U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

February 2025 Semester End Main Examinations

Programme: B.E. **Semester: IV**
Branch: EEE/ECE/ETE/EIE/MDE **Duration: 3 hrs.**
Course Code: 23ES4PCAPP **Max Marks: 100**
Course: ARM Processor and Programming

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

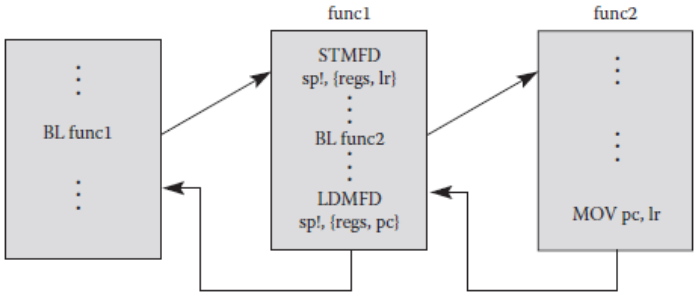
Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

			MODULE - I	*CO*	*PO*	Marks
1	a)		Identify the CISC and RISC features of ARM. Analyze the ARM design philosophy	*CO1*	*PO1*	**10**
	b)		List the registers of ARM processor and explain how are they referred in Load/Store architecture for loading/storing the memory location content in user mode	*CO1*	*PO1*	**10**
			OR			
2	a)		Explain the control bits of CPSR register. How ALU functioning is reflected in control flag bits? Explain the operation of Zero Flag and Carry flag	*CO1*	*PO1*	**10**
	b)		With an example each, give the applications of ARM processors	*CO1*	*PO1*	**10**
			MODULE - II			
3	a)		Identify the addressing mode and differentiate operation of the following instructions. i. LDR R1, [R0, #0x4] ii. STR R1, [R0], #0x4	*CO2*	*PO2*	**04**
	b)		Analyze the following instructions and write appropriate addressing mode and output. Given: r0 = 0x00000000, r1 = 0x00080000, mem32[0x00008000] = 0x02020202 mem32[0x00008004] = 0x03030303 a) LDR r0, [r1, #4]! b). LDR r0, [r1, #4] c). LDR r0, [r1], #4 d) STR r0,[r1,#12] , r1=0x200, r0=0x55 e) LDMIA r0!, {r1-r3}	*CO2*	*PO2*	**06**

		c)	Build the algorithm or flow chart to add five 16 bit numbers stored from memory location labeled 'Value1' and save the sum in location labelled 'Result'. Build the ARM assembly program to perform the task as perthe flow chart/algorithm	*CO2*	*PO2*	**10**
			OR			
	4	a)	List at least five assemble directives and show how they are used in programming	*CO2*	*PO1*	**05**
		b)	What do you mean by pre and post indexing? How are they incorporated in indirect register addressing with offset? Explain	*CO2*	*PO1*	**07**
		c)	Write an Assembly language program to simulate the following equation y=(a*b)+c. Where y is the output to be calculated. a=0x0023 and b=0x0034 and c=0x0045. Show the multiplication in hexadecimal number system	*CO2*	*PO2*	**08**
			MODULE - III			
	5	a)	Discuss about C Data Types used in Embedded C programming	*CO3*	*PO1*	**06**
		b)	Explain ARM register allocation in Embedded C programming	*CO3*	*PO1*	**06**

		c)	Develop equivalent Embedded 'C' code for the following ARM Assembly Language code segment.
			
			Checksum
			MOV R2, R0; This is for loading address
			MOV R0, #0 ; sum =0
			MOV R1, #0 ; i=0
			Loop1 LDR R3, [R2, R1, LSL, #2]; to access successive memory location.
			ADD R1, R1, #1
			AND R1, R1, #0XFF: to convert char to int
			CMP R1, #0X40
			ADD R0, R3, R0
			BCC Loop1 ; check if N=64
			MOV PC, r14
				CO3	*PO2*	**08**

			OR			
	6	a)	Develop an Embedded C code for the calculation of checksum of a data packet containing 64 words to justify that use of "int" type local variables.	*CO3*	*PO3*	**08**
		b)	What do you mean by unrolling of 'C' loops? Create a 'C' function for computation of sum of 20 numbers stored using a pointer named 'data'. Use N as 5 and create four rollovers.	*CO3*	*PO2*	**07**
		c)	What do you mean by compiler optimization? How is it achieved in Embedded C programming?	*CO3*	*PO1*	**05**

			MODULE - IV			
	7	a)	 Fig.1 Subroutine The Fig.1 shows the stacking the link register during the subroutine execution. Elaborate on the subroutine concept with relevant instructions used and the sequence of operations performed during the execution of subroutines.	*CO4*	*PO2*	**10**
		b)	How are subroutines different from Interrupt service routines? Explain the sequence of operations whenever any of the Exceptions are activated in ARM processor	*CO4*	*PO2*	**10**
			OR			
	8	a)	Illustrate the working of IRQ handler in ARM Processor. The illustration must include an example with main program and an IRQ handler, indicating stack utilization.	*CO4*	*PO3*	**10**
		b)	What is stack? How do you access Stack using LDM and STM instructions? Explain Stack Operation in general	*CO4*	*PO3*	**10**
			MODULE - V			
	9	a)	Develop a program to configure the DAC module of LPC 2148 for the square waveform generation	*CO5*	*PO3*	**10**
		b)	With the help of register configurations build the program to transmit a string of characters serially using ARM based LPC2148 microcontroller	*CO5*	*PO3*	**10**
			OR			
	10	a)	Develop a program to configure the DAC module of LPC 2148 for the triangular waveform generation	*CO5*	*PO3*	**10**
		b)	Write an embedded C program to blink the LEDs connected to P1.24 to P1.31 with appropriate comments	*CO5*	*PO3*	**10**

****