The ARM Cortex-M4 Embedded Systems:

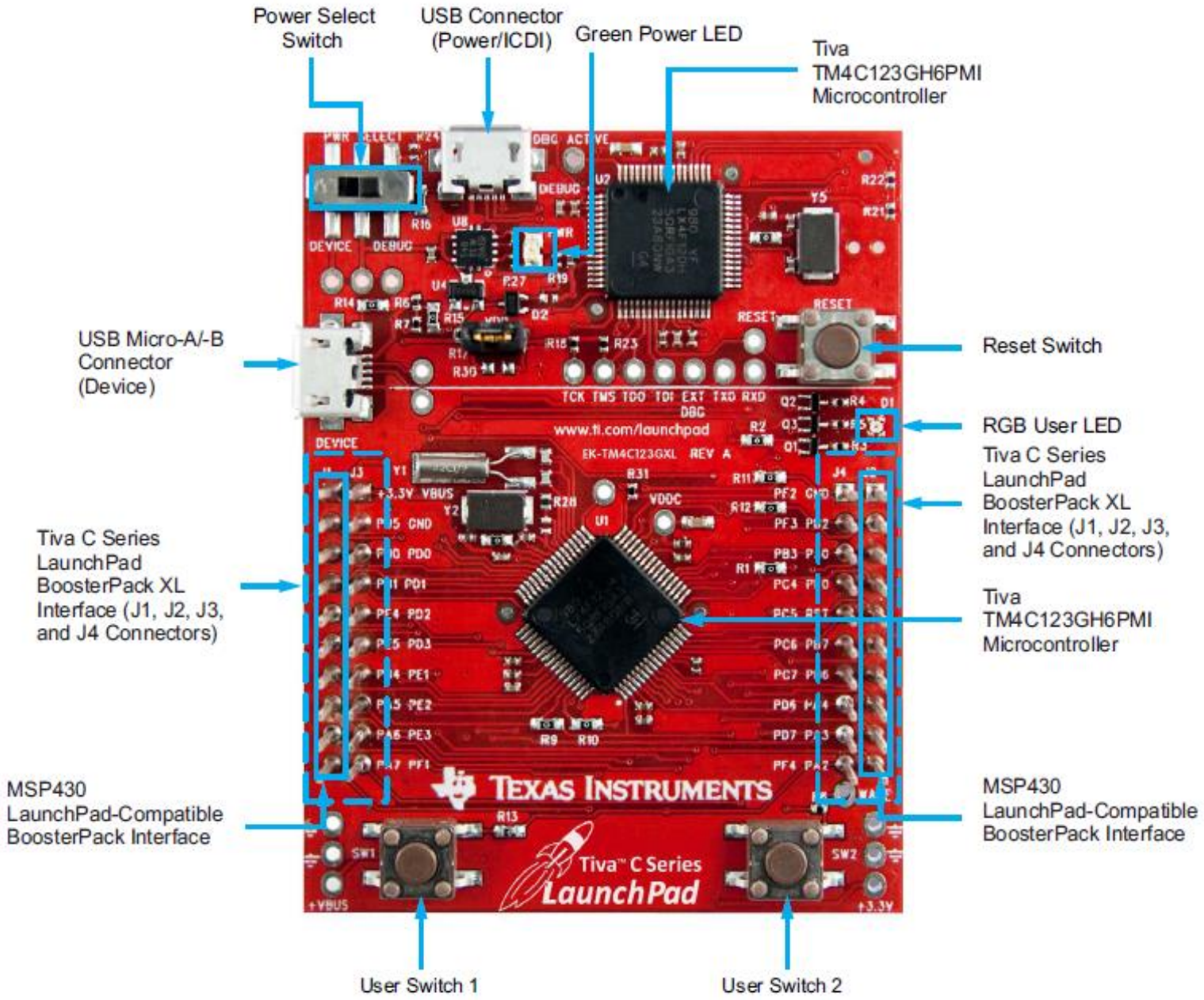
Tiva™ TM4C123GH6PM Microcontroller
General-Purpose Input/Outputs
Session 2

By: Zakriua Gomma
Email: zakriua.gomma37@gmail.com

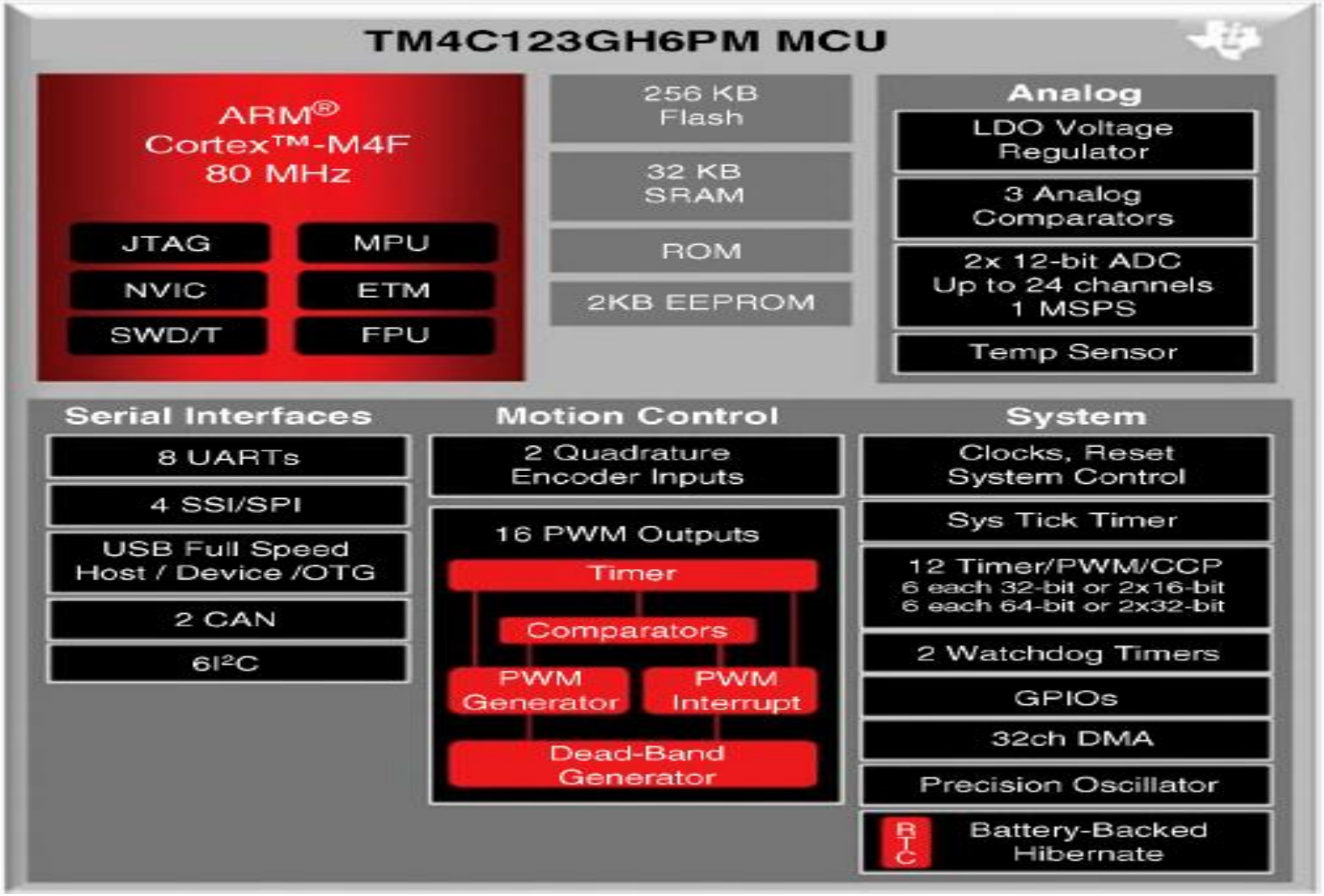
Agenda

- **TM4C123 Overview**
- General-Purpose Input/Outputs
- Bitwise operators
- System Timer (SysTick)
- Delay Library
- Interface 74595

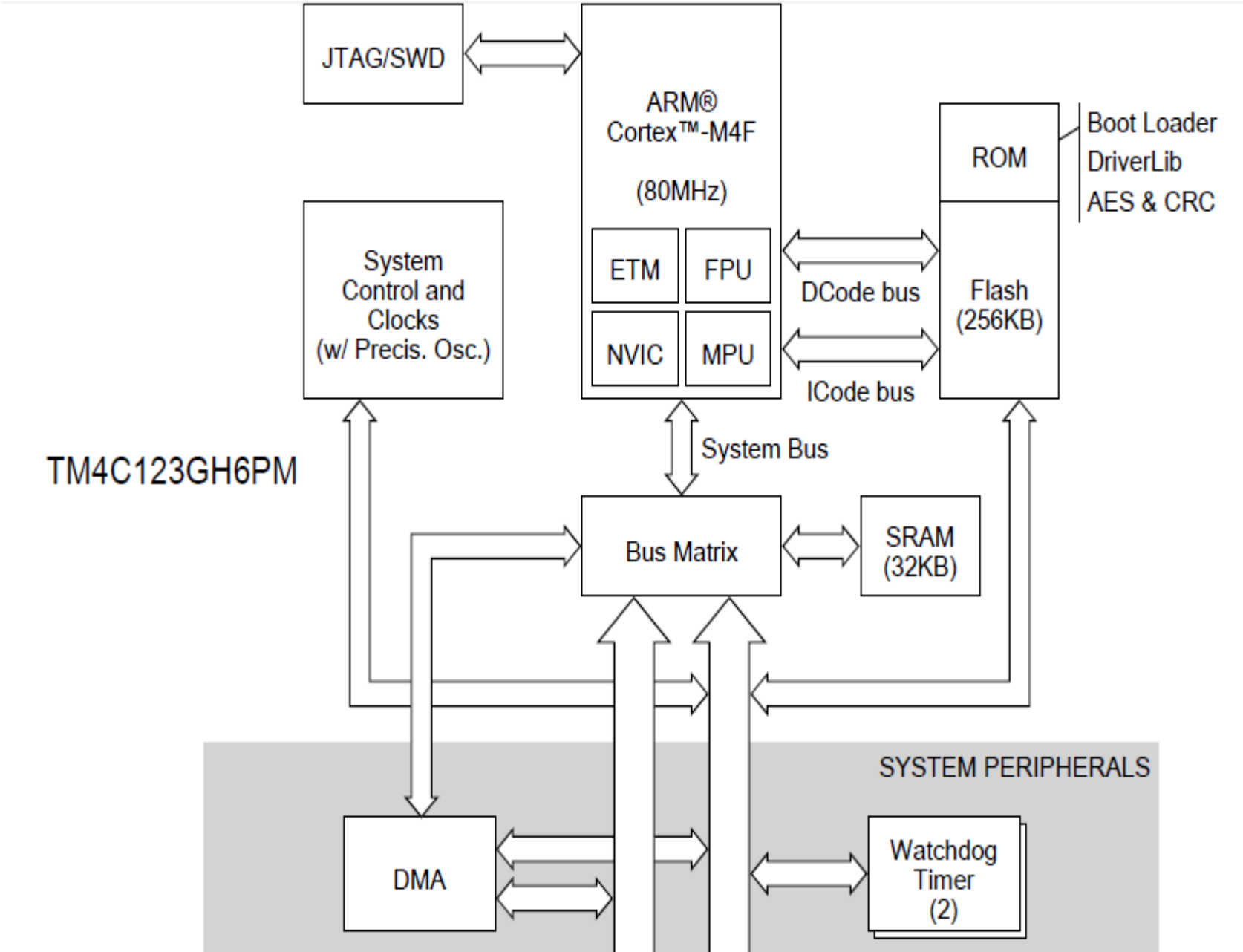
• TM4C123 Overview:



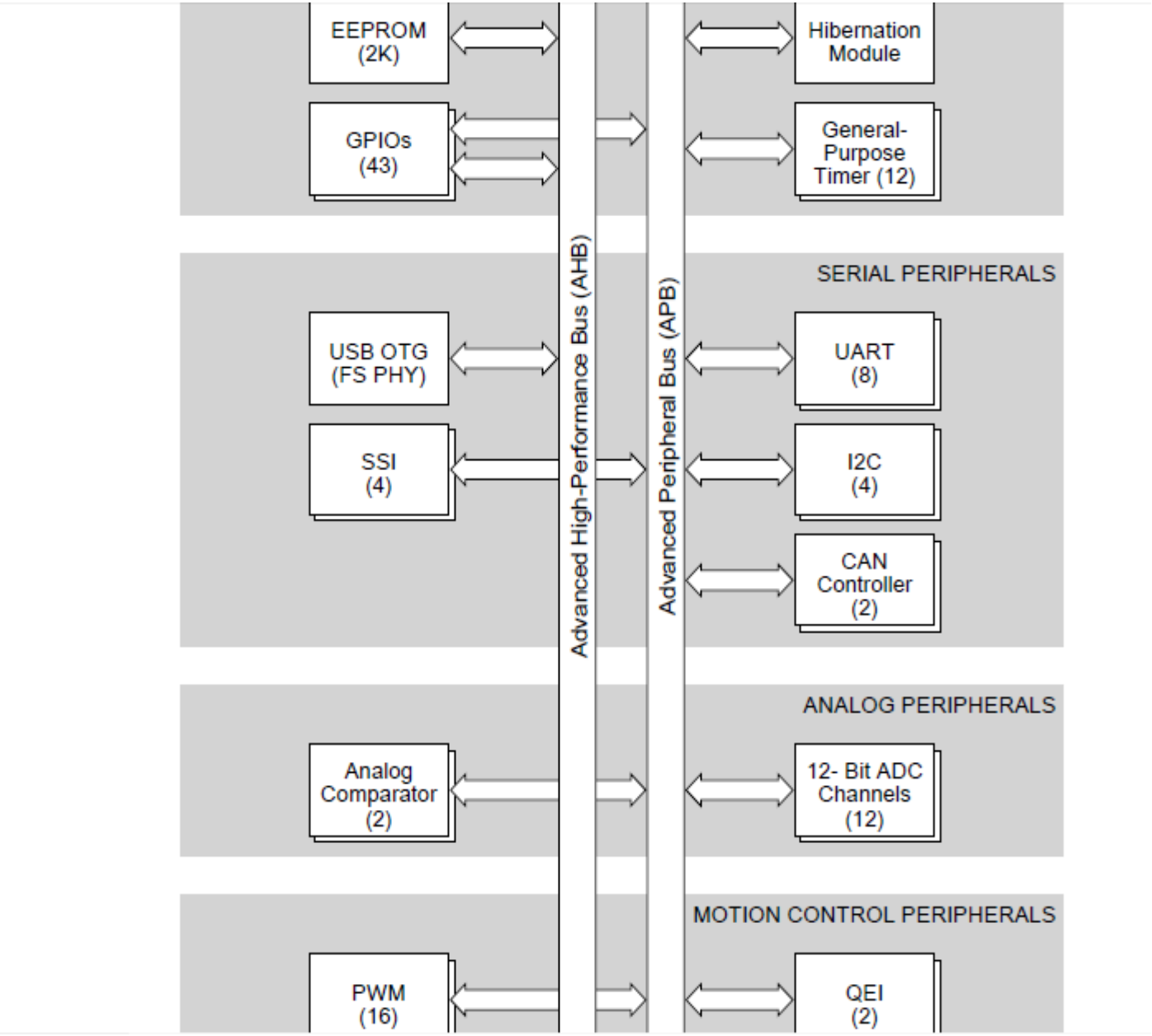
- **TM4C123GH6PM Overview:**



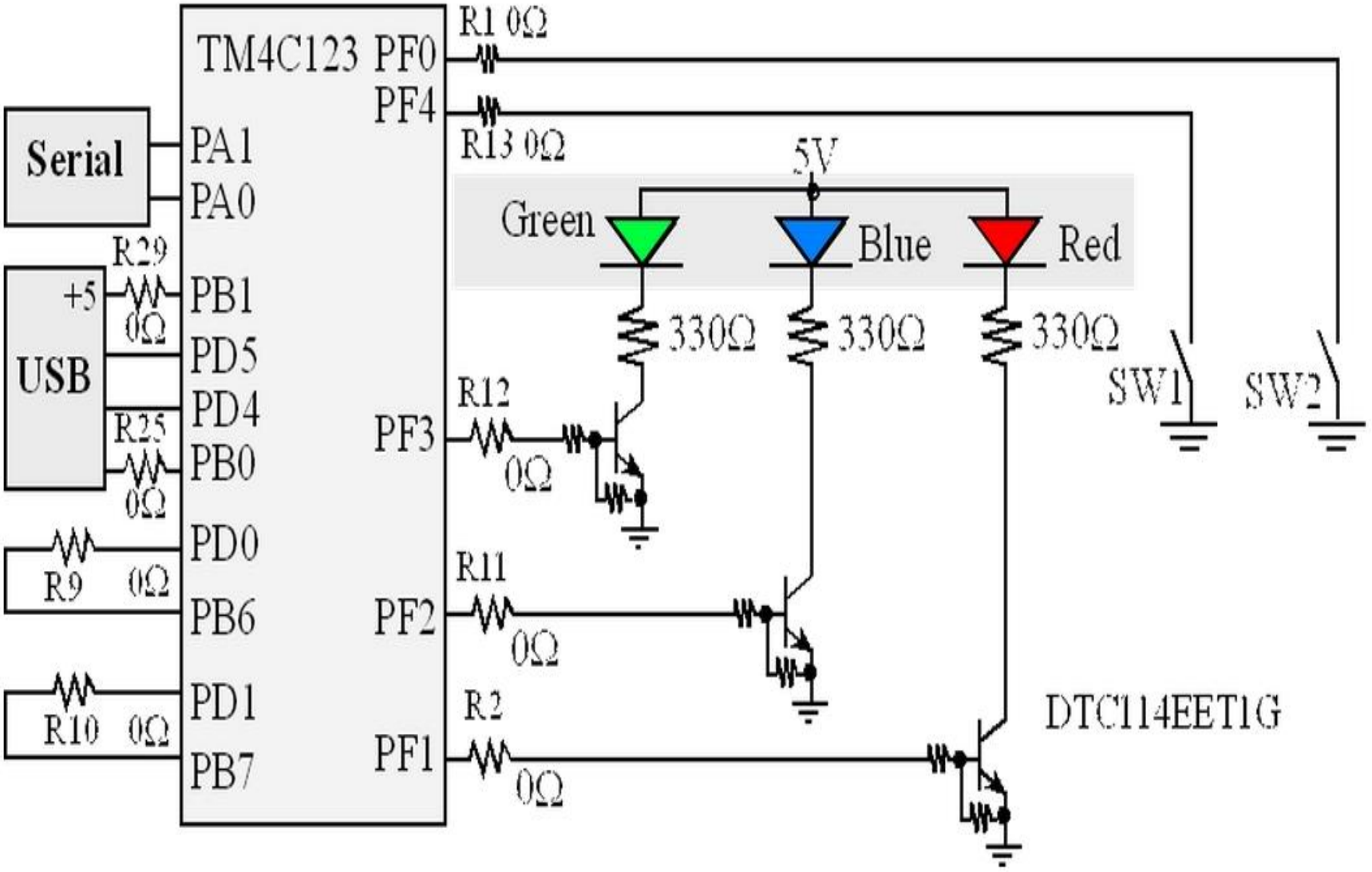
- **TM4C123GH6PM High-Level Block Diagram :**



- **TM4C123GH6PM High-Level Block Diagram :**



- **TM4C123GH6PM Overview:**



- **TM4C123GH6PM Overview:**

In-Circuit Debug Interface(ICDL)

GPIO Pin	Pin Function
PC0	TCK/SWCLK
PC1	TMS/SWDIO
PC2	TDI
PC3	TDO/SWO

- **TM4C123GH6PM Overview:**

Virtual COM Port

GPIO Pin	Pin Function
PA0	U0RX
PA1	U0TX

- **TM4C123GH6PM Overview:**

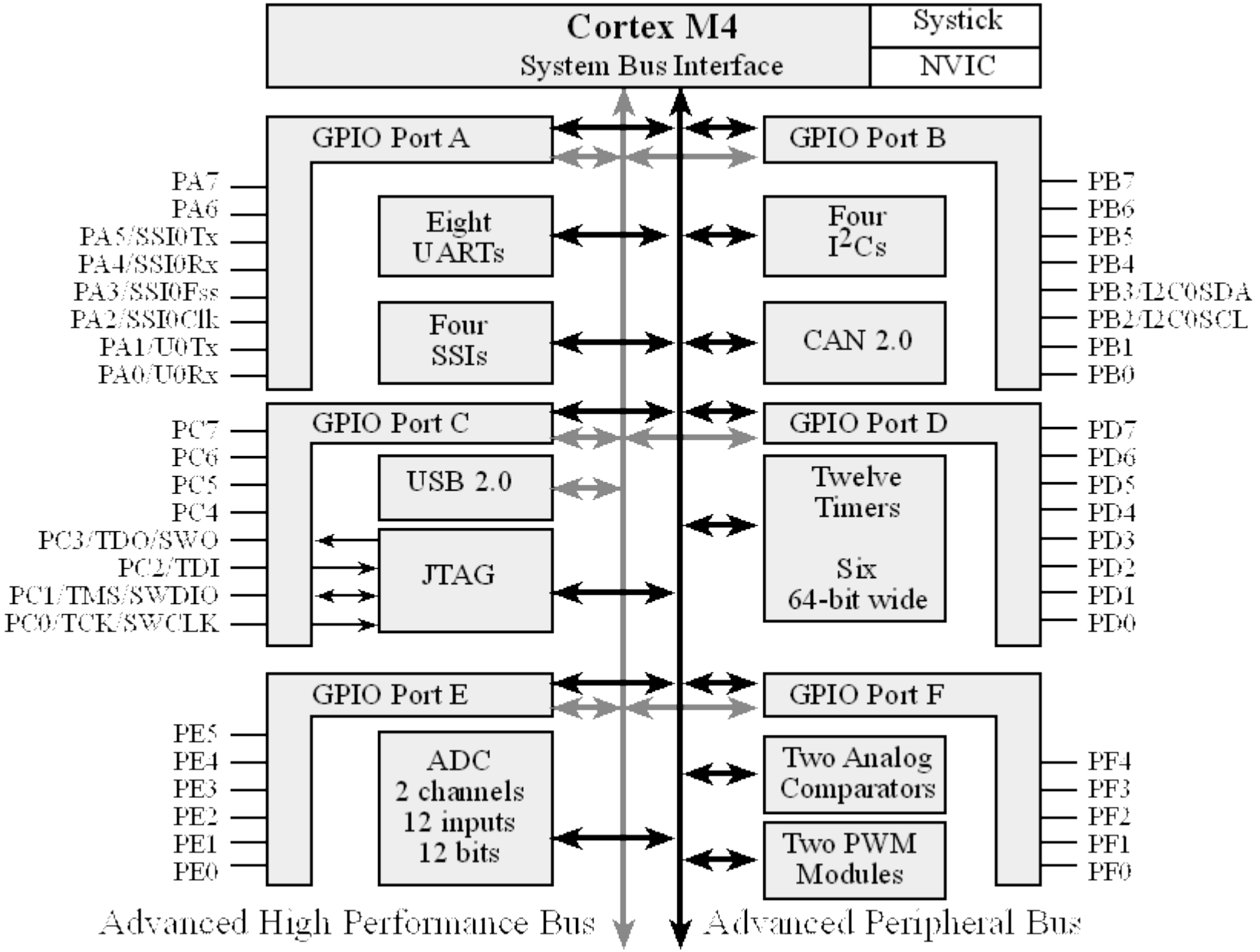
Use Switches & RGB User LED

GPIO Pin	Pin Function	USB Device
PF4	GPIO	SW1
PF0	GPIO	SW2
PF1	GPIO	RGB LED (Red)
PF2	GPIO	RGB LED (Blue)
PF3	GPIO	RGD LED (Green)

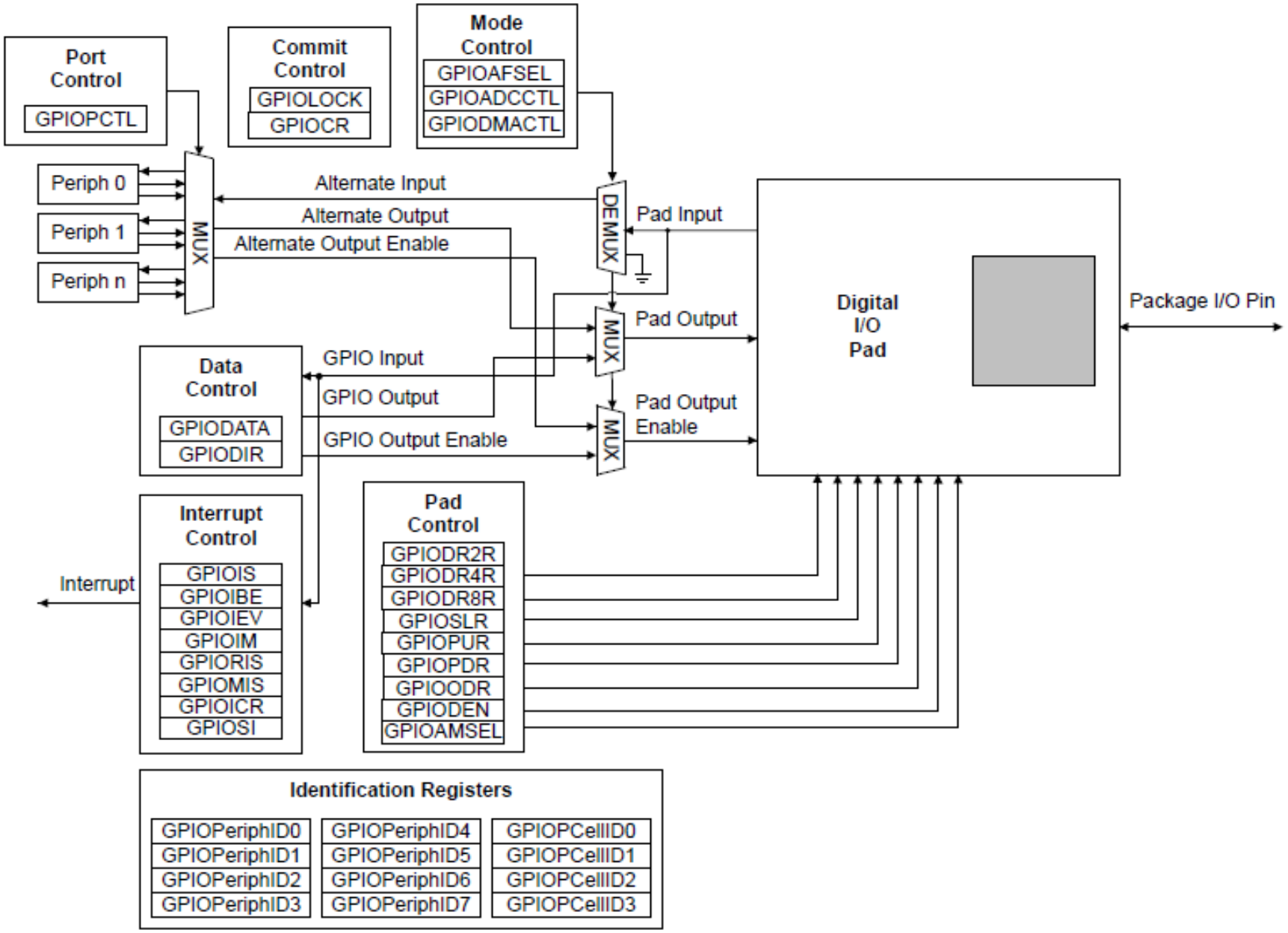
Agenda

- TM4C123 Overview
- **General-Purpose Input/Outputs**
- Bitwise operators
- System Timer (SysTick)
- Delay Library
- Interface 74595

• General-Purpose Input/Outputs



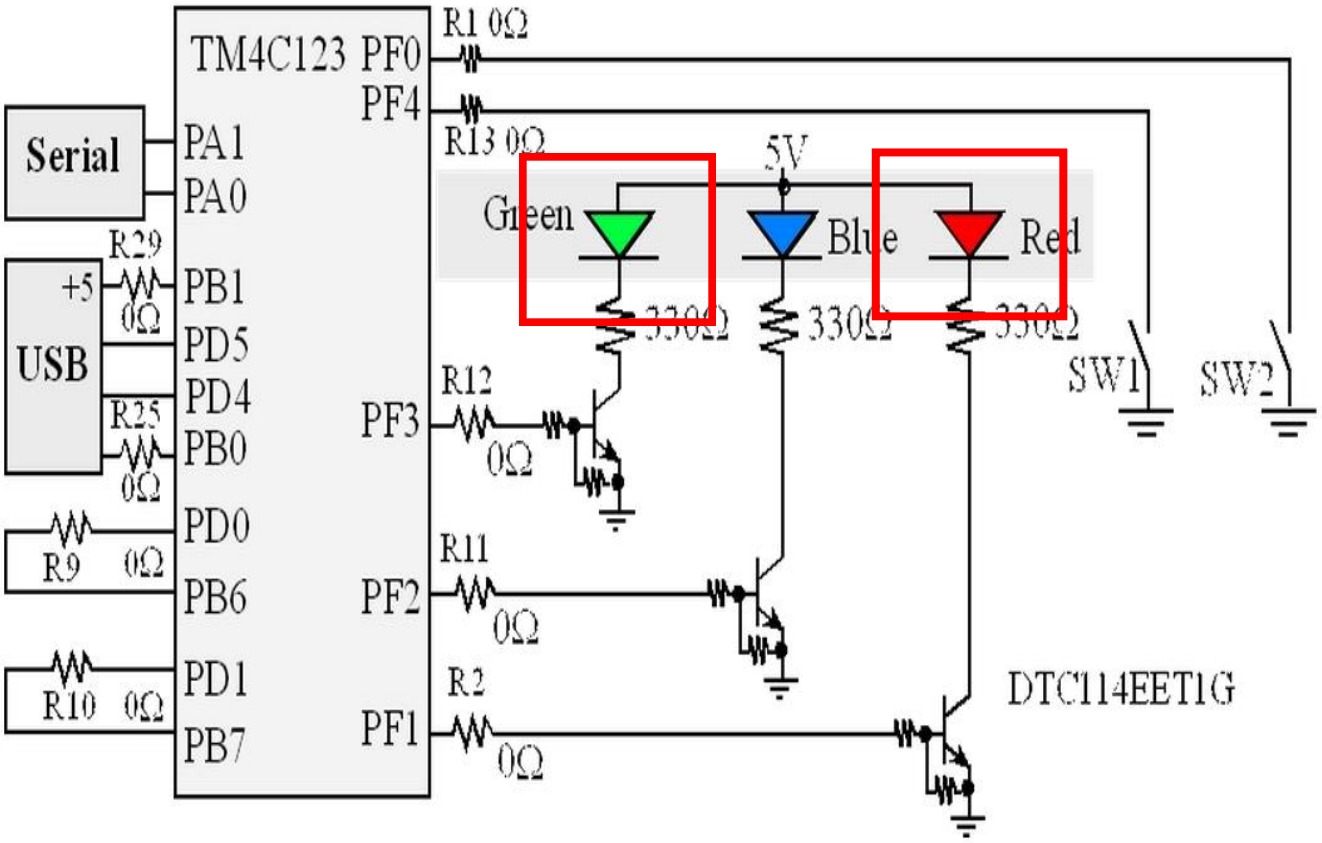
• General-Purpose Input/Outputs



- **TM4C123GH6PM Overview:**

First Program :

Blink Led red & Green



- **General-Purpose Input/Outputs**

Symbol

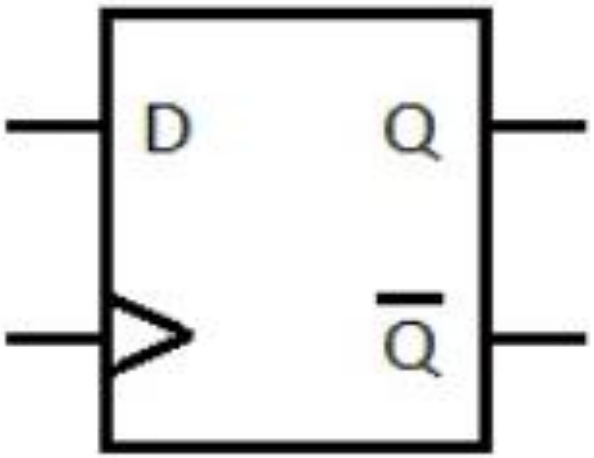
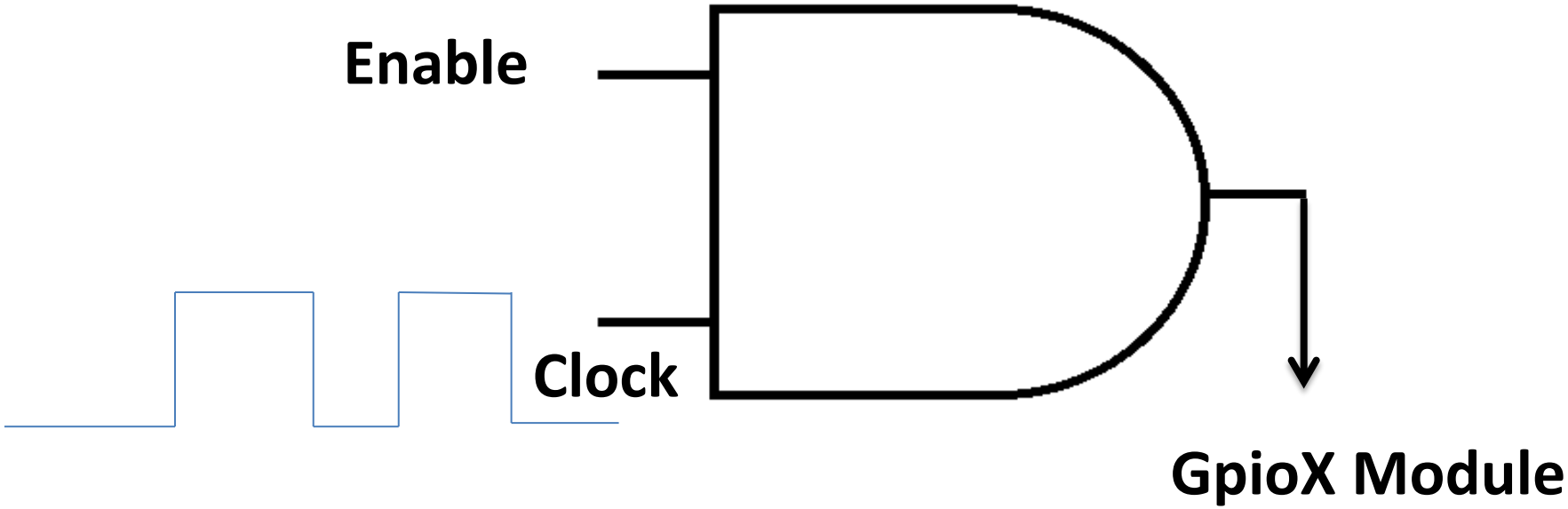


Table of truth:

clk	D	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	Q	$\overline{Q}$
1	0	0	1
1	1	1	0

- **General-Purpose Input/Outputs**

1-First We Must deliver clock to GpioX Module

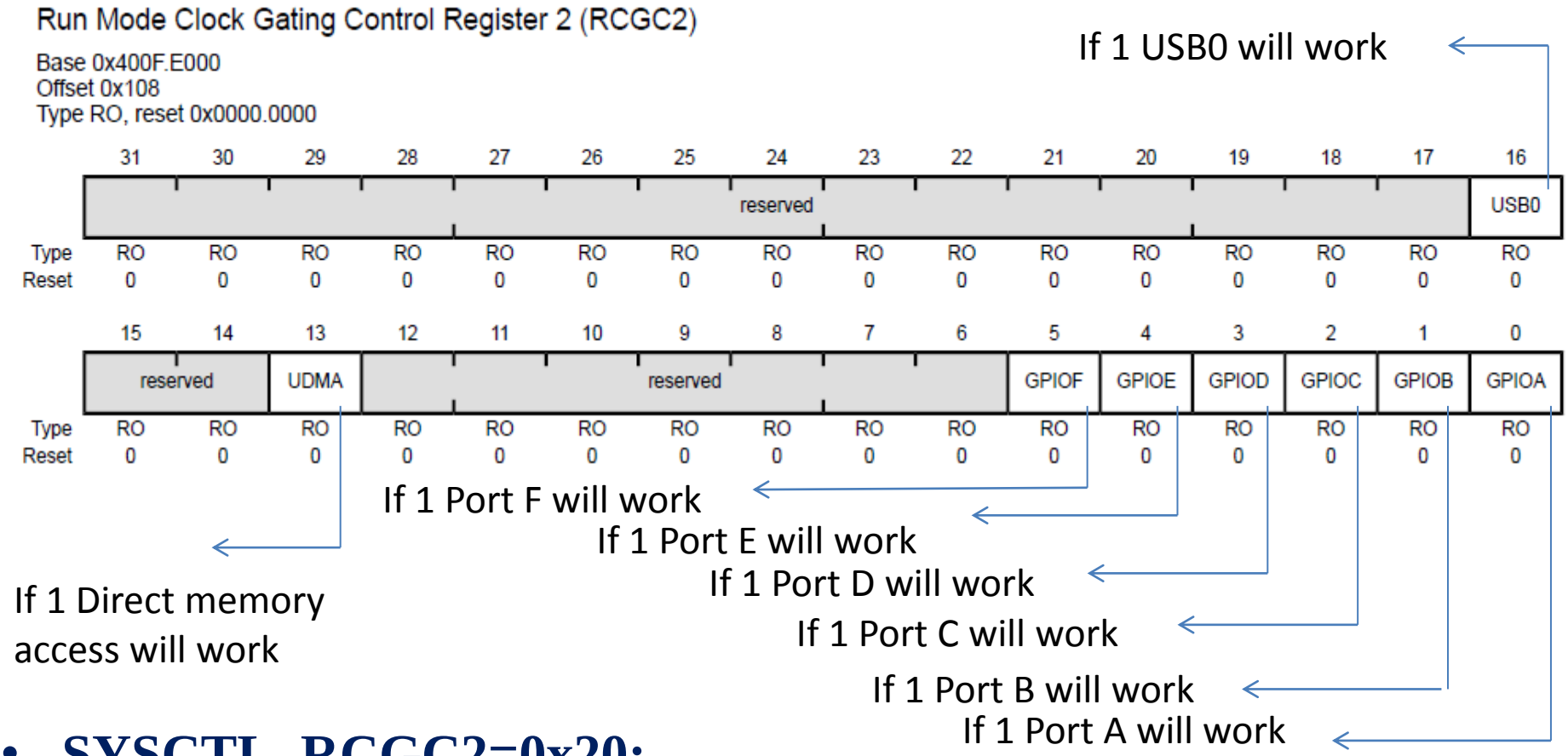


Warning : IF you try to read or write in any Register in GpioX Module without deliver Clock ,you will cause **bus fault Exception**

- **General-Purpose Input/Outputs**

1- Clock

Run Mode Clock Gating Control Register 2 (RCGC2) :



- **SYSCTL_RCGC2=0x20;**

- **General-Purpose Input/Outputs**

General-Purpose Input/Output Run Mode Clock Gating Control(RCGCGPIO),

General-Purpose Input/Output Run Mode Clock Gating Control (RCGCGPIO)

Base 0x400F.E000
Offset 0x608
Type R/W, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved										R5	R4	R3	R2	R1	R0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

If 1 Port F will work

If 1 Port E will work

If 1 Port D will work

If 1 Port C will work

If 1 Port B will work

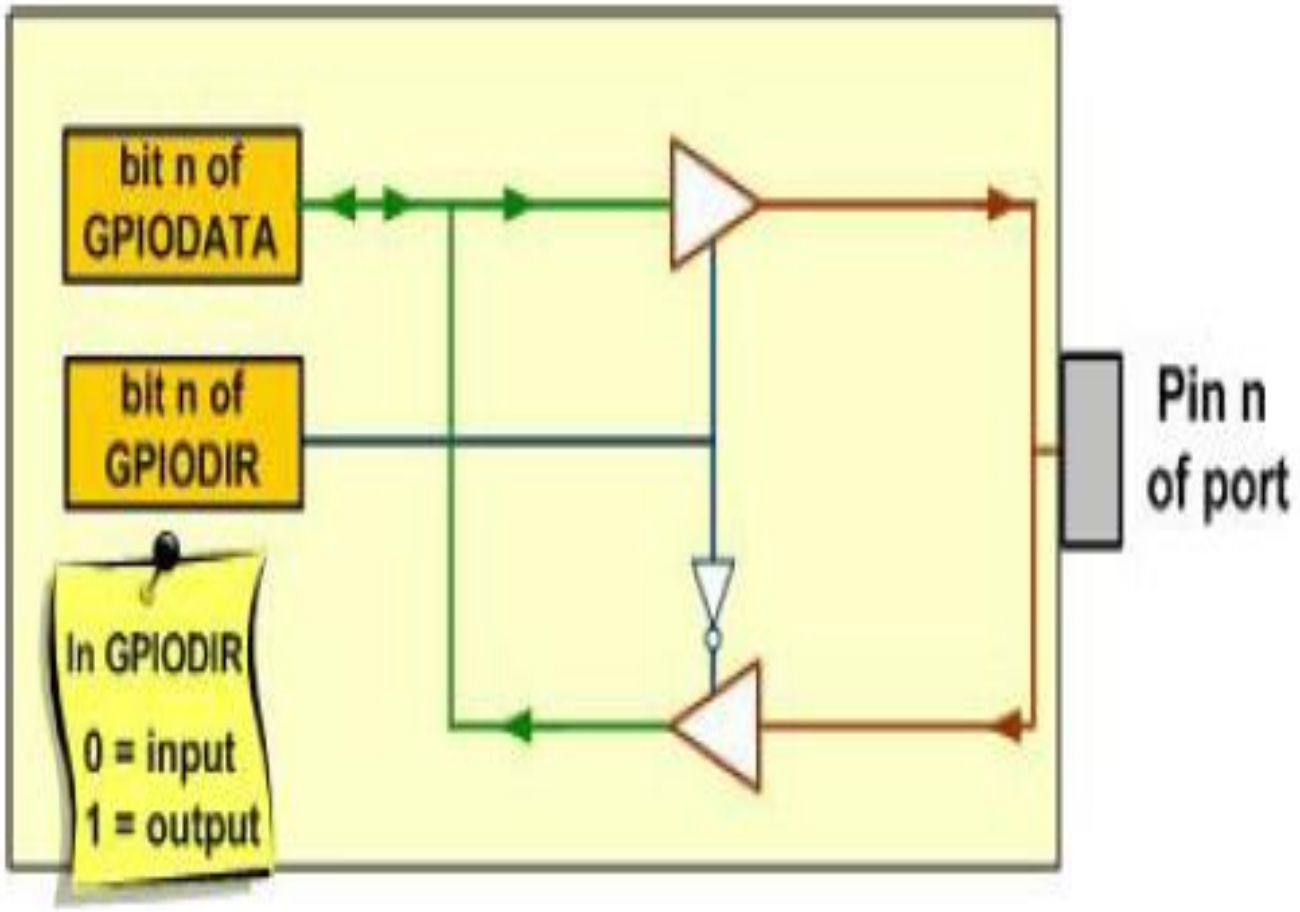
If 1 Port A will work

This register should be used to control the clocking for the GPIO modules. To support legacy software, the RCGC2 register is available. **A write to the RCGC2 register also writes the corresponding bit in this register**.

The both do the same Job. **RCGCGPIO=0x20;**

- **General-Purpose Input/Outputs**

2- GPIODIR



- **General-Purpose Input/Outputs**

2- GPIODIR

Register 2: GPIO Direction (GPIODIR), offset 0x400

The **GPIODIR** register is the data direction register. Setting a bit in the **GPIODIR** register configures the corresponding pin to be an output, while clearing a bit configures the corresponding pin to be an input. All bits are cleared by a reset, meaning all GPIO pins are inputs by default.

GPIO Direction (GPIODIR)

GPIO Port A (APB) base: 0x4000.4000
GPIO Port A (AHB) base: 0x4005.8000
GPIO Port B (APB) base: 0x4000.5000
GPIO Port B (AHB) base: 0x4005.9000
GPIO Port C (APB) base: 0x4000.6000
GPIO Port C (AHB) base: 0x4005.A000
GPIO Port D (APB) base: 0x4000.7000
GPIO Port D (AHB) base: 0x4005.B000
GPIO Port E (APB) base: 0x4002.4000
GPIO Port E (AHB) base: 0x4005.C000
GPIO Port F (APB) base: 0x4002.5000
GPIO Port F (AHB) base: 0x4005.D000
Offset 0x400
Type R/W, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								DIR							
Type	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

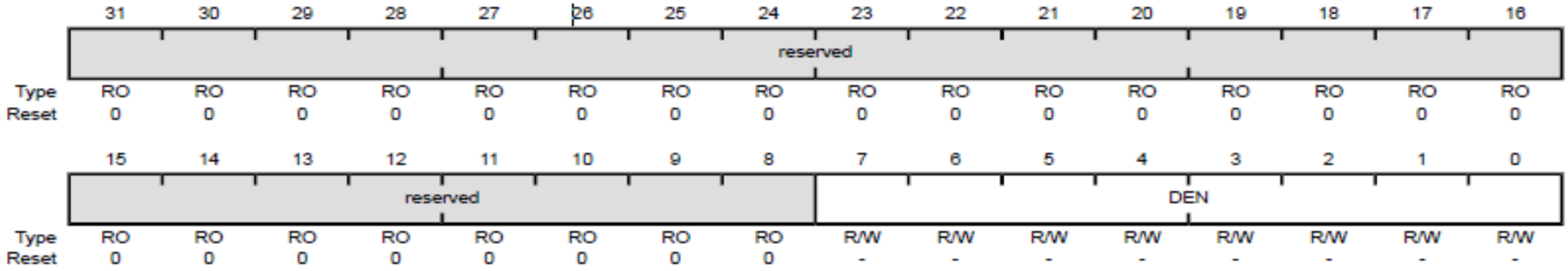
Input 0,output =1

- **GPIO_PORTF_DIR=0X0A**

=00001010 Means pin0,2,4,5,6,7 is input & pin 1,3 output

- **General-Purpose Input/Outputs**

3- GPIODEN (GPIO Digital Enable):



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DEN	R/W	-	Digital Enable

Value	Description
0	The digital functions for the corresponding pin are disabled.
1	The digital functions for the corresponding pin are enabled.

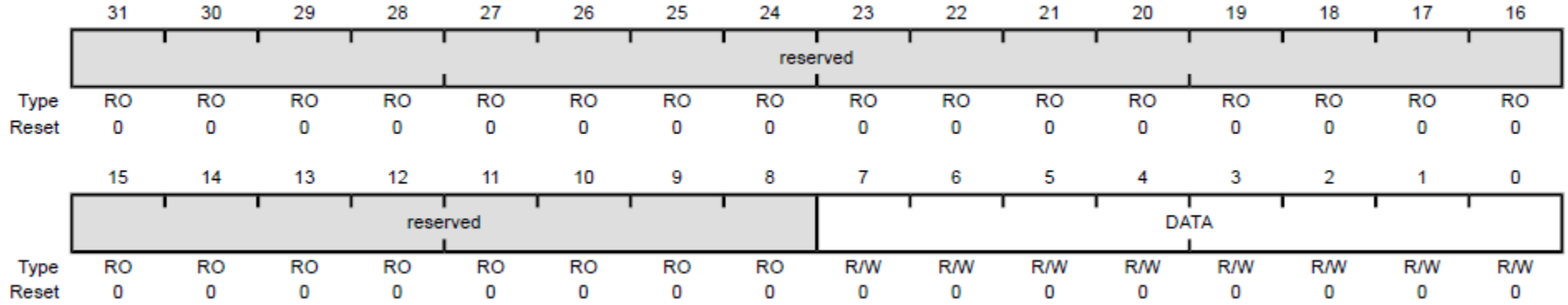
The reset value for this register is 0x0000.0000 for GPIO ports that are not listed in Table 10-1 on page 648.

The GPIODEN (Digital Enable) special function register allows us to enable the pin to be used as digital I/O pin instead of analog function. Each PORT of A-F has its own GPIODEN register and one can enable the digital I/O for each pin of a given port

GPIO_PORTF_DEN=0X0A

- **General-Purpose Input/Outputs**

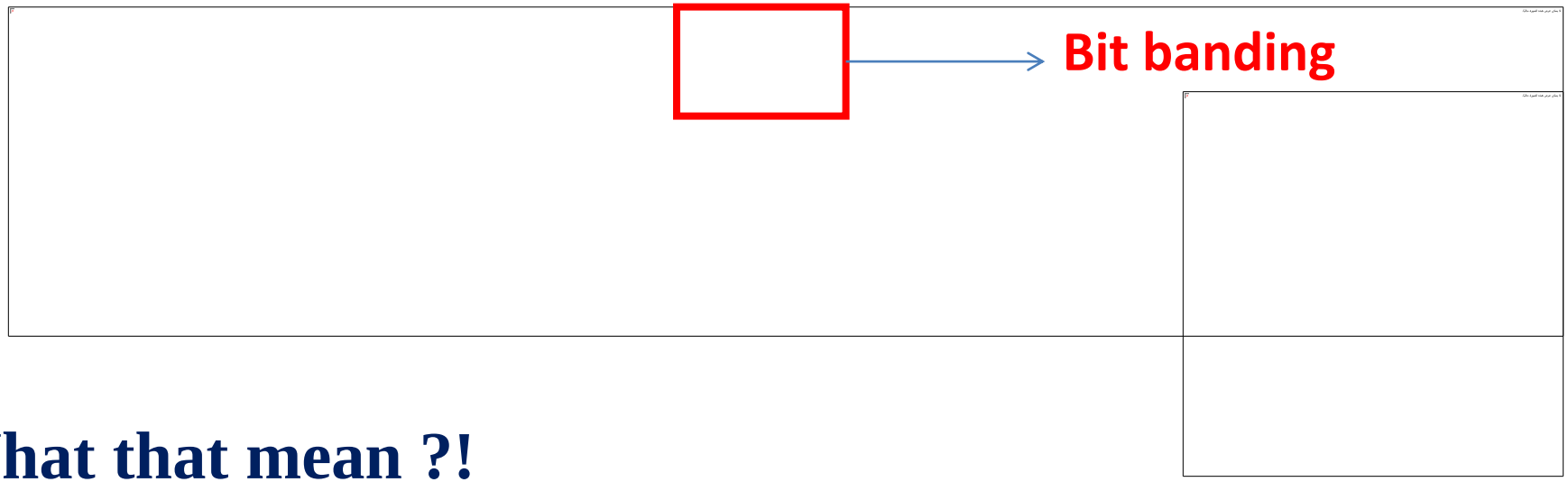
4- GPIODATA



If we wish to access bit	*Constant*
7	0x0200
6	0x0100
5	0x0080
4	0x0040
3	0x0020
2	0x0010
1	0x0008
0	0x0004

- **General-Purpose Input/Outputs**

4- GPIODATA



What that mean ?!

I can access to any pin directly without needing to **(read-modify-write operation to set or clear an individual GPIO pin.)**

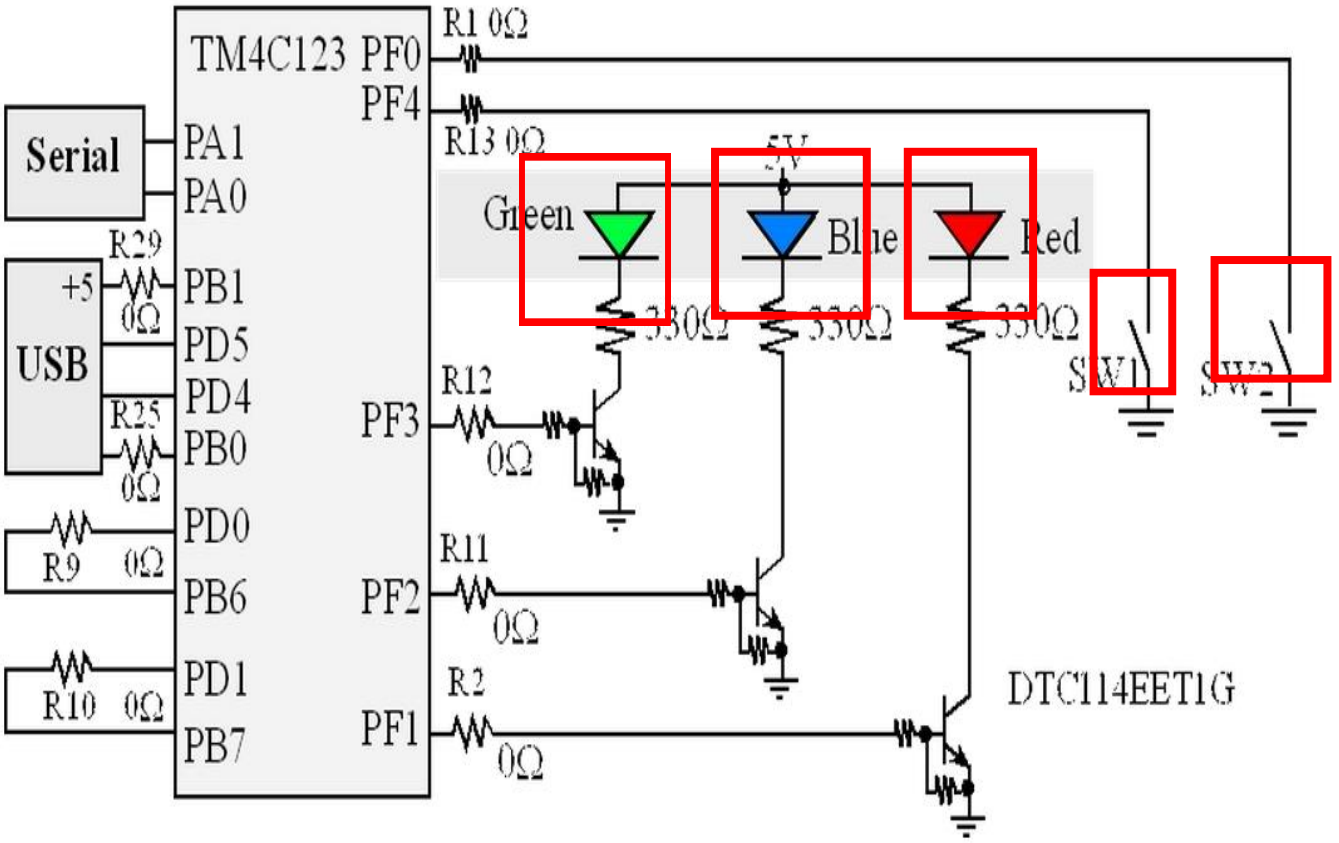
```
#define GPIO_PORTF_DATA (*((volatile unsigned long *)0x40025028))
```

GPIO_PORTF_DATA=0X0A

- **TM4C123GH6PM Overview:**

Second Program :

Blink Led red & Green &Blue using switch 1&2



- **TM4C123GH6PM Overview:**

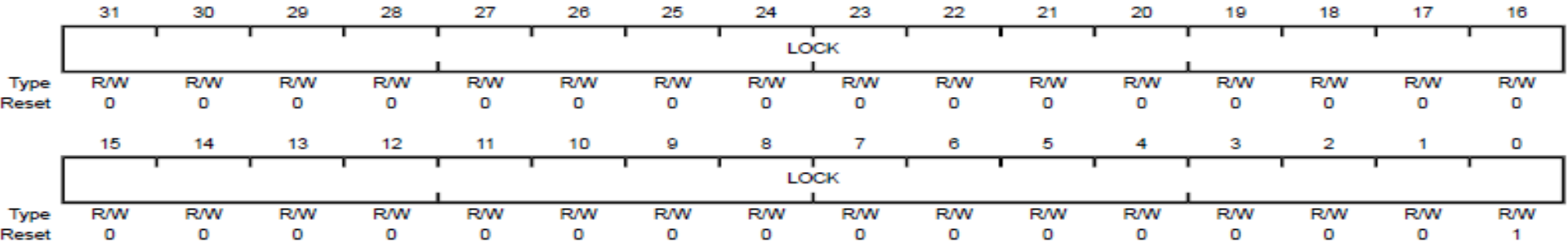
Second Program :

Blink Led red & Green &Blue using switch 1&2

1. **#define GPIO_PORTF_DATA (*((volatile unsigned long *)0x4002507C)) to acess pin 0 to pin 4**
2. **GPIO_PORTF_DIR=0X0E**
3. **GPIO_PORTF_DEN=0X0E;**
4. **After we deliver clock we will do the next**

- **General-Purpose Input/Outputs**

2- GPIOLOCK



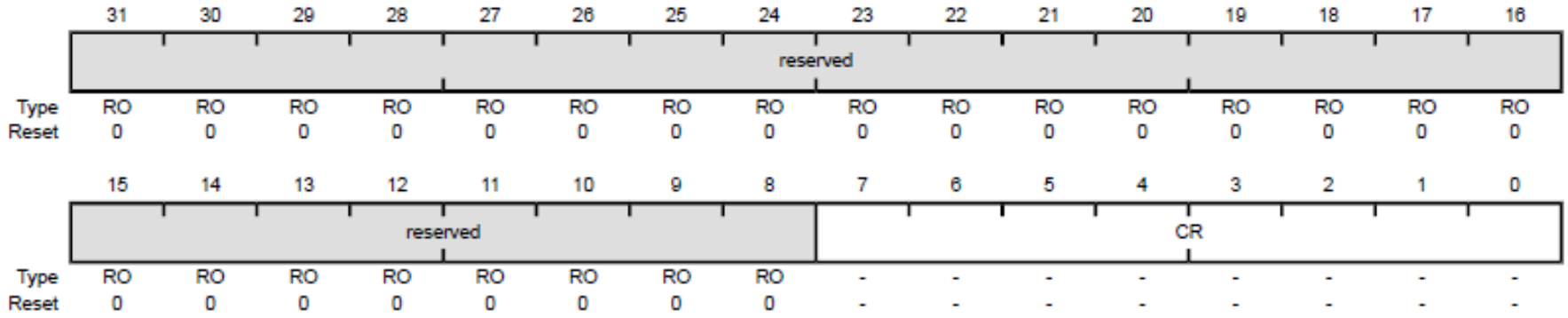
Bit/Field	Name	Type	Reset	Description
31:0	LOCK	R/W	0x0000.0001	GPIO Lock A write of the value 0x4C4F.434B unlocks the GPIO Commit (GPIOCR) register for write access.A write of any other value or a write to the GPIOCR register reapplies the lock, preventing any register updates. A read of this register returns the following values:

With Port C & Port F ,we need to write in GPIOLock Register 0x4C4F434B to can use all of pins of Port C &F , because some of this pin can make damage to the internal Flash memory & microcontroller

Writing 0x4C4F434B In **GPIOLock** enable to write In **GPIOCR**

- **General-Purpose Input/Outputs**

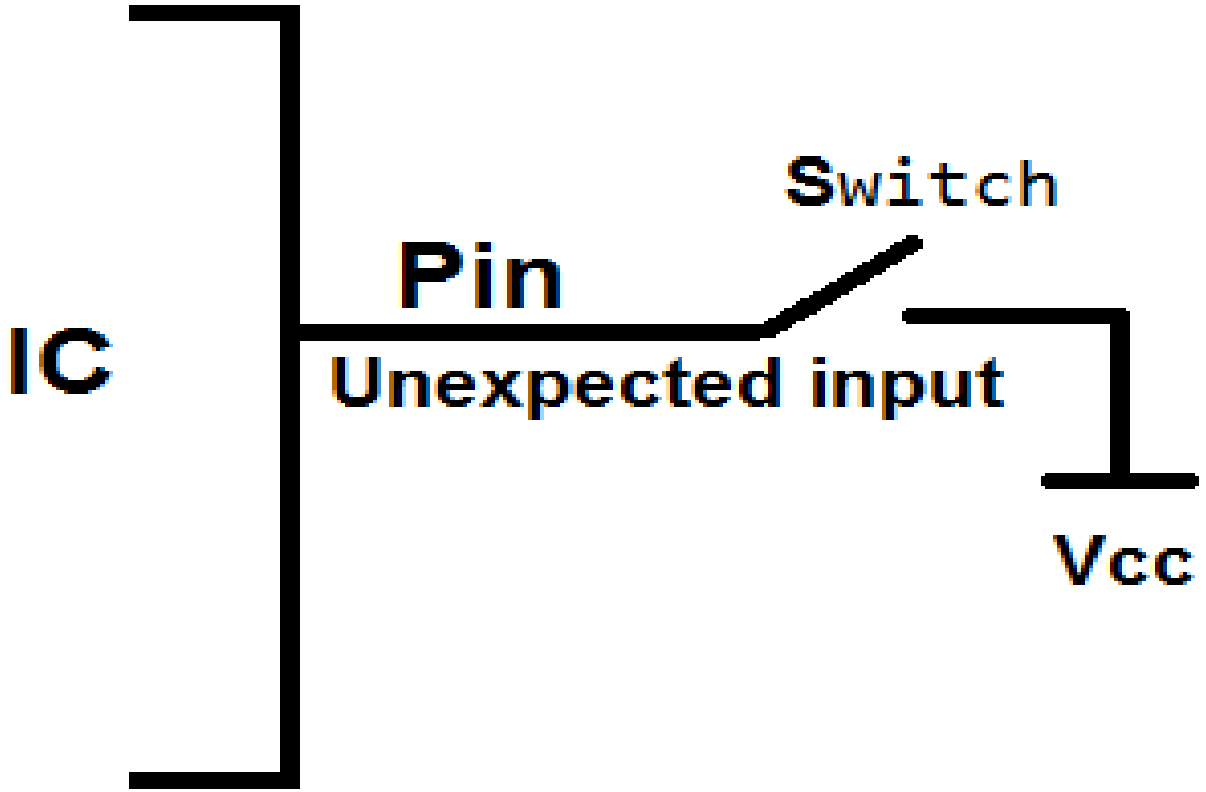
3- GPIOCR



This register is designed **to prevent accidental programming of the registers** that control connectivity to the NMI and JTAG/SWD debug hardware. By initializing the bits of the GPIOCR register to 0 for PD7, PF0, and PC[3:0].

- **General-Purpose Input/Outputs**

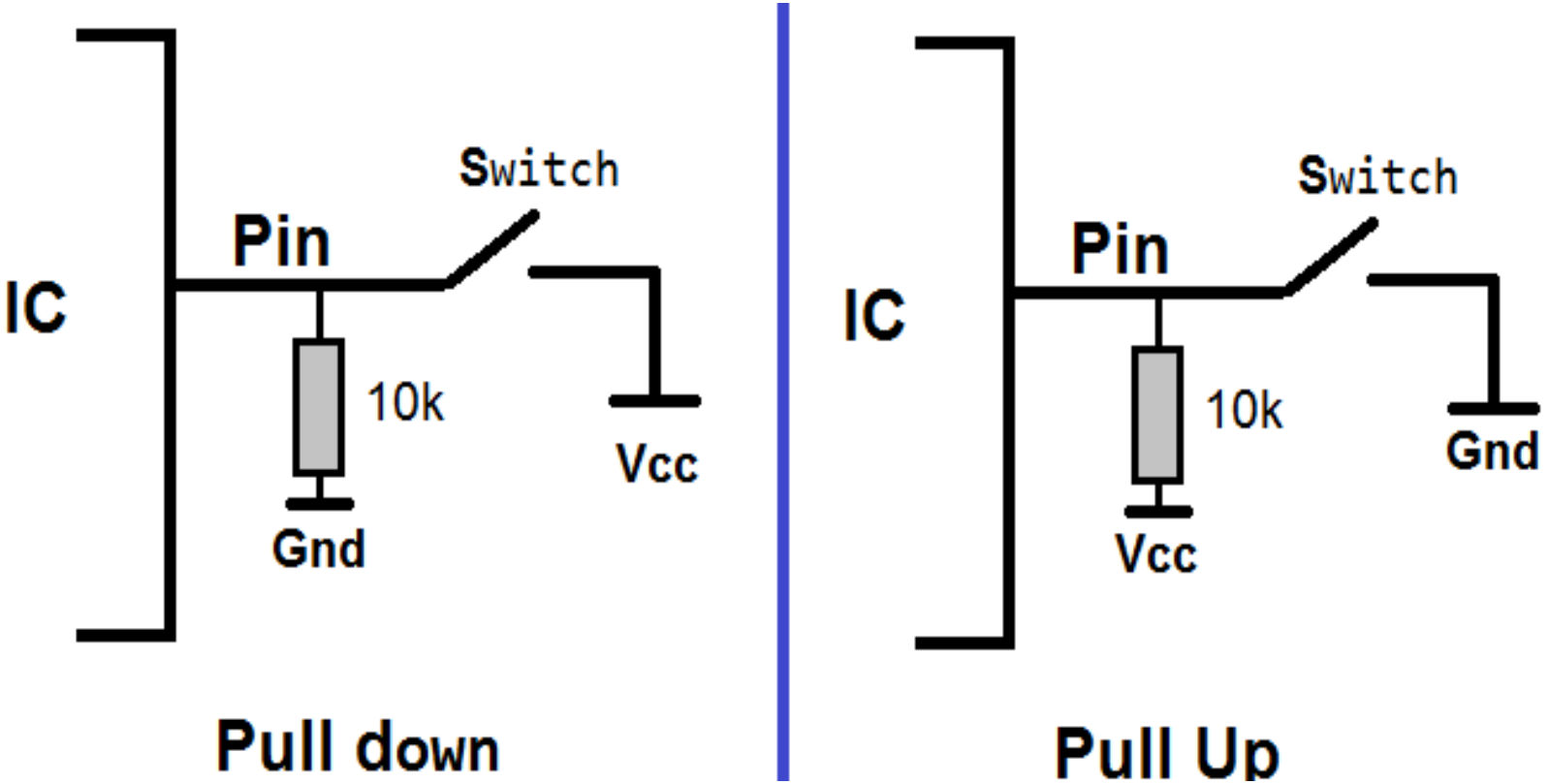
3- Un Excepted input



- **General-Purpose Input/Outputs**

3- Un Excepted input

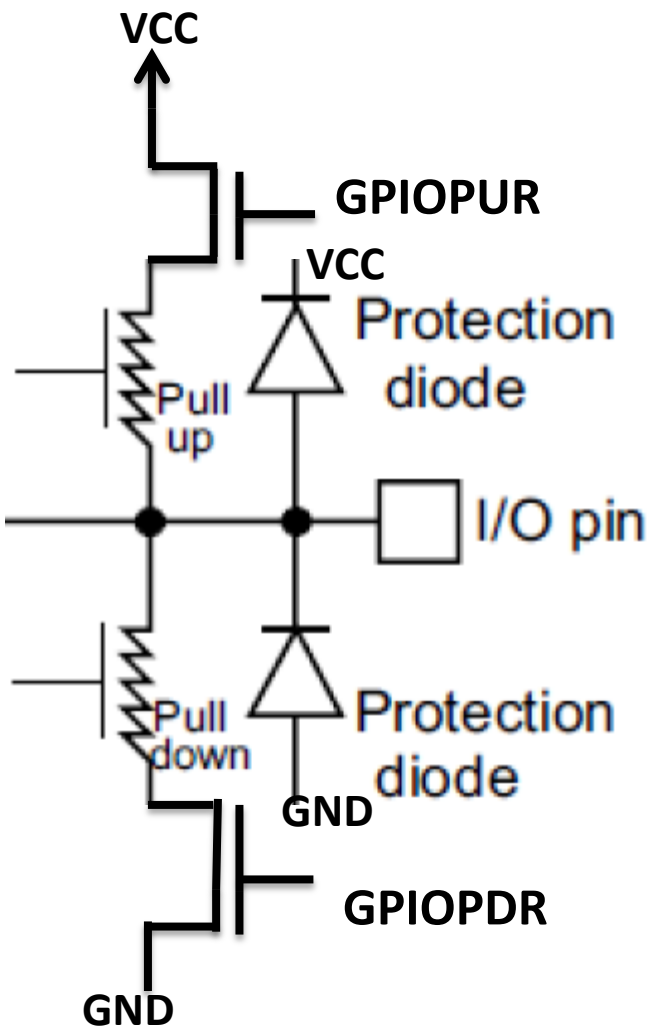
In tiva-c is available internal Pull up Resistor & internal pull down Resistor on every pin .
Switches on PF0,PF4 Is suitable For Pull up resistor



- **General-Purpose Input/Outputs**

3- GPIO Pull-Up Select (GPIOPUR)

GPIO Pull-down Select (GPIOPDR)



Agenda

- TM4C123 Overview
- General-Purpose Input/Outputs
- **Bitwise operators**
- System Timer (SysTick)
- Delay Library
- Interface 74595

- **Bitwise operators**

Symbol	operator
&	bitwise AND
|	bitwise inclusive OR
^	bitwise XOR (eXclusive OR)
<<	left shift
>>	right shift
~	bitwise NOT (one's complement)

- **Bitwise operators**
  - **OR**

    GPIO_PORTF_DATA=0X0A | 0x04

    0b00001010

    0b00000100=

    0b00001110 =0x0E

    GPIO_PORTF_DATA|=0x04 ☺
  - **Xor**

    GPIO_PORTF_DATA=0X0E^0x04

    0b00001110

    0b00000100=

    0b00001010 =0x0A

    GPIO_PORTF_DATA^=0x04 ☺

- **Bitwise operators**
  - **Not**

    GPIO_PORTF_DATA=0x0A

    GPIO_PORTF_DATA=~GPIO_PORTF_DATA

    0b00001010=

    0b11110101 =0xF5

    GPIO_PORTF_DATA~=0x0A ☺
  - **And or Mask**

    GPIO_PORTF_DATA=0X0E&0x04

    0b00001110

    0b00000100=

    0b00000100 =0x04

    GPIO_PORTF_DATA&=0x04 ☺

- **Bitwise operators**
  - **(>>) right shift**

    Data>>Number of bit

    GPIO_PORTF_DATA=0x0A>>1

    0b00001010=

    0b00000101 =0x05

    GPIO_PORTF_DATA>>=1 ☺
  - **(<<) left shift**

    Data<<Number of bit

    GPIO_PORTF_DATA=0x0A<<1

    0b00001010=

    0b000101000 =0x14

    GPIO_PORTF_DATA<<=1 ☺

- **Bitwise operators**

Exercise

```
GPIO_PORTF_DATA=0XFF;
GPIO_PORTF_DATA&=~(0X02);
```

Answer is GPIOF_PORTF_DATA=0XFD;

Agenda

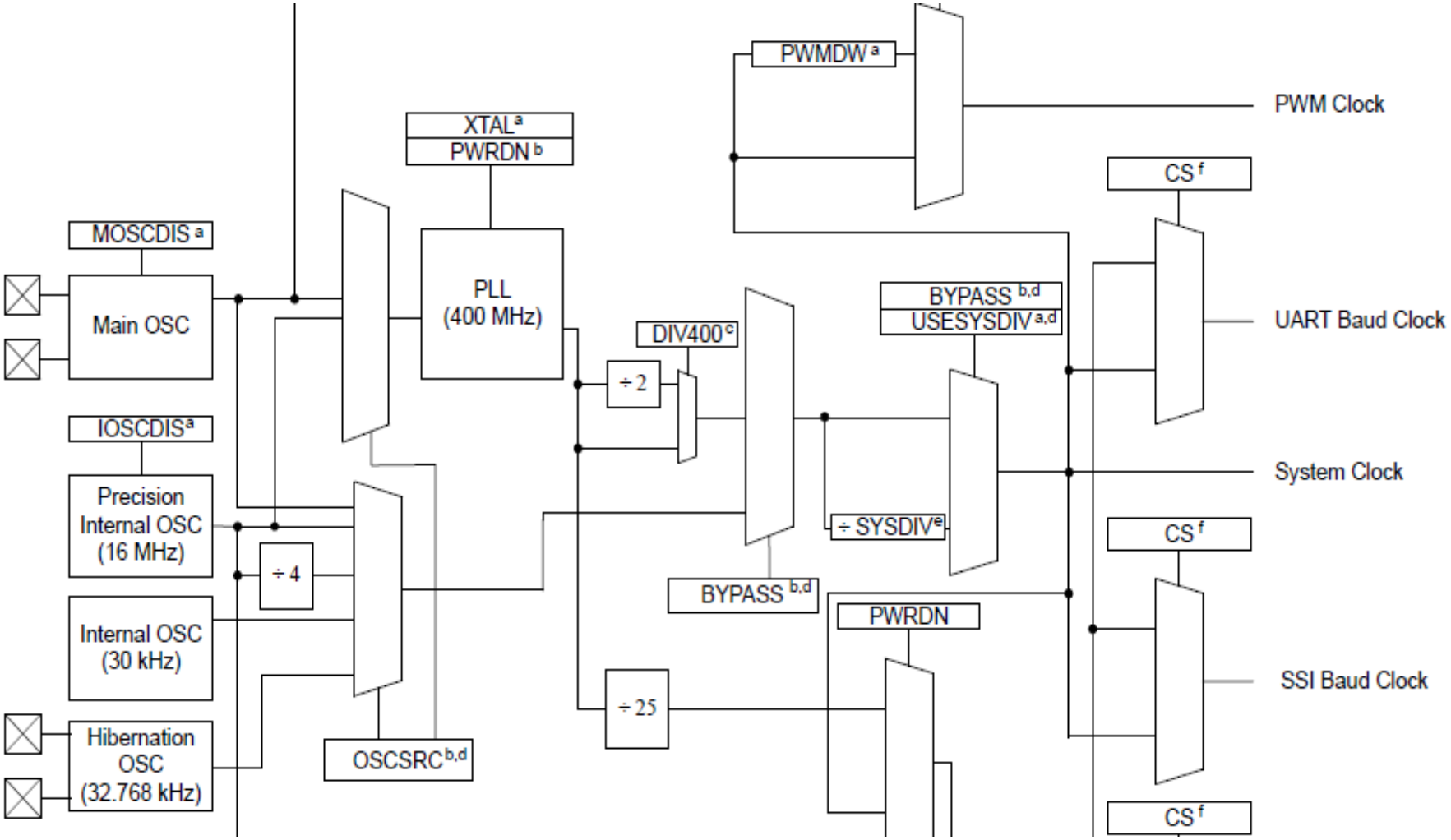
- TM4C123 Overview
- General-Purpose Input/Outputs
- Bitwise operators
- **System Timer (SysTick)**
- Delay Library
- Interface 74595

System Timer (SysTick)

- **Tiva-c includes an integrated system timer (SysTick) :**
  - SysTick is part of Cortex-M4 Core.
  - SysTick is 24 bit timer wide.
  - SysTick is use in RTOS operations .
  - SysTick is decrement timer.
  - 24 bit timer mean the timer can count 16777215 Cycle.
  - With **16 MHz** speed the SysTick can generate 1 second delay when count 16000000 cycles which mean **15999999 because timer count to 0 .**

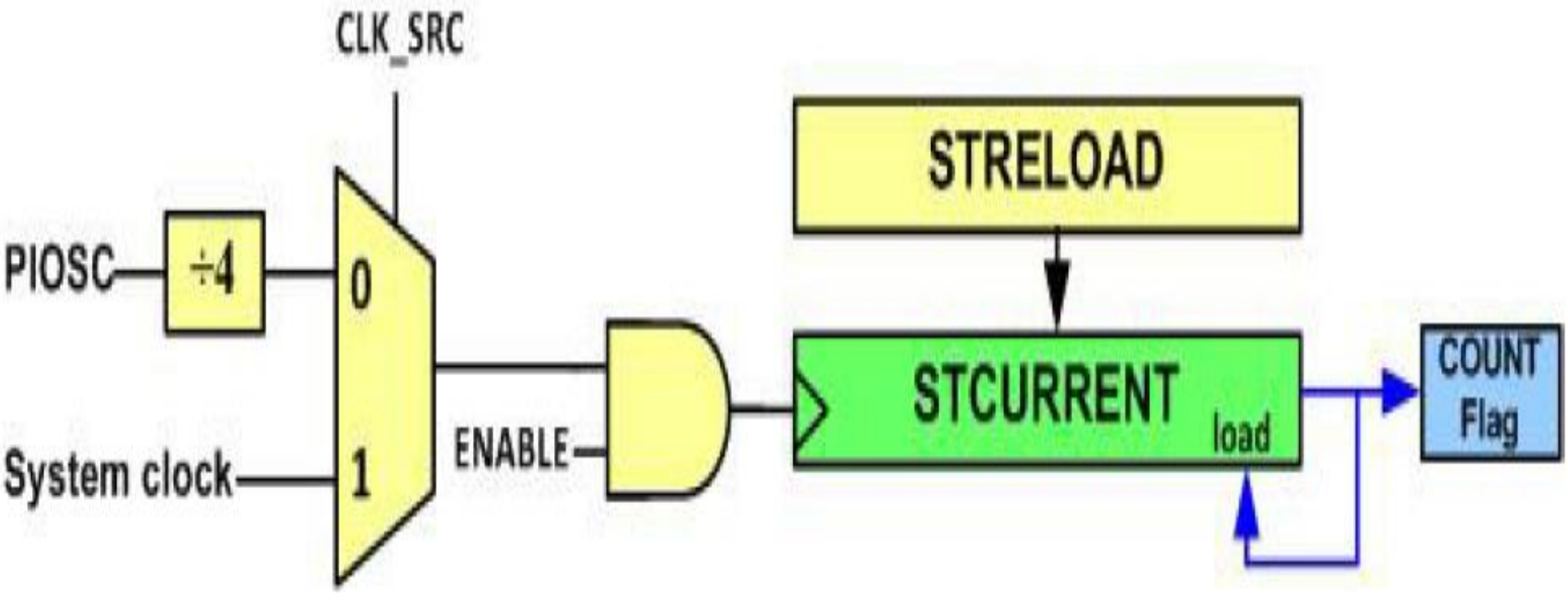
- **System Timer (SysTick)**

Clock Diagram In Tiva-c



- **System Timer (SysTick)**

SysTick diagram In Tiva-c



PIOSC (Precision Internal OSC 16 MHz) .

System Clock can be from 32,768 KHz To 80 MHz .

• System Timer (SysTick)

SysTick consist from 3 Register :

- **Register 1: SysTick Control and Status Register (STCTRL):**

SysTick Control and Status Register (STCTRL)

Base 0xE000.E000
Offset 0x010
Type R/W, reset 0x0000.0004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															COUNT
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved													CLK_SRC	INTEN	ENABLE
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

- **Enable : 0 timer is disabled , 1 timer is enabled .**
- **INTEN : 0 Interrupt is disabled , 1 An interrupt is generated to the NVIC when SysTick counts to 0.**
- **CLK_SRC : 0 (PIOSC) divided by 4 ,1 System clock.**
- **COUNT : 0 The SysTick timer has not counted to 0 yet , 1 The SysTick timer has counted to 0 . This bit is cleared by a read of the register or if the STCURRENT register is written with any value.**

• System Timer (SysTick)

SysTick consist from 3 Register :

- **Register 2: SysTick Reload Value Register (STRELOAD)**

SysTick Reload Value Register (STRELOAD)

Base 0xE000.E000
Offset 0x014
Type R/W, reset -

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved								RELOAD							
Type	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RELOAD															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- **The STRELOAD register specifies the start value to load into the SysTick Current Value (STCURRENT) register when the counter reaches.**
- **The start value can be between 0x1 and 0x00FFFFFF.**
- **The STRELOAD should contain the value N – 1 for the COUNT to fire every N clock cycles because the counter counts down to 0. For example, if we need 1000 clocks of interval, then we make STRELOAD =999 .**

• System Timer (SysTick)

SysTick consist from 3 Register :

- **Register 3: SysTick Current Value Register (STCURRENT)**

SysTick Current Value Register (STCURRENT)

Base 0xE000.E000
Offset 0x018
Type R/WC, reset -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved								CURRENT							
Type	RO	RO	RO	RO	RO	RO	RO	RO	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

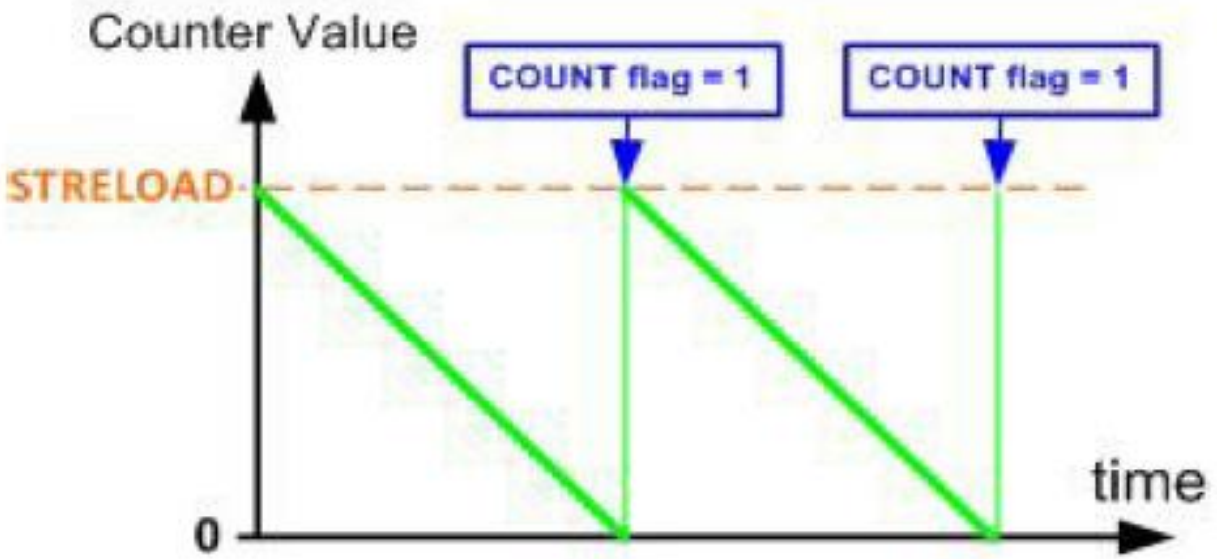
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CURRENT															
Type	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- **This register is write-clear. Writing to it with any value clears the register. Clearing this register also clears the COUNT bit of the STCTRL register .**

- **System Timer (SysTick)**

Working with SysTick

I. We **Must** Disable SysTick during setup , STCTRL=0.
II. We put the value that SysTick count it in STRELOAD.
III. The value must be not more than 16777215 or FFFFFF .
IV. We Enable SysTick & choose the configurations.
V. STCTRL=0x05 (enable it, no interrupt, use system clock).
VI. Wait Flag



Agenda

- TM4C123 Overview
- General-Purpose Input/Outputs
- Bitwise operators
- System Timer (SysTick)
- **Delay Library**
- Interface 74595

Agenda

- TM4C123 Overview
- General-Purpose Input/Outputs
- Bitwise operators
- System Timer (SysTick)
- Delay Library
- **Interface 74595**

- **Interface 74595**