DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY,
LONERE – RAIGAD -402 103

Mid Semester Examination – October - 2017

Branch: M.Tech (Embedded System) **Sem.:- I**

Subject with Subject Code:- System Design using Embedded Processors (MTESC101)

Marks: 20

Date:- **Time:- 1 Hr.**

Q1 a) What are the Characteristics of embedded systems? Explain with its significance.

Characteristics of embedded systems

• Single-functioned - Executes a single program, repeatedly

• Tightly-constrained - Low cost, low power, small, fast, etc.

• Reactive and real-time - Continually reacts to changes in the system's environment ▫ Must compute certain results in real-time without delay

• Application-specific functionality – specialized for one or one class of applications

• Deadline constrained operation – system may have to perform its function(s) within specific time periods to achieve successful results

• Resource challenged – systems typically are configured with a modest set of resources to meet the performance objectives

• Power efficient – many systems are battery-powered and must conserve power to maximize the usable life of the system.

• Form factor – many systems are light weight and low volume to be used as components in host systems

• Manufacturable – usually small and inexpensive to manufacture based on the size and low complexity of the hardware.

Discuss any 8 each 1 mark.

Q1 b) Discuss register architecture of ARM processor.

Registers R0 through R7 are the same across all CPU modes; they are never banked. Registers R8 through R12 are the same across all CPU modes except FIQ mode. FIQ mode has its own distinct R8 through R12 registers.

R13 and R14 are banked across all privileged CPU modes except system mode. That is, each mode that can be entered because of an exception has its own R13 and R14. These registers generally contain the stack pointer and the return address from function calls, respectively.

R13 is also referred to as SP, the Stack Pointer. R14 is also referred to as LR, the Link Register. R15 is also referred to as PC, the Program Counter.

The Current Program Status Register (CPSR) has the following 32 bits

- M (bits 0–4) is the processor mode bits.
- T (bit 5) is the Thumb state bit.
- F (bit 6) is the FIQ disable bit.
- I (bit 7) is the IRQ disable bit.
- A (bit 8) is the imprecise data abort disable bit.
- E (bit 9) is the data endianness bit.
- IT (bits 10–15 and 25–26) is the if-then state bits.
- GE (bits 16–19) is the greater-than-or-equal-to bits.
- DNM (bits 20–23) is the do not modify bits.
- J (bit 24) is the Java state bit.
- Q (bit 27) is the sticky overflow bit.
- V (bit 28) is the overflow bit.
- C (bit 29) is the carry/borrow/extend bit.
- Z (bit 30) is the zero bit.
- N (bit 31) is the negative/less than bit.

Register architecture is shown as follows:

Registers across CPU modes

usr	sys	svc	abt	und	irq	fiq
R0						
R1						
R2						
R3						
R4						
R5						
R6						
R7						
R8						R8_fiq
R9						R9_fiq
R10						R10_fiq
R11						R11_fiq
R12						R12_fiq
R13		R13_svc	R13_abt	R13_und	R13_irq	R13_fiq
R14		R14_svc	R14_abt	R14_und	R14_irq	R14_fiq
R15						
CPSR						
		SPSR_svc	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

Architecture of resister of ARM 3 marks Discussion 5 marks.

Q2 a) Differentiate the various categories of embedded system.

Any 4 categories each of 1 mark.

Q2 b) Discuss the various versions of ARM architecture.

Any 4 versions each of 1 mark.

Q2 c) Write a note on thumb-2 instruction set.

Thumb-2 technology was introduced in the *ARM1156 core*, announced in 2003. Thumb-2 extends the limited 16-bit instruction set of Thumb with additional 32-bit instructions to give the instruction set more breadth, thus producing a variable-length instruction set. A stated aim for Thumb-2 was to achieve code density similar to Thumb with performance similar to the ARM instruction set on 32-bit memory.

Thumb-2 extends the Thumb instruction set with bit-field manipulation, table branches and conditional execution. At the same time, the ARM instruction set was extended to maintain equivalent functionality in both instruction sets. A new "Unified Assembly Language" (UAL) supports generation of either Thumb or ARM instructions from the same source code; versions of Thumb seen on ARMv7 processors are essentially as capable as ARM code (including the ability to write interrupt handlers). This requires a bit of care, and use of a new "IT" (if-then) instruction, which permits up to four successive instructions to execute based on a tested condition, or on its inverse. When compiling into ARM code, this is ignored, but when compiling into Thumb it generates an actual instruction.

Q2 d) Describe the debugging features of ARM architecture.

All modern ARM processors include hardware debugging facilities, allowing software debuggers to perform operations such as halting, stepping, and breakpointing of code starting from reset. These facilities are built using JTAG support, though some newer cores optionally support ARM's own two-wire "SWD" protocol. In ARM7TDMI cores, the "D" represented JTAG debug support, and the "I" represented presence of an "EmbeddedICE" debug module. For ARM7 and ARM9 core generations, EmbeddedICE over JTAG was a de facto debug standard, though not architecturally guaranteed.

The ARMv7 architecture defines basic debug facilities at an architectural level. These include breakpoints, watchpoints and instruction execution in a "Debug Mode"; similar facilities were also available with EmbeddedICE. Both "halt mode" and "monitor" mode debugging are supported. The actual transport mechanism used to access the debug facilities is not architecturally specified, but implementations generally include JTAG support.

There is a separate ARM "CoreSight" debug architecture, which is not architecturally required by ARMv7 processors.