You are a starting engineer in a new firm called DK Technology Inc. This firm is in the business of designing integrated circuits (IC) modules for various consumer products such as, low power hand held devices, PDA, cell phones, etc. Your first task is to design a 4-bit arithmetic logic unit (ALU) core that will be part of the IC to control a new color touch pad display. In order to make the hand held device compete head-on with larger competitors, the management is under heavy pressure to add as much functionality as possible to the display control chip. Consequently, the primary emphasis at this stage will be on a working ALU design with minimum transistor count and core layout area. If successful, your core ALU layout design will be incorporated into other designs of the firm, as well. So it is also important to document your design in a professional manner.

4-bit ALU Specifications:

Using Tanner tools, design, simulate, layout and verify the operation of an ALU integrated circuit shown in the following block diagram:

This circuit accepts two 4-bit operand inputs, A[3:0] and B[3:0], and generates a 4-bit output R[3:0] (8-bit – R[7:0] in the case of multiplication and parity checks – see Table below). The relationship between operand input vectors (A[4:0], B[4:0]) and the output vector (R[7:0], C, V)*, in which C and V are 1-bit status signals, is defined by the 4-bit control input S[3:0]. The table below defines the behavior of the ALU based on the positive-logic assumption.

+ C: arithmetic carry or borrow bit, V: arithmetic overflow bit.

<table>
<thead>
<tr>
<th>S3</th>
<th>S2</th>
<th>S1</th>
<th>S0</th>
<th>R</th>
<th>Comments</th>
<th>C</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>A</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>B</td>
<td>Add A and B</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>A + B</td>
<td>A plus B</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>A - B</td>
<td>Subtract B from A**</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

0 1 0 0  Right A   Right circular shift of A by one bit 0 0
0 1 0 1  Left B    Left circular shift of B by one bit 0 0
0 1 1 0  0 minus A Negate A** 0 0
0 1 1 1  0 minus B Negate B** 0 0
1 0 0 0  A plus 1 Increment A 0/1 0/
1 0 0 1  A exclusive B A ⊕ B 0 0
1 0 1 0  A minus 1 Decrement A 0/1 0/
1 0 1 1  Abs(A)   |A| 0/1 0/
1 1 0 0  AxB      A multiply B *** 0 0
1 1 0 1  (i)      Even Parity Check of A*** 0 0
1 1 1 0  (ii)     Odd Parity Check of A*** 0 0
1 1 1 1  Min A, B Select minimum 0 0

* 0/1 means the value depends on the result.
** Two’s complement
*** Use R7-R4 in addition to R3-R0
   (i) If number of 1’s in A is even: R7-R0 = 1A_H else R7-R0 = 0A_H
   (ii) If number of 1’s in A is even : R7-R0 = 0A_H else R7-R0=1A_H

Other Design Rules:

1. The design must be able to operate at a 5 MHz clock and 5V-power source. At this stage you are not responsible for the design of the clock.

2. The design documentation is an essential part of the project. No design is acceptable unless accompanied by a detailed professional engineering documentation.

3. Each engineer will make a brief 15-minute professional presentation to highlight the important features of their design.

4. The code S3-S0 will be assigned to each engineer individually. (The code given in the above table is just an example.)

5. All designs will employ the SCMOS logic gate library that is in TannerLB.

6. All layouts will employ the Hewlett Packard 0.5µm n-well technology.

Design Notes:
There are 16 operations corresponding to 16 combinations of the 4-bit control input S[3:0]. Using the hexadecimal representation for the control variable S, we can describe the behavior of the arithmetic logic unit in words. For S=0, the output R is set equal to A, and the carry and overflow signals will be set

to 0. For S=1, the output R is set equal to B; again C and V are set to 0. Arithmetic addition and subtraction operations are defined by S=2 and S=3, respectively. The output R represents the sum or difference of the two 4-bit input signals, and C and V are set to 0 or 1, depending on the outcome of the operation.

The next four operations, corresponding to S=4 to S=7, produce an output that is either the shift or negative (two’s complement representation) of either 4-bit input signal. For these four operations the arithmetic overflow and carry signals C and V are set to 0. With S=8, the output R is A incremented by one. While S=9, bit-wise exclusive-OR of A and B is output to R. With S=10, the output R is generated by decrementing the operand input A. For S=11, the output vector R will be equal to absolute value of A. Since these represent arithmetic operations, the status signals C and V are set based on the result. Operation defined by S=12 is multiplication operation, and the output R is an 8 bit signal. Status signals C and V are set to 0. For S=13, 14, the lower half-byte of R will be set to A, while the upper half-byte of R may take the value of either $0_H$ or $1_H$ depending on the number of 1’s in A. For S=15, the smallest of the two 4-bit inputs A and B is connected to the output R with C and V set to 0.

Good Luck to All!