



**Hochschule**  
**Augsburg** University of  
Applied Sciences

**Hardwaresysteme**  
**Kürprojekt**

**Fakultät für**  
**Informatik**

Studienrichtung  
M. Sc. Informatik

**Sascha Binkert**  
**Entwicklung eines ICs - Vorstellung von verschiedenen Tools**

Abgabe bei: Prof. Dr.-Ing. Gundolf Kiefer  
Abgabe der Arbeit am: 09.07.2023

Hochschule für angewandte  
Wissenschaften Augsburg  
University of Applied Sciences

An der Hochschule 1  
D-86161 Augsburg

Telefon +49 821 55 86-0  
Fax +49 821 55 86-3222  
[www.hs-augsburg.de](http://www.hs-augsburg.de)  
[info@hs-augsburg.de](mailto:info@hs-augsburg.de)

**Fakultät für Informatik**  
Telefon +49 821 5586-3450  
Fax +49 821 5586-3499

Verfasser der Ausarbeitung:  
Sascha Binkert  
Matr. Nr.: 2146627  
M. Sc. Informatik  
Teamkollegen:  
Mathias Schoppe und  
Timo Winklbauer

# Inhaltsverzeichnis

|          |                                 |           |
|----------|---------------------------------|-----------|
| <b>1</b> | <b>Einleitung</b>               | <b>1</b>  |
| <b>2</b> | <b>GDS II Format</b>            | <b>1</b>  |
| 2.1      | Entwicklung . . . . .           | 1         |
| 2.2      | Verwendung . . . . .            | 2         |
| 2.3      | Alternativen . . . . .          | 3         |
| 2.3.1    | OASIS . . . . .                 | 3         |
| <b>3</b> | <b>Wokwi</b>                    | <b>3</b>  |
| 3.1      | Beschreibung . . . . .          | 4         |
| 3.1.1    | Einzigartige Features . . . . . | 4         |
| 3.2      | Unterstützte Hardware . . . . . | 5         |
| <b>4</b> | <b>Open-Lane</b>                | <b>6</b>  |
| 4.1      | Architektur . . . . .           | 7         |
| <b>5</b> | <b>Prüfpfad</b>                 | <b>8</b>  |
| 5.1      | Hintergrund . . . . .           | 8         |
| 5.2      | Fehlermodelle . . . . .         | 9         |
| <b>6</b> | <b>Zusammenfassung</b>          | <b>11</b> |

## 1 Einleitung

In diesem Kürprojekt soll ein eigener integrierter Schaltkreis (IC) erstellt werden, welcher in einen Zustand gebracht werden soll, um ihn mit Hilfe des Tiny-Tapeout Projekts auf einen Chip zu bringen. Dabei wurden verschiedene Aspekte des Projekts in unterschiedlichen Ausarbeitungen beleuchtet. Während Timo Winklbauer sich näher mit dem Tiny-Tapeout Projekt und dem Herstellungsverfahren beschäftigt hat [13], handelt die Ausarbeitung von Mathias Schoppe über den von dieser Projektgruppe entwickelten IC [9]. In dieser Arbeit soll nun die von Tiny-Tapeout verwendeten Tools und Datei-Formate näher beleuchtet werden.

## 2 GDS II Format

Seit über 37 Jahren hat sich GDSII als branchenübliche Datenbank für IC-Layouts etabliert. Obwohl es alternative Formate gibt, die als potenzielle Ersatzkandidaten diskutiert wurden, ist GDSII nach wie vor die vorherrschende Methode zur präzisen Beschreibung des physischen Layouts für Masken, die bei der Herstellung von Chips verwendet werden. Die langjährige Akzeptanz und Anwendung von GDSII in der Industrie spiegeln die Zuverlässigkeit und Effektivität dieses Formats wider. [2] In diesem Kapitel werden wir uns mit der Entwicklungsgeschichte, der Verwendung und einigen Alternativen von GDSII befassen, um ein umfassendes Verständnis für seine Bedeutung und seine Rolle in der heutigen IC-Designlandschaft zu erlangen.

### 2.1 Entwicklung

GDS II, auch bekannt als Calma GDS II-Stream-Format, wurde in den 80er Jahren von Calma entwickelt, und im Laufe der Jahre wechselte die Eigentümerschaft der Spezifikation von Calma zu GE, dann zu Valid und schließlich zu Cadence. Dabei ist zu erwähnen, dass GDSII kein offener Industriestandard ist. [2]

Das GDSII- oder Stream-Dateiformat speichert seine Informationen in Datensätzen. Jeder Datensatz beginnt mit zwei Bytes, die die Länge des Datensatzes enthalten. Die maximale Größe eines Datensatzes ist also auf  $2^{16}$  Bytes begrenzt. Nach dem Längen-Datensatz folgen ein Datensatztyp und ein Datentyp-Byte. Die GDSII-Spezifikation umfasst die Beschreibung von 69 verschiedenen Datensatztypen, von denen jedoch weniger als die Hälfte üblicherweise verwendet werden. Zu den häufig verwendeten Typen gehören LAYER, BOUNDARY (Polygon), PATH, SREF (Zellreferenz) oder XY (um Koordinaten zu speichern). Sieben

verschiedene Stream-Datentypen werden definiert, und jeder Datensatztyp muss den angegebenen Datentyp haben. Die unterstützten Datentypen sind: kein Datenfeld, Bit-Array, zweibyte-ganzzahlige Zahl, vierbyte-ganzzahlige Zahl, vierbyte-gleitkommazahl, achtbyte-gleitkommazahl und ASCII-Zeichenkette. Die Rohdaten eines GDSII-Formats sind binär, jedoch kann der integrierte Texteditor des LayoutEditors sie in lesbaren Text konvertieren und umgekehrt. Dadurch kann der integrierte Texteditor für die Low-Level-Manipulation von GDSII-Dateien verwendet werden. [4], [5], [8]

## 2.2 Verwendung

GDSII ist das am häufigsten verwendete Format zur Beschreibung des physischen Layouts von integrierten Schaltkreisen (ICs) und spielt eine entscheidende Rolle im gesamten Produktionsprozess von Halbleitern. Eine der Hauptanwendungen von GDSII liegt in der Maskenherstellung. Dabei verwenden Layout-Designer GDSII-Dateien, um das physische Layout eines ICs zu beschreiben, einschließlich der Platzierung und Verbindung von Transistoren, Leiterbahnen, Kontakten und anderen Strukturen. Diese Layout-Informationen werden dann zur Erzeugung der Masken verwendet, die in der Fotolithographie zur Übertragung des Layouts auf den Halbleiterwafer verwendet werden.

Dabei bietet GDSII eine Vielzahl von Vorteilen. Durch die Verwendung eines einheitlichen Formats ermöglicht es einen reibungslosen Datenaustausch zwischen verschiedenen CAD-Systemen und Fertigungsstätten. Dadurch können verschiedene Akteure in der Halbleiterindustrie, wie Layout-Designer, Halbleiterhersteller und Chipentwickler, effizient zusammenarbeiten und Informationen nahtlos teilen. Ein weiterer Vorteil von GDSII liegt in seiner Effizienz bei der Speicherung großer Layout-Daten. Das binäre Format von GDSII ermöglicht eine kompakte Darstellung des Layouts, wodurch Speicherplatz gespart wird. Dies ist besonders wichtig, da moderne ICs eine immer höhere Komplexität aufweisen und enorme Mengen an Layout-Daten erzeugen. Die Unterstützung von GDSII durch eine Vielzahl von CAD-Tools und Fertigungsanlagen ist ein weiterer Faktor, der zu seiner weit verbreiteten Verwendung beiträgt. Viele Softwarepakete für das IC-Design bieten Funktionen zur Generierung, Bearbeitung und Überprüfung von GDSII-Dateien. Darüber hinaus unterstützen die meisten Fertigungsanlagen den Import des Formats, um die Produktion der erforderlichen Masken durchzuführen.

Trotz seiner langjährigen Dominanz gibt es auch einige Herausforderungen bei der Verwendung von GDSII. Da es sich nicht um einen offenen Industriestandard handelt, liegt die Weiterentwicklung und Aktualisierung des Formats in den Händen weniger Unternehmen. Dies kann zu Kompatibilitätsproblemen führen,

insbesondere wenn verschiedene Versionen von GDSII verwendet werden. Daher ist es wichtig, sicherzustellen, dass alle beteiligten Parteien die gleiche Version von GDSII unterstützen.

Insgesamt spielt GDSII eine entscheidende Rolle in der Halbleiterindustrie und ist unverzichtbar für das Layout-Design und die Maskenherstellung von ICs. Es bietet eine effiziente und standardisierte Methode zur Beschreibung und Übertragung von Layout-Informationen, wodurch die Zusammenarbeit zwischen verschiedenen Akteuren erleichtert und die Produktionsprozesse optimiert werden.

## 2.3 Alternativen

Durch das langjährige bestehen von GDSII ist es folgerichtig das verschiedene Alternativen für dieses Format entwickelt wurden. Nachfolgend wird die bekannteste OASIS näher vorgestellt.

### 2.3.1 OASIS

OASIS (Open Artwork System Interchange Standard) ist eine Sprache zur Darstellung von elektronischen Mustern für integrierte Schaltkreise während ihres Designs und ihrer Herstellung [7]. Es bietet eine effiziente Datenrepräsentation und ermöglicht die Organisation von geometrischen Formen in Zellen, die wiederum andere Zellen und Formen enthalten können. OASIS wurde entwickelt, um die Grenzen des GDSII-Formats zu überwinden und wird branchenweit als Alternative für komplexe Maskenlayouts verwendet.

Dank seiner kompakten Größeninteger-Repräsentation, effizienten Datenkompression und vielfältigen Optimierungstechniken ermöglicht OASIS kleinere Dateigrößen im Vergleich zu GDSII [6]. Die Verwendung von OASIS erleichtert den Austausch von Layoutdaten zwischen verschiedenen EDA-Softwaretools, IC-Maskenschreibgeräten und Maskeninspektionssystemen. Durch die weite Verbreitung und Akzeptanz in der Mikroelektronikindustrie trägt OASIS maßgeblich zur Verbesserung der Effizienz und Genauigkeit bei der Entwicklung und Herstellung von integrierten Schaltkreisen bei [12].

## 3 Wokwi

Wokwi ist ein online Elektronik-Simulator, welcher in Verbindung mit dem Tiny-Tapeout Projekt erstellt wurde. Während in der Arbeit von Timo Winklbauer in

Kapitel 3 das Projekt näher vorgestellt wird [13], soll hier der Simulator Wokwi näher beschrieben werden.

### 3.1 Beschreibung

Woki ist ein frei und kostenlos verfügbarer Simulator, welcher es ermöglicht verschiedene elektronische Schaltungen auf Bauteil- oder Gatterebene zu simulieren. Er bietet dafür eine benutzerfreundliche Umgebung, um IoT-Projekte zu entwickeln und Programmierkenntnisse im Bereich der Mikrocontroller-Programmierung zu erweitern. Dabei ermöglicht Wokwi einen einfachen Einstieg und erfordert keine Wartezeit auf nicht vorhandene Komponenten oder das Herunterladen umfangreicher Software. Da der Simulator komplett Browser basiert ist können Benutzer innerhalb von Sekunden mit der Erstellung ihres nächsten IoT-Projekts beginnen. Ein besonderer Vorteil des Wokwi-Simulators liegt darin, dass Fehler und Experimente keine negativen Auswirkungen auf die virtuelle Hardware haben. Es besteht keine Gefahr, wertvolle Komponenten zu beschädigen, da der Simulator robuste Schutzmechanismen bietet. Im Gegensatz zur realen Hardware besteht zudem die Möglichkeit, Änderungen jederzeit rückgängig zu machen und den Entwicklungsprozess zu optimieren.

Eine weitere Stärke von Wokwi liegt in der einfachen Möglichkeit, Unterstützung und Feedback zu erhalten. Durch das Teilen eines Links zu einem Wokwi-Projekt können Nutzerinnen und Nutzer problemlos Hilfe von anderen Mitgliedern der Community erhalten. Dabei ist insbesondere die rege und hilfsbereite Discord-Community hervorzuheben.

Des Weiteren ermöglicht der Simulator eine klare Trennung von Hardware- und Softwareproblemen. Durch diese Unterscheidung können Entwicklerinnen und Entwickler sich gezielt auf die Verbesserung ihres Codes konzentrieren und ihre Programmierfähigkeiten weiterentwickeln.

Ein weiterer Vorzug des Wokwi-Simulators besteht in der uneingeschränkten Verfügbarkeit von Hardware-Ressourcen. Es ist nicht notwendig, Bauteile aus vergangenen Projekten zu verwenden oder Kosten und Verfügbarkeit zu beachten. Benutzer können so viele Komponenten verwenden, wie sie für ihr Projekt benötigen, ohne Einschränkungen durch Preis oder Bestand [1].

#### 3.1.1 Einzigartige Features

Der Simulator enthält einige einzigartige Fähigkeiten welche ihn von allen anderen Hardware-Simulatoren hervorhebt. Da diese jedoch nicht innerhalb des Projektes zu tragen kamen, werden einige hier nur kurz erwähnt.

- **WiFi Simulation:**

Wokwi ermöglicht die Simulation eines WiFi-Netzwerks mit uneingeschränktem Internetzugang. Dadurch kann man einen ESP32 in Verbindung mit dem virtuellen WiFi nutzen, um IoT-Projekte zu prototypisieren. Damit ist es möglich den ESP32 mit MQTT-Servern verbinden, um Sensordaten zu senden. Dies ermöglicht eine nahtlose Integration in bestehende IoT-Infrastrukturen. Des Weiteren kann man über HTTP, HTTPS und Websockets Webdienste abfragen und so eine direkte Kommunikation mit externen Systemen herstellen. Zudem ist es möglich, einen HTTP-Server innerhalb des ESP32 zu betreiben und eine Verbindung dazu vom Browser aus herzustellen. Diese Funktion erfordert jedoch die Verwendung des Wokwi IoT Gateway [1].

- **Virtueller Logikanalysator:**

Der Logikanalysator ermöglicht die Aufzeichnung der Werte digitaler Signale eines Projektes. Dieses leistungsstarke Debugging-Tool hilft bei der Diagnose von Problemen in Schaltungen und Code. Darüber hinaus eignet es sich hervorragend als Lernhilfe. Dabei bietet er eine Vielzahl von Anwendungsmöglichkeiten. Es ist möglich das PWM-Signal untersuchen, das von der analogWrite() Funktion erzeugt wird oder das Steuersignal für Servos analysieren. Darüber hinaus ermöglicht der Logikanalysator die Entwicklung und Fehlerbehebung von PIO-Maschinen für den Raspberry Pi Pico. Des Weiteren hilft der Logikanalysators das I2C-Protokoll genauer verstehen, indem die SCL/SDA-Signale betrachtet werden [1].

## 3.2 Unterstützte Hardware

Wokwi unterstützt eine zahlreiche Menge an unterschiedlicher Hardware zur Simulation. Ein Ausschnitt davon soll nun hier kurz genannt werden:

- **Arduino-Boards:**

Es wird eine breite Palette von Arduino-Boards unterstützt, darunter Arduino Uno, Arduino Mega, Arduino Nano und viele mehr. Die Funktionen und Eigenschaften dieser Boards können in der Simulation vollständig genutzt werden [1].

- **ESP32-Boards:**

ESP32 ist ein leistungsstarker Mikrocontroller, der für IoT-Anwendungen beliebt ist. Mit Wokwi ist es möglich ESP32-Boards wie NodeMCU, Wemos D1 Mini und andere verwenden, um WiFi-Verbindungen herzustellen, Sensordaten zu erfassen und komplexe IoT-Projekte zu simulieren.

- **STM32-Boards:**

STM32-Mikrocontroller bieten eine hohe Leistung und Vielseitigkeit. Wokwi unterstützt STM32-Boards wie STM32F103C8T6 und STM32F407VGT6, mit denen anspruchsvolle Projekte entwickelt und simuliert werden können.

- **Raspberry Pi Pico:**

Der Raspberry Pi Pico ist ein beliebtes und kostengünstiges Mikrocontroller-Board.

- **ESP8266-Boards:**

Das ESP8266 ist ein WiFi-Modul, das für seine einfache Verbindung mit dem Internet bekannt ist. Wokwi unterstützt verschiedene ESP8266-Boards, mit denen WiFi-Verbindungen herstellt und IoT-Projekte simuliert werden können.

- **Sensoren und Module:**

Neben den Mikrocontroller-Boards unterstützt Wokwi auch eine Vielzahl von Sensoren und Modulen. Darunter sind LEDs, Taster, LCD-Displays, Servos, Motoren, WiFi-Module und viele weitere Komponenten, welche frei in den Simulationen verwendet werden können.

## 4 Open-Lane

OpenLane ist ein automatisierter RTL-zu-GDSII-Workflow. Dabei basiert es auf einer Reihe von Komponenten, darunter OpenROAD, Yosys, Magic, Netgen, CVC, SPEF-Extractor, KLayout sowie mehreren benutzerdefinierten Skripten für die Design-Exploration und die Optimierung. Somit ermöglicht es diese Software-Suite, alle Schritte der ASIC-Implementierung vom RTL bis hin zum GDSII durchzuführen.

Derzeit unterstützt OpenLane sowohl die A- als auch die B-Varianten des sky130 PDKs sowie die C-Variante des gf180mcu PDKs. Ein Hauptmerkmal von OpenLane besteht darin, dass es die zugrunde liegenden Open-Source-Tools abstrahiert und den Benutzern ermöglicht, das Verhalten der Tools mithilfe einer einzigen Konfigurationsdatei anzupassen. Dies ermöglicht eine flexible und anpassbare Arbeitsumgebung, in der Benutzer die gewünschten Einstellungen leicht konfigurieren können [3].



## 4.1 Architektur

OpenLane ist ein umfassender Workflow, welcher aus mehreren Stufen besteht. Standardmäßig werden alle Schritte des Workflows nacheinander ausgeführt. Jede Stufe kann aus mehreren Unterstufen bestehen. Der vollständige Ablauf wird in Abbildung 1 dargestellt.

In der ersten Stufe erfolgt die Synthese, bei der das RTL-Design mithilfe von „yosys/abc“ in ein technologisches Mapping überführt wird. Anschließend erfolgt die statische Timing-Analyse der resultierenden Netlists mithilfe von „OpenSTA“, um Timing-Berichte zu generieren.

Die nächste Stufe ist die Floorplanung, bei der die Kernfläche für die Makro definiert wird, sowie die Reihen (für die Platzierung) und Tracks (für das Routing). Mit Hilfe von „ioplacer“ werden die Ein- und Ausgangs-Ports der Makro platziert. „pdngen“ generiert das Verteilungsnetzwerk für die Stromversorgung und „tapcell“ fügt Welltap- und Decap-Zellen in den Floorplan ein.

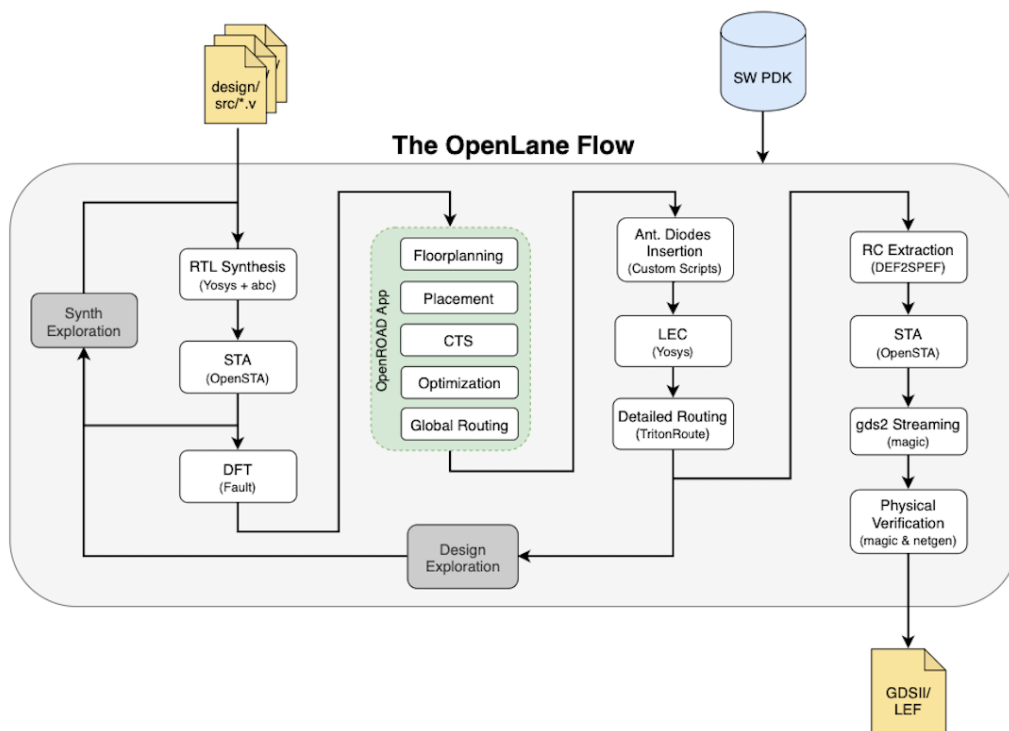


Abbildung 1: Graphische Darstellung des Gesamten OpenLane-Workflows, entnommen von [3].

Die darauf folgende Platzierung umfasst mehrere Schritte. „RePLace“ führt die

globale Platzierung durch, gefolgt von optionalen Optimierungen mit „Resizer“ und der detaillierten Platzierung zur Legalisierung der global platzierten Komponenten mit „OpenDP“.

Darauffolgend wird die CTS (Clock Tree Synthesis), bei der das Taktverteilungsnetzwerk (der Clock Tree) mit „TritonCTS“ synthetisiert wird, durchgeführt.

Die Routing-Stufe umfasst die globale Routenplanung mit „FastRoute“, um eine Leitfaden-Datei für den detaillierten Router zu generieren, gefolgt von der detaillierten Routenplanung mit „TritonRoute“. Anschließend wird mit „OpenRCX“ die SPEF-Extraktion durchgeführt.

Die Tapeout-Stufe beinhaltet das Generieren der finalen GDSII-Layoutdatei aus dem gerouteten DEF mit „Magic“ und optional als Backup mit „KLayout“.

In der Signoff-Stufe werden verschiedene Überprüfungen durchgeführt. „Magic“ und „KLayout“ führen DRC-Checks (Design Rule Checks) durch, „Magic“ führt zusätzlich noch Antenna-Checks durch, „Netgen“ führt LVS-Checks (Layout vs. Schematic) durch und „CVC“ führt Überprüfungen zur Schaltungsgültigkeit durch [3].

## 5 Prüfpfad

### 5.1 Hintergrund

Mit zunehmender Komplexität von Logikbausteinen wurde es immer aufwändiger, Tests manuell zu erstellen und zu validieren. Es war schwierig festzustellen, wie umfassend die Tests waren, und sie dauerten zu lange, um ausgeführt zu werden. Daher entschied sich die Industrie für einen Design-for-Test-Ansatz (DFT), bei dem das Design modifiziert wurde, um den Testvorgang zu vereinfachen. Der Ansatz, der sich im IC-Test durchgesetzt hat, nennt sich struktureller oder SScan-Test, weil er das Scannen von Testmustern in interne Schaltungen des zu testenden Bausteins (DUT) umfasst. Die Flip-Flops des Designs werden modifiziert, um sie während des Tests als Anregungs- und Beobachtungspunkte oder SScan-Zellen zu nutzen, während sie während des normalen Betriebs ihre beabsichtigte funktionale Rolle erfüllen.

Die modifizierten Flip-Flops oder Scan-Zellen ermöglichen es, das gesamte Design als viele kleine Segmente kombinatorischer Logik zu betrachten, die leichter getestet werden können. Bei einem Design mit einer Million Flip-Flops ist es durch die Einführung von Scan-Zellen, als ob man eine Million Steuerungs- und Beobachtungspunkte hinzufügen würde. Durch diese Segmentierung wird es möglich, automatisch generierte Testmuster zu erzeugen, die die Logik zwischen den Flip-Flops anregen können. Die Testsoftware muss die Funktion der Logik

nicht verstehen ? sie versucht lediglich, die von einer Scan-Zelle beobachteten Logiksegmente anzuregen. Da der Scan-Test die Flip-Flops modifiziert, die bereits im Design vorhanden sind, um sie auch als Scan-Zellen zu nutzen, ist der Einfluss der Testschaltung relativ gering und trägt in der Regel nur etwa 1-5 % zur Gesamt-Gatterzahl bei.

Die Scan-Zellen werden zu „Scan-Ketten“ miteinander verbunden, die im Testmodus wie große Schieberegister funktionieren. Die Scan-Ketten werden von externen automatischen Testgeräten (ATE) genutzt, um Testmusterdaten aus dem Speicher in das Bauteil zu übertragen. Nachdem das Testmuster geladen wurde, wird das Design wieder in den funktionalen Modus versetzt, und die Testantwort wird in einem oder mehreren Taktzyklen erfasst. Das Design wird erneut in den Testmodus versetzt, und die erfasste Testantwort wird ausgelesen, während das nächste Testmuster gleichzeitig in die Scan-Zellen eingeschoben wird. Das ATE vergleicht dann die erfasste Testantwort mit den in seinem Speicher gespeicherten erwarteten Antwortdaten. Abweichungen werden als potenzielle Defekte betrachtet und für weitere Untersuchungen protokolliert.

Um automatische Testmustererzeugungssoftware (ATPG) in die Lage zu versetzen, die Testmuster zu erstellen, werden Fehlermodelle definiert, die das erwartete Verhalten des ICs vorhersagen, wenn Defekte vorhanden sind. Das ATPG-Tool verwendet dann diese Fehlermodelle, um die Muster zu bestimmen, die erforderlich sind, um diese Defekte an allen Punkten in der Schaltung zu erkennen. Dabei erreicht das ATPG-Tool üblicherweise eine Abdeckung von 95 % oder mehr [10].

## 5.2 Fehlermodelle

Es gibt zahlreiche verschiedene Fehlermodelle, welches ein ATPG-Tool anwenden kann. Im folgenden sollen einige davon kurz vorgestellt werden:

- **Stuck-at-Test:**

Das grundlegendste und häufigste Fehlermodell ist das „Stuck-at“-Fehlermodell, das jeden Knoten im Design auf ein festgelegtes Verhalten (stuck-at-1 oder stuck-at-0) überprüft. Wenn beispielsweise ein NAND-Gatter im Design einen Eingangspin auf Masse (Logikwert 0) durch einen Defekt kurzgeschlossen hat, würde der Stuck-at-0-Test für diesen Knoten diesen Defekt erkennen. Das Stuck-at-Modell kann auch andere Defekttypen wie Brücken zwischen zwei Netzen oder Knoten erkennen. Das Stuck-at-Modell wird als statisches Modell bezeichnet, da es ein langsamer Test ist und nicht von der Gatterzeit (Anstiegs- und Abfallzeiten und Verzögerungen) abhängt [10].

- **At-Speed-Test:**

Ein weiteres häufig verwendetes Fehlermodell ist das „Transition“- oder

„At-Speed“-Fehlermodell, das ein dynamisches Fehlermodell ist und Probleme mit der Zeitsteuerung erkennt. Es ähnelt dem Stuck-at-Modell, da es für jeden Knoten im Design zwei Fehler gibt, die als slow-to-rise (langsam aufsteigend) und slow-to-fall (langsam abfallend) Fehler klassifiziert sind. Das Transition-Fehlermodell verwendet ein Testmuster, das einen Übergangsreiz erzeugt, um den Logikwert entweder von 0 auf 1 oder von 1 auf 0 zu ändern. Die für den Übergang zur Verfügung stehende Zeit ist festgelegt, sodass bei einem nicht erfolgten Übergang oder einem Übergang außerhalb der festgelegten Zeit ein Timing-Defekt vermutet wird [10].

- **Path Delay-Test:**

Das „Path Delay“-Modell ist ebenfalls dynamisch und führt At-Speed-Tests auf gezielten zeitkritischen Pfaden durch. Während Stuck-at- und Transition-Fehlermodelle normalerweise alle Knoten im Design abdecken, testet das Path-Delay-Modell nur die spezifischen Pfade, die vom Ingenieur festgelegt werden. Der Ingenieur führt statische Timing-Analysen durch, um die kritischsten Pfade zu bestimmen. Diese Pfade werden dem ATPG-Tool zur Erstellung der Path-Delay-Testmuster angegeben. Die Theorie besagt, dass, wenn die kritischsten Timing-Pfade die Tests bestehen, alle anderen Pfade mit längeren Slack-Zeiten keine Timing-Probleme haben sollten. Path-Delay-Tests sind in gewisser Weise eine Form der Prozesskontrolle (z.B. zeigen sie Timing-Fehler, wenn eine Prozessvariable zu weit abweicht) und dienen neben der Prüfung auf Herstellungsdefekte einzelner Bauelemente [10].

- **IDDQ-Test:**

Der „IDDQ-Test“ basiert auf der Messung des Ruhestroms ( $I_{dd}$ ) im quieszenten Zustand (wenn die Schaltung nicht schaltet und die Eingänge statische Werte haben). Testmuster werden verwendet, um das DUT in verschiedenen ausgewählten Zuständen zu platzieren. Durch die Durchführung von Strommessungen in jedem dieser statischen Zustände können Defekte erkannt werden, die einen übermäßigen Stromverbrauch verursachen. Der Vorteil des IDDQ-Tests besteht darin, dass viele Arten von Fehlern mit sehr wenigen Mustern erkannt werden können. Der Nachteil besteht in der zusätzlichen Testzeit, die für die Durchführung der Strommessungen erforderlich ist [10].

- **Toggle-Fault-Test:**

Der „Toggle-Fault-Test“ stellt sicher, dass ein Knoten sowohl auf logischen Wert 0 als auch auf logischen Wert 1 geschaltet werden kann, und gibt Aufschluss über den Einfluss auf die Schaltungsknoten. Da das Toggle-Fehlermodell schneller ist und weniger Overhead erfordert als der Stuck-at-

Fehlertest, kann man verschiedene Schaltungskonfigurationen ausprobieren und schnell feststellen, wie viel Kontrolle man über die Schaltungsknoten hat. Da das Toggle-Fehlermodell nur Fehlerstellen anregt und die Antworten nicht zu Erfassungspunkten weiterleitet, kann es nicht zur Defekterkennung verwendet werden. Dieses Fehlermodell wird manchmal für das Einbrenntesten verwendet, um eine hohe Aktivität in der Schaltung zu erzeugen [10].

- **N-Detect und Embedded Multiple Detect (EMD):**

Die Grundidee des „N-Detect“ (oder Multi-Detect) besteht darin, jeden Fehler zufällig mehrmals anzugreifen. Die Art und Weise, wie der Fehler angegriffen wird, wird zufällig geändert, ebenso wie die Füllbits (Bits, die für den Fehler nicht relevant sind) im Mustersatz. Dieser Ansatz beginnt mit einem Standard-Stuck-at- oder Transition-Mustersatz, der potenzielle Defekte im Design anspricht. Der Mustersatz wird analysiert, um festzustellen, welche potenziellen Defekte durch mehr als ein Muster im Gesamtmustersatz erfasst werden. Anschließend werden zusätzliche unterschiedliche Muster generiert, um speziell auf Defekte abzielen, die weniger als die vom Benutzer festgelegte Mindestschwelle erfasst werden. Die kombinierten Informationen für alle resultierenden Muster erhöhen das Potenzial zur Erkennung eines Brückenfehlers, der sonst unentdeckt bleiben könnte [10].

## 6 Zusammenfassung

Insgesamt bieten GDSII, OpenLane und der Scan-Test wichtige Werkzeuge und Methoden für das Design und die Herstellung von ICs. Sie tragen zur Effizienz, Zuverlässigkeit und Qualität der Halbleiterindustrie bei und ermöglichen es Entwicklern, komplexe Schaltungen zu entwerfen, zu simulieren und zu testen. Durch den Einsatz dieser Technologien kann die Entwicklung und Produktion von ICs optimiert und verbessert werden, um den Anforderungen der heutigen digitalen Welt gerecht zu werden. Darauf aufbauend ist Wokwi ein hervorragendes Werkzeug, um einen einfachen Einstieg in die Welt der elektronischen Schaltungen zu ermöglichen.

Hierbei ist Tiny-Tapeout ein großartiges Projekt, welches genau die in dieser Arbeit vorgestellten Werkzeuge und Methoden in ihren Github Actions [11] verwendet, um die zahlreichen Designs welche in verschiedenen Repositorys [14] eingereicht werden, zu einem Chip zusammenzufassen.

## Literatur

- [1] CodeMagic LTD. *Wokwi Documentation*. 2021. URL: <https://docs.wokwi.com/>.
- [2] Steve DiBartolomeo. *GDSII File Format Specification*. Online. Applications Manager, Artwork Conversion Software, Inc. 2011. URL: <https://www.artwork.com/gdsii/gdsii/>.
- [3] Tim Edwards u. a. *OpenLANE Documentation*. Efabless Corporation und contributors, 2020. URL: <https://openlane.readthedocs.io/en/latest/>.
- [4] juspertor GmbH. *GDSII File Format Specification*. Online. LayoutEditor - Download, Support, Services. 2023. URL: <https://www.layouteditor.org/layout/file-formats/gdsii>.
- [5] R. Minixhofer. „Integrating Technology Simulation into the Semiconductor Manufacturing Environment“. In: *Proceedings of the International Conference on Semiconductor Manufacturing (2023)*. URL: <https://www.iue.tuwien.ac.at/phd/minixhofer/node52.html>.
- [6] OASIS. <https://www.layouteditor.org/layout/file-formats/oasis>. juspertor GmbH, 2023.
- [7] OASIS (*Open Artwork System Interchange Standard*). <https://en-academic.com/dic.nsf/enwiki/329427>. Wikimedia Foundation, 2010.
- [8] S. Rubin. *Computer Aids for VLSI Design, appendix C: GDSII Format*. Reading, Massachusetts: Addison-Wesley, 1987. URL: <http://www.rulabinsky.com/cavd/text/chapc.html>.
- [9] Mathias Schoppe. *Entwicklung eines ICs - Der Schaltkreisentwurf*. 2023.
- [10] Ed Sperling. *Scan Test*. Date unknown. URL: [https://semiengineering.com/knowledge\\_centers/test/scan-test-2/%22,%20organization%20=%20%22SemiEngineering](https://semiengineering.com/knowledge_centers/test/scan-test-2/%22,%20organization%20=%20%22SemiEngineering).
- [11] Tiny-Tapeout. *tt04-submission-template*. 2023. URL: <https://github.com/TinyTapeout/tt04-submission-template/tree/main/.github/workflows> (besucht am 09.07.2023).
- [12] Stephen R. Whiteley. *OASIS Format*. <http://www.wrcad.com/manual/xicmanual/node775.html>. Zugriff am 28. Mai 2022. 2022.
- [13] Timo Winklbauer. *Entwicklung eines ICs - Tiny Tapeout*. 2023.

- [14] Timo Winklbauer, Mathias Schoppe und Sascha Binkert. *tt04-reaction-game*. 2023. URL: <https://github.com/OSDungen/tt04-reaction-game> (besucht am 09.07.2023).