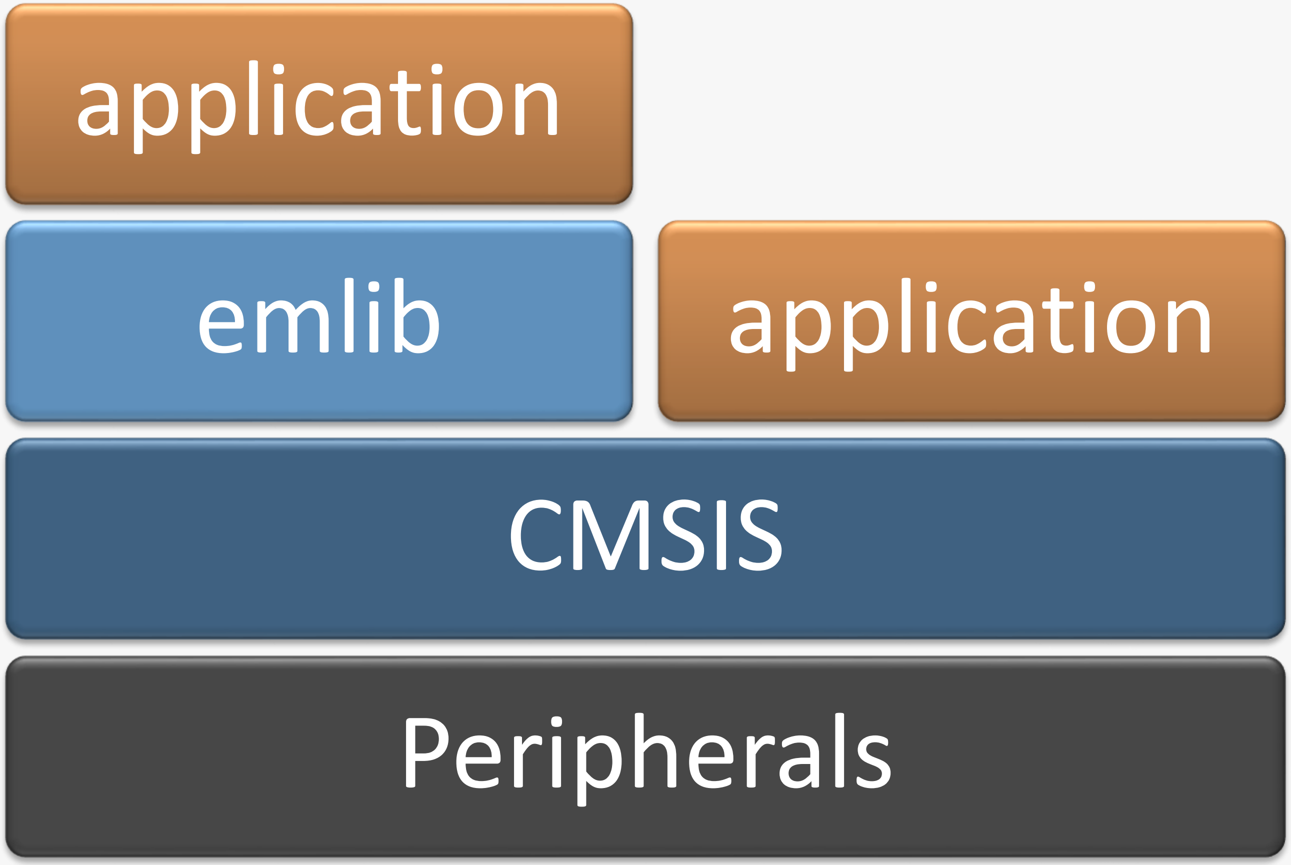
EFM32 Series 0: Introduction to EFM32 API

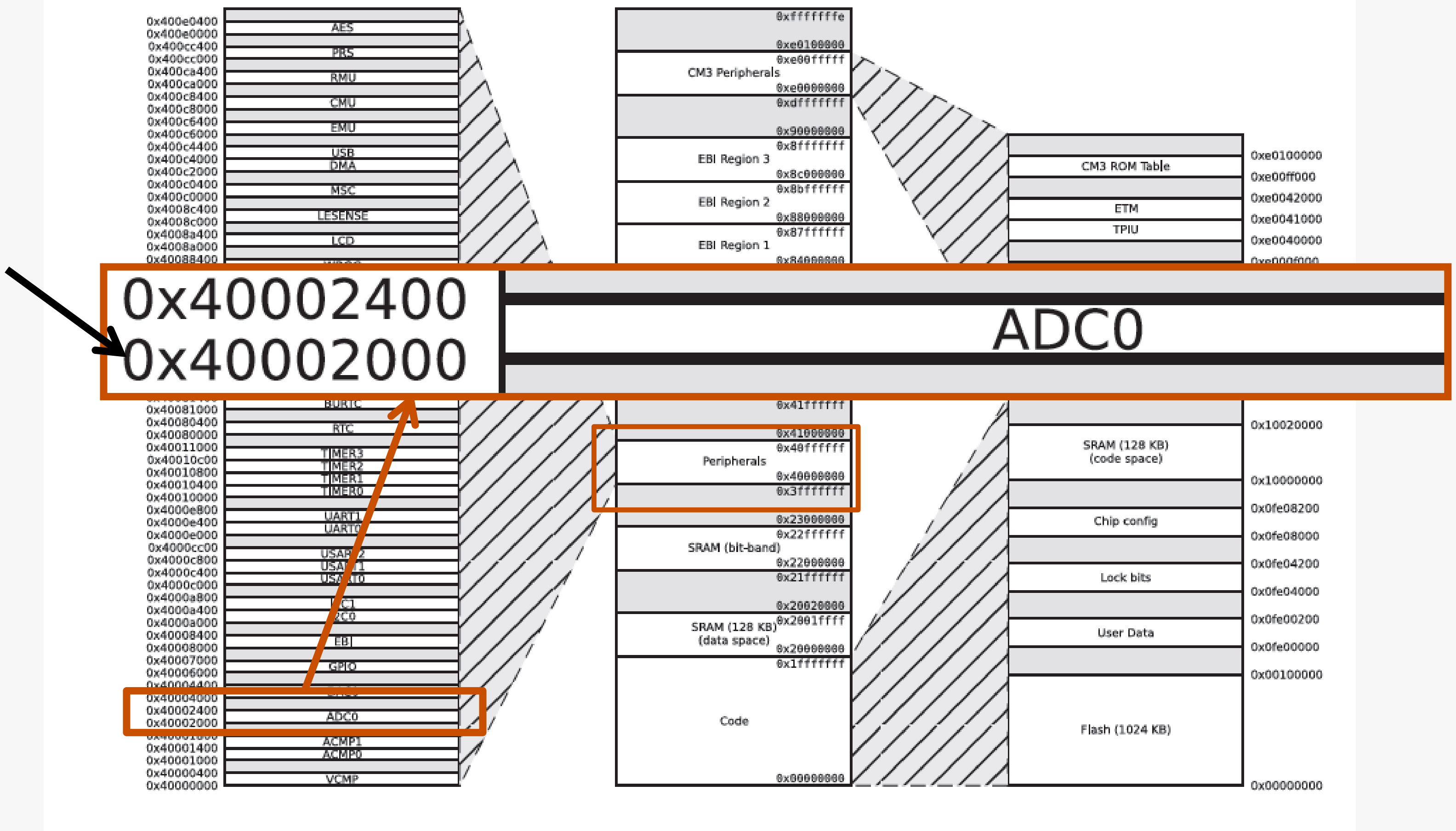
Introduction to EFM32 API



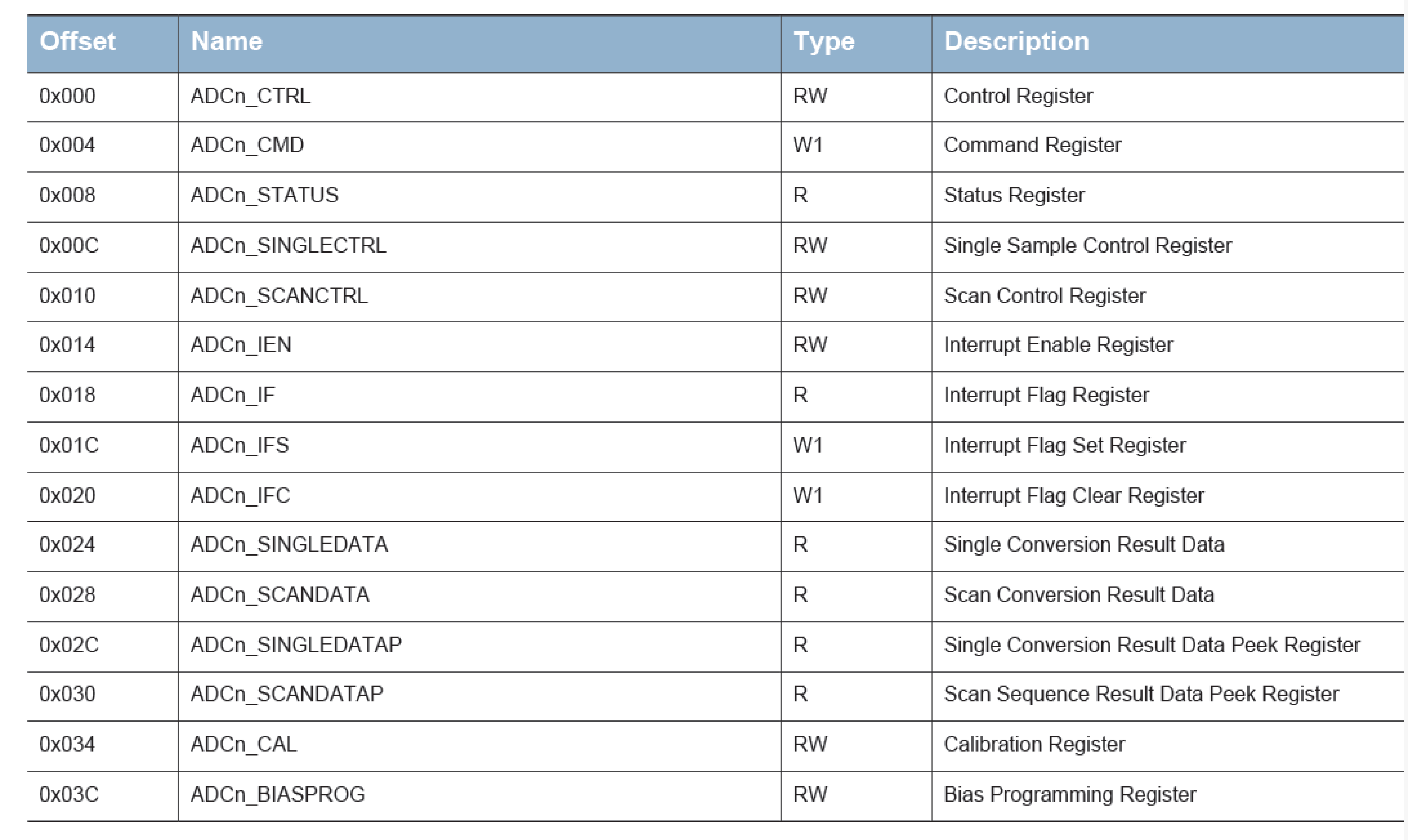
Memory Mapped Peripherals

Figure 5.2. System Address Space

Base Address

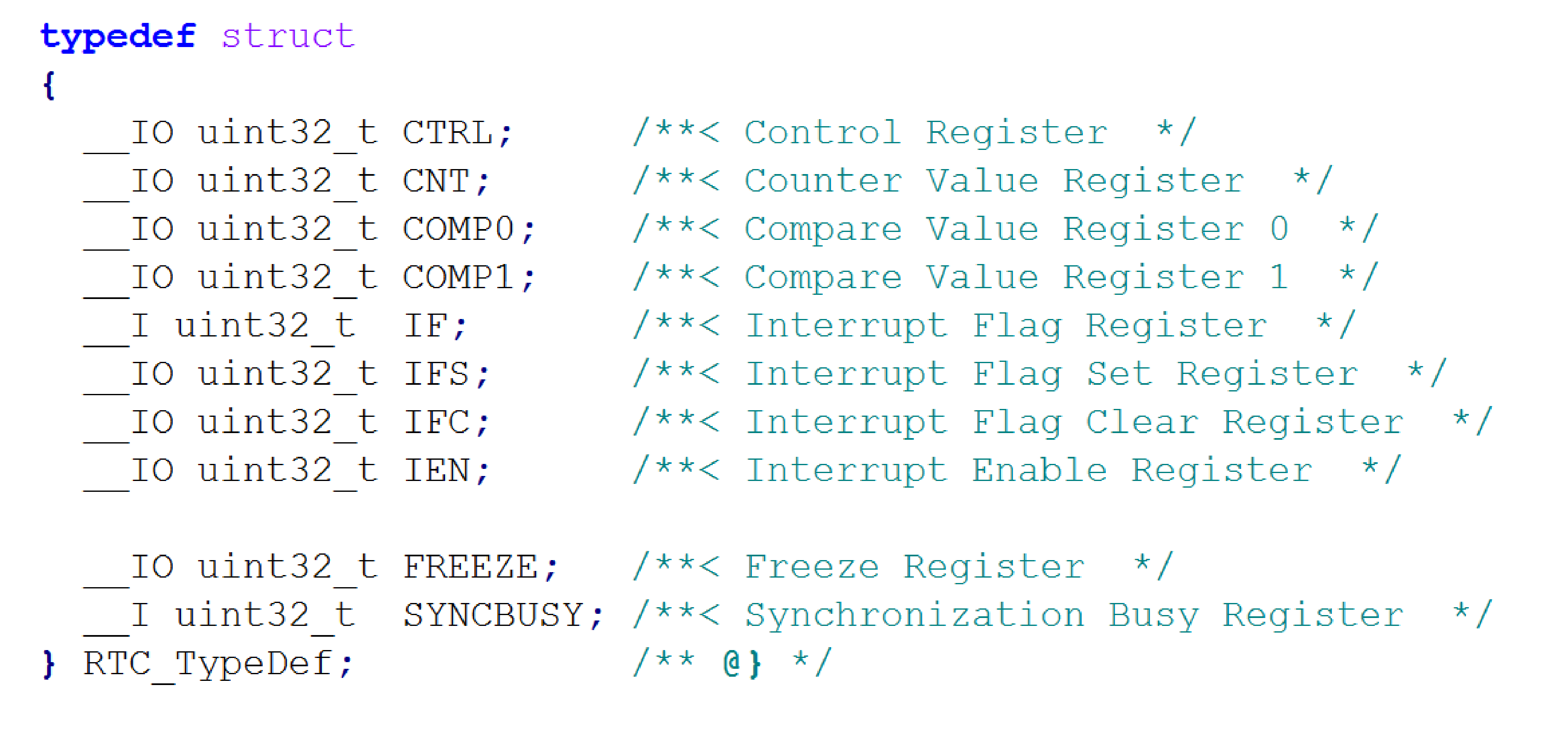


Peripheral Registers



Offset	Name	Type	Description
0x000	ADCn_CTRL	RW	Control Register
0x004	ADCn_CMD	W1	Command Register
0x008	ADCn_STATUS	R	Status Register
0x00C	ADCn_SINGLECTRL	RW	Single Sample Control Register
0x010	ADCn_SCANCTRL	RW	Scan Control Register
0x014	ADCn_IEN	RW	Interrupt Enable Register
0x018	ADCn_IF	R	Interrupt Flag Register
0x01C	ADCn_IFS	W1	Interrupt Flag Set Register
0x020	ADCn_IFC	W1	Interrupt Flag Clear Register
0x024	ADCn_SINGLEDATA	R	Single Conversion Result Data
0x028	ADCn_SCANDATA	R	Scan Conversion Result Data
0x02C	ADCn_SINGLEDATAP	R	Single Conversion Result Data Peek Register
0x030	ADCn_SCANDATAP	R	Scan Sequence Result Data Peek Register
0x034	ADCn_CAL	RW	Calibration Register
0x03C	ADCn_BIASPROG	RW	Bias Programming Register

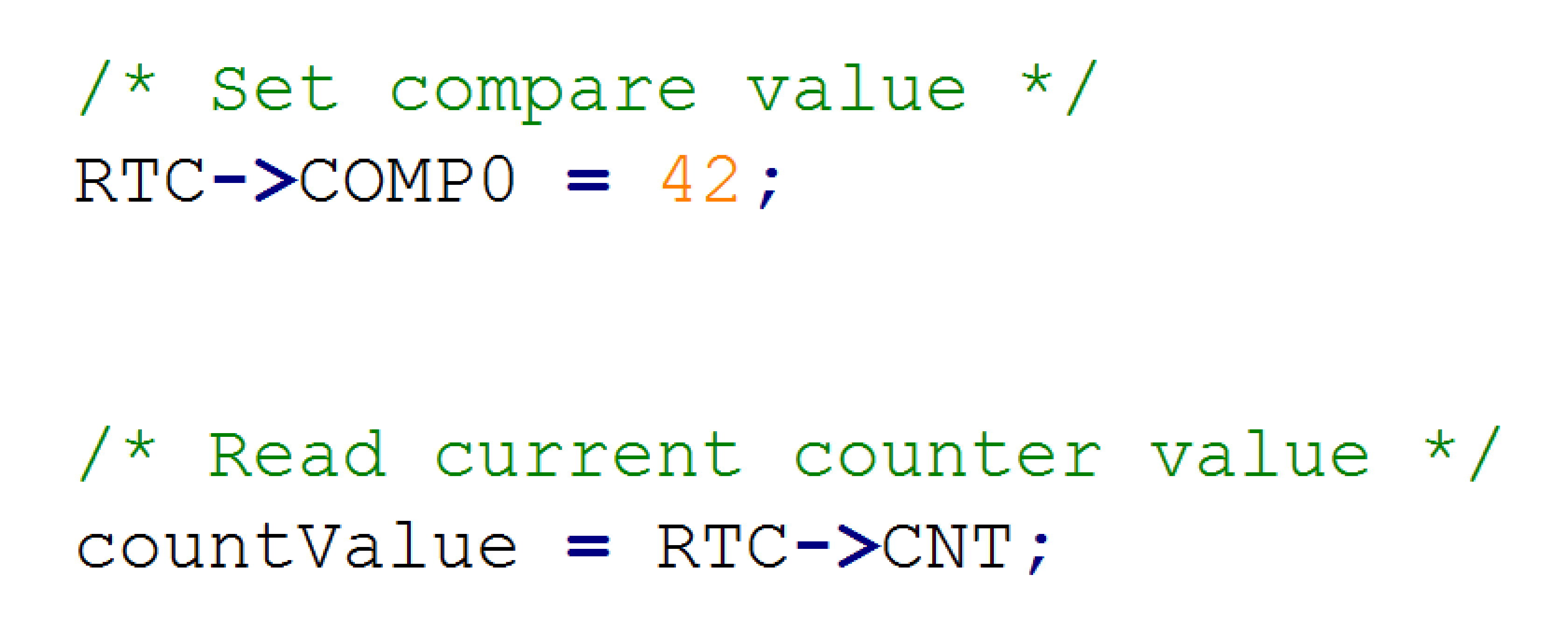
CMSIS Struct



```
typedef struct
{
  __IO uint32_t CTRL;       /**< Control Register  */
  __IO uint32_t CNT;        /**< Counter Value Register  */
  __IO uint32_t COMP0;      /**< Compare Value Register 0  */
  __IO uint32_t COMP1;      /**< Compare Value Register 1  */
  __I uint32_t  IF;         /**< Interrupt Flag Register  */
  __IO uint32_t IFS;        /**< Interrupt Flag Set Register  */
  __IO uint32_t IFC;        /**< Interrupt Flag Clear Register  */
  __IO uint32_t IEN;        /**< Interrupt Enable Register  */

  __IO uint32_t FREEZE;     /**< Freeze Register  */
  __I uint32_t  SYNCBUSY;   /**< Synchronization Busy Register  */
} RTC_TypeDef;              /** @} */
```

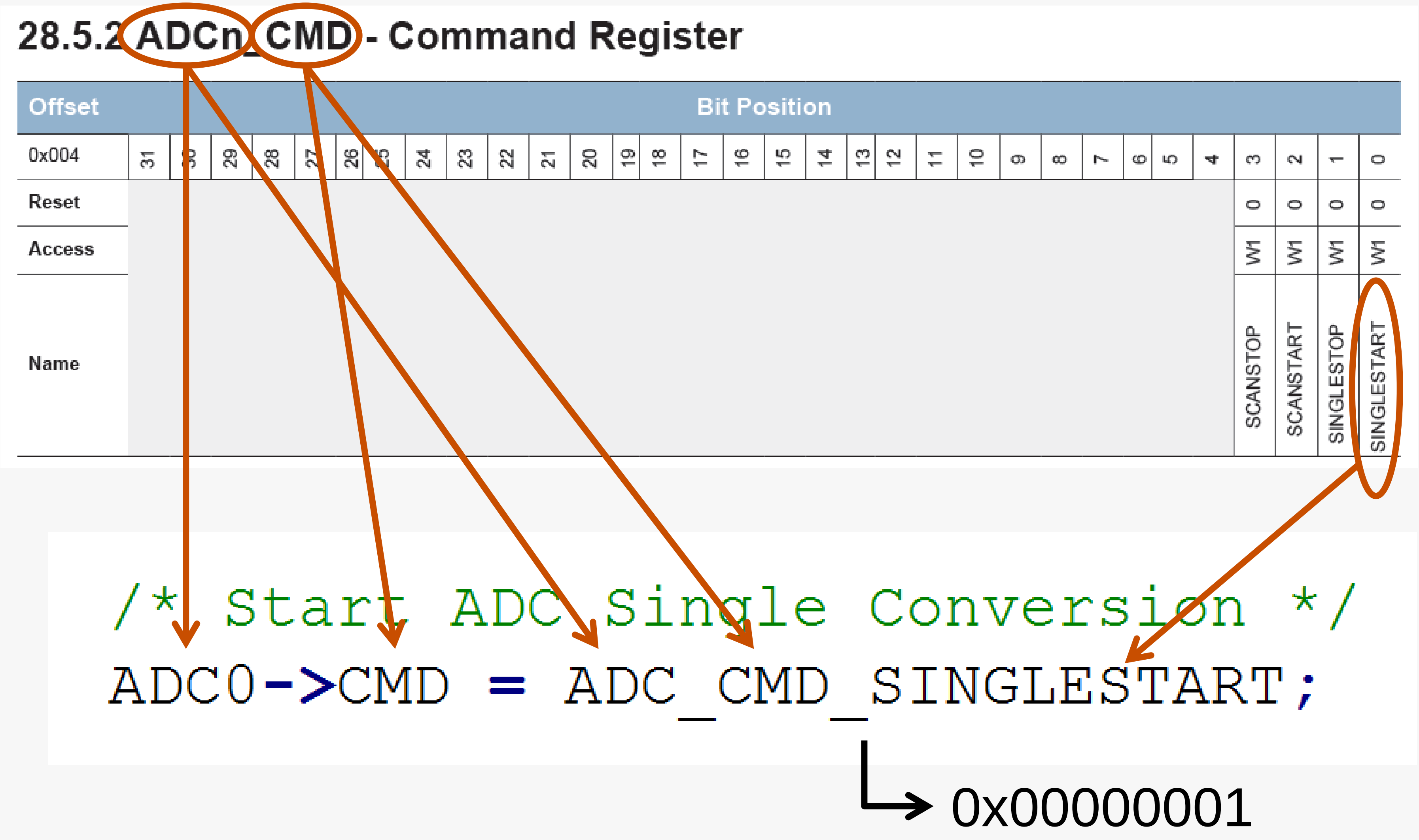
Read and Write Registers



```
/* Set compare value */
RTC->COMP0 = 42;


/* Read current counter value */
countValue = RTC->CNT;
```

CMSIS Naming Convention



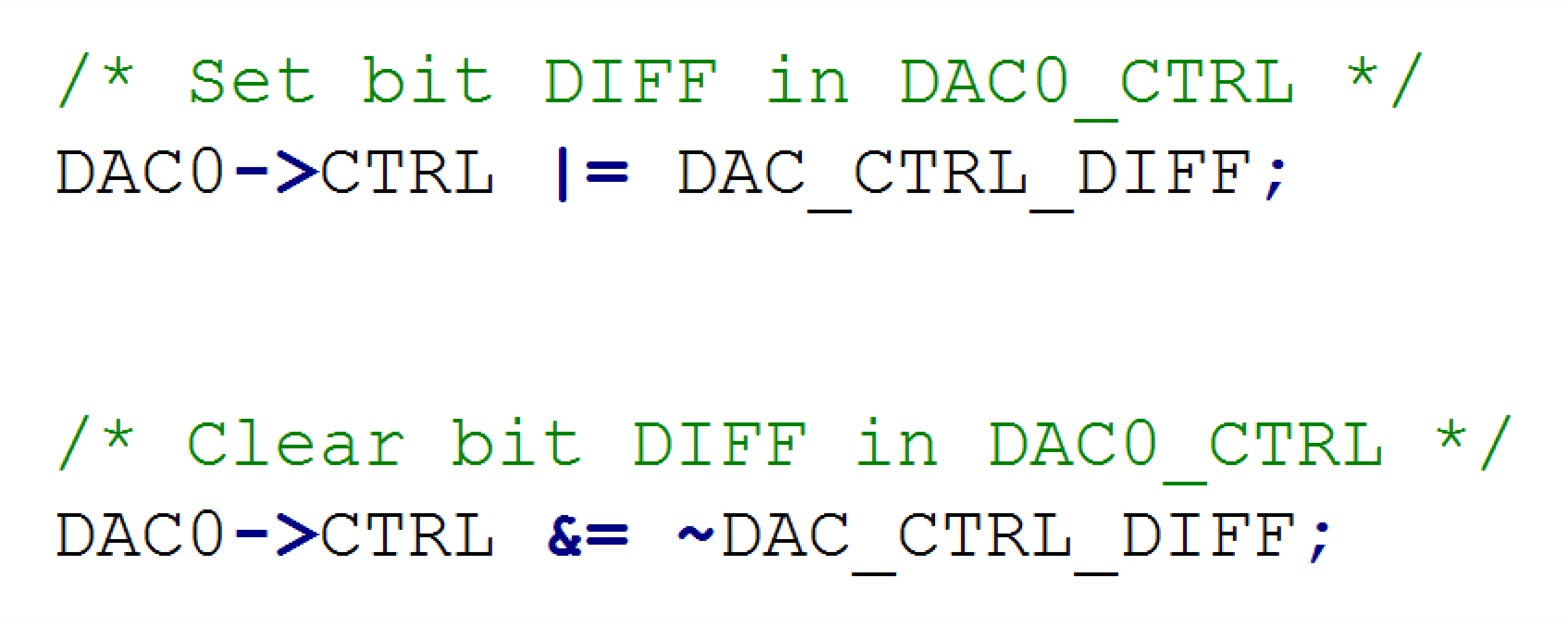
28.5.2 ADCn_CMD - Command Register

Offset	Bit Position																															
0x004	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
Reset																													0	0	0	0
Access																													W1	W1	W1	W1
Name																													SCANSTOP	SCANSTART	SINGLESTOP	SINGLESTART

```
/* Start ADC Single Conversion */
ADC0->CMD = ADC_CMD_SINGLESTART;
```

0x00000001

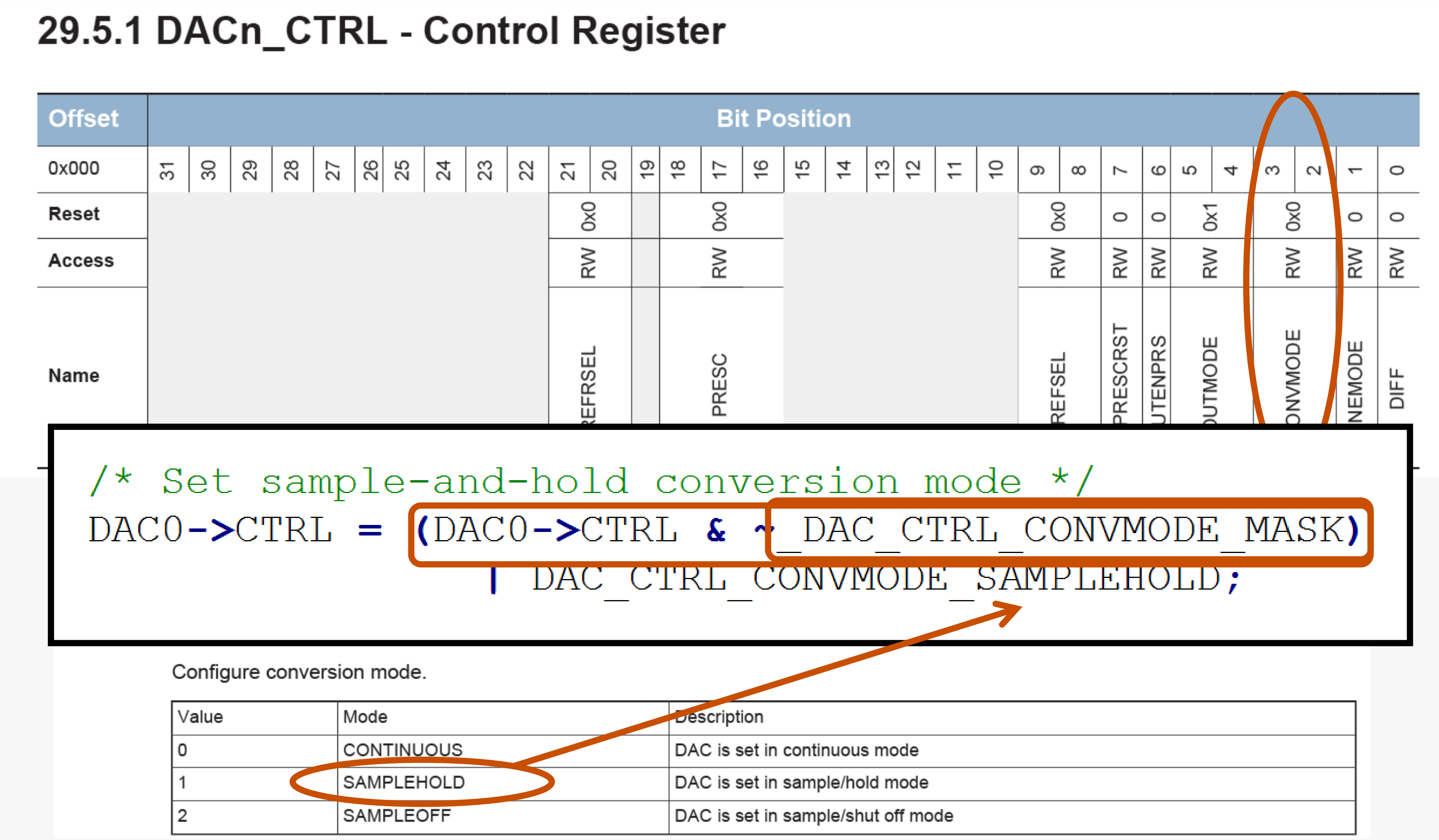
Set and Clear Bits



```
/* Set bit DIFF in DAC0_CTRL */
DAC0->CTRL |= DAC_CTRL_DIFF;


/* Clear bit DIFF in DAC0_CTRL */
DAC0->CTRL &= ~DAC_CTRL_DIFF;
```

Multiple-Bit Fields



29.5.1 DACn_CTRL - Control Register

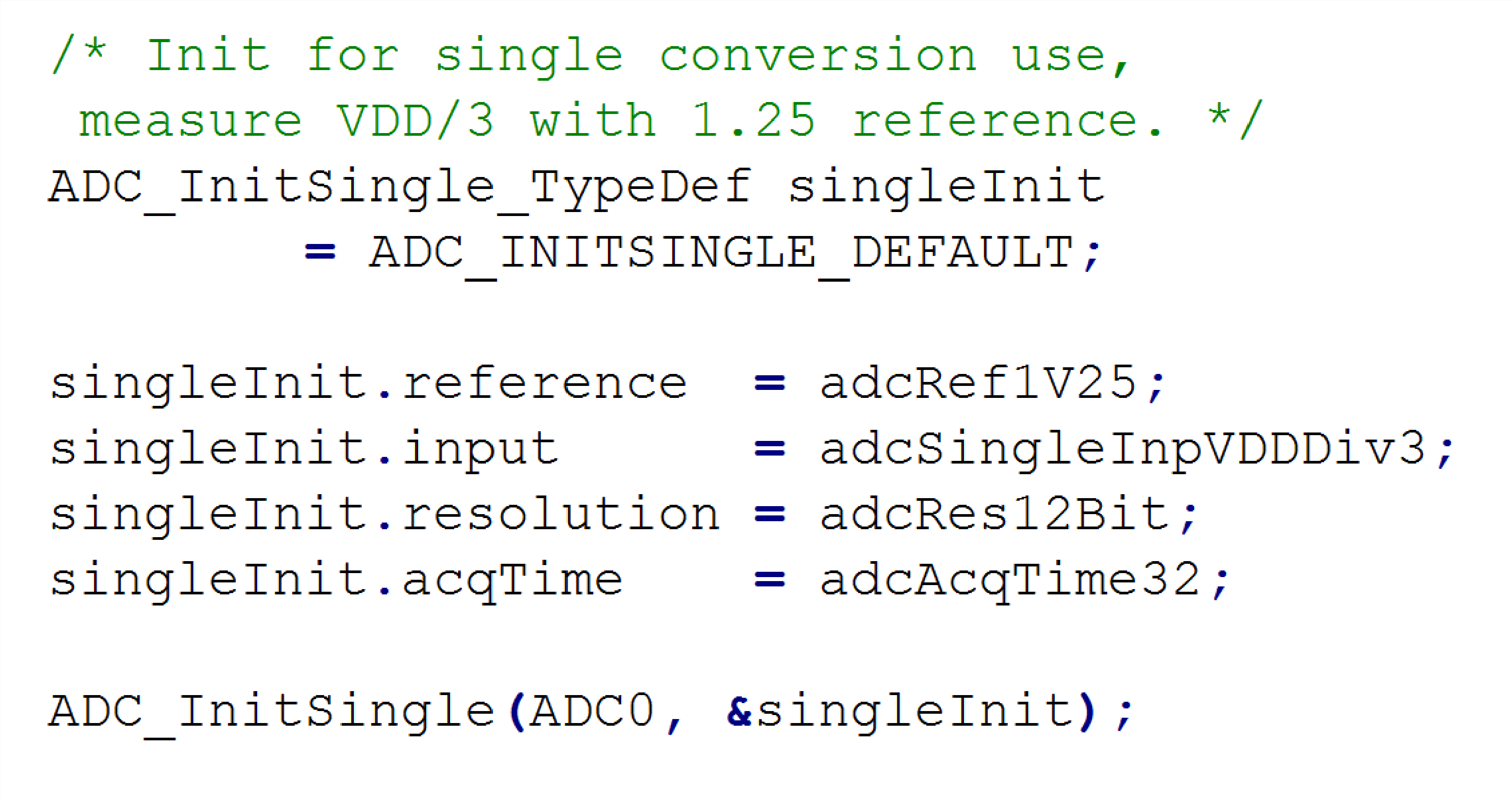
Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0			0x0									0x0		0	0	0x1		0x0		0	0
Access											RW			RW									RW		RW	RW	RW		RW		RW	RW
Name											EFRSEL			PRESC									REFSEL		PRESCRST	UTENPRS	UTMODE		ONVMODE		NEMODE	DIFF

```
/* Set sample-and-hold conversion mode */
DAC0->CTRL = (DAC0->CTRL & ~_DAC_CTRL_CONVMODE_MASK)
                | DAC_CTRL_CONVMODE_SAMPLEHOLD;
```

Configure conversion mode.

Value	Mode	Description
0	CONTINUOUS	DAC is set in continuous mode
1	SAMPLEHOLD	DAC is set in sample/hold mode
2	SAMPLEOFF	DAC is set in sample/shut off mode

emlib



```
/* Init for single conversion use,
 measure VDD/3 with 1.25 reference. */
ADC_InitSingle_TypeDef singleInit
        = ADC_INITSINGLE_DEFAULT;

singleInit.reference  = adcRef1V25;
singleInit.input      = adcSingleInpVDDDiv3;
singleInit.resolution = adcRes12Bit;
singleInit.acqTime    = adcAcqTime32;

ADC_InitSingle(ADC0, &singleInit);
```

SILICON LABS

www.silabs.com/mcu

Hands-On Lesson:

TM0007 in Simplicity Studio