National University of Singapore

Department of Electrical and Computer Engineering

CG2007 – Microprocessor Systems

AY2011/2012 Semester 2

Major Project 2: Interrupt System

1. Objectives

1) To design and develop an **elevator system** using 80188 microprocessor on a PCB.

2) To learn how to use CPU interrupt controller to handle hardware interrupt requests in assembly language.

2. Components and Equipment Needed

- ✓ All the tools, chips and hardware components distributed
- ✓ SuperPro500P EEPROM Programmer
- ✓ DC Power Supply

3. Project Requirements

In major project 1, you have designed and implemented a microprocessor system for basic I/O control. In this project, you will build an industrial elevator system for engineering applications. The system shall be able to control the elevator to head up/down to a specific storey according to the user's command. Minimum number of storeys required for this application is 16 (i.e. 0~15). User can specify the wanted storey by using DIP switch which is connected to the microprocessor system. The elevator starts to move immediately after a "GO" button is pressed. Detailed specifications of this system are as follows.

1. The required storey from the DIP switch input (4-bit data) shall be displayed in the system. It is recommended to be displayed using 4-LEDs in Port_A of 82'55. You may also construct external circuits for this display.

2. Current storey shall be displayed using 7-segment components. This number will change from time to time. When RESET is activated, the current storey must be reset to Storey 1 (ground floor) or Storey 0 (basement), based on your preference. When the GO button is pressed, the current storey shall approach and finally stops at the required storey. **Speed of this elevator should be set to roughly 1 second per storey**, to make the system user friendly.

3. While the elevator is rising, an LED indicator should be blinking to represent its movement. You may use one of the LEDs connected to Port_A of 82'55. Another LED indicator shall be designed when the elevator is falling.

The GO push-button interrupt should be edge-triggered, instead of level-triggered. Please note that during the interrupt period, the system should **NOT** respond to any further interrupts until the current interrupt service routine is completed served. That is, any pressing of the interrupt push-button should not be entertained within this period. The "RESET" push-button switch should be able to reset the microprocessor any time.

In addition to the basic requirements mentioned above, you can develop innovative extra features which are suitable for the elevator system. For example, you can make the system working for higher buildings (32 storeys, 64 storeys or even more). You may also design a buzzer circuit such that the system will produce sound when the elevator reaches the required storey. **Extra features will be credited to the project assessment.**

4. Steps to complete the project

You can use the maskable hardware interrupts (INTR) for this project. The available interrupt pins on the 80C188 microprocessor are INT0, INT1, INT2 and INT3. Since the microprocessor already has an interrupt controller inside, you do not need to use the 82C59 PIA chip.

In the software part, you should perform the following operations in your program:

1. According to the interrupt pin you are using, find the Peripheral Control Block (PCB) offset address of the corresponding Interrupt Control Unit Register. This register will allow you to define the behavior of the interrupt source. Assign proper values to the register. Please refer to Section 8.4 and Section 4.1 of the 80C188 User Manual for details. (*Hint: the offset address of the ICU register for INT0 is 0FF38H.*)

2. The maskable interrupts INT0-INT3 have fixed interrupt types. You need to find the interrupt type number according to the interrupt pin you use, and therefore calculate the interrupt vector table address for your interrupt. Please refer to Table 8-2 of the 80C188 User Manual for the interrupt type.

3. Write your interrupt service routine (ISR) for the elevator system. You need to make use of "CLI" and "STI" instructions to mask or allow the hardware interrupt at appropriate situations. After "interrupt display" period is finished, an End-of-Interrupt (**EOI**) command needs to be issued to the Interrupt Control Unit. You need to find the Peripheral Control Block (PCB) offset address of the EOI Register and assign proper values to the register. Please refer to Table 8-3 for the offset address and Section 8.4.7 for more details of

the EOI Register. Here is a skeleton code to issue EOI command.

```
MOV     DX,     EOI     ;EOI is the offset address
MOV     AX,     <INTERRUPT TYPE NUMBER>     ;your interrupt
OUT     DX,     AL
```

Last but not least, "**IRET**" should be issued at the end of the ISR, so that control can be returned to main program.

4. Link your ISR and the Interrupt vector table together. That is, the starting address of your ISR should be stored in the corresponding interrupt vector table location.

You may approach your Graduate Assistant for difficulties in debugging. Please plan your schedule well so that you do not struggle till the last minute.

5. Project Demonstration and Submission of report

Demonstration Date: Week 13 (starts from 9th April).
You need to demonstrate your system to your respective GA during the lab session in Week 12. In addition to verifying the functionality of your system, your GA will ask you several questions to test your knowledge about the system.

Report Submission Date: Week 13 (same day of your demonstration).
You will submit your report (hard copy) for major project 1 and 2 together (**ONE report in total**). Submit your report to your respective GA at the end of your demonstration for major project 2.

Note: Copying should be strictly avoided. If any circuit/report is found copied, then the candidate must face the disciplinary action from the Department/University.

THE END