Automated Yield Forecasting In a High Product Mix ASIC Facility

Duane Barber[a], Mark Giewont[a], Jeff Hanson[a], and Jun Shen[b]

[a]LSI Logic Corporation, 23400 NE Glisan St., Gresham, OR 97030

[b]LSI Logic Corporation, 3098 W Warren Ave., Fremont, CA 94539

ABSTRACT

Yield forecasting is a key component in running a successful semiconductor fab. It is also a significant challenge for facilities such as ASIC houses, which fabricate a wide range of devices using multiple technologies. Yield forecasting takes on increased significance in these environments, with new products introduced frequently and many products running only in small numbers. An accurate yield prediction system can greatly accelerate the process of identifying design bugs, test program issues and process integration problems. To this end, we have constructed a forecasting model geared for our ASIC manufacturing line. The model will accommodate an arbitrary number of design and/or process elements, each with an associated defectivity term. In addition, we have automated the generation of the yield forecast through passively linking to the already existing EDA design tools and scripts used by LSI Logic.

Once the model is constructed, an automated query engine can extract the design and process parameters for any requested device, insert the data into the forecasting model, and deliver the resulting yield prediction. The actual yield for any lot or group of lots may thus be compared to the forecast, greatly assisting yield enhancement activities. This is especially useful for prototype lots and low-volume devices, for which it eliminates a great deal of manual computation and searching of design files. Using the model in conjunction with the query engine, any deviations from expected yield performance are generated automatically, quickly and efficiently highlighting opportunities for improvement.

Keywords: yield, forecast, model, ASIC

1. INTRODUCTION

Semiconductor yield prediction models generally fall into one of two categories. The first class addresses the yield learning curve as a technology matures. These models may seek to predict the learning curve based on historical yield data[1], or may strive to link production parameters to final yield in order to more effectively drive the learning curve[2]. The second group of models is concerned with predicting the yield of specific products[3, 4]. These models may incorporate information such as die size and layer-specific yield loss, and are particularly useful for the design or introduction of a new device.

LSI Logic's Gresham, OR fab is similar to many ASIC fabs, with a high product mix and frequent introduction of new devices on existing technology nodes. In this environment, the ability to predict the yield of a new product is very useful. A yield prediction for a prototype lot can quickly separate products running as expected from those with design, test, or integration issues. To this end, LSI sought to develop a model which incorporated device-specific data into its yield predictions. However, some provision for tracking and driving yield learning over time was also desired. In addition, an automated yield prediction was needed, in order to avoid loading additional tasks onto existing human resources.

LSI's production environment lends itself well to automated yield forecasting. LSI design teams use EDA tools and scripts which already populate a central database with exactly the kind of product-specific data required for an ASIC-based yield model. Once the model is developed, it is incorporated into a query engine which probes the database to determine the relevant parameters for any given device. The query engine can run autonomously, or the model which drives it can be modified by a user to reflect the most up-to-date data. The resultant yield forecasting system blends and

Data Analysis and Modeling for Process Control II, edited by Iraj Emami,
Christopher P. Ausschnitt, Kenneth W. Tobin, Jr., Proceedings of SPIE Vol. 5755
(SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.600079

builds on concepts introduced in models developed elsewhere. It provides a useful check for existing products as well as new devices, and is flexible enough to accommodate new technology nodes as the industry advances.

2. MODEL BASICS

The most general form for a yield model is the negative binomial model[5], given by:

$$Y = (1 + AD_0 / \alpha)^{-\alpha} \tag{1}$$

where A is the chip area, D_0 is the average defect density, and α describes the degree of defect clustering. This general relationship will often reduce to one of the commonly used yield models, Poisson, Murphy, or Seeds, depending on the value of the cluster parameter (α)[6]. In particular, high values of α (8 or greater), indicating relatively little clustering of defects, cause the negative binomial model to approximate the Poisson model:

$$Y = e^{-AD_0} \tag{2}$$

Mid-range values of α (4.2 – 4.5) correspond to a moderate amount of defect clustering, for which case the negative binomial model closely matches the Murphy model:

$$Y = \left(\frac{1 - e^{-AD_0}}{AD_0} \right)^2 \tag{3}$$

High degrees of defect clustering are reflected in low α values (around 1), and result in the negative binomial model approximating the Seeds model:

$$Y = \frac{1}{1 + AD_0} \tag{4}$$

To select a specific yield model to use, we plotted yield versus chip area for several different devices. These devices were chosen to minimize the number of compounding factors which might alter the relationship between yield and defect density. All the codes were drawn from the same technology node, had the same number of metal layers, featured no optional high-defectivity layers, and utilized no high-density memory. The results are shown in Figure 1 below.

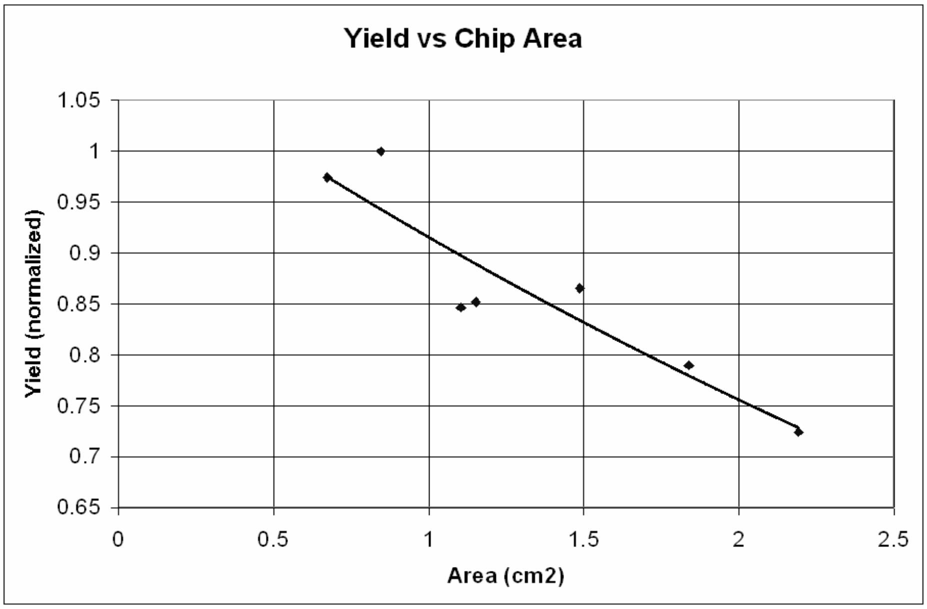

Figure 1: Yield versus Chip Area for a baseline process. The solid line shows a best-fit curve to the data using a Poisson model.

The Poisson model and the Murphy model yield equivalent fits to the data, while the Seeds model does not fit the data quite as well. For simplicity, we choose the Poisson model as the baseline relationship between yield and defect density. It should be noted that even with the limited sample of low-complexity devices used for this graph, there are still deviations from the model. This is not unexpected, since there still remain complexity differences between these chip codes. For example, the two devices falling significantly below the curve for the model both contain more medium-density memory (~ 10%) than do the other devices (less than 5%). The fit may be expected to improve when the model is expanded to account for the percentage of a chip devoted to memory. Nevertheless, the initial fit is good enough to proceed with this approach.

The fact that the Poisson and Murphy models match the actual data equally well, and the Seeds model only somewhat less well, indicates that our product of defect density and chip area (AD_0) is low enough that all three models provide generally adequate fits[6]. As stated previously, we will choose the Poisson model to incorporate into the forecasting system for ease of use. However, the manipulations we perform on the model may be applied to any yield model which describes yield in terms of die size and defect density.

3. MODEL EXPANSION

Once the basic model has been established, we can move on to incorporating different complexity factors which will impact the yield for a given chip size and defect density. One common tool is to note that the net yield can be expressed as a product of the yields of multiple independent components:

$$Y_{tot} = Y_1 \bullet Y_2 \bullet Y_3 \bullet .... \bullet Y_n \quad (5)$$

The component yield terms might be the yields from independent defects[3], the yields associated with different processing steps[7], or the yields for different blocks of circuitry within the device[4].

Some researchers have constructed extremely detailed models, including variables such as the number of n-type and p-type transistors and the total gate oxide area within each discrete portion of circuitry within a device[4]. This level of detail permits a very finely-tuned model to optimize the accuracy of the yield prediction. In our case, we are pursuing a yield forecasting system for an ASIC fab, with a large number of different devices running. It would be impractical to manually obtain this degree of detailed information for each product. Our approach is to automate the yield forecasting by linking the model to an already-existing database for LSI designs. We are therefore limited to parameters which are recorded in this database. It contains the chip area for each device, which is an absolute requirement for any yield forecasting model. It also includes other factors which are key yield determinants, so that we can perform some reasonable expansions of our basic model.

Empirically, one of the biggest factors influencing yield is the amount of high-density memory present in a chip. Our design database includes which memory cells are incorporated into each device, and also what fraction of the total chip area is occupied by each type of memory cell. We can therefore express our predicted yield as

$$Y_{tot} = Y_{\log ic} \bullet Y_{memory\_1} \bullet Y_{memory\_2} \bullet Y_{memory\_3} \quad (6)$$

$$= e^{-A_{\log ic} D_0} \bullet e^{-A_{memory\_1} D_1} \bullet e^{-A_{memory\_2} D_2} \bullet e^{-A_{memory\_3} D_3} \quad (7)$$

In this case, D_1, D_2, and D_3 reflect the fact that each high-density memory type will have a different effective defect density. The idea of effective defect density derives from earlier work in which researchers proposed describing yield in the form:

$$Y = e^{-\theta A D_0}, \quad (8)$$

where θ is used to describe the ratio of the critical area of one circuit versus another[8]. Various means for calculating critical areas of circuits have been put forth[8, 9, 10]. For our purposes, we will opt to rewrite this expression as

$$Y = e^{-A(D_0 \theta)} \quad (9)$$

$$= e^{-AD_{effective}} \quad (10)$$

$D_{effective}$ may then be determined empirically for each memory type by fitting the model to actual data, with the logic circuitry being used as the baseline for D_0. Another way of viewing this is that the effective defect density for a memory type is equal to the logic defect density scaled by a multiplier which accounts for the more demanding critical area of that memory design:

$$D_n = D_{\log ic} \bullet M_{memory\_n} \tag{11}$$

Up to this point we have referred to the effective defect density of the logic circuitry simply as D_0. However, just as the presence of high-density memory may give some chips a higher overall effective defect density, so too will certain factors give the logic on some chips a higher overall defect density than other chips. Again, these factors can be delineated in great detail to construct a very precise model[4]. But even within the restrictions of using only the information available in our existing design database, several important factors can still be obtained. The first such factor is the number of metal layers used on a chip. Other workers have noted that each layer carries an inherent contribution to overall defectivity[7], and we see this fact clearly in comparing devices with differing numbers of metal layers. While a majority of our devices use 5 metal layers, others may use 3, 4, or 6 metal layers. We therefore choose 3 metal layers as our baseline, and provide a defectivity "adder" for each additional metal layer. We find that the increased defectivity is comparable for each metal layer added beyond 3, so we now write the logic yield as:

$$Y_{\log ic} = e^{-A_{\log ic}(n_{met}D_{met}+D_0)}, \tag{12}$$

where n_{met} is the number of metal layers above 3, and where D_{met} is the adder assigned for an additional metal layer. It can be seen from the above expression that the adders for all metal layers present are in fact added together to produce a net additional defect density term which is added to D_0 to arrive at the overall effective defect density. In a similar manner, we have observed that higher yield loss is experienced in conjunction with the use of a MIMCAP layer and with the use of a flip-chip option, both of which are identifiable within our design database. We therefore provide adder terms for these factors as well, expanding the above expression to read:

$$Y_{\log ic} = e^{-A_{\log ic}(n_{met}D_{met}+D_{MIM}+D_{flip}+D_0)} \tag{13}$$

In this expression, D_{MIM} and D_{flip} are the adder values assigned to the use of MIMCAP and flip-chip options (and are set to zero if the corresponding option is not used for a given device).

We now have a reasonably comprehensive description of the logic yield in terms of parameters contained within our design database. And we had previously expressed the effective defect density for a memory block n as

$$D_n = D_{\log ic} \bullet M_{memory\_n} \tag{14}$$

Looking at our expression for the logic yield, we see that we can write the effective defect density for logic as

$$D_{\log ic} = (n_{met}D_{met} + D_{MIM} + D_{flip} + D_0) \tag{15}$$

Or, more simply, we can write

$$D_{\log ic} = Adders + D_0, \tag{16}$$

in which "Adders" is the sum of all the adder terms applicable for a given chip. The expression for the yield of memory block n then becomes:

$$Y_{memory\_n} = e^{-A_{memory\_n}(Adders+D_0)\bullet M_{memory\_n}} \tag{17}$$

At this point we have expanded expressions for both logic yield and memory yield in terms of the area of the circuitry, a defectivity scaling term for memory, defectivity adder terms for appropriate options on the chip, and an overall baseline defectivity D_0. After the scaling and adder values are assigned, the model can then be incorporated into the automated yield forecasting system. It is worth noting that this format offers a fair degree of flexibility in forecasting future yields. For example, the baseline defectivity term D_0 can be ramped down on a periodic basis in order to include anticipated yield improvements in long-range forecasts. Alternately, an individual term (such as D_{MIM}) can be ramped down or stepped down if it is expected that focused efforts will reduce the defectivity contribution from that particular source.

4. MODEL FITTING

Once the expression for yield has been expanded to include all significant and/or available data on factors contributing to yield fallout, the next step is to determine values for each of the defect adder and defectivity scaling terms included in the expanded model. The most straightforward method is to collect yield results for multiple devices with different terms in the yield prediction formula present or absent, and then fit the adder and scaling terms to the yield data. One option is to set up a matrix equation embodying the expanded yield expression, and then solving simultaneously for a large number of devices to obtain an optimized fit[11]. We take a somewhat simpler approach, consistent with the limited number of factors available from our design database. For each term needing a value assigned, we identify a number of devices for which all adder and/or scaling terms are the same except the one in question. This isolates the impact of the term in question, and lets us identify a best-fit value.

An example of this method is the determination of the value for a memory defectivity scaling term M_{memory_1}. For this evaluation, we identify products which do not use the MIMCAP or flip-chip options, and which have the same number of metal layers. The only variables are the chip size and the percentage of chip area occupied by high-density memory. Note that in this case we can write our yield expression as:

$$Y_{tot} = Y_{\log ic} \bullet Y_{memory\_1} \tag{18}$$

$$= e^{-A_{\log ic}(n_{met}D_{met}+D_0)} e^{-A_{memory\_1}(n_{met}D_{met}+D_0)\bullet M_{memory\_1}} \tag{19}$$

Noting that, for these devices with just one high-density memory design,

$$A_{memory\_1} = A \bullet (\%\_Memory\_1), \tag{20}$$

and similarly

$$A_{\log ic} = A \bullet (1-(\%\_Memory\_1)), \tag{21}$$

we see that the effective defect density for these products can be written as a linear function of the percentage of high-density memory content. The results are shown in Figure 2 below. Although there is clearly some deviation from the best-fit line, this nevertheless allows us to set an approximate value for M_{memory_1} which will deliver a reasonably close prediction for most products. As was the case in fitting a general yield model to our data, we expect an imperfect fit due to the limited number of yield- impacting parameters we are able to consider.

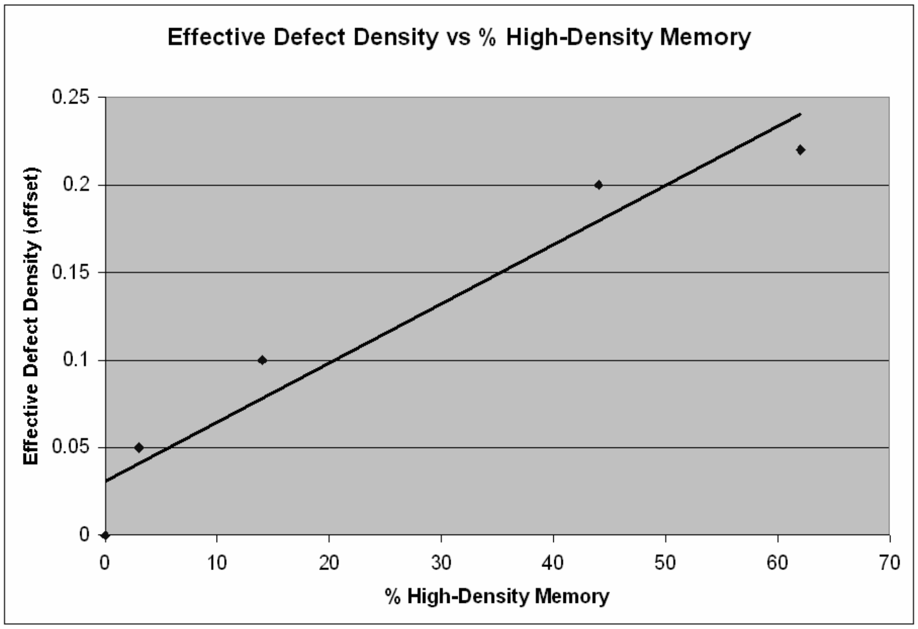

Figure 2: Effective defect density as a function of the percentage of die area occupied by high-density memory. The solid line shows a best-fit linear curve.

Another example is the determination of the value for the defectivity adder term associated with the MIMCAP option, D_{MIM}. One option is to identify devices with no high-density memory, which do not use the flip-chip option, and which have the same number of metal layers. The only variable aside from chip size is the presence or absence of the MIMCAP option. In fact, this lets us directly compare effective defect densities of the various products, since our effective defect density in this case will reduce to

$$D = D_{MIM} + (n_{met} D_{met} + D_0) \tag{22}$$

The results are shown in Figure 3 below. The drawback to this approach, as seen in Figure 3, is that most of our products which use the MIMCAP option also use high-density memory. This gives us relatively few data points with the MIMCAP option. Nevertheless, the data provide a reasonable estimate of the additional defectivity resulting from the MIMCAP option.

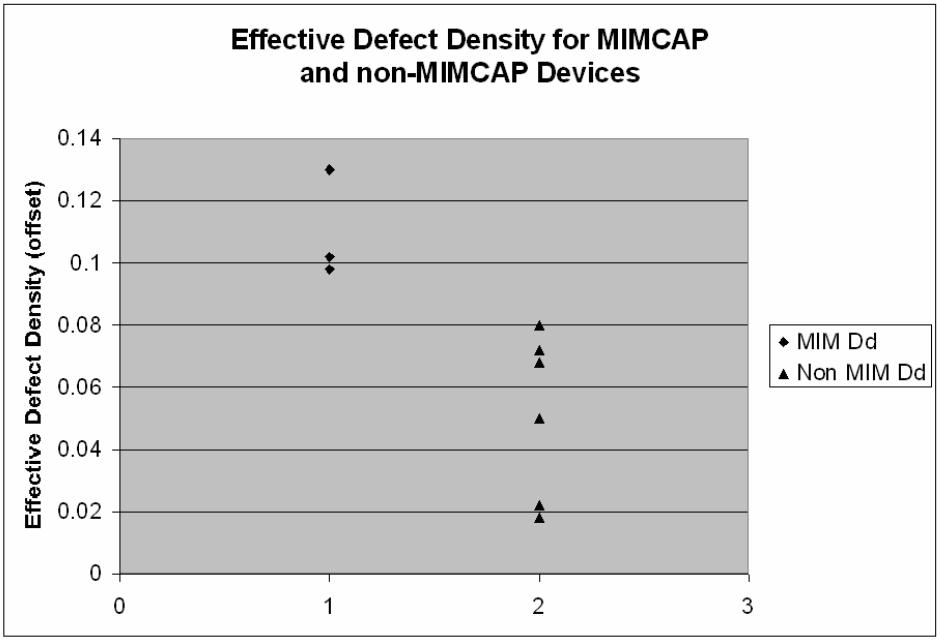

Figure 3: Effective defect density for devices both with and without the MIMCAP option.

A second approach for determining Add_{MIM} is to follow the method used to determine $M_{\mathrm{memory\_1}}$. In this case, we will again choose devices with no flip-chip option and which have the same number of metal layers, but we will now restrict ourselves to devices which use the MIMCAP option. Again we will write our yield expression as

$$Y_{tot} = Y_{\log ic} \bullet Y_{memory\_1} \tag{23}$$

$$= e^{-A_{\log ic}(n_{met}D_{met} + D_{MIM} + D_0)} e^{-A_{memory\_1}(n_{met}D_{met} + D_{MIM} + D_0) \bullet M_{memory\_1}} \tag{24}$$

As before, we note that both the logic and the memory areas can be expressed in terms of the total area and the memory percentage of total area. Once again, we see that the effective defect density can be written as a linear function of the percentage of high-density memory content. The results are shown in Figure 4 below. The fit is close enough to let us determine a reasonable value for the MIMCAP adder term, and to corroborate the MIMCAP value we determined by the first method given above.

When all the adder and scaling term values are assigned, the resulting model performance is shown in Figure 5 below. This chart plots the actual effective defect density as a function of the forecast effective defect density for products run within the last three months. While a few products deviate significantly from their predicted yields, most fall fairly close to the predicted values. This confirms that, despite the limited number of parameters available to be incorporated into the model, the resulting yield forecasting system is sufficiently accurate for our purposes.

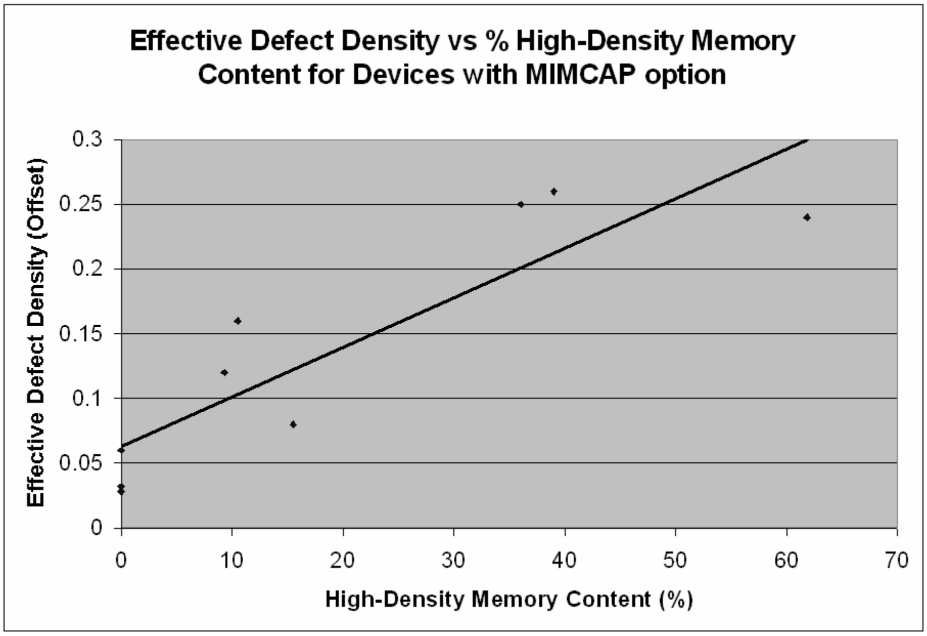

Figure 4: Effective defect density as a function of the percentage of die area occupied by high-density memory. The solid line shows a best-fit linear curve.

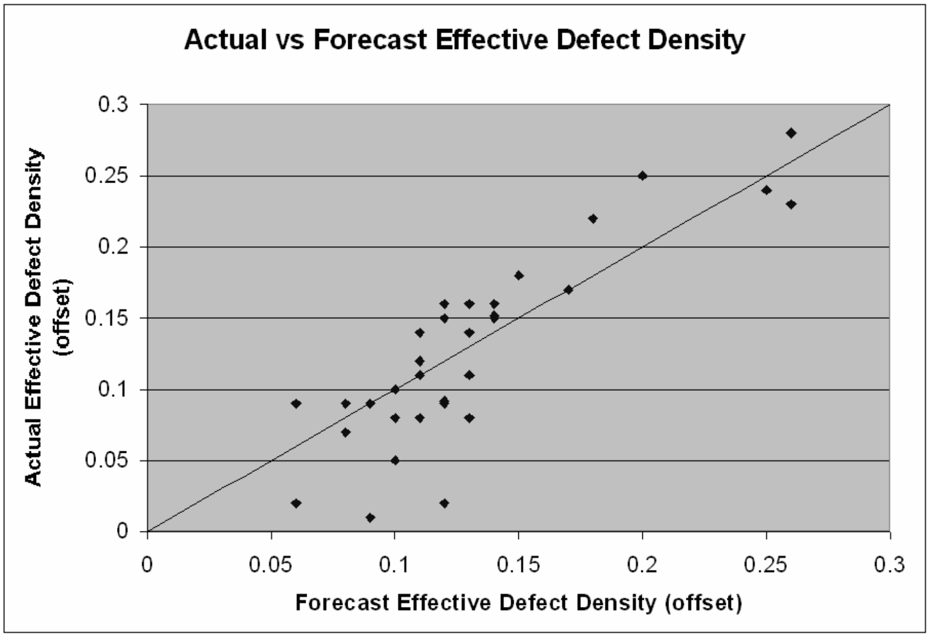

Figure 5: Actual effective defect density as a function of forecast effective defect density. The solid line represents prefect agreement between actual and forecast values.

5. FORECASTING SYSTEM APPLICATIONS

Once the model is developed and calibrated, we build it into an automated query engine. This software links to LSI's design database, so that a yield can be forecast for any device. In addition, it also links to the fab's database of tested wafers, and is routinely used to review wafer sort results. This combination means that as wafer sort results are reviewed, they can be immediately compared to the forecast yield performance. If a device shows a relatively low yield, but that yield is in agreement with the forecast, engineering resources will not be unnecessarily allocated to investigate the apparent low yield. Conversely, if a product yields significantly worse than the forecast, engineers will be alerted to investigate, even if the yield in and of itself is not glaringly poor.

An example of this application occurred last year. The fab introduced a new high-volume product, which consistently yielded an average of 4% below forecast. This discrepancy prompted an inquiry into possible processing, testing, or design issues. The investigating team found a timing issue with a particular scan test, as well as an improper state initialization prior to a leakage test. Once the issues with the test program were resolved, the device yielded within 0.5% of the yield forecast.

The system is also useful for financial forecasting. The finance team routinely needs to estimate revenue for an anticipated product mix and fab loading. They require reasonable projections of yields on high-running products to generate the fiscal projections. The yield forecasting system will provide current forecasts for any device, but it will also provide future yield estimates. To make this a useful ability, the baseline defectivity D_0 can be predicted on a month-by-month basis to reflect the learning curve for the technology as a whole. The system will then adjust the yield forecasts for individual products according to the improvements [presumably] in D_0 in conjunction with the adders and scaling terms applied to that device. In a similar manner, each adder value and each scaling term value can be predicted on a month-by-month basis. As an example of this, an intensive 6-month effort steadily reduced the defect contribution of the MIMCAP option by a factor of two. The yield forecasting system was able to reflect this ongoing improvement in the yield projections provided to the finance team.

Two additional features have been added to the yield forecasting system to enhance its utility. The first is that an adder term for product normalization has been inserted into the model. For high-running devices which consistently run with an offset from the predicted yield (positive or negative), this term allows the forecast to be brought into line with the actual performance. The product normalization term will accommodate products for which the forecast yield deviates from the actual yield due to factors intentionally excluded from the model, such as atypical critical areas or pad-limited designs. A second feature is that an adder term for excursions has been incorporated into the model. This term may be either device specific (for example, if a reticle defect is found) or technology-wide (for an event like a tool malfunction). In either case, once an excursion is identified and its impact and duration is understood, this term can be set appropriately to show the depressed yield forecast while the excursion holds sway, and then return to normal once it is past. These features provide improved projections for the finance team, and also improve the data evaluated after wafer sort by the yield team.

6. FORECASTING SYSTEM EVOLUTION

Once the automated yield forecasting system is established, it is not intended to be a static entity. As noted previously, the baseline defectivity D_0 and the individual adder and scaling terms can be adjusted over time to reflect baseline improvements or targeted improvements to specific defect sources. In addition, the excursion term allows the forecasting system to respond in real time to variations in fab performance and yield output. Beyond these functions, the model itself can be further augmented. As new technology options are introduced, new adder or scaling terms can be added to the model to reflect the impact of the new options. If a new technology node is introduced, the entire process of establishing and calibrating the model can be repeated at the new node.

Another example of the yield forecasting system changing over time is the case of the defectivity scaling factor (multiplier) for high-density memory. Figure 6 below shows the high-density memory multiplier for a single device plotted against the logic effective defect density, sampled at two different points in time. Each time sample includes hundreds of wafers subjected to extensive testing to independently determine the logic yield and the memory yield. Time A is sampled early in the life of this technology, when the defect density is still quite high. Time B is later, after defect densities have been reduced nearly by a factor of three. The data from both time periods show that as the logic effective defect density decreases, the memory multiplier increases. This is not surprising: since the high-density memory has more stringent design rules than the logic circuitry, the memory yield will not improve as quickly as the logic yield when the baseline defect density improves. Nevertheless, it is worth explicitly noting that the memory multiplier is in fact a function of the baseline defect density. This is not a major issue as long as the average baseline

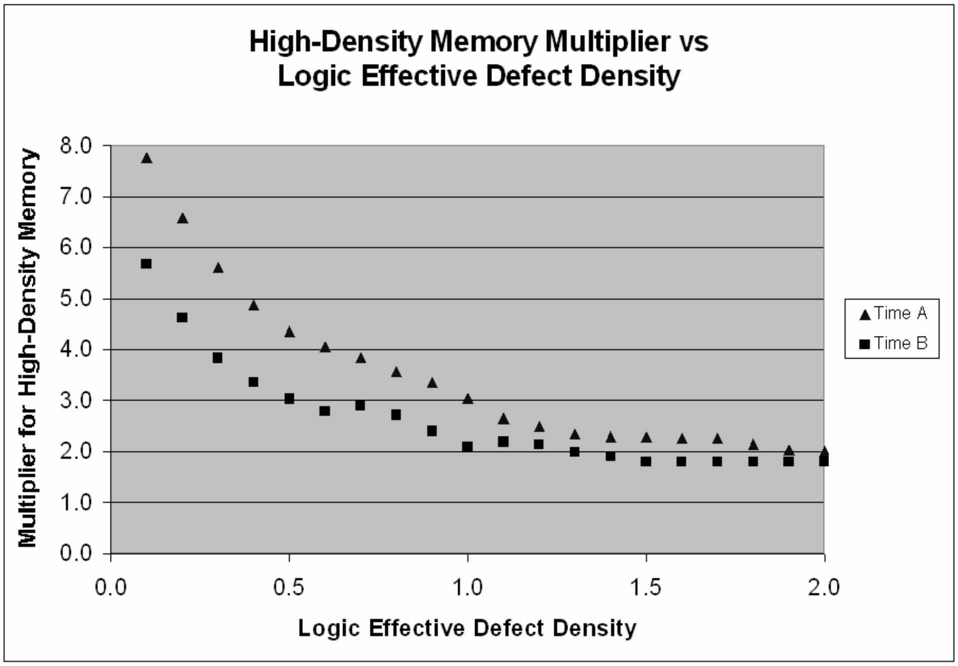

Figure 6: High-density memory multiplier as a function of logic effective defective density, evaluated at two different points in time.

defect density does not change much, but the more the average defect density improves (or worsens), the less accurate the yield forecasts will be. Significant strides in defect reduction require a re-evaluation of the memory scaling factors. A perhaps more surprising point is that at two different points in time, the same device required different memory multipliers for the same logic effective defect density. This is perhaps best understood by noting the threefold reduction in defect density that accompanies the maturation of a new technology. Such a dramatic reduction in defect density is likely to produce a change in the types of defects which are predominant, as well as simply changing the number of defects. This shift in the character of the defect population will impact the high-density memory differently than the logic due to the different critical areas of the different circuitry. Indeed, the morphing defect demographics may alter the relative importance of other failure mechanisms as well. Therefore, significant improvements in defect density will require a re-evaluation of all adder and scaling terms.

7. CONCLUSION

We have taken the Poisson yield model, and expanded it to include a number of yield-impacting factors which are identifiable in an already existing design database. We then calibrate the model, assigning values to each of the new terms introduced so as to produce an adequate fit to the actual yield performance of the devices we run. Once calibrated, the model is then incorporated into an automated query engine which links to both the design database and the wafer sort database. The resulting yield forecasting system can then generate a yield forecast for each lot sorted, regardless of device type or volume. This allows the yield team to check for lots or products missing the forecast, and quickly identify potential processing issues, test program issues, and design bugs. The system also assists the finance team in accurately predicting fab revenues for a given product mix. And the system gives factory planners confidence that they are starting the right number of wafers to meet customer requirements. All three of these functions are particularly useful in an ASIC facility, where a large number of products are run, new products are introduced frequently, and the product mix can change dramatically with little warning. Finally, the system is constructed with enough flexibility to respond to changes in fab processing and to accommodate new technology options.

REFERENCES

1. T. Chen and M. J. Wang, “A Fuzzy Set Approach for Yield Learning Modeling in Wafer Manufacturing,” IEEE Transactions on Semiconductor Manufacturing, **Vol. 12**, pp. 252-258, 1999.
2. C. K. Shin and S. C. Park, “A machine learning approach to yield management in semiconductor manufacturing,” Int. J. Prod. Res., **Vol. 38**, pp. 4261-4271, 2000.
3. H. T. Heineken, J. Khare, and W. Maly, “Yield Loss Forecasting in the Early Phases of the VLSI Design Process,” *Proc. IEEE 1996 Custom Integrated Circuits Conf.,* pp. 27-30, 1996.
4. P. Simon, K. Veelenturf, P. Adrichem, J. Jong, S. Sprij, and W. Maly, “A Layout Based Manufacturability Assessment and Yield Prediction Methodology,” *Proc. SPIE*, Vol. 3743, pp. 282-288, 1999.
5. T. Okabe, M. Nagata and S. Shimada, “Analysis of yield integrated circuits and a new expression for the yield,” Electrical Engineering in Japan, **Vol. 92**, pp. 135-141, 1972.
6. J. A. Cunningham, “The Use and Evaluation of Yield Models in Integrated Circuit Manufacturing,” IEEE Transactions on Semiconductor Manufacturing, **Vol. 3**, pp. 60-71, 1990.
7. D. Dance and R. Jarvis, “Using Yield Models to Accelerate Learning Curve Progress,” IEEE Transactions on Semiconductor Manufacturing, **Vol. 5**, pp. 41-46, 1992.
8. C. H. Stapper, “Modeling of Integrated Circuit Defect Sensitivities,” IBM J. Res. Develop., **Vol. 27**, pp. 549-557, 1983.
9. A. V. Ferris-Prabhu, “Modeling the Critical Area in Yield Forecasts,” IEEE Journal of Solid-State Circuits, **Vol. SC-20**, pp. 874-878, 1985.
10. R. M. Warner, Jr., “Applying a Composite Model to the IC Yield Problem,” IEEE Journal of Solid-State Circuits, **Vol. SC-9**, pp. 86-95, 1974.
11. S. P. Cunningham, C. J. Spanos, and K. Voros, “Semiconductor Yield Improvement: Results and Best Practices,” IEEE Transactions on Semiconductor Manufacturing, **Vol. 8**, pp. 103-109, 1995.