

## ЗМІСТ

ВСТУП .....	4
РОЗДІЛ 1	
ТЕОРЕТИЧНІ ОСНОВИ ЗВУКУ ТА МУЗИКИ.....	5
1.1. Звук як фізичне явище .....	5
1.2. Спектральний склад звуку .....	6
1.3. Цифрове представлення звуку .....	7
1.4. Музичні терміни та правила.....	12
1.5. Методи та основи цифрового синтезу.....	14
1.6. Форми хвиль, що використовуються у музиці .....	15
1.7. Архітектура синтезатора.....	16
1.7.1. Осцилятор .....	16
1.7.2. Зміна гучності.....	16
1.7.3. Фільтр .....	16
1.7.4. Генератор обвідної хвилі .....	17
1.7.5. LFO. Низькочастотний осцилятор.....	18
ВИСНОВКИ ДО РОЗДІЛУ 1 .....	19

					ІАЛЦ.466454.002 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розробив		Антонов Є. А.			Цифровий синтезатор на основі ПЛІС  Пояснювальна записка	Літ.	Аркуш	Аркушів
Перевірив		Стешин В. В.					1	59
Реценз.						НТУУ КПІ, ФІОТ, ІО-51		
Н. Контр.		Сімоненко В. П.						
Затвердив								

## РОЗДІЛ 2

АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ДЛЯ ІНТЕГРУВАННЯ ТА МОЖЛИВИХ РЕАЛІЗАЦІЙ ПРОЕКТУ .....	20
2.1. Системи на мікроконтролері .....	20
2.2. Програмний додаток на ПК(синтез на ЦП) .....	22
2.3. Схема синтезатора складена з окремих мікро чипів, пам'яті, периферії .....	22
2.4. ПЛІС(FPGA).....	24
ВИСНОВКИ ДО РОЗДІЛУ 2 .....	33

## РОЗДІЛ 3

СХЕМА СИНТЕЗУ ЗВУКУ. ОСНОВНІ ЧАСТИНИ СПРОЕКТОВАНОГО СИНТЕЗАТОРА .....	34
3.1 GEN .....	34
3.2 OSCILL .....	36
3.3 CONV_INT_TO_FLOAT .....	37
3.4 CONV_FLOAT_TO_INT .....	38
3.5 TIMER.....	39
3.6 MUL .....	40
3.7 ADSR.....	41
3.8 BLOCK.....	42
3.9 ADD.....	43
3.10 UART_RX.....	44
3.11 BUTTON_PROC.....	44
3.12 UART_PROC .....	45
3.13 ЦАП ЗА СХЕМОЮ R2R.....	46

3.14 РЕЗУЛЬТАТИ СИМУЛЯЦІЙ .....	46
ВИСНОВКИ ДО РОЗДІЛУ 3 .....	52
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	54
Додаток А.....	56
Додаток Б .....	57
Додаток В .....	58
Додаток Г .....	59

					ІАЛЦ.466454.003 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Після купівлі компанією Intel одного з найбільших виробників ПЛІС компанії Altera, розвиток технологій та тенденцій пішов угору. Через це все більше компаній використовують сучасні ПЛІС для прискорення та поліпшення своїх проєктів. Сьогодні ПЛІС може реалізувати як мікроконтролер для управління роботом так і надшвидку нейронну мережу або прискорювач для центрального процесора. Все частіше будь-яка обробка відео потоку чи швидкісного радіо потоку асоціюється з ПЛІС, але можливості ПЛІС ще не використовують виробники аудіоапаратури. Альтернативі ПЛІС може будуть спеціалізовані схеми на замовлення, але при малосерійному виробництві це не є вигідним, або процесори, які через програмний метод реалізації алгоритмів є повільнішими за ПЛІС. Іншою відмінністю ПЛІС окрім швидкості є енергоефективність, що може стати великою перевагою при розробці малих та мобільних проєктів. Як і відео аудіо потоки є досить великими та вимагають відсутності часових затримок. Через те що програмні методи обробки є досить повільними і не можуть використовуватися у real-time рішеннях, вибір падає саме на апаратні методи, що може реалізувати ПЛІС.

Саме через ці фактори було вирішено розробити цифровий синтезатор саме на базі ПЛІС. Метою проєкту є розробити схему генерації стандартних форм сигналів, та подальші блоки обробки сигналу, такі як помножувач на коефіцієнт, суматор, блоки управління та настройки параметрів та стандартний інтерфейс для спілкування з ПК, а саме UART.

					ІАЛЦ.466454.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

# РОЗДІЛ 1

## ТЕОРЕТИЧНІ ОСНОВИ ЗВУКУ ТА МУЗИКИ

### 1.1. Звук як фізичне явище

Як фізичне явище, звук представляє собою механічне коливання у деякому середовищі. Як і будь-яка інша хвиля, звук характеризується амплітудою, частотою, тиском та складом [3].

Частота або ж висота звуку – фізична величина, що характеризує кількість повторів певної події за 1 секунду часу. Але через те, що звук є складним процесом і найчастіше являє собою суму кількох гармонік, то висотою звуку називають частоту найбільш помітної гармоніки(основного тону).

Амплітуда – фізична величина, що характеризує найбільше зміщення сигналу, що періодично змінюється. Гучність залежить не лише від амплітуди, а й від частоти та спектрального складу [10,11].

Гучність звуку є відносною величиною і дорівнює рівню звукового тиску, створюваного синусоїдальним тоном з частотою 1кГц. Тобто частоти з однаковою амплітудою створюють різним за рівнем тиск в залежності від частоти. На рисунку 1.1 наведено графік рівня звукового тиску.

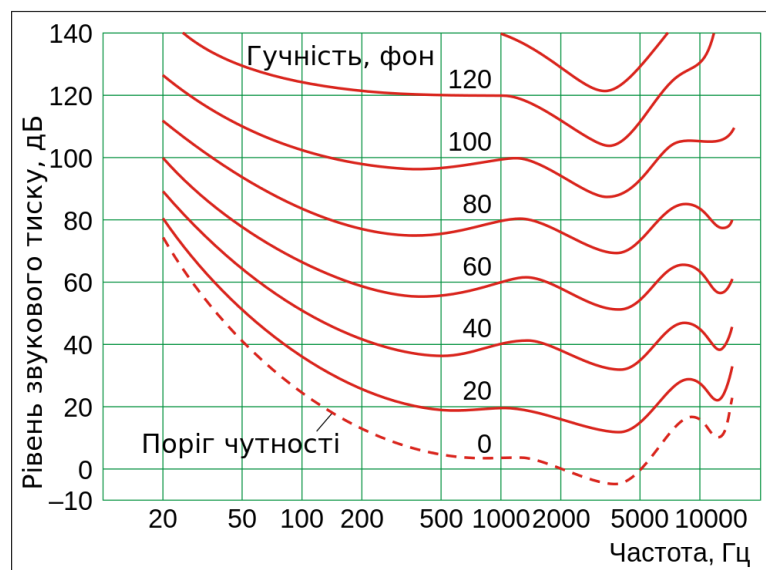


Рисунок 1.1 Криві рівнів гучності

## 1.2. Спектральний склад звуку

Отже будь-який звук, як в музиці так і в природі, є складним звуком, тобто складається з багатьох гармонік різної частоти та амплітуди.

$$Y(t) = \sum_{i=1}^n A_i * \sin(\omega_i + \varphi), \quad (1.1)$$

де  $A$  – амплітуда, Дб;

$\omega$  – частота, Гц;

$\varphi$  – фаза, рад.

Але при цьому можна виділити основний фон з найбільшою амплітудою. Інші з гармоніки називаються обертонами. Так для кожного інструменту є приклади його спектрального складу або ж як кажуть у музиці – тембру. Так як звук музичних інструментів є періодичним, то будь-який тембр можна виразити сумою гармонік з різними параметрами. На рисунку 1.2 наведено тембри декількох інструментів [4].

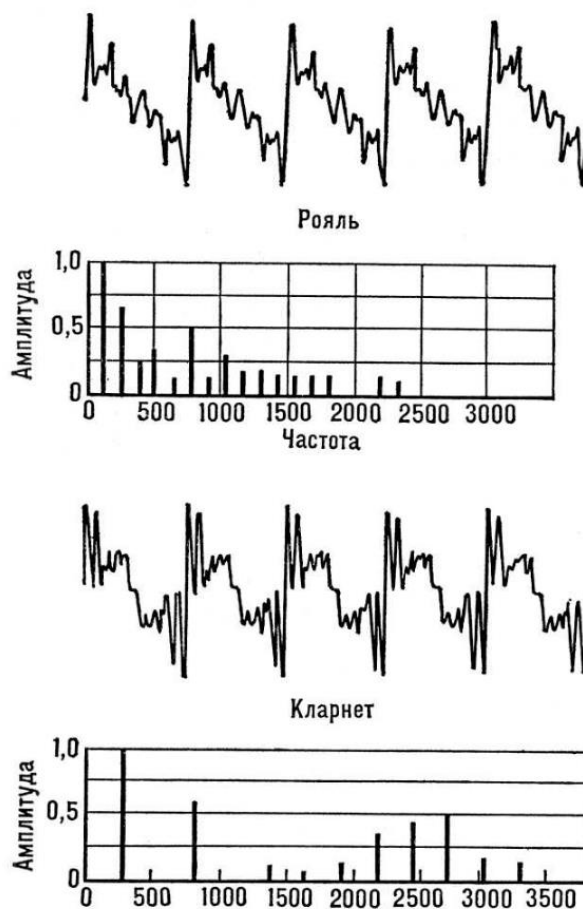


Рисунок 1.2 Тембр

Цифрові синтезатори не можуть в повній мірі зімітувати гру справжнього інструменту, але від них цього і не вимагають, так як очікують отримати як раз неординарний звук. Але все ж сучасні синтезатори можуть мати до 5 осциляторів, що генерують тон та 4 додаткові обертони. Як правило частота кожного наступного обертона у два рази більша попереднього. Тобто ця операція не вимагатиме багато ресурсів для множення, а буде реалізована простим зсувом числа.

### 1.3. Цифрове представлення звуку

Як відомо, аналоговий сигнал є неперервним, тобто він не має якихось певних відліків його значення, на відміну від дискретного який власне і є основою цифрової техніки. Через це дуже важливою особливістю сигналу є його квантування. В залежності від обраної розрядності числа, у яке буде переведено аналоговий сигнал, залежить з якою похибкою цифрове представлення буде відрізнятися від аналогового.

Крок квантування – це розрив між двома цифровими кодами і аналогове значення, що знаходиться між цими кодами буде округлятися.

$$h = \frac{Y_{max} - Y_{min}}{2^n}, \quad (1.2)$$

де  $h$  – крок квантування;

$Y_{max}$  – максимальне значення амплітуди;

$Y_{min}$  – мінімальне значення амплітуди;

$n$  – розрядність числа.

Звідси виникло поняття шуму квантування. Цей шум спеціально додають до сигналу, щоб якимось зменшити відмінності між оригіналом та отриманим цифровим значенням. На рисунку 1.2 наведено приклад.

					ІАЛЦ.466454.003 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

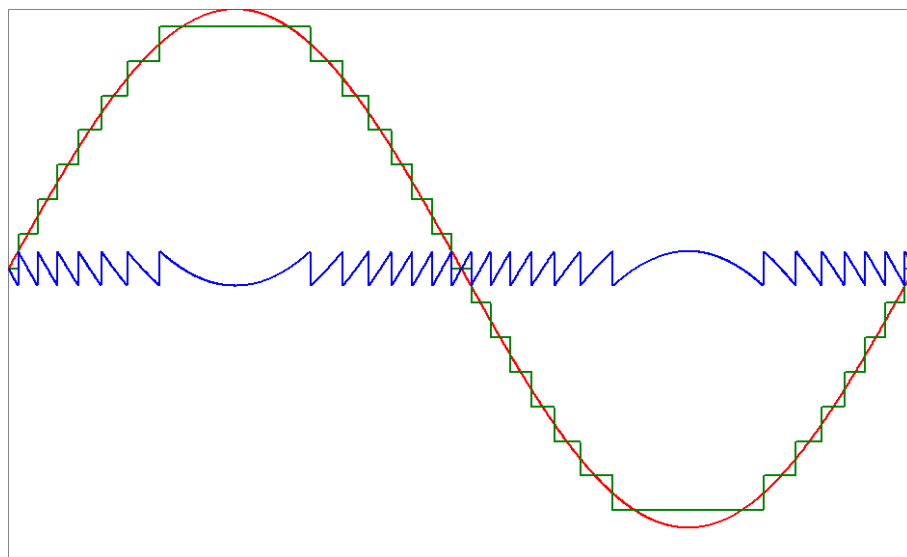


Рисунок 1.3 Різниця між оригінальним та оцифрованим сигналом

На цьому рисунку червоною лінією зображено оригінальний сигнал, зеленою цифрове представлення, а синьою різницю, що була втрачена під час перетворення. Звісно схема не може з повною ймовірністю знати значення цієї різниці, але існують спеціальні методи, наприклад алгоритм Флоїд-Стейнберга.

Інша, найбільш популярна операція з сигналами, це фільтрація. Стандартні для DSP цифрові фільтри це CIC та FIR, обидва характеризуються порядком.

CIC фільтр – інтегрально-гребінчастий фільтр використовується для децимації та інтерполяції (зменшення та збільшення частоти дискретизації). Головною відмінністю і перевагою цього фільтра є відсутність операцій множення, що займають багато ресурсів на ПЛІС та часу. Через це використання його у ПЛІС є дуже популярним. Цей фільтр складається з двох послідовно з'єднаних інтегратора та гребінчастого фільтрів. Інтегратор задається таким рівнянням(1.3).

$$y(n) = y(n - 1) + x(n) , \quad (1.3)$$

де  $x(n)$  – вхідний потік;

$y(n)$  – вихідний потік.

					ІАЛЦ.466454.003 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		



На рисунку 1.4 зображена структура. Функція  $z^{-i}$  представляє собою затримку на «i» відліків.

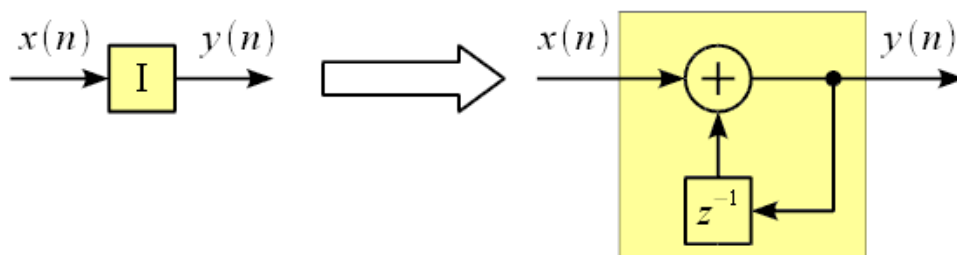


Рисунок 1.4 Структурна схема інтегратора

Тепер розглянемо гребінчастий КІХ фільтр:

$$y(n) = x(n) - x(n - D) \quad (1.4)$$

Блок  $Z^{-D}$  реалізує затримку на  $D$  відліків. На рисунку 1.5 зображена структура фільтра.

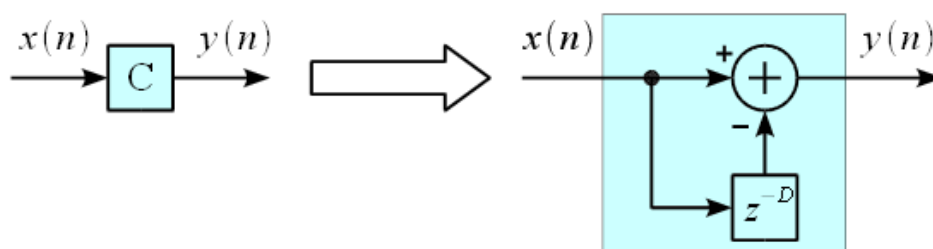


Рисунок 1.5 Структура гребінчастого фільтра

На рисунках 1.6(а, б, в) зображені АЧХ цього фільтра з різним значенням затримки. І як видно від цього залежить кількість повторів АЧХ (гребінок).

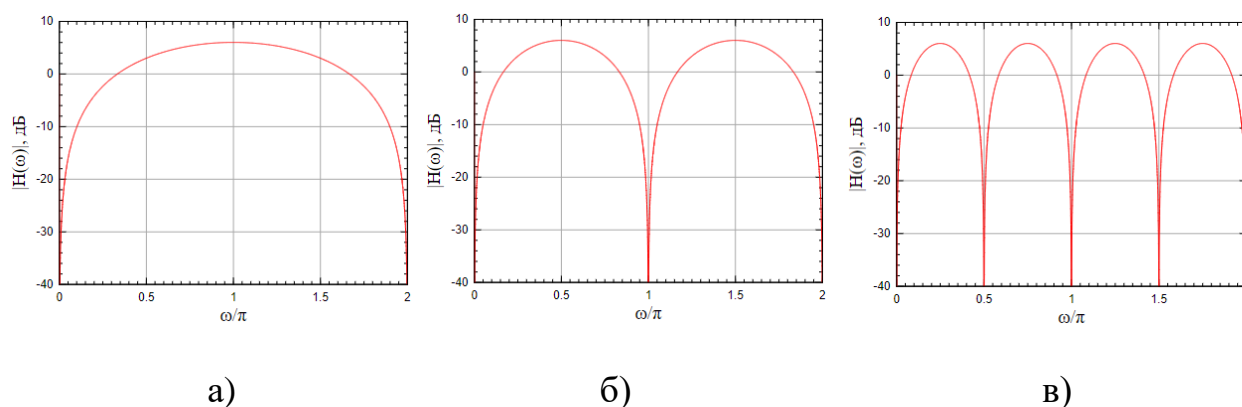


Рисунок 1.6 АЧХ гребінчастого фільтра (а –  $D=1$ , б –  $D=2$ , в –  $D=4$ )

Тепер з'єднаємо у каскад ці фільтри і отримаємо СІС фільтр 1-го порядку.

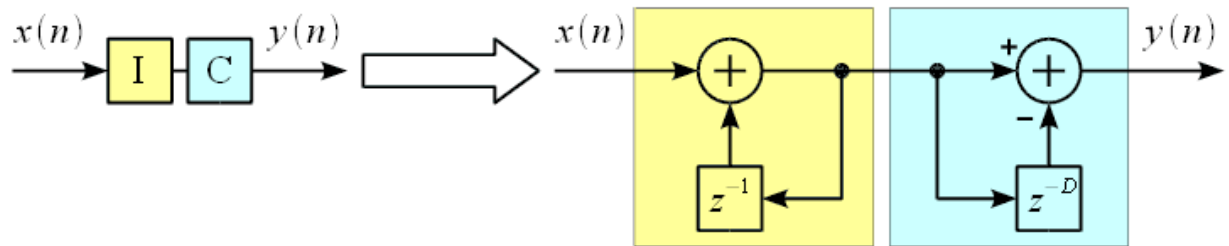


Рисунок 1.7 СІС-фільтр 1-го порядку

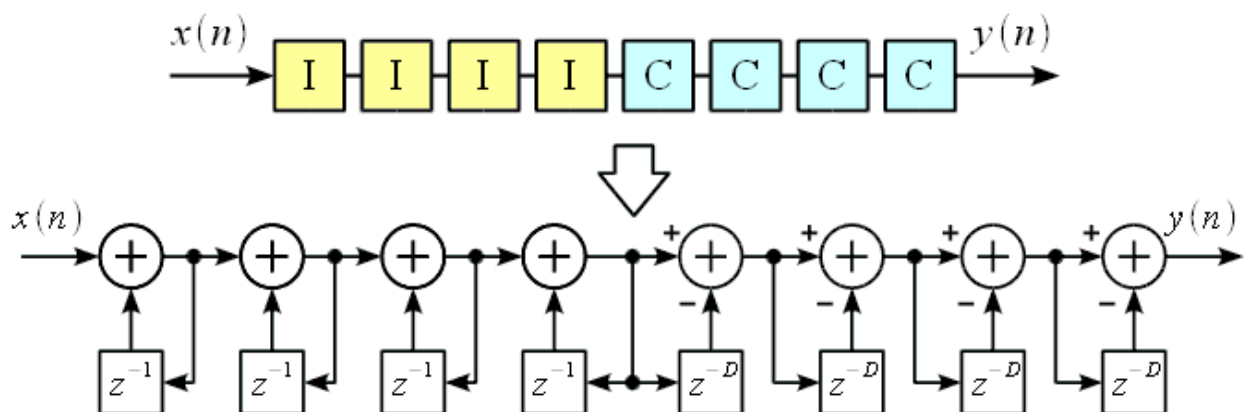


Рисунок 1.8 СІС-фільтр 4-го порядку

Передаточна характеристика такого фільтра:

$$H(z) = \frac{1-z^{-D}}{1-z^{-1}} \quad (1.5)$$

При частоті рівній нулю чисельник і знаменник будуть рівні нулю, тому значення АЧХ буде стримитися до D. Таким чином при D=1 цей фільтр буде пропускати все, три інших значення СІС можна використовувати як фільтр нижніх частот. На рисунку 1.9 (а, б, в) зображені АЧХ при різних затримках.



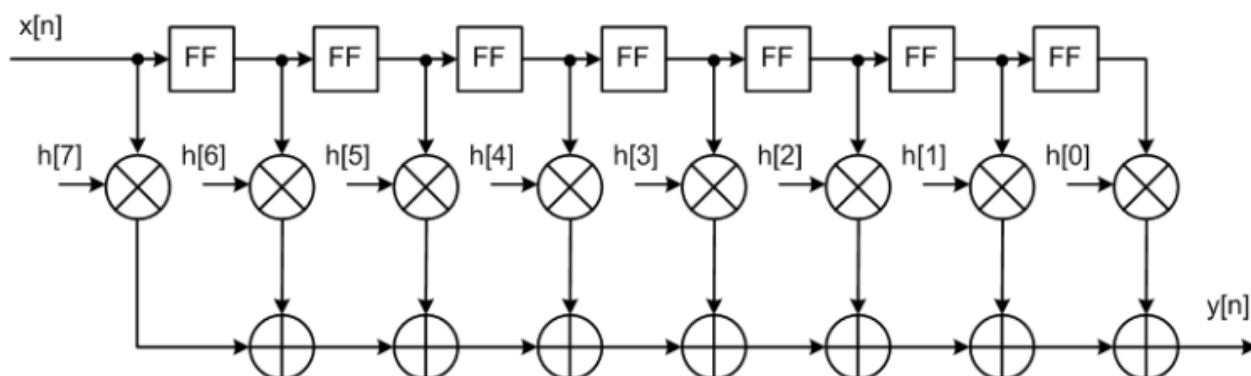


Рисунок 1.9 КІХ фільтр

Велике значення відіграє розрядність коефіцієнтів та проміжних даних і власне довжина фільтра. Мала розрядність чисел може призвести до спотворення АЧХ фільтра та до помилок квантування. Загалом КІХ фільтр є набагато більшим, адже потребує багато блоків множення та блоки пам'яті для збереження коефіцієнтів та пам'ять для затримок даних, але його перевагою є саме можливість реалізувати обвідну хвилі майже будь-якої форми, змінюючи коефіцієнти.

#### 1.4. Музичні терміни та правила

Нота – умовний символ, що характеризує висоту звуку.

Октава – відстань між двома тонами, частоти яких відрізняються у 2 рази, наприклад нота До 2-ї октави у 2 рази вище за ноту До 1-ї. Також октавою називають 7 основних тонів та 5 полу тонів. На рисунку 1.10 зображена одна октава, що містить 12 тонів. На рисунку 1.11 зображені всі існуючі октави [13].



Рисунок 1.10 Ноти



Рисунок 1.11 Перелік октав, що містяться у фортепіано

Частота ж між двома сусідніми нотами в одній октаві відрізняється у 1,059463 рази ( $2^{1/12}$ ). У таблиці 1.1 та 1.2 представлені частоти усіх нот.

Табл. 1.1 – Частоти у герцах усіх нот звукоряду

	ДО	РЕ	МИ	ФА	СОЛЬ	ЛЯ	СИ
<b>Субконтр</b>	16,352	18,354	20,602	21,827	24,5	27,5	30,868
<b>Контр</b>	32,704	36,708	41,204	43,654	49	55	61,736
<b>Велика</b>	65,408	73,416	82,408	87,308	98	110	123,472
<b>Мала</b>	130,816	146,832	164,816	174,616	196	220	246,944
<b>1</b>	261,632	293,664	329,632	349,232	392	440	493,888
<b>2</b>	523,264	587,328	659,264	698,464	784	880	987,776
<b>3</b>	1046,528	1174,656	1318,528	1396,928	1568	1760	1975,552
<b>4</b>	2093,056	2349,312	2637,056	2793,856	3136	3520	3951,104
<b>5</b>	4186,112	4698,624	5274,112	5587,712	6272	7040	7902,208
<b>6</b>	8372,224	9397,248	10548,22	11175,42	12544	14080	15804,42

Табл. 1.2 – Частоти у герцах усіх нот дієз звукоряду

	<b>ДО ДІЕЗ</b>	<b>РЕ ДІЕЗ</b>	<b>ФА ДІЕЗ</b>	<b>СОЛЬ ДІЕЗ</b>	<b>ЛЯ ДІЕЗ</b>
<b>Субконтр</b>	17,324	19,445	23,125	25,957	29,135
<b>Контр</b>	34,648	38,89	46,25	51,914	58,27
<b>Велика</b>	69,296	77,78	92,5	103,828	116,54
<b>Мала</b>	138,592	155,56	185	207,656	233,08
<b>1</b>	277,184	311,12	370	415,312	466,16
<b>2</b>	554,368	622,24	740	830,624	932,32
<b>3</b>	1108,736	1244,48	1480	1661,248	1864,64
<b>4</b>	2217,472	2488,96	2960	3322,496	3729,28
<b>5</b>	4434,944	4977,92	5920	6644,992	7458,56
<b>6</b>	8869,888	9955,84	11840	13289,98	14917,12

### 1.5. Методи та основи цифрового синтезу

Існує багато способів синтезувати звук [5] :

- Адитивний. Даний метод синтезу полягає у додаванні до основного тону ще декількох гармонік кратної частоти. Прикладом адитивного синтезу є орган.
- Субтрактивний. Цей метод є зворотнім до адитивного. Заснований на «відніманні» від сигналу його складових, тобто заснований на фільтрах.
- Семплювання. Цей метод заснований на отриманні звуку з записаного файлу та його подальшій обробці. Схожий на wavetable.
- FM/AM синтез. Вхідний тембр звуку спотворюється завдяки іншому(моделюючому) сигналу, що впливає на його форму.
- Фізичне моделювання. Використовує математичні моделі для отримання звуку.
- Wavetable синтез
- Інші

## 1.6. Форми хвиль, що використовуються у музиці

Раніше пристрої могли синтезувати тільки синусоїдальні сигнали (аналогові синтезатори). З приходом транзисторів з'явилася можливість синтезувати трикутні, прямокутні, пилу та імпульсні сигнали. На рисунку 1.12 наведені ці форми сигналів [12].

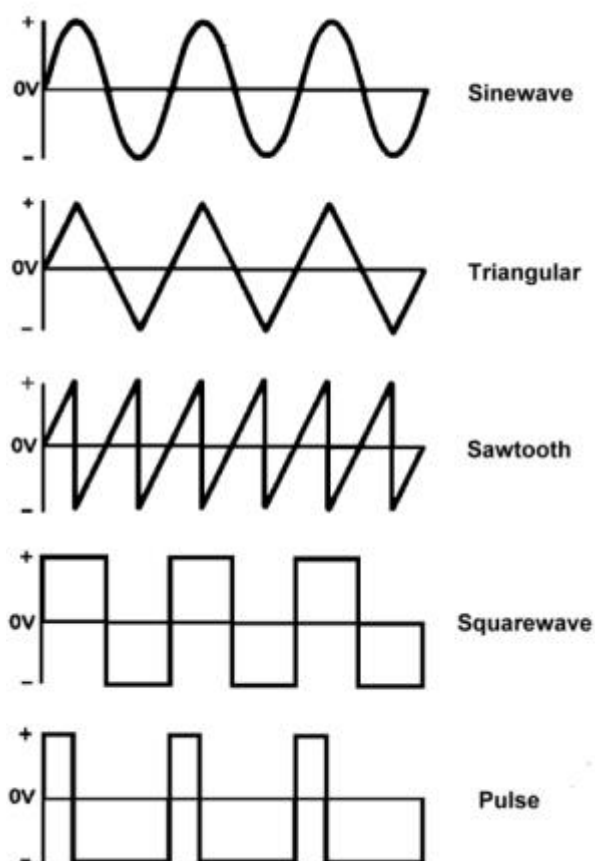


Рисунок 1.12 Форми сигналів

Синусоїдальні або як їх ще називають прості, адже з них отримується складний звук.

Прямокутні(меандр) – така форма сигналу використовується для імітації басових звуків і духовних інструментів.

Пиловидні – використовуються для смичкових інструментів.

Трикутні сигнали – їх звучання найбільш схоже на звук труби органу.

Імпульсні сигнали є прямокутними з певною скважністю. Скважність  $K$  означає, що сигнал включає всі гармоніки, крім гармонік кратних  $K$ .

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466454.003 ПЗ

Арк.

15

## 1.7. Архітектура синтезатора

На рисунку 1.13 зображена архітектура синтезатора. Далі описані функції усіх цих модулів.

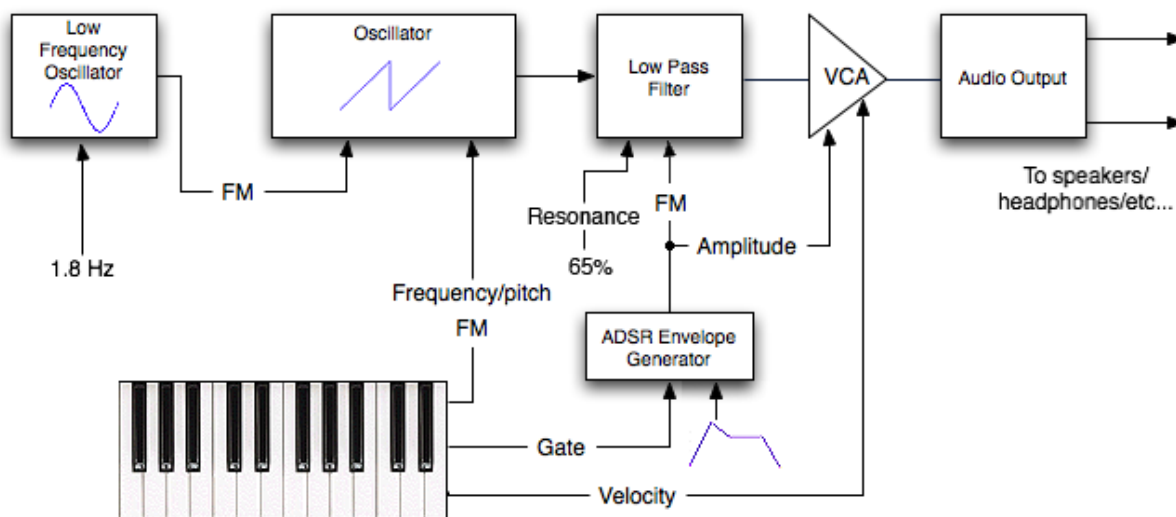


Рисунок 1.13 Архітектура модульного синтезатора

### 1.7.1. Осцилятор

В класичних синтезаторах усі форми звуку генеруються осцилятором (VCO/DCO) і далі вже йдуть на обробку. Генерація відбувається безперервно [7]. У VCO частота регулюється рівнем напруги, у цифровому ж подається значення що додається до акумулятора.

### 1.7.2. Зміна гучності

В аналогових синтезаторах обробка гучності відбувається у VCA – підсилювач контрольований напругою. В цифровому синтезі це можна реалізувати за допомогою помножувача на певний коефіцієнт.

### 1.7.3. Фільтр

Використовується для обмеження чи контрастності певного діапазону частот. Як правило це фільтр низьких частот, фільтр високих частот та смуговий фільтр.

Зм.	Арк.	№ докум.	Підпис	Дата



#### 1.7.4. Генератор обвідної хвилі

Генератором обвідної хвилі може бути низькочастотний фільтр з можливістю вибору форми або генератор ADSR, що є більш складним. За допомогою цього генератора вихідна гучність змінюється за кривою, що задана у ньому. Таким чином можна досягти звучання схожого на певний інструмент. На рисунку 1.14 можна побачити обвідні, що характерні деяким інструментам [8].

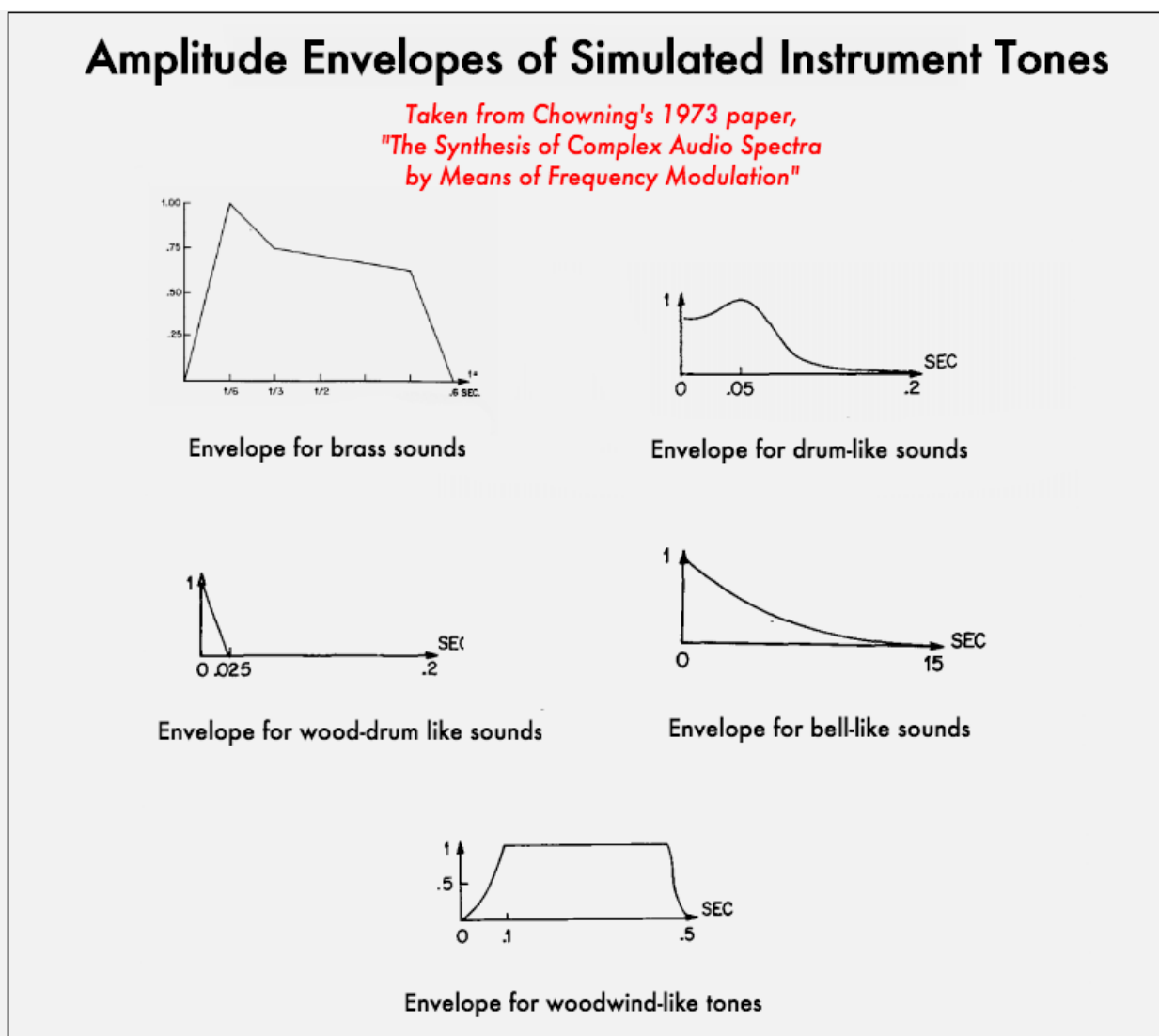


Рисунок 1.14 Графіки обвідних деяких інструментів

Як видно з рисунку ADSR складається з 4 частин:

- Attack(атака) – гучність з 0 збільшується до максимуму.
- Decay(спад) – гучність зменшується до рівня утримування

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466454.003 ПЗ

Арк.

17

- Sustain(утримування або підтримка) – тривалість цієї ділянки не залежить від параметрів, а триває доки натиснута клавіша. Рівень на якому утримується сигнал є 80-90 %.
- Release(відпускання) – ця ділянка починається після відпускання клавіші. Гучність падає з рівня підтримування до нуля.

#### **1.7.5. LFO. Низькочастотний осцилятор**

Блок генерує сигнал малої частоти від 1 до 2 Гц певної форми. Цей сигнал можна використовувати як обвідну амплітуди або для осцилятора для корегування частоти сигналу.

					ІАЛЦ.466454.003 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ ДО РОЗДІЛУ 1

В цьому розділі описана усі основні поняття музики та стадії синтезу. Отриманих знань достатньо для подальшого проектування обраного продукту. Виконання усіх вищенаведених модулів забезпечить повноцінну роботу продукту на рівні існуючих конкурентних рішень. У наступному розділі буде проведено аналіз можливих реалізацій та обґрунтування обраного мною способу.

					ІАЛЦ.466454.003 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 2

### АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ДЛЯ ІНТЕГРУВАННЯ ТА МОЖЛИВИХ РЕАЛІЗАЦІЙ ПРОЕКТУ

#### 2.1. Системи на мікроконтролері

Архітектура ARM представлена мікроконтролерами Cortex STM32. Дані мікроконтролери досить розвинені, найбільш продуктивна версія цього мікроконтролеру Cortex-M4. Наразі найпопулярніші рішення мають 32-бітну архітектуру, постійну пам'ять, оперативну пам'ять, таймери, інтерфейси обміну даними, швидку обробку переривань та працюють на частотах до 168 МГц.

На рисунку 2.1 зображена архітектура Cortex-M4.

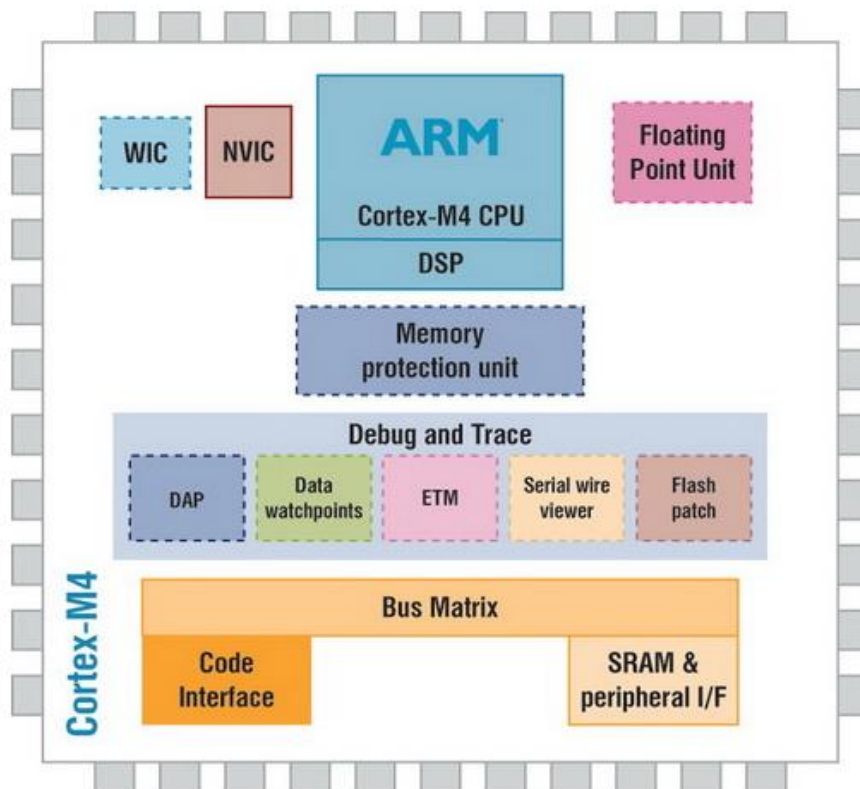


Рисунок 2.1 Ядро Cortex-M4

Компанія розміщує на кристалі мікроконтролер та DSP. Завдяки окремому блоку DSP, що виконує операції з плаваючою комою, цей мікроконтролер можна використовувати в нестандартних для нього проектах.

Ядро підтримує множення з накопленням (MAC) за 1 такт, обчислювальний блок FPU, що виконує операції за 1 такт та SIMD.

Структура також вражає, мікроконтролер має безліч інтерфейсів (I<sup>2</sup>C, SPI, USART, USB, camera interface, Ethernet) та аналогових засобів (АЦП, ЦАП).

На рисунку 2.2 зображена структурна схема ядра.

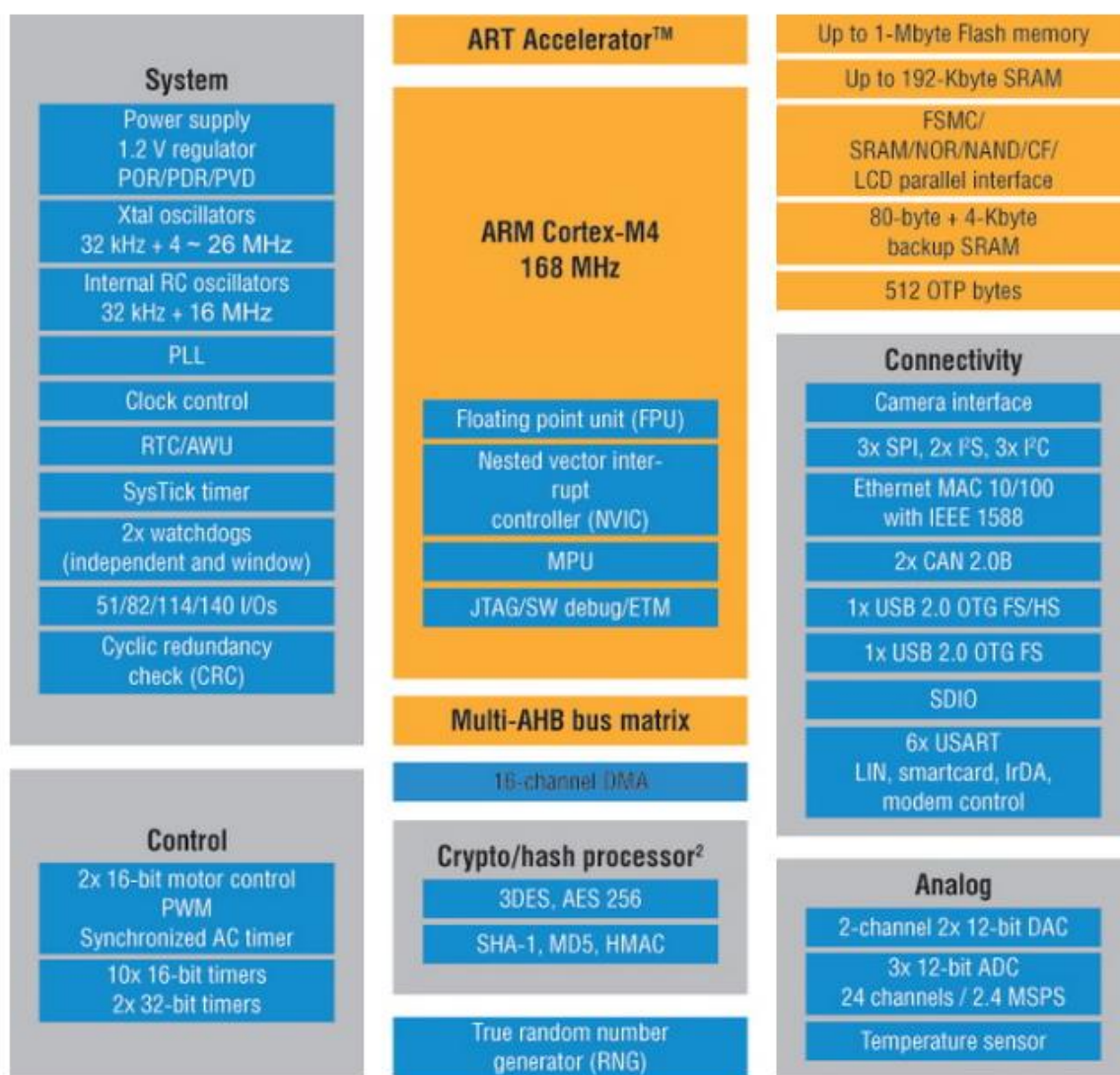


Рисунок 2.2 Структурна схема ядра Cortex-M4

Дійсно це сімейство мікроконтролерів, вражає та розширює ринок їх використання. Необхідність у використанні окремих чипів DSP зникла, а багата периферія дозволяє реалізувати проекти значної складності.

Але мікроконтролер виконує послідовність команд, тому його можливості не безграничні. У разі досить довгого ланцюга фільтрів та

обробок, час реакції на зовнішні події збільшиться, і звісно досягти паралельного генерування не вдасться, що призведе знову ж таки до неприпустимих затримок.

## **2.2. Програмний додаток на ПК(синтез на ЦП)**

Звичайно ЦП комп'ютера як і мікроконтролер виконує послідовність команд, але його архітектура набагато розвиненіша за мікроконтролер.

Перевагою процесора є наявність спеціалізованих блоків та інструкцій що виконують швидкі перетворення над числами великої розрядності. Також неможна забувати про наявність, як мінімум 2-х ядр, що працюють паралельно. Тому ця система підходить для реалізації даного проекту. Це підтверджено багатьма програмними доданками, що існують у даний час. Але існують і недоліки:

- Вартість такої системи перевищує вартість окремих пристроїв, адже необхідний досить потужний, а отже і кошковий процесор та кошкове програмне забезпечення
- Вимагає окремого комп'ютера для роботи, наприклад у студії або на концерті
- Вимагає окремої клавіатури, що приєднується до ПК
- Навіть на швидкісних процесорах виникають деякі затримки

## **2.3. Схема синтезатора складена з окремих мікро чипів, пам'яті, периферії**

На даний момент, це найпопулярніший варіант синтезатора. Його переваги полягають у спеціалізований чипах, що реалізують цифрову обробку сигналів (DSP), де виконується усе генерування та обробка ефектами. Наявний процесор виконує функції керування екраном та усіма клавішами налаштувань, що наявні на корпусі, також реалізує зв'язок з оперативною пам'яттю. Хоча і сам чип DSP має підключення до постійної пам'яті. На рисунку 2.3 наведена схема такого синтезатора.

					ІАЛЦ.466454.003 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

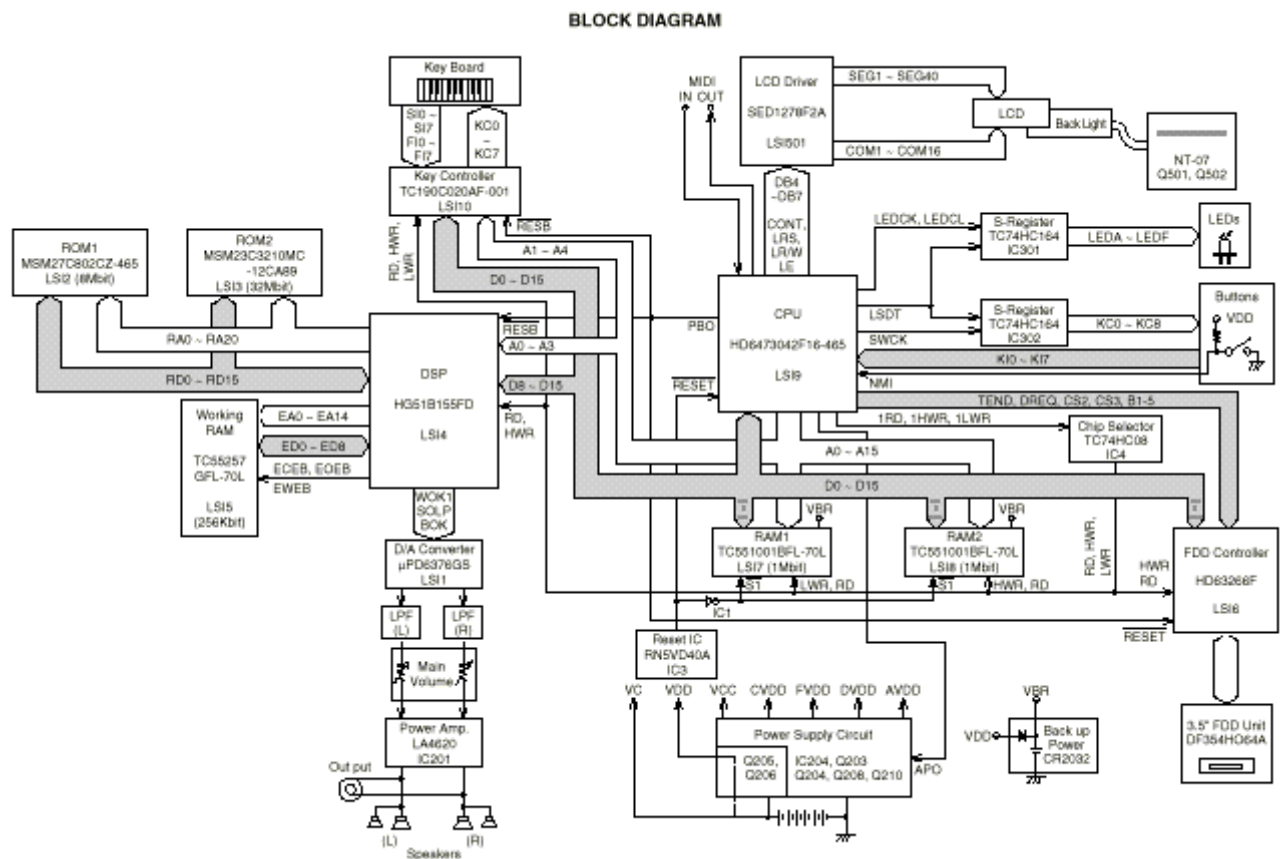


Рисунок 2.3 Схема цифрового синтезатора з DSP процесором

Найбільшу увагу треба звернути саме на мікросхему DSP. Ці процесори виконують операції над числами з фіксованою, або плаваючою комою, найчастіше мають архітектуру VLIW або SIMD і працюють на частотах до 1 ГГц. Нижче представлений опис найбільш популярних процесорів з фіксованою комою: сімейство Blackfin від компанії ADI, MSC81XX від Freescale, TMS320C64X від Texas Instruments. Усі ці процесори являються 16-бітними, використовують архітектуру VLIW та мають декілька блоків MAC(multiply-accumulate), що виконують функцію множення і додавання. Середня ціна таких процесорів 30 доларів. Але існують і багатоядерні рішення з масовим паралелізмом, що мають нижчу продуктивність і меншу ціну. Найкращим же рішенням є процесори з плаваючою комою від виробників Texas Instruments та Analog Devices. Побудовані на базі 8-тактної архітектури VLIW, мають набір інструкцій з фіксованою комою, але й спеціалізовані окремі блоки для роботи з плаваючою комою, можуть



виконувати дві 32-бітні MAC-операції на частоті 350МГц. Ціна такого процесора близька до 200 доларів [9].

## 2.4. ПЛІС(FPGA)

До цього усі можливі рішення були засновані на процесорах, спеціалізованих процесорах, швидкому виконанні послідовності команд. Але тепер розглянемо **ПЛІС** (Програмовану логічну інтегральну схему).

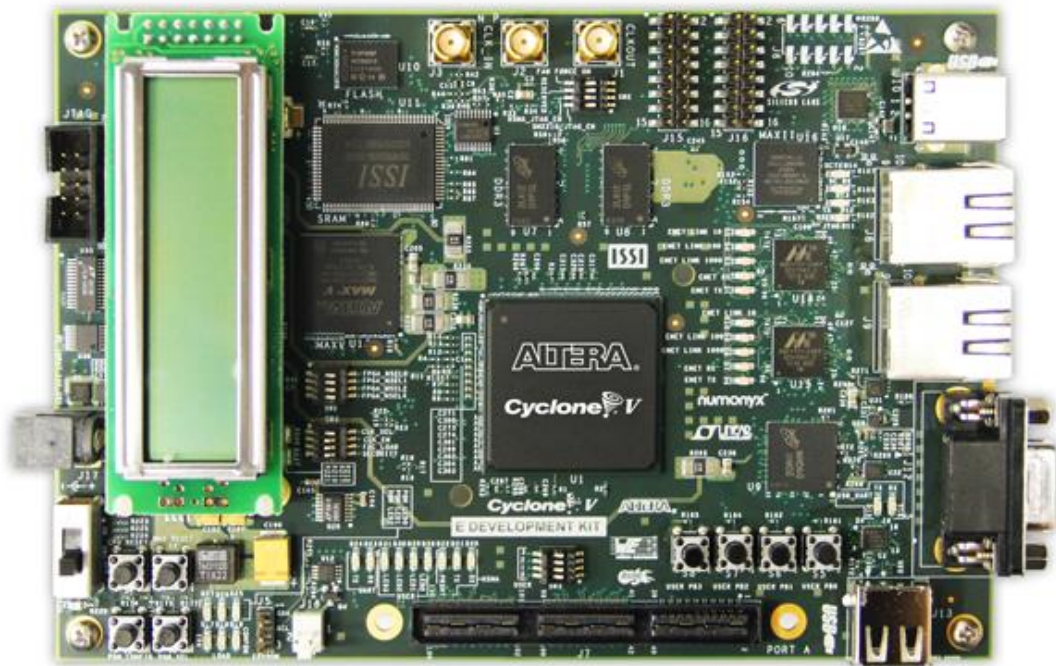


Рисунок 2.4 FPGA Altera Cyclone V

ПЛІС не виконує команди, а за допомогою мови опису апаратури, наприклад VHDL або Verilog, дозволяє написати такі блоки, що виконують певну дію безперервно в режимі конвеєру. Використовуючи ПЛІС ми не прив'язані до кількості ядер, що можуть одночасно виконати обмежену кількість операцій, ми можемо встановити певну кількість блоків, що будуть працювати одночасно і незалежно. Сучасні архітектури ПЛІС навіть не вимагають додаткової периферії такої як інтерфейс обміну даними з пам'яттю або комп'ютером, адже будь-яку схему можна реалізувати всередині кристалу. Сучасні ПЛІС вже мають вбудовану пам'ять, швидкісні

					ІАЛЦ.466454.003 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		



інтерфейси вводу\виводу(PCI Express), спеціалізовані блоки множення, блоки PLL, що дозволяють отримати надвисокі частоти роботи. І все це починаючи від 10 доларів за кристал з 6 тисячами логічних комірок. В цьому випадку найбільшою проблемою є саме проектування схеми. І так як уся схема знаходиться на одному кристалі і звільняє нас від з'єднань між різними мікросхемами, то надійність такою системи вища. Як результат, FPGA може бути імплемінтована в будь-який проект:

- пристрої комунікації(маршрутизатори)
- обробка картинок та відео
- прискорення математичних розрахунків
- інші швидкісні схеми

Інший варіант SoC, що поєднує мікропроцесор та ділянку FPGA, дозволяє поєднувати 2 способи роботи, мікрокоманди процесора та паралельні блоки на FPGA.

Існують багато технологій за якими виробляють програмовані логічні схеми. На рисунку 2.5 наведені основні типи ПЛІС [14,15].

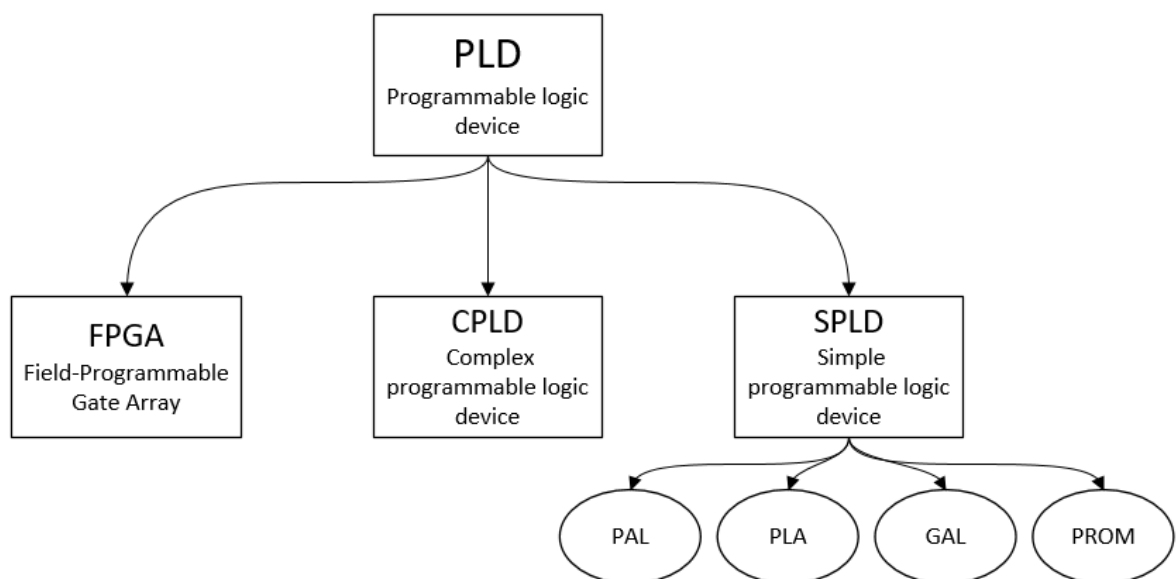


Рисунок 2.5 Види програмованих інтегральних схем

Кожне з цих сімейств відрізняється шляхом програмування, способом з'єднання ліній та багато чим іншим. Старші сімейства починалися з плавких контактів, тобто схему можна було прошити лише раз, інші мали енергонезалежну пам'ять, яку потрібно було прошивати. І лише з приходом CPLD та FPGA вдалося отримати енергонезалежну схему з безмежною кількістю перепрограмувань, що відбувалися дуже легко. На рисунку 2.6 показані види контактів, що «програмуються» [14,15].






Technology	Symbol	Predominantly associated with ...
Fusible-link		SPLDs
Antifuse		FPGAs
EPROM		SPLDs and CPLDs
E <sup>2</sup> PROM/ FLASH		SPLDs and CPLDs (some FPGAs)
SRAM		FPGAs (some CPLDs)

Рисунок 2.6 Види контактів у ПЛІС

Fusible-link – легкоплавке з'єднання, за допомогою спеціального програматора, непотрібні зв'язки руйнувалися, перепрограмування неможливе.

Antifuse – як і легкоплавке з'єднання antifuse є одноразовим, в початковому стані antifuse є розімкненим, і лише при протіканні току – з'єднується. Фізика процесу зворотна до запобіжника.

EPROM транзистор – до його виникнення у 1971 році усі зв'язки програмувалися лише раз. Тому створення транзистора було обов'язковим. Плаваючий затвор повністю ізолюваний від усіх інших частин і тому може зберігати свій стан та придатний до стирання за допомогою ультрафіолету. За допомогою більш високої напруги проходив запис у транзистор, що відповідав

одному біті. Так програмування інтегральної схеми звелось до очистки пам'яті ультрафіолетом та прошиванню чипа на програматорі.

EEPROM – ця пам'ять стала наступним кроком. Комірка пам'яті тепер могла бути очищена напругою.

SRAM – постійна оперативна пам'ять. Відрізнялась зворотнім зв'язком, що звільнило від періодичних перезаписів динамічної пам'яті. Енергозалежна пам'ять, але найпростіший спосіб читання\запису. Стирання пам'яті при вимкненні живлення призводить до встановлення Flash-пам'яті. При кожному вмиканні FPGA прошиває сама себе.

FPGA від різних виробників можуть відрізнятися архітектурою і структурними частинами, але все ж мають однакову базу. Концепція полягає у програмованих блоках, що складаються з LUT( look-up table)(таблиця істинності), тригера та мультиплексора. FPGA складається з багатьох блоків, починаючи від декількох тисяч, закінчуючи мільйонами.

Та логічні блоки також бувають різних типів. На рисунку 2.7 зображений MUX-блок, що реалізує необхідну функцію за допомогою багатьох мультиплексорів.

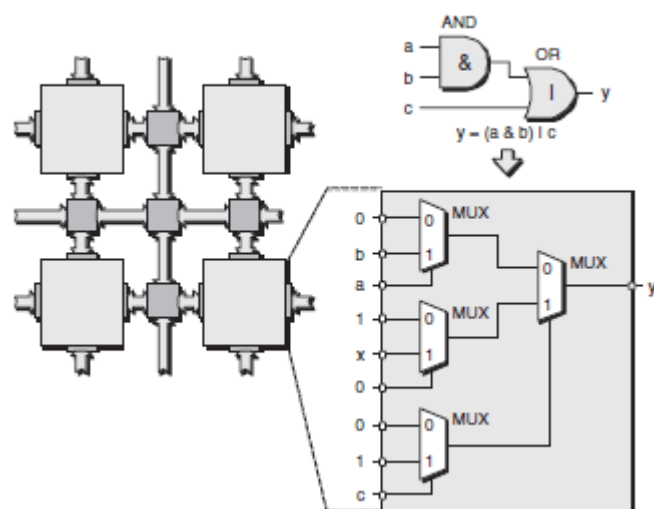


Рисунок 2.7 Приклад логічного елемента на MUX

Альтернативою і основним типом логічних блоків є LUT. На рисунку 2.8 зображена реалізація тої самої функції. Ця таблиця істинності програмується у елемент. Для забезпечення гнучкості FPGA має логічні елементи з різної кількістю входів, тобто LUT, що реалізує функцію від різної кількості змінних.

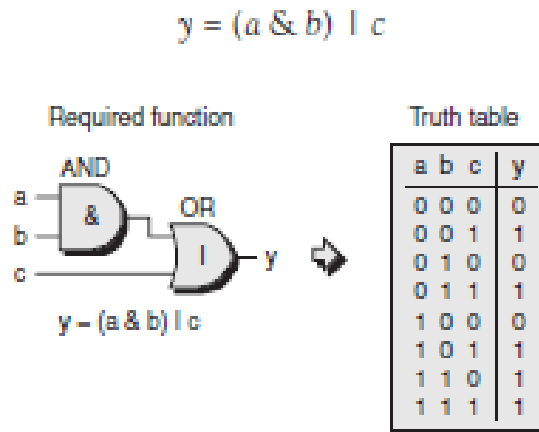


Рисунок 2.8 Приклад логічного елемента на LUT

Деякі функції, наприклад множення, ділення, підведення до степені, займають багато місця і працюють повільно якщо вони реалізовані на логічних елементах. Саме через це створили жорсткі блоки всередині ПЛІС – вбудовані помножувачі. Вони розподілені по всьому кристалу, розміщуються поряд с блоками RAM. Деякі ПЛІС мають виділені блоки суматорів, бо однією з популярних операцій DSP є MAC(multiply-and-accumulate). На рисунку 2.9 зображена ця операція.

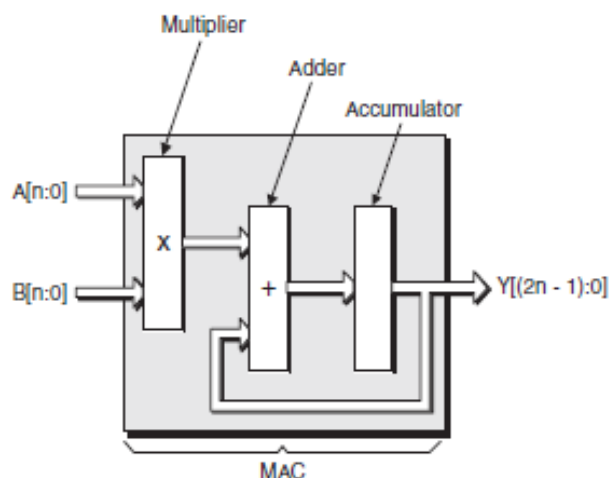


Рисунок 2.9 MAC

Коли написаний код перетворюється у схему, важливо розуміти як саме компілятор це зробить. Адже однакові на перший погляд ділянки коду можуть мати різну швидкість та займати більше ресурсів. Далі наведено приклади різниці між послідовним та паралельним способами опису схеми.

На рисунку 2.10 Зображена згенерована схема, як видно з коду схема реалізує мультиплексор, але конструкція if-then-else призвела до послідовного поєднання багатьох мультиплексорів, саме така поведінка призведе до затримки та займе більше логічних елементів.

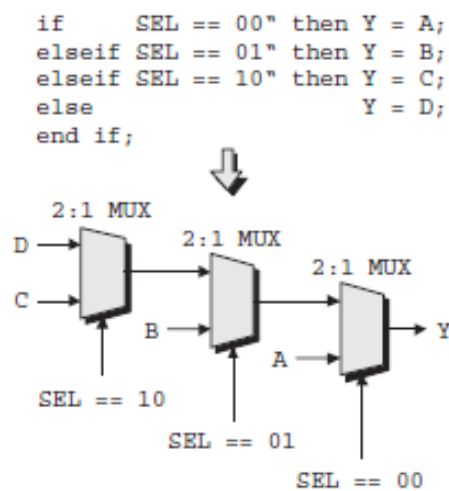


Рисунок 2.10 Приклад синтезу послідовного коду

Правильною поведінкою буде схема з одного мультиплексора з 4 входами. На рисунку 2.11 зображена ця схема, що реалізована за конструкцією case.

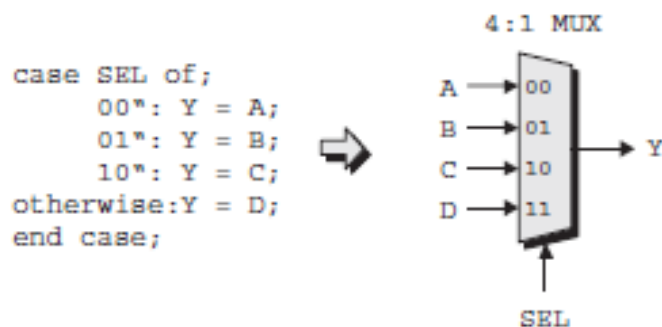


Рисунок 2.11 Приклад синтезу паралельного коду

Можна зробити висновок, що будь-який проект можна реалізувати сотнями способів, що будуть працювати абсолютно по-різному, мати різну конфігурацію, будуть використовувати різну кількість логічних блоків та спеціалізованих. Часто при використанні IP cores, користувачу дається вибір чи використовувати спеціалізовані блоки чи будувати схему на логічних елементах, чи акцентувати увагу на швидкість чи на економію місця. На рисунку 2.12 наведений приклад, що показує 2 різні реалізації однієї функції.

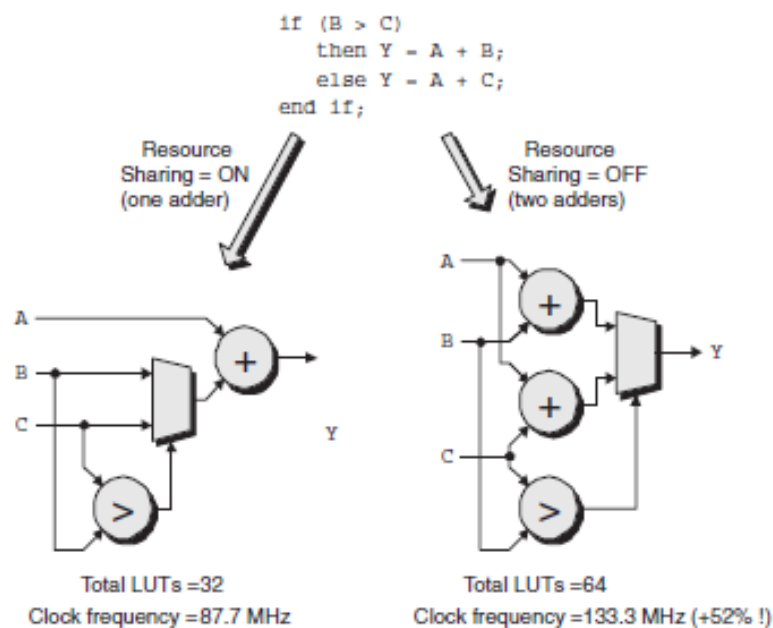


Рисунок 2.12 Приклад синтезу з економією ресурсів

Як видно з рисунку, перший варіант реалізовано послідовно. Спочатку відбудеться порівняння, результат його потрапить на мультиплексор і вже його результат потрапить на суматор. Цей варіант займає 32 логічні елементи, але через те що в ланцюгу стоїть три блоки, максимальна частота такої схеми 87 МГц. Інший варіант спрямований на швидкість. Додавання в цьому варіанті відбувається в обох випадках і паралельно з цим відбувається порівняння В і С. В цьому ланцюгу послідовність з 2 блоків, тому частота цієї схеми може бути 133 МГц, що в півтора рази більше.

## 2.5. Огляд архітектури FPGA на прикладі Intel Cyclone IV

Логічні елементи(LEs) є найменшими блоками в архітектурі ПЛІС. Кожен LE має [1,2] :

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

Регістр можна запрограмувати на D,RS,T,JK тригер. Кожен регістр має порти вхідних даних, синхронізації, дозволу синхронізації, очистки.

Кожен LE має три виходи, які керують локальними, рядковими та колонковими ресурсами маршрутизації. На рисунку 2.13 зображений один логічний елемент.

Кожна LAB(logic array block) складається з таких функцій [1,2]:

- 16 LE;
- Сигнали керування LAB;
- LE-сполучення;
- Регістрові сполучення;
- Місцеві з'єднання між LE.

Локальні сполучення передають сигнали між LE в одній LAB. Регістрові сполучення передають сигнали між регістрами LE в одній LAB. Cyclone IV включає в собі вбудовані модулі пам'яті М9К, що налаштовуються на різні функції, такі як: оперативна пам'ять, регістри зсуву, ROM, FIFO. На рисунку 2.14 зображений кристал Cyclone IV, розміщення основних частин [1,2].

					ІАЛЦ.466454.003 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

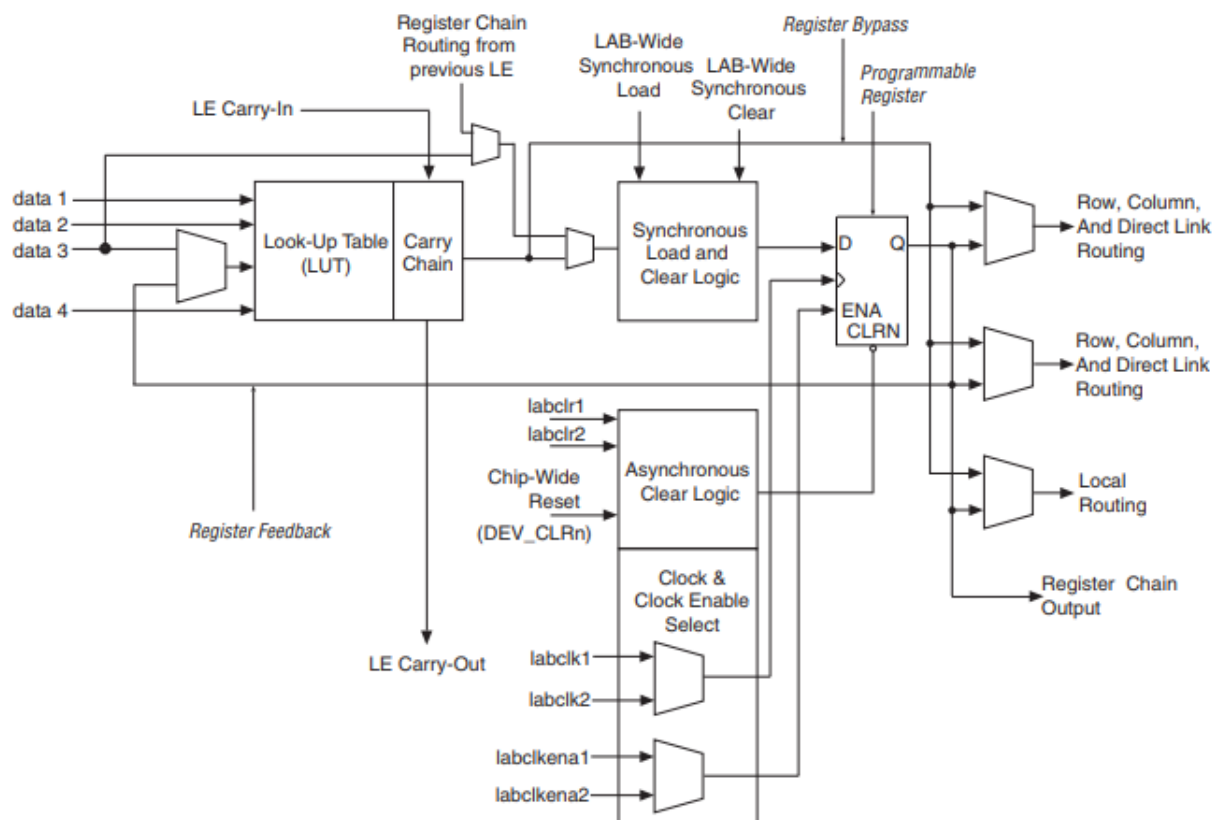


Рисунок 2.13 Логічний елемент

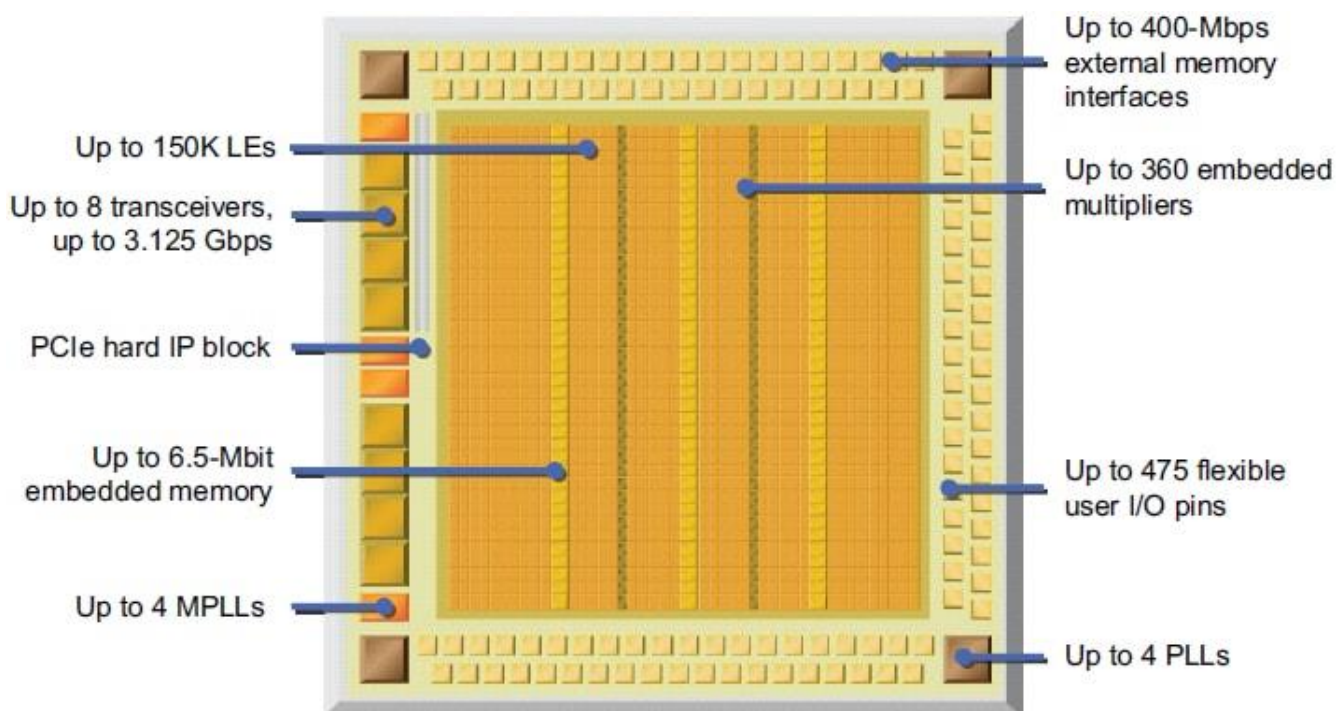


Рисунок 2.14 Топологія кристалу Cyclone IV

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466454.003 ПЗ

Арк.

32



## ВИСНОВКИ ДО РОЗДІЛУ 2

Переглянувши усі доступні системи реалізації цифрового синтезатора, оптимальний вибір падає на ПЛІС. Вартість такого рішення набагато менша за вартість спеціалізованих процесорів, а швидкодія і часові затримки набагато менші ніж у мікроконтролерах. Гнучкість ПЛІС дозволяє проектувати різні фільтри, обробки сигналу, інтерфейси зв'язку з ПК та багато іншого. При тому маючи продукт його покращення не потребує заміни деталей, корпусу та всього іншого, а полягає лише у зміні налаштувань ПЛІС. Це дозволяє робити унікальні продукти для кожного користувача в залежності від його потреб. В наступному розділі будуть описані модулі, що були розроблені для синтезатора на мові опису апаратури VHDL.

					ІАЛЦ.466454.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

## РОЗДІЛ 3

### СХЕМА СИНТЕЗУ ЗВУКУ. ОСНОВНІ ЧАСТИНИ

#### СПРОЕКТОВАНОГО СИНТЕЗАТОРА

##### 3.1 GEN

Модуль складається з накопичувального суматора. Доданком є число, що вираховується в залежності від тактової частоти схеми, необхідної частоти на виході та розміру акумулятору.

$$ACC = ACC + FREQ_{REG} \quad (3.1)$$

$$FREQ_{REG} = \left( \frac{FREQ_{OUT}}{FREQ_{CLOCK}} \right) * 2^{ACC_{SIZE}} \quad (3.2)$$

Старші біти акумулятора є адресом комірки пам'яті або джерелом сигналу типу «Пила», старший біт є джерелом сигналу «Меандр». На рисунку 3.2 зображена симуляція цього модуля. Для забезпечення якості сигналу було вибрано 32-бітний акумулятор, що дозволяє редагувати вихідну частоту з точністю  $FREQ_{CLOCK}/2^{32} = 0,0116$  Гц. Для забезпечення плавності вихідного сигналу було обрано довжину пам'яті 4096 комірок, що дорівнює 12 бітам адреси. Ширина пам'яті є другим параметром плавності сигналу, але використання ЦАП великої розрядності призведе до великих похибок. Для досягнення відмінного сигналу бажано використовувати спеціалізовані мікросхеми ЦАП. Обрана ширина пам'яті дорівнює 12 бітам, тому ширина шини як адреси, так і даних дорівнює 12 бітів. На рисунку 3.1 наведене графічне представлення даного модуля.

					ІАЛЦ.466454.003 ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

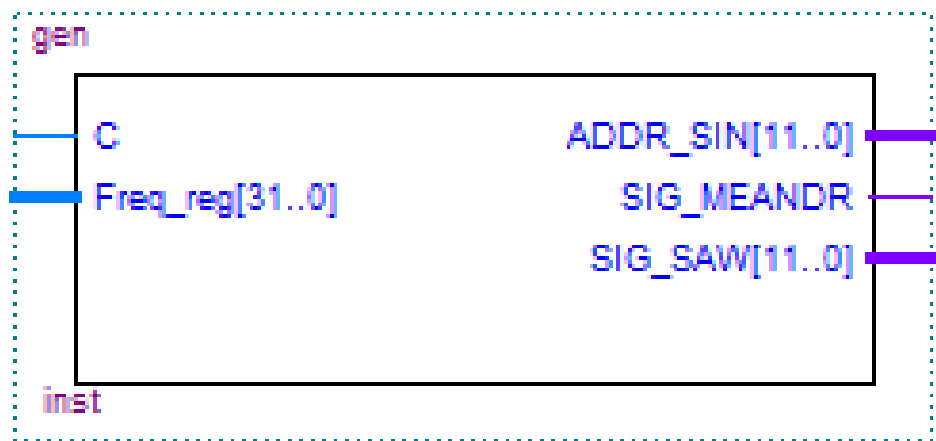


Рисунок 3.1 Модуль GEN

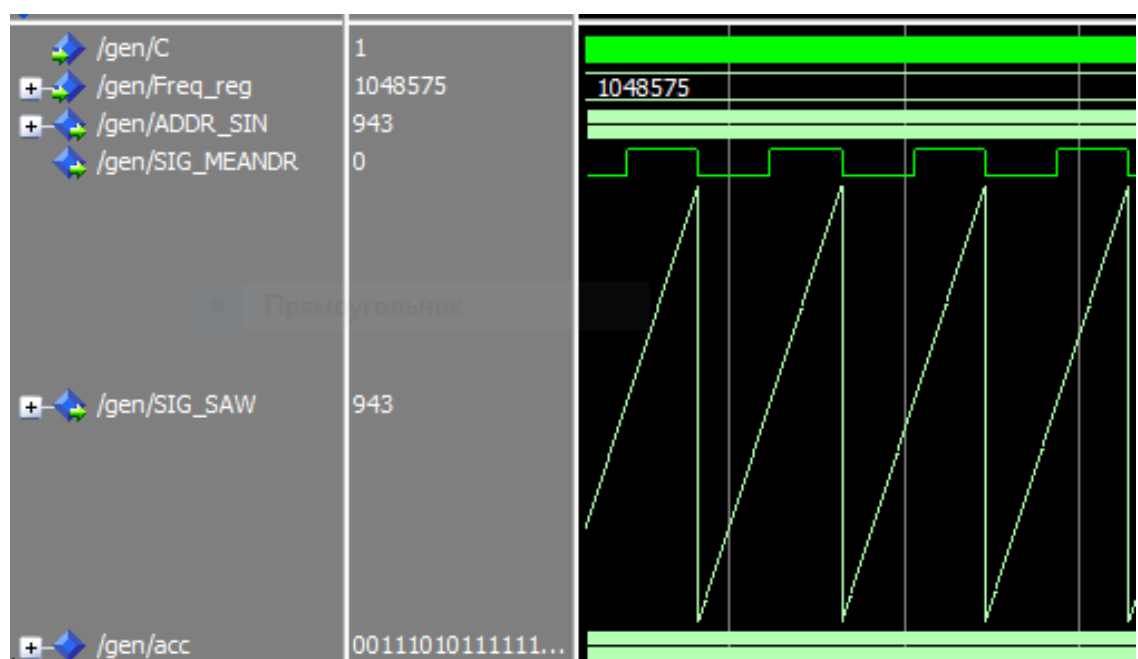


Рисунок 3.2 Симуляція GEN

У таблиці 3.1 наведено значення доданку для отримання певної частоти звуку. У пам'яті ПЛІС зберігаються лише значення частот для субконтрактави, що є найпершою октавою. Інші значення отримуються зсувом вліво.

Таблиця 3.1 – Значення доданків акумулятора для кожного значення частоти

Октава	Клавiші											
	1	2	3	4	5	6	7	8	9	10	11	12
1	1405	1488	1577	1670	1770	1875	1986	2105	2230	2362	2503	2652
2	2809	2976	3153	3341	3539	3750	3973	4209	4459	4724	5005	5303
3	5619	5952	6306	6681	7079	7500	7946	8418	8919	9449	10011	10606
4	11237	11905	12613	13363	14158	14999	15891	16836	17838	18898	20021	21212
5	22474	23810	25226	26725	28315	29999	31783	33673	35675	37796	40043	42425
6	44948	47620	50451	53450	56630	59998	63566	67345	71350	75591	80086	84849
7	89896	95240	100902	106900	113261	119995	127131	134690	142700	151183	160171	169699
8	179792	190479	201804	213800	226521	239990	254262	269380	285400	302366	320343	339397
9	359584	380959	403609	427600	453043	479981	508524	538761	570800	604731	640685	678794

### 3.2 OSCILL

Модуль є оболонкою для модуля GEN. Адреса отримана з GEN потрапляє до пам'яті, дані з пам'яті виводяться через порт. Наявність мультиплексора забезпечує вибір вихідного сигналу(синусоїда з пам'яті, меандр і пила з GEN). На рисунку 3.4 наведена симуляція з усіма вихідними сигналами. На рисунку 3.3 зображено графічне представлення модуля. Вхід MULT\_OUT служить для вибору вихідного сигналу :

- «00» - X «000»;
- «01» - меандр;
- «10» - пила;
- «11» - дані з пам'яті (синусоїда).

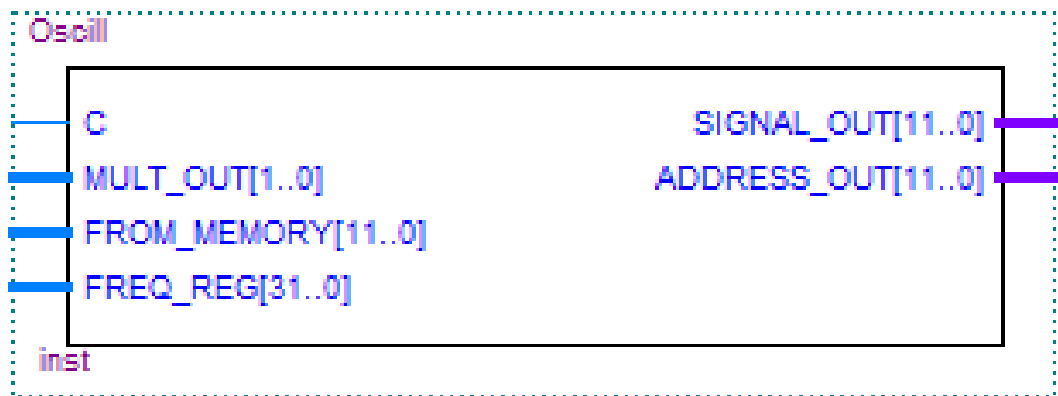


Рисунок 3.3 Модуль OSCILL

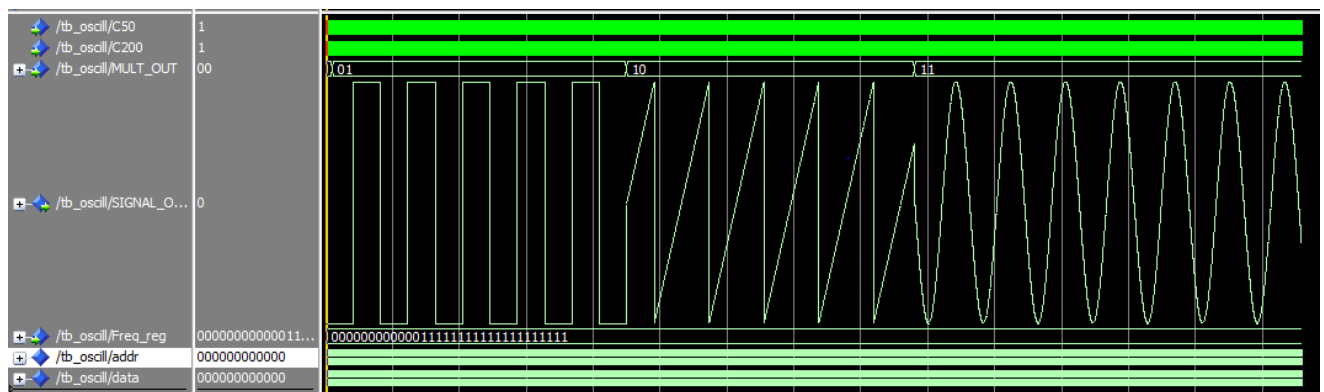


Рисунок 3.4 Симуляція OSCILL

### 3.3 CONV\_INT\_TO\_FLOAT

Модуль виконує перетворення цілого числа від 0 до 4095 у число з плаваючою комою для подальших перетворень амплітуди сигналу.

Перетворення виконується у форматі IEEE754. На рисунку 3. Зображене графічне представлення модуля. Даний модуль працює синхронно, про що свідчить вхід CLOCK, модуль має часову затримку між зміною даних на вході та видачою результату на виході. Ця затримка дорівнює 6 тактам, для зменшення часової затримки частоту роботи задано максимально можливою – 200 МГц. На рисунку 3.3 наведена симуляція цього блоку.

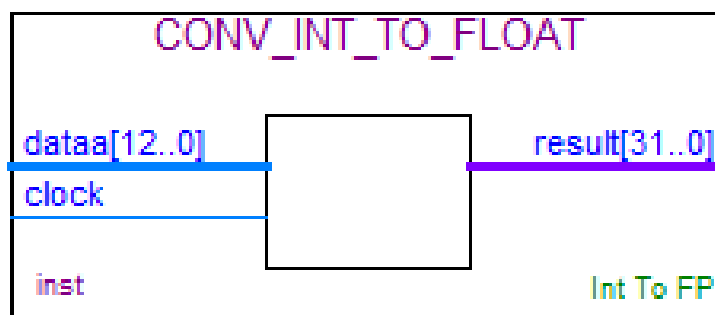


Рисунок 3.5 Модуль CONV\_INT\_TO\_FLOAT

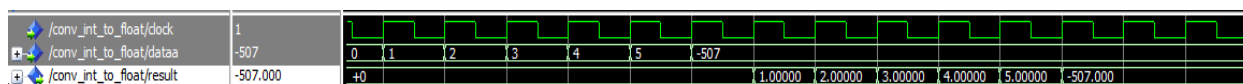


Рисунок 3.6 Симуляція CONV\_INT\_TO\_FLOAT

### 3.4 CONV\_FLOAT\_TO\_INT

Модуль виконує перетворення числа з плаваючою комою у ціле число для передачі числа на суматор і ЦАП.

На рисунку 3. Зображене графічне представлення модуля. Даний модуль працює синхронно, про що свідчить вхід CLOCK, модуль має часову затримку між зміною даних на вході та видачою результату на виході. Ця затримка дорівнює 6 тактам, для зменшення часової затримки частоту роботи задано максимально можливою – 200 МГц.

Перетворення виконується у форматі IEEE754. На рисунку 3.4 наведена симуляція цього блоку.

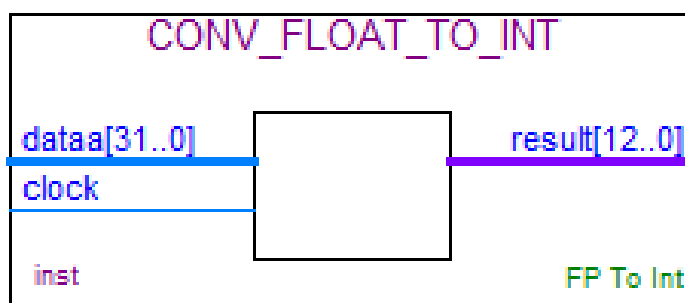


Рисунок 3.7 Модуль CONV\_FLOAT\_TO\_INT

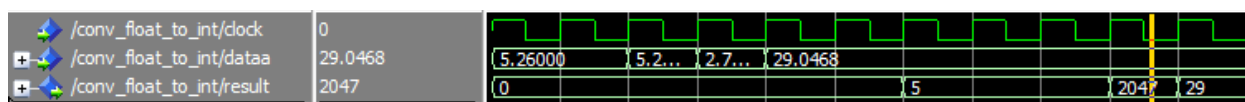


Рисунок 3.8 Симуляція CONV\_FLOAT\_TO\_INT

### 3.5 TIMER

Модуль чекає на сигнал GATE, що свідчить про натискання клавіші синтезатора. Лічильник рахує кількість відпрацьованих тактів та переводить автомат у певний стан. Стан автомату у реальному часі передається у модуль ADSR. Користувач задає довжину кожного стану, кількість тактів кожного режиму задаються на відповідних входах модуля. На рисунку 3.5 наведено модуляцію цього блоку. Звісно від заданої частоти цього модуля залежить плавність зміни рівня гучності. Чим більше частота модуля, тим більше треба робити значення ATTACK\_TIME, DECAY\_TIME, RELEASE\_TIME, та робити менше значення відповідної дельти. Максимальне значення часу кожного стану дорівнює  $2^{31}$ , що є границею типу natural. Тобто при частоті 200МГц довжина стану може бути 10 секунд, при частоті 10МГц – 214 секунд і так далі. Мінімальний числом дельти у форматі IEEE754 може бути  $1.4E-45(1.4 \cdot 10^{-45})$ .

Перед тим як змінити стан автомату модуль видає імпульс сигналу CLEARADD, що приходить на вхід суматора, що очищує усі дані всередині нього, так як суматор має затримку результату, і навіть після зміни стану продовжить видавати дані минулого стану.

Так як по сигналу GATE не можна перевірити зайнятий Block чи ні, було введено сигнал BUSY, що дорівнює 1 поки TIMER знаходиться у робочих станах.

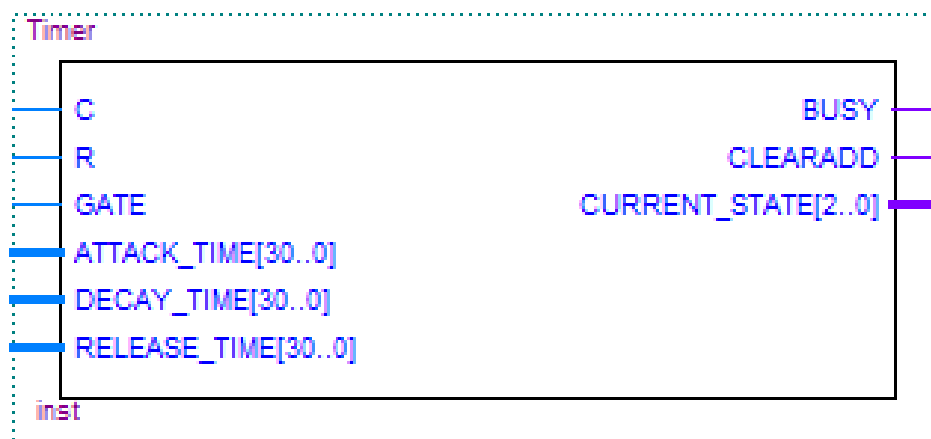


Рисунок 3.9 Модуль TIMER

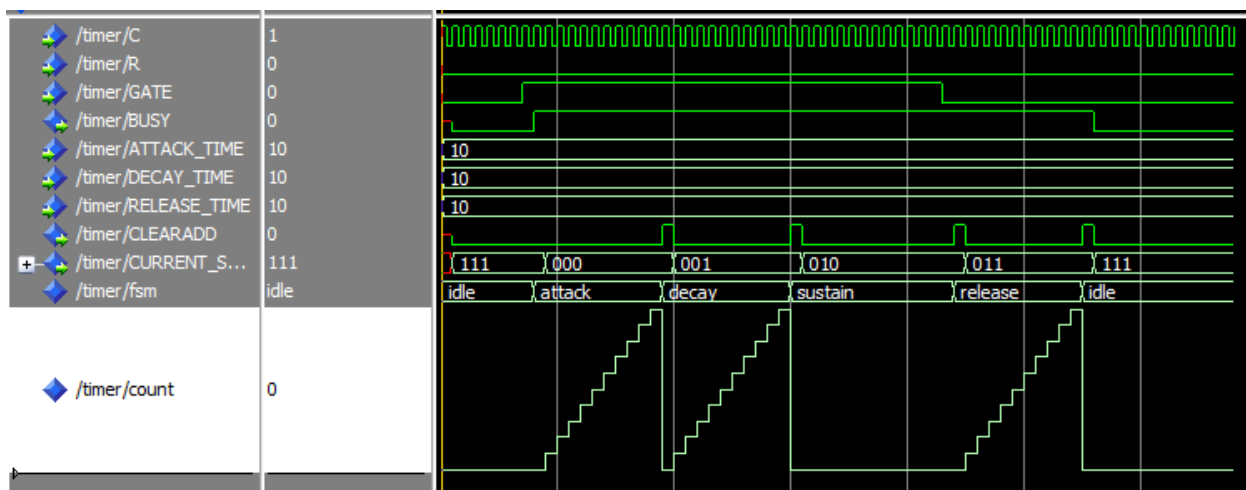


Рисунок 3.10 Симуляція TIMER

### 3.6 MUL

Модуль виконує множення чисел з плаваючою комою. Реалізує стадію обробки сигналу VCA, де вхідний сигнал з максимальною амплітудою обмежується коефіцієнтом, що отриманий від модуля ADSR. На рисунку 3.6 наведена симуляція даного блоку, де видно затримку у 6 тактів. Через це рекомендовано обирати максимально допустиму частоту роботи модуля – 200 МГц.

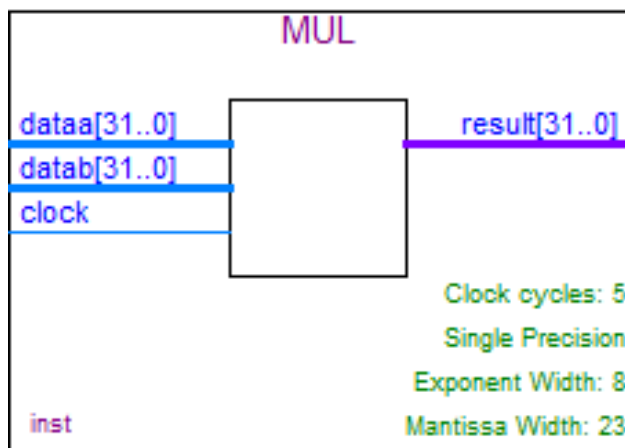


Рисунок 3.10 Модуль MUL

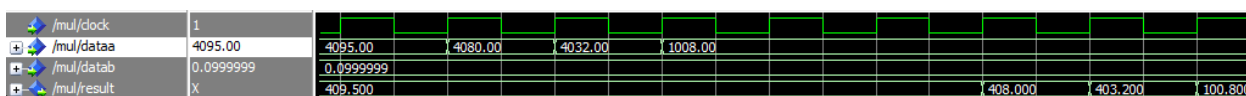


Рисунок 3.11 Симуляція MUL



### 3.7 ADSR

Модуль складається з TIMER та ADDSUB. Суматор працює в режимі накоплення, тобто одним з входів суматора є його вихід. Значення числа на іншому вході залежить від поточного стану, що приходить з TIMER. ADSR приймає на вхід три числа з плаваючою комою, ці числа є доданками, що подаються на вхід суматора. Так на виході ADSR виникає число з плаваючою комою, що росте з 0 до 1 в режимі ATTACK, спадає у режимі DECAY та RELEASE знову до 0. Це значення потрапляє на MUL, цим саме регулюється амплітуда вихідного сигналу. Нижче на рисунку 3.7 наведено симуляцію роботи. Завдяки зміні параметрів можна досягти різних звучань однієї і тої самої ноти.

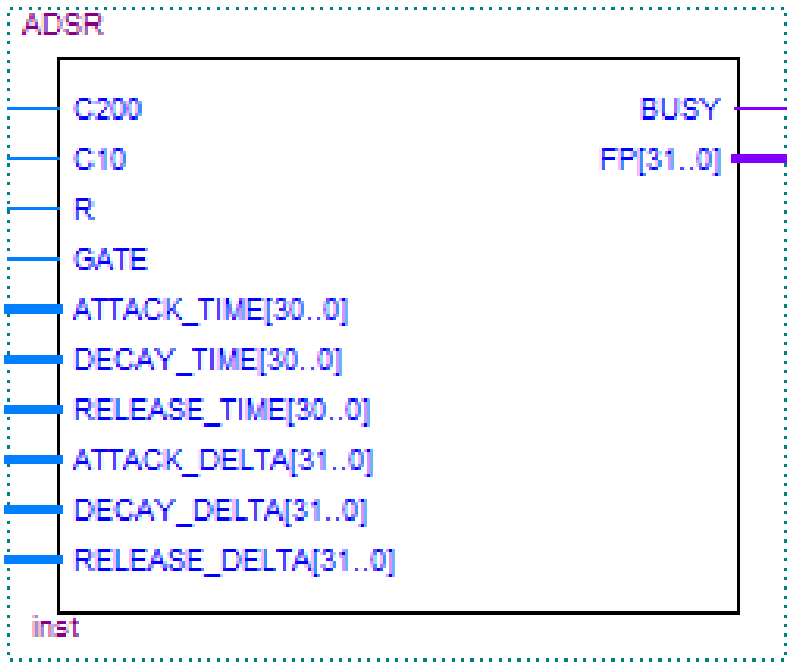


Рисунок 3.12 Модуль ADSR

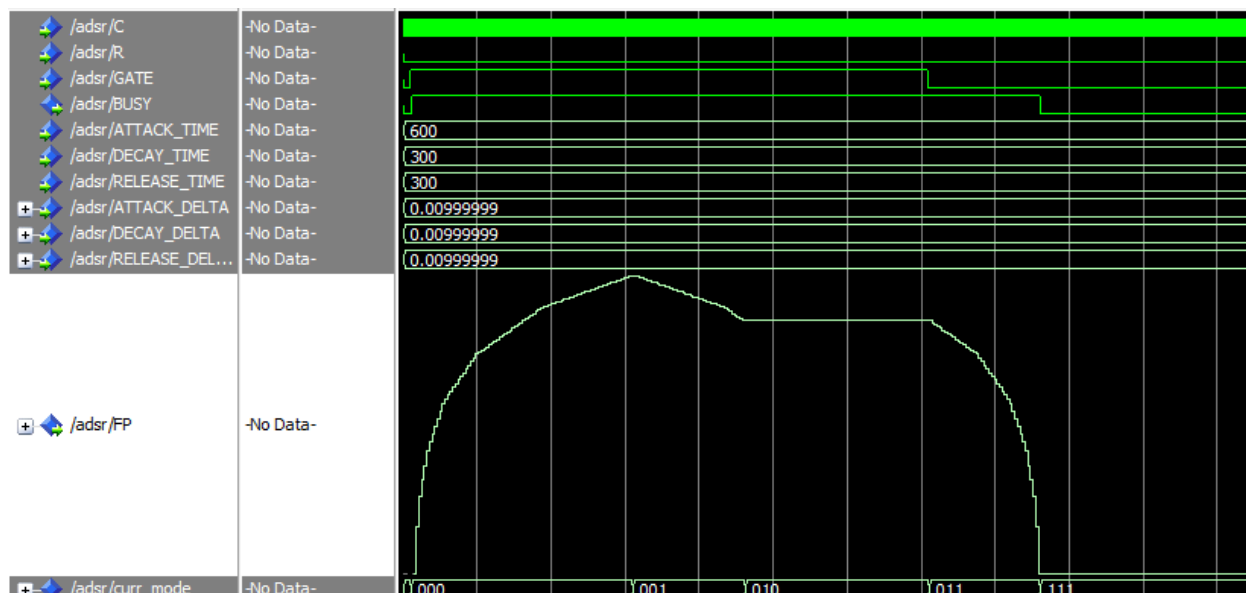


Рисунок 3.13 Симуляція ADSR

### 3.8 BLOCK

Цей модуль поєднує усі блоки, що були описані вище. Один блок може оброблювати одну натиснуту клавішу. Для повноцінної роботи синтезатор має мати як мінімум 5 таких блоків. Модуль приймає на свої входи такі параметри: натиск клавіші, що запускає таймер, число, необхідне GEN для генерації сигналу та параметри для блоку ADSR.

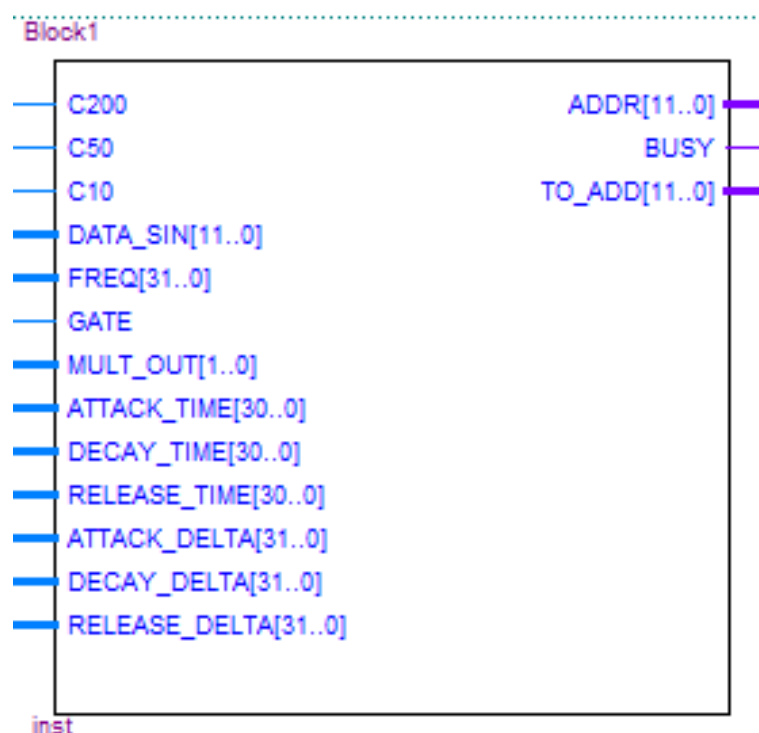


Рисунок 3.14 Модуль Block

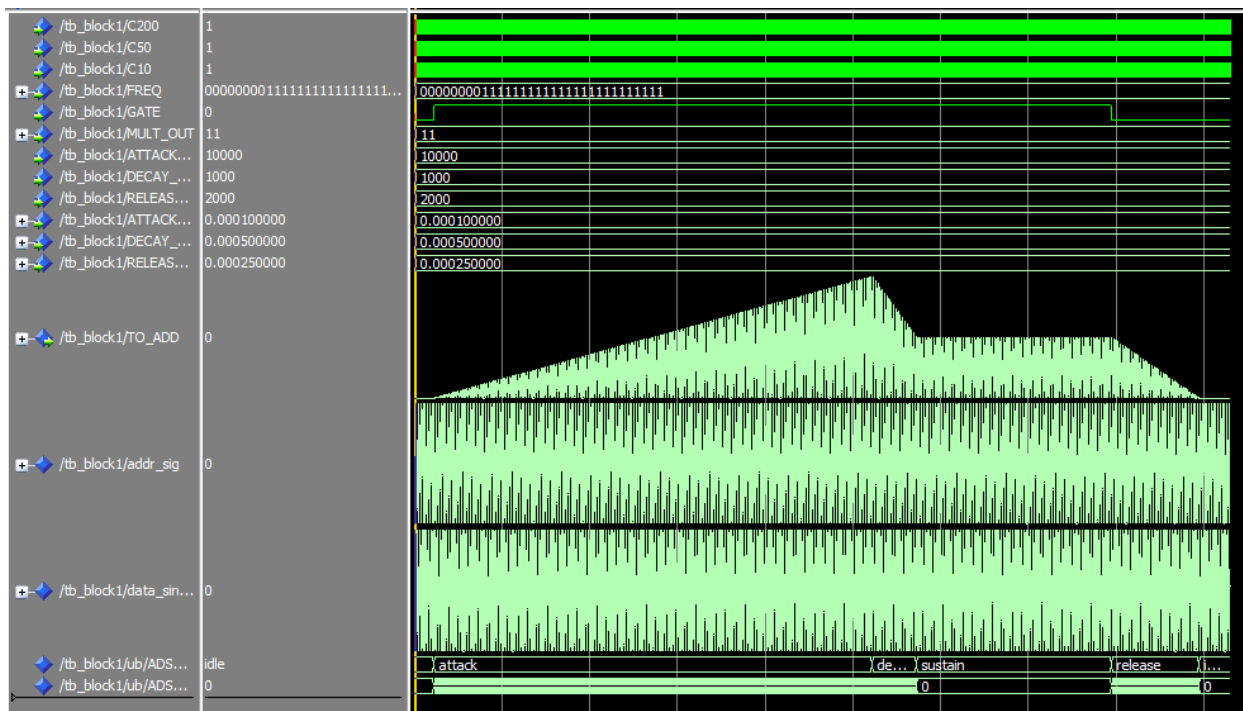


Рисунок 3.15 Симуляція BLOCK

### 3.9 ADD

Виконує поєднання сигналів, що виходять з модулів Block. Реалізований на паралельному суматорі з IP Core від компанії Altera. На рисунку 3.16 зображене графічне позначення модуля. На рисунку 3.17 зображена симуляція цього модуля.

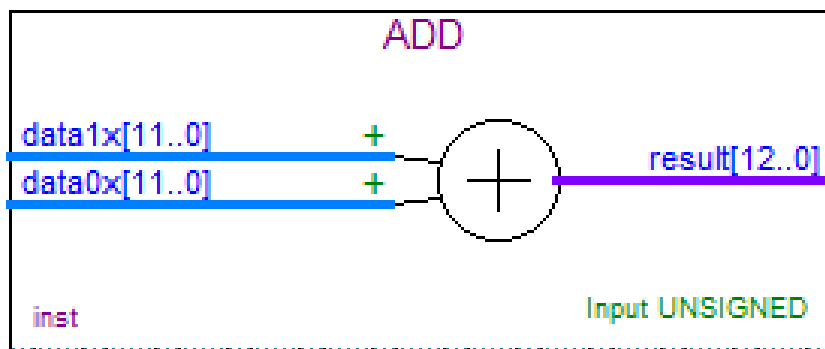


Рисунок 3.16 Модуль ADD

/add/data0x	13	4	12	13
/add/data1x	783	8		15
/add/result	796	12	20	27
				796

Рисунок 3.17 Модуль ADD

### 3.10 UART\_RX

UART – універсальний асинхронний приймач\передавач, виконує побітову передачу. Часто використовується для з'єднання комп'ютера з периферією. Усі сучасні мікроконтролери мають вбудований UART.

Модуль UARTRX реалізує лиш приймач від комп'ютера, для отримання параметрів синтезатора. UART використовує зсувний регістр як для отримання, так і для передачі.

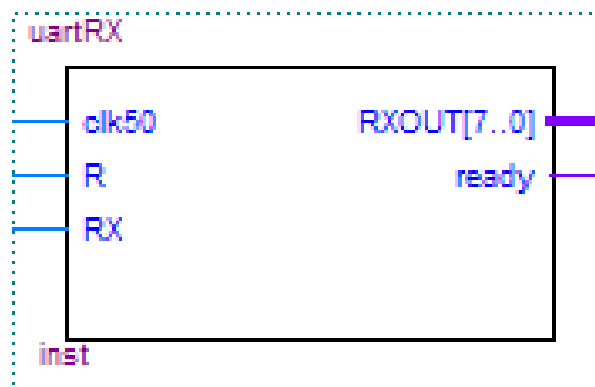


Рисунок 3.18 Модуль UART\_RX

### 3.11 BUTTON\_PROC

Так як ресурси ПЛІС обмежені, то встановлення 12 модулів Block для паралельної роботи усіх 12 клавiш неможливе. Це змушує робити модуль арбітра, що реалізований у BUTTON\_PROC. Цей модуль отримує сигнал BUSY від кожного Block'у. Якщо є вільний Block та була натиснута клавiша, то модуль починає її відтворювати, якщо ж усі Block'и зайняті, то реакції на натиск клавiші не буде. Використана мною ПЛІС має лише 6 тисяч логічних елементів, тому може вмістити 2 або 3 модуля Block. Також в залежності від натиснутої клавiші та поточного значення октави, Button\_Proc дасть потрібне значення частоти для генерування у модулі Oscill.

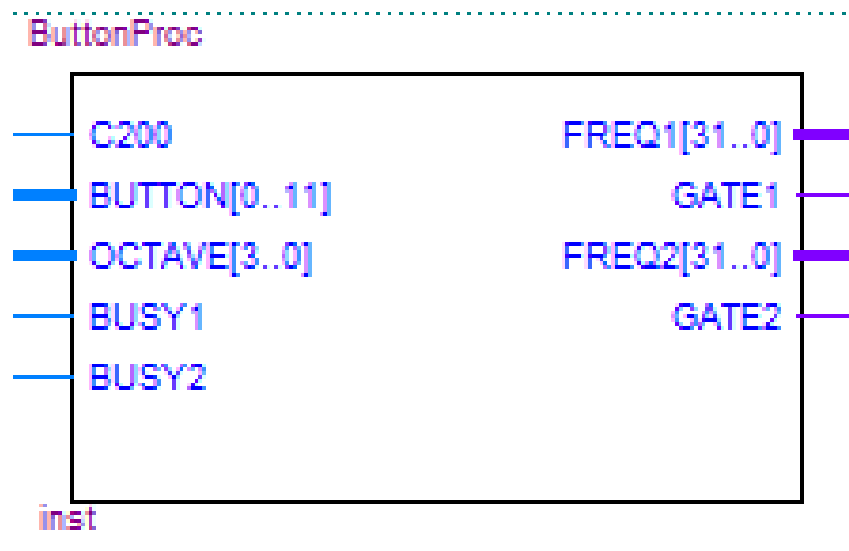


Рисунок 3.19 Модуль BUTTON\_PROC

### 3.12 UART\_PROC

Дані що приходять від комп'ютера на плату, складаються в байти у модулі UART\_RX. Ці байти потрапляють до даного модуля. Так як параметрів, що приходять багато, то треба реалізувати процедуру отримання адреси і далі вже самих даних, що складаються з 4 байтів. UART\_PROC працює за принципом автомату і формує адресу і дані, що далі потрапляють на потрібний вихід модуля. Синтезатор не обов'язково мусить бути підключений до ПК, так як цей модуль має стандартні налаштування, та 2 клавіші для ще 2 режимів налаштувань.

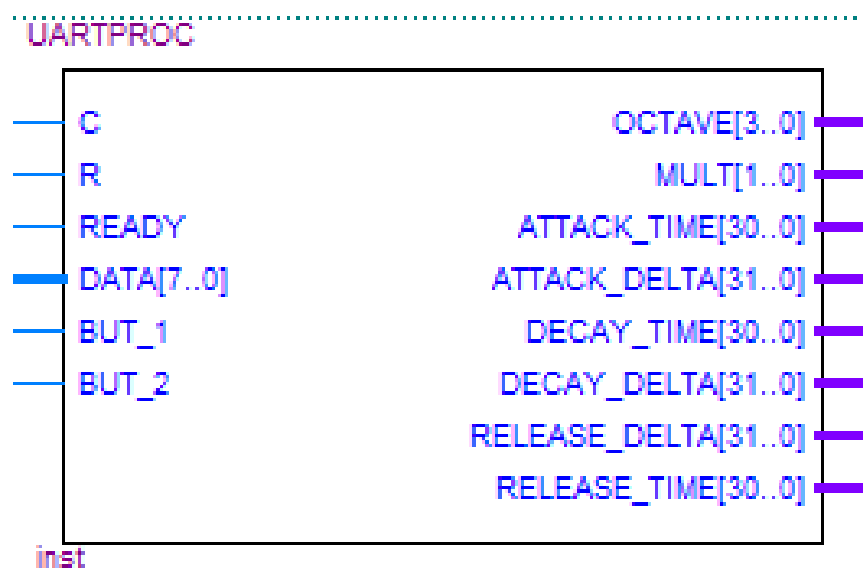


Рисунок 3.20 Модуль UART\_PROC

### 3.13 ЦАП ЗА СХЕМОЮ R2R

Так як ПЛІС працює з цифровими сигналами, то отримання аналогового сигналу на ній неможливе. Для цього використовується схема ЦАП (цифро-аналоговий перетворювач). ЦАП має декілька варіантів роботи: з паралельною входом і з послідовним входом. Як правило, послідовні ЦАП працюють за схемою дельта-сігма модуляції, де шириною імпульсу, регулюється вихідна напруга. Паралельний ЦАП можна реалізувати за допомогою матриці резисторів, де вага розряду отримується на ділнику напруги. На рисунку 3.9 зображено приклад схема ЦАП на матриці R2R.

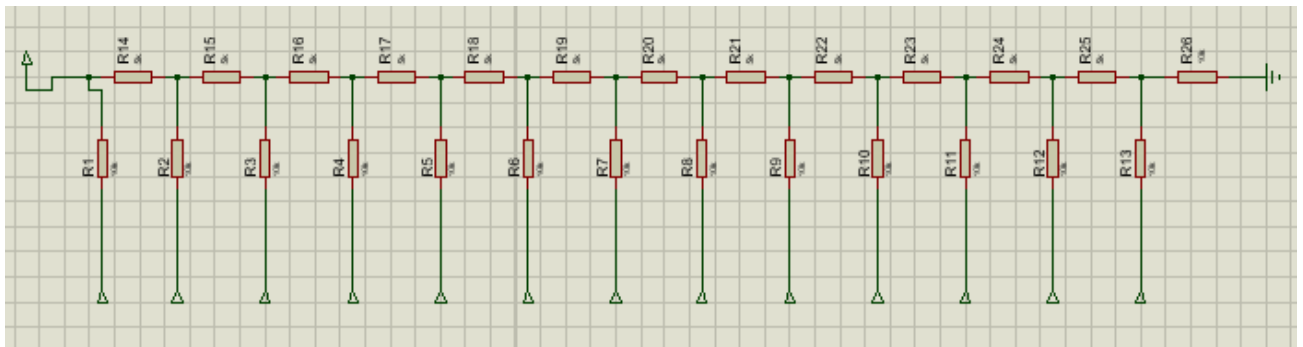


Рисунок 3.21 ЦАП

### 3.14 РЕЗУЛЬТАТИ СИМУЛЯЦІЙ

На рисунку 3.22 та 3.23 наведена діаграми симуляції усього проекту. Отриманні дані співпадають з очікуваними.

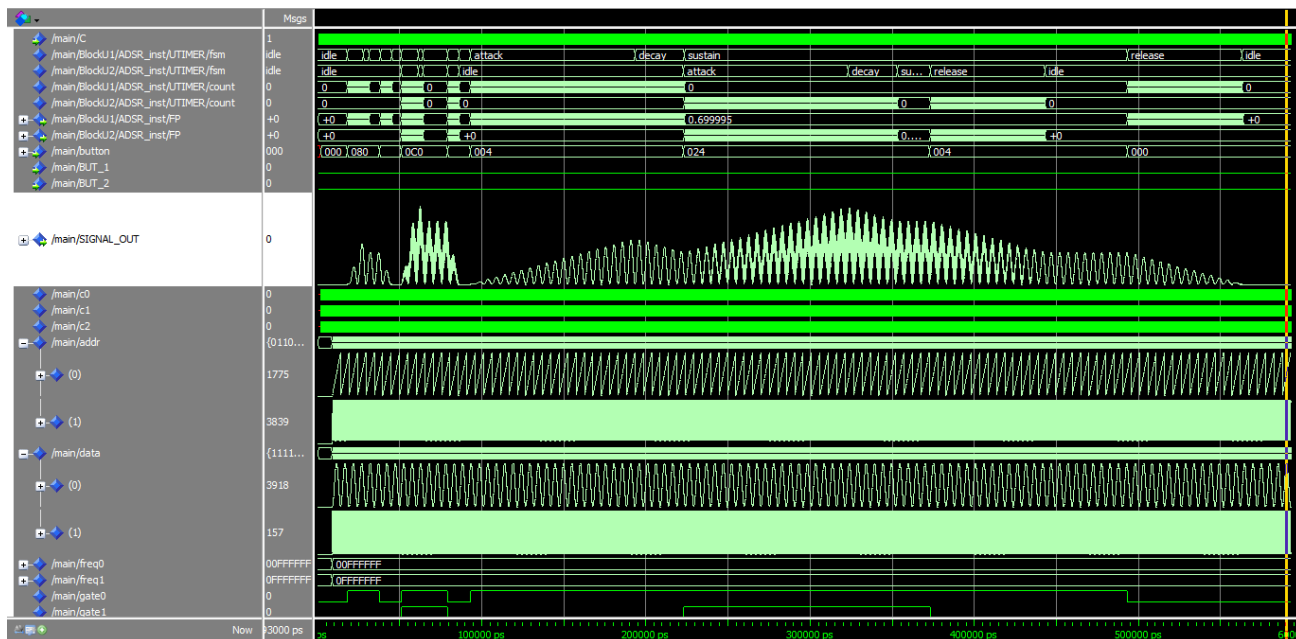


Рисунок 3.22 Симуляція проекту

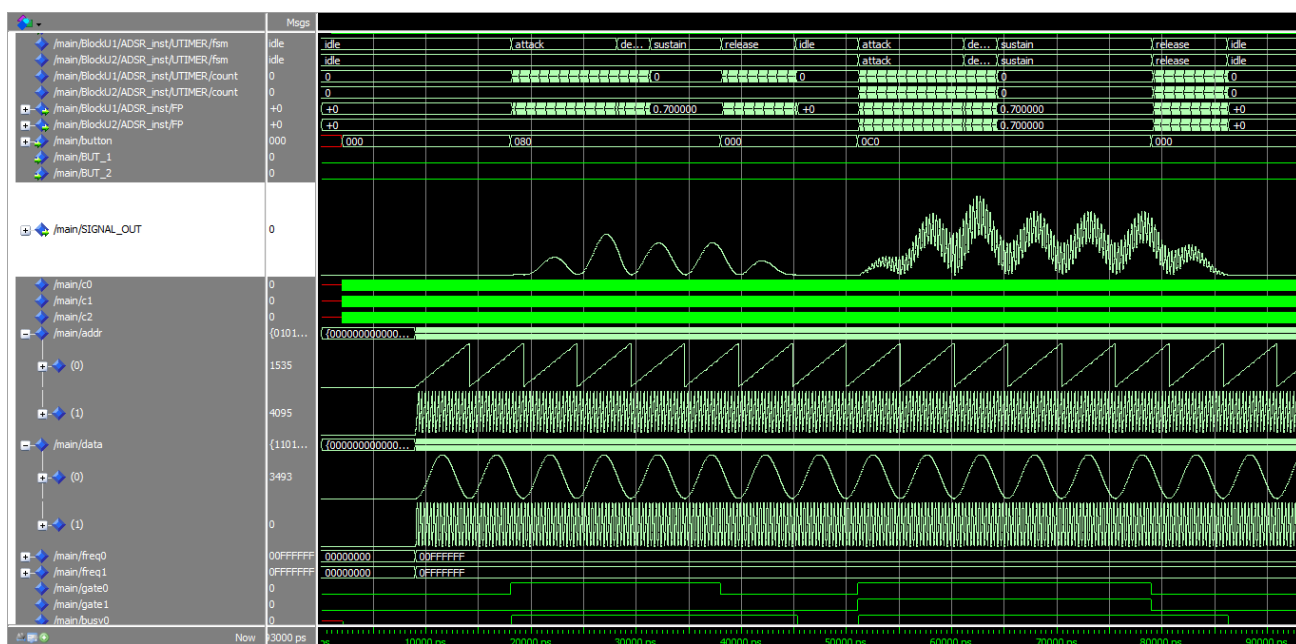


Рисунок 3.23 Симуляція проекту

На рисунку 3.23 зображений збільшену ділянку діаграми. На рисунку зображені дві діаграми адрес та дві діаграми сигналу синусоїди. Спочатку була натиснута одна клавіша, про це свідчить рисунок 3.24.

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466454.003 ПЗ

Арк.

47

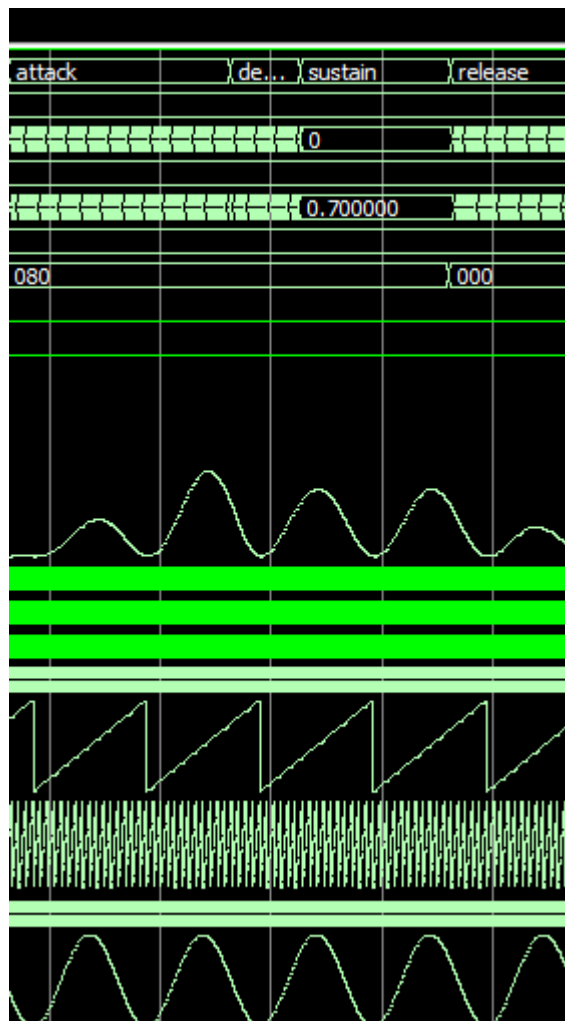


Рисунок 3.24 Симуляція проекту

Наступний рисунок 3.25 зображує дві натиснуті клавіші, частота другої у декілька разів більша за першу. Тому на перший сигнал наче наклали шум.



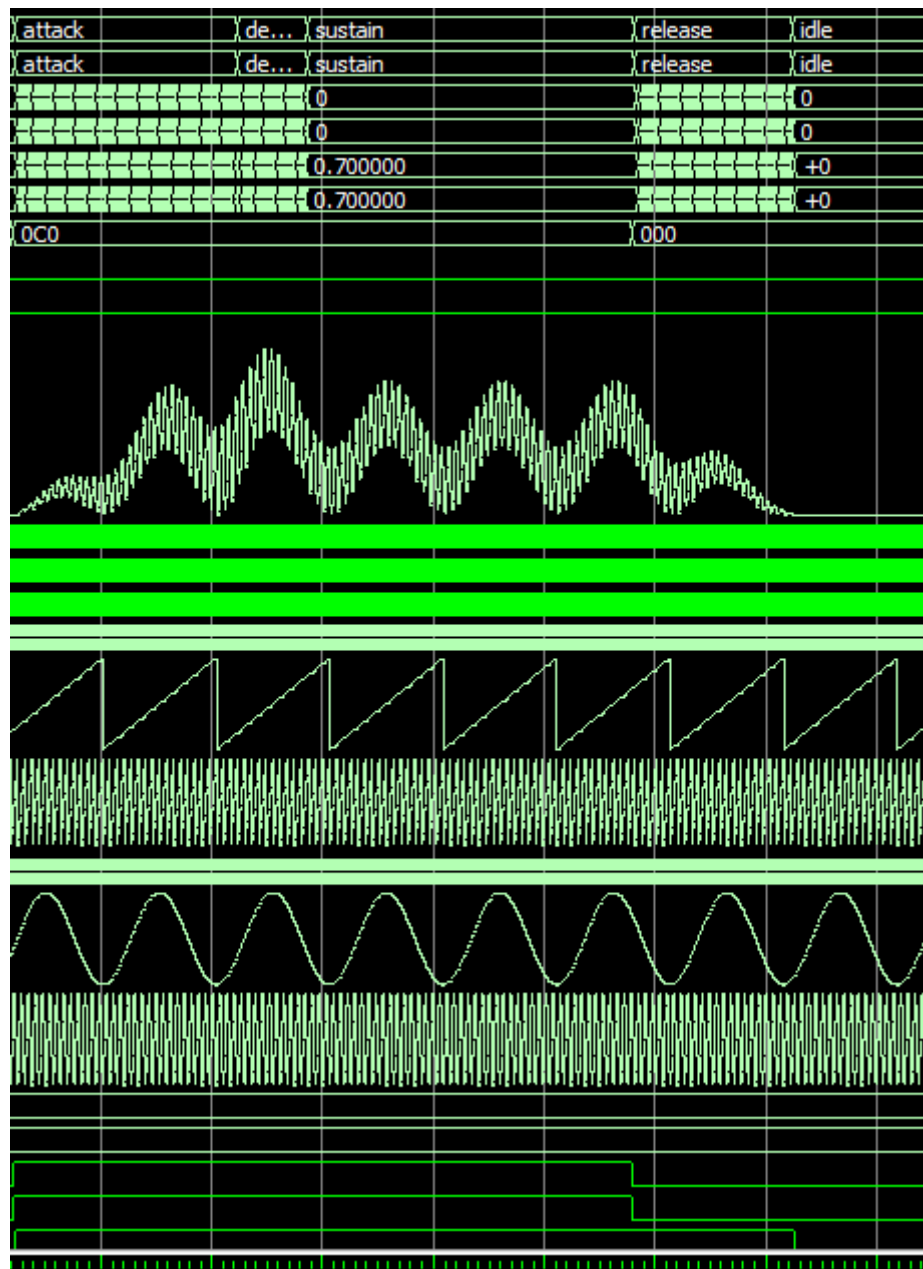


Рисунок 3.25 Симуляція проекту

У наступній симуляції, що зображена на рисунку 3.22 були збільшені значення часу атаки та інших станів, про це свідчить більш довгий сигнал.

На рисунку 3.26 зображений збільшений фрагмент діаграми.

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466454.003 ПЗ

Арк.

49

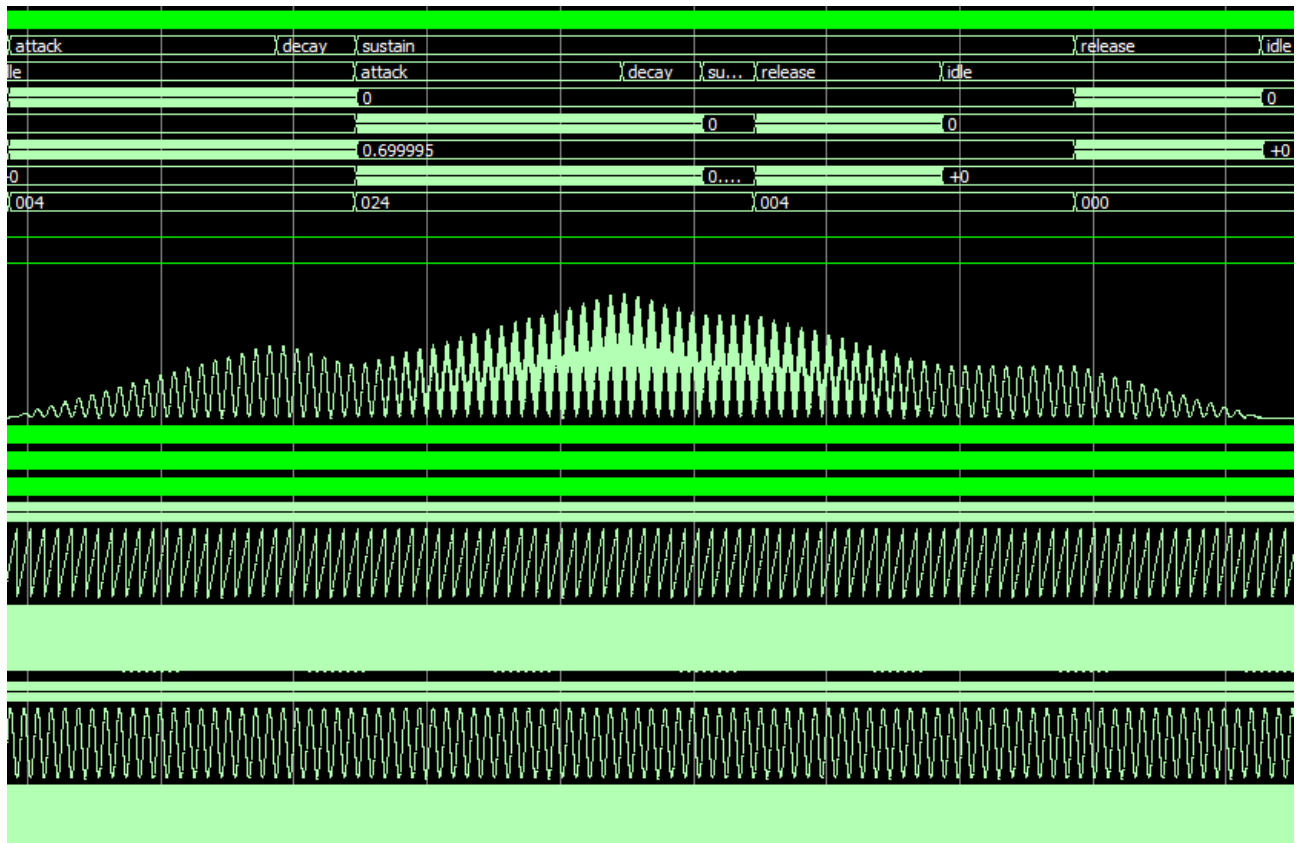


Рисунок 3.26 Симуляція проекту

Спочатку у стан атаки було переведено перший блок, на діаграмі вихідного сигналу видно зростання простої синусоїди. Після того як перший блок пройшов стани атаки та спадання і перейшов у стан утримання, була натиснута друга клавіша і другий блок переведено у стан атаки. Ця діаграма доводить повноцінну та коректну роботу як окремих модулів, так і усього проекту в цілому.

Далі, на рисунках 3.27 та 3.28, наведені діаграми сигналу інших форм (пила та меандр).

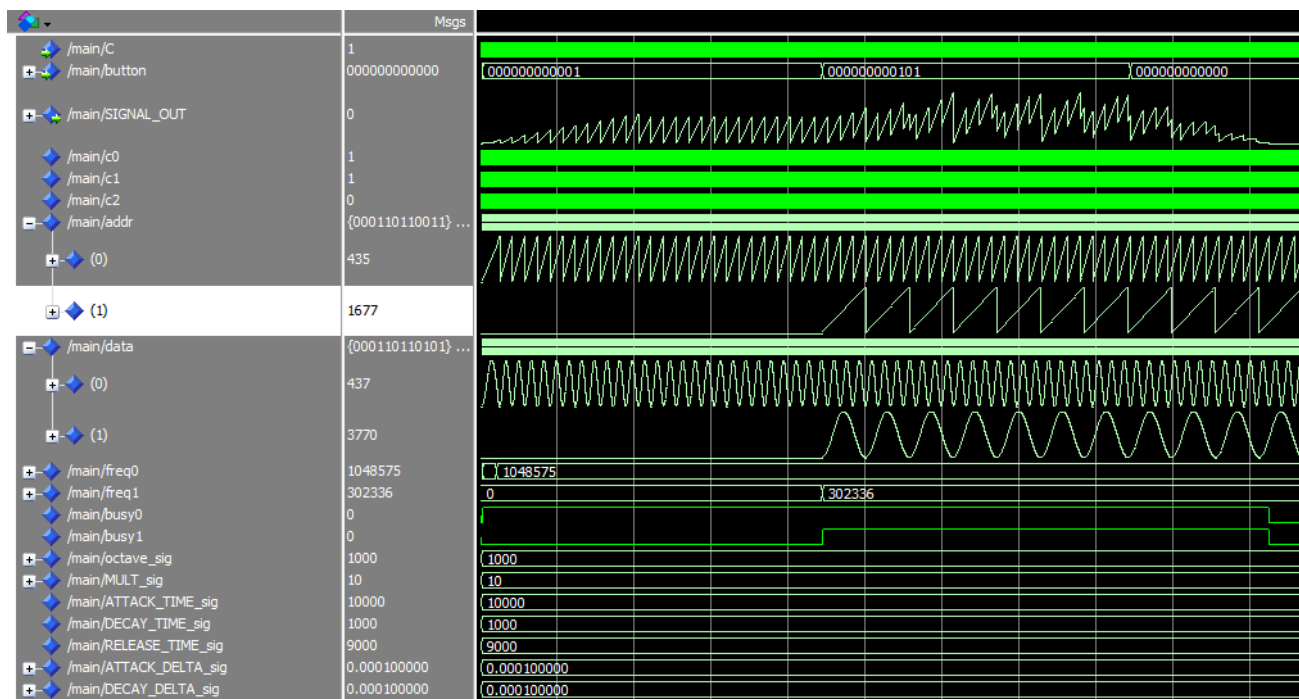


Рисунок 3.27 Симуляція проекту

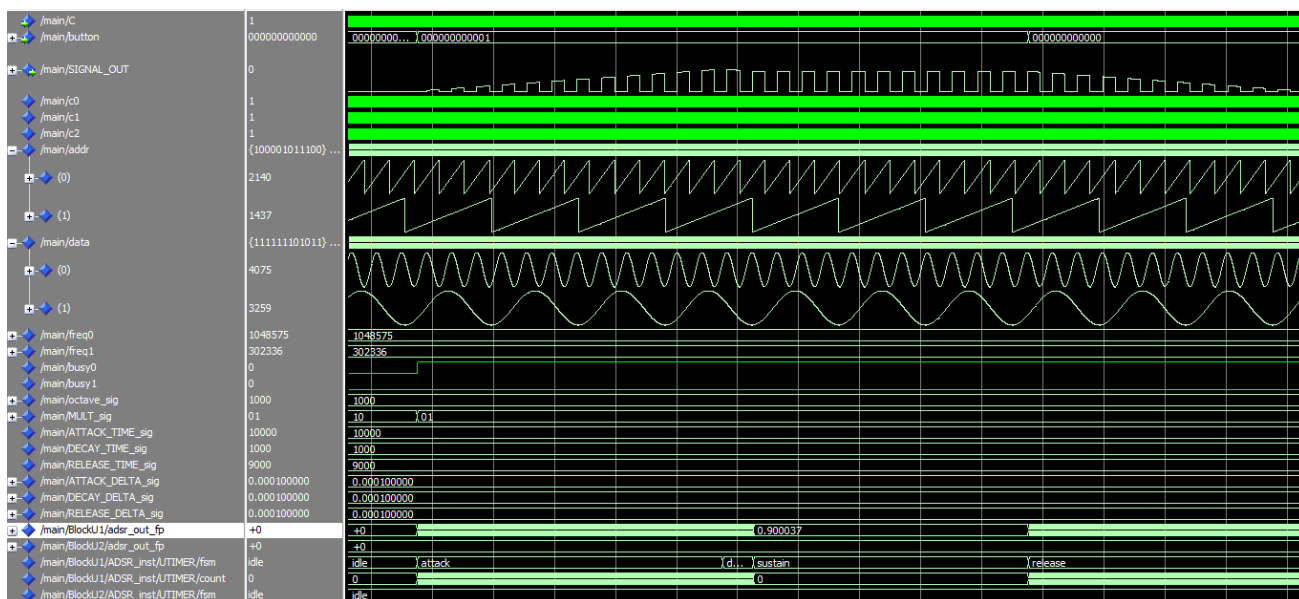


Рисунок 3.28 Симуляція проекту

### ВИСНОВКИ ДО РОЗДІЛУ 3

Увесь проект зайняв на кристалі ПЛІС:

4188 логічних елементів

2252 регістри

30 контактів ІО

98524 біти пам'яті ( $12 \text{ бітів довжина слова} * 2^{12} \text{ кількість слів} * 2 \text{ кількість модулів}$ )

14 вбудованих помножувачів з розрядністю 9 бітів

1 блок PLL

У цьому розділі було описано роботу усіх розроблених модулів. В залежності від обраної кількості каналів, проект буде займати різну кількість логічних елементів. Тестування проводилося для двох каналів. Звісно це є обмежена версія продукту. Маючи на кристалі більше логічних елементів у склад проекту можна включити до 12 каналів синтезу, що будуть обробляти кожну клавішу, цим саме можна звільнитися від арбітру клавіш. Також можна додати фільтри та різного роду ефекти. Маючи більше модулів пам'яті, можлива реалізація запису у цю пам'ять оригінальних сигналів замість звичайної синусоїди. Це дозволить синтезувати звук будь-якого реального інструменту.

					ІАЛЦ.466454.003 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

В результаті виконання проекту було розроблено схему синтезатору звукового сигналу. Даний пристрій розроблений на мові VHDL та може бути розміщений на будь-якій програмованій логічній схемі. Так як модулі цього синтезатора є незалежні, то даний продукт досить гнучкий, це дозволяє з легкістю додавати інші модулі обробки. Даний синтезатор може використовуватися початківцями для гри та звукозапису.

Було досліджено методи та технології синтезу сигналів та основні стадії їх обробки. Розглянуто особливості архітектури FPGA та було використані спеціалізовані ресурси схеми для даного проекту.

В результаті дослідження процесу синтезу, було описано та створено основні модулі.

Було зібрано тестовий синтезатор з двома незалежно генеруючими модулями та проведено усі можливі симуляції з різними параметрами. Дані симуляції показують бездоганні результати із швидкодії. Простота окремих модулів, що потім складаються у велику структуру забезпечила високу надійність схеми.

Отже ці результати підтверджують високу ефективність використання програмованих логічних інтегральних схем для цифрової обробки сигналів. Високі швидкості роботи та паралелізація процесів генерування сигналів дає виграш у порівнянні з виконанням тих самих дій на потужних мікропроцесорних системах.

Розвиток та тенденції ринку FPGA та SoC не дарма знаходять застосування у процесах обробки сигналів та виконання складних операцій з числами.

					ІАЛЦ.466454.003 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Cyclone IV Device Handbook Volume 1 [Електронний ресурс] – 2016. – Режим доступу до ресурсу :  
<https://www.intel.ru/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-iv/cyclone4-handbook.pdf>.
2. Cyclone IV Device Datasheet [Електронний ресурс] – 2-16. – Режим доступу до ресурсу :  
<https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-iv/cyiv-53001.pdf>
3. Звук как физическое явление [Електронний ресурс]. – Режим доступу: URL: <http://www.digitalmusicacademy.ru/lesson-sound-as-a-physical-phenomenon>
4. Тембр звука [Електронний ресурс]. – Режим доступу: URL: <http://www.digitalmusicacademy.ru/lesson-sound-timbre>
5. Основы звукового синтеза [Електронний ресурс]. – Режим доступу: URL: <http://www.digitalmusicacademy.ru/lesson-synthesis-fundamentals>
6. Аддитивный синтез [Електронний ресурс]. – Режим доступу: URL: <http://www.digitalmusicacademy.ru/lesson-additive-synthesis>
7. Осцилятор [Електронний ресурс]. – Режим доступу: URL: <http://www.digitalmusicacademy.ru/lesson-oscillator>
8. ADSR-оглабающая (Envelope Generator) [Електронний ресурс]. – Режим доступу: URL: <http://www.digitalmusicacademy.ru/lesson-adsr-envelope>
9. Обзор DSP процессоров [Електронний ресурс]. – Режим доступу: URL: <http://www.russianelectronics.ru/leader-r/review/2192/doc/48223/>
10. Смаглий Г. А. Маловик Л. В. Основы теории музыки. -Харків, 2004.
11. Побережна Г. І., Щериця Т. В. Загальна теорія музики. К: Вища школа, 2004–303 с.

					ІАЛЦ.466454.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

12. Волновые формы [Электронный ресурс]. – Режим доступа: URL:  
[http://virartech.ru/articles/wave\\_forms.php](http://virartech.ru/articles/wave_forms.php)
13. Музыкальная энциклопедия. — М.: Советская энциклопедия, 1981. —  
Т. 5. — С. 488—490. — 1056 с
14. Clive Maxfield. The Design Warrior`s Guide to FPGAs. – Newnes, 2004. –  
542 с.
15. Uwe Meyer-Baese. Digital Signal Processing with Field Programmable  
Gate Arrays . – Springer, 3rd edition . – 744 с.

					ІАЛЦ.466454.003 ПЗ	Арк.
						55
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток А

					ІАЛЦ.466454.003 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		



Додаток Б

					ІАЛЦ.466454.003 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток В

					ІАЛЦ.466454.003 ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Г

					ІАЛЦ.466454.003 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		