Estimation of Speed, Area, and Power of Parameterizable, Soft IP

Jagesh Sanghavi
Tensilica, Inc.
sanghavi@tensilica.com

Albert Wang
Tensilica, Inc.
awang@tensilica.com

ABSTRACT

We present a new approach to estimate speed, area, and power of a parameterizable, soft IP. By running the ASIC implementation flow only on selected configurations, we predict the performance for any arbitrary configuration. We exploit performance function decomposability to address the combinatorial explosion challenge. The estimator has been used successfully to configure Xtensa processor cores for numerous embedded SOC designs.

1. INTRODUCTION

IP cores are key to exploiting the increasing VLSI integration capability. A parameterized IP can be customized for a specific application. The parameterization leads to the estimation challenge due to combinatorial explosion in the number of configurations. For example, if an IP has 10 parameters, each of which can take one of four values, then the number of configurations is over one million. Furthermore, the process of characterizing a non-trivial parameterizable core using a state-of-the-art ASIC flow, even for one configuration, is a time consuming and resource intensive process. It can take anywhere from several hours to a few days to obtain the characterization data on speed, area, and power by running synthesis, layout, and power analysis tools. In this paper, we address the combinatorial explosion problem by exploiting the decomposability of performance metric functions.

The rest of the paper is organized as follows. In Section 2, we begin with a brief overview of Xtensa parameterizable IP. In Section 3, we develop the decomposition theory. In Section 4, we present the estimation algorithms. Next, we present experimental results in Section 5. Finally, we present conclusions in Section 6.

2. XTENSA PARAMETERIZABLE IP

Although the concepts presented in this paper are applicable to any parameterizable IP, we focus our discussion on Xtensa configurable processor cores [1] from Tensilica [4]. The present generation of Xtensa III cores range in complexity from base configuration with approximately 25000 gates to configurations with several hundred thousand gates that include extensions like a Vectra SIMD DSP, a floating point unit, and a 32-bit multiplier.

The what-if analysis to customize a parameterizable IP requires interactive and reasonably accurate feedback. The interactivity implies that we estimate the cost and performance impact of a feature. The accuracy implies that we rely on actual data obtained by using the ASIC implementation flow. In constrast to previous estimation approaches [2, 5, 3], we exploit underlying mathmatical structure of the problem to cope with the combinatorial complexity. In the next section, we present the decomposition theory that drastically reduces the number of configurations on which to run the ASIC implementation flow.

3. DECOMPOSITION

A configuration variable is a discrete variable that takes a value from an ordered, finite set. A configuration space is a multidimensional discrete space, where each dimension corresponds to a configuration variable and each point corresponds to a configuration. A metric function over a configuration space is a function that maps each configuration point to a unique real value. Each of speed, area, and power is a metric function.

The characterization problem over a configuration space $S(X)$ spanned by variables X is defined as knowing the metric function $f(X)$ at the minimum number of configuration points so that it is possible to determine the metric function at any arbitrary point in the configuration space $S(X)$. A characterization set $CS(f,X)$ for $f(X)$ over $S(X)$ is the minimum set of configuration points at which metric function should be known to determine the metric function $f(X)$ at any arbitrary point in the configuration space $S(X)$.

In the following discussion, let x_i be a configuration variable that takes value from an ordered, finite set C_i. X is a k tuple of configuration variables $X = \{x_1, x_2, \ldots, x_j, x_{j+1}, \ldots, x_k\}$. Let X_1 be a j tuple $\{x_1, x_2, \ldots, x_j\}$ and X_2 be a $k-j$ tuple $\{x_{j+1}, \ldots, x_k\}$ such that k tuple X can be expressed as $\{X_1, X_2\}$.

A metric function $f(X_1, X_2)$ is independent of X_2 if $f(X_1, X_2) = f_1(X_1)$. If a metric function f is independent of variable x_i, we can fix its value to one of the values from set C_i, while determining the characterization set of f.

Let $X[p]$ be a point in the configuration space $S(X)$. Let $X_1[p]$ and $X_2[p]$ refer to points in $S(X_1)$ and $S(X_2)$ respectively such that each configuration variable is set to the value of the corresponding configuration variable for the point $X[p]$.

THEOREM 3.1. *If $f(X_1, X_2) = f_1(X_1) + f_2(X_2)$ then for a configuration point $X[p]$,*

$$f(X_1, X_2) = f(X_1, X_2[p]) + f(X_1[p], X_2) - f(X[p])$$

DAC 2001, June 18-22, 2001, Las Vegas, Nevada, USA.

THEOREM 3.2. *If* $f(X_1,X_2) = f_1(X_1) * f_2(X_2)$ *then for a configuration point* $X[p]$ *such that* $f(X[p]) \neq 0$,

$$f(X_1,X_2) = f(X_1,X_2[p]) * f(X_1[p],X_2)/f(X[p])$$

THEOREM 3.3. *If* $f(X_1,X_2) = Max\,(f_1(X_1), f_2(X_2))$, *where Max is maximum operator then for a configuration point* $X[p]$ *such that* $f(X[p]) \leq f(X) \forall X$,

$$f(X_1,X_2) = Max(f(X_1,X_2[p]), f(X_1[p],X_2))$$

The decomposition of $f(X)$ using operator $\otimes$ with respect to X_1 and X_2 is defined as $f(X) = f_1(X_1) \otimes f_2(X_2)$. From theorems above, if a metric function is decomposed as a set of component functions, each of which depend only on disjoint subset of variables, then it is possible to determine the metric function completely only by knowing its value for each subset of configuration variables while keeping the remaining variables constant. The implication for parameterizable IP is that we can perform black box characterization and exploit decomposability.

An augmented characterization set $ACS(f_1,\{X_1,X_2[p]\})$ refers to a set of configuration points obtained by augmenting each element of $CS(f_1,X_1)$ with $X_2[p]$. Each element of $CS(f_1,X_1)$ is a j tuple $\{x_1,x_2,\ldots,x_j\}$. $X_2[p]$ is $k-j$ tuple evaluated at point p. Each j tuple element X_1 is augmented by $k-j$ tuple X_2 evaluated at point p. After augmentation the element is $\{x_1,x_2,\ldots,x_j,x_{j+1}[p],\ldots,x_k[p]\}$. The augmentation can be postfix resulting in $\{X_1,X_2[p]\}$ or prefix $\{X_1[p],X_2\}$.

THEOREM 3.4. *If* $f(X_1,X_2) = f_1(X_1) \otimes f_2(X_2)$ *and if* $X[p]$ *is an arbitrary configuration point such that any one of the following is true:*

- $\otimes$ *equal to* $+$
- $\otimes$ *equal to* $*$ *and* $f(X[p]) \neq 0$
- $\otimes$ *equal to Max and* $f(X[p]) \leq f(X) \forall X$

then

$$CS(f,\{X_1,X_2\}) = ACS(f_1,\{X_1,X_2[p]\}) \cup ACS(f_2,\{X_1[p],X_2\})$$

THEOREM 3.5. *If* $f(X_1,X_2,X_3) = f_1(X_1) * (f_2(X_2) \otimes f_3(X_3))$ *and if* $X[p]$ *is an arbitrary configuration point such that* $f(X[p]) \neq 0$ *and either one of the following is true:*

- $\otimes$ *equal to* $+$ *or* $*$
- $\otimes$ *equal to Max and* $0 < f(X[p]) \leq f(X) \forall X$

then

$$CS(f,\{X_1,X_2,X_3\}) = ACS(f_1,\{X_1,X_2[p],X_3[p]\}) \cup ACS(f_2,\{X_1[p],X_2,X_3[p]\}) \cup ACS(f_3,\{X_1[p],X_2[p],X_3\})$$

THEOREM 3.6 (DECOMPOSITION). *If*

$$f(X_{11},\ldots,X_{1n_1},X_{21},\ldots,X_{2n_2},\ldots,X_{m1},\ldots,X_{mn_m}) = \prod_{i=1}^{m}(f_{i1}(X_{i1}) \otimes_i f_{i2}(X_{i2}) \otimes_i \ldots f_{in_i}(X_{in_i}))$$

and if $X[p]$ *is an arbitrary configuration point such that* $f(X[p]) \neq 0$ *and either one of the following is true:*

- $\otimes_i$ *equal to* $+$ *or* $*$
- $\otimes_i$ *equal to Max and* $0 < f(X[p]) \leq f(X) \forall X$

then

$$CS(f,\{X_{ij}, i=1,\ldots,m, j=1,\ldots,n_m\}) = \bigcup_{i=1}^{m}\bigcup_{j=1}^{n_m} ACS(f_{ij}, X = X[p] \text{ except } X_{ij})$$

The last theorem is the key to combat combinatorial complexity. By decomposing f into disjoint set of variables, we can replace a large characterization problem with many small characterization problems.

4. ESTIMATION

In this section, we apply decomposition theory to characterize each of speed, area, and power metric function. The underlying assumption is that variables can be partitioned into disjoint subsets such that second order interaction between variables in different subsets is very small. Each of speed, area, and power metric function is decomposed using operator $*$ into technology dependent and technology independent variables.

$$f(T,M) = f_T(T) * f_M(M)$$

The technology dependent variables (T) are target implementation geometry, process corner, and operating condition. The technology independent variables (M) are architectural and microarchitectural variables. From Theorem 3.2 and Theorem 3.4, we can now solve two smaller characterization problems. First, we characterize each metric function with respect to different technologies for a single configuration of architectural and microarchitectural variables. Second, we characterize each metric function with respect to different architectural and microarchitectural variables for a specific target technology.

4.1 Characterizing Speed

The speed metric function is represented by the clock period. For a target technology, the clock period is determined by the maximum length timing path that depends only on architectural and microarchitectural variables. Assume that the architectural and microarchitectural variables M can be grouped into disjoint subsets $M_i, i=1,\ldots,k$ such that the length of each timing path depends only on variables in any one subset. The subsets $M_i, i=1,\ldots,k$ imposes a partition on the set of timing paths. The maximum length of path in partition i is a function of variables M_i. Therefore, the maximum length function over the set of timing paths is equal to the maximum length function over the subsets $M_i, i=1,\ldots,k$.

$$t(M_1,\ldots,M_k) = Max_{i\in\{1,\ldots,k\}} t_{M_i}(M_i)$$

where t_{M_i} is the maximum length function that depends only on variables M_i.

The clock period is smallest for the minimum configuration. Hence, we have a configuration point p for which $t(M[p]) \leq t(M)$. From Theorem 3.3 and Theorem 3.4, we can now characterize the speed metric function only with respect to each subset of variables M_i.

Example: Consider a processor IP with datapath that includes a parameterizable ALU and a parameterizable MAC. Depending on the choice of parameters, the critical path can be from read port of a register file through either ALU or MAC. For a target technology, we characterize the clock period for the IP by computing clock period values for configurations obtained as follows. First, vary parameters related to ALU with minimum parameter values for MAC. Second, vary parameters related to MAC with minimum parameter values for ALU. ■

In summary, the speed metric function is decomposed using operator $*$ into component functions that depend on technology dependent variables and technology independent variables. The technology independent component function is further decomposed using *Max* operator into disjoint subsets of architectural and microarchitectural variables. From Theorem 3.6, we determine a characterization set that is very small compared to the number of points in the configuration space.

For each configuration in the characterization set, the speed is computed by running a state-of-the-art standard cell implementation flow. We make use of industry standard tools such as Design Compiler from Synopsys and Silicon Ensemble from Cadence.

4.2 Characterizing Area

The area metric function is decomposed using operator $*$ with respect to technology dependent variables and technology independent variables. Furthermore, the component function that depends only on technology dependent variables is decomposed using operator $*$ into a set of functions, each of which depends only on one variable. The component function that depends only on the technology independent variables is decomposed using operator $+$ into a set of functions, each of which depend on a disjoint subsets of related architectural and microarchitectural variables.

If we represent implementation variables by $x_1, x_2, \ldots, x_k$ and disjoint subsets of architectural and microarchitectural variables by $A_j \forall j \in \{1, 2, \ldots, m\}$ then the area metric function is decomposed as follows:

$$f(x_1, \ldots, x_k, A_1, \ldots, A_m) = \prod_{i=1}^{k} f_i(x_i) * \sum_{j=1}^{m} g_j(A_j)$$

The Theorem 3.6 is applied to characterize area only with respect to variables $x_1, \ldots, x_k$ and subsets $A_j \forall j \in \{1, \ldots, m\}$.

Example: Consider once again a processor IP with datapath that includes a parameterizable ALU and a parameterizable MAC. The area due to ALU and MAC contribute additively to the area of the IP. The area contribution due to ALU and MAC depends on the choice of parameters. We can characterize the area of the IP by computing area for configurations obtained as follows. First, vary parameters related to ALU while choosing fixed parameter values for MAC. Second, vary parameters related to MAC while choosing fixed parameter values for ALU. ■

For Xtensa configurable processor the area estimation algorithm starts with the area of the base Xtensa processor. Next, the area due to each of the following architectural and microarchitectural features is added:

1. optional Vectra SIMD DSP;
2. optional floating point unit;
3. optional 32-bit multiplier;
4. optional instructions;
5. number and types of interrupts;
6. number of timers;
7. register file entries and register file building block;
8. write buffer entries and write buffer building block;
9. width of processor interface;
10. debug support (trace port, on-chip debug module); and
11. cache control logic.

Once the area estimate is obtained by adding up contributions from architectural and microarchitectural variables for the specific values of implementation variables, the impact of changing implementation variables is accounted for by appropriate scaling factors. The scaling factors are provided for each of the following:

- technology: 0.18 micron, 0.25 micron, and 0.35 micron;
- process corner; and
- operating condition

Note that the area calculation above relies on generalization of the following identities:

$$f(X_1, X_2[p]) + f(X_1[p], X_2) - f(X[p])$$
$$= f(X[p]) + (f(X_1, X_2[p]) - f(X[p])) + (f(X_1[p], X_2) - f(X[p]))$$

$$f(X_1, X_2[p]) * f(X_1[p], X_2) / f(X[p])$$
$$= f(X[p]) * (f(X_1, X_2[p]) / f(X[p])) * (f(X_1[p], X_2) / f(X[p]))$$

4.3 Characterizing Power

Characterizing power is difficult, since it depends not only on configuration variables, but also on dynamic behavior of the circuit. The switching activity can vary widely for a processor between running "wait" instructions in power-down mode to running arithmetic instructions.

Besides implementation technology, voltage, and operating conditions, the implementation feature that has significant impact on power dissipation is the clock gating based power management technique.

The power metric function is decomposed using operator $*$ with respect to technology dependent variables, presence or absence of clock gating, and architectural configuration variables. The impact of architectural configuration variables is accounted for by making power dissipation proportional to the area.

4.4 Error Mitigation

The two sources of estimation errors are interaction between variables in disjoint subsets and reduced order modeling of a metric function. To mitigate errors, we rely on redundancy. For example, we characterize technology with respect to multiple configurations of architectural and microarchitectural variables. The geometric mean is used to determine technology related scale factors. For a function in one variable, only two datapoints are sufficient for a linear model. However, the error due to reduced order modeling is mitigated by linear fit for a set of datapoints. The theoretical error analysis, which might provide a better insight into improving the estimation, is a topic of future research.

5. EXPERIMENTAL RESULTS

The algorithms are implemented as a part of the Xtensa estimator tool. The Xtensa estimator tool is very easy to use. The user can change the configuration variables using pull down menus from a web browser GUI. In response to the changes in configuration variables, the estimation bars for speed, area, and power are updated interactively.

We synthesized 118 different configurations using Design Compiler. Due to resource limitation, the layout was completed on 39 of them using Silicon Ensemble. The speed is determined using post layout clock period. Since post layout area is related to post synthesis area by a utilization factor, the area metric function is represented by the post synthesis area. For each configuration, the percentage error in either area or speed is computed as 100 * (estimated - actual) / actual.

For area estimation on 118 configurations, the statistical measures of percentage error are as follows: the arithmetic mean is -0.9419; the standard deviation is 2.3277; and the range is (-10.97, 10.90). The geometric mean of ratio of estimated to actual area is 0.9903. The histogram of number of configurations vs. percentage error in area is shown in Figure 1.

For speed estimation on 39 configurations, the statistical measures of percentage error are as follows: the arithmetic mean is -0.2741; the standard deviation is 3.2573; and the range is (-10.42, 7.84). The geometric mean of ratio of estimated to actual speed is 0.9967. The histogram of number of configurations vs. percentage error for speed is shown in Figure 2.

The statistical measures and histograms provide a good degree of confidence in speed and area estimation algorithms. To provide a better understanding of estimated and actual metrics for different Xtensa configurations, a representative set of sample points is shown in Figure 3. The first column is the name of a configuration. The second

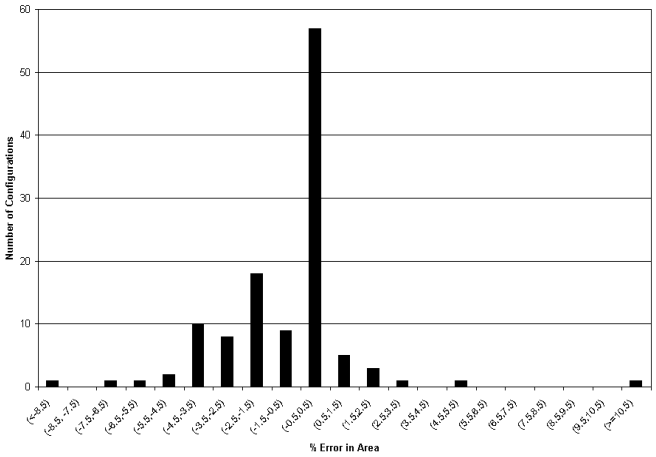

Figure 1: Number of Configurations vs. % Error in Area

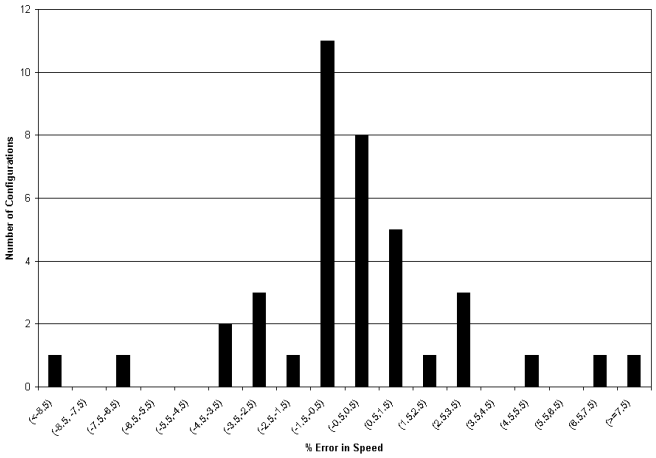

Figure 2: Number of Configurations vs. % Error in Speed

and third columns are actual area and estimated area in square microns. The fourth column is the percentage error in area. The fifth and sixth columns are actual speed and estimated speed in MHz. The final column is the percentage error in speed.

A representative set of results for power estimation are shown in Figure 4. The first column is the name of a configuration. The second column is the post layout clock frequency in MHz. The third and fourth columns are actual power and estimated power in milliwatts. The power is computed using Synopsys DesignPower on the post layout netlist taking into account net parasitics. The switching activity data is obtained by simulating a representative set of architectural and microarchitectural verification programs on the post layout netlist. The fifth column is the percentage error in power. The results point to the weakness of area based scaling algorithm for power estimation. We are experimenting to identify architectural and microarchitectural parameters that can have significant impact on the dissipated power.

6. CONCLUSIONS AND FUTURE WORK

We address the combinatorial explosion problem in characterizing parameterizable IP. We outline algorithms based on function decomposition theory to estimate speed, area, and power for a parameterizable IP. The experimental results show a close correlation between actual and estimated values of speed and area for a set of configurations. In future, we plan to improve estimation accuracy for power and estimation accuracy across different vendor libraries.

Cfg	A	EsA	ErA	S	EsS	ErS
XFS	683279	672114	-1.6	200	192	-4.0
XS	625017	609220	-2.5	202	202	0.0
XT	537244	548514	2.1	296	310	4.7
Xm25	1284870	1290070	0.4	147	150	2.0
Mx	1573318	1492685	-5.1	169	171	1.2
MxS	1612216	1560321	-3.2	153	165	7.8
MxT	1493620	1404844	-5.9	237	254	7.2
Mx25	3504281	3336404	-4.8	125	129	3.2
ML32	836836	836729	-0.0	195	194	-0.5
PID4	825679	827405	0.2	195	194	-0.5
DbTr	694077	686412	-1.1	199	199	0.00
Roam	670795	647551	-3.5	187	186	-0.5
Intr	671687	668111	-0.5	204	201	-1.5
IsOp	717514	699143	-2.6	201	198	-1.5
P	838868	837116	-0.2	197	194	-1.5
Pm25	2101837	1871096	-11.0	152	147	-3.3
MW	905682	903523	-0.2	199	191	-4.0
XTie	1328737	1330322	0.1	170	175	2.9

Figure 3: Area and Speed Data for Xtensa configurations

Cfg	S	P	EsP	ErP
X	209	87.78	92	4.81
XLCG	207	91.08	96	5.40
XT	296	142.08	165	16.13
DbTr	199	103.48	93	-10.13
Roam	187	91.63	85	-7.24
Intr	204	108.12	93	-13.98
P	197	106.38	98	-7.88
MW	199	101.49	99	-2.45

Figure 4: Power Data for Xtensa configurations

7. ACKNOWLEDGEMENTS

Thanks to Kaushik Sheth and Chris Rowen for their inspiration to write this paper.

8. REFERENCES

[1] R. Gonzalez. Xtensa: A configurable and extensible processor. *IEEE Micro*, March 2000.

[2] Farid N. Najm. A survey of power estimation techniques in vlsi circuits. In *Design Automation Conference*, 1994.

[3] S. Narayan and D. Gajski. Area and performance estimation from system-level specifications. Technical report, Dept of Information and Computer Science, UC Irvine, 1992.

[4] Tensilica. www.tensilica.com.

[5] V. Tiwari, S. Malik, and A. Wolfe. Power analysis of embedded software: A first step towards software power minimization. Technical report, Dept of Elect. Engg, Princeton University, April 1994.