Study Guide for Midterm 1

Dr. Jordan Hanson - Whittier College Dept. of Physics and Astronomy

March 22, 2020

1 Chapter 1 - Introductory Concepts

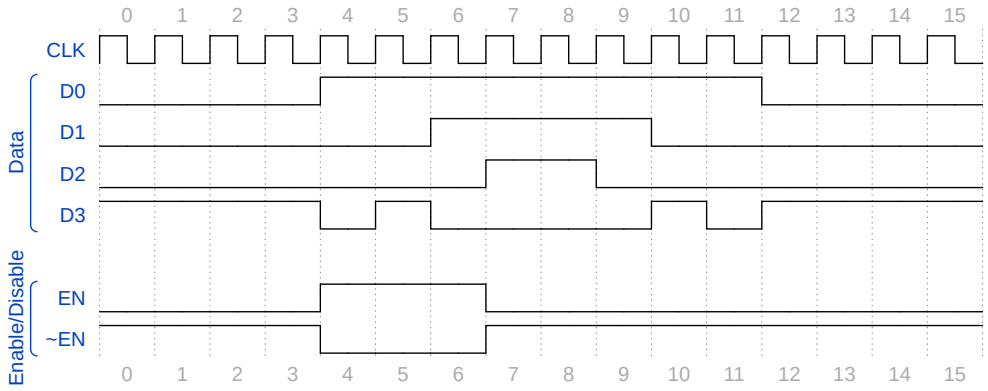


Figure 1: A timing diagram including a clock signal (CLK), a 4-bit parallel data stream (D0-D3), and enable/disable signals (EN/∼EN).

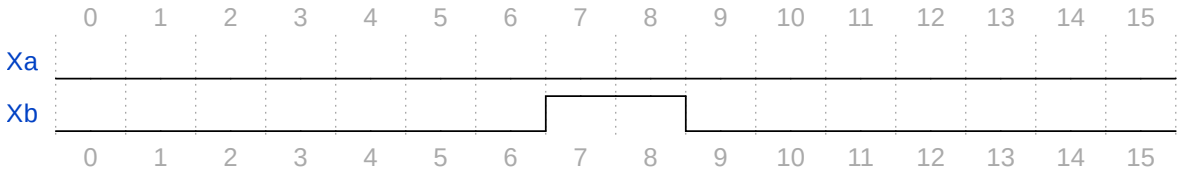
1. Consider Fig. 1. (a) If the clock frequency is 10 MHz, what is the period? (b) What is the length of time represented in the waveform? (c) What is the length of the pulse in channel $D0$?

   (a) $0.1\mu s$*, (b)* $1.6\mu s$ *(c)* $0.8\mu s$*.*

2. (a) What is the bit sequence represented by $D3$? (b) If $D0$ is a periodic signal, what is the duty cycle?

   (a) 1111010000101111 *(b)* 50%*.*

3. (a) Imagine that the four D_i signals enter an AND gate, along with the EN signal. Draw the resulting timing diagram. (b) Imagine that the four D_i signals enter an AND gate, along with the ∼EN signal. Draw the resulting timing diagram.



4. Suppose D_i represents parallel data. Each line gives 16 bits, and the clock frequency is 10 MHz. (a) How many total bits are being sent across the D_i, and how long does it take? What is the ratio of total bits to time? (b) What would this time be if the data was transferred *serially*, instead of in *parallel*?

   (a) 64 bits, in $1.6\mu s$*, so the ratio is* 40 *Megabits per second, (b) Four times as long, or 6.4* μs*.*

2 Chapter 2 - Number Systems, Operations, and Codes

1. Convert the following numbers to binary: (a) 33 (b) 65 (c) 129. (d) What does each number have in common? (e) Sketch a short python script to convert a decimal number to binary, and return the result.

   (a) 100001 *(b)* 1000001 *(c)* 10000001 *(d) Each number is one higher than a multiple of two, meaning the MSB and LSB are 1 with the rest of the digits being 0. (e)*

```
DecimalToBinary(num):
        if num > 1: #1 in binary is just 1
            DecimalToBinary(num // 2) #Call recursively to get higher MSBs
            print num % 2 #If the number is divisible by current weight, 0, else 1
```

2. Represent the following numbers in 8-bit binary: (a) -31 (b) -15 (c) -7. (d) Show that each number, when added to its opposite is zero, in binary. (e) Did you know that the *shift* operators in C and Python can act as binary multipliers? For example `2 << 1` results in 4. Why? Because the code is saying 0010 $\rightarrow$ 0100. What is the result of the following codes in Python?

```
j = 1
for i in range(0,5):
print(j << i)
```

(a) 00001*, (b)* 0001*, (c)* 001*, (d)* $11111 + 00001 = 1(00000)$*, drop the MSB. Similarly,* $1111 + 0001 = 1(0000)$*,* $111 + 001 = 1(000)$*.*

3. What is the largest number that can be represented in binary with (a) $n = 4$ (b) $n = 8$ bits, if one bit must be used as a sign bit?

(a) 15 (b) 127 *(The range is -128 to 127 including 0).*

4. Convert the following numbers to binary: (a) 0xFE (b) 0x7F (c) 0x3F.

(a) $0xFE = 11111110$*, (b)* $0x7F = 01111111$ *(c)* $0x3F = 00111111$*.*

5. Convert to hexadecimal: 1000100001000100

$0b1000100001000100 = 0x8866$

6. What is the following gray code sequence: 0011 0010 0111 0100 1110?

2...3...5...7...11 *(The first five prime numbers).*

3 Chapter 3 - Logic Gates

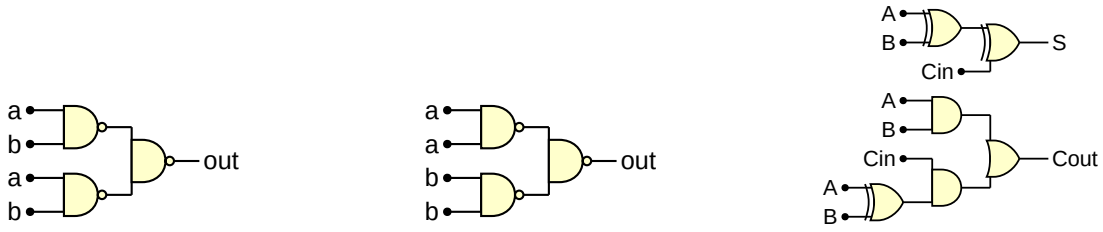


Figure 2: Examples of logic gate combinations.

1. Generate the truth tables for the gate diagrams in Fig. 2. Pay attention to the domain.

(a) Like an AND gate:

a	b	out
0	0	0
0	1	0
1	0	0
1	1	1

(b) Like an OR gate:

a	b	out
0	0	1
0	1	1
1	0	1
1	1	0

(c) Like an:

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

2. Identify the actions of the gate combinations in the previous problem from the truth tables. (a) Fig. 2, left: **AND gate** (b) Fig. 2, middle: **OR gate** (c) Fig. 2, right: **2-bit adder with carry**.

3. Suppose the signals in Fig. 3 are fed into the circuit in Fig. 2, right. Fill in the S and Cout signals in Fig. 3

See below in the figure.

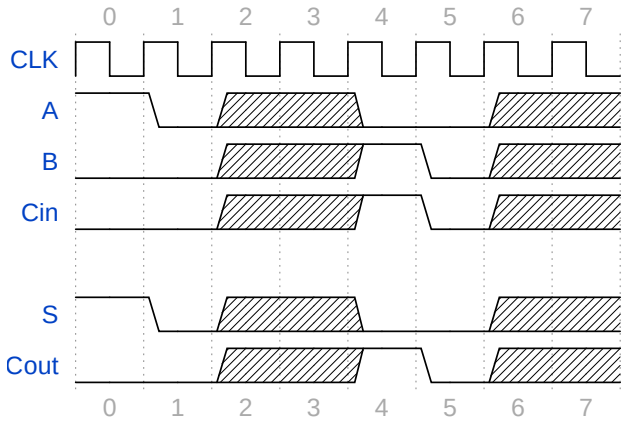


Figure 3: An example timing diagram for Fig. 2, right. The shaded regions indicate "don't care" conditions, or times when the data is not relevant.

4. Imagine a system for an electronic door lock which requires two electronic keys (two signals), S1 and S2. The circuit has a red LED, a yellow LED, and a green LED. The system unlocks when the output LOCK signal is LOW, caused by a LOW S1 and a LOW S2. The red LED is activated when S1 and S2 are both HIGH. The yellow LED is activated when S1 and S2 are different. The green LED is activated when the system is unlocked. *By default, S1 and S2 are HIGH.* Draw the solution below.

   Let S1 and S2 be the two signals. They are by default HIGH signals. They are fed into an OR gate. The output of the OR gate is the Unlock signal UL. A plain HIGH signal is connected to a resister and then to the green LED, which is connected to UL. Thus, when both S1 and S2 go low (active low), then the green LED will be pulled down and on as UL goes LOW to unlock the lock. S1 and S2 are connected in parallel to an AND gate, which powers a red LED and series resister with a HIGH output. Finally, an XOR gate powers a resister and yellow LED when S1 and S2 are different.

4 Chapter 4 - Boolean Algebra and Logic Simplification

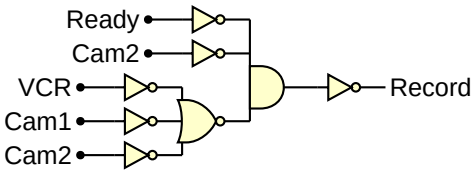


Figure 4: A gate combination triggering a signal named "record."

1. Consider the circuit in Fig. 4. (a) Write a logic expression for *Record*. (b) Simplify the expression to as minimal a form as possible. (c) Are any signals or gates unnecessary?

   (a-b) The output record always evaluates to true:

$$\bar{Rec} = \bar{R}\bar{C_2}\overline{(\bar{V} + \bar{C_1} + \bar{C_2})} \tag{1}$$
$$\bar{Rec} = \bar{R}\bar{C_2}VC_1C_2 \tag{2}$$
$$\bar{Rec} = \bar{R}VC_1(C_2\bar{C_2}) = 0 \tag{3}$$
$$\bar{Rec} = 0 \tag{4}$$
$$Rec = 1 \tag{5}$$

   (c) What if we remove C_2? Then the expression does not always evaluate to true $Rec = R + \bar{V} + \bar{C_1}$.

2. Consider again the logic expression that represents the circuit in Fig. 4. (a) Write the corresponding domain-4 Karnaugh map. (b) Use the Karnaugh map to produce the S-POS expression and truth table.

   (a) This is a special case of the Karnaugh map: all boxes contain a 1, and the largest group is 16. All variables are redundant, so the resulting expression is just TRUE. (b) The truth table is trivial, and the S-POS expression is empty (cannot be written in standard form).

5 Chapter 5 - Combinatorial Logic Analysis

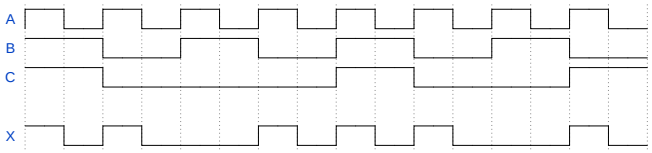


Figure 5: Timing diagram for Sec. 5.

1. (a) Draw the circuit corresponding to the timing diagram in Fig. 5. (b) Create the Karnaugh map. (c) Using the Karnaugh map, write the simplest expression for this logic function and re-draw the circuit.

   (a) The circuit follows the expression $X = ABC + A\bar{B}C + A\bar{B}\bar{C}$*. It requires one OR gate and three 3-input AND gates. (b) The Karnaugh map is the usual domain-3 map with two groups of two true states each. (c) Using the Karnaugh map, the expression reduces to* $X = A\bar{B} + AC$*, which requires two 2-input AND gates and one 2-input OR gate.*