

## آزمایش ۶ – آشنایی با تایمر و کانتر در میکروکنترلر AVR

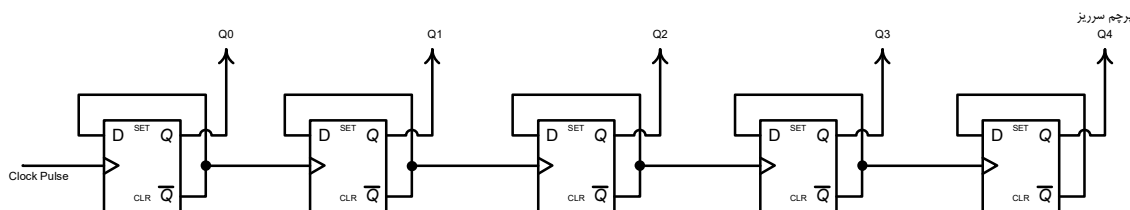
### ۶-۱ مقدمه

تایمر/کانتر یکی از بخش های مهم میکروکنترلرها می باشد. در بیشتر مواقع لازم که تعدادی وقایع خارجی (با سرعت بالا) شمارش شود و یا گاهی لازم است که در یک زمان خاص و دقیق، کاری صورت گیرد. تنها توسط تایمر/کانتر ها می توان این کارهای دقیق و با سرعت بالا را انجام داد.

میکروکنترلرهای AVR حداکثر دارای شش عدد تایمر کانتر هشت بیتی و شانزده بیتی هستند. برخی از آنها دارای عملکرد ساده و برخی دیگر دارای امکانات بیشتر نظیر تولید موج PWM، حالت مقایسه CTC، حالت تسخیر، عملکرد غیر همزمان و ... می باشند.

### ۶-۲ تئوری عملکرد تایمر/کانتر

تایمر و کانترها از تعدادی فلیپ فلاپ که به صورت پشت سر هم قرار گرفته و به صورت یک شمارنده آسنکرون عمل می کنند، تشکیل شده اند (شکل ۶-۱).



شکل ۶-۱: شمارنده آسنکرون

در این شمارنده پایه های Q3-Q0 به عنوان خروجی استفاده می شود. با فرض وارد نمودن مقدار 0000 و با اعمال پالس ساعت، شمارنده شروع به شمارش می نماید و حداکثر تا مقدار 1111 بالا می رود. پس از این حالت و با اعمال پالس ساعت بعدی، بیت سرریز (Q4) فعال می شود که نشان دهنده این است که شمارنده به حداکثر مقدار خود رسیده است.

هنگامی که پالس ساعت اعمالی به این شمارنده یک مقدار مشخص داشته باشد، می توان با در نظر گرفتن مدت زمان شمارش از مقدار xxxx تا 1111 و سپس سرریز شدن شمارنده، زمان های معینی را ایجاد نمود. در این حالت شمارنده به صورت تایمر عمل می کند. به عنوان نمونه برای شکل ۱ اگر پالس ساعت اعمالی دارای فرکانس ۲ Hz باشد و شمارنده نیز با مقدار پیش فرض 0000 در نظر گرفته شود، حداکثر ۸ ثانیه طول می کشد تا شمارنده سرریز شود.

هنگامی که منبع پالس ساعت اعمالی از یک منبع منظم و معین نباشد (منبع خارجی)، در این صورت می توان از شمارنده برای شمارش وقایع خارجی نظیر عبور تعداد قطعات از جلوی یک سنسور نوری استفاده نمود که در این حالت شمارنده به صورت کانتر به کار گرفته می شود.

### ۳-۶ تایمر/کانتر صفر

تایمر کانتر صفر در AVR ها را می توان در سه نوع عملکرد زیر دسته بندی کرد :

- ۱- نوع عملکرد ساده هشت بیتی : این مدل فقط در سری ATTiny , AT90S به کار گرفته شده است.
- ۲- نوع عملکرد پیشرفته هشت بیتی : این مدل در سری های ATmega (به غیر از ATmega8 و ATmega163 که از مدل ساده ۸ بیتی استفاده می کنند) به کار گرفته شده است.
- ۳- نوع عملکرد پیشرفته شانزده بیتی : این مدل فقط در AVR های سری ATTiny2313 و ATTiny13 به کار گرفته است.

### ۴-۶ تایمر کانتر صفر در حالت هشت بیتی پیشرفته

حالت عملکرد هشت بیتی در سری های ATmega (به غیر از ATmega8 و ATmega163) قرار دارد. از خصوصیات تایمر/کانتر در این مدل می توان به موارد زیر اشاره کرد :

- ۱- تایمر / کانتر در حالت عادی
- ۲- تایمر / کانتر در حالت مقایسه CTC
- ۳- تایمر / کانتر در حالت PWM سریع (تک شیب)
- ۴- تایمر / کانتر در حالت PWM تصحیح فاز (دو شیب)

### ۵-۶ رجیستر های تایمر/کانتر صفر در حالت عملکرد هشت بیتی پیشرفته

#### ۱-۵-۶ رجیستر مقایسه خروجی (OCR0)

این رجیستر هشت بیتی، خواندنی و نوشتنی بوده و به طور مستقیم با مقدار شمارنده TCNT0 مقایسه می شود. از تطابق این دو برای تولید وقفه خروجی یا تولید یک شکل موج روی پایه OC0 می توان استفاده نمود.

| Bit           | 7          | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|---------------|------------|-----|-----|-----|-----|-----|-----|-----|
|               | OCR0 [7:0] |     |     |     |     |     |     |     |
| 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   |

#### ۲-۵-۶ رجیستر تایمر کانتر صفر TCNT0

این رجیستر هشت بیتی امکان دسترسی مستقیم برای خواندن و نوشتن در شمارنده را فراهم می کند. این رجیستر، هم خواندنی است و هم نوشتنی. به هنگام خواندن، مقدار شمارش شده را از خروجی شمارنده برمی گرداند و به هنگام نوشتن، مقدار جدید را به ورودی شمارنده انتقال می دهد.

| Bit           | 7           | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|---------------|-------------|-----|-----|-----|-----|-----|-----|-----|
|               | TCNT0 [7:0] |     |     |     |     |     |     |     |
| 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   |

#### ۳-۵-۶ رجیستر کنترلی تایمر/کانتر صفر TCCR0

این رجیستر دارای هشت بیت کنترلی برای انتخاب پالس ساعت، حالت خروجی هنگام تطابق مقایسه، عملکردهای PWM، و بیت مقایسه خروجی می باشد.

با استفاده از رجیستر TCCR0 می توان فرکانس پالس ساعت تایمر را انتخاب کرد که برای این کار از بیت های انتخاب پالس ساعت (CS00, CS01, CS02) استفاده می شود.

| Bit           | 7    | 6     | 5     | 4     | 3     | 2    | 1    | 0    |
|---------------|------|-------|-------|-------|-------|------|------|------|
|               | FOC0 | WGM00 | COM01 | COM00 | WGM01 | CS02 | CS01 | CS00 |
| 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    |

#### – بیت های انتخاب پالس ساعت (CS00, CS01 و CS02)

این سه بیت برای انتخاب فرکانس پالس ساعت تایمر صفر مورد استفاده قرار می گیرند. جدول ۶-۱ و جدول ۶-۲ انتخاب پالس ساعت برای میکروکنترلرهای ATmega128, ATmega64 و ATmega32 را نشان می دهد. تایمر در زمانی که هیچ منبع پالس ساعتی انتخاب نشود، غیرفعال است.

جدول ۶-۱: انتخاب پالس ساعت برای میکروکنترلرهای ATmega64 و ATmega128

| CS02 | CS01 | CS00 | Description                               |
|------|------|------|-------------------------------------------|
| 0    | 0    | 0    | No Clock Source (Timer/Counter Stopped)   |
| 0    | 0    | 1    | CLK <sub>I/O</sub> (No Prescaling)        |
| 0    | 1    | 0    | CLK <sub>I/O</sub> /8 (From Prescaler)    |
| 0    | 1    | 1    | CLK <sub>I/O</sub> /32 (From Prescaler)   |
| 1    | 0    | 0    | CLK <sub>I/O</sub> /64 (From Prescaler)   |
| 1    | 0    | 1    | CLK <sub>I/O</sub> /128 (From Prescaler)  |
| 1    | 1    | 0    | CLK <sub>I/O</sub> /256 (From Prescaler)  |
| 1    | 1    | 1    | CLK <sub>I/O</sub> /1024 (From Prescaler) |

جدول ۶-۲: انتخاب پالس ساعت برای میکروکنترلرهای ATmega32

| CS02 | CS01 | CS00 | Description                                             |
|------|------|------|---------------------------------------------------------|
| 0    | 0    | 0    | No Clock Source (Timer/Counter Stopped)                 |
| 0    | 0    | 1    | CLK <sub>I/O</sub> (No Prescaling)                      |
| 0    | 1    | 0    | CLK <sub>I/O</sub> /8 (From Prescaler)                  |
| 0    | 1    | 1    | CLK <sub>I/O</sub> /64 (From Prescaler)                 |
| 1    | 0    | 0    | CLK <sub>I/O</sub> /256 (From Prescaler)                |
| 1    | 0    | 1    | CLK <sub>I/O</sub> /1024 (From Prescaler)               |
| 1    | 1    | 0    | External clock source on T0 pin. Clock on falling edge. |
| 1    | 1    | 1    | External clock source on T0 pin. Clock on rising edge.  |

#### – بیت های حالت خروجی تطابق مقایسه (COM01:0)

این بیت ها رفتار پایه مقایسه خروجی (OC0) را کنترل می کنند. زمانی که این بیت ها هر دو صفر باشند، پایه OC0 که یکی از پایه های یک Port است، به صورت I/O معمولی استفاده می شود. اما اگر یکی یا هر دوی این بیت ها یک شوند، پایه OC0 به خروجی واحد تولید شکل موج متصل خواهد شد و دیگر نمی توان از آن به عنوان I/O معمولی استفاده نمود (جدول ۴).

#### – بیت های WGM01 و WGM00

توسط این دو بیت می توان حالت های عملکرد پشتیبانی شده تایمر/کانتر را عبارتند از : عملکرد عادی، عملکرد مقایسه ای، عملکرد PWM سریع و عملکرد با PWM تصحیح فاز را انتخاب نمود (جدول ۶-۳).

جدول ۶-۳: انتخاب مد تولید موج

| Mode | WGM01 (CTC0) | WGM00 (PWM0) | Timer/Counter Mode of Operation | TOP  | Update of OCR0 at | TOV0 Flag Set on |
|------|--------------|--------------|---------------------------------|------|-------------------|------------------|
| 0    | 0            | 0            | Normal                          | 0xFF | Immediate         | MAX              |
| 1    | 0            | 1            | PWM, Phase Correct              | 0xFF | TOP               | BOTTOM           |
| 2    | 1            | 0            | CTC                             | OCR0 | Immediate         | TOP              |
| 3    | 1            | 1            | Fast PWM                        | 0xFF | TOP               | TOP              |

### – بیت مقایسه خروجی (FOC0)

این بیت زمانی فعال می شود که بیت های WGM، یک حالت غیر PWM (عملکرد عادی و مقایسه) را انتخاب کرده باشند. در هنگام نوشتن یک منطقی در این بیت، یک تطابق مقایسه فوری روی واحد تولید شکل موج رخ می دهد و پایه OC0 را با توجه به تنظیم بیت های COM01 و COM00 تغییر می دهد.

در زمان یک کردن FOC0 پرچم مقایسه خروجی (OCF0) فعال خواهد شد و وقفه ای نیز تولید نمی شود. این بیت در کل، تأثیری بر هیچ رجیستر و پرچمی نمی گذارد و فقط وضعیت پایه OC0 را با توجه به تنظیمات آن تغییر می دهد.

### ۶-۵-۴ رجیستر پرچم وقفه تایمر کانتر صفر TIFR

از این رجیستر برای تشخیص یک سرریز یا تطابق در مقایسه در تایمر/کانتر صفر رخ داده باشد.

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

### – بیت پرچم سرریز تایمر/کانتر صفر (TOV0)

بیت TOV0 زمانی فعال (یک) می شود که یک سرریز در تایمر/کانتر صفر به وقع بپیوندد. برای پاک کردن این بیت باید در آن یک نوشت. البته در زمان اجرای بردار وقفه، TOV0 توسط سخت افزار پاک می شود. در حالت PWM تصحیح فاز، TOV0 زمانی فعال می شود که جهت شمارش در 00\$ عوض شود.

### – بیت پرچم مقایسه خروجی صفر (OCF0)

بیت OCR0 زمانی فعال (یک) می شود که یک تطابق مقایسه مابین تایمر/کانتر صفر (TCNT0) و داده درون رجیستر OCR0 به وقوع بپیوندد. برای پاک کردن این بیت باید در آن یک نوشت. البته در زمان اجرای بردار وقفه، OCR0 توسط سخت افزار پاک می شود.

### ۶-۵-۵ رجیستر پوشش وقفه تایمر/کانتر صفر (TIMSK)

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

### – بیت فعالساز وقفه سرریز تایمر/کانتر صفر (TOIE0)

زمانی که بیت TOIE0 یک شود و همچنین بیت I در SREG نیز فعال باشد وقفه سرریز تایمر/کانتر صفر فعال می شود. هنگامی که بیت TOV0 در رجیستر پرچم وقفه (TIFR) فعال شود، یک وقفه درخواست خواهد شد.

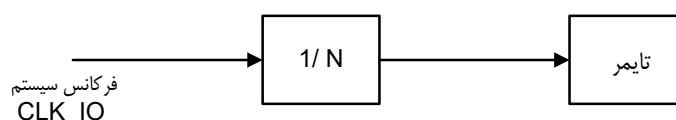
### – بیت فعالساز وقفه تطابق مقایسه خروجی تایمر/کانتر صفر

وقتی OCIE0 یک شده و بیت I (فعالساز وقفه سراسری) در SREG نیز فعال باشد، وقفه تطابق مقایسه تایمر/کانتر صفر فعال می شود. هنگامی که بیت OCF0 در رجیستر پرچم وقفه (TIFR) فعال شود، یک وقفه درخواست خواهد شد.

## ۶-۶ نحوه محاسبه زمان بندی برای تایمرها

برای محاسبه زمان بندی مورد نظر در تایمرها باید مراحل زیر انجام پذیرد :

- ۱- فرکانس داخلی پالس ساعت سیستم را در نظر گرفت.
- ۲- ضریب تقسیم پالس ساعت سیستم برای تولید پالس ساعت تایمر توسط بیت های CS00، CS01 و CS02 را تعیین نمود (شکل ۶-۲).



شکل ۶-۲: تعیین ضریب تقسیم پالس ساعت تایمر

۳- مقدار مطلوب را رجیستر TCNTx قرار داد (x شماره تایمر مورد نظر است).

مثال ۱: با فرض نوسان ساز داخلی ۱ MHz و ضریب تقسیم  $N=32$  و  $TCNT0=0AH$  مقدار زمان تا فعال شدن پرچم سرریز را محاسبه کنید.

$$\text{فرکانس پالس ساعت تایمر} = \frac{1 \text{ MHz}}{32} = 31.25 \text{ KHz}$$

$$\text{مدت زمان یک شمارش} = \frac{1}{31.25 \text{ KHz}} = 32 \mu s$$

$$256 - TCNT0 = 256 - 10 = 246 = \text{تعداد پالس برای سرریز شدن تایمر}$$

$$\text{مدت زمان شمارش تایمر} = 246 \times 32 \mu s = 7.872 \text{ ms}$$

مثال ۲: با فرض استفاده از کریستال خارجی ۱۶ MHz برای تولید مدت زمان ۱۰ ms چه عددی را باید در تایمر یک (TCNT1) قرار داد ؟

با فرض انتخاب ضریب تقسیم ۶۴ ( $N=64$ ) داریم :

$$\text{فرکانس پالس ساعت تایمر} = \frac{16 \text{ MHz}}{64} = 250 \text{ KHz}$$

$$\text{مدت زمان یک شمارش} = \frac{1}{250 \text{ KHz}} = 4 \mu s$$

$$\text{تعداد پالس لازم برای مدت ۱۰ ms} = \frac{10 \text{ ms}}{4 \mu s} = 2500$$

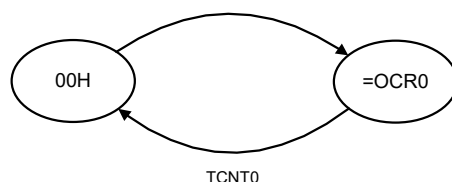
$$65536 - 2500 = 63036 = F63CH = \text{عدد لازم برای TCNT1}$$

## ۶=۷ تایمر/کانتر صفر در حالت عادی

ساده ترین حالت عملکرد تایمر/کانتر، حالت عادی می باشد. جهت شمارش، رو به بالا (افزایشی) بوده و پس از رسیدن مقدار رجیستر TCNT0 به TOP دوباره از مقدار BOTTOM شروع به شمارش نموده و پرچم سرریز TOV0 فعال خواهد شد. مقدار TOP برابر حداکثر مقدار شمارش در یک چرخه است و BOTTOM مقداری است که با رسیدن شمارنده به آن، شمارش صفر می شود.

## ۶-۸ عملکرد تایمر/کانتر صفر در حالت مقایسه (CTC)

در حالت مقایسه، رجیستر TCNT0 به طور دایم با رجیستر OCR0 مقایسه و در صورت تطابق، رجیستر TCNT0 برابر با صفر می شود (شکل ۶-۳).



شکل ۶-۳: نمودار شمارش تایمر/کانتر صفر در حالت CTC

از نتیجه مقایسه می توان برای تولید شکل موج روی پایه مقایسه خروجی OC0 استفاده کرد (قسمت اول آزمایش). در هنگام تطبیق مقایسه در صورت فعال بودن وقفه و تولید پرچم OC0 می توان یک وقفه مقایسه را ایجاد نمود و از روال وقفه برای به روز رسانی مقدار رجیستر OCR0 (TOP) استفاده نمود. به هر حال تغییر رجیستر OC0 (TOP) به یک مقدار جدید در زمانی که تایمر در حال شمارش است باید با احتیاط انجام شود، زیرا حالت CTC دارای بافر مضاعف نمی باشد. حالت های متفاوت بیت های COM00 و COM01 در جدول ۶-۴ آورده شده است.

جدول ۶-۴: تعیین حالت خروجی تطابق مقایسه (CTC)

| COM01 | COM00 | Description                              |
|-------|-------|------------------------------------------|
| 0     | 0     | Normal port operation, OC0 disconnected. |
| 0     | 1     | Toggle OC0 on compare match              |
| 1     | 0     | Clear OC0 on compare match               |
| 1     | 1     | Set OC0 on compare match                 |

همانگونه که از جدول ۶-۴ مشاهده می شود، چهار حالت به وجود می آید که عبارتند از:

- ۱ - پایه OC0 که پایه ای از یک Port می باشد به صورت I/O معمولی استفاده می شود.
- ۲ - پایه OC0 با خروجی مقایسه منطبق شده و در هر بار تطابق، خروجی تغییر وضعیت می دهد (Toggle).
- ۳ - پایه OC0 با خروجی مقایسه منطبق شده و در زمان تطابق، این پایه را صفر می کند.
- ۴ - پایه OC0 با خروجی مقایسه منطبق شده و در زمان تطابق، این پایه را یک می کند.

**نکته:** در حالت های ۲، ۳ و ۴ باید پین OC0 به صورت خروجی تعریف شود. (با نوشتن یک در DDRx).

## ۶-۹ عملکرد تایمر/کانتر صفر در حالت PWM سریع (تک شیب)

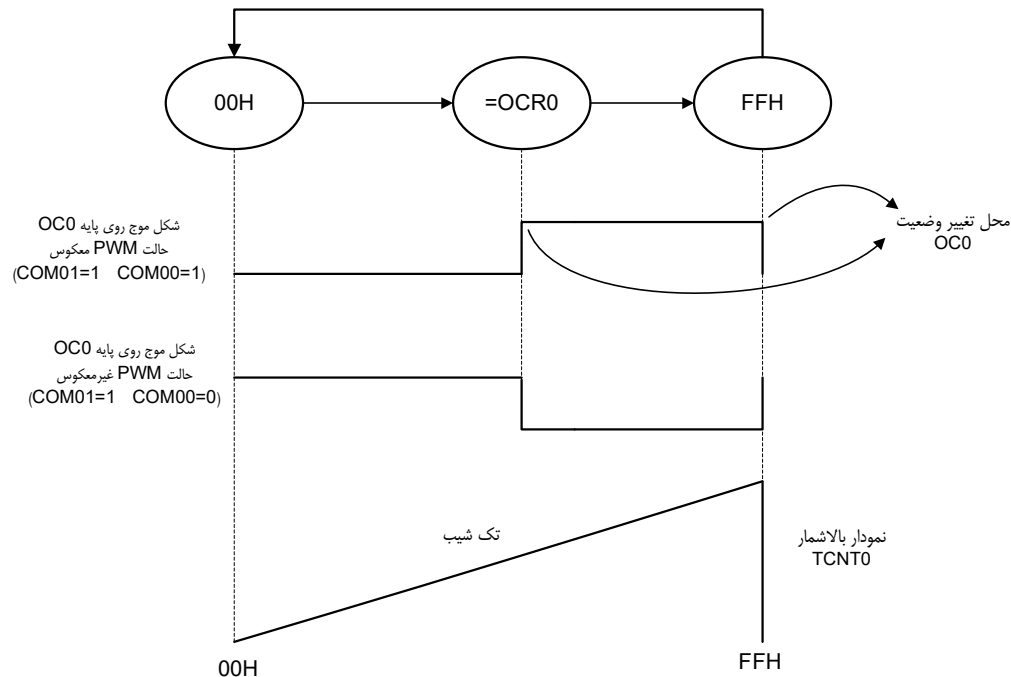
واژه PWM مخفف عبارت Pulse Width Modulation یا مدولاسیون پهنای پالس می باشد. در این مدولاسیون پهنای پالس تولیدی را می توان تحت کنترل داشت، به طوری که این پهنای در برخی مواقع می تواند متأثر از مقدار دامنه یک موج دیگر باشد. از کاربردهای PWM می توان به کنترل دور موتورهای AC، DC و منابع تغذیه سوییچینگ و ... اشاره نمود.

جهت ورود و دسترسی به مد عملکرد PWM سریع باید بیت های  $WGM00=1$  و  $WGM01=1$  قرار داد. همانگونه که درباره حالت مقایسه CTC توضیح داده شد، رجیستر OCR0 به طور دایم با رجیستر TCNT0 مقایسه می شود و پایه OC0 با توجه به تنظیمات مربوطه، تغییر وضعیت می دهد. عملکرد تایمر در حالت PWM سریع همانند حالت مقایسه می باشد با این تفاوت که در زمان تطابق بین OCR0 و TCNT0 رجیستر TCNT0 با مقدار 00H پر نمی شود، بلکه شمارش خود را تا حداکثر مقدار (FFH) ادامه داده و پس از سرریز شدن تایمر، رجیستر TCNT0 با مقدار 00H پر می شود و شمارش ادامه پیدا می کند. رجیستر OCR0 در حالت PWM دارای بافر مضاعف می باشد (شکل ۶-۴). باید توجه داشت که در حالت PWM پایه OC0 در دو حالت زیر تغییر وضعیت می دهد:

۱ - در حالت تطابق

۲ - در زمان سرریز تایمر

در صورتی که در حالت CTC فقط در زمان تطابق، وضعیت پایه OC0 تغییر می کند.



شکل ۶-۴: نمودار عملکرد تایمر/کانتر صفر در حالت PWM سریع

در زمانی که تایمر/کانتر صفر در حالت PWM سریع قرار می گیرد، بیت های COM00 و COM01 عملکرد متفاوتی با حالت مقایسه (CTC) پیدا می کنند. این حالت ها در جدول ۶-۵ آورده شده است.

جدول ۶-۵: حالت خروجی مقایسه در PWM سریع

| COM01 | COM00 | Description                                |
|-------|-------|--------------------------------------------|
| 0     | 0     | Normal port operation, OC0 disconnected.   |
| 0     | 1     | Reserved                                   |
| 1     | 0     | Clear OC0 on compare match, set CO0 at TOP |
| 1     | 1     | Set OC0 on compare match, clear CO0 at TOP |

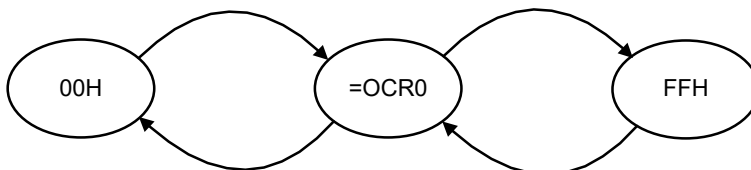
در صورتی که وقفه سرریز تایمر فعال باشد (وقفه سراسری فعال)، با هر بار فعال شدن TOV0 می توان در زیربرنامه روال سرویس وقفه (ISR) رجیستر OCR0 را Update نمود تا پهنای PWM به مقدار دلخواه تنظیم شود. فرکانس PWM خروجی را می توان از رابطه زیر محاسبه کرد:

$$f_{PWM} = \frac{f_{clk_{I/O}}}{N * 255} \quad (1-6)$$

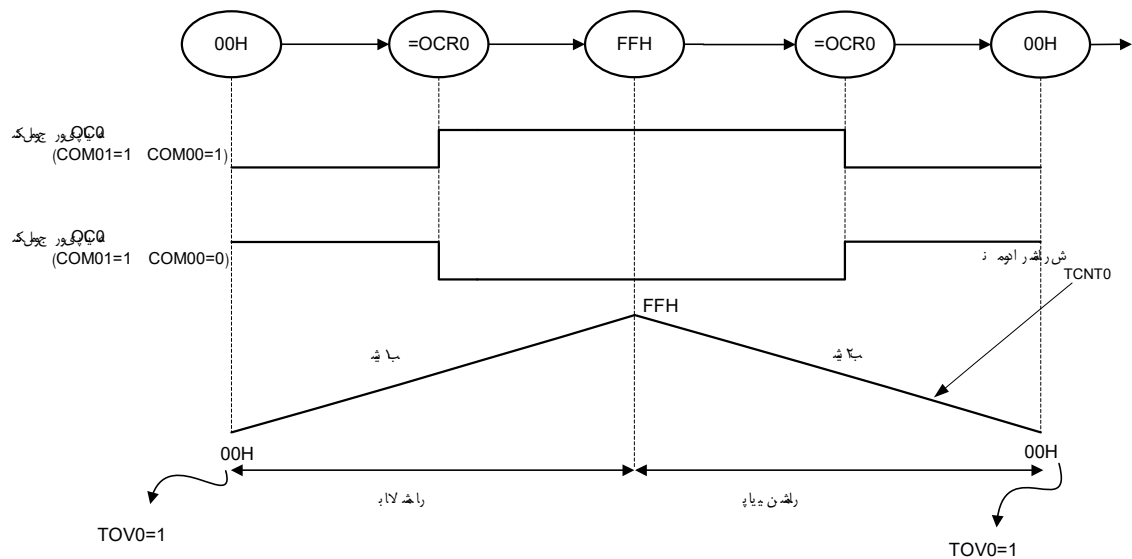
که در آن N بیانگر ضریب تقسیم پالس ساعت سیستم برای تولید پالس ساعت تایمر می باشد و برابر یکی از اعداد ۱، ۸، ۳۲، ۶۴، ۱۲۸، ۲۵۶ و ۱۰۲۴ است.

### ۶-۱۰ عملکرد تایمر/کانتر صفر در حالت PWM تصحیح فاز (دو شیب)

حالت PWM تصحیح فاز (WGM00=1 و WGM01=0) امکان تولید شکل موج PWM با تفکیک پذیری بالا را مهیا می سازد. در PWM تصحیح فاز مکرراً از 00H (BOTTOM) تا FFH (TOP) و سپس از FFH تا 00H شمارش می نماید. در هنگام شمارش به سمت بالا مقایسه گر، رجیستر TCNT0 را با OCR0 مقایسه می کند و هنگام برابر شدن، یک تغییر وضعیت روی پایه OC0 ایجاد می نماید و هنگام شمارش به سمت پایین، نیز همین عمل مجدداً تکرار می شود. حالت عملکرد PWM به روش تصحیح فاز دارای فرکانس تولیدی پایین تری نسبت به PWM سریع می باشد. با این وجود با توجه به متقارن بودن عملکرد PWM تصحیح فاز، از آن بیشتر در کنترل دور موتور استفاده می شود. شکل ۶-۵ و شکل ۶-۶ نحوه عملکرد تایمر را در PWM تصحیح فاز نشان می دهند.



شکل ۶-۵: نمودار شمارش تایمر در حالت PWM تصحیح فاز



شکل ۶-۶: عملکرد تایمر در حالت تصحیح فاز

یک نکته مهم این است که پرچم سرریز تایمر صفر (TOV0) زمانی فعال می شود که رجیستر TCNT0 برابر صفر باشد نه FFH. پس باید توجه داشت که در زمان شروع به کار تایمر، اگر TCNT0=0 باشد، پرچم سرریز فعال خواهد شد. فرکانس موج PWM در حالت تصحیح فاز طبق معادله زیر قابل محاسبه است :

$$f_{PW} = \frac{f_{clk\_I/O}}{N * 510} \quad (۲-۶)$$

که در آن N بیانگر ضریب تقسیم پالس ساعت سیستم برای تولید پالس ساعت تایمر می باشد و برابر یکی از اعداد ۱، ۸، ۳۲، ۶۴، ۱۲۸، ۲۵۶ و ۱۰۲۴ است.

برای حالتی که تایمر/کانتر صفر در حالت PWM تصحیح فاز قرار می گیرد، بیت های COM00 و COM01 عملکرد متفاوتی پیدا می کنند که در جدول ۶-۶ آورده شده است.

جدول ۶-۶: حالت خروجی مقایسه در PWM تصحیح فاز

| COM01 | COM00 | Description                                                                              |
|-------|-------|------------------------------------------------------------------------------------------|
| 0     | 0     | Normal port operation, OC0 disconnected.                                                 |
| 0     | 1     | Reserved                                                                                 |
| 1     | 0     | Clear OC0 on compare match when up-counting. Set OC0 on compare match when downcounting. |
| 1     | 1     | Set OC0 on compare match when up-counting. Clear OC0 on compare match when downcounting. |

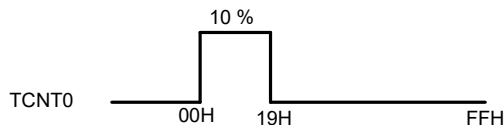
## ۱۱-۶ آزمایش اول

ابتدا مدار شکل ۷-۶ را ببندید.



## ۱۲-۶ آزمایش دوم

در این قسمت، می خواهیم برنامه ای را بنویسیم که یک موج PWM با عرض پالس ۱۰٪ را روی پایه OC0 ایجاد نماید (با استفاده از کریستال ۱۶ MHz خارجی). برای به دست آوردن پهنای پالس ۱۰٪ می توان به صورت زیر عمل نمود :



شکل ۶-۸: ایجاد پهنای پالس ۱۰٪

بنابراین داریم:

$$Duty\ Cycle = \frac{Time\ On}{Time\ On + Time\ Off} \times 100 \quad (۴-۶)$$

و

$$\frac{255}{OCR0} = \frac{100\%}{10\%} = OCR0 = 25.5 \rightarrow OCR0 = 25 \quad (۵-۶)$$

البته در این روش، یک خطا در پهنای پالس ایجاد می شود که می توان از آن صرف نظر نمود. برنامه زیر را می توان برای ایجاد چنین شکل موجی به کار برد.

```
#include <mega64.h>
void main() {
    DDRB=0x10;
    TCNT0=0x00;
    OCR0=25;
    TCCR0=0x6A;
    while(1);
}
```

مدار این قسمت از آزمایش، همانند قسمت قبل می باشد.

## ۱۳-۶ آزمایش سوم

با استفاده از تایمر صفر برنامه ای بنویسد که اعداد صفر تا ۹ را به ترتیب بر روی یک 7segment و با فاصله زمانی ۰/۲۵ ثانیه نشان دهد. فرکانس پالس ساعت را برابر با ۱ MHz و N را برابر با ۱۰۲۴ در نظر بگیرید.

## ۱۴-۶ پرسش

۱- در برنامه قسمت دوم آزمایش، منظور از خطای ذکر شده چیست و چگونه می توان آن را برطرف کرد؟ (برنامه آن را بنویسید).

۲- منظور از بافر مضاعف چیست؟

۳- در برنامه ساعت دیجیتال در آزمایش ۵ برای ایجاد تأخیر به جای استفاده از دستور delay\_ms() از تایمر استفاده کنید.