Федеральное агентство по образованию

Государственное образовательное учреждение

высшего профессионального образования

«ЧЕЛЯБИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

**Дипломная работа**

**Счетное устройство видеоимпульсов на ПЛИС**

|  |  |  |  |
| --- | --- | --- | --- |
| Факультет: | Физический | Исполнитель: | Ларин Кирилл Владимирович |
| Кафедра: | Радиофизика и электроника | Группа: | ФФ‑504 |
| Специальность: | 013800 – Радиофизика и электроника | Научный руководитель: | ОАО ЧРЗ «ПОЛЕТ»,  Инженер-конструктор 1 кат., Рацебуржинский С.Л. |
|  |  | Рецензент: | Доктор технических наук, ЮУрГУ, профессор, Родионов В.В. |

Дата защиты: 21 июня 2006г.

Оценка: отлично

Челябинск – 2006

**Содержание**

Введение 3

1 Актуальность темы 3

2 Цели и задачи 5

3 Анализ задания и выбор платформы 5

1. Составление схемы устройства 7

2. Выбор элементов 9

2.1 Выбор ПЛИС. Описание внутренней структуры ПЛИС 9

2.2 Компаратор 13

2.3 Генератор тактовой частоты 15

2.4 Индикатор 16

2.4.1 Описание контроллера HD44780 17

2.4.2 Подключение ЖКИ-модуля 18

2.4.3 Программирование и управление ЖКИ-модуля: 25

2.5 Стабилизаторы 32

2.5 Программатор ByteBlaster 34

2.6 Сборка устройства 35

3. Конфигурирование ПЛИС 36

3.1 Система проектирования MAX+plus II 36

3.2 Описание программы конфигурации ПЛИС 36

3.2.1 Антидребезговая система (antibounce) 37

3.2.1,1 Встроенная макрофункция – счетчик lpm\_counter 38

3.2.2 Двоичо-десятичный счетчик (counter10) 40

3.2.3 Устройство управлением индикатора (wh1602LCD) 41

3.2.3,1 Машина конечных автоматов 42

3.2.4 Делитель тактовой частоты для работы индикатора(divFreq) 46

3.2.5 Общая структура программы 46

Заключение 47

Литература 48

Приложение 1 (Принципиальная схема устройства) 49

Приложение 2 (Список портов ввода вывода ПЛИС epm 3256a) 50

Приложение 3 (Текст программы) 51

### Введение

#### 

#### 1 Актуальность темы

Реальная перспектива использования человеком огромных энергий, скрытых в недрах атома, появилась впервые в 1939 году. На сегодняшний день широкое практическое применение получают различного рода ядерные излучения, несмотря на то, что они опасны для организма человека и в то же время неощущаемы, поэтому для обнаружения и измерения ядерных излучений необходимы специальные приборы.

Основной частью приборов для регистрации ядерных излучений является элемент, воспринимающий излучения, - детектор излучения. Для этой цели используются счетчики разных типов, позволяющие зарегистрировать попавшую в него частицу в виде кратковременного электрического тока – импульса. Наиболее широкое применение имеют газоразрядные счетчики, работа которых основана на ионизирующем действии ядерного излучения. Постепенно их начинают вытеснять сцинтилляционные счетчики (СС), действие которых основано на регистрации вспышек света, возникающих в некоторых веществах под ударами частиц. Основными элементами такого счетчика являются вещество, люминесцирующее под действием заряженных частиц (сцинтиллятор), и [фотоэлектронный умножитель](http://www.cultinfo.ru/fulltext/1/001/008/117/318.htm) (ФЭУ). В начале 20 века визуальные наблюдения световых вспышек (сцинтилляций) под действием ионизирующих частиц (-частиц, осколков деления ядер) были основным методом ядерной физики. Позднее СС был полностью вытеснен.

Его возвращение в ядерную физику произошло в конце 40-х гг., когда для регистрации сцинтилляций были использованы многокаскадные ФЭУ с большим коэффициентом усиления, способные зарегистрировать чрезвычайно слабые световые вспышки. Принцип действия СС состоит в следующем: заряженная частица, проходя через сцинтиллятор, наряду с ионизацией атомов и молекул возбуждает их. Возвращаясь в невозбуждённое (основное) состояние, атомы испускают фотоны ([Люминесценция](http://www.cultinfo.ru/fulltext/1/001/008/072/061.htm)). Фотоны, попадая на катод ФЭУ, выбивают электроны ([Фотоэлектронная эмиссия](http://www.cultinfo.ru/fulltext/1/001/008/117/317.htm)), в результате чего на аноде ФЭУ возникает электрический импульс, который далее усиливается и регистрируется. Детектирование нейтральных частиц (нейтронов, -квантов) происходит по вторичным заряженным частицам, образующимся при взаимодействии нейтронов и -квантов с атомами сцинтиллятора.

Для того чтобы световая вспышка была зарегистрирована ФЭУ, необходимо, чтобы спектр излучения сцинтиллятора совпадал со спектральной областью чувствительности фотокатода ФЭУ, а материал сцинтиллятора был прозрачен для собственного излучения. Для регистрации «[медленных нейтронов](http://www.cultinfo.ru/fulltext/1/001/008/074/870.htm)» в сцинтиллятор добавляют Li или В. Для регистрации «быстрых нейтронов» используются водородсодержащие сцинтилляторы. Для спектрометрии -квантов и электронов высокой энергии используют Nal (Tl), обладающий большой плотностью и высоким эффективным атомным номером.

ФЭУ, предназначенные для СС, должны обладать высокой эффективностью фотокатода (до 2,5%), высоким коэффициентом усиления (108—108), малым временем собирания электронов (~ 10–8 сек), при высокой стабильности этого времени. Последнее позволяет достичь разрешающей способности по времени СС 10–9 сек. Высокий коэффициент усиления ФЭУ наряду с малым уровнем собственных шумов делает возможной регистрацию отдельных электронов, выбитых с фотокатода. Сигнал на аноде ФЭУ может достигать 100 В в виде импульсов трапециевидной формы (видеоимпульсов).

Чтобы не только обнаружить ядерное излучение, но и измерить его интенсивность, недостаточно одного детектора излучения. Необходимы еще электронные устройства, подсчитывающие число электрических импульсов, то есть число попавших в детектор частиц, и устройства, показывающие результат подсчета [1].

Данная работа посвящена разработке электронного устройства считающего число электрических импульсов.

#### 

#### 2 Цели и задачи

1. Необходимо реализовать счетное устройство апериодических видеоимпульсов с заданными параметрами:

* Диапазон изменения амплитуды входного сигнала 5…20В;
* Длительность импульса ф, не менее 10 нс;
* Минимальный интервал между импульсами , 10 мкс.

1. Для отображения счета необходимо наличие индикатора. Реализовать индикатор, на котором высвечивается число импульсов в непрерывном режиме счета через некоторый промежуток времени.
2. Управление устройством осуществляется посредством кнопок: старт/стоп – (начало счета/конец счета), сброс – (сброс счетчика).

#### 3 Анализ задания и выбор платформы

Конечно, реализовать простой счетчик на дискретных элементах (триггерах), что может быть проще? Однако сложность заключается в том, что разработка счетчика на дискретных элементах потребует сложной настройки, что увеличит время разработки и цену устройства. Для моих целей нужен высокоскоростной счетчик. Реализовать его нужно на современной элементной базе. Платформы, на которых можно реализовать счетчик, на сегодняшний день нашлось две ПЛИС и микроконтроллеры, был сделан выбор в пользу первой, то есть ПЛИС так как она легче поддается функциональным изменениям (в дальнейшем это устройство может быть использовано в других целях) и тактовая частота обработки сигнала не фиксирована как у микроконтроллера, её можно задавать аппаратно и делить её в зависимости от необходимости. Итак ПЛИС (Программируемая Логическая Интегральная Схема ).

Из наиболее известных производителей ПЛИС следует отметить фирму Altera. Небольшая, вначале, компания удачно решила задачи стоящие перед ними в начале (определить элементарные базис ПЛИС, разработать математические методы синтеза устройств в выбранном базисе, создать интегрированную систему проектирования цифровых устройств на ПЛИС), путем постепенного согласованного усложнения элементной базы и средств проектирования. Ее успех ко второй половине 90-х годов вывели её в число основных производства микросхем ПЛИС.

Была выбрана ПЛИС семейства MAX 3000 EPM3256A

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип микросхемы | Выходы | I/O | Триггеры | Ячейки | Мах частота MHz |
| EPM3032A | 4 | 30 | 32 | 32 | 192 |
| EPM3064A | 4 | 30  62 | 64 | 64 | 192 |
| EPM3128A | 4 | 76  92 | 128 | 128 | 182 |
| EPM3256A | 4 | 112  154 | 256 | 256 | 156 |

Технология EEPROM обеспечивает сохранение конфигурации при отключении питания. Число логических эквивалентных вентилей ПЛИС находится в диапазоне 600-5000, количество программируемых пользователем выводов 44-208. Микросхемы могут быть запрограммированы с помощью программатора, в этом случае можно использовать все линии Ввода/Вывода (I /O). Кроме того, все ПЛИС имеют возможность внутрисистемного программирования (in-system programmability) через порт типа JTAG с использованием устройств типа BitBlaster, ByteBlaster и MasterBlaster, тогда 4 порта JTAG резервируются для этой цели. Выводы имеют возможность эмуляции режимов открытого коллектора и третьего (высокоимпедансного) состояния [2].

### 1. Составление схемы устройства

Входной сигнал, подаваемый на счетное устройство, представляет собой случайные трапецвидные импульсы разной амплитуды и длительности. Обнаружения импульсов производиться по амплитуде, для этого необходим компаратор. Сигнал с выхода компаратора подается на цифровую микросхему (ПЛИС), работающую с TTL уровнями, т.е. логический ноль 0.8...1.6 В логическая единица 1.65...2.0 В.

Для счета количества импульсов, поступаемых с компаратора, необходим счетчик. Счетчик планируется реализовать на ПЛИС.

Еще понадобиться индикатор, который будет отображать число этих импульсов. Целесообразно выбрать LCD индикатор со встроенным контроллером TTL логики, для облегчения работы, и напряжением питания +5В.

Генератор тактовой частоты необходимо выбрать исходя их следующих критериев: максимальная частота определяется скоростью работы ПЛИС, минимальная частота определяется минимальным интервалом между импульсами входного сигнала. Следовательно, интервал частот будет в пределах от 10 МГц до 150 МГц.

Источник питания: Проще всего решить проблему питания, взять готовый источник от персонального компьютера, но он имеет широкий диапазон выходных токов, с выходными напряжениями +5В, -5В, +12В, -12В. Для питания счетного устройства выберем напряжение +12В. В самом устройстве установим интегральные стабилизаторы для питания ПЛИС +3.3В, а для питания компаратора, генератора тактовой частоты и индикатора +5В.

Управление устройством осуществляется с помощью двух кнопок, следовательно, необходимо антидребезговое устройство, которое можно реализовать на ПЛИС.

На конец мне нужен программатор ByteBlaster для программирования (прошивки) ПЛИС.

Структурная схема изображена на рис. 1.1.

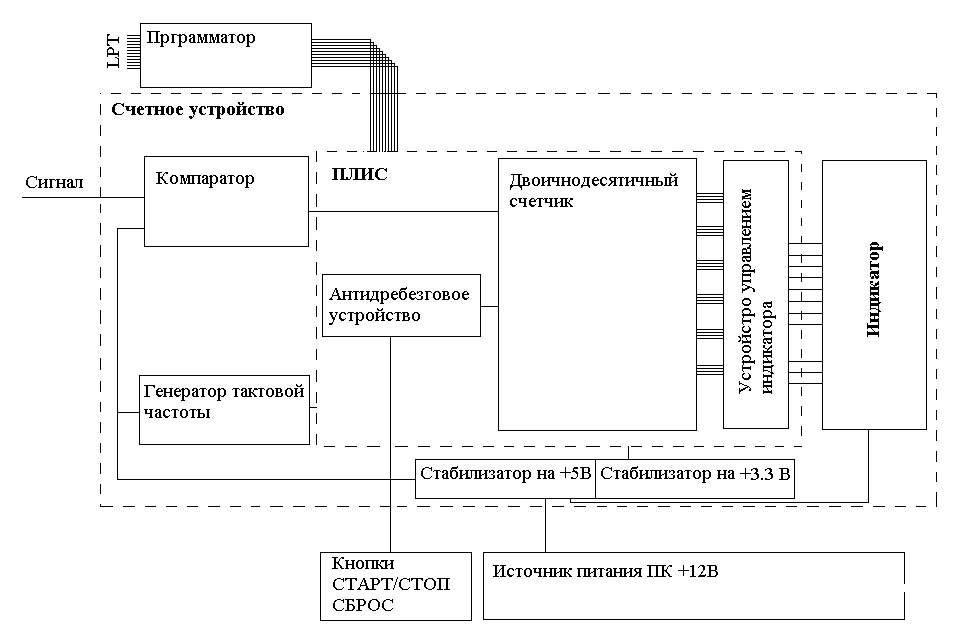


Рис 1.1 Структурная схема устройства

### 2. Выбор элементов

#### 

#### 2.1 Выбор ПЛИС. Описание внутренней структуры ПЛИС

Основными элементами структуры ПЛИС семейства MAX3000 являются:

* логические блоки (ЛБ, LAB, Logic Array Blocks);
* макроячейки (МЯ, macrocells);
* логические расширители (expanders), параллельный (parallel) и разделяемый (shareable);
* программируемая матрица соединений (ПМС, Programmable Interconnect Array, PIA);
* элементы ввода/вывода (ЭВВ, I/Ocontrol block).

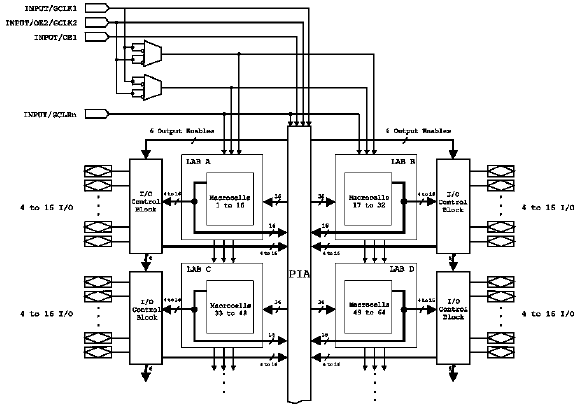


Рис 2.1.1 Функциональная схема ПЛИС семейства MAX3000

ПЛИС семейства MAX3000 имеют четыре вывода, закрепленных за глобальными цепями (dedicated inputs). Это глобальные цепи синхронизации сброса и установки в третье состояние каждой макроячейки. Кроме того, эти выводы можно использовать как входы или выходы пользователя для «быстрых» сигналов, обрабатываемых в ПЛИС.

Как видно из рис 2.1.1, в основе архитектуры ПЛИС семейства МАХ3000 лежат логические блоки, состоящие из 16 макроячеек каж­дый. Логические блоки соединя­ются с помощью программируемой матрицы соединений (ПМС). Каждый логи­ческий блок имеет 36 входов с ПМС.

На рис 2.1.2 приведена структурная схема макроячейки ПЛИС семейства MAX3000. Макроячейка ПЛИС семейства MAX3000 со­стоит из трех основных узлов:

* локальной программируемой матрицы (LAB local array);
* матрицы распределения термов(product-term select matrix);
* программируемого регистра (programmable register).

Комбинационные функции реализуются на локальной программируемой матрице и матрице распределения термов, позволяющей объединять логические произведения либо по ИЛИ (OR), либо по исключающему ИЛИ (XOR). Кроме того, матрица распределения термов позволяет скоммутировать цепи управления триггером макроячейки.

Режим тактирования и конфигу­рация триггера выбираются автоматически во время синтеза проекта в САПР Max+Plus II в зависимости от выбранного разработчиком типа триггера при описании проекта.

В ПЛИС семейства MAX 3000 до­ступно 2 глобальных тактовых сигнала, что позволяет проектировать схемы с двухфазной синхронизацией.

Для реализации логических функций большого числа переменных используются логические расширители.

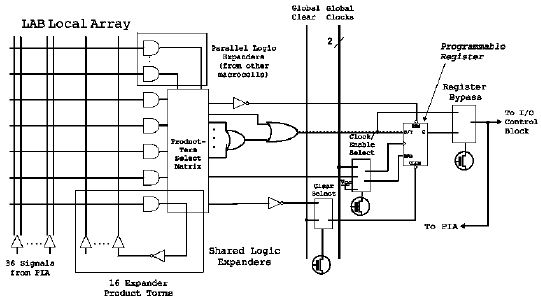


Рис 2.1.2 Структурная схема макроячейки ПЛИС

Разделяемый логический расширитель (рис.2.1.3) позволяет реализовать логическую функцию с большим числом входов, предоставляя возможность объединить макроячейки, входящие в состав одного логического блока. Таким образом, разделяемый расширитель формирует терм, инверсное значение которого передается матрицей распределения термов в локальную программируемую матрицу и может быть использовано в любой макроячейки данного логического блока. Как видно из рис.2.1.3, имеется 36 сигналов локальной ПМС, а также 16 инверсных сигналов с разделяемых логических расширителей, что позволяет в пределах одного логического блока реализовать функцию до 52 термов ранга 1.

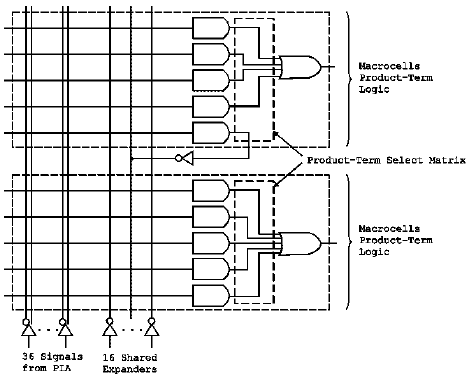


Рис 2.1.3 Разделяемый логический расширитель

Параллельный логический расширитель (рис. 2.1.4) позволяет использовать локальные матрицы смежных макроячеек для реализации функций, в которые входят более 5 термов. Одна цепочка параллельных расширителей может включать до 4 макроячейки, реализуя функцию 20 термов. Компилятор системы Max+Plus II поддерживает размещение до 3-х наборов не более 5 параллельных расширителей в каждом.

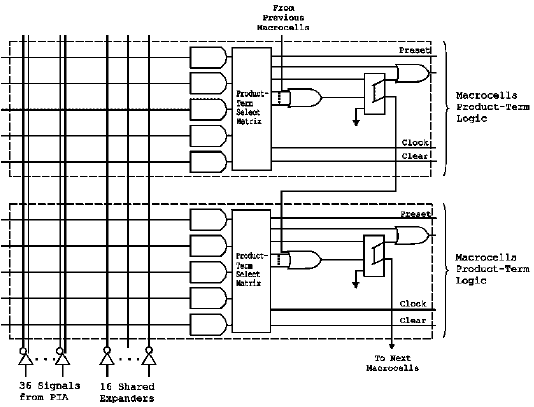


Рис 2.1.4 Параллельный логический расширитель

На рис. 2.1.5 приведена структура программируемой матрицы соединений.

На ПМС выводятся сигналы от всех возможных источников: элементов ввода-вывода, сигналов обратной связи логического блока, специализированных выделенных выводов. В процессе программирования только необходимые сигналы «заводятся» на каждый логический блок.

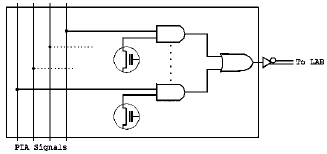


Рис 2.1.5 Структура ПМС ПЛИС семейства МАХ3000

На рис 2.1.6 приведена схема элемента ввода-вывода (ЭВВ) ПЛИС семейства MAX3000. ЭВВ позволяет организовать режимы работы с открытым коллектором и третьим состоянием.

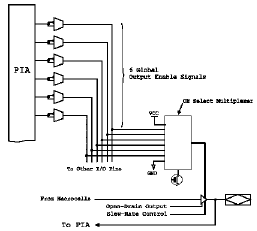


Рис 2.1.6 Элемент ввода вывода

Выберем микросхему ПЛИС семейства MAX3000 – EPM3265ACT144. Данная микросхема является недорогой, содержит достаточное количество макроячеек (256), удовлетворяет параметрам по быстродействию (максимум 156 МГц). Единственная сложность, микросхема имеет большое количество выводов при маленькой площади, что существенно усложняет ее монтаж [3][4].

#### 

#### 2.2 Компаратор

Компаратор сравнивает сигнал с опорным напряжением и оставляет только импульсы с заданной амплитудой. Выходной сигнал компаратора должен быть в виде TTL уровня. В соответствии с заданными характеристиками был выбран компаратор фирмы Analog Devices AD8561.

Рис 2.2.1 Расположение выводов компаратора



Таблица 2.2.1 Общее техническое описание V+ = +5.0 V, V– = VGND = 0 V, TA = +25 C

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Параметры | Обозначения | Условия эксплуатации | | min | max | Единицы измерения |
| Входная емкость | CIN |  | |  | 3 | пФ |
| Диапазон входных напряжений | VCM |  | | 0 | 3.0 | В |
| Длительность импульса | tPW(E) |  | |  | 6 | нс |
| время срабатывания | ts | |  |  | 1 | нс |
| время удержания | tH | |  |  | 1.2 | нс |
| Цифровой выход (по напряжению) | | | | | | |
| Логическая “1” | VOH |  | | 3.5 | V+ | В |
| Логический “0” | VOL |  | | 0.25 | 0.4 | В |
| Динамические характеристики | | | | | | |
| задержка распространения | tP | Продолжительность  200 mV с шагом 100 mV  –40C TA +85 | | 6.7 | 13 | нс |
| Питание | | | | | | |
| Ток источника питания(положительный) | I+ | –40C TA +85C | | 4.5 | 6.0 | мА |

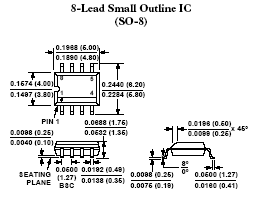


Рис 2.2.2(Физические параметры)

Перед компаратором необходимо поставить делитель, для обеспечения согласования по напряжению и сопротивлению [5].

#### 2.3 Генератор тактовой частоты

Генератор тактовой частоты необходим для задания временного масштаба счетного устройства на ПЛИС. По тактам генератора будет работать счетное устройство. Был выбран генератор фирмы Auch JCO series с высотой импульсов 5В для CMOS и TTL логики.[[1]](#footnote-1)\*

Таблица 2.3.1 Общее техническое описание:

|  |  |
| --- | --- |
| Частота | 20.0 МГц |
| Ток потребления | 25 мА |
| Стабильность частоты | ± 100 ppm (A-type)\* |
| Емкость нагрузки | 50 пФ или 10 TTL |
| Максимальное время спада и подъема | 6 нс |
| Уровень выходного сигнала максимально низкого (лог. «0») | 0.4 В |
| Уровень выходного сигнала минимально высокого(лог. «1») | VDC - 0.5 В |

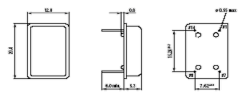
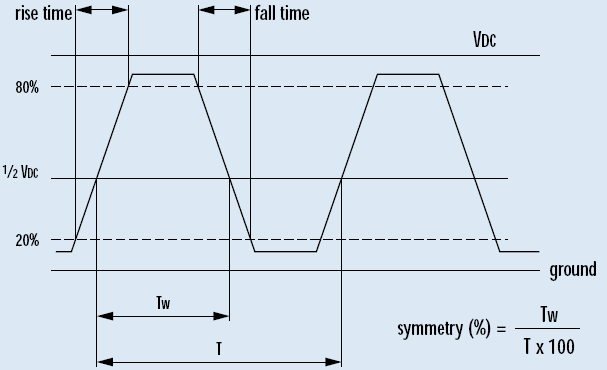


Рис 2.3.1 (Физические параметры)

Рис 2.3.2 (Форма выдаваемого сигнала)



#### 

#### 2.4 Индикатор

Был выбран двух строчный индикатор WH1602A с внутренней видео памятью и контролером HD44780.

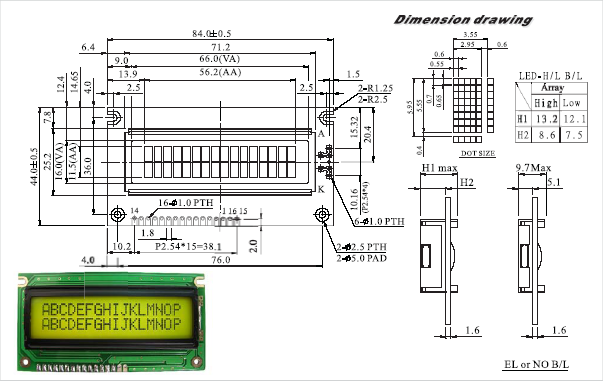


Рис 2.4.1(Внешний вид и габариты индикаторы)

#### 2.4.1 Описание контроллера HD44780

Алфавитно-цифровые ЖКИ-модули представляют собой недорогое и удобное решение, позволяющее сэкономить время и ресурсы при разработке новых изделий, при этом обеспечивают отображение большого объема информации при хорошей различимости и низком энергопотреблении. Возможность оснащения ЖКИ-модулей задней подсветкой позволяет эксплуатировать их в условиях с пониженной или нулевой освещенностью, а исполнение с расширенным диапазоном температур (-20°С...+70°С) в сложных эксплуатационных условиях, в том числе в переносной, полевой и даже, иногда, в бортовой аппаратуре.

Контроллер HD44780 потенциально может управлять 2-мя строками по 40 символов в каждой (для модулей с 4-мя строками по 40 символов используются два однотипных контроллера), при матрице символа 5 х 7 точек.[6]

Изучая каталоги различных фирм-производителей ЖКИ-модулей, можно убедиться, что одни форматы и конструктивы являются собственными разработками и не обнаруживают аналогов в номенклатуре остальных фирм, другие являются фактическими стандартами и производятся большинством изготовителей. В качестве примера можно назвать ЖКИ-модуль формата 24 х 2, именуемый PC2402-A у Powertip, ED24200 у EDT, DMC-24227 у Optrex, SC2402A у Bolymin, MDLS-24265 у Varitronix, PVC240202 у Picvue и др., все эти модули имеют одинаковые конструктивные размеры и являются взаимозаменяемыми.

В рамках одного конструктива ЖКИ-модуль может иметь еще ряд модификаций. В частности, могут применяться несколько типов ЖКИ, отличающихся цветом фона и цветом символов, а также по применяемым ЖК-материалам и структуре: TN, STN и FSTN типа. ЖКИ STN и FSTN типа имеют более высокую стоимость, но одновременно обладают повышенной контрастностью и вдвое большим максимальным углом обзора, причем ЖКИ FSTN типа имеют лучшие характеристики, чем STN.

ЖКИ-модули могут оснащаться задней подсветкой, размещаемой между ЖКИ и печатной платой, для чего ЖКИ производятся с полупрозрачным или прозрачным задним слоем (в последнем случае считывание информации возможно только при наличии подсветки). Собственно подсветка может быть реализована несколькими способами: с помощью электролюминесцентной панели, представляющей собой тонкую пленку, излучающую свет при прикладывании переменного тока повышенного напряжения порядка 100...150 В; люминисцетной лампой с холодным катодам (также работающей при повышенном напряжении), излучение которой равномерно распределяется по всей площади ЖКИ с помощью отражателя или плоского счетовода; третий вариант - подсветка на основе светодиодной матрицы.

Первые два способа подсветки обеспечивают высокую яркость и могут иметь белый тон свечения при относительно низком потреблении, но требуют наличия источника повышенного напряжения, что создает некоторые трудности при создании аппаратуры с автономным питанием. Напротив, светодиодная подсветка не требует высоковольтного источника (прямое падение напряжения составляет 4,2 В) и при использовании несложного источника тока позволит производить питание от источника с напряжением 5 В диапазоне температур (-20°С...+70°С).[7]

#### 

#### 2.4.2 Подключение ЖКИ-модуля

Для соединения ЖКИ-модуля с управляющей системой используется параллельная синхронная шина, насчитывающая 8 или 4 (выбирается программно) линий данных DB0...DB7, линию выбора операции R/W, линию выбора регистра RS и линию стробирования/синхронизации Е. Кроме линий управляющей шины имеются две линии для подачи напряжения питания 5 В - GND и VCC, и линия для подачи напряжения питания драйвера ЖКИ - V0.

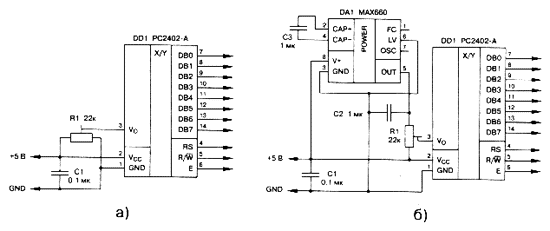


Рис 2.4.2. Подача питания на ЖКИ-модуль

На начальном этапе необходимо подать питание на ЖКИ-модуль и добиться от него признаков работоспособности. Схема включения модуля, рассчитанного на стандартный диапазон температур, показана на рис.2.4.2а. Подстроечный резистор R1 позволяет плавно менять напряжение питания драйвера ЖКИ, что приводит к изменению угла поворота жидких кристаллов. Этим резистором можно отрегулировать фактическую контрастность при некотором преимущественном угле наблюдения (снизу-вверх или сверху-вниз). Включение в данную схему ЖКИ-модуля, рассчитанного на расширенный диапазон температур, не приведет к успеху, так как из-за особенностей применяемых в них ЖК-материалов, эти ЖКИ требуют повышенного напряжения питания и при питании напряжением 5 В изображение либо будет отсутствовать совсем, либо будет слабоконтрастным. Для преодоления ситуации необходимо подать на вывод V0 отрицательное напряжение (напряжение на ЖКИ определяется разностью VCC и V0), составляющее в предельном случае - 5 В. Если в схеме отсутствует источник отрицательного напряжения, то не составляет труда собрать простейший преобразователь, например, по схеме на рис 2.4.2б.

Для активизации ЖКИ-модуля подается напряжение питания и регулируется движок резистора R1 (рис. 2.4.2). После окончания цикла внутренней инициализации модуль включается в режим развертки одной верхней строки. При изменении напряжения на выводе V0 сегменты этой строки должны менять свое состояние от прозрачного до непрозрачного, что является свидетельством правильного подключения питания модуля и работоспособности контроллера и драйверов ЖКИ. Движок устанавливается в такое положение, при котором изображение сегментов в верхней строке едва проступает на основном фоне ЖКИ. Теперь ЖКИ-модуль готов к приему и отображению информации. После того, как появится отображения на индикаторе какого-либо текста, можно более точно отрегулировать изображение в соответствии с необходимой контрастностью и требуемым углом наблюдения.

Для соединения модуля с управляющей системой можно выбрать один из двух вариантов: по 8-ми или 4-х разрядной шине. В первом случае потребуется 11 сигнальных линий, во втором - только 7. Сам процесс обмена также может быть организован двояко.

Можно подключить ЖКИ-модуль к системной шине (если таковая имеется) и выполнить обмен в синхронном режиме с максимальной скоростью. Но это может потребовать введения дополнительных схем для замедления скорости работы шины при выполнении операций обмена с ЖКИ-модулем, и подключение к системной шине в большинстве случаев потребует вводить схемы дешифрации и формирования сигналов Е и R/W, что опять приведет к дополнительным затратам. Все сказанное выше не означает, что вариант с подключением к шине принципиально неэффективен. В какой-то конкретной системе этот способ, наоборот, может быть самым оптимальным. Кроме того, некоторые современные процессоры, например, семейство МС68НС12, имеет встроенные средства для формирования сигналов выборки (CS), с возможностью программно определить скорость обмена с каждым конкретным устройством.

Другой вариант обмен с ЖКИ-модулем выполняется чисто программными средствами, через порты ввода-вывода. Вариант соединения с системной шиной, напротив, требует рассмотрения конкретных устройств, поэтому необходимо изучить временные диаграммы операций чтения и записи, приведенные на рис 2.4.4 и 2.4.5, а также значения временных параметров, приведенные в табл.2.4.5, 6 и сконструировать соответствующие управляющие схемы.

В данной работе используется именно этот вариант.

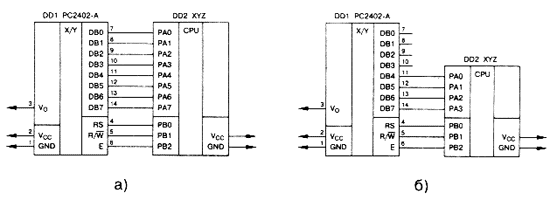


Рис 2.4.3 Подключение к управляющей системе

В соответствии с временной диаграммой (рис.2.4.4, 5 ) в исходном состоянии сигнал Е = 0, сигнал R/W = 0, значение сигнала RS - произвольное, шина данных DBO...DB7 в состоянии высокого импеданса (НI). Такое состояние управляющих сигналов (E и R/W) должно поддерживаться все время в промежутках между операциями обмена с ЖКИ-модулем. Шина данных в эти моменты в принципе свободна, и может использоваться в мультиплексном режиме для каких-либо других целей, например, для сканирования матрицы клавиатуры. Естественно, необходимо позаботиться об исключении конфликтов на шине данных в момент совершения операций обмена с ЖКИ-модулем.[8]

Последовательности действий, которые необходимо выполнять управляющей системе при совершении операций записи и чтения для 8-ми и 4-х разрядной шины приведены соответственно в табл 2.4.1...4.

Таблица 2.4.1 Операции записи для 8-ми разрядной шины

|  |
| --- |
| 1. Установить значение линии RS |
| 1. Вывести значение байта данных на линии шины DB0...DB7 |
| 1. Установить линию Е = 1 |
| 1. Установить линию E = 0 2. Установить линии шины DB0...DB7 = HI |

Таблица 2.4.2 Операции чтения для 8-ми разрядной шины

|  |
| --- |
| 1. Установить значение линии RS |
| 1. Установить линию R/W = 1 |
| 1. Установить линию Е = 1 |
| 1. Считать значение байта данных с линий шины DB0...DB7 |
| 1. Установить линию Е = 0 |
| 1. Установить линию R/W = 0 |

Таблица 2.4.3 Операция записи для 4-х разрядной шины

|  |
| --- |
| 1. Установить значение линии RS |
| 1. Вывести значение старшей тетрады байта данных на линии шины DB4...DB7 |
| 1. Установить линию Е = 1 |
| 1. Установить линию Е = 0 |
| 1. Вывести значение младшей тетрады байта данных на линии шины DB4...DB7 |
| 1. Установить линию Е = 1 |
| 1. Установить линию Е = 0 |
| 1. Установить линии шины DB4...DB7 = HI |

Таблица 2.4.4 Операция чтения для 4-х разрядной шины

|  |
| --- |
| 1. Установить значение линии RS |
| 1. Установить линию R/W = 1 2. Установить линию Е = 1 |
| 1. Считать значение старшей тетрады байта данных с линий шины DB4...DB7 |
| 1. Установить линию Е = 0 |
| 1. Установить линию Е = 1 |
| 1. Считать значение младшей тетрады байта данных с линий шины DB4...DB7 |
| 1. Установить линию Е = 0 |
| 1. Установить линию R/W = 0 |

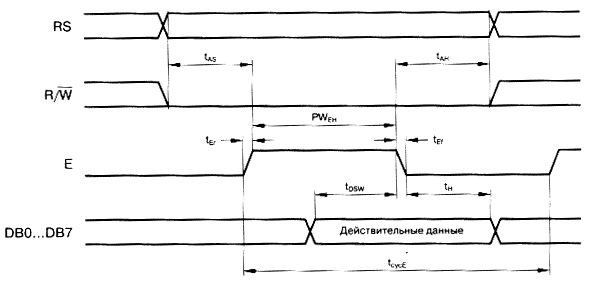


Рис 2.4.4. Временная диаграмма операции записи

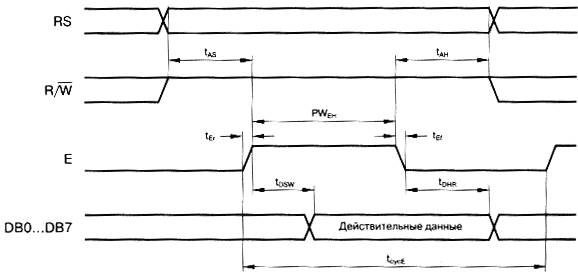


Рис 2.4.5 Временная диаграмма операции чтения

Приведенные в табл 2.4.1...4 операции подразумевают, что время выполнения каждого шага составляет не менее 250 нс. Однако это условие может быть легко нарушено, поэтому необходимо тщательно контролировать минимальные значения временных интервалов, чтобы они всегда находились в области допустимых значений, указанных в таблице 2.4.5, 6 и при необходимости вводить задержки.

Таблица 2.4.5 Значения временных характеристик. Операция записи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Обозначение | Мин. | Макс. | Единица |
| Период сигнала Е | tcycE | 500 | - | нс |
| Положительный полупериод сигнала Е | PWEH | 230 | - | нс |
| Фронт/спад сигнала Е | tEr, tEf | - | 20 | нс |
| Установление адреса | tAS | 40 | - | нс |
| Удержание адреса | tAH | 10 | - | нс |
| Установление данных | tDSW | 80 | - | нс |
| Удержание данных | tDSW | 10 | - | нс |

Таблица 2.4.6 Значения временных характеристик. Операция чтения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Обозначение | Мин. | Макс. | Единица |
| Период сигнала Е | tcycE | 500 | - | нс |
| Положительный полупериод сигнала Е | PWEH | 230 | - | нс |
| Фронт/спад сигнала Е | tEr, tEf | - | 20 | нс |
| Установление адреса | tAS | 40 | - | нс |
| Удержание адреса | tAH | 10 | - | нс |
| Установление данных | tDSW | - | 160 | нс |
| Удержание данных | tDSW | 5 | - | нс |

Описанные выше операции записи/чтения байта являются базовыми для осуществления обмена с ЖКИ-модулем. Реализация этих двух операций - единственное, что отличает процесс обмена по 8-ми разрядной шине от обмена по 4-х разрядной шине. На основе этих двух операций, реализованных программно (когда модуль подключен к портам Ввода/Вывода какого нибудь устройства), или аппаратно (когда модуль подключен к системной шине), строятся все виды операций программирования и управления.

#### 2.4.3 Программирование и управление ЖКИ-модуля:

Управление контроллером ведется посредством интерфейса управляющей системы. Основными объектами взаимодействия являются регистры DR и IR. Выбор адресуемого регистра производится линией RS, если RS = 0 - адресуется регистр команд (IR), если RS = 1 - регистр данных (DR).

Данные через регистр DR, в зависимости от текущего режима, могут помещаться (или прочитываться) в видеопамять (DDRAM) или в ОЗУ знакогенератора (CGRAM) по текущему адресу, указываемому счетчиком адреса (АС). Информация, попадающая в регистр IR, интерпретируется устройством выполнения команд как управляющая последовательность. Прочтение регистра IR возвращает в 7-ми младших разрядах текущее значение счетчика АС, а в старшем разряде флаг занятости (BF).

У контроллера HD44780 существует набор внутренних флагов, определяющих режимы работы различных элементов контроллера (таблица 2.4.7). В таблице 2.4.8 приведены значения управляющих флагов непосредственно после подачи на ЖКИ-модуль напряжения питания. Переопределение значений флагов производится специальными командами, записываемыми в регистр IR, при этом комбинации старших битов определяют группу флагов или команду, а младшие содержат собственно флаги.

Таблица 2.4.7 Флаги, управляющие работой контроллера HD44780

|  |  |
| --- | --- |
| /D: | режим смещения счетчика адреса АС, 0 - уменьшение, 1 - увеличение. |
| S: | флаг режима сдвига содержимого экрана. 0 - сдвиг экрана не производится, 1 - после записи в DDRAM очередного кода экран сдвигается в направлении, определяемым флагом I/D: 0 - вправо, 1 - влево. При сдвиге не производится изменение содержимого DDRAM. изменяются только внутренние указатели расположения видимого начала строки в DDRAM. |
| S/C: | флаг-команда, производящая вместе с флагом R/L операцию сдвига содержимого экрана (так же, как и в предыдущем случае, без изменений в DDRAM) или курсора. Определяет объект смещения: 0 - сдвигается курсор, 1 - сдвигается экран. |
| R/L: | флаг-команда, производящая вместе с флагом S/C операцию сдвига экрана или курсора. Уточняет направление сдвига: 0 - влево, 1 - вправо. |
| D/L: | флаг, определяющий ширину шины данных: 0 - 4 разряда, 1 - 8 разрядов. |
| N: | режим развертки изображения на ЖКИ: 0 - одна строка, 1 - две строки |
| F: | размер матрицы символов: 0 - 5 х 8 точек, 1 - 5 х 10 точек. |
| D: | наличие изображения: 0 - выключено, 1 - включено |
| С: | курсор в виде подчерка: 0 - выключен, 1 - включен |
| В: | курсор в виде мерцающего знакоместа: 0 - выключен, 1 - включен. |

Таблица 2.4.8 Значения управляющих флагов после подачи питания

|  |  |
| --- | --- |
| I/D = 1: | режим увеличения счетчика на 1 |
| S = 0: | без сдвига изображения |
| D/L = 1: | 8-ми разрядная шина данных |
| N = 0: | режим развертки одной строки |
| F = 0: | символы с матрицей 5 х 8 точек |
| D = 0: | отображение выключено |
| С = 0: | курсор в виде подчерка выключен |
| В = 0: | курсор в виде мерцающего знакоместа выключен |

Список управляющих комбинаций битов регистра IR и выполняемые ими команды приведены в таблице 2.4.9. Так как на момент включения ЖКИ-модуль ничего не отображает (флаг D = 0), то для того, чтобы вывести какой-либо текст необходимо, как минимум, включить отображение, установив флаг D = 1. Вот пример широко распространенной последовательности для инициализации ЖКИ-модуля: , $OC, 6 (знак "$" перед числом указывает на шестнадцатеричное основание) устанавливает режим отображения 2-х строк с матрицей 5 х 8 точек и работу с 8-ми разрядной шиной данных; $OC включает отображение на экране ЖКИ-модуля, без отображения курсоров; 6 устанавливает режим автоматического перемещения курсора слева-направо после вывода каждого символа.

Таблица 2.4.9. Управляющие комбинации битов регистра IR

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D1 | D2 | D3 | D4 | D5 | | D6 | D7 | D8 | Назначение |
| 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | Очистка экрана, АС = 0, адресация АС на DDRAM |
| 0 | 0 | 0 | 0 | 0 | | 0 | 1 | - | АС = 0, адресация на DDRAM, сброшены сдвиги, начало строки адресуется в начале DDRAM |
| 0 | 0 | 0 | 0 | 0 | | 1 | I/D | S | Выбирается направление сдвига курсора или экрана |
| 0 | 0 | 0 | 0 | 1 | | D | C | B | Выбирается режим отображения |
| 0 | 0 | 0 | 1 | S/C | | R/L | - | - | Команда сдвига курсора/экрана |
| 0 | 0 | 1 | DL | | N | F | - | - | Определение параметров развертки и ширины шины данных |
| 0 | 1 | AG | AG | | AG | AG | AG | AG | Присвоение счетчику АС адреса в области CGRAM |
| 1 | AG | AG | AG | | AG | AG | AG | AG | Присвоение счетчику АС адреса в области DDRAM |

Контроллер HD44780 поддерживает как операции записи, так и операции чтения. Чтение регистра DR приводит к загрузке содержимого DDRAM или CGRAM, в зависимости от текущего режима, при этом курсор смещается на одну позицию, как и при записи. Чтение регистра IR возвращает 8 значащих разрядов, причем в 7-ми младших содержится текущее значение счетчика АС (7 разрядов, если адресуется DDRAM, и 6 - если CGRAM), а в старшем - флаг занятости BF. Этот флаг имеет значение 1 когда контроллер занят и 0 - когда свободен. Необходимо учитывать, что большинство операций, выполняемых контроллером, занимают значительное время, около 40 мкс, а время выполнения некоторых доходит до единиц миллисекунд, поэтому цикл ожидания снятия флага BF должен обязательно присутствовать в программах драйвера ЖКИ-модуля и предшествовать совершению любой операции (естественно, кроме операции проверки флага BF).

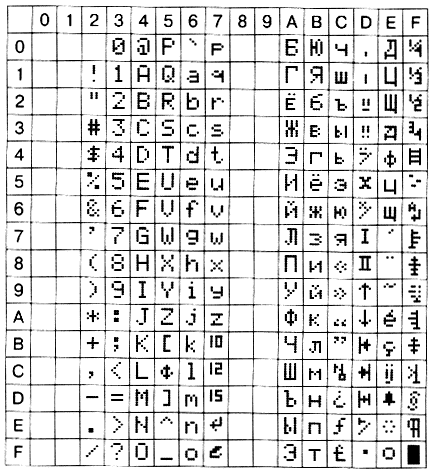
Один важный момент! После совершения операции записи или чтения DDRAM и появления после нее признака готовности (BF = 0), прочитанное в этом же цикле (вместе с флагом BF) значение АС, скорее всего не будет достоверным. Дело в том, что между появлением признака готовности и вычислением контроллером нового значения АС существует некоторый временной интервал, составляющий около 4 мкс при тактовой частоте контроллера 270 кГц. Поэтому, если необходимо получить истинное значение АС, нужно совершить повторную операцию прочтения IR спустя не менее чем 4 мкс (если контроллер работает на частоте 270 кГц время ожидания необходимо пропорционально увеличить).

Вывод на экран символа производится записью его кода в регистр DR. При этом символ размещается в DDRAM по текущему адресу, указываемому АС, а значение АС увеличивается или уменьшается на 1. Чтобы произвести переустановку курсора на нужную позицию, необходимо присвоить АС соответствующее значение (см. таблицу 2.4.9). Здесь есть одна тонкость. Когда производится последовательная запись символов и в результате заполняется вся строка, курсор автоматически переходит на вторую строку, но если необходимо принудительно установить курсор, скажем, на начало второй строки, то будет неверным присвоить АС казалось бы логичное значение (40), правильным является значение (64). Значения адресов DDRAM в диапазоне ...fF (а равно и ...F) являются неопределенными и результаты работы с ними могут быть непредсказуемыми.

Необходимо учитывать, что контроллеры, устанавливаемые на ЖКИ-модули, могут иметь различные наборы символов, причем это может зависеть как от производителя контроллера, так и от модификации данной конкретной модели. Например, фирма Powertip выпускает ЖКИ-модули с четырьмя базовыми модификациями наборов символов: японской, европейской, французской и русской.

Более того, существует как минимум два варианта русского набора символов: контроллер фирмы Hitachi (H2 по маркировке фирмы Powertip) и контроллер фирмы Epson (EH по маркировке Powertip). Контроллер фирмы Hitachi обладает существенным недостатком - у него весьма ограниченный набор русских символов, фактически у него имеются только прописные русские буквы, и даже среди них отсутствует символ "Ф". Напротив, контроллер фирмы Epson содержит полный набор русских символов в прописном и строчном вариантах, поэтому он весьма удобен для отечественных применений. Это свойство контроллеров фирмы Epson обеспечило им заслуженную популярность на российском рынке, поэтому в последнее время основная масса импортируемых в нашу страну ЖКИ-модулей оснащены именно этим контроллером; в качестве примера в таблице 2.4.10 приведен набор символов этого контролера.

Таблица 2.4.10.Таблица кодов символов контроллера Epson набор "Russian"



Из допустимых для размещения в DDRAM кодов символы с кодами <--DESCRIPTION-->0...7 (и их дубликат с кодами 8...<--DESCRIPTION-->F) имеют специальное назначение - это переопределяемые символы, графическое изображение которых может назначить сам потребитель, разместив соответствующую информацию в области CGRAM. Для программирования доступны 8 переопределяемых символов в режиме с матрицей 5 х 7 точек и 4 с матрицей 5 х 10 (в режиме 5 х 10 переопределяемые символы адресуются кодами DDRAM через один: <--DESCRIPTION-->0, 2, 4, 6). Для кодирования матрицы используются горизонтально "уложенные" байты, пять младших битов которых несут информацию о рисунке (причем 1 означает, что сегмент будет включен), 4-й разряд каждого из 8-ми (или 11-ти в режиме 5 х 10) байтов матрицы определяет левую колонку символа, а 0-й - правую. Старшие три бита не используются, равно как и старшие пять байтов, составляющих полную область матрицы символа (16 байтов) в режиме 5 х 10 (обратите внимание, что матрица программируемых символов допускает использование полной высоты строки (8 строчек для режима 5 х 7 и 11 строчек для режима 5 х 10), то есть можно размещать точки в области подчеркивающего курсора).

Чтобы определить собственный символ необходимо установить счетчик АС на адрес начала матрицы требуемого символа в CGRAM - <--DESCRIPTION-->0, 8, и т.д. (<--DESCRIPTION-->0, , для режима 5 х 10 точек) - произвести перезапись всех байтов матрицы, начиная с верхней строки. После этого, записав в DDRAM код запрограммированного символа: <--DESCRIPTION-->0, 1, 2 (<--DESCRIPTION-->0, 2, 4 для режима 5 х 10 точек), на экране в соответствующем месте будет отображаться переопределенный символ.

Несколько слов о процессе инициализации ЖКИ-модуля. Производитель контроллера рекомендует выполнять следующую последовательность действий для инициализации. Выдержать паузу не менее 15 мс между установлением рабочего напряжения питания (> 4,5 В) и выполнением каких-либо операций с контроллером. Первой операцией выполнить команду, выбирающую разрядность шины (это должна быть команда независимо от того, какой разрядности интерфейс вы собираетесь использовать в дальнейшем), причем перед выполнением этой операции не проверять значение флага BF. Далее опять выдержать паузу не менее 4,1 мс и повторить команду выбора разрядности шины, причем перед подачей команды вновь не производить проверку флага BF. Следующим шагом необходимо вновь выдержать паузу, на этот раз 100 мкс, и в третий раз повторить команду установления разрядности шины, вновь без проверки BF. Эти три операции являются инициализирующими и призваны вывести контроллер в исходный режим работы (то есть перевести в режим работы с 8-ми разрядной шиной) из любого состояния. Следом за ними нормальным порядком (без выдерживания пауз, но с проверкой флага BF) выполняется инициализация режимов работы с выдачей инициализирующей последовательности, аналогичной указанной в таблице 2.4.7 (содержащей в том числе команду выбора необходимой разрядности шины).

Необходимо помнить, что когда Вы объявляете режим работы с 4-х разрядной шиной, то есть выдаете команду , то делаете это обычно из 8-ми разрядного режима, который устанавливается автоматически после подачи напряжения питания, а значит вы не сможете адекватно объявить необходимое значение флагов N и F, располагающихся в младшей тетраде команды установки разрядности шины. Поэтому команду необходимо повторить в уже установившемся 4-х разрядном режиме путем последовательной передачи двух тетрад, то есть для 4-х разрядного режима образом.[9]

#### 

#### 2.5 Стабилизаторы

Для питания ПЛИС нужно +3.3В, компаратора, индикатора и генератора +5В. Для питания всего устройства взят источник питания от персонального компьютера, который выдает +12 В. Было выбрано два интегральных стабилизатора.

Первый на +5В фирмы Fairchild Semiconductor MC78T05

Таблица 2.5.1 Технические характеристики

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр | Обозначение | Характеристики | Min | Max | Единицы измерения |
| Выходное напряжение | V0 | 5мА ≤ Io ≤ 3.0A , TJ = +25° | 4.8 | 5.2 | В |
| Отклонение напряжения по нагрузке | ΔV0 | 5мA ≤ Io ≤ 3.0A , TJ = +25°C | 10 | 30 | мВ |
| Максимальный выходной ток | IPK | TA = +25°C | 5.0 | - | A |
| Выходное сопротивление | R0 | f = 1.0kHz | 2.0 | - | мОм |
| Ток короткого замыкание | Isc | VI = 35V, TJ =+25°C | 1.5 | 2.5 | A |
| Входное напряжение (для V0 = 5V) | V1 | TJ =+25°C | 35 | 40 | В |
| Рассеваемая мощность | W |  |  | 25 | Вт |

* Внутреннее короткое замыкание тока исключено
* Внутренняя защита от тепловой перегрузки [10]

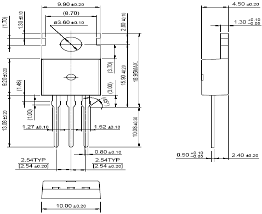


Рис 2.5.2 (Физические параметры стабилизатора MC78T05)

Второй на +3.3 В фирмы National Semiconductor LMS1585A

Основные характеристики:

* Интервал выходных напряжений……………..1.5V и 3.3V
* Не стабильность тока в сети…………………………………0.005%
* Не стабильность тока на нагрузке……………………………0.05%

Таблица 2.4.3 Технические характеристики:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр | Обозначение | Условие | min | max | Един. измер. |
| Выходное напряжение | Vout | IOUT = 0mA, VIN = 5V  0 IOUT 5A, 4.75VVIN7V | 3.267  3.235 | 3.333  3.365 | В  В |
| Отклонение выходного напряжения по нагрузке | ΔVout | VIN = 5V, 0 IOUT 5A | 0.05 | 0.5 | % |
| Рабочий ток |  | VIN = 5V | 7.0 | 13.0 | мА |

* Максимальное входное и выходное напряжение (относительно земли)…………………………………………….…………………………….13V
* Электростатический допуск………………………………….2000V

[11]

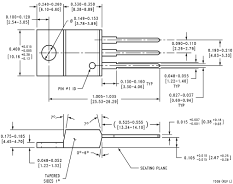


Рис 2.4.3 (Физические параметры стабилизатора LMS1585A)

#### 2.5 Программатор ByteBlaster

Устройство для программирования ПЛИС. На вход подается шина с LPT порта а на выходе 7-и разрядная шина из которой 4 провода на порт JTAG, 2 на питание +5 В и 2 на землю смотри приложение 1( схема устройства)

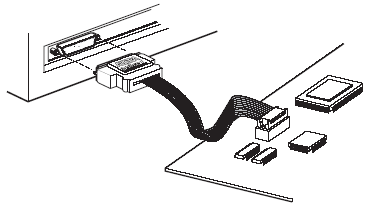


Рис 2.5.1.(Подключение ПЛИС к Компьютеру через LPT порт с помощью Byte Blaster)

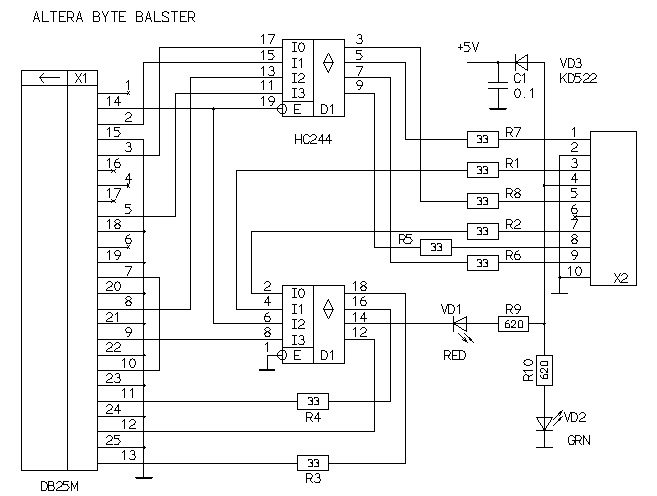


Рис 2.5.2(Схема ByteBlaster)

Как видно из схемы Байтбластер содержит одну микросхему НС244 которая содержит в себе всего лишь 4 буфера и резисторы. Это значит она очень дешевая и простая, её можно изготовить в домашних условиях.[12]

#### 

#### 2.6 Сборка устройства

Как уже ранее говорилось, ПЛИС была выбрана EPM3256ATC144-10 семейства MAX3000, для прошивки к ней подключаются программатор (Byte Blaster). Также в устройство входит компаратор (AD8561), генератор тактовой частоты (Auch JCO series), кнопки (СТАРТ/СТОП, СБРОС), источник питания ПК, стабилизаторы напряжения (MC78T05, LMS1585A) и индикатор (WH1602A). Принципиальная схема приведена в приложение 1. В приложение 2 приведен список выводов ПЛИС. Подключение компонентов схемы ведется согласно тех. документации прилагаемое с микросхемой.

### 3. Конфигурирование ПЛИС

#### 

#### 3.1 Система проектирования MAX+plus II

В качестве средства автоматизации проектирования устройств на основе собственных микросхем фирма Altera разработала систему MAX+plus II . Эта система состоит из 11 программ-обработчиков проекта (приложений), которые функционируют под управлением программы Manager. Система является полноцикловой, т.е. поддерживает не отдельные этапы проектирования (как система типа CAD - computer aided design), а сквозной процесс от ввода и контроля описания до программирования микросхемы ПЛИС. Такие системы получили название EDA - electronic design automation.  
Проект в системе MAX+plus II может быть иерархическим и состоять из набора модулей. Каждый модуль содержит описание части проекта, форма проекта может быть графическое представление принципиальной схемы, текст на языках AHDL/ VHDL/ Verilog, логико-временные диаграммы функционирования.

Для создания исходных модулей в состав приложений включены редакторы: графический – Graphic Editor, текстовый – Text Editor, логико-временных диаграмм Waveform Editor. Любой файл(модуль) может быть представлен в графическом виде как символ. Это можно сделать явно с использования редактора символов – Symbol Editor. Далее эти символы могут быть использованы при создании иерархического проекта. Пользователю обширные библиотеки примитивов и макрофункций.[2]

#### 

#### 3.2 Описание программы конфигурации ПЛИС

Данная программа состоит из нескольких модулей. Каждый модуль выполняет определённую функцию. Ниже приводиться структура программы конфигурации.

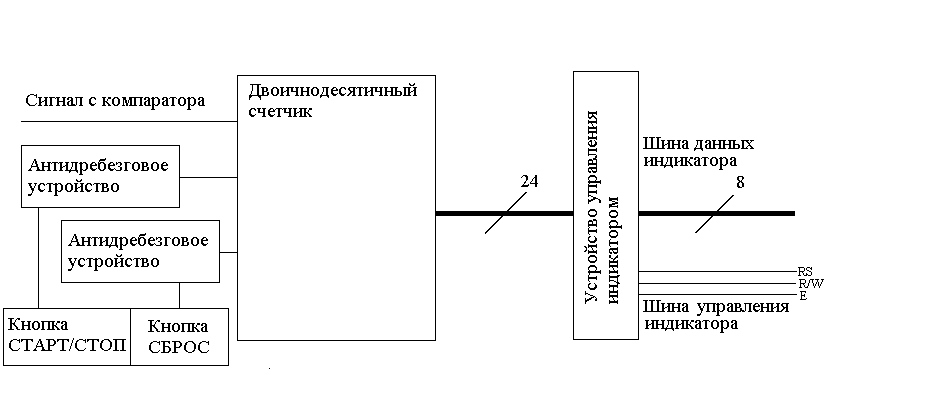


Рис 3.2 Структура программы

#### 

#### 3.2.1 Антидребезговая система (antibounce)

Как известно при нажатии кнопок, при контакте возникает дребезг, и вместо нужного простого перехода из нуля в единицу одной ступенькой возникает сигнал в несколько скачков как показано на рисунке 3.2.1

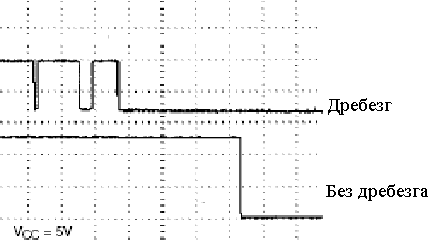


Рис 3.2.1 Сигнал включения с дребезгом и без дребезга

Антидребезговая защита была реализована в модуле antibounce. Принцип его работы, основан на счетчике и одном триггере защелке. При возникновении дребезга счетчик начинает считать до определенной константы в программе она названа delay т.е. задержка и если по истечении счета на входе по прежнему стоит единица то триггер защелка перещелкивается и возникает переключение, смотри рис 3.2.2.

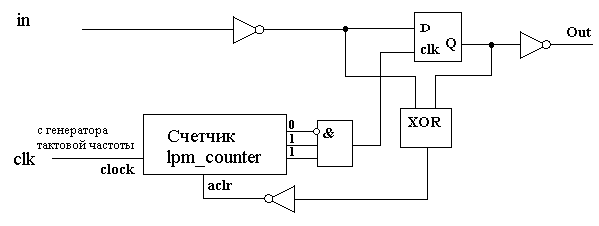


Рис 3.2.2 Схема антидребезгового устройства

Счетчик реализован с посредством макрофункции Lpm\_counter. Встроенная в среду макрофункция Lpm\_counter, есть ни что иное, как счетчик с задаваемыми пределами и параметрами, созданная для облегчения процесса разработки подобных устройств.

#### 

#### 3.2.1,1 Встроенная макрофункция – счетчик lpm\_counter

Двоичный счетчик с возможностью прямого и обратного счета и синхронного и асинхронного сброса. Lpm\_counter описывается как переменная в разделе VARIABLE. Форма описания счетчика следующая:

Имя\_счетчика : lpm\_counter WITH STATES (Параметры счетчика);

Например

COUNTER : lpm\_counter WITH (lpm\_width=4);

Таблица 3.2.1 Параметры счетчика

|  |  |  |  |
| --- | --- | --- | --- |
| Параметры счетчика | Тип | Обязательный | Описание |
| LPM\_WIDTH | Целочисленный | Да | Ширина счетчика(число битов на выходе) |
| LPM\_DIRECTION | Строковый | Нет | Устанавливается в прямом “UP” или обратном “DOWN” направлении идет счет, по умолчанию стоит прямой. |
| LPM\_MODULUS | Целочисленный | Нет | Установка максимума счета, если параметр не усыновлен то счет не ограничен. |
| LPM\_AVALUE | Целочисленный/Строковый | Нет | Ограничивает максимум счета 32 битами во избежания перегрузки счетчика, когда поведение счетчика не определено при переполнении |
| LPM\_SVALUE | Целочисленный/Строковый | Нет | Постоянное значение означает что нагрузка включается по переднему фронту clock когда sset или sconst в “1”. Этот параметр должен быть использован если sconst используется |
| LPM\_HINT | Строковый | Нет | Позволяет с большой точностью перевести текстовый файл из AHDL в VHDL. По умолчанию не используется. |
| LPM\_TYPE | Строковый | Нет | Связывает LPM\_имя с VHDL графическим файлом |
| CARRY\_CNT\_EN | Строковый | Нет | Принимает значения “SMART”, “ON”, “OFF”. При включении lpm\_counter функция cnt\_en передает сигнал через несущую цепочку. В некоторых случаях, этот параметр используется как параметр установки появления изменения в скорости, но при желании это можно выключить. По умолчанию стоит “SMART”, который сохраняет лучшие отношение между скоростью и размером. |
| LABWIDE\_SCLR | Строковый | Нет | Принимает значения “ON”,”OFF” или не используется. По умолчанию стоит “NO”. Он позволяет выключить использование LAB-ширины sclr особенно при использовании устройств на базе FLEX 6000. Уменьшает использование случайных библиотек, и таким образом даёт возможность использовать высшей логики плотнее, когда SCLR не используется в LAB. |

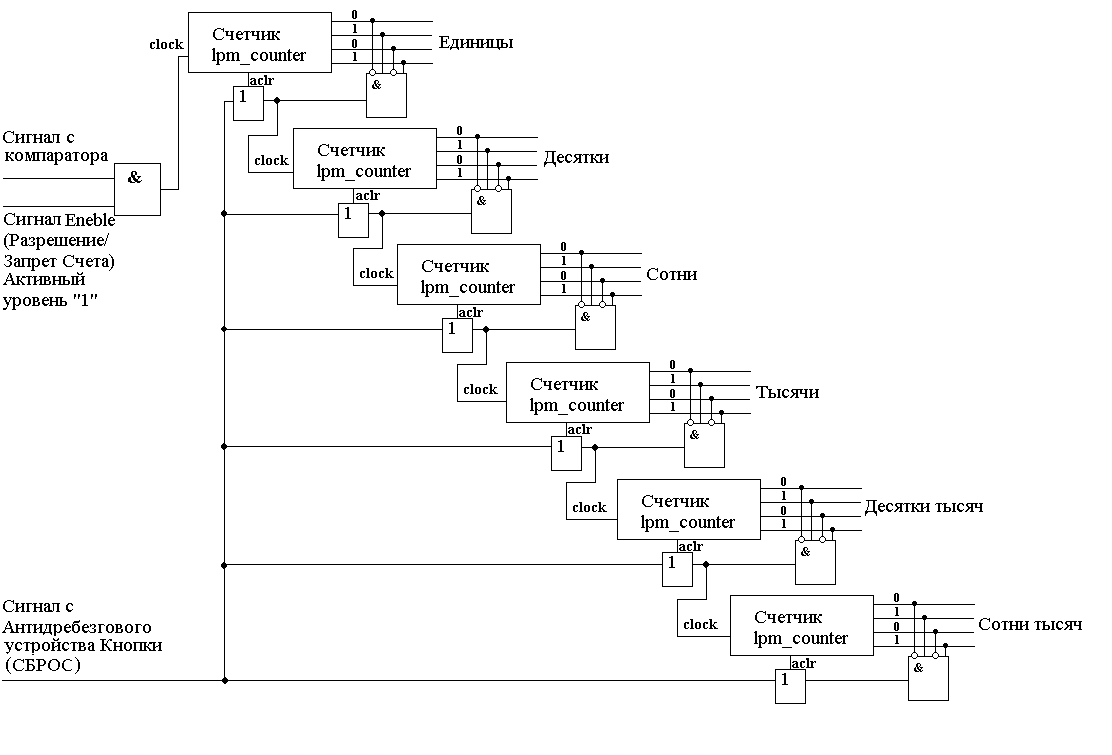
Таблица 3.2.2 Таблица истинности lpm\_counter

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Входы | | | | | | | | | | Выходы | Функции |
| aclr | aset | aload | clk\_en | clock | Sclr | sset | sload | cnt\_en | updoun | q[] |  |
| 1 | x | x | x | x | x | x | x | x | x | 000… |  |
| 0 | 1 | x | x | x | x | x | x | x | x | 111… |
| 0 | 1 | x | x | x | x | x | x | x | x | LPM\_AVALUE | Асинхр. установка значений |
| 0 | 0 | 1 | x | x | x | x | x | x | x | data[] | Асинхр. загрузка данных |
| 0 | 0 | 0 | 0 | x | x | x | x | x | x | q[] | Текущее значение |
| 0 | 0 | 0 | 1 | ∫ | 1 | x | x | x | x | 000… | Синхронный сброс |
| 0 | 0 | 0 | 1 | ∫ | 0 | 1 | x | x | x | 111… | Синхронная установка |
| 0 | 0 | 0 | 1 | ∫ | 0 | 1 | x | x | x | LPM\_SVALUE | Синхронная установка значении для LPM\_SVALUE |
| 0 | 0 | 0 | 1 | ∫ | 0 | 0 | 0 | 0 | 0 | q[] | Текущее значение |
| 0 | 0 | 0 | 1 | ∫ | 0 | 0 | 1 | x | x | data[] | Синхронная загрузка данных |
| 0 | 0 | 0 | 1 | ∫ | 0 | 0 | 0 | 1 | 1 | q[]+1 | Счет в прямом направл. |
| 0 | 0 | 0 | 1 | ∫ | 0 | 0 | 0 | 1 | 0 | q[]-1 | Счет в обратном направл. |

#### 3.2.2 Двоичо-десятичный счетчик (counter10)

При написании программы возникла проблема поразрядного десятичного счета. Двоично-десятичный счетчик реализован помощью макрофункции lpm\_counter. Сброс одного десятичного разряда двоично-десятичного счетчика производиться при достижении величины 10. Если один разряд счетчика досчитал до десяти, в двоичном коде B”1010”, то на вход aclr (асинхронный сброс счетчика) этого же счетчика и на счетчик отвечающего за следующий разряд на вход clock выставляется логическая единица. т.е. следующий счетчик считает десятки предыдущего, со следующим счетчиком также и так далее, таким образом есть 6, соединенных последовательно, счетчиков каждый из которых отвечает за разряд рис 3.2.1.

Рис 3.2.1 Схема десятично-двоичного счетчика



Следовательно, счетное устройство пока может считать до 999999, далее обнуляется, но добавить разрядность не составляет особого труда, просто добавляем счетчик lpm\_counter отвечающий за следующий разряд программно и перепрашиваем ПЛИС. Увеличивать возможности данного устройства можно на столько, на сколько хватает ресурсов ПЛИС (т.е. число триггеров внутри неё ) и фантазии создателя.

#### 

#### 3.2.3 Устройство управлением индикатора (wh1602LCD)

Основан на машине конечных автоматов. Как уже говорилось в пункте 2.4.2, для того чтобы на индикаторе появилось изображение, после подачи на него питания, нужно выполнить с ним ряд действий. К описанию, которых я и приступаю, но прежде нужно заметить, что для подключение индикатора использовалась 8 разрядная шина.

Машина конечных автоматов используется в устройстве управления индикаторам для его инициализации, а за тем и вывод на экран цифр обозначающих число импульсов.

#### 

#### 3.2.3,1 Машина конечных автоматов

Машина конечных автоматов или Машина состояний в AHDL – это структура, позволяющая описывать конечный автомат в виде множества внутренних состояний проекта. Переходы между состояниями синхронизируется тактовым сигналом. Условие и направление перехода определяется для каждого состояния индивидуально. Каждому состоянию можно поставить в соответствие один или несколько входных управляющих сигналов.

Машина состояний описывается как переменная в разделе VARIABLE. Поведение машины описывается в разделе логики после ключевого слова BEGIN. Форма описания машины состояний следующая:

Имя\_переменной : MACHINE [ OF BITS <список битов> ]

WITH STATES (состояние1, состояние2,……);

где

состояние ::=<имя> [= значение ],

<значение> ::= <число>⎢<имя>.

Например.

ss: MACHINE OF BITS (q1,q2,q3)

WITH STATES ( s1 = B”000”,

s2 = B”001”)

s3 =B”010”);

или

ss: MACHINE WITH STATES(s0,s1);

Обязательным является перечисление списка состояний (s0,s1). Если не обязательна конструкция OF BITS, то объявление имени состояния равносильно объявлению переменной типа NODE.

Если объявлена конструкция OF BITS, то перечисленные биты должны существовать физически. В этом случае рассматриваются комбинация значений выходов этих битов (логических ячеек), аналогично константам. Кроме того, количество возможных состояний в этом случае равно 2 в степени “количества битов”, поэтому желательно явно описать все состояния, включая ложные. Первое по списку состояние является состоянием после сигнала сброса.

Машина состояний имеет следующие порты.

.clk – входной тактовый сигнал;

.reset – сигнал сброса, активный уровень – “1”;

.ena – разрешение перехода, активный уровень – “1”.

Поведение машины можно описать с помощью конструкции CASE

SUBDESIGN StateMachine

(

clk, reset, d: INPUT; -- входные порты

q: OUTPUT; -- выходной порт

)

VARIABLE

ss: MACHINE WITH STATES(s0,s1); --переменная ss –машина с с-ми s0, s1

BEGIN

ss.clk = clk; --соединим входной сигнал clk с тактовым входом машины

ss.reset = reset; --соединим входной сигнал reset с входом сброс

--Далее рассматриваем каждое состояние и анализируем условие переходов

CASE ss IS

WHEN s0 => -- в состоянии s0

q = GND; -- устанавливаем выход q в 0

IF d THEN --если на входе d высокий уровень,

ss = s1; --то следующее состояние будет s1,

END IF; -- иначе ss останется прежнем

WHEN s1=> --в состоянии s1

q = VCC; --устанавливаем выход q в 1

IF !d THEN --если на входе d низкий уровень,

ss = s0; --следующее состояние будет s0, иначе ss останется END IF; --прежним;

END CASE;

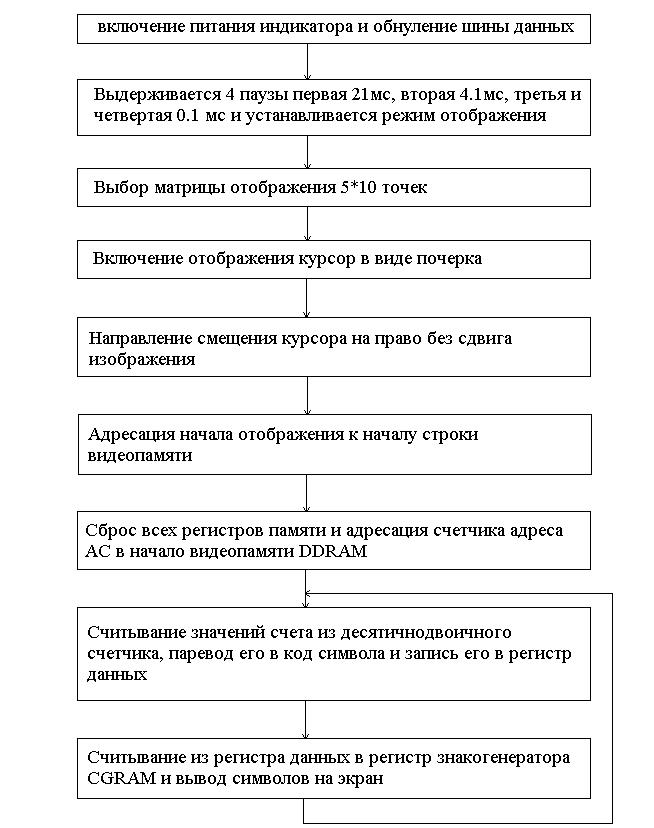
END;

Итак, для работы нужно сначала выставить нужные режимы и флаги через регистр IR(регистр команд) обращение к нему ведется через линию RS выставлением на неё логического нуля. В состоянии Z0 обнуляются все переменные. С состояния P1 по состояние P4 выдерживается четыре паузы 21 мс, 4.1 мс, 0.1 мс и 0.1мс по причине указанной в пункте 2.4.4 до полной готовности индикатора и загружается управляющее слово Н ”38”, что означает: 8 разрядная шина данных, режим развертки одной строки и символы с матрицей 5х8. Далее в состоянии b4, c4 загружается управляющее слово Н”3C” что означает точно тоже самое, но матрица 5х10 точек, чтобы по крупнее показывал. С состояния r5 выбирается режим отображения управляющим словом Н”0C” что означает: включение изображения, курсор в виде почерка . С состояния r6 выбирается направление сдвига курсора вправо без сдвига изображения управляющим словом Н”06”. В состоянии r7 сбрасываются сдвиги, и начало строки адресуется в начало видео памяти DDRAM. В состоянии r8 идет очистка экрана, и счетчик адреса AC адресуется на видео память DDRAM.

На этом этапе все флаги необходимые для формирования изображения загружены. Само формирование изображения начинается с состояния dz0. В состояниях ..z0 значения с счетчика поступают и перекодируются согласно табл3.2.10 дальше записываются в регистр данных. В состоянии ..z1 идет считывание из регистра данных в регистр знакогенератора CGRAM и выводится на экран. И все это начиная с ds циклически, повторяется через заданный интервал времени. Смотри рисунок 3.2.3.1

В данном описании программы я опускал состояния, в которых выдерживались необходимые паузы для общения ПЛИС и микроконтроллера индикатора. Также нужно заметить, что данные в этот модуль программы поступают в параллельном виде, название этой шины Line[].

Рис 3.2.3.1 Блок-схема устройства управления индикатором



**3.2.4 Делитель тактовой частоты для работы индикатора(divFreq)**

Обыкновенный двоичный счетчик выступает в роли делителя тактовой частоты для работы индикатора т.к. индикатору нужна частота меньше чем, выдает кварцевый генератор. Коэффициент деления определяется расчетным и опытным путем (подбором).

#### 

#### 3.2.5 Общая структура программы

Структуру проекта можно увидеть, используя специальное приложение Max+plus/ Hierarhy Display. Откроется окно, в котором проект представлен в виде дерева, указаны имя каждого файла и исходный тип, а также иконка. Двойной щелчок по иконке открывает исходный файл нижнего уровня с помощью соответствующего редактора. Слева от каждой ветви указаны файлы с такими же именами, но другими расширениями, которые созданы в процессе обработки исходного файла.

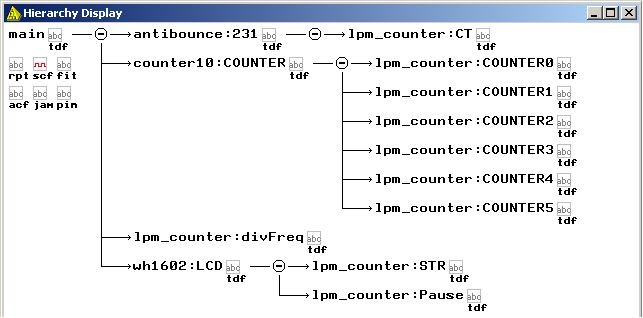


Рис 3.2.5.1 Дерево проекта конфигурации

Тексты каждого модуля можно увидеть в Приложении 3.

**Заключение**

Подводя итог выше проведенной работы, хочу заметить, что получившееся устройство оказалось вполне удачным. Работа устройства была проверена с помощью генератора прямоугольных импульсов. Плата устройства показана на рисунке 4.1.

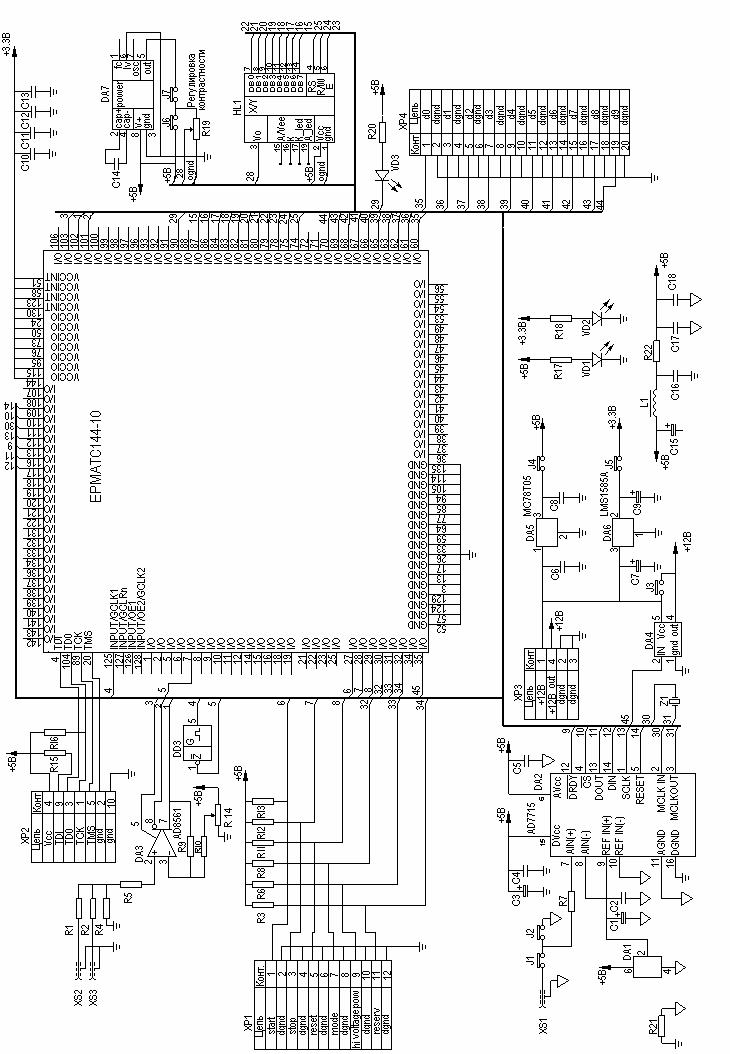
Данное устройство можно подключать не только к СС с ФЭУ, но и к другим детекторам радиоактивных излучений, например счетчик Гейгера. Для этого необходимо пересчитать делитель, который установлен до компаратора.

На этапе разработки в схему счетного устройства была добавлена АЦП Analog Device, AD7715с последовательным интерфейсом SPI. АЦП планируется использовать для измерения напряжения подаваемого на детектор.

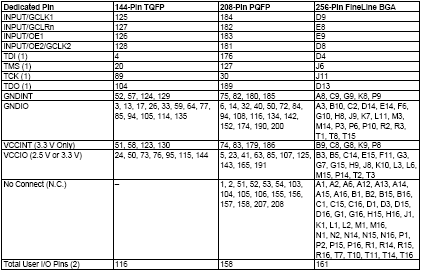
Литература

1. Корсунский М.И.,«Атомное ядро» - М.: Гостехиздат, 1957.
2. **Бродин В.Б., Калинин А.В., «Системы на микроконтролерах и БИС программируемой логики» - М.: Эконом, 2002.**
3. Вицын Н. «Современные тенденции развития систем автоматизированного проектирования в области электроники» // Chip News, № 1, 1997. С. 12–15.
4. Губанов Д. А., Стешенко В. Б., Храпов В. Ю., Шипулин С. Н. «Перспективы реализации алгоритмов цифровой фильтрации на основе ПЛИС фирмы ALTERA» // Chip News, № 9–10, 1997, с. 26–33.
5. Тех. Док. «One Technology Way, P.O. Box9106, Norwood, MA 02062-9106, U.S.A.», http://www.analog.com
6. LCD Controller/Driver LSI. Data Book. © 1994 Hitachi America Ltd.
7. «Алфавитно-цифровые индицирующие ЖК-модули фирмы Powertip.» Каталог, 1-е издание, © 1998 КТЦ-МК.
8. Микросхема: СЕ110. Техническое описание, 1-е издание, © 1997 КТЦ-МК,
9. Микросхема: DB-CE110. Техническое описание, 1-е издание, © 1997 КТЦ-МК.
10. Тех. Док.«MC78TXX», Fairchild Semiconductor Corporation, 2002, http://www.fairchildsemi.com
11. Тех. Док. «LMS1585A», National Semiconductor Corporation, April, 2000, http://www.national.com, Email: europ.support@nsc.com
12. Тех. Док. «ByteBlasterMV Parallel Port Download Cable» Altera Corporation, July, 2002, Version 3.3

#### Приложение 1 (Принципиальная схема устройства)



#### Приложение 2 (Список портов ввода вывода ПЛИС epm 3256a)



#### Приложение 3 (Текст программы)

--Антидребезговая система

INCLUDE "LPM\_COUNTER";

CONSTANT DELAY = 6;

SUBDESIGN ANTIBOUNCE

(

IN : INPUT;

CLK : INPUT;

OUT : OUTPUT;

)

VARIABLE

CT : LPM\_COUNTER WITH(LPM\_WIDTH=DELAY);

TRIG : DFF;

BEGIN

TRIG.D = !IN;

CT.ACLR = !(!IN $ TRIG.Q);

CT.CLOCK = CLK;

TRIG.CLK=CT.Q[DELAY-1];

OUT = !TRIG.Q;

END;

--Десятичнодвоичный счетчик

CONSTANT DIGITS = 6;

INCLUDE "LPM\_COUNTER";

SUBDESIGN COUNTER10

(

-- входные данные

CLOCK : INPUT;

ACLR : INPUT;

COUNT\_ENA : INPUT;

-- выход

Q[DIGITS-1..0][3..0] : OUTPUT;

)

VARIABLE

RESET[DIGITS-1..0] : NODE;

COUNTER[DIGITS-1..0] : LPM\_COUNTER WITH (LPM\_WIDTH=4);

BEGIN

IF COUNTER[0].Q[3..0]==B"1010" THEN RESET[0]=vcc; ELSE RESET[0]=gnd; END IF;

IF COUNTER[1].Q[3..0]==B"1010" THEN RESET[1]=vcc; ELSE RESET[1]=gnd; END IF;

IF COUNTER[2].Q[3..0]==B"1010" THEN RESET[2]=vcc; ELSE RESET[2]=gnd; END IF;

IF COUNTER[3].Q[3..0]==B"1010" THEN RESET[3]=vcc; ELSE RESET[3]=gnd; END IF;

IF COUNTER[4].Q[3..0]==B"1010" THEN RESET[4]=vcc; ELSE RESET[4]=gnd; END IF;

IF COUNTER[5].Q[3..0]==B"1010" THEN RESET[5]=vcc; ELSE RESET[5]=gnd; END IF;

COUNTER[0].clock=clock;

COUNTER[DIGITS-1..1].clock=RESET[DIGITS-2..0];

Q[DIGITS-1..0][3..0]=COUNTER[DIGITS-1..0].Q[3..0];

COUNTER[DIGITS-1..0].aclr=RESET[DIGITS-1..0]#aclr;

COUNTER[0].cnt\_en=COUNT\_ENA;

END;

--Устройство управления индикатором

CONSTANT DATA\_WITH = 8;

INCLUDE "LPM\_COUNTER";

SUBDESIGN WH1602

(

CLK : INPUT;

IN[5..0][DATA\_WITH-1..0] : INPUT;

D[DATA\_WITH-1..0] : BIDIR;

E : OUTPUT;

RW : OUTPUT;

RS : OUTPUT;

)

VARIABLE

clk\_a : NODE;

Ddff[DATA\_WITH-1..0] : dff;

Din[DATA\_WITH-1..0] : dff;

Dmsh[DATA\_WITH-1..0] : NODE;

\_RW : NODE;

\_RS : NODE;

Eena : NODE;

Pause : LPM\_COUNTER WITH (LPM\_WIDTH=9);

STR : LPM\_COUNTER WITH (LPM\_WIDTH=4);

LINE[DATA\_WITH-1..0] : NODE;

ST : MACHINE

WITH STATES ( z0, p1, r1, p2, r2, p3, r3, p4,

r4, b4, c4, r5, b5, c5, r6, b6, c6, r7, b7, c7, r8, b8, c8,

ds, bs, cs, dz0, bz0, cz0, dz1, bz1, cz1 );

Dtri[DATA\_WITH-1..0] : TRI;

OE : NODE;

INC[3..0] : NODE;

INBUF[5..0][DATA\_WITH-1..0] : DFF;

READ : DFF;

BEGIN

clk\_a = dff(!clk\_a, clk, vcc, vcc);

Pause.clock = clk\_a;

-- outputs

Ddff[DATA\_WITH-1..0].d = Dmsh[DATA\_WITH-1..0];

Ddff[DATA\_WITH-1..0].clk = !clk\_a;

Dtri[DATA\_WITH-1..0].in = Ddff[DATA\_WITH-1..0].q;

D[DATA\_WITH-1..0] = Dtri[DATA\_WITH-1..0].out;

Din[DATA\_WITH-1..0].d = D[DATA\_WITH-1..0];

Din[DATA\_WITH-1..0].clk = !dff (clk\_a & Eena , clk, vcc, vcc);

Dtri[DATA\_WITH-1..0].oe = dff (OE, !clk\_a, vcc, vcc);

RW = dff (\_RW, !clk\_a, vcc, vcc);

RS = dff (\_RS, !clk\_a, vcc, vcc);

E = dff (clk\_a & Eena , clk, vcc, vcc);

-- machine

ST.clk = clk\_a;

CASE ST IS

WHEN z0 =>

Dmsh[DATA\_WITH-1..0] = H"FF"; OE = vcc;

Eena = gnd; \_RW = gnd; \_RS = gnd; Pause.aclr = vcc;

ST = p1;

-- загрузка индикатора после включения питания

-- пауза перед включением 21мс

WHEN p1 =>

Dmsh[DATA\_WITH-1..0] = H"38"; OE = vcc;

Eena = gnd; \_RW = gnd; \_RS = gnd; Pause.aclr = gnd;

IF Pause.q[]==200 THEN ST = r1; ELSE ST = p1; END IF;

-- INITIAL SETTING ONCE 00111000

WHEN r1 =>

Dmsh[DATA\_WITH-1..0] = H"38"; OE = vcc;

Eena = vcc; \_RW = gnd; \_RS = gnd; Pause.aclr = vcc;

ST = p2;

-- пауза 7 мс > 4.1 мс

WHEN p2 =>

Dmsh[DATA\_WITH-1..0] = H"38"; OE = vcc;

Eena = gnd; \_RW = gnd; \_RS = gnd; Pause.aclr = gnd;

IF Pause.q[]==100 THEN ST = r2; ELSE ST = p2; END IF;

-- INITIAL SETTING TWICE 00111000

WHEN r2 =>

Dmsh[DATA\_WITH-1..0] = H"38"; OE = vcc;

Eena = vcc; \_RW = gnd; \_RS = gnd; Pause.aclr = vcc;

ST = p3;

-- пауза 0.7 мс > 0.1 мс

WHEN p3 =>

Dmsh[DATA\_WITH-1..0] = H"38"; OE = vcc;

Eena = gnd; \_RW = gnd; \_RS = gnd; Pause.aclr = gnd;

IF Pause.q[]==10 THEN ST = r3; ELSE ST = p3; END IF;

-- INITIAL SETTING THIRD 00111000

WHEN r3 =>

Dmsh[DATA\_WITH-1..0] = H"38"; OE = vcc;

Eena = vcc; \_RW = gnd; \_RS = gnd; Pause.aclr = vcc;

ST = p4;

-- пауза 0.7 мс > 0.1 мс

WHEN p4 =>

Dmsh[DATA\_WITH-1..0] = H"38"; OE = vcc;

Eena = gnd; \_RW = gnd; \_RS = gnd; Pause.aclr = gnd;

IF Pause.q[]==10 THEN ST = r4; ELSE ST = p4; END IF;

-- Function Set 0011NF\*\*

WHEN r4 =>

Dmsh[DATA\_WITH-1..0] = H"3C"; OE = vcc;

Eena = vcc; \_RW = gnd; \_RS = gnd; Pause.aclr = vcc;

ST = b4;

WHEN b4 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = vcc; \_RW = vcc; \_RS = gnd; Pause.aclr = vcc;

ST = c4;

WHEN c4 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = gnd; \_RW = vcc; \_RS = gnd; Pause.aclr = gnd;

IF Din[DATA\_WITH-1].q==0 THEN ST = r5; ELSE ST = b4; END IF;

-- Display ON/OFF = 00001DCB, Display ON,Cursor&Blink OFF

WHEN r5 =>

Dmsh[DATA\_WITH-1..0] = H"0C"; OE = vcc;

Eena = vcc; \_RW = gnd; \_RS = gnd; Pause.aclr = vcc;

ST = b5;

WHEN b5 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = vcc; \_RW = vcc; \_RS = gnd; Pause.aclr = vcc;

ST = c5;

WHEN c5 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = gnd; \_RW = vcc; \_RS = gnd; Pause.aclr = gnd;

IF Din[DATA\_WITH-1].q==0 THEN ST = r6; ELSE ST = b5; END IF;

-- Entry Mode Set = 000001IS, Increment & Shift OFF

WHEN r6 =>

Dmsh[DATA\_WITH-1..0] = H"06"; OE = vcc;

Eena = vcc; \_RW = gnd; \_RS = gnd; Pause.aclr = vcc;

ST = b6;

WHEN b6 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = vcc; \_RW = vcc; \_RS = gnd; Pause.aclr = vcc;

ST = c6;

WHEN c6 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = gnd; \_RW = vcc; \_RS = gnd; Pause.aclr = gnd;

IF Din[DATA\_WITH-1].q==0 THEN ST = r7; ELSE ST = b6; END IF;

-- Return Home

WHEN r7 =>

Dmsh[DATA\_WITH-1..0] = H"02"; OE = vcc;

Eena = vcc; \_RW = gnd; \_RS = gnd; Pause.aclr = vcc;

ST = b7;

WHEN b7 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = vcc; \_RW = vcc; \_RS = gnd; Pause.aclr = vcc;

ST = c7;

WHEN c7 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = gnd; \_RW = vcc; \_RS = gnd; Pause.aclr = gnd;

IF Din[DATA\_WITH-1].q==0 THEN ST = r8; ELSE ST = b7; END IF;

-- Clear Display

WHEN r8 =>

Dmsh[DATA\_WITH-1..0] = H"01"; OE = vcc;

Eena = vcc; \_RW = gnd; \_RS = gnd; Pause.aclr = vcc;

ST = b8;

WHEN b8 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = vcc; \_RW = vcc; \_RS = gnd; Pause.aclr = vcc;

ST = c8;

WHEN c8 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = gnd; \_RW = vcc; \_RS = gnd; Pause.aclr = gnd;

IF Din[DATA\_WITH-1].q==0 THEN ST = ds; ELSE ST = b8; END IF;

--формирование изображения

WHEN ds =>

Dmsh[DATA\_WITH-1..0] = LINE[]; OE = vcc;

Eena = vcc; RW = gnd; \_RS = vcc; Pause.aclr = vcc;

ST = bs;

WHEN bs =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = vcc; \_RW = vcc; \_RS = gnd; Pause.aclr = vcc;

ST = cs;

WHEN cs =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = gnd; \_RW = vcc; \_RS = gnd; Pause.aclr = gnd;

IF STR.q==H"F" THEN ST=dz0;

ELSIF Din[DATA\_WITH-1].q==0 THEN ST = ds; ELSE ST = bs; END IF;

--перемещение в начало

WHEN dz0 =>

Dmsh[DATA\_WITH-1..0] = H"40"; OE = vcc;

Eena = gnd; \_RW = gnd; \_RS = gnd; Pause.aclr = vcc;

ST = bz0;

WHEN bz0 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = vcc; \_RW = vcc; \_RS = gnd; Pause.aclr = vcc;

ST = cz0;

WHEN cz0 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = gnd; \_RW = vcc; \_RS = gnd; Pause.aclr = gnd;

IF Din[DATA\_WITH-1].q==0 THEN ST = dz1; ELSE ST = bz0; END IF;

WHEN dz1 =>

Dmsh[DATA\_WITH-1..0] = H"80"; OE = vcc;

Eena = vcc; \_RW = gnd; \_RS = gnd; Pause.aclr = vcc;

ST = bz1;

WHEN bz1 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = vcc; \_RW = vcc; \_RS = gnd; Pause.aclr = vcc;

ST = cz1;

WHEN cz1 =>

Dmsh[DATA\_WITH-1..0] = H"00"; OE = gnd;

Eena = gnd; \_RW = vcc; \_RS = gnd; Pause.aclr = gnd;

IF Din[DATA\_WITH-1].q==0 THEN ST = ds; ELSE ST = bz1; END IF;

END CASE;

INBUF[5..0][DATA\_WITH-1..0].clk=READ.q;

INBUF[5..0][DATA\_WITH-1..0].d=IN[5..0][DATA\_WITH-1..0];

IF ST==dz1 THEN READ.d=vcc; ELSE READ.d=gnd; END IF;

READ.clk=clk;

IF ST==ds THEN STR.clock=vcc; ELSE STR.clock=gnd; END IF;

STR.aclr=READ.q;

CASE STR.q[] IS

WHEN H"1" => LINE[]=INBUF[5][DATA\_WITH-1..0];

WHEN H"2" => LINE[]=INBUF[4][DATA\_WITH-1..0];

WHEN H"3" => LINE[]=INBUF[3][DATA\_WITH-1..0];

WHEN H"4" => LINE[]=INBUF[2][DATA\_WITH-1..0];

WHEN H"5" => LINE[]=INBUF[1][DATA\_WITH-1..0];

WHEN H"6" => LINE[]=INBUF[0][DATA\_WITH-1..0];

WHEN H"7" => LINE[]=H"20";

WHEN H"8" => LINE[]=H"43";

WHEN H"9" => LINE[]=H"44";

WHEN H"A" => LINE[]=H"45";

WHEN H"B" => LINE[]=H"46";

WHEN H"C" => LINE[]=H"47";

WHEN H"D" => LINE[]=H"48";

WHEN H"E" => LINE[]=H"49";

WHEN H"F" => LINE[]=H"4A";

END CASE;

END;

1. \* промиль (миллионная часть, 10-6). Стабильность частоты   
   Допустимое отклонение частоты от номинального значения. Измеряется в процентах или в "ppm" [↑](#footnote-ref-1)