DALPU-4

NEOWINE Crypto and Authentication

DATASHEET

Features

32-bit ARM Cortex-M3 Core

- Nested Vectored Interrupt Controller(NVIC): 1 Wake up and 1 peripheral interrupt
- 16-bit or 32-bit System timer (Sys Tick): System timer for OS task management
- Creation and Management of Cipher Key

On-chip Memory

- EEPROM
  - 128 KB
  - Configuration/Key/User data storage
  - 15 User Zones of 2 Kbits Each
  - Retention 10 years
  - Erase/Write Endurance: 100K
- SRAM
  - On chip 32 Kbytes

Serial Interface

- UART
  - Full duplex double buffer
  - Parity can be enabled or disabled
  - Built-in dedicated baud rate generator
  - Various error detection functions (parity error, framing errors, and overrun errors)
  - External x-tal for UART
- SPI0, SPI2
  - Slave, Mode 0
  - Up to 40 MHz SCK
  - Symmetric cipher core control
- SPI1
  - Master/Slave
  - Master: Up to 10 MHz SCK
  - Slave: Up to 1.5 MHz SCK
  - Mode 0, 1, 2, 3
- GPIO
  - 4 GPIOs

Clock, Reset and Voltage

- Clock
  - Built in OSC.
  - Main Clock: 50 fMHz
- Reset
  - Built in power on reset
  - Software reset
- 1.5V, 3.3V Supply Voltage

Debug

- Serial Wire Debug Port(SW-DP)

Low Power Consumption Mode

- The GPIO is sufficient to power up and down
- PMU clock gating of Cortex-M3

Asymmetric cipher function

- ECC-P256, RSA-4096
- ECDSA, ECDH

Symmetric cipher function

- AES-128/256, ARIA-128/256
- Modes of Operation: Confidentiality (ECB, CBC, CFB, OFB, CTR)
- Creation and Management of Cipher Key

Crypto Device function

- User ID, User Serial (Manufacture ID), MIDR, RVC
- **PUF**
- Generate random using two different phase counters

Application

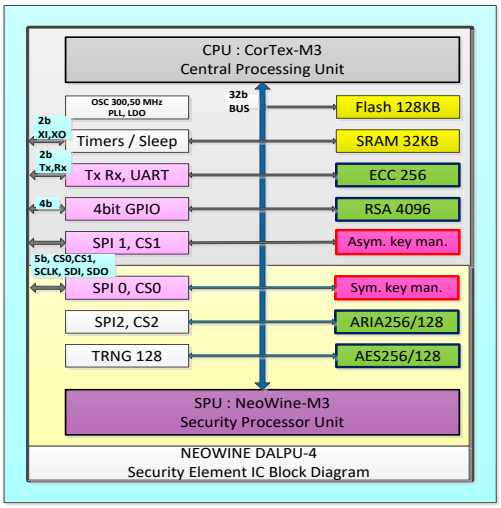
- Print cartridge, GPS, Navigation
- Mobile Device, IPC, CCTV, DVD
- Set-Top Boxes (STBs), Etc.

Standards

- ECC, RSA FIPS 186-3, 186-4
- AES-128/256 FIPS 140-2
- TRNG NIST SP 800-90B compatibility

Benefits

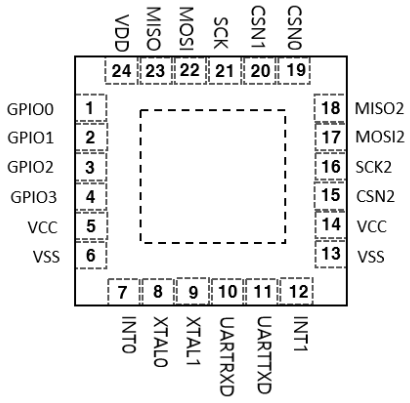
- Read/Write, Encrypted, or Read-only User Zone Options
- Ease use of Crypto Device to replace of existing EEPROM devices
- Authenticate Consumer products, Components, and Network equipment
- Protect Sensitive Firmware
- Securely Store Sensitive Data
- Manage Warranty Claims
- Securely Store Identity Data



Block Diagram

Package

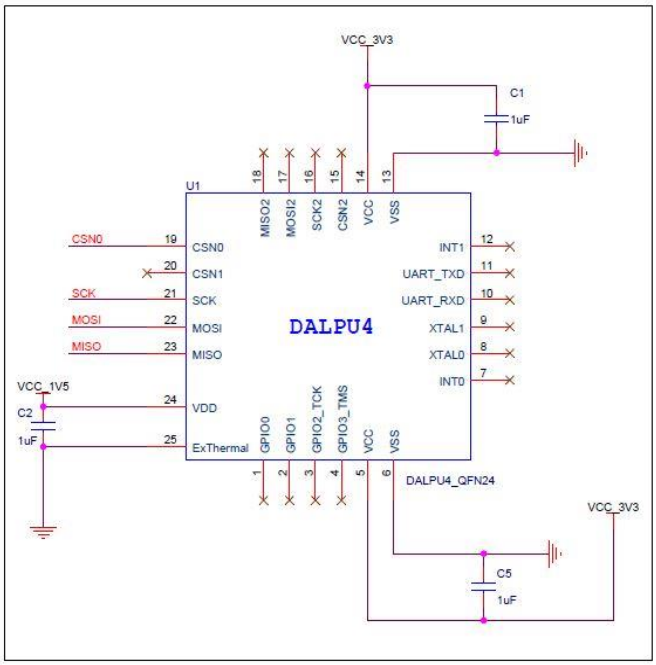
- QFN 4x4-25L (4mm X 4mm X 0.75mm)



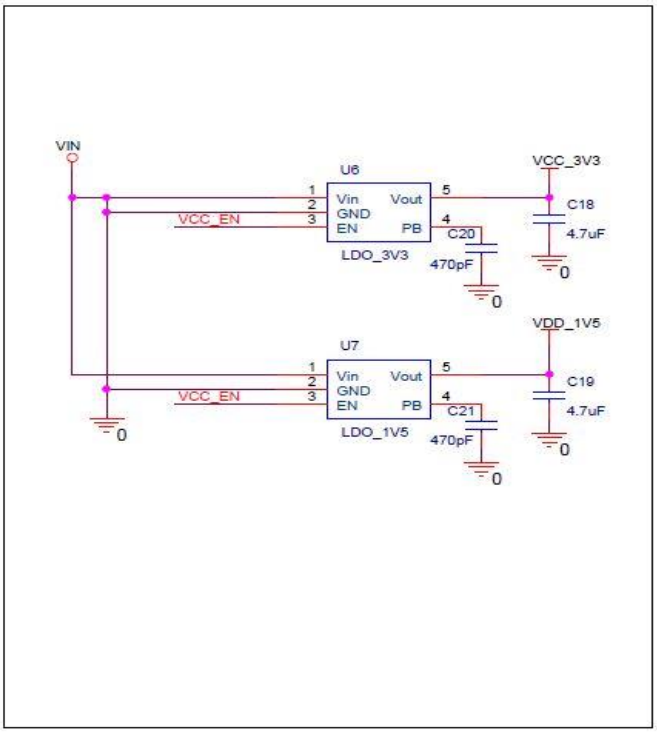
Top view

Schematic Diagram

- Application Circuit



- DALPU-4 Power Circuit



Contents

Contents 3
Figure 4
1 Introduction 5
1.1 Applications 5
1.2 Device Features 5
1.3 Crypto Operation 5
2 Device configurations 6
2.1 Symmetric Cipher Parts 6
2.2 Asymmetric Cipher Parts 8
3 Device Functions 10
4 E-MCU to DALPU-4 Interface 11
4.1 SPI0 Interface 11
4.2 SPI1 Interface 12
5 Address Map 13
5.1 CORTEX-M3 AMBA Bus Address Map 13
5.2 Symmetric Cipher parts Address Map 14
6 Registers 16
6.1 SPI0 registers 16
6.2 CORTEX-M3 registers 41
7 EEPROM Configuration 46
8 Revision History 47

Figure

Figure 2-1 Main Control state machine diagram 7
Figure 4-1 SPI0 Normal Mode Write in Address Mode 11
Figure 4-2 SPI0 Normal Mode Read in Address Mode 11
Figure 4-3 Motorola SPI frame format with SPO=0 and SPH=1 12

1 Introduction

The following sections introduce the features and functions of the DALPU-4 crypto device.

1.1 Applications

DALPU-4 is designed to apply high security rules to the product. These security rules can be used to protect the data, to protect the functionality of the product, and to prevent replication.

- Product authentication

  DALPU-4 has the function of preventing reproduction or illegal modification of products.

- Exchanging Security Keys

  DALPU-4 has Public-Key Cryptosystems. User can use this function to exchange keys safely.

- Storing Security Data

  You can store secret keys used for ciphering. Can save configuration, calibration or other secret data.

1.2 Device Features

The DALPU-4 has an Electrically Erasable Programmable Read-Only Memory (EEPROM). The EEPROM can be used for key storage, miscellaneous write/read data, read-only, secret data, consumption logging, and security configuration. DALPU-4 has 32-bit ARM Cortex-M3 Core. This core is in charge of public key operation. DALPU-4 has 32 Kbytes SRAM, it is used for M3 code execution region and user code region. DALPU-4 has SPI0, SPI1, UART and GPIO interfaces. SPI0 can have a slave mode. With SPI0 user can control symmetric cipher core. SPI1 can have both a slave and a master mode. With SPI1 in a slave mode, user can control asymmetric cipher core. DALPU-4 has a power saving mode. In sleep mode internal oscillator is disabled. DALPU-4 has a symmetric cipher function which is ECC-P256, ECDSA and ECDH. DALPU-4 has a symmetric cipher function which is AES-128/256. AES supports ECB, CBC, CFB, OFB, CTR operating modes..

1.3 Crypto Operation

DALPU-4 save control information to the EEPROM. These control information is a configuration data. The configuration data is protected by password. DALPU-4 can encrypt or decrypt an input data with AES. And the result is read by an external MCU. DALPU-4 encrypt user data and save to a EEPROM. External MCU(E-MCU) can read saved data. When E-MCU request the saved data, the DALPU-4 returns encrypted data. DALPU-4 has authentication function using SHA.

2 Device configurations

DALPU-4 is composed of two parts. One is asymmetric cipher part and the other is symmetric cipher part. The asymmetric cipher part is composed of a CORTEX-M3 and an asymmetric cipher hardware. The symmetric cipher part is a Security Processor Unit(SPU). The symmetric cipher part is composed of a symmetric cipher hardware and a main control hardware. The asymmetric cipher part take charge of ECC-P256, RSA-2048, ECDSA and ECDH. The symmetric part take charge of AES-128/256, SHA-256 and main control function. The main control function consists of state machine hardware. The following sections explain operation of each functions. An external MCU controls DALPU-4. The DALPU-4 has two interfaces to the external MCU. One is SPI0 for the symmetric cipher part. The other is SPI1 for the asymmetric cipher part. The external MCU can control DALPU-4 main control hardware through SPI0 interface. The external MCU can control a CORTEX-M3 and asymmetric cipher hardware through SPI1.

2.1 Symmetric Cipher Parts

The Symmetric Cipher part consist of a Symmetric Cipher Hardware and a Main Control Hardware. First the Main Control Hardware parts are as follows.

2.1.1 MAIN CONTROL HARDWARE

The Main Control Hardware can have 14 main states. Each main state has independent operation. Most of operations are processed in one main state, but some operations are processed in several main states. When the DALPU-4 wakes up, it processes the initial procedures automatically, then goes to ST0_STANDBY state. Usually, if DALPU-4 finishes a certain function, it always goes to ST0_STANDBY state. A hardware logic sends main state to ST0_STANDBY state when the DALPU-4 finishes a function, or E-MCU must control to send main state to ST0_STANDBY state in some functions. If DALPU-4 finishes state abnormally, it may can't process another function normally.

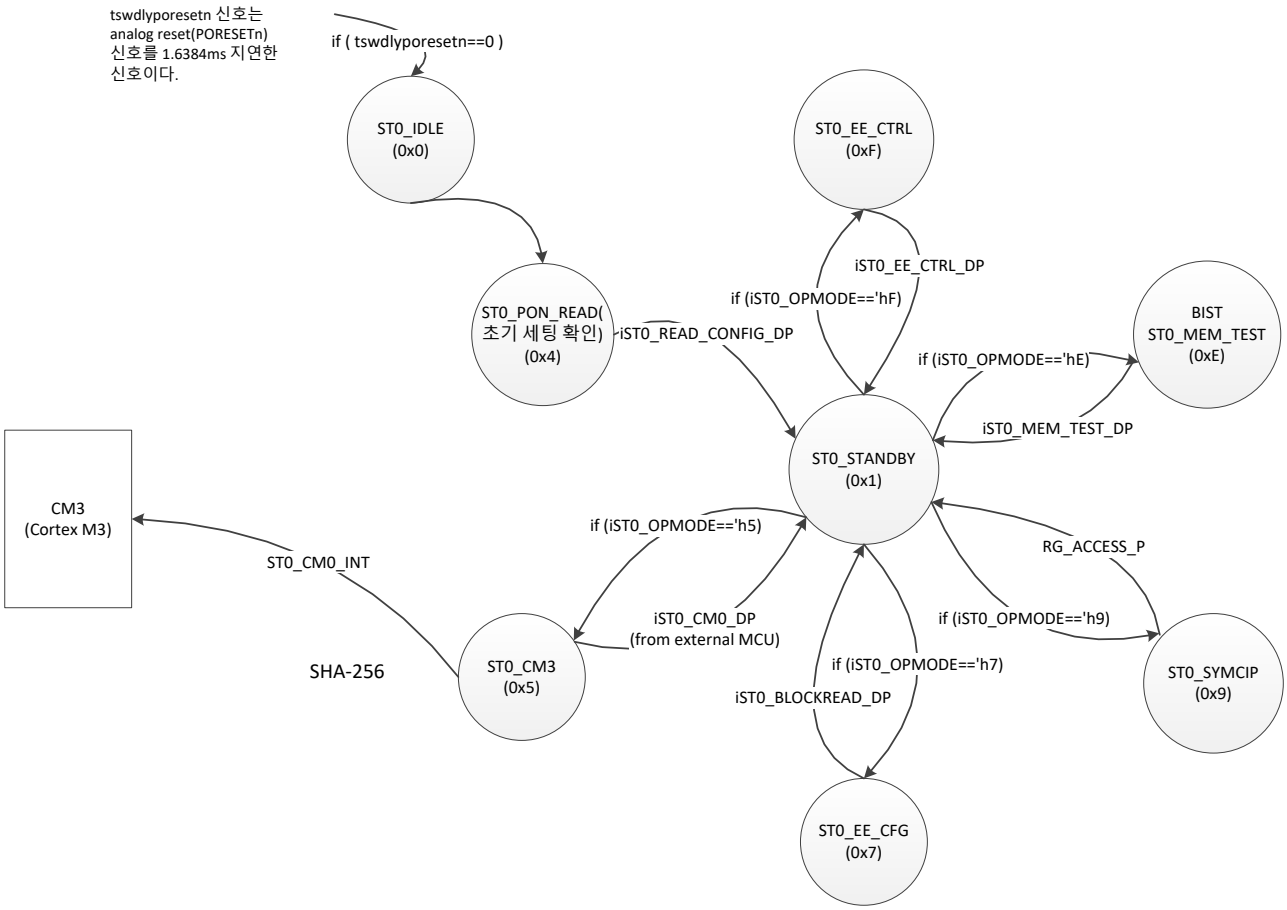


Figure 2-1 Main Control state machine diagram

Figure 2-1 shows every state which the main state can have. When power is on, the DALPU-4 begins an initial procedure. The initial procedure starts from ST0_IDLE state, and stops to ST0_STANDBY state. The initial procedure starts automatically when the power is up. User can skip ST0_CHK_RSFLAG state. See the following sections to control skip function of ST0_CHK_RSFLAG state.

2.1.1.1 ST0_PON_READ STATE CONTROL

The hardware prepares initial values to process a normal operation. The DALPU-4 main control hardware executes this state automatically

2.1.1.2 ST0_CM3 STATE CONTROL

The hardware cannot set this state, the E-MCU can set this state through SPI0. The E-MCU controls RG_ST0_OPMODE(0x1_0604) register to set ST0_CM0 state. If the E-MCU want to communicate with CORTEX-M0, E-MCU sets ST0_CM0 state first. After then, E-MCU sends control information to CORTEX-M0. If CORTEX-M0 receive control information, it controls asymmetric cipher hardware. Some of CORTEX-M0 control may affect the symcipher hardware blocks. A detail explanation is given later of this document.

2.1.1.3 ST0_SYMCIP STATE CONTROL

The hardware cannot set this state, the E-MCU can set this state through SPI0. The E-MCU controls RG_ST0_OPMODE(0x1_0604) register to set ST0_SYMCIP state. DALPU-4 operates symmetric cipher functions in this state.

Second the Symmetric Cipher Hardware configurations are as follows.

2.1.2 SYMMETRIC CIPHER HARDWARE

Symmetric cipher hardware handles the encryption and decryption using AES and the authentication using SHA. It also generates random. It manages writing and reading of EEPROM. It manages a key generation, key storage and key change.

2.2 Asymmetric Cipher Parts

Asymmetric cipher parts consist of the CORTEX-M3 and the Asymmetric Cipher Hardware. The CORTEX-M3 controls the asymmetric Cipher Hardware. The Asymmetric Cipher Hardware is responsible for performing ECC, ECDH and ECDSA algorithms.

2.2.1 CORTEX-M3 HARDWARE

2.2.1.1 CORTEX-M3 CORE

The Cortex-M3 processor is an entry-level 32-bit ARM Cortex processor designed for a broad range of embedded applications. It offers significant benefits to developers, including:
- simple, easy-to-use programmers model
- highly efficient ultra-low power operation
- excellent code density
- deterministic, high-performance interrupt handling
- upward compatibility with the rest of the Cortex-M processor family.

2.2.1.2 CORTEX-M3 CORE PERIPHERALS

The Cortex-M3 core peripherals are:

NVIC

An embedded interrupt controller that supports low latency interrupt processing.

System Control Block

The System Control Block (SCB) is the programmers model interface to the processor. It provides system implementation information and system control, including configuration, control, and reporting of system exceptions.

Optional system timer

The optional system timer, SysTick, is a 24-bit count-down timer. If implemented, use this as a Real Time Operating System (RTOS) tick timer or as a simple counter.

Public Key Engine (Asymmetric Cipher)

The PK Crypto Engine is a very flexible solution based on a scalable array of dual-field processing elements that can be used to execute all operations & algorithms required for PK Crypto-systems:

- Elliptic Curve Cryptography (ECC)
- Diffie-Hellman (D-H & ECD-H) Key Exchange
- Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)
- Primality Test (Rabin-Miller) & Key Generation
- Any other crypto algorithm can be supported on request

Embedded EEPROM and SRAM

DALPU-4 has 128 KB of EEPROM and SRAM. The EEPROM stores the cortex-M3 code and the rom code of the asymmetric cipher. The SRAM is shared by the Cortex-M3 and Asymmetric Cipher, Symmetric Cipher for TLS.

Registercm3

Registercm3 is basically used by Cortex M3 to control the EEPROM. Generates the control signals needed to write to or read from the EEPROM. It is also used when selecting IO or SPI, and also used when Cortex M3 generates a random value.

Timers

The Dual Input Timers module, Timers is an AMBA slave module and connects to the APB. The Dual-Timer module consists of two programmable 32/16-bit down counters that can generate interrupts on reaching zero. A Timer module can be programmed for a 32-bit or 16-bit counter size and one of three timer modes using the Control Register. The operation of each Timer module is identical. It has one of three timer modes:

- free-running
- periodic
- one-shot

UART

The UART is an AMBA slave module that connects to the Advanced Peripheral Bus (APB).
The UART provides:

- Compliance to the AMBA Specification (Rev 2.0) onwards for easy integration into SoC implementation.
- Separate 16x8 transmit and 16x12 receive First-In, First-Out memory buffers(FIFOs) to reduce CPU interrupts.
- Programmable FIFO disabling for 1-byte depth.
- Programmable baud rate generator. This enables division of the reference clock by (1x16) to (65535 x16) and generates an internal x16 clock. The divisor can be a fractional number enabling you to use any clock with a frequency >3.6864MHz as the reference clock.
- Standard asynchronous communication bits (start, stop and parity). These are added prior to transmission and removed on reception.

GPIOs (General-Purpose Input/Output)

The GPIO is a general purpose I/O device. It has the following properties:

- three registers : Data, Direction, Interrupt Registers
- 32 input or output lines with programmable direction
- word and halfword read and write access
- address-masked byte write to facilitate quick bit set and clear operations
- address-masked byte read to facilitate quick bit test operations
- maskable interrupt generation based on input value change.

SPI

It has SPI0, SPI1 and SPI2, SPI0 and SPI2 only operates as slave, and it is used by EMCU to control Symmetric Cipher. SPI1 is used by Cortex-M3 to control Symmetric Cipher. It can operate as master or slave.

2.2.2 ASYMMETRIC CIPHER HARDWARE

This Public Key Engine(Asymmetric Cipher) has following features.

- 1 multiplier architecture
- ECC/ECDH/ECDSA operations up to 512 bits
- ECDSA p256
- Supports prime field GF(p) and binary field GF(2^m) fields .

3 Device Functions

3.1.1 AES ENCRYPTION(DECRYPTION) FUNCTION

AESEncrypt control takes the plaintext from 16 bytes and encryptions and outputs ciphertext. The key used for AES encryption are notified to DALPU-4 by E-MCU using the RG_EE_KEY_AES_xN register. AESDecrypt control takes the ciphertext input from 16 bytes and decodes and outputs a plaintext. The keys used for AES decryption are notified to DALPU-4 by E-MCU using the RG_EE_KEY_AES_xN register.

3.1.2 AES ENCRYPTION WRITE(READ) FUNCTION

With AESEncwrite control, 16 bytes of data can be written to the EE_USER_ZONE_Mx area of the EEPROM. AESEncwrite procedures are as follows. E-MCU encrypts the 16 bytes plaintext into ciphertext using the EE_KEY_AES_xN key and writes it to DALPU-4. For DALPU-4, receive the 16 bytes data and decrypt it using the EE_KEY_AES_xN key and store it in the appropriate EEPROM. The AESEncREad procedure is as follows. E-MCU sends control that DALPU-4 reads user data 16 bytes from EEPROM. For DALPU-4, read the 16 bytes plain text from EEPROM and cipher it using EE_KEY_AES_xN key. E-MCU reads the cipher text after waiting DALPU-4 finish decryption. AES Encryption Write (or Read) function encrypt or decrypt 16 bytes at one time. E-MCU can process 4 encryptions (or decryptions) continuously. But E-MCU should not control EEPROM address cross over the page boundary of EEPROM.

3.1.3 EEPROM ERASE FUNCTION

This function enables you to read or clear EEPROM specific information. User use this function for special purpose. Users should review sufficiently before using this feature to determine its intended use. This function is usually not used.

3.1.4 PUF (PHYSICAL UNCRONABLE FUNCTION)

This chapter describes how to create and use the Root Serial corresponding to the unique number of DALPU-4. The Root Serial is a unique number for each device. And this value is a fixed value and cannot be changed. When

3.1.5 RANDOM GENERATION FUNCTION

Random Generator can generate random values in three ways. The first is to generate a random value through SPI0 when the user wants a random value. The second one can be created when a random value is desired in Cortex-M0. Finally, Symmetric cipher can generate and take random values when they are needed.

3.1.6 ECDH (ELLIPTIC CURVE DIFFIE HELLMAN) FUNCTION

Elliptic Curve Diffie-Hellman key exchange is one way to generate key values on an elliptic curve and exchange encryption keys so that they can share a shared key with other keys on an unencrypted network. DALPU-3 supports ECC P-256, P384, P-521 curves etc and supports up to 512 bits.

3.1.7 ECDSA(ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM) FUNCTION

ECDSA implements electronic signatures on elliptic curves and works on ECC P-256, P-384, and P-521 curves etc. ECDSA operations can be executed in both fields GF(p)-prime field or GF(2^m)-binary field. ECDSA signatures can be generated and verified.

4 E-MCU to DALPU-4 Interface

DALPU-4 has SPI0, SPI1, SPI2, UART and GPIO interfaces. In generally DALPU-4 is used as a security function chip not as a MCU. When the DALPU-4 is used as a security function chip, SPI0 and SPI2 is slave mode. When the DALPU-4 is used as a MCU, SPI1 is master mode.

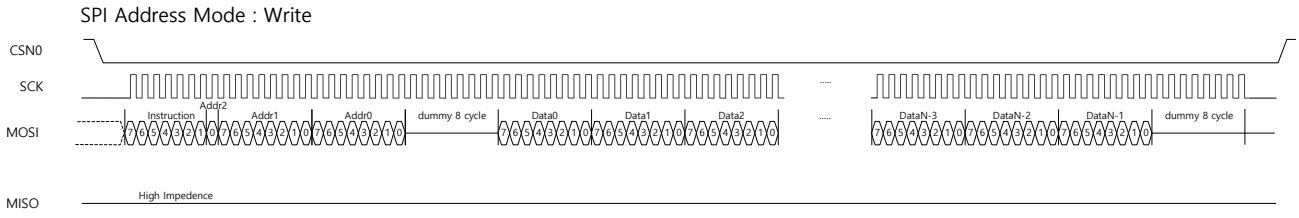
4.1 SPI0, SPI2 Interface

SPI0, SPI2 has write / read protocol as shown in Figure 4-1 and 4-2 below. SPI0, SPI2 is primarily used by external MCUs to control symmetric cipher.

4.1.1 SPI0 PROTOCOL TIMING DIAGRAM

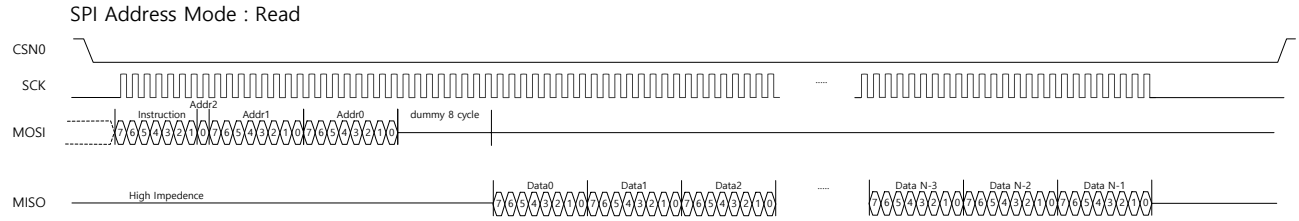
4.1.1.1 SPI0 NORMAL MODE WRITE

Figure 4-1 SPI0 Normal Mode Write in Address Mode



4.1.1.2 SPI0 NORMAL MODE READ

Figure 4-2 SPI0 Normal Mode Read in Address Mode



4.2 SPI1 Interface

SPI1 is used by EMCU to control PKE(Asymmetric Cipher) through Cortex-M3. It basically supports Motorola SPI frame type. The main feature of the Motorola SPI format is that the inactive state and phase of the SCK signal are programmable through the SPO and SPH bits within the SPI1 control register.

SPO, clock polarity

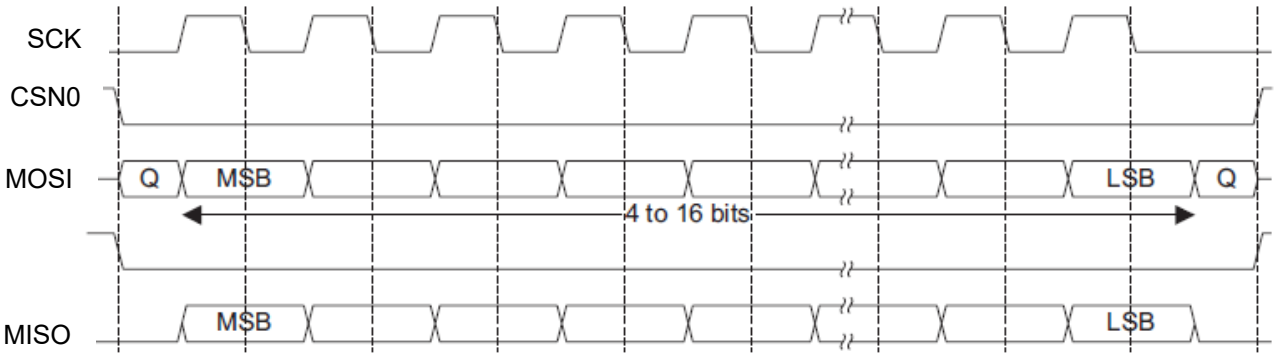
When the SPO clock polarity control bit is LOW, it produces a steady state low value on the SCK pin. If the SPO clock polarity control bit is HIGH, a steady state high value is placed on the SCK pin when data is not being transferred.

SPH, clock phase

The SPH control bit selects the clock edge that captures data and allows it to change state. It has the most impact on the first bit transmitted by either allowing or not allowing a clock transition before the first data capture edge. When the SPH phase control bit is LOW, data is captured on the first clock edge transition. If the SPH clock phase control bit is HIGH, data is captured on the second clock edge transition.

For Frame format used between EMCU and Cortex-M3, set SPO to 0 and SPH to 1. The transfer signal sequence for Motorola SPI format with SPO=0, SPH=1 is shown in Figure 4-3, which covers both single and continuous transfers.

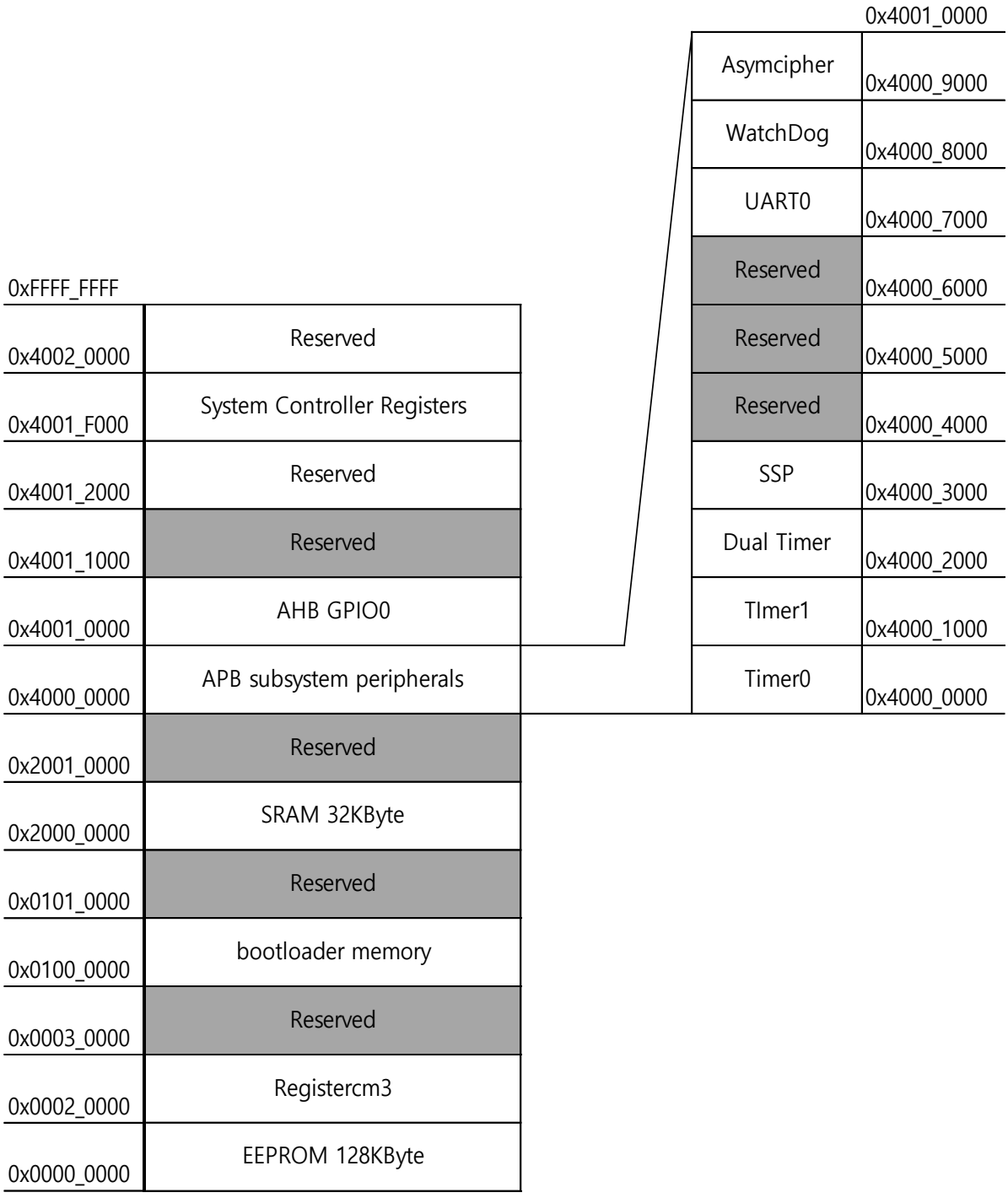
Figure 4-3 Motorola SPI frame format with SPO=0 and SPH=1



5 Address Map

DALPU-4 has CORTEX-M3. The CORTEX-M3 has AMBA bus. And DALPU-4 has address map for the Symmetric Cipher parts and the Asymmetric Cipher parts. CORTEX-M3 AMBA Bus Address Map

Table 5-1 CORTEX-M3 AMBA Bus Address Map



5.1 Symmetric Cipher parts Address Map

Symmetric Cipher parts include EEPROM and the symmetric cipher core. E-MCU can access EEPROM and registers with SPI0 interface.

Table 5-2 EEPROM and register Address Map(SPI0)

ADDR(HEX)	M0 ACCESS	CIP CORE ACCESS	Type	NAME/RANGE			BYTE SIZE(DEC)	DESCRIPTION
				Group1	Group2	Group3		
0x0E800			EEPROM	EE_CM0/ EE_SYMCIP	EE_KEY_ASYMCIP_xN	EE_KEY_ASYMCIP_x0 . . EE_KEY_ASYMCIP_x3	256	1. KEY zone 2. Asymmetric Key storage area (64Byte * 4)
…								
E8FF								
E900				EE_SYMCI	EE_KEY_AES_xN	EE_KEY_AES_x0 . . EE_KEY_AES_x3	256	1. KEY zone 2. Symmetric Key storage area (64Byte * 4)
…								
E9FF								
EA00					EE_RS_xN	EE_RS_x0 . . EE_RS_x3	256	1. KEY zone 2. ROOT SERIAL storage area (64Byte * 4)
…								
EAFF								
EB00					EE_CONFIG	EE_CONFIG_NW 등	1536	1. Configuration zone 2. This zone contains all information to control the CM0 zone, KEY zone, Configuration zone, and User zone.
…								
F0FF								
F100					EE_USER_ZONE_M	EE_USER_ZONE_M01 . . EE_USER_ZONE_M15	3840	1. 15 user zones 2. Store user data.
…								
0x0FFFF								
0x10000				IUM			64	IUM(RESERVED)
…								
1003F								
…			REGISTER	RESERVED				
0x10100				RG_EEBUF			64	Used for EEPROM write operations and BIST test application. Not used for EEPROM read operation.
…								
1013F								
0x10140				RESERVED				
…				RESERVED				
0x10200				RESERVED				
…								

ADDR(HEX)	M0 ACCESS	CIP CORE ACCESS	Type	NAME/RANGE			BYTE SIZE(DEC)	DESCRIPTION
				Group1	Group2	Group3		
1023F								
...				RESERVED				
0x10300				RG_EEBUF	RG_ENCINBUF	RG_ENCINBUF0	16	First ENC input 128 bits buffer share with RG_CMDBUF[15:0].
...								
1030F								
0x10310						RG_ENCINBUF1	16	Second ENC input 128 bits buffer share with RG_CMDBUF[31:16].
...								
1031F								
0x10320					RG_ENCOUTBUF	RG_ENCOUTBUF0	16	First ENC output 128 bits buffer share with RG_CMDBUF[47:32].
...								
1032F								
0x10330						RG_ENCOUTBUF1	16	Second ENC output 128 bits buffer share with RG_CMDBUF[63:48].
...								
1033F								
0x10400					RG_DECINBUF	RG_DECINBUF0	16	First DEC input 128 bits buffer share with RG_CMDBUF[15:0].
...								
1040F								
0x10410						RG_DECINBUF1	16	Second DEC input 128 bits buffer share with RG_CMDBUF[31:16].
...								
1041F								
0x10420					RG_DECOUTBUF	RG_DECOUTBUF0	16	First DEC output 128 bits buffer share with RG_CMDBUF[47:32].
...								
1042F								
0x10430						RG_DECOUTBUF1	16	Second DEC output 128 bits buffer share with RG_CMDBUF[63:48].
...								
1043F								
...				RESERVED				
10500				RESERVED			32	RESERVED
...								
1051F								
...				RESERVED				
0x10600				RG_SYMCIP			512	DALPU-3 control registers.
...								
107FF								

6 Registers

6.1 SPI0 registers

6.1.1 RG_EEBUF BUFFER ADDRESS MAP

RG_EEBUF is used for encryption, decryption and EEPROM write.

SPI0 address width is 17 digits.

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
RG_EEBUF100 Use this register to write and read the data in DALPU-3 internal RG_EEBUF buffer. And it is used to write data to EEPROM. The RG_EEBUF buffer size is 512 bits(64 Bytes).					
0x10100	WR	[7:0]	MCU : RG_EEBUF[0]	RG_EEBUF[0]	0x00
…	WR	[7:0]			0x00
1010F	WR	[7:0]	MCU : RG_EEBUF[15]	RG_EEBUF[15]	0x00
0x10110	WR	[7:0]	MCU : RG_EEBUF[16]	RG_EEBUF[16]	0x00
…	WR	[7:0]			0x00
1011F	WR	[7:0]	MCU : RG_EEBUF[31]	RG_EEBUF[31]	0x00
0x10120	WR	[7:0]	MCU : RG_EEBUF[32]	RG_EEBUF[32]	0x00
…	WR	[7:0]			0x00
1012F	WR	[7:0]	MCU : RG_EEBUF[47]	RG_EEBUF[47]	0x00
0x10130	WR	[7:0]	MCU : RG_EEBUF[48]	RG_EEBUF[48]	0x00
…	WR	[7:0]			0x00
1013F	WR	[7:0]	MCU : RG_EEBUF[63]	RG_EEBUF[63]	0x00
10140			RESERVED		
102FF			RESERVED		
RG_EEBUF300 Use this registers as the input and output buffer when performing AES encryption.					
0x10300	WR	[7:0]	RG_EEBUF[0] MCU : RG_ENCINBUF0[0]	Encoder0 input buffer or Key0 input buffer	0x00

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
…		[7:0]	…	Encoder0 input buffer or Key0 input buffer	0x00
1030F		[7:0]	RG_EEBUF[15] MCU : RG_ENCINBUF0[15]	Encoder0 input buffer or Key0 input buffer	0x00
0x10310		[7:0]	RG_EEBUF[16] MCU : RG_ENCINBUF1[0]	Encoder1 input buffer or Key1 input buffer	0x00
…	WR	[7:0]	…	Encoder1 input buffer or Key1 input buffer	0x00
1031F		[7:0]	RG_EEBUF[31] MCU : RG_ENCINBUF1[15]	Encoder1 input buffer or Key1 input buffer	0x00
0x10320		[7:0]	RG_EEBUF[32] MCU : RG_ENCOUTBUF0[0]	Encoder0 output buffer	0x00
…	WR	[7:0]	…	Encoder0 output buffer	0x00
1032F		[7:0]	RG_EEBUF[47] MCU : RG_ENCOUTBUF0[15]	Encoder0 output buffer	0x00
0x10330		[7:0]	RG_EEBUF[48] MCU : RG_ENCOUTBUF1[0]	Encoder1 output buffer	0x00
…	WR	[7:0]	…	Encoder1 output buffer	0x00
1033F		[7:0]	RG_EEBUF[63] MCU : RG_ENCOUTBUF1[15]	Encoder1 output buffer	0x00
RESERVED			RESERVED		
RG_EEBUF400 Use this registers as the input and output buffer when performing AES decryption.					
0x10400	WR	[7:0]	RG_EEBUF[0] MCU : RG_DECINBUF0[0]	Decoder0 input buffer or Key0 input buffer	0x00

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
…		[7:0]	…	Decoder0 input buffer or Key0 input buffer	0x00
1040F		[7:0]	RG_EEBUF[15] MCU : RG_DECINBUF0[15]	Decoder0 input buffer or Key0 input buffer	0x00
0x10410	WR	[7:0]	RG_EEBUF[16] MCU : RG_DECINBUF1[0]	Decoder1 input buffer or Key1 input buffer	0x00
…		[7:0]	…	Decoder1 input buffer or Key1 input buffer	0x00
1041F		[7:0]	RG_EEBUF[31] MCU : RG_DECINBUF1[15]	Decoder1 input buffer or Key1 input buffer	0x00
0x10420	WR	[7:0]	RG_EEBUF[32] MCU : RG_DECOUTBUF0[0]	Decoder0 output buffer	0x00
…		[7:0]	…	Decoder0 output buffer	0x00
1042F		[7:0]	RG_EEBUF[47] MCU : RG_DECOUTBUF0[15]	Decoder0 output buffer	0x00
0x10430	WR	[7:0]	RG_EEBUF[48] MCU : RG_DECOUTBUF1[0]	Decoder1 output buffer	0x00
…		[7:0]	…	Decoder1 output buffer	0x00
1043F		[7:0]	RG_EEBUF[63] MCU : RG_DECOUTBUF1[15]	Decoder1 output buffer	0x00

6.1.2 REGISTER ADDRESS MAP

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
RG_SOFT_RESET					
0x10600		[7:2]	RESERVED		
	WR	[1]	RG_SWRESET_EE	Reset EEPROM. Test register. To reset EEPROM, write 1 then Write 0 to this register bit. 0 : Normal 1 : RESET	0x0
	WR	[0]	RG_SWRESET	Reset symcipher hardware parts. To reset symchpher hardware parts, write 1 then write 0 to this register bit. 0 : Normal 1 : RESET	0x0
0x10601	A	-	RG_ACCESS	Use this register to control DALPU-3. This register is access register. That is, it is not a register that writes and reads values.	-
0x10602	A	-	RG_ACCESS2	Use this register to control DALPU-3. This register is access register. That is, it is not a register that writes and reads values. Used to control ST0_EEP_OW_CTRL function.	-
0x10603			RESERVED		
RG_ST0_OPMODE Use this register to designate the DALPU-3 main control state. When you finish the control action on each function state, go to ST0_STANDBY State and wait for the next control. The register values corresponding to the state for each function are shown below. ST0_STANDBY is in standby mode and in standby mode. The other state is the state that DALPU-3 performs specific actions. When DALPU-3 ends a particular operation, E-MCU sets this register to ST0_STANDBY state.					

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
0x10604		[7:4]	RESERVED		

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
	WR	[3:0]	RG_ST0_OPMODE	To enter the desired main state, write the following corresponding values in this register : To end control and change to standby state, write a value of " 0x1 " to this reigister and write access to the RG_ACCESS register. 4'h1 : Set main state(ST0) to ST0_STANDBY state. 4'h5 : Set main state(ST0) to ST0_CM0 state. 4'h6 : Set main state(ST0) to ST0_STDSPI state. 4'h7 : Set main state(ST0) to ST0_EE_CFG state. 4'h8 : Set main state(ST0) to ST0_RANDOM state. 4'h9 : Set main state(ST0) to ST0_SYMCIP state. 4'hA : Set main state(ST0) to ST0_OKA state. 4'hB : Set main state(ST0) to ST0_MIDR state. 4'hC : Set main state(ST0) to ST0_PERM_GET state. 4'hF : Set main state(ST0) to ST0_EEP_OW_CTRL state. The correct order of control for this register is as follows. (Correct use examples.) PWR_ON(or SW RESETB)-> ST0_STANDBY -> ST0_STDSPI-> ST0_STANDBY -> ST0_CM0 -> ST0_STANDBY -> ST0_EE_CFG -> ST0_STANDBY (Incorrect use example.) PWR_ON(or SW RESETB) -> ST0_STANDBY -> ST0_STDSPI->	0x0

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
RG_ST1_CM0_OPMODE, RESERVED					
0x10605			RESERVED		
RG_ST1_STDSPI_OPMODE User(E-MCU) can control ST1_STDSPI state with this register.					
0x10606		[7:3]	RESERVED		
		[2:0]	RG_ST1_STDSPI_OPMODE	3'h4 : Set ST1_STDSPI state to ST1_STDSPI_SHA state. Other values : Not defined as a specific action. If user wants to ends ST1_STDSPI_SHA state and writes ' 1 ' to this register.	0x0
RG_ST1_EE_CFG_OPMODE, RESERVED					
0x10607			RESERVED		
RG_ST1_RND_OPMODE User(E-MCU) can control ST1_RND state with this register.					
0x10608		[7:3]	RESERVED		
		[2:0]	RG_ST1_RND_OPMODE	3'h2 : Set ST1_RND state to ST1_RND_GEN_SPI0 state. - E-MCU write '0x2' to this register to create a random value through SPI0 interface. At this state the E-MCU controls random generation function. 3'h4 : Set ST1_RND state to ST1_RND_GEN_SYMCIP state. - E-MCU write '0x2' to this register, to make the symcipher creates a random value. At this state the hardware (symcipher) controls random generation function.	0x0

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
RG_ST1_SYMCIP_OPMODE User(E-MCU) can control ST1_SYMCIP state with this register E-MCU sets this register as ST1_SYMCIP_STANDBY state at the end of a specific operation.					
0x10609		[7:4]	RESERVED		
		[3:0]	RG_ST1_SYMCIP_OPMODE	4'h1: Set ST1_SYMCIP state to ST1_SYMCIP_STANDBY state. 4'h2: Set ST1_SYMCIP state to ST1_SYMCIP_AESEncrypt state. 4'h3: Set ST1_SYMCIP state to ST1_SYMCIP_AESDecrypt state. 4'h4: Set ST1_SYMCIP state to ST1_SYMCIP_AESEncWrite state. 4'h5: Set ST1_SYMCIP state to ST1_SYMCIP_AESEncRead state. 4'h6: Set ST1_SYMCIP state to ST1_SYMCIP_AESKeyLoad state. 4'h7: Set ST1_SYMCIP state to ST1_SYMCIP_RSCreate state. 4'h8: Set ST1_SYMCIP state to ST1_SYMCIP_RSSHARead state. 4'h9: Set ST1_SYMCIP state to ST1_SYMCIP_RSDirRead state. 4'hA: Set ST1_SYMCIP state to ST1_SYMCIP_SHAAuth state. 4'hB: Set ST1_SYMCIP state to ST1_SYMCIP_AESLock state. 4'hC Reserved 4'hD Reserved 4'hE: Set ST1_SYMCIP state to ST1_SYMCIP_STOP0 state. 4'hF: Set ST1_SYMCIP state to ST1_SYMCIP_STOP1 state.	0x0

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
RG_ST1_OKA_OPMODE User(E-MCU) can control ST1_OKA state with this register E-MCU sets this register as ST1_OKA_STANDBY state at the end of a specific operation.					
1060A		[7:3]	RESERVED		
		[2:0]	RG_ST1_OKA_OPMODE	3'h1 : Set ST1_OKA state to ST1_OKA_STANDBY state. 3'h2 : Set ST1_OKA state to ST1_OKA_OKA2_KEY_GEN state. 3'h3 : Set ST1_OKA state to ST1_OKA_OKA2_ED state.	0x0
RG_ST1_MIDR_OPMODE This register controls MIDR counter backup procedure.					
1060B		[7:1]	RESERVED		
		[0]	RG_ST1_MIDR_EEP_RD_START	0 : CONFIG PAGE Read, Backup, RSFLAG SET finish 1 : CONFIG PAGE Read, Backup, RSFLAG SET start	0x0
RG_ST1_PERM_GET_OPMODE, RESERVED					
1060C			RESERVED		
1060D			RESERVED		
1060E			RESERVED		
RG_ST1_EEP_OW_CTRL_OPMODE User(E-MCU) can control ST1_EEP_OW_CTRL state with this register E-MCU sets this register as ST1_EEP_OW_CTRL_STANDBY state at the end of a specific operation.					
1060F		[7:3]	RESERVED		

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
	WR	[2:0]	RG_EEP_OW_CTRL_OPMODE	3'h0 : RESERVED 3'h1 : Set ST1_EEP_OW_CTRL state to ST1_EEP_OW_CTRL_STANDBY state. 3'h2 : Set ST1_EEP_OW_CTRL state to ST1_EEP_OW_CTRL_DETOUR state. 3'h3 : Set ST1_EEP_OW_CTRL state to ST1_EEP_OW_CTRL_DESTROY0 state. 3'h4 : Set ST1_EEP_OW_CTRL state to ST1_EEP_OW_CTRL_DESTROY1 state. 3'h5 ~ 3'h7 : RESERVED	0x0
0x10610			RESERVED		
....					
0x10618			RESERVED		
RG_ST2_SYMCIP_OPMODE User(E-MCU) can control ST2_SYMCIP state with this register. When the E-MCU write registry values, hardware performs control actions corresponding to the values. When control operation is completed, E-MCU sets the register to ST2_AES_STANDBY state. E-MCU sets this register as ST2_AES_STANDBY state at the end of a specific operation.					
0x10619		[7:4]	RESERVED		

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
	WR	[3:0]	RG_ST2_SYMCIP_OPMODE_AES	3'h1 : Set ST2_SYMCIP_OPMODE_AES state to ST2_SYMCIP_OPMODE_AES_STANDBY state. 3'h2 : Set ST2_SYMCIP_OPMODE_AES state to ST2_SYMCIP_OPMODE_AES_INITTIC state. 3'h3 : Set ST2_SYMCIP_OPMODE_AES state to ST2_SYMCIP_OPMODE_AES_KEYTIC state. 3'h4 : Set ST2_SYMCIP_OPMODE_AES state to ST2_SYMCIP_OPMODE_AES_RUNREADY state. ... 3'h8 : Set ST2_SYMCIP_OPMODE_AES state to ST2_SYMCIP_OPMODE_RSCREATE state. 3'h9 : Set ST2_SYMCIP_OPMODE_AES state to ST2_SYMCIP_OPMODE_AES_KEYLOAD state. 3'hA ~3'hD : This state is used by a hardware control part. 3'hE : Set ST2_SYMCIP_OPMODE_AES state to ST2_SYMCIP_OPMODE_AES_DEC_WR state. At this state a input cipher text is decrypted and save to the EEPROM. 3'hF : Set ST2_SYMCIP_OPMODE_AES state to ST2_SYMCIP_OPMODE_AES_ENC_RD state. At this state a plain text from	0x0

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
RG_EE_USER_ZONE_SEL The EEPROM has 15 user zones. One user zone consists of 4 pages. One page consists of 4 sub pages. One page size is 64 Bytes(512-bit). One sub page consists of 16 Bytes(128-bit). The user can write or read in sub-page or page units.					
1061A		[7:6]	RG_EE_UZ_SUBFRAMENUM	2'h0 : [127:0] of selected page 2'h1 : [255:128] of selected page 2'h2 : [383:256] of selected page 2'h3 : [511:384] of selected page	0x0
	WR	[5:4]	RG_EE_UZ_SUBPAGENUM	2'h0 : subpage 0 (0x00 ~ 0x3F) 2'h1 : subpage 1 (0x40 ~ 0x7F) 2'h2 : subpage 2 (0x80 ~ 0xBF) 2'h3 : subpage 3 (0xC0 ~ 0xFF)	0x0
		[3:0]	RG_EE_UZ_PAGENUM	4'h0 : RESERVED 4'h1 : EE_USER_ZONE_M01 (0xF100 ~ 0xF1FF) … 4'hF : EE_USER_ZONE_M15 (0xFF00 ~ 0xFFFF)	0x0
0x1061B			RESERVED		
RG_EE_CFG_RD_RG_EEBUF_ST This is a access register. If the user write any data at this register, One page EEPROM data is read and save to RG_EEBUF. Before wirte a data to the EEPROM, The user controls this register first.					
0x1061C	A	-	RG_EE_CFG_RD_RG_EEBUF_ST	Access control is writing '0x0' to this register.	-

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
RG_ST3_SYMCIP_RSCREATE_OPMODE This register is used to create the root serial(RSCreate operation). User(E-MCU) can control ST3_SYMCIP_RSCREATE_OPMODE state with this register. When the E-MCU write registry values to this register and access RG_ACCESS register, hardware performs control actions corresponding to the values. When control operation is completed, E-MCU sets the register to ST3_SYMCIP_RSCREATE_STANDBY state and access RG_ACCESS register. E-MCU sets this register as ST3_SYMCIP_RSCREATE_STANDBY state at the end of a specific operation.					
0x1061D		[7:3]	RESERVED		

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
	WR	[2:0]	RG_ST3_SYMCIP_RSCREATE_OPMODE	3'h1 : Set ST3_SYMCIP_RSCREATE_OPMODE state to ST3_SYMCIP_RSCREATE_STANDBY state. 3'h2 : Set ST3_SYMCIP_RSCREATE_OPMODE state to ST3_SYMCIP_RSCREATE_ENC1 state. 3'h3 : Set ST3_SYMCIP_RSCREATE_OPMODE state to ST3_SYMCIP_RSCREATE_ENC2 state. 3'h4 : Set ST3_SYMCIP_RSCREATE_OPMODE state to ST3_SYMCIP_RSCREAETE_WR_EEP state. 3'h7 : Set ST3_SYMCIP_RSCREATE_OPMODE state to ST3_SYMCIP_RSCREATE_WR_EEBUF state.	0x0
1061E			RESERVED		

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
RG_ST3_SYMCIP_KEYLOAD_OPMODE This register is used to create the root serial(AESKeyLoad operation). User(E-MCU) can control RG_ST3_SYMCIP_KEYLOAD_OPMODE state with this register. When the E-MCU write registry values to this register and access RG_ACCESS register, hardware performs control actions corresponding to the values. When control operation is completed, E-MCU sets the register to ST3_SYMCIP_KEYLOAD_STANDBY state and access RG_ACCESS register. E-MCU sets this register as ST3_SYMCIP_KEYLOAD_STANDBY state at the end of a specific operation.					
1061F		[7:3]	RESERVED		
	WR	[2:0]	RG_ST3_SYMCIP_KEYLOAD_OPMODE	3'h1 : Set ST3_SYMCIP_KEYLOAD_OPMODE state to ST3_SYMCIP_KEYLOAD_STANDBY state. 3'h2 : ST3_SYMCIP_KEYLOAD_DEC1 state. 3'h3 : ST3_SYMCIP_KEYLOAD_DEC2 state. 3'h4 : ST3_SYMCIP_KEYLOAD_WR_EEP state.	0x0
RG_EE_KEY_AES_CTRL Provides the location of EEPROM storage for the keys used for AESEncrypt, AESDecrypt, AESEncRead and AESEncwrite operations. It also tells the location of the EEPROM to store the keys that were created when performing AESKeyLoad operations. Keys from AESKeyLoad operations are used for AESEncrypt, AESDecrypt, AESEncRead, and AESEncwrite operations.					
0x10620		[7:2]	RESERVED		
	WR	[1:0]	RG_EE_KEY_AES_xN	2'h0 : EE_KEY_AES_x0 2'h1 : EE_KEY_AES_x1 2'h2 : EE_KEY_AES_x2 2'h3 : EE_KEY_AES_x3	0x0
RG_UZID					

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
0x10621			RESERVED		
RG_KL_CTRL Use this register for AESKeyLoad operations.					
0x10622		[7:5]	RESERVED		
	WR	[4]	RG_KL_KeySaveSel	This register selects between the key made with AES decryptor and the value entered with AES text input. 0 : Select a value made with AES decryption. 1 : Select a value that enters the AES text input.	0x0
	WR	[3:2]	RG_KL_TextSel	The register that selects the text message into the AES decryptor input. 2'h0 : Select ciphertext that E-MCU enters as AES text input. 2'h1 : Use the key value EE_KEY_ASYMCIP_x0 as the AES text value that you created as an ECDH result. 2'h2 : Use the full key value made with OKA as the AES text entry.	0x0
	WR	[1:0]	RG_KL_KeySel	A registry that selects the key message that enters the AES decryptor input. 2'h0 : Use the EE_key_SEEDs stored in the EEPROM with the AES key input. 2'h1 : Use the EE_key_AES_x0 stored in the EEPROM with the AES key input. 2'h2 : Use the EE_key_AES_x1 stored in the EEPROM with the AES key input. 2'h3 : Use the EE_key_AES_x2 stored in the EEPROM with the AES key input.	0x0

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
RG_RSCREATE_CTRL Use this register for RSCreate, RSSHARead, and RSDirRead operations.					
0x10623		[7]	RESERVED		
		[6]	DirReadAES_KEY_x3	This register is used to read EE_AES_KEY_x3. You can read EE_AES_KEY_x3 with UID_PWM permission. If the value of RG_EE_RS_xN is " 1 " (RS_x1), or " 3 " (RS_x3), then EE_AES_KEY_x3 is not readable. That is, if the RG_EE_RS_x1 value is " 1 " and DirReadAES_key_x3 (AES_KEY_x3) value is " 1 ", the RS_x1 value can be read.	0x0
	WR	[5:4]	RG_EE_RS_xN	Used for RSCreate, RSSHARead, and RSDirRead operations. In RSCreate mode, you specify the keys to generate. In RSSHARead, RSDirRead mode, specify the key to be read.	0x0
		[3]	RESERVED		
	WR	[2]	RG_RSC_KeySaveSel	Used for RSCreate, RSSHARead operations.	0x0
	WR	[1]	RG_RSC_GEN_RND1	Used for RSCreate operations. When creating RND1 (using the AE256 key as [255:128]), This register must be set to ' 1 ' before the RND_GEN command and clear to ' 0 ' after creation.	0x0

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
	WR	[0]	RG_RSC_GEN_RND0	Used for RSCreate operations. When creating RND0 (using the AE256 key as [127:0]), This register must be set to ' 1 ' before the RND_GEN command and clear to ' 0 ' after creation.	0x0
RG_SHAAUTH_CTRL Used for authentication(SHAAuth) operations. Authentication can be made in two directions. For the first method, DALPU-3 performs authentication. If E-MCU gives the certification message to DALPU-3, DALPU-3 performs the authentication using the authentication message. For the second method, DALPU-3 performs authentication. If DALPU-3 gives the certification message to E-MCU, E-MCU performs the authentication using the authentication message. Both methods are necessary for full certification.					
0x10624		[7:2]	RESERVED		
	WR	[1]	rST2_SYMCIP_SHAAuth_STAY_DP	0 : None 1 : Write " 1 " to complete the SHA-authentication operation.	0x0
	WR	[0]	RG_SHAAuthQuest_SYMCIP_EMCU	0 : E-MCU asks the question. E-MCU creates authentication messages(AuthMsgMCU[255:0] and AuthText[127:0]) and send it to DALPU-3. 1 : DALPU-3 asks the questions. DALPU-3 creates authentication messages(AuthMsgDevice[255:0] and AuthRND[127:0]) and send it to E-MCU.	0x0
0x10625			RESERVED		

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
RG_PERM_GET_CTRL This register writes the information that DALPU-3 require to control ST0_PERM_GET state.					
0x10626		[7:3]	RESERVED		
	WR	[2:0]	RG_EE_PW_ADDR	The register that tells the EEPROM where the password is stored. 3'h5 : EE_SUPER_PW 3'h4 : EE_DETOUR_PW 3'h3 : EE_DESTROY0_PW 3'h2 : EE_DESTROY1_PW 3'h1 : EE_EEPROM_PW 3'h0 : EE_UID_PW	0x1
RG_PERM_GET_CTRL1 This register tells each password permission acquisition state.					
0x10627		[7:6]	RESERVED		
	R	[5]	RG_PERM_SUPER_PASS	0 : Failed to acquire SUPER_PASS password permission(authorization). 1 : Succeeded to acquire SUPER_PASS password permission(authorization).	
	R	[4]	RG_PERM_DETOUR_PASS	0 : Failed to acquire DETOUR_PASS password permission(authorization). 1 : Succeeded to acquire DETOUR_PASS password permission(authorization).	
	R	[3]	RG_PERM_DESTROY0_PASS	0 : Failed to acquire DESTROY0_PASS password permission(authorization). 1 : Succeeded to acquire DESTROY0_PASS password permission(authorization).	

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
	R	[2]	RG_PERM_DESTROY1_PASS	0 : Failed to acquire DESTROY1_PASS password permission(authorization). 1 : Succeeded to acquire DESTROY1_PASS password permission(authorization).	
	R	[1]	RG_PERM_EEPROM_PASS	0 : Failed to acquire EEPROM_PASS password permission(authorization). 1 : Succeeded to acquire EEPROM_PASS password permission(authorization).	
	R	[0]	RG_PERM_UID_PASS	0 : Failed to acquire UID_PASS password permission(authorization). 1 : Succeeded to acquire UID_PASS password permission(authorization).	
RG_PERM_RELEASE If E-MCU writes ' 0 ' to this register in ST0_PERM_GET state, the DALPU-3 returns all acquired password permissions.					
0x10628	A	-	RG_PERM_RELEASE		
RG_PERM_GET_EE_RD_PRE_SP When E-MCU writes ' 0 ' to this register in ST0_PERM_GET state, the DALPU-3 starts the process of obtaining the password permission. In other words, DALPU-3 reads and backs up the corresponding EEPROM configuration area and waits for E-MCU to write the PW_CT.					
0x10629	A	-	RG_PERM_GET_EE_RD_PRE_SP		
0x1062A			RESERVED		
....			RESERVED		
0x10634			RESERVED		
RG_AES_CTRL This register controls the AES and ARIA operations. The operations include encryption and decryption.					

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
0x10635		[7]	RESERVED		-
	WR	[6:4]	RG_AES_OPMODE	Register for the selection of five modes of operation. 3'h0 : ECB 3'h1 : CBC 3'h2 : OFB 3'h3 : CTR 3'h4 : CFB	0x0
	WR	[3]	RG_AES_2_1_FRAME	This register selects frame length. This register selects one or two frame encryption(decryption) processing in ST1_SYMCIP_AESEncrypt state of ST0_SYMCIP state. Once set in two frame mode, the symcipher performs encryption or decryption after E-MCU writes 2 frames. The first frame performance result is saved to RG_EEBUF[383:256] and the second frame performance result is saved to RG_EEBUF [511:384]. 1 : Two frame mode 0 : One frame mode	0x0
	WR	[2]	RG_BYPASS	1(BYPASS), 0(Normal) In BYPASS mode, the LSB bit value is changed for each byte of the input text. Example) INPUT TEXT : 0xC7 ..... 5D OUTPUT TEXT: 0xC6 ..... 5C	0x0
	WR	[1]	RG_128_256	1(128), 0(256)	0x0
	WR	[0]	RG_AES_ARIA	1(AES), 0(ARIA)	0x0

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
0x10636			RESERVED		
0x10637			RESERVED		
RG_SHA_CTRL : 1. Check register contorl oder at TV0610001 1.1. RG_ST0_OPMODE -> RG_ST1_STDSPI_OPMODE -> RG_SHA_CTRL 1.2. RG_SHA_CTRL -> RG_ST0_OPMODE -> RG_ST1_STDSPI_OPMODE					
0x10638		[7:2]	RESERVED		
	WR	[1]	RG_SHA_ONLY_FRM_SEL	1 : SHA only multi frame. 0 : SHA only single frame.	0x0
	WR	[0]	RG_SHA_MF_STOP	1 : SHA multi frame stop. 0 : normal	0x0
0x10639			RESERVED		
1063A			RESERVED		
1063B			RESERVED		
RG_OKA_CTRL This register controls AES in ST0_OKA state.					
1063C		[7:2]	RESERVED		
	WR	[1]	RG_OKA_10_11N	The following function can be controlled only when the value of EE_CONFIG_NW:EE_CONFIG_NW_CTRL 0:EE_OKA_10_11N is '0'. 1 : OKA 1:0 communication 0 : OKA 1:1 or 1:N communication (default)	0x0

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
	WR	[0]	RG_OKA_2_1_FRAME	1. Precautions for running OKA in two frame mode. (1) DALPU-3 conduct a key initialization in the wait time for the first frame input. (2) Therefor, in two frame mode, E-MCU should encrypt(or decrypt) even number of frames. (3) If E-MCU finishes encryption(or decryption) in odd number frames, then following enryption(or decryption), E-MCU must begin with Key generation operation. 1 : 2 frame mode 0 : 1 frame mode	0x0
RG_AES_TVALUE7 This register sets AES twist value.					
1063D	WR	[7:0]	RG_AES_TVALUE7	This register only works in AES. 0x00 : Standard AES Mode 0xXX : Twist AES Mode	0x0
RG_AES_TVALUE8 This register sets AES twist value.					
1063E		[7:4]	RESERVED		
	WR	[3:0]	RG_AES_TVALUE8	This register only works in AES. 0x0 : Standard AES Mode 0xX : Twist AES Mode	0x0
1063F			RESERVED		
....			RESERVED		
1064F			RESERVED		

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
RG_SLEEP_TIMER[12:0] RESET VALUE : 0x1FFF Set this register value to 0 after power on.					
0x10650	WR	[4:0]	RG_SLEEP_TIMER_MSB		0x1F
0x10651	WR	[7:0]	RG_SLEEP_TIMER_LSB		0xFF
0x10652			RESERVED		
….			RESERVED		
0x1065F			RESERVED		
….			RESERVED		
….			RESERVED		
106AF			RESERVED		
106B5	WR	[7:2]	RESERVED		
		[2:0]	RESERVED		
106B6			RESERVED		
….					
106C0			RESERVED		
….					
106DF			RESERVED		
….					
106EF			RESERVED		
106F0		[7:1]	RESERVED		
106F1			RESERVED		
RG_RNDGEN_USER					
10700		[7:1]	RESERVED		

ADDR(HEX)	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
	WR	[0]	RG_RNDGEN_USER	1 : RNDGEN user mode User can enter random values that the user specifies. 0 : RNDGEN normal mode The internal random generator produces a random.	0x0
RG_RNDGEN_EEBUF_CLR This register can clear generated random value in RNDGEN user mode. To clear a random value, write '1', then '0' to this register.					
10701		[7:1]	RESERVED		
	WR	[0]	RG_RNDGEN_EEBUF_CLR	1 : RG EEBUF Clear 0 :	0x0
RG_MCUAuthResult					
10720		[7:1]	RESERVED		
	R	[0]	RG_MCUAuthResult	1 : Auth Pass 0 : Auth Fail	0x0

6.2 CORTEX-M0 registers

6.2.1 SSP(SPI1) FEATURES

• Compliance to the AMBA Specification (Rev 2.0) for easy integration into SoC implementation.

• Master or slave operation.

• Programmable clock bit rate and prescale.

• Separate transmit and receive first-in, first-out memory buffers, 16 bits wide, 8 locations deep.

• Programmable choice of interface operation, SPI, Microwire, or TI synchronous serial.

• Programmable data frame size from 4 to 16 bits.

• Independent masking of transmit FIFO, receive FIFO, and receive overrun interrupts.

• Internal loopback test mode available.

6.2.2 SSP(SPI1) OPERATION

Following reset, the PrimeCell SSP logic is disabled and must be configured when in this state.

Control registers SSPCR0 and SSPCR1 need to be programmed to configure the peripheral as a master or slave operating under one of the following protocols:

• Motorola SPI

• Texas Instruments SSI

• National Semiconductor.

The bit rate, derived from the external SSPCLK, requires the programming of the clock prescale register SSPCPSR.

You can either prime the transmit FIFO, by writing up to eight 16-bit values when the PrimeCell SSP is disabled, or allow the transmit FIFO service request to interrupt the CPU. Once enabled, transmission or reception of data begins on the transmit (SSPTXD) and receive (SSPRXD) pins.

6.2.3 SSP(SPI1) REGISTERS

SSP Base Address : 0x4000_2200

SSP Register Address : SSP Base Address + Offset

Table 6-1 Summary of SSP Registers

ADDR(HEX) Offset	Type	Width	NAME	DESCRIPTION	RESET
0x00	WR	16	SSPCR0	Control register 0.	0x0
0x04	WR	4	SSPCR1	Control register 1.	0x0
0x08	WR	16	SSPDR	Receive FIFO(read) and transmit FIFO data register(write).	0x---
0x0C	R	5	SSPSR	Status register.	0x03
0x10	WR	8	SSPCPSR	Clock prescale register.	0x0
0x14	WR	4	SSPIMSC	Interrupt mask set and clear register.	0x0
0x18	R	4	SSPRIS	Raw interrupt status register.	0x8
0x1C	R	4	SSPMIS	Masked interrupt status register.	0x0

ADDR(HEX) Offset	Type	Width	NAME	DESCRIPTION	RESET
0x20	W	4	SSPICR	Interrupt clear register.	0x0
0x24	WR	2	SSPDMACR	DMA control register.	0x0

Table 6-2 SSP Registers Details

ADDR(HEX) Offset	WR	BIT	NAME/RANGE	DESCRIPTION	RESET VALUE
Control register 0 (SSPCR0) SSPCR0 is control register 0 and contains five bit fields that control various functions within the SSP.					
0x0	WR	[15:8]	SCR	Serial clock rate. The value SCR is used to generate the transmit and receive bit rate of the SSP. The bit rate is: where CPSDVSR is an even value from 2-254, programmed through the SSPCPSR register and SCR is a value from 0-255.	0x0
	WR	[7]	SPH	**SSPCLKOUT** phase, applicable to Motorola SPI frame format only.	0x0
	WR	[6]	SPO	**SSPCLKOUT** polarity, applicable to Motorola SPI frame format only.	0x0
	WR	[5:4]	FRF	Frame format 00 Motorola SPI frame format. 01 TI synchronous serial frame format. 10 National Microwire frame format. 11 Reserved, undefined operation.	0x0
	WR	[3:0]	DSS	Data Size Select: 0000 Reserved, undefined operation. 0001 Reserved, undefined operation. 0010 Reserved, undefined operation. 0011 4-bit data. 0100 5-bit data. 0101 6-bit data. 0110 7-bit data. 0111 8-bit data. 1000 9-bit data. 1001 10-bit data. 1010 11-bit data. 1011 12-bit data. 1100 13-bit data. 1101 14-bit data. 1110 15-bit data. 1111 16-bit data.	0x0
Control register 1 (SSPCR1) SSPCR1 is the control register 1 and contains four different bit fields, that control various functions within the SSP.					
0x4	-	[15:4]	-	Reserved, read unpredictable, should be written as 0.	-
	WR	[3]	SOD	Slave-mode output disable. This bit is relevant only in the slave mode, MS=1. In multiple-slave systems, it is possible for an SSP master to broadcast a message to all slaves in the system while ensuring that only one slave drives data onto its serial output line. In such systems the **RXD** lines from multiple slaves could be tied together. To operate in such systems, the SOD bit can be set if the SSP slave is not supposed to drive the **SSPTXD** line: 0 SSP can drive the **SSPTXD** output in slave mode.	0x0

				1 SSP must not drive the **SSPTXD** output in slave mode.	
	WR	[2]	MS	Master or slave mode select. This bit can be modified only when the SSP is disabled, SSE=0: 0 Device configured as master, default. 1 Device configured as slave.	0x0
	WR	[1]	SSE	Synchronous serial port enable: 0 SSP operation disabled. 1 SSP operation enabled.	0x0
	WR	[0]	LBM	Loop back mode: 0 Normal serial port operation enabled. 1 Output of transmit serial shifter is connected to input of receive serial shifter internally.	0x0
Data register (SSPDR) SSPDR is the data register and is 16-bits wide. When SSPDR is read, the entry in the receive FIFO, pointed to by the current FIFO read pointer, is accessed. As data values are removed by the SSP receive logic from the incoming data frame, they are placed into the entry in the receive FIFO, pointed to by the current FIFO write pointer. When SSPDR is written to, the entry in the transmit FIFO, pointed to by the write pointer, is written to. Data values are removed from the transmit FIFO one value at a time by the transmit logic. It is loaded into the transmit serial shifter, then serially shifted out onto the **SSPTXD** pin at the programmed bit rate. When a data size of less than 16 bits is selected, the user must right-justify data written to the transmit FIFO. The transmit logic ignores the unused bits. Received data less than 16 bits is automatically right-justified in the receive buffer.					
0x8	WR	[15:0]	DATA	Transmit/Receive FIFO: **Read** Receive FIFO. **Write** Transmit FIFO. You must right-justify data when the SSP is programmed for a data size that is less than 16bits. Unused bits at the top are ignored by transmit logic. The receive logic automatically right-justifies.	-
Status register (SSPSR) SSPSR is a RO status register that contains bits that indicate the FIFO fill status and the SSP busy status.					
0x10	-	[15:5]	-	Reserved, read unpredictable, should be written as 0.	-
	R	[4]	BSY	SSP busy flag (read only): 0 SSP is idle. 1 SSP is currently transmitting and/or receiving a frame or the transmit FIFO is not empty	0x0
	R	[3]	RFF	Receive FIFO full (read only): 0 Receive FIFO is not full. 1 Receive FIFO is full.	0x0
	R	[2]	RNE	Receive FIFO not empty, (read only): 0 Receive FIFO is empty. 1 Receive FIFO is not empty.	0x0
	R	[1]	TNF	Transmit FIFO not full, (read only): 0 Transmit FIFO is full. 1 Transmit FIFO is not full.	0x1
	R	[0]	TFE	Transmit FIFO empty, (read only): 0 Transmit FIFO is not empty. 1 Transmit FIFO is empty.	0x1
Clock prescale register (SSPCPSR) SSPCPSR is the clock prescale register and specifies the division factor by which the input **SSPCLK** must be internally divided before further use. The value programmed into this register must be an even number between 2-254. The least significant bit of the programmed number is hard-coded to zero. If an odd number is written to this register, data read back from this register has the least significant bit as zero.					
0x14	-	[15:8]	-	Reserved, read unpredictable, must be written as 0.	-
	WR	[7:0]	CPSDVSR	Clock prescale divisor. Must be an even number from 2-254, depending on the frequency of **SSPCLK**. The least	0x0

				significant bit always returns zero on reads.	
Interrupt mask set or clear register (SSPIMSC) The SSPIMSC register is the interrupt mask set or clear register. It is a RW register. On a read this register gives the current value of the mask on the relevant interrupt. A write of 1 to the particular bit sets the mask, enabling the interrupt to be read. A write of 0 clears the corresponding mask. All the bits are cleared to 0 when reset.					
0x18		[15:4]	-	Reserved, read as zero, do not modify.	-
	WR	[3]	TXIM	Transmit FIFO interrupt mask: 0 Transmit FIFO half empty or less condition interrupt is masked. 1 Transmit FIFO half empty or less condition interrupt is not masked.	0x0
	WR	[2]	RXIM	Receive FIFO interrupt mask: 0 Receive FIFO half full or less condition interrupt is masked. 1 Receive FIFO half full or less condition interrupt is not masked.	0x0
	WR	[1]	RTIM	Receive timeout interrupt mask: 0 Receive FIFO not empty and no read prior to timeout period interrupt is masked. 1 Receive FIFO not empty and no read prior to timeout period interrupt is not masked.	0x0
	WR	[0]	RORIM	Receive overrun interrupt mask: 0 Receive FIFO written to while full condition interrupt is masked. 1 Receive FIFO written to while full condition interrupt is not masked.	0x0
Raw interrupt status register (SSPRIS) The SSPRIS register is the raw interrupt status register. It is a RO register. On a read this register gives the current raw status value of the corresponding interrupt prior to masking. A write has no effect.					
0x1C	-	[15:4]	-	Reserved, read as zero, do not modify	-
	R	[3]	TXRIS	Gives the raw interrupt state, prior to masking, of the SSPTXINTR interrupt	0x1
	R	[2]	RXRIS	Gives the raw interrupt state, prior to masking, of the SSPRXINTR interrupt	0x0
	R	[1]	RTRIS	Gives the raw interrupt state, prior to masking, of the SSPRTINTR interrupt	0x0
	R	[0]	RORRIS	Gives the raw interrupt state, prior to masking, of the SSPRORINTR interrupt	0x0
Masked interrupt status register (SSPMIS) The SSPMIS register is the masked interrupt status register. It is a RO register. On a read this register gives the current masked status value of the corresponding interrupt. A write has no effect.					
0x20	-	[15:4]	-	Reserved, read as zero, do not modify	-
	R	[3]	TXMIS	Gives the transmit FIFO masked interrupt state, after masking, of the SSPTXINTR interrupt	0x0
	R	[2]	RXMIS	Gives the receive FIFO masked interrupt state, after masking, of the SSPRXINTR interrupt	0x0
	R	[1]	RTMIS	Gives the receive timeout masked interrupt state, after masking, of the SSPRTINTR interrupt	0x0
	R	[0]	RORMIS	Gives the receive over run masked interrupt status, after masking, of the SSPRORINTR interrupt	0x0
Interrupt clear register (SSPICR) The SSPICR register is the interrupt clear register and is write-only. On a write of 1, the corresponding interrupt is cleared. A write of 0 has no effect.					
0x24	-	[15:2]	-	Reserved, read as zero, do not modify	-
	W	[1]	RTIC	Clears the SSPRTINTR interrupt	0x0
	W	[0]	RORIC	Clears the SSPRORINTR interrupt	0x0

DMA control register (SSPDMACR) The SSPDMACR register is the DMA control register. It is a RW register. All the bits are cleared to 0 on reset.					
0x28	-	[15:2]	-	Reserved, read as zero, do not modify	-
	WR	[1]	TXDMAE	Transmit DMA Enable. If this bit is set to 1, DMA for the transmit FIFO is enabled	0x0
	WR	[0]	RXDMAE	Receive DMA Enable. If this bit is set to 1, DMA for the receive FIFO is enabled.	0x0

7 EEPROM Configuration

8 Revision History

Version	Revision	Date	Comments	Editors
004		2018.11.01	Operation Cuicuit	Justin KIM
003		~	Editing.	HCLEE
002		2018.01.16	Release to Dream Security wo SPI0 register.	HCLEE
001	-	2017.12.07	Document creation.	HCLEE