AN4465
Application note

Serial Programming: BAM vs. BAF differences

Introduction

This document describes the differences between the Boot Assist Module (BAM) presents on SPC56 family and the Boot Assist Flash (BAF) presents on SPC57 family.

The purpose of this document is to highlight the differences between the two modules (BAM and BAF).

Contents

1 Boot Assist Module (BAM)

1.1 Boot modes

The SPC56 family supports the following boot modes:

- Boot from internal flash. The device boots from the first bootable section of the Flash main array.
- Serial Boot. The device downloads boot code from either SCI or CAN interface and then execute it.

If booting is not possible with the selected configuration (e.g. if no valid Boot ID is found in any of the possible boot locations) then the device enters the static mode. The device enters power safe mode and waits for an external reset.

1.2 BAM overview

The Boot Assist Module is a block of read-only memory containing VLE code which is executed according to the boot mode of the device. The BAM downloads code into internal SRAM through the following serial protocols and executes it afterwards:

- CAN (with and without auto baud depending the target)
- UART (with and without auto baud depending the target)

The BAM provides the following features:

- Programmable 64-bit password protection for serial boot mode
- Serial boot loads the application boot code from a CAN or UART bus into internal SRAM
- Censorship protection for internal flash module

The BAM code resides in 8 kilobytes of ROM mapped from address 0xFFFF_C000.

The RAM location where to download the code can be any 4 byte-aligned location in the SRAM starting from the address 0x4000_0100.

The SPC56 detects the boot mode based on external pins and device status. The following sequence applies:

1. to boot either from CAN or UART, the device must be forced into an Alternate Boot Loader Mode via the FAB (Force Alternate Boot Mode) pin which must be asserted before initiating the reset sequence. The type of alternate boot mode is selected according to the ABS (Alternate Boot Selector) pins.
2. if FAB is not asserted, the device boots from the first flash-memory sector which contains a valid boot signature.
3. if no flash memory sector contains a valid boot signature, the device goes into static mode

1.3 Censorship

The internal Flash memory can be enabled or disabled it depends on the values stored in the censorship word and serial boot control word in the shadow row of the internal Flash memory, the Nexus port can be enabled or disabled, the password received in the serial

boot mode is compared with the fixed public password or compared to a user programmable password in the internal Flash memory.

The censorship word is a 32-bit word of data stored in the shadow row of internal Flash memory. This memory location is read and interpreted by hardware as part of the boot process and is used in conjunction with the BAM configuration pin to enable/disable the internal Flash memory and the Nexus interface. The address of the Censorship word is 0x00FF_FDE0. The censorship word consists of two fields: censorship control and serial boot control. The censorship word is programmed during manufacturing to be 0x55AA_55AA. This results in a device that is not censored and uses a Flash-based password for serial boot mode.

The BAM program uses the state of the SIU_CCR DISNEX bit to determine whether the serial password received in serial boot mode, should be compared to a public password (fixed value of the 0xFEED_FACE_CAFE_BEEF) or needs to be compared to a Flash password - 64-bit data, stored in the shadow row of internal Flash at address 0x00FF_FDD8. If the bit is set, the BAM uses the Flash serial password, if the bit is cleared, it uses the public password.

- A valid serial password must be always programmed, regardless the boot mode used. This provides capability to “rescue” the part using the serial boot mode, if the Flash content becomes corrupted for whatever reason.

1.4 BAM Download protocol

From high level perspective, the download protocol follows steps:

1. Send message and receive acknowledge message for auto baud or auto bit rate selection (optional step depending the target and the ABS pins configuration).
2. Send 64 bits password.
3. Send start address, size of downloaded code in bytes and VLE bit.
4. Download data.
5. Execute code from start address.

Each step must be complete before the next step starts. These steps are correct if auto baud is disabled. Otherwise, to measure the baud rate some data from host to MCU are sent before the step 1.

The communication is done in half duplex manner and any transmission from host is followed by the MCU transmission:

- Host sends data to MCU and starts waiting
- MCU echoes to host the data received
- Host verifies if echoes are correct:
  - If data is correct, the host can continue to send data;
  - If data is not correct, the host stops to transmit and MCU need to be reset.

All multi-byte data structures are sent with MSB first.

2 Boot Assist Flash (BAF)

2.1 Boot modes

Depending on the state and the content of its flash memory the device can enter one of three different boot modes:

- Boot from internal flash. If the internal flash contains application code with a valid boot header, the application is booted. The boot header contains flags that enable/disable individual cores and their reset vectors
- Serial boot. If no valid boot header is found and the current lifecycle phase of the device is less than “In Field”, an attempt is made to download the application by a serial protocol using the UART. The downloaded code is then executed by the initial boot core.
- Static mode. The device enters power safe mode and waits for an external reset.

2.2 BAF Overview

The MCU is booted through a collaboration of several blocks, hardware, and firmware. The first boot phases are performed by a state machine inside the System Status and Configuration Module (SSCM).

Finally, the SSCM sends a reset vector to the boot core (the I/O Processor (IOP) on this device) pointing into the Boot Assist Flash (BAF).

The BAF code then checks the life cycle of the device. If it is “Failure Analysis”, BAF enters a loop in which it will service the watch dog. BAF will not execute further since the read access to flash boot sectors for cores may be disabled if device life cycle is “Failure Analysis”. Otherwise it searches for a boot header and boots the application code in internal flash.

If no boot header is found in internal flash, it downloads application code serially using the UART module. The package pins used by UART are the same pins that can be used by CAN module. So the external PHY can be either UART or CAN.

The BAF is located in a 16 KB block of flash that is mapped adjacent to the UTEST flash memory block.

The BAF block base address is 0040_4000h. It is one time programmable (OTP) and is programmed during factory test.

2.3 BAF Download Protocol

From high level perspective, the download protocol follows steps:

1. Host transmits 64 bit password to MCU.
2. Host transmits start address, size of downloaded code in bytes and VLE bit13 to MCU.
3. Host transmits a sequence of data bytes that are to be executed.
4. CPU2 executes code from start address provided in step 2.

Each step must be complete before the next step starts.

The communication is done in half duplex manner and any transmission from host is followed by the MCU transmission:

1. Host sends data to MCU and start waiting.
2. The MCU echoes to host the data received.
3. The host verifies if the echo is correct:
   – If data is correct, the host can send the next byte of data
   – If data is not correct, it is assumed the MCU has not correctly received the data and the only practical option is for the Host to reset the MCU and start again.

All multi-byte data structures are sent with MSB first.

3 Differences between BAM and BAF

3.1 Activation mode

The sequences to start the serial boot are different: while the BAM mode is enabled by hardware configuration using FAB pin and ABS pins, the BAF mode is automatically enabled only if no valid boot header is found and the lifecycle phase of the device is less than “In Field”. This means that, to use BAM, the user needs to have the configuration pins accessible. In this case the flash can be managed without limitation. Instead, the BAF module allows managing the flash only in specific condition but without external hardware configuration.

3.2 Serial boot protocol

From high level perspective, the serial boot protocols are similar except the behavior of the device when the user sends a wrong password or not sends the password:

- BAM: If the password check fails, the device stops responding. To get the device out of that state, the RESET signal must be asserted.
- BAF: If an incorrect password is supplied, BAF puts the device into static mode (low power safe mode). If 64 bits of password data is not received after approximately 30s, BAF causes a destructive reset of the MCU.

3.3 Code mode

Another difference between BAM and BAF regarding the code mode available: SPC57 family executes only VLE code. So, the bit present into the BAF protocol is used only for a backward compatibility.

3.4 Password managing

BAF has a public fixed password (fixed value: 0xFEEDFACECAFEBEEF) not modifiable while BAM can use a public password (fixed value: 0xFEEDFACECAFEBEEF) or can use a 64 bit Flash password, stored in the shadow row of internal Flash.

3.5 Baud rate detection

SPC57 family does not support baud rate detection option, instead the BAM has the option to use the baud rate detection.

3.6 RAM reservation

SPC57 family does not need RAM reservation for executing BAF. The BAM protocol, on the contrary, requires the reservation of a part of RAM (from 0x4000_0000 to 0x4000_0100). This RAM cannot be overwritten during the download phase.

3.7 Serial interface baud rate calculation

UART and CAN baud rate depending by external clock but with different formulas.

- BAM module formulas:
  - UART baud rate = f_{sys} / 833.33
  - CAN baud rate = f_{sys} / 40
- BAF module formulas:
  - UART baud rate = f_{sys} / 80
  - CAN baud rate = f_{sys} / 40

4 Revision history

Table 1. Document revision history

Date	Revision	Changes
09-Apr-2014	1	Initial release.