

Implementation Of Security Framework

Major Project Report

Submitted in fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Mit Patel

(17MECE13)



Electronics & Communication Engineering Department

Institute of Technology

Nirma University

Ahmedabad-382 481

May 2019

Implementation Of Security Framework

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

By

Mit Patel
(17MECE13)

Under the guidance of

External Project Guide:

Mr. Rahul Dhobi

Project Lead

System Level Solution.(I) Pvt. Ltd.,

Anand.

Internal Project Guide:

Dr Dilip Kumar Kothari

Professor and Head EC, EC Department,

Institute of Technology,

Nirma University, Ahmedabad.



Electronics & Communication Engineering Department

Institute of Technology-Nirma University

Ahmedabad-382 481

May 2019

Declaration

This is to certify that

1. The thesis comprises my original work towards the degree of Master of Technology in Embedded Systems at Nirma University and has not been submitted elsewhere for a degree.
2. Due acknowledgment has been made in the text to all other material used.

- Mit Patel
17MECE13

Disclaimer

“The content of this paper does not represent the technology, opinions, beliefs, or positions of System Level Solutions (I) Pvt.Ltd., its employees, vendors, customers, or associates.”



Certificate

This is to certify that the Major Project entitled **“Implementation of Security Framework”** submitted by **Mit Patel (17MECE13)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of our knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Date:

Place: Ahmedabad

Dr Dilip Kumar Kothari

Internal Guide

Dr. N. P. Gajjar

Program Coordinator

Dr Dilip Kumar Kothari

Head, EC Department

Dr. Alka Mahajan

Director, ITNU

Certificate

This is to certify that the Major Project entitled “**Implementation of Security Framework**” submitted by **Mit Patel (17MECE13)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.

Mr. Rahul Dhobi

Project Lead

System Level Solutions(I)Pvt.LTD

Anand

Statement of Originality

I, **Mit Patel**, Roll. No. **17MECE13**, give undertaking that the Major Project entitled **Implementation of Security Framework** submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Electronics and communication (Embedded System)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Date:

Place:

Endorsed by
Dr Dilip Kumar Kothari

Acknowledgements

I would like to express my gratitude and sincere thanks to **Dr Dilip Kumar Kothari**, PG Coordinator of M.Tech Embedded Systems and **Dr Dilip Kumar Kothari** for guidelines during the review process.

I take this opportunity to express my profound gratitude and deep regards to **Dr Dilip Kumar Kothari**, guide of my internship project for his exemplary guidance, monitoring and constant encouragement.

I would also like to thank **Mr. Rahul Dhobi**, external guide of my internship project from **System Level Solution (I).Pvt.Ltd**, for guidance, monitoring and encouragement regarding the project.

- Mit Patel
17MECE13

Abstract

In today's era, data security is the most essential. A current embedded device provides software-based security. So, hardware-based security is needed. The Nuvoton PFM m487 has an ARM Cortex M4 core. In ARM Cortex M4 controller supports Memory Protection Unit (MPU). MPU provides hardware-based security. MPU supports up to 8 regions and size of MPU is 4GB. MPU can be configured using MPU registers. Configuring MPU, we can set different attributes using for user and kernel access. MPU permits access rules to be discovered for privileged access and user program access. Memory Protection Unit provides security for unauthorized access. We can set critical data as no access so access from that region or that address gives memory fault. We can store critical data in flash memory. Communication between flash memory and other devices there is flash memory controller. Flash memory controller is interfaced with flash memory. It handles all signals required by the flash memory. Flash memory always kept in read-only mode. Flash memory can be divided into several parts like boot loader, XOM, OTP. One part of flash memory is for user configured. There is one part is called OTP (One Time Programmable) memory. OTP can be programmed using ISP registers like a command, trigger, status, read and write. We have to program OTP very carefully because once we write data in OTP, it can not be changed or modified. OTP has the 2KB size and 1KB lock bit. We can read and write in OTP in 64-bit chunks. Keys can be stored in OTP. We are stored certificate into OTP which is useful for the firmware update. We can store cryptography keys, certificates and digital signature in One time Programmable memory. Now when firmware update request comes in embedded devices, it checks cryptography keys which are stored in OTP, if keys, digital signature matches, the firmware updating process further.

Contents

Declaration	iii
Disclaimer	iv
Certificate	v
Statement of Originality	vii
Acknowledgements	viii
Abstract	ix
1 Introduction	3
1.1 Board specification	4
2 Memory Protection Unit	5
2.1 Memory Mapping in MPU	5
2.2 Memory Types in ARMv8	7
2.2.1 Normal Memory	7
2.2.2 Device Memory	9
2.3 MPU Registers	10
2.3.1 MPU TYPE	10
2.3.2 MPU CTRL	11
2.3.3 MPU RNR	12
2.3.4 MPU RBAR	12
2.3.5 MPU RASR	13
2.4 Initializing and configuring an MPU	17
2.4.1 SETTING UP THE MPU	17
2.4.2 pseudo code for MPU configuration	17
2.4.3 MPU OUTPUT	19
3 Flash Memory	20
3.1 Flash Memory Controller registers	21
3.2 OTP (One Time Program)	22
3.2.1 ISP Control Register	24
3.2.2 ISP Address Register	24
3.2.3 ISP Data Register	25
3.2.4 ISP Command Register	25
3.2.5 ISP Trigger Register	27

3.2.6	ISP Data Flash Address Register	27
3.2.7	ISP Status Register	28
3.3	OTP Programming Flow chart	28
4	Cryptography	31
4.1	Definition and History of cryptography	31
4.2	classification of cryptography	32
4.2.1	AES(Advance Encryption Standard)	34
4.3	OUTPUT	37
4.4	Use Cases	38
5	SUMMARY	40
	Bibliography	41

List of Tables

2.1	MPU Type Register	10
2.2	MPU Control Register	11
2.3	MPU Region Number Register Register	12
2.4	MPU Region Base Address Register Register	13
2.5	MPU Region Attributer Register	14
2.6	Access Permission Configurations	15
2.7	Memory Attributes	15
2.8	Inner and Outer Cache Policy	16
3.1	FMC Control Registers	22
3.2	ISP Registers	23
3.3	ISP Control Register	24
3.4	ISP Address Register	25
3.5	ISP Data Register	25
3.6	ISP Command Register Bits	26
3.7	ISP Command Register	26
3.8	ISP Trigger Register	27
3.9	ISP Data Flash Register	27
3.10	ISP Status Register	28

Chapter 1

Introduction

The ARM cortex m4 core is a high performance 32-bit processor design. It has high-speed performance core, 3 stage pipeline, low power and Harvard architecture. The cortex m4 supports Thumb mode-2 technology. We are using NUVOTON PFM m487 board which is the NuMicro Family Cortex-M4 based MCUs provide a high-performance system. The Nuvoton PFM m487 board has 512 computer memory unit embedded twin bank non-volatile storage supports OTA (Over-The-Air) code upgrade, and the 160 KB embedded SRAM includes 32 KB cache to speed up external SPI Flash code execution. It has an optional Memory Protection Unit supports up to 4GB memory space. It provides hardware-based security. MPU permits access rules to be discovered for privileged access and user program access. once Associate in Nursing access rule is desecrated, a fault exception is generated, and also the fault exception handler are able to analyze the matter. Communication between flash memory and other devices, there is a Flash memory inside MPU which always keeps in Read-only mode so critical data like bootloader, security keys can be prevented from the unwanted access. Communication between flash memory and other devices there is Flash memory Controller. In flash memory, there is a One-Time-Programable memory. In nuvoton m487 supports 2Kb OTP data memory and 1KB of lock bit. Once we write data in OTP, it can not be changed. The comprehensive hardware cryptography engines in the M487 board supports ECC (Eclipse Curve Cryptography), AES-256, DES, Triple DES, SHA-512, and HMAC. The AES accelerator is an implementation fully compliant with the AES (Advanced Encryption Standard) encryption and decryption algorithm. The key size of AES algorithmic program is 128,192,256 bits.

Flash memory always in read only mode because it contains many critical data like security key,bootloader. Communication between data stored in flash memory and computer device or peripherals using flash memory controller. Flash memory has been divided into many different parts and each part has a special communication purpose for a devices.

1.1 Board specification

we are using Nuvoton m487 Board. It has arm cortex m4 core. It supports arm mbed IOT device platform.The left portion of this board is the M487 Platform that includes the target chip M487 MCU which embedded ARM Cortex -M4 core with DSP extensions and a Floating Point Unit (FPU) and the other related on-board application parts and connectors. The right portion of this board is a Nu-Link-Me ICE Bridge based on the SWD (Serial Wire Debug) interface connected with the target chip, allowing user to program the application code to the flash of target chip through the USB port from PC Host. It has optinal Memory Protection Unit. Nuvoton m487 board figure shown in below:

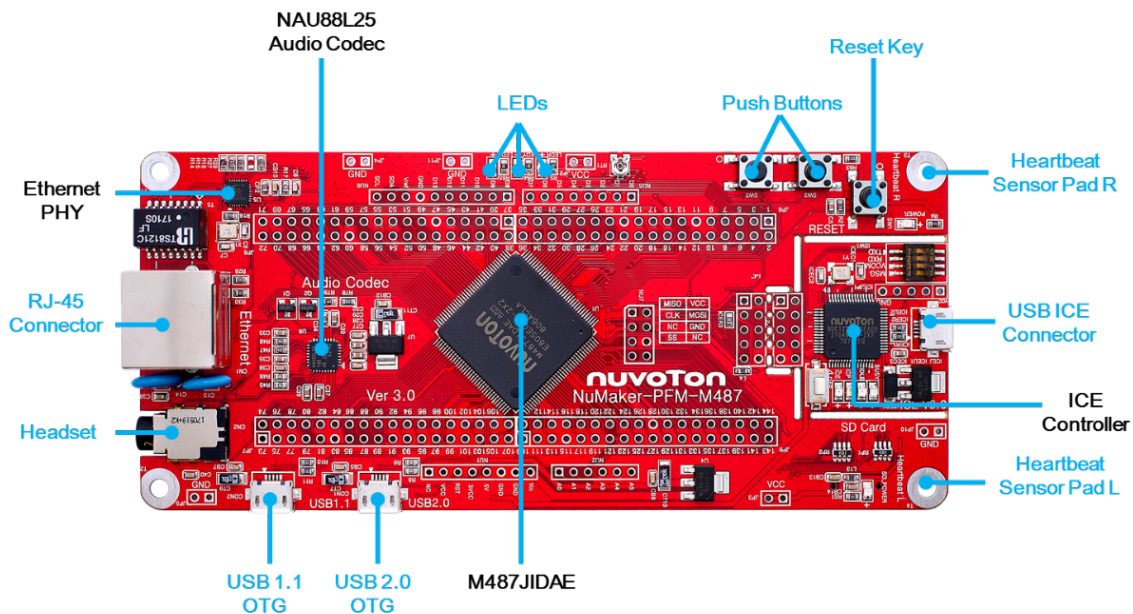


Figure 1.1: NUVOTON m487 Board

Chapter 2

Memory Protection Unit

2.1 Memory Mapping in MPU

ARMv8-M could be a memory-mapped design with facultative MPU. Before the MPU is designed or initialized, the memory is split into eight x 512MB segments. The cortexM4 processor contains a total of 4GB of address area. The program code are often placed within the code region, the Static Random Access Memory(SRAM) fetches and knowledge accesses area unit distributed at the same time on 2 separate bus interface. The SRAM memory vary is for connecting internal SRAM. in line with this region is distributed via the system interface bus. during this region, the 32-MB vary is outlined as a touch band alias. inside the 32-bit-band alias memory vary, every word address area represents 32-bit-band region. a knowledge—a knowledge—an information write access to third-bit band alias memory vary are going to be born-again to anatomic READ-MODIFY-WRITE operation to the bit-band region thus on permit a program to line or clean individual data bits within the memory.[1]

Vendor Specific Memory	0xFFFF_FFFF
Private Peripheral Bus	0xE010_0000 0xE00F_FFFF
External Devices	0xE000_0000 0xDFFF_FFFF
External RAM	0xD000_0000 0xAFFF_FFFF
Peripheral	0x6000_0000 0x5FFF_FFFF
SRAM	0x4000_0000 0x3FFF_FFFF
Code	0x2000_0000 0x1FFF_FFFF
	0x0000_0000

Figure 2.1: Memory Map

2.2 Memory Types in ARMv8

2.2.1 Normal Memory

The Normal Memory sort is meant to be used for MPU regions that square measure wont to access general instruction or knowledge memory. Normal memory permits the processor to perform some operation optimizations, like access re-ordering or merging. Normal memory put together permits memory to be cached and is suitable for holding viable code.

Normal memory should not be wont to access peripheral MMIO registers. The Device memory type is supposed for that use. Normal memory will have many attributes applied thereto. the subsequent memory attributes square measure available: Cacheability recollections is cacheable or non-cacheable. Shareability traditional memory is shareable or Non-shareable. eXecute ne'er recollections could also be marked as possible or eXecute ne'er (XN). Cacheability The cacheability attribute is additional divided: Cache policy Write-Through / Write-Back. Allocation Cache line allocation hints, for scan and write accesses. Transient hint slightly to the cache that the data might exclusively be needed among the cache concisely.

The design supports 2 levels of cache attributes. These unit of measurement the inner cache and outer cache attributes. Typically, the inner cache attribute is employed by any integrated caches, whereas the outer cache attributes square measure exported on mistreatment the bus system sideband signals. Depending on the processor implementation, the inner cache attributes will even be exported to the memory system pattern further sideband signals.

Configuring academic degree MPU region with a cacheable memory kind does not imply that the knowledge ought to be cached, however solely indicates to the hardware that it would be cached. If a locality is printed as cacheable, package takes responsibility for acting any necessary cache maintenance operations. Shareability several systems have multiple bus masters, either multiple processors or a combination of processors and different masters like DMA engines. The shareability attribute permits software package to advertise to the hardware that of these devices ought to be able to see any updates to a particular house of memory.

The architecture manages this by grouping all masters into shareability groups. Non-

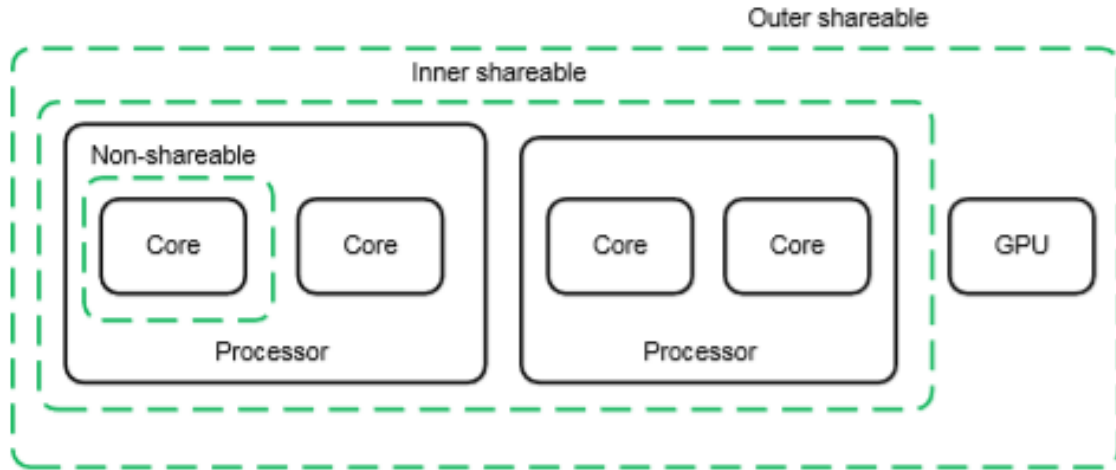


Figure 2.2: SHAREABLE MEMORY

shareable:- This represents memory accessible only by a single processor or other agent, so memory accesses never have to be synchronized with other processors. Only the processor itself must see the information, though it can be made visible to other agents. Inner Shareable:- This represents a shareability domain that can be shared by multiple masters, but not necessarily all the agents in the system. A system might have multiple Inner Shareable domains. An operation that have an effect ons one Inner Shareable domain doesn't affect different Inner Shareable domains within the system. All agents within this domain can be able to see the memory.

Outer Shareable:- An Outer Shareable (OSH) domain reorder is shared by multiple agents and can consist of one or more Inner Shareable domains. An operation that affects Associate in Nursing Outer Shareable domain conjointly implicitly affects all Inner Shareable domains within it. However, it doesn't otherwise behave as Associate in Nursing Inner Shareable operation.

Defining the shareability of a memory region imposes some useful needs on the hardware however it doesn't limit however the hardware implements that practicality.

The OSH requirement is that all masters in the outer sharable domain can see the effects of any memory updates: In a system without caches and just one level of RAM any master can see any memory update. in an exceedingly system with caches, not all masters will access all caches. The system might employ hardware cache coherency to make updates visible.

2.2.2 Device Memory

Device memory should be used for memory regions that control peripheral management registers. Some of the optimizations that are allowed to Normal memory, such as access merging or repeating, would be unsafe to a peripheral register. The Device memory type has several attributes: G or nG Gathering or non-Gathering. Multiple accesses to a tool are often incorporate into one group action aside from operations with memory ordering linguistics, as an example, memory barrier directions, load acquire/store release.

R or nR Reordering. E or nE Early Write Acknowledge (similar to bufferable). Only four combinations of these attributes are valid: Device-nGnRnE Device-nGnRE Device-nGRE Device-GRE Device-nGnRnE is equivalent to ARMv7-M Strongly Ordered memory type and Device-nGnRE is equivalent to ARMv7-M Device memory. Typically peripheral control registers must be either Device-nGnRE, or Device-nGnRnE. This prevents reordering of the transactions in the programming sequences. Device-nGRE and Device-GRE memory sorts are often helpful for peripherals wherever access sequence and ordering doesn't have an effect on results, as an example, in image or show buffers during a show interface. If the bus interface of such peripheral can only accept certain transfer sizes, the peripheral must be set to Device-nGRE. [1]

2.3 MPU Registers

2.3.1 MPU TYPE

The MPU contains variety of registers. the primary one is that the MPU sort register. The MPU kind register is used to verify whether or not or not the MPU is fitted. If the DREGION field is browse as zero, the MPU isn't enforced. This register is browse solely. The MPU TYPE register is read only.

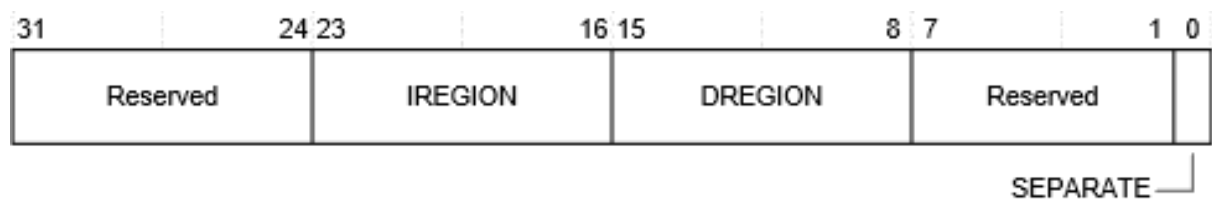


Figure 2.3: MPU TYPE REGISTER

bit	field	Description
31:16	RESERVED	RESERVED
15:8	DREGION	Number of MPU regions that are supported by the MPU in selected security state.
7:1	RESERVED	Reserved
0	SEPERATE	Indicate support for separate instruction information address regions. ARMv8-M solely supports unified MPU regions and so this bit is ready to zero.

Table 2.1: MPU Type Register

2.3.2 MPU CTRL

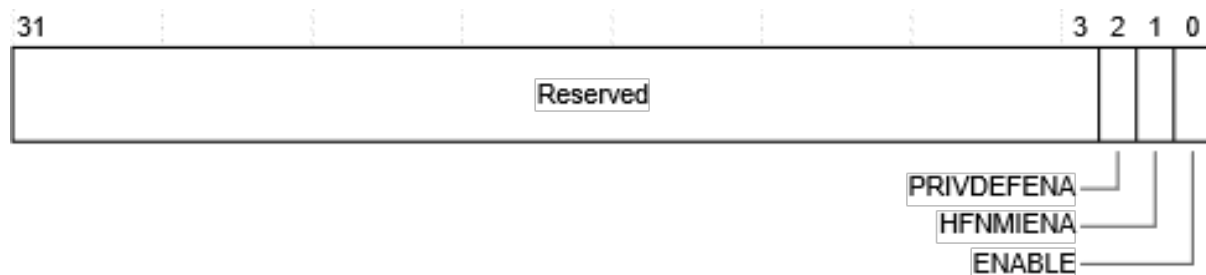


Figure 2.4: MPU CONTROL REGISTER

The MPU is controlled by variety of registers. The first one is the MPU Control register (see Table). This register has three control bits. After reset, the reset value of this register is zero, which disables the MPU. To modify the MPU, the software package ought to come upon the settings for every MPU regions, and then, set the modify bit within the MPU management register.

bit	field	Description
31:3	RESERVED	RESERVED
2	PRIVDEFENA	Privileged background region enable. When set to 1, this enables the default memory map for privilege code when the address accessed does not map into any MPU region. Unprivileged accesses to unmapped addresses result in faults. When cleared to 0, all accesses to unmapped addresses result in faults.
1	HFNMIENA	MPU Enable for HardFault and NMI (Non-Maskable Interrupt). When set to 1, MPU access rules apply to HardFault and NMI handlers. When cleared to 0, HardFault and NMI handlers bypass MPU configuration as if MPU is disabled.
0	ENABLE	Enable control. When set to 1, the MPU is enabled. When cleared to 0, the MPU is disabled.

Table 2.2: MPU Control Register

Setting the alter bit within the MPU management register is typically the last step within the MPU setup code. Otherwise, the MPU would possibly generate faults accidentally before the region configuration is finished.

2.3.3 MPU RNR



Figure 2.5: MPU REGION NUMBER REGISTER

The MPU Region Number Register selects the region that is accessed by the MPU-RBAR and MPU-RLAR.

bit	field	Description
31:8	RESERVED	Base address of the region; N depends on the region size for example, a sixty four K size region can have a base address field of [31:8].
7:0	REGION	This field overrides the MPU Region variety register if VALID is 1; otherwise, it's neglected.

Table 2.3: MPU Region Number Register Register

2.3.4 MPU RBAR

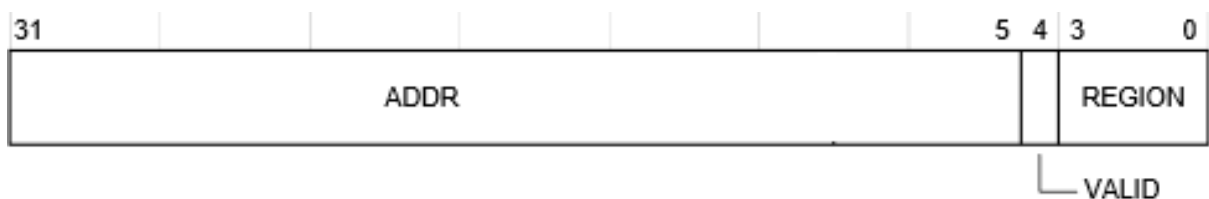


Figure 2.6: MPU REGION BASE ADDRESS REGISTER

The starting address of each region is defined by the MPU Region Base Address register (see Table 13.4). Using the VALID and REGION fields in this register, we can skip the step of programming the MPU Region Number register. This might reduce the complexity of the program code, especially if the whole MPU setup is defined in a lookup table.

bit	field	Description
31:5	ADDR	Base address of the region; N relies on the region size for example, a sixty four K size region can have a base address field of [31:16]
4	VALID	If this is often one, the REGION outlined in bit [3:0] are going to be employed in this programming step; otherwise, the region selected by the MPU Region variety register is employed.
3:0	REGION	This field overrides the MPU Region variety register if VALID is 1; otherwise, it's unnoticed.

Table 2.4: MPU Region Base Address Register Register

The base address is aligned to the size of the region. For example, a 64KB region must be aligned on a multiple of 64KB, for example, at 0x00010000 or 0x00020000. The ADDR field is bits[31:N] of the MPU RBAR. The region size, as specified by the SIZE field in the MPU RASR, defines the value of N:

$$N = \text{Log}_2(\text{Region size in bytes})$$

If the region size is configured to 4GB, in the MPU RASR, there is no valid ADDR field. The region occupies the complete memory map, and the base address is 0x00000000. The base address is aligned to the size of the region. For Example, a 32KB region must be aligned on a multiple of 32KB, For Example , at 0x00000000 or 0x00008000.

2.3.5 MPU RASR

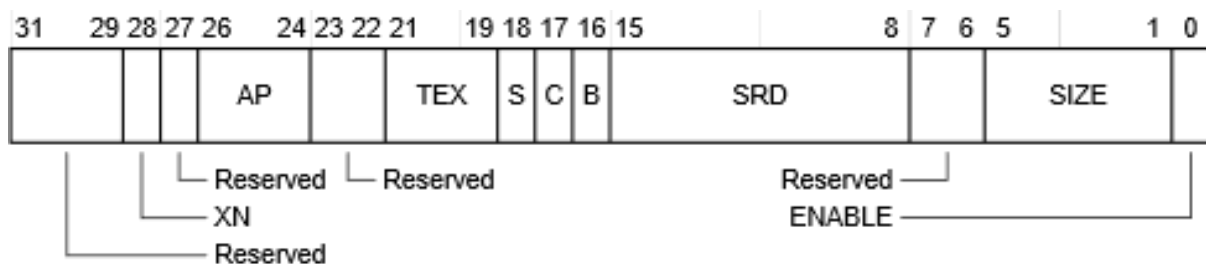


Figure 2.7: MPU REGION ATTRIBUTE ADDRESS REGISTER

The MPU RASR defines the region size and memory attributes of the MPU region

specified by the MPU RNR, and enables that region and any subregions. MPU RASR is accessible using word or halfword accesses:

- The most significant halfword holds the region attributes.
- The least significant halfword holds the region size and the region and subregion enable bits.

bit	field	Description
31:29	RESERVED	Reserved
28	XN	Instruction access disable. (1 = disable instruction fetch from this region; an attempt to do so will result in a memory management fault)
27	RESERVED	Reserved
26:24	AP	Access Permission
23:22	RESERVED	Reserved
21:19	TEX	Type Extension Field
18	S	Sharable
17	C	Cacheable
16	B	bufferable
15:8	SRD	Sub-region Disable
7:6	RESERVED	Reserved
5:1	VALID	MPU Protection Region size
0	REGION	Region enable.

Table 2.5: MPU Region Attribute Register

The subregion disable field (bit [15:8] of the MPU Region Base Attribute and Size register) is used to divide a region into eight equal subregions and then to define each as enabled or disabled. If a subregion is disabled and overlaps another region, the access rules for the other region are applied. If the subregion is disabled and does not overlap any other region, access to this memory range will result in a memory management fault. Subregions cannot be used if the region size is 128 bytes or less.

AP field([26:24]) describes the MPU access permission attributes. The access permission bits control access to the corresponding memory region. If an access is made to an

AP value	Privilege Access	User Access	Description
000	No Access	No Access	No Access
001	Read/write	No Access	Privileged access only
010	Read/write	Read only	Write in a user program generates a fault
011	No Access	Read/write	Full access
100	Unpredictable	Unpredictable	Unpredictable
101	Read only	No Access	Privileged read only
110	Read only	Read only	Read only
111	Read only	Read only	Read only

Table 2.6: Access Permission Configurations

area of memory without the required permissions, then the MPU generates fault.

TEX	C	B	Memory Type	Description	Sharable
000	0	0	Strongly Ordered	Strongly ordered	Yes
000	0	1	Device	Shared Device	Yes
000	1	0	Normal	Write through, no write allocate	S bit
000	1	1	Normal	Write back, no write allocate	S bit
001	0	0	Normal	Non-cacheable	S bit
001	0	1	Reserved	Reserved	Reserved
001	1	0	Undefined	Undefined	Undefined
001	1	1	Normal	Write back, write and read allocate	S bit
010	0	0	Device	Strongly ordered	No
010	0	1	Reserved	Reserved	Reserved

Table 2.7: Memory Attributes

- [S] indicates that shareability is determined by the S bit field (shared by multiple processors).

The S field is for a shareable memory region: the memory system provides data synchronization between bus masters in a system with multiple bus masters, for example, a

processor with a DMA controller. Strongly-ordered memory is always shareable. If multiple bus masters can access a non-shareable memory region, the software must ensure the data coherency between the bus masters. the S field is equivalent to non-cacheable memory.

The TEX, C and B bits are used to define cache properties for the region, and to some extent, its shareability. In v6 and v7 architecture, the memory system can have two cache levels: inner cache and outer cache. They can have different caching policies. Because the Cortex-M3 processor itself does not have a cache controller, the cache policy only affects write buffering in the internal BusMatrix and possibly the memory controller.

A microcontroller with cache memory, then you should program the MPU according to the cache policy you want to use (e.g., cache disable/write through cache/write back cache).

Memory Attribute encoding	Cache Policy
00	Non cacheable
01	Write back, write, and read allocate
10	Write through, no write allocate
11	Write back, no write allocate

Table 2.8: Inner and Outer Cache Policy

- Write through with no write allocate: on hits it writes to the cache and the main memory, on misses it updates the block in the main memory not bringing that block to the cache.
- Write-back with no write allocate: on hits it writes to the cache setting dirty bit for the block, the main memory is not updated. On misses it updates the block in the main memory not bringing that block to the cache.
- Write-back with write and read allocate: on hits it writes to the cache setting dirty bit for the block, the main memory is not updated. On misses it updates the block in the main memory and brings the block to the cache. [2]

2.4 Initializing and configuring an MPU

2.4.1 SETTING UP THE MPU

The MPU register might look complicated, but as long as you have a clear idea of the memory regions that are required for your application, it should not be difficult. Typically, you need to have the following memory regions:

The MPU register may look sophisticated, however as long as you've got a transparent plan of the memory regions that are needed for your application, it shouldn't be troublesome. Typically, you wish to own the subsequent memory regions: Program code for privileged programs (for example, OS kernel and exception handlers). Program code for user programs. Data memory for privileged and user programs in various memory regions (e.g., data and stack of the application situated in the SRAM (Static Random Access Memory) memory region-0x20000000 to 0x3FFFFFFF).] Other peripherals.

The MPU registers can be referenced from the MPU RASR register for each region. MPU RNR selects which region MPU RBAR and MPU RASR are currently configuring. The start and end address of each region can be programmed into the MPU RBAR and MPU RLAR registers, along with the required access permissions, shareability, and executability. When all the required regions have been configured the MPU can then be enabled by setting the ENABLE bit in MPU CTRL register. To ensure that any subsequent memory accesses use the new MPU configuration, software must execute a DMB followed by an ISB. When all the required regions have been configured, the MPU can then be enabled by setting the ENABLE bit in MPU CTRL. To ensure that any subsequent memory accesses use the new MPU configuration, software must execute a DMB followed by an Instruction Synchronization Barrier (ISB). If a memory location falls on 2 regions, the operation attributes and permission are going to be supported the highest-numbered region to be used.

2.4.2 pseudo code for MPU configuration

```
MPU ->RNR = 0; // select region
MPU ->RBAR = 0x00000000; // Base Address = 0x00000000
MPU ->RASR = 0x0307002F; // R/W, TEX=0,S=1,C=1,B=1, 16MB, Enable=1
MPU->RNR = 1; // select region 1
```

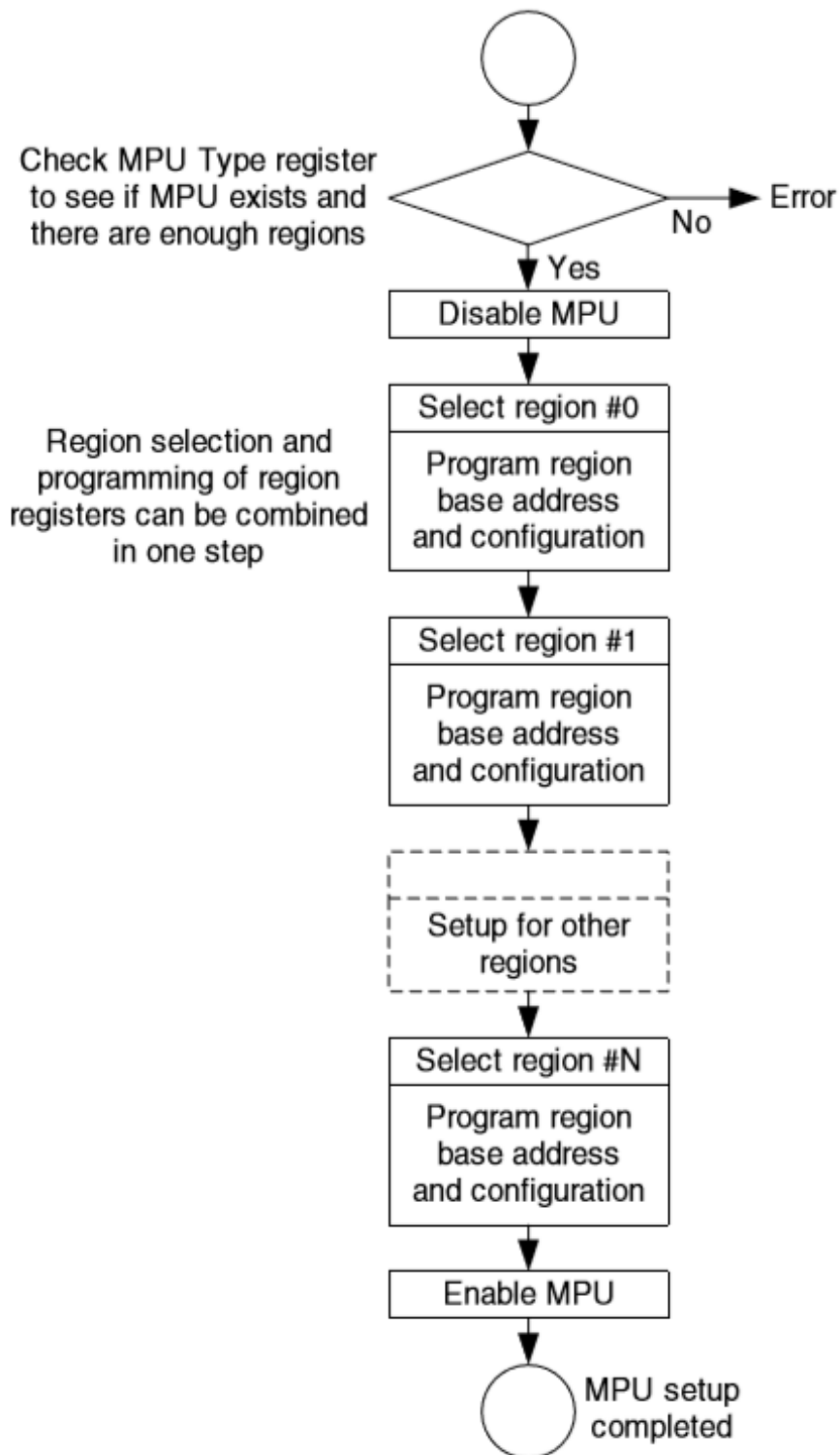


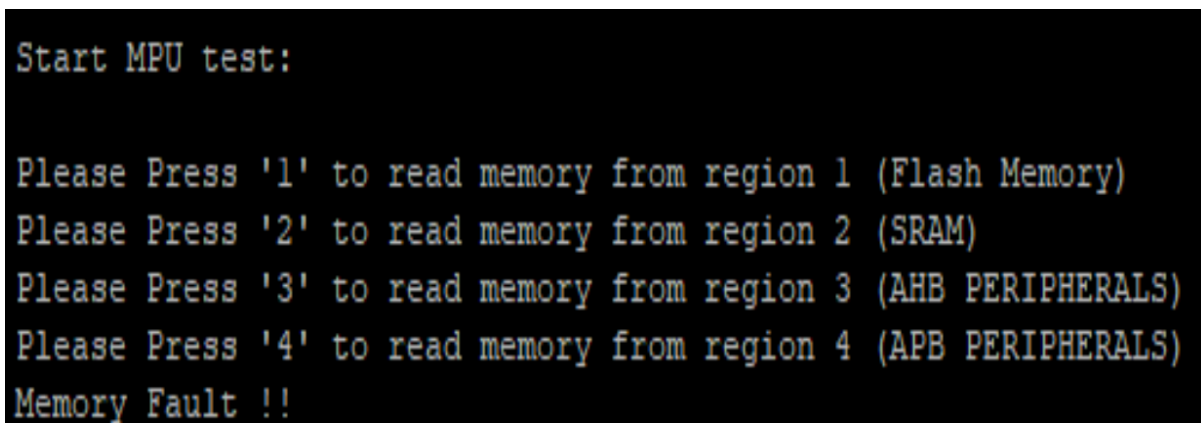
Figure 2.8: MPU REGION ATTRIBUTE ADDRESS REGISTER

MPU->RBAR = 0x20000000; // Base Address = 0x20000000

MPU->RASR = 0x03070033; // R/W, TEX=0,S=1,C=1,B=1, 64MB, Enable=1

```
MPU->RNR = 2; // select region 2
MPU->RBAR = 0x40000000; // Base Address = 0x40000000
MPU->RASR = 0x03050033; // R/W, TEX=0,S=1,C=0,B=1, 64MB, Enable=1
MPU->RNR = 3; // select region 3
MPU->RBAR = 0xA0000000; // Base Address = 0xA0000000
MPU->RASR = 0x01040027; // Pri R/W, TEX=0,S=1,C=0,B=0, 1MB, Enable=1
SCB ->SHCSR = 0x00000100; //Enable MemFault enable bit
MPU->CTRL = 5; // MPU Control register Enable MPU[3]
```

2.4.3 MPU OUTPUT

A terminal window with a black background and white text. The text displays the output of an MPU test, including instructions for reading memory from four regions and a memory fault message.

```
Start MPU test:

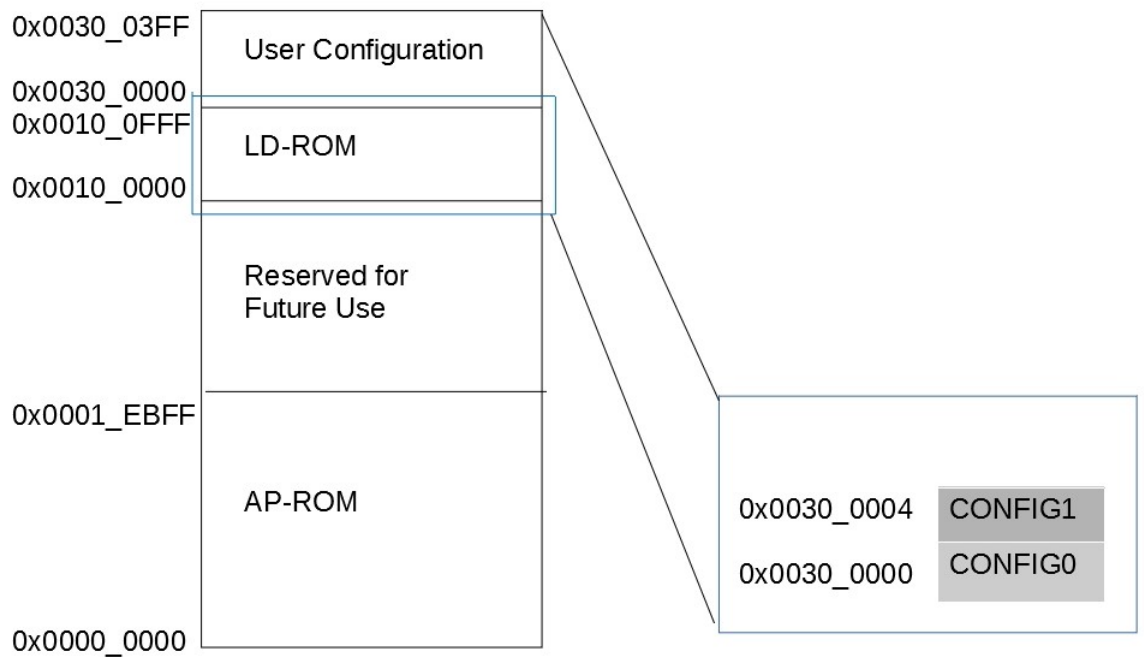
Please Press '1' to read memory from region 1 (Flash Memory)
Please Press '2' to read memory from region 2 (SRAM)
Please Press '3' to read memory from region 3 (AHB PERIPHERALS)
Please Press '4' to read memory from region 4 (APB PERIPHERALS)
Memory Fault !!
```

Figure 2.9: MPU OUTPUT

Chapter 3

Flash Memory

Flash memory is an electronic (solid-state) non-volatile computer storage medium that can be electrically erased and reprogrammed. The two main types of flash memory are named after the NAND and NOR logic gates.^[4]



memory.jpg

Figure 3.1: Flash Memory

The Nuvoton m487 supports 512Bytes.Flash memory divided into several parts.its including bootloader, OTP, IAP, KPROM etc. Communication between flash memory and other devices has been controlled by the flash memory controller (FMC). FMC man-

ages the data stored on flash memory and communicate with other devices and I/O. For each operation such as PROGRAM, READ,ERASE certain type of sequence is need to be generated to perform specific operation on flash memory.[4]

Flash Memory Features in Nuvoton:-

- Dual bank 512/256 KB on-chip Application ROM (APROM) for Over-The-Air (OTA) upgrade
- 192 MHz maximum frequency, with performance at zero wait cycle in continuous address read access.
- 4 KB on-chip Flash for user-defined loader (LDROM).
- 8 KB non-readable Key Protection ROM (KPROM) for firmware programming protection.
- 4 KB non-readable Security Protection ROM (SPROM) for intellectual property protection.
- 2 KB One Time Programable (OTP) ROM for data security.
- All on-chip Flash support 4 KB page erase.
- Fast Flash programming verification with CRC.
- On-chip Flash programming with In-Chip Programming (ICP), InSystem Programming (ISP) and In-Application Programming (IAP) capabilities.
- Configurable boot up sources including boot loader, user-defined loader (LDROM) or Application ROM (APROM).
- Data Flash with configurable memory size.
- 2-wired ICP Flash updating through SWD interface

3.1 Flash Memory Controller registers

Flash memory controller is interfaced with flash memory. It handles all signals required by the flash memory. With the required signals different types of commands are also

sent through flash memory controller to the flash memory. For each operation like PROGRAM, READ, ERASE sure form of sequence is ought to be generated to perform specific operation on non-volatile storage. These are the flash memory controller registers.[4]

Registers	Offset	Attribute	Description	Base Address
ISPCON	0x0	R/W	ISP CONTROL REGISTER	0x0000 0000
ISPADR	0x4	R/W	ISP ADDRESS REGISTER	0x0000 0000
ISPDAT	0x8	R/W	ISP CONTROL DATA	0x0000 0000
ISPCMD	0x0	R/W	ISP CONTROL COMMAND	0x0000 0000
ISPTRG	0x10	R/W	ISP CONTROL TRIGGER	0x0000 0000
ISPBASR	0x14	R	ISP CONTROL BASE ADDRESS	User Config1
ISPSTA	0x40	R/W	ISP CONTROL STATUS	0x0000 0000

Table 3.1: FMC Control Registers

These all are registers are used to configured flash memory. flash memory has been divided into several parts. If we want to configure a particular part of flash memory, we can identify that part or region by using base address or offset. In our case we want to configure the ISP (In system programming) OTP.[4]

3.2 OTP (One Time Program)

Some microprocessors provide a hardware-based one-time-programmable. (OTP) key generation function. OTP is One time programmable memory in flash memory. Sometimes it is also called fuse bit or EEPROM. Electrically programmable read-only memory is a one-time programmable, non-volatile memory. OTP (one time programmable) memory could be a special sort of non-volatile memory (NVM) that allows information to be written to memory just once.

In nuvoton PFM 487 board has a 3 Kbytes with the location address 0x310000 0x310BFF. The maximum size of OTP data is 2 Kbytes from 0x310000 0x3107FF. Every 64 bits of OTP data has one 32 bits LOCK BIT from 0x310800 to 0x310BFF. The purpose of LOCK BIT is for recording if the prograded address had been barred (LOCK BIT=0xFFFFFFFF) or not (LOCK BIT= 0xFFFFFFFF) in OTP. For example, when LOCK BIT0 is not FFFFFFFF, OPT0 cannot be programmed again, regardless its content is all 1 or not. The Flash page erase /mass erase command/ FLASH 64-bit Program/ FLASH Multi-

Offset	Fields	Description
[0x0000]	ISPCTL	ISP Control Register .
[0x0004]	ISPADDR	ISP Address Register .
[0x0008]	ISPDAT	ISP Data Register .
[0x000C]	ISPCMD	ISP Command Register .
[0x0010]	ISPTRG	ISP Trigger Register .
[0x0014]	ISPDFBA	ISP Data Flash Address Register .
[-]	RESERVE0 [10]	Reserved.
[0x0040]	ISPSTS	ISP Status Register .
[-]	RESERVE1 [2]	Reserved..
[0x004C]	CYCCTL	Flash Access Cycle Control Register .
[0x0050]	KPKEY0	KPROM KEY0 Data Register .
[0x0054]	KPKEY1	KPROM KEY1 Data Register .
[0x0058]	KPKEY2	KPROM KEY2 Data Register .
[0x005C]	KPKEYTRG	KPROM KEY Comparison Trigger Control Register .
[0x0060]	KPKEYSTS	KPROM KEY Comparison Status Register .
[0x0064]	KPKEYCNT	KPROM KEY-Unmatched Counting Register .
[0x0068]	KPCNT	KPROM KEY-Unmatched Power-On Counting Register.
[-]	RESERVE2 [5]	Reserved .
[0x0080]	MPDAT0	ISP Data0 Register.
[0x0084]	MPDAT1	ISP Data1 Register.
[0x0088]	MPDAT2	ISP Data2 Register.
[0x008C]	MPDAT3	ISP Data3 Register.
[-]	RESERVE3 [12]	Reserved.
[0x00C0]	MPSTS	ISP Multi-Program Status Register.
[0x00C4]	MPADDR	ISP Multi-Program Address Register .

Table 3.2: ISP Registers

Word Program are never allowed to be executed in OTP. Before updating the content of OTP from 0x310000 to 0x3107FF, users must check their LOCK BIT first. After finishing programming OTP data, users must write a non 0xFFFFFFFF in the LOCK BIT of the above programed data to make sure that no one can modify them; only make sure OTP

data cannot be changed. In some cases OTP data would be read. For example, when chip is in SCRLOCK or ARLOCK, it cannot be read by NON-SEC code. Then, do chip erase SCRLOCK and ARLOCK would be unlock, and OTP data would not be erased. OTP is in ISP. OTP Registers OTP can be programmed through ISP registers. We have to program the registers in OTP. Base address of OTP is starting from 0x310000. We can write data in OTP but we have to write data in 64 bit chunks. And 32 bit lock address. IN nuvoton microcontroller has an embedded flash and OTP size is 2KB. We can consider that 64 bit data is equal to one page so we can program up to 256 pages.[4]

3.2.1 ISP Control Register



Figure 3.2: ISP CONTROL REGISTER

Bits	Command	Description
[31:7]	-	Reserved
[6]	ISPPF	ISP Fail Flag
[5]	LDUEN	LDROM Update Enable
[4]	CFGUEN	Enable Config-bits Update by ISP
[3]	APUEN	APROM Update Enable
[2]	-	Reserved
[1]	BS	Boot Select (1:LDROM; 0:APROM)
[0]	ISPEN	ISP Enable

Table 3.3: ISP Control Register

3.2.2 ISP Address Register



Figure 3.3: ISP Address Register

Bits	Command	Description
[31:0]	ISPADDR	ISP erase program read Address

Table 3.4: ISP Address Register

3.2.3 ISP Data Register



Figure 3.4: ISP Data Register

[31:0]	ISPDATA	ISP program/read data word.
--------	---------	-----------------------------

Table 3.5: ISP Data Register

3.2.4 ISP Command Register



Figure 3.5: ISP Command Register

Here, in command register last 6 bits are used for different operation. By changing the value of last six bits we can configure flash read, write, verification and many other operation. And last three bits are used for flash control bits. It also divided into FC-TRL[3:0], FCEN[4], FOEN[5].

[31:6]	Reserved
[5]	FOEN
[4]	FCEN
[3:0]	FCTRL

Table 3.6: ISP Command Register Bits

Bits	Command	Description
[31:7]	CMD	Reserved
[6:0]		<p>ISP Command</p> <p>0x00= FLASH Read.</p> <p>0x04= Read Unique ID.</p> <p>0x08= Read Flash All-One Result.</p> <p>0x0B= Read Company ID</p> <p>0x0C= Read Device ID</p> <p>0x0D= Read Checksum.</p> <p>0x21= FLASH 32-bit Program..</p> <p>0x22= FLASH Page Erase. Erase any page in two banks, except for OTP.</p> <p>0x23= FLASH Bank Erase. Erase all pages of APROM in BANK0 or BANK1.</p> <p>0x25= FLASH Block Erase. Erase four pages alignment of APROM in BANK0 or BANK1.</p> <p>0x27= FLASH Multi-Word Program.</p> <p>0x28= Run Flash All-One Verification.</p> <p>0x2D= Run Checksum Calculation..</p> <p>0x2E= Vector Remap.</p> <p>0x40= FLASH 64-bit Read..</p> <p>0x61= FLASH 64-bit Program..</p>

Table 3.7: ISP Command Register

3.2.5 ISP Trigger Register



Figure 3.6: ISP Trigger Register

Bits	Command	Description
[31:1]	-	Reserved
[0]	ISPGO	ISP start trigger .

Table 3.8: ISP Trigger Register

3.2.6 ISP Data Flash Address Register



Figure 3.7: ISP Data Flash Address Register

[31:0]	DFBA	ISP Data Flash Base Address
--------	------	-----------------------------

Table 3.9: ISP Data Flash Register

This register indicates Data Flash start address. It is a read only register. The Data Flash is shared with APROM. the content of this register is loaded from CONFIG1. This register is valid when DFEN (CONFIG0[0]) =0 .

3.2.7 ISP Status Register



Figure 3.8: ISP Status Register

Bits	Command	Description
[31:21]	-	Reserved
[20:9]	VECMAP	Vector Page Mapping Address
[8:7]	-	LDROM Update Enable
[6]	-	Reserved
[5:4]	ISPFF	ISP Fail flag
[3]	-	Reserved
[2:1]	CBS	Boot Selection Status
[0]	ISPBUSY	ISP BUSY

Table 3.10: ISP Status Register

3.3 OTP Programming Flow chart

The FMC controller provides embedded Flash memory read, erase and program operation. Several control bits of FMC control register are write-protected, thus it is necessary to unlock before setting. After unlocking the protected register bits, user needs to set the FMC ISPCTL control register to decide to update LDROM, APROM or user configuration block, and then set ISPEN (FMC ISPCTL[0]) to enable ISP function. Note that, when FMC is doing ISP command user cannot enter any power down mode.

Once the FMC ISPCTL register is set properly, user can set FMC ISPCMD 40; refer to Table 6.7-3 ISP command list 41; for specify operation. Set FMC ISPADDR for target Flash memory based on Flash memory organization. FMC ISPDAT can be used to set the data to program or used to return the read data according to FMC ISPCMD. Finally, set the ISPGO (FMC ISPTRG[0]) register to perform the relative ISP function. When ISP

function is active, the ISPBUSY(FMC ISPSTS[0]) and MPBUSY(FMC MPSTS[0]) be set to 1. The ISPGO(FMC ISPTRG[0]), ISPBUSY(FMC ISPSTS[0]) and MPBUSY(FMC MPSTS[0]) are selfcleared when ISP function has been done.

Several error conditions area unit about to be checked once ISP is completed. If a blunder condition happens, ISP operation isn't started and therefore the ISP fail flag are set instead. ISPPFF(FMC ISPSTS[6]) flag will solely be cleared by software system. ensuing ISP procedure are often started even ISPPFF(FMC ISPSTS[6]) bit is kept as 1. Therefore, it's suggested to examine the ISPPFF(FMC ISPSTS[6]) bit and clear it once each ISP operation if it's set to 1.[5]

When the ISPGO(FMC ISPTRG[0]) bit is ready so C.P.U. access an equivalent bank, C.P.U. can watch for ISP operation to finish throughout this period; the peripheral still keeps in operation as was common. If any interrupt request happens, constituent will not service it till ISP operation is finished. once ISP operation is finished, the ISPGO bit are cleared by hardware mechanically. User will check whether or not ISP operation is finished or not by the ISPGO(FMC ISPTRG[0]) bit. When C.P.U. access operation and ISP command area unit dead in numerous bank, C.P.U. and ISP command will operate in parallel.[5]

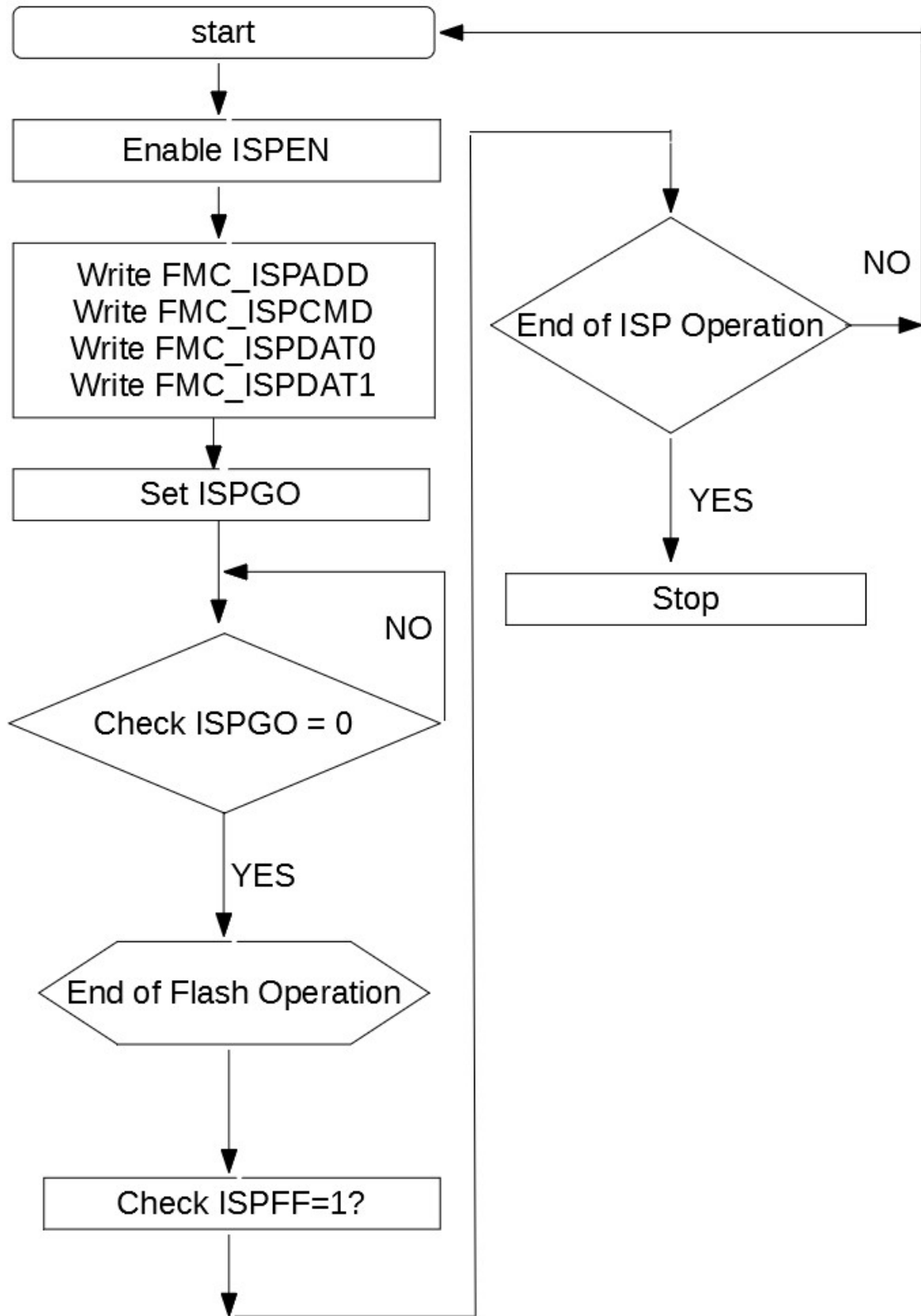


Figure 3.9: ISP program

Chapter 4

Cryptography

4.1 Definition and History of cryptography

Cryptography or cryptology is that the follow and study of techniques for secure communication. More usually, cryptography is concerning constructing and analyzing protocols that forestall third parties or the overall public from reading personal messages; varied aspects in information security like data confidentiality, data integrity, authentication, and non-repudiation area unit central to trendy cryptography. trendy cryptography exists at the intersection of the disciplines of arithmetic, engineering, technology, communication science, and physics. Applications of cryptography embrace electronic commerce, chip-based payment cards, digital currencies, laptop computer passwords, and military communications.

The cryptography literature typically uses the names Alice ("A") for the sender, Bob ("B") for the supposed recipient. Modern cryptography is heavily supported arithmetic theory and algorithms.

cryptology referred nearly alone to cryptology, that's that the tactic of fixing traditional information (called plain text) into unintelligible sort (called). decoding is that the reverse, in alternative words, moving from the unintelligible cipher text back to plain text. A cipher conjointly be—is also a attempt of algorithms that build the cryptology and also the reversing secret writing. the flowery operation of a cipher is controlled every by the rule and in each instance by a "key". The secret's a secret (ideally noted alone to the communicants), generally a quick string of characters, that's needed to rewrite the cipher text. Formally, a "cryptosystem" is that the ordered list of compo-

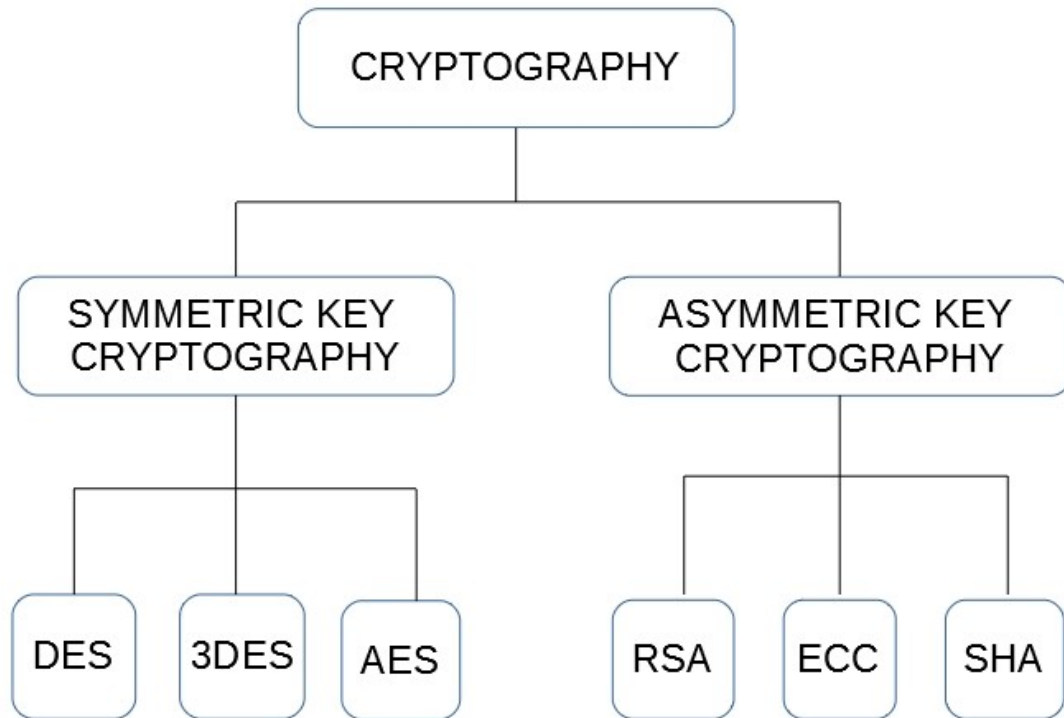


Figure 4.1: Cryptography

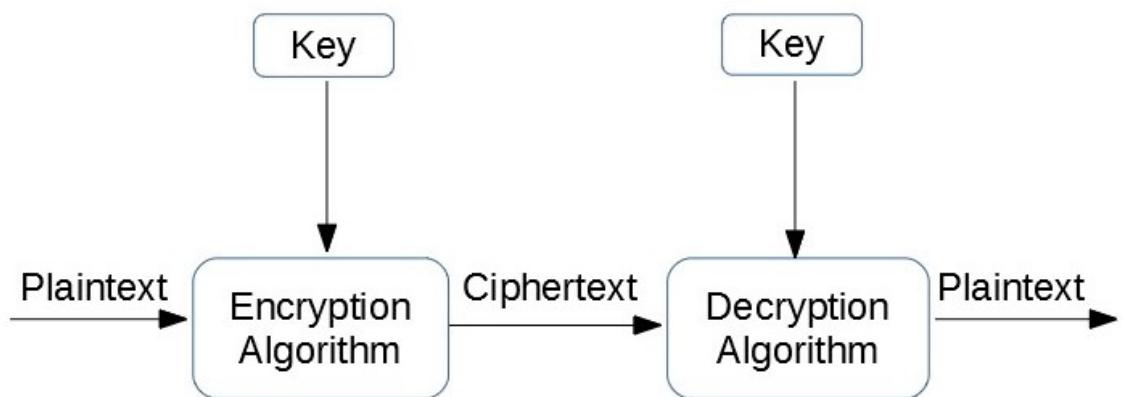
nents of finite doable plain texts, finite doable cyphertexts, finite doable keys and also the coding and decoding algorithmic rule that correspond to every key. Keys area unit vital each formally and in actual observe, as ciphers while not variable keys will be trivially broken with solely the data of the cipher user area unit thus useless for many functions..[6]

4.2 classification of cryptography

Before the fashionable era, cryptography targeted on message confidentiality (i.e., encryption)conversion of messages from a visible kind into Associate in Nursing incomprehensible one and back all over again at the opposite finish, rendering it indecipherable by interceptors or eavesdroppers whereas not secret knowledge (namely the key needed for secret writing of that message). secret writing tried to verify secrecy in communications, like those of spies, military leaders, and diplomats. In recent decades, the world has distended on the so much aspect confidentiality concerns to include techniques for message integrity checking, sender/receiver identity authentication, digital signatures, interactive proofs and secure computation, among others.[7]

1. Symmetric Key Cryptography
2. Asymmetric Key Cryptography.

1. Symmetric Key Cryptography :- The art of secret writing. Symmetric-key algorithms for cryptography that use a similar cryptographical keys for each encoding of plain text and decipherment of . The keys is also identical or there is also a straightforward transformation to travel between the 2 keys. The keys, in observe, represent a shared secret between 2 or additional parties which will be wont to maintain a personal info link.



key.jpg

Figure 4.2: Symmetric Cryptography

Symmetric-key encryption can use either stream ciphers or block ciphers.

- Stream ciphers cypher the digits (typically bytes), or letters (in substitution ciphers) of a message one at a time. associate example is that the Vigenere Cipher.
- Block ciphers take variety of bits and code them as one unit, artefact the plain text in order that it's a multiple of the block size. Blocks of sixty four bits were normally used. The Advanced encoding customary (AES) formula approved by NIST in Dec 2001, and also the GCM block cipher mode of operation use 128-bit blocks.

Symmetric ciphers ar typically accustomed succeed different cryptographic primitives than merely cryptography. Symmetric-key algorithms want every the sender and additionally the recipient of a message to have identical secret key. All early cryptographic systems required one in all those of us to somehow receive a reproduction of that secret key over a physically secure channel. Nearly all stylish

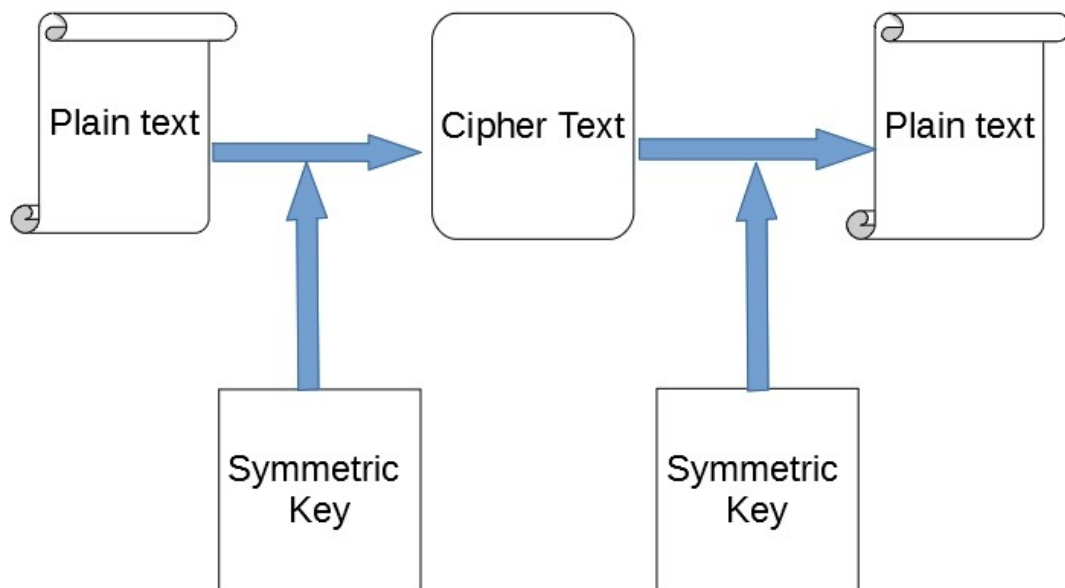


Figure 4.3: AES Algorithm

cryptographic systems still use symmetric-key algorithms internally to put in writing the bulk of the messages, but they eliminate the need for a physically secure channel. The sender and additionally the recipient have to be compelled to perceive the key key that is accustomed write and decipher all the messages. Blowfish, AES, RC4, DES, RC5, and RC6 ar samples of symmetric cryptography. the foremost wide used symmetric formula is AES-128, AES-192, and AES-256. we have a tendency to ar victimization AES key commonplace rule for the info cryptography.

4.2.1 AES(Advance Encryption Standard)

1. AES Key Description and its features.

The Advanced Encryption Standard (AES), also known by its original name Rijndael is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.For AES, NIST selected three members of the Rijndael family, each with a block size of 128 bits, but three different key lengths: 128, 192 and 256 bits.[7]

The features of AES are as follows

- Symmetric key symmetric block cipher.

- 128-bit data, 128/192/256-bit keys.
 - Stronger and faster than Triple-DES.
 - Provide full specification and design details.
1. High-Level Description of the algorithm
 - (I) KeyExpansionround keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.
 - (II) AddRoundKeyeach byte of the state is combined with a block of the round key using bitwise xor.
 - (III)
 - A. SubBytes : A non-linear substitution step where each byte is replaced with another according to a lookup table.
 - B. ShiftRows : A transposition step where the last three rows of the state are shifted cyclically a certain number of steps.
 - C. MixColumns : A linear mixing operation which operates on the columns of the state, combining the four bytes in each column.
 - D. AddRoundKey
 - (IV)
 - A.Subbyte.
 - B.ShiftRows.
 - C.AddRoundKey.

2. Asymmetric Key Cryptography :-

Public-key cryptography, or asymmetric cryptography, is a cryptographic system that uses pairs of keys: public keys which may be disseminated widely, and private keys which are known only to the owner. In such a system, a person will code a message exploitation the receiver's public key, however that encrypted message will solely be decrypted with the receiver's non-public key. A sender will mix a message with a non-public key to make a brief digital signature on the message. Anyone with the corresponding public key will mix a message, a reputed digital signature thereon, and the known public key to verify whether the signature was valid, i.e. made by the owner of the corresponding non-public key.

There are two best-known uses of public key cryptography:

- Public key encoding, during which a message is encrypted with a recipient's public key. The message can't be decrypted by anyone United Nations agency doesn't possess the matching non-public key, United Nations agency is so probable to be the owner of that key and also the person associated with the public key. This is utilized in an endeavor to confirm confidentiality.
- Digital signatures, during which a message is signed with the sender's non-public key and may be verified by anyone United Nations agency has access to the sender's public key. This verification proves that the sender had access to the non-public key, and thus is probably going to be the person related to the general public key. This additionally ensures that the message has not been tampered with, as a signature is mathematically guaranteed to the message it originally was created with, and verification will fail for much the other message, notwithstanding however almost like the first message.^[7]

One necessary issue is confidence/proof that a specific public key's authentic, i.e. that it is correct and belongs to the person or entity claimed, and has not been tampered with or replaced by a malicious third party. There are several possible approaches, including: A public key infrastructure (PKI), in which one or more third parties known as certificate authorities certify ownership of key pairs. TLS relies upon this.

A "web of trust" that decentralizes authentication by mistreatment individual endorsements of the link between user and public key. PGP uses this approach. Lookup in the domain name system (DNS). The DKIM system for digitally signing emails uses this approach. The most obvious application of a public key coding system is in encrypting communication to produce confidentiality a message that a sender encrypts mistreatment the recipient's public key can be decrypted solely by the recipient's paired non-public key. Another application publicly key cryptography is that the digital signature. Digital signature schemes are often used for sender authentication.

Non-repudiation system use digital signatures to confirm that one party cannot with success dispute its authorship of a document or communication.

Further applications designed on this foundation include: digital money, password-authenticated

key agreement, time-stamping services, non-repudiation protocols, etc.

Because uneven key algorithms square measure nearly continually far more computation-ally intensive than bilaterally symmetrical ones, in many cases it is common to exchange a key using a key-exchange algorithm, then transmit data mistreatment that key and a bilaterally symmetrical key algorithmic program. PGP, SSH, and therefore the SSL/TLS family of schemes use this procedure, and square measure so referred to as hybrid crypto systems.[7]

4.3 OUTPUT

```
|      FMC OTP      |
0x0000  e9 17 82 53 cc 49 c6 be 6d 3f 79 d1 91 71 3a a5 .
0x0010  41 91 cb b4 55 ab f2 30 68 30 35 68 c9 39 fb 2e A

OTP116 is not locked. It should be a free entry.
Read back OTP116:0xe9178253-0xcc49c6be.
OTP key write done
Read back OTP117:0x6d3f79d1-0x91713aa5.
OTP key write done
Read back OTP118:0x4191cbb4-0x55abf230.
OTP key write done
Read back OTP119:0x68303568-0xc939fb2e.
OTP key write done
AES encrypt done.

0x0000  3d ec 68 4b 2d f3 98 45 08 33 48 c9 9b a0 ff 50
0x0010  7e a1 7a 8b 1b 5d 70 90 53 58 d2 4d 3f dd c0 1d

AES decrypt done.

0x0000  e9 17 82 53 cc 49 c6 be 6d 3f 79 d1 91 71 3a a5
0x0010  41 91 cb b4 55 ab f2 30 68 30 35 68 c9 39 fb 2e
```

Figure 4.4: OUTPUT

4.4 Use Cases

In recent years computer code update become essential for embedded system in electronic devices. there's a considerable increase in devices victimisation IOT with wired or wired network. The computer code updation has to be updated fairly often for development of devices. Embedded devices got to be updated firmly. A malicious, unauthorized and corrupted computer code update within the devices could cause severe malfunction. once the computer code valid and located authentic for installation. Firmware hold on into the non-volatile storage that is scan solely. Some controller has Associate in Nursing elective programmable memory is named the memory protection unit. MPU may be programmed victimisation totally different MPU register. Flash and SRAM forever unbroken as a read-only mode. Configuring MPU, non-volatile storage forever be in read-only mode. however some portion of non-volatile storage unbroken as a user configuration by change MPU. Therefore, we will store important information in OTP(one time programmable) memory. And victimisation this OTP we will store security keys, certificate, and digital signature for supportive information. Purpose to did this, once the computer code update comes victimisation FOTA, the Key hold on in OTP checks certificate or signature verification or some encrypted computer code comes than it may be decrypted victimisation this encoding key. Once the computer code is valid and located authentic for installation, it'll write the computer code to flash firmly. Once computer code is downloaded the computer code is checks authentication or corruption. Signature and certificate that is hold on in OTP check the computer code corruption. If the computer code is corrupted, it's not match the signature or certificate and computer code update results in fail.

(2) :-Results The processor or controller features a memory. MPU is element for memory protection. It divides memory into many variety of regions.MPU enforced in ARMv7 design follows protected memory system design model. Every region is split into sub region. It defines the poignant vary of the access. The MPU memory map is unified, means that instruction access and information access have a similar region attribute. The region attribute and size register defines the memory attribute of access. The AP bits are information access permission field, it is wont to started a privileged system wherever a number of the memory regions aren't allowed for user. IN nuvoton m487 board features a 3 on Board Leds. by victimisation MPU configuration and AP bits is about for the

kernel R/W access and user no access. once user tries to put in writing to show on LED it offers memory fault as a result of we tend to set the region as a user has no access and user tried to access LED. Same as we tend to has totally different peipheral GSM atatch to the board and tried to access GSM. causation AT command in GSM we tend to check the GSM peripheral region is lock or not. we tend to set the attribute no access. means that user or kernel browse and write from that region it offers memory or laborious fault. and that we set the attribute as a Kernel browse Write and user browse write it offers access to the GSM.

Chapter 5

SUMMARY

The Cortex-M4 has MPU. The purpose of the MPU is to protect the memory regions like flash, SRAM from illegal accessing by some other programs and devices. MPU can be configured using MPU registers. We can configured upto 8 number of regions. In MPU RASR register has a Access Permission bit which can be set for different purposes like kernel or user access R/W, Read only and no access. There is a Execute Never bit also in MPU RASR register. If this bit is set 1 and any access from that region gives memory fault. we can divide region into 8 sub region. Critical data stores in flash memory. flash memory data can interact using a flash memory controller to other peripherals. Furthermore, flash memory can be divided into several parts like OTP, bootloader, XOM. Flash memory kept always in read only mode. We can program OTP carefully under software control. A 3 Kbytes one-time-program ROM (OTP) is used for recording one-time-program data. 2Kb data size and 1KB is lock bit for OTP. We have to programmed OTP very carefully because once we write data in OTP, it can not be changed or modified. We can read and write in OTP in 64 bit chunks. Keys can be stored in OTP. We are stored certificate into OTP which is useful for firmware updation. Using crypto acceleration tool in nuvoton we can encrypt and decrypt data. We can store cryptography keys, certificates or digital signatures in One Time Programmable memory. Now when firmware updating request comes, it checks cryptography keys which are stored in OTP, if keys, digital signature matches, the firmware updating processes further.

Bibliography

- [1] https://static.docs.arm.com/100699/0100/armv8m_architecture_memory_protection_unit_100699_0100_00_en.pdf.
- [2] http://centaur.sch.bme.hu/~holcsik_t/sem/The%20Definitive%20Guide%20to%20the%20ARM%20Cortex-M3.pdf.
- [3] https://web.eecs.umich.edu/~prabal/teaching/eecs373-f10/readings/ARMv7-M_ARM.pdf/.
- [4] http://www.nuvoton.com/resource-files/DS_M2351_Series_EN_Rev1.01.pdf.
- [5] http://www.nuvoton.com/resource-files/DS_M480_Series_EN_Rev1.00.pdf.
- [6] <https://en.wikipedia.org/wiki/Cryptography>.
- [7] <https://www.geeksforgeeks.org/cryptography-introduction-to-crypto-terminologies/>.