

Multimedia System

Course Code: CACS457 | Year/Semester: IV / VIII

Unit 2: Sound / Audio System

(5 Hours)

Sound is one of the five basic multimedia building blocks introduced in Unit 1. This unit looks closely at how sound is captured, digitized, stored, analyzed, transmitted, and synthesized by computers, covering both natural recorded audio and computer-generated music through MIDI.

1. Overview of Sound System

Sound is a form of energy produced when an object vibrates, creating pressure waves that travel through a medium such as air. When these pressure waves reach our ears, the eardrum vibrates and the brain interprets this as sound. In plain words: sound is simply 'vibrating air' that our ears can detect.

A sound wave is described by three main physical properties that determine what we hear:

Property	Plain Meaning	What We Perceive
Frequency	Number of vibrations (cycles) per second, measured in Hertz (Hz)	Pitch — higher frequency sounds higher
Amplitude	The size/strength of the pressure wave	Loudness/volume — bigger amplitude sounds louder
Waveform / Timbre	The exact shape of the repeating wave pattern	Quality/character of sound — why a violin and a flute playing the same note still sound different

A basic computer sound/audio system is the set of hardware and software that lets a computer capture real-world sound, convert it into digital form, store/process it, and play it back. It typically has four parts: an input transducer (microphone) that turns sound into an electrical signal, an Analog-to-Digital Converter (ADC) that turns the electrical signal into digital data, software/storage that holds and processes this digital data, and a Digital-to-Analog Converter (DAC) with a speaker that turns the digital data back into audible sound.

Basic Audio System: Analog → Digital → Analog Pipeline

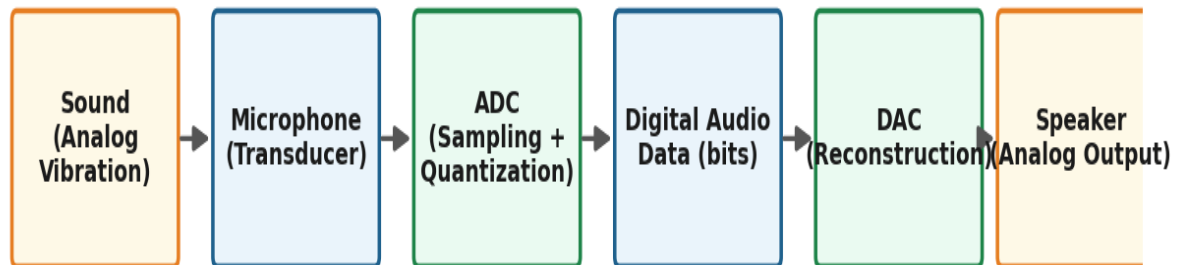


Figure 1: A basic audio system converts analog sound to digital data and back to analog sound.

Note: The human ear can typically perceive frequencies from about 20 Hz to 20,000 Hz (20 kHz) — this range is called the audible frequency range, and it directly affects how digital audio is sampled, as shown in Section 2.

2. Producing Digital Audio

Producing digital audio means converting a continuous (analog) sound wave into a sequence of numbers that a computer can store and process. This conversion happens in two key steps: sampling and quantization.

Sub-step (a): Sampling

Sampling means measuring (reading) the amplitude of the analog signal at regular time intervals. The number of measurements taken per second is called the sampling rate (or sampling frequency), measured in Hertz (Hz) or kHz. For example, audio CDs use a sampling rate of 44,100 Hz (44.1 kHz), meaning the signal is measured 44,100 times every second.

Sub-step (b): Quantization

Quantization means rounding each sampled amplitude value to the nearest available digital level. The number of levels available depends on the bit depth (number of bits used per sample) — e.g. 16-bit audio has $2^{16} = 65,536$ possible amplitude levels. More bits give a more accurate, less 'noisy' representation of the original sound, but use more storage.

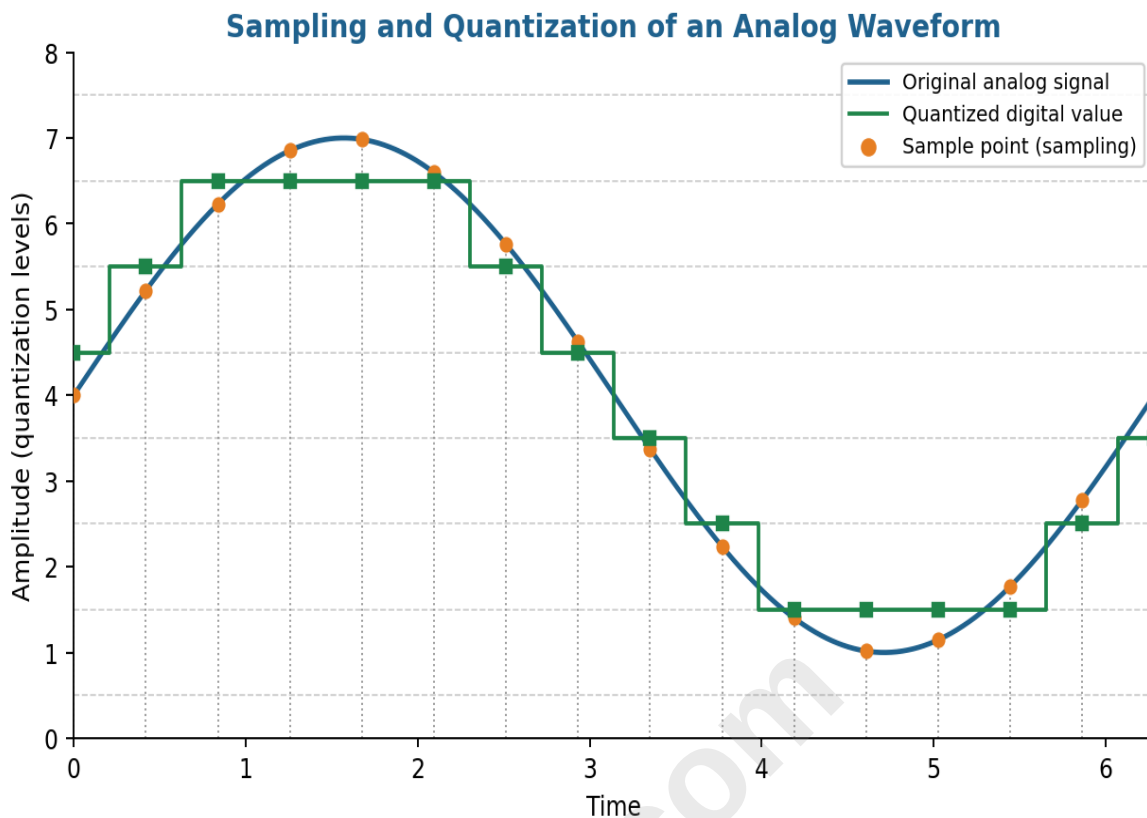


Figure 2: Sampling (when we measure) and quantization (rounding to the nearest digital level) together turn an analog wave into digital data.

The Nyquist Sampling Theorem

A fundamental rule called the Nyquist theorem states that to accurately represent a signal digitally, the sampling rate must be at least twice the highest frequency present in the signal. This is why audio CDs sample at 44.1 kHz — slightly more than double the 20 kHz upper limit of human hearing, leaving margin for filtering. Sampling below this rate causes a distortion called aliasing, where high frequencies are wrongly recorded as lower, false frequencies.

Example 1: Applying the Nyquist Theorem

Question: A particular musical instrument produces sound frequencies up to 8 kHz. What is the minimum sampling rate required to digitize this sound accurately without aliasing?

Solution / Explanation:

By the Nyquist theorem, minimum sampling rate = $2 \times$ highest frequency.

Minimum sampling rate = $2 \times 8,000 \text{ Hz} = 16,000 \text{ Hz}$ (16 kHz).

Any sampling rate of 16 kHz or higher will correctly capture this signal; sampling below 16 kHz risks aliasing distortion.

Example 2: Calculating Digital Audio File Size

Question: A stereo (2-channel) audio recording is made at a sampling rate of 44,100 Hz, using a bit depth of 16 bits, for a duration of 30 seconds. Calculate the uncompressed file size in megabytes (MB).

Solution / Explanation:

Formula: File size (bits) = Sampling rate × Bit depth × Number of channels × Duration (seconds).

File size = $44,100 \times 16 \times 2 \times 30 = 42,336,000$ bits.

Convert to bytes: $42,336,000 \div 8 = 5,292,000$ bytes.

Convert to MB: $5,292,000 \div (1024 \times 1024) \approx 5.05$ MB.

This shows why uncompressed digital audio needs compression (Unit 4) for practical storage and streaming.

3. Music and Speech

Although both music and speech are 'audio', they have very different characteristics, and multimedia systems often process them differently.

Aspect	Music	Speech
Source	Multiple instruments/voices, often simultaneous	Single human vocal tract
Frequency range	Wide — roughly 20 Hz to 20 kHz	Narrower — mainly 300 Hz to 3.4 kHz carries most intelligible content
Structure	Often periodic/rhythmic, harmonic patterns	Sequence of phonemes forming words and sentences
Typical processing goal	Preserve full fidelity/quality for listening	Preserve intelligibility, allow compression and recognition
Common digital format	High-quality formats: WAV, FLAC, MP3	Often compressed further: speech codecs, low bit-rate formats

Note: Because speech occupies a much narrower and more predictable frequency range than music, speech-specific techniques (Sections 4-6) can compress and transmit voice far more efficiently than general music compression methods.

4. Speech Generation

Speech generation (also called speech synthesis or Text-to-Speech, TTS) is the process of making a computer produce artificial human speech, usually from written text. The goal is for the output to sound as natural and intelligible as possible.

Main approaches to speech generation:

Method	How It Works (Plain Language)
Articulatory synthesis	Computer-simulates the physical movement of the vocal tract (tongue, lips, vocal cords) — most natural in theory but very complex to model
Formant synthesis	Builds speech using mathematical models of the resonant frequencies (formants) of the vocal tract, without storing real recorded voice

Method	How It Works (Plain Language)
Concatenative synthesis	Stitches together small pre-recorded snippets of real human speech (phonemes, syllables, or words) to form new sentences
Parametric / Statistical synthesis	Uses statistical models (e.g. machine learning) trained on recorded speech to generate new speech parameters, then converts them to sound

A typical Text-to-Speech pipeline works in two broad stages: (1) Text analysis — the input text is broken into words, then into phonemes (basic speech sounds), with rules applied for pronunciation, stress and intonation; (2) Speech synthesis — the phoneme sequence is converted into an actual audio waveform using one of the methods above.

5. Speech Analysis

Speech analysis is the reverse problem of speech generation: given a recorded speech signal, extract useful information from it — such as the words spoken (speech recognition), the speaker's identity (speaker recognition), or properties like pitch and energy used for compression and synthesis.

- Pitch detection: finding the fundamental frequency of the voice — how 'high' or 'low' the voice sounds at each moment.
- Formant analysis: identifying the resonant frequency peaks of the vocal tract, which distinguish different vowel sounds.
- Linear Predictive Coding (LPC): a widely used analysis technique that models each speech sample as a weighted sum of previous samples, capturing the vocal-tract filter shape with very few parameters — heavily used for speech compression.
- Speech recognition (Automatic Speech Recognition, ASR): analyzing the acoustic signal to determine the actual words spoken, usually using acoustic models combined with language models.

Note: LPC is important because it allows the essential characteristics of speech to be described with a handful of numbers (parameters) instead of the raw waveform — this is the basis of many low-bandwidth speech transmission systems described next.

6. Speech Transmission

Speech transmission deals with how a speech signal is sent from one place to another — for example over a telephone line or the internet — while keeping bandwidth requirements low and intelligibility high.

Traditional analog telephone systems transmit speech over a limited bandwidth of about 300 Hz to 3,400 Hz — enough to keep speech understandable while saving transmission capacity. Digital systems go further by using vocoders (voice coders) that analyze the speech signal (often using LPC), transmit only a small set of descriptive parameters, and reconstruct (synthesize) the speech at the receiving end, rather than sending the raw waveform.

Transmission Approach	Typical Bit Rate	Plain Description
Uncompressed PCM (telephone quality)	~64 kbps	Raw digitized samples sent directly, simple but bandwidth-hungry

Transmission Approach	Typical Bit Rate	Plain Description
ADPCM (Adaptive Differential PCM)	~16-32 kbps	Sends the difference between samples instead of full values, since speech changes gradually
LPC-based vocoders	~2-4.8 kbps	Sends only vocal-tract model parameters; receiver re-synthesizes speech
Modern speech codecs (e.g. used in VoIP)	~6-13 kbps	Combine LPC-style modeling with perceptual coding for clear, low-bandwidth calls

Note: Notice the huge bandwidth savings — from 64 kbps down to a few kbps — which is exactly why VoIP calls and mobile phone calls use parameter-based vocoders rather than sending raw audio samples.

7. Representation of Audio Files

Once audio has been digitized, it must be stored in a file using some agreed structure (a file format) so that other software/hardware can correctly read it back. An audio file generally contains a header (information describing the audio: sample rate, bit depth, number of channels, format/codecs used) followed by the actual sample data.

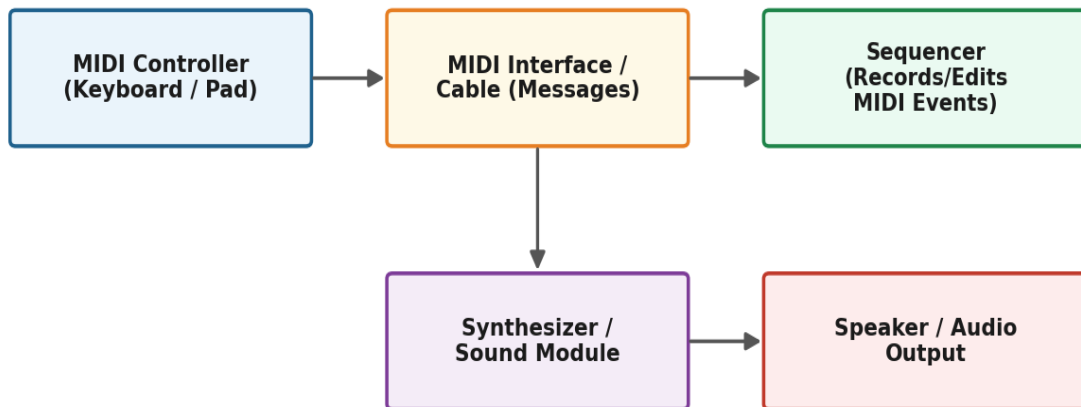
Format	Compression	Typical Use
WAV (Waveform Audio)	Usually uncompressed (PCM)	High-quality studio recording and editing
AIFF (Audio Interchange File Format)	Usually uncompressed (PCM)	Common on Apple/Mac systems, similar role to WAV
MP3 (MPEG-1 Audio Layer 3)	Lossy compression	General music distribution and streaming, small file sizes
AAC (Advanced Audio Coding)	Lossy compression	Streaming services, mobile devices, generally better quality than MP3 at same bit rate
FLAC (Free Lossless Audio Codec)	Lossless compression	Archiving and high-fidelity listening with smaller size than WAV
MIDI (.mid)	Not audio data at all — instructions only	Computer music composition and playback (see Sections 8-9)

Note: It is important to distinguish 'compression' (reducing file size of actual recorded sound, as in MP3/AAC/FLAC) from MIDI, which never stores recorded sound at all — this distinction is explored fully in Section 9.

8. Computer Music — MIDI

MIDI (Musical Instrument Digital Interface) is a technical standard that lets electronic musical instruments, computers, and other devices communicate with each other. Crucially, a MIDI file does not contain any actual recorded sound — it contains a set of instructions (events) that describe how music should be played, such as which note to play, how hard (velocity), on which instrument, and for how long.

Typical MIDI System Architecture



MIDI carries 'instructions' (note number, velocity, duration) — not actual sound — the synthesizer/sound module generates the real audio from these instructions.

Figure 3: A typical MIDI system — the controller sends instructions, and a synthesizer turns those instructions into actual sound.

Key elements of MIDI:

Element	Plain Description
MIDI message/event	A small instruction such as Note On, Note Off, or Control Change (e.g. volume, pitch bend)
Channel	MIDI supports 16 channels; each can be assigned a different instrument/sound, allowing multi-instrument arrangements
Note number	A number (0-127) identifying which musical pitch to play
Velocity	A number (0-127) indicating how 'hard' a note is struck, controlling loudness/intensity
Patch / Program	Selects which instrument sound (piano, guitar, drum kit, etc.) a channel should use
Sequencer	Software/hardware that records, edits, and plays back streams of MIDI events in time
Synthesizer / Sound module	Hardware or software that receives MIDI instructions and generates the actual audio waveform

Note: Because a MIDI file only stores instructions and not audio samples, a complete multi-instrument song can often be stored in just a few kilobytes — but the final sound quality depends entirely on the synthesizer/sound module used to play it back.

Example 3: Comparing File Size — MIDI vs Recorded Audio

Question: A 3-minute song is available both as a MIDI file (about 15 KB) and as an uncompressed CD-quality WAV recording (44.1 kHz, 16-bit, stereo). Estimate the WAV file size and compare it with the MIDI file size.

Solution / Explanation:

WAV file size = Sampling rate × Bit depth × Channels × Duration (seconds), as in Example 2.

Duration = 3 minutes = 180 seconds.

File size = $44,100 \times 16 \times 2 \times 180 = 254,016,000$ bits = 31,752,000 bytes $\approx$ 30.3 MB.

Comparison: 30.3 MB (WAV) versus about 15 KB (MIDI) — the WAV file is roughly 2,000 times larger.

This enormous difference exists because WAV stores every recorded sample of the actual sound, while MIDI stores only a short list of 'play this note now' instructions.

9. MIDI versus Digital Audio

Although both MIDI and digital audio (e.g. WAV/MP3) are used for computer music, they represent sound in fundamentally different ways, each with its own strengths and weaknesses.

Aspect	MIDI	Digital Audio (e.g. WAV/MP3)
What is stored	Instructions/events (note, velocity, instrument, timing)	Actual sampled sound waveform data
File size	Very small (kilobytes)	Much larger (megabytes for a few minutes)
Sound quality dependency	Depends entirely on the playback synthesizer/sound module	Fixed — sounds the same on every device (same recorded performance)
Editability	Very flexible — easy to change instrument, pitch, tempo, individual notes	Difficult to change individual notes; usually only whole-track edits (volume, trimming, effects)
Can represent vocals/speech?	No — MIDI only controls musical instrument-style events, no real human voice	Yes — can capture any real recorded sound including vocals
Best used for	Composing, arranging, learning music, lightweight background music	Final, faithful reproduction of an actual performance/recording

Note: In practice, many modern music production workflows use both together: MIDI to compose and arrange a piece, which is then often rendered into a final, fixed digital audio file (e.g. WAV/MP3) for distribution and consistent playback quality.

Key Takeaways — Unit 2

- Sound is a pressure wave described by frequency (pitch), amplitude (loudness), and waveform (timbre).
- Producing digital audio requires sampling (measuring at regular intervals) and quantization (rounding to digital levels); the Nyquist theorem requires sampling at least twice the highest frequency present.

- Music and speech differ in frequency range and structure, so they are often processed using different techniques.
- Speech generation (synthesis) converts text/parameters into speech using articulatory, formant, concatenative, or statistical methods.
- Speech analysis extracts information like pitch, formants, and LPC parameters from a speech signal, often as the basis for recognition or compression.
- Speech transmission uses techniques like ADPCM and LPC-based vocoders to send speech using far less bandwidth than raw samples.
- Audio files are stored with a header plus sample data; formats range from uncompressed (WAV, AIFF) to lossy (MP3, AAC) to lossless (FLAC).
- MIDI stores musical instructions, not recorded sound, making files extremely small but quality-dependent on the synthesizer.
- MIDI and digital audio trade off file size and editability against fidelity and the ability to capture real voices/instruments.

— End of Unit 2 —