社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ARC－173 (15)
2007／6／1

相互結合網簡単化を考慮した遺伝的アルゴリズムに基づく電源・しきい値電圧割当

ウィシディスーリヤ ハシタ ムトゥマラ　張山 昌論　亀山 充隆

東北大学 大学院情報科学研究科

あらまし 本論文では，演算器（FU）における消費エネルギーおよびFU間の配線における消費エネルギー，両エネルギーを統合した総合消費エネルギー最小化のためのハイレベルシンセシスを提案する．FUの静的・動的消費エネルギー削減のために複数電源・しきい値電圧割り当てを利用する．大規模問題を高速に解くために遺伝的アルゴリズムに基づく解法を用いる．データ転送の種類が同一であるデータ転送ではすべての起点ノードを同一の演算器に，かつ，すべての終点ノードを同一の演算器に割り当てることにより，それら演算器間の配線を共有でき，配線消費エネルギー削減できる．提案手法により消費エネルギー50％を削減できることが実験的に確かめられた．

キーワード ハイレベルシンセシス，低消費電力，相互結合網，遺伝的アルゴリズム

GA-Based Assignment of Supply and Threshold Voltages and Interconnection Simplification for Low Power VLSI Design

Hasitha M. WAIDYASOORIYA, Masanori HARIYAMA, and Michitaka KAMEYAMA

Graduate School of Information Sciences, Tohoku University

Abstract This paper presents a method to minimize the total energy consumption under time and area constraints, considering interconnection and functional unit (FU) energy. Multiple supply and threshold voltage scheme is used to minimize the static and dynamic energy in the FUs. A genetic-algorithm-based-search method is proposed for the energy consumption minimization problem, so that near-optimal solution can be found in a reasonable time for large-size problems. Interconnection energy reduction is achieved by increasing the sharing of interconnections among FUs. Experimental results show that up to 50% of energy can be saved by our proposed method.

Key words high-level synthesis, low power, interconnection, genetic algorithm

1. Introduction

In recent years, low power/energy has become a primary concern in VLSI design. As a result, different power reduction techniques has been proposed, focusing on various areas of power consumption in VLSI processors. Some of those focused on dynamic energy [1], [2] in functional units (FUs) and some others focused on static energy in FUs [3]. Interconnection energy reduction techniques based on interconnection simplification has been proposed in [4].

Dynamic energy has a quadratic relationship with the supply voltage. Therefore, an efficient way to reduce the dynamic energy is to use a low supply voltage. However, this increases the delay time. Therefore, a better way to reduce the dynamic energy while maintaining the time constraint is to use low supply voltages in non-critical paths and high supply voltage in the critical path [1], [2].

Static energy due to leakage current is a major concern in deep sub-micron process. The scaling of supply and threshold voltages has reduced the threshold voltage significantly. This has coursed a severe increase in subthreshold leakage current, because there is an exponential relationship between the leakage current and subthreshold voltage. A dual threshold voltage scheme, that uses low threashold voltage in the critical path and high threshold voltage in non-critical paths, is a very efficient and popular way to reduce the static energy [3].

However, above-mentioned methods comes with an additional overhead penalty such as additional multiplexers, wires, functional units, level converters, power supply lines, etc. This overhead of extra functional units and multiplexers can increase the interconnection complexity as well as the interconnection energy. As the process technology proceeds, the interconnection energy consumption is increasing and comparable to the FU energy consumption. As a result, interconnection energy is a severe problem in VLSI design.

Interconnection simplification based on the regularity of the computation patterns is proposed in [4]. Even though this effectively simplifies the interconnections, it does not consider the leakage power due to the increased number of functional units. This leads to a greater leakage power.

Therefore, to reduce the total energy consumption, functional unit energy and interconnection energy have to be considered together. This paper presents an efficient method based on the genetic algorithm (GA) to assign supply and threshold voltages to minimize the FU energy consumption. Then, it performs interconnection simplification to minimize the interconnection energy. After the circuit is optimized for both FU and interconnection energy, the best solution with the minimum total energy can be found.

Experimental results demonstrate that our method reduces up to 50% of total energy consumption in comparison with the conventional method considering only the energy consumption of FUs.

2. Problem description

2.1 Architecture model

Our targeted architecture model has functional units with two different supply and threshold voltages. Therefore each operation has four different modules as shown in the module library in table 1. Each functional (FU) unit has a register or a register file in their input ports to store the inputs they need. Note that, each register uses gated clock to avoid unnecessary data writings. Level converters are used to drive the outputs of low-supply-voltage functional units to high-supply-voltage functional units. Therefore we also have a set of high-supply voltage functional units that include level converters in their input ports. We consider a multiplexer based interconnection network. All the functional units has dedicated input and output buses. The supply voltages of the registers and multiplexers are depend on the voltage of their input signals. If all the inputs are in high voltage, we use high voltage registers to minimize the delay. Otherwise low voltage modules are used.

Assumption 1: Operations are driven by a clock signal. One clock cycle is equal to one step.

Assumption 2: Area and delay of the level converters are negligible.

Assumption 3: Delays of writing data into Registers and reading from registers are negligible. Data transfer delays are also negligible.

Table 1 Module library

Operation	FU type	V_{dd}	V_{Th}	Area	Delay	Dynamic energy	Static energy
ADD	FU_1	3.3	High	1	1	20	2.0
	FU_2	3.3	Low	1	1	18	0.18
	FU_3	1.8	High	1	2	6	0.6
	FU_4	1.8	Low	1	4	5	0.08
MUL	FU_5	3.3	High	16	2	240	24.0
	FU_6	3.3	Low	16	3	220	0.22
	FU_7	1.8	High	16	5	65	6.5
	FU_8	1.8	Low	16	10	55	0.55

Assumption 4: All the functional units have two input ports.

2.2 Energy estimation

In the energy consumption minimization problem, a data flow graph (DFG) with n nodes is considered. Our objective is to minimize the total energy consumption under area and time constraints. For the interconnection energy consumption, we assume that the number of *fan-outs* and *fan-ins* have a linear relationship with the energy, based on the experimental observations. The objective function is given by

$$\sum_{i=1}^{n} DE_{FU(\text{node } i)} + \sum_{i=1}^{m} (SE_{FU_i} \times N_i) + IE \qquad (1)$$

where $DE_{FU(\text{node } i)}$ is the dynamic energy dissipation per operation of the FU that executes node i. The number of FU types is m. The term N_i is the total number of idle control steps for FUs of type FU_i. The static energy dissipation of the FU_i per control step is denoted by SE_{FU_i}. The term IE is the interconnection energy consumption given by

$$IE = \alpha \times (fan{-}in + fan{-}out) \times V_{dd}^2 \qquad (2)$$

The term α is a constant that depends on the architecture. It shows the ratio of interconnection energy to FU energy. Fig. 1 shows the experimental results obtained using spice simulation under $0.18\mu m$ CMOS design rules. It shows there is a linear relationship between the objective function and the actual energy consumption. Since there is a linear relationship exists between the objective function and the actual energy consumption, the total energy can be reduced by minimizing the objective function.

3. GA-based efficient search method

3.1 Overview

A GA is a stochastic search technique [1] based on the mechanism of natural selection and natural genetics. It starts with an initial set of random solutions called population. Each individual in the population is called a chromosome which represents a solution to the problem at hand.

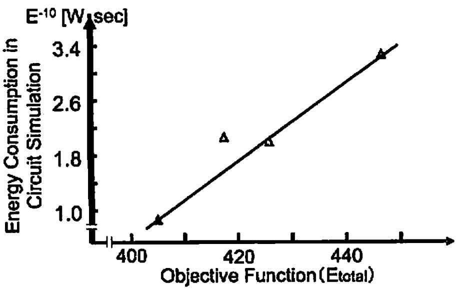


Fig. 1 Objective function vs actual energy consumption

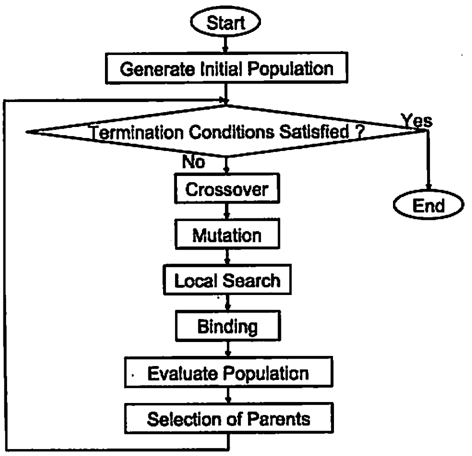
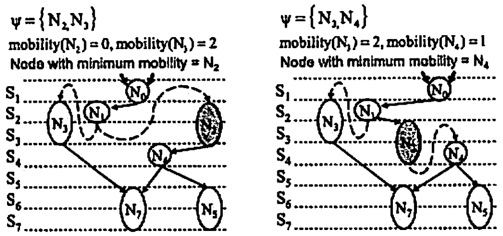


Fig. 2 Genetic algorithm based search method

The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated, using some measures of fitness. In order to create the solutions for the next generation, new chromosomes, called children are formed by either (i) merging two chromosomes from current generation using a crossover operator or (ii) modifying a chromosome using a mutation operator. A new generation is formed by selecting some of the parents and rejecting others, according to their fitness values, to keep the size of the population constant. Fitter chromosomes have higher probabilities of being selected. After processing for several generations, the algorithm converges to the best chromosome, which hopefully represents the optimal or suboptimal solution to the problem. Figure 2 shows the flow chart of the GA based search algorithm.

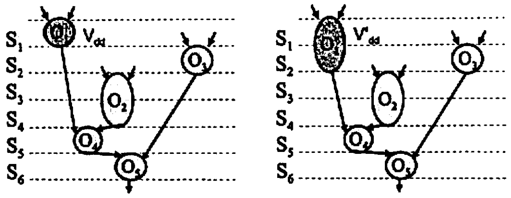
3.2 Reducing invalid chromosomes

The main problem in the GA is the formation of invalid chromosomes. This decreases the variety of the chromosomes. As a result, the local optima does not improve after processing for a small number of generations. Therefore, conventional GA approaches do not give a good solution for the energy consumption minimization problem. There are 3 types of invalid chromosomes.


$\psi = \{N_2, N_3\}$
mobility(N_2) = 0, mobility(N_3) = 2
Node with minimum mobility = N_2
$\psi = \{N_3, N_4\}$
mobility(N_3) = 2, mobility(N_4) = 1
Node with minimum mobility = N_4
S_1
S_2
S_3
S_4
S_5
S_6
S_7
N_0
N_1
N_2
N_3
N_4
N_5
N_7

(a) (b)

Fig. 3 Re-scheduling of node N_1



(a) (b)

Fig. 4 Example of a local search for an operation o_1

Type 1: chromosomes which violate the data dependency. For example, in Fig 3(a), nodes N_2 and N_3 are scheduled before the execution of their predecessor N_1 is completed. As a result, Fig 3(a) does not represent a valid scheduling result for the problem.

Type 2: chromosomes which violate the time constraint.

Type 3: chromosomes which violate the area constraint.

For type 2 and 3, the possibility of producing a valid chromosome after merging with another valid or invalid chromosome, is relatively high. However, for type 1, the probability of producing a valid chromosomes after merging is extremely low. Our proposed algorithm eliminates all the invalid chromosomes of type 1 by re-scheduling the nodes. The re-scheduling algorithm is as follows.

Let us explain the re-scheduling algorithm using Fig. 3. In Fig. 3(a), nodes N_2 and N_3 violate the data dependency. According to step 1, ψ equals to the set of N_2 and N_3. Step 2 is skipped because the set ψ is not empty. In Step 3, the node N_2 is selected, because it has the minimum mobility. According to Step 4, N_2 is re-scheduled after its predecessor N_1 as shown in Fig. 3(b). Let us assume that the FU of N_2 has the shortest processing time. Therefore, the algorithm returns to Step 1, without replacing the FU of N_2. The algorithm continues until the set ψ becomes empty, *i.e.* all the data dependency problems are solved.

3.3 Selection

Even though, the above method effectively removes the invalid chromosomes of type 1, it may also increases the

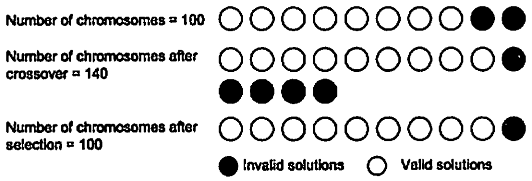

Number of chromosomes = 100
Number of chromosomes after crossover = 140
Number of chromosomes after selection = 100
Invalid solutions
Valid solutions

Fig. 5 Selection

chromosomes that violate the time constraint, after the rescheduling. Therefore we propose method 2 to remove the rest of the invalid chromosomes. Method 2 focuses on the selection process. According to the experimental results of various examples, the invalid chromosomes produced after the first generation were about 10% $\sim$ 40% of the total population. For example, let us assume that the population equals to 100 and the invalid chromosome percentage is x% after the initial generation. Now, we increase the number of genes generated after the crossover to less than $100/(100-x)$. After that, we select only 100 chromosomes from the pool of $100/(100-x)$. Note that, for the better performance, the number of chromosomes generated after the crossover should not be extremely larger than the population. If we produce huge amount of chromosomes and select only a small amount of high fit ones, then most of the selected chromosomes will be identical, so that the variation of the chromosomes will decrease resulting bad performance. In case of a extremely higher number of invalid chromosomes, we recommend to increase the population, in order to keep the variety of the chromosomes. Fig. 5 shows an example of this selection method.

3.4 Local search

A local search is applied to new chromosomes generated by crossover and mutation operators. All the individuals in the population obtained by the local search represent the local optima. After these individuals evaluated based on their energy consumption values, promising individuals are selected to form the next generation. The local search algorithm is shown as follows.

Step 1: Select one individual (I_i) from the population (P), where P is a set of individuals generated by crossover and mutation operators. $P = P - I_i$.

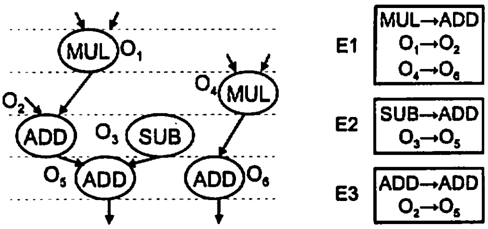
Step 2: Select one operation (o_i) from O_{I_i}, where O_{I_i} is a set of nodes in the individual (I_i). $O_{I_i} = O_{I_i} - o_i$.

Step 3: Search a feasible module selection for operation o_i to improve the solution, while the module selection for all the operations except o_i are fixed.

Step 4: if $O_{I_i} \neq \phi$ then go to Step2.

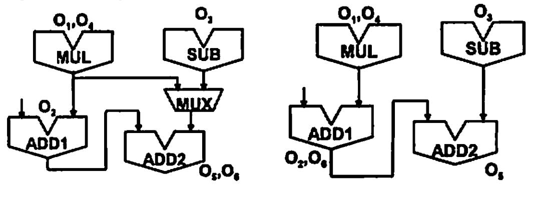
Step 5: if $P \neq \phi$ then go to Step1.

Since the module selection for every operation except operation o_i are fixed, local optima can be found in a reasonable time period. Suppose that an individual is shown in Fig 4(a), let us explain the local search for operation o_1. In this case, the module selection for all the operations except o_1, *i.e.* operations o_2, o_3, o_4 and o_5 are fixed. A feasible module selection for only operation o_1 is searched. The resulting individual obtained by the local search for operation o_1 is shown in Fig. 4(b), where $V'_{dd} < V_{dd}$. Therefore, the energy consumption is reduced, *i.e.* the solution is improved.



(a) DFG (b) E-templates

Fig. 6 E-templates in a single-supply-threshold voltage scheme



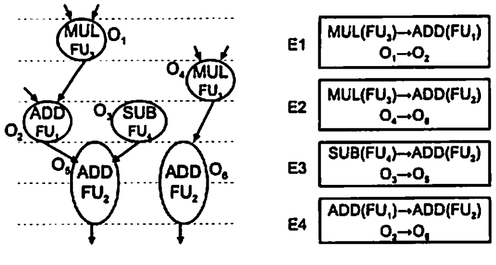
(a) Normal binding (b) E-template based binding

Fig. 7 Binding

3.5 Binding

We adopt the binding algorithm [4] since it efficiently reduces the interconnection complexity. An e-instance is a pair of nodes connected by an edge. E-instances are classified into types called e-templates, based on the operation types of their source and destination nodes. Fig. 6(b) shows the e-templates and e-instances that can be derived from the DFG in Fig. 6(a). E-instances in the same e-template can share the same functional units, if their operations are not overlapped. Figs 7(a) and 7(b) show the binding results based on the e-templates and without using the e-templates respectively. E-template based binding in Fig. 7(b) provides a simple interconnection network for the DFG. Nodes O_1 and O_2 share the same interconnection with O_4 and O_6. As a result, energy reduction in the interconnections can be achieved.

We extend the concept of e-templates to be applicable in the multiple supply and threshold voltage scheme. We define e-templates considering, not only the operation types but also the FU types specified in the module library (Table 1). Unlike the original e-templates, we use a DFG after scheduling and module selection as shown in Fig. 8(a). Note that in Fig. 8(b), e-instances with same operation type ($O_1 \rightarrow O_2$


MUL FU₃ O₁
O₄ MUL FU₃
ADD FU₁ O₂
O₃ SUB FU₄
O₅ ADD FU₂
ADD FU₂ O₆
E1 MUL(FU₃)→ADD(FU₁) O₁→O₂
E2 MUL(FU₃)→ADD(FU₂) O₄→O₆
E3 SUB(FU₄)→ADD(FU₂) O₃→O₅
E4 ADD(FU₁)→ADD(FU₂) O₂→O₅

(a) DFG (b) E-templates

Fig. 8 E-templates in a multi-supply-threshold voltage scheme

and $O_4 \rightarrow O_6$) are classified into different e-templates (E1 and E2) based on the FU type.

Our proposed method minimizes the functional unit energy consumption first and then performs binding to estimate the interconnection energy consumption. As shown in Fig. 2, binding is performed for all the chromosomes in the population after the local search. Then, FU energy and interconnection energy are calculated. We assume that the interconnection energy is proportional to *fan-outs* and *fan-ins* as mentioned in the section 2. All the chromosomes in the population are evaluated considering interconnection and functional unit energy consumptions. Then, the chromosome that has the minimum total energy is considered as the best solution. This gives a better result for the total energy consumption minimization problem.

The above process gives a near optimal solution for a particular α value. However the value α cannot be determined in high level synthesis, since it depends on the low level tasks such as placement and routing. Therefore, above high level synthesis process is performed for different α values. Then each solution will be subjected to the low level tasks such as placement and routing as shown in the Fig.9. After the low level tasks, each circuit is evaluated for the energy consumption and the one with the least energy consumption is chosen as the best design.

Fig. 10 shows the α value and the actual energy consumption obtained using spice simulation under 0.18μm CMOS design rules. According to the Fig. 10, we can say that there is an optimal value for α which gives the circuit design with the minimum energy consumption.

4. Evaluation

The evaluation is based on 0.18μm CMOS design process. We use several benchmark examples such as EW filter, FIR filter, etc. Evaluation was done for different α values and best solution is chosen according to the circuit simulation results. We compare our approach with conventional method that use a single supply and threashold voltage and only optimize FU energy. The comparisons with the conventional method gives up to 50% of energy savings in some of the benchmark examples and about 30% energy savings in average.

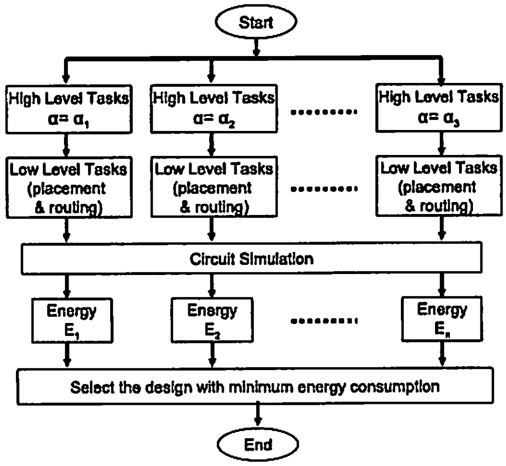

Start
High Level Tasks α= α₁
High Level Tasks α= α₂
High Level Tasks α= α₃
Low Level Tasks (placement & routing)
Low Level Tasks (placement & routing)
Low Level Tasks (placement & routing)
Circuit Simulation
Energy E₁
Energy E₂
Energy Eₙ
Select the design with minimum energy consumption
End

Fig. 9 Flowchart of the overall process

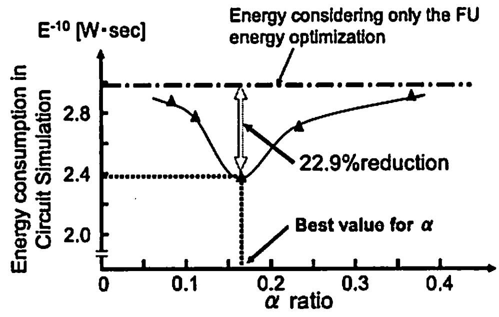


Fig. 10 Energy consumption vs α value

5. Conclusion

We have presented a method that considers functional unit and interconnection energy consumption together. It gives better results when the α is high. Therefore, higher percentage of energy savings can be achieved for the designs with complex interconnection networks.

We used a GA for the energy consumption minimization problem. Therefore, our approach can also be used in large size DFGs and reasonable solution can be found in a shorter time.

Acknowledgement This work is supported by VLSI Design and Education Center(VDEC), the University of Tokyo in collaboration with Synopsys, Inc.

References

[1] Masanori Hariyama, Tetsuya Aoyama and Michitaka Kameyama, "Genetic Approach to Minimizing Energy Con-

sumption of VLSI Processors Using Multiple Supply Voltages", IEEE Transactions on Computers, Vol. 54, No. 6, pp. 642-650, June 2005.

[2] Noureddine Chabini and Wayne Wolf, "Reducing Dynamic Power Consumption in Synchronous Sequential Digital Designs Using Retiming and Supply Voltage Scaling", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 12, No. 6, pp. 573-589, June 2004.

[3] L. Wei, Z. Chen, M. C. Johnson, K. Roy, and V. De, "Design and optimization of low voltage high performance dual threshold CMOS circuits", Design Automation Conference Proceedings pp. 489-494, June 1998.

[4] Renu Mehra and Jan Rabaey, "Exploiting Regularity for Low-Power Design", Proceedings of the International Conference on Computer-Added Design, 1996.