***Advanced System on Chip Course***

**QUIZ 5**

**Issue 1.0**

# MODULE 5: Arm Cortex-A9 Processor

**Question 1:**

The Cortex-A9 MPCore can contain up to:

1. 2 Cortex-A9 processors.
2. 4 Cortex-A9 processors.
3. 6 Cortex-A9 processors.
4. 8 Cortex-A9 processors.

**Question 2:**

The Cortex-A9 can be configured for synthesis as follows:

1. With Neon (SIMD) or without Neon.
2. Without MPE (no VFP no Neon), with VFP only (FPU), or with VFP and Neon.
3. Without MPE, with VFP only, or with Neon only.
4. With MPE (VFP and Neon) or with Neon only.

**Question 3:**

The Cortex-A9 processor pipeline:

1. is a variable length, scalar pipeline with three backend execution pipelines.
2. is a variable length, superscalar pipeline that executes three instructions per cycle, issues up to three instructions per cycle, and has five backend execution pipelines.
3. is a variable length, superscalar pipeline that executes two instructions per cycle, issues up to three instructions per cycle, and has five backend execution pipelines.
4. is a variable length, superscalar pipeline that executes two instructions per cycle, issues up to three instructions per cycle, and has three backend execution pipelines.

**Question 4:**

Register renaming facilitates out-of-order execution of WAW and WAR instructions, hence:

1. removes interlocks due to register and data dependencies in code.
2. removes interlocks due to register dependencies in code.
3. removes interlocks due to data dependencies in code.
4. None of the above.

**Question 5:**

Conditions that apply for entering the Fast Loop Mode are:

1. The loop fits in one cache line, it takes less than 64 bytes of memory and finishes with a branch instruction.
2. The loop fits in two cache lines, it takes less than 32 bytes of memory, and finishes with a branch instruction.
3. The loop fits in two cache lines, it takes less than 64 bytes of memory, and finishes with a branch instruction.
4. The loop fits in one cache line, it takes less than 32 bytes of memory, and finishes with a branch instruction.

**Question 6:**

Branch prediction in the Cortex-A9 processor is:

1. Static and speculative thanks to a Return Stack.
2. Dynamic thanks to a Global History Buffer and a Return Stack.
3. Dynamic thanks to a Return Stack and speculative thanks to a Global History Buffer and a Branch Target Address Cache.
4. Dynamic thanks to a Global History Buffer and a Branch Target Address Cache and speculative thanks to a Return Stack.

**Question 7:**

The Performance Monitoring Unit (PMU):

1. adds a little hardware overhead to the Cortex-A9 processor but performs non-intrusively when collecting execution information.
2. does not add any hardware overhead (is a piece of software) to the Cortex-A9 processor and performs non-intrusively when collecting execution information.
3. does not add any hardware overhead (is a piece of software) to the Cortex-A9 processor but performs intrusively when collecting execution information adding some code execution delays.
4. adds a little hardware overhead to the Cortex-A9 processor and performs intrusively when collecting execution information adding some code execution delays..

**Question 8**

Memory hierarchy in the Cortex-A9 processors include:

1. level 1 Harvard memory caches connected directly to the processors and a unified, larger level 2 cache.
2. a level 1 unified memory cache connected directly to the processors and a unified, larger level 2 cache.
3. level 1 Harvard memory caches connected directly to the processors and an optional, unified, larger level 2 cache.
4. optional, level 1 Harvard memory caches connected directly to the processors and an optional, unified, larger level 2 cache.

**Question 9:**

The cache line length in the Cortex-A9 is:

1. four words.
2. eight words.
3. sixteen words.
4. thirty-two words.

**Question 10:**

The instruction cache in the Cortex-A9 is:

1. physically indexed and physically tagged with a pseudo-random or round-robin replacement policy.
2. virtually indexed and physically tagged with a pseudo-random or round-robin replacement policy.
3. physically indexed and physically tagged with an exclusively pseudo-random replacement policy.
4. virtually indexed and physically tagged with an exclusively pseudo-random replacement policy.

**Question 11:**

The data cache in the Cortex-A9 is:

1. physically indexed and physically tagged with a pseudo-random or round-robin replacement policy.
2. virtually indexed and physically tagged with a pseudo-random or round-robin replacement policy.
3. physically indexed and physically tagged with an exclusively pseudo-random replacement policy.
4. virtually indexed and physically tagged with an exclusively pseudo-random replacement policy.

**Question 12:**

The main task of the MMU (Memory Management Unit) is:

1. Indexing the addresses of victims in the instruction cache.
2. Provide coherency between multiple processors sharing L2 cache memory.
3. Translating virtual addresses into physical addresses.
4. None of the above.

**Question 13:**

In a Cortex-A9 MPCore, L1 data cache coherency between processors is provided by:

1. The Memory Management Unit.
2. The Translation Lookaside Buffer (TLB).
3. The PLD Unit.
4. The Snoop Control Unit.

**Question 14:**

When processing a memory address translation, the MMU:

1. first checks the D micro TLB, if a miss, checks the I micro TLB, if a miss, performs a hardware translation table walk.
2. first checks the corresponding micro TLB, if a miss, checks the main TLB, if a miss, performs a hardware translation table walk.
3. first checks the main TLB, if a miss, performs a hardware translation table walk
4. There is no relation between the MMU and the TLB in the A9 processor.

**Question 15:**

In a Cortex-A9 MPCore:

1. There is an interrupt controller (IC) at each Cortex-A9 processor.
2. There is an interrupt controller (IC) at each Cortex-A9 processor and a general interrupt controller that controls all individual ICs.
3. There is only one generalized interrupt controller.
4. Interrupt controllers are not included in Cortex-A processors, only in Cortex-R processors.

**Answers**

Q1)2

Q2)2

Q3)3

Q4)2

Q5)3

Q6)4

Q7)4

Q8)3

Q9)2

Q10)2

Q11)3

Q12)3

Q13)4

Q14)2

Q15)3