DediProg Inc.
4F., No.7, Ln. 143, Xinming Rd., Neihu Dist., Taipei City 114, Taiwan

Application Note: AN0103

On-Board SPI programming with DediProg tools: Designer version

Table of content:

1. Introduction 3
1.1. Motherboard BIOS memory specific requirements.....3
1.2. DediProg solutions.....4
1.3. SF100 ISP programmer features overview.....6
1.3.1. SF100 and SF300 Programmer.....7
1.3.2. SF100 Programmer Power Design.....7
1.3.3. SF100 Connection to Motherboard: with ISP Cable 9
2. Design your motherboard for the In-system-programming 10
2.1. What are the benefits for your company?.....10
2.2. In-system-programming methods 11
2.3. Update the Bios memory when “Power ON and Power OFF”...13
2.3.1. Reference Schematic 1.....14
2.3.2. Reference Schematic 2.....15
2.4. Update the Bios memory only when “Power ON” 17
2.4.1. Reference Schematic 3:.....17
2.4.2. Reference Schematic 4.....18
2.5. Update the Bios memory only when chipset releases the SPI bus 19
2.5.1. Reference Schematic 5.....20
2.6. Conclusion.....21
3. Backup Boot Flash design.....22
3.1. What are the benefits for your company?.....22
3.2. Design the motherboard for the BBF method.....23
3.3. Conclusion.....25

1. Introduction

1.1. Motherboard BIOS memory specific requirements

Unlike other Applications using Serial Flash, motherboards have to be compliant with a large amount of different and new peripherals: Bios codes need continuous modifications during development and some update in the field.

An incorrect Bios selection or flawed manipulation, performed usually by inexperienced users, can corrupt the memory content and prevent the computer from being able to reboot: the motherboard would then need to be sent through a repairing channel to change the memory. For these reasons, motherboard makers have to take these constraints into account when designing the boards.

The past solution commonly used was to solder a socket on each motherboard produced, to remove manually the corrupted memory and replace it by a new one with the correct BIOS. The cumulative cost of this method over a number of years, considering the cost of each socket implemented on the millions of motherboards produced can easily lead us to conclude that this solution is not the optimal choice. Additional solutions would be to solder a backup memory on the motherboard to boot from it in case of main Bios corruptions. The cost of this solution is also quite significant.

To reduce motherboard cost, some manufacturers solder the flash memory on board but this adds to the cost and complexity of engineering Bios update or future repairs of the motherboard.

The small size of the Serial Flash increases the difficulty to find cheap and reliable sockets. The socket solution becomes now a risky solution:

- Increase the cost
- Increase the failure rate due to quality problem (SMT contact weakness, oxidation, degradation of the high speed SPI signals, insertion error..)
- Reduce dangerously the number of suppliers for the DIP solution

1.2. DediProg solutions

DediProg team is working closely with motherboard makers, Chipset makers and Serial Flash suppliers to offer the best solutions that fit each of your needs. These solutions are adapted to satisfy each motherboard conditions.

A) In System Programming solution (ISP)

Features: update your Main Bios memories soldered on your motherboard by using our low cost programmer: **SF100** and **SF300.** When connected to the motherboard, the **SF100** programmer can control the Serial Flash to read or update its content.

Advantages:
- Very fast update (10sec / 30 sec)
- Flexible update as it only requires the naked motherboard (convenient for development, production, storage area update or repairing channel)

Requirement:
- Some Chipsets need isolations for their SPI outputs protections.

Fig 1: In System Programming update

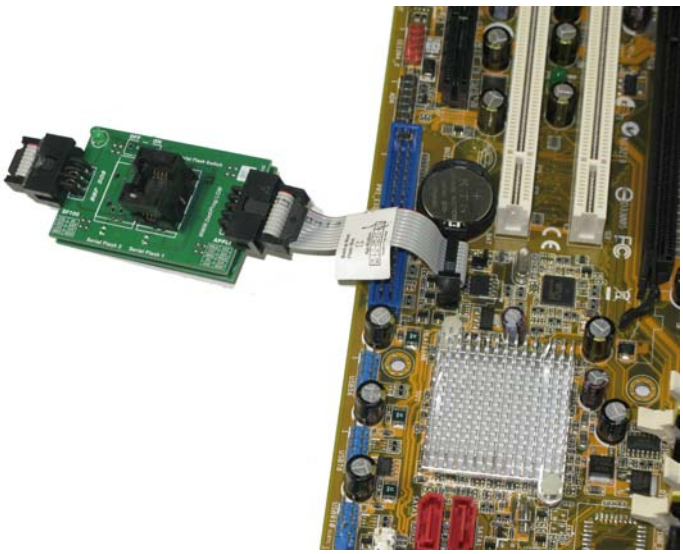
B) Backup Boot Flash solution (BBF)

Features: Our Backup Boot Flash disables the Main bios memory when connected to the motherboard. Then the Chipset can boot from our Backup Boot Flash with the appropriated Bios code and update the main memory by using the Flash utility tool.

Advantage:
- Low Cost
- Work with most of the market motherboard without hardware modification
- If coupled to our SF100 programmer, our Backup Boot Flash can be used as a "Serial Flash Emulator" (convenient for Bios development)

Requirement:
- Need to perform a Boot of the motherboard:
  - ➔ Longer updating time versus ISP method
  - ➔ Requires a complete computer environment for boot (ATX power, processor, monitor..)

Fig 2: Backup Boot Flash method



For more information, refer to our BBF dedicated documents in section 3.

1.3. SF100 ISP programmer features overview

The **SF100 ISP** programmer has been developed and designed to provide a **high performance** for the **lowest cost** that fits to the Bios development needs and solve the repair issues of motherboard applications.

The **SF100 ISP** programmer is dedicated for Serial flash only, to optimize the hardware and software performance and minimize the cost.
Why do you need to pay for memory support you don't need?
With the **SF100 ISP** programmer, your motherboard will not only benefit from the **cost reduction due to the Serial Flash memory** but also from the **cost reduction due to the socket removal**.

Actually, the **SF100 ISP** programmer has been designed to update the Serial Flash soldered directly on Board via an inexpensive Header connector on the SPI bus, removing the need for a socket. The corrupted memory no longer needs to be physically removed and replaced by a new one but simply updated just by connecting the Programmer to the motherboard connector.

The SF100 ISP programmer has been designed by taking into account all the different environments of the motherboard, from the development to the final user and repair channel.

Like:

- Motherboard powered / not powered
- Additional Computer available / not available
- Expert engineers / inexperienced or not technical operators
- Single update (repairing channel) / High throughput (production)

As most of the Bios update needed for development requires partial change in the memory, the SF100 software offers a "Smart update" to Erase, program and verify only the sectors with differences (new Bios versus previous Bios). The "Smart Update" feature reduces significantly the bios update time to few seconds only.

1.3.1. SF100 and SF300 Programmer

The **SF100** programmer can be used in **USB modes**:

USB mode:

The programmer is connected to the host computer via the USB bus.
The memory management is controlled from the DediProg tool.
This mode is very convenient, friendly and gives access to a large amount of features to be used by the experts (design, software, hardware, product, test or quality engineers). It also offers an automatic mode that can easily be used by inexperienced operators.
User can freely select between a friendly Window Graphic User Interface (GUI) or a convenient Dos Command Line.

The **SF300** programmer offers an additional mode, the stand Alone, that can be manually selected by a switch.

Stand Alone mode:

The programmer is a stand-alone device that doesn't need to be connected to a computer. This mode is very convenient to reduce the cost of production line programming (no computer needed) or to execute repetitive actions on memories. It also allows non-technical personnel to easily perform repeated command sequences.
The operations are started by clicking a button on the programmer.
In this mode, a Flash Card is used as a master reference to copy the code to the motherboard memory.

1.3.2. SF100 Programmer Power Design

A convenient, flexible and low cost power management designed for motherboard:

The BIOS memory soldered on the motherboard can be either powered **by the programmer** (if motherboard is OFF during the update) or powered **by the motherboard** (if motherboard is ON during the update).

A smart management system to protect the motherboard:

Our programmer has been optimised to provide safe management of the memory supply and signals to limit the impact of the ISP method on the application.

- The memory Vcc is managed safely by the programmer to supply the target motherboard memory **only when needed** and is switched OFF when operations are completed.

- The SPI signals are also applied **only when needed** and are switched in High Impedance when operations are completed.

- All the ISP pins are protected with ESD high performance protection devices to discharge the Electrostatics charge before the connection to protect the application.

Advantages:

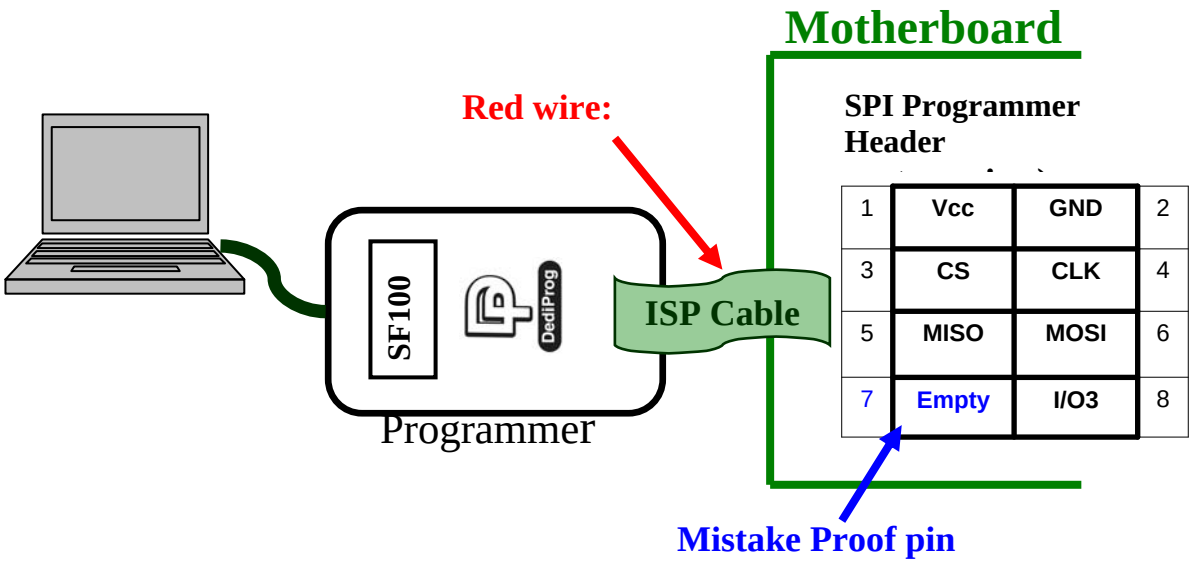
- The programmer is plugged on the application board with Vcc OFF and SPI signals have High Impedance to block inrush currents.

- Motherboard can work with the programmer connected without any functional interference (helpful for Bios development).

1.3.3. SF100 Connection to Motherboard: with ISP Cable

The motherboard requires a 4X2 pin Header straight type with 2.54mm pitch. It is used for the SF100 programmer to control the application memory, and if necessary: power the memory and reset the application chipset. The ISP cable will be wired accordingly.

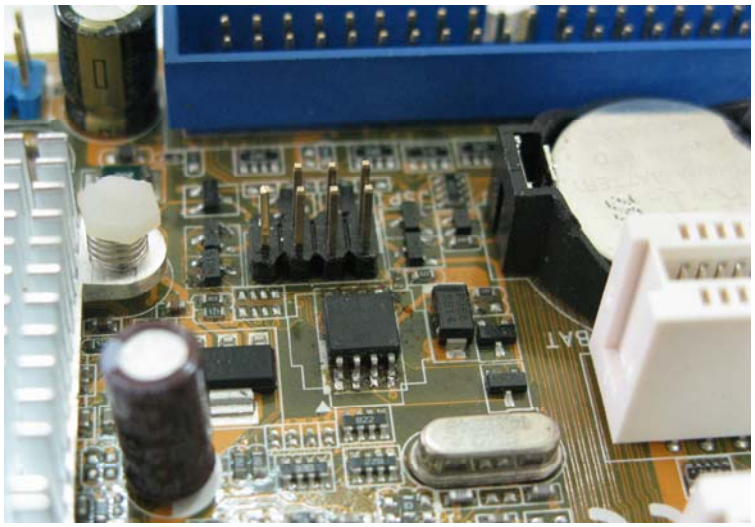
Figure 3: DediProg SF100 Programmer to Motherboard with standard connection



Tab 1: Description of the signals:

Pin	Name of signal	Description
1, 2	Vcc, Gnd	Vcc supplied from the programmer to the Serial Flash
3, 4, 5, 6	CS1, CLK, MISO, MOSI	SPI signals
5	Vcc	Vcc is used to supply the application memory
8	IO3	used to reset the Chipset or switch off the Mosfet
7	Mistake proof pin	Prevent from wrong connection

Figure 4: Motherboard ISP connector (not standard)



2. Design your motherboard for the In-system-programming

2.1. What are the benefits for your company?

After investing in a few low cost SF100 programmers for Bios development, production lines and repairing channels, your company will benefit from the following advantages:

- **Large COST REDUCTION on millions of motherboards over a long period of time by removing all the sockets or backup memories from the board.**
- **High performance:** DediProg programmers are dedicated and optimized for Serial Flash speed.
- **High flexibility of code update:** Bios development, Program in the production line, update before shipment with the last Bios version, on field storage area or on repairing channel.
- **Improve Reliability:** No problems from faulty mechanical contacts due to the socket or problem due to human manipulation of the memory.
- **Time saving:** No need to manipulate the memory or boot on back-up memory, just connect the programmer.
- **Perform a short Failure analysis:** to generate statistic reports on corruptions and take corrective actions to improve your product reliability.

The ISP (In System Programming) is particularly appreciated for its update **flexibility and performance**:

- **In development:**
  - **Easy** to update or read the memory on board (no need to unsolder parts).
  - **Fast update** thanks to the Smart update feature
- **In production:**
  - **Easy** to implement and no technical expertise required
  - **Safer:** programming after the memory manipulation (soldering, test)
  - **Flexible**: update at the last time in production line or in storage area with the last code revision or according to the last ordering
  - **Performance**: the DediProg ISP method uses a Fast and direct connection to the memory via the SPI bus optimised for the Serial flash speed (not limited by a slow interface).
  - SF100 can be used for **ICT programming** and can be controlled by tester using our CML or .dll.
  - SF100 can be used for **ISP mass production programming** by using our production software (Multi-Programmer).

- **In Field or repairing channel:**
  - **Easy** and Fast Bios update in case of Bios corruption.
  - **No technical expertise required**: Click or press button
  - **Time saving**: just connect the programmer with the motherboard OFF
  - **Perform a failure analysis before updating the Bios memory:** to generate statistic reports and to take corrective actions to decrease failure rates.
  - **Improve motherboard reliability**: no problem of contact on socket or human manipulation so finally, less return on repairing.

2.2. In-system-programming methods

When considering the methods to be selected for Bios update, the R&D engineers will have to take into consideration the need from their Bios team requirements but also from all the parties that will be finally involved in the motherboard development, manufacturing, programming and repairing. The choice implemented will then have to be clear enough for all the different actors despite the different motherboard models or Chipset suppliers.

DediProg team will help you to have a clear overview about all the solutions according to the chipset suppliers and select the optimized one. Our team will also help you to generate one document where all updating steps will be described simply, helping non expert or non technical employees to perform the bios update according to the models. Please download the application note **AN0106 "On board SPI programming with DediProg tools: End user version"** and contact us for adaptation to your portfolio.

For your Motherboard Design, we have categorized three different updating methods. Designers can select one of them which fit the most of your needs according to the chipset capabilities:

Important: In this document, “Chipset” includes all the application controllers who drive the Serial Flash: Southbridge, Super I/O, Embedded Controller etc..

Chipset capabilities:

Condition 1: If not powered, the Chipset is tolerant to the SPI signals on its SPI IO through 100 Ohm Serial Resistor.

Condition 2: SPI outputs are in High Impedance when the Chipset is reset.

Condition 3: When motherboard is in Stand By mode (supplied but not turned ON), the chipset and memories are supplied and the chipset release the SPI bus in high Impedance.

Condition 4: When motherboard has finished booting, the chipset release the SPI bus in high Impedance.

Updating Methods:

1) Update the Bios memory when "Power ON and power OFF"

This solution offers to the final user the same updating method what ever the hardware, chipset or motherboard conditions.

Method Name: "Universal ON/OFF"

If the chipset fulfill the **"Condition 1 and 2"**

➔ Designers can implement the **"schematic 1**"

If the chipset doesn't fulfill the **"Condition 1 and 2"**

➔ Designers can implement the **"schematic 2"**

2) Update the Bios memory only when "Power ON"

This solution offers to the final user the same updating method what ever the hardware, chipset but only with motherboard turned ON.

Method Name: "Universal ON"

If the chipset fulfill the **"Condition 2"**

➔ Designers can implement the **"schematic 3**"

If the chipset doesn't fulfill the **"Condition 2"**

➔ Designers can implement the **"schematic 4"**

3) Update the Bios memory only when Chipset releases the SPI bus

This solution offers to the final user the same updating method but requires from him to setup the motherboard in a specific updating mode.

Method Name: "Universal HZ"

If the chipset and motherboard fulfill the **"Condition 3"**

➔ Designers can implement the **"schematic 5**"

If the chipset doesn't fulfill the **"Condition 3"**

➔ Designers can implement the **"schematic 2"** if memory is not supplied

➔ Designers can implement the **"schematic 4"** if memory is supplied

If the chipset fulfill the **"Condition 4"**

➔ Designers can implement the **"schematic 5**"

If the chipset doesn't fulfill the **"Condition 4"**

➔ Designers can implement the **"schematic 4"** if memory is supplied

Designers can select one of these three methods to be applied on all their motherboards projects and adapt the hardware schematics according to the chipset capabilities. In this case, the Bios memories can be updated exactly with the same method for all the different projects.

2.3. Update the Bios memory when "Power ON and Power OFF"

Method name: "Universal ON/OFF"

The memory can be updated with the same method what ever the Chipset suppliers or motherboard conditions.

These designs are recommended by DediProg as it will offer a simple, flexible and safe method to update all the motherboards with the same tools and by the same way. Same method means same way to update the Bios at the user point of view but the design can be different according to the chipset used.

This is also the most convenient way to program the memory in production line or update the corrupted content in repairing channel or storage area. Actually, when the board is received, the operators just have to connect the programmer on the motherboard connector and start the update on the naked motherboard. Fast, simple and convenient, this method does not require to connect the motherboard to ATX power supply or to connect peripherals to perform a boot.

Update with motherboard not supplied (OFF):

The **SF100** programmer has been developed and designed to update the Bios memory with the Motherboard OFF (not supplied). The programmer can supply the bios memory via the pin Vcc of the ISP programmer but in this case, an isolation component (diode or Mosfet) must be inserted on the Vcc line to avoid supplying the entire motherboard with the 3V.

- **With a Diode:**
  The memory will be supplied through the diode when the motherboard is ON. The diode must be selected with a low threshold drop down to keep the serial Flash supply compliant with the memory supplier specification (Often 2.7V/3.6V). When the motherboard is OFF and memory supplied by the programmer, the diode will prevent the rest of the motherboard to be supplied with the programmer Vcc.

- **With a N Mosfet:**
  When the motherboard is supplied, the MOSFET is ON. The Serial Flash memory is then powered by the motherboard.
  When the motherboard is not powered, the MOSFET is OFF. The motherboard is then protected from the programmer Vcc.
  When compared to the Diode solution, the Mosfet isolation has the advantage to reduce the drop down of the memory Vcc when motherboard is ON thanks to it is small Ron value. It is usually recommended to drive it with the highest voltage (+5V/12V) to reduce the Ron.

  Remark: *The Mosfet must be connected carefully according to the parasitic diode effect when not powered. The diode effect must prevent the current leakage from the ISP Vcc to the Motherboard (contact our support team for more information).*

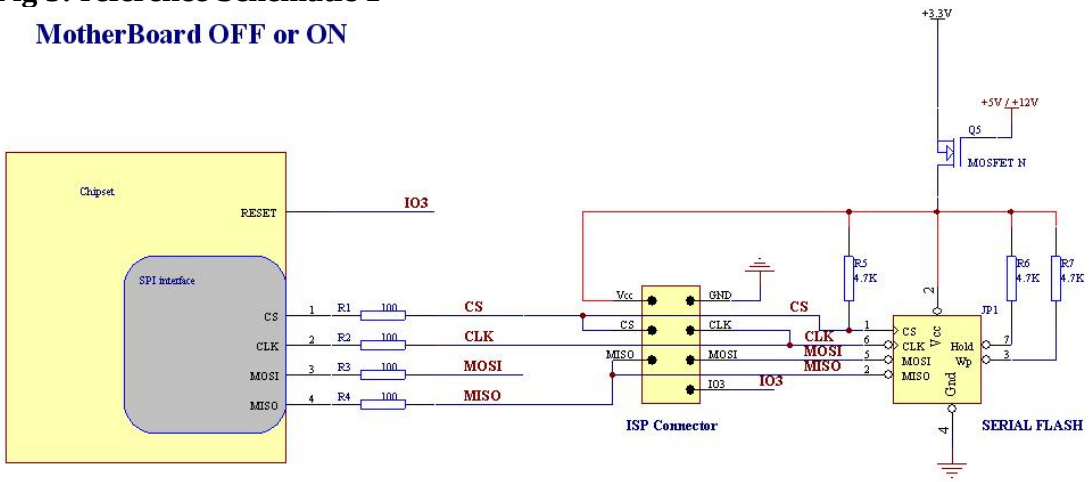
Update with motherboard supplied (ON):
The IO3 signal from the **SF100** programmer is used to isolate or reset the Chipset and avoid conflict on the SPI bus during the update.

2.3.1. Reference Schematic 1

Chipset requirement: Chipset must fulfill **conditions 1 and 2**

- **Condition 1:** If not powered, the Chipset is tolerant to the SPI signals on its SPI IO through 100 Ohm Serial Resistor.
- **Condition 2:** SPI outputs are in High Impedance when the Chipset is reset.

Fig 5: reference Schematic 1



a) Updating conditions: All
- Motherboard OFF (naked motherboard not supplied)
- Motherboard ON

b) How does it work?
- **Motherboard OFF:** As the Chipset is tolerant to the SPI signal on its SPI IO even if not supplied, the programmer can control the SPI bus to update the serial Flash without damaging the Chipset. The Serial resistors on the SPI bus will also help to limit the current injected in the Chipset IO. In this mode, the programmer will supply the serial Flash but not the motherboard thanks to the Mosfet isolation on the Vcc line (Q5).

- **Motherboard ON:** The programmer has been designed to be transparent for the motherboard even if connected. When the update is requested by the user, the programmer drives automatically the IO3 signal low to reset the Chipset and switch its SPI outputs on High impedance. Then the programmer can drive the SPI bus safely to update the Serial Flash. After update, the programmer switches its outputs in High impedance and release the Reset signal so the motherboard can boot on the new Bios version.

c) Hardware Requirements:
- One N Mosfet
- Four Serial resistors (the SPI bus parasitic capacitance must be minimized by placing the Chipset, Serial Flash and ISP connector as close as possible)
- One pin header connector (can be 2.54mm or 1.27mm)

Remark: The Serial resistors (R1 to R4) are also useful to filter the under and overshoot of the fast SPI signals. The CS pull up resistor (R5) deselects the memory when chipset does not drive the bus and protect the memory from noise.

2.3.2. Reference Schematic 2

Chipset requirement:

➔ no conditions required (work with all the chipsets)

For Cost reduction reasons, DediProg recommended to use the **schematic 1** if the Chipset is compliant with the **"condition 1 and 2"**. If not, designers can implement the **reference schematic 2**.

This schematic is compliant with all South bridges such as **Intel ICH7, ICH8, ICH9, etc..**

As these Chipsets are not tolerant to SPI signal when Chipset is not supplied (High current injected) or because the SPI outputs are not switched in High impedance when the Chipset is reset, designers must implement some isolations on the SPI bus to protect the Chipset during the update with motherboard OFF or avoid any conflict with the programmer during the update with motherboard ON.

Fig 6: Isolations needed when SPI signals from an external programmer

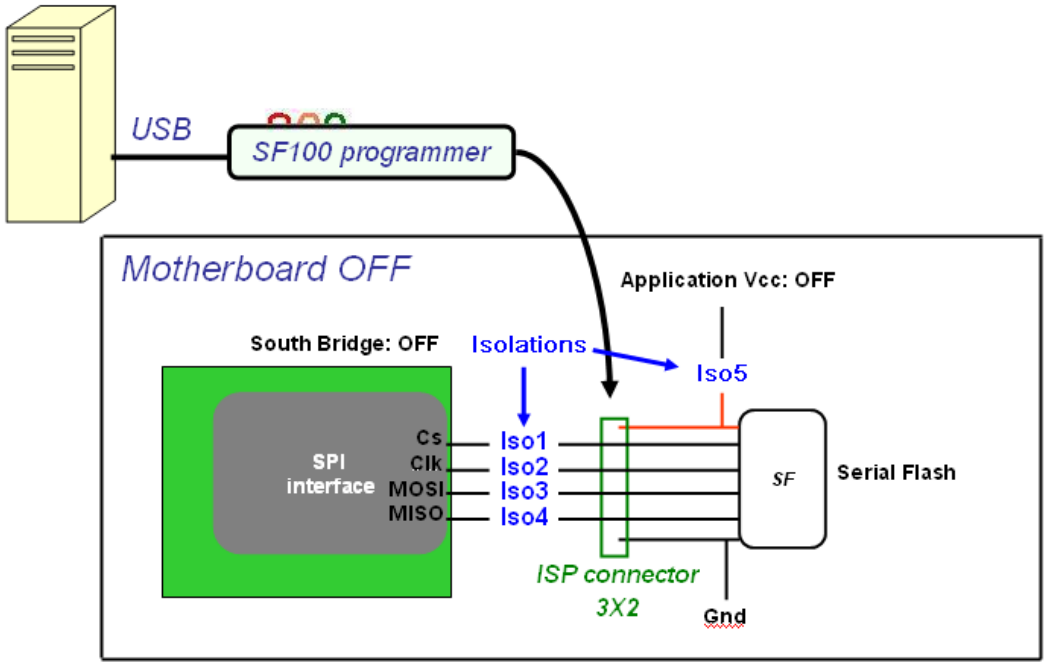
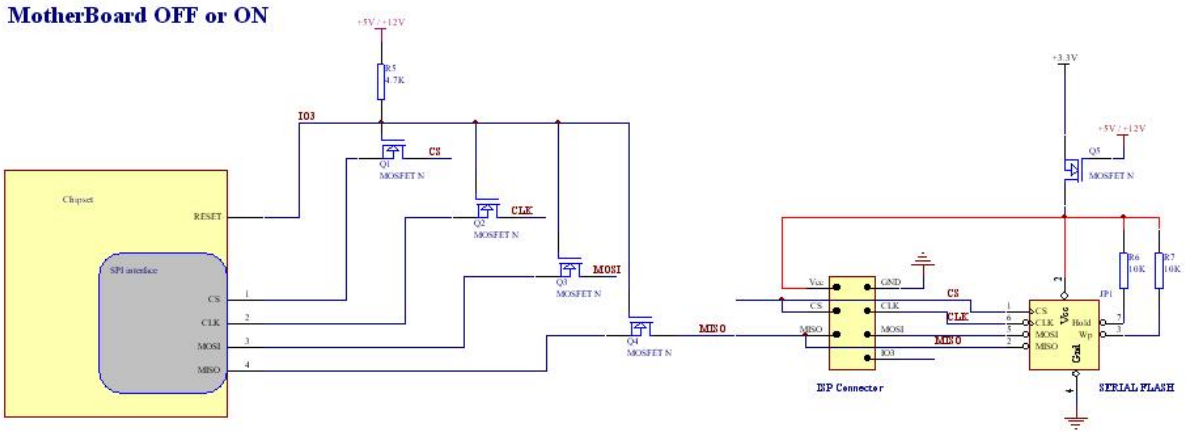
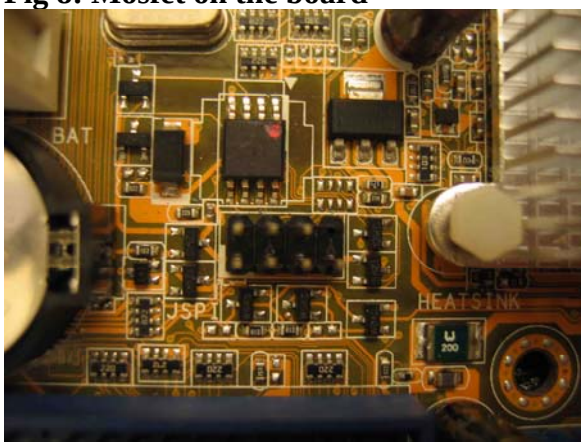


Fig 7: Reference Schematic 2



Remark: *The Mosfet must be connected carefully according to the parasitic diode effect when not supplied. The diode effect must prevent the current leakage from the ISP connector to the Chipset.*

Fig 8: Mosfet on the board



a) Updating conditions: All
- Motherboard OFF (naked motherboard not supplied)
- Motherboard ON

b) How does it work?
- **Motherboard OFF:** When the Motherboard is not supplied, the MOSFET are open and ensure a perfect isolation for the programmer. In this mode, the programmer will supply the serial Flash but not the motherboard thanks to the Mosfet isolation on the Vcc line.

- **Motherboard ON:** When the motherboard is supplied, the MOSFET are closed and ensure a perfect communication between the Chipset and the Serial Flash (small Ron if driven by 5V or 12V). The programmer has been designed to be transparent for the motherboard even if connected. When the update is requested by the user, the programmer drives automatically the IO3 signal low to switch OFF and open the MOSFET. Then the programmer can drive the SPI bus safely to update the Serial Flash as they are isolated from the Chipset. After update, the programmer switches its outputs in High impedance and release the IO3 signal so the MOSFET are closed and the Chipset can boot on the new Bios version.

c) Hardware requirements:
- Five N Mosfet
- One Serial resistor
- One pin header connector (can be 2.54mm or 1.27mm)

Conclusion:
If you implement the **schematic 1** when the Chipset fulfill the **Condition 1 and 2** and the **schematic 2** when the Chipset doesn't fulfill the **Condition 1 and 2** then all your motherboards will be compliant with **our "Universal ON/OFF" method** and will be optimized for cost reduction. The users (Bios engineer, R&D, operators in production or repairing) will be able to implement a unique and simple way to update the bios what ever the motherboard model, the Chipset used or the Motherboard conditions.

2.4. Update the Bios memory only when "Power ON"

Method name: "Universal ON"
The memory can be updated with the same method what ever the Chipset suppliers but only with the motherboard supplied (ON).

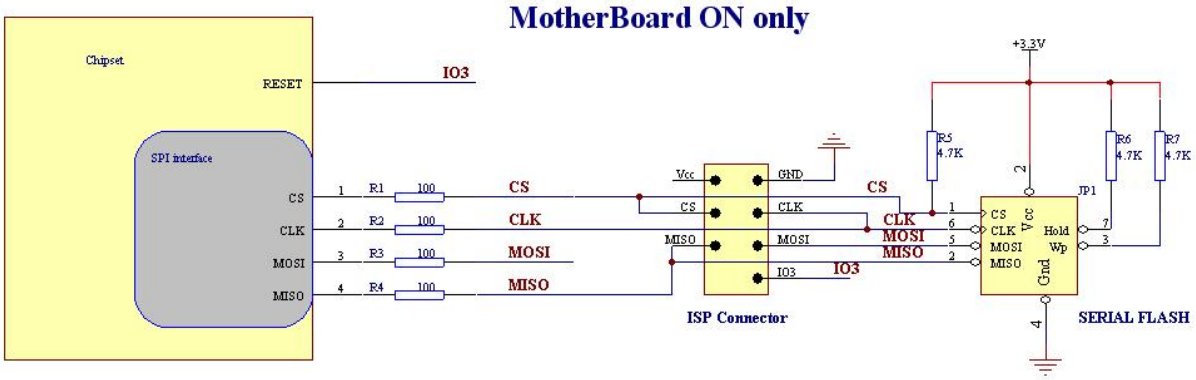
These designs offer a simple, flexible and safe method to update all the motherboards with the same tools and by the same way (motherboard ON).
Same method means same way to update the Bios at the user point of view but the design can be different according to the chipset used.

2.4.1. Reference Schematic 3:

Chipset condition:
- **Condition 2:** SPI outputs are in High Impedance when the Chipset is reset.

This schematic 3 is a simplification of the schematic 1 (fig 5) limited to the update with motherboard ON only to reduce the number of MOSFET.

Fig 9: Reference Schematic 3



a) Updating conditions:
- Motherboard ON only

b) How does it work?
In this case, the Serial flash is supplied by the motherboard during the update, so the Vcc doesn't need to be connected to the programmer and the Vcc Mosfet isolation is not needed. The IO3 resets the Chipset during the memory update to release the SPI bus in High Impedance.

c) Hardware Requirements:
- Four Serial resistors (the SPI bus parasitic capacitance must be minimized by placing the Chipset, Serial Flash and ISP connector as close as possible)
- One pin header connector (can be 2.54mm or 1.27mm)

Remark: The Serial resistors (R1 to R4) are also useful to filter the under and overshoot of the fast SPI signals. The CS pull up resistor (R5) deselects the memory when chipset does not drive the bus and protect it from noise.

2.4.2. Reference Schematic 4

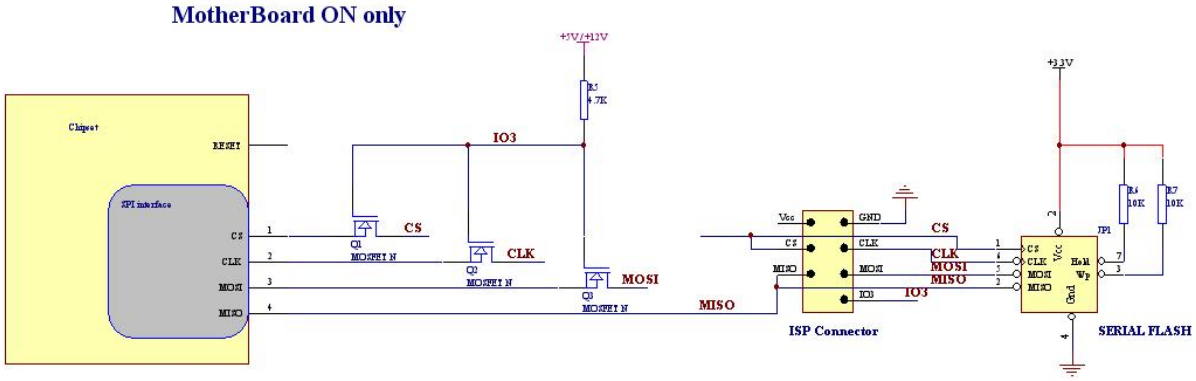
Chipset condition:

➔ no conditions required (work with all the chipsets)

This schematic 4 is a simplification of the schematic 2 (fig 7) limited to the update with motherboard ON only to reduce the number of MOSFET. It is applicable to all the Chipset suppliers including Intel ICH7, ICH8, ICH9..

For Cost reduction reasons, DediProg recommended to use the **schematic 3** if the Chipset is compliant with the **"condition 2"**. If not, designers can implement the **reference schematic 4**.

Fig 10: Reference Schematic 4



a) Updating conditions:
- Motherboard ON only

b) How does it work?
In this case, the Serial flash is supplied by the motherboard during the update, so the Vcc doesn't need to be connected to the programmer and the Vcc Mosfet isolation is not needed. Three Mosfets are needed to isolate the Chipset outputs (CS, CLK and MOSI) during the serial flash update to avoid conflict with the programmer. The IO3 is needed to switch the MOSFET OFF. The MISO signal doesn't need to be isolated as it is a Soutbridge input (no possible conflict).

Remark: on the schematic of figure 8, the MISO Mosfet (Q4) is needed to protect the Chipset when update is performed with motherboard OFF (Chipset not supplied) and avoid current injection in the Chipset input buffer.

c) Hardware requirements:
- Three N Mosfet
- One Serial resistor
- One pin header connector (can be 2.54mm or 1.27mm)

Remark: *The Mosfet must be connected carefully according to the parasitic diode effect when not supplied. The diode effect must prevent the current leakage from the ISP connector to the Chipset.*

Conclusion:
If you implement the **schematic 3** when the Chipset fulfill the **Condition 2** and the **schematic 4** when the Chipset doesn't fulfill the **Condition 2** then all your motherboards will be compliant with **our "Universal ON" method** and will be optimized for cost reduction. The users (Bios engineer, R&D, operators in production or repairing) will be able to implement a unique and simple way to update the bios what ever the motherboard model, the Chipset used but only with the Motherboard ON.

2.5. Update the Bios memory only when chipset releases the SPI bus

Method name: "Universal HZ"
Designers can decide to select a specific time when the Chipset releases automatically the bus in High Impedance to perform the update.

- Chipset condition 3: When motherboard is in Stand By mode (supplied but not turned ON), the chipset and memories are supplied and the chipset release the SPI bus in high Impedance.

Remark: Designers need to be sure that in Stand By mode, the motherboard will supply the Chipset and the Serial Flash memory (3.3V) and that the Chipset releases the SPI bus in high impedance.

- Chipset condition 4: When motherboard has finished booting, the chipset release the SPI bus in high Impedance.

Remark: Designers need to be sure that after have booted, the Chipset will not try to access the serial Flash during the ISP update

In this case, DediProg recommends to provide a clear dedicated document where all the methods are clearly detailed according to the motherboard models. This document (motherboard maker specific) will then be useful for the final user that will attempt to update the Bios memory in these different conditions.

Download the Application **AN0106 "On-Board SPI programming with DediProg tools: End user version"** and contact our DediProg technical team to adapt it to your specific motherboard portfolio and update conditions.

If the chipset and motherboard fulfill the "Condition 3"

➔ Designers can implement the **"schematic 5**" and user can update the memory during the motherboard Stand by mode.

If the chipset doesn't fulfill the "Condition 3"

➔ Designers can implement the **"schematic 2"** if memory is not supplied

➔ Designers can implement the **"schematic 4"** if memory is supplied

If the chipset fulfill the "Condition 4"

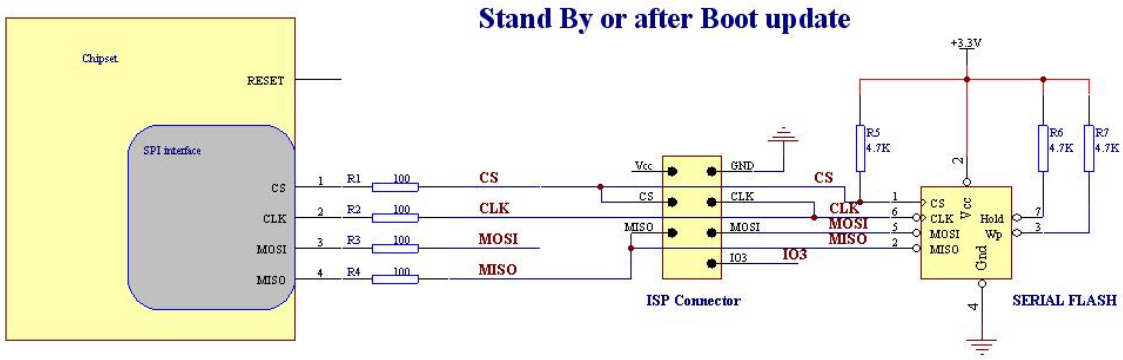
➔ Designers can implement the **"schematic 5**" and user can update the memory after the motherboard boot.

If the chipset doesn't fulfill the "Condition 4"

➔ Designers can implement the **"schematic 4"** if memory is supplied

2.5.1. Reference Schematic 5

Fig 11: Schematic 5



a) Updating conditions:

- Motherboard supplied but update only during Stand By mode or after boot.

b) How does it work?

We select the appropriate time when the Chipset is in waiting mode so, does not try to access the memory and does not drive the SPI bus (SPI is in High Impedance).

c) Hardware requirements:
- Four Serial resistors (the SPI bus parasitic capacitance must be minimized by placing the Chipset, Serial Flash and ISP connector as close as possible)
- One pin header connector (can be 2.54mm or 1.27mm)

Remark: The Serial resistors (R1 to R4) are also useful to filter the under and overshoot of the fast SPI signals. They are also a protection in case, the Chipset attempt to access the Serial Flash during the ISP update. The CS pull up resistor (R5) deselects the memory when chipset does not drive the bus and protect it from noise.

2.6. Conclusion

We recommend to the motherboard makers to select one of these updating methods in order to rationalize their solutions and offer to all the actors involved on the motherboard chain **ONE UNIQUE METHOD** to update all your motherboards portfolio and make it simple and convenient.

- **“Universal ON/OFF” method**
- **“Universal ON” method**
- **“Universal HZ” method**

If for cost reason, you decide to select different updating methods according to your chipset capabilities then we recommend you to provide a documentation to explain clearly the updating conditions according to the motherboard models.
DediProg team has generated a Generic Application Note to explain step by step the Bios updating process to the field operators. This Application note can be easily adapted to your solutions portfolio.
Please Download the **AN0106 "On-Board SPI programming with DediProg tools: End user version"** and contact us for your specific adaptation.

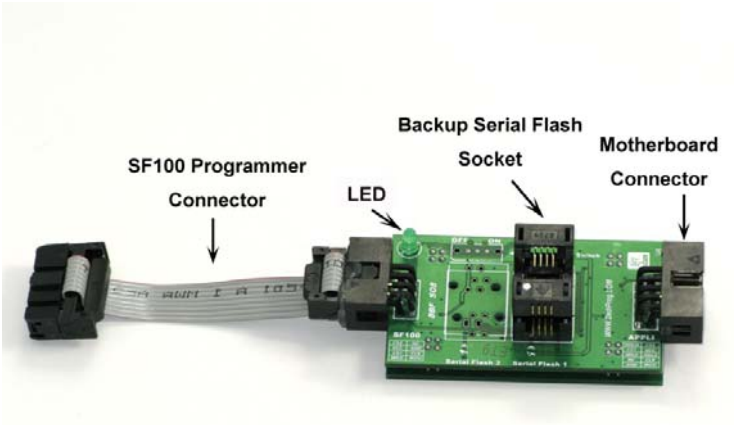
3. Backup Boot Flash design

3.1. What are the benefits for your company?

The DediProg Backup Boot Flash tool can be differentiate from the In System Programming method on the fact that it will not be used to update directly the Serial Flash on board but ingenuously oblige the computer to work on a backup memory.

This tool is very convenient for Bios development and for repairing in case, the ISP method cannot be supported by the motherboard.

Fig 12: Backup Boot Flash tool



Convenient for the Bios development:

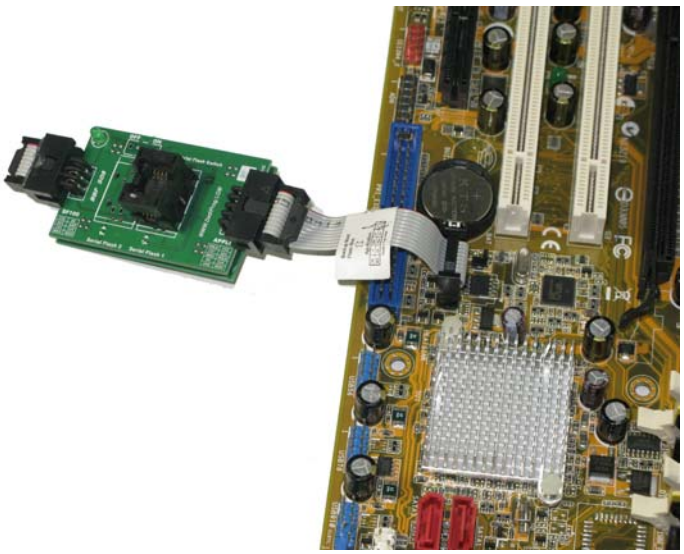
Bios team don't need to unsolder the part from the board to update the bios. With Our Backup Boot Flash tool, all the Bios trials and development could be done from the Serial Flash inserted in our tool socket. The Backup Serial Flash can been easily changed (different densities or suppliers) and even easily updated by connecting our SF100 programmer as below:

Fig 13: Backup Boot Flash tool connected on programmer

Useful for the Bios repairing if ISP method is not supported:
When the Bios of the motherboard is corrupted, the computer is not able to boot anymore. When connected on the motherboard, the DediProg Backup Boot tool will disable automatically the Main Serial Flash soldered on board so that chipset will boot on the backup Serial Flash inserted in the socket. After the motherboard boot completion, operator must disconnect the BBF tool from the motherboard to enable the main Serial Flash again and use the Flash Utility tool (usually provided by the Chipset maker, bios maker or motherboard maker) to update its Bios content and repair it.

The DediProg Backup Boot Flash tool can be connected on the motherboard pin header as below:

Fig 14: Backup Boot Flash connection on the motherboard connector



3.2. Design the motherboard for the BBF method

The Backup Boot Flash method does not require complex design:

- **Connect the Hold Pin to the Vcc through a Pull up resistor**
  Most of the market motherboards already fulfill this condition.

- **Select the method to connect the BBF tool on the motherboard**

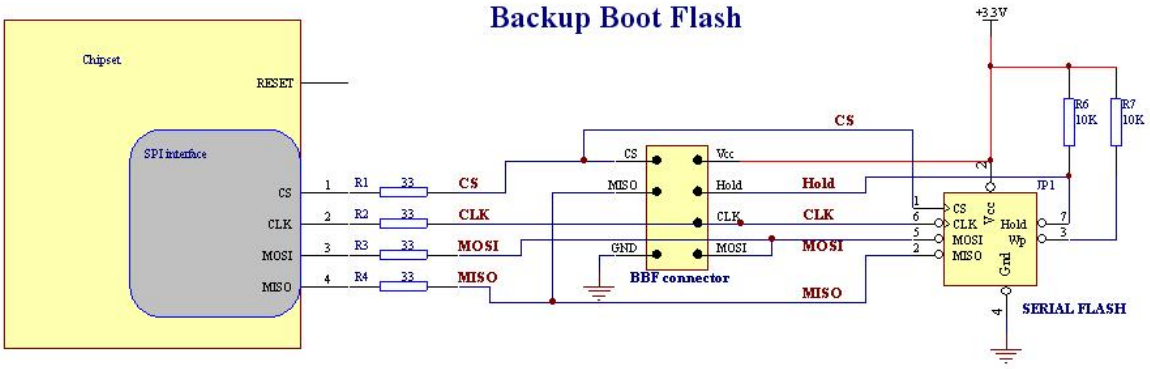
  1) A Standard BBF pin header on the motherboard
  The Standard BBF Pin header (8 pins 2.54mm pitch) has the equivalent pin-out than the SO8 Serial flash pin out.

Table 2:

1	CS	VCC	2
3	MISO	Hold	4
5	x	CLK	6
7	GND	MOSI	8

Remark: the pin 5 can be used as a mistake proof pin to prevent from wrong connection.

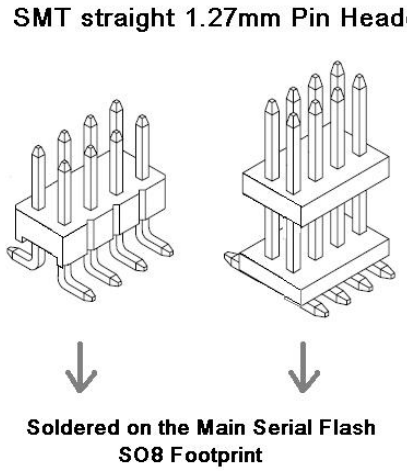
Figure 15: Standard connector for the BBF method



2) Replace the Main Serial Flash by a SMT Straight 1.27mm pin Header

This solution is very convenient for Bios development. The main Serial Flash is unsoldered from motherboard and our SMT Straight 1.27mm Pin header is soldered on its SO8 footprint. The Backup Boot Flash tool can then be connected for a perfect stability (no impact on the production version).

Figure 16: SMT Straight 1.27mm pin Header



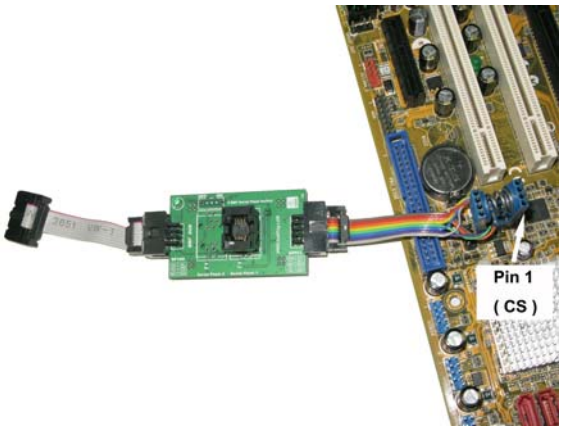
3) Connect the BBF on the Serial Flash Package:

Our SO Test Clip can be used to connect the Backup Boot Flash tool directly on the pins of the Serial Flash soldered on the board (SO8N, SO8W or SO16 packages only). This connection must be carefully handle to avoid wrong connection or unstable connection.

This solution can be used if the motherboard has not been designed to support the ISP or BBF connectors.

DediProg recommends to use soldered connector when ever is possible.

Fig 17: Backup Boot Flash connection on the Serial Flash package



3.3. Conclusion

The Backup Boot Flash method can be easily implemented without any complex hardware modification and even used in most of the motherboard without any modification but you have to remember that this method will not be the best choice for all the parties involved in the bios management (production, repairing) versus the ISP method.

Actually, The Backup Boot Flash method needs a Boot of the motherboard so:

- Requires the complete computer booting environment (ATX power, processor, monitor, Keyboard, mousse, RAM..).
- Requires the Flash utility tool (dos or window version)
- Requires more time to complete the Bios update versus the ISP method

DediProg Technology Co. Ltd.

4F., No.7, Ln. 143, Xinming Rd., Neihu Dist., Taipei City 114, Taiwan

Email for technical support: **support@dediprog.com**

Email for Sale information: **sales@dediprog.com**

www.DediProg.com