

گزارشکار جلسه سوم ۶ آبان

آزمایشگاه سیستم های ریزپردازنده و مدارهای واسطه گروه ۳

آنوشا شریعتی ۹۹۲۳۰۴۱ مهشاد اکبری سریزدی ۹۹۲۳۰۹۳

سوال:

۱. سایز هر خط در حافظه چقدر است؟

سایز هر خط در حافظه ۸ بیت است. این مقدار با تقسیم کردن اندازه حافظه به تعداد خطوط حافظه به دست می آید.

۲. سایز حافظه از آدرس ۰X۲۰۰۰ ۰۰۰۰ تا ۰X۳۰۰۰ ۰۰۰۰ چقدر است؟

$$۳۰۰۰۰۰۰ - ۲۰۰۰۰۰۰ = ۱۰۰۰۰۰۰$$

با تبدیل این مقدار به دسیمال و ضرب کردن آن در سایز هر خط در حافظه جواب حدود ۲ مگابایت به دست می آید.

$$۲۶۸,۴۳۵,۴۵۶ * ۸ = ۲,۱۴۷,۴۸۳,۶۴۸$$

۳. با مطالعه map memory سایز فلش را تعیین کنید.

طبق شکل زیر سایز فلش معادل FFFFFF هگز است که با تبدیل آن به دسیمال عدد ۱,۰۴۸,۵۷۵ به دست می آید. با ضرب کردن آن در سایز هر خط حافظه جواب حدود ۸ کیلوبایت به دست می آید.

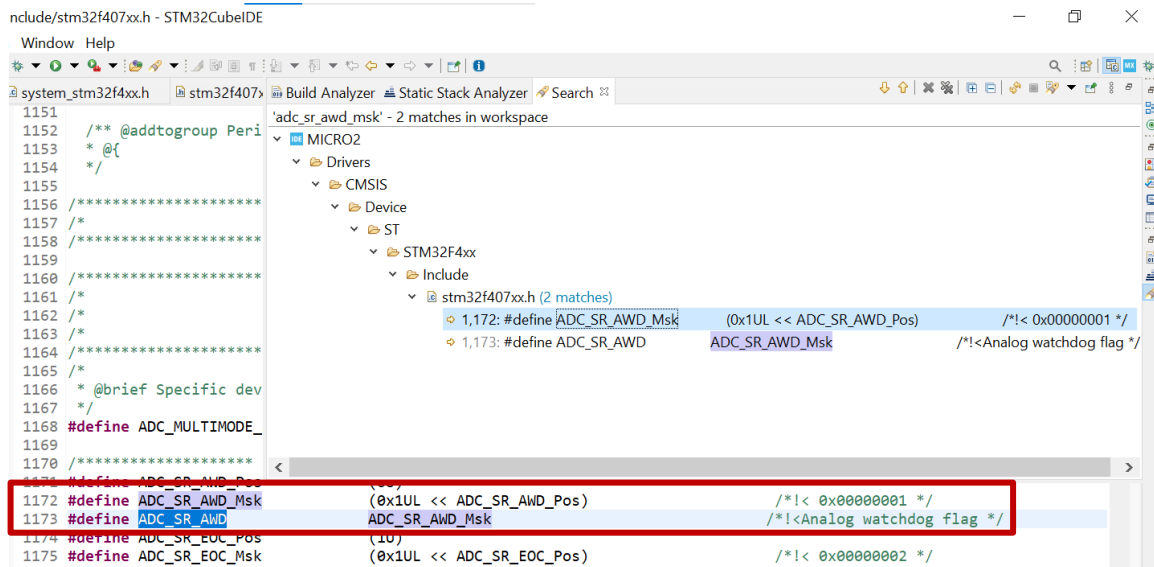
$$۱,۰۴۸,۵۷۵ * ۸ = ۸,۳۸۸,۶۰۰$$

Reserved	0x1FFF C008 - 0x1FFF FFFF
Option Bytes	0x1FFF C000 - 0x1FFF C007
Reserved	0x1FFF 7A10 - 0x1FFF 7FFF
System memory + OTP	0x1FFF 0000 - 0x1FFF 7A0F
Reserved	0x1001 0000 - 0x1FFE FFFF
CCM data RAM (64 KB data SRAM)	0x1000 0000 - 0x1000 FFFF
Reserved	0x0810 0000 - 0x0FFF FFFF
Flash	0x0800 0000 - 0x080F FFFF
Reserved	0x0010 0000 - 0x07FF FFFF
Aliased to Flash, system memory or SRAM depending on the BOOT pins	0x0000 0000 - 0x000F FF

تمرین :

۱. مقادیر `ADC_SR_AWD_MSK` و `ADC_SR_AWD` را بیابید.

با توجه به تصویر زیر مقادیر تعریف شده برای هردو یکسان و برابر با 0×00000001 می‌باشد.



۲. به کدام پریفرال تعلق دارند؟

به پریفرال ADC تعلق دارند.

۳. به کدام رجیستر تعلق دارند؟

به رجیستر SR تعلق دارند.

دستور کار: قسمت اول

۱. آدرس پایه پریفرال **GPIOD** را در فایل **User manual STM32F4۰۷** بیابید و مشخص کنید به کدام باس متصل است.

طبق تصویر زیر آدرس پایه این پریفرال ۰X۴۰۰۲۰C۰۰ است و به باس **AHB1** متصل است.

0x4002 3C00 - 0x4002 3FFF	register		Section 3.9: Flash interface registers
0x4002 3800 - 0x4002 3BFF	RCC	AHB1	Section 7.3.24: RCC register map on page 265
0x4002 3000 - 0x4002 33FF	CRC		Section 4.4.4: CRC register map on page 115
0x4002 2800 - 0x4002 2BFF	GPIOK		Section 8.4.11: GPIO register map on page 287
0x4002 2400 - 0x4002 27FF	GPIOJ		
0x4002 2000 - 0x4002 23FF	GPIOI		Section 8.4.11: GPIO register map on page 287
0x4002 1C00 - 0x4002 1FFF	GPIOH		
0x4002 1800 - 0x4002 1BFF	GPIOG		
0x4002 1400 - 0x4002 17FF	GPIOF		
0x4002 1000 - 0x4002 13FF	GPIOE		
0x4002 0C00 - 0x4002 0FFF	GPIOD		
0x4002 0800 - 0x4002 0BFF	GPIOC		
0x4002 0400 - 0x4002 07FF	GPIOB		
0x4002 0000 - 0x4002 03FF	GPIOA		
0x4001 6800 - 0x4001 6BFF	LCD-TFT	APB2	Section 16.7.26: LTDC register map on page 512
0x4001 5800 - 0x4001 5BFF	SAI1		Section 29.17.9: SAI register map on page 963
0x4001 5400 - 0x4001 57FF	SPI6	APB2	Section 28.5.10: SPI register map on page 925
0x4001 5000 - 0x4001 53FF	SPI5		

۲. مشخص کنید **LED** های **USER** به کدام پین های میکرو متصل اند.

طبق تصویر زیر **LED** ها به پایه های ۱۵-۱۲ **PD** متصل اند.

PD12	FSMC_A17/ TIM4_CH1/ USART3_RTS	59					GREEN
PD13	FSMC_A18/ TIM4_CH2	60					ORANGE
PD14	FSMC_D0/ TIM4_CH3	61					RED
PD15	FSMC_D1/ TIM4_CH4	62					BLUE

۳. رجیسترهای GPIO را بنویسید و کارکرد آنها را توضیح دهید و آدرس پایه آنها را بیابید. (در گزارش کار این جلسه فقط به MODER و BSRR و ODR پرداخته میشود)

- GPIO port mode register (GPIOx_MODER)
- GPIO port output type register (GPIOx_OTYPER)
- GPIO port output speed register (GPIOx_OSPEEDR)
- GPIO port pull-up/pull-down register (GPIOx_PUPDR)
- GPIO port input data register (GPIOx_IDR)
- GPIO port output data register (GPIOx_ODR)
- GPIO port bit set/reset register (GPIOx_BSRR)
- GPIO port configuration lock register (GPIOx_LCKR)
- GPIO alternate function low register (GPIOx_AFR1)
- GPIO alternate function high register (GPIOx_AFR2)

8.4.1 GPIO port mode register (GPIOx_MODER) (x = A..I/J/K)

Address offset: 0x00

Reset values:

- 0xA800 0000 for port A
- 0x0000 0280 for port B
- 0x0000 0000 for other ports

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
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

Bits 2y:2y+1 MODERy[1:0]: Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O direction mode.

00: Input (reset state)

01: General purpose output mode

10: Alternate function mode

11: Analog mode

طبق تصویر بالا آدرس پایه مربوط به این رجیستر ۰X۰۰ است و با استفاده از این رجیستر میتوان مود خروجی پایه ها را تعیین کرد.

8.4.6 **GPIO** port output data register (**GPIOx_ODR**) (x = A..I/J/K)

Address offset: 0x14

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
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

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **ODRy**: Port output data (y = 0..15)

These bits can be read and written by software.

*Note: For atomic bit set/reset, the ODR bits can be individually set and reset by writing to the **GPIOx_BSRR** register (x = A..I/J/K).*

طبق تصویر بالا آدرس پایه مربوط به این رجیستر ۰X۱۴ است. این رجیستر دیتا خروجی را ذخیره میکند.

8.4.7 **GPIO** port bit set/reset register (**GPIOx_BSRR**) (x = A..I/J/K)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:16 **BRy**: Port x reset bit y (y = 0..15)

These bits are write-only and can be accessed in word, half-word or byte mode. A read to these bits returns the value 0x0000.

0: No action on the corresponding ODRx bit

1: Resets the corresponding ODRx bit

Note: If both BSx and BRx are set, BSx has priority.

Bits 15:0 **BSy**: Port x set bit y (y = 0..15)

These bits are write-only and can be accessed in word, half-word or byte mode. A read to these bits returns the value 0x0000.

0: No action on the corresponding ODRx bit

1: Sets the corresponding ODRx bit

طبق تصویر بالا آدرس پایه مربوط به این رجیستر ۰X۱۸ است. از این رجیستر برای مقدار دهی به رجیستر ODR استفاده میشود.

۴. با باز کردن فایل `STM32F4xx.h`، `CLEAR_BIT` و `SET_BIT` را پیدا کرده و کارکرد آنها را توضیح دهید.

با استفاده از ابزار `SEARCH`، `CLEAR_BIT` و `SET_BIT` را پیدا کردیم. با استفاده از دستور `SET_BIT` میتوان یک بیت خاص از رجیستر را ۱ کرد. همچنین با دستور `CLEAR_BIT` میتوان یک بیت دلخواه از رجیستر را ۰ کرد.

```

192 {
193     DISABLE = 0U,
194     ENABLE = !DISABLE
195 } FunctionalState;
196 #define IS_FUNCTIONAL_ST
197
198 typedef enum
199 {
200     SUCCESS = 0U,
201     ERROR = !SUCCESS
202 } ErrorStatus;
203
204 /**
205  * @}
206  */
207
208
209 /** @addtogroup Exported
210  * @{
211  */
212
213 #define SET_BIT(REG, BIT) ((REG) |= (BIT))
214
215 #define CLEAR_BIT(REG, BIT) ((REG) &= ~(BIT))
216
217 #define READ_BIT(REG, BIT) ((REG) & (BIT))
218
219 #define CLEAR_REG(REG) ((REG) = (0x0))

```

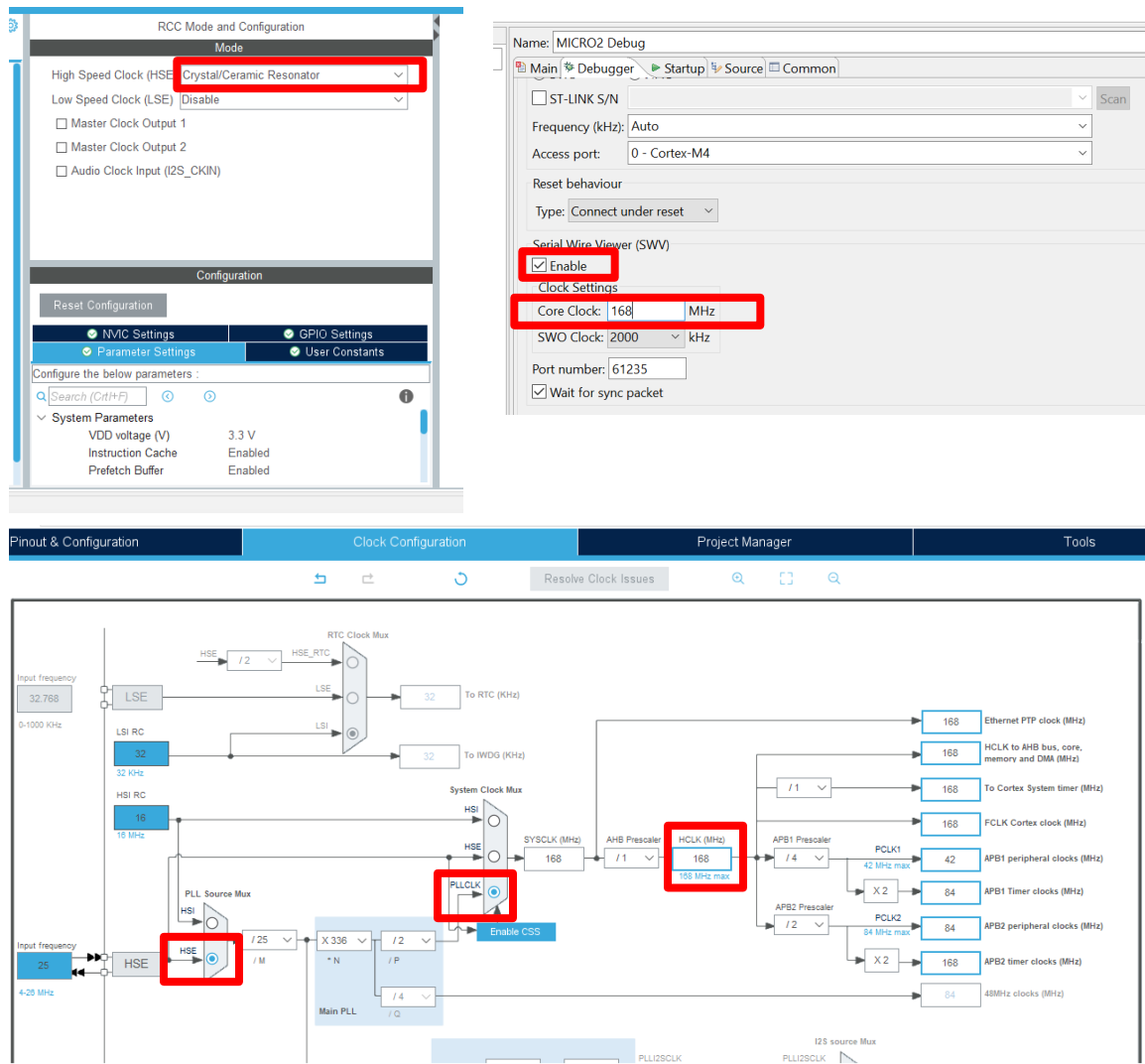
Occurrences of 'CLEAR_BIT' in workspace (29 matches)

- MICRO2
 - Drivers
 - CMSIS
 - Device
 - ST
 - STM32F4xx
 - Include
 - stm32f4xx.h
 - line 215: #define CLEAR_BIT(REG, BIT) ((REG) &= ~(BIT))

۵. تفاوت بین `atomic function` و `non atomic function` را توضیح دهید.

در توابع از نوع `atomic` عملکرد تابع به صورت یکباره و پیوسته اتفاق می‌افتد و نمی‌تواند در وسط متوقف شود. برای مثال با تغییر یک بیت در حین اجرای تابع آن را بررسی نمی‌کند. ولی در توابع از نوع `non atomic` عملکرد تابع میتواند به بخش‌های مختلف تقسیم شده و به صورت جدا از هم و موازی صورت بگیرد در این حالت با تغییر یک بیت تابع می‌تواند آن را بررسی کرده و تغییرات را اعمال کند.

۶. تنظیمات اولیه برای ساختن پروژه در CUBE IDE را انجام دهید.
طبق تصاویر زیر تنظیمات مربوط به کلاک را انجام میدهیم.



۷. تمام رجیسترهای GPIOD را برای LED های خروجی برد تعریف کنید و کلاک مربوط به آنها را فعال کنید.
برای تعریف کردن رجیستر ها آدرس پایه آنها را با آدرس پایه پرفیوال GPIOD جمع کردیم.
همچنین برای فعال کردن کلاک مربوط به پرفیوال GPIOD که به باس ۱ AHB وصل بود رجیستر RCC_AHB۱ را نیز به صورت زیر تعریف کردیم.

```

44
45 /* USER CODE BEGIN PV */
46 //MODER
47 #define GPIOD_MODER (*((volatile unsigned long*)(0x40020c00)))
48 //BSRR
49 #define GPIOD_BSRR (*((volatile unsigned long*)(0x40020c18)))
50 //ODR
51 #define GPIOD_ODR (*((volatile unsigned long*)(0x40020c14)))
52 //AHB1 CLK
53 #define RCC_AHB1ENR (*((volatile unsigned long*)(0x40023830)))
54
55 /* USER CODE END PV */

```

6.3.10 RCC AHB1 peripheral clock register (RCC_AHB1ENR)

Address offset: 0x30

Reset value: 0x0010 0000

Access: no wait state, word, half-word and byte access.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	OTGHS ULPIEN	OTGHS SEN	ETHMACPTP EN	ETHMACRXE N	ETHMACTXE N	ETHMACEN	Res.	DMA2DEN	DMA2EN	DMA1EN	CCMDAT ARAMEN	Res.	BKPSR AMEN	Reserved	
	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
Reserved			CRCE N	Res.	GPIOK EN	GPIOJ EN	GPIOIE N	GPIOH EN	GPIOG EN	GPIOFE N	GPIOEEN	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31 Reserved, must be kept at reset value.

Bit 30 **OTGHSULPIEN**: USB OTG HSULPI clock enable

This bit is set and cleared by software. It must be cleared when the OTG_HS is used in FS mode.

0: USB OTG HS ULPI clock disabled

1: USB OTG HS ULPI clock enabled

Bit 29 **OTGHSSEN**: USB OTG HS clock enable

This bit is set and cleared by software.

0: USB OTG HS clock disabled

1: USB OTG HS clock enabled

با توجه به تصویر بالا برای فعال کردن کلاک مربوط به پرفیفرال GPIOD باید بیت چهارم رجیستر RCC_AHB1ENR را که در قسمت قبل تعریف کردیم یک کنیم که با اضافه کردن دستور زیر در تابع MAIN انجام میشود.

```
RCC_AHB1ENR |= (1<<3);
```


۹. مقدار آن را در بخش SFR یافته و با expression live مقایسه کنید.

مقادیر به دست آمده در SFR بخش همان مقدار نشان داده شده توسط live expression بود. همچنین با ۰ و ۱ کردن مقدار بیت های ۱۲ تا ۱۵ رجیستر ODR میتوان LED های مورد نظر را خاموش و روشن کرد.

type filter text		
Register	Address	Value
▼ ODR	0x40020...	0xf000
ODR15	[15:1]	0x1
ODR14	[14:1]	0x1
ODR13	[13:1]	0x1
ODR12	[12:1]	0x1
ODR11	[11:1]	0x0
ODR10	[10:1]	0x0
ODR9	[9:1]	0x0
ODR8	[8:1]	0x0
ODR7	[7:1]	0x0
ODR6	[6:1]	0x0
ODR5	[5:1]	0x0
ODR4	[4:1]	0x0
MSB 00000000 00000000 11110000 00000000 LSB		
Register:	ODR	
Address:	0x40020c14	
Value:	0xf000	
Size:	32	
Reset value:	0x0	

