VIVEKANANDA COLLEGE
THAKURPUKUR
KOLKATA-700063

NAAC ACCREDITED 'A' GRADE

Topic: Microcontrollers

Course Title: Microprocessor and Microcontrollers

Paper: GE-4

Unit:

Semester: 4

Name of the Teacher: Romela Dutta

Name of the Department: Electronics

FEB 2017 22 WEDNESDAY

08th Wk. 053-312

FEBRUARY-2017

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

Microcontroller (CYE-4)

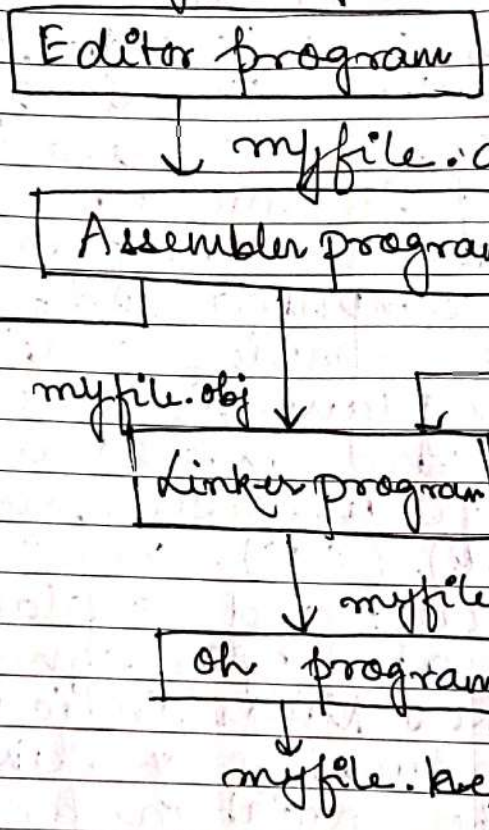
09.00

Assembling and Running of 8051 Program

10.00

Let us see the steps for creating, assembling and running an assembly language program are as follo

11.00



12.00

01.00

02.00

03.00

04.00

Editor program: At first, we use an editor fo type is a program. Editor like MSDOS program tha comes with all MS operating Systems can be used for creating or edit a program. The editor produces an ASC file. The asm extension for a source file is used by assembler during next step.

05.00

06.00

Assembler program: The 'asm' source file contain the code created in step I. It is transfered to an as assembler. The assembler is used for converting

Remember

the assembly language instructions into machine code instructions and it produced the .obj file (object file) and .lst file (list file). It is also called as source file because some assembler requires that this file must have ".src" extension.

• Linker program: The linker program is used for generating one or more object files and produces an absolute object file with an extension "abs".

• OH program: The OH program fetches the 'abs' file and fed it to a program called "OH". OH is called as object to hex converter it creates a file with an extension "hex" that is ready for burn in to the ROM.

Interrupts in 8051 microcontroller.

It is a sub routine calls that given by the micro-controller when some other program with high priority is request for acquiring the system bus as than Interrupt occur in current running program.

Interrupts provide a method to postpone or delay the current process, performs a sub routine task, and then restart the standard program again.

Let's see the five sources of interrupts in 8051 Microcontroller:

- Timer 0 overflow interrupt - TF0
- External hardware interrupt - INT0
- Timer 1 overflow interrupt - TF1

- External hardware interrupt - INT I
- Serial communication interrupt - RI/TI

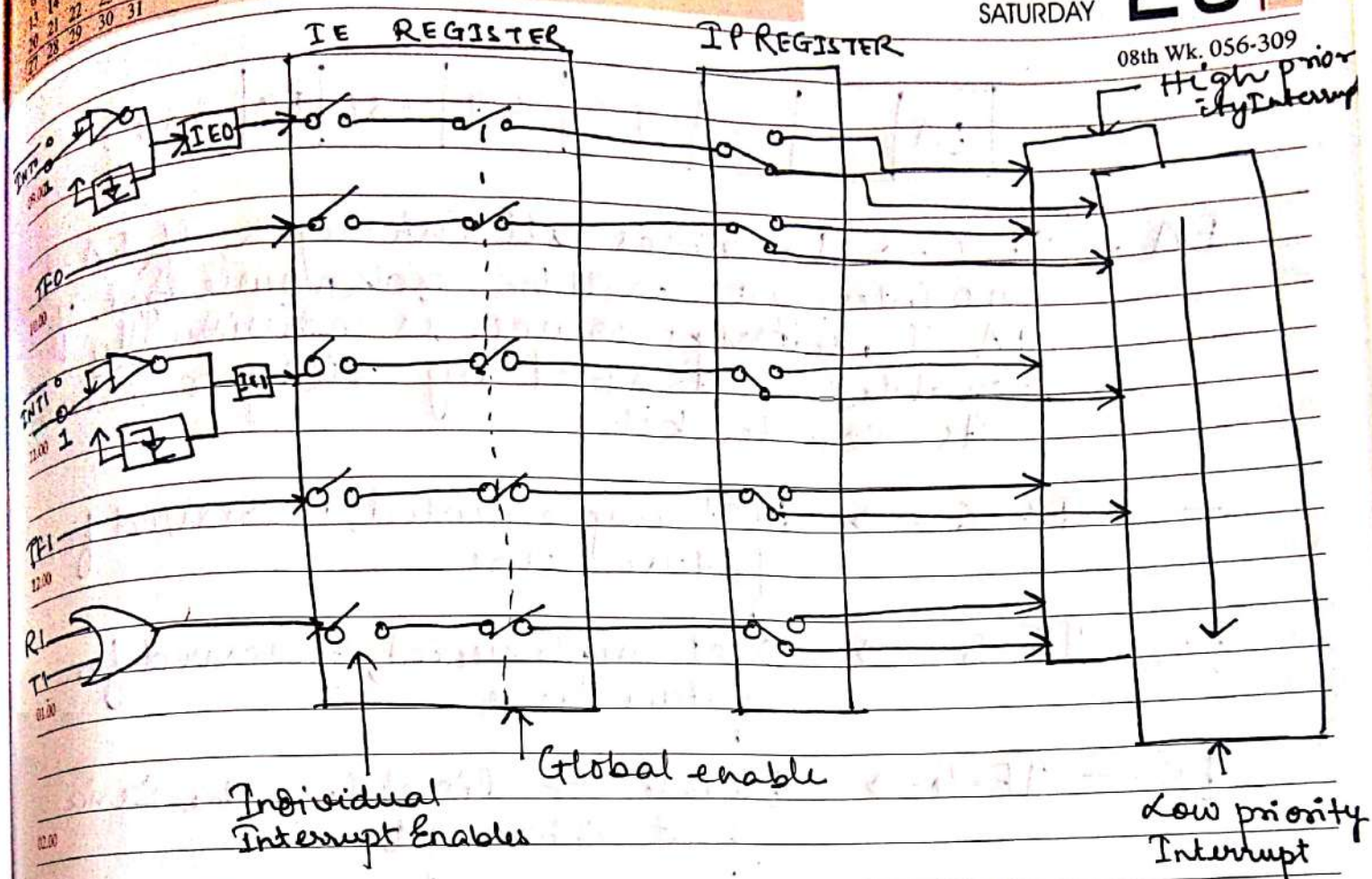
The timer and serial interrupts are internally produced by the microcontroller, whereas the external interrupts are produced by additional interfacing devices or switches that are externally connected with the microcontroller. These external interrupts can be level triggered or edge triggered.

When interrupt occurs, then the microcontroller executes the interrupt service routine. Therefore the memory location corresponds to interrupt enables it. Consider the interrupt corresponding to the memory location is shown in the table below.

Interrupt No.	Interrupt desc	Address
0	External INT0	0003H
1	TIMER/COUNTER 0	000BH
2	EXTERNAL INT 1	0013H
3	TIMER/COUNTER 1	001BH
4	SERIAL PORT	0023H

Interrupt structure of 8051 Microcontroller

After 'RESET' all the interrupts get disabled, and therefore, all interrupts are enabled by software. From all the five interrupts, if anyone or all interrupts are activated, this will set the corresponding interrupt flag as represented in the fig



All the interrupts can be set or cleared by some special function register that is also known as interrupt enabled (IE) and it is totally dependent on the priority which is executed by using interrupt priority register.

Sunday 26

Interrupt Enable (IE) Register

IE register is used for enabling and disabling the interrupt. This is a bit addressable register in which EA value must be set to one for enabling interrupts. The individual bits in this register enables the particular interrupt like timer, serial and external inputs. Consider the below IE register bits corresponds to 1 activate the interrupt and 0 disable the interrupt.

Wk. 058-307

EA	-	-	ES	ET1	EX1	ET0	EX0

EA - IE.7 → Disables all interrupts, If EA = 0, no interrupt will be acknowledged. If EA = 1, interrupt source is individually enabled or disabled by setting or clearing its enable bit.

\- IE.6 → Not implemented, reserved for future use.

\- IE.5 → Not implemented, reserved for future use.

ES - IE.4 → Enable or disable the Serial port interrupt.

ET1 - IE.3 → Enable or disable the Timer 1 overflow interrupt.

EX1 - IE.2 → Enable or disable external interrupt 1

ET0 - IE.1 → Enable or disable Timer 0 overflow interrupt

EX0 - IE.0 → Enable or disable external interrupt 0.