IPSJ Transactions on System LSI Design Methodology Vol. 3 257–267 (Aug. 2010)

Regular Paper

# A Resource Binding Method to Reduce Data **Communication Power Dissipation on LSI**

# HIDEKAZU SETO<sup> $\dagger 1$ </sup> and KAZUHITO ITO<sup> $\dagger 1$ </sup>

The energy dissipation by data communications on a LSI chip depends on the layout of modules as well as how data are communicated among modules. The requirement of data communications are determined by the schedule of computations and by the resource binding of computations to functional units and data to registers. In this paper, a method of resource binding is proposed to derive a binding which is able to obtain the floorplan with reduced energy dissipation by data communications.

#### 1. Introduction

Data communication over interconnects between modules such as functional units and registers is one of the major sources of energy dissipation on LSIs<sup>1</sup>). Transferring data for communications is achieved by charging or discharging the corresponding interconnect wire in CMOS technology. Therefore the energy dissipation strongly depends on the capacitance of the wire. It is reported that for a moderate wire length, the energy dissipation is almost proportional to the wire length<sup>2)</sup>. The length of wire necessary for a particular data communication can be minimized by using point-to-point interconnection scheme or by using bus architecture with bus splitting  $^{3)-5)}$ . In order to fully utilize these techniques to minimize the energy dissipation by data communication on interconnects, how much data are communicated as well as how long these data are transfered between sources and destinations have the great importance. The resource binding in high-level design determines how data are communicated on interconnects. The floorplanning determines the physical locations of modules and therefore the length of each data communication.



Fig. 1 Design methods. (a) Ref. 6). (b) Ref. 7). (c) Ref. 8) and proposed.

There have been some design methods to reduce the energy dissipation by data communications  $^{6)-8)}$ . In Ref. 6) (Fig. 1 (a)), scheduling, binding, and floorplanning are simultaneously considered to minimize the interconnect power dissipation. Given a combination of scheduling and binding, a floorplan is derived and the interconnect power dissipation is computed. By using the simulated annealing (SA), the combination of scheduling and binding is explored which leads to the minimum interconnect power dissipation. In Ref. 7) (Fig. 1 (b)), the optimum binding which leads to the minimized energy dissipation is searched as follows: starting from the initial binding, (i) change the current binding, (ii) reschedule operations if necessary, (iii) estimate the interconnect lengths and compute the power dissipation, (iv) accept the change if the power is reduced, and (v) repeat (i)-(iv) with the current power dissipation information fed back to the binding changed in (i).

In Ref. 8) (Fig. 1 (c)), the scheduling is explored by SA which leads to a desirable binding where data communications are concentrated onto a small number of pairs of source and destination modules. Then the pairs of modules with many data communications are placed adjacently to each other to minimize the wire length in floorplanning, which is explored by using the combination of SA and the sequence-pair<sup>9</sup>). In this method, however, a particular module may be paired with more than one modules. In that case, some of these module pairs cannot placed adjacently and thus the reduction of energy dissipation is limited.

In this paper, a binding method is proposed to be used in the same framework as Ref. 8), which derives the binding where a module is paired with at most two

<sup>&</sup>lt;sup>†1</sup> Saitama University

#### 258 A Resource Binding Method to Reduce Data Communication Power Dissipation on LSI

other modules with many data communications. Then the information about the module pairs are used as the constraints in the floorplanning so that the paired modules are placed adjacently to minimize the wire length between them.

This paper is organized as follows. The hardware model is defined in Section 2. In Section 3, the proposed binding method is described in detail. Experimental results are shown in Section 4. Section 5 concludes this paper.

## 2. Hardware and Energy Dissipation Models

# 2.1 Functional Unit and Register

Functional units (FU), such as adders and multipliers, are assumed to have no input and output registers. A pipelined FU contains internal register(s) for its intermediate results of pipeline stages except for the last stage. To store the computation result, registers outside the FU is used.

Let a module refer to either an FU or a register. The shape of module type t is rectangle with width  $w_t$  and height  $h_t$ . The input ports of a module are located on the middle of a wider edge of the rectangle and the output port is located on the edge opposite to the input ports. This is illustrated in **Fig. 2** (a). If a module receives data from several sources, one or more multiplexors (MUXs) are used and placed on the input side of the module as shown in Fig. 2 (b).

## 2.2 Module Interconnection and its Energy Dissipation

Modules are connected with each other to transfer data from one module to another. Data are transferred from FU to register, from register to FU, from FU to FU for chaining<sup>10)</sup>, and from register to register. The last transfer is used when a data needs to be stored for the duration longer than the iteration period.

The interconnect energy dissipation is proportional to the wire length from the source to the destination<sup>2</sup>). The wire length can be computed as the shortest



Fig. 2 The model of a module. (a) Size and input/output ports. (b) A module with multiplexor(s) on input port.

path between the source and the destination, with the constraint that the path does not go over any module. The source is an output port of a module and the destination is an input port of another module. The energy dissipation by data communications, E, is given by the following equation

$$E = \sum_{d \in D} \left\{ K_d \times WL_d \times M_d \right\},\tag{1}$$

where D is the set of all the data communication source and destination pairs,  $WL_d$  is the wire length between the source and the destination of data communication d,  $M_d$  is how many times data are communicated for d, and  $K_d$  is a constant. For simplicity, we assume  $K_d$  is common to all the wires. Minimizing the energy dissipation by data communications is achieved by minimizing the energy dissipation parameter EC defined as follows.

$$EC = \sum_{d \in D} \left\{ WL_d \times M_d \right\}$$
<sup>(2)</sup>

## 3. The Proposed Binding Method

Minimization of energy dissipation by data communications can be achieved by minimizing wire lengths  $WL_d$  as can be seen from Eq. (2). The wire length from one module to another becomes shortest if these two modules are placed so that the output port of the former is next to the input port of the latter. However, it is not possible to achieve such placements for every module pairs. Therefore the binding is done so that data communications are concentrated to a small number of module pairs and little data communications are done between other module pairs. To obtain such binding, the idea of communication concentrated module group (CCMG) is introduced here.

# 3.1 Definition of CCMG

A CCMG is an ordered set of two or more modules (adders, multipliers, registers, etc.). Let a CCMG of k modules be denoted as  $\langle m_1, m_2, \ldots, m_k \rangle$ . The modules  $m_1$  and  $m_k$  are respectively the *head* and *tail* of the CCMG. The modules of a CCMG are ordered so that the number of data communications from  $m_i$  to  $m_{i+1}$  is sufficiently large. In other words, a set of k modules, each of which outputs many data to the next module, forms a CCMG as illustrated in **Fig. 3**. The CCMG(s) is used as the placement constraint in the

259 A Resource Binding Method to Reduce Data Communication Power Dissipation on LSI



floorplanning so that the modules in a CCMG are placed adjacently as ordered by the CCMG. For example, **Fig. 4** shows the placement of modules of a CCMG  $\langle \text{Reg3}, \text{Mul2}, \text{Reg4}, \text{Add2} \rangle$ . Reg3 and Mul2 are placed so that the output of Reg3 is next to the input of Mul2. Therefore, the data communications from Reg3 to Mul2 are done with very short wire length and thus dissipate very small energy.

The aim of introducing CCMGs is to constrain the placement of modules so that modules within a CCMG are placed adjacently to minimize the wire length between the modules. However, as shown in **Fig. 5**, large CCMGs sometimes force the wire lengths to become long for the modules in a CCMG but not adjacent to each other, the modules of different CCMGs, and modules not included in any CCMG. Thus it is important to maintain CCMGs in adequate size both in the number of modules and in physical dimensions. The following three conditions are imposed in constructing a CCMG.

- (1) the number of modules is less than or equal to  $K_n$
- (2) the sum of module heights is less than or equal to  $K_l$
- (3) the number of data communications between modules is larger than or equal to  $K_c$

Theses parameters are determined based on the application (DFG) and the schedule of the computations. An example of how to determine these parameters is given in Section 4.

## 3.2 Output Efficiency of Data

In binding process, computations are bound to FUs. Then computations are divided into two groups: one is already bound to some FUs and the other is not yet bound to any FU.  $NUB_t$  denotes the set of not yet bound computations of type t. The computation type t is addition, multiplication, and so on. Similarly, data are divided into already bound and not yet bound to registers. DUB denotes the set of not yet bound data.

Let d denote a data.  $NUB_t(d)$  is defined as the set of computations in  $NUB_t$ which use d as the input data. The parameter DR(d, m) is defined as the number of times the data d is communicated to module m. The life time LT(d) of d is the difference in clock cycles between the time when d is generated and the time when d is last referenced. Two kinds of output efficiency of d,  $OE_t(d)$  and DE(d, m), are defined as

$$OE_t(d) = \frac{|NUB_t(d)|}{LT(d)},\tag{3}$$

$$DE(d,m) = \frac{DR(d,m)}{LT(d)}.$$
(4)

 $OE_t(d)$  indicates how frequently d is used by unbound computations of type t and DE(d, m) indicates how frequently d is communicated to module m.

# 3.3 Extension of CCMG

Starting from a seed of CCMG, the CCMG is extended by adding modules one by one with one of the extensions described in this section. These extensions are illustrated in **Fig. 6**. The seed of CCMG is a pair of modules. A CCMG is extended as long as the CCMG condition is satisfied.

# 3.3.1 Forward FU Extension

An FU, FU<sub>N</sub>, is appended after the CCMG *Tail* (Fig. 6 (a)). For each computation type t, the set NS(t) is constructed to include as many computations in  $NUB_t$  as possible with respect to the conditions that these computations refer data from *Tail* and the execution time of these computations do not overlap with each other. The set NS(t) is constructed as follows. First construct a set of the

260 A Resource Binding Method to Reduce Data Communication Power Dissipation on LSI



extension. (c) Backward FU extension. (d) Backward register extension.

computations which are in  $NUB_t$  and use data sent from *Tail*. Then one computation for each clock cycle is selected from the set. If more than one computations are executed in the same clock cycle (because of either parallel execution or overlap of iterations), one is selected so that (1) the data output by the computation is not yet bound to any register, or is not sent to a computation already bound to an FU, and (2) the life time of the data is shortest.

Figure 7 shows an example of constructing NS(t). The life times of data  $d_i$  output by the computation i are shown in Fig. 7 (b). It is assumed that the data  $d_4$  is already bound to a register. The computations 1 and 6 are selected because these are only the computations in the clock cycles. The computations 2 and 3 are executed in the same clock cycle but computation 3 is selected because  $LT(d_3)$  is shorter than  $LT(d_2)$ . Among the computations 4 and 5, the computation 5 is selected because  $d_4$  is already bound but  $d_5$  is not yet bound. The result of computation selection is as shown in Fig. 7 (c).

Among all the possible computation types, the computation type t with the largest |NS(t)| is chosen. FU<sub>N</sub> of type t is appended to the CCMG and the computations in the selected NS(t) is bound to FU<sub>N</sub>.

#### 3.3.2 Forward Register Extension

A register  $\operatorname{Reg}_N$  is appended after the CCMG *Tail* (Fig. 6 (b)). The set of data  $DS \subseteq DUB$  to be bound to  $\operatorname{Reg}_N$  is selected so that not only the number of data communications from *Tail* to  $\operatorname{Reg}_N$  is maximized but the number of data communications from  $\operatorname{Reg}_N$  to an FU, which will be appended to the CCMG in the future, would become as large as possible.

The parameter  $\tau$  denotes the computation type of such a virtual FU. For each computation type  $\tau$ , the output efficiency  $OE_{\tau}(d)$  is calculated for each data d





Fig. 7 An example of forward FU extension. (a) Computations in  $NUB_t$  and using data from *Tail*. (b) The life times of data output by the computations in (a). (c) The selected computations.

Fig. 8 An example of forward register extension. (a) The data in DUB. (b) The selected data to be bound to  $\text{Reg}_N$ .

in DUB. Then, as many data in DUB as possible are selected in descending order of  $OE_{\tau}(d)$  so that the life time of the data do not overlap with each other. **Figure 8** shows an example. There are 6 data in DUB as shown in Fig. 8 (a).  $OE_{\tau}(d)$  of each data d is also shown in the figure in parenthesis near the data d. The selected data are shown in Fig. 8 (b). In this example, data  $d_6$  instead of data  $d_3$  is selected because  $OE_{\tau}(d_6) = 2 > OE_{\tau}(d_3) = 1$ . In addition,  $d_4$  instead of data  $d_5$  is selected because  $OE_{\tau}(d_4)$  and  $OE_{\tau}(d_5)$  are the same but the life time of  $d_4$  is longer than  $d_5$ .

Among all the possible computation types, the one with the most data communication from Tail to DS is selected. Then the data in DS is bound to  $\text{Reg}_N$ .

# 3.3.3 Backward FU Extension

In the backward FU extension, an FU,  $FU_N$ , is inserted before the *head* of the CCMG (Fig. 6 (c)). For each computation type t, the set  $NS(t) \subseteq NUB_t$  of computations whose output data are sent to *Head* is constructed. Among all the possible computation types, one with the largest |NS(t)| is chosen and the computations are bound to  $FU_N$ .

### 3.3.4 Backward Register Extension

A register  $\operatorname{Reg}_N$  is inserted before the *head* of the CCMG (Fig. 6 (d)). The

261 A Resource Binding Method to Reduce Data Communication Power Dissipation on LSI





Fig. 9 An example of rebinding in backward register extension. (a) The data in DUB and communicated to Head. (b) The selected data to be bound to  $\operatorname{Reg}_N$ .

Fig. 10 A seed of CCMG. (a) FR type. (b) RF type.

set of data  $DS \subseteq DUB$  is selected so that the sum of data communications  $\sum_{d \in DS} DR(d, Head)$  is maximized and then the sum of output efficiencies  $\sum_{d \in DS} DE(d, Head)$  is maximized. DS is constructed greedily by including  $d \in DUB$  into DS in descending order of DR(d, Head) and DE(d, Head) so that the life times do not overlap with each other. In an example shown in **Fig. 9** (a), 5 data are in DUB. DR(d, Head) of each data d is shown in parenthesis near each data. In this example, data  $d_3$ ,  $d_4$ , and  $d_5$  are selected as shown in Fig. 9 (b). The selected data in DS are bound to  $\text{Reg}_N$ .

# 3.4 Creation of a Seed of CCMG

A seed of CCMG is a pair of an FU and a register. A seed of CCMG is either FR type or RF type as shown in **Fig. 10**.

An FR type seed is created in the similar way as the forward register extension, except that *Tail* has no bound computation. For each possible combination of computation types t and  $\tau$ , among the output data of computations in  $NUB_t$ , the set  $DS(t,\tau)$  of data is selected so that  $|DS(t,\tau)|$  is the largest and then the sum of output efficiencies  $\sum_{d \in DS(t,\tau)} OE_{\tau}(d)$  is the largest. The  $\tau$  is the computation type of FU which would be appended after the register in the future CCMG extension. Among the possible combinations of t and  $\tau$ , one which achieves the largest  $|DS(t,\tau)|$  and secondarily the largest  $\sum_{d \in DS(t,\tau)} OE_{\tau}(d)$  is chosen. These data in  $DS(t,\tau)$  are bound to the register of the seed. Next, the computations of type t are selected in just the same way as the backward FU extension and bound to the FU of the seed.

An RF type seed is created similarly to the backward register extension except that *Head* has no bound computation. By assuming that *Head* is an FU of computation type t, the set DS(t) of data is selected so that the sum of data communications,  $\sum_{d \in DS(t)} |NUB_t(d)|$ , is maximized. DS(t) is constructed by including  $d \in DUB$  into DS(t) in descending order of  $NUB_t(d)$  so that the life times do not overlap with each other. The t which achieves the largest  $\sum_{d \in DS(t)} |NUB_t(d)|$  is identified and the data in DS(t) are bound to the register of the seed. Next, the computations of type t are selected in just the same way as the forward FU extension and bound to the FU of the seed.

# 3.5 Bindingc Algorithm with CCMG

The algorithm to bind computations to FUs and data to registers is proposed here so that desirable CCMGs can be derived from the binding result.

Binding algorithm with CCMG

Input: DFG = (N, E) (N is the set of computations,

E is the set of edges between computations) the schedule of computations in N with the iteration clock cycle Tr CCMG condition

Output: the binding of computations to FUs, data to registers the constructed CCMGs

Objective: construct as many CCMG as possible

(1) (Construction of CCMGs)

Create a seed of CCMG with the most data communications from an FU to a register (FR type) or from a register to an FU (RF type).If no seed satisfies the CCMG condition, go to step 4.

- (2) Bind computations and data so that the current CCMG is extended by one module. If the current CCMG cannot be extended without violating the CCMG condition, go to step 1.
- (3) Repeat step 2.
- (4) (Binding of remaining computations and data)
   Execute the conventional binding<sup>8)</sup> for the not yet bound computations and data.

262 A Resource Binding Method to Reduce Data Communication Power Dissipation on LSI



Figure 11 shows the statistics of an example binding result. The figure shows how many times data are communicated for each pair of modules for the DFG of WEF3 and the iteration period of 54. The parameters of CCMG condition are  $K_n = 4$ ,  $K_l = 36.7$ , and  $K_c = 13$ . The DFG and the parameters are explained in Section 4. There are 14 modules and  $14 \cdot 13 = 182$  module pairs. Two CCMGs are constructed in this case. Only the seven module pairs in CCMGs require more than 10 data communications and about 49% of 286 data communications are done on these module pairs. This result shows the effectiveness of the proposed binding with CCMG.

The computational complexity of the algorithm is calculated as follows. The computational complexities of the CCMG extensions are O(FNTr) for forward and backward FU extension,  $O(FN \log N)$  for forward register extension, and  $O(N \log N)$  for backward register extension, where F is the number of computation types. The computational complexities of creating a CCMG seed are  $O(F^2N \log N + FNTr)$  for FR type and  $O(N \log N) + O(FNTr)$  for RF type. The creation of CCMG seeds and CCMG extensions are repeated at most N/2 times. Therefore, the CCMG construction requires  $O(F^2N^2 \log N + FN^2Tr)$  computations. The remaining computations and data are bound by the conventional method<sup>8</sup> with the computational complexity of O(P!NTr) where P is the number of FUs.

### 3.6 Evaluation of the Binding Result

For the obtained binding of computations and data, the following value  $S_C$  is calculated as the evaluated cost of the binding.

$$S_{C} = \sum_{\langle m,n \rangle \in \text{CCMGMP}} C(m,n) - \varepsilon \left\{ \sum_{m} \left( \sum_{n} L_{d}(n,m) \right)^{2} + \sum_{m} \left( \sum_{n} L_{d}(m,n) \right)^{2} \right\} - \beta |\text{SM}| + W$$
(5)

Here CCMGMP is the set of ordered pairs of modules in CCMGs. C(m,n) is the number of data communications from module m to module n. Hence the first term of  $S_C$  is the total number of data communications within CCMGs. The parameter  $L_d(m,n) = 1$  if there exist data communication(s) from module m to module n, or  $L_d(m,n) = 0$  otherwise.  $\sum_n L_d(n,m)$  and  $\sum_n L_d(m,n)$  are respectively the number of fanins and the number of fanouts of m.  $\varepsilon$  is a positive weighting factor. SM is the set of modules,  $\beta$  is a positive factor, and W is a sufficiently large constant to make  $S_C$  positive. When  $\beta = 0$ , only the energy dissipation by data communications is considered. When  $\beta > 0$ , the number (in some sense the total area) of modules is also taken into account.

The evaluated cost  $S_C$  becomes large for the desired binding where many data communications are done between modules within CCMGs (the first term of  $S_C$ is large) and data communications are not spread over variety of module pairs (fanouts and fanins are small and therefore the second term of  $S_C$  is small). Hence  $S_C$  is used as the cost function in the SA for optimization of schedule and its binding in the schedule exploration of the design method in Ref. 8).

#### 4. Experimental Results

The proposed binding method is implemented by using programming language C++ and it replaces the binding procedure in the design system proposed in Ref. 8). The scheduling exploration is done to maximize the binding cost  $S_C$ . The same floorplanning as in Ref. 8) is used except that the modules in CCMGs are restricted to be placed adjacently with each other as described in Section 3.1. The software is run on a PC with a 2.66 GHz processor and 2 GB memory.

The modules are designed for TSMC 0.18  $\mu$ m CMOS technology. The module width  $w_M = 28$  and height  $h_M = 20$  for multipliers,  $w_A = 28$  and  $h_A = 3$  for adders, and  $w_R = 28$  and  $h_R = 2$  for registers. The width of multiplexor (MUX)





Fig. 12 Comparison of interconnect energy dissipation savings against the area-optimized results.

is  $w_{MUX} = 28$ . The height of MUX depends on the input count and assumed as  $h_{MUX} = 1$  for 1 to 4 inputs,  $h_{MUX} = 2$  for 5 to 7 inputs,  $h_{MUX} = 3$  for 8 to 10 inputs, and so on. All of these length are noted in unit of length. The multiplier is two stage pipelined. Therefore, a new multiplication can be started one clock cycle after the previous multiplication was started. Although the proposed binding supports chaining of computations  $^{10)}$ , the chaining is not used in these experiments since the multiplier is pipelined to shorten the clock cycle. Energy dissipation of FUs, registers, and data communications are evaluated by circuit simulations with HSPICE.

The parameters of CCMG conditions were chosen as follows. The  $K_n$  is set as 4.  $K_l$  is the square root of the total area of the least modules required by the given schedule. Let NC denote the total number of data communications determined based on the given schedule.  $K_c$  is NC/20 if  $NC \leq 200$  or NC/10otherwise. If 60% of the iteration period (Tr) is smaller than  $K_c$ , then it is used as  $K_c$ . The number of data communications from one module to another is limited by the iteration period. Therefore if NC is large and Tr is small. no module pair could be included in CCMG. To avoid it, the last condition is



Tr=8 = 11 = 14

Tr=8 =11 =14

Diffea

results against the area-optimized results.

Tr=8 =9

1DDCT

Fig. 13 Comparison of interconnect energy dissipation savings by the energy optimized

60%

40

Energy dissipation s

0

saving

Diffea 1DDCT WEF (Elliptic) WEF3 Average Fig. 14 Comparison of interconnect energy dissipation savings by the energy and area optimized results against the area-optimized results.

=10

Tr=8 = 9

introduced in determining  $K_c$ . The factor  $\varepsilon$  in Eq. (5) is either  $NC^2/1,000,000$ or 0.01, whichever is larger.

First, the savings of energy dissipation by data communications were compared among the conventional (Refs. 7) and 8)) and the proposed methods. The same simulated annealing parameters were used for the method in Ref. 8) and the proposed method. Because the different technology is assumed and different

Reference 8)

Proposed

=10 Tr=16 =17 =18 Tr=48 =51 =54

Tr=16 =17 =18 Tr=48 =51 =54

WEF3

Average

WEF (Elliptic)

264 A Resource Binding Method to Reduce Data Communication Power Dissipation on LSI

		AO		PO (Ref. 8))		PO (Proposed)			WA (Ref. 8))			WA (Proposed)			WA2 (Proposed)			
DFG	Tr	Power	$\mathbf{SM}$	Power	Area	SM	Power	Area	SM	Power	Area	SM	Power	Area	SM	Power	Area	$\mathbf{SM}$
Diffeq	8	44.1	$1,\!1,\!5$	42.6	188.1%	$2,\!1,\!5$	42.0	328.3%	2,2,5	42.7	107.2%	$1,\!1,\!5$	42.9	107.2%	1,1,5	42.9	107.2%	1,1,5
				96.5%			95.2%			96.7%			97.2%			97.2%		
	11	32.2	$1,\!1,\!5$	31.1	200.8%	2,1,5	30.2	421.8%	$^{3,3,6}$	31.2	131.2%	$1,\!1,\!5$	31.0	105.4%	$1,\!1,\!5$	31.2	107.1%	1,1,5
				96.6%			93.7%			96.9%			96.2%			96.9%		
	14	25.5	$1,\!1,\!5$	24.6	112.6%	$1,\!1,\!5$	23.7	408.8%	$^{3,3,6}$	24.5	114.0%	$1,\!1,\!5$	24.4	102.2%	$1,\!1,\!5$	24.5	103.8%	1,1,5
				96.1%			92.9%			96.1%			95.4%			96.0%		
1DDCT	8	110	$^{4,2,9}$	98.0	236.5%	$6,\!4,\!11$	96.7	216.6%	$6,\!4,\!11$	99.4	136.9%	$^{4,2,9}$	98.3	135.7%	4,2,9	98.2	125.6%	4,2,9
				89.3%			88.2%			90.7%			89.6%			89.6%		
	9	97.3	4,2,9	87.2	224.6%	5,3,11	85.8	163.1%	4,2,11	87.4	151.1%	4,2,9	85.9	144.2%	4,2,9	85.9	113.2%	4,2,9
				89.6%			88.2%			89.8%			88.3%			88.4%		
	10	89.1	3,2,9	78.0	184.1%	4,2,12	76.9	197.6%	4,2,12	78.7	158.5%	4,2,9	78.2	124.6%	3,2,9	77.6	123.6%	3,2,9
				89.9%			87.8%			89.9%			89.4%			88.6%		
WEF	16	41.9	$3,\!1,\!10$	39.7	202.1%	3,2,10	36.9	155.4%	3,1,11	38.2	222.0%	3,2,9	37.0	176.7%	3,1,11	37.0	120.2%	3,1,11
				90.4%			88.0%			91.1%			88.2%			88.2%		
	17	39.2	$2,\!1,\!9$	35.9	226.0%	2,2,9	34.6	213.3%	4,1,11	35.9	239.6%	$2,\!1,\!9$	34.8	215.8%	3,1,10	34.6	140.6%	3,1,10
				91.5%			88.3%			91.5%			88.8%			88.3%		
	18	37.0	$2,\!1,\!9$	33.7	273.6%	$^{3,2,9}$	32.5	221.8%	4,1,11	33.9	178.5%	$2,\!1,\!9$	32.7	155.2%	$3,\!1,\!10$	32.8	147.3%	3,1,10
				91.1%			87.8%			91.5%			88.3%			88.4%		
WEF3	48	42.1	$3,\!1,\!10$	38.8	235.8%	4,2,10	37.1	175.6%	3,1,11	38.8	177.0%	3,1,10	37.1	169.5%	3,1,10	37.2	112.1%	3,1,10
				92.2%			88.1%			92.0%			88.1%			88.2%		
	51	39.4	$3,\!1,\!9$	36.2	210.1%	3,2,10	34.9	165.6%	3,1,11	36.7	217.1%	3,2,9	35.0	168.0%	3,1,10	35.0	122.6%	3,1,10
				91.9%			88.5%			93.1%			88.8%			88.8%		
	54	37.2	$2,\!1,\!9$	34.3	235.3%	3,2,9	32.9	150.1%	$3,\!1,\!10$	34.4	237.1%	2,2,9	33.0	182.1%	3,1,10	33.0	130.0%	$3,\!1,\!10$
				92.2%			88.4%			92.5%			88.6%			88.7%		
Average				92.2%	211.6%		89.6%	234.8%	-	92.7%	172.5%		90.6%	148.9%		90.6%	121.1%	

Table 1Comparison of Designs.

modules (FUs and registers) are used, directly comparing the energy dissipation of the methods is difficult. Hence, how much energy dissipation is reduced is compared. Two benchmark DFGs are used. The differential equation (Diffeq) consists of 6 multiplications and 5 additions. The 5th order wave elliptic filter (WEF)<sup>11)</sup> (referred as Elliptic in Ref. 7)) consists of 8 multiplications and 26 additions. Three iteration clock cycles (Tr) are tried for each DFG. Figure 12 shows the interconnect energy dissipation saving of the energy optimized design against the area optimized design. The area optimized designs were obtained by minimizing the number of modules in scheduling and binding, and then floorplanning with the objective to minimize the area of the rectangle bounding all the modules<sup>8)</sup>. Figure 12 shows that the proposed method can save about 15 points and 10 points more energy dissipation against the methods in Refs. 7) and 8), respectively.

Next, the binding scores of the proposed method were compared. Two more DFGs, 8-point 1D DCT<sup>12)</sup> and WEF3, are used in addition to Diffeq and WEF. 1DDCT consists of 11 multiplications and 29 additions. WEF3 is derived by

Bounding rectangle

265 A Resource Binding Method to Reduce Data Communication Power Dissipation on LSI

	exploration.							
-	DFG	Tr	Ref. 8)	Proposed				
-	Diffeq	8	11.1	11.3				
	(PO)	11	12.2	13.1				
		14	12.9	15.4				
	1DDCT	8	2898.7	48.8				
	(PO)	9	827.1	61.4				
_		10	638.8	67.8				
	WEF	16	74.1	41.9				
	(PO)	17	76.7	46.1				
		18	80.6	49.1				
	WEF3	48	516.0	275.7				
	(PO)	51	650.1	358.0				
		54	733.4	436.6				

 Table 2
 Comparison of CPU times for schedule

Fig. 15 The floorplan result of WA for WEF, Tr = 16. CCMGs are indicated by thick broken rectangles. Gray regions are the dead space in the bounding rectangle.

unfolding<sup>13)</sup> WEF by factor 3. WEF3 consists of 24 multiplications and 78 additions. Again three iteration clock cycles (Tr) are tried for each DFG.

Figure 13 shows the savings of the energy dissipation by data communications where only the minimization of the energy dissipation is considered by using  $\beta = 0$ . Figure 14 shows the savings of the energy dissipation by data communications where the minimization of the energy dissipation as well as the number of modules is considered by using  $\beta = 100$ . In both cases, the proposed method saves the energy about 10 points more on the average than the conventional method.

**Table 1** summarizes the experimental results. *Tr* is the iteration clock cycles. 'AO' indicates the result where only the area is minimized. 'PO' indicates the result where the binding is done to minimize only the energy dissipation by data communications ( $\beta = 0$ ) and the objective of the floorplanning is to minimize *EC*  in Eq. (2). 'WA' indicates the result where the energy dissipation as well as the area is minimized in binding ( $\beta = 100$ ) but the same floorplanning. The power dissipation including the computations and data communications is shown in  $\mu$ W by assuming the designed circuit operating at a clock frequency of 100 MHz. The area is measured for the rectangle bounding all the modules relative to the AO design. SM is the content of the modules, where the numbers of adders, multipliers, and registers are shown from left to right. In some cases WA gives larger

 Table 3
 Constructed CCMGs.

-	DFG	Tr	CCMGs
	Diffeq	8	$\langle M1, R2, A0, R0 \rangle \langle M0, R1 \rangle \langle R3, A1 \rangle$
	(PO)	11	$\langle M0, R0, M2, R3 \rangle \langle A0, R1, M1, R2 \rangle \langle R4, A1 \rangle$
		14	$\langle M0, R0, M2, R3 \rangle \langle A0, R1, M1, R2 \rangle \langle R4, A1 \rangle$
	1DDCT	8	$\langle R2, A0, R0, A1 \rangle \langle R1, M0 \rangle \langle R4, A2, R3, A3 \rangle$
	(PO)		$\langle R4,M1 \rangle \langle R6,A4 \rangle$
		9	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle \langle R4, A3 \rangle$
		10	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle \langle R4, A3 \rangle$
	WEF	16	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle$
	(PO)	17	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle \langle R4, A3 \rangle$
		18	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle \langle R4, A3 \rangle$
	WEF3	48	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle$
	(PO)	51	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle$
		54	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle$
	Diffeq	8	$\langle R2, M0, R1, A0 \rangle$
	(WA)	11	$\langle R4,M0,R1 \rangle \langle R2,A0,R3 \rangle$
		14	$\langle R4,M0,R1 \rangle \langle R2,A0,R3 \rangle$
	1DDCT	8	$\langle R2, A0, R0, A1 \rangle \langle R1, M0 \rangle \langle R3, A2, R4 \rangle$
	(WA)	9	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle \langle R4, A3 \rangle$
		10	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle$
	WEF	16	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle$
	(WA)	17	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle$
		18	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle$
	WEF3	48	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle$
	(WA)	51	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle$
		54	$\langle R2, A0, R0, A1 \rangle \langle R3, M0, R1, A2 \rangle$

area than PO. The reason is that the floorplanning is done without minimizing the area and hence relatively large dead space exists as shown in **Fig. 15**. Thus another experiment was done where the floorplanning is done to minimize the weighted sum of data communication energy dissipation and the area of the bounding rectangle. The weighting factor of the area is 0.05 against the energy dissipation. The results are also shown in Table 1 in the column 'WA2'. The smaller area is obtained by WA2 than PO in every case at the cost of less energy dissipation reduction.

On average, 'PO' achieves about 10% reduction of power dissipation at the

266 A Resource Binding Method to Reduce Data Communication Power Dissipation on LSI

expense of 135% increase in area and 'WA2' achieves about 9% reduction of power dissipation at the expense of 21% increase in area. The proposed method saves about 2% more energy than the conventional method. By using modules designed for lower power, reduction of power dissipation by data communication may give larger impact on reduction of total power.

Table 2 compares the CPU times for the schedule exploration. Because the proposed binding method is faster than that used in Ref. 8), the speed up is achieved especially when the design includes many modules. The table shows only for 'PO', but the similar results are obtained for 'WA'.

Finally, **Table 3** shows the CCMGs constructed in the binding of the schedule explored for minimized data communication energy dissipation. In this table, A, M, and R are the abbreviations of adder, multiplier, and register, respectively.

The experimental results show that the proposed binding method achieves the smaller energy dissipation by data communications with less area overhead than the conventional method. In addition, the CPU time of the method is improved.

# 5. Conclusions

In this paper, a resource binding method for minimization of energy dissipation by data communications on LSI is proposed. The method is combined in the existing design framework<sup>8)</sup>. The experimental results show the effectiveness of the proposed method.

The investigation for the systematic method to determine the parameters, especially for CCMG conditions, remains as future work.

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

## References

- Komatsu, S., Ikeda, M. and Asada, K.: Low Power Microprocessors for Comparative Study on Bus Architecture and Multiplexer Architecture, *Proc. ASP-DAC '98*, pp.323–324 (1998).
- 2) Taylor, N., Dey, S. and Zhao, Y.: Modeling and Minimization of Interconnect Energy Dissipation in Nanometer Technologies, *Proc. DAC 2001*, pp.754–757 (2001).
- 3) Raghunathan, V., Srivastava, M.B. and Gupta, R.K.: A Survey of Techniques for

Energy Efficient On-Chip Communication, Proc. DAC 2003, pp.900–905 (2003).

- Hsieh, C.T. and Pedram, M.: Architecural Energy Optimization by Bus Splitting, IEEE Trans. Comput.-Aided Des. Intergrated Circuits & Syst., Vol.21, pp.408–414 (2002).
- 5) Lu, R. and Koh, C.-K.: A High Performance Bus Communication Architecture through Bus Splitting, *Proc. ASP-DAC 2004*, pp.751–755 (2004).
- 6) Prabhakaran, P., Banerjee, P., Crenshaw, J. and Sarrafzadeh, M.: Simultaneous Scheduling, Binding and Floorplanning for Interconnect Power Optimization, 12th Int. Conf. VLSI Design, pp.423–427 (1999).
- 7) Zhong, L. and Jha, N.K.: Interconnect-aware low-power high-level synthesis, *IEEE Trans. Comput.-Aided Des. Intergrated Circuits & Syst.*, Vol.24, pp.336–351 (2005).
- 8) Ito, K. and Seto, H.: Reducing Power Dissipation of Data Communications on LSI with Scheduling Exploration, *IPSJ Trans. SLDM*, Vol.2, pp.53–63 (2009).
- 9) Murata, M., Natakake, S., Fujiyoshi, K. and Kajitani, Y.: VLSI Module Placement based on Rectangle Packing by Sequence-Pair, *IEEE Trans. Comput.-Aided Des. Intergrated Circuits & Syst.*, Vol.15, pp.1518–1524 (1996).
- McFarland, M.C., Parker, A.C. and Camposano, R.: The High-Level Synthesis of Digital Systems, *Proc. IEEE*, Vol.78, No.2, pp.301–318 (1990).
- Heemstra de Groot, S.M., Gerez, S.H. and Herrmann, O.E.: Range-Chart-Guided Iterative Data-Flow Graph Scheduling, *IEEE Trans. Circuits Syst.-I: Fund. Theory* & Appl., Vol.39, pp.351–364 (1992).
- 12) Loeffler, C., Ligtenberg, A. and Moschytz, G.S.: Practical Fast 1-D DCT Algorithms with 11 Multiplications, *Proc. IEEE ICASSP '89*, pp.988–991 (1989).
- Parhi, K.K. and Messerschmitt, D.G.: Static Rate-Optimal Scheduling of Iterative Data-Flow Programs via Optimum Unfolding, *IEEE Trans. Computers*, Vol.40, pp.178–195 (1991).

(Received December 1, 2009) (Revised February 26, 2010) (Accepted April 14, 2010)

(Released August 16, 2010)

(Recommended by Associate Editor: Shigetoshi Nakatake)

267  $\,$  A Resource Binding Method to Reduce Data Communication Power Dissipation on LSI  $\,$ 



Hidekazu Seto received his BE and ME degrees in electrical and electronic engineering from Saitama University, Japan, in 2007 and 2009, respectively. He is currently with Canon Inc, Japan. His research interests include high-level synthesis in VLSI design and design automation of system LSIs.



**Kazuhito Ito** received his BE, ME, and Ph.D degrees in electrical engineering from Tokyo Institute of Technology, Japan, in 1987, 1989, and 1992, respectively. He is an associate professor of the Graduate School of Science and Engineering, Saitama University, Japan. His research interests include high-level synthesis in VLSI design, VLSI signal processing, and design automation of system LSIs. He is a member of IEICE, IPSJ, and IEEE.