Kyle Romero

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.

Jair Hinojosa ; discussing ideas
Steven Lara ; discussing ideas

Online resources

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

Wolfram Alpha; calculations involving long numbers, checking boolean expression truth tables
Calendar App
Calculator App

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?

$\boxed{1.1}$ a) With $\boxed{\text{2 bits}}$ we can represent all 4 nucleotides, since $2^2 = 4$

b) We would need $\boxed{\text{5 bits}}$ to uniquely label 20 amino acids. We know this is true because:

* With 5 bits we can create $2^5 = 32$ unique combinations
* With 4 bits we can create $2^4 = 16$ unique combinations, which is not enough

$\boxed{1.2}$ a) To uniquely label a single amino acid you would need at least $\boxed{\text{3 nucleotides}}$. We know this is true because:

* With 3 nucleotides, we can create $4^3 = 64$ unique combinations, which is enough to represent all 20 distinct amino acids
* With 2 nucleotides, we could only create $4^2 = 16$ unique combinations, which is not enough to represent all 20 distinct amino acids

b) Given that we have 64 unique combinations of 3 nucleotides and only 20 amino acids, we can use $\lfloor \frac{64}{20} \rfloor = 3$ to find that the # of labels of this length that we can assign to each amino acid equally is $\boxed{3}$.

Use this page for more work on Problem 1.

$\boxed{1.3}$ a) Given that we need 3 nucleotides to represent each amino acid, we would need $3 \cdot 9 = \boxed{27 \text{ nucleotides}}$ to label Oxytocin.

b) In our current base-4 numbering system we would need 27 digits to represent Oxytocin. To translate this to decimal, we simply do the calculation $4^{27} = 18014398509481984$. So, 27 base-4 digits can represent that many numbers in decimal. Therefore, to represent Oxytocin using decimal symbols we would need $\boxed{17 \text{ decimal digits}}$, or the decimal length of 4^{27}.

c) In a base-20 numbering system, we can represent Oxytocin with 9 digits, since each amino acid gets its own digit. With 9 digits and a base-20 system, we have $20^9 = 512000000000$ possible codings for Oxytocin. Therefore, to represent Oxytocin using decimal symbols we would need $\boxed{12 \text{ decimal digits}}$, or the decimal length of 20^9.

2 October events (Boolean algebra)

Consider the month of October, 2017. It has 31 days (1-31), each with 24 hours (0-23 à 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.

$\boxed{2.1}$ a) This problem is due on October 16^{th} @ 4PM (16 military time). Therefore, we want to represent 16-16. Since the 4^{th} digit in binary is the 16 place, the 10-bit input corresponding to the date and time this pset is due is: $\boxed{1000010000}$

b) $\boxed{d_4 \bar{d}_3 \bar{d}_2 \bar{d}_1 \bar{d}_0 h_4 \bar{h}_3 \bar{h}_2 \bar{h}_1 \bar{h}_0}$ is true only when the input represents the day and time when this pset is due and false otherwise

c) This is a $\boxed{\text{minterm}}$ because the output is 1 for only one case, in a maxterm the output is 0 for only one case

Use this page for more work on Problem 2.

$\boxed{2.2}$ a) In general terms, for October this function would look like the following: $f(n) = min(1) + min(7) + min(8) + min(14) + min(15) + min(21) + min(22) + min(28) + min(29)$

Decimal	Binary
1	00001
7	00111
8	01000
14	01110
15	01111
21	10101
22	10110
28	11100
29	11101

Boolean Expression:

$$\boxed{\begin{aligned} f(d_4 d_3 d_2 d_1 d_0) = {} & \bar{d}_4\bar{d}_3\bar{d}_2\bar{d}_1 d_0 + \bar{d}_4\bar{d}_3 d_2 d_1 d_0 + \bar{d}_4 d_3 \bar{d}_2 \bar{d}_1 \bar{d}_0 \\ & + \bar{d}_4 d_3 d_2 d_1 \bar{d}_0 + \bar{d}_4 d_3 d_2 d_1 d_0 + d_4 \bar{d}_3 d_2 \bar{d}_1 d_0 \\ & + d_4 \bar{d}_3 d_2 d_1 \bar{d}_0 + d_4 d_3 d_2 \bar{d}_1 \bar{d}_0 + d_4 d_3 d_2 \bar{d}_1 d_0 \end{aligned}}$$

b) This is $\boxed{\text{disjunctive normal form}}$ because we are using minterms (sum of products). I used minterms because implementing this function w/ maxterms would require significantly more terms and therefore writing.

$\boxed{2.3}$ a) The conditions that make a date/time invalid in our case:

- Date = 0
- Time > 23

Therefore, in minterm/maxterm shorthand and considering the following functions we have:

$$\boxed{f(\text{date}, \text{hour}) = g(\text{date}) + h(\text{hour})}$$

$$g(\text{date}) = \text{Max}(0) \qquad h(\text{hour}) = \text{Max}(24-31)$$

b) Basically, we need to ensure that the 5 date bits are not all 0 and that the 5th and 4th hour bit are not on at the same time. Keeping this in mind I was able to derive the following boolean expression:

$$f(d_4, d_3, d_2, d_1, d_0, h_4, h_3) = \boxed{(d_4 \vee d_3 \vee d_2 \vee d_1 \vee d_0) \wedge \neg(h_4 \wedge h_3)}$$

c) To determine the dual of this expression you simply swap ANDs and ORs and vice versa. Therefore, the dual is:

$$d(d_4, d_3, d_2, d_1, d_0, h_4, h_3) = \boxed{(d_4 \wedge d_3 \wedge d_2 \wedge d_1 \wedge d_0) \vee \neg(h_4 \vee h_3)}$$

$A - h_4$
$B - h_3$
$C - h_2$

Use this page for more work on Problem 2.

$\boxed{2.4}$ a) Valid afternoon hours range from 12-23 inclusive. Since we have assumed that the 5-bits are a valid input (<24) we can implement this function with the following expression:

$$\boxed{f(h_4, h_3, h_2) = (h_3 \wedge h_2) \vee h_4}$$

This expression works for the following reasons:

* $h_3 \wedge h_2$ covers the cases 12, 13, 14, 15 because their bit patterns are 011??
* h_4 covers the cases ≥16 because any of those numbers follows the bit pattern 1????

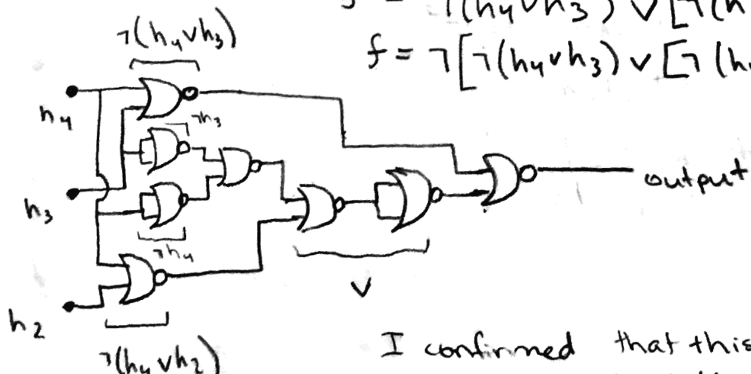
b) Yes, if we have to check for valid input then the expression changes to

$$\boxed{f(h_4, h_3, h_2) = [(h_3 \wedge h_2) \vee h_4] \wedge \neg(h_4 \wedge h_3)}$$

to ensure that the time is not > 23, this was taken from my answer to 2.36.

c) To help us do this problem we should simplify our boolean expression to only use ORs and NOTs:

$$f(h_4, h_3, h_2) = [(h_3 \wedge h_2) \vee h_4] \wedge \neg(h_4 \wedge h_3)$$
$$= (h_4 \vee h_3) \wedge (h_4 \vee h_2) \wedge (\neg h_4 \vee \neg h_3)$$
$$\overline{f} = \neg(h_4 \vee h_3) \vee [\neg(h_4 \vee h_2) \vee \neg(\neg h_4 \vee \neg h_3)]$$
$$f = \neg[\neg(h_4 \vee h_3) \vee [\neg(h_4 \vee h_2) \vee \neg(\neg h_4 \vee \neg h_3)]]$$



Note: We can use a NOR gate to make a NOT gate (inverter)

I confirmed that this worked properly using the following truth table that I generated:

h_4	h_3	h_2	out
1	1	1	0
1	1	0	0
1	0	1	1
1	0	0	1
0	1	1	1

h_4	h_3	h_2	out
0	1	0	0
0	0	1	0
0	0	0	0

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.

Some topics that I've struggled with in class are:

- Contamination delay vs propagation delay
- Minterms / Maxterms
- Simplification of boolean expressions

I will try to write a pset that incorporates all of these topics.

$\boxed{3.1}$ Min/Max

a. Let $x_2x_1x_0$ be a 3-digit binary number. The function $f(x_0, x_1, x_2)$ is defined to be true if there are ≥ 2 zeros in the input and false otherwise. Draw the truth table for this function.

b. Using some canonical normal form, write out a boolean expression for f.

$\boxed{3.2}$ Simplify

a. Using the laws of boolean algebra, simplify your expression from 3.1.b to use fewer operations than the canonical normal form.

b. Using your answer to 3.2.a, draw the circuit representing f using 1 OR gate and 3 AND gates (unlimited inverters).

$\boxed{3.3}$ Delays

a. Assuming a contamination delay of 5s for AND gates, a contamination delay of 7s for OR gates, and 2s for inverters, what is the contamination delay of

b. Assuming the same values for the propagation delays of each gate, what would be the propagation delay?

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

$\boxed{3.1}$ a)

x_0	x_1	x_2	out
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0

x_0	x_1	x_2	out
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

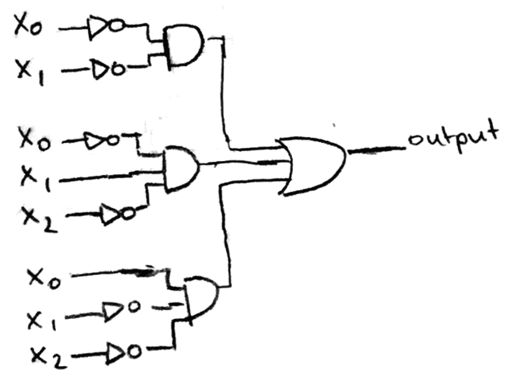
b) $f(x_0, x_1, x_2) = \bar{x}_0\bar{x}_1\bar{x}_2 + \bar{x}_0\bar{x}_1x_2 + \bar{x}_0x_1\bar{x}_2 + x_0\bar{x}_1\bar{x}_2$

$\boxed{3.2}$ a) $f(x_0, x_1, x_2) = \bar{x}_0\bar{x}_1\bar{x}_2 + \bar{x}_0\bar{x}_1x_2 + \bar{x}_0x_1\bar{x}_2 + x_0\bar{x}_1\bar{x}_2$

$= \bar{x}_0\bar{x}_1(\bar{x}_2 + x_2) + \bar{x}_0x_1\bar{x}_2 + x_0\bar{x}_1\bar{x}_2$

$= \bar{x}_0\bar{x}_1 + \bar{x}_0x_1\bar{x}_2 + x_0\bar{x}_1\bar{x}_2$

b)



$\boxed{3.3}$ a) To find the contamination delay we must find the shortest path through the circuit and add up the contamination delays along the path. There are multiple shortest paths consisting of 1 OR gate and 1 AND gate, therfore the CD is $7+5 = \boxed{12 \text{ seconds}}$

b) For the propagation delay we find the critical path through the circuit and add up the propagation delays along that path. Doing this we find multiple longest (critical) paths made up of 1 OR gate, 1 AND gate, and 1 inverter. Therefore, the PD is $7+5+2 = \boxed{14 \text{ seconds}}$