

CPTR-215
HW # 17 - Due Firday

The goals of this home work are to gain experience with:

- converting 16 bit binary numbers to binary coded decimal
- converting binary coded decimal numbers to ASCII
- writing characters on the LCD display

Write a subroutine for the ARM processor on an Embedded Artists LPC2148 board that will take a 16 bit binary number and display it in decimal notation (5 digits) on row 1 of the LCD display starting in the left column (column zero). To test and demonstrate it you will write a main program that calls the subroutine and passes to it a number to be displayed (this number can be “hard coded” into the main routine and changed for testing the subroutine by changing the source code). Pass the 16 bit number to your subroutine in register zero. For this version you will always write 5 digits even if it means there are leading zeros in the number that is displayed.

I suggest that you may wish to break the problem up into one or more subroutines that are used inside the overall subroutine. All the subroutines that are used should be included in the source file with the main program. As noted in class, program 8.3 in the textbook is a division routine that you can use to do the required division.

As you take the binary number and start determining what value each of the 5 decimal digits will have, recall that each decimal digit’s value ranges from 0 to 9. Note that it takes 4 bits to represent 9 in binary. This use of 4 bits to represent the value of a decimal digit is called Binary Coded Decimal (BCD). Thus if we wanted to we could put an 8 digit decimal number into a 32 bit register (4 bits per digit X 8 digits = 32).

I will be supplying the routines to be used to write to the LCD display. Here is a description of those:

```
;  
; Procedures you will call are:  
; lcdReset      must be called first. no parameters.  
; lcdClear      use to clear the display & home the cursor  
;               no parameters are required.  
; lcdWRdata     use to write one character on the display  
;               place an ASCII char in r0 before calling.  
;               Does not advance the cursor.  
; lcdCursorAt  use to move the cursor. Two parameters are passed to this routine  
;               r0 = row (1 or 2)  
;               r1 = column to move cursor to (0-15)  
;  
; Example use of the LCD routines:  
; bl lcdReset   ; do this once at the top of your program  
;  
; mov r0,#0x31 ; place the ascii code for number one in r0  
; bl lcdWRdata ; display the character '1' on LCD display  
;
```

```
;  
;      bl  lcdClear    ; clear the display  
;  
;  
;      To move the cursor do the following  
;  
;      mov  r0,#1      ; select LCD row 1  
;  
;      mov  r1,#4      ; select LCD column 4 (replace 4 with desired column #)  
;  
;      bl  lcdCursorAt ; move the cursor  
;  
;  
;  
;  
;  
;  
;  
;      NOTE: All registers are saved by these routines, i.e. upon return  
;  
;      from the subroutine all registers, except flags, have the  
;  
;      same value in them as before the subroutine was called (this  
;  
;      breaks with the ARM convention for subroutines but is done  
;  
;      to make it easier for you).  
;  
;  
;
```

Before writing assembly code design the program. Create NS diagrams or flow charts that show program operation. Then code and test. Testing the writing to the display will be difficult using the debugger, so testing the program on the hardware board (where you also can use the debugger) will be best.

As mentioned in class, as you find the 4-bit binary value for each decimal digit you can convert that to an ASCII code that then can be sent to the lcdWRdata subroutine. After writing each digit on the screen move the cursor one position to the right.

Turn in: hard copy of your design documentation and a program listing. On the program listing write a note describing the success (or failure) of your program and any particular issues you had in creating it. E-mail me a copy of your source file.