

AKADEMIE DER WISSENSCHAFTEN DER DDR
– Zentralinstitut für Kybernetik und Informationsprozesse –

11. Arbeitstagung
Entwurf von Schaltsystemen

Dresden, 16.–18. März 1982

Kurzfassungen der Vorträge

Berlin
März 1982

Zech, K.-A. (INT Berlin)
Über Möglichkeiten diagnosefreundlichen
Schaltungsentwurfs

0. Einleitung. Motivation

Infolge der ohnehin erheblichen und laufend steigenden Kosten für die Qualitätssicherung bei elektronischen Produkten stellt sich überall die Frage der effektiven Gestaltung von Prüfprozessen. Aufgrund der inzwischen sehr hohen Komplexität der Bauelemente und Teilprodukte ist diese nur durch Rechnereinsatz zu beantworten, und zwar in dem Sinne, daß einerseits die Möglichkeiten herkömmlicher Verfahren (mit wenig oder ohne Rechner) überhaupt erschöpft zu sein scheinen, andererseits Kostensenkungen nur durch den Einsatz intelligenter Maschinen erreichbar wird. Die Automatisierung stellt jedoch ihre Forderungen nicht nur an Prüftechnologen und Operateure, sondern insbesondere an die Prüfobjekte selbst. Darüber hinaus machen heutige Einsatzkriterien (Verfügbarkeit) eine sehr effektive Wartung notwendig, die mit kleinen Prüfgeräten auskommen muß. Auch dies bedingt eine Berücksichtigung der Diagnosebelange bereits beim Entwurf. Der Trend zur höheren Integration bietet jedoch neben Diagnoseproblemen auch neue Lösungsmöglichkeiten.

Der Beitrag versucht eine Zusammenfassung von Möglichkeiten, auf der Logikebene strukturell die Diagnosefreundlichkeit bzw. -fähigkeit zu verbessern.

1. Begriff der Diagnosefreundlichkeit

Diagnose (in Fertigung und Wartung) umfaßt sowohl die Prüfung (Fehlererkennung, Testung) als auch die Fehlerortung (-lokalisierung, Diagnose im engeren Sinne). Qualitativ ausgedrückt ist die Diagnosefreundlichkeit eine Schaltungseigenschaft, die folgendes ermöglicht bzw. vereinfacht:

- vollständige Prüfbarkeit
- gute Automatisierbarkeit der Diagnose, d.h. leichte Testgenerierung, kurze Testfolgen, einfaches Fehlermodell
- Nutzung verfügbarer Diagnosesysteme
- Ausschluß dynamischer Probleme (in der Diagnose)
- gute Lokalisierbarkeit (wenige Suchoperationen, hohe Genauigkeit) /26/

Maße für die Diagnosefreundlichkeit sind die Beobachtbarkeit und die Steuerbarkeit der inneren Situation des Objektes. Zu letzterer gehört die Initialisierbarkeit, die von besonderer Bedeutung für die Prüfautomatisierung ist. /8/ und /9/ geben Verfahren zur Bewertung der Prüfbarkeit (Testability) gegebener Schaltungen an.

Durch Berücksichtigung der Diagnose im Entwurf wird der Prüf- und Lokalisierungsprozeß mehr oder weniger aktiv durch die Schaltung unterstützt. In Abhängigkeit von der Gesamtkonzeption der Diagnose ist der diagnosefreundliche Entwurf auf allen Ebenen (System/Logik/Layout) möglich und nötig /7/. Vereinfacht könnte man passive und aktive Testhilfen unterscheiden /5/.

2. Passive und aktive Testhilfen

Passive Testhilfen sind Entwurfserweiterungen bzw. -veränderungen zum Zwecke einer vereinfachten Fehlerdiagnose bei Fremdtestung durch ein Prüfgerät o.ä., während solche zur Eigenbeteiligung eines Prüfobjektes bei der Testbereitstellung bzw. -generierung sowie der Ergebnisaus- und Bewertung aktive Testhilfen genannt werden können.

Passive Testhilfen auf der Layoutebene /10,11/ können die Fehlermodellierung erheblich vereinfachen, aber auch die physikalischen Fehlermöglichkeiten einschränken. Auf der Logikebene können passive Testhilfen nach Beendigung des funktionellen Entwurfs eingebracht werden. Sie können auch bereits während der Funktionsfestlegung einfließen, z.B. auf der Ebene der abstrakten Automaten beim Entwurf eines Steuerteils /2/. Einige Forderungen und Ansätze haben hier grundsätzliche Bedeutung:

- Vermeidung von Redundanzen (sofern sie nicht eine funktionelle Bedeutung haben)
- Bevorzugung synchron-getakteter Betriebsweise
- Aufteilung der Schaltung in übersichtliche Funktionsgruppen
- Initialisierbarkeit der Speicherzustände
- Verwendung zusätzlicher Hardware wie blockierende Gatter, Meßpunkte usw.
- Verwendung linearer oder 2-Ebenenrealisierungen (vgl./2/; bisher nur akademische Bedeutung).

Auf der Systemebene spielen zusätzliche Aspekte eine große Rolle /7/.

Während die bisher genannten Hilfen Fehler ausschließen sollen (fault avoidance technique), ist die Fehlertoleranz darauf gerichtet, Fehler in ihrer Wirkung unschädlich zu machen oder abzuschwächen. Hierher gehören die Verwendung fehlererkennender und fehlerkorrigierender Codes, die im laufenden Betrieb eine gewisse Überwachungsfunktion ausüben und daher den aktiven Testhilfen zuzurechnen sind /7,12/. Die Ansätze zum Selbsttest sowie zur Selbstdiagnose und Selbstverifikation gehören auch zu aktiver Testhilfe. Elemente des Selbsttestkonzepts sind die eigenständige Testerzeugung und -auswertung. Sie können funktionell unter Ausnutzung der vorhandenen Fähigkeiten des Systems (z.B. mit intelligenten Bausteinen) oder strukturell durch speziellen Zuschnitt der Hardware realisiert sein. Die Diagnose erfolgt off-line (im Testmodus), on-line (im laufenden Betrieb) bzw. zyklisch (jeweils inaktive Teile werden getestet).

Die folgenden beiden Abschnitte stellen stichpunktartig bewährte Techniken und Regeln vor (vgl. hierzu die vorzügliche Übersicht von GRASON u.a./1/) sowie neuere Ansätze, die für die Integration erfolgversprechend erscheinen. Manche Möglichkeiten sind nur für Karteneinschübe einsetzbar.

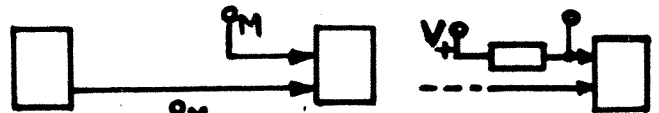
3. Passive Testhilfen auf Logikebene

3.1. Elemente zusätzlicher Hardware

3.1.1 Zusätzliche Prüf- und Meßpunkte

zum verbesserten Zugriff zu internen Leitungen:

- . Steuereingänge



- . Meßpunkte



- . kombiniert
(normalerweise verbunden)



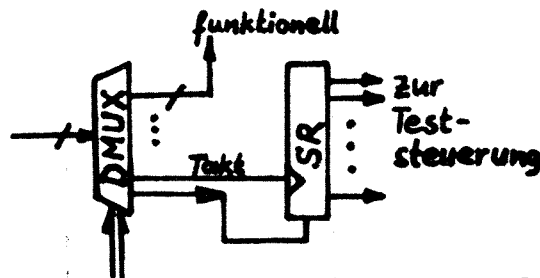
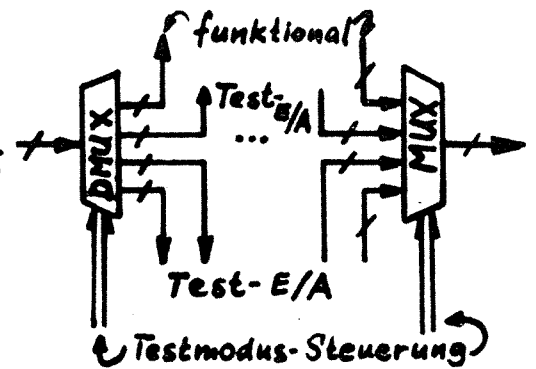
- . mit Tristate-Treiber



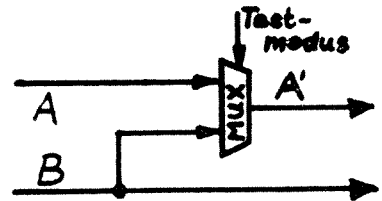
Zugriff über Steckverbinder, DIP-Sockel, Clips, Meßklemmen und -inseln

3.1.2. "Pin-Verstärkung" durch:

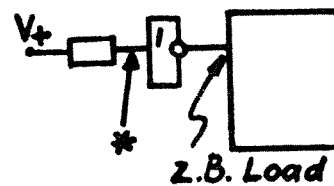
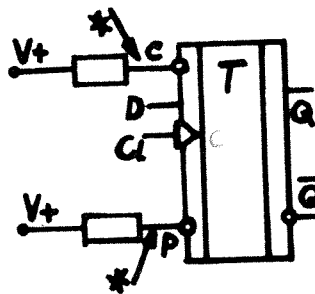
- Demultiplexer an Eingängen
- Multiplexer an Ausgängen
- interne Test-Schieberegister



- Zuschaltung von Steuer- und Meßpunkten:
 - Itg.A gut, Itg.B schlecht beobachtbar oder:
 - Itg.A schlecht, Itg.B gut steuerbar
- Sammlung von Meßpunkten über Parity-Bäume

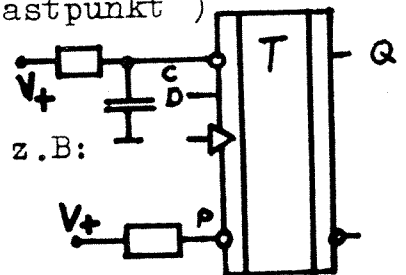


3.1.3. Lastwiderstände vor Direkteingänge und andere nicht benutzte, funktionelle Eingänge zwecks Stabilisierung und manueller Initialisierbarkeit durch "ziehen" auf 0

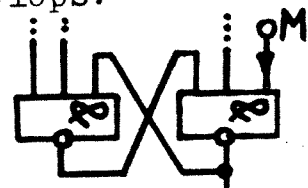
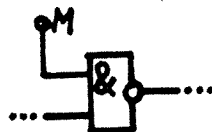


(* = Antastpunkt)

Selbstinitialisierung sollte vermieden werden, da schwer zu testen, z.B:

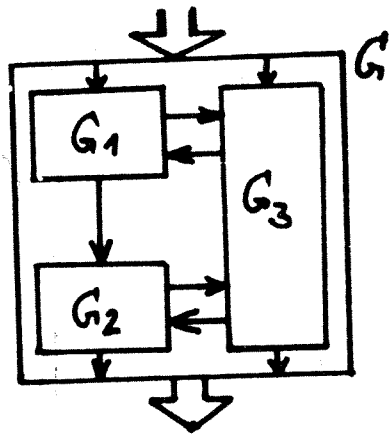


3.1.4. Blockierende Gatter (Steuerbare Negatoren und Buffer)
Zur Unterbrechung von Rückführungen (insbesondere bei großen Schleifen), zur Trennung rekonvergenter Signalwege und zur Blockierung von Flip-Flops:

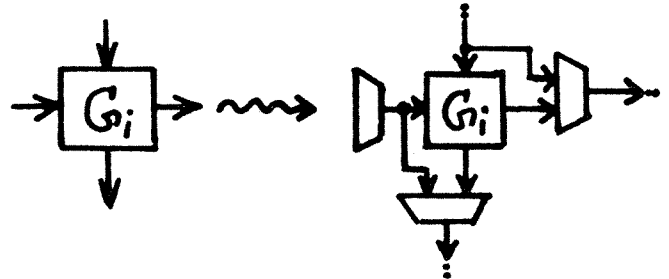


3.1.5. Zur Absicherung der Modularität muß die Schaltung in logisch voneinander isolierte Blöcke zerlegbar sein (Partitionierung). Sofern dies nicht bei der funktionellen Synthese berücksichtigt wurde, können obenge-

nannte Mittel wie Meß- und Steuerpunkte, blockierende Gatter und Tristate-Treiber Schaltungsteile trennen, d.h. direkt zugreifbar machen. Multiplexereinsatz erlaubt eine systematische Schaltungsaufteilung /23/:

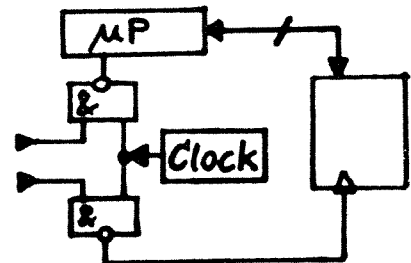
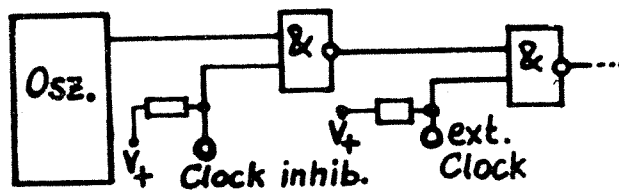


Durch Steuerung der Multiplexer kann jeder Block G_i in seinen normalen Funktionsmodus versetzt sowie im Testmodus direkt angesteuert oder übergangen werden.

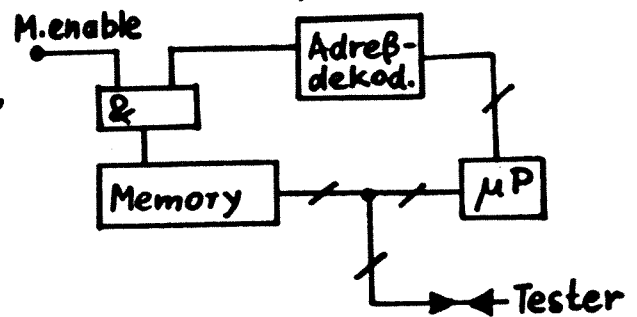


3.2. Hinweise zur Verwendung zusätzlicher Hardware gemäß 3.1.

- sequentielle Komponenten initialisierbar machen /3,13/
- Zählerketten steuerbar, beobachtbar und teilbar gestalten, insbesondere, wenn Takte abgeleitet werden
- Taktgeneratoren innerhalb des Prüflings:
abtrennbar gestalten: oder Takt untersetzbar:



- Monoflopausgänge beobachtbar u. steuerbar halten
- Testpunkte: zum Auftrennen großer Rückführungen und rekonvergenter Pfade, bei großem Fan-In und Fan-Out, zur Beobachtung von Display-Ansteuerungen (auch Lämpchen u.dgl.) und Tastaturen u.ä.
- Realisierung des Teile-Und-Herrsche Prinzips (vgl. 3.1.5.); z.B. auch: LSI-Komponenten trennbar montieren (Sockel, logisch); bei Mikroprozessoren: Speicher vom Datenbus trennbar machen: ↘
- Bei Bussen: Adreß- und Datenbusse sowie wichtige Steuersignale (z.B. READY, HOLD) gut beobachtbar und steuerbar gestalten



3.3. Verbesserung der Diagnosefreundlichkeit ohne zusätzliche Hardware

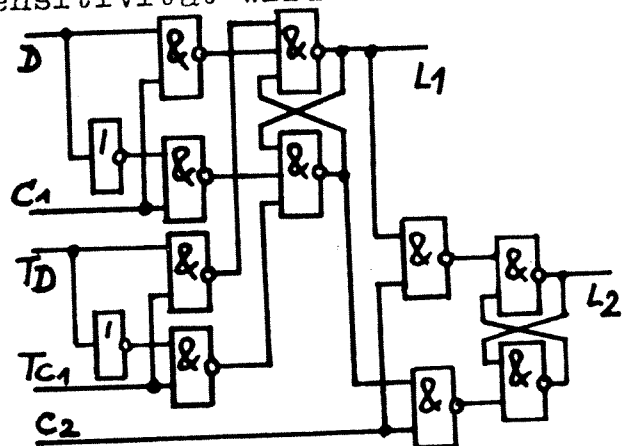
Neben der Empfehlung synchroner Betriebsweise sind hier zu nennen: Vermeidung von Monoflops und wired-Verbindungen; bei rückgeführten Signalen, Negatorketten und wired-Verbindungen: Funktionselemente möglichst in gemeinsamen Schaltkreis. Ferner erleichtert es die Diagnose, wenn sehr früh Simulationsmodelle für die verwendeten Moduln bereitgestellt werden.

3.4. Berücksichtigung vorhandener Testhardware u. -systeme /1,3/

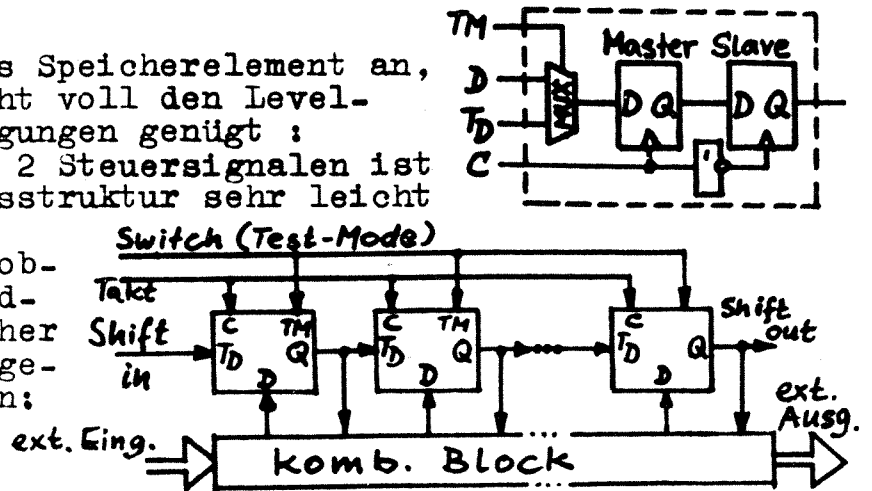
- Geschwindigkeit des Testers
- Störsicherheit der Signale am Steckverbinder (ggf. Buffer vorsehen; Eingangsänderungen können Sprünge induz.)
- unterschiedliche Logikfamilien sowie analoge und digitale Schaltungsteile physikalisch trennen
- Taktversorgung durch Tester ermöglichen
- Ordnung der Eingangssignaländerungen durch Prüfgerät bei Erstellung der Diagnosedaten beachten (Simulation)
- Testpunkte und Impulsfänger für relativ lange und rel. kurze Impulse vorsehen
- bei Karteneinschüben Verwendung eines einzigen Standardsteckverbinders
- Berücksichtigung Interface Prüfling-Prüfgerät
- unbenutzte log. Stifte mit Lastwiderständen stabilisieren (Störsicherheit)
- Tristate-Busse mit Lastwiderständen versehen (Simulation; stabile Prüfschwellen) /32/
- bei Karteneinschüben: regelmäßige IC-Plazierung für Pfadverfolgung (Platz für Antastung)

3.5. Der level-sensitive Scan-Design (LSSD) /1,2,14,15/

LSSD ist eine Verknüpfung des Scan-Path-Verfahrens /27, 28,29/, bei dem im Testmodus die Flip-Flops ein oder mehrere, seriell von außen les- und ladbare Schieberegister bilden, mit level-sensitiver Entwurfsweise /17/, die gewährleistet, daß Zustandsänderungen nicht von dynamischen Schaltbedingungen abhängen. Mittel hierzu sind hasardfreie Speicherschaltungen. Level-Sensitivität wird z.B. durch zwei überlappungsfreie Takte und das Master-Slave-Prinzip erreicht:

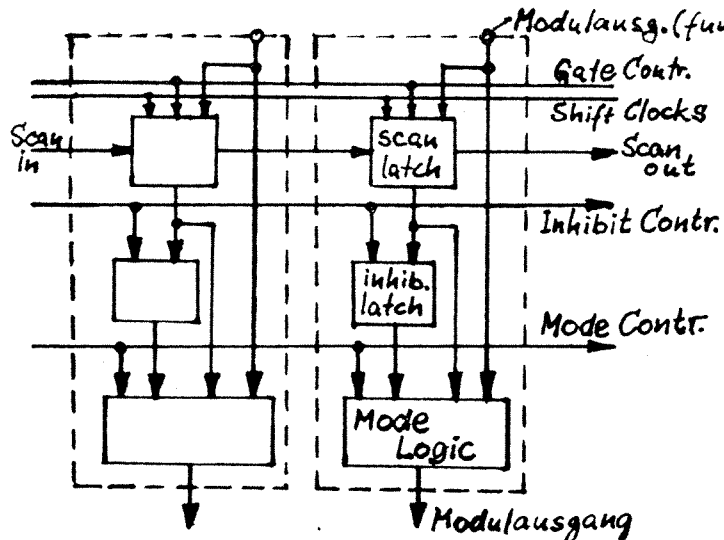


/16/ gibt folgendes Speicherelement an, das allerdings nicht voll den Level-Sensitivitätsbedingungen genügt :
 Bei Verwendung von 2 Steuersignalen ist folgende Schaltungsstruktur sehr leicht prüfbar, da alle algorithmischen Probleme auf die Behandlung kombinatorischer Schaltungen zurückgeführt werden können:



Testanalyseverfahren können kostengünstig gestaltet werden, da der kombinatorische Block sehr einfach simuliert werden kann /18/. Durch die Register wird die Schaltung gleichzeitig funktionell zerlegt; Blockgrößen von 1000 bis 2000 Gatter werden genannt. Als Nachteil wirkt der Zusatzaufwand an Logik, der mit 5-20%, gelegentlich noch höher, angegeben wird.

3.6. Testbare Moduln (selektive Steuerung) /1/



Ähnlich dem LSSD, werden hier Speicher zu Schieberegistern zusammengefasst, die Testinformationen an interne Schaltungspunkte transportieren. Jedoch werden nicht notwendig alle Speicher erfaßt. Die Register verlaufen peripher. Moduln können enabled oder in den Tristate versetzt und Rückführungen blockiert werden. Der Zusatzaufwand ist

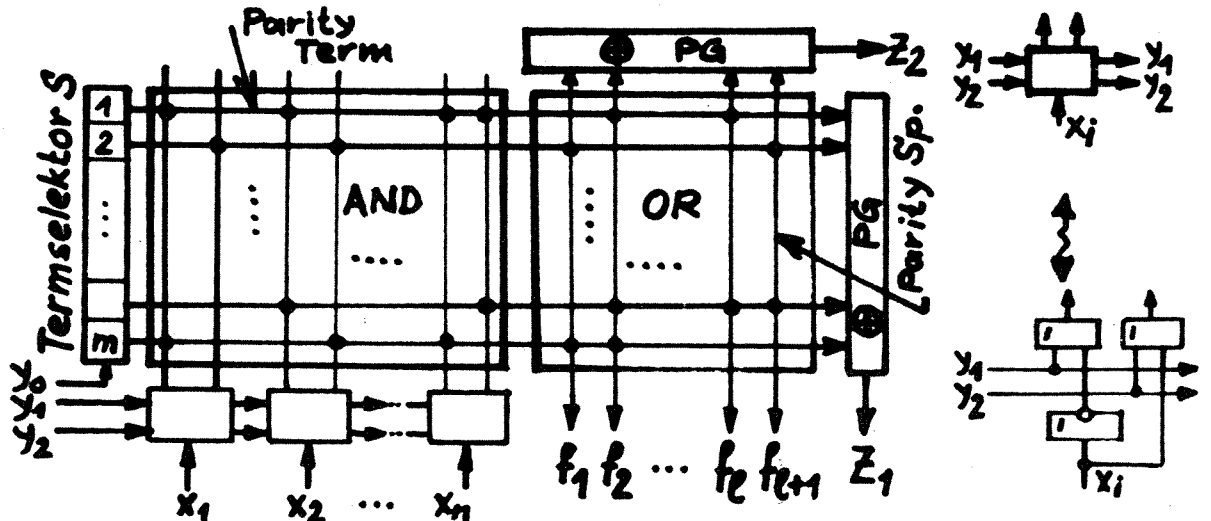
geringer als beim LSSD. Die Darstellung zeigt zwei benachbarte Registerzellen der selektiven Steuerung.

3.7. Direktzugriff zu Speicherelementen /19/

Hier ist jedes Speicherelement im Testmodus einzeln adressierbar. Die Testausgänge von Flip-Flops werden durch einen AND-Baum auf einen Testausgang des Schaltungsteils geführt. Speicher wie Flip-Flops können arrayartig angeordnet sein; für RAMs und ROMs ist diese Methode ebenfalls verwendbar. Je Flip-Flop werden 3-4 Gatter zusätzlich nötig sowie 6-20 Zusatzanschlüsse für die Schaltung. Eine ähnliche Methode wird in /20/ zur Isolierung ganzer Karteneinschübe angegeben.

3.8. PLA mit funktionsunabhängigen Tests

Von den verschiedenen Ansätzen zur prüfgünstigen PLA-Gestaltung sei die Methode von Fujiwara et al/21/ angeführt. (Dieses Konzept kann erweitert werden zur autonomen Testfähigkeit.) Die PLA wird erweitert um ein Schieberegister S, das die AND-Terme selektiert. AND-Array und OR-Array erhalten je einen Parity-Term. Zwei Parity-Bäume fassen alle AND-Terme bzw. Funktionen zusammen und liefern Prüfbits. Es läßt sich ein funktionsunabhängiger Testsatz für alle einfachen Fehler angeben.



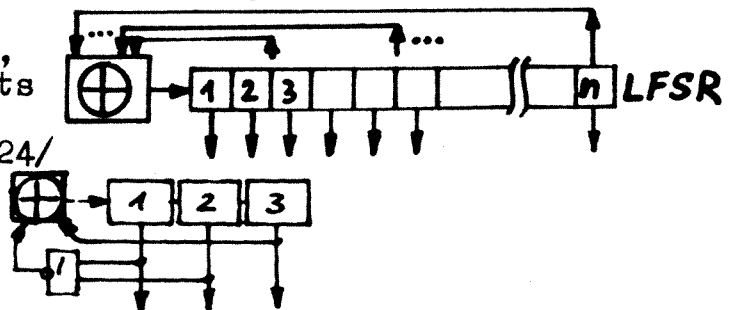
4. Aktive strukturelle Testhilfen

Durch Selbsttest (On-Chip/On-Board-Testing) werden Testprobleme in die Logik verlagert. Testen in Betriebsfrequenz wird möglich, Testgeräte können einfach gehalten werden, die Qualifikation des Personals ist weniger kritisch. Bei zunehmender Regularität der Entwürfe wird der Mehraufwand bei dem geringen Anteil Randomlogik nur eine beschränkte Rolle spielen.

4.1. Einige Elemente struktureller Selbsttestmethoden

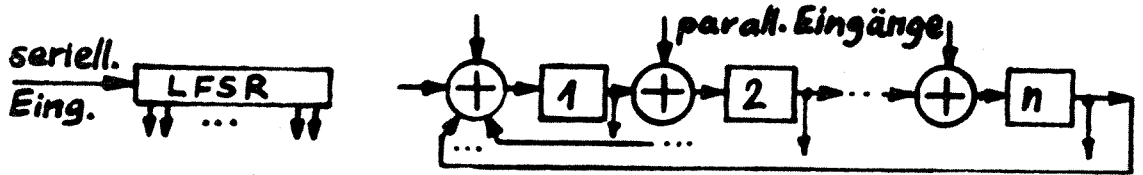
Neben abgespeicherten Tests bei (funktionellem) Selbsttest spielen folgende Generatoren eine große Rolle:

- pseudostochastischer Testgenerator (PSTG), der fast alle 2^n Tests erzeugt
- modifizierter PSTG /24/ zur Erzeugung aller 2^n Tests, z.B.:



- n-stufiger Zähler

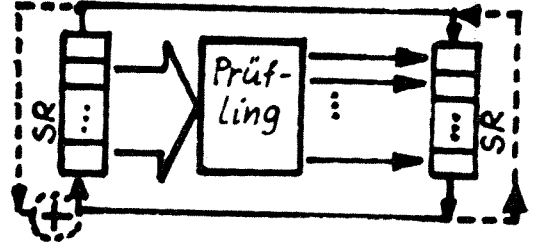
Für eine On-Chip-Diagnosedatenauswertung bzw. -kompression werden folgende beiden Registertypen eingesetzt:



Auch Zähler zum Aufsummieren von Übergängen (Transition Count), der 1-Werte einer Bitfolge usw. werden verwendet.

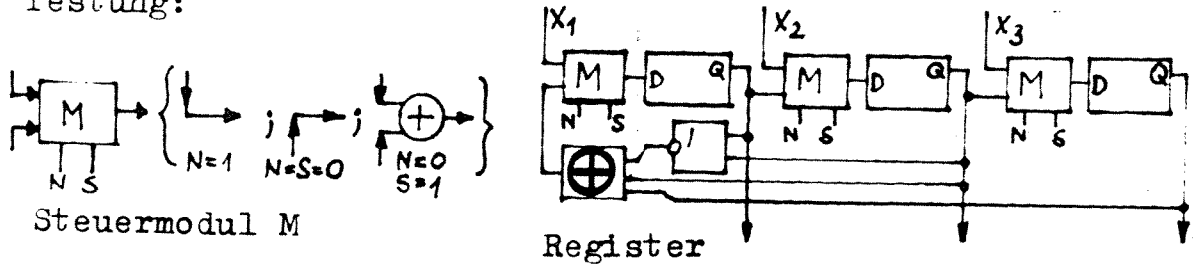
4.2. Beispiele aktiver Testhilfen

4.2.1. Ansatz von EIKI u.a. /30/
Die Speicherelemente werden im Testmodus zu PSTGs oder einfachen Schieberegistern zusammengefaßt und dienen als Generatoren und Kompressoren, wobei diese Funktionen austauschbar sind.



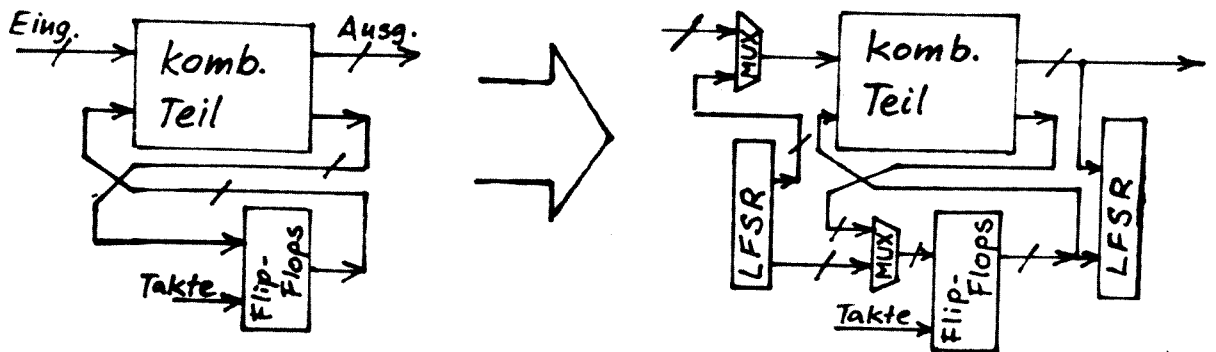
4.2.2. BIT (Built-In-Test) /34/
Testgenerierung im jeweiligen Subsystem durch PSTG; modulweises Ableiten von Bewertungsbits; Verwendung von Codes; kollektive Auswertung der Prüfbits

4.2.3. Rekonfigurierbares LFSR /24/
Erweiterung von BILBO /25,31/ zur ausschöpfenden Testung:



Durch Wahl der Steuergrößen N und S kann der Normalbetrieb, Datenkompression oder Testgenerierung (alle 2^n Tests) eingestellt werden.

4.2.4. Ausschöpfendes Testen mit erzwungenen Zuständen /24/



4.2.5. Verwendung adressierbarer Speicher gemäß 3.7.
und Adressenberechnung durch Zähler oder LFSR

5. Literaturswahl

- /1/ Grason u.a.: Digital Testgeneration & Design for Testability. Proc. 17th DA conf. Juni 1980, 175-189
- /2/ Albrecht; Methoden des prüfgünstigen Entwurfs digitaler Schaltungen. Studie Humboldt Univ.,Sekt.Math.,Juni'79
- /3/ Boswell: Designing Testability into Complex Logic Boards. Electronics 45(72) 116-119
- /4/ Bennetts u.a.: Fault Diagn. of Dig.Syst. Comp.J.14 H.2
- /5/ Bernstein: 3. Kolloquium Elektronikprüftechnol. THK 1980
- /6/ Buks: Elektronik Produktion & Prüftechnik 1981 H.5
- /7/ Reinert: Prüftheorie diskreter Systeme. Berlin 1979
- /8/ Grason: Proc. 16th DA Conf. 1979
- /9/ Bennetts u.a.: Int.Conf.Circuits & Computers, Okt.1980
- /10/ Crouzet u.a. Manuskript LAAS Toulouse 1981
- /11/ Galiay u.a.: FTCS 9
- /12/ Crouzet u.a.: FTCS9
- /13/ Bennetts u.a.: The Radio&Electronic Eng. 45(1975)11
- /14/ Williams u.a. 14th DA Conf.
- /15/ Healy; 1978 Semiconductor Test Conf. Okt.1978
- /16/ Funatsu u.a.: ebenda
- /17/ Eichelberger,E.B.: Method of Level-Sensitive Testing
A Functional Logic System. US-Patent 3781695 Sept'73
- /18/ Goel u.a.: 1980 IEEE Test Conference , Cherry Hill
- /19/ Ando: 1977 Semiconductor Test Symp.Cherry Hill,Okt.'77
- /20/ Yamada u.a. Proc 15th DA Conf. 1978
- /21/ Fujiwara u.a.: FTCS10, Tokyo, Okt. 1980
- /22/ Yajima, S. u.a.: Manuskript Mai 1981 Kyoto Univ.Japan
- /23/ Bozorgui-Nesbat u.a.: FTCS10
- /24/ McCluskey u.a.: 1980 IEEE Test Conference Cherry Hill
- /25/ Fasang: 1980 IEEE Test Conf. Cherry Hill
- /26/ Waltrich: ebenda
- /27/ Trischler: ebenda
- /28/ Huelters: Verfahren und Anordnung zur Fehlerdiagnose
bei taktgesteuerten Geräten.BRD-Patent 2111493,1971
- /29/ Williams u.a.: IEEE Transactions on Comp.22(1973)1
- /30/ Eiki u.a.: FTCS10 1980
- /31/ Koenemann u.a.: NTG Fachberichte Band 68(1979)
- /32/ Grason: Design Aids & Hardware Testing of Mikro-
processor System Circuit Packs. Manuskript 1980
- /33/ Eichelberger u.a. 14th DA Conf 1977
- /34/ Benowitz u.a.: IEEE Transactions on C-24(1975)5

(Ein ausführlicheres Verzeichnis kann vom Autor nachgefordert werden)