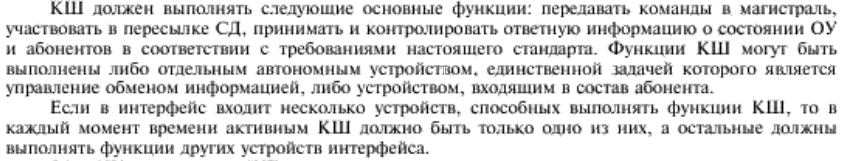
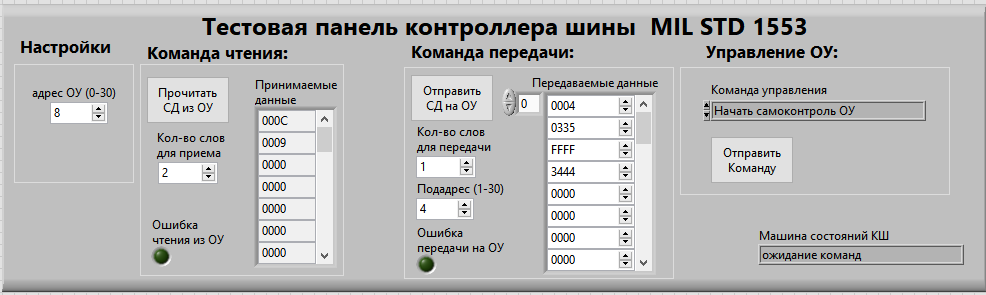
1 Разработка программы Контролера шины (КШ)

* 1. Основные функции, выполняемые КШ



Контроллер шины ожидает команд от интерфейса верхнего уровня (рис 1)

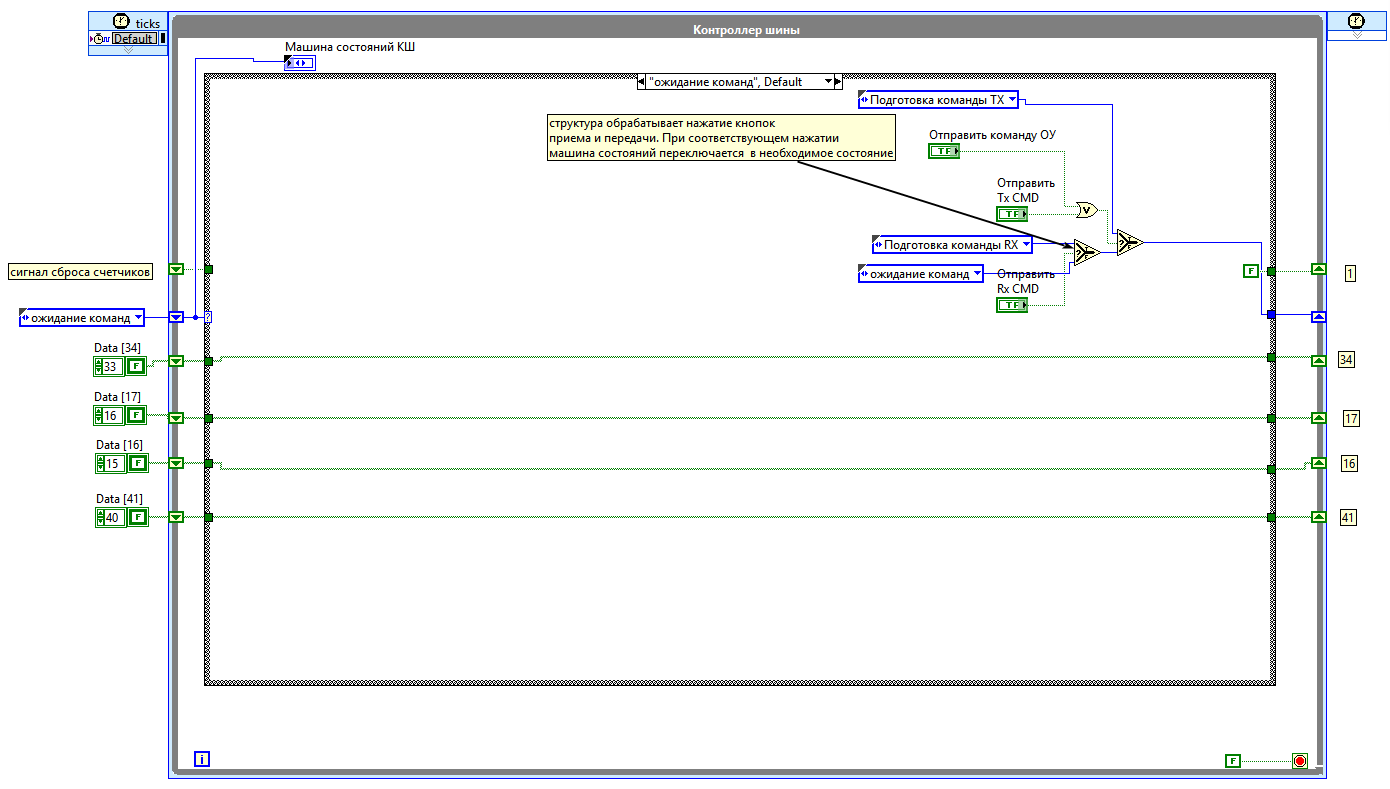


1.2 За основу механизма обработки команд взята так называемая машина состояний. Она состоит из case структуры, в каждом из состоянии которой выполняется своя функция:

* Ожидание команд
* Подготовка команды RX
* Подготовка команды TX
* Кодер манчестер2 КС
* Таймаут t2 между словами
* Передача
* Ожидание rising СД
* Замер длительности true СД
* Чтение бит СД
* Ожидание rising ОС
* Замер длительности false ОС
* Замер длительности true ОС
* Чтение бит ОС
* Манчестер декодер ОС
* Чтение ОС
* Подготовка СД
* Кодирование СД
* Отправка СД

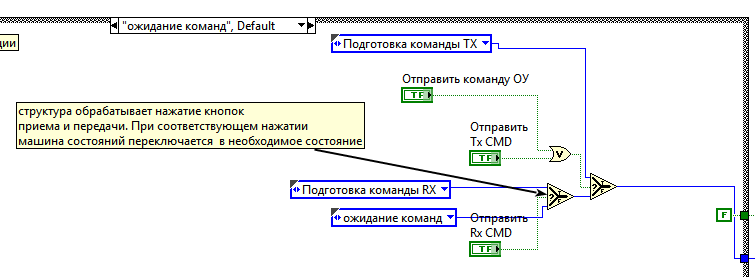
Рассмотрим каждое состояние в отдельности.

1.2.1 После запуска программы, машина состояний находится в положении «ожидание» (рис2)

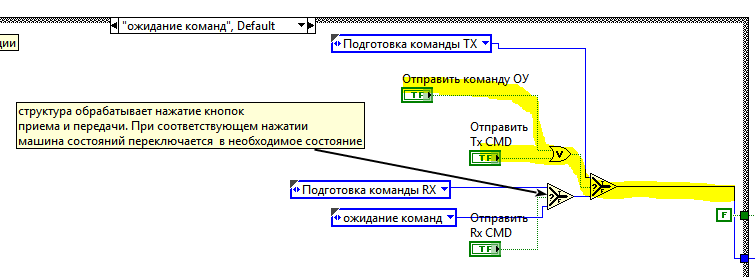


Внутрений код обрабатывает нажатие кнопок приема, передачи и отправки команды управления (КУ).

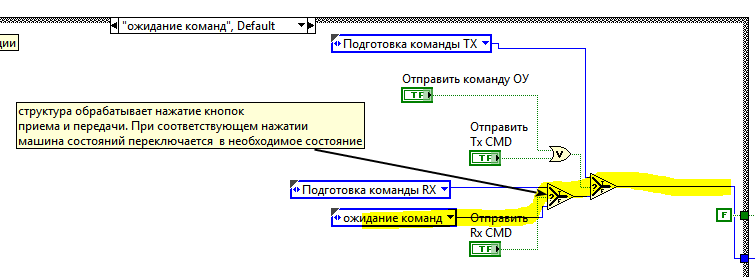
При нажатии кнопки «Отправить Tx CMD» сигнал «истина» поступает в функцию «select». Соответственно на выход передается текст «Подготовка команды TX». Машина состояний переключается в состояние «Подготовка команды TX». (рис3)



При нажатии кнопки «Отправить Rx CMD» или «Отправить команду ОУ» сигнал «истина» поступает в функцию «ИЛИ», а затем через «Select» на выход передается текст «Подготовка команды RX», а машина состояний переключается в состояние «Подготовка команды RX». (рис 4)

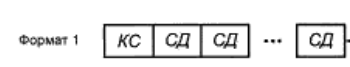


В случае, если никаких кнопок не нажимается, на выход передается текст «ожидание» и машина состояний переходит в соответствующую позицию. (рис 5)

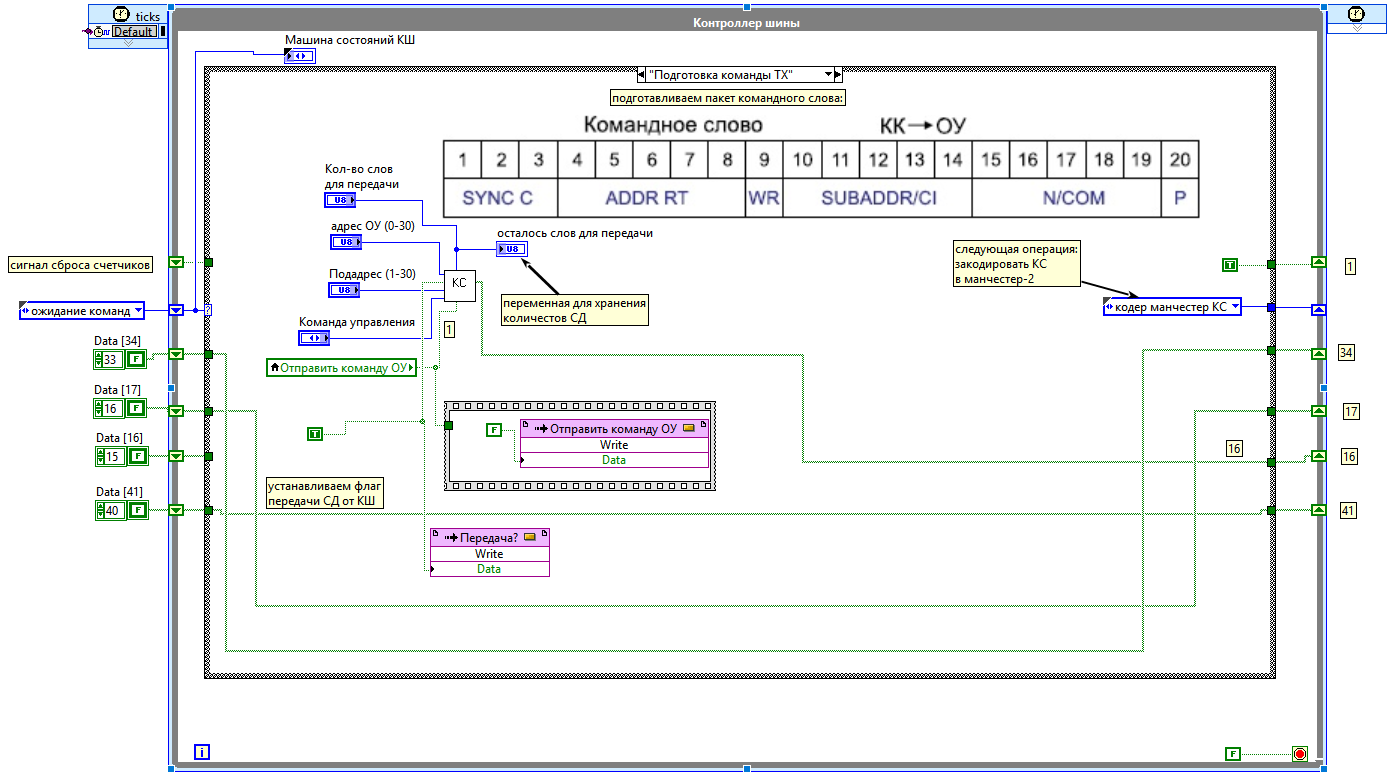


1.3 Состояние «Подготовка команды TX»

Для передачи данных от КШ к ОУ необходимо сформировать пакет формата 1: (рис 6)

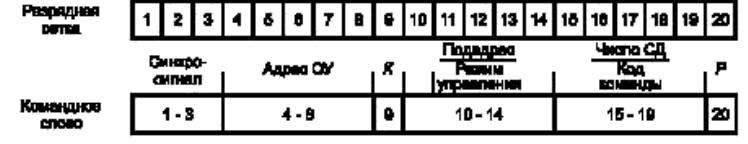


Который состоит из командного слова (КС) и слов данных (СД), идущих без пауз друг за другом. Код сборки этого пакета находится в состояниях

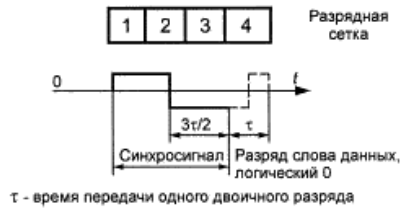


Здесь представлен ВП (виртуальный прибор) «подготовка контрольного слова».

ВП (виртуальный прибор) «подготовка контрольного слова» преобразует данные с элементов управления в 20 битный пакет, структура которого показана на рис 8



Первые три бита это сигнал пословной синхронизации командного слова. (рис 9).

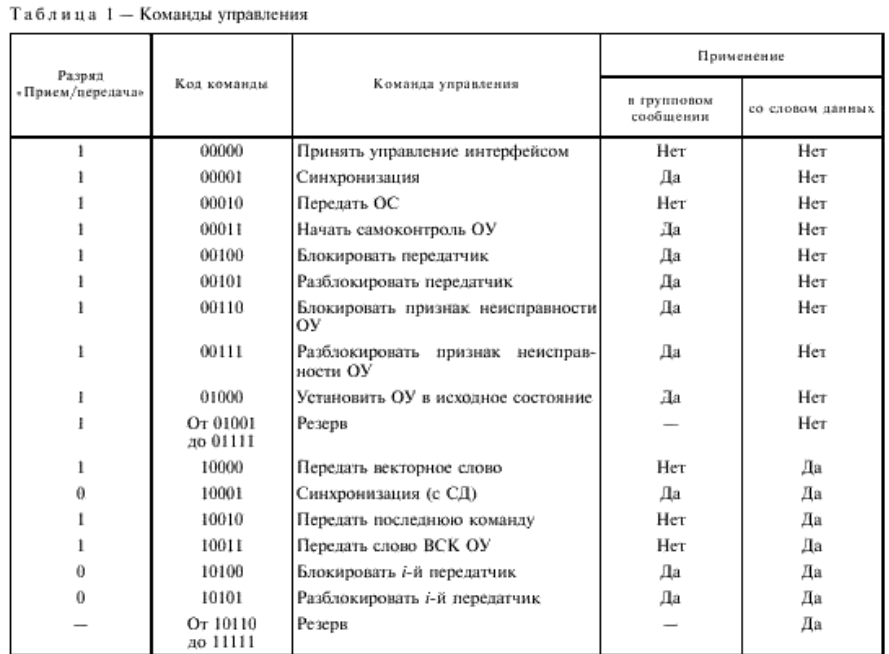


Он не кодируется с помощью манчестер-2, поэтому 1й и 3й биты можно опустить, а добавить их на этапе кодировки.

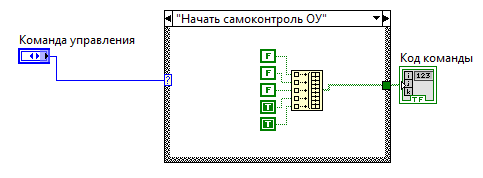
Биты с 4 по 8й – адрес оконечного устройства(ОУ), которому КШ передает данные. Преобразуем цифровое значение номера ОУ с помощью функции «number to array» в бинарный код.

9й бит – разряд приема\передачи. «Истина» говорит о передаче данных, «ложь» - прием.

С 10го по 14й бит указывается подадрес или режим управления. Режим управления записывается в 10й бит, при этом в биты 15-19 записывается код команды в бинарном виде. Этот код собирается согласно таблицы (рис 10)



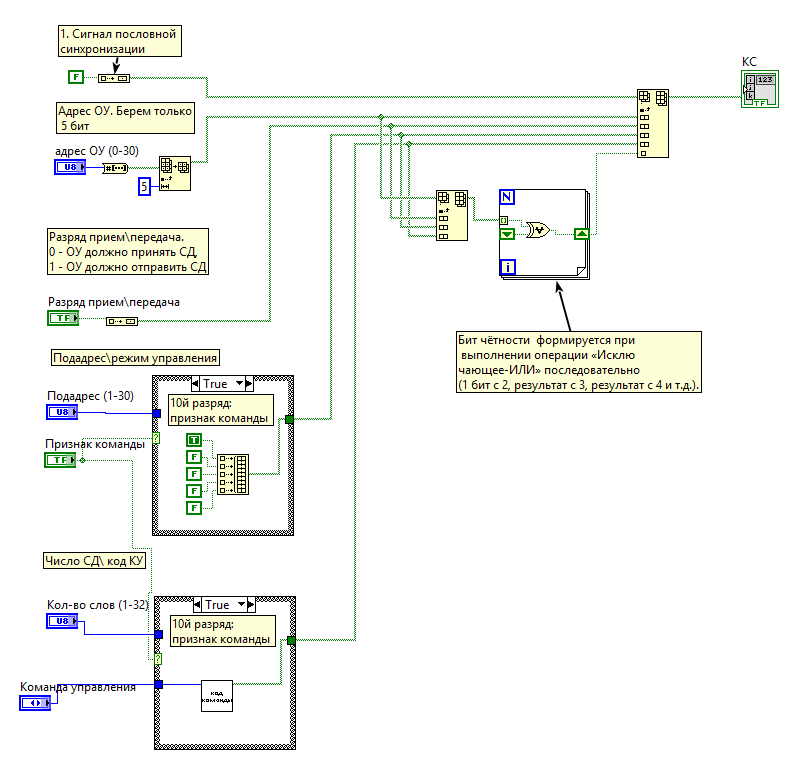
С помощью подВП «команда управления». Этот ВП состоит из «case» структуры, выбирающей соответствующий набор пяти бит. (рис 11).



Если же обсуждаемое слово не содержит команды управления (КУ) то в биты 15-19 записывается число запрашиваемых у ОУ слов данных (СД). А в биты 10-14 подадрес ОУ.

Последний, 20й бит – это бит четности. Он формируется при выполнении операции «исключающее или» последовательно со всеми битами, исключая сигнал пословной синхронизации.

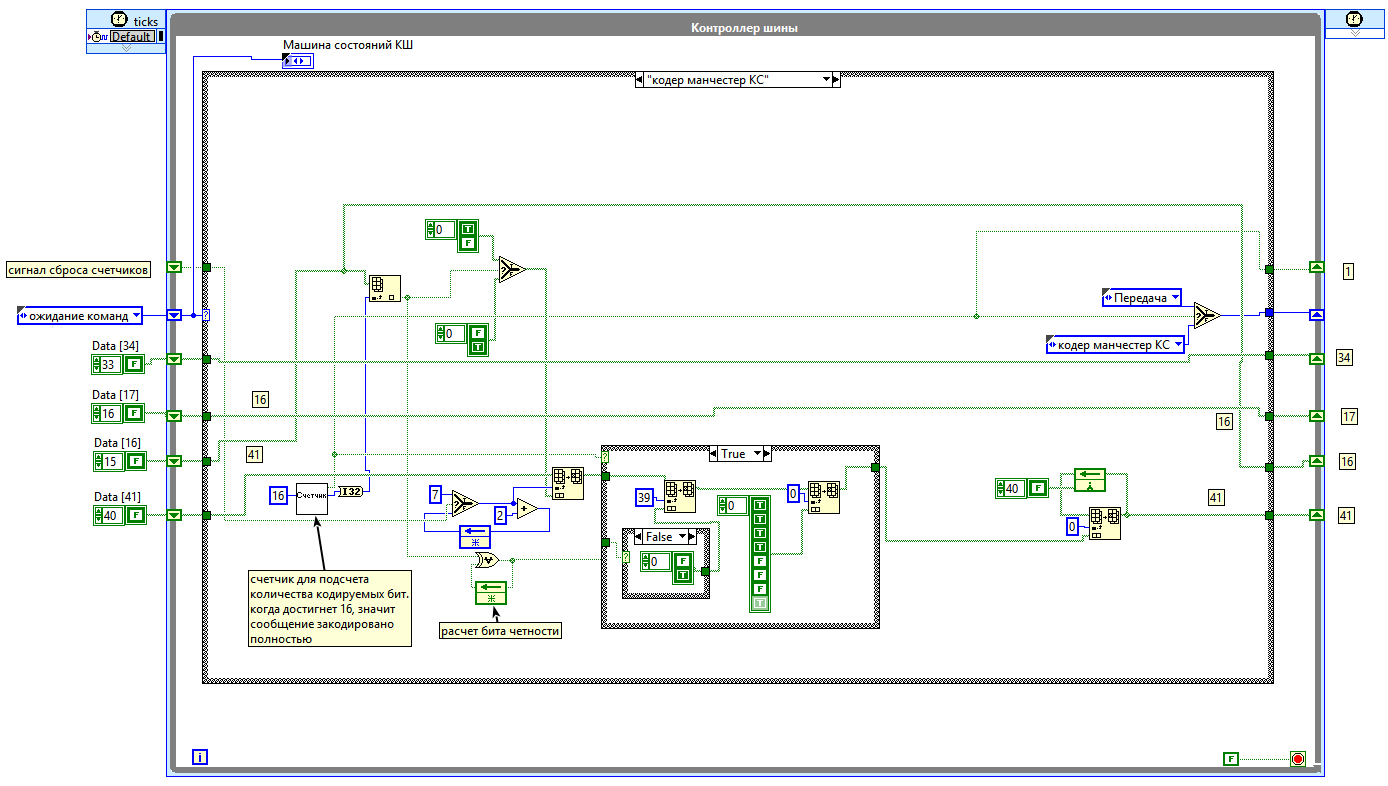
Все получившиеся биты с помощью функции «Insert to array» составляются в цепочку друг за другом, получая КС (рис 12).



Следующей операцией идет кодирование КС кодом манчестер-2.

1.4 Состояние «Кодер Манчестер КС»

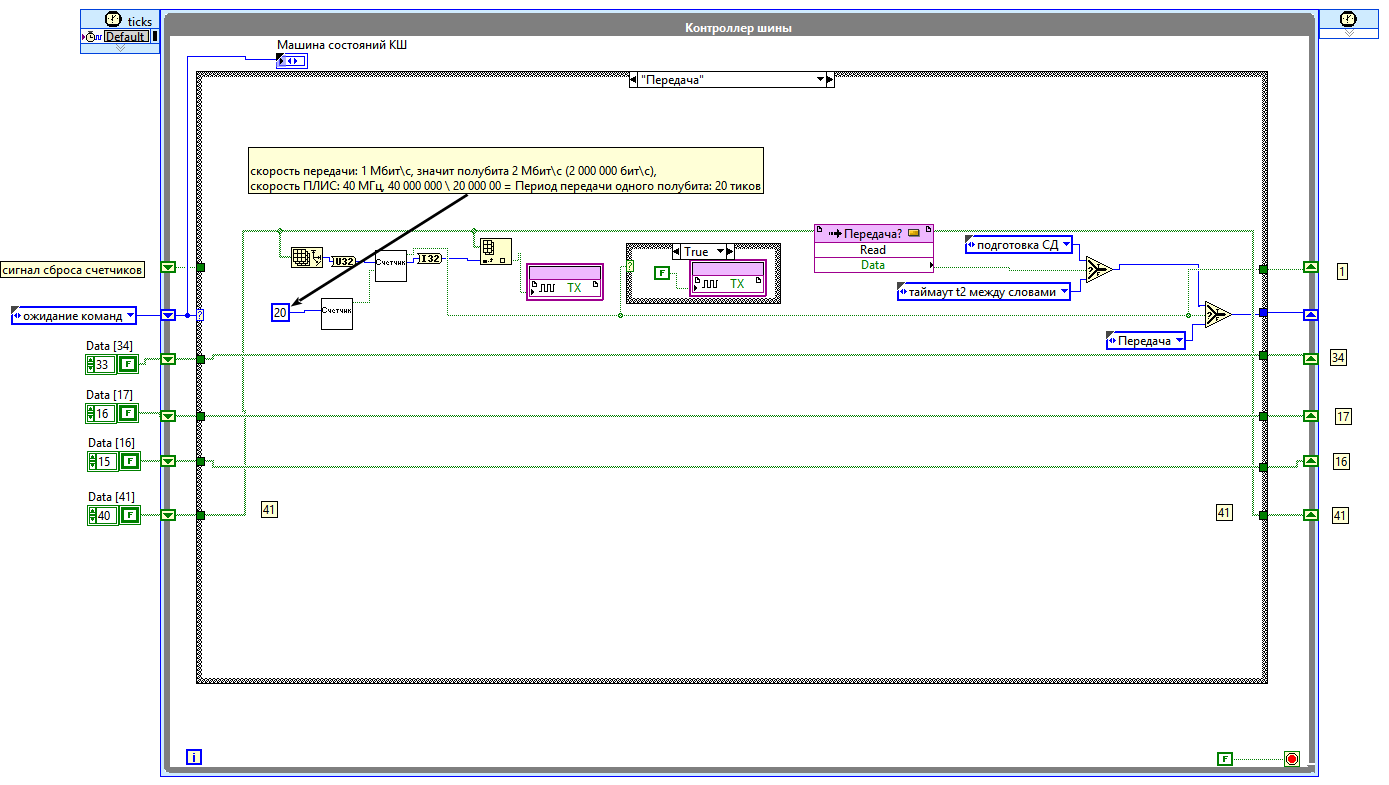
В этом состоянии имеется счетчик закодированных бит. Каждый бит кодируется последовательностью полубит. Когда счетчик бит достигнет 16, в массив добавляется синхросигнал и бит четности.



* "Истина" кодируется битами "истина" и "ложь" идущими друг за другом.
* "Ложь" кодируется битами в обратном порядке

1.5 Состояние «Передача».

В этом состоянии работают два счетчика: переданных бит и счетчик длительности полубита.



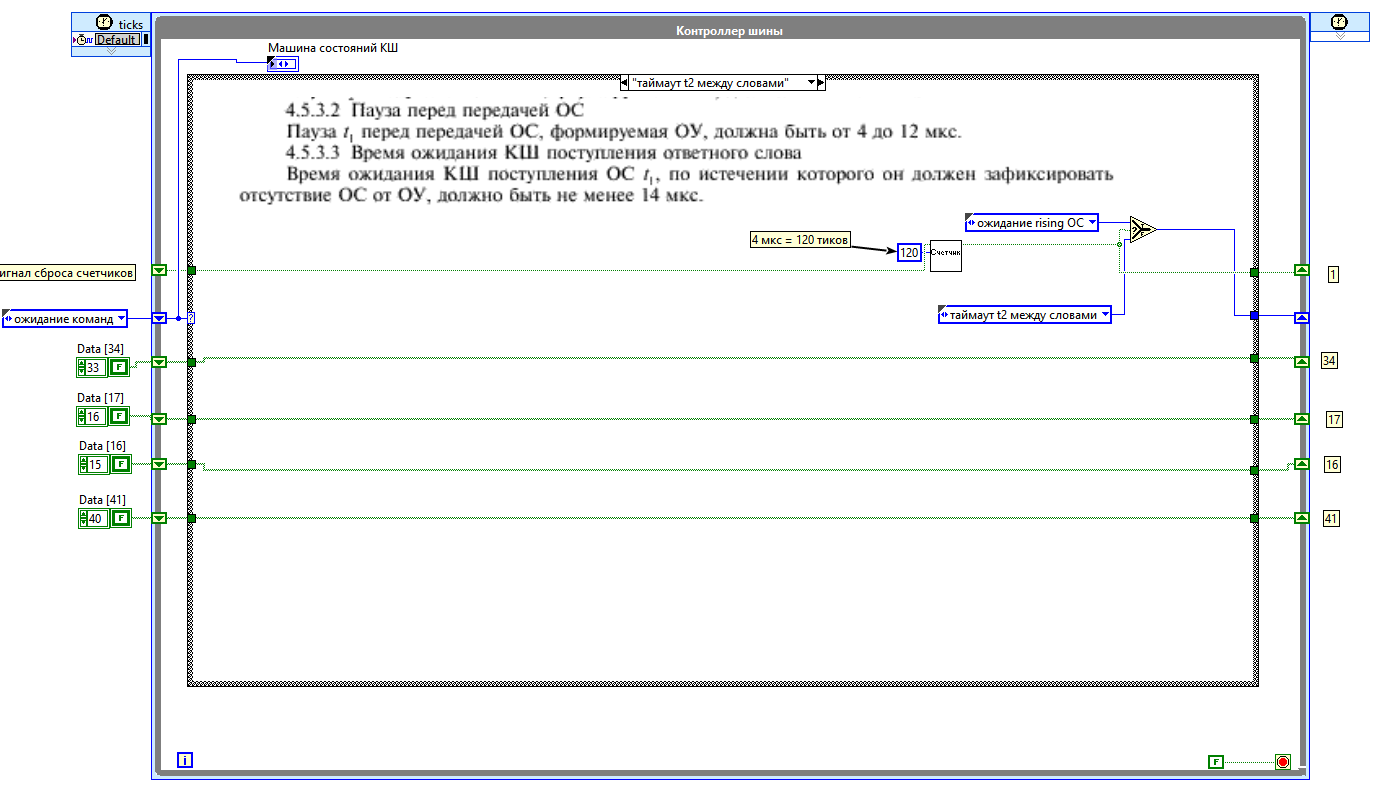
Скорость передачи: 1 Мбит\с, значит полубита 2 Мбит\с (2 000 000 бит\с), скорость ПЛИС: 40 МГц, 40 000 000 \ 20 000 00 = Период передачи одного полубита: 20 тиков.

Записываем данные в линию передачи TX.

Когда счетчик переданных бит достигнет конца, то выполняется проверка: если флаг передачи в контрольном слове – истина, значит следующим состоянием будет подготовка слова данных. Если же ложь, тогда таймаут t2 и последующее ожидание ответного слова.

1.6 Состояние Таймаут t2 между словми

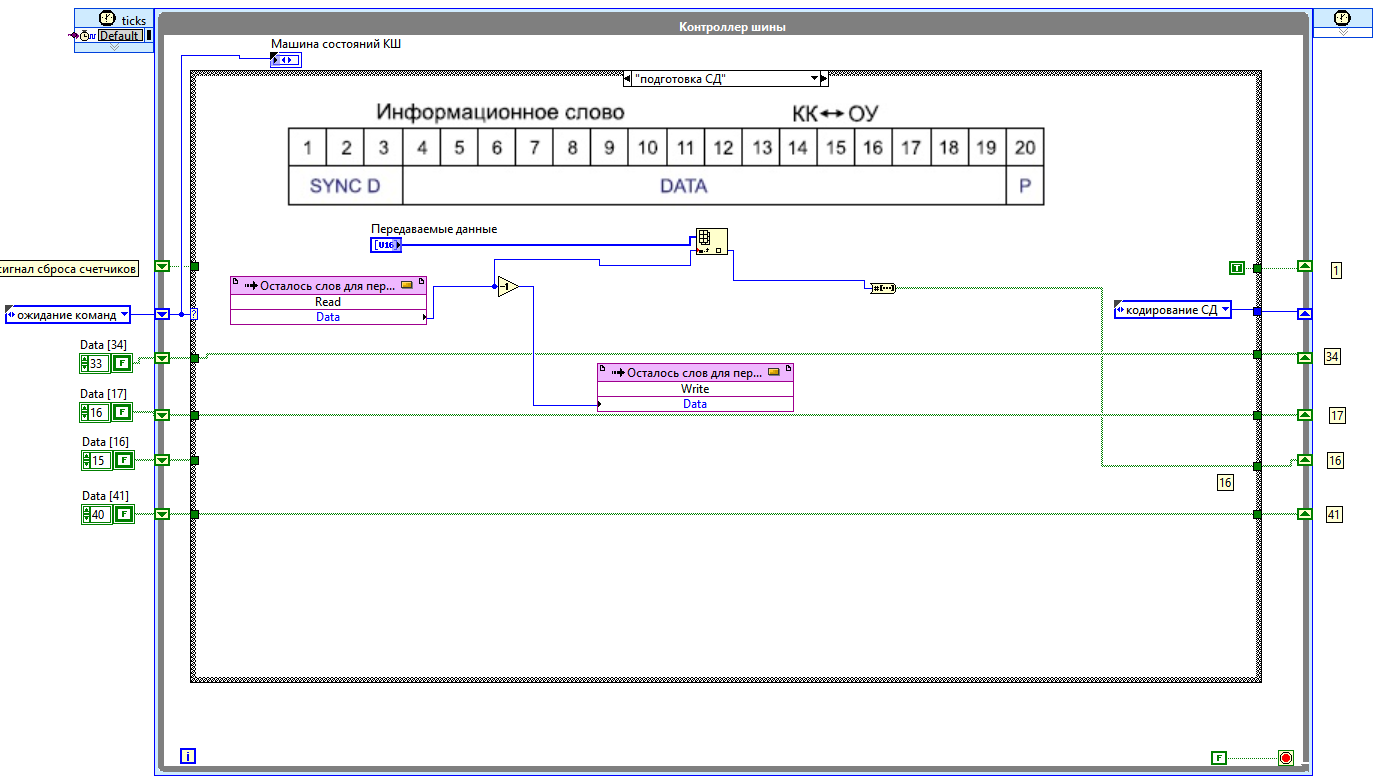
Согласно стандарту, необходимо выдержать таймаут 4 мкс между словами. 4 мкс при частоте ПЛИС 40 МГц это 120 тиков.



После выдержки таймаута следующее состояние ожидание rising ОС.

1.7 Состояние «Подготовка СД»

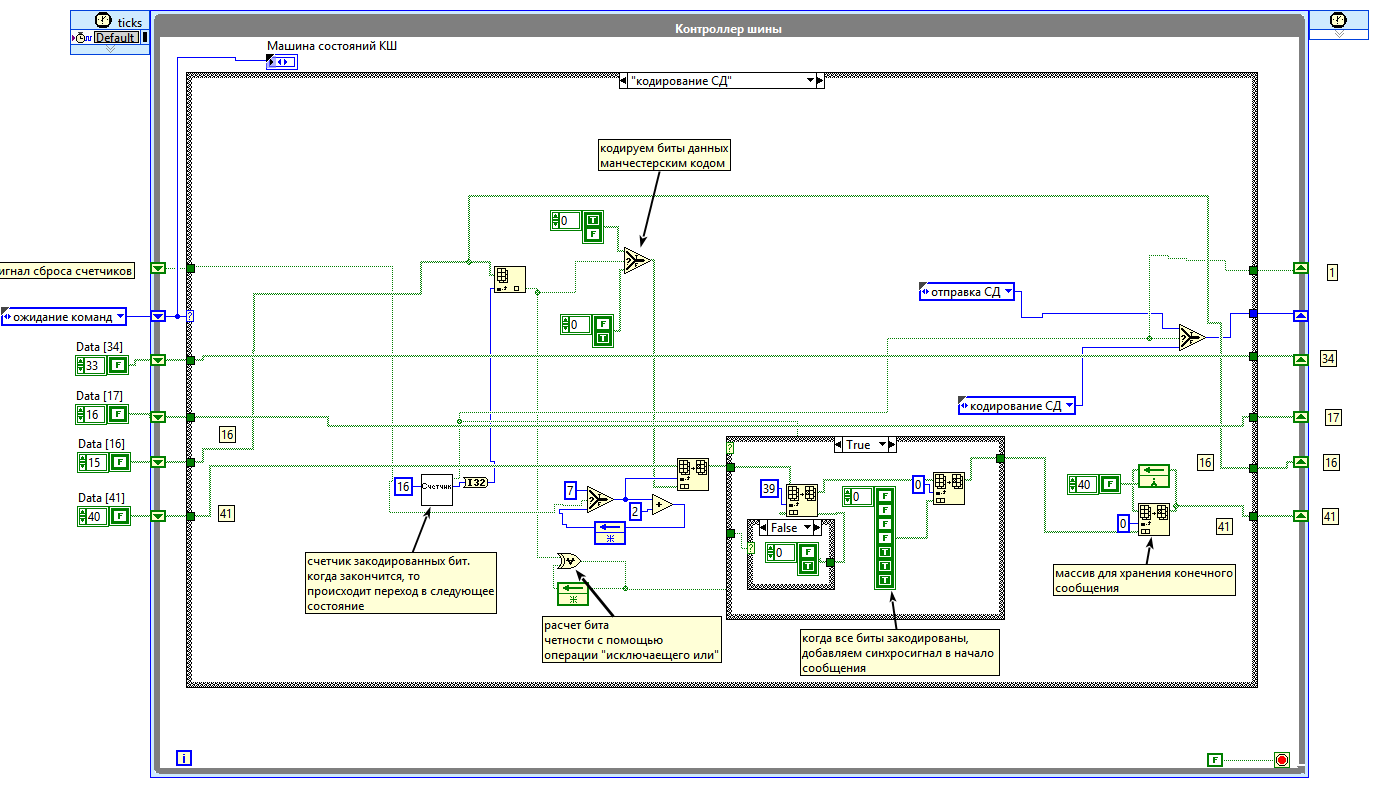
Преобразуем данные из массива для передачи в бинарный массив 16 бит.



Следущее состояние – кодирование СД.

1.8 Состояние «кодирование СД»

В этом состоянии имеется счетчик закодированных бит. Каждый бит кодируется последовательностью полубит (рис 13). Когда счетчик бит достигнет 16, в массив добавляется синхросигнал и бит четности.



* "Истина" кодируется битами "истина" и "ложь" идущими друг за другом.
* "Ложь" кодируется битами в обратном порядке

Следующим состоянием является отправка СД.

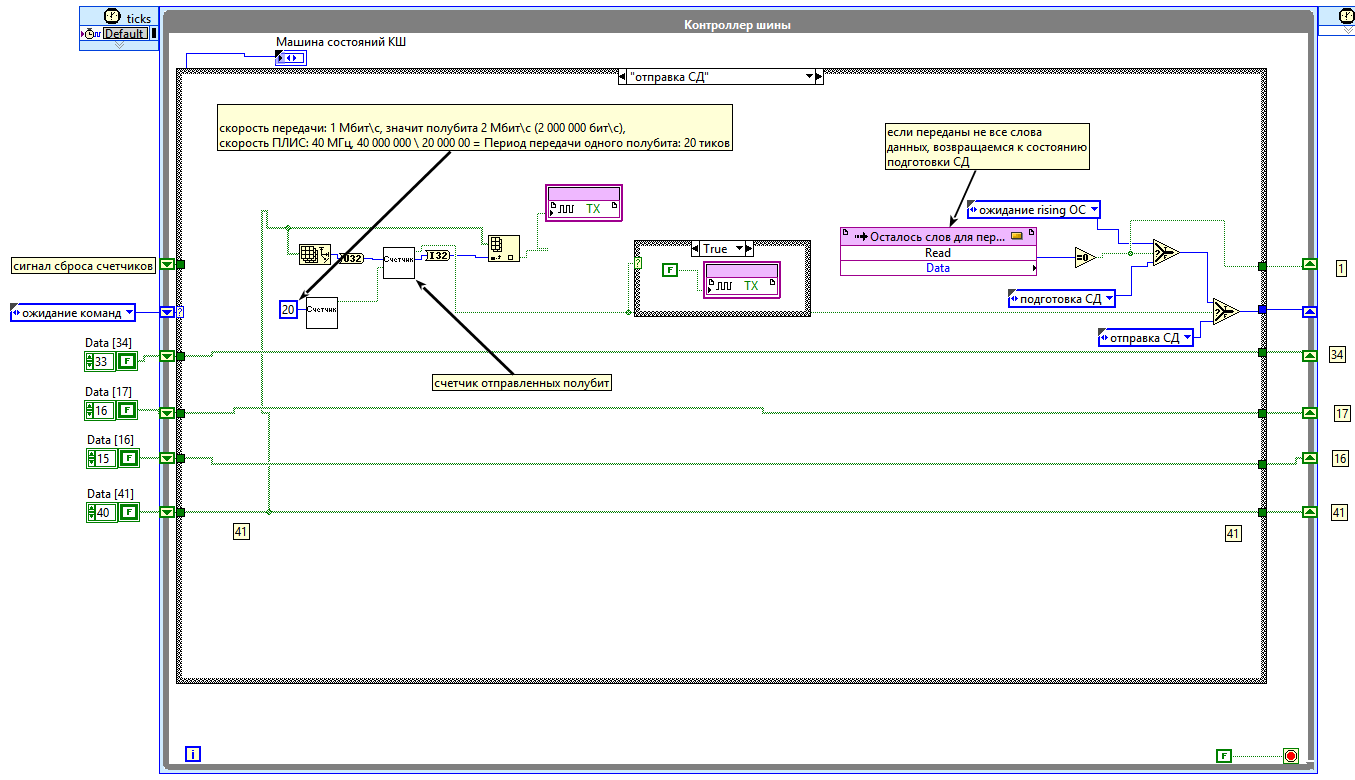
1.9 Состояние «отправка СД».

В этом состоянии работают два счетчика: переданных бит и счетчик длительности полубита.

Скорость передачи: 1 Мбит\с, значит полубита 2 Мбит\с (2 000 000 бит\с), скорость ПЛИС: 40 МГц, 40 000 000 \ 20 000 00 = Период передачи одного полубита: 20 тиков.

Записываем данные в линию передачи TX.

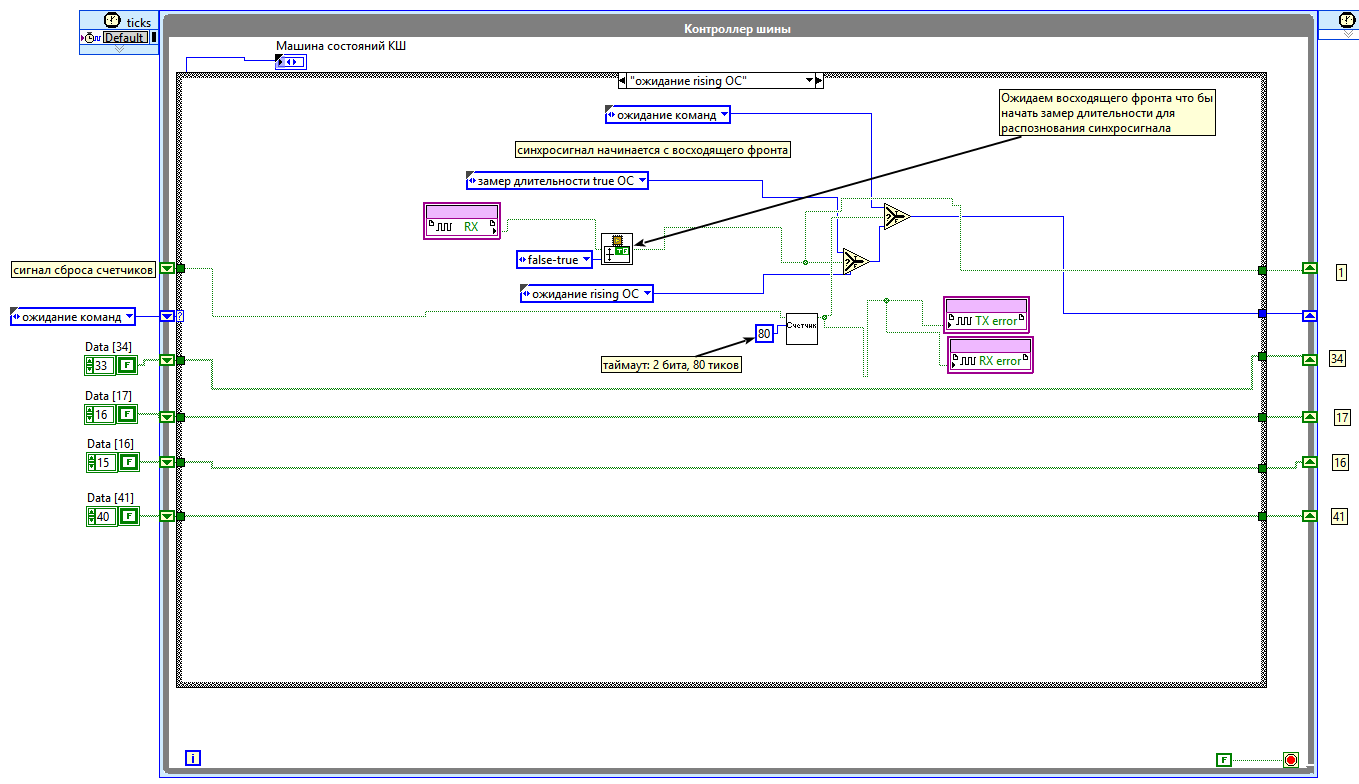
Когда счетчик переданных бит достигнет конца, то выполняется проверка: если переданы не все слова данных, то возврат к подготовке следующего слова данных. В противном случае переходим к ожиданию ответного слова.



1.9 Состояние «Ожидание rising OC»

В этом состоянии программа ожидает восходящего фронта что бы начать замер длительности для

распознавания синхросигнала.

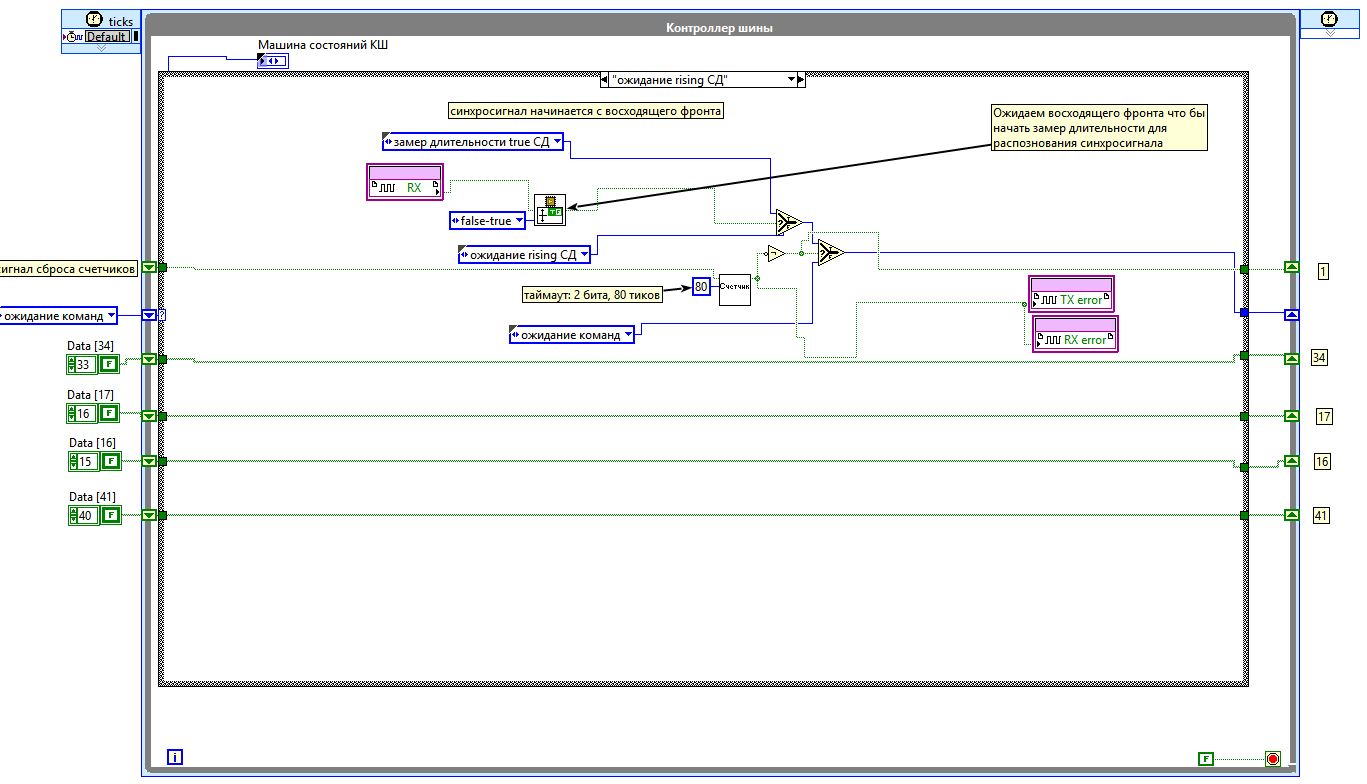


В случае если не обнаружено импульсов в течении 2 бит (80 тиков) включаем флаг ошибки по таймауту. Если же обнаружен нарастающий фронт – переходим в состояние замер длительности true ОС.

1.10 Состояние «Ожидание rising СД»

В этом состоянии программа ожидает восходящего фронта что бы начать замер длительности для

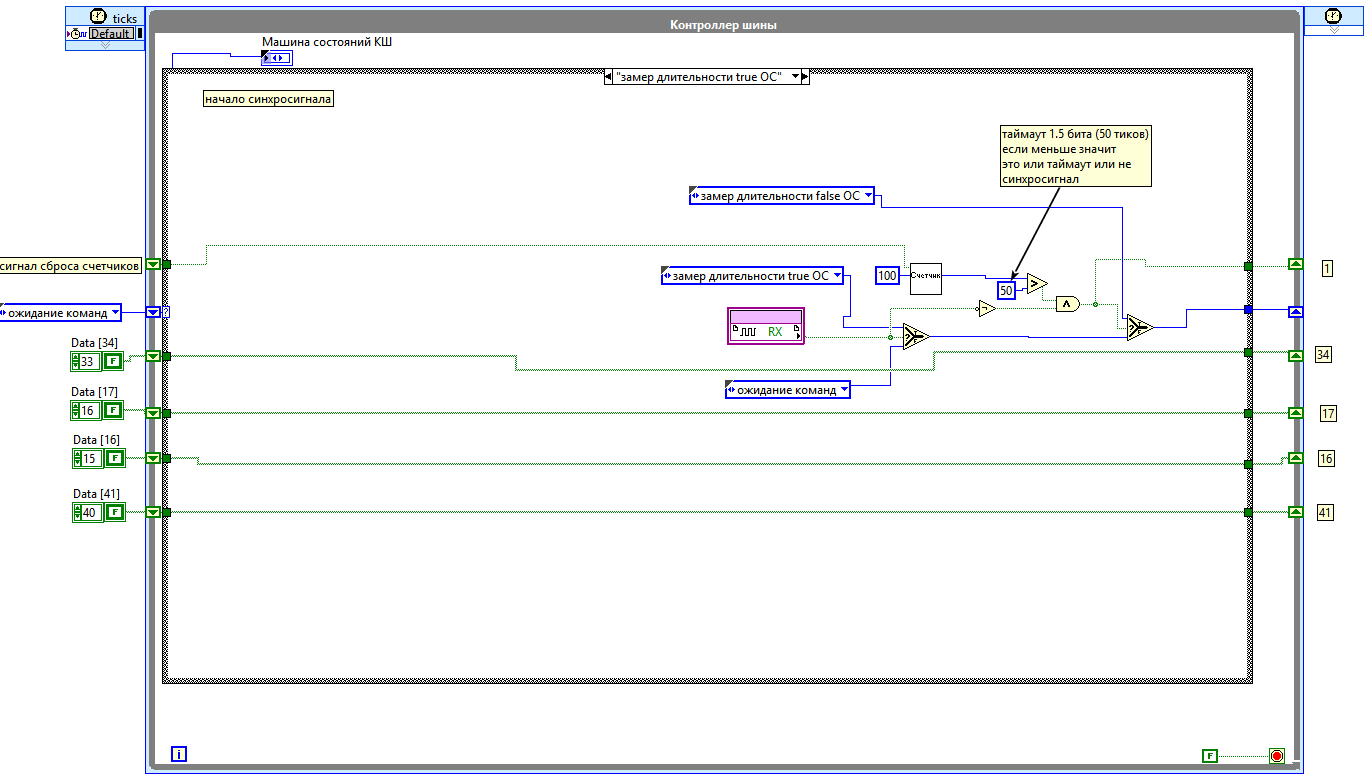
распознавания синхросигнала слова данных.



В случае если не обнаружено импульсов в течении 2 бит (80 тиков) включаем флаг ошибки по таймауту. Если же обнаружен нарастающий фронт – переходим в состояние замер длительности true СД.

1.11 Состояние «замер длительности true ОС»

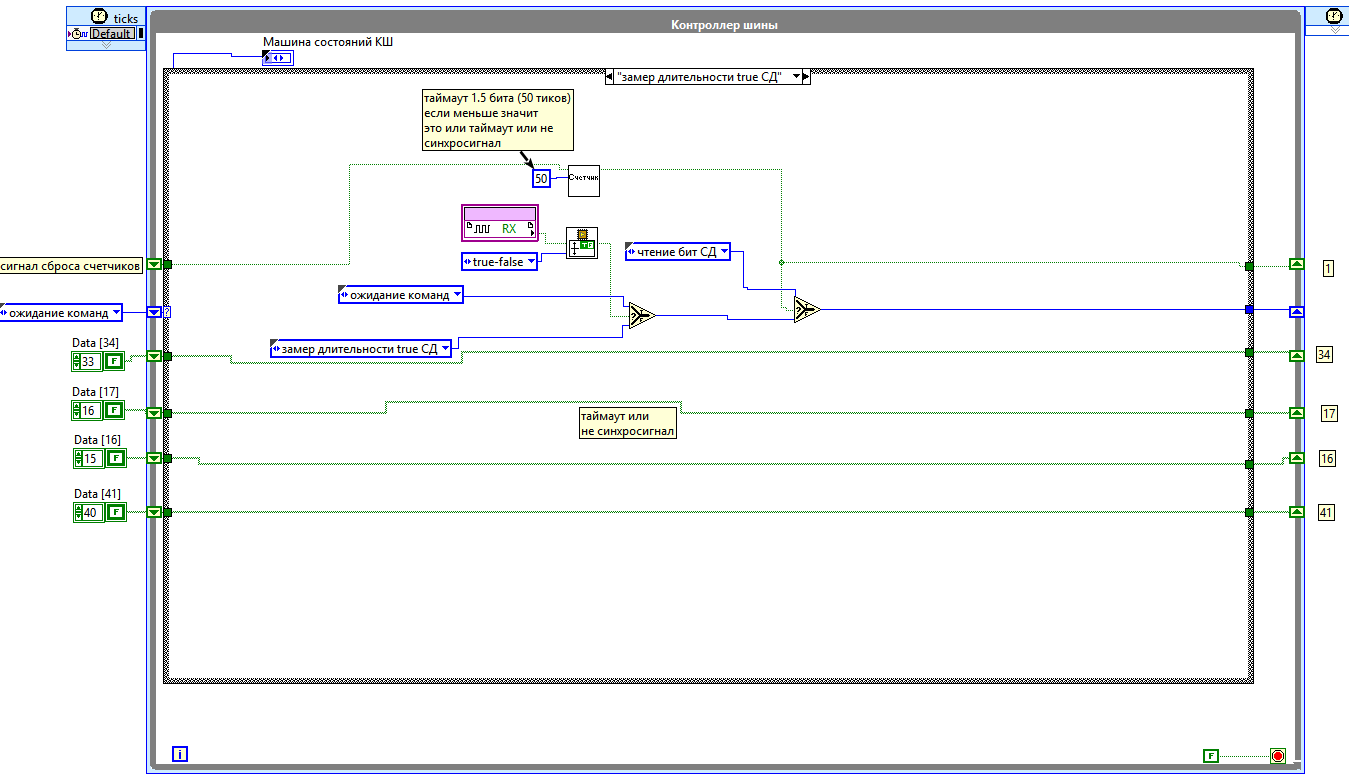
В этом состоянии программа ожидает спадающего фронта. Когда он происходит, то значение счетчика проверяется: если больше 1.5 бит – значит возможно это синхросигнал ответного слова.



Переходим в состояние замера длительности false ОС.

1.12 Состояние «замер длительности true СД»

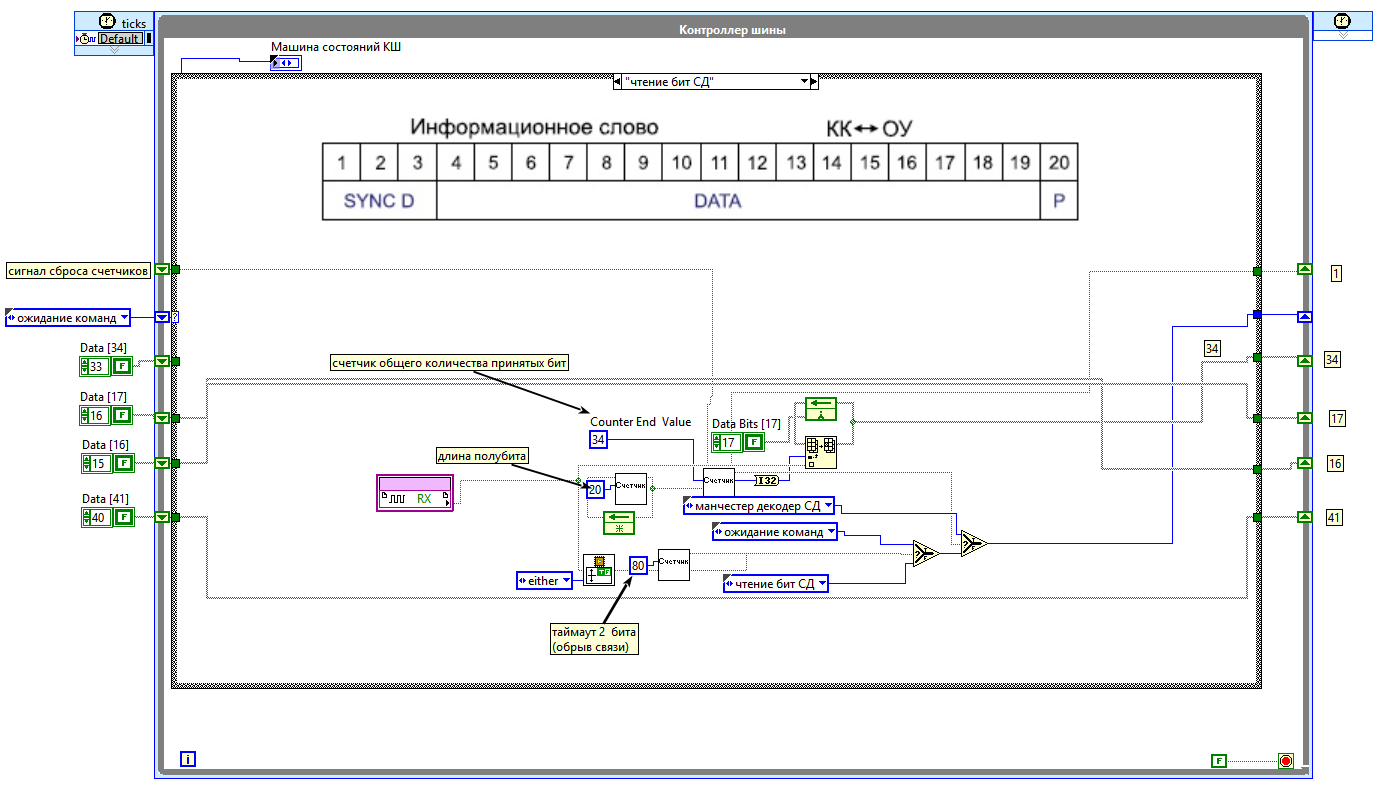
В этом состоянии программа ожидает спадающего фронта. Когда он происходит, то значение счетчика проверяется: если больше 1.5 бит – значит это синхросигнал слова данных.



Переходим в состояние чтения бит СД.

1.13 Состояние «Чтение бит СД»

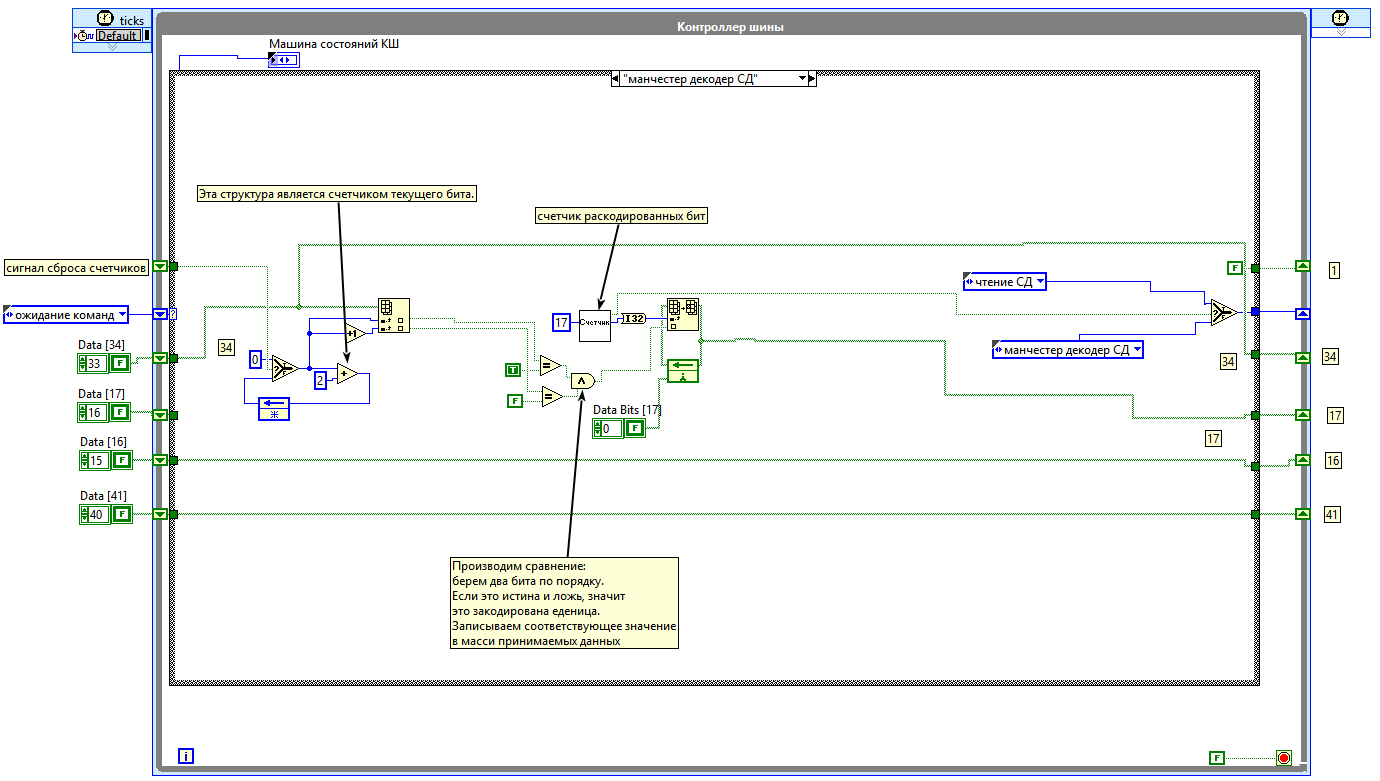
Программа использует два счетчика: счетчик длины полубита (20 тиков) и счетчик таймаута (80 тиков). Каждые 20 тиков считывается состояние линии и записывается в массив принимаемых данных.



Когда счетчик принятых полубит достигнет 34, переходим в состояние манчестер декодер СД.

1.14 Состояние манчестер декодер СД

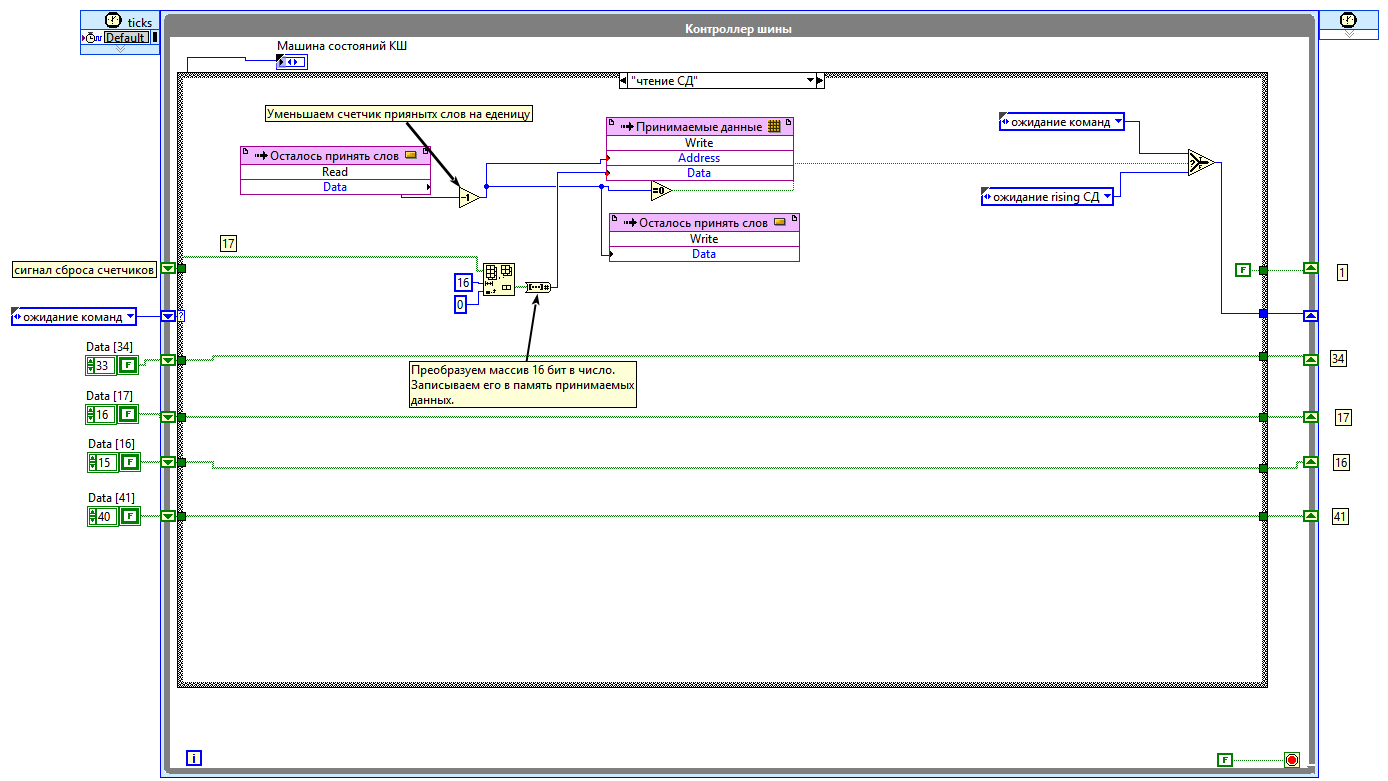
В начале состояния находится счетчик текущего бита. Он используется для считывания из принятого массива бит попарно. Далее происходит сравнение пары полубит с константами. Таким образом преобразуем манчестерский код в биты сообщения.



Следующим состоянием является чтение СД.

1.15 Состояние «чтение СД»

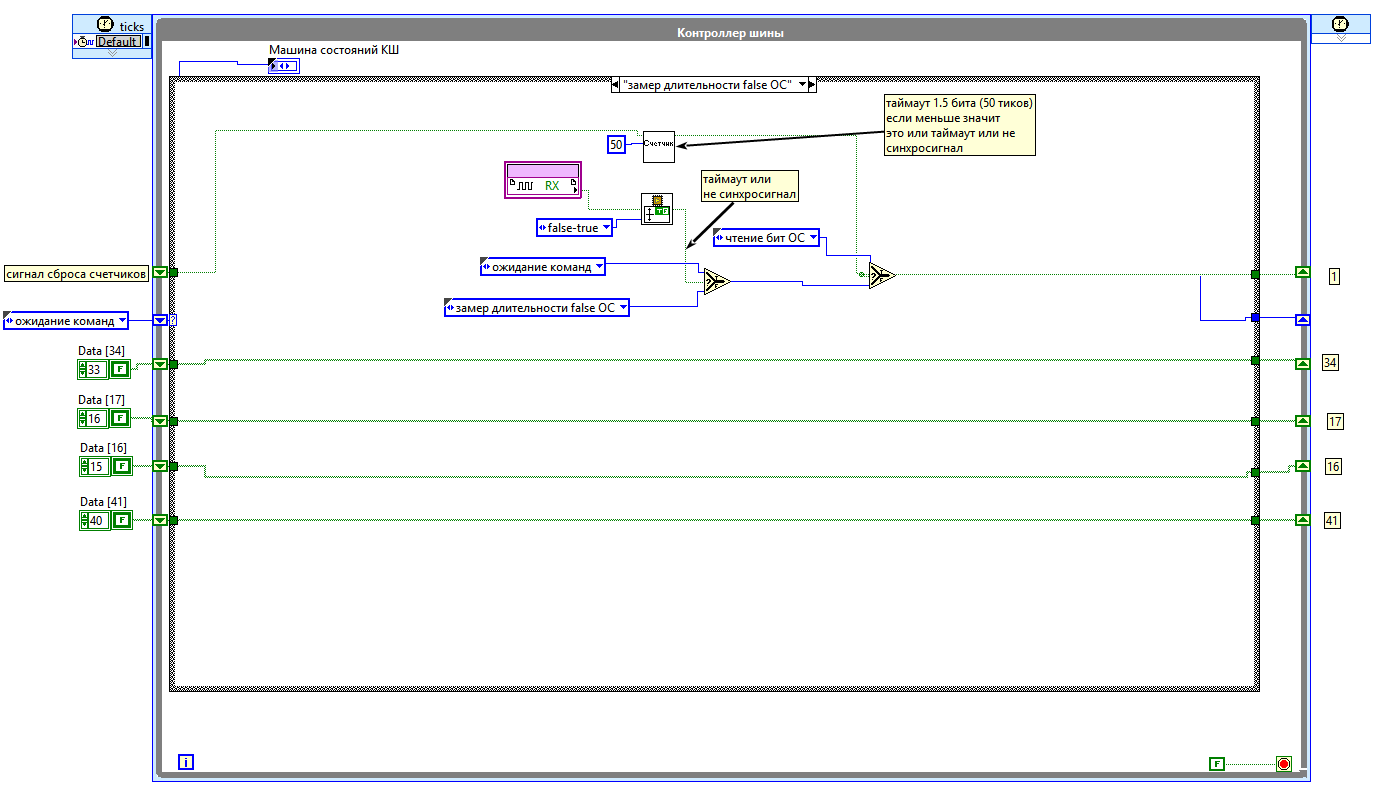
Преобразуем раскодированный массив 16 бит в число. Записываем его в память принимаемых данных. Уменьшаем счетчик принятых слов данных.



Если счетчик достиг нуля, то транзакция завершена. Переходим в ожидания команд. В противном случае переходим в состояние поиска синхросигнала СД.

1.16 Состояние «Замер длительности false OC»

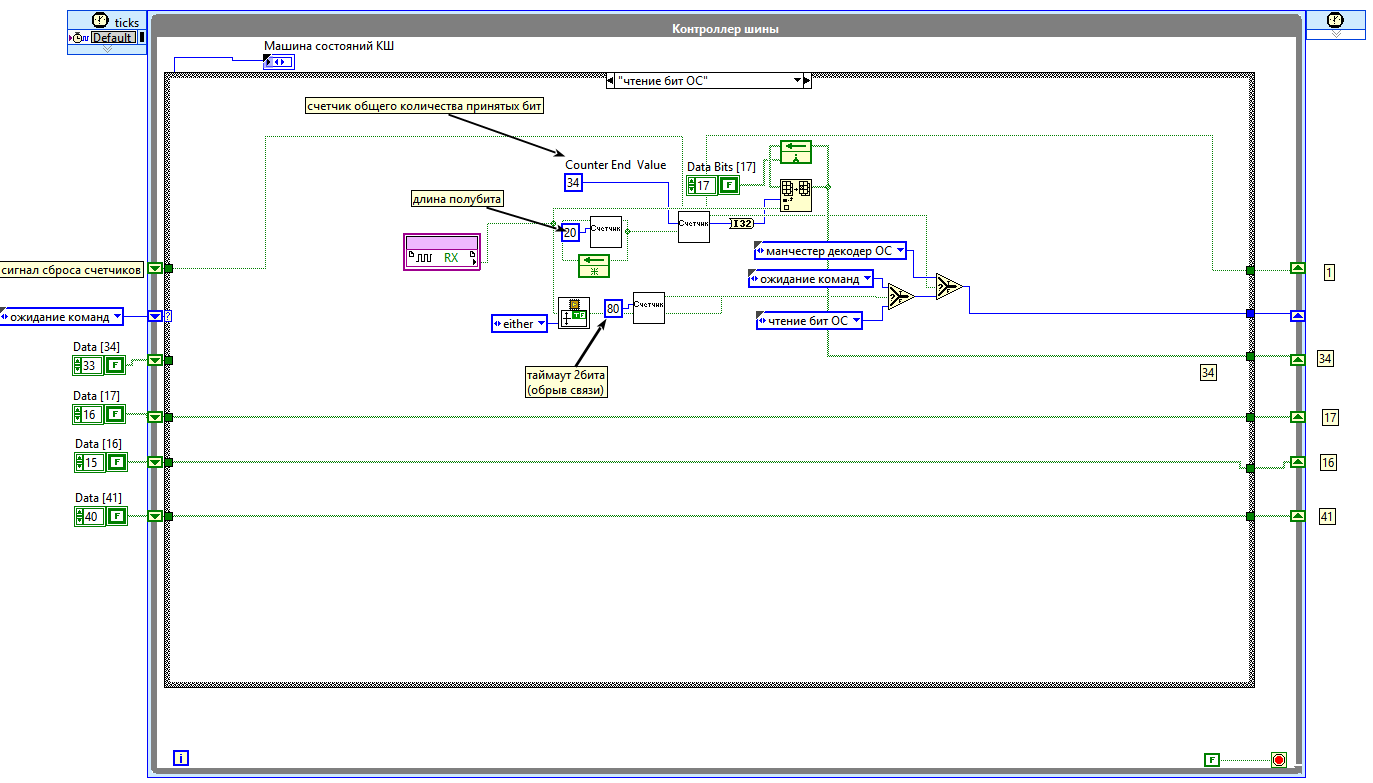
Программа ожидает нарастающего фронта. Если он произошел менее чем через 1.5 бита – значит это не синхросигнал ОС. Возвращаемся в ожидание команд.



В противном случае переходим в состояния чтения бит ОС.

1.17 Состояние «чтение бит ОС»

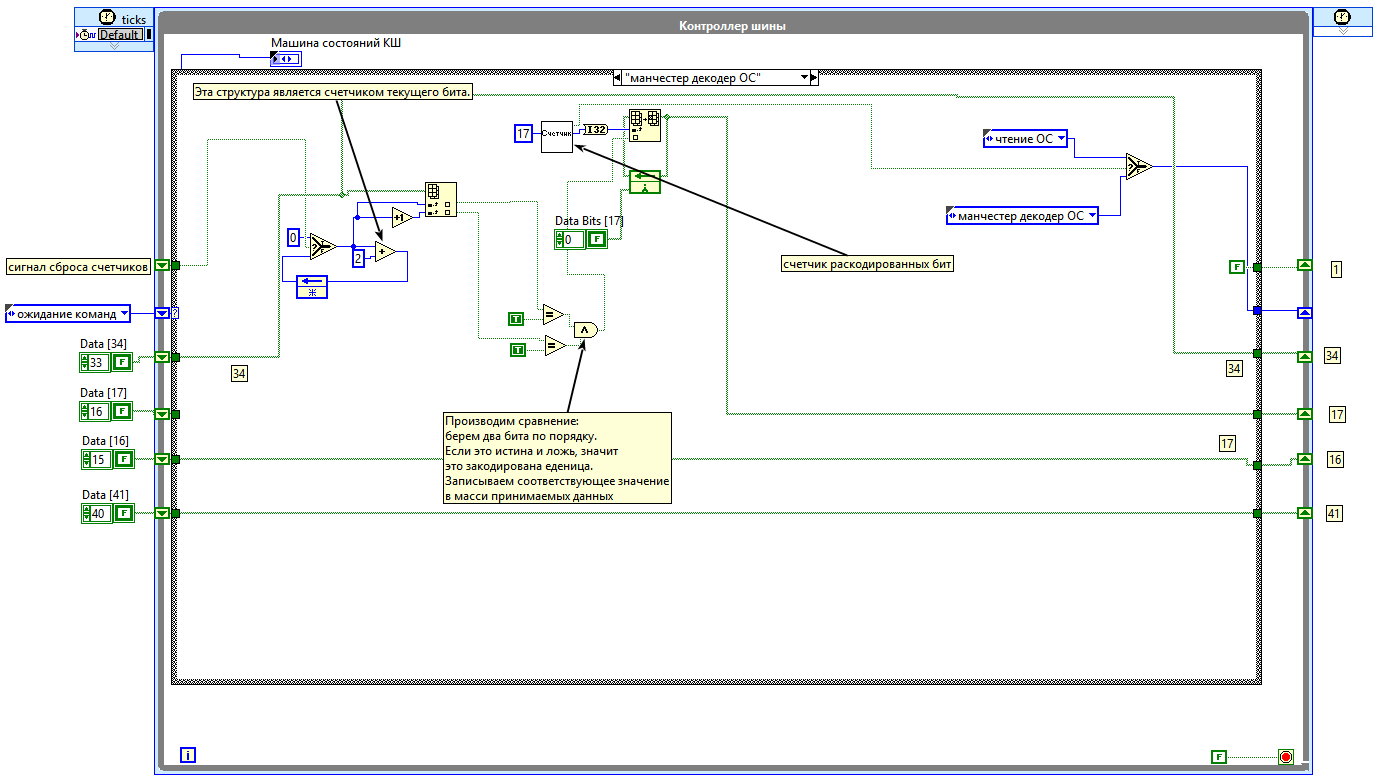
Программа использует два счетчика: счетчик длины полубита (20 тиков) и счетчик таймаута (80 тиков). Каждые 20 тиков считывается состояние линии и записывается в массив принимаемых данных.



Когда счетчик принятых данных достигнет 34, переходим в соостояние «манчестер декодер ОС».

1.18 Состояние «Манчестер декодер ОС»

В начале состояния находится счетчик текущего бита. Он используется для считывания из принятого массива бит попарно. Далее происходит сравнение пары полубит с константами. Таким образом преобразуем манчестерский код в биты сообщения.

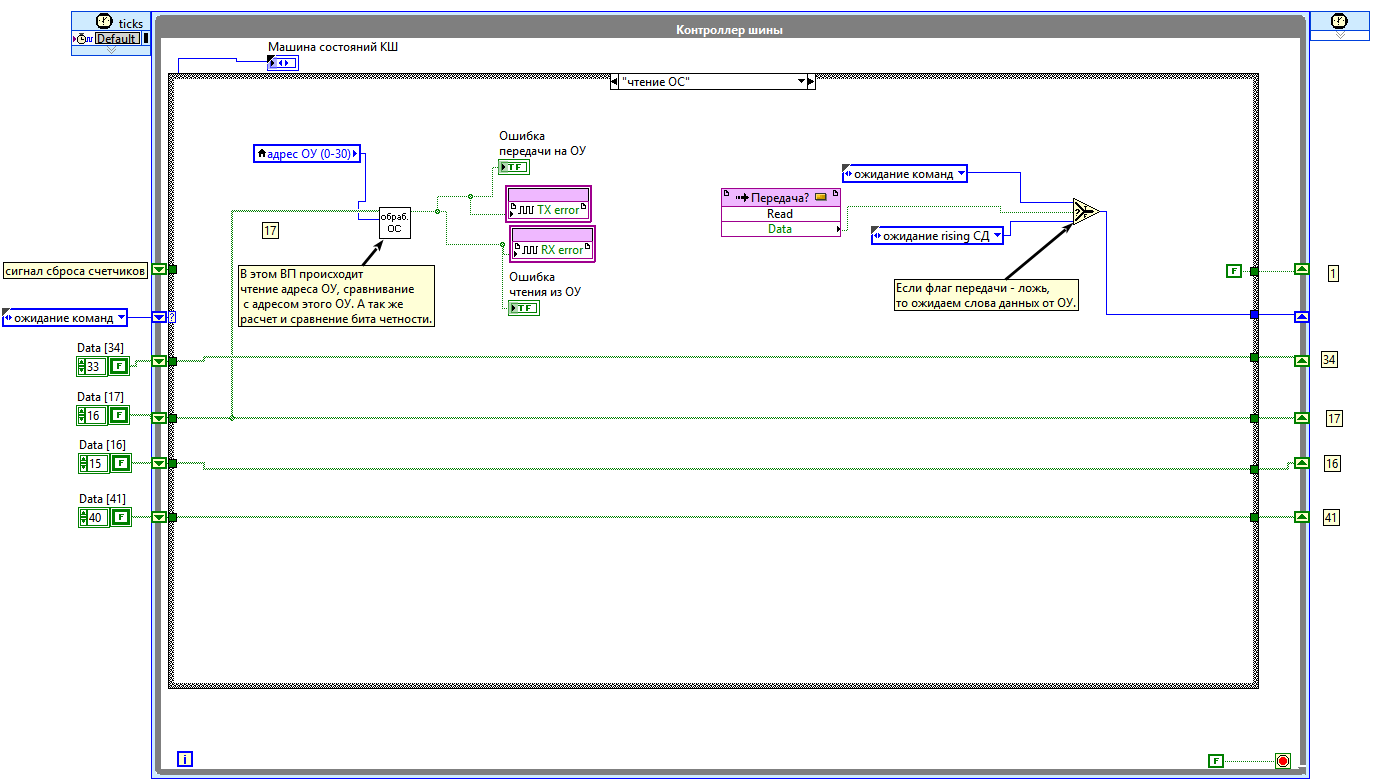


После декодирования переходим в состояние чтения ОС.

1.19 Состояние «Чтение ОС»

В начале находится ВП «обработка ОС», в котором происходит чтение адреса ОУ, сравнивание

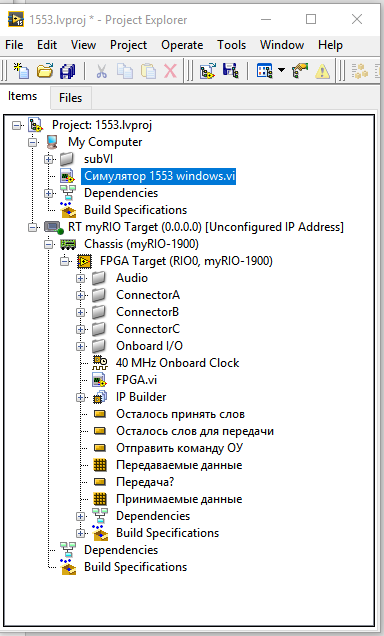
с адресом этого ОУ. А так же расчет и сравнение бита четности.



В конце алгоритма происходит проверка флага передачи. Если ложь – значит переходим в состояние ожидания слов данных от ОУ.

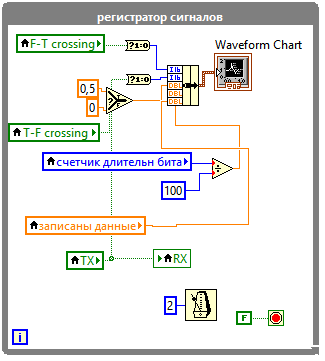
2. Программа для отладки и проверки.

Проверим программу для ПЛИС, выполнив ее на компьютере с Windows. Для этого переместим программу в дереве проекта в ветку My computer. Что бы оценить работу протокола визуально, снизим скорость передачи до 20 бит в секунду.

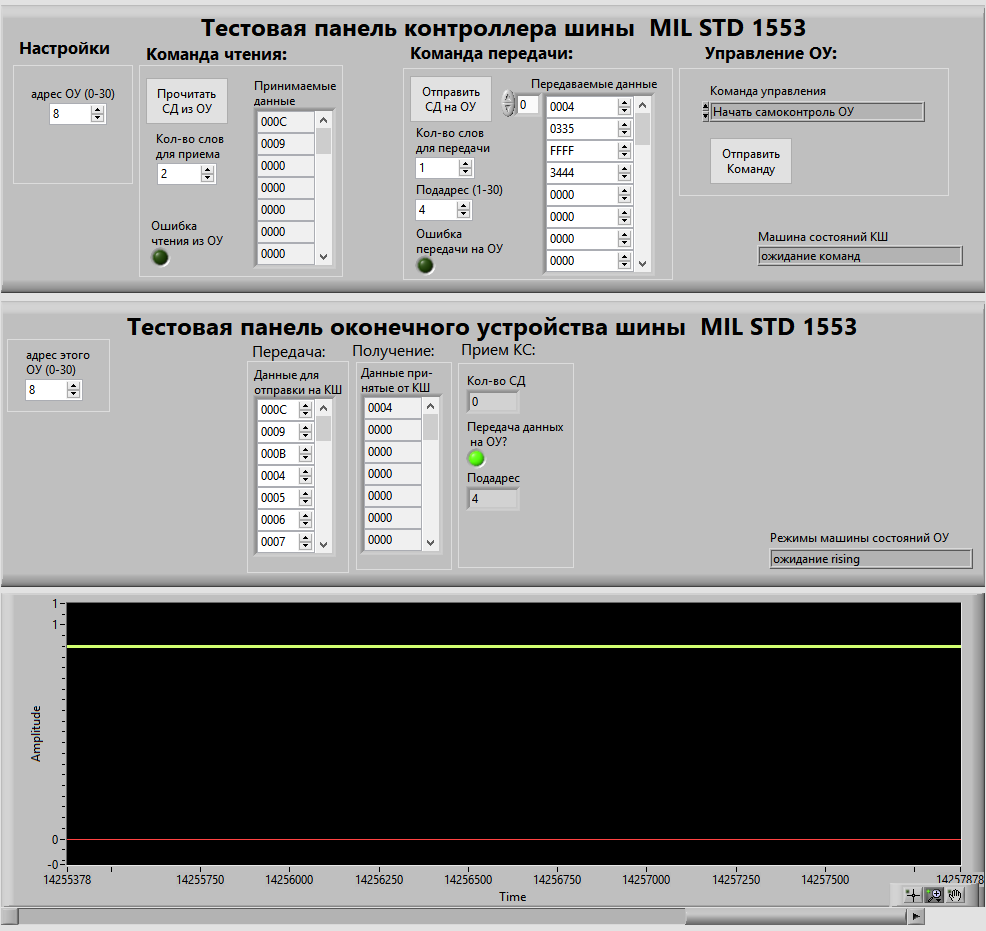


Для проверки нам понадобится разработать программу ОУ. Используем выше описанную машину состояний, для ОУ.

Соединим линию приема и передачи программно. Добавим график для чтения осциллограмм.



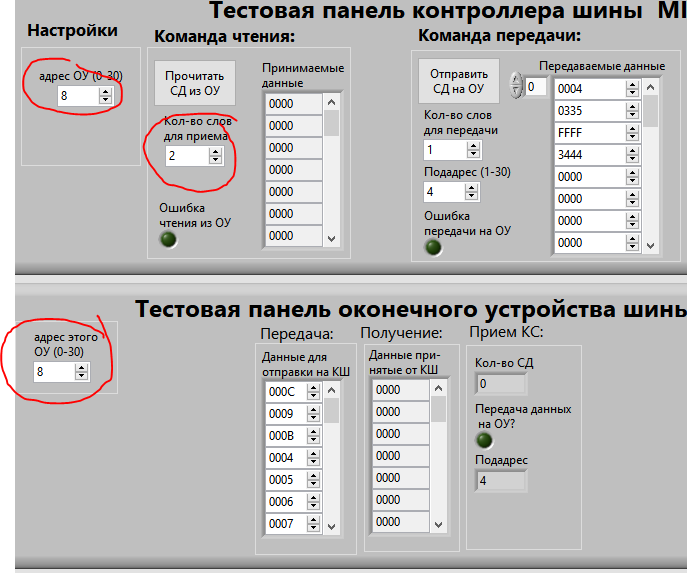
В результате была получена следующая программа:



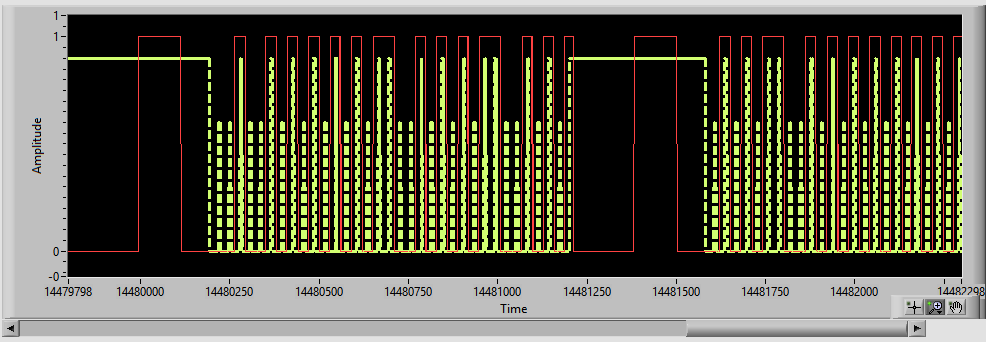
2.1 Проверка режима чтения СД из ОУ.

Установим на ОУ (1) и КШ (2) один и тот же адрес ОУ. Установим нужное количество слов

Данных для приема.

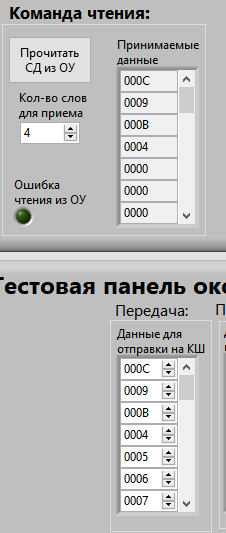


Включаем Чтение из ОУ нажатием кнопки «Прочитать СД из ОУ». Программа сгенерирует КС. Красной линией обозначено состоянии линии приема\передачи.



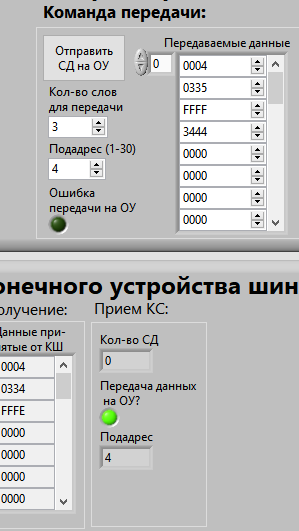
Желтая линия – распознавание полубит в состояниях чтения бит.

В результате мы увидим как в окне «принимаемые данные» появятся данные из ОУ.



2.2 Проверка режима записи СД в ОУ.

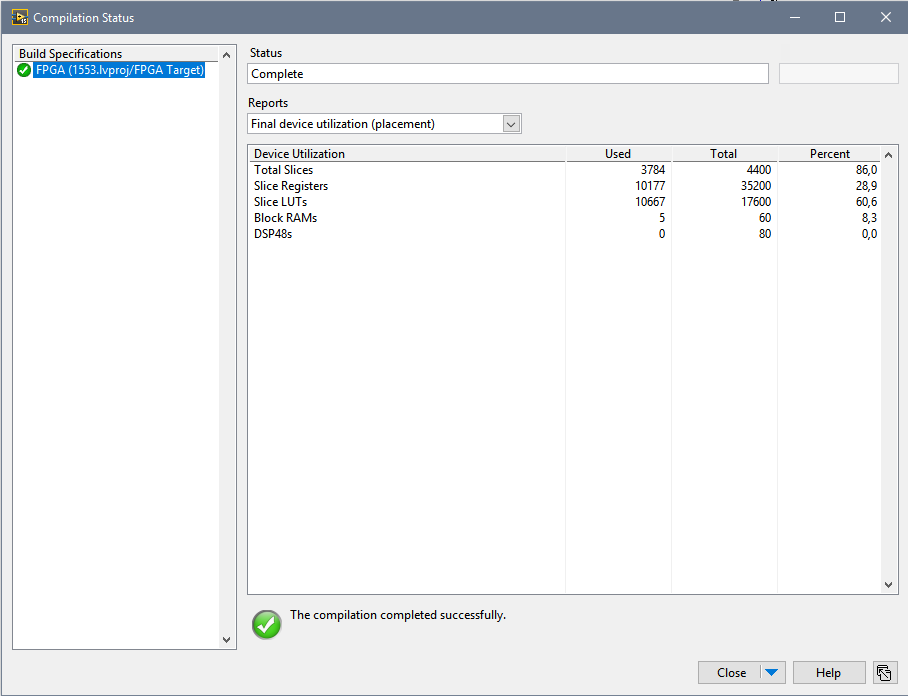
Проделаем то же самое с режимом записи:



Проверку разработанной для ПЛИС программы можно считать успешной.

3. Компилирование программы под ПЛИС.

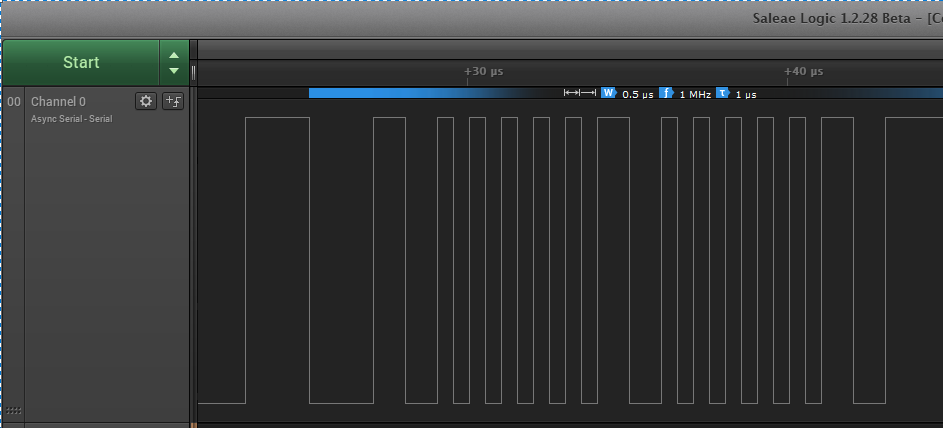
LabVIEW самостоятельно передает код в компилятор Xilinx и генерирует битфайл для загрузки на ПЛИС. Рассмотрим результат работы компилятора:



Программа скомпилирована без ошибок. Использовано 86% ресурсов ПЛИС.

Подключим к выходу DIO 0 MXP A myRIO осциллограф и убедимся в

в работоспособности устройства, а так же в правильно выбранной скорости.



На осциллографе видно, что частота передачи данных выбрана правильно: 1 МГц. Форма сигналов соответствует стандарту. Отчетливо виден синхросигнал КС и биты данных.