

使用 Altera MAX 系列作为微控制器 I/O 扩展器

2014.09.22

AN-265



订阅



反馈

很多微控制器和微处理器芯片都会限制可用的 I/O 端口和芯片数量，以便节省引脚数量并减小封装大小。在采用这些有限芯片的设计中，可以使用 Altera 器件来实现 I/O 扩展器。利用这种方法，无需更改设计中的微控制器或微处理器，即可增加 I/O 引脚的数量。

本文档中的设计实例介绍了使用以下 Altera 器件实现微处理器 I/O 扩展器的方法：

- MAX[®] 10 FPGA
- MAX V
- MAX II
- MAX 3000A

相关链接

- [MAX II、MAX V 和 MAX 3000A 设计实例](#)
提供与这篇应用笔记 (AN 265) 对应的设计文件。
- [MAX 10 设计实例](#)
提供与这篇应用笔记 (AN 265) 对应的 MAX 10 设计文件。

使用 Altera 器件作为 I/O 扩展器的优势

表 1: 使用 Altera 可编程器件作为 I/O 扩展器的优势

有关每个器件封装对 MultiVolt I/O 的支持情况以及可用 I/O 引脚数量的详细信息，请参阅相关信息。

特性	优势
可编程逻辑功能	可以将 I/O 扩展器以及其他一些功能一起集成到可编程器件中。例如，可以集成 LED 驱动程序或总线桥接逻辑。此外，利用 MAX II、MAX V 或 MAX 10 器件中的闪存 (UFM)，还可以集成使用存储器的功能。得益于这种集成功能，可以减少整个电路板上的组件数量。
MultiVolt I/O 支持情况	Altera 可编程器件支持 MultiVolt I/O 功能。可以使用此功能来与电压为 3.3 V、3.0 V、2.5 V、1.8 V、1.5 V 和 1.2 V 的 I/O 引脚进行交互。利用此功能，可以实现电压电平转换，从而使较新的 2.5 V、1.8 V、1.5 V 或 1.2 V 器件能与较旧的 3.3 V 器件进行交互。

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2008
Registered

ALTERA
now part of Intel

特性	优势
大量 I/O 引脚	Altera 可编程器件中提供了大量 I/O 引脚，因而它们非常适合实现微控制器 I/O 扩展。例如，如有 500 个 I/O 引脚，就可以实现 62 个 8 位端口。
系统内可编程性	得益于系统内可编程性 (ISP)，在电路板上重新编程时就有了更大的灵活性。

相关链接

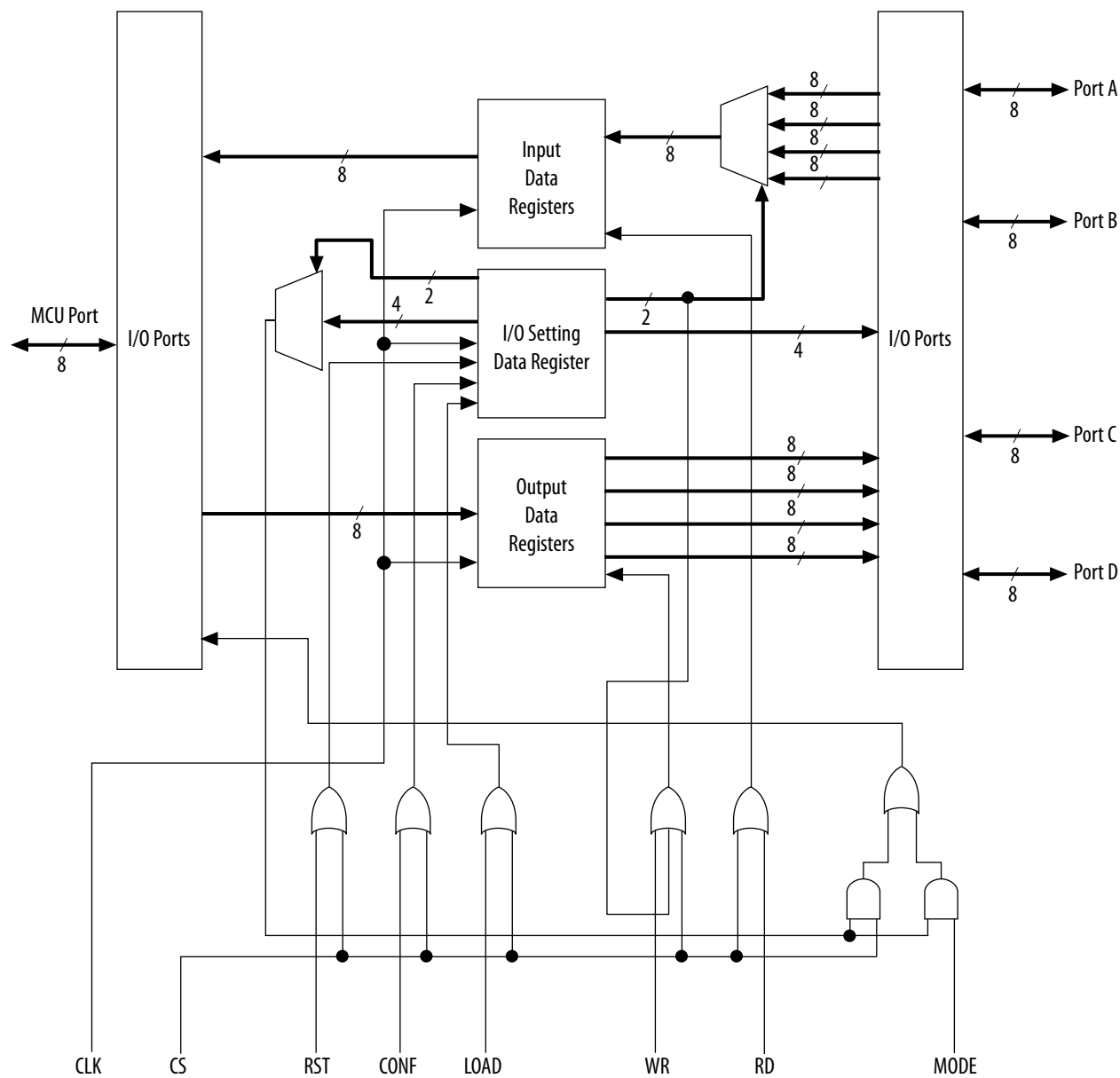
- [MAX 10 FPGA 器件概述](#)
提供 MAX 10 FPGA 器件系列中的每款器件可用的最大 GPIO 资源数量。
- [MAX 10 通用 I/O 用户指南](#)
列出 MAX 10 FPGA 器件系列支持的每种 I/O 标准的 I/O 标准电压和引脚支持信息。
- [MAX II 器件手册的简介部分](#)
提供 MAX II 器件系列中每款器件的最大用户 I/O 引脚数目
- [MAX II 器件手册的 MAX II 体系结构部分](#)
列出 MAX II 器件对 MultiVolt I/O 的支持情况。
- [MAX 3000A 可编程逻辑器件系列数据手册](#)
提供每款器件的最大用户 I/O 引脚数目并列出 MAX 3000A 器件对 MultiVolt I/O 的支持情况。
- [MAX V 器件系列概述](#)
提供 MAX V 器件系列中的每款器件可用的最大 GPIO 资源数量。
- [MAX V 体系结构](#)
列出 MAX V 器件对 MultiVolt I/O 的支持情况。

微控制器 I/O 扩展器的实现

本设计实例中的 I/O 扩展器包含四个端口 — A、B、C 和 D：

- 可以将每个端口作为 I/O 端口独立编程。
- 每个端口为 8 位宽。
- 每个端口都是双向的，并在处于三态时用作输入端口。
- 要执行读取和写入操作，请将这些端口中的任意一个与微控制器单元 (MCU) 端口相连。

图 1: 微控制器 I/O 扩展器设计实例的结构图



微控制器 I/O 扩展器中的引脚功能

表 2: 微控制器 I/O 扩展器设计实例中的输入和输出引脚

I/O 名称	类型	说明	有效
CLK	输入	用于触发输入数据寄存器、输出数据寄存器和 I/O 设置数据寄存器的输入时钟信号。	正边沿

I/O 名称	类型	说明	有效
MCU 端口	双向	与 MCU 数据总线进行交互并发送配置数据的 8 位双向数据线路。 <ul style="list-style-type: none"> 要向 I/O 扩展器写入或从中读取 8 位数据，请将 WR 或 RD 信号与正边沿触发器一起使用。 要将 MCU 端口设置为数据或地址总线，请使用 MODE 信号。 	—
CS	输入	用于选择器件的输入信号。	低电平
RST	输入	用于以异步方式将器件及所有内部寄存器复位的输入信号。	低电平
CONF	输入	<ul style="list-style-type: none"> 将来自 CLK 上升沿 MCU 端口的 address[5..0] 存储到配置寄存器中。 启用发往配置寄存器的时钟信号。 忽略 address[7..6] 上的数据。 	低电平
LOAD	输入	<ul style="list-style-type: none"> 将来自 CLK 上升沿 MCU 端口的 address[1..0] 存储到配置寄存器中。 启用发往配置寄存器的时钟信号。 忽略 address[7..2] 上的数据。 	低电平
WR	输入	<ul style="list-style-type: none"> 使 MCU 端口上的数据写入到 CLK 上升沿的已启用端口（A、B、C 或 D）。 启用发往输出数据寄存器的时钟信号。 	低电平
RD	输入	<ul style="list-style-type: none"> 启用从 CLK 上升沿的已启用端口（A、B、C 或 D）到 MCU 端口的数据总线传送。 启用发往输入数据寄存器的时钟信号。 	低电平
MODE	输入	<ul style="list-style-type: none"> 处于高电平时，会使 MCU 端口三态化。 处于低电平时，则根据配置寄存器将 MCU 端口设置为输入或输出。 	—
端口 A	双向	可编程为输出模式或输入模式的 8 位通用 I/O 端口。	—
端口 B	双向	可编程为输出模式或输入模式的 8 位通用 I/O 端口。	—

I/O 名称	类型	说明	有效
端口 C	双向	可编程为输出模式或输入模式的 8 位通用 I/O 端口。	—
端口 D	双向	可编程为输出模式或输入模式的 8 位通用 I/O 端口。	—

相关链接
[微控制器 I/O 扩展器的配置](#) (第 6 页)

微控制器 I/O 扩展器的功能说明

I/O 扩展器作为一个从器件加以运行，负责通过 MCU 端口向微控制器发送数据和从微控制器接收数据。

表 3: 微控制器 I/O 扩展器的操作模式

在读取和写入操作期间，请将 MODE 信号控制引脚设为低电平。

模式	说明
读取	<p>在读取模式下，微控制器通过 I/O 扩展器接收输入数据。</p> <ul style="list-style-type: none">所选的端口（A、B、C 或 D）成为输入端口，MCU 端口则成为输出端口。当 RD 和 CS 信号处于低电平时，应用将通过输入数据寄存器启用时钟。在 CLK 的上升沿，I/O 扩展器会将来自所选端口的数据写入到输入数据寄存器。
写入	<p>在写入模式下，微控制器通过 I/O 扩展器将数据发送出去。</p> <ul style="list-style-type: none">所选的端口（A、B、C 或 D）成为输出端口，MCU 端口则成为输入端口。当 WR 和 信号处于低电平时，应用将通过输出寄存器启用时钟。在 CLK 信号的上升沿，I/O 扩展器会将 MCU 端口接收到的数据写入到所选端口的输出数据寄存器。
禁用	<p>在此模式下，微控制器无法通过 I/O 扩展器接收或发送数据。</p> <p>当 WR 或 RD 信号处于高电平时，应用会通过 I/O 数据寄存器禁用 CLK。</p>

相关链接
[微控制器 I/O 扩展器的设计验证](#) (第 7 页)
提供读取和写入操作示例。

微控制器 I/O 扩展器的配置

要将端口设置为输入、输出或禁用，必须配置 I/O 扩展器。

I/O 扩展器的配置数据是一组六位寄存器：

- 四个 I/O 端口配置位（位 5、4、3 和 2）
- 两个已启用端口地址位（位 1 和 0）

表 4: I/O 扩展器地址总线配置功能

地址	位	功能	配置
A7:A6	7 - 6	无功能	—
A5	5	端口 D 的配置位	<ul style="list-style-type: none"> • 0 = 配置为输入 • 1 = 配置为输出
A4	4	端口 C 的配置位	<ul style="list-style-type: none"> • 0 = 配置为输入 • 1 = 配置为输出
A3	3	端口 B 的配置位	<ul style="list-style-type: none"> • 0 = 配置为输入 • 1 = 配置为输出
A2	2	端口 A 的配置位	<ul style="list-style-type: none"> • 0 = 配置为输入 • 1 = 配置为输出
A1:A0	1 - 0	端口地址	<ul style="list-style-type: none"> • 00 = 启用端口 A • 01 = 启用端口 B • 10 = 启用端口 C • 11 = 启用端口 D

当 **MODE** 信号处于高电平且 **CS** 信号处于低电平时，此设计会将配置数据置于输入地址总线（MCU 端口）上。

要激活新配置，请发送 **CONF** 信号的单个脉冲，并持续一个时钟周期。当 **CONF** 信号和 **CS** 信号都处于低位时，时钟通过配置寄存器加以启用。由此带来的结果是，I/O 扩展器会将配置数据写入到配置寄存器中。这种发送方式会将所选的配置设为 I/O 扩展器的状态。

如果希望选择要使用这四个端口中的哪一个（A、B、C 或 D）而不配置这些端口，请在 **CS** 信号处于低电平时将 **LOAD** 信号置位为低电平。在 **CLK** 信号的上升沿，只有已启用的端口地址数据位写入到配置寄存器。

配置所选端口（输入或输出）时还会配置 MCU 端口方向。例如，如果将端口 A 配置为输入并将其启用，则 MCU 端口的方向将变为输出。无法直接配置 MCU 端口的输入或输出方向。不过，

如果需要访问配置寄存器，则可以使用 `MODE` 信号控制引脚将 MCU 端口三态化，然后将其转变成输入端口。

I/O 扩展器的设计支持复位模式。如果在 `CS` 信号处于低电平时置位 `RST` 信号，则所有寄存器内容都会以异步方式复位为零。包括 MCU 端口在内的所有 I/O 扩展器都会被设为输入模式，这也是 I/O 扩展器的初始状态。

相关链接

[微控制器 I/O 扩展器中的引脚功能](#) (第 3 页)

微控制器 I/O 扩展器目标器件中的资源使用情况

目标器件中有充足的资源可用来实现本设计实例及其他逻辑。

表 5: 可用来实现微控制器 I/O 扩展器设计实例的器件目标

下表列出了可以以哪些器件为目标来使用 Quartus® II 软件实现本设计实例，以及这些器件的资源使用情况。

系列	器件	资源使用情况			
		资源	可用量	使用量	利用率 (%)
MAX 3000A	EPM3128ATC100	宏单元	128	51	39
		触发器	128	50	39
		I/O 引脚	80	48	60
		可共享的扩展器	128	2	1
MAX II	EPM240T100C3	逻辑单元	240	67	27
		触发器	240	50	20
		I/O 引脚	80	48	60
MAX 10 FPGA	10M08SAE144C8G	逻辑单元	8064	75	0.9
		触发器	8064	59	0.7
		I/O 引脚	103	48	47

微控制器 I/O 扩展器的设计验证

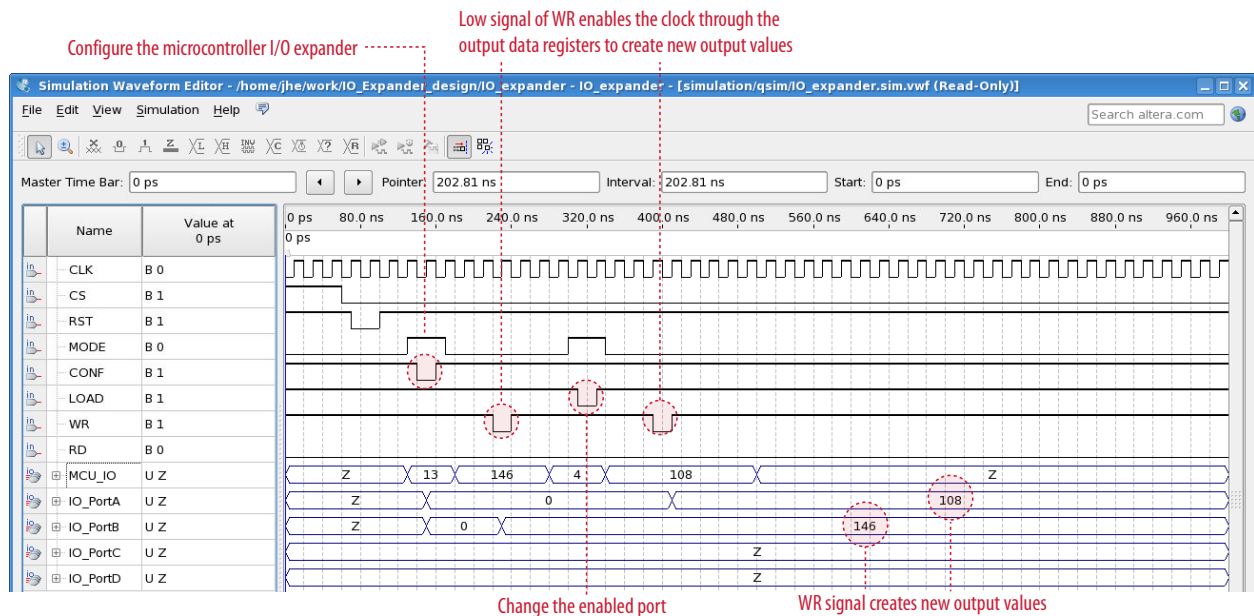
可以使用 Quartus II 软件在目标设备上验证本设计实例的功能仿真和时序仿真。

相关链接

[微控制器 I/O 扩展器的功能说明](#) (第 5 页)

对微控制器 I/O 扩展器写入模式的时序进行仿真

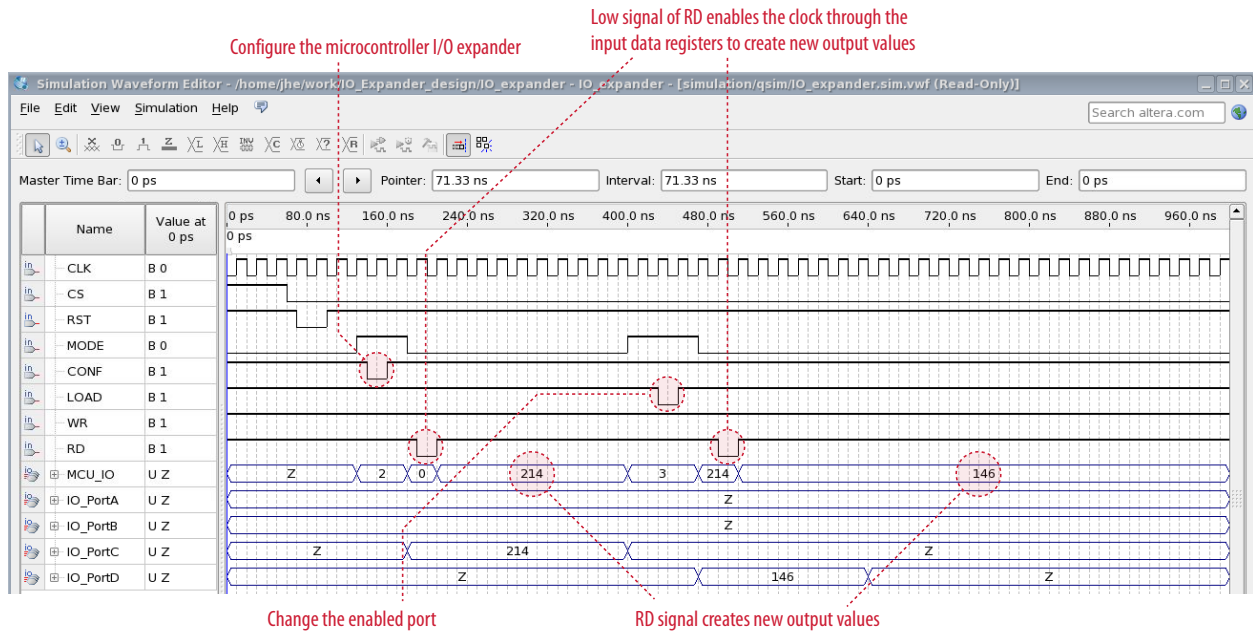
图 2: 微控制器 I/O 扩展器的写入模式时序仿真



1. 在所有 I/O 引脚都处于三态模式的情况下，对每一个端口进行初始化。
2. 初始化之后，将端口 A 和端口 B 配置为输出，将端口 C 和端口 D 配置为输入。
3. 为端口 B 启用写入操作。
当 WR 置位低电平信号时，通过输出数据寄存器实现的时钟将处于启用状态，并且会在 MCU 端口中的输出端口 B 处创建一个新的输出值 (146)。
4. 在 LOAD 信号置位为低电平后，请启用端口 A。
当 WR 信号处于低电平以便通过输出数据寄存器启用时钟时，会在输出端口 A 位置创建 MCU 端口数据 (108)。

对微控制器 I/O 扩展器读取模式的时序进行仿真

图 3: 微控制器 I/O 扩展器的读取模式时序仿真



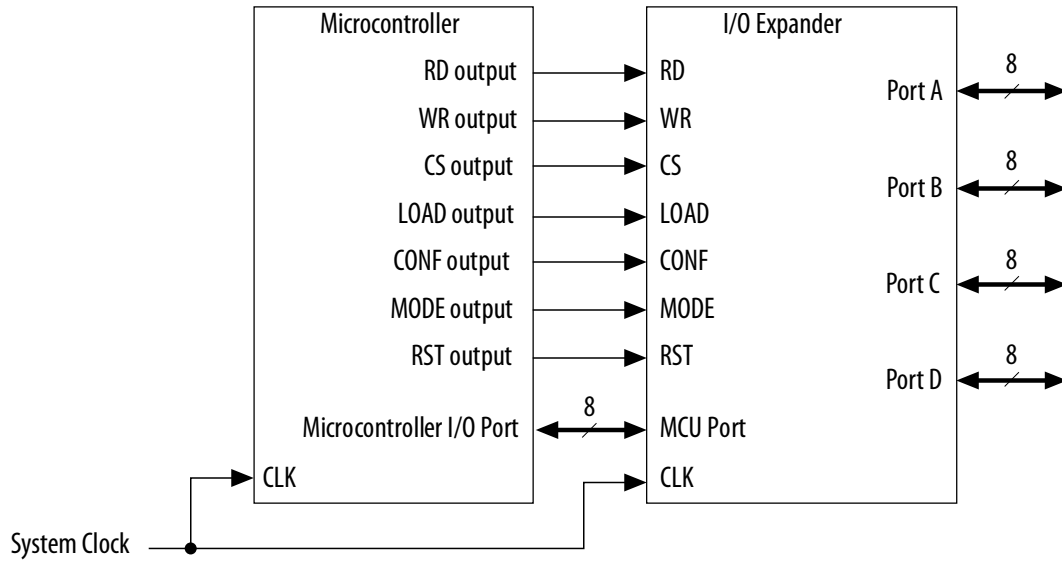
1. 在所有 I/O 引脚都处于三态模式的情况下，对每一个端口进行初始化。
2. 初始化后，将端口 A、B、C、D 配置为输入。
3. 为端口 C 启用读取操作。
当 RD 置位低电平信号时，通过输入数据寄存器实现的时钟将处于启用状态，并且会在端口 C 中的 MCU 端口处创建一个新的输出值 (214)。
4. 在 LOAD 信号置位为低电平后，请启用端口 D。
当 RD 信号处于低电平以便通过输入数据寄存器启用时钟时，会在 MCU 端口位置创建端口 D 数据 (146)。

微控制器 I/O 扩展器的应用

可以修改本示例设计，以使其适合任何 8 位微控制器，例如 PIC 和 8051 微控制器。

图 4: 使用微控制器 I/O 扩展器增加微控制器 I/O 端口数目

下图显示了为增加微控制器的 I/O 端口数目而应用于微控制器系统的 I/O 扩展器。



文档修订历史

日期	版本	更改内容
2014 年 9 月	2014.09.22	<ul style="list-style-type: none">• 增加了 MAX 10 器件的相关信息。• 增加了对 MAX V 器件的支持。• 增加了 MAX V、MAX II 和 MAX 3000A 器件系列所缺少的设计实例文件下载链接。• 增加了 MAX 10 器件的设计实例文件下载链接。• 删去了一些表格，这些表格列出 MAX II 和 MAX 3000A 器件对 MultiVolt I/O 的支持情况以及最大用户 I/O 引脚数量；添加了一些指向相关器件特有的文档的链接，这些文档中列出了上述表格中的信息。所链接到的这些文档提供了更多详细信息，并且包含有关每个器件系列的最新信息。• 将显示 I/O 扩展器地址总线配置的图更改为一张表，这样参考起来更加方便。• 删去了列出端口地址配置值的表格，合并了列出 I/O 扩展器地址总线配置详情的表格中的信息。• 添加了一个列出设计实例目标设备的表格，并将资源利用信息合并到此表格中以方便参考。• 更新了微控制器 I/O 扩展器的读取和写入时序仿真数据。• 重新整理并重新撰写了所有章节，使其更加清晰且反映最新风格。• 更新了模板。
2004 年 3 月	2.0	未提供修订历史信息。
—	1.0	初始版本。