ICID - A Robust, Low Cost Integrated Circuit Identification Method

v0.9 March 19, 2007 Keith Lofstrom, SiidTech

Summary

ICID (**I**ntegrated **C**ircuit **ID**entification) is an integrated circuit IP block that uniquely identifies the individual integrated circuit on which it is placed. The IDs extracted at test time from each chip can be stored in a database, and rapidly retrieved.

Introduction

ICID produces a stream of repeatable, robust identification information based on the intrinsically statistical nature of the integrated circuit transistor manufacturing process. ICID IP blocks can be added to any design on any submicron CMOS process, consume little room, and are impossible to fraudulently modify. ICID requires no changes to standard integrated circuit manufacturing or testing processes. No programming steps are involved; the ICID block simply needs a clock and a few microwatts of power to produce a serial identification sequence. ICID blocks function better as devices and supply voltages are scaled down for deep submicron processes.

ICID uses intrinsic device variations that are permanently incorporated into transistors by the manufacturing process itself. While these variations are a result of the random placement of dopant atoms in transistor channels, they are as permanent as the silicon itself, and are highly resistant to change. As process technologies develop towards small, submicron geometries, these variations are accentuated, much as smooth sandpaper appears grainy when viewed under a microscope.

The ICID block measures device-to-device variations, but is insensitive to process, temperature, or power variations. In addition, ICID blocks are insensitive to variations in feature size that correlate to the photomask. This insures that each ICID block is different from every other ICID block (the correlation is low) while each individual block retains its identity over time and environmental variations.

The ICID block produces its output ID record based on measurements of the outputs of an array of ID bit cells, with each ID bit cell producing a fixed output based on permanent statistical variations. When the number of bit cells is large enough, the chances of duplication become very small.

The IDs are not completely deterministic. Because they are based on random variation, and some of those variations are small, some small fraction of the bits may change between measurements. However, most of the bits are permanent and unchanging, and a stable identification can be robustly extracted from them.

The acquisition and logging of the chip's ID can be easily and quickly done by the same digital IC tester that tests the chip's logic. The accumulated data can be rapidly searched to find out about a chip and its history.

The ICID IP Block

ICID is a hard IP (Intellectual Property) block, provided as a GDSII layout, and targeted at particular processes. In addition, Verilog and SPICE descriptions are provided. The customization to a process makes the ICID block as small and as robust as possible, given a particular customer's requirements.

An ICID block uses one layer of polysilicon and the first two layers of metal on a typical 130nm CMOS process, with an area of 50 x 80 microns for a 256 bit cell. Upper metal layers can be used to route power supplies over the top . Special processing is not required; an ICID block uses standard digital CMOS processing and fab flows. The ICID block does not need any time-consuming, yield-threatening programming.

ICID power consumption is tailored to the customer's requirements. The ICID block draws no power when unselected. When selected, the power drain is proportional to the maximum designed clock rate, per customer requirements. An ICID block running at 20 Mhz (5Mbps) uses less than 100 microamps, and slower cells can be designed to draw less power. A typical 256 cell ICID block will need 1030 clocks to produce an ID record. Nanopower cells requiring fewer clock edges are in design.

The ICID block is insensitive to metal migration and hot carrier effects that threaten many submicron circuits. In fact, when the block is unselected, the identification core is powered down in a special protection mode, completely disabling any wear-out mechanisms that could affect identification information over time. Since the ICID circuit is based on atomic structure rather than stored charges, it is not subject to tunnel discharge like an EEPROM based ID cell, nor the regrowth effects of poly fuse ID cells. The ICID block can tolerate anything the integrated circuit can tolerate.

The ICID block requires power, ground, clock and reset inputs, and a logical output providing the identification bit sequence. These signals may be connected as a private

data register on an IEEE 1149.1 JTAG bus. Alternately, they may be connected to internal buses, or external pads. The drive signals are easily produced from the signals available on a very small RFID chip.

Operation

A typical ICID block contains an array of 256 identity cells . The number of cells can vary, depending on the customer application, from 64 up to many thousands. Each cell has a permanent analog value, the voltage threshold difference between a pair of matched MOS transistors. The ICID block contains a stimulus circuit that selects each identity cell in sequence, and a circuit that compares successive array outputs, producing a digital identification output.

Although the threshold of an MOS device is a function of many process and environmental variables, the threshold voltage difference between a pair of devices stays fixed over time. Most other changes - current levels, temperature effects on threshold, even charge trapping in slow surface states tend to affect both devices in the pair, and are ignored by the differential circuitry. This also makes the device insensitive to high levels of power supply and substrate noise.

Some of the identity cells are "hard-coded", or type identification cells, whose outputs are guaranteed to be the same on every die made from the same mask. These cells are useful for identifying which mask revision is used to make a particular die. In addition, if there are many die images on one step-and-repeat mask reticle, each die image can have a different hard-coded type identification, simplifying the task of finding an ID in a database. Type identification cells reduce the effort required to find a particular ID sequence in a large database.

ICID Quality Verification

Since we are intentionally making a circuit that is difficult to modify, it is not practical to include a "checksum" to determine if the ICID circuit block is working correctly. Instead, we look at the serial bit stream out of the part, which is designed to have "anticorrelation" between neighboring bits. This makes long series of zeros or ones that appear on broken ICID blocks factorially unlikely on working blocks, and such patterns are easy to identify and discard. A 10 bit run on uncorrelated blocks has a probability of 977ppm, while the probability is 0.27ppm for an anticorrelated block.

The anticorrelation also makes the data from an ICID block self-clocking, which

simplifies sending the ID bits over a communications link, such as in an RFID cell.

The ICID block is small, and the chances of the block mis-performing due to a yield defect are on the order of 100PPM, but it is still important to detect these rare failures so they can be specially handled.

Determinism

The ID array values will have a random probability distribution. Some ID cells will have values very close to zero. Because of thermal and system noise, comparator offsets, and other effects, the bits measured from these few cells may be non-deterministic. Repeated measurement of the array will average out into a more robust sequence, but there will still be small uncertainties in measurement that will change over time. Most bit measurements will always produce repeatable results, but a few will sometimes produce a one and sometimes a zero. Thus, no ID sequence produced by an ICID array will ever be completely deterministic. This is intrinsic to the nature of IDs extracted from non-modifiable “found” information.

Some small portion of the ID bit sequence may change value or “drift” as the circuit is exposed to contaminants or stress. However, changed bits in messages are nothing new in communications theory. For a sufficiently large number of bits, and relatively few messages, many bits can be changed and a message can still be identified.

The statistical variations affecting ICID come in three flavors: Desirable, Drift, and Global variation. Desirable variation occurs at the die level only, and is permanent and unalterable. It results from the nature of physical transistors themselves, which are made of randomly placed atoms in non-uniform substrates. As processes go towards increasingly small geometries, the effects of individual atomic variation becomes more prominent. ICID benefits from process shrinkage.

Drift variation comes from temperature and operating changes, thermal noise, and synchronous system noise - these variations are not predictable and make the ID message nonrepeatable. This type of drift is reduced by careful circuit design and repetitive averaging. Other drift changes can come from contamination or voltage or heat stress, which can cause the dopant and impurity atoms in the silicon to move, potentially changing the ID. The levels of stress necessary to significantly alter an ICID circuit will probably damage the integrated circuit function itself. In any case, it is impossible to alter an ICID circuit to match another ICID circuit by modification of its environment.

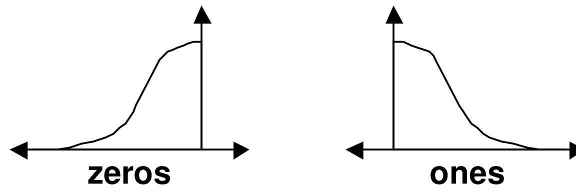
Typical drift values have been measured to be well under 5% of the desirable cell variation. The blocks are normally unpowered, which reduces electrical stress on the ID device pairs. Even with all the precautions taken to reduce cell drift, the ICID blocks are designed to accommodate up to 25% drift, to insure that they will operate robustly even in the presence of unpredictable fabrication and environmental problems.

The Heavy Math:

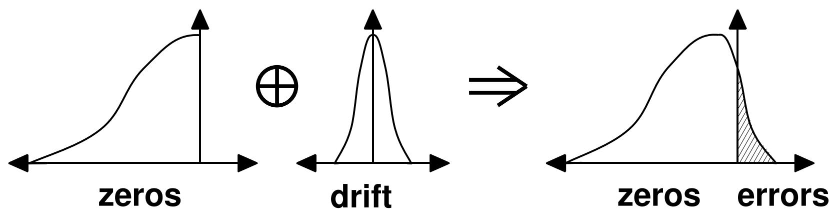
The probability of a bit error may be computed by assuming the measured ID variable x follows a normal distribution with a deviation of σ:

$$P(x)=\frac{1}{\sigma\sqrt{\pi\,\pi}}e^{-x^2/\sigma\sigma^2}$$

Applying a perfect comparator to variable ***x***, half the cells will be negative and half will be positive, producing zeros and ones. The result will be two truncated normal distributions of zeros and ones, as shown:



The effect of drift or other random variation added to subsequent measurements is to convolve these two distributions with another normal distribution, the drift. Typically, the drift will be a narrower normal distribution. When they are convolved, some zeros will be erroneously changed into ones, like so:



And a similar portion of ones will be turned into zeros. The probability of a bit error **p** can be computed from the following double integral:

$$p=2\int_{0}^{\infty}dx\int_{-\infty}^{\infty}P_{bit}(y)P_{drift}(x-y)\,dy$$

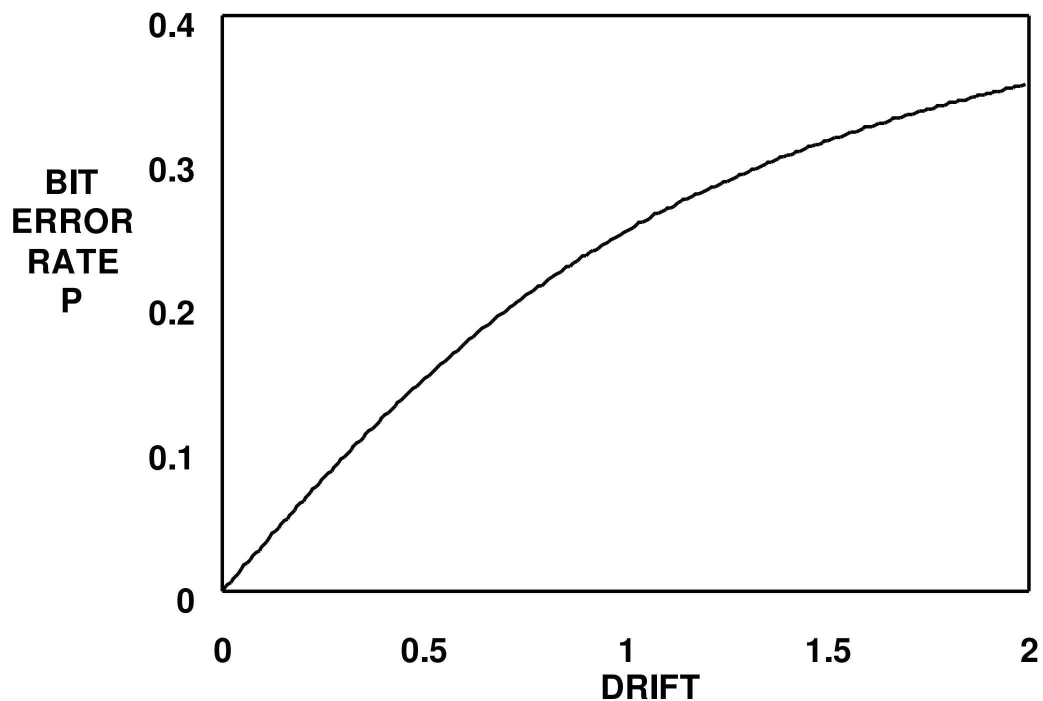
Substituting the probability functions, and recognizing that $P_{bit}(y) = 0$ for y>0 yields:

$$p = \left(\frac{2}{\pi d}\right) \int_{0}^{\infty} dx \int_{-\infty}^{0} e^{-y^2} e^{-(x-y)^2/d^2} dy$$

After suitable manipulation, the double integral may be reduced to a more computable form, using erf(), for which numerical formulas are available:

$$p = \frac{1}{2} - \frac{1}{\sqrt{\pi\pi}} \int_{0}^{\infty} erf\left(x/d\right) e^{-x^2} dx$$

This integral is difficult to solve analytically, but easy to solve numerically. The result (assuming a perfect comparator and normal distributions) is the following graph:



The graph shows the probability of a bit change in the identity sequence as a function of drift. The probability of bit errors never reaches 0.5 even for very large values of drift, so a large enough ICID block can always produce a reliable ID. As the graph shows, an enormous drift of 100% causes a bit error rate of only 25%, which means that an ID sequence with 256 bits can reliably identify one ID block out of a million ID sequences. For the small drifts encountered in modern VLSI processes, the bit error rate will be much less, and trillions of ID sequences can be reliably distinguished.

Global correlation comes from variations that occur in design and mask making.

The edges of the ID cell array will be different than the center, and there will be variations in the edges of the patterns on the photomasks. If 20% of the final variation of the cell array is globally correlated, we can expect about a 4% correlation in the bit sequence. The entropy, or effective number of identification bits in the sequence, will be reduced by the square of the correlation. The effects of global correlation can be compensated for by slightly increasing the number of ID bits.

Two binary IDs can be compared by computing the **Hamming distance** between them. The Hamming distance is the number of bits that differ between the two IDs. Two identical IDs have a Hamming distance of zero. The Hamming distance between two randomly different IDs will have an average of N/2.

Assuming the bits are independent, the distribution due to drift can be computed from a weighted binomial distribution. The probability for a part with **N** cells and a bit error probability **p** drifting to an Hamming distance of **A** is given by:

$$P_{drift} = \left(\frac{N!}{(N-A)!\,A!} \right) (1-p)^{(N-A)} p^{A} \qquad mean = Np$$

$$\sigma = \sqrt{Np(1-p)}$$

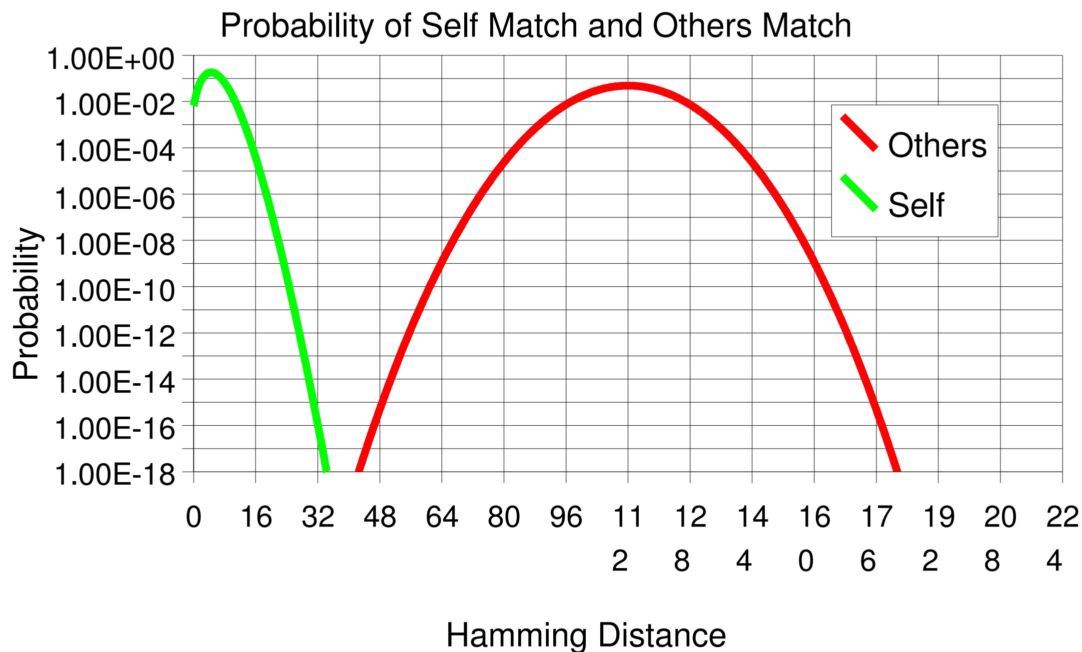
This is shown as the left curve in the graph below, for N=224 cells and 2.5% drift (p=0.008). Meanwhile, for any two dissimilar parts, the probability of false match versus Hamming distance is given by a probability distribution modified by correlation C (a small positive number, typically C < 0.05) :

$$P_{falsematch} = \left(\frac{N!}{(N-A)!\,A!} \right) \left(2^{-N}\right) \left(1-C\right)^{A} \left(1+C\right)^{(N-A)}$$

$$mean = N(1-C^2)/2 \qquad \sigma = \sqrt{N}/2$$

This formula is shown as the right curve. Each curve sums to one.

The graph shows the expected probabilities from comparing one binary ID to a database of 224 bit Ids, drifting by 2.5% There is less than one chance in a trillion that the Hamming distance will be between 30 and 50. The second curve is multiplied by the typical numbers of comparisons required to find an ID in the database; however, it is also reduced by convolution as described below.

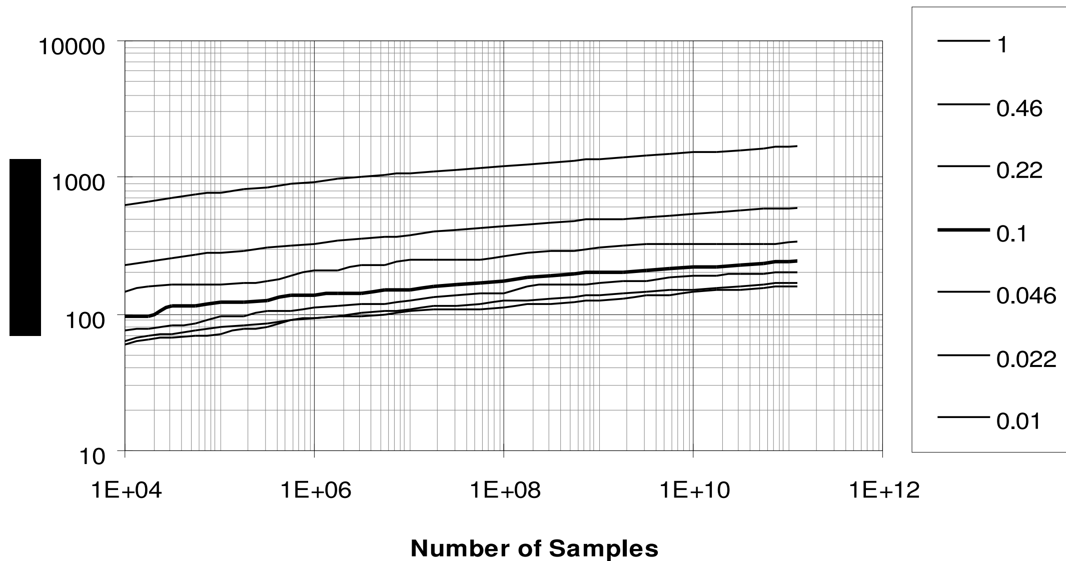


The false positive and false negative rates will not be mathematically zero, but they will be immeasurably small when the array is sufficiently large, certainly better than fingerprint identification and other legally acceptable forms of identification. We can set a threshold in our test of 40, and easily distinguish the correct ID in the data base. Modern semiconductor processes can be expected to drift 2.5%, and if drift ever becomes a problem, we can simply increase the number of bits.

If the part is not in the database with a Hamming distance of less than 40, the component has either been badly mistreated, it has not been logged, the identification circuit has failed, or the component is a counterfeit produced by some other fab.

A 224-cell array was employed in the example ICID illustrated here. However, with less drift, fewer chips to be identified, or when less reliable identification is permitted, fewer array cells are needed. For example, with a 10% maximum drift, and a 1 in 1 million permitted error rate, as few as 64 cells provide adequate identification. For a 1 in 1 quadrillion error rate (10-15) and a drift of 240%, 4096 cells may be needed. For any given drift, acceptable error rates may be achieved with a sufficient number of cells.

The number of cells required to distinguish a given number of samples is shown in the curve below:

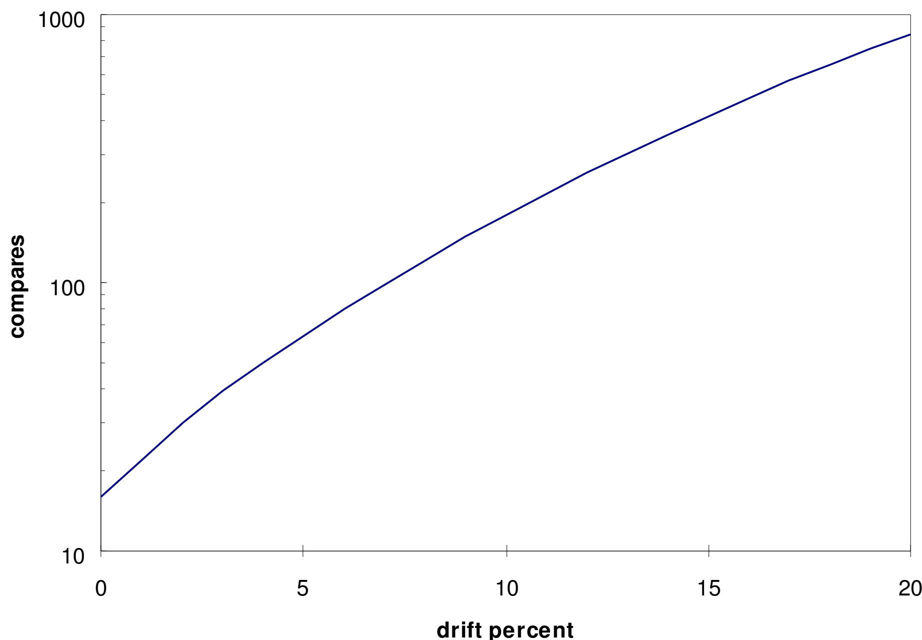


Comparing an ID to a Database

There are two common ID operations. The first, verification of a single chip ID to a given ID record, is easy. The ID record is compared to the chip ID, the Hamming distance is computed, and the Hamming distance indicates whether the chip matches or not.

The second, finding a particular chip ID in a large database containing many IDs, is more difficult, because of nondeterminism. However, clever search techniques can still find the ID quickly, if the drift is low.

The following graph illustrates one measure of difficulty, the average number of byte-wise "search compares" versus drift for one million devices stored in a 16 bit hash table with 65536 entries. A fast processor can perform millions of search compares per second, so even if the drift is large we can still rapidly find a particular ID in a large database, or find thousands of IDs per second.



The rare IDs that drift, and unluckily have large variations in the first few bits, will take longer to search - they might have to be compared to most IDs in the database. As a result, these rare IDs are exposed to more comparisons, increasing the chance of a false positive comparison. However, most of those false positives will have a high Hamming distance, almost certainly higher than the norm of the true comparison. The chance of encountering a false positive match is reduced to the integral of the *convolution* of the two probability curves. That is, the true probability of a false positive match after an extensive search of the database can be computed by multiplying the two curves together and integrating the result, turning two small probabilities into an extremely tiny one.

ICID with Error Correction

The ICID cell is most useful when the ID is compared to itself in subsequent measurements, for identification. Hamming distance measurements are simple, and if the ID is long enough the average number of error bits will converge on the drift. Is there any way to remove the drift using error correction bits?

We can store additional bits, perhaps in an EPROM, that apply block error correction

to an ICID cell. Most block error correction algorithms are designed to work on small numbers of bits, but small blocks of bits are more likely to have a high percentage of drift than large blocks. With an average drift of 5%, a 224 bit random ID will have an average of 11 bits changing. If an authentication error rate of less than 10ppm is desired, then error correction capable of correcting 36 bits out of 224 bits is needed. That requires more error correction bits than the ID itself. It is probably better to copy a measurement of the ICID into the EEPROM, then generate a fault condition if the EEPROM does not match the ICID within a specified Hamming distance.

SiidTech Successes

The SiidTech ICID cell has been successfully deployed at many fabs and in many process technologies, from 350nm down to 90nm. Experiments have been performed with millions of sample ICID blocks, verifying the behavior and the mathematical descriptions. LSI Logic is using ICID to track all integrated circuits through fab and production, and our Asian partner, Hitachi ULSI, is deploying ICID in their own and other manufacturer's products.

References:

1) See our website, http://www.siidtech.com.

2) US patent 6161213, System for providing an integrated circuit with a unique identification.

3) US patent 6738788, Database system using a record key having some randomly positioned, non-deterministic bits.

4) K. Lofstrom, W. Daasch, D. Taylor, IC Identification Circuit using Device Mismatch, 2000 IEEE International Solid-State Circuits Conference Digest of Technical Papers Volume 43 Catalog #OOCH37056

4a) PDF at http://www.kl-ic.com/isscc2K.pdf

5) K. Lofstrom, D. Castaneda, B. Graff, A. Cabbibo, ICID - Tracing Individual Die from Wafer Test through End-Of-Life, 10th International Mixed Signal Test Workshop, June 23-25, 2004, Portland, OR.

5a) PDF at http://www.kl-ic.com/imstw2004.pdf