CS99S
Laboratory 3 Solution

October 22, 2001

Prep question 1: Convert 19_{10} to binary and convert 10101_2 to decimal.

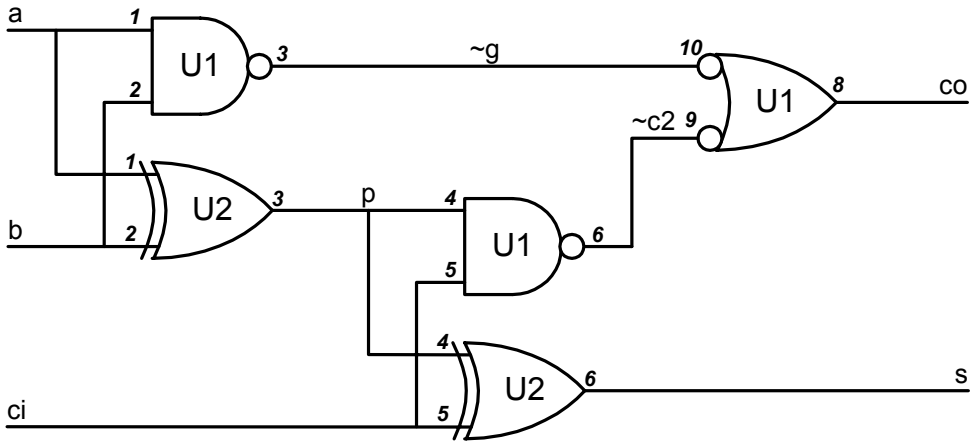
$19_{10} = 10011$ (16+2+1), $10101_2 = 21$ (16+4+1)

Prep question 2: Add 11011_2 to 10010_2. Express the result in both binary and decimal.

```
 11011
 10010
------
101101
```

or 45_{10}

Prep question 3: Draw a gate level schematic for the full adder shown above complete with pin assignments using 74AC86 (4 2-input XOR gates) and 74AC00 (4 2-input NAND gates) chips. (See the schematics linked from the class web page.) Hint: generate the complement[1] of c1 and c2 rather than the true value of these signals so you can use NANDs for both the carry gates in the half adders and for the final OR. You will wire the full adder you build in the lab from this schematic.

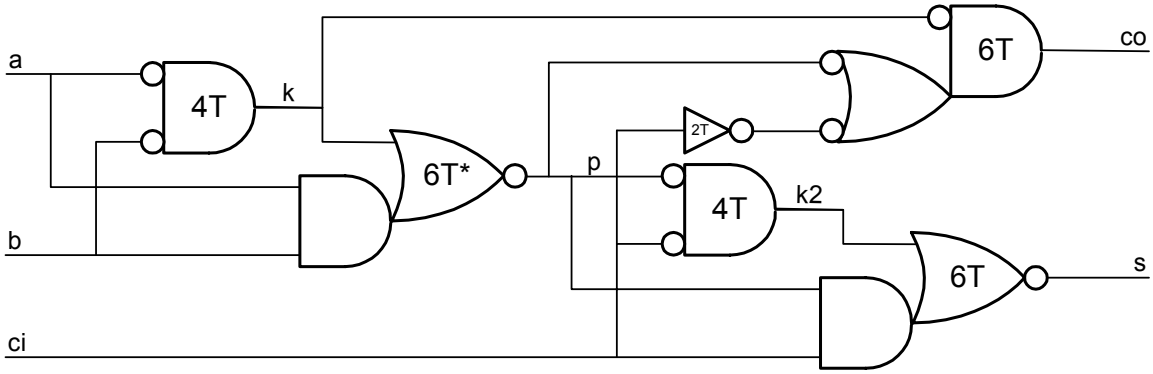


U1 - 74AC00 +14, -7 spare (13,12,11)
U2 - 74AC86 +14, -7 spare (13,12,11),(10,9,8)

[1] The complement of a signal is its opposite signal; i.e. the complement of a 0 is 1, the complement of a 1 is a 0.

Prep question 4: (optional challenge question) What is the smallest number of transistors you can use to realize a full adder?

So far, no one has submitted a solution for this. My best solution is shown below. It consists of two NOR gates (4 transistors each), three AOI gates (AOI stands for AND-OR-INVERT gate. They compute the logic function x = ~((a ∧b)∨c) and require 6 transistors each), and one inverter (2 transistors) for a total of 28 transistors if done using a static CMOS circuit style[2]. If the first AOI gate (indicated with an asterisk) is made a domino gate[3] we can reduce this to 27 transistors. Making all of the NORs and AOIs pseudo-nmos[4] gives a circuit with 20 transistors but one that consumes too much static power to be useful.



The Lab

In the lab you will do the following.

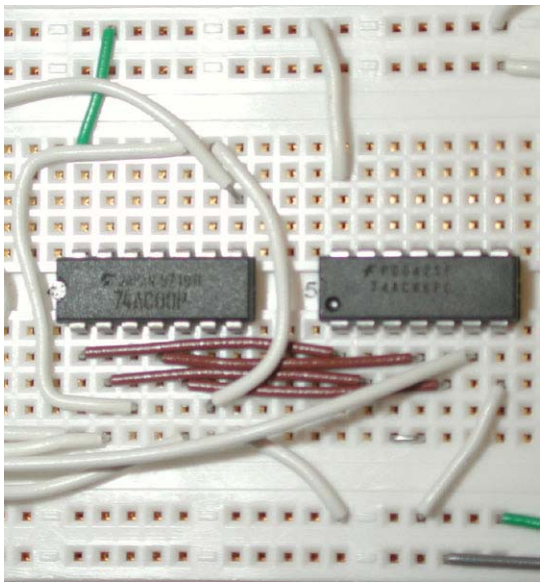
1. Wire up your circuit from prep question 3. First wire up one half adder and verify that it works using your switches and LEDs for input and output. Then wire up the other half adder and verify that the whole full adder works.

   Here's a photo of the solution of prep question 3 above wired up neatly:

[2] A static CMOS circuit is one where a pull-down network of NFETs drives the output to zero and a complementary pull-up network of PFETs drives the output to one. All of the gates we've looked at so far are static CMOS gates.

[3] A domino gate replaces the PFET pull-up network of a static CMOS gate with a single PFET gated by the clock so that the circuit is *precharged* high on one phase of the clock and evaluates on the other phase. There are many restrictions on the use of domino gates. In particular, once the evaluate phase starts, their inputs must increase monotonically (they can't fall after they're high).

[4] A pseudo-nmos gate replaces the PFET pull-up network of a static CMOS gate with a single PFET that is always on and hence acts as a resistor. These gates are just like the ones discussed in Feynman where an NFET network pulls down and a resistor pulls up.



2. Once everyone has completed step 1 we will wire all of the lab kits together to make a multi-bit adder. To combine two lab kits we need to first connect a ground wire between them. Then we tie the carry out of one kit to the carry in of the next kit. Once we have constructed a multi-bit adder in this fashion we will have it work a number of addition problems to verify its operation.

   This photo shows 5 of our lab kits wired together to make a 5-bit adder