

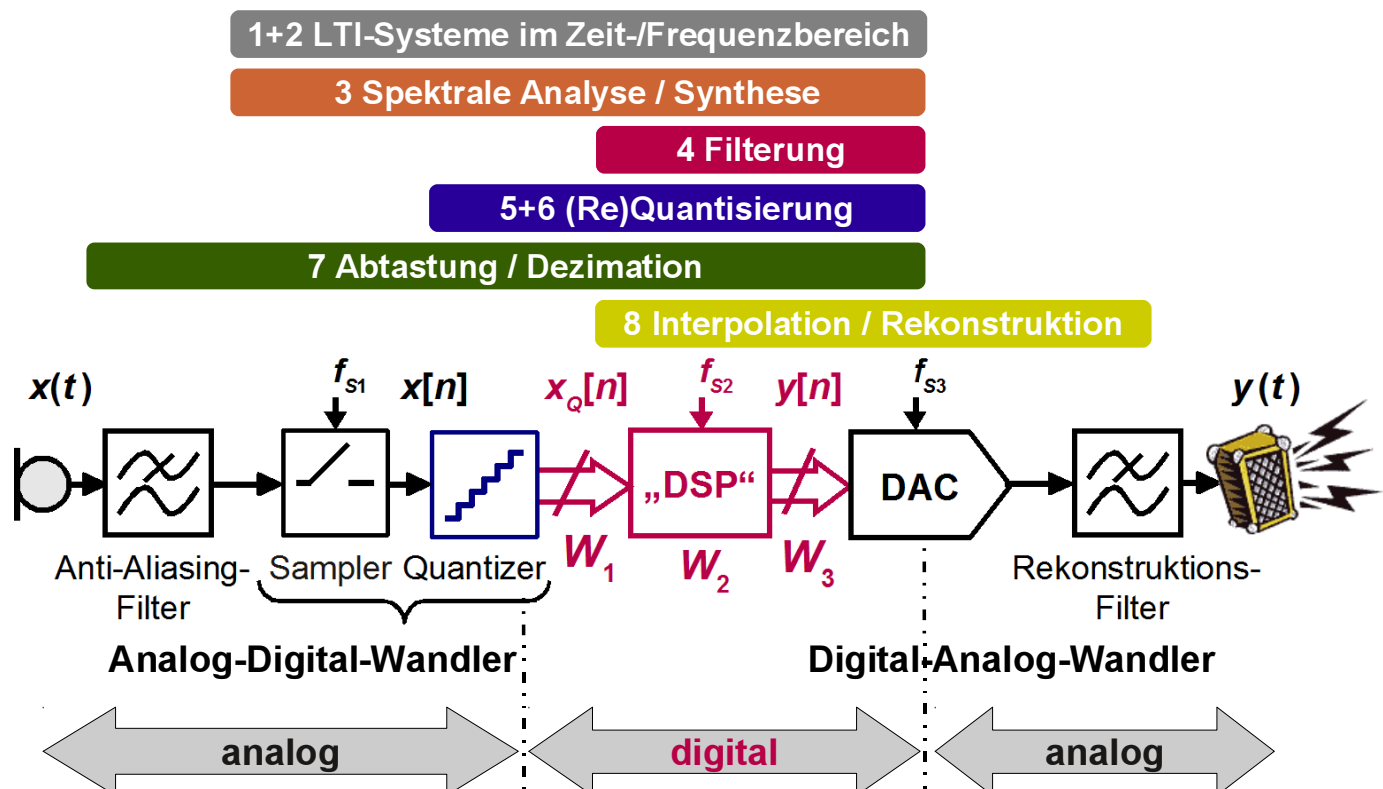
# Digitale Signalverarbeitung auf FPGAs

## Kap. 0 – Einführung

2015

Prof. Dr. Christian Münker

## Ein allgemeines DSP – System





1. Grundlagen: Zeitdiskrete Signale und Systeme im Zeitbereich
2. Grundlagen: Zeitdiskrete Signale und Systeme im Frequenzbereich
3. Diskrete Fourier Transformation und FFT
4. Digitale Filter (FIR und IIR)
5. Quantisierung und Arithmetik (Zeitbereich)
6. Quantisierungsrauschen (Frequenzbereich)
7. Abtastung und Downsampling
8. Upsampling und Digital-Analog Wandlung
9. Multiraten-Signalverarbeitung
10. Zustandsraum
11. Wavelets und Filterbänke

## Organisatorisches (1)



<b>eMail</b>	<a href="mailto:christian.muenker@hm.edu">christian.muenker@hm.edu</a>
<b>Büro</b>	R3.063, Tel. 3466 (3. Stock über der Fachschaft)
<b>Sprechstunde</b>	Mi. 9:00 – 10:00 ... oder nach Vereinbarung (eMail)
<b>Videos</b>	Kap. 1 - 3 zu Hause, Übungen an der HM
<b>Download</b>	von Vorlesungsunterlagen und Videos über Moodle-Seite - Password
<b>Vorlesung</b>	Fr., 10:00 – 13:15 Uhr / 1x Di. 14:15 – 17:30
	<b>Letzte reguläre Vorlesung: 4.12.2015</b>

# Organisatorisches (2)



- Python Crashkurs: Di. 13.10. 14:15 – 17:30 in RBG 052
- Praktikum (freiwillig), Fr. 10:00 – 13:15 + Di. 14:15 – 17:30 (2 Grp)
  - 1. **Versuch:** Grundlagen Simulink / System Generator 08. / 11.12.
  - 2. **Versuch:** Entwurf eines Multiratenfilters in Simulink 15. / 18.12.
  - 3. **Versuch:** Implementierung des Filters auf einem FPGA mit System Generator 8. / 12. 1. 2016
- Prüfungsvorbereitung **15.1.2016**
- Prüfung besteht aus **einem Teil** (90 min.)  
**Hilfsmittel : 4 DIN-A4 Seiten** (2 Blätter) Unterlagen

## Literatur



Grundlage fur „DSV auf FPGAs“ ist der Stoff der Vorlesung „Signale und Systeme“ (Fourier- und Laplace-Transformation, zeitdiskrete Signale und Systeme)

Uber meine Moodle-Seite konnen Skript und Ubungsaufgaben von Prof. Geng heruntergeladen werden:

### **Signale und Systeme Teil 2: Zeitdiskrete Signale und Systeme**

**Diese Inhalte werden als bekannt vorausgesetzt!**

Weitere (optionale) Literatur: Siehe Unterlagen zu „DSV auf FPGAs“



- Laut Modulbeschreibung hat *DSV auf FPGAs* 5 ECTS-Punkte und 4 SWS
- Ein ECTS-Punkt entspricht laut Vorgaben der internationalen Studien-Kommissionen ca. 30h Arbeitsaufwand → 150h für *DSV auf FPGAs*
  - 33h (Zeitstunden) seminaristischer Unterricht
  - 9h Praktikum
  - **ca. 100h Eigenstudium** (Vor- und Nachbearbeitung, Prüfungsvorbereitung etc.)
- „Early Warning“: 30 ECTS-Punkte pro Semester entsprechen damit 900h, bei 15 Wochen im Semester haben Sie also theoretisch eine 60h-Woche ...
- Daher:
  - Arbeiten Sie in den Vorlesungen mit
  - Lassen Sie sich nicht abhängen – bereiten Sie den Stoff vor / nach

## Was ist ein FPGA?



(Das wird nicht in der Prüfung abgefragt ...)

- **Geschichte: Was ist ein FPGA (nicht)?**
- Technologie
- Architektur
- Entwurf

Siehe auch Webinar:

Max Maxfield,  
„Fundamentals of FPGAs“ [Xilinx]

[www.techonline.com/learning/course/210605004](http://www.techonline.com/learning/course/210605004)



## Universal – Prozessoren (Intel, AMD, Microcontroller, ...)

- Preisgünstig, begrenzte Performance, nicht energieeffizient für DSP

## DSP - Prozessoren

- Hardware und Befehlsset optimiert für DSP (z.B. schnelle Multiplizierer)

## FPGAs

- Sehr hohe Performance möglich durch Parallelisierung
- Hohe Effizienz durch beliebige Wortlängen
- „Software-frei“ (schwer zu kopieren und hacken, definiertes Einschaltverhalten)

## ASICs

- Noch höhere Effizienz und Performance als FPGAs, aber kostengünstig nur bei hohen Stückzahlen (> 100.000; für Multimedia, Mobilfunk, ...)

## Vergleich: ASIC, FPGA, Microcontroller



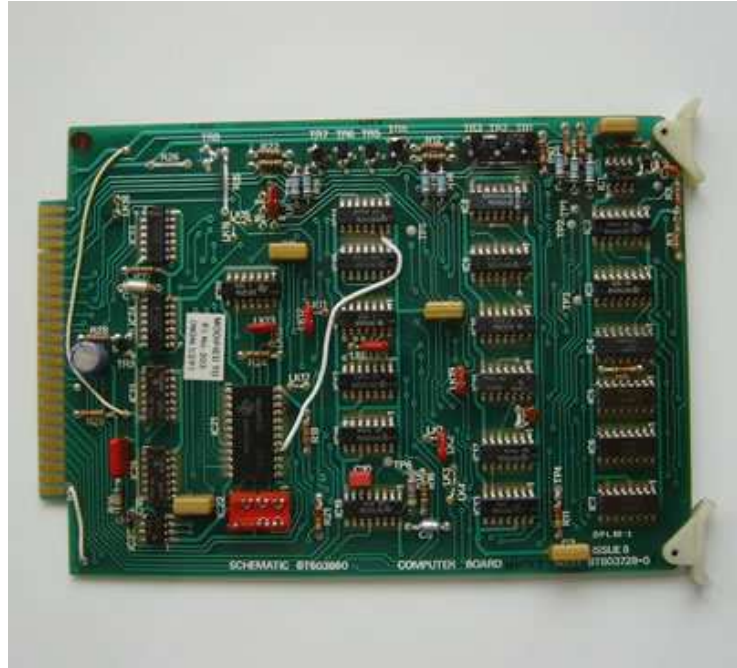
← **\$\$\$**, Flexibilität, Rechenleistung

	ASIC	FPGA	Microcontroller
Maskenkosten	10k..1M \$	Entfällt	Entfällt
Entwurfssoftware	> 20 k\$	Kostenlos ... 1000 \$	Kostenlos ... 1000 \$
Entwicklungszeit	Hoch (Gates, VHDL, IPBlocks, Analog)	Mittel ... schnell Schematic, VHDL, IP	Schnell Assembler, C, IP
Stückpreis	\$25000 pro 25 Wafer mit jeweils 25k Chips	< 1 \$ ... 5000 \$	0.1 \$ ... 100 \$ (?)
Lieferzeit	Mehrere Monate	i.A. schnell verfügbar	i.A. schnell verfügbar
Test- und Entwicklungsboard	Spezielle Entwicklung	50 \$ ... 1 k\$, Standardprodukt	1 \$ ... 1 k\$, Standardprodukt
Effizienz (Power und Fläche pro Funktion)	Optimal, Wortlänge / Clock / Technologie (frei) wählbar	Langsamer Clock, aber parallelisierbar, Wortlänge beliebig	Langsamer Clock, nur serielle Datenverarbeitung, analoge Functions

„TTL-Friedhöfe“ bestehen aus zahllosen Logik-Chips

- unzuverlässig
- schwer zu modifizieren
- groß
- stromhungrig
- Logistik-Albtraum  
(viele Spezial-Bausteine)

→ **Programmable Logic  
Devices (PLDs) !**



[siehe „programmable logic“ in Wikipedia.org !]

## Simple PLDs (SPLDs) - PROMs



**(P)ROM** kann beliebige kombinatorische  
Logik ersetzen:

- Adressdekoeder entspricht  
fester AND-Verknüpfung
- Speicherinhalt entspricht  
OR-Verknüpfung
- Andere Bezeichnung:  
„**Look-Up Table**“ (LUT)
- Nur wenige Eingänge möglich,  
da Dekoder ansonsten sehr  
komplex wird

PROM bereits 1957 patentiert  
(US-Pat. 3028659)

**Adressdekoeder  
(festes AND – Array)**

**Programmierbares OR – Array**



Mitte der 1980er Lücke zwischen PLDs und ASICs mit hohen NRE \* - Kosten, speziell für schnelle Datenverarbeitung

\* Non-Recurring Engineering / Expenditure

## Auftritt der FPGAs

---



1985 erstes FPGA XC2064 von Xilinx mit 64 Logic Blocks (jeweils zwei 3-Input LUTs und einem FF), 58 I/O – Pins und 1200 equiv. Gates



- **Bildverarbeitung** (Driver Assistance Systeme, medizinische Bildgebung) → *Rechenleistung, Parallelisierung*
  - **Schnelle Regelungen** (Vektor- und observerbasierte Motorsteuerungen) → *geringe, definierte Latenzzeit*
  - **Kommunikationssysteme** (Pay-TV, 3D-TV, Smartphones und Basisstationen) → *Rechenleistung, Rekonfigurierbarkeit*
  - **Messsysteme** (NI / Labview FPGA, Spektrumanalyzer) → *schnelle Real-Time Signalverarbeitung, geringe Stückzahlen*
  - **High-Speed Datenübertragung** → *Gb/s Serializer / Deserializer*
- Generell:** Prototyping, Stückzahlen < 100.000, „Softwarelose“ Systeme, schnelle parallele Datenverarbeitung

## Was ist ein FPGA?



- Geschichte: Was ist ein FPGA (nicht)?
- **Technologie**
- Architektur
- Entwurf



# Technologie: SRAM



Standard CMOS-Prozess, daher immer in aktuellster Technologie verfügbar → schnell und günstig (in €/Gate)!

- Konfiguration durch SRAM-Zellen (viel höherer Stromverbrauch)
- Verwendet für FPGAs
- ± Verliert Inhalt beim Abschalten
- Ext. Flash oder CPU benötigt

## Marktführer:

Xilinx (Spartan, Virtex)

Altera (Stratix, Cyclone)

Optional: Flash im gleichen Gehäuse (SIP)

(Xilinx Spartan-3AN, Lattice XP)

# Technologie: EEPROM / Flash



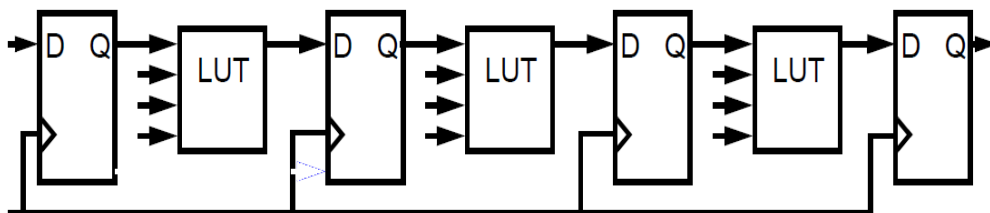
Zusätzliche Technologieschritte für nicht-flüchtige Speicherzellen

- Technologieknoten werden ca. 18 Monate nach Standard-CMOS erreicht
- Verwendet für SPLDs, CPLDs, FPGAs (z.B. Actel Igloo and ProASIC3)
- Elektrisch programmier- und löschar
- Synonym: „Non-Volatile-...“
- + In-System **Re**-programmierbar
- + Beim Einschalten funktionsfähig
- Langsamer + teurer als SRAM
- Unterschied zwischen EEPROM und Flash: nur in der Architektur (NOR vs. NAND), nicht in der Technologie



- Geschichte: Was ist ein FPGA (nicht)?
- Technologie
- **Architektur**
- Entwurf

## FPGA = Synchrone Logik



„kritischer Pfad“ begrenzt max. Taktrate



- Dedizierte Clock Pins zum Anschluss eines Oszillators mit  $f_{ref}$
- PLLs erzeugen davon abgeleitete Clock Signale  $f = M / D f_{ref}$  mit einstellbaren Phasen (z.B. Virtex-5:  $f \leq 550$  MHz,  $M \leq 33$ ,  $D \leq 32$ )
- Separate gepufferte („lokale“) Clock-Netze zum Ausgleich von Laufzeiten und Lasten an den Clock Pfaden

## FPGA-Makros: Block RAM



- Einzelne Flip-Flops sind ineffizient für größere Speicherblöcke
- ➔ Embedded SRAM-Blöcke im Fabric verteilt (insgesamt bis zu mehreren MB) „Block RAM“ oder BRAM



- Über 1000 I/O-Pins pro FPGA möglich mit verschiedensten Standards
- Digitally-Controlled Impedance (DCI), on-chip active I/O termination
- Delay-Equalization in 75 ps – Schritten (DLL) für z.B. parallele Busse

## Weitere FPGA-Makros



- Komplexe 32 bit uC-Cores (z.B. von ARM) als Hard-Makros
- Kleinere uPs werden oft als Soft-Makro implementiert (Xilinx: Picoblaze, Microblaze)
- Serializer / Deserializer (Hard-Makro) für bis zu 25 GB/s pro differenziellem I/O-Kanal
- Unzählige Softmakros kommerziell oder als OpenCore verfügbar
- Komponenten werden oft verbunden über ein Bussystem mit **ARM Advanced Microcontroller Bus Architecture (AMBA)**
  - **Advanced eXtensible Interface Bus (AXI)**
  - **Advanced Peripheral Bus (APB)**
  - ...



- Geschichte: Was ist ein FPGA (nicht)?
- Technologie
- Architektur
- **Entwurf**

## Design Software



### VHDL / Verilog

Erlaubt Entwurf auch sehr komplexer Systeme – aber viel Erfahrung nötig!

### Blocksets (hardwarespezifisch)

Vom FPGA-Hersteller zur Verfügung gestellte Blöcke erlauben grafischen Entwurf, eher für geringere Komplexität geeignet

### Matlab / Simulink (Mathworks)

Quasi-Standard für System Level Simulationen / Design, sollte begleitend zum Kurs installiert werden. Zusätzliche Tools bzw. -boxen ermöglichen automatisierte Hardwaresynthese aus Simulinkmodellen:

**Simulink HDL Coder:** Toolbox, generiert synthetisierbaren VHDL oder Verilog-Code aus Simulink / Stateflow – Designs (unabhängig von Zielplattform)

**Xilinx System Generator:** Hardwarespezifische Blocksets für Simulink, die synthetisierbaren HDL-Code für die entsprechende Plattform erzeugen

# 4-Bit Addierer in VHDL



```
entity Adder_VHDL is
  port(
    a, b: in std_logic_vector (3 downto 0);
    sum: out std_logic_vector (3 downto 0);
    cout : out std_logic
  );
end entity Adder_VHDL;

architecture Behavioral of Adder_VHDL is
  signal sum_sig: std_logic_vector (4 downto 0);
begin
    sum_sig <= ("0" & a) + b;
    sum <= sum_sig (3 downto 0);
    cout <= sum_sig (4);
end architecture Behavioral;
```

2010

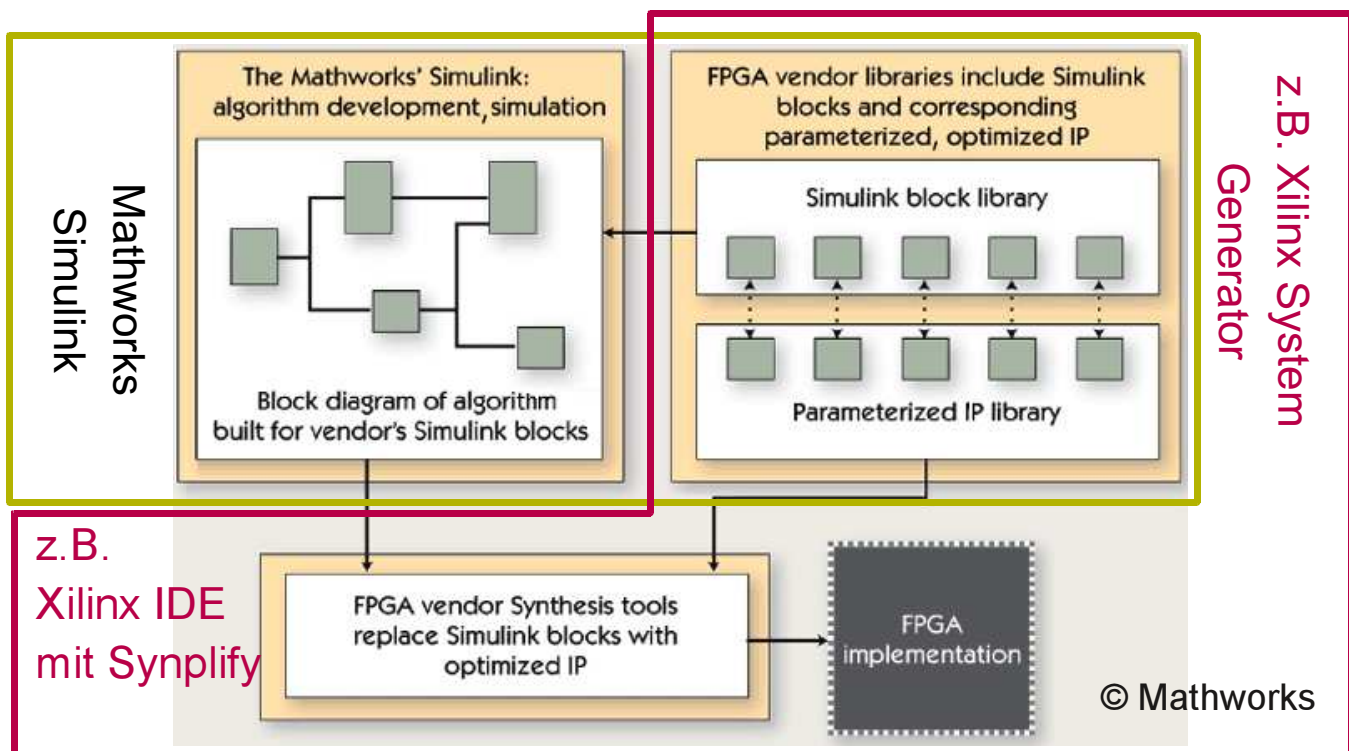
Dr.-Ing. Jörg Vollrath 27

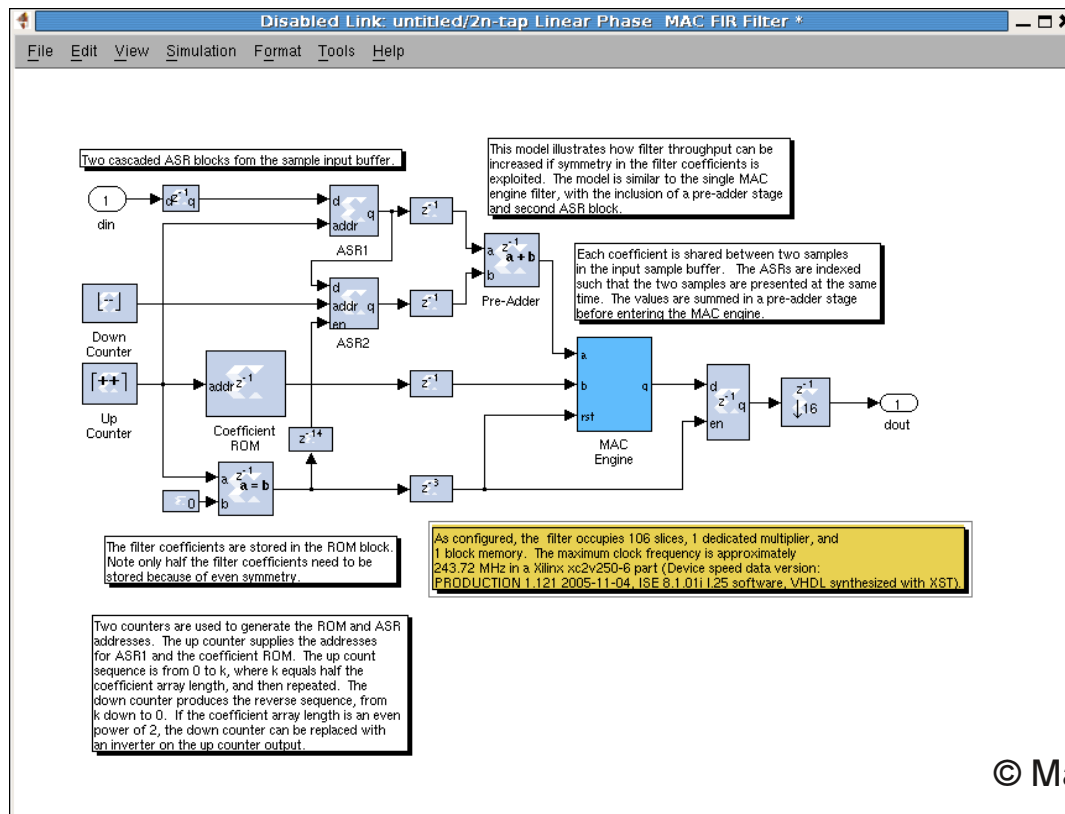
Prof. Dr. Christian Munker

Oct 2, 2015

Digitale Signalverarbeitung auf FPGAs 0-27

## Simulink / System Generator





## Die Zukunft ist da: System-On-Chip



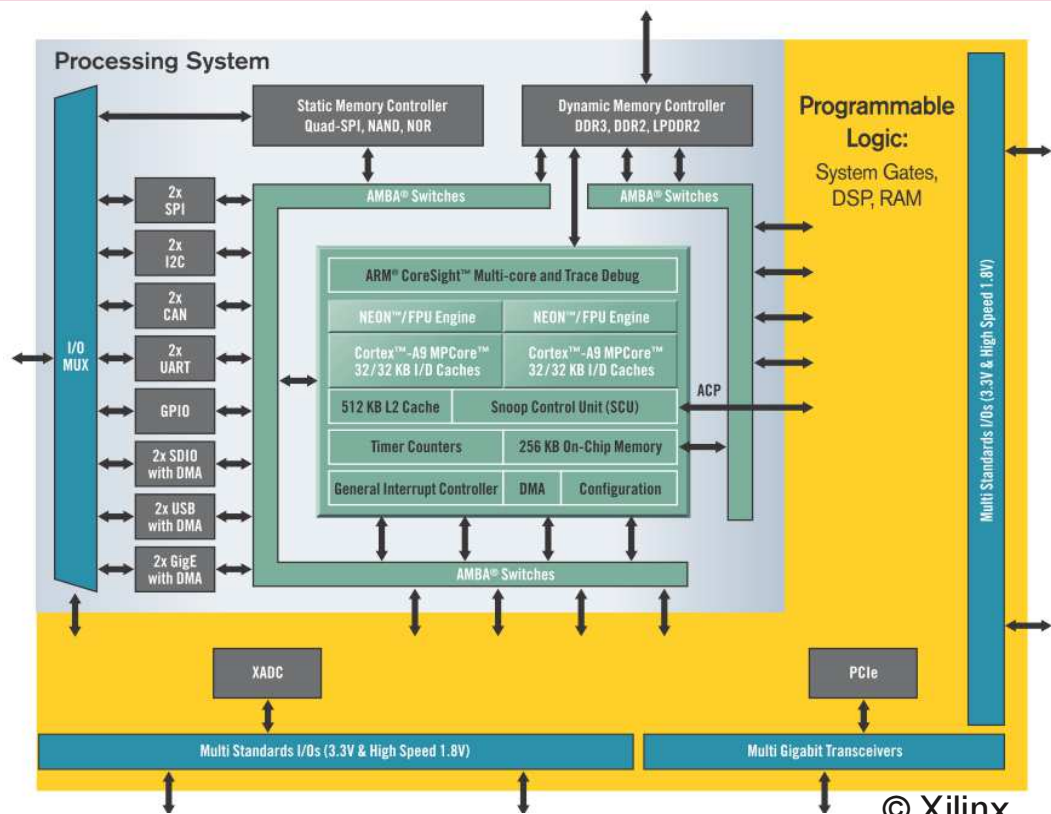
**Xilinx ZYNQ-7000**  
System-On-Chip,  
2x ARM Cortex A9  
Processor 800 MHz  
10 ... 20 €

### Vergleichbar:

- Actel Fusion,
- Cypress PSoC,
- Altera SoC

...

**Design  
Prozess ???**



Produktion von kundenspezifischen ICs wird mit jedem Technologieknoten exponentiell teurer

**Sind FPGAs die Kakerlaken, die übrig bleiben, wenn sich der Staub von Moore's Apokalypse gelegt hat?**

## Atlys FPGA-Boards (SPARTAN-6)



- 15 Stück (180 € ) für Praktikum DSV auf FPGAs und fürs Labor bestellt
- Simulink / System Generator Support (Lizenzen?)
- Hardware-in-the-Loop
- HDMI, Audio Codec, USB, GB Ethernet, ...
- PMOD-Interface



© Digilent





- Simulation ist zentraler Teil der Systementwicklung (und der Lehre) von DSV Systemen
- Quasi-Monopol von Mathworks mit Matlab / Simulink
- Für Hochschulen vergleichsweise unfreundliche Lizenzpolitik

**Fragestellung:** Gibt es Alternativen, die ...

- DSV-Simulationen mit Wortlängeneffekten
  - automatisierte Generierung von FPGA / DSP – Code
  - Kommunikation mit Messgeräten, HW / SW-Cosimulation, ...
- ... für Studierende, Professor und Laborprojekte erlauben?