

Minimising Dynamic Power Consumption in On-Chip Networks

Robert Mullins

Computer Laboratory, University of Cambridge, UK

Robert.Mullins@cl.cam.ac.uk

Abstract—The provision of a general-purpose on-chip communication network is becoming increasingly important in the design of complex SoCs and chip-multiprocessors. In this paper we explore how the power consumed by such on-chip networks may be reduced through the application of clock and signal gating optimisations. We describe how the effectiveness of such optimisations may be maximised and demonstrate that very large reductions in power requirements are possible. A detailed analysis of where power is consumed in an optimised on-chip network is then provided. Results are obtained from fully place and routed standard-cell designs implemented in a 90nm technology. Our 64-bit on-chip network has a maximum operating frequency of 800MHz, in the best-case the combined latency for traversing a network router and link is a single clock cycle.

I. INTRODUCTION

Rising design complexity has encouraged the construction of SoCs through the integration of high-level library IP components. As the number of IP blocks in a SoC increases the focus of system design naturally shifts from computation to communication. This shift is reinforced by technology trends which suggest that communication will increasingly dominate delay and power budgets. The need to manage power consumption through the exploitation of coarser-grained parallel architectures also shifts the focus of design to communication and the orchestration of data-flow in the system. In addition, the need to manage power and thermal budgets, implement system-level fault-tolerance mechanisms and provide flexible multi-use platforms will further boost the utility of a flexible global communication infrastructure. For these reasons, the introduction of on-chip networks and the optimisation of system-wide communication should be viewed as far more than simply the replacement of on-chip buses with a scalable alternative.

Central to the challenge of providing such a network is the minimisation of energy and power overheads. This paper applies a number of existing power optimisation techniques to the design of a low-latency on-chip network. In particular, the application of both local and router-level clock gating are explored in detail. The resulting optimised implementation is analysed to discover where the remaining power is dissipated. A number of additional optimisation techniques are then discussed that could be used to further reduce network power.

II. THE EVALUATION ENVIRONMENT

In order to simplify the analysis and interpretation of results we explore the implementation of an on-chip network in the context of a simple tiled SoC or multiprocessor architecture. The routers employed support virtual-channel

flow control [1] and are able to route data from one tile to another in a single clock cycle [2], [3]. When configured to support 64-bit flits (datapath width), 4 virtual-channels per input and 4 input buffers per virtual-channel, they consume approximately 5% of our system's core die area. Figure 1. illustrates the simple benchmark architecture. The total core area is $64mm^2$, each of the sixteen tiles area is $4mm^2$ and each of the 48, 80-bit (64-bit data + 16-bit control) network links is 1.5mm long. A global clock frequency of 250MHz is assumed unless otherwise stated. Dynamic power results for different frequencies can be computed from the results provided as required.

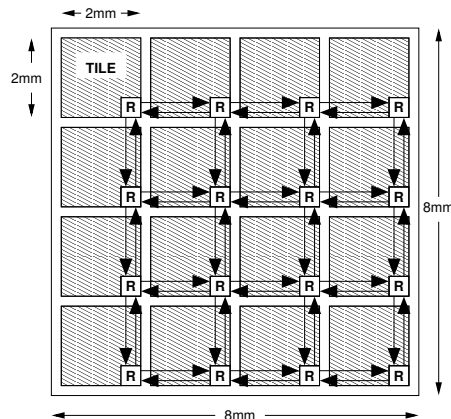


Fig. 1. Networked Tile-based Architecture (our test case)

Results are obtained using a standard ASIC tool flow targeting a high-performance 1.2V 90nm process (nominal V_t).

All power and delay results are obtained after back annotating post place-and-route parasitic data. Delay results are reported at worst-case PVT (SS, 1.08V, 125°C) and power results at nominal PVT (TT, 1.2V, 85°C). Interconnect data is reported assuming inter-router links are placed on the intermediate metal layers, M3-M6.

Delay/Energy trade-offs were explored for the 1.5mm inter-router links for a range of repeater configurations, wire widths and spacings. Magma's QuickCap tool was used to extract a capacitance matrix from five parallel wires placed on Metal3. Metal2 and Metal4 were assumed to contain densely packed wires and were approximated as ground planes. Delay and energy values were then obtained by running HSPICE on a distributed RC model of the interconnect – including all coupling capacitances. Results are summarised in Table I for both in-line and staggered repeater placements [4]. Energy is

reported as the energy per transition, per mm, averaged over 500 random input vectors. For the remainder of this paper we assume minimum spaced wires and in-line repeaters sized to optimise the energy delay product. It was observed that the improvements offered by a staggered repeater layout, although already significant, were tempered by the relatively short inter-router links.

Configuration	Min. Delay		Opt. $E \times D$	
	Delay (FO4)	Energy (pJ/mm)	Delay (FO4)	Energy (pJ/mm)
In-line	6.5	0.34	7.9	0.22
Staggered Repeaters	5.6	0.24	6.1	0.19
In-line, Width x2, Spacing x2	4.6	0.17	5.0	0.13

TABLE I

DELAY/ENERGY TRADE-OFFS FOR DIFFERENT INTERCONNECT CONFIGURATIONS (1.5MM LINK LENGTH)

A. Global Clock Distribution

A global clock tree was generated by placing sixteen token loads across an empty (8mm x 8mm) core area. The loads were kept small to maximise potential power savings in those parts of the tree that could be gated. A clock tree was then synthesised using a standard ASIC CTS (Clock Tree Synthesis) tool. The clock signal was routed on double minimum width interconnect restricted to the upper metal layers. Total clock tree wire length was measured at 42mm. Clock tree shield wires were also routed at a distance of twice the minimum spacing from the clock wire. Clock skew was reported at approx. 50ps. Global clock tree power was measured at 3mW. For larger designs with much tighter skew targets or restricted clock routing, this figure would be expected to rise. In our test case the global component of the clock tree is not of particular concern from a power perspective.

If each tile in the system operates at a different and potentially adaptive clock frequency, the network must be isolated from the tiles using an appropriate FIFO design capable of spanning clock domains at the tile/network interface. If operating each network router (or regions of the network) within its own clock domain is thought to be beneficial, more elaborate local clocking techniques may be employed [5].

B. Low-Latency Routers

Our test case architecture interconnects its tiles or IP blocks using a packet-switched on-chip network. The network routers support virtual-channel flow control in order to improve both throughput and latency. In the best case, the extensive use of speculation within these routers enables network traffic to be routed from one router to another in a single clock cycle [2]. Dimension-ordered (XY) routing is employed with relative addressing. A detailed view of the architecture and results from a 16-node test chip implemented in a 0.18μ technology is provided in [3]. The clock cycle time of this test chip is around 35 FO4 delays. This delay accounts for both the router and inter-router link delay. The maximum clock frequency possible

(for post place-and-route results) at 90nm, using the fastest link repeater configuration, is 800MHz. In this case the router clock period is 32 FO4, including 3 FO4 delays of clock skew and a 4.6 FO4 inter-router link delay.

III. APPLYING POWER OPTIMISATIONS

A. Clock Gating

The global clock in a synchronous circuit is often designed to clock all state-holding components regardless of the need to update a particular stored value. Similarly, signal inputs to a functional unit may change forcing a result to be computed when no result is actually required. This superfluous switching activity and its associated dynamic power consumption may be reduced by gating elements of the clock tree and datapath.

We apply clock gating to the routers in our on-chip network at two levels. Firstly, opportunities within the router to gate portions of the clock tree are identified and exploited. For example, input FIFOs need only be clocked when they are required to store new valid data. Similarly, arbiter state only need be updated when at least one valid request has been served. Secondly, we generate the necessary control signals to allow each router's clock to be gated at a high level. This allows the clock tree associated with a complete router to be isolated from the global network clock.

1) *Local Clock Gating*: The insertion of the necessary clock gating elements to achieve localised clock gating can be performed automatically with modern EDA tools. The majority of tools only require the presence of appropriate load-enable expressions in the source RTL design.

We created such expressions wherever possible in our router design to maximise opportunities for clock gating. These included adding load-enable expressions to the input FIFOs and the matrix arbiters used for virtual-channel and switch arbitration.

2) *Router Level Clock Gating*: Gating the clock as it enters the router ensures no dynamic power is dissipated in the clock tree (or it's associated flip-flops), during periods when the router is inactive. If all input signals are also quiescent, only leakage power is consumed during these idle periods.

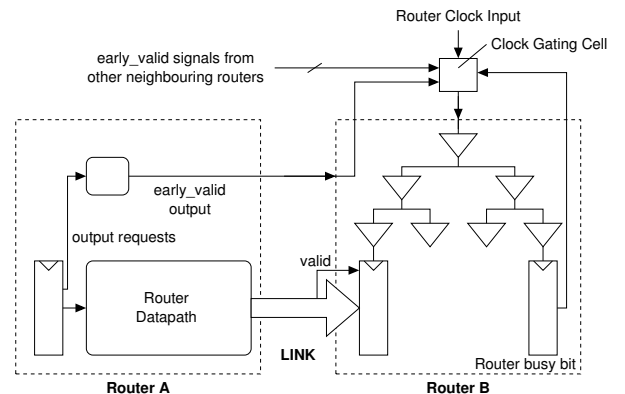


Fig. 2. Generation of a router-level clock enable signal.

The insertion of clock gating cells high in the clock tree reduces the time available to generate a clock enable condition.

The generation of an enable signal is now constrained to be completed in a time of $T_{clk} - T_{insertion}$, where T_{clk} is the clock period and $T_{insertion}$ is the router's clock tree insertion delay. This timing constraint prevents us from generating the enable condition by simply examining the valid signals associated with incoming data.

To overcome this problem we generate an *early-valid* signal in each router for each of its outputs. These signals are generated quickly and simply determine if it is possible that a particular output port will be used. These signals are then communicated to the router at the end of each output channel, serving as an indication of whether new data will be sent in the current clock cycle or not. In contrast to actual network data, these signals arrive early enough in the clock cycle to be used in the generation of a router's clock enable signal. While this approach is valid for generating a clock enable signal, it should be noted that it does slightly overestimate the usage of each output channel. This is because each *early-valid* signal is generated prior to virtual-channel or switch scheduling or the application of aborts (due to misspeculation).

To determine if it is possible to gate a particular router's clock, we examine its incoming *early-valid* signals together with a state bit generated by the router in the previous clock cycle (the *router busy* bit). This bit indicates if the router has any buffered packet data or blocked output virtual channels. The router must be clocked if any of its output virtual channels are blocked to ensure that the channel-level flow control signals that will unblock each channel are observed. The 'stop/go' protocol used in our router requires that the router is only clocked until each channel is unblocked. If a credit-based protocol was used, care would need to be taken to ensure all credits were received. Isolation of these flow-control state machines on their own branch of the clock tree could be used to reduce power further. The complete clock gating scheme is illustrated in Figure 2.

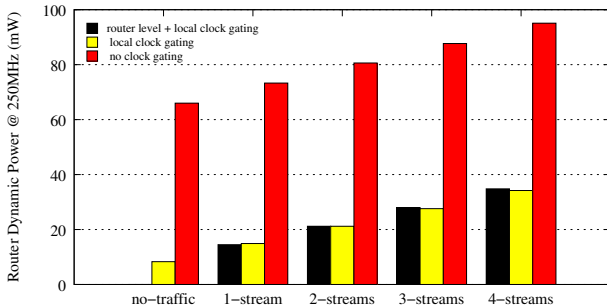


Fig. 3. Impact of clock gating on router power dissipation (stream traffic)

3) *Power Savings from Clock Gating*: The impact of clock gating on dynamic power dissipation was explored by investigating the power consumed under two simple traffic models. All results were collected and analysed for the router at position (2,2) in the mesh. The first experiments streamed a large number of short packets through the router. A continuous stream of packets was injected on the east port and routed to the west, similar streams were then added south to north, west to east and north to south. This experiment

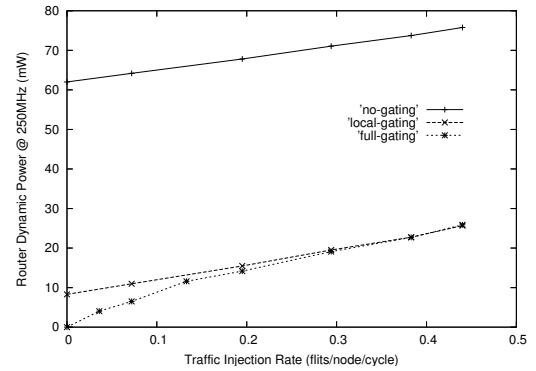


Fig. 4. Impact of clock gating on router power dissipation (random traffic)

provided both an estimate of the lowest (no streams) and worst-case (4-streams) power dissipation in the router. In the second set of experiments uniform random traffic was injected at various rates. Packets in both cases were 4-flits long and carried a random data payload. Link power is not considered in these clock gating results as it is unaffected by clock gating. Simulation results were recorded for a period of 1K clock cycles (this provided results within 0.5% of those for 10K cycles for comparisons at low and high injection rates). Results reported for clock gating account for dynamic power only, static or leakage power was approximately 3mW in all cases.

In all the experiments the impact of introducing clock gating is considerable. These savings are perhaps exaggerated a little due to the use of standard-cell flip-flops to construct the input FIFOs. While in many cases a fully synthesizable design is desirable, custom SRAM-based FIFO implementations would offer some performance and power improvements. In the stream based traffic experiments (see Figure 3) the impact of full router-level clock gating is only apparent when there is no traffic, as this is the only point at which the router can be gated at this level. Employing router-level clock gating at this point reduces dynamic power to zero from 8.3 and 66mW for the locally-gated and ungated routers respectively. The diminishing impact of router-level clock gating, as traffic levels increase, is also evident when power consumption under uniform random traffic is evaluated (see Figure 4). The impact of router-level clock gating is more pronounced when the traffic patterns are bursty in nature. Further experiments examining real traffic, where such behaviour is common, are ongoing.

B. Signal Gating and Gate-Level Optimisations

Automated signal gating and gate-level optimisations were explored with the EDA tools available, initially with little impact on power consumption. Manual insertion of signal gating logic was then pursued. Significant power savings were achieved by gating the data input to each of the virtual-channel input FIFOs, this prevents the data input to each of the FIFO's flip-flops toggling when a write is not required. In the worst-case (4-stream) this reduced total

Component	Worst-Case	Low-Random
<i>Input Logic/Input FIFOs</i>	21%	9%
<i>VC Muxes/Crossbar</i>	8%	5%
<i>Scheduling/Arbitration</i>	6%	12%
<i>Other</i>	9%	14%
<i>Local Clock Tree</i>	17%	25%
Router Total (Dynamic)	63%	65%
Router Total (Static)	8%	22%
Inter-Router Link	28%	11%
Global Clock Tree (div. 16)	< 0.5%	1%
Total Power	38.4mW	15.2mW
(Power without optimisations)	86.9mW	66.9mW

TABLE II

BREAKDOWN OF NETWORK ROUTER POWER CONSUMPTION (250MHZ)

power consumption by around 7mW.

Our final optimised router design (excluding link power) had a worst-case (4-stream traffic model) power consumption of 27.5mW, including 3.1mW dissipated as static power.

IV. ANALYSIS OF POWER CONSUMPTION

In this section we examine the proportion of power consumed in different components of the network. The router used to gather these results exploits all the optimisations described previously. As previously, we assume packets carry random data payloads, this results in an inter-router link switching activity factor of approximately 0.4 when switching activity in the associated control signals is considered.

Table II shows the breakdown of power consumption for the major components of the network. Results are reported for two traffic scenarios: our assumed *worst-case* traffic pattern of four continuous data streams and under uniform random traffic with approximately 2 flits injected into the network per cycle (we call this the *low-random* case). With this injection rate the router under test routes on average 0.44 flits/cycle. The sum of the power dissipated by the italicised entries is equal to the router's total dynamic power.

In the router design both the input FIFOs and scheduling/arbitration logic contain a significant number of flip-flops. As a result, the local clock tree power is associated primarily with these components. Further power reductions in both these areas are feasible and are under investigation. These include adopting a more compact format to store our precomputed router schedule. At present this speculative schedule requires $2 \times P^2 \times V$ bits, where P is the number of router ports and V is the number of virtual channels. Of course, it should be remembered that our single cycle router already benefits from the lack of pipelining registers in the datapath.

The synthesis of the routers analysed is restricted from performing optimisations across module boundaries. This allows the contribution of each module to the total power consumption to be easily determined. If this restriction is removed, worst-case router power consumption under the 4-stream traffic model is reduced by 3.5mW (12%). The area of the final router was $150K\mu m^2$. Clock skew within this router was reported as 55ps.

V. FURTHER OPTIMISATIONS AND RELATED WORK

The power consumption of on-chip networks has been explored previously in [6] and through the creation of analytical power models in [7]. Network power results have also been reported for the RAW microprocessor which exploits both statically and dynamically routed on-chip networks [8].

Further power optimisations will require link and static power to be targeted. While our results show that these components do not dominate power consumption at present, scaling trends suggest that the fraction of power required by these components will grow. Fortunately, low-power interconnects suitable for on-chip networks have already been explored extensively, including [9]. Techniques to reduce the often significant idle power of such links have also been investigated in [10]. A router's static power may be reduced using a range of design and runtime techniques [11], including router-level power gating.

VI. CONCLUSIONS

This paper has demonstrated that clock and signal gating techniques may be successfully applied to on-chip network router designs. The power savings are significant and should clearly be employed before evaluating other power-saving techniques. A breakdown of power consumption in a optimised on-chip network router has also been presented highlighting the most promising opportunities for future savings.

ACKNOWLEDGEMENTS

This research is supported by the Engineering and Physical Sciences Research Council (EPSRC) (Grant EP/D036895/1)

REFERENCES

- [1] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2003.
- [2] R. D. Mullins, A. F. West, and S. W. Moore, "Low-latency virtual-channel routers for on-chip networks," in *Proc. of the 31st Annual Intl. Symp. on Computer Architecture (ISCA)*, 2004, pp. 188–197.
- [3] —, "The design and implementation of a low-latency on-chip network," in *Proceedings of the 11th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2006.
- [4] A. B. Kahng, S. Muddu, E. Sarto, and R. Sharma, "Interconnect tuning strategies for high-performance ICs," in *Conference on Design, Automation and Test in Europe (DATE)*, 1998, pp. 471–478.
- [5] R. D. Mullins, "Asynchronous versus synchronous design techniques for NoCs," Tutorial at the Intl. Symp. on System-on-Chip, 2005.
- [6] P. T. Wolkotte, *et al.*, "Energy model of networks-on-chip and a bus," in *Proc. of the Intl. Symp. on System-on-Chip (SoC 2005)*. IEEE, Nov. 2005, pp. 82–85.
- [7] H.-S. Wang, L.-S. Peh, and S. Malik, "Power-Driven Design of Router Microarchitectures in On-Chip Networks," in *Proceedings of the 36th MICRO*, Nov. 2003.
- [8] J. S. Kim, M. B. Taylor, J. Miller, and D. Wentzloff, "Energy characterization of a tiled architecture processor with on-chip networks," in *Proc. of the Intl. Symp. on Low power Electronics and Design (ISLPED'03)*, 2003.
- [9] R. Ho, K. Mai, and M. Horowitz, "Efficient on-chip global interconnects," in *Symposium on VLSI circuits*, 2003.
- [10] V. Soteriou and L.-S. Peh, "Design-space exploration for power-aware on/off interconnection networks," in *Proc. of the 22nd Intl. Conf. on Computer Desing (ICCD)*, 2004.
- [11] A. Agarwal *et al.*, "Leakage power analysis and reduction for nanoscale circuits," *IEEE Micro*, vol. 26, pp. 68–80, Mar. 2006.