AN1175
APPLICATION NOTE

In-System Programmability
How Much Money can it Save?

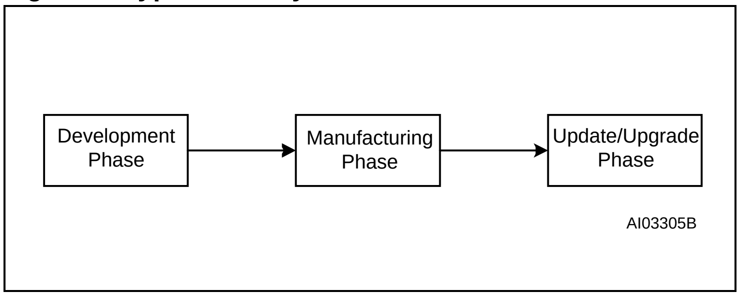
CONTENTS

This document illustrates how integrated-circuit devices that are In-System Programmable can save you money and reduce your time-to-market. We look well beyond the initial purchase price of the device and track the cost of ownership over its lifetime. You will see that ISP can save time and money throughout the life-cycle of a part—from development to manufacturing to updates and upgrades.

In-System Programmable (ISP) devices provide a major technological advantage that can reduce time-to-market and save you money. With ISP, a blank device can be soldered directly to a circuit board and programmed and re-programmed at any time. Because no socket is required, the device need only be handled once in its lifetime.

This document compares the cost of ownership of an ISP part versus the traditional method of proto-typing with a “windowed” EPROM-based part and going into production with a One-Time Programmable (OTP) part. The typical life-cycle of an electronic product is as shown in Figure 1.

Figure 1. Typical Life-cycle of an Electronic Product



Each phase is discussed in a separate section and shows how ISP can save money and reduce your time-to-market.

Until recently, ISP has been used mainly for loading fuse maps into programmable logic devices. However, with today’s increased use of microcontrollers (MCUs), ISP is being used for other applications. In addition to programmable logic, ISP is being used to program firmware into the memory from which the MCU will operate.

ST offers the PSD8XXF family of MCU peripheral devices. This family contains devices that integrate both programmable logic and nonvolatile memory (NVM) on the same die. All PSD8XXF devices are 100% ISP through an IEEE 1149.1 compliant

JTAG channel.

The PSD8XXF family is unique to the industry. It is the only family to offer NVM that is *directly* programmable through the JTAG channel. This is important because many systems are already using JTAG for testing because of its simplicity and low overhead. Now, an entire product can be programmed (including firmware) and tested over a single communication channel. Equally important: no download software has to be written and debugged, and no separate device programmer is required to achieve ISP.

In addition to JTAG, the PSD8XXF also offers other methods to achieve ISP to save costs and reduce time-to-market. Let's look at the details.

ASSUMPTIONS

The following assumptions are used as a basis for the comparison of an ISP product versus a non-ISP product.

1. The lifetime of the end product is five years
2. Over the five year period, 10,000 units will be produced
3. The average cost of each unit is $1000
4. Each unit takes 1 hour to disassemble and 1 hour to reassemble
5. The average pay rate for a technician is $60 per hour (including benefits)
6. The fallout of socketed parts due to mishandling is 33%.

DEVELOPMENT PHASE

In the development phase, ISP can:

- Reduce up-front costs
- Eliminate the need for a socket
- Reduce the overall development time.

Up-Front Costs

There are two up-front costs associated with the traditional development using an EPROM-based windowed part:

- The stand-alone device programmer, ranging in price between $300 and $3000.
- The UV eraser, averaging around $50.

These costs would be incurred on a per-station basis. Note: the cost of the programmer can increase substantially if it has the ability to program more than one part at a time.

With JTAG compliant parts, only a single cable that plugs into a standard PC parallel port is required. This cable can program one part or many parts if they are connected in a chain configuration. The cost of this cable for ST's Flash-based ISP PSDs is $59 US.

Eliminate Sockets

Due to the nature of ISP, design iterations can be handled in-system. Therefore, blank ISP devices can be soldered directly onto the circuit board. This eliminates the need for sockets for ISP devices.

This is not the case with windowed or OTP EPROM parts. A socket is required throughout the life-cycle of the unit. The average 1000-piece price for ten different 52-pin sockets from five different manufacturers was found to be $1.17 per socket. (1000 piece pricing was used because the 10,000 parts would be ordered over a period of five years, 1000 pieces at a time.) This cost is added for each unit made. Therefore, sockets for all 10,000 units would add an extra $11,700 to the overall production cost.

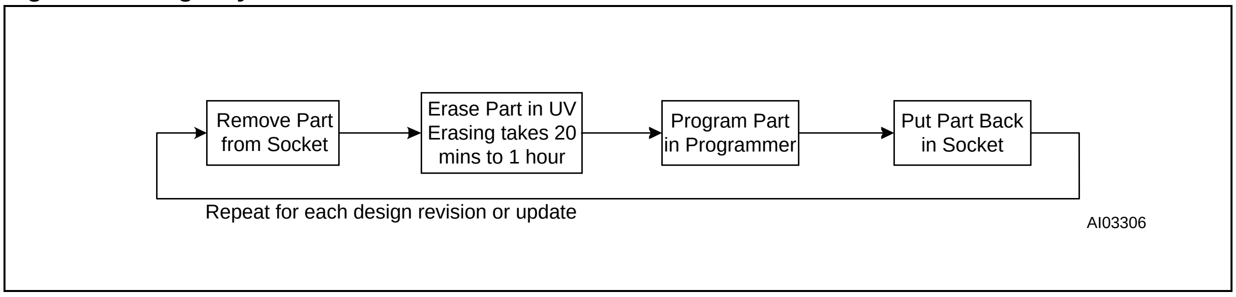
If the design won't allow for a socket because of electrical loading, cost issues, or hostile environments, de-soldering and re-soldering the part each time the contents change would be required. This would not only add tremendous design time, but each circuit board may become unworkable after only a few iterations. This could add tremendous cost to the design, all of which could be avoided with an ISP part.

Development Time

ISP reduces the time-to-market. Since each design environment is different, it is difficult to produce exact numbers regarding time reduction. However, the information provided here should make it clear that time-to-market will be reduced.

The design cycle for a windowed EPROM-based part is shown in Figure 2.

Figure 2. Design Cycle for a Windowed EPROM-Based Part


Remove Part from Socket
Erase Part in UV Erasing takes 20 mins to 1 hour
Program Part in Programmer
Put Part Back in Socket
Repeat for each design revision or update
AI03306

With ISP, simply update the device using a programming cable connected to your circuit board. Notice how three steps are eliminated using this method, saving vast amounts of time during development. Also, since there is no handling of the device with ISP, there will be no fallout due to damaged parts.

The PSD8XXF family saves development and debug time by providing a JTAG channel to program firmware directly into Flash memory. The traditional method of downloading firmware to Flash memory in-system is through a UART channel controlled by the MCU. As such, code to implement a UART download to Flash memory must be developed and debugged before any real application code is written. Download code is difficult to create because you are developing code to deposit code into memory. Until you can get code into memory, you can't debug the code that gets it there.

With the PSD8XXF family, one does *not* have to develop this download code up front. Simply connect the JTAG download cable and program the part. This implies the designer can start writing application code immediately, without having to worry about writing download code.

MANUFACTURING PHASE

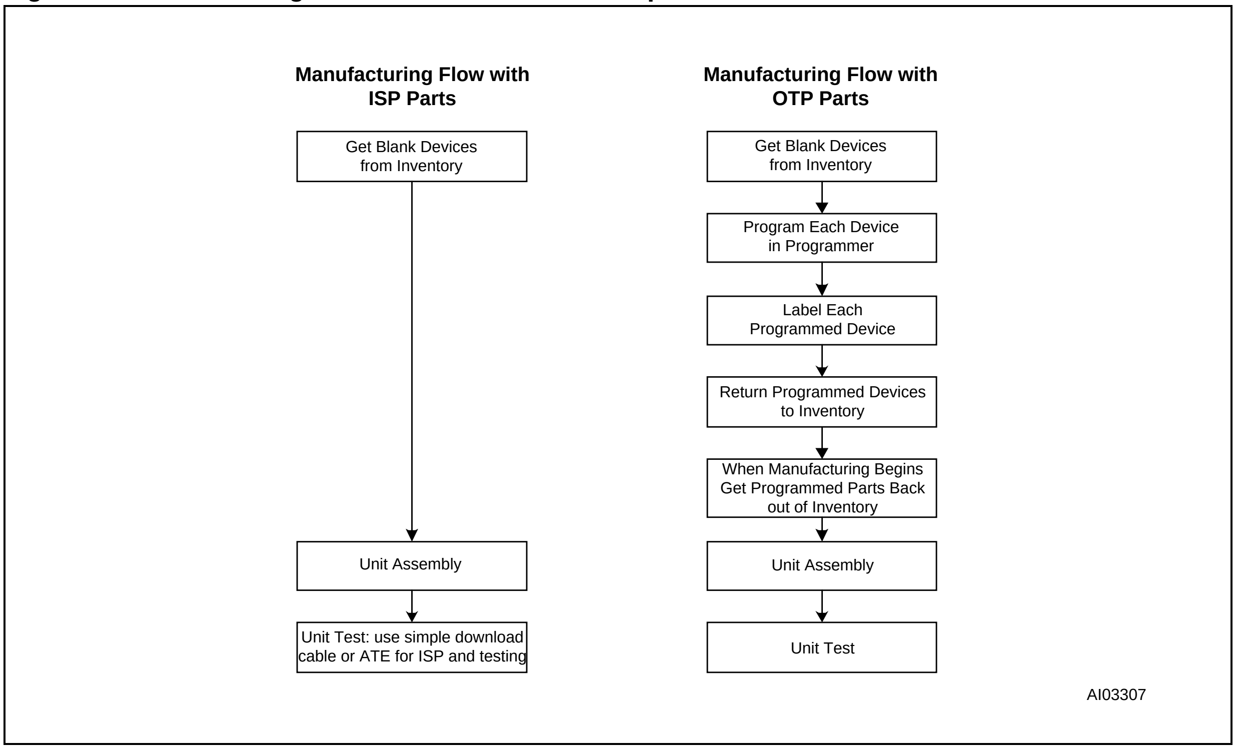
There is a critical factor that will save time and money during manufacturing: ISP devices do not have to be programmed until the board is tested! That fact leads to the following:

- Many manufacturing steps are eliminated
- Excess inventory is eliminated
- Design revisions do not require parts to be wasted
- Quality problems are reduced.

These factors lead to a higher yield in less time.

Figure 3 shows how manufacturing steps are eliminated when manufacturing with ISP versus OTP parts.

Figure 3. Manufacturing Flow with ISP versus OTP parts


Manufacturing Flow with ISP Parts
Get Blank Devices from Inventory
Unit Assembly
Unit Test: use simple download cable or ATE for ISP and testing
Manufacturing Flow with OTP Parts
Get Blank Devices from Inventory
Program Each Device in Programmer
Label Each Programmed Device
Return Programmed Devices to Inventory
When Manufacturing Begins Get Programmed Parts Back out of Inventory
Unit Assembly
Unit Test
AI03307

The first thing to notice with ISP is that programming each part up front with a stand-alone programmer is not required. Programming the part can be delayed until just before the entire assembly is tested.

The PSD8XXF JTAG interface allows the device to be programmed using an inexpensive cable that attaches to a PC parallel port. You can use this cable to download manufacturing test code, then use it again to load the final application code.

Alternately, commercial Automated Test Equipment (ATE) (manufactured by HP, GenRad, and others) can be used to program the PSD8XXF over the JTAG channel. In this case, all other non-PSD8XXF JTAG devices that reside on the printed circuit board may be tested and programmed as well.

As can be seen from the flowchart, manufacturing with ISP parts requires four fewer steps compared to manufacturing with OTP parts.

Manufacturing with ISP Eliminates Excess Inventory

Using ISP during manufacturing can eliminate excess inventory for the following reasons:

1. No separate inventories for separate versions of the software are required.
2. Manufacturing with OTP parts normally requires extra parts to be kept on hand in case there are any last minute software changes.
3. Products can be “made-to-order” as demand dictates.

ISP Eliminates Wasted Parts

Imagine the following scenario: your company is ready to ship 100 units when an unacceptable software bug is discovered. With OTP parts, you would have to throw away the programmed parts and start at the beginning of the manufacturing flow. With ISP, you simply need to reprogram the device with the new software.

Manufacturing with ISP Reduces Quality Problems

Since ISP devices can be soldered directly to the circuit board, no socket will be required, as previously discussed. Because there is no socket involved and the part is not programmed until test time, the following quality problems can be eliminated:

- Bent leads when inserting devices into sockets
- Putting an OTP part with the wrong software version into the socket
- Installing devices into sockets with incorrect orientation
- Installing the wrong device in one of multiple sockets.

This section showed that simply waiting to program the part until the unit is ready for testing reduces manufacturing steps, eliminates excess inventory, and reduces quality problems. This translates into huge cost savings and reduced time-to-market.

UPDATE/UPGRADE PHASE

This is where ISP parts really excel in cost savings over traditional methods.

Updating or upgrading an OTP part can get quite costly and complicated, depending on the path chosen. Some manufacturers simply choose to scrap the product in the field and send out a replacement. Since we assumed each unit costs $1000 to build, it would cost $10,000,000 to update/upgrade all the units. Alternately, the product can be recalled for an update. (See the right-hand column of Figure 2 on the next page.)

For products using a PSD8XXF device, updates and upgrades are very easy to perform. No disassembly of the product is required to load new firmware and configuration into the product. In fact, depending on the product, these upgrades and updates can be done remotely over the Internet or other communication medium.

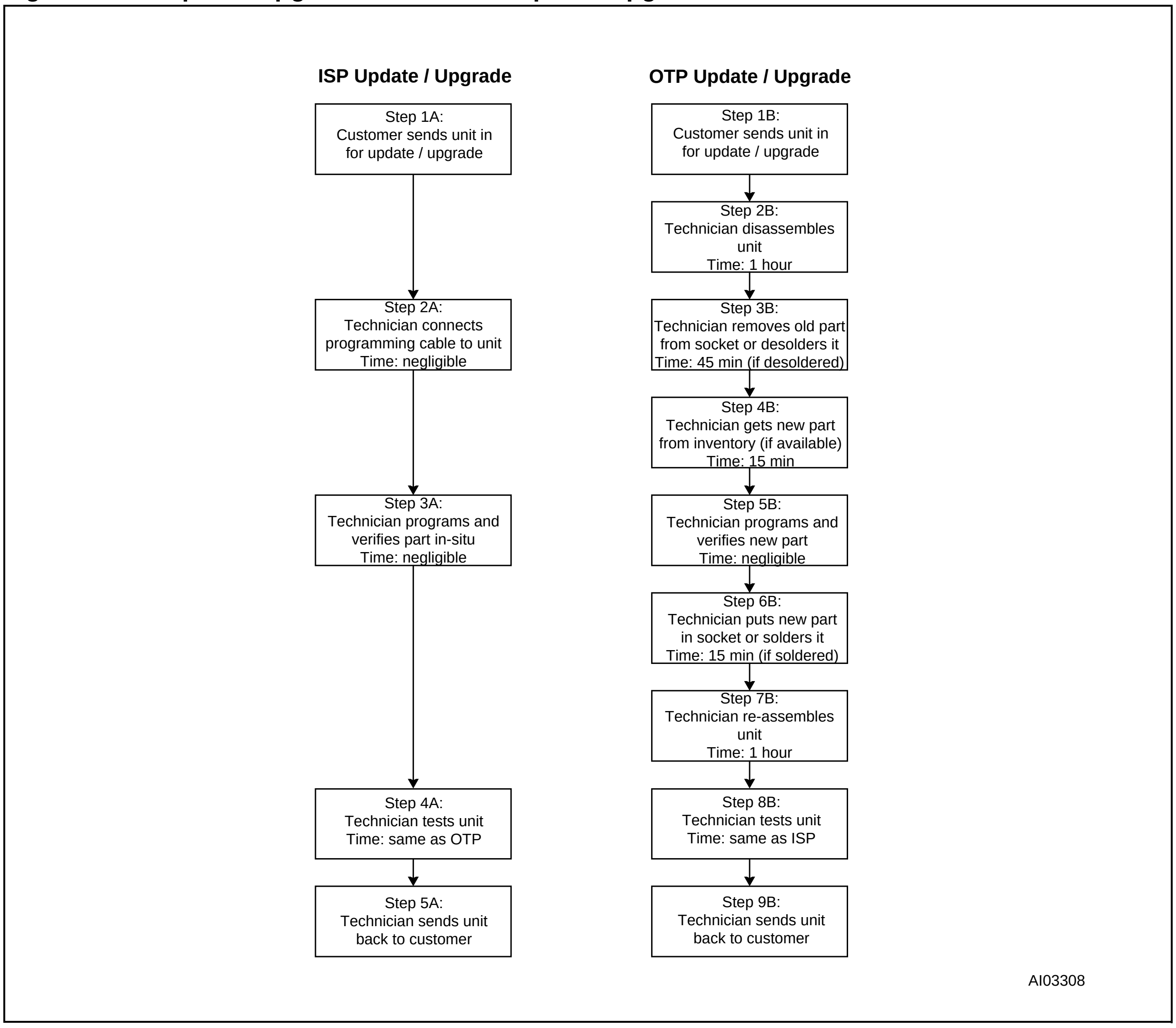
The PSD813F offers two methods to upgrade products in the field. One is JTAG, and the other is by the MCU using the concurrent memory arrays inside the PSD8XXF. When JTAG is used, the MCU does not need to participate in any way. When concurrent memory is used, the MCU operates from one memory while writing (upgrading) the other. This method allows updates and upgrades to be done over any communication channel available to the MCU. (Examples include Ethernet, modem, CAN, RS-232, and RS-485.) Concurrent memory operation allows the MCU to continue some system functions during ISP, which may be critical in some applications. For example, in a production-line controller, it may be crucial for the MCU to continue to control the process during ISP to prevent accident or injury.

With ISP devices, more options are available to handle field updates and upgrades, including:

- Upgrade products remotely over a communication channel.
- Send a service person to the customer site to upgrade with a laptop computer.
- Recall products and upgrade them at the factory (no disassembly needed).

Note that the first option is not available for OTP parts. For simplicity, let’s assume that the product is recalled. Figure 4 compares an ISP part update (left) to an OTP part update (right).

Figure 4. ISP Update/Upgrade versus OTP Update/Upgrade


ISP Update / Upgrade
Step 1A: Customer sends unit in for update / upgrade
Step 2A: Technician connects programming cable to unit Time: negligible
Step 3A: Technician programs and verifies part in-situ Time: negligible
Step 4A: Technician tests unit Time: same as OTP
Step 5A: Technician sends unit back to customer
OTP Update / Upgrade
Step 1B: Customer sends unit in for update / upgrade
Step 2B: Technician disassembles unit Time: 1 hour
Step 3B: Technician removes old part from socket or desolders it Time: 45 min (if desoldered)
Step 4B: Technician gets new part from inventory (if available) Time: 15 min
Step 5B: Technician programs and verifies new part Time: negligible
Step 6B: Technician puts new part in socket or solders it Time: 15 min (if soldered)
Step 7B: Technician re-assembles unit Time: 1 hour
Step 8B: Technician tests unit Time: same as ISP
Step 9B: Technician sends unit back to customer
AI03308

If we compare the times of the two flowcharts, we see that the unit with the OTP part inside will require an *extra* three and a quarter hours of technician work. At $60 per hour, this equates to $195 per unit. If all 10,000 units had to be recalled and updated because of a bug in the software, the total would come out to $1,950,000 extra!

Table 1. Document Revision History

Date	Rev.	Description of Revision
Jun-1999	1.0	Document written (AN062) in the WSI format
03-Jan-2002	1.1	Front page, and back two pages, in ST format, added to the PDF file References to Waferscale, WSI, EasyFLASH and PSDsoft 2000 updated to ST, ST, Flash+PSD and PSDsoft Express
16-Apr-2002	1.2	Document converted to ST format

For current information on PSD products, please consult our pages on the world wide web:

www.st.com/psd

If you have any questions or suggestions concerning the matters raised in this document, please send them to the following electronic mail addresses:

apps.psd@st.com	(for application support)
ask.memory@st.com	(for general enquiries)

Please remember to include your name, company, location, telephone number and fax number.