60-265-01 School of Computer Science
Project: Unsigned Integer Division Circuit

This Project is mandatory for all students registered in the course. The mark for the Project is 10% of the course grade and the Project must be submitted for assessment no later than Thursday November 22, 2012 – all Project submissions will be assessed in the laboratory during scheduled laboratory times.

For this Project it is assumed that all students are quite familiar, and practiced, in applying the technique of *long-division* to numbers expressed in decimal (ie. base-10). For all positional representations of non-negative integers, the basic algorithm for long-division is the same.

For this Project each student must complete the design and representation of an integer division circuit (called DIV_32by16) that accepts two unsigned binary inputs, a 32-bit non-negative dividend and a 16-bit non-negative divisor. The outputs from the circuit normally consist of a 16-bit non-negative quotient and a 16-bit non-negative remainder; however, there are some exceptions to this and to deal with the exceptions there are three additional flag-register (that is, single flip-flop registers) outputs, namely: (a) Z-flipflop that is 0 if the quotient and remainder are both zero and 1 if either is non-zero; (b) O-flipflip that is 1 if the size of the quotient exceeds 16 bits (Overflow) and the remainder cannot be properly computed, and 0 if both the quotient and remainder can be expressed in 16 bit formats; and (c) U-flipflop that is 1 if the divisor is zero (hence the division operation is Undefined) and 0 if the division operation can be carried out (with or without overflow).

The approach and design are left to each student to determine, including defining single flip-flop and multiple flip-flop registers and other specialized combinational and sequential circuit elements, as required by the approach.

The solution must, however, incorporate the following logical aspects:

1. The original data (ie. dividend and divisor) is located in the computer RAM (ie. memory) and must be loaded into the registers used for the division hardware located within the CPU. Assume that the RAM is divided into 16-bit byte units so that a single load operation moves 16-bits from a designated address to a designated CPU register. Since the dividend is assumed to be 32-bits, it is located in two successive RAM locations and treated logically like a 32-bit non-negative binary number. The divisor is located in another, arbitrary RAM location.
2. The division operation is carried out by implementing the long division algorithm in hardware using a loop-driven approach and a minimum number of intermediate register storages to accumulate intermediate results during each loop. The loop control must be implemented using registers designed for that purpose.
3. Assume that the final answer, when it can be achieved without any exceptions, is expressed as a 16-bit quotient in a register R_n and a 16-bit quotient in a register R_{n+1}, where R_n and R_{n+1} are coupled to form an effective 32-bit register to hold the original dividend; to load the registers,

assume that the low-order portion of the dividend is placed in R_n while the high order portion is placed in R_{n+1}. Any other register R_m (m not equal to n) may be used to hold the 16-bit divisor.

For the final submission of the circuit design, each student is required to:

A. Submit a complete RTL specification for the circuit, including control over the timing logic.
B. Submit a complete block diagram of the circuit.
C. For each block diagram element used in part B, a complete specification of the internal logic must be provided (for which students may assume such primitive logic elements as comparators and single-bit adder/subtractor circuits have been defined).

Final comments:

Students are advised to practice performing long-division to fully acquaint themselves with the operations and the algorithm, and those cases that represent exceptions as described in the Project description above. Consider the requirements carefully and ask clarifying questions if you are in doubt.

Some cases to consider include:

i. 0000000000000000) 00000000000000000000000000000001 (1/0)
ii. 0000000000000001) 00000000000000000000000000000000 (0/1)
iii. 0000000000000010) 00000000000000010000000000000000 (2^{16}/2)
iv. 0000000000000001) 00000000000000010000000000000000 (2^{16}/1)

There are many other examples, of course, but these illustrate the following issues. Example (i) involves division by zero which is not permitted – hence the U flag is set to 1. Example (ii) involves a dividend that is zero, hence the final answer for both quotient and remainder is zero – hence the Z flag is set to 1. Example (iii) involves dividing a dividend greater than 16-bits by a divisor such that the final quotient and remainder can both be expressed in 16 bit formats. Finally, example (iv) involves a dividend greater than 16 bits long where the quotient must also be expressed in more than 16-bits, thereby not leaving sufficient room to store the remainder – hence the O flag is set to 1.

Evaluation:

a. All Project work must be completed and submitted for grading by the due date provided.
b. Students are evaluated on all stated requirements.
c. It is mandatory that students complete their own work and must be able to justify their answers when asked to do so by teaching staff.