Details of the ARM Architecture

Sommerakademie in Leysin
AG 2 – Effizientes Rechnen

John Doe

University of Awesomeness

August 2016

Studienstiftung
des deutschen Volkes

Outline

Introduction

- ARM is a RISC computer architecture / processor design

Introduction

- ARM is a RISC computer architecture / processor design
- Developed by Advanced RISC Machines Ltd. (ARM), formerly Acorn, first released 1985

Introduction

- ARM is a RISC computer architecture / processor design
- Developed by Advanced RISC Machines Ltd. (ARM), formerly Acorn, first released 1985
- Licensing technology (IP) to semiconductor fabricators, NOT manufacturing themselves

Introduction

- ARM is a RISC computer architecture / processor design
- Developed by Advanced RISC Machines Ltd. (ARM), formerly Acorn, first released 1985
- Licensing technology (IP) to semiconductor fabricators, NOT manufacturing themselves
- Mainly used in **embedded systems**, e.g. mobile phones, tablets, ...

Introduction

- ARM is a RISC computer architecture / processor design
- Developed by Advanced RISC Machines Ltd. (ARM), formerly Acorn, first released 1985
- Licensing technology (IP) to semiconductor fabricators, NOT manufacturing themselves
- Mainly used in **embedded systems**, e.g. mobile phones, tablets, ...
- Advantages:
  - Low power comsumption
  - Price sensitive
  - Flexibility

Introduction

- ARM is a RISC computer architecture / processor design
- Developed by Advanced RISC Machines Ltd. (ARM), formerly Acorn, first released 1985
- Licensing technology (IP) to semiconductor fabricators, NOT manufacturing themselves
- Mainly used in **embedded systems**, e.g. mobile phones, tablets, ...
- Advantages:
  - Low power comsumption
  - Price sensitive
  - Flexibility
- As of 2014 50 billion ARM chips produced thus far, 15 bn in 2015
  Share of about 90 % at smartphones

In General

- **RISC / CISC: R**educed / **C**omplex **I**nstruction **S**et **C**omputing

In General

- **RISC / CISC: R**educed / **C**omplex **I**nstruction **S**et **C**omputing
- RISC: simple, but powerful instructions, executed in one cycle
- CISC: more and complex instructions, more work done in the core

In General

- **RISC / CISC: R**educed **/ C**omplex **I**nstruction **S**et **C**omputing
- RISC: simple, but powerful instructions, executed in one cycle
- CISC: more and complex instructions, more work done in the core
- RISC architecture reduces hardware complexity and emphasizes software/compiler complexity

In General

- **RISC / CISC: R**educed **/ C**omplex **I**nstruction **S**et **C**omputing
- RISC: simple, but powerful instructions, executed in one cycle
- CISC: more and complex instructions, more work done in the core
- RISC architecture reduces hardware complexity and emphasizes software/compiler complexity

Summary

RISC reduces amount of hardware, CISC focuses performance

Major Design Rules of RISC

1. **Instructions:** Only basic instructions, complicated operations synthesized by elementary instructions, fixed length

Major Design Rules of RISC

1. **Instructions:** Only basic instructions, complicated operations synthesized by elementary instructions, fixed length
2. **Pipelines:** Instructions always decoded in one pipeline stage, no need for microprograms

Major Design Rules of RISC

1. **Instructions:** Only basic instructions, complicated operations synthesized by elementary instructions, fixed length
2. **Pipelines:** Instructions always decoded in one pipeline stage, no need for microprograms
3. **Registers:** Larger, general-purpose register set. Serves as fast memory store

Major Design Rules of RISC

1. **Instructions:** Only basic instructions, complicated operations synthesized by elementary instructions, fixed length
2. **Pipelines:** Instructions always decoded in one pipeline stage, no need for microprograms
3. **Registers:** Larger, general-purpose register set. Serves as fast memory store
4. **Load/Store architecture:** Processor operates only on registers. Load and Store instructions access external memory

Registers

- Total of 16 registers, each 32-bit in size

Registers

- Total of 16 registers, each 32-bit in size
- Usually three dedicated registers:
    - r13 as stack pointer, designates stack memory frame
    - r14 as link register, return address for subroutines
    - r15 as program counter, address of next instruction

Registers

- Total of 16 registers, each 32-bit in size
- Usually three dedicated registers:
  - r13 as stack pointer, designates stack memory frame
  - r14 as link register, return address for subroutines
  - r15 as program counter, address of next instruction
- Additionally "Current Program Status Register" (CPSR)

ARM-specific Instruction Set Features

ARM Instruction Set differs from a strict RISC implementation:

- Load/Store multiple instructions have variable cycle length, higher code density, better performance by sequential memory access

ARM-specific Instruction Set Features

ARM Instruction Set differs from a strict RISC implementation:

- Load/Store multiple instructions have variable cycle length, higher code density, better performance by sequential memory access
- Conditional executions possible, reducing branches

ARM-specific Instruction Set Features

ARM Instruction Set differs from a strict RISC implementation:

- Load/Store multiple instructions have variable cycle length, higher code density, better performance by sequential memory access
- Conditional executions possible, reducing branches
- Inline barrel shifter

ARM-specific Instruction Set Features

ARM Instruction Set differs from a strict RISC implementation:

- Load/Store multiple instructions have variable cycle length, higher code density, better performance by sequential memory access
- Conditional executions possible, reducing branches
- Inline barrel shifter

Supported instruction set modes:

- Jazelle provides direct support of Java bytecode
- Thumb instructions

Thumb Instructions

- 16-Bit long instruction subset with most used instructions

Thumb Instructions

- 16-Bit long instruction subset with most used instructions
- Enhances code density substantially, about 30 % saving
- Important for embedded devices

Thumb Instructions

- 16-Bit long instruction subset with most used instructions
- Enhances code density substantially, about 30 % saving
- Important for embedded devices
- Lack of performance due to less functionality

Basic Pipeline Scheme

- Up to ARM7 : Fetch - Decode - Execute
- With ARM9 : Fetch - Decode - Execute - Memory - Write
- With ARM10 : Fetch - Issue - Decode - Execute - Memory - Write
- Newer generations up to 14 stage pipeline (e.g. ARM Cortex-A8)
- ARM Cortex-A8 first to incorporate superscalar elements

Basic Pipeline Scheme

- Up to ARM7 : Fetch - Decode - Execute
- With ARM9 : Fetch - Decode - Execute - Memory - Write
- With ARM10 : Fetch - Issue - Decode - Execute - Memory - Write
- Newer generations up to 14 stage pipeline (e.g. ARM Cortex-A8)
- ARM Cortex-A8 first to incorporate superscalar elements

Classical distinction between RISC and CISC blurs the more it has been developing

Power Consumption: RISC Architecture

- RISC architecture means processor design with less transistors
- ⇒ Less switching activity and thus less power consumption!

Power Consumption: RISC Architecture

- RISC architecture means processor design with less transistors
- ⇒ Less switching activity and thus less power consumption!

Comparison:	ARM Cortex-A9	Intel Core i7-960
	26 Million Transistors	731 Million Transistors

Power Consumption: RISC Architecture

- RISC architecture means processor design with less transistors
- ⇒ Less switching activity and thus less power consumption!

Comparison:	ARM Cortex-A9	Intel Core i7-960
	26 Million Transistors	731 Million Transistors
	0.5 - 1.9 W	130 W (TDP)

Power Consumption: SoCs

- ARM design allows purpose-oriented optimization

Power Consumption: SoCs

- ARM design allows purpose-oriented optimization
- **System-on-a-Chip** (SoC): Suitable additional components melted together with ARM core
- Only required components ⇒ Only required power

Heat Dissipation

- Due to low power consumption: heat dissipation generally lower

Heat Dissipation

- Due to low power consumption: heat dissipation generally lower
- Hot spots can be problematic during longer computing times
- Difficult to spread the heat in little cases

Heat Dissipation

- Due to low power consumption: heat dissipation generally lower
- Hot spots can be problematic during longer computing times
- Difficult to spread the heat in little cases
- Liquid cooling/heat pipe technique in smartphones

Heat Dissipation

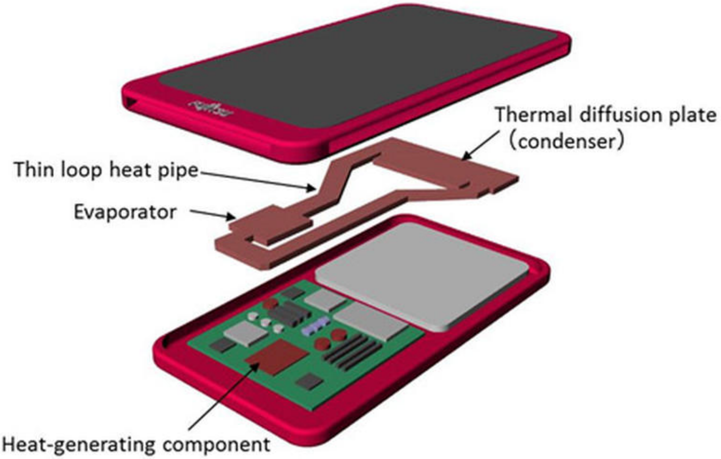
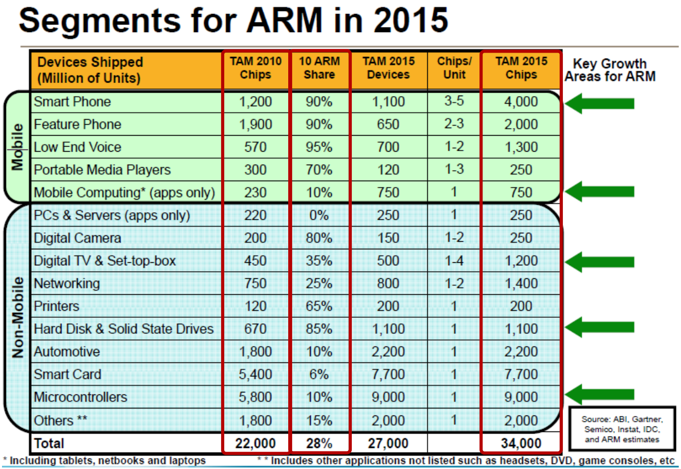


Figure: `http://www.zdnet.com/article/fujitsu-has-a-cool-liquid-answer-to-hot-spots-in-smartphones/`

The ARM Ecosystem

- Licensing model allows various companies to implement technology
- SoCs for every conceivable purpose powered by ARM cores
- Creating versatile, specialized market based on a common architecture
- ARM instruction set already dominant on smartphone market (Android and iOS)

Applications of ARM



Segments for ARM in 2015

	Devices Shipped (Million of Units)	TAM 2010 Chips	10 ARM Share	TAM 2015 Devices	Chips/ Unit	TAM 2015 Chips	Key Growth Areas for ARM
Mobile	Smart Phone	1,200	90%	1,100	3-5	4,000	←
	Feature Phone	1,900	90%	650	2-3	2,000	
	Low End Voice	570	95%	700	1-2	1,300	
	Portable Media Players	300	70%	120	1-3	250	
	Mobile Computing* (apps only)	230	10%	750	1	750	←
Non-Mobile	PCs & Servers (apps only)	220	0%	250	1	250	
	Digital Camera	200	80%	150	1-2	250	
	Digital TV & Set-top-box	450	35%	500	1-4	1,200	←
	Networking	750	25%	800	1-2	1,400	
	Printers	120	65%	200	1	200	
	Hard Disk & Solid State Drives	670	85%	1,100	1	1,100	←
	Automotive	1,800	10%	2,200	1	2,200	
	Smart Card	5,400	6%	7,700	1	7,700	
	Microcontrollers	5,800	10%	9,000	1	9,000	←
	Others **	1,800	15%	2,000	1	2,000	
	Total	**22,000**	**28%**	**27,000**		**34,000**	

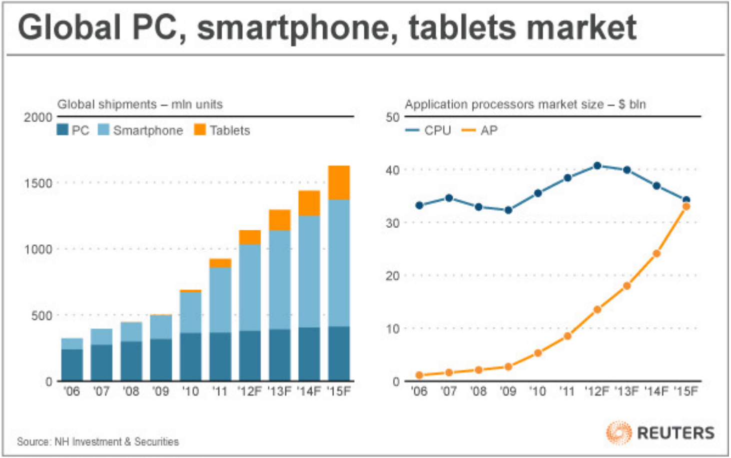
Source: ABI, Gartner, Semico, Instat, IDC, and ARM estimates

* Including tablets, netbooks and laptops

* * Includes other applications not listed such as headsets, DVD, game consoles, etc

Figure: http://www.cityindex.co.uk/market-analysis/market-news/40069092016/arm-shares-steady-on-firm-grip-of-smartphone-market/

Competition with PC Market



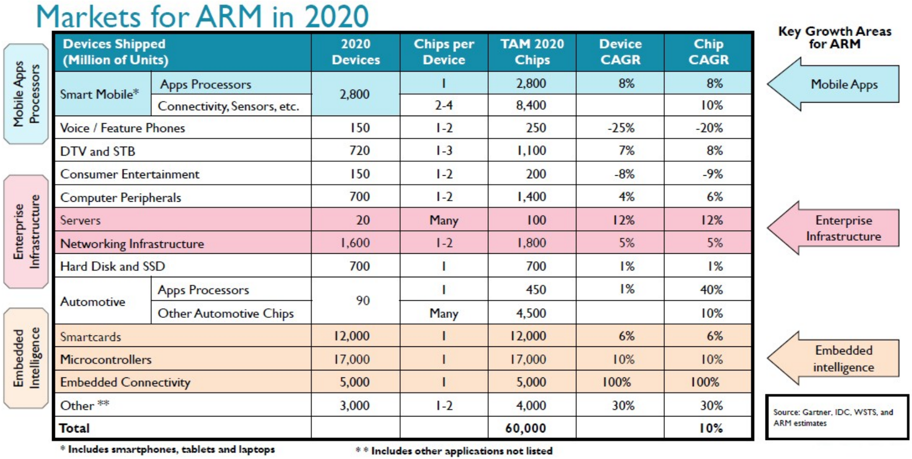
Reuters graphic/Christine Chan 20/06/12

Figure: http://gadgets.ndtv.com/mobiles/news/smart-logic-samsung-chips-away-at-intel-lead-234639

64-bit ARM Architecture

- ARM Cortex-A50 series released in 2013 with 64-bit architecture
- First consumer device: iPhone 5S with Apple A7 processor
- Improvements:
  - Using 64-bit adresses, no 3.5 GB memory limitation
  - 31 instead of 15 general purpose registers

Future Markets



Markets for ARM in 2020

Devices Shipped (Million of Units)		2020 Devices	Chips per Device	TAM 2020 Chips	Device CAGR	Chip CAGR
Smart Mobile*	Apps Processors	2,800	1	2,800	8%	8%
	Connectivity, Sensors, etc.		2-4	8,400		10%
Voice / Feature Phones		150	1-2	250	-25%	-20%
DTV and STB		720	1-3	1,100	7%	8%
Consumer Entertainment		150	1-2	200	-8%	-9%
Computer Peripherals		700	1-2	1,400	4%	6%
Servers		20	Many	100	12%	12%
Networking Infrastructure		1,600	1-2	1,800	5%	5%
Hard Disk and SSD		700	1	700	1%	1%
Automotive	Apps Processors	90	1	450	1%	40%
	Other Automotive Chips		Many	4,500		10%
Smartcards		12,000	1	12,000	6%	6%
Microcontrollers		17,000	1	17,000	10%	10%
Embedded Connectivity		5,000	1	5,000	100%	100%
Other **		3,000	1-2	4,000	30%	30%
Total				**60,000**		**10%**

* Includes smartphones, tablets and laptops

** Includes other applications not listed

Source: Gartner, IDC, WSTS, and ARM estimates

Figure: http://www.nextplatform.com/2015/10/06/why-are-we-still-waiting-for-arm-servers/

Mill Architecture

- Radical new approach to processor design

Mill Architecture

- Radical new approach to processor design
- Key concept: Register Belt instead of random access registers
- Shall improve performance by less register accesses

Mill Architecture

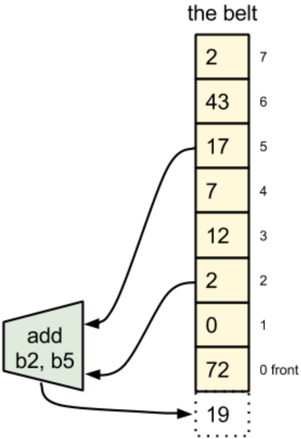


Figure: http://millcomputing.com/blog/wp-content/uploads/2014/02/intro_belt.png

Mill Architecture

- Radical new approach to processor design
- Key concept: Register Belt instead of random access registers
- Shall improve performance by less register accesses

Problems:

- Very, very, complex programming

`<insert obligatory cookie/potato meme here>`

Sänk ju!