

### توجه :

ویرایش قبلی این مقاله قبلا در مجله ی PMM ( مجله میکروکنترلر فارسی - Persian Microcontroller Magazine ) شماره 4 منتشر شده بود . ویرایشی که همکنون مشاهده میکنید برای کتاب " مرجع کامل میکروکنترلرهای سری AT91SAM شرکت ATMEL " تهیه شده است .

برای کسب اطلاعات بیشتر در مورد این کتاب پست شماره 11 تاپیک زیر را مطالعه کنید :

<http://www.iranmicro.ir/forum/showthread.php?t=12189>

جهت ارتباط با نویسنده به آدرس زیر مراجعه کنید :

<http://www.iranmicro.ir/forum/member.php?u=3640>

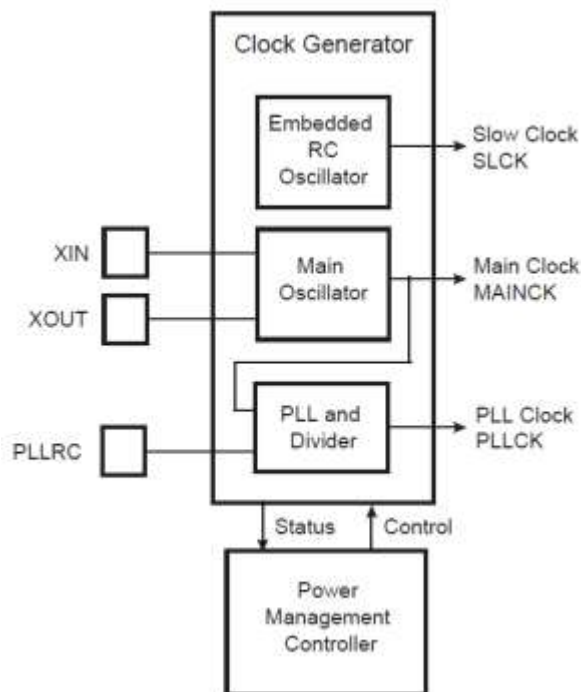
قبل از مطالعه مطالب این بخش شما باید بخش های 1 این کتاب که دارای مطالب زیر است را مطالعه کرده باشید:

1- مباحث مقدماتی : در این بخش شما با میکروکنترلر های مبتنی بر هسته ی ARM آشنا شده و نحوه ی استفاده از آنها در کامپایلر KEIL را فرا گرفته و بعد از آشنایی با برخی از دستورات زبان C ، می آموزید که چگونه از ورودی/خروجیهای این میکروکنترلرها استفاده کرده و چگونه کتابخانه های مورد نیاز خود را ایجاد کنید . (نویسنده : 1nafar )

## راه اندازی واحد های PMC و CLKG:

واحد های PMC ( Power Management Controller ) و CLKG ( Clock Generator ) از جمله قسمت های اصلی میکروکنترلرهای سری AT91SAM می باشند که وظیفه ی تولید کلاک سیستم و مدیریت توان مصرفی بخش های جانبی را بر عهده دارند .

کلاک مورد نیاز برای بخش های مختلف میکروکنترلر میتواند از اسیلاتور اصلی ( Main Oscillator ) یا کریستال متصل شده به میکروکنترلر ( یا واحد PLL یا اسیلاتور RC تامین شود:

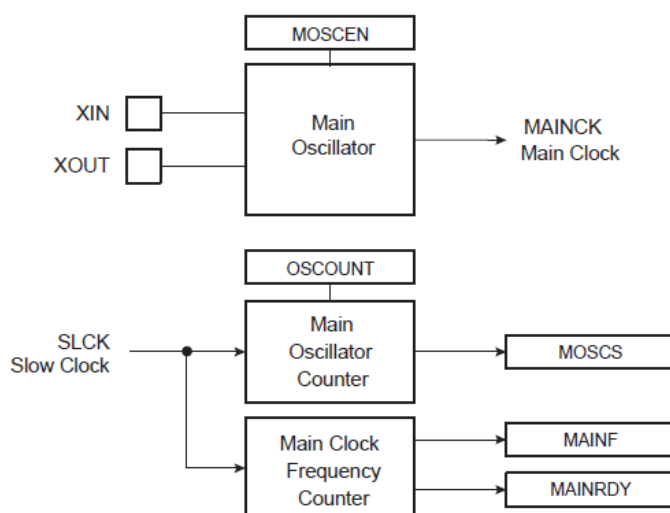


- SLCK : این کلاک که مخفف کلمات Slow Clock می باشد ، یک کلاک با فرکانس پایین ( تقریباً 22 الی 42 کیلو هرتز ) می باشد که برای مصارفی مانند ساعت بلادرنگ (Real-time Timer (RTT) و تامین کلاک CPU قبل از پایدار شدن کلاک اسیلاتور اصلی و... مورد استفاده قرار می گیرد . البته لازم به ذکر است این کلاک هم می تواند به عنوان کلاک کل سیستم معرفی شود .
- MAINCK : کلاک خروجی اسیلاتور Main می باشد که مقداری برابر با کریستال متصل شده به پایه های XIN و XOUT دارد .
- PLLCK : خروجی کلاک قسمت PLL می باشد که معمولاً ضرب شده ی کلاک ورودی خود است و برای تولید کلاک هایی با فرکانس بالاتر از فرکانس کریستال متصل شده به میکروکنترلر استفاده می شود .

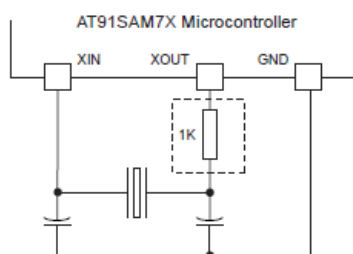
## اسیلاتور اصلی سیستم :

در شکل زیر دیاگرام اسیلاتور اصلی سیستم را مشاهده می کنید ، بخش بالای تصویر بلوک دیاگرام کلی واحد تولید کلاک و بخش پایین تصویر بلاک های مربوط به تولید زمان Startup اسیلاتور ، شمارش فرکانس اسیلاتور و ... را نمایش میدهد. خوشبختانه در کامپایلر KEIL کلیه تنظیمات مربوط به اسیلاتور و بخش های جانبی آن به صورت کرافیکی در دسترس کاربر قرار دارد و مطالبی که در ادامه آورده شده است صرفا جهت آشنایی هرچه بیشتر و استفاده در پروژه های حرفه ای میباشد .

Main Oscillator Block Diagram



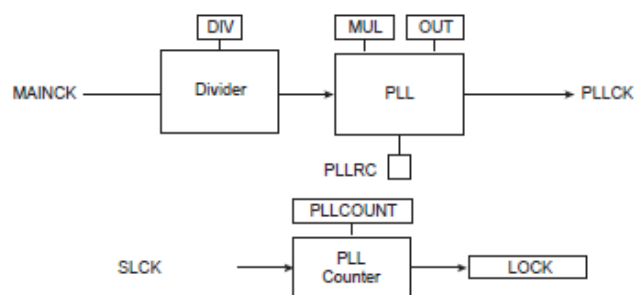
همانطور که در تصویر مشاهده می کنید، پایه های ورودی و خروجی کریستال کوآرتز که از خارج به میکروکنترلر متصل میشوند ، مستقیما به اسیلاتور اصلی متصل خواهند شد . برای میکروکنترلر های سری AT91SAM حداکثر فرکانس کریستال متصل شده به پایه های مربوط به کریستال بین 3 تا 20 مگاهرتز میباشد ، همچنین کریستال باید مانند تصویر زیر به میکروکنترلر متصل شود . استفاده از کریستال های با فرکانس بالاتر به دلیل افزایش خطر نویز پذیری توسط شرکت اتمل توصیه نشده است .



همچنین کاربر میتواند یک سیگنال کلاک خارجی را در عوض اتصال کریستال ، به میکروکنترلر بدهد . در این حالت کلاک موردنظر باید به پایه ی Xin اعمال شود ، به این حالت بای پاس کردن اسیلاتور Main گفته می شود .

### تقسیم کننده و بلاک PLL:

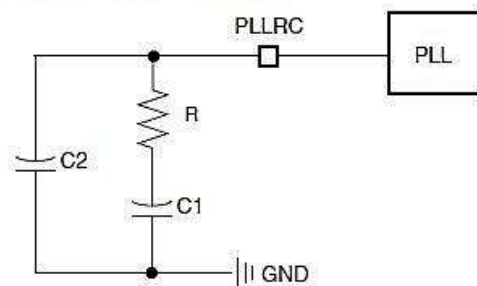
بعد از اسیلاتور اصلی ، واحد PLL برای ضرب فرکانس ورودی خود و افزایش فرکانس آن قرار داده شده است . در این بلاک یک تقسیم کننده نیز وجود دارد که برای افزایش دقت فرکانس خروجی PLL کاربرد دارد . توجه داشته باشید که این تقسیم کننده به صورت کاملاً مجزا نسبت به تقسیم کننده ای که در بخش های بعد توضیح داده خواهد شد ، عمل می کند .



### فیلتر PLL:

مدارات داخلی PLL به یک فیلتر نیاز دارند که باید از طریق پایه ی PLLRC ، به صورت شکل زیر قرار گیرند . مقادیر خازن و مقاومت بستگی به یک سری پارامترهای فعلی PLL دارد و باید محاسبه شوند . اما در شرایط معمولی  $R = 5k.1$  ،  $C1 = 10n$  و  $C2 = 1n$  مناسب هستند . در ضمیمه شماره 3 ، توضیحات بیشتری در این باره آورده شده است .

#### PLL Capacitors and Resistors



اما گفته شد که تقسیم کننده ، دقت فرکانس خروجی PLL را بالا می برد ، با گفتن یک مثال علت قرار دادن تقسیم کننده در این واحد را توضیح می دهیم . فرض می کنیم این تقسیم کننده وجود نداشت و می خواهیم مقادیری را برای PLL حساب کنیم که فرکانس 18.432 مگاهرتز ورودی خود را به فرکانس 72 مگاهرتز تبدیل کند . بهترین عدد ، 4 می باشد لذا

$$18.432 * 4 = 73.728$$

که فرکانس خروجی خطایی حدود 2.39 % دارد . اما با وجود تقسیم کننده و گذاشتن مقادیر 32 برای تقسیم کننده و 125 برای ضرب کننده ، فرکانس خروجی دقیقا 72 مگاهرتز خواهد شد .

$$( 18.432 * 125 ) / 32 = 72$$

این ویژگی زمانی دارای اهمیت می شود که بخواهیم برای مثال فرکانس 22 مگاهرتز تولید کنیم . در این صورت بدون در نظر گرفتن تقسیم کننده ، خطا به حداقل 26 % می رسد که با وجود تقسیم کننده خطا بسیار ناچیز و حدود 0.001 % می شود .

مقدار تقسیم کننده می تواند بین 0 تا 255 باشد . در صورتیکه مقدار 0 داشته باشد ، خروجی تقسیم کننده غیر فعال ( خاموش ) می شود . مقدار فیلد ضرب کننده ی PLL نیز در صورتیکه 0 باشد ، خروجی غیر فعال می شود . فرکانس خروجی مجموعه ی PLL با در نظر داشتن مقادیر پارامترهای تقسیم ( DIV ) و ضرب ( MUL ) از طریق رابطه ی زیر محاسبه می شود .

$$PLL_{fout} = ((1 + MUL) / DIV) * F_{osc}$$

در میکروکنترلر های سری at91sam برای واحد CLKG سه ثبات ( رجیستر ) به شرح زیر تعریف شده است که کاربر توسط آنها میتواند فرکانس کاری میکروکنترلر را مشخص کند :

#### 1- رجیستر CKGR\_MOR ( PMC Clock Generator Main Oscillator Register ) :

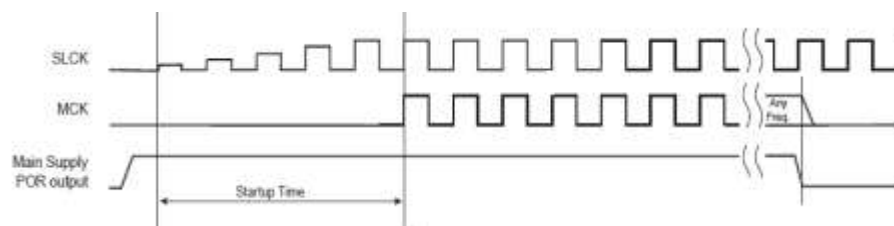
31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
OSCCOUNT							
7	6	5	4	3	2	1	0
-	-	-	-	-	-	OSCBYPASS	MOSCEN

بیت های OSCBYPASS: Oscillator Bypass و MOSCEN: Main Oscillator Enable :

یک بودن بیت MOSCEN و صفر بودن بیت OSCBYPASS باعث فعال شدن اسیلاتور اصلی سیستم و صفر بودن بیت MOSCEN باعث غیر فعال شدن اسیلاتور خواهد شد. در صورتی که MOSCEN صفر شود، بیت OSCBYPASS باید یک باشد تا کلاک مورد نیاز سیستم از پالس کلاک خارجی که به پایه XIN اعمال میشود، تامین گردد.

بیت های OSCOUNT: Main Oscillator Start-up Time :

بعد از ریست شدن میکروکنترلر (ریست شدن به صورت دستی یا قطع و وصل شدن ولتاژ تغذیه یا ...) کلاک مورد نیاز CPU از طریق SLCK (نوسان ساز RC داخلی) تامین میشود. در همین حین در صورت یک بودن بیت MOSCEN یا بیت OSCBYPASS، اسیلاتور اصلی سیستم فعال شده و با استفاده از کلاک ایجاد شده توسط کریستال خارجی (1=MOSCEN) یا با استفاده از پالس اعمالی به پایه XIN (1=OSCBYPASS) کلاک مورد نیاز CPU را تولید میکند. به زمانی که طول میکشد تا اسیلاتور فعال شده و فرکانس خروجی آن به مقدار تعیین شده برسد (پایدار شود)، زمان راه اندازی یا Start-up Time گفته میشود. در این زمان CPU نباید کلاک خود را از اسیلاتور اصلی تعیین کند، چون فرکانس خروجی آن هنوز به حالت پایدار نرسیده و مقدار آن مشخص نیست.



با مقدار دهی بیت های OSCOUNT میتوان زمان Start-up را مشخص نمود، مقدار این زمان از رابطه ی زیر محاسبه میشود:

$$\text{Start Up Time} = 8 * \text{OSCOUNT} / \text{SLCK} = 56 \text{ Slow Clock Cycles.}$$

در صورتی که تمامی بیت های OSCOUNT را یک کنیم (آن را با 0xFF یا 255 مقدار دهی کنیم) و فرکانس نامی SLCK (نوسان ساز RC داخلی) برابر با 32 کیلو هرتز باشد، زمان Start-up برابر با 64 میلی ثانیه خواهد بود. زمان Start-up بعد از یک شدن بیت MOSCEN آغاز شده و بعد از سپری شدت آن بیت MOSCS یک میشود.

2- رجیستر CKGR\_MCFR (PMC Clock Generator Main Clock Frequency Register) :

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	MAINRDY
15	14	13	12	11	10	9	8
MAINF							
7	6	5	4	3	2	1	0
MAINF							

**MAINRDY: Main Clock Ready** بیت

یک بودن این بیت به معنای فعال بودن اسیلاتور اصلی و صحیح بودن مقدار بیت های MAINF میباشد ، صفر بودن این بیت به معنای غیر فعال بودن اسیلاتور اصلی است . در صورت صفر بودن این بیت ، مقدار فیلد MAINF نامعتبر است .

بیت های MAINF: Main Clock Frequency :

همانطور که قبلا گفته شد ، واحد نوسانساز دارای یک شمارشگر داخلی است که میتواند تعداد پالس های کلاک ورودی ( دریافتی از کریستال ) را شمارش کند ، بیت های MAINF در بر گیرنده تعداد پالس های شمرده شده یا فرکانس کریستال متصل شده به میکروکنترلر هستند .

بعد از فعال شدن نوسان ساز اصلی و یک شدن بیت MOSCS تعداد پالس های دریافتی از ورودی کریستال در 16 سیکل کاری نوسانساز فرکانس پایین شمرده میشود ، در صورتی که فرکانس نوسان سازی داخلی میکروکنترلر برابر با 32.768 کیلوهرتز باشد ، تعداد پالس های دریافتی در مدت زمان 0.00048828125 ثانیه شمرده میشود . در صورتی که این عدد را به پالس های شمرده شده (عدد موجود در فیلد MAINF) تقسیم کنیم ، میتوانیم مقدار زمان تناوب پالس های دریافتی از کریستال متصل شده به میکروکنترلر را محاسبه کنیم . با تقسیم کردن عدد یک بر زمان تناوب میتوان مقدار فرکانس کریستال را محاسبه کرد .

**3- رجیستر CKGR\_PLLR ( PMC Clock Generator PLL Register ) :**

31	30	29	28	27	26	25	24
-	-	USB DIV		-	MUL		
23	22	21	20	19	18	17	16
MUL							
15	14	13	12	11	10	9	8
OUT		PLL COUNT					
7	6	5	4	3	2	1	0
DIV							

همانطور که قبلا نیز ذکر شد، برای واحد pll یک مقسم و یک ضریب در نظر گرفته شده است، که به ترتیب میتوان در فیلد های DIV (Divider) و MUL (PLL Multiplier) آنها را مقدار دهی کرد. قرار دادن عدد صفر در فیلد های DIV و MUL به ترتیب باعث غیر فعال شدن مقسم و غیر فعال شدن PLL میشود.

بیت های USB DIV (Divider for USB Clock):

برای عمل کرد صحیح میکروکنترلر در حالت USB device باید کلاک واحد USB برابر با 48 مگاهرتز باشد. با مقدار دهی بیت USB DIV میتوان کلاک پردازنده را علاوه بر 48 مگاهرتز بر روی 96 مگاهرتز و یا 192 مگاهرتز نیز تنظیم کرد، در این حالت باید مطابق جدول زیر بیت های USB DIV را طوری مقدار دهی کرد که فرکانس کلاک اعمالی به واحد USB برابر با 48 مگاهرتز شود:

USB DIV		Divider for USB Clock(s)
0	0	Divider output is PLL clock output.
0	1	Divider output is PLL clock output divided by 2.
1	0	Divider output is PLL clock output divided by 4.
1	1	Reserved.

بیت های OUT (PLL Clock Frequency Range):

با مقدار دهی این بیت ها میتوان رنج فرکانس کاری واحد PLL را تعیین کرد. رنج فرکانس کاری PLL برای میکروکنترلر های مختلف متفاوت بوده و معمولا در جدول "PLL Characteristics" در دیتاشیت میکروکنترلر آورده میشود، در زیر جدول "PLL Characteristics" مربوط به میکروکنترلر AT91SAM7X256 را مشاهده میکنید:



## 38.5 PLL Characteristics

Table 38-12. Phase Lock Loop Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
F <sub>OUT</sub>	Output Frequency	Field out of CKGR_PLL is: 00	80		160	MHz
		10	150		200	MHz
F <sub>IN</sub>	Input Frequency		1		32	MHz
I <sub>PLL</sub>	Current Consumption	Active mode			4	mA
		Standby mode			1	μA

Note: Startup time depends on PLL RC filter. A calculation tool is provided by Atmel.

همانطور که مشاهده میکنید، با مقدار دهی بیت های OUT با 00 واحد PLL در محدوده فرکانس 80 تا 160 مگاهرتز و با مقدار این بیت ها با عدد 10 در محدوده 150 تا 200 مگاهرتز پیکربندی میشود.

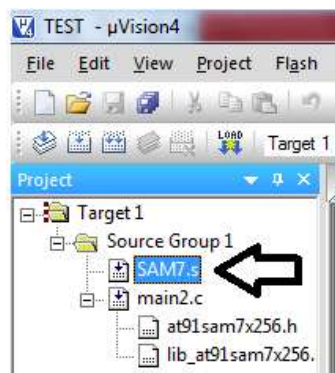
بیت های PLLCOUNT (PLL Counter):

بعد از فعال شدن PLL و ایجاد شدن پالس در خروجی آن، مدت کوتاهی طول میکشد تا فرکانس این پالس ها به فرکانس تعیین شده برسد. در این حالت کاربر باید در برنامه نوشته شده، منتظر پایدار شدن فرکانس خروجی PLL بماند، بعد از پایدار شدن فرکانس خروجی PLL بیت LOCK در رجیستر PMC\_SR فعال میشود. با مقدار دهی بیت های PLLCOUNT میتوان تاخیری را در یک شدن بیت LOCK ایجاد کرد. این تاخیر بعد از پایدار شدن خروجی PLL شروع شده و بعد از اتمام آن بیت LOCK یک میشود. مقدار تاخیر برابر با  $PLL\_COUNT * 30 \mu s$  (تعداد پالس های کلاک) میباشد.

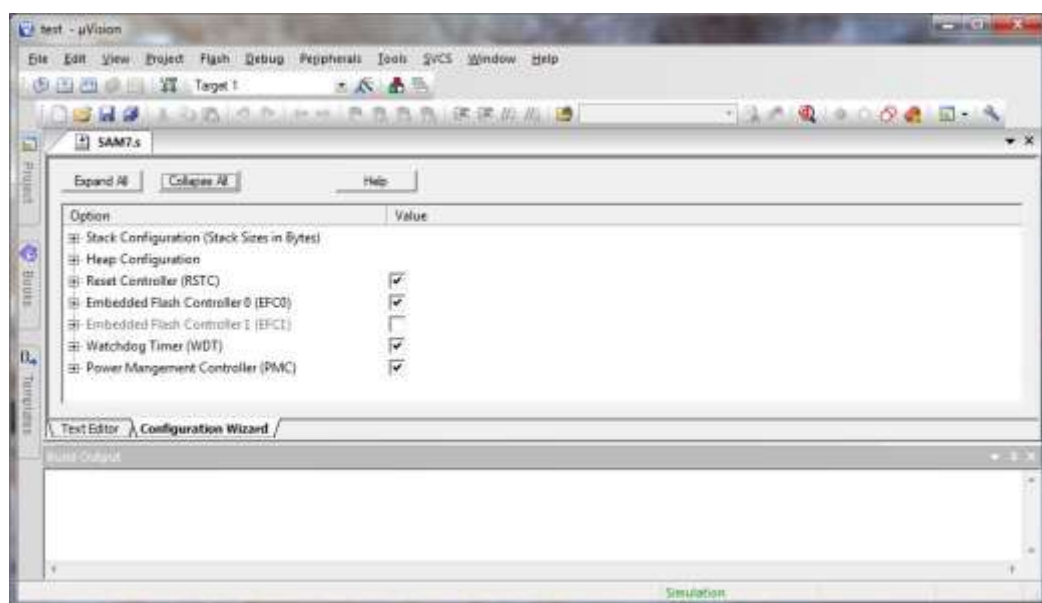
## اعمال تنظیمات CLKG در جادوگر KEIL:

در کامپایلر KEIL بخشی به نام Configuration Wizard وجود دارد که در آن میتوان تنظیمات مربوط به واحد CLKG و PMC را به صورت گرافیکی انجام داد. در ادامه به بررسی تنظیمات مربوط به واحد CLKG شرح داده شده است (تنظیمات واحد PMC در بخش بعدی و بعد از بررسی رجیستر های مربوطه بررسی خواهد شد).

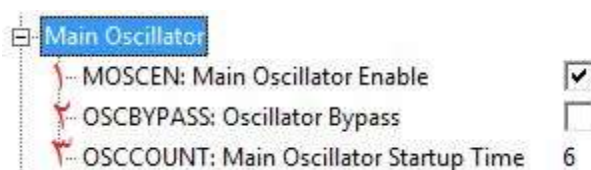
برای مشاهده پنجره Configuration Wizard در پالت Project گزینه SAM7.S را انتخاب کنید، در صورتی که پالت Project در محیط نرم افزار وجود ندارد، از منوی VIEW گزینه ی WINDOWS Project را انتخاب نمایید:



تصویر پنجره Configuration Wizard در زیر آورده شده است :



علامت تیکی که جلوی هر قسمت می بینید دلالت بر فعال بودن همان قسمت است . در این جا تنها به بررسی Power Management Controller (PMC) می پردازیم . ابتدا تنظیمات مربوط به Main Oscillator را توضیح می دهیم . به تصویر زیر نگاه کنید .



- شماره 1: در صورت تیک خوردن اسیلاتور اصلی ( Main ) فعال می شود .

- شماره 2 : در صورت تیک خوردن ، اسیلاتور اصلی بای پاس شده و کاربر باید برای اعمال کلاک به سیستم میکروکنترلر ، توسط پایه ی Xin اقدام کند .
- شماره 3 : عددی که ملاحظه می فرمایید زمان لازم برای راه اندازی اولیه ی اسیلاتور ( StartUp Main Oscillator ) می باشد که بستگی به شرایط کار میکروکنترلر از جمله فرکانس کاری و محیط بستگی دارد . برای اطلاعات بیشتر به قسمت DC Characteristic دیتا شیت مراجعه فرمایید .

**تکته :** اعداد نمایش داده شده در تصاویر ، مقادیر پیش فرض نرم افزار برای راه اندازی میکروکنترلر به صورت ساده می باشند ، ممکن است این اعداد با تنظیمات فعلی کامپایلر شما متفاوت باشند .

تصویر زیر ، تنظیمات مربوط به واحد PLL را نشان می دهد .

Phase Locked Loop (PLL)	
DIV: PLL Divider	32
MUL: PLL Multiplier	125
OUT: PLL Clock Frequency Range	80..160MHz
PLLCOUNT: PLL Lock Counter	28
USBDIV: USB Clock Divider	None

- شماره 1 : این قسمت همان فیلد DIV (ضریب تقسیم واحد PLL) می باشد .
- شماره 2 : پارامتر MUL یا عدد ضرب کننده در فرکانس ورودی
- شماره 3 : این قسمت رنج فرکانسی PLL را تعیین می کند . در صورتیکه PLL شما در حوزه ی فرکانس 80 الی 160 مگاهرتز را می کند مورد 80...160MHz و برای فرکانس های حدود 150 تا 220 مگاهرتز مورد 150...220MHz را انتخاب نمایید .

- شماره 4 : این عدد هم مربوط به زمان قفل شدن PLL است که همین مقدار مناسب می باشد
- شماره 5 : این فیلد هم مربوط به تقسیم فرکانس خروجی PLL برای تغذیه ی فرکانسی USB Device می باشد . در صورتیکه گزینه ی None انتخاب شود ، هیچ تقسیم فرکانسی صورت نمی گیرد .

و دو مورد آخر که در شکل زیر نشان داده شده اند نیز در ادامه بررسی خواهند شد :

CSS: Clock Source Selection	PLL Clock
PRES: Prescaler	None

چند نکته :

1- در دیتاشیت میکروکنترلر ، حداکثر فرکانس کاری میکروکنترلر برابر با 55 مگاهرتز قید شده است . اما واحد PLL توانایی تولید فرکانس هایی تا 200 مگاهرتز را دارد . عدد 200 نباید اشتباها به عنوان حداکثر فرکانس کاری میکروکنترلر لحاظ شود (البته با تحقیقاتی که اینجانب انجام داده ام روشن شد که فرکانس تا حدود 80 الی 100 مگاهرتز قابل اعمال به میکروکنترلر است).

2- اسیلاتور SLCK ( نوسان ساز فرکانس پایین ) کالیبره نشده است و خود شرکت اتمل برگه اطلاعاتی را در مورد کالیبره کردن این اسیلاتور ، برای مصارفی همچون RTT ، قرار داده است . فرکانس نامی این اسیلاتور برابر با 32.768 کیلوهرتز است اما ممکن است فرکانس آن در صورت کالیبره نبودن در بازه 22 تا 42 کیلوهرتز تغییر کند .

<http://www.atmel.com/Images/doc6148.pdf>

3- برای بدست آوردن پارامترهای ضرب و تقسیم ( DIV و MUL ) برای واحد PLL ، شرکت اتمل نرم افزاری را در آدرس زیر ایجاد کرده است :

[http://www.atmel.com/dyn/resources/prod\\_documents/AT91SAM\\_pll.htm](http://www.atmel.com/dyn/resources/prod_documents/AT91SAM_pll.htm)

صفحه ی بالا توسط خود شرکت اتمل تهیه شده است و نمایی به صورت زیر دارد .

**AT91 TOOLS: PLL Ratio Computing**

**. How Using This Tools ?**

1. Capture the input frequency which drives the PLL.
2. Capture the targeted PLL output frequency.
3. Optional: Filling in the right minimum input frequency last field or remaining blank. This field can be left blank, but, filling in with the correct information (ref. to the PLL electrical characteristics) can reduce the computing time.

add up now ...

**IMPORTANT:**  
The divider field is a 8-bit field.  
The Multiplier is a 21-bit field.  
The user has to be careful when he is going to use the results from this tool: The PLL has input and output frequency constraints. Look after the electrical characteristics to have more information about those constraint numbers.

Input Frequency (in MHz)	<input type="text"/>	
Targeted PLL output (in MHz)	<input type="text"/>	
Minimum PLL Input Frequency (in MHz)	<input type="text"/>	
Divider Value	<input type="text"/>	
Multiplier Value	<input type="text"/>	
Real Output Frequency (in MHz)	<input type="text"/>	Resulting Error (in %) <input type="text"/>
<input type="button" value="RESULT"/>		

در قسمت Input Frequency ، فرکانس ورودی PLL خود را وارد کنید و در قسمت Target PLL Output ، مقدار فرکانس خروجی مورد نظر را وارد کنید . قسمت بعد که Minimum PLL Input Frequency نام دارد ، مربوط به حداقل فرکانس کاری PLL می باشد و اگر این مسئله برای شما مهم نیست ( که در اکثر موارد هم قابل چشم پوشی است و معمولاً این فیلد یک سری محدودیت ایجاد می کند ) ، این قسمت را خالی بگذارید . بعد از فشار دادن دکمه ی Result مقادیر DIV و MUL به ترتیب در قسمت های Divider Value و Multiplier Value ظاهر می شود که مقدار Multiplier Value باید یک واحد کم شده و در جادوگر keil قرار داده شود .

4- در صورتی که قصد دارید تنظیمات PLL را به صورت دستی انجام دهید ( با مقدار دهی رجیستر ها ، در برنامه اصلی تنظیمات را انجام دهید ) ، باید بخش definitions (Power Mangement Controller (PMC) را از فایل SAM7.S حذف کنید . (خطوطی که با \_PMC شروع میشوند).

### واحد مدیریت توان یا PMC :

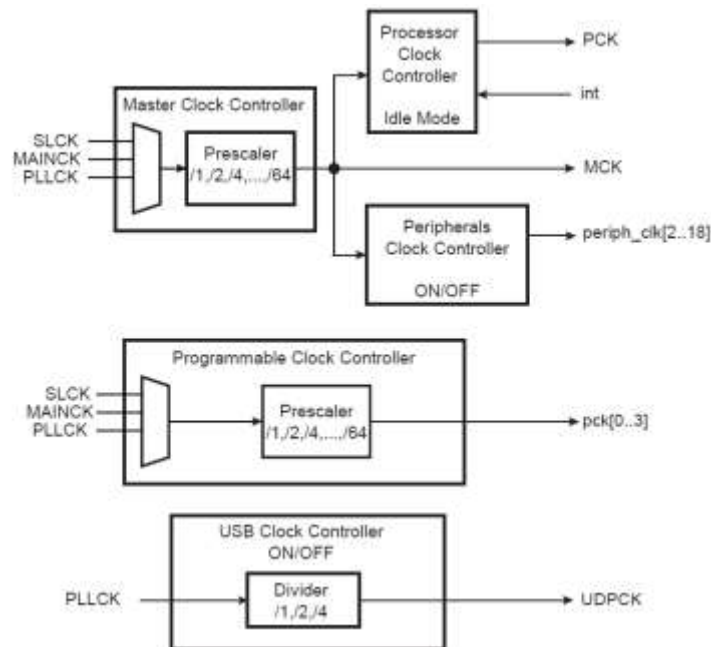
بعد از تولید کلاک توسط واحد CLKG نوبت به کنترل و عملیات های مختلف روی این کلاک می رسد . وظیفه ی اصلی واحد PMC بهینه سازی توان مصرفی کل سیستم توسط کنترل کلاک هر یک از بخش های جانبی میکروکنترلر می باشد . واحد PMC قابلیت فعال یا غیر فعال سازی کلاک بسیاری از قسمت های جانبی و همچنین کلاک CPU را دارد . این خاصیت به ما اجازه می دهد که در صورت نیاز کلاک هر قسمت جانبی میکروکنترلر که به آن نیاز نداریم را غیر فعال کنیم و در نهایت توان مصرفی کل سیستم را به حداقل کاهش دهیم .

واحد PMC کلاک های زیر را در خروجی خود بوجود می آورد :

- MCK : مخفف MASTER CLOCK ، کلاک اصلی سیستم است و می تواند فرکانسی از چند هرتز تا ماکسیمم رنج فرکانسی میکروکنترلر ، داشته باشد . کلاک های PCK (مخفف PROCESSOR CLOCK ، کلاک اختصاصی پردازنده ی مرکزی (CPU)) و PERIPH\_CLK (کلاک مورد نیاز برای لوازم جانبی نظیر مبدل آنالوگ به دیجیتال ، پروتکل های ارتباطی SPI و TWI و ... و کنترلر حافظه (MEMORY CONTROLLER و...) از این کلاک تامین میشوند .

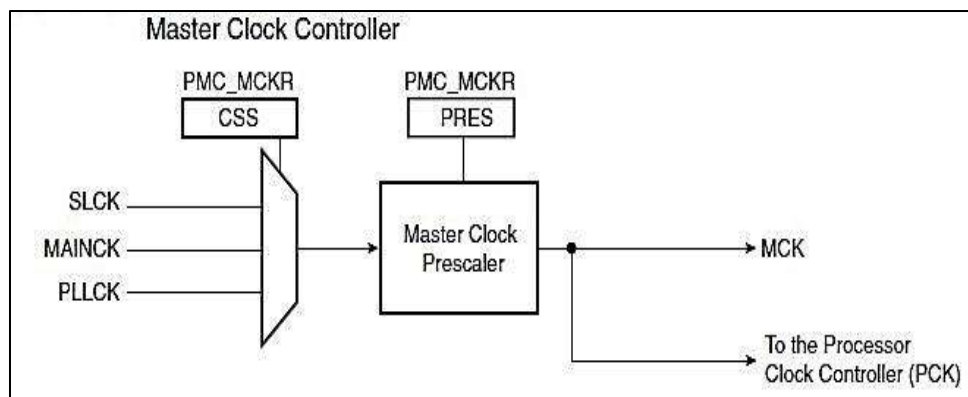
- UDPCCK : کلاک مورد نیاز برای واحد USB

- کلاک های قابل برنامه ریزی روی پایه های PCK1 تا 4 .



### کلاک MCK:

همانطور که گفته شد این کلاک مختص قسمت های جانبی میکروکنترلر ( Peripheral ) از جمله پورت های ورودی /خروجی ، مبدل آنالوگ به دیجیتال ، تایمر کانتر ( TC ) ، واحد کنترل کننده وقفه ( ACI ) ، واحد SPI ، کنترلر حافظه ( Memory Controller ) و ... می باشد . این کلاک بعد از گرفته شدن از یکی از کلاک های واحد CLKG ، بر یک ثابت ( در صورت نیاز ) تقسیم شده و به مقصد ارسال می شود . پس کنترل روی این کلاک شامل یک واحد تقسیم کننده و یک انتخاب کننده ی کلاک ، است که در شکل زیر نحوه ی ارتباط این قسمت ها را می بینید .



با مقدار دهی رجیستر PMC\_MCKR میتوان منبع تامین کننده کلاک MCK و عدد مقسم را تعیین کرد :

## رجیستر PMC\_MCKR (PMC Master Clock 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
-	-	-	PRES			CSS	

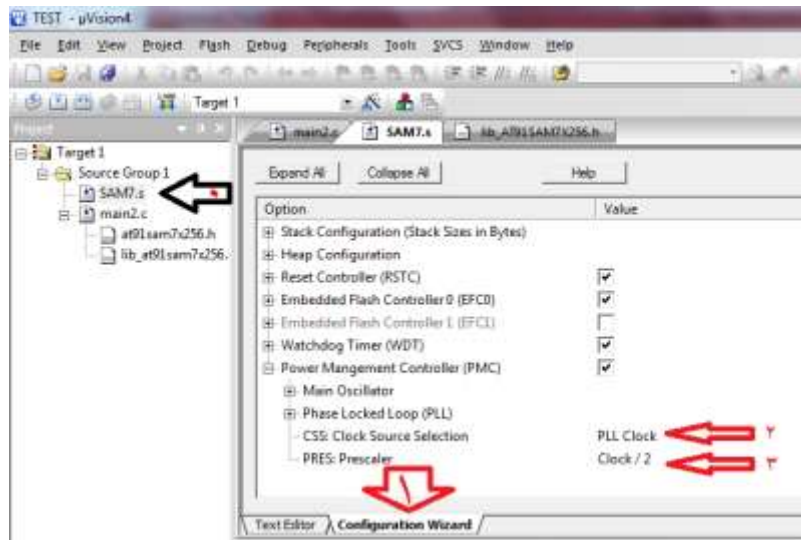
دو فیلد CSS و PRES در این ثابت وجود دارند. فیلد CSS جهت انتخاب منبع تامین کننده کلاک اصلی استفاده میشود. تصویر زیر این فیلد را به همراه کلاک انتخابی نسبت به مقدارش را نشان می دهد. همانطور که در تصویر صفحه قبل مشاهده میکنید، منبع تامین کننده کلاک MCK میتواند یکی از گزینه های SLCK (کلاک خروجی نوسانساز فرکانس پایین با فرکانس تقریبی 32.768 کیلو هرتز) و MAINCK (کلاک دریافتی از کریستال متصل شده به پایه های XIN و XOUT میکروکنترلر) یا PLLCK (فرکانس کریستال که در عدد (MUL + 1) / DIV ضرب شده است) باشد.

CSS		Clock Source Selection
0	0	Slow Clock is selected
0	1	Main Clock is selected
1	0	Reserved
1	1	PLL Clock is selected

پارامتر PRES مقدار مقسم کلاک این پایه را مطابق تصویر زیر تعیین میکند:

PRES			Processor Clock
0	0	0	Selected clock
0	0	1	Selected clock divided by 2
0	1	0	Selected clock divided by 4
0	1	1	Selected clock divided by 8
1	0	0	Selected clock divided by 16
1	0	1	Selected clock divided by 32
1	1	0	Selected clock divided by 64
1	1	1	Reserved

امکان مقدار دهی بیت های بالا از طریق بخش Configuration Wizard در کامپایلر KIEL وجود دارد.



- **شماره 2:** این فیلد تعیین کننده ی سورس کلاک کل سیستم می باشد. همانطور که گفته شد یکی از سه مورد کلاک SLCK، MAINCK و PLLCK می تواند به عنوان کلاک کل سیستم تعریف شود.
- **شماره 3:** اگر در خاطر داشته باشید، یک فیلد تقسیم کننده بعد از مالتی پلکسر انتخاب سورس کلاک سیستم وجود داشت که سورس کلاک را بر یک ثابت تقسیم و بعدا به خروجی واگذار می شد. این همان ضریب تقسیم کننده است. در صورتیکه گزینه ی None را انتخاب کنید هیچ تقسیم فرکانسی اتفاق نمی افتد و کلاک منبع انتخاب شده مستقیما به لوزام جانبی و CPU اعمال میشود.

#### کلاک ( Processor Clock ) PCK :

واحد PMC ویژگی مهم کنترل کلاک پروسسور ARM را برای بوجود آوردن وضعیت IDLE (مد کم مصرفی) میکروکنترلر دارد. برای غیرفعال کردن کلاک پردازنده ی مرکزی کافی است مقدار بیت شماره صفر رجیستر PMC\_SCDR یک شود تا پروسسور وارد حالت IDLE گردد. بعد از ورود به مد IDLE پردازنده توسط یک وقفه و یا ریست میکروکنترلر از حالت IDLE خارج می شود (کلاک پروسسور فعال می شود). دستور زیر در کامپایلر KEIL پروسسور را به وضعیت IDLE می برد:

```
*AT91C_PMC_SCDR = 0x0000001;
```

➤ نکته: برای وارد شدن به حالت Idle میکروکنترلر، پروسسور ARM دستور جاری را به اتمام و سپس به این وضعیت،

تغییر حالت می دهد.



رجیستر PMC\_SCDR (PMC System Clock Disable 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
—	—	—	—	PCK3	PCK2	PCK1	PCK0
7	6	5	4	3	2	1	0
UDP	—	—	—	—	—	—	PCK

### کلاک های قابل برنامه ریزی :

واحد PMC می تواند کلاک خود را روی چهار پایه از میکروکنترلر (PCKx) قرار دهد . به این ترتیب کنترل این کلاک های اختیاری در رجیستر PMC\_PCKx که x می تواند عدد 0 تا 4 باشد ، وجود دارد . پایه ی PCKx به طور مستقل می تواند یکی از کلاک های SLCK ، MAINCK و PLLCK ، به همراه یک ضریب تقسیم کننده ، برنامه ریزی شود . اما قبل از اعمال تنظیمات باید این پایه فعال شود . به همین ترتیب سراغ ثبات PMC\_SCER می رویم و برای غیر فعال کردن آن ها در ثبات PMC\_SCDR غیر فعال می کنیم . شکل های زیر ثبات PMC\_SCER را نشان می دهد .

رجیستر PMC\_SCER (PMC System Clock Enable 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
—	—	—	—	PCK3	PCK2	PCK1	PCK0
7	6	5	4	3	2	1	0
UDP	—	—	—	—	—	—	—

بیت UDP مربوط به این بخش نیست اما بیت های PCKx که در بالا نشان داده شده است ، در صورتیکه یک شوند ، کلاک میکروکنترلر روی پایه ی x قرار می گیرد . برای مثال با دستور زیر کلاک میکروکنترلر روی پایه ی PCK2 فعال می شود .

```
*AT91C_PMC_SCER = (1<<10);
```

همچنین در همین موقعیت بیتی و در رجیستر PMC\_SCDR ، قابلیت غیر فعال سازی این ویژگی وجود دارد . پس

```
*AT91C_PMC_SCDR = (1<<10);
```

موجب غیر فعال شدن کلاک روی پایه ی PCK2 می شود .

وضعیت فعال یا غیر فعال بودن کلاک CPU ، کلاک پورت USB و کلاک ایجاد شده بر روی پایه های PCK0-4 از طریق رجیستر PMC\_SCSR قابل مشاهده است :

رجیستر PMC\_SCSR (PMC System Clock Status 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
—	—	—	—	PCK3	PCK2	PCK1	PCK0
7	6	5	4	3	2	1	0
UDP	—	—	—	—	—	—	PCK

برای پایه های PCK0 تا PCK4 چهار رجیستر PMC\_PCK0 تا PMC\_PCK4 تعبیه شده است ، تنظیمات این ثبات ها مشابه به هم بوده و در ادامه بررسی شده است. به جای x شماره ی پایه ی مورد برنامه ریزی قرار داده می شود .

ثبات PMC\_PCKx (PMC Programmable Clock 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
—	—	—	—	PRES	—	CSS	—

دو فیلد CSS و PRES در این ثبات وجود دارند . فیلد CSS جهت انتخاب منبع تامین کننده کلاک پایه استفاده میشود. تصویر زیر این فیلد را به همراه کلاک انتخابی نسبت به مقدارش را نشان می دهد .

CSS		Clock Source Selection
0	0	Slow Clock is selected
0	1	Main Clock is selected
1	0	Reserved
1	1	PLL Clock is selected

و پارامتر PRES مقدار مقسم کلاک این پایه را مطابق تصویر زیر تعیین میکند :

PRES			Programmable Clock
0	0	0	Selected clock
0	0	1	Selected clock divided by 2
0	1	0	Selected clock divided by 4
0	1	1	Selected clock divided by 8
1	0	0	Selected clock divided by 16
1	0	1	Selected clock divided by 32
1	1	0	Selected clock divided by 64
1	1	1	Reserved

برای مثال ( بعد از فعال کردن کلاک روی پایه ی PCK2 در مثال قبل ) ، دستور زیر ثبات PMC\_PCK2 را برنامه ریزی می کند ، به طوری که کلاک MAINCK روی پایه ی PCK2 به 4 تقسیم شود :

AT91C\_PMC\_PCKR[2] = (0x010<<2) | 0x01;

در دستور بالا ملاحظه می فرمایید که این ثبات در کامپایلر KEIL به صورت آرایه ای تعریف شده است که عدد داخل [] نشان دهنده ی ثبات های PMC\_PCKx می باشد .

نکته ی مهم : بعد از انجام دادن کامل تنظیمات PCK ، اقدامات لازم برای فعال سازی خروجی همین پایه و اختصاص Peripheral های A و B در قسمت PIO میکروکنترلر انجام شود . (در ادامه توضیحات بیشتری در این رابطه وجود دارد )

### نحوه ی فعال یا غیر فعال سازی وسایل جانبی میکروکنترلر در PMC:

همانطور که در بخش های قبل گفته شد ، واحد PMC وظیفه ی کنترل کلاک هر قسمت جانبی میکروکنترلر ( به استثنا چند قسمت مانند ADC که آن ها هم به طور دایم از این واحد کلاک می گیرند ) ، جهت فعال یا غیر فعال سازی همان قسمت ، را دارد . نکته ای که باید توجه داشت این است که قبل از کار با هر قسمت جانبی میکروکنترلر ، باید کلاک همان قسمت در PMC فعال شود . در این جا چگونگی این کار را به همراه ثبات های کنترلی مربوطه توضیح خواهیم داد.

ثبات هایی که وظیفه ی فعال یا غیر فعال سازی کلاک قسمت های مختلف میکروکنترلر را دارند به PMC\_PCER ، PMC\_PCDR نامگذاری شده اند . ثبات سوم یعنی PMC\_PCSR ، وظیفه ی اطلاع رسانی وضعیت کلاک هر قسمت را دارد . در تصویر زیر بیت های موجود در این ثبات را مشاهده می کنید :

رجیستر PMC\_PCER ( PMC Peripheral Clock Enable Register ):

31	30	29	28	27	26	25	24
PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24
23	22	21	20	19	18	17	16
PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
15	14	13	12	11	10	9	8
PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
7	6	5	4	3	2	1	0
PID7	PID6	PID5	PID4	PID3	PID2	-	-

رجیستر PMC\_PCDR (PMC Peripheral Clock Disable Register):

31	30	29	28	27	26	25	24
PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24
23	22	21	20	19	18	17	16
PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
15	14	13	12	11	10	9	8
PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
7	6	5	4	3	2	1	0
PID7	PID6	PID5	PID4	PID3	PID2	-	-

هر بیتی که با PIDx نشان داده شده اند ، در صورت یک شدن ، Peripheral مربوط به خود را یک می کند . پس از این جا مشخص می شود که هر قسمت جانبی یک ID مخصوص به خود دارد که در صورت یک شدن در ثبات بالا ، کلاک قسمت موردنظر فعال می شود . برای مثال ID قسمت PIOA برابر 2 می باشد ، به این معنی که برای فعال شدن کلاک PIOA باید بیت PID2 یک شود . شکل زیر ID همه ی Peripheral ها را نشان می دهد .

Peripheral ID	Peripheral Mnemonic	Peripheral Name	External Interrupt
0	AIC	Advanced Interrupt Controller	FIQ
1	SYSC <sup>(1)</sup>	System Controller	
2	PIOA	Parallel I/O Controller A	
3	PIOB	Parallel I/O Controller B	
4	SPI0	Serial Peripheral Interface 0	
5	SPI1	Serial Peripheral Interface 1	
6	US0	USART 0	
7	US1	USART 1	
8	SSC	Synchronous Serial Controller	
9	TWI	Two-wire Interface	
10	PWMC	Pulse Width Modulation Controller	
11	UDP	USB Device Port	
12	TC0	Timer/Counter 0	
13	TC1	Timer/Counter 1	
14	TC2	Timer/Counter 2	
15	CAN	CAN Controller	
16	EMAC	Ethernet MAC	
17	ADC <sup>(1)</sup>	Analog-to Digital Converter	
18 - 29	Reserved		
30	AIC	Advanced Interrupt Controller	IRQ0
31	AIC	Advanced Interrupt Controller	IRQ1

به عنوان مثال در دستور زیر کلاک PIOA فعال و کلاک TC0 غیر فعال می شود .

```
*AT91C_PMC_PCER = (1<<2);
```

```
*AT91C_PMC_PCDR = (1<<12);
```

نکته : بعد از نوشتن یک در PMC\_PCDR ، کلاک قسمت موردنظر فوراً قطع می شود و بعد از ریست شدن میکروکنترلر ، کلاک همه ی قسمت های جانبی به طور پیش فرض غیر فعال می باشد .