

Vortrag zum Hauptseminar „SOC – Designflow“

Martin Fröhlich, s0238944@mail.inf.tu-dresden.de

Dresden, 16.Juli.2008

Inhalt

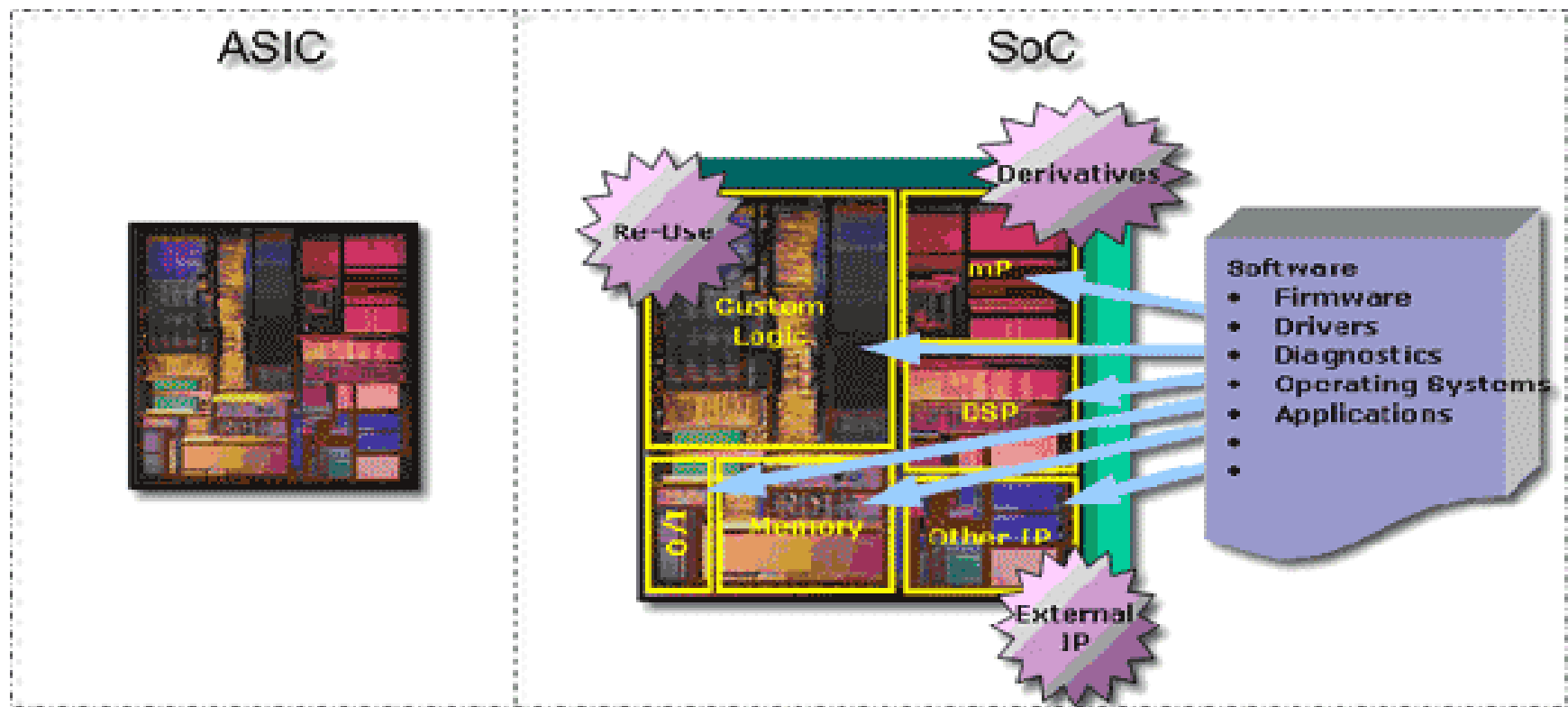
- 1. System on a chip**
- 2. SOC Designflow**
- 3. FPGA - Prototyping**
- 4. Silicon Virtual Prototyping**
- 5. Zusammenfassung**

Quellen

1. SOC

- **System on a Chip** oder **Ein-Chip-System**
- Integration mehrerer Komponenten oder ganzer Systeme auf einem Stück Silizium
- Software wird parallel zur Hardware entwickelt
- Aufteilung in Funktionsblöcke
- Wiederverwendung von Baublöcken (Makros)
- System wird *top-down*, Baublöcke werden *bottom-up* entworfen

1. SOC



1. SOC - Vorteile

- ↓ Systemgröße (Verbrauch, Fläche ...)
- ↓ Produktionskosten und Entwicklungskosten
- ↑ Automatisierungsgrad
- ↑ Wiederverwendbarkeit
- ↓ Entwicklungszeit - time to market
- ↑ Produktqualität

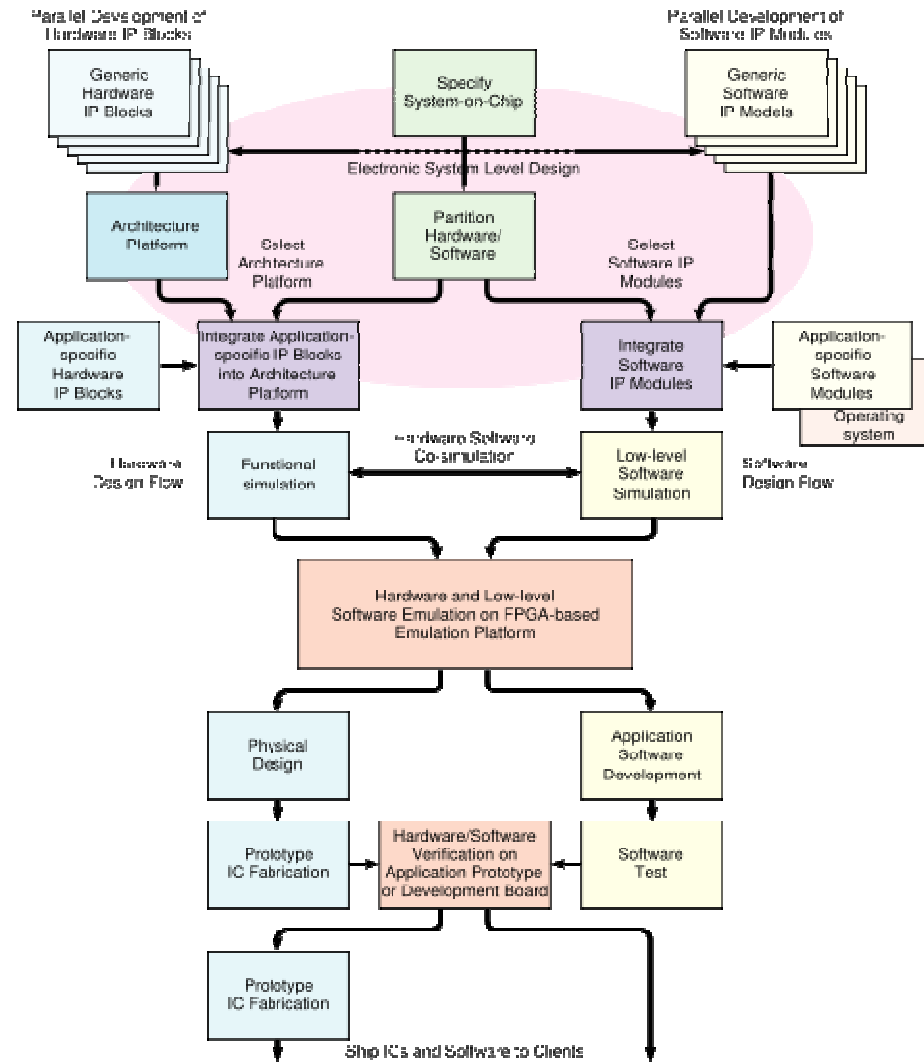
1. SOC - Einsatzbereiche

- general purpose Prozessoren
- Eingebettete Computer
- Mobilfunk
- Kommunikation (drahtlos, -gebunden)
- ...

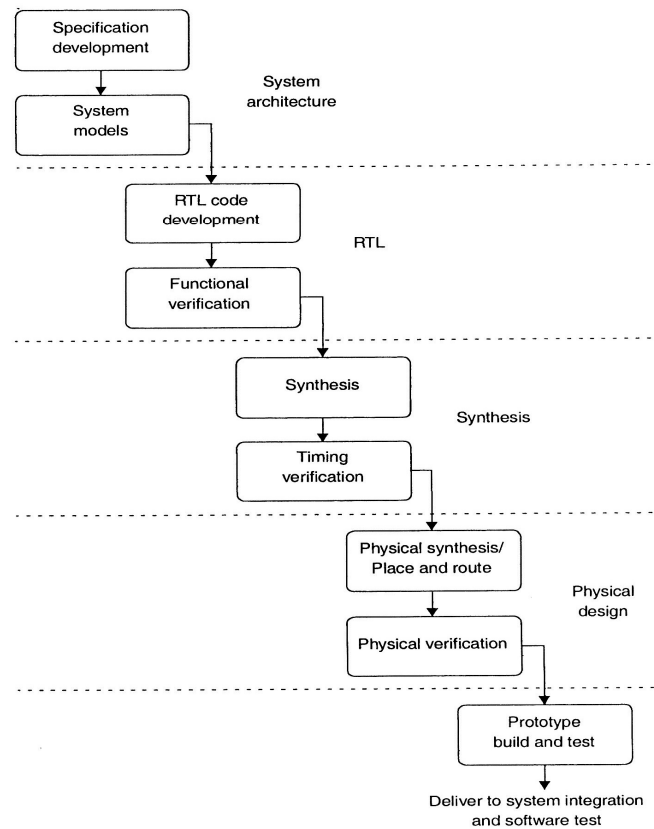
1. SOC - Wiederverwendung

- Hard IP:
 - Physikalisches Layout
 - Auf Technologie festgelegt → schwer modifizierbar
- Soft IP:
 - Komplet in einer HW-Beschreibungssprache beschrieben
 - Leichter modifizier- und verifizierbar
- Firm IP:
 - Bestehen aus HW-Beschreibung und technologieabhängigen Platzierungsdaten

2. SOC – Entwurfsfluss



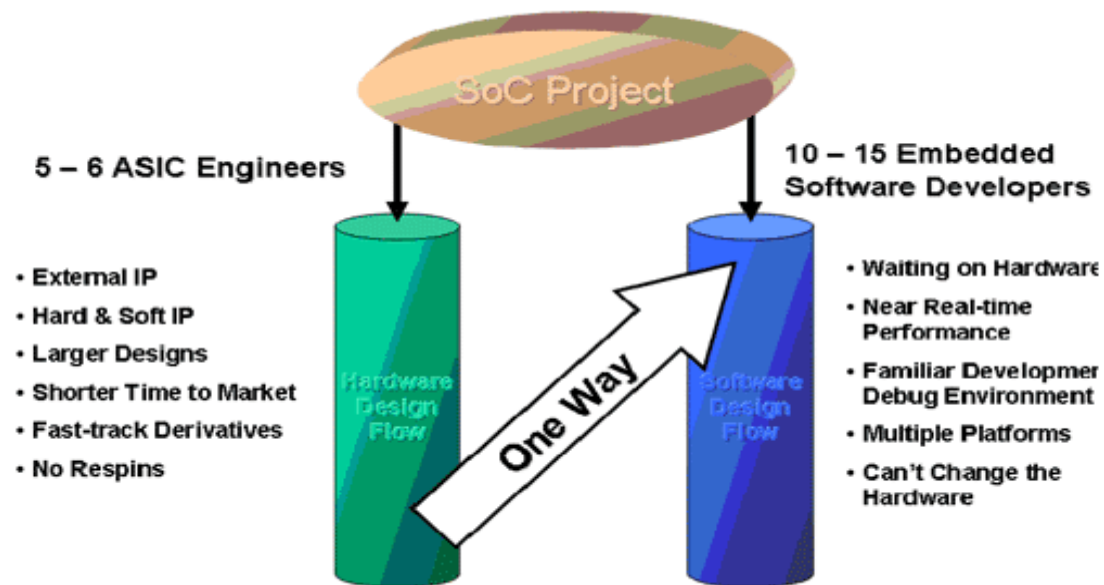
2. SOC - Wasserfallmodell – One-Way Process



Traditional waterfall ASIC design flow

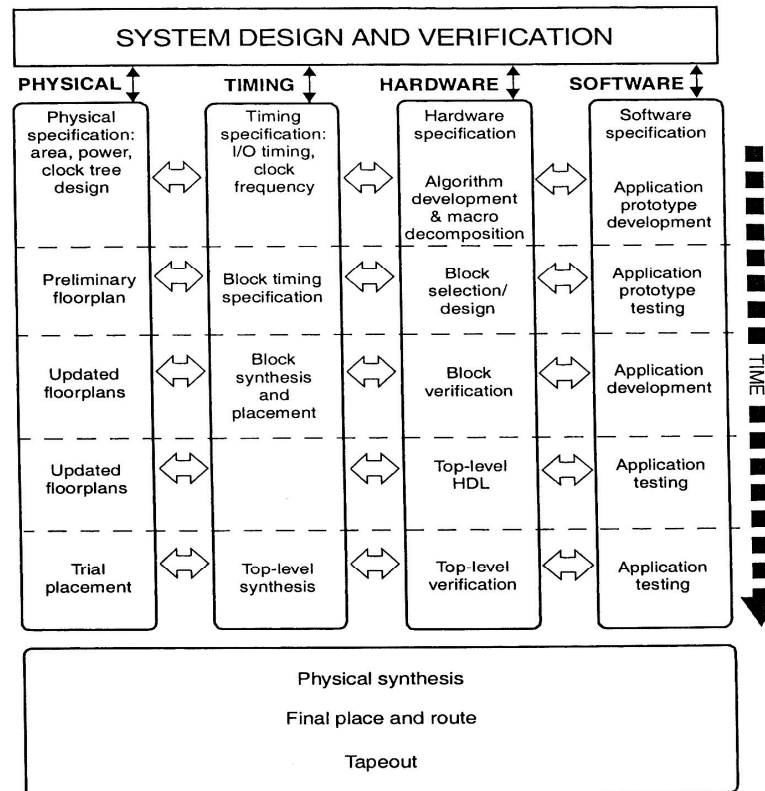
2. SOC - Wasserfallmodell – One-Way Process

- Definition eines Hardware- und eines Softwareprojektes



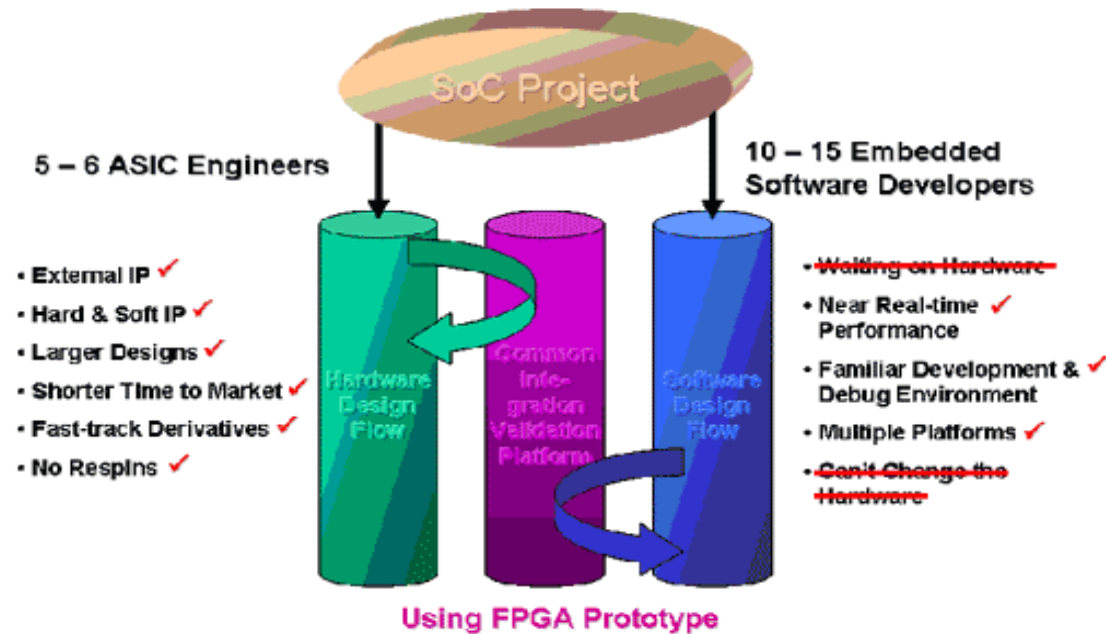
3. FPGA Prototyping

Goal: Maintain parallel interacting design flows



3. FPGA Prototyping

- Einführung eines FPGA Prototyps als Schnittstelle
→ Softwareentwicklung parallel zur Hardwareentwicklung

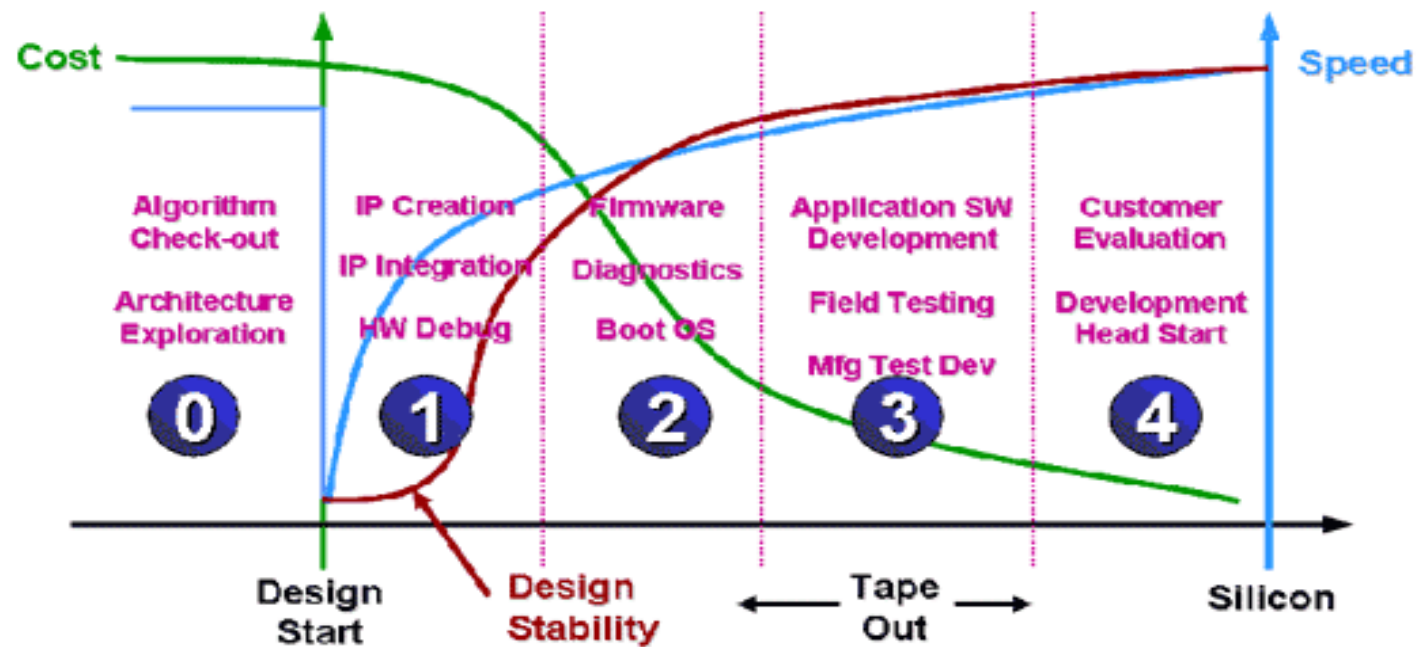


3. FPGA Prototyping

Probleme:

- Design for prototyping: nicht alle Konstrukte sind auf FPGA abbildbar
z.B.: Multiple Drivers, Gated Clock, ...
- Logiksynthese: Design häufig zu groß für ein FPGA →
Aufteilung auf mehrere FPGA für Synthesetools oft problematisch
- Inkrementelles Design: häufig nur kleine Änderungen an Schaltung →
Mapping sollte inkrementell erfolgen können

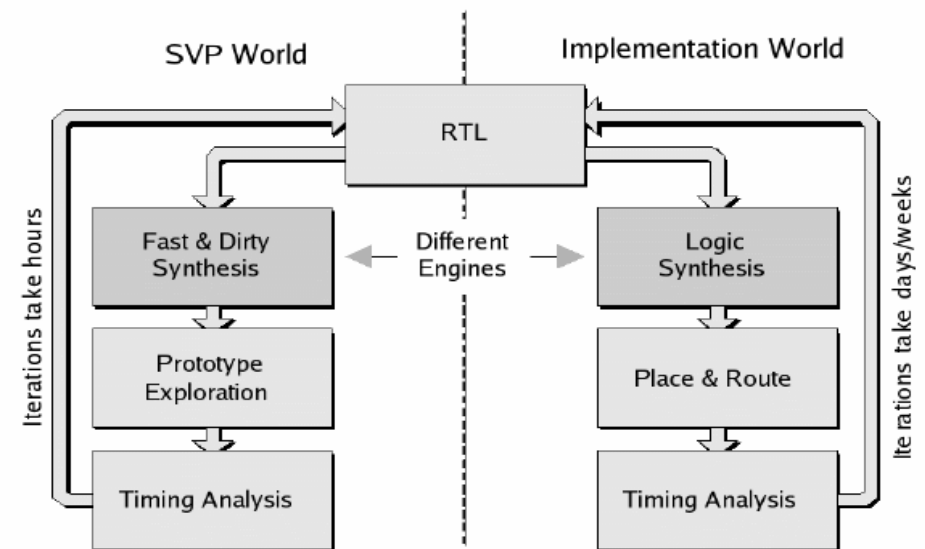
3. FPGA Prototyping



→ Verkürzung der „time to market“ um 3 bis 6 oder mehr Monate

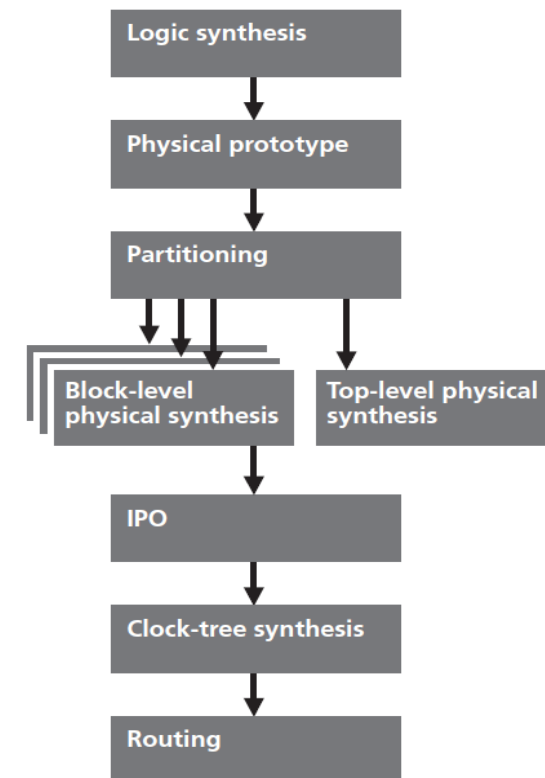
4. Silicon Virtual Prototyping

- Designergebnisse und Probleme sind erst nach einer genauen Timinganalyse ersichtlich
 - Dazu muss der gesamte Designflow (Synthese, Place&Route) durchlaufen werden
 - Dauer einer Iteration dauert mehr als einen Tag (bei mehreren 10 Mio. Gattern)
- Prototyp, der schnell erzeugt werden kann



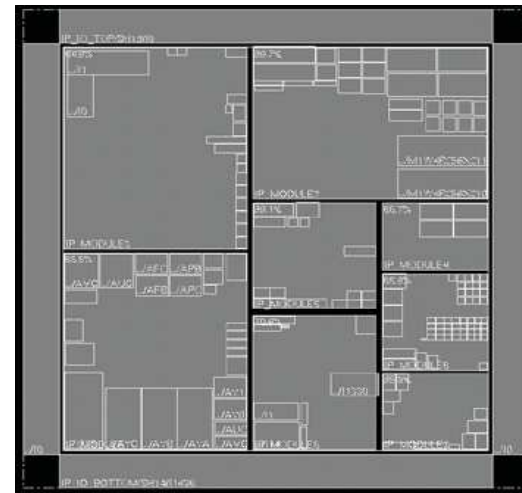
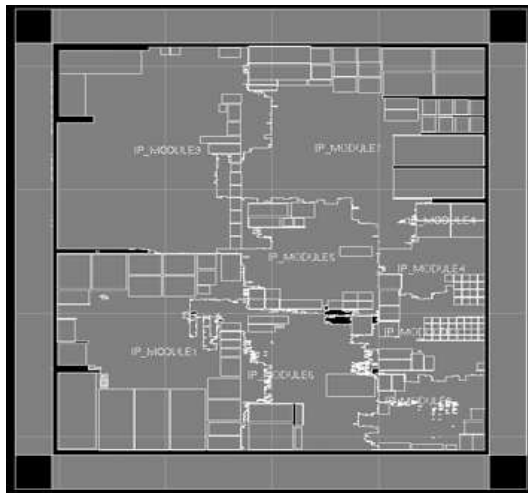
4. Silicon Virtual Prototyping

- **Logiksynthese:** einfache Leitungsmodelle
- **Floorplanning:** automatische Anordnung von I/O ports u. Blöcken, Aufteilung in kleine verarbeitbare Blöcke
- **Physical synthesis:** platzieren der Standardzellen mittels „timing-driven“ Algorithmen, RC-Extraktion u. Timing Analyse
- **In-Place Optimization:** Behebung von Timingproblemen



4. Silicon Virtual Prototyping

- Vergabe von Timing- und Flächenbudget für jedes Makro
- Iterative Verfeinerung mit fortschreitender Implementierung



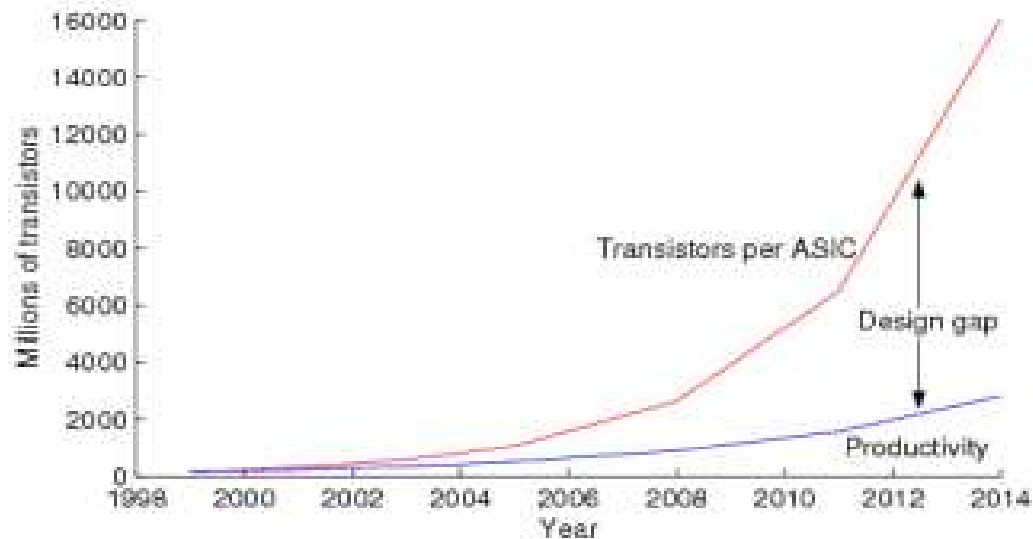
4. Silicon Virtual Prototyping

Design step	Run time
Verilog netlist import	2 minutes
Full-chip flat placement	54 minutes
Full-chip flat trial route	3 minutes
Full-chip RC extraction	<1 minute
Full-chip SDF output	4 minutes
Full-chip timing analysis	5 minutes
Fixing transition violations	10 minutes for 18,352 max_tran violations Optimization summary: • 528 nets fanout optimized • 964 buffers inserted
Clock-tree synthesis	15 minutes reaching an approximately 400 picosecond skew

Bsp.: Prozessorkern mit ca. 100.000 platzierbaren Zellen

Ca. 14 - 40 h mit Standardtools

5. Zusammenfassung



Schließen der Designlücke durch:

- Hoher Grad der Wiederverwendung von Hard- und Software
- Früherer Start der Softwareentwicklung dank FPGA - Prototypen
- Schnell erzeugbarer virtueller physikalischer Prototyp zur Analyse von Timing, Fläche und Verlustleistung

Quellen

Abbildungen:

[s. 4,10,12,14] http://www.fpgajournal.com/articles/soc_aptix.htm#top

[s. 8] <http://en.wikipedia.org/wiki/System-on-a-chip>

[s. 9,11] Michael Keating u. Pierre Bricaud, Reuse methodology manual: for system-on-a-chip designs, 2007, ISBN 978-0-387-74098-0

[s. 15] <http://www.cs.umbc.edu/~plusquel/415/slides/SVP.pdf>

[s. 16,17,18,21] <http://www.cadence.com/pages/pagenotfound.aspx>

[s. 19] <http://cas.ee.ic.ac.uk/people/nps/research.html>

Zusätzlich:

[1] Michael Keating, Low Power Methodology Manual: System-on-ChipDesign, August 2007, ISBN 978-0-387-71818-7

[2] Sachin S. Sapatnekar ; Prashant Saxena ; Rupesh S. Shelar, Routing congestion in VLSI circuits: estimation and optimization, 2007, ISBN 0-387-30037-6

[3] http://www.vlsichipdesign.com/signal_integrity.html

[4] http://www.lecroy.com/tm/Library/LABs/PDF/LAB_WM729.pdf

[5] <http://www.vlsichipdesign.com/ocv.html>

[6] Annette Reutter: Rechnergestützte Wiederverwendung digitaler Schaltungsmodule, 1999, ISBN 3-933346-83-5