Coupling-Aware High-level Interconnect Synthesis for Low Power

Chun-Gi Lyuh Taewhan Kim
Dept. of Electrical Engineering & Computer Science,
Advanced Information Technology Research Center
KAIST, Korea

Ki-Wook Kim
Pluris, Incorporation,
Cupertino, CA USA

Abstract – Ultra deep submicron (UDSM) technology and system-on-chip (SoC) have resulted in a considerable portion of power dissipated on buses, in which the major sources of the power dissipation are (1) the transition activities on the signal lines and (2) the coupling capacitances of the lines. However, there has been no easy way of optimizing (1) and (2) simultaneously at an early stage of the synthesis process. In this paper, we propose a new (on-chip) bus synthesis algorithm *to minimize the total sum of (1) and (2) in the microarchitecture synthesis.* Specifically, unlike the previous approaches in which (1) and (2) are minimized sequentially without any interaction between them, or only one of them is minimized, we, given a scheduled dataflow graph to be synthesized, *minimize (1) and (2) simultaneously by formulating and solving the two important issues in an integrated fashion*: *binding data transfers to buses* and *determining a (physical) order of signal lines in each bus*, both of which are the most critical factors that affect the results of (1) and (2). Experimental results on a number of benchmark problems show that the proposed integrated low-power bus synthesis algorithm reduces power consumption by 24.8%, 40.3% and 18.1% on average over those in [12] (for minimizing (1) only), [1] (for (2) only) and [12, 1] (for (1) and then (2)), respectively.

1 Introduction

With the advent of portable and high-density micro-electronic devices such as laptop personal computers and wire communication equipment, power dissipation of very large scale integrated (VLSI) circuits has become a critical concern. Further, ultra deep submicron (UDSM) VLSI and system-on-chip (SoC) have resulted in a considerable portion of power dissipated on buses, causing an increased attention on savings for power at the architectural-level synthesis.

The two major sources of power dissipation in buses are *transition activities* on buses and *coupling capacitances* of the signal lines of each bus. The dynamic power consumption on a signal line in a CMOS circuit is proportional to the frequency of transitions on the line itself, which we refer to as *self transition power*. On the other hand, as the scale of process technology shrinks, the lateral component of the capacitance dominates the total capacitance of the lines. For example, the lateral component of capacitance in metal 3 layer in a 0.35 μm CMOS process reaches 5 times the sum of fringing and vertical components when the substrate serves as a bottom plane [1]. Consequently, coupling becomes an important issue when we consider signal integrity and power dissipated by coupling (i.e., lateral) capacitance, which we refer to as *coupled transition power*. In this paper, we study the (on-chip) bus synthesis problem of minimizing the self and coupled transition powers in the microarchitecture synthesis.

There are many researches which have addressed the problem of minimizing power consumptions on buses. [2] tried to reduce power dissipation in memory intensive applications by minimizing transitions on the (off-chip) memory address buses. They reduced the activity on the memory address buses by analyzing the access patterns of behavioral arrays in the specification and organizing the arrays in memory. Various bus encoding schemes (e.g., [3, 4, 5, 6, 7, 8]) have been proposed to decrease the number of transitions at input/output (I/O) bus transitions. [9] proposed an encoding scheme to minimize coupling switchings, in which a slim encoder and decoder architecture is proposed to minimize the hardware overhead. [10, 11] have proposed scheduling and binding algorithms for minimizing (on-chip) data bus transitions. The algorithm is based on a simulated annealing process. [12, 13] proposed a technique for reducing power consumption during the bindings of hardware components (registers, buses, functional units). The problem is formulated as a max-cost multi-commodity flow problem and solve it optimally. Since the multi-commodity flow problem is NP-hard, they restricted the domain of pipelined designs with a short latency. [14] proposed a bus binding heuristic for minimizing transition activities by integrating the scheduling effects. [15] enhanced the method in [14] by proposing an (polynomial-time) optimal algorithm for every schedule instance of an input data flow graph (DFG). [1] proposed a method to determine a relative placement order of bus lines to reduce effective lateral component of capacitance.

All the forementioned approaches [1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15] are designed to minimize either (1) the transition activities on the signal lines to reduce the self transition power or (2) the coupling capacitances of the lines to reduce coupled transition power, but not both, causing locally optimized low-power bus designs. On the contrary, in this paper we propose a new bus synthesis algorithm which considers the minimization of (1) and (2) together to generate globally optimized bus designs. Specifically, given a scheduled dataflow graph to be synthesized, we *minimize (1) and (2) simultaneously by formulating and solving the two important issues in an integrated fashion*: *binding data transfers to buses* and *determining a (physical) order of signal lines in each bus*, both of which are the most critical factors that affect the results of (1) and (2).

2 Preliminaries

2.1 Interconnect Power Model

The dynamic power consumed by interconnects and drivers for the period of execution of T clock steps is given by

$$P_{dyn} = (X_T \cdot (C_s + C_l) + Y_T \cdot C_c) \cdot V_{dd}^2 \tag{1}$$

where C_s are C_l are self capacitances, C_c is coupling capacitance (See Figure 1.), and V_{dd} is the supply voltage [9]. X_T and Y_T are the numbers of effective transitions during T clock steps for C_s (and C_l) and C_c, respectively. X_T and Y_T are formulated in the following ways.

The self transition activity for the self capacitances C_s and C_l is proportional to the number of rising switching activities of interconnects. Let $p_{r,s}(i,j,t)$ denote the transition probability that the

0-7803-7607-2/02/$17.00 ©2002 IEEE

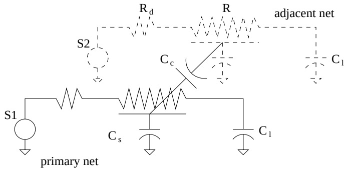


Figure 1: A distributed RC model for the interconnects.

signal value of bit-line j^1 of bus i changes from state $r \in \{0,1\}$ to $s \in \{0,1\}$ at clock step t. Then, since the capacitances C_s are C_l will be charged up only when a low-to-high signal transition takes place, the amount of self transition activities, $X_T(i,j)$, on line j of bus i for the execution of T clock steps is expressed as

$$X_T(i,j) = \sum_{t=1}^{T} p_{0,1}(i,j,t).$$

Let $X_T(i) = \sum_{j=0}^{W-1} X_T(i,j)$, which is the total sum of the self transition activities over all lines of bus i. Let $\mathcal{B}$ be the set of buses. Then, the total amount of self transition activities, X_T, for T clock steps on the buses in $\mathcal{B}$ is computed by

$$X_T = \sum_{\forall i \in \mathcal{B}} X_T(i). \tag{2}$$

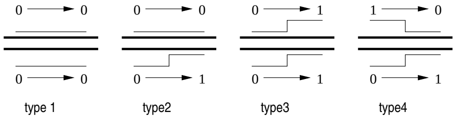


Figure 2: Signal transition relations on two bit-lines: No switching (type 1), single line switching (type 2), both line switching to the same states (type 3), and both line switching to the opposite states (type 4).

On the other hand, the amount of coupled transition activities, Y_T, is computed based on the switching relation between physically adjacent wires. There are four types of possible transitions, as depicted in Figure 2, when we consider the dynamic charge distribution over coupling capacitance C_c at the presence of two parallel wires placed with minimum spacing. In type 1, no signal transitions occur on both lines. Consequently, no dynamic charge distribution over C_c takes place. Type 2 refers to the case when exactly one of the two signals makes a transition to cause C_c being charged up to $\alpha C_c V_{dd}$ where α is a constant factor. In type 3, both signals make transitions (high-to-low or low-to-high) to the same states, resulting in C_c not being charged. (We assume that there is no misallignment of the two transitions.) Finally, in type 4 one signal transits from low state to high while the other signal transits from high to low, charging up to $\beta C_c V_{dd}$ where β is a constant factor. The effective capacitance by type 4 is larger than that by type 2, and the value of β is usually two times of the value of α. (We use β=2 and α=1 in our estimation of the (scaled) coupled transition activity.)

Let $p_{pq,rs}(i,j_1,j_2,t)$ denote the probability that the signal on line j_1 of bus i is in state p at clock step $t-1$ and in r at t while the signal on line j_2 of bus i is in state q at clock step $t-1$ and in s at t ($p,q,r,s \in \{0,1\}$). Then, the amount of coupled transition activities, $Y_T(i,j_1,j_2)$, for T clock steps on a pair of lines j_1 and j_2 in bus i is expressed as

$$\begin{aligned} &Y_T(i,j_1,j_2) = \\ &\quad \sum_{t=1}^{T} (\alpha \cdot (\sum_{s=0,1} (p_{ss,01}(i,j_1,j_2,t) + p_{ss,10}(i,j_1,j_2,t))) \\ &\quad + \beta \cdot (p_{01,10}(i,j_1,j_2,t) + p_{10,01}(i,j_1,j_2,t))). \end{aligned}$$

Then, the total amount of coupled transition activities, Y_T, for T clock steps on the buses[1] in $\mathcal{B}$ is computed by

$$Y_T = \sum_{\forall\, i \in \mathcal{B}} \sum_{j=0}^{W-2} Y_T(i,j,j+1). \tag{3}$$

2.2 Problem Definition and Examples

Suppose we have a scheduled DFG as input, from which we can determine the data values to be transferred at each clock step. We assume that the number of buses available to use is given. We define $\gamma = \frac{C_c}{C_s+C_l}$ for a line, which we refer to as the *capacitance ratio*. The capacitance ratio increases as the aspect ratio of the interconnect increases. Then, *the synthesis problem is, given a scheduled DFG with an execution profile for T clock steps and a value of capacitance ratio* γ, *(i) to assign the data transfers to the buses and (ii) to determine the physical order of the bit-lines in each bus so as to minimize the total self and coupled transition power,* P_{dyn} in *Eq.*(1), which is then equivalent to minimize the weighted sum of self and coupled transition activities Z_T:

$$Z_T = X_T + \gamma \cdot Y_T \tag{4}$$

because $P_{dyn} = (X_T \cdot (C_s + C_l) + Y_T \cdot C_c) \cdot V_{dd}^2 = (C_s + C_l) \cdot (X_T + \gamma \cdot Y_T) \cdot V_{dd}^2$.

Figure 3 shows several examples to illustrate how the bus binding and bit-line ordering affect the results of self and coupled transition activities. We are given a set of 4-bit data transfers $D_0, \cdots, D_7$ with an execution profile: D_0 (=0111) and D_1 (=0010) at clock step 1, D_2 (=1110) and D_3 (=0110) at clock step 2, D_4 (=0110) and D_5 (=1110) at clock step 3, and D_6 (=1100) and D_7 (=1101) at clock step 4. Figure 3(a) shows a random binding of the data transfers to two buses A (=$[a_3a_2a_1a_0]$) and B (=$[b_3b_2b_1b_0]$), given a fixed (physical) order of the bit-lines of the buses. (The orders from left to right are (a_3,a_2,a_1,a_0) and (b_3,b_2,b_1,b_0).) Thus, the number of effective bit-transitions (marked with boxes in Figure 3(a)), which accounts for the amount of self transition activities X_T, is 5, while the number of the adjacent pairs of bit-transitions, which accounts for the amount of coupled transition activities Y_T, is 7. Consequently, when we set capacitance ratio γ to 3, the total amount of transition activities Z_T becomes 5+3·7 = 26.

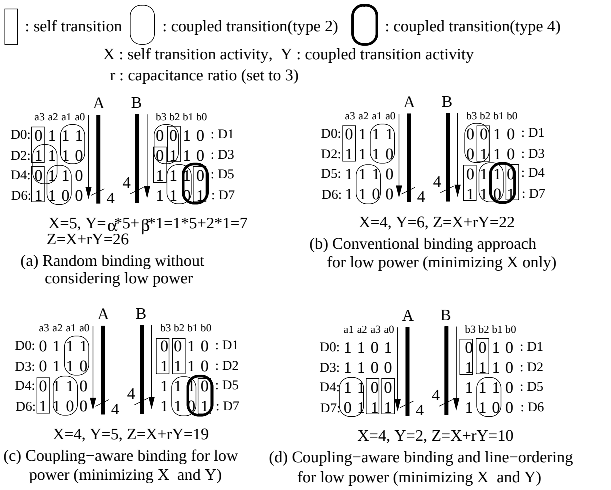
Figure 3(b) shows a binding solution with a minimum value of X_T, given a fixed order of the bit-lines. Most of the existing high-level binding approaches belong to the optimization in Figure 3(b), and attempt to reorder the bit-lines to reduce the coupled transition activity at a later stage of the synthesis process. On the other hand, Figure 3(c) shows the binding solution obtained by simultaneously minimizing the self and coupled transition activities, assuming a fixed order of the bit-lines. Since the flexibility of line ordering is not taken into account, there is still a room to improve the solution. Finally, Figure 3(d) shows a binding solution obtained by integrating line-ordering so that the effects of line-ordering on binding is exploited to minimize the self and coupled transition activities together, where the total amount of transition activities becomes 10. In fact, Figure 3(d) is the one produced by our proposed coupling-aware (integrated) binding algorithm for low power.

3 Coupling-Aware Interconnect Synthesis

3.1 The Algorithm

The input to our algorithm is a set of scheduled data transfers to be bound to buses. For example, Figure 4(a) shows a segment of scheduled DFG with T clock steps, using two ALUs and four buses. The corresponding input data transfers scheduled at each clock step is summarized in Figure 4(b). The data transfers scheduled at different clock steps can share the buses. Table 1 shows the values of the data transfers generated (i.e., execution profile) when the DFG is simulated, starting with a set of random input data values, for I

[1] For W-bit bus, bit-line 0 corresponds to LSB and $W-1$ to MSB. However, by a physical reordering of bit-lines the indexes are updated according to the physical order.



(a) Random binding without considering low power

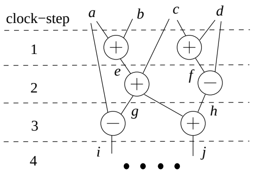
(b) Conventional binding approach for low power (minimizing X only)

(c) Coupling−aware binding for low power (minimizing X and Y)

(d) Coupling−aware binding and line−ordering for low power (minimizing X and Y)

Figure 3: Examples demonstrating the effects of bus binding and/or bit-line ordering on the self and/or coupled transition powers.

(=4) number of iterations. The highlighted entries indicate the data values to be transferred to the corresponding clock step of the iteration. Then, our optimization problem is, given an execution profile for the data values of the scheduled data transfers, to bind the data transfers to buses and (physically) order the bit-lines of buses so that the quantity of Z_T in *Eq*.(4) is minimized.



(a) A scheduled CDFG

c−step	data transfers
1	*a, b, c, d*
2	*c, d, e, f*
3	*a, g, h*
• • •	

(b) The scheduled data transfers for (a)

Figure 4: An example of scheduled DFG and its data transfers.

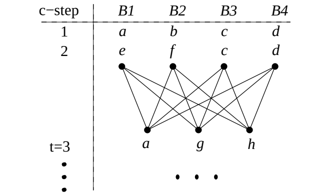
iteration	c-step	a	b	c	d	e	f	g	h
1	1	**0111**	**0011**	**0100**	**0010**	0000	0000	0000	0000
	2	0111	0011	**0100**	**0010**	**0010**	**0110**	0000	0000
	3	**0111**	0011	0100	0010	0010	0110	**1110**	**0000**
	…				…				
2	1	**0001**	**0000**	**0011**	**1110**	0010	0110	1110	0000
	2	0001	0000	**0011**	**1110**	**0001**	**0001**	1110	0000
	3	**0001**	0000	0011	1110	0001	0001	**1110**	**1111**
	…				…				
3	1	**0111**	**0101**	**1000**	**1111**	0001	0001	1110	1l11
	2	0111	0101	**1000**	**1111**	**0100**	**1111**	1110	1111
	3	**0111**	0101	1000	1111	0100	1111	**0100**	**1110**
	…				…				
4	1	**1101**	**0100**	**1001**	**0110**	0100	1111	0100	1110
	2	1101	0100	**1001**	**0110**	**0001**	**1111**	0100	1110
	3	**1101**	0100	1001	0110	0001	1111	**0000**	**0100**
	…				…				

Table 1: Data value profile for the execution of four iteration of the DFG in in Figure 4(a).

In our algorithm, bus binding and bit-line ordering are performed incrementally, one clock step (of the DFG) at a time, from the first clock step to the last. Suppose we have already performed bus binding and ordering up through clock step $t-1$. Let $\mathcal{B}$ denote the set of buses and $\mathcal{D}$ denote the set of data transfers to be bound to the buses in $\mathcal{B}$ at clock step t. $X_t(B,x)$ is recursively defined as the amount of the self transition activities on B up to t, in which data transfer $x \in \mathcal{D}$ is bound to bus $B \in \mathcal{B}$ at t, given that $X_{t-1}(B)$ (i.e., *Eq*.(2)) has already been computed during the preceding iterations of our algorithm. Likewise, $Y_t(B,x)$ is similarly defined as the amount of the coupled transition activities up to t. (The calculation procedures for $X_t(B,x)$ and $Y_t(B,x)$ will be described in section 3.2.)

The total amount of self and coupled transition activities on B, $Z_t(B,x)$, up to clock step t in which data transfer x is to be bound bus B at t is computed by

$$Z_t(B,x) = X_t(B,x) + \gamma \cdot Y_t(B,x). \qquad (5)$$



(a) Possible bndings of data transfers to buses at t=3

	B1	*B2*	*B3*	*B4*
a	156	**137**	58	184
g	154	154	**46**	178
h	**155**	147	48	182

(b) A cost table for (a)

Figure 5: An example of optimal binding.

The transition activity costs $Z_t(\cdot)$'s (in *Eq*.(5)) can be summarized in a cost table. Figure 5(b) shows an example of cost table for clock step 3 in the first iteration of the DFG in Figure 5(a) using the data values in Table 1. Thus, the optimization problem is to bind data transfers to buses (together with reordering of bit-lines of buses implied by the cost computation) so that the sum of the corresponding transition activity costs $Z_t(\cdot)$'s is minimum. Since each data transfer is bound to one bus and two data transfers cannot be bound to the same bus, the problem of minimizing the total cost can be modeled as bipartite weighted matching problem (BWMP). Consequently, the *Hungarian method* [16] which finds an optimal solution in $O(m^3)$ arithmetic operations where m (=$|\mathcal{B}|$) is the number of buses can be employed. For example, the circled entries in Figure 5(b) correspond to an optimal binding.

```
CBUS-lp: Coupling-aware Bus synthesis for low power(DFG, γ)
• Generate a scheduled data transfers from DFG;
• Simulate DFG for I iterations and obtain the profile of data values;
• Randomly bind data transfers at clock step 1 to buses;
• Set Q = {the binding result in clock step 1};
• Set t=2;
while (t ≤ T) do /* T: latency of DFG */
    foreach (pair of x ∈ D, b ∈ B) do /* repeats I times */
        • Compute $X_t(B,x)$;
              /* self transitions up to t (section 3.2) */
        • Compute $Y_t(B,x)$;
              /* coupled transitions up to t (C-Order in section 3.2) */
        • Compute $Z_t(B,x) = X_t(B,x) + \gamma \cdot Y_t(B,x)$;
    endfor
    • Construct a cost table, R, using $Z_t(\cdot)$'s;
    • Find an optimal binding, $Q_t$, from R;
    • Set Q = Q∪{$Q_t$};
    • Update $X_t(\cdot)$ from $X_{t-1}(\cdot)$ and $Q_t$; /* Eq.(6) */
    • Update $Y_t(\cdot)$ from $Y_{t-1}(\cdot)$ and $Q_t$; /* Eq.(7) */
    • Set t = t+1;
endwhile
• return Q;
```

Figure 6: The proposed coupling-aware bus binding algorithm for low power.

The overall flow of our algorithm is summarized in Figure 6. Set $\mathcal{Q}$ after $t-1$ iterations of the *while-loop* contains the bindings of the data transfers to clock step t to buses. Also, by product we obtain, from $\mathcal{Q}$, the most 'promising' ordering of the bit-lines of buses for low-power when the data values generated up to t clock steps are taken into account. (The details will be discussed in section 3.2.) If we have simulated the input DFG for an I number of iterations, the *for-loop* of the algorithm is executed I times, one for using the data values at clock step t of each iteration of the DFG simulation. Consequently, the time complexity of the algorithm is bounded by $O(T \cdot I \cdot |\mathcal{B}|^2 \cdot (m_s + m_c))$ + $O(T \cdot |\mathcal{B}|^3)$ where the second term include the complexity of the BWMP, and m_s and m_c stand for the times in computing $X_t(\cdot)$ and $Y_t(\cdot)$ (in *Eq*.(5)) for a pair of bus and data transfer, and are polynomial-time bounded (section 3.2). We now provide the details on the cost computations.

3.2 Cost Computations

Suppose we have binding result $\mathcal{Q}$ from clock step 1 to t-1. Specifically, we have known (*info-1*) a sequence of data values transferred from clock step 1 to t-1 by each bus (used for computing X_{t-1}) and (*info-2*) the amount of coupled transitions for every pair of bit-lines in each bus from clock step 1 to t-1 (used for computing Y_{t-1}). Now, given *info-1* and *info-2* up to $t-1$, we want to compute $X_t(B,x)$ and $Y_t(B,x)$.

- **Self transition activity** $X_t(B,x)$: Let $(D_1, D_2, \cdots, D_{t-1})$ denote the sequence of data values (i.e., *info-1*) bound to bus B and $D_{t-1} = (d_{W-1} \cdots d_1 d_0)$, $d_j \in \{0,1\}, j = 0, \cdots W-1$. Let D_t^x be the value of data transfer x at clock step t and $D_t^x = (d_{W-1}^x \cdots d_1^x d_0^x)$, $d_j^x \in \{0,1\}, j = 0, \cdots W-1$. We define $\delta_{p,q}(v,w) = 1$ if $v = p$ and $w = q$, and 0 otherwise, $p,q,x,y \in \{0,1\}$. Then, the *incremental self transition activity*, $\Delta X_t(B,x)$, for binding data transfer x to bus B at t is computed by

$$\Delta X_t(B,x) = \sum_{j=0}^{W-1} \delta_{0,1}(d_j, d_j^x).$$

Consequently, the total amount of self transition activities in B up to t in which x is bound to B at t is computed by

$$X_t(B,x) = X_{t-1}(B) + \Delta X_t(B,x) \qquad (6)$$

in constant time where $X_{t-1}(B)$ is the amount of self transition activities in B up to $t-1$, and has already been known according to *info-1*.

If the binding of x to B is contained in the BWMP solution (in section 3.1) at clock step t, we update *info-1* by setting $X_t(B)$ to $X_{t-1}(B)$ + $\Delta X_t(B,x)$.

- **Coupled transition activity** $Y_t(B,x)$: Let B and D_t^x be the bus and data values defined before. Let $\delta_{pq,rs}(x,y,v,w) = 1$ if $x = p$, $y = q$, $v = r$ and $w = s$, and 0 otherwise, $p,q,r,s,x,y,v,w \in \{0,1\}$. Then, the *incremental coupled transition activity*, $\Delta Y_t(B,\ j_1, j_2, x)$, for bit-lines $j_1, j_2 \in B$ when data transfer x is bound to B at t (i.e., coupled transitions for the signals only at t) is given by

$$\begin{aligned}
&\Delta Y_t(B, j_1, j_2, x) = \\
&\alpha \cdot (\sum_{s=0,1} (\delta_{ss,01}(d_{j_1}, d_{j_1}^x, d_{j_2}, d_{j_2}^x) + \delta_{ss,10}(d_{j_1}, d_{j_1}^x, d_{j_2}, d_{j_2}^x))) \\
&\quad + \beta \cdot (\delta_{01,10}(d_{j_1}, d_{j_1}^x, d_{j_2}, d_{j_2}^x) + \delta_{10,01}(d_{j_1}, d_{j_1}^x, d_{j_2}, d_{j_2}^x)).
\end{aligned}$$

Then, the amount of coupled transition activities, $Y_t(B, j_1, j_2, x)$, between bit-lines j_1 and j_2 up through clock step t in which x is bound B at t is computed by

$$Y_t(B, j_1, j_2, x) = Y_{t-1}(B, j_1, j_2) + \Delta Y_t(B, j_1, j_2, x) \qquad (7)$$

in constant time where the value of $Y_{t-1}(B, j_1, j_2, x)$ has already been known according to *info-2*.

Now, we want to compute the amount of coupled transition activities on B, $Y_t(B,x)$, up to t at which x is bound to B at t. Let $G_{x,B}(V,E)$ be an edge weighted graph where each node in V represents a bit-line of B and weight $w(u,v)$ = $Y_t(B, j_1, j_2, x)$ is assigned to the edge connecting two nodes u and v corresponding to bit-line j_1 and j_2 of B. Since we want to find an order of bit-lines so that the sum of associated coupled transition activity costs $Y_t(B,\cdot,\cdot,x)$'s (in *Eq*.(7)) is minimum, the problem is to find a minimum weighted path cover (MWPC) in $G_{x,B}(V,E)$.

Since the MWPC problem is NP-complete (reducible from Hamiltonian path problem), we use a heuristic algorithm, called C-Order, similar to Kruskal's maximum spanning tree algorithm [17]. The algorithm is greedy in that at each step, the edge with the smallest weight is selected that does not cause a cycle and does not increase the degree of a node to more than two. Let $PH(G_{x,B}(V,E))$ denote the path cover produced by applying C-Order to $G_{x,B}(V,E)$. C-Order is shown in Figure 7. The input $G_{x,B}(V,E)$ to C-Order can be constructed in $O(W^2)$ because there are $\frac{W \cdot (W-1)}{2}$ edges in G. Since the checking of a cycle and node with degree > 2 associated with an edge e_i for each *for-loop* of the C-Order can be done in $O(|E|)$ = $O(W^2)$. Thus, the time complexity of C-Order is bounded by $O(|E| \cdot \log|E|)$ = $O(W^2 \cdot \log W)$ since the time to sort edges is dominant. Figure 8 shows an example of finding a min-cost path cover using C-Order. Then, we estimate the quantity of $Y_t(B,x)$ by

$$\tilde{Y}_t(B,x) = \sum_{\forall\ edge\ e \in PH(G_{a,B}(V,E))} w(e). \qquad (8)$$

```
C-Order: Coupling-aware bit-line ordering($G_{x,B}(V,E)$)
                                  /* $B$: bus, $x$: data transfer */
• Set $L$ = sorted edge list of $E$, the smallest weight first;
• Let $L$ = $(e_1, e_2, \cdots, e_n)$;
• Set $G'(V',E')$ with $V'$ = $E'$ = $\phi$;
for ($i = 1, 2, \cdots, n$) do
    • Add $e_i$ to $G'$;
    if ($G'$ has a cycle) or
          ($G'$ has a node $v$ of degree $> 2$) do
         • Delete $e_i$ from $G'$; /* undo */
    endif
endfor
• return $G'$; /* a min-cost path cover */
```

Figure 7: Heuristic algorithm for line ordering with min-cost coupled transition activity.

If the binding of x to B is contained in the BWMP solution (in section 3.1) at clock step t, we update *info-2* by setting $Y_t(B,\cdot,\cdot)$ to $Y_{t-1}(B,\cdot,\cdot)$ + $\Delta Y_t(B,\cdot,\cdot,x)$, for every pair of bit-lines of bus B to be used in the next clock step.

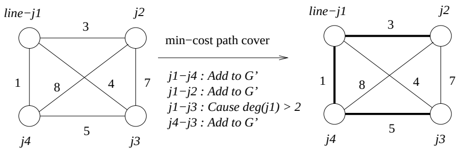


Figure 8: An example of finding min-cost path cover by C-Order.

4 Experimental Results

The proposed coupling-aware binding algorithm CBUS-lp was implemented in C++ and is executed on a Intel Pentium IV computer. We tested a set of high-level synthesis benchmark designs [18] in the experiments. The experiments were performed to check how much the dynamic power consumption (in terms of the amount of self and coupled transition activities in *Eq*.(4)) is reduced using CBUS-lp compared to the existing approaches.

Table 2 shows comparisons of the amounts of self and coupled transition activities, measured in terms of the quantity of Z_T in *Eq*.(4), for the designs produced by the *random-binding*, the designs produced by the network-flow based *optimal-binding* in [12] that minimizes the power dissipated by the self transitions only, the designs produced by the *bit-line ordering* heuristic in [1] that minimizes the power dissipated by the coupled transitions only, the designs produced by the application of *optimal-binding* followed by *bit-line ordering* ([12]+[1]), and the designs generated by CBUS-lp.

We tested each design three times, setting the value of capacitance ration γ (in *Eq*.(4)) to 1, 3, and 5. Note that the value of γ indicates the relative importance of the coupled transitions over the self transitions. Consequently, when the minimization of coupled transitions is less emphasized than the minimization of self transitions, i.e., γ=1, the improvements by CBUS-lp over [12]+[1] are not significant, (even 3.4% worse in EWF). However, as the value of γ increases, the effectiveness of CBUS-lp is clear as indicated in the table since it takes into account the minimization of self and coupled transitions simultaneously. In summary, the comparisons of the results reveal that CBUS-lp[4] is quite effective to produce high-quality bus binding solutions for low-power, reducing both the self and coupled transition activities by 44.0%, 24.8%, 40.3% and

[4] Each design is tested by CBUS-lp within 1 minute.

	γ	*random*	[12]	[1]	[12]+[1]	CBUS-lp	reduction(%) over			
							random	[12]	[1]	[12]+[1]
COMPLEX	1	665	571	600	491	424	41.18	32.00	33.01	17.72
	3	1,467	1,273	1,272	1,033	830	44.70	36.39	35.76	20.78
	5	2,269	1,975	1,944	1,575	1,236	46.10	38.13	36.89	22.05
DIFF	1	237	120	248	115	97	62.87	29.02	64.96	25.14
	3	501	266	534	251	186	63.98	32.74	66.32	28.53
	5	765	412	820	387	270	65.22	35.67	67.60	31.43
DIFF.2	1	377	155	380	152	110	73.10	35.89	73.37	34.30
	3	805	341	814	332	205	75.37	42.17	75.65	40.52
	5	1,233	527	1,248	512	294	76.85	45.97	77.13	44.35
EWF[3]	1	537,661	362,374	532,608	355,153	371,548	31.72	-0.84	30.88	-3.48
	3	1,134,267	769,716	1,119,108	748,053	770,078	32.34	0.42	31.37	-2.63
	5	1,730,873	1,177,058	1,705,608	1,140,953	1,168,608	32.59	0.93	31.57	-2.28
IDCT	1	3,835	3,516	3,462	3,268	2,930	26.76	20.01	16.89	12.45
	3	8,719	7,978	7,600	7,234	6,318	28.49	21.82	17.46	13.41
	5	13.603	12,440	11,738	11,300	9,644	29.53	22.93	18.11	14.23
KALMAN	1	16,747	16,044	16,164	15,014	13,744	20.71	17.86	17.08	10.71
	3	36,543	35,396	34,794	32,306	28,510	22.71	20.37	18.63	12.35
	5	56,339	54,748	53,424	49,598	43,276	14.91	15.60	13.33	6.65
average							**44.01**	**24.84**	**40.33**	**18.12**

Table 2: Comparisons of the amounts of self and coupled transition activities (Z_T in *Eq.*(4)) for HLS benchmark designs.

18.1% on average compared to those by *random-binding*, [12], [1] and [12]+[1], respectively.

Figure 9 graphically shows the (normalized) comparisons of the total amount of self and coupled transition activities on buses, averaged over γ = 1, 2, 3, 4, 5, and 6, optimized by existing algorithms and CBUS-lp. The comparisons strongly suggest that the proposed approach, which simultaneously minimizes the self and coupled transitions in the early stage of the synthesis process, can save the power consumption on interconnects considerably, which otherwise, could be hard or too late to achieve in a later synthesis process (e.g., logic/circuit-level) or layout phase.

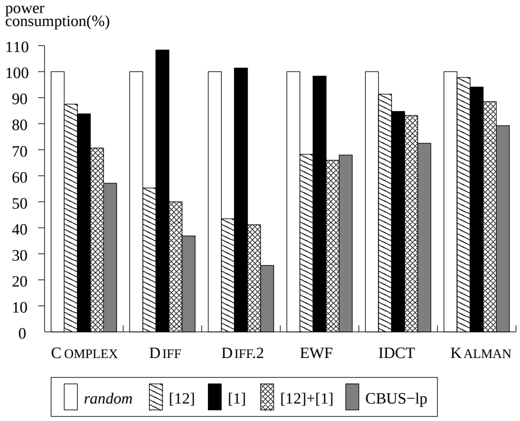


Figure 9: Summary of the comparisons of the dynamic power consumptions averaged over γ = 1, 2, 3, 4, 5, and 6.

5 Conclusions

In this paper, we proposed a new interconnect optimization algorithm for low power, which considers the minimization of (1) *the transition activities on the signal lines* and (2) *the coupling capacitances of the lines simultaneously* in the microarchitecture synthesis to overcome the limitation of the previous works in which (1) and (2) are minimized sequentially without any interaction between them, or only one of them is minimized, resulting in locally optimized interconnect designs. Specifically, for given a scheduled dataflow graph to be synthesized, we *minimized (1) and (2) simultaneously by formulating and solving the two important issues in an integrated fashion*: *binding data transfers to buses* and *(physical) ordering signal lines in each bus*, both of which are the most critical factors that affect the results of (1) and (2). From a set of experimental results on a number of benchmark problems we confirmed that the proposed interconnect synthesis algorithm is quite useful in designing reliable and low-power interconnects in UDSM technology, reducing the power consumption by 24.8%, 40.3% and 18.1%, on the average, over those by [12] (for minimizing (1) only), [1] (for (2) only) and [12, 1] (for (1) and then (2)), respectively.

Acknowledgement: This work was supported by the Korea Science and Engineering Fundation (KOSEF) through the Advanced Information Technology Research Center (AITrc).

REFERENCES

[1] Y. Shin and T. Sakurai, “Coupling-Driven Bus Design for Low-Power Application-Specific Systems,” *Proc. of DAC*, 2001.

[2] P. R. Panda and N. D. Dutt, “Low-Power Memory Mapping Through Reducing Address Bus Activity,” *IEEE Tran. on VLSI Systems*, Vol. 7, No. 3, 1999.

[3] S. Ramprasad, N. R. Shanbhag, and I. Hajj, “A Coding Framework for Low-Power Address and Data Busses,” *IEEE Trans. on VLSI Systems*, Vol. 3, No. 1, 1995.

[4] H. Mehta, R. M. Owens and M. J. Irwin, “Some issues in Gray code addressing,” *Proc. of Sixth Great Lakes Symposium on VLSI*, 1996.

[5] C. L. Su, C. Y. Tsui, and A. M. Despain, “Saving power in the control path of embedded processors,” *IEEE Design and Test of Computers*, Vol. 11, No. 4, 1994.

[6] E. Musoll, T. Lang and J. Cortadella, “Working-zone encoding for reducing the energy in microprocessor address buses,” *IEEE Trans. on VLSI Systems*, Vol. 6, No. 4, 1998.

[7] M. R. Stan and W. P. Burleson, “Bus-invert coding for low-power I/O,” *IEEE Trans. on VLSI Systems*, Vol. 3, No. 1, 1995.

[8] S. Hong, T. Kim, U. Narayanan and K.-S. Chung, “Decomposition of bus-invert coding for low-power I/O,” *Journal of Circuits, Systems and Computers*, Vol. 10, Nos. 1 & 2, 2000.

[9] K.-W. Kim, K.-H. Baek, N. Shanbhag, C. L. Liu, and S.-M. Kang, “Coupling-Driven Signal Encoding Scheme for Low-Power Interface Design,” *Proc. of ICCAD*, 2000.

[10] A. Dasgupta and R. Karri, “Simultaneous Scheduling and Binding for Power Minimization During Microarchitecture Synthesis,” *Proc. of ISLPED*, 1995.

[11] A. Dasgupta and R. Karri, “High-Reliability, Low-Energy Microarchitecture Synthesis,” *IEEE Trans. on CAD*, Vol. 17, No. 12, 1998.

[12] J.-M. Chang and M. Pedram, “Register Allocation and Binding for Low Power,” *Proc. of DAC*, 1995.

[13] J.-M. Chang and M. Pedram, “Module Assignment for Low Power,” *Proc. of EDAC*, 1996.

[14] S. Hong and T. Kim, “Bus Optimization for Low-Power Data Path Synthesis based on Network Flow Method,” *Proc. of ICCAD*, 2000.

[15] C. Lyuh, T. Kim and C. L. Liu, “An Integrated Data Path Optimization for Low Power Based on Network Flow Method,” *Proc. of ICCAD*, 2001.

[16] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Prentice Hall, pp.247-254, 1982.

[17] A. Aho, J. Hopcroft, and J. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[18] P. R. Panda and N. D. Dutt, “High-Level Synthesis Design Repository”, *Proc. of ISSS (http://www.ics.uci.edu/ dutt)*, 1995.

[19] L. Benini, G. De Micheli, E. Macii, D. Sciuto and C. Silvano, “Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems,” *Proc. of Seventh Great Lakes Symposium on VLSI*, 1997.