

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ
«МЭИ»

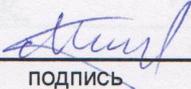
Институт ИРЭ

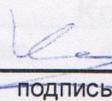
Кафедра РТС

ВЫПУСКНАЯ РАБОТА
БАКАЛАВРА

по направлению 11.03.01 Радиотехника
(номер и название)

Тема: Универсальный генератор дальномерных
кодов сигналов навигационных систем

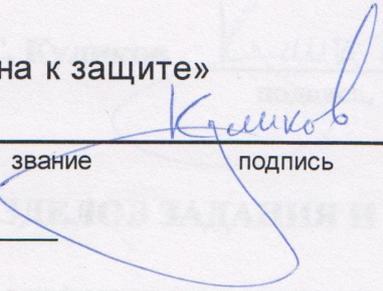
Студент ЭР-11-15  Михайлова О.К.
группа подпись фамилия, и., о.

Научный
руководитель доц.  Корогодин И.В.
должность звание подпись фамилия, и., о.

Консультант _____
должность звание подпись фамилия, и., о.

Консультант _____
должность звание подпись фамилия, и., о.

«Работа допущена к защите»

Зав. кафедрой  Куликов Р.С.
звание подпись фамилия, и., о.

Дата _____

Москва

2019 г.

2. Найти инициализирующие параметры для различных сигналов, разработать инициализирующие функции, провести анализ формируемых кодов: сравнить коды с проверочными, заданными в ИКД.
3. Провести оценку ресурсоемкости реализации генератора на ПЛИС, оценить возможности по снижению затрат ресурсов.
4. Сформировать часть сигналов на векторном генераторе, осуществить прием, построить корреляционные функции. Осуществить прием эфирных сигналов.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. ГЛОНАСС. Принципы построения и функционирования/ под ред. А.И. Перова, В.Н. Харисова. Изд. 4-е, перераб. и доп. – М.: Радиотехника, 2010. 800 с., ил.
2. Практические основы аналоговых и цифровых схем/ Каплан Д., Уайт К. – Техносфера, 2006.
3. ГЛОНАСС. Интерфейсный контрольный документ. Навигационный радиосигнал в диапазонах L1, L2, 2010.
4. ГЛОНАСС. Интерфейсный контрольный документ. Навигационный радиосигнал открытого доступа с кодовым разделением в диапазоне L1, 2016.
5. ГЛОНАСС. Интерфейсный контрольный документ. Навигационный радиосигнал открытого доступа с кодовым разделением в диапазоне L2, 2016.
6. ГЛОНАСС. Интерфейсный контрольный документ. Навигационный радиосигнал открытого доступа с кодовым разделением в диапазоне L3, 2016.
7. Интерфейсный контрольный документ NAVSTAR IS-GPS-200J, 2018.
8. Интерфейсный контрольный документ Galileo signal-in-space, 2018.
9. Интерфейсный контрольный документ BeiDou Open Service Signal (Version 2.1), 2016.

Адрес электронной почты студента:

olyaorient@yandex.ru

Примечание: задание брошюруется вместе с выпускной работой и с отзывами на нее руководителя и рецензента.

Оглавление

Введение	6
Глава 1 Формирование псевдослучайных последовательностей сигналов GPS, ГЛОНАСС, Galileo и Beidou	7
1.1 Объект работы и постановка задачи	10
1.1.1 Исходный генератор дальномерных кодов	11
1.2 Анализ особенностей формирования дальномерных кодов современных и перспективных навигационных сигналов системы ГЛОНАСС	14
1.2.1 LxOF, LxSF - сигналы с частотным разделением	14
1.2.2 L1OC, L2OC, L3OC - сигналы с кодовым разделением	16
1.3 Анализ особенностей формирования дальномерных кодов современных и перспективных навигационных сигналов системы NAVSTAR GPS	25
1.3.1 L1C/A, L1 P(Y), L2 P(Y)	25
1.3.2 L1C, L2C, L5 – открытые перспективные сигналы	31
1.4 Анализ особенностей формирования дальномерных кодов современных и перспективных навигационных сигналов системы Galileo	38
1.4.1 E5 - сигнал с кодовым разделением	38
1.4.2 E1, E6 - сигналы с кодовым разделением	41
1.5 Анализ особенностей формирования дальномерных кодов современных и перспективных навигационных сигналов системы Beidou	43
1.5.1 B1, B2 - сигналы с кодовым разделением	43
1.5.2 B3I - сигнал с кодовым разделением	46
1.5.3 B2a - сигнал с кодовым разделением	48
1.6 Выводы	52

Глава 2	Модель универсального генератора дальномерного кода	56
2.1	Модернизация генератора дальномерного кода	56
2.2	Модель универсального генератора дальномерного кода	58
2.3	Интерфейс генератора дальномерного кода	60
2.4	Функции инициализации	62
2.5	Анализ использования разрядов	68
Глава 3	Формирование и прием навигационных сигналов	73
3.1	Векторные генераторы сигналов	73
3.2	Формирование и прием сигнала с модуляцией дальномерным кодом и навигационным сообщением	77
3.3	Формирование и прием сигнала с модуляцией поднесущей и временным мультиплексированием	83
3.4	Прием референсной записи	87
Заключение	93
Список литературы	95
Приложение А.	Листинг программы, формирующей файлы низкочастотных эквивалентов сигналов для генератора SMBV	97

Введение

Спутниковые навигационные системы служат для определения положения, скорости и синхронизации потребителей этих систем. Для этого разворачивается орбитальная группировка спутников, излучающих радиосигналы. Формирование сигналов жестко привязано к единой шкале времени, но на их распространение требуется время, около 70 мс. В итоге принимаемые потребителем сигналы имеют задержку, причем, соотношение задержек сигналов разных спутников зависит от положения приемника. Оценив эти задержки, потребитель может рассчитать собственное положение. Чем точнее измеряется задержка сигнала, тем лучше можно определить положение. Для упрощения задачи оценки задержки сигналы навигационных спутников промодулированы специальными дальномерными кодами [1].

В настоящее время спутниковые группировки навигационных систем — самые многочисленные из представленных на земной орбите. Системы NAVSTAR GPS и ГЛОНАСС полностью развернуты, Beidou и Galileo близки к штатной численности аппаратов. В каждой из систем около 30 спутников, каждый спутник испускает до десятка навигационных сигналов. Различные системы, различные спутники, различные типы сигналов на них - всё это порождает большое число дальномерных кодов и видов их генераторов.

Объектом данной работы является универсальный генератор дальномерных кодов, позволяющий осуществить прием большинства существующих и перспективных сигналов.

1 Формирование псевдослучайных последовательностей сигналов GPS, ГЛОНАСС, Galileo и Beidou

Навигационные системы постепенно развиваются: увеличивается численность орбитальной группировки, модифицируется и расширяется наземный комплекс управления, вводятся новые типы навигационных сигналов. На рисунке 1.1 отображены полосы частот, занимаемые конкретными навигационными сигналами различных систем.

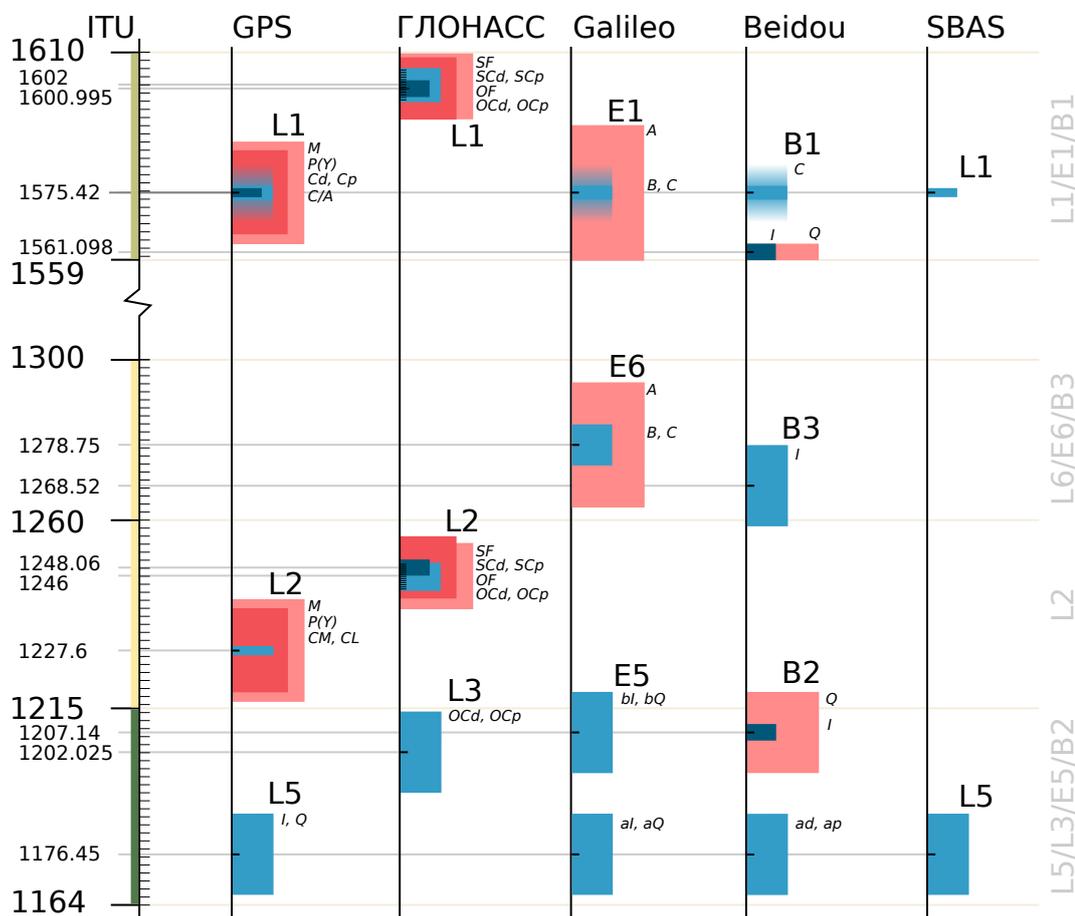


Рисунок 1.1 — Распределение навигационных сигналов по L-диапазону

Новые сигналы обладают более сложной структурой по сравнению с гражданскими сигналами, используемыми с 80-х годов прошлого века – ГЛОНАСС СТ и GPS L1C/A.

Старые сигналы модулируются данными и периодическими псевдослучайными последовательностями, их математическая модель проста:

$$S(t) = A \cdot C \cdot D \cdot \cos(f_0 t + \varphi), \quad (1.1)$$

где

- A – амплитуда сигнала;
- C – модуляция дальномерным кодом, принимает значения $+1$ и -1 при значениях дальномерного кода 0 и 1 соответственно;
- D – модуляция навигационным сообщением, принимает значения $+1$ и -1 при значениях символа цифровой информации 0 и 1 соответственно;
- f_0 – несущая частота.

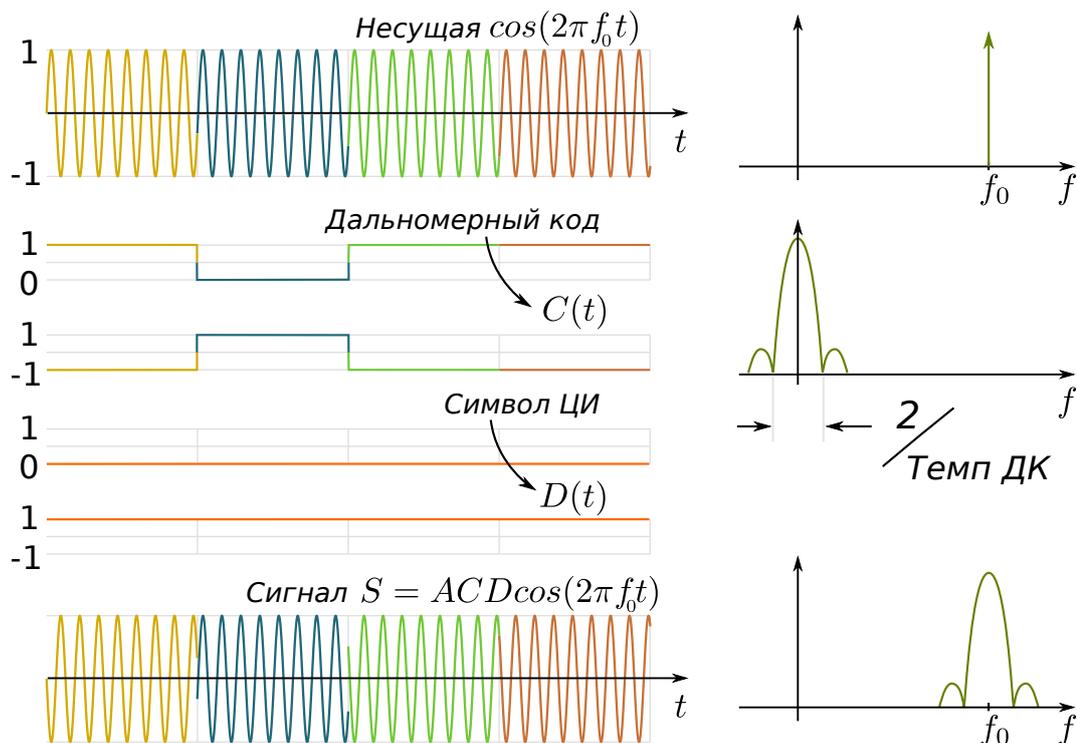


Рисунок 1.2 — Структура сигнала с модуляцией дальномерным кодом и навигационным сообщением

Таким образом, для этих сигналов понятия дальномерного кода и псевдослучайной последовательности совпадают. Современные и перспективные сигналы имеют более сложную структуру. В них может использоваться модуляция поднесущей для увеличения потенциальной точности, дополнительные оверлей-

ные коды для снижения внутрисистемных помех и временное уплотнение сигналов. Математическая модель современного сигнала в общем случае может быть записана как:

$$S(t) = A \cdot C \cdot B \cdot M \cdot O \cdot D \cdot \cos(f_0 t + \varphi), \quad (1.2)$$

где

- B – модуляция цифровой поднесущей, принимает значения $+1$ и -1 ;
- M – функция стробирования при временном уплотнении;
- O – модуляция оверлейным кодом, принимает значения $+1$ и -1 при значениях символа оверлейного кода 0 и 1 соответственно.

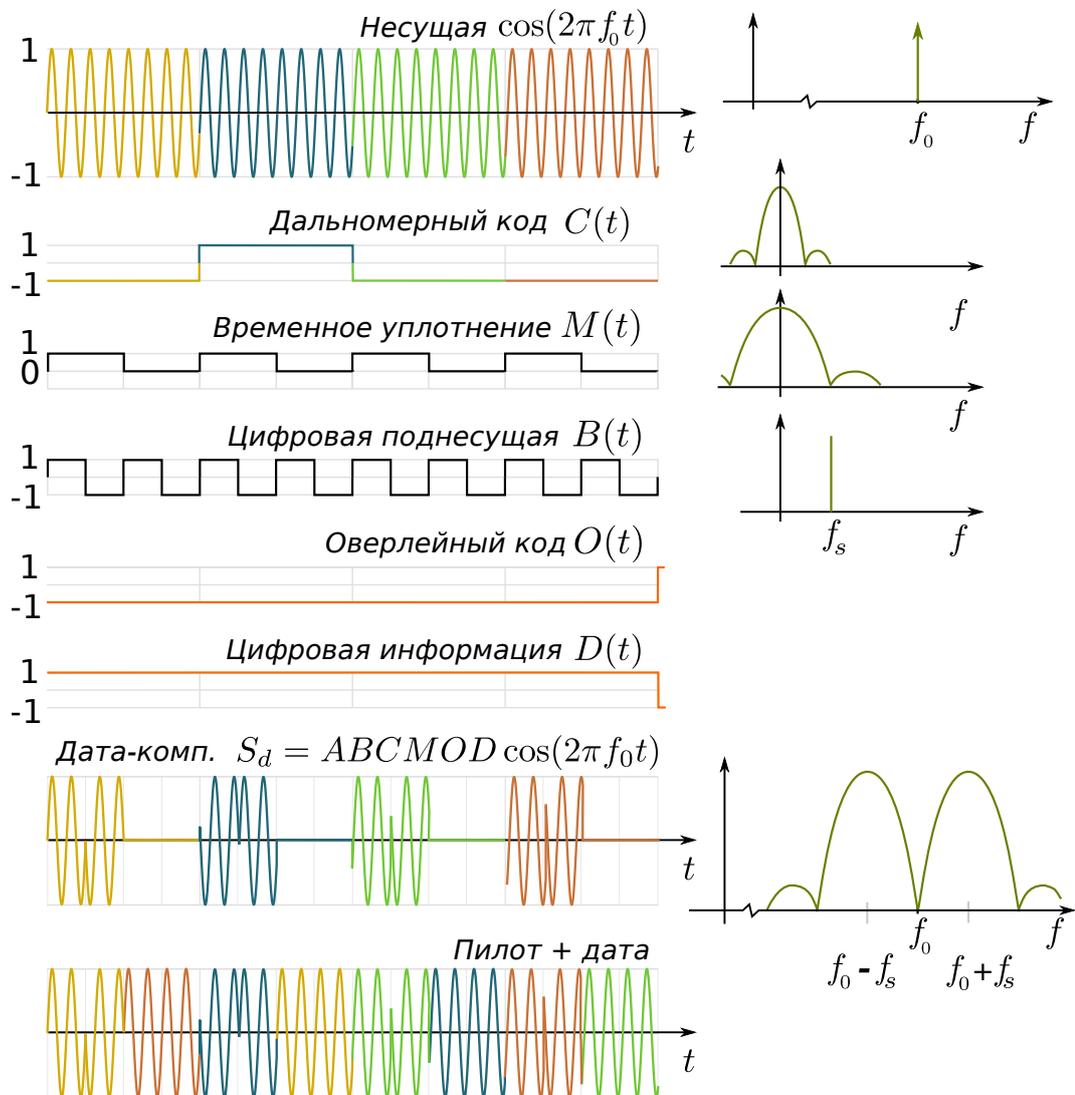


Рисунок 1.3 – Структура сигнала с модуляцией дальномерным кодом, цифровой поднесущей, оверлейным кодом и навигационным сообщением

Вид спектра сигналов (1.2) определяется произведением функций дальномерного кода, модуляции поднесущей и временного уплотнения, это произведение будем называть спектрорасширяющей последовательностью:

$$G(t) = B(t) \cdot C(t) \cdot M(t) \quad (1.3)$$

Модели сигналов более простой структуры могут быть получены из общей модели (1.2) путем исключения лишних множителей. Например, если в сигнале не используется временное уплотнение или модуляция цифровой поднесущей, то соответствующие множители принимаем равными единице.

1.1 Объект работы и постановка задачи

Сигналы глобальных навигационных систем можно разделить на три группы по типу используемых генераторов псевдослучайных последовательностей:

- с формированием ПСП на сдвиговых регистрах с линейной обратной связью;
- с использованием табличной ПСП;
- с формированием ПСП как функции времени с использованием криптографических ключей.

В УИЦ ЛНС с 2010 года используется модуль формирования дальномерных кодов, позволяющий формировать различные ПСП, относящиеся к первой группе, на одной и той же структуре генератора. Он использовался для формирования ПСП для сигналов GPS L1C/A и сигналов ГЛОНАСС с частотным разделением. Это проверенная и работающая конструкция, но она занимает много аппаратных ресурсов при реализации ее на ПЛИС. Отсюда возникают две задачи:

1. Расширение списка принимаемых сигналов,
2. Минимизирование расхода аппаратных ресурсов на реализацию генератора.

Универсальный генератор псевдослучайных последовательностей и даль-
номерных кодов стал объектом настоящей работы, опишем его подробнее.

1.1.1 Исходный генератор дальномерных кодов

Универсальный генератор имеет структуру, приведенную на рис. 1.4. Та-
кой генератор содержит два регистра сдвига – SR_1 и SR_2. Каждый регистр
имеет по 3 управляющих слова:

- CODE_STATE – для установки начального состояния регистра сдвига,
- CODE_BITMASK – для выбора отводов обратной связи,
- CODE_OUT_BITMASK – для выбора отводов на выход.

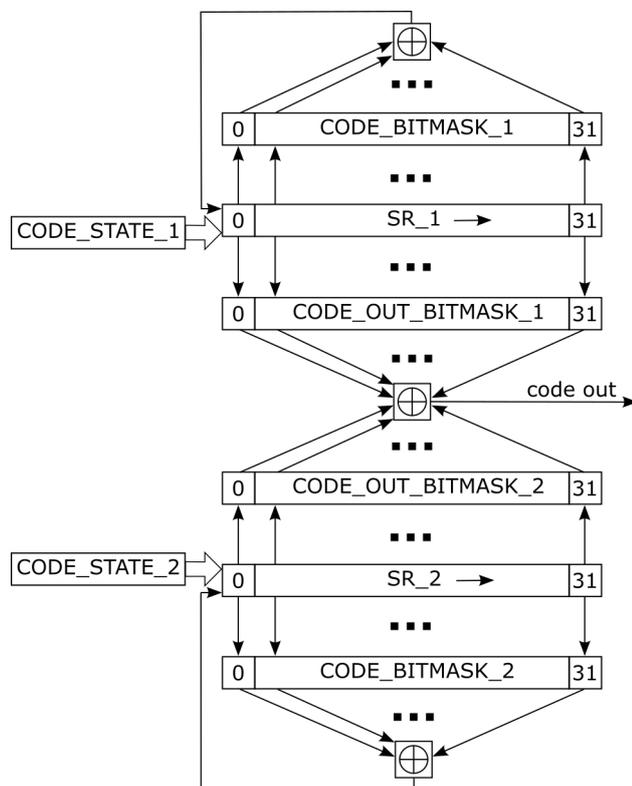


Рисунок 1.4 — Схема генератора дальномерных кодов

Входной интерфейс модуля генератора реализован посредством шины ад-
реса и шины данных, через которые организован доступ к конфигурационным
регистрам модуля. Выходным сигналом является псевдослучайная последова-
тельность (рис. 1.5).

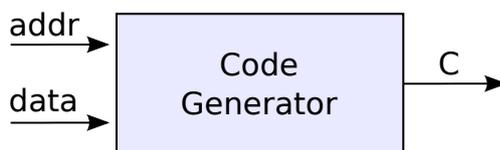


Рисунок 1.5 — Модуль - генератор дальномерного кода

Описанная выше структура генератора отображается на карту памяти в виде регистров представленных в таблице 1.1.

Таблица 1.1 — Карта памяти модуля Code Generator

Смещение	Название	Описание
0x00(0x00)	CODE_STATE_1	Начальное состояние регистра сдвига 1
0x04(0x01)	CODE_BITMASK_1	Регистр маски обратных связей 1
0x08(0x02)	CODE_OUT_BITMASK_1	Регистр маски выхода 1
0x0C(0x03)	CODE_STATE_2	Начальное состояние регистра сдвига 2
0x10(0x04)	CODE_BITMASK_2	Регистр маски обратных связей 2
0x14(0x05)	CODE_OUT_BITMASK_2	Регистр маски выхода 2

Для формирования дальномерного кода сигналов GLONASS L1OF, L2OF генератор настраивается следующим образом. В регистры генератора записываются значения приведенные в таб. 1.2.

Таблица 1.2 — Инициализирующие параметры для сигнала Gln LxOF

Название регистра	Конфигурационное слово
CODE_STATE_1	0xFFFFFFFF
CODE_BITMASK_1	0x08800000
CODE_OUT_BITMASK_1	0x02000000
CODE_STATE_2	0x00000000
CODE_BITMASK_2	0x00000000
CODE_OUT_BITMASK_2	0x00000000

Для формирования дальномерного кода сигнала GPS L1 C/A в регистры

генератора записываются значения приведенные в таб. 1.3.

Таблица 1.3 — Инициализирующие параметры для сигнала Gps L1CA

Название регистра	Конфигурационное слово
CODE_STATE_1	0xFFFFFFFF
CODE_BITMASK_1	0x20400000
CODE_OUT_BITMASK_1	0x00400000
CODE_STATE_2	0xFFFFFFFF
CODE_BITMASK_2	0x65C00000
CODE_OUT_BITMASK_2	зависит от номера спутника

1.2 Анализ особенностей формирования дальномерных кодов современных и перспективных навигационных сигналов системы ГЛОНАСС

Для решения первой задачи – расширения списка принимаемых сигналов, проведем анализ интерфейсных контрольных документов (ИКД) сигналов различных систем.

В ИКД для каждого сигнала приведена рекомендуемая схема формирования дальномерного кода. Эти схемы попробуем привести к виду универсального генератора дальномерных кодов. Выясним, дальномерные коды каких сигналов возможно формировать с помощью нашей структуры генератора, а какие нет. Для сигналов, дальномерные коды которых удастся наложить на структуру универсального генератора найдем инициализирующие параметры.

Сигналы навигационной системы ГЛОНАСС:

- L1OF, L2OF – открытые сигналы с частотным разделением
- L1SF, L2SF – сигналы санкционированного доступа с частотным разделением
- L1OC – сигнал с кодовым разделением в диапазоне L1
- L2OC – сигнал с кодовым разделением в диапазоне L2
- L3OC – сигнал с кодовым разделением в диапазоне L3
- L1SC, L2SC – сигналы санкционированного доступа с кодовым разделением

1.2.1 LxOF, LxSF - сигналы с частотным разделением

- L1OF(L1CT), L2OF(L2CT) - открытые сигналы с частотным разделением в диапазонах L1 и L2.
- L1SF(L1BT), L2SF(L2BT) - сигналы санкционированного доступа с частотным разделением в диапазонах L1 и L2.

Структура сигналов

Навигационный радиосигнал, передаваемый каждым НКА системы ГЛОНАСС на собственной несущей частоте в поддиапазонах L1 и L2, является многокомпонентным фазоманипулированным сигналом. Фаза несущего колебания модулируется двоичной последовательностью, образованной суммированием по модулю два дальномерного кода, навигационного сообщения и вспомогательного меандрового сигнала[2].

$$S(t) = A \cdot C_{СТ}(t) \cdot D(t) \cdot \cos(2\pi\omega t + \varphi) + A \cdot C_{ВТ} \cdot D(t) \cdot \sin(2\pi\omega t + \varphi) \quad (1.4)$$

Формирование дальномерного кода стандартной точности

Дальномерный код стандартной точности представляет собой псевдослучайную последовательность максимальной длины регистра сдвига с периодом повторения 1 мс и скоростью передачи символов 511 кбит/с. Псевдослучайный дальномерный код снимается с 7-го разряда 9-ти разрядного регистра сдвига. Код начального состояния регистра сдвига соответствует наличию "1" во всех разрядах регистра. Начальным символом в периоде дальномерного кода является 1-ый символ в группе 111111100, повторяющийся через 1 мс. Образующий полином, соответствующий регистру сдвига, формирующему ПС дальномерный код, имеет следующий вид:

$$G(x) = 1 + x^5 + x^9 \quad (1.5)$$

Упрощенная структурная схема формирования дальномерного кода приведена на Рис. 1.6

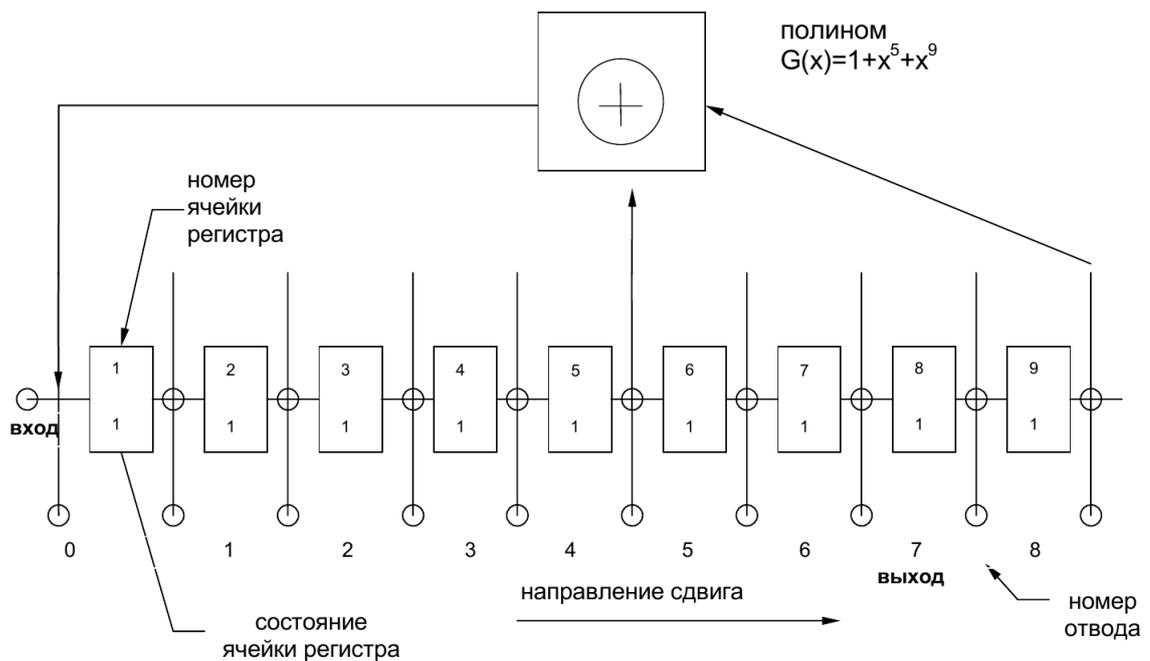


Рисунок 1.6 — Структурная схема формирования дальномерного кода

Формирование дальномерного кода высокой точности

Алгоритм формирования дальномерного кода - закрытая информация.

1.2.2 L1OC, L2OC, L3OC - сигналы с кодовым разделением

L1OC - сигнал с кодовым разделением в диапазоне L1

Структура сигнала

Сигнал L1OC излучается на несущей частоте $f = 1600,995$ МГц и содержит информационную L1OCd и пилотную L1OCp компоненты равной мощности, которые уплотняются путем чередования чипов дальномерных кодов этих компонент[3].

$$S_{L1OC}(t) = A \cdot C_d \cdot O(t) \cdot D(t) \cdot M_d \cdot \cos(2\pi ft + \varphi) + A \cdot C_p \cdot B(t) \cdot M_p \cdot \cos(2\pi ft + \varphi) \quad (1.6)$$

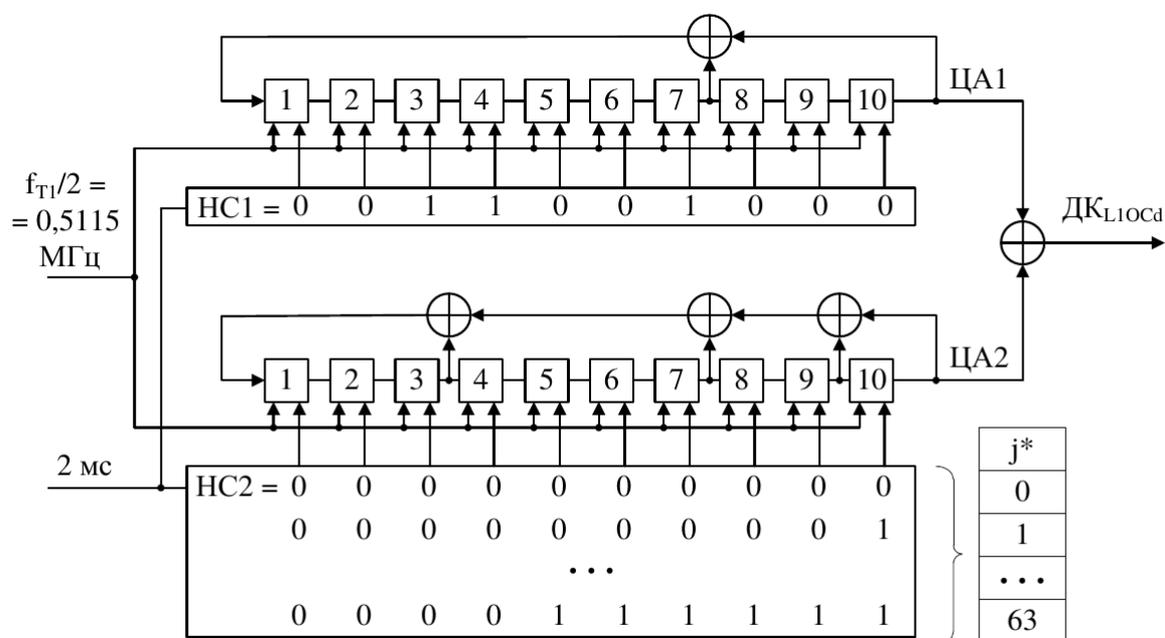
Оверлейный код представляет собой периодический код – 01 (в бинарном формате), передаваемый старшим разрядом вперед.

Формирование дальномерных кодов L1OCd

Дальномерные коды информационной компоненты L1OCd являются кодами Голда, имеют длину $N = 1023$, период $T = 2$ мс и образуются суммированием по модулю 2 двоичных символов поступающих с тактовой частотой $f = 0,5115$ МГц от двух сдвиговых регистров. Сдвиговый регистр в ЦА1 имеет 10 разрядов, номера отводов обратной связи – 7 и 10. Сдвиговый регистр в ЦА2 также имеет 10 разрядов, номера отводов обратной связи – 3, 7, 9 и 10. Сдвиг во всех регистрах осуществляется от младшего разряда к старшему.

С периодичностью 1 раз в 2 мс происходит установка следующих кодов начального состояния (НС) в регистры сдвига:

- в регистр ЦА1 код – 0011001000;
- в регистр ЦА2 код – $j = 0000000000, 0000000001 \dots 0000111111$, где j – системный номер навигационного космического аппарата (НКА) в орбитальной группировке (ОГ). Коды НС1 и НС2 (двоичные числа) записываются в ЦА1 и ЦА2 таким образом, чтобы младший разряд числа поступал в последний триггер регистра.



* $j = 0 \dots 63$ – системный номер НКА в ОГ
 НС2 = j

ЦА – цифровой автомат;
 ОГ – орбитальная группировка

Рисунок 1.7 – Структурная схема формирования дальномерного кода L1OCd

Отразим данную структуру на имеющуюся структуру универсального генератора дальномерного кода.

Для реализации ЦА1 будем использовать регистр SR_1. Обратная связь снимается с 7 и 10 разрядов сдвигового регистра, следовательно для получения конфигурационного слова code_bitmask1 необходимо в 7 и 10 разряды записать "1" , а в остальные разряды записать "0" , таким образом code_bitmask1 = 0x2400000.

Для реализации ЦА2 используем сдвиговый регистр SR_2. Обратная связь снимается с 3, 7, 9 и 10 разрядов, для получения конфигурационного слова code_bitmask2 записываем в соответствующие разряды "1" в остальные разряды записываются "0" , таким образом code_bitmask2 = 0x22C00000.

Каждые 2 мс регистры сдвига SR_1 и SR_2 устанавливаются в начальные состояния, которые задаются конфигурационными словами code_state1, code_state2 соответственно. Таким образом code_state1 = 0x32000000. Начальное состояние регистра SR_2 соответствует номеру ПСП(спутника), например, для первой ПСП начальное состояние – 0000000001, причем младший бит поступает в старший разряд регистра, поэтому code_state2 = 0x200(для первой ПСП).

Выходная последовательность снимается с 10 разряда каждого регистра, а значит в конфигурационных словах code_out_bitmask1 и code_out_bitmask2 нужно записать "1" в 10 разряд, остальные разряды "0". Далее эти последовательности поступают на общий сумматор(XOR).

Для формирования дальномерного кода информационной компоненты с помощью универсального генератора необходимо записать в его регистры следующие инициализирующие параметры, представленные в hex формате:

- code_state1 – 0x32000000,
- code_state2 – соответствует номеру ПСП(спутника),
- code_bitmask1 – 0x2400000,
- code_bitmask2 – 0x22C00000,

- code_out_bitmask1 – 0x400000,
- code_out_bitmask2 – 0x400000.

Формирование дальномерных кодов L1OCp

Дальномерные коды пилотной компоненты L1OCp являются усеченными последовательностями Касами, имеют длину $N = 4092$, период $T = 8$ мс и образуются суммированием по модулю 2 двоичных символов, поступающих с тактовой частотой $f = 0,5115$ МГц от двух сдвиговых регистров. Сдвиговый регистр в цифровом автомате ЦА1 имеет 12 разрядов, номера отводов обратной связи – 6, 8, 11 и 12. Сдвиговый регистр в ЦА2 имеет 6 триггеров, номера отводов обратной связи – 1 и 6. Сдвиг во всех регистрах осуществляется от младшего разряда к старшему.

С периодичностью 1 раз в 8 мс происходит установка следующих кодов НС в регистры ЦА:

- в ЦА1 код НС1 = 000011000101;
- в ЦА2 код НС2 = $j = 000000, 000001 \dots 111111$, где j – системный номер навигационного космического аппарата (НКА) в орбитальной группировке (ОГ).

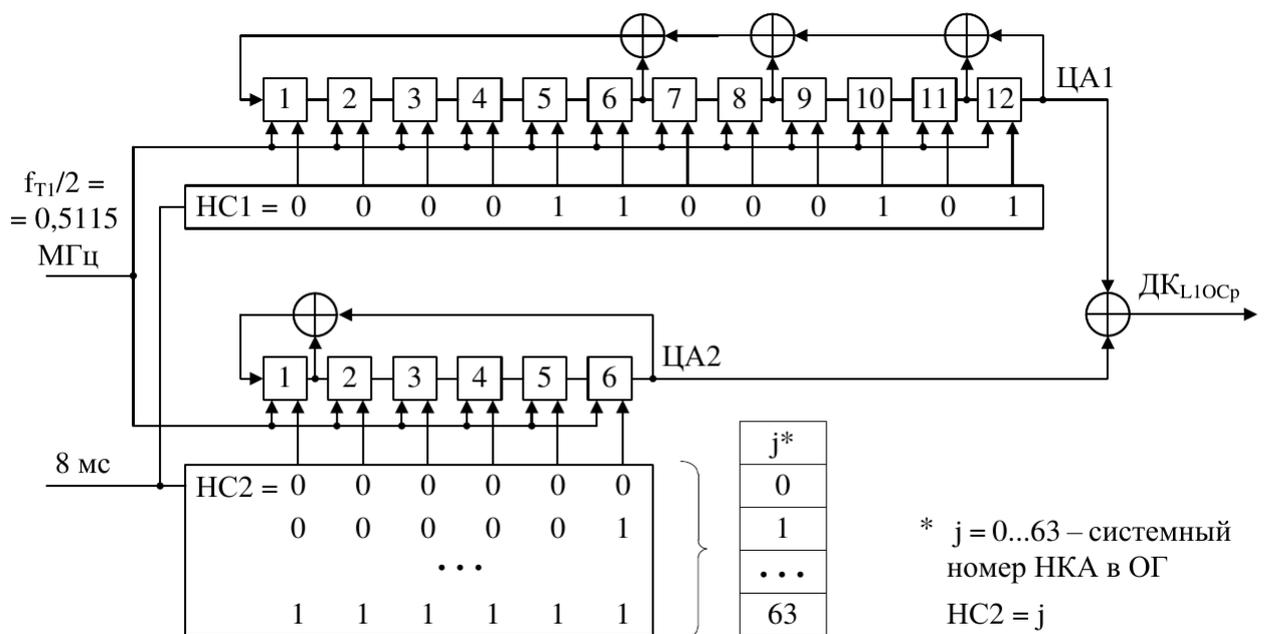


Рисунок 1.8 — Структурная схема формирования дальномерного кода L1OCp

Для формирования дальномерного кода пилотной компоненты с помощью универсального генератора необходимо записать в его регистры следующие инициализирующие параметры, полученные аналогично информационной компоненте:

- code_state1 – 0x0C500000,
- code_state2 – соответствует номеру ПСП(спутника),
- code_bitmask1 – 0x5300000,
- code_bitmask2 – 0x84000000,
- code_out_bitmask1 – 0x100000,
- code_out_bitmask2 – 0x4000000.

L2OC - сигнал с кодовым разделением в диапазоне L2

Структура сигнала

Сигнал L2OC излучается на несущей частоте $f = 1248,06$ МГц, имеет две компоненты: L2OCp и L2КСИ(которая предназначена для передачи служебной информации в диапазоне L2)с временным уплотнением.

$$S_{L2OC}(t) = A \cdot C_{КСИ}(t) \cdot D(t) \cdot M_d \cdot \cos(2\pi ft + \varphi) + A \cdot C_p \cdot B(t) \cdot O(t) \cdot M_p \cdot \cos(2\pi ft + \varphi) \quad (1.7)$$

Оверлейный код представляет собой периодический 50-символьный код – 00101101011111000110000100011010000001011000100010, передаваемый старшими разрядами вперед (левый символ в записи формируется первым по времени). Длительность символов оверлейного кода равна 20 мс [4].

Формирование дальномерных кодов сигналов L2OCp

Дальномерные коды L2OCp являются усеченными кодами Касами, имеют длину $N = 10230$, период $T = 20$ мс, и образуются суммированием по модулю 2 двоичных символов, поступающих с тактовой частотой $f = 0,5115$ МГц от цифровых автоматов ЦА1 и ЦА2. Сдвиговый регистр в ЦА1 имеет 14 разрядов,

номера отводов обратной связи – 4, 8, 13 и 14. Сдвиговый регистр в ЦА2 имеет 7 триггеров, номера отводов обратной связи – 6 и 7.

С периодичностью 1 раз в 20 мс происходит установка следующих кодов начального состояния (НС) в регистры ЦА:

- в ЦА1 код НС1 = 00110100111000;
- в ЦА2 код НС2 = $j + 64 = 1000000, 1000001 \dots 1111111$, где j – системный номер навигационного космического аппарата (НКА) в орбитальной группировке (ОГ).

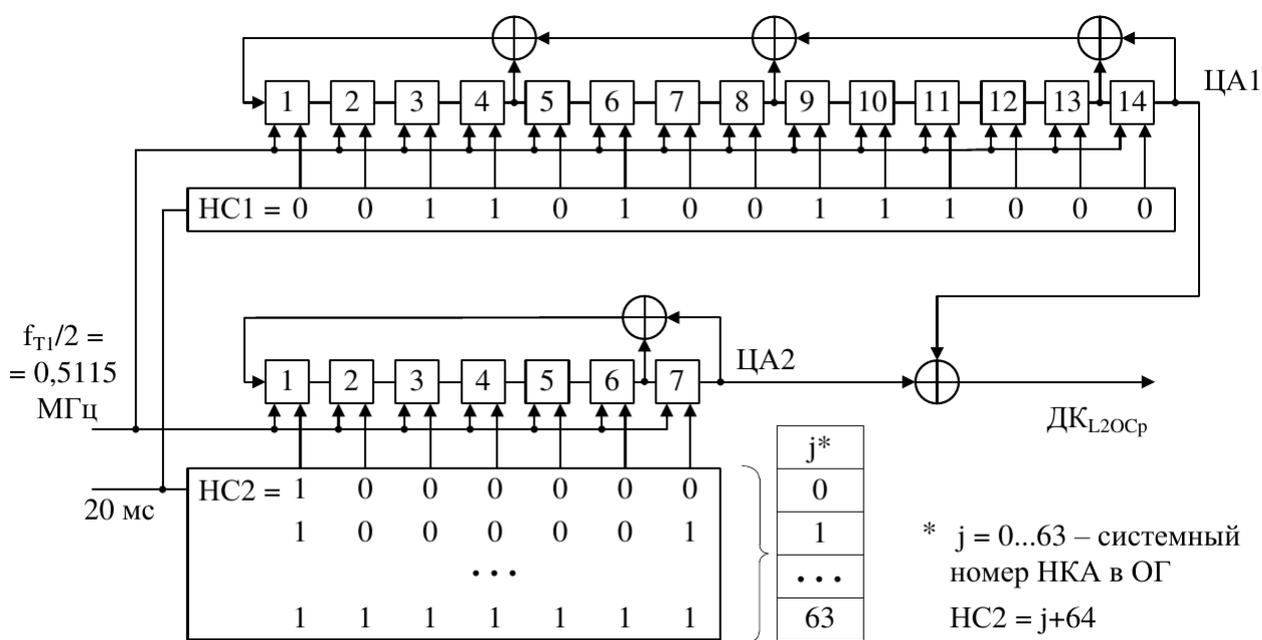


Рисунок 1.9 – Структурная схема формирования дальномерного кода L2OC

Отразим данную структуру на имеющуюся структуру универсального генератора дальномерного кода.

Для реализации ЦА1 будем использовать регистр SR_1. Обратная связь снимается с 4, 8, 13 и 14 разрядов сдвигового регистра, следовательно для получения конфигурационного слова code_bitmask1 необходимо в 4, 8, 13 и 14 разряды записать "1", а в остальные разряды записать "0", таким образом $code_bitmask1 = 0x110C0000$.

Для реализации ЦА2 используем сдвиговый регистр SR_2. Обратная связь снимается с 6 и 7 разрядов, для получения конфигурационного слова code_bit-

mask2 записываем в соответствующие разряды "1" в остальные разряды записываются "0" , таким образом code_bitmask2 = 0x6000000.

Каждые 20 мс регистры сдвига SR_1 и SR_2 устанавливаются в начальные состояния, которые задаются конфигурационными словами code_state1, code_state2 соответственно. Таким образом code_state1 = 0x34E00000. Начальное состояние регистра SR_2 соответствует номеру ПСП(спутника), например, для первой ПСП начальное состояние – 0000001, причем младший бит поступает в старший разряд регистра, поэтому code_state2 = 0x20(для первой ПСП).

Выходная последовательность снимается с 14 разряда ЦА1, а значит в слово code_out_bitmask1 нужно записать "1" в 14 разряд, остальные разряды "0" таким образом code_out_bitmask1 = 0x40000.

Выходная последовательность снимается с 7 разряда ЦА2, а значит в слово code_out_bitmask2 нужно записать "1" в 7 разряд, остальные разряды "0" , таким образом code_out_bitmask2 = 0x2000000. Далее эти последовательность поступают на общий сумматор(XOR).

Для формирования дальномерного кода пилотной компоненты с помощью универсального генератора необходимо записать в его регистры следующие инициализирующие параметры:

- code_state1 – 0x34E00000,
- code_state2 – соответствует номеру ПСП(спутника),
- code_bitmask1 – 0x110C0000,
- code_bitmask2 – 0x6000000,
- code_out_bitmask1 – 0x40000,
- code_out_bitmask2 – 0x2000000.

L3OC - сигнал с кодовым разделением в диапазоне L3

Структура сигнала

Сигнал L3OC излучается на несущей частоте $f = 1202,025$ МГц и содержит информационную L3OCd и пилотную L1OCp компоненты, которые располага-

ются на двух квадратурах I и Q (фаза L3OCp опережает фазу L3OCd на 90°). Обе компоненты имеют модуляцию BPSK(10) и одинаковую мощность [5].

$$S_{L3OC}(t) = A \cdot C_P(t) \cdot O(t) \cdot \cos(2\pi ft + \varphi) + A \cdot C_D(t) \cdot O(t) \cdot D(t) \cdot \sin(2\pi ft + \varphi) \quad (1.8)$$

Оверлейный код информационной компоненты представляет собой периодический 5-символьный код 00010 (код Баркера) передаваемый старшими разрядами вперед. Оверлейный код пилотной компоненты представляет собой периодический 10-символьный код 0000110101 (код Ньюмана-Хофмана) передаваемый старшими разрядами вперед.

Формирование дальномерных кодов

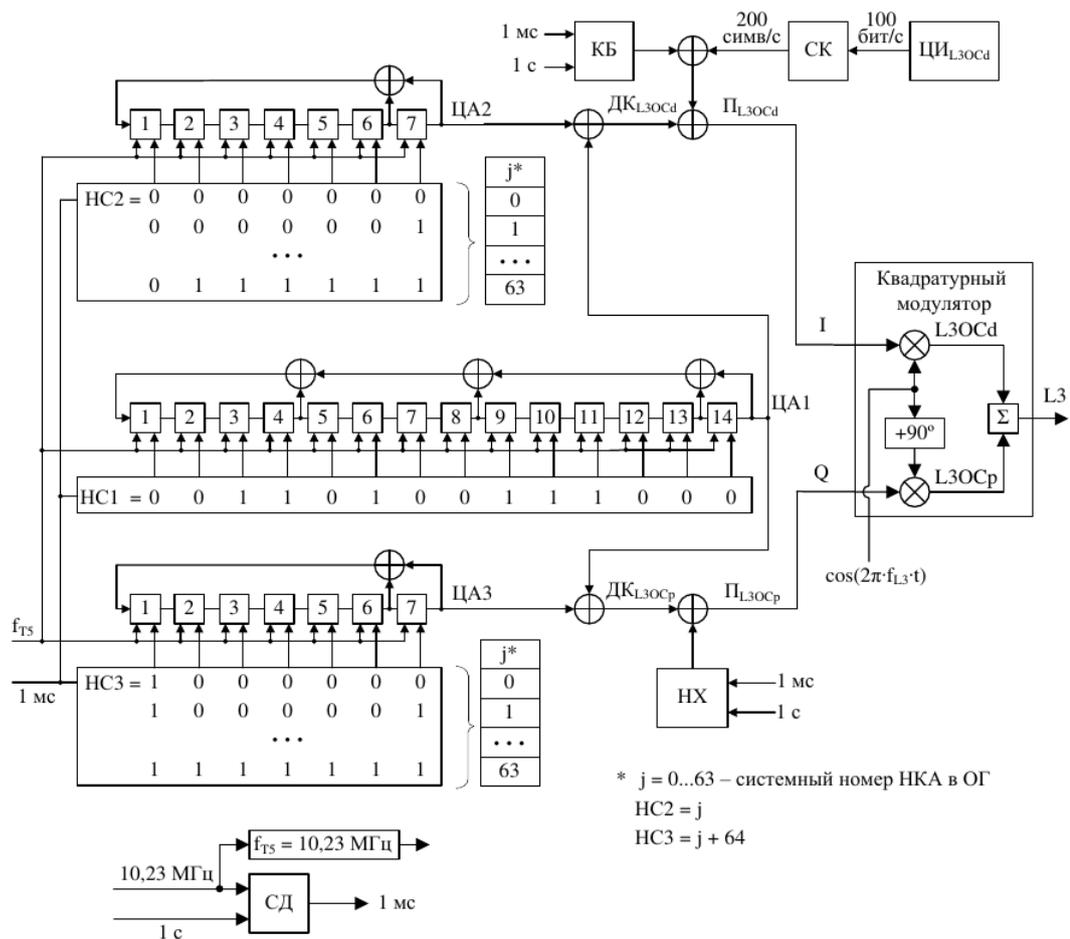
Структура генераторов ДК-L3OCd и ДК-L3OCp приведена на рисунке 1.10. Дальномерные коды ДК-L3OCd и ДК-L3OCp являются усеченными кодами Касами, имеют длину $N = 10230$, период $T = 1$ мс, и образуются суммированием по модулю 2 двоичных символов («1» и «0»), поступающих с тактовой частотой $f = 10,235$ МГц от ЦА1, ЦА2 и ЦА3 в соответствии с рисунком 1.10:

- ЦА1 и ЦА2 для формирования ДКЛ3OCd;
- ЦА1 и ЦА3 для формирования ДКЛ3OCp.

Сдвиговый регистр в ЦА1 имеет 14 разрядов, номера отводов обратной связи – 4, 8, 13, 14. Сдвиговый регистр в ЦА2 и ЦА3 имеет 7 разрядов, номера отводов обратной связи – 6 и 7.

С периодичностью 1 раз в 1 мс происходит установка следующих кодов начального состояния (НС) в регистры ЦА:

- в ЦА1 код НС1 = 00110100111000;
- в ЦА2 код НС2 = $j = 0000000, 0000001 \dots 0111111$, где j – системный номер НКА в ОГ;
- в ЦА3 код НС3 = $j + 64 = 1000000, 1000001 \dots 1111111$, где j – системный номер НКА в ОГ.



КБ – код Баркера 00010; СК – сверточный код; ЦИ – цифровая информация; ЦА – цифровой автомат;
 ДК – дальномерный код; П – последовательность; НХ – код Ньюмана-Хофмана 0000110101;
 СД – синхронный делитель; ОГ – орбитальная группировка

Рисунок 1.10 — Структурная схема формирования дальномерного кода Л3ОС

Для формирования дальномерных кодов сигнала Л3ОС с помощью универсального генератора необходимо записать в его регистры следующие инициализирующие параметры:

- code_state1 – 0x34E00000,
- code_state2 – зависит от номера ПСП(спутника),
- code_bitmask1 – 0x110C0000,
- code_bitmask2 – 0x60000000,
- code_out_bitmask1 – 0x400000,
- code_out_bitmask2 – 0x20000000.

1.3 Анализ особенностей формирования дальномерных кодов современных и перспективных навигационных сигналов системы NAVSTAR GPS

Сигналы навигационной системы GPS:

- L1 C/A – открытые сигналы в диапазоне L1
- L1C – открытые сигналы с кодовым разделением в диапазоне L1
- L2C – открытые сигналы с кодовым разделением в диапазоне L2
- L5 – открытые сигналы в диапазоне L5
- P(Y) – сигналы санкционированного доступа с модуляцией BPSK(10) в диапазонах L1, L2
- M code – сигналы санкционированного доступа с BOC модуляцией в диапазонах L1, L2

1.3.1 L1C/A, L1 P(Y), L2 P(Y)

Структура сигналов

Сигналы L1C/A, L1 P(Y) и L2 P(Y) излучаются спутниками системы GPS в двух диапазонах L1=1575,42 МГц и L2=1227,6 МГц.

$$S_{GPSL1}(t) = A \cdot C_{C/A}(t) \cdot D(t) \cdot \sin(2\pi f_{L1}t + \varphi) + A \cdot C_P(t) \cdot C_W(t) \cdot D(t) \cdot \cos(2\pi f_{L1}t + \varphi) \quad (1.9)$$

$$S_{GPSL2}(t) = A \cdot C \cdot C_W(t) \cdot D(t) \cdot \cos(2\pi f_{L2}t + \varphi) \quad (1.10)$$

где $C_W(t)$ – криптостойкая ПСП.

Эти сигналы модулируются дальномерными кодами трех видов:

- Открытый код C/A
- P-код

- Y-код

Открытый код C/A (Coarse/Acquisition)

Этот код сначала использовался лицензированными пользователями для первичного вхождения в режим слежения и последующего захвата точного P или P(Y) кода. Сейчас код C/A находится в распоряжении мирового сообщества для использования в целях позиционирования.

P-код

Основной дальномерный код, индивидуальный для каждого НКА: $P_i(t)$, где i – индивидуальный номер НКА. $P_i(t)$ представляет собой последовательность символов, передающихся с частотой $f = 10,23$ МГц, и имеет период $T = 7$ дней. Эта последовательность формируется сложением по модулю 2 двух подпоследовательностей, обозначаемых, как X_1 и X_{2i} , их длина соответственно 15345000 и 15345037 элементов. Последовательность X_{2i} формируется из последовательности X_2 избирательной задержкой на длительность от 1 до 37 элементов [6].

Y-код

Применяется вместо P-кода при включении режима предотвращения преднамеренных помех и несанкционированного доступа к информации A/S (Antispoofing). Y-код представляет собой закрытый P-код, поэтому Y-код принято обозначать как P(Y).

Формирование дальномерных кодов

Формирование C/A кода:

Дальномерный код C/A является кодом Голда, имеет длину $N = 1023$, период $T = 1$ мс и образуется суммированием по модулю 2 двоичных символов, поступающих с тактовой частотой $f = 1,023$ МГц от двух сдвиговых регистров. Последовательность G_{2i} , формируется путем задержки G_2 на целое число тактов. Образующие полиномы для G_1 и G_2 описываются следующими выражениями:

$$G_1(x) = 1 + x^3 + x^{10} \quad (1.11)$$

$$G_2(x) = 1 + x^2 + x^3 + x^6 + x^8 + x^9 + x^{10} \quad (1.12)$$

Сдвиговые регистры G1 и G2 инициализируются единицами по началу эпохи X1. Инициализация регистров в фазе с эпохами X1 гарантирует, что первый бит С/А-кода начнется синхронно с первым битом Р-кода.

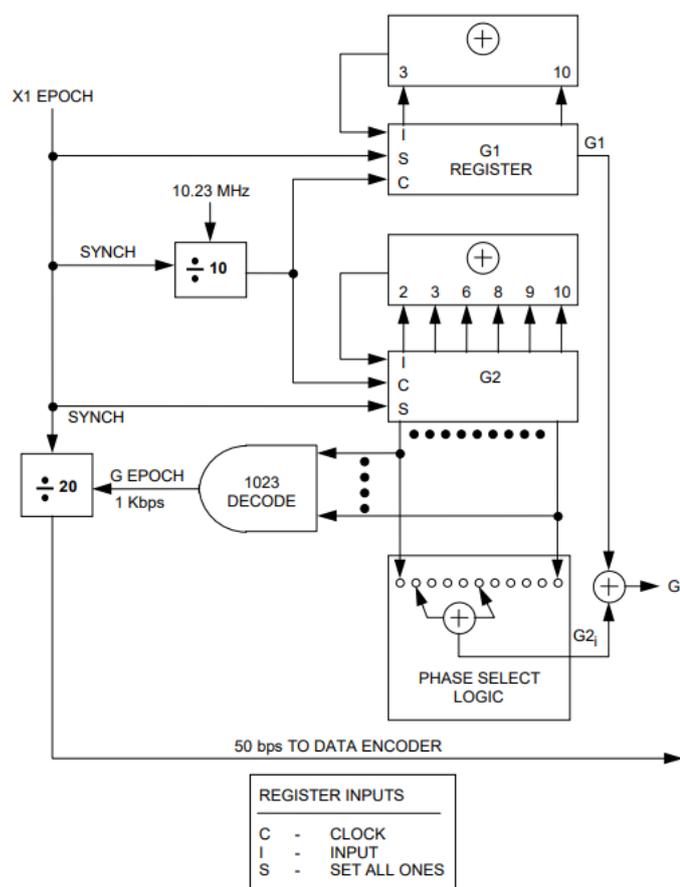


Рисунок 1.11 — Структурная схема формирования С/А кода

Формирование Р-кода:

Дальномерный Р-код является суммой по модулю 2 двух последовательностей X1 и X2_i тактируемых с частотой 10,23 МГц. В свою очередь, X1 формируется сложением по модулю 2 выходных сигналов двух 12-разрядных регистров сдвига (X1A и X1B) генерирующих сокращенные последовательности длиной 4092 и 4093 бит соответственно. Когда число кадров X1 достигает 3750, генерируется признак конца эпохи X1. Эпоха X1 генерируется каждые 1.5 с, когда

сформирована последовательность X1 из 15345000 разрядов. Образующие полиномы для последовательностей X1A и X1B, относящиеся к соответствующим регистрам сдвига, имеют вид:

$$X1A = 1 + X^6 + X^8 + X^{11} + X^{12} \quad (1.13)$$

$$X1B = 1 + X^1 + X^2 + X^5 + X^8 + X^9 + X^{10} + X^{11} + X^{12} \quad (1.14)$$

В начале каждой эпохи X1, сдвиговый регистр X1A инициализируется кодовым вектором 001001001000 и сдвиговый регистр X1B инициализируется кодовым вектором 010101010100. Первый разряд последовательности X1A и первый разряд последовательности X1B формируются одновременно, во время первого такта каждого периода X1. Для формирования последовательности $X2_i$ сначала формируется последовательность X2, которая затем задерживается на целое число тактов $i = 1...37$. Каждая последовательность $X2_i$ суммируется по модулю 2 с последовательностью X1. Образующие полиномы для регистров X2A и X2B, имеют вид:

$$X2 = 1 + X^1 + X^3 + X^4 + X^5 + X^7 + X^8 + X^9 + X^{10} + X^{11} + X^{12} \quad (1.15)$$

$$X2 = 1 + X^2 + X^3 + X^4 + X^8 + X^9 + X^{12} \quad (1.16)$$

Регистр X2A инициализируется вектором 100100100101, регистр X2B вектором 010101010100. Когда начинается новая неделя GPS, сдвиговые регистры X1A, X1B, X2A и X2B инициализируются для выработки первого разряда недели.

Схемы формирования кодовых последовательностей показаны на рис. 1.12 – 1.15.

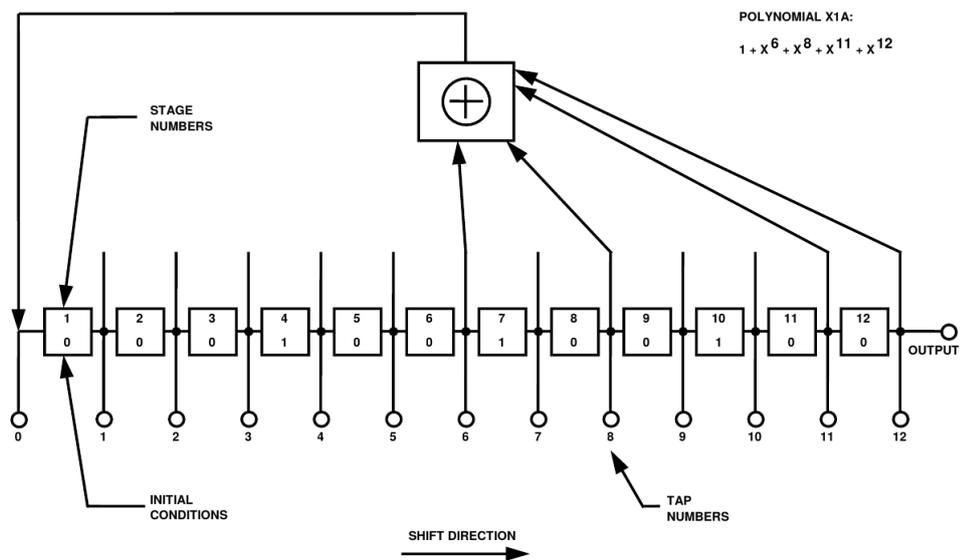


Рисунок 1.12 — Структурная схема формирования последовательности X1A

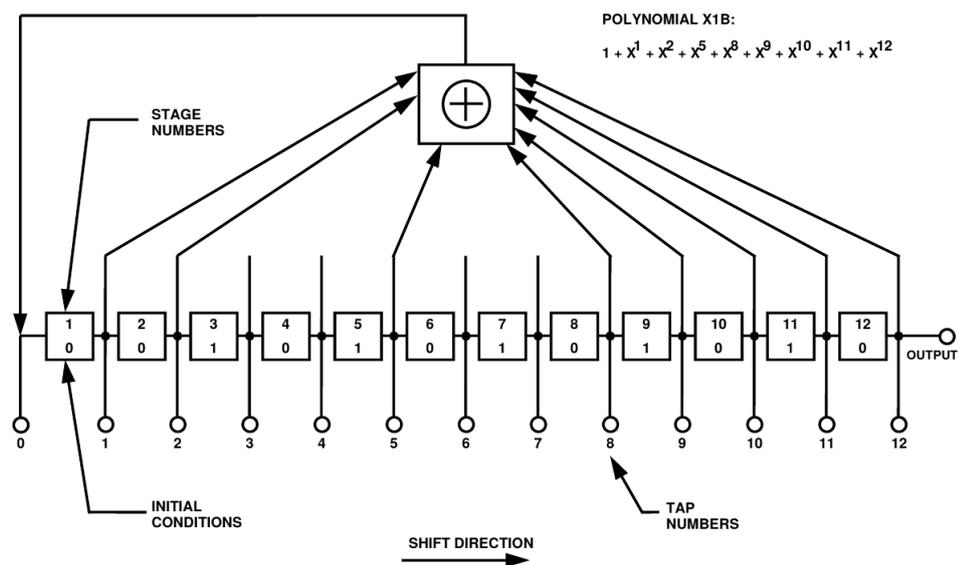


Рисунок 1.13 — Структурная схема формирования последовательности X1B

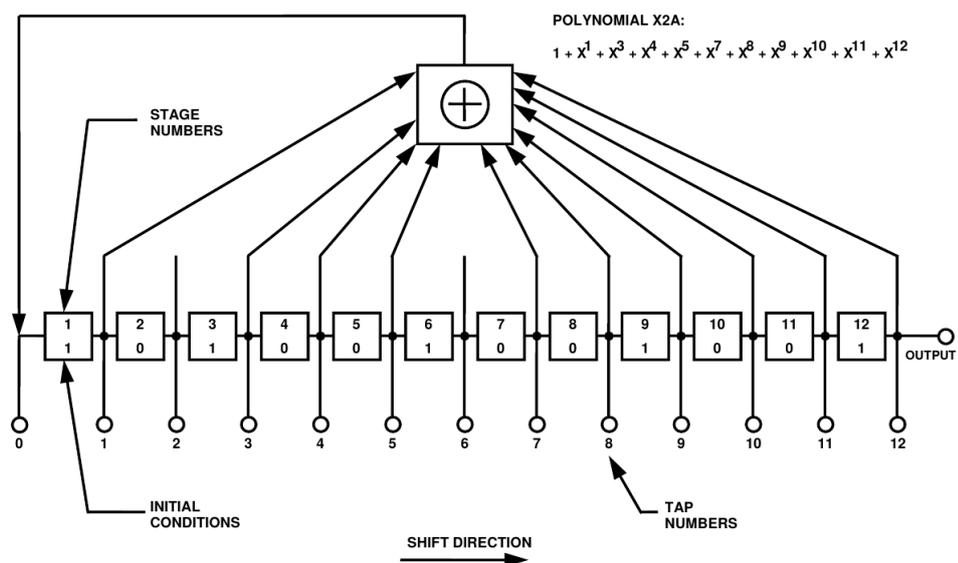


Рисунок 1.14 — Структурная схема формирования последовательности X2A

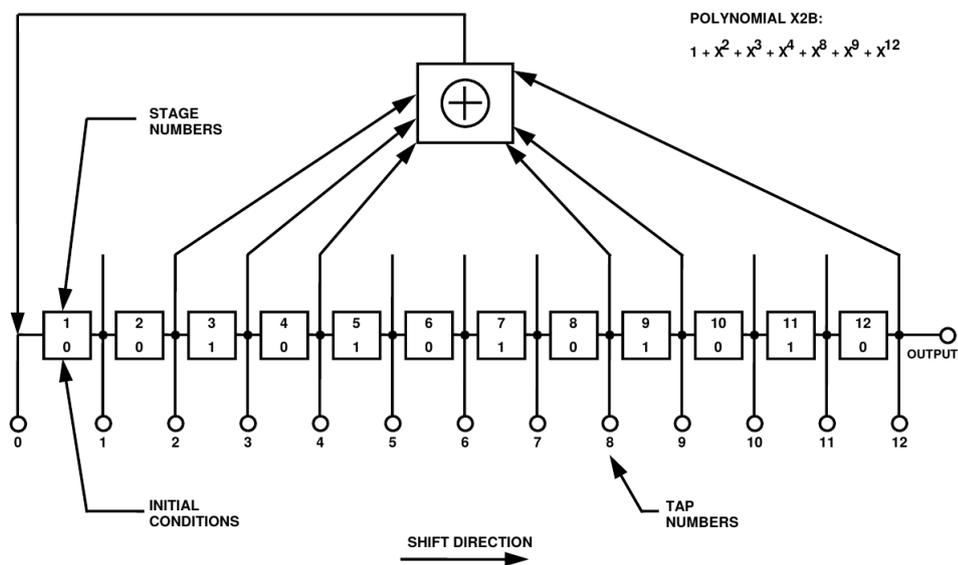


Рисунок 1.15 — Структурная схема формирования последовательности X2B

Формирование P кода невозможно с помощью универсального генератора дальномерных кодов, рассматриваемого в работе, так как для получения такой последовательности необходимо 4 регистра сдвига.

Формирование P(Y) кода:

$$P(Y)(t) = P(t) \oplus W(t), \quad (1.17)$$

где W(t) - криптостойкая последовательность

1.3.2 L1C, L2C, L5 – открытые перспективные сигналы

L1C - открытый сигнал в диапазоне L1

Структура сигнала

Сигнал L1C излучается на несущей частоте $f = 1575,42$ МГц и содержит две компоненты: информационную L1Cd и пилотную L1Cp. Уплотнение компонент – синфазная сумма [7].

$$S_{L1}(t) = A \cdot C_d \cdot D(t) \cdot B(t) \cdot \cos(2\pi ft + \varphi) + A \cdot C_p \cdot O(t) \cdot B_{TMBOC}(t) \cdot \cos(2\pi ft + \varphi), \quad (1.18)$$

где B_{TMBOC} – сложная модуляция TMBOC(BOC(1,1) и BOC(6,1))

Оверлейный код является укороченным кодом Голда, имеет период $T = 18$ мс, длину $N = 1800$, символьную скорость $f = 100$ бит/с.

Формирование дальномерного кода

Дальномерные коды сигнала Gps L1C являются кодами Вейла, имеют длину $N = 10230$, период $T = 10$ мс, символьную скорость $f = 1,023$ Мбит/с.

Формирование таких кодов с помощью генератора, рассматриваемого в работе, невозможно. Поэтому будем считать такие коды табличными.

L2C - открытый сигнал в диапазоне L2

Структура сигнала

Сигнал L2C излучается на несущей частоте $f = 1227.6$ МГц и содержит две компоненты: информационную L2 CM и пилотную L2 CL, которые имеют временное уплотнение и модуляцию BPSK(10) [6].

$$S_{L2}(t) = A \cdot C_{CM} \cdot M_{CM} \cdot D(t) \cdot \cos(2\pi ft + \varphi) + A \cdot C_{CL} \cdot M_{CL} \cdot \cos(2\pi ft + \varphi) \quad (1.19)$$

Формирование дальномерного кода

Дальномерный код информационной компоненты L2 CM имеет длину $N = 10230$, период $T = 20$ мс, символьную скорость $f = 0.5115$ Мбит/с.

Дальномерный код пилотной компоненты L2 CL имеет длину $N = 767250$, период $T = 1.5$ с, символьную скорость $f = 0.5115$ Мбит/с.

Коды сигналов L2 CM и L2 CL формируются по одной схеме, являются укороченными M-последовательностями. Для формирования такой последовательности регистр CM сбрасывается в начальное состояние после 10230 бита результирующего кода, что соответствует периоду 20 мс, регистр CL сбрасывается после 767250 бита. Начальные состояния генератора зависят от номера ПСП и заданы в ИКД. Схема генератора для формирования дальномерных кодов сигнала Gps L2C, предложенная в ИКД представлена на рис. 1.16.

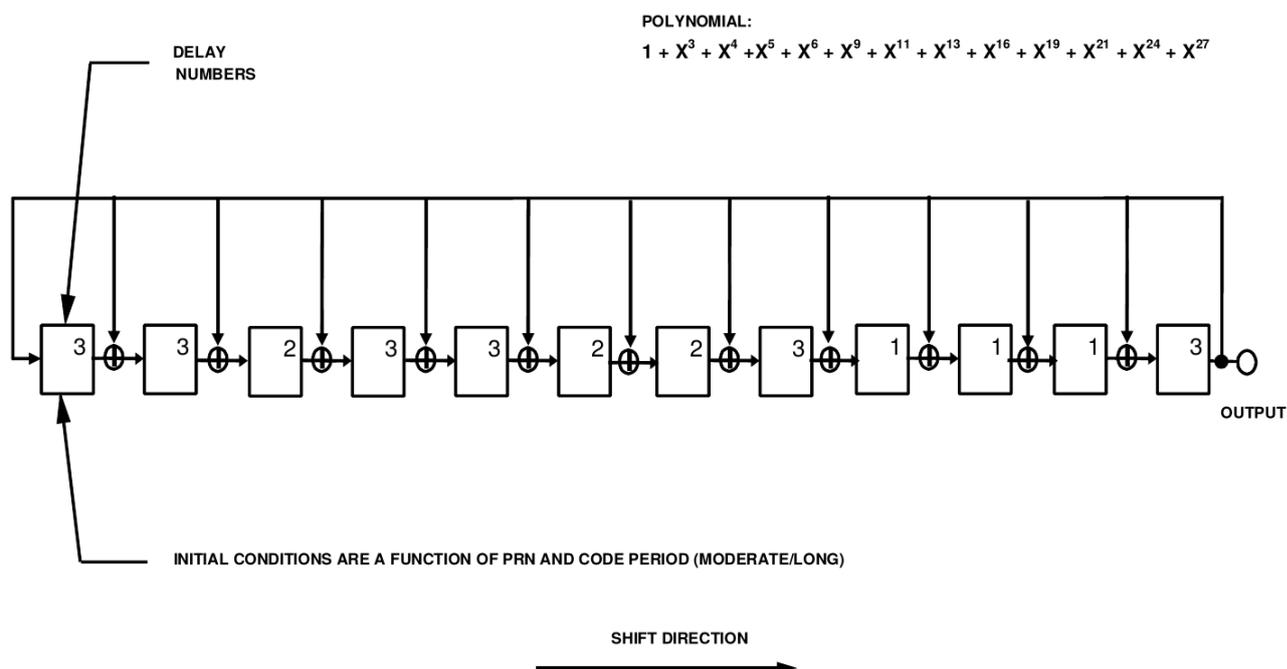


Рисунок 1.16 — Структурная схема формирования дальномерных кодов L2 CM/CL

Генератор дальномерного кода в ИКД представлен в конфигурации Галуа, а универсальный генератор, рассматриваемый в работе, имеет конфигурацию Фибоначчи. Но так как дальномерный код является усеченной M-последовательностью, то для нахождения начального состояния регистра сдвига, то есть значения конфигурационного слова `code_state1`, необходимо найти нужную фазу этой последовательности. Результирующая последовательность снимается с последнего разряда регистра, а значит первые 27 бит и будут нужным началь-

ным состоянием(`code_state1`) регистра сдвига генератора, рассматриваемого в работе.

Найденные начальные состояния регистра сдвига SR1 в шестнадцатиричном виде приведены в таблице 1.4.

Таблица 1.4 — Начальные состояния регистра сдвига для сигнала Gps L2C

№ ПСП	CM code_state1	CL code_state1
1	0x015EF0F5	0x029BE220
2	0x050F811E	0x02012ED7
3	0x010E553D	0x03D7D64B
4	0x016B0258	0x012B1C4A
5	0x0416F3BC	0x05E3A308
6	0x065BC21E	0x031C0719
7	0x00F5BE58	0x03B5179F
8	0x0496777F	0x074429A6
9	0x04A5A8E2	0x01D5FC3B
10	0x036E44D6	0x03BF943A
11	0x05E84705	0x0587C624
12	0x0345EA19	0x00BE84CE
13	0x06965B5B	0x057D8717
14	0x0447FB02	0x06A8376F
15	0x00043A6E	0x05A13F5D
16	0x035E5896	0x04A5F5DF
17	0x03059DDD	0x0046B92B
18	0x05C16D2A	0x07A7C2AE
19	0x010C80DB	0x045886A6
20	0x01C754B4	0x05A9A643
21	0x0650324E	0x068872F2
22	0x07FB4E14	0x03E759F6
23	0x074E048F	0x06B6FDBD
24	0x00663507	0x031B717B
25	0x01F887F9	0x0048FCB0
26	0x0487C247	0x01CBC9E3
27	0x05FD6D8C	0x06B38D5B

продолжение на следующей странице...

Таблица 1.4 – продолжение таблицы

№ ПСП	CM code_state1	CL code_state1
28	0x020818D1	0x06F5B8FA
29	0x01ECE400	0x0121A76E
30	0x07AEB923	0x05F23C35
31	0x0656B597	0x0326FD21
32	0x0602E157	0x03CB4E3C

Отразим данную структуру на имеющуюся структуру универсального генератора дальномерного кода.

Для формирования дальномерного кода сигнала L2C с помощью универсального генератора используется только один регистр сдвига, реализуем его с помощью регистра сдвига SR_1. Регистр SR_2 и его конфигурационные слова использовать не будем(во все разряды нужно записать "0").

Отводы на обратную связь определяются полиномом:

$$G(X) = 1 + X^3 + X^4 + X^5 + X^6 + X^9 + X^{11} + X^{13} + X^{16} + X^{19} + X^{21} + X^{24} + X^{27} \quad (1.20)$$

Для формирования слова code_bitmask1 необходимо записать в 3, 4, 5, 6, 9, 11, 13, 16, 19, 21, 24, 27 разряды "1" , а в остальные разряды "0" . А значит конфигурационное слово code_bitmask1 = 0x3CA92920.

Выходная последовательность снимается с последнего разряда, соответственно нужно записать "1" в 27 разряд регистра сдвига, таким образом code_out_bitmask1 = 0x20.

Для формирования дальномерных кодов информационной компоненты с помощью универсального генератора необходимо записать в его регистры следующие инициализирующие параметры:

- code_state1 – зависит от номера ПСП и приведены в в таб. 1.4,
- code_state2 – 0x00,
- code_bitmask1 – 0x3CA92920,
- code_bitmask2 – 0x00,

- code_out_bitmask1 – 0x20,
- code_out_bitmask2 – 0x00.

L5 - открытый сигнал в диапазоне L5

Структура сигнала

Сигнал L5 излучается на несущей частоте $f = 1176.45$ МГц и содержит информационную I5 и пилотную Q5 компоненты, которые располагаются на двух квадратурах I и Q. Обе компоненты имеют модуляцию BPSK(10) и одинаковую мощность [8].

$$S_{L5}(t) = A \cdot C_{I5}(t) \cdot O(t) \cdot D(t) \cdot \cos(2\pi ft + \varphi) + A \cdot C_{Q5}(t) \cdot O(t) \cdot \sin(2\pi ft + \varphi) \quad (1.21)$$

Оверлейный код информационной компоненты представляет собой периодический код 0000110101 (код Неймана-Хоффмана). Оверлейный код пилотной компоненты представляет собой периодический код 00000100110101001110 (код Неймана-Хоффмана).

Формирование дальномерного кода

Дальномерный код имеет длину $N = 10230$, период $T = 1$ мс, символьную скорость $f = 10.23$ Мбит/с. Псевдослучайная последовательность формируется сложением по модулю 2 двух последовательностей XA и XBI(или XBQ). Последовательность XA имеет длину 8190 бит, начальное состояние регистра XA – "1" во всех разрядах. Сброс регистра XA происходит попеременно либо при достижении состояния – 111111111101, либо при достижении счетчика символов значения 10230. Регистр XBI(или XBQ) устанавливаются в начальное состояние каждую миллисекунду. Начальные состояния регистров XBI и XBQ зависят от номера ПСП и заданы в ИКД. [8]

Отводы на обратную связь регистров сдвига задаются полиномами:

$$XA = 1 + X + X^9 + X^{10} + X^{12} + X^{13} \quad (1.22)$$

$$XBI(XBQ) = 1 + X + X^3 + X^4 + X^6 + X^7 + X^8 + X^{12} + X^{13} \quad (1.23)$$

Схема генератора дальномерного кода, предложенная в ИКД на рис. 1.17

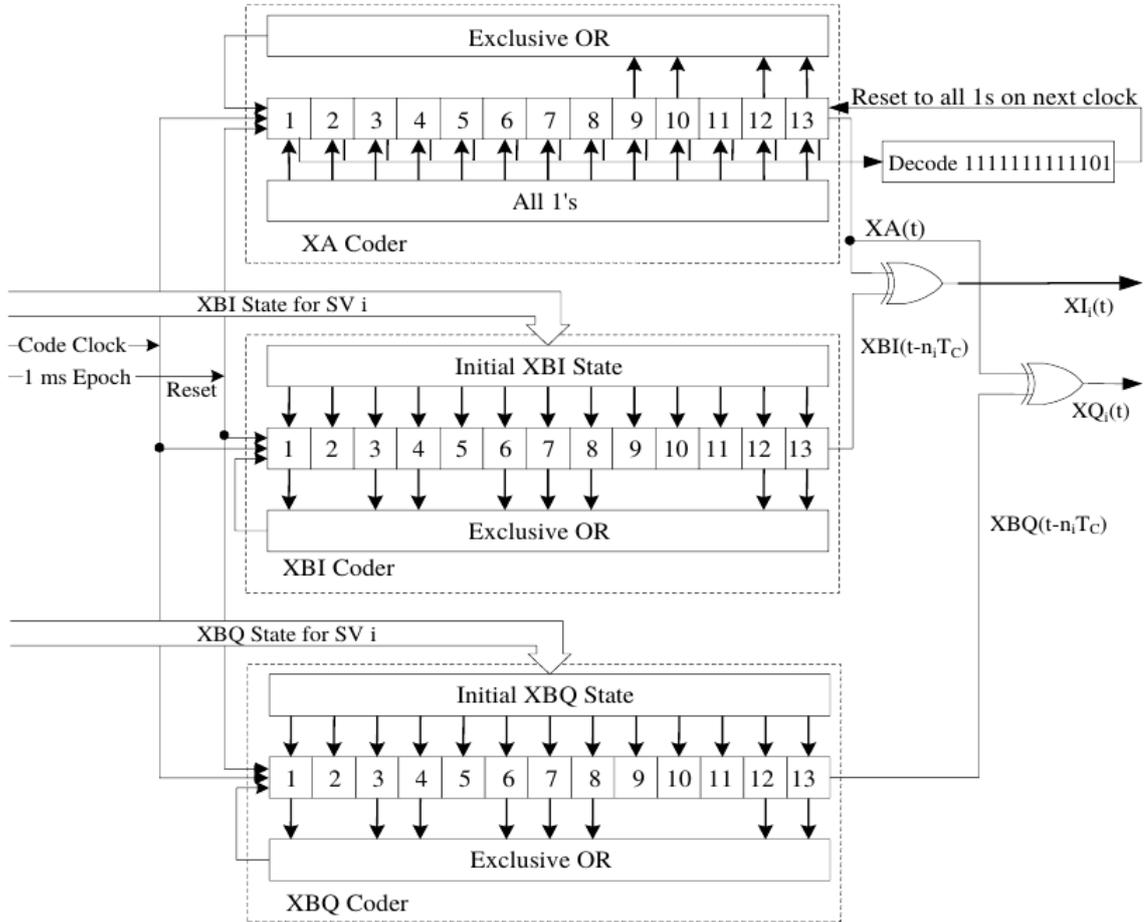


Рисунок 1.17 — Структурная схема формирования дальномерного кода L5

Отразим данную структуру на имеющуюся структуру универсального генератора дальномерного кода.

Для реализации XA будем использовать регистр SR_1. Обратная связь снимается с 9, 10, 12 и 13 разрядов сдвигового регистра, следовательно для получения конфигурационного слова `code_bitmask1` необходимо в 9, 10, 12 и 13 разряды записать "1", а в остальные разряды записать "0", таким образом $code_bitmask1 = 0x00D80000$.

Для формирования дальномерного кода информационной компоненты используется регистр XBI, для пилотной компоненты XBQ. Эти регистры отличаются только начальными состояниями. Для реализации $XBI(XBQ)$ используем

сдвиговый регистр SR_2. Обратная связь снимается с 1, 3, 4, 6, 7, 8, 12, 13 разрядов, для получения конфигурационного слова code_bitmask2 записываем в соответствующие разряды "1" в остальные разряды записываются "0", таким образом code_bitmask2 = 0xB7180000.

Начальные состояния задаются конфигурационными словами code_state1, code_state2 соответственно. Начальное состояние регистра SR_1 соответствует "1" во всех разрядах, таким образом code_state1 = 0xFFF80000. Начальное состояние регистра SR_2 зависит от номера ПСП(спутника).

Выходная последовательность снимается с 13 разряда каждого регистра, а значит в конфигурационных словах code_out_bitmask1 и code_out_bitmask2 нужно записать "1" в 13 разряд, остальные разряды "0" таким образом code_out_bitmask1 = 0x80000 и code_out_bitmask2 = 0x80000. Далее эти последовательности поступают на общий сумматор(XOR).

Для формирования дальномерных кодов сигнала L5 с помощью универсального генератора необходимо записать в его регистры следующие инициализирующие параметры:

- code_state1 – 0xFFF80000,
- code_state2 – зависит от номера ПСП(спутника),
- code_bitmask1 – 0x00D80000,
- code_bitmask2 – 0xB7180000,
- code_out_bitmask1 – 0x80000,
- code_out_bitmask2 – 0x80000.

1.4 Анализ особенностей формирования дальномерных кодов современных и перспективных навигационных сигналов системы Galileo

Сигналы навигационной системы Galileo:

- E1-B, E1-C – сигналы с табличным дальномерным кодом в диапазоне E1
- E6-B, E6-C – сигналы с табличным дальномерным кодом в диапазоне E6
- E5 – сигнал с модуляцией AltBOC:
 - E5a – сигнал с кодовым разделением внизу диапазона E5
 - E5b – сигнал с кодовым разделением вверху диапазона E5

1.4.1 E5 - сигнал с кодовым разделением

Структура сигнала

Радиосигналы Galileo E5a и E5b образуют единый сигнал Galileo E5 с модуляцией AltBOC(15,10), в котором уплотнено 4 сигнала: E5a-I, E5a-Q, E5b-I, E5b-Q. Спектр такого сигнала имеет два главных лепестка, причем левый лепесток спектра содержит только радиосигналы E5a-I и E5a-Q, которые имеют квадратурное уплотнение и модуляцию BPSK(10), а правый лепесток содержит только сигналы E5b-I и E5b-Q с квадратурным уплотнением и модуляцией BPSK(10) [9]. Сигнал Galileo E5a излучается на несущей частоте $f = 1176.45\text{МГц}$, сигнал Galileo E5b на несущей частоте $f = 1207.14\text{МГц}$.

$$S_{E5a}(t) = A \cdot C_{E5aI}(t) \cdot O(t) \cdot D(t) \cdot \sin(2\pi ft + \varphi) + A \cdot C_{E5aQ}(t) \cdot O(t) \cdot \cos(2\pi ft + \varphi) \quad (1.24)$$

$$S_{E5b}(t) = A \cdot C_{E5bI}(t) \cdot O(t) \cdot D(t) \cdot \sin(2\pi ft + \varphi) + A \cdot C_{E5bQ}(t) \cdot O(t) \cdot \cos(2\pi ft + \varphi) \quad (1.25)$$

Оверлейный код зависит от номера ПСП(спутника) и от типа сигнала. Все оверлейные коды представлены в ИКД.

Формирование дальномерного кода

Для формирования дальномерных кодов может быть использован один из методов: табличный или на сдвиговых регистрах. В ИКД приведены дальномерные коды, заданные таблично для каждой из компонент [10]. Для формирования дальномерных кодов на сдвиговых регистрах используется следующая схема 1.18.

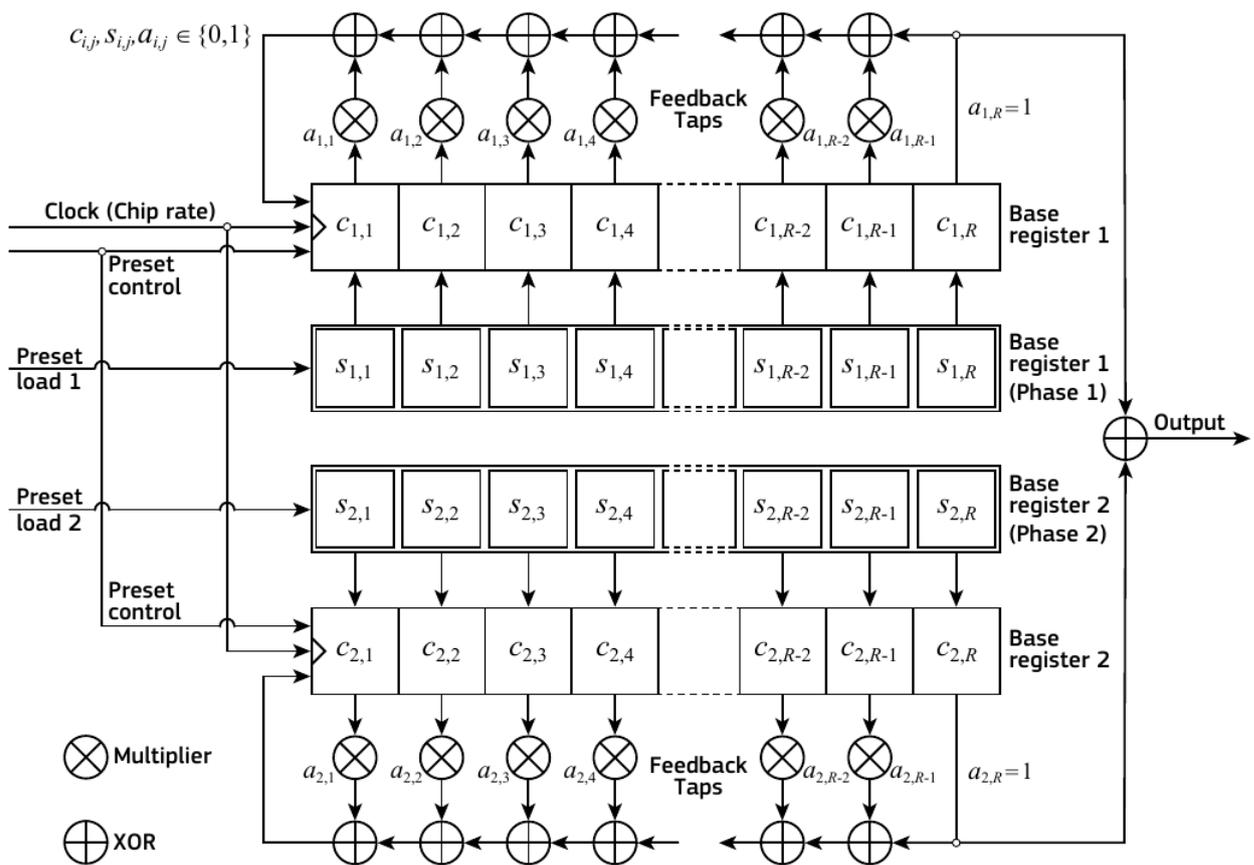


Рисунок 1.18 — Структурная схема формирования дальномерного кода E5

Дальномерные коды всех 4-х компонент являются усеченными кодами Голда, имеют длину $N = 10230$, период $T = 1$ мс и образуются суммированием по модулю 2 двоичных символов поступающих с тактовой частотой $f = 10,23$ МГц от двух сдвиговых регистров. Оба сдвиговых регистра имеют 14 разрядов, номера отводов обратной связи заданы в таблице 1.19. Сдвиг во всех регистрах

осуществляется от младшего разряда к старшему.

С периодичностью 1 раз в 1 мс происходит установка следующих кодов начального состояния (НС) в регистры сдвига:

- в первый регистр код – 11111111111111 ("все единицы");
- во второй регистр код – определяется в соответствии с ИКД.

Component	Shift Register Length (polynomial order)	Feedback Taps (octal)	
		Register 1	Register 2
E5a-I	14	40503	50661
E5a-Q	14	40503	50661
E5b-I	14	64021	51445
E5b-Q	14	64021	43143

Рисунок 1.19 – Номера отводов обратной связи для сигнала E5

Для формирования дальномерного кода сигнала E5a с помощью универсального генератора необходимо записать в его регистры следующие инициализирующие параметры:

- code_state1 – 0xFFFC0000,
- code_state2 – зависит от номера ПСП(спутника) и представлен в ИКД,
- code_bitmask1 – 0x85040000,
- code_bitmask2 – 0x1B140000,
- code_out_bitmask1 – 0x40000,
- code_out_bitmask2 – 0x40000.

Для формирования дальномерного кода сигнала E5b с помощью универсального генератора необходимо записать в его регистры следующие инициализирующие параметры:

- code_state1 – 0xFFFC0000,
- code_state2 – зависит от номера ПСП(спутника) и представлен в ИКД,
- code_bitmask1 – 0x102C0000,
- code_bitmask2 – 0x49940000,
- code_out_bitmask1 – 0x40000,

- `code_out_bitmask2` – 0x40000.

1.4.2 E1, E6 - сигналы с кодовым разделением

Структура сигнала

Сигнал E1 излучается на несущей частоте $f = 1575,42$ МГц и имеет две компоненты. Сигнал E6 на частоте $f = 1278,75$ МГц.

- E1-B информационный радиосигнал
- E1-C пилотный радиосигнал
- E6-B информационный радиосигнал
- E6-C пилотный радиосигнал

Сигналы E1-B, E1-C имеют составной вид модуляции “Composite BOC” (СВОС). Суть такого вида модуляции заключается в том, что каждый бит дальномерного кода модулируется суммой двух цифровых поднесущих, соответствующих функциям $\text{BOC}(1, 1)$ и $\text{BOC}(6, 1)$. В суммарном радиосигнале, присутствует как компонента с модуляцией $\text{BOC}(1, 1)$, так и компонента с модуляцией $\text{BOC}(6, 1)$. Эти компоненты могут приниматься как совместно, так и отдельно [9]. Сигнал E6 имеет модуляцию $\text{BPSK}(5)$.

Формирование дальномерного кода

Дальномерный код сигнала E1 является оптимизированной квазислучайной последовательностью, имеет длину $N = 4092$, период $T = 4$ мс, частоту выборки символов $f = 10,23$ МГц. Формирование осуществляется табличным методом. Дальномерный код сигнала E6 также является табличным. Все дальномерные коды приведены в ИКД.

Формирование табличных дальномерных кодов с помощью универсального генератора

Дальномерный код сигнала Gal E1 имеет длину $N = 4092$, является квазислучайной последовательностью и в ИКД задан таблично.

Проведем оценку возможности формирования такого кода с помощью универсального генератора на сдвиговых регистрах.

Универсальный генератор дальномерного кода состоит из шести 32 битных сдвиговых регистров, следовательно может иметь $2^{(32 \cdot 6)}$ начальных состояний. А значит, с помощью такого генератора возможно сформировать 2^{192} различных последовательностей.

Дальномерный код сигнала Galileo E1 имеет длину 4092 бита, следовательно такую последовательность можно сформировать 2^{4092} способами.

Вероятность того, что на основе универсального генератора на сдвиговых регистрах возможно формирование дальномерного кода сигнала Galileo E1:

$$P = \frac{2^{192}}{2^{4092}} = 2^{-3900}$$

Вероятность успешного формирования очень мала, поэтому в данной работе решением этой задачи заниматься не будем. Для формирования дальномерного кода сигнала Galileo E1 будем использовать дополнительный блок.

1.5 Анализ особенностей формирования дальномерных кодов современных и перспективных навигационных сигналов системы Beidou

Сигналы навигационной системы Beidou:

- B1I, B1Q – сигналы с кодовым разделением в диапазоне B1
- B1C – сигнал с табличным дальномерным кодом в диапазоне B1
- B2I, B2Q – сигналы с кодовым разделением в диапазоне B2
- B2ad, B2ar – сигналы с кодовым разделением в диапазоне B2
- B3I – сигнал с кодовым разделением в диапазоне B3

1.5.1 B1, B2 - сигналы с кодовым разделением

Структура сигналов

Сигнал B1 излучается на несущей частоте $f = 1561,098$ МГц, сигнал B2 на несущей частоте $f = 1207,140$ МГц. Каждый сигнал содержит открытую и военную компоненты, которые располагаются на двух квадратурах I и Q. Сигналы имеют модуляцию QPSK. [11].

- B1I, B2I – гражданский дальномерный код (2 мс);
- B1Q, B2Q – военный дальномерный код;

Модели сигналов:

$$S_{B1}(t) = A \cdot C_{B1I} \cdot D(t) \cdot \cos(2\pi ft + \varphi_{B1I}) + A \cdot C_{B1Q}(t) \cdot D(t) \cdot \sin(2\pi ft + \varphi_{B1Q}) \quad (1.26)$$

$$S_{B2}(t) = A \cdot C_{B2I} \cdot D(t) \cdot \cos(2\pi ft + \varphi_{B2I}) + A \cdot C_{B2Q}(t) \cdot D(t) \cdot \sin(2\pi ft + \varphi_{B2Q}) \quad (1.27)$$

Формирование дальномерного кода

Дальномерные коды B1I и B2I являются кодами Голда, имеют длину $N = 2046$, период $T = 1$ мс и образуются суммированием по модулю 2 двоичных символов

поступающих с тактовой частотой $f = 2,046\text{МГц}$ от двух сдвиговых регистров. Оба сдвиговых регистра имеют 11 разрядов, номера отводов обратной связи задаются полиномами:

Для первого регистра – $G1(X) = 1 + X + X^7 + X^8 + X^9 + X^{10} + X^{11}$

Для второго регистра – $G2(X) = 1 + X + X^2 + X^3 + X^4 + X^5 + X^8 + X^9 + X^{11}$

С периодичностью раз в 1 мс происходит установка следующих начальных состояний регистров сдвига:

- начальное состояние первого регистра – 01010101010,
- начальное состояние второго регистра – 01010101010.

Последовательность символов первого регистра снимается с 11 разряда. Для второго регистра номера отводов на выход зависят от номера ПСП(спутника) и представлены в ИКД. Сдвиг во всех регистрах осуществляется от младшего разряда к старшему.

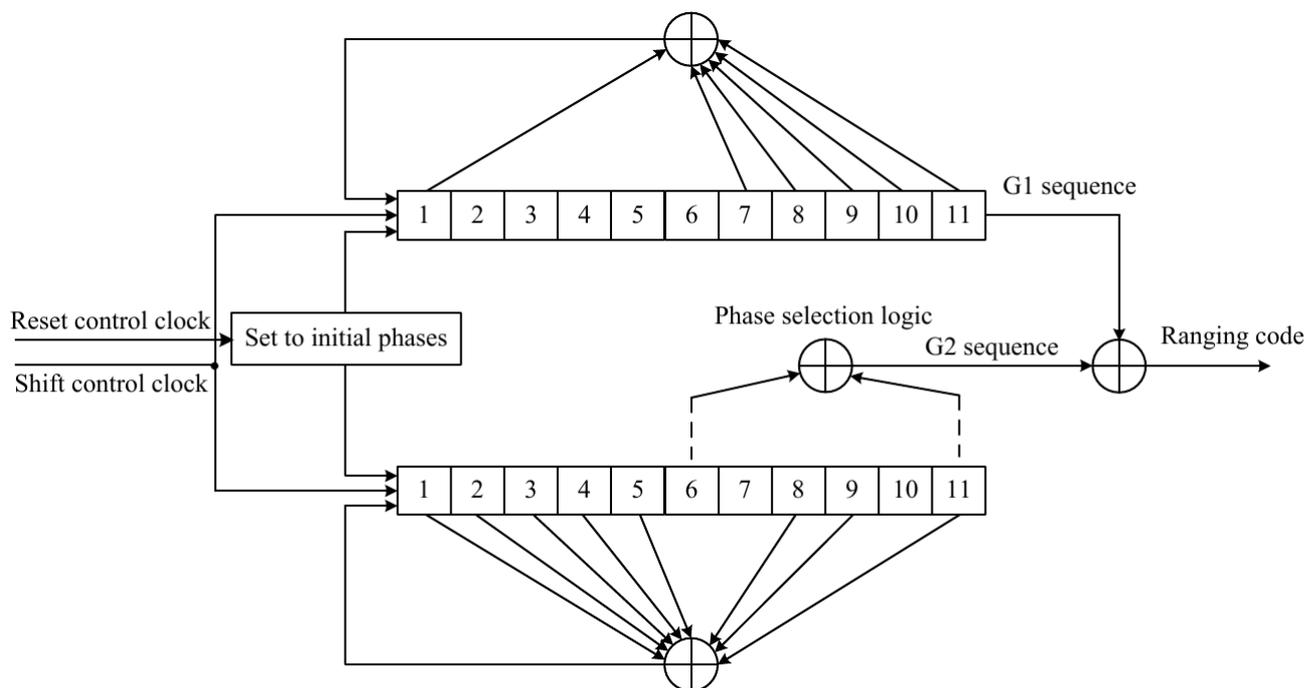


Рисунок 1.20 – Структурная схема формирования дальномерного кода В1I и В2I

Отразим данную структуру на имеющуюся структуру универсального генератора дальномерного кода.

Для реализации первого регистра G1 будем использовать регистр SR_1.

Обратная связь снимается с 1, 7, 8, 9, 10, 11 разрядов сдвигового регистра, следовательно для получения конфигурационного слова `code_bitmask1` необходимо в 1, 7, 8, 9, 10, 11 разряды записать "1" , а в остальные разряды записать "0" , таким образом `code_bitmask1 = 0x83E00000`.

Для реализации второго регистра G2 используем сдвиговый регистр `SR_2`. Обратная связь снимается с 1, 2, 3, 4, 5, 8, 9, 11 разрядов, для получения конфигурационного слова `code_bitmask2` записываем в соответствующие разряды "1" в остальные разряды записываются "0" , таким образом `code_bitmask2 = 0xB3E00000`.

Каждую 1 мс регистры сдвига `SR_1` и `SR_2` устанавливаются в начальные состояния, которые задаются конфигурационными словами `code_state1`, `code_state2` соответственно. Начальные состояние для регистров `SR_1` и `SR_2` одинаковые, `code_state1` и `code_state1` равны `0x55400000`.

Выходная последовательность первого регистра снимается с последнего разряда, соответственно для получения конфигурационного слова `code_out_bitmask1` необходимо в 11 разряд записать "1" , а в остальные разряды записать "0" , таким образом `code_out_bitmask1 = 0x200000`. Отводы на выход второго регистра зависят от номера ПСП и представлены в ИКД.

Для формирования дальномерного кода с помощью универсального генератора необходимо записать в его регистры следующие инициализирующие параметры:

- `code_state1` – `0x55400000`,
- `code_state2` – `0x55400000`,
- `code_bitmask1` – `0x83E00000`,
- `code_bitmask2` – `0xB3E00000`,
- `code_out_bitmask1` – `0x200000`,
- `code_out_bitmask2` – зависит от номера ПСП(спутника).

1.5.2 ВЗІ - сигнал с кодовым разделением

Структура сигнала

Сигнал излучается на несущей частоте $f = 1268,520$ МГц, имеет модуляцию ВРСК. Структура сигнала:

$$S_{B3I}(t) = A \cdot C_{B3I} \cdot D(t) \cdot \cos(2\pi ft + \varphi_{B3I}) \quad (1.28)$$

Формирование дальномерного кода

Структура формирования псевдослучайной последовательности одинакова для информационной и пилотной компонент.

Дальномерный код является усеченным кодом Голда, имеет длину $N=10230$, период $T = 1$ мс, символьная скорость $f = 10,23$ Мбит/с. Псевдослучайная последовательность формируется сложением по модулю 2 двух последовательностей, G1 и G2. Последовательности G1 и G2 получены из двух 13-ти разрядных регистров сдвига, их период 8191 бит. Отводы на обратную связь регистров сдвига задаются полиномами:

$$G1(X) = X^{13} + X^4 + X^3 + X + 1 \quad (1.29)$$

$$G2(X) = X^{13} + X^{12} + X^{10} + X^9 + X^7 + X^6 + X^5 + X + 1 \quad (1.30)$$

Схема генератора дальномерного кода, предложенная в ИКД [12] на рис. 1.21

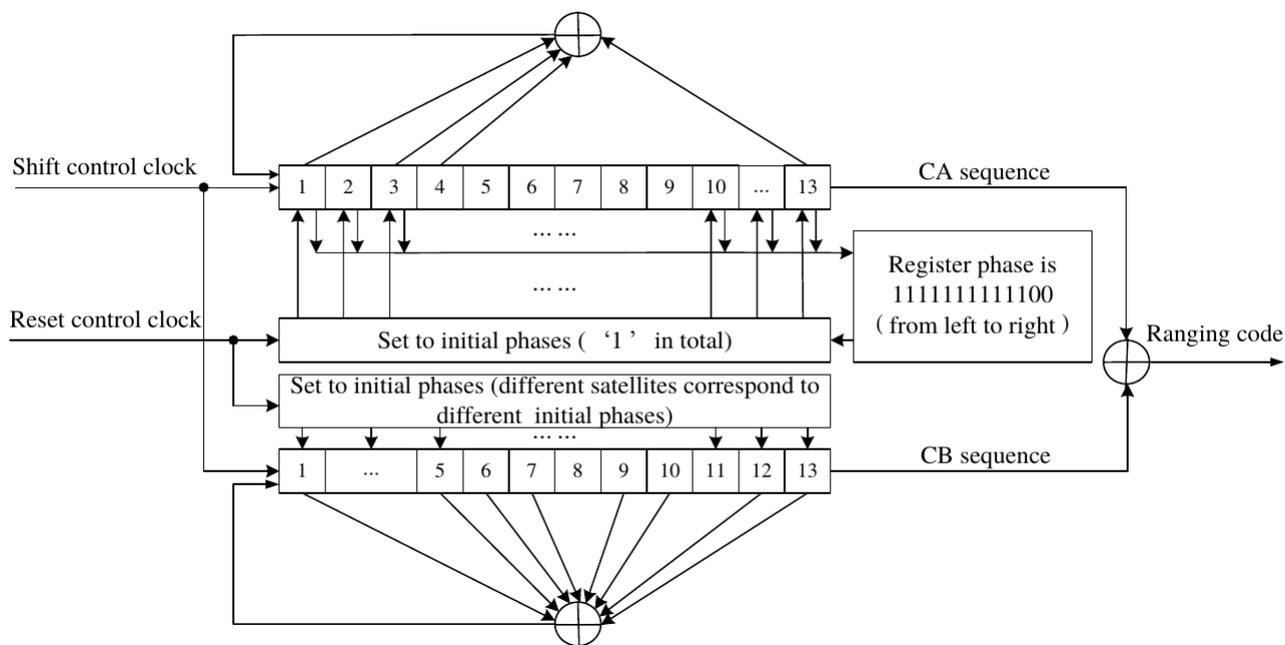


Рисунок 1.21 — Структурная схема формирования дальномерного кода V3I

Кодовая последовательность сгенерированная регистром G1 усекается и преобразуется в последовательность CA с периодом 8190 бит. Последовательность CB с периодом 8190 бит генерируется регистром G2. Дальномерный код получается путем сложения по модулю 2 (XOR) последовательностей CA и CB. Сброс регистра G1 происходит попеременно либо при достижении состояния – '1111111111100', либо при достижении счетчика символов значения 10230. Регистр G2 устанавливается в начальное состояние при достижении счетчика символов значения 10230. Начальное состояние для регистра G1 – '1111111111111' (во все разряды записывается 1), для регистра G2 начальное состояние зависит от номера ПСП.

Для формирования дальномерного кода с помощью универсального генератора необходимо записать в его регистры следующие инициализирующие параметры:

- code_state1 – 0xFFF80000,
- code_state2 – зависит от номера ПСП(спутника),
- code_bitmask1 – 0xB0080000,
- code_bitmask2 – 0x8ED80000,

- code_out_bitmask1 – 0x80000,
- code_out_bitmask2 – 0x80000.

В1С - открытый сигнал в диапазоне В1

Структура сигнала

Сигнал В1С излучается на несущей частоте $f = 1575,42$ МГц и содержит две компоненты: информационную В1Сdata и пилотную В1Сpilot. Уплотнение компонент – квадратурное [13].

$$S_{B1C}(t) = A \cdot C_{data} \cdot B(t) \cdot D(t) \cdot \sin(2\pi ft + \varphi) + A \cdot C_{pilot} \cdot O(t) \cdot B_{QМВОС}(t) \cdot \cos(2\pi ft + \varphi), \quad (1.31)$$

где $B_{QМВОС}$ – сложная модуляция QМВОС(ВОС(1,1) и ВОС(6,1))

Оверлейный код пилотной компоненты является кодом Вейла, имеет период $T = 18000$ мс.

Формирование дальномерного кода

Дальномерные коды сигнала Vds В1С являются кодами Вейла, имеют длину $N = 10230$, период $T = 10$ мс, символьную скорость $f = 1,023$ Мбит/с.

Формирование таких кодов с помощью генератора, рассматриваемого в работе, невозможно. Поэтому будем считать эти коды табличными.

1.5.3 В2а - сигнал с кодовым разделением

Структура сигналов

Сигнал В2а излучается на несущей частоте $f = 1176,45$ МГц и содержит информационную В2ад и пилотную В2ар компоненты, которые располагаются на двух квадратурах I и Q (фаза В2ар опережает фазу В2ад на 90°). Обе компоненты имеют модуляцию ВPSK(10) и одинаковую мощность [14].

$$S_{B2a}(t) = A \cdot C_P(t) \cdot O(t) \cdot D(t) \cdot \sin(2\pi ft + \varphi) + A \cdot C_D(t) \cdot O(t) \cdot \cos(2\pi ft + \varphi) \quad (1.32)$$

Оверлейный код информационной компоненты не зависит от номера ПСП, это фиксированная 5-ти битовая последовательность – 00010(в бинарном формате), старший бит передается первым.

Оверлейный код пилотной компоненты является кодом Вейла и зависит от номера ПСП(спутника). Схема формирования представлена в ИКД, для работы нашего генератора представим его в табличной форме.

Формирование дальномерного кода

Дальномерный код является кодом Голда имеет длину $N = 10230$, период $T = 1\text{мс}$, символьная скорость $f = 10,23\text{ Мбит/с}$. Псевдослучайная последовательность формируется сложением по модулю 2 двух последовательностей, полученных из двух 13-ти разрядных регистров сдвига. Отводы на обратную связь регистров сдвига задаются полиномами:

Для информационной компоненты В2ad:

$$g1(X) = X^{13} + X^{11} + X^5 + X + 1 \quad (1.33)$$

$$g2(X) = X^{13} + X^{12} + X^{11} + X^9 + X^5 + X^3 + 1 \quad (1.34)$$

Для пилотной компоненты В2ар:

$$g1(X) = X^{13} + X^7 + X^6 + X^3 + 1 \quad (1.35)$$

$$g2(X) = X^{13} + X^{12} + X^8 + X^7 + X^5 + X + 1 \quad (1.36)$$

Схемы генераторов дальномерных кодов информационной и пилотной компонент сигнала, предложенные в ИКД представлены на рис. 1.22 и 1.22.

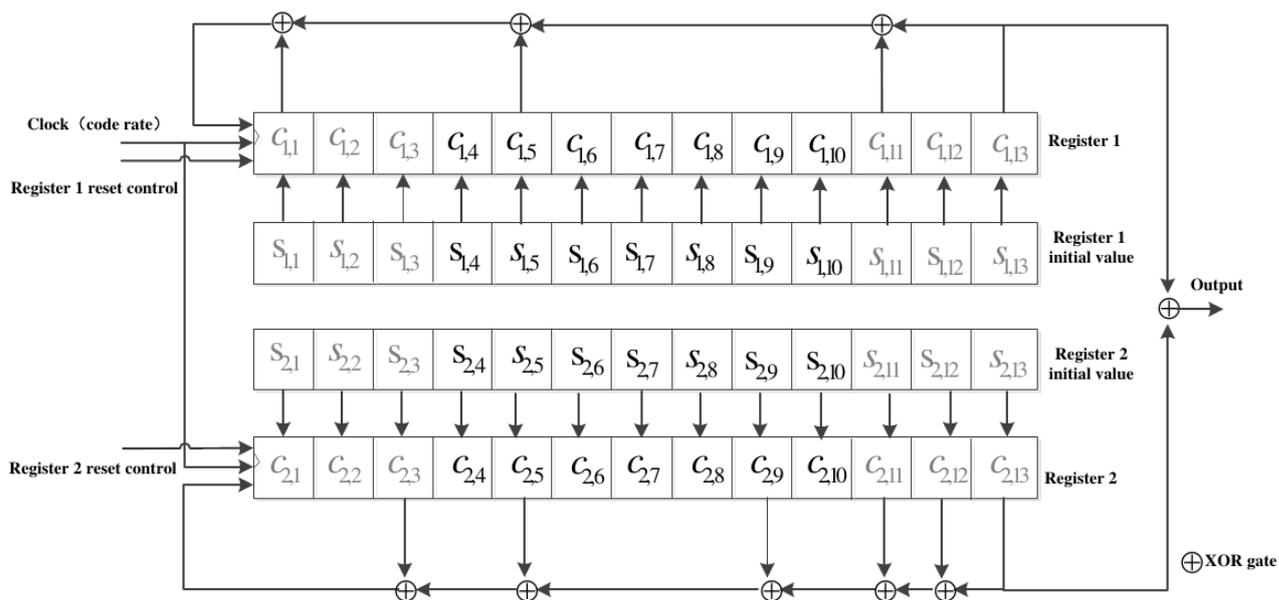


Рисунок 1.22 — Структурная схема формирования дальномерного кода сигнала B2ad

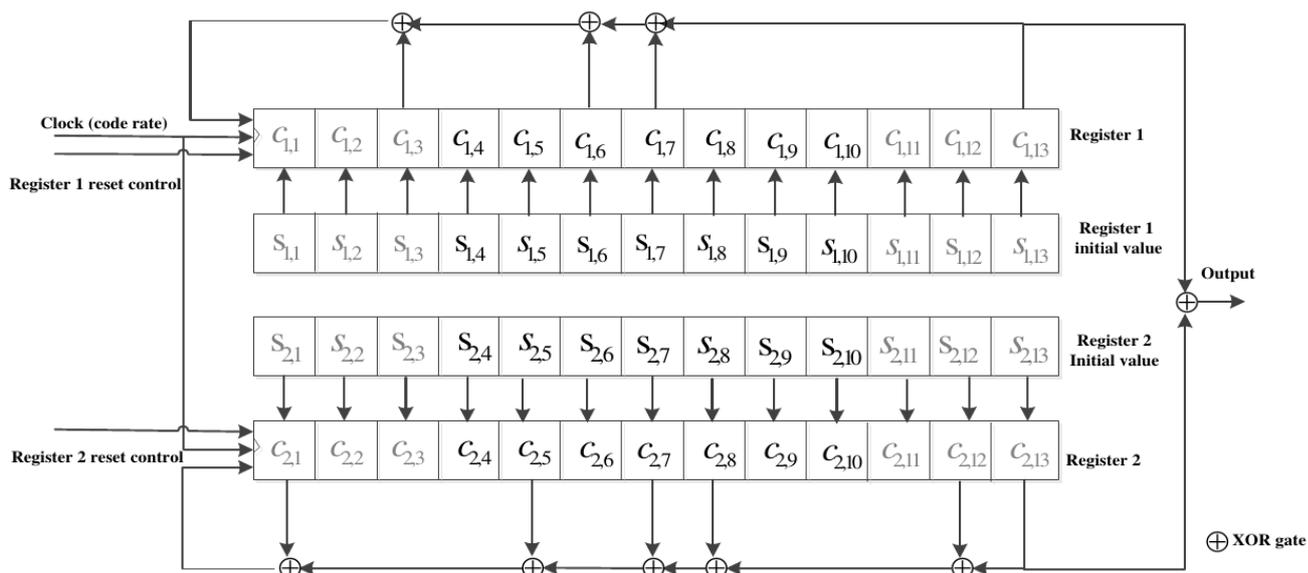


Рисунок 1.23 — Структурная схема формирования дальномерного кода сигнала B2ap

Первый регистр устанавливается в начальное состояние каждую 1 мс или после 8190 бита дальномерного кода. Второй регистр устанавливается в начальное состояние каждую 1 мс. Начальное состояние для первого регистра – '111111111111' (во все разряды записывается 1), для второго регистра начальное состояние зависит от номера ПСП.

Для формирования дальномерного кода информационной компоненты Bds B2ad с помощью универсального генератора необходимо записать в его реги-

стры следующие инициализирующие параметры:

- `code_state1` – 0xFFF80000,
- `code_state2` – зависит от номера ПСП(спутника),
- `code_bitmask1` – 0x88280000,
- `code_bitmask2` – 0x28B80000,
- `code_out_bitmask1` – 0x80000,
- `code_out_bitmask2` – 0x80000.

Для формирования дальномерного кода пилотной компоненты Vds B2ap с помощью универсального генератора необходимо записать в его регистры следующие инициализирующие параметры:

- `code_state1` – 0xFFF80000,
- `code_state2` – зависит от номера ПСП(спутника),
- `code_bitmask1` – 0x26080000,
- `code_bitmask2` – 0x8B180000,
- `code_out_bitmask1` – 0x80000,
- `code_out_bitmask2` – 0x80000.

1.6 ВЫВОДЫ

Проанализированы способы формирования дальномерных кодов открытых сигналов систем ГЛОНАСС, GPS, Galileo и BeiDou.

Найдены инициализационные параметры для сигналов: Gln L1OF, Gln L2OF, Gln L1OCd, Gln L1OCp, Gln L2OCp, Gln L3OCd, Gln L3OCp, GPS C/A, GPS L2C CM, GPS L2C CL, GPS L5 I/Q, Gal E5aI, Gal E5aQ, Gal E5bI, Gal E5bQ, Bds B1I, Bds B2I, Bds B3I, Bds B2ad, Bds B2ap, позволяющие формировать их дальномерные коды на универсальном генераторе.

Сигналы Gps P code, Gps L1C, Gal E1, Bds B1C требуют изменения структуры генератора. Это показано в таб. 1.5.

Таблица 1.5 — Формирование сигналов ГНСС

СРНС	всего открытых сигналов	ВОЗМОЖНО сформировать	НЕВОЗМОЖНО сформировать
GLONASS	7	7	0
GPS	9	5	4
Galileo	8	4	4
BeiDou	7	5	2
Summary	31	21	10

Удалось расширить список принимаемых сигналов до 21, таким образом с помощью такой структуры генератора возможно формирование почти 70% всех открытых сигналов.

Инициализирующие параметры генератора, для формирования кодов представлены в hex' формате и сведены в таблицу 1.6.

В графе len reg указано количество используемых разрядов каждого сдвигового регистра. Если регистр не используется для настройки генератора, то количество используемых разрядов равно нулю, а во все разряды этого регистра записываются нули, а в таблицу записывается «-».

Например, для настройки генератора на сигнал Gps C/A используется только один регистр сдвига, в котором используется всего 10 разрядов из 32, при этом в столбце len reg записано – (10+0).

Для сигналов с кодовым разделением, как правило одно из управляющих слов зависит от номера ПСП(т.е. от номера спутника). В таблице в таком случае в фигурных скобках указывается совокупная маска. В ней отражаются используемые биты для формирования всех ПСП.

Обратим внимание, что далеко не все разряды регистров сдвига используются для настройки генератора, что дает возможность модернизировать структуру генератора дальномерных кодов, тем самым сократив разрядность сдвиговых регистров.

Таблица 1.6 — Инициализирующие параметры генератора дальномерных кодов

Сигнал	len reg	code state1	code state2	bitmask1	bitmask2	out bitmask1	out bitmask2
GPS C/A	10+0	0xFFC00000	0xFFC00000	0x20400000	0x65C00000	0x00400000	{0xFFC00000}
GPS L2C CM	27+0	{0xFFFFFFE0}	—	0x3CA92920	—	0x20	—
GPS L2C CL	27+0	{0xFFFFFFE0}	—	0x3CA92920	—	0x20	—
GPS L5 I/Q	13+13	0xFFF80000	{0xFFF80000}	0x00D80000	0xB7180000	0x80000	0x80000
Gln LxOF	9+0	0xFF800000	—	0x8800000	—	0x02000000	—
Gln L1OCd	10+10	0x32000000	{0x0FC00000}	0x2400000	0x22C00000	0x400000	0x400000
Gln L1OCp	12+6	0x0C500000	{0xFC000000}	0x5300000	0x84000000	0x100000	0x400000
Gln L2OCp	14+7	0x34E00000	{0xFE000000}	0x110C0000	0x6000000	0x40000	0x2000000
Gln L3OCd	14+7	0x34E00000	{0x7E000000}	0x110C0000	0x6000000	0x40000	0x2000000
Gln L3OCp	14+7	0x34E00000	{0xFFFC0000}	0x110C0000	0x6000000	0x40000	0x2000000
Gal E5aI	14+14	0xFFFC0000	{0xFFFC0000}	0x85040000	0x1B140000	0x40000	0x40000
Gal E5aQ	14+14	0xFFFC0000	{0xFFFC0000}	0x85040000	0x1B140000	0x40000	0x40000
Gal E5bI	14+14	0xFFFC0000	{0xFFFC0000}	0x102C0000	0x49940000	0x40000	0x40000
Gal E5bQ	14+14	0xFFFC0000	{0xFFFC0000}	0x102C0000	0x8CC40000	0x40000	0x40000
Bds B1I/B2I	11+11	0x55400000	0x55400000	0x83E00000	0xB3E00000	0x200000	{0xFFE00000}

продолжение на следующей странице...

Таблица 1.6 – продолжение таблицы

Сигнал	len reg	code state1	code state2	bitmask1	bitmask2	out bitmask1	out bitmask2
Bds B3I	13+13	0xFFFF80000	{0xFFFF80000}	0xB0080000	0x8ED80000	0x80000	0x80000
Bds B2ad	13+13	0xFFFF80000	{0xFFFF80000}	0x88280000	0x28B80000	0x80000	0x80000
Bds B2ap	13+13	0xFFFF80000	{0xFFFF80000}	0x26080000	0x8B180000	0x80000	0x80000

2 Модель универсального генератора дальномерного кода

2.1 Модернизация генератора дальномерного кода

В первой главе нам удалось найти инициализирующие параметры генератора для формирования дальномерных кодов различных сигналов. Теперь перейдем к решению второй задачи – сокращение аппаратных ресурсов.

Проанализируем таблицу 1.6. В ней обратим внимание на второй столбец `len reg`. Там записано количество используемых разрядов, которое необходимо для настройки генератора для конкретного сигнала.

Универсальный генератор дальномерного кода содержит два 32-х разрядных сдвиговых регистра, но это явно избыточно. Ни один сигнал не требует для настройки генератора использования всех 32 разрядов.

Как видно из таблицы 1.6 максимальная длина первого регистра – 27 разрядов для сигналов Gps L2CM и Gps L2CL. А максимальная длина второго регистра сдвига – 14 разрядов (например, для сигналов Galileo).

Учитывая это можем сократить разрядности сдвиговых регистров: первого регистра (SR_1) до 27 разрядов, регистра (SR_2) до 14 разрядов, см. рис. 2.1

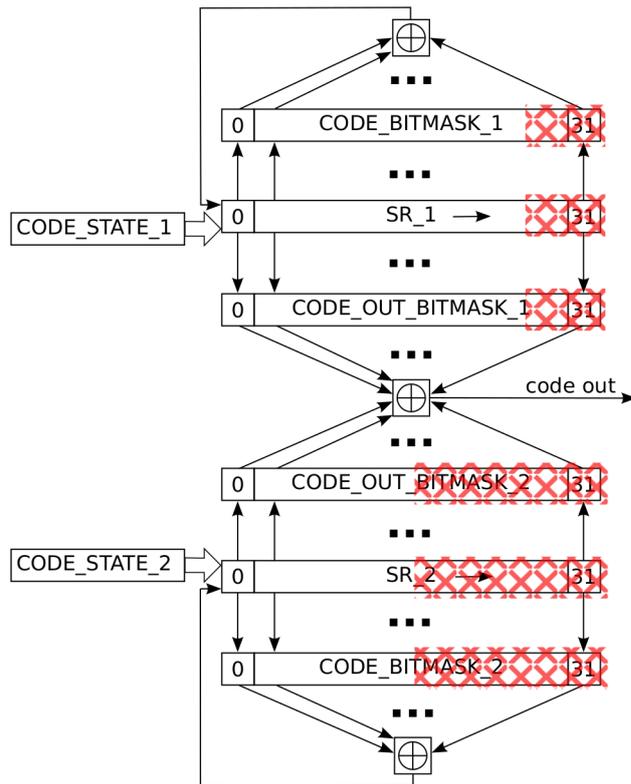


Рисунок 2.1 — Модернизация генератора

Такое сокращение разрядности дает нам выигрыш уже примерно в четверть.

Заметим, что для сигналов, использующих все 27 разрядов в первом сдвиговом регистре, второй регистр вообще не используется.

Для реализации таких сигналов реализуем систему, которая будет объединять два регистра сдвига в один. Таким образом получим один длинный, 28 разрядный регистр.

Учитывая такую систему, сократим разрядности регистров сдвига SR_1 и SR_2 до 14 разрядов, см. рис. 2.2.

Итак, нам удалось сократить разрядность сдвиговых регистров с 32 разрядов до 14, таким образом сократили структуру генератора более, чем в два раза.

2.2 Модель универсального генератора дальномерного кода

С учетом модернизаций, приведенных выше, получаем новую структуру универсального генератора дальномерных кодов.

Генератор дальномерных кодов состоит из двух 14 разрядных регистров сдвига SR_1 и SR_2 с обратными связями, начальное состояние которых записывается из соответствующих регистров CODE_STATE_1 и CODE_STATE_2 при наступлении новой эпохи.

Номера отводов сдвигового регистра, с которых снимается сигнал для обратной связи, задается битовой маской, которая находится в регистрах CODE_BITMASK_1 и CODE_BITMASK_2. С помощью регистров CODE_OUT_BITMASK_1 и CODE_OUT_BITMASK_2 задаются номера отводов для снятия выходного сигнала.

На каждом такте работы формируется один чип дальномерного кода. Управляя частотой поступления тактовых импульсов на генератор, можно управлять скоростью формирования символов.

В модели универсального генератора дальномерного кода, для экономии ресурсов, будет использоваться два режима работы:

1. два отдельных 14 разрядных регистра,
2. два регистра объединяются в один, образуя 28 разрядный регистр сдвига.

Объединение регистров сдвига SR_1 и SR_2 осуществляется по флагу cons.

В начале работы регистры сдвига SR_1 и SR_2 инициализируются начальными состояниями. Далее работа генератора зависит от выбранного режима.

При использовании первого варианта схема представляет собой два генератора M-последовательности с различными начальными состояниями и разными номерами отводов обратной связи. При этом дальномерный код формируется

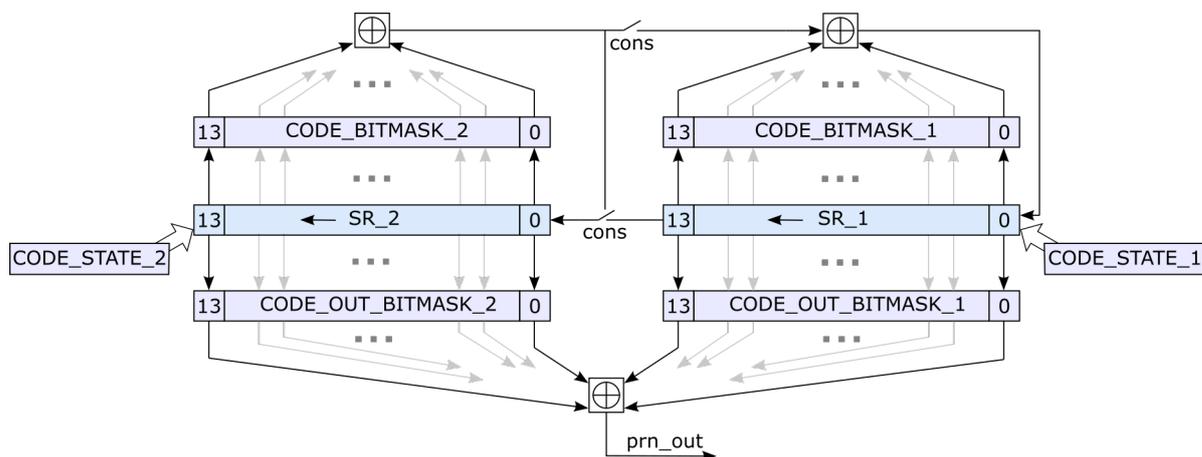


Рисунок 2.2 — Структурная схема универсального генератора дальномерного кода

путем суммирования (по модулю 2) полученных M -последовательностей.

На каждом шаге биты, определяемые регистрами `CODE_BITMASK_1` и `CODE_BITMASK_2`, суммируются по модулю 2. Далее регистры `SR_1` и `SR_2` сдвигаются влево, а рассчитанные биты записываются в младшие разряды сдвиговых регистров.

При использовании второго варианта, два регистра сдвига последовательно объединяются в один, образуя генератор M -последовательности большей длины, что позволяет генерировать дальномерные коды для сигналов, требующих 27 разрядный регистр (сигналы Gps L2 CM, Gps L2 CL).

На каждом шаге над битами, определяемыми регистрами `CODE_BITMASK_1` и `CODE_BITMASK_2`, осуществляется операция "исключающее ИЛИ". Регистр `SR_2` сдвигается влево, на место младшего бита записывается старший бит регистра `SR_1`, в то же время сдвигается влево регистр `SR_1`, и на место младшего бита регистра `SR_1` записывается рассчитанный бит.

Выходная последовательность формируется путем суммирования (по модулю два) битов, определенных регистрами `CODE_OUT_BITMASK_1` и `CODE_OUT_BITMASK_2`.

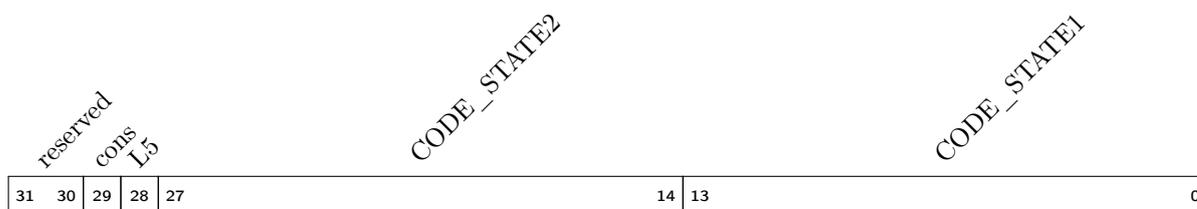
2.3 Интерфейс генератора дальномерного кода

Структура универсального генератора дальномерного кода отображается на карту памяти в виде регистров, представленных в таблице 2.1.

Таблица 2.1 — Карта памяти модуля Code Generator

Смещение	Название	Описание
0x00 (0x00)	CODE_STATES	начальное состояние регистров сдвига
0x04 (0x01)	CODE_BITMASKS	выбор отводов обратной связи
0x08 (0x02)	CODE_OUT_BITMASKS	выбор отводов на выход

Register 2.1 — CODE_STATES

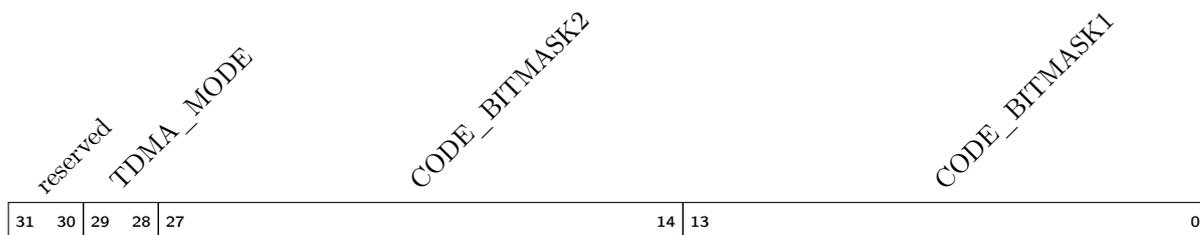


CODE_STATE1, CODE_STATE2 — определяют начальное состояние регистров сдвига.

L5 — формирование сигнала GPS L5.

cons — объединение регистров SR_1 и SR_2 в один.

Register 2.2 — CODE_BITMASKS

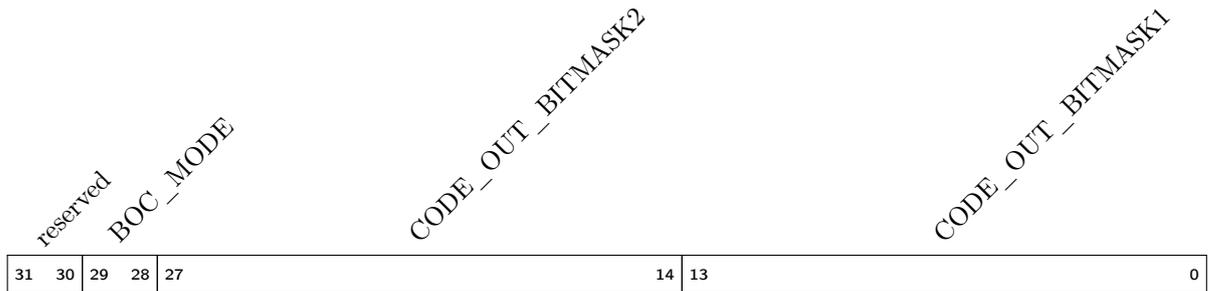


CODE_BITMASK1, CODE_BITMASK2 – выбор ветвей на обратную связь.

TDMA_MODE – временное уплотнение:

- 00 – без временного уплотнения,
- 01 – зануляется первая половина чипа,
- 10 – зануляется вторая половина чипа,
- 11 – без временного уплотнения (эквивалент 00).

Register 2.3 – CODE_OUT_BITMASKS



CODE_OUT_BITMASK1, CODE_OUT_BITMASK2 – выбор ветвей на выход.

BOC_MODE BOC_MODE:

00 – без поднесущей,

01 – частота поднесущей равна частоте смены символов ПСП, например, BOC(n,n) без TDMA,

10 – частота поднесущей равна удвоенной частоте смены символов ПСП, например, BOC(n,n) с TDMA или BOC(2n, n) без TDMA,

11 – частота поднесущей равна четырем частотам смены символов ПСП, например, BOC(2n,n) с TDMA или BOC(4n, n) без TDMA.

2.4 Функции инициализации

Регистры модуля доступны с процессора по шине адреса и данных. С помощью механизма mmap системы Linux эти регистры отображаются на виртуальную память программы приемника, где они интерпретируются с помощью структуры данных класса CodeGen.

Пример такой структуры:

```
class CodeGen {
public:
    uint32_t CODE_GEN_ID;
```

```

uint32_t CODE_STATES;
uint32_t CODE_BITMASKS;
uint32_t CODE_OUT_BITMASKS;
uint32_t CNTR_LENGTH;

CodeGen();
CodeGen *getCodeGen();

void set_code_states(uint32_t state1, uint32_t state2, bool 15,
    bool consolidation);
void set_code_bitmasks(uint32_t bitmask1, uint32_t bitmask2,
    uint8_t tdma_mode);
void set_code_out_bitmasks(uint32_t out_bitmask1, uint32_t
    out_bitmask2, uint8_t boc_mode);
void set_cntr_length (uint32_t x, uint8_t psp_mux);

uint32_t get_code_state1();
uint32_t get_code_state2();

uint32_t get_code_bitmask1();
uint32_t get_code_out_bitmask1();

uint32_t get_code_bitmask2();
uint32_t get_code_out_bitmask2();

uint32_t get_cntr_length    ();
uint32_t get_boc();

bool isConsolidated();
bool isL5();

void printfCodeGen();

};

```

CodeGen наследует классу Replica, в котором реализуются функции инициализации вида `initChannel`. Функции `initChannel` заполняют поля класса `CodeGen`.

Пример такой функции для сигнала `Gln L3OCd`:

```
int32_t Replica2Regs::initChannel_GLN_L3OCd(int8_t num, int32_t
    Fclk) {

    if ((num > 0) && (num <= 64)) {
        set_code_states      (0x072C, GLN2_rep2_CODESTATE2_d[num],
            false, false);
        set_code_bitmasks    (0x3088, 0x060, 0);
        set_code_out_bitmasks(0x2000, 0x040, 0);

    } else {
        return 0;
    }
}
```

Функции `set_code_states()`, `set_code_bitmasks()`, `set_code_out_bitmasks()` прописывают значения в соответствующие поля регистров.

В первой главе после анализа ИКД найдены инициализационные константы. На их базе составлены функции инициализации для сигналов: `Gln L1OF`, `Gln L2OF`, `Gln L1OCd`, `Gln L1OCp`, `Gln L2OCp`, `Gln L3OCd`, `Gln L3OCp`, `GPS C/A`, `GPS L2C CM`, `GPS L2C CL`, `GPS L5 I/Q`, `Gal E5aI`, `Gal E5aQ`, `Gal E5bI`, `Gal E5bQ`, `Bds B1I`, `Bds B2I`, `Bds B3I`, `Bds B2ad`, `Bds B2ap`.

Требуется проверить эти функции. Для решения этой задачи составлены юнит-тесты, они в свою очередь потребовали создания модели генератора на языке C++.

В каждом юнит-тесте функция инициализации настраивает модель генератора, проводится симуляция его работы. Выходные символы сравниваются с проверочными значениями, заданными в ИКД.

Юнит-тесты не только позволили исправить допущенные ошибки и опечатки в функциях инициализации, но и будут использоваться дальше при развитии

проекта - выявлять ошибки при рефакторинге кода и смене интерфейсов. Они вызываются каждый раз автоматически при правке разработчиками связанных участков кода.

Схема формирования дальномерного кода представлена на рисунке 2.3.

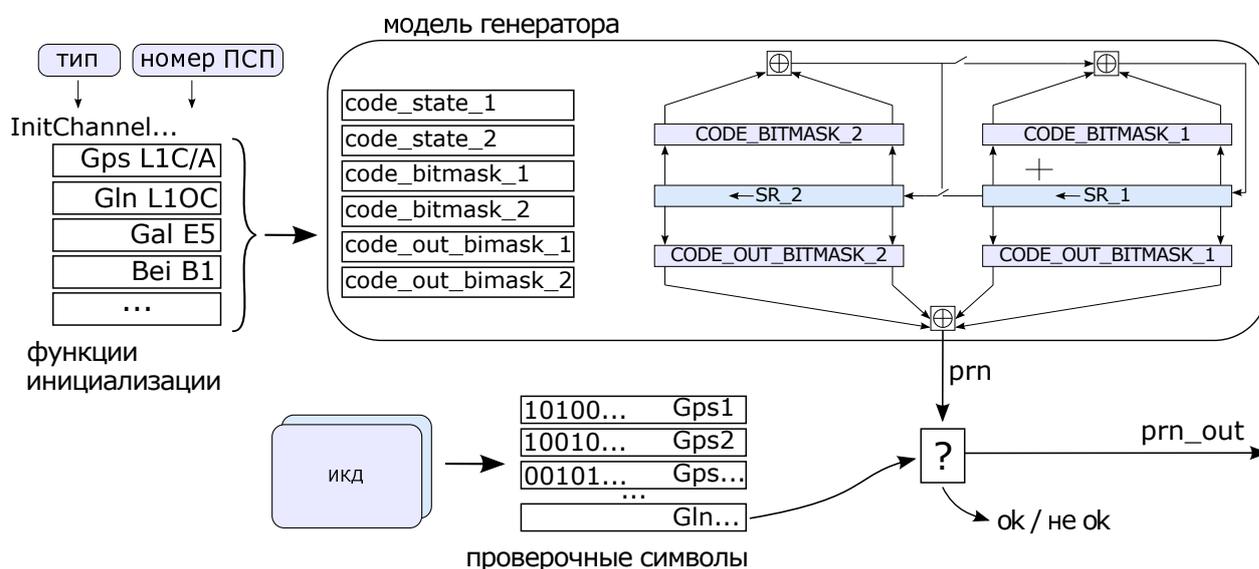


Рисунок 2.3 — Формирование дальномерного кода

Инициализационные параметры для настройки генератора сведены в таблицу 2.2. Каждое управляющее слов представлено в hex' формате, а так же для большей наглядности, в бинарном формате.

Для сигналов с кодовым разделением, как правило одно из управляющих слов зависит от номера ПСП(т.е. от номера спутника). В таблице в таком случае в фигурных скобках указывается совокупная маска, отражающая используемые биты для формирования всех ПСП.

Таблица 2.2 — Инициализирующие параметры генератора дальномерного кода

Сигнал	len reg	code state1	code state2	bitmask1	bitmask2	out bitmask1	out bitmask2
GPS C/A	10	0x3FF 00001111111111	0x3FF 00001111111111	0x204 00001000000100	0x3A6 00001110100110	0x200 00001000000000	{00001111111111}
GPS L2C CM	27 *	{01111111111111}	{11111111111111}	0x1252 1001001010010	0x153C 1010100111100	0x1000 01000000000000	0x0000 00000000000000
GPS L2C CL	27 *	{01111111111111}	{11111111111111}	0x1252 1001001010010	0x153C 1010100111100	0x1000 01000000000000	0x0000 00000000000000
GPS L5 I	13+13	0x1FFF 01111111111111	{01111111111111}	0x1B00 01101100000000	0x18ED 1100011101101	0x1000 01000000000000	0x1000 01000000000000
Gln LxOF	9	0x1FF 00000111111111	—	0x110 00000100010000	—	0x40 00000001000000	—
Gln L1OCd	10+10	0x4C 00000001001100	{00001111110000}	0x240 00001001000000	0x344 00001101000100	0x200 00001000000000	0x200 00001000000000
Gln L1OCp	12+6	0xA30 00101000110000	{00000000111111}	0xCA0 00110010100000	0x21 00000000100001	0x800 00100000000000	0x20 00000000100000
Gln L2OCp	14+7	0x72C 00011100101100	{00000000111111}	0x3088 11000010001000	0x60 00000000110000	0x2000 10000000000000	0x40 00000000100000
Gln L3OCd	14+7	0x72C 00011100101100	{00000000111110}	0x3088 11000010001000	0x60 00000000110000	0x2000 10000000000000	0x40 00000000100000
Gln L3OCp	14+7	0x72C 00011100101100	{00000000111111}	0x3088 11000010001000	0x60 00000000110000	0x2000 10000000000000	0x40 00000000100000
Gal E5aI	14+14	0x3FFF 11111111111111	{11111111111111}	0x20A1 10000010100001	0x28D8 10100011011000	0x2000 10000000000000	0x2000 10000000000000

продолжение на следующей странице...

Таблица 2.2 – продолжение таблицы

Сигнал	len reg	code state1	code state2	bitmask1	bitmask2	out bitmask1	out bitmask2
Gal E5aQ	14+14	0x3FFF 11111111111111	{11111111111111}	0x20A1 10000010100001	0x28D8 10100011011000	0x2000 10000000000000	0x2000 10000000000000
Gal E5bI	14+14	0x3FFF 11111111111111	{11111111111111}	0x3408 11010000001000	0x2992 10100110010010	0x2000 10000000000000	0x2000 10000000000000
Gal E5bQ	14+14	0x3FFF 11111111111111	{11111111111111}	0x3408 11010000001000	0x2331 10001100110001	0x2000 10000000000000	0x2000 10000000000000
Bds B1I/B2I	11+11	0x2AA 00001010101010	0x2AA 00001010101010	0x7C1 00011111000001	0x7CD 00011111001101	0x400 00010000000000	{00011111111111}
Bds B3I	13+13	0x1FFF 01111111111111	{01111111111111}	0x100D 01000000001101	0x1B71 01101101110001	0x1000 01000000000000	0x1000 01000000000000
Bds B2ad	13+13	0x1FFF 01111111111111	{01111111111111}	0x1411 01010000010001	0x1D14 01110100010100	0x1000 01000000000000	0x1000 01000000000000
Bds B2ap	13+13	0x1FFF 01111111111111	{01111111111111}	0x1064 01000001100100	0x18D1 01100011010001	0x1000 01000000000000	0x1000 01000000000000
summary	14+14	{11111111111111}	{11111111111111}	{11111111111111}	{11111111111111}	{11111001000000}	{11011111111111}

* – cons – используется объединение регистров SR_1 и SR_2 в один 28 разрядный регистр.

2.5 Анализ использования разрядов

Написание функций инициализации позволило проанализировать как используются те или иные разряды управляющих слов.

Для каждой системы управляющие слова разбиты на две группы. Первые три слова: `bitmask1`, `bitmask2`, `out bitmask1` – зависят только от номера спутника, последние три слова: `out bitmask2`, `state1` и `state2` – зависят как от типа сигнала, так и от номера спутника. Инициализирующие параметры для настройки генератора сведены в таблицы 2.3 – 2.10 для каждой из систем. Для большей наглядности все управляющие слова приведены в бинарном формате.

Если управляющее слово зависит от номера спутника, то для него записывается совокупная маска, которая показывает все используемые разряды для формирования сигналов для всех номеров спутников. В строке `summary` показаны все используемые разряды в управляющем слове для формирования всех типов сигналов системы и всех номеров спутников.

Таблица 2.3 – Инициализирующие параметры – ГЛОНАСС

Сигнал	<code>bitmask1</code>	<code>bitmask2</code>	<code>out bitmask1</code>
LxOF	00000100010000	-	00000001000000
L1OCd	00001001000000	00001101000100	00001000000000
L1OCp	00110010100000	00000000100001	00100000000000
L2OCp	11000010001000	00000001100000	10000000000000
L3OCx	11000010001000	00000001100000	10000000000000
Summary	11111111111000	00001101100101	10101001000000

Таблица 2.4 — Инициализирующие параметры – ГЛОНАСС

Сигнал	out bitmask2	state1	state2
LxOF	-	000001111111111	-
L1OCd	00001000000000	00000001001100	{00001111110000}
L1OCp	00000000100000	00101000110000	{00000000111111}
L2OCp	00000001000000	00011100101100	{00000001111111}
L3OCd	00000001000000	00011100101100	{00000001111110}
L3OCp	00000001000000	00011100101100	{00000001111111}
Summary	00001001100000	00111111111111	{00001111111111}

Проанализировав таблицы для системы ГЛОНАСС, заметим, что не все разряды управляющих слов используются, их можно занулить, тем самым сократить расход аппаратных ресурсов при реализации генератора. Например, в слове out bitmask2 используются лишь 3 разряда: 6, 7 и 10, в слове out bitmask1 только 4 разряда: 7, 10, 12 и 14. Остальные разряды не требуют конфигурации.

Но современный приемник должен принимать сигналы не только системы ГЛОНАСС, но и других систем. Нужна возможность формировать дальномерные коды систем GPS, Galileo, Beidou с помощью универсального генератора. Проанализируем используемые разряды в управляющих словах и для других систем.

Таблица 2.5 — Инициализирующие параметры – GPS

Сигнал	bitmask1	bitmask2	out bitmask1
C/A	00001000000100	00001110100110	00001000000000
L2C CM	01001001010010	01010100111100	01000000000000
L2C CL	01001001010010	01010100111100	01000000000000
L5 I	01101100000000	01100011101101	01000000000000
Summary	01101101010110	01111111111111	01001000000000

Таблица 2.6 — Инициализирующие параметры – GPS

Сигнал	out bitmask2	state1	state2
C/A	{000011111111111}	000011111111111	000011111111111
L2C CM	000000000000000	{011111111111111}	{111111111111111}
L2C CL	000000000000000	{011111111111111}	{111111111111111}
L5 I	010000000000000	011111111111111	{011111111111111}
Summary	{010011111111111}	{011111111111111}	{111111111111111}

В сигналах системы GPS в словах bitmask2, state1 и state2 неиспользуемым остается не более одного разряда. Оптимизировать таким образом целесообразно только конфигурационные слова out bitmask1 и bitmask1. Например, в слове out bitmask1 используются лишь два разряда: 10 и 13, в слове bitmask1 используются восемь разрядов. Остальные разряды не требуют конфигурации.

Таблица 2.7 — Инициализирующие параметры – Galileo

Сигнал	bitmask1	bitmask2	out bitmask1
E5aI	10000010100001	10100011011000	100000000000000
E5aQ	10000010100001	10100011011000	100000000000000
E5bI	11010000001000	10100110010010	100000000000000
E5bQ	11010000001000	10001100110001	100000000000000
Summary	11010010101001	10101111111011	100000000000000

Таблица 2.8 — Инициализирующие параметры – Galileo

Сигнал	out bitmask2	state1	state2
E5aI	10000000000000	11111111111111	{11111111111111}
E5aQ	10000000000000	11111111111111	{11111111111111}
E5bI	10000000000000	11111111111111	{11111111111111}
E5bQ	10000000000000	11111111111111	{11111111111111}
Summary	10000000000000	11111111111111	{11111111111111}

Для формирования дальномерных кодов сигналов системы Galileo в конфигурационных словах state1, state2 используются все разряды. Оптимизировать можно слова out bitmask1 и out bitmask2, так как для формирования всех сигналов системы в этих словах используется лишь один разряд.

В системе BeiDou в конфигурационных словах bitmask2, out bitmask2, state1, state2 используются практически все разряды (12-13 из 14 разрядов), соответственно путем зануления неиспользуемых разрядов значительно оптимизировать структуру генератора не получится.

Таблица 2.9 — Инициализирующие параметры – BeiDou

Сигнал	bitmask1	bitmask2	out bitmask1
B1I/B2I	00011111000001	00011111001101	00010000000000
B3I	01000000001101	01101101110001	01000000000000
B2ad	01010000010001	01110100010100	01000000000000
B2ap	01000001100100	01100011010001	01000000000000
Summary	01011111111101	01111111111101	01010000000000

Таблица 2.10 — Инициализирующие параметры – BeiDou

Сигнал	out bitmask2	state1	state2
B1I/B2I	{000111111111111}	00001010101010	00001010101010
B3I	01000000000000	01111111111111	{01111111111111}
B2ad	01000000000000	01111111111111	{01111111111111}
B2ap	01000000000000	01111111111111	{01111111111111}
Summary	{010111111111111}	01111111111111	{01111111111111}

Итак, мы перебрали все возможные значения для всех типов сигналов и всех номеров спутников. Оказалось, что если ограничиться приемом сигналов систем ГЛОНАСС и GPS, то можно сократить 15 бит из оставшихся 84. Но если рассматривать и прием сигналов остальных систем, то практически все разряды задействованы, это хорошо видно в таблице 2.2 в строке summary.

Также проанализировали, какое количество комбинаций используется для настройки генератора для приема всех сигналов рассматриваемых систем. Число комбинаций, которые дают 6 управляющих слов по 14 разрядов, равняется 2^{84} . Оказалось, что группа регистров bitmask1, bitmask2 и out bitmask1 принимает только 14 разных состояний. Таким образом, их инициализацию можно заменить 4-битным словом. Из-за технических особенностей это не имеет смысла в нашей текущей ПЛИС, но может дать выигрыш при реализации в интегральных схемах.

3 Формирование и прием навигационных сигналов

Главная задача этой главы – добавление поддержки новых сигналов в приемник.

Все сигналы можно разделить на три группы:

1. Эфирные сигналы, которые уже умеет принимать приёмник,
2. Эфирные сигналы, которые мы пока не умеем принимать,
3. Сигналы, которые пока не излучаются и недоступны на имитаторах, хотя и описаны в ИКД.

Для приема сигналов второй группы нам необходимо проинициализировать генератор дальномерного кода. Самое сложное – прием сигналов третьей группы. В этом случае необходимо не только научиться принимать сигналы, но и формировать их. Для формирования сигналов можно воспользоваться векторным генератором сигналов.

3.1 Векторные генераторы сигналов

Векторные генераторы сигналов позволяют формировать узкополосные сигналы сложной формы. Для этого сигнал представляется в виде своего низкочастотного эквивалента, который используется генератором для модуляции несущего колебания в соответствии со следующей математической моделью:

$$S(t) = A \cdot I(t)\cos(f_0t) - A \cdot Q(t)\sin(f_0t) \quad (3.1)$$

В случае сигнала BPSK и отсутствии ограничений на значение начальной фазы, низкочастотный эквивалент может быть представлен в виде:

$$\begin{cases} I(t) = \cos(\pi \cdot \theta(t)) \\ Q(t) = 0 \end{cases} \quad (3.2)$$

где $\theta(t)$ - модулирующая последовательность, принимающая значения $\{0, 1\}$.

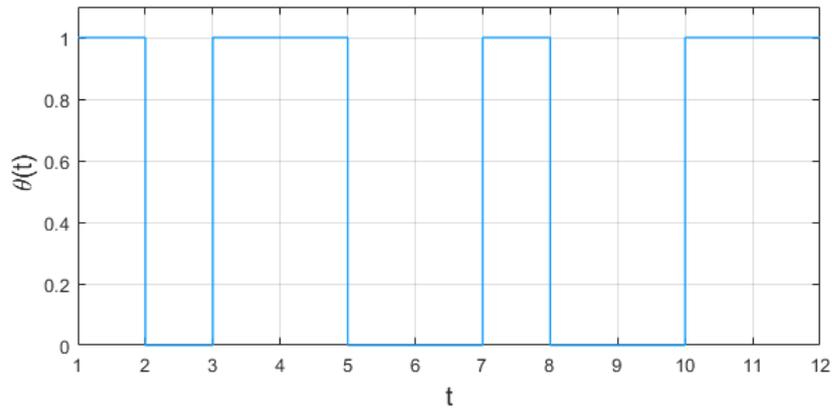


Рисунок 3.1 — Пример модулирующей функции

В большинстве радиотехнических приложений ось времени можно разбить на равные интервалы, в течение которых смена значений функции $\theta(t)$ не происходит - манипуляция сигнала производится с некоторым периодом τ_{chip} . Тогда функцию $\theta(t)$ можно описать последовательностью её значений на каждом интервале:

$$\theta_k = \theta(t_k). \quad (3.3)$$

В современных векторных генераторах присутствует возможность формирования соответствующих BPSK-сигналов по записанной в файл последовательности 3.3.

В свою очередь, манипуляция несущего колебания на π позволяет генерировать большинство существующих и перспективных радионавигационных сигналов.

Структура управляющего файла для векторного генератора SMBV

Для формирования низкочастотного эквивалента векторный генератор использует управляющие файлы - DataList-файлы. Файл имеет расширение `.dm_iqd` и состоит из двух основных разделов. Первый раздел, так называемый заголовок, содержит несколько обязательных и необязательных полей в формате ASCII, которые используются для предоставления информации о содержимом файла. Второй раздел содержит биты данных в двоичном формате.

Пример файла для сигнала GlnL1OF, без навигационного сообщения:

```
00000000 7b 54 59 50 45 3a 53 4d 55 2d 44 4c 7d 7b 43 4f |{TYPE:SMU-DL}{CO|
00000010 4d 4d 45 4e 54 3a 20 73 69 67 6e 61 6c 20 3d 20 |MMENT: signal = |
00000020 47 6c 6e 4c 31 4f 46 5f 30 30 30 2c 20 72 61 74 |GlnL1OF_000, rat|
00000030 65 20 35 31 31 30 30 30 20 73 79 6d 2f 73 7d 7b |e 511000 sym/s}{|
00000040 44 41 54 41 20 42 49 54 4c 45 4e 47 54 48 3a 35 |DATA BITLENGTH:5|
00000050 31 31 7d 7b 44 41 54 41 20 4c 49 53 54 2d 36 35 |11}{DATA LIST-65|
00000060 3a 23 fe 0f 7c 5c c8 25 3b 47 9f 36 2a 47 1b 57 |:#..|\.%;G.6*G.W|
00000070 13 11 00 84 61 39 56 1b d3 72 28 56 9f b2 4b 7e |...a9V..r(V..K~|
00000080 4d 4c c0 63 28 d2 fe 8b 1d 65 9e 3e e8 35 b7 60 |ML.c(...e.>.5. '|
00000090 b5 f5 50 29 5e 5d c0 e7 49 eb a8 90 ce 17 b6 68 |..P)^]..I.....h|
000000a0 77 86 7d                                     |w.}|
```

Значение полей:

- Тип прибора (обязательное поле) - это поле должно быть первым в файле.

{TYPE:SMU-DL}

- Комментарий (необязательное поле) - записывается тип сигнала и символьная скорость

{COMMENT: signal = GlnL1OF, rate = 511000 sym/s}

- Дата создания файла (необязательное поле)

{DATE:2019-04-17;18:30:10}

- Длина последовательности (обязательное поле)

{DATA BITLENGTH:511}

- Данные (обязательное поле) - длина последовательности в полных байтах, далее идет обязательный символ «:#», а после него записывается последовательность данных(по 8 бит, старший бит первым)

{DATA LIST-65:#bbbbbb...}

Формирование управляющего файла для векторного генератора

Для формирования управляющих файлов для векторного генератора SMBV написана программа на базе созданной ранее модели универсального ге-

нератора дальномерного кода. При этом воспроизводится навигационный сигнал во всей своей сложности:

1. модуляция дальномерным кодом,
2. модуляция оверлейным кодом,
3. модуляция цифровой информацией (навигационным сообщением),
4. модуляция цифровой поднесущей (ВОС),
5. временное уплотнение компонент.

Математическая модель такого сигнала:

$$S(t) = A \cdot C \cdot B \cdot O \cdot D \cdot M \cdot \cos(f_0 t + \varphi), \quad (3.4)$$

где A – амплитуда сигнала, C – модуляция дальномерным кодом, B – модуляция поднесущей (ВОС), O – модуляция оверлейным кодом, D – модуляция навигационным сообщением, M – функция стробирования при временном уплотнении, f_0 – частота несущей, φ – фаза несущей.

Как видим, сигнал можно описать как ФМ-2 сигнал. Поэтому мы сформируем файл, который содержит последовательность символов для двухуровневой фазовой манипуляции, включающий все пять особенностей нашей математической модели.

Схема формирования такого файла приведена на рис. 3.2

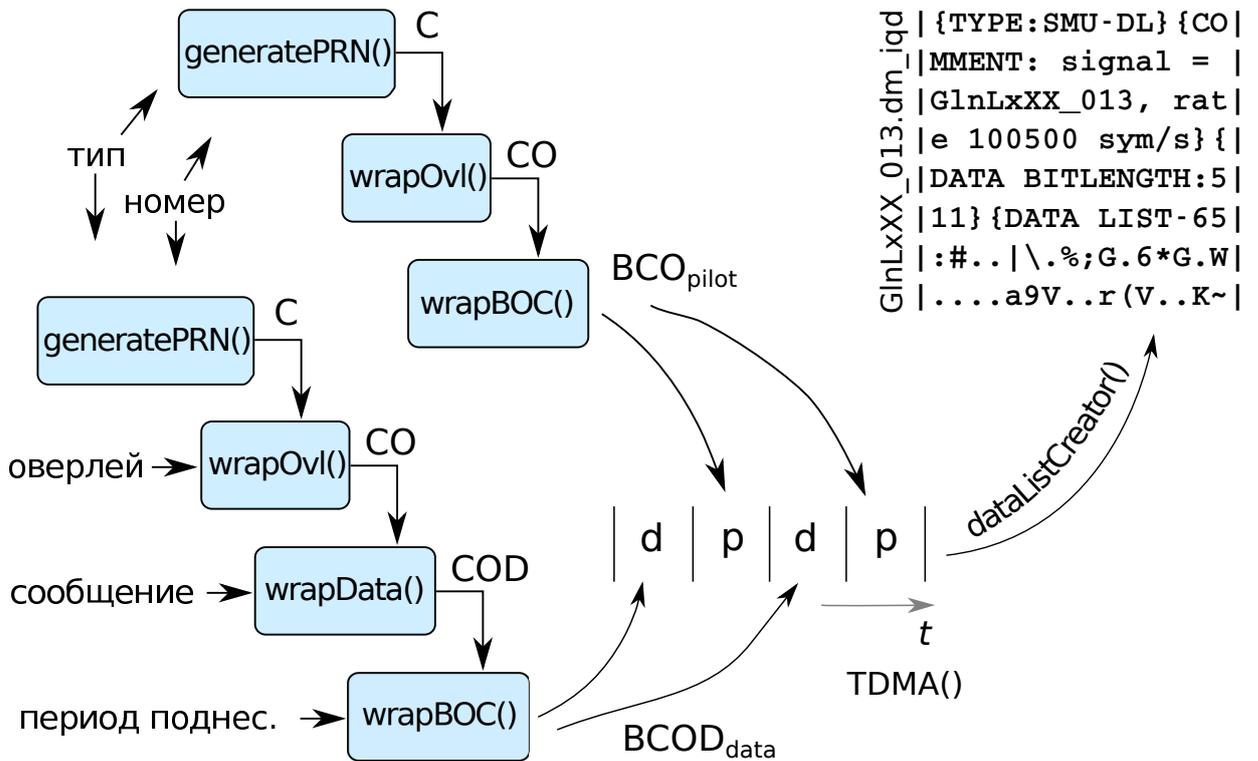


Рисунок 3.2 – Формирование DataList-файла

Итоговый файл имеет расширение `.dm_iqd` и содержит уложенные в байты символы манипуляции. Это позволяет сделать файл маленьким (порядка нескольких килобайт), и проигрывать очень длительные сэмплы на генераторе, в том числе воспроизводить кадры навигационного сообщения.

3.2 Формирование и прием сигнала с модуляцией дальномерным кодом и навигационным сообщением

Для того чтобы отладить формирователь сигналов мы начали с самого простого случая – сигнала ГЛОНАСС СТ(GlnL1OF), он имеет модуляцию дальномерным кодом и модуляцию данными. Математическая модель такого сигнала:

$$S(t) = A \cdot C \cdot D \cdot \cos(\omega t + \varphi), \quad (3.5)$$

Этот сигнал уже принимается приёмником. А значит, если приёмник начнет принимать сформированный сигнал, и выделять из него данные, то мы смогли

повторить реальный сигнал.

Формирование сигнала с использованием управляющего файла

Для использования сформированного DataList-файла необходимо предоставить его генератору с помощью программы RSCCommander, либо через флэш-накопитель.

- Для применения DataList-файла следует перейти в *config* блока *Baseband* рис.3.3, где выбрать подпункт *Custom Digital Modulation*.

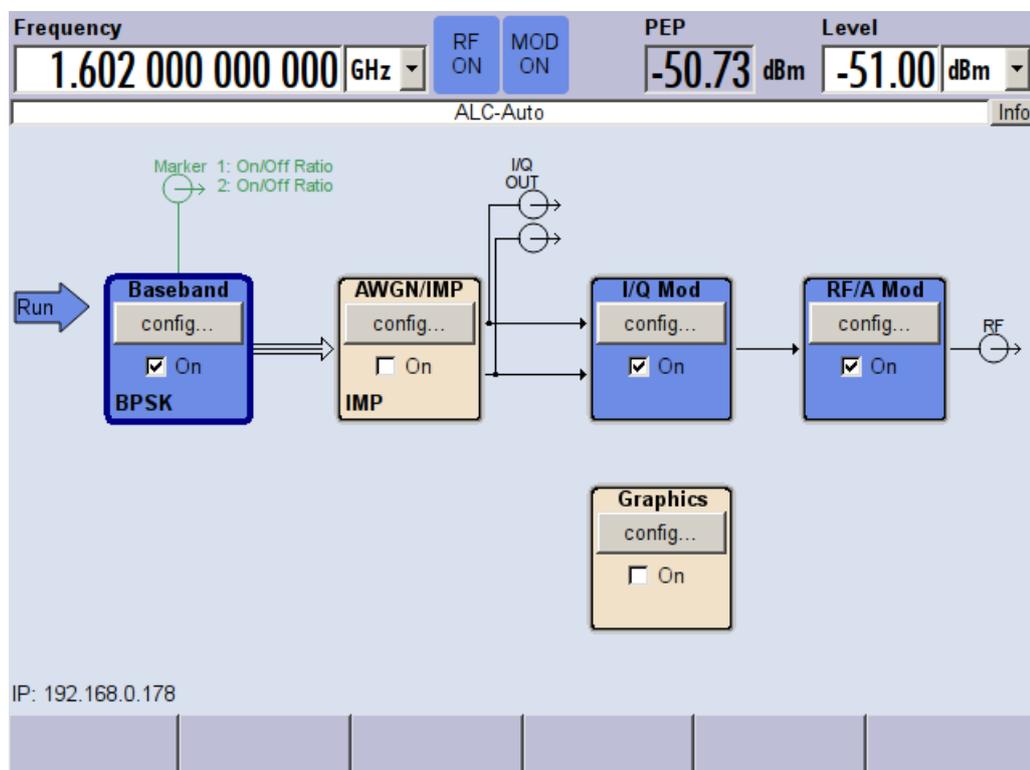


Рисунок 3.3

- В окне *Custom Digital Modulation* расположен блок настроек *Data Source* с помощью которого устанавливается источник данных для манипуляции сигнала. Следует выбрать *DataList* рис.3.4. При этом появится выбор файлов *Select Data List*. Там следует выбрать сформированный файл, в нашем случае *GlnL1OF_000*.

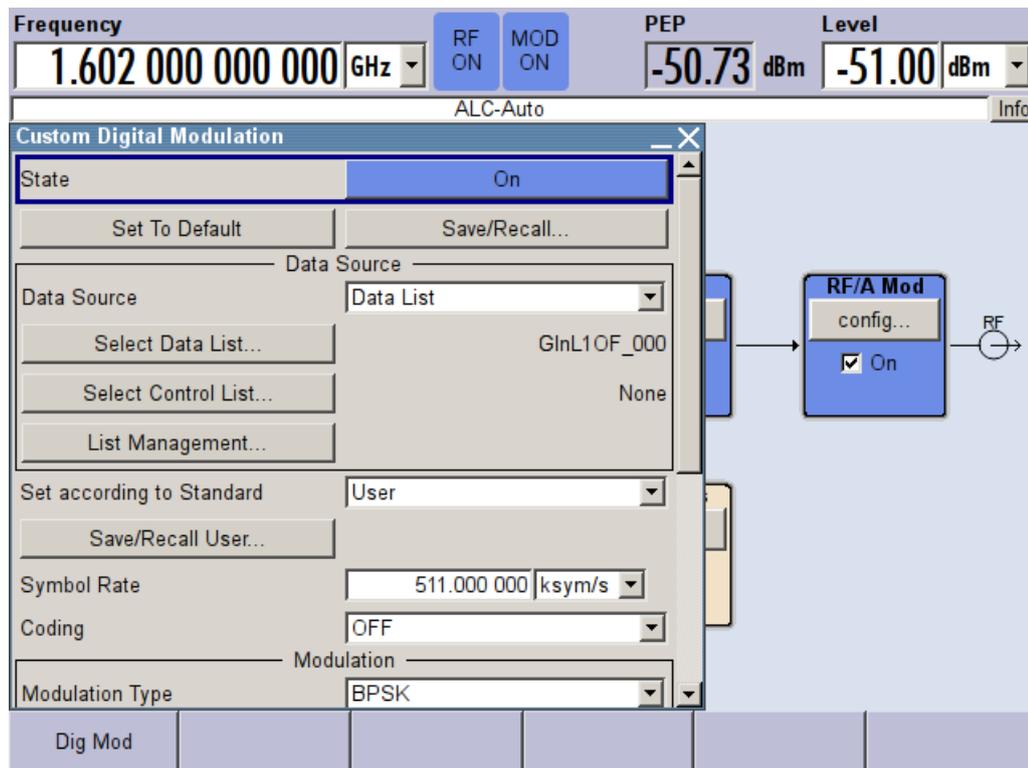


Рисунок 3.4

- В графе *Symbol Rate* устанавливаем скорость следования символов кода, в нашем случае $SymbolRate = 0,511$ Мбит/с, кодирование отключаем.
- Выбираем требуемый вид манипуляции, в нашем случае - BPSK.
- Выбираем вид фильтра для манипулирующих I,Q сигналов.
- Запускаем расчет квадратур для модулятора нажатием кнопки – *On*.
- Выбираем параметры несущей - частоту (*Frequency*) $F = 1602$ МГц и мощность (*Level*).
- Подаем её на модулятор нажатием кнопки – *RF on*.

Обработка сформированного сигнала навигационным приемником

Для проведения эксперимента использовался стенд (3.5) состоящий из векторного генератора сигналов Rohde&Schwarz SMBV100A, анализатора спектров и сигналов Rohde&Schwarz FSV, приемника и персонального компьютера.

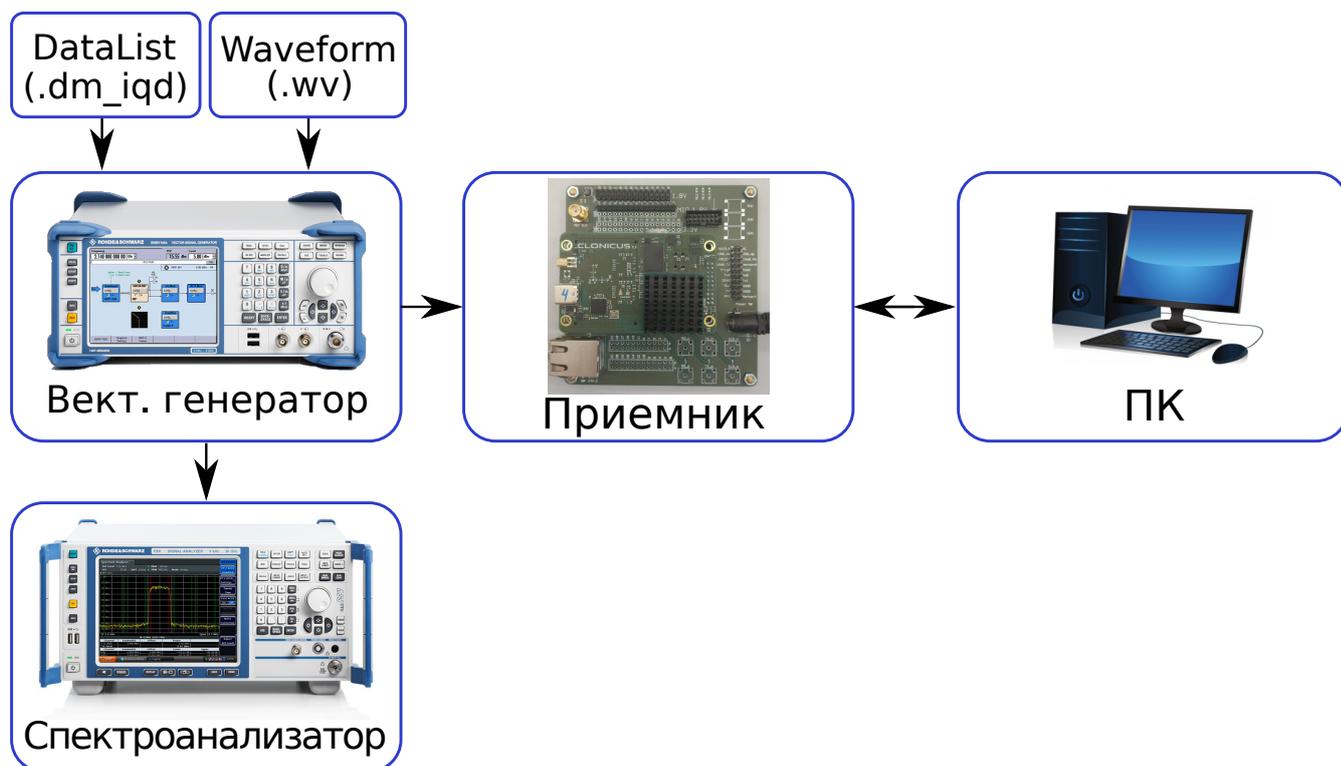


Рисунок 3.5 — Структурная схема стенда

Для обмена данными с приемником используется бинарный протокол обмена информацией. Он предполагает передачу данных в виде пакетов. Программа `srns2txt` для работы с протоколом обмена выделяет из бинарного потока данных соответствующий пакет и формирует текстовый файл, содержащий передаваемую в пакете информацию. Далее данный текстовый файл можно проанализировать в программном пакете MATLAB и представлять данные в графическом виде, либо просто в текстовом редакторе.

Структурная схема приемника приведена на рис. 3.6.

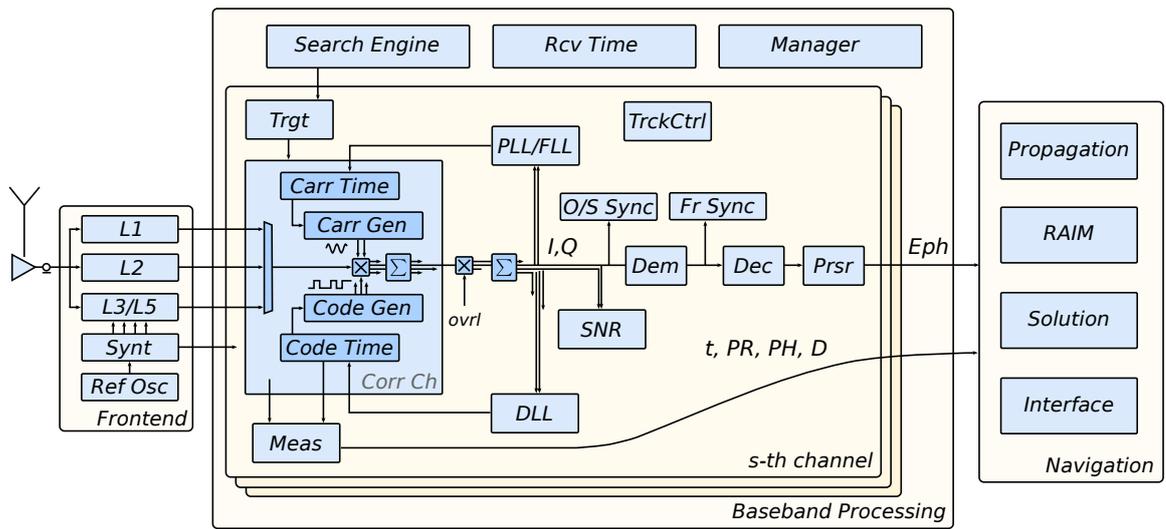
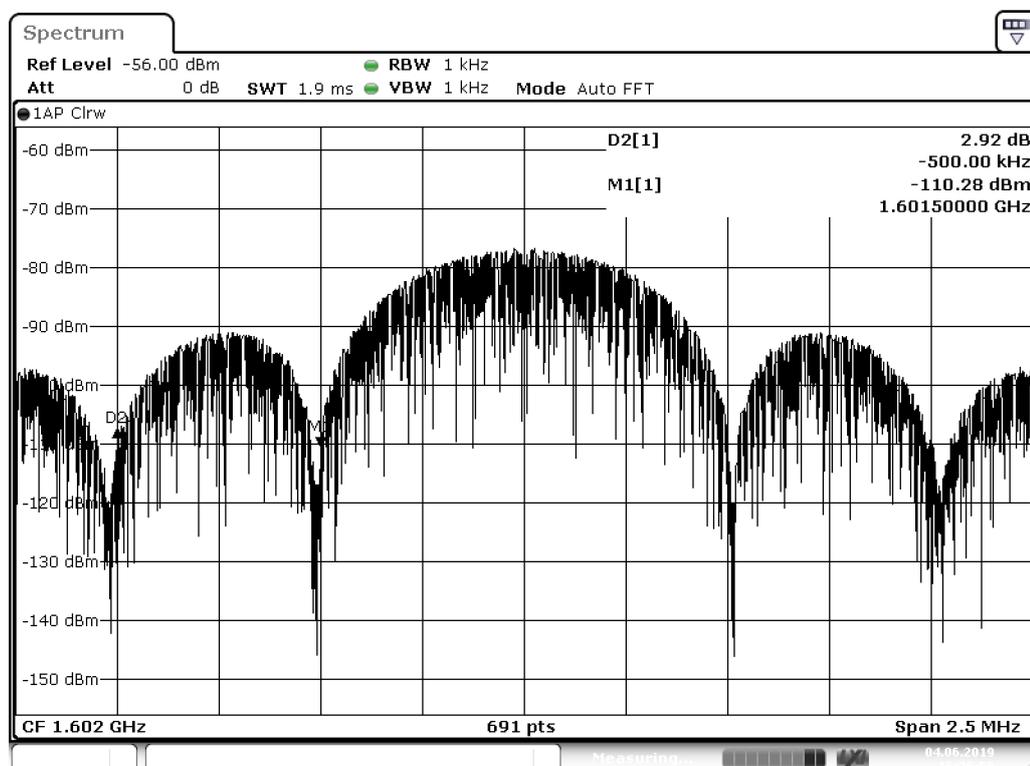


Рисунок 3.6 — Структурная схема приемника

В приемнике в качестве генератора ПСП используется наш универсальный генератор дальномерного кода.

С помощью анализатора спектра получили спектр сформированного сигнала, он представлен на рис.3.7. Ширина спектра сигнала по первым нулям равна 1.022 МГц, что соответствует действительности.



Date: 4.JUN.2019 16:26:59

Рисунок 3.7 — Спектр сигнала Gln L1OF

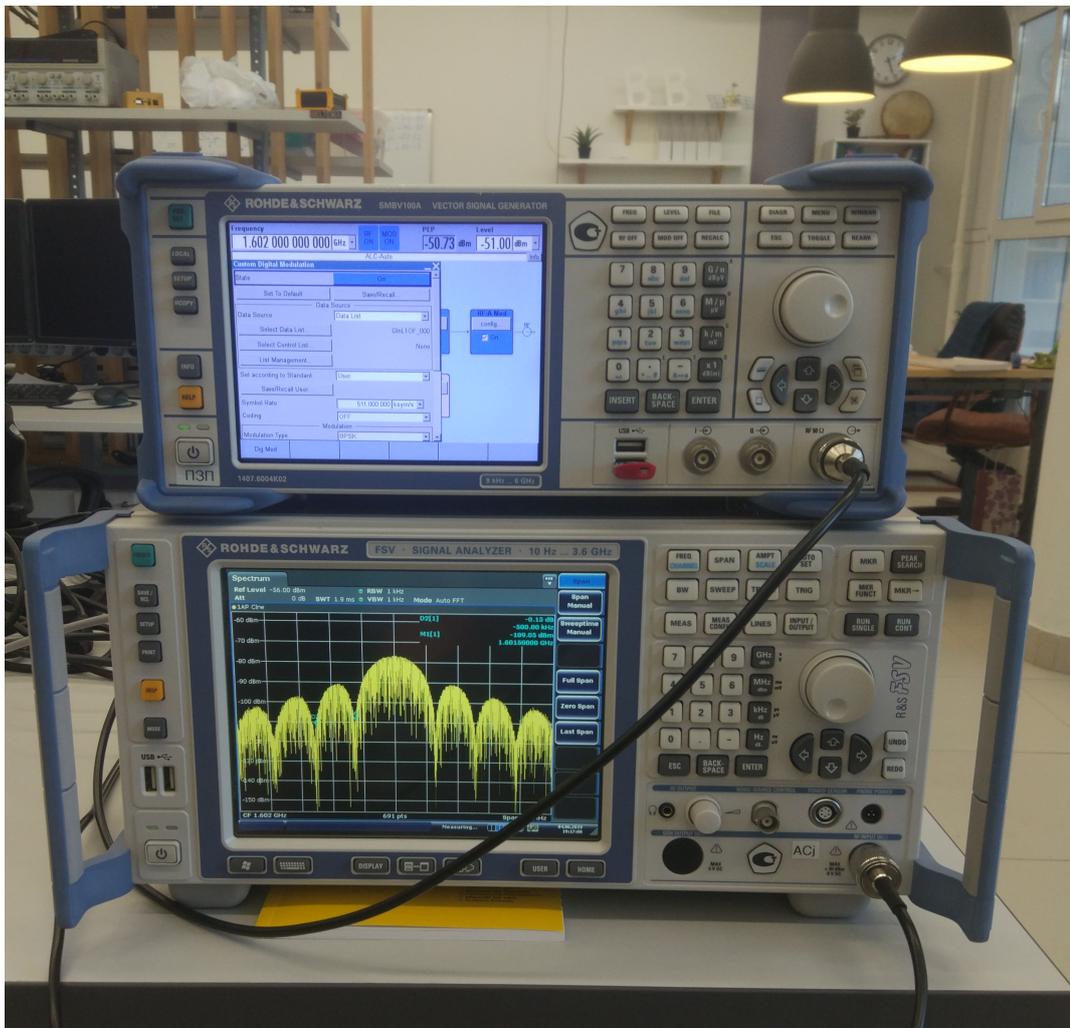


Рисунок 3.8 — Исследование выходного сигнала векторного генератора с помощью анализатора спектра

В протоколе обмена есть пакет 0x0022, который содержит информацию о канале слежения за одним радионавигационным сигналом, и пакет 0x0041, который содержит не декодированные символы цифровой информации. Проанализировав эти пакеты, проверили правильность принятого сигнала.

Мы сформировали сигнал Gln L1OF и приняли его нашим приемником, который в качестве генератора ПСП использует универсальный генератор дальномерного кода, рассматриваемый в работе. А значит нам удалось повторить реальный сигнал, который излучается спутниками.

Таким образом мы проверили работу формирователя сигналов, но это был простой сигнал, в нем нет оверлейного кода, модуляции поднесущей и временного уплотнения.

Для проверки всех функций формирователя реализуем более сложный сигнал, содержащий помимо модуляции ПСП и данными, также модуляцию оверлейным кодом, модуляцию поднесущей и временное уплотнение компонент. Примером такого сигнала является Gln L1OC. Сформируем его и примем нашим приемником.

3.3 Формирование и прием сигнала с модуляцией поднесущей и временным мультиплексированием

Сигнал L1OC излучается на несущей частоте $f = 1600,995$ МГц и содержит информационную L1OCd и пилотную L1OCp компоненты равной мощности, которые уплотняются путем чередования чипов дальномерных кодов этих компонент. Модель сигнала:

$$S_{L1OC}(t) = A \cdot C_d \cdot O \cdot D \cdot M_d \cdot \cos(2\pi ft) + A \cdot C_p \cdot B \cdot M_p \cdot \cos(2\pi ft) \quad (3.6)$$

Как видно, сигнал Gln L1OC содержит в себе пять особенностей. Дата-компонента содержит дальномерный код, модуляцию оверлейным кодом и навигационное сообщение. Пилот-компонента имеет модуляцию поднесущей. Также компоненты сигнала имеют временное уплотнение.

Проведем формирование сигнала с помощью векторного генератора. Для этого настраиваем его как описано выше: загружаем сформированный файл, устанавливаем скорость следования символов кода – $SymbolRate = 2,046000$ Мбит/с и частоту несущей $F = 1600,995$ МГц

С помощью анализатора спектра получили спектр сформированного сигнала, он представлен на рис. 3.9.

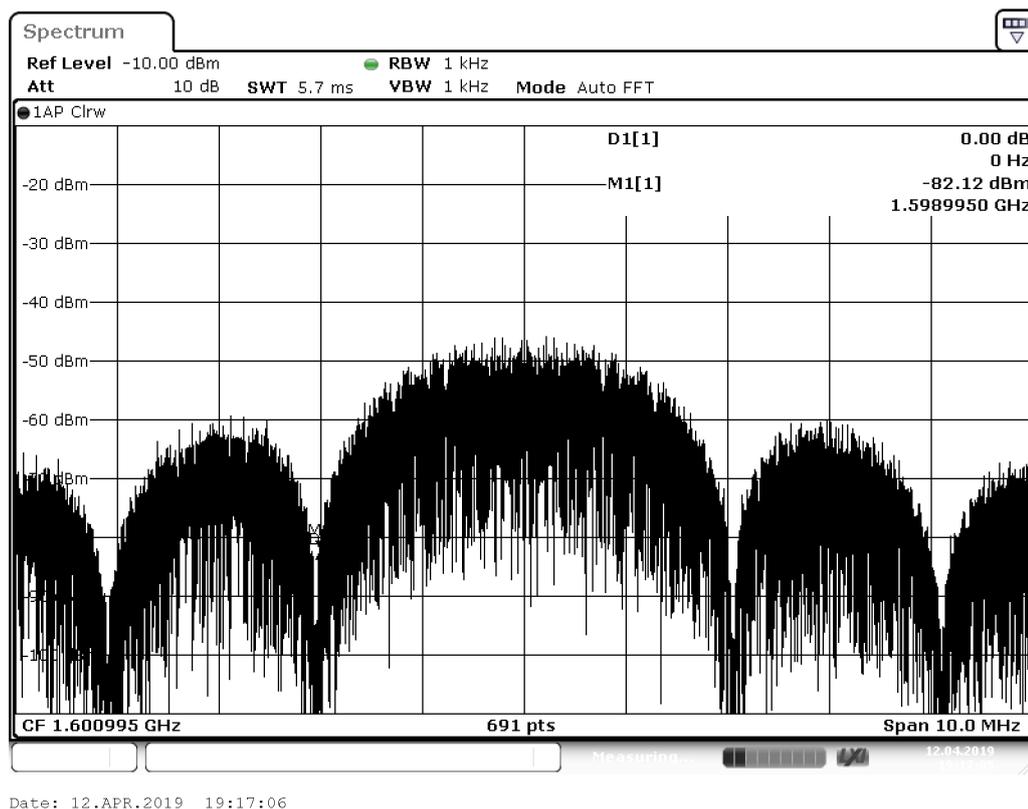


Рисунок 3.9 — Спектральная плотность мощности сформированного сигнала GlnL10C

Центральная частота $F = 1600,995$ МГц, ширина спектра по первым нулям равна 4,092 МГц.

Также отдельно приведем спектр пилотной компоненты сигнала, для этого установим $M_d(t) = 0$, чем исключим из сигнала информационную компоненту, см. рис. 3.10. Пилотная компонента имеет модуляцию поднесущей, что хорошо видно по форме спектра.

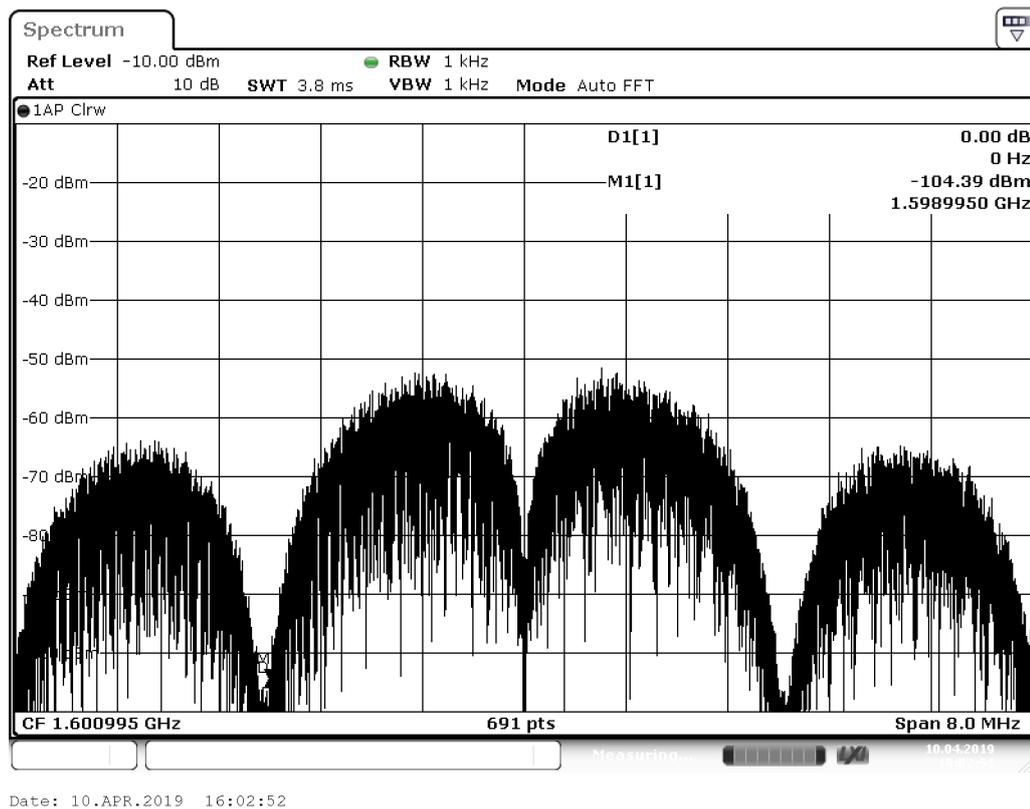


Рисунок 3.10 — Спектральная плотность мощности сформированного сигнала GlnL1OCp

Обработка пакет 0x0022 с помощью MATLAB и получим семейство корреляционных функций (рис. 3.11). Помимо корреляционной функции сигнала, показаны также корреляционные функции опережающего(early) и запаздывающего(late) сигналов.

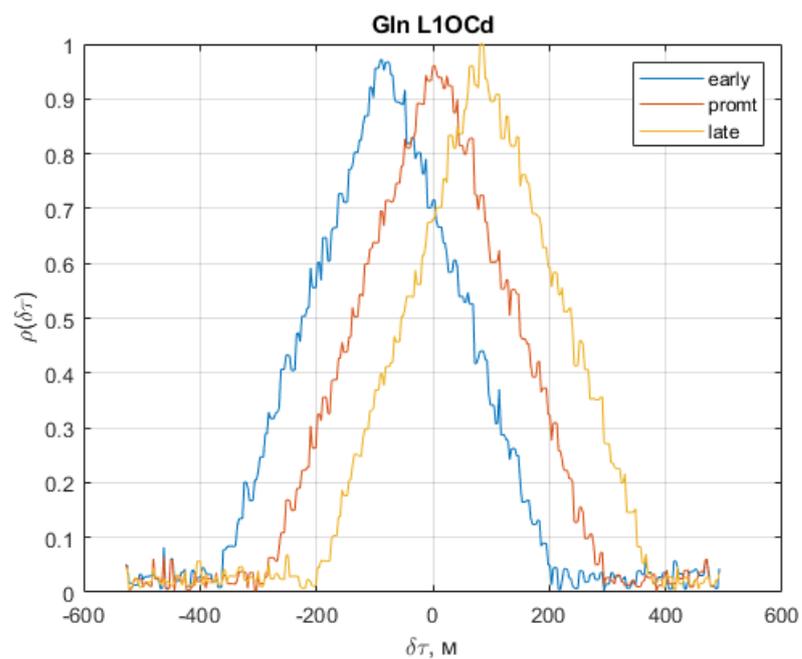


Рисунок 3.11 — Корреляционные функции для сигнала Gln L10Cd

На рисунке 3.11 показана корреляционная функция пилотной компоненты сигнала Gln L10C.

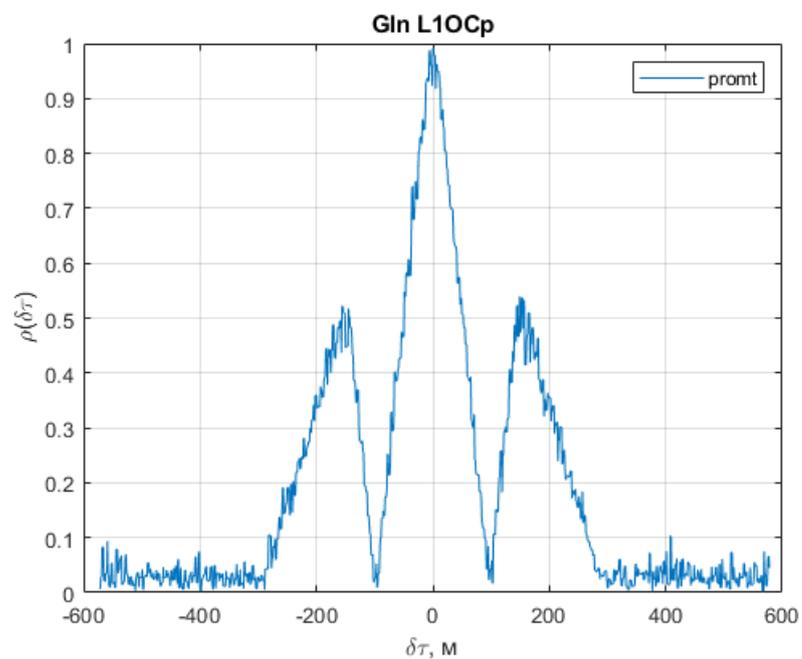


Рисунок 3.12 — Корреляционная функция для сигнала Gln L10Cp

Также из пакета 0x0022 с помощью MATLAB получим глаз-диаграмму.

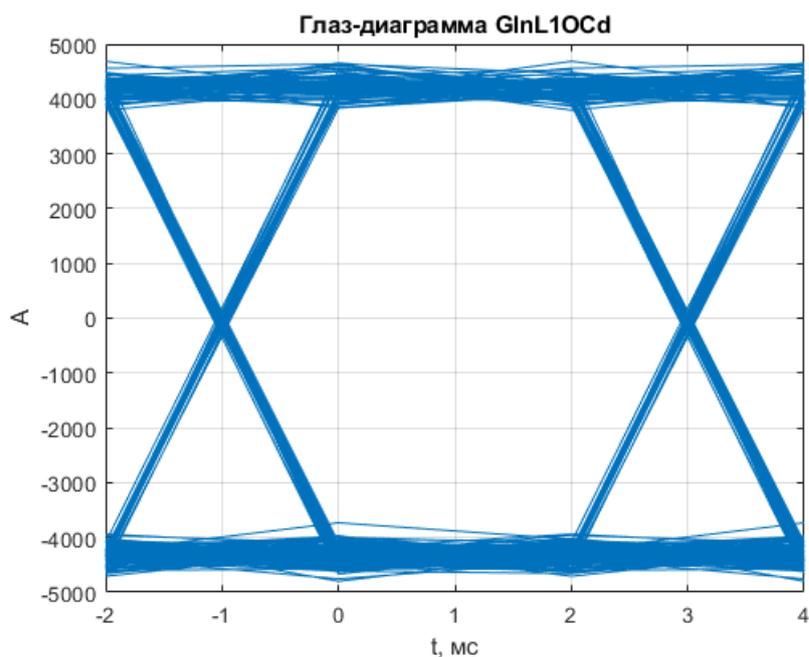


Рисунок 3.13 — Глаз-диаграмма

Пакет 0x0041 содержит не декодированные символы цифровой информации за 1 сек. Сигнал промодулирован цифровым сообщением "0001111000". С помощью программы srns2txt переведем этот пакет в текстовый формат. Видно, что записанное нами сообщение выделяется:

```
0 6 7 0 2 0 0 0 0 270017998 -1357 250 0x1E0781E0 0x0781E078 0x81E0781E 0xE0781E07 0x781E0781
0x1E0781E0 0x0781E078 0x01E0781E 000001111000000111100000011110000001111000000111100000011110000
001111000000111100000011110000001111000000111100000011110000001111000000111100000011110000001111
0000001111000000111100000011110000001111000000111100000011110000001111000000111100000011110
```

3.4 Прием референсной записи

В предыдущем разделе мы сформировали и приняли сигнал Gln L10C со сложной модуляцией, который содержит модуляцию дальномерным кодом, оверлейным кодом, модуляцию поднесущей, а так же временное уплотнение компонент. Но так как формирование низкочастотного эквивалента сигнала для векторного генератора и формирование опорного сигнала в приемнике происходит по одному и тому же алгоритму, возможно возникновение одинаковой ошибки как на приемной, так и передающей стороне.

Разработчиком бортовой аппаратуры предоставлен файл, в котором содержится выборка сигнала GlnL1OC без навигационного сообщения. С помощью этого файла мы можем проверить правильность работы нашего стенда.

Кодовая последовательность сформирована на нулевой частоте с частотой дискретизации $f_d = 122.76$ МГц. Последовательность сформирована на интервале времени 1 сек.

Предоставленный файл имеет расширение Waveform (.wf). Попробуем сформировать сигнал на векторном генераторе с помощью этого файла, а так же принять его. Таким образом проверим наш формирователь.

Структура Waveform-файла для векторного генератора SMBV

Радиосигналы можно рассматривать как произведения двух процессов: низкочастотного эквивалента (baseband signal) и несущей гармоника (radio frequency). На этом принципе основаны векторные генераторы сигналов. Они умножают квадратуры I и Q, задающие комплексный низкочастотный сигнал, и гармоническое колебание несущей частоты. В векторных генераторах Rohde&Schwarz для задания низкочастотного эквивалента сигнала также используются Waveform-файлы (расширение .wv).

Файл состоит из двух основных разделов. Первый раздел, так называемый заголовок содержит несколько обязательных и необязательных полей в формате ASCII, которые используются для предоставления информации о содержимом файла. Второй раздел содержит биты данных в двоичном формате.

Пример файла:

```
00000000 7b 54 59 50 45 3a 20 53 4d 55 2d 57 56 2c 20 30 |{TYPE: SMU-WV, 0|
00000010 7d 7b 4f 52 49 47 49 4e 20 49 4e 46 4f 3a 20 52 |}{ORIGIN INFO: R|
00000020 53 20 4d 61 74 6c 61 62 20 54 6f 6f 6c 6b 69 74 |S Matlab Toolkit|
00000030 7d 7b 4c 45 56 45 4c 20 4f 46 46 53 3a 20 32 2e |}{LEVEL OFFS: 2. |
00000040 36 32 32 38 2c 20 30 7d 7b 44 41 54 45 3a 20 32 |6228, 0}{DATE: 2|
00000050 30 31 39 2d 30 33 2d 32 31 3b 31 31 3a 30 30 3a |019-03-21;11:00:|
00000060 33 33 7d 7b 43 4c 4f 43 4b 3a 20 31 32 32 37 36 |33}{CLOCK: 12276|
00000070 30 30 30 30 7d 7b 53 41 4d 50 4c 45 53 3a 20 31 |0000}{SAMPLES: 1|
```

```

00000080 32 32 37 36 30 30 30 30 7d 7b 57 41 56 45 46 4f |22760000}{WAVEFO|
00000090 52 4d 2d 34 39 31 30 34 30 30 30 31 3a 20 23 82 |RM-491040001: #.|
000000a0 5a 82 |Z.Z.Z.Z.Z.Z.Z.Z.|

```

Значение полей:

- Тип прибора (обязательное поле) - это поле должно быть первым в файле.

{TYPE:SMU-WV}

- Смещения среднеквадратического и пикового значений (необязательное поле)

Формат: {LEVEL OFFS:RMSOffset_dB,PeakOffset_dB}

- Дата создания файла (необязательное поле)

Формат: {DATE: 2019-03-21;11:00:33}

- Частота дискретизации низкочастотного эквивалента сигнала, Гц (обязательное поле)

Формат: {CLOCK:122760000}

- Число отсчетов низкочастотного эквивалента сигнала (обязательное поле)

Формат: {SAMPLES:122760000}

- Последовательность квадратур (обязательное поле)

Формат: {WAVEFORM-491040001: # $I_0Q_0I_1Q_1\dots I_{N-1}Q_{N-1}$ }, где I_n и т.д. - отсчет квадратурной компоненты в формате little endian int16(порядок байтов от младшего к старшему), а 491040001 - это количество байт Length, вычисляемое по следующей формуле $Length = N * 4 + 1$, т.е. по 4 байта на каждый комплексный отсчет и 1 байт на символ #.

Формирование навигационных сигналов с помощью Waveform-файла

Формировать сигнал будем с помощью векторного генератора SMBV, для этого проведем следующие настройки:

- Для применения DataList-файла следует перейти в *config* блока *Baseband*3.14, где выбрать подпункт *ARB*.
- В окне *Arbitrary Wave Modulation* есть выбор файла с помощью которого устанавливается источник данных для манипуляции сигнала. Следует выбрать сформированный файл, в нашем случае *GlnL1OF_000.3.15*.

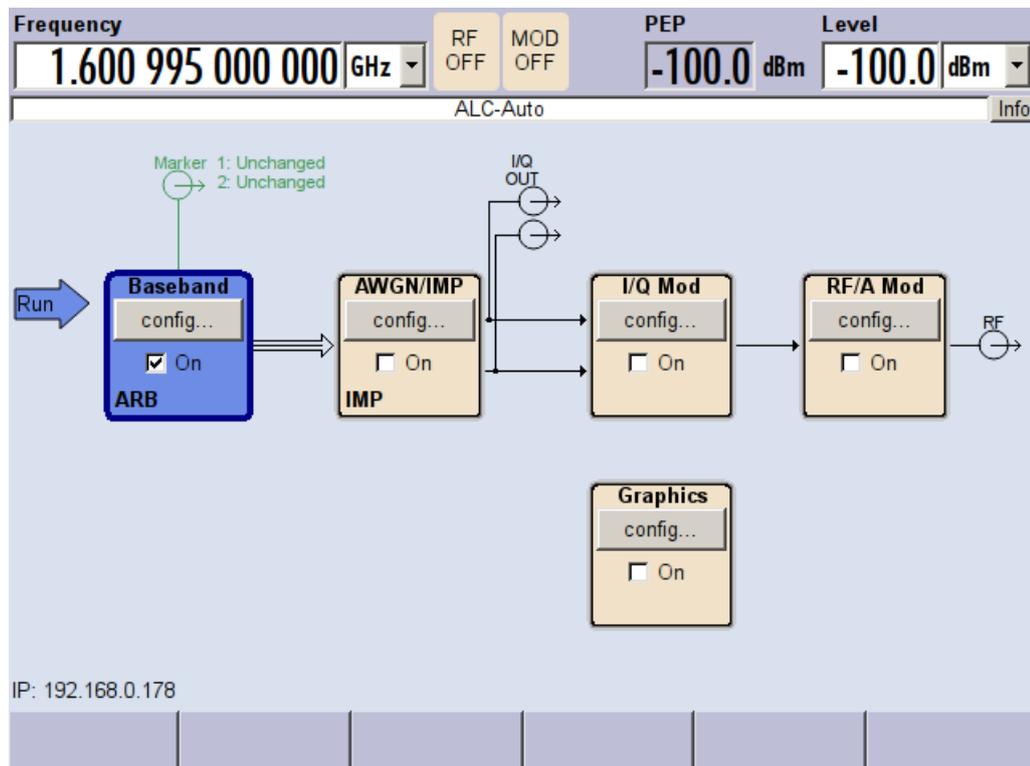


Рисунок 3.14

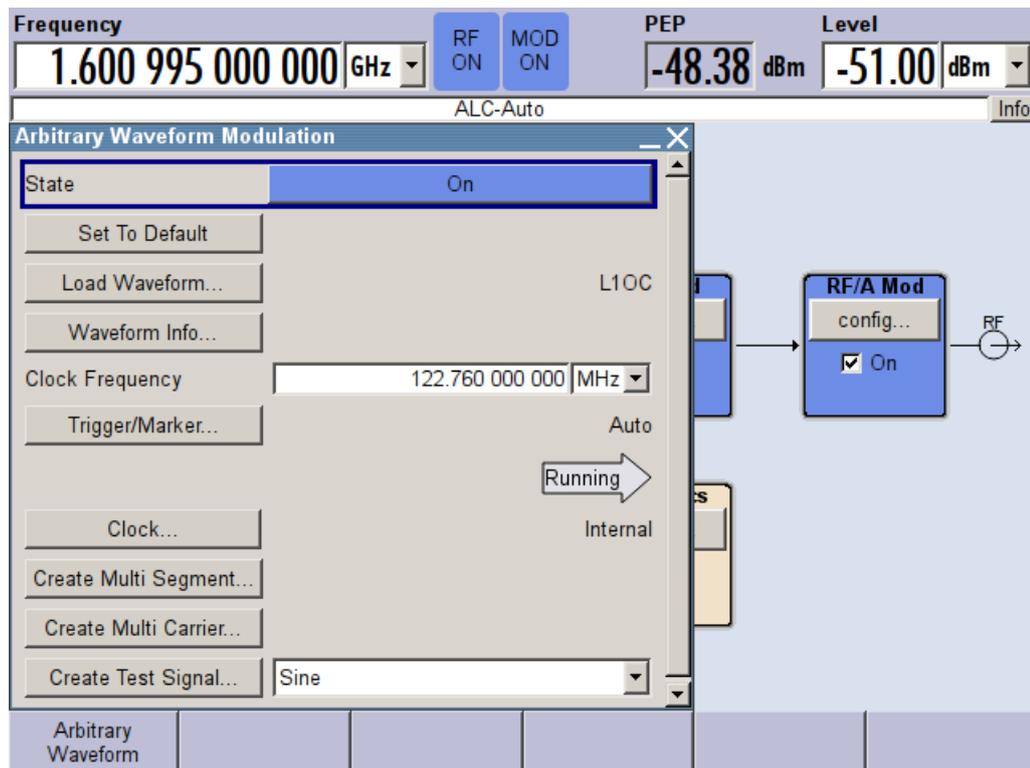
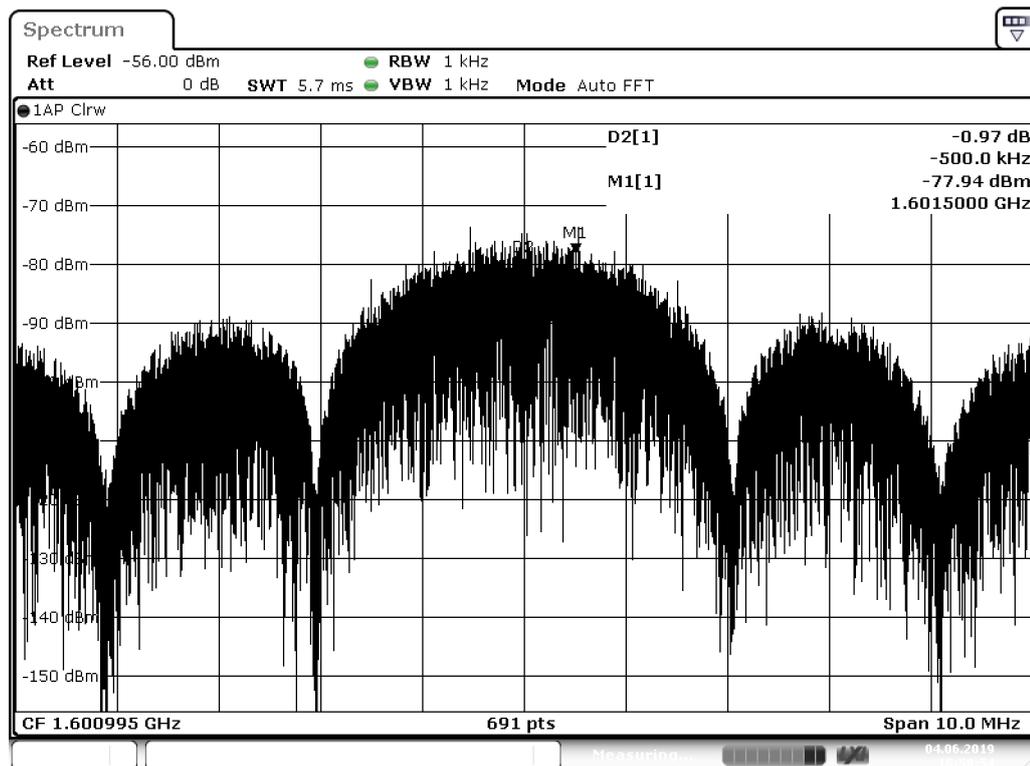


Рисунок 3.15

Обработка сформированного сигнала навигационным приемником

Спектр сформированного сигнала представлен на рис. 3.16. Он полностью совпадает со спектром сигнала Gln L1OC, сформированного в предыдущем разделе. Ширина спектра по первым нулям равна 4,092 МГц.



Date: 4.JUN.2019 16:58:54

Рисунок 3.16 — Спектр референсного сигнала L10С

Проанализировав пакеты 0x0022 и 0x0041, выяснили, что цифровых данных в этом сигнале нет, есть только модуляция оверлейным кодом.

Таким образом, мы отработали методику формирования сигналов с модуляцией дальномерным кодом, цифровой информацией, оверлейным кодом, цифровой поднесущей и временным мультиплексированием.

Заключение

В ходе работы был проведен анализ интерфейсных контрольных документов открытых сигналов систем ГЛОНАСС, GPS, Galileo и Beidou. Проанализированы способы формирования дальномерных кодов и найдены инициализирующие параметры для настройки универсального генератора этих кодов, используемого для приема навигационных сигналов различных систем.

Проведен анализ разрядности регистров этого генератора. По результатам анализа сокращена разрядность регистров с 32 до 14.

Переход к новой структуре генератора позволил сократить расход аппаратных ресурсов на его реализацию. В итоге число каналов коррелятора с использованием такой структуры поднялось с 135 до 200 в Zynq 7045. Также сформулированы рекомендации по дальнейшей оптимизации структуры универсального генератора при реализации в интегральных схемах.

Были найдены инициализирующие параметры для новой структуры генератора. На их базе составлены функции инициализации для сигналов ГЛОНАСС (L1/L2 OF, L1OCd, L1OCp, L2OCp, L3OCd, L3OCp), GPS (C/A, L2C CM/CL, L5 I/Q), Galileo (E5aI, E5aQ, E5bI, E5bQ), BeiDou (B1I, B2I, B3I, B2ad, B2ap), что позволило расширить список принимаемых сигналов до 21. Таким образом, с помощью универсального генератора дальномерных кодов можно формировать около 70% всех открытых навигационных сигналов систем ГЛОНАСС, GPS, Galileo и Beidou.

Для проверки правильности формирования дальномерных кодов используются юнит-тесты. В каждом тесте функция инициализации настраивает модель генератора, проводится симуляция его работы. Выходная последовательность сравнивается с проверочными символами, заданными в интерфейсных контрольных документах. Юнит-тесты не только позволили исправить допущенные ошибки и опечатки в функциях инициализации, но и будут использоваться дальше при развитии проекта приемника.

Разработана программа, позволяющая формировать файлы низкочастотных эквивалентов сигналов для генератора SMBV. Показана работа формирователя на примере простого BPSK сигнала ГЛОНАСС L1OF и сложного сигнала L1OC с BOC модуляцией, временным мультиплексированием и оверлейным кодом. По предложенной методике возможно формирование и других сигналов не только спутниковых, но и локальных навигационных систем.

Список литературы

1. Перов А. И., Харисов В. Н. ГЛОНАСС. Принципы построения и функционирования. — изд. 4-е, перераб. и доп. — М.: Радиотехника, 2010.
2. ГЛОНАСС. Интерфейсный контрольный документ. Навигационный радиосигнал в диапазонах L1, L2. — 2010. — (http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_rus_v5.1.pdf).
3. ГЛОНАСС. Интерфейсный контрольный документ. Навигационный радиосигнал открытого доступа с кодовым разделением в диапазоне L1. — 2016. — (<http://russianspacesystems.ru/wp-content/uploads/2016/08/IKD-L1-s-kod.-razd.-Red-1.0-2016.pdf>).
4. ГЛОНАСС. Интерфейсный контрольный документ. Навигационный радиосигнал открытого доступа с кодовым разделением в диапазоне L2. — 2016. — (<http://russianspacesystems.ru/wp-content/uploads/2016/08/IKD-L2-s-kod.-razd.-Red-1.0-2016.pdf>).
5. ГЛОНАСС. Интерфейсный контрольный документ. Навигационный радиосигнал открытого доступа с кодовым разделением в диапазоне L3. — 2016. — (<http://russianspacesystems.ru/wp-content/uploads/2016/08/IKD-L3-s-kod.-razd.-Red-1.0-2016.pdf>).
6. Navstar GPS Space Segment/User Segment Interfaces, IS-GPS-200J. — 2018. — (<https://www.gps.gov/technical/icwg/IS-GPS-200J.pdf>).
7. Navstar GPS Space Segment/User Segment L1C Interfaces, IS-GPS-800 Revision B. — 2018. — (<https://www.gps.gov/technical/icwg/IS-GPS-800E.pdf>).
8. Navstar GPS Space Segment/User Segment L5 Interfaces, IS-GPS-750. — 2018. — (<https://www.gps.gov/technical/icwg/IS-GPS-705E.pdf>).

9. Шатилов А. Ю. Характеристики радиосигналов глобальных спутниковых радионавигационных систем ГЛОНАСС, GPS, GALILEO, BEIDOU и функциональных дополнений SBAS. — Изд-во МЭИ, 2015.
10. European GNSS (Galileo) open service signal-in-space interface control document. — 2018. — (https://www.gsc-europa.eu/system/files/galileo_documents/Galileo-OS-SIS-ICD.pdf).
11. BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal (Version 2.1). — 2016. — (<http://en.beidou.gov.cn/SYSTEMS/ICD/201806/P020180608523308843290.pdf>).
12. BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B3I (Version 1.0). — 2018. — (<http://en.beidou.gov.cn/SYSTEMS/ICD/201806/P020180608516798097666.pdf>).
13. BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B1C (Version 1.0). — 2018. — (<http://www.beidou.gov.cn/xt/gfxz/201712/P020171226741342013031.pdf>).
14. BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B2a (Version 1.0). — 2017. — (<http://en.beidou.gov.cn/SYSTEMS/ICD/201806/P020180608518432765621.pdf>).

А Листинг программы, формирующей файлы НИЗКОЧАСТОТНЫХ ЭКВИВАЛЕНТОВ СИГНАЛОВ ДЛЯ генератора SMBV

```
struct GenSgnCompVars {
    SignalType        type;
    std::vector<bool> prn;
    int32_t           prnRate;
    std::vector<bool> prn_ovl;
    std::vector<bool> prn_ovl_data;
    std::vector<bool> prn_ovl_data_boc;
    int32_t           prn_ovl_data_bocRate;
    std::vector<bool> ovl;
    std::vector<bool> boc;
    int32_t           dataSymbLen;
    std::vector<bool> data_mes;
};

void dataListCreator(char* filename, std::vector<bool> data, char* signalname,
    int rate) {
    int L = data.size();
    int bytes = ceil(L/8.0)+1;

    FILE* fid = fopen(filename, "w");
    fprintf(fid, "{TYPE:SMU-DL}{COMMENT:_signal_=%s,_rate_%d_sym/s}",
        signalname, rate);
    fprintf(fid, "{DATA_BITLENGTH:%d}{DATA_LIST-%d:#", L, bytes);

    uint8_t newcoolbyte;
    int32_t l = 0;
    while (l < L) {
        newcoolbyte = 0;
        for (int i = 0; i < 8; i++) {
            if (l < L) {
                newcoolbyte |= (data[l] << (8-1 - i));
                l++;
            }
        }
    }
}
```

```

    }
    fputc(newcoolbyte, fid);
}
fprintf(fid, "}");
fclose(fid);
}

void generatePRN(SignalType type, int32_t prn_num, std::vector<bool> &prn,
int32_t &prnRate) {
    prn.clear();

    if (type == GnssNone) {
        prnRate = 0;
        return;
    }

    double IN_CLK = 106.344e6;

    Correlator2BitModel cor;
    Replica2 rep;
    rep.setHwChnl((Replica2Regs*)&cor);
    rep.setIN_CLK(IN_CLK);

    rep.initRepChannel(type, prn_num-1);

    cor.IRQ();

    for (int i = 0; i < int(rep.get_cntr_length()+1); i++) {
        prn.push_back(cor.getCodeOutput());

        cor.codeClk();
    }
    prnRate = rep.prnRate;
}

void wrapOvl(std::vector<bool> ovl, std::vector<bool> const prn, std::vector<
bool> &prn_ovl){

    if (ovl.size() == 0) {
        ovl.push_back(false);
    }
}

```

```

prn_ovl.clear();

for (uint32_t i = 0; i < ovl.size(); i++) {
    for (uint32_t j = 0; j < prn.size(); j++) {
        prn_ovl.push_back(ovl[i] ? (!prn[j]) : prn[j]);
    }
}

}

void wrapData(std::vector<bool> data_mes, std::vector<bool> prn_ovl, int32_t
dataSymbLen, std::vector<bool> &prn_ovl_data){
    if (data_mes.size() == 0) {
        data_mes.push_back(false);
        dataSymbLen = prn_ovl.size();
    }
    prn_ovl_data.clear();

    int32_t j = 0;
    for (uint32_t i = 0; i < data_mes.size(); i++) {
        for (int32_t k = 0; k < dataSymbLen; k++) {
            prn_ovl_data.push_back(data_mes[i] ? (!prn_ovl[j%prn_ovl.size()]) :
                prn_ovl[j%prn_ovl.size()]);
            j++;
        }
    }
}

void combineComponentsTDMA(std::vector<bool> data, int32_t dataRate, std::
vector<bool> pilot, int32_t pilotRate, std::vector<bool> &out, int32_t &
outRate){
    int32_t p2d = pilotRate / dataRate;
    if (p2d == 0) {
        out = data;
        outRate = dataRate;
    } else {
        out.clear();
        outRate = (dataRate > pilotRate) ? (2*dataRate) : (2*pilotRate);

        int32_t dataLen = p2d*data.size();
        int32_t pilotLen = pilot.size();
    }
}

```

```

int32_t outLen = (dataLen > pilotLen) ? (2*dataLen) : (2*pilotLen);

int32_t jp = 0;
int32_t jd = 0;

while (out.size() < (uint32_t)outLen) {
    for (int32_t j = 0; j < p2d; j++) {
        out.push_back(data[jd]);
    }
    jd = (jd + 1) % data.size();

    for (int32_t j = 0; j < p2d; j++) {
        out.push_back(pilot[jp]);
        jp = (jp + 1) % pilot.size();
    }
}

if (out.size() != (uint32_t)outLen) {
    printf("combineComponentsTDMA::Error out.size() != outLen\n");
    exit(0);
}
}
}

void wrapBOC(std::vector<bool> boc, std::vector<bool> const in, int32_t const
inRate, std::vector<bool> &out, int32_t &outRate){
    if (boc.size() == 0) {
        boc.push_back(false);
    }
    out.clear();

    for (uint32_t i = 0; i < in.size(); i++) {
        for (uint32_t j = 0; j < boc.size(); j++) {
            out.push_back(in[i] ? (!boc[j]) : boc[j]);
        }
    }
    outRate = inRate * boc.size();
}

char filename[] = "/tmp/xxxLxCAd_12345678.dm_iqd";

```

```

char signal[] = "xxxLxCAd_12345678";

GenSgnCompVars p;
GenSgnCompVars d;

std::vector<bool> data_and_pilot;

std::vector<GenSgnType> types = {GenSgnType::GlnL10F, GenSgnType::GlnL10C};

for (auto const &it : types) {
    int16_t prnMax = 32;

    switch (it) {
    case GenSgnType::GlnL10F:
        prnMax = 0;
        d.type = GlnL10F;
        d.ovl = {};
        d.boc = {};
        d.dataSymbLen = 511*10;
        d.data_mes = dataL10F;

        p.type = GnssNone;
        p.ovl = {};
        p.boc = {};
        p.dataSymbLen = 1;
        p.data_mes = {};
        break;

    case GenSgnType::GlnL10C:
        prnMax = 32;
        d.type = GlnL10Cd;
        d.ovl = {0, 1};
        d.boc = {};
        d.dataSymbLen = 1023*2;
        d.data_mes = {};

        p.type = GlnL10Cp;
        p.ovl = {};
        p.boc = {0, 1};
        p.dataSymbLen = 1;
    }
}

```

```

    p.data_mes = {};
    break;

case GenSgnType::G1nL20C:
    prnMax      = 32;
    d.type      = GnssNone;
    d.ovl       = {};
    d.boc       = {};
    d.dataSymbLen = 1;
    d.data_mes  = {};

    p.type      = G1nL20C;
    p.ovl       = {0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1,
                  0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1,
                  0, 0, 0, 1, 0, 0, 0, 1, 0};
    p.boc       = {0, 1};
    p.dataSymbLen = 1;
    p.data_mes  = {};
    break;

default:
    printf("Unknown_gen_sgn_type\n");
    break;
}
for (int32_t prn_num = 0; prn_num <= prnMax; prn_num++) {

    generatePRN(d.type, prn_num, d.prn, d.prnRate);
    generatePRN(p.type, prn_num, p.prn, p.prnRate);

    wrapOvl(d.ovl, d.prn, d.prn_ovl);
    wrapOvl(p.ovl, p.prn, p.prn_ovl);

    wrapData(d.data_mes, d.prn_ovl, d.dataSymbLen, d.prn_ovl_data);
    wrapData(p.data_mes, p.prn_ovl, p.dataSymbLen, p.prn_ovl_data);

    wrapBOC(d.boc, d.prn_ovl_data, d.prnRate, d.prn_ovl_data_boc, d.
            prn_ovl_data_bocRate);
    wrapBOC(p.boc, p.prn_ovl_data, p.prnRate, p.prn_ovl_data_boc, p.
            prn_ovl_data_bocRate);
}

```

```
std::vector<bool> out;
int32_t outRate;

combineComponentsTDMA(d.prn_ovl_data_boc, d.prn_ovl_data_bocRate, p.
    prn_ovl_data_boc, p.prn_ovl_data_bocRate, out, outRate);
sprintf(signal, "%s_%03d", (char*)GenSgnTypeStr[(int)it], prn_num);
sprintf(filename, "/tmp/%s.dm_iqd", signal);

dataListCreator((char*)filename, out, (char*)signal, outRate);
}
}
```