ARM Architecture

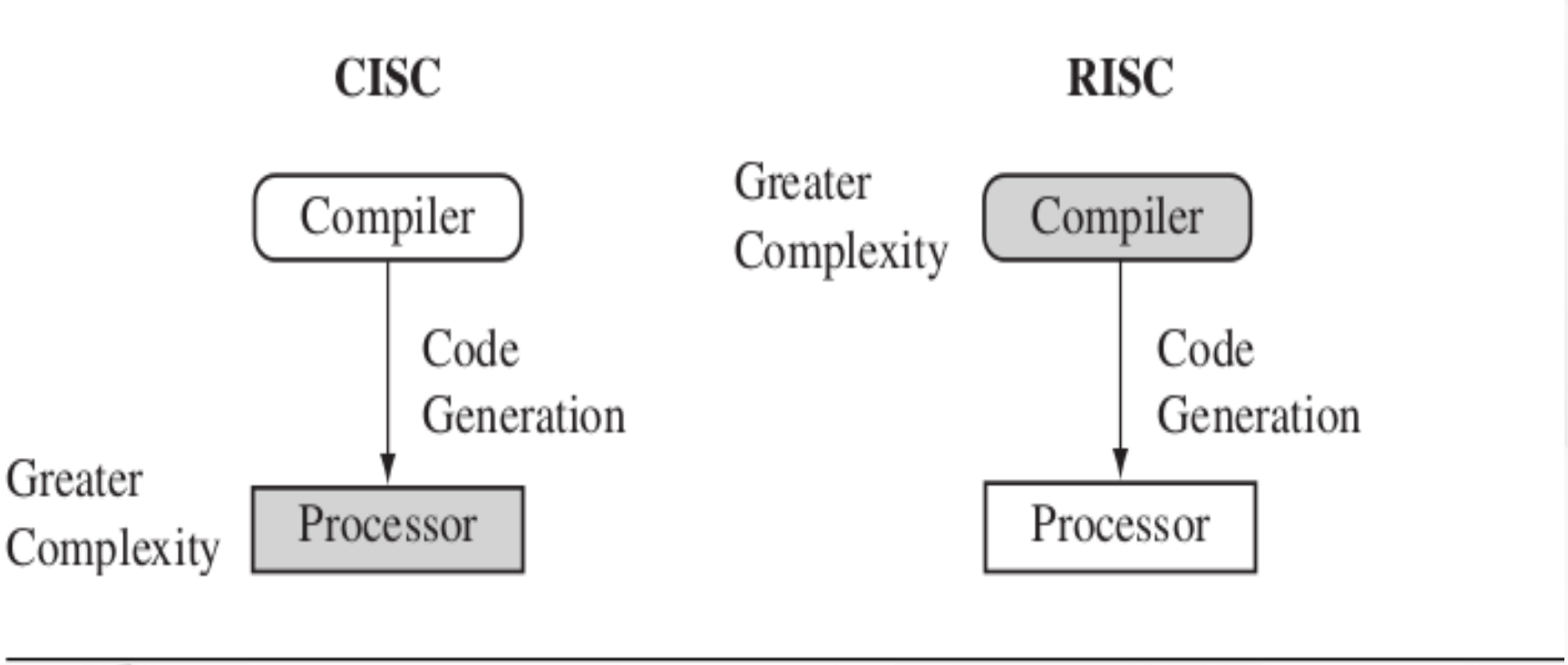
ARM Architecture

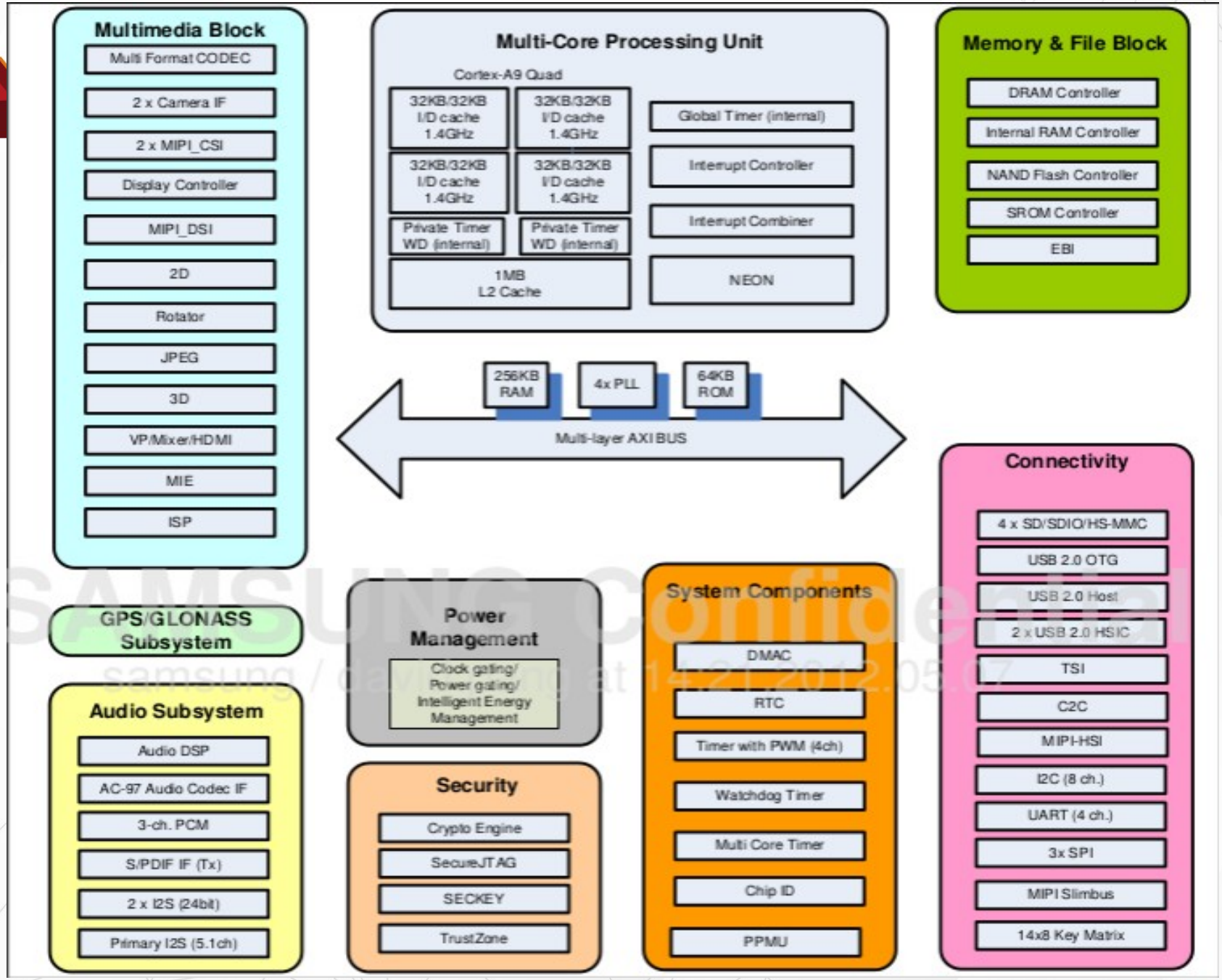
- ARM Architecture
- The Memory Hierarchy and Cache Memory
- Logical and physical caches
- Cache Architecture
- Memory Management Units

ARM Core and Processor Family Overview

ARM Processor Family		ARM Architecture	Core
Classic ARM	ARM7	ARMv3	ARM700
			ARM710
			ARM710a
	ARM11	ARMv6	ARM1136J
			ARM1156T2
			ARM1176JZ
			ARM11MPCore
Embedded	Cortex-M	ARMv6-M	Cortex-M0
			Cortex-M1
		ARMv7-M	CortexM3
		ARMv7E-M	Cortex-M4
Real-Time	Cortex-R	ARMv7-R	Cortex-R4
			Cortex-R5
			Cortex-R7
Application	Cortex-A	ARMv7-A	Cortex-A5
			Cortex-A7
			Cortex-A8
			Cortex-A9
			Cortex-A15
64-bit Core	Cortex-A50	ARMv8-A	Cortex-A53
			Cortex-A57

The RISC design philosophy





AMBA Bus Protocol

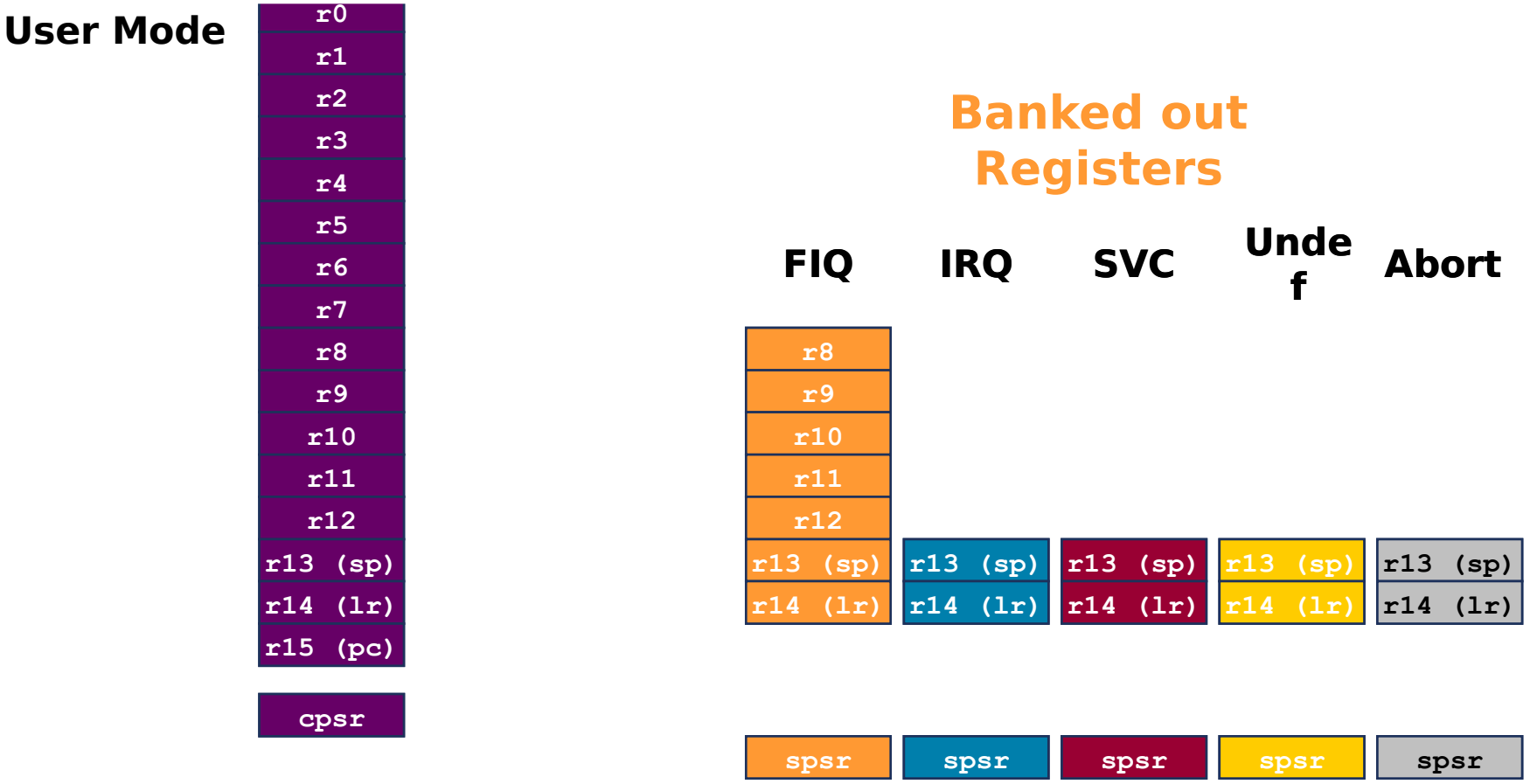
- **A**dvanced **M**icrocontroller **B**us **A**rchitecture
- ASB - ARM System Bus
- APB - ARM Peripheral Bus
- AHB - ARM High Performance Bus
- AXI - Advanced eXtensible Interface
- ACE - AXI Coherency Extensions

Processor Modes

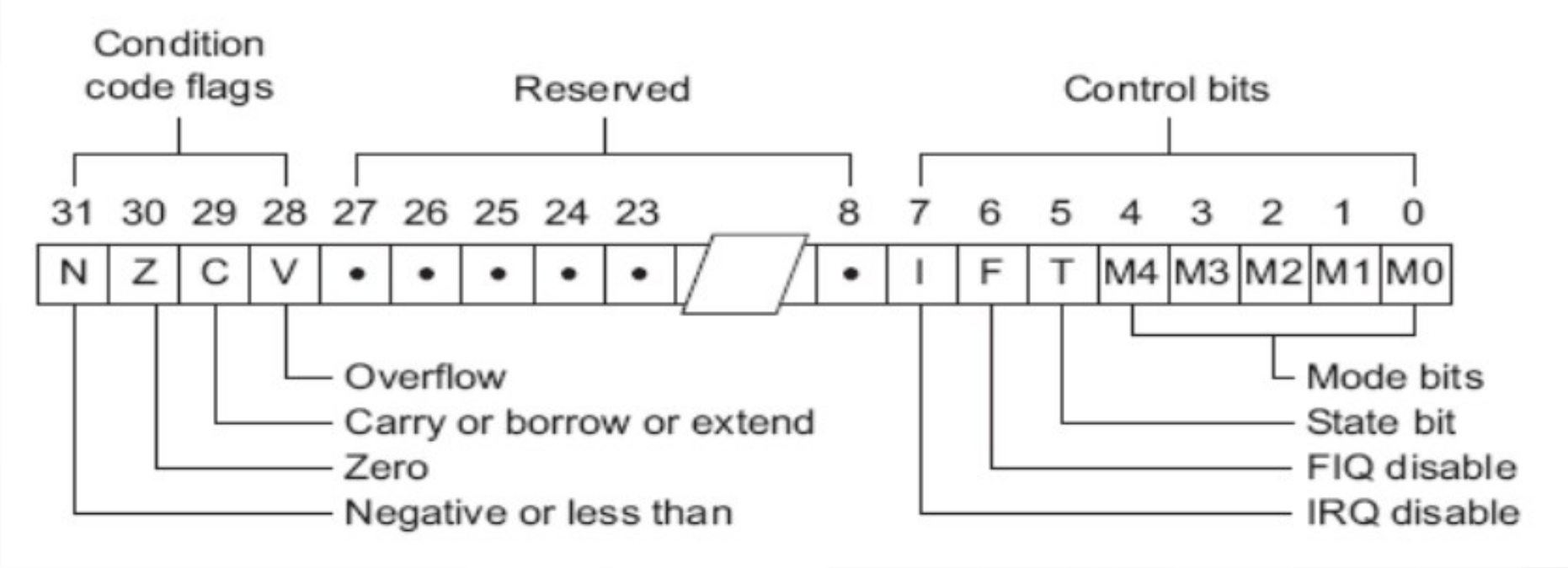
- **ARM has seven basic operating modes**
  - Each mode has access to its own stack space and a different subset of registers
  - Some operations can only be carried out in a privileged mode

	Mode	Description	
Exception modes	**Supervisor (SVC)**	Entered on reset and when a Supervisor call instruction (SVC) is executed	**Privileged modes**
	FIQ	Entered when a high priority (fast) interrupt is raised	
	IRQ	Entered when a normal priority interrupt is raised	
	Abort	Used to handle memory access violations	
	Undef	Used to handle undefined instructions	
	System	Privileged mode using the same registers as User mode	
	User	Mode under which most Applications / OS tasks run	**Unprivileged mode**

The ARM Register Set



Program Status Registers (CPSR, SPSR)



Current Processor Status Register

SPSR (Saved Processor Status Register)

Program Status Registers

- The condition code flags
  - The N, Z, C, and V bits are the condition code flags
- The T bit
  - T bit is set : processor in Thumb state
  - T bit is clean : processor in ARM state
- Mask bits
  - I bit is set, IRQ interrupts are disabled
  - F bit is set, FIQ interrupts are disabled
- The Mode bits
  - M[4:0] are the mode bits

Mode	Mode encoding in the PSRs	Function
Supervisor (SVC)	10011	Entered on reset or when a Supervisor Call instruction (SVC) is executed
FIQ	10001	Entered on a fast interrupt exception
IRQ	10010	Entered on a normal interrupt exception
Abort (ABT)	10111	Entered on a memory access violation
Undef (UND)	11011	Entered when an undefined instruction executed
System (SYS)	11111	Privileged mode, which uses the same registers as User mode
User (USR)	10000	Unprivileged mode in which most applications run

The vector table

Exception/interrupt	Shorthand	Address	High address
Reset	RESET	0x00000000	0xffff0000
Undefined instruction	UNDEF	0x00000004	0xffff0004
Software interrupt	SWI	0x00000008	0xffff0008
Prefetch abort	PABT	0x0000000c	0xffff000c
Data abort	DABT	0x00000010	0xffff0010
Reserved	—	0x00000014	0xffff0014
Interrupt request	IRQ	0x00000018	0xffff0018
Fast interrupt request	FIQ	0x0000001c	0xffff001c

Introduction to the ARM Instruction Set

Mnemonics	ARM ISA	Description
ADC	v1	add two 32-bit values and carry
ADD	v1	add two 32-bit values
AND	v1	logical bitwise AND of two 32-bit values
B	v1	branch relative +/− 32 MB
BIC	v1	logical bit clear (AND NOT) of two 32-bit values
BKPT	v5	breakpoint instructions
BL	v1	relative branch with link
BLX	v5	branch with link and exchange
BX	v4T	branch with exchange
CDP CDP2	v2 v5	coprocessor data processing operation
CLZ	v5	count leading zeros
CMN	v1	compare negative two 32-bit values
CMP	v1	compare two 32-bit values
EOR	v1	logical exclusive OR of two 32-bit values
LDC LDC2	v2 v5	load to coprocessor single or multiple 32-bit values
LDM	v1	load multiple 32-bit words from memory to ARM registers
LDR	v1 v4 v5E	load a single value from a virtual address in memory
MCR MCR2 MCRR	v2 v5 v5E	move to coprocessor from an ARM register or registers

Introduction to the ARM Instruction Set

Instruction Syntax	Destination register (*Rd*)	Source register 1 (*Rn*)	Source register 2 (*Rm*)
ADD r3, r1, r2	*r3*	*r1*	*r2*

Data Processing Instructions

```
Syntax: <instruction>{<cond>}{S} Rd, N
```

MOV	Move a 32-bit value into a register	$Rd = N$
MVN	move the NOT of the 32-bit value into a register	$Rd = \sim N$

```
PRE     r5 = 5
        r7 = 8
        MOV     r7, r5      ; let r7 = r5
POST    r5 = 5
        r7 = 5
```

Introduction to the Thumb Instruction Set

- Thumb encodes a subset of the 32-bit ARM instructions into a 16-bit instruction set space
- Since Thumb has higher performance than ARM on a processor with a 16-bit data bus
- but lower performance than ARM on a 32-bit data bus, use Thumb for memory-constrained systems
- Thumb has higher code density

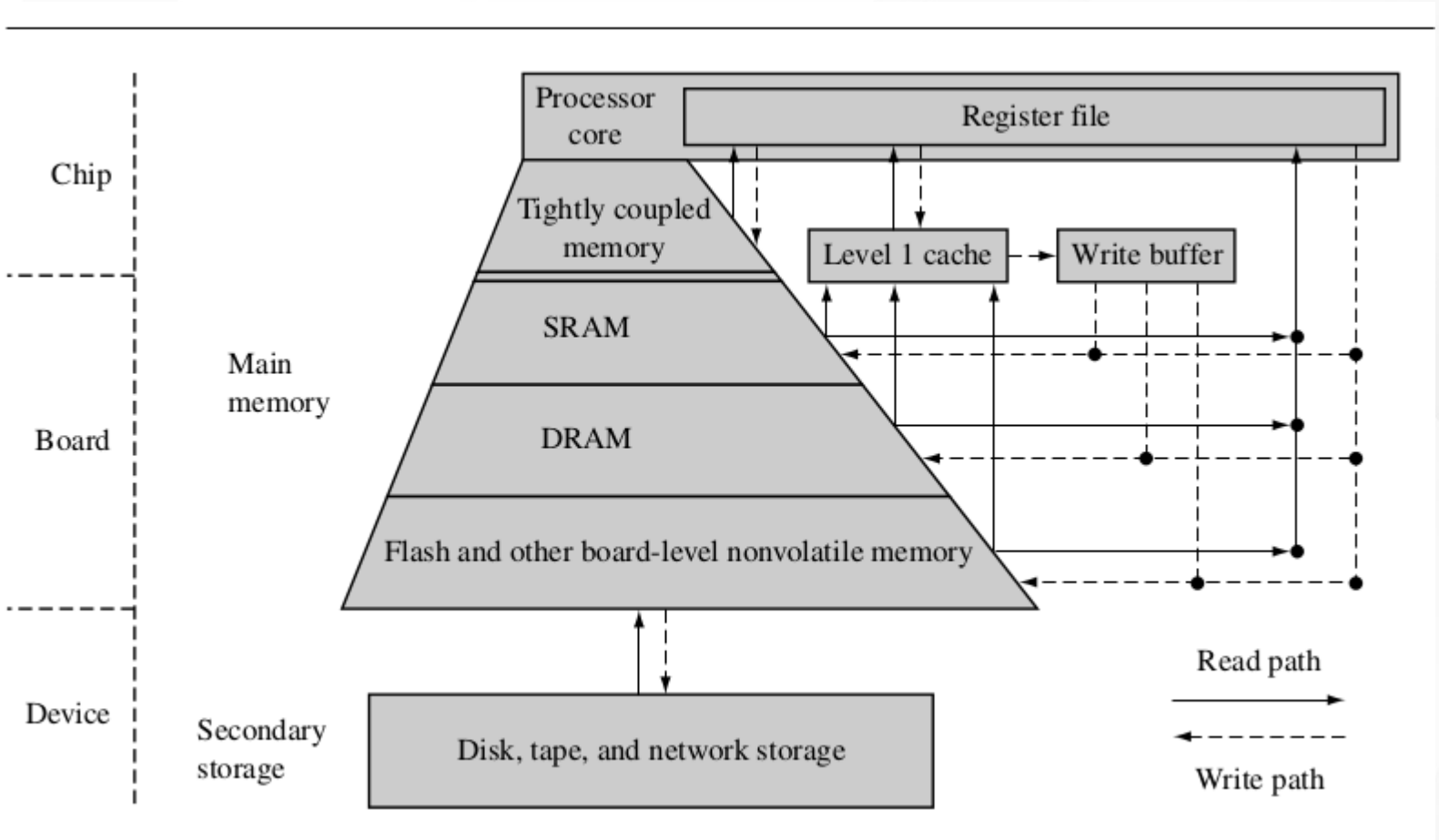
Caches

- Cache
  - a small, fast array of memory placed between the processor core and main memory that stores portions of recently referenced main memory.
- Often used with a cache is a write buffer
  - a very small first-in-first-out (FIFO) memory placed between the processor core and main memory.

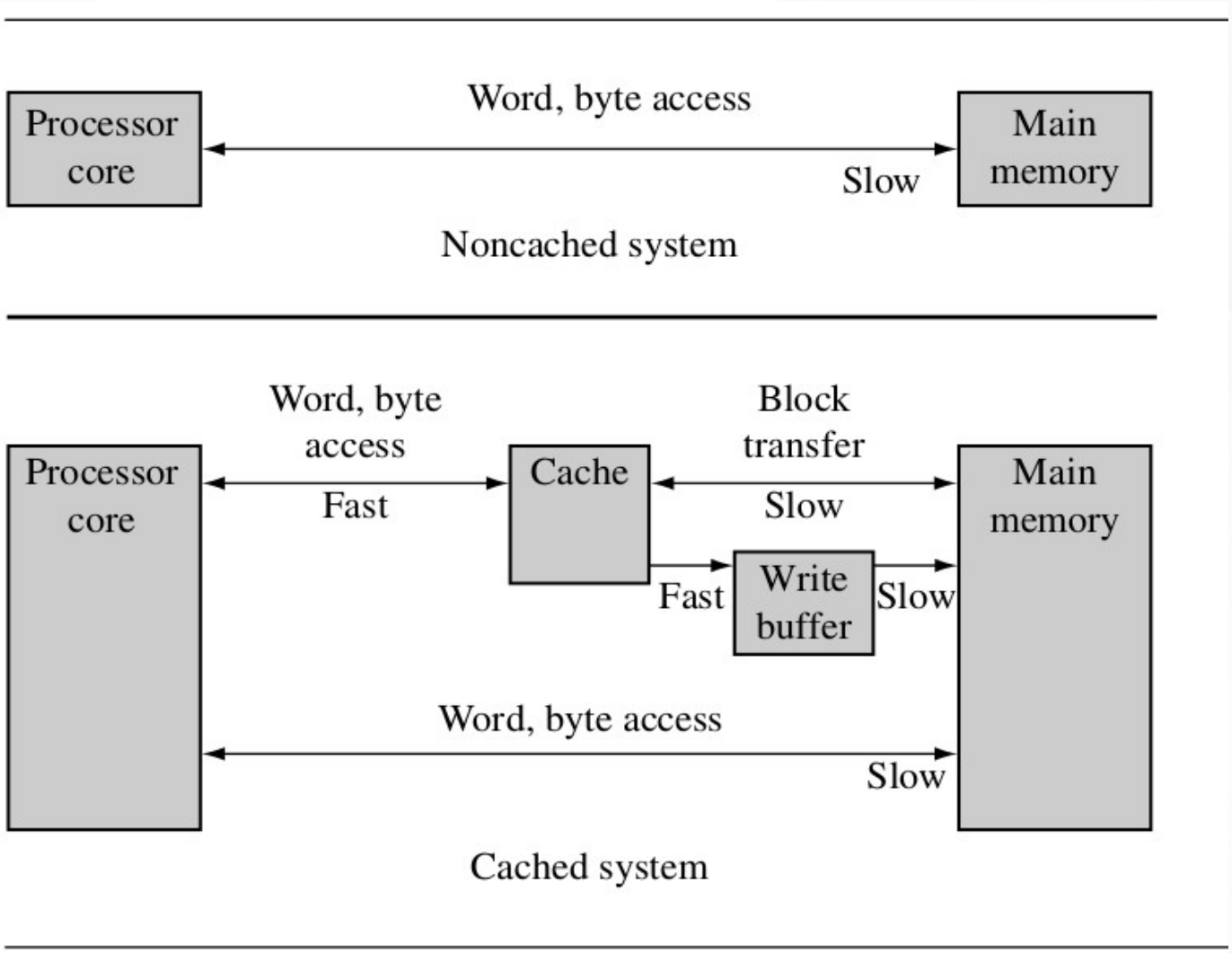
Advantage and Drawbacks

- Advantage
  - Help you create programs that run faster on a specific ARM core
- Drawbacks
  - The difficulty of determining the execution time of a program

The Memory Hierarchy and Cache Memory



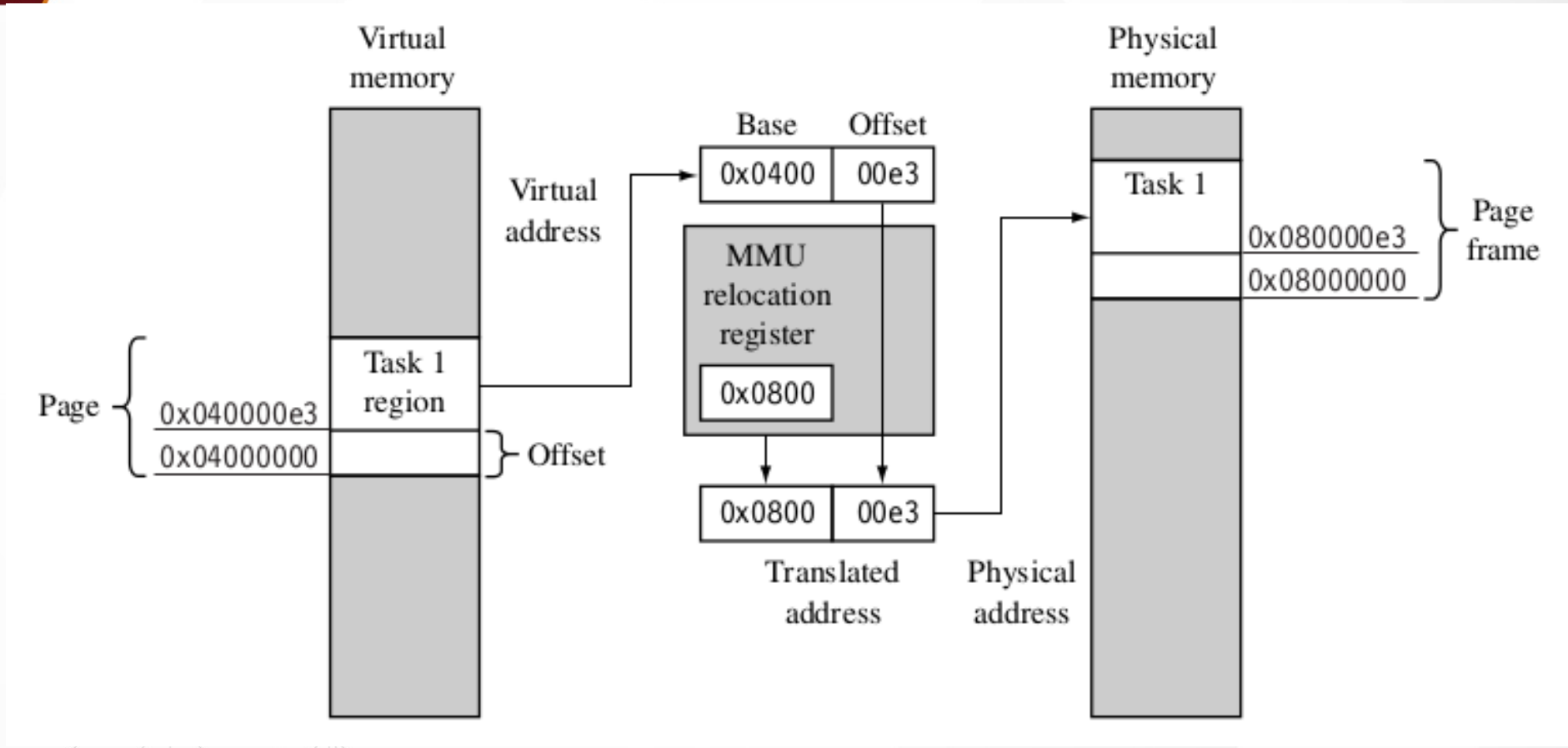
Cache, processor core and main memory



Memory Management Units

- virtual memory
  - an additional memory space that is independent of the physical memory attached to the system
- **Virtual addresses** are assigned by the compiler and linker when locating a program in memory
- **Physical addresses** are used to access the actual hardware components of main memory where the programs are physically located.

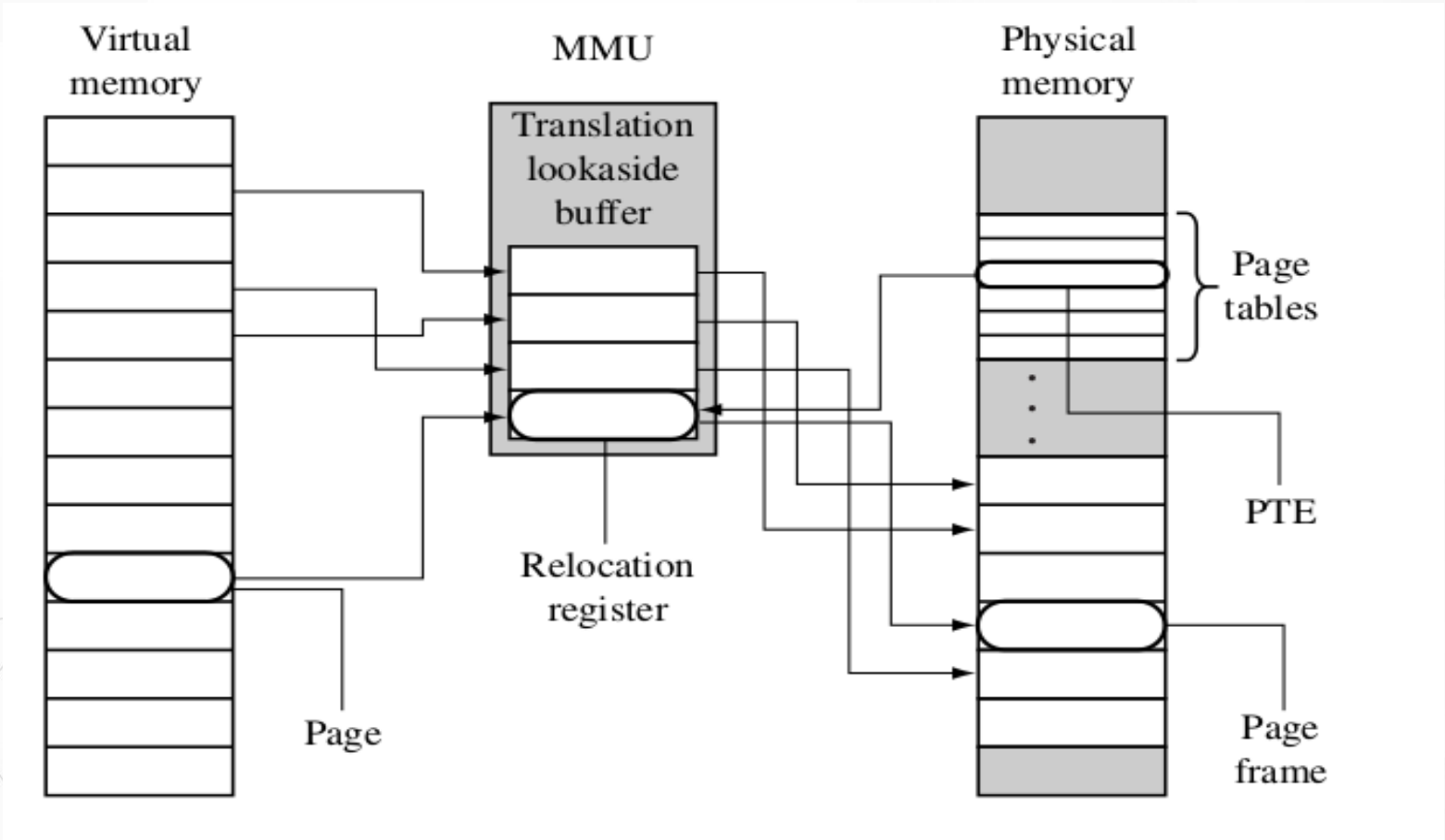
How Virtual Memory Works



Mapping a task in virtual memory to physical memory using a relocation register

TLB

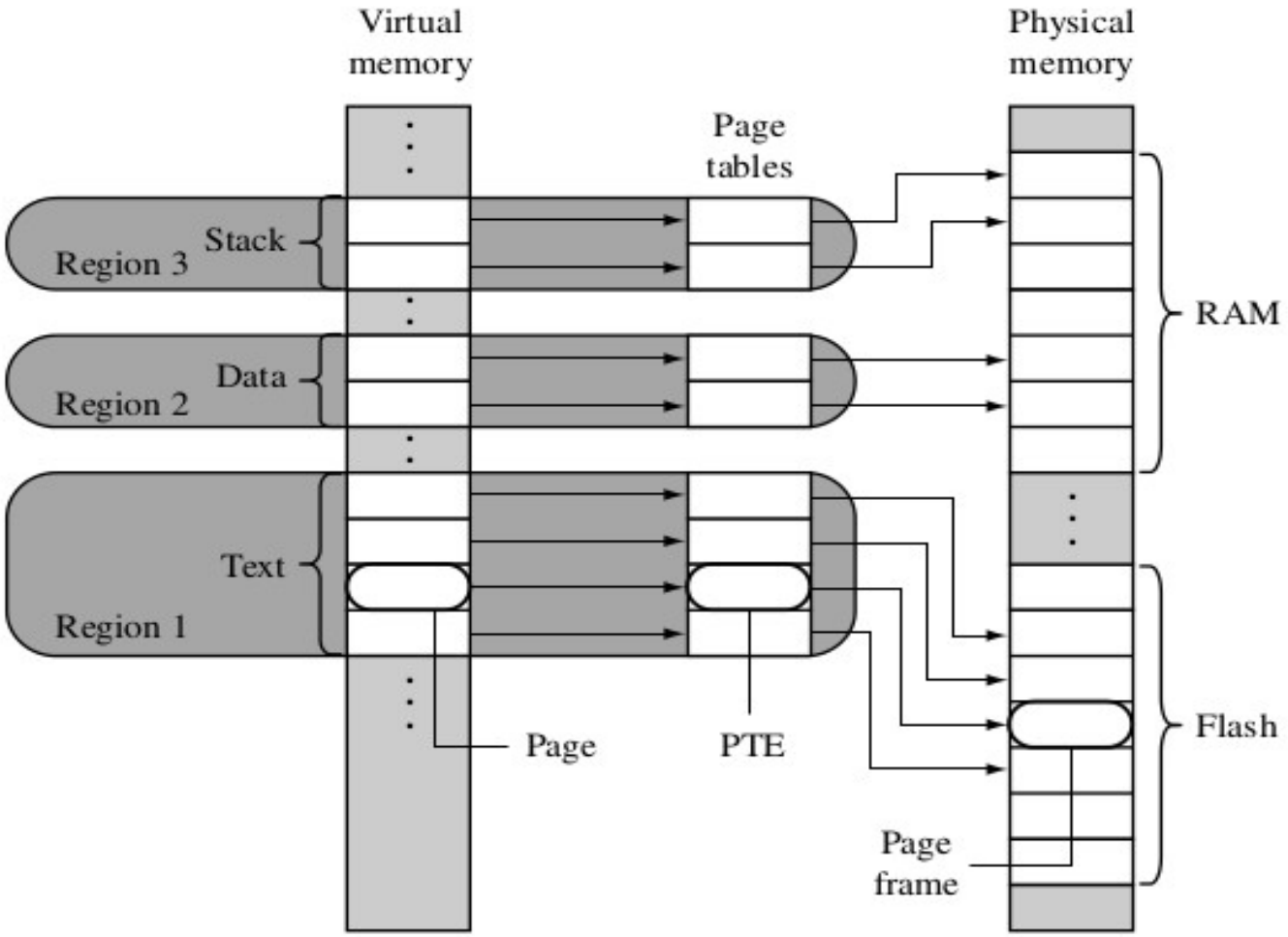
- TLB : Translation Lookaside Buffer



Page Tables

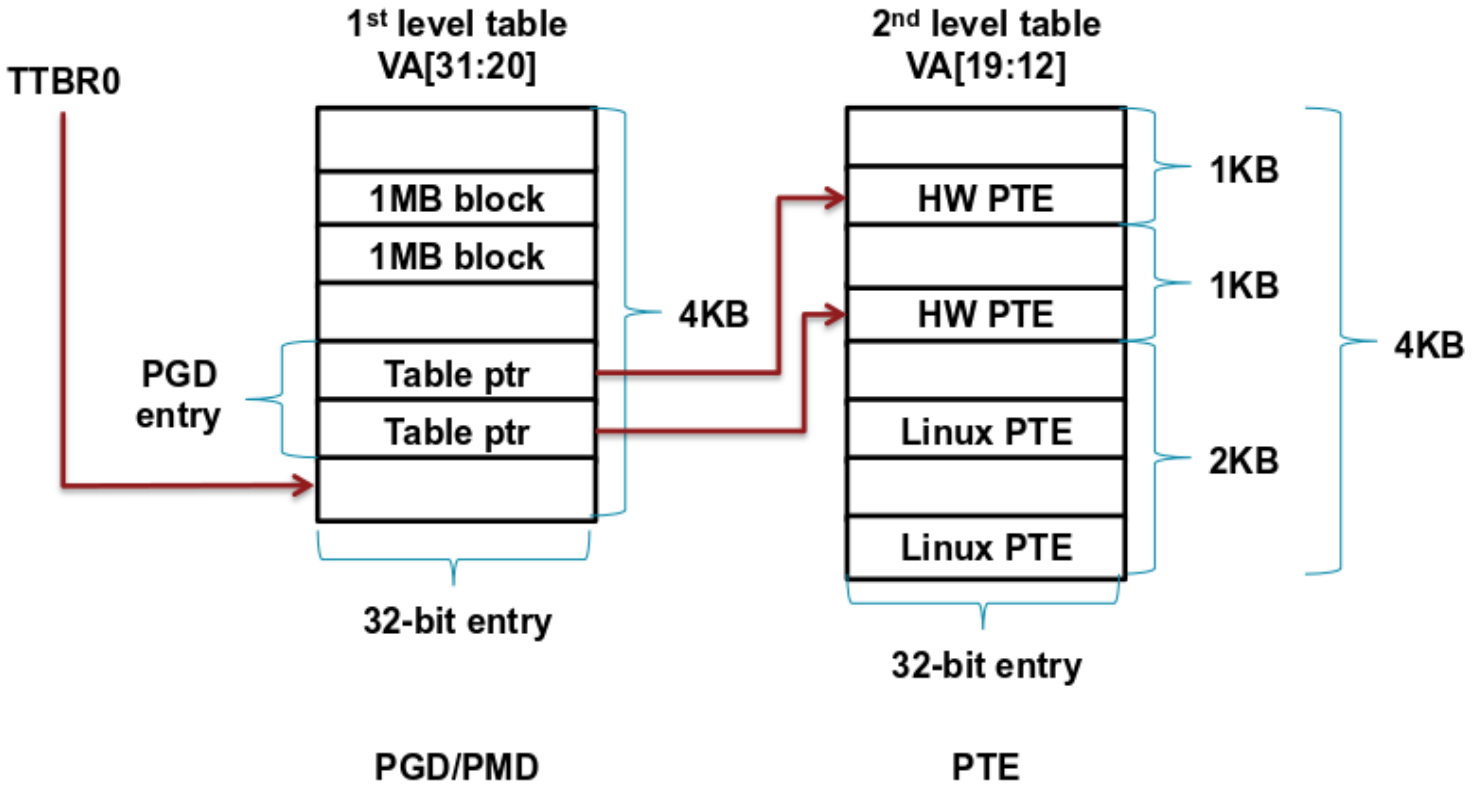
- the MMU uses tables in main memory to store the data describing the virtual memory maps used in the system
- PTE - page table entry

Example : Signal Task and MMU



LPAE

Classic ARM MMU Limitations



LPAE