

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Вычислительной техники

КУРСОВАЯ РАБОТА
по дисциплине «Элементная база цифровых систем»

Тема: Преобразователь параллельного кода в последовательный для
асинхронного протокола передач
Вариант 7.2

Студент гр. 9308

Преподаватель

Соболев М.С.
Ельчанинов М.Н.

Санкт-Петербург,

2022

Оглавление

Введение.....	4
1. Задание на проектирование узла.....	5
2. Описание предлагаемых вариантов реализации узла.....	6
2.1. Схема на основе параллельно-последовательного регистра.....	7
2.2. Схема на основе параллельного регистра и мультиплексора.....	9
3. Описание основных элементов библиотеки САПР Quartus II и стандартных микросхем, необходимых для реализации вариантов узла.....	10
4. Описание процесса синтеза и моделирования работы предложенных вариантов средствами САПР Quartus II при графическом вводе проектов и полученных при этом результатов.....	15
4.1. Синтез и моделирование узла на основе параллельно-последовательного регистра.....	15
4.1.1. Схема узла на основе параллельно-последовательного регистра.....	15
4.1.2. Компиляция схемы узла на основе параллельно-последовательного регистра	18
4.1.3. Временное моделирование работы узла на основе параллельно-последовательного регистра.....	19
4.2. Синтез и моделирование узла на основе мультиплексора.....	27
4.2.1. Схема узла на основе мультиплексора.....	27
4.2.2. Компиляция схемы узла на основе мультиплексора.....	30
4.2.3. Временное моделирование работы узла на основе мультиплексора.....	31
4.3. Выбор оптимальной схемы.....	39
5. Определение предельной допустимой частоты генератора тактовых импульсов (ГТИ).....	40
6. Разработка интерфейса сопряжения схемы узла с процессорной системой, для которой проектируемый узел является внешним устройством.....	42
7. Подробное описание функционирования узла с использованием необходимых временных диаграмм.....	44
8. Полная принципиальная электрическая схема типового элемента замены.....	47
9. Краткое заключение по проделанной работе.....	49

10. Список использованной литературы.....	50
---	----

Введение

Разработка специализированного цифрового узла (устройства), выполняемая в рамках курсового проекта, имеет целью:

1. Научить ориентироваться в разнообразии возможностей цифровой техники при проектировании специализированных устройств и эффективно применять современную элементную базу;
2. Показать многовариантность решений при синтезе цифровых узлов и устройств, научить выбирать вариант их реализации, отвечающий требованиям ТЗ и наилучший по заданным критериям качества;
3. Научить пользоваться возможностями современных САПР при проектировании устройств на базе микросхем программируемой логики;
4. Научить технически грамотно описывать функционирование разрабатываемых узлов и устройств и правильно оформлять техническую документацию.

1. Задание на проектирование узла

Разработать узел, преобразующий параллельный код в последовательный с добавлением к нему старт-бита, стоп-бита и бита паритета.

Таблица 1. Вариант задания

Вариант	Разрядность входного кода, бит	Скорость передачи, Кбит/с	Область АП
7.2	16	4,8; 9,6; 19,2	80h

Предусмотреть возможность передачи последовательного кода со скоростями, указанными в вариантах задания. Входной параллельный код, код выбора скорости передачи и сигнал пуска передаются из управляющего устройства (процессора). Критерий выбора варианта реализации схемы — минимум аппаратных затрат.

2. Описание предлагаемых вариантов реализации узла

На вход узла подаётся сигнал «load», происходит параллельная загрузка кода. Далее на вход узла подаются сигналы «start» и «speed» для начала работы узла и для выборы режима скорости соответственно, затем начинает выполняться пересылка последовательного кода, полученного из параллельного. Сигнал «stop» запускает остановку передачи. Сигнал «reset» приводит к принудительному сбросу устройства в начальное состояние. Сигнал «с» приходит от генератора тактовых импульсов. Выбранная скорость загрузки находится на шине данных.

Выходной сигнал «serial_code» представляет собой выходной последовательный код. Выходной сигнал «ready» устанавливается в высокий логический уровень после того, как разрабатываемый узел передаст все входящие параллельные разряды в последовательном коде из выхода регистра.

Всего использовано 4 адреса:

80h – «load», адрес регистра загрузки кода;

81h – «speed», адрес регистра загрузки скорости;

82h – отдельный адрес, пропускающий сигналы «start» и «ready»;

83h – отдельный адрес, пропускающий сигналы «stop» и «reset».

Таблица 2. Служебные биты

x[0]	x[1]	Режим работы узла
0	0	4,8 Кбит/с
0	1	9,6 Кбит/с
1	0	19,2 Кбит/с

2.1. Схема на основе параллельно-последовательного регистра

При поступлении сигнала «load» соответствующие биты данных загружаются и сохраняются в сдвиговом регистре. После поступления всех битов данных вычисляется бит паритета, сохраняющийся в том же регистре. Старт-бит и стоп-бит также записываются в указанный регистр.

При поступлении сигнала «start» схема считывает код скорости и начинает передачу данных посредством сдвига с заполнением 1 в регистре на выход схемы с заданной скоростью.

При приостановке выдачи сигнала на сдвигающий регистр перестаёт поступать тактовый сигнал, что останавливает сдвиг с сохранением текущего положения битов в регистре. Сигнал «reset» принудительно переводит все компоненты схемы в начальное состояние.

Сдвиг производится заданное количество тактов. Счётчик тактов также приостанавливает своё действие при подаче сигнала «stop» и сбрасывается в начальное состояние при подаче сигнала «reset».

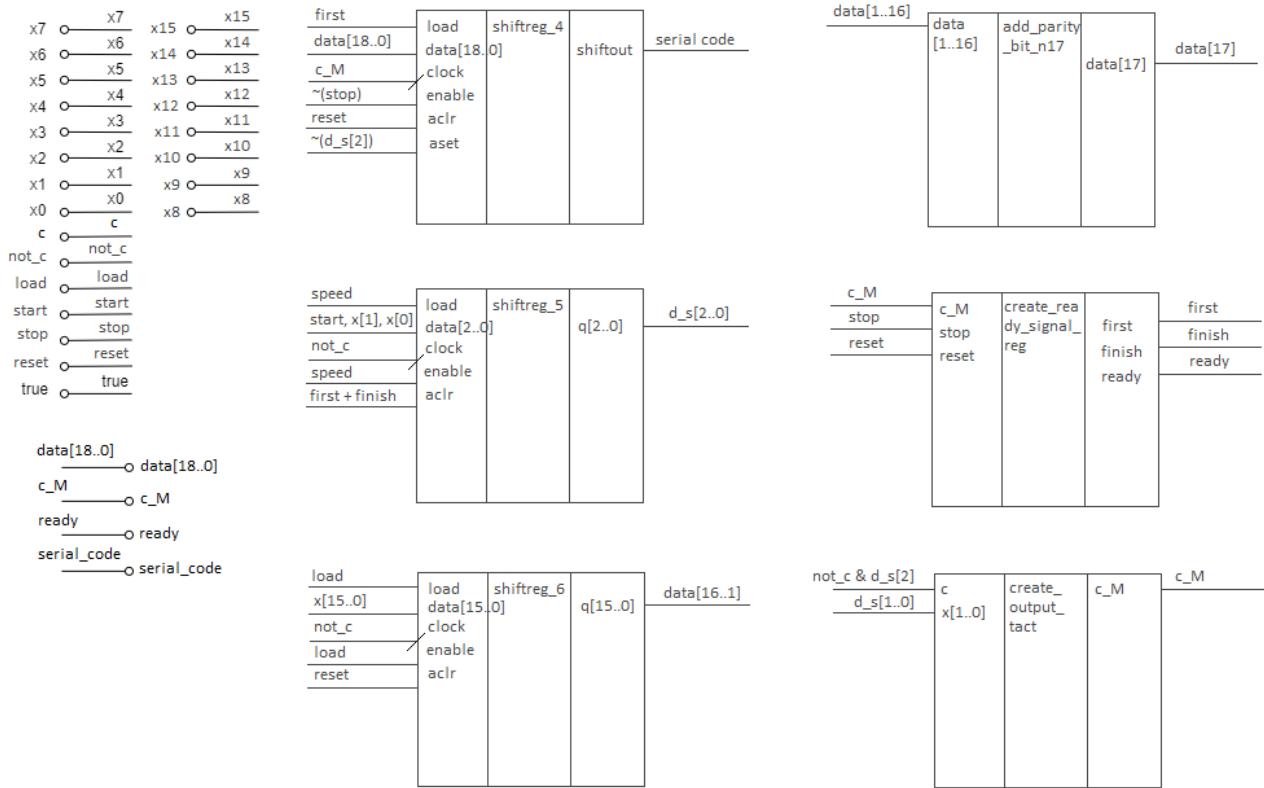


Рисунок 1. Функциональная схема узла на основе параллельно-последовательного регистра

2.2. Схема на основе параллельного регистра и мультиплексора

Отличие данной схемы на основе параллельного регистра и мультиплексора от первой схемы на основе параллельно-последовательного регистра заключается в том, что биты данных сохраняются в параллельном регистре, а передаются на выход через мультиплексор.

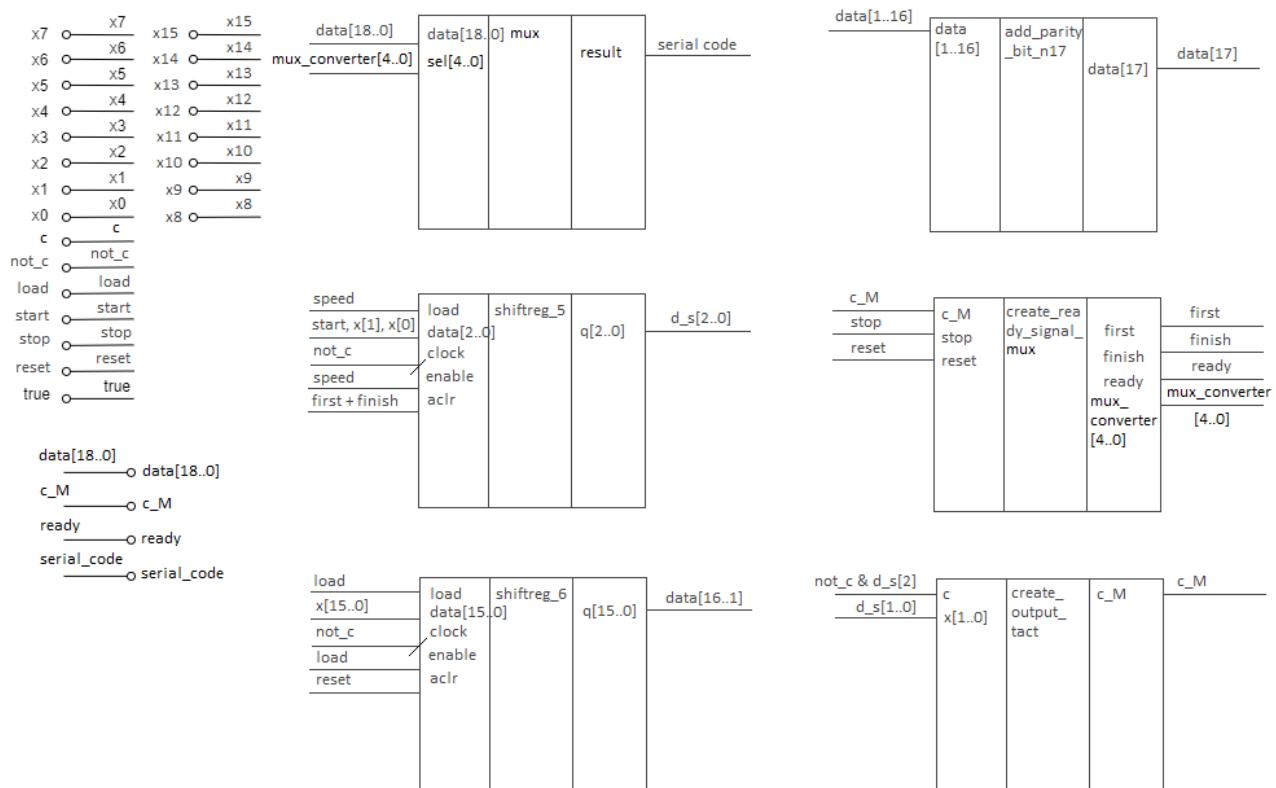


Рисунок 2. Функциональная схема узла на основе параллельного регистра и мультиплексора

3. Описание основных элементов библиотеки САПР Quartus II и стандартных микросхем, необходимых для реализации вариантов узла

Для реализации узла в САПР «QUARTUS II», помимо логических элементов (примитивов) используются основные элементы библиотеки.

TFF-триггер используется для формирования тактовой частоты.

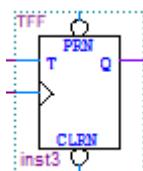


Рисунок 3. TFF-триггер

Мегафункция «lpm_mux_1» используется для выбора скорости выдачи сигнала.

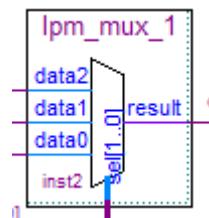


Рисунок 4. Мультиплексор «lpm_mux_1»

Мегафункция «lpm_counter_2» используется для понижения частоты тактового сигнала.

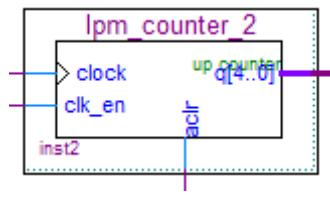


Рисунок 5. Счётчик «lpm_counter_2»

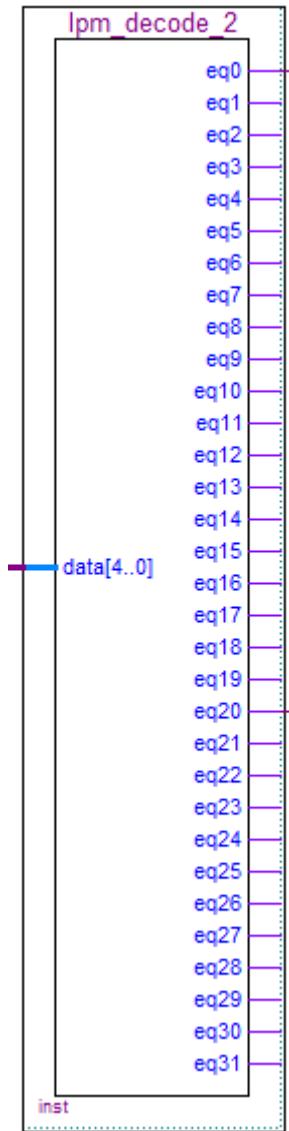


Рисунок 6. Дешифратор «lpm_decode_2»

Мегафункция «lpm_constant_0» используется для формирования константы логического 0.

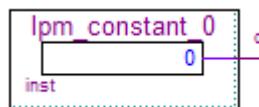


Рисунок 7. Константа «lpm_constant_0»

Мегафункция «lpm_constant_1» используется для формирования константы, логического 1.

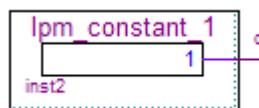


Рисунок 8. Константа «lpm_constant_1»

Мегафункция «lpm_shiftreg_4» используется для сдвигового регистра.

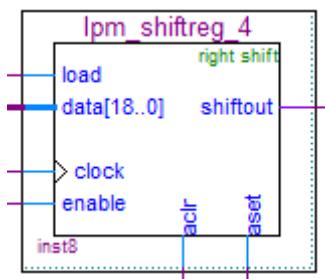


Рисунок 9. Регистр «lpm_shiftreg_4»

Мегафункция «lpm_shiftreg_5» используется для регистра выбора скорости.

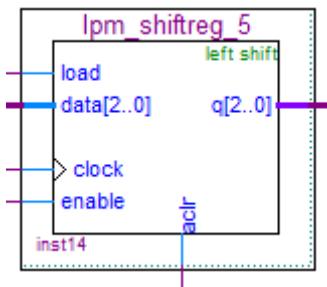


Рисунок 10. Регистр «lpm_shiftreg_5»

Мегафункция «lpm_shiftreg_6» используется для параллельного регистра.

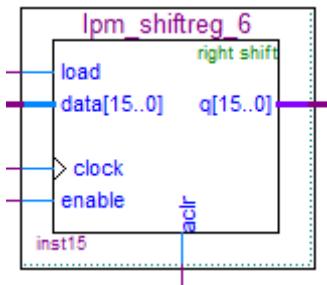


Рисунок 11. Регистр «lpm_shiftreg_6»

Мегафункция «lpm_mux_1» используется для реализации передачи заданного параллельного кода последовательно.

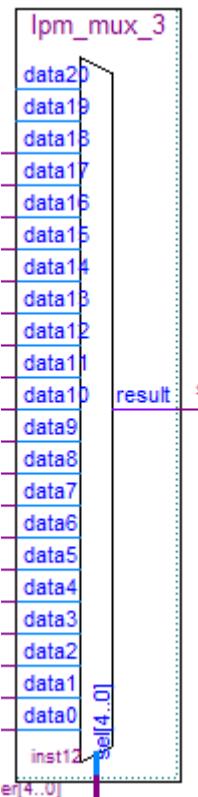


Рисунок 12. Мультиплексор «lpm_mux_3»

4. Описание процесса синтеза и моделирования работы предложенных вариантов средствами САПР Quartus II при графическом вводе проектов и полученных при этом результатов

4.1. Синтез и моделирование узла на основе параллельно-последовательного регистра

Основываясь на схеме и примитивах, описанных в предыдущем разделе, был синтезирован узел в САПР Quartus II.

Реализация узла на основе параллельно-последовательного регистра в САПР Quartus II представлена далее.

4.1.1. Схема узла на основе параллельно-последовательного регистра

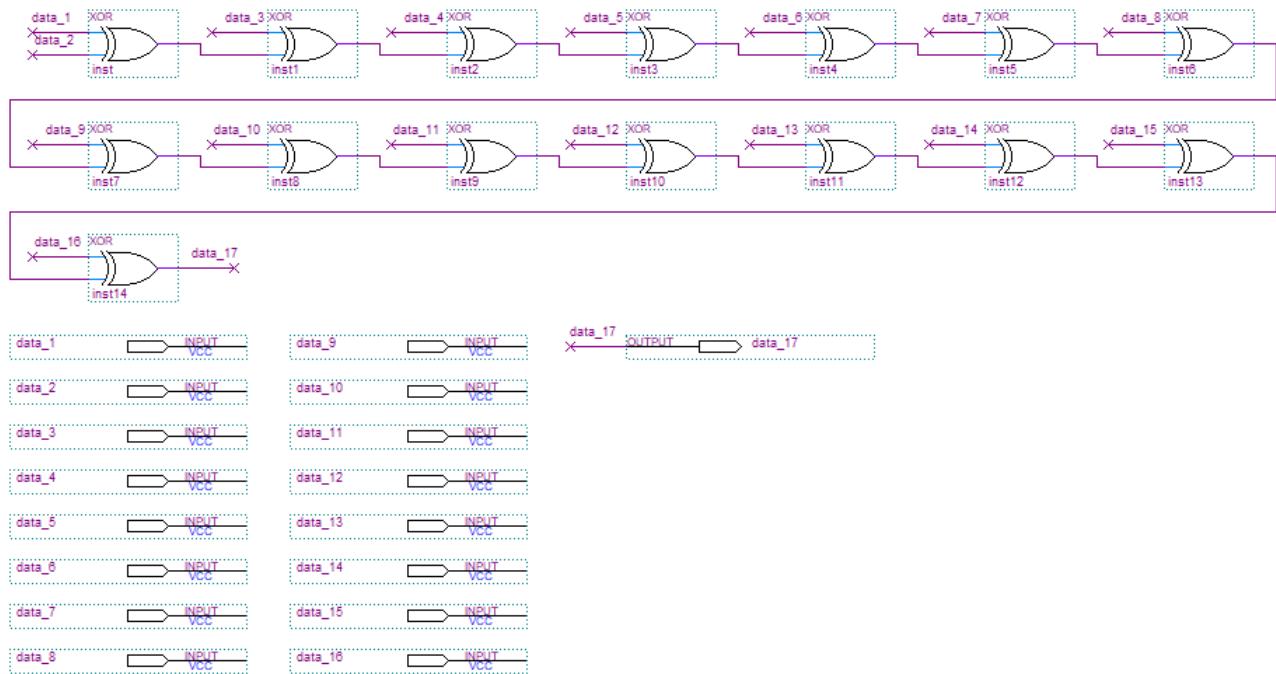


Рисунок 13. Схема «add_parity_bit_n17» узла на основе параллельно-последовательного регистра

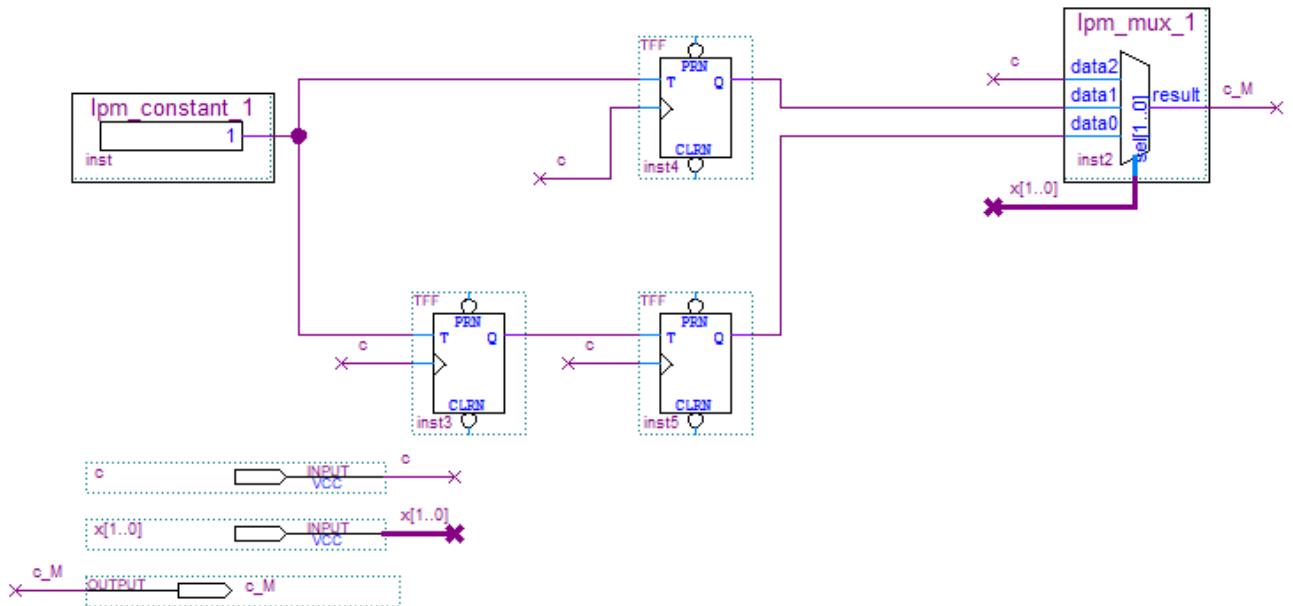


Рисунок 14. Схема «create_output_tact» узла на основе параллельно-последовательного регистра

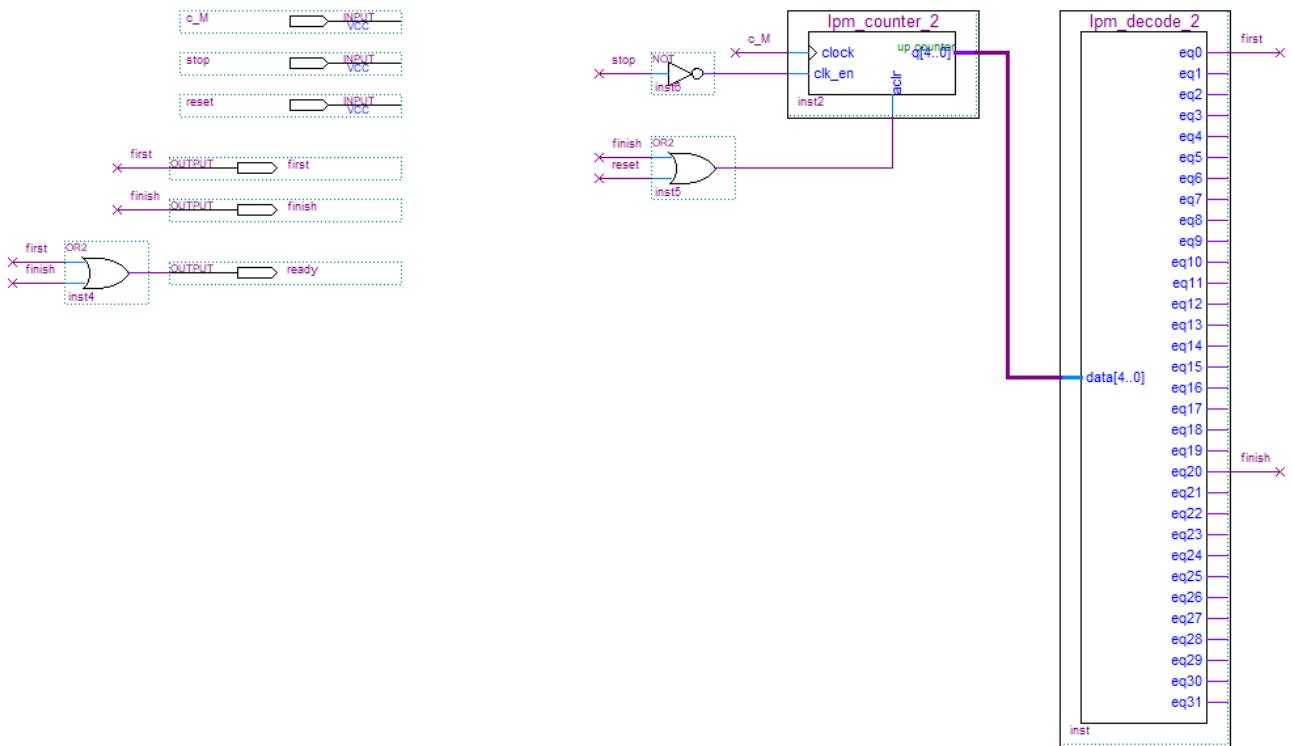


Рисунок 15. Схема «create_ready_signal_reg» узла на основе параллельно-последовательного регистра

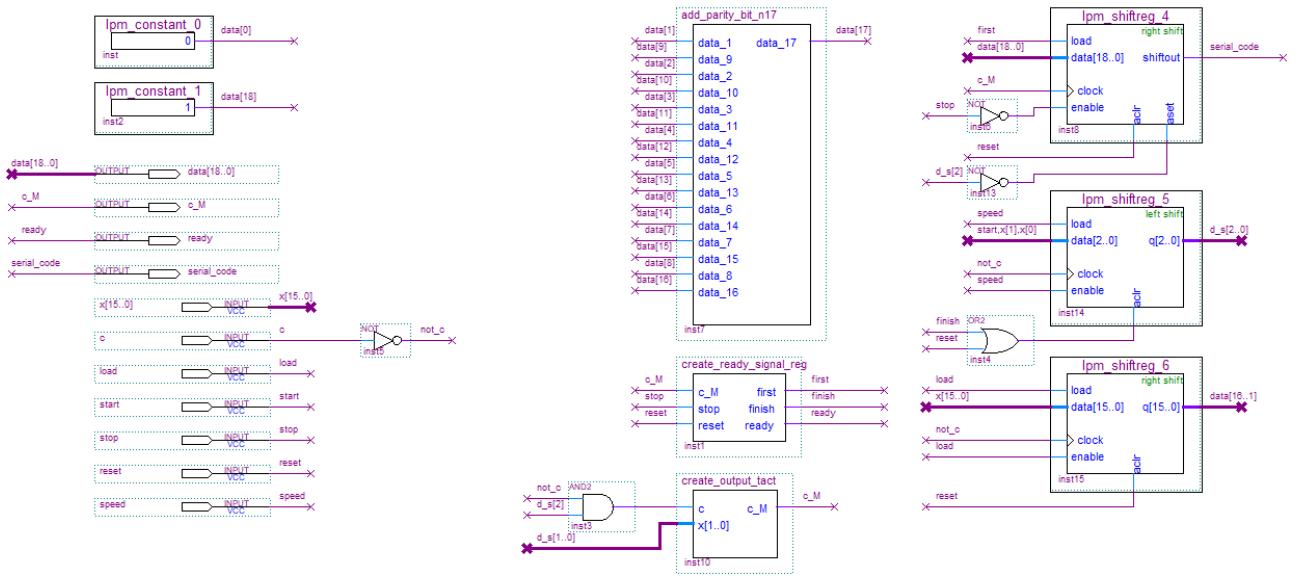


Рисунок 16. Схема «create_serial_from_parallel_reg» узла на основе параллельно-последовательного регистра

4.1.2. Компиляция схемы узла на основе параллельно-последовательного регистра

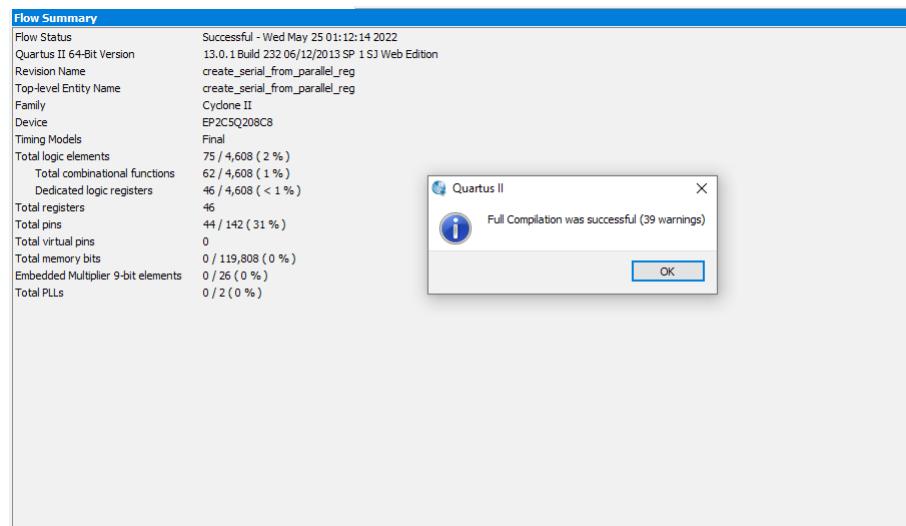


Рисунок 17. Компиляция схемы узла на основе параллельно-последовательного регистра

4.1.3. Временное моделирование работы узла на основе параллельно-последовательного регистра

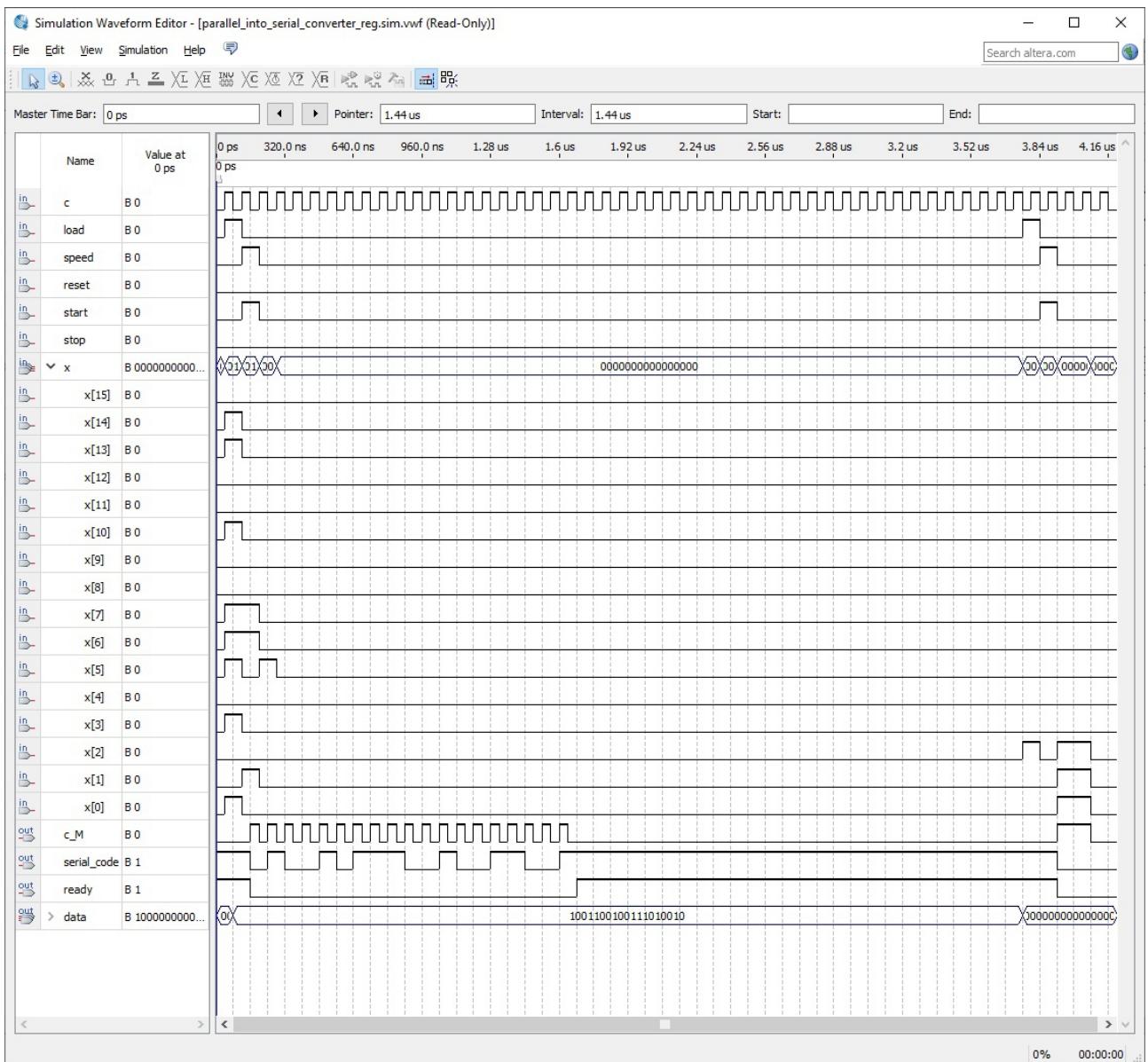


Рисунок 18. Функциональная диаграмма узла на основе параллельно-последовательного регистра, пример 1

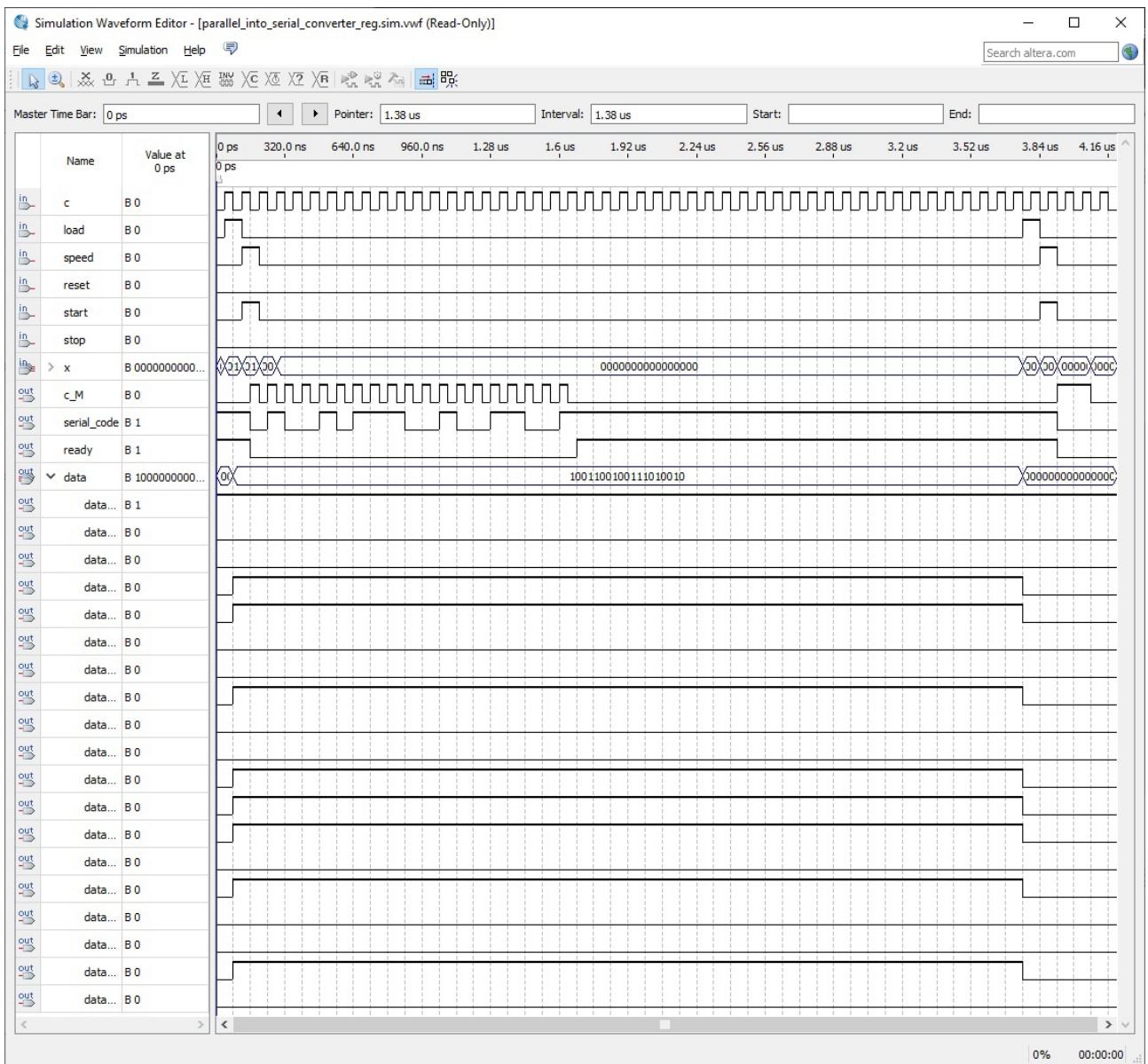


Рисунок 19. Функциональная диаграмма узла на основе параллельно-последовательного регистра, пример 1

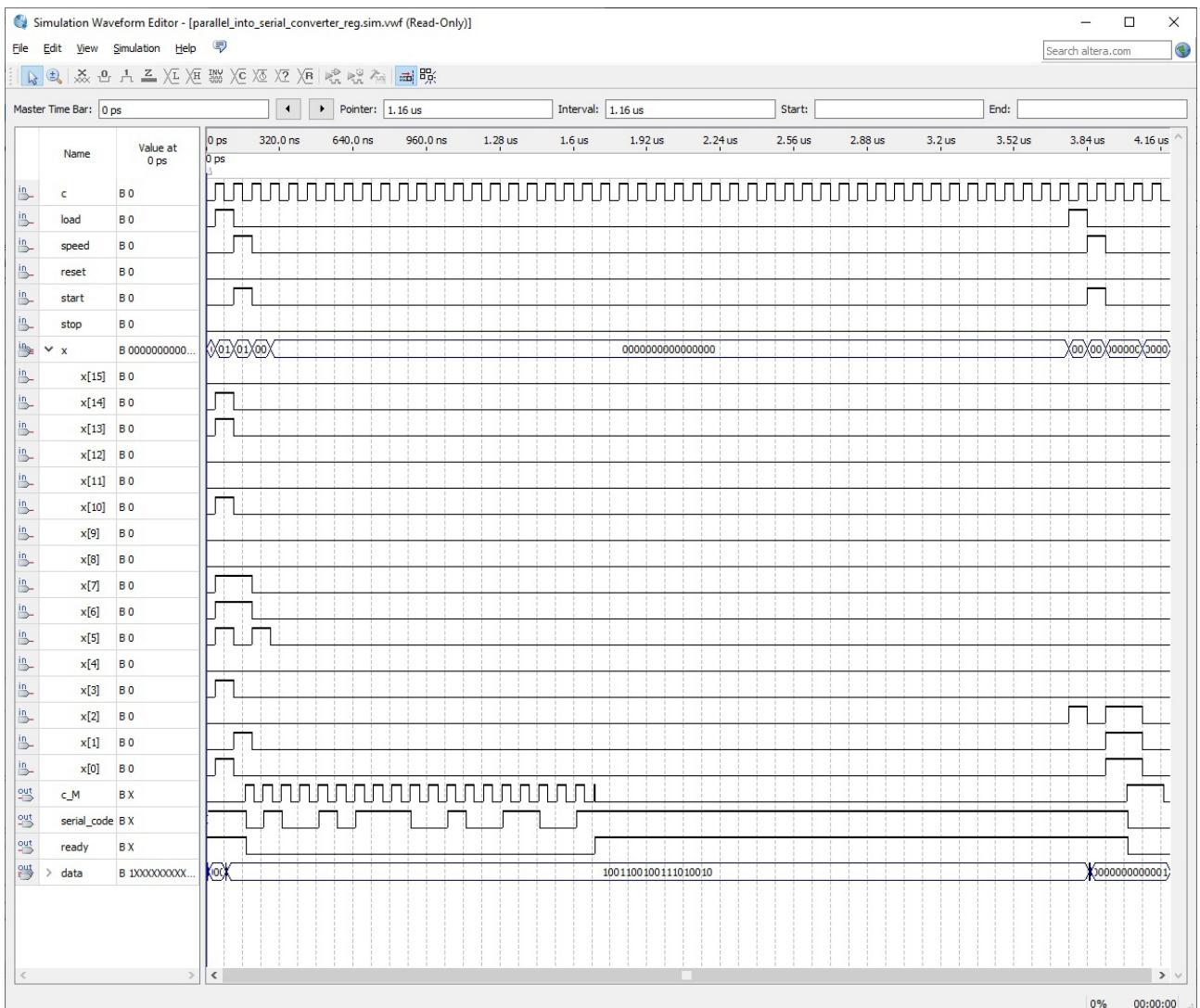


Рисунок 20. Временная диаграмма узла на основе параллельно-последовательного регистра, пример 1

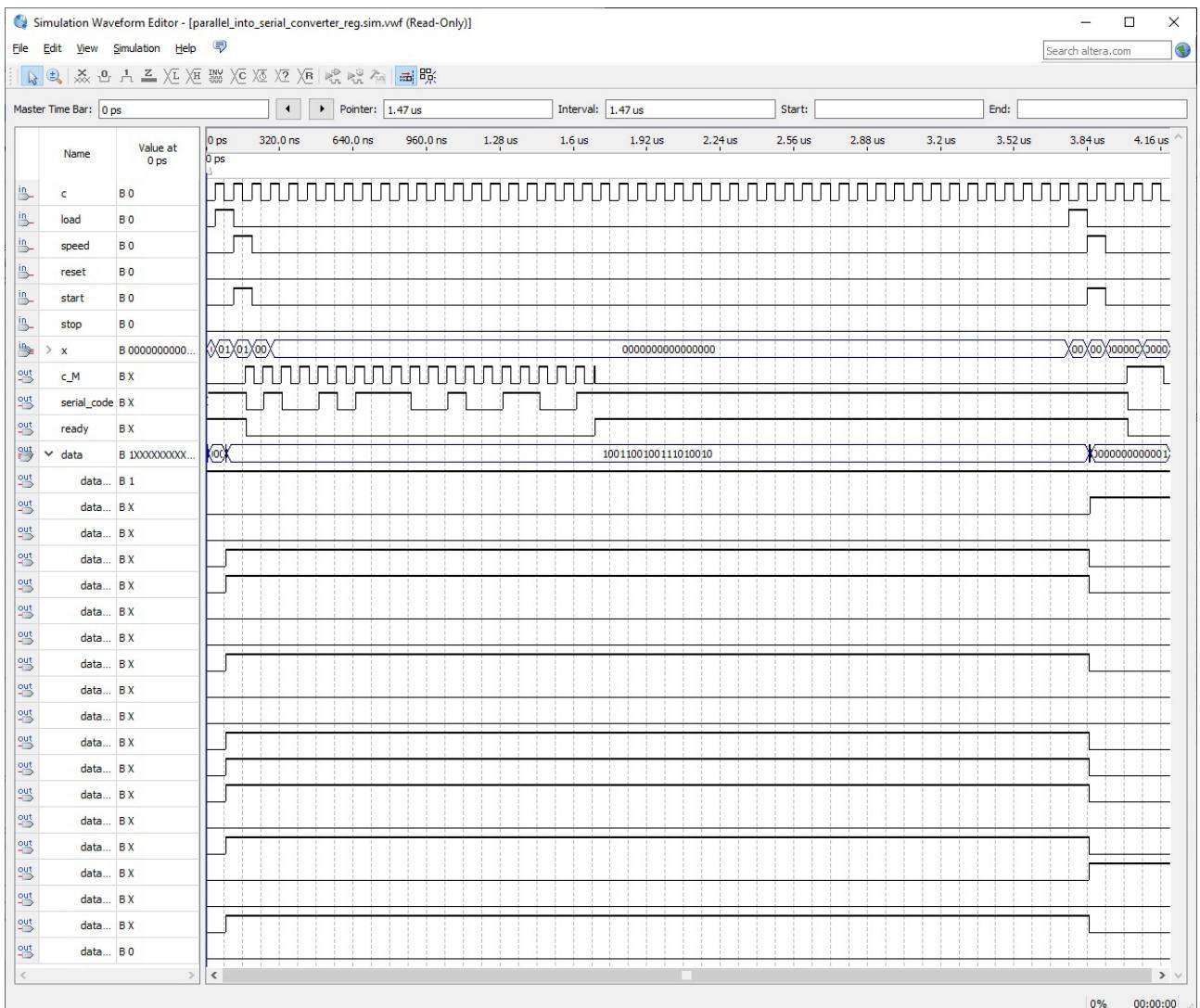


Рисунок 21. Временная диаграмма узла на основе параллельно-последовательного регистра, пример 1

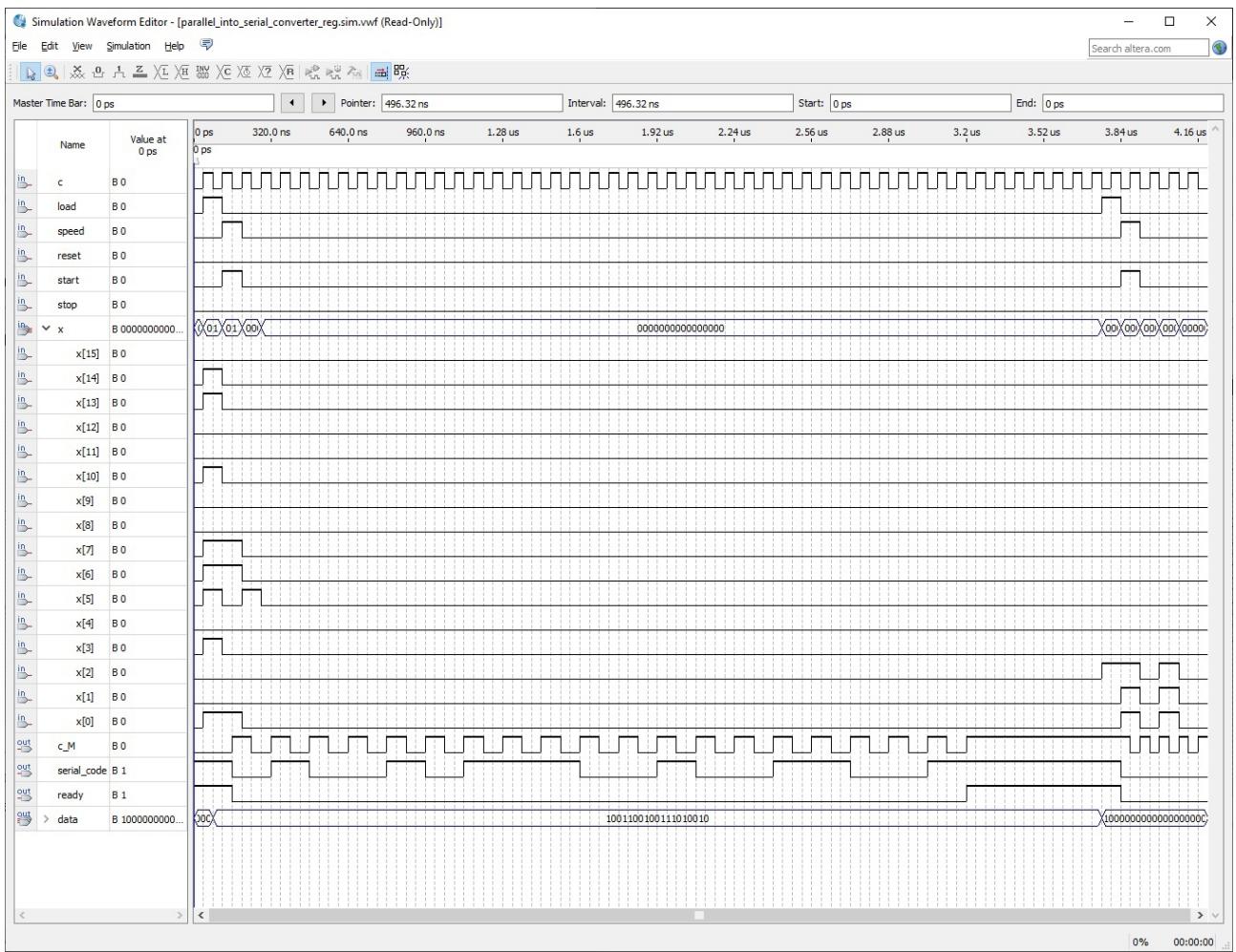


Рисунок 22. Функциональная диаграмма узла на основе параллельно-последовательного регистра, пример 2

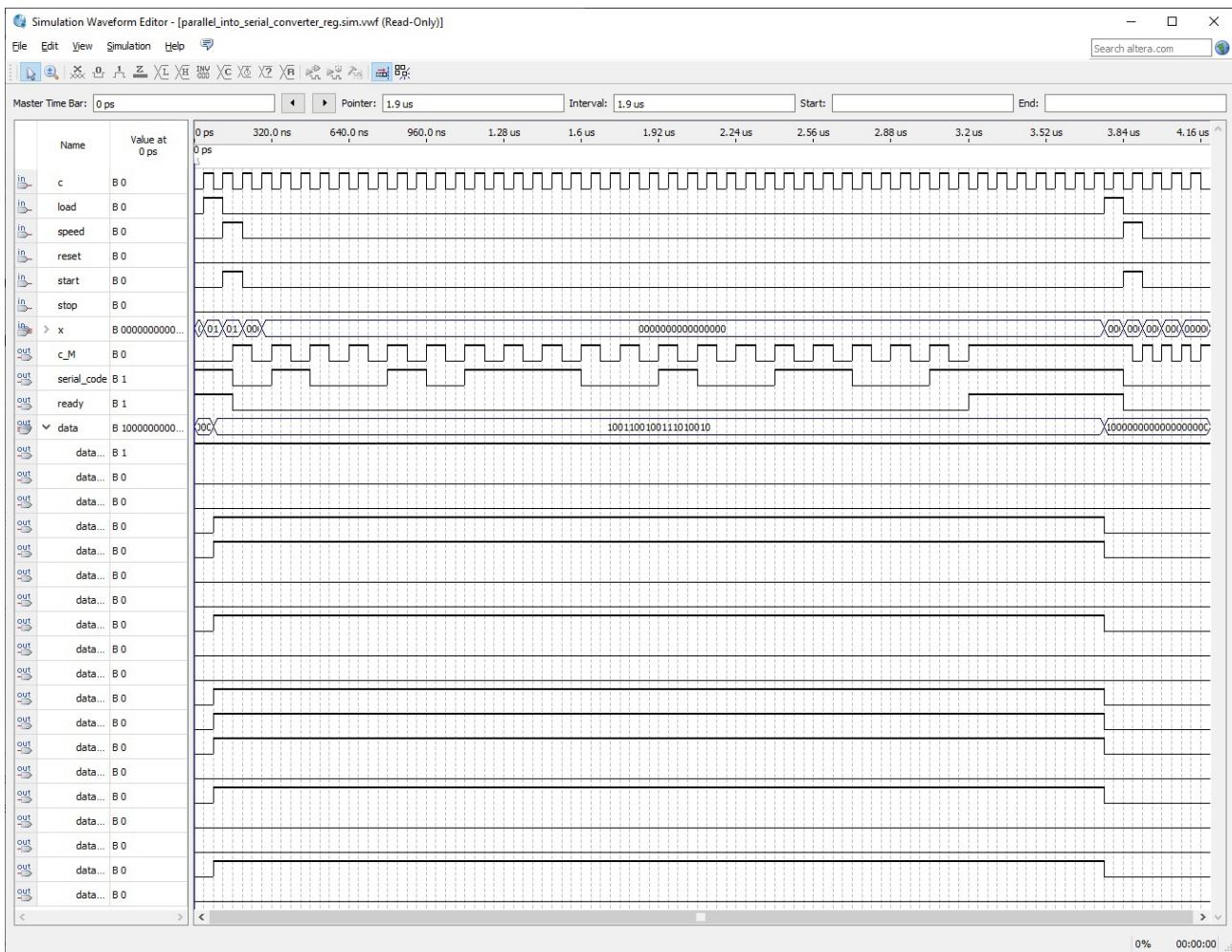


Рисунок 23. Функциональная диаграмма узла на основе параллельно-последовательного регистра, пример 2

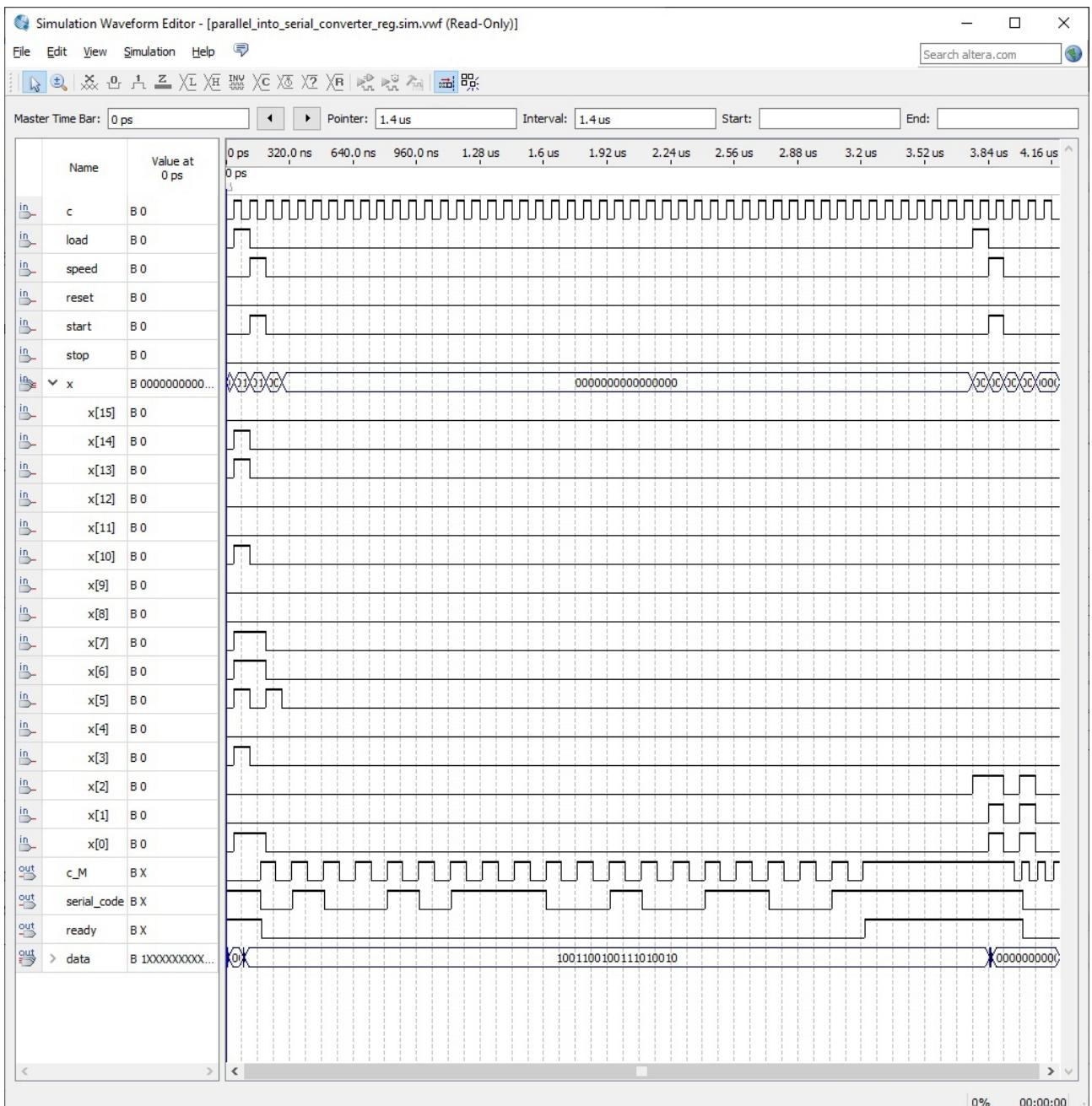


Рисунок 24. Временная диаграмма узла на основе параллельно-последовательного регистра, пример 2

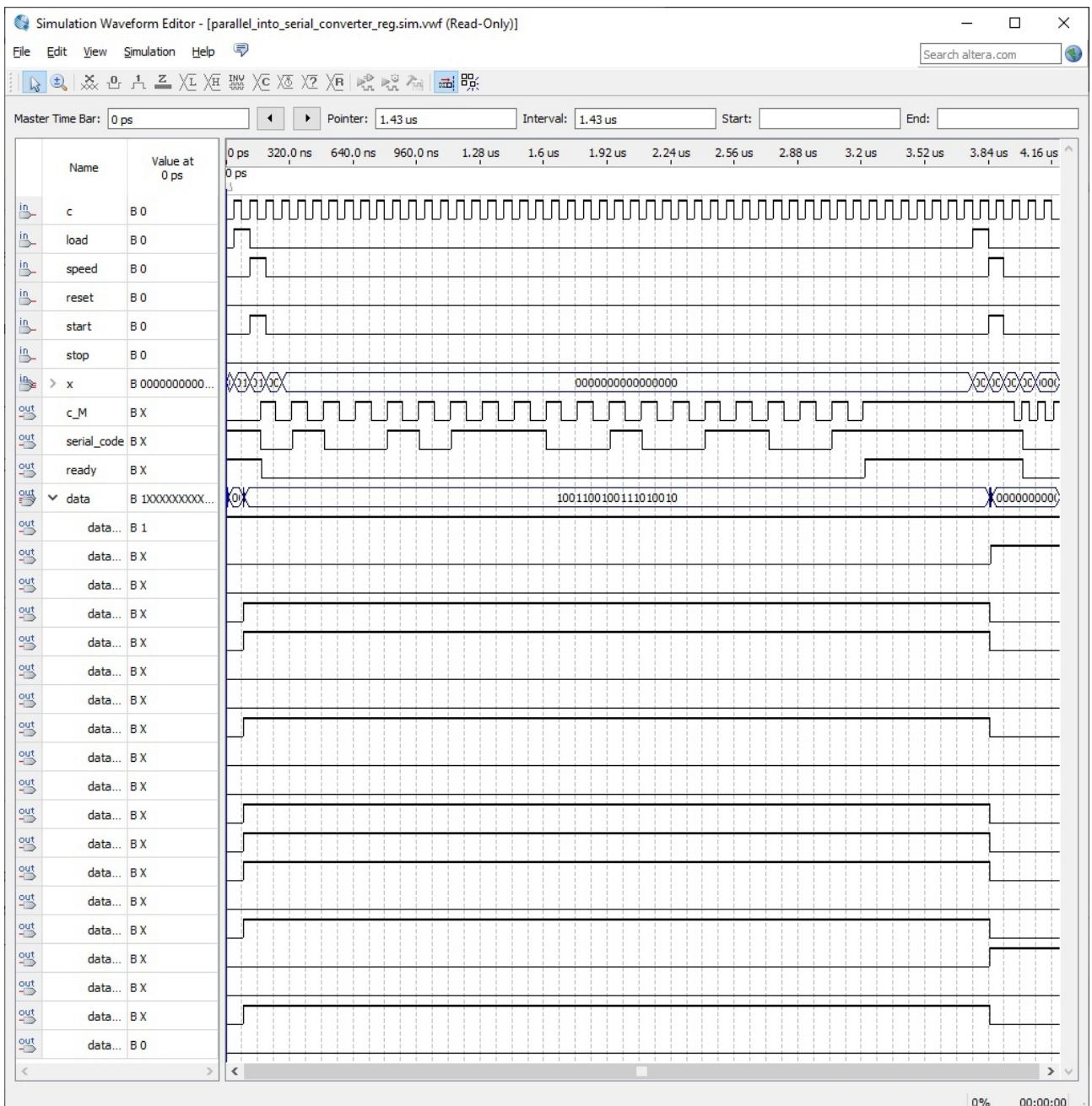


Рисунок 25. Временная диаграмма узла на основе параллельно-последовательного регистра, пример 2

4.2. Синтез и моделирование узла на основе мультиплексора

Основываясь на схеме и примитивах, описанных в предыдущем разделе, был синтезирован узел в САПР Quartus II.

Реализация узла на основе мультиплексора регистра в САПР Quartus II представлена далее.

4.2.1. Схема узла на основе мультиплексора

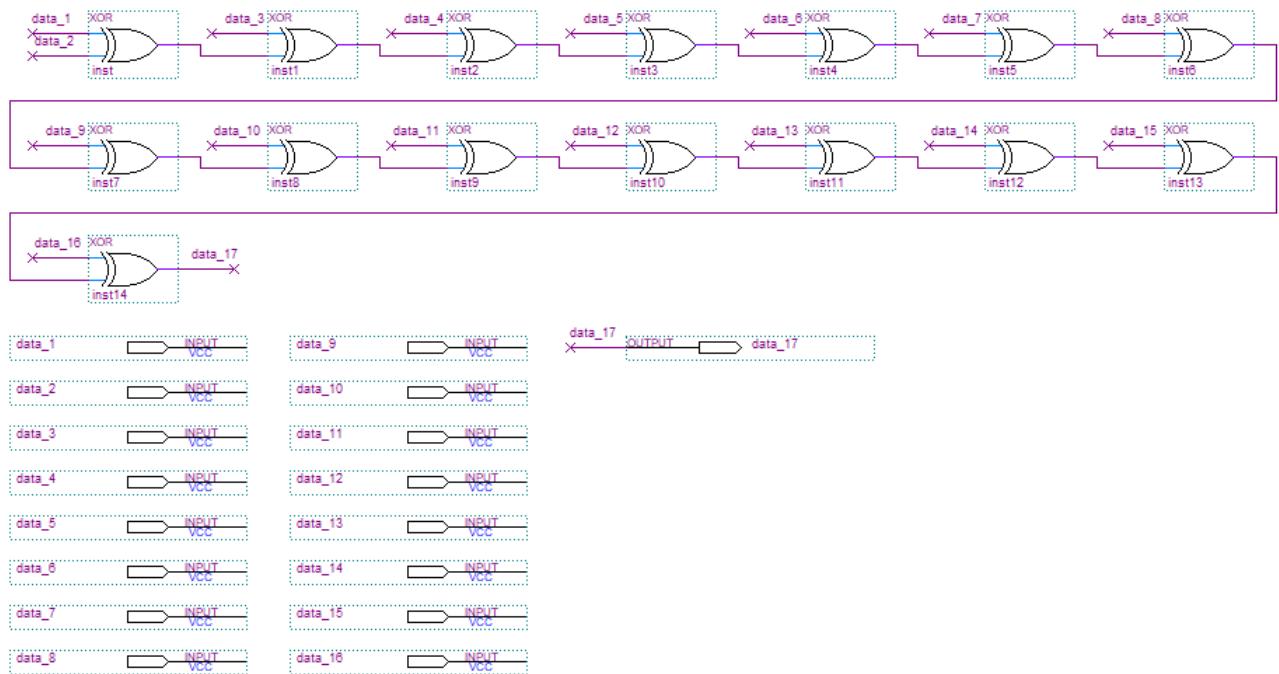


Рисунок 26. Схема «`addParity_bit_n17`» узла на основе мультиплексора

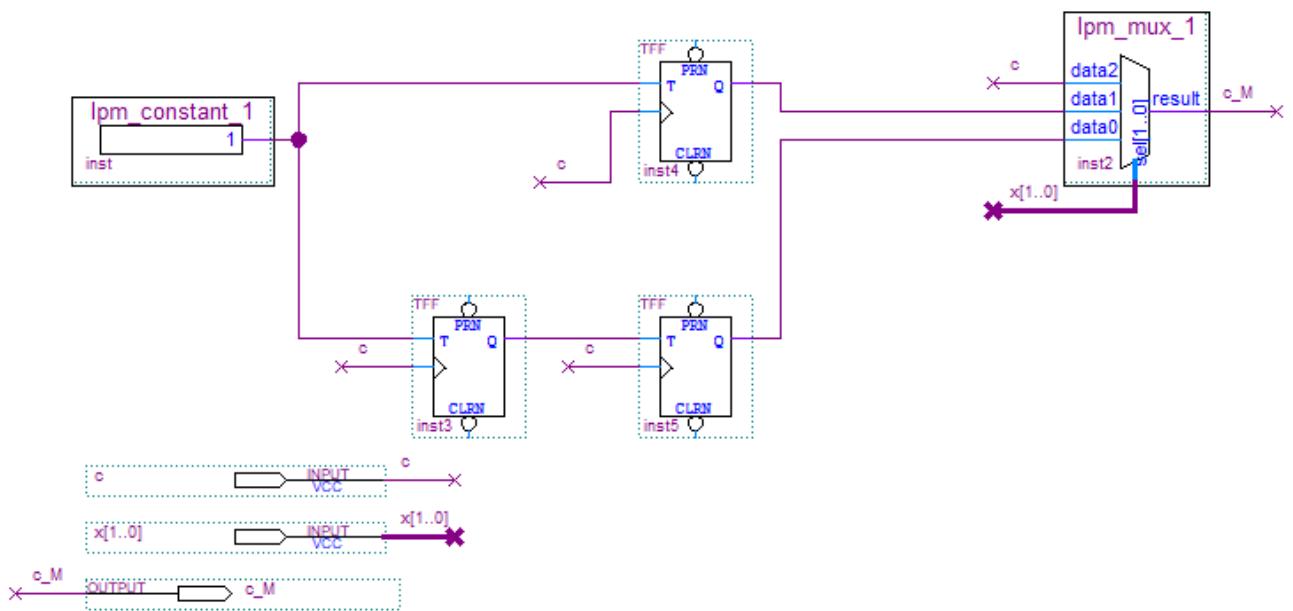


Рисунок 27. Схема «create_output_tact» узла на основе мультиплексора

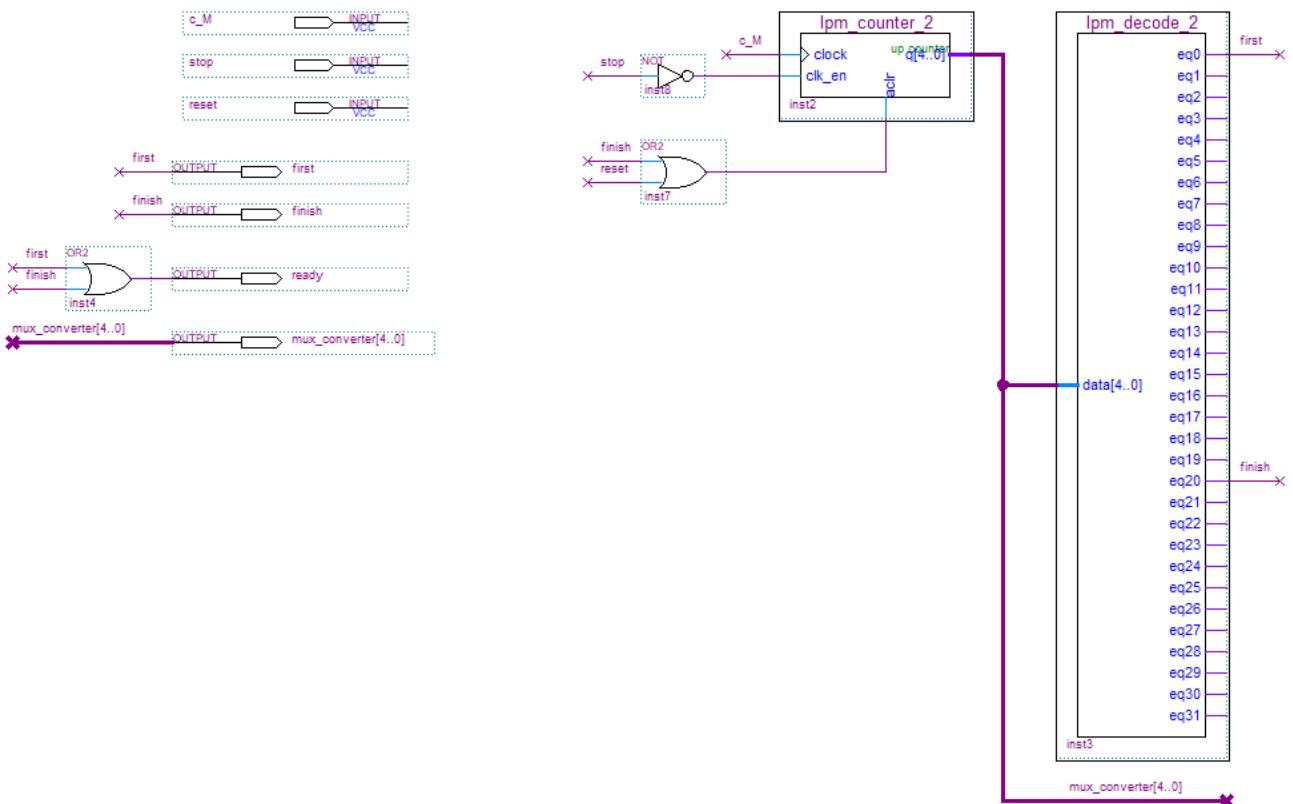


Рисунок 28. Схема «create_ready_signal_mux» узла на основе мультиплексора

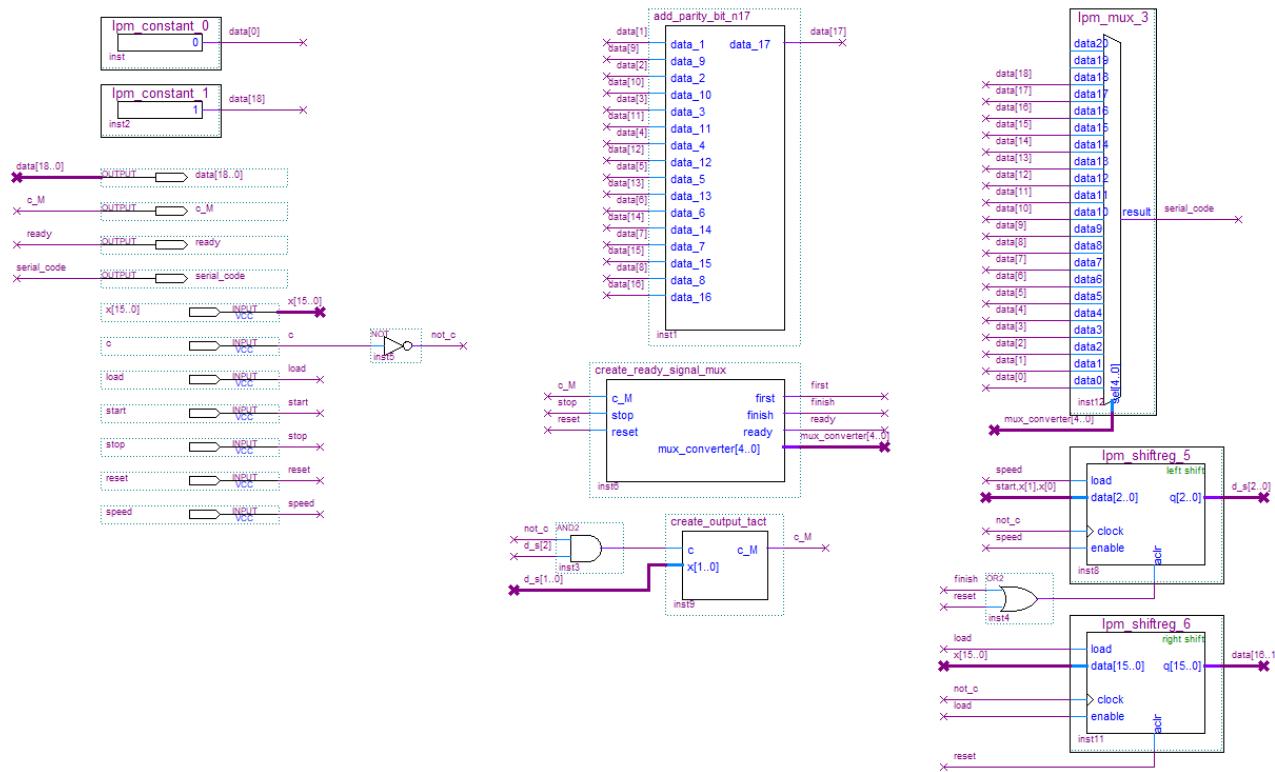


Рисунок 29. Схема «create_serial_from_parallel_mux» узла на основе мультиплексора

4.2.2. Компиляция схемы узла на основе мультиплексора

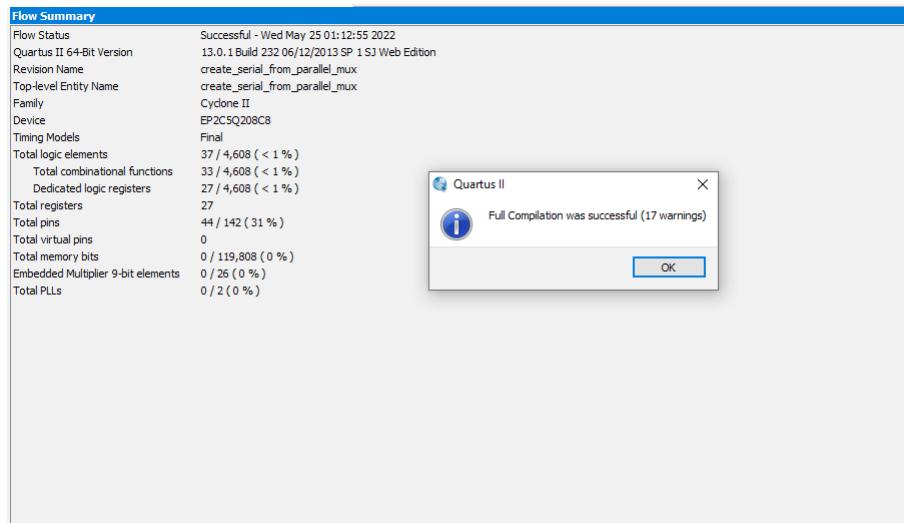


Рисунок 30. Компиляция схемы узла на основе мультиплексора

4.2.3. Временное моделирование работы узла на основе мультиплексора

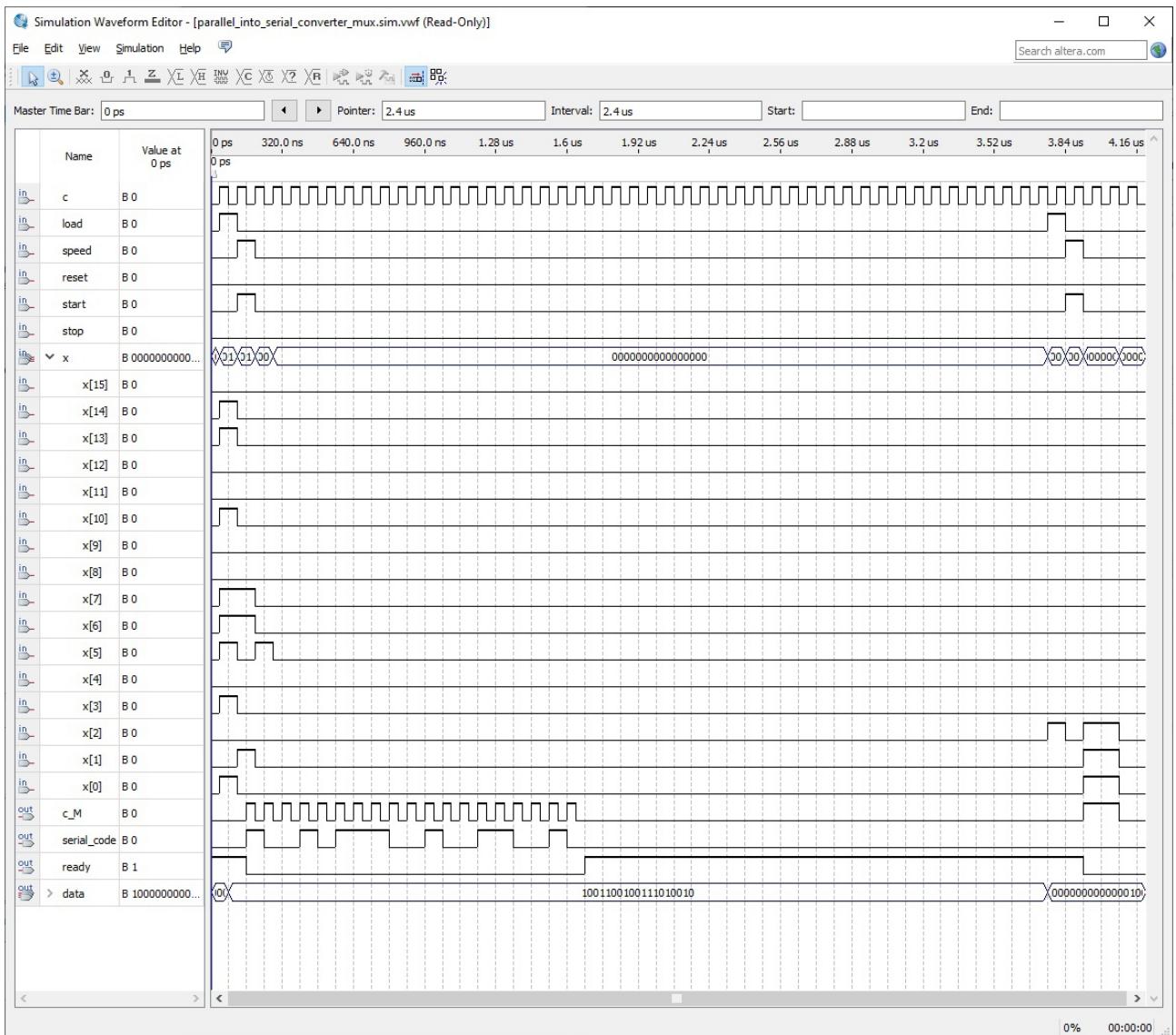


Рисунок 31. Функциональная диаграмма узла на основе мультиплексора,
пример 1

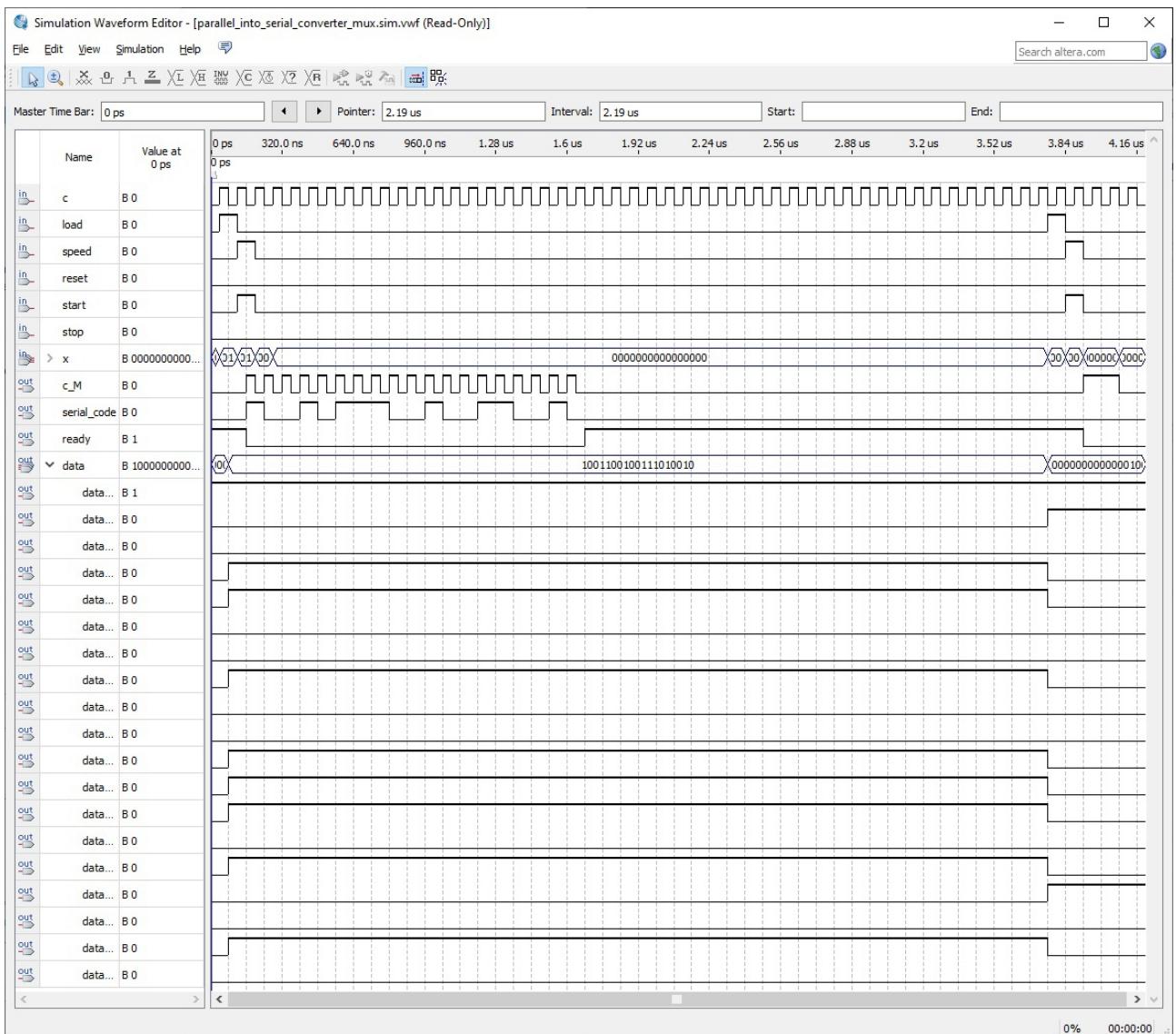


Рисунок 32. Функциональная диаграмма узла на основе мультиплексора,
пример 1

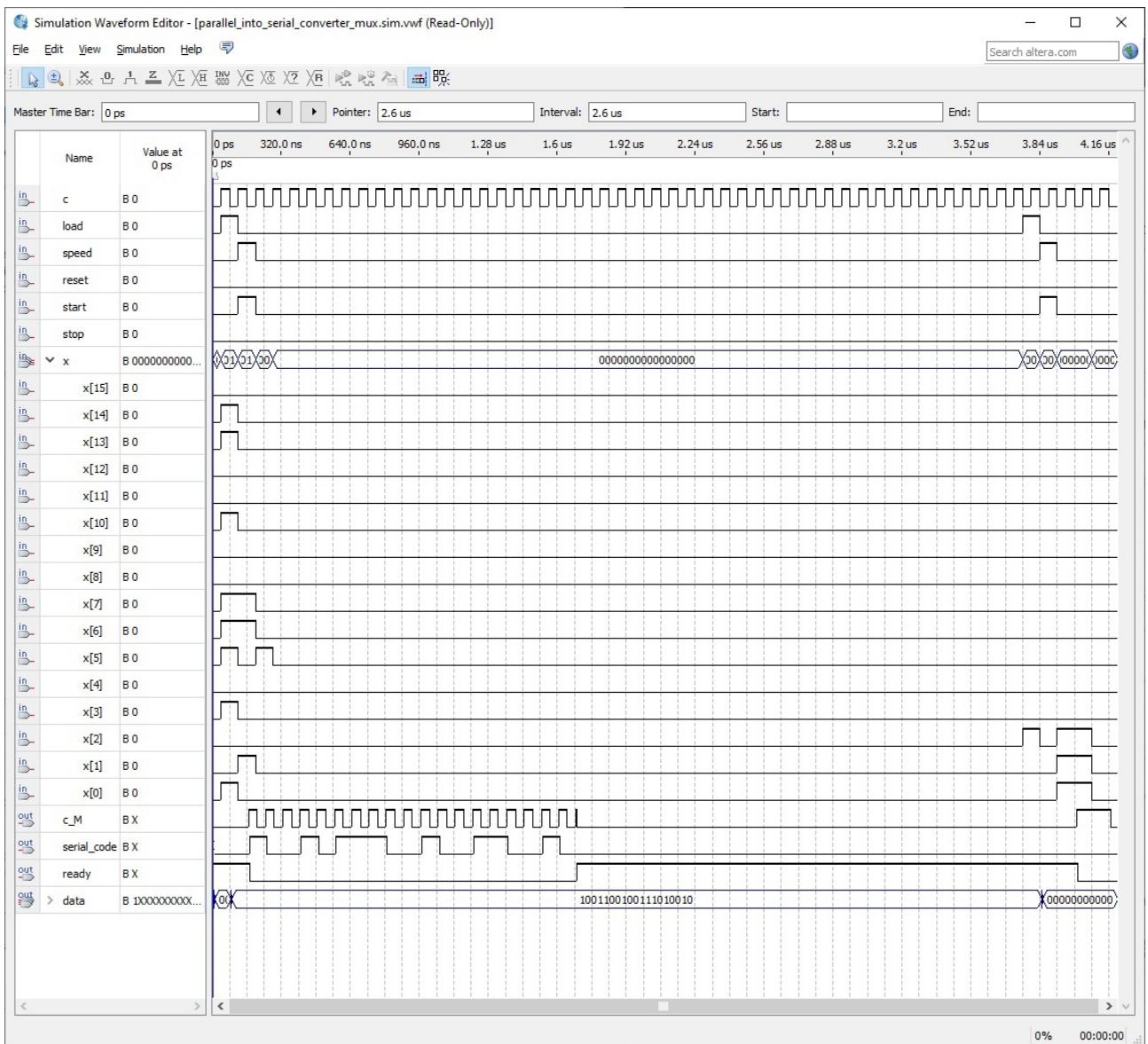


Рисунок 33. Временная диаграмма узла на основе мультиплексора, пример 1

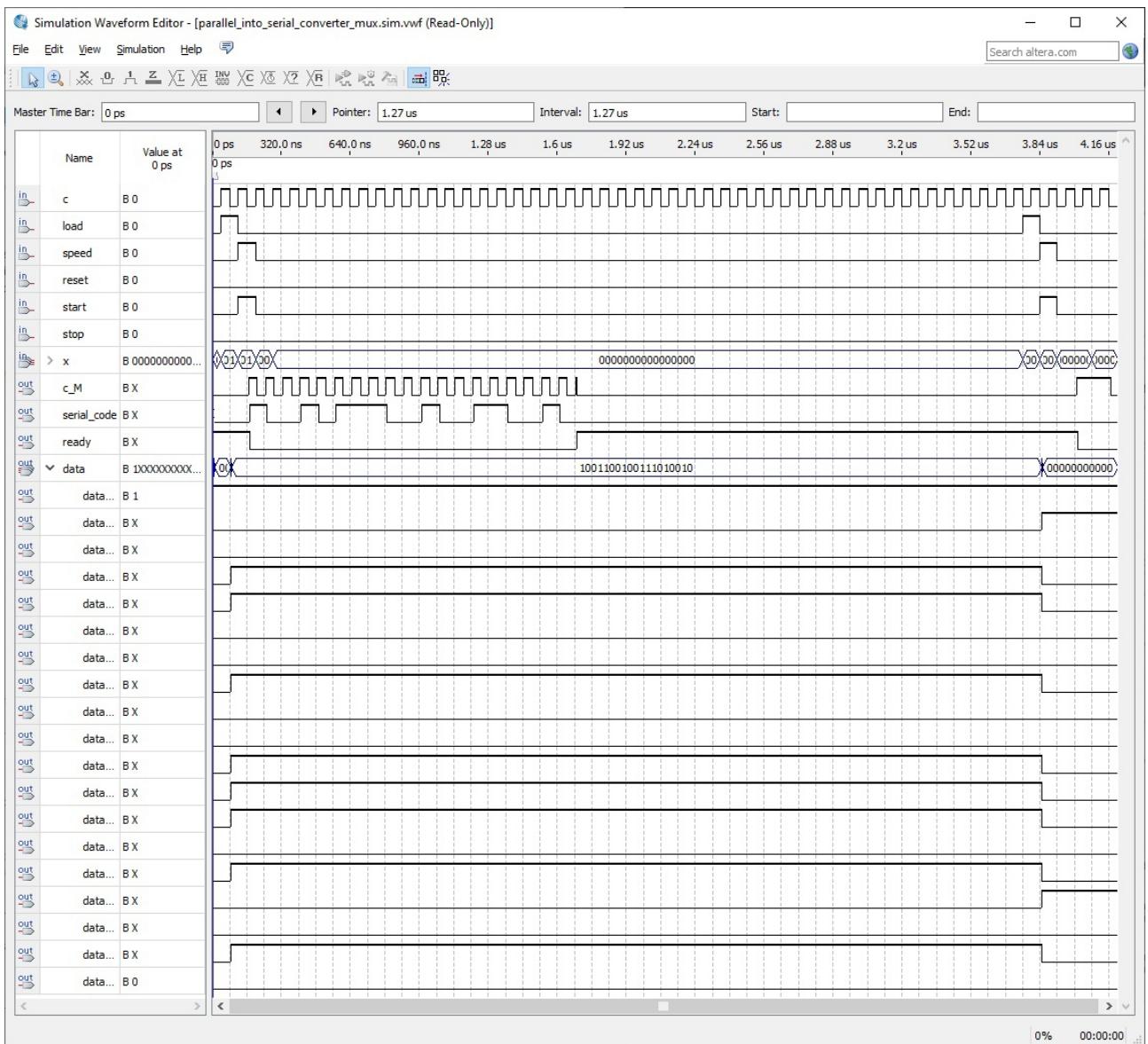


Рисунок 34. Временная диаграмма узла на основе мультиплексора, пример 1

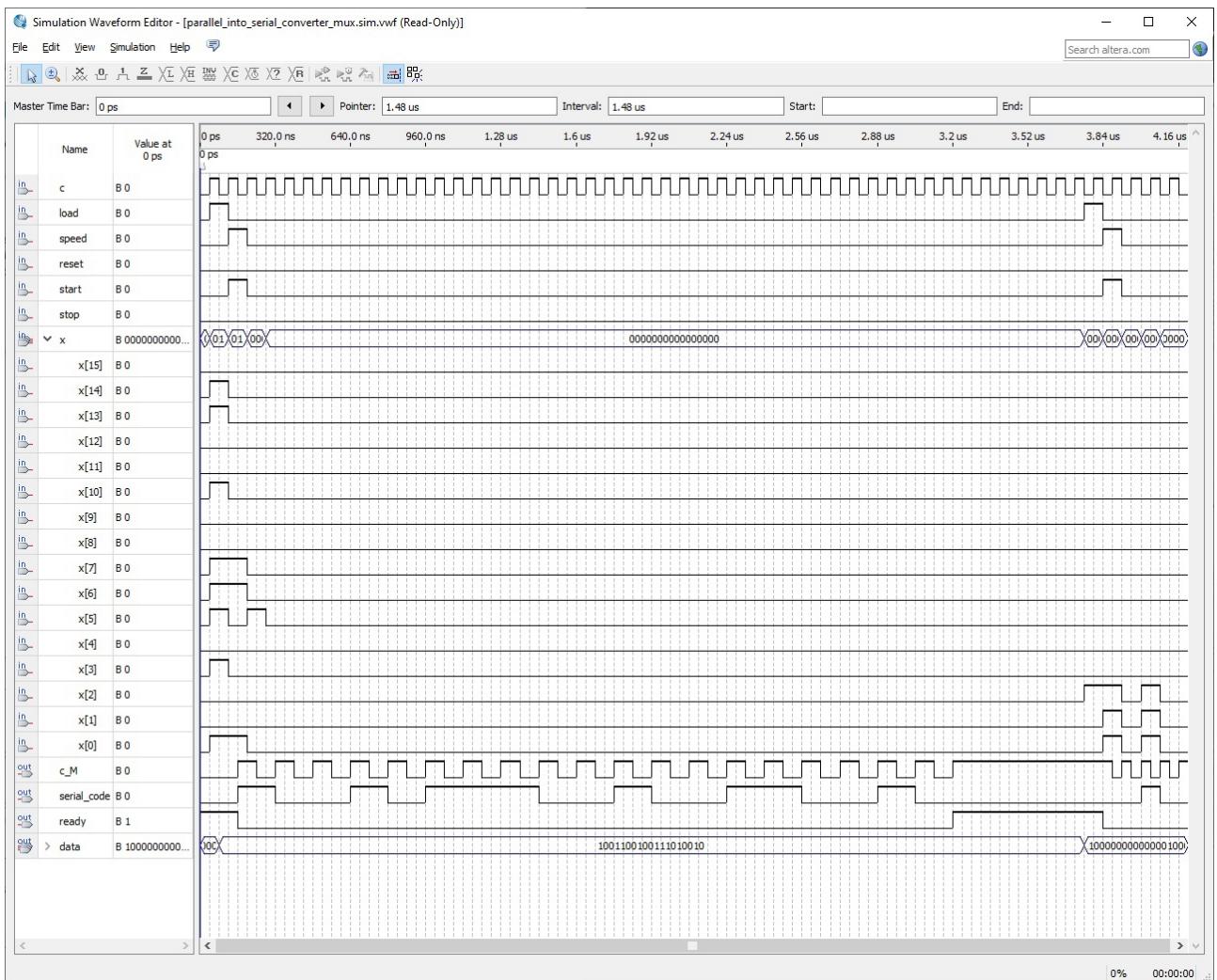


Рисунок 35. Функциональная диаграмма узла на основе мультиплексора,
пример 2

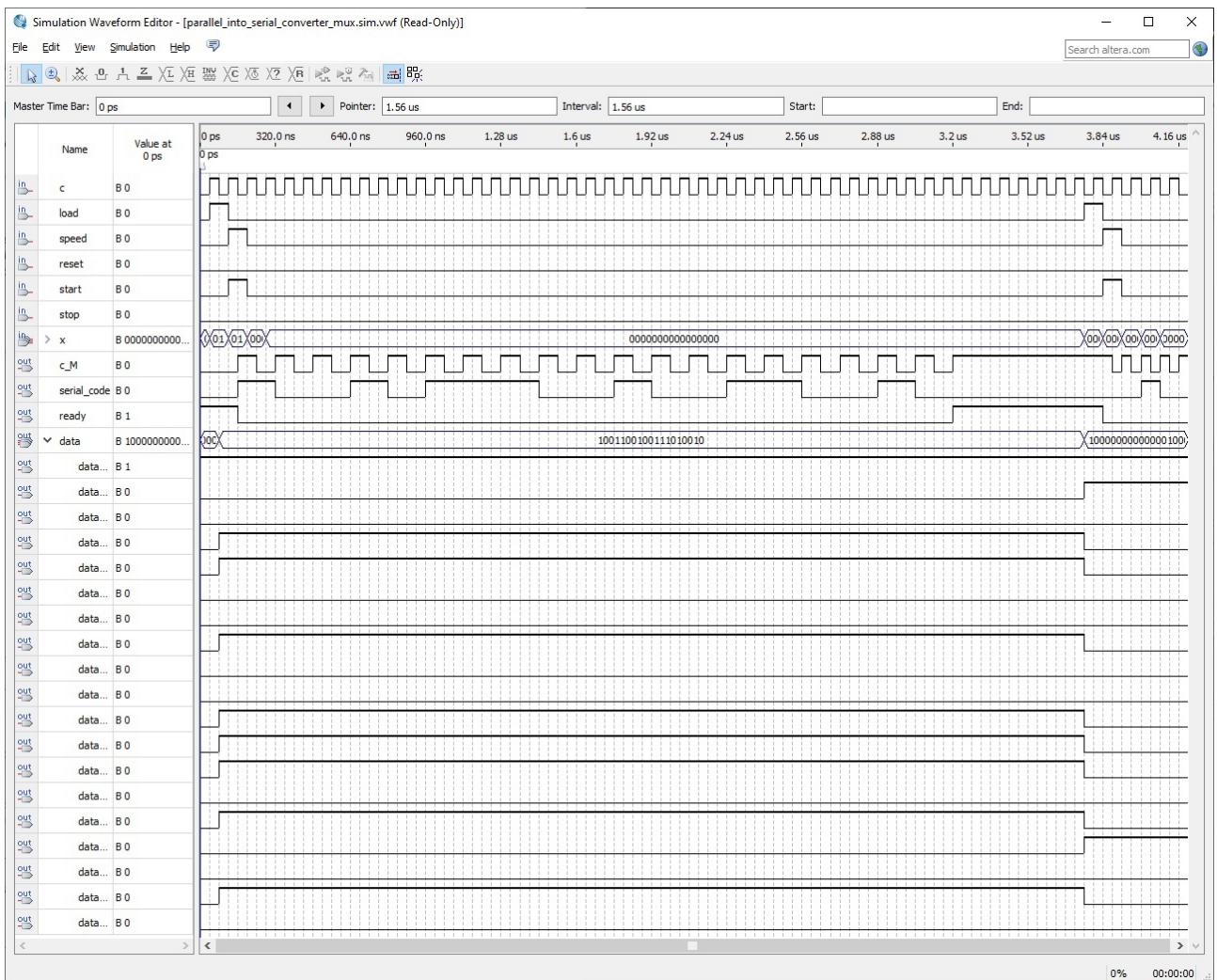


Рисунок 36. Функциональная диаграмма узла на основе мультиплексора,
пример 2

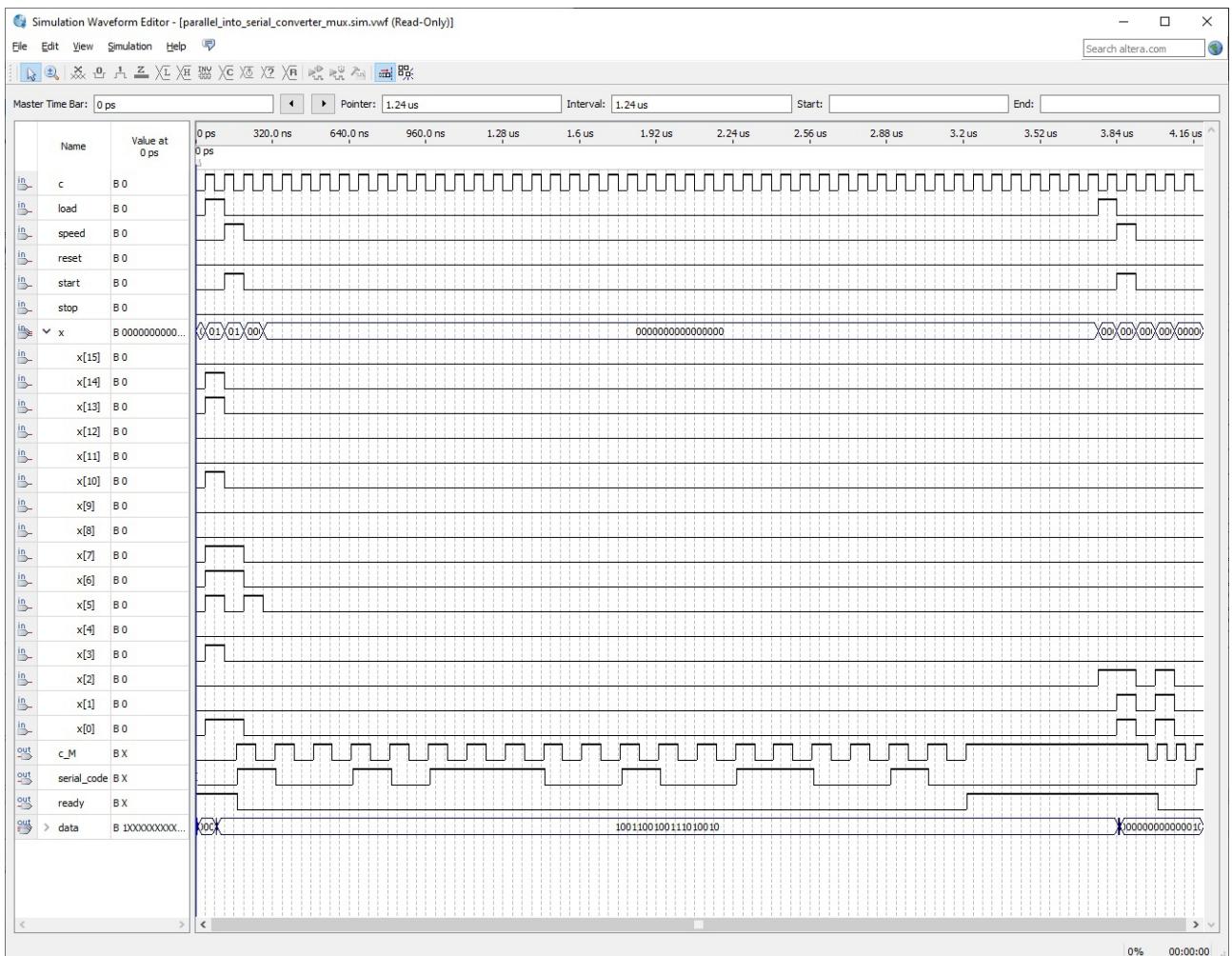


Рисунок 37. Временная диаграмма узла на основе мультиплексора, пример 2

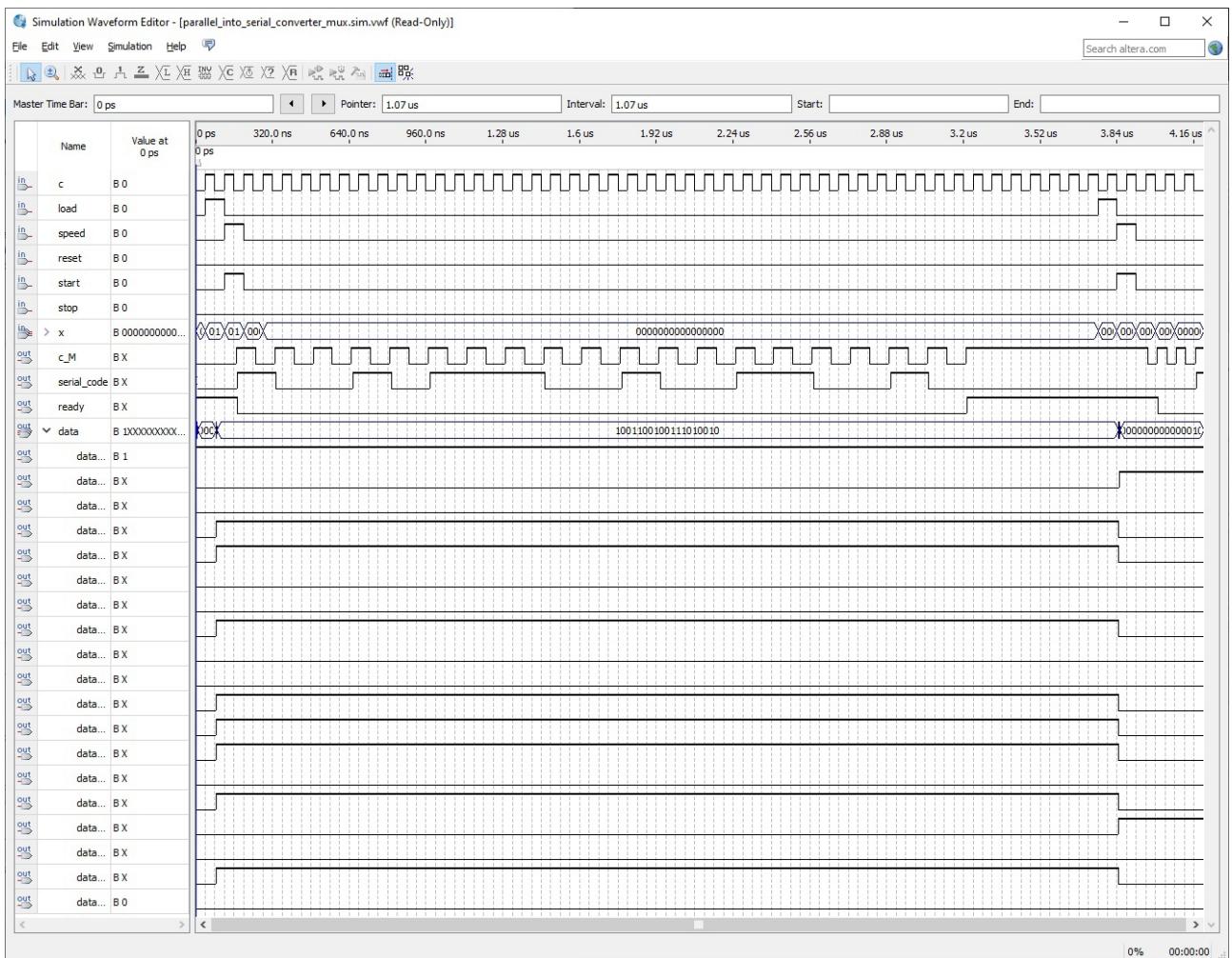


Рисунок 38. Временная диаграмма узла на основе мультиплексора, пример 2

4.3. Выбор оптимальной схемы

Рассмотрев два варианта реализации схемы, можно сделать вывод, что реализация узла на мультиплексоре и параллельном регистре будет более оптимальной, так как данная схема включает в себя использование меньшего количества элементов по сравнению со схемой на параллельно-последовательном регистре, следовательно, включает в себя минимум аппаратных затрат.

В первой реализации схемы на параллельно-последовательном регистре присутствуют два дополнительных инвертора, а также дополнительные подводки сигналов, что заметно увеличивает аппаратные затраты. Несмотря на то, что в реализации на основе мультиплексора и параллельного регистра сделан дополнительный вывод на схеме «`create_ready_signal_mux`», аппаратные затраты это практически не меняет.

5. Определение предельной допустимой частоты генератора тактовых импульсов (ГТИ)

Для обеспечения корректной работы устройства необходима частота генератора в 19200 Гц.

Выбор технической реализации генератора тактовых импульсов зависит от целого ряда факторов. Определяющим фактором является требуемая стабильность частоты выходных импульсов.

Наибольшую стабильность обеспечивают схемы, содержащие в своём составе кварцевые резонаторы. Промышленностью выпускается достаточно обширный круг таких ИС. Основной недостаток этих схем – достаточно высокая стоимость (несколько долларов).

При допустимой нестабильности частоты порядка единиц процентов применяют простые кольцевые генераторы или RC-генераторы, с времязадающими элементами в виде цепочек, содержащих конденсаторы и резисторы.

Для этой схемы времязадающая цепочка RC определяет частоту выходных импульсов, равную приблизительно $0,53/RC$.

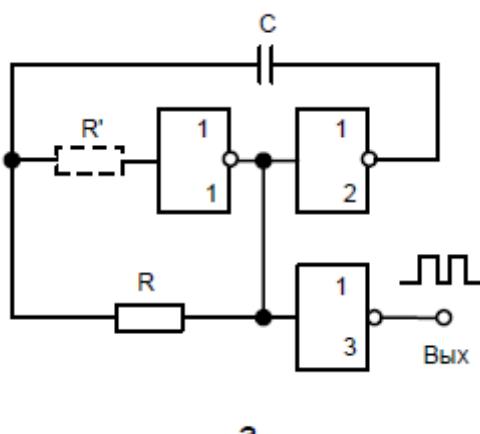


Рисунок 39. Схема генератора тактовых импульсов

Тогда при $R=270\text{ Ом}$:

$$C = \frac{0,53}{W_{импульса} \cdot R} = \frac{0,53}{19,2 \cdot 2 \cdot 10^3 \cdot 270} \approx 51 \cdot 10^{-8} = 51 \text{ нФ}.$$

6. Разработка интерфейса сопряжения схемы узла с процессорной системой, для которой проектируемый узел является внешним устройством

Сопряжение узла с процессорной системой происходит посредством шины Microbus. Для работы устройства используются следующие сигналы шины:

«IOR» – сигнал чтения ПС статуса устройства с шины данных;

«IOW» – сигнал подачи ПС очередной команды на шину данных;

«reset» – сигнал сброса устройства в исходное состояние;

«ready» – сигнал, передаваемый на шину устройством при готовности к обмену данными с ПС.

Шина данных используется для передачи устройству сигнала старта, остановки, сброса, также для загрузки параллельного кода и кода выбора скорости передачи. Адреса регистра загрузки – 80h и 81h. Адрес для команд «start» и «ready» – 82h, адрес для команд «stop» и «reset» – 83h.

Разрешение на запись данных в регистр или триггеры производится с помощью логической схемы управляющего устройства. На узел подается тактирующая последовательность с внешнего генератора. Выходной сигнал узла «serial_code» подаётся на выход устройства.

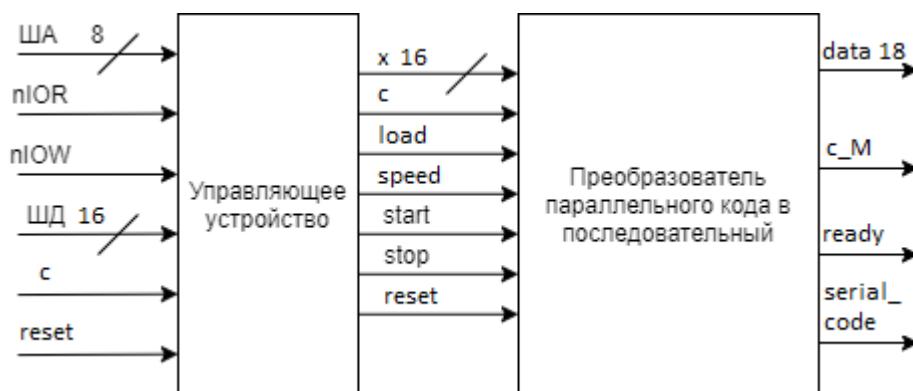


Рисунок 40. Структурная схема интерфейса сопряжения узла с ПС

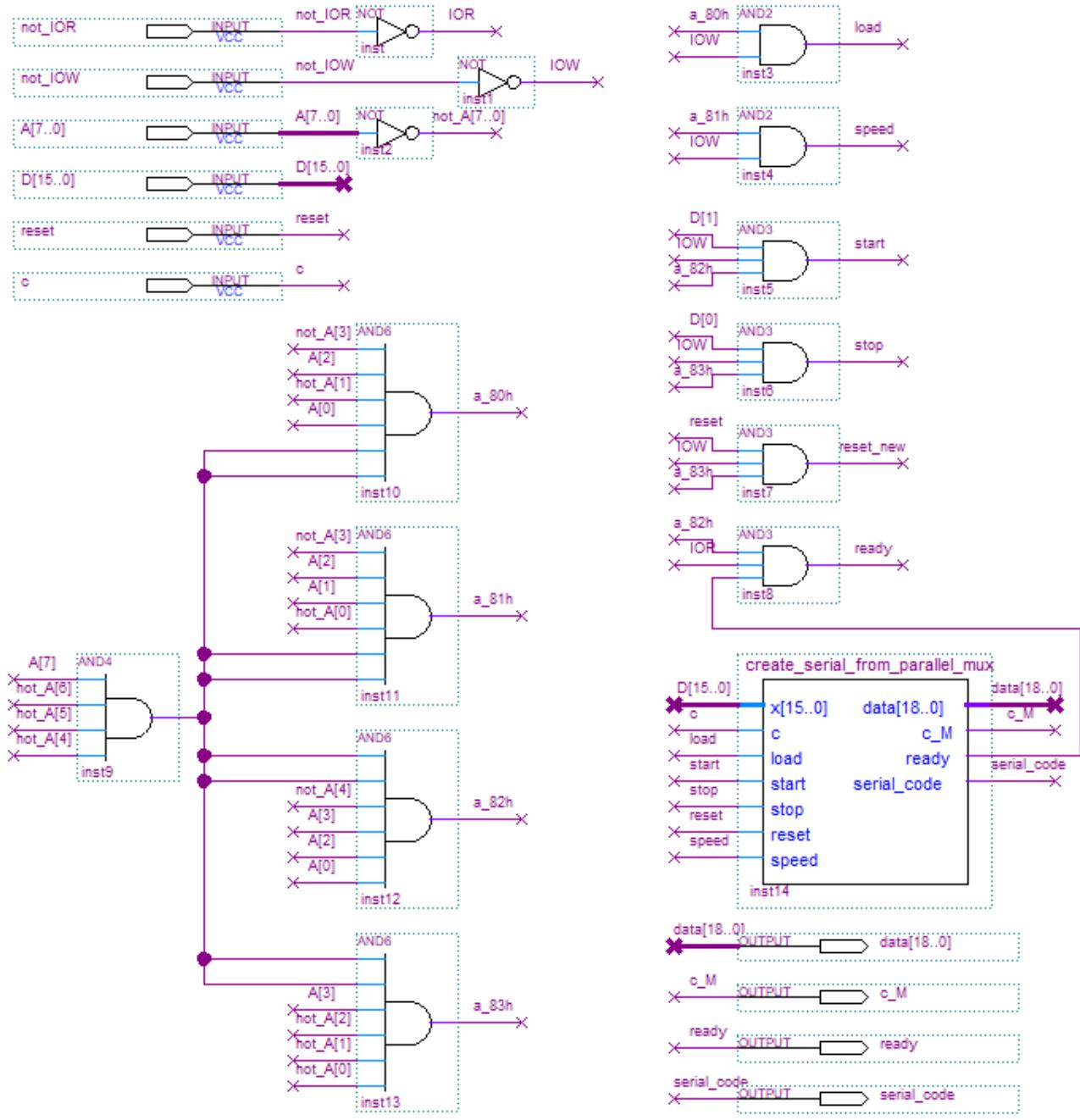


Рисунок 41. Схема интерфейса сопряжения узла с ПС

7. Подробное описание функционирования узла с использованием необходимых временных диаграмм

На примере временной диаграммы для варианта реализации узла на основе мультиплексора приведём подробное описание функционирования узла с использованием необходимых временных диаграмм.

Сигналы «x» передаются в качестве данных для параллельной записи, сигнал «load» в том же такте, что и сигналы «x», передаётся для самой параллельной записи. Сигнал «speed» передаётся в следующем за сигналом «load» такте, чтобы сигналы «x[0]» и «x[1]» могли корректно указать режим работы. Как можно заметить, сигналы «x[0]» и «x[1]» используются именно для скорости работы узла, а единичные сигналы «x[6]» и «x[7]» в том же, что и «x[0]» и «x[1]» такте, учитываться никоим образом не будут.

В том же такте, что и сигнал «speed» подаётся единичный сигнал «start» для начала преобразования параллельного кода в последовательный и для его вывода в последующих тактах сигналом «serial_code», который выводится с помощью синхросигнала «c_M». «c_M» («C message») имеет частоту, указанную пользователем. Параллельно с сообщением выводятся сигналы «data», соответствующий сигналам «x», константам и биту чётности. По завершении работы подаётся единичный сигнал «ready», который сообщает о готовности узла к загрузке новых данных. Как видно из временной диаграммы, вывод сообщения «serial_code» соответствует введённым сигналам «x», а следующая загрузка идёт с другой скоростью, что говорит о корректном функционировании узла.

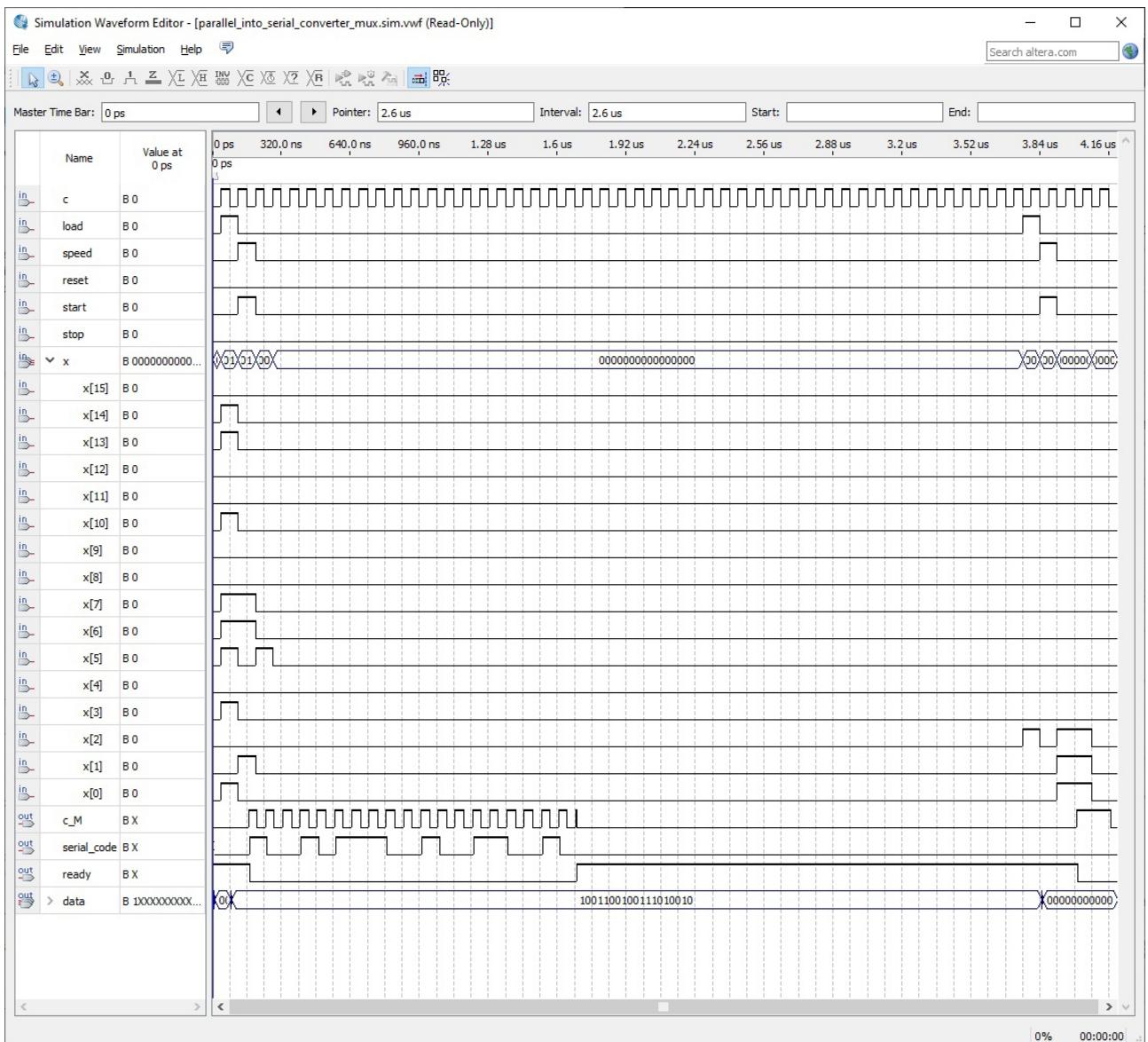


Рисунок 42. Временная диаграмма узла на основе мультиплексора, пример 1

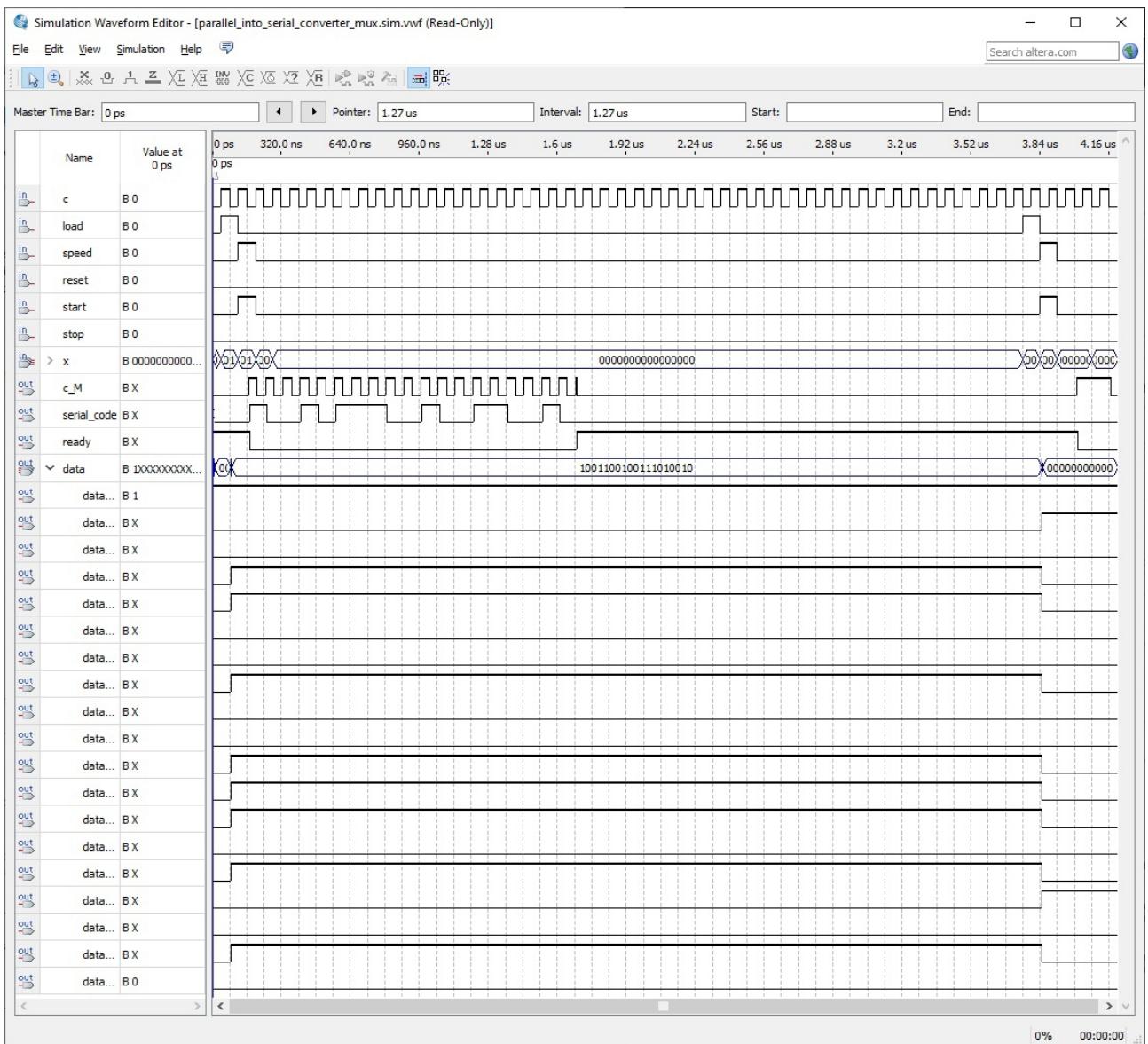


Рисунок 43. Временная диаграмма узла на основе мультиплексора, пример 1

8. Полная принципиальная электрическая схема типового элемента замены

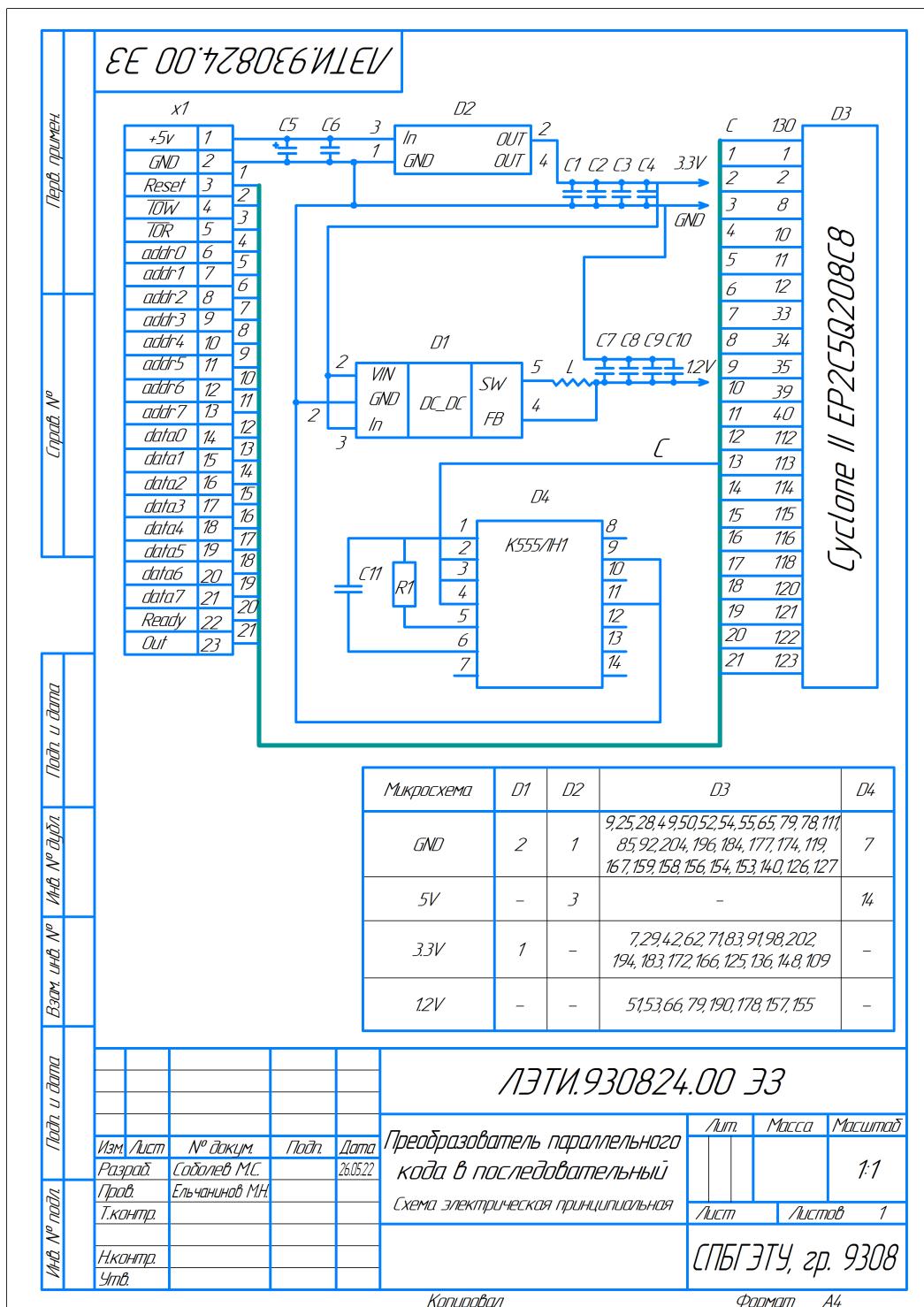


Рисунок 44. Принципиальная схема типового элемента замены

Порядковый №	Наименование	Кол.	Примечание				
				Поз. обозначение	Падж. и дата	Изд. № подл.	Взам. изд. №
<u>Конденсаторы</u>							
C1..C4	K50-5a-H90-0,1мкФ +80%..-20% 0Ж0460.1617У	4					
C5	K50-20-10мкФ-6,38 +50%..-20% 0Ж0464.1837У	1					
C6	K50-5a-H90-0,1мкФ +80%..-20% 0Ж0460.1617У	1					
C7..C10	K50-5a-H90-0,1мкФ +80%..-20% 0Ж0460.1617У	4					
C11	K10-17Б-М47-100пФ +5%..-5% 0Ж0460.1077У	1					
<u>Индуктивности</u>							
L	LQH43CN1ROM, 10мкГн, 1812, 20%, SMD	1					
<u>Микросхемы</u>							
D1	ADP2108-AUJZ-12	1	Предобразователь				
D2	ADP3338AKC-3.3	1	Регулятор напряжения				
D3	Cyclone II EP2C5Q208C8	1	ПЛИС				
D4	K155/Н1	1	Инвертор				
<u>Разъемы</u>							
X1	PLD-40 (DS1021-2x20), Вилка штыревая 254мм 2x20 прямая	1					
ЛЭТИ.930824.00 ГЭЗ							
Изм.	Лист	№ докум.	Падж.	Дата			
Разраб.	Соболев МС		26.05.22				
Проб.	Ельчанинов МН						
Иконопр.							
Утв.							
					Лист	Лист	Листов
							1
Перечень элементов							
СЛБГЭТУ, гр. 9308							

Копировал

Формат A4

Рисунок 45. Принципиальная схема типового элемента замены

9. Краткое заключение по проделанной работе

В ходе выполнения курсовой работы «Проектирование цифровых узлов на микросхемах программируемой логики» на тему «Преобразователь параллельного кода в последовательный для асинхронного протокола передач» были разработаны и рассмотрены два варианта схемы для узла, переводящего параллельный 16-битный код в последовательный и выводящего его, построены их схемы в САПР Quatus II. Затем после построения функциональных схем, функциональных и временных диаграмм была выбрана схема на основе мультиплексора и параллельного регистра, которая несла меньше аппаратных затрат в отличие от схемы на параллельно-последовательном регистре. Далее был разработан и описан интерфейс сопряжения схемы узла с процессорной системой, построена его схема в САПР Quartus II и его принципиальная схема. Также была построена принципиальная схема узла. Таким образом и был разработан узел, преобразующий параллельный код в последовательный с добавлением к нему старт-бита, стоп-бита и бита паритета.

10. Список использованной литературы

1. Онлайн-курс «Элементная база цифровых систем» в LMS Moodle [сайт]. URL: <https://vec.etu.ru/moodle/course/view.php?id=8252>.
2. Справочник по элементам LPM компании «Altera» [сайт]. URL: www.altera.com/literature/catalogs/lpm.pdf.
3. Проектирование цифровых узлов на микросхемах программируемой логики: Методические указания к курсовому проектированию / Сост.: Е. П. Угрюмов, Р. И. Грушвицкий. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2011. 32 с.
4. Бондаренко П. Н., Буренева О. И., Головина Л. К. / Узлы и устройства средств вычислительной техники: учеб.-метод. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2017. 64 с.
5. Угрюмов Е. П. / Цифровая схемотехника: Учеб. пособие для вузов. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2007. – 800 с.: ил.
6. Угрюмов Е. П. / Цифровая схемотехника: учеб. пособие для вузов. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2010. – 816 с.: ил.