

# Mikrocomputer – Stand der Technik

## BMK Elektronikforum 2010

Hubert Högl

Hubert.Hoegl@hs-augsburg.de

<http://www.hs-augsburg.de/~hhoegl>

Hochschule Augsburg

17. Juni 2010

## Fakten zur Technischen Informatik (TI)

- Beginn 2006
- Mischung aus Informatik
  - ▶ Programmierung
  - ▶ Software Engineering
  - ▶ Embedded Systems
  - ▶ Viele Wahlfächer

## und Elektrotechnik

- ▶ Grundlagen E-Technik
- ▶ Bauelemente, Schaltungstechnik
- ▶ Messtechnik

Absolventen sollen sowohl die Hardware moderner eingebetteter Systeme verstehen als auch alle Aspekte der Programmierung der darauf laufenden hochkomplexe Software kennen.

- Erste Bachelor-Absolventen jetzt fertig
- Kollege Prof. Gundolf Kiefer: Rechnen mit rekonfigurierbarer Logik, Entwurf digitaler Systeme, VHDL, Soft-Cores, ...

# Meine Arbeitsgebiete

- Systemnahe Programmierung
  - *Assembler, C, Python*
- Embedded Systems (ehem. Rechnertechnik)
  - 8-Bit (AVR) bis 32-Bit (AVR32, ARM7, Cortex M3)
  - Interface-Techniken, USB, low-power Geräte
  - Programmierung mit FreeRTOS
  - ARM Debugger OpenOCD
- Embedded Linux
  - AVR32, ARM9, PowerPC, x86
  - Buildroot, OpenWRT, OpenEmbedded, Echtzeit-Linux
- Open-Source
  - Vorlesung *Open-Source Software*
  - O.-S. als generelles Arbeitsmotto, z.B. freie Diplomarbeiten:  
<http://elk.informatik.fh-augsburg.de/da>

# Erster Eindruck

- Unüberschaubar schnelle Entwicklung des Mikrocontroller-Marktes
- Immer mehr Firmen und Produkte
- Nicht nur *ein* neuer Controller, sondern bis zu 10, 20 Varianten werden auf einmal geschaffen.
- Energiebewusstsein in allen Leistungsklassen
- Fast nur noch System-on-Chip, weniger Prozessor + Chipsatz
- ARM überall
- Intel spielt nun mit
- Immer mehr Geräte laufen mit Linux
- Boards und Bausteine in den letzten Jahren stark im Preis gefallen (z.B. <http://www.olimex.com>, <http://www.propox.com>).

# Der ideale Prozessor/Controller

## Technische Faktoren

- Ausreichende Rechenleistung (DMIPS  $\uparrow$ )
- Hohe Recheneffizienz (DMIPS/MHz  $\uparrow$ )
- Geringe dynamische Stromaufnahme (mW/DMIPS  $\downarrow$ )
- Preiswert (Euro  $\downarrow$ )
- Geringe statische Leistungsaufnahme ( $I_{stop}$   $\downarrow$ )
- Es gibt Peripherie, die zur Anwendung passt

## Weiche Faktoren

- Lieferbarkeit, langfristige Verfügbarkeit
- Antwortet der Hersteller bei Problemen?
- Wie lang ist die Fehlerliste?
- Wechselmöglichkeit auf andere Controller (vermeidet *lock-in*)
- Gibt es Software vom Hersteller?
- Gibt es Werkzeuge und Software von mehreren Anbietern?
- Gibt es die GNU Compiler Collection GCC?
- Gibt es freie Software dafür, z.B. FreeRTOS, Linux, ... ?

## Hardware-Kategorien mit Anwendungen

**K1** Kleine Controller mit 1-8K Flash, bis 1K SRAM, bis einige MHz, geringe Pin-Zahl, stromsparend, preiswert, oft 8-Bit und 16-Bit, kleiner 1 DMIPS. Meist ohne Betriebssystem, C Hauptprogramm mit ISRs. Vertreter: kleine AVR, MPS430, PIC, 8051, und andere. Zunehmend low-cost 32-Bit wie Cortex M0.

*simple Sensoren/Aktoren ansteuern, einfache I/O Erweiterungen, Auto (LIN)*

**K2** Ähnlich K1, jedoch mehr Flash und RAM Speicher (bis 128K), umfangreichere Peripherie (z.B. CAN), mehr Pins, Programmierung ohne OS oder kleines RTOS, oft 8-Bit und 16-Bit, kleiner 5 DMIPS. Beispiele: Atmel AVR Mega128, zunehmend 32-Bit SoC wie AVR32UC, Cortex M0/M3.

*Funknetze, RFID, umfangreichere Protokolle, RTOS, CAN, USB, bis zu zig-1000 Zeilen C Code*

## Hardware-Kategorien mit Anwendungen (2)

**K3** Hauptsächlich 32-Bit, 16 bis 60 MHz, 4-128K SRAM, 128K-512K Flash, preiswert, kleiner 40 DMIPS, Programmierung mit kleinem (FreeRTOS) oder mittlerem RTOS (eCos). Vertreter: Alle Arten von SoC mit ARM7 und Cortex M0/M3.

*Signalverarbeitung, MP3-Dekodierung in Software (ca. 40 MIPS), zig-10.000 Zeilen C Code, RTOS, schnelle Steuerungen, BLDC Motoren, Ethernet und grundlegende Protokolle, USB, CAN*

**K4** Immer 32-Bit, 60 MHz bis 1 GHz, externen Speicher (SDRAM 16-128M, Flash 1M-64M), 80 bis 2000 DMIPS, grosse Betriebssysteme wie Windows oder Linux. Vertreter: ARM9, ARM11, Cortex A8, A9, PowerPC, MIPS, etc.

*Grosse Betriebssysteme, TFT Displays, komplexe Software ähnlich Desktop, high-end Smartphones, Video-Streaming MPEG4, Batteriebetrieb, ...*

# Wichtige Masszahlen

- Maximale Frequenz
  - Höhe und Anzahl der Versorgungsspannungen
  - MIPS (keine Aussagekraft)
  - DMIPS = „Dhrystone MIPS“ (Benchmark)
  - DMIPS/MHz (Ausführungsleistung)
  - mW/MHz, mA/MHz (geringe Aussagekraft)
  - DMIPS/W, DMIPS/mW (Leistungseffizienz)
  - W/DMIPS, mW/DMIPS (Leistungseffizienz)
  - $I_{stop}$  Leckstrom im Schlafmodus
  - Integrationsdichte des Flash- und RAM-Speichers
  - Codeeffizienz (geringer Speicherbedarf)
  - Gehäusetyp
- Achtung: Unterscheide bei DMIPS/mW Core und SoC!

# Physikalische Zusammenhänge

**Verlustleistung**  $P_{switch} + P_{short} + P_{leak} + P_{static}$

Dynamisch:  $P_{switch}$ ,  $P_{short}$ , statisch:  $P_{leak}$ ,  $P_{static}$

$$P_{switch} = A \times C_L \times U^2 \times f$$

$$P_{short} = (U_{DD} - 2 \times U_t)^3$$

$P_{leak}$  : steigt mit Faktor 5 bei jeder Generation, exponentiell von Temperatur abhängig: (1) gate leakage, (2) subthreshold leakage.

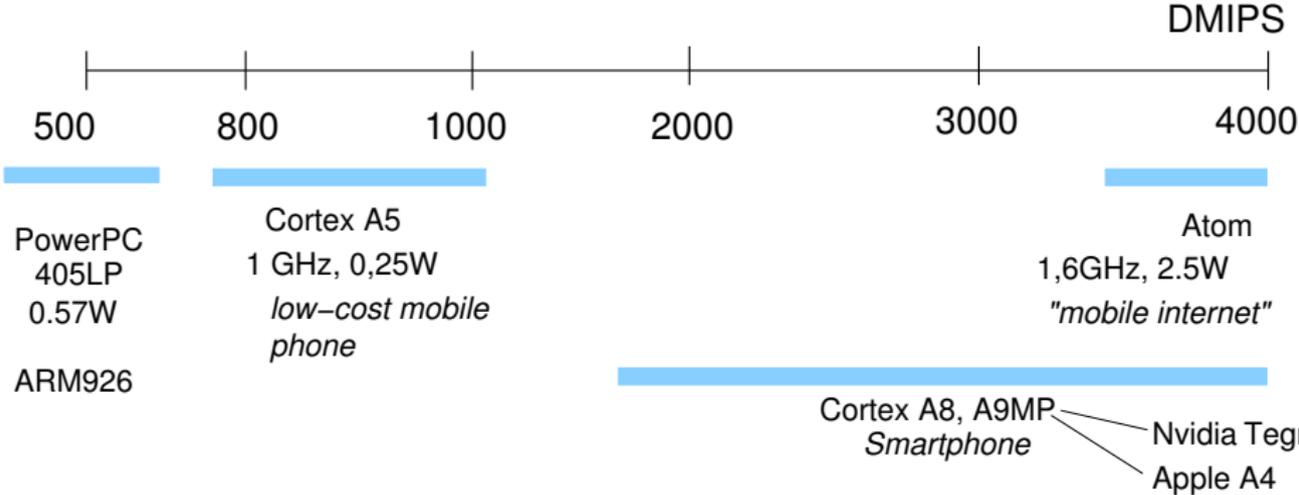
$P_{static}$  : fast keine statischen Pfade (sub- $\mu$ Watt)

Verkleinerung der Strukturen:

↪ niedrigere dynamische Leistungsaufnahme

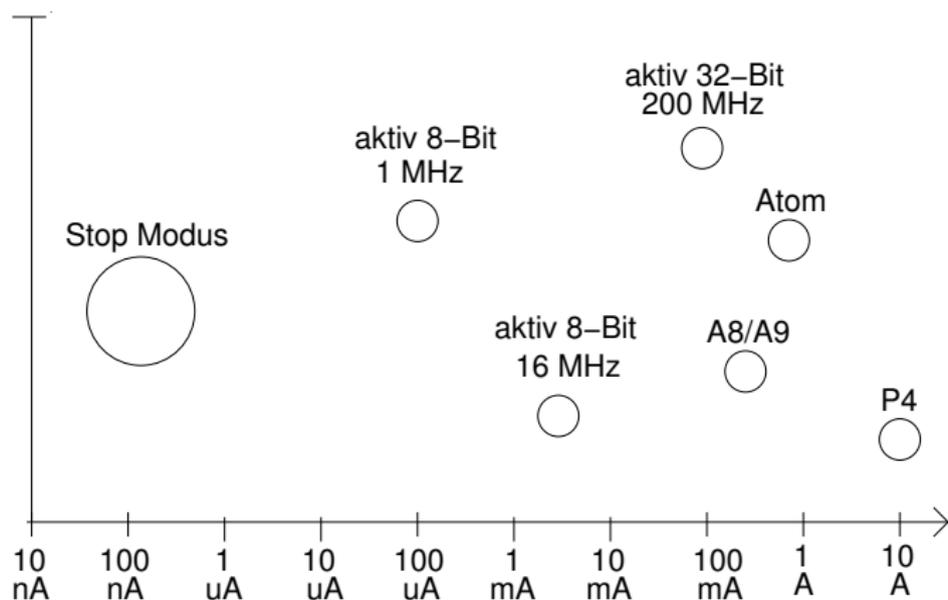
↪ höhere Leckströme

# DMIPS-Skala



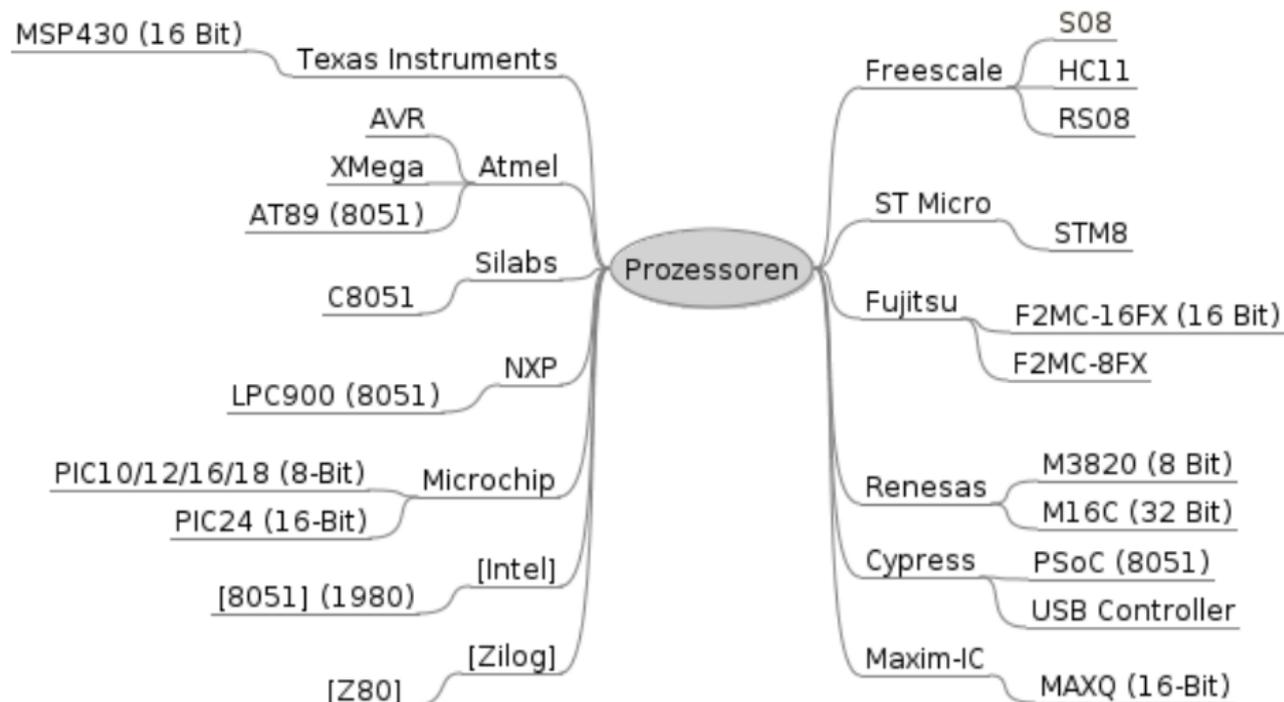
# Leistungs-Skala

Strom bzw. Leistung geht über 10 Zehnerpotenzen!



# Prozessorhersteller 8- und 16-Bit

Einige wichtige Hersteller von 8-/16-Bit Mikrocontrollern (nicht vollständig).

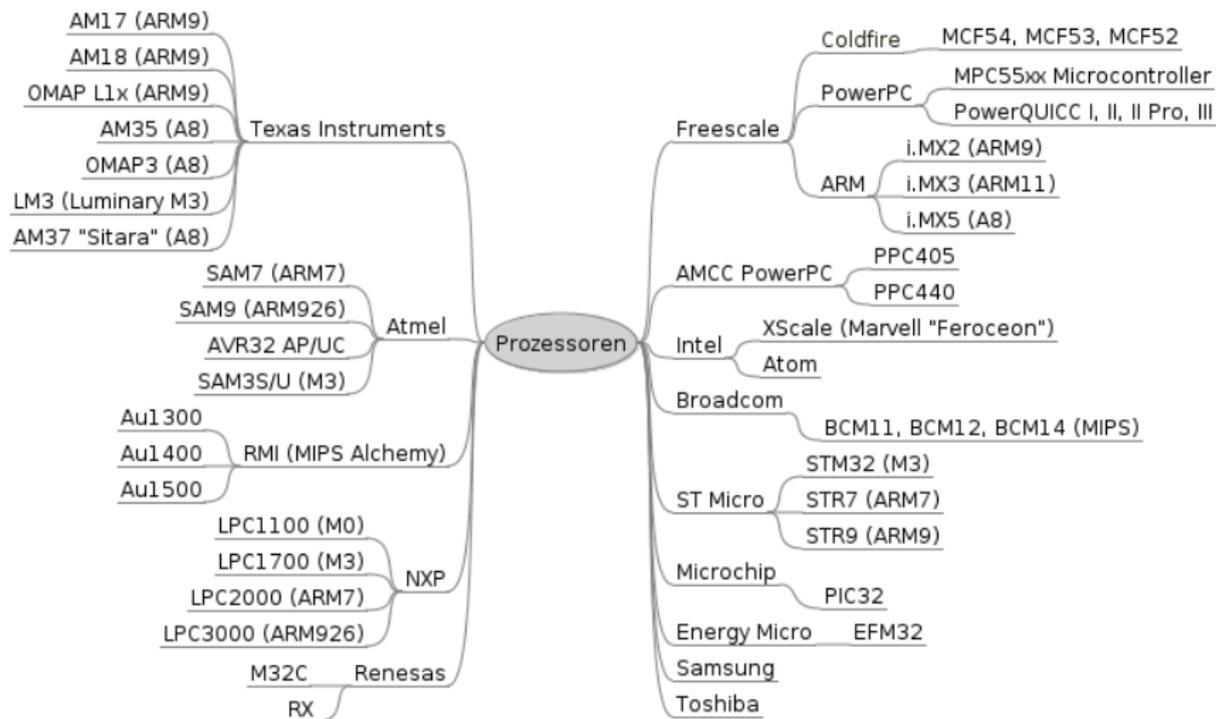


## Diskussion 8-Bit

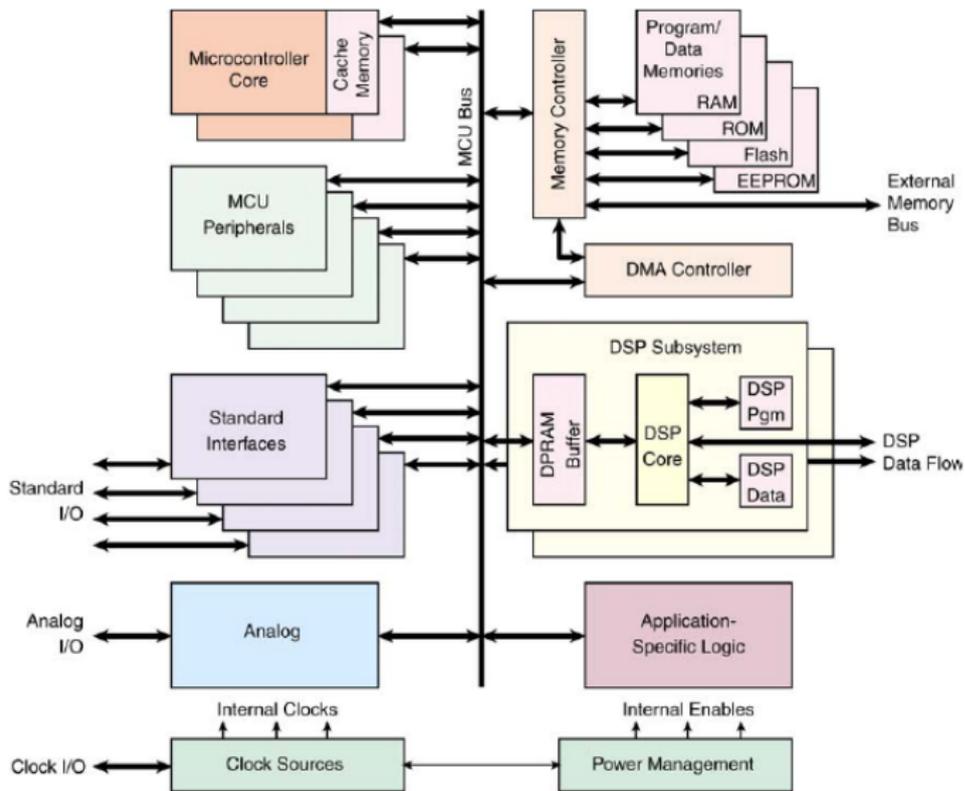
- Immer noch die geringste Ruhestromaufnahme  
↪ RFID, Funkmodule, Medizintechnik, Energy Harvesting, ...  
32-Bit Controller kommen aber immer näher.
- 8051 lebt immer weiter mit sehr guten technische Eigenschaften, z.B. 160  $\mu\text{A}/\text{MHz}$ , schlafen mit 25 nA (Silabs, Atmel, NXP, Cypress und andere).
- Druck von beiden Seiten: 8-Bit  $\rightarrow$  **16-Bit**  $\leftarrow$  32-Bit.
- Praktisch kein Preisunterschied zu bestimmten 32-Bit Controllern mit Faktor **40** mehr MIPS (ARM7, Cortex M0/M3).
- Neue Ideen
  - ▶ Integrierte Spannungsregler
  - ▶ Integrierter Step-Up/Step-Down Konverter (z.B. ATtiny43U)
  - ▶ Integrierter HF Transceiver (Silabs)
  - ▶ DMA zwischen Speicher und Peripherie (XMega)
  - ▶ Ein-Draht Debug

# Prozessorhersteller 32-Bit

Einige wichtige Hersteller von 32-Bit Mikrocontrollern (nicht vollständig).



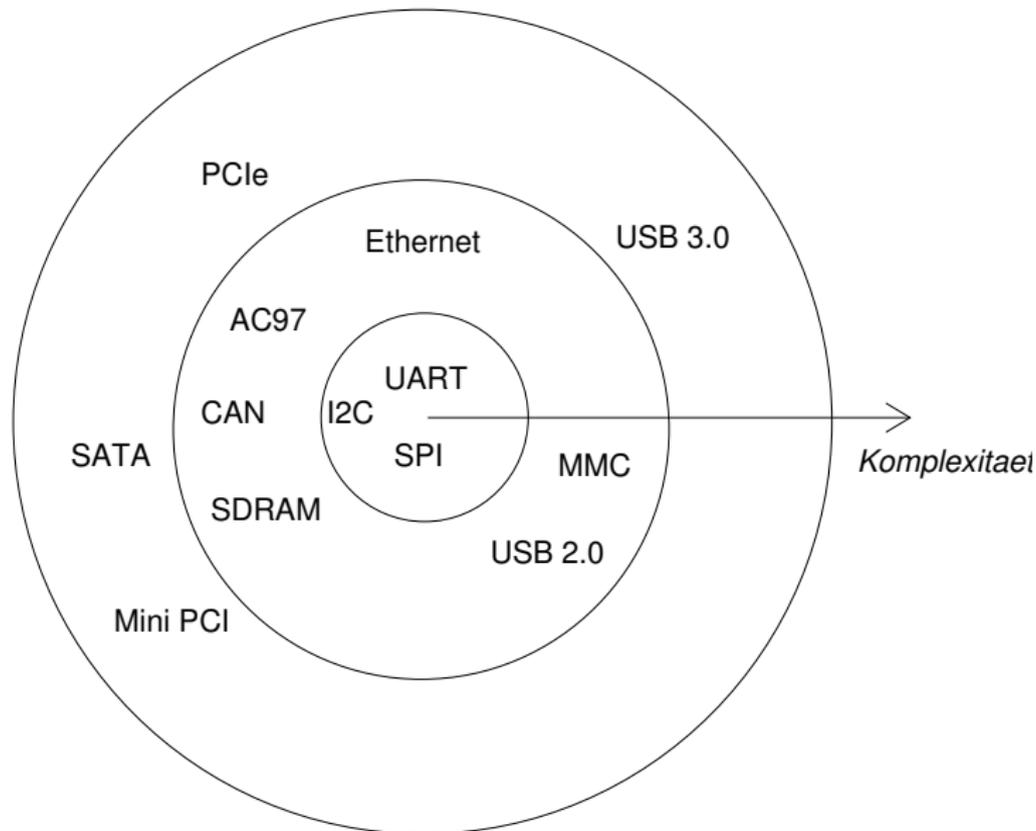
# Generisches SoC Blockdiagramm



# Bestandteile

- Kern (oft auch zusätzlicher DSP)
- Interner Speicher (Flash, SRAM)
- Schnittstellen zum externen Speicher (SDRAM, DDR-SDRAM, NAND, NOR, MMC/SD)
- UART, SPI, I2C, SSC
- USB
  - ▶ 2.0, bald 3.0
  - ▶ dev, host, otg
  - ▶ ls, fs, hs, ss
- Ethernet
- CAN
- Timer
- PWM
- ADC
- DAC
- Audio (AC97, I2S)
- Grafikcontroller
- PC-typisch: SATA, Cardbus (PCMCIA), PCIe, Mini PCI

# Schnittstellenkomplexität



## Diskussion 32-Bit

- Technische Eigenschaften im Durchschnitt alle sehr ähnlich.
- Manche Produkte haben Besonderheiten, z.B. PowerQUICC, OMAP RTU (Real Time Unit), MPC5xxx TPU.
- Verdrängung der klassischen „grossen“ und teuren standalone Prozessoren durch preiswerte SoC.
- TI: bekannt für Core + DSP. Aber: kompliziertes Programmiermodell. Aufgabe des DSP kann auch Core mit Befehlssatzerweiterung (SIMD) erledigen, bzw. Multicore.
- MPEG4 Benchmark
- ARM macht PowerPC (AMCC, Freescale) das Leben schwer.
- Intel Atom könnte ARM zurückdrängen (zwei Monopolisten).
- Preise für SoC von 5 bis 50 Euro.

## Low-Power 32-Bit

- Ein 32-Bit Core liegt mit dem  $\mu\text{A}/\text{MHz}$  Wert bei 8-bit (z.B. PIC12) Controller.
- Core nur zu etwa 1/10 am Stromverbrauch der MCU beteiligt  
Was soll das ganze Gefeilsche um Nachkommastellen beim Core?
- 9/10 geht in der Peripherie verloren

Energy Micro EFM32 „Gecko“ (Vorstellung auf ARM Techcon 2009)

- Erster 32-Bit Controller mit  $180 \mu\text{A}/\text{MHz}$
- Stop Modes: 900 nA, 600 nA, 20 nA.
- 16-128K Flash
- CR2032 Zelle (3V): 0.5% aktiv, 40 ms Periode  $\rightsquigarrow$  14 Jahre Laufzeit.  
Alle vergleichbaren 8- und 16-Bit MCUs zwischen 1 und 4 Jahren.

Speisung über Solarzelle mit 1 Watt (ca. 10 x 20 cm) kann einen 200 - 500 DMIPS Linux/Windows Rechner versorgen (0.5W). nimmt ca. 0.5 W auf.

# ARM

- Cores standardisiert, offen dokumentiert
- Peripherie vom Lizenznehmer.
- Codebasis (bis auf Peripherie) auf andere Hersteller mit ARM übertragbar.
- Cortex Microcontroller Software Interface Standard (CMSIS) – Kompatibilität unter MCUs mit Cortex Core.
- ARM7/9 technisch in Ordnung, aber nicht ultra-modern. Es gibt technisch bessere Konkurrenten im unteren DMIPS Bereich, z.B. AVR32AP/UC.
- Erfolgreiche Cores: ARM7, ARM9. Nachfolger: Cortex M Serie (M0, M3), A Serie (A5, A8, A9).
- Von ARM7 auf Cortex M0/M3 viele kleine Verbesserungen, z.B. schnellerer Interrupt. Hersteller steigen langsam um.
- Schnellster und sparsamster ARM: Cortex A9 (MP), pro CPU 2000 DMIPS bei 0.25W, bis 4 CPUs. Beispiele: Nvidia Tegra, Apple iPad A4.
- Arbeitspferd: ARM926, 400 MHz. Ablösung durch Cortex A8.

# Intel Atom

- Eroberung von Marktanteilen von den angestammten nicht-x86 SoC  
1 PC ↔ 50 Embedded Chips
- Oberstes Prinzip: kompatibel zu x86
- 45 nm, High-K, Metal-Gate (gewaltiger Forschungsaufwand von Intel)
- N270: 3900 DMIPS bei 1.6 GHz, 2.5W  
Vergleiche ARM A9 Dual Core: 4000 MIPS bei 800 MHz, 0.5W /  
10.000 MIPS bei 2 GHz, 2 W.
- Einfachstes Design seit Pentium 1993 (ca. 50 Mio Transistoren)
- Anwendungen
  - ▶ Netbooks
  - ▶ „Mobile Internet Devices“ (MID), zwischen Smartphone und Netbook
  - ▶ Embedded Markt (z.B. „Infotainment“ im Auto)
- Vorteile
  - ▶ Riesige x86 Codebasis kann ohne Änderung übernommen werden
  - ▶ Windows Desktop-Programme laufen
  - ▶ Virtualisierung, z.B. Windows + Linux

# Intel Atom (2)

- Nachteile

- ▶ Zur Zeit noch zu hoher Stromverbrauch: Z5xx im Durchschnitt 160 bis 250 mW, max. 0.6 bis 2.5 W.  
DMIPS/W um Faktor 2 schlechter als Cortex A9.  
Bald weniger Strom durch Prozessverkleinerung.
- ▶ Prozessor + **Chipsatz (US15W, i945)**: DDR2, Grafik, PCIe, USB, IDE, SDIO/MMC, LPC, SMBus, GPIO).
- ▶ SoC noch am Anfang

- Windows

- ▶ Windows CE (x86, ARM, MIPS, SuperH)
- ▶ Windows 7 Embedded (x86)

# Allgemeines

- Tools wichtiger als Chip
- Open-Source auch in Embedded angekommen: uIP, lwIP, lufa-lib, FreeRTOS, eLua, Embedded Linux, ...
- Welche Betriebssysteme gibt es für welchen Prozessor/SoC?  
RTOS (z.B. VxWorks, QNX, FreeRTOS), Windows CE, Linux, Symbian, Android, ...
- Kommerzielle IDEs ↔ Open-Source Werkzeuge
- Alle wollen Eclipse (man kann aber auch ohne leben)
- Neuer Ansatz: Entwicklungsumgebung als Web-Applikation: MBed von NXP
- Debuggen umfasst immer öfter auch die Leistungsaufnahme (siehe *PowerScale* Messgerät von Hitex).

# Eclipse Framework

- Plugins für Sprachen: CDT (C), Java, JavaScript, Ruby, XML
- Aufgabenmanagement (aufgabenorientierte Sicht mit „Mylin“)
- Versionsverwaltung (svn, git)
- Sourcecode Navigation
- Graphical Editor Framework
- Emitter Templates
- Grafische Eingabeframeworks
- Eclipse Rich Client Platform
- Eclipse Foundation
- Vielleicht in Zukunft einheitliches Framework für Embedded Entwicklung?

# Debuggen mit OpenOCD

- Freier Debugger für alle Kerne von ARM. Beliebige andere möglich, falls Dokumentation frei zugänglich.
- Fehlendes Glied in der GNU Toolchain für Embedded Entwicklung.
- Funktionsumfang wie kommerzielle Debugger für mehrere tausend Euro.
- Diplomarbeit von Dominic Rath bei H. Högl, 2005. Mittlerweile weltweit verstreute Entwicklung nach Open-Source Prinzip.
- Viele tausend Anwender weltweit
- Preiswerter USB-JTAG Adapter (FT2232), auch auf Target
- Bei jedem Linux als Standard-Paket dabei
- Integration in Eclipse möglich
- Auch Trace bereits vorgesehen.
- <http://openocd.berlios.de>, <http://www.yagarto.de>
- <http://www.hs-augsburg.de/~hhoegl/proj/openocd/index.html>

# Ausblick

- Welcher Halbleiterhersteller hat Zukunft?
  - ▶ TI wächst stark
  - ▶ PowerPC (Freescale, AMCC)?
  - ▶ NXP?
  - ▶ ARM verdrängt traditionelle Embedded Anbieter.
  - ▶ Intel möchte ARM verdrängen.
- Altbewährtes (wie 8051) wird sicher bleiben.
- Konvergenz von Embedded- und Desktop durch Atom?
- Wo liegen die physikalischen Grenzen für stromsparendes Hochleistungsrechnen?

# Das Ende

Danke für's Zuhören

Das Material zum Vortrag finden Sie unter

<http://elk.informatik.fh-augsburg.de/pub/bmk-vortrag>

# Eine Bitte

Wir suchen Unterstützer für das freie

## EMBEDDED PROJECTS JOURNAL



<http://www.embedded-projects.net/epjournal>

## Board mit SAM9G45

**Ziel:** Ein Board bauen mit SAM9G45 (ARM926EJ-S, 400 MHz), das für 50 Euro von Studenten und sonstigen Interessierten gekauft werden kann. Es wird bei den Veranstaltungen Embedded Systems und Embedded Linux eingesetzt werden.

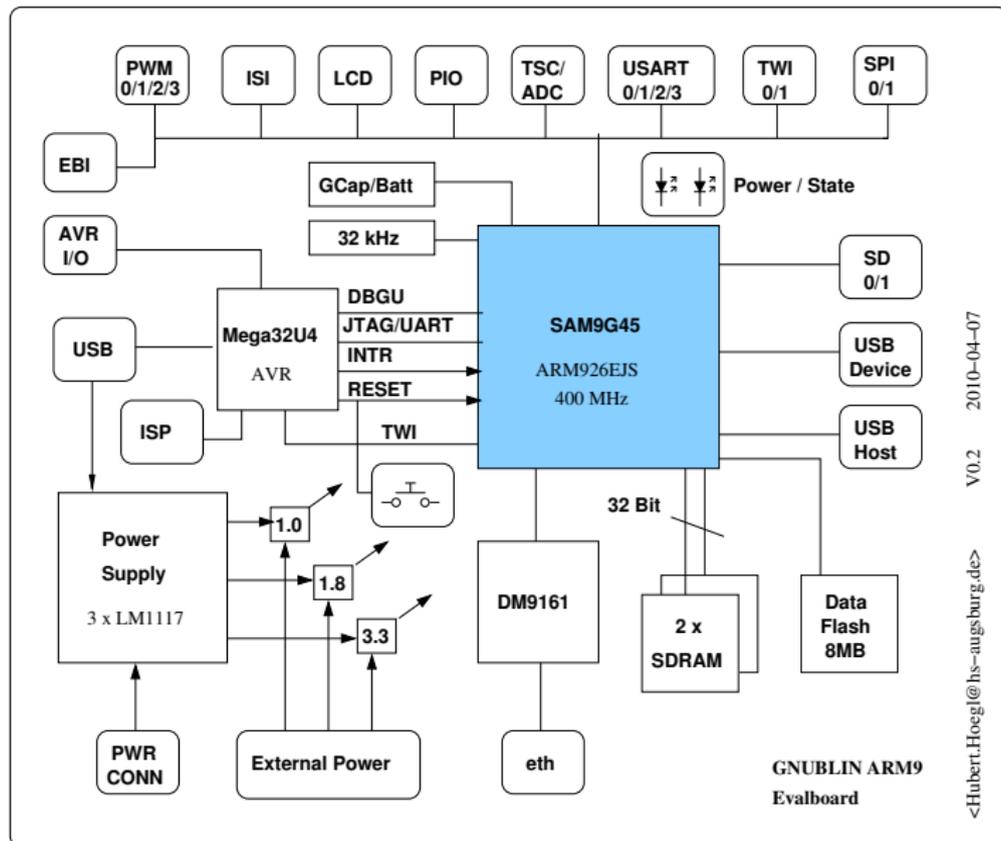
Besonderheit: Eine USB-Verbindung am Atmel Mega32U4 reicht für USB-UART, USB-JTAG und Stromversorgung.

Offenes Gemeinschaftsprojekt:

```
git clone  
git://elk.informatik.fh-augsburg.de/srv/git/repositories/g45board.git
```

Interessenten melden sich bei [Hubert.Hoegl@hs-augsburg.de](mailto:Hubert.Hoegl@hs-augsburg.de).

# Board mit SAM9G45 (2)



# Board mit SAM9G45 (3)

## Atmel SAM9G45 mit ARM926EJ-S Core

