Exam 2 review

16 April 2024

Some topics we've covered in the second half of the course:

- Data compression
  - General principles
  - Run-length encoding
  - Sliding window encoding
  - Compression ratio
  - Lossy vs. lossless
- Digital logic
  - AND/OR/NOT gates
  - NOR/NAND/XOR gates
  - Truth tables
- Circuit design
  - Circuit diagrams
  - Gate logic expressions
  - Sum-of-products form
  - Functional equivalence
  - Simplifying circuits
  - Boolean algebra
- Evaluating a system
  - Comparing measurements
  - Prefixes (kilo, mega, etc)
  - Dimensional analysis (cancelling units)
- Verifying data
  - Error detection
  - Parity bits, check digits
  - Error correction

Some connections to think about:

- Representing a particular piece of information on a computer may require using more than one of the techniques we've seen in this course. What are some combinations of techniques that would be used together to represent real-world information?
- For compressing text written in English, would a run-length encoding or a sliding-window encoding be better? Why?
- Why are the boolean algebra properties useful in designing and simplifying circuits?
- Why does the existence of the XOR gate make parity bits easy to compute?
- When you're performing a unit-cancelling dimensional analysis to solve a problem, how would you take account of a compression ratio if you find out the data in question is compressed?
- In the ASCII table, each pair of upper- and lowercase letters are separated by 32 positions, so `'A'` is 65 and `'a'` is 97, and `'B'` is 66 and `'b'` is 98, and so on. Looking at the problem from a circuit design perspective, why is this property convenient? (Hint: what are the binary representations of these numbers?)

Some skills to think about and practice:

- Circuit representations
  - Convert a gate logic expression into its truth table, and vice versa.
  - Convert a truth table into a circuit diagram, and vice versa.
  - Convert a circuit diagram into a gate logic expression, and vice versa.
  - Convert the more verbose gate logic expressions into the Boolean algebra notation, and vice versa.
- Decide whether two different circuits are functionally equivalent (i.e. whether they compute the same function).
  - using truth tables
  - using boolean algebra properties
- Use the units given in a problem to decide which numbers to combine and how, including inserting known conversion factors like somethings-per-megasomething, or seconds-per-minute.
- Compute a parity bit or check digit for a particular piece of data.
- Verify whether a particular piece of data that includes a parity bit or check digit is valid.

A reminder: Other than simple four-function calculators, you MAY NOT use electronic devices on the exam. That includes graphing calculators, cell phones (even just to use the calculator app), laptops, tablets, etc.

Another reminder: in general, you don't have to worry about memorising a lot of things; below is a reference sheet that will also be attached to the exam. You should still study them and understand how to apply them in practice, of course.

Laws of boolean algebra

Commutative	$xy = yx$ $x + y = y + x$
Associative	$x(yz) = (xy)z$ $x + (y + z) = (x + y) + z$
Identity	$1x = x = x1$ $0 + x = x = x + 0$
Double negation	$\overline{\overline{x}} = x$
Null	$0x = 0 = x0$ $1 + x = 1 = x + 1$
Distributive Law	$x(y + z) = xy + xz$ $x + yz = (x + y)(x + z)$
DeMorgan's Law	$\overline{(x + y)} = \overline{x}\,\overline{y}$ $\overline{x} + \overline{y} = \overline{(xy)}$
Inverse	$x\,\overline{x} = 0 = \overline{x}\,x$ $x + \overline{x} = 1 = \overline{x} + x$
Idempotent	$xx = x$ $x + x = x$
Absorption Law	$x(x + y) = x$ $x + xy = x$

Standard digital logic gates