ARM Accredited Engineer Certification
Mock Test with Answers

Question 1
Which of the following processors would be best suited to a system requiring hard real-time responses, such as a hard drive controller?

A) ARM1136
B) Cortex-A5
C) Cortex-R4 ✓
D) Cortex-A9

Answer C is correct. Cortex-A Programmer Guide, section 2.1 states “The Real-time profile defines an architecture aimed at systems that need deterministic timing and low interrupt latency…”

Question 2
Program execution has caused an abort. The Call Stack view is as follows:

setup() -> main() -> foo() -> bar()

Which function caused the abort?

A) main()
B) bar() ✓
C) foo()
D) setup()

Answer B is correct. ARM DS-5 Using the Debugger, section 6.2 “Examining the call stack” explains that the call stack is “…an area of memory used to store function return information and local variables. As each function is called, a record is created on the call stack.” Also, “When a function completes execution the associated stack frame is removed from the call stack.” So the call stack contains entries for all functions which are currently active. The last one in the chain must, logically, be the one which was executing when the abort occurred.
Question 3
Assume a control register has a value of 0xFECBCA56.
Which of the following actions is needed to set bit 3 of the control register without changing other bits?

A) A “Read-Modify-Write” sequence of the control register ✔
B) No operation is required since bit 3 is already 1
C) A bitwise AND of the control register value with 0x08
D) A “Write-Modify-Read” sequence of the control register

Answer A is correct. ARM recommends following good coding practice which is to use a read-modify-write sequence to change hardware registers. Answer B is incorrect because Bit 3 is clearly not already 1. Answer C is incorrect because this operation would clear all bits except bit 4. Answer D is incorrect because Write-Modify-Read clearly won’t achieve the desired result!

Question 4
Which of the following instructions are called Program Status Register transfer instructions?

A) LDR, STR
B) LDM, STM
C) MCR, MRC
D) MSR, MRS ✔

Answer D is correct. Cortex-A Programmer Guide, section 6.8.4 describes the functionality of the MSR/MRS instructions as transferring values between CPSR/SPSR and a general purpose register. LDR/STR are used to transfer single words between memory and general purpose registers. LDM/STM are used to transfer multiple words between memory and general purpose registers. MCR/MRC are used to transfer values between general purpose registers and coprocessor registers.
Question 5
What are the values of the I and F bits in the Program Status Register on reset?

A) I=0, F=0
B) I=1, F=1 ✓
C) I=0, F=1
D) I=1, F=0

Answer B is correct. Cortex-A Programmer Guide, section 12.2 shows the state of the I and F bits on taking a Reset exception. The values shown mean that both IRQ and FIQ are disabled on reset.

Question 6
In a Cortex-A processor, which exception vector is located at the highest memory address of the exception vector table?

A) Undefined Instruction
B) Data Abort
C) IRQ
D) FIQ ✓

Answer D is correct. Cortex-A Programmer Guide, section 12.1 (Table 12-1) lists the offset of each vector from the base address of the vector table. FIQ is listed as having the highest offset (0x1C).

Question 7
After enabling or disabling a coprocessor (for example VFP and/or NEON), what instruction must be executed before executing any instructions for that coprocessor?

A) DMB
B) DSB
C) ISB ✓
D) NOP

Answer C is correct. Cortex-A Programmer Guide, section 11.2, “This flushes the pipeline and prefetch buffer(s) in the processor, so that all instructions following the ISB are fetched from cache or memory, after the instruction has completed. This ensures that the effects of
context altering operations (for example, CP15 or ASID changes or TLB or branch predictor operations), executed before the ISB instruction are visible to any instructions fetched after the ISB." Enabling/disabling a coprocessor is a context-altering CP15 operation, so an ISB is required.

**Question 8**
According to the ARM Architecture Procedure Call Standard (AAPCS), what is the maximum number of arguments passed to a function to be considered most efficient?

A) 4 Arguments ✓
B) 6 Arguments
C) 8 Arguments
D) 16 Arguments

**Answer A is correct.** Cortex-A Programmer Guide, section 17.1, Table 17-1 shows that four registers (R0-R3) are available for passing parameters. In the following paragraphs, we find “For optimal code, therefore, the programmer should always try to limit arguments to four words or fewer.” This strategy avoids passing parameters on the stack, which would cost time.

**Question 9**
An ARM processor using a Generic Interrupt Controller (GIC) is servicing an active interrupt I1 with a priority value 0x10, when a new interrupt I2, with a priority value 0x0F, is received by the GIC. What action does the GIC take? Assume that the number of priority levels implemented is 256, and the priority mask for that processor interface is 0xFF.

A) I2 is forwarded to the processor ✓
B) I2 is rejected and signalled to the sending peripheral
C) I2 is forwarded to another processor that is currently idle
D) I2 is held pending until I1 is completely serviced

**Answer A is correct.** ARM Generic Interrupt Controller Architecture Specification, section 3.3, “In the GIC prioritization scheme, lower numbers have higher priority, that is, the lower the assigned priority value the higher the priority of the interrupt. The highest interrupt priority always has priority field value 0, and the lowest value depends on the number of implemented priority levels…” This implies that interrupt I2 has a higher priority (0x0F) than interrupt I1 (priority 0x10).

The fact that the priority mask is set to 0xFF means that all interrupts are enabled (0xFF is the lowest possible priority).
The statement that 256 levels of priority are implemented implies that all bits of the 8-bit priority field are significant (the least significant bits are progressively dropped as the number of supported priority levels decreases).

**Question 10**
Which of the following diagrams illustrates the cache hierarchy of a Cortex-A8 processor?

A) ![Diagram A]

B) ![Diagram B]

C) ![Diagram C] ✓

D) ![Diagram D]

**Answer C is correct.** Cortex-A Programmer Guide, section 2.5.3, On Cortex-A8, “…the separate instruction and data level 1 caches are 16KB or 32KB in size. They are supplemented by an integrated, unified level 2 cache, which can be up to 1MB in size.” Diagram C is the only one which matches this description.