/f}) \tag{5}$$

where P_d is the dynamic power, which is constant as we discussed above. n is the number of resource, a_0 is the area of one resource. Using linear Lagrange interpolation, we substitute $L_1(T_x) = ax + b$ into equation (5) and solve for T_x,

$$T_x = \frac{\frac{a_0}{f} T_a + bh + h\frac{P_d}{n}}{\frac{a_0}{f} - ah} \tag{6}$$

it can be seen that T_x decreases monotonically when n increases. □

LEMMA 2. *The leakage power in the form of $n \cdot L_1(T_x)$ monotonically increases with increasing number of resources.*

THEOREM 1. *The leakage power in the form of $n \cdot L_p(T_x)$, $p \neq 1$, is not a monotonic function. It obtains a minimal value at some resource number n^*.*

PROOF. We only analyze the situation where $p = 2$ here. Higher order Lagrange interpolation can be analyzed numerically in the similar way. Suppose $L_2(T_x) = ax^2 + bx + c$, substitute it into Equation (5),

$$T_x = \frac{\frac{a_0}{f} - bh + \sqrt{(\frac{a_0}{f} - bh)^2 - 4ah(hc + \frac{a_0}{f} T_a + P_d h/n)}}{2ah} \tag{7}$$

Therefore the total leakage power in the form of $n \cdot L_2(T_x)$ becomes,

$$P_l = n \cdot L_2(T_x) = \sqrt{s_1 n^2 + s_2 n} + t_1 n + t_2 \tag{8}$$

where s_1, s_2, t_1, t_2 are some coefficients. The optimal solution can be found by setting the derivative to zero. It is in the form of a quadratic equation. □

We proved theoretically that there exists an optimal number of resources which minimizes the total leakage power. In the next section we will show how to reach the optimal solution by a numerical method.

3. REDUNDANT RESOURCE ALLOCATION FOR LEAKAGE OPTIMIZATION

Our main goal is to achieve low power density by introduction of redundant resources in the search of the optimal point where the reduction in thermal-induced leakage still brings a higher benefit compared to the additional leakage due to the redundant resources.

However, deriving an analytic formula for the optimal number of resources is only possible for 2-degree Lagrange interpolation. In reality, we will use at least a 10-degree Lagrange formula (therefore at least 10 experiment data points) in order to maintain good accuracy. Another way to solve this problem is to perform an incremental search in the solution space. This is feasible because of the number of resources will take discrete values. The main algorithm is illustrated in Figure 2.

```
Algorithm  Redundant Resource Allocation
Input:   Resource library with power
          characterization, resource scheduled DFG,
          minimum required leakage power reduction a%
Output:  Number of resources after redundant
          allocation

 For each resource type
 Do
   find_avg_dynamic_power();
   find_resnum_bounds();
   find_package_parameter();
   n = min_resource_number;
   While ($\Delta P_l$ > a%)
     add_resource_redundancy(n);
     steady_temperature = secant(n, $F(T_x)$);
     $\Delta P_l$ = $\frac{P_l(T'_x) - P_l(T_x)}{P_l(T'_x)}$;
     $T'_x = T_x$;
   End
   Return number of resources n in new allocation;
 End
```

Figure 2: Pseudocode of the redundant resource allocation algorithm.

The basic idea of this algorithm is to increment the number of resources until the benefits of leakage power reduction become less than some expectation constraint. In each iteration, we use a numerical method to solve equation (4). In this equation, T_x is the variable. Before we can solve it, we have to know the dynamic power value P_d and package heat coefficient h. Leakage power scaling factor f is derived empirically [12].

Therefore, based on the information given by the scheduled DFG, we first calculate the average dynamic power for each resource type. At such a high level, we have to ignore the thermal coupling between different resources because we have no physical position information available. However, our methodology is still applicable if thermal coupling information is available. The new steady state temperature can be calculated by combining our results and the information of thermal coupling. Moreover, ignoring coupling only underestimates the total leakage power, because when one resource temperature reduces due to resource redundancy, other resources can also reduce their temperature through thermal coupling. In other words, we can at least get as much leakage reduction as our result shows. Higher benefits can be expected if thermal coupling is introduced into the leakage estimation model. The lower bound and upper bound for the number of resources can also be derived from these DFG files and incorporated into the search. The next step is to decide the package heat coefficient according to different power density levels. Using a very low package heat coefficient h is always good, because the chip temperature can be controlled effectively. However, such very low h always implies high packaging cost. Therefore, we will find the lowest cost (highest h) feasible package for each binding based on the relationship between power density and package heat coefficient. This packaging characteristics will be used in our experiments.

We will discuss estimating the average dynamic power in subsection 3.1. The algorithm for identifying the lowest cost package is presented in subsection 3.2. In subsection 3.3 we will show how to use a numerical method to obtain the expected steady state temperature, and relate it to the leakage trends.

3.1 Average Resource Dynamic Power

We assume that each resource will consume a typical average dynamic power for executing one operation. In other words, the total dynamic power will be represented by a constant after the scheduled DFG is given. The total power will be decided by the total number operations that will be executed in a given number of control steps. This approximation helps us focus on the contribution of leakage power. This is a reasonable assumption as we have discussed in Section 1. Also, at the high-level synthesis stage input switching probabilities are highly unpredictable. Individual dynamic power consumptions of operations can be weighted with respective input switching behavior if an appropriate statistical model is provided.

We first derive a typical dynamic power value of the module P_0 by some existing power estimation technique. We have used the power estimations obtained after synthesizing different modules using Synopsys Design Compiler. Assume the signal toggle rate is TR. It represents how many logic transitions there are per unit time when the dynamic power is P_0. Given a scheduled DFG, which spans a total of m control steps and with the clock cycle time of the design being s, we can calculate the dynamic power of each operation as:

$$P_{opt} = \frac{P_0}{TR \cdot m \cdot s} \tag{9}$$

Dynamic power consumption per operation corresponds to the power consumption when there is only one operation scheduled on the resource within m control steps. By using this metric, we can scale the dynamic power of any resource by the total number of operations assigned to it.

3.2 Estimating the Package Properties

The chip temperature, hence leakage power, is highly related to the cooling package. Using an arbitrarily low h package will always guarantee a low temperature. However, it also means the package cost will increase. We show that for each power density level, there is a maximum h (minimum cost) package. If the h exceeds this maximum value, the package heat dissipation curve and the chip heat generation curve will not have any intersection, which means that the heat dissipation is always slower than heat generation. Eventually, the chip temperature will increase to an uncontrolled high level. This phenomenon is called thermal runaway. Mathematically, we can get the minimum cost h value when Equation (4) has only one root.

We use a binary search algorithm to find the maximum package coefficient. The basic idea in this algorithm is to find a point on the power density curve such that its tangent line intersects the zero point of the x-axis. We can select any two points as our initial values as long as one of them intersects the x-axis at a negative value and the other intersects at a positive value. The algorithm runs recursively, and finally stops when the intersection point is close enough to the zero point.

After getting the maximum package coefficient, we will decrease its by some constant value, *e.g.*, 10%, in order to make sure that it is safely far away from the thermal run-away condition, but still very low cost. This may also be needed to identify the applicable safe and lowest cost coefficient among a discrete set of values. We will use this package parameter in the process of estimating the steady state temperature level.

3.3 Steady State Temperature

The calculation of steady state temperature is basically to find the solution of a nonlinear equation. Newton-Raphson method can be a good candidate. However, this method is only applicable when the order of Lagrange interpolation is not too high.

Therefore, we use the secant method, which has the iteration expression as shown below.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - f(x_i)[\frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})}] \tag{10}$$

It substitutes the derivative value by a secant estimation. The convergence speed depends on how far the initial point is from the real solution. Therefore, finding a good starting point is critical in order to guarantee the running time of our algorithm.

One such good start point can be obtained by finding the intersection of two lines. One is the heat package dissipation line, the other is the simplified heat generation line by assuming that there is no leakage power.

$$T_x = T_a + h\frac{P_d}{A} \tag{11}$$

It can be seen analytically that this point is very near the solution. Starting from this initial point and searching in the positive direction, we can find the solution within a few iterations.

Having obtained the steady state temperature by the secant method, we use $P_l(T_x) = n \cdot \sum_{j=0}^{p} \frac{\prod_{i\neq j}^{p}(T_x - x_i)}{\prod_{i\neq j}^{p}(x_j - x_i)} y_i$ to calculate the total leakage power for a given resource allocation, that is, for certain number of resources.

4. EXPERIMENTAL RESULTS

4.1 Experimental Flow

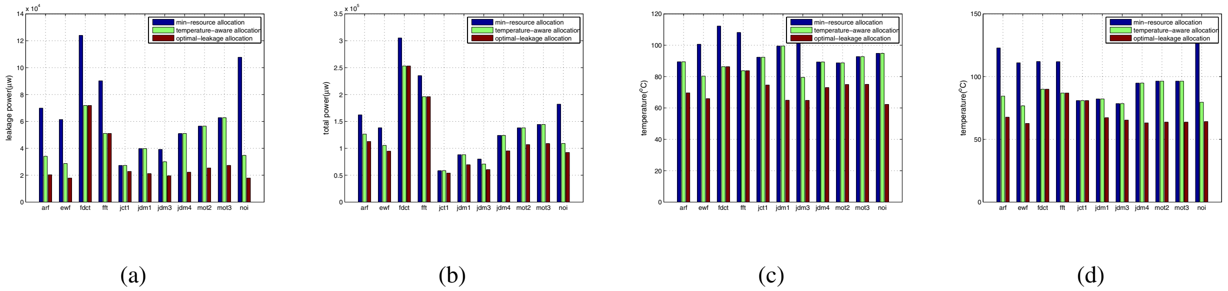

Figure 3: (a)Leakage power of our redundancy resource allocation technique compared with thermal-aware resource allocation technique and minimum resource number allocation; (b)Total power of our technique and other resource allocation techniques; (c)Average temperature of adders in three different resource allocation schemes; (d)Average temperature of multiplier in three different resource allocation schemes.

We used two types of functional units (adders and multipliers) to bind operations in a set of scheduled DFGs. The minimum number of resources required is determined by the compatibility between operations as dictated by the schedule. The maximum number of operations of the same type, which are scheduled in the same control step correspond to the minimum number of resources required of that type.

The area value and the average dynamic power consumption of each module type is obtained after synthesizing them using Synopsys Design Compiler with the tsmc 180nm library. We scale down these values to 70nm technology by full-scale methodology after synthesis.

4.2 Results

The relevant information regarding our benchmarks is given in Table 1. Our benchmark DFGs are extracted from popular DSP and multimedia kernels [1]. Their names are listed in the first column. The second column is the total number of operations of each type in these DFGs. The third column presents the minimum number of resources required by the schedule of each DFG. The remaining columns present the average dynamic power consumption estimated per adder and multiplier module during the execution of these DFGs, using the method described in Section 3.

Table 1: Properties and Relevant Information on the Scheduled DFGs

Schedule Name	Num. of Nodes [add,mul]	Minimum Resources [add,mul]	Dyn. Power μW per ADD	Dyn. Power μW per MUL
arf	[12,16]	[2,2]	534.19	3446.26
ewf	[26, 8]	[3,2]	659.89	4257.15
fdct	[26,16]	[4,4]	934.84	6030.96
fft	[26,16]	[3,3]	747.87	4824.77
jctrans1	[13,2]	[3,2]	801.29	5169.40
jdmerge1	[23,4]	[3,3]	659.89	4257.15
jdmerge3	[30,4]	[3,3]	487.74	3146.59
jdmerge4	[18,12]	[3,3]	509.91	3289.62
motion2	[26,14]	[4,3]	467.42	3015.48
motion3	[26,14]	[5,3]	467.42	3015.48
noise_est	[17,9]	[3,2]	659.89	4257.15

Figure 4 illustrates the trends for total leakage power of one resource type (multiplier in this case) with allocations of the resource in the same design. The most important observation is that there exists an optimal number of resources which achieves the least total leakage power. We have observed similar trends for all test cases. As we mentioned before, adding extra resources is not free. The total leakage power will start to increase after some point with further increase in number of resources. The sharpest leakage power reduction happens at high temperatures, *i.e.*, when using few resources at high power densities. At that point allocating one more resource impacts the power density and thermal-induced leakage most. As we introduce more and more redundancy the return diminishes. This is expected, since the thermal-induced leakage power only becomes significant at high temperature levels.

When there are more than one resource type in a DFG, we first add redundancy for the module with highest power density. Because such a module will be very likely to contain a hotspot leading to high thermal-induced leakage power.

In practice, we set a lower bound on leakage power reduction to accept the addition of a new resource. Only if adding further redundancy can reduce the leakage power by a percentage larger than a predefined level, we add an extra resource. In our experiment, we set the value to be 20% for every additional resource. This value plays the role of judging how important power is compared to area. However, as seen from our results, there is an optimal number of resources, which can achieve minimum total leakage power. In the power-critical design, we can perform a full search and use as many resources as that optimal number indicates. Otherwise, if we choose to stop the search earlier we might not have reached that optimal number yet.

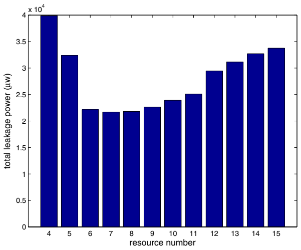

Figure 4: Trends in leakage for different allocations of the multiplier module for FFT design.

Figure 3 illustrates our results. We compared our results against the thermal-aware resource binding techniques [9, 10]. These techniques try to meet a temperature constraint while using minimum number of resources during binding. The temperature constraint is 100^{o}C, exactly the same as what has been used in these works. As we can see from the results, we achieved at most 56.5%, on average 35.7%, leakage power reduction compared to thermal-aware resource binding technique. The difference is more dramatic if we compare our results with non-thermal-aware single stage dynamic power optimization (labeled as min-resource allocation in our figure). The maximum leakage reduction is 71.0%, and the average reduction is 53.8% compared to non-thermal-aware resource allocation and binding. We also depict the total power of all these benchmarks in Figure 3(b).

Figure 3(c) and 3(d) demonstrate the steady state temperature which three different power optimization techniques will reach. As seen in both figures, the chip temperature will probably increase to an arbitrarily high level in non-thermal-aware resource binding technique. On the other hand, for thermal-aware resource binding techniques, it guarantees that no temperature exceeds the given constraint, *i.e.*, 100^oC in our experiments. This constraint is assumed to be given in their techniques. Our technique actually finds one of such constraints that minimize the total leakage power. In other words, if we give this constraint (*e.g.*, 85^oC) to the thermal-aware technique, it will eventually give out the same results as our technique. Therefore, our leakage optimal resource allocation technique is a good guide for other thermal-driven resource binding techniques to achieve an optimal total power reduction.

Of course, these significant leakage power reduction is achieved by trading off area. The same trade-off occurs for the thermal-aware resource binding techniques as well. They also add new resources when some resource temperature exceeds the given constraint. However, they stop adding more once the temperature constraint is satisfied. We claim that we should not stop until the total leakage power reaches its absolute minimum point and our results presented above concur with our claims. The number of resources used for non-thermal-aware, thermal-aware, and our leakage-optimal resource allocation and binding techniques are shown in Table 2.

Table 2: Number of resources for three resource allocation and binding techniques

Benchmark	minimum resource	thermal aware	leakage optimal
arf	[2, 2]	[2, 3]	[3, 4]
ewf	[3, 2]	[4, 3]	[5, 4]
fdct	[4, 4]	[5, 5]	[5, 5]
fft	[3, 3]	[4, 4]	[4, 4]
jctrans1	[3, 2]	[3, 2]	[4, 2]
jdmerge1	[3, 3]	[3, 3]	[5, 4]
jdmerge3	[3, 3]	[4, 3]	[5, 4]
jdmerge4	[3, 3]	[3, 3]	[4, 5]
motion2	[4, 3]	[4, 3]	[5, 5]
motion3	[5, 3]	[5, 3]	[6, 5]
noise_est	[3, 2]	[3, 3]	[5, 4]

In this set of results, we observe that leakage optimal allocation uses the a maximum of 67% more resources in comparison to thermal-aware resource binding. On average, our technique uses 33% more resources. We only consider the area of multipliers because it is dominant in comparison to adders. In our experiments, the largest increase in number of resources for the thermal-driven technique is 50%, while it is 20% on average. Compared to thermal-driven resource allocation and binding, our overhead is still within a reasonable range. It worths noting that that 33% more resources does not mean that the area of the design will increase as much as 33%. In fact, more area is occupied by memory, communication resources, and controller compared to the datapath in modern chips. These components may require as much as 50-80% of total area cost. However, due to the high activity of the datapath, it consumes much more power per unit area than other parts of the circuit. That is the reason why most hotspots occur within the datapath and the leakage implications of the datapath at high temperatures becomes significant.

5. CONCLUSION

It is necessary to budget and optimize for the leakage power as early as possible in the design flow of future technology. In this paper, we claim that the low power high-level resource allocation and binding technique should implemented in two stages. The first stage is for optimizing the leakage power, because leakage power is highly sensitive to the number of resources allocated (hence temperature). On the second stage, dynamic power optimization considering switching activity is implemented, taking the results in the first stage as its initial condition. We present techniques for optimizing the total leakage power by adding resource redundancy. On average, our technique can save 35.7% total leakage power compared to other thermal-aware techniques, and 53.8% compared to non-thermal-aware techniques.

6. ACKNOWLEDGMENT

This work has been funded by the National Science Foundation Career Award CNS-0546305.

7. REFERENCES

[1] W. H. Mangione-Smith C. Lee, M. Potkonjak. MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems. In *International Symposium on Microarchitecture (MICRO)*, Research Triangle Park, NC, November 1997.

[2] J. Chang and M. Pedram. Register allocation and binding for low power. In *Design Automation Conference (DAC)*, San Francisco, CA, June 1995.

[3] J. Chang and M. Pedram. Module assignment for low power. In *Conference on European Design Automation Conference*, Geneva, Switzerland, June 1996.

[4] A. Davoodi and A. Srivastava. Effective graph theoretic techniques for the generalized low power binding problem. In *nternational Symposium on Low Power Electronics and Design (ISLPED)*, Seoul, Korea, August 2003.

[5] F. Fallah and M. Pedram. Standy and Active Leakage Current Control and Minimization in CMOS VLSI Circuits. *IEICE Trans. on Electronics, Special Section on Low-Power LSI and Low-Power IP*, E88-C(4):509–519, April 2005.

[6] F. Gao and J. P. Hayes. Gate Sizing and Vt Assignment for Active-Mode Leakage Power Reduction. In *ICCD*, 2004.

[7] S. Im and K. Banerjee. Full Chip Thermal Analysis of Planar (2-D) and Vertically Integrated (3-D) High Performance ICs. In *IEDM*, pages 31.4.1–31.4.4, 2000.

[8] H. S. Kim, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin. Adapting Instruction Level Parallelism for Optimizing Leakage in VLIW Architectures. In *Conference on Languages, Compilers, and Tools for Embedded Systems (LCTEC)*, San Diego, CA, June 2003.

[9] R. Mukherjee, S. Ogrenci Memik, and G. Memik. Peak Temperature Control and Leakage Reduction during Binding in High-Level Synthesis. In *International Symposium on Low Power Electronics and Design (ISLPED)*, San Diego, CA, August 2005.

[10] R. Mukherjee, S. Ogrenci Memik, and G. Memik. Temperature-Aware Resource Allocation and Binding in High-Level Synthesis. In *Design Automation Conference (DAC)*, Anaheim, CA, June 2005.

[11] D. Nguyen, A. Davare, M. Orshansky, D. Chinnery, D. Chinnery, B. Thompson, and K. Keutzer. Minimization of Dynamic and Static Power Through Joint Assignment of Threshold Voltages and Sizing Optimization. In *ISLPED*, Seoul, Korea, August 2003.

[12] M. Pedram and S. Nazarian. Thermal Analysis of VLSI Circuits: Basic Principles and Design Implications. *under review*, 2005.

[13] T. Sakurai and A. R. Newton. Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas. *IEEE JSSC*, April 1990.

[14] X. Tang, H. Zhou, and P. Banerjee. Leakge Power Optimization With Dual-Vth Library In High-Level Synthesis. In *DAC*, pages 202–207, Anaheim, CA, June 2005.

[15] U.C.Berkeley. BSIM3v3.1 SPICE MOS Device Models. *On-line resources*, 1997.

[16] L. Wei, Z. Chen, M. Johnson, and K. Roy. Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits. In *DAC*, pages 489–494, San Francisco, CA, June 1998.