

# **Halbleitertechnologie von A bis Z**

Digitaltechnik

|                                                                           |                 |
|---------------------------------------------------------------------------|-----------------|
| <b>FPGA vs. ASIC: Syntheseablauf und HDL .....</b>                        | <b>3</b>        |
| <i><b>Zwei Wege vom Schaltungsentwurf zum Silizium .....</b></i>          | <i><b>3</b></i> |
| <i><b>Hardware-Beschreibungssprachen und der Syntheseablauf .....</b></i> | <i><b>4</b></i> |
| <i><b>Wann FPGA, wann ASIC? .....</b></i>                                 | <i><b>5</b></i> |

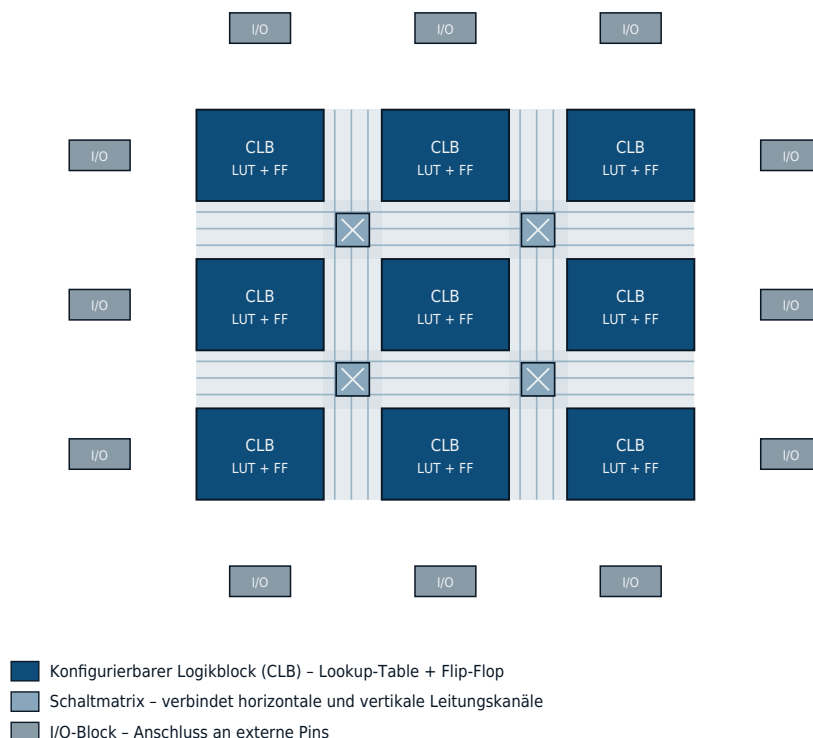
# FPGA vs. ASIC: Syntheseablauf und HDL

## Zwei Wege vom Schaltungsentwurf zum Silizium

Für die Umsetzung einer digitalen Schaltung in Hardware stehen grundsätzlich zwei Wege zur Verfügung. Ein ASIC (Application-Specific Integrated Circuit) wird für genau eine Anwendung entworfen und in den in früheren Kapiteln beschriebenen Prozessschritten – Fotolithografie, Ätzen, Dotierung, Metallisierung – als Standardzellen-Layout gefertigt; die Funktion liegt nach der Herstellung unveränderlich fest. Ein FPGA (Field-Programmable Gate Array) dagegen besteht aus einem Raster vorgefertigter, aber konfigurierbarer Logikblöcke (typischerweise Lookup-Tables mit wenigen Eingängen, kombiniert mit Flip-Flops) und einem programmierbaren Verbindungsnetzwerk, dessen Funktion erst nach der Fertigung – und im Feld beliebig oft erneut – über ein Konfigurationsbitstream festgelegt wird.

### FPGA-Grundarchitektur

Konfigurierbare Logikblöcke und programmierbares Verbindungsnetzwerk



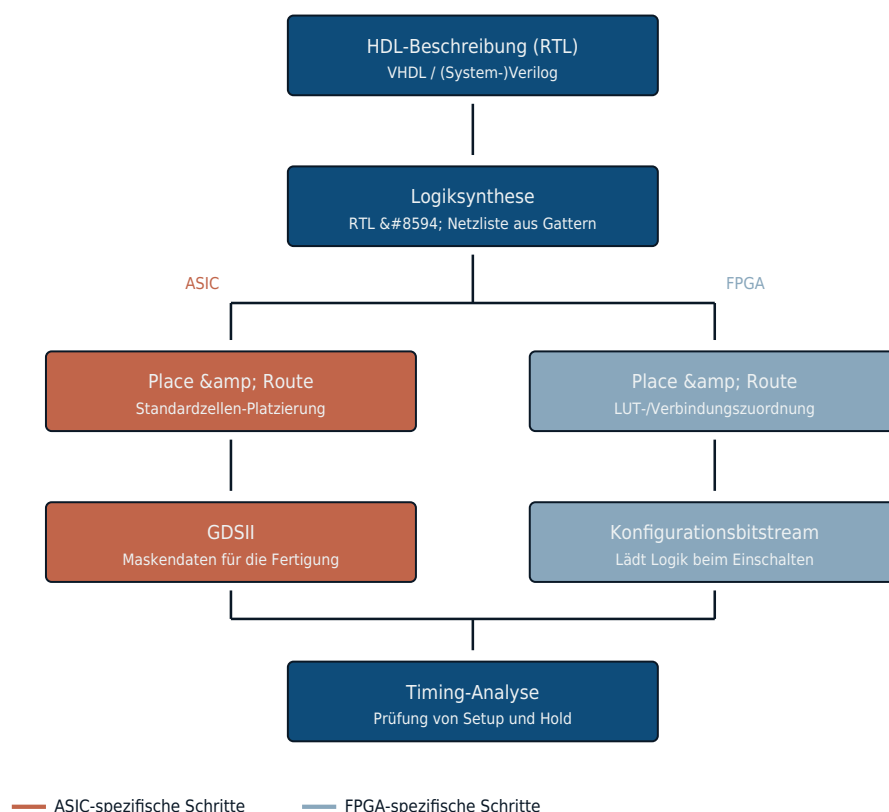
Der grundlegende Unterschied liegt damit im Zeitpunkt der Festlegung: Beim ASIC ist die Verdrahtung physisch in Metalllagen gegossen, beim FPGA wird sie elektronisch nachgebildet – mit entsprechenden Kompromissen bei Fläche,

Geschwindigkeit und Leistungsaufnahme.

## Hardware-Beschreibungssprachen und der Syntheseablauf

Sowohl ASICs als auch FPGAs werden heute nahezu ausschließlich über Hardware-Beschreibungssprachen (Hardware Description Languages, HDL) wie VHDL oder (System-)Verilog entworfen, die das gewünschte Schaltungsverhalten auf Register-Transfer-Ebene (RTL) beschreiben – also als Datenfluss zwischen Registern und die dazwischenliegende kombinatorische Logik, ohne bereits konkrete Gatter festzulegen. Ein Logiksynthese-Werkzeug übersetzt diese RTL-Beschreibung automatisch in eine Netzliste aus tatsächlichen Gattern (beim ASIC aus einer Standardzellenbibliothek, beim FPGA aus Lookup-Tables und Flip-Flops) und optimiert dabei nach Vorgaben für Fläche, Geschwindigkeit und Leistungsaufnahme.

Syntheseablauf: von der HDL zum Silizium  
Gemeinsame Schritte, dann getrennte Pfade für ASIC und FPGA



Im Anschluss übernimmt beim ASIC der Place-and-Route-Schritt die physische Platzierung der Zellen und Verdrahtung auf dem Chip, bevor daraus die

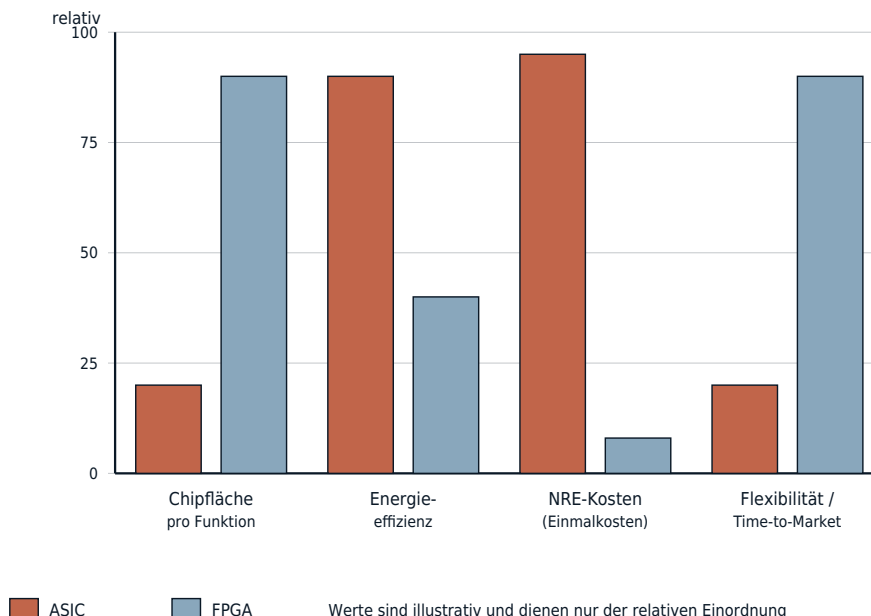
Maskendaten (GDSII) für die Fertigung entstehen; beim FPGA übernimmt ein vergleichbarer Schritt die Zuordnung der Logik auf konkrete Lookup-Tables und Verbindungsressourcen des jeweiligen Chips, bevor daraus ein Konfigurationsbitstream erzeugt wird, der beim Einschalten in das FPGA geladen wird. Beide Abläufe enden mit einer Timing-Analyse, die sicherstellt, dass alle Pfade die im vorherigen Kapitel beschriebenen Setup- und Hold-Bedingungen bei der Zielfrequenz einhalten.

## Wann FPGA, wann ASIC?

FPGAs eignen sich besonders für Prototyping, kleine bis mittlere Stückzahlen und Anwendungen, die auch nach der Auslieferung noch angepasst werden müssen, da sich ihre Funktion jederzeit neu programmieren lässt, ohne dass neue Masken oder ein neuer Fertigungslauf nötig wären. Diese Flexibilität hat jedoch ihren Preis: FPGAs benötigen für dieselbe Funktion typischerweise ein Vielfaches der Chipfläche eines vergleichbaren ASIC, takten niedriger und verbrauchen pro Rechenoperation deutlich mehr Energie, da ein erheblicher Teil der Chipfläche für die programmierbare Verdrahtung und Konfigurationsspeicher reserviert ist statt für eigentliche Logik.

FPGA vs. ASIC im Vergleich

Illustrative relative Einordnung entlang vier Kriterien



ASICs erreichen dagegen die bestmögliche Fläche, Geschwindigkeit und Energieeffizienz für eine feste Funktion, erfordern aber hohe einmalige

Entwicklungskosten (Non-Recurring Engineering, NRE) für Maskensatz und Verifikation, die sich – besonders in fortschrittlichen Technologieknoten mit Maskenkosten im zweistelligen Millionenbereich – erst ab entsprechend hohen Stückzahlen amortisieren, sowie deutlich längere Entwicklungszeiten bis zur Serienreife. Als Zwischenlösung bieten manche Anbieter Structured-ASIC-Ansätze oder eingebettete FPGA-Blöcke (eFPGA) innerhalb eines ASIC an, um einen Teil der Flexibilität bei geringeren NRE-Kosten als bei einem vollständig kundenspezifischen Design zu erhalten.