ITE1001	Digital Logic and Microprocessor	L	T	P	J	C
		3	0	2	0	4
Pre-requisite	**NIL**	**Syllabus version**				
		1.00				

Course Objectives:

- To learn logic circuits and converters
- To understand the components of a digital system
- To understand the microprocessor architecture and assembler instruction formats

Expected Course Outcome:

1) Study, design and experiment the various digital logic design and architectures of microprocessors.
2) An ability to design and use the various combinational logic circuits.
3) Design and evaluate the various flip flops and counters for sequential logic circuits.
4) Analyze, design and implement the architecture of 8085.
5) Comprehend the design details of architecture of 8086 microprocessor.
6) Design and implement the various programming models of 8066 architecture.
7) Analyze and design the application of peripheral chips in various microcontroller architectures.

Student Learning Outcomes (SLO):	**6, 14**

[6] Having an ability to design a component or a product applying all the relevant standards and with realistic constraints

[14] An ability to design and conduct experiments, as well as to analyze and interpret data

Module:1	**Introduction**	**4 hours**

Review of number systems - Logic gates: NAND, NOR gate as universal building blocks - Simplification of four-variable Boolean equations using Karnaugh maps

Module:2	**Combinational Logic circuits**	**5 hours**

Half adder, Full adder, Half subtractor, Full subtractor - 4-bit parallel adder and subtractor - 3-bit binary decoder – Decimal to BCD encoder – 8-to-1 multiplexer, 1-to-8 Demultiplexer

Module:3	**Sequential Logic Circuits**	**8 hours**

Flip-flops: SR flip-flop, Edge-triggered flip-flops (SR,D,JK and T), Master-slave JK flip-flop - 4-bit binary asynchronous and synchronous counter - Decade counter (asynchronous and synchronous) - Shift registers (SISO,SIPO,PISO,PIPO) - Ring counter – Memories (RAM, ROM, EPROM,FLASH)

Module:4	**The 8085 Microprocessor Architecture**	**4 hours**
Pin diagram - CPU architecture – Flags-Interrupts – Instruction Set-Addressing mode		
Module:5	**The 8086 Microprocessor**	**8 hours**
Pin diagram, CPU architecture, addressing mode, Segmentation- Minimum mode maximum mode operations -Memory Interfacing-I/O interfacing		
Module:6	**Programming model of 8086**	**7 hours**
Programming model of 8086, Addressing modes, Instruction Formats, Instruction set, Assembler directives and Assembly language Programming of 8086.		
Module:7	**Peripheral Chips**	**7 hours**
Block diagram – pin diagram**,** 8255 (PPI), 8254 (Timer), 8257 (DMA), 8259 (PIC), 8251 (USART)8279(Keyboard and Display Interfacing)		
Module:8	**Contemporary issues:**	**2 hours**
	Total Lecture hours:	**45 hours**

Text Book(s)

1.	Ramesh Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, Sixth Edition, Penram International Publishing, 2013.
2.	Morris Mano, Digital logic and Computer design, 4th Edition, Pearson, 2008.

Reference Books

1.	Yu-Cheng Liu, Glenn A. Gibson, Microcomputer Systems: The 8086/8088 Family-Architecture Programming and Design, Second Edition, Pearson, 2015.
2.	R.K. Gaur, Digital Electronics and Microcomputers, Dhanpat Rai Publications, 2012.

List of Challenging Experiments (Indicative)

Digital Logic Design

1. Basic Logic Gates
2. Combinational Circuits
3. Adders and Subtractors
4. Code Convertors
5. Parallel Adder and Magnitude Comparator
6. Decoder and Encoder
7. Multiplexer and De-multiplexer
8. Sequential Circuits and Shift registers
9. Counters

Microprocessors

	10. To write programs in Assembly Language using 8085 instruction set. 11. To write programs in Assembly Language using 8086 instruction set. 12. To perform interfacing of RAM chip 13. To perform interfacing of keyboard controller 14. To perform interfacing of DMA Controller 15. To perform interfacing of UART/USART
1.	Assume a large room has 3 doors and a switch near each door controls a light in the room. The light is turned on or off by changing the state of any one of the switches. More specifically the following should happen: 1. The light is OFF when all 3 switches are open. 2. Closing any one switch will turn the light ON. 3. Then closing the second switch will have to TURN OFF the light. 4. If the light is OFF when the 2 switches are closed, then by closing the third switch the light will TURN ON.

2.	Design hardware that implements the following pseudo-code using the provided Comparator, Adder and Registers, along with as many multiplexers and de-multiplexers as needed. The comparator has two inputs In1 and In2, and three outputs, C1, C2, and C3. If In1 < In2, C1 = 1; if In1 = In2, C2=1; if In1 > In2, C3 =1 (for a given In1 and In2, only one of the comparator outputs can be 1). The Adder takes as inputs two numbers p and q, and produces an output Sum. There are 5 registers for storing the 5 variables, A, B, X, Y, and Z. • Hint: You do not need to use truth table or K-maps. Insert the muxes/demuxes as appropriate, and show the signal connections from the input registers A, B, X to the output registers Y and Z, through the muxes, comparator, adder, and demuxes. Be sure to show the equations for the select lines of the multiplexers/demultiplexers in terms of the comparator outputs, C1, C2, and C3.
	Pseudo-code:
	If A<B then
	Z= X+ A
	Else if A=B then
	Z= X+ B
	Else
	Y = A + B

3.	Design a simplified traffic-light controller that switches traffic lights on a crossing where a north-south (NS) street intersects an east-west (EW) street. The input to the controller is the WALK button pushed by pedestrians who want to cross the street. The outputs are two signals NS and EW that control the traffic lights in the Ns and EW directions. When NS or EW are 0, the red light is on, and when they are 1, the green light is on. When there are no pedestrians, NS=0, EW=1 for a minute, follow by NS=1 and EW=0 for 1 minutes, and so on, when WALK button is pushed, Ns and EW both become 0 for a minute when the present minute expires. After that the NS and EW signals continue alerting. For this traffic-light controller: a) Develop a state diagram. (Hint: can be done using 3 states) b) Draw the state transition table. c) Encode the states using minimum number of bits. d) Derive the logic

	schematic for a sequential circuit which implements the state transition table.
4.	Many game shows use a circuit to determine which of the contestants ring in first. Design a circuit to determine which of two contestants rings in first. It has two inputs S1 and S0 which are connected to the contestants' buttons. The circuit has two outputs Z1 and Z0 which are connected to LED's to indicate which contestant rang in first. There is also a reset button that is used by the game show host to asynchronously reset the flip-flops to the initial state before each question. If contestant 0 rings in first, the circuit turns on LED 0. Once LED 0 is on, the circuit leaves it on regardless of the inputs until the circuit is asynchronously reset by the game show host. If contestant 1 rings in first, the circuit turns on LED 1 and leaves it on until the circuit is reset. If there is a tie, both LED's are turned on. The circuit requires four states: reset, contestant 0 wins, contestant 1 wins, and tie. One way to map the states is to use state 00 for reset, state 01 for contestant 0 wins, state 10 for contestant 1 wins, and state 11 for a tie. With this mapping, the outputs are equal to the current state, which simplifies the output equations.
5.	Design a simple circuit that could operate a car alarm. The circuit has one input Y which would be connected to the car's door switch to determine if the car door is open or shut. When the door is shut Y = 0, and when the door is open Y = 1. The circuit has one output Z which is used to operate a horn by shorting the wires that go to the horn switch in the steering wheel. When Z = 1, the switch is activated and the horn honks. The circuit would be asynchronously reset by the accessories power line that is high when the ignition is turned on or is in accessory-only mode, both of which require the key to the car.
6.	Design a 12 hour Digital clock which is usually set up to start at 12:00, and they count 12:01, 12:02, 12:03, 12:04, 12:05, 12:06, 12:07, 12:08, 12:09, 12:10, and eventually the clock gets to 12:58, 12:59, 1:00, and so on. The one's place of the minutes (the right-most digit) counts 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and then repeats. The ten's place of the minutes (second digit from the right) counts 0, 1, 2, 3, 4, 5, and then repeats. The hour counter counts 12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and repeats.
7.	Design a Microprocessor based combinational lock which has a combination of five digits. The five digits are entered from a keyboard and they are to be entered within a 10 seconds. If the right combination is entered the lock will open. If after 10 seconds either all five digits are not entered or a wrong combination is entered then the display will show an error message. Then the system will allow 5 seconds for the first digit to be entered the second time. If after this time the digit is not entered, the system will turn ON the alarm. If the second try fails, the alarm is also turned ON. Then to reset the system the power has to be turned OFF.(Scrambling Keypad)
8.	Design a microprocessor based Smart Pill Box Alarm System for Elderly people. The system will alert the user 3 times per day for taking up the pills. The user has to set the system into fixed slots: for example: Morning, Afternoon, Evening and Night. The system will deliver a display message such as “Take this Pill X “five minutes before the scheduled time. A real

	time clock is to be included in the system to display the current time and will show the alarm as per the time slots.
9.	Design an intelligent system for the following real time situation. Consider you are driving a car. You are having a limited display area, where you need to display the fuel status, temperature status, Speed limit, Gear Position based on the priority which suits the following context. "There is an obstacle at a distance of 100m and the same is sensed by a sensor. Based on the sensor input, the display has to be displayed to indicate the function to be performed by the driver."
10.	An event sequence recorder has to be designed for a hospital in your city which will monitor a patient's pulse rate, blood pressure, body temperature. The equipment accepts inputs from different sensors, and prints the sequence in which they operate. It scans the inputs every millisecond and prints in a compact, type of event (normal or abnormal) and time of occurrence. It also communicates these events over an RS232C link to a remote computer. A real-time clock is included. Design the processor unit using 8086.
11.	Elderly users often forget their daily routines. Hence you need to design a microprocessor based unit to help them remember their monthly expenses and bill payments. For example, their house rent, telephone bills, electricity bills, gas requirement, etc. An alarm has to be blown to remind them and when they reset it, it is understood that they have paid and the expense has to be calculated for the entire month and at the end of the month the total expense has to be intimated.
12.	Let say that you work in VIT. Each day there is a rush hour in lunch time - everyone wants to get in the food line first. Your school is at the top floor and only way to get to the lobby is to use a lift. So, you call the lift and wait... and wait. Your waiting time could be infinite because everyone in bottom floors are loading the lift, so it never reaches the top! And when it finally does, your lunch time is over. Design a system to overcome this infinite waiting time.

		Total Laboratory Hours	30 hours
Recommended by Board of Studies	04-12-2015		
Approved by Academic Council	No. 39	Date	12-12-2015