

# Architectural Power Optimization by Bus Splitting

Cheng-Ta Hsieh and Massoud Pedram  
EE-Systems, University of Southern California

**Abstract** – A split-bus architecture is proposed to improve the power dissipation for global data exchange among a set of modules. The resulting bus splitting problem is formulated and solved combinatorially. Experimental results show that the power saving of the split-bus architecture compared to the monolithic-bus architecture varies from 16% to 50%, depending on the characteristics of the data transfer among the modules and the configuration of the split bus. The proposed split-bus architecture can be extended to multi-way split-bus when a large number of modules are to be connected.

## 1 Introduction

To increase the level of integration and the performance, system-on-a-chip is widely deployed in today's designs. In such designs, communication resources are allocated to connect the on-chip modules for data exchange. Two widely used communication architectures are 1) *point-to-point connection* (uni-directional) and 2) *shared bus* (bi-directional). In addition to system-on-a-chip designs, microprocessors, digital signal processors and embedded controllers also use these two types of interconnection architecture. This paper proposes a split shared-bus architecture (c.f. Figure 1) to reduce the power consumption of the monolithic shared-bus (c.f. Figure 2).

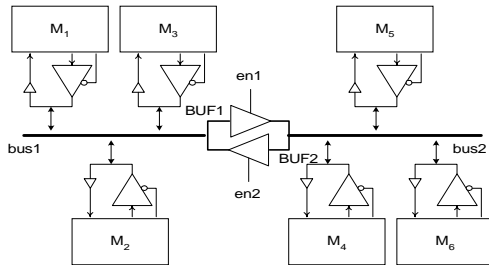


Figure 1. Split shared bus architecture.

The advantages of the shared bus architecture include simple topology, low area cost and extensibility. The disadvantages of the shared bus architecture are larger load per data-bus line, longer delay for data transfer, larger power consumption, and lower bandwidth. Fortunately, the above disadvantages, except the bandwidth, may be overcome by using a low-voltage swing signaling technique [1]. In a low-voltage swing architecture, the signal being transferred from a module is

first converted to a low-voltage swing signal and then propagated along the shared bus. The low-voltage swing is finally converted back to a full-swing signal at the input of the receiving module. In this way, the amount of the charge on the bus will only change by  $\Delta V \times C_{BUS}$ , where  $\Delta V$  is the voltage swing on the bus and  $C_{BUS}$  is the capacitive load of the bus. Therefore, the low-voltage swing bus achieves a power reduction of  $(V_{dd} - \Delta V) / V_{dd}$  compared to the case of a full-swing bus. The signal delay on the bus is also reduced by  $\Delta t = \frac{C_{BUS}(V_{dd} - \Delta V)}{I}$  where  $I$  is the average current of the driver.

Notice that bus encoding techniques reviewed in [2] can be used to further reduce the power consumption of the on-chip bus.

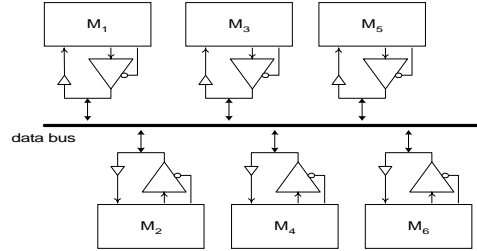


Figure 2. Monolithic shared-bus architecture.

## 2 Monolithic Bus Structure

Without loss of generality, we consider a one-bit bus. Results for a  $k$ -bit bus can be easily obtained by scaling the one-bit bus results by  $k$ . Assume we have  $n$  modules  $M_1, M_2, \dots, M_n$  connected to each other through a bi-directional shared bus as shown in Figure 2. During the architectural simulation, we simulate the system for  $p$  cycles, from cycle 1 to cycle  $p$ . In each cycle  $i$ , the data with logic value of  $V_i$  is transferred from module  $M_{SRC(i)}$  to module  $M_{DST(i)}$ .

Assume that the receiver gate for each module has minimum size and its input capacitance is  $C_g$ . Furthermore, the output capacitance of the driver for each module  $M_i$  is  $C_{o,i}$ .  $C_{BUS}$  is calculated as follows:

$$C_{BUS} = L_{BUS} \cdot C_u + C_c + \sum_{i=1}^n (C_{o,i} + C_g)$$

where  $L_{BUS}$  is the physical length of the bus,  $C_u$  denotes the capacitance per unit length of the bus, and  $C_c$  denotes the coupling capacitance due to the parallel running bus wires as well as other nearby wires on adjacent metal layers, and  $n$  is the number of modules connected to the bus.

The average energy consumption during the  $p$  cycles is:

This research was supported in part by SRC under contract No. 98-DJ-606 and by NSF under NSF-PECASE Award MIP-9628999.

$$E = 0.5 \cdot C_{BUS} \frac{V_{dd}^2}{P} \cdot \sum_{i=1}^p (V_{i-1} \oplus V_i) + \sum_{i=1}^n E_{driver, M_i}$$

where  $E_{driver, M_i}$  is the average internal energy dissipation per clock cycle of the bus driver of module  $M_i$ .

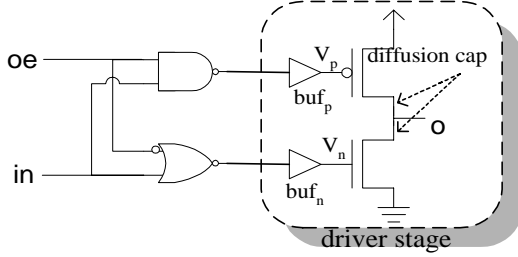


Figure 3 Circuit diagram of tri-state bus driver.

A typical tri-state driver is shown in Figure 3. Notice that  $V_p = \overline{oe \cdot in}$  and  $V_n = oe \cdot \overline{in}$ . The switching activity of  $V_p$  and  $V_n$  are:

$$\begin{aligned} sw(V_p) &= prob(oe = 1 \rightarrow 1, in = 0 \rightarrow 1) + prob(oe = 1 \rightarrow 1, in = 1 \rightarrow 0) \\ &\quad + prob(oe = 1 \rightarrow 0, in = 1 \rightarrow x) + prob(oe = 0 \rightarrow 1, in = x \rightarrow 1) \\ sw(V_n) &= prob(oe = 1 \rightarrow 1, in = 0 \rightarrow 1) + prob(oe = 1 \rightarrow 1, in = 1 \rightarrow 0) \\ &\quad + prob(oe = 1 \rightarrow 0, in = 0 \rightarrow x) + prob(oe = 0 \rightarrow 1, in = x \rightarrow 0) \end{aligned}$$

where  $prob(oe = v1 \rightarrow v2, in = v3 \rightarrow v4)$  denotes the probability for  $(oe, in) = (v1, v3)$  in the current cycle and  $(oe, in) = (v2, v4)$  in the next clock cycle;  $x$  denotes don't care. If input  $in$  is not correlated with  $oe$ , above equations can be simplified as:

$$sw(V_p) = 2prob(in)prob(oe)[1 - prob(in)prob(oe)]$$

$$sw(V_n) = 2prob(in=0)prob(oe)[1 - prob(in=0)prob(oe)]$$

where  $prob(x)$  and  $prob(x=0)$  denote the probability for  $x=1$  and  $x=0$  in a clock cycle.

The average internal energy dissipation of the driver stage per clock cycle is:

$$E_{driver} = 0.5(sw(V_p)C_{eff, bufp} + sw(V_n)C_{eff, bufn})V_{dd}^2$$

where  $C_{eff, bufp}$  ( $C_{eff, bufn}$ ) denote the physical capacitance driven by NAND2 (NOR2).

### 3 Split Bus Architecture

For a long bus line, the parasitic resistance and capacitance of the bus line are large. For example, in Figure 2, the propagation delay from module  $M_1$  to module  $M_6$  is very large. To improve the timing and power consumption of the long bus, we can partition the bus into two bus segments as shown in Figure 1. The dual-port driver at the boundary of bus1 and bus2 relays the data from one bus to the other when such data transfer is needed. Therefore the split bus architecture works in the same way as a single bus. If the intrinsic delay (and power consumption) of the dual-port driver is small compared to the rest of the bus, which is the case for a long bus connection, then the new bus architecture will always be better than the single bus architecture.

Advantages of the bus splitting are:

- Smaller parasitic load: Because the bus length is reduced, the parasitic load of each bus segment is reduced.
- Larger timing slack: Due to the smaller parasitic load of the two bus parts and because smaller output capacitances from the driver outputs are added as load to any part of the split bus, the timing slack becomes more positive.
- Smaller driver size: Because the timing slack is larger, the driver size can be made smaller while meeting the timing constraint
- Lower power consumption: Since smaller load and smaller drivers are used, the effective physical capacitance for each bus part is smaller. In the case of data being transferred within the same bus partition, the power consumption is significantly reduced because there is no switching activity in the other bus partition.
- Lower noise problems: The parallel running buses are at the greatest risk with respect to coupling noise. Reducing the bus wire length effectively reduces the amount of capacitive coupling noise.

In Figure 1, modules  $M_1, M_2$  and  $M_3$  reside in the bus on the left and modules  $M_4, M_5$  and  $M_6$  sit on the other side. Let  $BUS1$  be the set of modules in the left bus and  $BUS2$  denote the set of modules in the right bus. When  $en1$  is '1', BUF1 will relay the data from bus1 to bus2. Similarly, BUF2 will pass the data from bus2 to bus1 when  $en2$  is '1'. Note that  $en1$  and  $en2$  should not be set to '1' at the same time. When both  $en1$  and  $en2$  are '0', bus1 and bus2 are isolated from one another. In this section, we assume the driver sizes are fixed.

#### 3.1 Examples

In the following examples, we assume that the output capacitance of drivers is zero and ignore the energy consumption within the drivers. The data being transferred by any module on the data bus is modeled as an independent random variable with an average switching activity equal to  $sw$ .

The average energy consumption of the single bus architecture is calculated as:

$$E1 = 0.5 \cdot sw \cdot C_{BUS} V_{dd}^2$$

Let  $C_{BUS1}$  and  $C_{BUS2}$  denote the physical capacitance on bus1 and bus2. The average energy consumption of the split bus architecture per clock cycle is calculated as:

$$\begin{aligned} E2 &= 0.5V_{dd}^2 [sw \cdot C_{BUS1} \sum_{i \in BUS1} \sum_{j \in BUS1, i \neq j} xfer(M_i, M_j) \\ &\quad + C_{BUS2} \sum_{i \in BUS2} \sum_{j \in BUS2, i \neq j} xfer(M_i, M_j) \\ &\quad + (C_{BUS1} + C_{BUS2}) \sum_{i \in BUS1} \sum_{j \in BUS2} xfer(M_i, M_j) \\ &\quad + (C_{BUS1} + C_{BUS2}) \sum_{i \in BUS2} \sum_{j \in BUS1} xfer(M_i, M_j)] \end{aligned}$$

where  $xfer(M_i, M_j)$  denotes the probability of module  $M_i$  transferring data to module  $M_j$  in any clock cycle.

In the following examples, we set  $sw=0.5$  and normalize  $\frac{C_{BUS1}}{|BUS1|} = \frac{C_{BUS2}}{|BUS2|} = \frac{C_{BUS}}{n} = 1, V_{dd}=1$ .

**Example 1** Assume we have  $n=2k$  modules and  $|BUS1|=k-a, |BUS2|=k+a$  where  $a \in \{0,1,\dots,k-2\}$ . The probability of transferring data from module  $M_i$  to module  $M_j$  in any clock cycle is  $\frac{1}{2k(2k-1)}$ , for  $i=1..2k, j=1..2k, i \neq j$ .

$$E1 = 0.5k$$

$$E2 = 0.25 \frac{3k^3 - k^2 + a^2(k-1)}{2k^2 - k}$$

The power saving of the split bus over the monolithic bus can be calculated by:

$$\frac{E1 - E2}{E1} = 0.5 \frac{k^3 - k^2 - a^2(k-1)}{2k^3 - k^2}$$

The power saving is maximized when  $a=0$ .

For the case of  $k=2$  and  $a=0$ , power saving is 16%. When  $k \rightarrow \infty$  and  $a=0$ , the power saving is 25%.

If we set  $a=k$ , which is the case of monolithic bus, then the power saving is 0.

**Example 2** Assume that there are four modules connected to the bus. The probability of transferring data between module  $M_i$  and module  $M_j$  is specified by the label of the edge  $(M_i, M_j)$  in Figure 4.

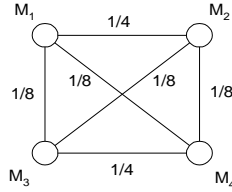


Figure 4. Data transfer probabilities for Example 2.

The energy consumption for various architectures is summarized in the following table:

Architecture	Energy
$BUS=\{M_1, M_2, M_3, M_4\}$	1
$BUS1=\{M_1, M_2\} \quad BUS2=\{M_3, M_4\}$	0.75
$BUS1=\{M_1, M_3\} \quad BUS2=\{M_2, M_4\}$	0.875
$BUS1=\{M_1, M_4\} \quad BUS2=\{M_2, M_3\}$	0.875

The bus partitioning solution with  $BUS1=\{M_1, M_2\}, BUS2=\{M_3, M_4\}$  consumes the least power because more of the data transfers are performed within each part.

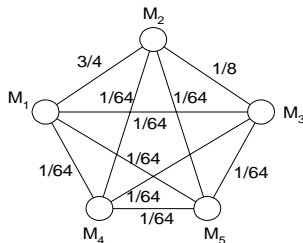


Figure 5. Data transfer probabilities for Example 3.

**Example 3** For the 5 module configuration shown in Figure 5, the power consumption for several configurations are listed below:

Architecture	Energy
$BUS=\{M_1, M_2, M_3, M_4, M_5\}$	1.25
$BUS1=\{M_1, M_2\} \quad BUS2=\{M_3, M_4, M_5\}$	0.66
$BUS1=\{M_1, M_2, M_3\} \quad BUS2=\{M_4, M_5\}$	0.79
$BUS1=\{M_2, M_3\} \quad BUS2=\{M_1, M_4, M_5\}$	1.13

The second bus splitting configuration has the lowest energy consumption, which achieves 47% reduction in the energy consumption compared to the single bus architecture. Note that although edge  $(M_2, M_3)$  has a weight of  $1/8$ , which is the second largest value in this example, adding  $M_3$  to  $BUS1=\{M_1, M_2\}$  increases  $C_{BUS1}$  and hence results in higher power dissipation.

### 3.2 An Accurate Power Consumption Model

Similar to the case of the monolithic bus, the physical capacitance on bus1 and bus2 can be calculated as:

$$C_{BUS1} = L_{BUS1} \cdot C_u + C_{c,1} + \sum_{i \in BUS1} (C_{o,i} + C_g) + C_{o,BUF1} + C_{in,BUF2}$$

$$C_{BUS2} = L_{BUS2} \cdot C_u + C_{c,2} + \sum_{i \in BUS2} (C_{o,i} + C_g) + C_{o,BUF2} + C_{in,BUF1}$$

$L_{BUS1}$  and  $L_{BUS2}$  are the bus lengths of bus1 and bus2;  $C_{ce,1}$  and  $C_{c,2}$  are the coupling capacitances for bus1 and bus2;  $C_{o,BUF1}$  and  $C_{o,BUF2}$  are the output capacitances of BUF1 and BUF2, respectively;  $C_{in,BUF1}$  and  $C_{in,BUF2}$  are input capacitance of BUF1 and BUF2. Here we assume that the wire widths of both buses are the same. Again, we assume the minimum gate size for the receiver of each module.

The logic values on bus1 and bus2 in each clock cycle  $i$  are calculated as follows:

$$V_{BUS1,i} = \begin{cases} V_{BUS1,i-1} & \text{if } M_{SRC(i)} \notin BUS1 \text{ and } M_{DST(i)} \notin BUS1 \\ V_i & \text{otherwise} \end{cases}$$

$$V_{BUS2,i} = \begin{cases} V_{BUS2,i-1} & \text{if } M_{SRC(i)} \notin BUS2 \text{ and } M_{DST(i)} \notin BUS2 \\ V_i & \text{otherwise} \end{cases}$$

where  $V_i$  denotes the logic value being transferred in clock cycle  $i$ .

The average energy consumption of the split bus architecture is calculated as:

$$E = E_{BUS1} + E_{BUS2} + E_{driver}$$

$$= 0.5C_{BUS1} \sum_{i=1}^p (V_{BUS1,i-1} \oplus V_{BUS1,i}) \frac{V_{dd}^2}{p}$$

$$+ 0.5C_{BUS2} \sum_{i=1}^p (V_{BUS2,i-1} \oplus V_{BUS2,i}) \frac{V_{dd}^2}{p}$$

$$+ \sum_{i=1}^n E_{driver, Mi} + E_{driver, BUF1} + E_{driver, BUF2}$$

where  $E_{driver, Mi}$  and  $E_{driver, BUFx}$  are the average energy consumptions per clock cycle for module  $M_i$  and buffer  $x$  and are calculated by equations in Section 2.  $p$  is the number of simulated cycles.

### 3.3 A Probabilistic Power Consumption Model

Usually  $p$  must be very large so that the collected trace becomes representative of real application data. To speed up the power consumption calculation, a probabilistic model can be used. Note that the model is only exact

under the assumption of data stationarity [4].

Assume that the data being transferred from each module can be modeled as a time-invariant random process with probability  $prob(M_i)$  for the data value to be '1'. Furthermore, assume that the data transfer at clock  $i$  ( $M_{SRC(i)}, M_{DST(i)}$ ) is not correlated with the data transfer pair ( $M_{SRC(i+1)}, M_{DST(i+1)}$ ) at clock  $i+1$ .

Let  $xfer(BUS1, BUS2)$  denote the probability of bus1 transferring data to bus2 in any clock cycle. It is calculated as:

$$xfer(BUS1, BUS2) = \sum_{i \in BUS1} \sum_{j \in BUS2} xfer(M_i, M_j).$$

$xfer(BUS2, BUS1)$  is calculated similarly.

Let  $xfer(BUS1)$  denote the probability of data transfers occurring on bus1. It is calculated as:

$$xfer(BUS1) = xfer(BUS2, BUS1) + \sum_{i \in BUS1} \sum_{j \in BUS1 \cup BUS2, j \neq i} xfer(M_i, M_j)$$

$xfer(BUS2)$  is calculated similarly.

Let  $prob(BUS1)$  denote the probability for the bus1 having a logic value '1' in a clock cycle. It is calculated as:

$$prob(BUS1) = \left\{ \sum_{i \in BUS1} \sum_{j \in BUS1 \cup BUS2, j \neq i} xfer(M_i, M_j) prob(i) + \sum_{i \in BUS2} \sum_{j \in BUS1} xfer(M_i, M_j) prob(j) \right\} / xfer(BUS1)$$

$prob(BUS2)$  is calculated similarly.

The switching activities of bus1 and bus2 (assuming temporal independence of data values on the bus) are:

$$sw(BUS1) = 2prob(BUS1)[1 - prob(BUS1)]xfer(BUS1)$$

$$sw(BUS2) = 2prob(BUS2)[1 - prob(BUS2)]xfer(BUS2)$$

Therefore, the average energy consumption per clock cycle of the split bus architecture is calculated as:

$$E = 0.5(C_{BUS1}sw(BUS1) + C_{BUS2}sw(BUS2))V_{dd}^2 + \sum_{i=1}^n E_{driver, Mi} + E_{driver, BUF1} + E_{driver, BUF2}$$

where  $E_{driver, x}$  can be calculated by the equations in Section 2.

## 4 Bus Splitting for Low Power

Assume that we perform bus splitting after the modules on the bus have been placed and the bus wires have been routed. During this design phase, the order of the modules on the bus is already known; therefore the only degree of freedom is in selecting a bus segment, from 2 to  $n-2$ , to place the dual-port driver. Let  $sw_{BUS1}(i)$  and  $E(i)$  denote the switching activity of the data on bus1 and energy dissipation on bus1 with the dual-port driver positioned at bus segment  $i$ . The symbols with subscript 'BUS2' are also defined similarly.

### Greedy Algorithm

1. Calculate the  $sw_{BUS1}(i)$  and  $sw_{BUS2}(i)$  for buffer position at segment  $i$ ,  $i=2 \dots n-2$ .
2. Calculate  $E(i)$  for buffer position at segment  $i$ ,  $i=2 \dots n-2$ .
3. Find the minimum  $E(i)$

The complexity of the algorithm is dominated by that of the first step which is  $O(p \cdot n)$ . The algorithm is obviously optimal.

When the bus splitting is performed before the system-level floor-planning is completed, we have the freedom to rearrange the order of the modules to maximize the power reduction. We first show that this problem is NP-complete.

**Theorem:** The bus splitting problem with unknown module order is an NP-hard problem.

*Proof outline:* The proof is done by converting the 'minimum cut into bounded sets' (MCBS) problem [3] with equal set sizes into the bus splitting problem. The conversion is done by forming a bus splitting problem in which the number of modules is equal to the number of vertices in the MCBS and the data transfer probability  $xfer(M_i, M_j)$  is proportional to  $x + w_{i,j}$  where  $x$  is a constant and  $w_{i,j}$  is the weight between vertex  $v_i$  and vertex  $v_j$  in the MCBS. If  $x$  is sufficiently large ( $x \gg \max(w_{i,j})$ ), then partitioning into equal-size subsets will be necessary to obtain the optimal solution to the bus splitting problem (c.f. Example 1). This completes the proof.

### Heuristic Algorithm

Because the bus splitting with unknown module order is an NP-hard problem, we may use exhaustive search for a small value of  $n$ . The number of feasible splitting is  $2^{n-1} - 1$ . In our experiments, the exhaustive search for  $n=30$  can be done within 10 minutes on a Pentium-II 266Mhz machine. If  $n$  is large, then a module clustering step is first performed to make the effective  $n$  less than or equal to some predefined value. The clustering step can be done by minimizing inter-cluster data transfers while avoiding the case that the size of certain cluster becomes much larger than the sizes of the others (to avoid the pitfall of the third configuration in Example 3). Clustering is performed by a recursive max-weight matching algorithm [5]. Next, all possible ways of bus splitting are enumerated exhaustively based on the clustering solution.

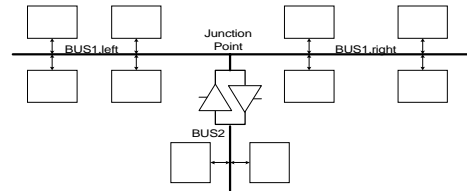


Figure 6. T-shaped bus structure.

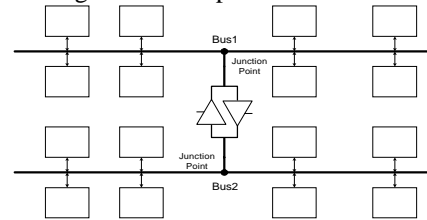


Figure 7. H-shaped bus structure.

## 5 Bus Topology Variation

Instead of aligning all the modules horizontally, we may resort to other connection topologies, when allowed,

to improve the timing or power consumption. A T-shaped configuration is suitable for unbalanced partitioning while the H-shaped configuration is suitable for balanced partitioning. Note that both configurations have better delay characteristics than a horizontally-aligned configuration.

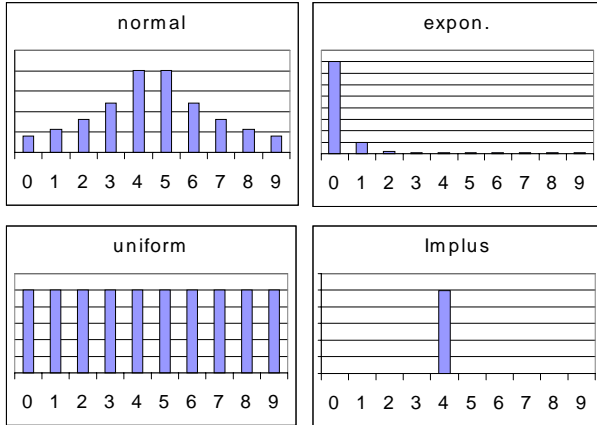


Figure 8 Transfer frequencies for various distributions.

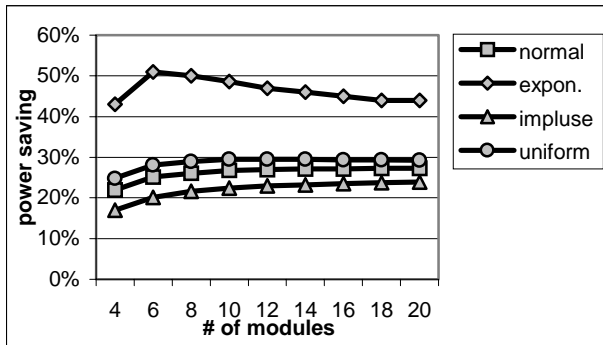


Figure 9. Power saving for various distributions of transfer frequencies.

## 6 Experimental Results

There are no existing benchmarks to use for this problem. We therefore generated our own test benches. In our experimental setup, the assumptions discussed in Section 3.1 are adopted. In addition, the data exchange frequency between any two modules  $M_i$  and  $M_j$  is randomly weighted by an integer between 0 and 9 and follows one of the probability distributions specified in Figure 8 in a randomly generated test case. The height of each bar in Figure 8 shows the (relative) probability of the data exchange frequency between a pair of modules to be equal to the x-axis value.

Each point in Figure 9 represents the average power savings of the split bus over the monolithic bus, given  $k$  modules ( $k=4...20$ ) are connected to the bus, for 500 randomly generated test cases in which the transfer frequencies between pair of modules follow a given distribution  $dist$ . In the following discussion, we refer to each point in Figure 9 by  $(k, dist)$ , e.g., (4,normal).

Simulation results show that the test cases with exponential distributions have the largest average power saving while the test cases with impulse distributions have the smallest power saving. This is because test cases with exponential distribution have fewer high-frequency transfers between modules and therefore it becomes easier to keep the modules with high-frequency transfers within one part of the split bus. On the other hand, the impulse distribution has no variation in transfer frequencies and therefore it provides the smallest opportunity for optimization.

One important observation is that an anomaly occurs at point (6, exponential), which has the highest average power saving compared to other points of the same distribution (c.f. Figure 9). The reason is that (6, exponential) has a higher power saving opportunity compared to (4, exponential) due to the fact that the unbalance bus partitions ( $|BUS1|=2, |BUS2|=4$ ) or ( $|BUS1|=4, |BUS2|=2$ ) can result in larger power saving as was illustrated in Example 3. For points  $(k, exponential)$  where  $k > 6$ , it is harder to achieve power saving because modules are more likely to be tightly coupled. For distributions other than the exponential distribution, the frequency distributions have much lower variance than that of the exponential distribution. Therefore the results follow the trend predicted by Example 1 in Section 3.1.

## 7 Conclusion

A split-bus architecture was proposed to improve the speed and power dissipation for global data exchange among a set of modules. The power model for split bus was presented and the bus splitting problem was solved combinatorially. Experimental results showed that the power saving of the split-bus compared to the monolithic-bus architecture varies from 16% to 50%, depending on the characteristics of the data transfer among the modules and the configuration of the split bus. T-shaped bus and H-shaped bus structures were proposed to further improve the bus performance. The proposed split-bus architecture can be extended to multi-way split bus when a large number of modules are to be connected.

## 8 Reference

- [1] Y. Nakagome *et al.* "Sub-1-V Swing Internal Bus Architecture for Future Low-Power ULSI's," in *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 4, pp. 414-419, 1993.
- [2] E. Macii, M. Pedram and F. Somenzi, "High level power modeling, estimation and optimization," *IEEE Trans. on Computer Aided Design*, Vol. 17, No. 11, pp. 1061-1079, Nov. 1998.
- [3] M. R. Garey, D. S. Johnson, "Computer and Intractability, A Guide to the Theory of NP-Completeness," *W.H. Freeman and Company*, 1979.
- [4] A. Leon-Garcia, "Probability and Random Processes for Electrical Engineering," Second Edition, Addison Wesley, 1993.
- [5] T. Lengauer, "Combinatorial Algorithms for Integrated Circuit Layout," John Wiley & Sons, 1990.