

Session 5

Clock signals, sample & hold

VA309 Modular Sound Synthesis @ EKA
Aubery Lis

Informational / Organisational

2 weeks before Coursework 2 deadline

The earlier you start the better. Don't postpone till deadline day!

Recap of previous sessions

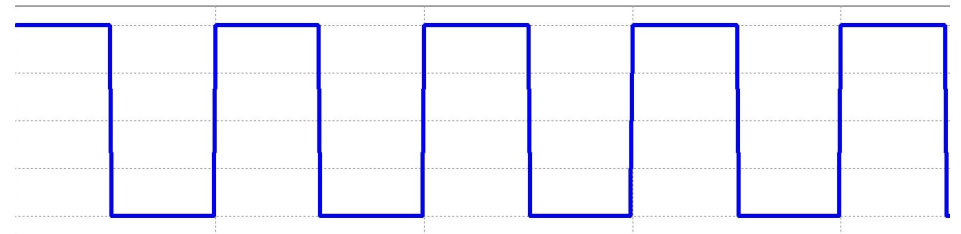
- What is an envelope? What types of envelope do you know?
- What is it useful for?

Recap of previous sessions

- What four sections does the most popular envelope have?
- What can we control about each section?

Clock signal

- Is a constantly repeating logic signal
- Sort of what would happen on the gate output of a keyboard if you hit the key repeatedly
- (but with much more accuracy)
- Is rather a concept than a particular module: a lot of modules can be clocks. This includes:
 - LFOs (square/pulse output)
 - VCOs (square/pulse output)
 - Specialized clock generator modules
 - Etc...



Clock signal

- Used to time musical events, like the keyboard gate output
- Runs on its own indefinitely
- Can be used as gate/trigger for an envelope, a clock input for a sequencer, a tempo sync input for some effect modules (like tempo-synced delays)...
- More often than not, are set to a particular speed, considered a “beat speed setter” and not interacted with throughout the performance (sad!)
- If the clock runs forever at a fixed speed, can we extract any other, more interesting rhythms/patterns out of it?

Clock division

- A special module called “**clock divider**” can be used to derive related lower tempo clocks from one “master” clock
- We **divide** the clock’s **frequency**: say, 4Hz (4 times per second), divided by 2 is 2Hz (2 times per second, → two times slower)
- A clock “divided by X” also means “pulses each X original clock pulses”
- Can be used to create intricate rhythmic patterns, especially when multiple outputs are used!



A-160-2 Controls

- **Gate/Trigger switch.** Gate = longer pulses for bigger divisions, Trig = equally short pulses on all outs (CST does nothing)
- **Division mode.** Up → left column, middle → center column, down → right column



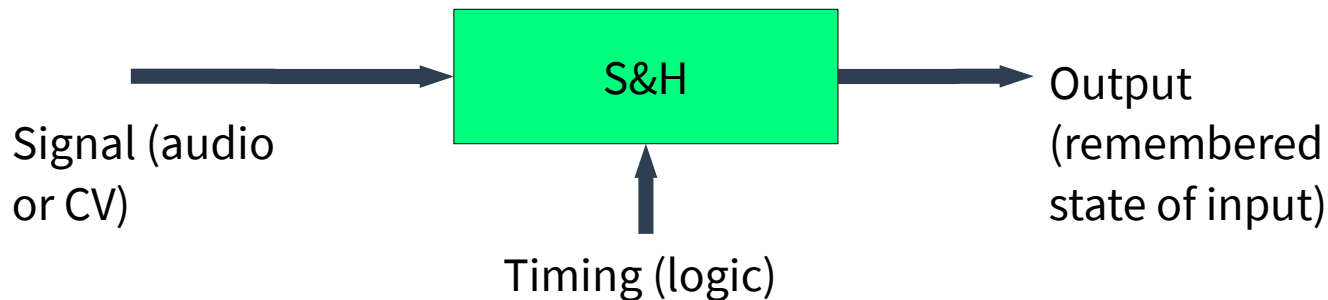
- **Clock input:** initial clock to get divided
- **Reset input:** drops all counters to 0, all the outputs turn on
- **Seven divided clock outputs:** each X times slower than the input clock (look at the number in the correct column next to the output)

Exercise!

- Take turns coming to the rack and setting up a division that I call
- The rest of the students listen and try to guess what did the source clock get divided by!

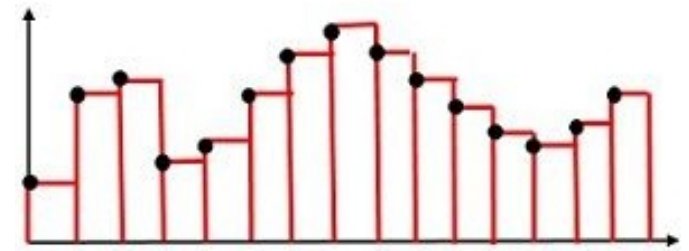
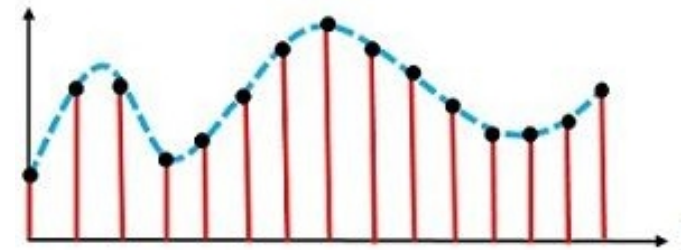
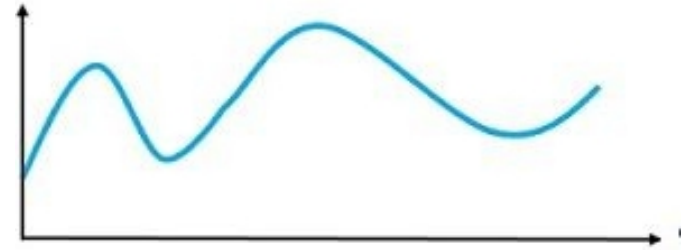
Sample & Hold

- A device for “remembering” a voltage
- A gate to the timing input tells S&H to **sample**:
 - 1) Look at the input voltage at the exact moment the gate was received
 - 2) Copy that voltage to the output
- Until the next gate is received, S&H **holds** (remembers) the voltage it just saw at the input



Sample & Hold

- Has many uses in music (and even more outside music!)
- Mostly used to chop a smooth continuous CV into a stepped granular one
- Clock signals are used as trigger inputs to sample a new voltage over and over continuously



[[link](#)]

A-148

- **Trig In:** logic signal input, low → high transition = “sample now!”
- **Smp. In** (sampling source in): source signal (CV or audio) to “remember” upon receiving a gate
- **S&H Out:** “remembered” voltage output, new voltage appears on each gate
- **Output indicator** to see what’s going on at the output

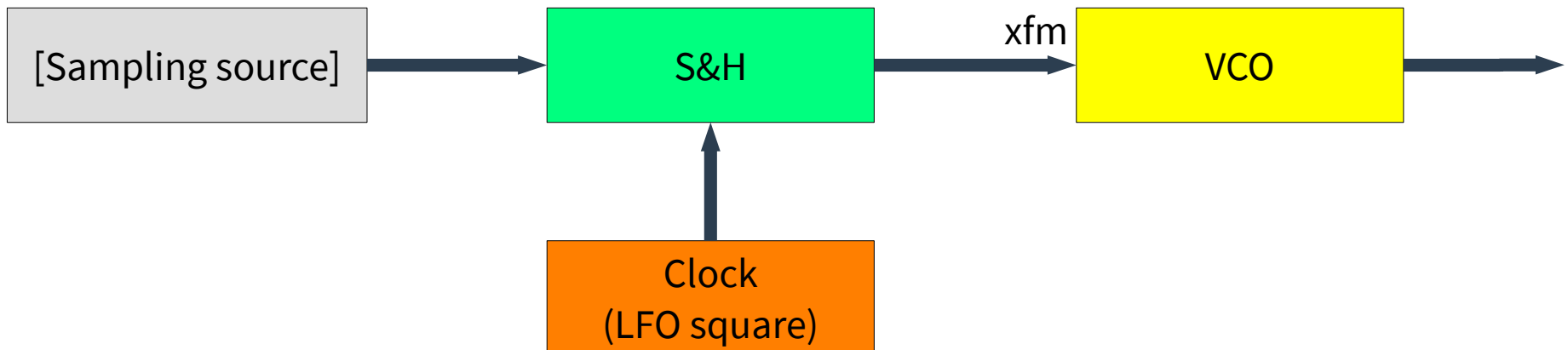


Using S&H to create randomness & chaos

- Chaos: an unpredictable phenomenon that can be described by some general tendencies
- Randomness: a completely unpredictable phenomenon with no tendencies to possibly figure out
- Both can be created using a S&H module

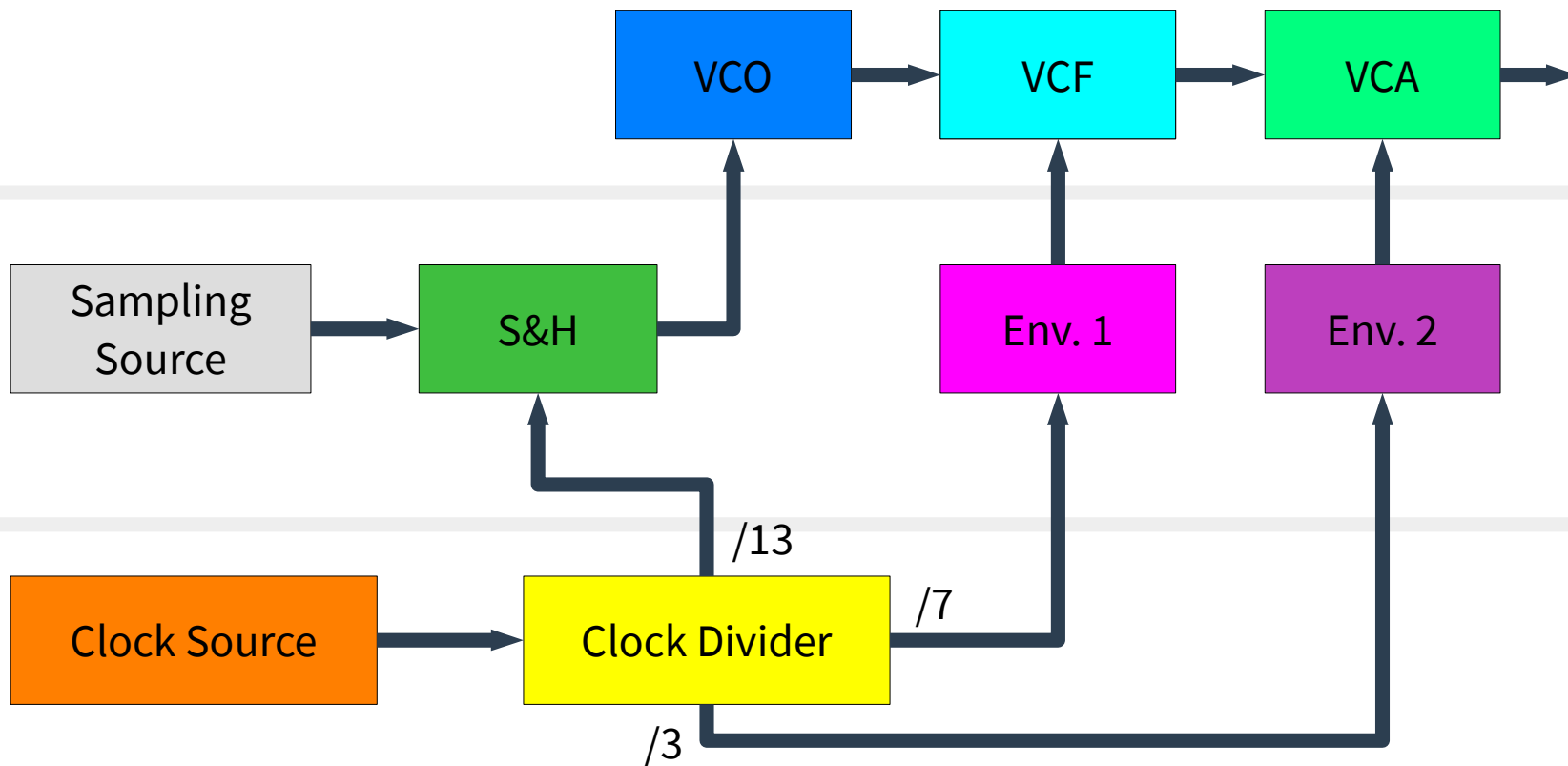
Using S&H to create randomness & chaos

- **White noise** for source: randomness! (you never know which voltage noise sits at exactly until you measure it)
- **Triangle LFO** for source: chaos! (the difference in clock speed and LFO speed create a peculiar pattern)



Exercise!

- Create a patch that uses a **clock divider** and a **S&H**



Simple S&H Demo

