MCU Port Expansion Using ATF15xx CPLDs

ATF15xx

Application Note

1. Introduction

Today many microcontrollers (MCUs) provide a limited number of I/O ports and pins in order to reduce the package size. However, there are instances when the target application calls for more I/O pins than those provided by the MCU itself. Since substituting the MCU is not always a feasible option, a CPLD based port expansion unit can be used to provide the perfect solution.

This application note describes the implementation of such an I/O port expansion unit, called I/O Expander. Using only six I/O pins, up to four 8-bit I/O ports can be controlled, depending on the density and package type of the CPLD used. With a smaller device like the ATF1502, one 8-bit I/O port plus an additional input-only 8-bit port can be implemented. With a larger ATMEL CPLD device such as the ATF1508, four 8-bit I/O ports can be implemented occupying only about half of the CPLD resources. The design can be easily customized to support either a larger number of ports, or a different combination of port direction and/or width.

As a reference, the VHDL source files for the I/O Expander design are included in this application note. Assembly source code is also provided to demonstrate the connection of the I/O Expander with an ATMEL AVR MCU. This document covers the case of the I/O Expander design having four I/O ports. For using the design with a smaller CPLD, the available device resources should be taken into consideration.

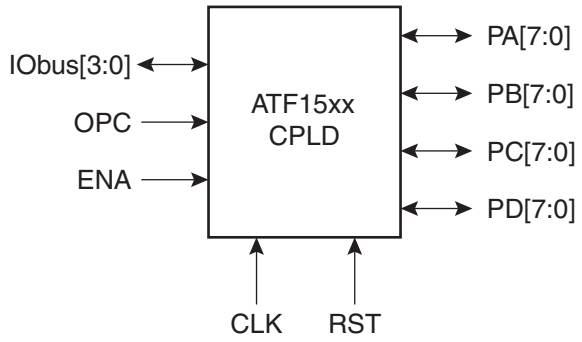
Although the design provided herein is compatible with the ATF15xxAS (5V), ATF15xxASV (3.3V) and ATF15xxBE (1.8V) families of the ATF15xx ATMEL CPLD devices, the ultra-low-power characteristics of the ATF15xxBE family along with its low cost make it an ideal candidate, especially when designing for portable applications.

3635A–PLD–09/06

2. Block Diagram

Figure 2-1 presents the generic diagram of the I/O Expander in the case of a full implementation. The four I/O ports PA…PD[7:0] are controlled by the IObus[3:0] data I/O port, supported by two control signals (OPC and ENA).

Figure 2-1. I/O Expander Block Diagram



3. Pin Description

Table 3-1 describes the I/O pins used in the I/O Expander design:

Table 3-1. I/O Expander Pins Description

Signal Name	Type	Description
IObus[3:0]	Bidirectional	4-bit bidirectional data IO that connects to an MCU port
OPC	Input	Used to indicate presence of an opcode and control the duration of the CPLD data output
ENA	Input	Used for clocking port data to/from the CPLD device
PA[7:0]	Bidirectional	8-bit general purpose IO port that can be programmed as input or output
PB[7:0]	Bidirectional	8-bit general purpose IO port that can be programmed as input or output
PC[7:0]	Bidirectional	8-bit general purpose IO port that can be programmed as input or output
PD[7:0]	Bidirectional	8-bit general purpose IO port that can be programmed as input or output

4. Functionality

Communication of the I/O Expander with the MCU is done through the 4-bit bidirectional bus IObus[3:0]. Data are transferred in a nibble-by-nibble basis, while two control signals (OPC, ENA) coordinate the operation. Writing to a CPLD port (such as PA) is a transaction where the MCU first sends to the CPLD an opcode indicating the kind of operation and selected port, followed by the two data nibbles with the high order nibble transferred first. Similarly, to read a port the MCU will first send the opcode indicating the read operation and port. Then the MCU will read the IObus twice to get the value from the specified port.

The opcode is used to specify the port address, and to set the port direction and MCU-I/O Expander data transaction type (read/write). As the direction of an I/O Expander port is common

for all its pins, port pins cannot be individually set to input or output. Upon power-up all ports are configured as inputs. To configure a port as output, the MS bit of the opcode has to be set to one (1). Writing zero (0) configures the port as input. Output ports support both write and read operations, in which the read operation returns the actual value present on the pins. Input ports only support the read operation.

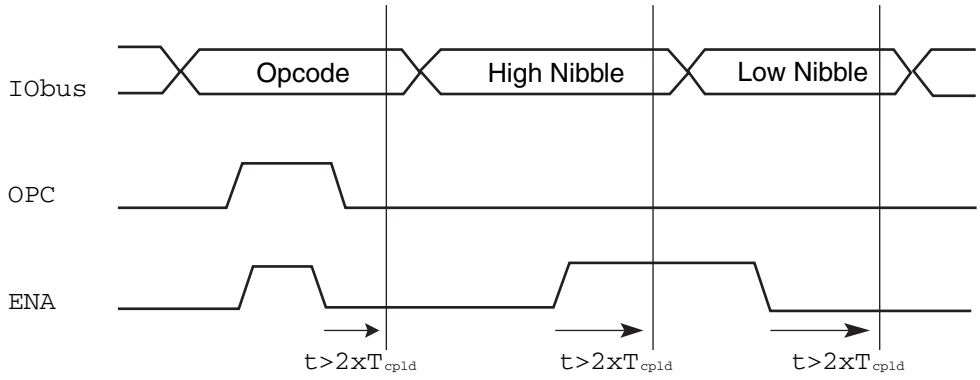
The OPC signal denotes the presence of the opcode command on the IObus. When OPC is asserted high, the data on IObus are decoded in accordance with Table 4-1.

Table 4-1. I/O Expander Opcode Format

Bit 3	0	Read from Port	1	Write to Port
Bit 2	0	Set port as Input	1	Set port as Output
Bits (1:0)	00	Access Port A		
	01	Access Port B		
	10	Access Port C		
	11	Access Port D		

The ENA signal is used to differentiate the two nibbles. When data nibbles are transferred from the MCU to the CPLD, the CPLD latches internally the high-order nibble right after ENA is asserted. Similarly, when ENA is deasserted, the low-order nibble is latched and the whole byte value is output to the port specified in the opcode. Reading an output port returns the value already written to it. Figure 4-1 shows the transaction for an output port.

Figure 4-1. MCU Writing to an Output Port



It should be noted that timing constraints apply to ENA and OPC signals: to avoid potential metastability issues, both are internally buffered for two CPLD clock cycles before being evaluated by the I/O Expander. Consequently, the code running in the MCU needs to account for this and hold IObus signals stable for a minimum of two CPLD clock cycles. In the provided MCU assembly source code, a macro has been defined that provides a delay of one CPLD cycle. This macro should be modified according to the relationship between the CPLD clock and the AVR clock. Please refer to the MCU assembly source code comments for further information.

A similar procedure is followed in the case of read access: after receiving the opcode, the CPLD drives the IObus and sends the high-order data nibble. Then, after the ENA pin is asserted, the low-order nibble is also sent. The low-order nibble is present for one CPLD clock cycle. Figure 4-2 presents the I/O Expander signals when MCU reads data from an input port. An option is provided to extend the duration of the low-order nibble. By asserting OPC together with ENA, the

CPLD will keep driving IObus until OPC is deasserted. This way, the duration of the low-order nibble is extended for as long as the OPC is asserted, as shown in Figure 4-3. Writing to an input port has no effect.

Figure 4-2. MCU Reading an Input Port

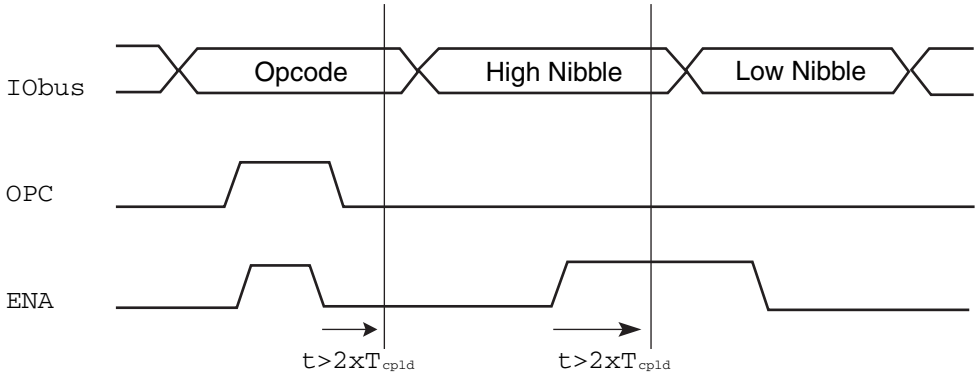
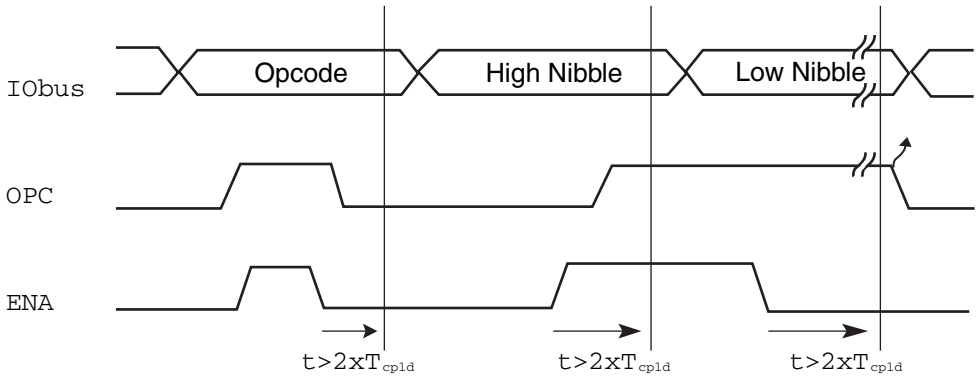


Figure 4-3. MCU Extending IObus Output While Reading an Input Port



Regarding the I/O Expander implementation, the core of the design consists of a state machine with six states. Two states are required for sending/receiving each nibble, while an additional one is necessary for decoding the opcode. More details about the design are provided by the in-line comments that accompany the VHDL and the assembly source code files.

5. Resources Utilization

Two versions of the I/O Expander are provided: one having four bidirectional I/O ports and one with reduced functionality, having one programmable I/O port and one input-only port. An ATF1508 device can be used for the full-functionality version while the smaller ATF1502 is sufficient for the reduced one. Table 5-1 summarizes the required resources for both versions.

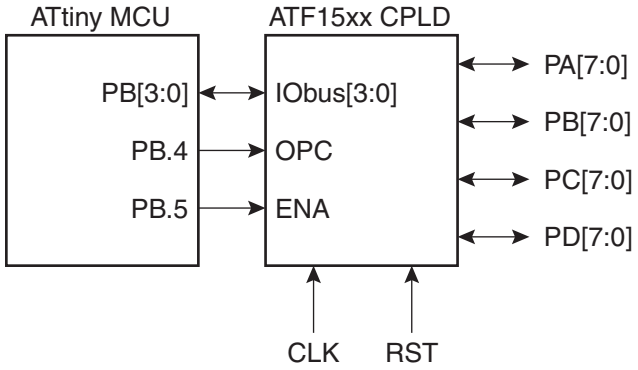
Table 5-1. Resource Utilization of I/O Expander

	Reduced Functionality (ATF1502)	Full Functionality (ATF1508)
I/O Pins	26/32 (81%)	42/64 (65%)
Flip Flops	27/32 (84%)	55/128 (42%)
Logic Cells	30/32 (93%)	66/128 (51%)

6. Application Example

Figure 6-1 shows how the I/O Expander can be used to provide additional ports to an MCU. ATtiny13 AVR MCU, which has only six I/O pins, is used as a test case. With the I/O Expander described herein, up to four I/O ports can be made available to the ATtiny13. Since some of the smaller ATtiny devices do not have an SPI peripheral integrated, this approach of port expansion is the most effective in terms of required memory and execution speed (when compared to a software implementation of an SPI peripheral for example).

Figure 6-1. Application of the I/O Expander for MCU



The provided AVR assembly source file (IOexpT13.asm) increments an internal counter and outputs its value to CPLD_PortA which is configured as output port. Minor modifications are required to use the assembly source with other AVR MCUs. In most cases, the available MCU port may need to be changed as well as interrupt vector table and the stack location (since ATtiny13 does not use SPH register).

Two VHDL source files are provided: the IOexp_full.vhd is the implementation of the full version of the I/O Expander, while the IOexp_red.vhd is a reduced-functionality implementation that can fit in ATF1502 devices.

This design is provided as a reference and should be used only as an example design. The performance and functionality of this design are not guaranteed to meet each user's requirements. To obtain the design files, please contact the ATMEL PLD group via e-mail at pld@atmel.com.

7. Conclusion

CPLDs can provide a flexible low cost solution for implementing extra peripherals that are often missing from off-the-shelf MCUs. For small pin count MCUs (like the ATtiny devices that have only six IO pins), the I/O Expander design described herein can be a perfect match, offering a low cost solution for extra I/O ports. The design can be easily modified and adapted for use with various kinds of end-applications like keypad scanners, LED displays, etc.

8. Technical Support

Submit questions online at: http://www.atmel.com/dyn/products/support.asp

FAQ: http://www.atmel.com/dyn/products/tech_support.asp?Faq=y

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High-Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature