Name: ______________________________

CS 4/55111 **Midterm Exam** **VLSI Design**

Friday 17 October 2003

1. **The cost for an individual FPGA containing 30,000 gates may be an order of magnitude more than the part cost of an ASIC of the same size, yet FPGA-based designs are usually considered more cost-effective for designs with a volume of tens or hundreds of thousands of units. Explain. (25 points)**

   The "part cost" being referred to here is a variable cost, which scales as the volume of parts increases. However, the total cost must also include the fixed costs, which do not scale and are the same whether one part is produced or a million parts are produced. These fixed costs are much higher for ASICs than FPGAs, including higher cost of training and higher cost of hardware and CAD software, as well as including mask costs and test costs that are not applicable for FPGAs (no masks are needed, and the chips were tested previously at the factory). As a result, these higher ASIC fixed costs overwhelm the smaller ASIC variable costs at low volume, making ASICs cost-effective only at high volumes where the large fixed cost can be distributed over a large number of parts.

2. **The Altera UP1 Education Board is a good way to test simple designs, but you have to understand how the various components work and how they are integrated onto the board.**

   a. **What does it mean to say the pushbuttons are "active low"? (10 points)**

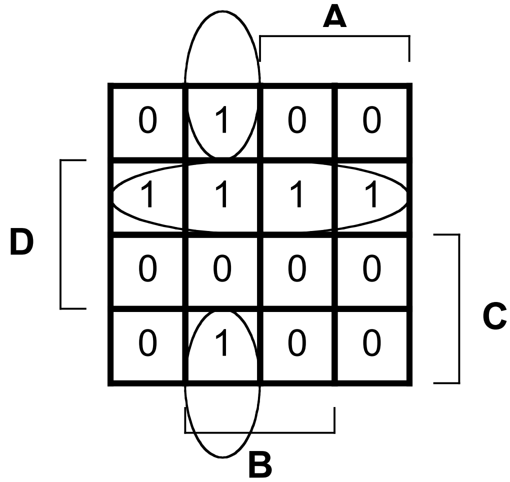
      "Active low" means that when the signal is "active" _ true _ it has the value of '0', so in this case, when the button is pressed, it produces a '0' value; otherwise it produces '1'.

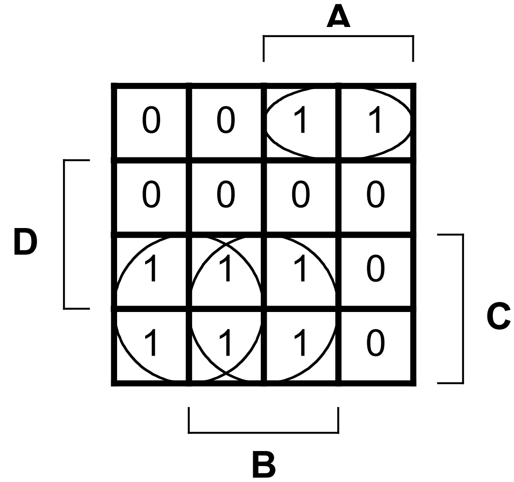
   b. **Explain the procedure for assigning pins when using one of the pushbuttons with the FLEX chip and why this procedure is necessary. (10 points)**

      On the UP1 education boards used in class, two pushbuttons are hardwired to specific input pins on the FLEX chip. This means that if those pins are used, the compiler needs to be told that the signals from those pushbuttons will enter the chip on a pre-assigned input pin, instead of giving the compiler the freedom to assign the input pin. This is done in MaxPlusII by clicking on an input signal and then on Assign _ Pin/Location/Chip to assigning the appropriate input pin to that signal.

Name: ________________________________

3. Given the two 4-variable Karnaugh maps below, circle the 1's and write the minimized expression below each map. (20 points)





Ouput = A'BD' + C'D

Output = AC'D' + A'C + BC

4. Consider the following AHDL code:

```
SUBDESIGN bool2
{
   a0, b0, a1, b1:  INPUT;
   s1:  OUTPUT;
}
VARIABLE
   inter:  NODE;
BEGIN
   inter = a0 & a1 & !b1;
   s1 = inter # b0;
END;
```

a. Draw a schematic diagram that corresponds to this code. (10 points)

See *Digital System Design and Prototyping, Second Edition*, page 151.

b. What would the effect be of interchanging the two lines inside the "BEGIN" / "END" block? Explain. (10 points)

There would be effect at all. AHDL is a concurrent language, so both lines are executed concurrently, regardless of the order in which they are written.

Name: ______________________________

5. Given the following AHDL code, explain how this code "debounces" a pushbutton. What happens when the key is not pressed? What happens when the key is pressed? Refer to parts of the code and be specific in your answer. (15 points)

```
SUBDESIGN debounce
{
   clk, key_pressed:  INPUT;
   strobe:  OUTPUT;
}
VARIABLE
   count[6..0]:  DFF;
BEGIN
   count[].clk = clk;
   count[].clrn = key_pressed;

   IF (count[].q <= 126) & key_pressed THEN
      count[].d = count[].q+1;

   IF count[].q == 126 THEN
      strobe = Vcc;
   ELSE
      strobe = GND;
   END IF;
END;
```

The key_pressed input is connected to the active-low clrn input (the "n" indicates active-low) on the DFF named count. If a key is pressed, key_pressed is high; if no key is pressed, key_pressed is low. Therefore as long as no key is pressed, the clrn input is active, forcing the counter to continuously clear its value.

When a key is pressed, the counter stops continuously clearing its value. Moreover, if a key is pressed and the counter values is less than or equal to 126, the counter increments its count value. If the count value reaches 126, strobe goes high for one clock cycle, but then the counter keeps counting to 127, where it stops counting and strobe goes low again. If the key is released before the count value reaches 126, the counter clears as explained in the paragraph above, and counting resumes from 0 when the key is pressed again.