Columbia University
Department of Electrical Engineering
EECS E4340. Homework/Laboratory #1.
Instruction-Set Architecture and Assembly Language Programming of the PDP/8.
Due: February 4, 2004 (5 PM) in class

In this laboratory, you will learn about the PDP-8 instruction set and about PDP-8 Assembly Language (PAL) programming. For reference, please refer to the "Programmer's Reference Manual for the PDP-8" and chapters 7 and 9 of Prosser and Winkel (both contained in the course bluebook). Although the emulator we are using is for a PDP-8E, we are assuming that we can employ only the capabilities of the PDP-8I (in particular, we only have 12 bits of address for 4K 12-bit words of memory, no extended arithmetic element, and no group-three microcoded instructions). We will be employing only two I/O devices: console TTY input-output and high-speed paper tape input-output. We will use the latter with the emulator to load programs by mounting a file onto the paper tape device. When we actually implement the PDP-8I in hardware, we will only support the TTY device through a serial port and we will use this device to load software as well.

While the PDP-8 machine is old (1967), working with PAL is not entirely an academic exercise. For example, forty PDP-8 machines still control the signs in the stations at BART (Bay Area Rapid Transit). One of the current PAL assemblers was actually written for use by BART to modify the software for these machines.

For this lab, you will need to use the Sun machines in the VLSI CAD Lab (1218 Mudd), since the executable has not yet been ported to Linux. Example files are kept in `/tools2/courses/ee4340/public`. Software is kept in `/tools2/courses/ee4340/bin`, which can be added to your UNIX search path.

The program `square.pal` gives an example of PAL programming. Please note the extensive use of comments to document the program. This absolutely essential to make assembly code at all understandable by someone else.

- A block of comments at the top of the program identify what the program does and provide detail on inputs and outputs

- A block of comments are provided for each subroutine to provide the same information.

- Comments on groups of statements can provide "high-level" detail on the function performed by the group, while comments on individual lines can describe the contribution of that line.

- Comments on variables and block storage describe contents and usage.

First get familiar with the assembler and the emulator by assembling the `square.pal` program. To run the PAL assembler, type `pal -r square.pal`, where `square.pal` is the file name of your PAL program. A listing file will be generated in `square.lst`. You will also notice a file `square.rim`, which is your program in PDP-8 RIM (Read-In Mode) format. You will work with this in more detail below (or you can skip ahead now to Problem 3).

You should try to enter and run the program on the PDP-8 emulator. To bring up the emulator, type `pdp8e`. You will then get an X-windows display that resembles the front-panel of the actual machine. There are two rows of LEDs in the front panel. The first row displays the contents of the memory address register. The second row displays displays the contexts of the facility selected from the options: `STATE`, `STATUS`, `AC`, `MD`, `MQ`, and `BUS`. The most useful of these are `AC`, which displays the contents of the accumulator, and `MD`, which displays the contents of the memory data bus. You will now need to toggle in your program (in much the same way you will do once you have built the real machine). To do this, enter the first address into the switch register (the row of switches on the front panel). Down on the switches is logic '0', while up is logic '1'. Then hit "ADDR" to load the address. Next enter the data to be loaded to this address and hit "DEP" for deposit. The address will automatically increment and you can enter the data for your program sequentially from this point.

Another valuable feature of the emulator is the ability to save the memory image as a core file. This will allow you to restart the emulator and your program without having to re-key in everything through the front panel. To do this, click on the `+` sign in the top corner of the front-panel screen to select the `CORE` device. Then specify a file name in the field for your core file dump. When you bring up the emulator, you should be able to reattach a core file in the same way.

1. Write a program in PAL that converts a positive binary number less than decimal 100 into two binary numbers, one representing the ten's digit and the second representing the one's digit of a decimal representation. Assume that the binary number to be converted is located in memory location 4000_8. Store the ten's digit in location 4001_8 and the one's digit in location 4002_8. Test you program on the emulator. Please turn in the PAL listing of your program.

2. You will now write a program that uses console TTY input/output, such as you will also do on the PDP-8 you will build for this class. Examples of how to poll the terminal are found on pages 354-355 in Prosser and Winkel and were discussed in class.

Your program will accept a single digit between 0 and 9 from the console as input (terminated by a carriage return). Your program will then print out the square of the number in decimal form to the console (two decimal digits) followed by a carriage return and a line feed. The attached table describes the 7-bit ASCII code. The left-most bit in the byte going to and coming from the terminal should be zero. Test your program on the emulator and get this problem signed-off on the attached sign-off sheet, indicating that you have demonstrated a working program to the instructor or TA. For this problem, please also turn in your PAL listing file.

3. *Using the RIM loader.* In `/tools2/courses/ee4340/public/rim.pal`, you will find the PAL assembly code for the RIM loader. This routine is toggled into the machine and run to allow programs in RIM format to be quickly read into the PDP-8 through the TTY or paper-tape reader devices. From this PAL program, describe the RIM format. From the point-of-view of the PDP-8, what differences are there between the behavior of the TTY device and the paper-tape reader?

If you are ambitious you can try to use the RIM loader to read in the longer program you wrote in Problem 2 (rather than having to key it in through the front panel). To do this, you will need to key in the RIM loader and attach the RIM file generated by the assembler for your program as the `PTR` (for paper-tape reader) device.

Sign-off sheet

Problem #2 ______________________________