Lecture 1

or “01001000 01101001”

Announcements

My website is still down, I will have it up by tonight

Digital Systems

- They are everywhere
- Usually operate on 2 valued signals
  - What do we know of that has 2 values?
- Take an arbitrary number of inputs and produce an arbitrary number of outputs
- Some systems use a clock for timing

Kinds of Digital Systems

Combinational

Sequential

Clocks

- A clock in this case is a bit different that what you are used to
- Basically acts like a metronome, (tick, tock, tick, tock) with each tick being a high state and each tock being a low state
- Helps with synchronization

Pictures from the book!

Example of a System

- A system with 3 inputs, A, B, and C, and one output Z, such that Z = 1 if and only if (iff) 2 of the inputs are 1 (Example 1.1 in the book)

Real life example

- A traffic controller on 2 streets: the light is green on each street for a fixed period of time, the goes to yellow for another fixed period and finally to red. The only input for this system is the clock

Truth tables

Time to do something on the board!

Number systems

- Integers are written using a positional number system

$N = a_{n-1}r^{n-1} + a_{n-2}r^{n-2} + a_{n-3}r^{n-3} + \ldots + a_2r^2 + a_1r + a_0$

where $0 <= a_i < r$

r = base (2, 10, 16, etc)

Different number systems

- We all should know these, but I will put them up anyways
- Binary: 1011_2
- Decimal: 11_{10}
- Hex: B_{16}
- These are all different ways to represent the same number

Conversion between them?

- How do we convert from binary to decimal?
  - Example: 101101_2 = ?
- Each digit is a power of 2:
  - $2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$
  - Match these up and add the ones with a 1 together
  - $2^5 + 2^3 + 2^2 + 2^0 = 32 + 8 + 4 + 1 = 45_{10}$

How many bits?

How many bits does it take to represent the following numbers?

- 459_{10}
- 1025_{10}

Good thing to memorize

- It’s probably a good idea to memorize the first 10 or so powers of 2, that will make your life a bit easier going forward

Powers of 2

n	**2^n**	**n**	**2^n**
1	2	6	64
2	4	7	128
3	8	8	256
4	16	9	512
5	32	10	1024

How many bits?

How many bits does it take to represent the following numbers?

- 459_{10} - 9 (111001011_2)
- 1025_{10} - 11 (10000000001_2)

Methods for reverse

- We have two methods to do this conversion (Decimal to binary)
  - 1.) Repeatedly subtract the largest power of 2 less than that number and put a 1 in the corresponding position
  - 2.) Repeatedly divide the number by 2 and put the remainder from right to left

Time to try both of these

- Convert the following numbers to binary:
  - 21_{10}
  - 34_{10}

Hexadecimal

- Base 16 (numbers 0-F)
- Used as shorthand for binary
- Group 4 bits in binary to get 1 digit of hexadecimal

Example

- Convert the following to hex:
  - 10110011_2 = ?

Example solution

- First we will split this up into 4 bit parts
  - 1011 0011
- Now we will convert them to hex by part
  - First digit (0011) is 3, which is the same in base 10 and 16
  - Next digit is $2^3 + 2^1 + 2^0 = 11_{10}$
  - $11_{10} = ?_{16}$
  - 0-F is our hex number set, so 11 = B
- Our answer is $B3_{16}$

Another example

- $1111\ 0110_2$

Decimal to Hex

Pretty much the same as binary

Divide by the base (16), put remainders from right to left

Example

33_{10}

Example solution

33/16 = 2 remainder 1

2/16 = 0 remainder 2

21_{16}

Binary addition

- Same as the addition you learned way back when
- In base 10 you would add the digits, mod 10 for the place digit, divide by 10 and floor the result for the carry digit
- We will do the same for this, but instead of base 10, we have base 2

Example

- 111
  + 10

???

One-bit adder

a	**b**	**c_{in}**	**c_{out}**	**s**
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

That’s it for today

- We will have a quiz next class!
- Probably will have homework to assign and a syllabus to hand out, but they will also be online