ساختار داخلی میکروکنترلرهای AVR

دکتر حسین غلامعلی نژاد

زمستان ۱۴۰۱

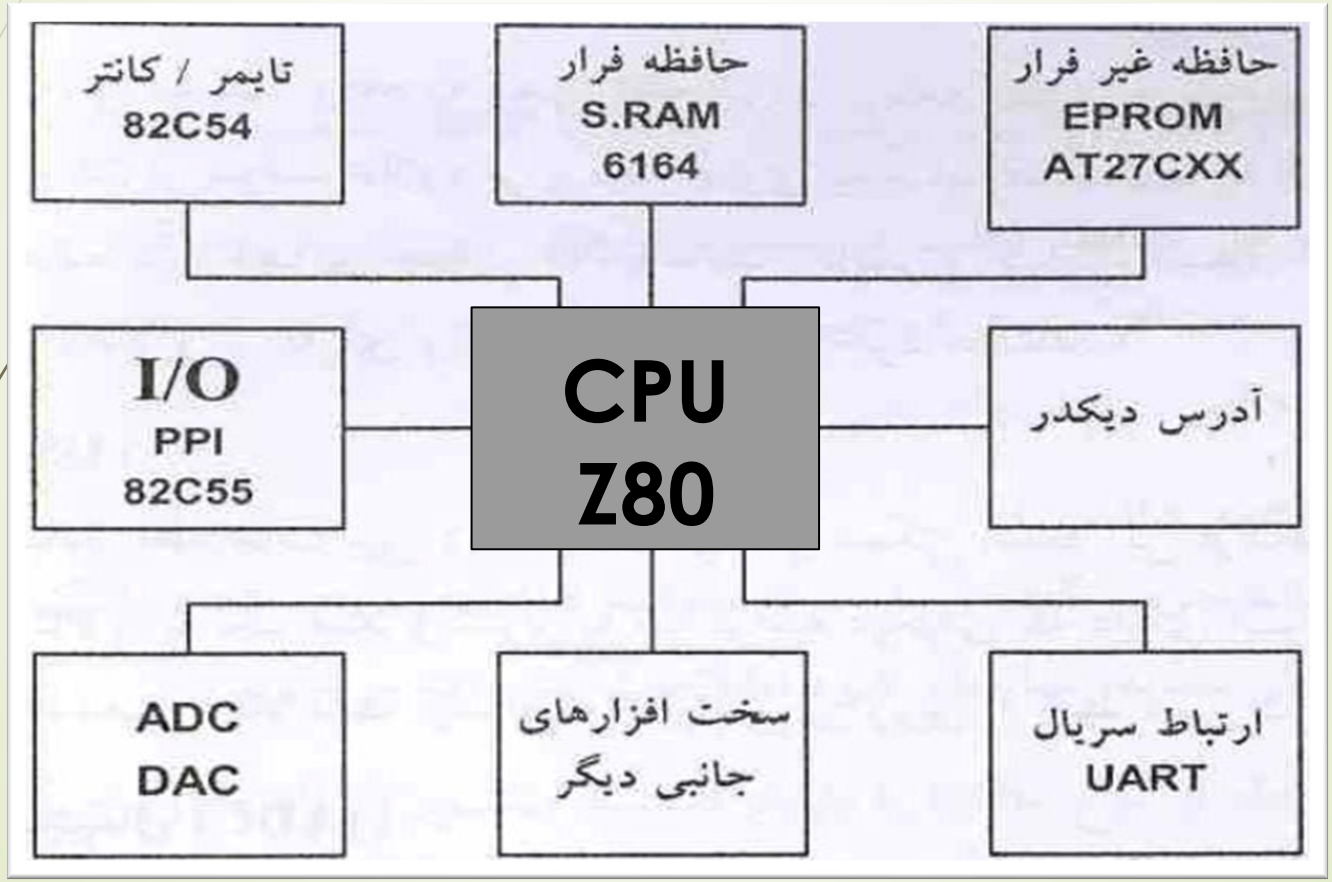
مقدمه ای بر میکروکنترلرها

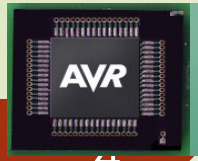
با پیشرفت علم و تکنولوژی در الکترونیک تراشه هایی به عنوان میکروپروسسورها طراحی و تولید شدند تا کاربر به هر صورت ممکن که می خواهد سیستم مورد نظر خود را طراحی کند به همین دلیل شرکت ZILOG میکروپروسسور Z80 را به بازار عرضه کرد. اما مشکل ریزپردازده Z80 این بود که نیاز به سخت افزار های جانبی داشت .

این عمل سبب افزایش قیمت و پیچیده شدن سخت افزار برای یک پروژه می باشد.

مقدمه ای بر میکروکنترلرها

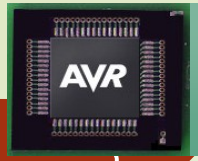
شکل زیر یک بلوک دیاگرام ریزپردازنده Z80 را به همراه امکانات جانبی نشان می دهد.





مقدمه ای بر میکروکنترلرها

- با توجه به مشکلات Z80 شرکت های سازنده قطعات به فکر ریزپردازنده ای شدند تا تمام امکانات جانبی را به همراه خود CPU داشته باشد تا شخص طراح راحت تر سیستم مورد نظر خود را پیاده سازی کند.
- در سال ۱۹۸۱ شرکت Intel تراشه ای را با عنوان میکروکنترلر خانواده ۸۰۵۱ به بازار عرضه کرد که دارای CPU ۸ بیتی ،تایمر ،وقفه،تبادل سریال ،حافظه SRAM و حافظه غیر فرار (FLASH)داخلی بود.
- شرکت های سازنده دیگری ،تحت لیسانس شرکت Intelاز میکروکنترلر ۸۰۵۱ تولید کردند که از جمله این شرکتها می توان به شرکت Atmel اشاره کرد.



مقدمه ای بر میکروکنترلرها

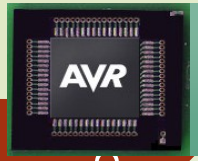
بعدها شرکت Atmel در سال 1997 میکروکنترلرهای جدیدی را با نام خانواده AVR به بازار عرضه نمود. که به سه سری تقسیم می شوند.

۱- سری AT90S

۲- سری Attiny

۳-سریATmega

میکروکنترلرهای AVR قابلیت های بیشتری نسبت به میکرکنترلر خانواده 8051 دارند و از نسل جدیدی از حافظه های FLASH و معماری RISC که در ادامه به آنها اشاره خواهیم کرد بهره می برند.



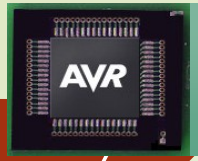
مقدمه ای بر میکروکنترلرها

۱- سری AT90S :

این سری اعضای کلاسیک خانواده AVR را تشکیل می دهند و قابلیتهای کمتری نسبت به دو سری بعدی دارند و کمتر استفاده می شوند.

جدول زیر میکرکنترلرهای سری AT90S را نشان می دهد.

AT90S4434	AT90S8535	AT90S4433	AT90S2323	AT90S1200
AT90S8534	AT90S4414	AT90S8515	AT90S2343	AT90S2313



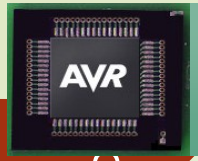
مقدمه ای بر میکروکنترلرها

۲- سری ATtiny :

این میکروکنترلرها در ابعاد کوچک ۸، ۲۰ و ۲۸ پایه هستند و قابلیت های خوبی نسبت به سری اول دارند و بیشتر در سیستم هایی که نیاز به پورت بالا دارند استفاده می شوند

جدول زیر میکرکنترلرهای سری ATtiny را نشان می دهد.

ATtiny10	ATtiny12	ATtiny15	ATtiny25	ATtiny28	ATtiny85
ATtiny11	ATtiny13	ATtiny22	ATtiny26	ATtiny45	ATtiny2313

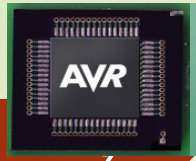


۳- سری ATmega:

این سری از میکروکنترلرهای AVR ،امکانات بیشتری نسبت به دو سری قبلی را دارندو توجه مخاطبان را به خود جلب کرده اند که در این کتاب ما روی میکروکنترلر ATmega16 بحث می کنیم.

جدول زیر میکرکنترلرهای سری ATmega را نشان می دهد.

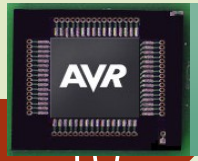
ATmega48	ATmega603	ATmega165	ATmega324
ATmega8	ATmega649	ATmega168	ATmega325
ATmega88	ATmega6490	ATmega169	ATmega329
ATmega64	ATmega16	ATmega128	ATmega3290
ATmega640	ATmega161	ATmega1280	ATmega3250
ATmega644	ATmega162	ATmega1281	ATmega256
ATmega6450	ATmega163	ATmega32	ATmega8535
ATmega670	ATmega164	ATmega323	ATmega8515



معماری میکروکنترلرهای AVR

به طور کلی دو نوع معماری برای ساخت میکروکنترلرها وجود دارد:
۱- معماری CISC: در این معماری تعداد دستورات بیشتر و پیچیده تر می باشد و سرعت اجرایی آن پاینتر است.
۲- معماری RISC: در این معماری تعداد دستورات کمتر و سرعت اجرایی آنها ۱۰ برابر نسبت به معماری cisc می باشد.

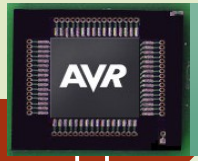
تفاوت معماری RISC با معماری CISC در تعداد و اندازه و همچنین اجرای دستورات،انتقال داده از ثباتها و حافظه ها به یکدیگر ،تعداد ثباتها ،برنامه نویسی دستورات و مصرف توان می باشد که این برتری های RISCبر CISC باعث شده تا میکروکنترلر های AVR از معماری RISCاستفاده کنند.



خصوصیات میکروکنترلر ATmega16

برخی از خصوصیات ATmega16 به شرح زیر میباشد.

- قابلیت اجرایی بالا و توان مصرفی کم
- اجرای ۱۳۱ دستورالعمل در یک کلاک سیکل وسرعتی تا ۱۶ مگاهرتز
- دارای ۳۲ رجیستر ۸ بیتی همه منظوره
- تحمل حافظه flashوEEPROM با قابلیت۱۰۰۰۰ تا ۱۰۰۰۰۰ بار نوشتن و پاک کردن و قفل برنامه جهت حفاظت از برنامه نوشته شده
- ۳۲ خط ورودی و خروجی قابل برنامه ریزی
- دارای منابع وقفه خارجی و داخلی و همچنین دارای ۶ مد Sleep
- و همچنین دارای چندین ویژگی جانبی و دیگر



فیوزبیت های میکروکنترلر ATmega16

فیوز بیت ها بخشی از حافظه FLASHهستند که با برنامه ریزی آن ها یک سری امکانات خاص در اختیار کاربر قرار می گیرد. در AVR ها حداکثر ۳ بایت برای فیوز بیت ها در نظر گرفته شده است. که به ترتیب زیر هستند:

1. بایت بالای فیوزبیت
2. بایت پایین فیوزبیت
3. فیوزبیت های توسعه یافته

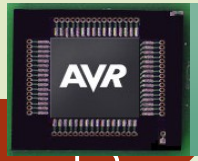
دقت شود که فیوز بیت ها با erase کردن یا پاک کردن حافظه FLASH از بین نمی روند. تعداد و نام فیوز بیت ها در سری های مختلف AVRتقریبا با هم یکسان است.(البته با کمی تغییرات جزئی)

نکته ی مهم: در فیوز بیت ها "۰" به معنی برنامه ریزی شدن و "۱" به معنی برنامه ریزی نشدن فیوز بیت است.

فیوزبیت های میکروکنترلر ATmega16

میکروکنترلر ATmega16دارای ۲ بایت فیوز طبق جداول زیر می باشد که در ادامه بر روی آنها بحث خواهیم کرد.

شماره بیت	۰	۱	۲	۳	۴	۵	۶	۷
فیوز بیت های بالا	BOOTRST	BOOTSZ0	BOOTSZ1	EESAVE	CKOPT	SPIEN	JTAGEN	OCDEN
مقدار پیش فرض	1	0	0	1	1	0	0	1
شماره بیت	۰	۱	۲	۳	۴	۵	۶	۷
فیوز بیت های پایین	CKSEL0	CKSEL1	CKSEL2	CKSEL3	SUT0	SUT1	BODEN	BODLEVEL
مقدار پیش فرض	1	0	0	0	0	1	1	1

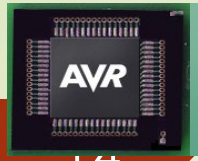


عملکرد فیوزبیتها

- **فیوزبیت OCDEN (on chip debug enable):** با فعال کردن فیوز JTAG می توان برنامه میکروکنترلر را به طور آنلاین (درحین اجرا)توسط نرم افزار AVR studio مشاهده کنیم.
- **فیوزبیت JTAGEN:** با فعال کردن این فیوز بیت می توان میکروکنترلر را از طریق ارتباط دهی استاندارد JTAG برنامه ریزی کرد.

  این بیت به صورت پیش فرض صفر بوده که باعث فعال بودن JTAGمی شود. همین امر باعث می شود بعضی بیت های پورت Cدر ATmega32کار نکند. برای عملکرد عادی پورت Cباید این بیت را غیر فعال کنید.

  JTAG یک رابط برای تست و برنامه ریزی ICهای دیجیتال که طبق استاندارد IEEE 1149.1مورد استفاده قرار می گیرد.
- **فیوز بیت SPIEN:** با فعال شدن این فیوز بیت می توان میکروکنترلر را از طریق ارتباط دهی SPI (ارتباط دهی سری) برنامه ریزی کرد.



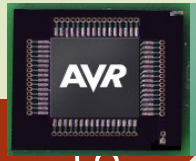
عملکرد فیوزبیتها

- **فیوزبیت CKOPT:** با فعال کردن این فیوز بیت می توان از حداکثر دامنه نوسان اسیلاتور خارجی استفاده کرد این حالت در مکانهایی که نویز زیادی دارند موثر است.
- **فیوزبیت EESAVE:**با پاک کردن میکروکنترلر EEPROMنیز پاک می شود در موقع پاک شدن حافظه flash اگر بخواهیم از محتوای حافظه EEPROM جلوگیری کنیم باید این فیوز بیت را فعال کنیم.

عملکرد فیوزبیتها

- **فیوزبیت BOOTRST:** این فیوز بیت برای انتخاب بردار است.اگر غیر فعال باشد بردار reset از آدرس 0X0000 حافظه خواهد بود اما اگر فعال باشد به ابتدای آدرسی که فیوز بیت های BOOTSZ1 وBOOTSZ0 طبق جدول زیر تعیین شده اند پرش می کند.(فیوز بیت های BOOTSZ1 وBOOTSZ0 میزان حافظه اختصاص داده شده BOOT را تعیین میکند)

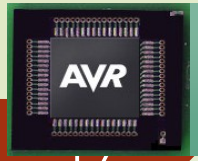
BOOTSZ1	BOOTSZ0	سایز بوت	صفحه	بخش حافظه کاربردی	بخش حافظه بوت	پایان بخش کاربردی	آدرس ریست بوت
1	1	۲۵۶ کلمه	4	$0000 - $3EFF	$3F00 - $3FFF	$3EFF	$3F00
1	0	۵۱۲ کلمه	8	$0000 - $3DFF	$3E00 - $3FFF	$3DFF	$3E00
0	1	۱۰۲۴ کلمه	16	0000 - $3BFF	$3C00 - $3FFF	$3BFF	$3C00
0	0	۲۰۴۸ کلمه	32	$0000 - $37FF	$3800 - $3FFF	$37FF	$3800



عملکرد فیوزبیتها

- **فیوزبیت BODEN:** برای فعال کردن مدار Brown-out باید این فیوز بیت فعال باشد.
  مدار Brown-out آشکار ساز ولتاژ تغذیه است که اگر از 2.7 یا 4 ولت کمتر شود میکروکنترلر را reset می کند.

- **فیوزبیت BODLEVEL:** اگر فیوز بیت BODENفعال شده باشد و فیوز بیت BODLEVEL برنامه ریزی نشده باشد آنگاه با کاهش ولتاژ VCC کمتر از 2.7v میکروکنترلر resetمی شود و اگر فیوز بیتBODLEVEL فعال شود آنگاه با کاهش ولتاژ VCCکمتر از 4v میکروکنترلر resetمی شود.



عملکرد فیوزبیتها

فیوزبیت های SUT0و SUT1: این دو فیوز بیت زمان شروع (start-up) را در موقع وصل تغذیه طبق جدول صفحه بعد تعیین می کنند.

این زمان فاصله بین اتصال تغذیه تا شروع به کار میکروکنترلر است و برای پایدار شدن خروجی نوسان ساز لازم است، چرا که ممکن است نوسان ساز در زمان اتصال تغذیه دقیقا همان فرکانس مطلوب را تولید نکند وعملا با تنظیم یک زمان طولانی تر فرصت پایدار شدن به نوسان ساز داده می شود. برای کاربردهای معمولی توصیه می شود با مقادیر پیش فرض کار کنید و این فیوز بیت ها را تغییر ندهید.

CKSEL0	SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay From Reset (Vcc=5.0V)	Recommended Usage
0	00	258 CK(1)	4.1 ms	Ceramic resonator, fast rising power
0	01	258 CK(1)	65 ms	Ceramic resonator, slowly rising power
0	10	1K CK(2)	-	Ceramic resonator, BOD rising power
0	11	1K CK(2)	4.1 ms	Ceramic resonator, fast rising power
1	00	1K CK(2)	65 ms	Ceramic resonator, slowly rising power
1	01	16K CK	-	Ceramic resonator, BOD rising power
1	10	16K CK	4.1 ms	Ceramic resonator, fast rising power
1	11	16K CK	65 ms	Ceramic resonator, slowly rising power

عملکرد فیوزبیتها

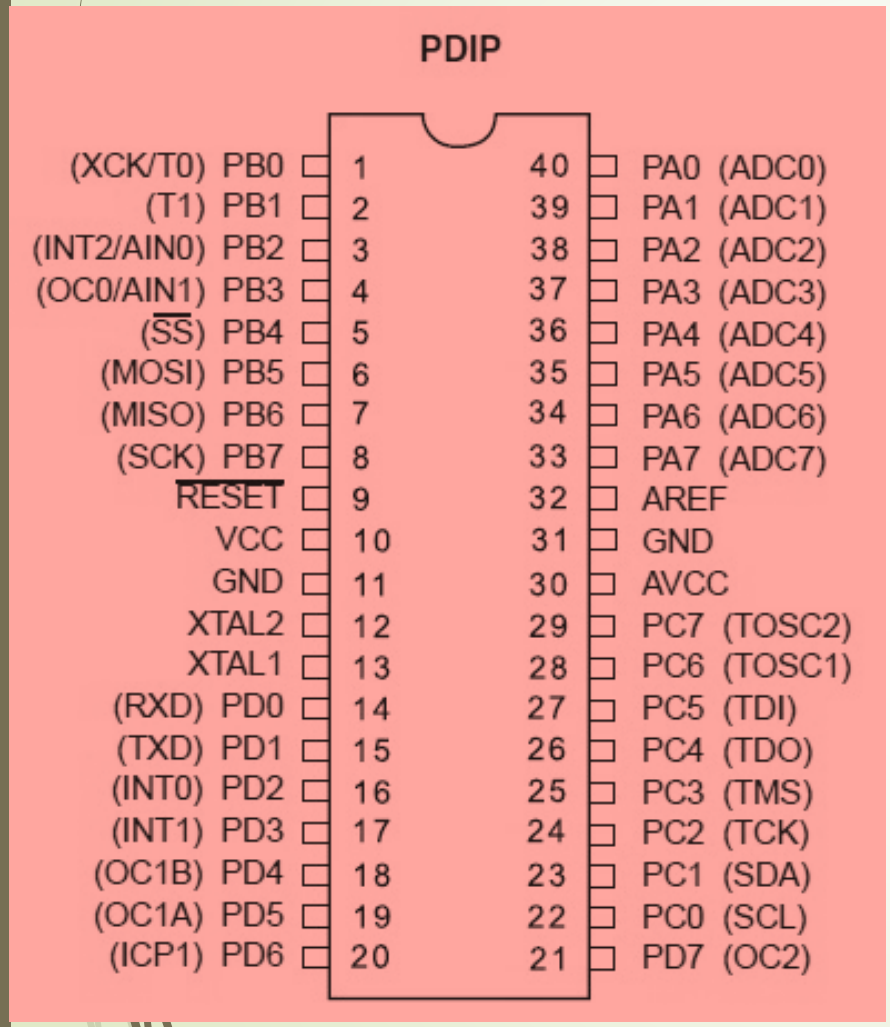
- **فیوزبیت های CKSEL3,CKSEL2,CKSEL1,CKSEL0:** توسط این فیوز بیت ها نوع و مقدار فرکانس اسیلاتور را تعیین می کنیم.

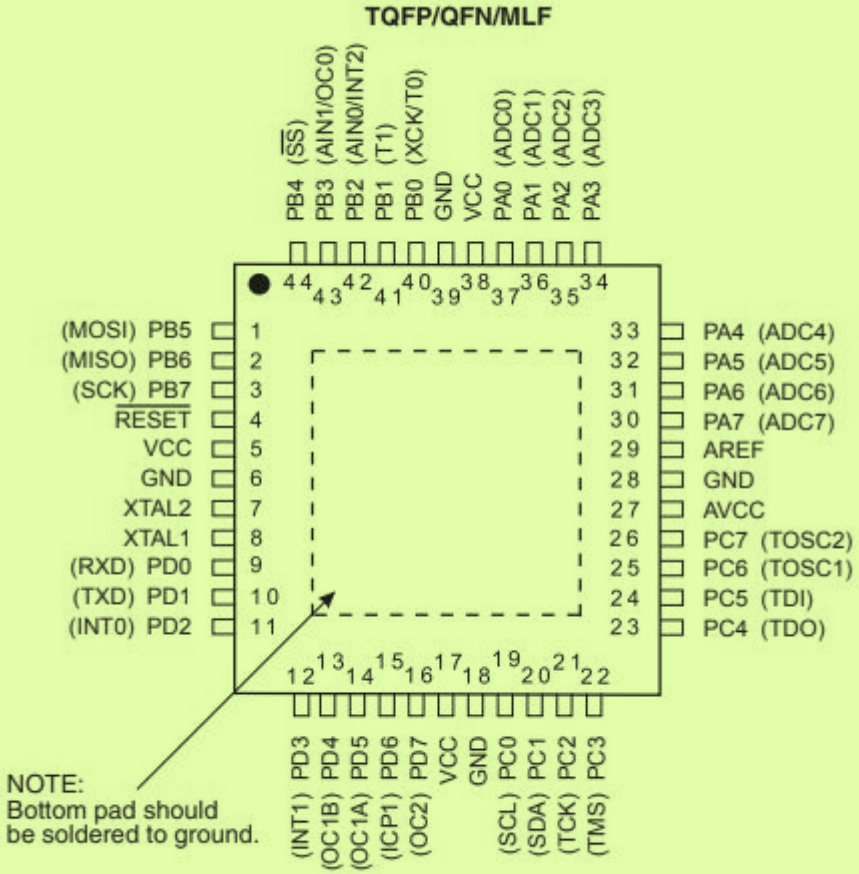
آپشن های کلاک میکروکنترلر	CKSEL3...0
کریستال خارجی/ رزوناتور سرامیکی	1111 - 1010
کریستال فرکانس پایین خارجی	1001
نوسان ساز RC خارجی	1000 - 0101
نوسان ساز RC داخلی کالیبره شده	0100 - 0001
کلاک پالس خارجی	0000

فرکانس نامی (MHz)	CKSEL3...0
1.0	0001
2.0	0010
4.0	0011
8.0	0100

میکروکنترلر AVRدارای ۴ حالت نوسان ساز RCداخلی است و مقدار پیش فرض آن فرکانس ۱ MHzاست.

پایه های ATmega16 با ترکیب PDIPو QFP/MLF





پورتهای ورودی و خروجی میکروکنترلر ATmega16

به طور کلی تمام پورتهای میکروکنترلرهای AVRدارای سه رجیستر تنظیم کننده می باشد.

1. **DDRx.n** :این رجیستر (data direction register)برای تنظیم هر پایه از پورت به عنوان ورودی و خروجی در نظر گرفته شده است اگر بیتی از این رجیستر **یک** شود نشان دهنده تعیین آن پایه به عنوان خروجی و اگر **صفر** شود آن پایه ورودی خواهد بود.

مثال:

```
DDRA=0xFF;          //تمام پایه های پورت A به عنوان خروجی
DDRA=0x00;          // تمام پایه های پورت A به عنوان ورودی
DDRC.0=1;           // پایه PC.0از پورت Cبه عنوان خروجی
DDRC.0=0;           //پایه PC.0از پورت Cبه عنوان ورودی
```

پورتهای ورودی و خروجی میکروکنترلر ATmega16

1. **PORTx.n** :این رجیستر(port data register)برای ارسال دیتا به خروجی می باشد هر موقع میکروکنترلر بخواهد داده ای را به خروجی بفرستد باید ابتدا رجیستر DDRx.n در حالت خروجی تنظیم شده باشد و سپس داده مورد نظر در رجیستر PORTx.n قرار می گیرد.

مثال:

```
DDRB=0xFF;          //تمام پایه های پورت Bبه عنوان خروجی
PORTB=46;           // عدد ۴۶ دسیمال به خروجی پورت B ارسال می گردد
```

پورتهای ورودی و خروجی میکروکنترلر ATmega16

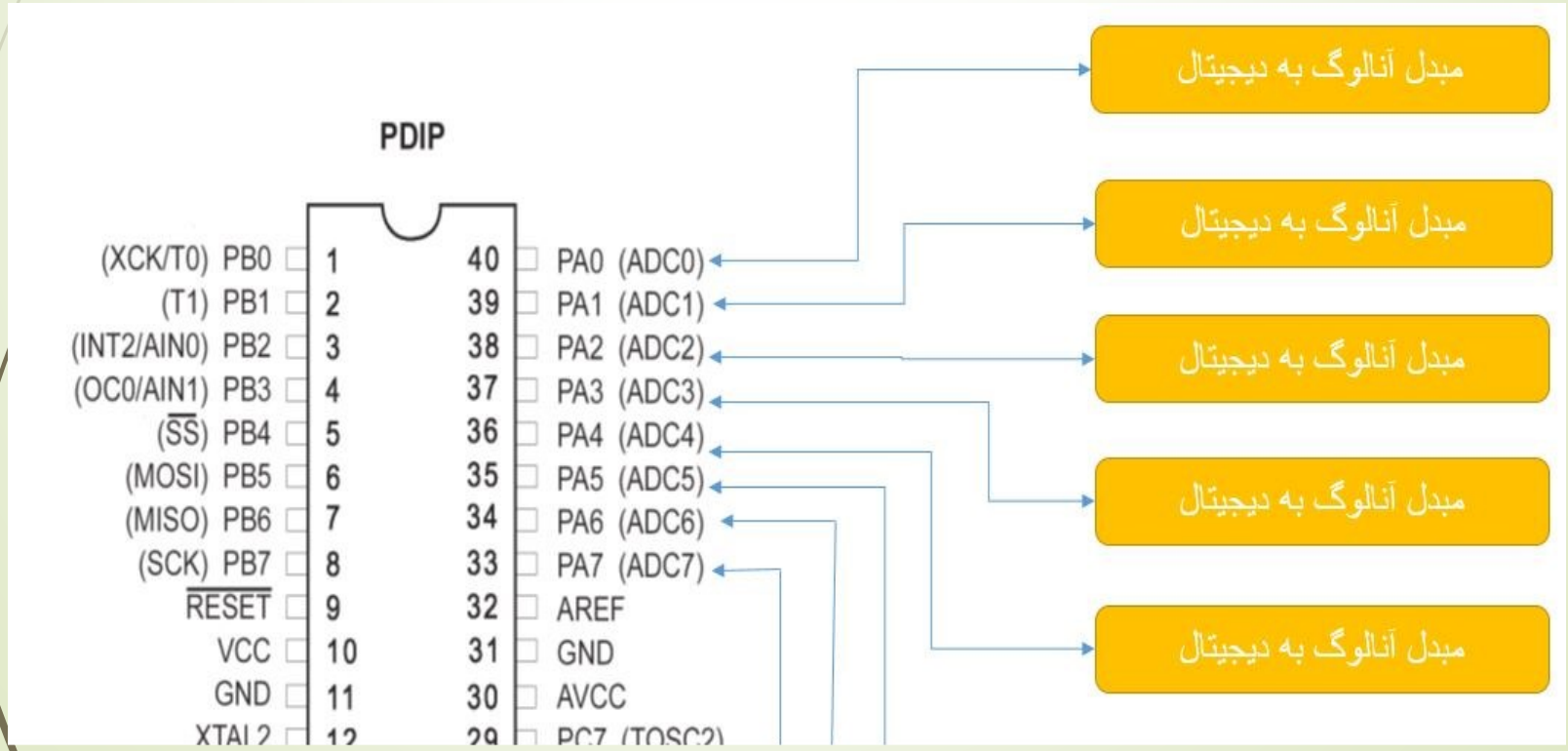
1. **PINx.n** :این رجیستر(port input pin address)برای دریافت داده از ورودی است . هرگاه میکروکنترلر بخواهد داده ای را از ورودی بخواند باید ابتدا رجیستر DDRx.n در حالت ورودی تنظیم شده باشد و سپس داده مورد نظر از رجیستر PINx.n به صورت بیتی توسط دستورهای شرطی خوانده شود همچنین خواندن به صورت بایتی ،با یک متغییر ۸ بیتی انجام می شود.

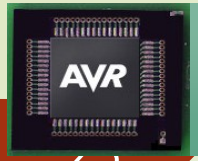
مثال:

```
// فعال کردن مقاومت pull-up داخلی پایه PC5
PORTC.5=1;
// تعیین پایه PC5 به عنوان ورودی
DDRC.5=0;
if(PIN.5==0) { //اگر پایه PC5برابر صفر باشد دستورالعمل ها اجرا شوند
        دستورالعملها;
}
```

کاربرد سایر پورتها در ATmega16

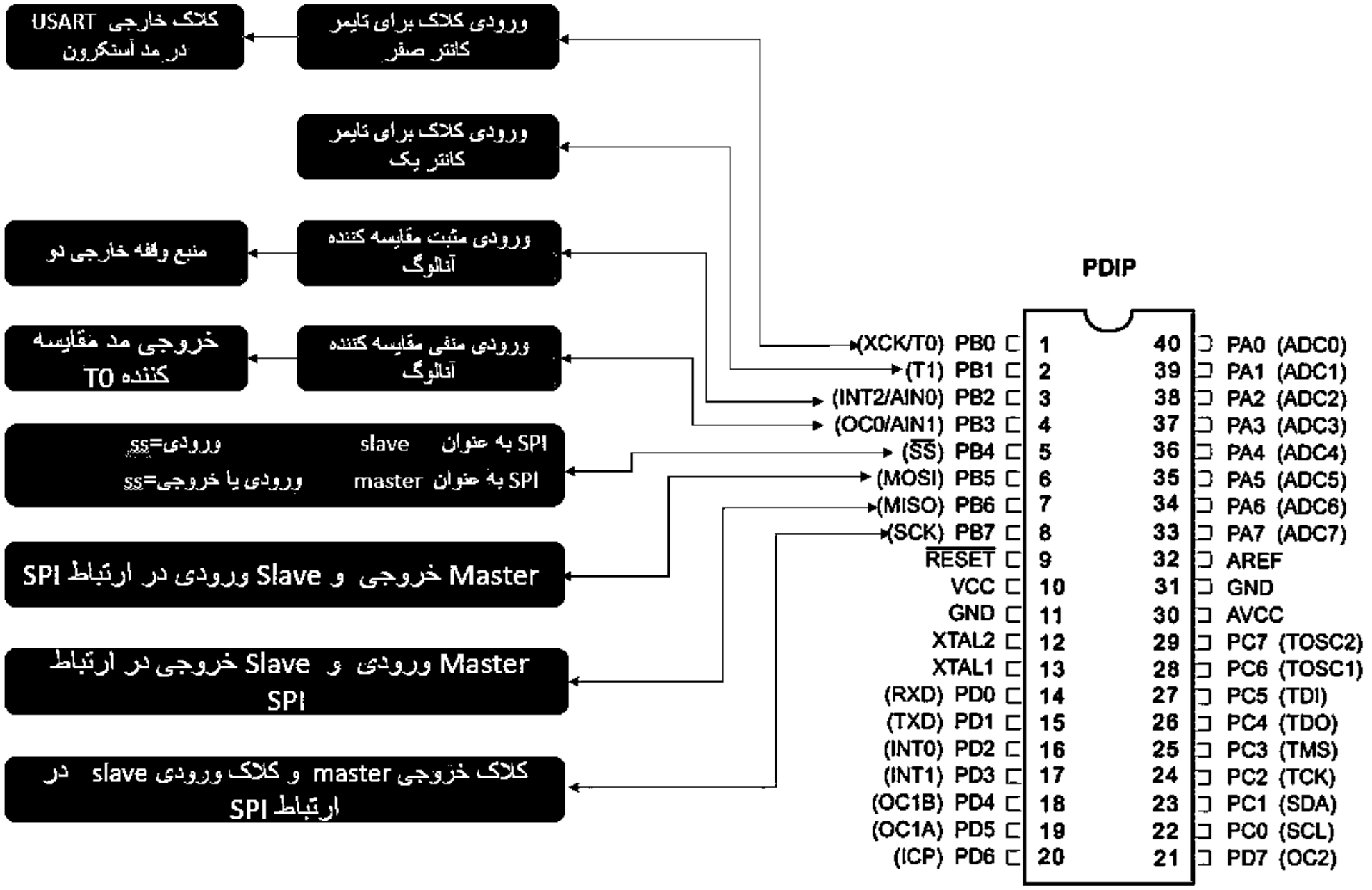
پورت A : پایه های PA0 تا PA7 پورت A را تشکیل می دهند.
در حالت عادی می توان ای این پایه ها به عنوان ورودی و خروجی استفاده کرد.
کاربرد دیگر پایه های A ،به عنوان ورودی مالتی پلکسر مبدل آنالوگ به دیجیتال می باشد





کاربرد سایر پورتها در ATmega16

پورت B : پایه های PB0 تا PB7 پورت B را تشکیل می دهند.
در حالت عادی می توان ای این پایه ها به عنوان ورودی و خروجی استفاده کرد.
کاربرد دیگر پایه هایB ارتباط دهی سریال SPI،وقفه خارجی دو ،ورودی مقایسه کننده آنالوگ ،خروجی مد مقایسه کننده تایمر صفر و ورودی کانتر صفر و یک می باشد


کلاک خارجی USART در مد آسنکرون
ورودی کلاک برای تایمر کانتر صفر
ورودی کلاک برای تایمر کانتر یک
منبع وقفه خارجی دو
ورودی مثبت مقایسه کننده آنالوگ
خروجی مد مقایسه کننده T0
ورودی منفی مقایسه کننده آنالوگ
SPI به عنوان slave ورودی=ss
SPI به عنوان master ورودی یا خروجی=ss
Master خروجی و Slave ورودی در ارتباط SPI
Master ورودی و Slave خروجی در ارتباط SPI
کلاک خروجی master و کلاک ورودی slave در ارتباط SPI
PDIP
(XCK/T0) PB0 1
(T1) PB1 2
(INT2/AIN0) PB2 3
(OC0/AIN1) PB3 4
(SS) PB4 5
(MOSI) PB5 6
(MISO) PB6 7
(SCK) PB7 8
RESET 9
VCC 10
GND 11
XTAL2 12
XTAL1 13
(RXD) PD0 14
(TXD) PD1 15
(INT0) PD2 16
(INT1) PD3 17
(OC1B) PD4 18
(OC1A) PD5 19
(ICP) PD6 20
40 PA0 (ADC0)
39 PA1 (ADC1)
38 PA2 (ADC2)
37 PA3 (ADC3)
36 PA4 (ADC4)
35 PA5 (ADC5)
34 PA6 (ADC6)
33 PA7 (ADC7)
32 AREF
31 GND
30 AVCC
29 PC7 (TOSC2)
28 PC6 (TOSC1)
27 PC5 (TDI)
26 PC4 (TDO)
25 PC3 (TMS)
24 PC2 (TCK)
23 PC1 (SDA)
22 PC0 (SCL)
21 PD7 (OC2)

کاربرد سایر پورتها در ATmega16

- **پایه PB0(XCK/T0)** : اگر کانتر صفر استفاده شود ورودی کانتر صفر پایه T0 خواهد بود . همچنین اگر USART در مد سنکرون کار کند پایهXCK به عنوان خروجی کلاک همزمان کننده USART عمل می کند.

- **پایه PB1(T0)** : اگر کانتر یک استفاده شود ورودی کانتر یک پایه T1 خواهد بود.

- **پایه PB2(INT2/AIN0):** اگر وقفه خارجی دو توسط رجیسترهای مربوطه فعال شود آنگاه پایه INT2 به عنوان ورودی وقفه خارجی دو عمل می کند.

- **پایه PB3(OC0/AIN1):** در صورتی که از مد مقایسه ای تایمر صفر استفاده کنیم و خروجی مقایسه ای فعال شده باشد پایه OC0 به عنوان خروجی مد مقایسه ای تایمر صفر و در مد PWM تایمر صفر به عنوان خروجی سیگنال PWM تولید شده عمل می کند.

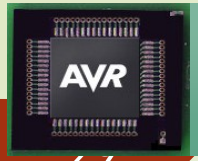
کاربرد سایر پورتها در ATmega16

- **پایه PB4(SS)** : در شرایطی که از ارتباط دهی SPIاستفاده کنیم و میکروکنترلر در حالت Slaveباشد پایه ss به عنوان ورودی انتخاب slaveعمل می کند.

- **پایه PB5(MOSI)** : اگر از ارتباط دهی SPI استفاده کنیم پایه mosi به عنوان خروجی در حالت master و به عنوان ورودی در حالت slaveعمل می کند .

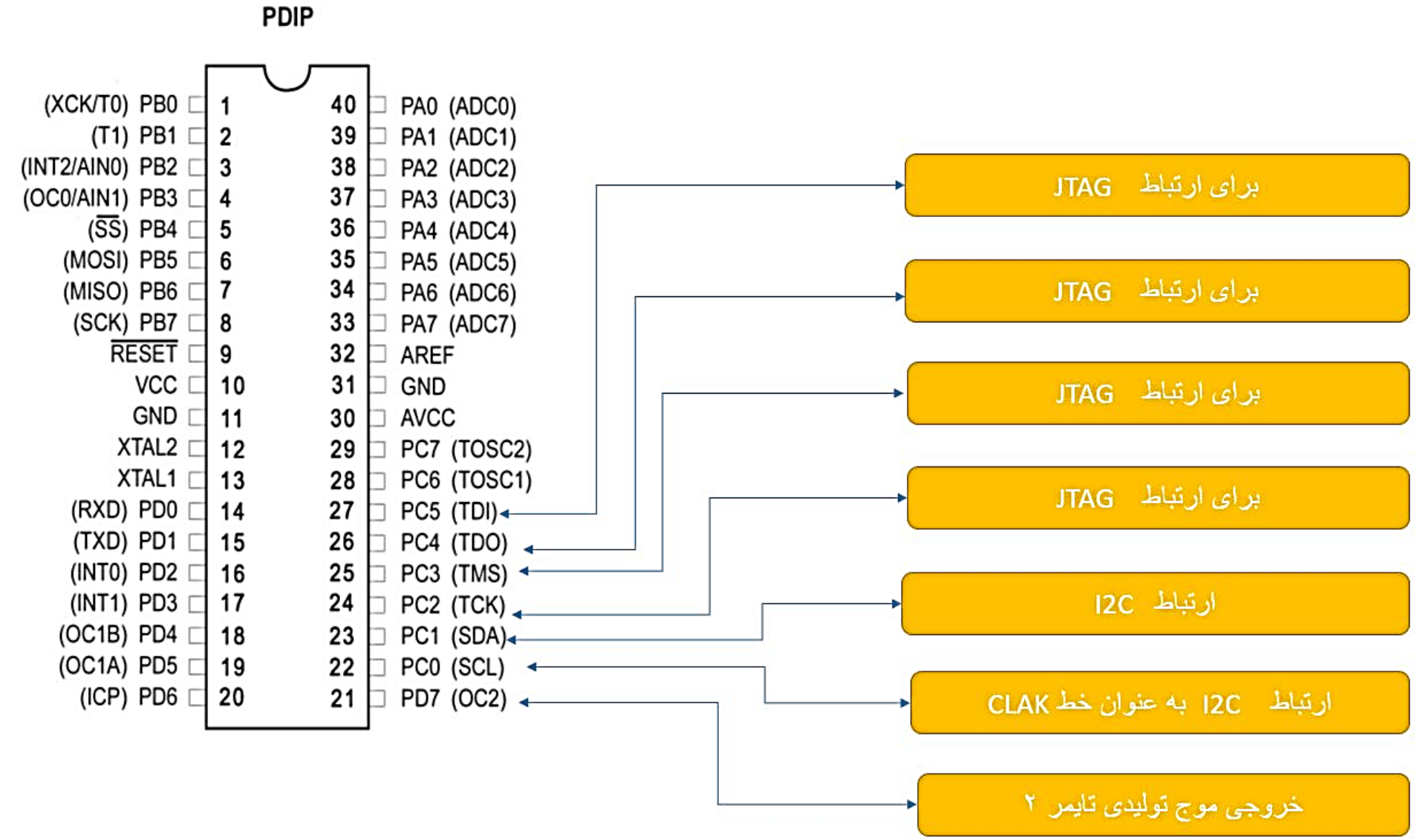
- **پایه PB6(MOSI):** اگر از ارتباط دهی SPI استفاده کنیم پایه mosi به عنوان ورودی در حالت master و به عنوان خروجی در حالت slaveعمل می کند .

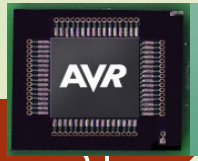
- **پایه PB7(SCK):** در ارتباط دهی SPI پایه SCK به عنوان خروجی master و به عنوان ورودی کلاک slaveعمل می کند زمانی که slave انتخاب شود این پایه ورودی خواهد بود و اگر master انتخاب شود باید توسط DDRB.7 جهت داده تنظیم گردد.



کاربرد سایر پورتها در ATmega16

پورت C : پایه های PC0 تا PC7 پورت C را تشکیل می دهند.
در حالت عادی می توان ای این پایه ها به عنوان ورودی و خروجی استفاده کرد.
کاربرد دیگر این پورت ارتباط دهی سریال TW1 ،ارتباط دهی استاندارد JTAG و کریستال پالس ساعت واقعی RTCتایمر دو است.


PDIP
(XCK/T0) PB0 1
(T1) PB1 2
(INT2/AIN0) PB2 3
(OC0/AIN1) PB3 4
(SS) PB4 5
(MOSI) PB5 6
(MISO) PB6 7
(SCK) PB7 8
RESET 9
VCC 10
GND 11
XTAL2 12
XTAL1 13
(RXD) PD0 14
(TXD) PD1 15
(INT0) PD2 16
(INT1) PD3 17
(OC1B) PD4 18
(OC1A) PD5 19
(ICP) PD6 20
40 PA0 (ADC0)
39 PA1 (ADC1)
38 PA2 (ADC2)
37 PA3 (ADC3)
36 PA4 (ADC4)
35 PA5 (ADC5)
34 PA6 (ADC6)
33 PA7 (ADC7)
32 AREF
31 GND
30 AVCC
29 PC7 (TOSC2)
28 PC6 (TOSC1)
27 PC5 (TDI)
26 PC4 (TDO)
25 PC3 (TMS)
24 PC2 (TCK)
23 PC1 (SDA)
22 PC0 (SCL)
21 PD7 (OC2)
برای ارتباط JTAG
برای ارتباط JTAG
برای ارتباط JTAG
برای ارتباط JTAG
ارتباط I2C
ارتباط I2C به عنوان خط CLAK
خروجی موج تولیدی تایمر ۲



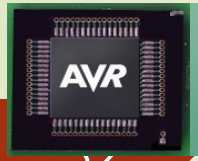
کاربرد سایر پورتها در ATmega16

- **پایه PC0(SCL)** : زمانی که ارتباط دهی سریال دو سیمه TW1 استفاده شود،پایه SCLبه عنوان کلاک عمل می کند. این پایه ،کلاک استاندارد 12C است.

- **پایه PC1(SDA)** : زمانی که ارتباط دهی سریال دو سیمه TW1 استفاده شود،پایه SDAبه عنوان خط دیتا عمل می کند. این پایه ،دیتا استاندارد 12C است.

- **پایه PC2(TCK):** در ارتباط دهی استاندارد JTAG که برای برنامه ریزی میکروکنترلر نیز استفاده می شود پایه TCK به عنوان کلاک تست به صورت سنکرون عمل می کند .

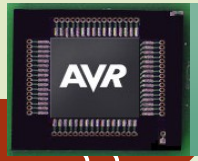
- **پایه PC3(TMS):** در ارتباط دهی JTAG پایهTMS برای انتخاب مد تست می باشد در صورت فعال بودن JTAG دیگر نمی توان از این پایه به عنوان ورودی و خروجی استفاده کرد.



کاربرد سایر پورتها در ATmega16

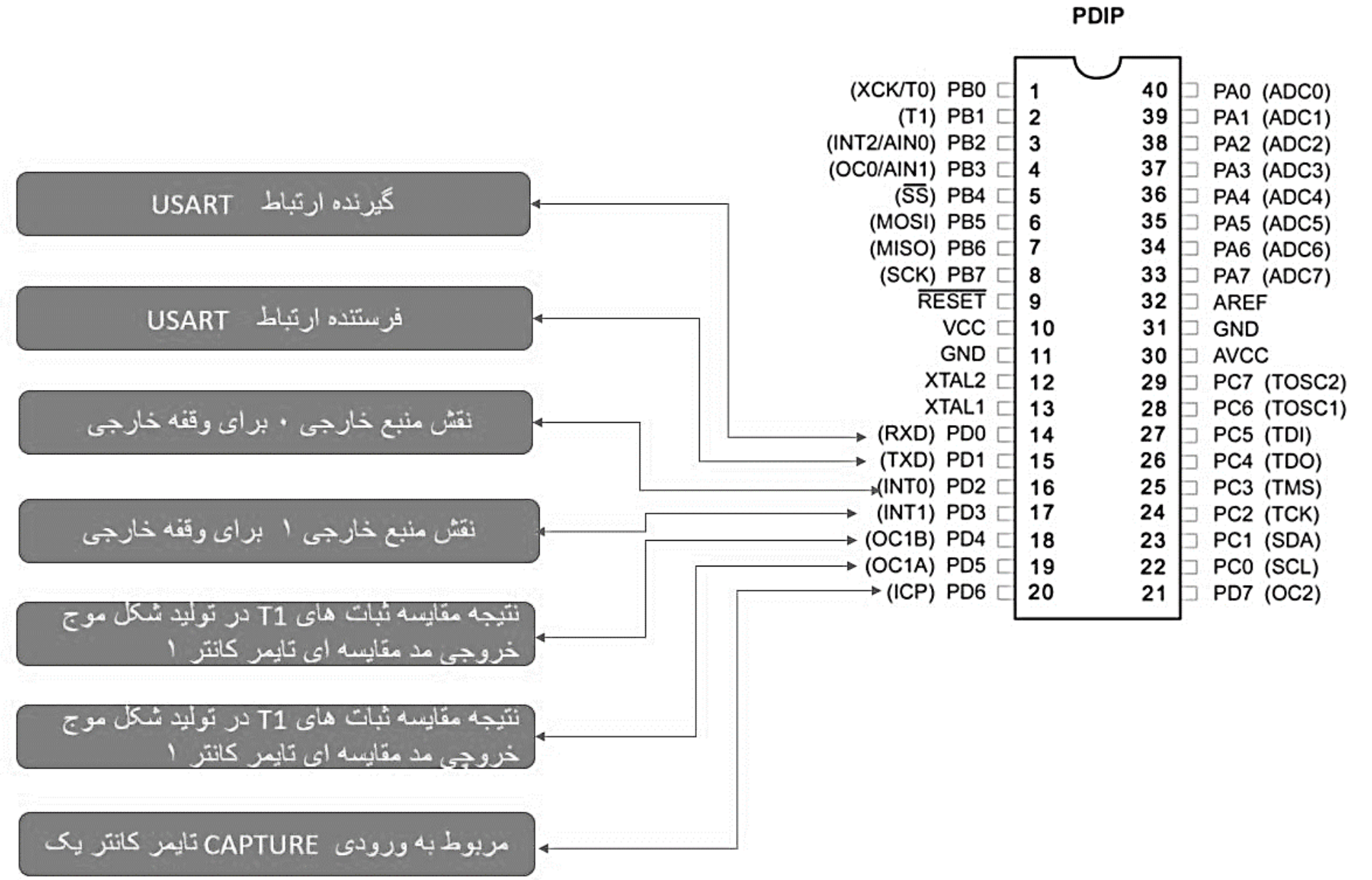
- **پایه PC4(TDO)** : در ارتباط دهی JTAG پایه TDO به عنوان خروجی داده سریال عمل می کند و در صورت فعال بودن JTAG دیگر نمی توان از این پایه به عنوان ورودی و خروجی استفاده کرد.
- **پایه PC5(TDI)** : در ارتباط دهی JTAG پایه TDI به عنوان ورودی داده سریال عمل می کند و در صورت فعال بودن JTAG دیگر نمی توان از این پایه به عنوان ورودی و خروجی استفاده کرد.

- **پایه PC6(TOSC1):** در صورتی که بخواهیم از RTC تایمر دو استفاده کنیم باید از کریستال 32.768HZ پالس ساعت استفاده کنیم پایه TOSC1 به عناون پایه اول اسیلاتور پالس زمان واقعی می باشد.
- **پایه PC7(TOSC2):** در صورتی که بخواهیم از RTC تایمر دو استفاده کنیم باید از کریستال 32.768HZ پالس ساعت استفاده کنیم پایه TOSC2 به عناون پایه دوم اسیلاتور پالس زمان واقعی می باشد.



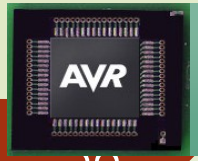
کاربرد سایر پورتها در ATmega16

پورت D: پایه های PC0 تا PC7 پورت C را تشکیل می دهند.
در حالت عادی می توان ای این پایه ها به عنوان ورودی و خروجی استفاده کرد.
کاربرد دیگر این پورت ارتباط دهی سریالUSART ، وقفه های خارجی صفر و یک ،خروجی های مد مد مقایسه ای تایمر یک و خروجی مد مقایسه ای تایمر ۲ و ورودی capture تایمر یک می باشد.


PDIP
(XCK/T0) PB0 1
(T1) PB1 2
(INT2/AIN0) PB2 3
(OC0/AIN1) PB3 4
(SS) PB4 5
(MOSI) PB5 6
(MISO) PB6 7
(SCK) PB7 8
RESET 9
VCC 10
GND 11
XTAL2 12
XTAL1 13
(RXD) PD0 14
(TXD) PD1 15
(INT0) PD2 16
(INT1) PD3 17
(OC1B) PD4 18
(OC1A) PD5 19
(ICP) PD6 20
40 PA0 (ADC0)
39 PA1 (ADC1)
38 PA2 (ADC2)
37 PA3 (ADC3)
36 PA4 (ADC4)
35 PA5 (ADC5)
34 PA6 (ADC6)
33 PA7 (ADC7)
32 AREF
31 GND
30 AVCC
29 PC7 (TOSC2)
28 PC6 (TOSC1)
27 PC5 (TDI)
26 PC4 (TDO)
25 PC3 (TMS)
24 PC2 (TCK)
23 PC1 (SDA)
22 PC0 (SCL)
21 PD7 (OC2)
گیرنده ارتباط USART
فرستنده ارتباط USART
نقش منبع خارجی ۰ برای وقفه خارجی
نقش منبع خارجی ۱ برای وقفه خارجی
نتیجه مقایسه ثبات های T1 در تولید شکل موج خروجی مد مقایسه ای تایمر کانتر ۱
نتیجه مقایسه ثبات های T1 در تولید شکل موج خروجی مد مقایسه ای تایمر کانتر ۱
مربوط به ورودی CAPTURE تایمر کانتر یک

کاربرد سایر پورتها در ATmega16

- **پایه PD0(RXD)** : در ارتباط دهی سریال USART پایه RXD بدون در نظر گرفتن DDRD.0 به عنوان ورودی داده سریال پیکربندی می شود.
- **پایه PD1(TXD)** : در ارتباط دهی سریال USART پایه TXD بدون در نظر گرفتن DDRD.1 به عنوان وخروجی داده سریال پیکربندی می شود.
- **پایه PD2(INT0):** اگر وقفه خارجی صفر شده باشد پایه INT0 به عنوان منبع ورودی وقفه صفر عمل می کند.
- **پایه PD3(INT1):** اگر وقفه خارجی یک شده باشد پایه INT1 به عنوان منبع ورودی وقفه یک عمل می کند.



کاربرد سایر پورتها در ATmega16

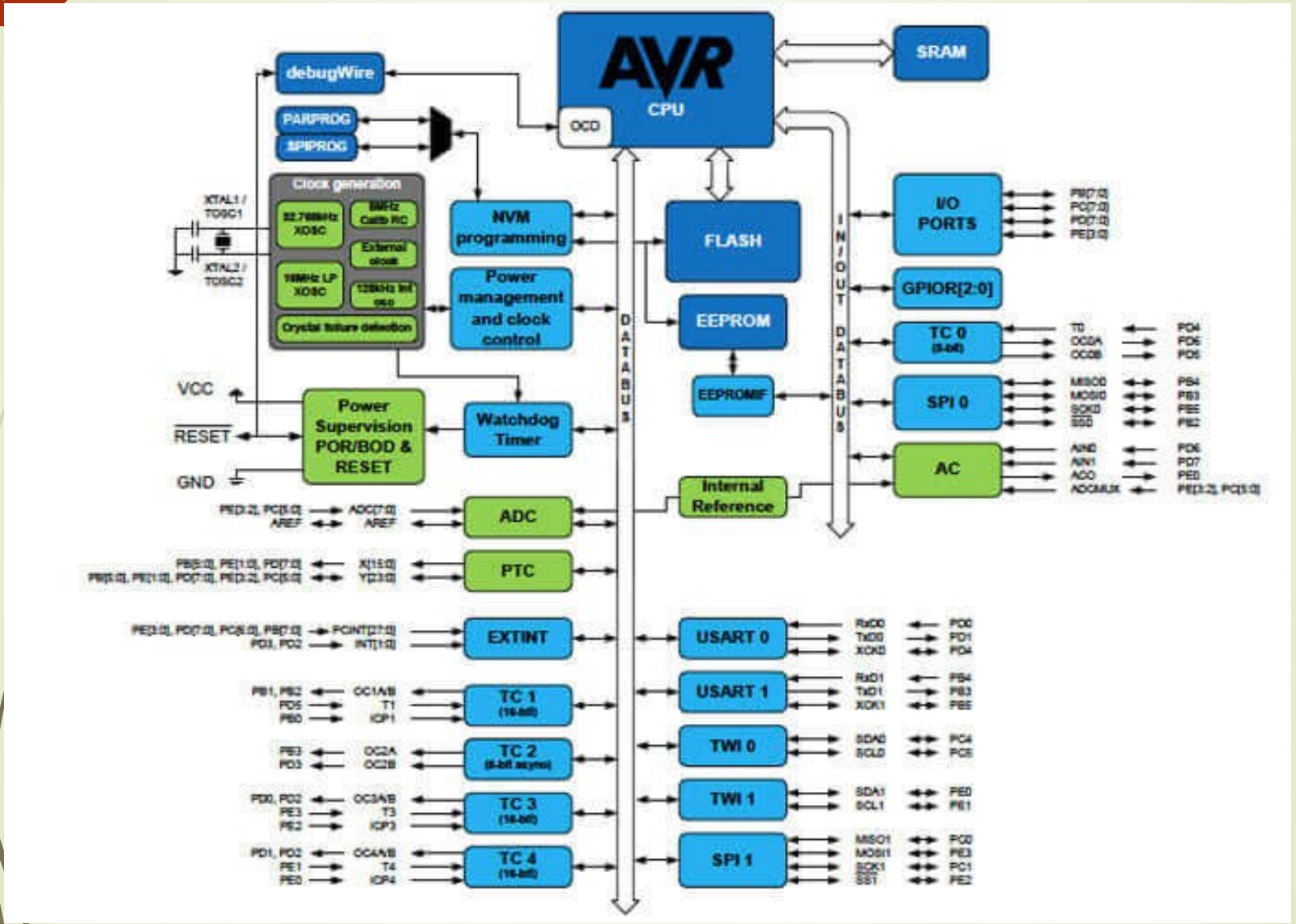
- **پایه PD4(OC1B)** : در صورت استفاده از مد مقایسه ای تایمر یک و فعال کردن خروجی مقایسه ای ،پایه OC1B به عنوان خروجی دوم مقایسه ای تایمر یک عمل می کند.

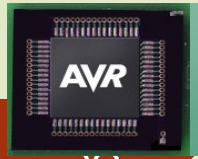
- **پایه PD5(OC1A)** :در صورت استفاده از مد مقایسه ای تایمر یک و فعال کردن خروجی مقایسه ای ،پایه OC1A به عنوان خروجی اول مقایسه ای تایمر یک عمل می کند.

- **پایه PD6(ICP):** اگر واحد تسخیر کننده یعنی مد captureتایمر یک فعال شده باشد پایه ICP به عنوان ورودی capture تایمر یک عمل می کند.

- **پایه PD7(OCP):** در صورت استفاده از مد مقایسه ای تایمر دو و فعال کردن خروجی مقایسه ای ،پایه OC2 به عنوان خروجی مقایسه ای تایمر دو عمل می کند.

ساختار داخلی میکروکنترلر ATmega16



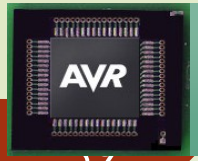


ساختار داخلی میکروکنترلرATmega16

همانطور که در شکل ۱-۶ می بینیم اجزای درونی میکروکنترلر توسط باس های داخلی به هم متصل شده اند و بخش دیگری که با نام خط چین با نام AVR CPU مشخص شده است پردازنده اصلی AVR می باشد.

کار اصلی یک CPU دسترسی به حافظه ها ،محاسبات ریاضی و منطقی ،کنترل وسایل جانبی و بررسی وقفه های هر یک از قسمت ها می باشد.

شمارنده برنامه ،اشاره گر پشته ،رجیستر دستورات ،آشکارساز دستورات ،رجیستر های X,Y,Z ،رجیسترهای همه منظوره ،واحد محاسبه و منطق (ALU) و رجیستر وضعیت،از اجزای تشکیل دهنده CPUمیکروکنترلر AVRهستند که به تشریح هر کدام از آنها می پردازیم.



ساختار داخلی میکروکنترلرATmega16

- **شمارنده برنامه PC:** CPU میکروکنترلر برای اینکه دستورات را از اولین آدرس حافظه برنامه خط به خط بخواند نیاز به یک شمارنده می باشد.افزایش PCآدرس خط بعدی را برای اجرای دستورات فراهم می کند.
- **اشاره گر پشته SP: در**میکروکنترلAVR اشاره گر پشته از دو رجیستر ۸ بیتی استفاده می کند. اشاره گر پشته برای ذخیره موقت اطلاعات در دستورهای فراخوانی CALL,PUSHوPOP و متغیر های محلی ،روتین های وقفه و توابع استفاده می شود.

- **رجیستر دستورات IR:** در اصل CPUمیکروکنترلر برای فهمیدن انجام یک دستورالعمل از کد ماشین استفاده می کند و به هر یک از این کد ها یک سنبل در زبان اسمبلی اختصاص داده می شود.
- **آشکار ساز دستورات ID:** شمارنده برنامه افزایش می یابد و کد هر دستور توسط CPUخوانده می شود و با توجه به رجیستر دستورات ،کد خوانده شده آشکار می گردد و CPUمتوجه می شود که کد دستور به چه معنی است.

ساختار داخلی میکروکنترلرATmega16

- **رجیستر های همه منظوره : (general purpose registers)**

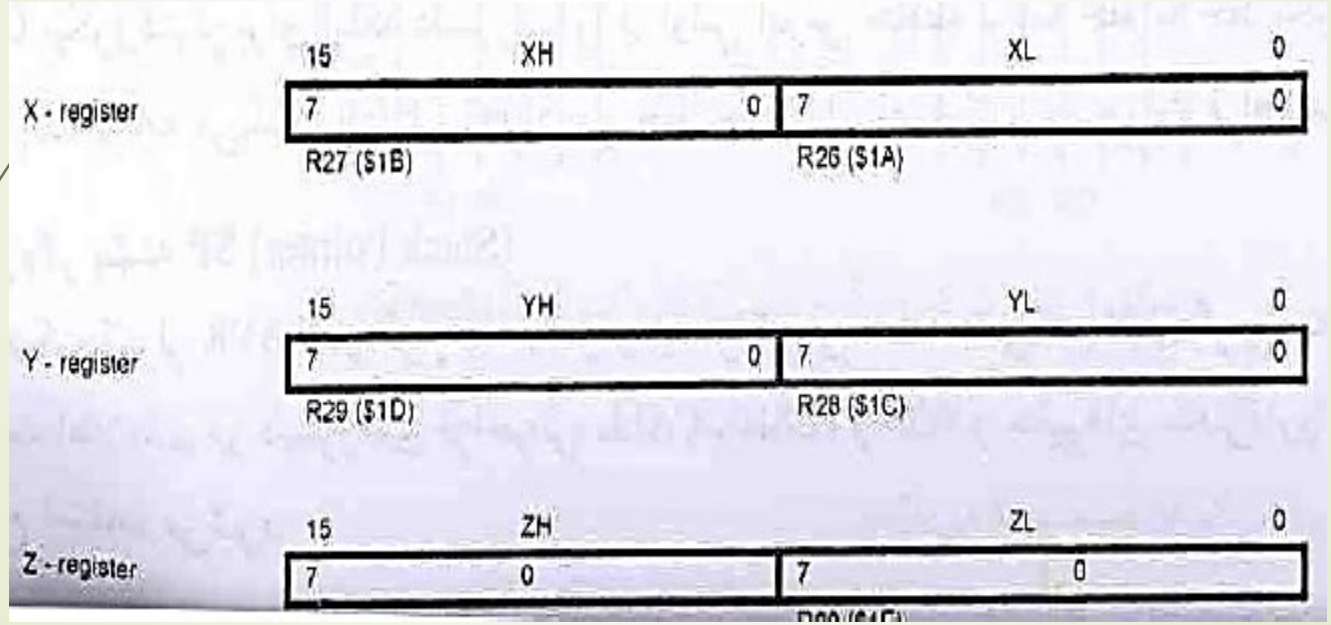
میکروکنترلر های AVRدارای ۳۲ رجیستر همه منظوره هستند این رجیستر ها قسمتی از حافظه SRAM داخلی میکروکنترلر می باشند که اکثر دستورات اسمبلی AVR مستقیما با این رجیسترها دسترسی دارند و در یک کلاک اجرا می شوند.

7 0	Addr.	
R0	$00	
R1	$01	
R2	$02	
...		
R13	$0D	
R14	$0E	
R15	$0F	
R16	$10	
R17	$11	
...		
R26	$1A	X-register Low Byte
R27	$1B	X-register High Byte
R28	$1C	Y-register Low Byte
R29	$1D	Y-register High Byte
R30	$1E	Z-register Low Byte
R31	$1F	Z-register High Byte

ساختار داخلی میکروکنترلرATmega16

- **رجیستر های X,Y وZ:**

وظیفه این سه رجیستر که از ترکیب رجیسترهای R26تا R31 بوجود می آیند اشاره گر ۱۶ بیتی جهت آدرس دهی غیر مستقیم فضای داده هستند و بسیاری از دستورات اسمبلی AVR با این سه رجیستر عمل می کنند.



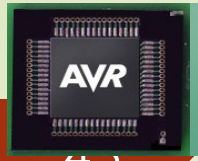
ساختار داخلی میکروکنترلرATmega16

- **واحد محاسبه و منطق: (arithmetic logic unit)**

ALU در میکروکنترلرAVR به صورت مستقیم با تمام ۳۲ رجیستر همه منظوره در ارتباط است.عملیاتهای محاسباتی با رجیسترهای همه منظوره در یک کلاک سیکل اجرا می شوند.

- **رجیستر وضعیت Sreg:** بیت های این رجیستر در واقع پرچمهایی هستند که CPU را از نتایج دستورات و وضعیت برنامه آگاه می سازد.

SREG (Status Register)								
Bit number	7	6	5	4	3	2	1	0
Bit name	I	T	H	S	V	N	Z	C
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0



انواع حافظه در میکروکنترلر های AVR

میکروکنترلر های AVR دارای سه نوع حافظه هستند.

1. **حافظه Flash**
2. **حافظه EEPROM**
3. **حافظه SRAM**

که به توضیح در مورد هر یک از آنها خواهیم پرداخت.

انواع حافظه در میکروکنترلر های AVR

1. **حافظه Flash :** در این حافظه کدهای برنامه یعنی همان فایل HEX.* که توسط پروگرامر بر روی تراشه لود می شود قرار می گیرد و CPU میکروکنترلر برنامه را که اجرا می کند کد دستورالعمل را از این حافظه برداشت می کند.

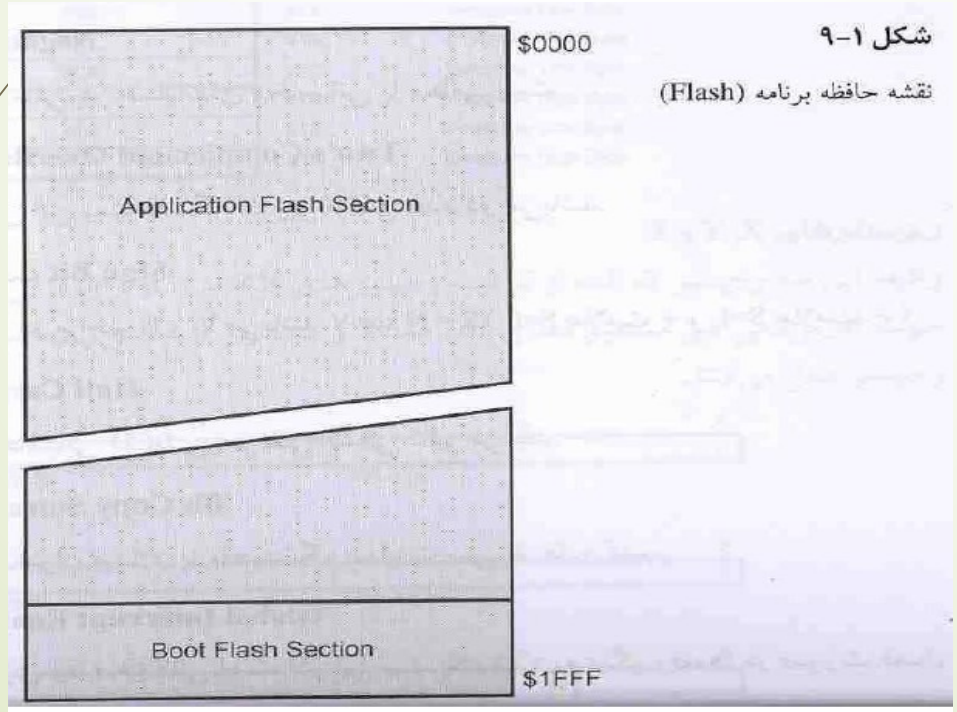
این حافظه دارای دو قسمت Aplication وBoot loader هستند که در قسمت aplication کد های برنامه قرار می گیرند اما در قسمت boot این امکان را فراهم می کند که میکروکنترلر بدون استفاده از ابزار پروگرامر ،برنامه حافظه فلاش را تغییر دهد.

انواع حافظه در میکروکنترلر های AVR

به طور مثال نحوه تعیین یک آرایه در حافظه ثابت بصورت زیر است:

```
flash char row []={0xfe,0xfd,0xfb,0xf7}  //flash در ذخیره شده بیتی ۸ عدد ۴
```

شکل ۱-۹ آدرس و تقسیم بندی حافظه flash را به دو قسمت نشان می دهد.



شکل ۱-۹

نقشه حافظه برنامه (Flash)

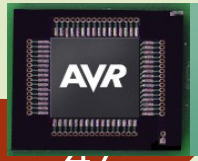
انواع حافظه در میکروکنترلر های AVR

1. حافظه EEPROM: این حافظه جزو حافظه های ماندگار است که میکروکنترلر می تواند اطلاعاتی را ازآن بخواند .یعنی در صورت قطع تغذیه اطلاعات آن از بین نمی رود
از این حافظه زمانی استفاده می شود که میکروکنترلر باید دیتای را در خود ثبت کند و بعدا آن دیتا را به کاربر اعلام کند .
برای خواندن و نوشتن در حافظه EEPROM باید یک زمان تاخیری در نظر گرفت .در جدول ۱–۱۵ مدت زمان مناسب با کلاک یک مگاهرتز 8.5 ms بیان شده است.

Symbol	Number of Calibrated RC Oscillator Cycles(1)	Typ Programming Time
EEPROM write (from CPU)	8448	8.5 ms

Note: 1. Uses 1 MHz clock, independent of CKSEL Fuse setting.

جدول ۱–۱۵ زمان برنامه‌ریزی حافظه eeprom



انواع حافظه در میکروکنترلر های AVR

1. **حافظه SRAM :** همان طور که بحث شد کد های برنامه در حافظه FLASH قرار می گیرند و CPUمیکروکنترلر کدهای دستورات را آشکار می کند حال باید حاصل دستورات انجام شده در یک حافظه موقت ذخیره گردد این نوع حافظه در میکروکنترلر AVR از نوع static RAM می باشد.

محتوای این حافظه به قطع تغذیه پاک می گردد.و در صورتی که میکروکنترلر راresetکنیم محتوای رجیسترها صفر می شود اما محتوای حافظه sram صفر نمی شود.

به طور مثال متغییر M را در حافظه SRAM ذخیره کنید.

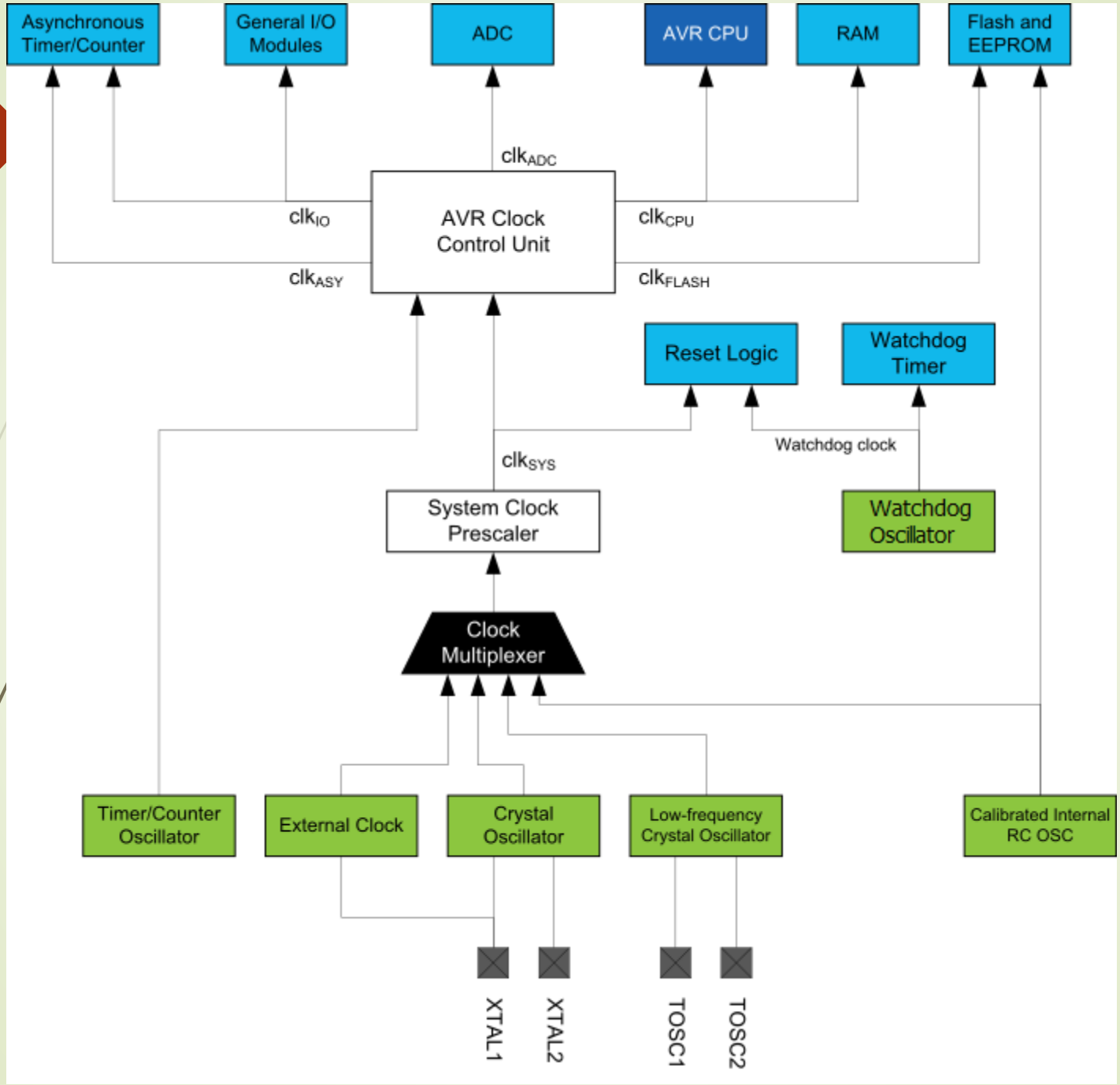
```
Unsigned char M=0x12;
```

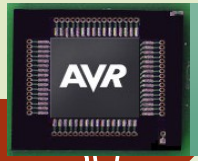
بنابراین مشخص می شود در تعریف هر متغیری که قبل از آن از کلمه ای کلیدی Flash و eepromاستفاده نشود به مفهوم ذخیره متغیر در حافظه SRAM می باشد.

کلاک سیستم در میکروکنترلرهای AVR

سیستم پالس ساعت در avr متنوع است .علاوه بر کریستالهای خارجی می توانند از نوسان ساز داخلی نیز استفاده کنند.

شکل ۱-۱۱ سیستم پالس ساعت را برای قسمت های مختلف درونی نشان می دهد.


Asynchronous Timer/Counter
General I/O Modules
ADC
AVR CPU
RAM
Flash and EEPROM
clkADC
clkIO
AVR Clock Control Unit
clkCPU
clkASY
clkFLASH
Reset Logic
Watchdog Timer
Watchdog clock
clkSYS
System Clock Prescaler
Watchdog Oscillator
Clock Multiplexer
Timer/Counter Oscillator
External Clock
Crystal Oscillator
Low-frequency Crystal Oscillator
Calibrated Internal RC OSC
XTAL1
XTAL2
TOSC1
TOSC2



کلاک سیستم در میکروکنترلرهای AVR

همانطور که در شکل ۱-۱۱ میبینیم منبع کلاک توسط یک مالتی پلکسر ،پالس لازم را به واحد کنترل کننده کلاک اعمال می کند. در واقع کلاک لازم جهت راه اندازی می تواند یکی از نوسان سازها باشد و امکان استفاده همزمان از آنها وجود ندارد.

پالس CLKCPU به هسته CPU و SRAM داخلی اعمال می شود،پالس CLKADC کلاک لازم را جهت مبدل آنالوگ به دیجیتال فراهم می سازد که توسط قسمت مبدل ADC می تواند به طور مجزا تقسیم گردد،پالس CLKFLASH کلاک لازم را برای حافظه eepromوflash داخلی فراهم می سازد،پالس CLKI/O برای تولید پالس ماژولهای ورودی و خروجی نظیر USART,SPI,شمارنده و وقفه ها بکار برده می شود.
پالس CLKASY برای راه اندازی آسنکرون تایمر یا کانتر دو برای استفاده از فرکانس 32.768KHZ اسیلاتور RTC است.همچنین تایمر WATCHDOG از یک اسیلاتور مجزا داخلی استفاده می کند.

کلاک سیستم در میکروکنترلرهای AVR

منابع پالس ساعت میکروکنترلرهای AVR

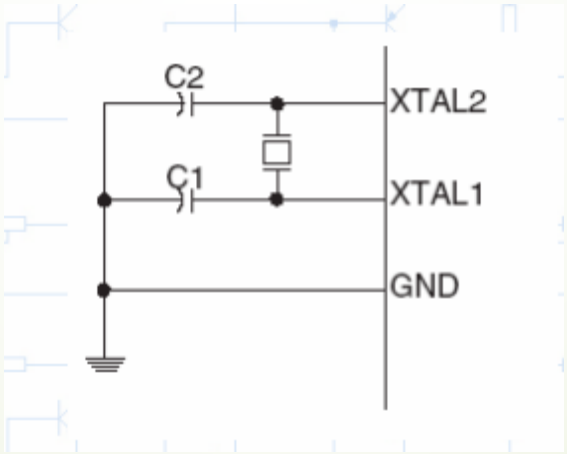
قبلا گفته شد که میکروکنترلرهای AVR می توانند از نوسانساز های داخلی و یا خارجی استفاده کنند. برای تعیین نوسان ساز سیستم باید از فیوزبیت های میکروکنترلر ATmega16 طبق جدول ۱-۱۶ استفاده کنیم.

CKSE3...0	DEVICE CLOCKING OPTION
1111-1010	کریستال یا رزوناتور سرامیکی خارجی
1001	کریستال فرکانس پائین خارجی
1000-0101	اسیلاتور RC خارجی (مقاومت – خازن)
0100-0001	اسیلاتور RC کالیبره شدهٔ داخلی
0000	کلاک خارجی

کلاک سیستم در میکروکنترلرهای AVR

نوسان ساز با کریستال خارجی

برای استفاده از کریستال خارجی ،باید فیوز بیت های CKSEL3.0 را طبق جدول ۱-۱۶ به صورت 1111 برنامه ریزی کنیم.پایه های خارجی XTAL1وXTAL2 طبق شکل ۱-۱۲ توسط دو خازن عدسی با مقادیر یکسان به یک کریستال متصل میگردد.



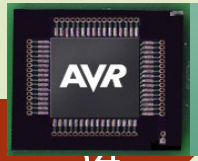
کلاک سیستم در میکروکنترلرهای AVR

خازنهای C2وC1 را معمولا 22PF انتخاب می کنیم . وظیفه این دو خازن حذف نویز الکترومغناطیس اطراف کریستال می باشد که طبق **جدول ۱-۱۷** با توجه به کریستال استفاده شده تعیین می شود.

CKOPT	CKSEL3..1	Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
1	101[1]	0.4 - 0.9	–
1	110	0.9 - 3.0	12 - 22
1	111	3.0 - 8.0	12 - 22
0	101, 110, 111	1.0 ≤	12 - 22

۱- این گزینه در صورت استفاده از رزوناتور سرامیکی، انتخاب می‌شود.

جدول ۱-۱۷ انتخاب خازن‌ها در فرکانس‌های مختلف



کلاک سیستم در میکروکنترلرهای AVR

هنگام استفاده از کریستال خارجی می توان با فعال کردن فیوز بیت CKOPT دامنه نوسان اسیلاتور را حداکثر کرد. در این حالت اسیلاتور به صورت Rail –to-Rail عمل می کندیعنی اگر تغذیه میکروکنترلر ۵ولت باشد حداکثر دامنه پالس ساعت نیز ۵ ولت خواهد بود.

این حالت برای محیطهای پر نویز مانند کارخانه های صنعتی بسیار مفید می باشد.

زمان start-up برای استفاده از کریستال خارجی توسط فیوزبیت های SUT0وSUT1 طبق جدول ۱–۱۸ تعیین می گردد.
منظور از زمان start-up مت زمانی است که تغذیه به میکروکنترلر وصل می شود.

کلاک سیستم در میکروکنترلرهای AVR

CKSEL0	SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
0	00	258 CK(1)	4.1 ms	Ceramic resonator, fast rising power
0	01	258 CK(1)	65 ms	Ceramic resonator, slowly rising power
0	10	1K CK(2)	–	Ceramic resonator, BOD enabled
0	11	1K CK(2)	4.1 ms	Ceramic resonator, fast rising power
1	00	1K CK(2)	65 ms	Ceramic resonator, slowly rising power
1	01	16K CK	–	Crystal Oscillator, BOD enabled
1	10	16K CK	4.1 ms	Crystal Oscillator, fast rising power
1	11	16K CK	65 ms	Crystal Oscillator, slowly rising power

۱- این گزینه‌ها فقط برای زمانی استفاده می‌شوند که فرکانس بالا نباشد و برای کریستال‌ها مناسب نیستند.
۲- این گزینه‌ها فقط برای نوسان‌سازهای رزوناتور سرامیکی مفید هستند.

جدول ۱-۱۸ انتخاب زمان شروع (Start-up) در حالت نوسان‌ساز خارجی

کلاک سیستم در میکروکنترلرهای AVR

نوسان ساز با کریستال فرکانس پایین

منظور از کریستال فرکانس پایین ، کریستال 32.768KHZ می باشد در صورت که فیوزبیت های CKSEL3.0 به صورت 1001 برنامه ریزی شوند پالس ساعت سیستم از کریستال خارجی فرکانس پایین استفاده می کند.

در این حالت اگر فیوز بیت CKOPT فعال شود خازن داخلی بین دو پایه XTAL1وXTAL2 فعال می گردد و مقدار این خازن 36pF است.

زمان start-up در این حالت در جدول ۱-۱۹ تعیین می شود.

SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	1K CK(1)	4.1 ms	Fast rising power or BOD enabled
01	1K CK(1)	65 ms	Slowly rising power
10	32K CK	65 ms	Stable frequency at start-up
11	Reserved		

۱- این گزینه‌ها برای کاربردهایی است که پایداری فرکانسی در زمان شروع مهم نباشد.

جدول ۱-۱۹ انتخاب زمان شروع (Start-up) در حالت نوسان‌ساز خارجی فرکانس پایین

کلاک سیستم در میکروکنترلرهای AVR

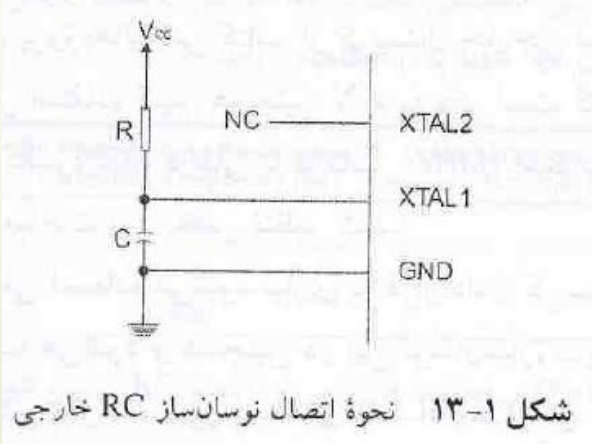
نوسان ساز با RC خارجی

از اسیلاتور با RCخارجی در کاربردهایی که به تغییرات زمان و فرکانس حساسیت نداشته باشیم استفاده می کنیم..

فرکانس این اسیلاتور از رابطه زیر به دست می آید که در آن مقدار خازن حداقل باید 22PF انتخاب شود و مقدار مقاومت بین 3Kاهم تا 100K اهم انتخاب شود.

$$F=1/(3RC)$$

شکل ۱-۱۳ نحوه ی استفاده از اسیلاتور خارجی RC را نشان می دهد.



شکل ۱-۱۳ نحوهٔ اتصال نوسان‌ساز RC خارجی

کلاک سیستم در میکروکنترلرهای AVR

نوسان ساز با اسیلاتور RC کالیبره شده داخلی

اسیلاتور RCکالیبره شده می تواند فرکانس های ثابت1MHZ,2MHZ,4MHZ و8mhz را در شرایط تغذیه 5v+ و در دمای 25c سانتیگراد ایجاد نماید.

فرکانس کاری این اسیلاتور به شده به ولتاژ تغذیه ،درجه حرارت محیط و مقدار بایت رجیستر OSCCAL وابسته می باشد.

SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	6 CK	–	BOD enabled
01	6 CK	4.1 ms	Fast rising power
10[1]	6 CK	65 ms	Slowly rising power
11	Reserved		

۱- زمان شروع به طور پیش فرض بر روی 65ms انتخاب شده است.

جدول ۱-۲۳ انتخاب زمان شروع (Start-up) در حالت نوسان‌ساز کالیبره شده داخلی

کلاک سیستم در میکروکنترلرهای AVR

رجیستر کالیبراسیون OSCCAL

Bit	7	6	5	4	3	2	1	0	
	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	OSCCAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	Device Specific Calibration Value								

در صورت استفاده از اسیلاتور RCکالیبره شده داخلی در هر بار که میکروکنترلر reset می شود مقدار رجیستر osccal بار می شود و اسیلاتور به طور خودکار تنظیم می گردد.

در حالت عادی استفاده از این نوع اسیلاتور در حدود ۳٪ خطا دارد اما اگر از شرایط تغذیه 5v+ و یا دمای ۲۵ درجه سانتی گراد خارج گردد ممکن است این خطا به ۱۰٪ افزایش یابد که با توجه به اینکه برای دسترسی به حافظه flash و حافظه eeprom از این اسیلاتور استفاده شود ممکن است نتایج نامشخصی داشته باشد

کلاک سیستم در میکروکنترلرهای AVR

با نوشتن مقدار 0x00 کمترین و با نوشتن مقدار 0xFF در این رجیستر بیشترین فرکانس ممکن انتخاب می گردد.

تنظیم دقبق این کالیبراسیون خیلی تضمینی نیست اما مقدار بایت کالیبراسیون را می توانیم در هر شرایط دمایی طوری تنظیم کنیم که خطا حدودا به ۱ کاهش یابد.

OSCCAL Value	Min Frequency in Percentage of Nominal Frequency (%)	Max Frequency in Percentage of Nominal Frequency (%)
$00	50	100
$7F	75	150
$FF	100	200

جدول ۱-۲۴ محدوده فرکانسی نوسان‌ساز RC کالیبره شده داخلی

کلاک سیستم در میکروکنترلرهای AVR

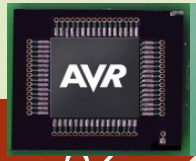
نوسان ساز با کلاک خارجی

در صورت تنظیم فیوزبیت های CKSEL3.0 به صورت 0000 میکروکنترلر AVR پالس ساعت خود را از یک منبع خارجی که به پایه ورودی تقویت کننده نوسانساز یعنی XTAL1 اعمال می شود دریافت می کند.

زمان start-up در موقع استفاده از کلاک خارجی به عنوان پالس ساعت سیستم می تواند توسط فیوزبیت های STU0و STU1 طبق جدول ۱-۲۵ تنظیم گردد.

SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	6 CK	–	BOD enabled
01	6 CK	4.1 ms	Fast rising power
10	6 CK	65 ms	Slowly rising power
11	Reserved		

جدول ۱-۲۵ انتخاب زمان شروع (Start-up) در حالت نوسان‌ساز کلاک خارجی



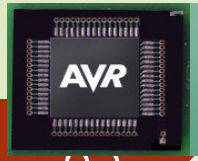
کلاک سیستم در میکروکنترلرهای AVR

نوسان ساز مجزا تایمر یا کانتر دو

برخی از میکروکنترلر های AVR از جمله ATmega16 دارای پایه های TOSC1وTOSC2 می باشند.

زمانی که تایمر دو به عنوان RTC عمل می کند از کریستال پالس ساعت 32.768MHZ که به این دو پایه متصل می گردد کلاک دریافت می کند.

اسیلاتور برای استفاده از این نوع کریستال بهینه شده و نیازی به قرار دادن خازن های حذف نویز نمی باشد زیرا این اسیلاتور فرکانس پایین است و ممکن است سبب توقف نوسانات اسیلاتور شود.



مدهای مختلف Sleep

میکروکنترلرAVR دارای شش مد sleep می باشد که هر یک از این مدها در عملکرد های متفاوتی کاربرد دارند.
هدف از مد های توان کاهش توان مصرفی میکروکنترلر می باشد.
این مد ها باعث می شود تا در پروژه هایی که از باتری برای تغذیه استفاده می شود مصرف جریان میکروکنترلر کاهش یابد.

هنگامی که از مد های مختلف sleepاستفاده می کنیم کلاک قسمتی از اجزای درونی متوقف می شود و با رخ دادن یک وقفه CPU و دیگر قسمت های میکروکنترلر از مد خواب (sleep) بیدار می شوند.

مدهای مختلف Sleep

رجیستر **MCUCR** (MCU Control Register)

Bit	7	6	5	4	3	2	1	0	
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

نیبل بالای این رجیستر برای تنظیم مد های توان می باشد.
بیت های 4,5,7 که طبق جدول ۱-۲۶ یکی از مد های sleep را تعیین می کنند.
بیت 6 ، اگر این بیت یک شود مد sleepفعال می شود.

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby(1)
1	1	1	Extended Standby(1)

۱- مدهای Standby و Extended Standby فقط برای حالت کریستال یا رزوناتور خارجی می‌باشند.

جدول ۱-۲۶ انواع مُدهای Sleep

مدهای مختلف Sleep

- **مد Idle :** در این حالت CPU میکروکنترلر متوقف می شود اما ارتباط دهی سریال SPI,TWIو ,USART متقایسه کننده آنالوگ ،مبدل ADC،تایمر یا کانترها ،watchdog و وقفه ها می توانند بکار خود ادامه دهند.

```
void  idle (void);          //میکروکنترلر به مد Idle می رود.
```

- **مد ADC Noise Reduction :** در این حالت نیز CPU میکروکنترلر متوقف می شود اما ارتباط دهی سریال TW1،مبدل ADC،تایمر یا کانتر ۲ ،watchdog و وقفه های خارجی می توانند بکار خود ادامه دهند.

کاربرد این مد برای کاهش نویز محیط برایADC می باشد.

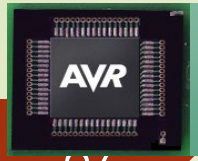
مدهای مختلف Sleep

- **مد Power-down :** در این حالت نوسان ساز خارجی متوقف می گردد.در این مد وقفه های خارجی ،ارتباط سریال TWI و تایمر watchdog می تواند بکار خود ادامه دهند.

```
void powerdown(void);        //میکروکنترلر به مد powerdown می رود.
```

- **مد Power-save :** این حالت بسیار شبیه Power-down **می باشد** . تنها تفاوت بین آنها این است که تایمر یا کانتر دو،می تواند به صورت غیر همزمان (یعنی بیت AS2 در رجیستر ASSRیک شده باشد) در این مد بکار خود ادامه می دهد.

```
void powersave(void);        //میکروکنترلر به مد powersave می رود.
```



مدهای مختلف Sleep

- **مد Standby:** زمانی که کریستال یا رزوناتور خارجی به عنوان منبع پالس ساعت استفاده می شود می توان از این مد که بسیار عملکردی مشابه مد powerdown دارد استفاده کرد با این تفاوت که اسیلاتور می تواند بکار خود ادامه دهد.

```
void standby(void);          //میکروکنترلر به مد standby می رود.
```

- **مد Extended Standby :** زمانی که کریستال یا رزوناتور خارجی به عنوان منبع پالس ساعت استفاده می شود می توان از این مد که بسیار عملکردی مشابه مد powersave دارد استفاده کرد با این تفاوت که اسیلاتور می تواند بکار خود ادامه دهد.

```
void extended_standby(void);
میکروکنترلر به مد extended standby می
رود.//
```

منابع Reset میکروکنترلر ATmega16

Reset در واقع به نوعی یک وقفه است که می تواند توسط هر یک از منابع تحریک کننده آن رخ دهد.

هنگامی که reset رخ می دهد تمامی رجیستر های ورودی و خروجی و همچنین دیگر رجیسترهای کنترلی با توجه به مقادیر پیش فرض در نظر گرفته شده تنظیم می شوند.

بعد از reset یک مدت زمان کوتاه به اندازه زمان تعیین شده توسط start-up طول می کشد ،سپس بعد از پایداری نوسان ساز برنامه از بردار resetآدرس 0x0000 شروع می شود.

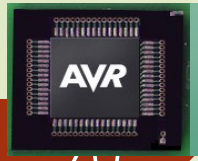
Bit	7	6	5	4	3	2	1	0	
	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	MCUCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	See Bit Description					

منابع Reset میکروکنترلر ATmega16

۱. Power-on Reset : هنگامی که ولتاژ تغذیه از ولتاژ آستانه (VPOT) پایینتر شود بیت PORF در رجیستر MCUCSR یک شده و میکروکنترلر reset می شود.

۲. External Reset : اگر یک پالس با سطح صفر منطقی به مدت طولانی تر از سیکل پالس ساعت میکروکنترلر ،به پایه خارجی Reset اعمال شود ،بیت EXTRF در رجیستر MCUCSR یک شده و میکروکنترلر reset می شود.

۳.Watchdog Reset: اگر تایمر نگهبان زمان (watchdog timer) فعال شود و از مقدار نهایی خود سرریز کند ،بیت WDRF در رجیستر MCUCSR یک شده و میکروکنترلر را reset می کند.



منابع Reset میکروکنترلر ATmega16

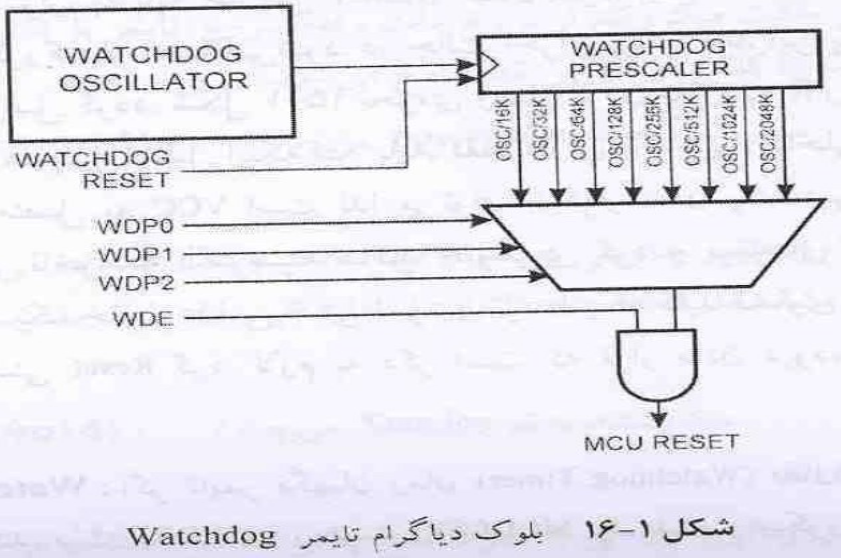
۱. Brown-out Reset : در صورت فعال بودن مدار brown out توسط فیوز بیت BODEN ،اگر مقدار ولتاژ تغذیه از حد ولتاژ 2.7v یا 4v که توسط فیوز بیت BODLEVEL تعیین می گردد پاینتر بیاید ،بیت BORF در رجیستر MCUCSR یک شده و میکروکنترلر را reset می کند.

۲. JTAG AVR Reset: تا وقتی که در حلقه اسکن سیستم JTAG،رجیستر RESET یک منطقی باشد بیت JTRF در رجیستر MCUCSR یک شده و میکروکنترلر reset خواهد شد.

تایمر Watchdog

میکروکنترلر های AVR دارای یک تایمر نگهبان زمان هستند. وظیفه این تایمر در صورت فعال شدن توسط برنامه نویسی ،این است که از عملکرد صحیح برنامه میکروکنترلر می توان اطمینان حاصل کرد .

این تایمر زمانی که فعال می شود توسط کلاک داخلی مجزا با فرکانس 1MHZ شروع به شمارش می کند و بعد از یک مدت زمانی طبق تقسیم فرکانسی تنظیم شده توسط کاربر ،این تایمر به مقدار نهایی خود می رسد و میکروکنترلر را reset می کند.



شکل ۱-۱۶ بلوک دیاگرام تایمر Watchdog

تایمر Watchdog

WDTCR- رجیستر کنترل تایمر Watchdog

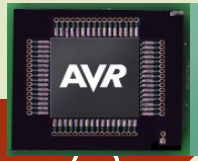
Bit	7	6	5	4	3	2	1	0	
	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **بیت های 0 تا 2 : WDP0,WDP1,WDP2** (watchdog timer prescaler)

توسط این سه بیت طبق جدول ۱-۲۷ می توان تقسیم فرکانسی را برای مدت زمان لازم جهت reset میکروکنترلر ،توسط watchdog تنظیم نمود.

WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 3.0V	Typical Time-out at V_{CC} = 5.0V
0	0	0	16K (16,384)	17.1 ms	16.3 ms
0	0	1	32K (32,768)	34.3 ms	32.5 ms
0	1	0	64K (65,536)	68.5 ms	65 ms
0	1	1	128K (131,072)	0.14 s	0.13 s
1	0	0	256K (262,144)	0.27 s	0.26 s
1	0	1	512K (524,288)	0.55 s	0.52 s
1	1	0	1,024K (1,048,576)	1.1 s	1.0 s
1	1	1	2,048K (2,097,152)	2.2 s	2.1 s

جدول ۱-۲۷ انتخاب تقسیم کنندهٔ فرکانسی کلاک تایمر نگهبان زمان (Watchdog)



تایمر Watchdog

- **بیت WDE-3** (watchdog enable)

با یک کردن این تایمر نگهبان زمان فعال می شود و برای غیر فعال کردن تایمر watchdog باید علاوه بر صفر کردن بیت WDE ، می بایست بیت WDTOE را نیز یک کرد.

روش غیر فعال کردن تایمر نگهبان زمان :

- نوشتن یک منطقی در بیت های WDTOE وWDE
- نوشتن صفر منطقی پس از چهار سیکل در بیت WDE

- **بیت WDTOE** (watchdog turn-off enabe)

زمانی که در بیت WDE صفر منطقی نوشته می شود باید این بیت یک شود و در غیر اینصورت تایمر نگهبان زمان نمی تواند غیر فعال شود.

با نوشتن یک منطقی در این بیت ،پس از چهار سیکل توسط سخت افزار به طور اتوماتیک صفر می شود.