Rules

- You may work together, but you are expected to turn in your own writeup of answers.
- HW is due on sakai by the beginning of class.
- Better to hand an answer in late than not at all.
- Email questions to TAs/professor or ask in class.
1. Gate Counts

To compare two 20-bit patterns to see if they are identical, we can construct an “equal20” gate.

(A) Using the “equal5” gate as a guide, figure out how many “and” gates, “or” gates, and “not” gates would be needed to make an “equal20” gate.

Gates Counts (cont’d)

A “k-less-than” gate takes two k-bit numbers and outputs a bit that is True if and only if the first number is less than the second number. It can be made out of

• k “1-less-than” gates: each is made of one “and” and one “not” gate.

• k “1-equal” gates: each is made of two “and”, one “or”, and two “not” gates.

• k “k-and” gates: each is made up of k-1 “and” gates.

• 1 “k-or” gate: made up of k-1 “or” gates.

(B) How many “or” gates do you need altogether to make one “15-less-than” gate?
2. Bit Patterns

Convert these 8-bit numbers to decimal.

(A) 00000011     (E) 00110101
(B) 10001101     (F) 00111010
(C) 00111011     (G) 00001001
(D) 00011010     (H) what’s next?

3. Color By Numbers

• First, convert the following three decimal numbers into 8-bit binary numbers.
  
  (A) 198, (B) 19, (C) 134

• Second, write the binary numbers on the next page, one number per line, one bit per box.

• Finally, color the “1” boxes black and leave the others white. (D) What is the picture?
4. Add ‘Em Up

(A)  
\[
\begin{array}{c}
10101110 \\
+ 00111011
\end{array}
\]

(B)  
\[
\begin{array}{c}
11110001 \\
+ 00110101
\end{array}
\]

(C)  
\[
\begin{array}{c}
11110001 \\
+ 00110101
\end{array}
\]

(D)  
\[
\begin{array}{c}
10100011 \\
+ 11000010
\end{array}
\]

- Add them in binary... don’t just convert to decimal and back!
5. Subtract

(A) 11101001
    - 00011101
    = 10010101

(B) 10010101
    - 01010101
    = 01001010

(C) 10110111
    - 01001000
    = 01110111

(D) 01110001
    - 01000101
    = 01101110

• Subtract in binary... don’t just convert to decimal and back!

6. And Three’s a Crowd

• (A) Give the 8-bit binary number that is the two’s complement of 11100010.

• (B) Now, take the two’s complement of the result from part (A).

• (C) What do you notice about your answer to part (B)?

• (D) Do you think it’s a coincidence? (Hint: Do the same thing with another number.) Explain.
7. Noel

- **A**: not A and not B
- **B**: A or not B
- **C**: False

(a) The “period” of a pattern is the number of steps it takes before the pattern repeats. What is the period of this pattern?

(b) Describe the pattern the state variables fall into. (Which variables will be true and false, step by step?)

8. Guess Who?


- Play around until you can guess the character in four guesses.

- What were the four guesses and their answers? (You can cut and paste from the display.)
9. Guess Who, Too

- Play around until the computer takes more than 4 guesses to win.
- What were the guesses and your answers? (You can cut and paste from the display.)

Coming Soon...

- Download scratch from scratch.mit.edu. Next HW, we’ll have a programming assignment.