

The Design Record and the Fifteen-Year Part

Design Data and IP Lifecycle Management for
Automotive Integrated Circuits

Executive Summary

An automotive semiconductor is required to remain traceable for longer than the project that produced it, longer than the tools that built it, and in many cases longer than the careers of the engineers who signed it off. The manufacturing side of the industry has accepted this and built for it. The design side has largely not.

This paper explains what the design record must survive in an automotive program, the three questions it must eventually answer, and how SOS Enterprise supports each answer. It is written for the functional safety manager who owns the evidence, the design data or CAD manager who owns the mechanics, and the engineering executive who carries the risk when evidence cannot be produced.

Introduction

Automotive unit level traceability carries a data retention requirement of at least fifteen years, compared to five years for commercial parts (AnySilicon, n.d.). Every die is assigned an identifier, every wafer is assigned to a lot, and the chain extends to lead frames, epoxy, bond wires and boards (UTAC, 2023). This is mature, audited practice.

The design record behind it carries no equivalent requirement, and is usually retained by whatever the design organization happened to be running at the time.

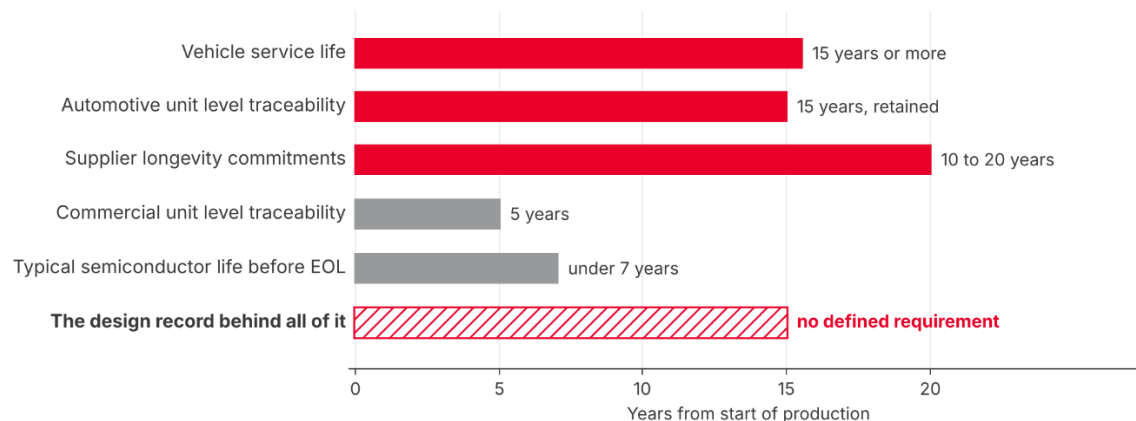


Figure 1. Everything in an automotive program is scoped to fifteen years or more. The design record that explains it has no defined retention at all.

That gap matters because the two records answer different questions. The manufacturing record establishes which part failed and where it came from. The design record establishes what that part contained: which IP revisions, which process design kit, which engineering changes, which waivers, and the evidence on which each was accepted. When a field return in year twelve resolves to a wafer lot and a design revision, the manufacturing record has done its job. The design record then has to explain the content behind that revision.

The exposure is commercial as well as technical. Car companies paid roughly \$40 billion a year in warranty claims over one five-year period. One European manufacturer attributes close to four cents in every warranty dollar to semiconductor failures, and individual automotive recalls routinely exceed \$100 million (KLA, 2018). Slow answers are judged against those numbers. Imprecise answers extend the risk from the parts that are affected to the much larger population that might be.

What the Design Record has to Survive

Three forces act on an automotive design record, and they compound.

The Part Outlives the Project

Vehicles are engineered for service lives of fifteen years or more, while many semiconductors reach end of life in under seven (Automotive IQ, n.d.). The industry has closed that gap with explicit longevity programs. STMicroelectronics commits to a minimum of fifteen years on a set of products and has extended some to twenty (STMicroelectronics, n.d.-a, n.d.-b). Renesas operates three supply patterns at ten, fifteen and twenty years or more (Renesas, n.d.). NXP commits to ten or fifteen years (NXP Semiconductors, n.d.). Texas Instruments states typical life cycles of ten to fifteen years, often longer (Texas Instruments, n.d.).

A fifteen-year supply commitment is also a fifteen-year engineering commitment. Within that window a team may have to respin for a process change, qualify a second source, answer a field failure, or port the design to a surviving node. Each of those requires the original design state, not a description of it.

Fifteen years spans at least one process design kit generation, usually more than one EDA tool migration, frequently a corporate acquisition, and almost always a complete turnover of the engineering team.

Each event damages a different part of the record. A process design kit update invalidates the assumption that the design can simply be rerun. A tool migration moves files and often strips away meaning that existed only in directory layout or naming convention. An acquisition merges estates whose conventions were never compatible and whose access rules reflected different organizations. Team turnover removes the undocumented knowledge that made the other three survivable.

Only a record held as structured data, with explicit relationships, can survive all four. A record that depends on conventions everyone understood at the time cannot.

One Design becomes Many Parts

An automotive device is rarely a single product. A generic functional design is configured into many SKUs, each a particular combination of enabled function, power target, autonomy level and customer preference. Microcontroller families are built around scalable architectures, with low-end devices for body electronics and high-end variants carrying lockstep cores for ASIL D systems, sharing instruction sets, peripheral register models, clocking and interrupt structures (Promwad, n.d.-a).

Safety content is reused the same way. A safety island is deliberately carried across derivatives, such as mirror replacement, front camera and radar, without recertifying it for each variant (Promwad, n.d.-b). That reuse is the economic argument for the platform, and it is also what makes the consequences of a defect wide rather than narrow.

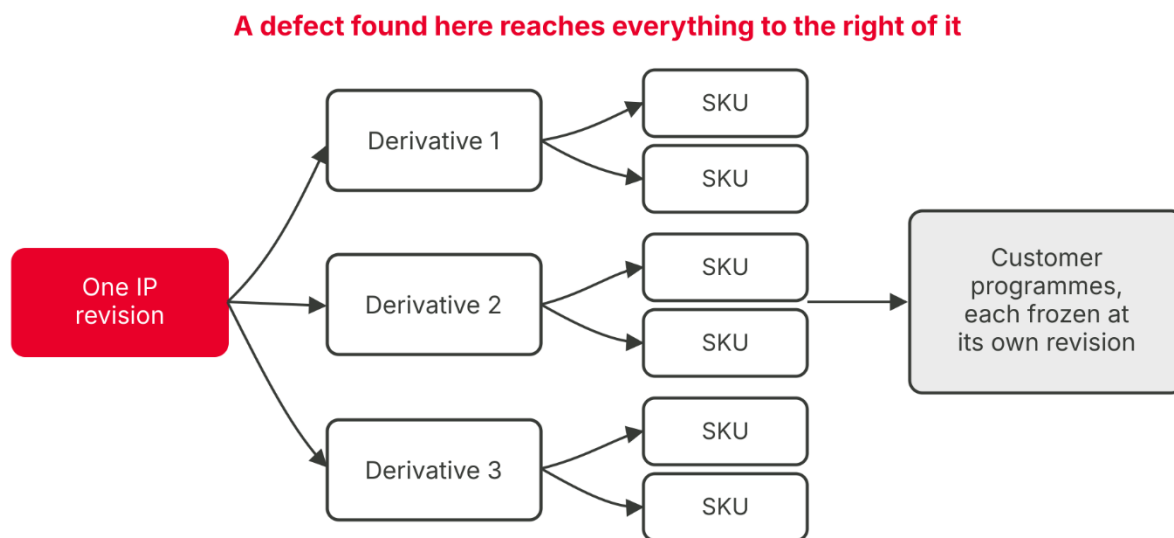


Figure 2. Platform reuse is the economic argument for an automotive program, and it is also what determines the size of a containment decision.

The design record therefore has to express composition, not just content. Knowing the state of an IP is of limited use without knowing every product that contains it and knowing which revision each of those products was frozen against.

Safety Evidence Detaches from the IP it Describes

ISO 26262 Part 8 covers configuration management, change management, documentation management and tool qualification, and requires that safety related work products be identified, versioned and controlled under a defined configuration management process. Part 10 defines the Safety Element out of Context, the element developed without a specific vehicle item in view (International Organization for Standardization, 2018).

An IP qualified as a Safety Element out of context does not travel alone. It travels with a safety manual, a stated set of assumptions of use, requirements traceability and verification evidence,

and those assumptions have to be validated on the target platform at integration (Intertek, 2026). The qualification is conditional, and the conditions are the work product.

In practice the block and its conditions separate. The netlist is in the design data management system, the safety manual is a document in a separate store, the assumptions of use are in an appendix that was current two revisions ago, and the integrator receives whichever copy reached them. The IP is then reused under assumptions nobody has checked against the version actually integrated.

This affects both sides of a supply relationship. The IP supplier carries the risk that its qualification is relied upon outside the conditions it was granted under. The integrator carries the risk of building a safety argument on conditions it cannot demonstrate were checked. Neither party can close that risk from its own records alone, because the binding between the block and its conditions is exactly what was never captured.

Three Questions Every Design Record must Answer

The shape of the problem is easiest to see in a single sequence. The following is illustrative rather than drawn from a particular program.

A vehicle built in year three returns a fault in year twelve. The Tier 1 isolates it to an ECU, the ECU supplier isolates it to a device, and unit level traceability resolves that device to a die, a wafer, a lot and a design revision. To this point the process works as designed, and it has taken days.

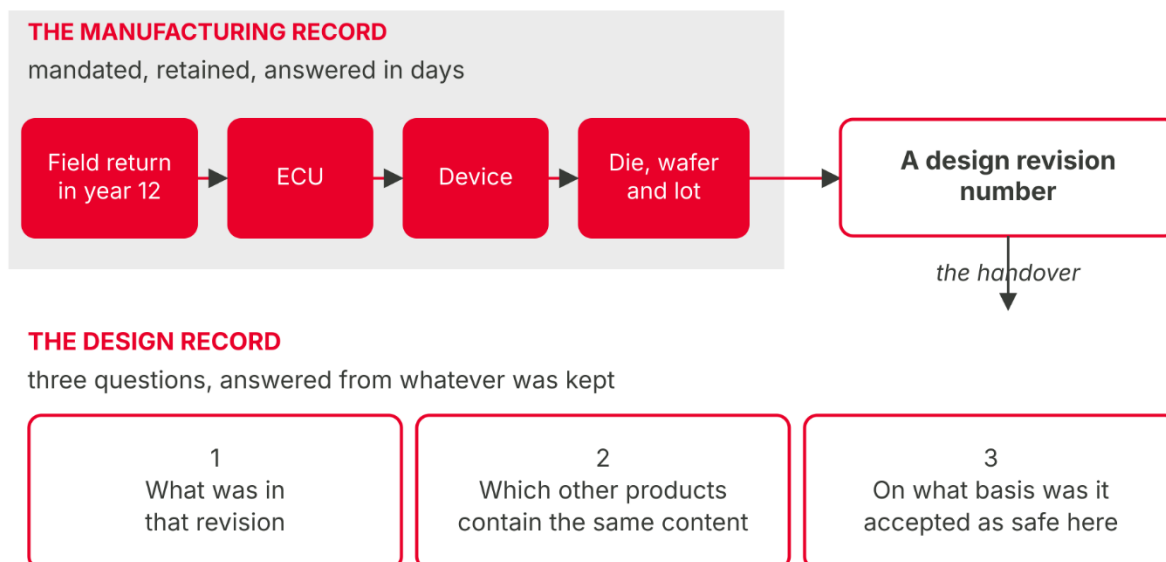


Figure 3. The manufacturing record hands over a revision number. Everything after that comes from the design record.

The question then passes to the design organization. Three answers are needed in sequence: what was in that revision, which other products contain the same content, and on what basis the content was accepted as safe in the context where it was used.

The first answer usually exists somewhere: the design data was retained and the revision can be found. The second and third answers are where programs lose time. Composition and justification were often never captured as data. They lived in a directory layout that made sense at the time, a spreadsheet maintained by someone who has since left, and documents whose relationship to specific revisions was obvious only to the people who produced them.

Nine years later none of that is obvious. The reconstruction is performed by engineers who were not present, from evidence that was never intended to be read by strangers, under time pressure set by a containment decision.

Establishing What a Part Contained

The design revision recovered from the manufacturing record is only a label. To turn it back into a state, the organization must recover the exact revision of every IP in the hierarchy, the process design kit and its version, the engineering changes applied after the last clean state, the accepted waivers and approvers, and the verification evidence that justified sign-off.

This is straightforward during the project, when the workareas still exist and the people are still there. It becomes difficult at the point where it is needed, which is years later.

When a defect is confirmed in an IP, the containment question follows immediately. Which derivatives contain that revision? Which SKUs were built from those derivatives? Which customer programs took those SKUs, and at which revision was each frozen?

A hierarchical composition reconstructed by hand from directory structures and engineer memory produces an answer in weeks. Containment decisions are made in days. Without a defensible answer, organizations make those decisions conservatively, which is the expensive direction. The difference between a precise containment boundary and a cautious one is measured directly in parts recalled.

Proving Why the IP was Safe in Context

An assessor asking about a reused safety element will ask which revision was integrated, which version of the safety manual and assumptions of use were applied to that revision, who confirmed the assumptions held in this context, and what evidence supported that confirmation.

Most organizations hold each part of that answer somewhere. The difficulty is that the parts sit in different systems with no binding between them. The answer is reconstructed rather than retrieved, and reconstruction is only as reliable as the memory of the person performing it.

What SOS Enterprise Provides

SOS Enterprise brings design data management, IP lifecycle management and governance together in one platform for semiconductor design organizations. These capabilities map directly to the three questions above: design data management establishes what a part contained, IP lifecycle management shows where that content went, and governance records the basis on which each use was approved and by whom.

The platform holds metadata about design objects together with a reference to where the design data itself lives. Data can be held in SOS Enterprise, or in an existing repository, a path on disk or an external location, and a single composition can span several of them.

Design Data Management: The Record of What Existed

Every change to a design object is captured as a revision, with branching and tagging for the normal working patterns of a design team. A snapshot records the revision of every file in every library at a single moment, as an immutable record, which is the mechanism behind a defensible tapeout record. Workareas give each engineer a working copy that is tracked against the versioned state, so what an individual has locally is always relatable to what the project holds.

Design data is replicated across sites and cached locally, so teams in different regions can work against the same versioned state at local speed instead of maintaining parallel copies that diverge. A snapshot taken at one site describes a state that is meaningful at every other site.

Two levels of version are maintained, because two different things are being versioned. The design data management layer carries every check-in, which is what a working team needs. The IP lifecycle layer carries only the versions that a consumer is entitled to reuse, and each of those points at an exact state in the layer below.

Released versions, the only ones offered for reuse

each one pointing at an exact state on the line below

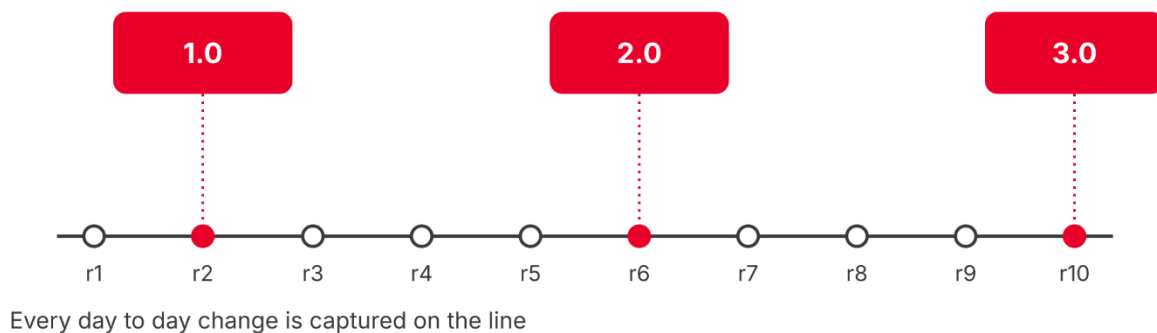


Figure 4. Day-to-day work and released versions are recorded separately, so the question of what was released does not have to be inferred from the history of what was worked on.

This separates the record of what was released from the history of work in progress, so released states do not have to be inferred years later. Projects are also separated into exploratory and

design work, and an IP is promoted as it matures. Early work therefore does not need to meet the same standard as the record that ships.

IP Lifecycle Management: The Record of Where it Went

Each IP is catalogued as a first-class object with a structured set of attributes. Attribute schemas are defined by the organization, with declared value types, classes, categories and tags, so qualification status, ASIL level, target process and conditions of use are held as structured data rather than as free text. Structured attributes are searchable, comparable across the catalogue, and reportable.

Each IP carries a hierarchical bill of materials, resolved on the server rather than assembled by the caller, and records its consumers, so the relationship is navigable in both directions. Impact analysis runs before an IP is changed, retired or deleted, reporting what downstream use exists.

The containment question becomes a query against recorded structure. The same structure also supports the routine development question: what breaks if this IP is upgraded? Instead of being asked once during an incident, that question can be asked continuously during development.

Documents are first-class objects and attach to a specific IP or project revision rather than sitting beside it. Document labels bind documents into classes, so a class of IP can be required to carry a class of document. A safety manual, a statement of assumptions of use and a verification report are therefore tied to the IP revision they describe. A consumer reusing that revision reaches the evidence that applies to it.

Reuse attributes are recorded per consumer, so the terms under which a particular consumer took a particular revision are part of the record. When an assessor asks on what basis an IP was used in a given context, the answer is a retrieved transaction.

An existing estate is brought in without a rebuild. The catalogue can be populated by an initial import at scale, either from a workbook or by walking the existing design estate directly, so the first catalogue reflects what the organization actually has rather than what it intended to have.

Governance: The Record of Who Approved What

Requests to view an IP's details, to download an IP, to upload a new IP or revision, or to join a project are handled as approval tasks. The request, the approver and the outcome are recorded, and the approval flow itself is defined by the organization rather than fixed by the platform.

Access is controlled at several levels. Roles define what a user may do; object permissions define where those actions apply. Access profiles bundle sites, groups and users into named visibility rules, allowing a program to be shared with the sites and partners that need it and withheld from those that do not. Attribute-level controls, enforced on the server, let commercially restricted conditions sit on the same object as the technical description without being exposed to everyone who can see the object.

Event driven automation allows an organization to encode its own rules, so that a condition on an object triggers a defined action. A change to a safety qualified IP can require a particular approval, and a consumer of that IP can be notified when the revision they depend on changes.

The combination is what allows a supplier to publish an IP into a shared catalogue with its safety evidence attached while controlling who may see the details, who may download it, and on what terms each consumer took it. The catalogue is common and the visibility is not.

Retrievable Records for Future Teams

The catalogue supports browsing by class, category, tag and custom attribute, full text search across every object type, and a structured query language for precise questions. Saved queries hold a filter centrally so that a recurring question is asked the same way each time it is asked. Comparison between IPs, revisions and documents is a first-class operation.

Each IP carries a discussion thread, keeping questions and answers attached to the IP instead of buried in the inbox of whoever responded. Over a fifteen-year life, the record of why is often more valuable than the record of what, and it is often the first part lost.

Everything available in the interface is also available through a documented REST interface, so the record can be pulled into whatever quality, safety or reporting system the organization already runs, and so that evidence assembled for an assessor can be produced repeatably rather than compiled by hand.

The Three Questions, Against the Mechanisms

Question	What Answers It
What was in this part	Two levels of version, with a released version pointing at an exact state in the design data; snapshots as an immutable record of every file revision at a moment
Which products share the content	Hierarchical bill of materials resolved on the server, consumer tracking in both directions, and impact analysis before any change, retirement or deletion
On what basis was it safe here	Documents and structured attributes bound to the IP revision they describe, reuse recorded as an approved transaction with per consumer terms
Can any of this still be read in year fifteen	A catalogue that references data locations rather than containing them; full text and structured search; comparison; per IP discussion history

Practice, Independent of Tooling

The following practices are worth settling before a platform is selected, regardless of the system an organization uses:

Decide what a released state means, and make producing one deliberate. A release that is a side effect of a routine operation will be produced inconsistently over fifteen years.

Bind evidence to revisions, not to products. A safety manual that describes an IP in general will be ambiguous at the exact moment it is relied upon. Attach it to the revision it was written against.

Record assumptions of use as data. Assumptions held only in prose cannot be checked automatically at integration, cannot be searched across the catalogue, and cannot be reported against. Structured attributes can.

Record consumers at the point of reuse. Reconstructing who used what after the fact and from directory structures is the single most common cause of a slow containment answer.

Decide the retention question explicitly. Manufacturing retention is set by requirement. Design retention is usually set by whatever the storage budget allowed. Make it a decision with a stated period and a stated scope.

Test the fifteen-year claim now. Take a design taped out three or more years ago and attempt to reconstruct its exact composition using only what the organization retained. The result is the honest measure of current exposure, and it is cheap to run.

Where to Start

Measure the Exposure First

The reconstruction exercise is the useful first step, because it converts an abstract concern into a measured one. It takes a few days and needs no new tooling.

1. Choose a design taped out three or more years ago that is still in production.
2. Choose one IP inside it that is reused elsewhere.
3. Using only what the organization retained, answer the three questions mentioned earlier. Establish the exact composition of the revision that shipped. List every other product containing that IP revision. Produce the safety evidence applied to it at the version integrated.
4. Record how long each answer took, and how much of it came from a system rather than from a person.

Most organizations find the first answer in hours, the second in weeks, and the third only if a particular individual is still available. That result is the honest measure of current exposure, and it is the number to take to a program review.

Then Close the Gap Going Forward

From there, the usual sequence is to catalogue the current IP estate, attach existing safety and verification evidence to the revisions it describes, and record consumers from that point forward. None of these steps requires a design flow change, and each one shortens the path to answering the three questions.

SOS Enterprise supports each step directly. The catalogue is populated from the existing estate rather than rebuilt, so the starting point is what the organization actually has. Evidence attaches to the IP revision it describes, keeping the safety manual and assumptions of use bound to the version they were written against. Consumers and reuse terms are recorded as reuse happens, so the composition answer accumulates through normal work instead of being assembled under pressure years later. The design data management, IP lifecycle and governance capabilities described in section 4 are the three parts of that record.

The effect on the three questions is cumulative rather than immediate. A catalogue built this year does not explain a part that shipped five years ago. It does mean that the part shipping next year will be explicable for the whole of its fifteen-year life, and in an automotive program that is the horizon that matters.

The programs that answer quickly in year twelve are the ones that decided in year one that the design record was a deliverable.

Click here for more information on SOS Enterprise or to book a free trial: [SOS Enterprise | Keysight](#)

References

AnySilicon. (n.d.). Unit-level traceability for automotive customers. <https://anysilicon.com/unit-level-traceability-for-automotive-customers/>

Automotive IQ. (n.d.). Bridging the lifecycle gap: Managing semiconductor obsolescence in automotive ECUs. <https://www.automotive-iq.com/industry-reports/articles/bridging-the-lifecycle-gap-managing-semiconductor-obsolescence-in-automotive-ecus>

International Organization for Standardization. (2018). ISO 26262:2018 Road vehicles, functional safety. <https://www.iso.org/standard/68383.html>

Intertek. (2026, January 15). Exploring safety elements out of context. <https://www.intertek.com/blog/2026/01-15-exploring-safety-elements-out-of-context/>

KLA. (2018). Process watch: The (automotive) problem with semiconductors. https://www.kla.com/documents/01_ProcessWatchAutomotive_2018_01.pdf

NXP Semiconductors. (n.d.). Product longevity. https://www.nxp.com/products/nxp-product-information/nxp-product-programs/product-longevity:PRDCT_LONGEVITY_HM

Promwad. (n.d.-a). Unified automotive MCU families: When platform reuse really cuts engineering cost across vehicle domains. <https://promwad.com/news/unified-automotive-mcu-families-platform-reuse>

Promwad. (n.d.-b). Safety island design: ASIL decomposition across heterogeneous compute fabrics. <https://promwad.com/news/safety-island-design-asil-decomposition-heterogeneous-compute-fabrics>

Renesas. (n.d.). Renesas launches product longevity program to ease supply management for customers with long-lifecycle products. <https://www.renesas.com/en/about/press-room/renesas-launches-product-longevity-program-ease-supply-management-customers-long-lifecycle-products>

STMicroelectronics. (n.d.-a). Product longevity. https://www.st.com/content/st_com/en/about/quality-and-reliability/product-longevity.html

STMicroelectronics. (n.d.-b). STMicroelectronics sets 20-year availability for popular automotive microcontrollers. <https://newsroom.st.com/media-center/press-item.html/n4718.html>

Texas Instruments. (n.d.). Product life cycle. <https://www.ti.com/quality-reliability/quality/product-life-cycle.html>

UTAC. (2023). Unit level traceability for automotive semiconductors [Application note]. https://www.utacgroup.com/wp-content/uploads/2023/04/UTAC_Die-Level-Traceability-Application-Note.pdf

As a mission-critical design enablement partner, Keysight connects design, emulation, and test so customers can solve the world's most complex engineering challenges. Accelerate innovation at www.keysight.com.



This information is subject to change without notice. © Keysight Technologies, 2026, Published in USA, October 7, 2026, 3126-1791.EN