Freescale Semiconductor
Application Note

Document Number: AN3762
Rev. 0, 08/2008

HCS08 SG Family Background Debug Mode Entry

by: Carl Hu
Sr. Field Applications Engineer
Kokomo, IN, USA

Contents

1 Introduction

The HCS08 family of 8-bit microcontrollers uses background debug mode (BDM) as a means of allowing both development access and programming over a simple interface. BDM defines a simple 6 - pin dual row header in which three signals are critical for proper operation. This standardized method of connection is used in a variety of Freescale Semiconductor microcontrollers, most notably the 16-bit HC12, S12, and S12X families and now the 8-bit S08 family. BDM and its operation are well documented, but the focus of this note is on the S08 SG family in particular with an emphasis on the production line programming environment.

2 Background

There are two general methods of entering BDM mode:

- Placing the part into a background based on the state of the BKGD pin on the rising edge of

$\overline{\text{RESET}}$. This puts the part in an active BDM mode, where the part has stopped running the user code and is awaiting commands from the tool. This method requires an intrusive reset of the system.

- Communicating directly with the part using BDM commands over the BKGD pin without interrupting the executing user code. This is a true “background” mode where the user code is still running and the tool may read or write to memory locations without disrupting the running application. The tool may also place the part into an active BDM mode by writing to a command register.

In either case, the tool can measure the oscillator frequency of the microcontroller by sending a sync command to the BKGD pin. The device will then respond with a known waveform to which the tool can synchronize to establish a common frequency for communications.

NOTE

BDM mode can be either active or not. Active BDM (or active background) mode is intrusive, and it means that the part is held at attention waiting for commands (like go, reset, etc.) from the debug environment (PC). BDM mode (or background) is non-intrusive and allows only for certain functions to be executed while the part is still running the application code.

3 SG Family

There are two general types of S08 parts, those that have dedicated $\overline{\text{RESET}}$ pins, and those that don’t.

Traditionally, parts that have dedicated $\overline{\text{RESET}}$ pins use the rising edge of $\overline{\text{RESET}}$ to qualify the state of the BKGD pin and operating mode of the part (the part enters an active BDM mode if the BKGD is low).

However, recently the S08 devices have become available in 6- and 8-pin packages. In these devices, where there aren’t a lot of pins, a dedicated $\overline{\text{RESET}}$ pin is not present. In these S08 MCUs, the reset function is shared as a secondary function on one of the pins, and can be selected through software after the part exits reset. These parts use an internally generated signal to reset the MCU. This signal, known as power on reset or POR, is based on circuitry sensitive to the transition of power from near ground levels, to operating levels.

These MCUs enter the background mode by using the rising edge of the internal POR signal as the qualifier for the BKGD pin. Refer to the data sheet for specifications on the voltage and timing conditions of the POR signal.

NOTE

Even though the S08SG family parts have a dedicated $\overline{\text{RESET}}$ pin, they use the internal POR signal for qualifying the BKGD signal rather than the rising edge of a signal applied to the $\overline{\text{RESET}}$ pin.

This means that if the $\overline{\text{RESET}}$ pin on an S08SG microcontroller is toggled with the BKGD pin held low, it won’t enter active BDM mode. It will ignore the state of the BKGD pin instead, and enter user mode, where it will fetch the reset vector and execute at the target of the vector.

The consequence of this behavior has implications related to programming an S08SG device on a production line after it has been mounted to the circuit board. Any programming equipment or tool used

for this purpose must be able to control the MCU's power, $\overline{\text{RESET}}$, and BKGD pins. To put the device into active background mode, power must be applied to the MCU while the BKGD pin is held low. This will generate the internal POR signal that will allow the MCU to recognize the state of the BKGD pin.

After power has risen to specified operating levels, the BKGD pin can be used by the programming equipment to program the S08SG's flash memory. During the programming operation, the $\overline{\text{RESET}}$ pin must be controlled to ensure the MCU is not reset by an external device or electrical glitch. Allowing the $\overline{\text{RESET}}$ pin to be asserted after power has been applied will cause the S08SG to exit reset in user mode.

The signal sequence required to put the S08SG device into active background mode is shown in Figure 1. It also shows what happens if a glitch occurs on the $\overline{\text{RESET}}$ pin after power has been applied and the device has entered active background.

There are additional considerations when attempting to program brand new, blank parts that are described in the following sections.

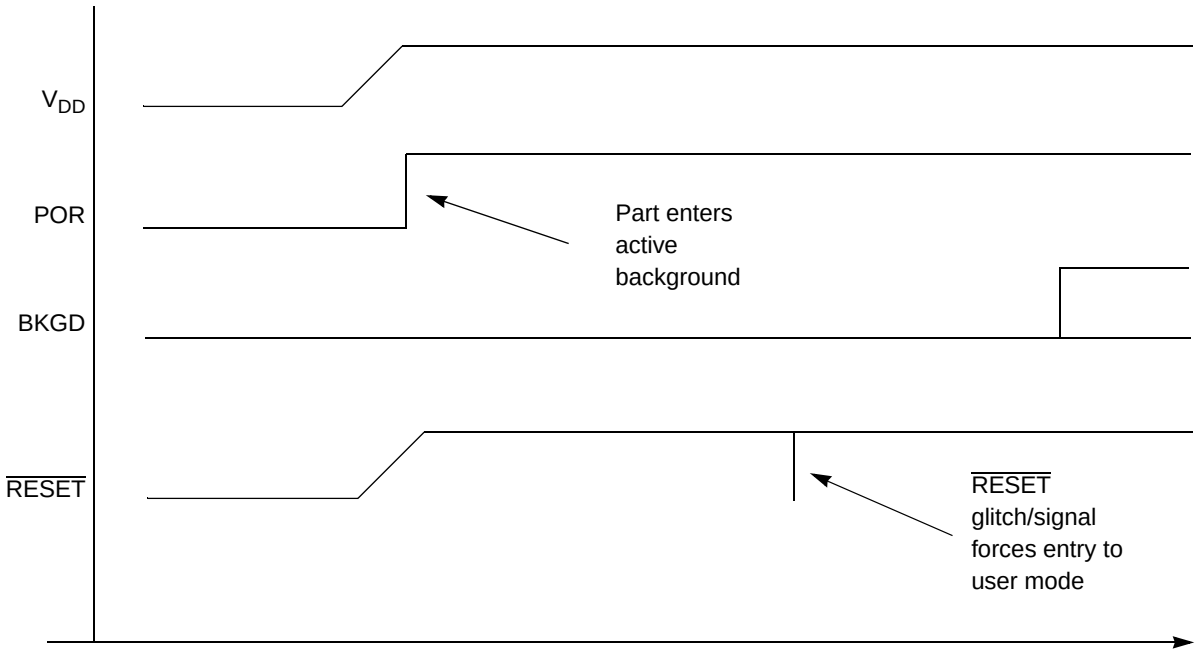

Figure 1. S08 SG Family BDM Entry Signals

3.1 Synchronization

Most BDM tools do not control the $\overline{\text{RESET}}$ and BKGD lines of a target device to enter the background mode, but communicate directly with the device in the background mode instead, using the sync command to establish a common data communication rate.

The sync command is a means of measuring the clock speed of a target device so that the tool may set its communication speed. The sync command begins with the tool driving the BKGD pin low for at least 128 target cycles at the lowest expected operating frequency of the BDM interface. The target device will then respond by waiting 16 BDM clock cycles to allow the host to stop driving the BKGD pin. The target then drives BKGD low for 128 cycles at the current BDM serial communication frequency.

The tool can measure this low going pulse from the target MCU and then begin to communicate with the target device using the non-intrusive background commands. Typically, the tool will send a command to place the target device in active background mode. This causes the device to stop executing application code and enter active background mode.

This method is used because it has the benefit of allowing a communication rate synchronization cycle to occur.

The sync command is designed to be unambiguous to the target MCU by asserting the BKGD pin for long enough that even at very low clock speeds, the MCU will recognize it as a sync command rather than normal BDM communications. If the sync signal is interrupted by a system reset (a reset that occurs during the recognition of a sync signal) the part will reset and won't generate the expected return signal.

If a tool that uses this method of entering active background mode is used for production line programming, the typical sequence will be:

1. Power up the part.
2. Programming tool executes the sync command.
3. If successful, programming tool gains control of the part and is able to program it.

3.1.1 Periodic Resets

There is a possibility that the sync command sequence that is being performed on a blank part will fail repeatedly due to a periodic MCU asserted reset that occurs before a sync sequence can be successfully completed.

When powered up, a blank S08 microcontroller will go into a self asserted reset loop of a period that is fairly consistent on a single part, but may vary across different parts.

What is happening is that the reset vector is blank (0xFFFF) and thus points to itself. This results in the execution of an STX 0,X instruction (opcode 0xFF), with H:X containing 0x00XX (which points to the first 256 bytes of memory – registers or RAM), followed by execution continuing at location 0x0000. This memory location is the port A data register area. Because all I/O ports are configured as inputs after reset, the port A data register contents can vary at power up based on input conditions at port A. If the contents of the port A data register contains a valid opcode, execution will continue until an illegal opcode or illegal address is encountered causing a self asserted reset. Note that the initial contents of RAM, while undefined, power up with remarkable consistency on a single part, but its contents can vary from part to part. For this reason, when blank parts are powered up, they will exhibit self-asserted reset loop times ranging from a few microseconds to a few milliseconds.

If a self asserted reset loop time is lesser than the time a sync command sequence needs to complete, the tool will not be able to synchronize with the part and establish background communications. Due to the variable period nature of the self asserted reset loop, it is difficult to guarantee that blank parts will have loop times long enough to allow a sync command sequence to complete.

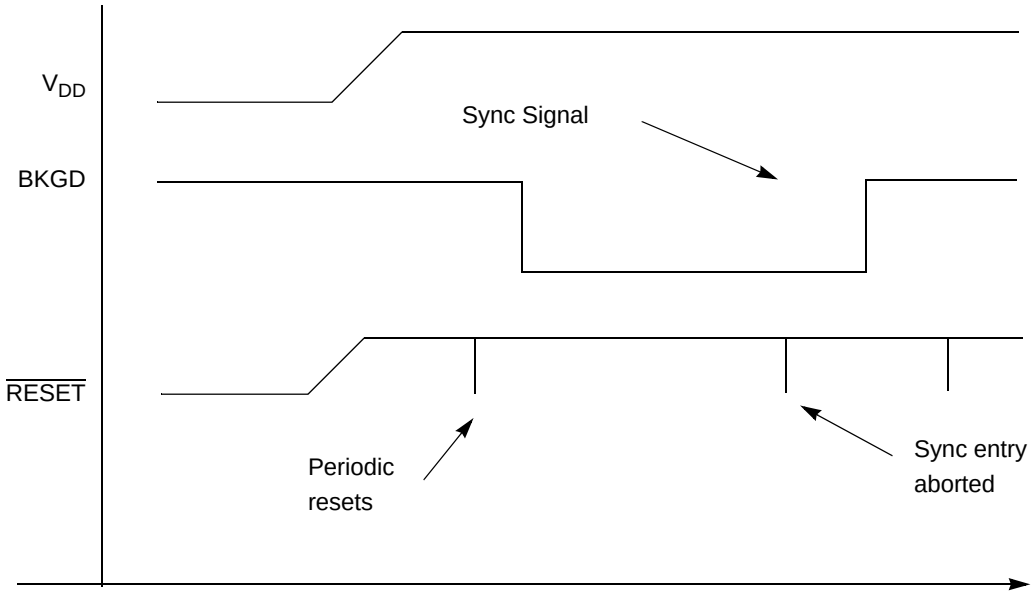

Figure 2. Sync Sequence Interrupted by Self-Asserted Resets

The above scenario limits the ability of tools using the sync command sequence to consistently establish a contact with blank parts that have been mounted to a PCB.

Due to this self asserted reset loop, programming of blank parts must be done by putting the part into active BDM mode (thus stopping user code from running) at power up and then allowing a tool to perform a sync sequence to establish communications with the part.

3.2 Reset Behavior

Understanding the reset behavior of the part is a key to designing a production line programming system. When the power is initially applied to an MCU, an internal mechanism sets a “POR” status bit and continues with the reset sequence. The $\overline{\text{RESET}}$ pin is then driven low by the MCU for a number of clock cycles (refer to particular family manuals for the number of cycles) and is then released. The $\overline{\text{RESET}}$ pin is then sampled to see if it is high — if so, then the reset is classified as a POR and if the BKGD pin is low, the device enters active BDM mode. If the $\overline{\text{RESET}}$ pin is low at this point, the reset is classified as a normal reset, and the reset vector is fetched. Note that even if an external pullup resistor is not connected to the reset pin, there’s an internal pullup resistor that will pull the reset pin high.

For a production line programming, the $\overline{\text{RESET}}$ pin must be passively pulled high and allowed to operate without external driving influences. The critical timing area is when the $\overline{\text{RESET}}$ pin is sampled after the internal circuitry releases the pin from being driven low. It’s best to let the passive circuitry pull this high; coordinating an active drive to correspond to this timing would be very difficult.

3.3 SG Family In-Line Programming Summary

To successfully program the SG family parts in a production line environment, all issues described above need to be addressed:

- The S08SG family uses an internally generated POR to qualify the state of the BKGD, not the rising edge of $\overline{\text{RESET}}$.
- Blank S08 parts will tend to exhibit self asserted reset loops of variable length on initial power up that may prevent a BDM tool from using the sync command sequence to gain control over the target MCU.
- Allow the $\overline{\text{RESET}}$ line to be passively pulled up and not actively driven by the programming tool.

How to do so can be summarized in the following steps:

1. Hold the BKGD pin low.
2. Apply power to V_{DD} while leaving the $\overline{\text{RESET}}$ to be passively pulled high allowing the part to enter active background mode. Do not allow any glitches or noise on the $\overline{\text{RESET}}$ line after this sequence.
3. Have the programming tool perform a sync command sequence to determine the target device's BDM operating frequency.
4. Program the part.

Programming tools generally do accommodate a sync sequence type of startup. They will typically hold the BKGD pin low, then ask the user to "cycle power to the target" and then go on to perform a sync command sequence and establish a contact with the part.

4 Recommended Production Line Programming Setup

A general production line programming setup consists of:

- A tester with a "bed of nails" probe or other means of connecting to the PCB
- An embedded BDM programming tool.
- A PC communicating with both a programming tool and a tester.

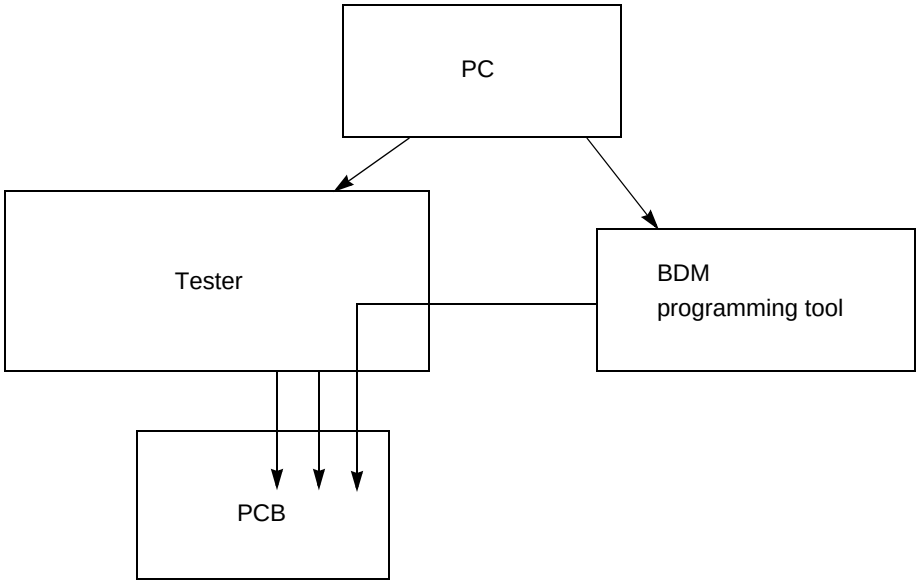

Figure 3. Recommended In-line Programming Circuit

Most applications using an S08 MCU tend to use one of two basic methods to control the reset pin:

- Systems where only passive components are on the $\overline{\text{RESET}}$ line. This is a traditional system with a pullup and a capacitor to ground off the $\overline{\text{RESET}}$ line. Because the S08SG has an internal pullup resistor on the reset pin, the external resistor can generally be eliminated.
- Systems where the $\overline{\text{RESET}}$ signal is driven by another logic, such as the GPIO of an another MCU.

In systems with a master MCU controlling the SG microcontroller's $\overline{\text{RESET}}$ pin with a GPIO pin, provision must be made when the system is designed to allow the SG microcontroller's $\overline{\text{RESET}}$ pin to be passively pulled high for production line programming. For example, when most of the MCUs are held in reset, all pins are configured as inputs, and thus present a high impedance or open circuit to the S08SG's $\overline{\text{RESET}}$ pin. Alternately, if the master MCU GPIO pin used to control the S08SG's $\overline{\text{RESET}}$ pin actively drives its pin during reset, a buffer that can be placed in a three-state or high-impedance mode may be inserted between the master MCU's GPIO pin and the S08SG's $\overline{\text{RESET}}$ pin.

After the S08SG is put into active background mode by holding the BKGD pin low while applying power, glitches or noise coupled onto the $\overline{\text{RESET}}$ pin can cause the S08SG to revert back to the user mode and prevent the programming tool from gaining access to the device. The time period during which this may occur is under the manufacturing station's control, and should be kept at an absolute minimum.

In addition, the programmer must be able to control the power of the S08SG device in conjunction with the assertion of the BKGD pin to force the part to enter active background mode.

The ability to control the power adjusted to the S08SG part in conjunction with its BKGD pin also needs to take into account the rest of the application's components. In many systems a common power supply is used for all logic devices. For example, if there are two microcontrollers in a system with one of the microcontrollers controlling the S08SG $\overline{\text{RESET}}$ line, a potential conflict could happen when attempting to apply power to the S08SG while holding its $\overline{\text{RESET}}$ line at an appropriate level. It may be helpful to design

a power supply isolation around the S08SG, so a tester can power the device without driving the remainder of the target board's power supply rail.

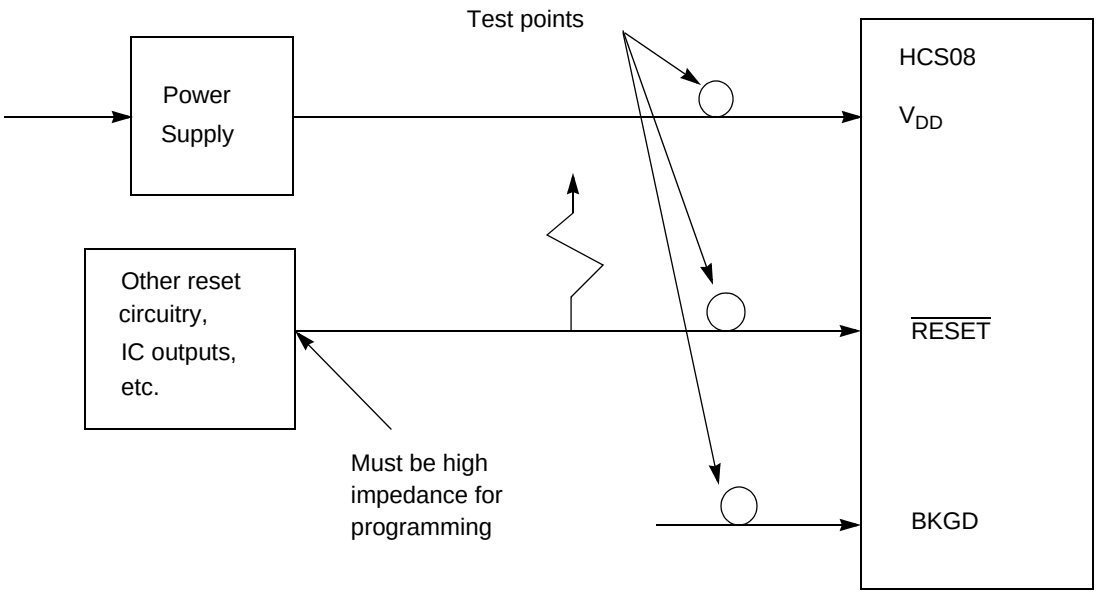

Figure 4. Recommended Application In-Line Programming Circuit

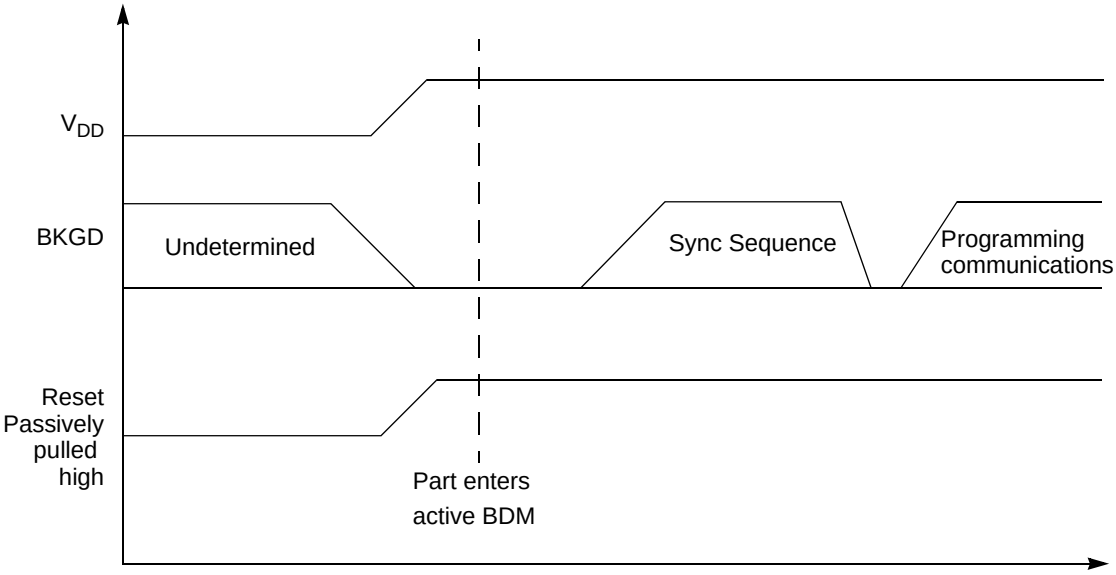

Figure 5. In-Line Programming Waveforms

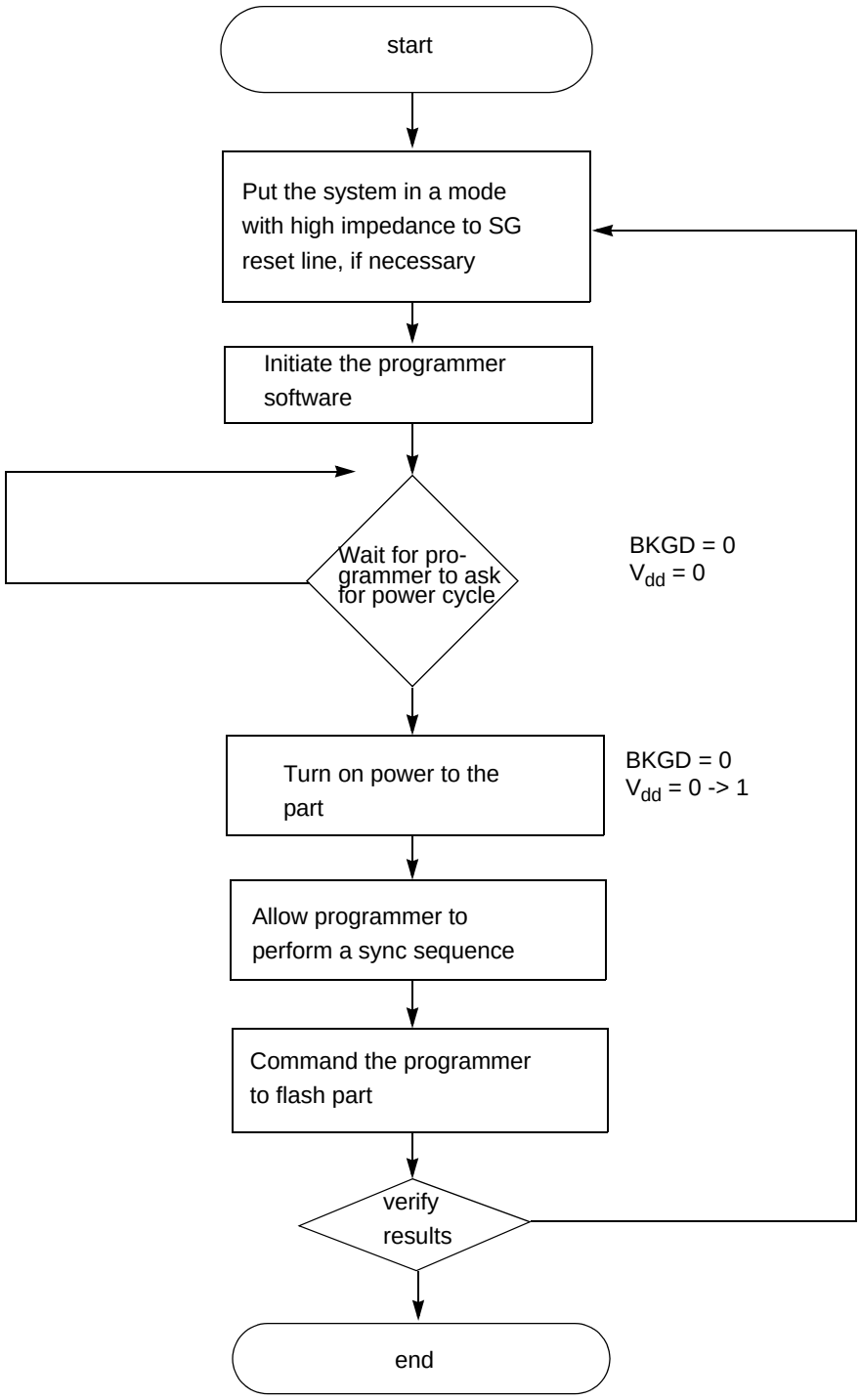

Figure 6. Flow Chart for Programming In Line with a BDM Programmer

5 Summary

A successful production line programming of blank S08 SG family devices relies on four points:

- Passive pullup to V_{DD} on $\overline{\text{RESET}}$ during programming sequences
- Power control of the S08SG in coordination with a programming tool
- Entry into active BDM mode followed by a sync command sequence to establish communications between the programming tool and the device
- Avoid glitches on the $\overline{\text{RESET}}$ line after the initial entry into an active BDM mode.

With a proper system design, these requirements can be met without any added cost to the application's PCB.

THIS PAGE IS INTENTIONALLY BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN3762
Rev. 0
08/2008

freescale™
semiconductor