Problem Set #3

Due: October 11 (+2 bonus)
Hard Deadline: October 16

Autonomous last-mile delivery routing

Your startup is implementing an autonomous food delivery fleet at a large technical university campus. Since you have some knowledge of digital logic, you take the initiative to design a routing device that will be installed on each vehicle in the fleet. Each vehicle's routing device aims to determine whether a current order's delivery is possible, or whether the vehicle needs to be re-routed. This is done by selecting the correct path based on the binary address of the destination, using a multiplexer, a decoder, and basic binary number systems.

Inputs

In your prototype roll out, you select **4 drop-off locations** in your campus, each represented by a unique 2-bit binary address, which is represented as a binary address $A = (A_1 A_0)_2$. Your device receives the following information for each order:

- The drop-off location of the customer, represented by the two signals A_1, A_0.
- Path obstruction detection signals given by a radar sensor for each path R_3, R_2, R_1, R_0, which report the safety of the path to delivery at $(A_1 A_0)_{10}$:

$$R_i = \begin{cases} 1 & \text{path to location } i \text{ is not obstructed} \\ 0 & \text{otherwise.} \end{cases} \qquad i = (A_1 A_0)_{10}$$

Outputs

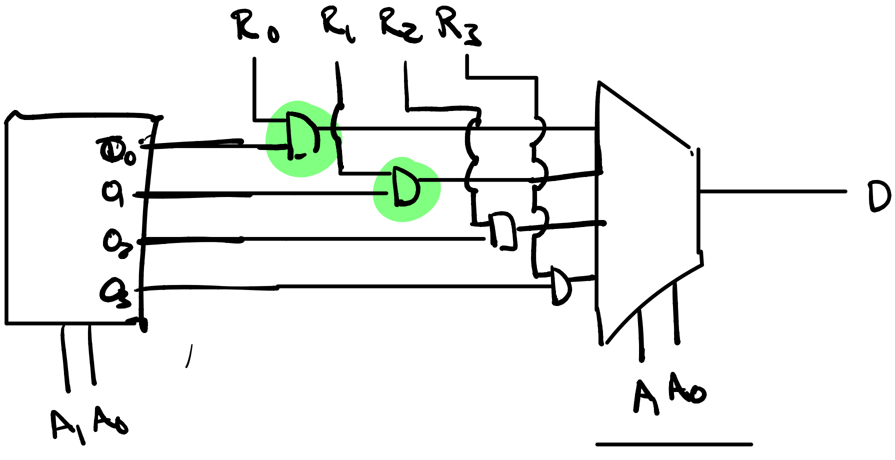
Your task is to design a routing device that outputs a *delivery-able* signal

$$D(A_1, A_0) = \begin{cases} 1 & \text{delivery to address } A_1 A_0 \text{ is possible} \\ 0 & \text{need re-routing} \end{cases}$$

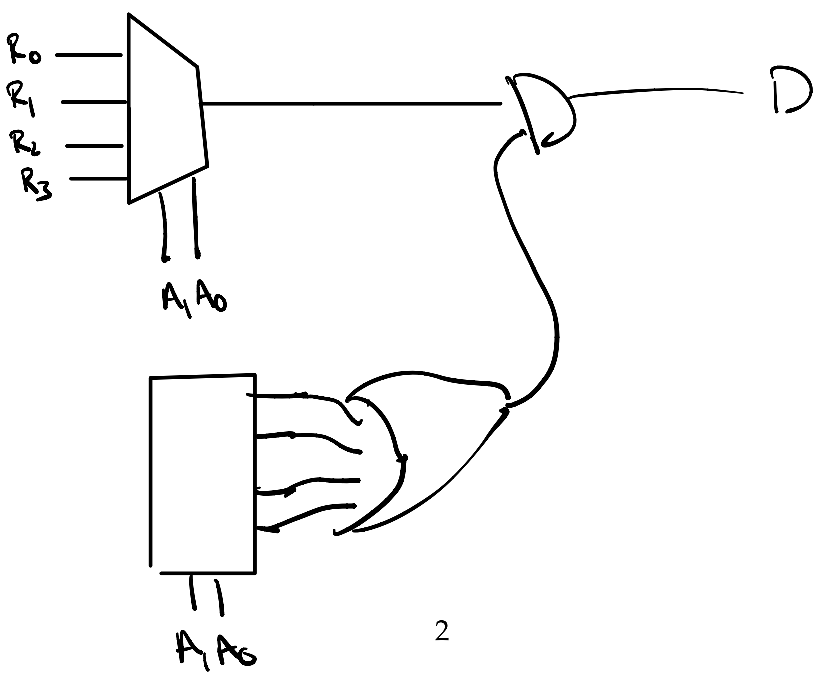
Problem 1 (6 points)

Design a logic circuit using a 2-to-4 decoder, a 4-to-1 multiplexer, and 4 logic gates to return the routing feasibility signal D described above.

Sol 1:



Sol 2:



Multiplexers and Decoders: The Sequel

Consider the following logic circuit using a 3-to-8 decoder and a 4-to-1 multiplexer.

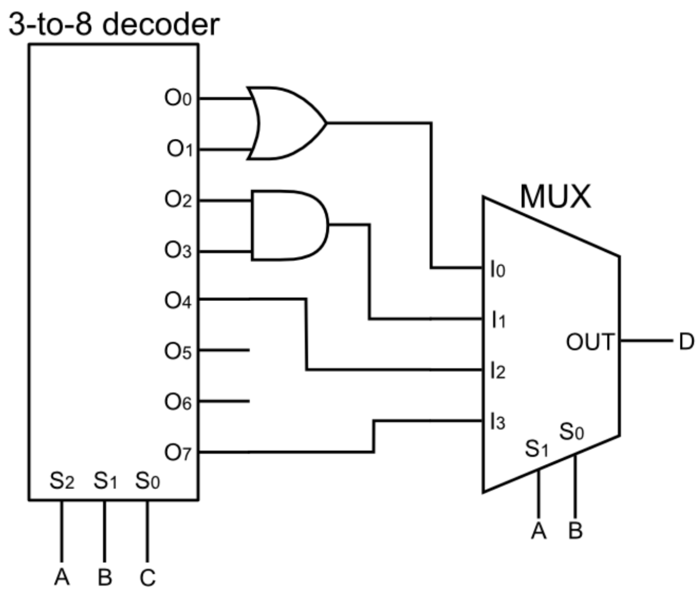


Figure 1: A logic circuit that combines a decoder and multiplexer

The outputs of a 3-to-8 decoder $O_0, \ldots, O_7$ with selection control inputs S_2, S_1, S_0, have the form

$$O_i = \begin{cases} 1 & i = (S_2 S_1 S_0)_{10} \\ 0 & \text{otherwise.} \end{cases} \qquad i = 0, 1, \ldots, 7$$

The output signal D of the 4-to-1 multiplexer (MUX) with inputs I_0, I_1, I_2, I_3 and selection control inputs S_1, S_0 is given as

$$D = I_k \quad \text{where} \quad k = (S_1 S_0)_{10}.$$

Problem 2 (4 points)
Compute the simplest possible sum-of-products expression for the output

$$D = F(A, B, C)$$

shown in Fig. 1. Use a truth table and K-Map to verify your result.

Solution

Solve the problem by designing the truth values of each MUX input.

1. For input 0: if we desire $I_0 = 1$ we must obtain the following:

$$\begin{aligned} I_0 &= O_0 + O_1 \\ &= \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C \\ &= \overline{A} \cdot \overline{B} \end{aligned}$$

2. For input 1, recall that for a decoder, we know that **only one output can be active at any time**; consequently, notice that

$$I_1 = O_2 \cdot O_3 = 0;$$

3. For $I_2 = 1$, notice that

$$I_2 = O_4 = A \cdot \overline{B} \cdot \overline{C}$$

4. Similarly,

$$I_3 = O_7 = A \cdot B \cdot \cdot C.$$

To wrap it all up together, we have

$$\begin{aligned} D &= I_0 + I_1 + I_2 + I_3 \\ &= \overline{A} \cdot \overline{B} + A \cdot \overline{B} \cdot \overline{C} + A \cdot B \cdot C \\ &= \overline{B}(\overline{A} + A \cdot \overline{C}) + A \cdot B \cdot C \\ &\overset{(1)}{=} \overline{B} \cdot \overline{A} + \overline{B} \cdot \overline{C} + A \cdot B \cdot C \end{aligned}$$

where step (1) is by disappearing opposite. Thus,

$$\boxed{D = \overline{B} \cdot \overline{A} + \overline{B} \cdot \overline{C} + A \cdot B \cdot C.}$$

Decoding the number system of an alien civilization

A few centuries in the future, you arise from cryogenic sleep orbiting in the Proxima Centauri system. You lead explorations to the Proxima Centauri b and Proxima Centauri d exoplanets, and shockingly, discover evidence of a previously existing multi-planetary civilization.

After decades of study, you amazingly find an artifact that indicates that the civilization knew how to solve quadratic equations. You are able to decode the following equation in an unknown base-b number system:

$$(5)_b x^2 - (50)_b x + (125)_b = 0 \implies x = (5)_b \text{ or } x = (8)_b$$

The value $x = (5)_b$ seems legitimate enough, but $x = (8)_b$ requires some further explanation, i.e., $b \neq 10$. You and your team reflect on the way that Earth's number system developed, and found evidence that the Proxima Centauri civilization had a similar history.

Problem 3 (4 points)

How many fingers did the extraterrestrials of the Proxima Centauri system most likely have? That is, what is the value of b?

Solution

We claim that $b = 13$.

The proof is as follows. Assume for the sake of contradiction that $b = 10$. Then, applying the quadratic formula, the roots of the above polynomial are $x \in \{(5)_{10}, (-5)_{10}\}$; thus, since $x \in \{(5)_b, (8)_b\}$ by assumption, it follows that $b \neq 10$. Therefore, let us represent the base b numbers in base-10 using our radix b conversion formula:

$$\begin{aligned}(5)_b &= 5b^0 = 5\\ (50)_b &= 5b^1 + 0b^0 = 5b\\ (125)_b &= 1b^2 + 2b + 5b^0 = b^2 + 2b + 5.\end{aligned}$$

Consequently, the polynomial in base-10 becomes

$$5x^2 - (5b)x + (b^2 + 2b + 5) = 0$$

As we know that $x = 5$ is a root of this polynomial regardless of the value of b, we can write

$$0 = 5(5)^2 - (5b)(5) + b^2 + 2b + 5 = 130 - 23b + b^2.$$

So, to wrap up, we want to solve the polynomial equation

$$b^2 - 23b + 130 = 0.$$

Applying the quadratic formula once more yields

$$b = \frac{23 \pm \sqrt{23^2 - 4(130)}}{2} = \frac{23 \pm 3}{2} \implies b \in \{10, 13\}.$$

As we already determined by contradiction that $b \neq 10$, we conclude that

$$\boxed{b = 13.}$$

Adder Chips in Real Life

Review the following data sheet for the 74HC283 adder chip:

Click for the data sheet for the 74HC283 adder chip

The following functional diagram is a high-level depiction of this chip.

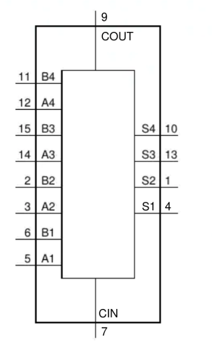


Figure 2: Functional diagram for the 74HC283 adder circuit

Below, the entire logic circuit is depicted.

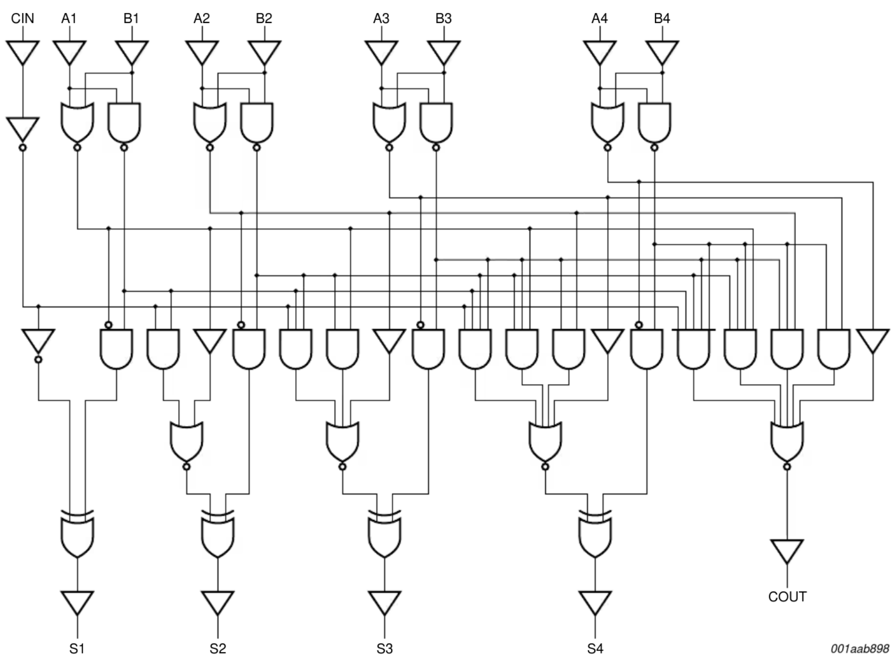


Figure 3: Full logic circuit for the 74HC283 adder circuit

Problem 4 (6 points)

Suppose you implement your 74HC283 chip in a project where the chip takes two 4-bit inputs $A := A_3A_2A_1A_0$ and $B := B_3B_2B_1B_0$, a single 1-bit *"carry-in"* input C_{in}. The circuit puts a 4-bit output *sum* $S := S_3S_2S_1S_0$ and a 1-bit *"carry-out"* output C_{out}. Perform the following analyses of the adder chip.

1. Let the inputs be $A = 0101$, $B = 0111$, and $C_{\text{in}} = 0$. Compute the outputs S and C_{out} using binary arithmetic.
2. Suppose A and B are 4-bit unsigned integers. What is the range of possible base-10 integers that can be obtained by S, without consider C_{out}?
3. What is the range of possible base-10 integers that can be obtained by S, considering C_{out}?

Solution

1. We have

$$\begin{array}{rl} & 0111 \\ & \ \ 0101 \\ + & \ \ 0111 \\ = & \ \ 1100. \end{array}$$

2. If we neglect C_{out}, only S is considered as an output. As S is interpreted as an unsigned integer, and A and B are 4-bit unsigned integers, then S is also a 4-bit unsigned integer and the output range is
$$0 \leq S \leq 2^4 - 1 \implies 0 \leq A + B \leq 15.$$
3. If C_{out} is also considered as an output, then the output is effectively a 5-bit binary unsigned integer; the range of any 5-bit unsigned integer X is $0 \leq X \leq 2^5 - 1 = 31$. However, A and B are **only 4 bit unsigned integers** with values in the range $0 \leq A \leq 15$ and $0 \leq B \leq 15$; hence,
$$0 \leq A + B \leq 30.$$

Discovering properties of complement systems

Problem 5 (4 points)

Suppose that a $4n$-bit number B is represented by an n-digit hexadecimal number H.

1. Prove that the two's complement of B is represented by the 16s' complement of H.
2. Prove that the one's complement of B is represented by the 15s' complement of H

Solution

First, let's recall that a hexadecimal digit corresponds to 4 binary digits. Therefore, a $4n$-bit binary number can be represented as an n-digit hexadecimal number H:

$$B = \sum_{i=0}^{4n-1} B_i \times 2^i = \sum_{i=0}^{n-1} H_i \times 16^i.$$

In a radix b number system, the complement of any n-digit number $X = (D_{n-1}D_{n-2}\dots D_1D_0)$ can be obtained by subtracting it from b^n, yielding

$$-X = b^n - \sum_{i=0}^{n-1} D_i \times b^i$$

.

1. In binary (base)-2, the radix complement (the b's complement) of an n-digit number X is given as $-X = b^n - X$. Therefore, observe that:

$$\begin{aligned} -B &= 2^{4n} - B \\ &= 2^{4n} - \sum_{i=0}^{4n-1} B_i \times 2^i \\ &= 16^n - \underbrace{\sum_{i=0}^{n-1} H_i \times 16^i}_{=H} \\ &= \underbrace{16^n - H.}_{\text{16's complement of } H} \end{aligned}$$

   Therefore, the 2's complement of B is the 16's complement of H.

2. In base b, the one's complement of an n-digit number X is given by simply **subtracting 1 from the two's complement.** Using the identities from above, note that

$$\begin{aligned} -B &= 2^{4n} - 1 - B \\ &= 16^n - 1 - H. \end{aligned}$$

Parenthetical note: This is an old textbook problem, and I'm not a huge fan of it. I don't think the question or the solution is particularly clear. Thus, it is being graded as a bonus question; thanks for participating if you attempted this :)

Cubes and Codes

Let $\mathcal{B}_n$ denote the n-dimensional *binary hypercube*[1]: $\mathcal{B}_n = \{\mathbf{u} : \mathbf{u} \in \{0,1\}^n\}$, which is the set of all n-dimensional binary vectors $\mathbf{u} \in \{0,1\}^n$.

Problem 6 (4 points, 1 bonus point possible)

Explore the following properties of hypercubes:

1. How many vertices, or unique vectors, are contained in $\mathcal{B}_n$? What do these represent? Explain in words. Then, write a formula that gives the number of m-dimensional subcubes of $\mathcal{B}_n$ for any values of m, n.
2. Consider the cases where $n = 3$ and $n = 4$. Draw the 3-dimensional and 4-dimensional hypercubes $\mathcal{B}_3$ and $\mathcal{B}_4$. For at least the $n = 3$ case, label the vertices, which represent all the natural numbers $\{0, 1, 2, \ldots, 7\}$ in binary.
3. **[BONUS] (1pt)** Use your drawing of $\mathcal{B}_4$ to draw a path along the edges that visits each vertex exactly once. Argue that this path orders the natural numbers $\{0, 1, 2, \ldots, 15\}$ represented by 4 bits while ensuring that only one bit changes at each step. What does this remind you of?

1. There are a total of 2^n unique vectors. They represent the binary numbers, where each digit is a bit. The total number of m-dimensional subcubes is given as $2^{n-m}\binom{n}{m}$ for any values of n, m. The reason is as follows: For each of the 2^n vertices of a hypercube, there are $\binom{n}{m}$ ways to choose a collection of m edges incident to that vertex; each of these collections defines one of the m-dimensional faces incident to the considered vertex. Repeating for each vertex of the hypercube, we repeat the process 2^m times for each face of the hypercube, so we need to compute $\frac{2^n\binom{n}{m}}{2^m} = 2^{n-m}\binom{n}{m}$.
2. Any drawing with a consistent label system is acceptable; you should have a tesseract for the 4d case.
3. This path is how the **Gray code** is constructed, because it is the path that **changes only one bit at a time** from $(0000)_2$ to $(1111)_2$.

[1] Hypercube

Working with number systems

Problem 7 (6 points)
Perform the following number system conversions.

1. Unsigned:

$$(1101)_2 = (?)_{10} \tag{1a}$$
$$(100111)_2 = (?)_{10} \tag{1b}$$
$$(10101.1001)_2 = (?)_{10} \tag{1c}$$
$$(36)_{10} = (?)_2 \tag{1d}$$
$$(132)_{10} = (?)_2 \tag{1e}$$
$$(11.75)_{10} = (?)_2 \tag{1f}$$
$$(ADC)_{16} = (?)_2 \tag{1g}$$
$$(CA)_{16} = (?)_8 \tag{1h}$$
$$(1010)_2 = (?)_8 \tag{1i}$$

2. Signed magnitude:

$$(100111)_2 = (?)_{10} \tag{2a}$$
$$(-86)_{10} = (?)_2 \tag{2b}$$

3. Two's complement:

$$(11101011)_2 = (?)_{10} \tag{3a}$$
$$(-77)_{10} = (?)_2 \tag{3b}$$

Solution:

1.

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 1 = (13)_{10} \tag{4a}$$
$$(100111)_2 = 2^5 + 2^3 + 2^1 + 2^0 = (39)_{10} \tag{4b}$$
$$(10101.1001)_2 = ((2^4 + 2^2 + 2^0).(2^{-1} + 2^{-4}))_{10} = (21.5625)_{10} \tag{4c}$$
$$(36)_{10} = (100100)_2 \tag{4d}$$
$$(132)_{10} = (10000100)_2 \tag{4e}$$
$$(11.75)_{10} = (1011.11)_2 \tag{4f}$$
$$(ADC)_{16} = (1010\,1101\,1100)_2 \tag{4g}$$
$$(CA)_{16} = (312)_8 \tag{4h}$$
$$(1010)_2 = (12)_8 \tag{4i}$$

2.

$$(100111)_2 = -(00111)_2 = (-7)_{10} \tag{5a}$$
$$(-86)_{10} = (1(86)_{10})_2 = (11010110)_2 \tag{5b}$$

3.

$$(11101011)_2 = (-21)_{10} \tag{6a}$$
$$(-77)_{10} = (10110011)_2 \tag{6b}$$

Binary arithmetic round 2

Problem 8 (6 points)

Perform signed two's complements binary arithmetic operations. Indicate whether overflow occured.

$$(0010010)_2 + (1001111)_2 = (?)_2 \quad (7a)$$
$$(0010010)_2 - (1001111)_2 = (?)_2 \quad (7b)$$
$$(0010010)_2 - (0110001)_2 = (?)_2 \quad (7c)$$

Solution

Recall that the MSB (leftmost bit) represents the sign; it is 0 if the number is positive and 1 if the number is negative.

1.

$$\begin{aligned} &0010010 \\ +&1001111 \\ =&1100001 \end{aligned}$$

Note that overflow did not occur here. Next, to check your work, convert the result back to base-10. To do this, treat the bits to the right of the MSB as a positive number:

$$1\ \underbrace{100001}_{=2^0+2^5=33},$$

then subtract $2^{i_{\text{MSB}}}$ from the result:

$$33 - 2^{i_{\text{MSB}}} = 33 - 2^6 = -31.$$

2. To perform subtraction, note that this is the same as addition with negative 2's complement values. That is, we can equivalently perform

$$\begin{aligned} &0010010 \\ +&0110001 \\ =&1000011. \end{aligned}$$

However, note that $n = 7$ bits are used here; therefore, setting $(S)_{10}$ to be the sum, the base-10 range of the sum that is admissable in two's complement is given as

$$\begin{aligned} -2^{n-1} \leq (S)_{10} \leq\ & 2^{n-1} - 1 \\ =\ & 2^6 - 1 \\ =\ & 63. \end{aligned}$$

Thus, we're trying to represent a number outside of of the range of our number system, since the base-10 number we desire is 67. **Overflow occurred, the result is invalid without sign extension:** We add leading zeros to yield:

$$\begin{aligned} & 00010010 \\ +& 00110001 \\ =& 01000011 \\ =& 2^6 + 2^1 + 2^0 \\ =& (67)_{10}, \end{aligned}$$

which is the desired quantity.

3. Here, we can equivalently consider

$$\begin{aligned} & 0010010 \\ +& 1001111 \\ =& 1100001. \end{aligned}$$

No overflow occurred.