



预测试的片上系统设计加快了可编程器件的开发

莱迪思半导体公司

2010年3月

Lattice Semiconductor
5555 Northeast Moore Ct.
Hillsboro, Oregon 97124 USA
Telephone: (503) 268-8000
www.latticesemi.com

引言

许多中等规模的可编程逻辑器件（PLD）设计，特别是控制方面的应用，通过片上总线互联一些接口至微处理器，这些微处理器可以是片上芯片或外部的。虽然每个接口往往是相对简单的，构建所有片上的互连和对它们进行调试的任务可能会非常耗时和令人沮丧。越来越多的设计人员使用预设计的基于处理器系统的开发板来加快开发过程。

这种方法有几个好处。首先，没有必要装配许多IP或构建选择的IP至片上总线的缓冲。此外，片上总线结构不需要选择和构建。第二是没有必要手工连接各种IP至片上总线。即使是本文中讨论的相对较小的设计，使用约1000个查找表（LUT），对于顶层的连接大约需要300行的 Verilog代码。第三，使用这种方法构建硬件的时间大大减少。设计人员可能需要花数天时间来组成基本的硬件部分，并进行所有重要的接口调试。一旦调试接口工作使设计已经处于稳定状态，添加或删除接口就变得非常简单。此外，如果修改设计导致调试接口失败，很容易返回到最近所知的成功设计版本。

典型的开发板

这里以开发环境为实例，允许设计人员用预先设计的系统作为起点，然后修改它来构建自己的样机，在将它移植到最终的系统之前使用莱迪思的MachXO迷你开发板。虽然适用于许多应用场合，这个电路板最适用于控制设计，这种设计往往用小型的非易失可编程逻辑器件来实现。图1说明了典型的用可编程逻辑器件实现控制应用的功能。

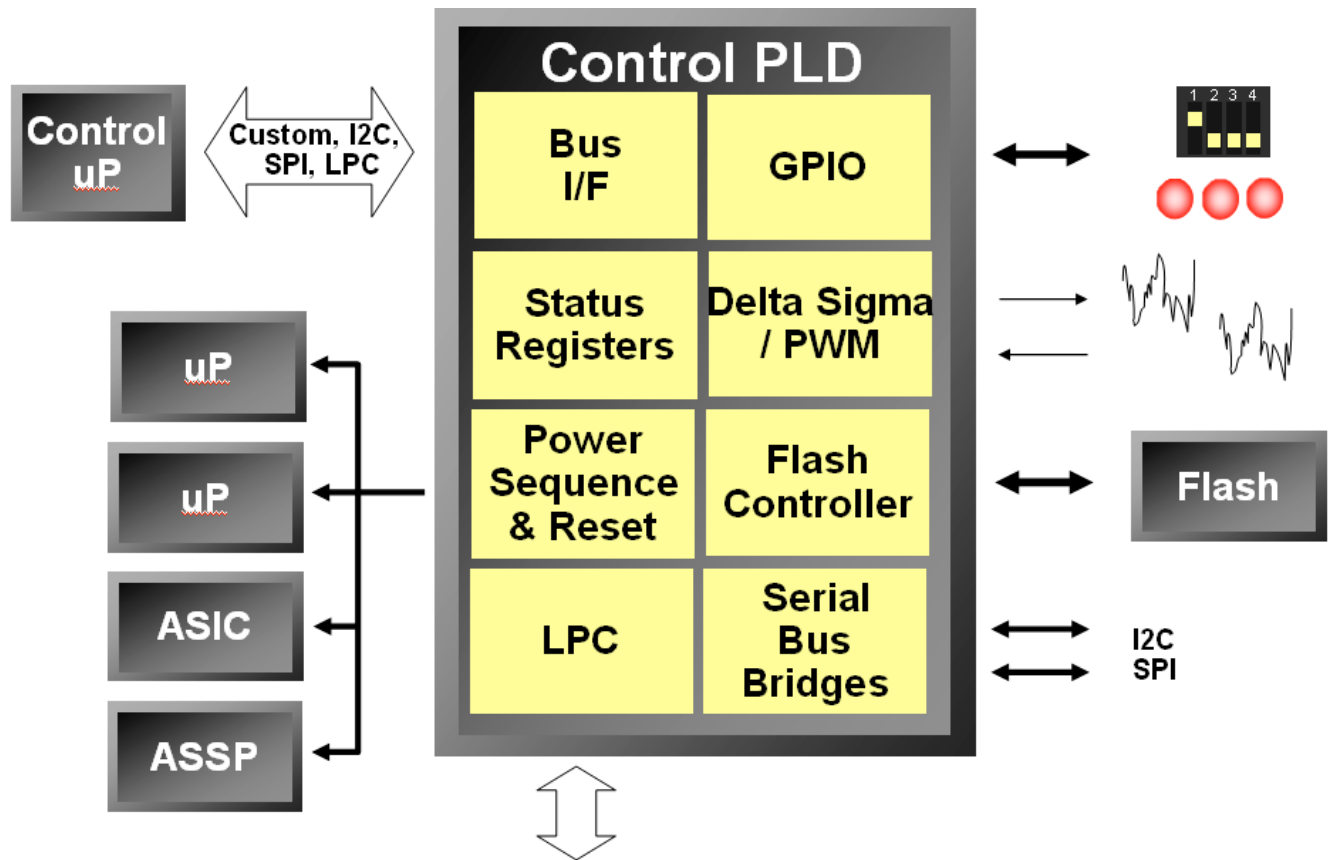


图1用可编程逻辑器件实现的典型控制功能

迷你开发板的中心是MachX0 2280非易失、基于闪存的拥有2280个LUT逻辑实现的可编程逻辑器件。图2展示了MachX0迷你开发板，并对主要接口加了注释。这个电路板提供五种主要的功能：

- 通过发光二极管组、扩展端、DIP开关和按钮开关提供GPIO和状态指示器。
- SPI闪存和I2C温度传感器用作这些常用串行接口的例子。
- SRAM内存提供了额外的便笺式存储器。
- 通过USB的RS232提供了一个调试接口。
- 通过USB的JTAG用作对器件编程的接口。

对于开发过程而言，通过USB接口至RS232的方式和通过USB至 JTAG的方式是至关重要的。利用这两个小的“B”型的USB连接器，可以通过电缆连接到任何个人电脑的USB接口。一旦加载适当的驱动程序（适用于Windows XP及更高版本），通过USB接口至JTAG的方式允许莱迪思的ispVM软件对所需修改设计的MachXO进行编程。采用相应的驱动程序，通过USB接口至RS232的方式对通过USB至RS232的方式进行仿真。驱动程序包括Linux的最新版本，也可用于Windows XP及更高版本。串行端口只作为主机的“COM”端口。此接口为片上设计提供了一个简便的方法，用于发送状态及适当修改、调试信息至主处理器。

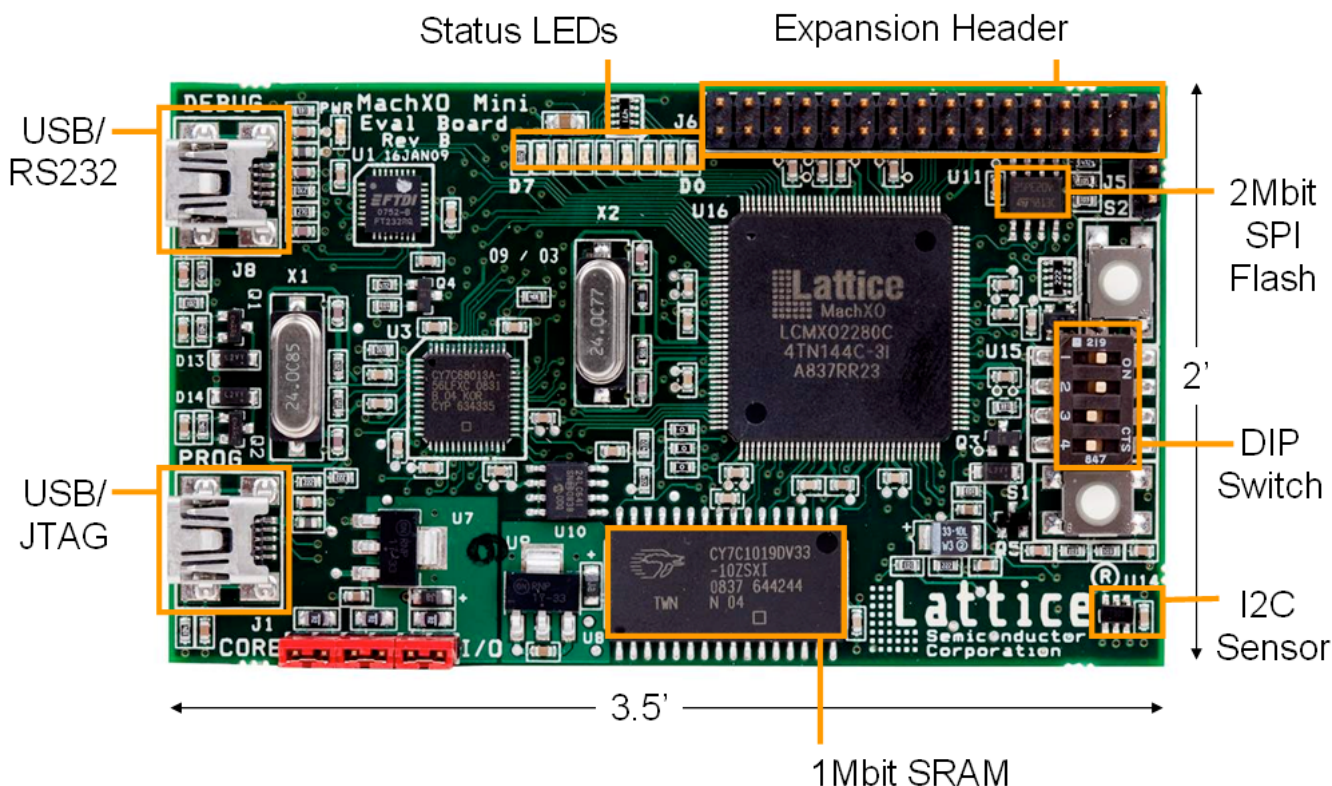


图2 MachXO迷你开发板

预测试的片上系统设计

迷你开发板加速开发的关键是采用预开发的，预加载的片上系统（SOC）设计，可以作为用PLD实现控制功能的起点。下面是一个典型的开发流程和相关的时间，假设相应的软件工具和驱动程序已被加载至PC主机：

1. 验证硬件和接口的正常运行如所期望的那样。（通常为5分钟。）此刻设计人员已经了解了硬件情况。
2. 确认可以使用ispVM编程软件将FPGA配置文件再次载入MachX0。（通常为5分钟。）
3. 使用ispLEVER设计软件重新编译设计，并确认所产生的配置如预期的情况工作。（通常为10分钟。）此时，设计者有一个已知的良好起点。
4. 首先修改设计，重新编译并加载新的配置。（通常是60分钟，但实际的时间取决于所修改的量。）

按照这个开发流程，初始开发需要几个星期的时间可以减少到数小时。SoC设计的概况如图3所示。可以看到设计由六个接口和一个微处理器组成，并通过一个片上总路线联在一起。

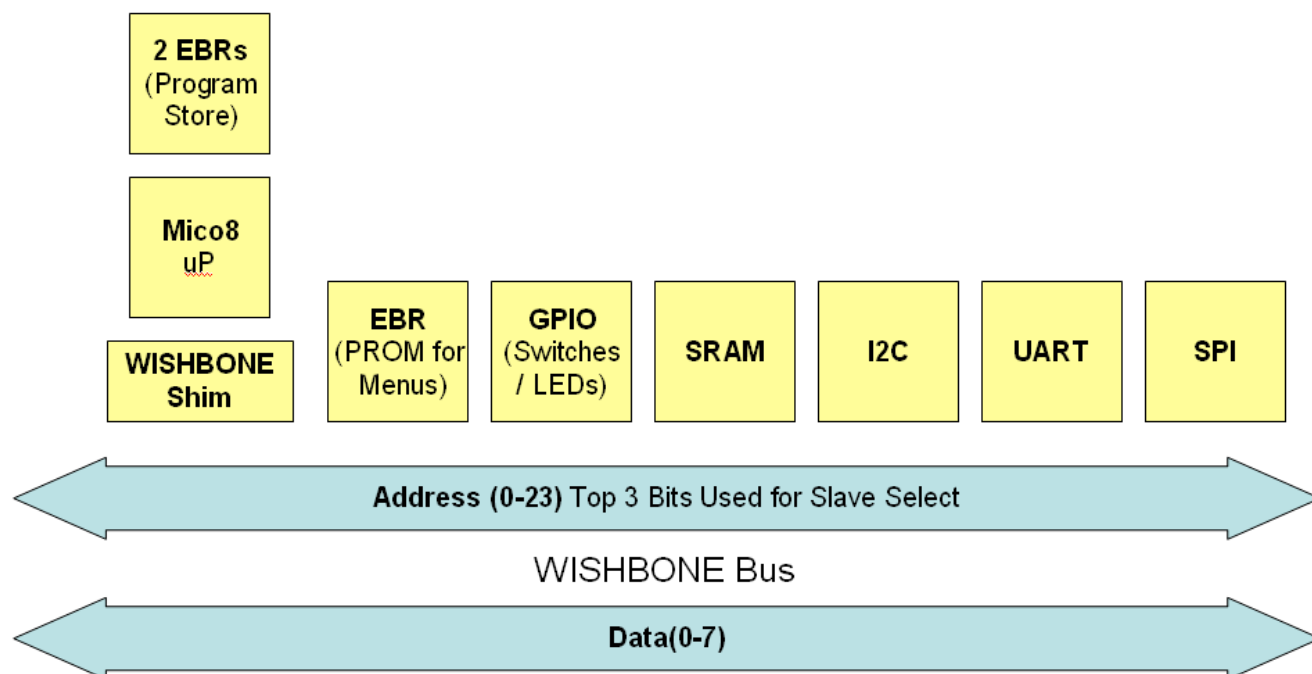


图3 MachX0迷你板预装载了SOC设计

这个设计的中心是片上总线。使用WISHBONE标准：WISHBONE是一个开放源代码的总线标准，为SOC设计提供了一个理想的基本结构。WISHBONE标准的优点有以下几个方面：

- WISHBONE是开放源代码，所以可以用于各种FPGA或ASIC。
- 有大量的适用于WISHBONE接口开放源码IP。
- 简单但灵活的定义使设计人员能够对性能的复杂性进行平衡。

由于许多用于控制的可编程逻辑器件的性能要求相对较低，Wishbone总线是与主微处理器作为主器件（**master**）的共享总线。地址总线有24位，保留的前3位用于产生从器件的选择信号。

LatticeMico8（参考设计RD1026）和相关的WISHBONE适配器（RD1043）用于构成设计核的处理器。Mico8具有开放源代码的优点。这是非常重要的，因为设计人员通常要为嵌入式处理器开发大量的代码。LatticeMico8开放源代码的性质意味着它可以在各种FPGA或ASIC中实现，保护了嵌入式代码的投资。

Mico8有8位的数据通道，18位宽的指令和16或32个通用寄存器。外部存储器是分页的，前面的8位地址包含在相关操作码中。用R15和R16寄存器控制地址总线的其他16位。程序存储器可以设置成512和4096之间的深度。在这种情况下，使用18位宽可达1024深度。程序存储器存储在两个嵌入式RAM块（EBR.）中，这些可以被初始化为器件启动时所需的内容。请注意，每个MachX0 EBR支持9位字节，深度达1024。Mico8大约使用300个LUT。

通过修改汇编清单可以很容易地生成针对Mico8的新代码，汇编清单是作为项目文件的一部分。于是通过修改后的代码可以LatticeMico8工具链和生成的相应HEX文件。

ispLEVER工具可以使用这些HEX文件，合适地初始化EBR用于编程的存储。Mico8工具链也是开放源代码，可从莱迪思网站下载源代码以及可执行文件。

其他的EBR存储器连接到Wishbone总线。这个EBR配置成1024深度 8位字节。针对通过RS232接口的Mico8显示的各种菜单，存储空间是用来共享数据。

SPI接口利用了莱迪思的SPI WISHBONE控制器参考设计（RD1044）的修改版本。SPI接口连接外部2兆位的SPI闪存至Wishbone总线。如果需要的话，SPI WISHBONE控制器可以用来控制多达8个从器件（slave）。尽管在这种情况下使用了8位寄存器，接收和发送寄存器可配置为1到32位长度，。寄存器是双缓冲的，当微处理器服务响应中断请求以清除其他缓冲区时，允许接收或发送数据。这个参考设计大约使用了113个LUT。这个I2C接

口使用了Wishbone总线接口的莱迪思的I2C主器件（master）参考设计（RD1046）。这个设计支持7位或10位寻址模式和一个8位接收/发送缓冲器。I2C参考设计大约使用了234个LUT。

UART的设计采用了**Lattice WISHBONE UART**参考设计（RD1042.）。这个设计用类似常用的独立NS16450 UART功能实现了一个灵活的UART。设计占用大约291个LUT的资源。SRAM接口相对比较简单，是针对这个项目的代码。这是一个简单的**WISHBONE**外设的很好的例子。

整合所有的一切

MachX0迷你开发板的目标是为了加快小的、非易失FPGA设计的开发速度。为了实现这一点，所有在本文中讨论的关于系统芯片设计的信息可作为一个软件包从莱迪思网站上下载。而且，除了文件和源文件，针对所讨论的SoC设计，还提供了一个完整的ispLEVER项目，以及**TransFR**和使用睡眠模式以减少功耗的演示。提供的信息包括项目文件和所有必要的优先选择和约束，以便在第一时间编译设计。

因此，如果要对你系统中的PLD控制设计进行全面的修改，请仔细看看这个开发系统和预测的设计。它可以为你的下一个设计节省几周的时间。

###