COMPUTER LOGIC AND DIGITAL CIRCUIT DESIGN (PHYS306/COSC330): UNIT 1.3

Jordan Hanson

January 15, 2018

Whittier College Department of Physics and Astronomy

SUMMARY

Reading: Digital Fundamentals (DF) Ch. 2 (see Moodle)

1. Number representation
2. Binary conversions
3. Binary arithmetic
4. The floating-point system
5. Hexidecimals, Binary-Coded Decimals (BCD), Gray codes, and ASCII

Homework: exercises 1-40 Ch. 2 (DF) (two weeks)

NUMBER REPRESENTATION

Questions:

- A simple question: how many students do we have in this class?
- Una simple pregunta: ¿Cuántos estudiantes tenemos en esta clase?
- Une question simple: Combien d'étudiants est-ce que nous avons dans cette classe?

What languages do computers speak? How can we store and transmit numbers through circuits? (We cannot use voltage magnitudes).

Consider the number 37

Numbers	Expanded notation
0d37	$3 \times 10^1 + 7 \times 10^0$
0b100101	$1 \times 2^5 + 1 \times 2^2 + 1 \times 2^0$
0x25	$2 \times 16^1 + 5 \times 16^0$

Consider the number 412

Numbers	Expanded notation
0d412	$4 \times 10^2 + 1 \times 10^1 + 2 \times 10^0$
0b110011100	$1 \times 2^8 + 1 \times 2^7 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2$
0x100	1×16^2

Number representations - Digits, weights, and a common base

- A number is written with **digits** that have **weights**.
- A **weight** is a power of a **base**.
- At right, the **base** is ten, and the **weights** are 10^2, 10^1, and 10^0.
- The **digits** are four, one, and two.
- The digits cannot represent numbers larger than the base.

Expanded notation

- $412 = 4 \times 10^2 + 1 \times 10^1 + 2 \times 10^0$

Number representations - (Aside) please use scientific notation, and here is why: digit minimization!

- Scientific notation **factors the largest weight**.
- Results in weights that are less than one.
- Weights that are less than one go to the right of the **decimal point**.
- Return here with *floating-point* representation.

Expanded notation

- $412,000,000 = 4 \times 10^8 + 1 \times 10^7 + 2 \times 10^6 + 0 \times 10^5 + 0 \times 10^4 + 0 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 0 \times 10^0$
- $412,000,000 = 4.12 \times 10^8 = (4 \times 10^0 + 1 \times 10^{-1} + 2 \times 10^{-2}) \times 10^8$

Number representations - (Aside) please use scientific notation, and here is why: arithmetic operations with large numbers!

1. $4200 \times 4200 = (4.2 \times 10^3)^2 = (4.2)^2 \times 10^6 \approx 16 \times 10^6$
2. $\approx 17.6 \times 10^6$ if you account for the 0.2 ...

---

1. $4000/3000 = 4 \times 10^3 \times \frac{1}{3} \times 10^{-3} = \frac{4}{3}$
2. $\frac{4}{3} \approx 1.33$

Expand the following numbers to expanded decimal notation:

- -10.432
- 800,000,144

Expand the following numbers to expanded binary notation:

- 10011010
- 11110000

Convert the following decimal numbers to binary notation:

- 260
- 560

Volunteer to board? - Key is explaining how you did the binary conversions

BINARY CONVERSION

How did you do the conversions to binary? Is there a systematic what to do this?

- *Successive Approximation* - like a number puzzle (*Sum of Weights Method*)
- *Successive Division Method* - Example of an algorithm

Successive approximation technique (does this remind you of doing division in your head?) 260...$2^8 = 256$. Now we need four more...$4 = 2^2$. So $2^8 + 2^2 = 0b100000100$

What is 328 divided by 3? Ok try 100 because three times one hundred is close...

Successive Division Method

$0d412 = 0b110011100 = 2^8 + 2^7 + 2^4 + 2^3 + 2^2$

Algorithm:

1. Divide the decimal number by 2, and write down the remainder. This is the *least-significant bit* or LSB.
2. Keep dividing and recording the remainders in order, until you reach a dividend of 1.
3. $1/2 = 0r1$, so the *most-significant bit*, or MSB, is always 1.

Convert 412 to binary using the successive division method.

Convert the numbers at right to binary.

- $2^0 = 1$
- $2^1 = 2$
- $2^2 = 4$
- $2^3 = 8$
- $2^4 = 16$
- $2^5 = 32$
- $2^6 = 64$
- $2^7 = 128$

1. 93
2. 189
3. 270

Note: how many bits do you need for that last one? For binary, show that the highest representable number with n bits is $2^n - 1$.

In decimal notation, we represent numbers $\in [0-1]$ with digits to the right of the *decimal point*.

In expanded notation:
$42.42 = 4 \times 10^1 + 2 \times 10^0 \ \ . \ \ 4 \times 10^{-1} + 2 \times 10^{-2}$

We have a similar notation in other number systems:
$101.11 = 2^2 + 2^0 \ \ . \ \ 2^{-1} + 2^{-2}$

Successive Multiplication Method

$0d0.48 = 0b0.011110101 = 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-7} + 2^{-9}$

Algorithm:

1. Multiply the decimal fraction by 2, and write down the *carry* (the 1 to the left of the decimal point, else 0). This is the *most-significant bit* or MSB.
2. Keep multiplying and recording the carries in order, to the desired precision.
3. Watch for repeating patterns.

Convert 0.48 to binary using the successive multiplication method until you identify a repeating pattern.

BINARY ARITHMETIC

Binary arithmetic, like decimal arithmetic, relies upon *carries* and *borrows*. For example in decimal:

- $8 + 7 = 5c1 = 15$
- $3 + 4 = 7c0 = 7$
- $8 - 2 = 6b0 = 6$
- $10 - 5 = (0b10) - 5 = 5$
- $13 - 7 = (0b10) - 4 = 6$

In binary there are limited combinations, so we may form a set of addition and subtraction rules that describe all possibilities:

- $0 + 0 = 0$
- $1 + 0 = 1$
- $0 + 1 = 1$
- $1 + 1 = 0c1$
- $0 - 0 = 0$
- $1 - 0 = 1$
- $0 - 1 = 0b1$
- $1 - 1 = 0$

In the fourth rule for the addition set, the carry works the same way as a decimal carry: we add that 1 to the digit corresponding to the next power of 2. Similarly, in the third rule of the subtraction set, we subtract the 1 on the left side of the equation from a 1 from the digit corresponding to the next highest power of 2 available.

Complete the following additions and subtractions:

- $10 + 11$
- $111 + 1$
- $111 + 111$
- $1010 + 101$

- $10 - 11$
- $111 - 1$
- $111 - 111$
- $1010 - 101$

Binary multiplication stems from another set of rules:

- $0 \times 0 = 0$
- $1 \times 0 = 0$
- $0 \times 1 = 0$
- $1 \times 1 = 1$

Binary division is exactly the same as decimal long division. Let's work these examples on the board:

- 145/12
- 1101/101
- 45/3
- 111/10

Of what logic operation does binary multiplication remind you?

We need a few tools to improve our arithmetic techniques in order to install these functions into circuits. Consider two actions:

- NOT to the bit sequence of a number: **1's compliment**
- Add 1 to the 1's compliment: **2's compliment**

Logical operator representation of these actions for 8-bits:

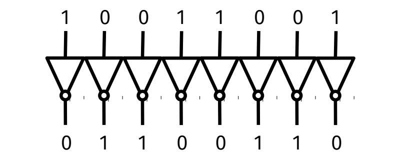

Figure 1: Representation of 1's compliment.

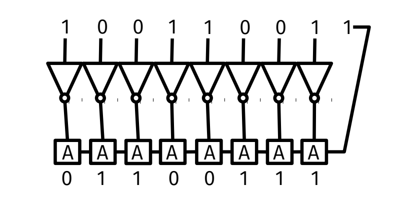

Figure 2: Representation of 2's compliment.

Exercises, and a realization for the purpose of 2's compliment.

Take the 1's compliment of the right-hand number, and complete the addition:

- $111 + (000)$
- $111 + (111)$

Take the 2's compliment of the right-hand number, and complete the addition:

- $111 + (000)$
- $111 + (111)$

Same, but drop the MSB:

- $1010 + (1010)$
- $1100 + (1100)$

For what purpose are we going to use the 2's compliment?

The answer: representing negative numbers. For *2's compliment form signed binary numbers, the MSB is the* ***sign bit***. To convert to decimal, give the MSB a negative weight:

Example: 10101010
This is a negative number because the MSB is a 1. Converting in expanded form: $-2^7 + 2^5 + 2^3 + 2^1 = -128 + 32 + 8 + 2 = -86$.

Consider the effect on the maximum *range* of binary numbers after sacrificing the MSB to flag numbers less than zero. For 8-bits unsigned numbers, the range is [0-255]:

n-bits	8	16
Lowest unsigned	0	0
Highest unsigned	255	65535
Lowest signed	-128	-32768
Highest signed	127	32767

Table 1: The ranges of unsigned and signed numbers using *n* digits.

By changing the role of the MSB, we are not reducing the *range* of numbers, just its *location*. Related to *conservation of information*...

CONCLUSION

Reading: Digital Fundamentals (DF) Ch. 2 (see Moodle)

1. Number representation
2. Binary conversions
3. Binary arithmetic
4. The floating-point system
5. Hexidecimals, Binary-Coded Decimals (BCD), Gray codes, and ASCII

Homework: exercises 1-40 Ch. 2 (DF) (two weeks)