PM0020
Programming manual

STR91xFA Flash programming manual

Introduction

This manual describes how to configure, program, erase and protect access to the Flash memory of the STR91xFA microcontroller family.

The STR91xFA embedded Flash memory can be programmed using In-Circuit Programming or In-Application programming.

The In-Circuit programming (ICP) method is used to update the entire contents of the Flash memory, using the JTAG protocol to load the user application into the microcontroller. ICP offers quick and efficient design iterations and eliminates unnecessary package handling or socketing of devices.

In contrast to the ICP method, In-Application Programming (IAP) can use any communication interface supported by the microcontroller (I/Os, USB, CAN, UART...) to download the data to be programmed in memory. IAP allows you to re-program the Flash memory while the application is executing. Nevertheless, part of the application has to have been previously programmed in one of the Flash banks using ICP.

Glossary

This section gives a brief definition of acronyms and abbreviations used in this document:

CUI (Command User Interface): is the software interface of the FPEC.

FPEC (Flash Program/Erase controller): The write operations to the 2 banks are managed by an embedded FPEC.

IAP (In-Application Programming): IAP is the ability to program a bank of Flash memory from the user application residing in another bank.

ICP (In-Circuit Programming): ICP is the ability to program the Flash memory of a microcontroller using JTAG protocol.

JTAG (Joint Test Action Group): The debug interface of the ARM966E-S core is based on the Joint Test Action Group (JTAG) protocol.

Word address: Address that aligns to a word boundary (address lines A[1:0]=00).

Contents

List of tables

List of figures

1 STR91xFA Flash memory

1.1 Features

- Two Flash memory banks with a total capacity up to 2048+128 Kbytes
- 32-bit burst read access, 16-bit write access
- Sequential Burst read up to 96 MHz
- I-TCM (Instruction Tightly Coupled Memory) interface to ARM966-ES core
- Branch Cache (BC) and Instruction prefetch queue (PFQ)
- AHB Interface to FMI registers
- Erasing on a sector or bank basis, and programming on a 16-bit halfword basis
- Each bank can be programmed and erased over 100,000 cycles
- 20-year data retention
- Each sector can be separately protected and unprotected against program and erase
- Electronic Signature memory
- 256 bits of OTP data memory
- Read-While-Write (RWW) Dual Bank operations
- Security protection bit to lock JTAG access or readout

1.2 Memory organization

The Flash program memory is organized in 32-bit wide memory cells which can be used for storing both code and data constants.

You can Program Bank 0 and Bank 1 independently, i.e. you can read from one bank while writing to the other.

The on-chip Flash is divided in 2 banks that can mapped independently in the 64 MByte address space 0x0000-0000 - 0x03FF.FFF by programming the FMI registers.

The STR91xFAxx2 embedded Flash Module is organized as shown in *Table 1*.

Table 1. STR91xFAxx2 Flash module organization

Bank	Sector	Address Offset	Size (bytes)
Bank 0 256 Kbytes	Bank 0 Sector 0	0x0000.0000 - 0x0000.FFFF	64K
	Bank 0 Sector 1	0x0001.0000 - 0x0001.FFFF	64K
	Bank 0 Sector 2	0x0002.0000 - 0x0002.FFFF	64K
	Bank 0 Sector 3	0x0003.0000 - 0x0003.FFFF	64K
Bank 1 32 Kbytes	Bank 1 Sector 0	0x0000.0000 - 0x0000.1FFF	8K
	Bank 1 Sector 1	0x0000.2000 - 0x0000.3FFF	8K
	Bank 1 Sector 2	0x0000.4000 - 0x0000.5FFF	8K
	Bank 1 Sector 3	0x0000.6000 - 0x0000.7FFF	8K
Bank 1	User Configuration Sector (OTP and Electronic Signature, Configuration and Protection Registers)	Access via CUI or JTAG	32

The STR91xFAxx4 embedded Flash Module is organized as shown in *Table 2*.

Table 2. STR91xFAxx4 Flash module organization

Bank	Sector	Address Offset	Size (bytes)
Bank 0 512 Kbytes	Bank 0 Sector 0	0x0000.0000 - 0x0000.FFFF	64K
	Bank 0 Sector 1	0x0001.0000 - 0x0001.FFFF	64K
	Bank 0 Sector 2	0x0002.0000 - 0x0002.FFFF	64K
	Bank 0 Sector 3	0x0003.0000 - 0x0003.FFFF	64K
	Bank 0 Sector 4	0x0004.0000 - 0x0004.FFFF	64K
	Bank 0 Sector 5	0x0005.0000 - 0x0005.FFFF	64K
	Bank 0 Sector 6	0x0006.0000 - 0x0006.FFFF	64K
	Bank 0 Sector 7	0x0007.0000 - 0x0007.FFFF	64K

Table 2. STR91xFAxx4 Flash module organization

Bank	Sector	Address Offset	Size (bytes)
Bank 1 32 Kbytes	Bank 1 Sector 0	0x0000.0000 - 0x0000.1FFF	8K
	Bank 1 Sector 1	0x0000.2000 - 0x0000.3FFF	8K
	Bank 1 Sector 2	0x0000.4000 - 0x0000.5FFF	8K
	Bank 1 Sector 3	0x0000.6000 - 0x0000.7FFF	8K
Bank 1	User Configuration Sector (OTP and Electronic Signature, Configuration and Protection Registers)	Access via CUI or JTAG	32

The STR91xFAxx6 embedded Flash Module is organized as shown in *Table 3*

Table 3. STR91xFAxx6 Flash module organization

Bank	Sector	Address Offset	Size (bytes)
Bank 0 1024Kbytes	Bank 0 Sector 0	0x0000.0000 - 0x0000.FFFF	64K
	Bank 0 Sector 1	0x0001.0000 - 0x0001.FFFF	64K
	Bank 0 Sector 2	0x0002.0000 - 0x0002.FFFF	64K
	Bank 0 Sector 3	0x0003.0000 - 0x0003.FFFF	64K
	Bank 0 Sector 4	0x0004.0000 - 0x0004.FFFF	64K
	Bank 0 Sector 5	0x0005.0000 - 0x0005.FFFF	64K
	Bank 0 Sector 6	0x0006.0000 - 0x0006.FFFF	64K
	Bank 0 Sector 7	0x0007.0000 - 0x0007.FFFF	64K
	Bank 0 Sector 8	0x0008.0000 - 0x0008.FFFF	64K
	Bank 0 Sector 9	0x0009.0000 - 0x0009.FFFF	64K
	Bank 0 Sector 10	0x000A.0000 - 0x000A.FFFF	64K
	Bank 0 Sector 11	0x000B.0000 - 0x000B.FFFF	64K
	Bank 0 Sector 12	0x000C.0000 - 0x000C.FFFF	64K
	Bank 0 Sector 13	0x000D.0000 - 0x000D.FFFF	64K
	Bank 0 Sector 14	0x000E.0000 - 0x000E.FFFF	64K
	Bank 0 Sector 15	0x000F.0000 - 0x000F.FFFF	64K

Table 3. STR91xFAxx6 Flash module organization (continued)

Bank	Sector	Address Offset	Size (bytes)
Bank 1 128Kbytes	Bank 1 Sector 0	0x0000.0000 - 0x0000.3FFF	16K
	Bank 1 Sector 1	0x0000.4000 - 0x0000.7FFF	16K
	Bank 1 Sector 2	0x0000.8000 - 0x0000.BFFF	16K
	Bank 1 Sector 3	0x0000.C000 - 0x0000.FFFF	16K
	Bank 1 Sector 4	0x0001.0000 - 0x0001.3FFF	16K
	Bank 1 Sector 5	0x0001.4000 - 0x0001.7FFF	16K
	Bank 1 Sector 6	0x0001.8000 - 0x0001.BFFF	16K
	Bank 1 Sector 7	0x0001.C000 - 0x0001.FFFF	16K
Bank 1	User Configuration Sector (OTP and Electronic Signature, Configuration and Protection Registers)	Access via CUI or JTAG	32

The STR91xFAxx7 embedded Flash Module is organized as shown in *Table 4*.

Table 4. STR91xFAxx7 Flash module organization

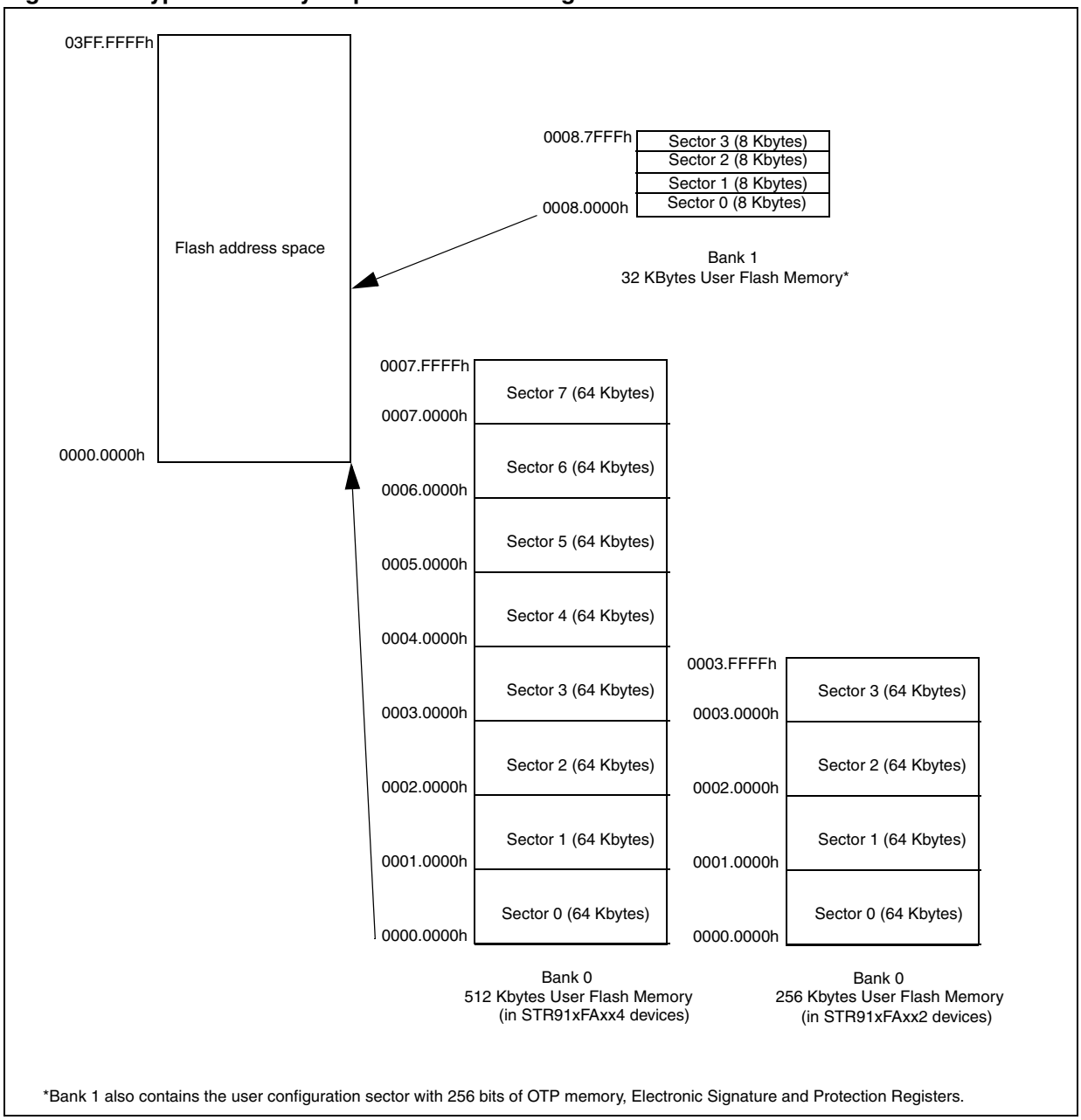
Bank	Sector	Address Offset	Size (bytes)
Bank 0 2048Kbytes	Bank 0 Sector 0	0x0000.0000 - 0x0000.FFFF	64K
	Bank 0 Sector 1	0x0001.0000 - 0x0001.FFFF	64K
	Bank 0 Sector 2	0x0002.0000 - 0x0002.FFFF	64K
	Bank 0 Sector 3	0x0003.0000 - 0x0003.FFFF	64K
	Bank 0 Sector 4	0x0004.0000 - 0x0004.FFFF	64K
	Bank 0 Sector 5	0x0005.0000 - 0x0005.FFFF	64K
	Bank 0 Sector 6	0x0006.0000 - 0x0006.FFFF	64K
	Bank 0 Sector 7	0x0007.0000 - 0x0007.FFFF	64K
	Bank 0 Sector 8	0x0008.0000 - 0x0008.FFFF	64K
	Bank 0 Sector 9	0x0009.0000 - 0x0009.FFFF	64K
	Bank 0 Sector 10	0x000A.0000 - 0x000A.FFFF	64K
	Bank 0 Sector 11	0x000B.0000 - 0x000B.FFFF	64K
	Bank 0 Sector 12	0x000C.0000 - 0x000C.FFFF	64K
	Bank 0 Sector 13	0x000D.0000 - 0x000D.FFFF	64K
	Bank 0 Sector 14	0x000E.0000 - 0x000E.FFFF	64K
	Bank 0 Sector 15	0x000F.0000 - 0x000F.FFFF	64K
	Bank 0 Sector 16	0x0010.0000 - 0x0010.FFFF	64K
	Bank 0 Sector 17	0x0011.0000 - 0x0011.FFFF	64K
	Bank 0 Sector 18	0x0012.0000 - 0x0012.FFFF	64K
	Bank 0 Sector 19	0x0013.0000 - 0x0013.FFFF	64K
	Bank 0 Sector 20	0x0014.0000 - 0x0014.FFFF	64K
	Bank 0 Sector 21	0x0015.0000 - 0x0015.FFFF	64K
	Bank 0 Sector 22	0x0016.0000 - 0x0016.FFFF	64K
	Bank 0 Sector 23	0x0017.0000 - 0x0017.FFFF	64K
	Bank 0 Sector 24	0x0018.0000 - 0x0018.FFFF	64K
	Bank 0 Sector 25	0x0019.0000 - 0x0019.FFFF	64K
	Bank 0 Sector 26	0x001A.0000 - 0x001A.FFFF	64K
	Bank 0 Sector 27	0x001B.0000 - 0x001B.FFFF	64K
	Bank 0 Sector 28	0x001C.0000 - 0x001C.FFFF	64K
	Bank 0 Sector 29	0x001D.0000 - 0x001D.FFFF	64K
	Bank 0 Sector 30	0x001E.0000 - 0x001E.FFFF	64K
	Bank 0 Sector 31	0x001F.0000 - 0x001F.FFFF	64K

Table 4. STR91xFAxx7 Flash module organization (continued)

Bank	Sector	Address Offset	Size (bytes)
Bank 1 128Kbytes	Bank 1 Sector 0	0x0000.0000 - 0x0000.3FFF	16K
	Bank 1 Sector 1	0x0000.4000 - 0x0000.7FFF	16K
	Bank 1 Sector 2	0x0000.8000 - 0x0000.BFFF	16K
	Bank 1 Sector 3	0x0000.C000 - 0x0000.FFFF	16K
	Bank 1 Sector 4	0x0001.0000 - 0x0001.3FFF	16K
	Bank 1 Sector 5	0x0001.4000 - 0x0001.7FFF	16K
	Bank 1 Sector 6	0x0001.8000 - 0x0001.BFFF	16K
	Bank 1 Sector 7	0x0001.C000 - 0x0001.FFFF	16K
Bank 1	User Configuration Sector (OTP and Electronic Signature, Configuration and Protection Registers)	Access via CUI or JTAG	32

The write operations of the two banks are managed by an embedded Flash Program/Erase Controller (FPEC). The high voltage needed for Program/Erase operations is internally generated.

Figure 1. Typical memory map with device configured to boot from Bank 0



1.3 Initialization

After reset, to define the mapping of the Flash memory banks, the user firmware has to write the start address and memory size of Bank 0 and Bank 1 in the FMI registers (see *Section 1.15*).

You must write the start address and the memory size of the bank configured as boot memory first and then the start address and the memory size of the other (non-boot) bank.

1.4 Boot configuration

In the default configuration, after reset the first sector of Bank 0 is enabled and resides at 0x0000.0000h so that the device boots from Bank 0, and Bank 1 is disabled.

Using the JTAG interface, you can configure the device to boot from Bank 1. The selection of which Flash memory is at the boot location is programmed in a non-volatile Flash-based configuration bit. The firmware cannot change this configuration bit, only the JTAG interface has access.

1.5 Electronic signature

The electronic signature can be read via JTAG or from the application using the RSIG command (*Section 1.11.2*).

The ISC IDCODE assigned to the Flash memory die is 04570041h.

Note: The electronic signature read by the RSIG command contains the JTAG ID code of the Flash memory die. To get the STR91xFA product ID and revision level, read the last two bytes of the OTP sector.

1.6 OTP sector

The device provides 240 bits of user OTP memory. The last 16 bits in the 256-bit sector are reserved for the STR91xFA product ID and revision level and programmed by ST. You can read all these bits, and program any unprogrammed bits via JTAG or from the application using the ROTP and WOTP commands (see *Section 1.11.3* and *Section 1.12.8* . The OTP sector can be locked via JTAG or CUI command to prevent any modification. The OTP lock status bit is stored in the Protection Level 2 register which can be read by the application. Refer to *Section 1.13.2*).

1.7 Power down mode

In STR91xFA low power modes, the Flash automatically reduces its power consumption and can be read immediately after wake-up.

When the STR91xFA is in low power mode, you can put the Flash in Power Down mode as well for lower power consumption. You do this by programming the PWD bit in the Flash Configuration register (see *Section 1.13.6*). The consumption is drastically reduced, but after wake-up from low power, a delay is inserted automatically to ensure the Flash is operational before the CPU starts execution.

1.8 JTAG interface

The device can be erased and programmed by sending commands to the command interface described in *Table* .

The device can also be programmed through the 4 pin JTAG port which is 1532 standard compliant and supports standard ISC (In System Configuration) programming instructions.

1.8.1 JTAG access protection

You can protect the entire STR91xFA device from read or debug access through the JTAG port by setting the SECURITY bit in the Protection Level 2 register (see *Section 1.13.2*.). Once SECURITY is set, the Flash content cannot be read by JTAG. Only a full chip erase via JTAG can erase the SECURITY bit.

A secured device will not respond to JTAG program or erase commands other than Full Chip Erase, Read User Code and Read ID Code commands. The SECURITY bit has no impact on CUI commands.

1.9 Sector write protection

Sector write protection is managed on 2 levels:

- Protection Level 1
  - Level 1 protection is controlled by the volatile Protection Level 1 register.
  - Each sector has a level 1 write protection bit (see *Section 1.13.4*). After a device reset, all sectors are protected.
  - You are free to modify the protection status of any sectors that are not protected by the Protection Level 2. This can be done from the application using a CUI Program Protection Level 1 command, see *Section 1.12.5*.
- Protection Level 2
  - Level 2 protection is controlled by the non-volatile Protection Level 2 register
  - Each sector has a Level 2 write protection bit (see *Section 1.13.4*). When the bit is set through the JTAG, the sector is always write protected. It can be unprotected only by erasing the bit via JTAG.

1.10 Command user interface (CUI)

Program and Erase commands are written to the Command User Interface. An on-chip Program/Erase Controller simplifies the process of programming or erasing the memory by taking care of all of the special operations that are required to update the memory contents. The end of a Program or Erase operation can be detected and any error conditions identified in the Bank Status Registers. The command set required to control the memory is consistent with JEDEC standards.

Erase can be suspended in order to perform either read or program in another sector and then resumed. Program can be suspended to read data in any other sector and then resumed. The two banks are protected during power-up.

The device features 2 different levels of protection to avoid unwanted program/erase operations. The sectors can be protected at Level 1 by the Sector Protect command or by the Protection Level 2 Register programmed via JTAG. All Program or Erase operations are aborted when the device is reset or powered off.

When the application accesses an address in Flash memory, the Command User Interface (CUI) decodes it as an instruction to the Program/Erase Controller (FPEC). The Bank Status Registers indicate the status resulting from the command. They can be read at any time during programming or erase to monitor the progress of the operation and to verify its successful completion.

The reset state of the CUI is Read Array. The command sequence must be followed exactly. Any invalid combination of commands will reset the device to Read Array mode.

Table 5. CUI commands

Hex Code	Command
00h	Invalid Reset
01h	Protect Level 1 Confirm
03h	Write Flash Configuration Register Confirm
10h	Alternative Program Set-up
20h	Sector Erase Set-up
40h	Program Set-up
50h	Clear Status Register
60h	Protect Level 1 Set-up and Write Flash Configuration Register Set-up
70h	Read Status Register
80h	Bank Erase Set-up
90h	Read Electronic Signature
98h	Read OTP sector
B0h	Program/Erase Suspend
C0h	Write (program) OTP or program OTP Lock bit
D0h	Program/Erase Resume, Erase Confirm or Unprotect Level 1 Confirm
FFh	Read Array

1.10.1 Memory bank concurrency

While one bank is in read mode, the other bank can be programmed or erased. The two Flash banks do not support programming or erasing in parallel.

A bank can always be read while erasure is in progress in the other bank.

Flash sector erasure may be suspended while data is read from other sectors of the same Flash memory bank and then resumed after reading.

Table 6 lists the allowed dual operations in both banks and on the same bank.

Table 6. Rules for dual bank operations

Bank Status	Commands allowed in other bank				
	Read Array/ Read Status Reg./ Read Electronic Signature	Program	Sector Erase	Program/Erase Suspend	Program/Erase Resume
Idle	Yes	Yes	Yes	Yes	Yes
Programming	Yes	-	-	Yes	-
Erasing	Yes	-	-	Yes	-
Programming Suspended	Yes	-	-	-	Yes
Erase Suspended	Yes	Yes	-	-	Yes

1.11 Read operations

1.11.1 RSR read bank status register (70h)

Each bank has a Bank Status Register which indicates when a program or erase operation is completed and the success or error status. Refer to *Section 1.13.1* for the register description.

To read the Bank Status Register:

1. Write a Read Status Register command (70h) to any word address in the bank. The status of the other bank is not affected by the command. You can write a Read Status Register command at any time, even while a Program/Erase operation is ongoing.
2. Read any address in the bank to obtain the content of the Status Register.

See also *Section 1.12.3: CLSR clear status register (50h)*.

1.11.2 RSIG read electronic signature (90h)

The Electronic Signature is stored in the user configuration sector of Bank 1 which you can read using the RSIG command. It contains factory-programmed identification data that is useful to allow other devices to automatically match their interface to the characteristics of the microcontroller. The RSIG sector also contains Sector protection and Flash Configuration registers.

To read the Electronic Signature content:

1. Write a Read Electronic Signature command (90h) to any word address in Bank 1.
2. Read any of the RSIG registers shown in *Table 7* Reading the RSIG locations can be non-sequential. To access the RSIG registers, read from any address in Bank 1, using the low byte of the address to select the register.
3. To exit RSIG mode, write a Read Array command (FFh) to any word address in Bank 1 to return it to Read Array mode.

Table 7. RSIG electronic signature data for STR91xFAxx2 and STR91xFAxx4

Code or register	Bank 1 word address LSB	Data D[31:0]
Manufacturer Code	00h	20h
Device Code	01h	04570041h
Die Revision Code	02h	xxxx0000h
Protection Level 2 Register	03h	xxxxxxxxh
Protection Level 1 Register	04h	xxxxxxxxh
Flash Configuration Register	05h	xxxxxxxxh

Table 8. RSIG electronic signature data for STR91xFAxx6 and STR91xFAxx7

Code or register	Bank 1 word address LSB	Data D[31:0]
Manufacturer Code	00h	20h
Device Code	01h	04570041h
Die Revision Code	02h	xxxx0000h
Protection Level 2 Register Bank 0	03h	xxxxxxxxh

Table 8. RSIG electronic signature data for STR91xFAxx6 and STR91xFAxx7

Code or register	Bank 1 word address LSB	Data D[31:0]
Protection Level 2 Register Bank 1	04h	xxxxxxxxh
Protection Level 1 Register Bank 0	05h	xxxxxxxxh
Protection Level 1 Register Bank 1	06h	xxxxxxxxh
Flash Configuration Register	07h	xxxxxxxxh

1.11.3 ROTP read OTP sector (98h)

The OTP (one-time programmable) sector is the user configuration sector of bank 1 which you can read using the ROTP command. It contains 16 halfwords and 15 are user-programmable.

To read the OTP sector:

1. Write a Read OTP sector command (98h) to any word address in Bank 1.
2. Read any of the OTP words see Reading the OTP words can be non-sequential. To access the OTP word, read from any address in Bank 1, using the low byte of the address to select the word.
3. To exit ROTP mode, write a Read Array command (FFh) to any word address in Bank 1 to return it to Read Array mode.

Table 9. OTP word addressing

OTP word	Bank 1 word address LSB
OTP Word 0	00h
OTP Word 1	01h
OTP Word 2	02h
OTP Word 3	03h
OTP Word 4	04h
OTP Word 5	05h
OTP Word 6	06h
OTP Word 7	07h

1.11.4 RD read array (FFh)

Read array is the normal operating mode for reading application data or executing code from a Flash bank. When a device reset occurs, the banks are in Read Array mode by default. However when you write some CUI commands to the bank they change the operating mode (for example to Program or Erase mode). When these operations are completed, you use the Read Array command to put the bank back into Read Array mode. A Read Array command will be ignored while a bank is programming or erasing. However in the other bank, a Read Array command will be accepted.

To put the bank in Read Array mode:

- Write a Read Array command (FFh) to any word address in the bank. The status of the other bank is not affected by the command.

1.12 Write operations

When programming or writing to a Flash Bank, typically a "write" CUI command is issued and followed by a "read" status register command. This "write" then "read" order must be followed to program the Flash properly.

Caution: This requires bit 18 (Instruction TCM order bit) in the Configuration Control Register of the ARM966E-S core to be set. This can be done by the following assembler code :

MOV R0, #0x40000

MCR P15,0x1,R0,C15,C1,0

When set, the write and read to the Flash Bank are performed in the order generated by the ARM966-ES core.This ensures that writes are committed to the Flash memory before any subsequent read.

1.12.1 SE sector erase setup (20h)

Erasing a sector sets all the bits in the selected sector to '1'. You can erase one sector at a time. You do not have to pre-program the sector, this is done automatically before erasing.

To erase a sector:

1. Write a Sector Erase Set-up command (20h) to any word address in the sector to be erased.
2. Write an Erase Confirm command (D0h) to any word address in the sector to be erased. If the second command given is not an Erase Confirm, the command aborts and the ES and PS error flags in the status register are set.
3. After writing the SE command, reading any address within the bank will return the Status Register data.
   - While erasure is in progress the PECS bit in the Status Register is ‘0’. When erasure is completed the PECS bit is ‘1’.
   - The ES bit in the Status Register returns ‘1’ if there has been an erase failure.
   - If the sector is protected, the erase operation aborts and the SP bit in the Status Register returns ‘1’
4. While the sector erase is ongoing, the bank with the sector being erased accepts only Read Status Register (70h) and Program/Erase Suspend (B0h) commands.
5. At the end of the sector erase operation:
   - Write a Clear Status Register (50h) command to the bank to reset the Status Register.
   - The bank will remain in Read Status Register mode until you write a Read Array (FFh) command.

1.12.2 PG program setup (40h)

The Program command programs a 16-bit array (halfword). Only one bank can be programmed at a time, the other bank must be in one of the read modes or in program/erase suspend mode (see *Table 6: Rules for dual bank operations*).

To program a halfword:

1. Write a Program Set-up command (40h) to any word address in the bank to be programmed.
2. Write the data halfword to be programmed to the destination address in the bank.The address must be pointing to the halfword location where the data is programmed.
3. After writing the data, reading any address within the bank will return the Bank Status Register data.
   - While programming is in progress the PECS bit in the Bank Status Register is '0'. When programming is completed the PECS bit is '1'.
   - The PS bit in the Bank Status Register returns '1' if there has been a programming error.
   - If the bank is protected, the program operation aborts and the SP bit in the Status Register returns '1'
4. While programming is ongoing, the bank accepts only Read Status Register (70h) and Program/Erase Suspend (B0h) commands.
5. At the end of the program operation:
   - Write a Clear Status Register (50h) command to the bank to reset the Status Register.
   - Write a Read Array (FFh) command to return to normal operations.

Note: Programming aborts if the MCU is powered down or a reset occurs. As data integrity cannot be guaranteed when the program operation is aborted, the sector containing the memory location must be erased and reprogrammed.

1.12.3 CLSR clear status register (50h)

Each bank has a Status Register which indicates when a program or erase operation is complete and the success or error status. (see also *Section 1.13.1: Bank status register*).

To clear the Status Register:

1. Write a Clear Status Register command (50h) to any word address in the bank.
2. After the Clear Status Register command is executed, the bank returns automatically to Read Array mode.

1.12.4 CR write Flash configuration register (60h)+(03h)

The Flash Configuration Register is located in the user configuration sector of Bank 1 which you can read using the RSIG command.

To program the Flash Configuration Register:

1. Write a Write Flash Configuration Register command (60h) to any word address in Bank 1.
2. Write a Write Flash Configuration Register Confirm command (03h) to an address in Bank 1. The A[12:9] and A4 address bit values are written to the Flash Configuration register. However the reserved bit values cannot be changed. Refer to *Section 1.13.6*.
3. At the end of the command, the memory returns to Read mode as if a Read Array command had been issued

1.12.5 PRP program protection level 1 register (60h)+(01h) or (D0h)

The Protection Level 2 Register and Protection Level 1 registers are located in the user configuration sector of bank 1 which you can read using the RSIG command See *Table* . The Protection Level 2 Register can only be programmed via JTAG. Refer to *Section 1.13.2* and *Section 1.13.4* for the register descriptions.

To program the Protection Level 1 Register:

1. Write a Program Set-up command (60h) to any word address in the sector you want to protect/unprotect.
2. Write a Protect Confirm command (01h) to any word address in the sector you want to protect or Unprotect Confirm (D0h) if you want to unprotect it.

1.12.6 BE bank erase setup (80h)

Bank erasure sets all the bits within the selected bank to '1'. It is not necessary to pre-program the bank as this is done automatically before erasing.

To erase a bank:

1. Write a Bank Erase Set-up command (80h) to any word address in the bank to be erased.
2. Write an Erase Confirm command (D0h) to any word address in the bank to be erased. If the second command given is not an Erase Confirm, the command aborts and the ES and PS error flags in the status register are set.
3. After writing the BE command, reading any address within the bank will return the Status Register data.
   - While erasure is in progress the PECS bit in the Status Register is ‘0’. When erasure is completed the PECS bit is ‘1’.
   - The ES bit in the Status Register returns ‘1’ if there has been an erase error.
   - If the bank is protected, the erase operation aborts and the SP bit in the Status Register returns ‘1’
4. While the bank erase is ongoing, the bank accepts only Read Status Register (70h) and Program/Erase Suspend (B0h) commands.
5. At the end of the Bank Erase operation:
   - Write a Clear Status Register (50h) command to the bank to reset the Status Register.
   - The bank will remain in Read Status Register mode until you write a Read Array (FFh) command.

1.12.7 PES program/erase suspend (B0h)

To suspend an ongoing Program or Erase operation:

- Write a Program/Erase Suspend command (B0h) to any word address in the bank being programmed or erased.

Erase suspend

The Erase Suspend command freezes the erase operation (after a latency period of < 25us) and allows you to read or program in either of the two banks.

You can read the Bank Status register after the erase suspend is issued. The PECS and ESS bits in the Bank Status register are set to ’1’ when the erase operation has been suspended. The ESS bit is cleared if the erase is completed or in progress.

The valid commands while erase is suspended are: Program/Erase Resume, Program, Read Array, Read Status Register, Read Electronic Signature, Sector Protect, and Sector Unprotect. You can protect the sector being erased by issuing the Sector Protect command.

During Erase Suspend, the Flash goes into standby mode which reduces power consumption. Erase is aborted if the device is powered off.

Program suspend

The Program Suspend command freezes the ongoing programming operation.

The PSS bit in the Bank Status Register is set to '1' (within 5µs) if the program operation has been suspended. The PSS bit is cleared if the program operation is completed or in progress.

The valid commands while program is suspended are: Program/Erase Resume, Read Array, Read Status Register, Read Electronic Signature. During program suspend mode, the Flash goes into standby mode which reduces power consumption. The Program operation is aborted if the device is powered off.

1.12.8 LOTP lock OTP sector (C0h)

The Lock OTP command allows you to modify the LOCK bit in the Protection Level 2 Register When it is set, the OTP sector is protected from any further write access. Refer to *Section 1.13.2*.

To program the LOCK bit:

1. Write a Write OTP command (C0h) to any word address in Bank 1.
2. Write the data value 01h to any word address in Bank 1 with 08h in the low byte of the address.
3. After writing the data, reading any address within the bank 1 will return the Bank 1 Status Register data.
   - While programming is in progress the PECS bit in the Bank 1 Status Register is ‘0’. When programming is completed the PECS bit is ‘1’.
   - The PS bit in the Bank 1 Status Register returns ‘1’ if there has been a programming error.
   - If the OTP sector is already locked, the program operation aborts and the SP bit in the Status Register returns ‘1’
4. While programming is ongoing, Bank 1 accepts only Read Status Register (70h) and Program/Erase Suspend (B0h) commands.
5. At the end of the Lock OTP Sector operation:
   - Write a Clear Status Register (50h) command to the bank to reset the Status Register.
   - Write a Read Array (FFh) command to return to normal operations.

1.12.9 WOTP write OTP sector (C0h)

The Write OTP command programs a halfword in the OTP sector. Once an OTP halfword is written, it cannot be erased. The other “unprogrammed” halfword can still be written at a

later time. If the LOCK bit in the Protection Level 2 Register is set, the OTP sector is protected from any further write access. Refer to *Section 1.13.2*. You can modify the LOCK bit using the LOTP command (see *Section 1.12.8*) or via the JTAG interface.

To program a halfword in the OTP sector:

1. Write a Write OTP command (C0h) to any word address in Bank 1.
2. Write the low or high halfword to be programmed to any address in Bank 1 using the low byte of the word address to select the OTP word. Refer to Writing the OTP locations can be non-sequential.
3. After writing the data, reading any address within the bank 1 will return the Bank 1 Status Register data.
   - While programming is in progress the PECS bit in the Bank 1 Status Register is '0'. When programming is completed the PECS bit is '1'.
   - The PS bit in the Bank 1 Status Register returns '1' if there has been a programming error.
   - If the OTP sector is protected, the program operation aborts and the SP bit in the Status Register returns '1'
4. While programming is ongoing, Bank 1 accepts only Read Status Register (70h) and Program/Erase Suspend (B0h) commands.
5. At the end of the program operation:
   - Write a Clear Status Register (50h) command to the bank to reset the Status Register.
   - Write a Read Array (FFh) command to return to normal operations.

1.12.10 PER program/erase resume (D0h)

Erase resume

If an Erase Suspend command was previously executed, the erase operation may be resumed by writing the D0h command to a word address within the suspended bank. When erase resumes:

- The PECS and ESS bits in the Bank Status register are cleared.
- Read operations to the bank being erased will return the Bank Status Register.

Erase cannot resume until program operations initiated during Erase Suspend have completed. You can also nest suspends as follows:

1. Suspend erase in the first bank
2. Start programming in the second or in the same bank
3. Suspend programming
4. Then read from the second or the same bank

Program resume

If a Program Suspend instruction was previously executed, the Program operation may be resumed by issuing the D0h command using an address within the suspended bank. When programming resumes,

- The PECS and PSS bits in the Bank Status register are cleared.
- Read operations to the bank being programmed will return the Bank Status Register.

1.13 CUI registers

1.13.1 Bank status register

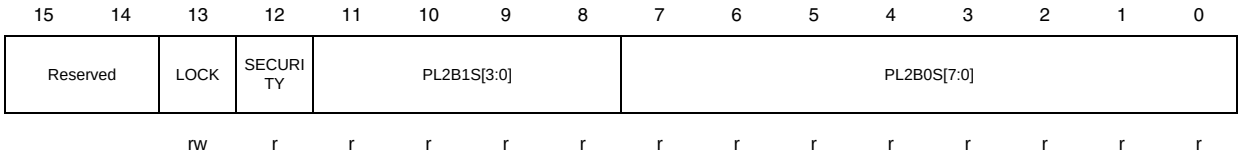
This register can be read using the RSR command (see *Section 1.11.1*). It can be cleared using the CLSR command (see *Section 1.12.3*)

7	6	5	4	3	2	1	0
PECS	ESS	ES	PS		PSS	BPS	
rc	rc	rc	rc		rc	rc	

Bit 7	**PECS**: *FPEC Status.* This bit indicates the status of the Program/Erase Controller (FPEC). You can poll it to check the progress of FPEC operations. It is set on completion of a Program or Erase operation. The PS and ES bits indicate success or failure. 0: Busy 1: Ready
Bit 6	**ESS**: *Erase Suspend Status.* This bit is set when you write an Erase Suspend command. It remains set until an Erase Resume command is received. 0: Erase in progress or completed 1: Erase suspended
Bit 5	**ES**: *Erase Status.* This bit is set to “1” if the FPEC has applied the max. number of erase pulses to the sector without a successful verify. 0: Erase successful 1: Erase failure
Bit 4	**PS**: *Program status.* This bit is set to “1” if the FPEC has failed to program a word. 0: Program successful 1: Program failure
Bit 3	Reserved.
Bit 2	**PSS**: *Program suspend status.* This bit is set when you write an Program Suspend command. It remains set until an Program Resume command is received. 0: Programming in progress or completed 1: Programming suspended
Bit 1	**SP**: *Sector protection status.* This bit is set to “1” if a Program or Erase operation has been attempted on a protected sector 0: No Protection error 1: Program/Erase attempted on protected sector
Bit 0	Reserved.

1.13.2 Protection level 2 register (STR91xFAxx2 and STR91xFAxx4)

This register can be read using the RSIG command (see *Section 1.11.2*) or by JTAG. The LOCK bit can be written via CUI or JTAG. Other bits can be written only via JTAG. The factory default settings may be modified by the user via JTAG. The register is not cleared by a device reset.



Bits 15:14	Reserved.
Bit 13	**LOCK**: *OTP sector lock bit.* See also *Section 1.12.8*. 0: OTP sector not locked (default) 1: OTP sector locked
Bit 12	**SECURITY**: *JTAG Access Security Bit.* 0: JTAG access allowed (default) 1: JTAG access not allowed
Bits 11:8	**PL2B1S[3:0]**: *Level 2 Protection Status on Bank 1 Sector x.* 0: Sector x of Bank 1 is not Level 2 write protected (default) 1: Sector x of Bank 1 is Level 2 write protected
Bits 7:0	**PL2B0S[7:0]**: *Level 2 Protection Status on Bank 0 Sector x.* 0: Sector x of Bank 0 is not Level 2 write protected (default) 1: Sector x of Bank 0 is Level 2 write protected

1.13.3 Protection level 2 register (STR91xFAxx6 and STR91xFAxx7)

This register can be read using the RSIG command (see *Section 1.11.2*) or by JTAG. The LOCK bit can be written via CUI or JTAG. Other bits can be written only via JTAG. The factory default settings may be modified by the user via JTAG. The register is not cleared by a device reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PL2B0S[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PL2B0S[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:0	**PL2B0S[7:0]**: *Level 2 Protection Status on Bank 0 Sector x.* 0: Sector x of Bank 0 is not Level 2 write protected (default) 1: Sector x of Bank 0 is Level 2 write protected

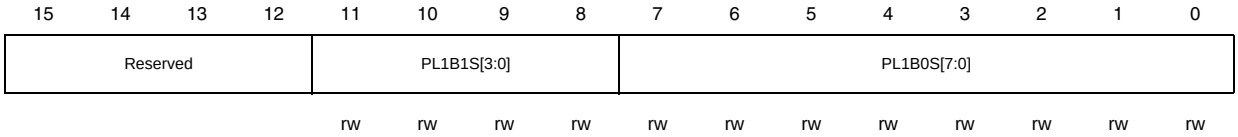
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						LOCK	SECURITY	PL2B1S[7:0]							
						rw	r	r	r	r	r	r	r	r	r

Bits 15:10	Reserved.
Bit 9	**LOCK**: *OTP sector lock bit.* See also *Section 1.12.8*. 0: OTP sector not locked (default) 1: OTP sector locked
Bit 8	**SECURITY**: *JTAG Access Security Bit.* 0: JTAG access allowed (default) 1: JTAG access not allowed
Bits 7:0	**PL2B1S[3:0]**: *Level 2 Protection Status on Bank 1 Sector x.* 0: Sector x of Bank 1 is not Level 2 write protected (default) 1: Sector x of Bank 1 is Level 2 write protected

1.13.4 Protection level 1 register (STR91xFAxx2 and STR91xFAxx4)

This register can be read using the RSIG command (see *Section 1.11.2*). It can be programmed via PRP command (see *Section 1.12.5*) or via JTAG.

If a sector is write protected in the Protection Level 2 register it cannot be unprotected via PRP command. See also *Section 1.9*. The register is set to default value at power up or by Low Voltage Detect Reset. System reset (external or watchdog reset) will not change the register value).



Bits 15:12	Reserved.
Bits 11:8	**PL1B1S[3:0]**: *Level 1 Protection Status on Bank 1 Sector x.* 0: Sector x of Bank 1 not write protected 1: Sector x of Bank 1 is Level 1 write protected (default)
Bits 7:0	**PL1B0S[7:0]**: *Level 1 Protection Status on Bank 0 Sector x.* 0: Sector x of Bank 0 not write protected 1: Sector x of Bank 0 is Level 1 write protected (default)

1.13.5 Protection level 1 register (STR91xFAxx6 and STR91xFAxx7)

This register can be read using the RSIG command (see *Section 1.11.2*). It can be programmed via PRP command (see *Section 1.12.5*) or via JTAG.

If a sector is write protected in the Protection Level 2 register it cannot be unprotected via PRP command. See also *Section 1.9*. The register is set to default value at power up or by Low Voltage Detect Reset. System reset (external or watchdog reset) will not change the register value).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PL1B0S[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PL1B0S[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0	**PL1B0S[7:0]**: *Level 1 Protection Status on Bank 0 Sector x.* 0: Sector x of Bank 0 not write protected 1: Sector x of Bank 0 is Level 1 write protected (default)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PL1B1S[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 7:0	**PL1B1S[7:0]**: *Level 1 Protection Status on Bank 1Sector x.* 0: Sector x of Bank 0 not write protected 1: Sector x of Bank 0 is Level 1 write protected (default)

1.13.6 Flash configuration register

This register can be read using the RSIG command (see *Section 1.11.2*). It can be written using the CR command (see *Section 1.12.4*). Reserved bit values cannot be modified. .The register is set to default value at power up or by Low Voltage Detect Reset. System reset (external or watchdog reset) will not change the register value).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			WSTATES[1:0]		PWD	LVD EN	reserved				BUS CFG	reserved			
			rw	rw	rw	rw					rw				

Bits 15:13	Reserved, default value =001b.
Bits 12:11	**WSTATES[1:0]**: *Wait states.* These bits define the number of wait states inserted in asynchronous read accesses. 00: 1 Wait state (default) 01: 2 Wait states 10: 3 Wait states **Note:** Wait States are inserted only for non-bursting Flash read bus cycles. One wait state is required for a FMI bus clock frequency of 66 MHz or less. Two wait states are required for 75 MHz bus clock frequency.
Bit 10	**PWD**: *Power-down configuration.* 0: Power-down disabled 1: Power-down enabled (default)
Bit 9	**LVDEN**: *Low Voltage Detector enable.* This bit indicates if the LVD (reset and early warning interrupt) is disabled 0: LVD enabled (default) 1: LVD disabled
Bits 8:5	Reserved, default value = 0110b
Bit 4	**BUSCFG**: *Flash bus clock configuration.* This bit selects the FMI Flash bus clock configuration. It can be set by software to configure the bus for frequencies greater than 66 MHz. 0: Low frequency bus clock (default) 1: Bus clock speed > 66 MHz
Bits 3:0	Reserved, default value =1111b.

1.14 CUI command summary

The Flash memory has a “word address bus”, that means every address points to a word in the memory (CPU address A[25:2] are mapped to Flash address A[23:0]).

When writing a command to the Flash memory, the command or Data byte must be placed on D[7:0] of the FMI memory bus. This requires the bank or sector address to be at a word boundary (word address), except when programming a halfword. Refer to *Section 1.12.2* on halfword programming.

Table 10. Command summary

	Command	Cycles	Operation	Address(1)	Data	Operation	Address(1)	Data
Read	RD Read array	1+	Write	Bank Addr	FFh	Read	Read Addr	Data
	RSR Read Status Register	1+	Write	Bank Addr	70h	Read	Bank Addr	Status Register
	RSIG Read Electronic Signature	1+	Write	Bank 1 Addr	90h	Read	EA	ED
	ROTP Read OTP sector	1+	Write	Bank 1 Addr	98h	Read	OTP Addr	OTP Data
	CLRS Clear Status Register	1	Write	Bank Addr	50h			
Program /E rase	SE Sector Erase	2	Write	Sector Addr	20h	Write	Sector Addr	D0h
	BE Bank Erase	2	Write	Bank Addr	80h	Write	Bank Addr	D0h
	PG Program Halfword	2	Write	Word Addr	40h	Write	Halfword Addr	Halfword
	PES Program Erase Suspend	1	Write	Bank Addr	B0h			
	PER Program Erase Resume	1	Write	Bank Addr	D0h			
Protect	SP Sector Protect	2	Write	Sector Addr	60h	Write	Sector Addr	01h
	BU Sector Unprotect	2	Write	Sector Addr	60h	Write	Sector Addr	D0h
Config.	CR Write Read Configuration Register	2	Write	Bank 1 Addr	60h	Write	Read Config. Addr (2)	03h
	WOTP Program OTP Sector	2	Write	Bank 1 Addr	C0h	Write	PA (00-07h)(3)	PD
	LOTP Program OTP Lock bit	2	Write	Bank 1 Addr	C0h	Write	PA (08h)(3)	01h

1. Address must be within the Bank 0 or 1 memory range as defined in the FMI registers
2. Refer to *Section 1.12.4* for the Read Configuration address.
3. Refer to *Section 1.11.3* for the OTP programming address.

Legend:
OTP and Electronic Signature reside in Bank 1 (Secondary Flash)
PA= OTP register address
PD= OTP data
EA= Electronic Signature register address
ED= Electronic Signature data (see *Section*)

1.15 FMI register description

The FMI Registers configure the size and base address of Bank 0 and Bank 1. The address ranges of Bank 0 and Bank 1 must not overlap each other.

The microcontroller boots from Bank 0 by default:

In the default configuration:

- Bank 0 is the Boot Bank, after reset the application program has to write the size and base address of Bank 0 in the FMI_BBSR and FMI_BBADR registers
- Bank 1 is the Non-Boot Bank, after reset the application program has to write the size and base address of Bank 1 in the FMI_NBBSR and FMI_NBBADR registers

Booting from Bank 1

The STR91FA microcontroller can also boot from Bank 1. The selection of the Boot Bank can be modified using JTAG.

If Bank 1 is the Boot bank, after reset, the application program has to write the size and start address of Bank 1 in the FMI_BBSR and FMI_BBADR registers and the size and start address of Bank 0 in the FMI_NBBSR and FMI_NBBADR registers.

1.15.1 Boot bank size register (FMI_BBSR)

Address: 5400 0000h

Reset value: 0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												BBSIZE[3:0]			
												rw	rw	rw	rw

Bits 31:4	Reserved, always read as 0
Bits 3:0	**BBSIZE[3:0]**: *Boot bank size* These bits are set and cleared by software. They define the address space for the boot bank, *Boot bank size = 2 $^{BBSIZE[3:0]}$ x 32Kbytes.* 0000: 32 Kbytes. 0001: 64 Kbytes .... 1011: 64 Mbytes Other values are reserved.

1.15.2 Non-boot bank size register (FMI_NBBSR)

Address: 5400 0004h

Reset value: 0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												NBBSIZE[3:0]			
												rw	rw	rw	rw

Bits 31:4	Reserved, always read as 0
Bits 3:0	**NBBSIZE[3:0]**: *Non-boot bank size* These bits are set and cleared by software. They define the address space for the non booting memory bank,*Boot bank size = 2 $^{NBBSIZE[3:0]}$ x 8Kbytes.* 0000: 8 Kbytes. 0001: 16 Kbytes .... 1101: 64 Mbytes Other values are reserved.

1.15.3 Boot bank base address register (FMI_BBADR)

Address Offset: 5400 000Ch

Reset value: 0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								BBADDR[23:16]							
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BBADDR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:24	Reserved, always read as 0
Bits 23:0	**BBADDR[23:0]:** *Boot bank base address* These bits are set and cleared by software. They define the base address of the boot bank. The Flash Bank address BBADDR[23:0] is a word address and is mapped to CPU core address A[25:2]. The boot address defaults to "0" at reset. If the Boot Bank is re-mapped later to a different address, the base address must be at a bank size boundary of the remapped bank. Example (STR91xFAxx4) : for For Bank0, it must be at a 512KB boundary and at a 32KB boundary for Bank1.

1.15.4 Non-boot bank base address register (FMI_NBBADR)

Address Offset: 5400 0010h

Reset value: 0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								NBBADDR[23:16]							
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NBBADDR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:24	Reserved, always read as 0
Bits 23:0	**NBBADDR[23:0]**: *Non-boot bank base address* These bits are set and cleared by software. They define the base address of the non-boot bank. The Flash Bank address NBBADDR[23:0] is a word address and is mapped to CPU core address A[25:2]. The Non-boot bank base address must be at a Non-boot bank size boundary. Example (STR91xFAxx4) : at 32KB boundary for Bank1 and for Bank 0, it must be at a 512KB boundary.

1.15.5 FMI control register (FMI_CR)

Address Offset: 5400 0018h

Reset value: 0000 0008h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							WWS	OMIE	BEIE	Res.	NBB EN	BBEN	Reserved		
							rw	rw	rw	rw	rw	rw			

Bits 31:9	Reserved, always read as 0
Bit 8	**WWS**: *Write Wait States* This bit is set and cleared by software. It defines the number of wait states in Flash write access. 0: Flash write is active for 1 clock cycle (Recommended setting) 1: Flash write is active for 2 clock cycles (Reserved for future use)
Bit 7	**OMIE**: *Out of Memory interrupt enable* This bit is set and cleared by software. It enables/disables the Out of Memory interrupts. 0: Disabled 1: Enabled. An interrupt is generated when the OM bit in the FMI_SR register is set.
Bit 6	**BERRIE**: *Flash Bank Error interrupt enable* This bit is set and cleared by software. It enables/disables Flash bank error interrupts. 0: Disabled 1: Enabled. An interrupt is generated when the B1ERR or B0ERR bit in the FMI_SR register are set.

Bit 5	Reserved, always read as 0
Bit 4	**NBBEN**: *Flash Non Boot Bank enable* This bit is set and cleared by software. It enables/disables Flash Non Boot bank. 0: Disabled 1: Enabled.
Bit 3	**BBEN**: *Flash Boot Bank enable* This bit is set and cleared by software. It enables/disables Flash Boot bank. 0: Disabled 1: Enabled
Bits 2:0	Reserved, always read as 0

1.15.6 FMI status register (FMI_SR)

Address Offset: 5400 001Ch

Reset value: 0000 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											PFQ BCEN	OM	Res.	B1ERR	B0ERR
											r	rc_w1		rc_w1	rc_w1

Bits 31:5	Reserved, always read as 0
Bit 4	**PFQBCEN**: *PFQBCEN Status* This bit is set and cleared by harware. 0: PFQ/BC disabled (bypassed) 1: PFQ/BC enabled
Bit 3	**OM**: *Out of Memory error* This bit is set by hardware and cleared by software writing 1. It indicates that an access was made outside the configured memory area. An interrupt is generated if the OMIE bit in the FMI_CR regsiter is set. 0: No OM error 1: An Out of Memory error occurred
Bit 2	Reserved, always read as 0
Bit 1	**B1ERR**: *Flash Bank 1 error* This bit is set by hardware and cleared by software writing 1. It indicates that an access was made to Bank 1 while it was disabled. An interrupt is generated if the BERRIE bit in the FMI_CR register is set. 0: No B1ERR error 1: A Flash Bank 1 error occurred
Bit 0	**B0ERR**: *Flash Bank 0 error* This bit is set by hardware and cleared by software writing 1. It indicates that an access was made to Bank 0 while it was disabled. An interrupt is generated if the BERRIE bit in the FMI_CR register is set. 0: No B0ERR error 1: A Flash Bank 0 error occurred

1.15.7 BC fifth entry target address register (FMI_BCE5ADDR)

Address Offset: 5400 0020h

Reset value: 0000 0006h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								BCE5ADDR[23:16]							
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCEADDR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:24	Reserved, always read as 0
Bits 23:0	**BCE5ADDR[23:0]**: *Branch Cache Fifth Entry Target Address* These bits are set and cleared by software. They define the target address of the BC 5th entry, provided to implement interrupt (IRQ) mode or any “special” branch not subject to the LRU algorithm. Defaults to 0x00000006 at reset i.e. IRQ exception at 0x18.

1.16 STR91x in-application programming (IAP)

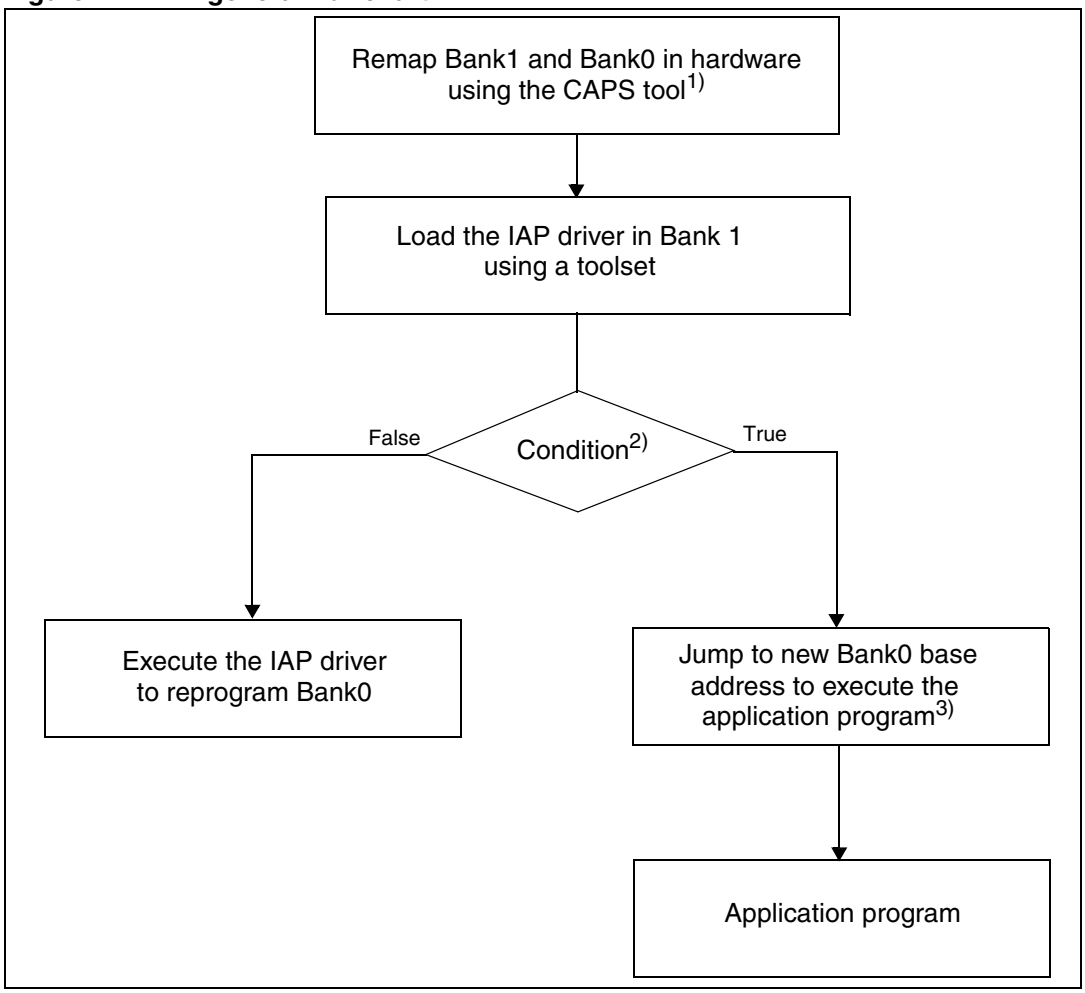
An important requirement for most Flash-based systems is the ability to update firmware while the system is installed in the end product. This is referred to as In-Application Programming (IAP).

STR91x MCUs have the capability of running user-specific firmware to perform In Application Programming of the MCU embedded Flash memory. This feature allows the use of any type of communication protocol for the reprogramming process (for example, CAN, UART, USB etc …).

The principle is to program the user application in Bank0 using code located in Bank1.

The diagram below outlines general STR91x IAP guidelines.

Figure 2. IAP general flowchart



1. Bank1 is remapped in hardware to address 0x00 and Bank0 is remapped in hardware to 0x80000 (if Bank0 size is 256 or 512 Kbytes) and to 0x200000 (if Bank0 size is 1 or 2 Mbytes). The base address must be at a 32 Kbyte boundary for Bank1. For Bank0, it must be at a 512 Kbyte boundary or at a 2 Mbyte boundary.
2. The condition can be, for example, a pin level allowing either to execute the IAP driver located in Bank1 or to jump to the user application located in Bank0 after being loaded using IAP.
3. If the user application doesn't contain interrupts, there is no problem. A problem is encountered when the user application (remapped to the new address) contains interrupts because the interrupt vector is located at address 0x00. Consequently, interrupts must be redirected to Bank0.

2 JTAG interface

An IEEE-1149.1 JTAG interface on the STR91xFA provides In-System-Programming (ISP) of all memory, boundary scan testing of pins, and the capability to debug the CPU. Six pins are used on this JTAG serial interface. The five signals JTDI, JTDO, JTMS, JTCK, and JTRSTn are all standard JTAG signals complying with the IEEE-1149.1 specification. The sixth signal, JRTCK (Return TCK), is an output from the STR91xFA and it is used to pace the JTCK clock signal coming in from the external JTAG test equipment for debugging. The frequency of the JTCK clock signal coming from the JTAG test equipment must be at least 10 times less than the ARM966E-S CPU core operating frequency. To ensure this, the signal JRTCK is output from the STR91xFA and is input to the external JTAG test equipment to hold off transitions of JTCK until the CPU core is ready, meaning that the JTAG equipment cannot send the next rising edge of JTCK until the equipment receives a rising edge of JRTCK from the STR91xFA.

The JTAG test equipment must be able to interpret the signal JRTCK and perform this adaptive clocking function. If it is known that the CPU clock will always be at least ten times faster than the incoming JTCK clock signal, then the JRTCK signal is not needed.

The two die inside the STR91xFA (CPU die and Flash memory die) are internally daisy-chained on the JTAG bus, see Figure 1 below. The CPU die has two JTAG Test Access Ports (TAPs), one for boundary scan functions and one for ARM CPU debug. The Flash memory die has one TAP for program/erase of non-volatile memory. Because these three TAPs are daisy-chained, only one TAP will converse on the JTAG bus at any given time while the other two TAPs are in BYPASS mode. The TAP positioning order within this JTAG chain is the boundary scan TAP first, followed by the ARM debug TAP, followed by the Flash TAP. All three TAP controllers are reset simultaneously by one of two methods:

- A chip-level global reset, caused only by a Power-On-Reset (POR) or a Low Voltage Detect (LVD).
- A reset command issued by the external JTAG test equipment. This can be the assertion of the JTAG JTRSTn input pin on the STR91xFA or a JTAG reset command shifted into the STR91xFA serially.

The JRTCK requirement is due to the inclusion of the ARM debug TAP. This TAP is NOT needed during ISP. To speed up programming time along with simplifying this JRTCK requirement, a “Turbo Programming Mode” is introduced. Once this Turbo mode is enabled, the ARM debug TAP is removed from the daisy-chain as shown in Figure 2B. The JRTCK and the 10X frequency requirement (between system clock and JTCK) are no longer required. The proper way to program STR91xFA is to first enable this Turbo Programming Mode. This is the assumption taken for the rest of this document.

Note: Prior to any JTAG programming operation, the STR91xFA RESET_IN pin must be asserted. This is to prevent the CPU from running while programming the device.

Figure 3. JTAG chaining inside the STR91xFA, normal mode

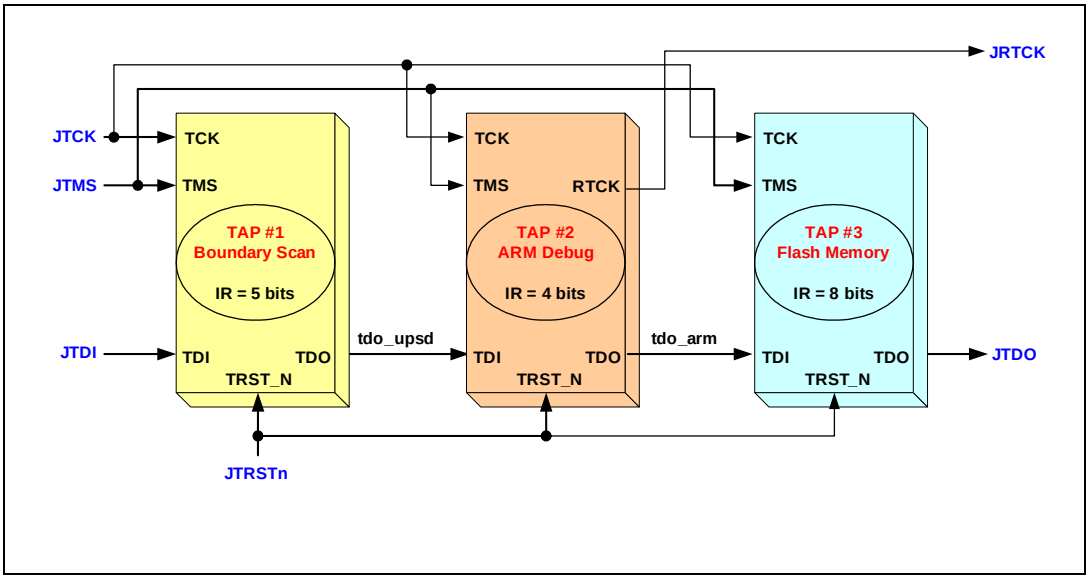
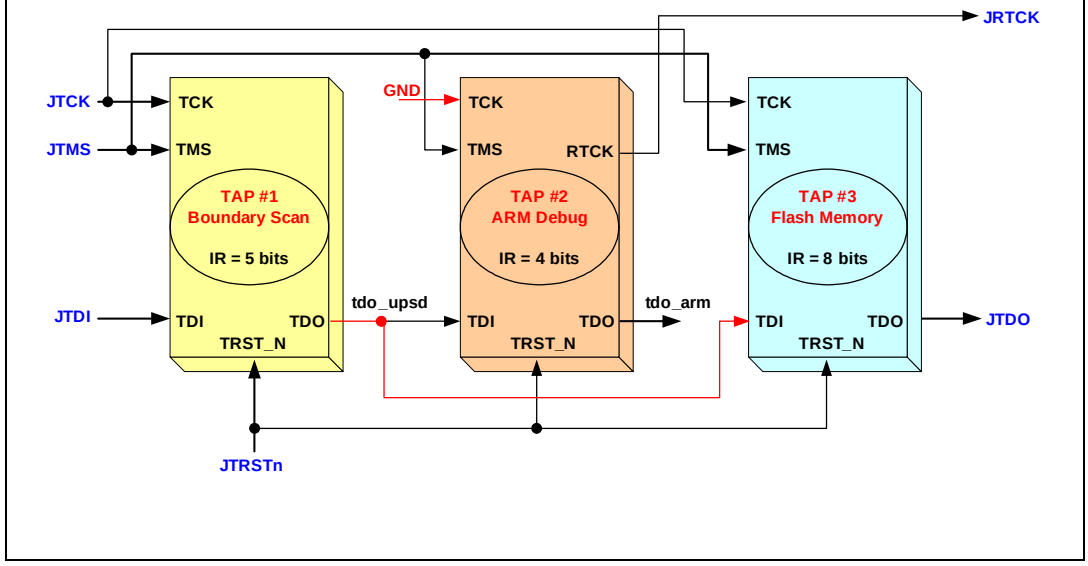


Figure 4. JTAG chaining inside the STR91xFA, turbo programming mode



3 Security status

It is possible to erase the chip after security is enabled. However, this is possible only when doing full chip erase, in which all sectors are selected. Partial erase of some, but not all, sectors are not possible after security is enabled.

4 Checksum calculation

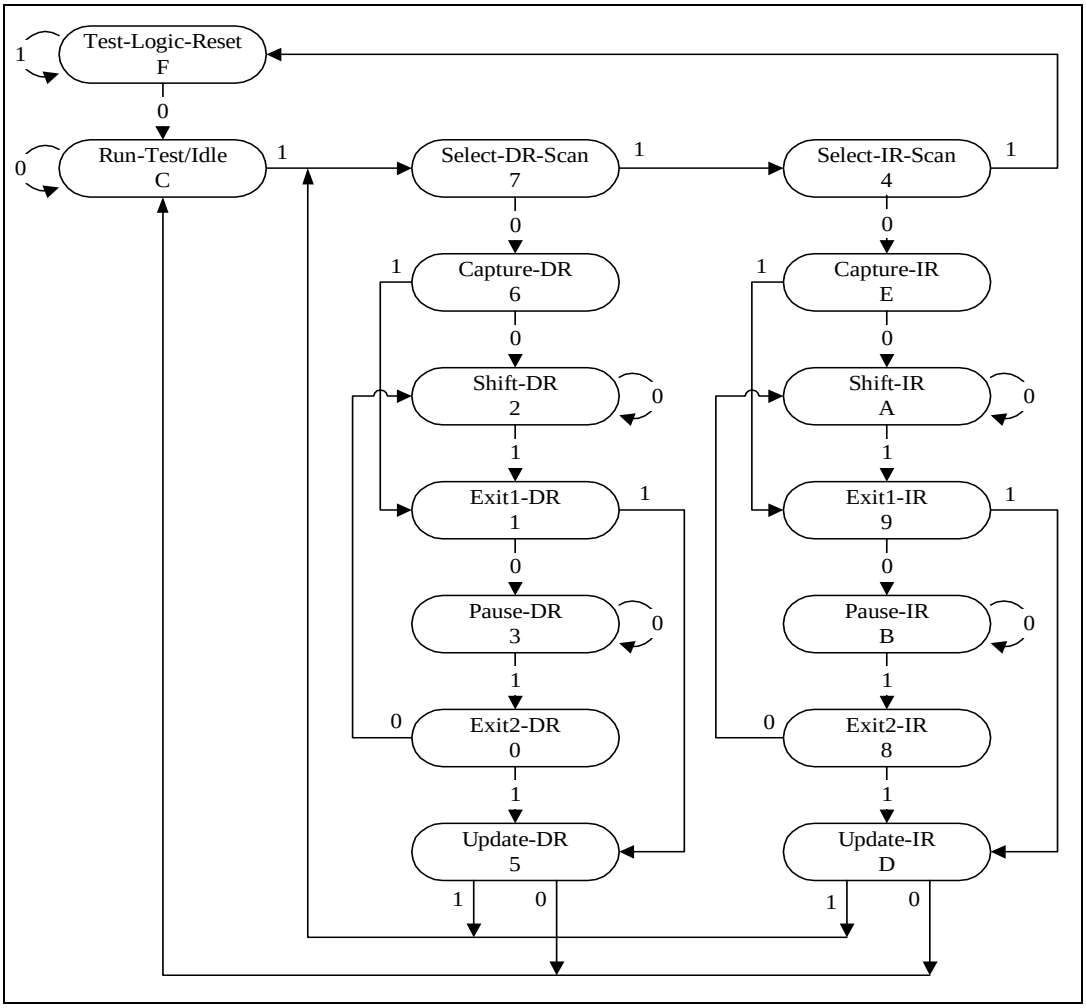
Checksum calculation is done with individual bit binary addition with carry. Data beyond the eighth bit is dropped.

Always mask out or set to zero on the followings:

- Security bit
- OTP_Lock bit
- OTP sector

5 JTAG sequence

Figure 5. JTAG sequence flowchart



Since STR91xFA devices include JTAG enabled CPU device in series with the Flash memory die, it is necessary to insert BYPASS instruction (opcode: all 1's) for the CPU device in all JTAG instruction cycle and account for the extra BYPASS bit(s) during data cycle.

5.1 JTAG timing specification

Table 11. Power up mode

Symbol	Parameter	Value		Unit
		Min	Max	
t_{JTCKL}	JTCK Low	$3(1/f_{CPUCLK})$		ns
t_{JTCKH}	JTCK High	$3(1/f_{CPUCLK})$		ns
t_{JTCKP}	JTCK Period	$10(1/f_{CPUCLK})$		ns
t_{JTSU}	JTDI, JTMS Setup before JTCK High	4		ns
t_{JTHLD}	JTDI Hold after JTCK High	4		ns
$t_{JTMSHLD}$	JTMS Hold after JTCK High	$3(1/f_{CPUCLK})$		ns
t_{JDVAL}	JTCK Low to JTDO Valid		28	ns
f_{JTCK}	JTCK Frequency		$f_{CPUCLK/ 10}$	

Table 12. Turbo mode

Symbol	Parameter	Value		Unit
		Min	Max	
t_{JTCKL}	JTCK Low	32		ns
t_{JTCKH}	JTCK High	32		ns
t_{JTCKP}	JTCK Period	66.7		ns
t_{JTSU}	JTDI, JTMS Setup before JTCK High	4		ns
t_{JTHLD}	JTDI, JTMS Hold after JTCK High	4		ns
t_{JDVAL}	JTCK Low to JTDO Valid		28	ns
f_{JTCK}	JTCK Frequency		15	MHz

5.2 TURBO-PROG-ENABLE

Table 13. TURBO-PROG-ENABLE sequence

Step	TAP activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load Turbo Mode Instruction: "01101 1111 11111111"
2	Run-Test/Idle	End

Note: While executing this Turbo Mode instruction, Turbo Mode is not active yet. As such, JTCK needs to be 10X slower than the system clock. Since the system clock can be anywhere between 4 MHz to 25 MHz, it is best to run JTCK at 400 kHz or below. Once Turbo Mode is active, JTCK can be at a much faster frequency (i.e. 10 MHz).

Once Turbo Mode is activated, it can only be disabled by a power down or by taking JTRSTn to low.

This instruction is NOT IEEE 1149.1 JTAG Standard compliant. There is no direct status check for the success status.

5.3 IDCODE

Table 14. IDCODE sequence

Step	TAP Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + IDCODE
2	DR-Scan	Shift out 1-bit BYPASS + IDCODE
3	Run-Test/Idle	End

5.4 ISC-CONFIGURATION

Table 15. ISC-CONFIGURATION sequence

Step	TAP Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + ISC-CONFIGURATION
2	DR-Scan	Shift out 1-bit BYPASS + sector protect, CSx, LVD Threshold and Lock OTP information
3	Run-Test/Idle	End

5.5 ISC-ENABLE

Table 16. ISC-ENABLE sequence

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + ISC_ENABLE
2	DR-Scan	Shift in1-bit BYPASS + 8-bits don't care
3	Run-Test/Idle	End

5.6 ISC-DISABLE

Table 17. ISC-DISABLE sequence

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + ISC_DISABLE
2	DR-Scan	Shift out 1-bit BYPASS + status
3	Run-Test/Idle	Wait 50uS
4	Test-Logic-Reset	TLR or load ISC_NOOP (or BYPASS) follow by RTI

5.7 ISC-ADDRESS-SHIFT

Table 18. ISC-ADDRESS-SHIFT sequence

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + ISC_ADDRESS_SHIFT
2	DR-Scan	Shift in 1-bit BYPASS + Sector address
3	Run-Test/Idle	End

5.8 ISC-CLR-STATUS

Table 19. ISC-CLR-STATUS sequence

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + ISC_CLR_STATUS
2	DR-Scan	Shift out 1-bit BYPASS + status (should still show error)
3	Run-Test/Idle	Clear status (including module's error), wait 50uS
4	DR-Scan	Shift out 1-bit BYPASS + status – expect no error
5	Run-Test/Idle	End

5.9 ISC-PROGRAM

Table 20. ISC-PROGRAM sequence

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + ISC_ADDRESS_SHIFT
2	DR-Scan	Shift in 1-bit BYPASS + 8 bits sector address
3	IR-Scan	Load 5-bit BYPASS + ISC_PROGRAM
4	DR-Scan	Shift in 1-bit BYPASS + 64 bits of data to be programmed
5	Run-Test/Idle	Program starts, address incremented
Wait for 50us. Begin polling		
6	IR-Scan	Load 5-bit BYPASS + ISC-NOOP
7	DR-Scan	Shift out 1-bit BYPASS + status
Repeat step 7 until ready = 1 Go back to step 3, repeat steps 3-7 until end of sector is reached.		
8	Run-Test/Idle	End

5.10 ISC-PROGRAM-UC

Figure 6. ISC-PROGRAM-UC sequence

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + ISC_PROGRAM_UC
2	DR-Scan	Shift in 1-bit BYPASS + User-Code to be programmed
3	Run-Test/Idle	Program starts
Begin polling		
4	IR-Scan	Load 5-bit BYPASS + ISC-NOOP
5	DR-Scan	Shift out 1-bit BYPASS + status
Repeat step 5 until ready = 1		
6	Run-Test/Idle	End

5.11 ISC-PROGRAM-SECURITY

Table 21. ISC-PROGRAM-SECURITY sequence

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + ISC_ADDRESS_SHIFT
2	DR-Scan	Shift in 1-bit BYPASS + 8-bit dummy address (80 Hex)
3	IR-Scan	Load 5-bit BYPASS + ISC_PROGRAM_SECURITY
4	Run-Test/Idle	Security bit program begins
5	DR-Scan	Shift out 1-bit BYPASS + status
Begin polling		
6	IR-Scan	Load 5-bit BYPASS + ISC-NOOP
7	DR-Scan	Shift out 1-bit BYPASS
Repeat step 7 until ready = 1		
8	Run-Test/Idle	End

5.12 ISC-READ

Table 22. ISC-READ sequence

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + ISC_ADDRESS_SHIFT
2	DR-Scan	Shift in 1-bit BYPASS + 8 bits sector address
3	IR-Scan	Load 5-bit BYPASS + ISC_READ
4	Run-Test/Idle	Wait 5uS; data read; address incremented
5	DR-Scan	Shift out 1-bit BYPASS + 64 bits of data
Repeat steps 4 & 5 until end of sector		
6	IR-Scan	Load 5-bit BYPASS + ISC_NOOP
7	Run-Test/Idle	End

5.13 ISC-ERASE

Table 23. ISC-ERASE sequence

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + ISC_ERASE
2	DR-Scan	Shift in 1-bit BYPASS + sector(s) to be erased. ‘1’ erase; ‘0’ don’t erase
3	Run-Test/Idle	Erase starts
Begin polling		
4	IR-Scan	Load 5-bit BYPASS + ISC-NOOP
5	DR-Scan	Shift out 1-bit BYPASS + status
Repeat step 5 until ready = 1		
6	Run-Test/Idle	End

5.14 ISC-BLANK-CHECK sequence

Table 24. ISC-BLANK-CHECK sequence

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load 5-bit BYPASS + ISC_BLANK_CHECK
2	DR-Scan	Shift in 1-bit BYPASS + sector(s) to be blank checked. ‘1’ check; ‘0’ don’t check
3	Run-Test/Idle	Blank check starts, wait 40ms
4	DR-Scan	Shift out 1-bit BYPASS + sector blank info. ‘1’ = Sector selected and blank ‘0’ = Sector either not selected or not blank.
5	IR-Scan	Load 5-bit BYPASS + ISC_NOOP (to replace ISC_BLANK_CHECK)
6	Run-Test/Idle	End

6 Full chip erase operation

Table 25. Full chip erase

Description	Flow in sequence
Check Flash Memory JTAG Device ID	IDCODE If ID not matched, exit with error
Enable Turbo Programming Mode	TURBO-PROG-ENABLE
Enter ISC mode	ISC-ENABLE

Full-chip erase	ISC-ERASE (*all*)
	if failed,
	{
	ISC-CLR-STATUS
	report erase rror
	}

Disable ISC mode	ISC-DISABLE
Disable Turbo Programming Mode	Disabling by either power-down or taking JTRSTn to low

7 Erase and program operation

Table 26. Erase and program operation description

Description	Flow in sequence
Check Flash Memory JTAG Device ID	IDCODE if ID not matched, exit with error
Enable Turbo Programming Mode	TURBO-PROG-ENABLE
Enter ISC mode	ISC-ENABLE

Full-chip erase	ISC-ERASE (*all*)
	if failed,
	{
	ISC-CLR-STATUS
	report error and exit
	}

Program the following sectors:
a. Main Flash (non-blank only)
b. Secondary Flash (non-blank only)
c. Configuration
d. User-Code
	program one sector at a time
	ISC-PROGRAM (Sector, data)
	if failed
	{
	ISC-CLR-STATUS
	report error and exit
	}

Table 26. Erase and program operation description (continued)

Description	Flow in sequence

If OTP sector is selected,
 	
Check if OTP_LOCK bit = “1”,
- If yes, exit with warning
- If no,
(a) upload OTP sector from DUT
(b) copy the last 2 bytes (16 bits) from uploaded buffer and overwrite to the program buffer
(c) program OTP sector and OTP_LOCK bit, if applicable.
	/ Check if OTP_LOCK bit is set
	ISC-CONFIGURATION
	if OTP_LOCK bit = “1”,
	exit with message indicating OTP Lock-bit status
	
	/upload OTP sector from DUT
	ISC-READ(OTP Sector)
	if read error, exit with error
	else copy the last 2 bytes (16 bits) from uploaded buffer and overwrite to the program buffer
	
	/ program OTP sector
	ISC-PROGRAM (OTP Sector, data)
	if failed
	{
	ISC-CLR-STATUS
	exit with error
	}
	
	/If file has OTP_LOCK bit = “1”, program OTP_LOCK bit
	ISC-PROGRAM (Lock-bit Sector, data)
	if failed
	{
	ISC-CLR-STATUS
	exit with error
	}

If security bit is set (D0=0), program security-bit	/program security bit
	ISC-PROGRAM-SECURITY
	if failed
	{
	ISC-CLR-STATUS
	report error and exit
	}

Disable ISC mode	ISC-DISABLE
Disable Turbo Programming Mode	Disabling by either power-down or taking JTRSTn to low

8 Verify operation

Table 27. Verify operation description

Description	Flow in sequence
Check Flash Memory JTAG Device ID	IDCODE if device is secured, exit with error if ID not matched, exit with error
Enable Turbo Programming Mode	TURBO-PROG-ENABLE
Enter ISC mode	ISC-ENABLE
Verify the following sector: a. Main Flash b. Secondary Flash c. OTP - do not compare the last 2 bytes (16 bits)	/verify one sector at a time /repeat until all sectors are verified ISC-READ (Sector) /compare with expected data if error, report verify error
Verify Configuration sector	ISC-CONFIGURATION /compare with expected data if error, report verify error
Verify User-Code	USERCODE /compare with expected data if error, report verify error
Disable ISC mode	ISC-DISABLE
Disable Turbo Programming Mode	Disabling by either power-down or taking JTRSTn to low

9 Blank-check operation

Table 28. Blank check operation description

Description	Flow in sequence
Check Flash Memory JTAG Device ID	IDCODE if device is secured, exit with error if ID not matched, exit with error
Enable Turbo Programming Mode	TURBO-PROG-ENABLE
Enter ISC mode	ISC-ENABLE
Blank-Check selected sectors, except OTP	ISC-BLANK-CHECK (all selected sectors except OTP) if not blank, report each non-blank sector
Blank-Check OTP sector: (a) read OTP sector from DUT (b) compare FF's with the first 30 bytes. Do not compare the last 2 bytes (16 bits)	ISC-READ (OTP Sector) /compare FF's with the first 30 bytes of data if error, report OTP non-blank error
Disable ISC mode	ISC-DISABLE
Disable Turbo Programming Mode	Disabling by either power-down or taking JTRSTn to low

10 Upload operation

Table 29. Upload operation description

Description	Flow in sequence
Check Flash Memory JTAG Device ID	IDCODE if device is secured, exit with error if ID not matched, exit with error
Enable Turbo Programming Mode	TURBO-PROG-ENABLE
Enter ISC mode	ISC-ENABLE
Upload the following sector: a. Main Flash b. Secondary Flash c. OTP	/upload one sector at a time /repeat until all sectors are uploaded ISC-READ (Sector) if read error, exit with error else write to OBJ file
Upload Configuration sector	ISC-CONFIGURATION if error, exit with error else write to OBJ file
Upload User-Code	USERCODE if error, exit with error else write to OBJ file
Disable ISC mode	ISC-DISABLE
Disable Turbo Programming Mode	Disabling by either power-down or taking JTRSTn to low

11 Known limitations

1.	Error Status with full chip erase for a secured device. 9-23-05	Full chip erase on a secured device is an allowable instruction. For the current revision of silicon, if the device is secured, full chip erase will return an error status. The erase operation is still going on as usual. Read/Busy status is fully working. Hence, for a full chip erase on a secured device, work around is to ignore error status and instead, only relies on ready/busy status to determine the completion of full chip erase operation.
2.	ISC_READ. 9-23-05	During data shift (64 bits), TDI has to be 1 (instead of don't care).
3.	Address skip during ISC_READ	Pulsing of TCK during RTI state increments address, resulting in skipping of addresses during ISC read back. Workaround: Keep TCK unchanged during RTI.
4.	PAUSE State	If PAUSE state is entered, resuming shift from PAUSE requires starting of shift from the beginning. Workaround: Avoid entering of PAUSE state.
5.	Daisy chain setup	Require STR91xFA to be first in the chain.
6.	ISC-PROGRAM failure. 6-29-06	Pulsing of TCK during RTI state causes programming failure. Workaround: Keep TCK unchanged during RTI.

12 ISC logic description

12.1 Preliminary concepts

Memory organisation. The STR91xFA internal Flash consists of two banks: Main Flash memory (Bank 0) and Secondary Flash memory (Bank 1).

Five dedicated sectors containing: MFG code, ID code, User code, Configuration information, OTP, Security Bit. The dedicated sectors are physically placed in the Secondary Flash, but are independent from it.

Sector write protection. Each sector in the Main and Secondary Flash has its own NVM protection bit. All these protection bits reside in the Protection Sector. The content of this sector is loaded in the ISC CONFIGURATION register at power-up and can be shifted out with the ISC_CONFIGURATION instruction. If this bit is not set, the sector is unprotected, it can be run time protected/unprotected by an MCU instruction (level 1 protection). When the bit is set the sector is always protected (level 2 protection).

Using ISC instructions, you can program and erase the Protection Sector (see Configuration Program Flow and ISC_ERASE instruction flow); after a modification of the Protection Sector, the content of the ISC CONFIGURATION register is updated with the new value. With ISC instructions you can modify all the sectors in the matrix regardless of the sector protection status.

Readout protection. You can protect the Main and Secondary Flash content (codes) from readout through the JTAG port by setting the Security bit. Once set, the Flash content cannot be read, programmed or erased by JTAG. Only a full chip erase can erase the Security bit.

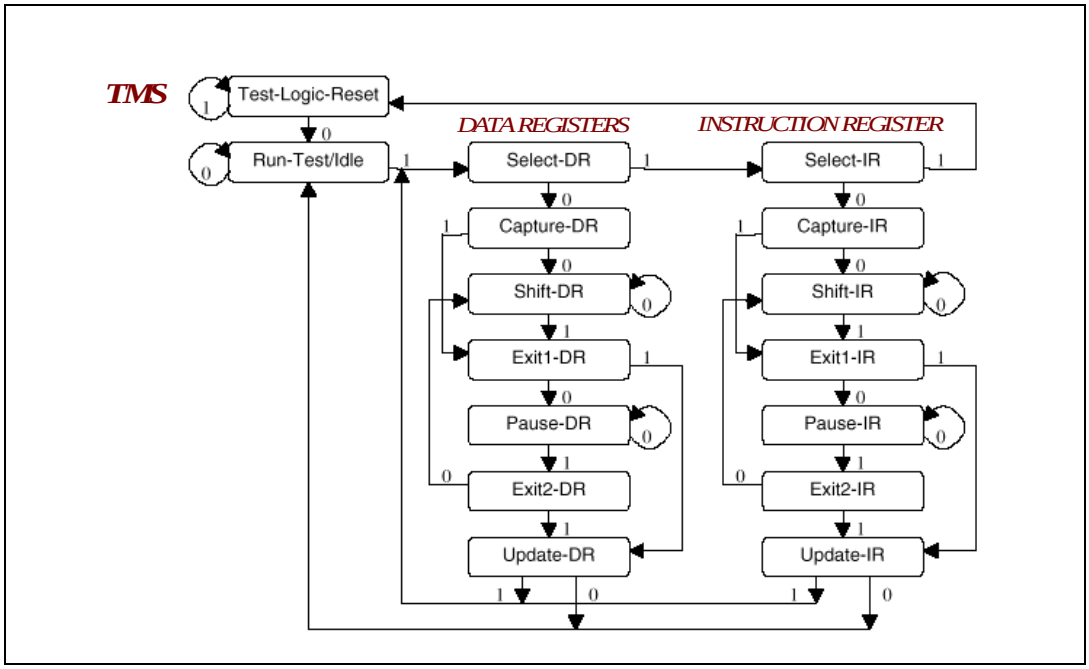
During the start-up procedure the device is secured.

If an ISC_PROGRAM_SECURITY instruction is executed, the security status will be not activated until the device exits from ISC mode, and, moreover, if the security bit is set and a full chip erase is executed, also the new unsecured status will be not activated until the device exits from ISC mode.

The non volatile security information is loaded in the SECURITY_BIT register at power-up and is stored both in the Instruction register and the Default register. The security status can be read by reading these registers.

OTP Sector. The OTP sector consists of 256 bits (8 words). It can be programmed through the JTAG - or through ARM CPU- it can not be erased. Other unprogrammed words can still be written at a later time. The OTP sector has a Lock bit which may be set by user through JTAG. After the bit is set, the OTP is protected from further writing. Reading and writing of the OTP Sector and programming the LOCK OTP BIT through JTAG are blocked if the device is secured. The Lock bit value is loaded in the LOCK_OTP register and the CONFIGURATION register at power-up. It can be shifted out with the configuration information (see the LOCK_OTP and CONFIGURATION register, the ISC_CONFIGURATION instruction flow and the Security ID and Lock OTP bit program flows).

Figure 7. JTAG state machine



Note: The state transition occurs on the rising edge of TCK. The value of TMS defines to which state it moves.

12.2 Instruction set

Any OPCODE not defined in *Table 30* will default to the BYPASS instruction.

Abbreviations:

BS = Boundary Scan, C = Capture, S = Shift, U = Update.

Table 30. Instruction set

Opcode (hex)	Instruction	Register	Reg. Size	Reg. Type	Origin			Mode		Secure		
					Mandatory	Optional	Unique	ISC	PSD	Functional	Limited	Blocked
00	EXTEST	Boundary Scan	TBD	BS	•				•	•		
01	SAMPLE/PRELOAD	Boundary Scan	TBD	BS	•				•	•		
02	INTEST	Boundary Scan	TBD	BS					•	•		
FC	HIGHZ	Bypass	1	S		•			•	•		
FA	CLAMP	Bypass	1	S		•			•	•		
FF	BYPASS	Bypass	1	S	•				•	•		
FE	IDCODE	Idcode	32	CS	•			•	•	•		
06	USERCODE	Usercode	32	CSU	•			•	•	•		
07	ISC-CONFIGURATION	Sector-Protect	64	CS			•	•	•	•		
4C	MFG_READ	MFG	32	CS			•	•	•	•		
0C	ISC_ENABLE	ISC-Enable	8	SU		•			•	•		
0F	ISC_DISABLE	ISC_Default	8	S		•		•		•		
10	ISC_NOOP	ISC_Default	8	S	•			•	•	•		
11	ISC_ADDRESS_SHIFT	ISC_Address	8	SU		•		•				•
13	ISC_CLR_STATUS	ISC-Default	8	S			•	•		•		

Table 30. Instruction set (continued)

Opcode (hex)	Instruction	Register	Reg. Size	Reg. Type	Origin			Mode		Secure		
					Mandatory	Optional	Unique	ISC	PSD	Functional	Limited	Blocked
20	ISC_PROGRAM	ISC-Data	64	CSU	•			•				•
22	ISC_PROGRAM_SECURITY	ISC_Default	8	S		•		•				•
23	ISC_PROGRAM_UC	Usercode	32	CSU		•		•				•
30	ISC_ERASE	ISC_Sector	64	CSU		•		•				
50	ISC_READ	ISC_Data	64	CSU		•		•				•
60	ISC_BLANK_CHECK	ISC_Sector	64	CSU			•	•				•
40	ISC_TEST_ENABLE	Test_Enable	16	S			•					•

12.3 Configuration bits

Except for the Security bit, you can read the rest of the configuration bits by shifting out the content of the Configuration register. As for the Security bit, you can read this through either the Default register or the Instruction register. Detailed information on how to read these bits is given in *Section 12.4: Register description*.

CSx Mapping 0: CS0 is mapped to Main Flash and CS1 is mapped to Secondary Flash. 1: CS0 is mapped to Secondary Flash and CS1 is mapped to Main Flash
LVD_th 0: The LVD threshold is set to 2.4 V. 1: The LVD threshold is set to 2.7 V
LVD_RESET_SELECT 0: LVD Reset Out is generated by VDD input only. 1: LVD Reset Out is generated by VDD or VDDQ input
LVD_RESET_WARNING 0: Early Warning is generated by VDD input only. 1: Early Warning is generated by VDD or VDDQ inputs
Security 0: The device is not secured. JTAG port can read Flash memory. 1: The device is secured and Flash memory content can not be read by JTAG. When this bit is set, the device sets DEBUG_EN output to low which disables the ARM MCU JTAG port

Sector Protection bits (one for each sector of the main and secondary Flash) 0: Sector is not protected from write or erase MCU operations.. 1: Sector is protected from write or erase MCU operations
OTP Lock bit 0: Write to unprogrammed OTP location is allowed. 1: Write to OTP location is inhibitedt

12.4 Register description

12.4.1 Instruction Register

7	6	5	4	3	2	1	0
N/A	SECURITY	INT_ERROR		READY/BUSY	MODE	0	1
	rw	rw	rw	rw	rw	rw	rw

Bit 7	N/A
Bit 6	**SECURITY**: *Security bit (Readout protection).* 0: Unsecured device. 1: Secured device.
Bits 5:4	**INT_ERROR:** *Internal Error/Program Error* 00: N/A 01: Fail 10: Success (default) 11: N/A.
Bit 3	**READY/BUSY**: 0:Busy* (see note) 1: Ready
Bit 2	**MODE**: 0: PSD mode 1: ISC mode
Bit 1	**0:** IEEE1149.1 mandatory
Bit 0	**1:** IEEE1149.1 mandatory

Note: With the shifting in of any instruction, the above status will be shifted out.

12.4.2 ISC default register

7	6	5	4	3	2	1	0
N/A	SECURITY	INT_ERROR		MODE	READY/BUSY	ISC_ERROR	
rw	rw	rw	rw	rw	rw	rw	rw

Bit 7	N/A
Bit 6	**SECURITY**: *Security bit (Readout protection).* 0: Unsecured device. 1: Secured device.
Bits 5:4	**INT_ERROR:** *Internal Erase/Program Error* 00: N/A 01: Fail 10: Success (default) 11: N/A. These bits will indicate an error if: 1. An attempt to read, program, or erase is made when the Flash is secured (SECURITY =1). 2. An attempt to program the OTP Sector is made while the OTP LOCK bit is set (OTP_LOCK_BIT = 1) 3. When the Flash reports an erase/program error. 4. When an ISC_DISABLE instruction is issued and the ISC is in busy mode. It is not mandatory to clear an INT_ERROR flagged by an internal ERASE or PROGRAM operation. The device will continue to function normally. However, the error flag will remain in the ISC_Default and the Instruction_Register bits <5:4>. If an INT_ERROR is issued, it can be cleared by a Test-Logic-Reset, a VCC power-down or an ISC_CLR_STATUS instruction. **Note:** If a Test-Logic-Reset is issued while ISC is in busy mode no error flag is set.
Bit 3	**MODE**: *Mode status bit* 0: PSD mode 1: ISC mode

Bit 2	**READY/BUSY**: *Ready/Busy status bit* 0: Busy* 1: Ready This bit goes low 1us after an internal ERASE, PROGRAM, READ, or BLANK CHECK operation is initiated. The polling loop which checks busy status, for PROGRAM or ERASE operations, must not begin before the 1us delay. This will avoid a false end of operation status. BUSY status should be polled to determine when any operation on the Flash is over.
Bits 1:0	**ISC_ERROR:** These bits comply with IEEE 1532 standard. The ISC Error bits indicate the ISC status of the previously executed instruction. They are updated any time a new instruction is loaded. The ISC Error bits indicate error only if an ISC instruction is loaded and the part is not in ISC mode. 00: N/A. 01: Error 10: Success (default) 11: N/A.

12.4.3 IDcode register

31	0
IDcode[31:0]	

rw

Bits 31:0	IDCode

12.4.4 Usercode register

31	0
Usercode[31:0]	

rw

Bits 31:0	Usercode 32 bits shift out only for USERCODE instruction 32 bits shift in of user code data for ISC_PROGRAM_UC instruction

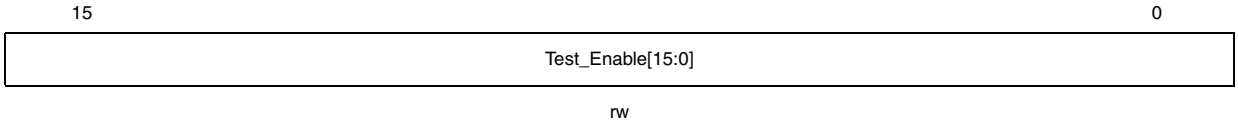
12.4.5 MFG register

31	0
MFG[31:0]	

rw

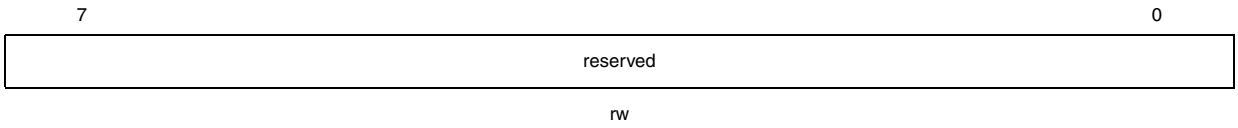
Bits 31:0	Manufacturing information. Factory use only.

12.4.6 Test enable register



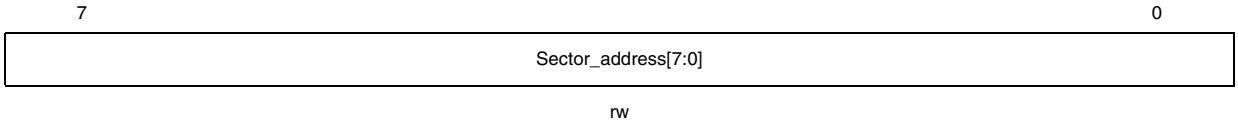
Bits 15:0	Reserved for factory use.

12.4.7 ISC enable register



Bits 7:0	Reserved

12.4.8 ISC address register



Bits 7:0	Sector Address [7:0] These bits contain the sector address listed in *Table 31*

Table 31. Sector addresses

Sector address	Sector
All others	N/A
76h	Lock bit of the Security ID
70h	OTP
60h	User Code
50h	Configuration (Sector protect , CSx mapping, LVD configuration bits)
40h	Reserved for future use
3Fh - 28h	N/A
27h - 20h	Bank 1 Flash Sectors 7:0
1Fh - 00h	Bank 0 Flash Sectors 1F: 0

Note: The Security bit is selected internally by ISC_PROGRAM_SECURITY and ISC_ERASE.

12.4.9 ISC sector register

Bits 63:52	Reserved
Bit 51	OTP Sector
Bit 50	User-Code
Bit 49	Configuration Sector (Sector protect , CSx mapping, LVD configuration bits)
Bit 48:40	Reserved
Bits 39:32	Flash Bank 1 sectors
Bits 31:0	Flash Bank 0 sectors

Shift in '1' to erase or blank check sector.

Shift out '0' if sector is blanked; '1' if sector checked is not blanked or not checked.

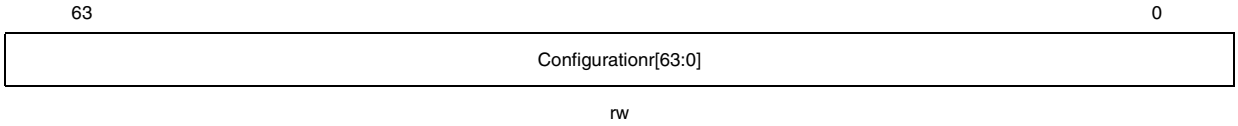
Error bits are set if at least one sector is not blanked.

If sector is not blank checked, the value of the corresponding bits is meaningless.

Note: Bit 51 can be used to address the OTP sector only for a blank check operation, the value of this bit is not considered during an Erase operation, in fact the OTP cannot be erased.

It is not possible to address the Security bit, OTP, and the CFI (MFG code and ID code) sectors.

12.4.10 Configuration register



Bit 63	Lock OTP
Bit 62:52	Reserved
Bit 51	LVD_WARNING_SELECT Defining the LVD warning signal input source.
Bit 50	LVD_RESET_SELECT Defining the LVD reset input source.
Bit 49	LVD_th Defining of the threshold of the LVD
Bits 48	CSx Mapping of CS0 and CS1
Bits 47:40	Reserved
Bits 39:32	Protection Flash Bank 1 sectors
Bits 31:0	Protection of Flash Bank 0 sectors

The Lock bit value stored in the bit 63 can only be shifted out, executing the ISC_CONFIGURATION instruction. It has no impact on the non volatile value of the Lock bit in the Flash sector.

12.5 Instruction description & flows

12.5.1 Bypass

The BYPASS instruction enables the 1-bit BYPASS_REGISTER between TDI and TDO, effectively causing the device to be "bypassed" in a JTAG chain. The Bypass Register is always loaded with a logic-0 on the rising edge of TCK in the Capture-DR state.

12.5.2 Sample/preload

The SAMPLE/PRELOAD instruction selects the BOUNDARY_REGISTER between TDI and TDO. It does not, however, interfere with the normal operation of the component pins or logic in anyway (BS_ON = 0). When this instruction is active the SAMPLE function of the instruction occurs on the rising edge of TCK that transitions the TAP from Capture-DR state to the Shift-DR state. At this point the boundary-scan cells at each I/O pin simply sample and store the state being driven into the pin for the case of inputs and the state being driven by the pin for outputs. In addition, the state of the tri-state enable signals for outputs are sampled. This information can then be shifted out within the Shift-DR state through TDO. As that data is shifted out, data can be shifted in through TDI for the PRELOAD function of the instruction, which basically takes effect on the falling edge of TCK in the Update-DR state, causing the shifted-in values to be loaded onto the latched parallel outputs of the boundary-scan cells. These values are not allowed to drive the pins, however. The purpose is to pre-load the values at component inputs and outputs so that a following EXTEST, or CLAMP instruction will cause these well-defined values to be driven onto the pins immediately upon selection of those instructions, rather than having random, unknown values at the pins.

12.5.3 Clamp

The CLAMP instruction enables the BYPASS_REGISTER between TDI and TDO. The purpose of this instruction is to allow the component pins to be "clamped" in a known state while the device is in "bypass" mode as part of a JTAG chain (BS_ON = 1). Normally, this instruction should be used in conjunction with SAMPLE/PRELOAD. The desired clamp values for the component pins should be loaded via the SAMPLE/PRELOAD instruction into the BOUNDARY_REGISTER. When the CLAMP instruction is updated in the Update-IR state, the pins will be driven with the values from the BOUNDARY_REGISTER. This allows the pins of a "bypassed" device to be controlled to a known static state during board testing.

12.5.4 HIGHZ

The HIGHZ instruction enables the BYPASS_REGISTER between TDI and TDO. The purpose of this instruction is to tri-state all component outputs. The outputs are tri-stated after this instruction is parallel loaded into the instruction register on the falling edge of TCK in the Update-IR state. The BYPASS_REGISTER operates exactly as it does when the BYPASS instruction is active.

12.5.5 EXTEST

The required EXTEST instruction places the IC into an external boundary-test mode and selects the boundary-scan register to be connected between TDI and TDO. The EXTEST instruction forces the output pins to drive the values from the BOUNDARY_REGISTER rather than from the normal internal logic (BS_ON = 1). The BOUNDARY_REGISTER cells

that drive the output pins are updated on the falling edge of the TCK in the Update-DR state. Input signals from external sources are sampled on the rising edge of TCK in the Capture-DR state. Typically, a SAMPLE/PRELOAD instruction is performed prior to EXTEST to pre-load the values, which will drive the component outputs.

12.5.6 INTEST

The optional INTEST instruction places the IC in an internal boundary-test mode and selects the boundary-scan register to be connected between TDI and TDO. During this instruction, the boundary-scan register is accessed to drive test data on-chip via the boundary inputs and receive test data on-chip via the boundary outputs.

12.5.7 IDCODE

Description: Shift out hardwired silicon signature

Requirement: None

Register: Device-ID

Shift-In: 32 bits don't care

Shift-Out: 32 bits ID-Code

Mode: ISC and PSD

Security: Fully functional

Table 32. IDCODE sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load IDCODE
2	DR-Scan	IDCODE shift out
3	Run-Test/Idle	End

12.5.8 MFG_READ

Description: Shift out manufacturing info. Wafer Sort and Final Test used only. NOT for customer use.

Requirement: None

Register: MFG

Shift-In: 32 bits don't care

Shift-Out: 32 bits manufacturing code

Mode: ISC and PSD

Security: Fully functional

Table 33. MFG_READ sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load MFG_READ
2	DR-Scan	Shift out manufacturing code
3	Run-Test/Idle	End

12.5.9 USERCODE

Description: Shift out customer programmable user code

Requirement: None

Register: Usercode

Shift-In: 32 bits don't care

Shift-Out: 32 bits User Code

Mode: ISC and PSD

Security: Fully functional

Table 34. USERCODE sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load USERCODE
2	DR-Scan	User Code shift out
3	Run-Test/Idle	End

12.5.10 ISC CONFIGURATION

Description: Shift out sector protect and bits and CSx and LVD configuration, Lock OTP bit.

Requirement: None

Register: Configuration

Shift-In: 64 bits don't care

Shift-Out: 64 bits Configuration

Mode: ISC and PSD

Security: Fully functional

Table 35. ISC CONFIGURATION sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC-CONFIGURATION
2	DR-Scan	Shift out sector protect, Csx, LVD configuration bits and Lock OTP information
3	Run-Test/Idle	End

Sector-Protect protects the Main and Secondary Flash from being programmed/erased by the controller. Once ISC is active, Sector-Protect has no impact on ISC. ISC can write to any sector regardless of protection status. Note that this is different from security. Configuration info is loaded upon power up to be read out during ISC_CONFIGURATION instruction.

12.5.11 ISC_ENABLE

Description: Enter ISC mode

Requirement: Part is not in ISC mode, all modules are not busy

Register: ISC_Enable

Shift-In: 8 bits don't care

Shift-Out: 8 bits don't care

Mode: PSD

Security: Fully functional

Table 36. ISC_ENABLE sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_ENABLE
2	DR-Scan	Shift in 8 bits don't care
3	Run-Test/Idle	End

Shifting in 8 bits of "don't care" to ISC_Enable register during step 2 (DR-Scan) is not needed. This ISC_Enable register is implemented for future use, to allow various ISC configurations.

12.5.12 ISC_DISABLE

Description: Exit ISC mode, going back to PSD mode

Requirement: Part is in ISC mode, all modules are not busy

Register: ISC_Default

Shift-In: 8 bits don't care

Shift-Out: 8 bits status

Mode: ISC

Security: Fully functional

Table 37. ISC_DISABLE sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_DISABLE
2	DR-Scan	Shift out status
3	Run-Test/Idle	Wait 50uS
4	Test-Logic-Reset	TLR or load ISC_NOOP (or BYPASS) followed by RTI

Upon RTI (step 3), security status is updated.

The 50 us in step 3 is a delay enough delay to correctly handle possible aborts.

Exit ISC mode while Busy status is till active aborts the current operation. Internal program/erase error is genterated in this case.

ISC_DISABLE clears ISC error but not internal program/erase error.

12.5.13 ISC_NOOP

Description: Shift out status bits

Requirement: None

Register: ISC_Default

Shift-In: 8 bits don't care

Shift-Out: 8 bits status

Mode: ISC and PSD

Security: Fully functional

Table 38. ISC_NOOP sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_NOOP
2	DR-Scan	Shift out status
3	Run-Test/Idle	End

Loaded while a module is busy does not terminate the previous instruction.

Use for polling purpose, checking of the internal operation status (ie. Ready/busy).

12.5.14 ISC_ADDRESS_SHIFT

Description: Load sector address

Requirement: Part is in ISC mode

Register: ISC_Address

Shift-In: 8 bits sector address

Shift-Out: 8 bits don't care

Mode: ISC

Security: Blocked

Table 39. ISC_ADDRESS_SHIFT sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_ADDRESS_SHIFT
2	DR-Scan	Sector address shift in
3	Run-Test/Idle	End

If the device is not in ISC mode, ISC_ADDRESS_SHIFT will not update the Address register. Furthermore, ISC_ERROR flag will be set.

12.5.15 ISC_CLR_STATUS

Description: Clear/reset ISC and other modules (ie. Flash) if error occurred

Requirement: Part is in ISC mode, error status exists.

Register: ISC_Default

Shift-In: 8 bits don't care

Shift-Out: 8 bits status

Mode: ISC

Security: Fully functional

Table 40. ISC_CLR_STATUS sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_CLR_STATUS
2	DR-Scan	Shift out status (should still show error)
3	Run-Test/Idle	Clear status (including module's error), wait 50uS
4	DR-Scan	Shift out status - expect no error
5	Run-Test/Idle	End

DR-Scan in steps 2 & 4 is optional.

If the part isn't in ISC mode, the instruction will not be executed and an ISC_ERROR is issued.

12.5.16 ISC PROGRAM

Description: Program a sector

Requirement: Part is in ISC mode, unsecured part

Register: ISC_Data

Shift-In: 64 bits data

Shift-Out: 64 bits don't care

Mode: ISC

Security: Blocked

Table 41. Main and Secondary Flash sectors program flow

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_ADDRESS_SHIFT
2	DR-Scan	Shift in 8 bits sector address
3	IR-Scan	Load ISC_PROGRAM

Table 41. Main and Secondary Flash sectors program flow (continued)

Step	Tap Activity	Comment
4	DR-Scan	Shift in 64 bits of data to be programmed
5	Run-Test/Idle	Program starts/address incremented
Wait for 50 us. Begin polling		
6	IR-Scan	Load ISC-NOOP
7	DR-Scan	Shift out status
Repeat step 7 until ready = 1		
Go back to step 3, repeat steps 3-7 until end of sector is reached.		
8	Run-Test/Idle	End

Once the end of sector is reached, the program flow cannot jump to the next sector.

After the end of the sector is reached, the execution of next ISC_PROGRAM instruction is not allowed without shifting in a new address. This avoids that, continuing to repeat the steps from 3 to 7 once the end of sector is reached, the program flow starts again from the beginning of the sector. This control isn't active if the program flow is interrupted before reaching the end of sector.

Table 42. OTP program flow (Lock bit: blocked)

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_ADDRESS_SHIFT
2	DR-Scan	Shift in the 8 bit OTP address - 70 Hex
3	IR-Scan	Load ISC_PROGRAM
4	DR-Scan	Shift in 64 bits of data to be programmed
5	Run-Test/Idle	Program starts/address incremented
Wait for 50 us. Begin polling		
6	IR-Scan	Load ISC-NOOP
7	DR-Scan	Shift out status
Repeat step 7 until ready = 1		
Go back to step 3, repeat steps 3-7 three more times (256 bits)		
8	Run-Test/Idle	End

OTP may be programmed many times until the OTP Lock bit is set, it cannot ever be erased. If the Lock bit is set, the program instruction is not executed and no error flag is set.

Once the end of sector is reached, the program flow cannot jump to the next sector.

After the end of the sector is reached, the execution of the next ISC_PROGRAM instruction is not allowed without shifting in a new address. This avoids the program flow starting again at the beginning of the sector (by continuing to repeat steps 3 to 7) when the end of the sector is reached. This control isn't active if the program flow is interrupted before reaching the end of sector.

Note: 1 This is the only flow that can be used to program the OTP Lock bit

2 DR-Scan in step 4 is optional.

3 The Lock bit may be programmed only once and can never be erased.

Table 43. OTP Lock Bit program flow

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_ADDRESS_SHIFT
2	DR-Scan	Shift in the 8 bit OTP Lock bit address - 76 Hex
3	IR-Scan	Load ISC_PROGRAM
4	DR-Scan	Shift in 64 bits don't care
5	Run-Test/Idle	Program starts
Wait for 50 us. Begin polling		
6	IR-Scan	Load ISC-NOOP
7	DR-Scan	Shift out status
Repeat step 7 until ready = 1		
8	Run-Test/Idle	End

Table 44. Configuration (sector protect, CSx mapping and LVD configuration bits) program flow

Step	Tap Activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_ADDRESS_SHIFT
2	DR-Scan	Shift in the 8 bit Configuration address - 50 Hex
3	IR-Scan	Load ISC_PROGRAM
4	DR-Scan	Shift in 64 bits to be programmed.
5	Run-Test/Idle	Program starts
Wait for 50 us. Begin polling		
6	IR-Scan	Load ISC-NOOP
7	DR-Scan	Shift out status
Repeat step 7 until ready = 1		
8	Run-Test/Idle	End

During DR-Scan of step 4, the 64th bit is reserved and has to be “1” to ensure correct operation of the device.

12.5.17 ISC_PROGRAM_UC

Description: Program User-Code

Requirement: Part is in ISC mode, unsecured part

Register: Usercode

Shift-In: 32 bits User-Code data

Shift-Out: 32 bits don't care

Mode: ISC

Security: Blocked

Table 45. ISC_PROGRAM_UC sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_PROGRAM_UC
2	DR-Scan	Shift in User-Code (32 bits)
3	Run-Test/Idle	User-Code program starts
Begin polling		
4	IR-Scan	Load ISC-NOOP
5	DR-Scan	Shift out status
Repeat step 5 until ready = 1		
6	Run-Test/Idle	End

Programming of User-Code does not require address.

After step 6, if USERCODE instruction is loaded, the correct latest User-Code should be shifted out.

12.5.18 ISC_PROGRAM_SECURITY

Description: Program security bit

Requirement: Part is in ISC mode, unsecured part

Register: ISC_Default

Shift-In: 8 bits don't care

Shift-Out: 8 bits status

Mode: ISC

Security: Blocked

Table 46. ISC_PROGRAM_SECURITY sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_ADDRESS_SHIFT
2	DR-Scan	Shift in 8 bits security dummy address, 80 Hex
3	IR-Scan	Load ISC_PROGRAM_SECURITY
4	Run-Test/Idle	Security bit program begins
5	DR-Scan	Shift out status
Begin polling		
6	IR-Scan	Load ISC-NOOP
7	DR-Scan	Shift out status
Repeat step 7until ready = 1		
8	Run-Test/Idle	End

Security status is not activated until the device exits from ISC mode (ISC_DISABLE instruction is executed or a Test-Logic-Reset is issued). For example, after ISC_PROGRAM_SECURITY is executed, array verification is still allowed until exiting of ISC mode, but, however, the only way to erase the Security bit is a full chip erase, even before exiting ISC mode.

If the Security bit is set and a full chip erase is executed, the device will be unblocked only after it exits from ISC mode.

When the Security bit is programmed the signal ISC_DEBUG_EN becomes low at once, without expecting to exit from ISC mode, in the same way, when a full chip erase is executed, the ISC_DEBUG_EN becomes high at once.

12.5.19 ISC_READ

Description: Read a sector (auto-incremented address)

Requirement: Part is in ISC mode, unsecured

Register: ISC_Data

Shift-In: 64 bits don't care

Shift-Out: 64 bits data

Mode: ISC

Security: Blocked

Table 47. ISC_READ sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_ADDRESS_SHIFT
2	DR-Scan	Shift in 8 bits sector address
3	IR-Scan	Load ISC_READ
4	Run-Test/Idle	Wait 5uS; data read; address incremented
5	DR-Scan	Shift out 64 bits of data
Repeat steps 4 & 5 until end of sector		
6	IR-Scan	Load ISC_NOOP
7	Run-Test/Idle	End

In step 4, 5uS is needed to allow time for capturing 64 bits.

In order to read a different sector, steps 0-7 need to be repeated. Address auto-increment does not go to the next sector. When the end of sector is reached, continuing to repeat the steps 4 & 5, the reading flow starts again from the beginning of the sector.

Note: The ISC_READ instruction can be used for reading the main and secondary Flash sectors and OTP. It cannot be used for reading the Configuration bits, the User code and the OTP Lock bit.

12.5.20 ISC_ERASE

Description: Erase single/multiple/all sectors

Requirement: Part is in ISC mode

Register: ISC_Sector

Shift-In: 64 bits for sector to be erased

Shift-Out: 64 bits don't care

Mode: ISC

Security allowed: Partially functional. If secured, only full chip erase (shift in of 64 '1's) is

Table 48. ISC_ERASE sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_ERASE
2	DR-Scan	Shift in sector(s) to be erased. '1' erase; '0' don't erase
3	Run-Test/Idle	Erase starts
Begin polling		
4	IR-Scan	Load ISC-NOOP
5	DR-Scan	Shift out status
Repeat step 5 until ready = 1		
6	Run-Test/Idle	End

When secured, only full chip erase is allowed; i.e. all sectors must be selected for erase. This is done by shifting in all 64 '1's in step 2 above.

Security status is updated at the end of full chip erase.

If an erase operation is executed on the Configuration Sector, it erases only the information related to sector protection, CSx mapping, and LVDconfiguration. The Lock OTP information is not affected. Remember that the Lock OTP can never be erased.

12.5.21 ISC_BLANK_CHECK

Description: Blank check single/multiple/all sectors

Requirement: Part is in ISC mode, unsecured

Register: ISC_Sector

Shift-In: 64 bits for sector to be blank checked

Shift-Out: 64 bits for blank sector data

Mode: ISC

Security: Blocked

Table 49. ISC_BLANK_CHECK sequence

Step	Tap activity	Comment
0	Run-Test/Idle	Start
1	IR-Scan	Load ISC_BLANK_CHECK
2	DR-Scan	Shift in sector(s) to be blank checked. '1' check; '0' don't check
3	Run-Test/Idle	Blank check starts, wait 40 ms
4	DR-Scan	'0' = Sector selected and blank '1' = Sector selected and not blank. If sector is not selected for checking, value is N/A
5	IR-Scan	Load ISC_NOOP (to replace ISC_BLANK_CHECK)
6	Run-Test/Idle	End

Configuration Sector "blanked" means all the bit are not programmed and are read '0' (default value).

OTP sector can be addressed in the SECTOR_REGISTER only for a blank check operation. It cannot be erased. OTP "blanked" means it is not programmed and unlocked.

12.5.22 ISC_TEST_ENABLE

Description:This instruction is for factory testing of the device only.

12.5.23 TRST, LVD_RESET_ON and TEST-LOGIC-RESET

TRST pin, active low, resets the JTAG Tap Controller and synchronizes the JTAG state machines to the same reset state on both the ARM MCU and the Flash Memory.

LVD_RESET_OUT pin, active low, detects a Low Voltage condition and resets the JTAG Tap controller.

When TRST or LVD_RESET_OUT are active, the JTAG state machine jumps asynchronously to the Test-Logic-Reset state, which activates an internal reset.

When a test logic reset is asserted while the Flash memory is in modify mode, the modify operation will be aborted and the memory content is not guaranteed. The aborted operation

needs time to complete and may take up to 50uS before JTAG is ready to accept the next command.

12.5.24 Abort handling

Table 50. Abort handling summary

Abort condition	Aborted Instruction	Abort handling
Vcc Power Down	All	None((1))
– Test-Logic-Reset (TRST, LVD_RESET) – ISC_DISABLE	Isc_Program, Isc_Program_Uc, Isc_Program_Security Isc_Erase Isc_Blank_Check	1) Abort condition is latched in Micro-osc domain. 2) An abort request is sent to the Micro. The Micro operates to switch off the pumps, to discharge the capacitances, and to clear its own status. When Micro finishes, it sends an acknowledgment to ISC 3) ISC resets the ISC_Algorithm and then the Flash_Interface and all the other blocks. Almost 50 us are required to complete the abort sequence. During this time no instruction should be issued.
	Isc_Read	1) Abort condition is latched in Micro-osc domain. 2) ISC resets the ISC_Algorithm and then the Flash_Interface and all the other blocks. All the abort sequence will finish within 1 us. In this time no instruction should be issued.

1. The Security and Lock bits have a protection in the double program cycle.

12.5.25 Register table or reset

Table 51. Register table or reset

Register	Reset
INSTRUCTION	RST_IR_N
STATUS	no reset signal
BYPASS	no reset signal
ENABLE	RST_DR_N
SECTOR	RST_DR_N
ADDRESS	RST_DR_N
DATA	RST_N
ID CODE	RST_N
MFG CODE	RST_N
USER CODE	RST_N
CONFIGURATION	RST_N
SECURITY BIT	RST_N
LOCK OTP BIT	RST_N
TEST ENABLE	RST_IR_N

RST_N: It becomes active low if a Vcc power down is issued.

RST_IR_N: It becomes active (low) if RST_N becomes low or if a Test Logic Reset is issued.

RST_DR_N: It becomes active (low) if RST_N becomes low or if a Test Logic Reset is issued or if a ISC_DISABLE instruction is issued.

13 Revision history

Table 52. Document revision history

Date	Revision	Changes
10-May-2006	1	Initial release.
11-Aug-2006	2	Added a wait time of 50 µs before polling in *Section 5.9: ISC-PROGRAM on page 48* Changed program OTP sector in *Section 7: Erase and program operation on page 53* Changed verify OTP sector in *Section 8: Verify operation on page 55* Changed blank check OTP sector *Section 9: Blank-check operation on page 56* Changed wait time to 50 µs *Section 12.5.16: ISC PROGRAM on page 76*
24-May-2007	3	Added JTAG timing specification in *Section 5.1*. Added 9 new STR9 partnumbers in *Table 8 on page 31* and *Table 15 on page 37*. Moved security-bit programming toward the end of programming flow in *Section 7*. Added remark on register default value to *Section 1.13.4: Protection level 1 register (STR91xFAxx2 and STR91xFAxx4) on page 30* and*Section 1.13.6: Flash configuration register on page 32* Changed description of bit 4 and 3 in *Section 1.15.5: FMI control register (FMI_CR) on page 37*
12-Dec-2007	4	Updated product references to STR91xFA throughout document. Added *Table 3 on page 11* and *Table 4 on page 13* Updated *Section 1.5: Electronic signature on page 16* Updated *Section 1.6: OTP sector on page 16* Updated *Section 1.11.2: RSIG read electronic signature (90h) on page 20* Updated *Section 1.12: Write operations on page 22* Updated *Section 1.13.2* and *Section 1.13.3* Updated *Booting from Bank 1 on page 35* Updated *Section 1.15.1* to *Section 1.15.4*
16-April-2008	5	Added *Section 1.16: STR91x in-application programming (IAP) on page 40*

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2008 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com