

Summary

The demo board described in this application note is a tool for demonstrating the In-System Programming (ISP) capabilities of the XC9500 CPLD family.

Xilinx Family

XC9500

Introduction

Using EZTag, the ISP download software from Xilinx, you can easily program any XC9500 device while it is attached to a PC board. EZTag allows you to program, verify, erase, and functionally test XC9500 devices. The schematic of the ISP demonstration board, using an XC9536 device, is shown in Figure 1.

Design Description

There are two versions of this demo board as described below. Both versions use a 555 surface-mount timer, with resistor and capacitor values set for 14 Hz operation. This oscillator clocks a simple test design (a Johnson counter) implemented in the XC9536; this counter drives LEDs used to verify operation. The two versions of the demo board are:

- FAE version - The FAE XC9536 ISP Demo Board comes in a plastic case which holds the Printed Circuit Board (PCB) and a 9V battery, 5V regulator, filter capacitor, and the on-off switch.
- Customer version - The Customer XC9536 ISP Demo Board is the same as the FAE version except for the addition of a proto-typing area. Also, the case, battery, 5V regulator, filter capacitor, and the on-off switch are not installed on the customer version.

Both PCBs come with a 44-pin VQFP XC9536 device with two bypass capacitors, (8) LEDs with current limiting resistors, and a header for attaching the JTAG Cable.

Both versions of the ISP Demo Board allow attachment of an external regulated +5V power supply via the pads at J2. If a +5V regulator is installed at location U2 with a 22uF (or larger) filter capacitor at C4, an external DC voltage of 7V to 12V can be applied at location J3. This is where the 9V battery is attached on the FAE version.

The PCB will accept either a DPDT switch or a permanent jumper at location SW1. The switch is used to connect or disconnect an external DC voltage from the +5V regulator.

All pins of the XC9536 device are connected to through-hole pads on the PCB, numbered 1 to 44. Header rows of .025" square posts (on .10" centers) can be installed at

these locations to provide connection points for plug-on jumpers or wire-wrapping.

The proto-typing area on the customer version has 299 holes (13 columns x 23 rows) for attaching additional circuitry. The holes are .038" diameter on .10" centers. Two pair of these holes are connected to +5V and GND along the left side of the prototyping area.

Figure 2 shows the ABEL code used to describe a Johnson Shift Counter; created and implemented with Xilinx XABEL-CPLD software. You can use this reference design as-is, or you can modify the design as required. Figure 3 shows the same design done in VHDL, using the Xilinx Foundation software.

Parts List

The FAE version of the ISP Demo Board comes complete with all parts installed. The Customer version does not include the power supply components which can be purchased from Digi-Key as shown in Table 1.

Table 1: Digi-Key Parts List

Qty.	Description	Ref.	Digi-Key Part No.
1	DPDT Switch, right angle	SW1	EG1909
1	5V, 1A, low dropout reg.	U2	LM2940CT-5.0
1	22uf, 16V, Tantalum cap.	C4	P2040

Digi-Key Corporation is located at 701 Brooks Ave. South, Thief River Falls, MN 56701-0677, Tel: 800-344-4539, Fax: 218-681-3380, (<http://www.digikey.com>).

The customer version of the PCB is designed to fit into a SERPAC plastic case, Model H-65 AC. This case can be purchased from SERPAC, 619 Commercial Ave., Covina, CA 91723, Tel: 818-331-0517, Fax: 818-331-8584 (<http://www.serpac.com>).

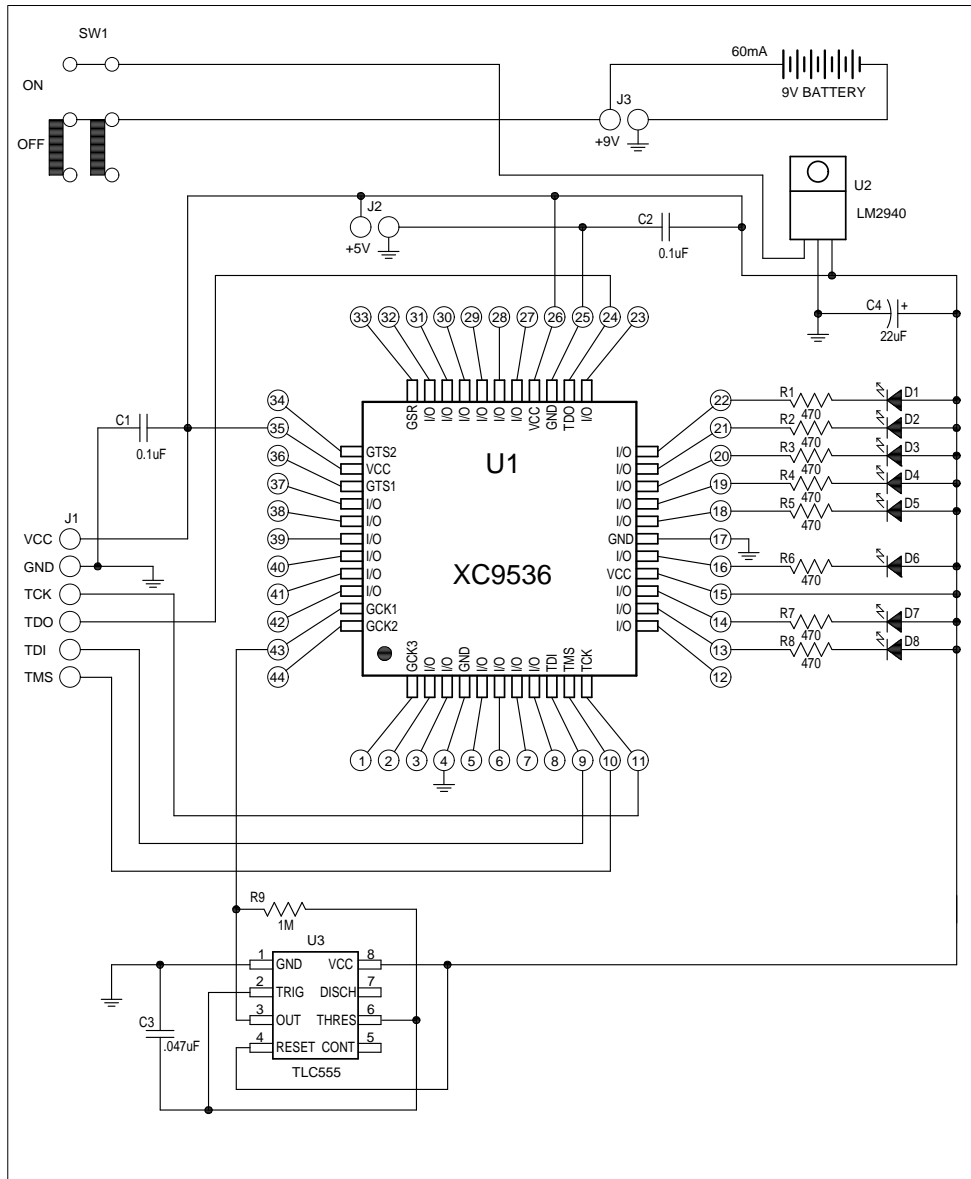


Figure 1: ISP Demo Board Schematic

```

module jcounter
Title 'Johnson Shift Counter'
Declarations
clk PIN 43;           // GCK1 on XC9536 CPLD
!q0 pin 22 istype 'reg'; // LED
!q1 pin 21 istype 'reg'; // LED
!q2 pin 20 istype 'reg'; // LED
!q3 pin 19 istype 'reg'; // LED
!q4 pin 18 istype 'reg'; // LED
!q5 pin 16 istype 'reg'; // LED
!q6 pin 14 istype 'reg'; // LED
!q7 pin 13 istype 'reg'; // LED
q = [q7..q0];         // LED Bus Declaration
Equations
//Johnson Shift Counter description
q.clk = clk;
q0 := !q7; q1 := q0; q2 := q1; q3 := q2;
q4 := q3; q5 := q4; q6 := q5; q7 := q6;
// Test Vectors can be used to functionally verify equation in-system.
test_vectors([ clk ] -> [ q0, q1, q2, q3, q4, q5, q6, q7]);
    [ .C. ] -> [ 0, 1, 1, 1, 1, 1, 1, 1 ];
    [ .C. ] -> [ 0, 0, 1, 1, 1, 1, 1, 1 ];
    [ .C. ] -> [ 0, 0, 0, 1, 1, 1, 1, 1 ];
    [ .C. ] -> [ 0, 0, 0, 0, 1, 1, 1, 1 ];
    [ .C. ] -> [ 0, 0, 0, 0, 0, 1, 1, 1 ];
    [ .C. ] -> [ 0, 0, 0, 0, 0, 0, 1, 1 ];
    [ .C. ] -> [ 0, 0, 0, 0, 0, 0, 0, 1 ];
    [ .C. ] -> [ 0, 0, 0, 0, 0, 0, 0, 0 ];
    [ .C. ] -> [ 1, 0, 0, 0, 0, 0, 0, 0 ];
    [ .C. ] -> [ 1, 1, 0, 0, 0, 0, 0, 0 ];
    [ .C. ] -> [ 1, 1, 1, 0, 0, 0, 0, 0 ];
    [ .C. ] -> [ 1, 1, 1, 1, 0, 0, 0, 0 ];
    [ .C. ] -> [ 1, 1, 1, 1, 1, 0, 0, 0 ];
    [ .C. ] -> [ 1, 1, 1, 1, 1, 1, 0, 0 ];
    [ .C. ] -> [ 1, 1, 1, 1, 1, 1, 1, 0 ];
    [ .C. ] -> [ 1, 1, 1, 1, 1, 1, 1, 1 ];

end jcounter

```

Figure 2: ABEL Code for the Johnson Counter (with test vectors)

```
library IEEE;
use IEEE.std_logic_1164.all;
library metamor;
use metamor.attributes.all;

entity jcounter is
    port (
        clk:in STD_LOGIC;
        Dout:buffer STD_LOGIC_VECTOR (7 downto 0)

    );

    -- Can use attributes to assign pin locations in Foundation VHDL
    attribute pinnum of Dout:signal is "p13,14,16,18,19,20,21,22";
end jcounter;

architecture jcounter_arch of jcounter is
begin

    process (CLK)
    begin
        if CLK'event and CLK='1' then--CLK rising edge
            Dout(7 downto 1) <= Dout(6 downto 0);--Shift register
            Dout(0) <= not Dout(7);--Last bit inverted back into first bit
        end if;
    end process;

end jcounter_arch;
```

Figure 3: VHDL Code for the Johnson Counter