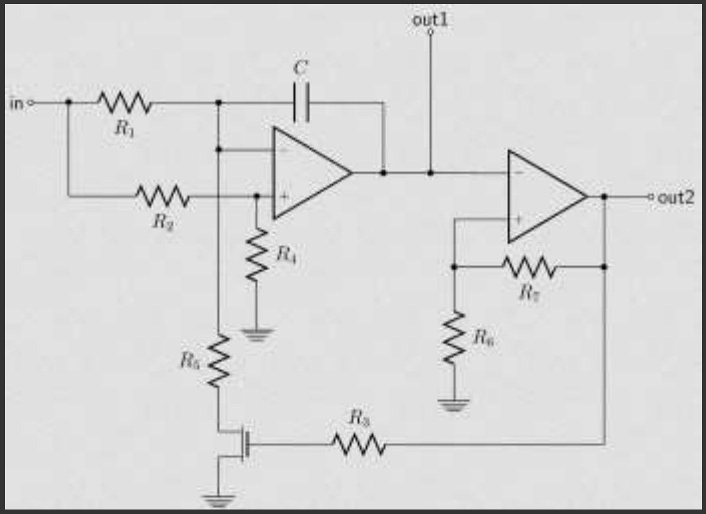
Electronic musical instruments

WAVETABLE SYNTHESIS

Digital generators

Analogue VCO generators

- Analogue circuits in synthesizers were imperfect, which attributed to their interesting, “warm” sound.
- The main problem was that oscillators were unstable, they often went out of tune.
- This happened because of thermal effect that changed properties of the analogue components.
- This was a huge problem in polyphonic synthesizers, as the voices became detuned.



Source: Wikipedia

Analogue VCO generators

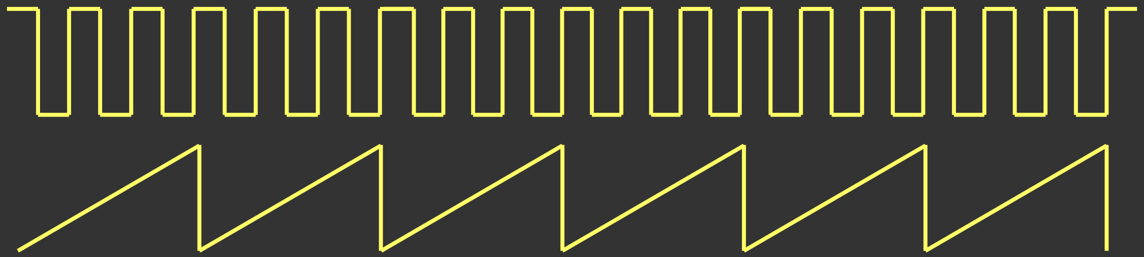
Basic concept of an analogue VCO

- A capacitor is charged with a current.
- A comparator detects the maximum charge value, discharges the capacitor and the cycle repeats.

- A sawtooth wave is obtained. Other wave shapes can be obtained by processing (e.g. integration).
- If the charging rate changes due to thermal effects, the period of the wave starts to change, and the pitch is altered – the oscillator is out of tune.

Digitally controlled oscillators

- DCO – *Digitally Controlled Oscillator*
- A comparator is replaced by a digital system:
  - a high frequency impulse train is generated,
  - a pulse counter discharges the capacitor.



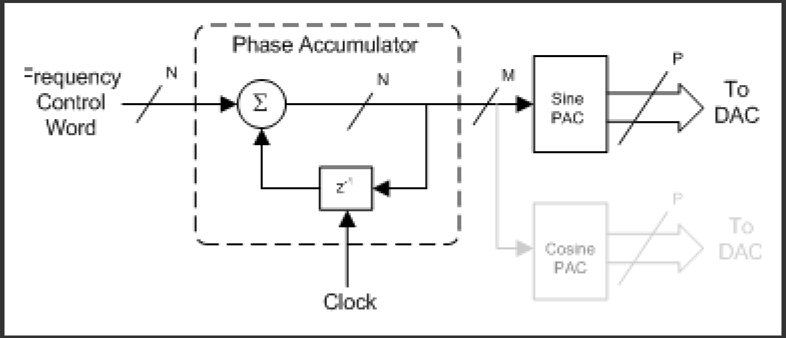
- High precision, high stability of the wave period.
- The remaining parts of the generator remain analogue (which is a good thing).

Digital generators

It is possible to generate waves with fully digital systems (*direct digital synthesizer*), also called DCO:

- pulse generator and counter determines frequency,
- phase accumulator creates a sawtooth wave,
- the wave is processed to obtain other wave shapes,
- digital-to analog converter creates an analogue signal.

Digital generators are stable, but we lose all imperfections of analogue oscillators: the sound becomes “cold”, too stable.



Źródło: Wikipedia

Digital generators

A phase accumulator generates a sawtooth wave by accumulating pulse amplitudes. Other wave shapes can be obtained by further processing:

- square wave:
  - – thresholding the sawtooth wave, or:
  - – adding a sawtooth to its shifted copy,
- triangle: integration (summing) the sawtooth wave,
- sine: with a phase-to-amplitude converter (using a look-up table); difficult to implement and sines are not particularly useful for the subtractive synthesis, so DCOs usually omitted sine generation.

Synthesizers with DCO

DCOs were used in many subtractive synthesizers in early 1980s (Roland, Korg, Akai, itp.).

Korg Poly-61 (1982)

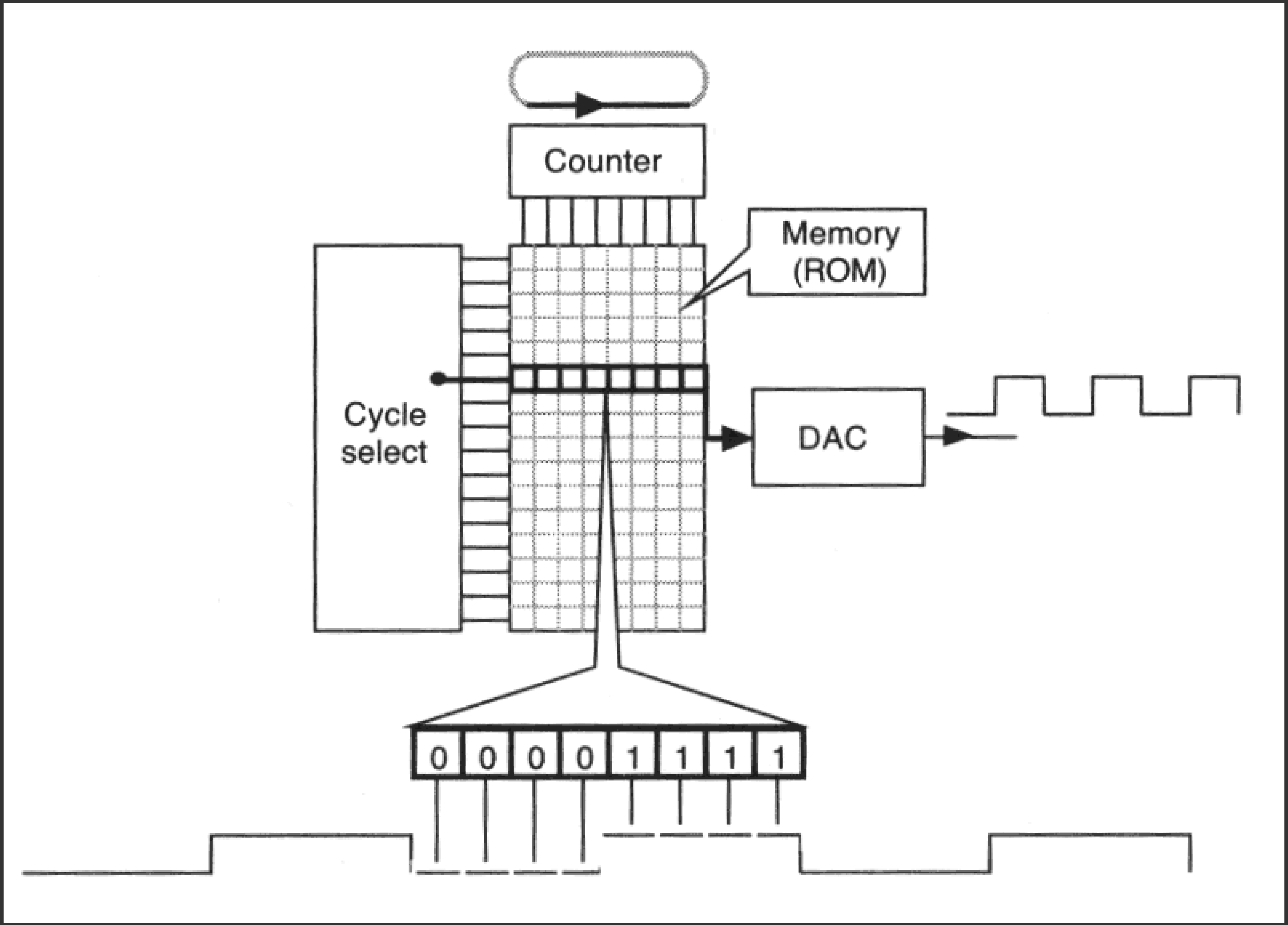
Roland Juno
6 / 60 / 106
(1982-84)

Memory (RAM) as a digital generator

A new approach to wave generation.

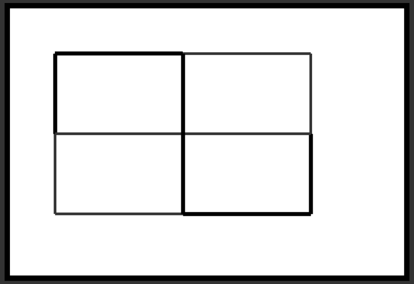
- Signals stored digitally in memory (RAM or ROM).
- A single period of each wave shape is stored.
- Signal generation by a looped reading of signal values.
- Any wave shape can be generated, not only the base shapes from analogue oscillators.
- However, new problems appear:
  - transposition – how to obtain different pitch,
  - aliasing – signal distortion by overlapped spectral components.

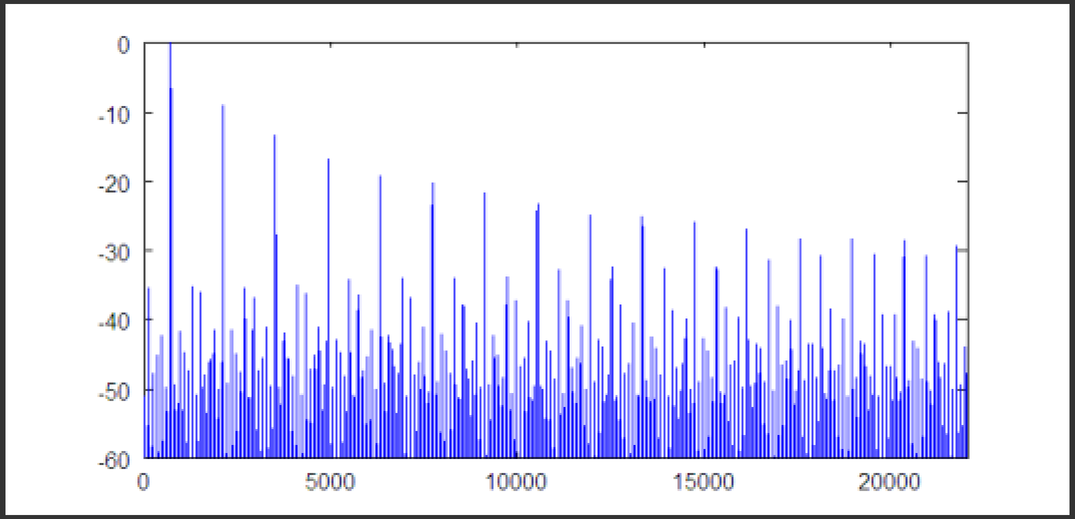
Memory (RAM) as a digital generator



The aliasing problem

- Signals stored in memory may be wideband.
- If the band exceeds the Nyquist frequency ($f_S/2$), aliasing happens – copies of the spectrum overlap.
- The sound becomes distorted because of inharmonic components that appear in the spectrum.
- We cannot generate a digital square wave like this:

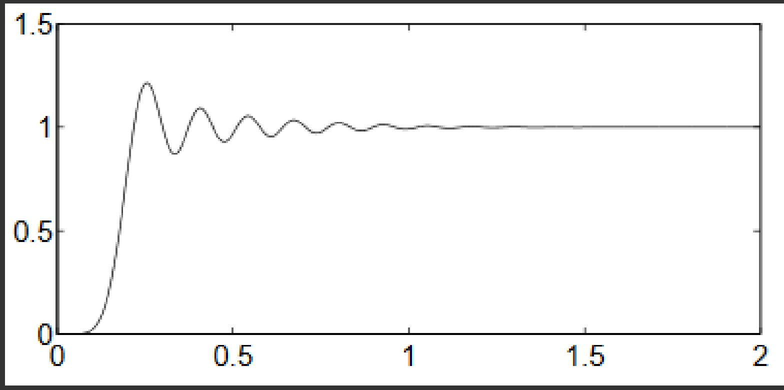




The aliasing problem

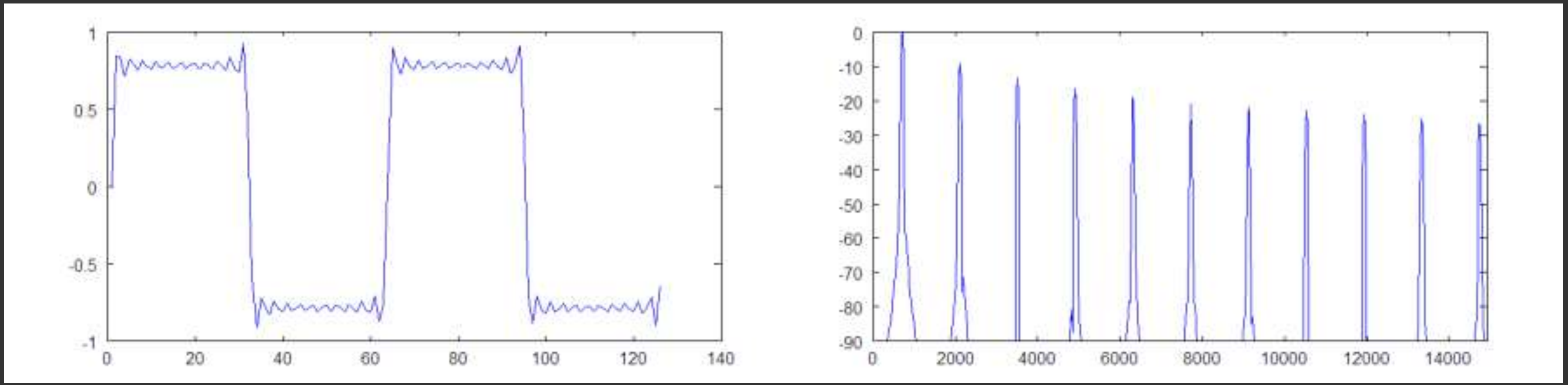
In order to avoid aliasing, signals stored in memory must be bandlimited. Some approaches are as follows.

- Fourier series – adding harmonic partials up to $f_S/2$ – requires complex computations.
- BLIT (*band limited impulse train*): a band-limited pulse train is generated and processed to obtain wave shapes.
- MinBLEPS – an aliased wave is generated, then a minimal phase pulse is inserted in wave sections representing rapid amplitude changes.



The aliasing problem

- We solved the problem of aliasing in the spectrum, but now the wave shape is distorted.
- This is caused by lack of high frequency components.
- The Gibbs effect: overshots for rapid amplitude changes and amplitude “ringing”.
- This is how a band-limited 1 kHz square wave looks like:



The transposition problem

- If we read signal samples with a constant rate, we obtain a single frequency (pitch) of the signal.
- *N* samples of a signal period are stored. Reading every sample with frequency f_S, we get a signal frequency:

  $$f = f_S / N,$$

  for example: f_S = 48 kHz, N = 1024: f = 46.875 Hz
- We need a different frequency for each note.
- In practice, we cannot store a separate signal for each frequency that we need.
- How to do a transposition, i.e. change the frequency of a signal stored in a memory?

The transposition problem

Method #1: varying rate of reading samples from memory (higher rate = higher frequency).

- This method was used in hybrid synthesizers: digital generator, D/A converter with a regulated rate and analogue processing modules.
- Transposition by altering the D/A conversion rate.
- For example: $N = 128$, we need $f = 440$ Hz $\rightarrow f_S = 56.32$ kHz.
- Problems:
  - – high rate of D/A conversion is needed to obtain high frequencies,
  - – a reconstruction filter tuned to $f_S / 2$ is needed.

The transposition problem

Method #2: varying step of reading samples from memory (larger step – higher frequency).

- This method is used in digital synthesizers.
- If we need a frequency f, we need to step the index by:

  $$s = f \cdot N / f_S$$

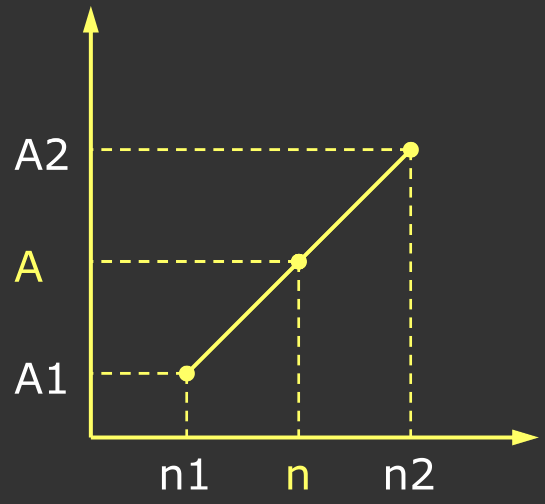
  e.g. f = 440 Hz, N = 1024 $\rightarrow$ s = 9.386 (for f_S = 48 kHz)

  f = 1 kHz, N = 1024 $\rightarrow$ s = 21.333
- Usually, the value s is not an integer.
- In order to read samples at “fractional positions”, we need to perform an interpolation.
- Interpolation distorts the signal. We need a sufficient number of signal samples per period.

Linear interpolation

- We need a sample at index *n* between *n1* and *n2* ($n2 - n1 = 1$)
- Samples stored in the memory: ($n1$, $A1$) and ($n2$, $A2$)
- We calculate (n, A) with linear interpolation:

$$A = A1 + (n - n1)(A2 - A1)$$



Transposition and aliasing

- Warning: if a signal is full-band and we transpose up:
  - the spectrum will stretch right,
  - aliasing occurs!
- Transposing down also may cause aliasing. Additionally, we lose high frequency components.
- It’s not enough to store a band-limited signal in memory. We also have to ensure that the transposition will not introduce aliasing.
- In commercial synthesizers from 1980s, aliasing was sometimes present in the sound.

Practical signal generation

A practical method of generating transposed waves by reading samples of wave period from memory:

- a separate set of samples for each octave, with a sufficient number of samples (e.g. 1024, 2048),
- frequency band limited to fs/4,
- transposition: only up, within an octave (up to fs/2),
- no aliasing is introduced,
- the problem: skipping between octaves may be audible, due to different bandwidth.

Practical signal generation (2)

We can allow some amount of aliasing:

- signal bandwidth limited to fs/3,
  for fs = 48 kHz: to 16 kHz,
- transposition: only up, within an octave,
- aliasing occurs above fs/3 (16 kHz), but it should be inaudible for most listeners, as it is masked by stronger signal components,
- we gain 4 kHz of bandwidth, timbre changes when skipping between octaves should be much less audible.

Source: http://www.earlevel.com/main/category/digital-audio/oscillators/wavetable-oscillators/

Wavetable synthesis

A wavetable synthesizer works as follows:

- wave shapes (single periods) are stored in memory,
- wavetable contains a set of (e.g. 60) wave shapes,
- waves in a table change smoothly, from simple shape (#0) to the most complex shape (#59),
- many different wavetables are available,
- signal is generated by a looped read of memory and conversion to an analogue signal,
- further processing by VCF and modulators LFO and EG, just like in a subtractive synthesizer.

The main stage of sound shaping takes place in the generator!

Wavetable synthesis

- Wave – a set of signal samples of a single wave period.
- Wavetable – a set of waves with a similar shape, with increasing harmonic content.
- Reading waves from tables:
  - one wave, looped,
  - a sum of several waves,
  - sweeping the table during sound generation: the wave index changes in time; this method is the main strength of the wavetable synthesis.

Wavetable

An example
of a wavetable
(60 wave versions
+ 4 base shapes)

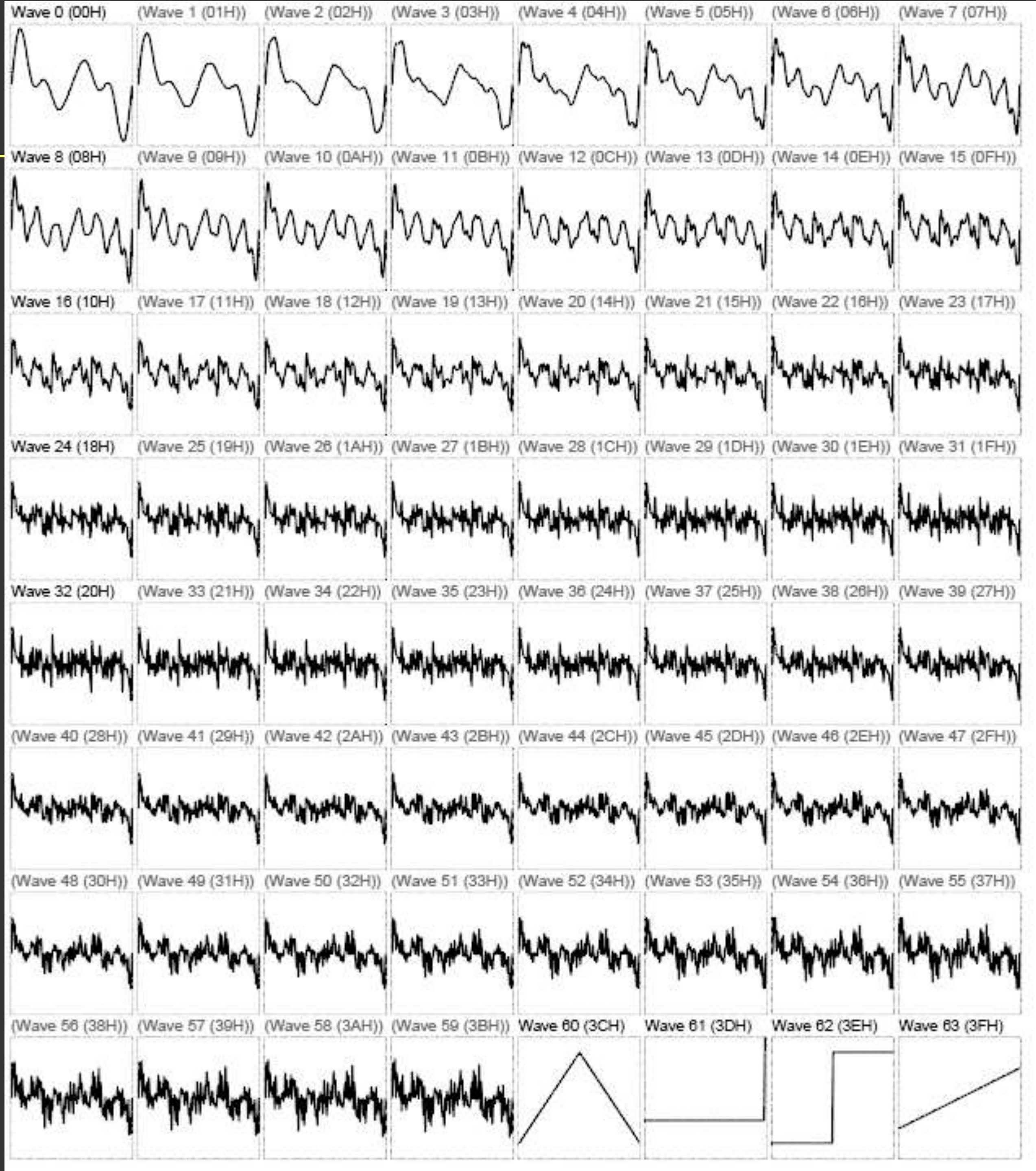


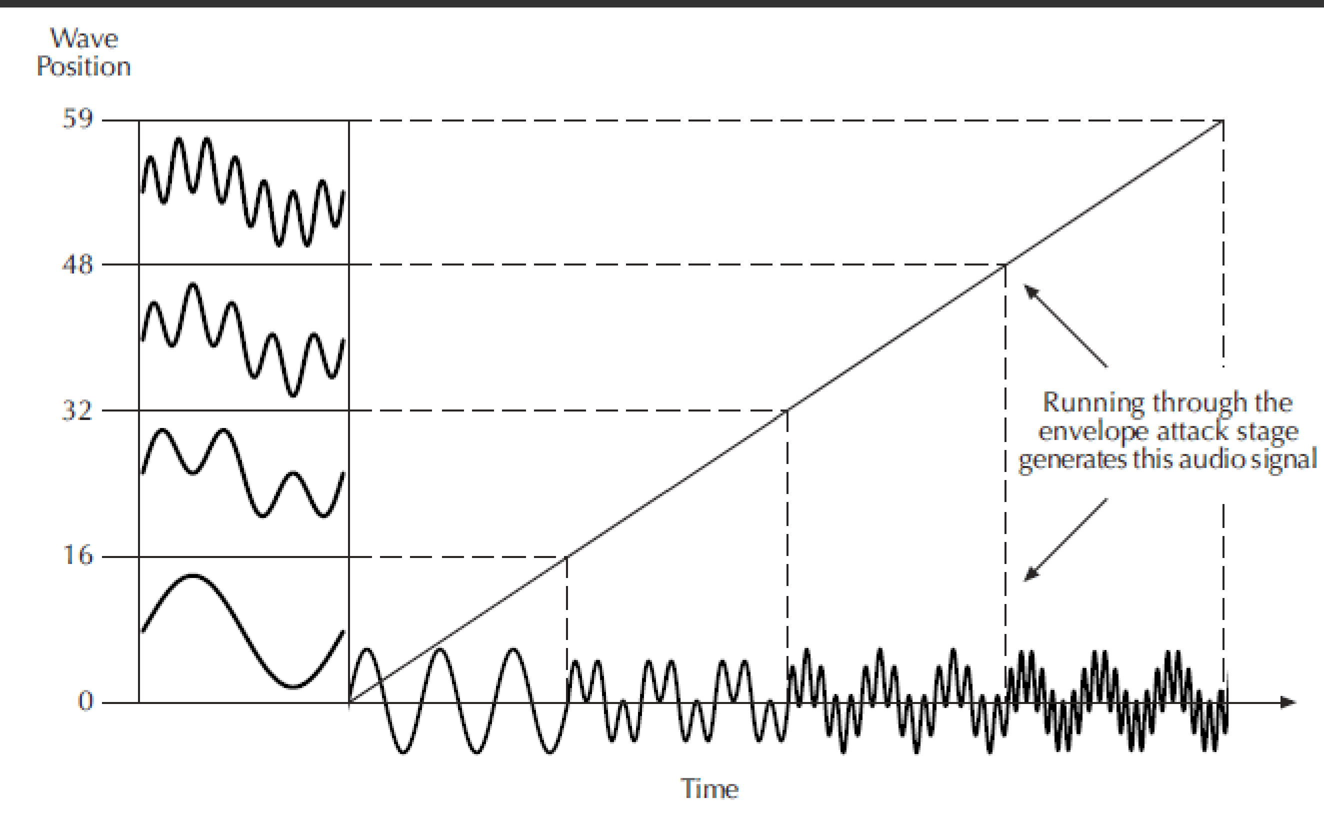
Table sweep

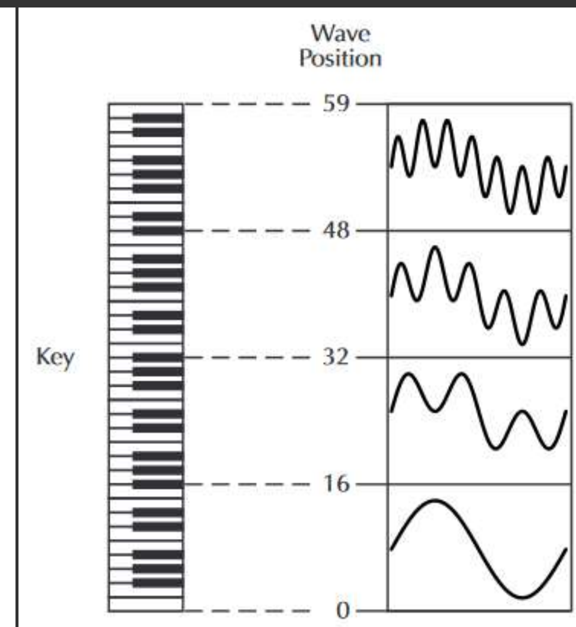
Modifying the index of the read wave:

- envelope generator:
  - envelope value determines the wave index,
  - timbre changes in the attack phase;
- LFO:
  - periodic modulation of the wave index,
  - timbre changes during the sustain phase,
- keyboard: more complex waves for lower frequencies, limits the aliasing,
- modulation wheel and similar controllers.

Table sweep

Wave index modulation with EG and keyboard





PPG instruments

Wavetable synthesis was used only in the PPG instruments, by Wolfgang Palm.

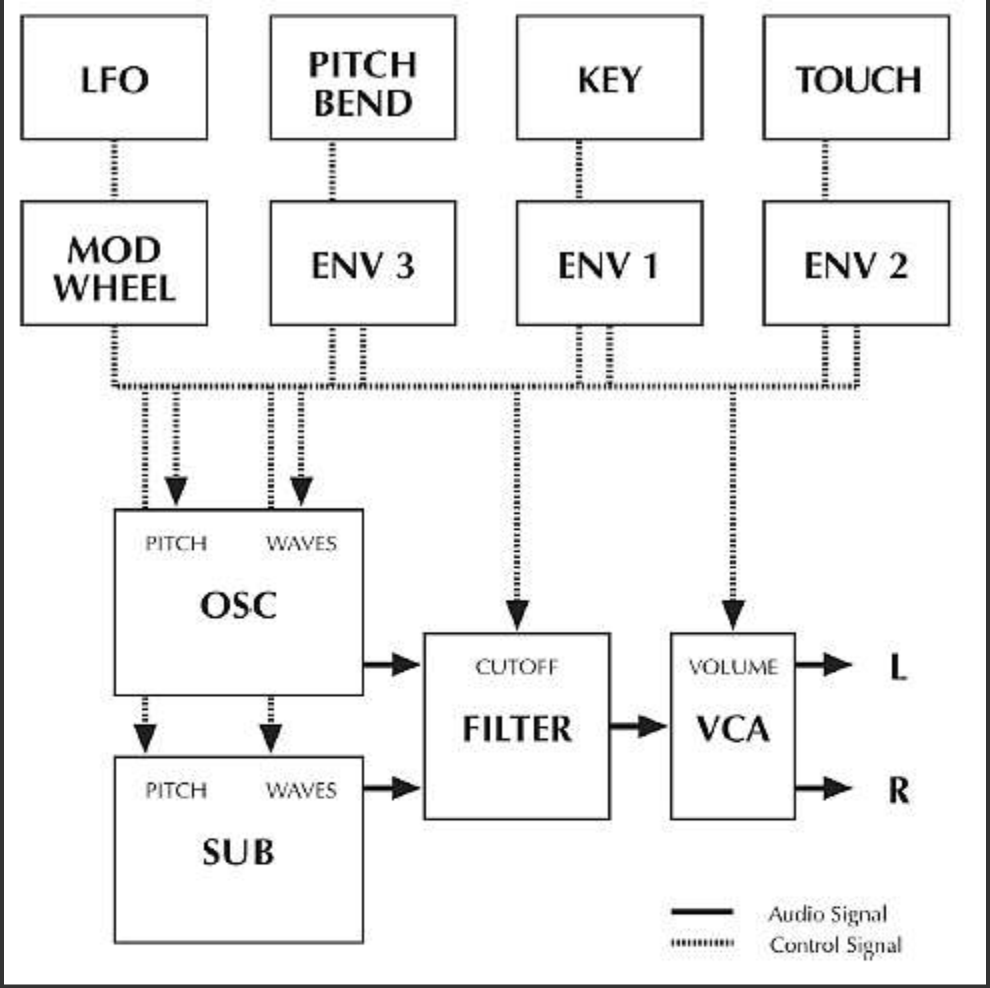
- *Wavecomputer 360* (1980) – first wavetable synthesizer.
- *Wave 2* (1981-87) – 30 tables, 64 waves in each table, a total of 1920 wave shapes, 8 voices, analogue VCF and VCO
- *Wave 2.2* and *Wave 2.3* – further development, MIDI, digital processing, sound samples.

PPG instruments

Waveterm:
a computer for
designing custom waves

PPG instruments

A block diagram of a PPG instrument



PPG – wavetable generators

- We can use:
  - one of 29 wavetables, each containing 60 wave versions and 4 base shapes (triangle, pulse, square, sawtooth),
  - an additional table (*upper wavetable*) - a set of 64 common wave shapes, always available.
- Two waves can be played simultaneously.
- Custom wave shapes may be designed.
- Waves can be created from short signal recordings, called *transients* (a simplified sampling).

Summary

Pros in comparison with the subtractive synthesis:

- more wave shapes available for generation,
- dynamic timbre changes in the generator,
- more possibilities for sound creation,
- stable pitch.

Cons:

- limited memory – only short signal periods,
- problematic transposition,
- aliasing is a problem,
- more stable, “cold” (but still interesting) sound.

Bibliography

- Martin Russ: *Sound Synthesis and Sampling*. Focal Press, Oxford 1996.
- Robert Bristow-Johnson: *Wavetable Synthesis. A Fundamental Perspective*. 101st AES Conv,. 1996.
- PPG Wave manual: http://www.hermannseib.com/english/synths/ppg/docs.htm
- Waldorf Wave 3.V (VST) manual http://tiny.pl/glhnm
- Wikipedia – waveshape synthesis
- PPG Wave Simulator (VST) – a free software emulator: http://www.hermannseib.com/english/synths/ppg/wavesim.htm
- Ear Level: Wavetable oscillator http://www.earlevel.com/main/category/digital-audio/oscillators/wavetable-oscillators/
- Tim Stilson, Julius Smith: *Alias-free digital synthesis of classic analog waveforms* (BLIT). https://ccrma.stanford.edu/~stilti/papers/blit.pdf
- Eli Brandt: *Hard sync without aliasing* (MinBLEPs). https://www.cs.cmu.edu/~eli/papers/icmc01-hardsync.pdf