

Einführung in die Technische Informatik

VLSI-Systementwurf

Field-Programmable Gate-Arrays (1)

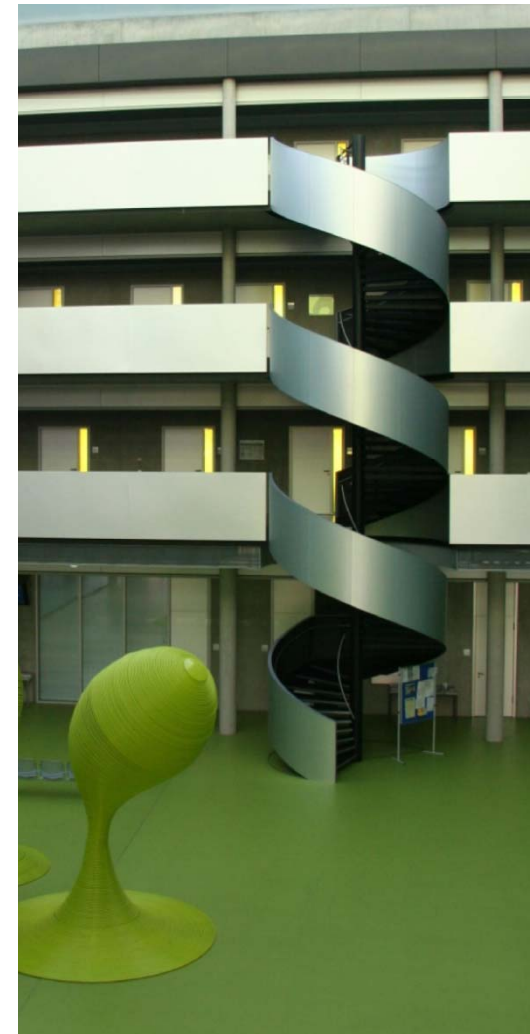
Rainer G. Spallek
Martin Zabel

TU Dresden, 07.08.2013



Gliederung

- 1 Wiederholung
- 2 FPGA-Architektur
- 3 Zusammenfassung



2 Wiederholung

Anwenderprogrammierbare IC

Merkmale:

- Field-Programmable \Leftrightarrow feldprogrammierbar.
- Vor Ort (im „Feld“) vom Anwender programmierbar.
- Hardware ist streng genommen fix. Funktionalität kann aber mittels spezieller Konfiguration „programmiert“ werden.

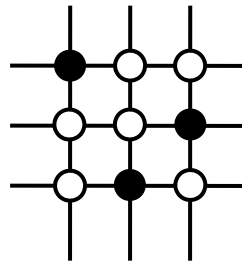
Anwendung:

- Anwendungsspezifische IC bei kleinen und mittleren Stückzahlen.
- Mehrfach neu programmierbar zwecks Optimierung und Fehlerbehebung, auch während des praktischen Einsatzes.
- Einfache Integration eines ganzen Systems auf einem Chip.
- Prototyping, Hardware-/Software-Codesign.

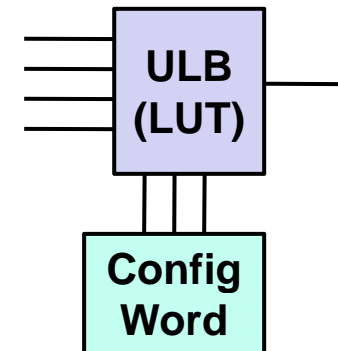
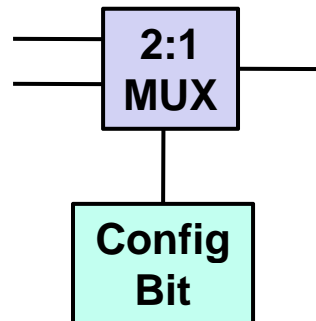
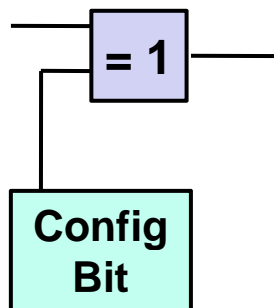
Hardwareprogrammierung

Programmiert (oder auch konfiguriert) werden können:

- Verdrahtung / Verbindung

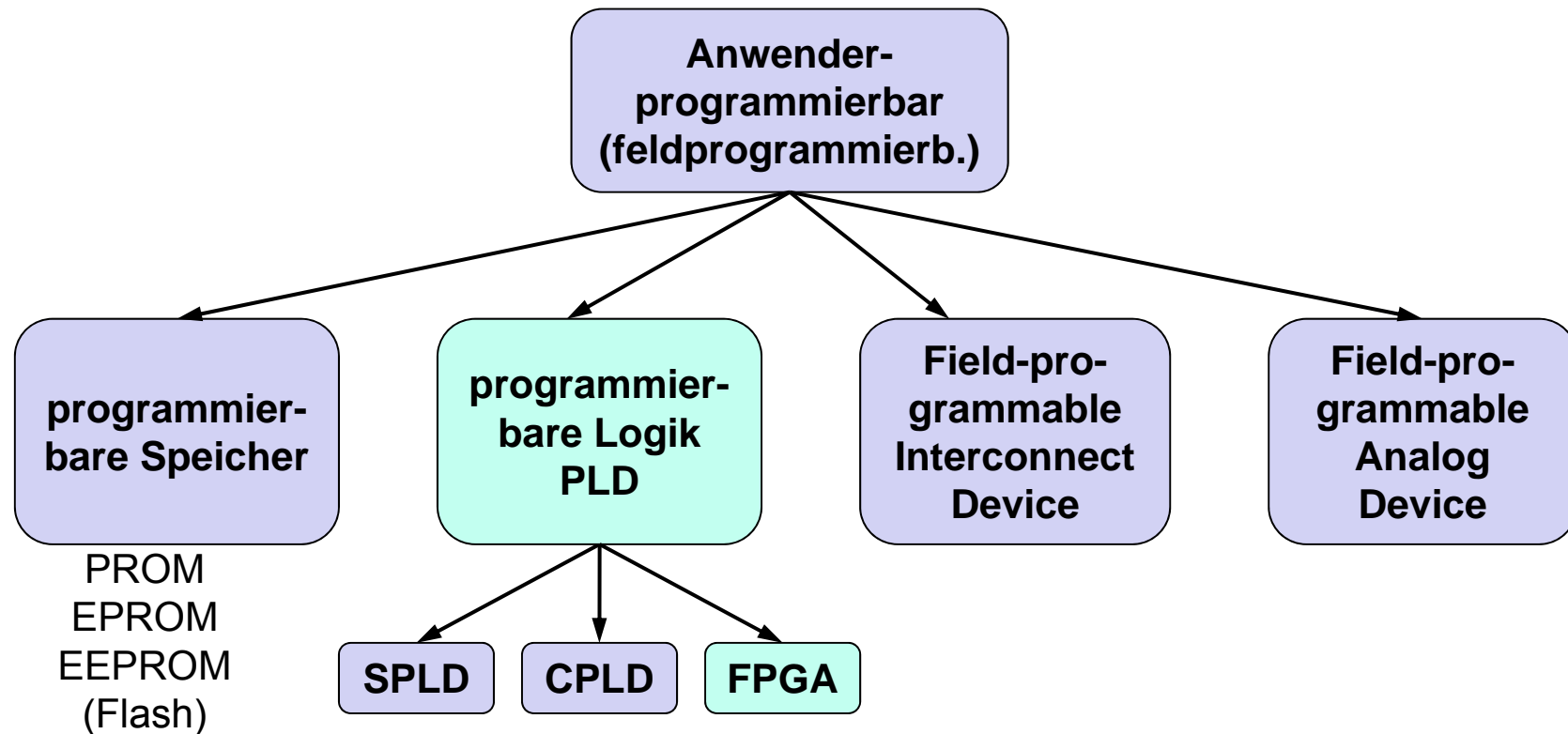


- Funktionen



- Speicher: PROM, EPROM, EEPROM (Flash)

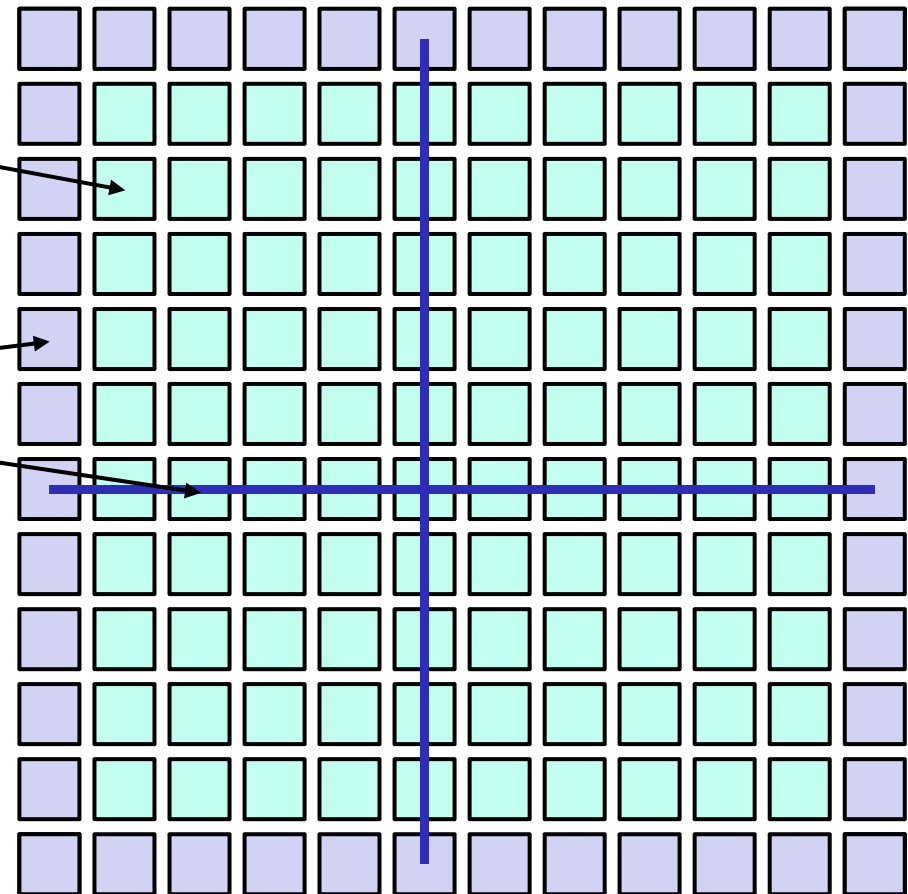
Klassifikation Anwenderprogrammierbare IC



2 FPGA-Architektur

Grundlegende Bestandteile:

- Funktionsblöcke (FB):
 - angeordnet als Matrix,
 - Multiplexer- oder LUT-basiert.
- I/O-Zellen als spezielle FB.
- Allgemeine lokale Verdrahtung, sowie globale und dedizierte Signalleitungen.
- Spezielle Hard-Makros.



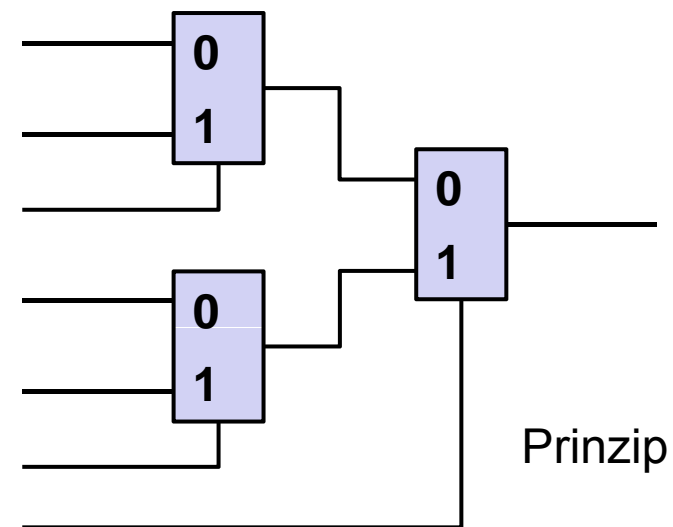
Multiplexer-basierte FB

Shannon'sches Expansionstheorem:

$$f(x_{n-1}, \dots, x_1, x_0) = (\bar{x}_i \wedge f(x_{n-1}, \dots, x_i = 0, x_1, x_0)) \vee (x_i \wedge f(x_{n-1}, \dots, x_i = 1, x_1, x_0))$$

Eigenschaften:

- Logikfunktion festgelegt durch programmierbare Verdrahtung der Multiplexer-Eingänge.
- Programmierertechnologie: zumeist Antifuse oder Flash.
- Verwendet in FPGAs von Actel (Axcelerator) und Quicklogic.

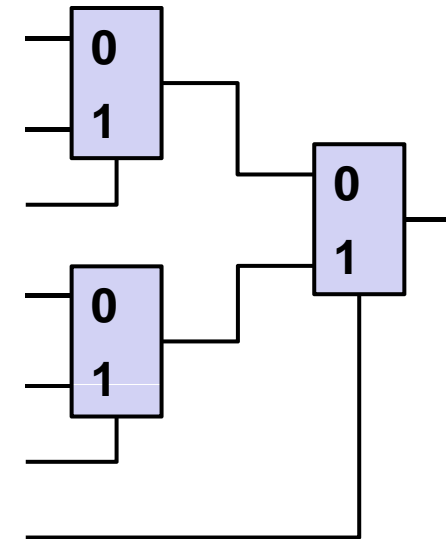


Flächenvergleich:

- Vergleichsgröße: Gatteräquivalent (GE)
- CMOS-NAND2: 1 GE = 4 Transistoren
- Multiplexerzelle nach Bild:
 - mindestens $3 \cdot 6 = 18$ Transistoren,
 - mittlere realisierte Schaltungskomplexität: 3,5 GE¹.

$$\frac{A_{MUX}}{A_{CMOS}} \approx \frac{18T / 3,5GE}{4T / 1GE} \approx 1.3$$

- Allein für die Multiplexer werden für eine typische Schaltung ca. 30% mehr Chipfläche benötigt als bei der direkten Realisierung in CMOS.



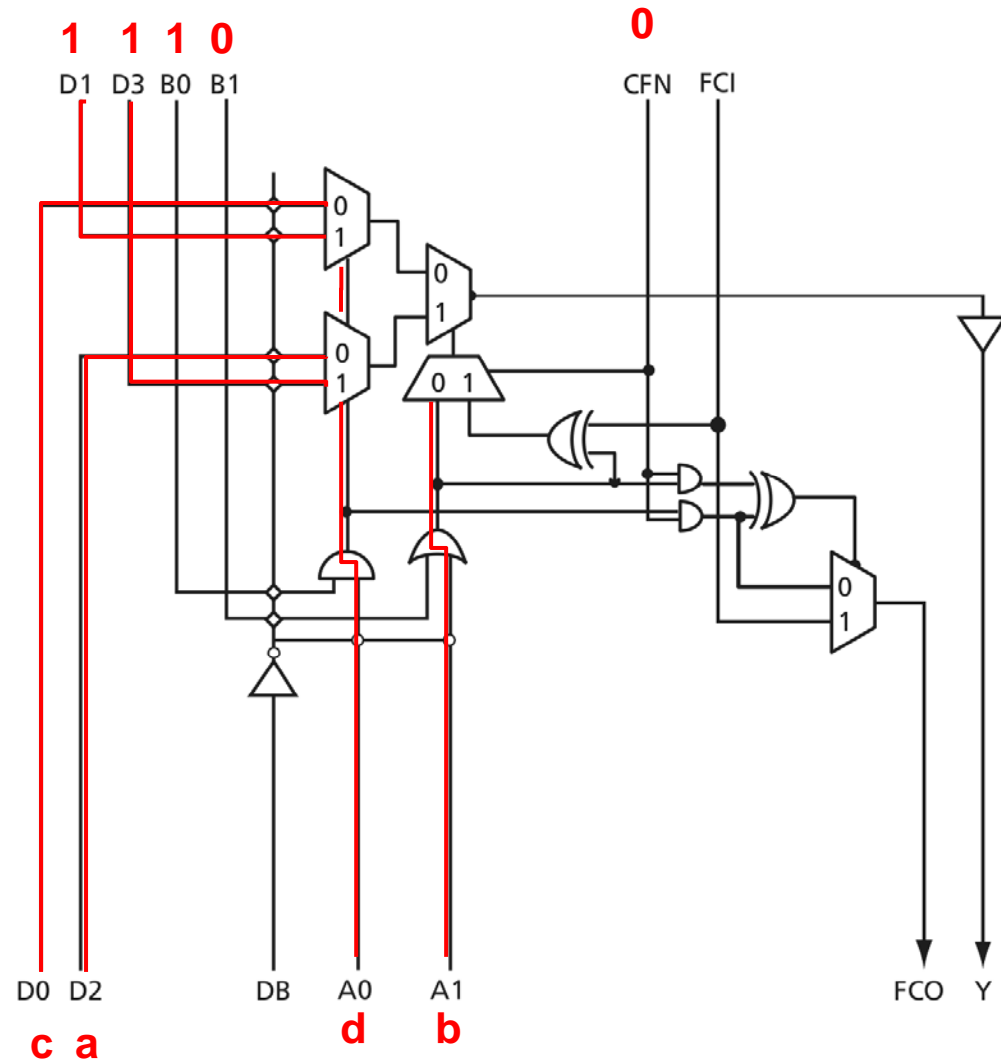
¹ F. Kesel und R. Bartholomä: Entwurf von digitalen Schaltungen und Systemen mit HDLs und FPGAs, 2. Auflage, Oldenbourg Wissenschaftsverlag, München, 2009.

Beispiel: Actel Axcelerator

- C-Cell: *Combinatorial Cell*
 - 4:1-Multiplexer + zusätzlichen Inverter:
 - ➔ >4000 mögliche Funktionen auf 5 Eingängen,
 - 4-fach-XOR benötigt 2 C-Cells.
 - Carry-Chain.
- (nächste Folie)

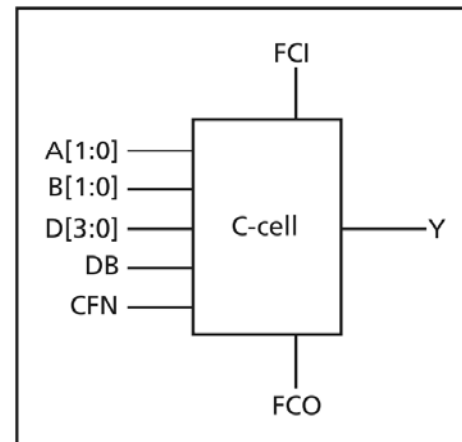
$$y = (ab) \vee (\bar{b}c) \vee d$$

Actel: Axcelerator Family FPGAs, v2.8, 2009

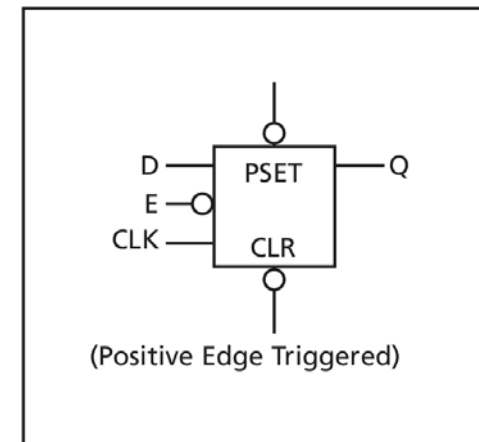


Beispiel Actel Axcelerator: (Forts.)

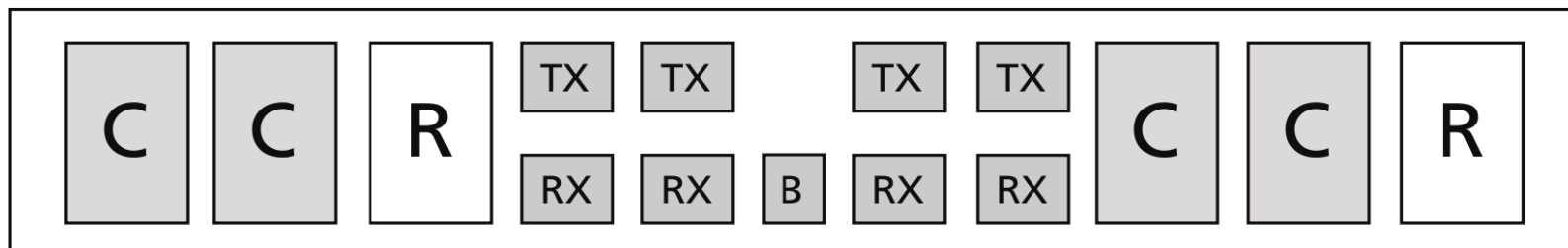
- R-Cell: *Register Cell*
 - D-FF mit Enable und asynchronem Set/Reset.
- Funktionsblock:
 - 1 SuperCluster = 2 Cluster.
 - 1 Cluster =
 - 2 C-Cell + 1 R-Cell + Carry-Lookahead für 2-Bit-Addition.



C-Cell



R-Cell



Actel: Axcelerator Family FPGAs, v2.8, 2009

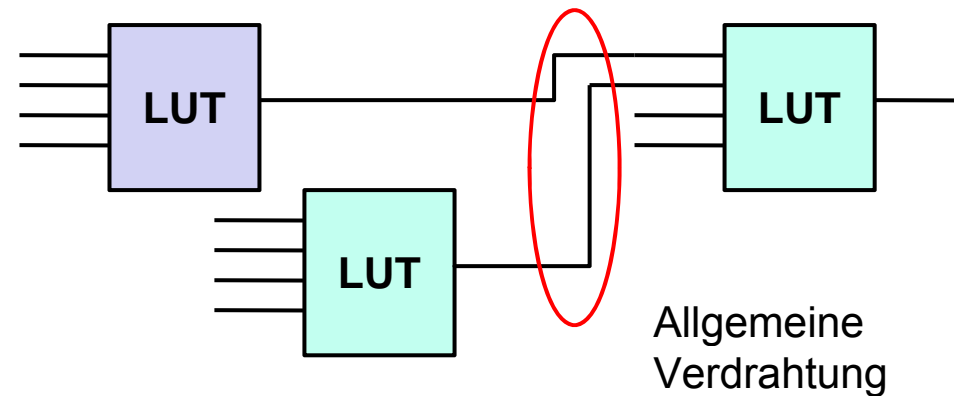
LUT-basierte FB

Eigenschaften:

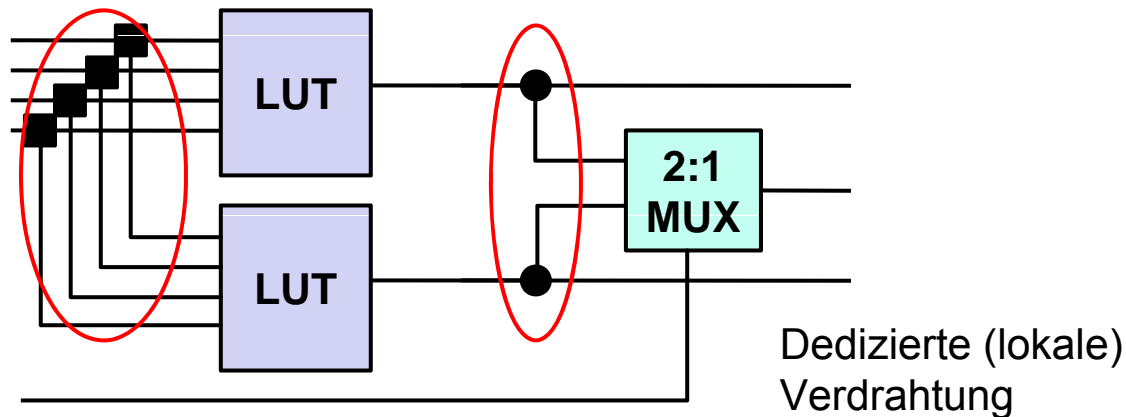
- *Look-Up-Table* → SRAM-Programmiertechnologie.
- K-LUT:
 - Beliebige Funktion mit K Eingängen = direkte Abbildung der Wahrheitstabelle
 - 2^k Tabellenplätze → $2^{(2^k)}$ unterschiedliche Boolesche Fkt. realisierbar.
- Im Allg.: Im Betrieb nur lesbar.
- Erweiterung: Im Betrieb ebenfalls schreibbar → „Distributed RAM“.
- Verwendet in FPGAs von Altera und Xilinx.
- Kompromiss Größe von K:
 - K klein: gute LUT-Auslastung, hohe LUT-Anzahl, viel Verdrahtung.
 - K groß: potenziell schlechte LUT-Auslastung, geringe LUT-Anzahl, weniger Verdrahtung.

Zusammensetzung komplexer Funktionen:

- Verkettung mehrere LUTs



- Zusätzliche Multiplexer



Flächenvergleich:

- Vergleichsgröße: Gatteräquivalent (GE)
- CMOS-NAND2: 1 GE = 4 Transistoren
- 4-LUT ohne Programmierlogik:
 - ≈ 100 Transistoren für 16 SRAM-Zellen,
 - mittlere realisierte Schaltungskomplexität: 5 GE¹.

$$\frac{A_{LUT}}{A_{CMOS}} \approx \frac{100T / 5GE}{4T / 1GE} = 5$$

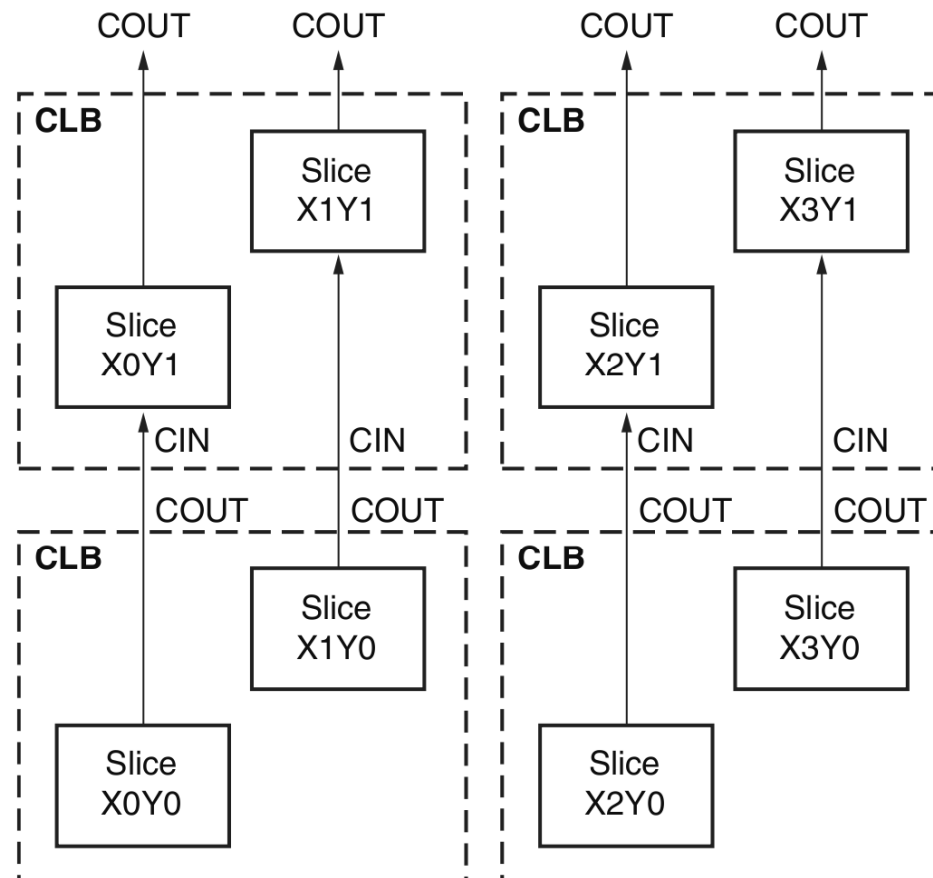
- Allein für die SRAM-Zellen wird für eine typische Schaltung ca. die **5-fache** Chipfläche benötigt als bei der direkten Realisierung in CMOS.
- Zusätzlich Chipfläche für Carry-Chain, FF und programmierbare Verdrahtung!

¹ F. Kesel und R. Bartholomä: Entwurf von digitalen Schaltungen und Systemen mit HDLs und FPGAs, 2. Auflage, Oldenbourg Wissenschaftsverlag, München, 2009.

Beispiel: Xilinx CLB (*Configurable Logic Block*)

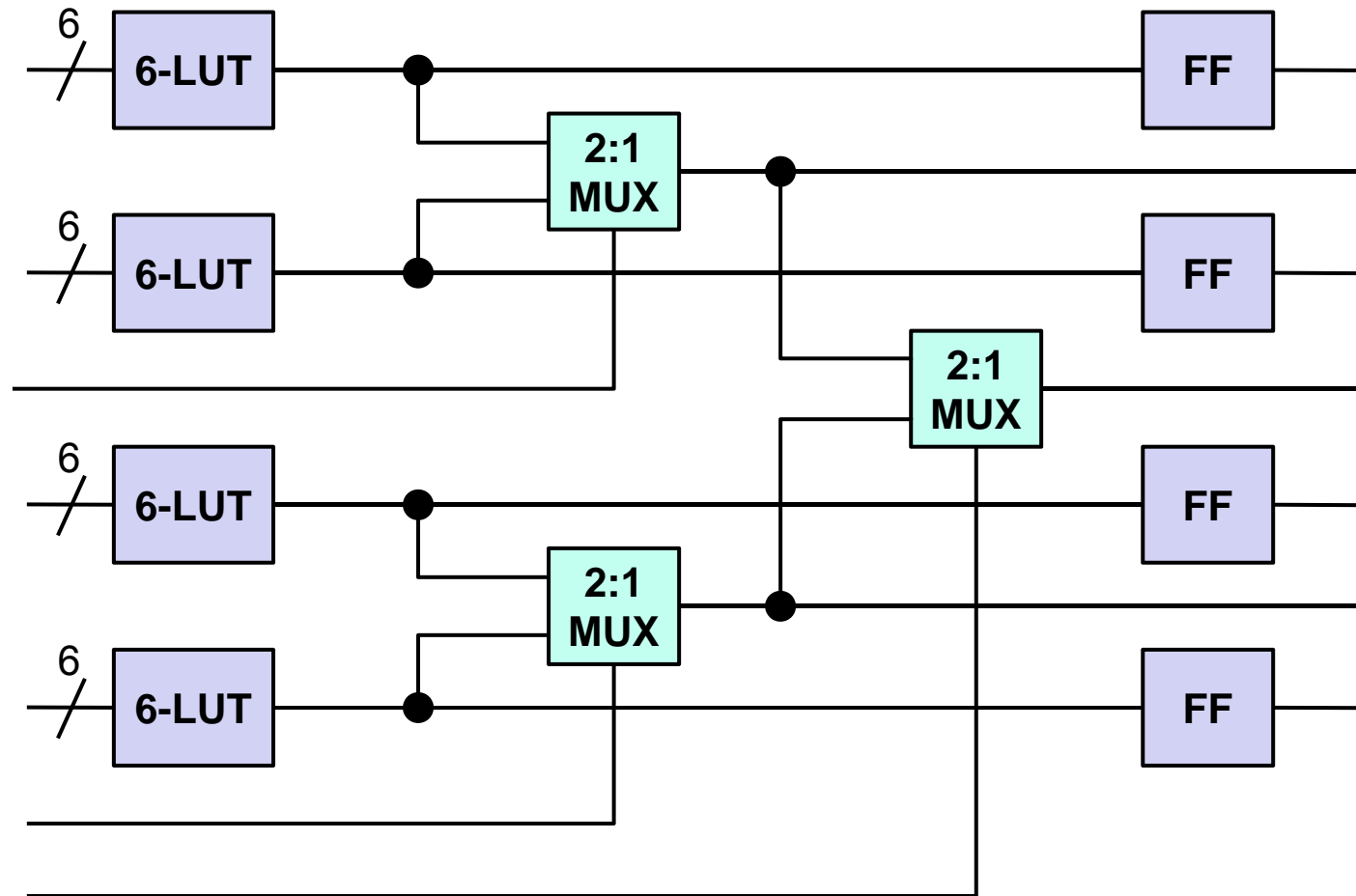
- Funktionsblock:
 - 1 CLB = S Slices (versch. Typen)
 - 1 Slice = L LUTs + L FFs + Carry-Chain + Multiplexer + ...
- Virtex-II:
 - S = 4, L = 2, 4-LUT, LUT auch als RAM oder Shift-Register.
 - Multiplexer für 5-LUT, 6-LUT, 7-LUT, 8-LUT.
- Virtex-5:
 - S = 2, L = 4, 6-LUT, auch als 2 5-LUT konfigurierbar.
 - SLICE-L : LUT nur lesbar.
 - SLICE-M: LUT auch schreibbar → RAM oder Shift-Register.
 - Multiplexer für 7-LUT, 8-LUT.

Beispiel: Xilinx Virtex-5 CLB



Xilinx: UG190 *Virtex-5 FPGA User Guide*, v5.2, 2009

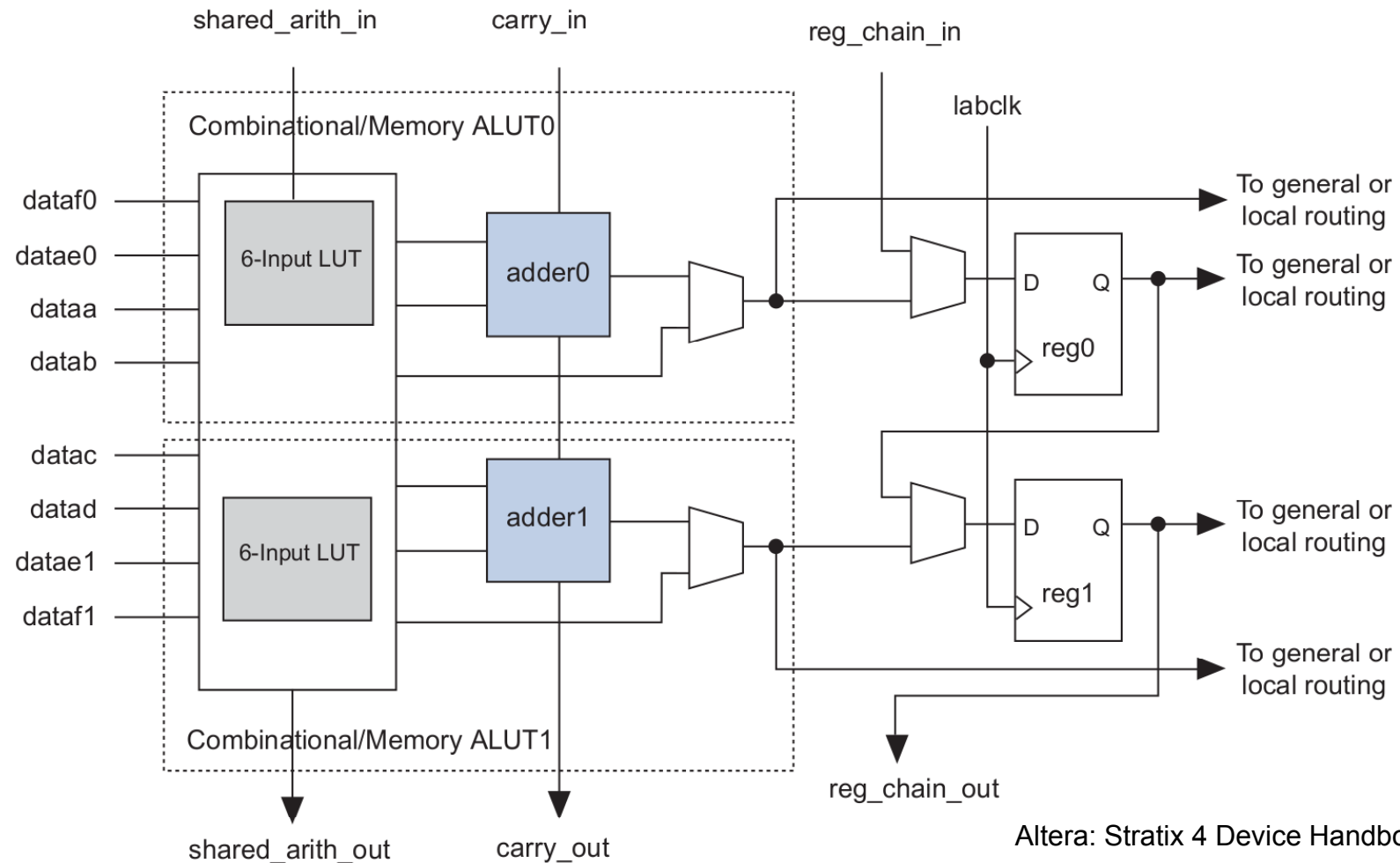
Beispiel Virtex-5 Slice (Vereinfachte Darstellung)



Beispiel: Altera LAB (*Logic Array Block*)

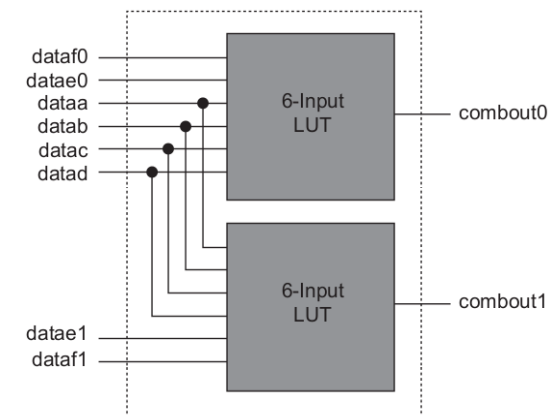
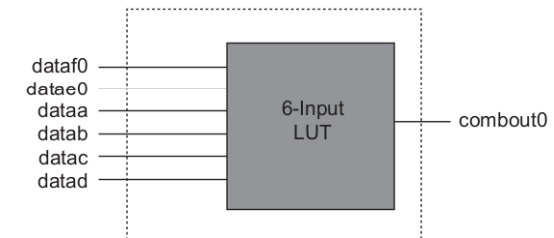
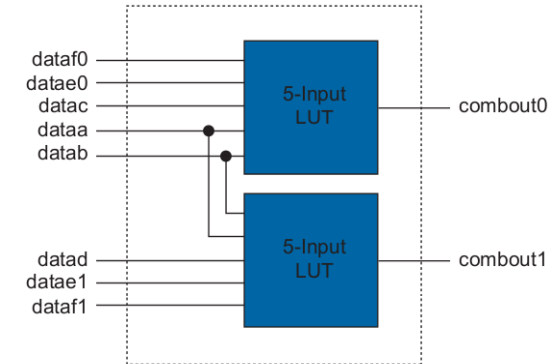
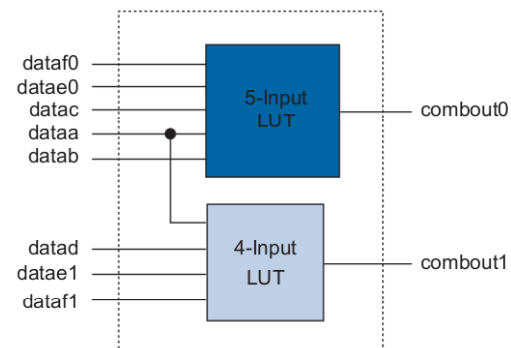
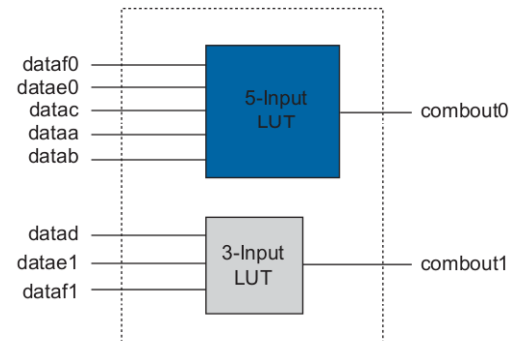
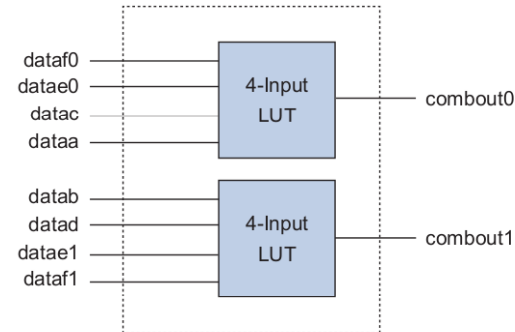
- Funktionsblock:
 - 1 LAB = L Logikzellen + Carry-Chain + ...
- Beispiel Stratix:
 - L=10; 1 Logikzelle (LE) = 1 4-LUT + 1 FF,
 - Direkte Verb. zwischen LE für ausgew. Fkt. mit >4 Eingängen.
- Beispiel Stratix-IV:
 - L=10; 1 Logikzelle (ALM) = 1 Complex-LUT + 2 FFs.
 - Complex-LUT mit 8 Eingängen:
 - 2 unabhängige 4-LUTs, oder
 - unabhängige 3-LUT + 5-LUT, oder
 - 1 6-LUT, oder
 - 2 6-LUT mit 2 gemeinsamen Eingängen, oder ...
 - LAB alternativ konfigurierbar als *Memory LAB* (MLAB)
→ Je Logikzelle: 64x1 Simple Dual-Port SRAM.

Beispiel: Altera Stratix-IV ALM



Altera: Stratix 4 Device Handbook

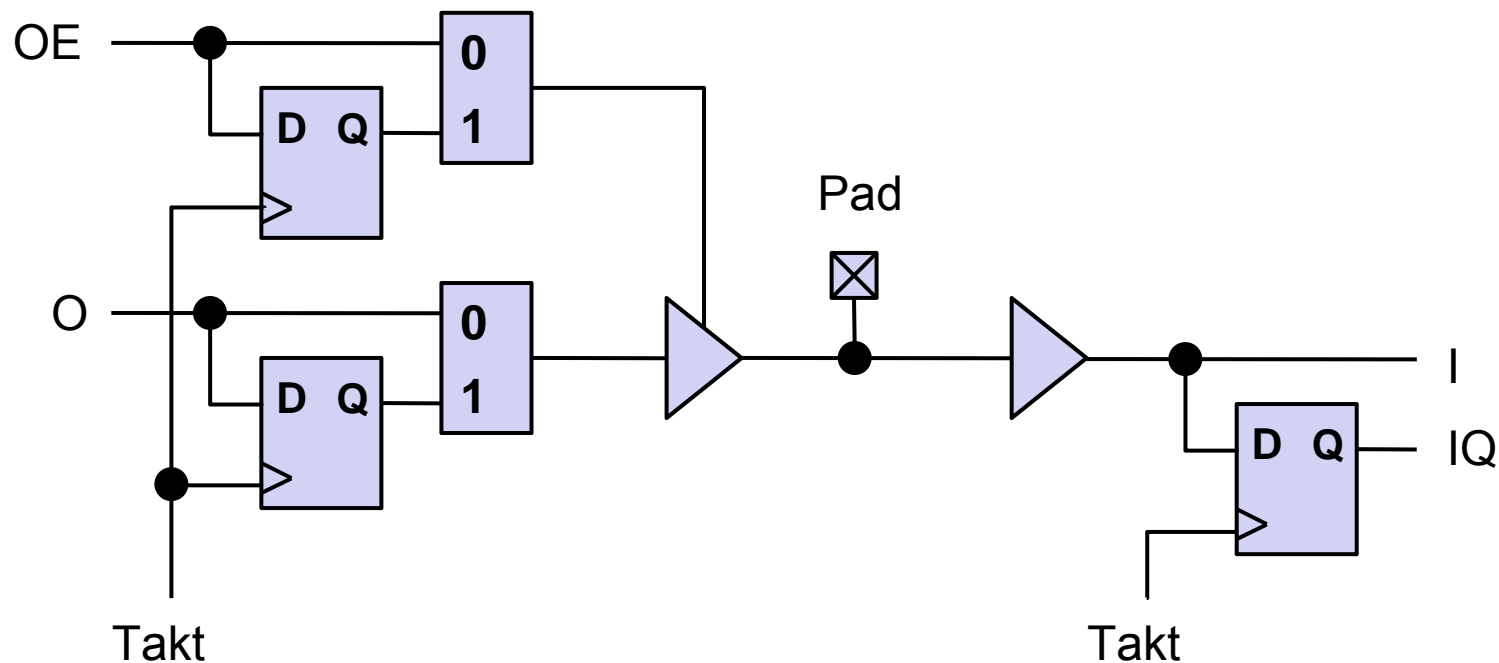
Beispiel: Altera Stratix-IV ALM LUT-Konfigurationen



Altera: Stratix 4 Device Handbook

I/O-Zellen

- Optionale Register für kurze Setup- / Hold- und Clock-to-Output-Zeiten.
- Programmierbar auf versch. I/O-Standards.
- Dual-Data-Rate (DDR) u.a. Spezialfunktionen.

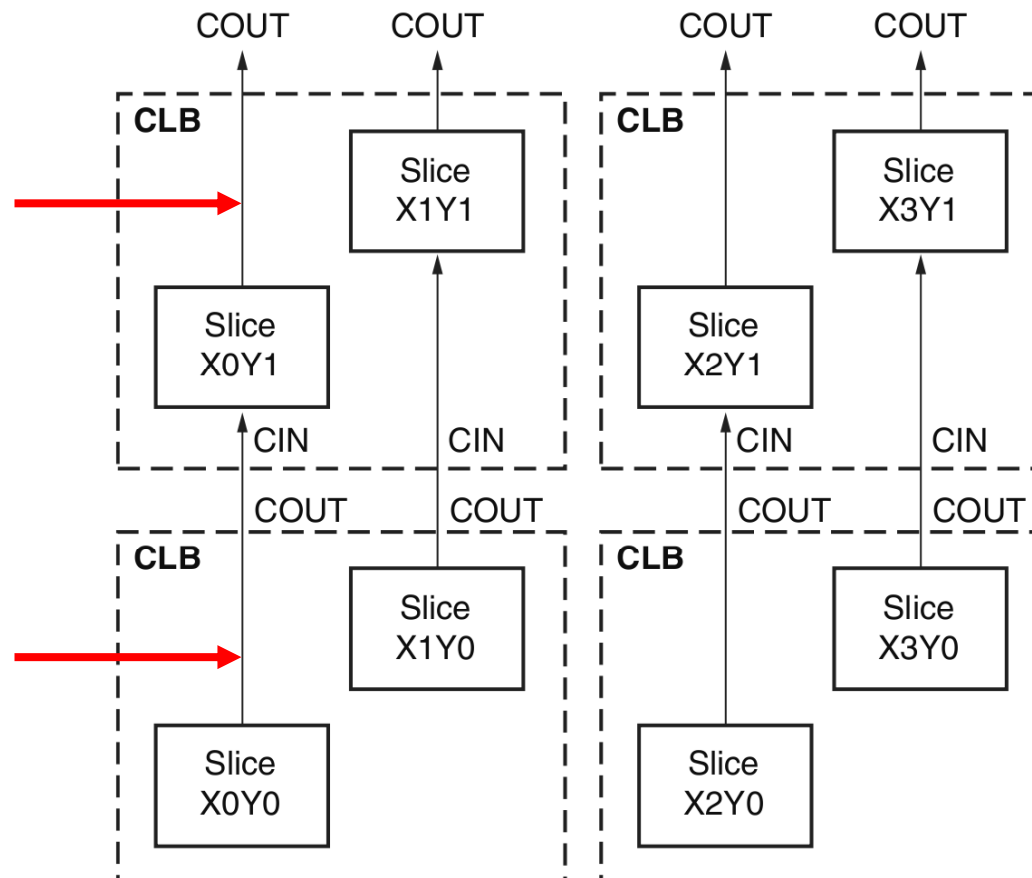


Verdrahtung

Typen:

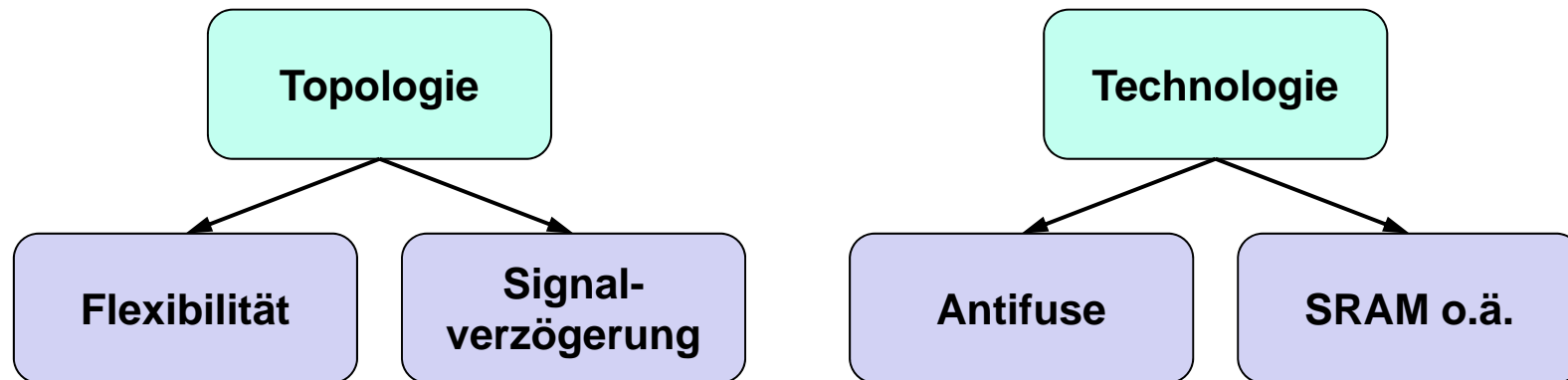
- Globale Signale:
 - Takt, Reset, 0 und 1
 - Taktsignale über Takt-Baum → low-skew Netzwerk.
- Dedizierte Verbindungen:
 - I.d.R. nicht programmierbar.
 - Carry-Chain, Register-Chain, LUT-Erweiterung.
- Allgemeine Verbindungen:
 - Regelmäßige Anordnung von Signalleitungen,
 - Verschaltung über *Programmable Interconnect Points* (PIPs) und progr. Switch-Matrizen.

Beispiel: Dedizierte Verbindungen im Virtex-5 FPGA



Xilinx: UG190 *Virtex-5 FPGA User Guide*, v5.2, 2009

Kriterien für die allgemeine Verdrahtung



Topologie

- Ziel: Hohe Flexibilität bei
 - kurzen Verzögerungszeiten,
 - kaum verschwendeter Chipfläche.
- Extrembeispiele:
 1. Viele lange Leitungen mit vielen PIPs.
 2. Viele kurze Leitungen, verknüpfbar mit PIPs.

Extrembeispiel 1:

- Gitterartige Anordnung von Leitungen, die von Rand zu Rand verlaufen.
- Anschluss aller FB in einer Zeile / Spalte.
- Sehr viele Leitungen je Zeile / Spalte.
- Vorteile:
 - Verbindung beliebiger Ein-/Ausgänge über maximal 3 PIPs (Hops).
- Nachteile für kurze realisierte Verbindungen:
 - Verzögerungszeit größer als Notwendig.
 - Schlechte Auslastung → potentiell verschwendete Chipfläche.
- Ungeeignet für SRAM-Technologie da hohe Anzahl von PIPs.

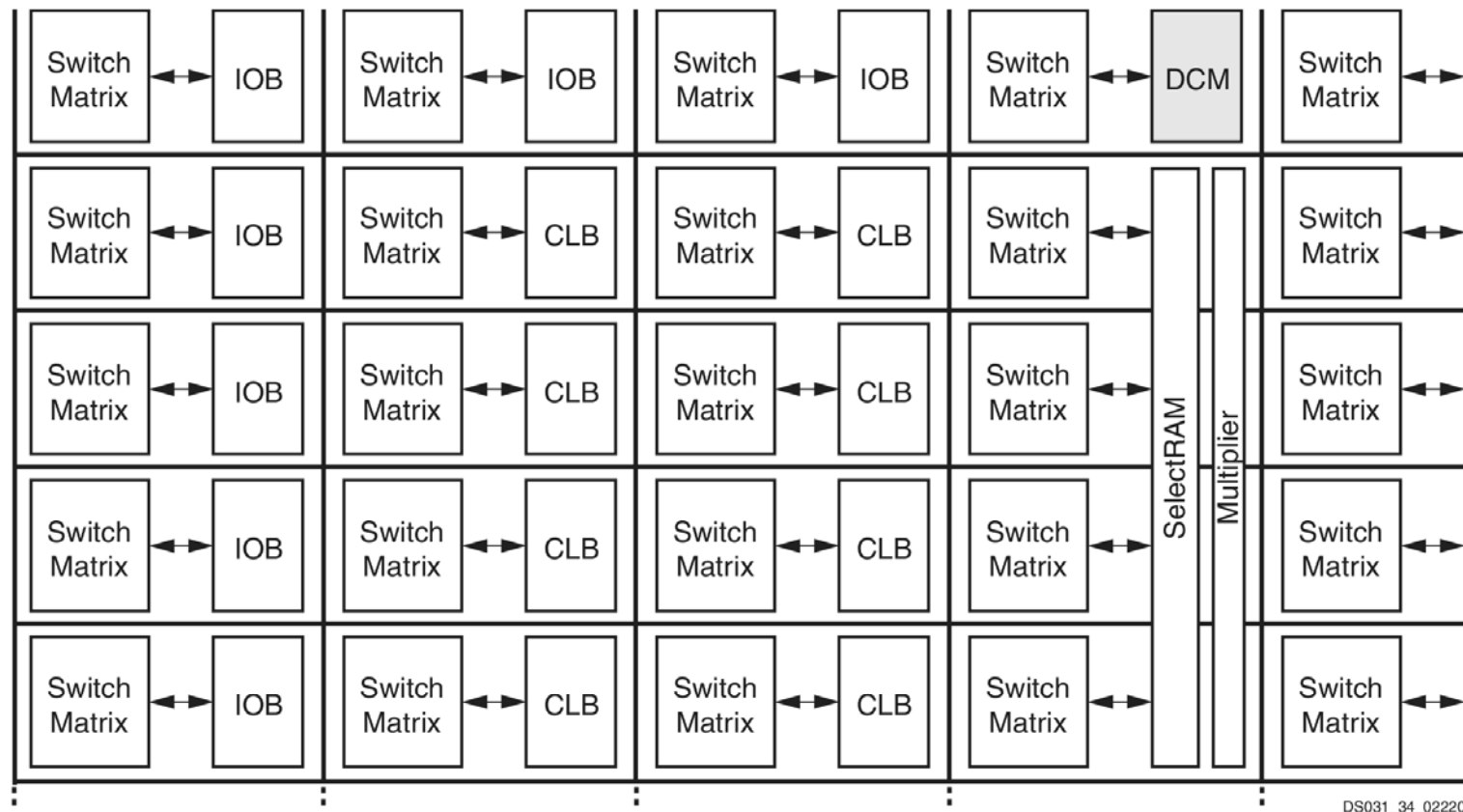
Extrembeispiel 2:

- Gitterartige Anordnung von Leitungen zwischen benachbarten FB.
- Verkettung zu langen Leitungen über PIPs.
- Anschluss nur eines FB an ein Leitungsstück.
- Beispiel: Xilinx XC3000.
- Vorteile:
 - Kurze Verzögerungszeiten für kurze Verbindungen.
 - Alternative Verdrahtungswege für lange V. (Manhattan-Distanz).
- Nachteile:
 - Überproportional lange Verzögerungszeiten für lange V., da viele Hops über viele PIPs.

Kompromiss:

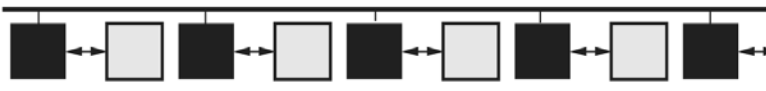
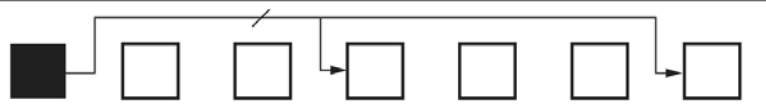



- Hierarchie: kurze + lange Signalleitungen mit wenigen / vielen angeschlossenen FB.
- Organisation:
 - 1 Hierarchie: Aufteilung in Blöcke und diese wiederum in Unterblöcke (Bsp.: Actel Axcelerator)
 - Hierarchie je FB: Je FB der gleiche Satz von kurzen u. langen Leitungen. (Bsp.: Xilinx Virtex/Spartan, Altera Stratix/Cyclone).

Beispiel: Allgemeine Verdrahtung im Virtex-II (1)



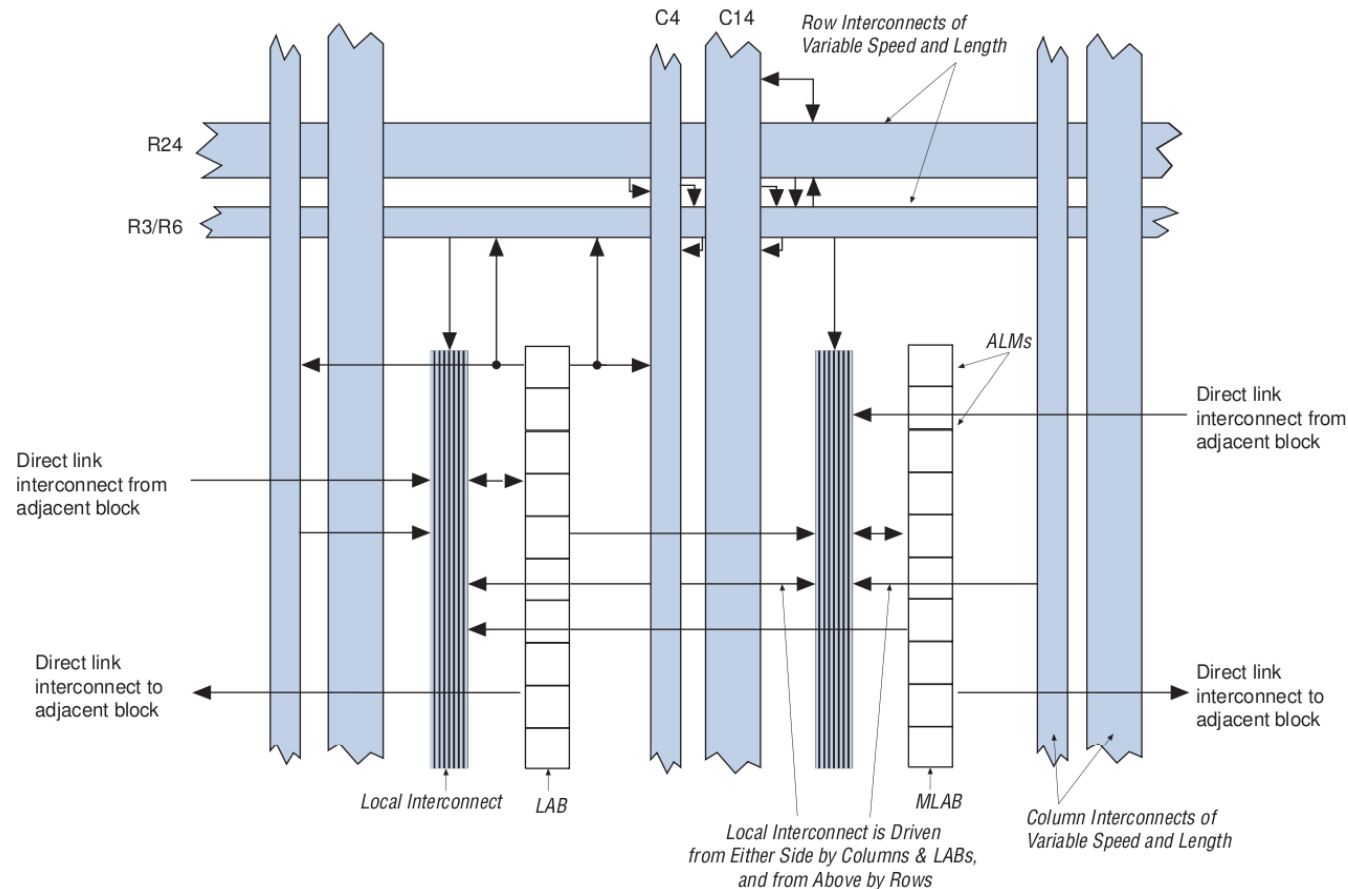
Xilinx: DS031 *Virtex-II Platform FPGAs Complete Datasheet*, v3.5, 2007

Beispiel: Allgemeine Verdrahtung im Virtex-II (2)

24 Horizontal Long Lines 24 Vertical Long Lines	
120 Horizontal Hex Lines 120 Vertical Hex Lines	
40 Horizontal Double Lines 40 Vertical Double Lines	
16 Direct Connections (total in all four directions)	
8 Fast Connects	

Xilinx: DS031 *Virtex-II Platform FPGAs Complete Datasheet*, v3.5, 2007

Beispiel: Allgemeine Verdrahtung im Stratix-5



Altera: Stratix 5 Device Handbook, Volume 1, Juni 2012, S. 1-1

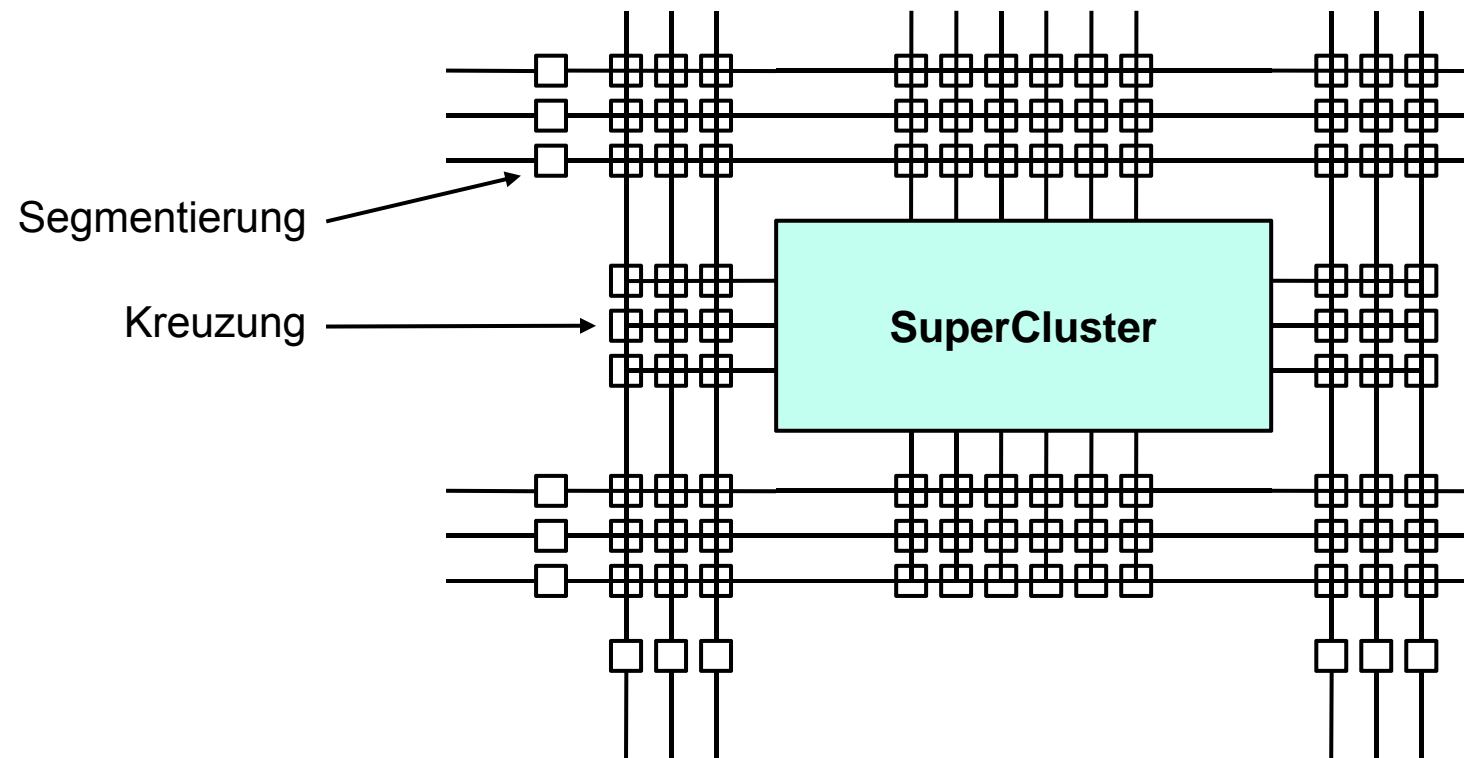
Technologie

- Antifuse-Technologie:
 - **Einmalige** Programmierung der Verdrahtung in der Metallebene.
 - ➔ Chipfläche steht komplett für Logik zur Verfügung (Sea of Gates).
 - ➔ Zusätzlicher Widerstand: Antifuse.
 - PIP-Beispiele: T-Kreuzung, X-Kreuzung, Segmentierung.

- SRAM-Technologie:
 - Pass-Transistoren oder Multiplexer gesteuert durch **mehrfach programmierbare** SRAM-Zelle.
 - ➔ Zusätzliche Chipfläche.
 - ➔ Zusätzliche Kapazitäten:
 - Pass-T leitend ➔ Source- und Drain-Kapazität.
 - Pass-T gesperrt ➔ Drain-Kapazität.
 - ➔ Zusätzlicher Widerstand: Source-Drain-W. des Pass-T.
 - PIP-Beispiele: Switch-Box, Connection-Box, Switch-Matrix.
- Ziel: So wenig wie nötige PIPs.
- Ausnutzung von:
 - Vertauschbaren Eingängen bei LUTs.
 - Platzierung der Funktion auf beliebigen FB, sofern nicht an Carry-Chain gebunden.

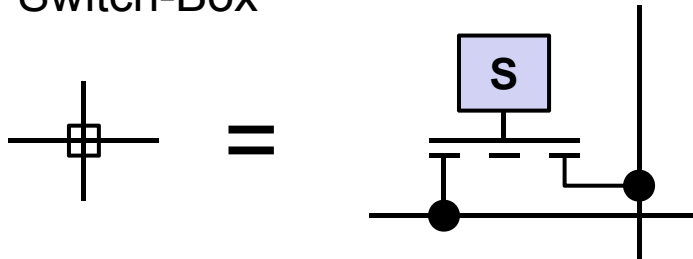
Antifuse-Verbindungen:

Beispiel: Actel Axcelerator (vereinfacht)

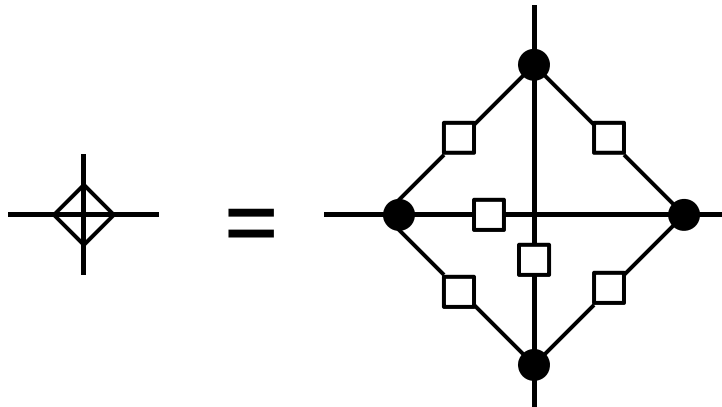


SRAM-basierte Verbindungen:

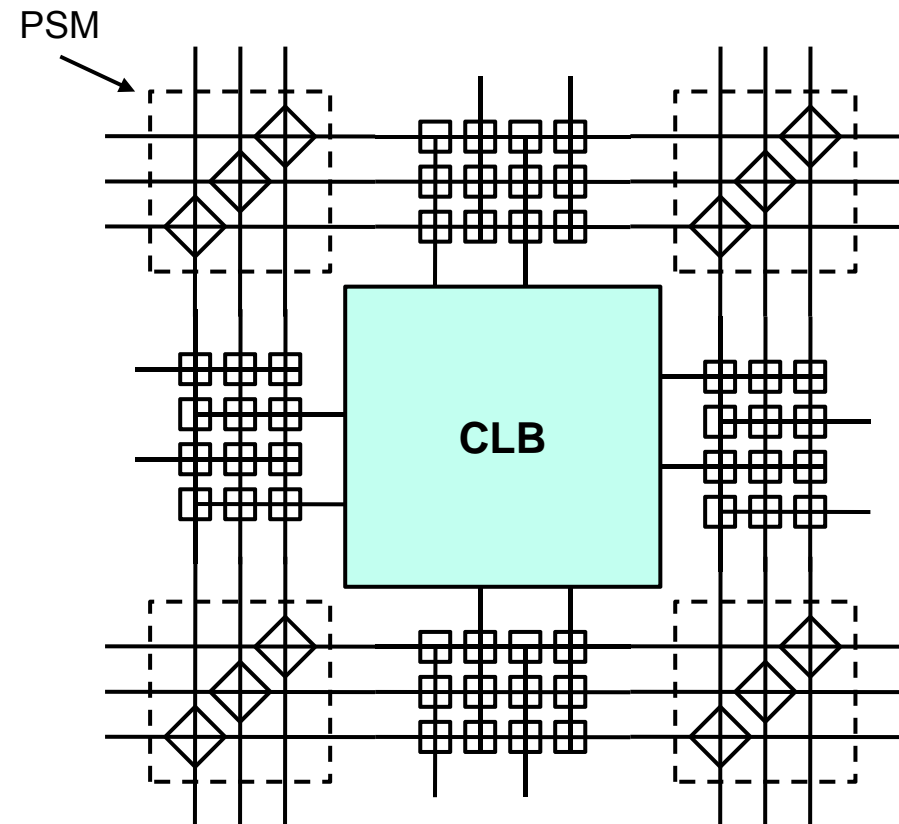
- Switch-Box



- Connection-Box

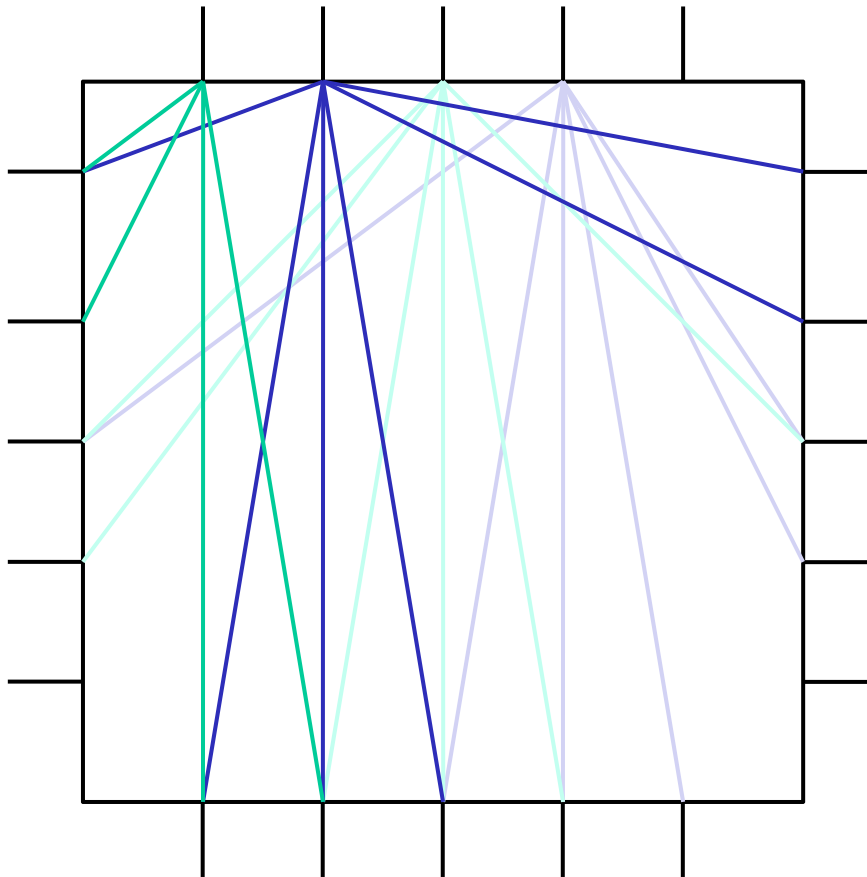


Beispiel: Spartan/XL (vereinfacht)



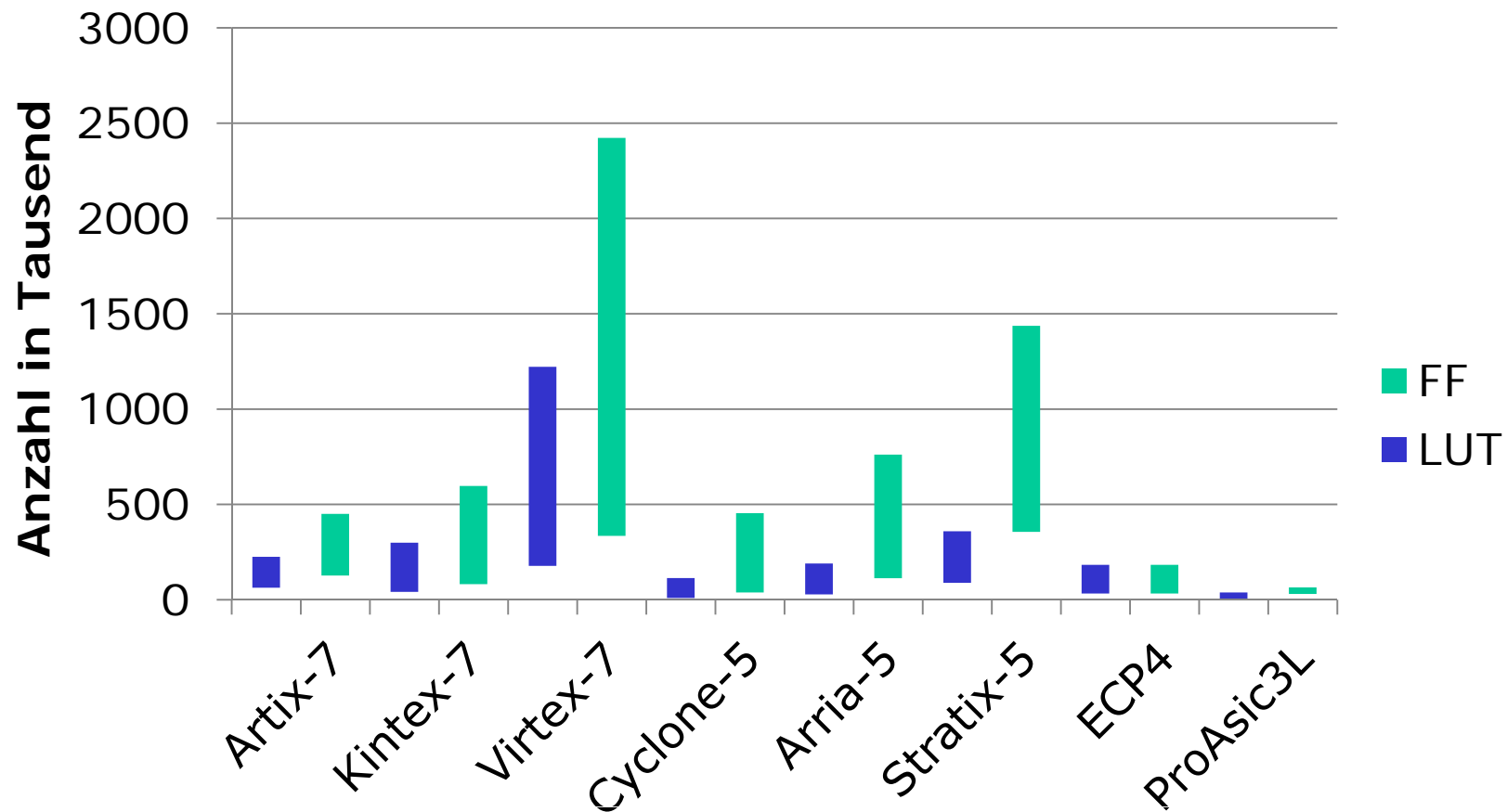
Xilinx: *Spartan and Spartan-XL FPGA Families Data Sheet*, v1.8, 2008

- Switch-Matrix (allg.)



Xilinx: XC3000 Series Field Programmable Gate Arrays, v3.1, 1998

Vergleich zwischen FPGA-Familien



3 Zusammenfassung

- Funktionsblöcke:
 - Multiplexer-basiert,
 - LUT-basiert.
- I/O-Zellen inklusive FFs.
- Verdrahtung:
 - Topologie: Baum-artig, „Periodisches Gitter“.
 - Technologie:
 - Antifuse → Chipfläche im Wesentlichen für Logik.
 - Pass-T / Multiplexer + SRAM-Zelle. → Mehrfach programmierbar.