EEM16/CSM51A (Fall 2017)

Logic Design of Digital Systems

Prof. Ankur Mehta : mehtank@ucla.edu

Problem set 1
assigned Monday Oct. 9, 2017
due 4pm Monday Oct. 16, 2017
Show all work.

Instructions

This homework is to be done individually. You may consult with others to share thoughts and ideas, but all of your submitted work must be yours alone. Be sure to indicate with whom you've collaborated and in what manner.

You may use any tools or refer to published papers, books, or course notes. Youre allowed to make use of online tools such as Logisim, WolframAlpha, etc., provided you properly cite them in the space below.

You must submit all pages in this file based on the procedure below. Because of the grading methodology, you may find it easier to print the document and write out your solutions in the space provided in this problem set. You may alternately opt to digitally enter your solutions into the form entries then download or print the filled PDF.

Answers written on sheets other that the provided space will not be looked at or graded. Please write clearly and neatly - if we cannot easily decipher what you have written, you will get zero credit.

Submission procedure

You need to submit your solution online at Gradescope:
https://gradescope.com/

Please see the following guide from Gradescope for submitting homework. You will need to upload a PDF and mark where each question is answered.
http://gradescope-static-assets.s3-us-west-2.amazonaws.com/help/submitting_hw_guide.pdf

Collaborators

Identify with whom you've collaborated and in what manner, if any.

Online Resources

Identify which online tools you've used, if any.

1 The genetic code (Number systems)

The blueprints for life are stored in DNA molecules, which are made up of strings of monomer building blocks. There are 4 distinct monomers, known as nucleotides: adenine (A), cytosine (C), guanine (G), and thymine (T). DNA is used to encode peptides and proteins, which are made up of strings of amino acid monomers. There are 20 distinct amino acids coded for by DNA. We can consider both DNA and protein as discrete (why?) signals.

1.1 What if engineers wrote bio textbooks?

What is the minimum number of bits necessary to uniquely label ...

1.1(a). ... a single DNA nucleotide?

1.1(b). ... a single amino acid?

1.2 A number of codons

We can consider DNA nucleotides as symbols in a base-4 number system.

1.2(a). How many nuclueotides do you need to uniquely label a single amino acid?

1.2(b). How many labels of this length (known as codons) can you assign to each amino acid equally?

1.3 What size are your genes?

Oxytocin is a neuropeptide responsible for a number of psychological effects, including feelings of love, trust, fear, anxiety, and bonding. It is made of 9 amino acids.

1.3(a). Using a labeling system as described in 1.2, how many nucleotides would be necessary to label Oxytocin?

1.3(b). If we translated that base-4 representation to a numerical value, how many decimal digits would be needed to write the numerical value for Oxytocin?

1.3(c). If instead we used the amino acids directly as a base-20 numbering system, how many decimal digits would be necessary to write out the numerical value for Oxytocin?

1.1.a
2 bits
because 4 possible nucleotides, so 2^2 = 4

1.1.b
5 bits
because 20 possible amino acids, so 2^5 = 32 > 20

1.2.a
3 nucleotides
because 20 possible amino acids, and nucleotides are base 4, so 4^3 = 64 > 20

1.2.b
3 labels
because 20 distinct amino acids, and 64 possibilities created by 3 nucleotides, and 64/3 >= 20

1.3.a
27 nucleotides
because 9 amino acids in oxytocin, 3 nucleotides per amino acid

Use this page for more work on Problem 1.

1.3.b
17 digits
because nucleotide is base 4, and 27 nucleotides in oxytocin, 4^27 ~= 1.8e+16, so 17 digits

1.3.c
9 digits
because 9 amino acids in oxytocin, and each amino acid corresponds to one digit

2 October events (Boolean algebra)

Consider the month of October, 2017. It has 31 days (1-31), each with 24 hours (0-23 `a la military time). We can represent each by 5 bit numbers $d = d_4d_3d_2d_1d_0$, $h = h_4h_3h_2h_1h_0$ respectively; we will create systems taking these 10 bits as inputs.

2.1 When is this pset due?

2.1(a). What is the 10 bit input $d_4d_3d_2d_1d_0h_4h_3h_2h_1h_0$ corresponding to the date and time when this pset is due?

2.1(b). Write the boolean expression of the 10 input bits that is true only when the input represents the day and time when this pset is due, and is false otherwise.

2.1(c). Is this a minterm or a maxterm of the system? Why?

2.2 Is it the weekend yet?

2.2(a). Taking just the 5 bit date input, write the function that is true if and only if the corresponding date is a weekend day as a boolean expression in some canonical normal form.

2.2(b). Which normal form did you use, and why?

2.3 Does this even make sense?

2.3(a). Not all sets of 10 input bits correspond to a valid date and/or time. Write the function—using minterm or maxterm shorthand—that returns true if the 10 bit input is valid, and false otherwise.

2.3(b). Find the simplest boolean expression for this function. *Hint: this will not be a canonical normal form, and will contain only 7 literals along with a number of operators.*

2.3(c). Write the dual of this expression.

2.4 Don't care.

If we specify a set of valid inputs for our system, we can choose what value our function assigns to invalid inputs ("don't care" values) whichever way we would like in order to simplify our boolean expressions.

2.4(a). If we assume that our 5 bit input will represent a valid hour, come up with a simple boolean expression (using no more than 3 literals) that returns true if the hour is a valid afternoon hour and false if it is a valid morning hour.

2.4(b). If instead we specify that the function must also be false for invalid hours, does the boolean expression change? If so, what is the new expression?

2.4(c). Draw this function using only NOR gates

2.1.a
1000010000
due 4pm Oct 16 2017, which is day 16 hour 16, and 16 is 10000 in binary

2.1.b
d4 ^ -d3 ^ -d2 ^ -d1 ^ -d0 ^ h4 ^ -h3 ^ -h2 ^ -h1 ^ -h0
because 1000010000 is the only input that should lead to output True

2.1.c
minterm
because it only consists of AND's. Also, the question asked for equal True

Use this page for more work on Problem 2.

2.2.a
(-d4 ^ -d3 ^ -d2 ^ -d1 ^ d0) v (-d4 ^ -d3 ^ d^2 ^ d1 ^ d0) v (-d4 ^ d3 ^ -d2 ^ -d1 ^ -d0) v (-d4 ^ d3 ^ d2 ^ d1 ^ -d0) v (-d4 ^ d3 ^ d2 ^ d1 ^ d0) v (d4 ^ -d3 ^ d2 ^ -d1 ^ d0) v (d4 ^ -d3 ^ d2 ^ d1 ^ -d0) v (d4 ^ d3 ^ d2 ^ -d1 ^ -d0)
The weekend dates for October 2017 are 1, 7, 8, 14, 15, 21, 22, 28.
These numbers are represented in binary as 00001, 00111, 01000, 01110, 01111, 10101, 10110, 11100.
The functions which result in True IFF these numbers are the following, respectively: -d4 ^ -d3 ^ -d2 ^ -d1 ^ d0,
-d4 ^ -d3 ^ d^2 ^ d1 ^ d0,
-d4 ^ d3 ^ -d2 ^ -d1 ^ -d0,
-d4 ^ d3 ^ d2 ^ d1 ^ -d0,
-d4 ^ d3 ^ d2 ^ d1 ^ d0,
d4 ^ -d3 ^ d2 ^ -d1 ^ d0,
d4 ^ -d3 ^ d2 ^ d1 ^ -d0,
d4 ^ d3 ^ d2 ^ -d1 ^ -d0.
Now, taking the sum of products using these, we get the equation (-d4 ^ -d3 ^ -d2 ^ -d1 ^ d0) v (-d4 ^ -d3 ^ d^2 ^ d1 ^ d0) v (-d4 ^ d3 ^ -d2 ^ -d1 ^ -d0) v (-d4 ^ d3 ^ d2 ^ d1 ^ -d0) v (-d4 ^ d3 ^ d2 ^ d1 ^ d0) v (d4 ^ -d3 ^ d2 ^ -d1 ^ d0) v (d4 ^ -d3 ^ d2 ^ d1 ^ -d0) v (d4 ^ d3 ^ d2 ^ -d1 ^ -d0).

2.2.b
Sum of products (disjunctive)
because I wanted an equation which results in True IFF it is the weekend and there are fewer weekends than weekdays, so fewer Trues than Falses, so I took the sum of the minterms.

2.3.a
m(32-55, 64-87, ..., 992-1015)
because m(32 thru prev_num+23, prev_num+9 thru prev_num+23, ..., 992-1015) where prev_num changes to the last calculated number
in otherwords, I am taking the OR of all valid date/hours

2.3.b
(d4 v d3 v d2 v d1 v d0) ^ (-h4 v -h3)
[00001-11111][00000-10111]
is the acceptable range, so the unacceptable input is when day = [00000] and hour has something in h4 and d3

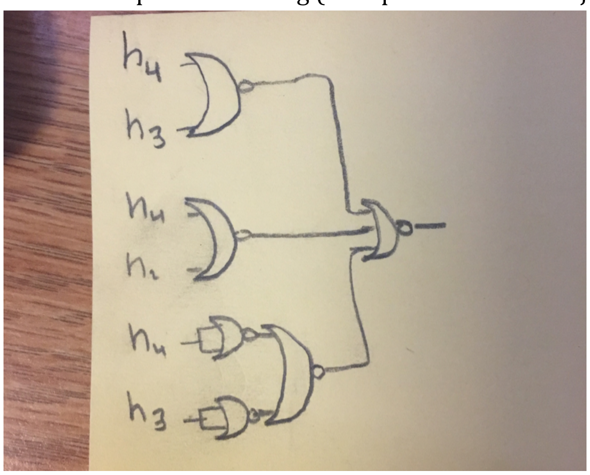
2.3.c
(d4 ^ d3 ^ d2 ^ d1 ^ d0) v (-h4 ^ -h3)
Change AND to OR and vv.

Use this page for more work on Problem 2.

2.4.a
h4 v (h3 ^ h2)
Valid afternoon hours are [12:23] and valid morning hours are [0:11]
Assuming input is valid,
so we need f([12:23]) = True.
we can say f([16:23]) = True, which can be represented by h4.
Now, for f([12:15]) = True, can be represented by h3 ^ h2.
So together, we have h4 v (h3 ^ h2).

2.4.b
Invalid hours before the morning hours are already false because they are in the morning, so we may disregard that.
Invalid hours after the afternoon hours are [24:31].
11000
So, we can say [24:31] = h4 ^ h3
Since we want the equation to be false during these hours, we can say -(h4 ^ h3).
So, the total equation is (h4 v (h3 ^ h2)) ^ -(h4 ^ h3)

2.4.c
this is a simplified drawing (I simplified the two adjacent INV's for each path going into the last NOR)



(h4 v (h3 ^ h2)) ^ -(h4 ^ h3) -- Original
(h4 v h3) ^ (h4 v h2) ^ -(h4 ^ h3) -- Distribute

3 Your turn

It's often said that you don't truly understand a subject until you can teach it. What was a topic that you struggled with so far in this class? Write and solve a pset problem that sheds light on this particular topic.

3.Q
xy|out
00|0
01|1
10|0
11|1
For the above function, write it in minterms, convert it to AND's and OR's, convert it to NOT's and AND's, then simplify it

3.A.a
minterm(1)+minterm(3)

3.A.b
(-x^y)v(x^y)

3.A.c
-(-(-x^y)^-(x^y))

3.A.d
-(-(-x^y)^-(x^y)) -- Original
-((x^-y)^-(x^y)) -- De Morgan
-(x^-y)v(x^y) -- De Morgan
(-xvy)v(x^y) -- De Morgan
y -- Combining

Use this page for more any more work that you may need to show. Clearly identify which problem it is for.