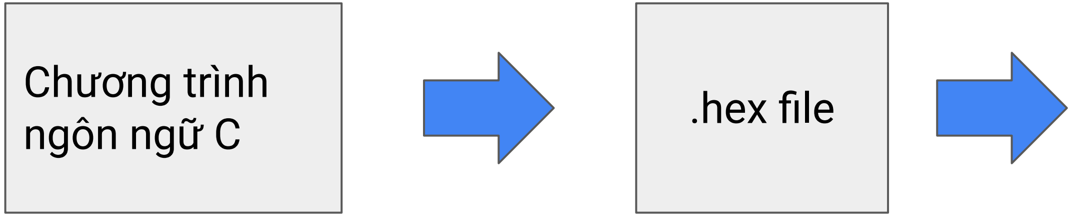
Bài 1: Setup project đầu tiên trên KeilC

Võ Thành Danh

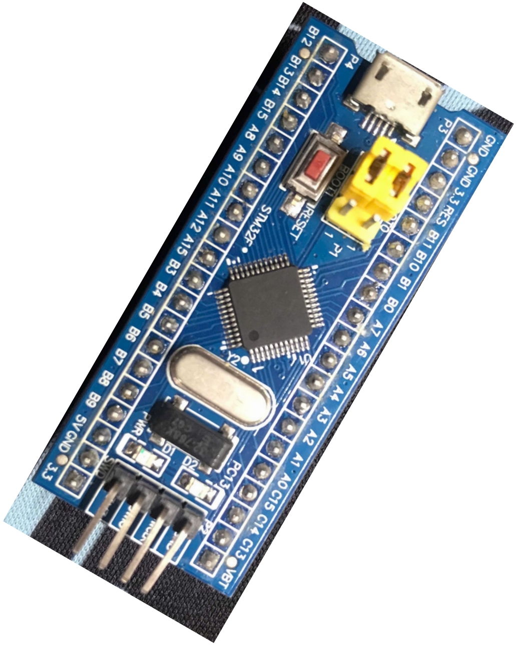
1. Keil C

KEIL™ Tools by ARM

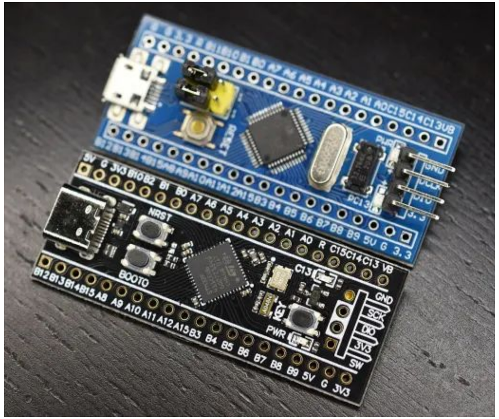


Chương trình ngôn ngữ C

.hex file



2. Tạo project đầu tiên với KeilC



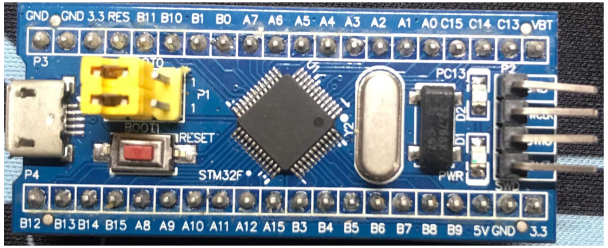
ST-Link driver: https://www.st.com/en/development-tools/stsw-link009.html

Thư viện chuẩn của MCU cho Keil: https://www.keil.arm.com/devices/

Datasheet, Reference Manual của MCU

3. Blink LED PC13

- Cấp xung clock cho ngoại vi
- Cấu hình chân của ngoại vi
- Sử dụng ngoại vi


GND GND 3.3 RES B11 B10 B1 B0 A7 A6 A5 A4 A3 A2 A1 A0 C15 C14 C13 VBT
P3
P1
BOOT1
RESET
PC13
P2
D2
D1
PWR
SWD
P4
STM32F
B12 B13 B14 B15 A8 A9 A10 A11 A12 A15 B3 B4 B5 B6 B7 B8 B9 5V GND 3.3

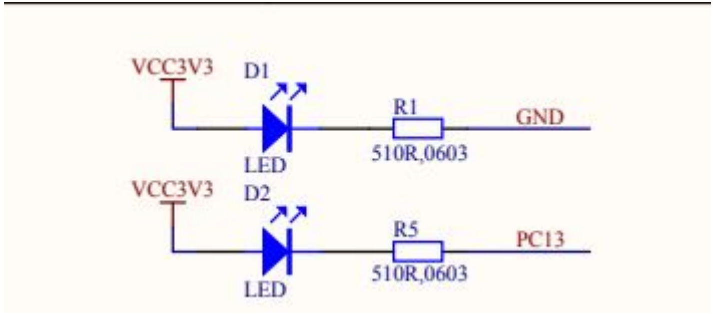
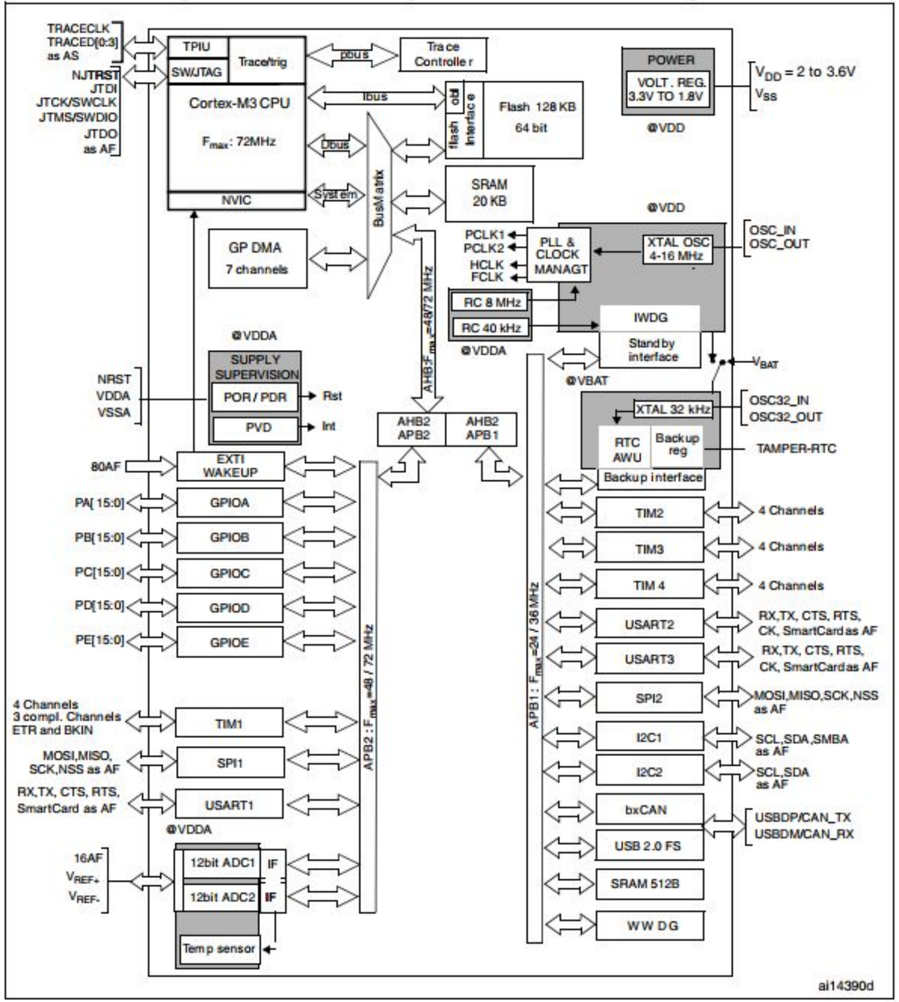


Figure 1. STM32F103xx performance line block diagram



3.1 Địa chỉ các thanh ghi

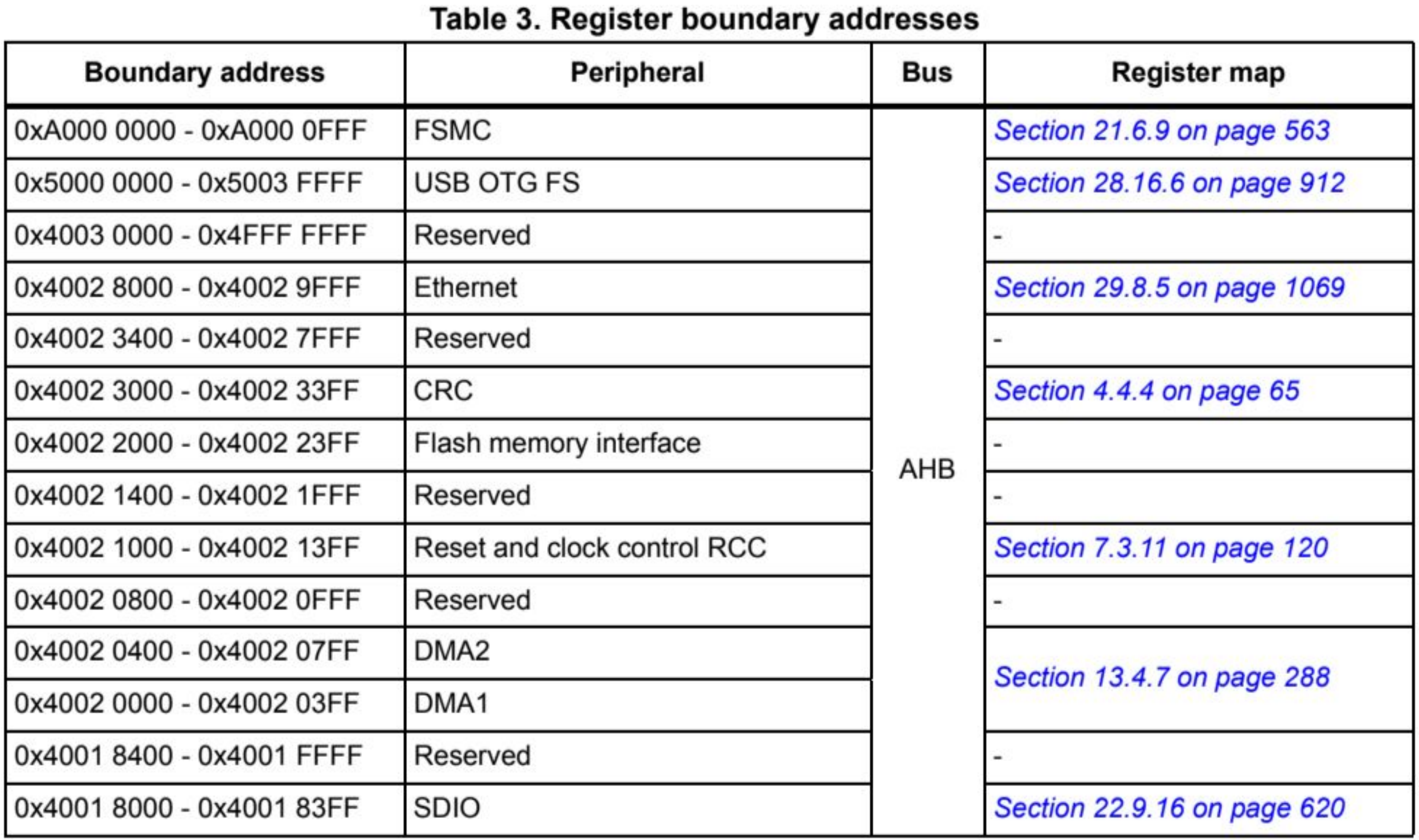
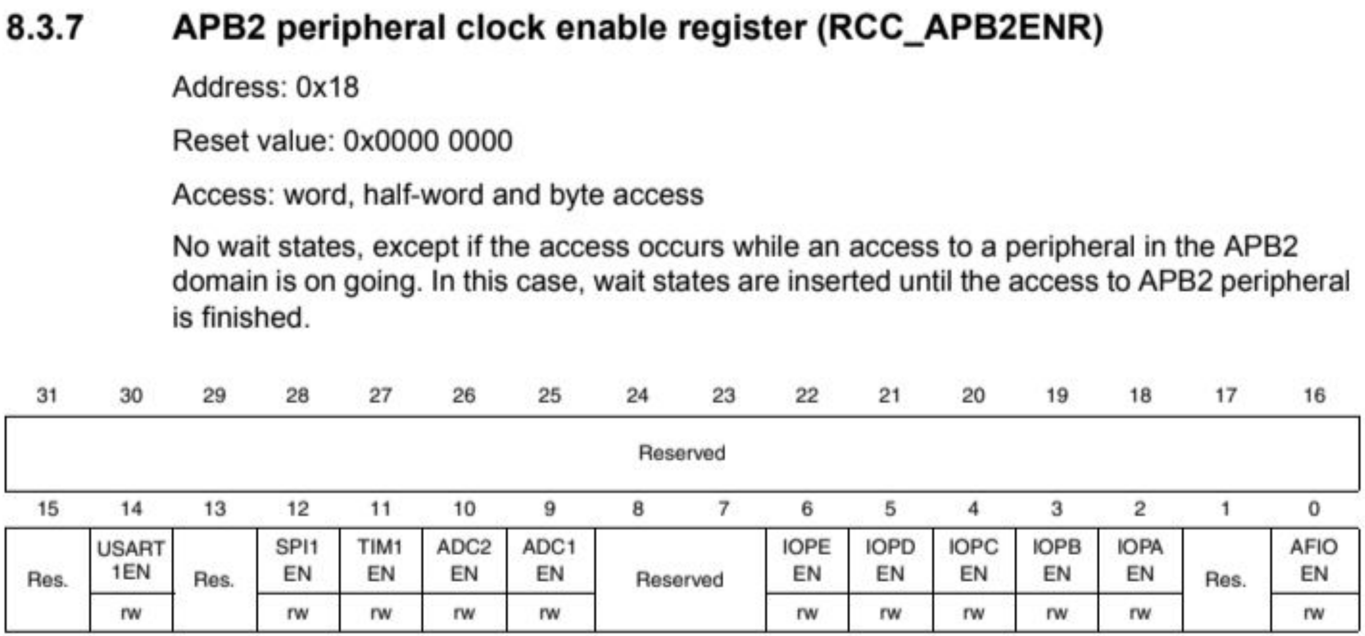


Table 3. Register boundary addresses

Boundary address	Peripheral	Bus	Register map
0xA000 0000 - 0xA000 0FFF	FSMC	AHB	*Section 21.6.9 on page 563*
0x5000 0000 - 0x5003 FFFF	USB OTG FS		*Section 28.16.6 on page 912*
0x4003 0000 - 0x4FFF FFFF	Reserved		-
0x4002 8000 - 0x4002 9FFF	Ethernet		*Section 29.8.5 on page 1069*
0x4002 3400 - 0x4002 7FFF	Reserved		-
0x4002 3000 - 0x4002 33FF	CRC		*Section 4.4.4 on page 65*
0x4002 2000 - 0x4002 23FF	Flash memory interface		-
0x4002 1400 - 0x4002 1FFF	Reserved		-
0x4002 1000 - 0x4002 13FF	Reset and clock control RCC		*Section 7.3.11 on page 120*
0x4002 0800 - 0x4002 0FFF	Reserved		-
0x4002 0400 - 0x4002 07FF	DMA2		*Section 13.4.7 on page 288*
0x4002 0000 - 0x4002 03FF	DMA1		
0x4001 8400 - 0x4001 FFFF	Reserved		-
0x4001 8000 - 0x4001 83FF	SDIO		*Section 22.9.16 on page 620*

3.2 Cấp clock cho ngoại vi



8.3.7 APB2 peripheral clock enable register (RCC_APB2ENR)

Address: 0x18

Reset value: 0x0000 0000

Access: word, half-word and byte access

No wait states, except if the access occurs while an access to a peripheral in the APB2 domain is on going. In this case, wait states are inserted until the access to APB2 peripheral is finished.

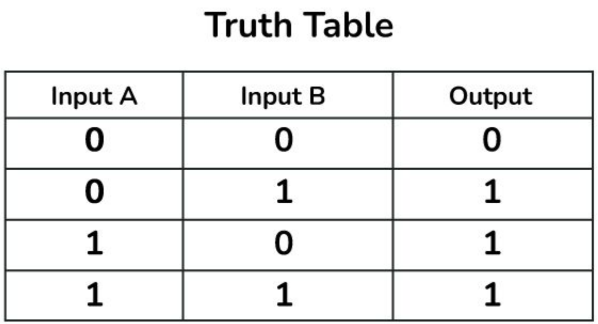
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART 1EN	Res.	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	Reserved		IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.	AFIO EN
	rw		rw	rw	rw	rw			rw	rw	rw	rw	rw		rw

```
#define RCC_APB2ENR*((unsigned int *)0x40021018)
```

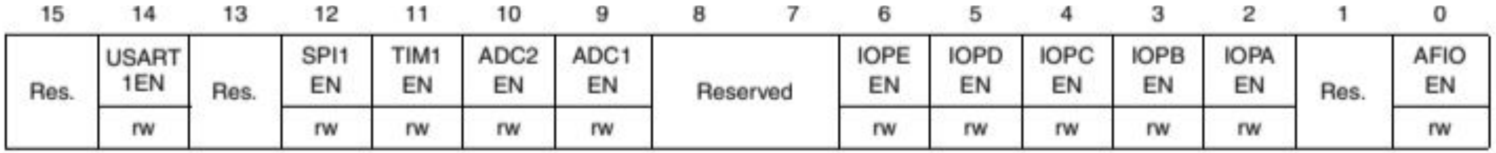
3.2 Cấp clock cho ngoại vi

a |= b ⇔ a = a | b



Truth Table

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	1

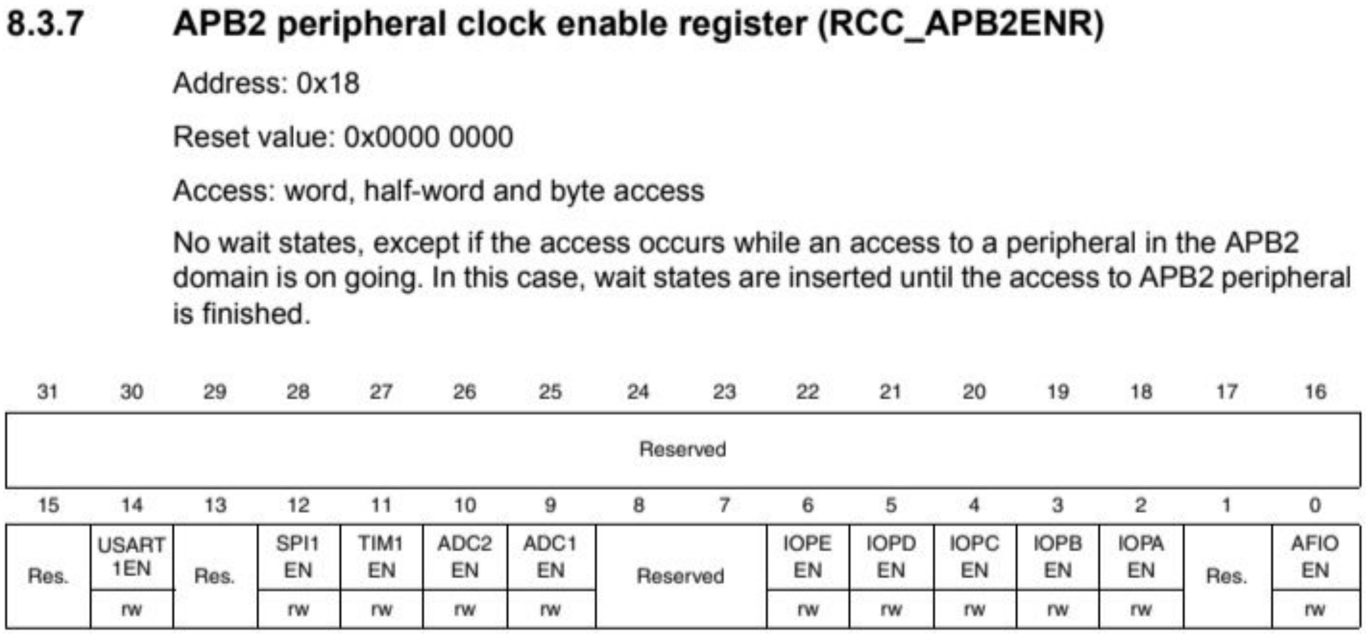


15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART 1EN	Res.	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	Reserved		IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.	AFIO EN
	rw		rw	rw	rw	rw			rw	rw	rw	rw	rw		rw

a	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

a | b	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0

3.2 Cấp clock cho ngoại vi



8.3.7 APB2 peripheral clock enable register (RCC_APB2ENR)

Address: 0x18

Reset value: 0x0000 0000

Access: word, half-word and byte access

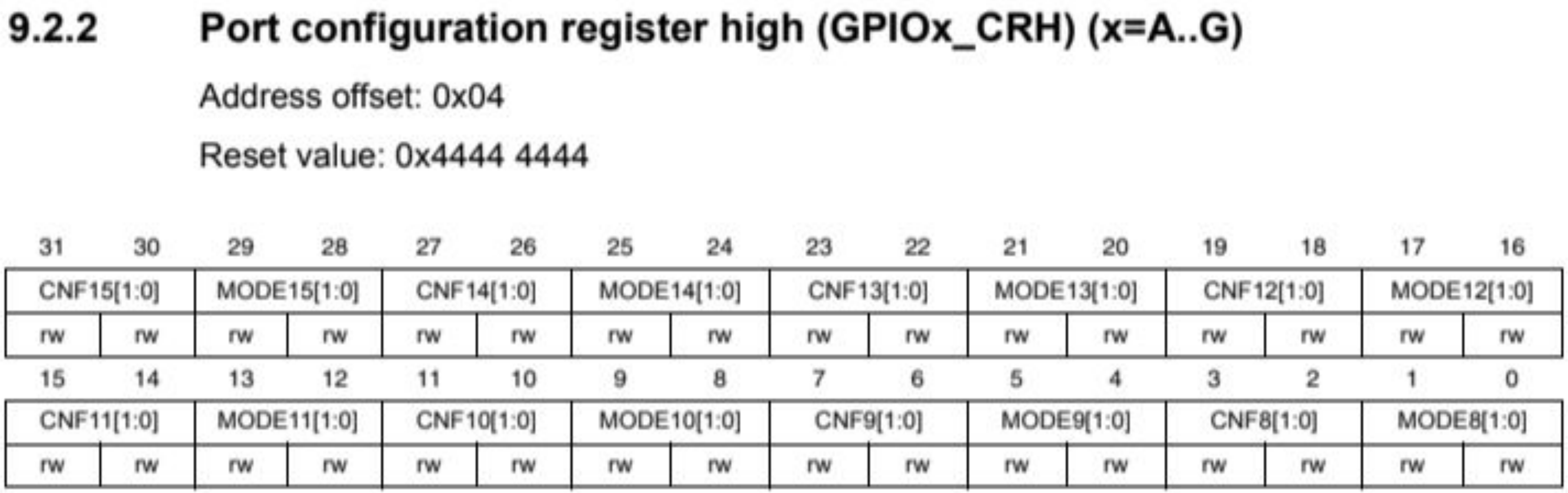
No wait states, except if the access occurs while an access to a peripheral in the APB2 domain is on going. In this case, wait states are inserted until the access to APB2 peripheral is finished.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1EN	Res.	SPI1EN	TIM1EN	ADC2EN	ADC1EN	Reserved		IOPEEN	IOPDEN	IOPCEN	IOPBEN	IOPAEN	Res.	AFIOEN
	rw		rw	rw	rw	rw			rw	rw	rw	rw	rw		rw

```
RCC_APB2ENR |= (1 << 4); // Kich hoat xung clock cap cho GPIOC
```

3.3 Cấu hình chế độ chân



9.2.2 Port configuration register high (GPIOx_CRH) (x=A..G)

Address offset: 0x04

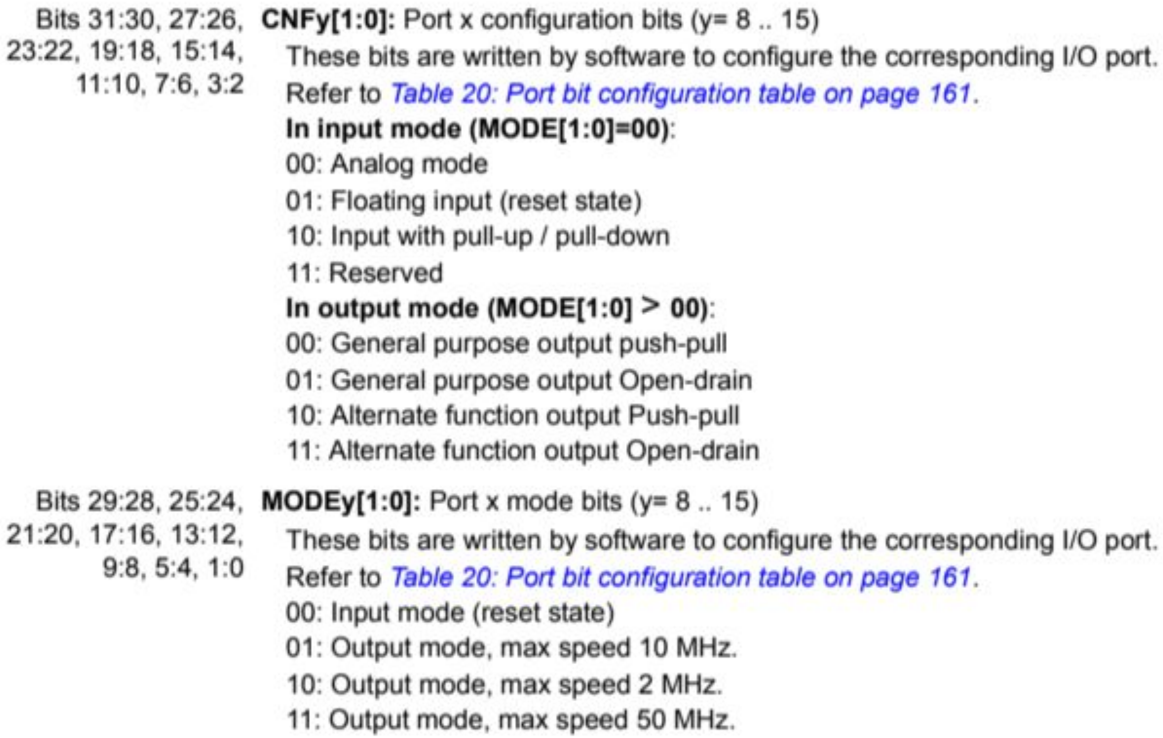
Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15[1:0]		MODE15[1:0]		CNF14[1:0]		MODE14[1:0]		CNF13[1:0]		MODE13[1:0]		CNF12[1:0]		MODE12[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11[1:0]		MODE11[1:0]		CNF10[1:0]		MODE10[1:0]		CNF9[1:0]		MODE9[1:0]		CNF8[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

```
#define GPIOC_CRH       *((unsigned int *)0x40011004)
```

3.3 Cấu hình chế độ chân



Bits 31:30, 27:26, 23:22, 19:18, 15:14, 11:10, 7:6, 3:2

CNFy[1:0]: Port x configuration bits (y= 8 .. 15)

These bits are written by software to configure the corresponding I/O port.

Refer to *Table 20: Port bit configuration table on page 161*.

In input mode (MODE[1:0]=00):

00: Analog mode

01: Floating input (reset state)

10: Input with pull-up / pull-down

11: Reserved

In output mode (MODE[1:0] > 00):

00: General purpose output push-pull

01: General purpose output Open-drain

10: Alternate function output Push-pull

11: Alternate function output Open-drain

Bits 29:28, 25:24, 21:20, 17:16, 13:12, 9:8, 5:4, 1:0

MODEy[1:0]: Port x mode bits (y= 8 .. 15)

These bits are written by software to configure the corresponding I/O port.

Refer to *Table 20: Port bit configuration table on page 161*.

00: Input mode (reset state)

01: Output mode, max speed 10 MHz.

10: Output mode, max speed 2 MHz.

11: Output mode, max speed 50 MHz.

3.3 Cấu hình chế độ chân

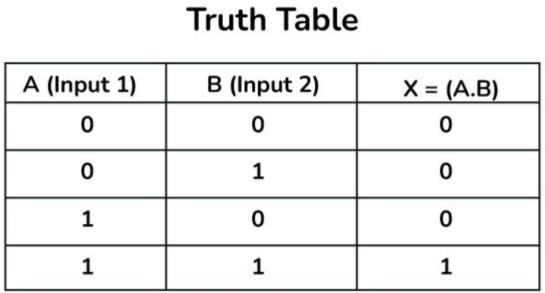
```
// MODE13[1:0] = 11: Output mode, max speed 50 MHz

GPIOC_CRH |= (1 << 20) | (1 << 21);

// CNF13[1:0] = 00: General purpose output
push-pull
GPIOC_CRH &= ~((1 << 22) | (1 << 23));
```

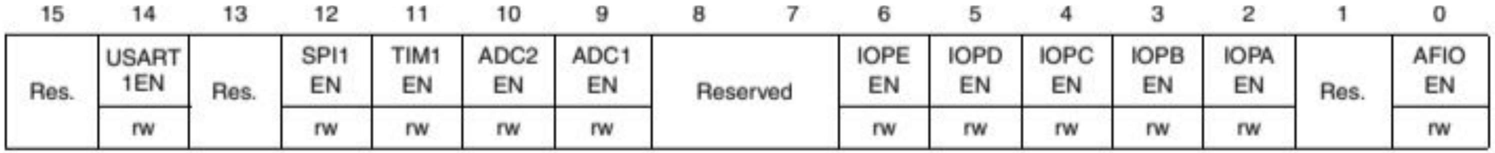
3.3 Cấu hình chế độ chân

a &= b ⇔ a = a & b



Truth Table

A (Input 1)	B (Input 2)	X = (A.B)
0	0	0
0	1	0
1	0	0
1	1	1

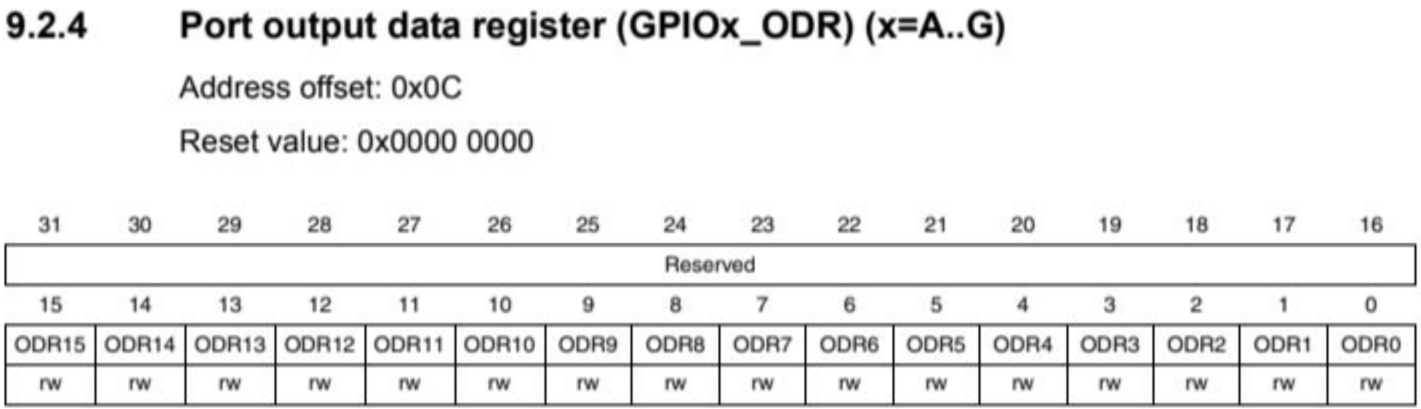


15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1EN	Res.	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	Reserved		IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.	AFIO EN
	rw		rw	rw	rw	rw			rw	rw	rw	rw	rw		rw

a	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
~b	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

a & ~b	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

3.4 Sử dụng ngoại vi



9.2.4 Port output data register (GPIOx_ODR) (x=A..G)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

```
while(1){
    GPIOC_ODR |= 1 << 13; // LED tắt
    delay(1000000);
    GPIOC_ODR &= ~(1 << 13); // LED sáng
    delay(1000000);
}
```

3.4 Sử dụng ngoại vi

Hàm delay được tạo bằng cách cho MCU không làm gì trong 1 khoảng thời gian bằng các vòng lặp

```
void delay(unsigned int timedelay){
    for(unsigned int i = 0; i < timedelay; i++){}
}
```

3.5 Xây dựng cấu trúc thanh ghi của các ngoại vi

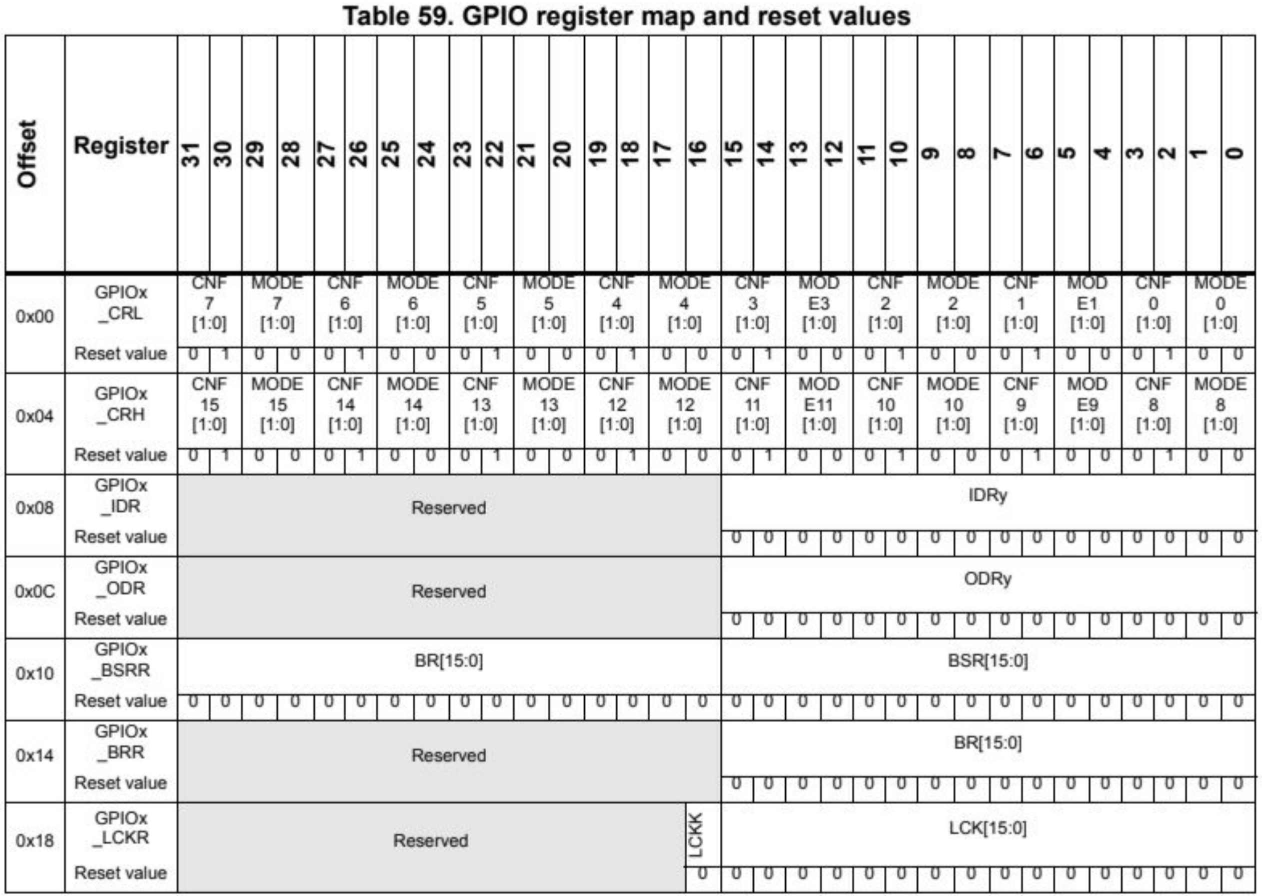


Table 59. GPIO register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	GPIOx_CRL	CNF 7 [1:0]		MODE 7 [1:0]		CNF 6 [1:0]		MODE 6 [1:0]		CNF 5 [1:0]		MODE 5 [1:0]		CNF 4 [1:0]		MODE 4 [1:0]		CNF 3 [1:0]		MOD E3 [1:0]		CNF 2 [1:0]		MODE 2 [1:0]		CNF 1 [1:0]		MOD E1 [1:0]		CNF 0 [1:0]		MODE 0 [1:0]	
	Reset value	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0x04	GPIOx_CRH	CNF 15 [1:0]		MODE 15 [1:0]		CNF 14 [1:0]		MODE 14 [1:0]		CNF 13 [1:0]		MODE 13 [1:0]		CNF 12 [1:0]		MODE 12 [1:0]		CNF 11 [1:0]		MOD E11 [1:0]		CNF 10 [1:0]		MODE 10 [1:0]		CNF 9 [1:0]		MOD E9 [1:0]		CNF 8 [1:0]		MODE 8 [1:0]	
	Reset value	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0x08	GPIOx_IDR	Reserved																IDRy															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	GPIOx_ODR	Reserved																ODRy															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	GPIOx_BSRR	BR[15:0]																BSR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	GPIOx_BRR	Reserved																BR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	GPIOx_LCKR	Reserved															LCKK	LCK[15:0]															
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3.5 Xây dựng cấu trúc thanh ghi của các ngoại vi

```
typedef struct
{
  unsigned int CRL;
  unsigned int CRH;
  unsigned int IDR;
  unsigned int ODR;
  unsigned int BSRR;
  unsigned int BRR;
  unsigned int LCKR;
} GPIO_TypeDef;
```

```
typedef struct
{
  unsigned int CR;
  unsigned int CFGR;
  unsigned int CIR;
  unsigned int APB2RSTR;
  unsigned int APB1RSTR;
  unsigned int AHBENR;
  unsigned int APB2ENR;
  unsigned int APB1ENR;
  unsigned int BDCR;
  unsigned int CSR;
} RCC_TypeDef;
```

4. Tổng kết và mở rộng

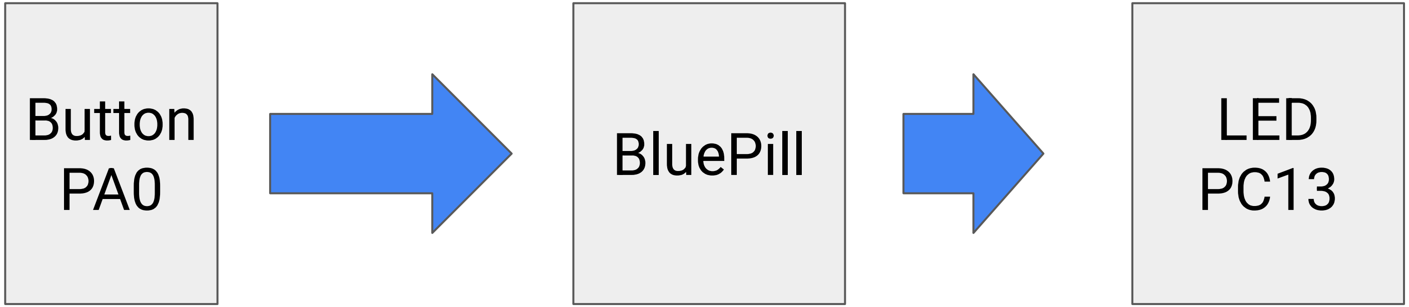
- Việc code trên thanh ghi nhằm giúp các bạn hiểu rõ cách hoạt động chi tiết của từng ngoại vi, cũng như tăng hiệu suất của chương trình.
- Tuy nhiên, việc lập trình thanh ghi có thể trở nên khá phức tạp

```
void WritePin(GPIO_TypeDef *GPIO_Port, uint8_t Pin,
uint8_t state)
{
    if(state == HIGH)
    GPIO_Port->ODR |= (1 << Pin);
    else
    GPIO_Port->ODR &= ~(1 << Pin);
}
```

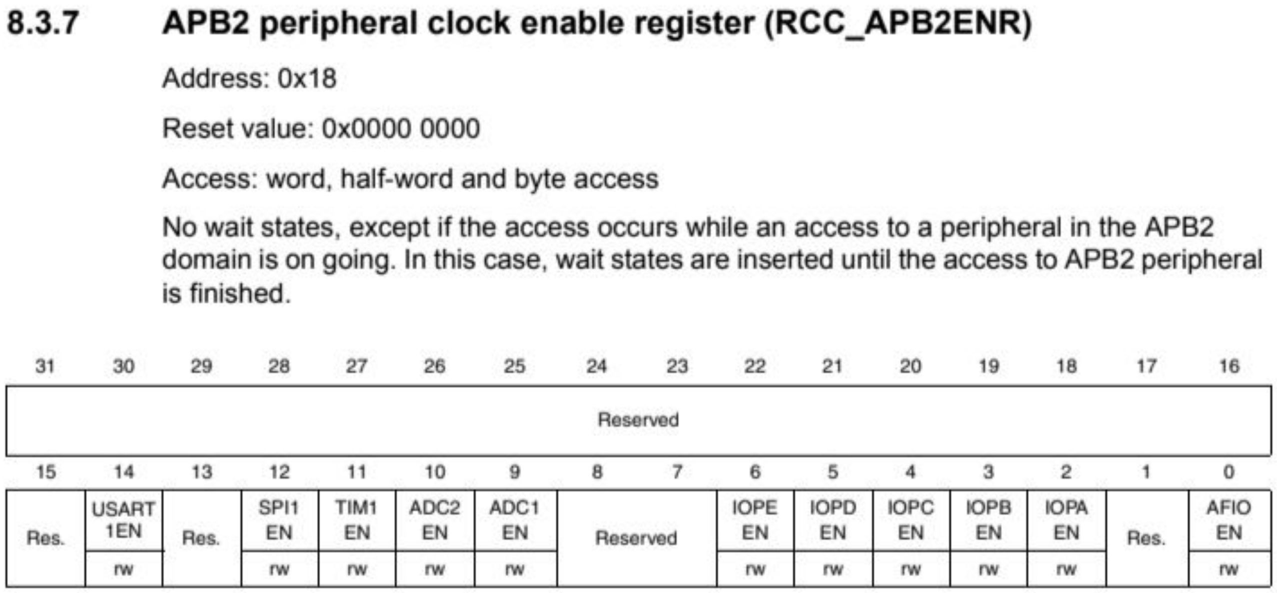
4. Tổng kết và mở rộng

```
void GPIO_Config(void){
    GPIOC->CRH |= GPIO_CRH_MODE13_0;  //MODE13[1:0] = 11: Output mode,
max speed 50 MHz
    GPIOC->CRH |= GPIO_CRH_MODE13_1;
    GPIOC->CRH &= ~GPIO_CRH_CNF13_0;              //CNF13[1:0] = 00:
General purpose output push-pull
    GPIOC->CRH &= ~GPIO_CRH_CNF13_1;
}
```

5. Đọc trạng thái nút nhấn



5.1. Cấp clock cho ngoại vi



8.3.7 APB2 peripheral clock enable register (RCC_APB2ENR)

Address: 0x18

Reset value: 0x0000 0000

Access: word, half-word and byte access

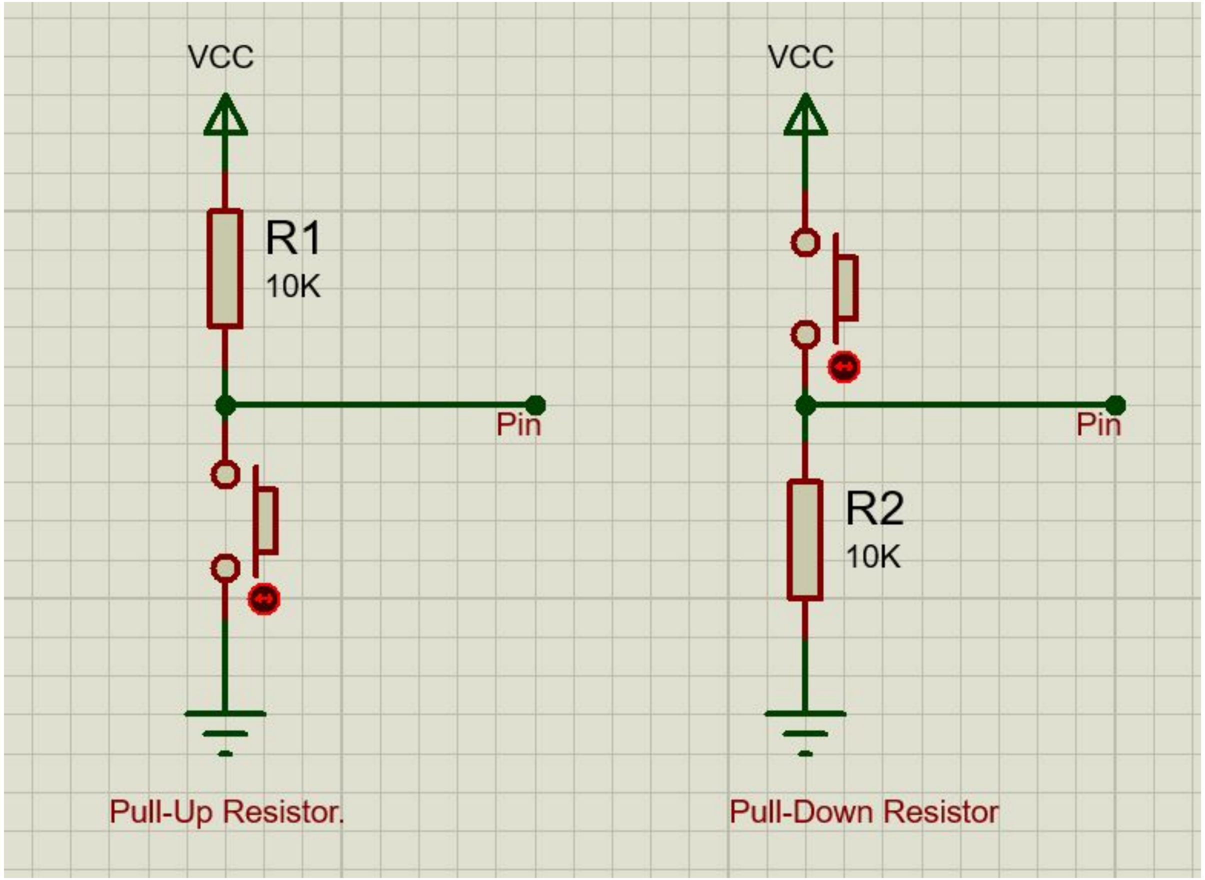
No wait states, except if the access occurs while an access to a peripheral in the APB2 domain is on going. In this case, wait states are inserted until the access to APB2 peripheral is finished.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

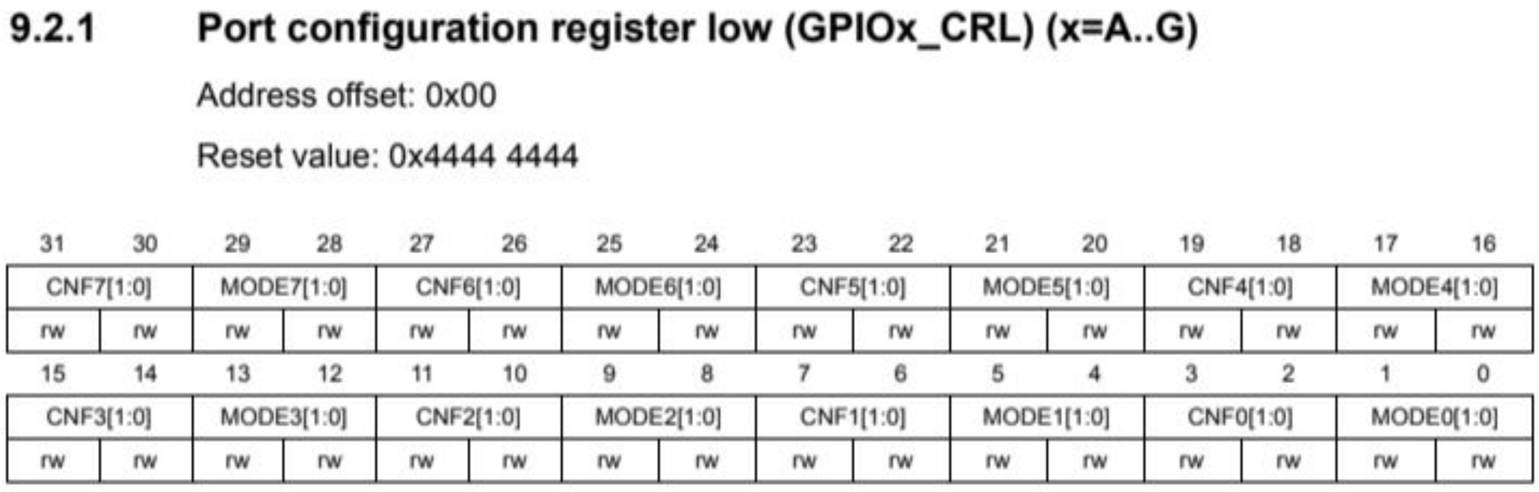
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART 1EN	Res.	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	Reserved		IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.	AFIO EN
	rw		rw	rw	rw	rw			rw	rw	rw	rw	rw		rw

```
RCC->APB2ENR |= (1 << 2) | (1 << 4); //Kich hoat xung clock cap cho GPIOA và GPIOC
```

5.2. Cấu hình chế độ chân



5.2. Cấu hình chế độ chân



9.2.1 Port configuration register low (GPIOx_CRL) (x=A..G)

Address offset: 0x00

Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		MODE7[1:0]		CNF6[1:0]		MODE6[1:0]		CNF5[1:0]		MODE5[1:0]		CNF4[1:0]		MODE4[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]		MODE3[1:0]		CNF2[1:0]		MODE2[1:0]		CNF1[1:0]		MODE1[1:0]		CNF0[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

5.2. Cấu hình chế độ chân

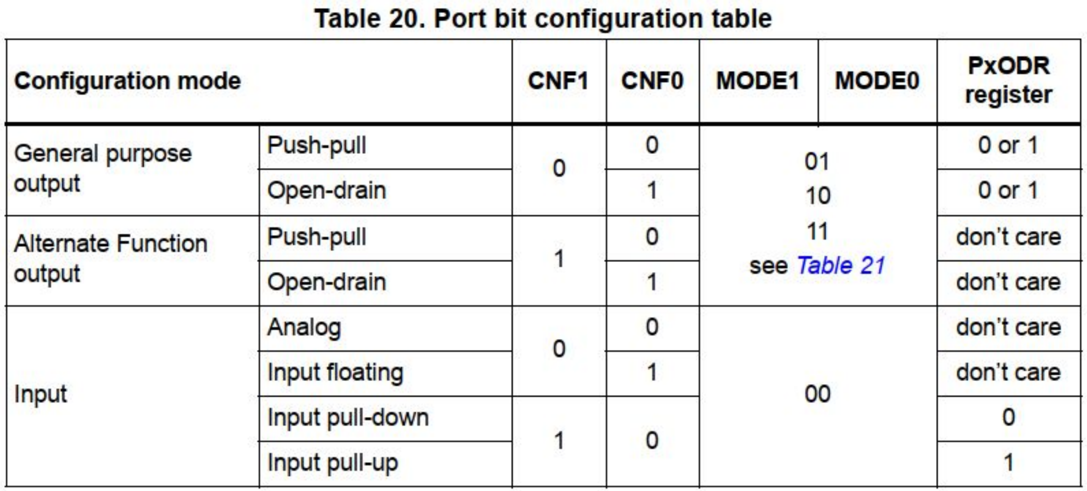


Table 20. Port bit configuration table

Configuration mode		CNF1	CNF0	MODE1	MODE0	PxODR register
General purpose output	Push-pull	0	0	01 10 11 see *Table 21*		0 or 1
	Open-drain		1			0 or 1
Alternate Function output	Push-pull	1	0			don't care
	Open-drain		1			don't care
Input	Analog	0	0	00		don't care
	Input floating		1			don't care
	Input pull-down	1	0			0
	Input pull-up					1

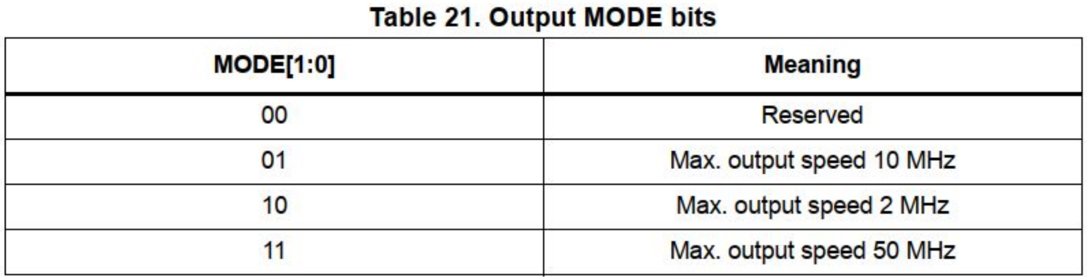


Table 21. Output MODE bits

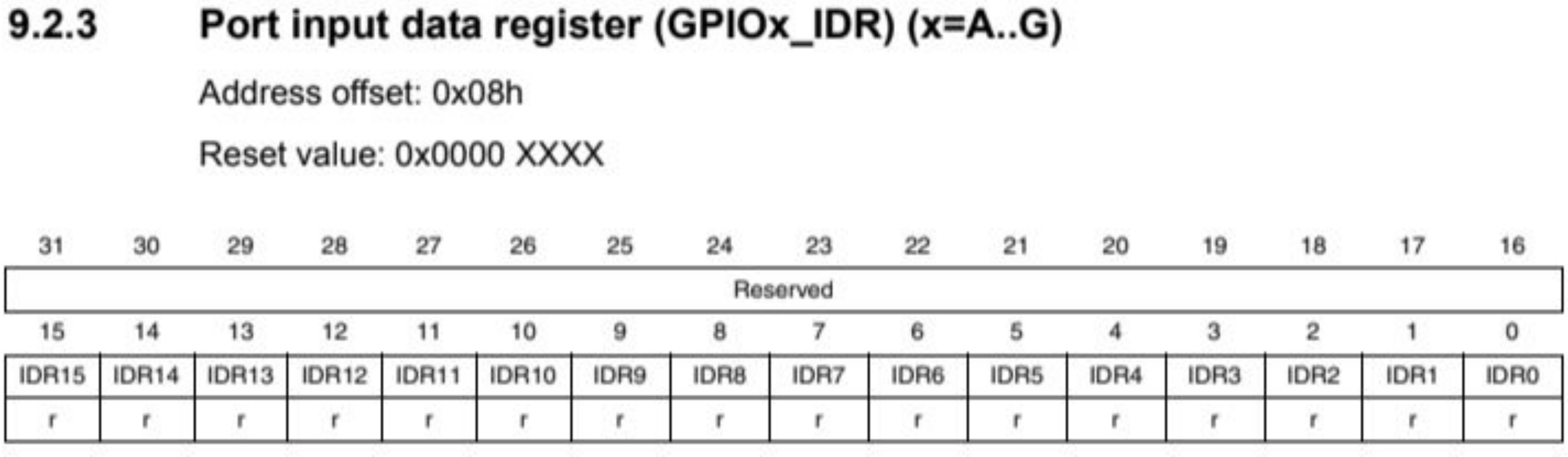
MODE[1:0]	Meaning
00	Reserved
01	Max. output speed 10 MHz
10	Max. output speed 2 MHz
11	Max. output speed 50 MHz

5.2. Cấu hình chế độ chân

```
    GPIOA->CRL |= (1 << 1); // CNF0 = 10: Input with pull-up/pull-down
    GPIOA->CRL &= ~(1 << 0);
    GPIOA->ODR |= (1 << 0); // Set ODR0, PA0 là input pull-up

    GPIOC->CRH |= (1 << 20) | (1 << 21); // MODE13 = 11: Output mode, max speed 50 MHz
    GPIOC->CRH &= ~((1 << 22) | (1 << 23)); // CNF13 = 00: General purpose output push-pull
```

5.3 Sử dụng ngoại vi



9.2.3 Port input data register (GPIOx_IDR) (x=A..G)

Address offset: 0x08h

Reset value: 0x0000 XXXX

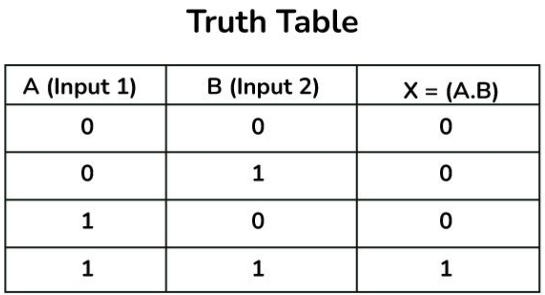
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

3.3. Sử dụng ngoại vi

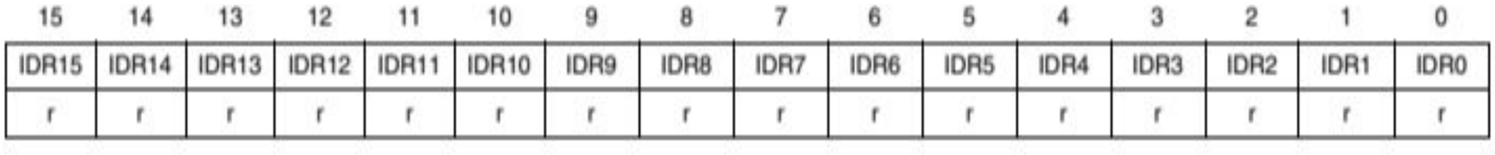
```
while(1){
    if((GPIOA->IDR & (1 << 0)) == 0) // Đọc trạng thái nút nhấn
    {
        GPIOC->ODR = 0 << 13;   // Nếu PA0 = 0 -> PC13 = 0
    }
    else
    {
        GPIOC->ODR = 1 << 13;   // Nếu PA0 = 1 -> PC13 = 1
    }
}
```

5.3 Sử dụng ngoại vi



Truth Table

A (Input 1)	B (Input 2)	X = (A.B)
0	0	0
0	1	0
1	0	0
1	1	1



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

a	1	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1
b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

a & b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1