

<b>EM2002</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
	<b>MICROPROCESSORS AND MICRO CONTROLLERS</b>	<b>3</b>	<b>0</b>	<b>2</b>	<b>4</b>
	<b>Total Contact Hours – 75</b>				
	<b>Prerequisite: Nil</b>				
<b>PURPOSE</b>					
To enable the student to understand the RISC (ARM) and CISC (Pentium) processors, which will be useful for designing high end embedded systems.					
<b>INSTRUCTIONAL OBJECTIVES</b>					
1.	To learn RISC and CISC architectures of processors.				
2.	To learn ARM processor and its programming with Embedded C.				
3.	To learn to use ARM development tools and carry out experiments				

### **UNIT I - MICROPROCESSOR ARCHITECTURE**

**(9 hours)**

Instruction set - Data formats - Instruction formats - Addressing modes - Memory Hierarchy - register file - Cache - Virtual memory and paging - Segmentation - Pipelining - The instruction pipeline - pipeline hazards - Instruction level parallelism - reduced instruction set - Computer principles - RISC versus CISC - RISC properties - RISC evaluation - On-chip register files versus cache evaluation.

### **UNIT II - HIGH PERFORMANCE CISC ARCHITECTURE – PENTIUM**

**(9 hours)**

The software model - functional description - CPU pin descriptions - RISC concepts - bus operations - Super scalar architecture - pipelining - Branch prediction - The instruction and caches - Floating point unit - protected mode operation - Segmentation - paging - Protection - multitasking - Exception and interrupts - Input/Output - Virtual 8086 model - Interrupt processing - Instruction types - Addressing modes - Processor flags - Instruction set - Basic programming the Pentium Processor. Lab exercise.

### **UNIT III - HIGH PERFORMANCE RISC ARCHITECTURE**

**(24 hours)**

ARM: The ARM architecture - ARM organization and implementation - The ARM instruction set - The thumb instruction set - Basic ARM Assembly language program - ARM CPU cores.

### **ARM DEVELOPMENT ENVIRONMENT**

The AMULET asynchronous ARM Processors. Embedded Operating Systems - Principle Components – Application case study – **VLSI Ruby II** Advanced communication processor – **nuvoTon Cortex M0 (Nu-LB-NUC140)** Microcontroller processor and its supporting tools. Lab exercise

### **UNIT IV - INTRODUCTION TO EMBEDDED C AND APPLICATIONS**

**(16 hours)**

C-looping structures – Register allocation – Function calls – Pointer aliasing – structure arrangement – bit fields – unaligned data and endianness – inline functions and inline assembly – portability issues. Embedded Systems programming in C – Binding and Running Embedded C program in Keil IDE – Dissecting the program -Building the hardware. Basic techniques for reading and writing from I/O port pins – switch bounce - LED Interfacing using Embedded C. Lab exercise

**UNIT V: EMBEDDED OPERATING SYSTEMS(sEOS):****(17 hours)**

Basics of sEOS – Timer Design consideration using sEOS- Multistate system design. Implementation of Traffic light sequencing using onchip UART for RS-232 communication-memory requirements. Case study – Intruder alarm system. HyperTerminal based control-packet based control for LED interfacing- Security challenges and authentication process for Embedded Systems.

**REFERENCES**

1. Daniel Tabak, “*Advanced Microprocessors-SIE*”, Tata McGraw Hill. Inc., 2011.
2. James L. Antonakos, “*The Pentium Microprocessor*”, Pearson Education, 2002.
3. Steve Furber, “*ARM system - on - chip architecture*”, Addison Wesley, 2000.
4. Andrew N. Sloss, Dominic Symes, Chris Wright and John Rayfield, “*ARM System Developer's Guide, Designing and Optimizing System Software*”, Elsevier, 2004.
5. David Seal, “*ARM Architecture Reference Manual*”, Pearson Education, 2007.
6. Michael J. Pont, “*Embedded C*”, Addison Wesley, 2002.
7. Jivan S. Parab, Vinod G. Shelake, Rajanish K.Kamot, and Gourish M.Naik, “*Exploring C for Microcontrollers- A Hands on Approach*”, Springer, 2007.
8. ARM Educational Weblink {<https://www.arm.com/resources/education/education-kits>}