

Начальная работа с max-plusII

I. Создание графического файла и проекта

1. Создать папку для проекта с названием из латинских букв и цифр
2. Запустить пакет
3. Вызвать графический файл на редактирование (Кнопкой вверху слева **NEW**. Если ЛК, то появится меню “**NEW**”\ **File Type**. В нем выбрать **Graphic Editor File**, расширение **.gdf** и нажать **OK**). Появится окно графического редактора с именем файла в заголовке.
4. Сохранить пустой файл командой **File \ Save as**. Дать файлу имя (латинскими буквами и цифрами). Такое же имя будет иметь проект и файлы проекта с разными расширениями. Сохранить файл в специально созданной папке.
5. Создать проект командой **File \ Project \ Set Project to Current File**
6. Проверить наличие проекта и главного файла командой **max-plusII \ Hierarchy Display**. Появится окно **Hierarchy Display**, в котором слева вверху будет имя проекта и рядом символ с именем графического файла. После проверки закрыть окно.
7. Заполнить графический файл символами элементов логической схемы. Для вызова списка символов активизировать команду **Symbol \ Enter Symbol** или двойным ЛК в поле графического окна. Откроется окно **Enter Symbol**, где в строке **Symbol Name** можно написать имя элемента (Например, **input** или **and2**) и нажать **OK**. Символ появится в графическом окне. Имя символа можно выбрать, также, из списка в **Symbol File**, который раскрывается при двойном ЛК на библиотеке из списка **Symbol Libraries**. Для передачи символа в графическое окно: ЛК на имени символа и **OK**.
8. Дать названия входным и выходным сигналам после двойного ЛК на каждом месте **PIN_NAME**.
9. Соединить входы и выходы сигналов и логических элементов рисованием трассы с нажатой ЛК.
10. Проверить схему на отсутствие ошибок и предупреждений командой **File \ Project \ Save & Check**. Должно появиться сообщение: **Project Save & Check was successful**

0 errors
0 warnings

Закрыть сообщения компилятора на экране.

При повторном открытии графического файла его нужно заново связать с проектом:

(**File \ Project \ Set Project to Current File**) и проверить на отсутствие ошибок:

(**File \ Project \ Save & Check**.)

II. Анализ логической схемы

1. Вызвать на экран пакета окно логического анализа **Waveform Editor file .scf**
2. Сохранить его под именем графического файла, но с расширением **.scf** в папке проекта (командой **File \ Save as** Окно сохранения предложит этот вариант).
3. Вызвать и откомпилировать графический файл: **File \ Project \ Save & Compile**
4. Если компиляция прошла успешно (без ошибок), то вызвать в окно логического анализа входы и выходы:

Node \ Enter Nodes from SNF.

Появится окно **Enter Nodes from SNF**

- 4.1. В этом окне нужно выбрать список **List**. В подокне **Available Nodes & Groups** появится список входных и выходных логических сигналов
- 4.2. Список сигналов перевести в подокно **Available Nodes & Groups** кнопкой **=>** и нажать **OK**. Список сигналов появится в окне анализатора слева.
- 4.3.. Отформатировать временную ось. Для этого:
 - 4.3.1. Вызвать редактирование конечного времени анализа командой **File \ End Time**.
 - 4.3.2. В появившемся окне **End Time** установить общее время анализа (например, **Time 100.0ms**).

4.3.3. Командой **View \ Time Range** вызвать окно **Time Range**, в котором задать **From**: - начальное время анализа и **To**: - конечное время анализа на экране. (Например, **From=0.0ms; To=100.0ms**).

5. Установить на график сетку, которая будет определять шаг входных сигналов командой **Options \ Grid Size**. (Например, **Grid Size=2.0ms**). Установить, также, режим отображения сетки на экране: **Options \ Show Grid** (если не установлен по умолчанию).

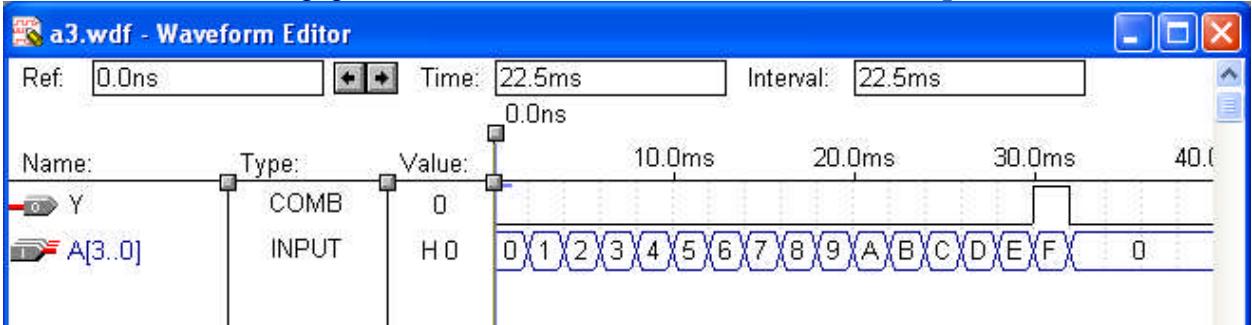
6. Задать временную диаграмму изменения входных логических сигналов во времени. Для этого на временной диаграмме выделять протяжкой с нажатой ЛК фрагменты значений конкретного сигнала и присваивать им логические значения (0 или 1) кнопками на экране пакета слева.

7. Запустить анализ командой **File \ Save, Compile & Simulate**. Входные и Выходные сигналы отразят логику работы устройства в поле диаграмм.

III. Создание проекта с помощью графического редактора **Waveform Editor file** с расширением **.wdf**

1. Вызвать в поле редактирования редактор **Waveform Editor file** с расширением **.wdf**
2. Присвоить ему имя и связать с проектом (как это делалось с графическим файлом)
3. В колонку **Name:** занести имена входных и выходных сигналов (с большой латинской буквы). Если вводятся имена элементов массива, то в поле **Name:** можно занести одно имя массива и его размерность (Например, **A[3..0]**).

4. Присвоить логические значения входным и выходным сигналам. Если группа входных сигналов задана массивом, то вводятся все варианты численных значений в двоично-десятичном формате с помощью команды **Edit\Insert\Group Value:**



Предварительно, участок диаграммы, в который вводится число, выделяется перетаскиванием курсора с нажатой левой кнопкой мыши.

5. Готовый файл нужно связать с проектом командой

File \ Project \ Set Project to Current File,

А также сохранить и откомпилировать командой

File \ Project \ Save & Compile (Ctrl+L).

Ошибок и предупреждений быть не должно.

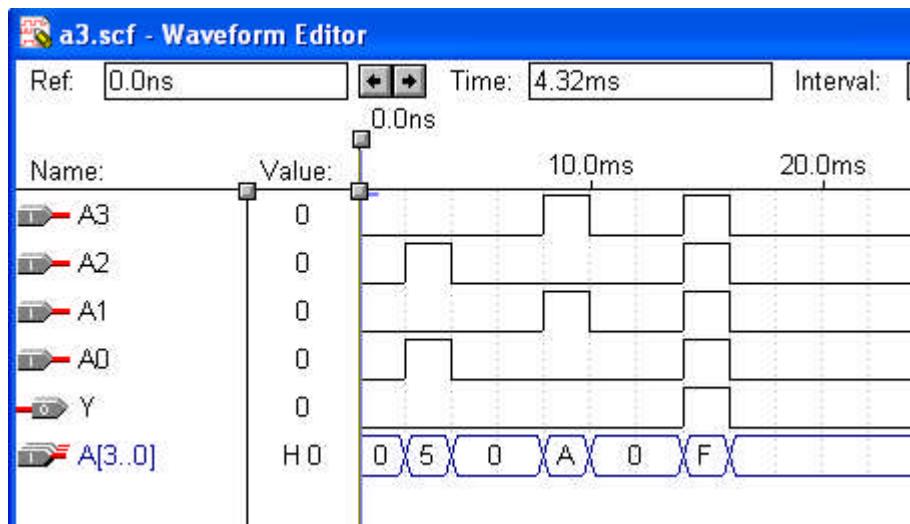
6. Анализ работы схемы выполняется в окне файла редактора **Waveform Editor file .scf**

Имя файла должно совпадать с именем проекта. Здесь вводятся имена входных и выходных сигналов, задаются произвольные варианты входных сигналов и наблюдается реакция выходных после команды

File \ Project \ Save, Compile & Simulate (Ctrl+Shift+L)

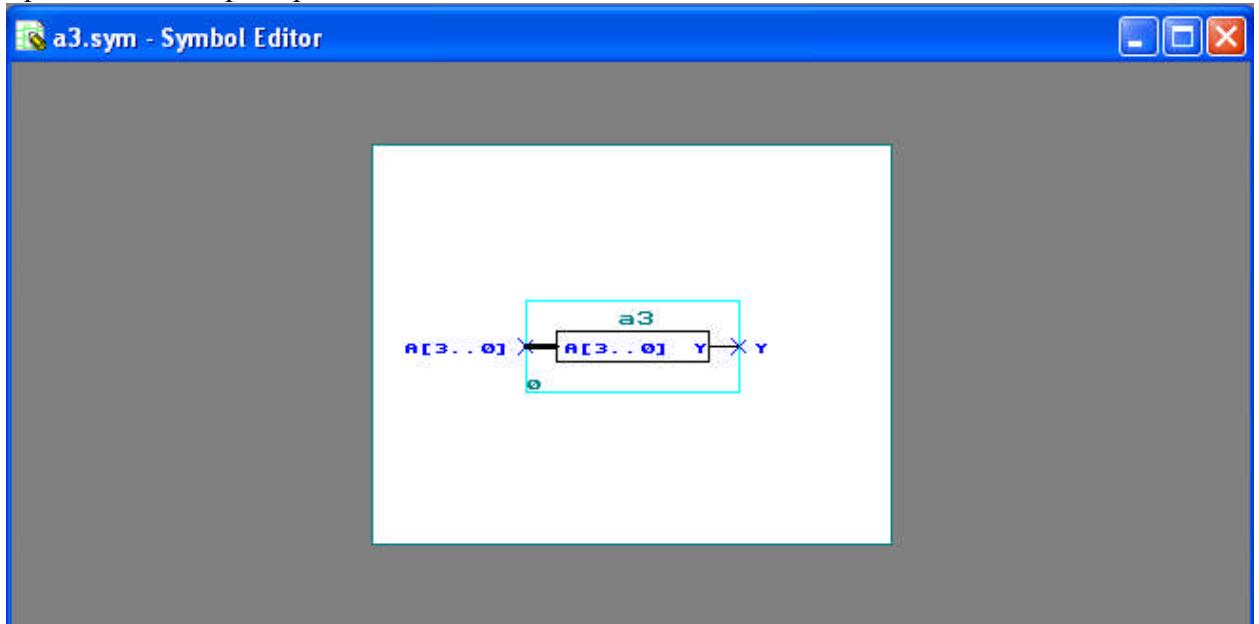
Пример результата:

В примере показано поле файла с именами входных компонент и их массива. Поэтому можно наблюдать значения отдельных компонент и численный код массива входных сигналов.



IV. Создание графического примитива разработанной схемы.

1. В окно программы вызывается главный файл проекта (графический с расширением **.gdf** или временной с расширением **.wdx**).
2. Если главный файл успешно откомпилирован и окна сообщений компилятора закрыты, то активизируется команда **File\Default Symbol**. При этом в проекте создается файл с расширением **.sym**, который можно предварительно посмотреть, если вызвать на экран схему иерархии проекта (**Hierarchy Display**), щелкнуть на примитиве этого файла правой кнопкой и запустить команду **Open Editor**. На экране появится изображение примитива. Например:



Теперь примитивом можно пользоваться, вызывая его по имени в поле графического файла из другого проекта. Имя можно изменить, если открытый для просмотра файл **.sym** пересохранить командой **File\Save as**.