HUMBER COLLEGE
COMPUTER SYSTEMS ARCHITECTURE **LAB 7**

CPU

The goal is to write a C program that will simulate a simple CPU. The simulation will be at a register level (i.e. will simulate register to register transfers). Your program should have a simple user interface that would allow it to be used to test machine level programs written for the CPU (read a binary file into memory, display memory, single step instructions, display registers, etc.) While single stepping the program should display the registers after the end of each instruction cycle.

Today's Assignment

1. Continue implementing the execute stage of the CPU. In particular, implement:
    a. the Load/Store instructions,
    b. the Stop instruction,
    c. the sign, zero and carry flag (see note below.)

Carry

A carry is produced when an addition produces a result that is greater than the register size. If we consider the ADC instruction we see that there are 3 values being added together.

Operation	Code	Description	Flags NCZ
ADC	0101	Rd := Rd + Rn + C	ncz

It would appear that detecting this would be simple:

```
if (Rd+Rn+C > MAX)
   CF=1;
else
   CF=0;
```

However, when you consider the situation with 32-bit registers and the code running on a 32-bit machine this code will not work. The following code is a suggested solution.

```
/**********************************************************
  iscarry()- determine if carry is generated by addition: op1+op2+C
  C can only have value of 1 or 0.
 **********************************************************/
int iscarry(unsigned long op1,unsigned long op2, unsigned C){
   if ((op2== MAX32)&&(C==1))
      return(1); // special case where op2 is at MAX32
   return((op1 > (MAX32 - op2 - C))?1:0);
}
```