

Chapter 4 Programmer’s Model of ARM

4.1 Processor operating modes

CPU executes a task on behalf of the operating system or a user. Therefore, it will operate in different modes, generally including:

normal program execution

interrupt processing – triggered by external devices; and

exception processing (triggered by abnormal events)

ARM7TDMI operates in one of the seven modes:

Processor mode			Description
1	User	(usr)	Normal ARM program execution mode
2	FIQ	(fiq)	Support a high-speed data transfer or channel process
3	IRQ	(irq)	Used for general-purpose interrupt handling
4	Supervisor	(svc)	A protected mode for the operating system
5	Abort	(abt)	Entered after a data or instruction prefetch abort
6	Undefined	(und)	Entered when an undefined instruction is executed
7	System	(sys)	Runs privileged operating system tasks

User mode: Most application programs are executed this mode. When the processor is in User mode, the program being executed is unable to access some protected system resources or to change mode, other than by causing an exception to occur. This allows a suitably-written operating system to control the use of system resources.

FIQ mode: Fast interrupt mode. Highest priority of interrupt request.

IRQ mode: Normal interrupt mode. Service interrupt request from different devices.

Supervisor mode: It is entered on reset or when SWI (Software Interrupt) instruction is received. This is the mode in which the operating system usually runs. It allows to access protected system resources.

Protected mode: OS must protect the memory area used by one application program and does not allow any other programs to access this allocated memory.

Abort mode: It is entered when prefetch abort or data abort occurs.

Prefetch abort (instruction fetch memory abort):

A memory abort is signaled by the memory system. Activating an abort in response to an instruction fetch marks the fetched instruction as invalid. A Prefetch Abort exception is generated if the processor tries to execute the invalid instruction.

Data abort: (data access memory abort):

A memory abort is signaled by the memory system. Activating an abort in response to a data access (load or store) marks the data as invalid.

Undefined: It is entered when an attempt is made to execute an undefined instruction.

System mode: It is intended for use by operating system tasks that need access to system resources.

Mode change can be caused by: (i) software control; (ii) external interrupts; and (iii) exception processing.

The seven modes can be broadly categorized into 3 modes:

User	Unprivileged mode	
System	Privilege mode	
FIQ		
IRQ		
Supervisor		
Abort		
Undefined		

Unprivileged mode: User mode is the ONLY unprivileged mode and it has limitation to use system resources.

Privileged mode: It has full access to system resources and can change mode freely.

Exception modes: It interrupts the normal execution of program in responding to specific abnormal conditions.

4.2 Processor states

ARM state: which executes 32-bit word-aligned ARM instructions

Thumb state: which executes 16-bit halfword-aligned Thumb instructions. In this State, the PC uses bit 1 to select between alternate halfwords.

Switching states:

- In assembly language program, `.arm` and `.thumb` directives are used to switch in between states.

Entering Thumb state

- When executing the BX instruction (Branch and Exchange: BX Rm) with bit 0 = 1 in Rm, processor will enter the Thumb state.
- Transition to Thumb state also occurs automatically on return from an exception (IRQ, FIQ, RESET, UNDEF, ABORT, SWI etc) if the exception was entered from the Thumb state.

Entering ARM state

- Whenever the processor enters exception mode (IRQ, FIQ, RESET, UNDEF, ABORT SWI etc.), it will enter the ARM state.
- When executing the BX instruction with bit 0 = 0 in Rm, the processor will enter the ARM state.

4.3 The ARM Registers

The ARM processor has a total of 37 registers, comprising:

- 30 general-purpose registers
- 6 status registers
- a program counter
- Not all of these registers can be seen at once.
- Depending on the processor mode, fifteen general-purpose registers (R0 to R14), one or two status registers and the program counter will be visible.
- Registers are arranged in partially overlapping banks with a different register bank for each processor mode.

Special registers:

PC (Program Counter): Also known as R15 in ARM state. It contains the address of next instruction to be executed.

SP (Stack Pointer): Also known as R13 in ARM state. It is used for storing return address of exception and subroutine; preserving registers, passing parameters to subroutine.

LR (Link Register): Also known as R14 in ARM state. It saves the address of next instruction – the return address for subroutine call occurs.

CPSR (Current Program Status Register):

This contains condition code flags (C, N, V, Z) and the current mode bits.

4.3.1 The ARM register Set

ARM State General Registers and Program Counter

System & User	FIQ	Supervisor	Abort	IRQ	Undefined
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
R8	 R8_fiq	R8	R8	R8	R8
R9	 R9_fiq	R9	R9	R9	R9
R10	 R10_fiq	R10	R10	R10	R10
R11	 R11_fiq	R11	R11	R11	R11
R12	 R12_fiq	R12	R12	R12	R12
R13	 R13_fiq	 R13_svc	 R13_abt	 R13_irq	 R13_und
R14	 R14_fiq	 R14_svc	 R14_abt	 R14_irq	 R14_und
R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)

ARM State Program Status Registers

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	 SPSR_fiq	 SPSR_svc	 SPSR_abt	 SPSR_irq	 SPSR_und

 = banked register

4.3.2 The Thumb register Set

Thumb State General Registers and Program Counter

System & User	FIQ	Supervisor	Abort	IRQ	Undefined
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
SP	 SP_fiq	 SP_svc	 SP_abt	 SP_irq	 SP_und
LR	 LR_fiq	 LR_svc	 LR_abt	 LR_irq	 LR_und
PC	PC	PC	PC	PC	PC

Thumb State Program Status Registers

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	 SPSR_fiq	 SPSR_svc	 SPSR_abt	 SPSR_irq	 SPSR_und

 = banked register

4.3.3 Program status registers

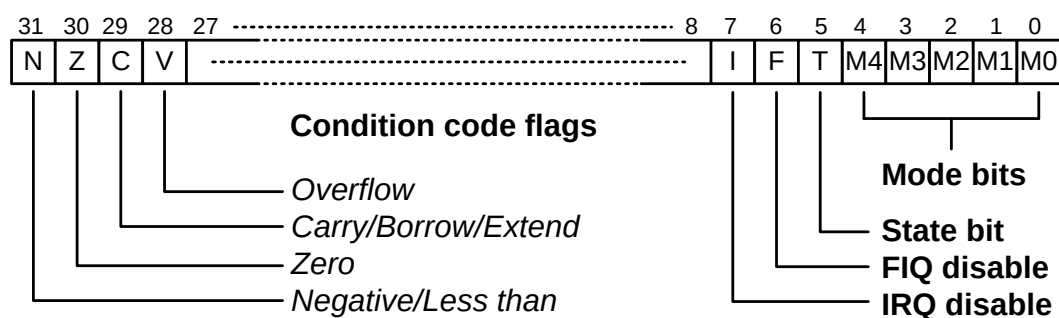
The ARM contains:

- 1 Current Program Status Register (CPSR)
- 5 Saved Program Status Registers (SPSRs) for use by exception handlers

CPSR:

- holds information about the most recently performed ALU operation
- controls the enabling and disabling of interrupts
- sets the processor operating mode
- sets the processor state (Architecture 4T only)

The CPSR is saved to the appropriate SPSR when the processor enters an exception.



Program Status Register format

4.3.3.1 The Condition Code Flags

The N, Z, C and V bits are the condition code flags.

- may be changed as a result of arithmetic and logical operations
- may be tested to determine whether an instruction should be executed
- In ARM state, all instructions may be executed conditionally.
- In THUMB state, only the Branch instruction is capable of conditional execution.

4.3.3.2 The Control Bits

- Bit 0 to Bit 7 of a CPSR are known collectively as the control bits.
- Will be changed when an exception arises.
- If the processor is operating in a privileged mode, they can also be manipulated by software.

Interrupt disable bits When I and F bits = 1, IRQ and FIQ interrupts are disabled respectively.

The state bit T is the processor state bit. In ARM state, T=0. In Thumb state, T=1.

The mode bits The M4, M3, M2, M1 and M0 bits (M[4:0]) are the mode bits. These determine the mode in which the processor operates.

M[4:0]	Mode	Visible THUMB state registers	Visible ARM state registers
10000	User	R7..R0, LR, SP PC, CPSR	R14..R0, PC, CPSR
10001	FIQ	R7..R0, LR_fiq, SP_fiq PC, CPSR, SPSR_fiq	R7..R0, R14_fiq..R8_fiq, PC, CPSR, SPSR_fiq
10010	IRQ	R7..R0, LR_irq, SP_irq PC, CPSR, SPSR_irq	R12..R0, R14_irq, R13_irq, PC, CPSR, SPSR_irq
10011	Supervisor	R7..R0, LR_svc, SP_svc, PC, CPSR, SPSR_svc	R12..R0, R14_svc, R13_svc, PC, CPSR, SPSR_svc
10111	Abort	R7..R0, LR_abt, SP_abt, PC, CPSR, SPSR_abt	R12..R0, R14_abt, R13_abt, PC, CPSR, SPSR_abt
11011	Undefined	R7..R0 LR_und, SP_und, PC, CPSR, SPSR_und	R12..R0, R14_und, R13_und, PC, CPSR
11111	System	R7..R0, LR, SP PC, CPSR	R14..R0, PC, CPSR

PSR Mode Bit Values

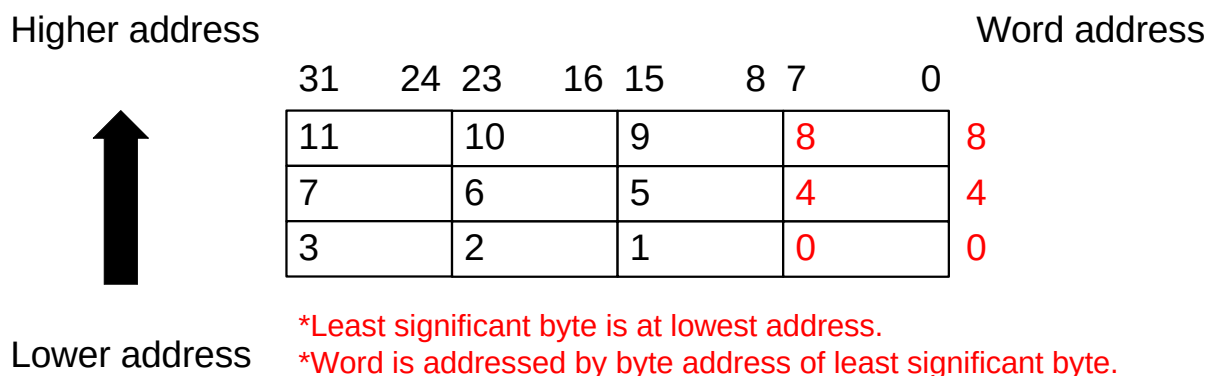
4.4 Memory storage formats

4.4.1 Big and Little endians

ARM7TDMI views memory as a linear collection of bytes numbered upwards from zero. Bytes 0 to 3 (32 bits) hold the first stored word, bytes 4 to 7 the second and so on. ARM7TDMI can treat words in memory as being stored either in Little-Endian or Big-Endian format.

Little-endian format

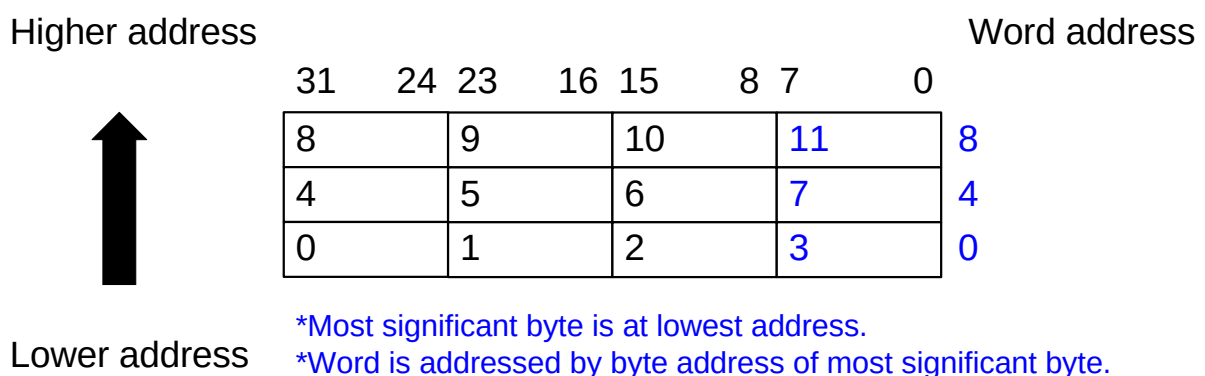
The lowest numbered byte in a word is considered the word's least significant byte, and the highest numbered byte the most significant.



Little-Endian Addresses of Bytes within Words

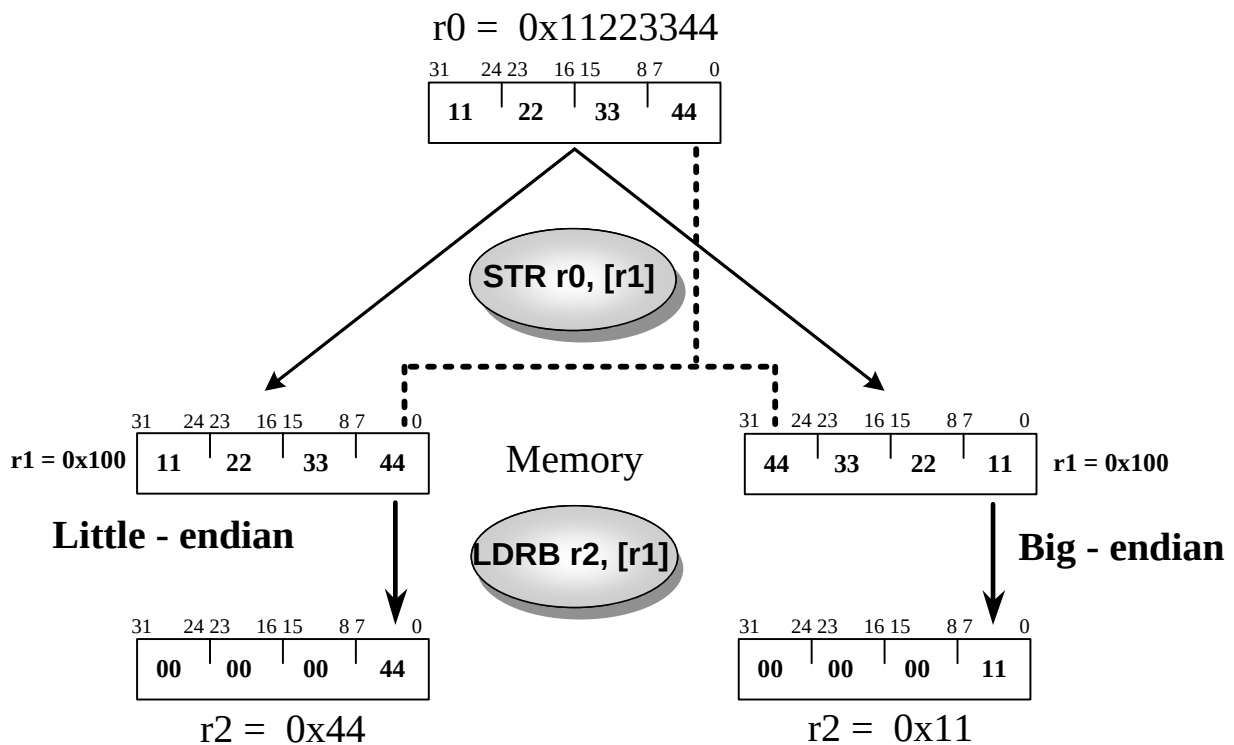
Big-endian format

The most significant byte of a word is stored at the lowest numbered byte and the least significant byte at the highest numbered byte.



never used

Big-Endian Addresses of Bytes within Words



4.4.2 Memory address alignment

The ARM supports the following data types:

Data size	Memory address alignment
Byte	can be aligned to any byte address
Halfword : 2 bytes	halfwords must be aligned to 2-byte address boundaries
Word : 4 bytes	words must be aligned to four-byte address boundaries

Example in Chapter 5 (.align directive)

```
.byte    0x55          /* inserts the byte 0x55 */
.align   /* inserts 3 bytes of 0x00 */
.word    0xAA55EE11    /* inserts 0x11 0xEE 0x55 0xAA */
.hword   0x7F83
.align   /* inserts 2 bytes of 0x00 */
.word    0x3498A5E
```

In Little-Endian format:

Address	+0	+1	+2	+3
0x100000	55	00	00	00
0x100004	11	EE	55	AA
0x100008	83	7F	00	00
0x10000C	5E	8A	49	03

In Big-Endian format:

Address	+0	+1	+2	+3
0x100000	55	00	00	00
0x100004	AA	55	EE	11
0x100008	7F	83	00	00
0x10000C	03	49	8A	5E

4.5 Exceptions

This section is a brief overview of the ARM’s exceptions. Detailed explanation will be covered in later Chapter.

- Generated by internal and external sources to divert the processor to handle an event.
- The CPSR just prior to handling the exception must be preserved to its respective SPSR for resuming to the original program when the exception routine has completed.
- More than one exception may arise at the same time.
- When an exception occurs, an exception service routine corresponding to the type of exception will be executed.
- The addresses of these exception service routines are called the ***hard vectors***.
- The ARM processor supports seven types of exception, and has a privileged processor mode for each type.

Exception processing modes

Exception type	Exception mode	Vector address
Reset	Supervisor	0x00000000
Undefined	Undefined	0x00000004
Software Interrupt (SWI)	Supervisor	0x00000008
Prefetch Abort (Instruction fetch memory abort)	Abort	0x0000000c
Data Abort (Data Access memory abort)	Abort	0x00000010
IRQ (Interrupt)	IRQ	0x00000018
FIQ (Fast Interrupt)	FIQ	0x0000001C

