NoCEE : Energy Macro-Model Extraction Methodology for Network on Chip Routers

Jeremy Chan[†], Sri Parameswaran[†‡]
†School of Computer Science and Engineering, The University of New South Wales, Australia
‡National ICT Australia (NICTA), Australia
{jeremyc, sridevan}@cse.unsw.edu.au

ABSTRACT

In this paper we present NoCEE, a fast and accurate method for extracting energy models for packet-switched Network on Chip (NoC) routers. Linear regression is used to model the relationship between events occurring in the NoC and energy consumption. The resulting models are cycle accurate and can be applied to different technology libraries. We verify the individual router estimation models with many different synthetically generated traffic patterns and data inputs. Characterization of a small library takes about two hours. The mean absolute energy estimation error of the resultant models is 5% (10% max) against a complete gate level simulation. We also apply this method to a number of complete NoCs with inputs extracted from synthetic application traces and compare our estimated results to the gate level power simulations (mean absolute error is 5%). Our estimation methodology has been integrated with commercial logic synthesis flow and power estimation tools (Synopsys Design Compiler and PrimePower), allowing application across different designs. The extracted models show the different trends across various parameterizations of Network on Chip routers and have been integrated into an architecture exploration framework.

1. INTRODUCTION

Networks on chip has arisen as a solution to the poor wire scaling and increasing complexity of large System on Chip (SoC) design. NoCs aim to replace long shared bus wires with scalable switched networks with higher performance, predictable wiring and better interconnect properties. These networks can be made latency insensitive, simplifying the design of complex systems since communication and computation problems can be treated separately.

Although NoCs provide many benefits, they add additional logic complexity to the communication architecture. Power consumption in the communication architecture is a major bottleneck in current design [17] and hence, energy-aware optimizations for NoCs is of primary importance. Several optimization methods have been proposed to reduce energy consumption through application specific customizations. These include: customized router buffer sizing[15]; custom topology generation [25]; adaptive routing [16]; and mapping processing elements to tiles [14, 21].

Motivation for this work

Various NoC architectures have been proposed, each with a differing implementation and hence, varying energy consumption characteristics. In order to explore the design space for energy minimization, a method for estimating energy is needed. Typically, a design's energy consumption is evaluated using a register transfer level (RTL) or gate level power simulator. Power simulations are essential for finding implementation bottlenecks but their lengthy run-times make them unsuitable for early design space exploration (days for a typical trace). For this reason, techniques are needed to quickly extract a fast and accurate estimation model. Estimation models also provide useful insights, especially towards architecture parameterization trends, and identification of potential energy reduction areas.

While energy estimation for a whole system is of paramount importance, the energy of specific components allows the system to be laid out to reduce hot spots in the final design, thus improving reliability of the whole design. Cycle accurate power estimation allows peaks of the power spectrum to be more accurately determined, so that the design can be re-engineered to extend battery life, as battery life is heavily influenced by power peaks. Therefore, this work is motivated by not only having to estimate energy of a router, but being able to rapidly estimate the consumption of each of the major components, and also being able to estimate energy in a cycle accurate manner. Finally, in an era where technologies are rapidly changing, it is important to be able to characterize a router built upon different technologies rapidly.

In this paper we present NoCEE[1], a methodology for obtaining a fast and accurate energy macro-model for a synthesizable packet switched NoC router. The NoC router is decomposed into constituent router components and a model is built for each component. Linear regression is used to obtain a good fit between the component's events and observed cycle energy obtained from a gate-level simulator. The combination of individual component models is used to predict the cycle energy consumption of a complete router. A single traffic pattern consisting of various traffic loads and random distributions is used to stimulate the NoC router during characterization. We apply our methodology to several 3, 4 and 5 port wormhole mesh routers with uniform FIFO depths at various levels, and validate that good correlation exists between the model and gate level power simulations. The models of several routers are used together to predict the energy consumption of a complete NoC. The predicted energy consumption is compared against several complete NoC gate level power simulations to evaluate its accuracy.

The rest of the paper is organized as follows: Section 2 surveys the related work and states our contribution. Section 3 gives some background and theory. The NoC model is described in Section 4 and Section 5 presents the experimental results. Section 6 concludes the paper.

2. RELATED WORK

Early work [22] in power estimation attributed a fixed energy to a structural block. These pattern independent models were inaccurate as they did not account for input statistics. Thus, models were formulated to capture the changes in power dissipation when inputs are stimulated [19]. Many pattern dependent models use linear regression to obtain a good fit between the model parameters and energy consumption. Regression-based macro-modeling at the RTL has been extensively applied to various combinational and sequential circuits [2, 13, 20], as well as application specific instruction set processors [7, 12]. However, this technique has yet to be applied to packet switched routers. *For the first time, we apply linear regression-based macro-modeling for energy model extraction of an NoC packet switched router.*

NoC power estimation has been addressed by several recent research works. In [29], Ye et. al. proposed an energy estimation flow to derive bit energy models that are used to evaluate different switch fabrics (e.g fully-connected, Banyan, Batcher-Banyan) in network routers. Their methodology derives the energy cost for transmitting a bit in the network router from ingress to egress ports but ignores clock power and leakage power. In [26], Wang et. al. created a network simulator to estimate dynamic power of router components from CACTI scaling equations[23]. This simulator was augmented in [6] to support leakage power. The same authors have used this simulator for architectural exploration to explore energy savings through segmented crossbars [28], and to trade off of router complexity and energy [27]. Banerjee et.

National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

[1] NoC Energy Estimation: pronounced gnocchi

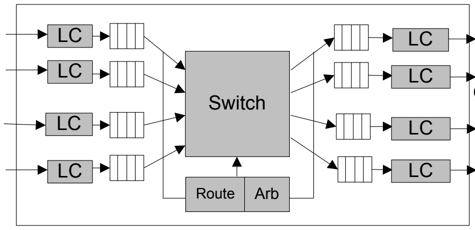


Figure 1: NoC router model

al. in [1] addressed leakage power modeling in mesh routers and derived a power-state machine for a wormhole router based on SPICE net-lists. The power-state machine allowed power to be estimated on a cycle accurate basis. However, both [26] and [1] contain models that are tightly coupled with circuit implementations. As such, these models cannot be migrated to different technology libraries without a large amount of re-characterization.

The authors in [4] automate the extraction of a power model for the STBus, a high performance communication architecture supporting shared buses as well as crossbars. They use regression based techniques to obtain a relationship between average power and the architectural parameterizations of the STBus. A packet switched router contains additional components that are not present in the STBus.

Our characterization differs from the models in [1, 26, 29] by being more adaptable to changing technologies and differing router types. We use a semi-automatic system containing linear regression to characterize each router, while the models in [1, 26, 29] have to be manually extracted.

While the work in [4] automates the extraction of a power model for an STBus using linear regression, their work does not address packet switched networks, nor cycle accurate models.

The novel contributions of our work are:

1. the ability to derive accurate and system-level cycle-accurate estimation models for power consumption of synthesizable NoC routers;
2. a methodology to automate the extraction of the power estimation for an NoC router library based on regression analysis between control signals and cycle-based power; and
3. that we demonstrate the use of this methodology to predict NoC power at the router level, NoC level and within a fast estimation framework.

3. BACKGROUND AND THEORY

3.1 The NoC Architecture and NoC Router

In NoCs, longer interconnect wires are broken down into shorter channels with routers or relay stations forming a micro-pipeline between processing elements. Each router is buffered to allow it to be latency insensitive and the communication between the routers uses a predefined protocol and flow control. Although many NoC router variations exist, the major component subsystems remain the same. These are: (i) *link controllers*; (ii) *crossbars switches*; (iii) *routing* and (iv) *arbitration units* (see Figure 1).

As NoC routers have a well-defined structure, the subsystems can be modularized with similar interfaces for different implementations. For example, two routing algorithms, X-Y and west-first routing can be designed to have the same inputs and outputs (address input and destination port). Standardizing the interfaces between subsystems allows the variations to be created and reused in different architectural parameterizations.

3.2 CMOS Power Consumption and Estimation

There are two broad categories of power consumption: (i) dynamic power and (ii) static power. Dynamic power is dissipated when the circuit is active caused by the charging and discharging of the internal and load capacitance of a gate. Static power is dissipated when a circuit is inactive, consisting mostly of source to drain sub-threshold leakage current where the gate does not turn off completely.

Commercial gate-level power simulators use look up tables based on input net transition delay and output net capacitance to determine energy dissipation. Dynamic switching power is determined by summing the switching energy with the number of transitions. Obtaining the number of transitions requires a full gate level simulation, or designer knowledge to estimate the switching probabilities of nets in the design. Leakage power in these tools is estimated by determining via table lookup given the inputs on the gates.

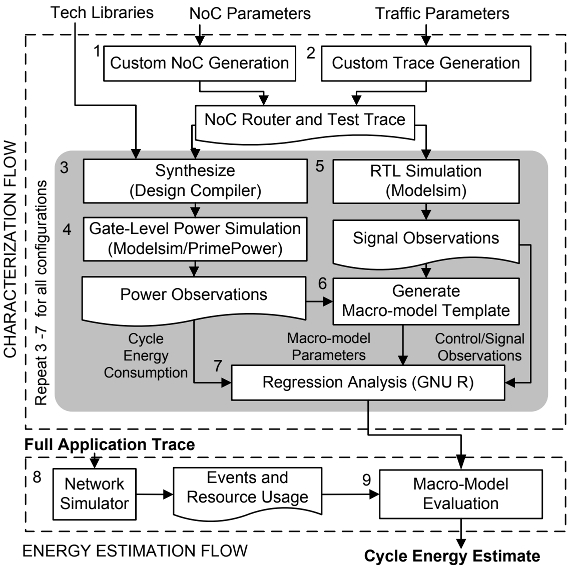


Figure 2: Energy model extraction methodology

4. METHODOLOGY

Much of this work is inspired by existing work on regression-based macro-modeling used to characterize smaller circuits at the RTL. We refer the interested reader to [3] for an overview of macro-modeling techniques. In this section, we first describe the procedure to construct an NoC energy macro-model. Next, we describe how model variables are chosen and construct macro-models for each of the major NoC subsystems in a wormhole NoC router.

4.1 Overview

There are two major parts in our energy estimation flow: (i) characterization and (ii) energy estimation (see Figure 2). The characterization step builds a macro-model given a set of technology libraries, NoC parameters and traffic patterns (Steps 1 - 7). The technology libraries and NoC parameters are used to create the synthesizable NoC while the traffic patterns are used to simulate the router under various conditions. The second part, energy estimation, predicts the cycle energy of an NoC router from the macro-models and the event timings from a high level simulation (Steps 8 and 9).

As NoCs have a well-defined structure, macro-models can be built for each parametrization of the major subsystems and reused. We characterize each of the subsystems separately by extracting their models while they are operating within a complete NoC router. This allows multiple subsystems to be characterized together in a single simulation while capturing the frequently occurring behaviors of individual subsystems. A range of possible router configurations are created in order to build models for various subsystem parameterizations (Step 1). We sweep through a range of possible parameters and create a number of NoC router implementations using the NoC generator[5].

Macro-model Inputs and Traffic Generation

To obtain a good model, a single traffic pattern must be generated that exercises the circuit under a wide range of possible conditions (Step 2). For an NoC router, it is possible to control these traffic patterns by varying the packet injection and acceptance rates, packet destination ports and switching activity on the data bus (changing the hamming distance between successive flits). Varying the injection rates and acceptance rates will synthetically exercise the router under different levels of contention. Meanwhile, adjusting the switching activity can be used to reveal the NoC router circuit's dependency on the input data. A calibration trace is created that includes a mixture of traffic patterns.

The power and signal values for every cycle are profiled via two separate simulations. The NoC configuration is synthesized and the cycle power is obtained through the gate-level power estimation flow (Steps 3 and 4). An RTL simulation is performed using the same calibration simulation to capture the signal waveforms on important control and data signals (Step 5). The signal waveforms are compared against the power simulation to see which signals may affect power dissipation. Events relating to control and data signal values are chosen for inclusion in the macro-model based on their importance to the energy behavior (Step 6).

Using Regression Analysis to Obtain the Macro-model

The energy consumption and signal values at each cycle are used as observations in the regression analysis. Regression analysis is performed between the cycle energy and the macro-model signal observation (Step 7). Statistical parameters such as the R^2 and p-values (probability values) from the regression output are used as a metric of the goodness of fit, and importance of variables [11]. The model parameters are modified and, if necessary Steps 5 to 7 are repeated until a good fit is found. Steps 3 to 7 are repeated for each of the configurations until the entire library is characterized.

Energy Estimation Using the Macro-model

Once a macro-model is built, it can be used to predict cycle power. Network statistics such as number of packets sent and bit switches can be used to quickly evaluate the total energy by substitution into the model. Multi-dimensional interpolation is used to obtain an estimate of the model coefficients, if the lie between two characterized points. A cycle accurate system-level network simulator is required to reproduce cycle-power at the system level. If a network simulator is able to produce a trace of the events (Step 8), a power waveform can be produced by substituting the timing of these events into the macro-model (Step 9).

4.2 Selection of Macro-model Variables

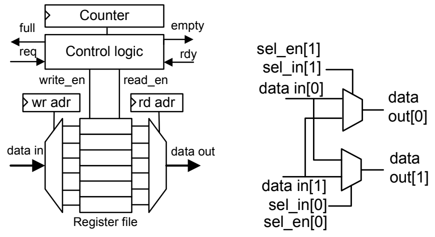
A macro-model is built by choosing variables that have a strong relationship to energy consumption. In this context, variables X_i are events that occur in the NoC router. Two types of events are considered for inclusion: control events, and data events. Control events occur when a control signal triggers energy consumption; these are: one signal transitions (0 to 1, 1 to 0), and two, the value of a particular signal being either 0 or 1 in a particular cycle. In some cases, a control event may affect energy in a future cycle (for example pipelined operations). Control events can be time-shifted such that a vector of events match the energy consumption. In NoC routers, energy is dominated by several major components such as FIFO buffers and multiplexors. The control signals of these major components have a strong influence on the total energy consumption.

A larger amount of activity on the data inputs also contributes to higher energy consumption. Hence, the hamming distance of the data inputs is also considered. Selecting inputs from these two domains results in the following expression for the energy of a component.

$$E_{component} = \beta_{residual} + \sum_{i=1}^{n} \beta_{event_i}\ \ X_i$$

where n is the number of events, $\beta_{residual}$ is the energy that is independent from the model variables, and β_{event_i} is the regression coefficient for the data or control event X_i. X_i takes the value 1 for control events when a signal is present or 0, when it is not present. In cases where X_i is a data variable such as hamming distance, it is represented using a integer. In the characterization of our router, we only use hamming distance of the combination of all the input ports together. In routers design or situations where individual router ports consume more energy, more data variables may be needed.

Model variables should be selected to be independent whenever possible. The decision to include or exclude a model variable can be aided by the use of statistical parameters such as the p-values. The process of model selection can be automated by using a fixed criterion using a threshold value for removal or selection of parameters with model variables iteratively removed. Clever algorithms such as branch and bound can be used to avoid fitting all combinations of variables [11]. In NoC routers, there are very few signals that have a large effect on energy consumption. The router designers are often aware of these signals as they are often similar to the signals used for debugging the circuit. Hence, these signals can be selected without much effort.



(a) Link Controller (b) Crossbar

Figure 3: Router Components

Evaluation of Macro-model Energy Consumption

The cycle energy consumption of a component $E_{component}$ can be expressed as:

$$E_{component}(t) = \beta_0 + \sum_{i=1}^{n} \beta_i X_i(t) + P_{leakage}\ \ t_{clk}$$

where β_i is the fitting energy coefficients for model parameter X_i, and β_0 is cycle energy not related to any of the predictor variables, and $P_{leakage}$ is the average component leakage power, and t_{clk} is the cycle period. Thus, given clock frequency f_{clk}, cycle power can be calculated as follows:

$$P_{component}(t) = (\beta_0 + \sum_{i=1}^{n} \beta_i X_i(t))\ \ f_{clk} + P_{leakage}$$

Leakage power and Clock power

The two major sources of independent power are transistor leakage power and clock-related power. For the purposes of system-level modeling, we assume that the variance of average leakage power inputs is small. We observed this phenomena experimentally, as leakage power for individual components are nearly invariant in the power simulations for the same configuration. We use average leakage power figures available in the power simulation report as the architecture's fixed leakage power.

Clock-related power includes dissipation in the clock network i.e. buffers, register clock gates and clock inputs to flip-flops. All these exhibit a static power dissipation because a similar number of transitions occur on every clock cycle. In the following models, the uncorrelated energy composed of the clock and leakage energies will be labeled as β_{x0} where x is the component name.

4.3 NoC Router Model

We develop energy macro-models for a complete router using the procedure described in the Section 4.1. For the sake of brevity, we only consider the model for an input-queued router; the output link controller is considered as pass-through wires that consume no energy. Although significant, energy consumed between the routers is not modeled as layout information was not available. The capacitance of wires can be found using logic synthesis tools once the length of wires is known and applied to the macro-model. For each of the major router components a macro-model is derived; each can be summed to obtain the total router energy consumption as follows:

$$E_{router} = E_{ilc} + E_{xbar} + E_{arb} + E_{route} + E_{other}$$

where E_{ilc}, E_{xbar}, E_{arb}, E_{route} and E_{other} is the energy values derived from the component macro-models for the input link controller, crossbar, arbiter and routing unit and the miscellaneous glue logic respectively.

In the following sections, we apply this method to the NoCGEN router libraries[5] but this method should be applicable to other NoC libraries such as Xpipes[8] and Proteo[24].

Link controller

We consider the energy macro-model for a single input-queued link controller (ILC). The significant contributors to the power consumption of the link controller are FIFO memory elements (implemented as

RAM or register files) and their control logic (Figure 3(a)). The empty and full signals are determined based on an internal counter. In the target routers, there are two FIFOs in each link controller port ($fifo0$ and $fifo1$) due to separate data and address buses. Hence, the power consumption of the link controller can be modeled as:

$$E_{ilc} = E_{fifo0} + E_{fifo1} + E_{count} + E_{ilc\_other}$$

where E_{fifo0} and E_{fifo1} are the FIFO energy consumptions, E_{count} is the counter energy consumption and E_{ilc_other} is energy of the less significant components in the link controller.

When a flow control digit (flit) arrives at the link controller, it is written to the storage element. Similarly, when a flit is acknowledged by the next router, the next read address is incremented and the outputs are updated. Energy can be attributed to the reading and writing of the memory elements, hence the *read_en* and *write_en* control signals are good candidates for inclusion. When contention occurs, the counters begin incrementing, resulting in a small increase in energy consumption. Hence, the link controller model can be formulated as:

$$E_{ilc} = \beta_{ilc0} + \beta_{ilc1}E_{dwrite} + \beta_{ilc2}E_{dread} + \beta_{ilc3}E_{awrite} + \beta_{ilc4}E_{aread} + \beta_{ilc5}E_{count\_en}$$

where E_{dread} and E_{dwrite} are the read and write control signals for the data FIFO ($fifo0$); E_{aread} and E_{awrite} are the read and write control signals for the address FIFO ($fifo1$); and E_{count_en} is the counter enable control signal.

Crossbar Switch

A two-port crossbar is depicted in Figure 3(b). In this crossbar implementation, an enable control input *sel_en* disables the output when a port is unused. Through experimentation, it was found that the two major contributing factors to energy consumption are data bus switching and the initial selection of the crossbar port when the first flit arrives. The initial selection of the crossbar port is mirrored by the $0 \rightarrow 1$ transition of the *sel_en* signal. The crossbar traversal energy is:

$$E_{xbar} = \beta_{xbar0} + \beta_{xbar1}E_{dist} + \beta_{xbar2}E_{sel_{0\rightarrow 1}}$$

where E_{dist} is the input hamming distance of the inputs of the crossbar and $E_{sel_{0\rightarrow 1}}$ is the selection energy when a packet first enters the crossbar.

Arbitration

When an arbitration decision is made, the selected port is stored in a register that is used to control the crossbar port selectors. In a simple priority arbiter, the energy can be decomposed into storage and computation. We selected a signal that occurred in the same cycle as computation and included it in the macro-model.

The arbitration energy can be estimated by:

$$E_{arb} = \beta_{arb0} + \beta_{arb1}E_{sel} + \beta_{arb2}E_{sel_{t\ 1}}$$

where E_{sel} energy consumption during the cycle that arbitration occurs and $E_{sel_{t\ 1}}$ is the energy related to control input from the previous cycle. $E_{sel_{t\ 1}}$ is an example of the same signal time-shifted to reflect energy dissipated due to the signal's value in a previous cycle.

Route Unit

The route units for both the X-Y routing algorithm and street sign routing algorithm are purely combinational and are determined based on the address presented by the input link controller. Based on this observation, some energy will be dissipated initially when the head flit of a packet E_{head} is passed through the routing unit and a smaller amount in following flits E_{flit}. The model is as follows:

$$E_{route} = \beta_{route0} + \beta_{route1}E_{head} + \beta_{route2}E_{flit}$$

Other

Several other miscellaneous glue logic circuits consume a small amount of total energy in the circuit. As the dynamic energy changes are small, we do not characterize these individual components. Instead, the average energy $E_{average}$ is included as part of the additional power uncorrelated power that is added to the model.

5. EXPERIMENTAL RESULTS

All experiments were conducted on a dual Opteron 224 with 2 GB of RAM. The routers were generated from the NoCGEN[5] library and generation tool (Step 1). Two types of routers were used in experiments: custom routers (capable of a variable number of ports) and mesh routers. All routers were configured with 32-bit data buses and 16-bit address buses. FIFO buffers were implemented using two-port clock-gated register files with counter logic determining FIFO full and empty signals. The routers were characterized at an operating frequency of 125 MHz, 250 MHz and 500 MHz. At these three frequencies, the model coefficients were similar with clock-cycle leakage energy scaled. The results shown here are for 250 MHz.

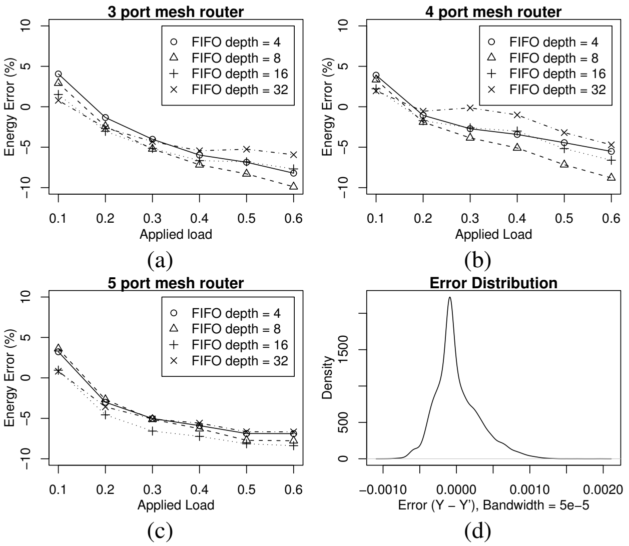
A C++ program was written to generate a trace files for the VHDL simulation given a random distribution (Poisson, Gaussian etc) for packet arrival times and packet length (Figure 2, Step 2). This trace file contains send packet, idle and receive packet commands that are use to control the traffic in and out of the router ports. Each send packet is annotated with the source and destination addresses to enable control of the destination and the data bus switching activity.

All router designs were synthesized using Synopsys Design Compiler and a 90 nm TSMC standard cell library (tcbn90gtc) with automatic clock gating enabled. Modelsim was used for all the VHDL simulations. As layout information is unavailable until after place and route, we assign an estimated load of 50 fF to the router outputs. Synopsys PrimePower was used to obtain the power estimates and the power waveform. A program was written in C++ to combine the outputs of PrimePower and Modelsim into a time-series comma separated file. The statistical package GNU R was used to perform regression analysis. Perl scripts were used to link all the tools together.

5.1 Experiments Conducted

Three separate experiments were performed to ensure the validity of our models: at the single router level; within a complete NoC (multiple routers connected together in a mesh arrangement); and in a fast power estimation environment. First, we characterize the NoC library for two library parameters: number of ports and FIFO depth using the method described in Section 4. We evaluated the model prediction error against several gate-level power simulations with a separate set of randomly generated traffic patterns (72 traces in total) with varying injected loads. Second, we investigate port scaling energy characteristics of different components in the router. Next we create and synthesize various complete NoCs and stimulate these using applications traces. The errors between the model and gate simulation experienced when running these applications on complete NoCs are also provided.

5.2 Validation



(a) (b)

(c) (d)

Figure 4: 3 port, 4 port and 5 port fitting errors, cycle errors

We characterized 3, 4 and 5 port mesh routers with uniform FIFO depths of 4, 8, 16 and 32. The resultant model coefficients (labeled

Comp.	Variables	Description	Energy Coeffcient (fJ)											
			3,4[a]	3,8	3,16	3,32	4,4	4,8	4,16	4,32	5,4	5,8	5,16	5,32
ILC	E_{awrite}	adr fifo write	566	849	1382	2362	561	842	1379	2371	602	881	1403	2391
	E_{aread}	adr fifo read	168	218	188	168	113	124	135	116	120	160	148	135
	E_{dwrite}	data fifo write	1237	1682	1799	1931	1218	1694	1826	1965	1273	1768	1900	2071
	E_{dread}	data fifo read	361	691	652	785	355	669	648	799	399	746	720	883
	E_{count_en}	counter en	218	319	494	839	224	314	463	757	209	298	429	672
	$constant$	residual	400	533	671	1195	400	534	673	1200	404	539	679	1209
XBAR0	E_{dist}	hamming dist	31	49	35	37	31	50	35	37	32	49	37	39
	$E_{sel_{0\rightarrow1}}$	head flit	714	534	874	934	862	639	994	1051	732	459	839	934
	$constant$	residual	23	73	42	40	38	106	65	62	54	149	97	94
XBAR1	E_{dist}	hamming dist	117	99	101	92	187	146	170	139	174	131	160	140
	$E_{sel_{0\rightarrow1}}$	head flit	147	153	55	164	216	250	56	255	345	454	218	438
	$constant$	residual	34	63	59	68	104	190	177	205	99	172	185	205
ARB	E_{sel}	arb	147	153	55	164	216	250	56	255	345	454	218	438
	$E_{sel_{t\ 1}}$	arb delayed	115	113	105	115	167	196	182	201	199	244	203	231
	$constant$	residual	47	58	55	57	55	68	62	67	58	65	65	69
Route	E_{route}	route	1	5	15	14	40	96	119	124	82	191	231	241
	$constant$	residual	1	1	1	1	7	6	7	7	13	13	12	12
Other	E_{other}	average misc.	147	153	55	164	216	250	56	255	345	454	218	438

[a] Ports = 3, FIFO depth = 4

Table 1: Coefficients for various configurations ports, depth

by {ports, FIFO depth}) are shown in Table 1. Characterization took about five to ten minutes per configuration and only a couple of hours for the complete set of configuration points. Note, that characterization only needs to be done once for each sub-component. For example, the input link controller circuit for a single port may be identical in 3 port and 4 port routers, hence, the model for a 3 port router can be used in the 4 port router model. Table 1 shows that the coefficients for the 3 port and 4 port routers exhibit similar coefficients.

Once the routers were characterized, we used these macro-models to predict the energy consumption for a set of traffic patterns that were different from the calibration set. These patterns were randomly generated with fixed applied load. Each router port trace consists of 500 packets sent to random destination ports with a given fixed applied load varying from 0.1 to 0.6 flits per port with each packet containing five flits. Figures 4(a - c) show the errors across different traffic loads for the 3 port, 4 port and 5 port routers respectively. From these figures the maximum absolute energy estimation error is 9.9% and the average absolute error is 4.6%. Given the application statistics such as number of packets, data switching rate and time in contention, it takes less than a second to determine the energy consumption.

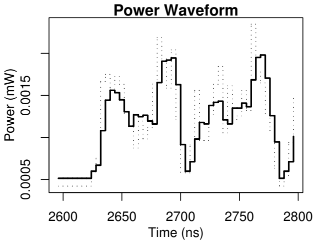


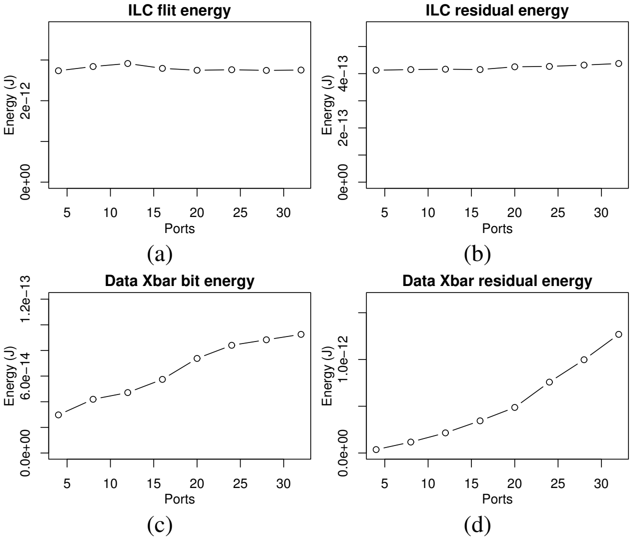
Figure 5: Predicted and Measured Power Waveforms

Figures 4(a - c) shows that the model tends to slightly under-predict power when there is a high applied load. This is caused by the fact that our model captures most but not all dynamic effects when there is a large amount of port contention. For cycle-based prediction, the absolute cycle mean relative error is less than 20% across the validation set. A proportion of the power waveform from the validation trace of the 4 port mesh-router with FIFO depth of 4 with 0.2 flits per port applied load is shown in Figure 5(a). The PrimePower waveform is the dotted line and the predicted is the unbroken line. Figure 4(d) shows the distribution of the cycle based energy estimation errors (measured energy - predicted energy) for the same trace as in Figure 5.

5.3 Custom Router and Router Port Scaling

An exploration of the energy coefficients trends was undertaken for several larger non-mesh routers. We parameterized the NoC generator to create a custom topology router with destination port embedded in the address with a varying number of ports from 4 to 32 and a uniform 4-flit FIFO buffer. Each router was synthesized using the same methods as in previous experiments and models were extracted.

Figure 6 shows the trends in the link controller and crossbar switch.



(a) (b)

(c) (d)

Figure 6: The relationship between the number of ports vs. energy

It can be seen from Figure 6(a) that in this custom router library the increase in ports does not greatly affect the link controller buffering energy for a single flit ($\beta_{aread} + \beta_{awrite} + \beta_{dread} + \beta_{dwrite}$). Figure 6(b) shows that for a single link controller, residual energy β_{ilc0} remains nearly constant. For a complete router, there is one link controller for every input port, hence the residual energy will increase linearly with the number of ports.

Figure 6(c) shows the increase in the crossbar hamming energy E_{dist} with the increase in the number of ports. The observed relationship is linear and can be explained by increased logic depth in larger crossbars. Meanwhile, the crossbar residual energy (Figure 6(d)) increases quadratically with the increase in the number of ports. This is due to the quadratic increase in area as the number of inputs ports and output ports increase.

The macro-models from the previous section can be used to rapidly predict the energy consumption of an application running on any supported network topology (2 2, 4 2 or 4 4 etc). We integrate the energy macro-model into a graph-based NoC performance estimation framework to predict the energy for a complete NoC. This framework estimates the performance and power given the application as a task graph, the NoC architecture as a topology graph, and energy model as specified in Section 4 (see Figure 7).

Task graphs model the application dependencies in a graph with three types of nodes: communication, computation and handshake nodes. Graph edges define the dependencies between the different types of nodes. The task graph format is similar to the Communication Analysis Graphs presented in [18]. Task graphs are obtained from various sources including synthetically generated task graphs and realistic hand-crafted task graphs [9]. We estimate the energy consumption us-

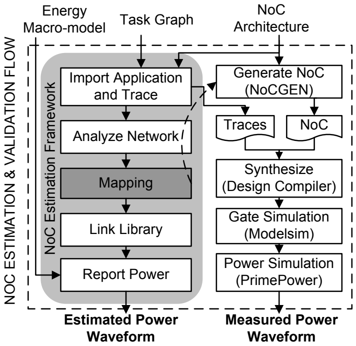


Figure 7: Full NoC Validation Flow

ing an assortment of task graphs from each of these three sources.

The synthetic task graphs were created using TGFF 3.1 [10] with arbitrary packet sizes and task lengths. These were created to give larger examples. TGFF is configured to generate a task graph containing computation tasks, communication dependencies, communication volume and processing element execution times. The task execution time and communication properties are summarized in Table 2. Each task is randomly assigned to a processing element. Communication between tasks that have been assigned to the same processor is omitted from the analysis as this is not transmitted over the network.

Benchmark	Trace Param		Task Param		Comm. Param	
	Tasks	PE	cycles	+/-	flits	+/-
TGFF1	500	8	3000	1500	32	16
TGFF2	2000	8	10000	5000	256	128
TGFF3	1500	8	1600	80	64	22
TGFF4	500	16	1200	600	64	32

Table 2: TGFF Benchmark Parameters

Two benchmarks, *auto-industry* and *telecom*, were collected from the E3S benchmark suite [9]. These benchmarks contain 24 and 30 tasks respectively. These benchmark consist of a task graph containing computation tasks (nodes), communication volumes (edge property), communication dependencies, and a library of processing elements that can perform the computation task (node properties). Each task is manually assigned to a processor based on the library of processing elements. This determines the length of a computation task in clock cycles.

To estimate performance and energy, a method similar to that used in [18] was used to estimate the timing of network events. Each router port is reserved when requested and where there is contention, waiting times are computed. We modify their algorithm to support arbitrary network topologies, routing algorithms and NoC protocol timings. The event durations are annotated in the graph and event timing was back annotated into the task graph. Once the graph is completely annotated, a breadth-first search longest path algorithm is performed to compute the starting times and duration of each node. An event list with starting and ending times was used to substitute into the energy macro-model.

Benchmark	Source	Conf.	Estimated Energy	Measured Energy	Relative Error %
auto-indust	E3S	2x2	1861nJ	1740nJ	6.95
telecom	E3S	3x3	6610nJ	6244nJ	5.86
TGFF1	TGFF	4x2	5919nJ	6346nJ	-6.72
TGFF2	TGFF	4x2	90.345uJ	88.237uJ	2.38
TGFF3	TGFF	4x2	10564nJ	10700nJ	-1.27
TGFF4	TGFF	4x4	4601nJ	4364nJ	5.43

Table 3: Prediction Errors for Complete NoC

Table 3 shows a summary of the different simulations and their respective predicted energy and measured energy. The fast power estimation technique produced a power waveform and energy estimate in under a second while the gate-level simulation took several minutes to hours to estimate the total energy and power waveforms, e.g. 30 minutes for the TGFF4 simulation. Table 4 summarizes the execution times for the full gate-level simulation and the predicted simulation. The energy estimation errors from the extracted models are well below the estimation errors of commercial estimation tools (which is typically around 20% abs. error with gate-level simulations) and the simulation time is often three to four orders of magnitude faster than a complete gate simulation. As such, we conclude that our models are accurate enough for the purpose of high-level design space exploration for NoC router architectures.

Benchmark	Graph-based CPU Time	PrimePower CPU Time	Speed Up
auto-industry	< 0.001 s	495 s	> 100000
telecom	0.001 s	1504 s	> 100000
TGFF1	0.03 s	1498 s	50000
TGFF2	0.43 s	7.58 h	63000
TGFF3	0.15 s	2951 s	20000
TGFF4	0.05 s	1800 s	36000

Table 4: Efficiency of Complete NoC simulation

6. CONCLUSIONS

We have presented a technique for extracting a fast and accurate energy macro-model from a Network on Chip library. Using this technique, we have characterized a set of routers generated using the NoCGEN framework. Validation between the macro-model and gate level estimates show that for the randomly generated test patterns, the energy estimation error had a mean absolute error of less than 5%. We are currently applying these models for use in an architecture synthesis environment.

References

[1] N. Banerjee, P. Vellanki, and K. S. Chatha. A power and performance model for network-on-chip architectures. In *DATE*, pages 1250–1255, 2004.

[2] L. Benini, A. Boliolo, M.Favalli, and G. D. Michelli. Regression models for behavioural power estimation. In *Workshop on Power and Timing Modeling, Optimization and Simulation*, pages 179–187, 1996.

[3] A. Bogliolo, L. Benini, and G. D. Micheli. Regression-based rtl power modelling. *ACM Transaction on Design Automation of Electronic Systems*, 2000.

[4] A. Bona, V. Zaccaria, and R. Zafalon. System level power modeling and simulation of high-end industrial network-on-chip. In *DATE*, 2004.

[5] J. Chan and S. Parameswaran. NoCGEN: A Template Based Reuse Methodology for Networks on Chip Architecture. In *VLSI Design*, 2004.

[6] X. Chen and L.-S. Peh. Leakage power modeling and optimization in interconnection networks. In *ISPLED*, 2003.

[7] N. Cheung, S. Parameswaran, and J. Henkel. A Quantitative Study and Estimation Models for Extensible Instructions in Embedded Processors. In *ICCAD*, 2004.

[8] M. Dall'Osso, G. Biccari, L. Giovannini, D. Bertozzi, and L. Benini. xpipes: a Latency Insensitive Parameterized Network-on-chip Architecture For Multi-Processor SoCs. In *ICCD*, 2003.

[9] R. Dick. Embedded system synthesis benchmark suite (E3S). http://www.ece.northwestern.edu/dickrp/˜e3s/.

[10] R. Dick, D. Rhodes, and W. Wolf. TGFF: task graphs for free. In *CODES/CASHE*, pages 97–101, Washington, DC, USA, 1998. IEEE Computer Society.

[11] J. J. Faraway. *Linear Models with R*. CRC Press, 2004.

[12] Y. Fei, S. Ravi, A. Raghunathan, and N. Jha. A hybrid energy-estimation technique for extensible processors. *IEEE Transactions on CAD*, 2004.

[13] C.-T. Hsieh, Q. Wu, C.-S. Ding, and M. Pedram. Statistical sampling and regression analysis for rt-level power evaluation. In *ICCAD*, pages 583–588, Washington, DC, USA, 1996. IEEE Computer Society.

[14] J. Hu and R. Marculescu. Energy-aware mapping for tile-based noc architectures under performance constraints'. In *ASPDAC*, 2003.

[15] J. Hu and R. Marculescu. Application-specific buffer space allocation for networks-on-chip router design. In *ICCAD*, 2004.

[16] J. Hu and R. Marculescu. DyAD - Smart Routing for Networks-on-Chip. In *DAC*, 2004.

[17] K. Lahiri and A. Raghunathan. Power analysis of system-level on-chip communication architectures. In *CODES+ISSS*, 2004.

[18] K. Lahiri, A. Raghunathan, and S. Dey. Fast Performance Analysis of Bus-Based System-on-Chip Communication Architectures. In *ICCAD*, 1999.

[19] P. E. Landman and J. M. Rabaey. Architectural power analysis: the dual bit type method. *IEEE Trans. VLSI Systems*, 3(2):173–187, 1995.

[20] H. Mehta, R. M. Owens, and M. J. Irwin. Energy characterization based on clustering. In *DAC*, pages 702–707, New York, NY, USA, 1996. ACM Press.

[21] S. Murali and G. D. Micheli. SUNMAP: a tool for automatic topology selection and generation for NoCs. In *DAC*, pages 914–919, 2004.

[22] S. Powell and P. Chau. Estimating power dissipation of VLSI signal processing chips: the PFA technique. *VLSI Signal Processing IV*.

[23] P. Shivakumar and N. P. Jouppi. Cacti 3.0: An integrated cache timing, power, and area model. Technical report, Compaq: Western Research Laboratory, 2001.

[24] D. A. Siguenza-Tortosa and J. Nurmi. VHDL-based simulation environment for Proteo NoC. In *HLDVT*, 2002.

[25] K. Srinivasan, K. S. Chatha, and G. Konjevod. Linear Programming based Techniques for Synthesis of Network-on-Chip Architectures. In *ICCD*, 2004.

[26] H. Wang, L.-S. Peh, and S. Malik. Orion: A power-performance simulator for interconnection networks. In *International Symposium on Microarchitecture*, 2005.

[27] H. Wang, L.-S. Peh, and S. Malik. A technology-aware and energy-oriented topology exploration for on-chip networks. In *DATE*, 2005.

[28] H.-S. Wang, L.-S. Peh, and S. Malik. Power-Driven Design of Router Microarchitectures in On-Chip Networks. In *MICRO*, 2003.

[29] T. T. Ye, L. Benini, and G. D. Micheli. Analysis of power consumption of switch fabrics in network routers. In *DAC*, 2002.