

Die Geschichte der ARM

Prozessoren

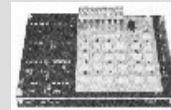
Zusammengestellt von Roland Schefer

1978 Gründung der Firma Acorn



1979 Acorn System 1

6502 CPU



Bausatz
Hex Tastatur

1980 Acorn Atom

6502 CPU



Heimcomputer
Bausatz

1981 BBC Microcomputer
Fernsehserie
"The Computer Programme"

2 MHz 6502



1.5 Mio
Geräte
(Bausätze)

1983 Entwicklung der 32 Bit ARM Architektur

1985 Acorn RISC Machine

ARM1

25'000 Transistoren



1987 Acorn Archimedes PC

ARM2

> mit RISC-OS

ARM3

1990 Gründung der Firma Advanced RISC Machines zur Vermarktung der ARM Architektur auf Basis eines Lizenzmodelles

Gründungsmitglieder:

APPLE hatte interesse an einem besonders stromsparenden Prozessors (-> 1993 Newton PDA)

ACRON war an der Weiterentwicklung seiner Workstations interessiert

VLSI (später Philips) wollte primär Prozessor und Mikrocontrollerchips fertigen und vermarkten

ARM- Core Entwicklungen 1/2

ARM 1 (1985)

Der Prozessor war vom Kern her 32-bittig, konnte aber nur 26-bittig Adressieren

ARM 2 (1987)

ARM2 wurde auch in einer statischen Ausführung gefertigt, was ermöglichte, im PowerDown-Modus den Takt zu verringern oder auch (ohne Datenverlust) zeitweilig ganz abzuschalten

ARM 3 (1989)

Weist jetzt sowohl einen Multiplizierer als auch Cache auf, unterstützt Co-Prozessoren.

ARM 4 (Thumb Befehlssatz)

Zusätzlicher 16Bit Befehlssatz "Thumb InstructionSet" der Version ARM4T beschränkt sich auf 16 Bit und ermöglichte so besonders kompakten Code.

Version 5TEJ(2000)

Die integrierte "Jazelle" verarbeitet Java Code direkt in der Hardware und beschleunigt damit Java Programme bis zu 8 mal gegenüber reinen Softwarelösungen. Ausserdem konnte der Energiebedarf (Bei Java Ausdrücken) um 80% gesenkt werden.

ARM 6 (2001)

Einführung der SIMD-Technik (ähnlich MMX), durch die sich mit einem Befehl mehrere Daten gleichzeitig bearbeiten lassen.

Dies kann Audio / Video Anwendungen (z.B. in PDA's) um bis zu Faktor 4 beschleunigen. Eine um 30 Prozent höhere Performance Multiprozessor-Support mit 'Shared Memory' 77 neue Befehle, 6 davon Thumps 8-stufige Integer Pipeline, neues L1 Cache System

ARM7TDMI (1993) 40-180MHz

Die ARM7TDMI besitzt eine dreistufige Pipeline und einen gemeinsamen Datenbus für Instruktionen und Daten (Von Neumann Architektur).

Er wird millionenfach in Handys und Spielkonsolen eingesetzt. Die diversen Lizenznehmer entwickelten ARM-Varianten, die sich durch Performance aber auch durch die geringe Leistungsaufnahme auszeichneten. So gab es mit dem StrongARM, entwickelt von Digital Equipment, eine Version die mit 233 MHz getaktet wurde und nur 1 W benötigte.

Die ARM9 (1997) 200- 400MHz

Unterschied der ARM9 gegenüber der ARM7 ist ein getrennter Datenbus jeweils für Instruktionen und Daten (Harvard-Architektur) und eine 5-stufige Pipeline.

ARM-Core Entwicklungen 2/2

ARM 10TDMI -800MHz

Mit Thumb extension

32x16 MAC-multiplier

6-stage pipeline for fixed point instructions

ARM11 (2002)

Der ARM11 basiert auf der Prozessor-Architektur des ARM6 und ist für Taktfrequenzen von mehreren hundert Megahertz bis sogar bis über 1 GHz konzipiert. Eingesetzt wird der ARM11 in der Automobil-Technik, in drahtlosen Konsumergeräten, in WLANs und Bluetooth. Nachfolger des ARM11 sind die Cortex-Prozessoren.

ARM11 Cortex (2005) 0.6-1,2GHz

Mit dem Cortex-A8 hat ARM einen leistungsfähigen 32-Bit-Prozessor mit niedriger Leistungsaufnahme entwickelt, der eine deutliche Leistungssteigerung gegenüber dem ARM-Prozessor ARM11 aufweist. Diese Leistungssteigerung verdankt der Cortex-A8 einer 13-stufigen Pipeline und dem superskalaren Aufbau. Er arbeitet mit zwei Recheneinheiten (ALU) und einer Load/Store-Einheit, die ihre Arbeit vom Prozessor zugeteilt bekommen.

Für spezielle Aufgaben stehen noch Erweiterungseinheiten wie die Signalverarbeitungseinheit zur Verfügung, die die CPU beim Abspielen von MP3-Dateien und Videos entlasten.

Die Cortex-Prozessoren, die in 65-nm-Technologie gefertigt werden, arbeiten mit Taktfrequenzen von bis zu 1 GHz und erreichen dabei 200 DMIPS. Ihre Leistungsaufnahme ist mit 300 mW bei 600 MHz äußerst gering. Als Befehlssatz benutzen sie den des ARM7.

Coprozessor für ARM-Kerne

Die ARM ist als Mikroprozessor-Kern in eingebetteten Systemen gedacht, in denen meist keine Gleitkomma-Arithmetik benötigt wird. Der ARM wurde jedoch speziell im Hinblick auf Erweiterbarkeit um Coprozessoren entwickelt und besitzt ein eigenes Coprozessor-Interface und Befehle für optionale Coprozessoren.

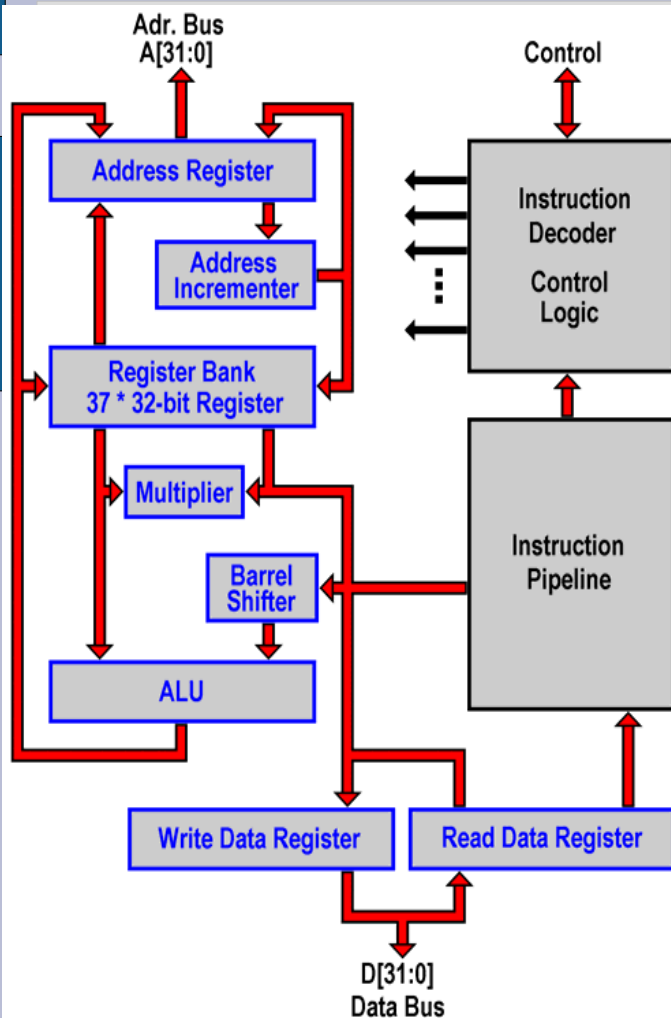
Erweiterungen für ARM-Kerne

ARM Ltd. verkauft neben den ARM-CPU Kernen auch Erweiterungen als synthetisierbare Makrozellen für den System-on-Chip-Entwurf, unter anderem Memory Management Unit, Floating-Point Coprozessor, Signalprozessor-Erweiterung (Piccolo).

Lizenznehmer:

Analog Devices, Apple, Atmel, Conexant, Freescale (ehemals Motorola), HTC Corporation, HP, IBM, Infineon, Intel (XScale), Luminary Micro, Motorola, NEC, NetSilicon, Nintendo, NXP (ehemals Philips), Oki, Palm, Samsung, Sony, STMicroelectronics, Texas Instruments, Toshiba und weitere

Der ARM7 Kern



Data Bus

Für die Verbindung zur Außenwelt besitzt ein ARM-CPU-Core einen 32-bit-Datenbus und Adressbus -> 4GB Adressraum!

Address Register:

Dieses CPU-interne 32-bit-Register enthält stets die Adresse des Speicherplatzes, auf den die CPU zugreifen soll.

Address Incrementer:

Über diese ARM-interne Funktionseinheit wird die interne 32-bit-Adresse inkrementiert. Als typische 32-bit-CPU adressiert ARM jeweils 32-bit-Worte

Register Bank:

Hier sind die internen 32-bit-Register einer ARM-CPU zusammengefasst, die bei der Befehlsbearbeitung eine wichtige Rolle spielen.

Multipliiert:

Diese Funktionseinheit führt sehr schnelle Multiplikationen mit Hilfe festverdrahteter Logik aus.

Barrel Shifter:

Mit Hilfe dieses speziellen Schieberegisters kann eine ARM-CPU Schiebe- und Rotationsaufgaben um eine beliebige Bitanzahl innerhalb eines Taktes durchführen.

ALU: Die ALU ist das Rechenwerk einer CPU

Write Data Register und Read Data Register:

In diesen beiden 32-bit-Registern werden die Daten für den Schreib- und Lesezugriff auf externe Funktionseinheiten (Speicher- und Peripherie) zwischengespeichert.

Instruction Decoder / Control Logic:

In dieser ARM-Funktionseinheit werden sämtliche Befehle durch eine festverdrahtete Logik dekodiert. Diese Komponente steuert auch die Ausführung der einzelnen Befehle in der CPU.

Instruction Pipeline:

Diese Pipeline realisiert die Funktionen zur Beschleunigung der Befehlsbearbeitung. Bei einem ARM7TDMI kommt eine 3-stufige Pipeline zum Einsatz.

Die ARM Register

Die ARM-CPU enthält 37 interne 32bit-Register.

31 dieser 37 Register sind als Universal-Register nutzbar. Darin ist der Programmzähler PC enthalten, über welchen der nächste auszuführende Befehl im Speicher adressiert wird. Sechs weitere 32-bit-Register dienen als Statusregister (CPSR). In diesen Registern werden aber nur jeweils 12 einzelne Bits benutzt. Alle anderen Bits eines CPSR sollten als reserviert betrachtet werden und ungenutzt bleiben.

Die ARM-Universal-Register tragen die Namen R0 bis R15. R15 ist der PC.

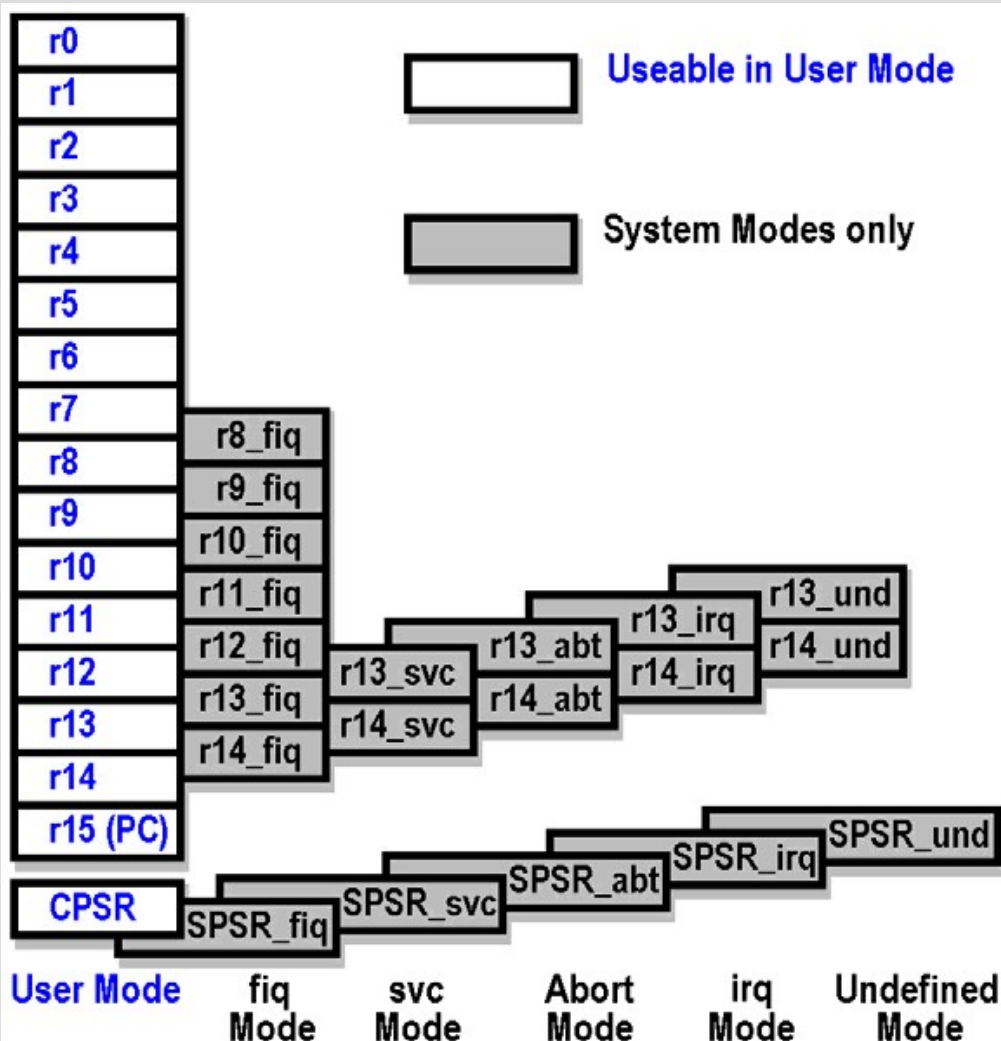
Die Register R0 bis R14 lassen sich weiterhin in zwei Gruppen unterteilen:

1. die Gruppe der Unbanked Registers R0 bis R7 d.H. stets die gleiche Funktion.
2. die Gruppe der Banked Registers R8 bis R14 d.H. hängt von derjeweiligen Betriebsart ab.

Thumb-Register

Lo/Hi-Register: R0...R7/R8...R15 (Hi-Register nur eingeschränkt benutzbar)

R13-Stackpointer, R14-Linkregister, R15-Programmzähler, 16-Bit 2-Adress Befehlsformat



Der ARM Befehlssatz

Alle ARM-CPU's besitzen einen 32-bit-Befehlssatz.

Die gesamte ARM-Speicherorganisation basiert auf einer 32-bit-Struktur.

Ein einziger Load oder Store-Befehl kann jeweils ein 32-bit-Wort zwischen einem CPU-Register und dem Speicher hin und her transferieren.. Die CPU sieht den Speicher als Folge von Bytes d.H. Linearer Speicher bis 4GB.

Besonderheiten des ARM Befehlssatzes

Sämtliche Befehle können bedingt ausgeführt werden. Damit entfällt die Notwendigkeit für Programmsprünge in vielen Standardsituationen, z. B. If-else-Abfragen

Der Thumb Befehlssatz

Bei den Architekturvarianten mit Thumb-Befehlssatz (zum Beispiel ARM7TDMI)ist zusätzlich noch ein 16-bit- (eingeschränkter) Befehlssatz vorhanden. Im ARM-Thumb-Modus bewegen die Load- und Store-Befehle jeweils 16-bit-Wörter.

Es kann per Software jederzeit zwischen beiden Befehlssätzen umgeschaltet werden.

Thumb vs. ARM-Befehlssatz:

ca. 40% mehr Befehle pro Programm

ca. 70% der Programmgröße

ca. 30% weniger Stromverbrauch für Speicherzugriffe

ca. 45% schneller (16-bit Speicher) / 40% langsamer (32-bit Speicher)

ARM unterstützt grundsätzlich die Little-endian- und Big-endian-Speicherorganisation

(Abbildung). Im ersten Fall (Little-endian) beinhaltet die niederwertige Byte-Adresse eines Speichers jeweils das niederwertige Byte eines Wortes. Beim Big-endian-Format ist es genau umgekehrt: Das höherwertige Byte eines Wortes ist in der niederwertigen Byte-Adresse gespeichert. Bei vielen ARM-MCUs und SoCs kann zwischen Little-endian- und Big-endian-Format umgeschaltet werden. Einige ARM-basierte Varianten unterstützen allerdings nur das Little-endian- Format. Diese Speicherorganisationsform sollte daher für ARM-Anwendungen als Favorit betrachtet werden.

Bit 31	Byte 3	Byte 2	Byte 1	Byte 0	Bit 0
	B	A	9	8	0x0000.0008
	7	6	5	4	0x0000.0004
	3	2	1	0	0x0000.0000

Little-endian Format

Bit 31	Byte 0	Byte 1	Byte 2	Byte 3	Bit 0
	8	9	A	B	0x0000.0008
	4	5	6	7	0x0000.0004
	0	1	2	3	0x0000.0000

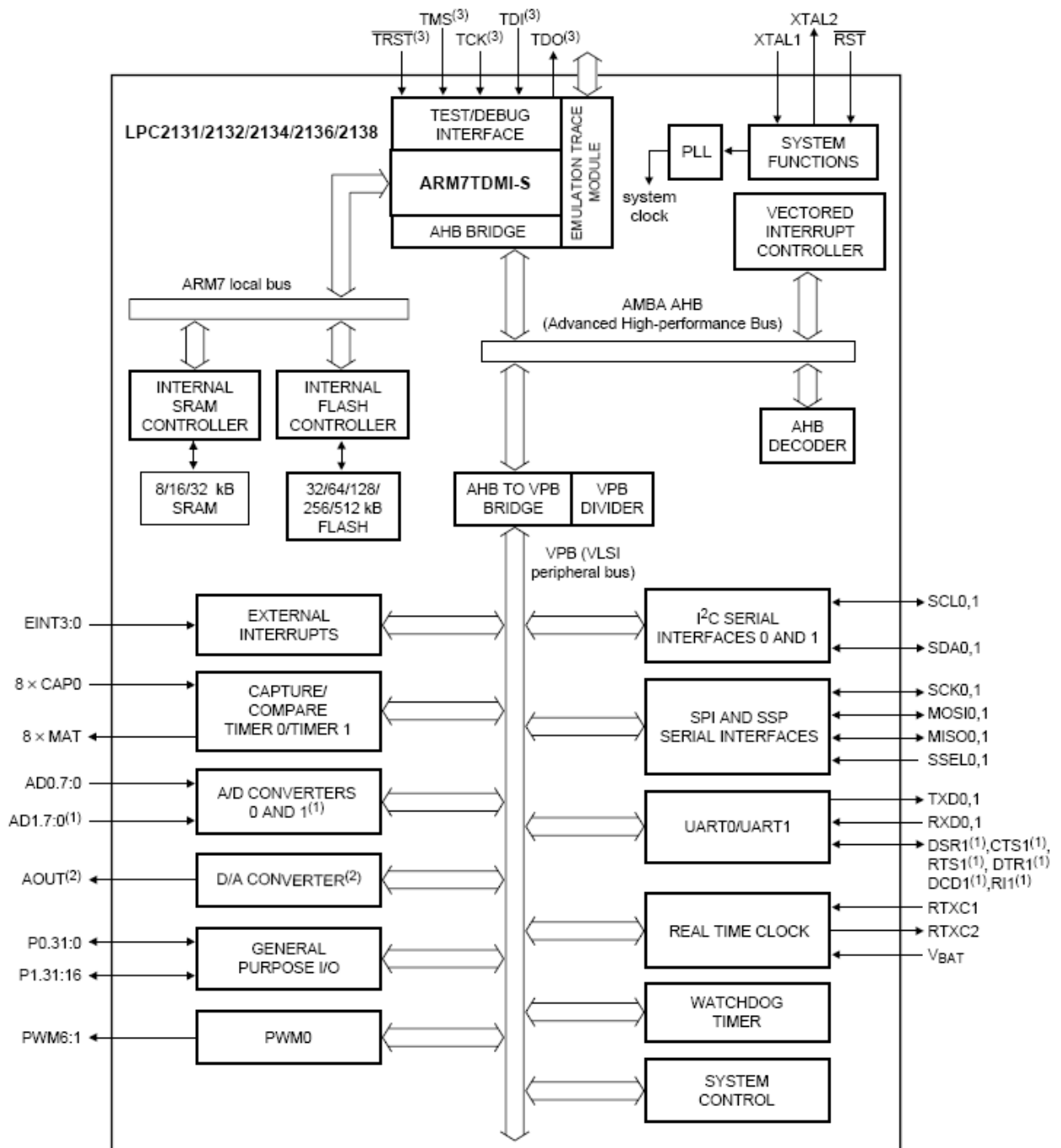
Big-endian Format

Philips Microcontroller LPC2138

mit ARM7TDMI Core

Blockschaltbild

LPC21XX Microcontroller mit ARM7 Core



Die LPC2000 Familie

Die 16/32-Bit LPC2000 Familie wurde für verschiedenste Anwendungen entwickelt, die von industriellen Steuerungen über Automobilelektronik und Consumer-Anwendungen bis zu sämtlichen Embedded-Applikationen reichen, bei denen von einem 32-Bit Mikrocontroller hohe Performance und geringe Leistungsaufnahme erwartet werden.

Diese auf einem 0,18- μ m-Prozess basierenden Bausteine arbeiten mit 60MHz und enthalten bis zu 512KB On-Chip-Flash-Programmspeicher und bis zu vier verbundene CAN-Schnittstellen. Der Flash-Speicher arbeitet praktisch ohne Wartezyklen und ermöglicht eine maximale Leistung, da im Gegensatz zu Lösungen der Konkurrenz, bei denen das Leistungsvermögen aufgrund von Flash-Speichern mit Wartezyklus eingeschränkt wird, die MCU mit gleicher Geschwindigkeit wie ein SRAM-Speicher arbeiten kann.

LPC21XX Familie

Wesentliche Merkmale:

- * Basierend auf dem 1,8V ARM7TDMI-S Core
- * Taktfrequenz bis zu 75MHz
- * Serielle Schnittstellenoptionen, CAN, I2C, UART, SPI, USB und Ethernet
- * 10-Bit A/D- und D/A-Wandler-Optionen
- * Geringe Leistungsaufnahme
- * Kleiner Footprint

Weitere Info unter <http://www.standardics.nxp.com/products/lpc2000/>