ECE 33 : Introduction to Computer Engineering
Printing a Binary Value as a Decimal Number

Program 4 requires you to print the final answers of sums to the screen. This note discusses printing such values.

Assume that H-L pair holds a binary string with value 3456. (In your problem, the value may be 5 digit long) Clearly from this string one should first extract the digit 3, then 4, etc. One can do this by repeatedly subtracting 1000 from the value *without* generating a carry (which indicates a borrow). The number of times we can do this gives the first digit. Note however that when you generate a carry, you have done one too many subtractions. One has to add back 1000 at this point to get the remaing part of the number. For the chosen number, the computation looks like:

$$3456 \xrightarrow{\text{-1000}} 2456 \xrightarrow{\text{-1000}} 1456 \xrightarrow{\text{-1000}} 456 \xrightarrow{\text{-1000}} -544 \xrightarrow{+1000} 456.$$

This procedure separates the first digit (with weight 1000) and the rest of the value.

Following flowchart of subroutine *print1* captures this.

```
; subroutine print1: to print the most significant digit
; input: H-L= value, B_C= weight of the position (say, 1000)
; output: prints the most significant digit, H-L= the rest of value
; registers destroyed: none (except H-L)

1. save the registers being destroyed
2. E <-- '0' (ASCII code because it will be printed)
3. H-L <-- H-L - B-C (use 8 bit subtraction with SUB and SBB)
4. if carry, goto 7.
5. E <-- E + 1
6. goto 3.
7. H-L <-- H-L + B-C
8. send E to screen to print the digit
9. restore registers
10. return
```

One can repeatedly call the above subrotine with different values to print all the digits. Thus to print a positive value with 5 digits, one will follow the suboutine on the next page.

```
; subroutine to print a five digit value
; input: H-L= value
; output: printing of the value
; registers destroyed: none

1. save the registers being destroyed
2. B-C <-- 10000
3. call print1
4. B-C <-- 1000
5. call print1
6. B-C <-- 100
7. call print1
8. B-C <-- 10
9. call print1
10. print value in L register
11. restore registers
12. return
```