EELE 461/561 – Digital System Design

Module #1 – Digital Signaling

- **Topics**

  1. Course Overview
  2. Signaling Definitions
  3. Signal Composition

- **Textbook Reading Assignments**

  1. 1.1 – 1.7, 2.1, 2.10

- **What you should be able to do after this module**

  1. Describe what signal integrity is and why it is important
  2. Understand the terminology used in digital signaling
  3. Describe and use the risetime-bandwidth-product
  4. Describe the frequency components of a digital signal

Course Overview

- **Instructor:**
  Brock J. LaMeres
  Office : 533 Cobleigh Hall
  Phone : (406)-994-5987
  Email : lameres@ece.montana.edu
  Web : www.coe.montana.edu/ee/lameres/

- **Time / Location:**
  Lecture : Monday, Wednesday, Friday
  9:00am – 9:50am
  110 EPS

- **Textbook:**
  "Signal & Power Integrity Simplified",
  Eric Bogatin,
  Prentice Hall, 2nd edition 2009

- **Website:**
  www.coe.montana.edu/ee/lameres/courses/eele461_spring12

  - all handouts and homework are found on the website
  - it is your responsibility to download assignments

Course Overview

- **Office Hours:** Check instructor website for most recent hours

- **Requisites:** Pre-requisite EE308, EE334, EE371 (or consent of instructor)

- **Grading:**

Homework	- 25%
Exam #1	- 25%
Exam #2	- 25%
Final Project	- 25%

- Homework Assignments are due at the beginning of class on indicated date.

- Late homework will be accepted for one week after the due date with a penalty of 50% point reduction. No credit will be given for assignments over one week late.

- No make up exams will be given. Plan on being available on the exam dates.

Course Overview

- **Final Project:**

  - A final design project will be assigned midway through the semester

  - the project will consist of a high speed link design including architecture, design, and simulation

  - a paper will be required that explains the design, its operation, simulation results, and layout

  - an in-class presentation will be required during the last week of the semester

  - you may work in teams of 2

Course Content

- **What is this course?**

  - We will look at how to design and analyze digital communication links in a wireline medium (i.e., conducting wires vs. wireless)

  - a communication link is the circuitry (Tx, Rx, and interconnect) used to transfer information between logical blocks

    ex) uP to memory system or peripherals
    computer-to-computer networking

  - We will look at the analog effects of a digital signal in order to understand how to design chip-to-chip communication links

  - We will learn how to create a noise budget that considers voltage and timing noise

  - We will see that the physical interconnect between IC's tends to limit the speed at which data can be transferred

  - We will also see that at modern integrated circuit speeds, interconnect needs to be treated as a transmission line (as opposed to just a simple capacitance)

  - We will learn to use modern CAD tools to help design and analyze these links

Course Content

- **What topics will be covered?**

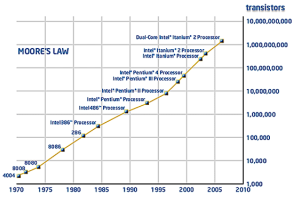
1)	Signaling	(Exam #1 Topics)
2)	Interconnect Analysis	
3)	Interconnect Fabrication and Modeling	
4)	Noise Sources & Budgeting	
5)	Power distribution	
6)	Link Architectures	
7)	Measurement Techniques	(Exam #2 Topics)
8)	Modern Bus Architectures	
9)	Design Trade-offs	

Course Content

- **Why do we need this course?**
  - We create all of our digital circuits using integrated circuit technology
  - In order for two digital circuits to communicate information, the transmitter (Tx) sends a signal to the receiver (Rx).
  - As long as the DC specifications are met at the Rx, the receiver can determine whether a 1 or 0 was sent.
  - As along as sufficient time is allowed before and after the associated clock, the Rx flip-flops can latch and store the information that was sent.

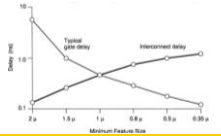
Course Content

- **Why do we need this course?**
  - So now all we need to do is meet DC and Timing specifications, and then go faster…
  - In the beginning, integrated circuit performance was the limiting factor in going faster.



Course Content

- **Why do we need this course?**
  - We used to only care about the delay associated with the IC gates.
  - On-chip and off-chip interconnect were modeled as capacitances, but they were secondary effects
  - As we moved beyond 1*um* process capability, the on-chip interconnect began to contribute more delay to the circuit than the gate itself
  - Also at this time, the rise times of the IC drivers became fast enough so that off-chip interconnect (i.e., PCB's and cables) needed to be modeled as transmission lines.
  - The mismatch in IC circuit performance and off-chip interconnect performance has to do with the manufacturing processes used (IC=photolithography/chemistry, Interconnect=chemistry/mechanical)



Course Content

- **Why do we need this course?**
  - Now, modern digital system speed is dominated by the interconnect between the integrated circuits making the Tx and Rx.
  - The interconnect must be modeled not as simple wires, but as distributed capacitances, inductances, and impedances.
  - This modeling also applies to the interconnect of the power distribution (i.e., V_{DD} and GND)
  - We now have to consider the analog behavior of the signal as it travels through the interconnect to understand how fast and how robust a digital system can be.
  - This course will present the techniques to design, model, analyze, and measure a modern digital system at the physical level.
  - This type of design and analysis is also called

"Signal Integrity"

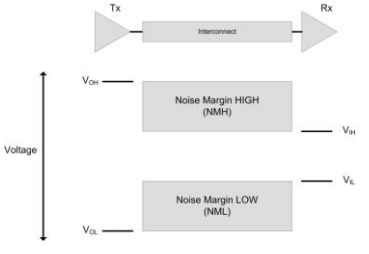
Course Content

- **Signal Integrity**
  - Signal integrity consists of 3 categories.
    1) Voltage Noise
       - sometimes just called *Noise*
       - what factors in the system eat into our Voltage Noise Margin
    2) Timing Noise
       - also called *Jitter*
       - the factors in the system eat into our Timing Margin
    3) Electromagnetic Interference (EMI)
       - when our system creates unwanted energy that interferes with the standards set by the Federal Communication Commission (FCC)

Signaling

- **Digital Signaling**

Voltage Noise



- what we're really after is a stable region between V_{IH} and V_{IL}

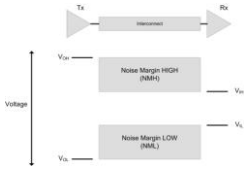
Signaling

- **Digital Signaling**

Voltage Noise

- this area of Signal Integrity can further be sub-categorized into 4 distinct sources of noise:

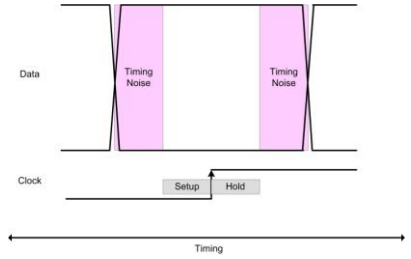
1) Single Net Quality
2) Cross-talk
3) Power Supply Quality
4) EMI



Signaling

- **Digital Signaling**

Timing Noise



- we need to ensure that the data is stable long enough to meet the setup/hold specification of the Rx

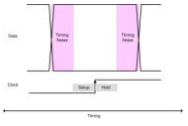
Signaling

- **Digital Signaling**

Timing Noise

- the entire timing of a signal can be broken down into 3 distinct parts:

1) Logical timing
   - i.e., the time for the gates to switch on chip
2) Interconnect Propagation
   - the time we have to wait for the signal to propagate
   - the time that we have to wait for any distortions to settle out
3) Receiver Setup/Hold
   - the time the data must remain stable around the receiver clock event



Signaling

- **Digital Signaling**

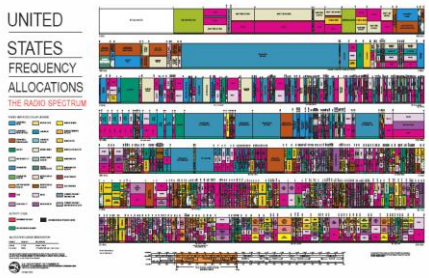
Electromagnetic Interference

- The FCC sets standards for how much Electromagnetic energy can be transmitted in a given frequency band.
- Some of the bands are *licensed* (i.e., you have to pay to use them)
- If you make a product that isn't intended to use one of the FCC bands, you cannot inadvertently transmit energy into an FCC band above a certain level.
- This is considered *breaking the law* and you will not be able to ship your product if it does this.
- Products undergo EMI testing prior to shipping.
- As our systems go faster, it becomes easier for the energy to radiate out of our product because our small interconnect begins to look like an antenna

Signaling

- **Digital Signaling**

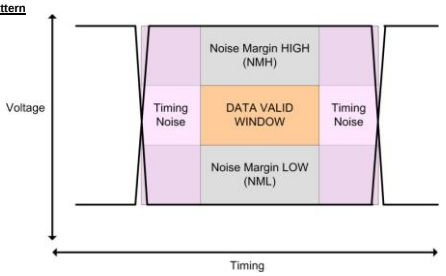
Electromagnetic Interference (EMI)



Signaling

- **Digital Signaling**

Eye Pattern

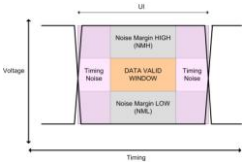


- we can combine the voltage and timing specifications into one diagram called an "Eye Pattern"

Signaling

- **Digital Signaling**

Data Valid Window - the stable region in Voltage & Time around a timing event (i.e, a clock) that will guarantee a logic level is received



Unit Interval (UI) - the time between transitions on the data line

Max Data Rate - the fastest we can transmit information on the data line: $DR_{max} = \frac{1}{UI}$

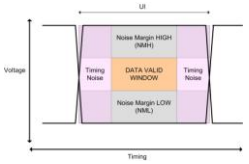
Signaling

- **Digital Signaling**

- in this class, we will learn how to create a *Noise Budget* that tabulates the contribution of each noise sources in our system.

- we can then predict if our system will reliably transmit data (i.e., does it work?) and how fast it can go.

- In order to do this, we must understand how to model and analyze the all of the components in a system (Tx, Interconnect, and Rx)



Signal Composition

- **A Digital Link**

- A digital communication *link*, consists of 3 elements

1) A Driver (Tx)
2) Interconnect
3) A Receiver (Rx)

- The Driver and Receiver are constructed using Integrated Circuit Technology (ICs)

- The interconnect consists of all *wires* that connect the Tx and Rx. These include:

On-chip traces, package leads, PCB traces, connectors, and cables

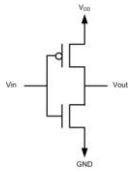
Signal Composition

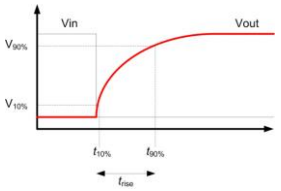
- **Driver Risetime**

- The driver produces a digital edge

- We define the risetime (t_{rise}) as the time it takes for the edge to go from 10% of steady state to 90% of steady state.

- If the IC driver is NOT connected to any interconnect (on-chip or package), then the rise time will follow an RC exponential ramp where the R and C come from the Tx transistors.



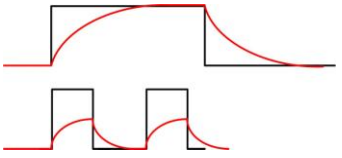


Signal Composition

- **Risetime and Frequency**

- The speed of the driver risetime is important because it tells us how fast we can possibly switch logic states (i.e., how fast of a frequency can be generated.

- If we try to switch too fast, there isn't enough time for the rise time to get to its final value. This leads to data loss.



Signal Composition

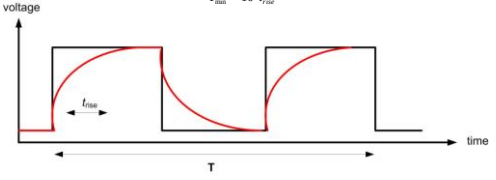
- **Risetime and Frequency**

- While the relative speed of the risetime to Period (T) is arbitrary, we can use a rule of thumb that it needs to be ~10% of the Fastest Period.

- We'll see how this is derived when we look at the Fourier Transform of a Square Wave

$$t_{rise} = (0.1) \cdot T_{min}$$

$$F_{max} = \frac{1}{T_{min}} = \frac{1}{10 \cdot t_{rise}}$$



Signal Composition

- **Risetime and Bandwidth**

  - Note that the driver rise time is NOT always what we get at the Receiver end of the interconnect

  - We can think of this as a Linear Time Invariant (LTI) System where we have:

    1) A forcing function, x(t) (i.e., the driver edge and pattern)
    2) A transfer function, h(t) (i.e., the response of the interconnect)
    3) The system output, y(t) (i.e., what is seen at the Rx)

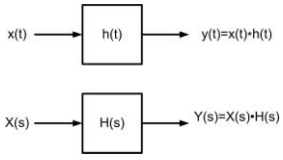
Signal Composition

- **Risetime and Bandwidth**

  - A very nice feature of an LTI system is that we can transform between the Time and Frequency domains easily.

  - Not only do the functions transform, but so do operators.

  - One of the main operator transforms that we take advantage of is *convolution* in the time domain becomes *multiplication* in the frequency domain.



Signal Composition

- **Risetime and Bandwidth**

  - We like to use the Frequency Domain because some things are easier to observe and some operations are easier.

  - We are interested in what our forcing function looks like in the Frequency Domain because it tells us:

    1) what spectrum of energy is stimulated in the system?

    2) what bandwidth our Interconnect needs in order to pass X(s)?

       - this helps us scale the cost of our materials and construction for the application.

         i.e., we don't need to build a $100k, 100GHz, PCB if we only stimulate 1GHz of spectral energy.

    3) what bandwidth do our equivalent circuit models need to have?

       - higher frequency models take more simulation time to compute.

       - we want sufficient bandwidth for accuracy, but not too much that our simulations run longer than necessary.

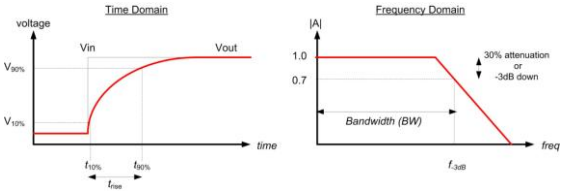
Signal Composition

- **Risetime and Bandwidth**

  - Let's start by creating a rule of thumb that will allow us to easily switch between the Time and Frequency Domain for our RC Exponential Ramp.

  - This is a relevant transform because this tends to be what CMOS edges look like.

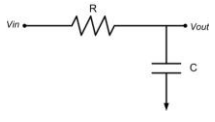
  - First, let's define two metrics that we will use to describe the RC circuit.



Signal Composition

- **Risetime and Bandwidth**

  - Let's Solve the RC circuit and derive the expressions for risetime and bandwidth



$$V_{out}(s) = V_{in}(s) \cdot \left(\frac{\left(\frac{1}{Cs}\right)}{R + \left(\frac{1}{Cs}\right)} \right)$$

$$V_{out}(s) = V_{in}(s) \cdot \left(\frac{1}{1 + RCs} \right)$$

Signal Composition

- **Risetime and Bandwidth**

  - RC circuit cont…

We want to get this into a form so that we can use the Laplace Transform for an exponential approach.

Let's set $\propto$= 1/RC

$$\left(1 - e^{-\alpha \cdot t}\right) \cdot u(t) \Leftrightarrow \frac{\alpha}{s(\alpha + s)}$$

Now let's plug in the transform for a unit step (1/s)

$$V_{out}(s) = V_{in}(s) \cdot \left(\frac{1}{1 + RCs} \right)$$

$$V_{out}(s) = V_{in}(s) \cdot \left(\frac{\left(\frac{1}{RC}\right)}{\left(\frac{1}{RC}\right) + s} \right)$$

$$V_{out}(s) = V_{in}(s) \cdot \left(\frac{\alpha}{\alpha + s} \right)$$

$$V_{out}(s) = \frac{\alpha}{s \cdot (\alpha + s)}$$

Signal Composition

- **Risetime and Bandwidth**

  - RC circuit cont…

Now we can transform into the time domain

$$V_{out}(s) = \frac{\alpha}{s \cdot (\alpha + s)}$$

$$V_{out}(t) = (1 - e^{-\frac{t}{RC}}) \cdot u(t)$$

Now we can plug in *u(t)*=1 for t>0

$$V_{out}(t) = (1 - e^{-\frac{t}{RC}})$$

This is the generic Time Domain solution for an exponential ramp (i.e., a rising edge)

Signal Composition

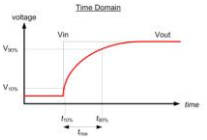
- **Risetime and Bandwidth**

  - RC circuit cont…

Let's solve for the risetime. First $t_{10\%}$

$$0.1 = (1 - e^{-\frac{t_{10\%}}{RC}})$$

$$t_{10\%} = (0.1) \cdot RC$$



Then $t_{90\%}$

$$0.9 = (1 - e^{-\frac{t_{90\%}}{RC}})$$

$$t_{90\%} = (2.3) \cdot RC$$

Finally, we solve for t_{rise}

$$t_{rise} = t_{90\%} - t_{10\%}$$

$$t_{rise} = (2.3) \cdot RC - (0.1) \cdot RC$$

$$t_{rise} = 2.2 \cdot RC$$

$$t_{rise} = 2.2 \cdot RC$$

Signal Composition

- **Risetime and Bandwidth**

  - Let's now solve for Bandwidth in the Frequency Domain

We go back to an earlier form of the RC solution:

$$V_{out}(s) = V_{in}(s) \cdot \left(\frac{1}{1 + RCs}\right)$$

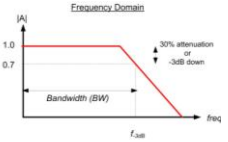
We are after the frequency at which the Magnitude of V_{out}/V_{in} is equal to 0.7.

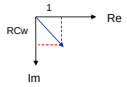
$$\left|\frac{V_{out}(s)}{V_{in}(s)}\right| = \left|\frac{1}{1 + RCs}\right|$$

In a logarithmic scale, this attenuation can also be described as where the Amplitude is 3dB less than the Amplitude at DC.

$$\frac{1}{\sqrt{2}} = \frac{1}{\sqrt{1^2 + (RC\omega_{-3dB})^2}}$$

We can use 1/√2 to make the math come out easier since (1/√2 = 0.7).





Signal Composition

- **Risetime and Bandwidth**

  - RC circuit cont….

$$\frac{1}{\sqrt{2}} = \frac{1}{\sqrt{1^2 + (RC\omega_{-3dB})^2}}$$

Now we rearrange to solve for f_{-3dB}

$$\sqrt{2} = \sqrt{1^2 + (RC\omega_{-3dB})^2}$$

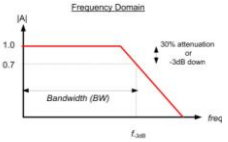
$$2 = 1 + (RC\omega_{-3dB})^2$$

$$1 = (RC\omega_{-3dB})^2$$

$$1 = RC\omega_{-3dB}$$

$$\omega_{-3dB} = \frac{1}{RC}$$

$$f_{-3dB} = \frac{1}{2\pi RC}$$



$$BW = \frac{1}{2\pi RC}$$

Signal Composition

- **Risetime Bandwidth Product**

  - We can relate the risetime to the bandwidth in what is known as the "Risetime Bandwidth Product"

$$t_{rise} = 2.2 \cdot RC \qquad BW = \frac{1}{2\pi RC}$$

$$RC = \frac{t_{rise}}{2.2} \qquad RC = \frac{1}{2\pi \cdot BW}$$

$$\frac{t_{rise}}{2.2} = \frac{1}{2\pi \cdot BW}$$

$$t_{rise} \cdot BW = \frac{2.2}{2\pi}$$

$$t_{rise} \cdot BW = 0.35$$

Signal Composition

- **Risetime Bandwidth Product**

  - We now have a quick expression to convert between key metrics in the Time and Frequency Domains for an RC circuit.

$$t_{rise} \cdot BW = 0.35$$

  - This not only tells us how much bandwidth a passive network has, but also how much Spectral Energy is stimulated by a driver that outputs an exponential approach (i.e., an RC step)

ex) how much spectral energy is created by a driver with a 1ns step?

$$t_{rise} \cdot BW = 0.35$$

$$(1ns) \cdot BW = 0.35$$

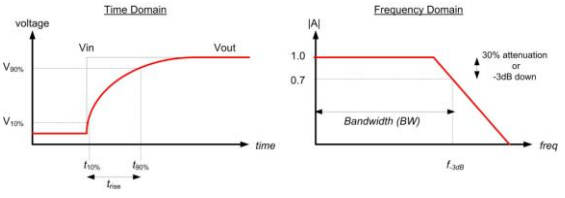
$$BW = \frac{0.35}{1ns} = 350MHz$$

Signal Composition

- **Risetime Bandwidth Product**

  - This expression also illustrates that as the risetime gets faster, higher frequencies are generated which makes sense…

$$t_{rise} \cdot BW = 0.35$$



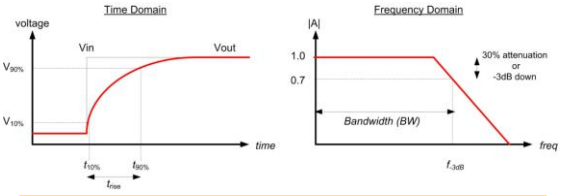
Signal Composition

- **System Risetime & Bandwidth**

  - We found a very powerful and quick way to convert between a Time Domain metric (risetime) and a Frequency Domain metric (BW) using the *Risetime Bandwidth Product*.

  - this was derived for a simple RC circuit, but we'll see it can be extended to a variety of step shapes.

$$t_{rise} \cdot BW = 0.35$$



Signal Composition

- **System Risetime & Bandwidth**

  - Note that this rule-of-thumb is used on a single element (i.e., a single driver, a single interconnect network, etc..)

  - Remember that BW is when the sine wave at f-3dB is reduced to 70% of its value at steady state.

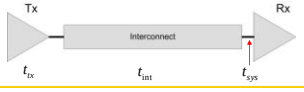
  - Now we look at how to combine multiple elements to derive the *system risetime* of cascaded blocks

  - We can combine multiple risetimes in a system to form the reduced equivalent system risetime using an RMS summation

$$t_{sys}^{\ 2} = t_{tx}^{\ 2} + t_{\text{int}}^{\ 2}$$

or

$$t_{sys} = \sqrt{t_{tx}^{\ 2} + t_{\text{int}}^{\ 2}}$$



Signal Composition

- **System Risetime & Bandwidth**

  - This expression can also be used on networks that have risetimes that aren't RC steps.

  - This expression can be used if the system follows the *Central Limit Theorem*

  - The *Central Limit Theorem* states:

    the sum of independent, random variables approaches a normal distribution with squared standard deviations equal to the sum of squares of individual standard deviations regardless of individual distributions as long as each element's distribution:

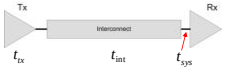
    - *follows a Gaussian distribution (i.e., what you get if you differentiate the step)*
    - *is symmetrical and uni-modal*
    - *is continuous*
    - *no one term dominates the expression*

Signal Composition

- **System Risetime & Bandwidth**

  - Let's see how we'd use this:

  ex) We have a digital link with a driver outputting a risetime of 300ps. The interconnect system looks like an RC network with bandwidth of 2GHz. What is the risetime we would expect to see at the receiver?



  - first convert the interconnect bandwidth to a risetime using the Risetime Bandwidth Product

$$t_{rise} \cdot BW = 0.35$$
$$t_{rise} \cdot (2GHz) = 0.35$$
$$t_{rise} = 175ps$$

  - now use the RMS sum to find the equivalent (or composite) risetime of the system.

$$t_{sys}^{\ 2} = t_{tx}^{\ 2} + t_{\text{int}}^{\ 2}$$
$$t_{sys} = \sqrt{(300ps)^2 + (175ps)^2}$$
$$t_{sys} = 347ps$$

Signal Composition

- **System Risetime & Bandwidth**

  - This can be expanded for as many elements that are in the system as long as they follow the *Central Limit Theorem.*

  - Since risetime and bandwidth are inversely proportional, we can also use this technique for Bandwidths directly.

$$\frac{1}{BW_{sys}^{\ 2}} = \frac{1}{BW_{tx}^{\ 2}} + \frac{1}{BW_{\text{int}}^{\ 2}}$$

Signal Composition

- **Square Waves**

  - Let's now look at the composition of a square wave.

  - We are interested in the Frequency Domain representation of square waves because this a digital driver will ultimately output a data pattern which will have a square wave spectrum.

  - We can visualize how much bandwidth a link will need to transmit a certain data pattern by looking at the data patterns spectrum.

  - For now, let's assume that the risetime is instantaneous (we'll incorporate real risetimes at the end)

Signal Composition

- **Time and Frequency Domains**

  Frequency Domain Basics

  - The Frequency Domain is completely made up by engineers and scientists. The only "Real" domain is the Time Domain.

  - We create the Frequency Domain in order to visualize problems in a more intuitive manner.

  - Also, some mathematical operations are easier in the Frequency Domain.

  - The Frequency Domain ***only contains Sine Waves***

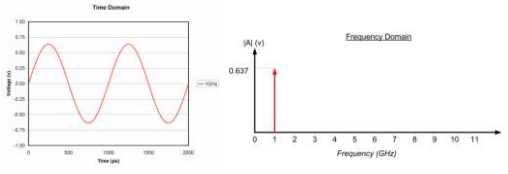
  - We represent a single Sine wave with a dot or line representing its Frequency and Amplitude

  - We can put phase on a 3rd axis, or on a different plot

Signal Composition

- **Time and Frequency Domains**

  - data in the Frequency Domain contains the same information as the Time Domain, but in *less space*



  - DC offset information for a waveform is represented in the Frequency Domain by placing magnitude information at ***0 Hz***

Signal Composition

- **Time and Frequency Domains**

  - Fourier Analysis tells us that any repetitive waveform in the Time Domain can be represented by one or more sine waves in the Frequency Domain.

  - When a waveform is created using more than one sine wave, we say it has a *Frequency Spectrum*

  - We can *Transform* back and forth between the domains using the *Fourier Transform*

  - The *Fourier Integral* will transform any repetitive function:

Time to Frequency Domain

$$X(f) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-i2\pi ft} dt$$

Frequency to Time Domain

$$x(t) = \int_{-\infty}^{+\infty} X(f) \cdot e^{i2\pi ft} df$$

Signal Composition

- **Time and Frequency Domains**

  Discrete Fourier Transform (DFT)

  - In the real world, we perform this transformation on discrete points of the function.

  - The number of points is fixed prior to the transform.

  - The pattern is assumed to repeat indefinitely every **T** seconds.

  - This is called a **Discrete Fourier Transform (DFT).**

  Fast Fourier Transform (FFT)

  - A special case of the DFT is when the number of points is a power of 2 (256, 512, 1024, …)

  - This special case enables the Matrix Math to be sped up considerably.

  - This is called a **Fast Fourier Transform (FFT)** and is the most common algorithm used today.

  - The FFT can be 100-10,000 times faster than a general DFT.

Signal Composition

- **Fourier Transform of Square Waves**

  - One way to construct the Fourier Transform of a Square Wave is to construct the Square Wave using multiple, basic functions.

  - The Fourier Transform not only allows functions to be transformed, but also operations.

  - This means we can construct a complex expression for a waveform, and then transform the individual parts and operations.

  - There are two common operations that we use when looking at Square Waves

  Fourier Transform of Operations

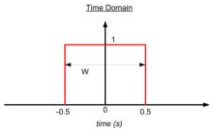
Operation	Time Domain	Frequency Domain
"Scaling"	$f(at)$	$\frac{1}{\lvert a \rvert} F\left(\frac{f}{a}\right)$
"Convolution"	$f(t) * g(t)$	$F(f) \cdot G(f)$

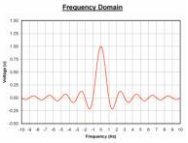
Signal Composition

- **Fourier Transform of Basic Functions**

  Rectangle Function $\Pi(t)$

  - A Rectangle Function is a pulse with unit width and height.
  - We can use this to represent a single pulse of a square wave with a finite pulse width (W).
  - This transforms into the *sinc* function with zero crossings at integer multiples of 1/W.





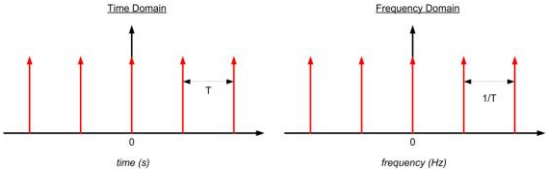
Function: $\Pi(t)$ $\quad\quad$ $\text{sinc}(f)$

Signal Composition

- **Fourier Transform of Basic Functions**

  Shaw Function $\text{Ш}(t)$

  - A Shaw Function is an infinite series of impulse functions.
  - This functions transforms into another Shaw function in the Frequency Domain



Functions: $\text{Ш}(t)$ $\quad\quad$ $\text{Ш}(f)$

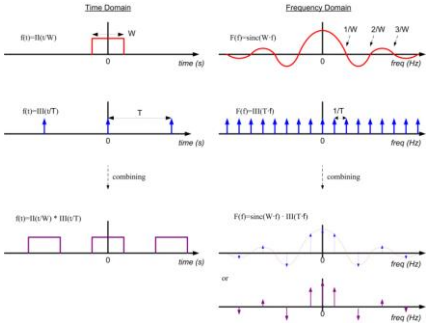
Signal Composition

- **Fourier Transform of Basic Functions**

  Square Wave

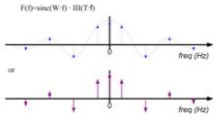
  - In the Time Domain, if we convolve a Rectangle Function with a Shaw Function, we will get a repetitive sequence of pulses.
  - If W is set to T/2, then we will have a 50% duty cycle square wave (a special case).

Signal Composition



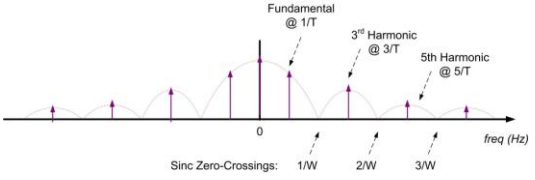
Signal Composition

- **Fourier Composition of a Square Wave**
  - Notice that when we have a 50% duty cycle, the *sinc* envelope eliminates all even harmonics.
  - We are left with only the DC offset, the Fundamental Frequency, and Odd Harmonics.
  - This is what most people are familiar with when they talk about the Fourier spectrum of a square wave.
  - However, this illustrates that if we alter the duty cycle, we **WILL** start getting even harmonics due to the *sinc* function spreading out.



Signal Composition

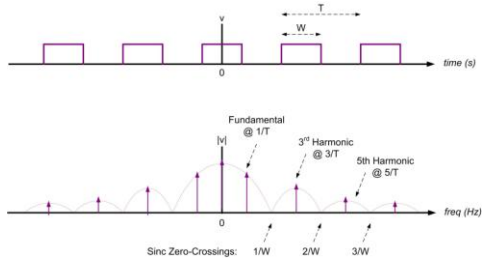
- **Fourier Composition of a Square Wave**
  - if we only concern ourselves with the Magnitude, we get the more familiar Fourier Spectrum.
  - Note that a smaller pulse increases the width of the *sinc* envelope
  - The amplitude of the spikes of energy at each multiple of the fundamental frequency depend the shape of the *sinc* envelope.



Signal Composition

- **Fourier Composition of a Square Wave**

  - We now have a transform for an Ideal Square Wave



Signal Composition

- **Fourier Composition of a Square Wave**

  - We now have two uses for the word *"Frequency"*.

  - We use *Frequency* to describe the rate at which the digital square wave toggles.

  - We also use Frequency to describe the harmonics that make up the square wave.

  - To distinguish the two, we typically use the term *Toggle Rate or Toggle Frequency* for the time domain square wave.

$$F_{toggle} = \frac{1}{T_{square\_wave}} = \frac{1}{T_{toggle}}$$

  - But we still sometimes use the word *frequency* when we discuss digital signals, so we always need to be aware of the difference between *Toggle Frequency* and the frequency of the Square Wave's *Spectral Content.*

Signal Composition

- **Fourier Composition of a Square Wave**

  - What the Frequency Domain shows us is that the Square wave is made up of a series of sine waves with different amplitudes and frequencies that are at harmonic multiples of the fundamental.

  - An ideal square wave is made up of only the ODD harmonics of the fundamental frequency.

  - A square wave with less than 50% duty cycle begins to include EVEN harmonics due to the *sinc* envelope expanding.

  - The amplitude of each harmonic is given by:

$$A_n = \frac{2}{\pi \cdot n}$$

where *n* is the harmonic number (i.e., 1, 3, 5, 7, etc…)

Signal Composition

- **Fourier Composition of a Square Wave**

  - An ideal square wave will have amplitudes that get smaller and smaller as the harmonic frequency goes up.

$$A_n = \frac{2}{\pi \cdot n}$$

Ex) a 1v square wave (V_{LOW}=-0.5v, V_{HIGH}=+0.5v) will be made up of sine waves with Amplitudes:

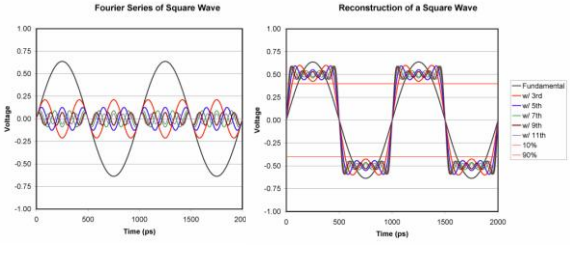
	Amplitude
Fundamental	0.637 v
3rd	0.212 v
5th	0.127 v
7th	0.091 v
9th	0.071 v
11th	0.058 v

  - notice that the higher the frequency, the less amplitude that the harmonic contributes to the reconstruction of the Square Wave.

Signal Composition

- **Fourier Composition of a Square Wave**

  - if we split out the sine waves in the Time Domain, we can see the individual components:

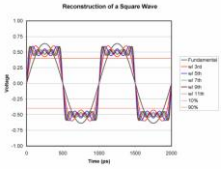


Signal Composition

- **Fourier Composition of a Square Wave**

  - Notice that as we add more harmonics to the fundamental, we get a waveform that looks more and more like an ideal square wave.

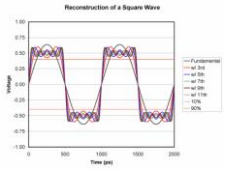
  - If we add all the harmonics (up to infinity), we will get a perfect square wave with instantaneous risetimes.



Signal Composition

- **Fourier Composition of a Square Wave**
  - However, we know that no system can output or transmit infinite frequency.
  - The big question for digital system designers becomes…

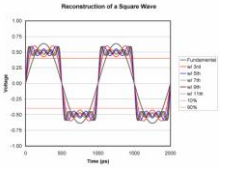
    "How many harmonics do I need to get a square wave that is good enough?



Signal Composition

- **Fourier Composition of a Square Wave**
  - This is an arbitrary question and depends on how fast of a risetime each system needs.
  - But, we can tie this question back to one of our rules-of-thumbs relating the risetime to the period of the square wave.
  - earlier we stated that the risetime should be ~10% of the period

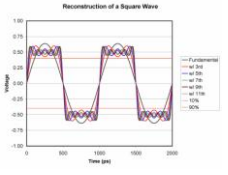
$$t_{rise} \le (0.1) \cdot T_{toggle}$$



Signal Composition

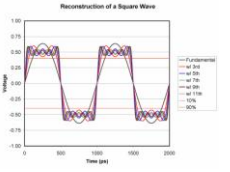
- **Fourier Composition of a Square Wave**
  - Let's start with only the fundamental frequency and see what percentage of the period that the risetime takes.
  - Then we'll add harmonics and see how it changes: $\%\_of\_T_{toggle} = \left(\frac{t_{rise}}{T_{toggle}}\right) x100$

Square Wave Composition	(t_{rise}/T_{period})%
Fundamental	*22%*
Fund + 3rd	*11%*
Fund + 3rd + 5th	*7.4% ***
Fund + 3rd + 5th + 7th	*5.7%*
Fund + 3rd + 5th +7th + 9th	*4.4%*
Fund + 3rd + 5th +7th + 9th + 11th	*3.6%*



Signal Composition

- **Fourier Composition of a Square Wave**
  - This tells us that if we include the fundamental + 3rd harmonic, the square wave risetime is 11% of the period.
  - If we include the 5th, this goes down to 7.4%, which meets our objective.
  - So we can say that we really should try to get the 5th harmonic of the square wave in order to reconstruct a reasonable representation.



Signal Composition

- **Fourier Composition of a Square Wave**
  - How does this all compare to the *Risetime Bandwidth Product?*
  - Remember our original expression was derived for a single-pole, RC circuit $t_{rise} \cdot BW = 0.35$
  - If we look at the risetime bandwidth product of a Square Wave made up of a Fourier Series of sine waves, we get:

Spectral Content	Risetime Bandwidth Product
Fundamental	*0.21*
Fund + 3rd	*0.33*
Fund + 3rd + 5th	*0.37 ****
Fund + 3rd + 5th + 7th	*0.4*
Fund + 3rd + 5th +7th + 9th	*0.4*
Fund + 3rd + 5th +7th + 9th + 11th	*0.4*

NOTE: The BW we use is the frequency of the highest harmonic present in the spectrum of the square wave.

  - the product is pretty close to 0.35, especially if we try to include up to the 5th harmonic.

Signal Composition

- **Fourier Composition of a Square Wave**
  - This also shows that for different shaped risetimes, we can simply change the risetime BW product to get more accurate results
    - RC step, use 0.35
    - Gaussian step, use 0.4
  - BUT REMEMBER, this is an approximation to get a *gut-feel*. So we don't need to get too hung up on the exact number as long as it is around 0.35.
  - One more nice thing is that the Fourier Series Representation has a Gaussian Distribution by nature, which means it follows the Central Limit Theorem.
  - That means that even if we choose to use 0.4 in our risetime BW product, we can still use a sum-of-squares expression to describe the composite risetime.

Signal Composition

- **Fourier Composition of a Square Wave**
  - Once again, remember that we are talking about the stimulated energy of the driver.
  - What gets to the Receiver in our digital link is another story.
  - At first glance, we think this is pretty straight forward because an interconnect system will simply attenuate the forcing function's harmonics.
  - In reality, when we look at the distributed nature of an interconnect system, we see that reflections can actually cause the harmonics to be *amplified*!!!
  - this makes the analysis a little more interesting.
  - Next, we look how to model the interconnect in order to understand how its response will effect the wave shape of the forcing function.