**FPGA** ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *field-programmable gate array*) містять блоки множення-накопичення ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *multiply-accumulate, MAC*), які широко застосовуються при [цифровій обробці сигналів](https://uk.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0_%D0%BE%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B0_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB%D1%96%D0%B2" \o "Цифрова обробка сигналів) (DSP), а також логічні елементи (як правило на базі таблиць перекодування — таблиць істинності) та блоки їх комутації. FPGA зазвичай використовуються для обробки сигналів, мають більше логічних елементів і гнучкішу архітектуру, ніж CPLD. Програма для FPGA зберігається в розподіленій пам'яті, яка може бути виконана як на основі енергозалежних осередків статичного ОЗП (подібні мікросхеми виробляють, наприклад, фірми Xilinx і Altera) — у цьому випадку програма не зберігається при зникненні електроживлення мікросхеми, так і на основі енергонезалежних комірок Flash-пам'яті або перемичок antifuse (такі мікросхеми виробляють фірми Microsemi й Lattice Semiconductor) — в цих випадках програма зберігається при зникненні електроживлення. Якщо програма зберігається в енергозалежній пам'яті, то при кожному ввімкненні живлення мікросхеми її необхідно заново конфігурувати за допомогою початкового завантажувача, який може бути вбудовано і в саму FPGA. Альтернативою ПЛІС FPGA є повільніші цифрові процесори обробки сигналів. FPGA застосовуються також, як прискорювачі [універсальних процесорів](https://uk.wikipedia.org/wiki/CPU" \o "CPU) в суперкомп'ютерах (наприклад: Cray — XD1, SGI — Проект RASC).

**Програмована логічна інтегральна схема, ПЛІС** ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *programmable logic device, PLD*) — електронний компонент, що використовується для створення цифрових інтегральних схем. На відміну від звичайних цифрових [мікросхем](https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%BA%D1%80%D0%BE%D1%81%D1%85%D0%B5%D0%BC%D0%B0" \o "Мікросхема), логіка роботи ПЛІС не визначається при виготовленні, а задається за допомогою [програмування](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F" \o "Програмування). Для цього використовуються програматори і налагоджувальні середовища, що дозволяють задати бажану структуру цифрового пристрою у вигляді принципової електричної схеми або програми на спеціальних [мовах опису апаратури](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B8_%D0%BE%D0%BF%D0%B8%D1%81%D1%83_%D0%B0%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D1%83%D1%80%D0%B8" \o "Мови опису апаратури) ([Verilog](https://uk.wikipedia.org/wiki/Verilog" \o "Verilog), [VHDL](https://uk.wikipedia.org/wiki/VHDL), [AHDL](https://uk.wikipedia.org/wiki/AHDL) та інші). Альтернативою ПЛІС є: [програмований логічний контролер](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B9_%D0%BB%D0%BE%D0%B3%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80" \o "Програмований логічний контролер), [базові матричні кристали](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%BE%D0%B2%D0%B8%D0%B9_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%B8%D0%B9_%D0%BA%D1%80%D0%B8%D1%81%D1%82%D0%B0%D0%BB" \o "Базовий матричний кристал), що вимагають заводського виробничого процесу для програмування; [ASIC](https://uk.wikipedia.org/wiki/ASIC) — спеціалізовані замовні ВІС (великі інтегральні схеми), які при малосерійному та одиничному виробництві істотно дорожчі; спеціалізовані [комп'ютери](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80" \o "Комп'ютер), [процесори](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80" \o "Процесор) (наприклад, [цифровий сигнальний процесор](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80_%D1%86%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B8%D1%85_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB%D1%96%D0%B2" \o "Процесор цифрових сигналів)) або [мікроконтролери](https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%BA%D1%80%D0%BE%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80" \o "Мікроконтролер), які через програмний спосіб реалізації алгоритмів повільніші, ніж ПЛІС.

Деякі виробники ПЛІС пропонують програмні процесори, які можуть бути модифіковані під конкретне завдання, а потім вбудовані в ПЛІС. Тим самим забезпечується зменшення використання місця на друкованій платі і спрощення проектування самої ПЛІС.

уществует множество различных методов синтеза звука, однако все их можно свести к нескольким основным типам:

* [аддитивный](http://www.digitalmusicacademy.ru/lesson-additive-synthesis)
* [субтрактивный](http://www.digitalmusicacademy.ru/lesson-subtractive-synthesis)
* [сэмплирование](http://www.digitalmusicacademy.ru/lesson-sample-and-synthesis)
* [FM-синтез](http://www.digitalmusicacademy.ru/lesson-fm-synthesis) и [PD-синтез](http://www.digitalmusicacademy.ru/lesson-pd-synthesis)
* [физическое моделирование](http://www.digitalmusicacademy.ru/lesson-physical-modelling)

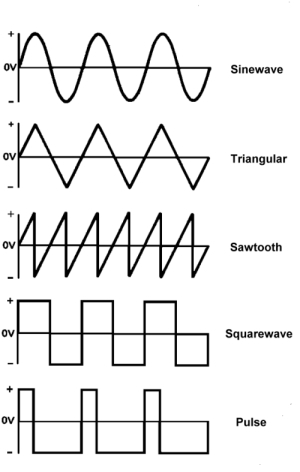
Кроме того, были разработаны их усовершенствованные варианты и многочисленные гибридные технологии, например:

* [wavetable-синтез](http://www.digitalmusicacademy.ru/lesson-sample-and-synthesis#wt)
* [гранулярный синтез](http://www.digitalmusicacademy.ru/lesson-sample-and-synthesis#gran)
* [линейно-арифметический синтез](http://www.digitalmusicacademy.ru/lesson-sample-and-synthesis#la)
* [векторный синтез](http://www.digitalmusicacademy.ru/lesson-vector-synthesis)
* [аналоговое моделирование](http://www.digitalmusicacademy.ru/lesson-physical-modelling#analog)
* [формантный синтез](http://www.digitalmusicacademy.ru/lesson-vocoder#formant)

и т.д.

Формы волны

В синтезаторах в качестве исходных сигналов используются колебания различной формы: синусоидальные (гармонические), квадратные, пилообразные, треугольные и импульсные. Использование именно таких колебаний обусловлено возможностями осцилляторов — первые ламповые осцилляторы могли генерировать только синусоидальные колебания, позже появилась возможность получать и другие (особенно с появлением транзисторов в 60-е гг.):



## **Синусоидальные (гармонические) колебания**

Синусоидальное колебание иногда называют «простым», потому что из таких колебаний, как из элементов, складываются все более сложные виды колебаний (cогласно физиологической [теории слуха Гельмгольца](http://www.digitalmusicacademy.ru/lesson-helmholtzs-hearing-theory) и математической [теореме Фурье](http://www.digitalmusicacademy.ru/lesson-fft), любое звуковое колебание является результатом сложения синусоидальных колебаний). Форма волны описывается, например, функцией y = sin(x). Такие колебания также называют гармоническими:

## **Квадратные (Square) колебания**

Звучание «квадратной» волны обычно описывается как «пустое», такие тембры используются для имитации басовых звуков и акустических духовых инструментов.

Спектр такой волны содержит только нечетные гармоники. Причем, амплитуда каждой гармоники обратно пропорциональна её частоте (относительно основного тона). Например, основной тон 100 Hz будет иметь гармонику с частотой 300 Hz (100\*3) и амплитудой, равной 1/3 амплитуды основного тона:

## **Пилообразные (Sawtooth) колебания**

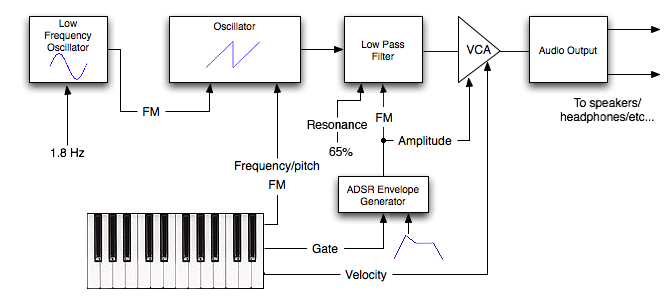
На основе таких волн обычно построены тембры струнных смычковых акустических инструментов. Характерный для них тип движения струны называют «stick-slip motion»:

## **Треугольные (Triangle) колебания**

Как и квадратная, треугольная волна содержит только нечетные гармоники, однако крутизна спада амплитуды гармоник здесь больше. Например, основной тон 100 Hz будет иметь первую гармонику с частотой 300 Hz (100\*3) и амплитудой, равной 0.1111 (1/3^2) амплитуды основного тона.

FM-синтез

Частотная модуляция или FM-синтез (Frequency Modulation, FM-synthesis) — это тип синтеза, при котором звуковое колебание формируется воздействием одной простой синусоидальной волны (модулирующего сигнала) на частоту другой волны (несущего сигнала) .  С помощью частотной модуляции можно получить спектрально богатые и сложные звуки, которых трудно добиться другими методами синтеза.



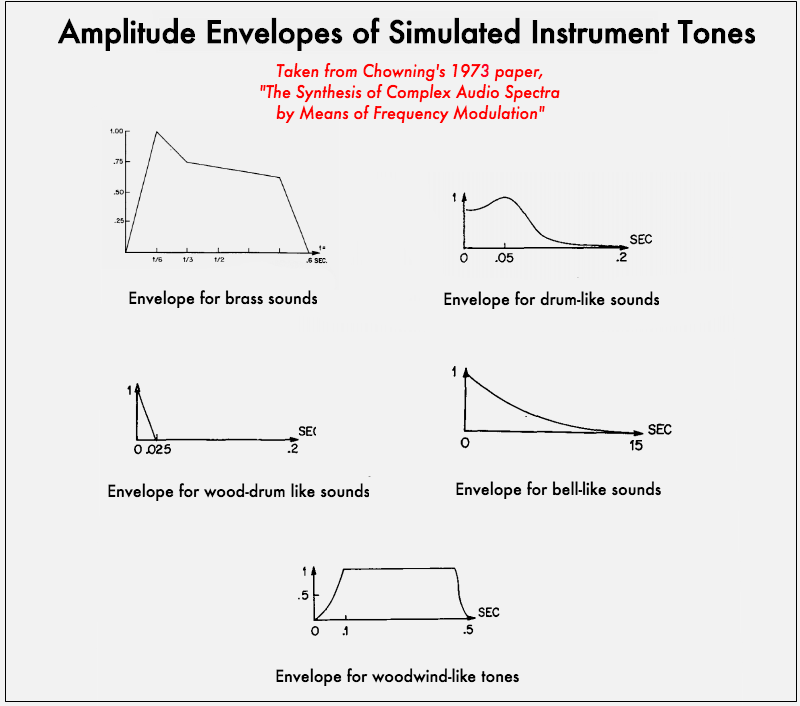
# Осциллятор (VCO)

В классическом синтезаторе звук генерируется осцилляторами (VCO — voltage-controlled oscillator). Это устройства, **непрерывно**генерирующие периодические колебания определенной формы:

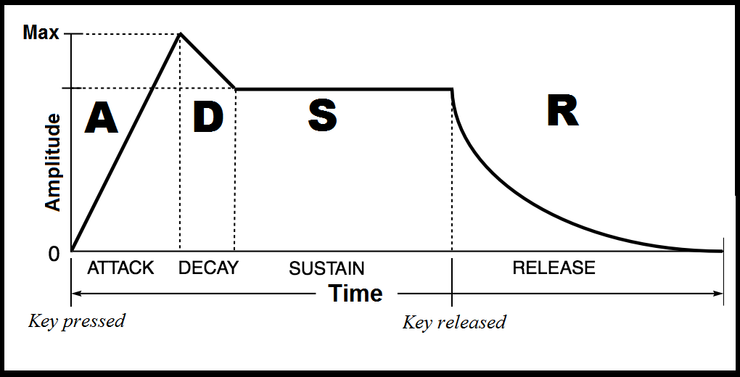
* синусоидальные
* квадратные
* треугольные
* пилообразные
* импульсные

# ADSR-огибающая (Envelope Generator)

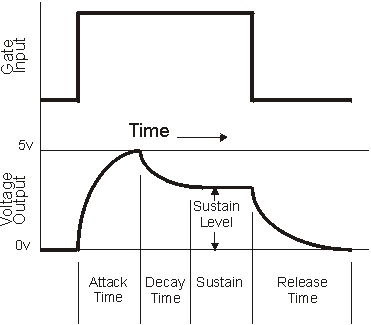
Динамическая огибающая является одной из важнейших характеристик звука, особенно музыкального, поэтому необходимо иметь возможность ей управлять:



Саму концепцию «огибающей» предложил в 1965 году [Владимир Усачевский](http://www.digitalmusicacademy.ru/lesson-academic-studios), обозначавший динамические стадии звука как T1, T2, Esus, T3, а её упрощённая и устоявшаяся в наши дни аббревиатура – заслуга [компании ARP](http://www.digitalmusicacademy.ru/arp-synthesizers):



Таким образом, функция модуля **ADSR**(или в более общем случае —  Envelope Generator) — преобразовать прямоугольный Gate-сигнал, который генерируется нажатием и снятием клавиши на клавиатуре синтезатора (или секвенсором) в нечто более естественное:



Основные фазы, на которые обычно разбивается Gate-сигнал:  
**A** — (attack) — начальная фаза, подъем  
**D** — (decay) — фаза перехода звука в установившееся состояние  
**S** — (sustain) — «удерживание», основная фаза, которая соответствует продолжительности нажатия клавиши  
**R** — (release) — затухание

Даний дипломний проект являє собою розробку схеми синтезатора, огляд основних методів цифрового синтезу сигналу, технологій та алгоритмів ЦОС( цифрової обробки сигналів), аргументацію обраних рішень та технологій.

Основною функцією синтезатора є генерування адреси комірки пам’яті, де зберігаються числові значення синусоїдального сигналу. Інші ж блоки фільтрують та оброблюють данні числові значення в залежності від параметрів.

Основні функціональні частини генеруючого елементу:

1. Oscillator (осцилятор) – даний блок являє собою накопичувальний суматор. Старші біти суматора є адресом комірки пам’яті. В залежності від необхідної якості сигналу розмір пам’яті збільшується або зменшується, від чого залежить довжина адреси.
2. Пам’ять – комірки пам’яті зі значеннями синусоїдального сигналу.
3. VCA – voltage controlled amplifier – даний блок реалізується шляхом множення вхідного сигналу на коефіцієнт. Коефіцієнт множення керується LFO або ADSR обвідною.
4. ADSR – обвідна – даний блок реалізується на накопичувальному суматорі. В залежності від поточного часу видає на VCA потрібний коефіцієнт у форматі числа з плаваючою комою.
5. Timer – рахує час у тактах від приходу сигналу (натискання клавіші).

Використання ПЛІС надає можливість спроектувати модулі, що будуть виконувати обробку паралельно. Це звільняє нас від непотрібних часових затримок, що є дуже критичними в цій сфері обладнання.

Використання мікропроцесорних систем є недопустимим. За приклад візьмемо задачу генерації одночасно двох значень синуса та їх подальше сумування. Мікропроцесор виконає опитування однієї клавіші, читання з пам’яті значення синуса, далі опитування другої клавіші, читання з пам’яті другого значення синуса, сумування двох отриманих значень, та перехід на наступну ітерацію. Ця послідовність дій призведе до досить великих затримок між ітераціями, враховуючи досить малу частоту роботи сучасних мікропроцесорних систем. ПЛІС же дозволяє створити модулі, що будуть працювати паралельно і незалежно на дуже високих частотах.

Саме незалежність та паралельна робота модулів призвела до вибору саме ПЛІС для даного проекту.

ПЛІС дозволяє на одному кристалі, в одному корпусі, розмістити і схеми обміну даними з ПК, і модулі генерації сигналів, та їх обробку. Що звільняє від проектування друкованих плат з багатьма мікро чіпами та периферії.

**Архітектура ПЛІС на прикладі Intel Cyclone IV**

Логічні елементи(LЕs) є найменшими блоками в архітектурі ПЛІС. Логічний елемент є компактним і забезпечує розширені можливості з ефективною логікою. Кожен ЛЕ має :

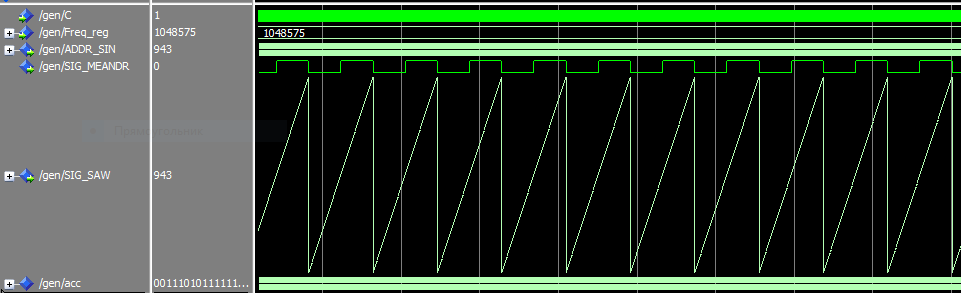
* LUT. Чотирьох входовий блок, що реалізує функцію за таблицею істинності.
* Програмований регістр
* Вихід регістру
* Вихід функції LUT

Логічні елементи складаються у LAB – масив логіки (logic array block). Всі логічні елементи всередині одного LAB мають спільні сигнали тактування, скидання та сигнал дозволу тактування.

Кожна ПЛІС(FPGA) Cyclone також має вбудовані блоки пам’яті M9K, блоки множення на 9 біт, PLL.

**Структура блока синтезу**

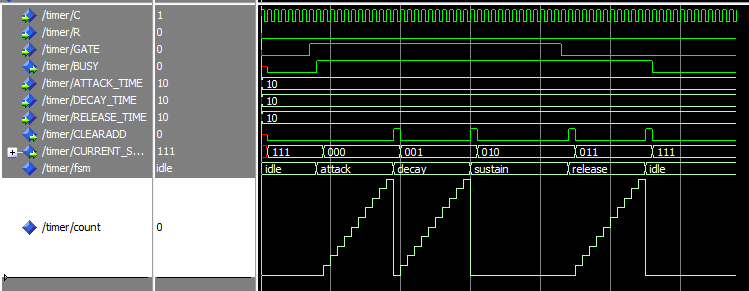
1. Gen



1. CONV\_INT\_TO\_FLOAT



1. TIMER



1. CONV\_FLOAT\_TO\_INT

