

Introduction to ARM

Matteo SONZA REORDA
Dip. Automatica e Informatica
Politecnico di Torino

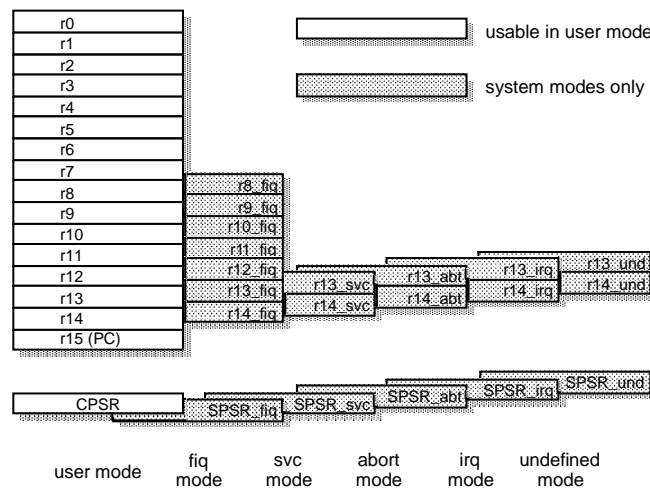
Some history

- ARM was originally developed by Acorn Computers Ltd, Cambridge, UK, between 1983 and 1985
- In 1990, ARM Ltd. became a separate company aimed at exploiting the ARM technology
- Today, ARM is a worldwide leader in low-power and cost-sensitive embedded applications

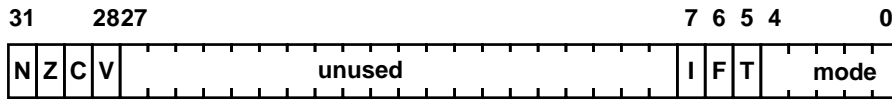
ARM and Berkeley RISC

- ARM has been one of the first commercial products based on the RISC philosophy
- ARM relies on the Berkeley RISC architecture
- ARM borrows from Berkeley RISC
 - Load-store architecture
 - Fixed-length 32-bit instructions
 - 3-address instruction formats
- A major characteristic of ARM is the design simplicity

The ARM programmer's model

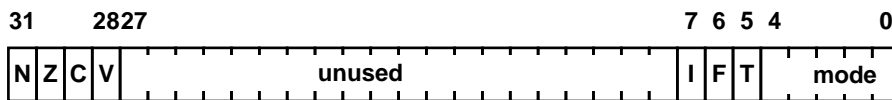


CPRS



CPRS stands for *Current Program Status Register*.

CPRS



Condition codes:
• Negative
• Zero
• Carry
• Overflow

Affect some processor features

Shows the processor operation mode

The memory system

- The ARM addressing space is 2^{32} byte wide
- Data items may be
 - 8-bit wide
 - 16-bit wide (aligned on 2-byte boundaries)
 - 32-bit wide (aligned on 4-byte boundaries)
- ARM adopts the *little-endian* organization, but may also support the *big-endian* one

Supervisor mode

- A protected supervisor mode is supported
- System-level functions running in protected mode only can be accessed through specified supervisor calls

The ARM instruction set

- General characteristics:
 - Load-store architecture
 - 3-address data processing instructions
 - Conditional execution of every instruction
 - Load and store multiple register instructions
 - Ability to perform a shift operation and an ALU operation in the same instruction
 - Possible instruction set extension by adding a coprocessor
 - Additional Thumb instruction set

I/O system

- ARM manages I/O devices through the memory-mapped scheme
- Two interrupt input signals
 - Normal interrupt (IRQ)
 - Fast interrupt (FIQ)

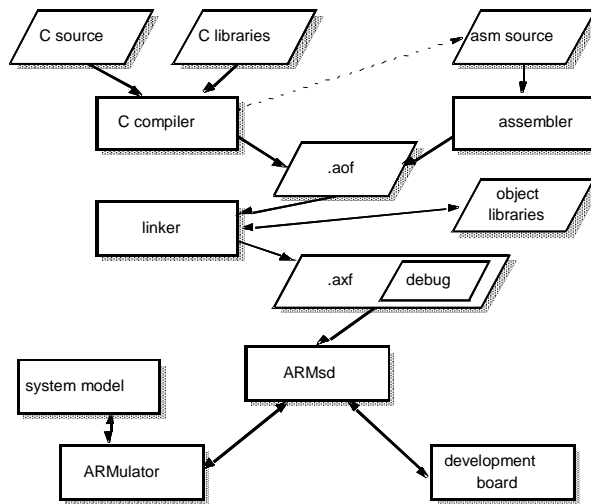
ARM exceptions

- They include
 - Interrupts
 - Traps
 - Supervisor calls
- They are all managed in the same way:
 - Saving the PC into `r14_exc` and the CPSR into `SPSR_exc` (`exc` depends on the exception type)
 - Changing the operating mode to the appropriate exception mode
 - Forcing the PC to a value between 00_{16} to $1C_{16}$ (which contain the *vector address*)

ARM development tools

- Their availability is crucial to the success of the ARM processors
- They may run either under the Windows or the UNIX OS

ARM cross-development toolkit



M. Sonza Reorda

Politecnico di Torino

13