2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)

Heterogeneity Aware Power Abstraction for Hierarchical Power Analysis

Arun Joseph, Spandana Rachamalla, Shashidhar Reddy, Nagu Dhanwada
IBM Systems
arujosep@in.ibm.com

Abstract*— In modern day FinFET-based microprocessors, dynamic power consumes ~80% of chip power under high utilization conditions. These microprocessors cater to a broad range of workloads, each with its own unique power signature. In addition to the heterogeneity across the workloads, there is notable heterogeneity in power consumption profile across the chip, including within IP blocks. With increasing focus on reducing the time to market, new methods for efficient generation of accurate power abstracts of IP blocks, while capturing the heterogenous signatures, are key for enabling rapid hierarchical chip power analysis. We present approaches to generate per clock gating domain parameterized power abstracts and evaluate it on real industry class high performance microprocessor designs. Experimental results demonstrate the accuracy and efficiency of the proposed approach when compared with direct gate level simulation-based power estimation and existing IP power abstraction techniques.

***Keywords*—Power models, low power, power analysis, clock gating, power abstraction**

I. Introduction

In modern day FinFET-based high performance microprocessors, dynamic power is the significant contributor to overall chip power consumption [1, 16, 17]. These microprocessors cater to a broad range of workloads, each with its own switching activity characteristics and profile, which thereby drives dynamic power and thermal profile of the chip. In addition to heterogeneity across this spectrum of workloads, there is notable heterogeneity in the power consumption profile across the chip, and within the IP blocks that make up the chip. Among all the techniques that are used for reducing dynamic power, clock gating is most widely adopted. Clock gating enables gating of unused resources without any instructions per cycle (IPC) loss. A major part of clock power is dissipated close to the leaf nodes of the clock tree that drive latch banks. Gating the clock at the last few levels of the clock buffers is therefore an effective way to reduce dynamic power consumption. Without clock gating, when an IP block is clocked, the sequential elements within it receive the clock, regardless of whether they will toggle in the next cycle. With clock gating, the clock signals are combined with explicitly predefined enabling signals which results in sequential elements to shut down, providing savings in dynamic power. Clock gating is used across several levels of design abstraction: system architecture, block design, logic design and gate level. Several methods that take advantage of clock gating are described in [2, 3, 4, 5, 6].

Full chip power analysis is a must have step in the design of next generation power efficient microprocessor chips. Hierarchical accumulation of gate level power simulation data for each of the IP block instances though accurate, is extremely cumbersome to manage and do rapid analysis on. With chip design and system schedules getting more and more compressed, there is an increasing demand for bottoms-up power abstraction-based methodologies.

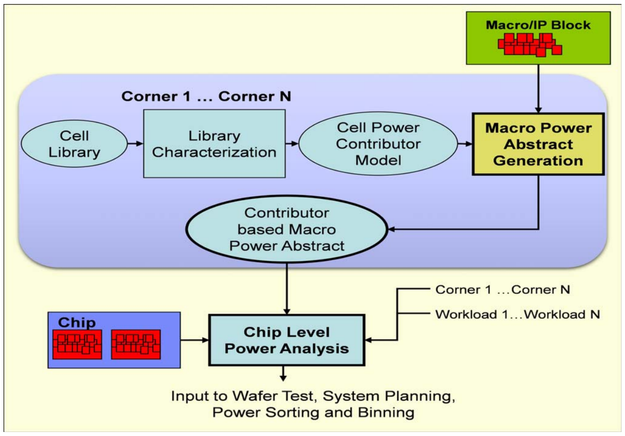

Macro/IP Block
Corner 1 ... Corner N
Cell Library
Library Characterization
Cell Power Contributor Model
Macro Power Abstract Generation
Contributor based Macro Power Abstract
Corner 1 ...Corner N
Workload 1...Workload N
Chip
Chip Level Power Analysis
Input to Wafer Test, System Planning, Power Sorting and Binning

Fig. 1. Contributor based power analysis

Such flows allow for rapid hierarchical power analysis across a range of workloads. One such flow, based on techniques described in [7, 8] is shown in Figure 1.

Workload sensitivity and aggressively clock gated designs pose accuracy challenges to the problem of generating generalized dynamic power abstracts like [8], as discussed later in the experimental results section. Drastic power abstraction techniques like [9] though useful for identification and early characterization of digital power proxies, are too high-level for enabling the continuous power efficient design of the chip during the pre-silicon design phase.

In this paper we build on the concepts introduced in [10] and present a dynamic power analysis flow which efficiently captures the heterogeneity induced by workloads. Even in comparison with [10], this is the first paper which provides comprehensive analysis backed by an actual implementation in an industry setting and its application on the next generation high performance microprocessor design. We believe this enables broader adoption of the proposed techniques. This work is further motivated by the following aspects of next generation microprocessor designs. (a) The IP blocks are becoming larger with increasing number of clock gating domains. (b) Much more aggressive solutions are being adopted for improving the clock gating of these very large IP blocks. (c) There are several workloads for which significant heterogeneity in both clock and data activity is seen across the multiple clock gating domains within an IP block.

We describe approaches to generate clock gating domain parameterized dynamic power abstractions. The first approach is a quick simulation-less approach that enables per domain parameterization of the clock power component. The next approach parameterizes both the clock and data power components, while additionally considering the complex interaction between the gating domains during data power abstraction. One of the key

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00099

novelties of the dynamic power abstraction solution is that we accomplish the domain separation through an efficient marking and tracing process, rather than doing a gate level simulation of one domain at a time. This attribution of switching due to each domain, without doing a domain by domain gate level simulation, makes the approach very efficient and scalable for IP blocks with very large number of clock gating domains. In comparison with [8], our work notably improves the data power accuracy for workloads with high data switching which is a critical aspect of next generation high performance microprocessor designs. Additionally, the abstractions presented in the paper enable real industry use-cases like improving idle clock gating, determining suspicious clock gating scenarios and providing finer grained (per clock gating domain) feedback to optimize the logic design for power. Leakage power abstraction techniques proposed in [8] were found to be accurate even for future designs and hence are not discussed in the context of this work.

In section II we discuss related work in the area. In section III we describe the clock gating domain parameterized power contributor model. In section IV, we describe approaches for generating the domain parameterized abstractions. Finally, in section V we present experimental results comparing the three approaches in terms of accuracy and efficiency, by comparing with detailed gate level power simulation on several IP blocks of varying sizes from an industry leading microprocessor design.

II. Related Work

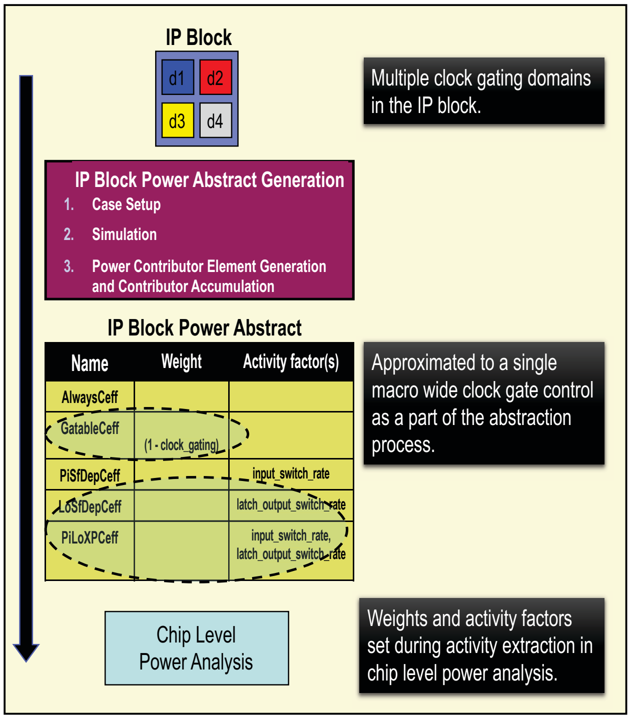
Existing power modeling standards in the industry do not support efficient modeling of large IP blocks [7, 11]. Prior work discusses generation of macro models for combinational data path blocks, soft macros parameterized on the size of the block based on techniques like regression analysis, hamming distance between input vectors, etc. [12, 13, 14, 15]. The intent of these models is to directly plug them into an RTL simulation, and generate power analysis results. The focus of most of the existing work on power modeling is on representing the power consumed. The notion of having a portable PVT independent contributor based models was presented in [8]. But here, during power abstraction the individual clock gating domains are approximated into a single control parameter. As mentioned before, there are several scenarios where this approximation is a limitation. In contrast, the work presented in this paper attempts to remove limitations in [8] by parameterizing the clock and data power abstracts on a per clock gating domain basis, while doing it in an efficient and scalable manner.

III. Power Abstract Definition

In the following sub-sections, we provide an overview of the base power abstraction introduced in [8] and the proposed multi clock gating domain power abstraction.

A. Base power abstract

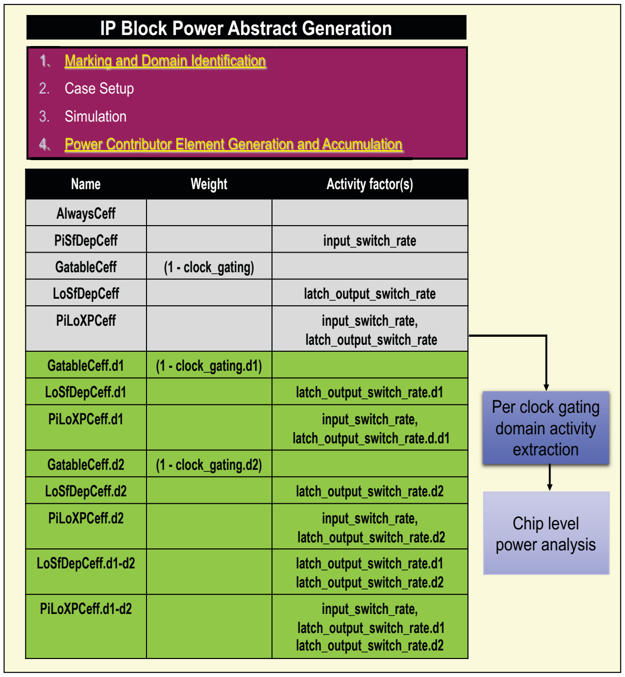
The dynamic power abstraction introduced in [8] is performed in terms of the dynamic power contributors. It characterizes power as a function of a clock_gating weight factor, input_switch_rate and latch_output_switch_rate activity factors. As shown in Figure 2, the Ceff, weight and activity factors (computed during higher level power analysis) are computed by approximating across the clock gating domains to compute power. AlwaysCeff component models the base power when clock gating is true and activity factors are zero. GatableCeff models the additional power when clock gating is turned off. PiSfDepCeff and LoSfDepCeff model the separate power adders due to primary input and latch output switching respectively. PiLoXPCeff models the synergistic delta in power due to combined switching of primary inputs and latch outputs.


IP Block
d1
d2
d3
d4
Multiple clock gating domains in the IP block.
IP Block Power Abstract Generation
1. Case Setup
2. Simulation
3. Power Contributor Element Generation and Contributor Accumulation
IP Block Power Abstract
Approximated to a single macro wide clock gate control as a part of the abstraction process.
Chip Level Power Analysis
Weights and activity factors set during activity extraction in chip level power analysis.

Name	Weight	Activity factor(s)
AlwaysCeff		
GatableCeff	(1 - clock_gating)	
PiSfDepCeff		input_switch_rate
LoSfDepCeff		latch_output_switch_rate
PiLoXPCeff		input_switch_rate, latch_output_switch_rate

Fig. 2. Single clock gate control base power abstract

B. Proposed multi-domain power abstract


IP Block Power Abstract Generation
1. Marking and Domain Identification
2. Case Setup
3. Simulation
4. Power Contributor Element Generation and Accumulation
Per clock gating domain activity extraction
Chip level power analysis

Name	Weight	Activity factor(s)
AlwaysCeff		
PiSfDepCeff		input_switch_rate
GatableCeff	(1 - clock_gating)	
LoSfDepCeff		latch_output_switch_rate
PiLoXPCeff		input_switch_rate, latch_output_switch_rate
GatableCeff.d1	(1 - clock_gating.d1)	
LoSfDepCeff.d1		latch_output_switch_rate.d1
PiLoXPCeff.d1		input_switch_rate, latch_output_switch_rate.d.d1
GatableCeff.d2	(1 - clock_gating.d2)	
LoSfDepCeff.d2		latch_output_switch_rate.d2
PiLoXPCeff.d2		input_switch_rate, latch_output_switch_rate.d2
LoSfDepCeff.d1-d2		latch_output_switch_rate.d1 latch_output_switch_rate.d2
PiLoXPCeff.d1-d2		input_switch_rate, latch_output_switch_rate.d1 latch_output_switch_rate.d2

Fig. 3. Multiple clock gate domain power abstract

Figure 3 shows the power abstraction parameterized on a per clock gating domain basis. In the proposed abstraction, there are clock and data Ceff components that correspond to the individual clock gating domains. The per domain Ceff, along with weight and activity factors (computed on a per clock gating domain basis during higher level analysis) are used for hierarchical per clock gating domain power analysis. This makes it more accurate and usable to drive more aggressive use of clock gating in the logic design process. The changes to the algorithms used in the abstract generation process are primarily the per clock gating domain marking and identification step and the per clock gating domain contributor ac element accumulation and generation step. The details and variants of per clock gating domain clock and data power abstract generation algorithms are described in the next section.

IV. Abstract Generation Approaches

The IP block dynamic power abstraction described in [8] computes the different Ceff values by performing a set of random gate level logic simulations under particular settings of control points in the design. The primary inputs, latch outputs, embedded IP outputs are set to be in a specific state, before performing a random logic simulation, to derive the switching factors at all points in the design. These along with the physical capacitances of the nets in the design are used to create a Ceff on per simulation case basis and each such simulation case Ceff is accumulated to create the base dynamic power abstract [8]. This parameterized power abstract can be used in a chip level power rollup flow, where the power abstract is evaluated under the activity and PVT conditions for the chip. Steps required for per gating domain parameterization are described in the rest of the section.

Figure 4 shows the overall clock gating domain parameterized abstract generation process which has three different approaches. The first one is a simple and quick no simulation based approach for per clock gating domain clock power abstraction. The next one is a gate level simulation based approach where both data and clock power abstraction is performed on a per clock gating domain basis. This approach also accounts for the interaction between the gating domains during data power abstraction. An extension to the second approach is also proposed, where a collapsing technique is used to handle larger designs with a very high number of clock gating domains.

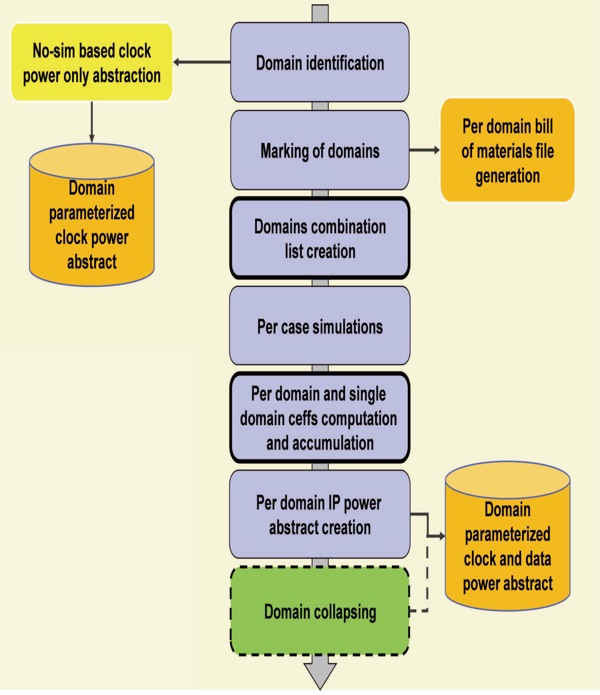

No-sim based clock power only abstraction
Domain parameterized clock power abstract
Domain identification
Marking of domains
Per domain bill of materials file generation
Domains combination list creation
Per case simulations
Per domain and single domain ceffs computation and accumulation
Per domain IP power abstract creation
Domain parameterized clock and data power abstract
Domain collapsing

Fig. 4. Domain parameterized abstract generation

A. *Approach 1: Quick no-sim based abstraction*

In this first approach, we perform a quick tracing of the netlist to determine the GateableCeff (clock power component) on a per clock gating domain basis. Here the data power related capacitive components in the abstraction remain the same as before [8], i.e. they are not broken down on per gating domain basis. This approach is a no simulation based approach and hence very quick. This is also useful for enabling logic power takedown and assessing the power impact of clock gating decisions taken by the logic designer. The algorithm for no-simulation based clock power abstraction is described below.

```
// Create clock gating domain list(D) for an IP block
Domain List = 0
Foreach Instance I in the IP block netlist
{
        If I is an LCB (local clock buffer)
        {
          Find net (N) driving clock gate enable
          Add N to Domain List
       }
}
// Compute effective capacitance (CeffN) for each clock
gating domain
For each domain N in Domain List
{
   CeffN = 0
   Foreach LCB L driven by N
   {
    Foreach functional output clock pin (CO) of L
         {
          CeffN += get_pin_capacitance (CO)
         // from power rule
         LI_CLK_NET = clock_net_feeding_latch_clock_input
          CeffN += get_wire_capacitance (LI_CLK_NET)
          For latch L driven by this clock net
          {
           CI = latch_input_clock_pin(L)
           CeffN += capacitance(CI)
           // from power rule
          }
         }
   }
   Create the ac contributor element for the clock gating
domain N with CeffN
}
```

B. *Approach 2: Clock and data power abstraction based on domain combination lists*

This is a gate level simulation based approach where both data and clock power abstraction is performed on a per clock gating domain basis. The interaction between the gating domains is also accounted for during data power abstraction. Additionally, a technique for combining domains in order to manage the size of the abstract for designs with high number of gating domains is also discussed.

a) Domain identification: The domain identification step comprises of finding the unique nets that drive the control pin of local clock buffers (LCB) that control the latch banks. The Local Clock Buffers (LCB) are circuit blocks which implement built-in clock adjustment controls to fine-tune the waveforms of the local clocks to the clocking elements and also perform local functional clock gating [2, 3]. These circuits have interface pins for local clock-gating along with overrides for disabling the local clock gating. There are also controls for scan testing or to bypass any clock-gating-associated problems seen in early hardware. These circuits also may be also used to control the activity of the local clocks on a cycle by-cycle basis during test along with other control signals for modifying the local clock waveforms.

b) Domain marking: The domain marking process consists of traversing the IP block netlist to mark domain information on each box and net in the design. This consists of tracing back from each clock gate (LCB control pin input) through inverters and buffers to find unique clock gate controls. Following which we forward propagate separate net marks (e.g. bit flags) from the clock output nets of LCBs controlled by each of the identified clock gate controls. The LCB outputs controlled by the same clock gate would get the same mark. This traversal would go through the remaining clock tree, into the clock pin of the latches, through combinational logic and stop at the data inputs to the next latch. The domain marking function handles more complex scenarios like: logic driven by multiple domains and feedback logic.

c) Domain combination list creation: Each box and net have a domains list associated with it, which is the list of domains that influence that box and net respectively. For the box or net which is not driven by any domain, this list will be empty. The step of domain combination list creation is done to abstract the power attributed to logic driven by multiple domains. This step traverses the individual domain lists from each box and net and combine together to create a master domains combination list. A key novelty in this approach is that we achieve separation of each domain's switching characteristics through the tracing and domain marking process, thereby precluding the need for separate gate level simulation per domain. Hence the next step of per case simulations, remains the same as what was done for the single clock gate control parameterized abstraction. This process was described in section III. The result of per case gate level simulations is an effective capacitance per case, which is used in per domain and single domain Ceff computation and accumulation step.

d) Per domain and single domain Ceffs computation: This step calculates per case Ceffs for each unique entry in the domains combination list, for the creation of per domain contributor based power abstracts. These per domain AC elements will be parameterized based on the domain or domains that drive that capacitance. This will map to the entry in the domains combination list for which per clock gating domain accumulation was performed. The LosfDepCeff element in the domain based abstraction will be parameterized on the Latch Output Switch Rate (LOSR) event of the domains that influence this element. During chip level power analysis, LOSR will be computed from the workload driven switching activities and the maximum LOSR among all the domains that influence a common logic, will be used for computing the LosfDepCeff power. For example, if the domain combination list contains domains D1 and D2, then the activity events on which the element would be parameterized would be {D1.LOSR, D2.LOSR}. Then the MAX {D1.LOSR, D2.LOSR} will be computed during chip power analysis and will be used for computing the power modelled by this Ceff.

```
// Collapsing of the per domain abstraction
For each losfdepceff AC Element A in per domain AC element
list
{
       If (capacitance(A) <= threshold * LOSFDEPCEFF)
       {
         Collapsed Element Capacitance += Capacitance(A)
       }
       Else
       {
         Add element A to final abstraction
       }
     Add collapsed Element to final abstraction
     Write out the final per domain abstraction
  }
```

e) Domain collapsing for handling large extensively gated designs: For the case where there are designs with several clock gating domains, we also enable the use of a collapsing process to merge the set of latch output switch factor dependent AC elements from the abstraction in order to manage the size of the generated abstract. The set of AC elements from the previous step is generated as a sorted list of capacitance values in descending order. We use a criterion based on the percentage contribution of a given AC element to the total latch output dependent switching capacitance (LosfDepCeff) to identify further candidates to merge. The above simple procedure shows the collapse process we employ for larger designs that are extensively gated.

V. Experimental results

We evaluated the accuracy and efficiency of the proposed power abstraction on IP (logic blocks) picked from the core and uncore of a next generation high performance microprocessor design. The gate level netlist for the logic block is simulated to generate per clock gating domain power abstract. Switching pattern files are generated from RTL simulations (for several thousand cycles) from higher level realistic workloads to capture the workload dependency on power. From switching patterns, switching activity factors like clock gating factor (CG), primary inputs switching factor (PISF) and latch output switch rate (LOSR) are computed on a per clock gating domain basis. The computed activity factors are applied to the generated power abstract to calculate abstract based power. The abstract based power is compared to the reference power model, which is the gate level simulation-based power, where the switching at every node is computed to produce an average power dissipated for that node and then accumulated across all the nodes in the design. Figure 5 shows the experimental setup, where we use the same workload to get the reference workload driven gate level simulation-based power numbers and compare it with power from the evaluation of the power abstract.

The designs of varying degrees of complexity - in terms of the size of the design as well as the amount of clock gating present in the designs - are used as representatives of next generation high performance microprocessor designs [16, 17]. In this paper, we show the results of the proposed approaches on five such unique design case studies. Similar experiments were conducted across all the macros from both core and uncore units of the microprocessor. Experiments were run on 24-core 2.6GHz Xeon machine, running RHEL 5, with 256GB memory. We used variants of a thermal design point workload to study the accuracy of the proposed approaches. In the rest of the section we refer to "TAT Benefit" as the improvement seen in runtime while using the abstraction-based power computation when

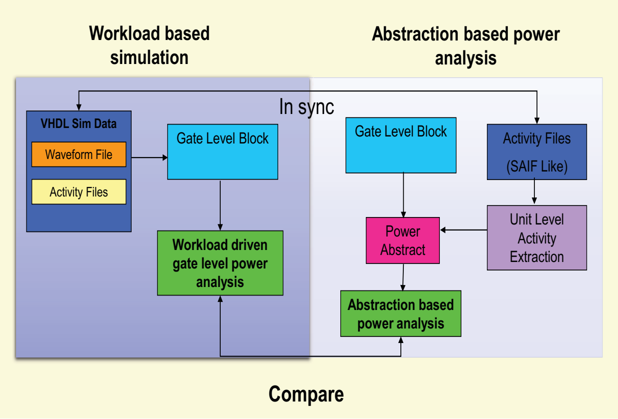

Workload based simulation
Abstraction based power analysis
In sync
VHDL Sim Data
Waveform File
Activity Files
Gate Level Block
Workload driven gate level power analysis
Gate Level Block
Activity Files (SAIF Like)
Power Abstract
Unit Level Activity Extraction
Abstraction based power analysis
Compare

Fig. 5. Experimental Setup

compared against the runtime of gate level simulation-based power computation. The runtime for abstraction-based power computation includes time for abstract generation and the time for evaluating the same abstract at a process, voltage, temperature and workload condition. “Model size increase” refers to the ratio of size of per clock gating domain power abstract (in bytes when stored on the disk) to the size of the base power abstract. The domain collapse procedure is triggered when the size of domain combinations list is greater than a certain threshold (DT) and is collapsed to DX% of the size of the domain combinations list. Both DT and DX are programmable and can be adjusted depending on accuracy/TAT/abstract size tradeoff requirements. For the studies in this paper, this was empirically chosen as DT=10 and DX=10%.

The first class of designs (D1) presented a huge challenge to single domain abstraction (base) approach (data power error is 54.87% and clock power error is 7.26%). This is dominated by several internal nets which switch non-uniformly under realistic switching conditions and needed pattern customization to generate a truly re-usable power abstract. This specific design D1 has 640 latches divided into 9 clock gating domains, 44 domain combinations and has around 13,000 standard cell instances and nets. Figure 6 shows the results comparing the different approaches for this design.

Cluster one of figure 6 shows the increase in size of the power abstract when compared to the single domain abstraction-based approach. The next shows the improvement in the turnaround time when compared to the gate level power simulation approach and the last two clusters show the clock and data power percentage error when compared with the gate level power simulation approach, using the TDP workload. The blue bars show the results of using the base abstraction where we approximate the multiple clock gating domains to a single clock gate control. The orange bars show results from the proposed per clock gating abstraction using approach 1, which brings down the overall error in clock power almost close to zero, with results from abstraction-based analysis matching gate level simulation. The grey and yellow bars show the results from clock and data power abstractions based on the domain combination approach and collapsing. Here we see the error in the data power come down significantly (data power error is now 16.30% and 17.41% respectively).

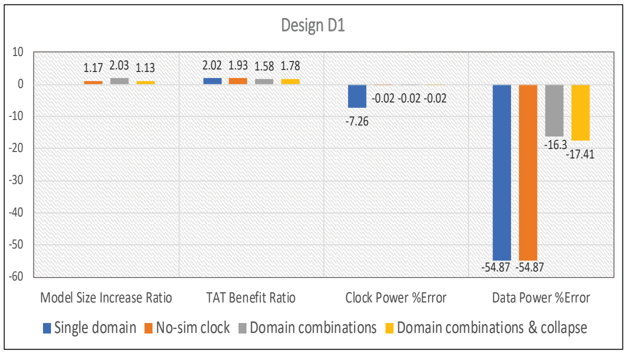


Fig. 6. Comparison for Corner Case Design D1

The next design, D2, is a much smaller design in terms of number of gated domains (3 clock domains, and 3 domain combinations) and has 10,000 gates and nets, 520 latches. Here, though the base abstraction [8] has “good enough” accuracy, using the proposed approach enables the accuracy of almost using gate level simulation-based power analysis. Figure 7 summarizes these results

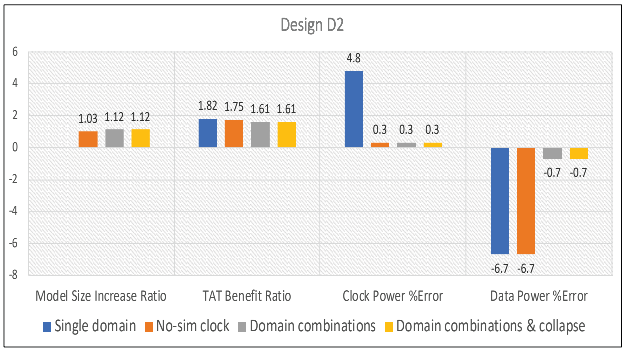


Fig. 7. Comparison for Design D2

Figure 8 summarizes the results for a larger design, D3. This has 1200 latches, 21 domains, and 83 domain combinations. Here the single domain abstraction’s data power error is 8.7% and clock power error is 4.1%. But with the proposed per clock gating domain abstraction, the accuracy improves further. Since, there were many gating domains in the design, domain collapsing step causes a notable reduction in the size of the abstract (1.98x to 1.08x), at an acceptable increase in the data power error (2.3% to 4.4%).

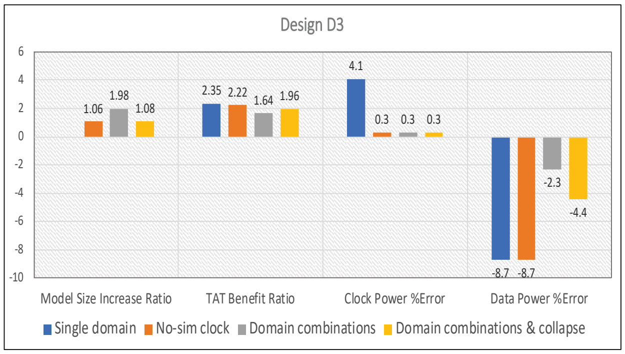


Fig. 8. Comparison for Design D3

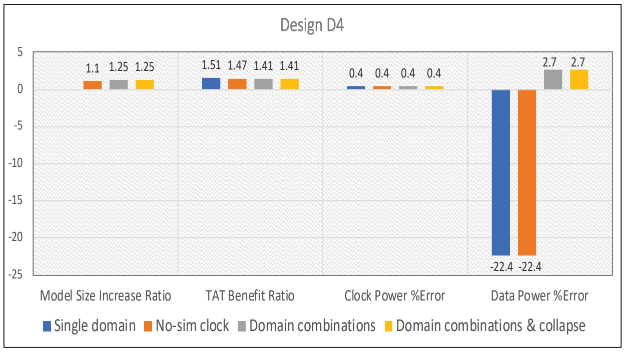


Fig. 9. Comparison for Design D4

Figures 9 and 10 show the comparison on other examples where the single domain abstraction without pattern customization was around 20% off in data power portion, but which could be brought down significantly with the proposed per clock gating domain abstractions.

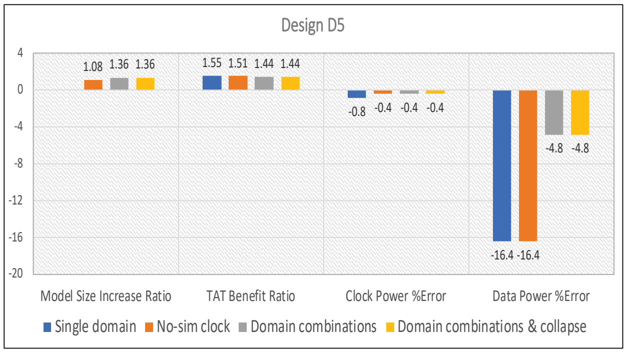


Fig. 10. Comparison for Design D5

VI. Conclusion

In this paper we presented industry class approaches for power modelling which efficiently captures the heterogeneity across workloads and within the distinctive design IP. This benefits rapid and accurate full chip power analysis. Experimental results comparing the proposed approach on IP blocks of varying sizes from a real industry strength microprocessor design, highlight the accuracy impact while keeping the run time and model size increase in an acceptable range. While this might seem like a straightforward extension to the base abstraction [8], we accomplish the gating domain parameterization by separating the attribution of switching due to each clock gating domain through a marking and tracing process. This precludes the need for multiple gating domain gate level simulations and makes the process of per clock gating domain parameterized power abstract generation to be very efficient and deployable in industry setting solutions. The accuracy benefits obtained are key for usage of such abstraction methodologies in FinFET-based microprocessor designs, where dynamic power is the major contributor, with its highly heterogenous signatures.

References

[1] Ioannis et al., "Putting On the Dynamic Power Glasses: A FinFET-Aware Approach for Early Realistic Block Activity Analysis and Exploration" Design Automation Conference, 2016.

[2] Warnock et al.,.: “Circuit and Physical Design Of The Zenterprise™ EC12 Microprocessor Chips AND Multi-Chip Module”, IEEE Journal Of Solid State Circuits, January 2014.

[3] Warnock et al, “22nm Next-Generation IBM System z Microprocessor”, Proceedings of 2015 International Solid State Circuits Conference.

[4] V. G. Oklobdzija, “Digital System Clocking—High-Performance and Low-Power Aspects,” New York, NY, USA,: Wiley, 2003.

[5] Jacobson, Hans, Pradip Bose, Zhigang Hu, Alper Buyuktosunoglu, Victor Zyuban, Rick Eickemeyer, Lee Eisen et al. “Stretching the limits of clock-gating efficiency in server-class processors”, In High-Performance Computer Architecture, 2005. HPCA-11.11th International Symposium on, pp. 238-242, IEEE, 2005.

[6] Arbel, E.; Eisner, C.; Rokhlenko, O., “Resurrecting infeasible clock-gating functions”, Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE , vol., no., pp.160,165, 26-31 July 2009.

[7] SI2 LPC “Power Modeling Requirements Document”, http://lpcwiki.si2.org.

[8] N. Dhanwada et al., “Efficient PVT Independent Abstraction of Large IP Blocks for Hierarchical Power Analysis”, Proc. ACM/IEEE ICCAD 2013.

[9] H. Jacobson et al "Empirically derived abstractions in uncore power modeling for a server-class processor chip," 2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), La Jolla, CA, 2014, pp. 147-152.

[10] Joseph et al “Efficient Techniques for Per Clock Gating Domain Contributor based Power Abstraction of IP Blocks for Hierarchical Power Analysis”, DAC 2016.

[11] Liberty Reference Manual, Sep. 2008, Version 2008.09.

[12] Z. Chen., K. Roy, “A power macro modeling technique based on power sensitivity”, ACM/IEEE DAC ‘98.

[13] Q. Qiu, Q. Wu, M. Pedram, C.-S. Ding, “Cycle-accurate macro-models for rt-level power analysis”, ISLPED ‘97.

[14] A. Bogliolo, R. Corgnati, E. Macii, M. Poncino, “Parameterized RTL Power Models for Soft Macros”, IEEE Trans. VLSI, Vol 9, December 2001.

[15] S. Gupta, F. Najm, “Power macromodeling for high level power estimation”, Proc. ACM/IEEE DAC ‘97.

[16] E. J. Fluhr et al., “IBM POWER9 circuit design and energy optimization for 14-nm technology”, IBM Journal of Research and Development, Volume: 62 , Issue: 4/5 , July-Sept. 1 2018.

[17] L. B. Arimilli et al, “IBM POWER9 processor and system features for computing in the cognitive era”, IBM Journal of Research and Development, Volume: 62 , Issue: 4/5 , July-Sept. 1 2018.