

Байт-Бластер - простой программатор.

© Александр Торрес, Харьков.
(altor@geocities.com АКА 2:461/28)

В настоящее время, в практику конструирования электронной аппаратуры прочно вошли программируемые логические микросхемы, выпускаемые различными фирмами (Lattice, Xilinx, Altera). Обладая высокой степенью интеграции, универсальностью применения, высоким быстродействием - они практически вытеснили микросхемы малой и средней степени интеграции.

Сама суть программируемой логики, напоминает компьютер, который представляет собой просто набор элементов до тех пор, пока в него не будет загружена программа. Точно также и программируемая логика ничего не "умеет" пока в нее не будет загружена программа конфигурации. По месту нахождения этой программы микросхемы бывают с внутренней и внешней памятью конфигурации. Внешняя память программы, как правило, находится в специализированной микросхеме памяти (обычно это EEPROM с последовательным доступом) и автоматически загружается при подаче питания. Возможна также загрузка конфигурации от микроконтроллера или компьютера, что позволяет менять конфигурацию "на ходу" во время работы. Внутренняя память конфигурации может быть однократно (EPROM) или многократно (EEPROM, Flash) программируемой. В большинстве случаев, возможно программирование микросхем, уже установленных в готовое изделие (плату).

Для начинающих работать с программируемой логикой, для небольших проектов наибольший интерес могут представлять микросхемы с внутренней многократно программируемой памятью фирмы Altera. В зависимости от типа микросхемы, ее программирование может быть осуществлено различными способами - для одних нужен весьма дорогой программатор, для других можно использовать более простые устройства.

Таким устройством служит описываемый ниже ByteBlasterMV (Рисунок 1).

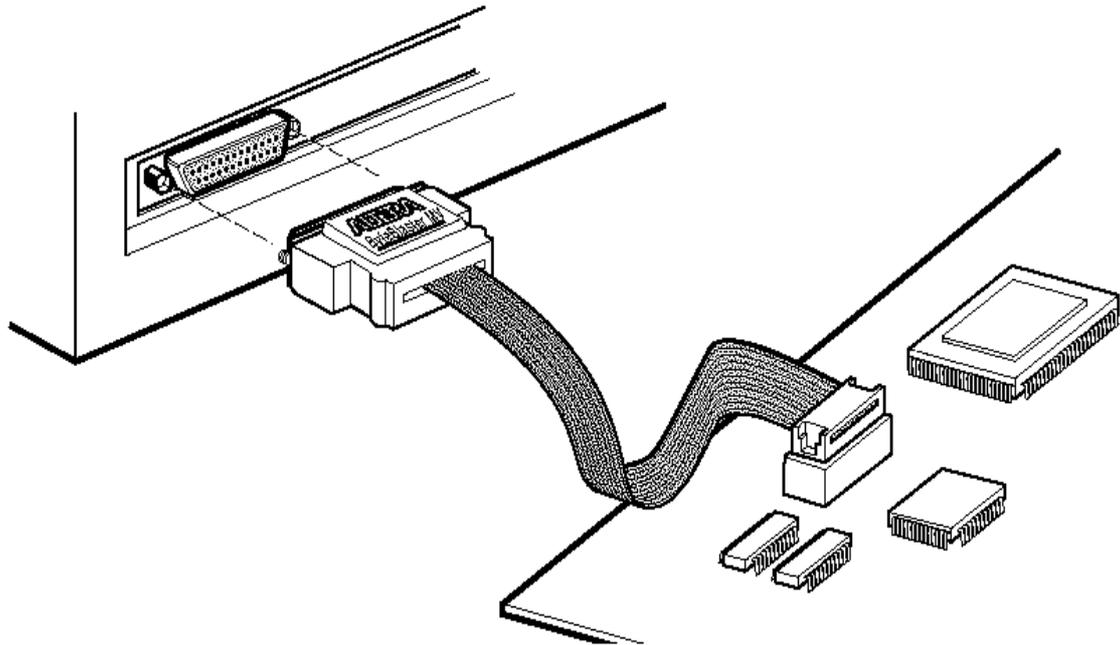


Рисунок 1.

ByteBlasterMV предназначен для подключения параллельному порту персонального компьютера. Поддержка Байт-Бластера встроена в систему проектирования фирмы Altera (MAX+Plus).

ByteBlaster может использоваться для:

- программирования MAX® 9000, MAX 7000S, и MAX 7000A (JTAG Mode);
- конфигурирование FLEX® 10K, KA,KE, FLEX 6000, FLEX 8000 (Passive Serial Mode - PS).

ByteBlaster поддерживает работу микросхем с питанием как 3.3 V и 5.0 V, это быстрый и дешевый метод внутрисхемного программирования, использует стандартный параллельный порт PC и стандартный 10-контактный соединитель, одинаковый для параллельного Байт-Бластера (ByteBlaster™) и последовательного Бит-Бластера (BitBlaster™).

Соответствие сигналов интерфейса и контактов параллельного порта приведено в таблице 1. Схема Байт-Бластера, приведена на рисунке 2. Все последовательные резисторы имеют сопротивление 100 Ом, подтягивающие резисторы - 2.2 кОм.

Таблица 1

Контакт	PS Mode	JTAG Mode
2	DCLK	TCK
3	nCONFIG	TMS
8	DATA0	TDI
11	CONF_DONE	TDO
13	nSTATUS	-
15	VCC	VCC
18-25	GND	GND

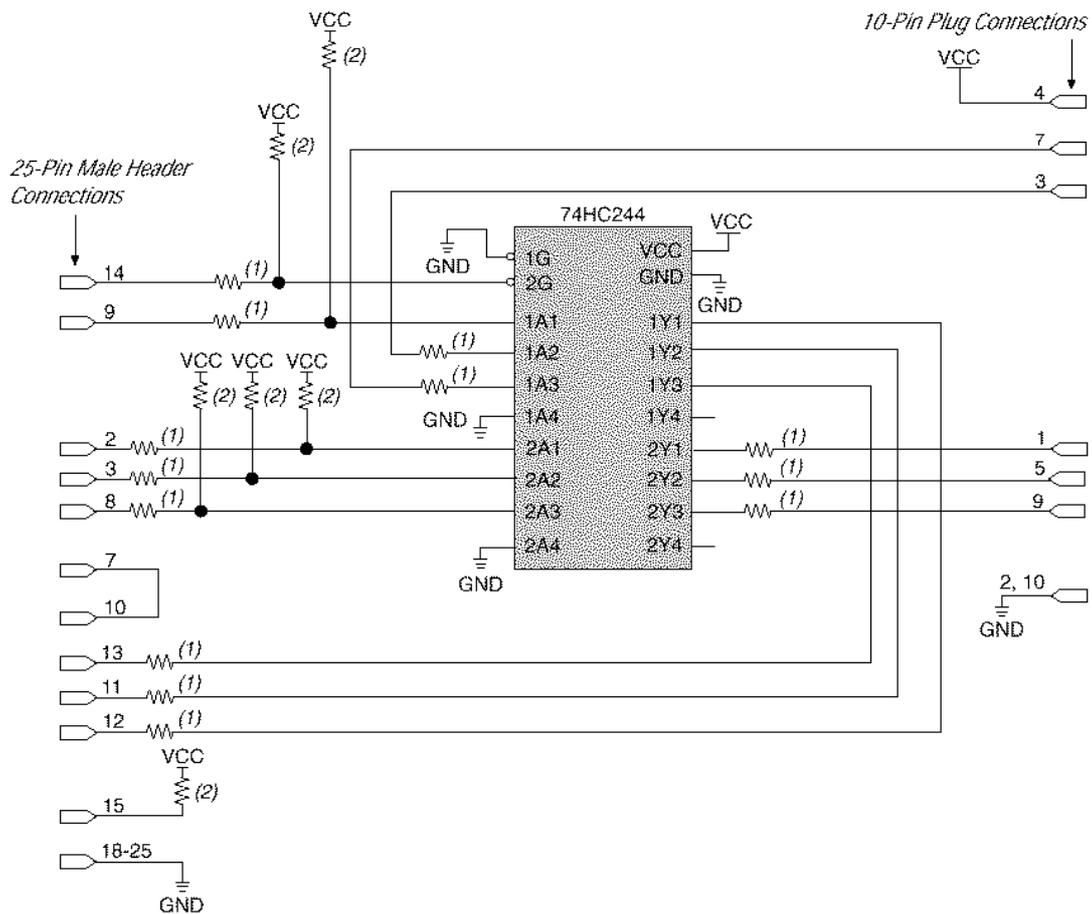


Рисунок 2.

10-контактный разъем показан на рисунке 3, назначение его контактов показано в таблице 2.

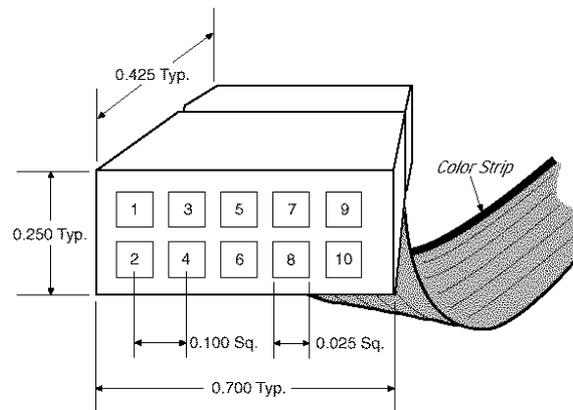


Рисунок 3.

Таблица 2.

Контакт	PS Mode		JTAG Mode	
	Сигнал	Описание	Сигнал	Описание
1	DCLK	Clock signal	TCK	Clock signal
2	GND	GND	GND	GND
3	CONF_DONE	Configuration control	TDO	Data from device
4	VCC	Power	VCC	Power
5	nCONFIG	Configuration control	TMS	JTAG state machine control
6	-	-	-	-
7	nSTATUS	Configuration status	-	-
8	-	-	-	-
9	DATA0	Data to device	TDI	Data to device
10	GND	GND	GND	GND

Питание Байт-Бластера осуществляется от платы, на которой установлено программируемое устройство. При использовании макетных плат или просто сокетов для программирования, необходимо обязательно установить по питанию блокировочные конденсаторы (на нормальных платах предполагается, что они стоят). Невыполнение этого требования может привести к сбоям при программировании и даже выходу из строя программируемой микросхемы.

Диапазон питающего напряжения:

- 4.5 - 5.5 для устройств с VCC=5v;
- 3.0 - 3.6 для устройств с VCC=3v.

На рисунке 4 показано подключение Байт-Бластера для конфигурирования микросхем серий 10K, 8000, 6000, подключение микросхем серий 9000 и 7000 - на рисунке 5.

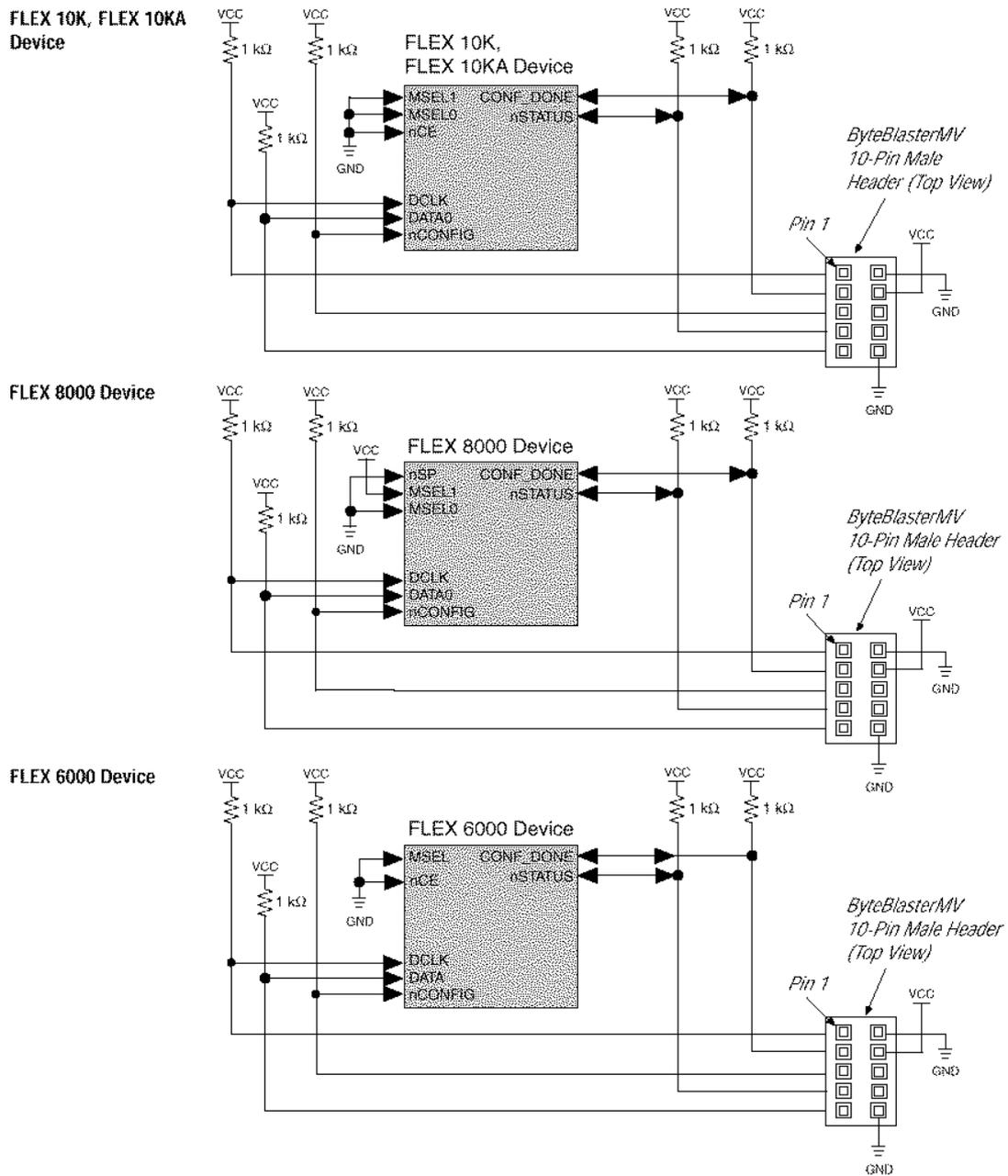


Рисунок 4.

