

Energy Optimization for Application-Specific NOC with Multi-Mode Switches

Kuei-Chung Chang
Department of Information Management
WuFeng Institute of Technology
Chia-yi Taiwan, R.O.C.

Abstract

As the number of cores on a chip increases, power consumed by the communication structures takes significant portion of the overall power-budget. As technology scales to deep sub-micron processes, leakage power becomes increasingly significant as compared to dynamic power. In this paper, we design a power-aware switch including three operating modes, and they are *normal mode*, *lease-line mode*, and *off mode*. We can dynamically control the mode of switches in a NoC according to the profiling results of specific applications. We take MPEG-4 decoder as our case studies, and experimental results show that the total power saving of the NoC can be approximated to 90% if we dynamically control the switch mode when the communication characteristics of the embedded software are well known.

Keywords: NoC, low power, switch, interconnects

1 INTRODUCTION

As parallel chip architectures scale in size, on-chip networks have becoming the main communication architecture, replacing dedicated interconnects and shared buses. NoC architectures have to deliver good latency-throughput performance in the face of very tight power and area budgets. These trends make on-chip network design to be one of the most challenging and significant design problems.

Since different architectures are efficient for different classes of applications, machine of the future will contain a number of different execution cores, each being efficient for a class of applications. The search for flexibility in design with-

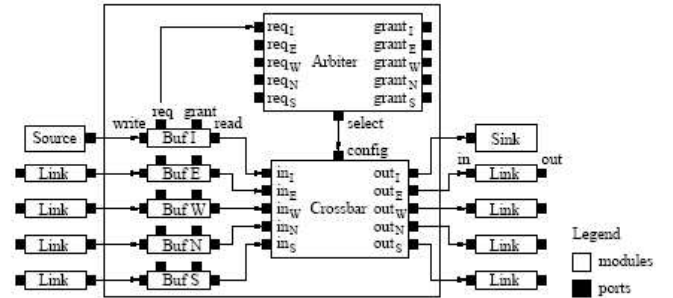


Figure 1: A simple wormhole router

out paying a significant area, time, power cost is the primary problem for application-specific SOC design. NoC-based interconnection performance correlates strongly with the topology selected for implementation. In the case of custom-built NoC architectures, switch blocks might not be identical; their design and placement depend on the specific communication requirements. Irregular network architectures might be necessary for realizing application-specific SoCs, such as those in mobile-phone systems, where different heterogeneous blocks with varying communicating requirements must be linked.

In a realistic SoC environment, different functions will be mapped to different parts of the SoC, and the traffic will exhibit highly localized patterns. Localization makes the shorter communication distance, and it takes low latency and energy for transmitting a message. Increasing the amount of traffic localization causes more messages to be injected without increasing the average energy dissipation. This happens because, on average, messages traverse fewer hops when there is greater localization.

For next generation SoCs with a large number of components, adopting router-based communi-

cation architecture can potentially lead to significant energy savings. Figure 1 shows a simple wormhole router modeled in [11]. However, the load on the input port is the total of the wire capacitance and the sum of all input capacitances of N switches. The bit energy will increase linearly with the number of input and output ports N . The power consumption and the design complexity will be very high for switch fabrics with large number of ports. Figure 2 shows the estimates for a SoC router [3]. As technology scales, leakage power becomes increasingly significant if clock frequency is kept invariant for the SoC network.

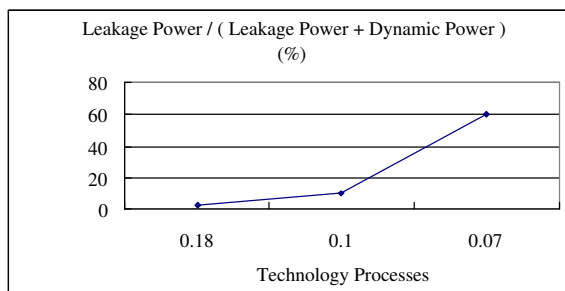


Figure 2: Dynamic and leakage power estimates of a SoC router under variant technology processes (μm)

In order to overcome these problems, we propose a multi-mode switch to save power consumption for specific applications. According to the application characteristics, we can predetermine the operating mode of each switch in the specific NoC during the execution of applications to use "light-weight" lease lines to pass signals or to turn off the switch to save the leakage power.

The rest of the paper is organized as follows. Section 2 summarizes the related work and gives detailed comparisons about NoCs. The predetermined assignment scheme will be detailed described in Section 3. We explain the experimental environment and show the results of our work in Section 4. Finally, we summarize our findings in Section 5.

2 Related Work

Many NoC architectures [1, 2, 5, 6, 7, 8, 9] have been proposed so far. An alternative approach is to use a router based on-chip interconnect architecture. The router-based architecture provides

flexibility in design composition with increasing importance in upcoming platform-based designs composed of intellectual property cores. In crossbar switch fabrics, crossbar is interconnection contention free as every input-output connection has its own dedicated data path. However, the load on the input port is the total of the wire capacitance and the sum of all input capacitances of N switches. The bit energy will increase linearly with the number of input and output ports N . The power consumption will be very high for switch fabrics with large port numbers. Full-connected network is an $N \times N$ fully connected network, which uses MUXes to aggregate every input to the output. Compared with crossbar switch, each bit only consumes energy on one of the MUXes, instead of N switches as in the case of crossbar. However, the N -input MUX has more complicated logic gates, and its power consumption and complexity scale up with the number of inputs N .

3 The proposed method

According to the application characteristics, we design an approach to predetermine the operating mode of switches during the execution of applications.

3.1 Basic Idea

Our basic idea is that if we can predetermine the mode of switches from the normal mode to the lease-line mode or the off mode, it can save the power consumption by passing signals through lease lines or turning off the idle switches. In NoC architectures, there are high frequent operating switches and idle switches during the execution of specific applications. If we can predetermine the modes (lease-mode/off-mode) of high frequent operating switches, it can save significant power consumption.

We can use our simulator or other profiling tools [10] to profile applications and get the communication characteristics first. Then, it will generate a mode table for each switch in the NoC. Finally, the global controller will control the mode of the switches at runtime. Figure 3 shows the concept of the dynamic mode assignment scheme of the NoC. According to the profiling results in the left side of Figure 3, core

C1 always communicates to core C2 at phase 1, and there are no other cores need to access to other modules passing through switches S1 and S2. In this way, we can predetermine the operating mode of switches S1 and S2 from the normal mode to the lease-line mode for the phase 1 period. At the same time, if S3 and S4 are idle during the period we can also turn off these switches.

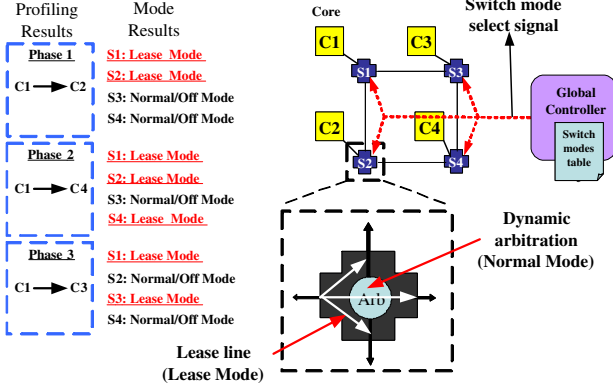


Figure 3: The concept of the multi-mode switching

3.2 The Multi-Mode Switch

We designed three operating modes for the switch, including the *normal mode*, the *lease-line mode*, and the *off mode*, and described as follows:

- *Normal mode*: The normal mode of a switch is that it can be controlled by request events from four-direction paths, and the switch will decide which path can use the bus. In this mode, the switch will arbitrate each request and decide which one can get the occupation permission.
- *Lease-line mode*: The lease-line mode of a switch is that the switch always passes signals from one path to another path by a *lease line* without applying arbitration and crossing the crossbars, shown as Figure 3. In this mode, the switch will not arbitrate requests. When signals arrive the switch will just pass signals through the lease line to save the power of the switch and enhance the transmission performance.
- *Off mode*: When a switch idles for a long time, we can turn off the switch to save the

leakage power.

The key idea is that if there are two cores communicate for a long time without changing the routing state, we can predetermine the mode of the switch from the normal mode to the lease mode. If a switch idles for a long period time, we can set the switch mode to the off mode. We can generate a *switch-mode table* (as shown in Figure 3) for switches by profiling applications first, and then the global controller controls switches during the execution of applications according to the table.

3.3 System Model

To formulate this problem more formally, we define the following terms:

Definition 3.1. The *Communication pattern* of a switch is characterized by the set of all messages, M , passed through the switch. Each message, m , is characterized by its source direction, $SD(m)$, destination direction, $DD(m)$, starting time at which it flit into the switch, $T_s(m)$, and finishing time at which it completely flit out of the switch, $T_f(m)$.

Definition 3.2. Two messages m_1 and m_2 are said to be *potentially colliding* if they overlap in time, given by an overlap relation, O , defined as follows:

$$O = \{(m_1, m_2) \in M \times M \mid ((T_s(m_2) \leq T_s(m_1) \leq T_f(m_2)) \vee (T_s(m_2) \leq T_f(m_1) \leq T_f(m_2)) \vee (T_s(m_1) \leq T_s(m_2) \leq T_f(m_1)) \vee (T_s(m_1) \leq T_f(m_2) \leq T_f(m_1))) \wedge ((SD(m_1) = SD(m_2)) \vee (DD(m_1) = DD(m_2)) \vee (SD(m_1) = DD(m_2)) \vee (DD(m_1) = SD(m_2))))\}.$$

The normal mode of a switch operates when there are several masters requesting the bus at the same time. In this case, the messages transmitted from different masters will potentially collide at the switch. We have to identify the potential colliding set, and control the switch to be the normal mode during these colliding periods.

Definition 3.3. The *normal mode period set*, N , of a switch is the set of all periods with potential collisions, each of which is defined by a 2-tuple representing the start time T_s of the period, and the end time T_f of the period: $N = \{(T_s, T_f) \mid (\exists a \text{ period during } T_s \text{ and } T_f) \wedge (\exists m_1, m_2 \in M, s.t. (m_1, m_2) \in O)\}$

Definition 3.4. The *lease-line mode period set*, L , of a switch is the set of all periods without contention, each of which is defined by a 4-tuple representing the routing source direction s , routing target direction d , the start time T_s of the period, and the end time T_f of the period: $L = \{(s, d, T_s, T_f) \mid (\exists \text{ a period during } T_s \text{ and } T_f) \wedge (\exists m_1 \in M, s.t. s = SD(m_1), d = DD(m_1)) \wedge (\nexists m_2 \in M, s.t. (m_1, m_2) \in O)\}$

Definition 3.5. The *off mode period set*, F , of a switch is the set of all periods without passing messages, each of which is defined by a 2-tuple representing the start time T_s of the period, and the end time T_f of the period: $F = \{(T_s, T_f) \mid \nexists m_1 \in M \text{ during } T_s \text{ and } T_f\}$

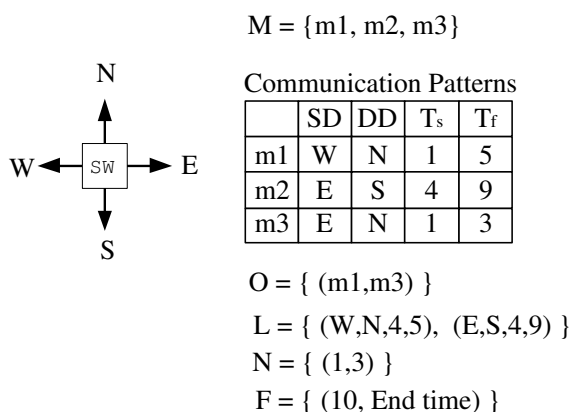


Figure 4: The example of the lease-line mode period set

Our approach has to find out the *lease-line mode period set* and the *off mode period set*. Then the global controller can use the information to control dynamically the switches in the NoC. Figure 4 shows the example of the lease-line mode period set, normal mode period set, and off mode period set. There are three messages $m1$, $m2$, and $m3$, and the communication patterns are shown in the figure. From the communication patterns, we can get the overlap set $O = (m1, m3)$. The lease-line mode period set $L = (W, N, 4, 5), (E, S, 4, 9)$. Our approach has to find out all the sets, and then the global controller can use the information to control dynamically the switches in the NoC at run time.

3.4 Predetermined Mode Table Generation

The mode table generation is shown as Figure 5. Based on the profiling results of applications, we can generate the switch-mode table by using the control flow to schedule mode assignment of each switch. The information of switch-mode table includes the start time and end time of the lease mode, and the routing states. The switch-mode table will be loaded into the global controller before execution of applications, and then control these switch modes according to the switch-mode table. The global controller is not a necessary component, and without global controller will not influence the activities of switches that are defaulted in normal mode. However, if we want to save more power consumption, we can utilize the global controller to achieve the goal.

4 Experimental Results

We implemented an event-driven flit-level interconnection network simulator to evaluate the power and performance of the proposed power-aware switching schemes. The simulator supports k-ary 2-cube network topologies consisting of pipelined virtual-channel routers [4]. We used Orion [11], a power modeling tool, for developing bit energy models for switches. In order to simulate the behaviors of a multi-mode switch, we modified the power model of the switch in Orion. We replace the crossbar and useless buffers with several lease lines (bus lines) between IN ports and OUT ports. We also use array power model to model our mode table in global controller. The control lines are model by the bus-line power model.

In order to evaluate the power savings of our power-aware switching schemes, workload models that accurately characterize real-world communication traffic are required. We take the MPEG-4 decoder as our case studies. In our experiments, we assume two-dimensional 3x3 mesh network with 9 1GHz routers in 0.1 μm process. We simulated 10000 packets generated according to the MPEG-4 traffic characteristics, and the simulation period is 120000 cycles. The profiled core flow graph of application MPEG-4 is shown as Figure 6. The Figure 7 is our evaluated NoC mesh network.

Algorithm: *Mode_Table_Generation (Profiling_file, SMT)*
Objective: *To predeterminedly generate the switch mode table*
Input: *Profiled information, profile_file*
Output: *Switch Mode Table, SMT*

```

Foreach(Switch){
While(Profiling_file != EOF){
  Cycle_pre = Cycle_now;
  Cycle_now = ReadFromFile(Profiling_file);
  IN_Port = ReadFromFile(Profiling_file);
  OUT_Port = ReadFromFile(Profiling_file);
  If(IN_Port == Pre_IN_Port .AND. OUT_Port == Pre_OUT_Port){
    Period = Cycle_now - Cycle_start;
  }else{
    If(Period >= Threshold){
      Cycle_end = Cycle_start + Period;
      Insert Record(Cycle_start, Cycle_end) into SMT;
      Insert Record(IN_Port, OUT_Port) into SMT;
      Period = 0;
      Cycle_start = Cycle_now;
    }
  }
  Pre_IN_Port = IN_Port;
  Pre_OUT_Port = OUT_Port;
} //end of While
} //end of Foreach
End Algorithm

```

Figure 5: The algorithm of mode table generation.

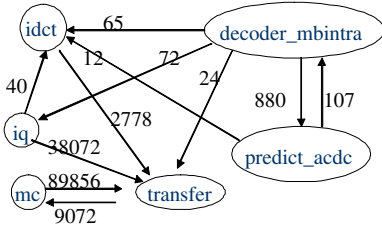


Figure 6: Core flow graph of our MPEG-4 decoder

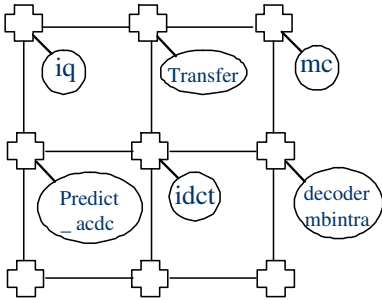


Figure 7: The mesh network of MPEG-4

Figure 8 compares the effectiveness of power consumption of MPEG-4 under different switch modes for different packet injection periods in 0.10 μm process. This figure shows that we can lower down the power consumptions under lease-mode and off-mode switching, and the ratio of the total power saving of the NoC approximates to 40%. Obviously, it saves power consumption if we carefully control the switch mode in the NoC. Figure 9 shows the power savings between variant technology processes. It is obviously that the total power saving of the NoC is approximated to 90% in 0.07 μm process. It is significant because the leakage power is large relative to switching power, and the power can be lower down by turning off idle switches to save the leakage power.

In our design, the global controller and the table space are the important roles of controlling the switch mode. We design a multi-mode switch my Modelsim, and simulate the RTL design by the workloads of MPEG-4. We found that the area of the global controller and table space occupied approximated to 5% of the NoC. The increased power of the multi-mode switch approximates to 4% more than the switch without multi-mode. However, we can save lots power consumption by controlling the mode of each multi-

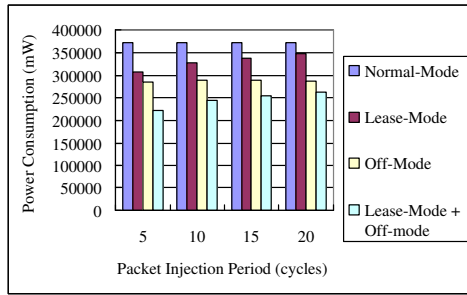


Figure 8: Power consumption under different switch modes

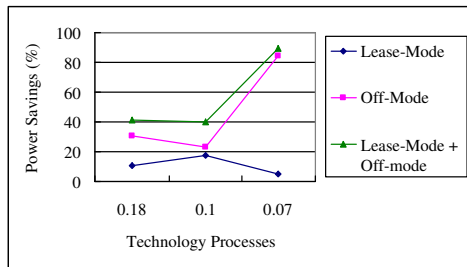


Figure 9: Power savings between different technology processes

mode switch according to the application characteristics.

5 Conclusions and Future Work

In this paper, we proposed a multi-mode switch, which can save significant power if we dynamically control switch modes when the communication characteristics are well known. The experimental results show that the power saving can approximate to 90% in 0.07 μm process. In the future, we will continue to merge our tools, which can automatically instantiate an application-specific NoC, which described in SystemC and automatically completes the whole design flow and evaluations.

References

- [1] Neal K. Bambha and Shuvra S. Bhattacharyya. Joint application mapping/interconnect synthesis techniques for embedded chip-scale multiprocessors. *IEEE Transaction on Parallel and Distributed Systems*, 16(2):99–112, February 2005.
- [2] L. Benini and G. De Micheli. Networks on chips: A new soc paradigm. *IEEE Computer magazine*, pages 70–78, January 2002.
- [3] X. Chen and L.-S. Peh. Leakage power modeling and optimization in interconnection networks. In *ISLPED 2003*, pages 90–95, 2003.
- [4] W.J. Dally. Virtual channel flow control. *IEEE Transactions on Parallel and Distributed Systems*, 3(2):194–205, March 1992.
- [5] W.J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *Proceedings of Design and Automation Conference*, pages 684–689, June 2001.
- [6] J. Hu and R. Marculescu. Energy-aware mapping for tile-based noc architectures under performance constraints. In *Proceedings of Asia and South Pacific Design Automation Conference*, pages 233–23, January 2003.
- [7] J. Hu and R. Marculescu. Exploiting the routing flexibility for energy/performance aware mapping of regular noc architectures. In *Proceedings of DATE Conference*, pages 688–693, March 2003.
- [8] S. Murali and G. De Micheli. Bandwidth constrained mapping of cores onto noc architectures. In *Proceedings of DATE Conference*, pages 896–901, 2004.
- [9] S. Murali and G. De Micheli. Sunmap: A tool for automatic topology selection and generation for nocs. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 914–919, 2004.
- [10] G. Varatkar and R. Marculescu. Traffic analysis for on-chip networks design of multimedia applications. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 795–800, 2002.
- [11] H.S Wang, X. Zhu, L. S. Peh, and S. Malik. Orion: A power-performance simulator for interconnection networks. *Proceedings of the 35th International Symposium on Microarchitecture (MICRO)*, pages 294–305, November 2002.