MANCHEstER 1824

The University of Manchester

ARM System Design

- Aim: to introduce
  - ARM-based embedded system design
  - the ARM and Thumb instruction sets
    - including hands-on programming sessions
  - the ARM software development toolkit
    - used in the hands-on sessions
  - the ARM hardware interface

MANCHEstER 1824

The University of Manchester

What is “ARM”?

- A developing series of microprocessor architectures
  - a 32-bit ‘RISC’ (Reduced Instruction Set Computer)
    - initially quite simple in design
    - has evolved over ~25 years
    - no longer really “reduced”!
- The company which develops it:
  - ARM Ltd, Cambridge, UK

MANCHEstER 1824

The University of Manchester

ARM versions

ARM core	Instruction set version	Operations
ARM7	4	Basic instruction set
ARM7TDMI	4T	v4 plus Thumb support
ARM9TDMI	4T	v4 plus Thumb support
ARM10T	5T	v4T plus: BKPT, BLX, CLZ, extra copro. ops.
ARM10E	5TE	v5T plus: LDRD/STRD, double word copro. moves, PLD, some signal processing extensions
ARM10EJ	5TEJ	v5TE plus Jazelle support
ARM11J	6	v5TEJ plus: more PSR/context switching ops, LDREX/STREX semaphore operations, 16-bit ‘SIMD’ operations

Thumb
Java acceleration
-Synthesized core

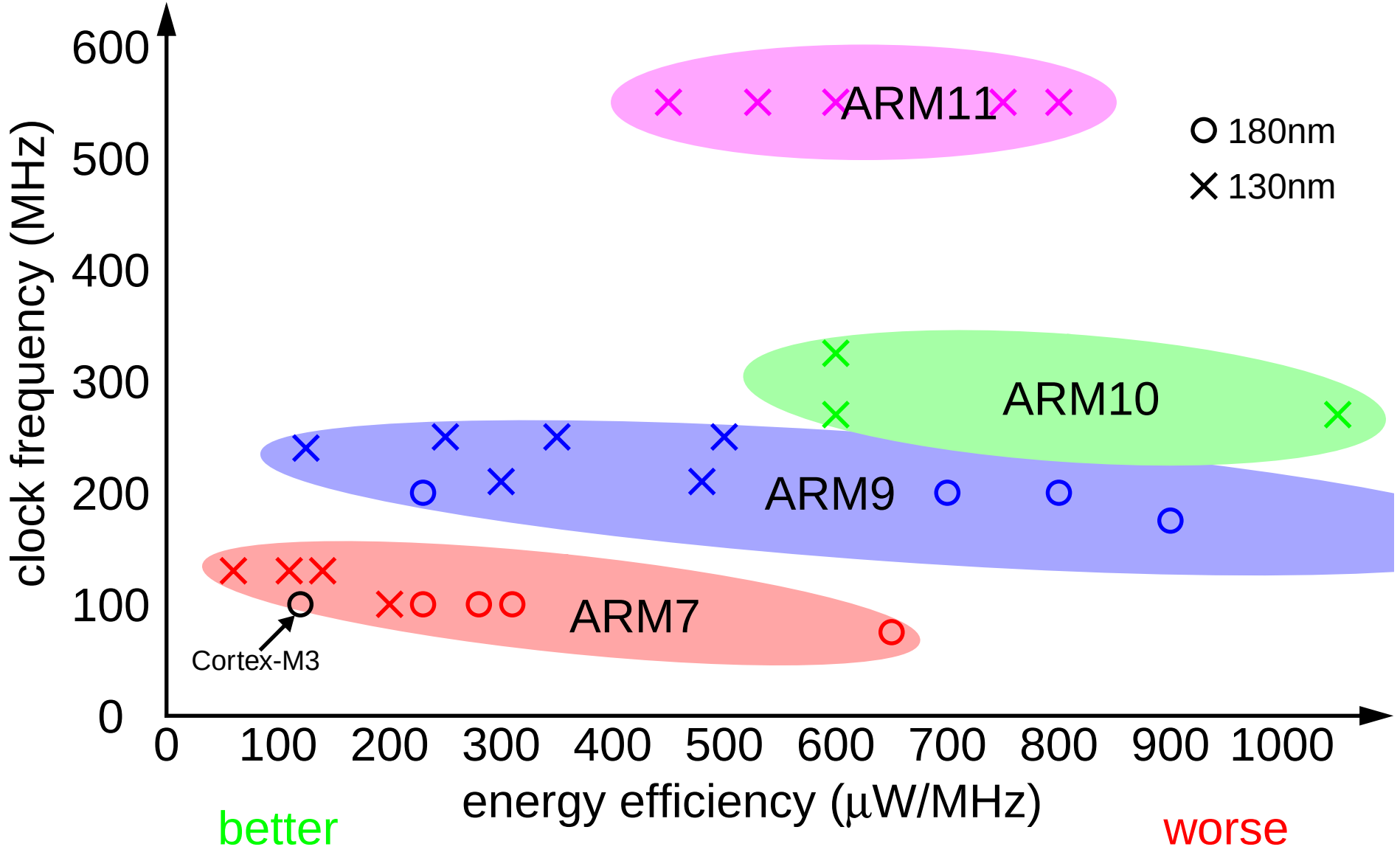
Debug
DSP **E**xtensions

long **M**ultiplies
Floating point copro.

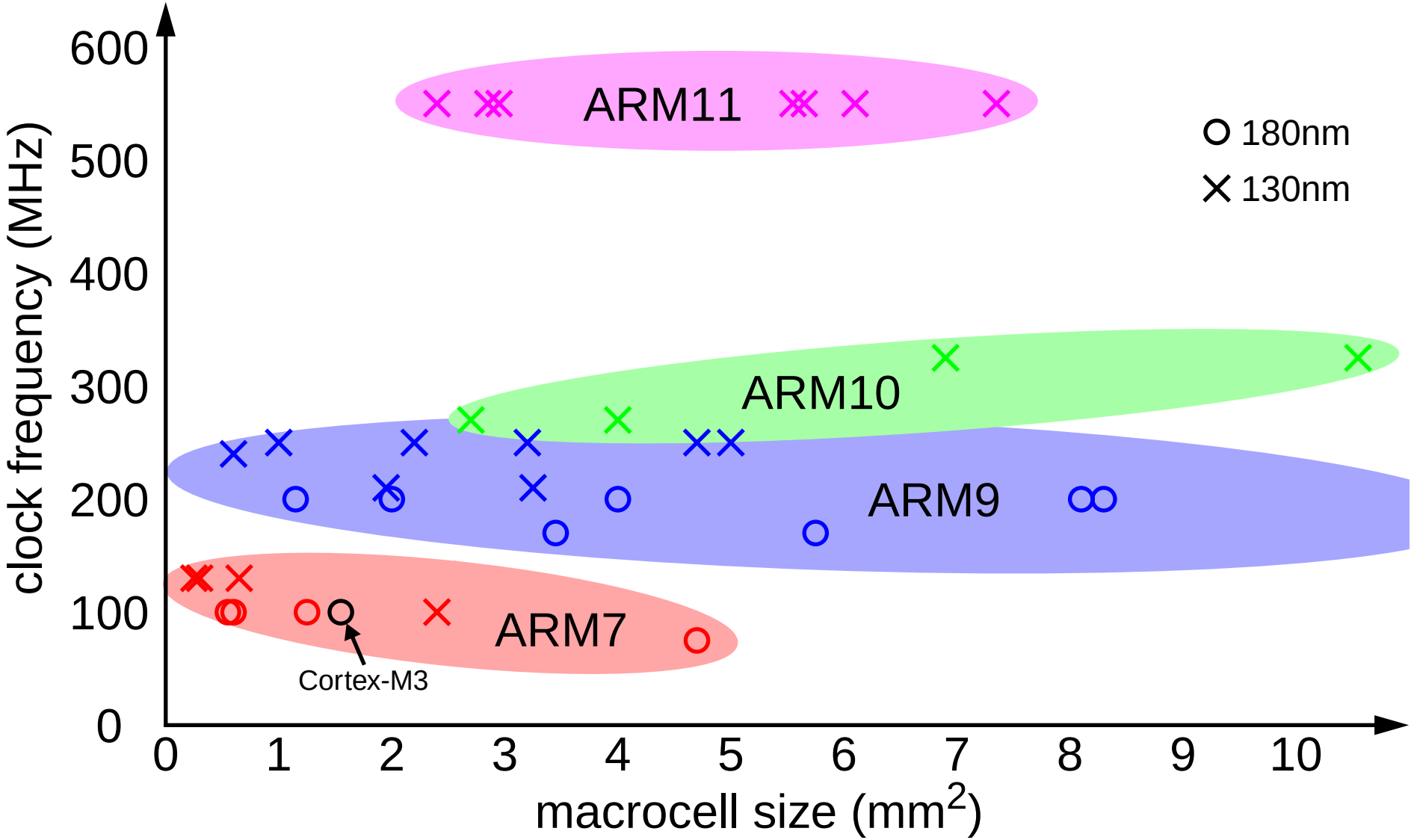
Embedded**I**CE
Trust**Z**one

Earlier architectures are obsolete and not discussed here.

ARM performance



ARM size


ARM11
ARM10
ARM9
ARM7
Cortex-M3
180nm
130nm
clock frequency (MHz)
macrocell size (mm2)
0
100
200
300
400
500
600
0
1
2
3
4
5
6
7
8
9
10

MANCHEstER 1824

The University of Manchester

Course structure

- This course covers both software and hardware
  - Software – first half
    - mainly architecture and instruction set
  - Hardware – second half
    - primarily the programmer's view
- Structure
  - interspersed presentations and practical sessions
- Please treat this informally!
  - tell us what *you* want out the course

MANCHEstER 1824

The University of Manchester

Schedule

- Day 1
  - The ARM software development toolkit
    - hands-on: exploring the toolkit
  - ARM assembly language programming
    - hands-on: writing simple assembly programs
  - Support for high-level languages
    - hands-on: C programs and debugging
  - The ARM instruction set in detail
    - hands-on: system software - SWI handler

MANCHEstER 1824

The University of Manchester

Schedule

- Day 2
  - The Thumb instruction set
    - hands-on: writing Thumb assembly
  - Architectural extensions
    - hands-on: Thumb C and cycle counts
  - ARM integer cores
    - hands-on: system software - interrupts
  - Coprocessors
    - hands-on: system software - semaphores

MANCHEstER 1824

The University of Manchester

Schedule

- Day 3
  - Memory hierarchy
    - hands-on: interfacing C and assembly code
  - Support for memory hierarchy
    - hands-on: memory protection
  - ARM CPUs
    - hands-on: code profiling
  - System development
    - hands-on: system modelling with ARMulator

MANCHEstER 1824

The University of Manchester

Toolkit introduction

- Outline:
  - ➜ **the ARM programmers' model**
  - ❍ the ARM software development tools
  - ☞ hands-on: introduction to the software development tools

The ARM programmers' model

- ARM is a Reduced Instruction Set Computer (RISC); it has:
  - a large, regular register file
    - any register can be used for any purpose
  - a load-store architecture
    - instructions which reference memory just move data, they do no processing
    - processing uses values in registers only
  - fixed-length 32-bit instructions

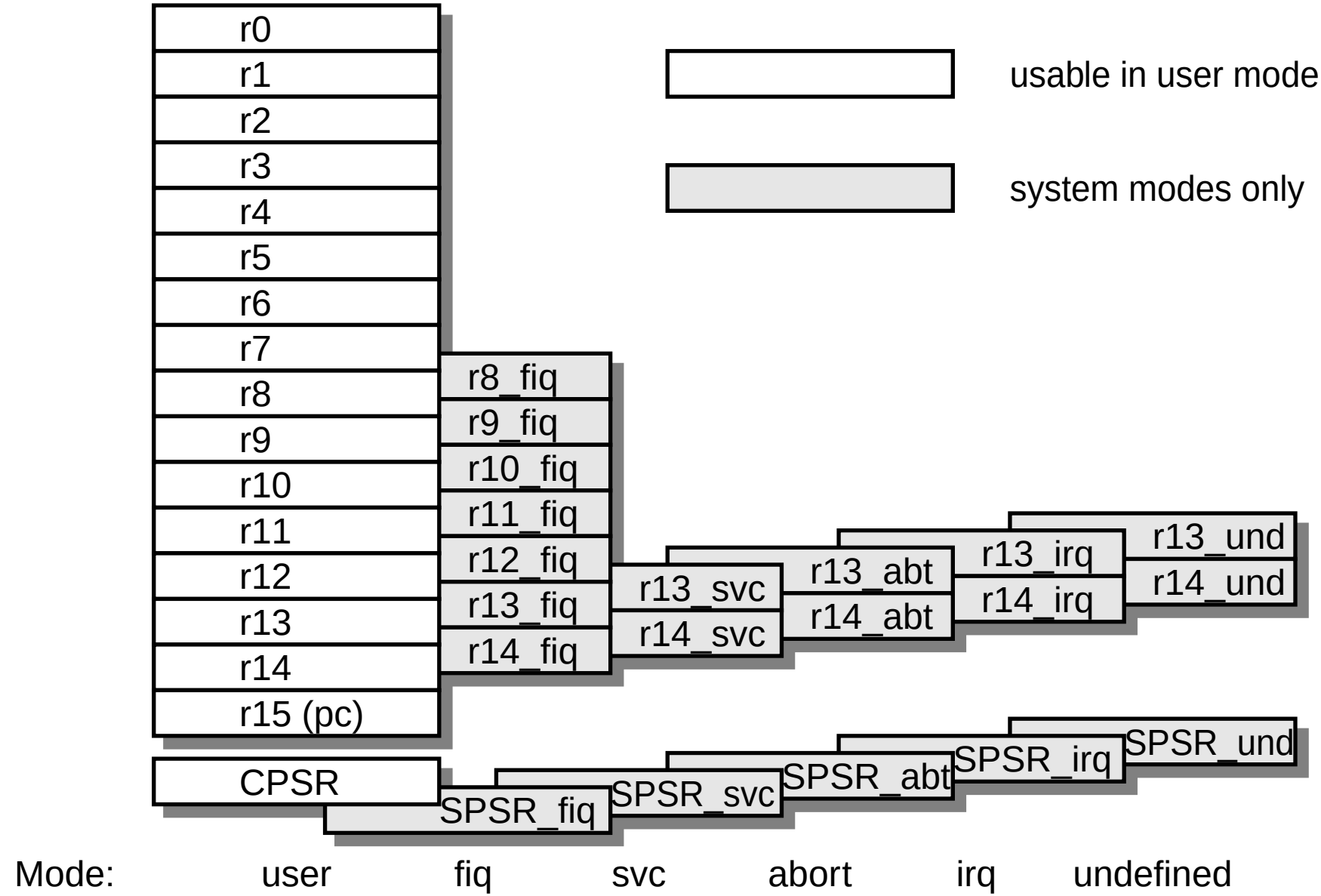
MANCHEstER 1824

The University of Manchester

The ARM programmers' model

- The user has sixteen 32-bit registers (R0-R15)
  - all register operations are 32-bit
  - memory transfers may be smaller but loads always extend
- R15 acts as the program counter (a.k.a. "PC")
- A status register (CPSR) holds some extra bits

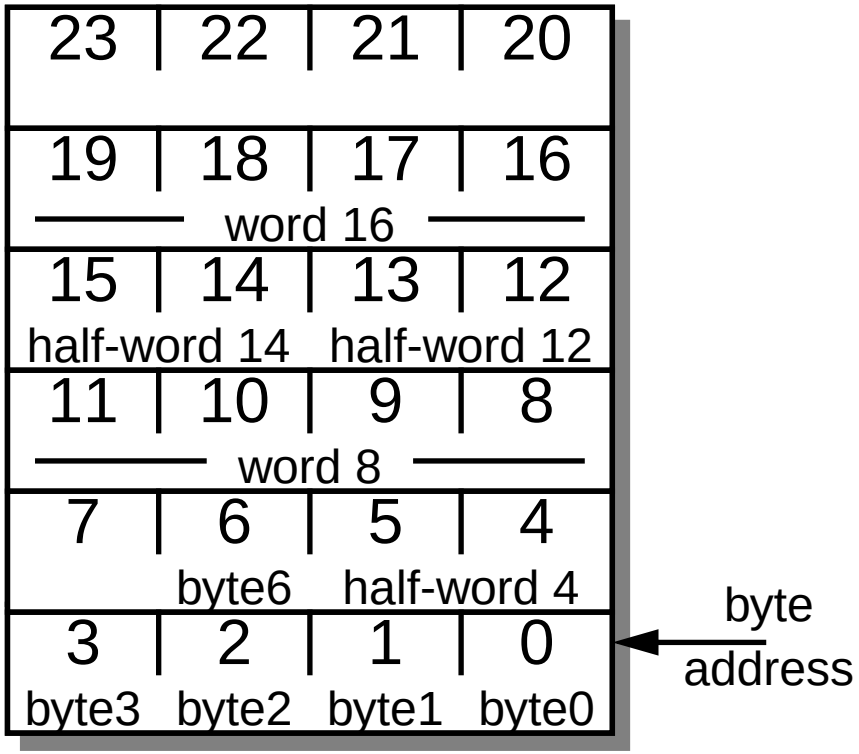
ARM register organization



ARM CPSR format

- In user programs only the top 4 bits of the CPSR are significant:
  - N - the result was negative
  - Z - the result was zero
  - C - the result produced a carry out
  - V - the result generated an arithmetic overflow

ARM memory organization



- Memory is a linear array of 2^{32} byte locations.
- ARM can address:
  - individual bytes
  - 32-bit words on 4-byte boundaries
  - 16-bit half-words on 2- byte boundaries

MANCHEstER 1824

The University of Manchester

Toolkit introduction

- Outline:
  - the ARM programmers' model
  - ➜ **the ARM software development tools**

  - ☞ hands-on: introduction to the software development tools

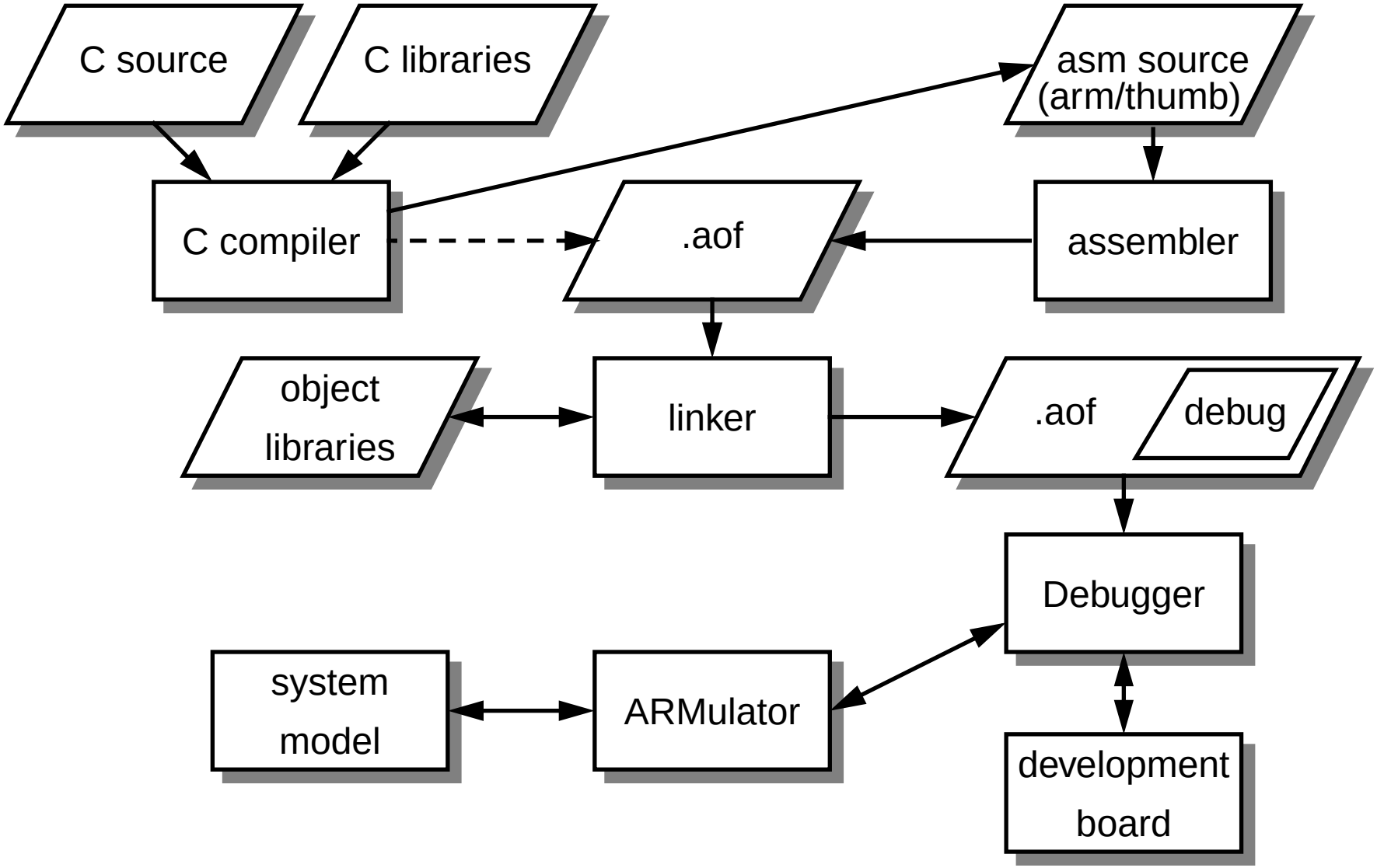
ARM software development tools

- Even experienced programmers approach a new environment by first getting a simple program to run
  - often a ‘Hello World’ program
- This requires some basic tools:
  - a text editor, to enter the program
  - an assembler to produce binary code
  - a system or emulator to test the code

ARM software development tools

- Code generation tools
  - C and Embedded C++ compilers
  - Assembler and Linker for ARM and Thumb instruction sets
- Debuggers
  - ARMsd (symbolic debugger), AXD, RVD
- Debug Targets
  - Simulation software: ARMulator
  - Hardware: RealView Ice, Multi-Ice, RV Trace, MultiTrace, Angel
- Project manager

The structure of the ARM cross development toolkit



MANCHEstER 1824

The University of Manchester

ARM Developer Suite

- ADS v1.2
  - ADS compilation tools
  - Metrowerks CodeWarrior IDE (Windows version only)
  - AXD debugger v1.2
  - command-line ARM symbolic debugger (armsd)
  - ARMulator
  - Real-time Debug and Trace support
  - Support for families of processors including ARM7, ARM9, ARM9E, ARM10, StrongARM and Intel XScale

ARM RealView Developer Suite

- RVDS v2.2
  - RVCT compilation tools
  - Metrowerks CodeWarrior 5.6 IDE (Windows only)
  - AXD debugger v1.3.1
  - RVD debugger featuring:
    - OS awareness
    - Multi-core and DSP Awareness (options)
  - ARM symbolic debugger (armsd)
  - RVISS instruction set simulator
  - Support for all families of ARM processors

3rd party tools

- GNU tools chain for ARM
  - free or embedded in commercial packages (e.g. Microcross / Nohau)
- Keil Software development tools
  - ARM7 TDMI
- Green Hills Software
  - support for all (?) families of ARM processors

MANCHEstER 1824

The University of Manchester

Hands-on: introduction to the ARM project manager

- Get introduced to the ARM software development tools
  - Build a simple ‘project’
  - Check that it works
  - Investigate other facilities of the toolkit

☞ Follow the ‘Hands-on’ instructions