

Parametrisierte Speicherzellen zur Unterstützung des Selbsttests mit optimierten und konventionellen Zufallsmustern

Frank Kesel, Hans-Joachim Wunderlich
Universität Karlsruhe
Institut für Rechnerentwurf und Fehlertoleranz
(Prof. Dr. D.Schmid)
D-7500 Karlsruhe 1, Haid-und-Neu-Straße 7

Kurzfassung

Viele Selbsttestverfahren für hochintegrierte Schaltungen beruhen auf der Erzeugung von Zufallsmustern mit rückgekoppelten Schieberegistern. Oft wird jedoch für eine ausreichende Fehlererfassung eine unwirtschaftlich große Menge von Zufallsmustern benötigt, falls diese gleichverteilt erzeugt werden. Mit ungleichverteilten Zufallsmustern kann die Testlänge entscheidend reduziert werden, ein entsprechendes Selbsttestkonzept wurde als GURT (Generator of Unequiprobable Random Tests) in [Wund87b] vorgeschlagen. Im vorliegenden Beitrag werden Grundzellen zur Synthese von Registern nach dem GURT-Prinzip vorgestellt und die Probleme beim Entwurf eines entsprechenden Syntheseprogrammes diskutiert. Anhand eines Beispiels werden die Selbsttestkonzepte nach dem GURT- und nach dem BILBO-Prinzip verglichen.

1. Einleitung

Bei wachsender Integrationsdichte fallen die Vorteile des Selbsttests wie die schnelle Testdurchführung, die Möglichkeit eines Hochgeschwindigkeitstests und des Wegfalls der Testmusterberechnung immer stärker ins Gewicht. Der Selbsttest kann mit multifunktionalen Registern, sogenannten BILBOs (Built-In Logic Block Observer) [KMZ79], realisiert werden, die während des Tests als linear rückgekoppelte Schieberegister (LRSR) die Testmuster für den kombinatorischen Teil der Schaltung erzeugen und die Testantworten zu einer Signatur komprimieren. Die Arbeitsweise der BILBOs beruht auf der Polynomdivision mittels eines LRSR vom Standardtyp, dieselbe Funktion läßt sich auch mit einem modularen LRSR realisieren (Bild 1).

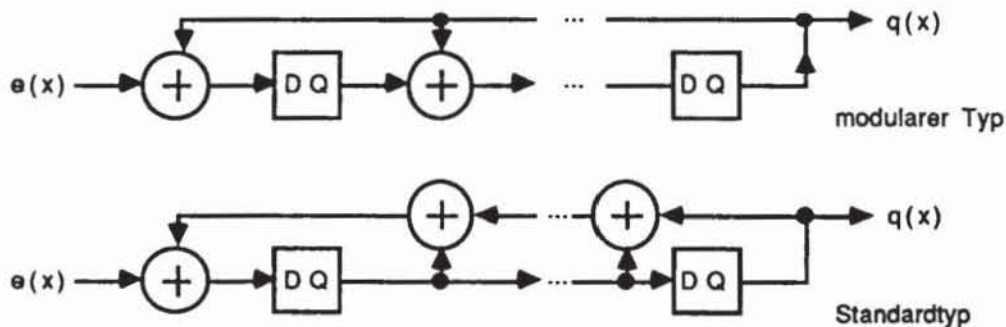


Bild 1: LRSR, modularer Typ und Standardtyp

Die Testlänge ist die notwendige Zahl von Testmustern, auf die die Schaltung korrekt reagieren muß, damit mit ausreichender Wahrscheinlichkeit Fehlerfreiheit angenommen werden kann. Sie hängt im wesentlichen nur von den Fehlern mit der geringsten Entdeckungswahrscheinlichkeit ab ([BaSa84], [Wu88]) und wächst daher im schlimmsten Fall exponentiell mit der Zahl der primären Eingänge. Durch den Test mit ungleichverteilten oder gewichteten Zufallsmustern [Wund87a] läßt sich die Testlänge verringern, hierbei wird jeder primäre Eingang des Schaltnetzes mit einer für ihn spezifischen, optimalen Wahrscheinlichkeit auf logisch 1 gesetzt. Verfahren zur Bestimmung optimaler Eingangswahrscheinlichkeiten wurden in [Wund87a], [Waic88] vorgestellt, eine entsprechende Schaltung hierfür wurde in [Wund87b] vorgeschlagen und GURT genannt.

Der Selbsttest nach dem BILBO-Prinzip wird bereits von einigen kommerziellen Systemen der Entwurfsautomatisierung unterstützt. In [Stas83] wurden kaskadierbare Grundzellen vorgeschlagen, aus denen BILBOs aufgebaut und als Standardzellen verwendet werden können. In [ArVa88] wird ein Compiler zur Erzeugung von Standardzellen für den Selbsttest beschrieben. Im vorliegenden Beitrag wird eine parametrisierte Standardzelle vorgestellt, welche den Selbsttest mit ungleichverteilten Zufallsmustern unterstützt. Damit können Klassen von Schaltungen durch Zufallsmuster getestet werden, für die das aufgrund unwirtschaftlicher Testlängen bislang nicht möglich war.

Im nächsten Abschnitt stellen wir die Architektur eines GURT's und die für ihn wesentlichen Parameter vor. In Abschnitt 3 präsentieren wir die Grundzellen, aus denen sich ein GURT aufbauen läßt, und im Abschnitt 4 skizzieren wir ein Programm zur Synthese des GURT nach vorgegebenen Parametern. Schließlich werden anhand eines Beispiels Hardware-Aufwand, Testlängen und Fehlererfassung diskutiert.

2. GURT

Eine ungleichverteilte Musterfolge kann durch Verknüpfung mehrerer unabhängiger, gleichverteilter Folgen erzeugt werden (vgl. Bild 2).



Bild 2: Erzeugung ungleichverteilter Zufallsfolgen

Wird ein modulares LRSR von einer Zufallsfolge gespeist, so lassen sich an geeigneten Punkten mehrere unabhängige Folgen abgreifen und zu einer ungleichverteilten Folge verknüpfen (Bild 3). Die Abgriffpunkte können so gewählt werden, daß eine unkorrelierte, pseudozufällige Folge entsteht [Golo82].

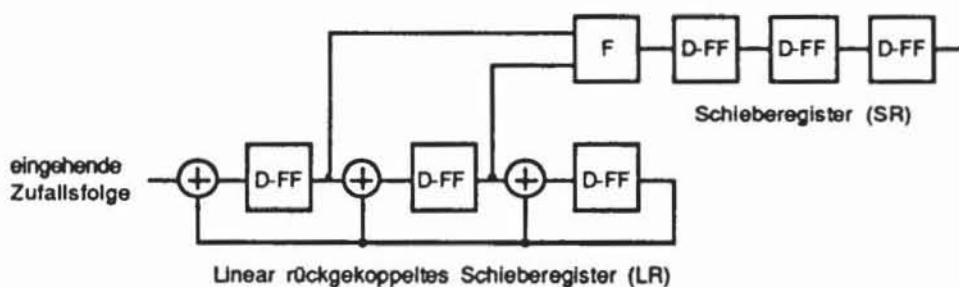


Bild 3: Erzeugung ungleichverteilter Folgen mit einem modularen LRSR

Die Schaltung aus Bild 3 erzeugt in den Bitpositionen des LR Musterfolgen der Wahrscheinlichkeit 1/2, und im anschließenden Schieberegister Musterfolgen der Wahrscheinlichkeit p oder 1-p, falls der invertierte Wert übernommen wird. Um jede Wahrscheinlichkeit im Raster von 1/8 zu realisieren, sind daher neben einem maximalen LRSR zur Erzeugung einer Zufallsfolge der Wahrscheinlichkeit $p=1/2$ mindestens drei Schaltungen nach Bild 3 nötig. Die Gesamtarchitektur zeigt Bild 4.

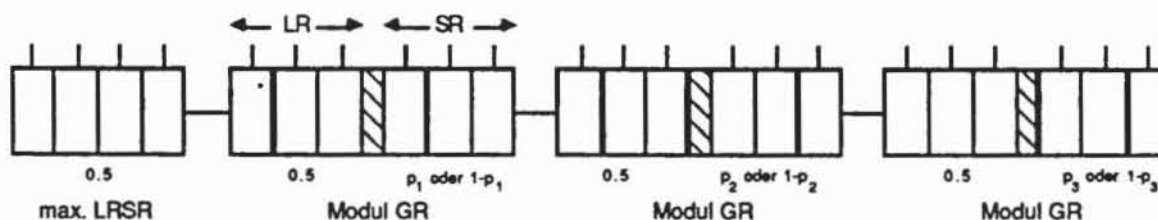


Bild 4: GURT-Architektur

Die oben gezeigte Registerkette soll nicht nur Muster erzeugen, sondern sie auch auswerten können. Die erwähnten Moduln GR sind daher etwas aufwendiger nach Bild 5 zu implementieren.

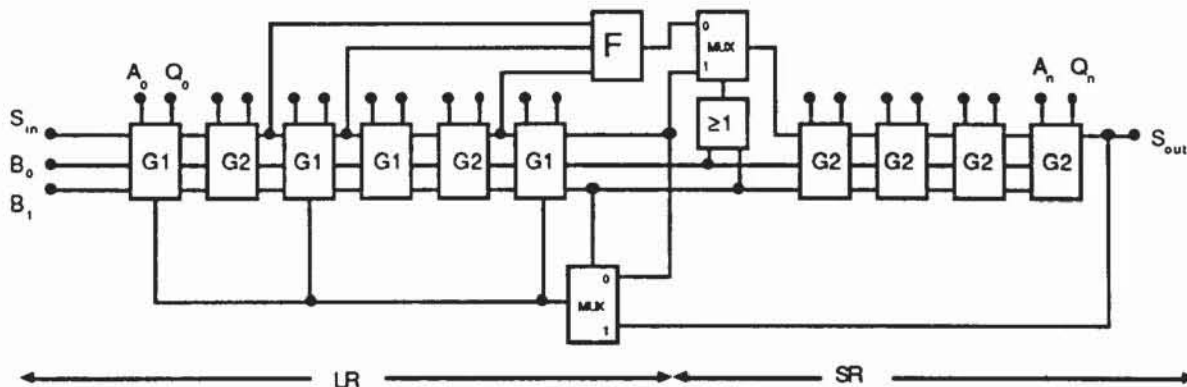


Bild 5: GURT-Modul GR

Wesentliche Bestandteile des GURT sind die Zellen G1 und G2 aus Bild 6, die jeweils aus einem D-Flipflop und einer vorgeschalteten Logik zur Steuerung der vier Betriebsarten (Signaturanalyse, Mustergenerator, Systemregister, Schieberegister) bestehen. Aus diesen beiden Zellen lassen sich modulare LRSR mit parallelen Eingängen aufbauen, wobei mit der Zelle G1 die Rückkopplung realisiert wird. Im SR-Teil kann durch Invertierung aus der von F erzeugten Wahrscheinlichkeit p für die Musterfolge die Wahrscheinlichkeit $1-p$ gewonnen werden. Durch mehrere Moduln GR mit unterschiedlicher Funktion F lassen sich beliebige Wahrscheinlichkeiten p bzw. $1-p$ erzeugen.

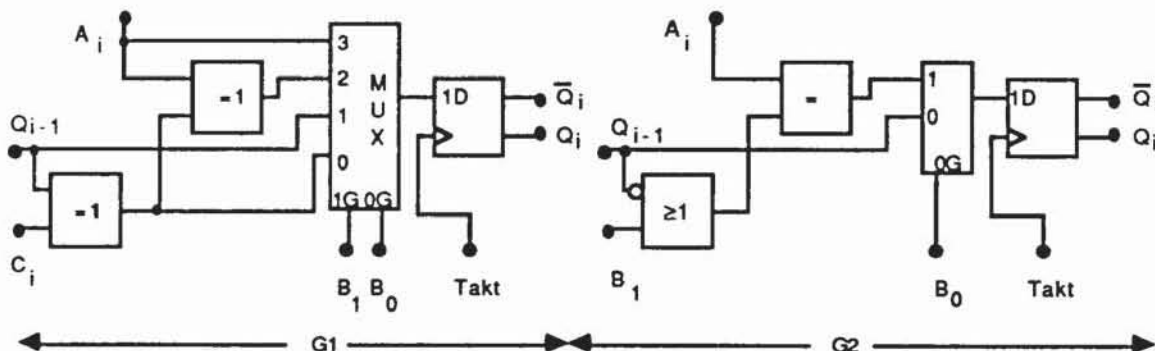


Bild 6: Grundzellen G1 und G2

Tabelle 1 beschreibt, wie der GURT über die beiden Steuersignale B_0 , B_1 in seine vier Betriebsarten geschaltet wird.

	Betriebsart	Steuersignale	
		B_1	B_0
1	Zufallsmustergenerator	0	1
2	Signaturanalyse	1	0
3	Schieberegister	0	0
4	Systemregister	1	1

Tabelle 1: GURT-Betriebsarten

Während der Mustererzeugung zerfällt der Modul GR in einen LRSR- und einen Schieberegisterteil, zur Signaturanalyse werden LR- und SR-Teil über die Multiplexer zu einem großen LRSR- und einem Schiebe-

registerteil zusammenschaltet. Dies reduziert die Gefahr der Fehlermaskierung, die exponentiell mit der Registerlänge abnimmt ([Leis81], [Wil87]). Die gesamte GURT-Architektur kann durch folgende Parameter beschrieben werden:

- Registerbreite
- Für jeden Ausgang des Registers eine Wahrscheinlichkeit p der Bitfolge für logisch 1 von $0 < p < 1$ im Raster $1/8$
- Rückkopplungspolynome des LRSR und des LR-Teils der Module GR
- Abgriffpunkte für die Funktion F
- Optionale Verwendung von statischen oder dynamischen Flipflops

3. Entwurf der Grundzellen

3.1. Systemkonzept

Als Schnittstelle eines GURT nach Bild 4 sind folgende Ein- /Ausgangssignale vorgesehen:

- Zwei Taktsignale, Phase A und Phase B;
- Zwei Steuerleitungen, B_1 und B_0 ;
- Versorgung, V_{DD} und GND;
- Schieberegistereingang und -ausgang, S_{IN} und S_{OUT} ;
- Ein- /Ausgänge der Register, $E(1..n)$ und $Q(1..n)$ bzw. $\bar{Q}(1..n)$;

Als Taktsignal soll ein nichtüberlappendes 2-Phasentakt verwendet werden. Die D-Flipflops werden aus zwei Latches in Master-Slave-Anordnung aufgebaut, wobei statische und dynamische Latches zur Verfügung stehen. Die unmittelbare Umsetzung der GURT-Architektur nach Bild 4 hat den Nachteil, daß der Aufwand, die Ausgänge des GURT mit den entsprechenden Eingängen des Schaltnetzes zu verdrahten, sehr hoch werden kann, wie Bild 7 zeigt.

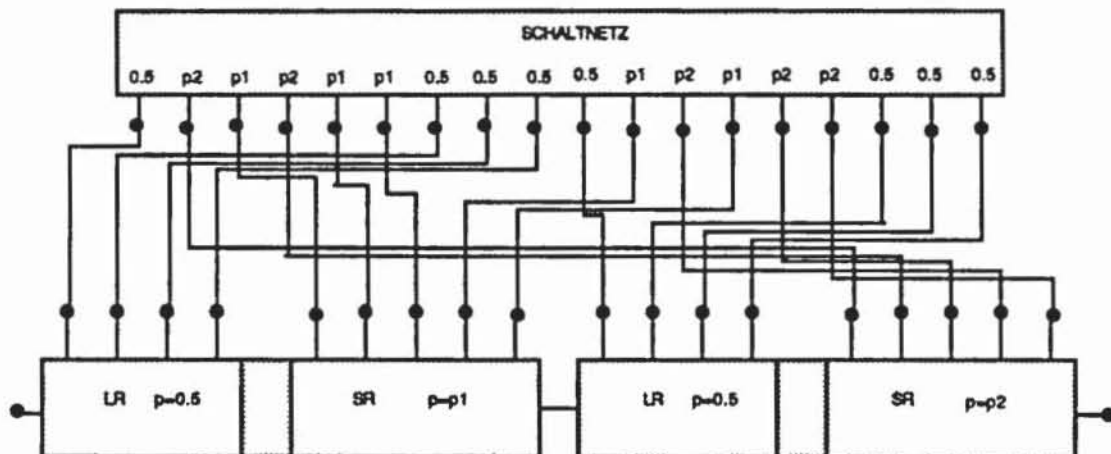


Bild 7: Verdrahtungsproblem

Daher wurde eine neue Architektur entwickelt, die es gestattet, die Grundzellen G_2 der einzelnen SR-Teile an beliebiger Stelle im Register einzufügen, also auch in den LR-Teilen und dem LRSR. Bild A1 (im Anhang) zeigt die neue GURT-Architektur, aus dem auch der Aufbau des Registers aus den Grundzellen ersichtlich ist. Bei diesem Aufbau zerfällt im Generatormodus der GURT in die einzelnen LR-Teile, SR-Teile sowie das LRSR. Jeder SR-Teil wird von einem eigenen LR-Teil über eine Funktion F gespeist. Im Signaturanalysemodus entsteht ein großes LRSR, und im Schieberegistermodus werden alle Zellen zu einem Schieberegister zusammengefaßt.

Durch äußere Verdrahtung und zusätzliche Grundzellen (CHANGE) werden die Schieberegisterketten der einzelnen SR- oder LR-Teile und des LRSRs gebildet. Für jeden SR-Teil und für die Verdrahtung des LRSRs und der LR-Teile wird jeweils ein Verdrahtungskanal benötigt, da diese Teile immer aufeinanderfolgend

angeordnet sind. Für einen GURT reichen also 4 parallele Verdrahtungskanäle aus, falls sämtliche Wahrscheinlichkeiten benötigt werden. Hinzu kommt noch die Verdrahtung der Funktionen F. Vorteilhaft ist, daß im Signaturanalysemodus alle Grundzellen zu einem LRSR zusammengefaßt werden und sich so die Fehlquote der Signaturanalyse im stationären Fall gegenüber der Architektur nach Bild 2 weiter verringert.

3.2. Logischer und physikalischer Entwurf der Grundzellen

Tabelle 2 stellt die für den Aufbau des GURTs benötigten Grundzellen zusammen. Die Funktionsbeschreibung der Zellen wurde im logischen Entwurf in CMOS-Transistorschaltungen mit minimaler Transistoranzahl umgesetzt. Zur Validierung des Systemkonzeptes wurde mit einem Logiksimulator eine Beispielschaltung des GURT auf Transistorebene simuliert.

Zelle	Funktion	Transistoren	Größe	
			Breite (μm)	Höhe (μm)
G1	Steuerung des Flipflopeingangs	20	187	137
G2		14	120	137
CHSIG	Steuerung der Konfiguration der LR- und SR-Teile	8	87	137
CHGEN		10	106	137
GEN		4	53	137
CHANGE		-	25	137
FUNC58	Boolesche Funktion F	12	106	137
FUNC68		10	94	137
FUNC78		12	112	137
FFSTAT1	Statische und dynamische Flipflops mit invertiertem und nichtinvertiertem Ausgang	16	122	137
FFSTAT2		16	122	137
FFDYN1		10	101	137
FFDYN2		10	101	137
INIT	Kopfzelle zur Erzeugung der Takt- und Steuersignale und für korrekten Abschluß des Blockes	20	243	137
TERM	Abschlußzelle für korrekten Abschluß des Blockes	-	10	137
DRCONT	Treiberzellen	12	150	137
DRCLK		16	187	137
THRU	Durchführungszelle für Metall1	-	87	137

Tabelle 2: Grundzellen GURT

Das Layout der Zellen beruht auf den Entwurfsregeln des SiliconCompilers GENESIL [GENE85] für einen CMOS-SiliconGate-Prozeß mit zwei Wannen und zwei Metallisierungsebenen. Zellen, die nach diesen Entwurfsregeln entworfen wurden, können auf sämtliche Prozesse, die GENESIL unterstützt, automatisch angepaßt werden. Den Größenangaben aus Tabelle 2 liegt eine Gatelänge von 4 μm zugrunde, die prozeßabhängig skaliert werden kann. Die internen Verbindungen der Grundzellen in dem durch die Kaskadierung entstehenden Block kommen durch Anlagern zustande. Im Layout der Grundzellen wurden daher die Steuer-, Takt- und Versorgungsleitungen in der zweiten Metallisierungsebene horizontal über jede Grundzelle gelegt, wobei diese Leitungen bei jeder Grundzelle auf gleicher Höhe liegen müssen, um eine Verbindung zu ermöglichen (Bild 8).

 Metall2
  Metall1
  Polysilizium

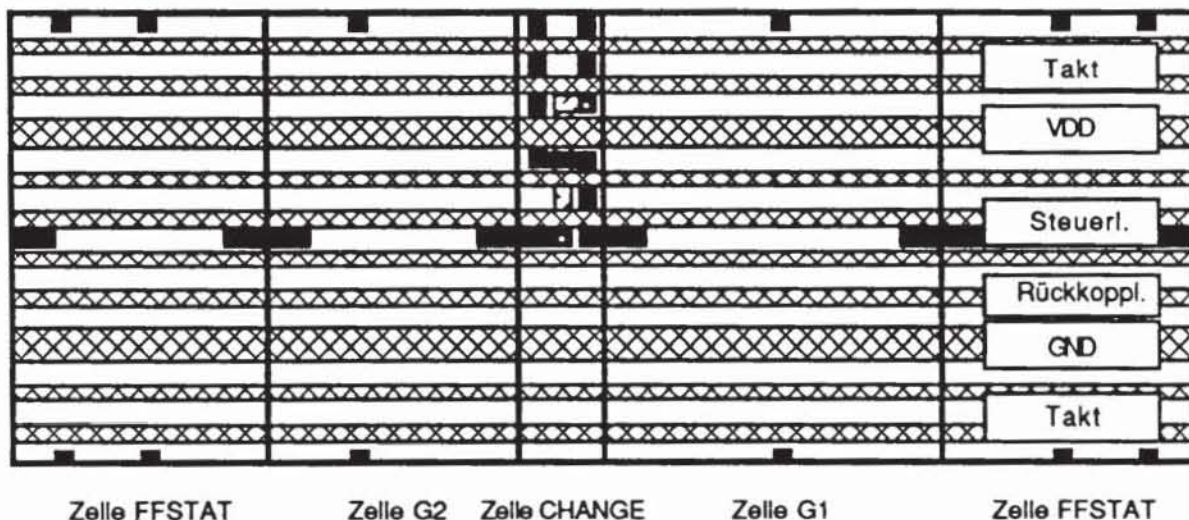


Bild 8: Layoutschema GURT

Für den Entwurf wurde der Graphikeditor des MENTOR-Systems verwendet [Chip86], anschließend wurde für jede Zelle eine Entwurfsregelüberprüfung sowie eine Extraktion der Transistoren und parasitären Kapazitäten durchgeführt, daran schloß sich eine Netzwerksimulation der extrahierten Schaltung und eine Überarbeitung des Layouts bei Schaltungsfehlern oder mangelhaftem Zeitverhalten an. Dabei wurde vorausgesetzt, daß das Register im Systembetrieb mit maximal 20 MHz Taktfrequenz betrieben werden soll. Das Layout der Zellen liegt im CIF- und im GDSII-Format vor.

4. Synthese des GURTs aus den Grundzellen

Die Erzeugung einer GURT-Standardzelle gliedert sich in folgende Teilaufgaben :

1. Bestimmung der Abfolge der Grundzellen
2. Externe Verdrahtung der Grundzellen
3. Zuordnung der globalen Netzliste zu den Anschlüssen der Grundzellen

ad 1) Zur Erzeugung von Mustern aller sieben möglichen Wahrscheinlichkeiten durch einen GURT sind ein maximales LRSR und maximal drei Moduln GR nötig. Letztere zerfallen wiederum in ein LR- und ein SR-Register (vgl. Bild 5), wobei Elemente aus SR die Wahrscheinlichkeit p oder $1-p$ repräsentieren. Es läßt sich daher ein GURT mit den in Tabelle 3 aufgeführten Zellen konfigurieren.

Zeichen	Bedeutung
a	LR1-Zelle
b	LR2-Zelle
c	LR3-Zelle
l	LRSR-Zelle
1	SR1-Zelle ($p = 1/8$)
2	SR2-Zelle ($p = 2/8$)
3	SR3-Zelle ($p = 3/8$)
5	SR3-Zelle ($p = 5/8$)
6	SR2-Zelle ($p = 6/8$)
7	SR1-Zelle ($p = 7/8$)

Zeichen	Bedeutung
t	Treiberzelle Takt
s	Treiberzelle Steuersignale
d	Durchführungszelle

Tabelle 3: Zeichen zur Konfiguration des GURT

Durch Eingabe einer Zeichenkette nach Tabelle 3 und durch Angabe der Rückkopplungspolynome für die LR-Teile und den LRSR-Teil kann der Benutzer den GURT vollständig spezifizieren. Für jeden LR-Teil und den LRSR-Teil wird vom Benutzer eine Zeichenkette eingegeben, die die Werte 0 (Rückkopplungskoeffizient = 0) und 1 (Rückkopplungskoeffizient = 1) als gültige Eingabewerte hat.

- ad 2) In diesem Schritt wird die zusätzliche äußere Verdrahtung der Grundzellen (vgl. Bild A1) durchgeführt. Für das Entwurfssystem GENESIL wurde vorgesehen, daß der Routing-Algorithmus des Systems dies durchführt, da die externe Verdrahtung nur aus den BUSOUT-Signalen der globalen Netzliste (Tabelle 4) besteht.
- ad 3) Hier kann der Benutzer für jedes Schnittstellensignal der Tabelle 4 einen Signalnamen der globalen Netzliste angeben.

BUSIN [1..n]	Ein/ Ausgangssignale
BUSOUT [1..n]	
PH A	Taktsignale
PH B	
CONT 0	Steuersignale
CONT 1	
SHIFTIN	Schieberegistereingang
SHIFTOUT	und -ausgang

Tabelle 4: Schnittstellensignale GURT

5. Ergebnisse

Aufgrund der entworfenen Grundzellen wurde die GURT-Standardzelle mit einem BILBO verglichen, das als parametrisierte Standardzelle unter den gleichen Randbedingungen entworfen wurde.

Für den Vergleich wurde die Benchmarkschaltung C880 [Brgl85] verwendet, welche 60 primäre Eingänge und 26 primäre Ausgänge aufweist. Eine Testbarkeitsanalyse [Wund87a] schätzt für den Test mit Zufallsmustern eine Testlänge von $3.7 \cdot 10^4$ gleichverteilten Zufallsmustern und $6.6 \cdot 10^2$ ungleichverteilten Zufallsmustern. Bild 9 zeigt die Gruppierung der Eingänge, die aus [Wund87a] entnommen wurde, und schematisch den Aufbau der GURT-Standardzelle. Mit Fehlersimulation wurde gezeigt, daß diese Konfiguration sehr geringe Testlängen benötigt, um zu einer vollständigen Fehlererfassung zu gelangen. Tabelle 5 vergleicht die Fehlererfassungen, die durch das BILBO und durch die Schaltung nach Bild 9 erzielt wurden.

Musterzahl	GURT	BILBO
293	100 %	94 %
6106	100 %	100 %

Tabelle 5: Fehlererfassung mit GURT und BILBO

Aufgrund von Simulationsergebnissen wurde das Zeitverhalten der Schaltungen verglichen. Im Registerbetrieb ergab sich eine nahezu identische Verzögerung durch die Zusatzlogik, im Testbetrieb begrenzt bei beiden Schaltungen die Anzahl der von Null verschiedenen Rückkopplungskoeffizienten die maximale Taktrate.

In Bild 9 repräsentieren die senkrechten Linien jeweils eine CHANGE-Zelle, die an dieser Stelle im Layout eingefügt werden muß. Die Bitfolgen für die Zelle FUNC78 werden dem LR1-Teil entnommen, die Bitfolgen für die Zelle FUNC68 werden dem LR2-Teil entnommen, hierfür sind drei Verdrahtungskanäle notwendig. Es ergeben sich somit insgesamt als zusätzliche äußere Verdrahtungsfläche 6 parallele Verdrahtungskanäle.

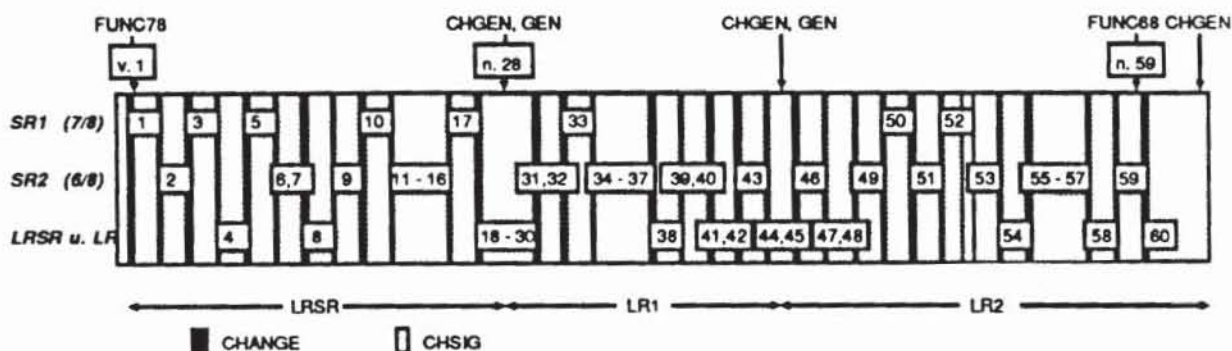


Bild 9: Aufbau GURT-Standardzelle für C880

In Tabelle 6 sind die Ergebnisse in einer Übersicht dargestellt. GURT2 ist eine Konfiguration, welche nur gleichverteilte Muster erzeugt, also wie ein BILBO arbeitet, GURT1 ist die Konfiguration aus Bild 9. Für den Vergleich ergeben sich als Verhältnisse der Gesamtflächen

$$\frac{\text{GURT1}}{\text{BILBO}} = 1.41 \quad \frac{\text{GURT2}}{\text{BILBO}} = 0.97$$

und für das Verhältnis der Transistoren

$$\frac{\text{GURT1}}{\text{BILBO}} = 0.97 \quad \frac{\text{GURT2}}{\text{BILBO}} = 0.99$$

	Transistoren Register	Transistoren Zusatzlogik	Fläche Register	Fläche Zusatzlogik
BILBO	960	1 044 (+108%)	1 104 900 μm^2	1 248 859 μm^2 (+113%)
GURT1	960	1 000 (+104%)	1 002 840 μm^2	1 335 613 μm^2 (+133%)
GURT2	960	1 040 (+108%)	1 002 840 μm^2	1 296 431 μm^2 (+129%)

	Fläche Verdrahtung	Zusatzfläche gesamt	Gesamtfläche
BILBO	-	1 248 859 μm^2 (+113%)	2 353 759 μm^2
GURT1	990 002 μm^2 (+98%)	2 325 615 μm^2 (+231%)	3 328 455 μm^2
GURT2	-	1 296 431 μm^2 (+129%)	2 299 271 μm^2

Tabelle 6: Vergleich GURT - BILBO

Die GURT-Standardzelle läßt sich also auch zur Erzeugung von gleichverteilten Mustern nach dem BILBO-Prinzip einsetzen, wobei sie leichte Vorteile gegenüber der BILBO-Standardzelle aufweist. Erzeugt man ungleichverteilte Zufallsmuster, so steigt der zusätzliche Flächenbedarf gegenüber einem BILBO zwar an, im Beispiel um 41%, die Testlänge N läßt sich aber verringern, im Beispiel auf 5%. Hierzu ist noch zu bemerken, daß zur Berechnung der zusätzlichen Verdrahtungsfläche angenommen wurde, daß nur zwei Metallisierungsebenen zur Verfügung stehen und die Zelle automatisch verdrahtet wird, beispielsweise von dem SiliconCompiler GENESIL. Durch einen Verdrahtungsalgorithmus, der auch die Polysiliziumebene nutzt, kann dieser Flächenbedarf reduziert werden. Dies wird Gegenstand weiterer Untersuchungen sein.

Literatur

- [ArVa88] Archambeau, E.; Van Egmond, K.; Built-In Test Compiler in an ASIC Environment; IEEE International Test Conference, 1988
- [Bard87] Bardell, P.H.; Built-In Test for VLSI: Pseudorandom Techniques; John Wiley & Sons Inc., 1987
- [BaSa84] Bardell, P.H.; Savir, J.; On Random Pattern Test Length; IEEE Trans. On Comp., Vol. C-33, No.6, Juni 1984
- [Brgl85] Brglez, F.; Pownall, P.; Hum, R.; Accelerated ATPG and Fault Grading via Testability Analysis; Proc. IEEE, International Symposium on Circuits and Systems, Kyoto, 1985
- [Chip86] ChipGraph User's Manual; Mentor Graphics, 1986
- [GENE85] GENESIL System; System Description Users Manual; Silicon Compilers Inc.; 1985
- [Golo82] Golomb, S.W.; Shift Register Sequences; Aegan Park Press, Laguna Hills, Cal., 1982
- [KMZ79] Koenemann, B.; Mucha, J.; Zwihehoff, G.; Built-In Logic Block Observation Techniques; IEEE Test Conference, Cherry Hill, New Jersey, 1979
- [Leis81] Leisengang, D.; Klassifikation und Einsatz von Signaturregistern zur Fehlererkennung in digitalen Schaltungen; Dissertation an der TU München, 1981
- [Stas83] Stassen, F.; A Data Register with Testfacilities; DTU, November 1983
- [Will87] Williams, T.W.; Daehn, W.; Gruetzner, M.; Starke, C.W.; Aliasing Errors with Primitive and Non-Primitive Polynomials; IEEE Test Conference, 1987
- [Wund85] Wunderlich, H.-J.; A Tool for Probabilistic Testability Analysis; Design Automation Conference, Las Vegas, 1985
- [Wund87a] Wunderlich, H.-J.; Probabilistische Verfahren für den Test hochintegrierter Schaltungen; Dissertation, Informatik Fachberichte 140, Springer, 1987
- [Wund87b] Wunderlich, H.-J.; Self Test Using Unequiprobable Random Patterns; International Symposium on Fault-Tolerant Computing, FTCS-17, Pittsburgh, 1987
- [Wu88] Wunderlich, H.-J.; Multiple Distributions for Biased Random Patterns; International Test Conference, Washington, 1988

Anhang

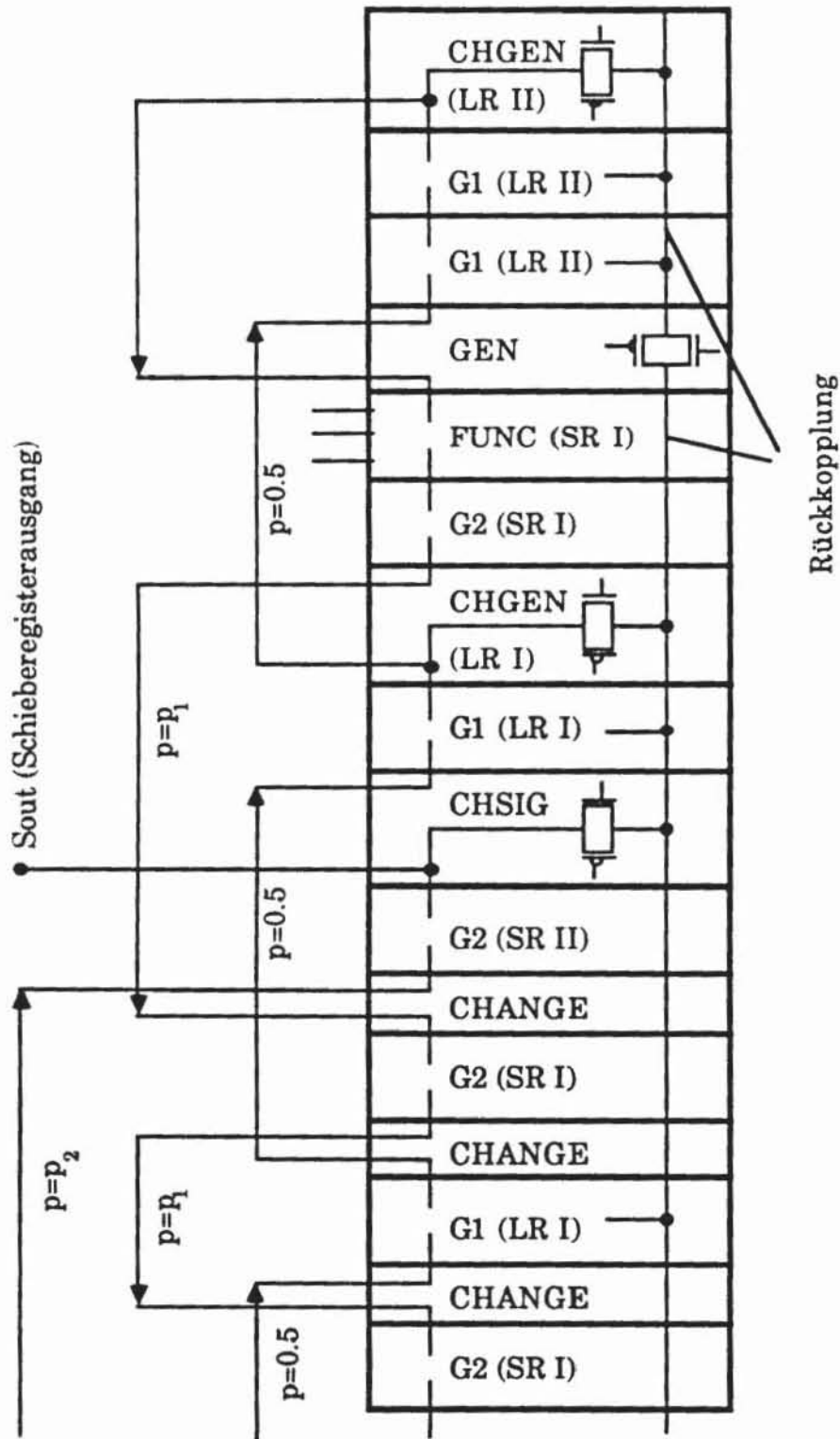


Bild A1: GURT-Architektur