19L503 MICROPROCESSORS AND MICROCONTROLLERS

MICROCONTROLLER ARCHITECTURE AND INSTRUCTION SET : Functional block diagram of 8051-power supply, clock and reset circuits-program and data memory organization-addressing modes and instruction set – assembly language programming. (9)

ON-CHIP PERIPHERALS AND PROGRAMMING : Architecture and programming of Parallel Port - timer/counter – Serial Port: UART. Hardware and software Interrupts - Power saving modes. (9)

OFF-CHIP PERIPHERALS INTERFACING AND PROGRAMMING : Display interfacing: LED, 7-segment, LCD - Keyboard interfacing: Push-to-On switch and matrix keyboard - Data convertors: ADC and DAC. (9)

ARM CORTEX-M ARCHITECTURE : ARM architecture, Programmers model, memory, Interrupts and exceptions - Memory Protection Unit (MPU) –Floating Point Unit (FPU) (9)

ARM CORTEX INSTRUCTION SET : Moving the data within the processor -memory access-arithmetic-logic- conversion- bit field-compare and test-program flow control-exception-sleep Mode, SIMD-Saturation-MAC-Floating Point Instructions. (9)

Total L: **45**

TEXT BOOKS:

1. Muhammad Ali Mazidi, J.G. Mazidi, R.D. McKinlay , "The 8051 Microcontroller and Embedded Systems", Second Edition, Pearson Education Limited, 2014.
2. Joseph Yiu , "The Definitive Guide to ARM Cortex M3 and Cortex M4 Processors", Third, Newnes, 2014.

REFERENCES:

1. Steve Furber , "ARM System-on – Chip Architecture", Second Edition, Addison Wesley, Pearson Education Limited, 2000.
2. Andrew Sloss, Dominic Symes, Chris Wright , "ARM System Developers Guide: Designing and Optimizing System Software", Morgan Kaufmann, 2004.

YOUTUBE VIDEO MATERIALS FOR ALL THE TOPICS & FOR ALL THE UNITS IS LISTED HERE

UNIT – I : MICROCONTROLLER ARCHITECTURE AND INSTRUCTION SET

TOPIC	LINK WITH DESCRIPTION
Functional Block Diagram Of 8051	1. https://youtu.be/XpeuwwUvGf0 How Transistors Become Microprocessor. Design of Microprocessor (ALU, Registers, Timing & Control Unit, etc.,) from transistors and how this microprocessor is executing an instruction (Fetching, Decoding & Executing) is explained clearly with op-code and operand is discussed clearly in this presentation. 2. https://youtu.be/cX7MAcKYKrE Difference between Microprocessors and Microcontrollers is demonstrated clearly in this presentation with King, Kingdom and Rainbow Story! 3. https://youtu.be/YAQfnQK8C1Y Overview of Functional Block Diagram of 8051 microcontroller with the functions of each block is discussed with datasheet analysis
Power Supply, Clock and Reset Circuits	https://youtu.be/TfzjlqZhSiU Pin diagram analysis with (VCC Pin, GND pin, Clock Oscillator Pins, Reset Pin with Power-on-Reset Circuit, Pull-up Resistors, Decoupling Capacitors, etc., is discussed clearly in this presentation video.
Program and Data Memory Organization	https://youtu.be/xQeEmW4NcMM Fundamentals of Memory, Basic Building Blocks of Memory (Flip-Flops, Tri-state Buffers, Decoders, Address Bus, Data Bus and Control Bus, Chip-Select, Memory Mapping Techniques, ROM (Program Memory), RAM (SRAM/Data Memory) is explained clearly with examples in this video presentation here.
Addressing Modes and Instruction Set	https://youtu.be/VwDhzZRT21c **Different Addressing Modes:** Immediate Addressing, Direct Addressing, Indirect Addressing, Relative Addressing (SJMP) All the addressing modes are discussed and demonstrated in KEIL IDE with several examples in this video presentation here.
	1. https://youtu.be/qjSSXVYoPcs **KEIL IDE Downloading Steps** for Programming (Assembly & C) is presented here, which helps us to download KEIL IDE in our system. 2. https://youtu.be/7MZzbZiaxuk **Simulating our first assembly language program (blink.a)** in KEIL IDE. Step-by-Step demonstration for the same is explained clearly in

Assembly Language Programming	this video presentation. 3. https://youtu.be/Jq7LG8TgVSI **Simulating our second assembly language program (add.a)** in KEIL IDE. Significance of **Carry Flag** is demonstrated clearly in the video presentation with Banking Examples. 4. https://youtu.be/WgGXc2vn70M **All Flags in 8051 Microcontroller**. Four Flags with 16 possible combinations of setting and clearing the same is simulated and demonstrated clearly with examples using KEIL IDE in this presentation here. 5. https://youtu.be/ILK0hehT-ik **Single Byte Addition with CARRY Flag** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 6. https://youtu.be/C88P-TXDywc **Multibyte Addition with CARRY Flag** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 7. https://youtu.be/1g0HQFWy6ck **Signed and Unsigned number representation in 8-bit, 16-bit and 32-bit,** etc., is discussed clearly with several examples in this presentation. 8. https://youtu.be/HLZX6xwE7MI **Single Byte Subtraction without BORROW** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 9. https://youtu.be/xN0TBDZQpOk **Single Byte Subtraction with BORROW** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 10. https://youtu.be/K01nrjguy3k **Multi Byte Subtraction with BORROW** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 11. https://youtu.be/qPIPEf1n7bI **Single Byte Unsigned Multiplication** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation.

Assembly Language Programming	12. https://youtu.be/CjueW0Ev1Pg **Single Byte Unsigned Multiplication** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 13. https://youtu.be/MYFYWHkuTpM **Multi Byte Unsigned Multiplication** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 14. https://youtu.be/2lY_yW498Ac **Multi Byte Unsigned Multiplication** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 15. https://youtu.be/r3_Tulyf4Os **Single Byte Division** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 16. https://youtu.be/DBy1BR2e8bE **Single Byte Division** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 17. https://youtu.be/2BWhe2g41NY **LOGICAL OPERATIONS** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 18. https://youtu.be/-VLxxIysJJc **SEARCHING FOR A NUMBER IN AN ARRAY** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 19. https://youtu.be/9CGaCVFXpqQ **SEARCHING FOR A NUMBER IN AN ARRAY** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 20. https://youtu.be/o-laiNY0s5c **SEARCHING FOR ALPHABETS IN AN ARRAY** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 21. https://youtu.be/dFdDN7YAwDk **SORTING THE NUMBERS IN AN ARRAY** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation.

Assembly Language Programming	22. https://youtu.be/QZ3SRLPdprk **SORTING THE ALPHABETS IN AN ARRAY** is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 23. https://youtu.be/KReMharybsQ **CODE CONVERSION FUNDAMENTALS WITH BINARY TO BCD** code conversion is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation. 24. https://youtu.be/Dp-YO8gnzGo **CODE CONVERSION FUNDAMENTALS WITH BINARY TO BCD AND BCD TO ASCII** code conversion is presented, coded, simulated and demonstrated clearly in KEIL IDE in this video presentation.

UNIT - II : ON-CHIP PERIPHEARLS AND PROGRAMMING

TOPIC	LINK WITH DESCRIPTION
Architecture and Programming of Parallel Port	https://youtu.be/YAQfnQK8C1Y Datasheet analysis of Parallel Ports of 8051 Microcontroller is presented here with examples.
Timer/Counter	1. https://youtu.be/bwjSN1rAN7c Introduction Timers / Counters: Difference between Adder & Counter?! Pulse Counter, Clock, Overflow Flag, Timing Calculations, Timer Modes, Producing necessary time delay using timers, TMOD register, TCON Register, Programming Timers using KEIL IDE, Mode setting, Checking Timer Overflow flag (TF0/TF1), Watching Variables in Watch Window in KEIL IDE, etc., 2. https://youtu.be/kHa8GP_8niM Timer is used for producing necessary time delay (20ms time delay generation using 16-bit timer mode), Programming and verifying the same in Logic Analyzer of KEIL IDE. Timer is operated in Counter Mode to count the number of pulses coming from T1 pin (Pin P3.5), programming and verifying the same is demonstrated here!

	3. https://youtu.be/tKtuag-CBc8 8-bit Auto Reload Mode of Operation to produce 10ms, 20ms, etc., and verifying the same in Logic Analyzer. 4. https://youtu.be/2zDOHt0y9F4 Hardware Demo of T0 in 16-bit timer mode to produce delay to blink LEDs with C-Code. 6. 5. https://youtu.be/ahE9zpZurhk Hardware Demo of T0 in 16-bit timer mode to produce **1-second delay** to blink LEDs with C-Code. 6. https://youtu.be/DctQdR6139g Hardware Demo of T0 in 8-bit auto-reload mode to produce **1-second delay** to blink LEDs with C-Code. 7. https://youtu.be/T4IZ5WvH2Go Hardware Demo of T0 in 8-bit auto-reload **COUNTER** with C-Code. 8. https://youtu.be/sR8uuyCegCs Hardware Demo of T0 in 8-bit auto-reload **COUNTER** displayed in 7-segment display after necessary code conversion, explained with C-Code. 9. https://youtu.be/gUi3W5neWRo Hardware Demo of ENTRY/EXIT sensing application using two timers (T0 & T1) as **COUNTERS.** Note: All applications including Speedometer, Odometer, Currency Counter, etc., use this counter mechanism.
Serial Port: UART	1. https://youtu.be/NEWNxHLmQ8M Understanding Communication through a game!, Parallel and Serial Communication Fundamentals, PISO, SIPO shift registers in Serial Transmitter and Receiver discussed here. 2. https://youtu.be/Zc_OpeMh938 Parallel Serial Communication Ports, Advantages and limitations of both parallel and serial communication. Simplex, Half-Duplex, Full-Duplex communication with examples. Synchronization between two serial communicating devices, COM Port and its associated signals, UART data frame, Start-bit, Stop-bit, Need for TI and RI Flags 3. https://youtu.be/2HKzcxOswBE Initializing/Configuring serial port registers (SCON, TMOD, TCON, SBUFF) and writing Assembly Language Programming for

	Transmitting data (A, B, BYE) using on-chip UART. Aligning the characters with Carriage Return (ASCII equ. Is 0x0D) and New line (ASCII equ. Is 0x0A) is discussed, programmed, simulated and demonstrated in the video Presentation. 4. https://youtu.be/0Q0LkevQgXY Tabulating the seven differences between **SYNCHRONOUS** and **ASYNCHRONOUS** serial communication with examples and applications is discussed in the video presentation. 5. https://youtu.be/0dYKL67ejeA Serial communication within the chip (**LOOP BACK**): Coding and Demonstration of on-chip UART data loop-back using 8051 Hardware Development Kit is demonstrated in this video presentation. This is one way to justify UART is Full Duplex. 6. https://youtu.be/59KH86ATReI UART is off-board serial communication protocol. Communication between two boards using UART. COM Port is used interconnect two boards which communicates using RS232 cable. MAX232 is a Voltage-Level-Converter, which is used to communicate through COM port, which is demonstrated here in this video presentation.
Hardware Interrupts	1. https://youtu.be/mrEcpaf-nWg Interrupt Fundamentals, its need, priority, need for priority, Hardware edge and level triggered interrupts, Datasheet analysis of Hardware interrupt structure in 8051 microcontroller, Interrupt Enable (IE) Register, Interrupt Priority (IP) register, Vector Address Location for Interrupts. Programming, Simulating and Demonstrating External Interrupt 0 (EX0) (configured as Level Triggered interrupt (Level-0)) in 8051 using KEIL IDE. Similarly, EX0 configured as falling edge triggered interrupt mode is simulated and demonstrated with examples in KEIL IDE. Datasheet to Application Demo of External (Hardware) interrupts is discussed very clearly in this video presentation here. 2. https://youtu.be/uq0VwEgKchw Hardware Demo of External Interrupt 0 in **Level Triggered Mode** in 8051 Development kit with C-code is explained clearly in this video presentation. 3. https://youtu.be/Gxs9LiCLIpo Hardware Demo of **both External Interrupt 0 (EX0) & (EX1) in Level Triggered Mode** in 8051 Development kit with C-code.

	Pre-emption concepts with two interrupts and one main routine is demonstrated highlighting priorities of interrupts in this video presentation. 4. https://youtu.be/LnLvIY0J0aE Hardware Demo of **External Interrupt 0 (EX0) in Edge Triggered Mode** in 8051 Development kit with C-code is explained clearly in this video presentation. 5. https://youtu.be/CrM3JfAmbf0 Hardware Demo of **both External Interrupt 0 (EX0) & (EX1) in Edge- Triggered-Mode** in 8051 Development kit with C-code.
Software Interrupts	1. https://youtu.be/5IXY58m4aIU Difference between Interrupts and Exceptions are demonstrated with divide-by-zero exception example here first. IE register is initialized to enable T0 interrupt; TMOD register is initialized to enable T0 in 16-bit timer mode; Coding, Executing, Simulating and Demonstrating the same with program execution jumps to necessary Vector Location of T0 (0x000B). 2. https://youtu.be/zVNaC5adKog **Timer-0 periodic interrupt with a period of 8ms is used to refresh the multiplexed 7-segment display.** Demonstrated with concepts, coding, programming, simulating and verifying the same in the hardware development kit. 3. https://youtu.be/elXWSFiw1vU **Two Hardware interrupts in Edge Triggered Mode for entry and exit** and **One Software (Timer Periodic Interrupt) for display refreshing of multiplexed 7-segment display** is demonstrated in the hardware for Display System for Entry Exit application.
Power Saving Modes	

UNIT - III: OFF-CHIP PERIPHEARLS INTERFACING AND PROGRAMMING

TOPIC	LINK WITH DESCRIPTION
Display interfacing: LED	1. https://youtu.be/TfzjlqZhSiU **(Hardware Initialization and LED Interfacing Circuit)**

	Pin Diagram, VCC, GND, Crystal Oscillator, Power-on-Reset (POR) Circuit, Pull-up Resistors, **LEDs interfaced with 8051 Microcontroller with Current Limiting Resistors!** 2. https://youtu.be/6tIhs0b8--s **(Coding and Application Programming Steps from the Scratch)** Persistence of Vision, Need for Delay routine in an application for blinking of LEDs. Writing delay routine for making LEDs to blink! Tool chain for software development, Host and Target Machine, Integrated Development Environment (IDE) (KEIL IDE), etc., ***Hardware Initialization with LED Interfacing circuit diagram; Programming techniques for Blinking of LEDs; Downloading the program from Host Machine and testing the same in the target machine is demonstrated in these two videos!***
7-Segment Display	1. https://youtu.be/qfDGaUidmGo 7-Segment Display Interfacing is discussed in detail from the scratch. 2. https://youtu.be/LfsT7pvxDwQ Displaying FUEL in Multiplexed 7-segment Display. Application of 7-segment Code is well understood with this video presentation. 3. https://youtu.be/19l0qlyCsc4 Displaying PUSH, PULL, FUEL, ALL in Multiplexed 7-segment display. Circuit Diagram of multiplexed 7-segment display is discussed clearly here.
LCD	https://youtu.be/-GF4OMvWh-M 16x2 LCD Display Interfacing is discussed in detail from the scratch.
Keyboard interfacing: Push-to-On switch	https://youtu.be/A-RQRdEty30 How a microcontroller identifies the key press
Matrix Keyboard	1. https://youtu.be/0oGD46xBBPA (4x4 Matrix Keyboard Interface with LEDs) Circuit Diagram for interfacing 4x4 Matrix Keyboard with 8051 Microcontroller. Concept of Identifying the Key Press in Matrix Format C-Program for identifying the Key press in 4x4 Key Matrix. Simulation of the same in KEIL IDE. Testing the Key Press Identification in Hardware Development Kit. Displaying the Key Press Value in 8-LEDs 2. https://youtu.be/4AgNGzWggOU (4x4 Matrix Keyboard interface with Multiplexed 7-segment Display)

Data convertors: ADC	1. https://youtu.be/er3sRatDKG8 (Need for ADC in an Application: Sampling, Quantizing & Encoding Concepts Demonstrated here) 2. https://youtu.be/Gmxr0IqbUVo (Refreshing ADC 0804 Datasheet, Interfacing Circuit, Potentiometer interface with ADC0804, Start-of-Conversion(SOC), End-of-Conversion (EOC), Vref+, Vref-, Analog Vcc & Analog GND, Digital Ground) 3. https://youtu.be/hJSiwnDnY_c (Testing of ADC0804 in Hardware with Multi-meter and 8-LEDs, Verifying ADC Resolution, Testing SoC, etc.,) 4. https://youtu.be/zDQ785P4oig (Interfacing ADC 0804 with AT89S52 microcontroller in 8051 Development Kit. Displaying the Analog to Digital Converted value in multiplexed 7-segment display in Kit)
DAC	https://youtu.be/3dVji3CyTwI (Testing the performance of on-chip 10-bit DAC in LPC2148 Microcontroller) (Just observe the Analog Output (**AOUT**) signal in Logic analyzer simulator in KEIL IDE where DAC produces a saw-tooth waveform for the 10-bit Digital Input)

UNIT - IV: ARM CORTEX-M ARCHITECTURE

TOPIC	LINK WITH DESCRIPTION
ARM Architecture (Overview of ARM Cortex-A ARM Cortex-R ARM Cortex-M Architecture)	https://youtu.be/ExmDZsAR4VE (ARM Cortex-A, ARM Cortex-R & ARM Cortex-M Core Architecture)
ARM Architecture (Overview of ARM Cortex-A ARM Cortex-R ARM Cortex-M Architectural Features & Target Application Areas)	https://youtu.be/fHTmmeRqYDQ (ARM-7, ARM-9 & ARM-11; ARM Cortex-A, ARM Cortex-R & ARM Cortex-M Core Architecture Design, System Features and Target Applications)
ARM Architecture	

(Overview of ARM Architecture Versions for Different Cores)	https://youtu.be/pysNOMwgbUc (ARMv4, ARMv5, ARMv6, ARMv6-M, ARMv7-A, ARMv7-R, ARMv7-M, ARMv8-A, ARMv8-R, ARMv8-M, etc.,)
Programmers model (ARM Registers)	https://youtu.be/oay3wjJUjwE (ARM Registers with ARM Memory Structure is explained with an example through LPC2148 ARM Processor based Microcontroller)
Memory Structure (Simple overview of ARM Memory Structure, Load Store Operations for data exchange between registers and Memory Locations with ARM Architecture Overview here)	https://youtu.be/aXBbTPoSznM (Program Memory, Data Memory (SRAM), Peripheral Memory Location, Loading and Storing Byte variable, Long integer variable, etc., Demonstrating Load/Store operation with Memory Structure through an example program Simulation
Interrupts and exceptions (Comparison with the features of 8051, interrupts and exceptions in ARM Processor Core is Explained here)	https://youtu.be/ZnfGLXzSMdc (Interrupts and Exceptions, NMI, IRQ, Nested Vectored Interrupt Controller (NVIC), Stack Overflow Exception, Memory Fault Exception (Fetching Wrong Instructions) Watch-Dog-Timer (WDT) Reset Interrupt, Brown-Out-Reset (BOR) Interrupt/Exception, Exception Priority (Fixed and Programmable) features discussed in this presentation link
Memory Protection Unit (MPU)	
Floating Point Unit (FPU)	

UNIT - V: ARM CORTEX-M INSTRUCTION SET

TOPIC	LINK WITH DESCRIPTION
Overview of ARM Processor Instructions	https://youtu.be/EPdeolbOF74 Data Processing Instructions, Data Transfer Instructions & Control Flow Instructions discussed with examples here in this presentation link.
Moving the data within the processor & Memory Access Instructions	https://youtu.be/9yJLG7bovqA (Data Transfer Instructions)
Arithmetic Instructions	https://youtu.be/_aRNxc_NCoA (Cortex-M4 Arithmetic Instructions)
Shift and Rotate Instructions	https://youtu.be/x9pbGCKH_h8 (Cortex-M4 Shift and Rotate Instructions)
Logical Instructions	https://youtu.be/M4MpNnxDx4Q (Cortex-M4 Logical Instructions)
Data Conversions Operations	https://youtu.be/MdSetY8WM3w (Extend & Reverse Ordering Instructions)
Bit Field Control Instructions	https://youtu.be/xyRkkIWcC1o (Instructions for Bit Field Clear, Bit Field Insert, Count Leading Zeros, etc.,)
Compare and Test	
Program Flow Control	https://youtu.be/8y_4EHin3ps Branch (B, BL, BLX), Conditional Branches, Compare and Branch if Zero/Non Zero (CBZ, CBNZ), IF-THEN (IT), Table Branch (TBB), etc., discussed with examples in this presentation.
Exception	
Sleep Mode	https://youtu.be/GTPzJpzwHOI (ARM Cortex-M4 Sleep Mode Instructions)
SIMD	
Saturation	
MAC	
Floating Point Instructions	https://youtu.be/PH6h7Vdu9gg (IEEE 754 format representation of Floating Point Numbers and verification)