Designing at the Optimal Compression Level

Chris Allsup
Synopsys, Inc., 700 E. Middlefield Road, Mountain View, CA 94043, (650) 584-5000
allsup@synopsys.com

Abstract

We present a systematic framework and methodology for calculating the scan compression level that minimizes test costs for any design. We first describe how to extract area parameters specific to a given process technology library and design rule set, and use these to easily and accurately predict die size as a function of compression level for designs containing many physical partitions with embedded compression logic. We next describe the relationship between test execution cost and silicon area overhead cost of compression, and how the cost equations can be used to calculate the optimal compression level. We then present efficient techniques for estimating the relevant ATPG parameters used in the analysis. Finally, we present detailed measurements of die size versus compression taken from industrial designs, and use the data to validate the cost savings advantage of designing at the optimal compression level.

1. Introduction

Advances in design automation technology have led to the widespread adoption of scan compression as a means to lower the costs of testing system-on-chip (SoC) designs. Scan compression lowers costs by reducing the test pattern data volume sufficiently to allow other types of tests to be loaded into tester memory, and it is no coincidence that its "mainstreaming" coincides with the widespread adoption of pattern-rich transition delay ATPG needed to test subtle defects associated with nanometer manufacturing processes. Semiconductor firms also benefit when scan compression succeeds in reducing the time spent testing production parts (i.e., the "test application time"), particularly in a high-volume production setting.

Designers therefore must take into consideration both the test pattern volume reduction benefit and the test time reduction benefit when implementing compression on-chip. Specifically, when considering requirements driven by full-scale production testing, designers should meet two objectives stemming from these dual cost benefits. The first objective is to add enough compression to permit a single download of all test patterns into memory to avoid the need to halt pattern testing midway through the test program to perform a vector reload of the remaining patterns. The second objective is to add enough compression to significantly reduce test application time and therefore test execution cost.

Decisions regarding how much compression to implement, however, typically are not based on economic principles and instead rely on simple rules-of-thumb or recommendations from EDA tool providers. This is because a methodology for determining how much compression to implement for any given design is lacking, although some progress was made by an economic theory of scan compression [1] that quantified the potential savings from test volume reduction and test time reduction, as well as the costs. According to the theory, once enough compression is applied to load all the test patterns into tester memory, it may be possible to reduce costs further by adding even more compression to reduce the test application time. But increasing

compression *ad infinitum* ignores important side-effects such as pattern inflation and the silicon area overhead cost of compression. Once these effects are accounted for, it is possible to calculate the compression level that achieves a balance between savings from further test time reduction and increased area overhead cost.

The absence of a straightforward, reliable approach for calculating the "optimal" compression level has made it difficult for designers to make economically-viable decisions regarding how much compression to implement for any given design. For example, what is the relationship between die size and compression level? The designer can find out by measuring total circuit area *after* physical implementation is complete—both with and without some nominal amount of compression inserted into the design. But the designer has little incentive to re-implement the design at this late stage in the flow just to optimize the compression. Moreover, even if the relationship between test execution time and silicon area overhead of compression were known, how does one determine, in the most efficient manner possible, the ATPG parameters needed for the cost analysis?

This paper focuses on these practical considerations by describing a systematic framework and methodology for calculating the optimal compression level λ that is easy for the designer to apply. Section 2 describes how to extract the area parameters to determine the relationship between die size and compression level. Once these area parameters are known for a given process technology library, compression tool, and design rule set, they can be used to easily predict the impact of compression on die size for any design that shares the same library/tool flow, as described in Section 3. Section 4 describes the effect of compression on test costs and how the cost equations can be used to calculate λ using techniques that minimize the amount of effort spent gathering ATPG data. Section 5 presents detailed measurements of die size for designs containing varying amounts of compression, implemented down to the physical level. We use this data to validate the methodology and substantiate the cost savings advantage of designing at the optimal compression level.

2. Extracting Area Parameters

The easiest method [2] for estimating the area impact of compression uses gate count as a proxy for area. The optimal compression level λ is calculated from measurements of a single parameter γ that represents the fractional increase in gate count per unit increase in compression. The main limitation of this approach is that it does not account for wire routing congestion, which can be significant enough at higher compression levels as to contribute to more than just a linear increase in area. Underestimating the area impact of compression leads to an inflated calculation of λ. To avoid this, we must formulate an expression for die size versus compression that predicts nonlinear area increase using *area parameters* extracted from area measurements of post-detailed-route designs containing compression.

We begin by measuring die size at different compression levels for a design with C scan channels in which the compression logic is inserted at the top level. Area measurements are made after performing logic synthesis, placement, and detailed route. We then perform a least-squares curve-fit of the data to determine the coefficients K_j of the quadratic that describes the die size A as a function of the compression level x:

$$A(x) = K_0 + K_1 x + K_2 x^2 \quad (1)$$

The compression level x is the ratio of the number of internal scan chains to the number of scan channels (the terms "compression ratio" and "compression level" will be used interchangeably throughout this discussion). Note that not all the coefficients extracted from the measured data need be positive. Once the coefficients are determined, the area parameters are calculated using:

$$A_F = K_0 - A_0, \quad A_\gamma = \frac{K_1}{C}, \quad A_\zeta = \frac{K_2}{C} \tag{2}$$

where A_F is the area added independent of compression level, A_γ is the linear area added per scan chain, A_ζ is the nonlinear area added per scan chain, and A_0 is the die size of the design implemented only with scan, not compression. The area parameters A_F, A_γ, and A_ζ are independent of design size, and the variation for each is small for a given manufacturing process technology, physical design rule set, and compression tool. Figure 1 shows how the area parameters are derived based on a least-squares curve-fit of area measurements for a design with C=24 scan channels and area without compression A_0=150,161,000 μm^2. The area parameters calculated from (2) are: A_F=10^7 μm^2, A_γ=25 μm^2, A_ζ=150 μm^2.

Figure 1. Extraction of K_j based on a least-squares curve-fit of measured die size data with C=24.

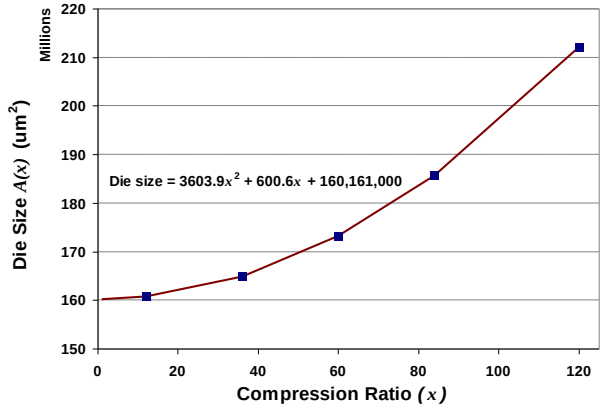
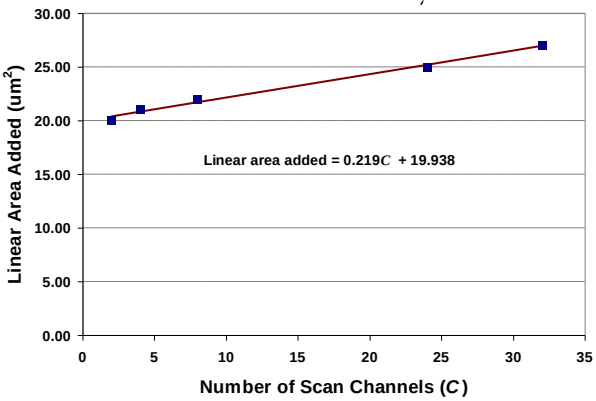


Figure 2. Derivation of $\overline{A}_\gamma(C)$ based on least-square fit of measured data for $\overline{A}_\gamma$ versus C.



For a given design technology library, the area parameters should be very similar for designs that share the same scan channel number, C. To account for slight variations, many measurements for the parameters should be taken across multiple designs that share the same number of scan channels and the same design technology library. The statistical average for each parameter, henceforth designated $\overline{A}_F$, $\overline{A}_\gamma$, and $\overline{A}_\zeta$, ensures the most predictable results. Even so, the area parameters $\overline{A}_F$, $\overline{A}_\gamma$, and $\overline{A}_\zeta$ actually increase slightly as a function of the number of scan channels C in a design because of wire routing congestion. To account for this effect, one should measure the parameters over a range of scan channel values to obtain $\overline{A}_F(C)$, $\overline{A}_\gamma(C)$, and $\overline{A}_\zeta(C)$. Once these measurements are made, a least-squares fit of the data for each parameter is performed to arrive at parameter values valid for any number of scan channels within the measurement range. Figure 2 shows how the *area parameter function* for linear area added per scan chain $\overline{A}_\gamma(C)$ = 0.219C + 19.938 μm^2 is derived based on a least-squares fit of measured $\overline{A}_\gamma$ versus C.

We can now predict the area impact of compression on a given design by expressing the relationship between die size and compression ratio as a quadratic in which die size $A(x)$ increases both linearly and nonlinearly according to:

$$A(x) = A_0 + A_F + C\left(A_\gamma x + A_\zeta x^2\right) \quad x > 1 \tag{3}$$

A more accurate prediction of die size can be made by augmenting this formula to include the area parameter functions so that we account for the average values of the area parameters across a continuous range of scan channel values:

$$A(x) = A_0 + \overline{A}_F(C) + C\left(\overline{A}_\gamma(C)x + \overline{A}_\zeta(C)x^2\right) \quad x_{min} \le x \le x_{max}, \quad C_{min} \le C \le C_{max} \tag{4}$$

where x_{min}, x_{max} and C_{min}, C_{max} are the minimum and maximum compression ratios and scan channel values, respectively, valid for the analysis. Strictly speaking, the least-squares curve-fits are valid only within the data measurement domains ($12 \le x \le 120$ and $2 \le C \le 32$ for the data in Figures 1 and 2), though it may be necessary to extend these domains in some situations. For example, the red curve defining *A(x)* in Figure 1 extends down to the compression level just above x=1 to represent the case in which λ is less than the minimum valid compression level, x_{min}=12.

3. Predicting Die Size versus Compression for Industrial Designs

Equation (4) can be used to accurately predict die size as a function of compression once the area parameter functions $\overline{A}_F(C), \overline{A}_\gamma(C),$ and $\overline{A}_\zeta(C)$ have been determined. However, the formula is somewhat limited in scope because it only applies to designs for which compression has been inserted "flat" at the top level. To minimize the amount of interconnect between physical partitions that could contribute to wire routing congestion, it is often advantageous to separately embed all compression logic, including both compressor and decompressor circuits, inside each partition so that there is no sharing of compression logic or scan I/O signals between partitions. Many large SoC designs today employ a "hybrid" approach that embeds compression logic in some, but not all partitions, with the remaining partitions sharing a common compressor/decompressor circuit. In this section we show how to apply the area parameter functions derived in the preceding section to predict the area impact of compression for this more general design scenario.

Figure 3 depicts a design with C scan channels and die size A_0 without compression. We wish to embed compression logic inside the two white partitions (designated "1" and "2"), *and* insert compression logic at the top level (designated "3"), shaded in gray. The "top level" in this context may contain other physical partitions that will not have embedded compression logic.

Figure 3. Conceptual diagram of a design without compression with area A_0. Compression circuits will be added to partitions 1 and 2 plus the top level.

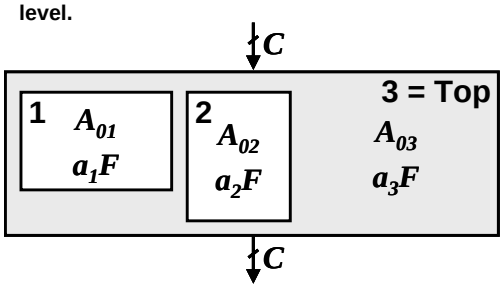
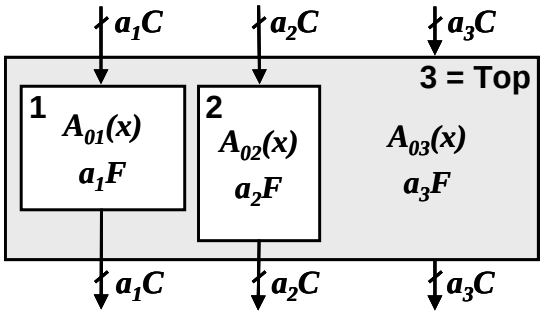


Figure 4. Compression increases the area of the design depicted in Figure 3 from A_0 to $A(x)$.



Each partition j has size A_{0j}, and A_{03} is the die size excluding the other partitions, equivalent to $A_0-(A_{01}+A_{02})$. The numbers a_jF represent the number of flops in each partition, where a_j is the fraction of the total number of scan flops in the design, F, in each partition. To balance the scan chains, we need to split the scan channels consisting of C scan I/O pairs into three channel sets that separately feed into each partition. The number of scan I/O pairs C_j in each channel should be chosen so that the scan chain depth is the same for all the partitions:

$$\text{Scan chain depth} = \frac{F}{C} = \frac{a_1F}{C_1} = \frac{a_2F}{C_2} = \frac{a_3F}{C_3}, \quad a_1 + a_2 + a_3 = 1 \tag{5}$$

which is satisfied if $C_1=a_1C$, $C_2=a_2C$, $C_3=a_3C$, as depicted in Figure 4. If we replace the variable C in (4) with these relationships, the area of each partition as a function of compression, $A_{0j}(x)$, is:

$$A_{0j}(x) = A_{0j} + \overline{A}_F(a_jC) + a_jC\left(\overline{A}_\gamma(a_jC)x + \overline{A}_\zeta(a_jC)x^2\right) \tag{6}$$

The total die size as a function of compression, $A(x)$, is the sum of the partitions $A_{0j}(x)$:

$$\begin{aligned}
&A(x) = A_{01}(x) + A_{02}(x) + A_{03}(x) \\
&= A_{01} + \overline{A}_F(a_1C) + a_1C\left(\overline{A}_\gamma(a_1C)x + \overline{A}_\zeta(a_1C)x^2\right) \\
&+ A_{02} + \overline{A}_F(a_2C) + a_2C\left(\overline{A}_\gamma(a_2C)x + \overline{A}_\zeta(a_2C)x^2\right) \\
&+ A_0 - A_{01} - A_{02} + \overline{A}_F(a_3C) + a_3C\left(\overline{A}_\gamma(a_3C)x + \overline{A}_\zeta(a_3C)x^2\right) \\
&= A_0 + \left(\overline{A}_F(a_1C) + \overline{A}_F(a_2C) + \overline{A}_F(a_3C)\right) + \left(a_1C\overline{A}_\gamma(a_1C) + a_2C\overline{A}_\gamma(a_2C) + a_3C\overline{A}_\gamma(a_3C)\right)x \\
&+ \left(a_1C\overline{A}_\zeta(a_1C) + a_2C\overline{A}_\zeta(a_2C) + a_3C\overline{A}_\zeta(a_3C)\right)x^2
\end{aligned} \tag{7}$$

We can extend this analysis to describe die size as a function of compression for a design containing an arbitrary number of partitions with embedded compression logic, n, one of which may represent the top-level:

$$\begin{aligned}
&A(x) = A_0 + \sum_{j=1}^{n}\overline{A}_F(a_jC) + \sum_{j=1}^{n}\overline{A}_\gamma(a_jC)a_jCx + \sum_{j=1}^{n}\overline{A}_\zeta(a_jC)a_jCx^2 \\
&x_{min} \le x \le x_{max}, \quad C_{min} \le \min(a_1C,\ldots,a_nC), \quad \max(a_1C,\ldots,a_nC) \le C_{max}
\end{aligned} \tag{8}$$

We assume that any increase in silicon area due to compression increases die size by the same amount according to (8).

Notice that once the area parameter functions are determined for a specific compression tool, technology library, and design rule set, the only measurement needed for a given design is the area of the design without compression, A_0. Also notice that the expression for $A(x)$ in (8) can be used for other compression implementation schemes. For example, when there are many embedded compression blocks but few scan pins available, it may be desirable to use the same pins to test each block in sequence. In this scenario, a_j=1in the expression for $A(x)$.

4. Determining the Optimal Compression Level

Our analysis assumes the designer wants to implement compression at the level $x=\lambda$ that minimizes test costs so long as this level is sufficient to load all patterns into tester memory. In this section we show how to determine this optimal compression level once the coefficients in the die size function (8) are known. We begin by examining how compression affects the costs of test.

4.1 Effect of Compression on Test Costs

Compression can be divided into two phases [1]: a test data volume reduction (TDVR) phase and a test application time reduction (TATR) phase. In the TDVR phase, every unit increase in compression ratio reduces the pattern volume more to allow "room" for extra patterns in memory that must be tested, and this reduction in test data volume exactly offsets any potential reduction in test time. The range of compression levels in the TDVR phase extends up to the compression level x_c, which is the amount of compression needed to fit all P_c patterns in the ATPG program into the amount of tester memory M allocated for digital stimulus and response patterns [1]:

$$x_c = \frac{1}{(P_0 / P_c) - \varepsilon}, \quad \varepsilon < P_0 / P_c \tag{9}$$

where ε represents the fractional increase in pattern count per unit increase in compression ratio due to pattern inflation from compression. A single parameter can be used to describe pattern inflation because, for most designs and most compression architectures, patterns increase linearly across a wide range of compression levels. P_0 is the number of patterns that can be loaded into tester memory without compression and is equal to $M/(3F)$, where F is the number of scan flops in the design. The coefficient "3" represents one scan stimulus bit and two response bits: one is the response bit itself and the other a mask or measure bit needed to determine if the response bit should be compared or not.

In the TATR phase, the range of compression levels above x_c, all patterns are loaded into tester memory, so there are *potential* cost savings from test time reduction. Our goal, therefore, is to determine the compression level $\lambda \geq x_c$ that minimizes test costs in the TATR phase. The two costs most sensitive to compression level in this phase are the silicon area overhead cost of compression, $C_{silicon}$, and the test execution cost, C_{exec}, which is proportional to the time it takes to apply the patterns on the tester. Both costs are expressed in terms of dollars per good die.

Compression *increases* the silicon area overhead cost of compression, $C_{silicon}$, according to [1, 3]:

$$C_{silicon}(x) = C_s\left(\frac{A(x)}{Y(x)} - \frac{A_0}{Y_0}\right) \quad x > 1 \tag{10}$$

C_s is the silicon area cost multiplier (\$/μm^2), A_0 is the die size without compression (μm^2), and Y_0 is the expected manufactured yield without compression. The yield $Y(x)$ is a user-defined function of the manufacturing defect density D (defects/μm^2) and the die size with compression $A(x)$ defined in (8). Yield estimates are needed to arrive at cost per good die. For this analysis, we will use the exponential yield equation [4] $Y(x)=1/(1+A(x)D)$ to predict the optimal compression level.

Compression *decreases* the test execution cost, C_{exec}, according to [1]:

$$C_{exec}(x) = \frac{R_{act}\alpha_0}{Y(x)\mathrm{f}_{test}}\left(\frac{F}{Cx}\right)P'_c(x) \quad x > 1 \qquad (11)$$

where $\alpha_0 = Y_0 + \beta_{fail}(1 - Y_0)$

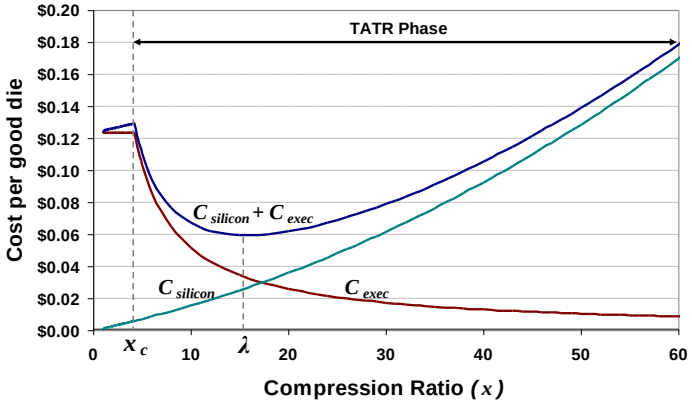
assuming the internal scan chains are well-balanced (else *F/C* is replaced by the number of flops in the longest scan chain). R_{act} is the cost of active testers ($/sec) and f_{test} is the tester scan shift frequency. The multiplier α_0 is used to account for a slight decrease on average in the test execution time due to less time spent testing failing die ($Y(x) \le \alpha_0 \le 1$). α_0 depends on β_{fail}, the percentage of good die test time required on average to test a defective die [3]. $P'_c(x)$ is the inflated pattern count of the test program reflecting a greater number of test patterns needed to achieve the same fault coverage as P_c. Since pattern inflation for most designs is linear over a wide range of compression levels, the inflated pattern count in (11) can be described as a linear function of the pattern inflation rate ε [1]:

$$P'_c(x) = P_c(1 + \varepsilon x) \quad x > 1 \qquad (12)$$

4.2 Calculation of Lambda

The optimal compression level in the TATR phase is the compression ratio $x=\lambda$ that minimizes the sum of the silicon area overhead cost of compression, $C_{silicon}(x)$, and the test execution cost, $C_{exec}(x)$, as shown in the example of Figure 5.

Figure 5. Test costs versus compression.



At compression levels below λ the incremental cost saving from test time reduction exceeds the incremental area overhead cost of compression, whereas above λ the incremental area overhead cost of compression exceeds the incremental cost saving from test time reduction. The optimal compression level occurs at the ratio where the rate of increase in the silicon area overhead cost of compression equals the rate of decrease in the test execution cost:

$$\frac{d}{dx}C_{silicon} = -\frac{d}{dx}C_{exec} \quad x > x_c \qquad (13)$$

To solve the equivalency of (13), we need a reasonable estimate of the defect density *D* for the manufacturing process. This can be derived from the exponential yield equation $Y=1/(1+AD)$ if the yield *Y* is known for a design of die size *A* manufactured using the same process technology and design rules as the targeted design:

$$D \cong \frac{1-Y}{AY} \tag{14}$$

Now let ω_j represent the coefficients of the quadratic defined in (8) so that $A(x) = \omega_0 + \omega_1 x + \omega_2 x^2$. We first substitute the expression for die size with compression, $A(x)$, into the expression for yield with compression, $Y(x)=1/(1+A(x)D)$. We then substitute $Y(x)$ into the cost formula $C_{exec}(x)$ in (11), and into the cost formula for $C_{silicon}(x)$ in (10) along with the expression for expected yield without compression, $Y_0=1/(1+A_0D)$. Solving (13) leads to the following equality:

$$m_5 x^5 + m_4 x^4 + m_3 x^3 + m_2 x^2 + m_1 x + m_0 = 0$$

where:

$$\begin{aligned} m_5 &= 4C_s D\omega_2^2 \\ m_4 &= 6C_s D\omega_1\omega_2 \\ m_3 &= 2C_s\left(\omega_2 + D\left(2\omega_0\omega_2 + \omega_1^2\right)\right) + 2\kappa\alpha D\omega_2 \\ m_2 &= C_s\left(\omega_1 + 2D\omega_0\omega_1\right) + \kappa D\omega_2 + \kappa\alpha D\omega_1 \\ m_1 &= 0 \\ m_0 &= -\kappa\left(1 + D\omega_0\right) \end{aligned} \tag{15}$$

and $$\kappa = \frac{R_{act}\alpha_0}{\mathrm{f}_{test}}\left(\frac{F}{C}\right)P_c$$

The expression for κ is a convenient notation but also represents the test execution cost without compression for each manufactured die, assuming the internal scan chains are well-balanced. Numerical techniques such as the Newton-Raphson method can be used to find the valid root for the fifth-degree polynomial in (15). Alternatively, the cost minimum can be found by tabulating the sum of cost formulas (10) and (11) versus compression.

4.3 Techniques for Estimating the ATPG Parameters

Values for the ATPG pattern count P_c and the pattern inflation rate ε are needed to solve (15). We now discuss techniques for estimating these ATPG parameters. We first run ATPG to completion on the design without compression to determine the non-inflated pattern count P_c. We next run ATPG to completion on the design at the maximum compression level x_{max} valid for the analysis (as defined at the end of Section 2) to determine the inflated pattern count $P'_c(x_{max})$. Once we have obtained both the complete pattern counts P_c and $P'_c(x_{max})$, the linear expression for inflated pattern count in (12) is used to derive the pattern inflation rate:

$$\varepsilon = \frac{P'_c(x_{max}) - P_c}{x_{max} P_c} \quad \varepsilon < P_0 / P_c \tag{16}$$

The inequality in (16) is one of the criteria for linearity and stems from the requirement that the denominator of equation (9) must be positive. For the analysis to be valid, the pattern inflation rate must be less than the ratio of the number of patterns that can be loaded into tester memory without compression, $P_0 = M/(3F)$, to the number of patterns in the non-inflated pattern set, P_c

We now have the parameters P_c and ε needed to calculate the optimal compression level λ in the range $x_c \le \lambda \le x_{max}$, where x_c is given by (9). Because the equality in (13) is valid only above x_c, any roots obtained by solving (15) that are less than x_c should be discarded. Also note that if the

ATPG pattern count P_c is high relative to the amount of available tester memory M, then a relatively high compression level x_c will be required just to load all the patterns into memory. In some situations, this could mean that there is no cost minimum above x_c and that the optimal compression level exists *below* x_c. Although methods to determine λ in the TDVR phase have been proposed [2], this analysis assumes the designer will implement compression at x_c in this scenario, as stated in the assumption at the beginning of this section.

Is there another way to estimate the ATPG parameters that requires fewer scan insertion/ATPG iterations? The answer is a qualified "yes." Inspection of (15) reveals that the pattern inflation rate ε appears only in the m_2 and m_3 coefficients, in the last term of each coefficient. Each of these terms has a very small contribution to the magnitude of the coefficients because $C_s >> \kappa\varepsilon D$ in most, perhaps all situations. This implies that *the optimal compression level is virtually independent of the pattern inflation rate ε in the TATR phase*, and calculations of λ using zero and nonzero ε lead to identical results. This behavior occurs because pattern inflation increases the magnitude of x_c while "flattening" the test execution time versus compression curve. Figure 5 shows the test execution cost curve, proportional to the test execution time, declining by a factor of $(x_c/x)(1+\varepsilon x)$. The net effect is no change to the optimal compression level even though pattern inflation increases total cost.

In light of these observations, a simpler approach to finding λ is feasible. As before, we run ATPG to completion on the design without compression to determine the non-inflated pattern count P_c. The solution to (15) is then arrived at using $\varepsilon=0$ to calculate λ. This efficient technique requires only one preliminary ATPG run before implementing compression at the optimal level. A drawback of this approach is that the designer has no way of knowing beforehand if the calculated λ exceeds x_c since the pattern inflation rate was not determined. This will not be an issue if the non-inflated pattern count P_c is low relative to the amount of available tester memory so that x_c is not large (referring to (9), this is true if $3FP_c\,\varepsilon << M$). Still, the designer should perform a simple "sanity check" calculation of x_c from (9). The formula to estimate ε is the same as (16) but now the calculation is based on ATPG data already generated from the λ-compressed design:

$$\varepsilon = \frac{P_c'(\lambda) - P_c}{\lambda P_c} \quad \varepsilon < P_0 / P_c \tag{17}$$

We note that if adding even a relatively small amount of compression to the design results in a significant step-increase in pattern count relative to ATPG without compression, then the calculation of ε using (17) may over-estimate the actual pattern inflation rate and thereby over-estimate x_c. This should not present a problem, however, as long as $\varepsilon < P_0/P_c$, since the calculation is needed only to confirm that our prior choice of λ exceeds x_c.

In summary, the following steps are performed to determine the optimal compression level for a design:

1. Extract the area parameter functions for the physical design technology library as described in Section 2.
2. Measure the die size without compression A_0 and determine the number of scan channels associated with each partition as described in Section 3. This data is used with the area parameter functions to calculate the coefficients in the die size function (8).

3. Obtain estimates for the design-, tester-, and process-specific parameters, including the cost infrastructure parameters, summarized in Table 1.
4. Run ATPG on the design without compression to determine the pattern count P_c.
5. Calculate the coefficients in (15) assuming zero pattern inflation ($\varepsilon=0$). λ is the solution to the equality in the valid compression range, $x_{min} \leq x \leq x_{max}$.
6. Run ATPG on the design at compression level λ.
7. Use (17) to calculate ε based on the pattern count numbers P_c and $P'_c(\lambda)$ from the ATPG runs in steps 4 and 6; determine x_c using (9) to ensure that $\lambda \geq x_c$.

Table 1. Parameters affecting calculation of optimal compression level.

Class	Parameter	Description
Design-Specific	F	Number of scan flops
	A_0	Die size without compression (μm^2)
Tester-Specific	f_{test}	Tester scan shift frequency (Hz)
	M	Available tester memory
	β_{fail}	Average % good die test time needed to test defective die
Process-Specific	D	Defect density (defects/μm^2)
DFT-Specific	P_c	Non-inflated ATPG pattern count
	C	Number of scan channels
Cost Infrastructure	C_s	Silicon area cost multiplier ($/$\mu m^2$)
	R_{act}	Cost of active testers ($/sec)

5. Measurements and Analysis

5.1 Measurements of Circuit Area versus Compression

Area parameters were extracted for two different technology libraries using the techniques described in Section 2. For each industrial design associated with a given library[1], compression was synthesized flat at the top level and measurements of total circuit area of the fully-routed design were obtained for compression ratios x = 1, 12, 36, 60, 84, 120 (x=1 represents the scan-only case). Each design was implemented with two different scan channel configurations: 4 scan I/O pairs and 8 scan I/O pairs. Least-squares curve-fits of the measured data were used to extract A_F, A_γ, and A_ζ for each design at C=4 and C=8, and these were averaged to obtain $\overline{A}_F$, $\overline{A}_\gamma$, and $\overline{A}_\zeta$. Table 2 displays both the area parameters and area parameter functions $\overline{A}_F(C), \overline{A}_\gamma(C)$, and $\overline{A}_\zeta(C)$ in μm^2. The slopes (m) and intercepts (b) of the area parameter functions are needed to estimate the area overhead of compression for designs containing an intermediate number of scan channels.

Figures 6 and 7 predict the expected area impact of compression for designs with different scan channel configurations that use the two technology libraries from Table 2. As we might expect, less area is added for designs that use the smaller process geometry. In all cases more scan channels increase the area of compression above nominal compression levels—a conclusion obvious from the graphs, though less obvious inspecting the parameters only.

[1] Measurements were based on one 90nm design and four 65nm designs.

Table 2. Area parameters extracted for two technology libraries.

Library	C	$\overline{A}_F$	$\overline{A}_\gamma$	$\overline{A}_\zeta$		$\overline{A}_F(C)$	$\overline{A}_\gamma(C)$	$\overline{A}_\zeta(C)$
90nm	4	16648	103.0	0.77	m	-2342	-7.0	-0.03
	8	7282	75.2	0.66	b	26015	130.9	0.87
65nm	4	311	21.8	-0.00	m	140	-3.7	0.03
	8	872	6.9	0.12	b	-249	36.8	-0.12

Figure 6. Average area added vs. compression for 90nm designs.

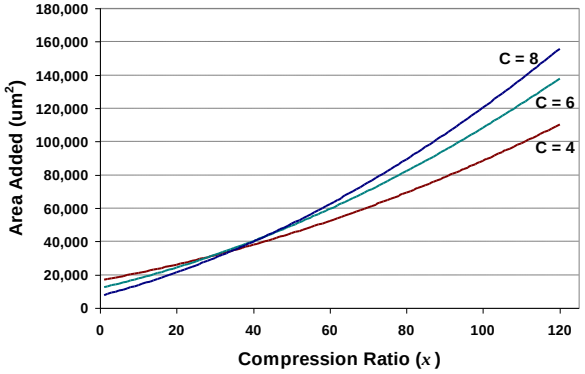
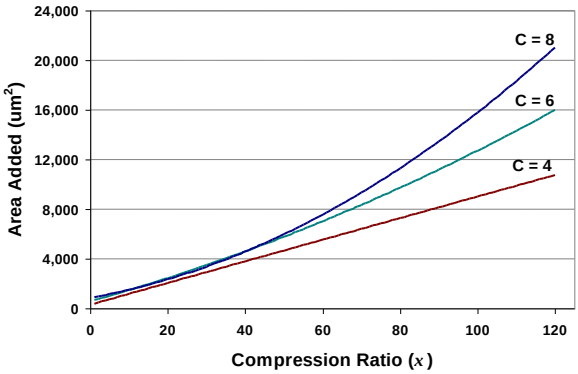


Figure 7. Average area added vs. compression for 65nm designs.



5.2 Optimal Compression and Cost Savings Estimates

ATPG patterns were generated without compression and at the maximum measured compression level, x_{max}=120, for the two industrial designs, each implemented with C=8 scan channels. The area parameters from Table 2 were then used to calculate the optimal compression level. The following input parameters were shared among the designs: available memory $M = C \cdot 256$Mb per scan channel, β_{fail} = 50%, C_s = \$4.00/cm^2, R_{act} = \$0.06/sec. The other input parameters and the calculated results are displayed in Table 3. For both designs, the non-inflated pattern count P_c and the inflated pattern count P'_c each represents the sum of both stuck-at and transition delay pattern counts (the fault coverages, all ranging between 95.5% and 98.9%, are not shown). The last column reflects the difference in cost per good die between compression at the optimal level and no compression.

Table 3. Optimal compression level and cost savings calculated for two designs.

Design	F	A_0 (cm^2)	f_{test} (MHz)	D (cm^{-2})	P_c	C	λ	$P'_c(x_{max})$	P_0/P_c	ε	x_c	ΔCost
A (90nm)	35,028	0.03720	20.0	0.6	4078	8	36	5498	5.0	0.3%	< 1	\$0.051
B (65nm)	31,672	0.00526	50.0	0.8	4142	8	55	8148	5.5	0.8%	< 1	\$0.019

The calculation of λ is valid for each design because the estimated value of the pattern inflation rate ε from (16) is less than P_0/P_c, and because λ is greater than the value of x_c obtained from (9). However, it is apparent from the circuit areas and scan flop counts that these examples are not representative of the very large SoCs fabricated using these manufacturing processes. To evaluate the merits of designing at the optimal compression level for much larger designs, in Table 4 we have scaled up A_0 and F for both designs while using the same pattern count numbers as before.

The cost saving for design A indicates that savings can be substantial for designs larger than those that were used to extract the area parameters. For design B, λ is very high because at $x_c \approx$ 17 the scan chains are still very long (18,738 flops) and a large amount of additional compression is needed to reduce test execution time significantly more than this. Because the area overhead per scan chain is much lower at 65nm than at 90nm (see Table 2), the incremental cost of adding compression is also much lower and this increases the optimal compression level.

Table 4. Calculations for designs A and B, scaled to reflect higher gate counts.

Design	F	A_0 (cm^2)	f_{test} (MHz)	D (cm^{-2})	P_c	C	λ	$P'_c(x_{max})$	P_0/P_c	ε	x_c	ΔCost
A (90nm)	398,663	0.42300	20.0	0.6	4078	8	87	5498	44%	0.3%	2	$0.287
B (65nm)	2,548,314	0.42300	50.0	0.8	4142	8	>120	8148	7%	0.8%	17	?
B (65nm)	2,548,314	0.42300	50.0	0.8	4142	24	109	8148	20%	0.8%	5	$0.111

But a good estimate of λ for this scenario cannot be determined because the calculation exceeds the upper compression range x_{max}=120 of the circuit area measurements for the 65nm library. Although it is tempting to increase the number of scan channels to estimate λ, the result would still be invalid if we simply insert compression at the top level. This is because the die size prediction in (4) is valid only between the measured range of scan channels, $4 \leq C \leq 8$.

Suppose, however, we embed compression logic inside several partitions in design B, as described in Section 3. Assume, for example, there are now three equivalent partitions containing 2,548,314/3 = 849,438 scan flops with eight scan channels allocated to each partition. There are now C=24 scan channels and we can use (8) to predict the total area overhead within the valid measured range of scan channels. Under these conditions, shown in the bottom row of Table 4, λ=109 and x_c=5. The cost saving from compression, now the difference in cost between compression at the optimal level and compression at x_c, is $0.11 per good die. Had there been no increase in available tester memory, x_c would have been equal to 17 and increasing compression to the optimal level, while achieving the same total test cost, would have resulted in substantially lower incremental cost saving (ΔCost = $0.028).

5.3 Assumption of Perfect Yield

One characteristic not reflected in the area versus compression graphs of Figures 6 and 7 is the relative flatness of the cost curves in the region of λ for the design examples that were scaled to reflect higher gate counts. In each case, it is more descriptive to refer to the optimal compression level as the "optimal compression *range*" in which the difference in cost saving from the minimum varies by only a few percentage points. To illustrate, Table 5 shows how assuming the case of "perfect yield" (i.e., zero defect density D) increases the optimal compression levels for these scaled designs[2] yet results in only small percentage cost increases from implementing compression at these higher λ values.

The expected yields of the scaled designs A and B are 80% and 75%, respectively. But since λ is positively correlated with yield, assuming perfect yield increases λ by 11% and 14%. In spite of this increase, the impact on total test cost for each design is small due to flatness of its cost curve in the optimal compression range.

[2] Differences in λ for the un-scaled designs were negligible.

Table 5. Effect of perfect yield on optimal compression level and cost.

Design	F	C	λ	λ (D=0)	Cost Increase*
A (90nm)	398,663	8	87	97 (+11%)	+1.3%
B (65nm)	2,548,314	24	109	124 (+14%)	+0.7%

* Cost difference calculated from standard (imperfect yield) model for the two different values of λ.

This insight has important ramifications to the design flow. Of all the parameters in Table 1, the one most difficult to estimate is A_0, the die size without compression. This is due to the fact that for large designs, compression is inserted as part of the normal synthesis process so that die size measurements without compression, if not infeasible, are inconvenient to perform. But if we assume perfect yield, we effectively eliminate the A_0 parameter from the analysis and hence the requirement to measure die size without area. This is because the equality in (15), assuming zero defect density, reduces to the simplified expression:

$$2C_s\omega_2 x^3 + C_s\omega_1 x^2 = \frac{R_{act}}{\mathrm{f}_{test}}\left(\frac{F}{C}\right)P_c \tag{18}$$

where ω_1 and ω_2 are the coefficients of the quadratic defined in (8).

Does calculating λ using (18) typically lead to only a relatively small percentage increase in test cost compared with calculating the true-λ using (15)? To evaluate the merits of assuming perfect yield, a parametric analysis was performed on the scaled design A in which the following input variables were swept between the limits shown:

A_0 = {0.42339, 0.76211 cm^2}; D = {0.6, 1.0 defects/cm^2}; ε = {0.3%, 5.2%}; C_s = {\$2.00, \$6.00/cm^2}; R_{act} = {\$0.04, \$0.12/sec}; f_{test} = {20M, 100M Hz}.

For each input combination, the percentage increase in test cost was determined based on the difference between the optimal compression level calculated using (18) and using (15). The analysis results are shown in Table 6. For example, λ increased by more than 20% in 43% of the data set. The disparity in calculated λ values increased under low-yield conditions (combinations of relatively high defect density D and die size A_0). Despite these disparities, total cost increased by less than 4% in 99.3% of the data set, and never exceeded 5%.

Table 6. Parametric analysis of scaled Design A.

λ Increase	Percentage of Data Set	Cost Increase	Percentage of Data Set
> 15%	85.7%	> 3%	13.3%
> 20%	43.0%	> 4%	0.7%
> 25%	9.5%	> 5%	0.0%

6. Conclusions

This paper has presented a practical methodology for determining the compression level that maximizes cost savings. Area parameter functions associated with a given technology library can be extracted and die size coefficients calculated to accurately predict die size as a function of compression for any design sharing the same manufacturing process, compression tool, and

design rule set. Once the die size function is known, it is easy to calculate the optimal compression level based on design, tester, process, DFT, and cost infrastructure variables.

Circuit area measurements of fully-routed industrial designs showed how the methodology can be used to predict the area impact of compression and the optimal compression level for designs that share common technology libraries. It is important to note that not enough measurements were made to yield reliable estimates for the area parameters presented in Table 2. At most, only four designs per technology library (in the case of the 65nm library example) were used to estimate these parameters, whereas at least 20-30 samples are required to calculate statistically-sound average and variance metrics needed for industrial settings. Nonetheless, it is safe to say that adding compression increases circuit area, often nonlinearly, and that increasing the number of scan channels increases area, especially at higher compression levels.

The predictive accuracy of the method might be improved by introducing parameter *classes* that reflect the effect certain characteristics of a design have on the area parameter functions. For example, does an increase in the area overhead of compression always lead to the same increase in die size? If not, how does gate utilization achieved during place-and-route influence this relationship? An area parameter class that accounts for this effect might be beneficial, though it is unclear without further investigation that this level of accuracy is needed.

Parametric analysis of one of the scaled designs indicated that it is probably not necessary to estimate die size without compression for designs having reasonable yield because the solution to (18), which assumes perfect yield, lies within the optimal compression range such that the cost penalty of adding higher-than-optimal compression is negligible. Additional measurements and more extensive sensitivity analysis are needed to determine the conditions for which perfect yield leads to an estimate of the optimal compression level with an arbitrarily small cost penalty.

7. Acknowledgements

The author would like to thank Tammy Fernandes, Synopsys corporate applications engineer, and Pramod Notiyath, Synopsys corporate applications engineer, for their valuable assistance in preparing the design data.

8. References

1. C. Allsup, "The Economics of Implementing Scan Compression to Reduce Test Data Volume and Test Application Time," *Proc. Int'l Test Conf.*, Lecture 2.2, 2006.
2. C. Allsup, "Optimizing Compression in Scan-Based ATPG DFT Implementations," *Test & Measurement World*, March 2007.
3. S. Wei, P.K. Nag, R.D. Blanton, A. Gattiker and W. Maly, "To DFT or Not to DFT?" *Proc. Int'l Test Conf.*, pp. 557-566, 1997.
4. T.M Michalka, R.C. Varshney, J.D. Meindl, "A Discussion of Yield Modeling with Defect Clustering, Circuit Repair, and Circuit Redundancy," IEEE Transactions on Semiconductor Manufacturing, Vol. 3, No. 3, Aug. 1990, pp. 116-127.