**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное автономное образовательное учреждение**

**высшего профессионального образования**

**«Национальный исследовательский ядерный университет «МИФИ»**

**ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ**

**КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ (№12)**

**Отчёт по лабораторной работе по дисциплине**

**«Схемотехника цифровых устройств»**

Преподаватель: Решетько В. М.

Выполнили студенты группы С17-501:

Вёрстов Данила Игоревич,

Вальшин Айдар Рамилевич

Москва, 2019 г.

**Содержание**

1. Описание устройства *Am29f400b*
2. Описание интерфейса
3. Описание сущности и поведения интерфейса
4. Временные диаграммы работы интерфейса
5. Заключение
6. Список использованных источников и инструментов

**Описание устройства *Am29f400b***

***Am29f400b*** — *Parallel NOR-Flash* память марки *Cypress/ Spansion* (производитель — *Cypress Semiconductor*) на 4 Мегабит. Как и любая *Flash*-память, *Am29f400b* имеет набор управляющих шин, в зависимости от конфигурации сигналов на которых происходит чтение, запись, стирание флешки и т.д. Набор данных сигналов показан далее на рис. 1 [1, стр. 6].

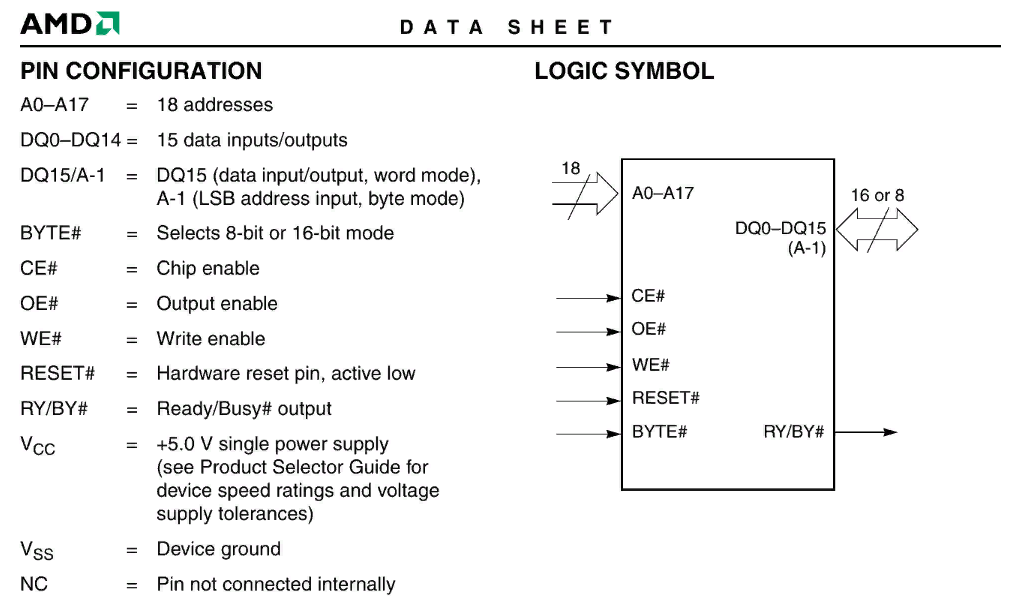


Рис. 1. Конфигурация сигналов *Am29f400b* [1, стр. 6]

При реализации интерфейса *Vcc*, *Vss*, *NC* входы не используются в силу отсутствия влияния разработчика на данные факторы.

Соответственно в документации к устройству описаны правила взаимодействия с данными шинами [1, стр. 8].

Для выполнения любой операции необходимо произвести соответствующую последовательность действий (т.н. командная последовательность). При реализации интерфейса к данному устройству был проведён анализ таблицы командных последовательностей данного устройства, которая вместе с таблицей таймингов и набором временных диаграмм составляют основу работы интерфейса. Соответствующая таблица командных последовательностей, взятая из документации к устройству [1, стр. 17].

Важно отметить, что число циклов последовательности варьируется в диапазоне от 1 до 6, причем чем сложнее проводимая операция, тем больше конкретных циклов необходимо совершить в определенной фиксированной последовательности. Одна из причин данного решения — обезопасить устройство от случайного выполнения “опасных” (неоткатываемых операций, например, стирание флешки).

Важно отметить двойственность реализаций многих операций таблицы, возникающей вследствие наличия в устройстве 2-х режимов работы: *#BYTE* (посылки размером 1 байт) и *#WORD* (посылки размером 2 байта).

Руководствуясь данной логикой выполнения последовательностей, интерфейс выполняет требования, поступающие извне со стороны хоста для взаимодействия с данным устройством.

**Описание интерфейса к устройству am29f400b**

Интерфейс взаимодействия с *Flash*-памятью можно представить в виде промежуточного блока между пользователем-клиентом (***host***) и непосредственно самим устройством (данная *Flash*-память *Am29f400b*).

Указанный блок составлен из двух частей: ***frontend*-часть**, которая взаимодействует с хостом, и ***backend*-часть**, которая взаимодействует с устройством. Взаимодействие есть конфигурация сигналов, существующих по набору портов между хостом и интерфейсом, интерфейсом и устройством.

Логику взаимодействия и обработку сигналов реализует интерфейс. Далее на рис. 2 представлена схема интерфейса.

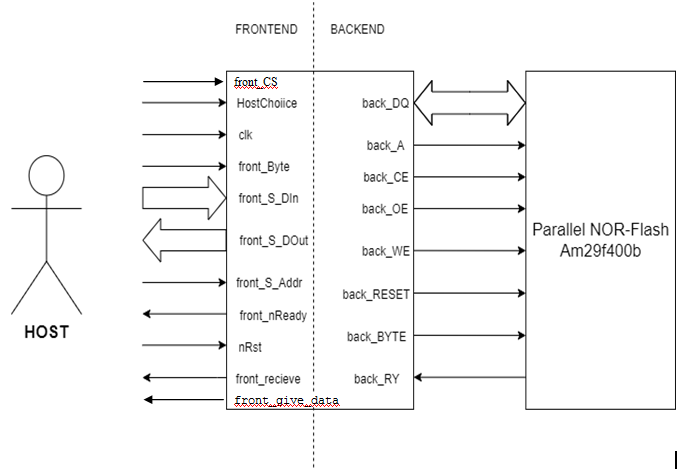
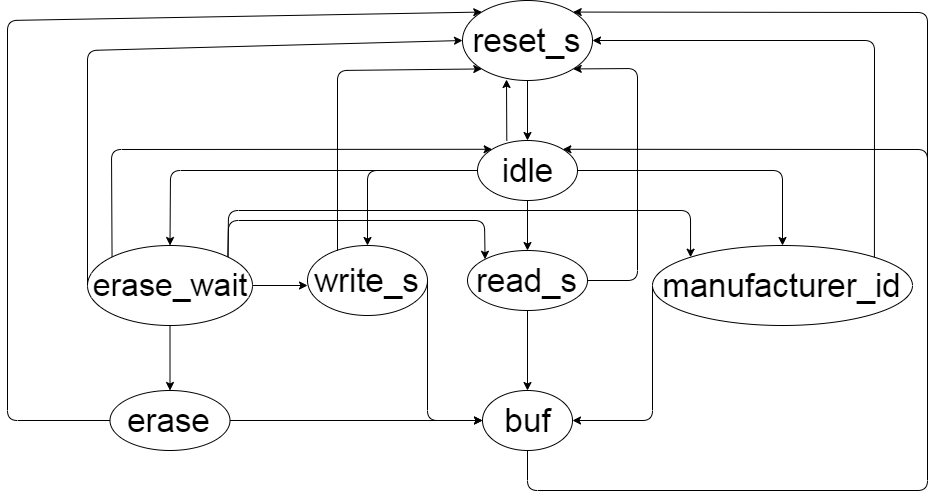


Рис. 2. Структурная схема интерфейса

**Конечный автомат интерфейса**

****

Условия переходов из состояния в состояние описаны в листинге кода.

**Описание сигналов интерфейса**

***Frontend:***

***HostChoice*** : IN std\_logic\_vector(2 downto 0) := (others => '0');

***clk***  : IN std\_logic;

***front\_Byte*** : IN std\_logic;

***nRst*** : IN std\_logic := 'U';

***front\_nReady*** : OUT std\_logic := 'U';

***front\_S\_Addr*** : IN std\_logic\_vector(17 downto 0) := (others => 'U');

***front\_S\_DIn***  : IN std\_logic\_vector(15 downto 0) := (others => 'U');

***front\_S\_DOut*** : OUT std\_logic\_vector(15 downto 0) := (others => 'U');

***front\_recieve*** : OUT std\_logic := '0';

***front\_give\_data*** : OUT std\_logic ;

***front\_CS*** : IN std\_logic := '0';

***Backend:***

***back\_A*** : OUT std\_logic\_vector(17 downto 0) := (others => 'U');

***back\_DQ*** : INOUT std\_logic\_vector(15 downto 0) := (others => 'U');

***back\_CE*** : OUT std\_logic := 'U';

***back\_OE*** : OUT std\_logic := 'U';

***back\_WE*** : OUT std\_logic := 'U';

***back\_RESET*** : OUT std\_logic := 'U';

***back\_BYTE*** : OUT std\_logic := 'U';

***back\_RY*** : IN std\_logic := 'U';

**Правила работы *frontend*** и ***backend* сигналов интерфейса**

Ниже приведена таблица 1, описывающая поведение сигналов *frontend*-части интерфейса.

Таблица 1. Поведение *frontend* и *backend* сигналов интерфейса

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Название сигнала** | **Операция/состояние *FSM*** | | | | | |
| ***idle*** | ***read* 3** | ***write* 4 *(program)*** | ***manufacturer\_ID*** | ***erase* 5** | ***reset* 6** |
| ***HostChoice* 1** | **000** | **001** | **100** | **010** | **011** | **X** |
| ***front\_Byte* 2** | **1** | **1** | **1** | **1** | **1** | **1** |
| ***nRst*** | **1** | **1** | **1** | **1** | **1** | **0** |
| ***front\_nReady*** | **1** | **0** | **0** | **0** | **0** | **0** |
| ***front\_S\_Addr*** | **X** | **ADDR** | **ADDR** | **X** | **X** | **X** |
| ***front\_S\_DIn*** | **X** | **X** | **DATA** | **X** | **X** | **X** |
| ***front\_S\_DOut*** | **X** | **DATA** | **X** | **DATA** | **X** | **X** |
| ***front\_recieve* 7** | **0** | **0-1-0** | **0-1-0** | **0-1-0** | **0-1-0** | **0-1-0** |
| ***back\_A*8** | **X** | **ADDR** | **BIT\_MASK** | **BIT\_MASK** | **BIT\_MASK** | **X** |
| ***back\_DQ*** | **X** | **X** | **DATA** | **DATA** | **DATA** | **X** |
| ***back\_CE*** | **1** | **0** | **0** | **0** | **0** | **1** |
| ***back\_OE*** | **0** | **0** | **1** | **1** | **1** | **0** |
| ***back\_WE*** | **1** | **1** | **0** | **0** | **0** | **1** |
| ***back\_RESET*** | **1** | **1** | **1** | **1** | **1** | **0** |
| ***back\_BYTE*9** | **X** | **X** | ***front\_BYTE*** | ***front\_BYTE*** | ***front\_BYTE*** | **X** |
| ***back\_RY*** | **U** | **U** | **U** | **U** | **U** | **U** |

**Комментарии к таблице 1:**

* [*Note 1*] Порт *HostChoice:*

Возможные ситуации:

Если со стороны *host* отсутствуют какие-либо входные импульсы, что соответствует состоянию ***idle***, то необходимо наличие со стороны host посылки на frontend интерфейса *HostChoice = "000"*. В случае отсутствия посылки по умолчанию всегда устанавливается данное значение.

Если со стороны *host* присутствует требование об обычной операции чтения (***read***), то помимо верной конфигурации прочих сигналов обязательна посылка *HostChoice = "001"*.

Если со стороны *host* присутствует требование об операции чтения *Manufacturer ID* (***manufacturer\_id***), то помимо верной конфигурации прочих сигналов обязательна посылка *HostChoice = "010"*.

Если со стороны *host* присутствует требование об операции записи (***write***), то помимо верной конфигурации прочих сигналов обязательна посылка *HostChoice = "100"*.

Если со стороны *host* присутствует требование об операции стирания всей флешки (***erase***), то помимо верной конфигурации прочих сигналов обязательна посылка *HostChoice = "011".*

Значение *HostChoice* считывается в начале операции. Далее изменения в посылках по порту *HostChoice* игнорируются до завершения выполнения выбранной операции.

* [*Note 2*] Порт *front\_Byte:*

Переключение сигнала *front\_Byte* из одного состояние в другое должно происходить не позднее установления в соответствующее состояние сигнала *HostChoice*. После переключения уровень *front\_Byte* должен оставаться неизменным на протяжение всего времени выполнения требуемой операции.

* [*Note 3*] Комментарии к операции чтения:

При процедуре чтения (***read***) из устройства одновременно с обработкой посылки по шине *HostChoice* устанавливается требуемый адрес. После чего любые защелкивание адреса происходит в течение 40 нс.

* [*Note 4*] Комментарии к операции записи:

При выполнении операции записи (***write***) валидные адрес и данные подаются спустя 310 нс после переключения сигнала *RY/BY* (который идет на *host*) в активный 0.

* [*Note 5*] Комментарии к операции стирания:

При выполнении операции стирания (***erase***) в момент выставления *HostChoice* на erase , происходит скачок *RY/BY* (0-1-0). После этого происходит задержка на 50 нс в целях возможного ожидания смены решения. Только по истечении этого времени начинается операция (при условии, что она не была сменена на другую. В данном случае выполняется соответствующая командная последовательность).

* [*Note 6*] Комментарии к переходу в состояние ***Reset***:

Переход в состояние ***Reset*** не требует окончания текущих операций.

Переход происходит при выставлении сигнала *nRst* = 0.

* [*Note 7*]Комментарии к поведению сигнала *front\_receive*:

При приёме команды со стороны *host* интерфейс подаёт сигнал о том, что команда воспринята и обслуживание началось. Если после импульса (0-1) сигнал не изменяется, то дальше он переходит активный 0.

* [*Note 8*]Комментарии к поведению сигнала *back\_A*:

**BIT\_MASK** - последовательность данных, требуемая устройством для корректного выполнения операции. Подробнее [1, стр. 17].

* [*Note 9*]Комментарии к поведению сигнала *back\_BYTE*:

***front\_BYTE* -** значение сигнала ***front\_BYTE***  на начало операции.

* Комментарии общего характера:

Все операции штатно завершаются при изменении сигналов, при которых они сработали, кроме случаев с ***idle*** (переход зависит от сигнала *HostChoice*), а также случаев, описанных выше.

Сигнал *front\_nReady* находится в активном 0, когда происходит какая-либо операция в устройстве. Если *front\_nReady* находится в активной 1, то начнётся выполняться команда, которая была подана, и после подачи которой произошёл импульс *front\_receive*.

Сигнал *front\_CS* находится в активном 0, когда происходит считывание *HostChoice*. В противном случае считывание последнего сигнала не происходит.

Когда сигнал *front\_give\_data* находится в активной 1, то необходимо подать адрес и/или данные (в случае операции записи).

По перепаду сигнала (установлении в активный 0) *nRst* устанавливаются и сбрасываются регистры, счетчики и т. д.

Состояние ***BUF***: в нем устанавливаются в исходные (необходимые ) значения все регистры.

После каждого состояния происходит переход в состояние в BUF ,а из него в ***idle*** (за исключение случаев, описанных в схеме конечного автомата).

**Описание сущности и поведения интерфейса на языке *VHDL***

Код описания сущности и поведения интерфейса подробно описан в файле по следующей ссылке (ссылка на код из репозитория [6] проекта):

[*https://raw.githubusercontent.com/VerstovDI/am29f400b\_interface/dev/Workspace\_files/behav1.vhd*](https://raw.githubusercontent.com/VerstovDI/am29f400b_interface/dev/Workspace_files/behav1.vhd)

**Временные диаграммы работы интерфейса**

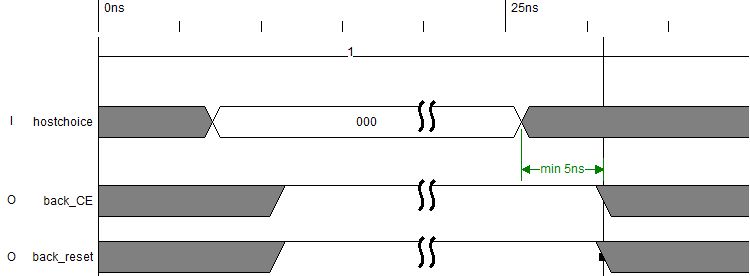
****

Рис. 3. Состояние *idle*

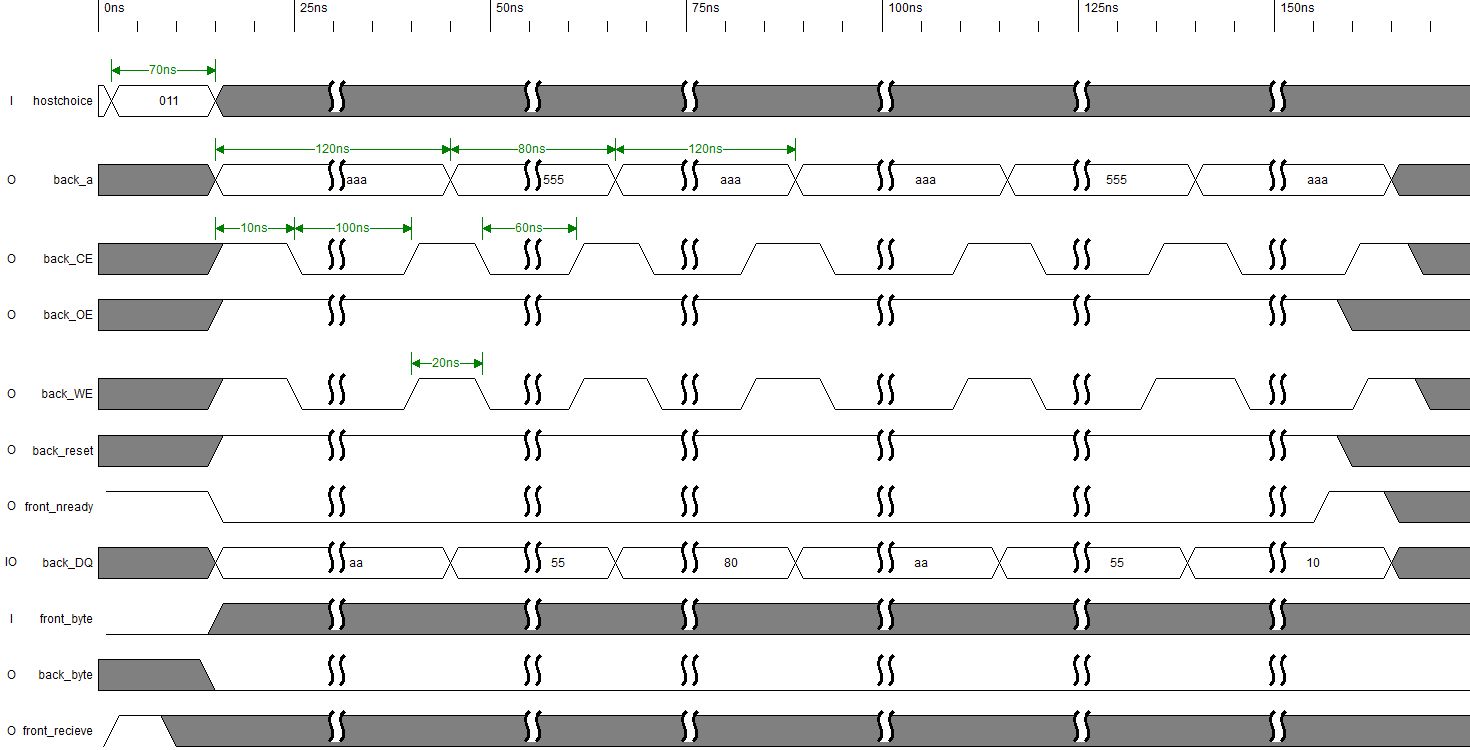


Рис. 4. Состояние *erase* (*byte*)

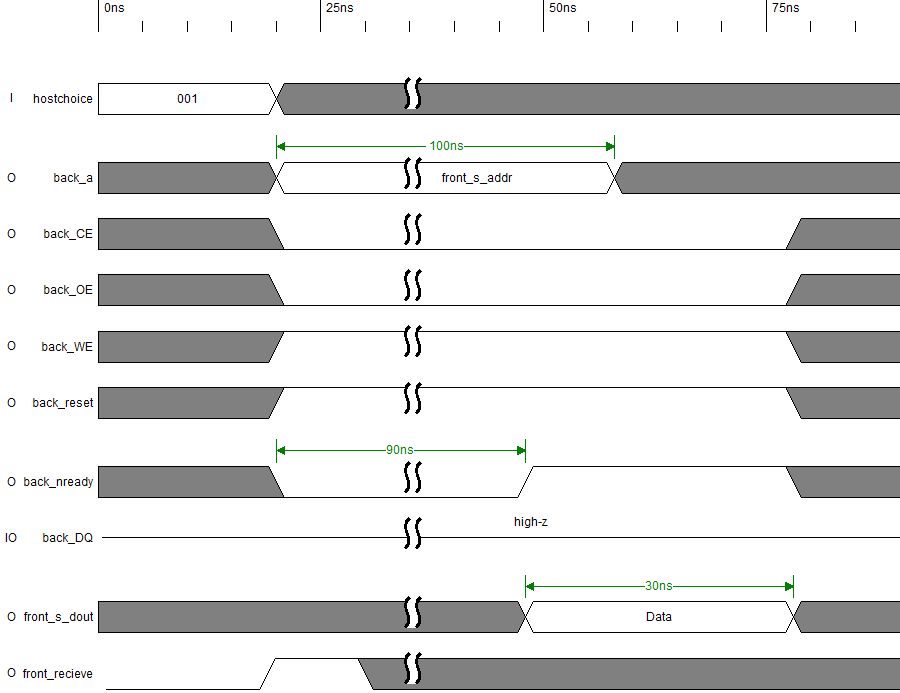


Рис. 5. Состояние *read*

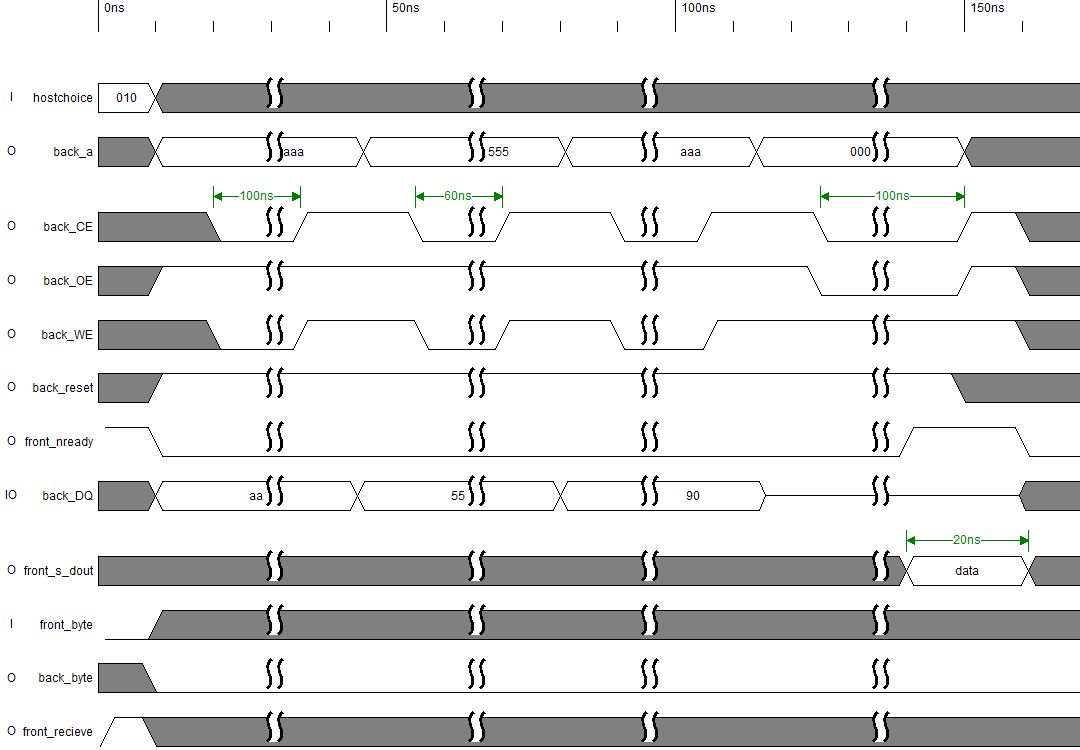


Рис. 6. Состояние *manufacturer\_id* (*byte*)

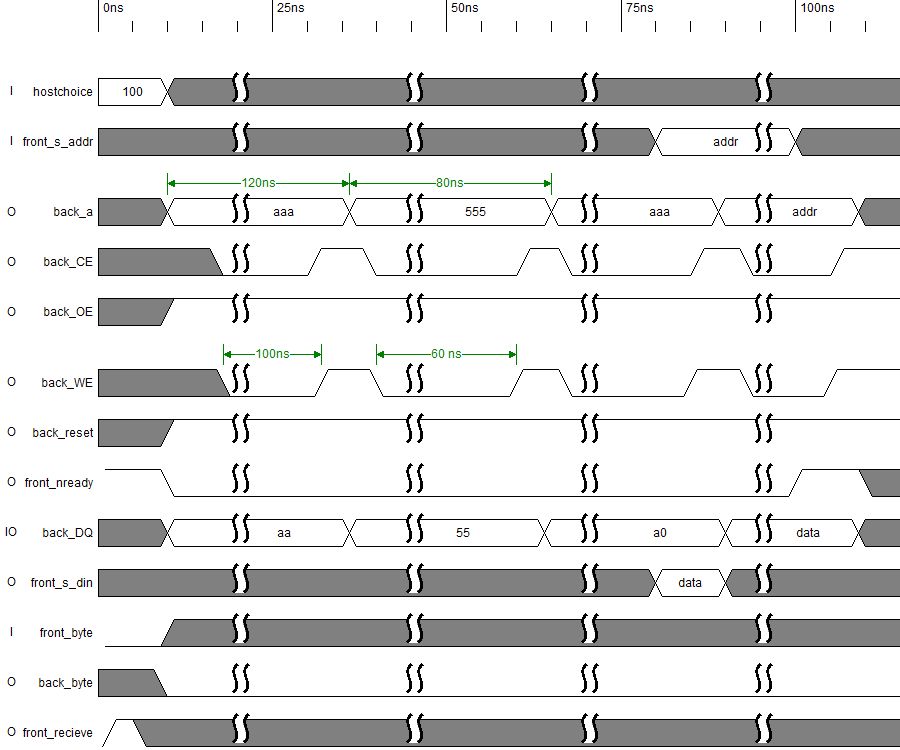


Рис. 7. Состояние *write* (*byte*)

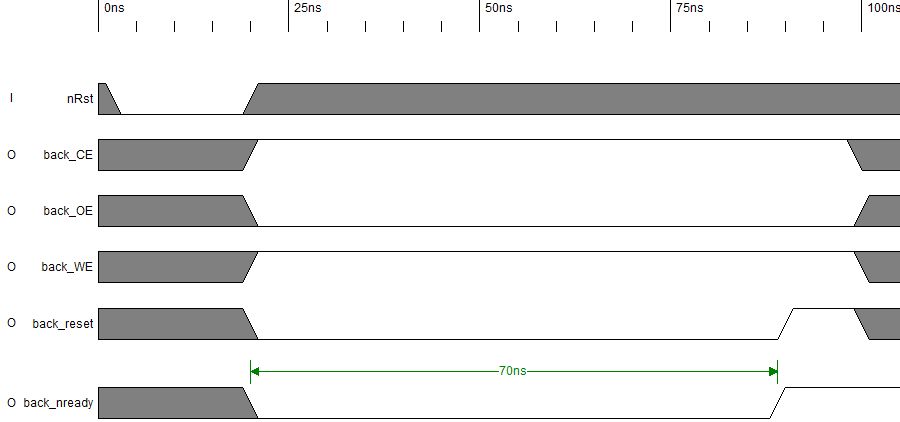


Рис. 8. Состояние *reset*

**Заключение**

На данный момент состояние разработки следующее:

* Составлена и описана логика и правила работы интерфейса к данному устройству
* Реализованы сущность (*entity*), поведение (*architecture*) и тестирующий код (испытательный стенд) *testbench* для интерфейса к данному устройству
* Реализован конечный автомат (*FSM*) для интерфейса к данному устройству
* Реализовано поведение интерфейса для операций чтения из устройства (*read*), чтения идентификатора из устройства (*read manufacturer\_id*), записи в устройство (*write (program)*), стирания информации с устройства (*erase*).
* Составлены соответствующие временные диаграммы.

**Список использованных инструментов и источников**

1. Datasheet Am29f400b. Publication Number 21505. Revision E Amendment 5. Issue Date: November 1, 2006. Последняя дата обращения: 24.12.19. Ссылка на источник:​ [www.farnell.com](http://www.farnell.com) Spansion.
2. <https://freemodelfoundry.com/> — интернет-ресурс — Дата обращения: 21.11.19. (Сайт с vhdl-кодом модели am29f400b и fmf-библиотеками)​.
3. <https://github.com/VerstovDI/am29f400b_interface> — Электрон. ресурс — Дата обращения: 24.12.19 (ссылка на репозиторий проекта).