

COE 538 Lab2 Report

The provided assembly code demonstrates basic interactions between input/output peripherals and the microcontroller, such as reading switches, keypads, controlling LEDs, and generating a tone.

Part 1: Reading Switches and Displaying on LED1

The first section reads the state of a switch (SW1) on Port T and displays it on an LED connected to Port H. The code enables pull-up resistors for the switch (**STAA PERT**) to maintain a stable logic signal. Inside a loop, **LDAA PTT** reads the switch state, and **STAA PTH** displays it on the LED, continuously updating in real time.

Part 2: Reading Keypad and Controlling LED2

The second part reads a keypress from a keypad connected to Port S. After shifting the key code to isolate the relevant bits (**LSRA**), these bits are stored in Port P to control an LED (**STAA PTP**). The loop ensures the LED responds to key presses dynamically.

Part 3: Generating a Tone

The final section generates a tone by toggling the most significant bit of ACCA using **EORA**, which drives a pin connected to a buzzer on Port P. A delay loop (**LDX, DEX**) controls the frequency of the tone by regulating the timing of the toggling action.

Conclusion:

The provided codes exhibit how fundamental assembly language instructions can be used to interface with external hardware components like switches, keypads, LEDs, and buzzers. The first part focuses on reading inputs and displaying their states, while the second part processes keypad input to control outputs. The third part illustrates how software can generate a tone by manipulating signal timing and pin states. Each segment reflects the low-level control necessary in embedded systems to interact with peripheral devices effectively.