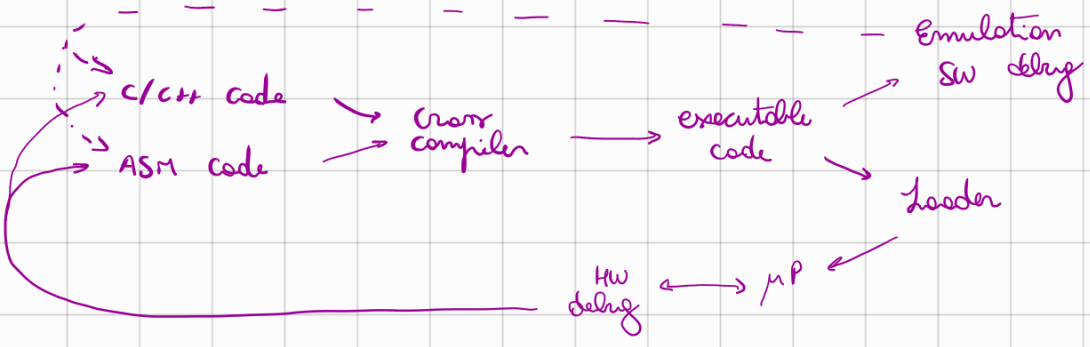
Applications & Tools

- Arm non fa niente in silicone, ha basi operative sparse in tutto il mondo
- I processori Arm sono venduti come core da usare per l'integrazione di System on a chip. I core possono essere hardcore, dove Arm mette a disposizione conoscenza a livello fisico di una data tecnologia, e soft core, dove Arm mette a disposizione una descrizione dettagliata che può essere sintetizzato in ogni tecnologia da un designer
- Noi useremo il cortex - M3
- I System on a chip sono sistemi integrati in un singolo pezzo di silicone e sono composti da embedded cores
- Catena di strumenti (Tool chain):



- Studiamo la Landtiger board

Introduction

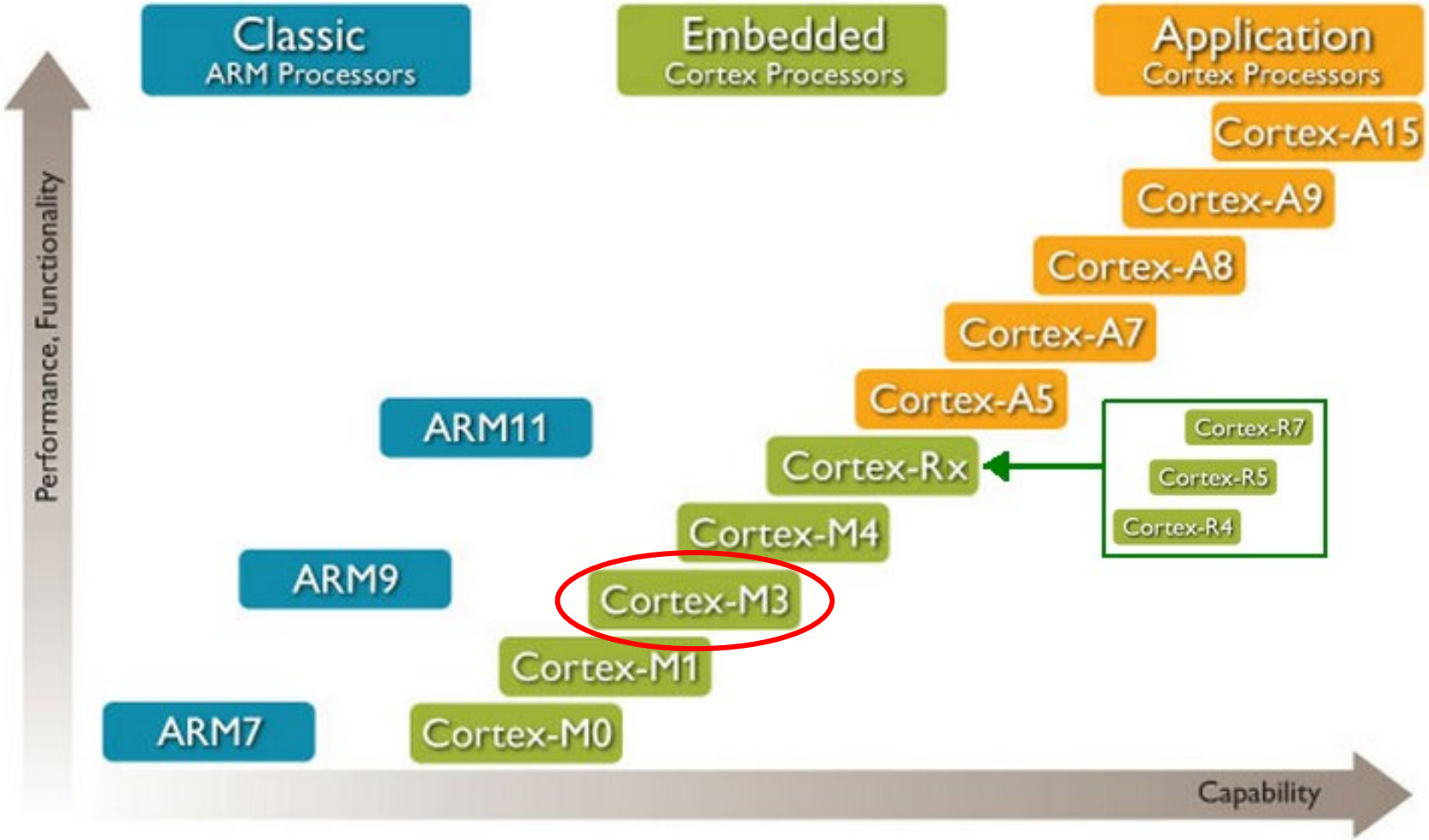
- The ARM processor was first developed (between 1983 and 1985) by Acorn Computers, Ltd., based in Cambridge (UK).
- ARM designers were heavily influenced by Berkeley RISC I.
- In 1990, ARM Ltd. was founded by Acorn, Apple and VLSI.
- Several versions of ARM processors were designed in the following years.
- Today, ARM cores are widely popular among SoC designers, mainly because they show a very good tradeoff between performance and power consumption.
- ARM does *not* manufacture silicon
- More information about ARM on the web site:
  - http://www.arm.com/aboutarm/

ARM Offices Worldwide

ARM processors

- They are mainly sold as cores, to be used for integration in Systems on Chip (SoCs).
- Cores can be
  - *Hard cores*: ARM provides a physical layout implemented in a given technology
  - *Soft cores*: ARM provides a high-level description that can be then synthesized to any technology by the designer.
- In a few cases, ARM processors have been delivered as stand-alone devices.

ARM Processors




Silicon Partners


Design Support Partners


Software, Training and Consortia Partners

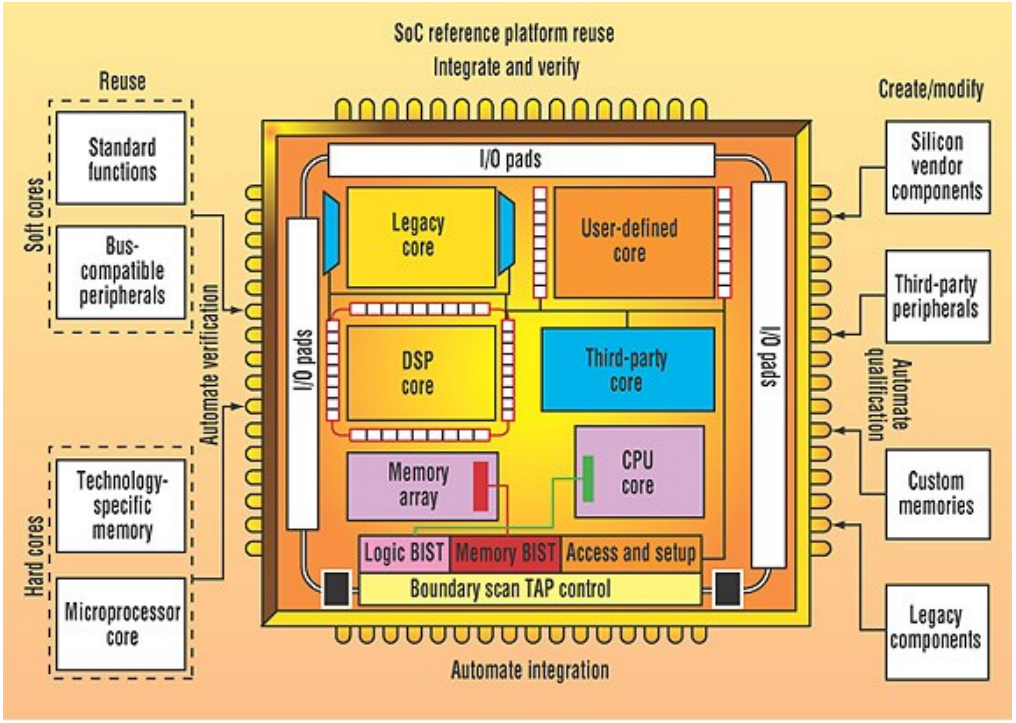
ARM Powered products

ARM world

- ARM architecture embedded in System-on-chip (SoC)
- ARM Operating Systems
- ARM Compile – Support – Debug tools

System-on-Chip (SoC)

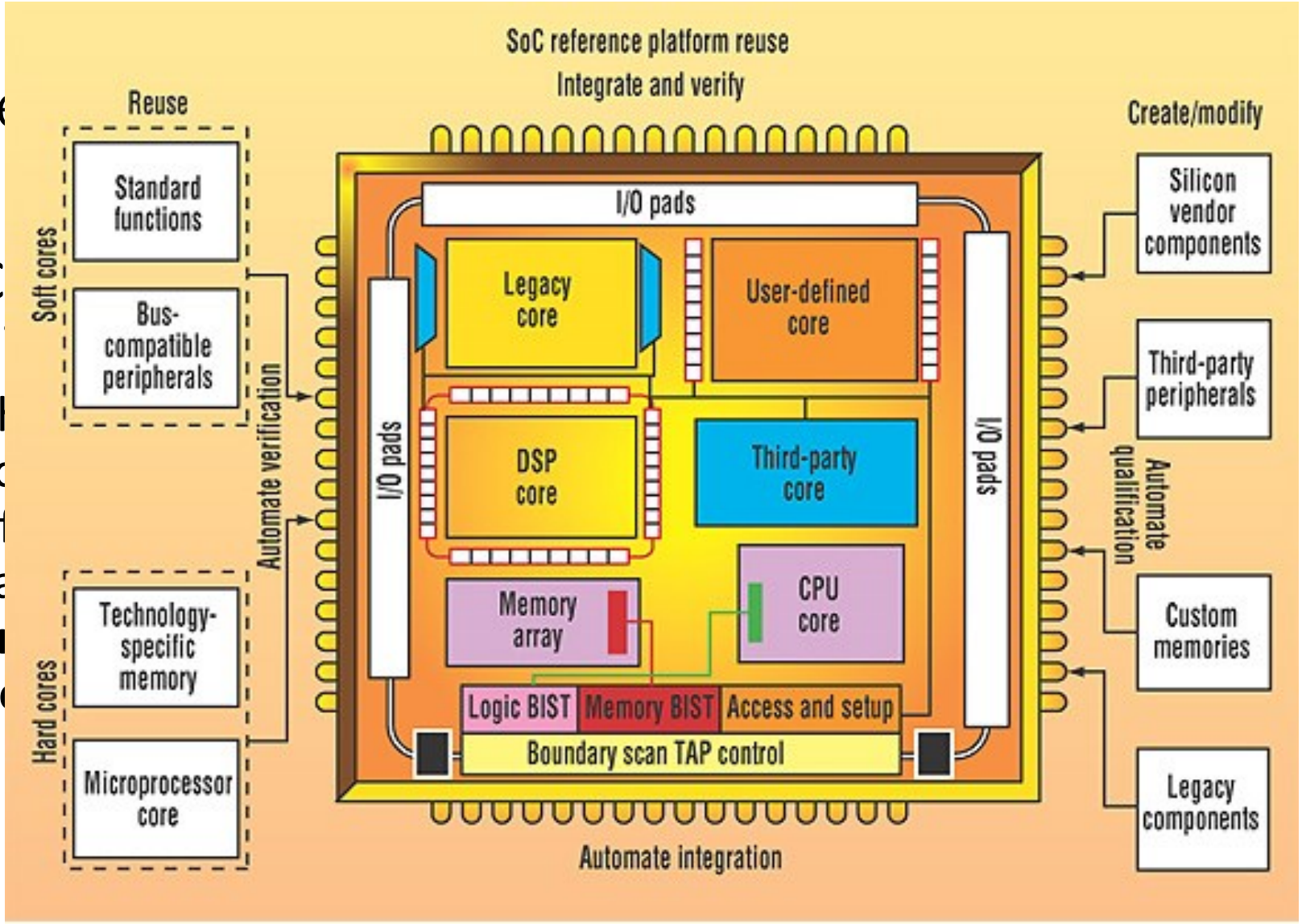
- SoCs are entire systems integrated in a single piece of silicon
  - They are composed of modules called **Embedded Cores**.



1. This diagram shows a usual SoC derivative built from a reuse platform in which over 70% of the design content could come from reuse.

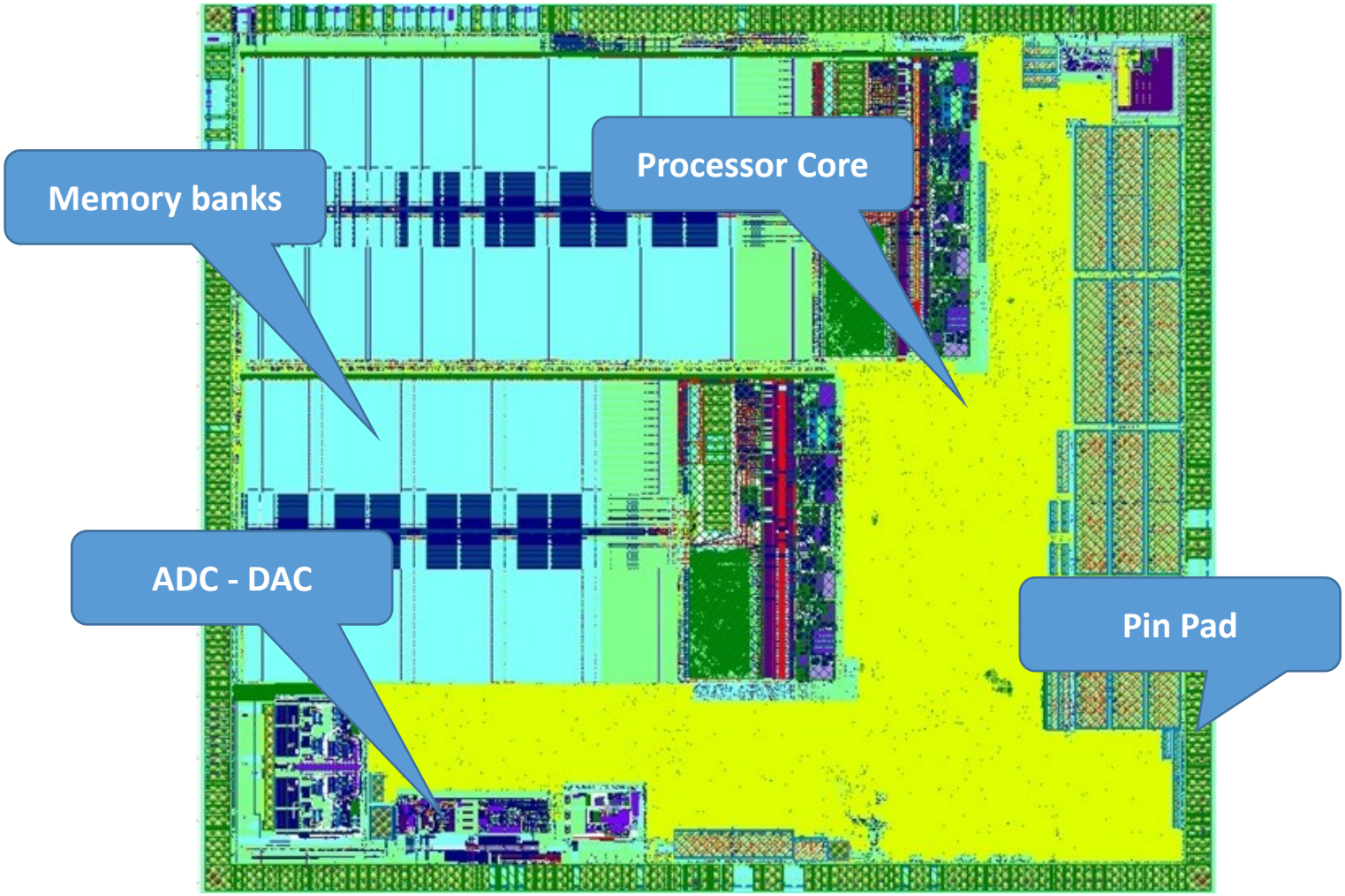
Syste

- SoC
  of s
  - Th
    co
    of
    ca
    Ex
    Co



1. This diagram shows a usual SoC derivative built from a reuse platform in which over 70% of the design content could come from reuse.

SoC layout example



ARM-based commercial SoCs

- *SAMSUNG:* http://www.samsung.com/global/business/semiconductor/products/mobilesoc/Products_ApplicationProcessor.html
  - http://pdf.datasheetcatalog.com/datasheet2/e/0lrp9fdj0zyd6e2k2e8ej8lkzupy.pdf (page 35)
- *NXP:* http://www.standardics.nxp.com/microcontrollers/
  - http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf (page 6)
- *STMicroelectronics:* http://www.st.com/mcu/
  - http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00067905.pdf?s_searchtype=keyword (page 8)
  - http://www.st.com/mcu/contentid-34-86-STR710_EVAL.html
- …and many others…

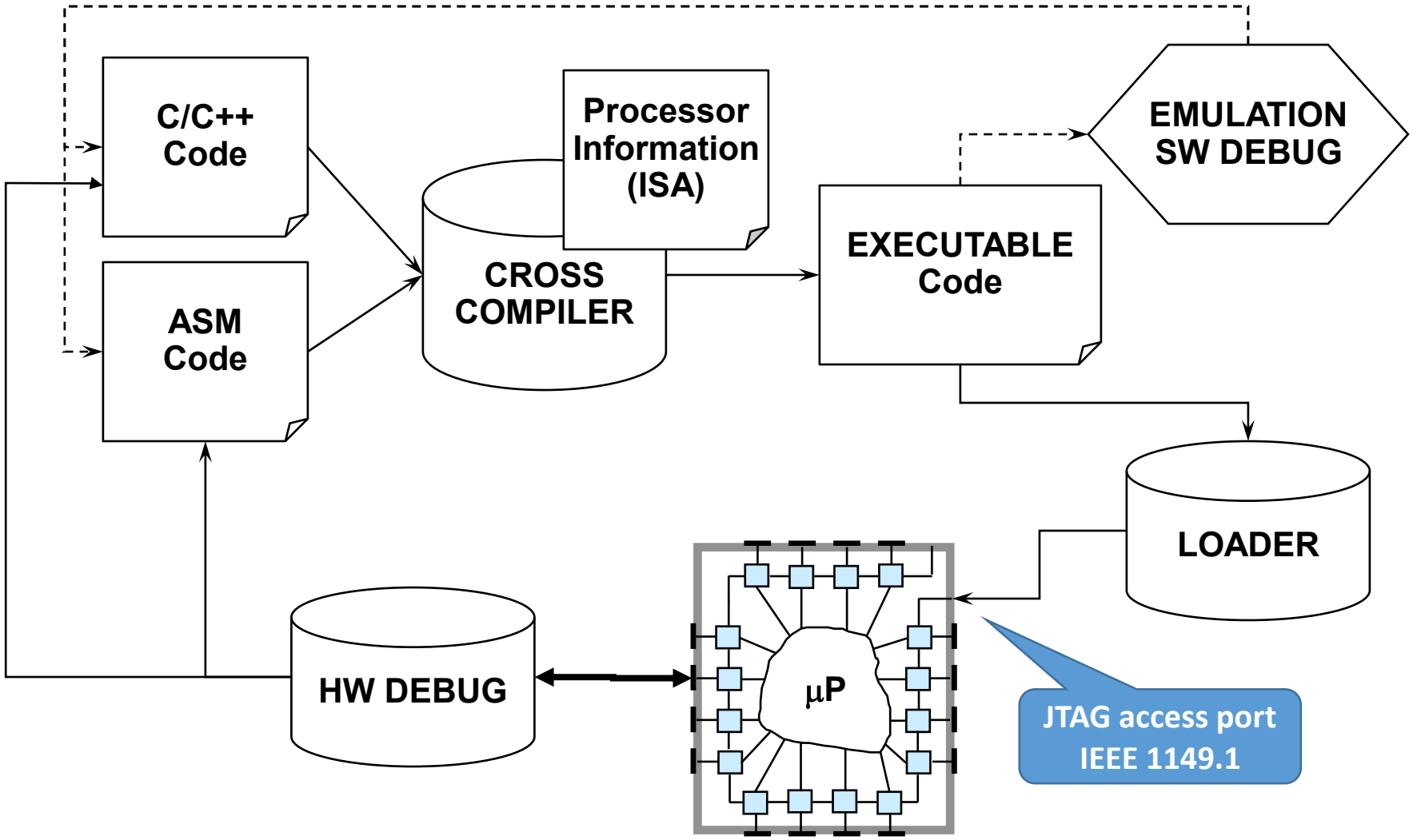
ARM compliant Operating Systems

- Microsoft Windows CE: http://www.microsoft.com/presspass/press/2002/sep02/09-18armsummitpr.mspx (old news removed)
- Linux: *many releases*
  - http://www.debian.org/ports/arm/
  - plenty of kernel to be customized
  - WIKI for problem solving

All of them requires a *bootloader* to be launched

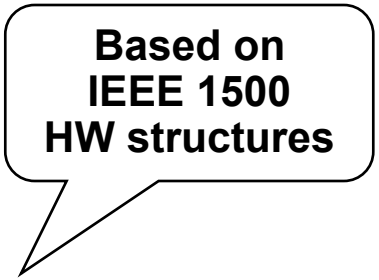
Das U-Boot: http://sourceforge.net/projects/u-boot/

Tool chain


C/C++
Code
ASM
Code
Processor
Information
(ISA)
CROSS
COMPILER
EXECUTABLE
Code
EMULATION
SW DEBUG
LOADER
HW DEBUG
µP
JTAG access port
IEEE 1149.1

ARM Tool chain

- CROSS-COMPILATION/EMULATION/SW DEBUG
  - WINDOWS: http://www.keil.com/
  - LINUX: http://www.codesourcery.com/gnu_toolchains/arm
- LOADING TOOLS
  - Ad-hoc tools released with products:
    - http://www.keil.com/
  - Generic and customizable tools
    - OPENWINCE: http://openwince.sourceforge.net/
- HW DEBUG TOOLS
  - Based on internal debug structures such as ***Embedded ICE***
    - http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0201a/index.html



What do we learn in this part of the course

- ARM assembly principles
  - Instruction Set Architecture
  - C + ASM programming by following ABI standards
  - System-on-Chip level programming including
    - Peripheral management
    - clock and power modes management
- Internal, SW and HW interrupts management
  - Exceptions due to unexpected execution flaws
  - SW interrupts towards system call understanding
  - HW interruptions
    - Possible sources of hw interrupt including internal modules (i.e., timers) and external evants (i.e., button pression)
    - Interrupt controller behavior
- Extended system on-board features.

Case of study

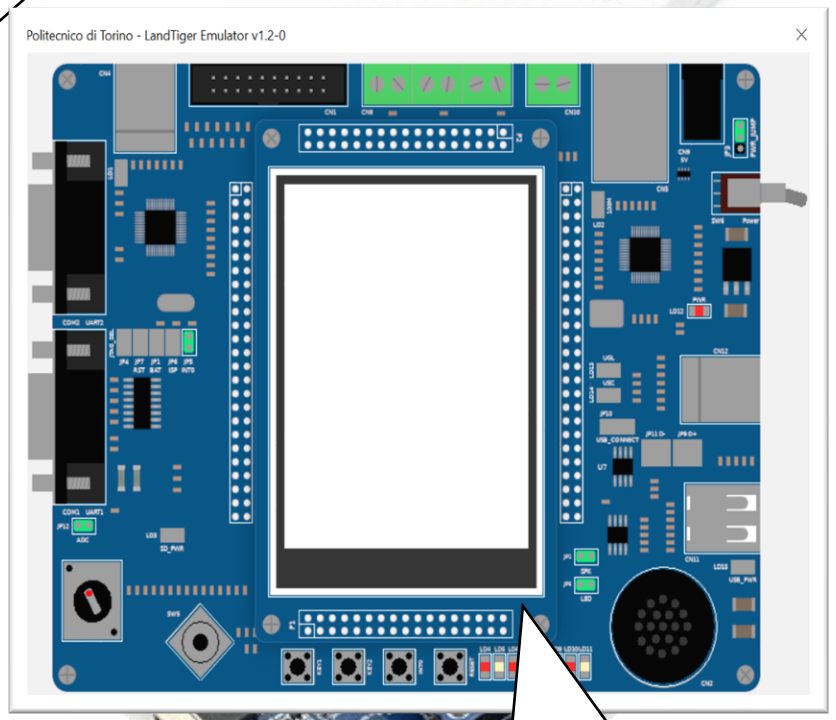
ARM V7-M Architecture

- **Landtiger board**
  - Based on a NXP system-on-chip **LPC1768** including a ARM 32-bit Cortex-M3 Microcontroller with a full set of on-chip peripheral cores
  - Mounting several additional devices and connectors on board
- KEIL uVision software
  - Trial version with 32K code limitation
  - Full use of the debugging features
  - Very accurate timing calculation
- HW debug enabled by an additional component called real-view, which implements a ULINK2 jtag based connection.

Case of study

- **Landtiger board**
  - Based on a NXP system-on-chip **LPC1768** including a ARM 32-bit Cortex-M3 Microcontroller with a full set of on-chip peripheral cores
  - Mounting several additional devices and connectors on board
- KEIL uVision software
  - Trial version with 32K code limitation
  - Full use of the debugging features
  - Very accurate timing calculation
- HW debug enabled by an additional component called real-view, which implements a ULINK2 jtag based connection.

ARM V7-M Architecture



Board Emulation system