

Coverstory

Glyn – Spezialdistributor als wichtiger Projektpartner

Der 1980 von Glyn Jones gegründete europäische Spezialdistributor Glyn mit Zentrale in Idstein bei Frankfurt am Main



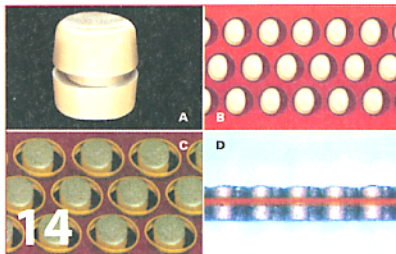
10

wurde von der Halbleiterkrise weniger stark getroffen als der Wettbewerb und wird seinen Umsatz gegenüber dem Vorjahr sogar steigern. Welche Faktoren diesen Erfolg des Unternehmens möglich gemacht haben und zukünftig sichern sollen, erklärt Thomas Gerhardt, Geschäftsführer und Vertriebsleiter von Glyn, im Gespräch mit der DESIGN&VERIFICATION.

Fokus

Direkte Verbindungstechnik für Backplanes und Board-zu-Board-Verbindungen

Mit 10 GBit/s ist die Grenze für Signalübertragung auf Leiterplattenmaterial noch nicht erreicht. Durch Optimierung des Transmissionspfads, speziell der Verbindungsstellen, kann auch ohne elektrische Hilfs- und Korrekturmittel die Integrität von schnellen differentiellen Signalen gewahrt bleiben. Traditionell kommen zweiteilige Steckverbinder zum Einsatz, die eingepresst oder gelötet werden. Eine neue direkte Verbindungstechnik umgeht diese problematischen Signalübergänge.



14

Schwerpunkt

Im D&V-Interview – „Demand Creator zu sein reicht nicht“ 18

Trendscouting ist einer breiteren Öffentlichkeit nur aus dem Konsumbereich bekannt. Ottmar Flach macht im Interview mit der DESIGN&VERIFICATION deutlich, wie wichtig das frühe Erkennen neuer Technologien und Anwendungen auch für Spezialdistributoren wie Atlantik ist und wie Trendscouts helfen können, den zukünftigen Erfolg des Unternehmens zu sichern.

PEOPLE & GAZETTE

06 News

COVERSTORY

10 **Spezialdistributor als wichtiger Projektpartner**

Im D&V-Interview: Managing Director Sales Thomas Gerhardt, Glyn GmbH & Co. KG

FOKUS: KOMMUNIKATION

14 **Direkte Verbindungstechnik für Backplanes und Board-zu-Board-Verbindungen**

Störungsfreie Signalübertragung mit Datenraten über 16 GBit/s

SCHWERPUNKT: DISTRIBUTION

18 **„Demand Creator zu sein reicht nicht“**

Im D&V-Interview: Ottmar Flach, CEO der Atlantik Networxx AG

COMPONENTS

20 News

DESIGN & DEVELOPMENT

28 **Verifikationsstrategie – Evolution oder Revolution?**

Erfahrungsbericht einer Design-Group

31 **Unterstützung bei der Anwendung der IEC61508**

Werkzeug zur Anforderungsermittlung

34 News

SUBSYSTEMS

36 **Design-In von Displays**

Kriterien bei der Auswahl und Integration von Displays

39 News

TEST & MEASUREMENT

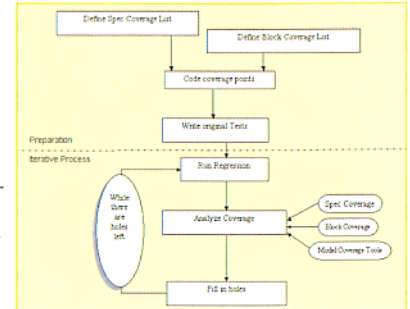
44 News

SOFTWARE	
48	Reverse Engineering Möglichkeiten und Grenzen verfügbarer Tools
51	News
EMC JOURNAL	
53	News
SERVICES	
58	Verlängerung des Produktlebenszyklus ohne Redesign Das Klonen digitaler Halbleiter mildert Folgen einer Abkündigung
61	News
DISTRIBUTION	
62	News
MANAGEMENT	
64	News
EVENTS & MORE	
66	Messevorbericht Electronica 2002
DEPARTMENTS	
03	Editorial
63	Impressum
<p>PLUS</p> <p>Dieser Ausgabe liegt das electronica 2002 Messe-Taschenbuch und der D&V TOOL GUIDE 2003 bei.</p> 	

Verifikationsstrategie – Evolution oder Revolution?

28

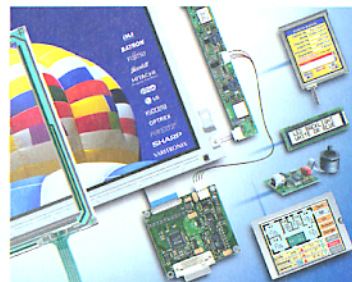
Die Verifikationsmethodik war in den vergangenen Jahren durch nicht weniger als fünf Paradigmenwechsel geprägt. Diese müssen allerdings miteinander verzahnt werden, wenn sie einen deutlichen Effizienzvorsprung bewirken sollen. Jeder dieser fünf Paradigmenwechsel kann sein Potenzial erst dann ausspielen, wenn er mit den anderen kombiniert wird, deren mittelbare und unmittelbare Auswirkungen die Voraussetzungen für einen echten Fortschritt sind. Jede der fünf Methoden geht auf eine bestimmte Weise an das Verifikationsproblem heran. Ein dramatischer Fortschritt ist nur durch eine revolutionäre Umwälzung möglich, die sich auf alle diese Elemente gemeinsam stützt.



Design-In von Displays

36

Die Auswahl und Implementation eines Displays scheinen auf den ersten Blick keine großen Herausforderungen zu bergen.



In der Praxis können jedoch Probleme gerade in Hinblick auf die geplante Lebensdauer und Darstellungsqualität auftauchen, wenn nicht einige grundlegende Kriterien bei der Selektion und dem Design-In berücksichtigt werden.

Reverse Engineering

48

Rund 60 % aller Softwareprojekte sind Wartungs- oder Weiterentwicklungsarbeiten in denen der Entwickler meist mit fremdem, historisch gewachsenen Code und nicht selten mit nicht ausreichender oder fehlender Dokumentation konfrontiert wird. Reverse Engineering verspricht in diesen Fällen ein gangbarer Lösungsweg zu sein. In wieweit diese Methode wirklich helfen kann, hängt wesentlich von der Funktionalität entsprechender Werkzeuge ab.

Verifikationsstrategie – Evolution oder Revolution?

Erfahrungsbericht einer Design-Group

Die Verifikationsmethodik war in den vergangenen Jahren durch nicht weniger als fünf Paradigmenwechsel geprägt. Diese müssen allerdings miteinander verzahnt werden, wenn sie einen deutlichen Effizienzvorsprung bewirken sollen. Jeder dieser fünf Paradigmenwechsel kann sein Potenzial erst dann ausspielen, wenn er mit den anderen kombiniert wird, deren mittelbare und unmittelbare Auswirkungen die Voraussetzungen für einen echten Fortschritt sind. Jede der fünf Methoden geht auf eine bestimmte Weise an das Verifikationsproblem heran. Ein dramatischer Fortschritt ist nur durch eine revolutionäre Umwälzung möglich, die sich auf alle diese Elemente gemeinsam stützt. AKIVA MICHELSON



Akiva Michelson ist Pre-silicon Verification Manager bei der Intel Communication Group Jerusalem (ICGj). Den Beitrag erstellte der Autor für Verity.

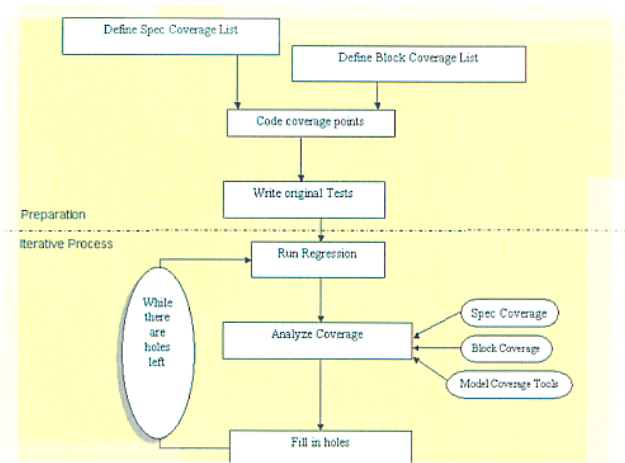


Abb. 1: Ablauf des coverage-orientierten Testens

Das Ziel der Anstrengungen ist es, während eines Projekts mit weniger Verifikations-Ressourcen auszukommen, die Verweildauer des Pre-Silicon in der kritischen Designphase zu verkürzen und das Fehleraufkommen im

„First Silicon“ zu reduzieren. Hierfür bedient sich die Communication Group von Intel folgender Methoden:

- ▶ Hardware-Verifikationssprachen
- ▶ Zufallstests

- ▶ Modulverifikation und Tests auf Modul-Ebene
- ▶ Coverage-orientiertes Testen
- ▶ Erstellen und Wiederverwenden von Design-Mustern

Hardware-Verifikationssprachen

Die erste wichtige Änderung betrifft die Hardware-Verifikationssprache. In Intel-Designzentrum entschieden sich der Verantwortlichen für ‚Specman Elite‘ und die Verifikationssprache ‚e‘ von Verisity, um die RTL-Verifikationsumgebungen zu erstellen und die Prüfprozesse zu automatisieren. Als greifbare Vorteile ergaben sich abstraktere Tests, On-the-Fly-Checks, abstraktere Datentypen und die Möglichkeit zur vermehrten Wiederverwendung von Verifikations-Komponenten. Um den geforderten Quantensprung zu erreichen, reicht dies jedoch noch nicht.

Die Codierung von Specman Elite kann von Hard- und Software-Designern vorgenommen werden. Allerdings bedarf es eines versierten Anwenders, um effektiv zu arbeiten. Da diese kundigen Anwender jedoch kaum verfügbar waren, bereitete es Probleme, das hohe Maß an Erfahrung aufzubauen, über das die RTL-Designingenieure des Teams verfügen.

Zufallstests

Auch Zufallstests sind mit dem Einsatz von Specman Elite möglich. Zunächst beschränkte sich die Zufälligkeit in den durchgeführten Tests auf die Paketlängen und die Daten in den Paketen. Mit zunehmendem Fortschritt begannen das Designteam jedoch, in das Design der Umgebung umfangreichere Zufallskomponenten einzubauen.

Zufallstests setzen eine robuste Umgebung voraus, erfordern eine Möglichkeit, Aussagen über die jeweiligen Aktionen einzuholen und erscheinen für den Designer zunächst weniger intuitiv.

Modulverifikation

Zur Modulverifikation gehört das Erstellen von Umgebungen und das Testen der Module als einzelne, isolierte Funktionseinheiten mit dem Ziel, die Integration und das Testen reibungsloser zu gestalten. Die Integration wird erleichtert, weil die einzelnen Module zuvor einzeln ausgetestet werden. Das Testen verläuft außerdem reibungsloser, da sich Fehler stark auf die Testzeit auswirken.

Coverage-orientiertes Testen

Das coverage-orientierte Testen (Abb. 1) ist ein iterativer Prozess. Der Begriff bezeichnet das

Lenken der Testabläufe auf der Basis von Informationen über die Funktionsabdeckung. Von Anfang an kann das Team sehen, wie effektiv eine Testsequenz das vorliegende Design abdeckt und wie gut der Test voranschreitet. Das bisher übliche Definieren von Testsituationen muss durch die Festlegung von Coverage-Punkten ersetzt werden. Wird hierbei methodisch vorgegangen, werden Testüberlappungen ebenso vermieden wie ungeprüfte Bereiche. Schließlich werden auch Rückmeldungen darüber ausgegeben, welchen kritischen Bereichen besondere Aufmerksamkeit gewidmet werden muss.

Design-Muster

Zu den am häufigsten wiederverwendeten Mustern gehören Busfunktionsmodelle (BFM). In verschiedenen Projekten immer wieder einsetzbare BFMs tragen zur Verringerung des Ressourcenaufwands für die Verifikation bei, reduzieren die Integrationszeit und geben dem Ingenieur eine Möglichkeit, vor Fertigstellung der gesamten Umgebung einfache Szenarien zu gestalten. Für einen wirklichen Quantensprung ist es jedoch notwendig, dass der Ingenieur eine Reihe weiterer sinnvoller Paradigmen für Probleme anderer Art findet. In Bereichen, in denen das Team selbst kein Know-how besitzt, sollten andere Gruppen, Berater und weitere Ressourcen genutzt werden, damit diese Muster in den Wissensbestand für das vorliegende Projekt aufgenommen werden können.

Die nachfolgende Aufstellung nennt einige Paradigmen, die von einem Projekt zum anderen übertragen oder von anderen Gruppen übernommen werden können. Jede Variante geht das Testbench-Problem aus einem anderen Blickwinkel an.

- ▶ Verzeichnisstruktur
- ▶ Projektdateien
- ▶ Tests; Erscheinungsbild der Test-Files; Testablauf

- ▶ Register: Modell zum Spiegeln, zum Verfolgen von Änderungen und zum Abstrahieren der Schreib/Lese-Zugriffe
- ▶ Modellierung von Puffern mit komplizierten Datenstrukturen zum Prüfen von Scatter/ Gather-DMAs
- ▶ Allgemeine Umgebungen auf Modul-Ebene
- ▶ Modulare Busschnittstellen

Die Integration eines neuen Projekts sollte folgende Elemente umfassen: Zuordnung der Signale zu wiederverwendeten BFMs, Import aller relevanten Datenstrukturen aus existierenden Projekten, Ausfüllen von Tabellen über das Layout von Registern und die Abwicklung von Reset- und Initialisierungsprozessen sowie das Hochfahren des Prüflings in verschiedenen Betriebsarten im Verlauf weniger Tage. Das Team sollte sich darauf konzentrieren, was das neue Projekt von den bisherigen Projekten unterscheidet.

Nebeneffekte

Diese Methoden haben eine Reihe unerwünschter Nebeneffekte, die häufig ausschlaggebend für den Erfolg anderer Lösungen sind. Einige der von der Intel Communication Group festgestellten Nebeneffekte sind im Folgenden aufgeführt.

Zufallstests können umgebungsbedingte Bugs schneller ausfiltern und führen zu übersichtlicheren, robusteren Modulen und Umgebungen. Dies wiederum erleichtert die Wiederverwendung und ermöglicht einen reibungslosen Ablauf des Coverage-orientierten Testens.

Das Erstellen modulatorientierter Umgebungen hat im Wesentlichen zwei Auswirkungen. Erstens erhält das Verifikations-Team ein fundiertes Verständnis der Mikroarchitektur des Designs. Zweitens ist es eine wirksame Lösung zum Bestimmen der Coverage-Punkte und internen Checks (Festlegungen und Annahmen) auf der RT-Ebene. Die fundierte Kenntnis



Tabelle 1: Auswirkungen/Bewertungskriterien der isoliert eingesetzten Komponenten

Isoliert benutzte Komponenten	Kürzere Validierungszeit	Weniger Validierungsressourcen	Reibungslosere Integration	Reibungsloseres Testen	Höhere Transparenz	Höhere Qualität	Zieht bessere Verifikationsingenieure an	Probleme
Abstrakteres Modellieren	Nein	Ja	Nein	Ja	Nein	Nein	Ja	Erfahrenes Personal ist knapp
Zufallstests	Nein	Ja	Nein	Nein	Nein	Ja	Nein	Funktioniert nur, wenn Umgebung und RTL stabil sind und die Testbench selbstprüfend ist.
Coverage Orientierung	Ja	Nein	Nein	Nein	Ja	Ja	Nein	Das Definieren und Codieren von Coverage Points führt zu keinen schnellen bzw. direkten Ergebnissen.
Modulweises Testen	Ja	Nein	Ja	Ja	Nein	Ja	Nein	Aufwand beim Erstellen der Umgebungen, verzögerte Integration.
Wiederverwendungs Paradigmen	Ja	Ja	Ja	Ja	Nein	Ja	Ja	Aufwand beim Aufbau der Wissensbasis

des Designs hat im Verbund mit den White-Box-Checks und der Abdeckung des Designs verschiedene Effekte:

- ▶ Das Verifikations-Team wird bei der Vermeidung der Fehlalarme unterstützt, die sonst bei Zufallstests an der Tagesordnung sind.
- ▶ Man gelangt wesentlich schneller als mit externer Evaluierung zum Kern eines Problems.
- ▶ Es stehen erheblich fundiertere und präzisere, für Coverage-orientierte Tests benötigte Aussagen über die Funktionsabdeckung zur Verfügung.
- ▶ Durch die Verfügbarkeit von Checks, die auf der Architektur-Ebene nicht codierbar sind, wird die Fehlersuche mit Zufallstests aufgewertet.

Das Definieren von Coverage-Points führt auf wesentlich organisiertere Weise zu einer detaillierten Feature-Liste. Dies kann zudem in einer sehr frühen Phase des Designs erfolgen und liefert dem Umgebungs-Entwickler Angaben darüber, wie robust das Modul sein muss, um zulässige und nicht zulässige Stimulus-Signale zu erzeugen.

Durch das modulweise Testen und das frühzeitigere Ausfiltern von Logikfehlern hat der Umgebungs-Entwickler die Möglichkeit, seine Interfaces vor der Integration des kompletten Chips zu prüfen. Der Designer kann sich außerdem unter Verwendung einfacher, in ihrer Gänze

verstehbarer Umgebungen mit Specman Elite vertraut machen.

Die abstrakte Modellierung hilft bei der Heranbildung und Bindung von Verifikations-Experten. Dabei führt der Entwurf und das Erstellen von Design-Patterns zu einer Erweiterung des Erfahrungsschatzes, was erneut fähigere Ingenieure anzieht und hält.

Zusammenführung aller Elemente

Intel wandte diese Umwandlung in zwei Organisationen an, mit denen das Designzentrum arbeitete. Beide hatten unter Qualitätsmängeln und nicht zeitgerechter Lieferung gelitten. Der erste Schritt war die Einsicht in die Notwendigkeit, auf ein anderes Tool umzustellen. Die Intel Communication Group setzt auf Specman Elite, da hiermit bereits Erfahrungen gesammelt werden konnten.

Der nächste Schritt war die modulweise Verifikation. Das Design-Team legte dem Zeitplan die modulatorientierte Verifikation zugrunde, da sie die Voraussetzung für eine reibungslose Integration ist. Hier bot sich die Gelegenheit, die Verifikations-Ingenieure schnell mit dem Tool vertraut zu machen und White-Box-Checker und den Coverage-Aspekt einzuführen.

Im nächsten Schritt ging es darum, eine Schulung für die älteren Verifikations-Ingenieure abzuhalten. Hier hatten sie die Möglichkeit, sich

mit den existierenden Design-Patterns vertraut zu machen und ihre Ideen für zusätzliche Design-Patterns einzubringen. Die Verantwortlichen empfahlen außerdem die Auseinandersetzung mit eVCs, die von verschiedenen Herstellern von Validations-Komponenten angeboten werden, sowie mit der Literatur, die die existierenden Design-Patterns beschreibt.

Bei reibungsloser Integration ermutigte das Unternehmen die RT-Designer, weiter auf der Block-Ebene und mit Bugfixes zu arbeiten, während sie gleichzeitig zu einem Blick auf die Coverage-Resultate veranlasst wurden. Das Verifikations-Team arbeitet daran, die verbleibenden Features in die Umgebung zu integrieren und die integrierten Features in Zufalls-Szenarien einzusetzen. Die Checks und Coverage-Aussagen aus der Block-Ebene werden auf die oberste Hierarchie-Ebene übertragen und dort verwendet.

Als sämtliche Features der Umgebung erfolgreich in das Design integriert waren, konnten die Zufallstests mit voller Geschwindigkeit aufgenommen werden. Die Verifikations-Ingenieure arbeiteten mit den RT-Designern an den Constraints, die für die Zufallstests benötigt werden, um die gesteckten Coverage-Vorgaben zu erreichen. Sobald die geforderte Abdeckung erreicht ist, kann die abschließende RT-Beschreibung zur Verwendung in der Back-End-Phase freigegeben werden.