***Advanced System on Chip Course***

**QUIZ 11**

**Issue 1.0**

# MODULE 11: System debugging

**Question 1:**

Arm CoreSight technology is basically:

1. a set of signal acquisition software libraries to be run on Arm processors for SoC debugging.
2. a set of software libraries and hardware IPs for halting-mode debugging.
3. a set of software libraries, design tools and hardware IPs to ease the implementation of debug and trace systems in SoCs based on Arm processors.
4. an architectural description of debug interfaces for SoCs based on Arm processors.

**Question 2:**

An *Embedded Trace Macrocell* (ETM) is:

1. a real-time trace module providing instruction and data tracing of a processor.
2. a real-time trace module providing instruction tracing of a processor.
3. a real-time trace module providing data tracing of a processor.
4. None of the above since it does not enable real-time tracing.

**Question 3:**

A *Program Trace Macrocell* (PTM) is:

1. a real-time trace module providing instruction and data tracing of a processor.
2. a real-time trace module providing instruction tracing of a processor.
3. a real-time trace module providing data tracing of a processor.
4. None of the above since it does not enable real-time tracing.

**Question 4:**

ADI defines a standard debug interface for debug components in an embedded System on Chip (SoC). Logically, it consists of:

1. A number of registers that are private to the interface, a means to access the DAP (Debug Access Port) registers, and a means to access the Debug registers of the debug components to which the DAP is connected.
2. A number of registers that are private to the interface and a means to access the DAP registers.
3. The Debug Port (DP) and the Access Ports (AP).
4. A number of registers that are private to the interface, the Debug Port (DP), and the Access Ports (AP).

**Question 5:**

Suppose you want to implement a debug &trace system with Debug APB bus that is able to provide instruction and data tracing of a single core SoC, and which outputs this information off-chip over one communication channel. You will need, at least, the following CoreSight components:

1. An ETM, an STM, a replicator, and two CTIs.
2. An ETM, an STM, a funnel, an ETB, and three CTIs.
3. An ETM, an STM, a funnel, a TPIU, and three CTIs.
4. An ETM, an STM, a funnel, a replicator, an ETB, a TPIU, and three CTIs.

**Question 6:**

Steps to build an Arm debugging subsystem with CoreSight SoC-400:

1. Use SoC-400 to build the debug&trace infrastructure, design your SoC with Arm CPU(s), use PILs to integrate processor and debug&trace subsystem.
2. Design your SoC with Arm CPU(s), use SoC-400 to build the debug&trace infrastructure, use PILs to integrate processor(s) into the debug&trace subsystem.
3. Use SoC-400 to build the debug&trace infrastructure, add TMC and/or STM, use PILs to integrate processor(s) into the debug&trace subsystem.
4. None of the above. In fact above options make no sense.

**Question 7:**

CoreSight components are compatible with:

1. All Arm processors.
2. Cortex processors only.
3. Cortex-A and Cortex-R processors only.
4. ARMv7 and ARMv8 architecture-based processors only.

**Question 8:**

Embedded Cross Triggers:

1. Provide debugger access to the cores and buses in an SoC, across multiple power and clock islands.
2. Combine multiple trace sources together.
3. Synchronize debug and trace across multiple cores.
4. Generate cycle-accurate, instruction trace of Arm processors running at full speed.

**Question 9:**

Trace Funnels:

1. Provide debugger access to the cores and buses in an SoC, across multiple power and clock islands.
2. Combine multiple trace sources together.
3. Synchronize debug and trace across multiple cores.
4. Generate cycle-accurate, instruction trace of Arm processors running at full speed.

**Question 10:**

Embedded Trace Buffers:

1. Store trace data on-chip at high rates at 32-bit data width.
2. Transmit trace data off-chip via 2-34 pins at frequencies asynchronous to the core.
3. Are single pin outputs for Instrumentation Trace.
4. Generate cycle-accurate, instruction trace of Arm processors running at full speed.

**Question 11:**

Trace Port Interface Units:

1. Store trace data on-chip at high rates at 32-bit data width.
2. Transmit trace data off-chip via 2-34 pins at frequencies asynchronous to the core.
3. Are single pin outputs for Instrumentation Trace.
4. Generate cycle-accurate, instruction trace of Arm processors running at full speed.

**Question 12:**

Which of the following statement is correct?

1. Trace mode and monitor mode are the two invasive debug-modes for the Armv7 architecture processors.
2. Halting mode and monitor mode are the two invasive debug-modes for the Armv7 architecture processors.
3. Halting mode and monitor mode are the two non-invasive debug-modes for the Armv7 architecture processors.
4. Halting mode and monitor mode are the two invasive debug-modes for the Armv7 architecture processors.

**Question 13:**

The *Debug Management Registers*define the standardized set of registers that is implemented by all CoreSight components. These are:

1. The Peripheral Identification Registers and the Component Identification Registers.
2. The Breakpoint Value Registers and Watchpoint Value Registers.
3. The Watchpoint Control Registers and Watchpoint Value Registers.
4. The Processor ID Registers and the Component Identification Registers.

**Question 14:**

In Cortex-A9 processors:

1. the DBGDSCCR register controls cache behavior while the processor is in Debug state.
2. the DBGDSCCR register controls cache and the TLB behavior while the processor is in Debug state.
3. the DBGDSCCR register controls the TLB behavior while the processor is in Debug state.
4. the DBGDSCCR register features are not implemented.

**Question 15:**

The CMSIS-DAP debugger interface:

1. provides a standardized way to access the CoreSight Debug Access Port (DAP) of any ARM-based microcontroller via USB.
2. provides a standardized way to access the CoreSight Debug Access Port (DAP) of an Arm Cortex-based microcontroller via USB.
3. provides a standardized way to access the CoreSight Debug Access Port (DAP) of any ARM-based microcontroller via synchronous serial communication.
4. provides a standardized way to access the CoreSight Debug Access Port (DAP) an Arm Cortex-based microcontroller via synchronous serial communication.

**Answers**

Q1) 3

Q2) 1

Q3) 2

Q4) 1

Q5) 3

Q6) 2

Q7) 1

Q8) 3

Q9) 2

Q10) 1

Q11) 2

Q12) 4

Q13) 1

Q14) 4

Q15) 2