

میکرو کامپیوتر ۱

جلسه ۵

مدرس : محمد جواد شاهسوندی

تایمرها در میکروکنترلر AVR – وقفه‌های تایمر میکروکنترلر AVR

تعریف وقفه

وقفه پیامی است که از طرف یک وسیله جانبی تولید می شود تا از CPU درخواست سرویس کند. با دریافت یک سیگنال وقفه کارهای جاری را متوقف کرده و به وسیله مورد نظر سرویس می دهد. درست شبیه زمانی که فردی در حال انجام کارهای روزمره است و با زنگ خوردن تلفن کارها را رها کرده تا به تلفن پاسخ دهد. در این مثال فرد مورد نظر نقش CPU، کارهای روزمره نقش برنامه اصلی، تلفن نقش وسیله جانبی و صدای زنگ آن نقش سیگنال وقفه را دارند.

تایمرها در میکروکنترلر AVR – وقفه‌های تایمر میکروکنترلر AVR

به دو روش می‌توان به وسایل جانبی سرویس دهی کرد :

(۱) سرکشی دوره ای یا polling :

سرکشی دوره ای به این صورت است که در خلال برنامه بیت های پرچم وسیله جانبی جهت آگاهی از وضعیت آن، توسط **cpu** بررسی می شود و اگر آن وسیله نیاز به سرویس دهی داشت این کار توسط **cpu** انجام می شود. در این روش **cpu** باید به طور مداوم به این وسایل سرکشی کند که این کار باعث کندی برنامه و از دست رفتن زمان برای بررسی وسایل دیگر می شود .

تایمرها در میکروکنترلر AVR – وقفه‌های تایمر میکروکنترلر AVR

به دو روش می‌توان به وسایل جانبی سرویس دهی کرد :

۲) استفاده از وقفه یا **interrupt** : در روش وقفه، **cpu** در حال اجرای روال عادی برنامه است. اما زمانی که یک وسیله جانبی نیاز به توجه **cpu** دارد با ارسال سیگنال وقفه باعث می‌شود تا **cpu** روال عادی برنامه را متوقف کرده و به آن توجه کند. این روش باعث می‌شود تا زمان **cpu** برای بررسی رخ دادن یک وضعیت تلف نشود.

تایمرها در میکروکنترلر AVR – انواع وقفه‌ها در AVR

معمولاً در میکروکنترلرها وقفه‌ها به دو دسته کلی تقسیم می‌شوند :

وقفه‌های خارجی : وقفه‌هایی هستند که منبع آنها خارج از میکرو قرار دارد و با ارسال سیگنالی به

پایه‌های مخصوصی در میکروکنترلر، وقفه ایجاد می‌کنند

وقفه‌های داخلی : همانطور که از اسمش پیداست این وقفه‌ها سیگنال‌هایی هستند که داخل میکرو

ایجاد می‌شوند. اکثر این سیگنال‌ها از وسایل جانبی خود میکرو مانند تایمر/کانترها ، ADC و ...

تولید می‌شوند. این سیگنال‌ها همان درخواست سرویس از CPU می‌باشند .

تایمرها در میکروکنترلر AVR – بردار وقفه

هر وقفهای در AVR خانه ای را در (ROM حافظه فلش) به خود اختصاص می دهد. زمانی که وقفه ایجاد می شود بسته به نوع وقفه، CPU بعد از اجراکردن دستور جاری به آن خانه مراجعه می کند. این خانه محتوی آدرس برنامههای است که به وقفه سرویس می دهد. به مجموعه این خانه ها که هرکدام مربوط به یک نوع وقفه هستند جدول بردار وقفه و به برنامه ای که به وقفه سرویس می دهد روال سرویس وقفه (ISR) میگویند. در جدول صفحه بعد می توانید آدرس وقفه های مختلف را مشاهده کنید)

تایمرها در میکروکنترلر AVR – بردار وقفه

آدرس یا وکتور روتین سرویس وقفه				دستگاه تقاضاکننده وقفه
میکروکنترلر ATtiny AVR	میکروکنترلر ATmega32 AVR	میکروکنترلر ATmega16 AVR	میکروکنترلر ATmega8 AVR	
0x000	0x000	0x000	0x000	Reset
0x001	0x002	0x002	0x001	وقفه خارجی شماره 0 در پایه INT0
—	0x004	0x004	0x002	وقفه خارجی شماره 1 در پایه INT1
—	0x006	0x024	—	وقفه خارجی شماره 2 در پایه INT2
—	0x008	0x006	0x003	تایمر 2 در Compare Match
—	0x00A	0x008	0x004	تایمر 2 در سرریز Overflow
—	0x00C	0x00A	0x005	تایمر 1 در input capture
0x003	0x00E	0x00C	0x006	تایمر 1 در Compare Match A
—	0x010	0x00E	0x007	تایمر 1 در Compare Match B
0x004	0x012	0x010	0x008	تایمر 1 در سرریز یا Overflow

تایمرها در میکروکنترلر AVR – بردار وقفه

0x005	0x016	0x012	0x009	تایمر 0 در سرریز یا Overflow
...	0x014	0x026	...	تایمر 0* در Compare Match
...	0x01A	0x016	0x00B	پورت سری UART برای دریافت کامل یک بایت (RXC)
...	...	...	...	---
0x008	0x020	0x01C	0x00E	مبدل آنالوگ به دیجیتال ADC
...	...	...	...	---
0x007	0x024	0x020	0x010	مقایسه کننده آنالوگ
...	...	...	...	---

تایمرها در میکروکنترلر AVR – فعال کردن پاسخگویی به وقفه‌ها

وقفه‌ها در AVR به صورت پیش فرض غیر فعال هستند. یعنی با روشن شدن میکرو هیچ وقفه‌ای کار نمی‌کند و باید به وسیله برنامه نویسی، وقفه‌هایی را که می‌خواهیم ایجاد شوند فعال کنیم. در AVR معمولاً هر وسیله‌ای به طور اختصاصی دارای بیتی به نام بیت فعال ساز وقفه می‌باشد. با یک شدن این بیت اجازه پاسخ‌گویی به وقفه‌ی مربوط به آن وسیله فعال و با صفر شدن بیت مذکور اجازه پاسخ‌گویی به وقفه غیر فعال می‌شود. مکان این بیت برای هر وقفه‌ای در ثبات خاصی از ثبات‌های I/O/امی باشد.

تایمرها در میکروکنترلر AVR – فعال کردن پاسخگویی به وقفه‌ها

یکی از بیت های ثابت وضعیت “یا فعال ساز وقفه‌ی عمومی (Interrupt Enable) نام دارد. به وسیله‌ی این بیت می توان کلیه وقفه‌ها را فعال و یا غیرفعال کرد. به این صورت که اگر این بیت را یک کنیم وقفه‌ها فعال و اگر آن را صفر کنیم وقفه‌ها غیرفعال می شوند. با صفر شدن این بیت حتی اگر بیت فعال ساز وقفه برای وسیله ای یک باشد، وقفه ایجاد نمی شود. پس برای رخ دادن یک وقفه لازم است بیت فعال ساز وقفه‌ی عمومی و بیت فعال ساز وقفه‌ی اختصاصی، همزمان یک باشند. درواقع بیت فعال ساز وقفه‌ی عمومی با بیت های فعال ساز وقفه‌های اختصاصی AND شده است. در شکل زیر می توانید ثابت یا ثابت وضعیت را مشاهده کنید.

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
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	

تایمرها در میکروکنترلر AVR – فعال کردن پاسخگویی به وقفه‌ها

در کد زیر می‌توانید نحوه فعال و غیر فعال کردن بیت ادر کدویژن را مشاهده کنید :

```
1 #asm("sei")          /*فعال کردن وقفه عمومی*/  
2  
3 #asm("cli")           /*غیر فعال کردن وقفه عمومی*/
```

تایمرها در میکروکنترلر AVR – فعال کردن پاسخگویی به وقفه‌ها

مراحل اجرای یک وقفه:

- ۱- هنگامی که یک وقفه اتفاق می افتد **cpu** دستور در حال اجرا را به پایان برده و آدرس دستور بعدی (این آدرس درون ثبات **pc** یا **program counter** قرار دارد) را بر روی پشته ذخیره می کند .
- ۲- بسته به نوع وقفه، میکروکنترلر به خانه ثابتی از حافظه فلشواچ در جدول بردار وقفه پرش می کند .
- ۳- در این خانه، آدرس روال سرویس وقفه یا **ISR** وجود دارد و میکرو با پرش به این آدرس، شروع به اجرای دستورات موجود در روال سرویس وقفه می کند تا به دستور آخر روال یعنی **RETI** برسد .
- ۴- این دستور شمارنده برنامه یا **pc** را با آدرس دستوری که در مرحله یک در پشته ذخیره شده بود بارگذاری می کند. این کار با برداشتن دو بایت بالای پشته و قرار دادن آن در **PC** انجام می شود. سپس میکرو شروع به اجرای دستورات از این آدرس می کند .

تایمرها در میکروکنترلر AVR – فعال کردن پاسخگویی به وقفه‌ها

این موضوع اهمیت مدیریت پشته را در حین نوشتن روال سرویس وقفه برای ما آشکار می‌کند. بهتر است در ابتدای هر روال سرویس وقفه‌های محتویات ثبات‌های همه منظوره‌ای را که می‌خواهیم در این روال استفاده کنیم به وسیله دستور **push** در پشته ذخیره شده و در آخر روال به وسیله دستور **pop** از پشته بازیابی شود. به این کار ذخیره سازی محتوا گویند. این مورد به طور کلی در مورد پورت‌ها و ثبات‌هایی که نیاز است محتوایشان ذخیره شود صدق می‌کند. می‌توان به جای پشته از هر فضایی در **RAM** برای این کار استفاده کرد.

تایمرها در میکروکنترلر AVR – فعال کردن پاسخگویی به وقفه‌ها

روال سرویس وقفه برنامه ای است که توسط برنامه نویس نوشته می شود تا عملیاتی را با ایجاد وقفه

انجام دهد. اما چرا این برنامه را در آدرسی که مربوط به بردار وقفه است نمی نویسیم؟

آدرس های بردار وقفه فقط ۱ بایت فضا دارند. پس باید ISR را در محل دیگری نوشت و آدرس آن را در محل بردار وقفه قرار داد .

مواردی که در مورد پشته و دستورات `push` و `pop` نوشتن ISR در آدرسی دیگر گفته شد مخصوص

زبان اسمبلی می باشد. اما در زبان C کامپایلر این کارها را مدیریت می کند و نیازی به نگرانی برنامه

نویس وجود ندارد. تنها کافی است یک تابع برای روال وقفه نوشته شود .

تایمرها در میکروکنترلر AVR – فعال کردن پاسخگویی به وقفه‌ها

شاید بپرسید در صورتی که در طول پردازش روال سرویس وقفه‌ای، وقفه‌ی دیگری رخ دهد چه اتفاقی می‌افتد؟ در جواب این سوال باید بگوییم که این بستگی به برنامه نویسی و دستورات موجود در روال سرویس وقفه جاری دارد. همان طور که گفتم در صورت ایجاد وقفه، AVR به طور اتوماتیک بیت فعال ساز وقفه‌ی عمومی را پاک می‌کند. اگر ما بخواهیم در حین سرویس دهی به یک وقفه، وقفه‌ی دیگری در صورت ایجاد سرویس دهی شود باید در روال سرویس وقفه‌ی قبلی بیت فعال ساز وقفه عمومی را یک کنیم. اما باید چنین کاری با احتیاط انجام شود چون ممکن است باعث گیر کردن برنامه درون یک وقفه شود. مثلاً اگر یک وقفه‌ی خارجی حساس به سطح فعال باشد و بیت ادر حالی که هنوز پایه وقفه در حالت تحریک است، یک شود باعث می‌شود تا به طور بی‌نهایت وارد ISR شده و سرریز پشته و اتفاقات غیر قابل پیش بینی رخ دهد. در ادامه درباره وقفه‌ی خارجی و حساس به

سطح صحبت می‌کنیم

تایمرها در میکروکنترلر AVR – فعال کردن پاسخگویی به وقفه‌ها

لازم به ذکر است که بیت وقفه‌ی اختصاصی یک وسیله جانبی در صورت درخواست وقفه توسط آن وسیله و پرش به جدول بردار وقفه برخلاف بیت فعال ساز وقفه‌ی عمومی هیچ تغییری نمی‌کند.

تایمرها در میکروکنترلر AVR – اولویت وقفه

ممکن است این سوال در ذهن شما ایجاد شود که اگر دو وقفه به طور هم زمان رخ دهند به کدام وقفه پاسخ داده می شود؟ در جواب باید بگوییم که در جدول بردار وقفه، هر وقفهای که آدرس کمتری دارد) هر چه در جدول بالاتر باشد) اولویت بیشتری هم در پاسخگویی خواهد داشت. پس در صورت وقوع چنین حالتی آن وقفهای که اولویت بالاتری دارد پاسخ داده خواهد شد.

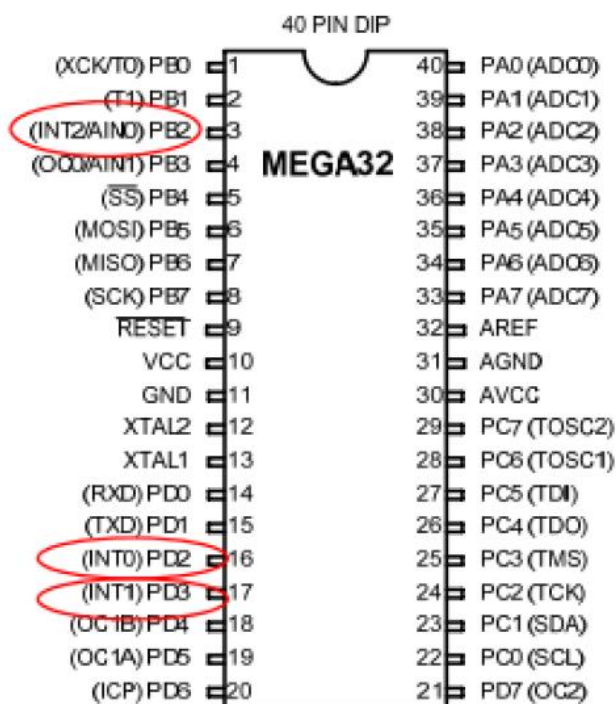
تایمرها در میکروکنترلر AVR – وقفه‌های خارجی

بعد از توضیح موارد کلی و اصول وقفه‌ها حال وقفه‌های خارجی را به صورت خاص بررسی می‌کنیم.

یک وقفه خارجی در اثر سیگنالی از یک وسیله جانبی خارج از میکرو که وارد پایه خاصی از میکرو

می‌شود اتفاق می‌افتد. برای مثال در میکروکنترلر ATMEGA32 سه پایه برای وقفه خارجی با نام

های INT0, INT1, INT2 وجود دارد که در شکل زیر مشخص شده است :



تایمرها در میکروکنترلر AVR – وقفه‌های خارجی

هر پایه در میکرو ممکن است برای چند کاربرد طراحی شده باشد که می‌توان در یک زمان از یک قابلیت آن طبق نیاز استفاده کرد. از سه پایه ای که برای سه وقفه خارجی استفاده می‌شوند به عنوان ورودی و خروجی معمولی هم می‌توان استفاده کرد. برای اینکه بتوانیم از هر کدام از این پایه ها به عنوان وقفه خارجی استفاده کنیم باید در ثباتهای مربوطه تنظیمات خاصی را انجام دهیم که در ادامه توضیح داده می‌شود

تایمرها در میکروکنترلر AVR – ثبات GICR

نام این ثبات از **Global interrupt control register 1** و ۹ که در تصویر زیر قابل مشاهده ، گرفته شده است. با استفاده از بیت های شماره ۵ است می تواند قابلیت پاسخ گویی به وقفه های خارجی ۸ تا ۱ را فعال کرد. **INT0** برای وقفه خارجی صفر استفاده می شود و سایر وقفه های خارجی نیز به همین ترتیب فعال می شوند. البته باید توسط ثبات **SREG** بیت فعال ساز وقفه عمومی فعال شود و توسط ثبات **MCUCR** حساسیت به لبه یا سطح مشخص شود که در ادامه توضیح داده می شود . بیت های ۱ تا ۴ رزرو هستند و از آن ها استفاده ای نمی شود. بیت ۸ و ۱ نیز برای تغییر مکان بردارهای وقفه مورد استفاده قرار می گیرد که خارج از بحث جلسه امروز است .

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

شکل ۱۲: ثبات GICR

تایمرها در میکروکنترلر AVR – ثبات GICR

نام این ثبات از **Global interrupt control register 1** و ۹ که در تصویر زیر قابل مشاهده ، گرفته شده است. با استفاده از بیت های شماره ۵ است می تواند قابلیت پاسخ گویی به وقفه های خارجی ۸ تا ۱ را فعال کرد. **INT0** برای وقفه خارجی صفر استفاده می شود و سایر وقفه های خارجی نیز به همین ترتیب فعال می شوند. البته باید توسط ثبات **SREG** بیت فعال ساز وقفه عمومی فعال شود و توسط ثبات **MCUCR** حساسیت به لبه یا سطح مشخص شود که در ادامه توضیح داده می شود . بیت های ۱ تا ۴ رزرو هستند و از آن ها استفاده ای نمی شود. بیت ۸ و ۱ نیز برای تغییر مکان بردارهای وقفه مورد استفاده قرار می گیرد که خارج از بحث جلسه امروز است .

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

شکل ۱۲: ثبات GICR

تایمرها در میکروکنترلر AVR – ثبات MCUCR

نام این ثبات از **MCU Control register** گرفته شده است. همان طور که گفتیم، وقفه های سخت افزاری یا حساس به سطح هستند یا حساس به لبه. با تنظیم بیت های ثبات **MCUCR** می توان نوع حساسیت را تعیین کرد. برای **INT0 1** و برای **INT1** از بیت های ۱ و ۳ استفاده می شود. در تصویر زیر این ثبات را مشاهده می کنید:

Bit	7	6	5	4	3	2	1	0	
	SE	SM2	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	

شکل ۱۳: ثبات MCUCR

تایمرها در میکروکنترلر AVR – ثبات MCUCR

در تصویر زیر نحوه تنظیم حساسیت به سطح یا لبه برای INT0 توضیح داده شده است :

ISC01	ISC00		Description
0	0		The low level of INT0 generates an interrupt request.
0	1		Any logical change on INT0 generates an interrupt request.
1	0		The falling edge of INT0 generates an interrupt request.
1	1		The rising edge of INT0 generates an interrupt request.

شکل ۱۴: تنظیم حساسیت برای INT0

تایمرها در میکروکنترلر AVR – ثبات MCUCSR

نام این ثبات از **MCU Control and Status Register** گرفته شده است.

نحوه حساسیت وقفه‌ی خارجی دو یا **INT2** در **ATMEGA32** توسط بیت **ISC2** در ثبات **MCUCSR**

تنظیم می‌شود. وقفه‌ی خارجی دو فقط در دو حالت تنظیم می‌شود.

اگر **ISC2** صفر باشد **INT2** حساس به لبه پایین رونده می‌شود

اگر **ISC2** یک باشد **INT2** حساس به لبه بالا رونده می‌شود

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					

شکل ۱۵: ثبات **MCUCSR**

تایمرها در میکروکنترلر AVR – ثبات GIFR

نام این ثبات از **Global interrupt flag register** گرفته شده است. این ثبات پرچم وقوع وقفه‌های خارجی است. هر وقت یک وقفه خارجی رخ دهد، پرچم متناظر با آن یک می شود. همان طور که در شکل زیر می بینید، پرچم متناظر با وقفه خارجی صفر **INTF0** است و برای وقفه‌های یک و دو نیز به همین ترتیب. اما نکته ی مهمی که در رابطه با این پرچم ها وجود دارد صفر شدن آن ها است. اگر پاسخ گویی به وقفه را فعال کرده باشیم، با وقوع وقفه و پرش به آدرس بردار وقفه، پرچم متناظر هم صفر می شود. اما اگر به صورت نرم افزاری بخواهیم آن را صفر کنیم باید روی آن ۱ نوشته شود. این قانون برای تمامی پرچم های **AVR** صادق است.

Bit	7	6	5	4	3	2	1	0	
	INTF1	INTF0	INTF2	–	–	–	–	–	GIFR
Read/Write	R/W	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

شکل ۱۶: ثبات GIFR

تایمرها در میکروکنترلر AVR – تایمر یک

تایمر یک نیز مانند تایمر صفر دارای ثباتهایی است که در ادامه ی مطلب با آن ها آشنا می شویم. لازم به ذکر است که در این جا فقط ثباتهایی را معرفی خواهیم کرد که به آن ها نیاز پیدا می کنیم. در مورد ثباتهای دیگر بعدا صحبت خواهد شد

تایمرها در میکروکنترلر AVR – ثبات TCCR1B

ثبات کنترل تایمر/کانتریک (Timer/Counter Control Register B) به صورت زیر است :

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

شکل ۲۳: ثبات TCCR1B

تایمرها در میکروکنترلر AVR – ثبات TCCR1B

در حال حاضر فقط بیت هایی که هایلایت شده اند مورد استفاده ی ما خواهند بود. بیت های ۸ تا ۱ این ثبات که با نام های CS11، CS10 و CS12 شناخته می شوند، بیت های انتخاب ساعت تایمر ۱ هستند. جدول زیر نشان می دهد که به ازای حالت های مختلف این سه بیت، وضعیت ساعت تایمر به چه صورت خواهد بود.

CS12:CS10	D2D1D0	Timer1 clock selector
	0 0 0	No clock source (Timer/Counter stopped)
	0 0 1	clk (no prescaling)
	0 1 0	clk / 8
	0 1 1	clk / 64
	1 0 0	clk / 256
	1 0 1	clk / 1024
	1 1 0	External clock source on T1 pin. Clock on falling edge.
	1 1 1	External clock source on T1 pin. Clock on rising edge.

شکل ۲۴: وضعیت ساعت تایمر ۱

تایمرها در میکروکنترلر AVR – ثبات TCNT1

ثبات تایمر/کانتریک در شکل زیر نشان داده شده است :

Bit	7	6	5	4	3	2	1	0	
	TCNT1[15:8]								TCNT1H
	TCNT1[7:0]								TCNT1L
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	

همان طور که در شکل نیز مشاهده می کنید، این ثبات ۱۱ بیتی است، که متشکل از دو ثبات هشت بیتی TCNT1H و TCNT1L است. TCNT1H بایت بالایی و TCNT1L بایت پایینی این ثبات را نشان می دهند. مقدار تایمر/کانتریک در این ثبات ذخیره می شود

تایمرها در میکروکنترلر AVR – ثبات TIMSK

ثبات وقفه ی تایمر/کانتر مانند شکل زیر است :

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
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	

شکل ۲۶: ثبات TIMSK

این ثبات قبلا نیز معرفی گردید چرا که بین هر سه تایمر صفر، یک و دو مشترک است. در شکل بالا بیت هایی که مربوط به تایمر ۱ نیستند، خاکستری شده اند. بیت های ۱ تا ۵ این ثبات با تایمر یک مرتبط هستند. در این جا از بیت TOIE1 یا بیت فعال کردن وقفه ی سرریز تایمر ۱ (Timer/Counter Overflow Interrupt Enable) استفاده می کنیم. در مورد دیگر بیت های این تایمر که مربوط به مد CTC هستند، در بعدا صحبت خواهیم کرد .

تایمرها در میکروکنترلر AVR – ثبات TIFR

ثبات پرچم وقفه ی تایمر/کانتر در شکل زیر نشان داده شده است :

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
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	

شکل ۲۷: ثبات TIFR

این ثبات نیز در بین هر سه تایمر AVR مشترک است. در شکل بالا بیت هایی که به رنگ خاکستری هستند مربوط به تایمرهای دیگر هستند. فقط بیت های ۱ تا ۵ مربوط به تایمر ۱ می باشند. از بین این بیت ها با بیت TOV1 کار داریم که بیت پرچم سرریز تایمر/کانتر ۱ (Timer/Counter Overflow Flag) می باشد. وقتی که تایمر سرریز می کند این بیت یک می شود و به محض این که روال وقفه اجرا شود به طور خودکار دوباره صفر می گردد. ولی هنگامی که روالی برای وقفه وجود نداشته باشد، با نوشتن یک بر روی TOV1 آن را صفر می کنیم