

Bild: MCD Elektronik

Parallele Testabläufe: Mit dem von MCD entwickelten Software TestMate lassen sich mehrere Testsequenzen innerhalb einer Instanz parallel und asynchron ausführen.

SOFTWARE VERBESSERT AUSLASTUNG VON PRÜFHARDWARE

Prüfdurchsatz steigern durch parallelisierte Testabläufe

Bei automatisierten Prüfprozessen entstehen Wartezeiten nicht nur durch die Messung selbst, sondern auch durch blockierte Ressourcen und serielle Ablaufstrukturen. Eine geeignete Softwarearchitektur kann unabhängige Testschritte parallel ausführen und dadurch die vorhandene Prüfhardware besser auslasten.



Bild: MCD

VERFASST VON
Sebastian Lakotta
Head of Development
MCD Elektronik

Steigende Variantenvielfalt und kürzere Produktlebenszyklen erhöhen den Druck auf die Elektronikproduktion. Prüfprozesse müssen unterschiedliche Produkte und Prüfaufgaben abbilden, dabei reproduzierbare Ergebnisse liefern und sich in automatisierte Fertigungsabläufe integrieren lassen. Entscheidend ist dabei nicht allein die Geschwindigkeit eines einzelnen Messgeräts. Auch die Organisation des gesamten Prüfablaufs beeinflusst den erzielbaren Durchsatz.

Viele Testsequenzen arbeiten grundsätzlich seriell: Ein Prüfschritt wird abgeschlossen, bevor der nächste beginnt. Dieses Ablaufmodell ist einfach zu beherrschen, kann aber zu unnötigen Wartezeiten führen. Muss eine Sequenz beispielsweise auf eine Antwort vom Messgerät, eine Stabilisierung des Prüflings oder die Freigabe einer gemeinsam genutzten Ressource warten, bleiben nachfolgende, eigentlich unabhängige Arbeitsschritte unter Umständen ebenfalls stehen. Eine mögliche Gegenmaßnahme ist die Parallelisierung von Testabläufen. Dabei werden unabhängige Sequenzen oder Untersequenzen gleichzeitig ausgeführt. Die Herausforderung besteht darin, die parallelen Abläufe so zu koordinieren, dass sich die einzelnen Prüfungen weder gegenseitig beeinflussen noch auf gemeinsam genutzte Ressourcen unkontrolliert zugreifen.

Parallelisierung auf mehreren Ebenen

Parallelisierung kann auf unterschiedlichen Ebenen eines Prüfprozesses stattfinden. Mehrere vollständige Testsequenzen lassen sich beispielsweise gleichzeitig ausführen. Damit können unterschiedliche Prüflinge oder mehrere Prüfstationen parallel bearbeitet werden.

Darüber hinaus ist auch eine Parallelisierung innerhalb einer einzelnen Sequenz möglich. Sind beispielsweise

zwei Prüfschritte voneinander unabhängig, müssen sie nicht zwangsläufig nacheinander ausgeführt werden. Eine Software kann sie gleichzeitig starten und erst dann mit einem nachfolgenden Schritt fortfahren, wenn beide Ergebnisse vorliegen. Bei der Prüfung mehrerer Leiterplatten innerhalb eines Panels kann dies beispielsweise bedeuten: Gemeinsame Vorbereitungsschritte werden zunächst für mehrere Prüflinge durchgeführt. Anschließend werden die eigentlichen Prüfungen der einzelnen Leiterplatten parallel gestartet. Abhängig von den Anforderungen können einzelne Abläufe später wieder zusammengeführt oder weitere Schritte parallel ausgeführt werden.

Dabei ist Parallelisierung nicht mit einer beliebigen Vervielfachung von Prüfprozessen gleichzusetzen. Die mögliche Anzahl gleichzeitig laufender Abläufe hängt unter anderem von den verfügbaren Messgeräten, Netzteilen, Kommunikationsschnittstellen und Kontaktierungseinrichtungen ab. Auch die Prüflinge selbst können sich gegenseitig beeinflussen, etwa wenn sie eine gemeinsame Versorgung oder eine gemeinsame Messressource nutzen.

Synchronisation verhindert Konflikte

Sobald mehrere Abläufe gleichzeitig ausgeführt werden, müssen Abhängigkeiten explizit beschrieben werden. Synchronisationspunkte legen fest, wann ein Ablauf auf einen anderen warten muss oder wann mehrere Abläufe gemeinsam fortgesetzt werden können. Das betrifft nicht nur die Reihenfolge von Prüfschritten. Auch gemeinsam genutzte Ressourcen müssen verwaltet werden. Dazu zählen beispielsweise Netzteile, Messgeräte und Kommunikationsschnittstellen. Greifen mehrere Testsequenzen auf dieselbe Ressource zu, muss die Testsoftware den Zugriff koordinieren. Andernfalls können sich Messungen beeinflussen oder Befehle in einer nicht reproduzierbaren Reihenfolge an ein Gerät übertragen werden.

Eine Zustandsmaschine kann dabei helfen, die verschiedenen Zustände eines Prüfprozesses strukturiert abzubilden. Sie beschreibt, unter welchen Bedingungen ein Ablauf von einem Zustand in den nächsten wechselt. Bei parallelen Prozessen muss eine solche Zustandslogik zusätzlich berücksichtigen, welche Sequenzen bereits abgeschlossen sind, welche Ressourcen belegt werden und an welchen Punkten eine Synchronisation erforderlich ist. Die technische Aufgabe besteht somit nicht nur darin, mehrere Threads oder Sequenzen gleichzeitig zu starten. Die Software muss auch definieren, welche Abläufe tatsächlich unabhängig sind und an welchen Stellen eine kontrollierte Reihenfolge erforderlich bleibt.

Mehrere Prüfstationen zentral koordinieren

Eine weitere Anwendung ist die zentrale Steuerung mehrerer Prüfstationen. Statt jede Station als vollständig un-

Good Will Instrument Co. Ltd
00406769-004
210.0 mm x 148.0 mm (Format: 12A4)

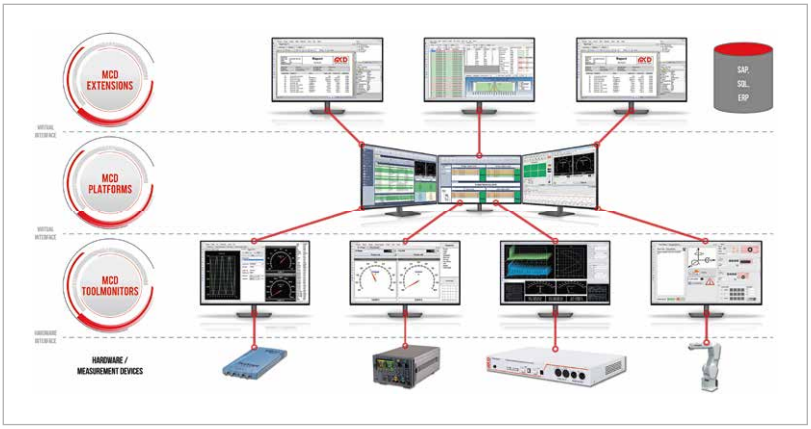


Bild: MCD Elektronik

Bild 1:
Die TestSuite verbindet zentrale Softwarekomponenten mit Toolmonitoren, Mess- und Prüfgeräten sowie übergeordneten Systemen.

abhängiges Testsystem zu betreiben, können Testabläufe, Konfigurationen und Ergebnisse in einer gemeinsamen Softwareumgebung verwaltet werden. Während an einer Station ein Messgerät auf eine Antwort wartet, kann an einer anderen Station bereits ein weiterer Prüfling vorbereitet oder geprüft werden. Die einzelnen Stationen benötigen dabei weiterhin ihre jeweiligen Prüfabläufe und Hardwareanbindungen. Die übergeordnete Software koordiniert jedoch die laufenden Prozesse und stellt die Ergebnisse in einem gemeinsamen Kontext bereit.

Eine zentrale Verwaltung kann außerdem Änderungen an Testabläufen und Konfigurationen vereinfachen. Voraussetzung ist allerdings, dass die Systeme sauber voneinander abgegrenzt sind. Ein Fehler in einer zentral verwalteten Sequenz kann sich prinzipiell auf mehrere Stationen auswirken. Deshalb sind Versionsverwaltung, Freigabeprozesse und eine nachvollziehbare Dokumentation der Testkonfigurationen wichtige Bestandteile einer solchen Architektur.

Der wirtschaftliche Nutzen der Parallelisierung entsteht vor allem dann, wenn unterschiedliche Prüfprozesse zeitweise auf dieselbe Hardware oder auf dieselben Software-Schnittstellen zugreifen können. Wartet ein Ablauf auf das Ende eines Messvorgangs, lassen sich andere unabhängige Schritte ausführen. Dadurch können Leerlaufzeiten reduziert werden. Das bedeutet jedoch nicht, dass sich der Durchsatz grundsätzlich proportional zur Zahl der parallelen Sequenzen erhöht. Begrenzende Faktoren bleiben beispielsweise die Messdauer, die Zahl verfügbarer Ressourcen, die Umschaltzeiten, die Kontaktierung sowie die Synchronisationspunkte innerhalb des Prüfablaufs.



Bild: MCD Elektronik

Bild 2:
Bei MCD Comet handelt es sich um eine automatisierte Prüfstation mit Software zur Steuerung und Durchführung von Testabläufen.

Auch thermische oder elektrische Wechselwirkungen zwischen mehreren Prüflingen können eine parallele Prüfung einschränken.

Für die Auslegung einer Prüfanlage ist daher eine Analyse des gesamten Ablaufes erforderlich. Entscheidend ist, an welchen Stellen tatsächlich Wartezeiten entstehen und welche Ressourcen währenddessen frei oder blockiert sind. Erst daraus lässt sich ableiten, welche Schritte parallelisiert werden können und ob zusätzliche Hardware, eine veränderte Sequenzstruktur oder eine andere Ressourcenverteilung den größeren Effekt erzielt.

Rückverfolgbarkeit im Parallelbetrieb

Mit zunehmender Parallelisierung steigt die Bedeutung einer eindeutigen Zuordnung von Prüfling, Testsequenz, Prüfstation und Ergebnis. Die Testergebnisse müssen auch dann korrekt zugeordnet werden, wenn mehrere Abläufe gleichzeitig bearbeitet werden. Eine zentrale Softwareumgebung kann diese Zuordnung unterstützen, indem sie die laufenden Testsequenzen und ihre Ergebnisse in einem gemeinsamen Kontext verwaltet. Für die Praxis sind dabei unter anderem Zeitstempel, Prüflingskennungen, Versionsstände der Testprogramme und Informationen zur verwendeten Hardware relevant.

Diese Daten bilden die Grundlage für die spätere Qualitätsauswertung und Fehleranalyse. Wird beispielsweise ein Fehler nur unter bestimmten Bedingungen oder an einer bestimmten Prüfstation beobachtet, müssen sich die zugehörigen Prozessdaten nachvollziehen lassen. Parallelisierung darf deshalb nicht zu einer unübersichtlichen Sammlung voneinander getrennt gespeicherter Ergebnisse führen.

Umsetzung mit MCD TestMate

MCD Elektronik setzt den beschriebenen Ansatz mit MCD TestMate um. Die Software kann mehrere Testsequenzen innerhalb einer Instanz parallel und asynchron ausführen. Auch Untersequenzen innerhalb eines Prüfablaufs lassen sich parallel strukturieren. Synchronisationspunkte und eine Zustandsmaschine dienen dazu, Abhängigkeiten zwischen den einzelnen Abläufen abzubilden. Die Software kann dabei unterschiedliche Prüflinge, Prüfstationen und gemeinsam genutzte Ressourcen koordinieren. Mess- und Prüfgeräte werden über entsprechende Software-Schnittstellen in die Testumgebung eingebunden. Welche Schritte parallel ausgeführt werden können, hängt von der jeweiligen Prüfaufgabe, der Hardware und den Ressourcenabhängigkeiten ab.

MCD TestMate ist Teil der MCD TestSuite. Zu dieser gehören außerdem MCD Comet und MCD TestManager. Während MCD Comet für Testumgebungen vorgesehen ist, die auf NI TestStand basieren, wird MCD TestManager vor allem als Plattform für die Steuerung klassischer Prüfabläufe in Fertigungsumgebungen eingesetzt. Toolmonitore und weitere Erweiterungen übernehmen die Anbindung von Mess- und Prüfgeräten sowie anderen Systemen. Dadurch verschiebt sich der Ansatz zur Optimierung des Prüfdurchsatzes von der Einzelmessung auf die Ablaufebene: Nicht jedes Messgerät muss schneller werden, wenn sich Wartezeiten durch eine geeignete Parallelisierung vermeiden lassen. Voraussetzungen hierfür sind eine sauber modellierte Testlogik, eine kontrollierte Ressourcenverwaltung und eine lückenlose Zuordnung der Prüfergebnisse. *(heh)*