Experiment - 8

AIM: Interfacing exercise on 8253 PIT.

APPARATUS REQUIRED: 8085 kit and power supply

THEORY: The 8253 is a programmable interval timer and can be programmed to generate accurate time delays under software control. The use of this chip avoids the timing loops in the system software and provides multiple time delays. This chip also provides counter/timer function apart from generation of delays. The 8253 is available in 24 Pin DIP package. It is a programmable interval timer. It has three independent 16 bit counters. The 8253 operates in the following six modes -

i) Mode 0 Interrupt on terminal count.
ii) Mode 1 Programmable one shot.
iii) Mode 2 Rate Generator.
iv) Mode 3 Square Wave Generator
v) Mode 4 Square Triggered strobe
vi) Mode 5 Hardware Triggered strobe

The function of the counters of 8253 can be programmed by the system software.

PROGRAM: Connect 2 MHz clock at CLK2 input through D-flip flop and +5V at Gate2 input of the PIT 8253. Then for programming the PIT 8253 to produce a square wave of frequency 20 Hz at the output 2 of PIT 8253

Teacher Sign. .....

counter 2 :

Counter Count : The required square wave frequency = 20Hz. Hence, time period = $\frac{1}{20}$ sec = 50 msec.

The D-flip flop divides the input clock frequency by 2. Thus the clock frequency input to counter 2 of 8253 is 1MHz and the time period is 1 μsec. Thus, the counter count is

$$\text{Count} = \frac{50 \text{ msec}}{1 \mu\text{sec}} = 50000\,D = C350H$$

Control Word : For square generation, the counter 2 should operate in Mode 3. The counter starts counting as soon as count is loaded into the counter.

Procedure :

1. Do the connections as required for the CLK2 and GATE 2 input of the PIT 8253.
2. Connect the supply to the kit.
3. Enter the program.
4. Execute the program.
5. Observe the square wave output at OUT2 output of the PIT.

Experiment - 9

AIM: Interfacing exercise on 8279 programmable KB/display interface to display the hex code of the key pressed on display.

Apparatus Required: 8085 kit and power supply

Theory: The 8279 programmable keyboard/Display interface is a hardware approach to interface a matrix keyboard and a multiplexed display with using 8279 in checking the keyboard and refreshing the display. In a software approach to interface microprocessor with keyboard and multiplexed display, microprocessor remain busy for a lot of time in checking the keyboard and multiplexed display.

The 8279 can operate without involving the microprocessor, except for relatively short time when actual data transfer between I/O devices and CPU takes place. It relieves the CPU from the normal house keeping job with I/O devices. 8279 can perform two important functions.

i) It scans a keyboard, detects any key press and transmits to the CPU information.

ii) It puts out data received from the CPU for the use of display devices.

The 8279 is a 40 Pin device and has two major sections - keyboard and display.

The keyboard section can be connected to a 64 key matrix in a number of modes. The display section can provide a 8 or 16 seven segment LED displays. It can be also used in other types of displays i.e. incandescent and other popular display technologies. It has 16x8 display RAM, which can be used to read/write information for display purposes. The display can be set up in either right entry or left entry format.
A matrix keyboard (8x8 maximum) and a display (sixteen seven-segment maximum) can be interfaced with 8085 microprocessor through 8279.

Control words : (i) Control word for Mode setting : Control word to set the keyboard/display mode.

(ii) Control Word for Program Clock :
All timing and multiplexing signals for the 8279 are generated by an internal prescaler. This prescaler divides the external clock (at pin 3) by a programmable integer.

(iii) Control Word to Read FIFO RAM :
The control word to read scan code of pressed key which is stored in FIFO RAM is

0 1 0 AI X A A A = 40H

0 1 0, 0 0 0 0 0

Fixed Code for Scan keyboard Mode

The format of data read will be

MSB LSB

D_7 D_6 D_5 D_4 D_3, D_2 D_1 D_0

CNTL SHIFT Row Column

In Hex keyboard, the keys are placed such that when CNTL & SHIFT keys are not pressed, the read data is the hex code of the pressed key.

(iv) Control Word to Write Display RAM:

CPU sets up the 8279 for a write to Display RAM by first writing this command. All subsequent data writes will be to the display RAM. The RAM address is auto incremented if AI = 1.

1 0 0 AI A A A A

1 0 0, 1 (↓) 0 1 0 0 = 94H

Fixed Code for Auto increment To select fifth Display from left

Program:

First two instruction of the program will initialize the display as 8 8-bit character, left entry and key board as encoded scan, 2-key lockout. Next two instructions will initialize the FIFO RAM for read operation. Instruction IN 18H will read the FIFO RAM and stores the contents (Hex code of the pressed key) in accumulator. Instruction MVI A, 94H and OUT 19H will initialize RAM to write onto 5th display. Instruction OUT 18H will write the pressed key hex code onto display RAM and thus 5th display will show the hex code of the pressed key hex code onto display RAM and thus 5th display will show the hex code of the pressed key in coded form.

Procedure:

1. Connect the power supply to the 8085 kit and switch on the power supply.
2. Enter the program.
3. Execute the program.
4. Check the result.

Experiment - 10

Aim : Study of 8085 emulator for hardware testing.

Apparatus Used : 8085 In-circuit Emulator, 8085 - kit & supply.

Theory : In-circuit emulators used for designing a microprocessor based system (both hardware and software designing). In-circuit emulation is a execution of a prototype software program in prototype hardware in control of a software development system. For this, the microprocessor is removed from the microprocessor based system under test or to be designed. A 40-pin cable from the In-circuit emulator is plugged into the socket previously occupied by the microprocessor. The emulator performs all the functions of the replaced microprocessor and in addition, it allows the system under test to share all its resources such as software, memory and I/O devices. It provides a window for looking into the dynamic real time operation of the prototype hardware.

To test subsystems (such as memory, I/O etc), the minimum prototype hardware required is 40 Pin microprocessor socket (without the microprocessor) and the power

supply. All other resources can be borrowed from the emulator. As more and more hardware is added to the system, fewer resources from emulators will be required.

Emulator can be a stand-alone unit or a part of software development system. A small program can be directly entered into the emulator or can be transferred into emulator from software development system. Once a program is loaded, a user can interact with the emulator through its keyboard.