|  |  |  |
| --- | --- | --- |
| دستورالعمل | مد آدرس دهی اُپرند اول  )مد آدرس دهی در صورت نداشتن اُپرند( | مد آدرس دهی اُپرند دوم  )در صورت وجود( |
| ADIW R1, K | R1: حالت آدرس دهی مستقیم ثبات(register file)  Add Immediate to Word  R1 + 1:R1 ← R1 + 1:R1 + K | K:immediate addressing  آدرس دهی مستقیم داده |
| EOR R0, R10 | R0: حالت آدرس دهی مستقیم ثبات (آدرس دهی دوثباتی)(register file)  Exclusive OR  R0 ← R0 ⊕ R10 | R10: آدرس دهی مستقیم با دوثباتی(register file) |
| RJMP 0xFF | حالت آدرس دهی نسبی حافظه برنامه  اجرای برنامه از آدرسPC+k+1 ادامه می یابد.  آدرس نسبی k بین -2048 تا 2048 تغییر می کند.  PC ← PC + k + 1 | \_ |
| IJMP | حالت آدرس دهی غیر مستقیم حافظه برنامه  Indirect Call to (Z)  اجرای برنامه از آدرس موجود در ثبات Z ادامه می یابد.(مقدار ثبات Z در PC قرار می گیرد.)  PC ← Z | \_ |
| EICALL | حالت آدرس دهی غیر مستقیم حافظه برنامه  Extended Indirect Call to (Z)  PC(15:0)← Z,  PC(21:16) ← EIND | \_ |
| CALL 0x2000 | حالت آدرس دهی مستقیم حافظه برنامه با آدرس ثابت  PC ← 0x2000 and STACK = PC + 1 | \_ |
| CPC R0, R2 | R0: حالت آدرس دهی مستقیم دو ثباتی (register file)  Compare with Carry  R0-R2-C | ثبات R2 : (register file) |
| BRTC 0x400 | حالت آدرس دهی نسبی برنامه تک ثباتی  Branch if T Flag Cleared  if (T = 0) then PC ← PC + 0x400 + 1 | \_ |
| Mov R0, R1 | حالت آدرس دهی مستقیم داده (آدرس دهی دو ثباتی)  ثبات اول R0 . مقدار دو ثبات R0 وR1 را با هم جا به جا می کند. | ثبات دوم R1( حالت آدرس دهی مستقیم داده) |
| LDI R0, 0x55 | R0: آدرس دهی مستقیم  Load Immediate  R0←0x55 | آدرس 0x55 immediate addressing |
| LDS R0, 0x100 | R0: حالت آدرس دهی مستقیم داده (ثبات مقصد)  آدرس داده 16 بیتی در 16 بیت کم ارزش یک دستورالعمل دو کلمه ای قرار دارد.  R0 ← 0x100 | 0x100: آدرس دهی مستقیم داده(محتوای خانه ی 0x100) |
| LDD R0, Y+0x15 | R0: حالت آدرس دهی مستقیم داده  آدرس عملوند نتیجه حاصل جمع محتوای ثبات Y با 6 بیت آدرس موجود در کلمه دستورالعمل می باشد.  R0 ← (Y + 0x15) | آدرس دهی غیر مستقیم با جابه جایی(محتوای خانه ی Y+0x15) |
| ST –X, R10 | حالت آدرس دهی غیر مستقیم داده با پیش کاهش  ثبات X قبل از عملیات، یک واحد کاهش می یابد.آدرس عملوند کاهش یافته محتوای ثبات X است.X←X-1 | R10: اپرند دوم همان آدرس منبعRr است که اینجا آدرس 10 امین خانه را برمی داریم.مستقیم |
| ELPM | R0: حالت آدرس دهی حافظه برنامه به صورت ضمنی  Extended Load Program Memory  R0 ← (RAMPZ:Z) | RAMPZ:Z: حالت آدرس دهی حافظه برنامه به صورت ضمنی |
| OUT SFIOR, R0 | حالت آدرس دهی مستقیم I/O  SFIOR 6 بیت از کلمه دستورالعمل را دارد.  I/O(SFIOR) ← R0 | R0 آدرس ثبات مبدا است.  آدرس دهی مستقیم |

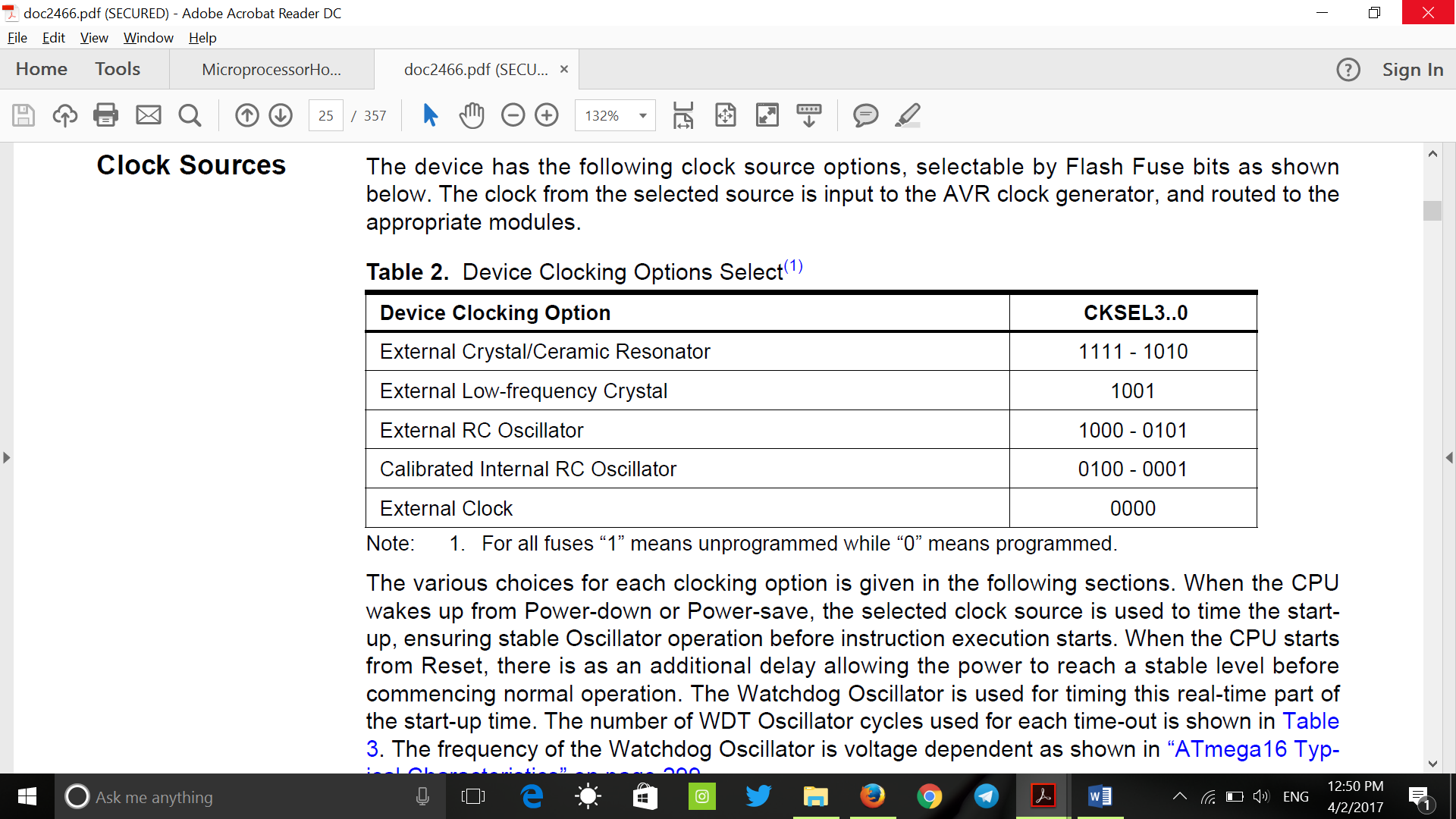
1. در هر یک از دستورات زیر از چه مد های آدرس دهی استفاده شده است؟

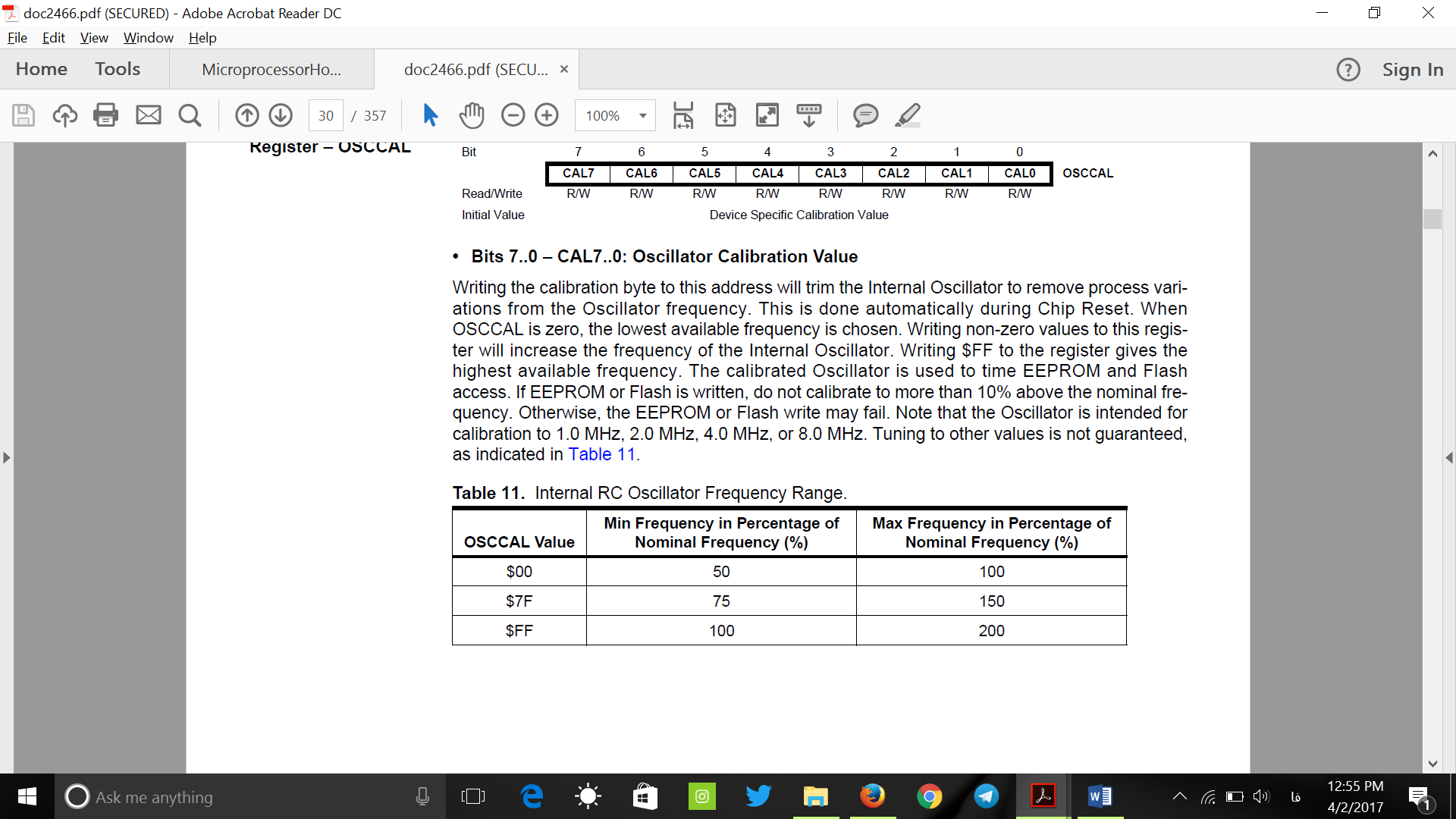
2. همان طور که می دانیم منابع مختلفی به عنوان سیگنال ساعت می تواند در میکروکنترلرهای AVR استفاده گردد. این منابع عبارتند از:

* اسیلاتور RC کالیبره شده
* ساعت خارجی
* اسیلاتور RC خارجی
* اسیلاتور کریستالی
* اسیلاتور کریستالی فرکانس پایین

الف) با استفاده از برگه داده های میکروکنترلر ATmega16، نحوه برنامه ریزی بیت های فیوز را برای تولید سیگنال ساعت به کمک کلیه روش های فوق مشخص نمایید.

چهار فیوز بیتی که در انتخاب روش تولید clock استفاده می شوند، CKSEL3-0 نام دارند. در جدول زیر نحوه برنامه ریزی این فیوز بیت ها برای هر یک از انواع روش تولید clock مشخص شده است:



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

ج) چه راهی برای بررسی تغییر فرکانس ساعت تولید شده در موارد فوق پیشنهاد می نمایید؟

می دانیم که فرکانس ساعت تولید شده در هر روش، با توجه به مقدار فیوزبیت های CKOPT و CKSEL و از روی جدول تعیین می شود، لذا با بررسی مقدار این فیوزبیت ها و مورد توجه قرار دادن روش انتخابی تولید CLOCK، می توان از تغییر فرکانس ساعت تولید شده اطلاع پیدا کرد.

3. بندهای ب، ج، د و ه از تمرین 2 از فصل 9 کتاب ریزپردازنده و زبان اسمبلی

Q3

4.برنامه ای بنویسید که ارقام 0 الی 9 را در حافظه EEPROM میکروکنترلر بنویسید.

Q4

5**.** یک نمایش دهنده 7-Segment را به پورت B میکروکنترلر ATMega16 متصل نمائید )مشابه شکل 17-9کتاب(. پایه های این پورت را در وضعیت خروجی قرار دهید. معادل 7-Segment ارقام 0 الی 9 را بدست آورده و توسط شبه کد DB در جدولی در حافظه برنامه در جدولی با آدرس شروع با برچسب BCDTo7-Seg قرار دهید.

Q5

6. برنامه ای بنویسید که ارقام 0 الی 9 نوشته شده در حافظه EEPROM میکروکنترلر را خوانده و هر کدام را با یک تاخیر

مناسب بر روی نمایش دهنده 7-Segment متصل به پورت B بنویسد. تاخیر را توسط یک زیرروال تاخیر ایجاد نمائید )یک

حلقه که در آن تعدادی دستورالعمل به دفعات تکرار شوند ایجاد یک تاخیر می نماید(.

7. وضعیت پرچم ها را پس از اجرای هر یک از دستورالعمل های برنامه زیر مشخص نمائید. فرض کنید کلیه پرچم ها پس از شروع برنامه 0 هستند.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | C | Z | N | V | S | H | T | I |
| LDI R0, 0x80 | \_ | \_ | \_ | \_ | \_ | \_ | \_ | \_ |
| BST R0, 7 | \_ | \_ | \_ | \_ | \_ | \_ | 1 | \_ |
| ADD R0, 0x7F | Rd7 • Rr7 + Rr7 • R7 + R7 • Rd7  ⇔  0 | R7 • R6 • R5 • R4 • R3 • R2 • R1 • R0  ⇔  0 | R7  0  ⇔  1 | Rd7 • Rr7 • R7 + Rd7 • Rr7 • R7  ⇔  0 | N ⊕ V  ⇔  1 | Rd3 • Rr3 + Rr3 • R3 + R3 • Rd3  ⇔  0 | \_ | \_ |
| INC R0 | \_ | ⇔  1 | ⇔  0 | ⇔  1 | ⇔  1 | \_ | \_ | \_ |
| SEI | \_ | \_ | \_ | \_ | \_ | \_ | \_ | 1 |

8. در یک زیر روال، یک بایت داده را از ثبات I/O به آدرس 0x25 دریافت، آنرا به ثبات R0 منتقل، نیبل های آنرا جابجا، بیت شماره 3 آنرا 0و بیت پنجم آن را تست کنید و پیرو آن اقدامات زیر را انجام دهید :

الف) اگر نتیجه تست بیت - پنجم 1 بود، مقدار نهایی R0 را در آدرس 0x10 نسبت به مقدار فعلی ثبات Z در حافظه داده

ذخیره نمائید Z=0x80))

ب) اگر نتیجه تست بیت - پنجم 0 بود، محتوای R0 را پس از یک شیف حسابی به راست، در عدد 5 ضرب و نتیجه را در دو

بایت متوالی در پشته ذخیره کنید.

ج) چنانچه بخواهیم مقدار - R0 حاصل از بند الف و دو بایت حاصل از بند ب در آدرس های متوالی در حافظه ذخیره شوند،

مقدار اولیه و نهایی SP را پیدا کنید.

Q8

9. برنامه ای به زبان اسمبلی ATMega16 بنویسید که حاصل جمع مربعات اعداد ضریب 5 در فاصله 1 تا n را محاسبه و در ثبات های R1:R0 قرار دهد. فرض کنید مقدار n پیش از اجرای برنامه در ثبات R10 قرار داشته است.

Q9

10. برنامه ای به زبان اسمبلی ATMega16 بنویسید که 100 عدد که در آدرس ARRAY در حافظه برنامه قرار گرفته اند را به صورت نزولی مرتب کند )فرض کنید این حافظه از پیش تعریف و مقدار دهی شده است(.

Q10