

Tunku Abdul Rahman University College

Microcontroller Peripherals

***Flash Memory***

|  |  |  |
| --- | --- | --- |
| **Name:** | Wong Jang Sing |  |
|  |  |  |
| **Reg. No** | 15 WAR 09371 |  |
|  |  |  |
| **Programme:** | RMB 2 |  |
|  |  |  |
| **Tutor:** | Dr. Poh Tze Ven |  |
|  |  |  |
|  |  |  |

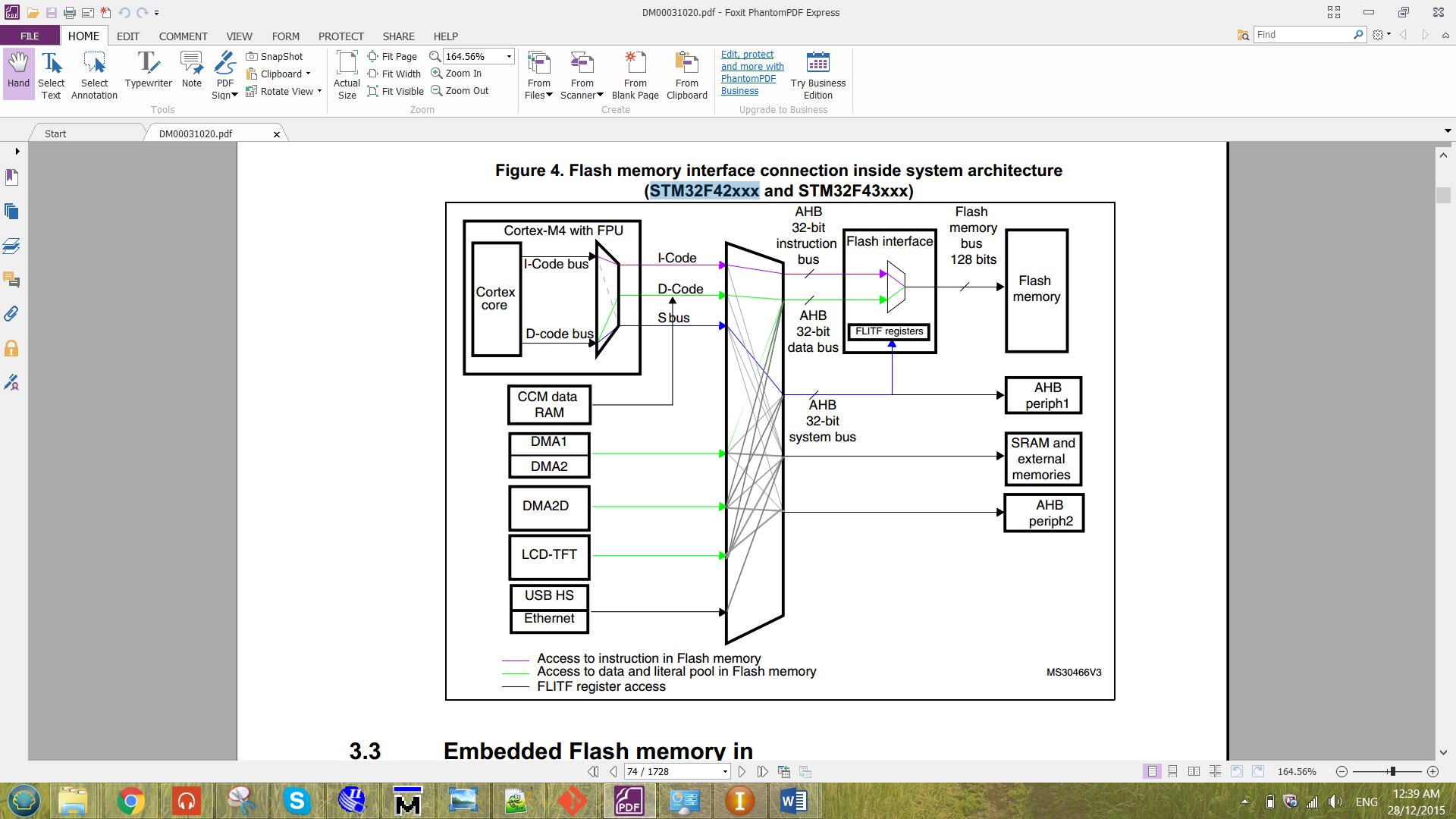
# CHAPTER I: INTRODUCTION

**1.1 Objective:**

Program has been written to the ARM Cortex®-M4 Microcontroller to communicate with the Embedded Flash memory interface to either perform read/program/erase operation to the Flash Memory. Result has been obtained and discussed in the following chapters.

**1.2 Brief Description:**

The Embedded Flash memory interface is an intermediate agent between the Cortex-M4 microcontroller and the Flash memory. Basically the microcontroller is telling the flash memory interface whether to perform read/program/erase operation to the Flash Memory. The Flash memory interface connection inside system architecture for STM32F429 is as shown below.

  
Figure 1.1 Flash memory interface connection inside system architecture for   
 STM32F429

The Flash memory interface manages CPU AHB (advanced high-performance bus) I-Code and D-Code accesses to the Flash memory. The I-Code and D-Code are Instruction Code and the Data Code respectively. For the I-Code, it connects the Instruction bus of the ARM microcontroller to the Flash instruction interface. While for the D-Code, it connects the D-Code bus of the ARM microcontroller to the Flash data interface. Before the flash memory interface, the data has been processed by the bus matrix. The bus matrix is as shown below.

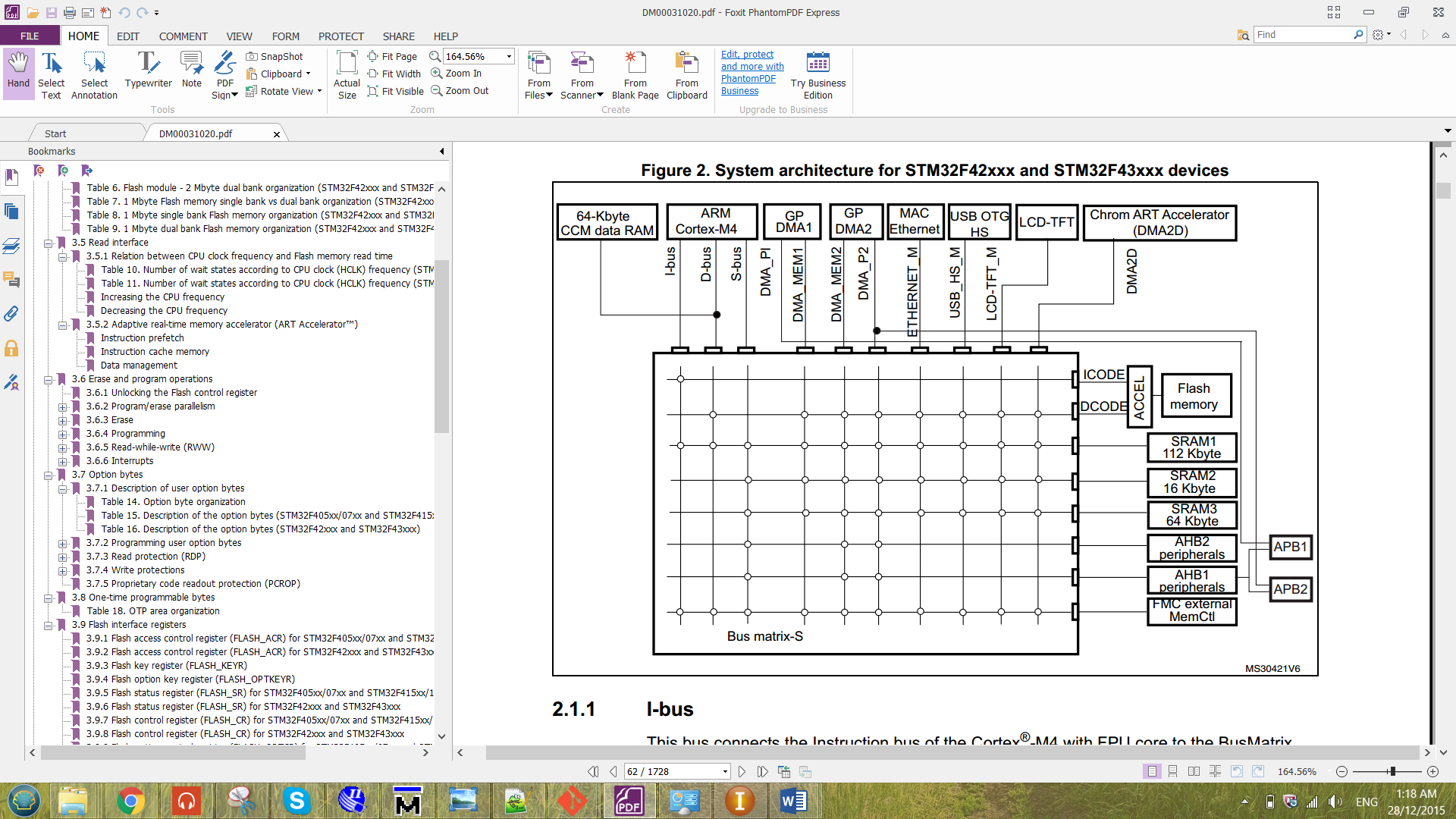


Figure 1.2 Bus matrix for STM32F429.

The flash memory for microcontroller STM32F429 is organized as follow:

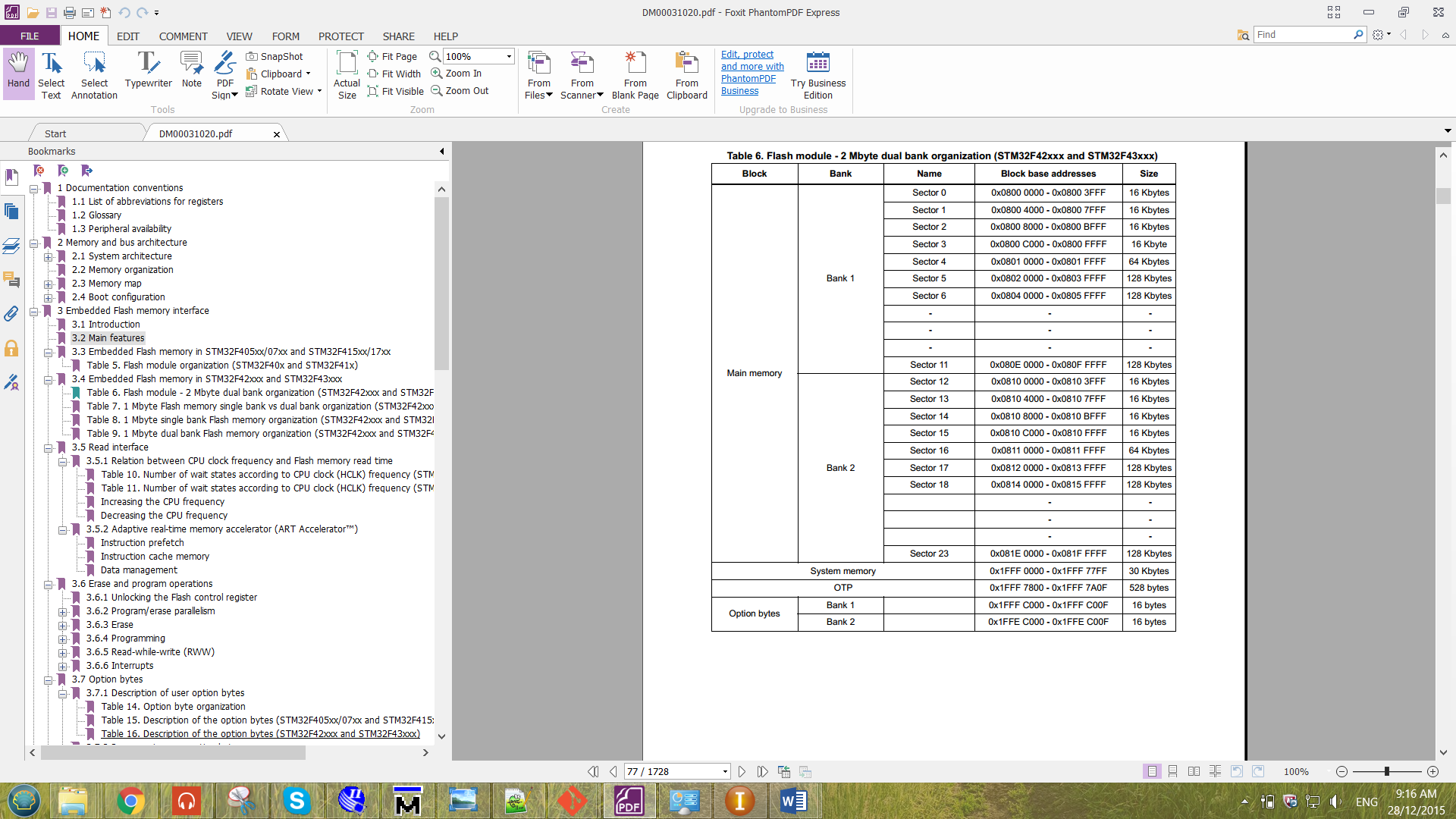
- A main memory block divided into sectors.

- System memory

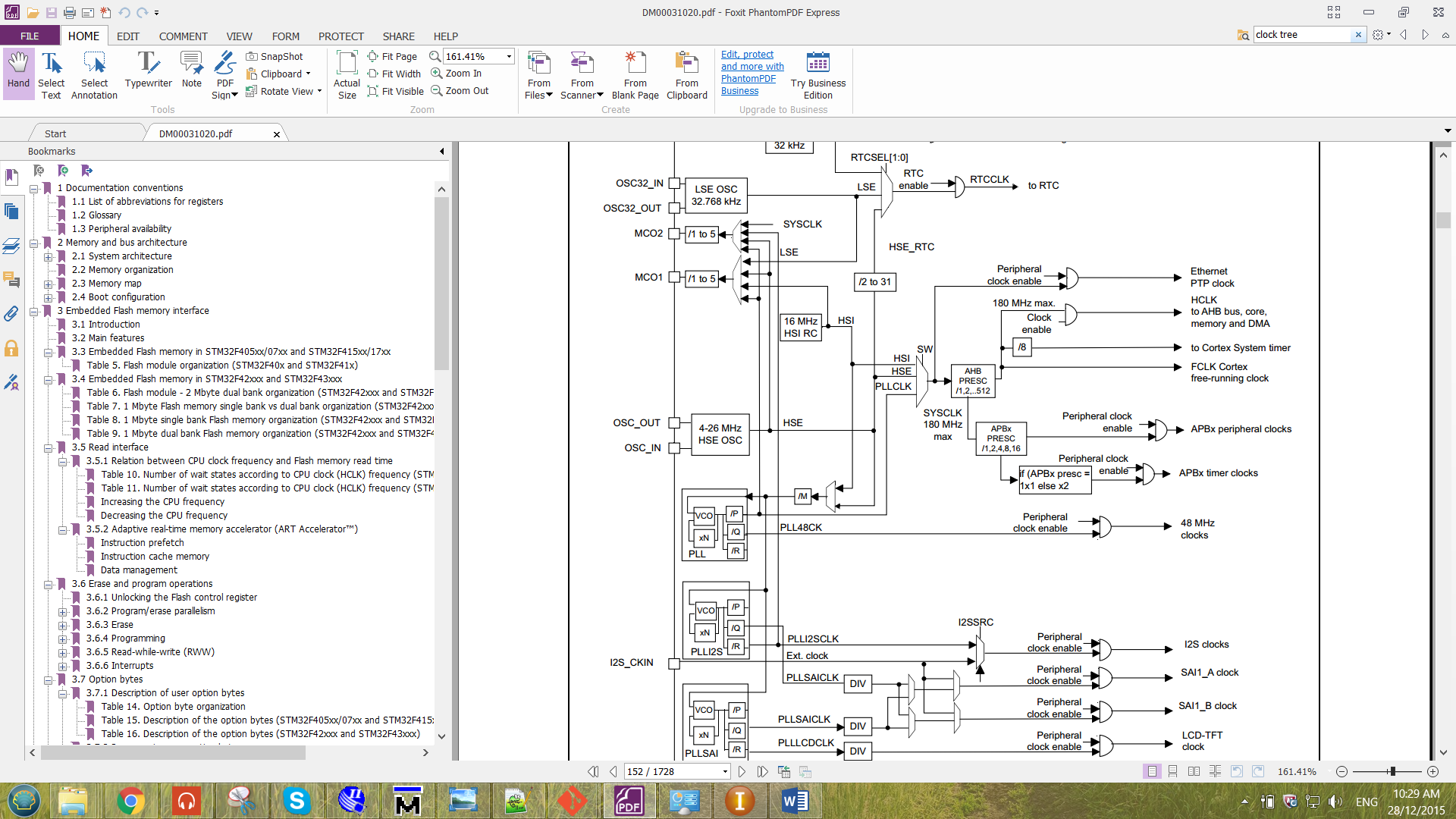
- 512 OTP (one-time programmable) bytes for user data.

- Option bytes.

The main memory block was divided into two bank with 1Mbyte. In each bank, it consists of 4 sectors with 16k bytes, 1 sector of 64k bytes and 7 sectors of 128k bytes. There are 16 additional byte in each bank which is the Option Bytes. The option bytes is to configure the read and write protection, BOR level, watchdog, dual bank boot mode, dual bank feature, software/hardware and reset when the device is in Standby or Stop mode. The STM32F429 is a device that have 2M byte with dual bank flash memory. The flash module for 2M byte dual bank organization is as shown below.

Table 1.1 flash module for 2M byte dual bank organization for STM32F429.

The clock used for the Flash Memory is the HCLK, which can operate up to 180MHz maximum. The clock source used to produce the HCLK is from the external clock (CRYSTAL CLOCK). The crystal clock frequency of the device is 8MHz. The overall picture the HCLK that produced to the flash memory is as shown below.

  
Figure 1.2 Clock tree for the Flash Memory.

From the figure 1.1, there are 32-bit for instruction bus and 32-bit for data bus. So the maximum bit that able to store into the Flash is 32-bit. The Flash is unable to store a 64-bit value. These will be proven in the following chapter.

**1.3 Flash memory Operation**

**i)** **Erase Operation**

There are three type of erase operation for the flash memory, which are sector erase, bank erase and Mass erase. After the erase operation has been done, the data in the erased address will all become ‘1’.

Sector Erase => to erase all the data in the selected sector.

Bank Erase => to erase all the data in the selected bank.

Mass Erase => to erase all the data in the Flash Memory.

**ii) Program Operation**

Before programming data into an address of the flash, the particular address has to be erased only it is ready to program. Otherwise, unexpected value will be stored to the particular address selected. This happens because the flash memory of this device are only able to write a ‘0’ to the address. When they see a ‘1’, it will not make any changes to the memory data. Thus, violation may happens.

# CHAPTER II: METHODOLOGY

In this chapter will discuss about the configuration of the flash memory. Before performing any operation to the Flash memory interface, the flash control register (FLASH\_CR) has to be unlocked by storing the key value into the flash key register (FLASH\_KEYR). There are two key that has to be keyed in into the register, the sequence is as shown below.

KEY1 = 0x45670123

KEY2 = 0xCDEF89AB

The KEY1 has to be keyed into the register only followed by the KEY2. Otherwise the Flash control register will still not be unlocked.

**Erase Operation**

1. Check whether the Flash memory interface is busy by checking the BSY bit in the Flash status Register (FLASH\_SR).
2. **For Sector Erase,**

* Setting the number of SNB bit to choose which sector to erase.

**For Bank Erase,**

* Setting MER or MER1 bit to choose which bank to erase. MER for bank1 and MER1 for bank2.

**For Mass Erase,**

* By setting MER and MER1 bit to erase both bank.

1. Set the STRT bit in FLASH\_CR register to start erasing.
2. Wait BSY bit to be cleared.

**Programming Operation**

1. Check whether the Flash memory interface is busy by checking the BSY bit in the Flash status Register (FLASH\_SR).
2. Setting PG bit in FLASH\_CR to enable programming.
3. Set parallelism (x8, x16, x32 and x64).
4. Wait BSY bit to be cleared.

# CHAPTER III: RESULT