

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

Technische Informatik an der Hochschule Augsburg

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 hochkomplexen 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*
 - 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
- Stark individualisierte Produkte
- System-on-Chip
- Extrem niedrige Preise
- Energiebewusstsein in allen Leistungsklassen
- ARM überall
- Intel spielt nun mit
- Immer mehr Geräte laufen mit Linux

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 (vermeide *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, ... ?

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

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, zunehmend 32-Bit. Beispiele: Atmel AVR Mega128, AVR32UC, SoC mit 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, ...

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)
- 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.

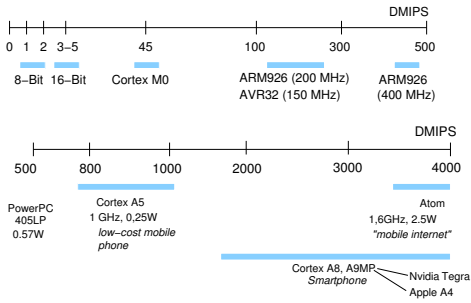
P_{static} : fast keine statischen Pfade (sub- μ Watt)

Neuer Prozess: Kleinere Strukturen \rightsquigarrow höhere Integration \rightsquigarrow niedrigere Spannung \rightsquigarrow niedrigere dynamische Leistungsaufnahme

Aber: Niedrigere Spannung kontraproduktiv für höhere Frequenzen \rightsquigarrow niedrigere MOS Schwellenspannung U_t \rightsquigarrow schnellere Transistoren \rightsquigarrow aber höhere Leckströme

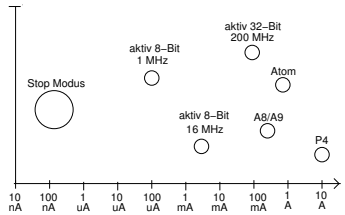
\rightsquigarrow Verwende zweierlei Transistoren: „langsam“/stromsparend, schnell/„stromverschwendend“.

DMIPS-Skala



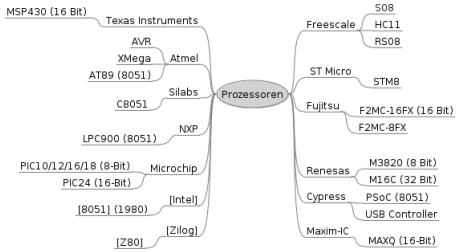
Leistungs-Skala

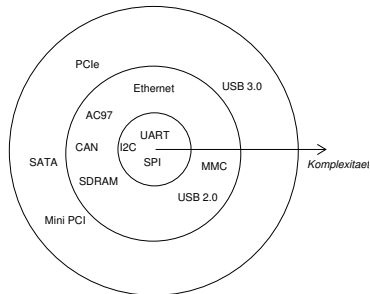
Strom bzw. Leistung geht über 10 Zehnerpotenzen!



Prozessorhersteller 8- und 16-Bit

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





- Technische Eigenschaften sehr ähnlich. Selten entscheiden diese die Auswahl.
- Verdrängung der klassischen „grossen“ und teuren standalone Prozessoren durch SoC.
- TI: bekannt für Core + DSP. Aber: kompliziertes Programmiermodell. Core und DSP haben ähnliche Rechenleistung.
- Cores mit SIMD Befehlssatz für Signalverarbeitung (fast alle).
- 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

- Die $\mu\text{A}/\text{MHz}$ Werte sind bei 8-bit (z.B. PIC12) und 32-bit (z.B. SAM9G45) fast identisch.
- Core nur zu etwa 1/10 am Stromverbrauch der MCU beteiligt
Was soll das ganze Gefälsche um Nachkommastellen beim Core?
- 9/10 geht in der Peripherie verloren

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

- Erster 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. $10 \times 20 \text{ cm}$) 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.
- Erfolgreiche Cores: ARM7, ARM9. Nachfolger: Cortex M Serie (M0, M3), A Serie (A5, A8, A9).
- Codebasis (bis auf Peripherie) auf andere Hersteller 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.
- Von ARM7 auf Cortex M0/M3 viele kleine Verbesserungen, z.B. schnellerer Interrupt. Hersteller steigen langsam um.
- Arbeitspferd: ARM926, 400 MHz. Ablösung durch Cortex A8 (siehe Board mit Atmel SAMG45 im Anhang).
- Schnellster und sparsamster ARM: Cortex A9 (MP), pro CPU 2000 DMIPS bei 0.25W, bis 4 CPUs. Beispiele: Nvidia Tegra, Apple iPad A4.

- Eroberung von Marktanteilen von den angestammten nicht-x86 SoC
1 PC ↔ 50 Embedded Chips
- Oberstes Prinzip: kompatibel zu x86
- Intel vs. ARM
- 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

- 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 Prozesserverkleinerung.
 - ▶ 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

- Welche Betriebssysteme gibt es für welchen Prozessor/SoC?
RTOS (z.B. VxWorks, QNX, FreeRTOS), Windows CE, Linux, Symbian, Android, ...
- Tools wichtiger als Chip
- Kommerzielle IDEs ↔ Open-Source Tools
- Eclipse
- Debuggen umfasst immer öfter auch die Leistungsaufnahme (siehe *PowerScale* Messgerät von Hitex).
- Neuer Ansatz: Entwicklungsumgebung als Web-Applikation: MBed von NXP

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?

- 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 (FT232), 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>

- 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.

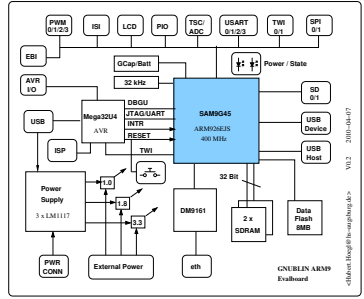
Besonderheit: Eine USB-Verbindung am Atmel Mega32U4 reicht für USB-UART, USB-JTAG und Stromversorgung.

Offenes Gemeinschaftsprojekt:

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

Interessenten melden sich bei Hubert.Hoegl@hs-augsburg.de.

Board mit SAM9G45 (2)



Board mit SAM9G45 (3)

Atmel SAM9G45 mit ARM926EJ-S Core

