Application Note

January 2005 Document #: **983-0759-001**

Tips for In-System Programming

In-System Programming, also referred to as "on-board" or "in-situ" programming, refers to the process of putting code or data into a programmable semiconductor that is attached to a PCB. This can be done either contacting the test pins with automated test equipment (ATE) or via a board level connector. Shrinking packages, increasing lead counts, and the proliferation of packages such as QFP introduces risks of damaging devices with manual handling. The potential to reduce board scrap and re-work costs, and the flexibility to program later in the production process increase the appeal of in-system programming.

This application note will address some of the hardware considerations to be taken into account when you plan to program devices on the board.

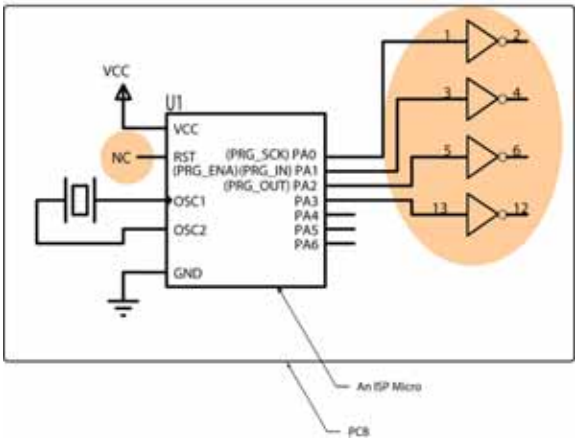
Not all devices support In-System Programming (ISP). See device manufacturers' specifications for information on whether a device can be programmed in-system, and what protocol it supports. There are many protocols and none has become a standard. The most common are: IEEE 1532, JTAG , I2C, and SPI. If the target device manufacturer claims the device supports In-System Programming, it means if *all* conditions for the device to be programmed in-system are met. This involves many elements, and each needs your attention to ensure the device can be programmed. Target device specifications and board design must be addressed to determine any issues. Data I/O experts may be able to provide you with answers to your ISP questions not addressed here.

Guidelines for Supporting ISP

Pin Access Issues

- All pins used for programming must be made available through connectors or test points.
- The pins used for programming must have no other circuitry on the PCB connected to them that prevents the line from being toggled or read.

Design Samples of Pin Access Issues:

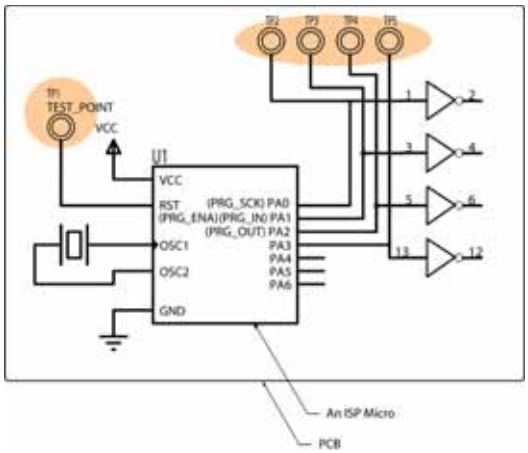


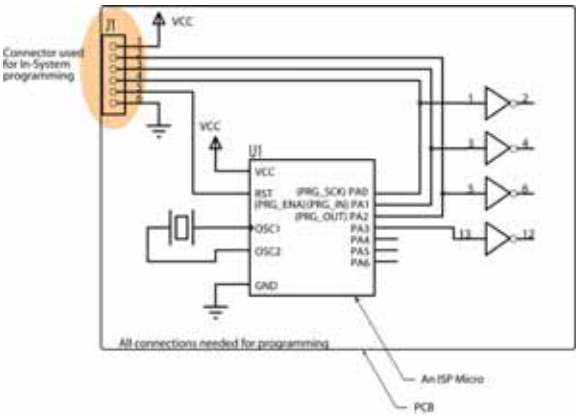
◄ Poor Pin Access Design

Pins used for programming are hard to access. Reset is not connected to anything.
All other pins used for programming have no test points, and may end up having no vias on the PCB for connection to a test fixture.

Better Pin Access Design ►

All Pins used for programming have test points, so access is easy in a test fixture. Without a fixture, access will be difficult. A connector would be better.





◄ Best Pin Access Design

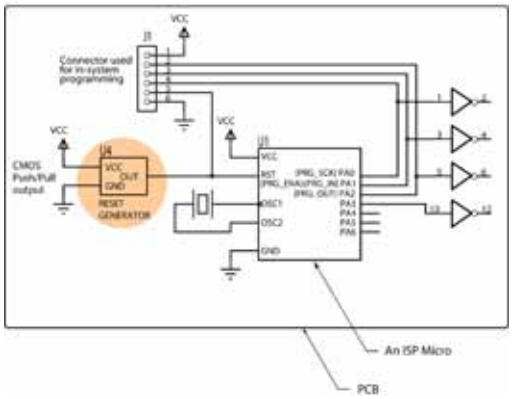
All connections needed for programming go to a connector. This design will work well with or without a test fixture.

Power and Contention Issues

- Power (Vcc) and programming voltage (Vpp), if used, must be present on the board at the proper levels for programming. Power levels are determined by the devices and other connected circuitry. Refer to device data sheets.
- Provisions for varying Vcc to the PCB must be made if the device to be programmed requires a verify cycle with variable voltages.

- If the ISP technique uses the RESET pin, special care must be taken if a board mounted reset generator is present. A reset generator might conflict with the programmer drive signal. This is known as *contention*. More specifically, contention is the minimum time a host must transmit before it can be sure that no other host's packet has collided with its transmission. If the maximum propagation delay from one host to any other is T, then a host that starts to transmit at time t0 may collide with a host that starts just before t0 + T. The first host will not detect the collision until time t0 + 2T.

Design Samples with a Reset Generator:

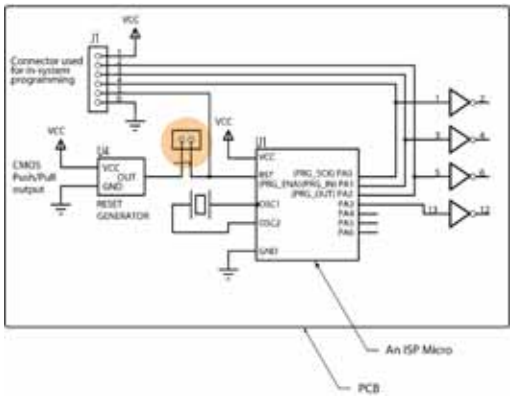


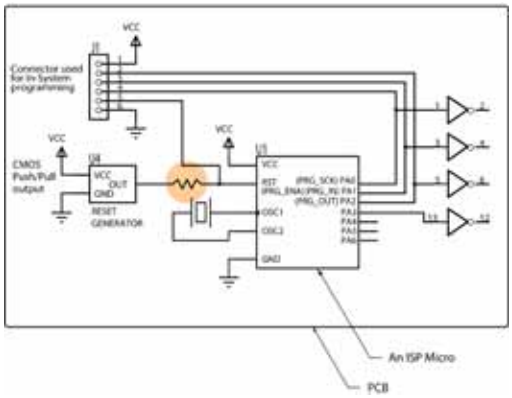
◄ Poor Reset Generator Design

Reset Generator conflicts with reset pin used for programming. In-System programmer will have contention when driving this pin.

Good Reset Generator Design ►

Adding a jumper that is removed for programming eliminates the problem, but is not very convenient.



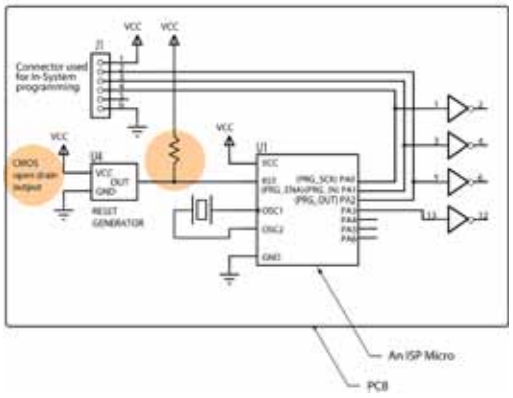


◄ Better Reset Generator Design

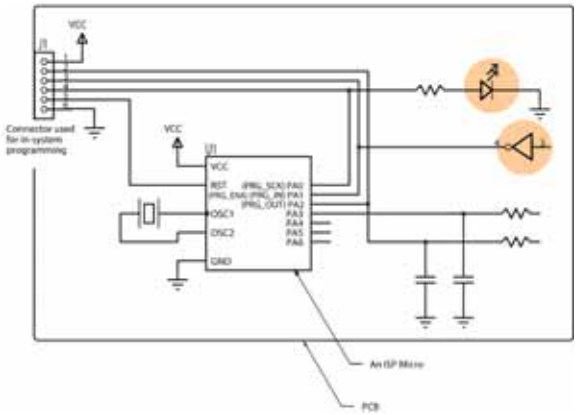
Adding a resistor with a high enough impedance eliminates the problem, but care must be taken to ensure pull up/down resistors, internal to the device, do not conflict with the input voltage level.

Best Reset Generator Design ►

Using a device with an open drain output and a pull-up resistor works best. There are, however, some reset generators that have built in delays when a reset is detected. This delay holds the part in reset longer and may contend with the ISP programmer. Check the specifications.



Design Samples with Contention Issues:

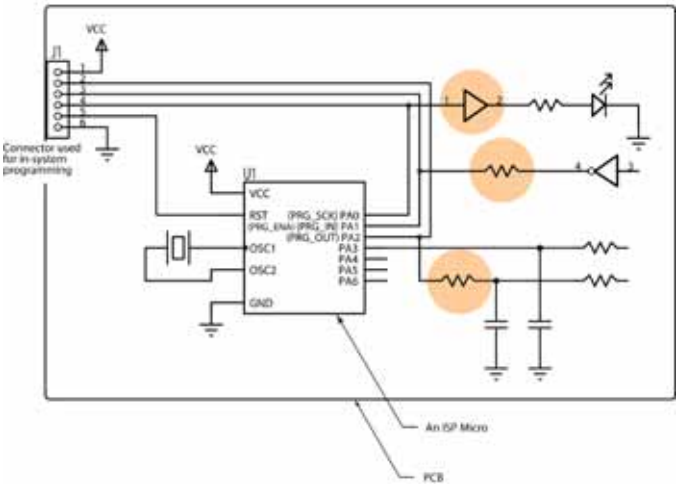


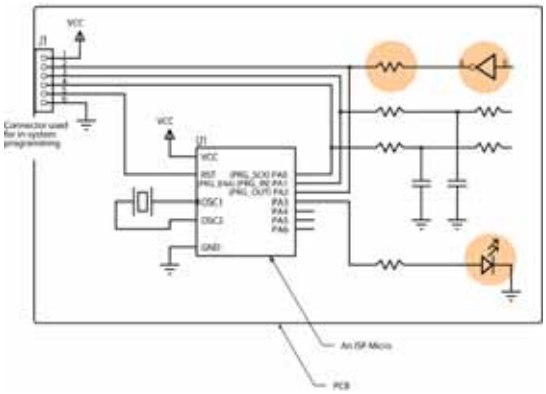
◄ Poor Contention Design

With this design, several problems exist with regards to ISP programming. The LED on PA0 may draw too much current for the ISP programmer's drivers. The Inverter on PA1 conflicts with programming signal PRG_IN. The Capacitor on PA2 will cause potentially slow edges on PRG_OUT, limiting the data throughput on this line.

Better Contention Design ►

Adding series resistors to PA1 and PA2 removes the contention issue on PRG_IN, and eliminates the problem with capacitance on PA2. The buffer added to PA0 keeps the ISP programmer from having to drive the LED current.





◄ Best Contention Design

By moving hardware to other pins of the processor, the buffer on the LED was eliminated at the expense of another resistor.

The best time to think about In-System programming is before the design is complete. Hopefully, this was an easy change.

Other Issues

- Watchdog circuits may trigger unintentionally during programming.
- If a bootloader (also known as *bootstrap loader* or *boot code*) is necessary for programming, it must already be loaded into the device unless the device has a mask ROM bootloader or factory preprogrammed bootloader. In the case of the factory preprogrammed bootloader, if this area of the device is erased or changed, further ISP operation may not be possible. Also, some devices require specific bootloaders that work only with that device or device family.
- If the device contains a security fuse, the ISP capability may be disabled by previous programming.
- If the programming paths have too much capacitance, or the cables from the programmer and target board are too long, ISP may not work.
- Some device manufacturers indicate ISP capabilities in the device name. For example, one manufacturer uses **NIS** to indicate *No ISP* memory modification, and **IS** to indicate that memory can be modified using In-System Programming methods.

Factors Not Directly Related to PCB Layout

Other factors may cause problems that prevent ISP even though every other aspect of the target product works properly and all obstacles mentioned above have been checked. Other factors that can interfere with In-System Programming:

- A noise source such as a switching power supply too close to the target device or programming lines.
- Not enough current available to supply the target device for programming, (Most devices draw higher current during programming).
- Voltages too low for programming. Some devices use a higher voltage for programming than the normal operating voltage for the device. For example, a device may run at 5 V in normal operating conditions. However, the same device may require 9 V or 12 V for programming. Refer to the device Manufacturer's programming specifications to verify the ICC Write and ICC Read parameters for the device.