ELEC 3300 – Tutorial 1A

Department of Electronic and Computer Engineering
HKUST
by WU Chi Hang 

Number System

- In **positional notation,** $a_n a_{n-1} a_{n-2} \ldots a_1 a_0 . a_{-1} a_{-2} \ldots a_{-n}$
  base *b* number equals to

$$a_n \times b^n + a_{n-1} \times b^{n-1} + a_{n-2} \times b^{n-2} + \ldots + a_1 \times b^1 + a_0 \times b^0 + a_{-1} \times b^{-1} + a_{-2} \times b^{-2} + \ldots + a_{-n} \times b^{-n}$$

  for hexadecimal, *b* = 16,
  for decimal *b* = 10,
  for octal *b* = 8,
  and for binary, *b* = 2

Number System

a is a number that between 0 to *b* – 1.

- for binary $a \in (0,1)$
- for octal $a \in (0,1,2,3,4,5,6,7)$
- for decimal $a \in (0,1,2,3,4,5,6,7,8,9)$
- for hexadecimal $a \in (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)$

Decimal-to-binary conversion, you successive division by 2.

How can you do the binary to octal notation ?

How can you do the binary to hexadecimal notation ?

Powers of 2

- In this course, it is better for you to familiar with the powers of 2. You should change your mind in binary ☺
- Try to fill the table

	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{16}	2^{20}	2^{32}
Dec	1	2	4	8	16									
Bin														
Hex														

- Please note that in this course, when we talk about address, and memory storage.
  - $1k = 2^{10} = 1024 \rightarrow 10$ bits,
  - $1M = 2^{20} = 2^{10} \times 2^{10} = 1k \times 1k = 1{,}048{,}576 \rightarrow 20$ bits
  - $1G = 2^{30} = 2^{10} \times 2^{10} \times 2^{10} = 1k \times 1k \times 1k = 1{,}073{,}741{,}824 \rightarrow 30$ bits

Number

- If I give you 3 numbers, in decimal format
  - 1, 60, 3600
- Are these 3 numbers different ?
- Can they represent the same thing ?
- If I give you another 3 numbers
  - 100_{10}, 64_{16}, 144_{8} ,
  - Are the values of 3 numbers different ?
  - Is their representation different ?

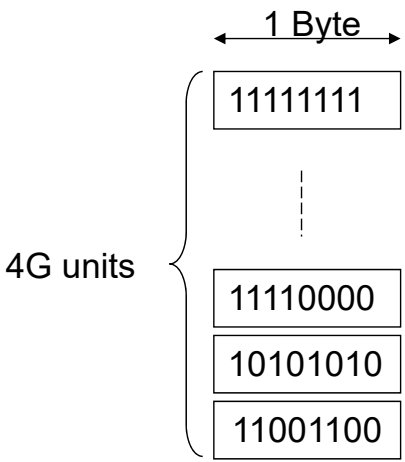
Number

- If I give you the following numbers in decimal ..
  - 20463398, 97532148, 46709394, 13141668, 5201314
- What can these numbers represent ?
- If I say “I 7486 You”.
  - How can you interpret the number 7486 ?
  - If 7 = L, 6 = E ?, If 7 = F, 6 = K ?
- IMPORTANT
  1. Number without unit means nothing, or can mean anything.
  2. It is important for you to know the meaning of the number in that particular context.
  3. Same number can have different representations.

Bit and Byte

- What is the basic unit of memory ? bit
- What is the common unit of memory ? byte
- What is the relation between bit and byte ? 8 bit = 1 byte
- When you go out to buy a SD Card, it said 4GB.
  - It means 4G bits or 4G bytes ?
- What 4G bytes actually means ?
  - Each unit is a byte
  - Total 4G units

32 64 128 256



Memory

- What is the function of memory ?
- Assume I have one byte of memory.

11111111

- What does this byte represent ?
- Is it a positive number or negative number ?
  - If positive, it means :
  - If negative, it means :
- Who defines the meaning of that number ?
- Remember that memory is only a storage, it does not give any meaning to the number it stored. It is the programmer who defines the meaning of the number.

Data Type

- What are the data type in C ?
- How do data type related to Memory ?

```
e.g. int x ;
x =  –1; // How do x stored in memory, how many bytes ?
         // What is the exact number stored
unsigned int y;
y = x;   // What is the value of y ?
         // What is the exact number stored
```

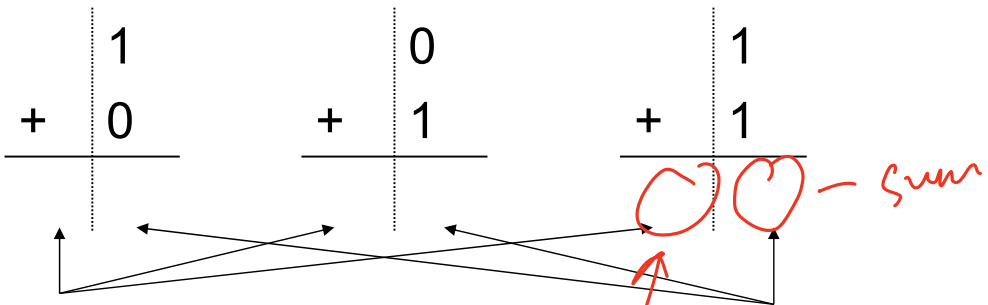
- Please note that in this course we use C, and you are required to familiarize the C programming.

Assembly instructions

- In Computer Organization course, you learnt MIPS, what are the sizes of register that MIPS has ?
- How many registers do MIPS got ?
- What will the following instruction do in MIPS ?
  - add $s0, $s1, $s2
  - addi $t0, $zero, 1
  - lw $s0, 0($s1)
- Well, in this course, we will use C instead of assembly, however, it would be necessary for you to understand the assembly instructions so that you know what the computer is doing.

Revision

- In the Digital Circuits and Systems course, you learnt a simple half adder.

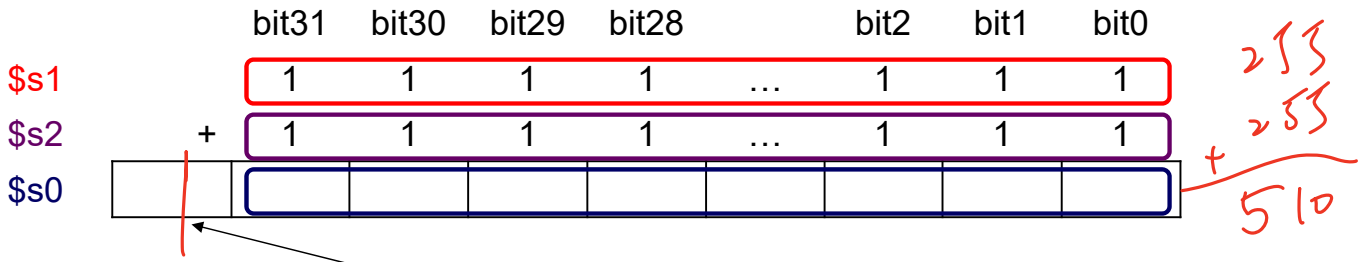


- These bits are called ?
- These bits are called ?

carry 進位

Revision

- In the Computer Organization course, if we implement the instruction : add $s0, $s1, $s2
- Remember that $s0, $s1, and $s2 are 32-bit registers
- Assuming $s1 and $s2 contains the value 0xFFFFFFFF



- What event happened ? This bit is called ?

overflow.

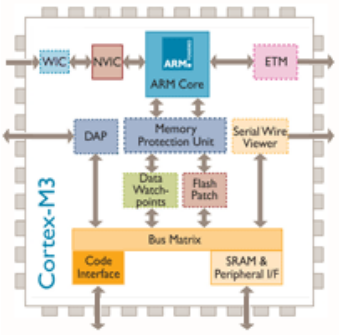
Misconception…

- In the Computer Organization course, you learn a term Overflow. What is Overflow ? What is Carry ? Do they the same thing ?

- Under what conditions will there be Carry, Overflow ?
- Although again you will use C in this course, you should be able to know how the computer makes the decision based on the status.

ARM Cortex-M3 microcontroller

- In this course, we will use a series of ARM Cortex-M3 based microcontroller.
- For general Cortex-M3 information, you can go to
  - http://www.arm.com/products/processors/cortex-m/cortex-m3.php
- We will use a evaluation board that is based on the controller developed by STMicroelectronics under the family of STM32F103xx
- You can get the datasheet from the course website.

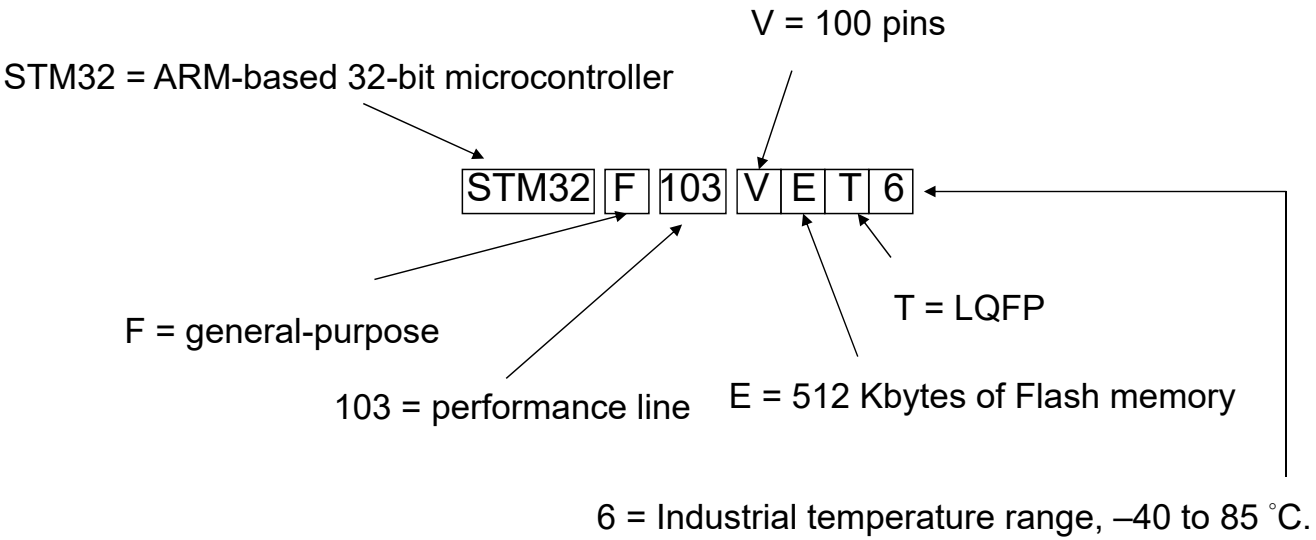


STM32F103xC STM32F103xD STM32F103xE

High-density performance line ARM-based 32-bit MCU with 256 to 512KB Flash, USB, CAN, 11 timers, 3 ADCs, 13 communication interfaces

STM32F103xx microcontroller

- Since different IC have different function and pinout, in particular, we are using STM32F103VET6 , for details, please refer to the datasheet.



STM32F103xx microcontroller

- Other Packages and their function summaries are shown here

Pin count	Part Number	CPU Max. Clock (MHz)	Program Memory (bytes)	RAM (bytes)	FSMC	Timer			Communication Interface									Analog port		I/O port	Package
						16-bit General Purpose (IC/OC/PWM)	16-bit Advanced (IC/OC/PWM)	16-bit Basic	SPI	I²C	USART* +UART	USB FS	CAN 2.0B	Ether net	CEC	I²S	SDIO	12-bit ADC (CH.)	12-bit DAC (CH.)		
36	STM32F103T4	72	16K	6K		2(8/8/8)	1(4/4/6)		1	1	2	1	1					2/(10)		26	VFQFPN36(6x6)
	STM32F103T6	72	32K	10K		2(8/8/8)	1(4/4/6)		1	1	2	1	1					2/(10)		26	
	STM32F103T8	72	64K	20K		3(12/12/12)	1(4/4/6)		1	1	2	1	1					2/(10)		26	
	STM32F103TB	72	128K	20K		3(12/12/12)	1(4/4/6)		1	1	2	1	1					2/(10)		26	
48	STM32F103C4	72	16K	6K		2(8/8/8)	1(4/4/6)		1	1	2	1	1					2/(10)		37	LQFP48(7x7)/ VFQFPN48(7x7)
	STM32F103C6	72	32K	10K		2(8/8/8)	1(4/4/6)		1	1	2	1	1					2/(10)		37	
	STM32F103C8	72	64K	20K		3(12/12/12)	1(4/4/6)		2	2	3	1	1					2/(10)		37	
	STM32F103CB	72	128K	20K		3(12/12/12)	1(4/4/6)		2	2	3	1	1					2/(10)		37	
64	STM32F103R4	72	16K	6K		2(8/8/8)	1(4/4/6)		1	1	2	1	1					2/(16)		51	LQFP64(10x10) TFBGA64(5x5)
	STM32F103R6	72	32K	10K		2(8/8/8)	1(4/4/6)		1	1	2	1	1					2/(16)		51	
	STM32F103R8	72	64K	20K		3(12/12/12)	1(4/4/6)		2	2	3	1	1					2/(16)		51	
	STM32F103RB	72	128K	20K		3(12/12/12)	1(4/4/6)		2	2	3	1	1					2/(16)		51	
	STM32F103RC	72	256K	48K		4(16/16/16)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(16)	2	51	LQFP64 (10x10) WLCSP64(4,5 × 4,4)
	STM32F103RD	72	384K	64K		4(16/16/16)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(16)	2	51	
	STM32F103RE	72	512K	64K		4(16/16/16)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(16)	2	51	
	STM32F103RF	72	768K	96K		10(24/24/24)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(16)	2	51	LQFP64 (10x10)
	STM32F103RG	72	1024K	96K		10(24/24/24)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(16)	2	51	

STM32F103xx microcontroller

100	STM32F103V8	72	64K	20K		3(12/12/12)	1(4/4/6)		2	2	3	1	1					2/(16)		80	LQFP100(14x14) LFBGA100(10x10)
	STM32F103VB	72	128K	20K		3(12/12/12)	1(4/4/6)		2	2	3	1	1					2/(16)		80	
	STM32F103VC	72	256K	48K	•	4(16/16/16)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(16)	2	80	
	STM32F103VD	72	384K	64K	•	4(16/16/16)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(16)	2	80	
	STM32F103VE	72	512K	64K	•	4(16/16/16)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(16)	2	80	
	STM32F103VF	72	768K	96K	•	10(24/24/24)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(16)	2	80	LQFP100(14x14)
	STM32F103VG	72	1024K	96K	•	10(24/24/24)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(16)	2	80	
144	STM32F103ZC	72	256K	48K	•	4(16/16/16)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(21)	2	112	LQFP144(20x20) BGA144(10x10)
	STM32F103ZD	72	384K	64K	•	4(16/16/16)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(21)	2	112	
	STM32F103ZE	72	512K	64K	•	4(16/16/16)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(21)	2	112	
	STM32F103ZF	72	768K	96K	•	10(24/24/24)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(21)	2	112	
	STM32F103ZG	72	1024K	96K	•	10(24/24/24)	2(8/8/12)	2	3	2	3+2	1	1			2	1	3/(21)	2	112	

STM32F103xx microcontroller

Selected features of the microcontroller

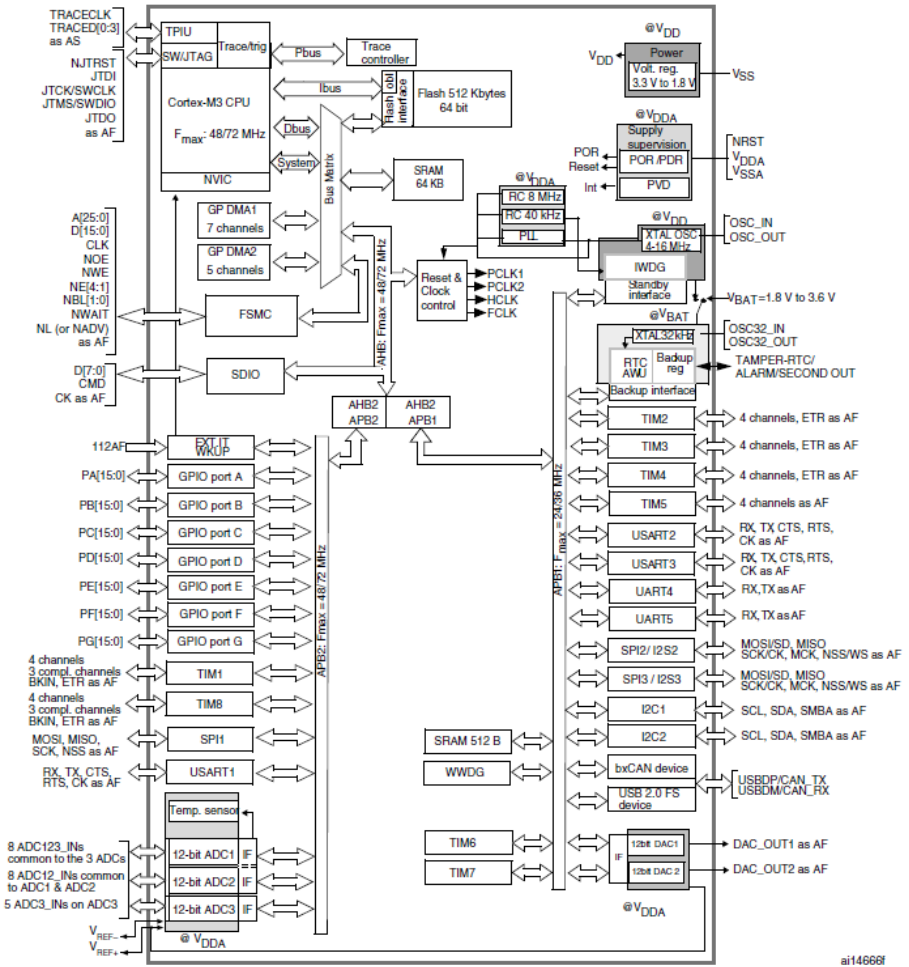
- Core: ARM 32-bit Cortex™-M3 CPU
  - 72 MHz maximum frequency
- Memories
  - 512 Kbytes of Flash memory
  - up to 64 Kbytes of SRAM
  - Flexible static memory controller with 4 Chip Select. Supports Compact Flash, SRAM, PSRAM, NOR and NAND memories
- Clock, reset and supply management
  - 4-to-16 MHz crystal oscillator
  - Internal 8 MHz factory-trimmed RC
  - Internal 40 kHz RC with calibration
  - 32 kHz oscillator for RTC with calibration
- 3 × 12-bit, 1 μs A/D converters (up to 21 channels)
- 2 × 12-bit D/A converters
- Up to 112 fast I/O ports
  - 51/80/112 I/Os, all mappable on 16 external interrupt vectors and almost all 5 V-tolerant

STM32F103xx microcontroller

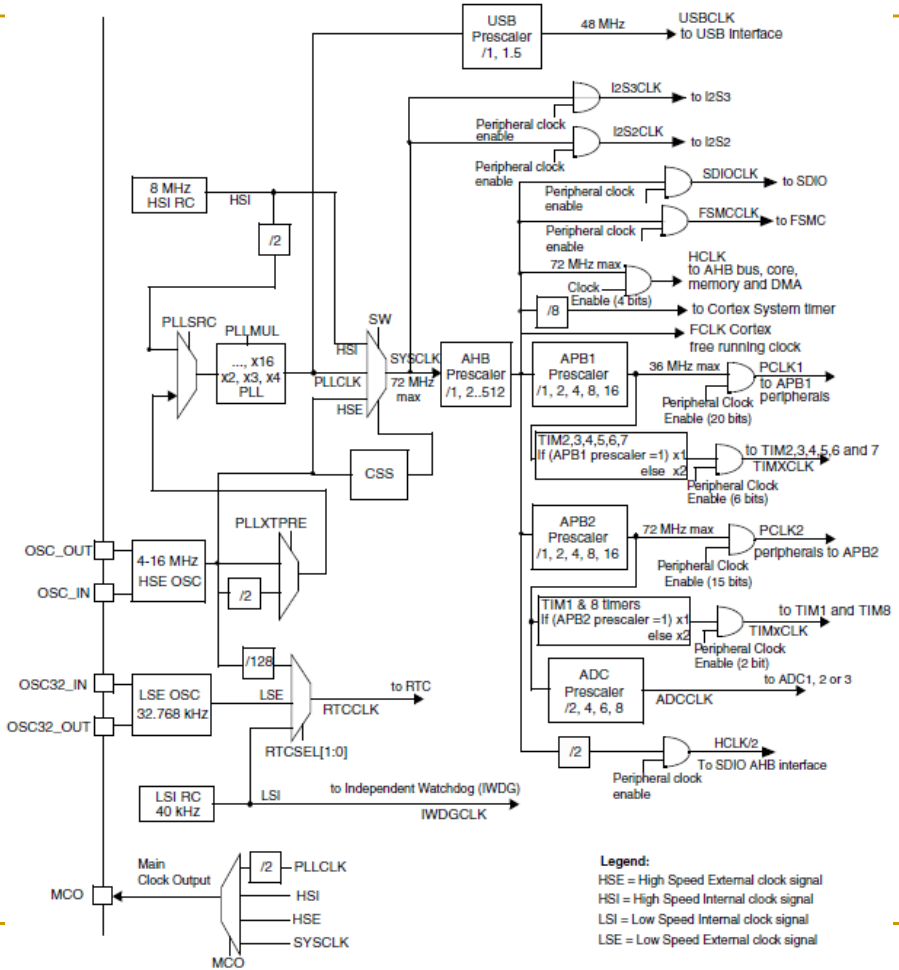
Selected features of the microcontroller

- Up to 11 timers
  - Up to four 16-bit timers, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
  - 2 × 16-bit motor control PWM timers with dead-time generation and emergency stop
  - 2 × watchdog timers (Independent and Window)
  - SysTick timer: a 24-bit downcounter
  - 2 × 16-bit basic timers to drive the DAC
- Up to 13 communication interfaces
  - Up to 2 × I^2C interfaces (SMBus/PMBus)
  - Up to 5 USARTs (ISO 7816 interface, LIN, IrDA capability, modem control)
  - Up to 3 SPIs (18 Mbit/s), 2 with I^2S interface multiplexed
  - CAN interface (2.0B Active)
  - USB 2.0 full speed interface
  - SDIO interface
- CRC calculation unit, 96-bit unique ID
- ECOPACK® packages

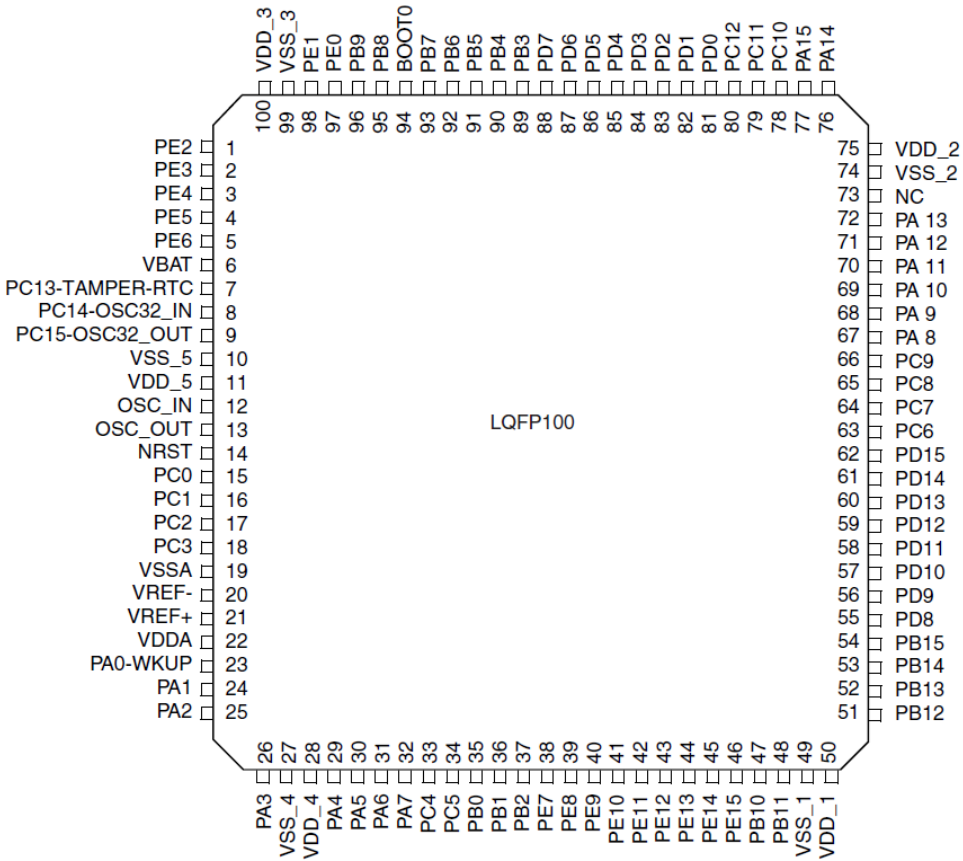
Block Diagram



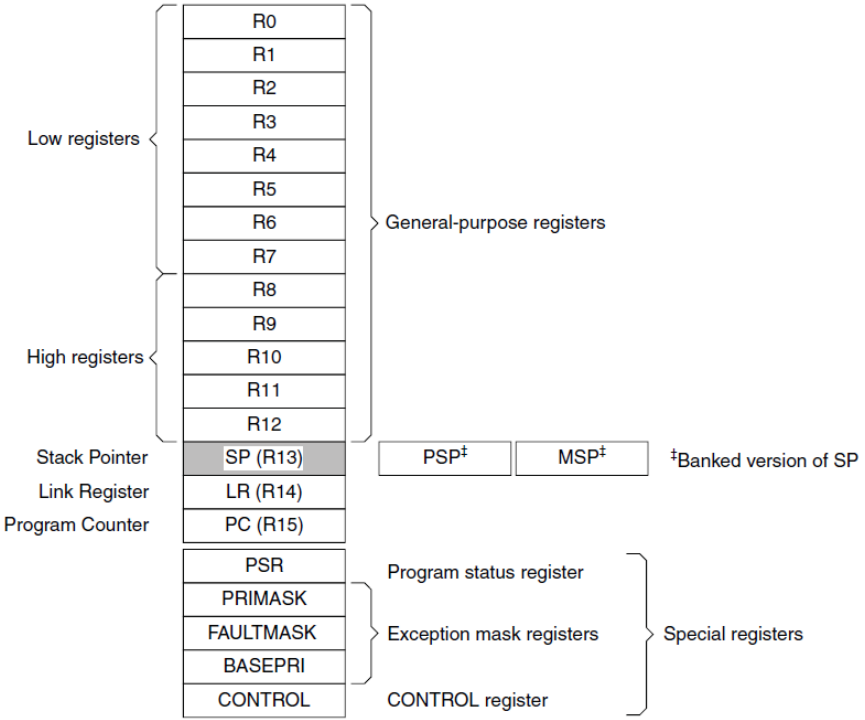
Clock Tree


USB Prescaler /1, 1.5
48 MHz
USBCLK to USB Interface
I2S3CLK
to I2S3
Peripheral clock enable
I2S2CLK
to I2S2
Peripheral clock enable
SDIOCLK
to SDIO
Peripheral clock enable
FSMCCLK
to FSMC
Peripheral clock enable
72 MHz max
HCLK to AHB bus, core, memory and DMA
Clock Enable (4 bits)
/8
to Cortex System timer
FCLK Cortex free running clock
8 MHz HSI RC
HSI
/2
PLLSRC
PLLMUL
..., x16 x2, x3, x4 PLL
HSI
PLLCLK
HSE
SW
SYSCLK
72 MHz max
AHB Prescaler /1, 2..512
APB1 Prescaler /1, 2, 4, 8, 16
36 MHz max
PCLK1
to APB1 peripherals
Peripheral Clock Enable (20 bits)
TIM2,3,4,5,6,7 If (APB1 prescaler =1) x1 else x2
to TIM2,3,4,5,6 and 7 TIMXCLK
Peripheral Clock Enable (6 bits)
CSS
PLLXTPRE
OSC_OUT
OSC_IN
4-16 MHz HSE OSC
/2
APB2 Prescaler /1, 2, 4, 8, 16
72 MHz max
PCLK2
peripherals to APB2
Peripheral Clock Enable (15 bits)
TIM1 & 8 timers If (APB2 prescaler =1) x1 else x2
to TIM1 and TIM8 TIMxCLK
Peripheral Clock Enable (2 bit)
/128
OSC32_IN
OSC32_OUT
LSE OSC 32.768 kHz
LSE
RTCCLK
to RTC
RTCSEL[1:0]
ADC Prescaler /2, 4, 6, 8
to ADC1, 2 or 3
ADCCLK
/2
HCLK/2
To SDIO AHB interface
Peripheral clock enable
LSI RC 40 kHz
LSI
to Independent Watchdog (IWDG)
IWDGCLK
Main Clock Output
/2
PLLCLK
HSI
HSE
SYSCLK
MCO
MCO
Legend:
HSE = High Speed External clock signal
HSI = High Speed Internal clock signal
LSI = Low Speed Internal clock signal
LSE = Low Speed External clock signal

Pinout

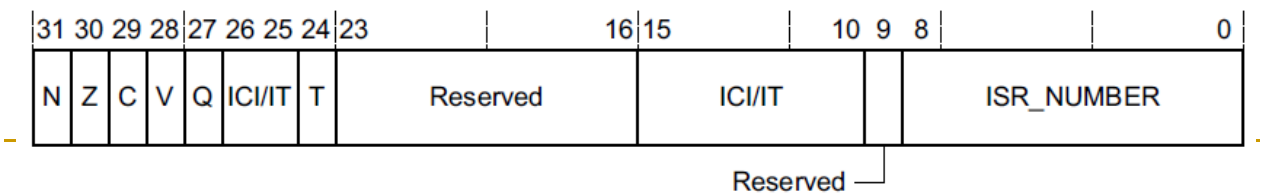

LQFP100
1 PE2
2 PE3
3 PE4
4 PE5
5 PE6
6 VBAT
7 PC13-TAMPER-RTC
8 PC14-OSC32_IN
9 PC15-OSC32_OUT
10 VSS_5
11 VDD_5
12 OSC_IN
13 OSC_OUT
14 NRST
15 PC0
16 PC1
17 PC2
18 PC3
19 VSSA
20 VREF-
21 VREF+
22 VDDA
23 PA0-WKUP
24 PA1
25 PA2
26 PA3
27 VSS_4
28 VDD_4
29 PA4
30 PA5
31 PA6
32 PA7
33 PC4
34 PC5
35 PB0
36 PB1
37 PB2
38 PE7
39 PE8
40 PE9
41 PE10
42 PE11
43 PE12
44 PE13
45 PE14
46 PE15
47 PB10
48 PB11
49 VSS_1
50 VDD_1
51 PB12
52 PB13
53 PB14
54 PB15
55 PD8
56 PD9
57 PD10
58 PD11
59 PD12
60 PD13
61 PD14
62 PD15
63 PC6
64 PC7
65 PC8
66 PC9
67 PA 8
68 PA 9
69 PA 10
70 PA 11
71 PA 12
72 PA 13
73 NC
74 VSS_2
75 VDD_2
76 PA14
77 PA15
78 PC10
79 PC11
80 PC12
81 PD0
82 PD1
83 PD2
84 PD3
85 PD4
86 PD5
87 PD6
88 PD7
89 PB3
90 PB4
91 PB5
92 PB6
93 PB7
94 BOOT0
95 PB8
96 PB9
97 PE0
98 PE1
99 VSS_3
100 VDD_3

Cortex-M3 Core Registers



Cortex-M3 Core Registers

- The Processor Core has the following 32-bit registers.
  - R0 to R12 – a total of 13 General-purpose register
  - R13 = SP (Stack Pointer)
  - R14 = LR (Link Register, stores the return information for subroutines, function calls and exceptions)
  - R15 = PC (Program Counter, contains the current program address)
- Program Status Register – Show the status of the Processor
  - A Combination of
    - APSR – Application Program Status Register
    - IPSR – Interrupt Program Status Register
    - EPSR – Execution Program Status Register

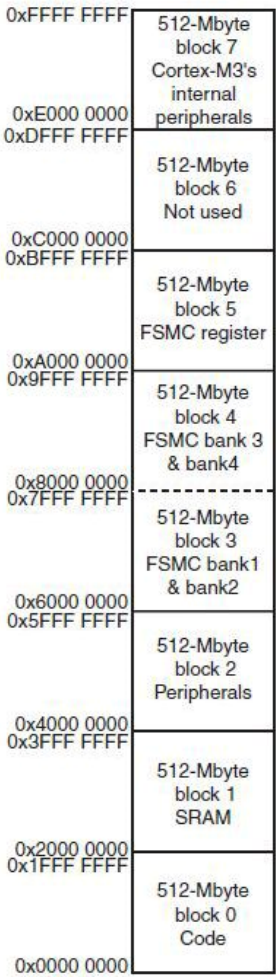


Application Program Status Register

Bits	Description
Bit 31	**N:** Negative or less than flag: 0: Operation result was positive, zero, greater than, or equal 1: Operation result was negative or less than.
Bit 30	**Z:** Zero flag: 0: Operation result was not zero 1: Operation result was zero.
Bit 29	**C:** Carry or borrow flag: 0: Add operation did not result in a carry bit or subtract operation resulted in a borrow bit 1: Add operation resulted in a carry bit or subtract operation did not result in a borrow bit.
Bit 28	**V:** Overflow flag: 0: Operation did not result in an overflow 1: Operation resulted in an overflow.
Bit 27	**Q:** Sticky saturation flag: 0: Indicates that saturation has not occurred since reset or since the bit was last cleared to zero 1: Indicates when an SSAT or USAT instruction results in saturation. This bit is cleared to zero by software using an MRS instruction.

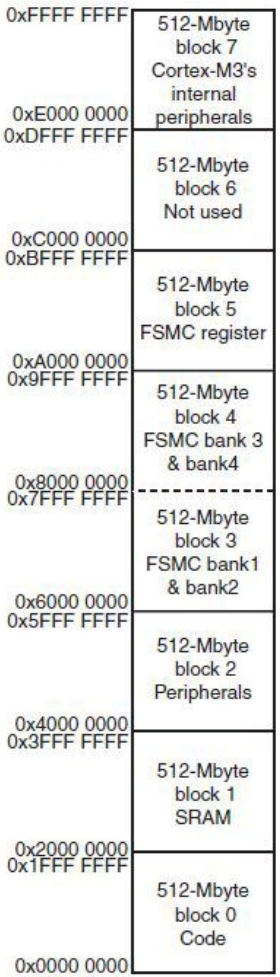
STM32F103xx Memory Map

- As the microcontroller is 32-bit, it has a 4G-byte of memory address space.



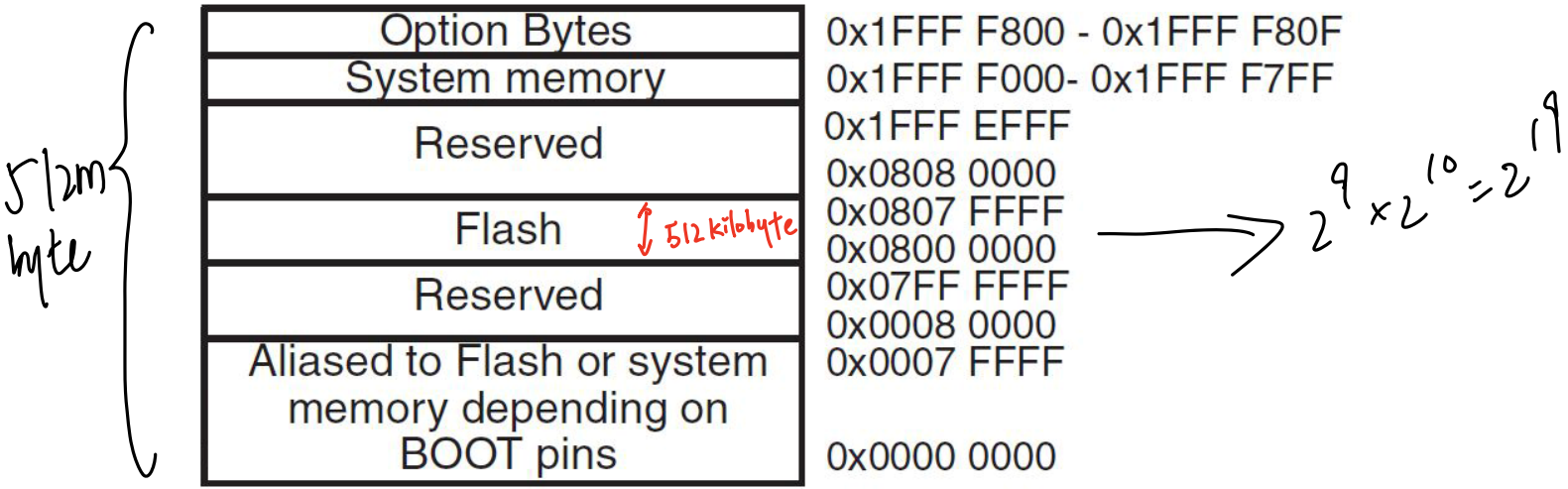
STM32F103xx Memory Map

- As the microcontroller is 32-bit, it has a 4G-byte of memory address space.
- It is divided into eight 512M-byte blocks.
  - Block 0 – Code
  - Block 1 – SRAM
  - Block 2 – Peripherals
  - Block 3 – FSMC Bank 1 and 2
  - Block 4 – FSMC Bank 3 and 4
  - Block 5 – FSMC Register
  - Block 6 – Not used
  - Block 7 – Cortex Internal Peripherals



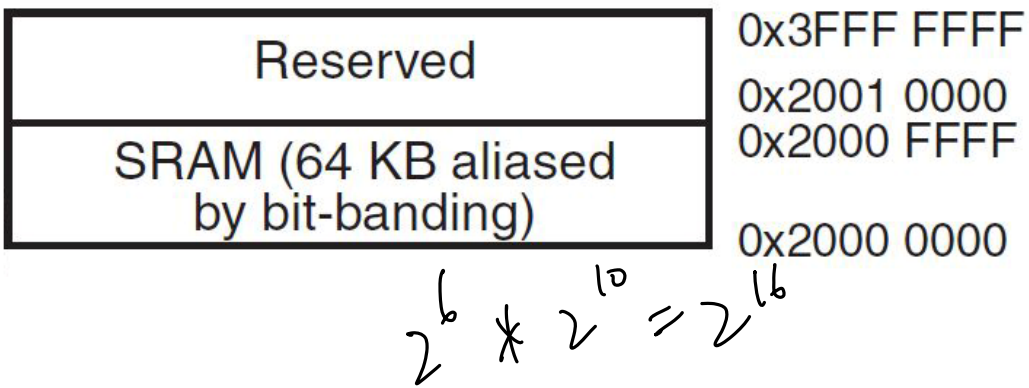
STM32F103xx – Block 0

- Block 0 is where the code is located. Normally, your code should be stored in the Flash Area.. Remember how many Flash do STM32F103VET6 have ? Can you figure it out ?



STM32F103xx – Block 1

- Block 1 is where the data are located. Normally, your data that declared and stored in the memory will be stored here. Will the data be there when you power down the microcontroller ?



END