



**Комплекс гидролокационный Гидра**

**Программное обеспечение**

**Руководство системного программиста**

**Прямой цифровой синтез сигналов (DDS)  
Редакция 14 (r14)**

Разработал	М.В. Седов
Статус	Внутренняя документация

## Содержание

	Стр.
<b>1. Общее описание</b> .....	4
<b>1.1. База данных ЗИ</b> .....	6
<b>1.2. Общий алгоритм работы</b> .....	7
<b>1.3. Формирование сигналов</b> .....	8
1.3.1. Расчет параметров DDS и накопления .....	10
<b>1.4. Использование КУ в режиме 0 (Mode=0, версия _v1)</b> .....	14
<b>1.5. Использование КУ в режиме ШИМ1 (Mode=1, версия _v2)</b> .....	16
<b>1.6. Использование КУ в режиме ШИМ2 (Mode=2, версия _v3)</b> .....	19
1.6.1. Регулировка шимирования в режиме ШИМ2 .....	24
1.6.2. Формирование ФЗМ в режиме ШИМ2 (_v3) .....	26
1.6.3. Формирование кода Баркера в режиме ШИМ2 (_v3) .....	26
<b>1.7. Метод прямого цифрового синтеза (генератор GCHIRP, версия _v4)</b> .....	34
<b>1.8. Метод прямого цифрового синтеза (генератор GDDS5, версия _v5)</b> .....	61
1.8.1. Логика работы и упаковка данных .....	61
<b>2. Файл данных DDS</b> .....	66
<b>3. Файл образа ЗИ (PPI файл)</b> .....	70
<b>3.1. Формат имени PPI файла</b> .....	70
<b>3.2. Формат данных PPI файла</b> .....	70
<b>3.2.1. Формат gPPI_ZI</b> .....	71
<b>3.2.2. Формат массива образа ЗИ</b> .....	73
<b>3.2.3. Использование различных полей</b> .....	74
<b>4. Контроль данных файлов DDS и PPI</b> .....	76
<b>5. Данные FLASH КОНГ</b> .....	77
5.1. Данные DDS (DataDDS) .....	78
<b>Приложение А (обязательное). Упаковка массива КУ для генератора GCHIRP</b> .....	79
<b>Приложение Б (обязательное). Упаковка массива КУ для генератора DDS5</b> .....	80
<b>Приложение В (обязательное). Проработка альтернативных режимов работы для DDS5</b>	82

## Сокращения

<b>ВС</b>	Ведущая система
<b>БПП</b>	Блок приема-передачи
<b>ГЛИ</b>	Гидролокационная информация
<b>ЗИ</b>	Зондирующий импульс
<b>КОНГ</b>	Контроллер КОНГ
<b>ЛЧМ</b>	Линейная частотная модуляция
<b>ПО</b>	Программное обеспечение
<b>ППФ</b>	Параметрический профилограф
<b>ФЗМ</b>	Фазо-задержанная модуляция
<b>ЦФД</b>	Цифровой фильтр-дециматор
<b>DDS</b>	Прямой цифровой синтез (Direct Digital Synthesis)
<b>Char</b>	Формат данных char (язык C) – 1 байт (значения от -127 до +128)
<b>Double</b>	Формат данных Double (язык C) – 8 байт
<b>Double[n]</b>	Массив данных типа Double, содержащий n данных. Размер массива = n*8 байт
<b>Fadc</b>	Частота отсчетов АЦП
<b>Fc</b>	несущая излучаемая частота ЗИ
<b>Fclk</b>	Опорная частота
<b>FLASH</b>	Внешняя FLASH контроллера КОНГ
<b>Float</b>	Формат данных Float (язык C) – 4 байта

<b>Fs</b>	Частота отсчетов при передаче в ВС
<b>PGL</b>	Процессор гидролокационный контроллера КОНГ
<b>Tadc</b>	Период отсчетов АЦП
<b>Tc</b>	Период несущей излучаемой частоты ЗИ
<b>Tclk</b>	Период опорной частоты
<b>Tclks</b>	Период тактовой синхрочастоты PGL
<b>U8</b>	Формат данных Unsigned char (язык C) – 1 байт (значения от 0 до 255)
<b>Uchar[n]</b>	Массив данных типа Uchar, содержащий n байт. Размер массива = n байт
<b>U32</b>	Формат данных Unsigned long (язык C) – 4 байта (значения от 0 до $2^{32}-1$ )
<b>Ulong[n]</b>	Массив данных типа Ulong, содержащий n двойных слов. Размер массива = 4*n байт
<b>U16</b>	Формат данных Unsigned short (язык C) – 2 байта (значения от 0 до 65535)
<b>Ushort[n]</b>	Массив данных типа Ushort, содержащий n слов. Размер массива = 2*n байт
<b>Wchar</b>	Символ в UNICODE (2 байта на символ). Размер = 2 байта
<b>Wchar[n]</b>	Строка символов UNICODE (2 байта на символ), заканчивающаяся символом конца строки (код 0000h). Размер строки = (n)*2 байт. Кол-во символов в строке = n-1

Данный документ содержит:

1. описание логики работы с DDS
2. описание формата файла данных DDS (DDS файл)
3. описание формата файла образа зондирующего импульса (ЗИ) – pilot pulse image's file (PPI файл)
4. описание упаковки данных во внешней FLASH контроллера КОНГ

## История редакций документа

- Редакция 1(r1) – начальная редакция
- Редакция 2 (r2) – добавлена поддержка для ППФ
- Редакция 3 (r3) – добавлена поддержка ЦДФ
- Редакция 4 (r4) – добавлена поддержка режима ШИМ внутри полупериода
- Редакция 5 (r5) – добавлена поддержка режима ШИМ2 внутри полупериода
- Редакция 6 (r6) – исправления и уточнения
- Редакция 7 (r7) – добавлена описание ФКМ
- Редакция 8 (r8) – добавлена описание ФЗМ
- Редакция 9 (r9) – добавлено описание использования параметров при работе с ЦДФ
- Редакция 10 (r10) – добавлено описание использования отсчетов образца для ППФ (07.2015)
- Редакция 11 (r11) – добавлено описание использования генератора GCHIRP (03.2016)
- Редакция 12 (r12) – добавлено описание использования генератора DDS5 (06.2017)
- Редакция 13 (r13) – добавлено описание типа амплитуда в поле gPRM\_ZI.SampleType, gPPI\_ZI.SampleType и gPRM\_ZI.Rzv2 при работе с H5p1 (03.2018)
- Редакция 14 (r14) – утверждена и зафиксирована упаковка данных для ppi версии 2 (04.2018)

## 1. Общее описание

DDS используется для формирования ЗИ с заданными параметрами.

Для работы с DDS используются:

1. Файлы данных DDS (DDS файлы)
2. Файлы образа ЗИ (PPI файлы)
3. База данных ЗИ (БДЗИ)
4. Внешняя FLASH контроллера КОНГ
5. Набор ЗИ
6. Программы для расчета сигналов
7. Программа HyScan

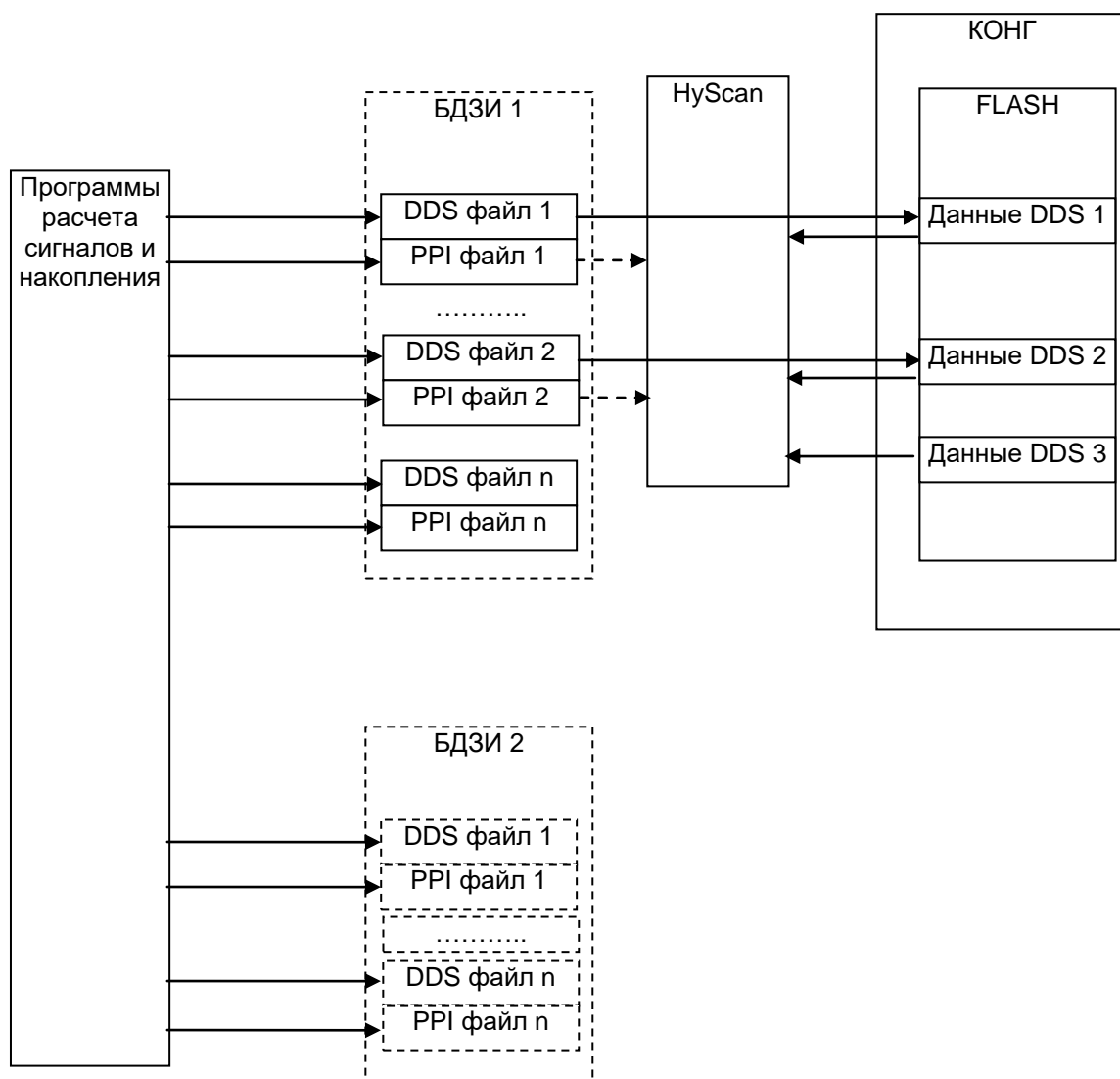


Рисунок 1. Файлы и данные DDS

Всего возможно использование до 65534 различных ЗИ.

Каждый ЗИ имеет набор уникальных параметров и соответственно – уникальный код, позволяющий однозначно определить данный ЗИ. При изменении любого из параметров ЗИ меняется его код. Каждому файлу образа ЗИ соответствует свой файл данных DDS, который содержит данные для формирования этого ЗИ методом DDS в PGL. Совместно с файлом DDS в ПО ВС может использоваться PPI файл, если ПО не использует самостоятельного расчета образа ЗИ по данным из DDS файла.

ЗИ бывают простыми и сложными. Простым ЗИ называется ЗИ, не требующий дополнительной обработки получаемой ГЛИ при приеме для определения амплитуды огибающей входного сигнала (кроме расчета амплитуды). Сложным называется ЗИ, которые требуют дополнительной обработки для определения амплитуды огибающей входного сигнала (например – свертка). К простым ЗИ относятся ЗИ с постоянной несущей частотой (тоновые ЗИ или тон). К сложным ЗИ относятся остальные ЗИ.

Для ЗИ, загружаемых во FLASH, код ограничивается объемом FLASH (см 5).

Для ЗИ, загружаемых в PGL, код ЗИ может быть любым внутри допустимого диапазона.

Параметры ЗИ связаны с параметрами накопления ГЛИ, поэтому рассчитываются одновременно в программах расчета сигналов.

Формирование сигналов обеспечивает модуль GZI в PGL контроллера КОНГ, использующий для синтеза записанные во внешнюю FLASH данные.

Каждая реализация ЗИ имеет код версии (не путать с версией упаковки данных), т.е. DDS файл имеет свою версию реализации и PPI файл имеет свою версию реализации. Кол-во реализаций каждого файла может быть различным (например, из-за исправления ошибок). Выполнение новой реализации PPI файла не требует перезаписи данных FLASH, новая реализация DDS файла требует записи новой реализации во FLASH.

При изменении любого файла (добавлении новых файлов или удаление старых) меняется и версия БДЗИ.

Код реализации ЗИ может иметь значения от 1 до 254 (Uchar).

Код версии БДЗИ может иметь значения от 1 до  $2^{32}-2$  (ULong).

БДЗИ является универсальной и используется для работы со всеми исполнениями комплекса. Т.к. в каждом исполнении комплекса используется только часть данных (набор ЗИ) их БДЗИ, то существует также версия набора ЗИ, хранящегося во FLASH.

Код набора ЗИ может иметь значения от 1 до 65534 (Ushort).

Код набора используется только для контроля версий и необходимости обновления.

## 1.1. База данных ЗИ

БДЗИ является единой базой для хранения информации о всех ЗИ, используемых в комплексе.

БДЗИ поставляется в составе ПО HyScan и ПО вторичной обработки.

БДЗИ используется при работе HyScan для загрузки данных образов ЗИ а также для обновления данных образов DDS во FLASH БПП. БДЗИ используется при работе ПО вторичной обработки для загрузки данных образов ЗИ.

Для всех исполнений комплекса используется одна и та же БДЗИ текущей версии.

БДЗИ содержит файлы нескольких типов:

- Файлы образа ЗИ
- Файлы данных DDS
- Файл описания БДЗИ

Каждый ЗИ описывается с помощью двух файлов:

- Файл образа ЗИ
- Файл данных DDS

При загрузке файлов образов ЗИ ПО должно проверять соответствие версии файлов образа ЗИ и соответствующую версию этого файла в файле описания БД.

При работе с комплексом ПО должно проверять текущие версии данных DDS во FLASH БПП и при необходимости предлагать оператору заменить старые данные DDS на более новые.

Для обновления БДЗИ достаточно переписать все новые файлы образов ЗИ, DDS и файл описания БД.

Файлы БДЗИ хранятся в отдельной папке.

Т.к. файлы имеют в названии код версии (имя файла уникально), в одной папке могут храниться файлы нескольких версий БДЗИ.

Разные версии БДЗИ могут содержать одни и те же версии файлов образа ЗИ и файлов данных DDS.

Текущей версией БДЗИ считается БДЗИ с наибольшим доступным кодом версии БДЗИ (код версии БДЗИ находится в имени файла описания БДЗИ).

Например:

**Gidra\DBZI\**

00001\_Tone\_1.ppi

00002\_Tone\_1.ppi

00001\_Tone\_1.dds

00002\_Tone\_1.dds

.....

00001\_Tone\_2.ppi

00002\_Tone\_2.ppi

00001\_Tone\_2.dds

00002\_Tone\_2.dds

.....

## 1.2. Общий алгоритм работы

На этапе настройки комплекса определяются рабочие частоты и параметры ЗИ

С помощью программ расчета для каждого ЗИ формируется соответствующие DDS и PPI файлы. Также возможно использование ранее рассчитанных ЗИ, хранящихся в БДЗИ.

Полученные новые файлы записываются в БДЗИ.

Далее формируется необходимый набор ЗИ, который будет использоваться в данном комплексе. На основе набора формируется соответствующий файл описания комплекса.

С помощью программы HyScan и сформированного CDF файла осуществляется запись во внешнюю FLASH КОНГ необходимых данных DDS.

В дальнейшем данные во FLASH могут быть обновлены или дополнены новыми ЗИ.

DDS, PPI и CDF файлы входят в состав ПО комплекса.

Во время работы с комплексом программа HyScan, на этапе инициализации, определяет количество и тип используемых в комплексе ЗИ на основе считанных из FLASH данных.

Для простых ЗИ записанные во FLASH данные позволяют полностью определить параметры ЗИ и дополнительных данных для работы с такими ЗИ не требуется. Для сложных ЗИ дополнительные данные (образец для свертки) рассчитывается либо на основе считанных из FLASH данных или загружается готовый из PPI файла. Соответствие принадлежности файлов и возможные ограничения на использования ЗИ (например, минимальная скважность излучения) определяется через CDF файл.

Обновление данных

Обновление данных может быть в двух случаях:

1. Обнаружена ошибка в данных DDS, которая не позволяет использовать соответствующий ЗИ (или набор ЗИ)
2. Добавлен новый ЗИ

В обоих случаях обеспечивается удаленное обновление БДЗИ у пользователя комплекса. Для этого ему высылаются новая БДЗИ и новый CDF файл. С помощью программы HyScan осуществляется обновление данных.

### 1.3. Формирование сигналов

При задании параметров ЗИ и его формировании используются следующие правила (Рисунок 2):

- 1) Длительность одного тика равна  $T_{clk}$  (10 нс для 100 МГц, 5нс для 200 МГц и т.д.).
- 2) Длительность любого интервала при формировании ЗИ кратна  $T_{clk}$
- 3) Формирование ЗИ осуществляется только целыми периодами.
- 4) Периоды внутри ЗИ обозначаются  $Prd[0]..Prd[n]$
- 5) ЗИ начинается с формирования мерфазы А периода  $Prd[0]$  и заканчивается формированием фазы В периода  $Prd[n]$
- 6) Внутри периода длительности полупериодов равны.
- 7) Длительность мертвого времени между фазой А и В одинакова для каждого полупериода и не меняется в процессе формирования ЗИ.

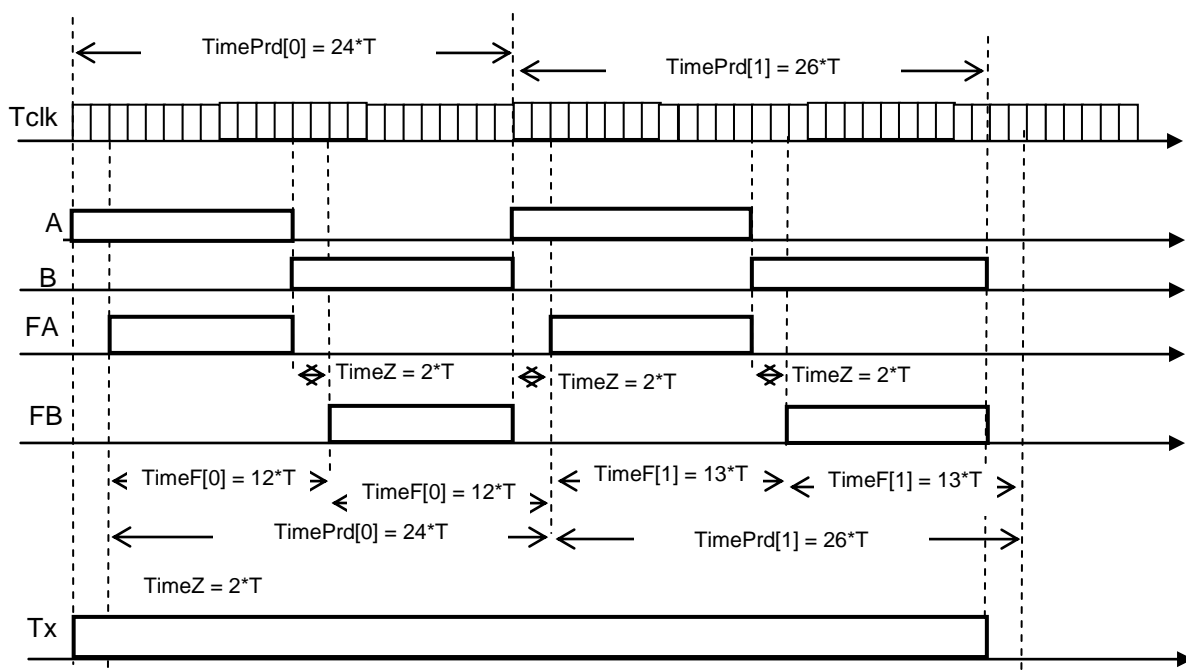


Рисунок 2. Пример формирования ЗИ (для всех исполнений кроме ППФ)



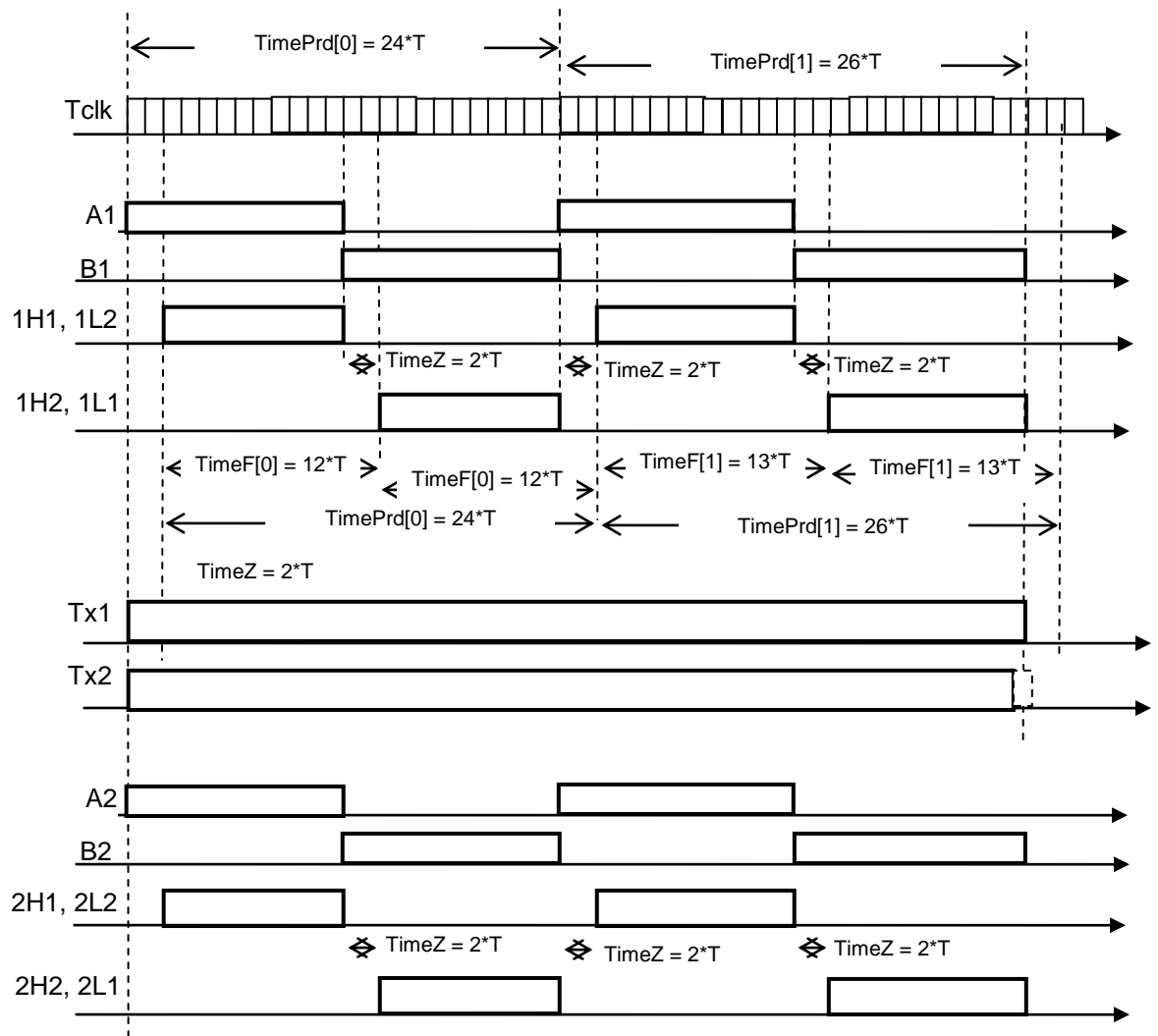


Рисунок 3. Пример формирования ЗИ (для исполнения ППФ)

Принятые обозначения и сокращения:

$T = T_{\text{clk}}$  – длительность одного тика

$\text{TimePrd}[i]$  – длительность периода  $i$ ,  $i = \{0, 1, \dots, n\}$

$\text{TimeFA}[i]$  – длительность первой половины полупериода (фаза A) периода  $i$

$\text{TimeFB}[i]$  – длительность второй половины полупериода (фаза B) периода  $i$

$\text{TimeFA}[i] = \text{TimeFB}[i] = \text{TimeF}[i]$

$\text{TimePrd}[i] = \text{TimeFA}[i] + \text{TimeFB}[i] = 2 \cdot \text{TimeFA}[i] = 2 \cdot \text{TimeFB}[i] = 2 \cdot \text{TimeF}[i]$

$\text{TimeF}[i] = N \cdot T_{\text{tick}} = N \cdot T$

$\text{TimeZ} = M \cdot T_{\text{tick}} = M \cdot T$

Минимально возможный шаг изменения периода равен  $2 \cdot T_{\text{clk}}$

Т.к. для реализации комплексных отсчетов (выборки Re-Im) используется частота  $F_{\text{im}}$ , равная  $4 \cdot F_{\text{нес}}$ .

Для тона:

$F_{\text{im}} = 4 \cdot F_{\text{нес}} = 4 \cdot (1/\text{TimePrd})$ ,

$T_{\text{im}} = 1/F_{\text{im}} = \text{TimePrd}/4 = K \cdot T/4$

Т.е. длительность  $\text{TimePrd}$  для тона должна быть кратна  $4 \cdot T$ , поэтому минимально возможный шаг изменения  $\text{TimePrd}$  для тона равен  $4 \cdot T$ .

### 1.3.1. Расчет параметров DDS и накопления

Исходными данными для расчета являются:

- 1) Несущая частота  $F_c$  (для тона)
- 2) Начальная и конечные частоты  $F_{c\_start}$  и  $F_{c\_end}$  (для ЛЧМ)
- 3) Длительность ЗИ
- 4) Длительность мертвого времени  $T_z$
- 5) Частота отсчетов  $F_s$
- 6) Режим формирования полупериода (обычный или ШИМ)

Существует два режима накопления – без использования ЦФД и с использованием ЦФД. Без использования ЦФД выходные отсчеты формируются из отсчетов АЦП без изменений. При работе ЦФД отсчеты АЦП поступают на ЦФД, а выходные отсчеты берутся с выхода ЦФД.

Без использования ЦФД отсчеты АЦП формируются в виде прореженных отсчетов, формирующих пару комплексных отсчетов  $Re-Im$ . Данная пара отражает один отсчет по дальности, который следует с частотой  $F_s$  (Рисунок 4).

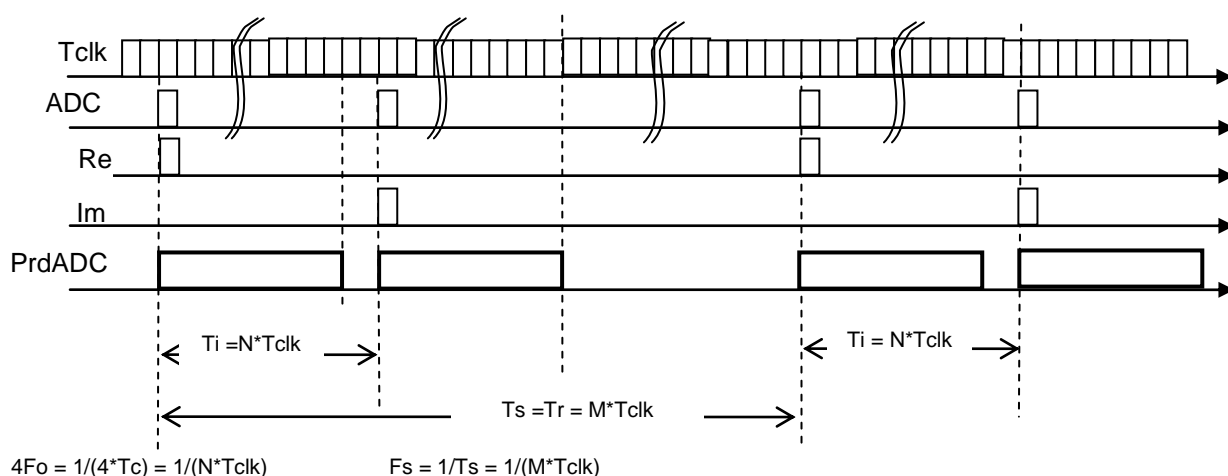


Рисунок 4. Формирование отсчетов  $Re-Im$  при накоплении без использования ЦФД (для всех исполнений кроме ППФ)

Значение параметра  $TimeR$  определяет период отсчетов по дальности (частоту  $F_s$ ) - Рисунок 4 и рассчитывается по формуле:

$$TimeR = [ Ts / Tclk ] - 3 = [ 1000000 / (Fs * Tclk) ] - 3,$$

где:

$TimeR$  – значение параметра  $TimeR = \{ TimeR\_min..65535 \}$

$Ts$  – период частоты  $F_s$  (нс)

$F_s$  – значение частоты  $F_s$  (кГц)

$Tclk$  – период опорной частоты (нс), определяется версией PGL

$TimeR\_min$  – минимальное значение параметра  $TimeR$  (определяется версией PGL)

$$(TimeR - TimeI) \geq 33$$

$TimeR\_min = TimeI + 33$ , где:

$TimeI$  – значение параметра  $TimeI$

ПРИМЕР. Для получения частоты  $F_s = 37,5$  кГц вычисляем:

$$TimeR = [ 1000000 / (37,5 * 10) ] - 3 = [ 2666,67 ] - 3 = 2666 - 3 = 2663$$

ПРИМЕЧАНИЯ.

- 1) Округление результатов деления осуществляется в меньшую сторону (для получения максимальной ближайшей частоты по отношению к необходимой).
- 2) Период частоты  $F_s$  может меняться с шагом в один  $Tclk$

- 3) Период частоты  $F_s$  должен быть таким, чтобы  $(T_s - T_i) \geq T_{adc\_min}$  (чтобы не превышать минимальный период опроса АЦП)

Значение параметра  $T_{imel}$  определяет период частоты  $4F_o$  - Рисунок 4 и рассчитывается по формуле:

$$T_{imel} = \lceil Prd\_4F_o / T_{clk} \rceil - 2 = \lceil 1000000 / (4F_o * T_{clk}) \rceil - 2,$$

где:

$T_{imel}$  – значение параметра  $T_{imel} = \{ T_{imel\_min}..65535 \}$

$Prd\_4F_o$  – период частоты  $4F_o$ , нс

$4F_o$  – значение частоты  $4F_o = 4 * F_{нес}$  (кГц)

$T_{clk}$  – период опорной частоты, нс (в зависимости от используемой версии PGL)

$T_{imel} \geq T_{imel\_min}$

$T_{imel\_min}$  – минимальное значение параметра  $T_{imel}$

$T_{imel\_min} = 33$

ПРИМЕР. Для получения частоты  $4F_o=1000$  кГц получаем:

$$T_{imel} = \lceil 1000000 / (1000 * 10) \rceil - 2 = \lceil 100 \rceil - 2 = 100 - 2 = 98$$

ПРИМЕЧАНИЯ.

- 1) Округление результатов деления осуществляется в меньшую сторону (для получения максимальной ближайшей частоты по отношению к необходимой).
- 2) Период частоты  $4F_o$  может меняться с шагом в один  $T_{tick}$
- 3) Период частоты  $4F_o$  должен быть таким, чтобы  $T_{imel} \geq Prd_{ADC\_min}$  (чтобы не превышать минимальный период преобразования АЦП)
- 4) Для тона значение несущей частоты  $F_{нес} = F_o$  должно быть строго равно  $\frac{1}{4}$  от  $4F_o$ :

$$4F_o = 4 * F_{нес}$$

$$Prd\_F_{нес} = Prd\_F_o = 4 * Prd\_4F_o$$

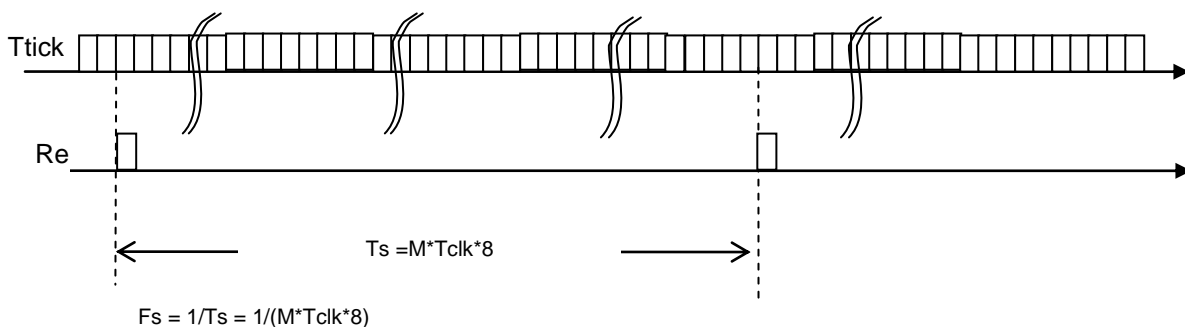
$$Prd\_F_o = 2 * (K * T_{tick}) = 4 * Prd\_4F_o = 4 * (L * T_{tick})$$

$K$  – кол-во тиков в одном полупериоде частоты  $F_o$ , длительность одного периода равна  $2 * K$  тиков

$L$  – кол-во тиков в одном периоде частоты  $4F_o$

$K = 2 * L$ ,  $L = K/2$ , значение  $K$  должно быть кратно 2

Из этого следует, что минимальный шаг изменения периода  $F_{нес}$  для тона равен  $4 * T_{tick}$ , длительность периода  $F_{нес}$  должна быть кратна  $4 * T_{tick}$



**Рисунок 5. Формирование отсчетов Re при накоплении без использования ЦФД (для ППФ)**

Для ППФ используется только один параметр – TimeR. Значение параметра TimeR определяет период отсчетов по дальности (частоту  $F_s$ ) - **Рисунок 5** и рассчитывается по формуле:

$$\text{TimeR} = \lceil Ts / (8 * Tclk) \rceil - 3 = \lceil 1000000 / (Fs * 8 * Tclk) \rceil - 3,$$

где:

**TimeR** – значение параметра TimeR = { TimeR\_min..65535 }

**Ts** – период частоты  $F_s$  (нс)

**Fs** – значение частоты  $F_s$  (кГц)

**Tclk** – период опорной частоты, нс (в зависимости от используемой версии PGL)

**TimeR\_min** – минимальное значение параметра TimeR

**Fs\_max** = 100 кГц, **TimeR\_min** = 123

ПРИМЕР. Для получения частоты  $F_s=57$  кГц вычисляем:

$$\text{TimeR} = \lceil 1000000 / (57 * 8 * 10) \rceil - 3 = \lceil 219,29 \rceil - 3 = 219 - 3 = 216$$

ПРИМЕЧАНИЯ.

- 1) Округление результатов деления осуществляется в меньшую сторону (для получения максимальной ближайшей частоты по отношению к необходимой).
- 2) Период частоты  $F_s$  может меняться с шагом в  $8 * Ttick$

Значение параметра TimeZ рассчитывается по формуле:

$$\text{TimeZ} = \lceil \text{Time} / Ttick \rceil - 2,$$

где:

**TimeZ** – значение параметра TimeZ

**Time** – требуемая длительность мертвого времени (в нс), мин. значение Time= 20 нс

**Tclk** – длительность тика (в нс)

Значение параметра TimeZ должно лежать в диапазоне 0..65535

ПРИМЕЧАНИЯ.

- 1) Изменение длительности TimeZ возможно с шагом Tclk.
- 2) Длительность мертвого времени не должна превышать длительность формируемого полупериода. Если длительность мертвого времени оказывается больше длительности формируемого полупериода, при суммировании текущее значение длительности периода (в тиках) оказывается больше чем  $65535 * 2$ , формирование ЗИ останавливается и формируется признак ошибки.
- 3) Если при вычитании текущее значение длительности периода (в тиках) оказывается менее 46, формирование ЗИ останавливается и формируется признак ошибки.

Значение параметра NumPrd рассчитывается по формуле:

$$\text{NumPrd} = \text{Num} - 1,$$

где:

NumPrd – значение параметра NumPrd

Num – требуемое кол-во формируемых периодов несущей внутри ЗИ, минимальное значение Num=1

Значение параметра NumPrd должно лежать в диапазоне  $0..2^{32}-1$

При использовании ЦФД отсчеты АЦП являются равномерными. Комплексные отсчеты Re-Im формируются только на выходе ЦФД (Рисунок 6).

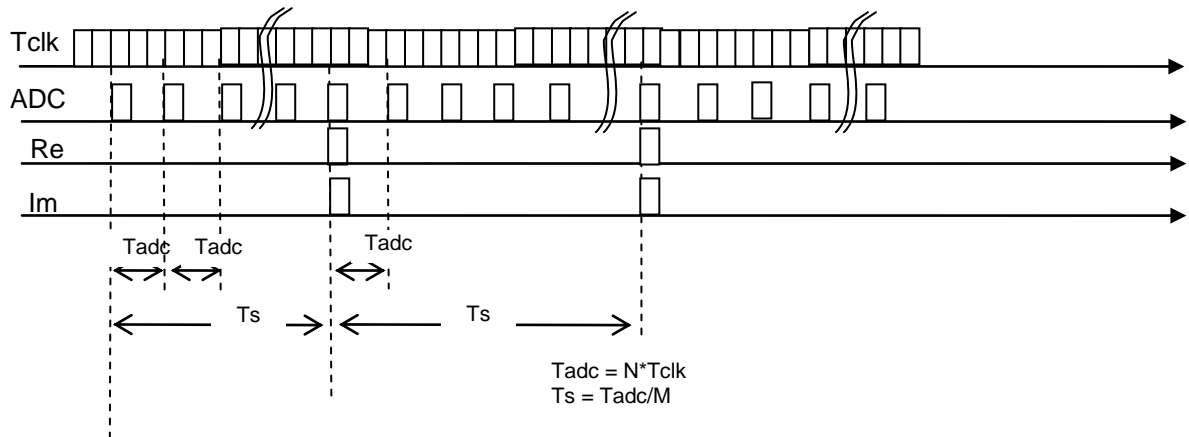


Рисунок 6. Формирование отсчетов Re-Im при использовании ЦФД (для всех исполнений кроме ППФ)

$$T_i = T_{adc}$$

$$T_r = 2 \cdot T_{adc} = 2 \cdot T_i$$

$$T_{adc} = N \cdot T_{clk}, N - \text{целое}$$

$$T_s = T_{adc} / M = N \cdot T_{clk} / M, M - \text{коэффициент децимации (целое)}$$

$$TimeI = (T_{adc} / T_{clk}) - 2 = N - 2$$

$$TimeR = (2 \cdot T_{adc} / T_{clk}) - 3 = 2 \cdot N - 3,$$

где:

$T_{adc}$  – период взятия отсчетов АЦП, нс

$T_{clk}$  – период опорной частоты, нс (в зависимости от используемой версии PGL)

#### 1.4. Использование КУ в режиме 0 (Mode=0, версия\_v1)

**Всего возможно использование до 512 КУ (CC0..CC511).**

Таблица 1 – Формат массива КУ (DDS\_CC), размер равен 2\*N байт (N – кол-во используемых КУ)

Название	Размер (байт)	Описание	Примечание
CC[0]	2	КУ[0]	Команда SetPrd
CC[1]	2	КУ[1]	Команда DeltaPrd
...	2*N-4	КУ[2]..КУ[510]	Команды DeltaPrd
CC[511]	2	КУ[511]	Команда DeltaPrd

Для формирования ЗИ используются следующие КУ:

- 1) Команда установки длительности текущего периода (SetPrd)
- 2) Команда изменения длительности периода и повторения формирования заданного кол-ва периодов (DeltaOut)

Каждая КУ имеет размер 16 бит (2 байта) и позволяет задать параметры формирования одного или сразу нескольких периодов ЗИ.

Первой КУ (CC[0]) должна быть всегда КУ загрузки длительности полупериода (SetPrd). Значение команды CC[0] трактуется всегда как параметр команды SetPrd и не зависит от режима работы генератора. Дальнейший набор команд зависит от режима работы генератора (параметра Mode).

Интерпретация дальнейших КУ (CC[1]..CC[n]) зависит от режима работы генератора. При Mode=0 в качестве остальных КУ возможно только использование КУ DeltaRep.

КУ SetPrd позволяет задать параметры формирования только одного периода.

После формирования первого периода возможно изменить длительность следующего формируемого периода в большую или меньшую сторону и задать кол-во таких формируемых периодов (кол-во повторений). Если изменения длительности происходят каждый период, кол-во повторений равно 1.

Таблица 2 – Формат КУ при Mode=0

Команда	Биты															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>SetPrd</b>	TimePrd															
<b>DeltaRep</b>	Rep										Sign		Delta			

КУ SetPrd имеет только один параметр – TimePrd.

КУ DeltaRep имеет три параметра – Rep, Sign и Delta.

Значение параметра TimePrd рассчитывается по формуле:

$$\text{TimePrd} = [ \text{Time} / (2 * \text{Ttick}) ] - 2,$$

где:

TimePrd – значение параметра команды SetPrd

Time – требуемая длительность первого периода несущей (в нс)

Ttick – длительность тика = Tclkс (в нс)

Значение параметра TimePrd должно лежать в диапазоне 20..65535

НАПРИМЕР:

1) Для формирования частоты 500 кГц (период = 2 мкс = 2000 нс)

$$\text{TimePrd} = [ 2000 \text{ нс} / (2 * 10 \text{ нс}) ] - 2 = 100 - 2 = 98$$

2) Для формирования частоты 250 кГц (период = 4 мкс = 4000 нс)

$$\text{TimePrd} = [ 4000 \text{ нс} / (2 * 10 \text{ нс}) ] - 2 = 200 - 2 = 198$$

Значение параметра Delta рассчитывается по формуле:

$$\text{Delta} = [ \text{Time} / (2 * \text{Ttick}) ],$$

где:

Delta – значение параметра Delta команды DeltaRep

Time – требуемое значение изменения периода по модулю (в нс)

Ttick – длительность тика (в нс)

Значение параметра Delta должно лежать в диапазоне 0..31

ПРИМЕЧАНИЯ.

- 1) Изменение периода возможно только с шагом  $2 \cdot Ttick$ .
- 2) При Delta=0 – изменение длительности периода не происходит
- 3) Если при суммировании текущее значение длительности периода (в тиках) оказывается больше чем  $65535 \cdot 2$ , формирование ЗИ останавливается и формируется признак ошибки.
- 4) Если при вычитании текущее значение длительности периода (в тиках) оказывается менее 46, формирование ЗИ останавливается и формируется признак ошибки.

Значение параметра Sign задает знак изменения длительности периода:

=1 – период увеличивается на заданное значение (уменьшение несущей частоты)

=0 - период уменьшается на заданное значение (увеличение несущей частоты)

Значение параметра Rep рассчитывается по формуле:

**Rep = Num,**

где:

Rep – значение параметра Rep команды DeltaRep

Num – требуемое значение повторов = {1..1023}

Значение параметра Rep должно лежать в диапазоне 1..1023

ПРИМЕЧАНИЯ.

- 1) Использование команды DeltaRep означает, что при ее выполнении будет сформирован минимум один новый период.
- 2) Для контроля сумма всех значений Rep во всех КУ должна быть равна значению NumPrd.

Использование параметров Sign и Delta

Значения		Описание	Примечание
Sign	Delta		
X	0	Нет изменения периода	
0	1..31	Уменьшение периода	
1	1..31	Увеличение периода	

НАПРИМЕР.

- 1) Необходимо сформировать ЗИ в режиме тона с кол-вом периодов 9.

NumPrd = 9-1 = 8

DataCC[0] = SetPrd – задает значение первого формируемого периода ЗИ

DataCC[1] = Out( Rep=8, Delta = 0 ) - формируем еще 8 периодов без изменения длительности

- 2) Необходимо сформировать ЗИ в режиме тона с кол-вом периодов 2.

NumPrd = 2-1 = 1

DataCC[0] = SetPrd – задает значение первого формируемого периода ЗИ

DataCC[1] = Out( Rep=1, Delta = 0 ) - формируем еще 1 период без изменения длительности

### 1.5. Использование КУ в режиме ШИМ1 (Mode=1, версия \_v2)

**ПРИМЕЧАНИЕ.** Режим ШИМ1 использовался для отладочных целей и в дальнейшем необходимо использовать режим ШИМ2.

Режим ШИМ1 (Mode=1) используется для плавной регулировки мощности излучения, приближения формы выходного сигнала на антенне к синусоидальной и для формирования импульсов успокоения (уменьшения длины хвоста ЗИ после прекращения возбуждения).

В режиме ШИМ1 может формироваться любой сигнал (тон, ЛЧМ и т.д.). Причем, изменение длительности може происходить хоть каждый полупериод.

При использовании ШИМ1 при формировании полупериода весь полупериод разбивается на 8 временных интервалов T0-T7. Длительность каждого интервала кратна одному периоду тактовой частоты (Ttick) и задается в параметрах КУ. Состояние сигнала (0 или 1) на каждом интервале также задается для каждого интервала в параметрах КУ. Состояния сигнала на интервалах T0-T7 обозначаются соответственно D0-D7.

Минимальное кол-во формируемых полупериодов равно 1.

Первый формируемый полупериод всегда начинается с фазы А.

Менять параметры полупериода (T0-T7, D0-D7) можно на каждом полупериоде или формировать несколько полупериодов с одинаковыми параметрами подряд.

Для формирования импульсов гашения хвоста ЗИ может использоваться дополнительный полупериод гашения, формируемый в той же фазе, что и последний полупериод ЗИ.

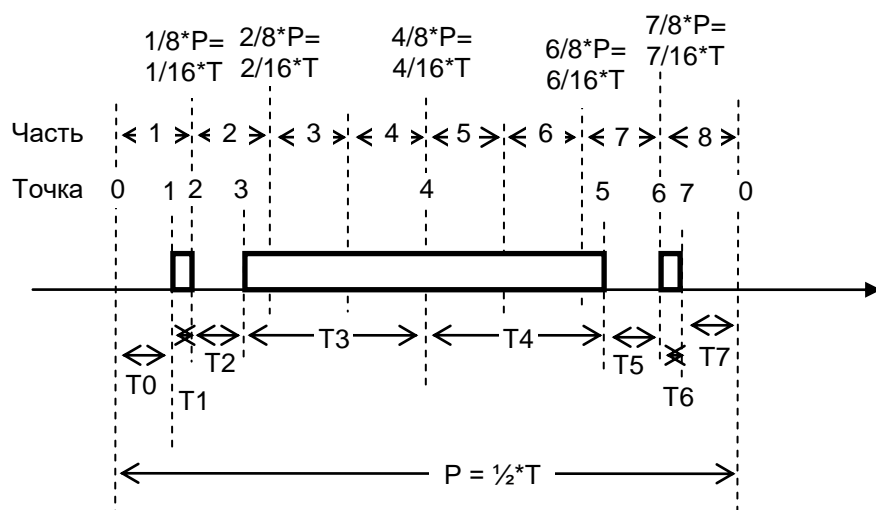


Рисунок 7. Формирование полупериода в режиме ШИМ

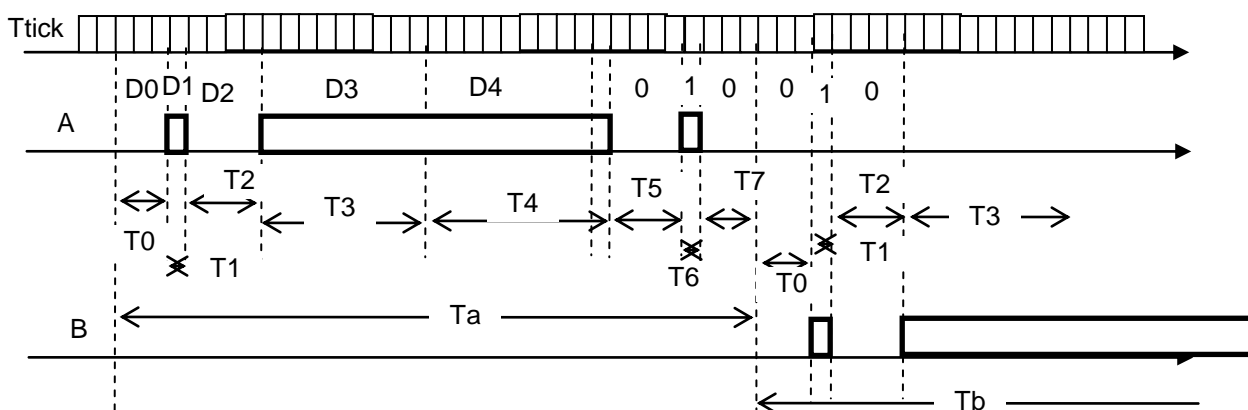


Рисунок 8. Формирование полупериодов в режиме ШИМ

Время T0+T7 определяет мертвое время между фазами.



При использовании четного кол-ва интервалов и получения симметрии:

$T0=T7$

$T1=T6$

$T3=T4$

$D0=D7=0$ ,  $D1=D6=1$ ,  $D2=D5=0$ ,  $D3=D4=1$  (формируются 3 импульса внутри полупериода).

Длительность полупериода =  $T0+T1+T2+T3+T4+T5+T6+T7$

Длительность периода =  $2 * (T0+T1+T2+T3+T4+T5+T6+T7)$

$T3$  может быть не равно  $T4$  (используется для более точной установки несущей частоты).

При изменении длительности на 1 тик период меняется на 2 тика. Т.е. шаг изменения периода = 2 тикам тактовой частоты.

Использование 8 интервалов сделано для упрощения расчетов и возможности получения низкой несущей частоты при ограничении разрядности задаваемых интервалов.

**Всего возможно использование до 102 КУ (CC0..CC101).**

Таблица 3 – Формат массива КУ (DDS\_CC), размер равен  $2*N$  байт ( $N$  – кол-во используемых КУ)

Название	Размер (байт)	Описание	Примечание
CC0_0..CC0_4	5x2=10	KY[0]	Команда PWM0
CC1_0..CC1_4	5x2=10	KY[1]	Команда PWM1
...		KY[2]..KY[100]	Команды PWM2..PWMn-1
CC101_0..CC101_4	5x2=10	KY[101]	Команда PWM101

Для формирования ЗИ используются одна или несколько последовательных КУ – команда PWM.

Каждая КУ имеет размер 5 слов по 16 бит (10 байт), располагается в соседних 5 словах памяти, и позволяет задать параметры формирования одного или сразу нескольких последовательных периодов ЗИ.

Таблица 4 – Формат КУ при Mode=1

Команда	Биты															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM_0 (+0)	D1	T1							D0	T0						
PWM_1 (+2)	D3	T3							D2	T2						
PWM_2 (+4)	D5	T5							D4	T4						
PWM_3 (+6)	D7	T7							D6	T6						
PWM_4 (+8)	Last	Polarity	Rep													

КУ содержит параметры:

**T0..T7** – время интервалов T0..T7

**D0..D7** – значение данных в интервалах D0..D7

**Rep** – кол-во формируемых одинаковых полупериодов

**Last** - признак последней КУ

**Polarity** – признак полярности первого полупериода в пачке

Значение параметров T0..T7 рассчитывается по формуле:

$T[i] = \lceil \text{Time}[i] / (\text{Ttick}) \rceil - 2$ ,

где:

Time[i] – требуемая длительность интервала в нс

Ttick – период тактовой частоты в нс

Минимальное значение  $\text{Time}[i] = 3 * \text{Ttick}$  ( $T[i]_{\min}=2$ )

Максимальное значение  $\text{Time}[i] = 128 * \text{Ttick}$  ( $T[i]_{\max}=2$ )

Значение параметра Rep рассчитывается по формуле:

$\text{Rep} = \text{NumSemiPrd} - 1$ ,

Где

NumSemiPrd – требуемое кол-во полупериодов

Минимальное кол-во формируемых полупериодов = 1 (Значение Rep\_min=0)

Максимальное значение формируемых полупериодов =  $2^{14}$  (Значение Rep\_max= $2^{14}$ -1)

Параметр Last=1 означает, что данная КУ является последней (после выполнения данной КУ формирование ЗИ завершается). Если Last=0, то данная КУ не является последней и происходит обработка и выполнение следующей КУ.

Параметр Polarity=1 означает, что следующий полупериод, который будет формироваться следующей КУ, будет начинаться той же фазы, что и последний полупериод в текущей КУ. Например, если последний полупериод в КУ n формировался в фазе В, то и последующий полупериод (формируемый КУ n+1), будет формироваться в фазе В.

Polarity=1 используется для формирования импульса успокоения в конце ЗИ.

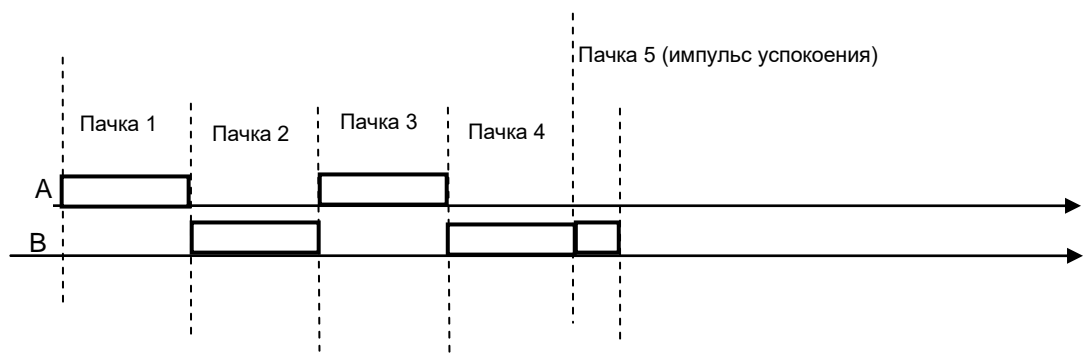
Если Polarity=0, то следующий полупериод, который будет формироваться следующей КУ, будет начинаться с противоположной фазы по отношению к последнему полупериоду в текущей КУ.

ПРИМЕЧАНИЕ. Если для текущей КУ установлен признак Last=1, то значение параметра Polarity может быть любым (следующая КУ выполняться не будет).

Одна КУ позволяет сформировать один или несколько последовательных одинаковых полупериодов (полупериоды с одинаковыми параметрами ШИМ). Несколько последовательных одинаковых полупериодов называются пачкой. Внутри пачки может быть четное или нечетное кол-во полупериодов (задается параметром Rep). Полупериоды внутри пачки нумеруются, начиная с 1.

Первый полупериод (в начале ЗИ) всегда начинается с фазы А. Каждый последующий полупериод автоматически инвертирует выходную фазу, и так до конца пачки. Таким образом, при задании четного кол-ва полупериодов будет сформировано одинаковое кол-во полупериодов фазы А и В.

Например, при задании Rep=3 будет сформировано 4 одинаковых полупериода (2 периода), полупериод 1 – фаза А, полупериод 2 – фаза В, полупериод 3 – фаза А, полупериод 4 – фаза В.



Пример формирования двух одинаковых периодов (4 полупериодов) и импульса успокоения.

НАПРИМЕР.

Необходимо сформировать ЗИ в режиме тона с кол-вом периодов 10 и импульсом успокоения в конце.

Используются две КУ. Первая КУ формирует 20 одинаковых полупериодов, вторая КУ – импульс успокоения.

Параметры КУ1: T0-T7, D0-D7, Rep=19, Last=0, Polarity=1

Параметры КУ2: T0-T7, D0-D7, Rep=0, Last=1, Polarity=x

### 1.6. Использование КУ в режиме ШИМ2 (Mode=2, версия \_v3)

Режим ШИМ2 (Mode=2) разработан с учетом недостатков режима ШИМ1.

Основные отличия ШИМ2 от ШИМ1:

- состояние в каждом временном интервале задается не отдельно, а с помощью единого кода режима формирования импульсов ModeD;
- интервалы сделаны симметричными, это позволило сократить кол-во КУ до 8 байт;
- введен режим формирования задержки между пачками для возможности работы с ФКМ.
- увеличена длина кол-ва повторов, что позволяет при формировании ЛЧМ в большинстве случаев обойтись одной КУ для формирования одного шага.

В режиме ШИМ2 может формироваться любой сигнал (тон, ЛЧМ и т.д.). Причем, изменение длительности может происходить хоть каждый полупериод.

При использовании ШИМ2 при формировании полупериода весь полупериод разбивается на 7 временных интервалов T0-T6. Длительность каждого интервала кратна одному периоду тактовой частоты (Ttick) и задается в параметрах КУ. Состояние сигнала (0 или 1) на каждом интервале также задается в параметрах КУ. Состояния сигнала на интервалах T0-T6 обозначаются соответственно D0-D6.

Минимальное кол-во формируемых с помощью одной КУ полупериодов равно 1.

Первый формируемый полупериод всегда начинается с фазы А.

Менять параметры полупериода (T0-T6, D0-D6) можно хоть на каждом полупериоде или формировать несколько полупериодов с одинаковыми параметрами подряд.

Для формирования импульсов гашения хвоста 3И может использоваться дополнительные полупериоды гашения.

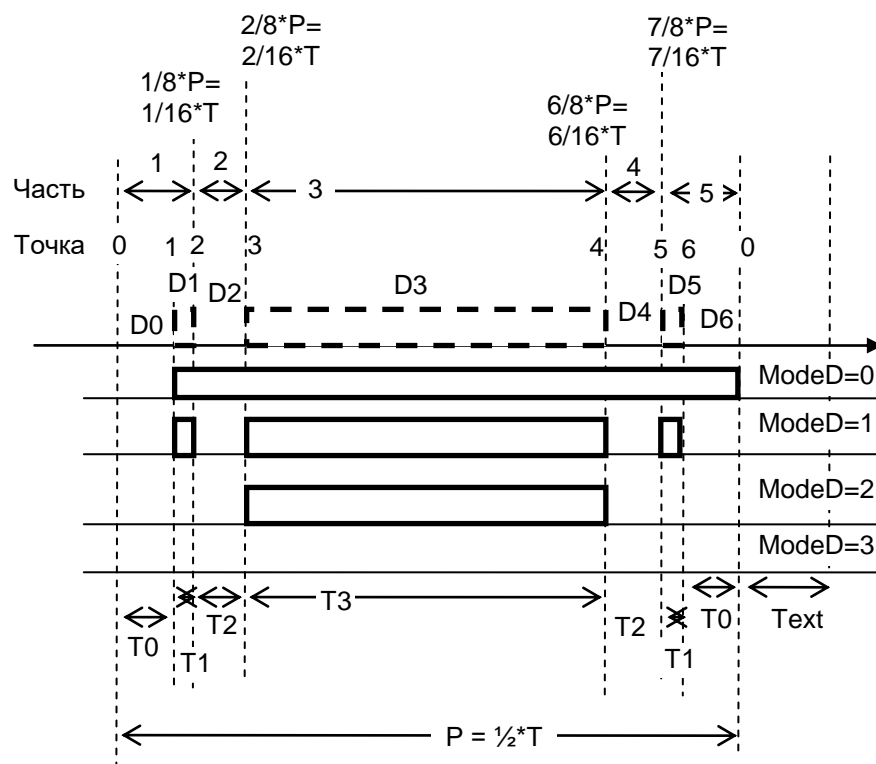


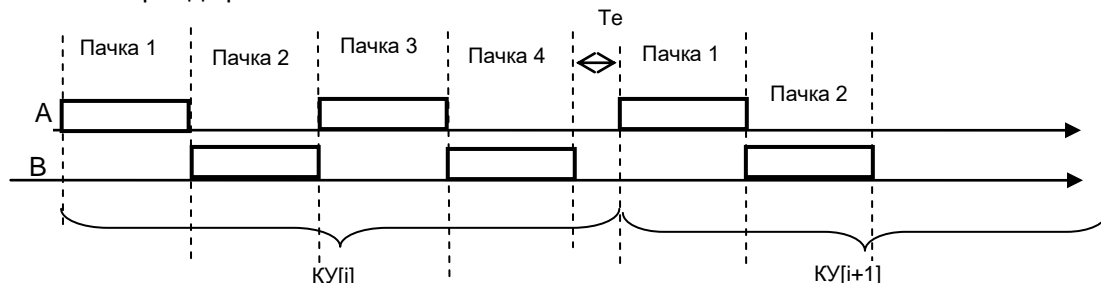
Рисунок 9. Формирование полупериода в режиме ШИМ2

Режимы формирования D0-D6 ограничены следующими вариантами:

- 1) D0=0, D1=1, D2=1, D3=1, D4=1, D5=1, D6=1 – режим ModeD=0. Время T0 определяет мертвое время между фазами. Используется для поддержки старого режима (без ШИМ).
- 2) D0=0, D1=1, D2=0, D3=1, D4=0, D5=1, D6=0 – режим ModeD=1. Время  $2 * T0$  определяет мертвое время между фазами. Используется для реализации ШИМ и регулировки мощности.
- 3) D0=0, D1=0, D2=0, D3=1, D4=0, D5=0, D6=0 – режим ModeD=2. Время  $2 * (T0 + T1 + T2)$  определяет мертвое время между фазами. Используется для формирования гасящих импульсов.

4) D0=0, D1=0, D2=0, D3=0, D4=0, D5=0, D6=0 – режим ModeD=3. Применимость пока непонятна, возможно, потребуется для формирования сверхдлинных задержек между пачками.

Длительность полупериода =  $2 \cdot T_0 + 2 \cdot T_1 + 2 \cdot T_2 + T_3$ . Длительность периода =  $2 \cdot (2 \cdot T_0 + 2 \cdot T_1 + 2 \cdot T_2 + T_3)$ . При изменении длительности интервала  $T_0$ ,  $T_1$  или  $T_2$  на 1 тик период меняется на 4 тика. При изменении длительности интервала  $T_3$  тик период меняется на 2 тика. Т.е. шаг изменения периода равен 2 или 4 тикам тактовой частоты.



Одна или несколько пачек формируется с помощью одной КУ. Одна КУ занимает 4 слова (Таблица 5).

Таблица 5 – Формат КУ при Mode=2

Команда	Биты															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCx_0 (+0)	T1								T0							
CCx_1 (+2)	T3								T2							
CCx_2 (+4)	Last	Polarity	EnaTe	ModeD					Rep_L							
CCx_3 (+6)	Te/Rep_H															

КУ содержит параметры:

**T0..T3** – время интервалов T0..T3

**ModeD** – режим формирования данных в интервалах D0..D6

**Rep** – кол-во формируемых одинаковых полупериодов

**Last** – признак последней КУ

**Polarity** – признак полярности первой пачки в следующем го полупериода в пачке

**Te** – время интервала  $T_e$ , формируемого после окончания пачки. Если  $EnaTe=0$ , интервал  $t_e$  не формируется. Если  $EnaTe=1$ , интервал  $T_e$  формируется. Значение параметра  $T_e$  рассчитывается по формуле:

$$T_e = \lceil \text{TimeE} / T_{\text{tick}} \rceil - 1$$

где:

$\text{TimeE}$  – требуемая длительность интервала  $T_e$  в нс

$T_{\text{tick}}$  – период тактовой частоты в нс

Минимальное значение  $\text{TimeE} = 1 \cdot T_{\text{tick}}$  (параметр  $T_e=0$ ). Максимальное значение  $\text{TimeE} = 65536 \cdot T_{\text{tick}}$  (параметр  $T_e=65535$ ). Шаг изменения интервала  $T_e = 1 \cdot T_{\text{tick}}$ .

Значение параметров T0..T3 рассчитывается по формуле:

$$T[i] = \lceil \text{Time}[i] / T_{\text{tick}} \rceil - 2,$$

где:

$\text{Time}[i]$  – требуемая длительность интервала в нс

$T_{\text{tick}}$  – период тактовой частоты в нс

Минимальное значение  $\text{Time}[i] = 2 \cdot T_{\text{tick}}$  ( $T[i]=0$ ). Максимальное значение  $\text{Time}[i] = 257 \cdot T_{\text{tick}}$  ( $T[i]=255$ ). Шаг изменения интервала  $T[i] = 1 \cdot T_{\text{tick}}$ .

Минимальное значение длительности полупериода = 7 интервалов \*  $2 \cdot T_{\text{tick}} = 14 \cdot T_{\text{tick}}$ . Минимальное значение длительности периода =  $28 \cdot T_{\text{tick}}$ .

Максимальное значение длительности полупериода = 7 интервалов \*  $257 \cdot T_{\text{tick}} = 1799 \cdot T_{\text{tick}}$ . Максимальное значение длительности периода =  $2 \cdot 1799 \cdot T_{\text{tick}} = 3598 \cdot T_{\text{tick}}$ .

Для опорной частоты 100 МГц (Ttick=10нс):

Среднее значение мертвого времени = 100 нс (для Q=100%).

Режим ModeD=0: T0min=100нс = 10\*Ttick

Мин. значение периода ЗИ =  $2 * (2 * (T0min + T1min + T2min) + T3min) = 2 * (2 * (100 + 20 + 20) + 20) = 600 \text{ нс}$  (макс. частота ~1,66 МГц)

Макс. значение периода ЗИ =  $2 * (2 * (T0min + T1max + T2max) + T3max) = 2 * (2 * (100 + 2570 + 2570) + 2570) = 26100 \text{ нс}$  (мин. частота ~38,3 кГц)

Режим ModeD=1: T0min=50нс = 5\*Ttick

T1=0,125\*P-T0min

T2=0,25\*P-T1-T0min = 0,25\*P-0,125\*P-T0min-T0min=0,125\*P-2\*T0min

T3=(6/8)\*P

Значение параметра Rep рассчитывается по формуле:

**Rep = NumSemiPrd-1,**

Где

NumSemiPrd – требуемое кол-во полупериодов

Минимальное кол-во формируемых полупериодов = 1 (Значение Rep\_min=0)

Разрядность параметра Rep зависит от режима формирования интервала Te: если EnaTe=0, то разрядность Rep = 24 бита (биты 23..8 хранятся в Rep\_H, биты 7..0 – в Rep\_L). Если EnaTe=1, то разрядность Rep ограничивается 8 битами (биты 7..0 – в Rep\_L).

Максимальное значение формируемых полупериодов (при EnaTe=0) = (255+65535)+1  
Максимальное значение формируемых полупериодов (при EnaTe=1) = 256 (Значение Rep\_max=256)

Параметр Last=1 означает, что данная КУ является последней (после выполнения данной КУ формирование ЗИ завершается). Если Last=0, то данная КУ не является последней и происходит обработка и выполнение следующей КУ.

Параметр Polarity=1 означает, что первый полупериод, который будет формироваться следующей КУ, будет начинаться с фазы В; если параметр Polarity=0 - первый полупериод, который будет формироваться следующей КУ, будет начинаться с фазы А (**Таблица 6**).

Таблица 6 – Использование признака Polarity

Фаза последнего полупериода текущей КУ	Значение Polarity для текущей КУ	Фаза первого полупериода при выполнении следующей КУ	Примечание
А	0	А	
А	1	В	
В	0	А	
В	1	В	

Например, если последний полупериод в КУ n формировался в фазе А, то и последующий полупериод (формируемый КУ n+1), будет формироваться в фазе В.

Polarity=1 используется для формирования импульса успокоения в конце ЗИ.

Если Polarity=0, то следующий полупериод, который будет формироваться следующей КУ, будет начинаться с противоположной фазы по отношению к последнему полупериоду в текущей КУ.

ПРИМЕЧАНИЕ. Если для текущей КУ установлен признак Last=1, то значение параметра Polarity может быть любым (следующая КУ выполняться не будет).

Таблица 7 – Формат массива КУ (DDS\_CC), размер равен 2\*4\*N байт (N – кол-во используемых КУ)

Название	Размер (байт)	Описание	Примечание
CC0_0..CC0_3	4x2=8	КУ[0]	Команда CC0

CC1_0..CC1_4	4x2=8	KY[1]	Команда CC1
...		KY[2]...KY[236]	Команды CC2..CC236
CC237_0..CC237_4	4x2=8	KY[237]	Команда CC237

**Максимальное кол-во КУ = 952 слова/4 = 238 (при 2048 байт на один ЗИ)**

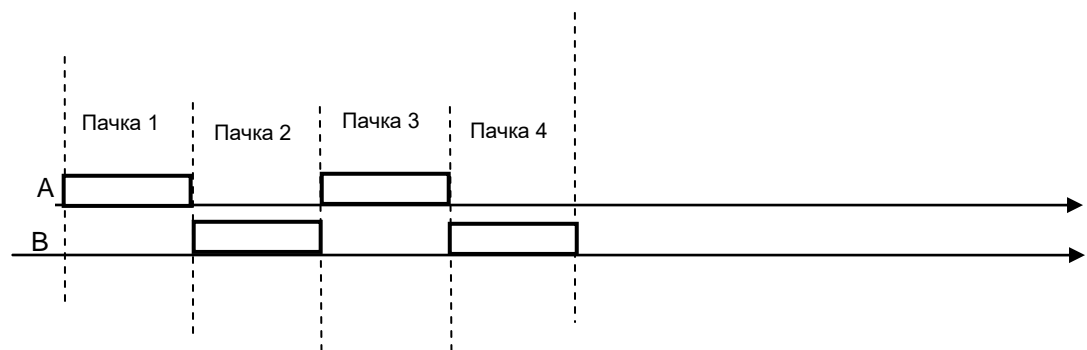
Для формирования ЗИ используются одна или несколько последовательных КУ – команда PWM.

Каждая КУ имеет размер 4 слова по 16 бит (8 байт), располагается в соседних 4 словах памяти, и позволяет задать параметры формирования одного или сразу нескольких последовательных периодов ЗИ.

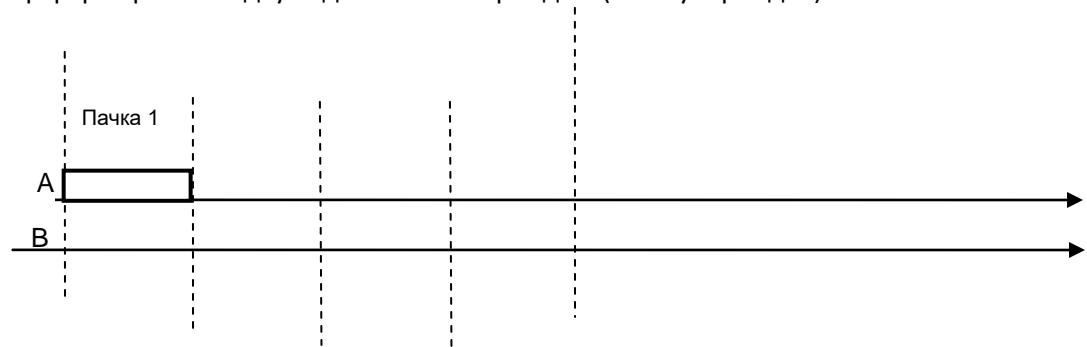
Одна КУ позволяет сформировать один или несколько последовательных одинаковых полупериодов (полупериоды с одинаковыми параметрами ШИМ). Несколько последовательных одинаковых полупериодов называются пачкой. Внутри пачки может быть четное или нечетное кол-во полупериодов (задается параметром Rep). Полупериоды внутри пачки нумеруются, начиная с 1.

Первый полупериод (в начале ЗИ) всегда начинается с фазы А. Каждый последующий полупериод автоматически инвертирует выходную фазу, и так до конца пачки. Таким образом, при задании четного кол-ва полупериодов будет сформировано одинаковое кол-во полупериодов фазы А и В.

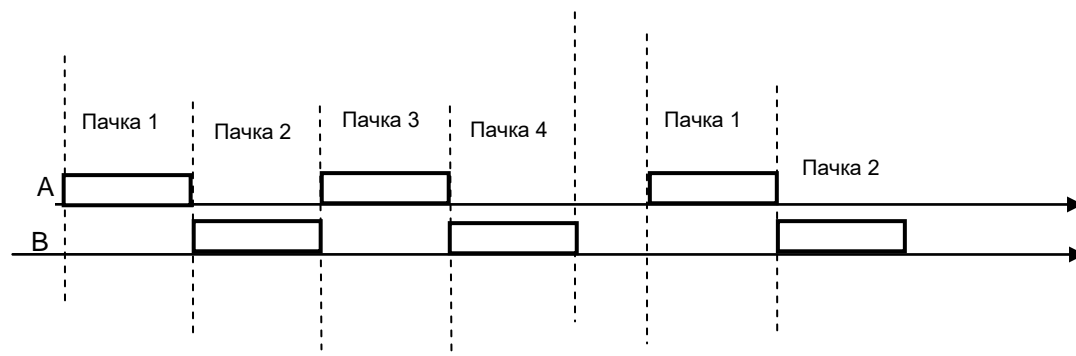
Например, при задании Rep=3 будет сформировано 4 одинаковых полупериода (2 периода), полупериод 1 – фаза А, полупериод 2 – фаза В, полупериод 3 – фаза А, полупериод 4 – фаза В.



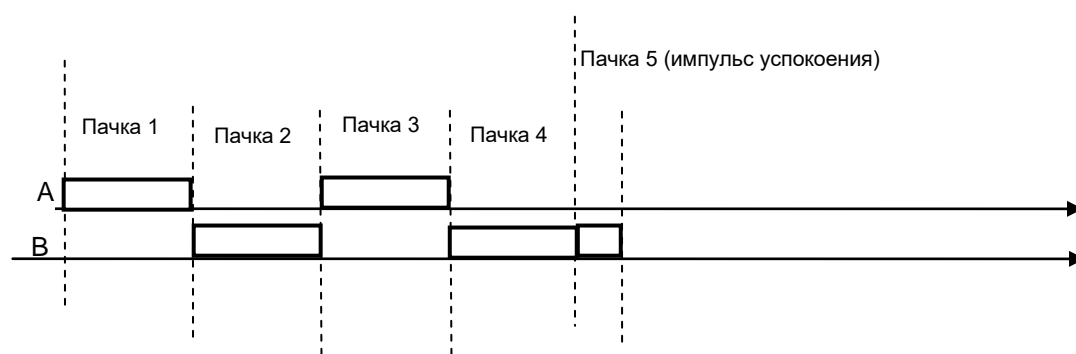
Пример формирования двух одинаковых периодов (4 полупериодов)



Пример формирования одного полупериода



Пример формирования двух одинаковых периодов (4 полупериодов)



Пример формирования двух одинаковых периодов (4 полупериодов) и импульса успокоения.

### 1.6.1. Регулировка шимирования в режиме ШИМ2

Для наиболее полного диапазона регулировки ШИМ необходимо использовать Mode=2 .

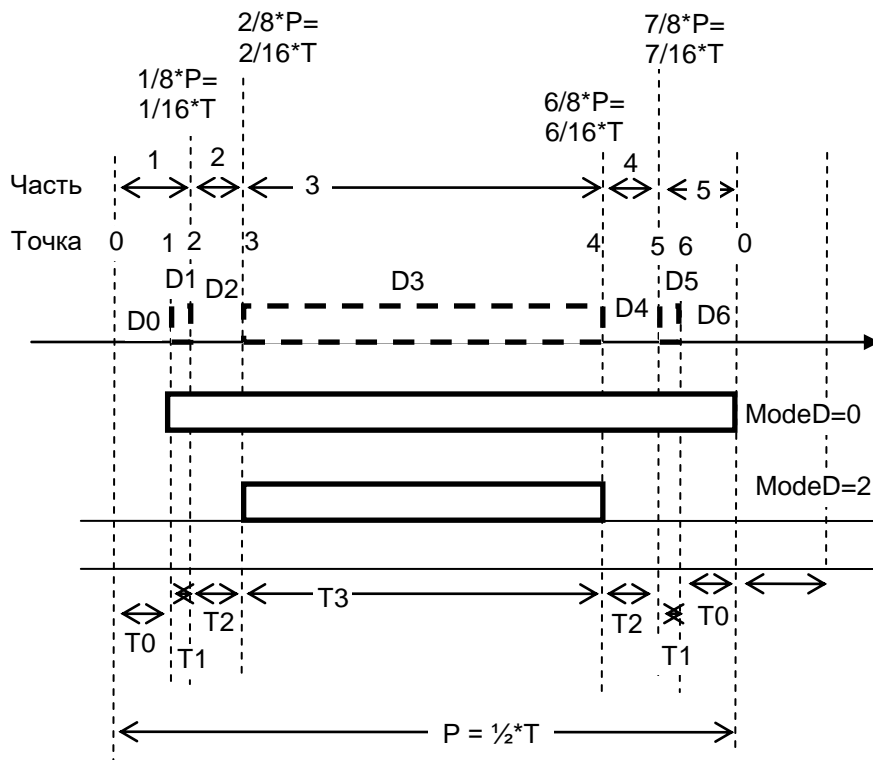


Рисунок 10. Шимирование

Исходные данные для шимирования:

- Длительность полупериода
- Минимальное мертвое время

Длительность полупериода является суммой интервалов:  $P = 2 * (T_0 + T_1 + T_2) + T_3$ .

Минимальное мертвое время  $T_{z\_min} = 2 * (T_0 + T_1 + T_2)$ .

При расчете параметров ЗИ рассчитанные параметры определяют  $T_{z\_min}$  и соответственно – максимальную скважность (мощность) - начальные значения  $T_{0\_beg}$ ,  $T_{1\_beg}$ ,  $T_{2\_beg}$  определены. При регулировке скважности разрешается только увеличивать сумму  $T_0 + T_1 + T_2$ .

Алгоритм регулировки суммы:

На входе:

$T_0(i)$ ,  $T_1(i)$ ,  $T_2(i)$ ,  $T_3(i)$  - считываются из FLASH

$K$  – (1..127) – задается через регистры PGL

- 1)  $T_{z\_sum\_min}(i) = T_0(i) + T_1(i) + T_2(i)$  // мин.  $T_z$ , которое нельзя уменьшать
- 2)  $Tact(i) = [ (K * (2 * (T_0(i) + T_1(i) + T_2(i)) + T_3(i))) / 128 ]$  // желаемая длительность сигнала
- 3)  $P(i) = 2 * (T_0(i) + T_1(i) + T_2(i)) + T_3(i)$  // требуемая длит полупериода
- 4) IF ( $Tact(i) < 2$ ) THEN  $Tact\_result(i) = 2$
- 5)  $Tzero\_sum\_result(i) = [ (P(i) - Tact\_result(i)) / 2 ]$  // длит. Нуля
- 6) IF ( $Tzero\_sum\_result(i) > 768$ ) THEN  $Tzero\_sum\_result(i) = 768$



Tz_min=96 нс (12t)				
Макс. скважность	T0=T1=T2=2t	T3=2t..256t	P=12+(2..256)=14..268	Fc=1/(2*P*0,008мкс)=4,46..0,233 МГц
Fc=1МГц	P=62	T0=12,T1=T2=2,T3=30	Qmax=0,8	ModeD=0
		T0=12,T1=T2=9,T3=2	Qmin=0	

1.6.2. Формирование ФЗМ в режиме ШИМ2 (\_v3)

ЗИ с ФЗМ состоит из нескольких битовых интервалов – квантов (кванты нумеруются от 1 до К).

Каждый квант содержит несколько периодов несущей (М) и задержку (Delay).  
Кол-во периодов в каждом кванте, как правило, одинаково ( $M_1=M_2=...=M_k=M$ ).  
Задержка может быть различна для каждого кванта (длительность квантов различна) и кратна  $T_{clk}$ .

Для последнего кванта задержка не используется (=0).

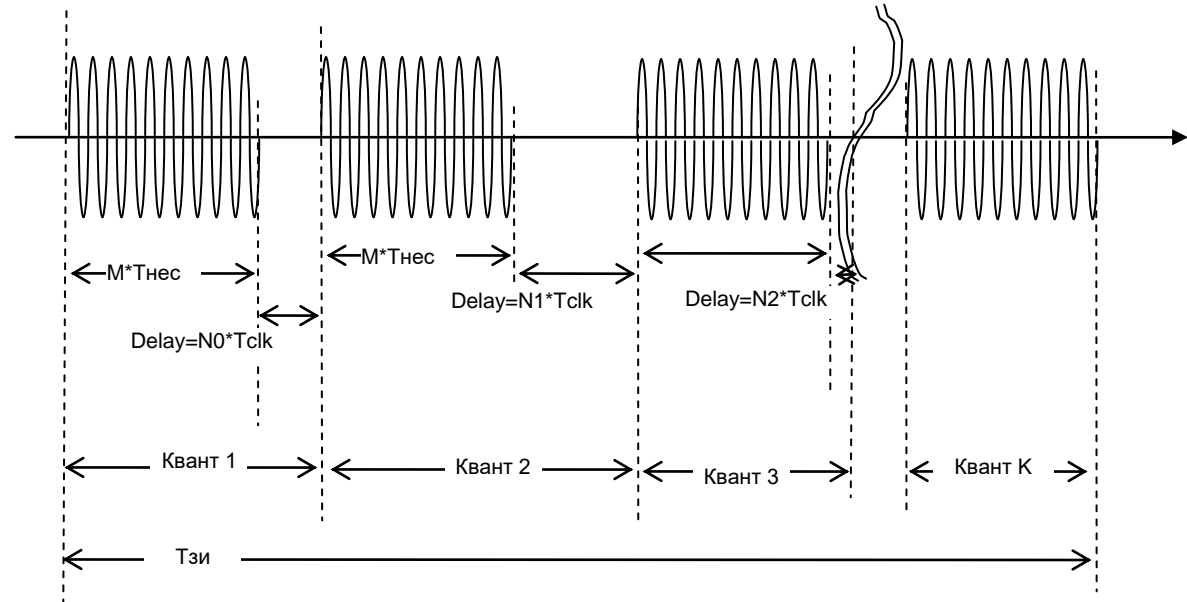


Рисунок 11. Пример формирования ФЗМ

1.6.3. Формирование кода Баркера в режиме ШИМ2 (\_v3)

ЗИ с кодом Баркера состоит из нескольких интервалов одинаковой длительности. Будем называть эти интервалы битами. Каждый бит содержит М периодов (М – четное) несущей частоты  $F_c$  и имеет один из двух вариантов начальной фазы. Положительная фаза – бит начинается с положительного полупериода А и заканчивается отрицательным полупериодом В. Отрицательная фаза – бит начинается с отрицательного полупериода В и заканчивается положительным полупериодом А. Бит с положительной начальной фазой будем называть положительным битом или битом +, бит с отрицательной начальной фазой будем называть отрицательным битом или битом -.

Кол-во битов называется длиной кода – N ( $N=2,3,4,5,7,11,13$ ).

Биты нумеруются от 0 до N-1.

N	k												
	0	1	2	3	4	5	6	7	8	9	10	11	12
2	+1	-1											
3	+1	+1	-1										
4	+1	+1	-1	+1									
5	+1	+1	+1	-1	+1								
7	+1	+1	+1	-1	-1	+1	-1						
11	+1	+1	+1	-1	-1	-1	+1	-1	-1	+1	-1		
13	+1	+1	+1	+1	+1	-1	-1	+1	+1	-1	+1	-1	+1

При смене полярности бита появляется скачок фазы (Рисунок 12).

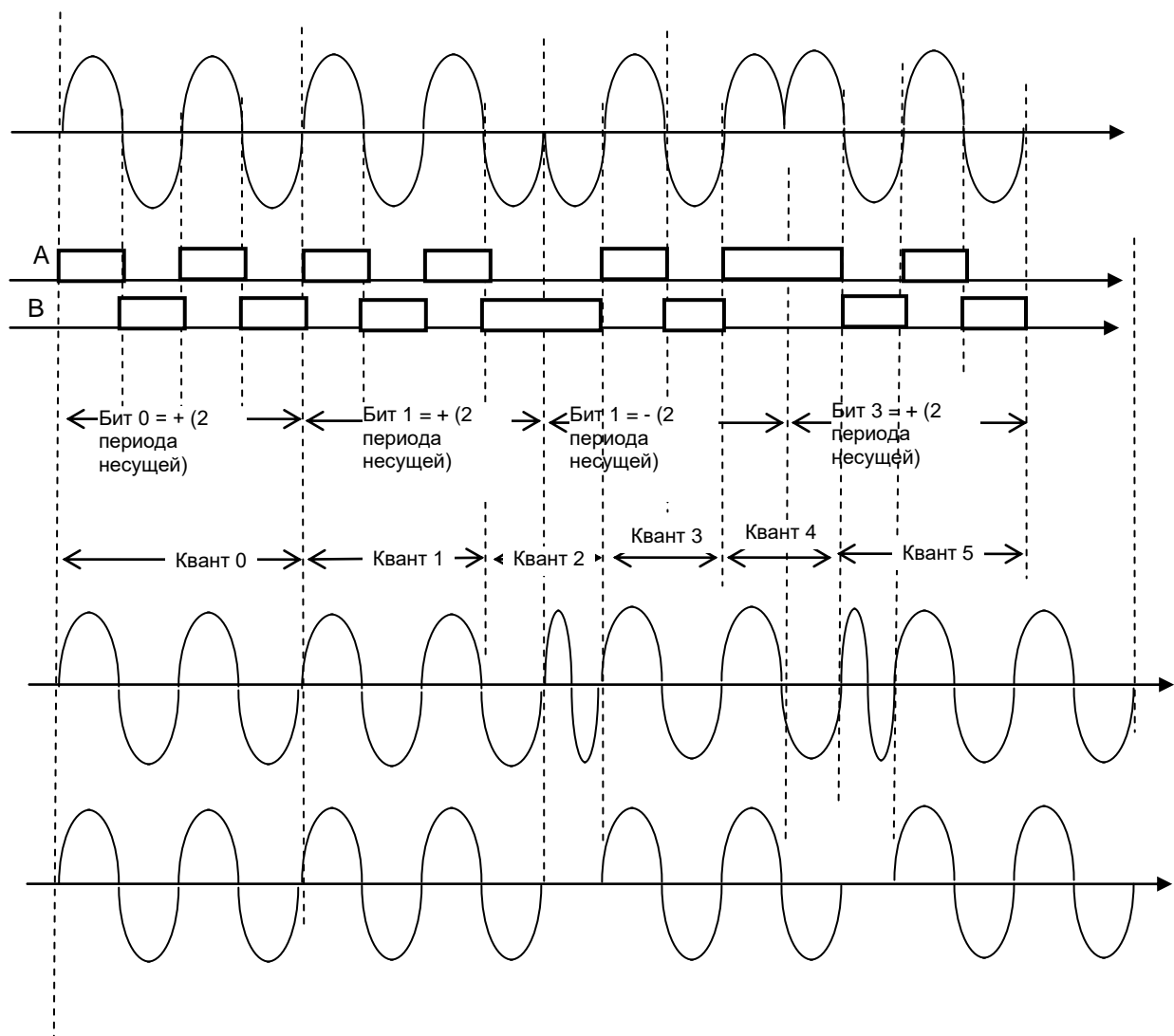


Рисунок 12. Пример формирования ФМ, Баркер N=4 (+, +, -, +), 2 периода несущей (M=4)

### 1.6.3.1. Формирование кода Баркера “классическим” методом (вариант 1)

В этом случае скачок фазы формируется как один полупериод с частотой  $= F_c/2$ .

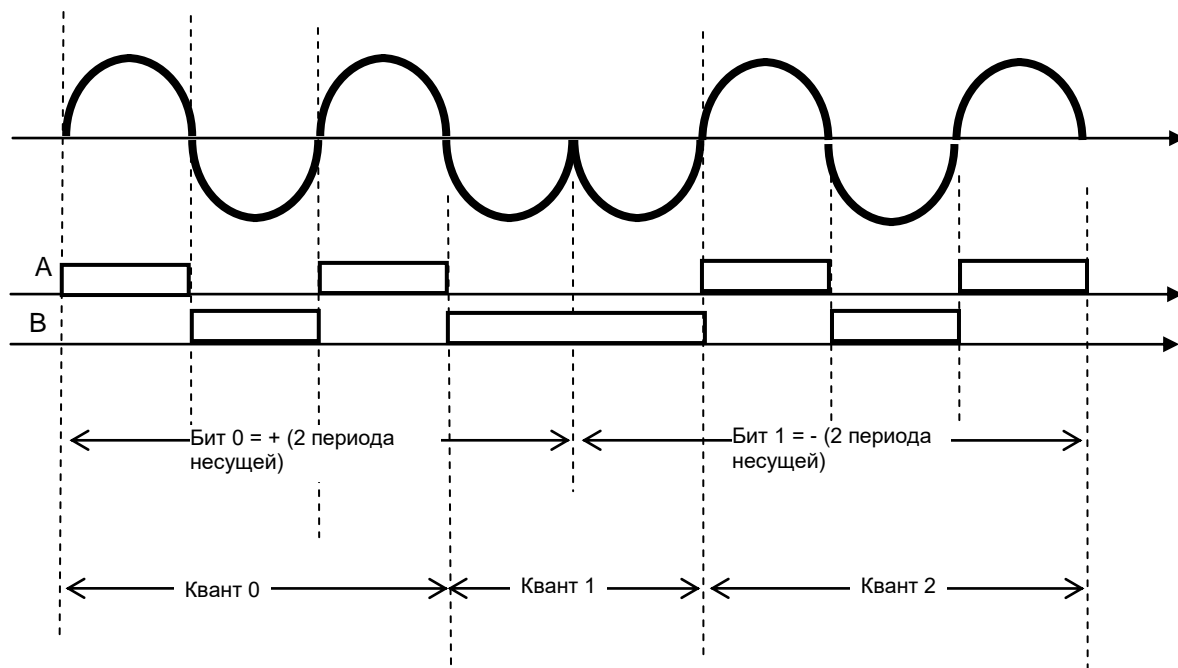
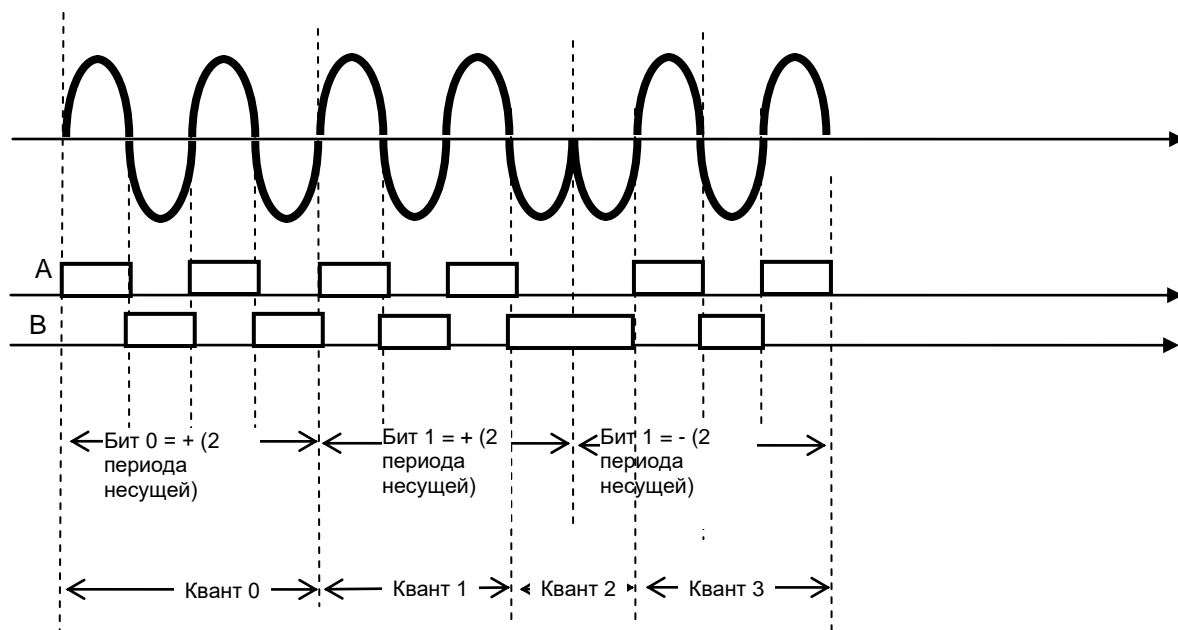
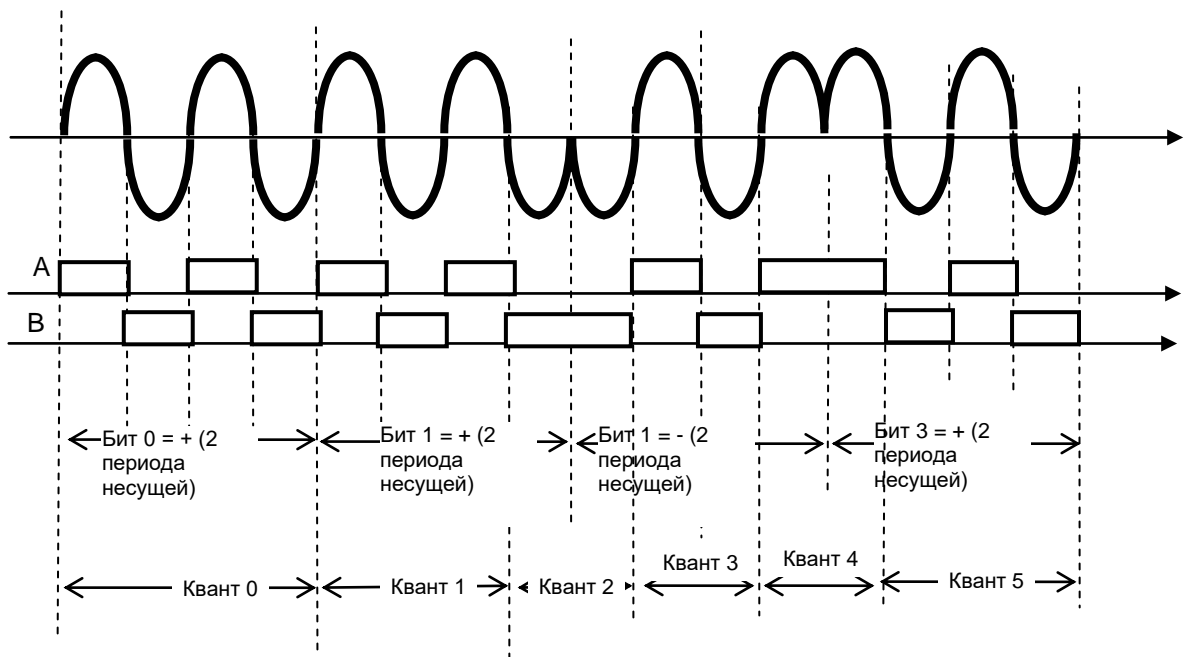


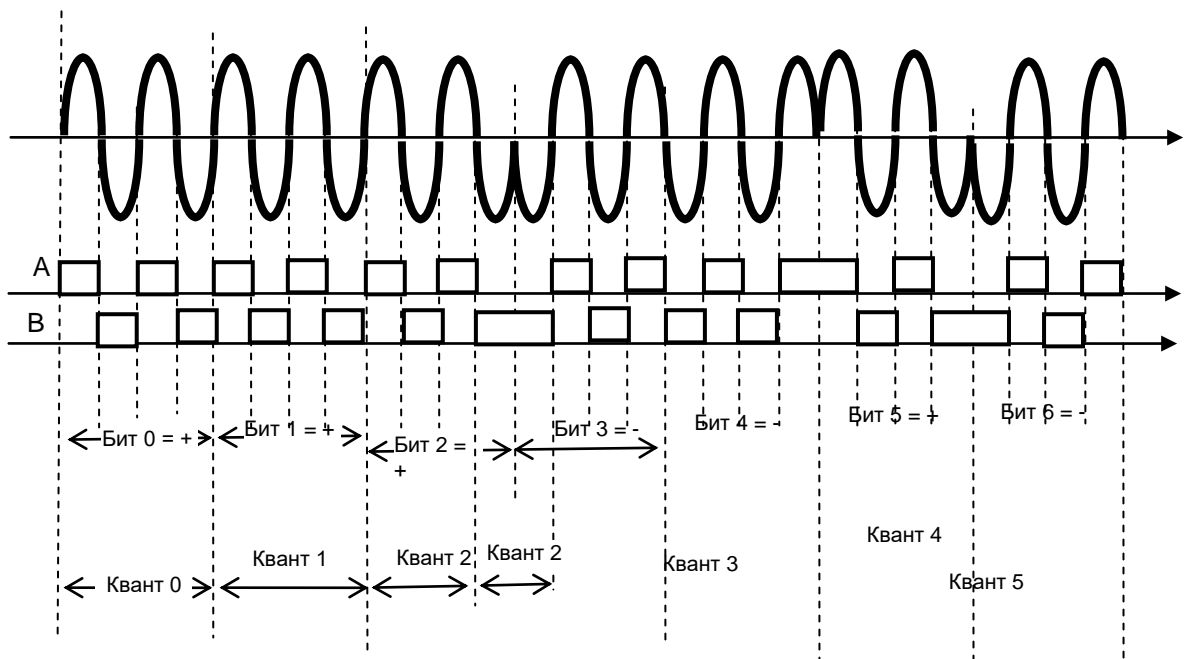
Рисунок 13. Пример формирования ФКМ, Баркер N=2 (+,-)



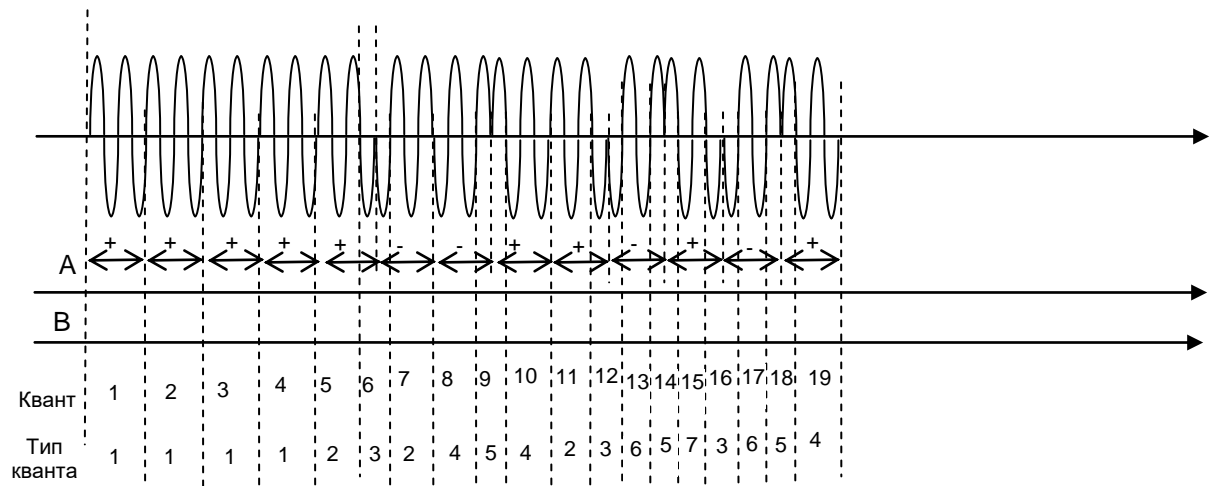
Пример формирования ФКМ, Баркер N=3 (+, +, -)



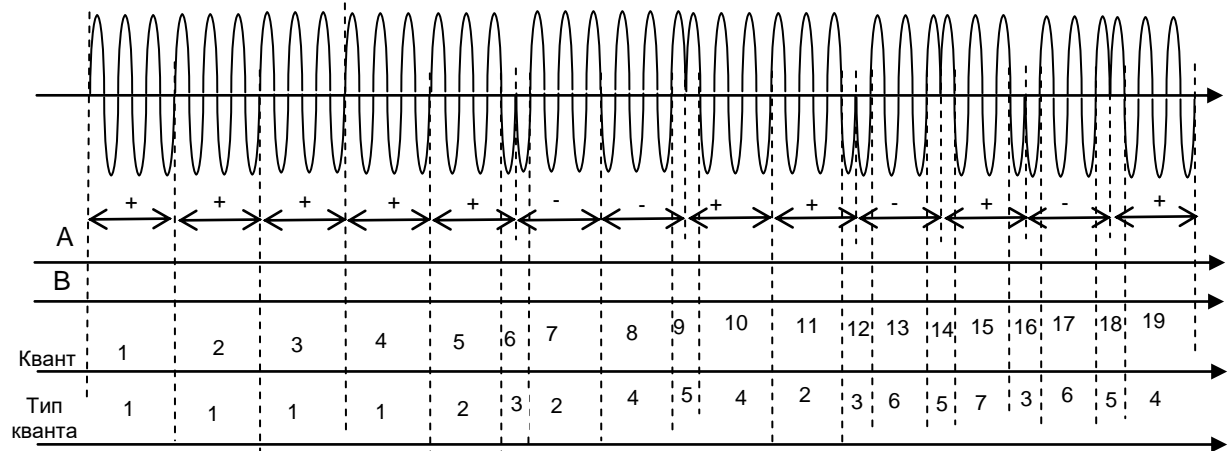
Пример формирования ФКМ, Баркер N=4 (+, +, -, +)



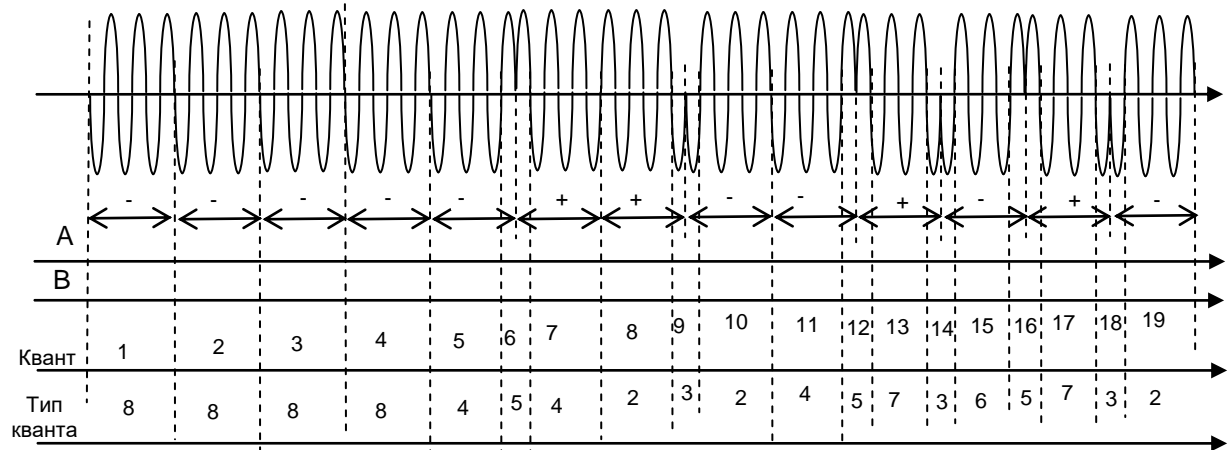
Пример формирования ФКМ, Баркер N=7 (+, +, +, -, -, +, -)



Пример формирования ФКМ, Баркер N=13 (+,+,+,+,-,-,-,-,-,-,-,-,-), M=4



Пример формирования ФКМ, Баркер N=13 (+,+,+,+,-,-,-,-,-,-,-,-,-), M=6



Пример формирования ФКМ, Баркер N=13 инверсный (-,-,-,-,-,-,-,-,-,-,-,-,-), M=6

Получаем следующие базовые типы квантов (8 типов):

- 1 - квант начинается с фазы А, длина М полупериодов (М – четное), частота  $F_c$
- 2 - квант, начинается с фазы А, длина М-1 полупериодов, частота  $F_c$
- 3 - квант, начинается с фазы В, длина 1 полупериод, частота  $F_c/2$
- 4 - квант, начинается с фазы В, длина М-1 полупериодов, частота  $F_c$
- 5 - квант, начинается с фазы А, длина 1 полупериод, частота  $F_c/2$
- 6 - квант, начинается с фазы А, длина М-2 полупериодов, частота  $F_c$
- 7 - квант, начинается с фазы В, длина М-2 полупериодов, частота  $F_c$
- 8 – квант начинается с фазы В, длина М полупериодов, частота  $F_c$

### 1.6.3.2. Формирование кода Баркера с переходом через двойную частоту (вариант 2)

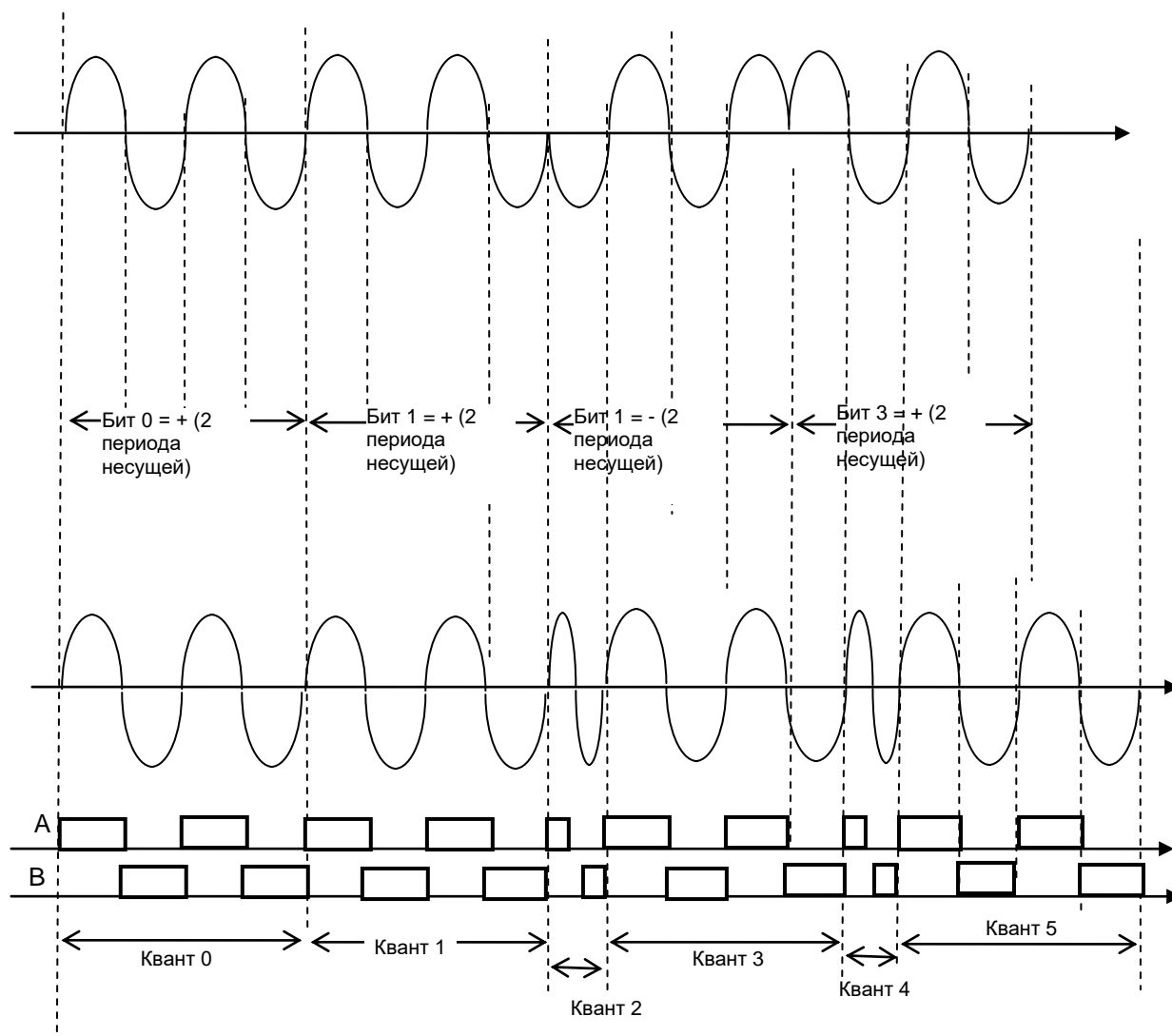
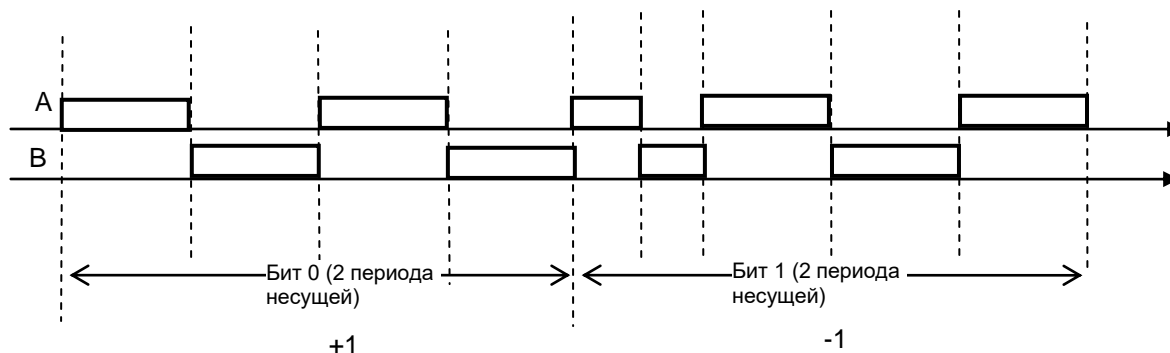


Рисунок 14. Пример формирования ФКМ, Баркер N=4 (+, +, -, +), 2 периода несущей (M=4)

Получаем следующие базовые типы квантов (2 типа):

- 1 - квант начинается с фазы A, длина M полупериодов, частота  $F_c$
- 2 - квант, начинается с фазы A, длина 2 полупериода, частота  $F_c * 2$



Пример формирования ФКМ, Баркер N=2, с переходом через удвоенную частоту

### 1.6.3.3. Формирование кода Баркера с переходом через паузу (вариант 3)

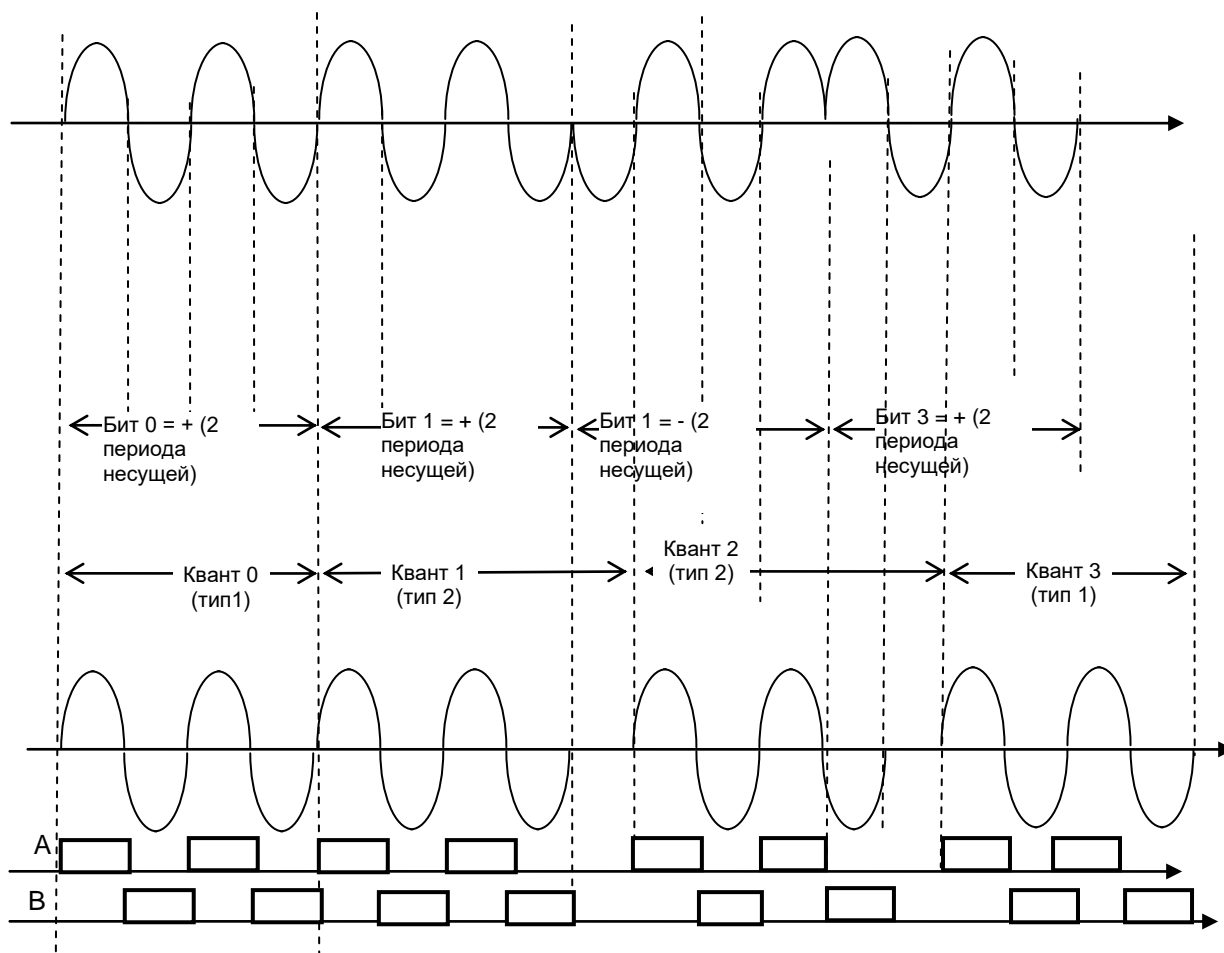


Рисунок 15. Пример формирования ФКМ, Баркер N=4 (+, +, -, +), 2 периода несущей (M=4)

Получаем следующие базовые типы квантов (2 типа):

Тип 1 - квант начинается с фазы A, длина M полупериодов (M-четное), частота  $F_c$

Тип 2 - квант начинается с фазы A, длина M полупериодов (M-четное), частота  $F_c$ , задержка = 0,5 периода?



#### 1.6.3.4. Формирование кода Баркера (вариант 4)

В этом случае скачок фазы формируется как один полупериод с частотой  $= F_c/2$  по аналогии с вариантом 1 за исключением того, что вместо единого полупериода В формируется два полупериода В с “дыркой” между ними, равной мертвому времени.

Этот вариант упрощает расчет КУ, т.к. вместо вставки специальной КУ при скачке фазы в варианте 1, дополнительной КУ не используется (меняется только признак Polarity).

Кол-во используемых КУ равно кол-ву используемых бит (один квант кодируется одной КУ). Остальные параметры КУ (кроме бита Polarity) одинаковы для всех КУ. Значение признака Polarity определяет знак бита: если последующий бит = “+”, то для текущей КУ значение Polarity=0 (начинаем следующую КУ с фазы А), если последующий бит “-”: для текущей КУ значение Polarity=1 (начинаем следующую КУ с фазы В).

**ПРИМЕЧАНИЕ.** Этот способ также подходит для регулировки мощности внутри полупериода с использованием ШИМ.

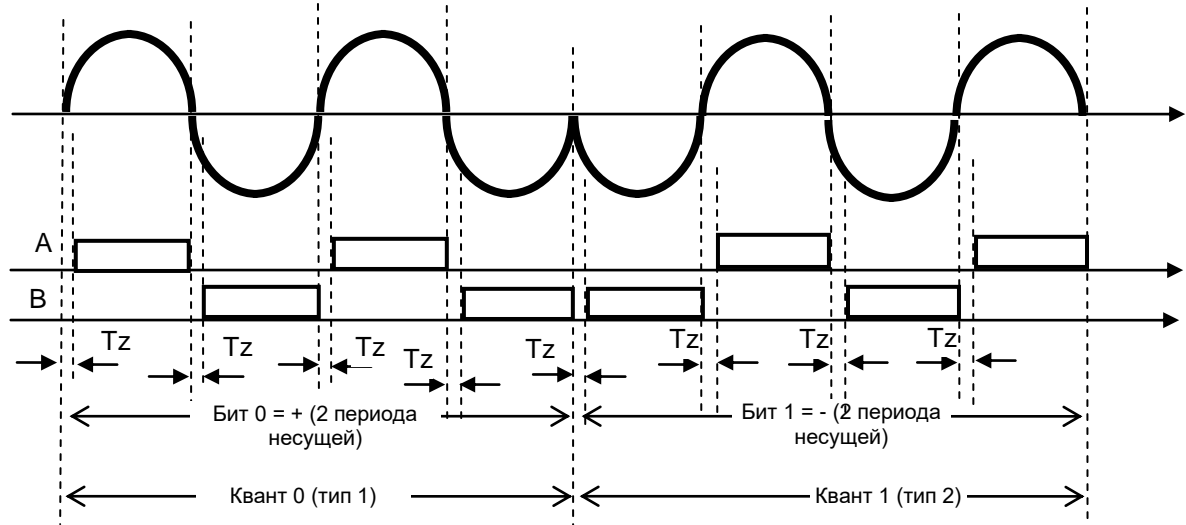


Рисунок 16. Пример формирования ФКМ, Баркер N=2 (+,-), вариант 4

Получаем следующие базовые типы квантов (2 типа):

Тип 1 - квант начинается с фазы А, длина М полупериодов (М-четное), частота  $F_c$

Тип 2 - квант начинается с фазы В, длина М полупериодов (М-четное), частота  $F_c$

### 1.7. Метод прямого цифрового синтеза (генератор GCHIRP, версия \_v4)

Метод прямого цифрового синтеза частот (ПЦС или DDS – direct digital synthesis).

Устройство представляет собой специализированный процессор, позволяющий с высокой скоростью вычислять значения фазы  $\varphi(t)$  сигнала и ставить им в соответствие значения амплитуды функции  $\sin \varphi(t)$ .

В общем случае устройства реализующие метод ПЦС обладает следующими основными достоинствами:

- высокое частотное разрешение ( $10^{-1}$ – $10^{-6}$ ) при формируемой частоте в несколько сотен мегагерц;
- высокая скорость перехода с частоты на частоту без разрыва фазы,
- быстрое изменение начальной фазы колебания,
- стабильность частоты и фазы определяется синхросигналом,
- малые габариты, масса и энергопотребление.

Недостатками метода являются: ограничение верхнего значения синтезируемой частоты половиной значения частоты синхросигнала, существование отдельных побочных составляющих, обусловленных дискретным характером цифрового синтеза и округлением при вычислении фазы и амплитуды, уровень которых может быть выше, чем у синтезаторов других типов.

Далее рассмотрим случай расширения спектра радиосигнала за счет применения внутриимпульсной линейной частотной модуляции (ЛЧМ).

Напомним, что ЛЧМ-радиосигнал можно представить в виде:

$$u(t) = U_0(t) \sin(2\pi f_c t + \pi v t^2), \quad (1)$$

где  $f(t) = \frac{df}{dt} = f_c + vt$  – частота колебания, а  $v = \frac{df}{dt}$  – скорость изменения частоты.

При цифровом формировании ЛЧМ будем иметь дискретное приближение линейного закона модуляции. Рис. 1.

Периоды  $T_1, T_2, T_3, \dots$  отличаются друг от друга на постоянную величину  $\Delta T$ , т.е.  $T_2 = T_1 + \Delta T$ , а  $T_3 = T_2 + \Delta T$  или  $T_3 = T_1 + 2\Delta T$  и т.д.

При этом соответствующие частоты  $f_1 = \frac{1}{T_1}, f_2 = \frac{1}{T_2}, f_3 = \frac{1}{T_3}, \dots$

На рис.1 представлены законы изменения частоты для идеального ЛЧМ-сигнала (непрерывная линия в верхней части рисунка) и дискретного приближения к нему (ступенчатая линия здесь же).

Идеальный ЛЧМ-сигнал можно представить в виде (15). Дискретное приближение к нему на  $k$ -м интервале времени можно записать следующей формулой:

$$u_d(t) = U_0(t) \sin 2\pi \left( f_c + \frac{\Delta F_k}{2} + k\Delta F_k \right) \tau, \quad (2)$$

где  $\tau \in [t_k, t_k + \Delta \tau_0]$ .

Пусть в середине этого отрезка частота идеального ЛЧМ-сигнала совпадает с частотой его дискретного приближения:

$$f_c + v_k t_k + v \frac{\Delta \tau_0}{2} = f_c + \frac{\Delta F_k}{2} + k\Delta F_k. \quad (3)$$

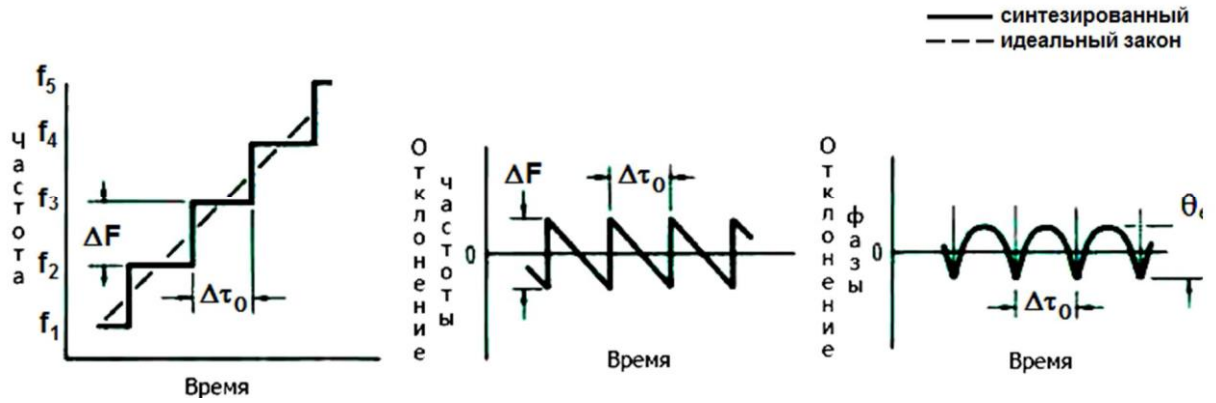


Рис.1. Дискретное приближение к линейному закону частотной модуляции

Решением уравнения (3) является:

$$v_k f = \frac{\Delta F_k}{\Delta \tau_0}. \quad (4)$$

Видно, что дискретное приближение приводит к частотным и фазовым ошибкам. При этом ошибки частоты на каждом временном отрезке меняются линейно. Очевидно, что амплитуда частотных ошибок составляет  $\frac{\Delta F_k}{2}$ .

Поскольку фаза является интегралом от частоты, то ошибки фазы  $\vartheta_s(t)$  будут на  $k$ -м интервале меняться по параболическому закону.

$$\begin{aligned} \theta_s(\tau) &= 2\pi \left( f_c + \frac{\Delta F_k}{2} + k\Delta F_k \right) (t_k + \tau) - 2\pi f_c (t_k + \tau) - \pi v (t_k + \tau)^2 \\ \theta_s(t) &= 2\pi \left( f_c + \frac{\Delta F_k}{2} + k\Delta F_k \right) t - 2\pi f_c t - \pi v t^2. \end{aligned}$$

Необходимое условие максимума данной функции имеет вид:

$$\frac{d\theta_s}{dt} = \frac{d\theta_s}{d\tau} = 2\pi \left( k\Delta F_k + \frac{\Delta F_k}{2} - v(t_k + \tau) \right) = 0.$$

Учитывая, что  $t_k = k \frac{\Delta F_k}{v}$ , для данного уравнения получим следующее решение:

$$\tau_m = \frac{\Delta F_k}{2v} = \frac{\Delta \tau_0}{2} \quad (5)$$

Разность фаз в точке  $\tau = 0$  равна:

$$\theta_s(\tau = 0) = 2\pi v t_m t_k - 2\pi v t_k^2, \text{ а в точке } \tau = \tau_m \text{ равна } \theta_s(\tau = \tau_m) = \pi v t_m^2.$$

Максимальное значение разности фаз (размах фазовой ошибки) будет иметь вид:

$$\max\{\Delta\theta_s\} = \theta_s(\tau = \tau_m) - \theta_s(\tau = 0) = \pi v (t_m - t_k)^2 = \pi v \tau_m^2 = \frac{\pi}{4} \Delta F \cdot \Delta \tau_0.$$

Фазовая ошибка меняется во времени, что при изменении частоты можно рассматривать как дрожание фазы – jitter у идеального ЛЧМ (см. рис. 1).

Дрожание фазы приводит к созданию шумовой составляющей у формируемого сигнала:

$$\tilde{u}(t) = U_0(t) \sin(2\pi f_c t + \pi v t^2 + \Delta\theta_s(t)). \quad (6)$$

Предполагая, что  $\Delta\theta_s(t) \ll 1$ , разложим тригонометрическую функцию в ряд Тейлора, ограничившись первым приближением:

$$\tilde{u}(t) \approx U_0(t) \sin(2\pi f_c t + \pi v t^2) + \Delta\theta_s(t) \cdot U_0(t) \cos(2\pi f_c t + \pi v t^2). \quad (7)$$

Итак, видим, что ступенчатое приближение к идеальному сигналу с ЛЧМ при малой амплитуде фазовой ошибки может быть представлено суммой двух сигналов: идеального ЛЧМ-сигнала и его копии, амплитуда которой промодулирована сигналом фазовой

ошибки. При малых фазовых ошибках вторым слагаемым можно пренебречь и считать, что приближенный сигнал практически не отличается от идеального. В данном случае важно подчеркнуть, что приближенный сигнал не имеет разрывов фазы и представляет собой элементы «оборванных» синусоид.

**Принцип метода прямого цифрового синтеза синусоидальных сигналов.** Выше было показано, что все радиосигналы с манипулированием параметров можно представить в виде элементарных примыкающих радиоимпульсов из «оборванных» синусоид с некоторой несущей  $f_c$  и длительностью  $\Delta t_0$ , которые можно записать в виде:

$$U(t) = \cos \varphi(t). \quad (8)$$

Для элементарного радиоимпульса фаза будет иметь вид:

$$\varphi(t) = 2\pi f_c t. \quad (9)$$

В методе ПЦС производится непосредственное вычисление значений функции (8), поэтому синтезатор представляет собой специализированный процессор.

Рассмотрим алгоритм его работы. Будем иметь в виду, что в цифровых вычислителях время меняется дискретно и его можно представить в виде:

$$t = nT_s, \quad (10)$$

где  $T_s$  – период тактовой частоты работы синтезатора  $f_s$ ,  $n$  – порядковый номер вычисления.

Таким образом, вычисляются отсчёты функции  $U(t) = U[n]$ .

Учитывая, что  $f_c = \frac{1}{2\pi} \frac{d\varphi(t)}{dt}$  или при линейном изменении фазы

$$f_c = \frac{1}{2\pi} \frac{\Delta\varphi}{\Delta t} = \frac{1}{2\pi} \frac{\Delta\varphi}{\Delta T_s}.$$

Итак, имеем

$$\varphi[n] = 2\pi f_c \cdot nT_s = 2\pi f_c \sum_1^n T_s = 2\pi \sum_1^n f_c T_s = \sum_1^n \Delta\varphi. \quad (11)$$

Видим, что для вычисления текущего значения фазы необходимо накапливать сумму её приращений. Для этого в синтезаторе используется устройство называемое накапливающим сумматором или аккумулятором фазы (accumulator).

Пусть фазовый аккумулятор имеет объем  $2^J$ . Будем считать, что при его заполнении фаза сигнала составляет  $2\pi$ , а после заполнения накопитель обнуляется. Этим обеспечивается периодичность изменения фазы во времени, поэтому приращение фазы можно представить как  $\Delta\varphi = 2\pi \cdot M$ , при том, что  $M \in [0,1]$ .

При шаге изменения времени, равном  $T_s$ , накопитель заполнится за время  $T_c = \frac{2^J}{M} T_s$ , которое и равно периоду формируемого колебания, Приращение фазы представляет собой двоичное число, поэтому частота формируемого элементарного сигнала, являющегося обрывком синусоиды, можно определить по формуле:

$$f_c = \frac{M \cdot f_s}{2^J}. \quad (12)$$

В данном случае  $M$  характеризует приращение фазы и фактически является кодом выходной частоты.

Код  $M$  можно оценить по значениям частот  $f_c$ ,  $f_s$  и разрядности аккумулятора фазы  $J$ , используя формулу

$$M = \left[ 2^J \cdot \frac{f_c}{f_s} + \frac{1}{2} \right], \quad (13)$$

где  $[x]$  целая часть числа  $x$ .

Минимальное приращение генерируемой частоты (частотное разрешение) будет иметь место при  $M=1$  и составит:

$$\delta f_c = \frac{f_s}{2^J}. \quad (14)$$

Видно, что частотное разрешение тем выше, чем больше разрядность накапливающего сумматора. Это значит, что при большой разрядности накопителя синтезатор может обеспечить синтез синусоидальных сигналов с частотами отличающимися на величину разрешения, т.е. сетку частот с малым шагом.

Максимальное значение фазы составляет  $\frac{2\pi(2^J - 1)}{2^J}$ . При постоянной величине  $T_s$  код частоты  $M$  определяет скорость изменения фазы от времени и при линейном законе – формируемую частоту  $f_c$ , поэтому, чем больше  $M$ , тем выше формируемая частота.

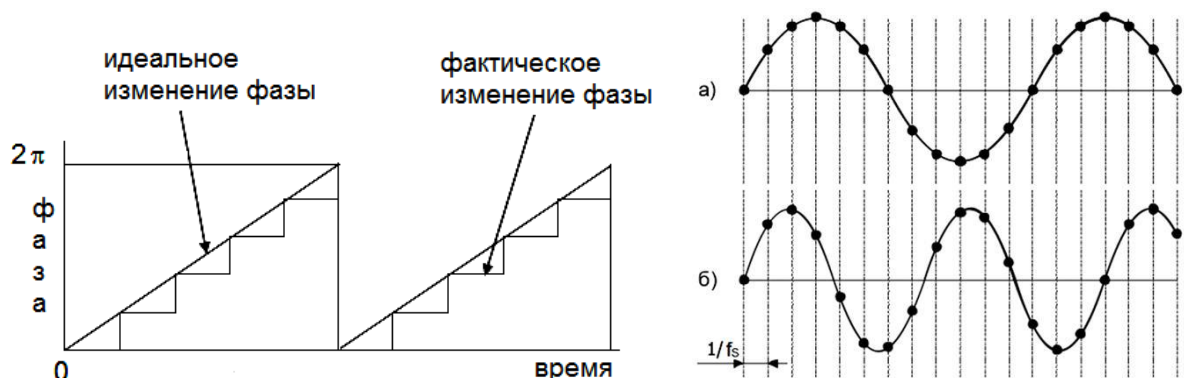
При цифровом методе формирования фазы линейный закон изменения фазы заменяется его дискретным (ступенчатым) приближением.

Для преобразования кода накопителя фазы в мгновенные отсчеты амплитуды синусоиды  $\varphi[n] \rightarrow \sin \varphi[n]$  используется ПЗУ со значениями в виде таблицы.

Отсчеты получаются с периодом тактовой частоты. Учитывается симметрия синуса, что позволяет хранить табличные данные только для 1/4 периода функции. При этом отсчеты для следующей четверти периода получаются путем чтения таблицы в обратном порядке, а для отрицательной полуволны умножением отсчетов на -1.

Рис.2,а иллюстрирует, каким образом выходной синусоидальный сигнал восстанавливается из отдельных отсчетов. На каждом новом периоде сигнала отсчеты лежат в новых местах (см. рис.9,б). При увеличении частоты формируемого колебания число отсчетов на его периоде будет уменьшаться. Теорема отсчетов (Найквиста) говорит о том, что для формирования функции синуса необходимо не менее двух отсчетов, поэтому максимально возможная генерируемая частота будет не более  $f_c \leq \frac{f_s}{2}$ . В

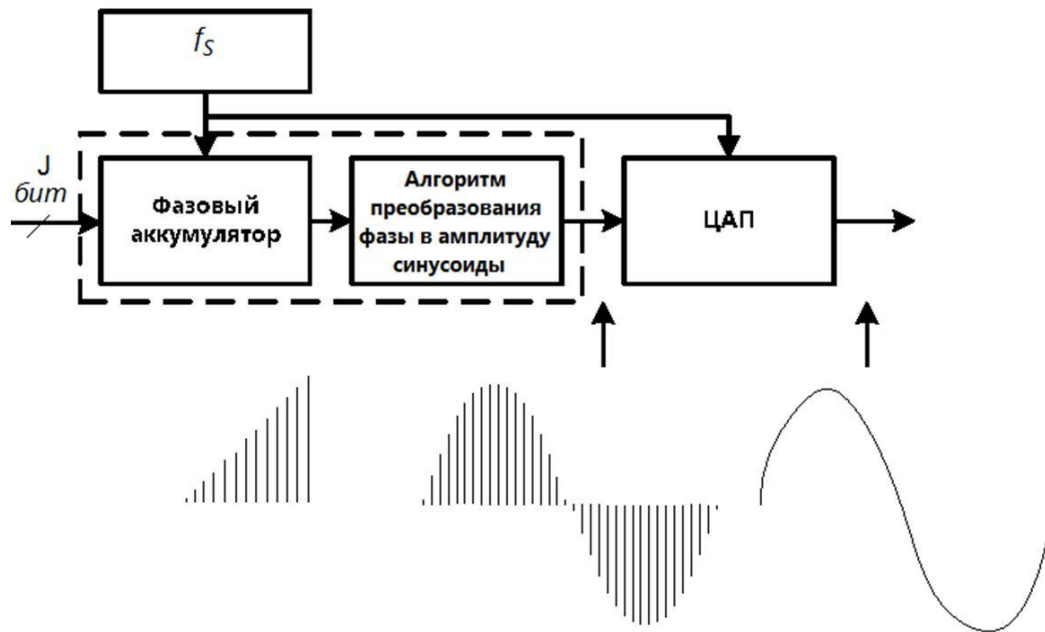
данном случае частота  $\frac{f_s}{2}$  называется частотой Найквиста.



**Рис.2.** Процесс накопления фазы (а) и положение выборок выходного сигнала для разных частот (б)

Далее дискретный синтезированный синусоидальный сигнал подвергается фильтрации с помощью ФНЧ.

Итак, принцип работы формирователя синусоидального сигнала можно представить в виде схемы, изображенной на рис.3.



**Рис.3.** Упрощённая структурная схема синтезатора ПЦС и диаграммы его работы

Очевидно, что синтезированный дискретно синусоидальный сигнал можно представить как идеальный сигнал, к которому добавлены шумы, связанные с дискретным приближением при его формировании и со стабильностью генератора  $G$  частоты синхронизации  $f_s$ . Точность задания частоты генератора определяется частотным разрешением  $\delta f_c = \frac{f_s}{2^J}$ . Это означает, что, если требуется получить сигнал частотой  $f_c$ , то ПЦС гарантированно позволит синтезировать сигнал с номиналами частоты либо  $\tilde{f}_c = f_c - \delta f_c$ , либо  $\tilde{f}_c = f_c + \delta f_c$ , которые при  $f_c \gg \delta f_c$  будут близки к необходимому значению. При этом отклонение частот тем меньше, чем выше разрядность аккумулятора фазы. Например, при  $\delta f_c = 2,2 \cdot 10^{-9}$  МГц в место требуемой частоты 20 МГц может быть получена частота 19,999999954 МГц или 20,000000009 МГц. Если такой точности недостаточно, то пользуются подбором значения частоты синхронизации. Такая возможность обусловлена тем, что синтезируемая частота равна  $f_c = M \cdot \frac{f_s}{2^J}$ . Видно, что ее номинал может быть немного подправлен (на величину частотного разрешения и больше) за счет небольшого изменения номинала частоты синхронизации (тактовой). Для реализации данного подхода созданы чипы гибридных синтезаторов, где в качестве опорного генератора ПЦС используется ГУН, подстраиваемый с помощью ФАПЧ (PLL) в зависимости от отклонения выходной частоты синтезатора от требуемого номинала. Такая структура позволяет получить на выходе точные значения частот, правда, шаг сетки будет такой же, как и у обычных ФАПЧ синтезаторов. Вследствие применения ГУН с ФАПЧ фазовый шум такого гибридного ПЦС намного меньше, чем у обычного синтезатора с ФАПЧ.

Важно отметить, что для минимальной и максимальной генерируемых частот имеют место следующие соотношения:

$$\min \{f_c\} = \frac{f_s}{2^J}; \quad \max \{f_c\} = \frac{f_s}{2^J} \cdot M. \quad (15)$$

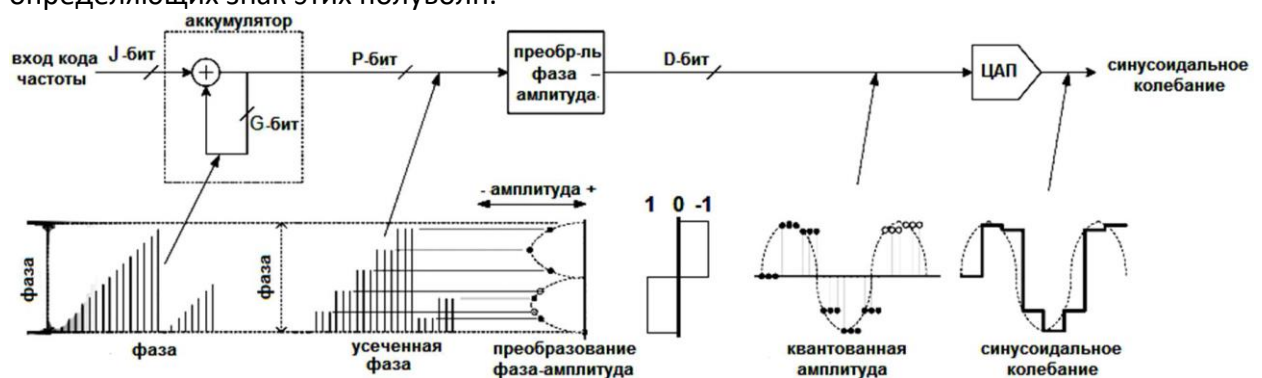
При больших значениях частоты синхронизации и разрядности фазового аккумулятора диапазон  $[\min \{f_c\}, \max \{f_c\}]$  оказывается достаточно большим.

Идеальный действительный гармонический сигнал в частотной области имеет лишь спектральные линии на частотах  $\pm f_c$ . При формировании сигнала всегда возникают

погрешности, часто носящие случайный характер, которые приводят к тому, что помимо спектральных линий полезного сигнала возникает множество других, носящих шумовой характер. Качество формируемого синусоидального сигнала оценивается уровнем его шума по отношению к уровню несущей. Шумы синтезатора ПЦС обусловлены:

- шумами тактового генератора (jitter),
- дискретным характером синтеза
- погрешностями при вычислениях фазы и амплитуды формируемого сигнала.

На рис.4 показана блок-схема синтезатора с более детальной, чем на приведенной выше, внутренней структурой преобразования цифровых сигналов. Видно, что интервал времени, за который фаза накапливает значение равное  $2\pi$ , разбивается на четыре интервала, на которых набег фазы составляет  $\pi/2$ . Далее формируется сигнал для прямого и обратного обращения к таблице синусов. На выходе функционального преобразователя формируются две полуволны кода амплитуды синусоиды одинаковой полярности. Далее значения кодов синуса для первой полуволны добавляются кодом логического нуля, а второй полуволны добавляются кодом логической единицы, определяющих знак этих полуволн.



**Рис.4.** Блок-схема синтезатора с внутренней структурой преобразования цифровых данных

**Шумы из-за усечения кода фазы.** В преобразователе фаза-амплитуда для определения значений синуса обычно используются либо алгоритмы вычисления, либо цифровые таблицы. Использование тех или иных схем, реализующих вычисление синусоидальной функции, зависит от того, какая из двух целей преследуется: увеличение точности формирования или увеличение быстродействия. Для обеспечения высокого быстродействия используются таблицы, записанные в ПЗУ, в которых для уменьшения ее объема уменьшается разрядность кода фазы с  $J$  до  $P$  разрядов, т.е. код фазы усекается.

На рис. 5 приведен практический пример изменения разрядности данных в процессе синтеза сигнала.

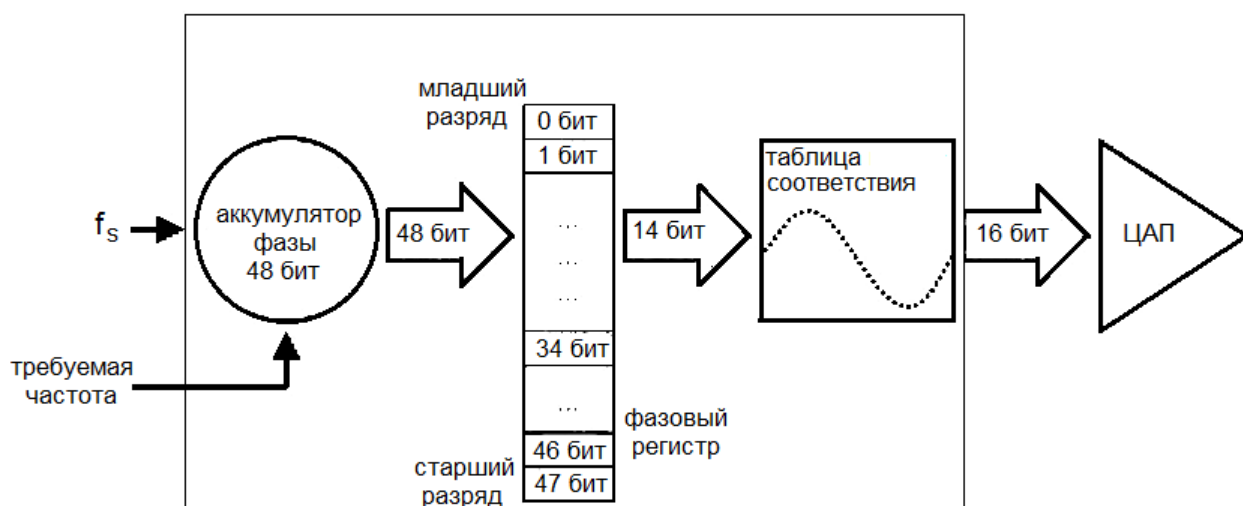


Рис. 5. Пример изменения разрядности данных в синтезаторе

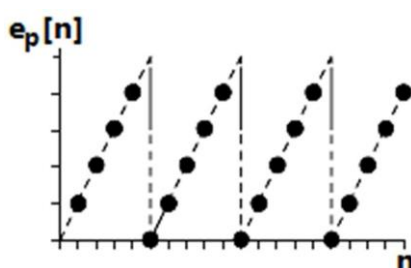


Рис. 6. Пример изменения от  $n$  ошибки фазы из-за усечения кода

Видим, что значения фазы вычисляются с высокой степенью точности (в нашем случае 48 бит). Далее младшие разряды фазы усекаются и на функциональный преобразователь подаются данные разрядностью 14 бит. Это вынужденная мера, вызванная необходимостью уменьшения размеров ПЗУ до разумных пределов. Действительно, если бы использовались все 48 бит, а каждый отсчет в ПЗУ кодировался бы одним байтом, то необходимый объем ПЗУ составил бы 4 Гб. В преобразователе добавляются два старших разряда для передачи информации о знаке полуволны синусоиды. Усечение кода фазы является внутренней операцией чипа ПЦС и извне изменить ничего нельзя.

Отбрасывание младших битов создает ошибки в представлении фазы. Данная погрешность носит периодический характер, т.к. в зависимости от кода частоты состояние аккумулятора фазы повторяется чаще или реже. В результате в спектре выходного сигнала появляются дополнительные составляющие, вызванные усечением кода фазы. На частотный диапазон и амплитуды этих составляющих влияют три основных фактора: разрядность аккумулятора фазы ( $J$  - бит) разрядность слова фазы после усечения ( $P$  - бит), разрядность кода частоты. При некоторых значениях кода частоты составляющие, вызванные усечением кода фазы, отсутствуют вовсе, в то время как при других они имеют максимальный уровень.

Аналитическое решение оценки помех из-за усечения фазы было дано в работе [67]. Следуя изложенному в ней подходу, получим необходимые оценки. Выражение, описывающее синтезируемый гармонический сигнал, представим в следующем виде:

$$U[n] = \sin \varphi[n] = \sin \left( 2\pi \cdot \frac{M}{2^J} \cdot n \right). \quad (16)$$

Сигнал  $\tilde{U}[n]$  на выходе функционального преобразователя с учётом округления разрядов фазы до  $P$  можно найти из выражения:



$$\tilde{U}[n] = \sin \varphi[n] = \sin \left( \frac{2\pi}{2^P} \cdot \text{Int} \left[ \frac{M}{2^{J-P}} \cdot n \right] \right) = \sin \left( \frac{2\pi}{2^J} (M \cdot n - e_p[n]) \right), \quad (17)$$

где  $2\pi \cdot e_p[n]$  – ошибка в фазе из-за усечения.

Очевидно, что функция  $e_p[n]$  равна:

$$e_p[n] = \frac{\varphi[n] - \tilde{\varphi}[n]}{2\pi} = \frac{1}{2^P} \left( \frac{M}{2^{J-P}} \cdot n - \text{Int} \left( \frac{M}{2^{J-P}} \cdot n \right) \right). \quad (18)$$

Видно, что в данном случае слово, которое необходимо записать в аккумулятор фазы, до момента округления равно  $2^{J-P}$ . Увеличение слова в аккумуляторе происходит с частотой синхронизации и выделенный объём заполняется за время:

$$T_p = \frac{2^{J-P}}{M} \cdot T_s. \quad (19)$$

Поэтому частота, с которой переполняется эта часть аккумулятора, равна:

$$f_p = \frac{M}{2^{J-P}} \cdot f_s \approx 2^P f_c. \quad (20)$$

Таким образом, частота округления превышает и, как правило, значительно, синтезируемую частоту.

Напомним, что объему аккумулятора  $2^J$  соответствует набег фазы, равный  $2\pi$ . Тогда объему  $2^{J-P}$  будет соответствовать набег фазы  $\frac{2\pi}{2^P}$ , поэтому амплитуда сигнала ошибки фазы составит:

$$\max \{ 2\pi \cdot e_p[n] \} = \frac{2\pi}{2^P}. \quad (21)$$

Итак, ошибка изменяется в пределах:

$$m0 \leq e_p[n] \leq 2^{-P}. \quad (22)$$

Отметим, что представленная на рис.6 функция является строго периодической, если отношение  $\frac{T_c}{T_p} = k$  является целым числом. Однако в общем случае это отношение является дробным и у сигнала ошибки появляются составляющие с другими частотами.

Основываясь на законе сохранения энергии, отметим, что появление дополнительных спектральных составляющих в сигнале ошибки будет приводить к уменьшению составляющих для преобладающих колебаний. Поскольку наша задача заключается в оценке худшего варианта для отношения спектра полезного сигнала к наибольшей спектральной составляющей сигнала ошибки, то нам можно ограничиться рассмотрением случая, когда  $k$  является целым числом.

Если обратиться к рис.5, то в данном характерном случае  $P=14$  и максимальная ошибка в фазе составит всего  $0,02^\circ$ , что много меньше  $2\pi$ . Поэтому можем воспользоваться тригонометрическим выражением для синуса разности углов, которое при  $\beta \ll 1$  имеет вид:

$$\sin(\alpha - \beta) \approx \sin(\alpha) - \beta \cdot \cos(\alpha). \quad (23)$$

С учётом этого выражение (17) можно представить следующей формулой:

$$\tilde{U}[n] \approx \sin \left( \frac{2\pi \cdot M}{2^J} n \right) - 2\pi \cdot e_p[n] \cdot \cos \left( \frac{2\pi \cdot M}{2^J} n \right), \quad (24)$$

Согласно (24), сигнал, усечённый по фазе, состоит из двух сигналов: исходного гармонического (первое слагаемое) и остаточного сигнала ошибки, являющегося произведением исходного сигнала на сигнал ошибки фазы. В рассматриваемом случае

сигнал ошибки является периодическим и «пилообразным». Его можно представить в виде ряда Фурье:

$$e_p[n] = 2^{-(P+1)} + \frac{2^{-P}}{\pi} \sum_{k=1}^{\infty} \frac{1}{k} \sin\left(2\pi k \frac{M}{2^{J-P}} \cdot n\right) = 2^{-(P+1)} + \frac{2^{-P}}{\pi} \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi k f_p \cdot t) = e_p[t], \quad (25)$$

где  $t = n \cdot T_s$ .

С учетом (25) равенство (24) преобразуется к виду:

$$\begin{aligned} \tilde{U}[n] &\approx \sin(2\pi f_c \cdot t) - 2\pi \cdot 2^{-(P+1)} \cdot \cos(2\pi f_c \cdot t) - 2\pi \left[ \frac{2^{-P}}{\pi} \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi k f_p \cdot t) \right] \cdot \cos(2\pi f_c \cdot t) \approx \\ &\approx \sin(2\pi f_c \cdot t - \vartheta_p) - \left[ 2^{-P+1} \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi k f_p \cdot t) \right] \cdot \cos(2\pi f_c \cdot t), \end{aligned} \quad (26)$$

где  $\vartheta_p = \frac{\pi}{2^P}$ .

Выражение для  $\sin(2\pi f_c \cdot t - \vartheta_p)$  получено путём обратного преобразования первых двух слагаемых в (26) с использованием формулы (23).

Для дальнейших оценок рассмотрим произведение  $\frac{2^{-P+1}}{k} \sin(2\pi k f_p \cdot t) \cdot \cos(2\pi f_c \cdot t)$ .

Ему соответствует полусумма синуса суммы и разности фаз этих тригонометрических функций. Таким образом, каждое слагаемое суммы порождает две спектральных составляющих на частотах  $(k f_p \pm f_c)$ .

Итак, кроме полезного сигнала на синтезируемой частоте  $f_c$  из-за усечения фазы возникают шумовые составляющие на частотах  $(k f_p \pm f_c)$ :

$$(SNR)_{pk} = 20 \lg 2^P \cdot k = 6,02 \cdot P + 20 \lg k. \quad (27)$$

Наибольшей спектральной амплитудой, равной  $2^{-P}$ , обладают составляющие с  $k = 1$ , поэтому:

$$\min(SNR)_p = 6,02 \cdot P \text{ (дБ)}.$$

На рис.7 приведён пример амплитудно-частотного спектра сигнала с усечённой фазой при его следующих параметрах: количество разрядов аккумулятора фазы  $J = 14$ , количество разрядов фазы, поступающих на функциональный преобразователь  $P = 11$ , код частоты  $M = 321$ .

$$J=14, P=11, M=321$$

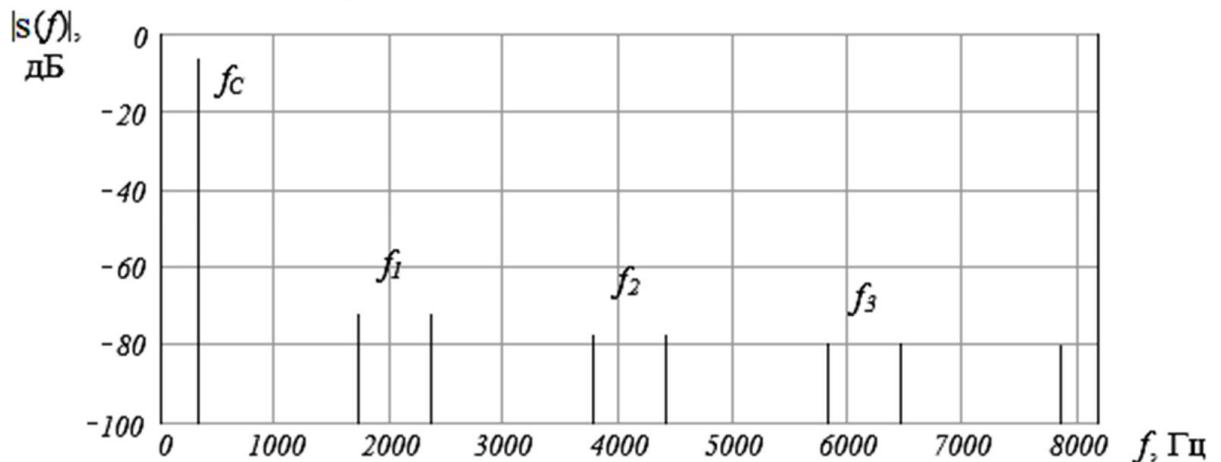


Рис.7. Пример спектра сигнала с усечённой фазой

Из рисунка видно, что при усечении фазы наряду с основным сигналом на частоте  $f_c = 321$  Гц имеют место побочные составляющие из-за усечения фазы на частотах: 1

$f_1 = 2048 \pm 321$  Гц,  $f_2 = 4096 \pm 321$  Гц и т.д. Уровень наибольшей из гармоник ошибки по отношению к уровню основного сигнала составляет -66 дБ.

Следует отметить, что существуют методы уменьшения влияния усечения кода фазы, основанные на добавлении к фазовой информации псевдослучайного шума. Таким образом, удастся уменьшить энергию соответствующих побочных компонентов, зато при этом увеличивается общий шумовой порог.

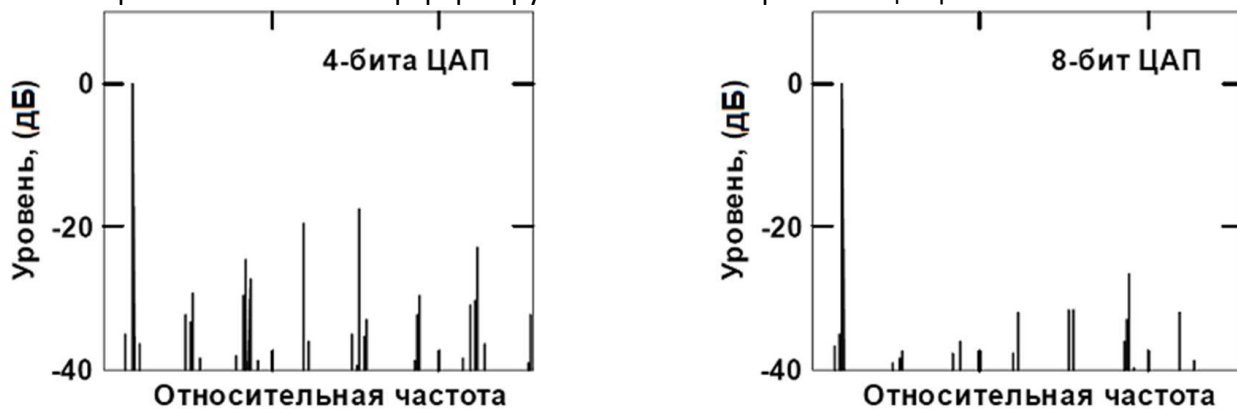
**Шумы из-за ограниченной разрядности ЦАП.** В данном методе для преобразования кодов фазы в аналоговые отсчеты функции синуса используются цифро-аналоговые преобразователи (ЦАП), имеющие разрядность  $D$  и связанное с ней конечное разрешение по амплитуде синтезируемого сигнала. На рис.8 приведен пример преобразования цифрового сигнала в аналоговый синусоидальный с помощью ЦАП, имеющего ограниченную разрядность. Частые точки показывают синусоиды, которые получались бы при высокой разрядности ЦАП на каждом такте преобразования. Вертикальные линии отображают моменты времени, в которые должно происходить обновление значений амплитуды в идеальном случае. Ступенчатое приближение имеет место из-за ограниченной разрядности ЦАП. Вариации около нулевого уровня отображают сигнал ошибки из-за квантования сигнала при цифроаналоговом преобразовании, приводящем к внутренним помехам синтезатора.



**Рис.8.** Причина возникновения шумов квантования амплитуды из-за конечной разрядности ЦАП и их вариации во времени

Этот сигнал определяется разницей между ступенчатым приближением синусоиды на выходе ЦАП и значениями идеальной синусоиды. Очевидно, что внутренние помехи создают дополнительные составляющие в спектре формируемого сигнала, а их частоты определяются комбинациями между частотой дискретизации и частотой изменения кода.

На рис. 9 показан спектр формируемого сигнала при помощи ЦАП с  $D = 4$  и  $D = 8$ .



**Рис.9.** Спектры сигналов на выходе 4-битного и 8-битного ЦАП

Как и следовало ожидать, с ростом разрядности ЦАП спектральные амплитуды шумов уменьшаются, поскольку при этом уменьшается и ошибка квантования. Обычно на практике используют 10–16-битные ЦАП. При этом в схемах с высоким быстродействием применяются ЦАП с меньшим числом разрядов (не выше 12 бит).

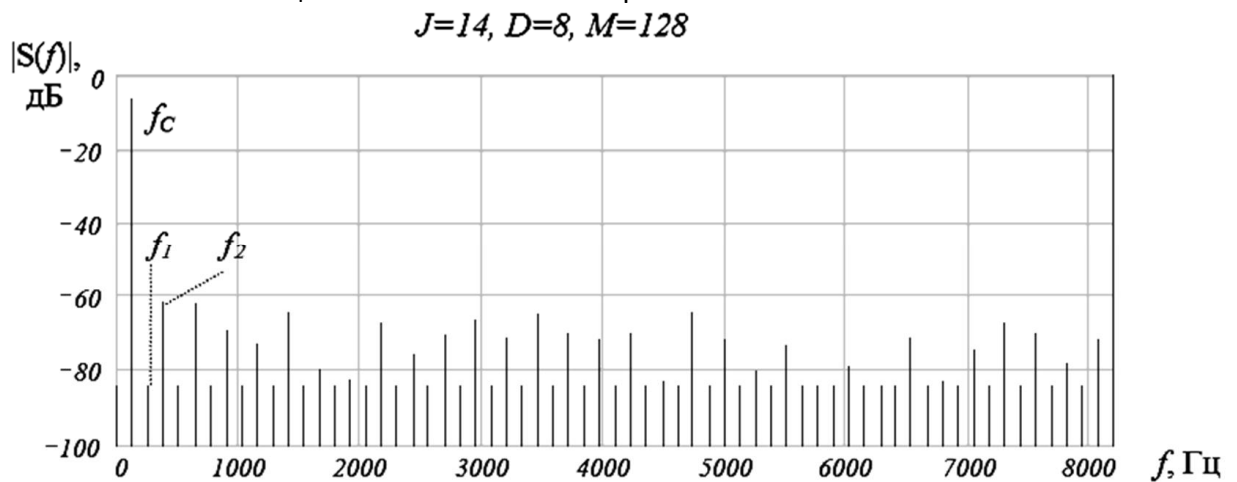
Получим формулы, позволяющие оценивать шумы амплитудного квантования. Пусть сигнал  $\tilde{U}[n]$  получен на выходе ЦАП с разрядностью  $D$ . Его можно записать следующим образом:

$$\tilde{U}[n] = \frac{1}{2^{D-1}} \cdot \text{Int} \left( 2^{D-1} \cdot \sin \left( 2\pi \cdot \frac{M}{2^J} \cdot n \right) \right). \quad (28)$$

С учётом (28) ошибки квантования амплитуды остаточный сигнал  $\delta U[n]$  можно представить в виде:

$$\delta U[n] = \frac{1}{2^{D-1}} \cdot \text{Int} \left( 2^{D-1} \cdot \sin \left( 2\pi \cdot \frac{M}{2^J} \cdot n \right) \right) - \sin \left( 2\pi \cdot \frac{M}{2^J} \cdot n \right). \quad (29)$$

На рис.10 изображён спектр синтезируемого сигнала. Обратим внимание на то, что наряду со спектром синусоидального сигнала на частоте  $f_c = 123$  Гц имеет спектр побочных составляющих сигнала ошибки на кратных ей частотах.



**Рис.10** Спектр синтезируемого сигнала с шумами квантования при разрядности ЦАП  $D=8$

Предполагая стохастический характер сигнала ошибки и равномерное распределение его значений на отрезке  $\delta U \in [-a, a]$ , оценим второй момент остаточного сигнала. Если число  $a$  задано  $D$  разрядами, то шаг квантования равен величине, приходящейся на младший разряд этого числа  $\delta a = \frac{a}{2^{D-1}}$ . Учтем, что при использовании его округления до  $D$  разрядов ошибка квантования не будет превышать половины значения, приходящегося на один младший разряд:

$$\max(|\delta U[n]|) \leq \frac{\delta a}{2}, \quad (30)$$

т.е. она будет находиться (см. рис.21) на отрезке:

$$-\frac{\delta a}{2} \leq \delta U[n] \leq \frac{\delta a}{2}. \quad (31)$$

$C/\mathcal{W}$  в данном случае можно определить как отношение средней мощности синусоидального сигнала на его периоде к средней мощности сигнала ошибки. Поскольку у ошибки квантования  $\delta U[n]$  и синусоидального сигнала первые моменты равны нулю, то отношение  $C/\mathcal{W}$  можно оценить как отношение их дисперсий.

Дисперсию  $\sigma_0^2$  ошибки можно найти по формуле:

$$\sigma_0^2 = \frac{\delta a^2}{12} = \frac{a^2}{3 \cdot 2^{2D}}. \quad (32)$$

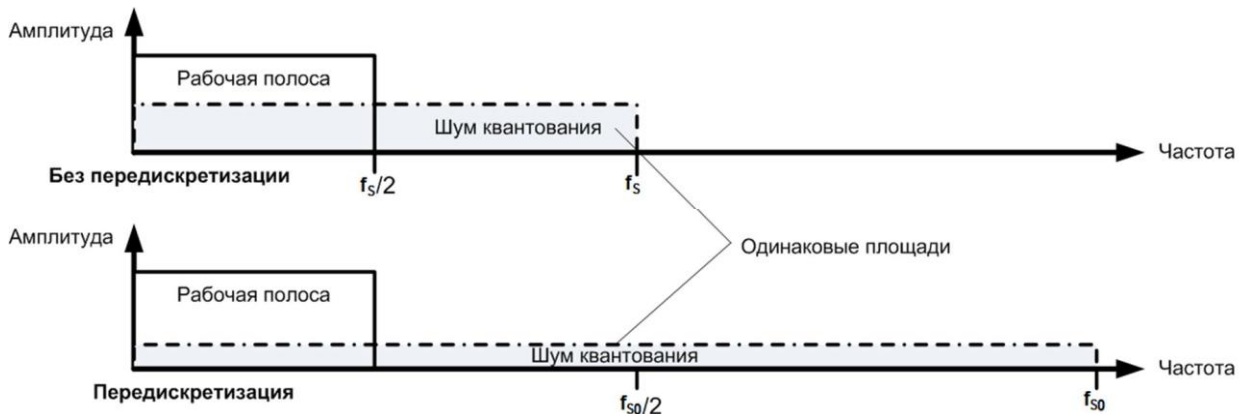
Средняя мощность синусоиды с амплитудой  $a$  равна  $\sigma^2 = \frac{a^2}{2}$ , поэтому для оценки  $C/\text{Ш}$  получим следующее выражение:

$$SNR_D = 10 \cdot \lg\left(\frac{\sigma^2}{\sigma_0^2}\right) = 10 \cdot \lg(3 \cdot 2^{2D-1}) = 6,02 \cdot D + 1,76 \approx 6,02 \cdot D \quad (\text{дБ}). \quad (33)$$

Согласно формуле (33)  $SNR_D$  прямо пропорционально количеству разрядов ЦАП. Важно отметить, что приведенное выше соотношение справедливо только для случая, когда используется полная шкала ЦАП. При уменьшении уровня выходного сигнала мощность шумов квантования не меняется. Однако при этом отношение  $C/\text{Ш}$  ухудшается пропорционально уменьшению используемой части шкалы ЦАП [61].

**Улучшение качества сигнала путем использования эффекта передискретизации.** Некоторого уменьшения уровня шумов квантования в рабочей полосе частот можно добиться использованием эффекта передискретизации, когда тактовая частота увеличивается в  $k$  раз по сравнению с удвоенной частотой Найквиста. Этот случай называют  $k$ -кратной передискретизацией. При этом  $k = \frac{f_{s0}}{f_s} > 1$ , где  $f_{s0}$  – новая частота дискретизации.

Рис.11 иллюстрирует эффект изменения спектральной плотности шума от передискретизации. Площадь прямоугольника характеризует энергию шумов квантования и зависит от разрядности ЦАП. При передискретизации энергия шума, равная площади второго прямоугольника, остается той же. Однако из-за увеличения длины прямоугольника его высота уменьшается во столько же раз, поэтому на рабочую полосу частот теперь приходится меньшая ее часть, а это означает, что отношение  $C/\text{Ш}$  при передискретизации повышается.



**Рис.11 Влияние передискретизации на величину SNR**

С учетом этих рассуждений в случае передискретизации уравнение (33) можно преобразовать к следующему виду:

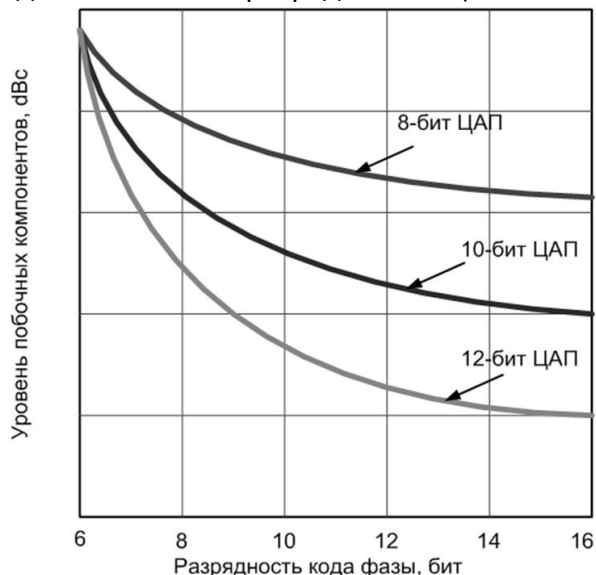
$$SNR_D = (6,02 \cdot D + 1,76 + 10 \lg k) \approx 6,02 \cdot D + 10 \lg k \quad (\text{дБ}). \quad (34)$$

Кроме того, при выполнении условия  $f_s > (m+1) \cdot \max f_c$ , где  $m$  – порядок побочного компонента, а  $\max f_c$  – верхняя рабочего диапазона частот, передискретизация позволяет подавить в спектре формируемого сигнала эти побочные компоненты с помощью ФНЧ, пропускающего частоты из рабочей полосы. Это означает,

что применение передискретизации при равных прочих условиях позволяет сформировать сигнал более высокого качества.

Очевидно, что шумы квантования амплитуды и шумы усечения фазы конкурируют между собой. Это говорит о том, что коды  $D$  и  $P$  каким-то образом связаны между собой.

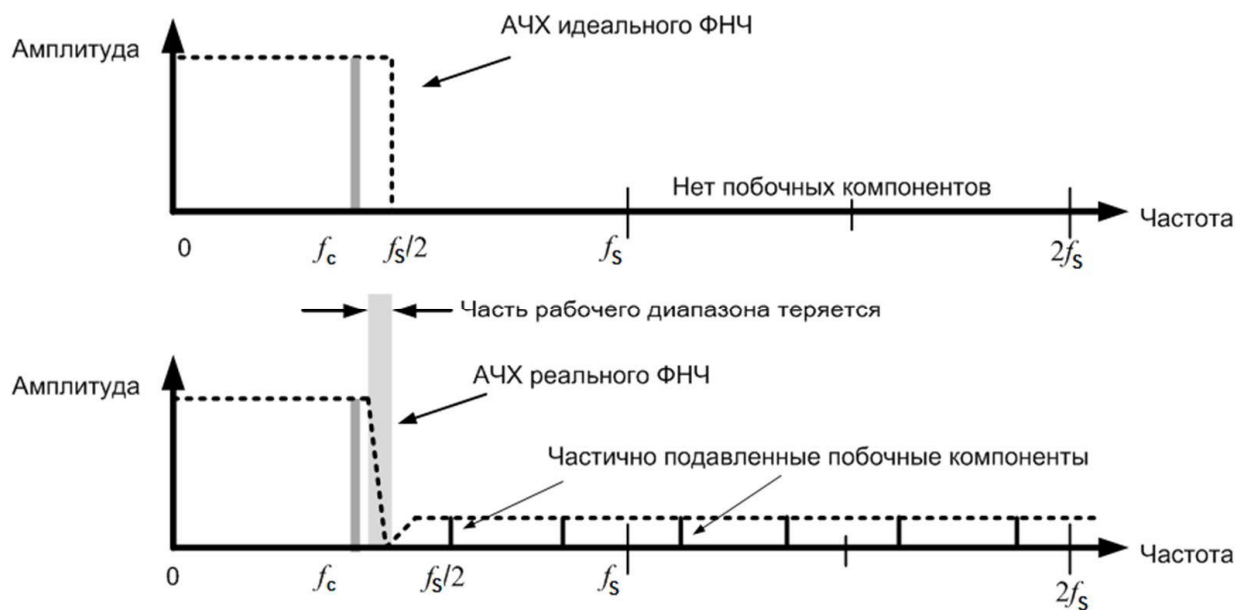
На рис. 12 приведены зависимости уровня шумов от разрядности усеченного кода фазы для ЦАП разной разрядности (т.е. суммарный шум от усечения фазы и квантования амплитуды). Видно, что при малой разрядности кода фазы уровень шума сильно зависит от  $P$ . Однако, начиная с некоторых значений  $P$ , уровень шума практически не меняется и, следовательно, он определяется только разрядностью ЦАП.



**Рис.12** Уровень шума в зависимости от разрядности усеченного кода фазы при заданной разрядности ЦАП

Поскольку  $SNR_D \approx 6 \cdot D$ , а  $SNR_P \approx 6 \cdot P$ , то при  $P = D + 1$  шумы квантования ЦАП будут преобладать над шумами из-за усечения фазы на 6дБ, поэтому на практике принято выбирать  $P$  на 1–2 разряда выше, чем  $D$ . Учитывая, что разрядность ЦАП ограничена техническими возможностями, то считается, что качество синтезируемого этим методом сигнала определяется ЦАП, а некоторого его улучшения можно добиться путем передискретизации.

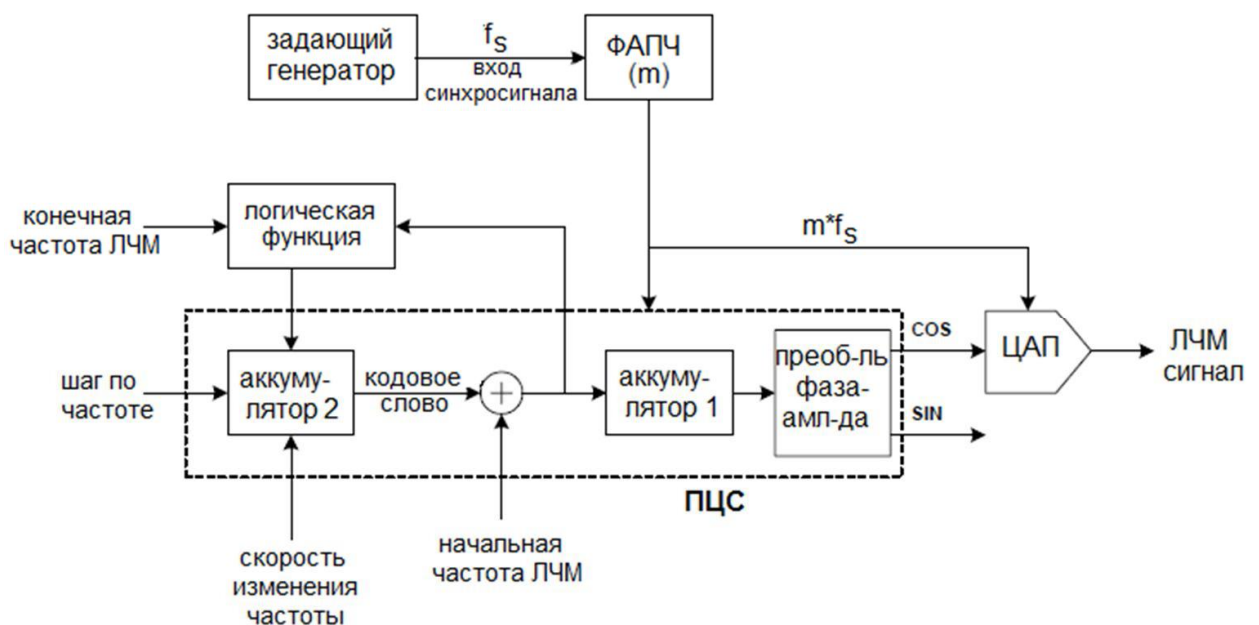
В методе ПЦС рабочая полоса частот выделяется с помощью фильтра нижних частот (ФНЧ). В связи с этим при проектировании необходимо учитывать следующие общие соображения. Во-первых, что только идеальный ФНЧ имеет единичный коэффициент передачи на частотах от 0 до частоты Найквиста и нулевой коэффициент передачи на других частотах (рис.13,а). Однако из теории радиотехники известно, что идеальный ФНЧ физически не реализуем. Существует много различных схем реальных ФНЧ. На рис.13,б приведена типичная АЧХ реального ФНЧ. Отметим, что спад АЧХ является постепенным, в результате этого из рабочего диапазона теряется некоторая полоса рабочих частот. Кроме этого, за пределами частоты Найквиста затухание реального фильтра отлично от нуля. Это означает, что шумовые компоненты в спектре формируемого сигнала такой ФНЧ будет подавлять лишь частично.



**Рис.13** Характеристики идеального (а) и реального (б) ФНЧ

Кроме того, исследования реальных ФНЧ показывают, что их АЧХ не является равномерной и в лучшем случае удастся получить с АЧХ с неравномерностью не хуже 10 %.

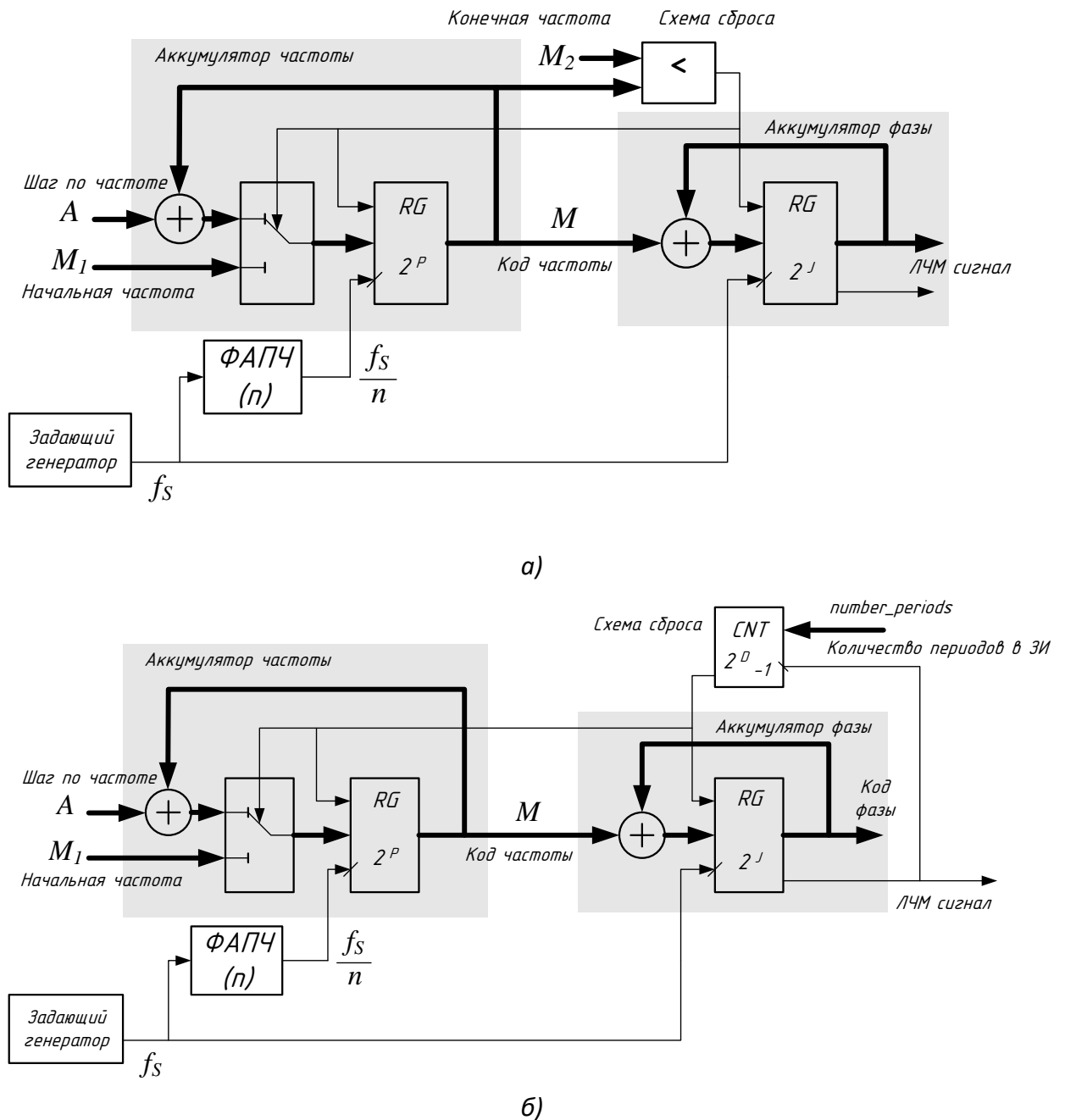
### Синтез модулированных радиосигналов.



**Рис.14** Блок-схема ЛЧМ-синтезатора на основе метода ПЦС

## Схема синтезатора

Возможны некоторые варианты схемы синтезатора ЛЧМ сигнала (рис.15).



**Рис.15** Блок-схема синтезатора ЛЧМ сигнала на основе метода ПЦС  
(а – сброс по достижению заданного кода конечной частоты  $M$ ,  
б – сброс по количеству сформированных периодов в ЛЧМ сигнале)



## Выбор параметров

Для генерации ЛЧМ сигнала необходимо задать следующие параметры:

$f_s$  – тактовая частота;

$n$  – коэффициент деления тактовой частоты для формирования тактовой частоты регистра аккумулятора частоты;

$A$  – шаг изменения кода частоты;

$M_1$  – код начальной частоты;

$M_2$  – код конечной частоты;

$P$  – разрядность регистра аккумулятора частоты;

$J$  – разрядность регистра аккумулятора фазы;

$D$  – разрядность счётчика периодов в ЗИ;

$number\_periods$  – количество периодов в ЗИ.

Как было показано выше максимальное значение разности фаз (размах фазовой ошибки) будет составлять:

$$\max\{\Delta\theta_\epsilon\} = \theta_\epsilon(\tau = \tau_m) - \theta_\epsilon(\tau = 0) = \pi\nu(t_m - t_k)^2 = \pi\nu\tau_m^2 = \frac{\pi}{4} \Delta F \cdot \Delta\tau_0.$$

Переходя к параметрам схемы синтезатора, получим

$$\Delta F = f_s \cdot \frac{A}{2^J},$$

$$\Delta\tau_0 = T_{SF} = n \cdot T_s = \frac{n}{f_s}, \text{ тогда}$$

$$\max\{\Delta\theta_\epsilon\} = \frac{\pi}{4} \Delta F \cdot \Delta\tau_0 = \frac{\pi}{4} \cdot f_s \cdot \frac{A}{2^J} \cdot \frac{n}{f_s} = \frac{\pi}{4} \cdot \frac{A}{2^J} \cdot n.$$

Т.е. для обеспечения минимальной погрешности по фазе  $n$  должно быть равно 1, т.е. аккумулятор частоты нужно тактировать с той же частотой, что и аккумулятор фазы (код частоты должен меняться на каждом такте).

Но в этом случае разрядность аккумулятора фазы должна составлять

$$\left. \begin{aligned} f_c &= M \cdot \frac{f_s}{2^J} \\ k &= \frac{f_s}{f_c} \end{aligned} \right\} \Rightarrow k = \frac{2^J}{M}$$

$$\text{При } f_s = 125 \text{ (МГц) и } f_c = 12,5 \dots 1250 \text{ (кГц) } k = \frac{f_s}{f_c} = 100 \dots 10\,000.$$

Рассмотрим самый «тяжёлый» случай:

$$\max\{T_{ЗИ}\} = 0,1 \text{ с;}$$

$$\min\{\Delta f\} = 0,1 \cdot \min\{f_c\} = 1,25 \text{ (кГц) пусть даже } 12,5 \text{ (кГц).}$$

$$\text{Максимально количество тиков за время длительности ЗИ} \\ s = f_s \cdot \max\{T_{ЗИ}\} = 125 \cdot 10^6 \cdot 0,1 = 12,5 \cdot 10^6$$

$$f_c = M \cdot \frac{f_s}{2^J} \Rightarrow 2^J = M \cdot \frac{f_s}{f_c} = s \cdot \frac{f_s}{\min\{\Delta f\}} = 12,5 \cdot 10^6 \cdot \frac{125 \cdot 10^6}{12,5 \cdot 10^3} = 12,5 \cdot 10^{10} \Rightarrow J = 37.$$

Уменьшить разрядность аккумулятора фазы можно, если код частоты будет меняться не на каждом такте тактовой частоты а только на  $n$ -ом (т.е. тактовую частоту аккумулятора частоты делить на  $n$ ).

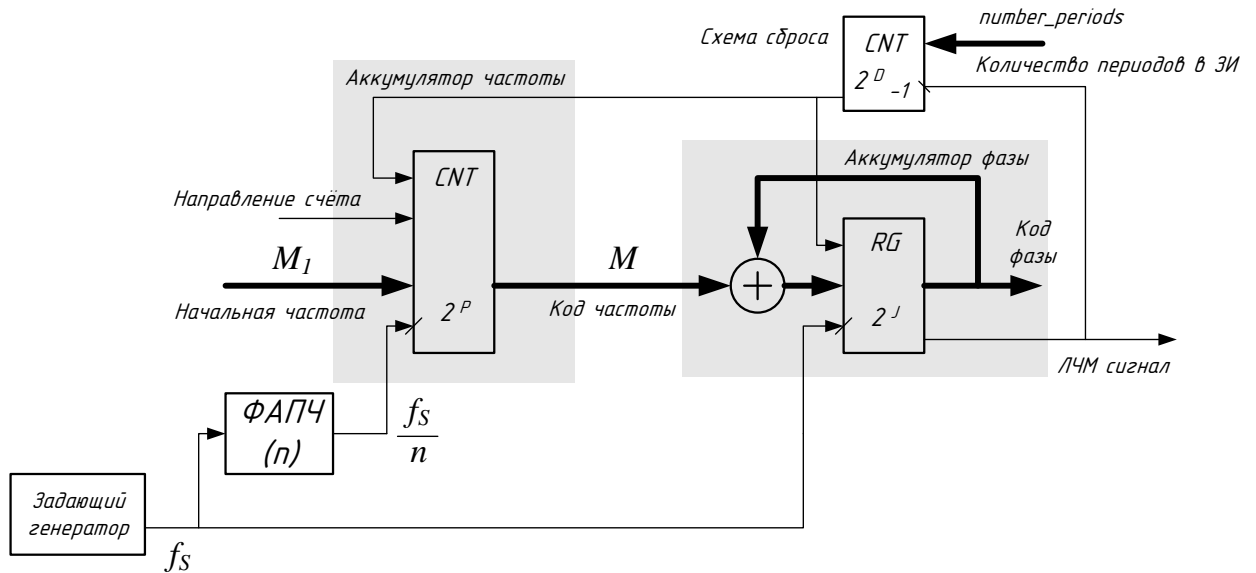
Если разрядность аккумулятора фазы  $J$  будет составлять 24 значения параметров для всех предполагаемых (возможных) режимов работы синтезатора будем иметь

№ п/п	$f_1$ , кГц	$f_2$ , кГц	$T_{ЗИ}$ , мс	$n$	$A$	$M_1$	$M_2$	$k$	$num\_periods$	$num\_steps$	$error\_phase$ , °
1	7	15	1	$2^7$	1	940	1917	0,025	11	977	0,041
2	7	15	16	$2^{11}$	1	940	1917	0,25	170	977	0,047
3	70	90	1	$2^5$	1	9396	13302	0,025	85	3906	0,29
4	70	90	64	$2^{11}$	1	9396	13302	2,5	5412	3906	0,29
5	262,5	337,5	1	$2^4$	1	35233	43046	0,05	292	7813	0,92
6	262,5	337,5	32	$2^9$	1	35233	43046	2	9332	7813	0,93
7	612,5	787,5	1	$2^2$	1	82209	113459	0,05	729	31250	2,43
8	612,5	787,5	16	$2^6$	1	82209	113459	0,5	11663	31250	2,43
9	875	1125	1	$2^2$	1	117441	148691	0,05	991	31250	3,19
10	875	1125	8	$2^5$	1	117441	148691	0,05	7931	31250	3,19

Обобщая полученные данные будем иметь

Обозначение	Название	Значение
$f_s$	тактовая частота	$f_s = 125$ (МГц)
$n$	коэффициент деления тактовой частоты для формирования тактовой частоты регистра аккумулятора частоты	$1...2^{11}$
$A$	шаг изменения кода частоты	1
$M_1$	код начальной частоты	$0...2^{17}$
$M_2$	код конечной частоты	$0...2^{18}$
$P$	разрядность регистра аккумулятора частоты	18
$J$	разрядность регистра аккумулятора фазы	24
$D$	разрядность счётчика периодов в ЗИ	17
$number\_periods$	количество периодов в ЗИ	$0...2^{17}$

Чтобы уменьшить период изменения кода частоты (максимальному приближению ступенчатой функции кода частоты к прямой (линейному закону изменения)) необходимо, чтобы шаг изменения кода частоты был минимальным (=1). В этом случае аккумулятор частоты трансформируется в счётчик и схема синтезатора принимает вид (рис.16).



**Рис.16** Блок-схема синтезатора ЛЧМ сигнала на основе метода ПЦС

При этом возможны затруднения при формировании коротких ЗИ с большой девиацией на высокой частоте. При этом необходимо быстро менять код частоты и тактовой частоты может быть недостаточно.

(т.е. увеличивая значение регистра аккумулятора фазы на единицу, тактовой частоты  $f_s$  может оказаться недостаточно, чтобы возникло переполнение)

Такая ситуация возникнет когда

$$\frac{f_s \cdot T_{\text{ЗИ}}}{\text{number\_step}} < 1 \Rightarrow f_s < \frac{\text{number\_step}}{T_{\text{ЗИ}}},$$

где  $\text{number\_step} = |M1 - M2|$  – количество шагов изменения кода частоты.

Учитывая, что  $M = \left\lceil 2^J \cdot \frac{f_c}{f_s} + \frac{1}{2} \right\rceil$ , получим

$$f_s < \frac{\text{number\_step}}{T_{\text{ЗИ}}} = \frac{|M1 - M2|}{T_{\text{ЗИ}}} = \frac{1}{T_{\text{ЗИ}}} \cdot \frac{2^J}{f_s} \cdot |f_{c1} - f_{c2}| = \frac{1}{T_{\text{ЗИ}}} \cdot \frac{2^J}{f_s} \cdot \Delta f = 0,13421778 \cdot \frac{\Delta f}{T_{\text{ЗИ}}}$$

$$f_s < \sqrt{\frac{\Delta f}{T_{\text{ЗИ}}}} \cdot 2^J = 2^{12} \sqrt{\frac{\Delta f}{T_{\text{ЗИ}}}} = 4096 \sqrt{\frac{\Delta f}{T_{\text{ЗИ}}}}$$

При  $J = 24$  и  $f_s = 125$  (МГц) получим

$$\frac{\Delta f}{T_{\text{ЗИ}}} > \left( \frac{f_s}{4096} \right)^2 = \left( \frac{125 \cdot 10^6}{4096} \right)^2 = 931\,322\,574,615478\,515625.$$

Так, например, при  $\Delta f = 200$  кГц и  $T_{\text{ЗИ}} = 0,5$  мс это неравенство не выполняется, и следовательно, такой режим сформировать с помощью данного синтезатора не возможно.

Но следует заметить, что необходимость использование таких режимов для работы гидролокатора предполагается маловероятным.

(если это ограничение необходимо преодолеть, нужно увеличить тактовую частоту аккумулятора фазы  $f_s$  или уменьшить разрядность аккумулятора фазы  $J$ . Можно также увеличить шаг изменения кода частоты  $A$ )

## Пример расчёта

### Дано:

частота синхронизации (тактовая частота):

$$f_s = \frac{1}{T_s} = 125 \cdot 10^6 \text{ (Гц);}$$

шаг изменения времени

$$T_s = \frac{1}{f_s} = \frac{1}{125 \cdot 10^6} = 8 \cdot 10^{-9} \text{ (с);}$$

синтезируемые частоты

$$f_c = (875 \dots 1125) \cdot 10^3 \text{ (Гц);}$$

длительность импульса

$$T_{\text{ЗИ}} = 32 \cdot 10^{-3} \text{ (с);}$$

### Расчёт:

Коды крайних частот

$$M1 = \left[ 2^J \cdot \frac{f_{c1}}{f_s} + \frac{1}{2} \right] = \left[ 2^{24} \cdot \frac{875 \cdot 10^3}{125 \cdot 10^6} + \frac{1}{2} \right] = 117441$$

$$M2 = \left[ 2^J \cdot \frac{f_{c2}}{f_s} + \frac{1}{2} \right] = \left[ 2^{24} \cdot \frac{1125 \cdot 10^3}{125 \cdot 10^6} + \frac{1}{2} \right] = 150995 .$$

(здесь и далее  $[x]$  – целая часть числа  $x$ )

Тогда необходимое количество шагов изменения частоты

$$\text{num\_periods} = M1 - M2 = 33554 .$$

Требуемый коэффициент деления тактовой частоты  $f_s$

$$n = 2^{\left\lceil \log_2 \left( f_s \cdot \frac{T_{\text{ЗИ}}}{\text{num\_periods}} \right) \right\rceil} = 2^7 = 128 .$$

(если не обеспечивать кратность степени 2, то можно считать  $n = \left\lceil f_s \cdot \frac{T_{\text{ЗИ}}}{\text{num\_periods}} \right\rceil = 119$  )

Тактовая частота аккумулятора частоты

$$f_{FS} = \frac{1}{T_{FS}} = \frac{f_s}{n} = 976562,5 \text{ (Гц).}$$

При этом длительность ЗИ будет составлять

$$T_{\text{ЗИ}} = \frac{\text{num\_periods}}{f_{FS}} = 0,0344 \text{ (с).}$$

Количество целых периодов ЛЧМ сигнала

$$\text{num\_periods\_lfm} = \left\lceil T_{\text{ЗИ}} \cdot \frac{f_{c1} + f_{c2}}{2} \right\rceil = 34359 .$$

После расчёта параметров синтезатора необходимо уточнить параметры формируемого ЛЧМ импульса.

$$f_{c1} = M1 \cdot \frac{f_s}{2^J} = 874999,9106$$

$$f_{c2} = M2 \cdot \frac{f_s}{2^J} = 1125004,1425$$

$$T_{3И} = num\_periods\_lfm \cdot \frac{2}{f_{C1} + f_{C2}} = 0,034359$$

Таким образом, параметры синтезатора должны быть

Обозначение	Название	Значение
$f_s$	тактовая частота	$f_s = 125$ (МГц)
$n$	коэффициент деления тактовой частоты для формирования тактовой частоты регистра аккумулятора частоты	$2^7$
$A$	шаг изменения кода частоты	1
$M_1$	код начальной частоты	117441
$M_2$	код конечной частоты	150995
$P$	разрядность регистра аккумулятора частоты	18
$J$	разрядность регистра аккумулятора фазы	24
$D$	разрядность счётчика периодов в ЗИ	19
$number\_periods$	количество периодов в ЗИ	34 359

При этом параметры формируемого ЛЧМ сигнала будут следующими

Обозначение	Название	Значение
$f_{C1}$	начальная частота	874 999,9106 (Гц)
$f_{C2}$	конечная частота	1 125 004,1425 (Гц)
$T_{3И}$	длительность ЗИ	0,034359 (с)

## Моделирование синтезатора

Для проверки работы синтезатора, расчёта параметров и качества формируемого сигнала в системе Simulink была построена модель. Схема модели представлена на рис.18.

Результаты моделирования (синтезируемый ЛЧМ сигнал и эталонный ЛЧМ сигнал) были записаны в mat-файлы, которые были проанализированы с помощью MATLAB (см. рис.11).

Параметры модели приведены в таблицах

Обозначение	Название	Значение
$f_s$	тактовая частота	$f_s = 125$ (МГц)
$n$	коэффициент деления тактовой частоты для формирования тактовой частоты регистра аккумулятора частоты	$2^2$
$A$	шаг изменения кода частоты	1
$M_1$	код начальной частоты	117441
$M_2$	код конечной частоты	150995
$P$	разрядность регистра аккумулятора частоты	18
$J$	разрядность регистра аккумулятора фазы	24
$D$	разрядность счётчика периодов в ЗИ	17
$number\_periods$	количество периодов в ЗИ	1074
$number\_step$	количество шагов изменения кода частоты	33554

При этом параметры формируемого ЛЧМ сигнала будут следующими

Обозначение	Название	Значение
$f_{c1}$	начальная частота	874 999,9106 (Гц)
$f_{c2}$	конечная частота	1 125 004,1425 (Гц)
$T_{зи}$	длительность ЗИ	0,001074 (с)
$err\_ph\_dlfn$	максимально возможная ошибка по фазе из-за ступенчатого кода частоты $M$	$1,07 \cdot 10^{-5}^\circ$
$err\_ph\_limiting$	максимально возможная ошибка по фазе из-за тактовой частоты $f_s$	$3,24^\circ$

### Ожидаемые ошибки по фазе

При таком методе формирования ЛЧМ сигнала можно выделить 2 источника ошибки:

- 1) ступенчатое задание кода частоты  $M$ , поступающего на вход аккумулятора фазы; Эту составляющую ошибки можно оценить используя выражение

$$err\_ph\_dlfn = \max\{\Delta\theta_s\} = \frac{\pi}{4} \Delta F \cdot \Delta\tau_0 = \frac{\pi}{4} \cdot f_s \cdot \frac{A}{2^J} \cdot \frac{n}{f_s} = \frac{\pi}{4} \cdot \frac{A}{2^J} \cdot n = \frac{\pi}{4} \cdot \frac{n}{2^J}.$$

Т.е. эта ошибка тем меньше, чем больше разрядность аккумулятора фазы  $J$  и меньше шаг изменения кода частоты (т.е. погрешность тем меньше, чем ближе закон изменения кода частоты к линейному закону – меньше проявляется ступенчатый характер).

Эта составляющая будет возрастать при увеличении длительности ЗИ и уменьшение девиации частоты формируемого сигнала.

При  $f_c = 70 \dots 90$  кГц и  $T_{ЗИ} = 100$  мс  $err\_ph\_dlfn$  составит  $0,011^\circ$  (при этом  $err\_ph\_limiting$  составит  $0,26^\circ$ ).

В нашем примере эта ошибка составит  $1,07 \cdot 10^{-5}^\circ$ .

2) тактовая частота фазового аккумулятора.

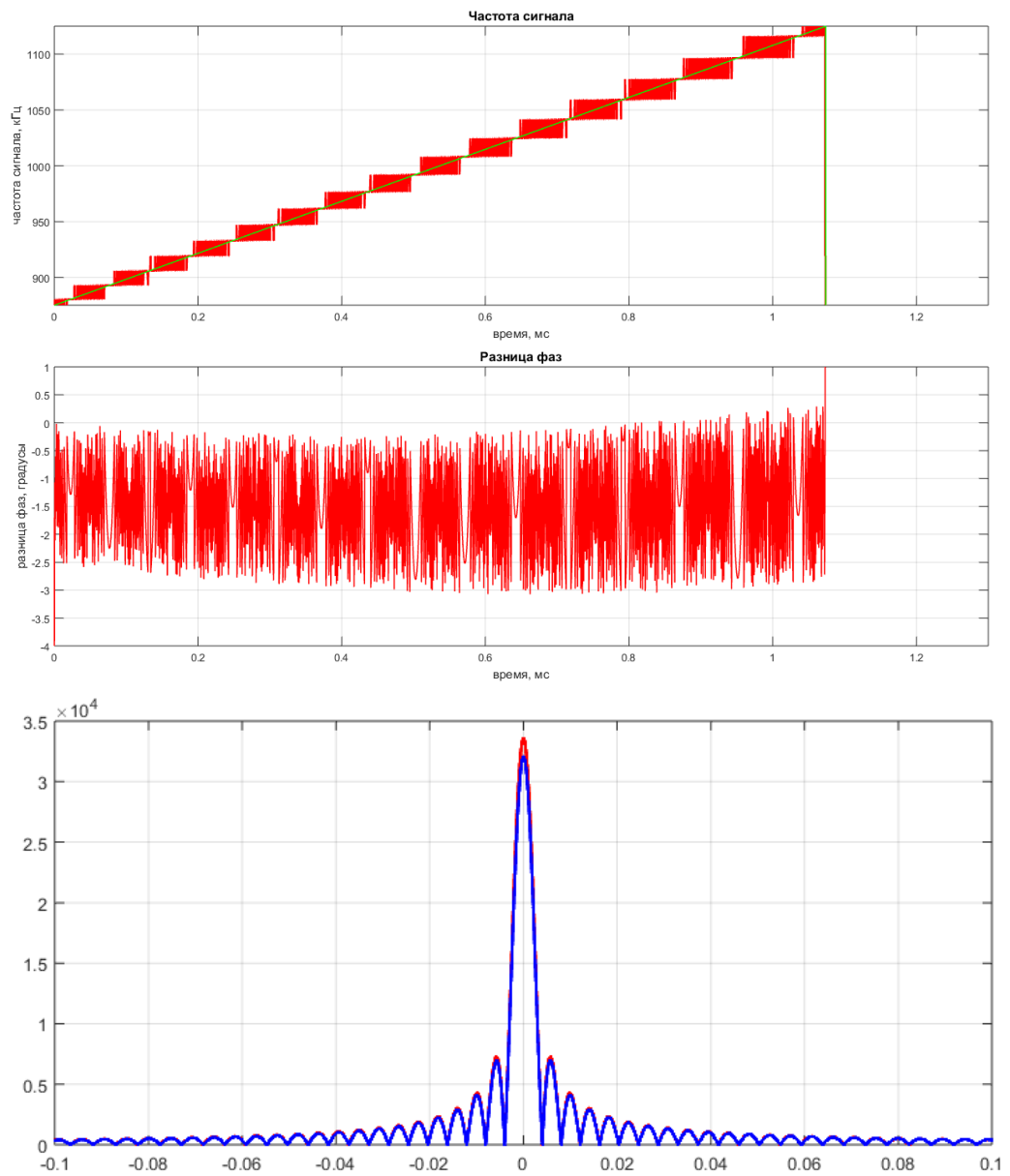
Мы способны генерировать импульсный сигнал период, которого меняется с шагом периода тактового сигнала  $T_s = \frac{1}{f_s}$ . Это вносит погрешность по фазе

$$err\_ph\_limiting = \frac{180}{\pi} \cdot \frac{T_s}{\min\{T_{C1} \ T_{C2}\}} = \frac{180}{\pi} \cdot \frac{\max\{f_{C1} \ f_{C2}\}}{f_s};$$

Очевидно, что эта ошибка тем больше, чем выше частота синтезируемого сигнала  $f_c$  (т.е. эта составляющая ошибки проявляется значительно при формировании сигнала на высоких частотах).

Для уменьшения этой ошибки необходимо увеличить частоту тактирования аккумулятора фазы  $f_s$ .

В нашем примере эта ошибка составит  $3,24^\circ$ .



**Рис.17** Результаты моделирования синтезатора ЛЧМ сигнала на основе метода ПЦС в среде Simulink



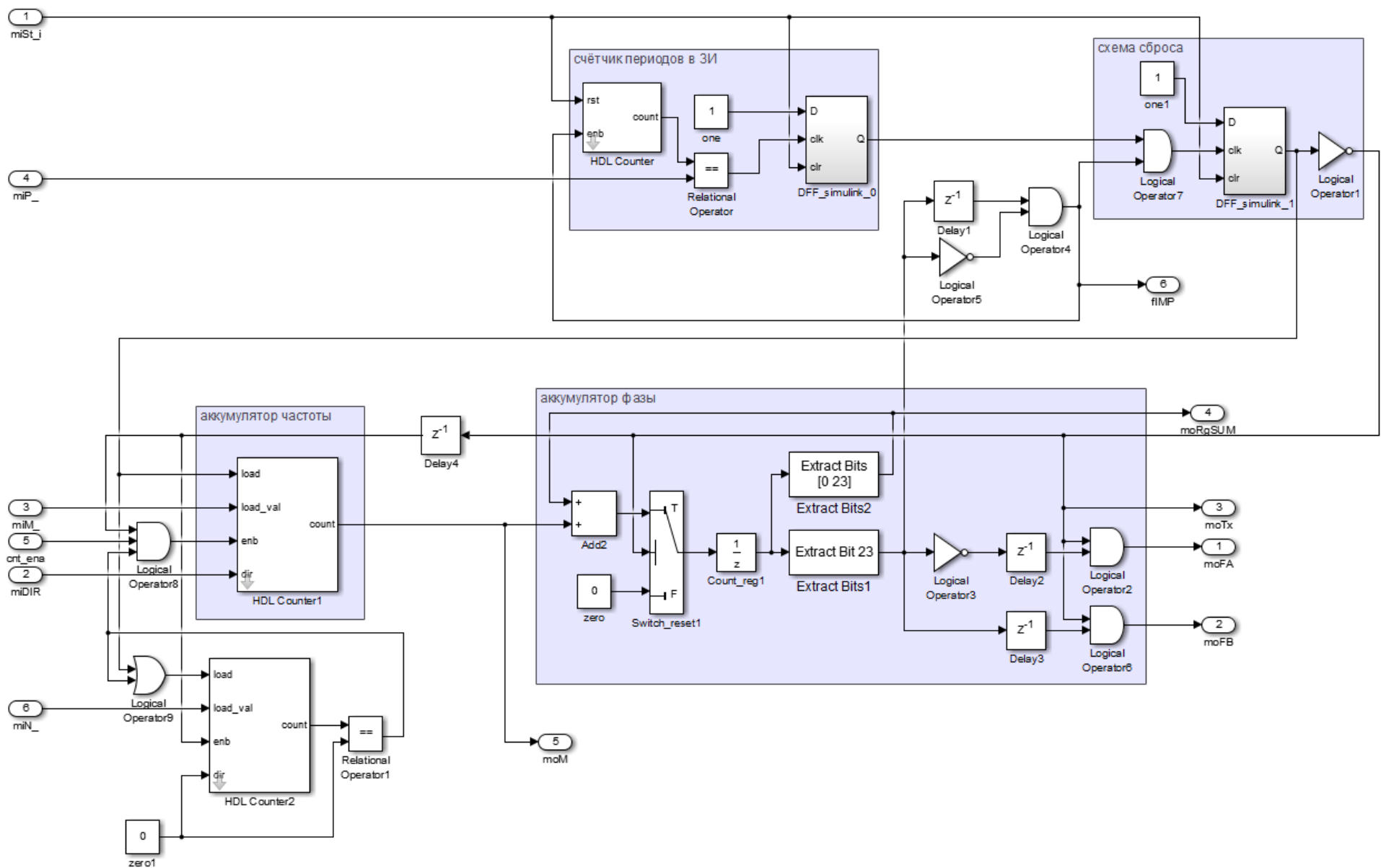


Рис.18 Модель синтезатора ЛЧМ сигнала на основе метода ПЦС в среде Simulink

## ООО “Экран”

### Листинг скрипта расчёта параметров синтезатора ЛЧМ сигнала на основе метода ПЦС написанный в среде MATLAB

```
clc
clear

%% исходные данные

% const - определяются схемой
frequency_sample = 125*10^6; % частота синхронизации // double
J = 24; % разрядность аккумулятора фазы // uint8
P = 18; % разрядность аккумулятора частоты // uint8
D = 19; % разрядность счётчика периодов в ЗИ // uint8
N = 12; % разрядность делителя частоты аккумулятора частоты // uint8

% var - задаёт пользователь
fc1 = 120*10^3; % начальная частота // double
fc2 = 250*10^3; % конечная частота // double
Tzp_u = 0.4*10^-3; % длительность зондирующего импульса (ЗИ) // double

Time_R = 14; % параметр определяющий частоту дискретизации // uint8

%% расчёт параметров синтезатора ЗИ

%ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ АККУМУЛЯТОРА ЧАСТОТЫ
if fc1 == fc2
    cnt_ena = 0; % частоты не меняется (тональный сигнал) // boolean
    DIR = 1; % направление изменения кода частоты // boolean
    A = 0; % шаг изменения кода частоты // int8
elseif fc1 < fc2
    cnt_ena = 1;
    DIR = 1; % (увеличение)
    A = 1; % шаг изменения кода частоты
else
    cnt_ena = 1;
    DIR = 0; % (уменьшение)
    A = -1; % шаг изменения кода частоты
end

% коды крайних частот
M1 = round(2^J*fc1/frequency_sample+0.5*A); % код начальной частоты // uint18
if M1 > 2^P
    disp(['fc1 слишком большая (fc1 должна быть меньше '...
        num2str((2^P-0.5*A)*frequency_sample/2^J)...
        ' Гц)']);
end

M2 = round(2^J*fc2/frequency_sample-0.5*A); % код конечной частоты // uint18
if M2 > 2^P
    disp(['fc1 слишком большая (fc2 должна быть меньше '...
        num2str((2^P-0.5*A)*frequency_sample/2^J)...
        ' Гц)']);
end

% количество периодов изменения кода частоты // uint18
num_step = abs(M2-M1);
```

```

% требуемый коэффициент деления тактовой частоты (для обеспечения заданной длительности ЗИ) //
uint12
n = round(frequency_sample*Tzp_u/num_step);
if n < 1
    n = 1;
    disp('ВНИМАНИЕ!!! Слишком большая скорость изменения частоты');
end
if n > 2^N
    n = 2^N;
    disp('ВНИМАНИЕ!!! Слишком малая скорость изменения частоты');
end

% крайнии частоты синтезируемого сигнала // double
fc1_c = (M1-0.5*A)*frequency_sample/2^J;
fc2_c = (M2+0.5*A)*frequency_sample/2^J;

Tzp = n*num_step/frequency_sample;

% количество периодов сигнала // uint19
num_periods = ceil(((fc1_c + fc2_c)/2)*Tzp);
if num_periods > 2^D
    num_periods = 2^D;
    disp('ВНИМАНИЕ!!! Слишком большая длительность ЗИ');
end

disp(' ');
disp('-----');
disp('Параметры синтезатора:');
disp('-----');
disp(['frequency_sample = ', num2str(frequency_sample)]);
disp(['      J = ', num2str(J)]);
disp(['      n = ', num2str(n)]);
disp(['      M1 = ', num2str(M1)]);
disp(['      M2 = ', num2str(M2)]);
disp(['      DIR = ', num2str(DIR)]);
disp(['      num_step = ', num2str(num_step)]);
disp(['      num_periods = ', num2str(num_periods)]);
disp('-----');
disp(' ');

%% формируем образ ЗИ

% скорость изменения частоты // double
Klfm = (fc2_c - fc1_c)/Tzp;

% центральная частота // double
f0 = (fc1_c + fc2_c)/2;

% количество "тиков" в ЗИ // uint32
num_point = Tzp*frequency_sample;
CODE = num_point*(M1 + M2)/2; % сумма арифметической прогрессии // uint32
num_point = num_point + ceil((((ceil(CODE/2^J) - (CODE/2^J))*2^J)/M2);

% длительность ЗИ // double
Tzp = num_point/frequency_sample;

% длительность одного "тика" // double

```

```

dtime = 1/frequency_sample;

% массив времени // double
time = (0:dtime:num_point*dtime);

% ЗИ // double (complex)
signal = ...
    exp(complex(0,2*pi*fc1_c*time + ...
        pi*Klfm*time.^2));

disp('-----');
disp('Параметры ЗИ:');
disp('-----');
disp(['начальная частота = ', num2str(fc1_c), ' Гц']);
disp(['конечная частота = ', num2str(fc2_c), ' Гц']);
disp(['длительность ЗИ = ', num2str(Tzp), ' с']);
disp('-----');

disp(['частота дискретизации = ', num2str(frequency_sample/Time_R), ' Гц']);
disp('-----');

% индексы выборки сигнала (берется значение через Time_R) // double (complex)
index_sample = (1:Time_R:num_point);
image_signal = signal(index_sample);

figure(1)
plot(time*10^3,imag(signal),'Color','red','LineWidth',2)
hold on
plot(time(index_sample)*10^3,imag(image_signal),...
    'Color','black','LineWidth',1,'LineStyle','none','Marker','o')
hold off
grid on
xlabel('время, мс');

```

## 1.8. Метод прямого цифрового синтеза (генератор GDDS5, версия \_v5)

Предназначен для формирования сложных сигналов в широком диапазоне частот.

Генератор содержит (Рисунок 17):

- Делитель DIV (может отсутствовать в зависимости от версии реализации)
- ОЗУ RAM\_DDS
- Синхронизатор
- Схема контроля мертвого времени

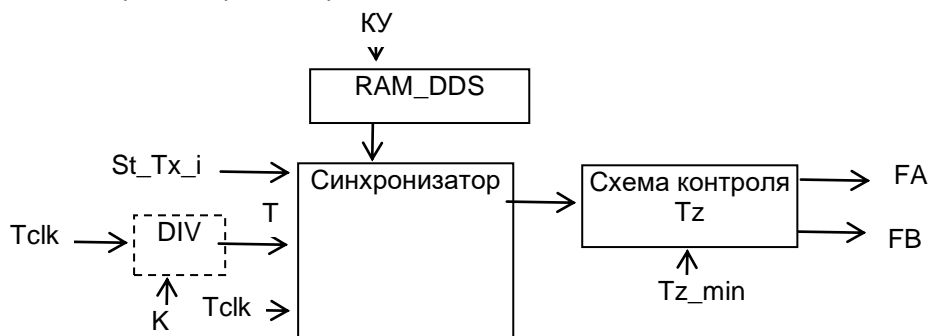


Рисунок 17. Структура генератора

Делитель DIV обеспечивает формирование опорного тика Т. Длительность всех интервалов фаз А и В всегда кратна длительности тика Т. При отсутствии делителя  $T=Tclk$ .

ОЗУ RAM\_DDS хранит КУ и может иметь различную организацию - Таблица 8 (в зависимости от версии прошивки).

Таблица 8 – Используемые ресурсы памяти в зависимости от организации RAM\_DDS

Кол-во используемых ЕАВ 9К	Организация	Кол-во КУ (макс. кол-во формируемых полупериодов)
1	1024x9 (2 страницы 512x9)	512 (255)
2	2048x9 (2 страницы 1024x9)	1024 (511)
4	4096x9 (2 страницы 2048x9)	2048 (1023)
8	8192x9 (2 страницы 4096x9)	4096 (2047)

Схема контроля контролирует время между сигналами возбуждения фаз А и В. Схема гарантирует, что между соседними импульсами возбуждения фаз всегда будет минимальный защитный интервал.

### 1.8.1. Логика работы и упаковка данных

Обеспечивается формирование одного или нескольких полупериодов с заданной длительностью и возможностью использования ШИМ внутри полупериода или периода. Также возможно произвольное удлинение длительности интервалов возбуждения или интервалов отсутствия возбуждения.

Параметры для работы, задаваемые ВС:

К – коэффициент деления делителя (если делитель предусмотрен)

КУ – команды КУ, записываемые в RAM\_DDS

Параметр К рассчитывается по формуле:

$$K = \text{ЦЕЛОЕ} (T/T_{clk}) - 1$$

где Т – требуемое время тика Т, с

Tclk – период системной синхрочастоты Tclk, с

Минимальное значение K = 0 (T=Tclk)

Максимальное значение K = 255 (T=256\*Tclk)

Минимальный шаг изменения K= 1.

Каждая КУ позволяет сформировать один уникальный временной интервал (длительность интервала кратна Т). Четная КУ определяет интервал отсутствия возбуждения по обеим фазам. В КУ используются 9 бит для задания длительности интервала. Нечетная КУ определяет интервал возбуждения одной из фаз. В КУ используются 1 бит для задания фазы и 8 бит для задания длительности интервала (Таблица 9). В общем случае, для формирования одного полупериода требуется 2 КУ.

**Таблица 9 – Формат упаковки КУ в RAM\_DDS**

Команда (адрес в ОЗУ)	Биты	
	8	7-0
CC0 (+0)	Time0[1]	
CC1 (+1)	Faze[2]	Time1[2]
CC2 (+2)	Time0[3]	
CC3 (+2)	Faze[4]	Time1[4]
...	.....	...
CCn-2 (+n-2)	Time0[n-1]	
CCn-1 (+n-1)	Faze[n]	Time1[n]

Параметр Time0[i] в КУ определяет длительность интервала [i] отсутствия возбуждения одной из фаз. Значение параметра Time0[i] рассчитывается по формуле:

$$\text{Time0}[i] = \text{БЛИЖАЙШЕЕ\_ЦЕЛОЕ} (T0[i]/T) - 2$$

где:

T0[i] – требуемое время интервала i, с

T – длительность тика Т, с

Если Time0[i]=0, это означает отсутствие интервала (пустая КУ). Используется для удлинения времени возбуждения предыдущего интервала за счет времени возбуждения последующего интервала.

Если Time0[i]=511 (код КУ = 1FFh), это означает окончание формирования ЗИ (пустая КУ). После окончания текущего интервала [i-1] формирование ЗИ будет остановлено.

Минимальное значение Time0[i] =3 (Ti=5\*T)

Максимальное значение Time0[i] = 510 (Ti=512\*T)

Минимальный шаг изменения Time0[i] = 1.

**ПРИМЕЧАНИЕ.** Time0[i] используется только для четных КУ (адреса +0,+2, ...).

Параметр Faze[i] определяет формируемую фазу для интервала i возбуждения одной из фаз (нечетные интервалы):

=0 – фаза А

=1 – фаза В

**ПРИМЕЧАНИЕ.** Faze[i] используется только для нечетных КУ (адреса +1,+3, ...).

Параметр Time1[i] в КУ определяет длительность интервала I возбуждения одной из фаз. Значение параметра Time1[i] рассчитывается по формуле:

$$\text{Time1}[i] = \text{БЛИЖАЙШЕЕ\_ЦЕЛОЕ} (T1[i]/T) - 2$$

где:

T1[i] – требуемое время интервала i, с

T – длительность тика Т, с

Если  $\text{Time1}[i]=0$ , это означает отсутствие интервала (пустая команда). Используется для удлинения времени отсутствия возбуждения предыдущего интервала за счет времени отсутствия возбуждения последующего интервала.

Минимальное значение  $\text{Time1}[i] = 3$  ( $T_i=5 \cdot T$ ).

Максимальное значение  $\text{Time1}[i] = 254$  ( $T_i=256 \cdot T$ ).

Минимальный шаг изменения  $\text{Time1}[i] = 1$ .

Если  $\text{Faze}[i]=1$  и  $\text{Time1}[i]=255$  (т.е. код КУ = 1FFh), это означает окончание формирования ЗИ (пустая КУ). После окончания текущего интервала  $[i-1]$  формирование ЗИ будет остановлено.

**ПРИМЕЧАНИЕ.**  $\text{Time0}[i]$  используется только для нечетных КУ (адреса +1,+3, ...).

#### Ограничения:

- СС0 не может использоваться в качестве КУ окончания ЗИ (для СС0  $\text{Time0}[1] > 0$ )
- не допускается наличие подряд двух пустых КУ
- обязательно наличие КУ окончания формирования ЗИ (может находиться в четной или нечетной КУ)

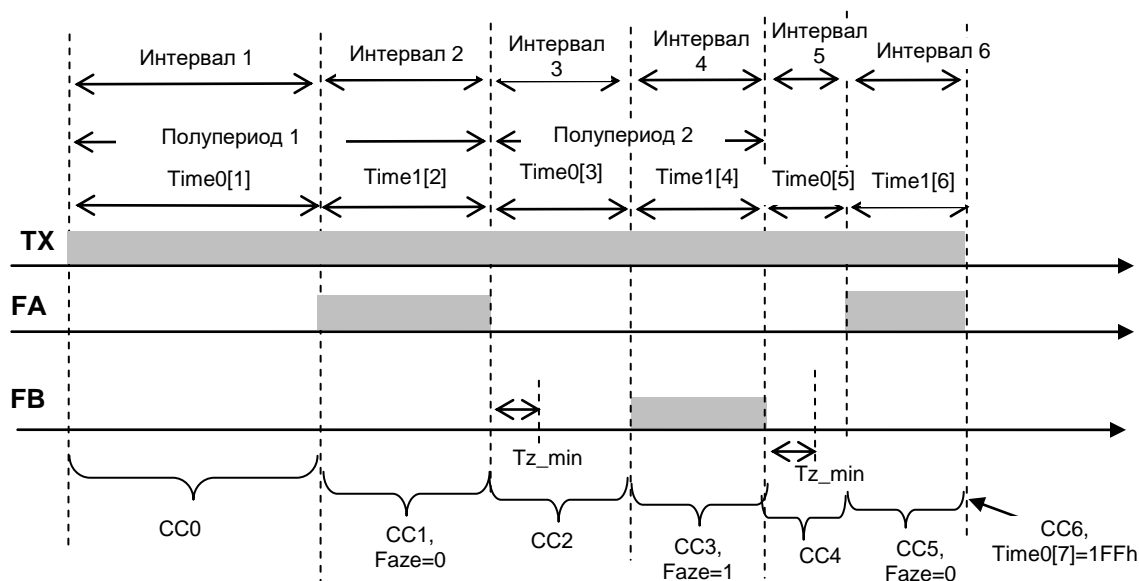
Мертвое время контролируется минимальным значением, зависящим от аппаратуры, для:

- четной КУ, если для данной КУ  $\text{Time0}[i] > 0$

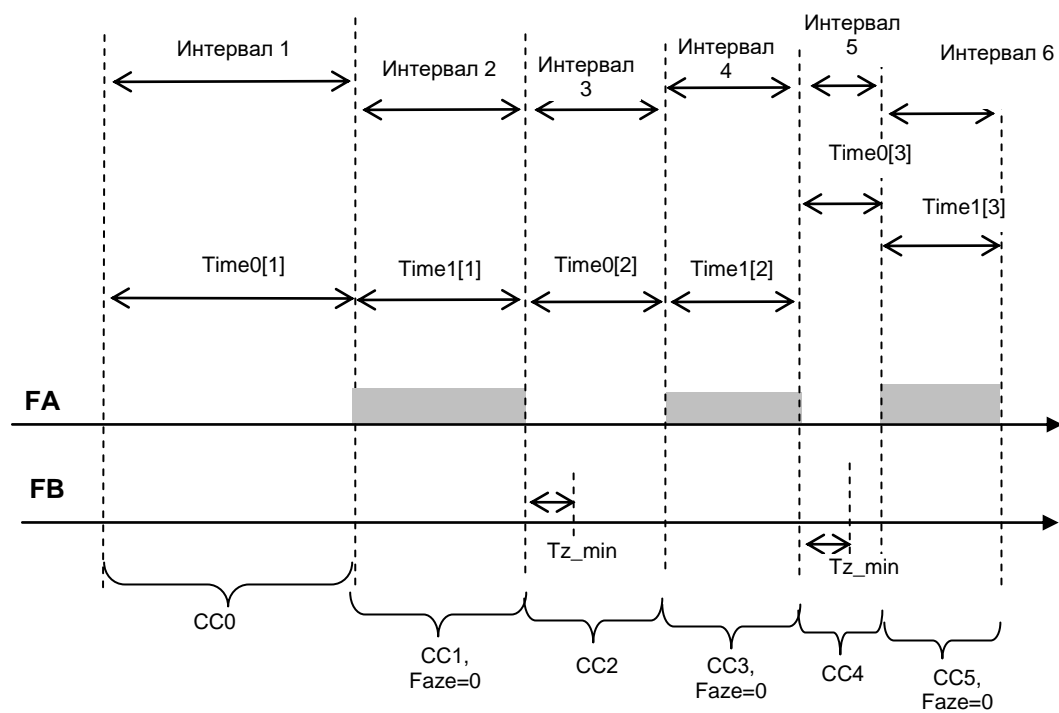
Параметр  $Tz\_min$  определяет минимальное время  $Tz$  в тиках  $T_{clk}$ , является внутренней константой и зависит от конкретной прошивки.

При необходимости увеличения длительности интервала одной из фаз между нечетными КУ вставляется КУ с  $\text{Time0}[i]=0$ , что приводит к объединению фаз (Рисунок 21).

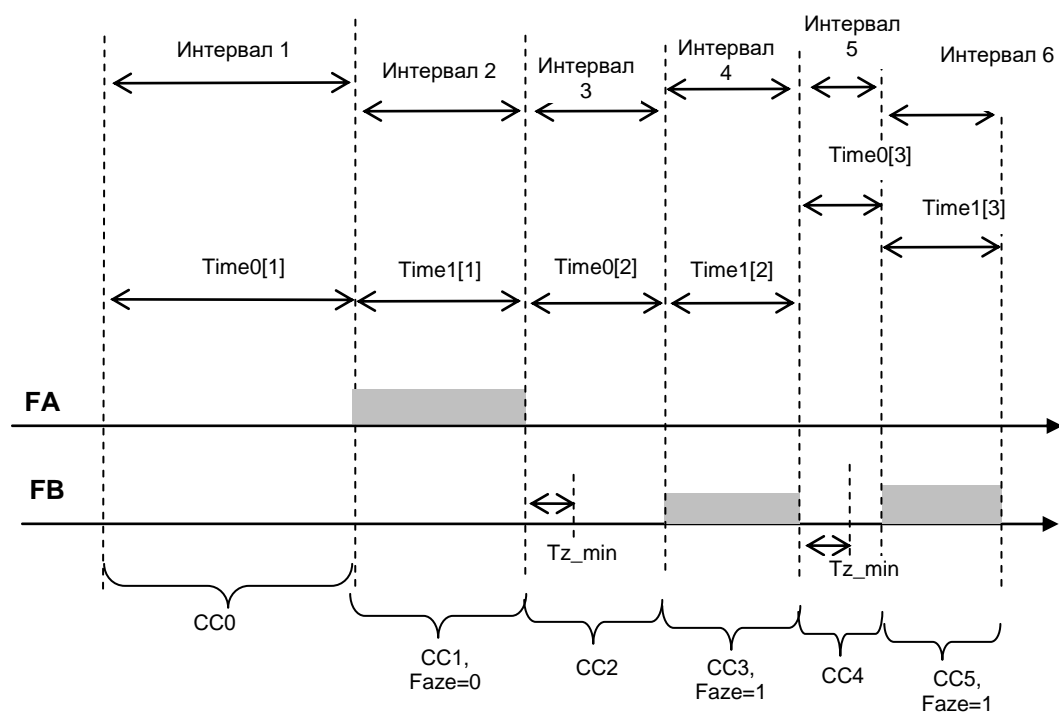
При необходимости увеличения длительности интервала отсутствия возбуждения между четными КУ вставляется КУ с  $\text{Time1}[i]=0$ , что приводит к объединению двух интервалов отсутствия возбуждения (Рисунок 22).



**Рисунок 18. Пример формирование фаз (3 полупериода)**

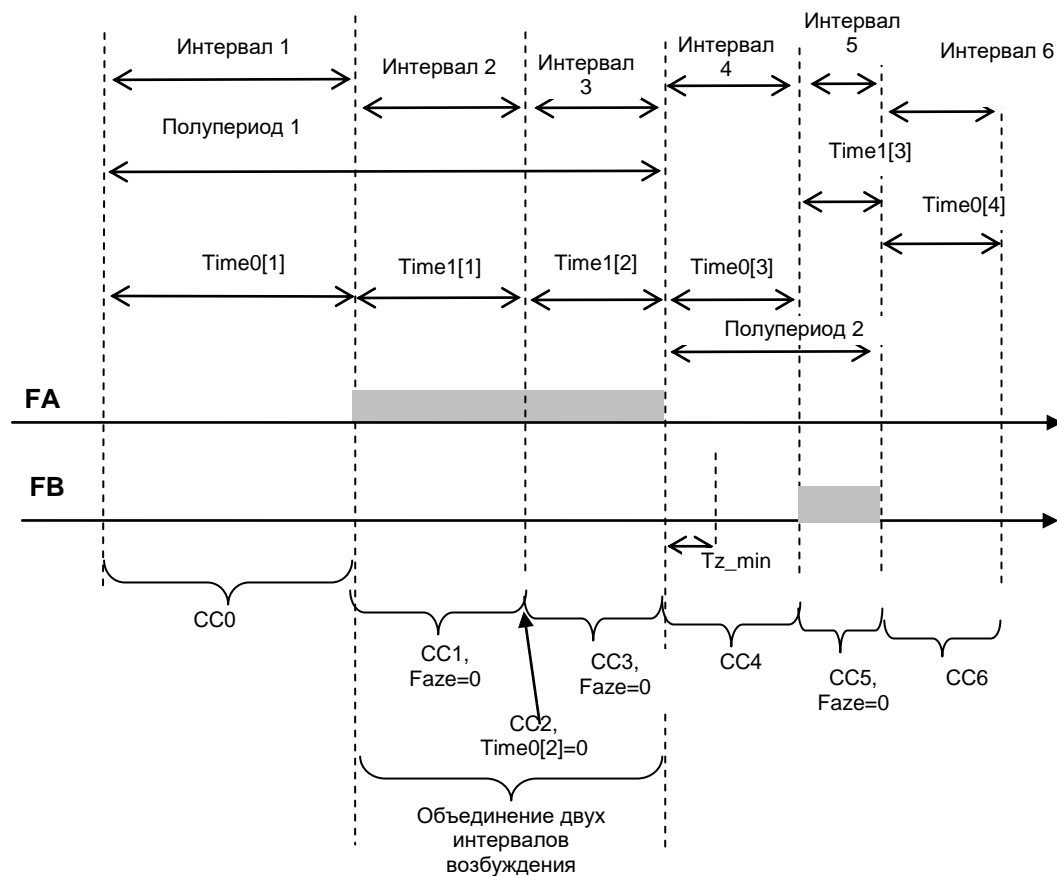


**Рисунок 19. Пример формирование фаз (ШИМ по фазе А)**

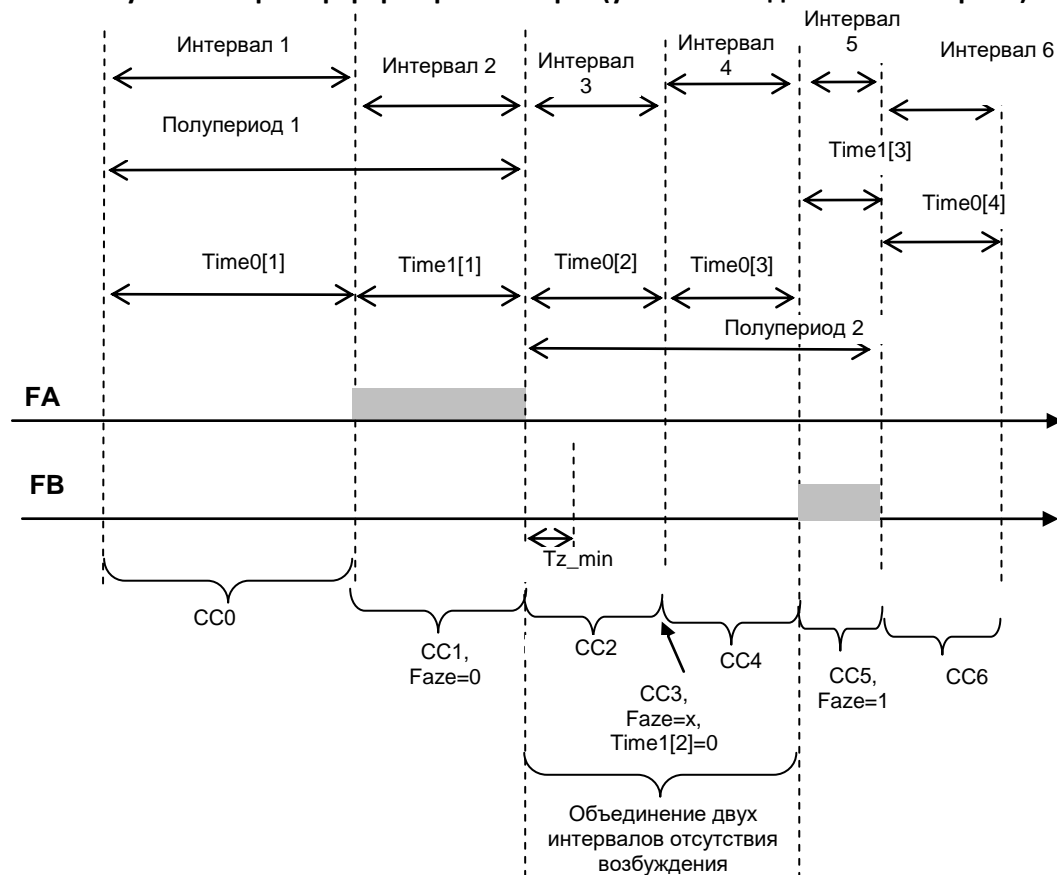


**Рисунок 20. Пример формирование фаз**





**Рисунок 21. Пример формирования фаз (увеличение длительности фазы)**



**Рисунок 22. Пример формирование фаз (увеличение длительности интервала отсутствия возбуждения)**

## 2. Файл данных DDS

Файл предназначен для хранения данных, описывающих параметры и КУ для формирования ЗИ методом DDS в PGL.

Данные DDS позволяют полностью определить параметры ЗИ и параметры накопления ГЛИ.

DDS файл является самодостаточным и может использоваться самостоятельно, без PPI файла. PPI файл используется до тех пор, пока ПО в ВС не будет самостоятельно рассчитывать образ ЗИ по данным из файла DDS. Данная версия DDS файла содержит все необходимые данные для расчета образа ЗИ и позволяет при отказе от использования PPI файла не менять версию файла DDS.

Файл создается с помощью специальной программы формирования файлов DDS и PPI.

Файл имеет следующий формат имени:

**Code[\_Name].dds,**

где:

Code – уникальный код ЗИ = {00001..65534} – всегда 5 символов (обязательное поле)

Name – Краткое название ЗИ (только цифры и английские буквы) – необязательное поле

dds – расширение имени файла

Рекомендуемая последовательность краткого описания параметров ЗИ в имени: тип, начальная частота, конечная частота (для ЛЧМ), длительность, мертвое время, версия описания.

ПРИМЕРЫ:

00001\_Tone\_248kHz\_10mks\_z40\_v1.dds (код ЗИ = 1, тон, 248025 Гц, длительность 10 мс, мертвое время 40 нс, версия 1)

01001\_LFM\_248kHz\_270kHz\_32ms\_z40\_v255.dds (код ЗИ = 1001, ЛЧМ, начальная частота 248000 Гц, конечная частота 270 кГц (растущий ЛЧМ), длительность 32 мс, мертвое время 40 нс, версия 255).

**ПРИМЕЧАНИЕ.** Поиск необходимого файла DDS по коду ЗИ осуществляется с помощью поиска файла, первые пять символов (поле Code) в имени которого соответствуют искомому коду ЗИ. Оставшаяся часть имени файла (поле Name) не учитывается. Для дополнительного контроля в найденном файле необходимо считать поле gPRM\_ZI.CodeZI, которое должно совпадать с искомым кодом ЗИ.

Файл является двоичным (бинарным) и состоит из одной записи (DDS\_REC), содержащей заголовок записи (DDS\_HDR) и данных DDS (DDS\_DATA) следующих в файле один за другим без разрыва (Рисунок 1). Формат заголовка описан в файле Гидра4\_Форматы файлов\_x.doc (код типа данных записи в заголовке = 0201h – признак файла DDS).

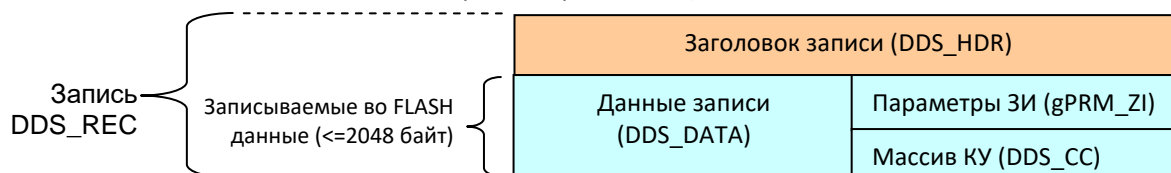


Рисунок 23. Формат файла DDS

Данные записи DDS\_DATA содержат:

- Параметры ЗИ (PRM\_ZI) – структура gPRM\_ZI (Таблица 10)

- Массив КУ ( DDS\_CC) – в зависимости от используемой упаковки КУ (см. приложение)

В зависимости от используемого ГЗИ, массив КУ хранит либо КУ для генератора DDS, DDS5, либо параметры генератора GCHIRP (Приложение А).

Размер файла определяется суммой размеров заголовка записи, параметров ЗИ и массива КУ. Массив КУ следует непосредственно за параметрами ЗИ. Максимальный размер массива КУ при загрузке во FLASH КОНГ не должен превышать  $2048-144 = 1904$  байта (952 слова).

Версия файла DDS задается в поле Version заголовка записи (см. документ Гидра4\_Форматы файлов\_xx.doc).

// Описание структуры gPRM\_ZI

struct gPRM\_ZI // Размер структуры = 16+128 = 144 байта

```
{
    U16 CodeZI;           // +0 (2 байта) Код ЗИ
    U8 Mode;              // +2 ( 1 байт)
    U8 Mode1;             // +3 (1 байт) - Резерв = 0
    U16 TimeZ;            // +4 ( 2 байта)
    U32 NumPrd;           // +6 ( 4 байта )
    U16 TimeR;            // +10 (2 байта)
    U16 TimeI;            // +12 (2 байта)
    U16 CRC;              // +14 (2 байта) – Всего 16 байт
    U32 Prm0;             // +16 (4 байта)
    U32 Prm1;             // +20 (4 байта)
    U8 Ver;               // +24 (1 байт)
    U8 Type;              // +25 (1 байт)
    Wchar Name_RU[ 20 ];  // +26 (40 байт) – строка UNICODE – до 19 символов
    Wchar Name_EN[ 20 ];  // +66 (40 байт) – строка UNICODE – до 19 символов
    Double StartFreq;      // +106 (8 байт)
    Double EndFreq;        // +114 (8 байт)
    Double TimeZI;         // +122 (8 байт)
    Double Fs;             // +130 (8 байт)
    U8 SampleType;         // +138 (1 байт)
    U8 Rzv2;               // +139 (1 байт) - резерв = 0
    U16 NumCC;             // +140 (2 байт)
    U8 Rzv3[ 2 ];         // +142 (2 байт) - резерв = 0
}
```

**Таблица 10 – Формат параметров ЗИ (структура gPRM\_ZI)**

Смещение, байт / размер в байтах	Название	Описание	Примечание
+00 / 2	CodeZI	Код ЗИ – уникальный номер ЗИ = { 1..65534 }, =0 или =65535 – нет данных DDS. Должен совпадать с полем Code в имени файла	Используется PGL при формировании сигналов в качестве кода ЗИ
+01 / 1	Mode	Биты 7..3 - резерв, =0 Биты 2..0 - режим работы генератора (формат упаковки КУ): =0 – режим ШИМ0 (CC[0] задает начальную длительность, остальные КУ – приращения и повторы) =1 - режим ШИМ1 =2 - резерв =3 – режим ШИМ2 – файлы _v3 =4 – режим GCHIRP (см. 1.7) – файлы _v4 =5 – режим DDS5 – файлы _v5 =6..7 – резерв	
+02 / 1	Mode1	Биты 7..1 = 0 Бит 0 - признак работы ЦФД: = 0 - ЦФД не используется (выключен) =1 – ЦФД используется (включен)	Используется PGL для включения/выключения ЦФД и режима формирования ЗИ
+04 / 2	TimeZ	Длительность мертвого времени между фазами А и В, тик (для КУ в режиме 0, для остальных режимов не используется)	Используется PGL при формировании сигналов
+06 / 4	NumPrd	Кол-во формируемых периодов (для КУ в режиме 0, для остальных режимов не используется)	Используется PGL при формировании сигналов
+10 / 2	TimeR	Период отсчетов R в тиках	Используется PGL при накоплении
+12 / 2	TimeI	<b>Интервал отсчета I в тиках (используется для всех исполнений кроме ППФ4)</b>	Используется PGL при накоплении
+14 / 2	CRC	Значение CRC для всех байт PRM_ZI (по смещениям 0..+13) и DDS_CC за исключением слова CRC	Если = 0 или 0xFFFF, CRC не используется
+16 / 4	Prm0	Резерв=0	
+20 / 4	Prm1	Резерв=0	
+24 / 1	Ver	Версия описания DDS = {1..254}	
+25 / 1	Type	Тип ЗИ: 0 – ТОН, 1 – ЛЧМ, 2 – ФКМ, 3 – ТОН_АМ 4..255 – резерв	
+26 / 40	Name_RU	Название ЗИ (RU) Если не используется, первым символом в строке должен быть символ конца строки.	Необязательный параметр
+3Ah / 40	Name_EN	Название ЗИ (ENG) Если не используется, первым символом в строке должен быть символ конца строки.	Необязательный параметр
+6Ah / 8	StartFreq	Начальная частота Fc, Гц	
+72h / 8	EndFreq	Конечная частота Fc, Гц	Для тона EndFreq = StartFreq
+7Ah / 8	TimeZI	Длительность ЗИ, с	

+82h / 8	Fs	Частота Fs, Гц	Рисунок 4
+138 / 1	SampleType	Тип отсчетов образца: 0 – комплексные, 1 – действительные 2 – действительные, амплитуда 3..255 - резерв	
+139 / 1	Rzv2	Параметры УМ (для H5p1): Скважность мощности УМ в %: Значение 1..100 отражает мощность от 1 до 100 %.	Если 0, то скважность = 100% (для совместимости со старыми ЗИ) Используется только для H5p1
+140 / 2	NumCC	Количество КУ в массиве DDS_CC Для GCHIRP или DDS5: кол-во байт в массиве КУ.	Для DDS5 кол-во используемых КУ = NumCC/4
+142 / 2	Rzv3	Резерв = 0	
<p>ПРИМЕЧАНИЯ.</p> <p>Серым цветом выделены поля, используемые программой PGL для формирования сигналов и накопления ГЛИ (в этих полях хранятся параметры ГЗИ и модулей накопления)</p> <p>Параметр Rzv2 используется для установки напряжения питания УМ в H5p1</p>			

### 3. Файл образа ЗИ (PPI файл)

Файл предназначен для хранения данных, описывающих параметры одного ЗИ. Данные позволяют полностью определить параметры ЗИ и параметры накапливаемых данных ГЛИ.

Каждый ЗИ имеет набор уникальных параметров и соответственно – уникальный код, позволяющий однозначно определить данный ЗИ. При изменении любого из параметров ЗИ меняется его код.

Каждому файлу образа ЗИ соответствует свой файл данных DDS, который содержит данные для формирования этого ЗИ методом DDS в PGL.

Файл является двоичным (бинарным) файлом и создается с помощью специальной программы.

#### 3.1. Формат имени PPI файла

Файл имеет следующий формат имени:

**Code[\_Name].ppi,**

где:

поле Code – уникальный код ЗИ = {00001..65534} – 5 символов – обязательное поле

поле Name – Краткое название ЗИ (только цифры и английские буквы) – необязательное поле

ПРИМЕРЫ:

00001\_Tone248000\_10mks\_z40\_v1.ppi

12345\_v1.ppi

#### 3.2. Формат данных PPI файла

Файл состоит из одной записи (PPI\_REC), содержащей заголовок записи (PPI\_HDR) и данных PPI (PPI\_DATA) следующих в файле один за другим без разрыва (Рисунок 24).

Формат заголовка описан в файле Гидра4\_Форматы файлов\_x.doc (код типа данных записи в заголовке = 0200h – признак файла PPI).

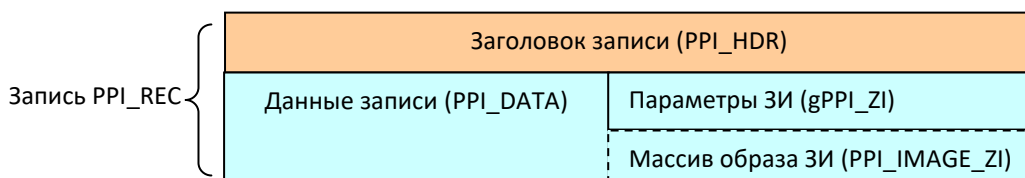


Рисунок 24. Формат файла PPI

PPI\_DATA содержат:

- Параметры ЗИ (PPI\_ZI) – структура gPPI\_ZI (см. 3.2.1)
- Массив выборок образа ЗИ – PPI\_IMAGE\_ZI (для сложных сигналов, требующих свертки, для тона не используется) – см. 3.2.2.

ПРИМЕЧАНИЕ. Массив выборок образа ЗИ может отсутствовать для простых сигналов (тон).

Размер файла определяется суммой размеров заголовка записи, параметров ЗИ и массива образа ЗИ.

ПРИМЕЧАНИЕ. Структура gPPI\_ZI имеет два варианта упаковки (версии 0,1 - вариант 1 и версия 2 – вариант 2). Версия упаковки 2 совместима с версиями упаковки 0,1 (первые 164 байта), в версии упаковки 2 размер структуры увеличен до 256 байт (добавлены новые поля размером 92 байта в конец структуры).

Версия упаковки находится в поле gPPI\_ZI.PacVer;

Упаковка версии 2 введена с момента поддержки H5p1.

### 3.2.1. Формат gPPI\_ZI

```
struct gPPI_ZI // Структура описания параметров ЗИ файла PPI
// версия упаковки файла .*ppi = 0 или 1
// размер структуры = 164 байт
{
    u16 Code;           // +0 (2 байта) – Код ЗИ
    u32 Prm0;           // +2( 4 байта)
    u32 Prm1;           // +6 (4 байта)
    u8 PacVer;          // +10 (1 байт) =0 или 1
    u8 Type;            // +11 (1 байт)
    Wchar Name_RU[ 20 ]; // +12 (40 байт) – строка UNICODE
    Wchar Name_EN[ 20 ]; // +52 (40 байт) – строка UNICODE
    double TickSize;     // +92 (8 байт)
    double StartFreq;    // +100 (8 байт)
    double EndFreq;      // +108 (8 байт)
    double TimeZl;       // +116 (8 байт)
    double TimeZ;        // +124 (8 байт)
    u32 NumImpFA;        // +132 (4 байт)
    u32 NumImpFB;        // +136 (4 байт)
    double Fs;           // +140 (8 байт)
    double Fim;          // +148 (8 байт)
    u32 NumSampleImage;  // +156 (4 байт)
    u8 SampleType;       // +160 (1 байт)
    u8 Prm2;             // +161 (1 байт)
    u8 Prm3;             // +162 (2 байт)
    u8 Prm4;             // +163 (1 байт)
}
```

```
// Структура описания параметров ЗИ файла PPI
// версия упаковки =2 (файла .*ppi = 2 (для H5p1)
// размер структуры = 256 байт
struct gPPI_ZI
{
    u16 Code;           // +0 (2 байта) – Код ЗИ
    u32 Prm0;           // +2( 4 байта)
    u32 Prm1;           // +6 (4 байта)
    u8 PacVer;          // +10 (1 байт)
    u8 Type;            // +11 (1 байт)
    Wchar Name_RU[ 20 ]; // +12 (40 байт) – строка UNICODE
    Wchar Name_EN[ 20 ]; // +52 (40 байт) – строка UNICODE
    double TickSize;     // +92 (8 байт)
    double F1_StartFreq; // +100 (8 байт)
    double F1_EndFreq;   // +108 (8 байт)
    double TimeZl;       // +116 (8 байт)
    double TimeZ;        // +124 (8 байт)
    U32 NumImpFA;        // +132 (4 байт)
    U32 NumImpFB;        // +136 (4 байт)
    double Fs;           // +140 (8 байт)
    double Fim;          // +148 (8 байт)
    u32 NumSampleImage;  // +156 (4 байт)
    u8 SampleType;       // +160 (1 байт)
    u8 Prm2;             // +161 (1 байт)
    u8 Prm3;             // +162 (2 байт)
    u8 Prm4;             // +163 (1 байт)
    //Вновь добавленные поля для версии 2
    double F2_StartFreq; // +164 (8 байт)
```

```

double F2_EndFreq;    // +172 (8 байт)
Wchar Rzv[ 76 ];     // +180 (76 байт)
}

```

**ПРИМЕЧАНИЕ.** Первые 164 байта структуры gPPI\_ZI версии 2 аналогичны упаковке байта структуры gPPI\_ZI версии 0 и 1 (Таблица 11). Описание вновь введенных полей для версии 2 - .

Таблица 11 – Формат параметров ЗИ версии 0,1 или 2 (структура gPPI_ZI)			
Смещение, байт / размер, байт	Название	Описание	Примечание
+0 / 2	Code	Код ЗИ = { 1..65534 }	Должен совпадать с полем Code в имени файла
+2 / 4	Prm0	Бит 31 - признак ЦФД: =0 – не используется, =1 – используется Бит 30 - признак ШИМ: =0 – не используется, =1 – используется	Биты 29..0 – резерв = 0
+6 / 4	Prm1	Если Type=2: Кол-во периодов несущей в одном бите (1..65535)	
+10 / 1	PacVer	Версия упаковки файла = 0,1 или 2	версия упаковки файла .*ppi
+11 / 1	Type	Тип ЗИ: 0 – ТОН, 1 – ЛЧМ, 2 – ФКМ, 3 – ТОН_AM 4..255 – резерв	Дополнительные параметры ЗИ в полях Prm2-Prm4
+12 / 40	Name_RU	Название ЗИ (RU) – до 19 символов UNICODE + 1 символ конца строки (0000h)	Необязательный параметр. Если не используется, первым символом в строке должен быть символ конца строки.
+52 / 40	Name_EN	Название ЗИ (ENG). Аналогично Name_RU	
+92 / 8	TickSize	Длительность одного опорного периода Tclk (нс) : для 100 МГц = 10 нс, для 300 МГц = 3,33(3) нс	Tclk (Рисунок 2, Рисунок 4)
+100 / 8	StartFreq	Начальная частота несущей Fc, Гц Или принимаемая начальная разностная частота для ТОН_AM	Для тона и ФКМ StartFreq = EndFreq
+108 / 8	EndFreq	Конечная частота несущей Fc, Гц Или принимаемая конечная разностная частота для ТОН_AM	
+116 / 8	TimeZl	Длительность ЗИ, с	
+124 / 8	TimeZ	Длительность мертвого времени между фазами А и В, нс	
+132 / 4	NumImpFA	Кол-во импульсов фазы А в ЗИ	
+136 / 4	NumImpFB	Кол-во импульсов фазы В в ЗИ	
+140 / 8	Fs	Частота Fs, Гц	Рисунок 4, Рисунок 5
+148 / 8	Fim	Частота взятия отсчетов Im, Гц	Рисунок 4
+156 / 4	NumSampleImage	Кол-во выборок в образе ЗИ =0 – не используется	обязательный параметр для сложных сигналов. Для тона должен быть =0.
+160 / 1	SampleType	Тип отсчетов образца: 0 – комплексные, 1 – действительные, 2 – действительные, амплитуда 3..255 - резерв	Для ППФ может быть 0 или 1 (см. <b>Таблица 14</b> )
+161 / 1	Prm2	Параметры ЗИ: Если Type=0 или Type=1 – не используется	Кол-во периодов несущей в одном бите определяется параметром



		Если Type=2 (ФКМ): =0 – Баркер N=2 =1 – Баркер N=3 =2 – Баркер N=4 =3 – Баркер N=5 =4 – Баркер N=7 =5 – Баркер N=11 =6 – Баркер N=13 =7..255 - резерв	Prm1
+162 / 1	Prm3	Параметры ЗИ: Скважность мощности ЗИ в %: Значение 1..100 отражает мощность от 1 до 100 %.	Если 0, то скважность = 100% (для совместимости со старыми ЗИ)
+163 / 1	Prm4	Параметры ЗИ: Если Type=0 или Type=1 – не используется Если Type=2 (ФКМ): Способ формирования скачка фазы: =0 – классический =1 – период двойной частоты =2 - дырка	

Таблица 12 – Формат дополнительных полей структура gPPI_ZI версии 2			
Смещение, байт / размер, байт	Название	Описание	Примечание
+164 / 8	F2_StartFreq	Начальная частота несущей F2, Гц	Для тона F2_StartFreq = F2_EndFreq
+172 / 8	F2_EndFreq	Конечная частота несущей F2, Гц	

### 3.2.2. Формат массива образа ЗИ

**ПРИМЕЧАНИЕ.** Расчет образца для ФКМ (Type=2) выполняется на основе следующих параметров из структуры gPPI\_ZI:

- 1) StartFreq (определяет частоту несущей Fc)
- 2) Prm1 (определяет кол-во периодов несущей в бите)
- 3) Prm2 (определяет вид ЗИ)
- 4) Prm4 (определяет параметры ЗИ)

Struct gIMAGE\_SAMPLE // Массив отсчетов образа ЗИ

```
{
    Short Re; // Отсчет Re, от -32767 до +32768, средняя точка (смещение) = 0
    Short Im; // Отсчет Im, от -32767 до +32768, средняя точка (смещение) = 0
} image_sample[ NumSampleImage ];
// Кол-во отсчетов задается в поле gPPI_ZI.NumSampleImage
```

**ПРИМЕЧАНИЕ.** Первый отсчет Re образца (Sample[0].Re) отстоит от начала ЗИ (начала фазы А) на время 1/Fs.

**ПРИМЕЧАНИЕ.** Первый отсчет Re образца (Sample[0].Re) отстоит от начала ЗИ (начала фазы А) на время 1/Fs.

Формат выборок для комплексных отсчетов - Таблица 13.

Для ППФ всегда используются только действительные отсчеты. Формат выборок для действительных отсчетов ППФ - Таблица 14. Данное отличие в упаковке (в зависимости от поля **gPPI\_ZI.SampleType**) сделано для совместимости с предыдущими выпущенными файлами ppi для ППФ. Для ППФ отсчеты образца при **gPPI\_ZI.SampleType = 1** трактуются в программе HyScan всегда как long, но в самом образце могут использоваться и как short (поле Im=0) и как long (поле Im образца в этом случае представляет из себя старшие разряды long). Программа определяет разрядность образца путем определения макс. значения из отсчетов в образце.

<b>Таблица 13 – Формат массива выборок образа ЗИ (PPI_IMAGE_ZI) – структура gIMAGE_SAMPLE</b>		
Название	ТипРазмер (байт)	Описание
image_sample[0].Re	short (2)	
image_sample[0].Im	short (2)	
...		
image_sample[NumSampleImage -1].Re	short (2)	
image_sample[NumSampleImage -1].Im	short (2)	

<b>Таблица 14 – Использование образца ЗИ для ППФ</b>	
Поле gPPI_ZI.SampleType	Упаковка массива image_sample[ NumSampleImage ]
0	Данных образца нет (ошибка в расчете образца)!!! Равносильно <b>gPPI_ZI.NumSampleImage=0</b>
1	Упаковка Re (long), $Re[0] = image\_sample[0].Im * 2^{16} + image\_sample[0].Re$ , $Re[1] = image\_sample[1].Im * 2^{16} + image\_sample[1].Re$ , $Re[2] = image\_sample[2].Im * 2^{16} + image\_sample[2].Re$ , ..... Кол-во отсчетов образца = <b>gPPI_ZI.NumSampleImage</b>

### 3.2.3. Использование различных полей

#### Определение использования ЦФД:

Если бит 31 в Prm0 = 0 – ЦФД не используется, иначе – ЦФД используется.

#### Определение параметров ЗИ без использования ЦФД для тона (не ППФ):

Prm0:31 = 0 (бит 31 в Prm0 = 0 – ЦФД не используется)

Type = 0 (тон)

Отсчеты АЦП- прореженные (пара Re-Im)

Отсчеты на выходе – комплексные (пара Re-Im), следуют с Fs

Fc = StartFreq

Длительность ЗИ = TimeZI

Кол-во периодов Fc внутри ЗИ = NumImpFA

#### Определение параметров ЗИ без использования ЦФД для ЛЧМ (не ППФ):

Prm0:31 = 0 (бит 31 в Prm0 = 0 – ЦФД не используется)

Type = 1 (ЛЧМ)

Отсчеты АЦП- прореженные (пара Re-Im)

Отсчеты на выходе – комплексные (пара Re-Im), следуют с Fs

Интервал  $T_i = 1/F_{im}$

Начальная частота Fc = StartFreq

Конечная частота  $F_c = \text{EndFreq}$   
Длительность ЗИ =  $\text{TimeZI}$   
Кол-во периодов  $F_c$  внутри ЗИ =  $\text{NumImpFA}$

**При использовании ЦФД:**

$\text{Prm0:31} = 1$  (бит 31 в  $\text{Prm0} = 1$  – ЦФД используется)  
Отсчеты АЦП – равномерные  
 $F_{adc}$  = параметр  $F_{im}$  (частота отсчетов АЦП на входе ЦФД)  
 $F_s$  = параметр  $F_s$  (частота выходных комплексных отсчетов на выходе ЦФД)  
ПРИМЕЧАНИЕ. В старых версиях ЦФД:  $F_{adc} = F_c * 4$ ;  $F_s = F_{adc} / 20 = F_c * 4 / 20 = F_c / 5$  (фазового сдвига между излучаемой частотой и “принимаемой” частотой не было, при расчете амплитуды для тона используется стандартная формула).  
В новых версиях ЦФД:  $F_{adc} = F_c * 9$ ;  $F_s = F_{adc} / 27 = F_c * 9 / 27 = F_c / 3$ , но при этом  $T_{adc}$  не всегда точно равно  $T_c / 9$  (т.к.  $T_c = 2 * N * T_{clk}$ ,  $N$  может не быть кратно 9). Это необходимо учитывать при расчете амплитуды.

**Использование в H5p1:**

Используется пара ЗИ. Один ЗИ используется для описания излучаемого сигнала и принимаемого сигнала в канале НЧ. Второй ЗИ используется для описания излучаемого сигнала и принимаемого сигнала в канале ВЧ.

Излучаемый ЗИ	Параметры ЗИ для борта 1 (НЧ)	Параметры ЗИ для борта 2 (Эл)	Примечание
ТОН_АМ	Type = ТОН_АМ StartFreq = EndFreq =  SampleType = действительные  Образ ЗИ – ???	Type = ТОН  StartFreq = EndFreq =  SampleType = комплексные  Образ ЗИ – не используется	

#### **4. Контроль данных файлов DDS и PPI**

Версия 01.00 файла DDS:

Для тона размер файла должен быть равен  $16(\text{заголовок}) + 144(gPRM\_ZI) + 4(2 \text{ КУ}) = 164$  байт.

Версия 01.00 файла PPI:

Для тона размер файла должен быть равен  $16(\text{заголовок}) + 164(gPPI\_ZI) = 180$  байт.

5. Данные FLASH КОНГ

FLASH КОНГ имеет объем 16 Мбит (8 Мбайт).

FLASH используется для хранения данных DDS, используемых при работе комплекса.

**ПРИМЕЧАНИЕ.** Запись данных DDS осуществляется при настройке комплекса а также при обновлении ПО комплекса пользователем класса SYSTEM и не должна допускаться при работе пользователей других классов.

Данные DDS используются PGL для формирования сигналов ЗИ методом прямого синтеза. Во время работы ВС обеспечивает указание генератору ЗИ в ПГЛ на требуемые данные DDS (в зависимости от выбранного ЗИ), при формировании ЗИ генератор ЗИ считывает данные DDS из FLASH и на их основе формирует сигналы ЗИ и обеспечивает контроль формирования сигналов ЗИ. Формат данных DDS зависит от версии используемого генератора DDS (версии PGL). Данные DDS хранятся в виде непрерывного массива данных в двоичном виде.

**КОД ЗИ ДОЛЖЕН НАХОДИТЬСЯ В ДИАПАЗОНЕ ОТ 1 ДО 1022 (FLASH позволяет одновременно хранить до 1024 ЗИ, ЗИ с кодом 0 и 1023 не должны использоваться).**

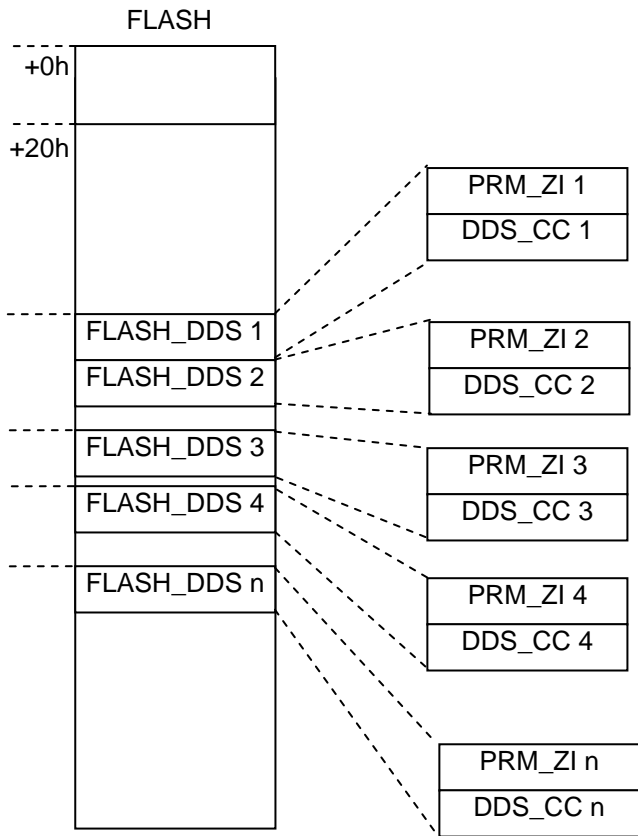


Рисунок 25. Размещение данных во FLASH

Данные во FLASH содержат:

1. Данные DDS (FLASH\_DDS)

### 5.1. Данные DDS (DataDDS)

Данные DDS содержат:

- Параметры ЗИ (PRM\_ZI)
- Массив КУ (DDS\_CC)

Следующие один за другим.

Упаковка PRM\_ZI аналогична упаковке PRM\_ZI в DDS файле (**Ошибка! Источник ссылки не найден.**).

Упаковка DDS\_CC аналогична упаковке DDS\_CC в DDS файле (Таблица 1).

Суммарный размер данных DDS для одного ЗИ является суммой размеров PRM\_ZI и DDS\_CC.

Этот размер и заносится

Данные DDS (заголовок+команды) могут занимать одну или несколько страниц FLASH.

Начальный адрес DataDDS может быть любым, кратным 2048 байтам.

Начальный адрес DataDDS вычисляется по формуле:

$\text{StartAddr} = \text{CodeZI} * 2048,$

Где:

CodeZI – код ЗИ

Для программы PGL42 максимальный размер DDS\_CC не должен превышать 512 КУ (512 слов или 1024 байт).

## Приложение А (обязательное). Упаковка массива КУ для генератора GCHIRP

Для генератора GCHIRP (см. 1.7) используется массив параметров, загружаемых в регистры генератора. Параметры работы генератора хранятся вместо массива КУ в файле \*.dds и описываются в виде следующей структуры GCHIRP\_PRM:

```
struct GCHIRP_PRM    // размер структуры = 17 байт
{
    u32 CodeM;        // параметр M, смещение +0, 4 байта
    u32 CodeN;        // параметр N, смещение +4, 4 байта
    u32 CodeP;        // параметр P, смещение +8, 4 байта
    u32 CodeZ;        // параметр TimeZ, смещение +12, 4 байта
    u8 Mode;          // параметр Mode, смещение +16, 1 байт
}
```

Данная структура записывается в файл \*.DDS в качестве массива КУ. Описание расчета и использования параметров приведено в описании генератора GCHIRP (см. 1.7). Описание параметров приведено ниже.

**Mode** – режим работы генератора GCHIRP:

=0 (ENA=0, DIR=0) или

=1 (ENA=0, DIR=1) – счетчик M не считает (значение DIR может быть любым), сумматор сбрасывается в ноль при переполнении (используется для генерации тона)

=2 (ENA=1, DIR=0) - счетчик M считает вниз (декремент), сумматор не сбрасывается в ноль при переполнении (используется для генерации ЛЧМ)

=3 (ENA=1, DIR=1) - счетчик M считает вверх (инкремент), сумматор не сбрасывается в ноль при переполнении (используется для генерации ЛЧМ)

=4..255 – резерв, не используются

**CodeM** – значение, загружаемое в счетчик M при запуске генератора (возможные значения от 0 до  $2^{18}-1$ )

**CodeN** – значение, загружаемое в делитель N (возможные значения от 0 до  $2^{23}-1$  отражают коэфф. деления от 1 до  $2^{24}$ )

**CodeP** – код кол-ва формируемых выходных периодов Fc (возможные значения от 0 до  $2^{18}-1$  отражают кол-во формируемых периодов от 1 до  $2^{18}$ ). Например, для формирования одного периода CodeP=0 (один полупериод фазы А и один полупериод фазы В), для двух периодов CodeP=1 (два полупериода фазы А и два полупериода фазы В) и т.д.

**CodeZ** – параметр TimeZ (код кол-ва тиков Tclk мертвого времени внутри каждого полупериода фазы А или В). Возможные значения от 0 до  $2^{16}-1$  отражают кол-во тиков от 2 до  $2^{16}+1$ . Например, для Tz=2 тик поле CodeZ=0, для Tz=3 тика поле CodeZ=1 и т.д.

Минимальное значение мертвого времени = 2 тика (CodeZ=0)

### ПРИМЕЧАНИЯ.

1. В зависимости от используемой версии PGL, в PGL могут вводиться аппаратные ограничения на минимальную длительность Tz. При этом если заданный с помощью CodeZ интервал Tz будет менее минимального значения Tz в PGL, реальный интервал Tz будет равен минимальному.
2. Если Tz больше времени полупериода, полупериод не формируется.

#### **Приложение Б (обязательное). Упаковка массива КУ для генератора DDS5**

Для генератора DDS5 (см. 1.8) используется массив КУ, загружаемых в ОЗУ генератора.

КУ хранятся в виде массива КУ в файле \*.dds и описываются в виде следующего массива:

U16 CC[];

Размер массива CC[] в байтах равен значению поля **gPRM\_ZI.NumCC** (см. 2).

Значащими являются только младшие 9 бит (биты КУ) из 16. Упаковка бит КУ приведена в

1.8.

Кол-во используемых КУ = **gPRM\_ZI.NumCC**/2.



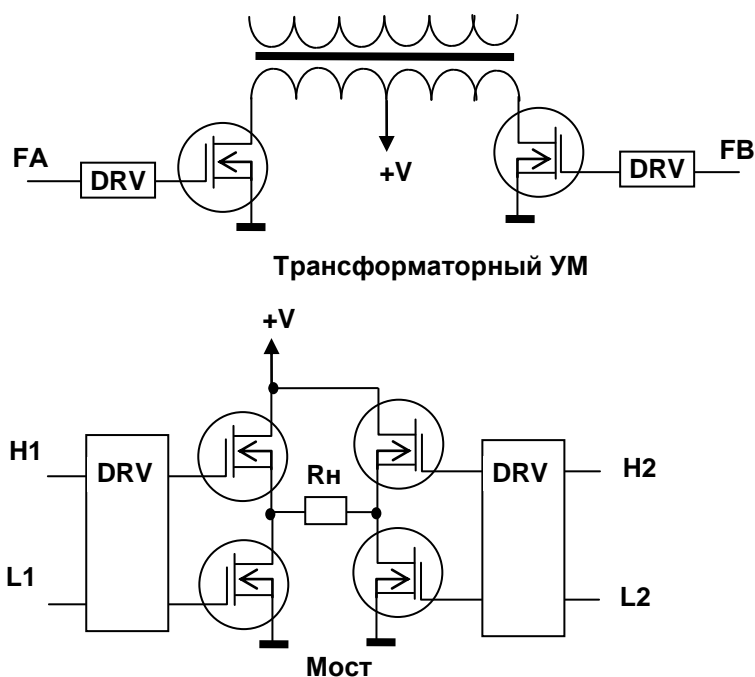


Рисунок 26. Сигналы возбуждения

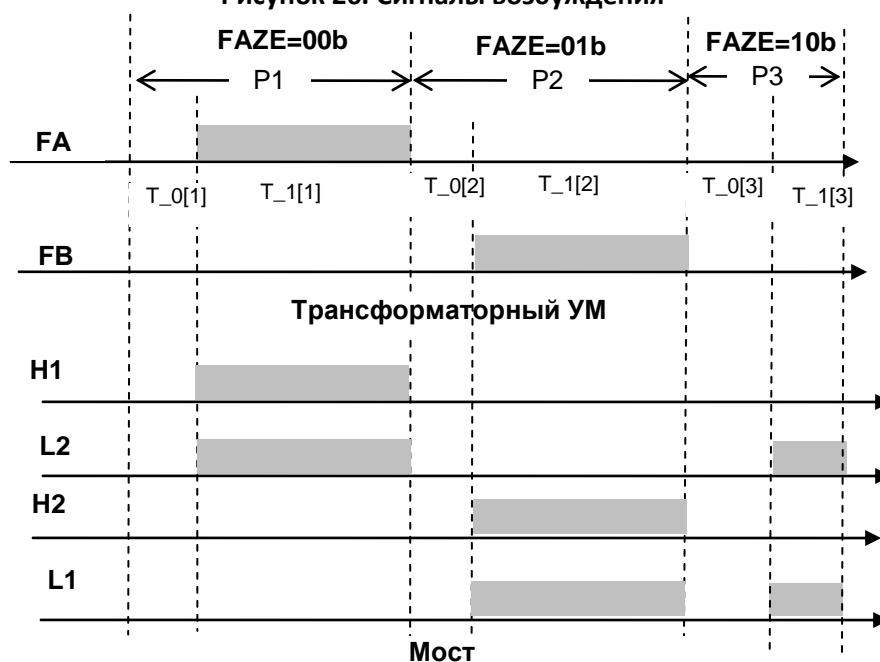


Рисунок 27. Формирование сигналов возбуждения (пример без использования ШИМ внутри полупериода)

Примеры использования генератора для формирования тонового ЗИ без ШИМ внутри полупериода:

Для задания одного полупериода достаточно использование одной КУ.

Мертвое время  $T_z = T_0$ .

$T_1 = P - T_z$ .

Для задания N периодов необходимо  $N \cdot 2$  КУ.

При мертвом времени в 12 тиков и  $T_{clk} = 8$ нс (96 нс): максимальное значение длительности полупериода составит  $12 + 256 = 268$  тиков, макс. длительность периода составит  $268 \cdot 2 = 536$  тиков или 4,3 мкс (233 кГц).

Вариант 1

Обеспечивается формирование одного или нескольких полупериодов с заданной длительностью и отсутствием ШИМ по мощности (Рисунок 28).

Для работы в режиме 0 параметр Mode=0.

Режим ориентирован на сигналы, в которых каждый период или полупериод уникальны.

Каждый из полупериодов может быть уникальным по длительности.

Четные КУ определяют полупериод А, нечетные – полупериод В. В КУ используются 9 бит для задания длительности полупериода (Таблица 15). Переключение фаз происходит автоматически.

Параметры для работы в режиме 0, задаваемые ВС:

- К – коэффициент деления делителя
- N – кол-во используемых КУ (кол-во формируемых полупериодов)
- КУ – команды КУ, записываемые в RAM\_DDS
- Mode=0 – режим работы = 0

Мертвое время контролируется минимальным значением, зависящим от аппаратуры. Параметр Tz\_min определяет минимальное время Tz в тиках Tclk, является внутренней константой и зависит от конкретной прошивки.

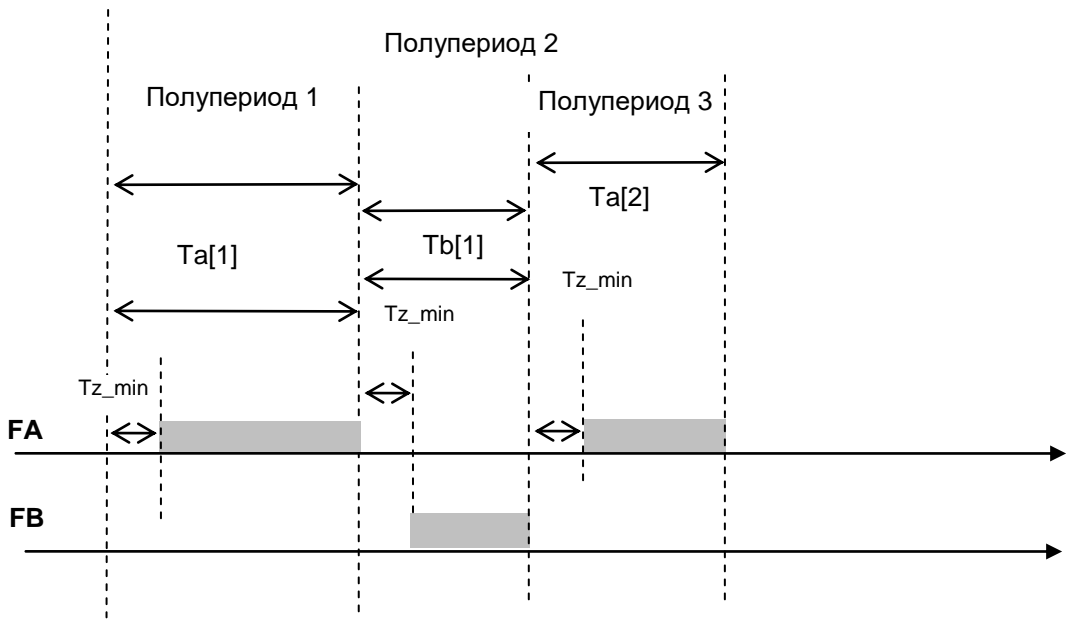


Рисунок 28. Работа в режиме 0

**ПРИМЕЧАНИЕ.** Если формируемый полупериод < Tz\_min, то импульс на соответствующей фазе не формируется.

Параметр К рассчитывается по формуле:

$$K = \text{ЦЕЛОЕ} (T/T_{clk}) - 1$$

где T – требуемое время тика T, с

Tclk – период системной синхрочастоты Tclk, с

Минимальное значение K = 0 (T=Tclk)

Максимальное значение K = 255 (T=256\*Tclk)

Минимальный шаг изменения K = 1.

Таблица 15 – Формат упаковки КУ в RAM\_DDS для режима 0

Команда (адрес в ОЗУ)	Биты 8-0
CC0 (+0)	TimeA[1]

CC1 (+1)	TimeB[1]
CC2 (+2)	TimeA[2]
CC3 (+3)	TimeB[2]
...	...
CCn-2 (+n-2)	TimeA[n/2]
CCn-1 (+n-1)	TimeB[n/2]

Значение параметра TimeA[i] и TimeB[i] в КУ рассчитывается по формуле:

$\text{TimeA}[i] = \text{ЦЕЛОЕ} (\text{Ta}[i]/T) - 1$

$\text{TimeB}[i] = \text{ЦЕЛОЕ} (\text{Tb}[i]/T) - 1$

где Ta[i] – требуемое время фазы А полупериода i, с

где Tb[i] – требуемое время фазы В полупериода i, с

T – длительность тика T, с

Минимальное значение TimeA[i] = 0 (Ta=T).

Минимальное значение TimeB[i] = 0 (Tb=T).

Максимальное значение TimeA[i] = 511 (Ta=512\*T)

Максимальное значение TimeB[i] = 511 (Tb=512\*T)

Минимальный шаг изменения TimeA[i] и TimeB[i] = 1.

Значение параметра N рассчитывается по формуле:

N = кол-во формируемых полупериодов -1

Минимальное значение N=0 (будет сформирован только один полупериод фазы А)

Максимальное значение N = N\_max (N\_max - максимальное кол-во КУ в RAM\_DDS)

## Вариант 2

Обеспечивается формирование одного или нескольких полупериодов с заданной длительностью и возможностью использования ШИМ внутри полупериода или периода. Также возможно формирование сигналов ФЗМ, импульсов гашения и т.д..

Параметры для работы в режиме 1, задаваемые ВС:

K – коэффициент деления делителя

N – кол-во используемых КУ

KY – команды КУ, записываемые в RAM\_DDS

Mode=1 – режим работы = 1

Каждая КУ позволяет сформировать один уникальный временной интервал (длительность интервала кратна T) или выполнить один или несколько раз формирование последних 4-х временных интервалов, предшествующих текущей КУ.

В КУ используются 2 бита для задания действия и 7 бит для задания параметра действия (Таблица 16).

**Таблица 16 – Формат упаковки КУ в RAM\_DDS для режима 1**

Команда (адрес в ОЗУ)	Биты	
	8-7	6-0
CC0 (+0)	Cntrl[1]	Time[1]/Rep[1]
CC1 (+1)	Cntrl [2]	Time[2] /Rep[2]
...	.....	...
CCn (+n-1)	Cntrl[n]	Time[n] /Rep[n]

Параметр Cntrl определяет выполняемое действие:

Cntrl=0 – формирование интервала отсутствия сигнала по обеим фазам

Cntrl=1 – формирование интервала на фазе А (FA=1)

Cntrl=2 – формирование интервала на фазе В (FB=1)

Cntrl=3 – выполнение повтора 4-х предыдущих интервалов (образца) заданное кол-во раз

Значение параметра Time[i] при Cntrl=0, Cntrl=1 или Cntrl=2 в КУ рассчитывается по формуле:

$\text{Time}[i] = \text{ЦЕЛОЕ} (T[i]/T) - 1$

где  $T[i]$  – требуемое время интервала  $i$ , с

$T$  – длительность тика  $T$ , с

Минимальное значение  $\text{Time}[i] = 0$  ( $T_i = T$ ).

Максимальное значение  $\text{Time}[i] = 127$  ( $T_i = 128 * T$ )

Минимальный шаг изменения  $\text{Time}[i] = 1$ .

Значение параметра  $\text{Rep}[i]$  при  $\text{Cntrl}=3$  в КУ рассчитывается по формуле:

$\text{Rep}[i] = \text{Rep} - 1$

где  $\text{Rep}$  – требуемое кол-во повторов

Минимальное значение  $\text{Rep}[i] = 0$  (один раз).

Максимальное значение  $\text{Rep}[i] = 127$  (128 раз)

Минимальный шаг изменения  $\text{Rep}[i] = 1$ .

Мертвое время контролируется минимальным значением, зависящим от аппаратуры, для:

- текущего интервала  $\text{Cntrl}=0$ , если предыдущий интервал был  $\text{Cntrl}=1$  или  $\text{Cntrl}=2$
- текущего интервала  $\text{Cntrl}=2$ , если предыдущий интервал был  $\text{Cntrl}=1$ .
- текущего интервала  $\text{Cntrl}=1$ , если предыдущий интервал был  $\text{Cntrl}=2$ .

Параметр  $Tz\_min$  определяет минимальное время  $Tz$  в тиках  $Tclk$ , является внутренней константой и зависит от конкретной прошивки.

Параметр  $K$  рассчитывается по формуле, аналогичной для режима 0 (**Ошибка! Источник ссылки не найден.**).

Значение параметра  $N$  рассчитывается по формуле:

$N = \text{кол-во формируемых КУ} - 1$

Например, для использования ШИМ внутри периода, для формирования каждого уникального полупериода используется две КУ. Одна КУ – для формирования мертвого времени, вторая КУ – для формирования импульса возбуждения соответствующей фазы (Рисунок 29).

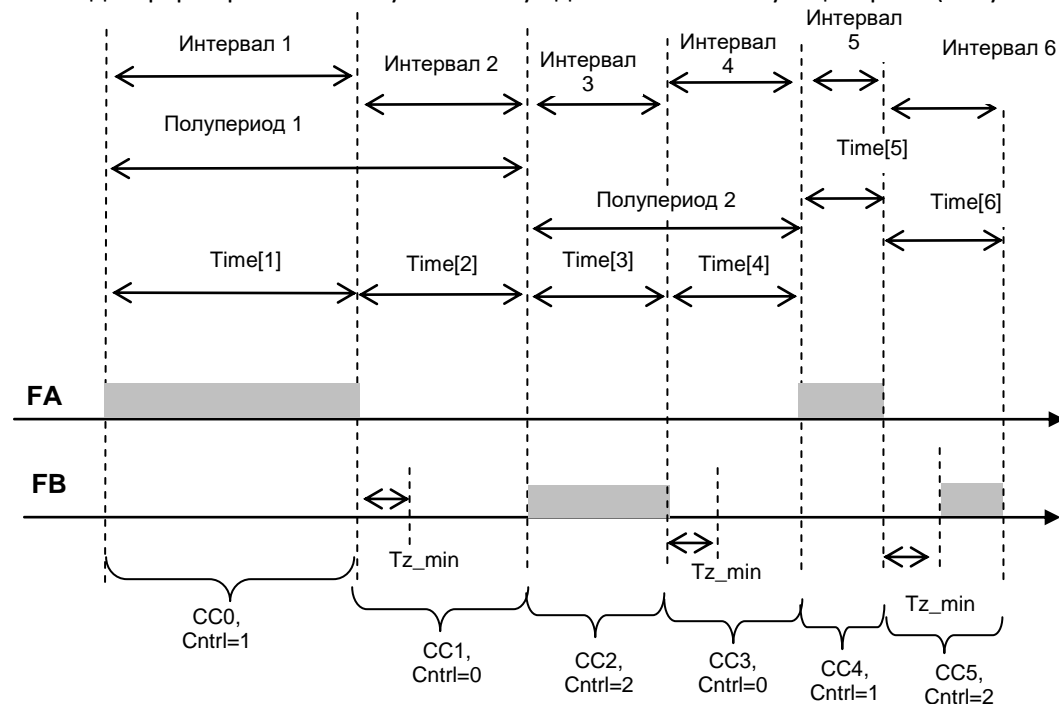


Рисунок 29. Работа в режиме 1 (использование ШИМ внутри периода)

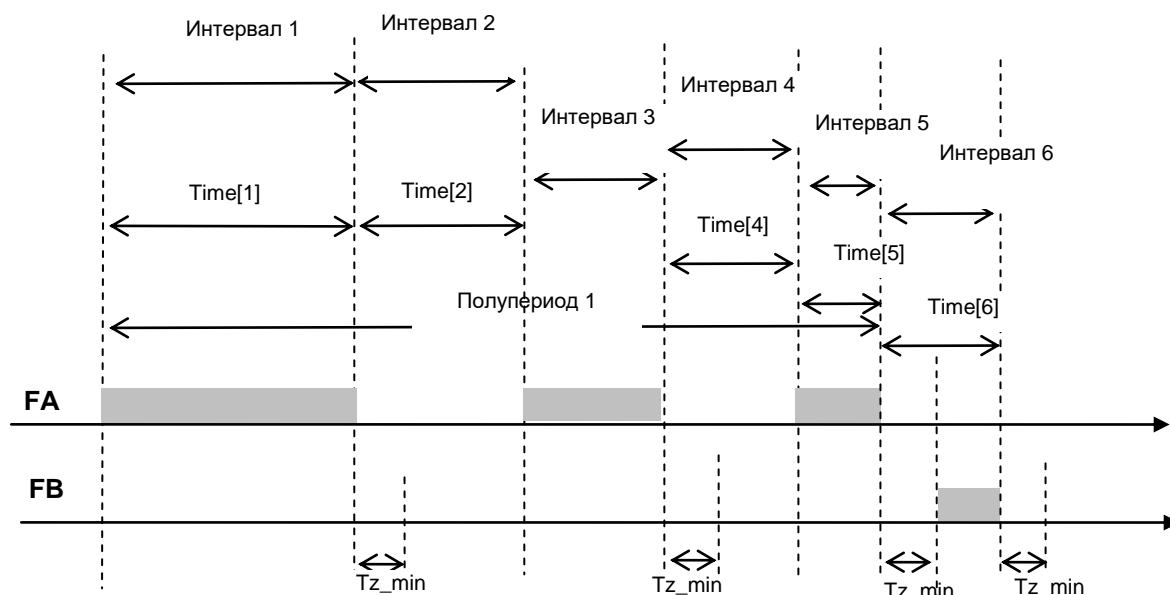


Рисунок 30. Работа в режиме 1 (использование ШИМ внутри полупериода)

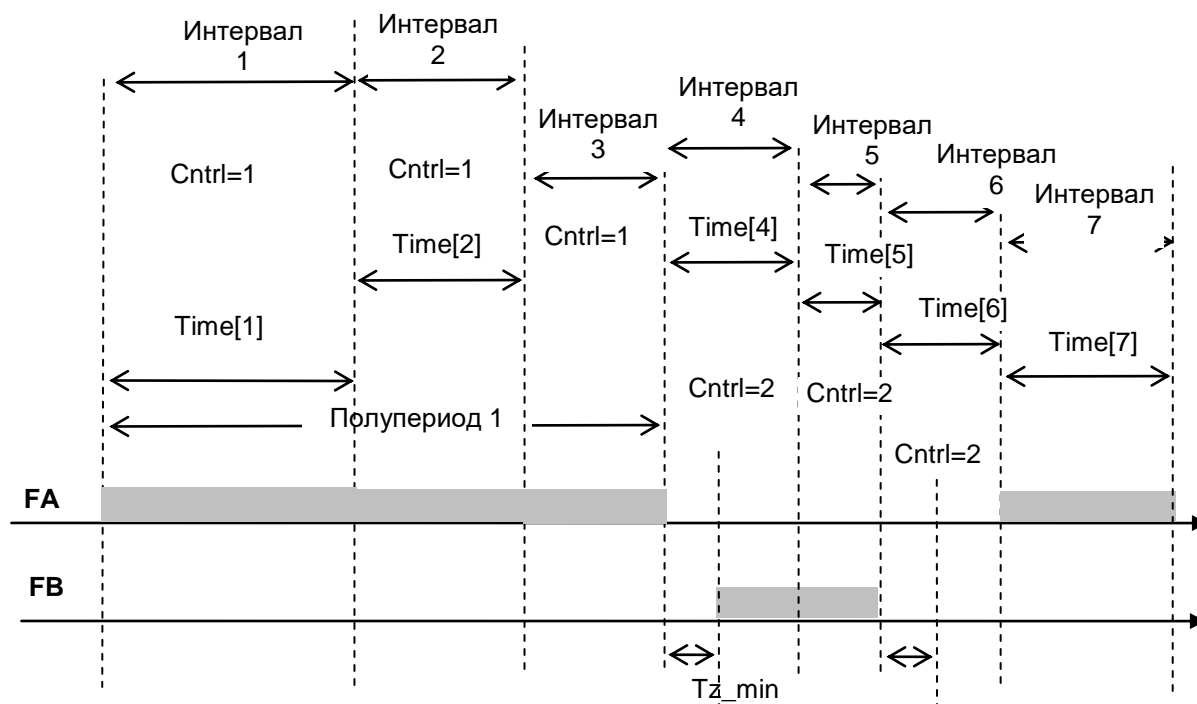


Рисунок 31. Работа в режиме 1 (удлинение импульса в полупериоде)

#### 5.1.1.1. Пример формирования ФЗМ в режиме 1

ЗИ с ФЗМ (Рисунок 32) состоит из нескольких битовых интервалов – квантов (кванты нумеруются от 1 до  $n$ ).

Каждый квант содержит несколько периодов несущей ( $M$ ) и задержку (Delay).

Кол-во периодов в каждом кванте, как правило, одинаково ( $M_1=M_2=\dots=M_k=M$ ).

Задержка может быть различна для каждого кванта (длительность квантов различна) и кратна  $T$ .

Для последнего кванта задержка не используется ( $=0$ ).

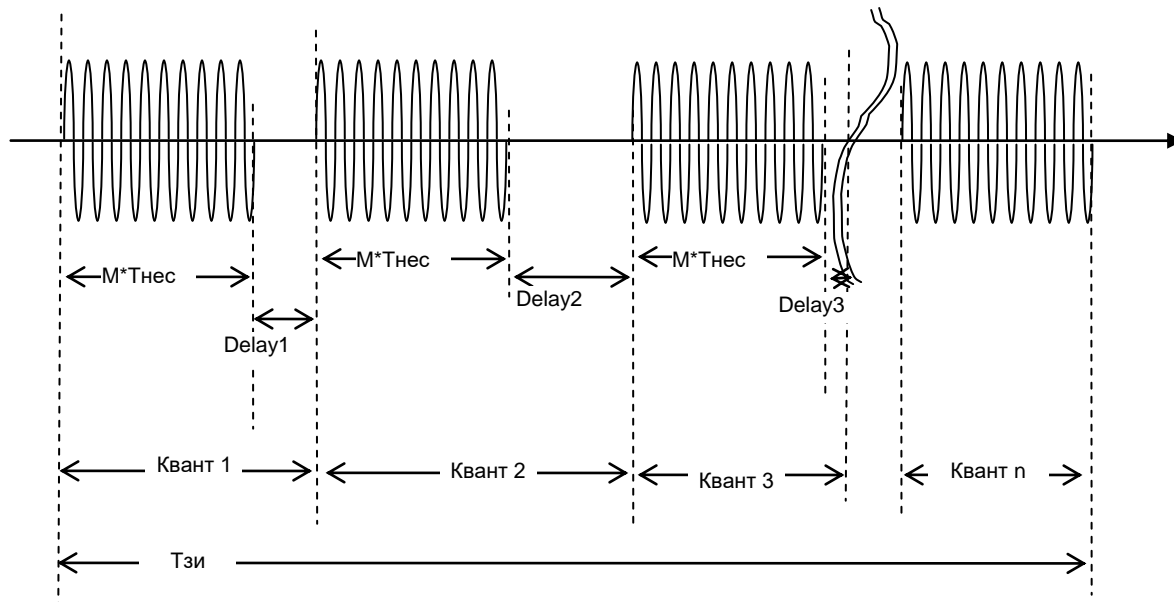


Рисунок 32. Пример формирования ФЗМ

Для формирования одного кванта ФЗМ используются 6 КУ (Рисунок 33):

- первые 4 КУ для описания одного периода внутри кванта 1
- следующая КУ для формирования оставшихся периодов внутри кванта 1
- следующая КУ для формирования задержки кванта 1

Если пауза большая, возможно использование еще одной или нескольких КУ для расширения длительности паузы.

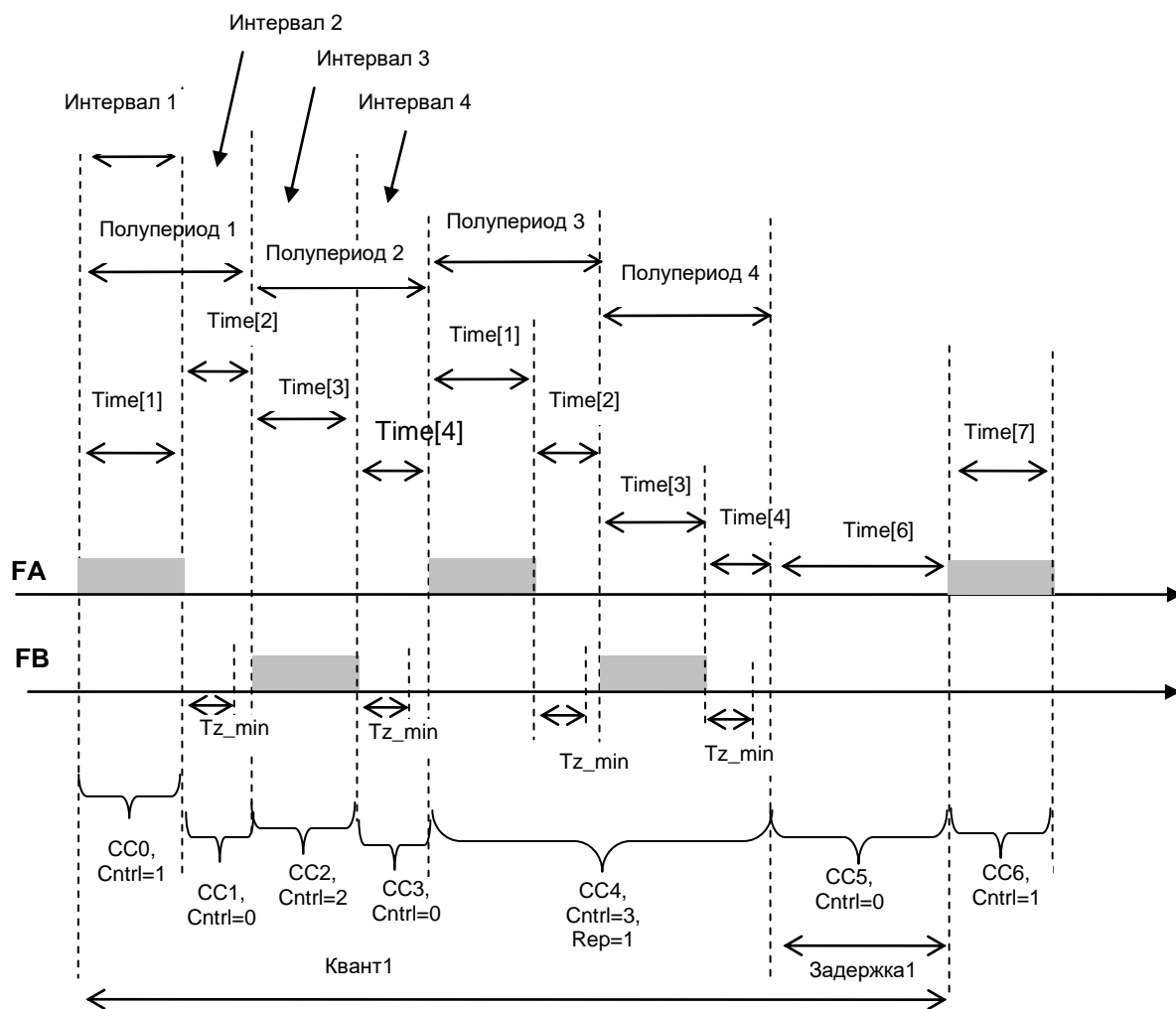


Рисунок 33. Работа в режиме 1 (формирование ФЗМ)