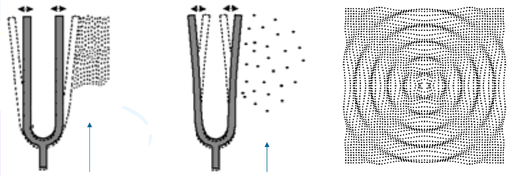
What is Sound?

As the tines move back and forth they exert pressure on the air around them.
(a) The first displacement of the tine compresses the air molecules causing high pressure.
(b) Equal displacement of the tine in the opposite direction forces the molecules to widely disperse themselves and so, causes low pressure.
(c) These rapid variations in pressure over time form a pattern which propogates itself through the air as a wave. Points of high and low pressure are sometimes reffered to as ’**compression’** and ’**rarefaction**’ respectively.



(a) compression

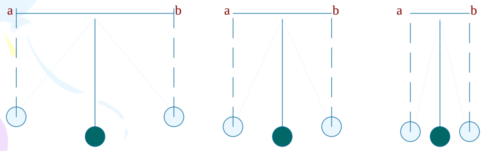
(b) rarefaction

(c) wave propegation of a tuning fork as seen from above

Simple Harmonic Motion -- a Pendulum

- When a pendulum approaches equlibrium it doesn’t slow down; it simply travels a smaller distance from the point of rest.
- Any body undergoing simple harmonic motion moves periodically with uniform speed.
- If the tuning fork is moving periodically then the pressure variations it creates will also be periodic.

The time taken to get from position a to b in all three cases is the same



Maximum displacement at 0 seconds

Maximum displacement after say, 3 seconds

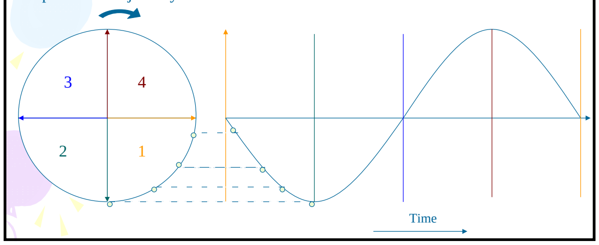
Maximum displacement after say, 6 seconds

The Unit Circle

These pressure patterns can be represented using as a circle.
Imagine the journey of the pendulum or the tine in four stages:

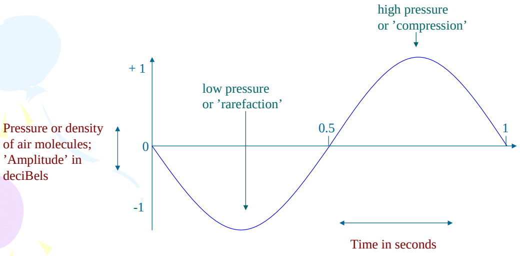
1) from its point of rest to its first point of maximum displacement...
2) its first point of maximum displacement back through the point of rest...
3) ... to its second point of maximum displacement...
4) ... and back from there through its point of rest again

We can map that journey to a circle. This is called the **Unit Circle**. The **sine wave** represents this journey around and around the unit circle *over time*.



Sine Waves

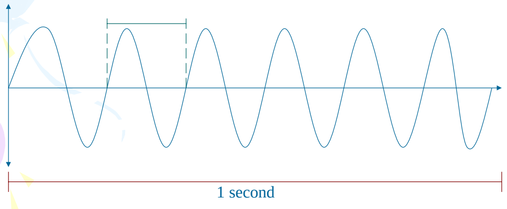
The **sine wave** or **sinusoid** or **sinusoidal signal** is probably the most commonly used graphic representation of sound waves. The diagram below shows one cycle or ’period’ of a wave, i.e., the build-up from equilibrium to maximum high pressure, to maximum low pressure, to equilibrium again.



Sine Waves

The specific properties of a sine wave are described as follows.

Frequency = the number of cycles per second (this wave has a frequency of 6 hertz)
Amplitude = variations in air pressure (measured in decibels)
Wavelength = physical length of 1 period of a wave (measured in metres per second)
= The starting point of a wave along the y-axis (measured in degrees)



1 second

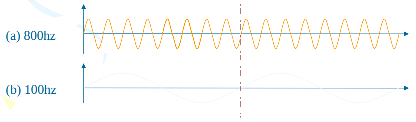
Frequency

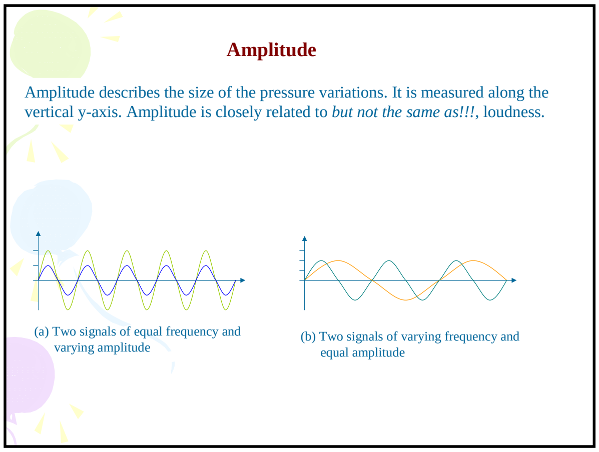
Frequency refers to the number of cycles of a wave per second. This is measured in **Hertz**. So if a sinusoid has a frequency of 100hz then one period of that wave repeats itself every 1/100th of a second. Humans can hear frequencies between 20hz and 20,000hz (20Khz).

1) Frequency is closely related to, *but not the same as!!!*, pitch.

2) Frequency does not determine the speed a wave travels at. Sound waves travel at approximately 340metres/second regardless of frequency.

3) Frequency is inherent to, and determined by the vibrating body – not the amount of energy used to set that body vibrating. For example, the tuning fork emits the same frequency regardless of how hard we strike it.



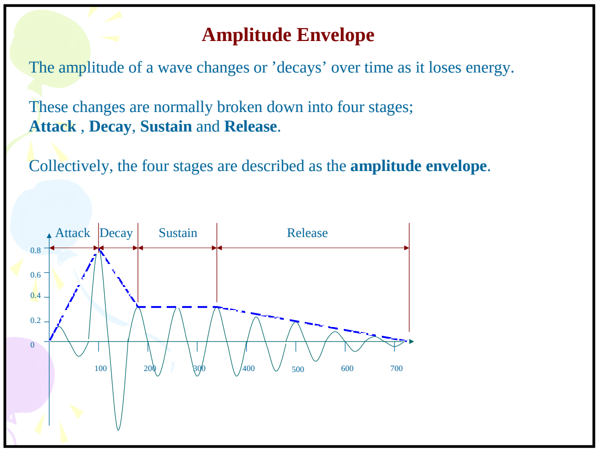


Amplitude

Amplitude describes the size of the pressure variations. It is measured along the vertical y-axis. Amplitude is closely related to *but not the same as!!!*, loudness.

(a) Two signals of equal frequency and varying amplitude

(b) Two signals of varying frequency and equal amplitude



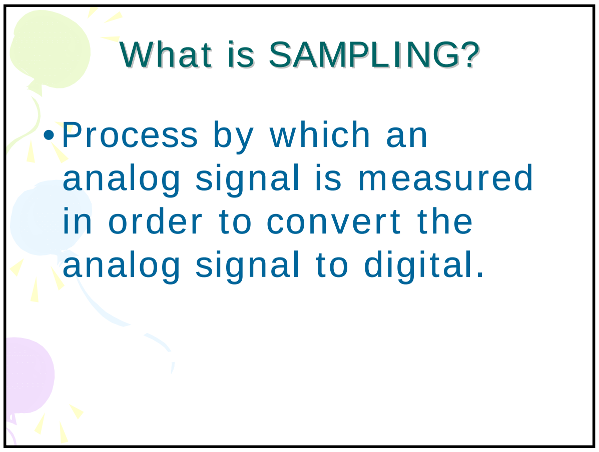
Amplitude Envelope

The amplitude of a wave changes or ’decays’ over time as it loses energy.

These changes are normally broken down into four stages;
Attack , **Decay**, **Sustain** and **Release**.

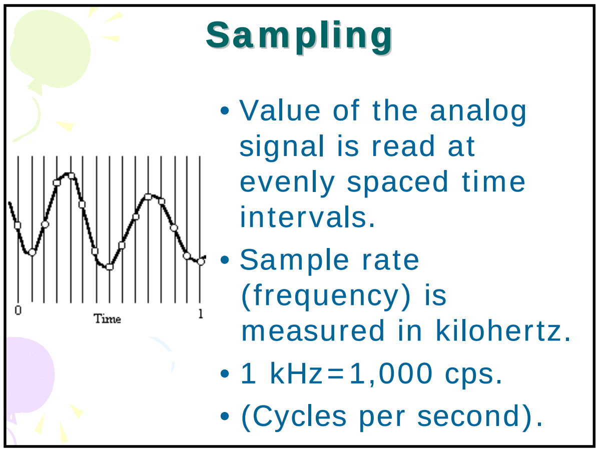
Collectively, the four stages are described as the **amplitude envelope**.

Introduction to Sampling



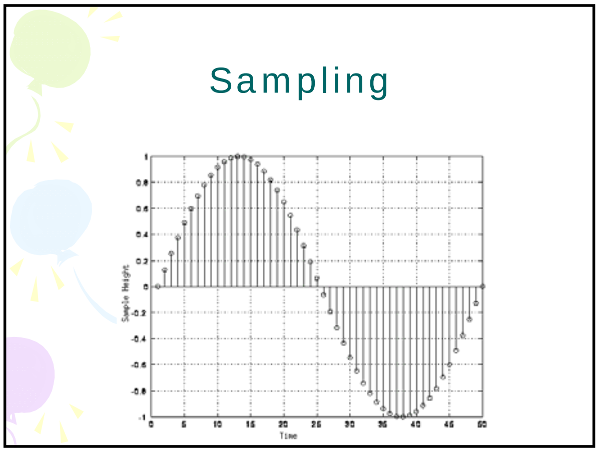
What is SAMPLING?

- Process by which an analog signal is measured in order to convert the analog signal to digital.

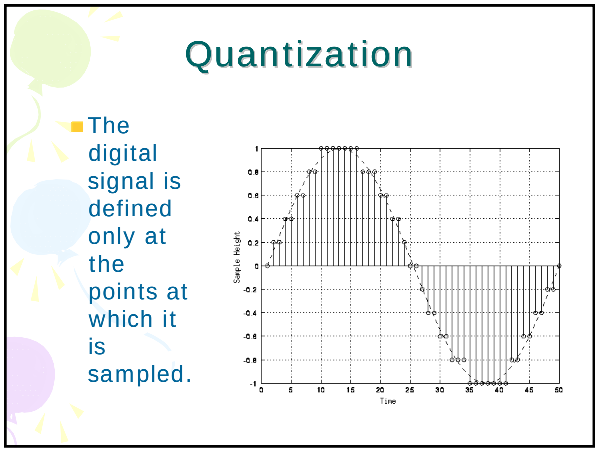


Sampling

- Value of the analog signal is read at evenly spaced time intervals.
- Sample rate (frequency) is measured in kilohertz.
- 1 kHz=1,000 cps.
- (Cycles per second).

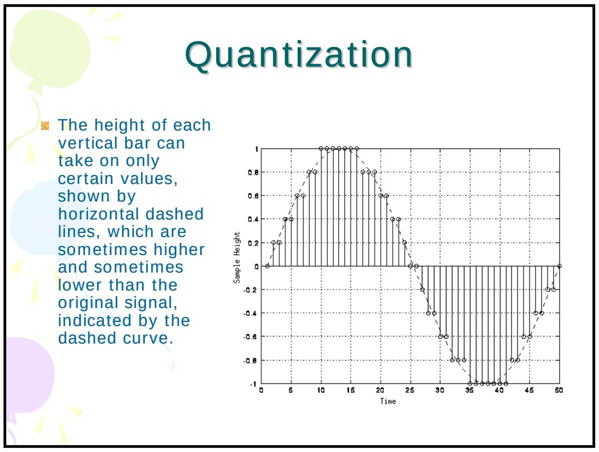


Sampling



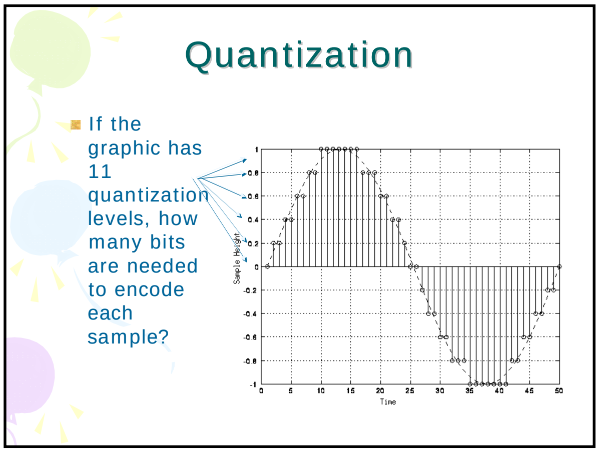
Quantization

- The digital signal is defined only at the points at which it is sampled.



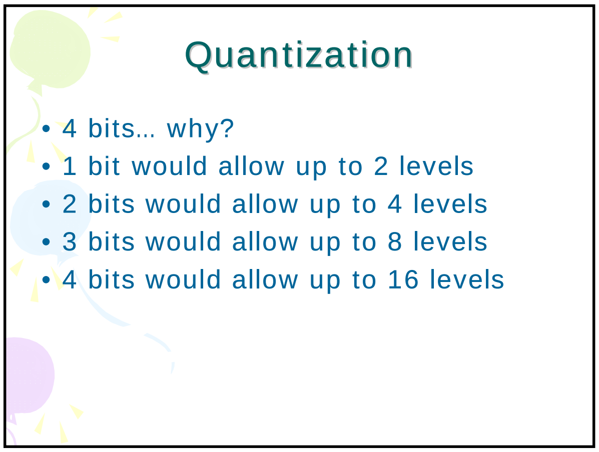
Quantization

- The height of each vertical bar can take on only certain values, shown by horizontal dashed lines, which are sometimes higher and sometimes lower than the original signal, indicated by the dashed curve.



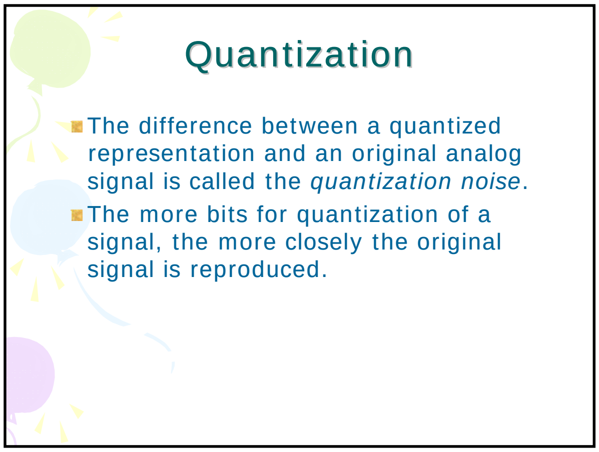
Quantization

- If the graphic has 11 quantization levels, how many bits are needed to encode each sample?



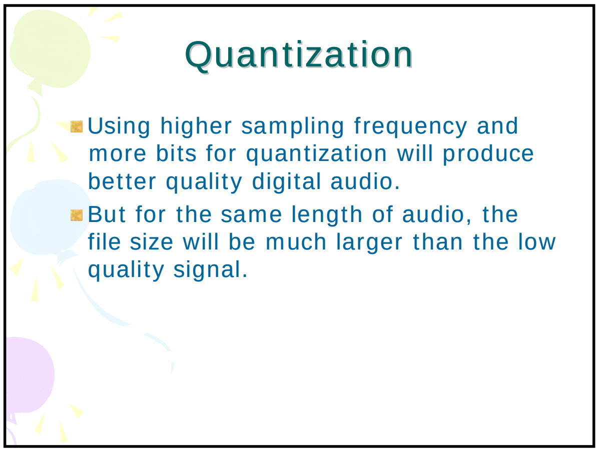
Quantization

- 4 bits… why?
- 1 bit would allow up to 2 levels
- 2 bits would allow up to 4 levels
- 3 bits would allow up to 8 levels
- 4 bits would allow up to 16 levels



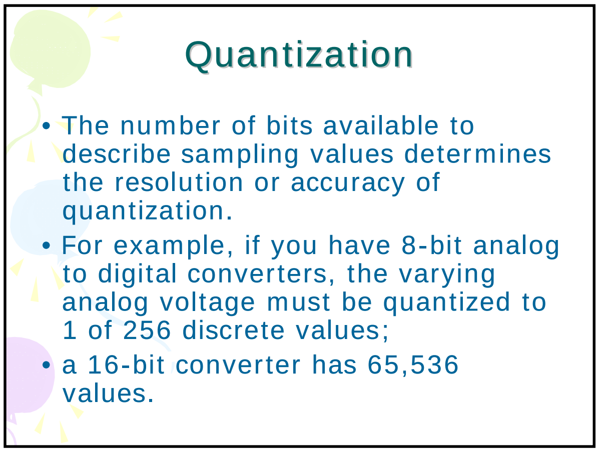
Quantization

- The difference between a quantized representation and an original analog signal is called the *quantization noise*.
- The more bits for quantization of a signal, the more closely the original signal is reproduced.



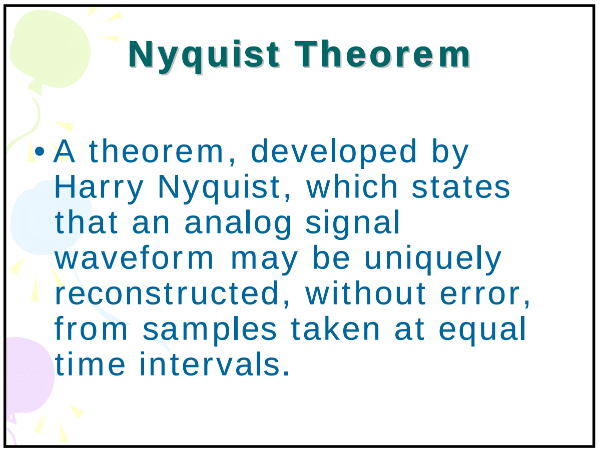
Quantization

- Using higher sampling frequency and more bits for quantization will produce better quality digital audio.
- But for the same length of audio, the file size will be much larger than the low quality signal.



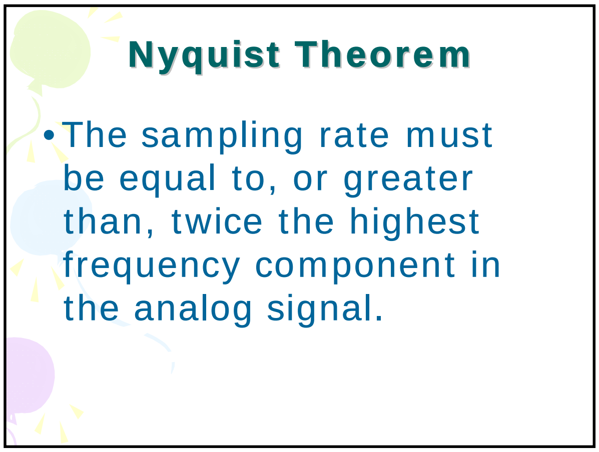
Quantization

- The number of bits available to describe sampling values determines the resolution or accuracy of quantization.
- For example, if you have 8-bit analog to digital converters, the varying analog voltage must be quantized to 1 of 256 discrete values;
- a 16-bit converter has 65,536 values.



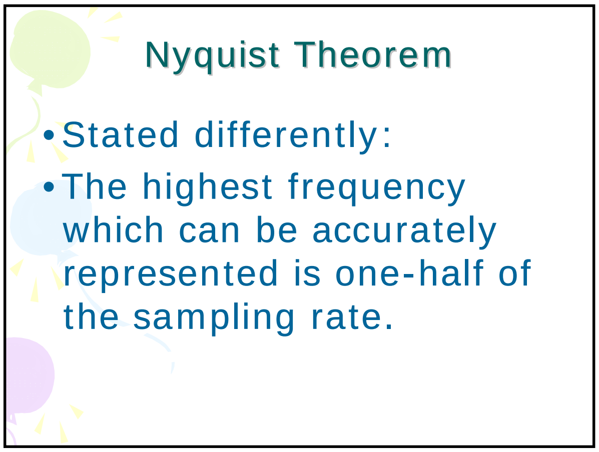
Nyquist Theorem

- A theorem, developed by Harry Nyquist, which states that an analog signal waveform may be uniquely reconstructed, without error, from samples taken at equal time intervals.



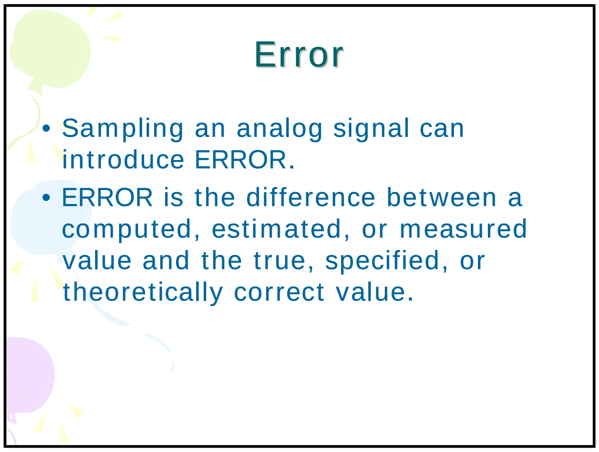
Nyquist Theorem

- The sampling rate must be equal to, or greater than, twice the highest frequency component in the analog signal.



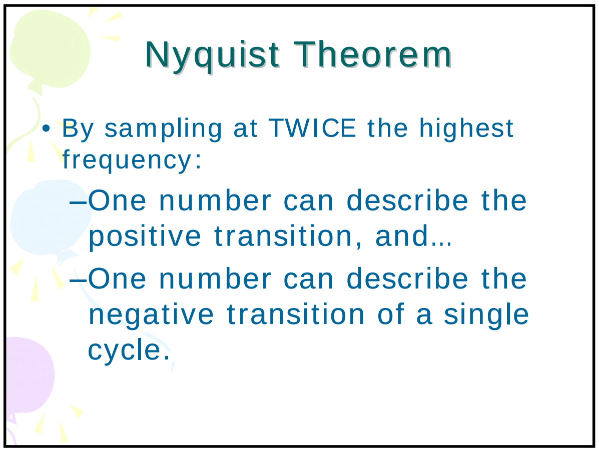
Nyquist Theorem

- Stated differently:
- The highest frequency which can be accurately represented is one-half of the sampling rate.



Error

- Sampling an analog signal can introduce ERROR.
- ERROR is the difference between a computed, estimated, or measured value and the true, specified, or theoretically correct value.



Nyquist Theorem

- By sampling at TWICE the highest frequency:
  - –One number can describe the positive transition, and…
  - –One number can describe the negative transition of a single cycle.

upper => sampling 6 times per cycle(fs=6f);
middle => sampling 3 times per cycle(fs=3f);

lower=> sampling 6 times in 5 cycles, from[1]

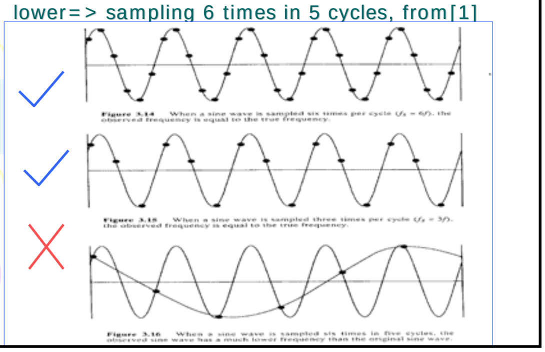


Figure 3.14 When a sine wave is sampled six times per cycle ($f_s = 6f$), the observed frequency is equal to the true frequency.

Figure 3.15 When a sine wave is sampled three times per cycle ($f_s = 3f$), the observed frequency is equal to the true frequency.

Figure 3.16 When a sine wave is sampled six times in five cycles, the observed sine wave has a much lower frequency than the original sine wave.

Audio Synthesis Basics

Analog Synthesis
Intro to Digital Oscillators

Analog Synthesis Overview

- Sound is created by controlling electrical current within synthesizer, and amplifying result.
- Basic components:
  - Oscillators
  - Filters
  - Envelope generators
  - Noise generators
- Voltage control

Oscillators

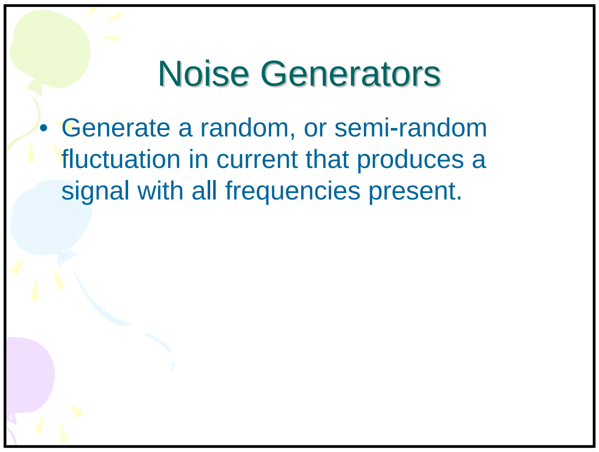
- Creates periodic fluctuations in current, usually with selectable waveform.
- Different waveforms have different harmonic content, or *frequency spectra*.

Filters

- Given an input signal, attenuate or boost a frequency range to produce an output signal
- Basic Types:
  - Low pass
  - High pass
  - Band pass
  - Band reject (notch)

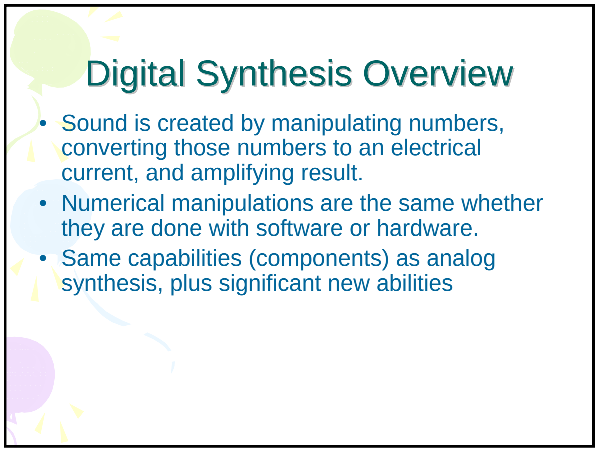
Envelope Generators

- Generate a control function that can be applied to various synthesis parameters, including amplitude, pitch, and filter controls.



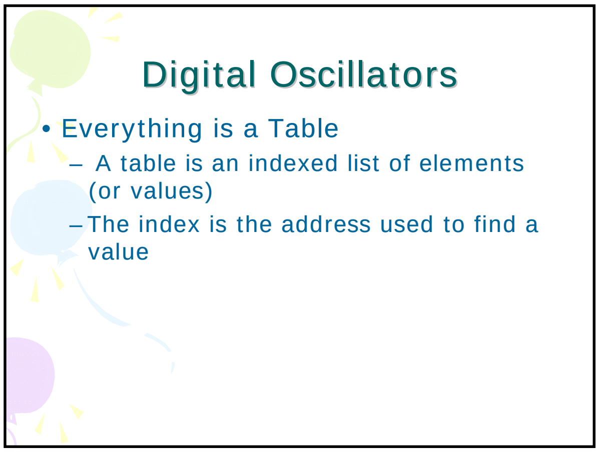
Noise Generators

- Generate a random, or semi-random fluctuation in current that produces a signal with all frequencies present.



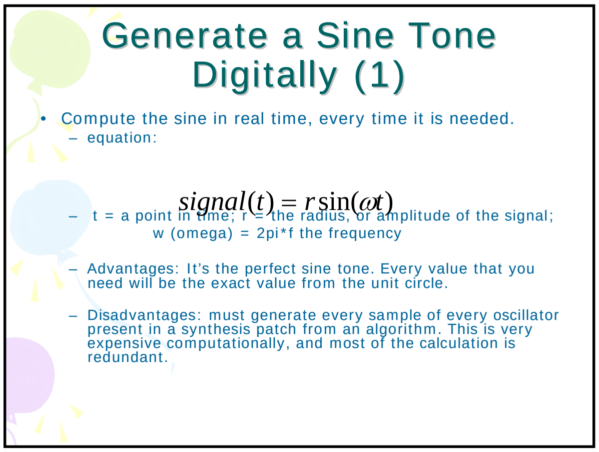
Digital Synthesis Overview

- Sound is created by manipulating numbers, converting those numbers to an electrical current, and amplifying result.
- Numerical manipulations are the same whether they are done with software or hardware.
- Same capabilities (components) as analog synthesis, plus significant new abilities



Digital Oscillators

- Everything is a Table
  - A table is an indexed list of elements (or values)
  - The index is the address used to find a value

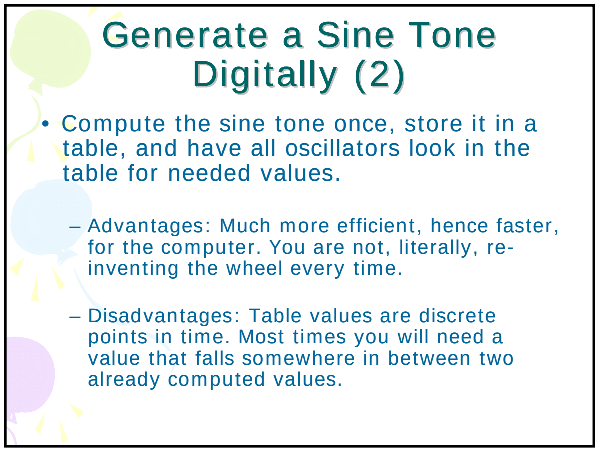


Generate a Sine Tone Digitally (1)

- Compute the sine in real time, every time it is needed.
  - equation:

$$signal(t) = r\sin(\omega t)$$

  - t = a point in time; r = the radius, or amplitude of the signal; w (omega) = 2pi*f the frequency
  - Advantages: It's the perfect sine tone. Every value that you need will be the exact value from the unit circle.
  - Disadvantages: must generate every sample of every oscillator present in a synthesis patch from an algorithm. This is very expensive computationally, and most of the calculation is redundant.



Generate a Sine Tone Digitally (2)

- Compute the sine tone once, store it in a table, and have all oscillators look in the table for needed values.
  - Advantages: Much more efficient, hence faster, for the computer. You are not, literally, re-inventing the wheel every time.
  - Disadvantages: Table values are discrete points in time. Most times you will need a value that falls somewhere in between two already computed values.

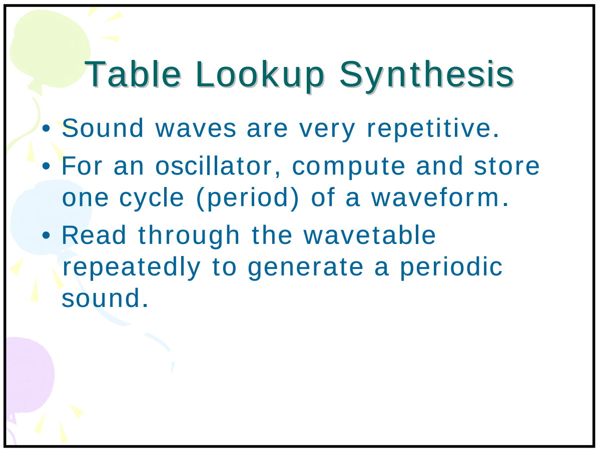


Table Lookup Synthesis

- Sound waves are very repetitive.
- For an oscillator, compute and store one cycle (period) of a waveform.
- Read through the wavetable repeatedly to generate a periodic sound.

Changing Frequency

- The Sample Rate doesn't change within a synthesis algorithm.
- You can change the speed that the table is scanned by skipping samples.
- skip size is the increment, better known as the phase increment.

  phase increment is a very important concept

Algorithm for a Digital Oscillator

- Basic, two-step program:
  - $phase\_index = \mathrm{mod}_L(previous\_phase + increment)$
  - $output = amplitude \times wavetable[phase\_index]$
- $increment = \dfrac{TableLength \times DesiredFrequency}{SampleRate}$

If You're Wrong, it's Noise

- What happens when the phase increment doesn't land exactly at an index location in the table?
  - It simply looks at the last index location passed for a value.
    In other words, the phase increment is truncated to the integer.
- Quantization
- Noise
- The greater the error, the more the noise.

Interpolation

- Rather than truncate the phase location…
  - look at the values stored before and after the calculated phase location
  - calculate what the value would have been at the calculated phase location if it had been generated and stored.

  Interpolate

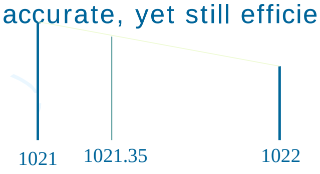
- More calculations, but a much cleaner signal.

Linear interpolation

- Interpolate between two audio samples

```
double inbetween = fmod(sample, 1);
return (1. – inbetween) * wave[int(sample)] +
        inbetween * wave[int(sample) + 1];
```

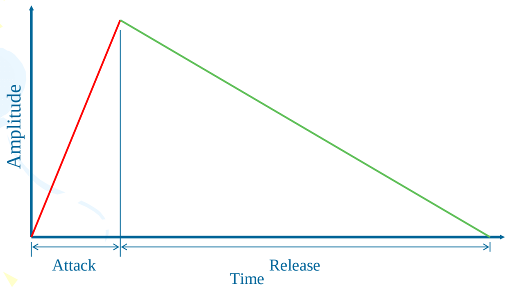
- More accurate, yet still efficient



Envelopes

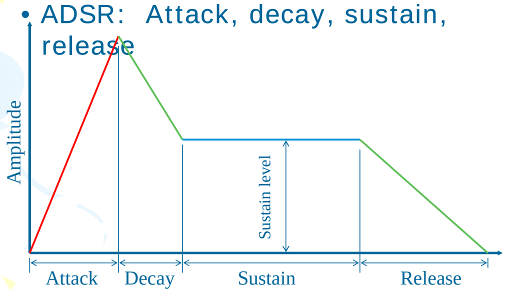
- We commonly will make samples with fixed amplitudes, then make a synthetic *envelope* for the sound event.

Attack and Release

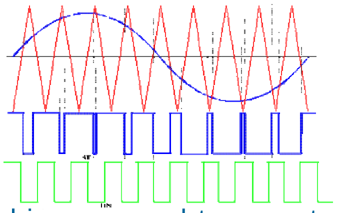


ADSR

- ADSR: Attack, decay, sustain, release

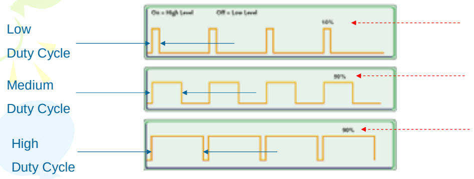


PWM: Pulse Width Modulation



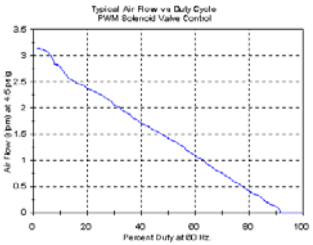
- Signal is compared to a sawtooth wave producing a pulse width proportional to amplitude

What Can Be Done With PWM?



- Question: What happens if voltages like the ones above are connected to an LED?
- Answer: The longer the duty cycle, the longer the LED is on and the brighter the light.

What Can Be Done With PWM?



- Average power can be controlled
- Average flows can also be controlled by fully opening and closing a valve with some duty cycle

PULSE WIDTH MODULATION

- Pulse Width Modulation (PWM) involves the generation of a series of pulses at a fixed period and frequency.
- The duty cycle defines the width of each pulse which is varied to generate waveforms.

PULSE WIDTH MODULATION

- A simple low pass filter is then used to generate an output voltage directly proportional to the average time spent in the HIGH state.
- (i.e., 50% duty cycle is equal to 2.5 volts when VDD =5.0V).

RC low pass filter (integrator)

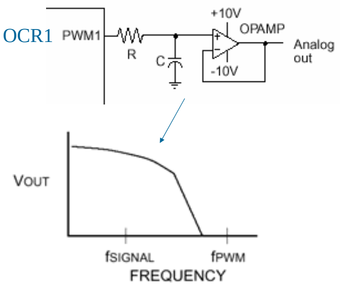
Choosing the -3 dB point at 4 kHz, and using the Relation:

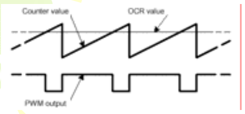
RC = 1/(2 · pi ·f)

we get R = 4 k, if C is chosen as 0.01 mF:

R = 4.0 k

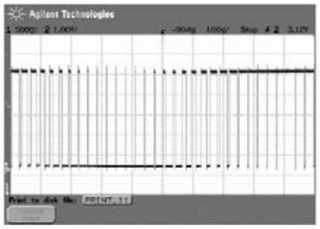
C = 0.01 mF



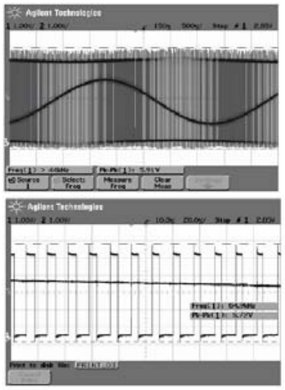


AVR PWM

- AVR counter/timer 1
- To change the PWM base frequency, the timer clock frequency and top counter value is changed.
- Faster clock and/or lower top value will increase the PWM base frequency, or timer overflow frequency.
- With full resolution (top value 255) the maximum PWM base frequency is 250 kHz.
- Altering the value of the Output Compare Registers (OCR) changes the duty cycle.
- Increasing the OCR value increases the duty cycle. The PWM output is high until the OCR value is reached, and low until the timer reaches the top value and wraps back to 0



Filtered and unfiltered PWM Outputs

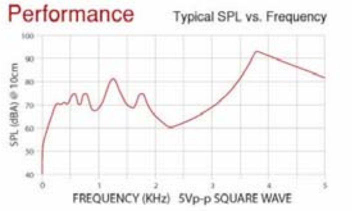


Algorithm for a Digital Oscillator

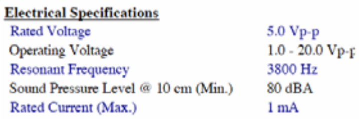
- Basic, two-step program:
  - *phase_index* = mod$_L$(*previous_phase* + *increment*)
  - *output* = *amplitude* x *wavetable*[*phase_index*]
- *increment* = $\dfrac{\textit{TableLength} \times \textit{DesiredFrequency}}{\textit{SampleRate}}$

The Flock-- Birdie Audio:

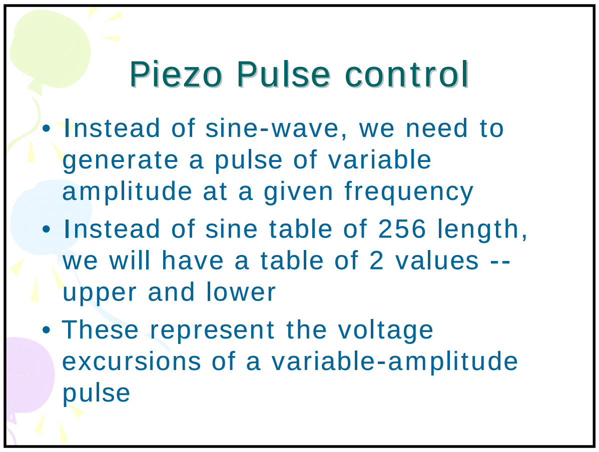
PWM Output: →



Piezo-electric Transducer

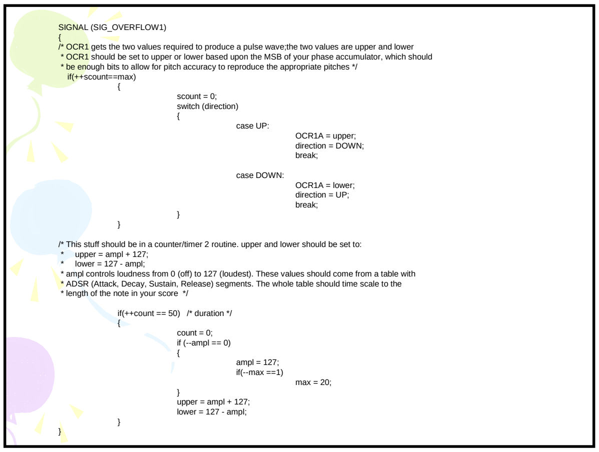


Electrical Specifications	
Rated Voltage	5.0 Vp-p
Operating Voltage	1.0 - 20.0 Vp-p
Resonant Frequency	3800 Hz
Sound Pressure Level @ 10 cm (Min.)	80 dBA
Rated Current (Max.)	1 mA



Piezo Pulse control

- Instead of sine-wave, we need to generate a pulse of variable amplitude at a given frequency
- Instead of sine table of 256 length, we will have a table of 2 values -- upper and lower
- These represent the voltage excursions of a variable-amplitude pulse



```
SIGNAL (SIG_OVERFLOW1)
{
/* OCR1 gets the two values required to produce a pulse wave;the two values are upper and lower
 * OCR1 should be set to upper or lower based upon the MSB of your phase accumulator, which should
 * be enough bits to allow for pitch accuracy to reproduce the appropriate pitches */
   if(++scount==max)
        {
                scount = 0;
                switch (direction)
                {
                        case UP:
                                OCR1A = upper;
                                direction = DOWN;
                                break;

                        case DOWN:
                                OCR1A = lower;
                                direction = UP;
                                break;
                }
        }

/* This stuff should be in a counter/timer 2 routine. upper and lower should be set to:
 *    upper = ampl + 127;
 *    lower = 127 - ampl;
 * ampl controls loudness from 0 (off) to 127 (loudest). These values should come from a table with
 * ADSR (Attack, Decay, Sustain, Release) segments. The whole table should time scale to the
 * length of the note in your score  */

        if(++count == 50)   /* duration */
        {
                count = 0;
                if (--ampl == 0)
                {
                        ampl = 127;
                        if(--max ==1)
                                max = 20;
                }
                upper = ampl + 127;
                lower = 127 - ampl;
        }
}
```