سیاوش کاوسی

9231048

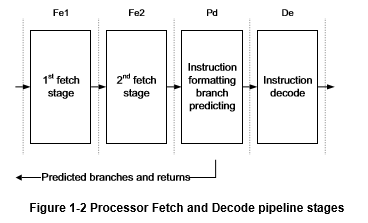
تمرین سوم درس ریزپردازنده

1-

**مراحل اجرا به صورت خط لوله (Execution Pipeline)**

فرآیند شامل مراحل اجرایی (اصلی) زیر است:

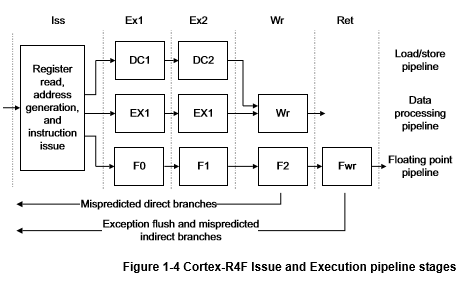
* مراحل واکشی دستور
* مراحل تشخیص (decode) دستورالعمل
* مرحله issue
* سه یا چهار مرحله اجرایی



Fe: دو مرحله واکشی دستورالعمل از حافظه اصلی

Pd: مرحله pre-decode (پیش تشخیص) که قالب دستورالعمل مشخص می شود و پیش بینی دستور های branch در این مرحله است

De: تشخیص دستورالعمل (Instruction decode)



**نام و توضیح کوتاه مراحل خط لوله:**

Iss: خواندن از ثبات و instruction issue (کار واحد instruction issue پر کردن خط لوله از دستورالعمل هاست در واقع جریان کنترلی (جریان برنامه) را پیش بینی میکند و دستورات را براساس آن واکشی می کند)

Ex: مراحل اجرایی

Wr: دوباره نوشتن(write-back) داده بعد از مراحل اجرایی

Ret: instruction retire ،چون ساختار به صورت خط لوله است توانایی اجرای دستورالعمل ها بیشتر از حدی است که جریان برنامه به آن نیاز دارد (speculative execution) در نتیجه دستورالعمل هایی که برنامه واقعا به آنها نیاز داشته باشد را retired instruction می گویند پس کار این بخش انتخاب این دستورات است

**نام و توضیح کوتاه مراحل بارگذاری/ذخیره خط لوله:**

DC1: اولین مرحله دسترسی به حافظه

DC2: دومین مرحله دسترسی به حافظه

**نام و توضیح کوتاه مراحل مربوط به ممیز شناور خط لوله:**

F0: خواندن ثبات ممیز شناور

F1: مرحله اول اجرای دستور ممیز شناور

F2: مرحله دوم اجرای دستور ممیز شناور

Fwr: بازنویسی داده ممیزشناور

ساختار خط لوله فوق امکان دسترسی با 2 سیکل به حافظه و 1 سیکل پنالتی بارگذاری-استفاده (در خط لوله اگر دستورالعمل فعلی به قبلی وابسته باشد اجرای دستور وابسته باید یک سیکل عقب بیفتد) را فراهم کرده است

2-

**تمرین 2 از فصل 8**

sbis: آدرس دهی مستقیم ثباتی (I/O)

in: آدرس دهی مستقیم I/O

sbrc: آدرس دهی مستقیم ثباتی

sbi: آدرس دهی مستقیم ثباتی

andi: آدرس دهی مستقیم ثباتی

spm: آدرس دهی حافظه برنامه با آدرس ثابت و پس افزایش

elpm: آدرس دهی حافظه برنامه با آدرس ثابت

std: آدرس دهی غیرمستقیم با جابجایی

brlt: آدرس دهی نسبی حافظه برنامه

cpi: آدرس مستقیم ثباتی

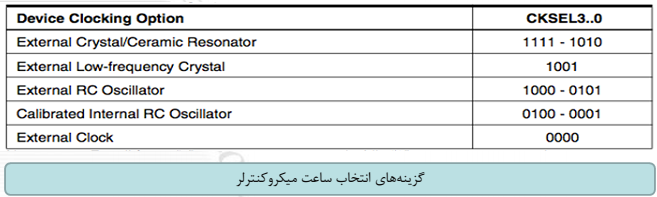
brbs: آدرس دهی نسبی حافظه برنامه

**تمرین 1 از فصل 9**

الف)

انتخاب منبع ساعت توسط فیوز بیت های که در جدول زير آمده انتخاب مي شود. سيگنال ساعت از منبع انتخاب شده به مولد ساعت ميکروکنترلر وارد شده و به ماژول هاي مناسب مسيردهي مي شود.

**توجه:** براي همه فيوزها 1 به معني برنامه ريزي نشده و 0 به معني برنامه ريزي شده ميباشد.

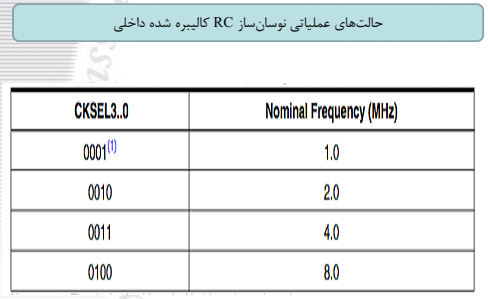


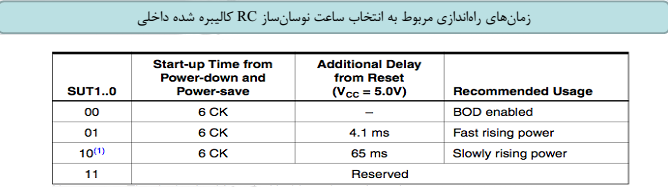
**مقدار پیش فرض فیوزبیت های میکروکنترلر**

**SUT** : start up time=”10”

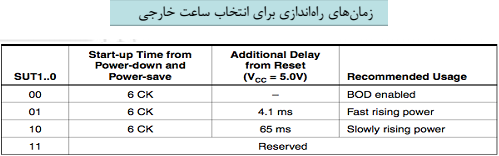
**CKSEL** : clock select=”0001”

**اسیلاتور RC داخلی کالیبره شده:**



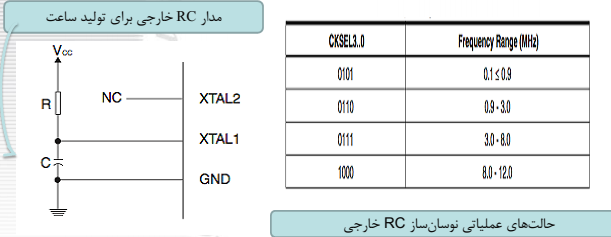


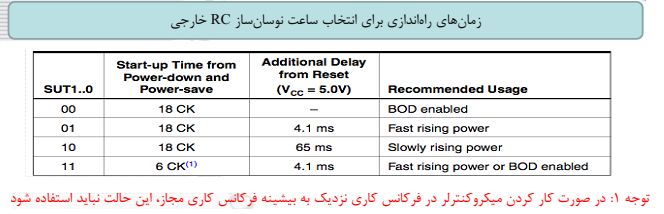
**کلاک خارجی :**



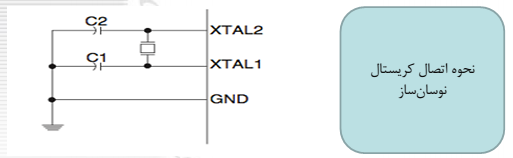
**اسیلاتور RC خارجی :**

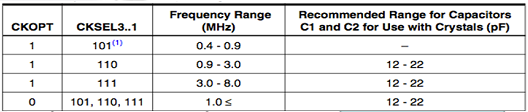
فرکانس بطور تقريبي توسط فرمول محاسبه مي شود. C بايد حداقل 22 پيکوفاراد باشد.  
با برنامه ريزي فيوز CKOPT، کاربر مي تواند يک خازن داخلي 36 پيکوفاراد را بين XTAL1 و GND فعال کند تا نيازي به خازن خارجي نباشد.

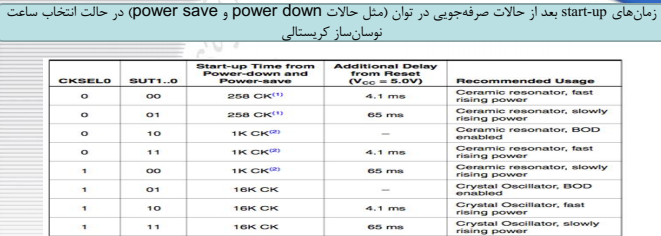


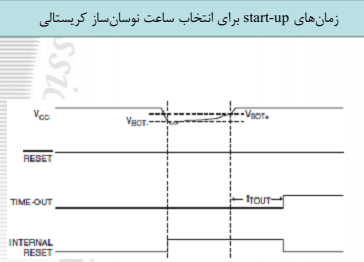


**اسیلاتور کریستالی:**



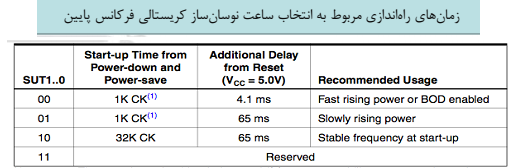




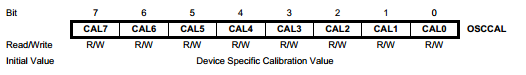


**اسیلاتور کریستالی فرکانس پائین :**

اگر اين نوسان ساز انتخاب شود، زمان مربوط به راه اندازي توسط فيوزهاي SUT مطابق جدول زير تعيين مي شود.



ب) در حالت استفاده از اسیلاتور RC داخلی کالیبره شده، از تنظیم ثبات OSCCAL برای کالیبره کردن ساعت RC داخلی تولید شده استفاده میکنیم:



نوشتن بایت کالیبراسیون در این آدرس باعث می­شود که اسیلاتور داخلی به گونه­ای تنظیم شود که تغییرات فرآیند از نوسان­ساز داخلی حذف شود. این کار به صورت خودکار در حین بازنشانی و شروع به کار مجدد تراشه انجام می­شود. وقتی که OSCCAL صفر شود، کمترین مقدار فرکانس ممکن انتخاب می­شود. نوشتن مقادیر غیر صفر در این ثبات، فرکانس نوسان­ساز داخلی را افزایش می­­دهد. نوشتن مقدار $FF در این ثبات بیشترین مقدار فرکانس ممکن را ایجاد می­کند.

ج) چون فیوز cksel3…0 برای تعیین فرکانس ساعت به کار می روند پس اگر این فیوزها تغییر نمایند میتوان نتیجه گرفت که فرکانس ساعت نیز تغییر نموده است.