

Microprocessor System Design

< BLOCK 1: Memory map and Assembly language >

CPU $\xrightarrow{\text{memory}}$ MPU $\xrightarrow{\text{peripherals}}$ MCU.

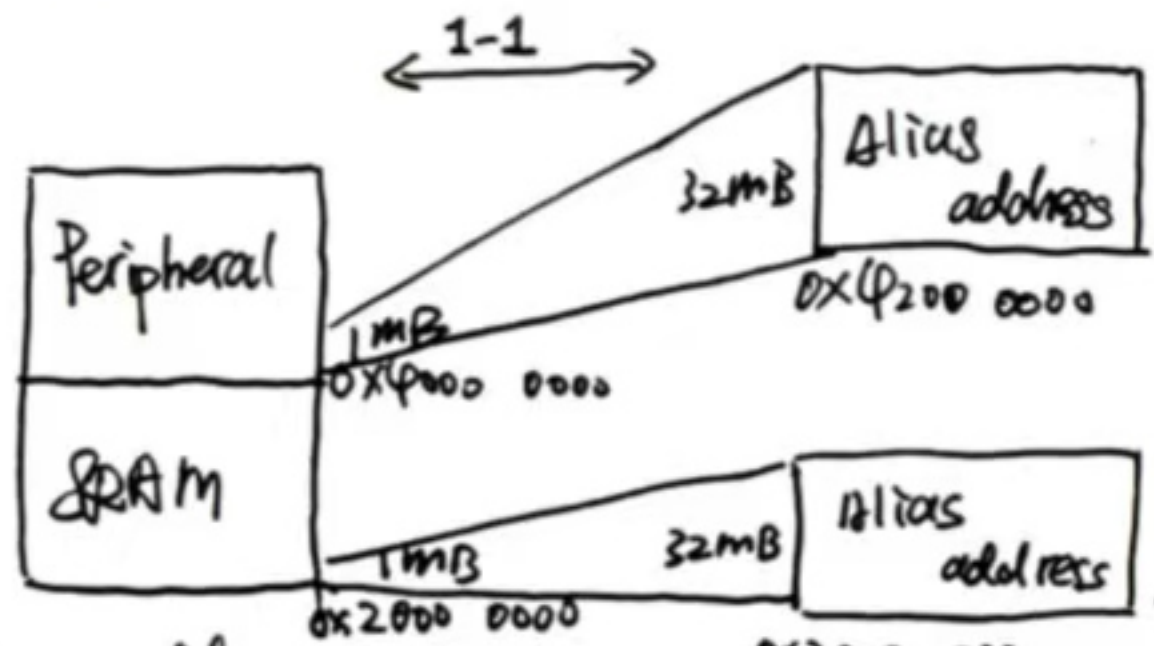
1. Processor Register (R)



2. Memory map & alias address.

Assumption 共 4GB 内存空间, 使用 32 位 bit 表示, 在文档中习惯 8 位 16 进制表示。

Model



Faster, Atomic.

通过 alias address 可直接对 peripheral/SRAM 中的单个 bit 进行操作。

别名地址 = 别名基地址 (0x2200 0000 或 0x4200 0000)
+ 字节偏移 $\times 32$ (绝对地址对于 0x2000 0000/0x4000 0000 的偏移)
+ 位偏移 $\times 4$ (位编号)。

3. Assembly Language. (对照 slides 学习) 每 register 有 32 bit.

① 内存操作 (读 MCU 的存储方式是 MBR 在基地址)。

LDR $\left\{ \begin{smallmatrix} R \\ B \end{smallmatrix} \right\}, \text{STR} \left\{ \begin{smallmatrix} R \\ B \end{smallmatrix} \right\}, \text{POP}, \text{PUSH}.$

应用 读寄存器, $\{ \text{LDR}, \#4; [\text{R0}, \#4]; [\text{R0}, \#4]! \}$, 直接 = mydata,

POP $\{ \text{R1} \}$
PUSH $\{ \text{R2} \}$
POP $\{ \text{R1} \}$
PUSH $\{ \text{R2} \}$

② 算术 (注意这里所有的算术只针对 32 进制) (补码)。

MOV, $\{ \text{ADD}, \text{ADC} \}, \{ \text{SUB}, \text{SBC}, \text{ASB} \}, \{ \text{LSL}, \text{LSR} \}, \text{ASR}, \text{ROR}, [\frac{S}{C}] \text{XTB}, [\frac{S}{C}] \text{XTH}.$

左移位, 在寄存器新插入值 (NZCV) 时使用 (中间移位)。

应用 取模值, 基本运算, 多 bit 取模运算, Bit 移位。

③ Logic

AND, ORR, EOR, 按位操作

④ 条件分支。

B, 左移位: $\{ \text{EQ}, \text{NE} \}, \{ \text{GT}, \text{LT} \}, \{ \text{GE}, \text{LE} \},$ 比较: CMP.

4. C & Assembly (配合 slides 学习)。

① C 中数据类型存储在 register 中。

② 调用函数时

r0-r3: 传参, 不用保存; 并且做返回值。

r4-r8, r10-r11: 变量, 若再调用函数 function 相关保存。

lr: 嵌套时需保存, 否则找不到。

注: 计科和法系主要看例子学习, 这里的标记简单提供回忆。

< BLOCK 2: Peripherals > 核心在复习的例题。

1. General-Purpose Input/Output (GPIO)

主要考虑使用汇编设置寄存器, 控制 Input/Output, 每个 GPIO 控制 16 个 I/Os.

△ ① GPIOx_MODER (2 bit)

$\{00\}$ input $\{01\}$ output.

② GPIO_OTYPER (1 bit)

$\{0\}$ push-pull $\{1\}$ open-drain

③ GPIO_PUPDR (2 bit)

$\{00\}$ No pull-up, pull-down.

△ ④ GPIO_IDR/GPIO_ODR (1 bit)

用来读/写。

output Bit	push-pull	open-drain
1	High	$\bar{Z}$
0	Low	Low

Prioritizing access
immediate offset.
更多看附的例子。

这里需要根据寄存器表格进行汇编的编写, 与寻址:

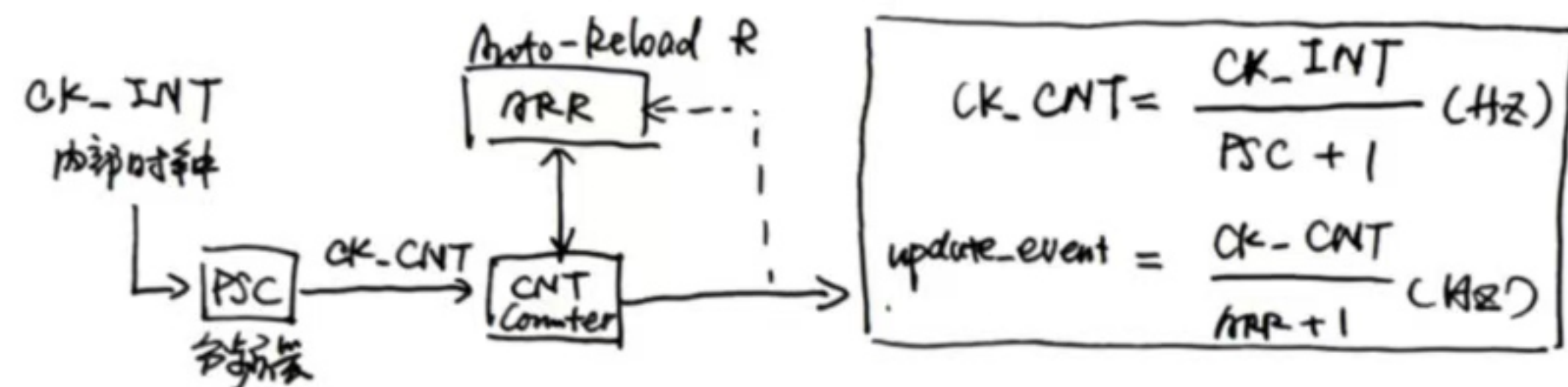
取地址 $\rightarrow$ 右数据 $\rightarrow$ 处理数据 $\rightarrow$ STR.
LDR, 地址, LDR, R1, [R0] ORR, R1, AND, STR.

2. Timer

主要 Base-time unit 和 PWM mode.

① Base-time unit

model



有3种 Counter mode (up, down, Center-aligned)

对于 up/down: 触发事件周期 = $(ARR+1) \cdot \frac{1}{CK_CNT}$
 对于 Center-aligned: 触发事件周期 = $2 \cdot ARR \cdot \frac{1}{CK_CNT}$

注: 触发事件可以是 PWM 信号, 即通过 ARR, PSC 设置 PWM 信号周期.

up/down: PWM 周期 = $(ARR+1) \cdot \frac{1}{CK_CNT}$

Center-aligned: PWM 周期 = $2 \cdot ARR \cdot \frac{1}{CK_CNT}$

② PWM mode

通过 Capture/Compare Register (CCR) 的值与 Counter 数值比较, 设置占空比, 所以这里的关键就是 CCR Value 的确定.

对于 Counter 的不同设置由 ARR 和 CCR 设置:

up/down: PWM pulse width = $CCR \times \frac{1}{CK_CNT}$
 Center-Aligned: PWM pulse width = $2 \times CCR \times \frac{1}{CK_CNT}$

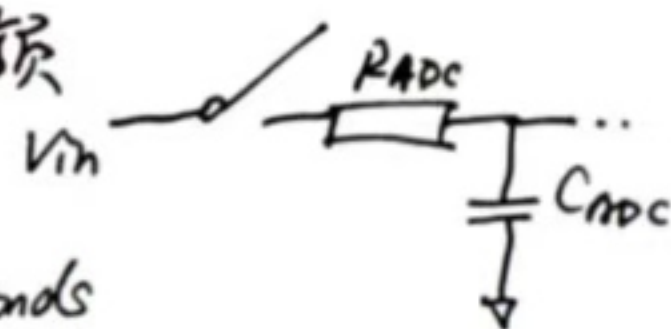
其余部分请参考 PPT.

3. Analogue to Digital Conversion (ADC)

EX

PCLK2 = 84 MHz, ADCCLK 为 PCLK2 的 4 分频

$R_{ADC} = 6 \text{ k}\Omega$, $C_{ADC} = 4 \text{ pF}$, 求:



- Minimum sampling time in both microseconds and clock cycles
- What is the total conversion time for 12-bit resolution based on minimum sampling
- Discuss impact of ADC resolution between 12-bit and 6-bit in terms of Quantisation size and Quantisation error.

Solution

$$(1) \text{ ADCCLK} = \frac{\text{PCLK2}}{4} = 21 \text{ MHz}$$

$$T_{ADC} = \frac{1}{21 \text{ MHz}} \approx 47.62 \text{ ns}$$

minimum sampling time:

$$t_{\text{sampling}} = 5 \cdot R_{ADC} \cdot C_{ADC} = 0.12 \mu\text{s}$$

$$\text{时钟周期数} = \frac{t_{\text{sampling}}}{T_{ADC}} \approx 2.52, \text{取整, 3 个周期.}$$

则 3 周期对应最小采样时间 $3T_{ADC} \approx 142 \mu\text{s}$

- 对于 12 bit resolution, $t_{\text{conversion}} = 12 \text{ Cycles}$. (同样, 6 bit 为 6 Cycles)

$$t_{\text{total}} = t_{\text{sampling}} + t_{\text{conversion}} = 15 \text{ Cycles} = 15 T_{ADC}$$

- Quantisation size: 最小分辨率. 5 用 10 位分辨率芯片

$$12 \text{ bit} = \left\lceil \frac{V_{ref}}{2^{12}} \right\rceil = \frac{3.3}{4096} \approx 0.805 \text{ mV}$$

$$6 \text{ bit} = \frac{V_{ref}}{2^6} = \frac{3.3}{64} \approx 51.56 \text{ mV}$$

Quantisation error:

最大量化误差为半个量化步长.

$$12 \text{ bit}: \text{error} \in [-0.805/2, 0.805/2] \text{ (mV)}, \text{最大 } 0.805/2 \text{ mV}$$

$$6 \text{ bit}: \text{error} \in [-51.56/2, 51.56/2] \text{ (mV)}, \text{最大 } 51.56/2 \text{ mV.}$$

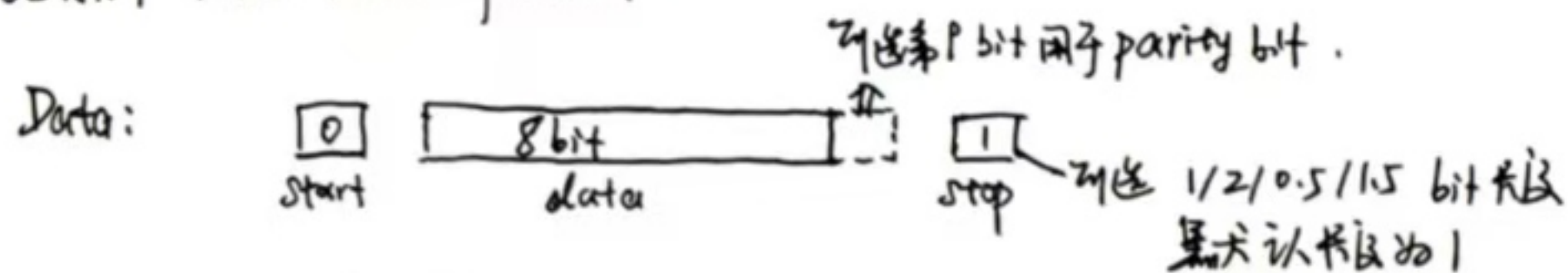
再简单讨论即可.

4. USART

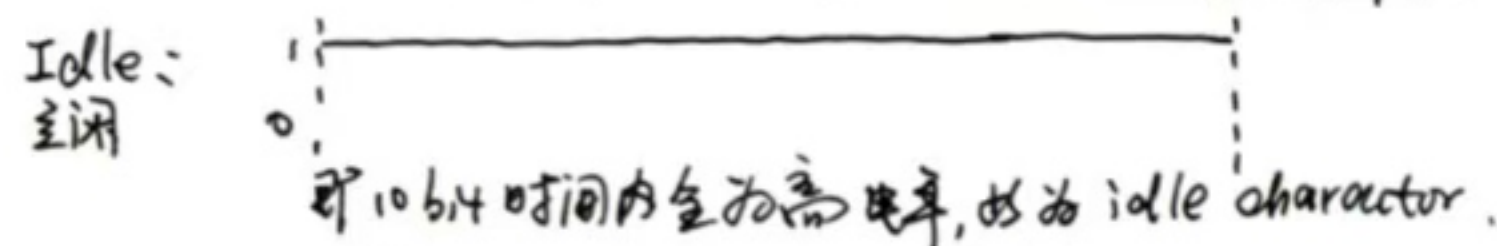
Universal Synchronous/Asynchronous Receiver Transmitter

一种硬件通信协议, 支持 full duplex.

① USART Data Packet format.

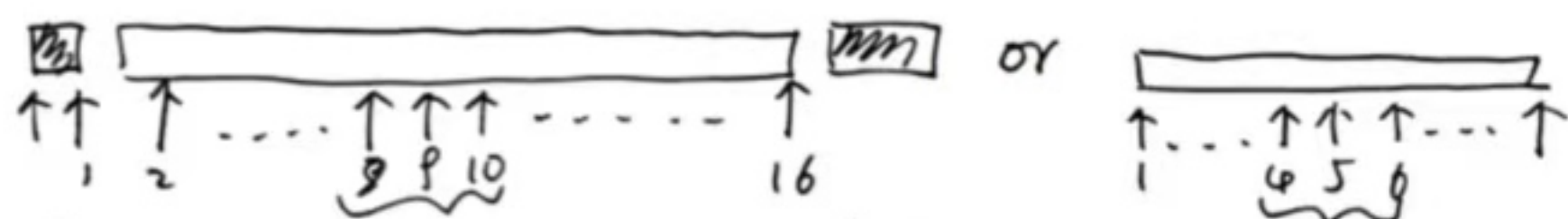


一般情况下, 每 10 bit 数据内含有 8 bit 有效数据.



② Oversampling of bit.

每 bit 选择 $16 \times / 8 \times$ oversampling, to improve bit detection accuracy and to mitigate noise.



取 8, 9, 10 (16x) 或 4, 5, 6 (8x) 3 个采样点, 根据 2:1 的压测决策实际信号.

Oversampling 还可用于 Start bit detection (111 0X 0X 0X 0000).

③ Band Rate

$$\text{Band Rate} = \text{bits/s} \Rightarrow \text{bps}$$

考虑 1 start bit, 1 stop bit, 8 有效数据的 USART 数据包.

$$\text{Band rate} = 115200 \text{ bps.}$$

$$\text{Transmission rate of actual data} = \frac{115200}{1+8+1} = 11520 \text{ Byte/s.}$$

每 10 bit 数据包中有 1 Byte (8 bit) 有效数据.

关于 Band Rate 的设置:

MCU 使用 USART-BRR (Band Rate Register) 来设置 Band Rate.

具体的:

$$\text{Band Rate} = \frac{\text{PCLK}}{\text{采样数} \times \text{USARTDIV}}, \text{ 采样数} = 8 \times (2 - \text{OVER8})$$

↓
16/8

L: 0, 16x
1: 8x

USARTDIV: USART Divisor, 与频率因子, 与寄存器 USART BRR 中

EX

$$\text{PCLK} = 62 \text{ MHz, Desired Band Rate} = 115200 \text{ bps.}$$

Oversampling 16, i.e. OVER8 = 0, 5.

1) USARTDIV (整数的)

2) USART-BRR 数值.

3) 真实的 USARTDIV, 真实设计出的 Band Rate.

4) error.

Solution

$$\text{Band Rate} = \frac{\text{PCLK}}{16 \times \text{USARTDIV}}$$

$$\text{USARTDIV} = \frac{\text{PCLK}}{16 \times \text{Band Rate}} = 22.786. \text{ (整数的)}$$

(2) USART-BRR 只给小数为 4 bit (16x) 或 3 bit (8x), 正好对应一个 16 进制或 8 进制, 所以 USARTDIV 需先转换为 16 进制再处理小数部分.

$$22.786 = 16 + 6 + 0.786 = 0x10.4 \quad 0x06.0 + 0x00.D$$

$$= 0x16D.$$

$$0.786 \times 16 = 12.576 \rightarrow 13 \rightarrow D$$

则用 0x16 表示整数位, 0xD 表示小数位.

USART-BRR 共 16 bit, 填入 0x16D, 其中 4 位是小数.

(3) 从 USART-BRR 算出的 USARTDIV 值最后使用的数值.

$$\text{USARTDIV} = 0x16D + \frac{0xD}{16} = 22 + \frac{13}{16} = 22.8125$$

所以发现, 与主频不同, 则设计出的 Band Rate:

$$\text{Band Rate} = \frac{\text{PCLK}}{16 \times \text{USARTDIV}} = 115068 \text{ bps.}$$

(4)

$$\text{error} = |115200 - 115068| \text{ bps. 一般相对误差} = \frac{|\text{Calculated} - \text{Desired}|}{|\text{Desired}|} \approx 0.11\%$$

注: 关键在于分清理想值, 与 BRR 中真实值, 并利用 $\text{USARTDIV} = \frac{\text{PCLK}}{16 \times \text{Band Rate}}$