The ARM processor

ARM - Acorn Risc Machine

ARM is a RISC processor with the following features

- Load-store architecture
- 16 registers, each 32-bits wide
- Three-operand instructions, 32-bits wide
- At each clock cycle a new instruction is issued
- Low energy consumption (W/MIPS)

It is designed by ARM Ltd. and licensed to manufacturers

- ARM7: nommu, very widespread as a microcontroller
- DEC, then Intel: StrongARM (now dead)
- Intel, then Marvell: Xscale: 80200, PXA255, PXA270, IXP425, ...
- Cirrus Logic: EP93xx: 9302, 9315, ...
- Freescale (now NXP): iMX1, iMX21, iMX27, iMX31, ..., iMX7, ...
- Atmel: AT91SAM: many flavours
- ... many more ...

http://en.wikipedia.org/wiki/ARM_architecture

ARM Machine Code (32bit: not Cortex-M)

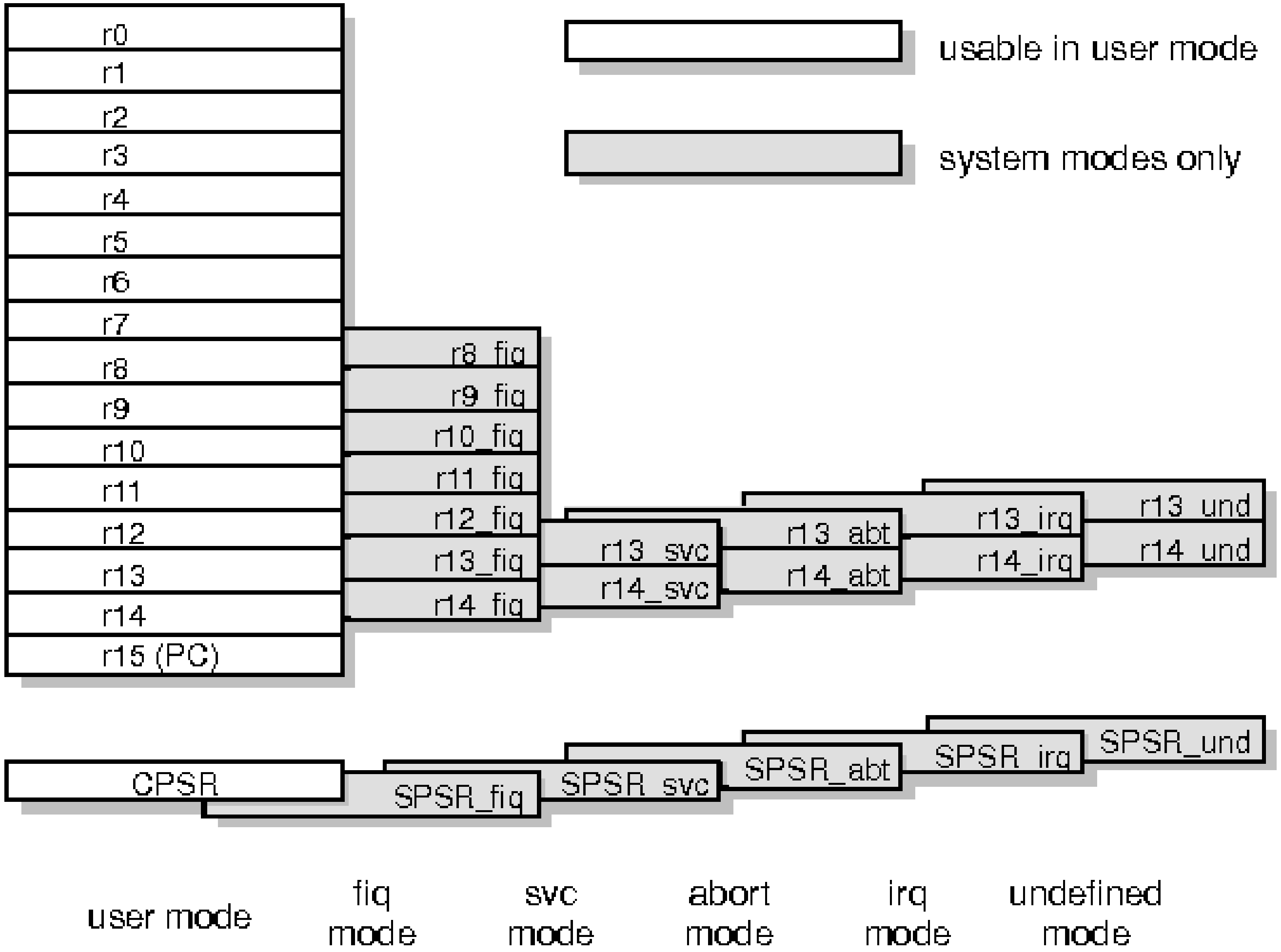
Major features of the ARM machine code

- Unaligned access is not allowed (like all RISC)
- No assignment of 32-bit constants is possible (like all RISC)
- There is no hardware-managed stack pointer (like all RISC)
- Load-multiple and store-multiple instructions
- Every instruction is conditionally executed
- The status bits are modified only optionally
- One operand can be shifted at no cost
- All addressing is register-relative

Other features typical of RISC processors, but missing in ARM:

- Register windows
- «Delay slot» for jumps
- Zero register

ARM Registers



usable in user mode

system modes only

r0
r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
r11
r12
r13
r14
r15 (PC)

r8_fiq
r9_fiq
r10_fiq
r11_fiq
r12_fiq
r13_fiq
r14_fiq

r13_svc
r14_svc

r13_abt
r14_abt

r13_irq
r14_irq

r13_und
r14_und

CPSR

SPSR_fiq

SPSR_svc

SPSR_abt

SPSR_irq

SPSR_und

user mode

fiq mode

svc mode

abort mode

irq mode

undefined mode

The Standard ABI, Coprocessors

r0	a1
r1	a2
r2	a3
r3	a4
r4	v1
r5	v2
r6	v3
r7	v4
r8	v5
r9	v6
r10	v7
r11	v8
r12	ip
r13	sp
r14	lr
r15	pc

Role of registers

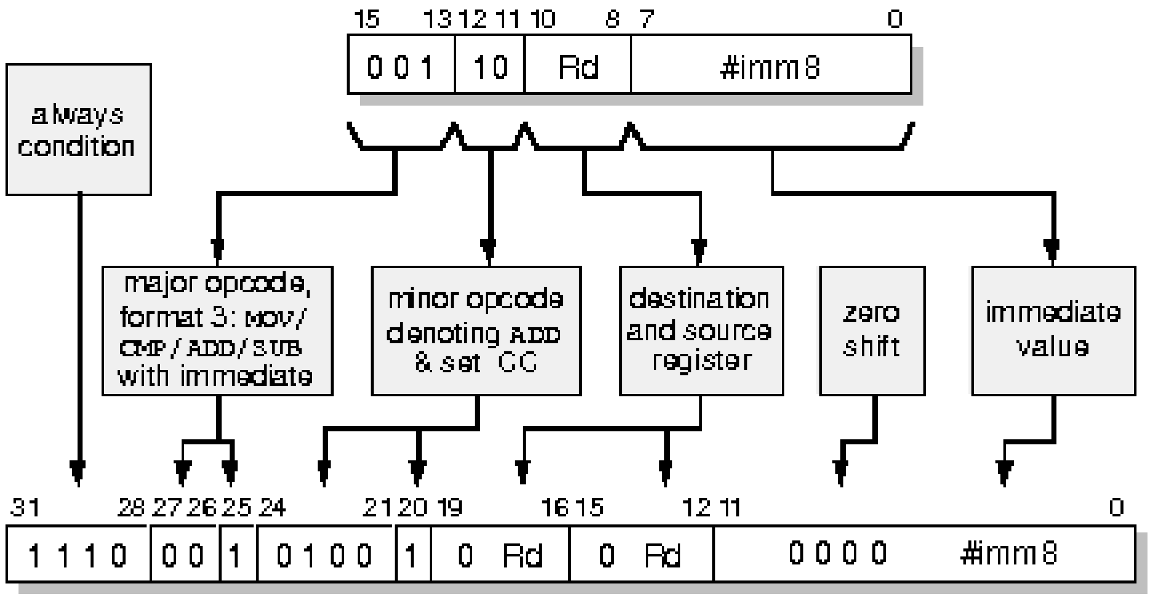
- A registers are function arguments (caller-saved)
- V registers are callee-saved
- R12 is the "intra-procedure scratch"
- R13 is the stack pointer (software defined)
- R14 is the link register (hardware defined)
- R15 is the program counter (hardware defined)

Coprocessors

- The architecture defines 16 coprocessors
  - CP15, if present, is used for cache and MMU
  - CP0 and CP1, if present, are used for FPU
- The following instructions are defined by the architecture:
  - Register move CPU/coprocessor: MRC, MCR
  - Coprocessor load and store: LDC, STC
  - Coprocessor data processing: CDP

The Thumb Extension (Thumb-1)

Thumb instructions are 16-bits wide



- Code is more compact
- Only a subset of the registers can be accessed
- 2-operand operations

BX and BLX (branch (and link) and exchange) switch mode

- Bit 0 of the program counter is used as a selector
  - ARM instructions run at even addresses
  - Thumb instructions run at odd addresses

The Thumb-2 Instruction Set

The "Cortex" (ARMv6/v7) family introduced Thumb-2

- It extended the thumb instruction set
- Thumb now accesses all registers (not only 8 of them)
- It includes coprocessor instructions
- New ITE instruction (if then else)
- It is a separate instruction decoder, not a decompressor any more

The company also changed the boot vectors

- There used to be 8 _instructions_ at address 00..1f
- Now the "VIC" is offered as part of the core
- Vectors are now pointers rather than instructions (and many of them)
- It is now possible to make thumb-only devices

The cortex-m family is a microcontroller flavour, thumb-2 only.