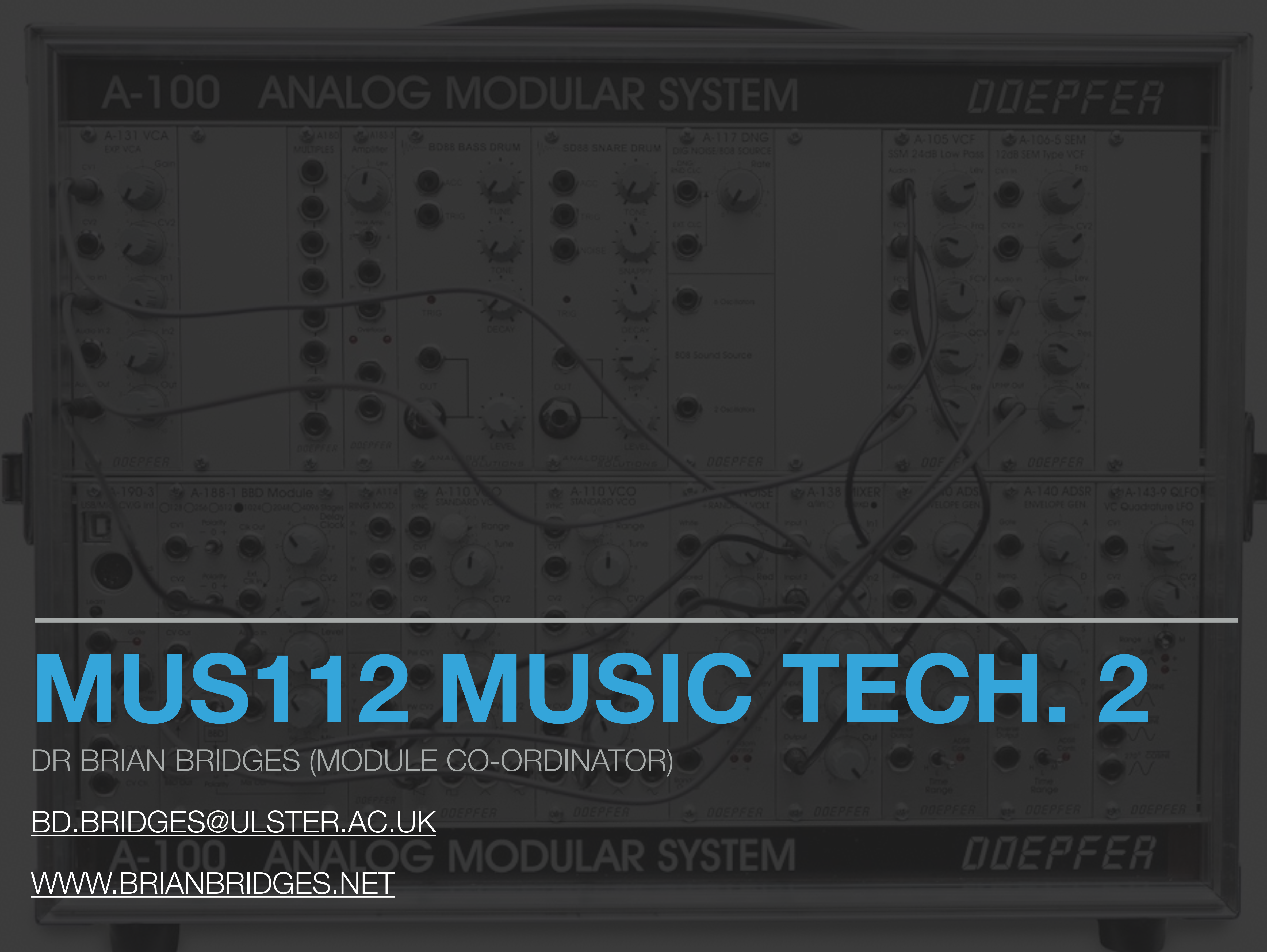



# MUS112 MUSIC TECH. 2

DR BRIAN BRIDGES (MODULE CO-ORDINATOR)

BD.BRIDGES@ULSTER.AC.UK

WWW.BRIANBRIDGES.NET

# 1980s - Digital, FM, MIDI

- The 1980s brought a new approach. Front panel and live control was reduced. Rather than a more limited range of easily customisable sounds, there was a move towards a more extensive range of 'preset' sounds.

# 1980s - Digital, FM, MIDI

- FM (Frequency Modulation) synthesis helped to encourage this approach, due to its comparatively non-intuitive programming.
- FM (briefly) is a form of synthesis derived from vibrato (variations in pitch) at very fast speeds. If this happens fast enough, frequency components (harmonics) are added to the sound. Two tone generators can be made to sound like many.
- Good for 1980s brass and bell sounds. Think 'Last Christmas'! Like many things in the 1980s, it must have seemed like a good idea at the time...It should be noted that FM was used to more subtle effect by the likes of Brian Eno and its inventor, John Chowning.

# 1980s - Digital, FM, MIDI

- MIDI stands for **Musical Instrument Digital Interface.** It was a standardised means of remotely controlling synthesizers and other music production and recording equipment.
- MIDI does not produce sound, it only tells a synthesizer module which notes to play using which preset, and when.
- Its wide adoption was facilitated by the new wave of digital synthesisers in the mid-1980s

# Alternative approaches - Wavetable synthesis

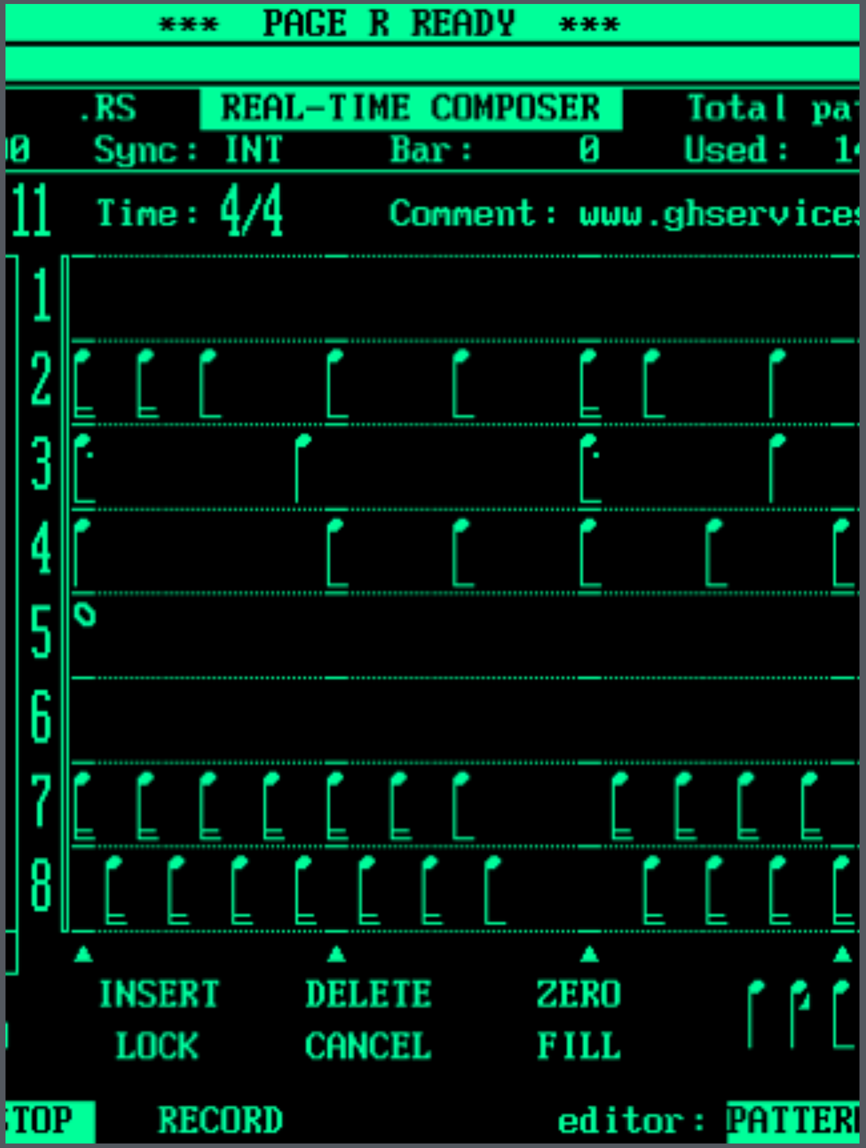
- Yamaha owned the patents to FM. (They bought them off Stanford University and John Chowning, the process's inventor.)

- Other manufacturers tried other approaches. PPG tried a particularly interesting one with Wavetable synthesis - a series of different audio 'frames' were scrolled through, producing a dynamically evolving and 'morphing' in the sound.

- This particular example, the PPG Wave, still used analogue filters for 'warmth'. 80s synthpop band Depeche Mode liked them a lot.

# Sampling/sequencing - Fairlight CMI

- The Fairlight Computer Musical Instrument was an early sampler/sequencer.
- Samplers allowed snippets of sound to be digitally recorded, edited and arranged in various orders.
- It had very limited memory to store recordings, and was so expensive that only large studio facilities could afford to purchase it.



# Drum Machines

- A much more specialised beast, the LinnDrum also used digital samples, but preset ones of drums. Compared to previous drum machines, it sounded very realistic.

- At around this time, the Musician's Union in England became concerned about the threat drum machines posed to the livelihoods of drummers!

# Drum Machines

- However, an unlikely success was to be found in the Roland drum machines of the day - the TR-808 and TR-909.
- These did not sound at all like acoustic drums, since they used basic analogue circuits, but they had a sound ‘signature’ which suited early dance music producers, and became quite sought-after.
- Many modern drum machines now offer copies of these sounds. They have become part of the music production ‘vernacular’.

# Virtual Analogue Synthesis

- As the 1990s rolled around, Fashions changed, and analogue synths came back into fashion.

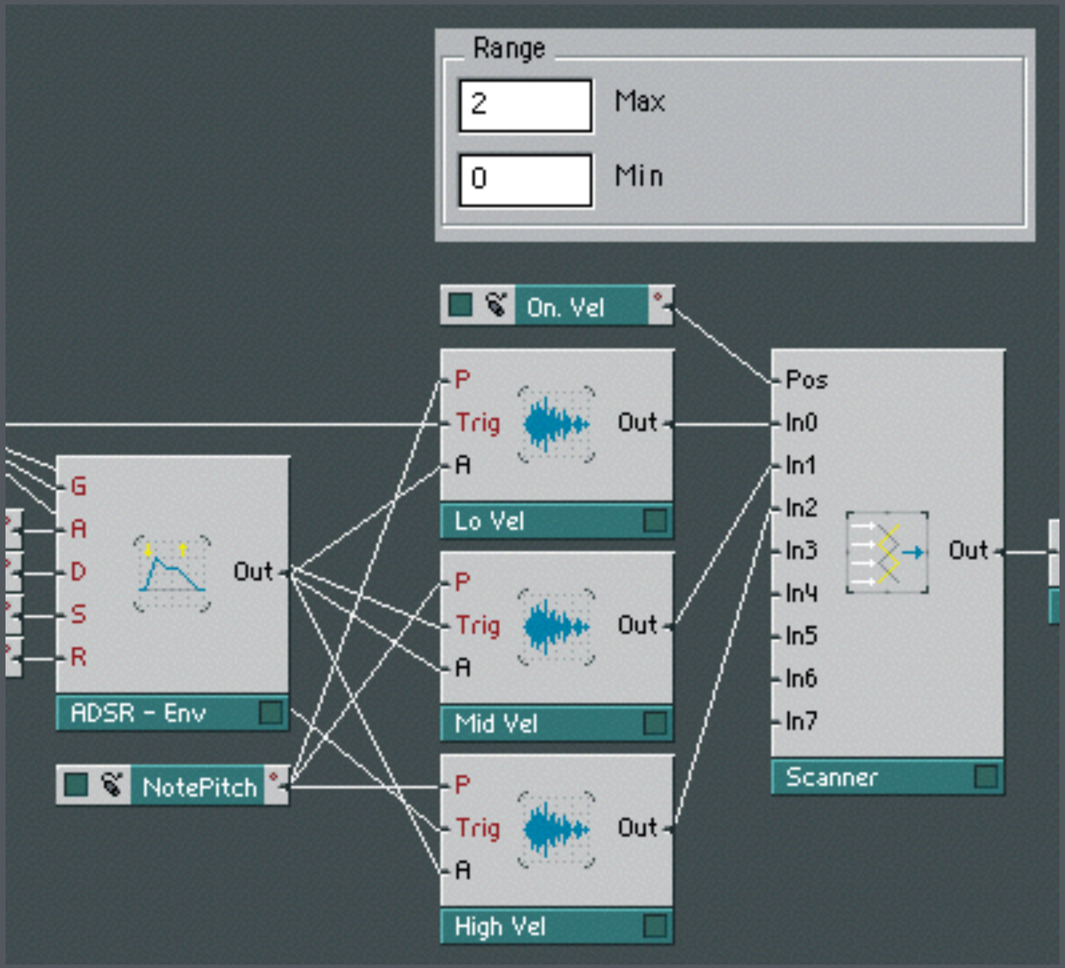
- However, analogue machines could be expensive, hard to find and unreliable.

- Clavia, a Swedish company, came to the rescue with their Nord line of synths, designed to use digital technology to 'model' the behaviour of analogue circuits.

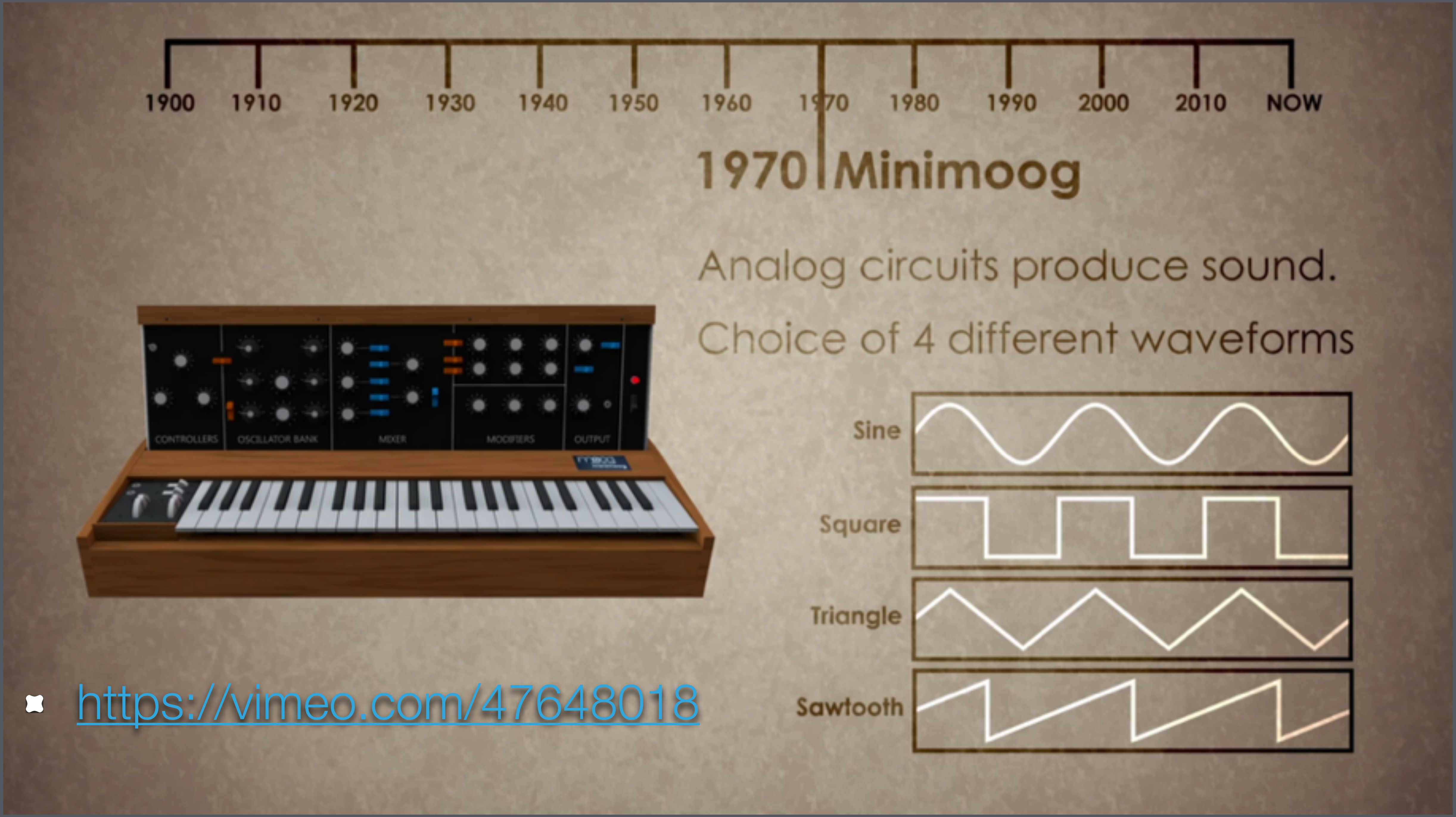
- This approach became known as 'virtual analogue synthesis' or 'analogue modelling'.

# Computer Control and Synthesis

- Mid-late 1990s - computers get faster, capable of generating live audio data
- Software synthesis systems become more important - pre-built synthesizers such as FM8, Massive and Imposcar or more open systems such as Reaktor and Max gain in popularity and feasibility
- Sonically, the 21st century becomes a truly postmodern age! A range of 'historical' synthesis methods become instantly accessible, along with some new experimental ones

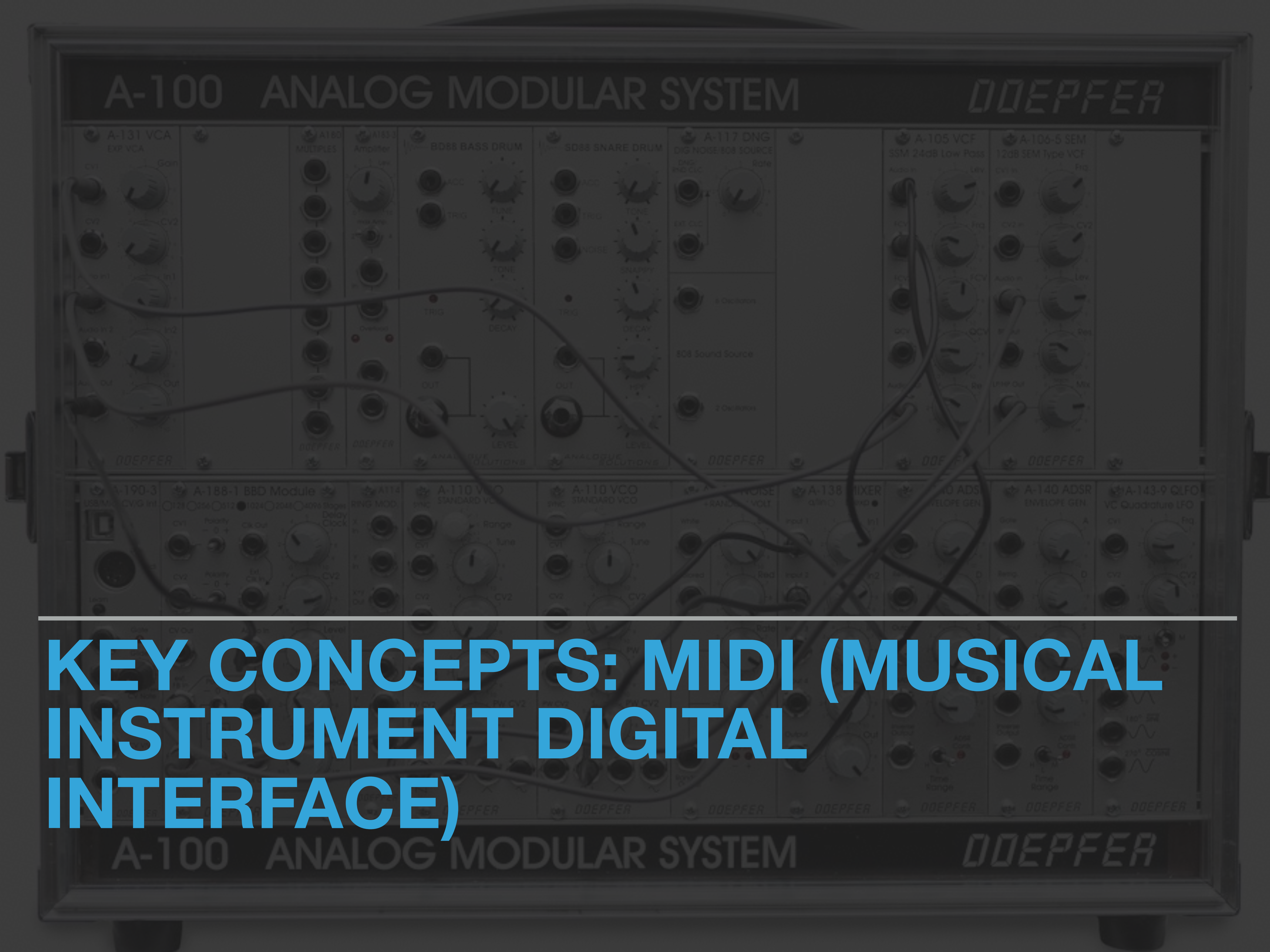


# Recap: Illustrated Video History of Electronic Instruments




- https://vimeo.com/47648018

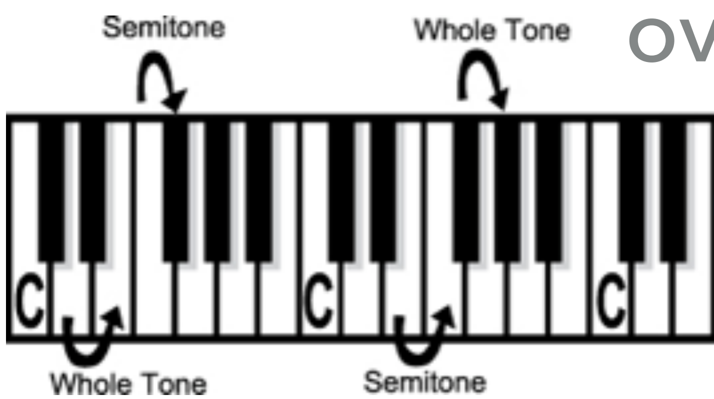
# KEY CONCEPTS: MIDI (MUSICAL INSTRUMENT DIGITAL INTERFACE)



# MIDI AND DIGITAL AUDIO—FLOWCHART

MIDI messages

over USB lead




numbers describing audio

Audio interface

electrical signal describing audio

Computer running Logic

(Logic hosting software instruments)

interprets MIDI messages,

sends to software instruments

# MIDI: MUSICAL INSTRUMENT DIGITAL INTERFACE

- Released in January 1983: control protocol to integrate control of synthesisers, controllers and other studio hardware
- ‘Remote’ control messages, *not audio signals* (i.e. may send instructions which contribute to triggering sounds, alteration of sound generation parameters)
- Small amounts of data/low bandwidth

# MIDI STRUCTURE: BASICS

- Event messages with modifiers
- **Play key= MIDI note number** from 0-127, **MIDI note velocity** in same range ***(note on)***
- **Release key= same MIDI note no, MIDI velocity=0** ***(note off)***
- e.g. send message from controller keyboard (*not synthesiser*) to software instrument (synthesiser) within Logic
- Q: Why design a MIDI message like this? Does it explain how a MIDI-related problem sometimes occurs when you stop playback in Logic?
- *The structure of these messages will become important when you work with Interactive Music Systems in year 3*

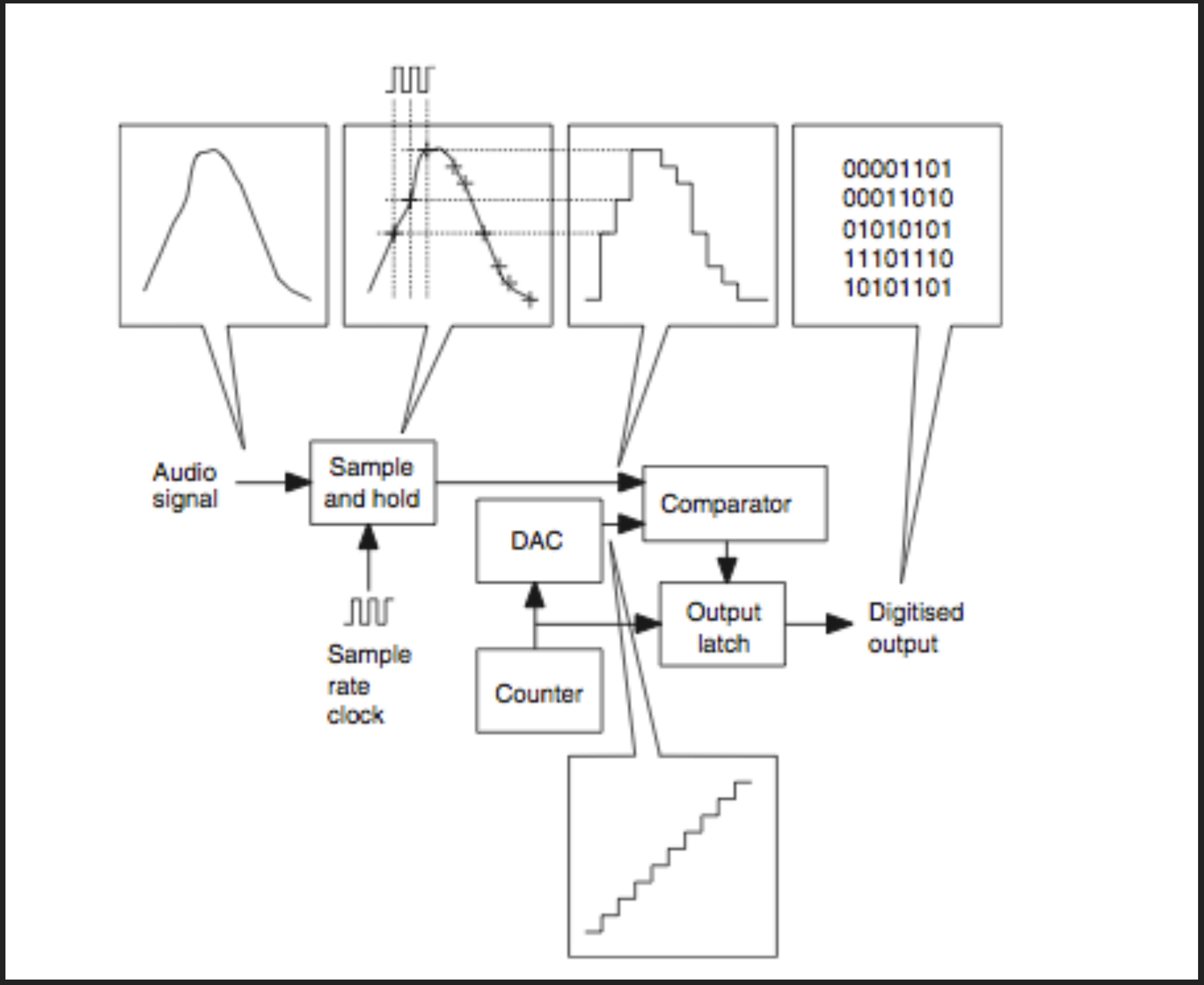
## MIDI/SYNTHESISER TERMINOLOGY/JARGON

- A number of confusing pieces of semi-contradictory jargon have grown up with synthesisers over the forty-odd years they have been in common usage
- A **sound** setting on a synthesiser is known as a **patch** or a **programme** or a **preset** - *it is nothing more than a collection of settings which describe the sound* - the position of the knobs and sliders on a front panel of a software instrument
- **Patch** comes from the **patch-cables** which were once used to connect the different modules in an old-style analogue synthesiser to define the sound


# KEY CONCEPTS: DIGITAL AUDIO

# DIGITAL AUDIO: BASICS

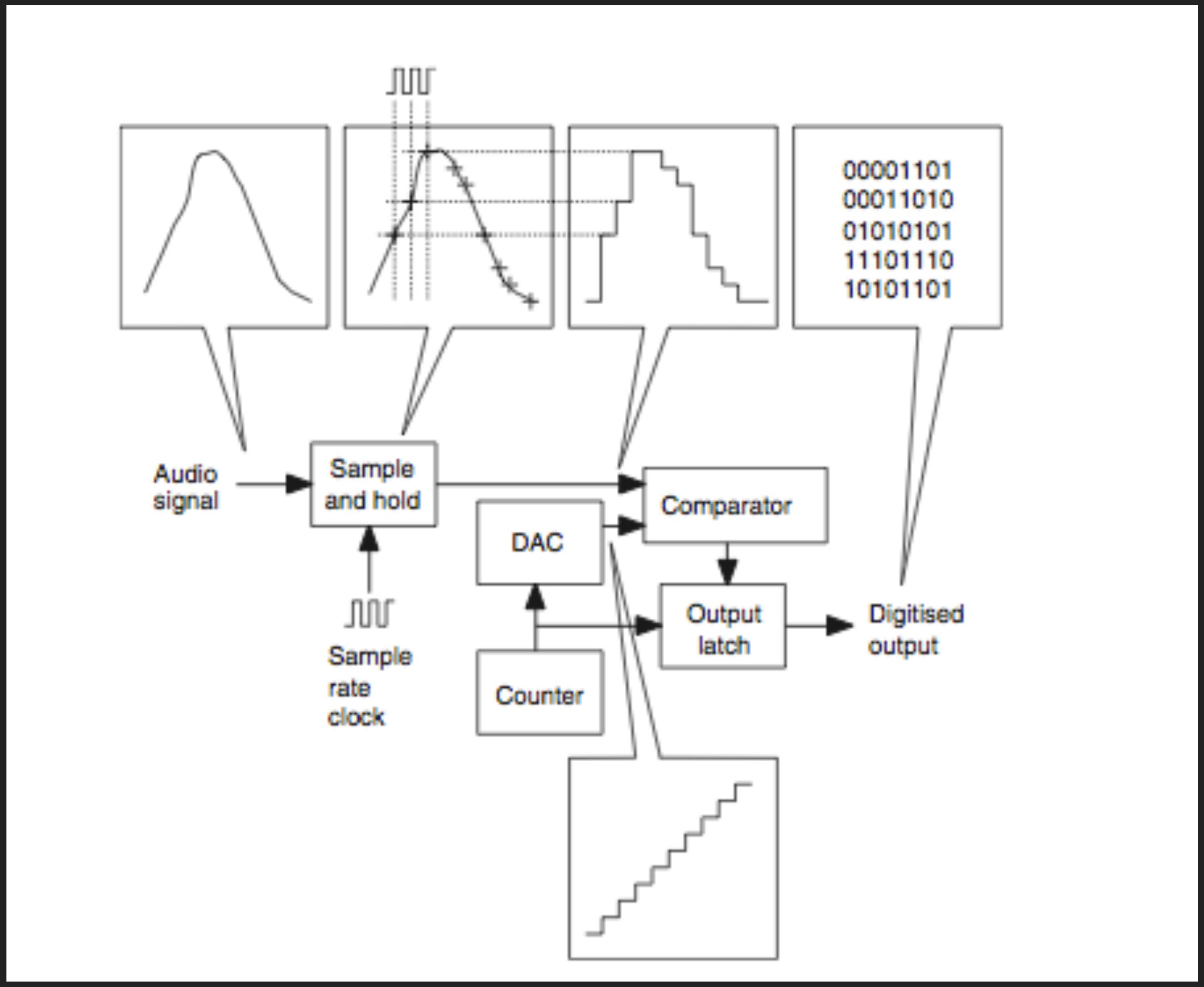
- Digital audio is simply 'sound (accurately) described by numbers'
- *Sound is a vibration in the air around us which we receive at our ears (and decode in our brains)*
- Digital audio is simply another means of *encoding* the air pressure vibration that is sound so that we can transmit or alter it
- Digital technology is the key source of the power of modern audio production tools




From Russ, M. 2009. *Sound Synthesis and Sampling*. 3rd ed. Oxford, Focal, p. 61

# DIGITAL AUDIO: BASICS

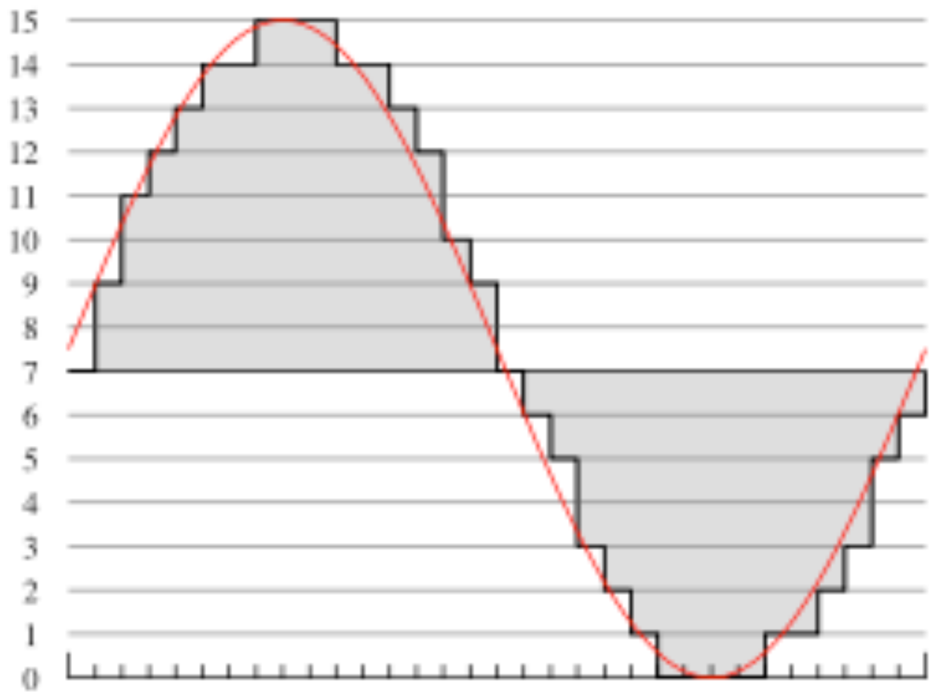
- Sampling is the process of changing the continuously-varying (analogue) signal into a discrete set of digital readings (number values for specified time intervals)
- The process here shows an audio signal being sampled at a regular rate: this reads off the size (amplitude) of the air pressure variation for each time interval
- We then have numbers which embody a simplified representation of the original audio (depending on how accurate our sampling system is)
- Accuracy in sampling depends on two things: amplitude resolution (bit depth) and frequency resolution (sample rate)

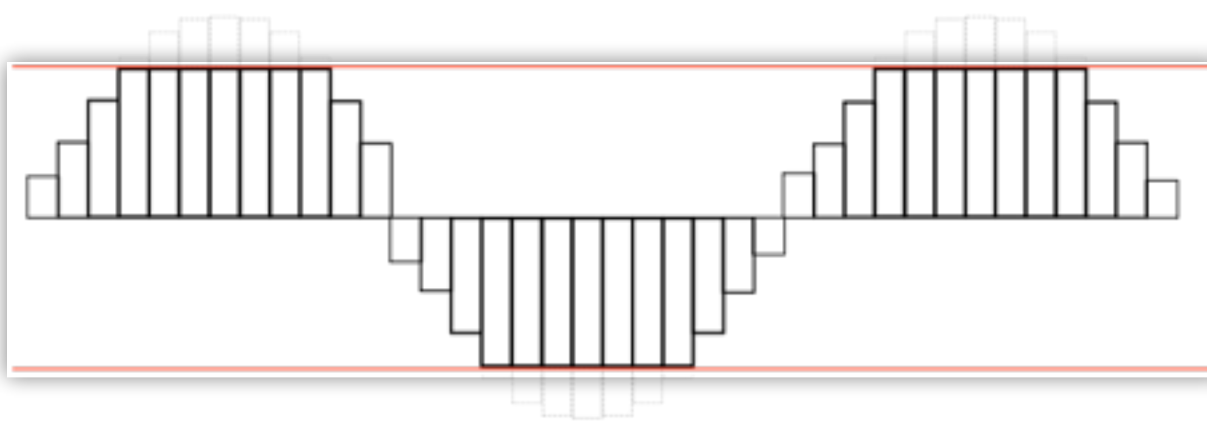



From Russ, M. 2009. *Sound Synthesis and Sampling*. 3rd ed. Oxford, Focal, p. 61

# DIGITAL AUDIO: CLIPPING

- Our digital audio sampling system will have minimum and maximum signal levels which it can measure/record reliably
- If we go over the maximum level, we will not be able to record the signal accurately, we will simply record the maximum level (i.e. number), whatever the signal itself is doing—this is known as clipping
- This results in extreme distortion: rather than the wave being progressively bent out of shape (as is the case with analogue distortion), clipping cuts off the peaks of the wave abruptly (chops the 'heads' off), resulting in sudden and significant audible distortion which is particularly harsh



clipping

## DIGITAL AUDIO: BASICS

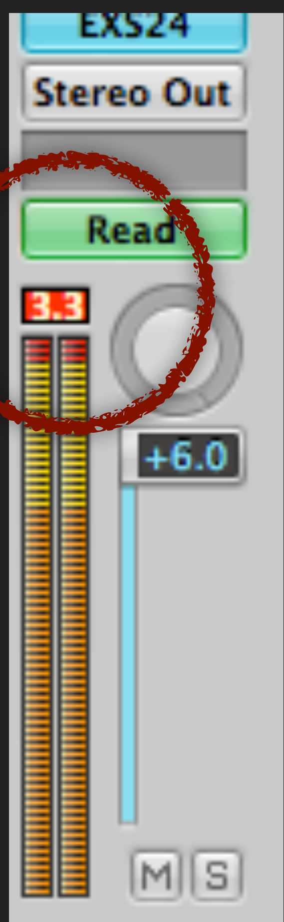
- **Resolution/bit depth gives us the amount of amplitude data and hence, dynamic range for recordings,** between minimum and maximum levels in a system: **expressed in bits**…numbers of on/off binary numbers needed to encode ‘normal’ decimal numbers **…24 bit=pro recording**=16,777,216 level values, **16 bit=CD audio**=65,536 level values

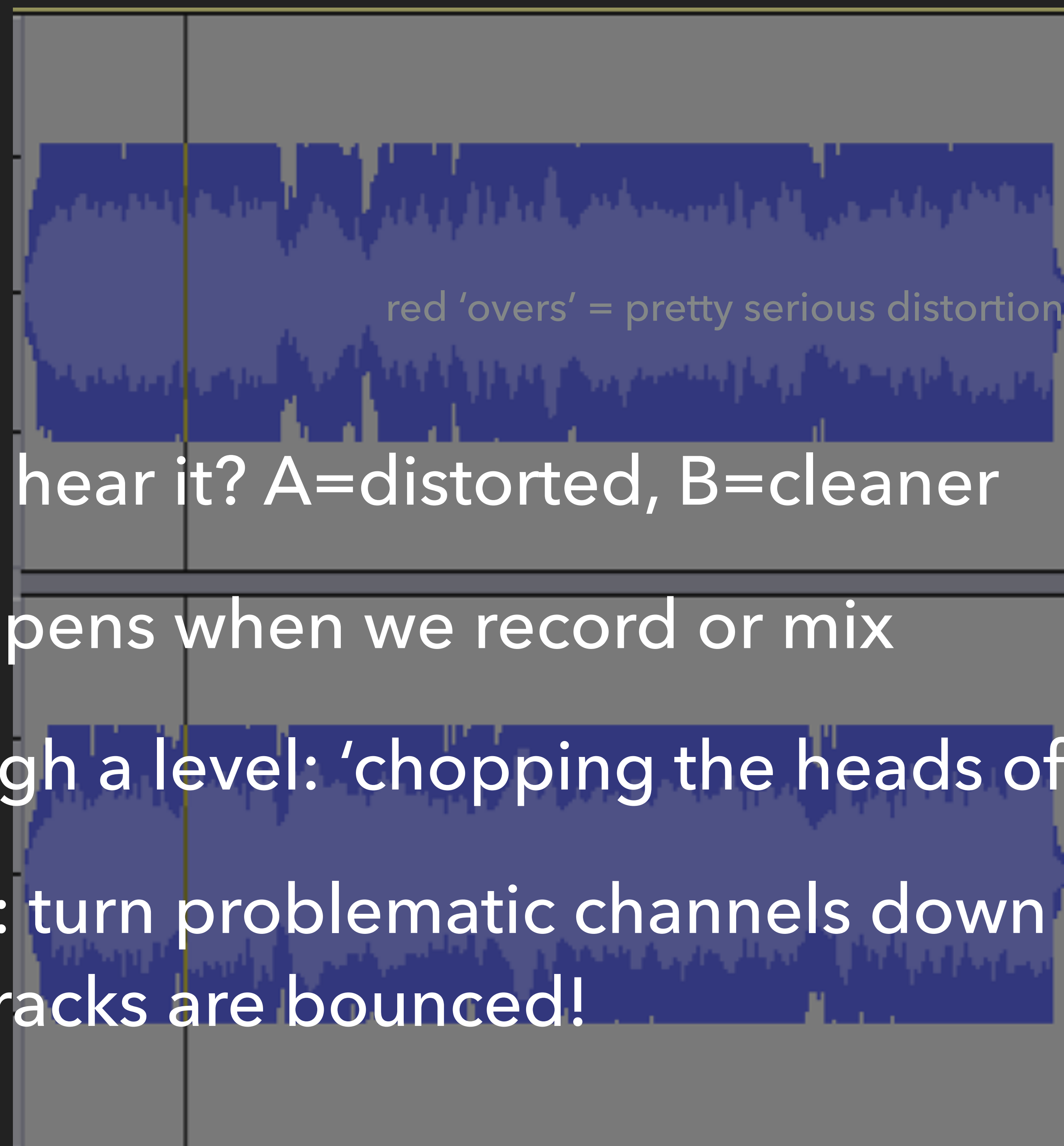
- **Sampling rate (for frequency resolution):** needs to be twice maximum audible frequency (c. 20 kHz), so rates of **44.1 kHz+** are commonly used

- **Uncompressed and compressed audio:** MP3 files are not full-quality audio, but use a lossy model based on our hearing abilities to reduce file size dramatically (nonetheless, many listeners can hear the difference between MP3 and uncompressed audio such as .wav or .aif/.aiff files)

- **About surround:** *don’t* use surround plugins/bounce options unless you are using a multi-speaker surround sound facility (such as studio 2=> you will not be using this facility this year!)

# DIGITAL AUDIO PROBLEMS: CLIPPING




Can you hear it? A=distorted, B=cleaner

This happens when we record or mix

at too high a level: ‘chopping the heads off’

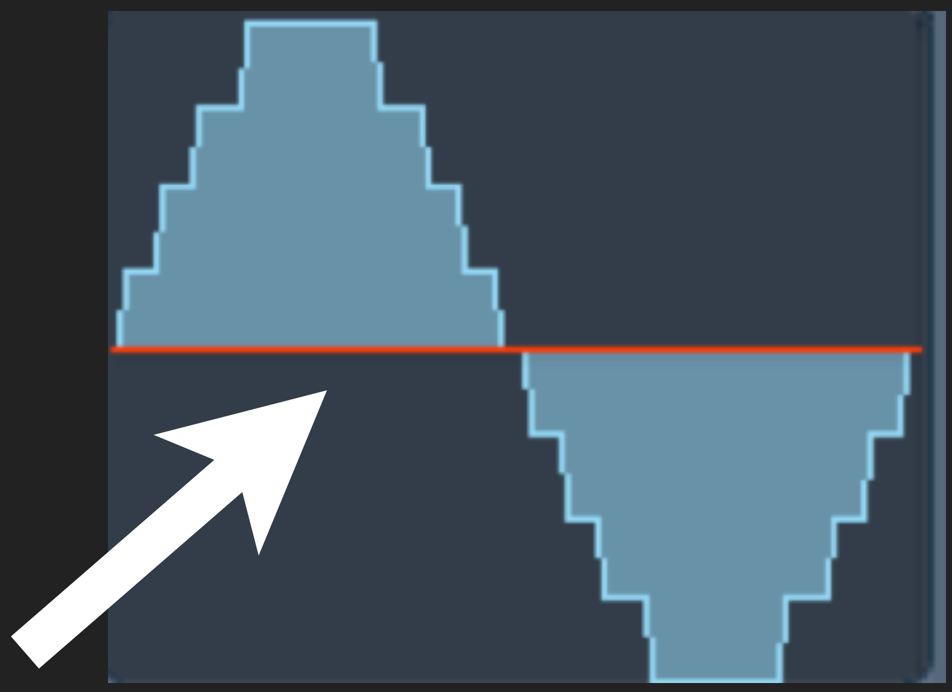
solution: turn problematic channels down before tracks are bounced!

## DIGITAL AUDIO PROBLEMS: LOW BIT DEPTH (AUDIO RECORD LEVEL TOO LOW)

If we record at too low a level, we may have problems at the other end: low-level/quiet sounds (e.g. the *decay phase* of a note or the *tail* of a reverb may become distorted: quantisation noise

This effect is exacerbated by lower sampling rates (16 bit, 12 bit…8 bit if you're feeling retro!)

Have you heard this effect before? Where?



sound wave becomes

'pixellated'

# ASIDE & TRIVIA: 8 BIT OR CHIPTUNE?

- Some people like to work with really low-fi digital recordings or synthesis
- So-called *8-bit* (AKA *chiptune*) music is generally not actually 8–bit: it's 4-bit (for even crunchier effects
- The name 8-bit comes from the main processor of old computers and games consoles, not their sound chip
- Chiptune is therefore a more accurate name!
- An example of the genre, here: [www.youtube.com/watch?v=A5TqDx7iWvQ](www.youtube.com/watch?v=A5TqDx7iWvQ)

# DIGITAL AUDIO PROBLEMS: LOW SAMPLING RATE

- Sampling rate is how frequently a computer records or generates a number for changing level data in an audio file
- All sound is vibration: digital sound is a set of numbers storing or generating that vibration
- **We can hear vibrations in a range from 20 per second (20 Hertz) to an upper limit of 16,000 to 20,000 per second (16–20 kiloHertz)**
- Samping rates need to be **double** the highest frequency in a piece of audio: frequencies higher than this limit are 'reflected down' into lower ranges, 'messing up' the frequency spectrum (and making the sound 'harsh', 'edgy', or 'metallic'…the technical term is **inharmonic**)
- Therefore, we use sampling rates of **44.1 kHz** plus to accurately encode audible sound

# DIGITAL AUDIO PROBLEMS: LOW BIT-RATE MP3S

- MP3 (MPEG 1, Layer 3) is an audio standard for **data compression** of an audio file so that it may be more easily transmitted over networks or stored on low-capacity devices
- Typically, it offers savings of 6:1 or greater on **uncompressed file sizes**
- However, there is a trade-off: it's a **lossy** compression scheme, removing audio data based on a model of the behaviour of our inner ear structures
- In some cases, **due to the structure of our ears, sounds in a similar frequency range may 'block' (mask) other sounds…**

# DIGITAL AUDIO PROBLEMS: ‘GOOD’ AND ‘BAD’ MP3S

- Sampling rate is how frequently a computer records or generates a number for changing level data in an audio file
- All sound is vibration: digital sound is a set of numbers storing or generating that vibration
- **We can hear vibrations in a range from 20 per second (20 Hertz) to an upper limit of 16,000 to 20,000 per second (16–20 kiloHertz)**
- Samping rates need to be **double** the highest frequency in a piece of audio: frequencies higher than this limit are ‘reflected down’ into lower ranges, ‘messing up’ the frequency spectrum (and making the sound ‘harsh’, ‘edgy’, or ‘metallic’…the technical term is **inharmonic**)
- Therefore, we use sampling rates of **44.1 kHz** plus to accurately encode audible sound

**‘GOOD’ MP3: 192 kbit example**

# DIGITAL AUDIO PROBLEMS: ‘GOOD’ AND ‘BAD’ MP3S

- Sampling rate is how frequently a computer records or generates a number for changing level data in an audio file
- All sound is vibration: digital sound is a set of numbers storing or generating that vibration
- **We can hear vibrations in a range from 20 per second (20 Hertz) to an upper limit of 16,000 to 20,000 per second (16–20 kiloHertz)**
- Samping rates need to be **double** the highest frequency in a piece of audio: frequencies higher than this limit are ‘reflected down’ into lower ranges, ‘messing up’ the frequency spectrum (and making the sound ‘harsh’, ‘edgy’, or ‘metallic’…the technical term is **inharmonic**)
- Therefore, we use sampling rates of **44.1 kHz** plus to accurately encode audible sound

**‘BAD’ MP3 example: 64 kbit**

‘dull’, ‘uneven’, ‘warbling’, ‘frequency details lost’,‘timing details lost’

# DIGITAL AUDIO PROBLEMS: LOW BIT-RATE MP3S

- Think of the effect of trying to listen to music in a car with a loud engine! You miss some of the lower frequencies because the car 'blocks' them
- MP3 works on this principle: certain sounds/parts of sounds in a mix will **mask** others, therefore you don't need to encode them
- MP3 **encoders** can even try to filter out sounds that you might here to reduce the file size further, but at a cost to audio quality
- Guideline: low bit-rate MP3s (e.g. 128 kbit or less) are likely to interfere with the quality of your audio (sometimes in unpredictable ways)
- Solution: use 148/196 kbit or higher for casual listening, but used **uncompressed file formats** (.aif, .wav) **for archiving**

# DIGITAL AUDIO PROBLEMS: ‘MORALS’

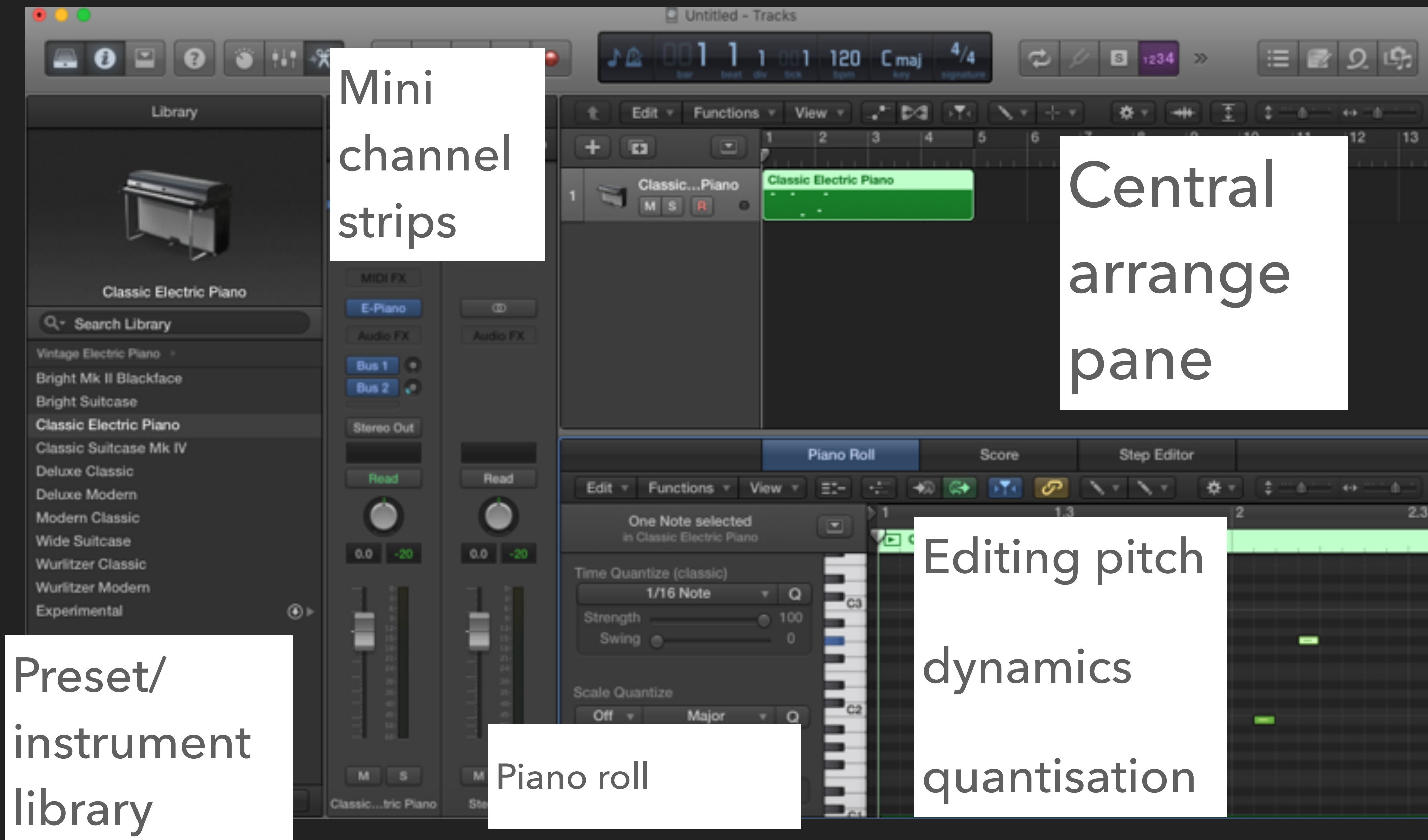
- Take care of levels when recording/mixing to prevent clipping
- Use at least 16 bit audio settings when recording (preferably, 24 bit) to prevent quantisation noise
- Use a sampling rate of 44.1 kHz or higher (opinion is still divided on the relative merits of higher rates, but that’s a more advanced issue)
- **NB: Don’t use compressed audio files (e.g. MP3s) as an archival format! Use uncompressed (PCM: .aif or .wav)**

# RECAP: INTERFACE AND KEY FUNCTIONS

# LOGIC'S INTERFACE: RECAP

- Setting up instrument tracks
- Arrange page panes (including mixer)
- Browsing for presets and Apple loops
- File management
- Bouncing

# LOGIC'S ARRANGE WINDOW: OVERVIEW

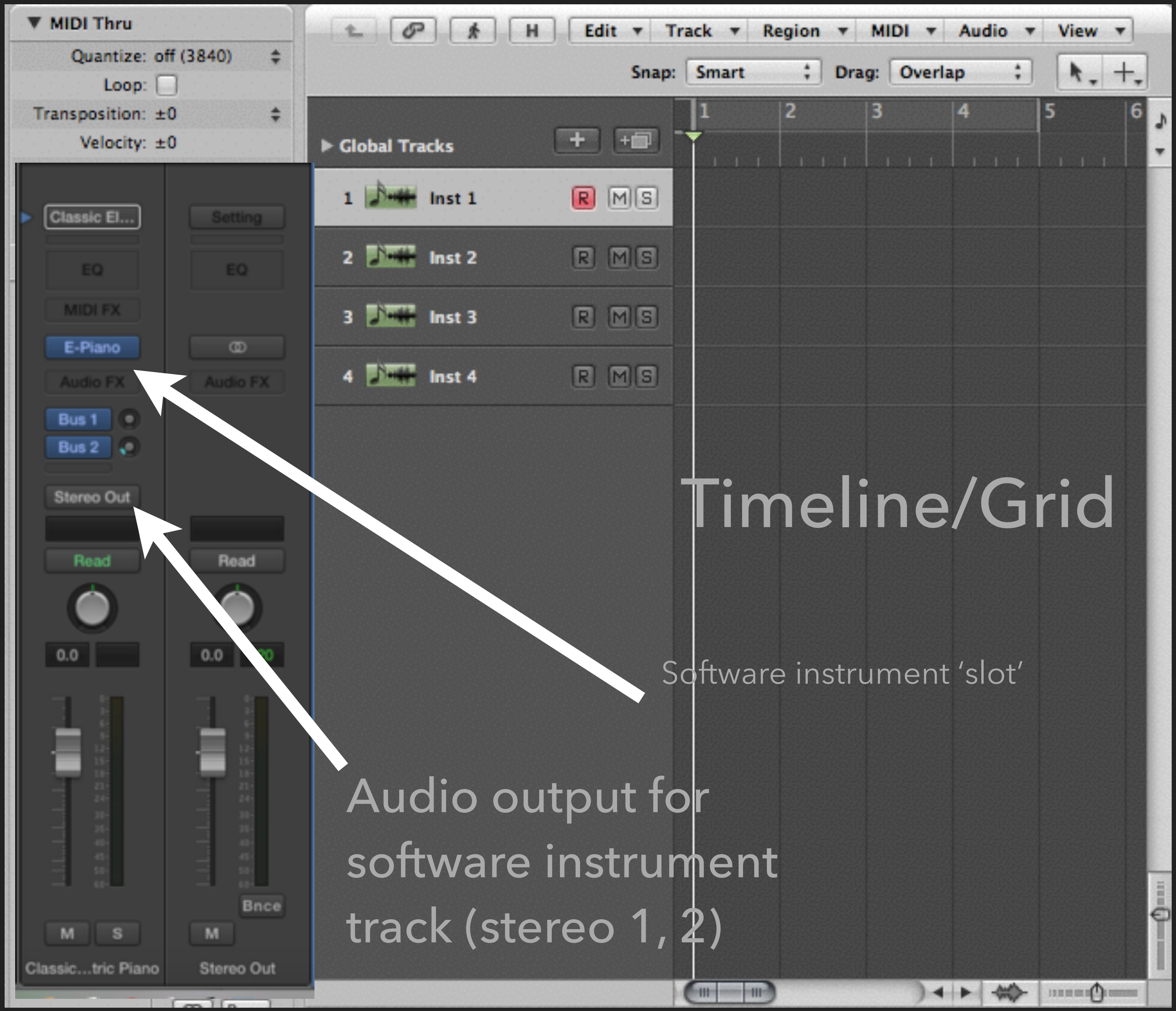


# LOGIC'S MIXER PANE

# RECAP: CREATING PROJECT AND SETTING UP TRACKS

- Set up **software instruments** to play with Logic's internal synths
- (External MIDI for external synthesiser: we won't use this here)
- Audio tracks (for digital audio recordings made in Logic or imported into Logic)

# ARRANGE WINDOW AND INSTRUMENTS



# INSTRUMENTS (SOFTWARE SYNTHESISERS)

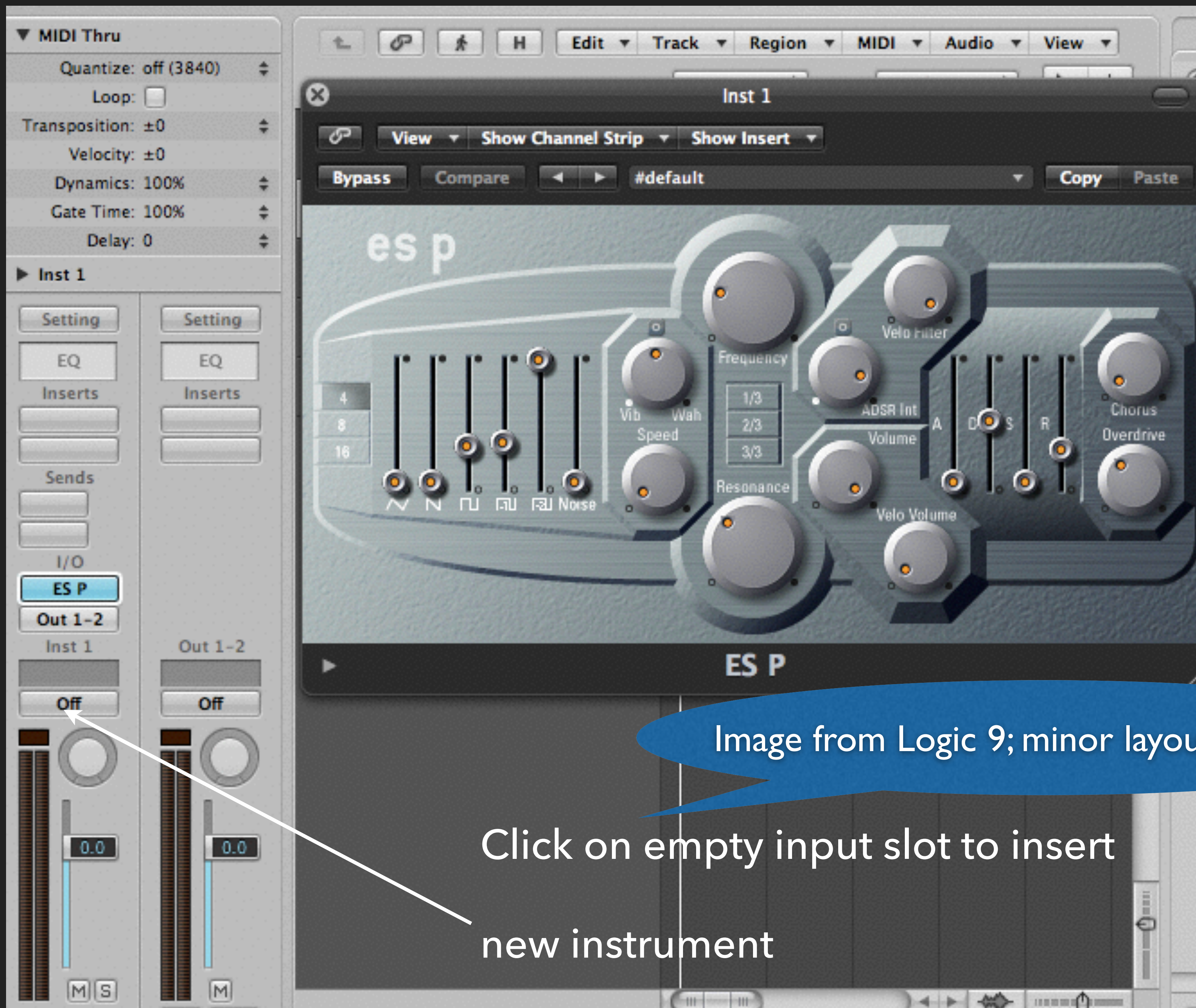
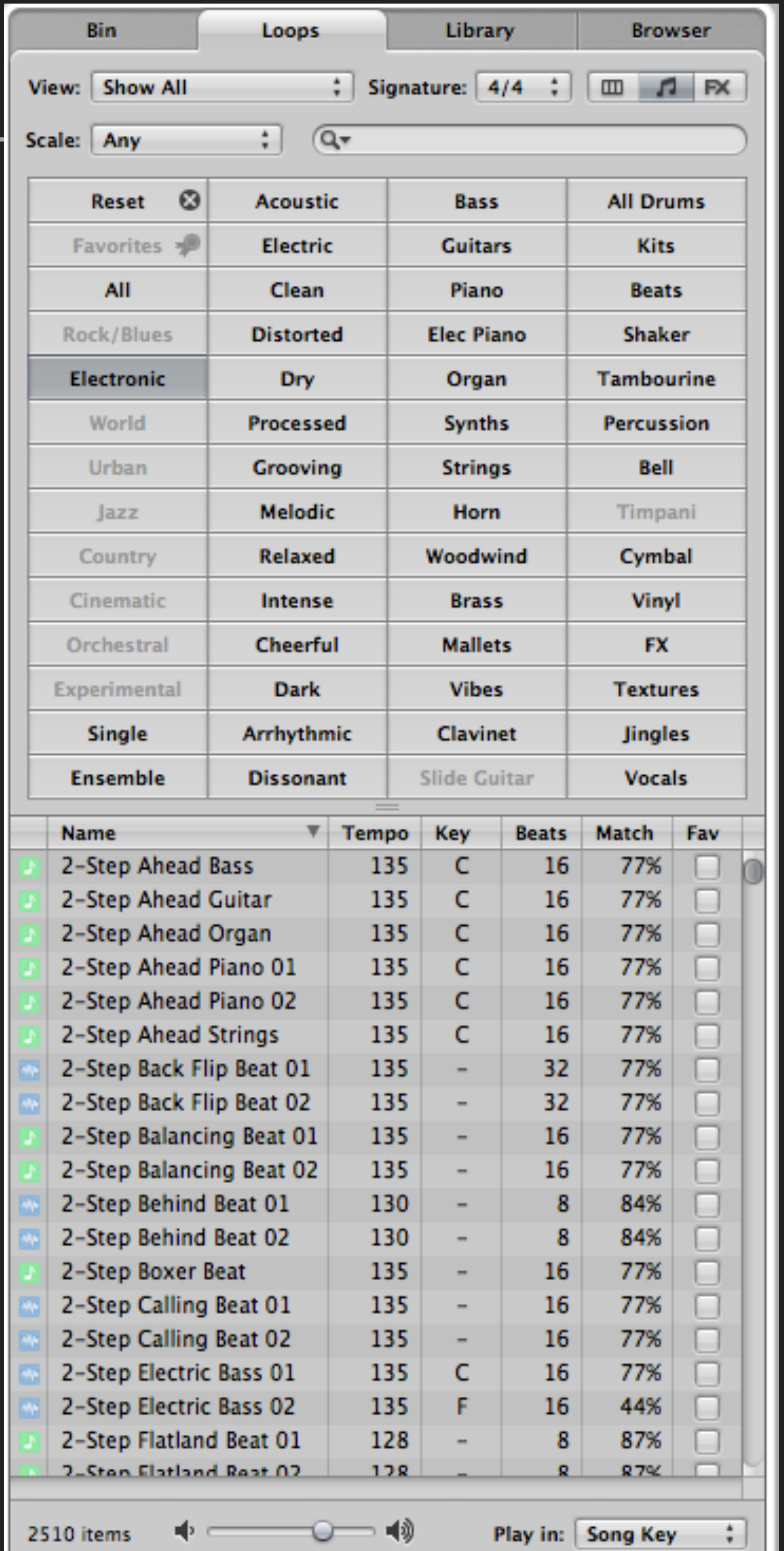



Image from Logic 9; minor layout difference

Click on empty input slot to insert new instrument

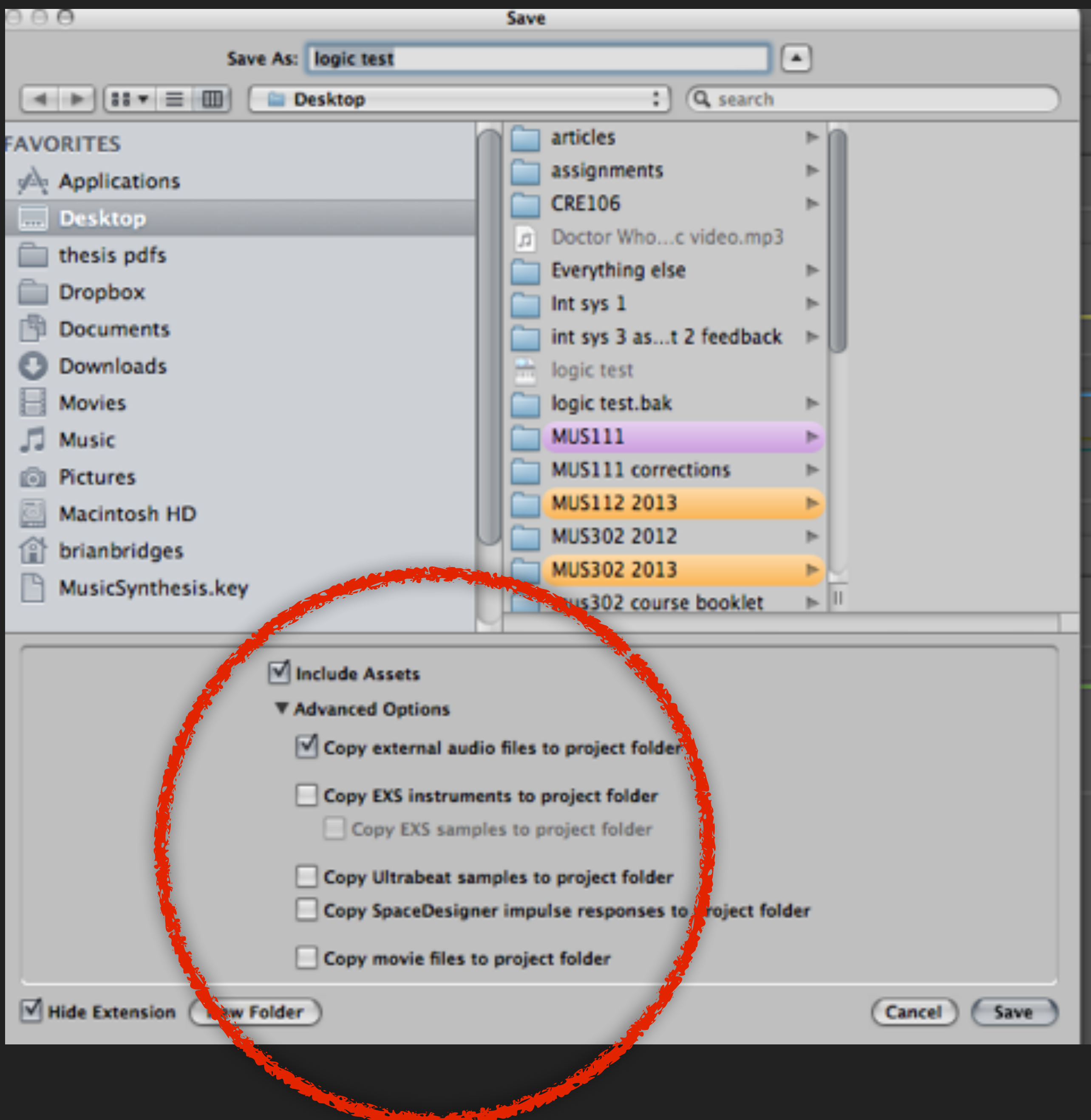
# APPLE LOOPS

- One topic we may not have yet investigated in a structured way is Apple Loops
- What is the difference between the two types of Apple Loops? Why is it useful to have these two types of Apple Loops? Have you experienced any problems with one or the other?
- Green Apple loops: software instrument loops (based on MIDI data)
- Blue Apple loops: audio loops (specially treated with beat detection so that they respond to tempo changes)
- See Chapter 2, *Exploring Logic Pro 9.*



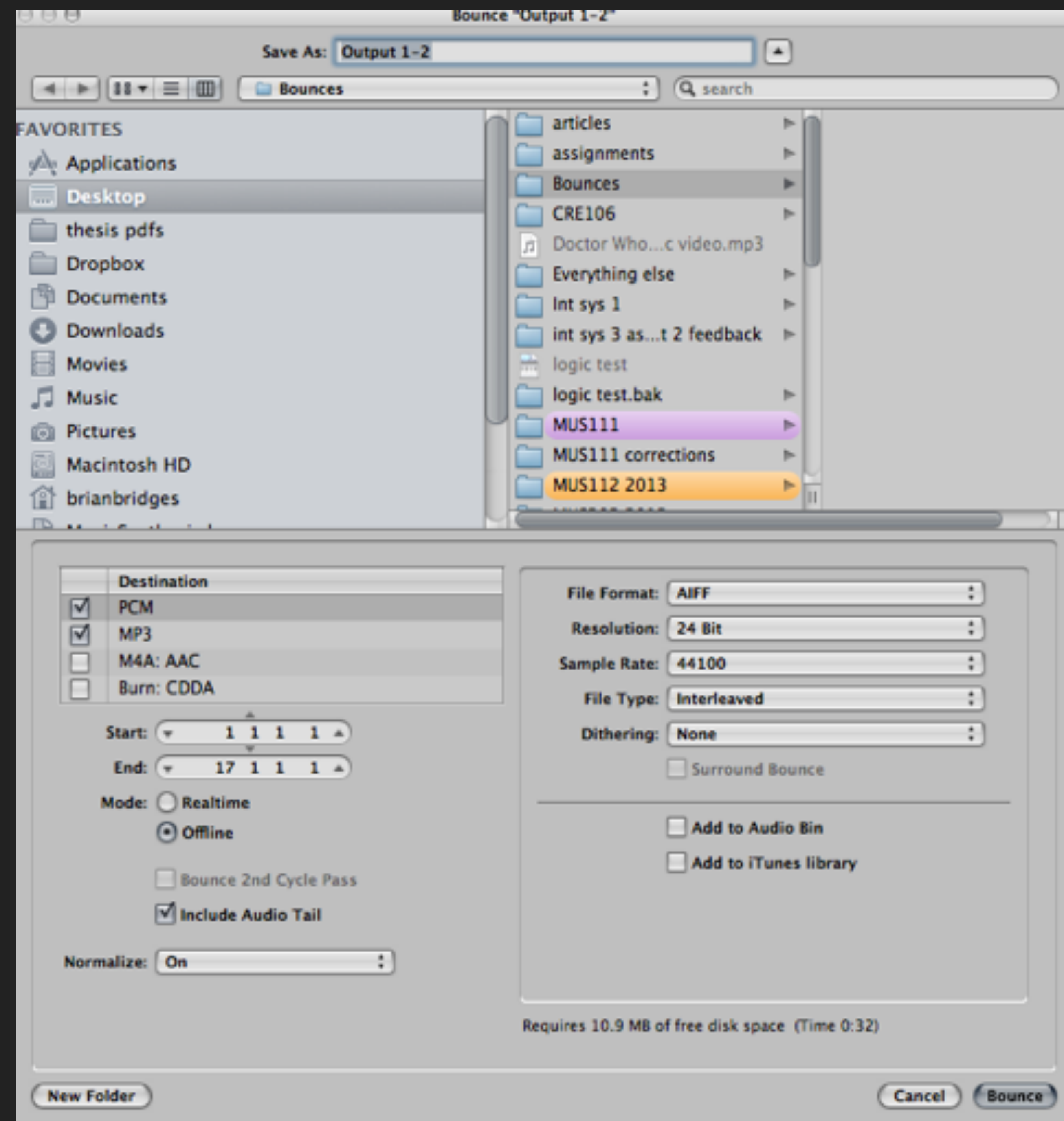
# FILE MANAGEMENT

- **File management will become increasingly important to you as you begin as you begin to use digital audio recordings**
- Remember, the Logic project file will only contain MIDI and automation/layout data
- This is fine if you are using Apple loops/ sampler instruments, but if you use your own audio, you will need to make sure that everything your project needs is on your CD/DVD: **use the 'include assets' setting when you 'save as'**



# MIXDOWN/BOUNCE

- Choosing file formats:
- Compressed - MP3 - or uncompressed - ‘PCM’
- Sample rate (44100 or above), resolution (24 bit for archiving, 16bit with dithering for CD output)
- Start/end points (and include audio tail)
- Normalise



# SOFTWARE SYNTHESIZERS AND SYNTHESIS TECHNIQUES (DEMO)


Analogue-style (subtractive) 1960s on



Digital approaches (wavetable and FM) 1980s on (commercialisation)

**Logic X's 'Retrosynth' gives you access to 3 key types of synthesis technique**

We'll cover these different techniques in more detail later on and in coming weeks

# MORE INFORMATION

- Russ, M. 2009. *Sound Synthesis and Sampling*. 3rd ed. Oxford: Focal
- ASK Video Tutorials (software tutorial DVD in library)
- Logic tutorial texts (see module descriptor/reading list)
- Questions about this lecture? BD.Bridges@ulster.ac.uk

# SOUND SYNTHESIS IN LOGIC PART 1: SUBTRACTIVE


BD.BRIDGES@ULSTER.AC.UK
WWW.BRIANBRIDGES.NET

## WHAT IS A SYNTHESISER?

- What does it do?
- What does it not do?
- Attempt at a definition?
- **Key point: synthesisers build up complex sounds from component processes.** It is important to understand how these components work so that you can eventually create more complex sounds (either 'from scratch', or customising existing sound presets)

## SOUND SYNTHESIS

- Generating musical or non-musical timbres (aka 'sound textures') from

  (1) fundamental building blocks such as pure tones (sine waves/single-frequency components - similar to the sound produced by a tuning fork)

  (2) complex sound waves electronically shaped by filters (tone controls) or distortion

- *Timbre* = tone colour or sonic 'signature'
- *Synthesisers* (aka synthesizers) - pieces of equipment which generate sound through a variety of processes (additive, subtractive, mathematical models and/or processes)

# SYNTHESIS TECHNIQUES

- As implied by the foregoing, there are many different types of technique by which you can synthesise sound
- Some are good at producing **'realistic' instrument sounds**, others are useful for more **abstract or 'electronic-sounding' timbres**
- **(1) Additive synthesis:** creates sounds based on **adding simpler component sounds**-- 'pure tones' (sine waves, which are vibrations which produce only one frequency component)--to construct a complex sound with many frequency components =>**not used in most software synthesisers due to its complexity**

# SYNTHESIS TECHNIQUES

- (2) **Subtractive synthesis** - a **rich sound** (with many frequency components) is **shaped by a filter**––early **analogue synthesisers were of this type** (although we now use digital methods to achieve the same type of sonic results)
- (3) **Modulation synthesis** - one **sound generator imposes its vibration on another one** (in a similar manner to vibrato in an acoustic instrument) at a high frequency, causing the resulting vibration to become more complex (original sound wave is ‘bent out of shape’), **resulting in a richer sound**

# SYNTHESIS TECHNIQUES

- (4) **Physical modelling synthesis** - a musical sound is generated based on a **simplified mathematical model of the physical system in a known musical instrument** (or an extrapolation of one)

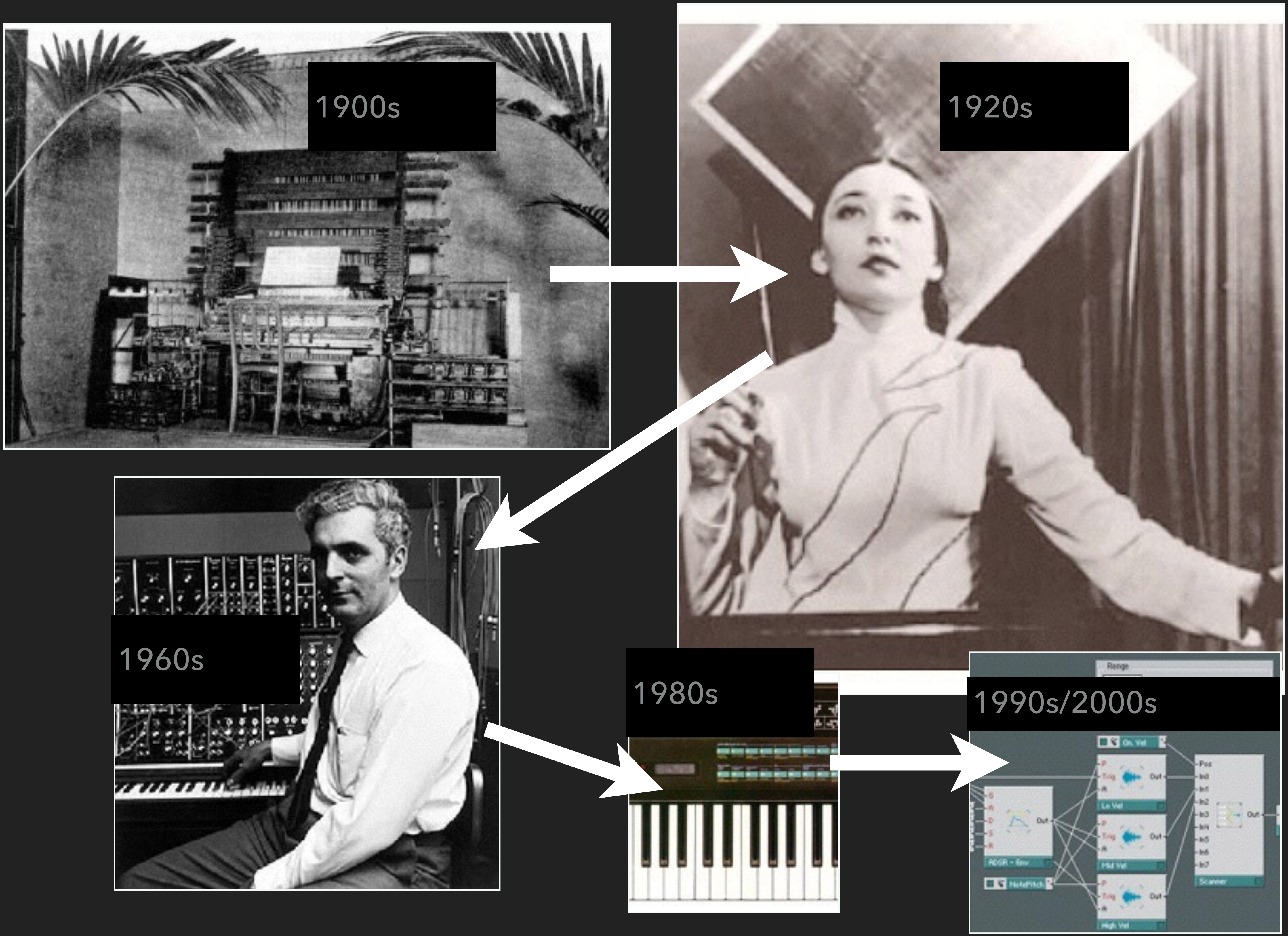
- (5) **Wavetable synthesis** - a **musical sound is generated through 'scrolling' (reading) through a 'table' of different soundwave shapes** (think of them as like film/video frames), resulting in a sound whose timbre evolves as the wave shapes change

- *[Confusingly, any digital sound generator which reads off a set of digital numbers to produce a given wave shape can be said to be using a wavetable, but* ***wavetable synthesis is properly reserved for this dynamic version]***

# OTHER SOFTWARE INSTRUMENT TECHNIQUE

- (Not synthesis) **Sampling** can be used to create realistic emulations of acoustic instruments - **digital recordings** are taken of individual notes which are combined into a compilation which evokes a useful range of musical materials to generate a new musical performance (different notes can be generated from a single recording based on playing it back faster or slower)

# RECAP: A PICTORIAL HISTORY

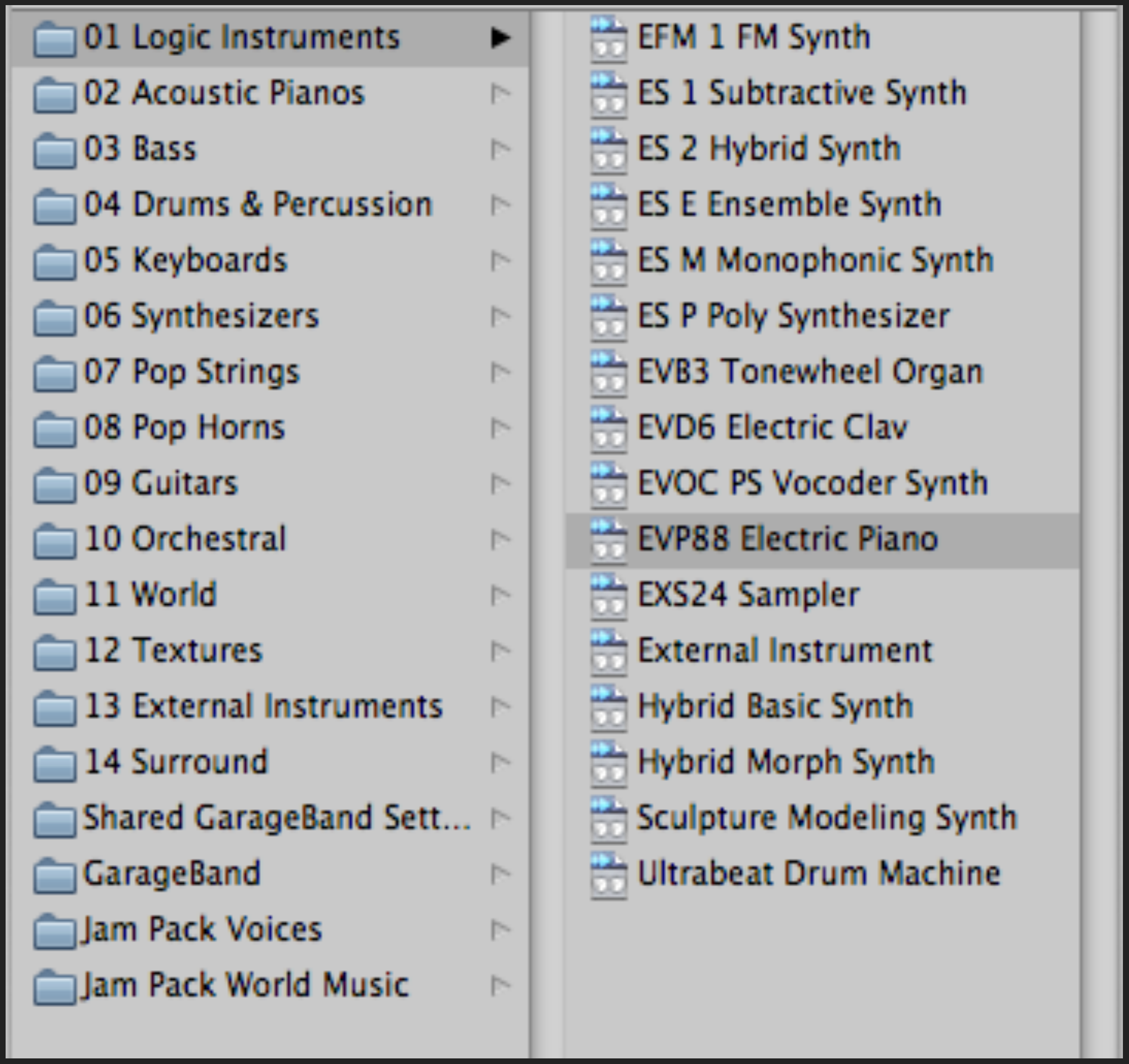


# SYNTHESIS IN LOGIC

- **Logic has a number of synthesisers built into its software environment**
- A **software synthesiser** is one which generate sound by means of a programme running on generic computer hardware, such as a personal computer
- This is in contrast to a **hardware synthesiser** (keyboard or sound module), which produces its sound using dedicated pieces of electronics (advantage - stability; disadvantage - lack of flexibility/expense)
- Software synthesisers are often called **plugins** because they are often separate from the music programme you are using, but are connected via the software equivalent of an audio lead and plug

# LOGIC'S MAIN SOFTWARE INSTRUMENTS/SYNTHESISERS

- **Subtractive:** ES1, ESP, ESM, ESP
- **Hybrid:** ES2 (subtractive plus wavetable synthesis)
- **Modulation:** EFM1
- **Physical Modelling:** Sculpture, single-instrument physical mods (EVP88, EVD6, EVB3)
- **Drum machine:** Ultrabeat (sampler plus synthesiser-based instrument components)



# FOCUS ON SUBTRACTIVE SYNTHESISERS

- *Subtractive synthesis: popularised by Moog in the 1960s and 1970s*
- **ES1:** Logic's most complex 'pure' subtractive synthesiser (can be monophonic or polyphonic)
- **ESM:** Monophonic subtractive synthesiser (useful for bass/lead sounds)
- **ESE:** 'Ensemble' synth--simple polyphonic subtractive synthesiser
- **ESP:** Polyphonic subtractive synthesiser (more complex than ESE, generally simpler than ES1, but somewhat distinct design choices)
- Retrosynth: New in Logic X: analogue, wavetable and FM