McPAT-Calib: A Microarchitecture Power Modeling Framework for Modern CPUs

Jianwang Zhai[1], Chen Bai[2], Binwu Zhu[2], Yici Cai[1], Qiang Zhou[1], Bei Yu[2]

[1]Tsinghua University
[2]The Chinese University of Hong Kong
{zhaijw18}@mails.tsinghua.edu.cn

Nov. 1, 2021

Outline

CPU Design

- Power consumption has become the main constraint limiting the performance of modern CPUs.
- Accurate power-performance tradeoff is necessary to ensure excellent CPU design.
- Large-scale design space (*e.g.*, RISC-V BOOM: $> 10^8$).

Challenges in Power Modeling

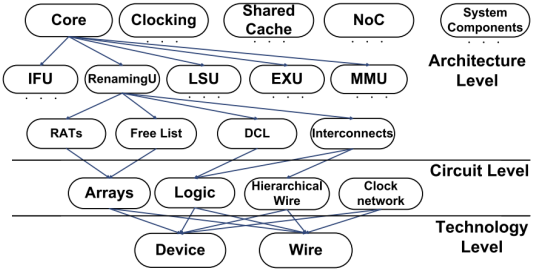
- High requirements: modeling speed, accuracy, and generality.
- **Speed**: the time required for the entire modeling flow.
- **Accuracy**: model complex microarchitectures and advanced technology nodes.
- **Generality**: model different CPU designs or different workload programs.

Table: Comparison of Existing Power Models

Model	Level	Speed	Generality	Accuracy
PrimeTime PX	Gate	Low	High	High
GRANNITE (DAC'20)	Gate	Medium	Medium	High
PRIMAL (DAC'19)	RTL	Medium	Medium	High
TCAD'17	Runtime	High	Low	High
McPAT (MICRO'09)	Arch	High	High	Low
McPAT-Calib	Arch	High	High	High

Limitations

- Cannot balance modeling speed, accuracy, and generality.
- Difficult to use in the early design stage of modern CPUs.



Hierarchical modeling methodology of McPAT [1]

Strengths

- Ease-of-use & Readiness; High speed & High generality.

Drawbacks

- Low accuracy; Lacks support for advanced technology nodes.

[1] Li, Sheng, et al. "McPAT: An integrated power, area, and timing modeling framework for multicore and

Definition (Power)

The total power can be expressed as:

$$P = P_{dynamic} + P_{leakage} = \underbrace{\alpha C V_{DD}^2 f + V_{DD} I_{leakage}}_{\text{Transistor level}} = \underbrace{\sum_{n=1}^{N} \beta_n f_n(E_n) + g(\boldsymbol{D})}_{\text{Microarchitecture level}} \quad (1)$$

Definition (Microarchitecture Configuration)

A CPU design characterized by a set of microarchitecture design parameters, such as *FetchWidth*, *DecodeWidth*, *FetchBufferEntry*, *etc.*.

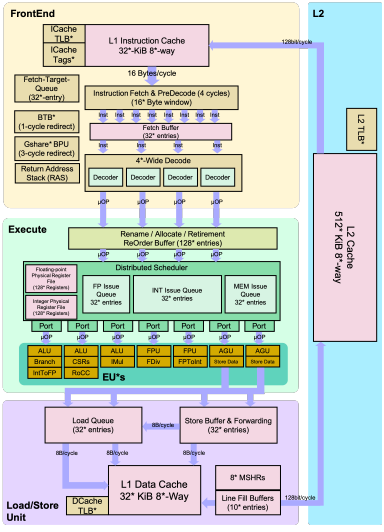
Definition (Benchmark)

The workload program executed on the target CPU.

Problem (Microarchitecture Power Modeling)

Given a set of CPU configurations $\mathcal{C}$ along with a set of benchmarks $\mathcal{B}$. The objective is to model the power P_{ij} of benchmark $B_j \in \mathcal{B}$ running on configuration $C_i \in \mathcal{C}$.

Detailed BOOM Pipeline [2]



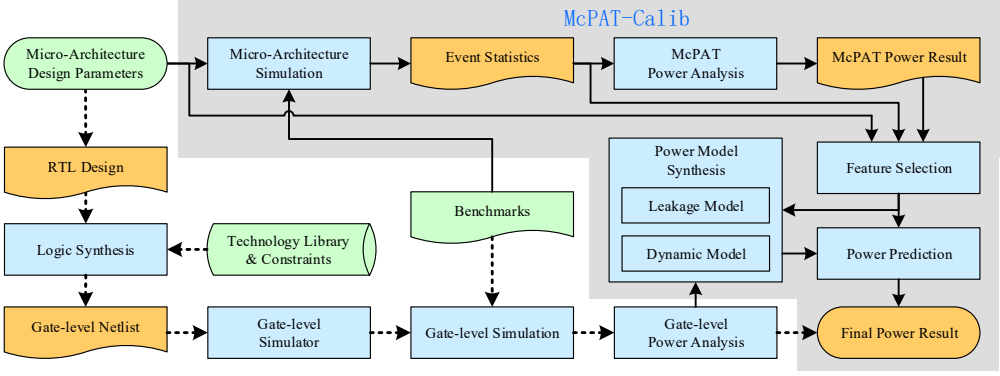
RISC-V

- Free & Open source; Easy to start.
- Has received great attention and support from academia and industry.

BOOM

- A family of out-of-order RISC-V designs.
- High performance & Parametric microarchitecture design & Automatic design flow.

[2] Zhao, Jerry, et al. "Sonicboom: The 3rd generation berkeley out-of-order machine." Fourth Workshop on Computer Architecture Research with RISC-V. 2020.



Power Modeling Flow

7nm FinFET Technology

Table: Key Parameters of 7nm FinFET PDK ASAP7 [3]

FinFET parameters	Value
Supply voltage, V_{DD} (V)	0.7
Gate length, L_G (nm)	21
Fin height, H_{FIN} (nm)	32
Fin thickness, T_{SI} (nm)	6.5
Fin pitch, F_P (nm)	27
Contacted poly-pitch, CPP (nm)	54

Empirical Coefficients Adjustment

Adjust empirical undifferetiated Core/FU coefficients to reduce modeling errors.

Microarchitecture Modification

Modify McPAT to support accurate modeling of the RISCV BOOM (*e.g.*, pipeline).

[3] L. T. Clark, et al., "ASAP7: A 7-nm finFET predictive process design kit," in Microelectronics Journal, 2016.

Total Power Calibration

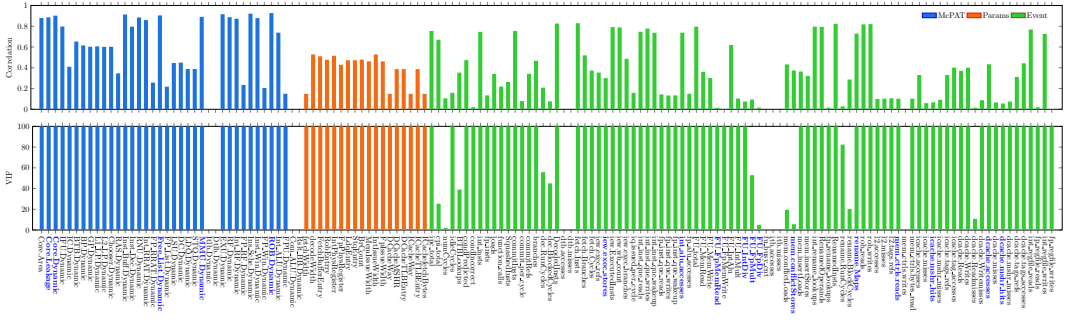
- Calibrate leakage and dynamic separately, and then take the sum.

Leakage Power Calibration

- Method: model the $\overline{leakage}$ of one CPU configuration.
- Feature: McPAT Results (2: *Core.Leakage* and *Core.Area*).

Dynamic Power Calibration

- Method: model the *dynamic* of each sample.
- Feature: McPAT Results (38) & Event Statistics (90) & Design Parameters (18).



Correlation with Dynamic and VIF of Dynamic Modeling Features.

Multicollinearity

- Variance Inflation Factor (VIF):

$$VIF = \frac{1}{1 - R^2} \tag{2}$$

- High model complexity & Lack of stability & Overfitting.
- Fail to accurately predict unknown configurations or benchmarks.

Algorithm 1 Filter Sequential Feature Selection

Require: *allFeatures*, all modeling features; k, the number of features to select;
varThreshold, the variance threshold used to filter features;
Ensure: *selectedList*, the selected k optimal features;

```
for tmpFeature in allFeatures do
    if var(tmpFeature) ≤ varThreshold then;
        Delete tmpFeature from allFeatures;
    end if
end for
selectedList = ϕ;
while selectedList.length < k do
    bestR^2 = −inf;
    for tmpFeature in allFeatures do
        Cross-Validation(selectedList + tmpFeature);
        if newR^2 > bestR^2 then;
            bestR^2 = newR^2; bestFeature = tmpFeature;
        end if
    end for
    Add bestFeature to selectedList; Delete bestFeature from allFeatures;
end while
```

Nonlinearity

- When complex workloads are executed on a CPU, the relationship between specific modeling feature X_i and resulting dynamic power $P_{dynamic}$ is nonlinear.

$$P_{dynamic} \sim f(X_i) \tag{3}$$

XGBoost Regressor

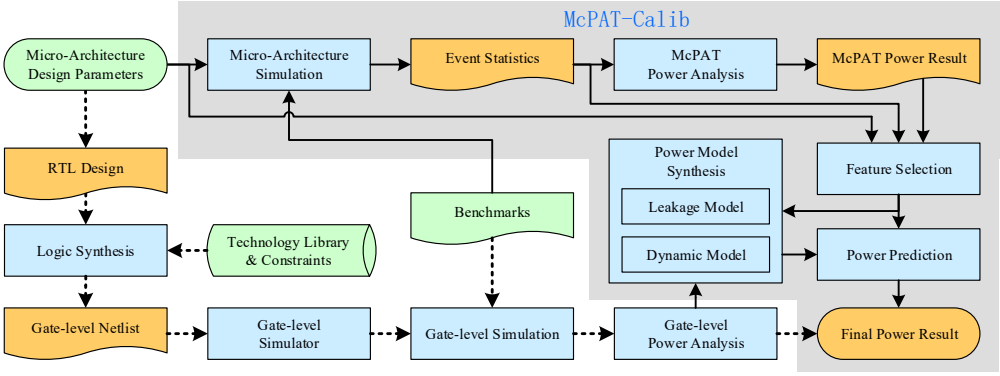
- A scalable end-to-end tree ensemble model based on gradient boosting:

$$\hat{y}_i = \phi(\boldsymbol{X}_i) = \sum_{k=1}^{K} f_k(\boldsymbol{x}_i), \quad f_k \in \mathcal{F} \tag{4}$$

- To learn the regression tree functions, minimize the following regularized objective:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \quad \textit{where} \quad \Omega(f) = \gamma T + \frac{1}{2}\lambda ||\omega||^2 \tag{5}$$

Modleing Flow



Power Modeling Flow

AL Sampling

Challenge

- A large number of labeled samples are needed to train the model.
- Labeling samples requires gate-level simulation and power analysis, which is time-consuming (about 5-20 hours) and an unacceptable cost.

Motivation

- It is easy to obtain modeling features and only takes a few seconds.
- How to select the most beneficial samples to label under a limited budget?

Pool-based Sequential Active Learning (AL)

- Obtain the features of all samples to form a sample pool $\{\boldsymbol{x}_n\}_{n=1}^{N}$.
- Each time the most useful sample is selected to label and added to the training set.

Initial Samples Selection

- Pre-clustering: to ensure the representativeness and diversity.

Sample Query Strategy

- To increase the diversity in both feature and label spaces.
- In each iteration, select the sample $\boldsymbol{x}_n$ with the maximum d_n^{xy} to label:

$$d_n^{xy} = \min_m ||\boldsymbol{x}_n - \boldsymbol{x}_m|| \, || f(\boldsymbol{x}_n) - y_m ||, \quad m = 1, ..., k; n = k+1, ..., N \tag{6}$$

where $f(\boldsymbol{x})$ is built by labeled samples $\{\boldsymbol{x}_m, y_m\}_{m=1}^k$; and unlabeled samples $\{\boldsymbol{x}_n\}_{n=k+1}^N$.

Stop Criteria

- The number of labeled samples reaches the budget M.

Algorithm 2 Pre-clustering Sequential AL Sampling

Require: $\mathcal{S}$, a set of unlabeled samples $\{\boldsymbol{x}_n\}_{n=1}^{N}$, where $\boldsymbol{x}_n \in \mathbb{R}^d$; M, the maximum number of samples to label;

Ensure: $\mathcal{K}$, the training set of labeled samples $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^{M}$; $f(\boldsymbol{x})$, the power model;

$\mathcal{K} = \phi$;
Perform k-means clustering on S to obtain d clusters, $\mathcal{C}_i, i = 1, ..., d$;
for $i = 1 : d$ **do**
    Select the sample $\boldsymbol{x}$ closet to the center of $\mathcal{C}_i$ to label;
    Add $(\boldsymbol{x}, y)$ to $\mathcal{K}$, delete $\boldsymbol{x}$ from $\mathcal{S}$;
end for
for $i = d + 1 : M$ **do**
    Use the sample query strategy to select the most beneficial sample $\boldsymbol{x}$ in $\mathcal{S}$ to label;
    Add $(\boldsymbol{x}, y)$ to $\mathcal{K}$, delete $\boldsymbol{x}$ from $\mathcal{S}$;
end for
Use the training set $\mathcal{K}$ to build the power model $f(\boldsymbol{x})$.

Experiments Settings

- 15 typical RISC-V BOOM configurations; 80 commonly used benchmarks.
- Total $15 \times 80 = 1200$ samples.

Table: Design Parameters and Power Statistics of Our 15 BOOM Configurations

Parameters	SmallBoomConfig			MediumBoomConfig			LargeBoomConfig			MegaBoomConfig			GigaBoomConfig		
	SE	Default	Pro	SE	Default	Pro	SE	Default	Pro	SE	Default	Pro	SE	Default	Pro
FetchWidth	4	4	4	4	4	8	8	8	8	8	8	8	8	8	8
DecodeWidth	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5
FetchBufferEntry	5	8	16	8	16	24	18	24	30	24	32	40	30	35	40
RobEntry	16	32	48	64	64	80	81	96	114	112	128	136	125	130	140
IntPhysRegister	36	52	68	64	80	88	88	100	112	108	128	136	108	128	140
FpPhysRegister	36	48	56	56	64	72	88	96	112	108	128	136	108	128	140
LDQ/STQEntriy	4	8	16	12	16	20	16	24	32	24	32	36	24	32	36
BranchCount	6	8	10	10	12	14	14	16	16	18	20	20	18	20	20
MemIssue/FpIssueWidth	1	1	1	1	1	1	1	1	2	1	2	2	2	2	2
IntIssueWidth	1	1	2	1	2	2	2	3	3	4	4	4	5	5	5
DCache/ICacheWay	2	4	8	4	4	8	8	8	8	8	8	8	8	8	8
DCache/ICacheTLBEntry	8	8	16	8	8	16	16	16	32	32	32	32	32	32	32
DCacheMSHR	2	2	4	2	2	4	4	4	4	4	4	8	8	8	8
ICacheFetchBytes	2	2	2	2	2	4	4	4	4	4	4	4	4	4	4
Min.Power(mW)	9.54	10.22	12.11	11.89	13.10	19.07	21.36	22.81	28.03	26.39	34.10	34.57	37.15	34.12	36.70
Max.Power(mW)	14.13	16.69	19.94	22.64	27.74	32.79	38.07	42.56	50.52	51.36	62.72	64.22	61.80	59.75	63.82
Avg.Power(mW)	11.76	13.53	15.64	16.42	17.94	24.60	28.02	30.02	35.97	36.55	44.06	45.52	45.62	43.26	46.38
Std.Power(mW)	1.22	1.70	1.73	2.81	3.95	3.76	4.62	5.00	5.56	6.06	7.27	7.84	6.00	6.64	7.10

Mean absolute percentage error (MAPE)

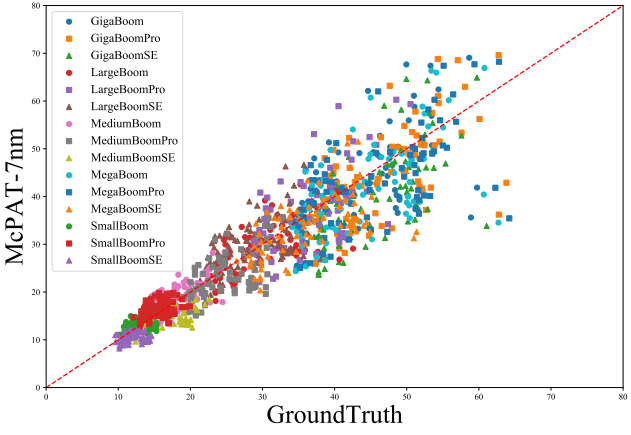
$$\text{MAPE} = \frac{1}{n}\sum_{i}^{n}\frac{|p_i^{pred} - p_i^{truth}|}{p_i^{truth}} \times 100\% \tag{7}$$

coefficient of determination (R^2)

$$R^2 = 1 - \frac{\sum_i^n (p_i^{pred} - p_i^{truth})^2}{\sum_i^n (p_i^{truth} - \overline{p}^{truth})^2} \tag{8}$$

Preliminary Power Modeling Results

Total 1200 samples: MAPE $= 13.02\%$ and $R^2 = 0.817$.



McPAT-7nm Modeling Results

Leakage Power and Dynamic Power

- Due to multicollinearity, most models cannot obtain good results using all features.
- Feature selection can effectively improve accuracy, especially for linear models.

Table: Leakage and Dynamic Modeling Results

Regressors	$\overline{Leakage}$	Dynamic-Total Features		Dynamic-Selected Features		
	MAPE	MAPE	R^2	k^*	MAPE	R^2
LR	7.34%	20.85%	0.816	48	7.40%	0.954
Lasso	8.08%	17.97%	0.869	48	7.55%	0.951
Ridge	7.10%	21.88%	0.790	48	7.31%	0.954
ElasticNet	6.77%	16.36%	0.889	22	9.20%	0.929
BRR	7.74%	18.50%	0.867	48	7.30%	0.954
GPR	7.72%	16.29%	0.895	15	9.32%	0.924
KNNR	8.21%	20.64%	0.783	13	13.21%	0.903
Poly_SVR	**4.47%**	35.04%	0.462	18	9.34%	0.923
RBF_SVR	6.09%	31.41%	0.504	21	8.99%	0.940
DTR	7.76%	14.70%	0.877	22	11.61%	0.914
RFR	7.46%	10.56%	0.943	6	8.09%	0.958
ABR	7.64%	14.24%	0.907	11	13.26%	0.893
GBR	8.88%	10.98%	0.936	28	9.25%	0.943
BAGR	7.59%	11.41%	0.931	6	9.92%	0.933
XGBR	7.81%	**7.40%**	**0.961**	17	**6.23%**	**0.969**

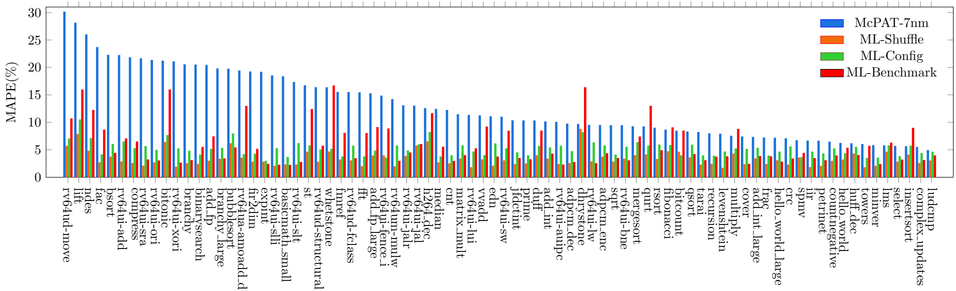
* k: The Number of Selected Features

Total Power: Model

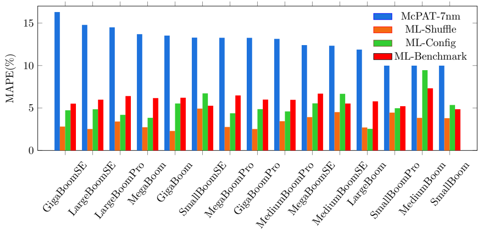
- Leakage Power: 2-degree Ploy_SVR.
- Dynamic Power: XGBoost Regressor.
- Total Power: $P_{total} = P_{dynamic} + P_{leakage}$.

Total Power: Cross-Validation (CV)

- 15-fold Shuffle-Split CV: MAPE $= 3.38\%$, $R^2 = 0.989$.
  Treat all samples as equal and perform random split validation.
- 15-fold Config-Split CV: MAPE $= 5.22\%$, $R^2 = 0.978$.
  Split according to configuration to simulate modeling unknown configurations.
- 20-fold Bench-Split CV: MAPE $= 5.96\%$, $R^2 = 0.958$.
  Split according to benchmark to simulate modeling unknown benchmarks.



Power Modeling Results of Different Benchmarks.



Power Modeling Results of Different Configurations.

Baselines

- Design parameter-based: HPCA'07 [BC Lee, DM Brooks. *HPCA*, 2007.]
- Event statistics-based: TCAD'17 [MJ Walker, S Diestelhorst, A Hansson, et al. *TCAD*, 2017.], TCAD'20 [M Sagi, NAV Doan, M Rapp, et al. *TCAD*, 2020.]
- McPAT result-based: PowerTrain [W Lee, Y Kim, JH Ryoo, et al. *ISLPED*, 2015.]

Table: Comparison with previous work

Methods	Shuffle-Split		Unknown Config.		Unknown Bench.	
	MAPE	R^2	MAPE	R^2	MAPE	R^2
HPCA'07	15.31%	0.807	18.37%	0.752	15.34%	0.807
TCAD'17	11.71%	0.899	14.31%	0.875	13.56%	0.842
TCAD'20	22.51%	0.746	24.58%	0.711	23.92%	0.690
PowerTrain	9.33%	0.926	11.36%	0.906	9.60%	0.921
McPAT-7nm	13.02%	0.817	13.02%	0.817	13.02%	0.817
McPAT-Calib	**3.38%**	**0.989**	**5.22%**	**0.978**	**5.96%**	**0.958**

Sampling Results

- 15-fold Config-Split CV: 1120 training samples, 80 testing samples.
- Our AL sampling algorithm can effectively reduce the demand for labeled samples.
- Reduce the demand for labeled samples by 50% with only a 0.44% loss of accuracy.

Table: MAPE under several typical sampling ratios

Ratio	10% (112)	20% (224)	30% (336)	40% (448)	50% (560)
MAPE	8.68%	6.91%	6.41%	5.92%	**5.66%**
Ratio	60% (672)	70% (784)	80% (896)	90% (1008)	100% (1120)
MAPE	5.65%	5.77%	5.47%	5.56%	**5.22%**

Conclusion

Why McPAT-Calib effective?

- McPAT-7nm: Supports analytical power modeling by introducing 7nm FinFET technology and microarchitecture modifications. It can also be used alone.
- ML Calibration: Separate calibration of leakage/dynamic & A wide range of feature sources & Automatic feature selection & Advanced nonlinear regression.
- AL Sampling: The pursuit of sample diversity greatly reduces the demand for labeled samples.

Prospect

- Performance/Area/Timing Modeling?
- The DSE of modern CPUs, *i.e.*, modeling a larger design space.

THANK YOU!