

时钟管理单元

6

- Exynos 4412时钟管理单元工作原理
- 时钟管理单元控制寄存器
- 应用实例





6.1 时钟管理单元工作原理

时钟管理单元(Clock Management Unit, CMU)是Cortex-A9用来管理其内部各个时钟模块的核心单元。Exynos 4412的时钟管理单元通过PLLs(Phase Locked Loops, 锁相环)产生CPU、总线和外设所需的各种频率的时钟信号。同时,时钟管理单元也与电能管理单元关联,通过在低功耗模式下关闭时钟的方法,来降低整个系统的功耗。



第6章 时钟管理单元

6.1.1 Exynos4412的时钟体系

在一块ARM芯片里，往往集成有多个不同的功能模块，这些功能模块在各自时钟频率下表现性能最优。为了让整块ARM芯片功耗最低，性能最好，只能采取多时钟频率的方法，才能达到要求。

第6章 时钟管理单元

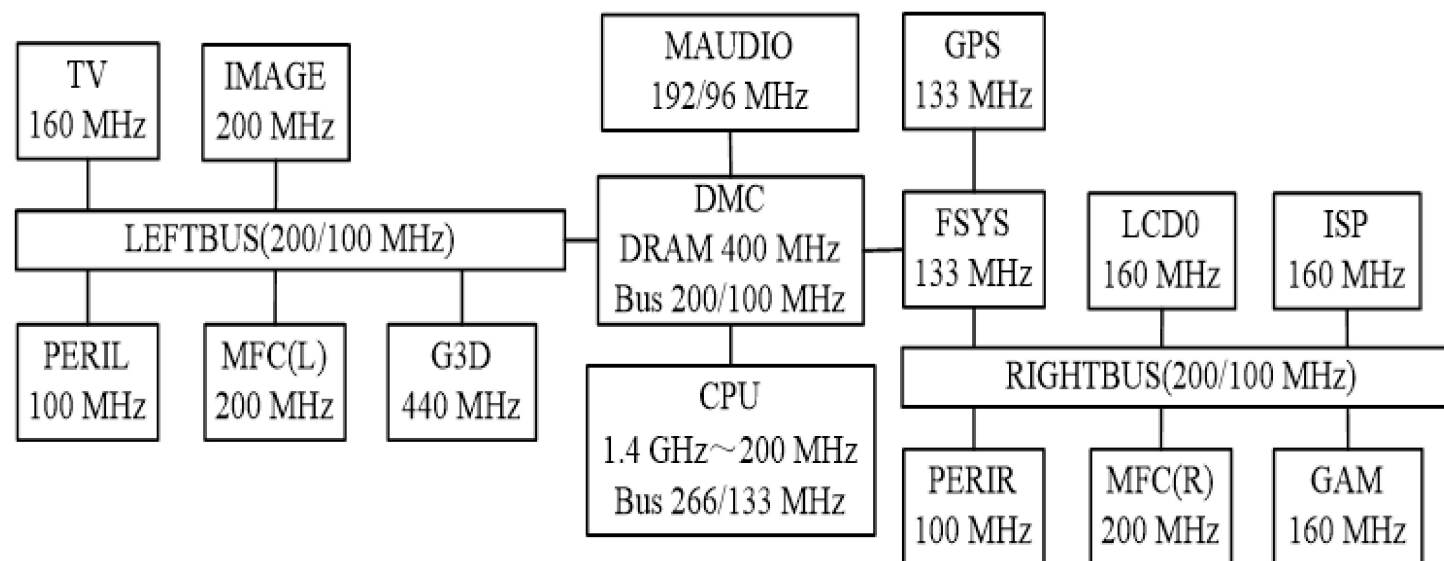
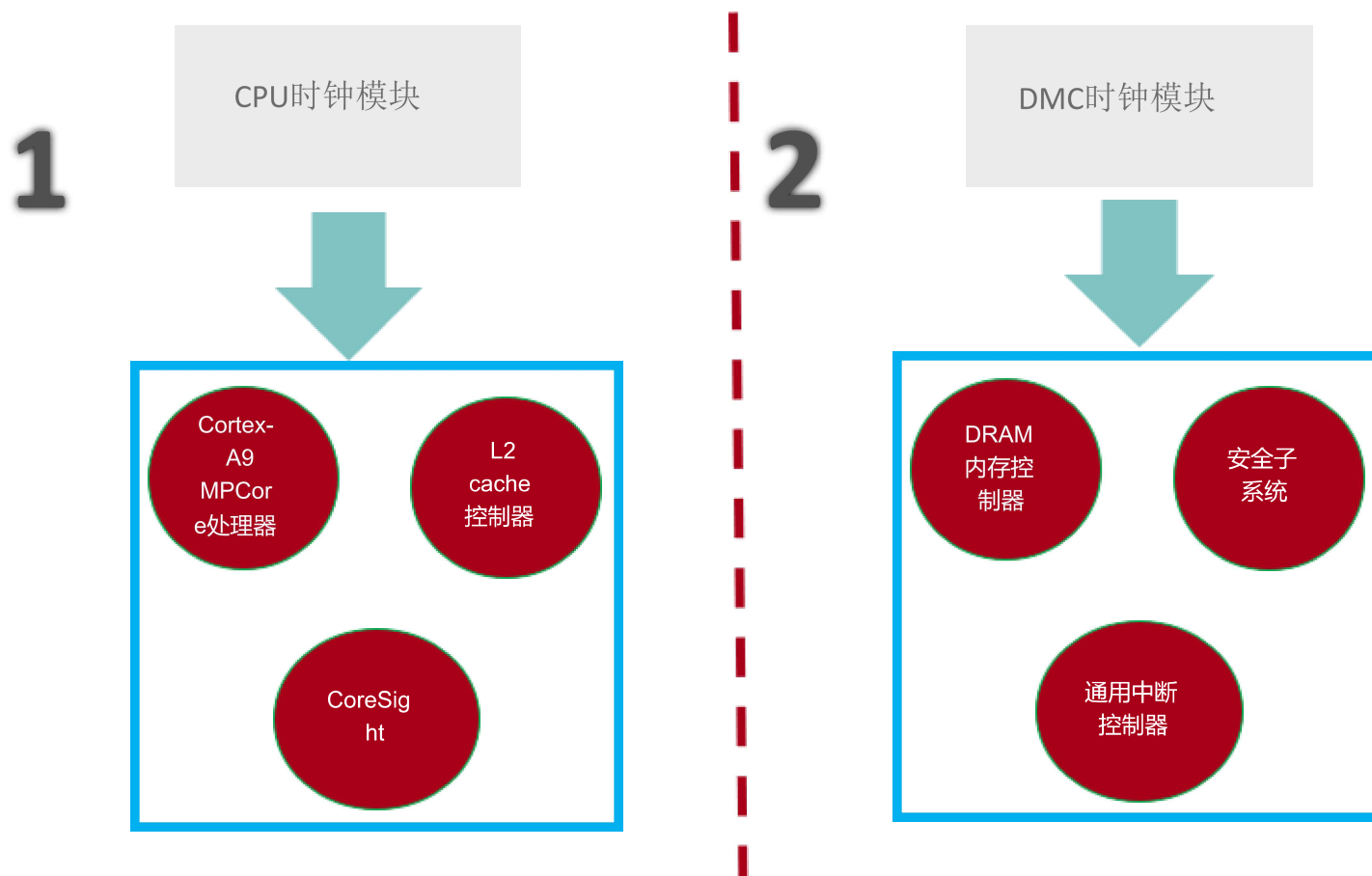


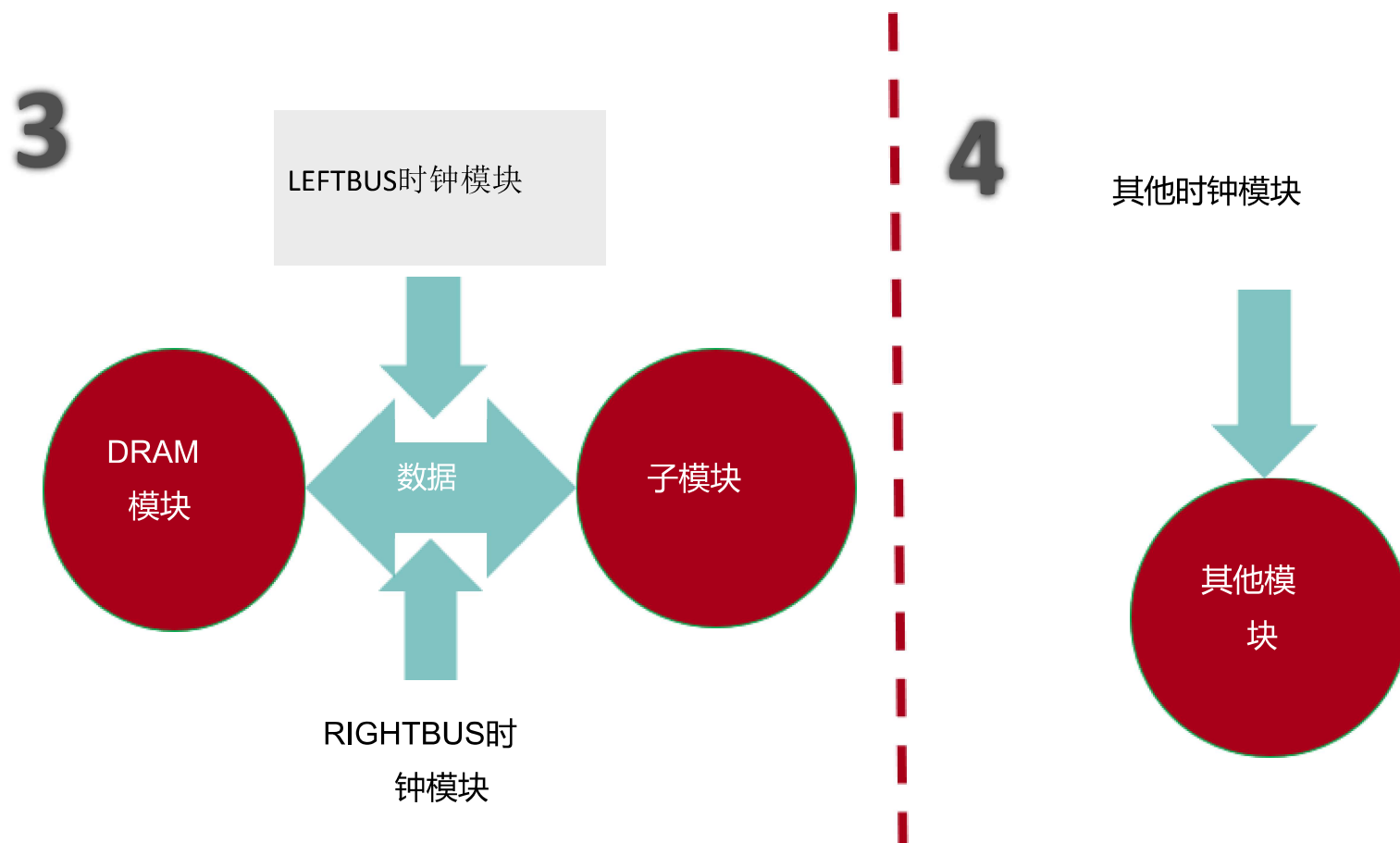
图6.1 Exynos 4412时钟体系框图

第6章 时钟管理单元

内部时钟:



第6章 时钟管理单元





第6章 时钟管理单元

外部时钟:

外部引脚接入的初始时钟源有3个, 分别是:

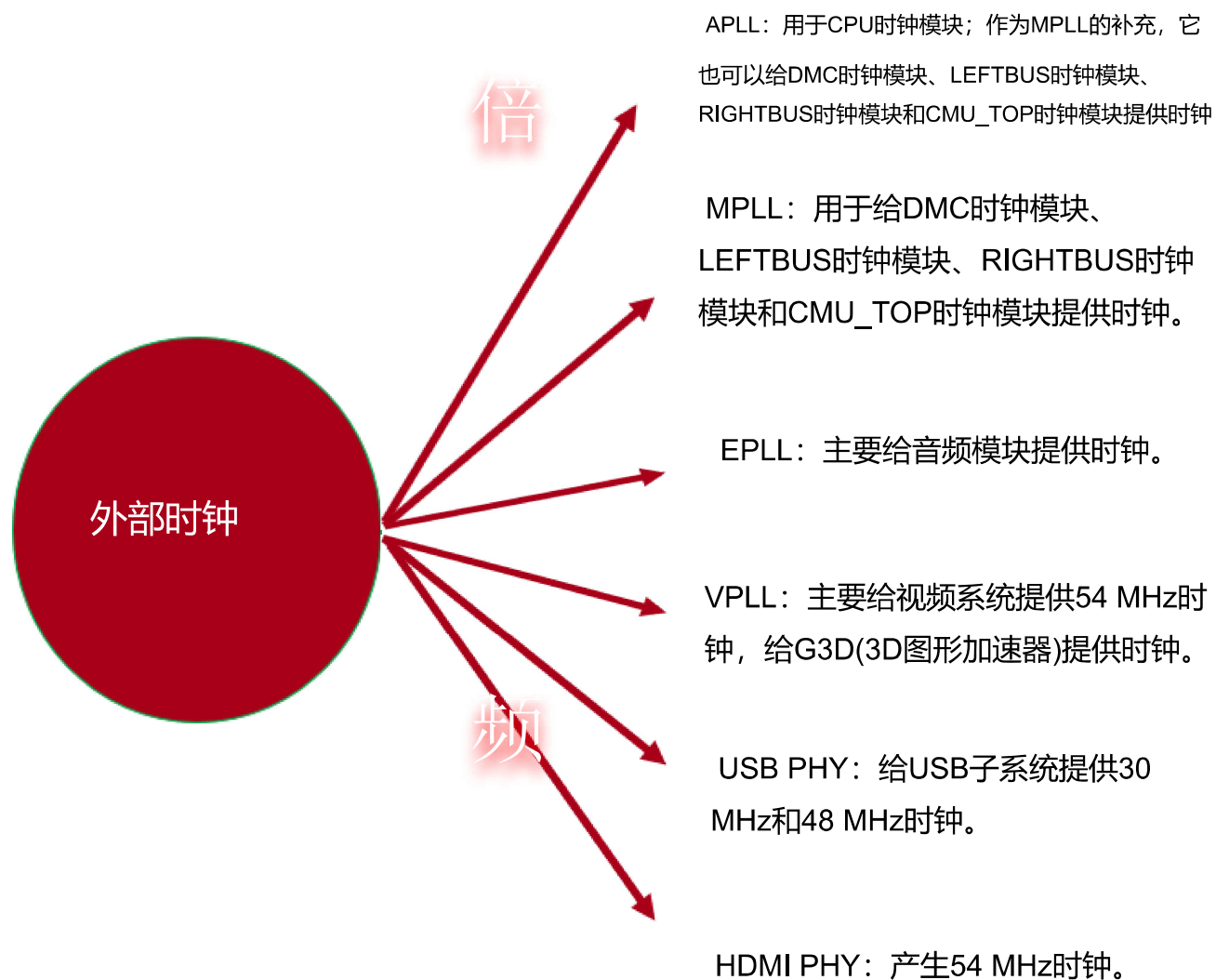
(1) XRTCXTI引脚: 接32.768 kHz的晶振, 用于向实时时钟

(RTC)提供初始晶振。

(2) XXTI引脚: 接12~50 MHz的晶振, 用于向系统提供时钟。不使用时, 应该下拉。

(3) XUSBXTI引脚: 由引脚X_{usb}XTI引入, 为APLL、MPLL、VPLL、EPLL和USB-PHY提供基础时钟。

第6章 时钟管理单元





第6章 时钟管理单元

6.1.2 Exynos4412时钟管理单元工作过程

1、倍频

由于外部晶振从开始起振到稳定的期间，PLL将CPU的频率锁定一段时间，在这期间内，需要将内部使用的频率锁定抬升以后才能供这些模块使用。因此，在锁定频率后，还需要设置一个倍频因子，抬升从外部晶振输入的时钟频率。



第6章 时钟管理单元

2、分频

通过设置分频相关寄存器中不同的位即可实现分频。

比如，设置某位分频值为4，那么，分频时，原来频率假设为1000 MHz，则该频率就被分为 $1000/(1 + 4) = 200$ MHz。

第6章 时钟管理单元

在选用倍频、分频值时，针对APLL、MPLL、EPLL和VPLL有不同的推荐值，应该参考表6.1～表6.3，从推荐值中选择倍频、分频函数P、M、S相近的组合方式。

F_{IN}/MHz	Target F_{OUT}/MHz	P	M	S	F_{OUT}/MHz
24	200	3	100	2	200
24	300	4	200	2	300
24	400	3	100	1	400
24	500	3	125	1	500
24	600	4	200	1	600
24	700	3	175	0	700
24	800	3	100	0	800
24	900	4	150	0	900
24	1000	3	125	0	1000
24	1100	6	275	0	1100
24	1200	4	200	0	1200
24	1300	6	325	0	1300
24	1400	3	175	0	1400

表6.1 APLL和MPLL的P、M、S推荐值



第6章 时钟管理单元

表6.2 EPLL的P、M、S推荐值

F_{IN}/MHz	Target F_{OUT}/MHz	P	M	S	F_{OUT}/MHz
24	90	2	60	3	90
24	180	2	60	2	180
24	180.6	3	90	2	180.6
24	180.6336	3	90	2	180.6336
24	192	2	64	2	192
24	200	3	100	2	200
24	400	3	100	1	400
24	408	2	68	1	408
24	416	3	104	1	416



第6章 时钟管理单元

表6.3 VPLL的P、M、S推荐值

F_{IN}/MHz	Target F_{OUT}/MHz	P	M	S
24	100	3	100	3
24	160	3	160	3
24	266	3	133	2
24	350	3	175	2
24	440	3	110	1

第6章 时钟管理单元

3、Exynos 4412配置时钟的方法

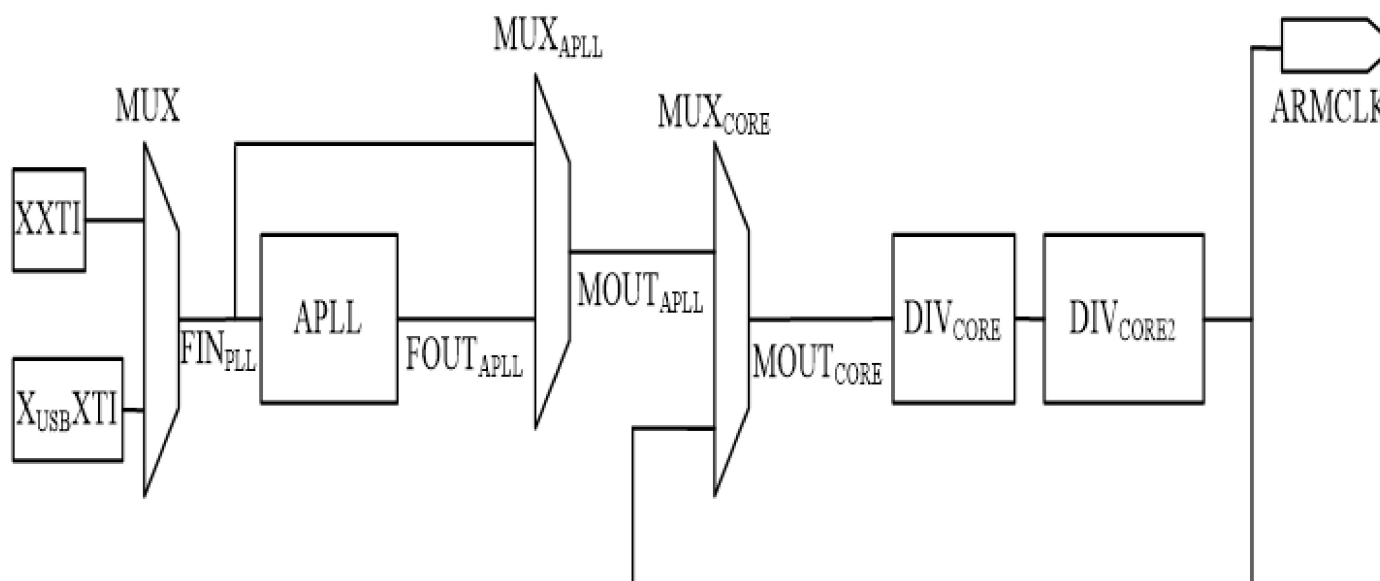


图6.2 ARMCLK时钟频率产生的原理框图

第6章 时钟管理单元

设置PLL的流程如下：

- (1) 设置PLL的P、M、S值，这是根据期望得到的频率用公式计算出来的，一般从推荐表中选择相近的组合。
- (2) 设置PLL的其他控制参数。
- (3) 使能PLL。
- (4) 设置锁定时间，PLL等待一段时间，使得时钟稳定。
- (5) 设置MUX，选择PLL输入时钟。如果当前正使用该PLL，那么先设置MUX改用其他时钟源或禁止使用此PLL，设置完PLL后再设置MUX改回原来PLL所使用的时钟源。





6.2 时钟管理单元控制寄存器

Exynos 4412中，与时钟管理单元相关的寄存器多达163个。按这些寄存器的映射地址可以分为6个部分，分别称为：CMU_LEFTBUS、CMU_RIGHTBUS、CMU_TOP、CMU_DMC、CMU_CPU和CMU_ISP。这6个部分各有16 KB地址空间容纳其相关的寄存器。这些寄存器在各自16 KB地址空间中的分布方式是类似的。

第6章 时钟管理单元

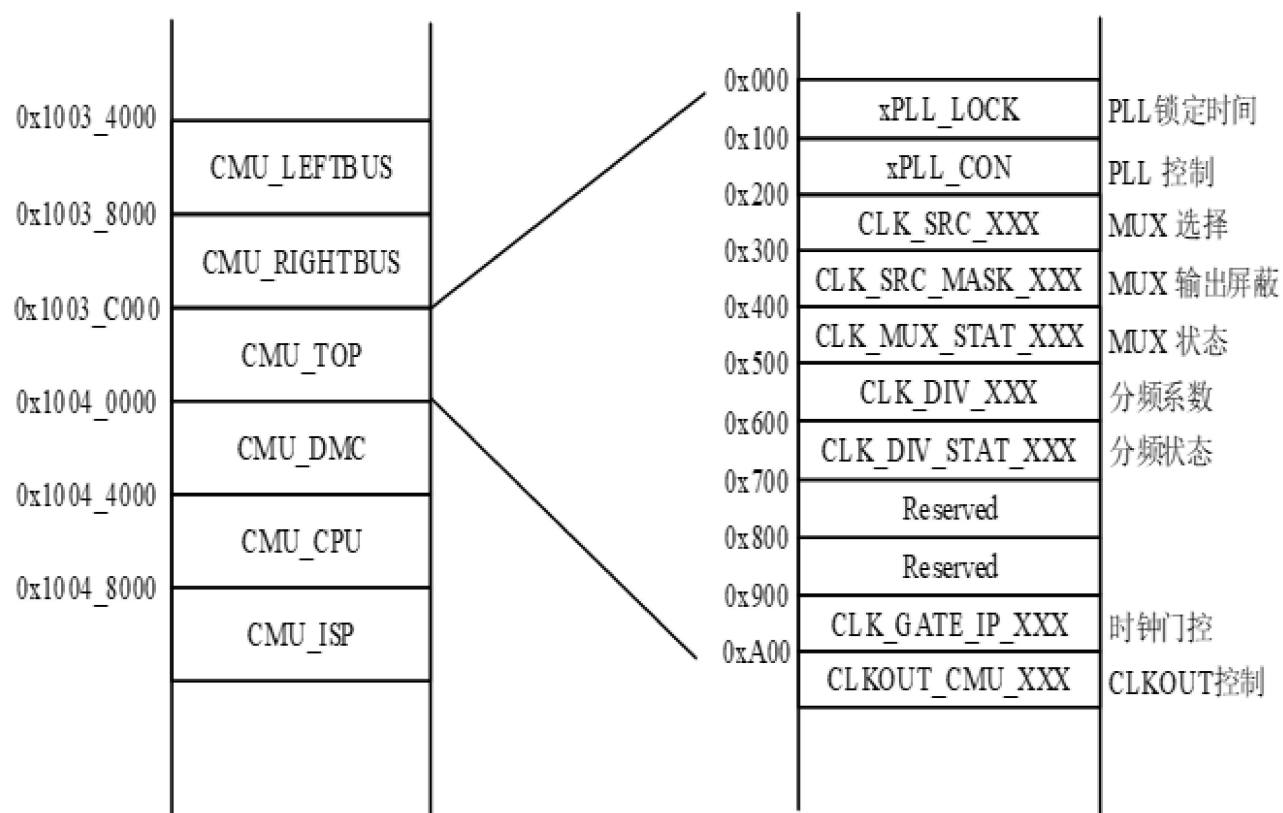


图6.3 Exynos 4412时钟控制器的地址映射



第6章 时钟管理单元

下面以CMU_CPU单元的设置为主线，介绍与其配置相关的寄存器的使用方法，具体步骤如下：

1. 初次选择时钟输入源

表6.5列出了CMU_CPU模块中，用于选择时钟源的4个多路开关的控制位。为了使用24 MHz的外部时钟进行倍频，查看表6.5，需设置MUX_APLL_SEL为0(选择FIN_{PLL})，MUX_CORE_SEL也需设置成0，选择MOUT_{APLL}。其他的两位也都设置为0，分别将MUX_MPLL_USER_SEL_C选择FIN_{PLL}、MUX_HPM_SEL选择MOUT_{APLL}。设置完成后需要等待一定时间，让其设置成功。

第6章 时钟管理单元

名 称	位域	类型	功 能 描 述	复位值
MUX_MPLL_USER_SEL_C	[24]	RW	控制 MUX_{MPLL} 。0 = FIN_{PLL} , 1 = $FOUT_{MPLL}$	0
MUX_HPM_SEL	[20]	RW	控制 MUX_{HPM} 。0 = $MOUT_{APLL}$, 1 = $SCLK_{MPLL}$	0
MUX_CORE_SEL	[16]	RW	控制 MUX_{CORE} 。0 = $MOUT_{APLL}$, 1 = $SCLK_{MPLL}$	0
MUX_APLL_SEL	[0]	RW	控制 MUX_{APLL} 。0 = FIN_{PLL} , 1 = $MOUT_{APLL}$	0

表6.5 时钟源选择寄存器CLK_SRC_CPU

第6章 时钟管理单元

上表中各寄存器各位的作用：

① BIT[0]控制第1个多路复用开关 MUX_{APLL} ，用于选择是使用 FIN_{PLL}

还是 $APLL$ 的输出时钟 $MOUT_{APLL}$ 。

② BIT[16]控制第2个多路复用开关 MUX_{CORE} ，通过设置0或1，选择

$MOUT_{APLL}$ 或者 $SCLK_{MPLL}$ ，以确定该级多路复用开关的输入频率源。

其中 $SCLK_{MPLL}$ 由 MUX_{MPLL} 控制。

③ BIT[20]控制第3个多路复用开关 MUX_{HPM} ，用于选择 $MOUT_{APLL}$ 或

者 $SCLK_{MPLL}$ ，以确定该级多路复用开关的输入频率源。

④ BIT[24]控制第4个多路复用开关 MUX_{MPLL} ，用于选择 FIN_{PLL} 或者

$MOUT_{MPLL}$ 。该级多路开关的输出被称为 $SCLK_{MPLL}$ 。其中， $FOUT_{MPLL}$

来自 $MPLL$ 的输出。

第6章 时钟管理单元

2. 设置APLL/MPLL/EPLL/VPLL锁相环的锁频时间

设置APLL的参数并使能它后，APLL并不能立刻输出稳定的时钟，它需要经历一个锁定时间(Lock Time)。APLL 的最大锁定时间是 $(270 \times \text{PDIV})$ 个周期。所以将APLL_LOCK寄存器相关位域设置为 $(270 \times \text{PDIV})$ 就可以了，见表6.6。

名称	位域	类型	功能描述	复位值
锁定时间	[15:0]	RW	用于指定 PLL 的锁定时间	0xFFFF

表6.6 APLL锁定时间设置寄存器APLL_LOCK寄存器



第6章 时钟管理单元

3.倍频

APLL/MPLL/EPLL/VPLL的倍频值需根据需求参考表6.1～

6.3中的推荐值进行设置。根据M、P、S的值，可以算出

APLL的输出时钟。比如，当选取推荐值M、P、S的组合为

175、3、0时，输出的频率为：

$$F_{OUT} = \frac{M \times F_{IN}}{P \times 2^S} = \frac{175 \times 24}{3 \times 2^0} = 1400 \text{ MHz}$$

第6章 时钟管理单元

名称	位域	类型	功能描述	复位值
使能	[31]	RW	用于使能 APLL。0=禁止, 1=使能	0
锁定状态	[29]	R	用于显示 PLL 的锁定状态。0=未锁定, 1=已锁定(表示 PLL 输出已稳定)	0
FSEL	[27]	RW	监测频率选择引脚。0= $F_{VCO_OUT} = F_{REF}$, 1= $F_{VCO_OUT} = F_{VCO}$	0
M	[25:16]	RWX	M分频值	0xC8
P	[13:8]	RWX	P分频值	0x6
S	[2:0]	RWX	S分频值	0x1

表6.7 APLL控制寄存器APLL_CON0



第6章 时钟管理单元

4.重新选择时钟源

在分频之前，必须重新选择时钟源。重新选择时钟源的寄存器配置方法和前面介绍的一样，只是这里需根据新选择的时钟源对相应的寄存器值进行设置，这一次我们需要选择倍频后的时钟。设置完成后，同样读取相应的状态寄存器，和其对应的值进行比较，以确定是否完成设置。选择完成时钟源后，就可以开始进行分频了。分频值需要根据表6.8、表6.9列出的分频寄存器CLK_DIV_CPU0、CLK_DIV_CPU1的相应位进行配置。

第6章 时钟管理单元

名 称	位域	类型	功 能 描 述	复位值
CORE2_RATIO	[30:28]	RW	$ARMCLK = DOUTCORE / (CORE2_RATIO + 1)$	0x0
APLL_RATIO	[26:24]	RW	$SCLKAPLL = MOUTAPLL / (APLL_RATIO + 1)$	0x0
PCLK_DBG_RATIO	[27]	RW	$PCLK_DBG = ATCLK / (PCLK_DBG_RATIO + 1)$	0x0
ATB_RATIO	[18:16]	RW	$ATCLK = MOUTCORE / (ATB_RATIO + 1)$	0x0
PERIPH_RATIO	[14:12]	RW	$PERIPHCLK = DOUTCORE / (PERIPH_RATIO + 1)$	0x0
COREM1_RATIO	[10:8]	RW	$ACLK_COREM1 = ARMCLK / (COREM1_RATIO + 1)$	0x0
COREM0_RATIO	[6:4]	RW	$ACLK_COREM0 = ARMCLK / (COREM0_RATIO + 1)$	0x0
CORE_RATIO	[2:0]	RW	$DIVCORE_OUT = MOUTCORE / (CORE_RATIO + 1)$	0x0

表6.8 CLK_DIV_CPU0控制寄存器

第6章 时钟管理单元

名称	位域	类型	功能描述	复位值
CORES_RATIO	[10:8]	RW	$ACLK_CORES = ARMCLK / (CORES_RATIO + 1)$	0x0
HPM_RATIO	[6:4]	RW	$SCLK_HPM = DOUTCOPY / (HPM_RATIO + 1)$	0x0
COPY_RATIO	[2:0]	RW	$DOUTCOPY = MOUTHPM / (COPY_RATIO + 1)$	0x0

表6.9 CLK_DIV_CPU1控制寄存器

第6章 时钟管理单元

下面以CPU的工作频率ARMCLK配置为例，计算ARMCLK的频率：

$$\begin{aligned}\text{ARMCLK} &= \text{MUX}_{\text{CORE}}\text{的输出} / \text{DIV}_{\text{CORE}} / \text{DIV}_{\text{CORE2}} \\ &= \text{MOUT}_{\text{CORE}} / (\text{CORE_RATIO} + 1) / \text{DIV}_{\text{CORE2}} \\ &= \text{MOUT}_{\text{CORE}} / (\text{CORE_RATIO} + 1) / (\text{CORE2_RATIO} + 1)\end{aligned}$$

其中， $\text{MOUT}_{\text{CORE}}$ 表示 MUX_{CORE} 的输出，在 MUX_{APLL} 和 MUX_{CORE} 都设置为0时，它等于 $M \times \text{FIN} / (P \times 2^S)$ 。





6.3 应用实例

通过Exynos 4412的一段时钟初始化程序，介绍时钟管理单元部分的设置方法。程序的编写思路按照6.2节的四个步骤进行。在代码中，我们采用注释的形式对代码进行解释说明，具体代码见课本所示。



问题与思考:



1. Exynos 4412 有几种外部输入时钟？分别是什么？各自的服务对象是什么？
2. Exynos 4412的5个模块分别是什么？
3. 以PCLK时钟频率(即ACLK_100)的配置为主线，参考Exynos 4412的手册画出其产生过程，并通过编程将其配置成100 MHz的时钟频率输出。

