24

Scheme	Academic Year
18	2021 - 22.

NATIONAL EDUCATION SOCIETY (R.)

Jawaharlal Nehru New College of Engineering

Savalanga Road, Navule, Shivamogga -577204

(Approved by AICTE, New Delhi, Recognized by Govt. of Karnataka and Affiliated to VTU, Belagavi)

ASSIGNMENT BOOK

Details to be filled by Student:

NAME OF THE STUDENT:	INDHUSHREE L.S				USN:	4JN20EC036.	
NAME OF THE PROGRAM :	(~~Microcontroller~~) (B.E) ECE						
SEMESTER:	4th	SECTION:	A	COURSE CODE:	18EC46	COURSE NAME:	Microcontroller.

DETAILS OF ASSIGNMENT MARKS

Date	Assignment / Other Activities No.	Max. Marks	Max. Obtained
25-6-22.	Assignment 1.	5	5
13-8-22	Assignment 2.	5	5
Final Marks		10	10

Signature of Faculty

Signature of the HOD

Professor & Head
Dept. of Electronics & Communication Engg.
JNN College of Engineering
SHIVAMOGGA-577 204.

Assignment 1

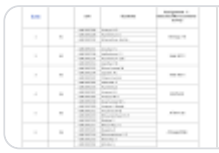
Dr. Pramod Kumar S • 16 Jun 2022 (Edited 3 Sept 2022)

CO mapped:
CO5: Design microcontroller-based system
Max. Marks: 5

Assignment as an activity to conduct a survey on various type of processors and its architecture, basic programs on the respective processors could be designed and executed using any open source platform.
Assignment book must include

- 1. Processor description
- 2. Abstract (overview) of the processor
- 3. Architecture diagram
- 4. Programs for that processor with results.

Submit on or before 25/06/2022



Microcontroller Assig...
Excel

Class comments

Add class comment...

Assignment 1

INTEL 8031 MICROCONTROLLER

Abstract ('overview) of the processor :

Intel 8051 is member of Intel MCS-51 family of 8-bit microcontrollers. 8031 have the same integrated peripherals as 8051 MCUs. 4 I/O ports, two 16-bit timers/counters, on-chip Oscillators & a serial port. The MCUs have 128 bytes of internal RAM, & in addition to that can utilize up to 64 KB of external data memory. This Microcontrollers don't have on-chip ROM, & must use external program memory.

Intel 80C31 is CMOS version of 8031 NMOS, and this Microcontroller; is fully pin- and object-code compatible with 8031. The 80C31 offers lower power consumption than the 8031 and adds two low power modes:

Idle mode turns off CPU core, but keeps peripherals & RAM active & power-down mode turns off CPU core & peripherals, and keeps only RAM memory active.

The manufacturers of this 8031 MC are AMD, Intel, Signetics

Intel 8031 is a single board Microcontroller configured around the most popular Intel 8031/8051, 8 bit Single chip microcontroller. 8031 can be used to train Engineers about the architecture, instruction set & capabilities of 8031 chip & also for actually controlling any industrial process etc.

This kit communicates with the outside world through a key board having 28 hex keys & seven segment display. This MC has the capability of interacting with an IBM PC compatible PCXT system.

This Microcontroller provides 32K byte of RAM & 32K bytes of EPROM. The total On board program & data memory can be very easily expanded upto 128K bytes in an appropriate combination of RAM & ROM. The monitor is incorporated from 0000-1FFF & the necessary 32K bytes of RAM has an address of 2000-3FFF.

The input/output structure of 8031 provides 48 programmable I/O lines. It has got 16 programmable Timer for generating any type of counting etc.

The on board resident system monitor software is very powerful & provides various software utilities. The kit provides various powerful software commands like INSERT, DELETE, BLOCK MOVE, RELOCATE, STRING, FILL & MEMORY., COMPARE etc. which are very helpful in developing the software.

The system also provides a serial monitor covering most of the command available in keyboard mode. A help menu makes the monitor more use freendly.

. 8031 is configured around the intentionally adopted Bus which is the most popular bus for process control & real time applications. All the address, data and control lines are available at the FRC connector. The kit is fully expandable for any kind of application. It has onboard battery backup to store the program incase of power failure.

Processor Description :-

The microcontrollers reading user's programs & executing the expected task as per instruction is stored there in. Its CPU contains ALU, Accumulator few more 8 bit registers, B register, Stack pointer SP, program status word (PSW) & 16 bit registers, Program counter (PC) & Data pointer register (DPTR).

The ALU performs arithmetic & logic functions on 8 bit input variables. Arithmetic operations include basic addition, subtraction, multiplication & division. Logical operations are AND, OR, Exclusive OR as well as rotate, clear, complement and etc. Apart from all the above, ALU is responsible in conditional branching decisions, and provides a temporary place in data transfer operations within the device B register is mainly used in multiply & divide operations. During execution, B register either keeps one of the two inputs & then retains a portion of the result. For other instructions, it can be used as another general purpose registers. Program status word keeps the current status of ALU in different bits. Stack pointer is an 8 bit register. This pointer keeps track of memory space where the important registers information are stored when the program flow gets into executing a subroutine. The stack portion may be placed in any where in the on chip RAM. But normally SP is initialized to 07H after a device reset & grows up from the location 08H. The SP is automatically incremented or decremented all PUSH or POP instructions and for all subroutine calls & returns.

Program counter is 16 bit register giving address of next instruction to be executed during program execution & it always points to the program memory space. Data pointer (DPTR) is another 16 bit addressing register that can be used to fetch any 8 bit data from the data memory space. When it is not being used for this purpose, it can be used as two eight bit registers

The 8031 I/O port structure is extremely versatile & flexible. The device has 32 I/O pins configured as 4 8bit parallel ports (P0 P1 P2 & P3). Each pin can be used as an i/p or as an o/p under software control. These I/O pins can be accessed directly by memory instructions during program execution to get required flexibility.

To access external memory it uses P0 as data bus P2 ports access external data & program will output the higher order byte on P2 during read cycle. Remaining P1 & P3 are available for standard I/O functions. But all the 8 lines of P3 support special features. Two external interrupt lines are there.

Each 8031 MC contains a high speed full duplex means you can simultaneously use the same port for both transmitting & receiving purpose

When it comes to program memory is extremely large & never lose information when power is removed. The program memory has a 16 bit address & any location is addressed using 16 bit PC.

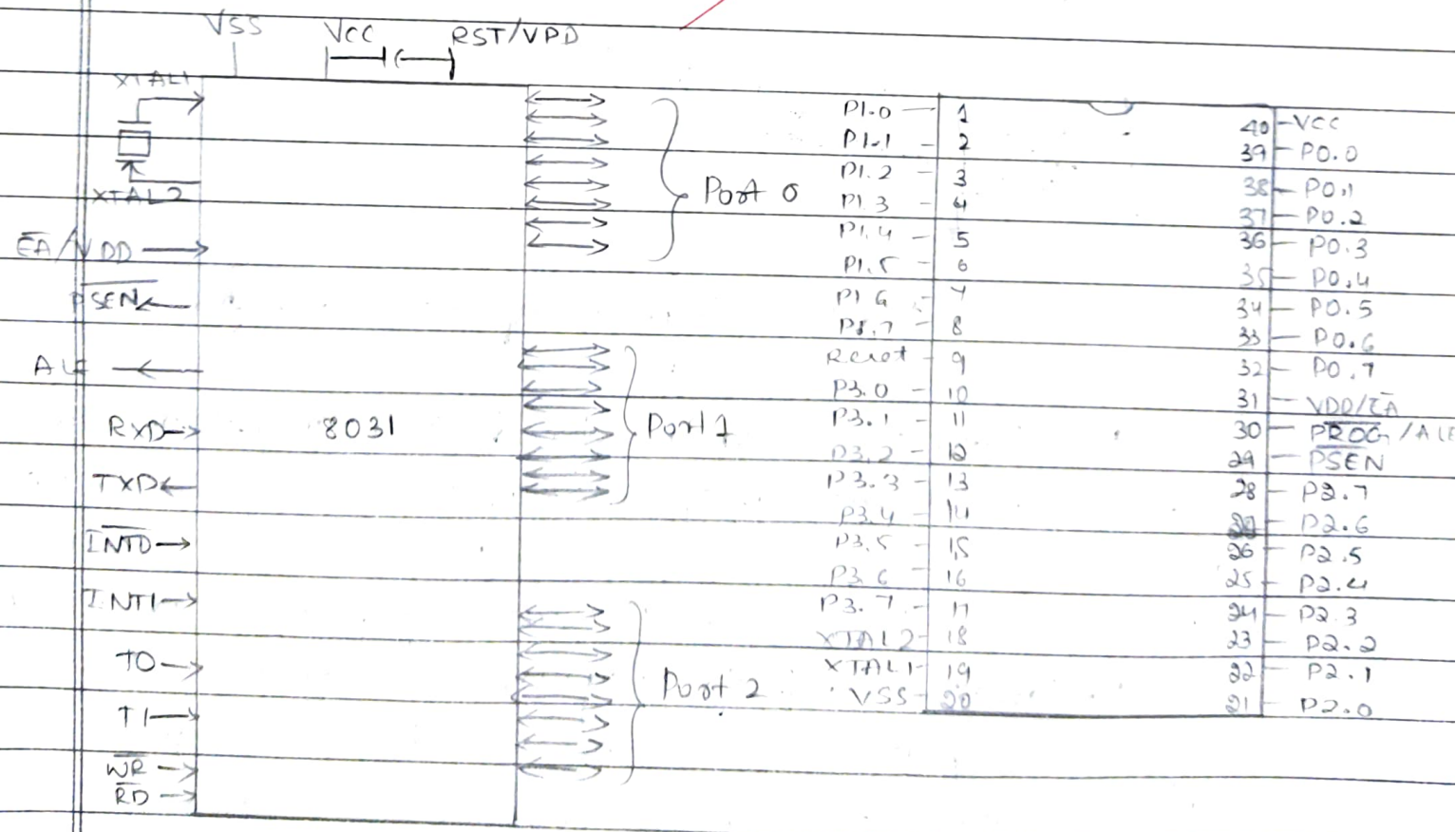
On chip data memory is smaller & therefore quicker than Program memory & it goes into a random state when power is removed. Onchip RAM is used for variables which are calculated when the program is executed.

In contrast to Program memory, On chip data memory accesses need a single 8 bit to specify unique location.

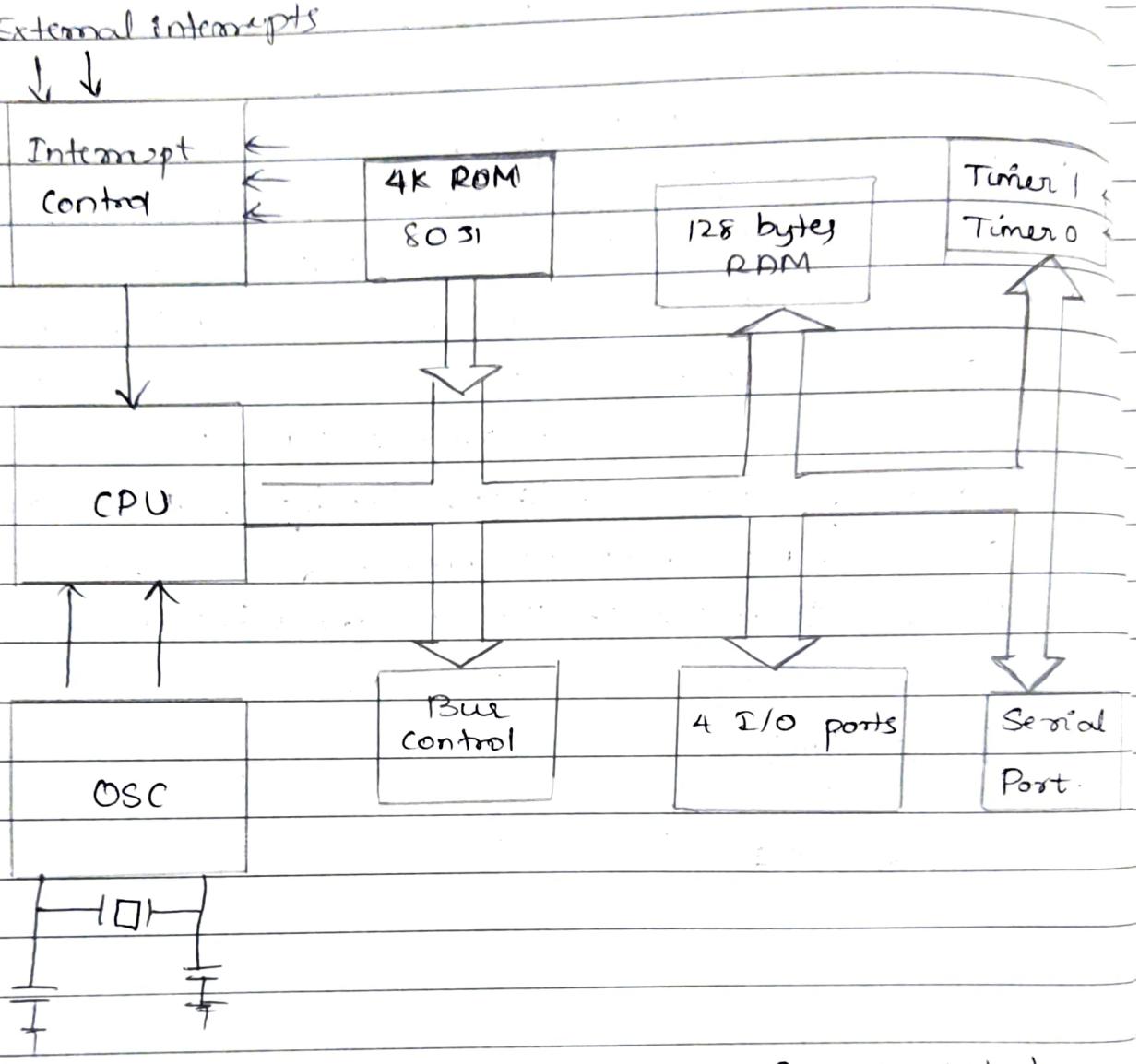
The 8031 have five interrupt sources: one from the serial port when a transmission or reception operation is executed; two from the timers when overflow occurs & two come from the two por i/p pins INT0 & INT1.

Manufacturers modified the basic 8031 architecture & added many new peripheral functions to make them attractive to the designers. known as 8031 derivatives.

Pin diagram:



8031 Architecture :



Block diagram of 8031 Core Architecture

8031 Architecture was introduced with onchip one time programmable version of program memory of size 4k x 8. Intel delivered all these MC (8031) with user's program fused inside the device. The memory portion was mapped at the lower end of the program memory area. But after getting devices, customer couldn't change any thing in their program code, which was already made inside during device fabrication.

The generic 8031 architecture sports a Harvard architecture which contains two separate buses for both program & data. So it has two memory space of 64K X 8 size for both program & data. It is based on an 8 bit central processing unit with 8 bit Accumulator & another 8 bit B register as main processing blocks. Other portions of the architecture include few 8 bit & 16 bit registers & 8 bit memory locations

Each 8031 device has some amount of data RAM built in the device for internal processing. This area is used for stack operations & temporary storage of data.

This base architecture is supported with onchip peripheral functions like I/O ports. Timers/counters versatile serial communication port. So it is clear that this 8031 architecture was designed to cater many real time embedded needs.

Features of 8031 architecture :-

* optimized for 8 bit CPU for control applications
* Extensive Boolean processing capabilities
* 64K program memory address space
* 64K Data memory address space
* 128 bytes of onchip data memory
* 32 Bi-directional & individually addressable I/O lines
* Two 16 bit timer/counters
* Full Duplex UART
* 6-source/5-vector interrupt structure with priority levels.
* Onchip clock oscillator.

Programe for 8031.

1)
```
MOV 81, #65
MOV A, #44
LCALL 1608
LIMP 12DB
```

This program displays a character on CRT when ASCII Code is in Accumulator.

2)
```
MOV R0, #04H
MOV R1, #05H
MOV A, R0
ADD A, R1
MOV DPTR, #3012H
MOVX @DPTR, A
SJMP 300A.
```

The result at the address 3012 will be 09H.

3)
```
MOV R0 #003H
MOV R1, #02H
MOV A, R0
MOV B, R1
MUL AB
MOV DPTR, #3012H
MOVX @DPTR, A
SJMP 300C
```

The result in the 3012 address will be 06H.

4.
```
MOV R0, #05H
MOV R1, #04H
CLR C
MOV A, R0
SUBB A, R1
MOV DPTR, #3012H
MOVX @DPTR, A
SJMP 500B
```

The result will be 01H at the address.

5.
```
MOV R0, #08H
MOV R1, #02H
MOV A, R0
MOV B, R1
DIV AB
MOV DPTR, #3012H
MOVX @DPTR, A
SJMP 300C
```

The result will be 04H at 3012 address

26.6.

Roll No.
48

Academic Year
2021-22

NATIONAL EDUCATION SOCIETY (R.)

J N N College of Engineering

Savalanga Road, Navule, Shivamogga -577204

(Approved by AICTE, New Delhi, Certified by UGC 2f & 12B, Accredited by NAAC –'B',
* UG Programs : CE, ME, EEE, ECE, CSE, ISE, TCE. Acredited by NBA : 1-7-2019 to 30-6-2022.
Recognized by Govt. of Karnataka and Affiliated to VTU, Belagavi)

ASSIGNMENT BOOK

Name: Vidyashree.R. **Semester** VII **Section:** B

Programme: B.E **USN:** 4 J N 1 8 E C 1 1 2

Course Code: 18EC73 **Course Name:** DIP

DETAILS OF ASSIGNMENT MARKS

Date	Assignment / Other Activities No.	Max. Marks	Max. Obtained
12/12/21	A-1	5	5
01/02/22 ~~04/2/22~~	A-2	5	5
Final Marks		10	10

Signature of Faculty

Signature of the HOD

Professor & Head
Dept. of Electronics & Communication
JNN College of Engineering
SHIVAMOGGA-577 204.

JAWAHARLAL NEHRU NATIONAL COLLEGE OF ENGINEERING- SHIVAMOGGA

Department of Electronics & Communication Engineering.

Subject:- *DIGITAL IMAGE PROCESSING (18EC733)*

Section: 7th

CO: 4.

Assignment-2

1. Write a code to blur your selfie image. The code should be executed in any of the computation tools. The executed code along with the input image and output image should be depicted in the assignment book. The code should contain comments.

2. Write a code to sharp your selfie image. The code should be executed in any of the computation tools. The executed code along with the input image and output image should be depicted in the assignment book. The code should contain comments.

ASSIGNMENT-2

1. Write a code to blur an input image. The code should be executed in any of the computation tools. The executed code along with the input input image and output image should be depicted.

```
# Python Program to blur image

#Importing cv2 module
 import cv2

# photo.jpg is my image
img = cv2.imread('photo.jpg')

blurImg = cv2.blur(img, (10, 10))
cv2.imshow = ('blurred image', blurImg)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

OR

```
import cv2
from matplotlib import pyplot as plt
image = cv2.imread('photo.jpg')
blur = cv2.blur(img(5,5))
plt.subplot(121), plt.imgshow(img), plt.title('Original')
plt.subplot(122), plt.imshow(blur), plt.title('Blurred')
plt.show()
```

OR

```
#importing opencv cv2 module
import cv2

#photo.jpg is the image which needs to be blurred.
img = cv2.imread('photo.jpg')

cv2.imshow('Original Image', img) #Display image

#Averaging
#Kernel size can be changed as per requirements
avg = cv2.blur(img, (10, 10))

cv2.imshow('Image after Averaging', avg)
cv2.waitKey(0)

# Gaussian Blurring
# Again, Kernel can be changed.
GaussBlur = cv2.GaussianBlur(img, (5,5), 0)

cv2.imshow('Image after Gaussian Blurring', GaussBlur)
#Display image after Gaussian Blurring

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Input

Output

Original Image | Image after Averaging | Image after Gaussian Blurring

2. Write a code to sharp an input image. The code should be executed in any of the computation tools. The executed code along with the Input & output image should be depicted.

```
# load the required packages
import cv2
import numpy as np

# load image into system memory
image = cv2.imread("photo.jpg", flags=cv2.IMREAD_COLOR)

# display the original image to the screen.
cv2.imshow('Original Image', image)
cv2.waitKey()
```

```
# kernel can be changed as per requirement
kernel = np.array ( [[ 0, -1, 0 ],
                     [-1, 5, -1],
                     [0, -1, 0 ]])

image_sharp = cv2.filter2D ( src = image, ddepth = -1,
                             kernel = kernel)

# display the sharpened image on the screen
cv2.imshow ('sharpened Image', image_sharp)
cv2.waitKey ()
cv2.destroyAllWindows()
```

Input

Output

Original Image

Image after Sharpening.

S.2