



دانشکده مهندسی کامپیوتر
دانشگاه صنعتی امیرکبیر

کنترل سیستم و ریست وقفه ها درگاه های ورودی / خروجی

در میکروکنترل های AVR



دانشکده مهندسی کامپیوتر
دانشگاه صنعتی امیرکبیر

فهرست مطالب

• کنترل سیستم و بازنشانی

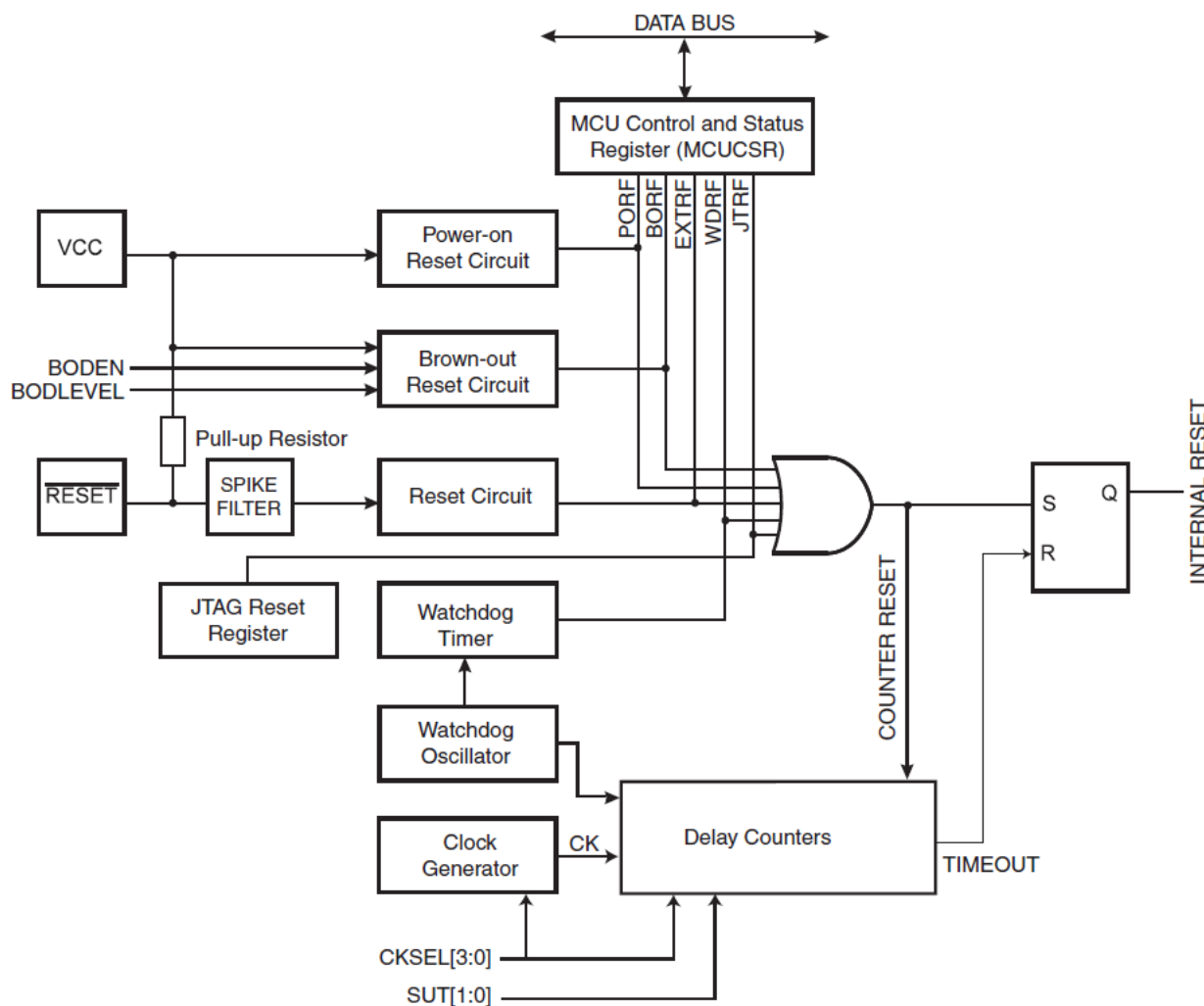
• زمان سنج نگهبان

بازنشانی

- در هنگام بازنشانی و شروع مجدد، همه ثبات‌های ورودی/خروجی مقدار اولیه خود را اختیار کرده و برنامه از بردار (آدرس) بازنشانی شروع به اجرا خواهد کرد.
- دستورالعمل قرار داده شده در بردار بازنشانی باید یک دستور پرش مطلق به روال اجرای بازنشانی باشد.
- اگر برنامه هیچگاه منابع وقفه را فعال نکند، بردارهای وقفه استفاده نشده و برنامه‌های عادی می‌توانند در این مکان‌ها قرار گیرند.

نمودار جعبه‌ای سیستم بازنشانی میکروکنترلر

• واحد بازنشانی

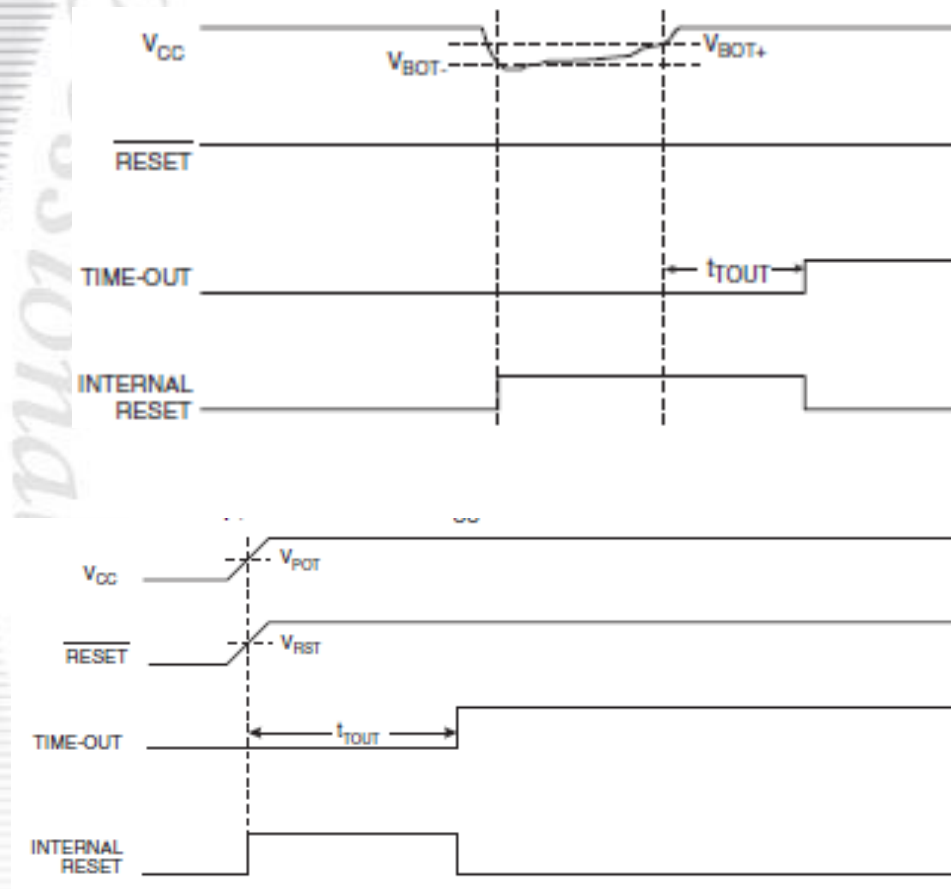


مشخصات بازنشانی

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{POT}	Power-on Reset Threshold Voltage (rising)			1.4	2.3	V
	Power-on Reset Threshold Voltage (falling) ⁽¹⁾			1.3	2.3	V
V_{RST}	\overline{RESET} Pin Threshold Voltage		0.1 V_{CC}		0.9 V_{CC}	V
t_{RST}	Minimum pulse width on \overline{RESET} Pin				1.5	μs
V_{BOT}	Brown-out Reset Threshold Voltage ⁽²⁾	BODLEVEL = 1	2.5	2.7	3.2	V
		BODLEVEL = 0	3.6	4.0	4.5	
t_{BOD}	Minimum low voltage period for Brown-out Detection	BODLEVEL = 1		2		μs
		BODLEVEL = 0		2		μs
V_{HYST}	Brown-out Detector hysteresis			50		mV

توجه ۲: بازنشانی Power-on تا زمانی که ولتاژ تغذیه به زیر حد V_{POT} نرسیده باشد کار نخواهد کرد.

BOT: Brown out تشخیص



بازنشانی

- درگاه‌های ورودی/خروجی مربوط به میکروکنترلر AVR بلافاصله پس از فعال شدن یک منبع بازنشانی، به حالت آغازین خود برمی‌گردند که این موضوع نیازی به فعال کردن هیچ منبع ساعتی ندارد.
- هنگامی که همه منابع بازنشانی غیرفعال شدند، یک شمارنده برای ایجاد تاخیر شروع به کار می‌کند تا زمان بازنشانی درونی را افزایش دهد (زمان **time-out**). این کار باعث می‌شود پیش از آغاز عملیات معمولی، میکروکنترلر به یک سطح پایدار برسد.
- زمان تاخیر این شمارنده به وسیله کاربر و توسط فیوزهای CKSEL تعیین می‌شود. انتخاب‌های متفاوت برای زمان‌های تاخیر در بخش "منابع ساعت" آمده است.

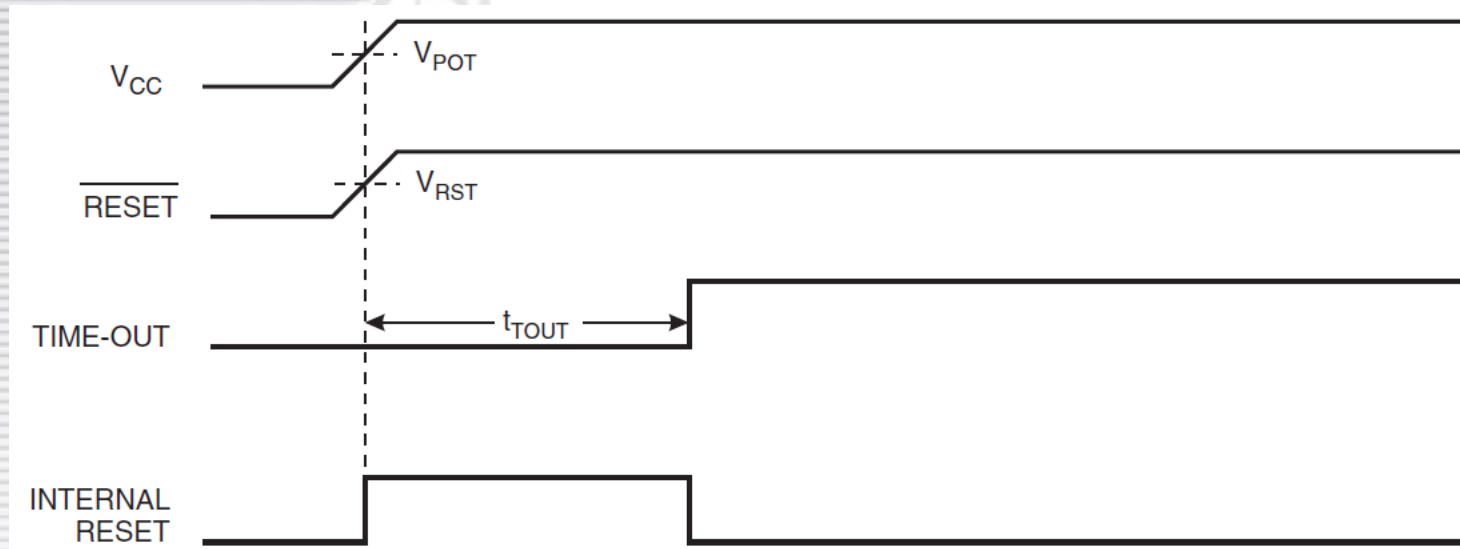
منابع ریست

- میکروکنترلر ATmega16، دارای ۵ منبع بازنشانی است:
- بازنشانی Power-on
- بازنشانی خارجی
- بازنشانی نگهبان
- بازنشانی افت ولتاژ تغذیه
- بازنشانی JTAG

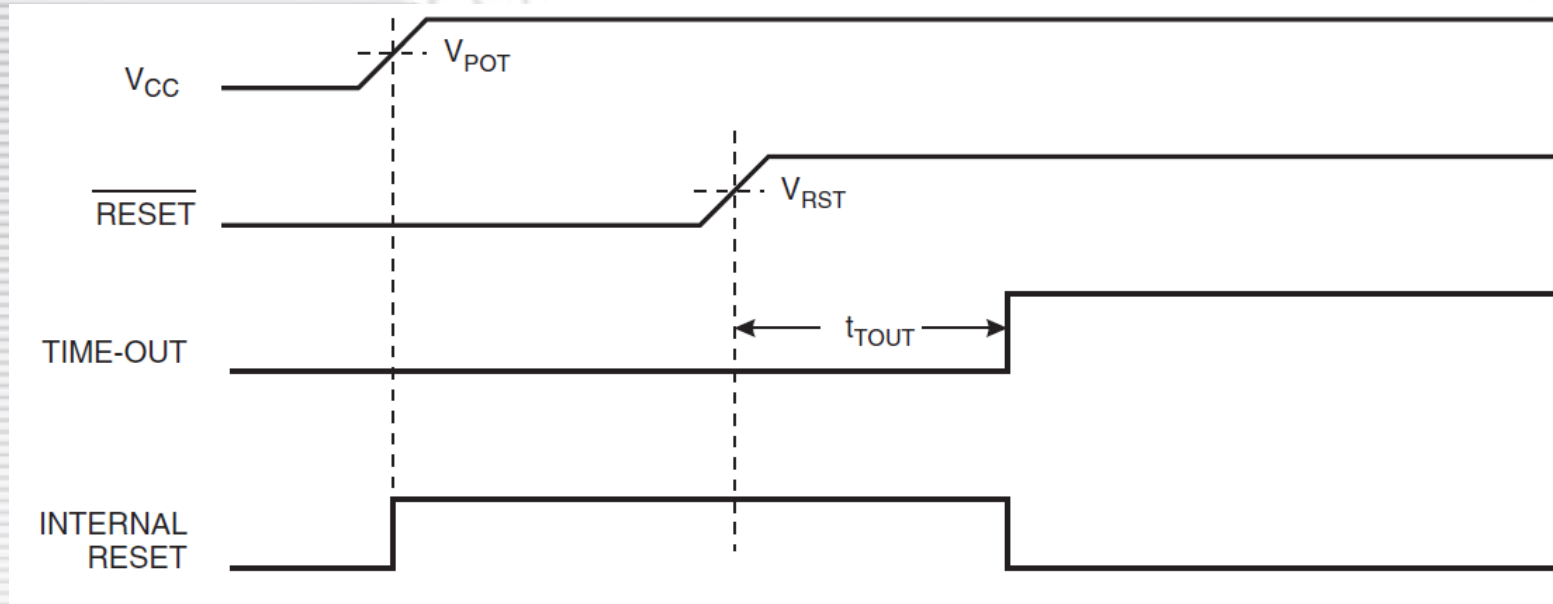
بازنشانی Power-on

- یک پالس بازنشانی Power-on توسط یک مدار تشخیص سوار بر تراشه (موجود در تراشه میکروکنترلر) تولید می‌شود.
- بازنشانی Power-on هنگامی که V_{CC} کم‌تر از مقدار ولتاژ تشخیص باشد فعال می‌شود.
- رسیدن به ولتاژ آستانه بازنشانی Power-on، شمارنده تاخیر را فعال می‌کند و تعیین می‌کند که میکروکنترلر چه مدت بعد از بالا رفتن V_{CC} در حالت بازنشانی باقی بماند.
- بازنشانی، هنگامی که V_{CC} کم‌تر از ولتاژ تشخیص شود، بدون تاخیر، فعال می‌شود.

بازنشانی میکروکنترلر هنگامی که سر RESET به VCC وصل است



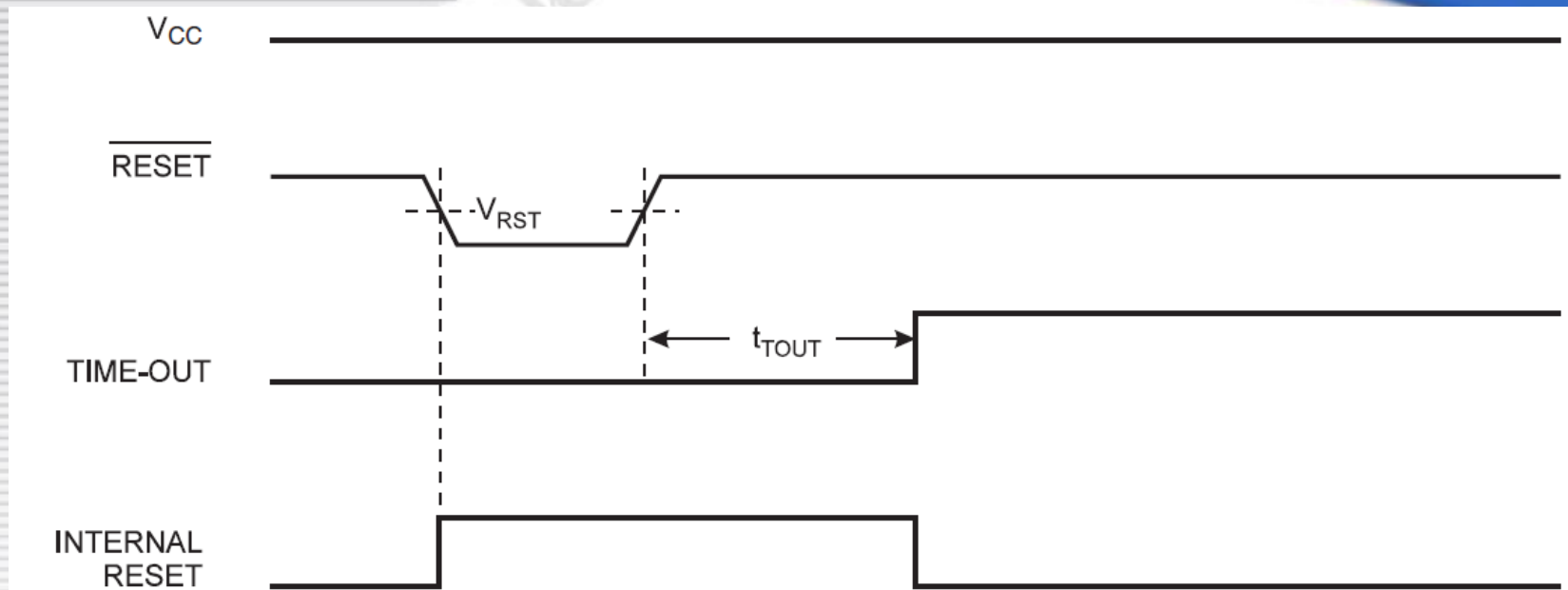
شروع به کار میکروکنترلر هنگامی که سر RESET توسط یک مدار بازنشانی خارجی فعال شود



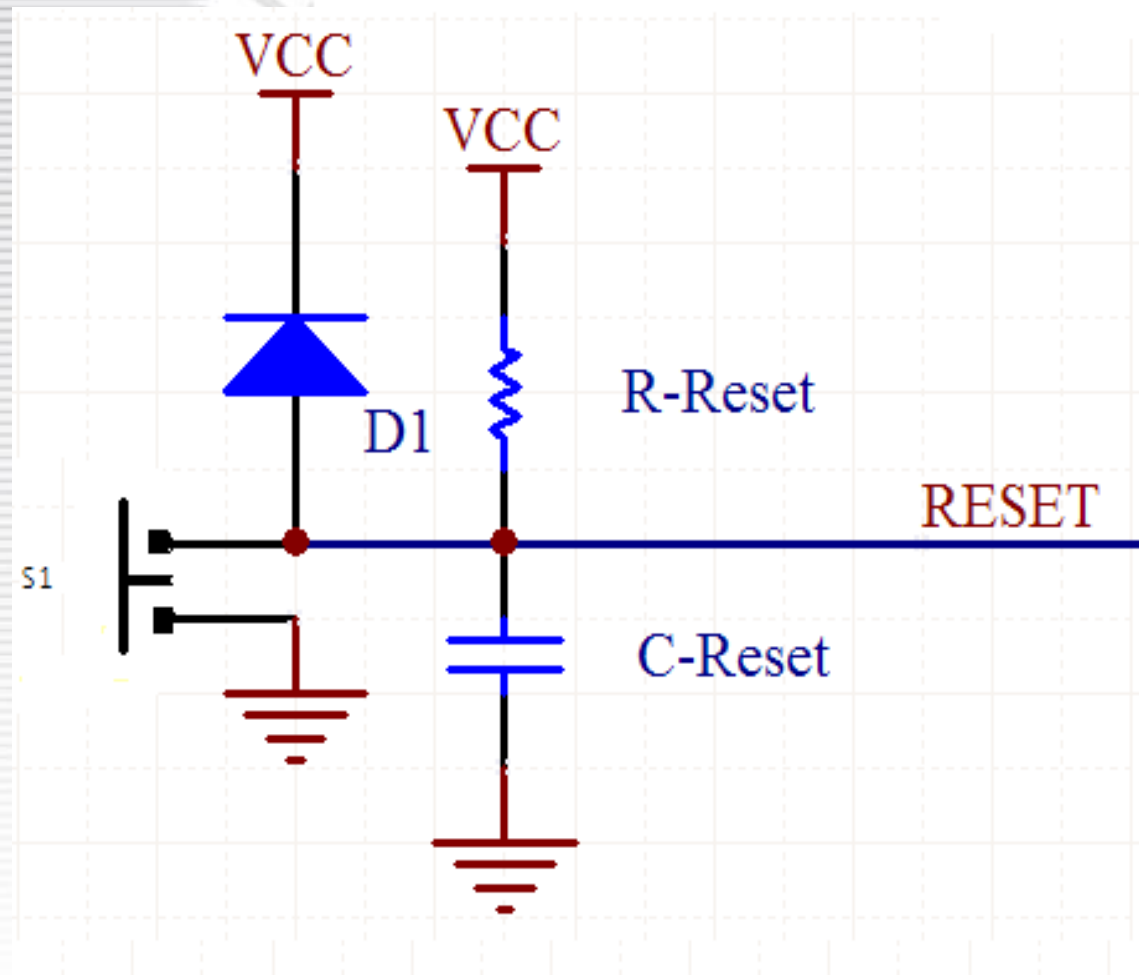
بازنشانی خارجی

- بازنشانی خارجی با وصل کردن یک ولتاژ سطح پایین (صفر منطقی) به پایه \overline{RESET} ایجاد می‌شود.
- سیگنال‌های بازنشانی طولانی‌تر از کمینه عرض پالس ریست، بازنشانی را حتی اگر ساعت فعال نباشد (در حال کار نباشد) ایجاد می‌کند.
- سیگنال‌های بازنشانی با زمان کوتاه‌تر از کمینه مورد نیاز، بازنشانی شدن میکروکنترلر را تضمین نمی‌کنند.
- هنگامی که سیگنال اعمال شده در لبه بالا رونده، به ولتاژ آستانه بازنشانی (V_{RST}) می‌رسد، شمارنده تاخیر، پس از سپری شدن زمان مهلت t_{TOUT} time-out، میکروکنترلر را مجدداً راه‌اندازی می‌کند.

بازنشانی خارجی در حین کارکرد میکروکنترلر



مدار بازنشانی خارجی



تشخیص افت ولتاژ تغذیه (Brown out detection)

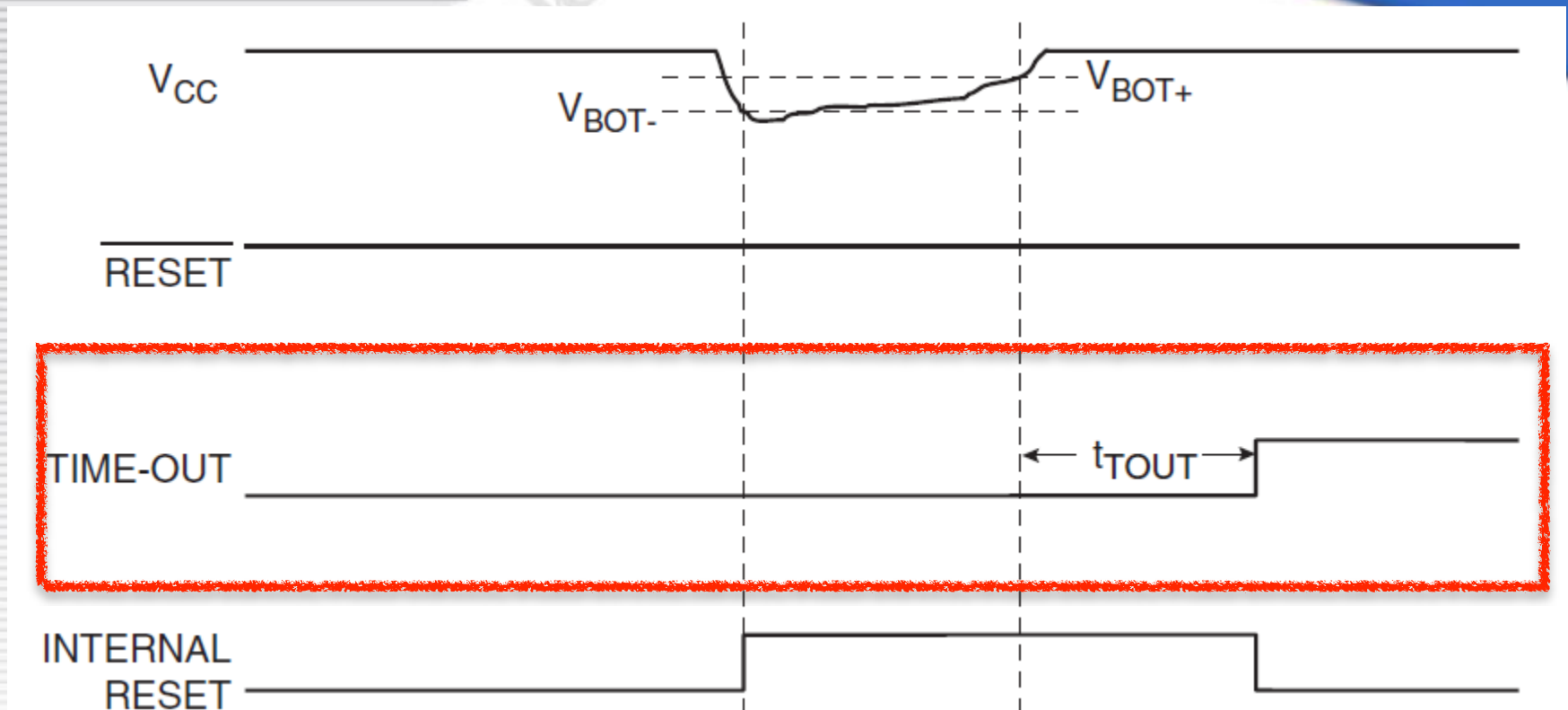
- میکروکنترلر ATmega16 یک مدار تشخیص افت ولتاژ تغذیه (Burn Out BOD: Detection) دارد که در زمان کار کردن میکروکنترلر، سطح ولتاژ V_{CC} را نظارت کرده آن را با یک مقدار سطح تحریک مقایسه می کند.
- سطح تحریک BOD می تواند توسط بیت فیوز **BODLEVEL** برابر **۲.۷** ولت (سطح تحریک برنامه ریزی نشده) یا **۴** ولت (سطح تحریک برنامه ریزی شده) برنامه ریزی شود.

تشخیص افت ولتاژ تغذیه

- سطح تحریک دارای یک هیستریزیس می باشد که باعث می شود تشخیص افت ولتاژ تغذیه بدون تاثیر از ولتاژهای سوزنی ناخواسته (spike) صورت گیرد.
- این سطح تشخیص می تواند به صورت زیر تخمین زده شود:

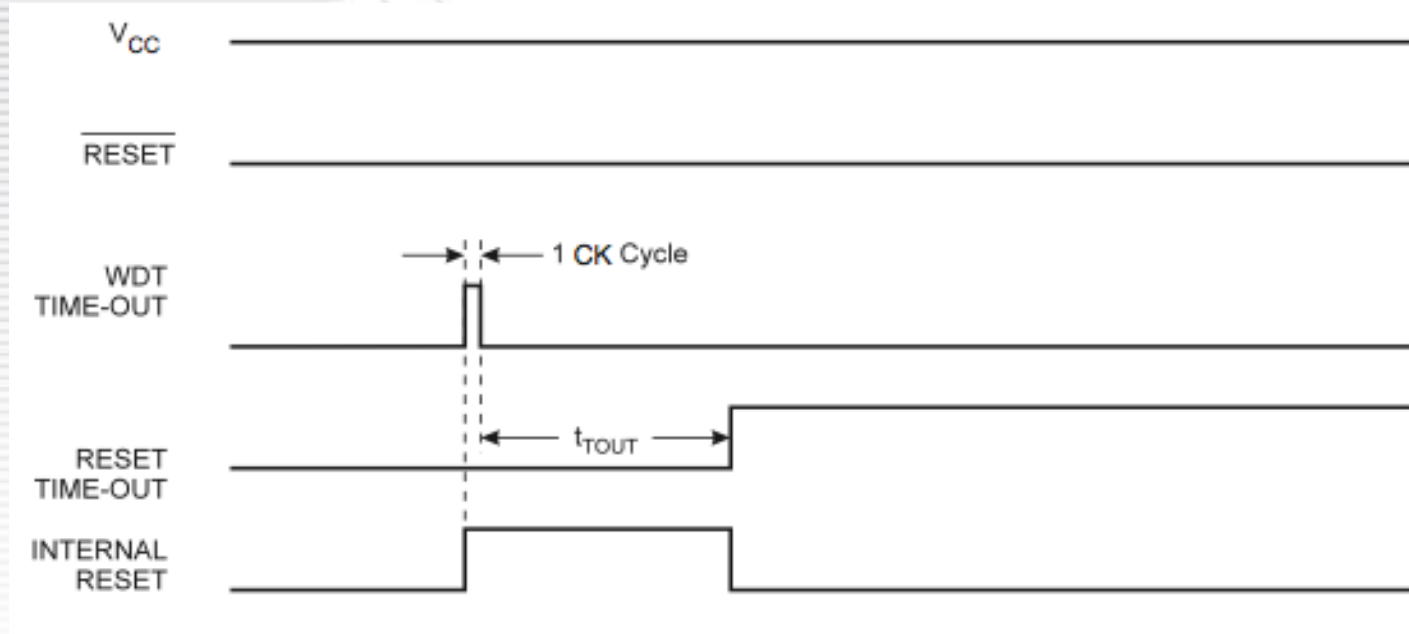
$$V_{\text{BOT}+} = V_{\text{BOT}} + V_{\text{HYST}}/2 \text{ و } V_{\text{BOT}-} = V_{\text{BOT}} - V_{\text{HYST}}/2$$

بازنشانی ناشی از افت ولتاژ تغذیه در حین عملیات



بازنشانی نگهبان

- وقتی که زمان زمان سنج نگهبان، سپری می شود، یک پالس بازنشانی به مدت زمان یک چرخه ساعت ایجاد می کند.
- در لبه پایین رونده این پالس، شمارنده تاخیر شروع به شمارش زمان t_{TOUT} time-out می کند.



ثبات کنترول و وضعیت میکروکنترلر (MCUCSR)

Bit	7	6	5	4	3	2	1	0	MCUCSR
	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	
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					

بیت ۴: پرچم بازنشانی JTAG (JTRF)

بیت ۳: پرچم بازنشانی watchdog (WDRF)

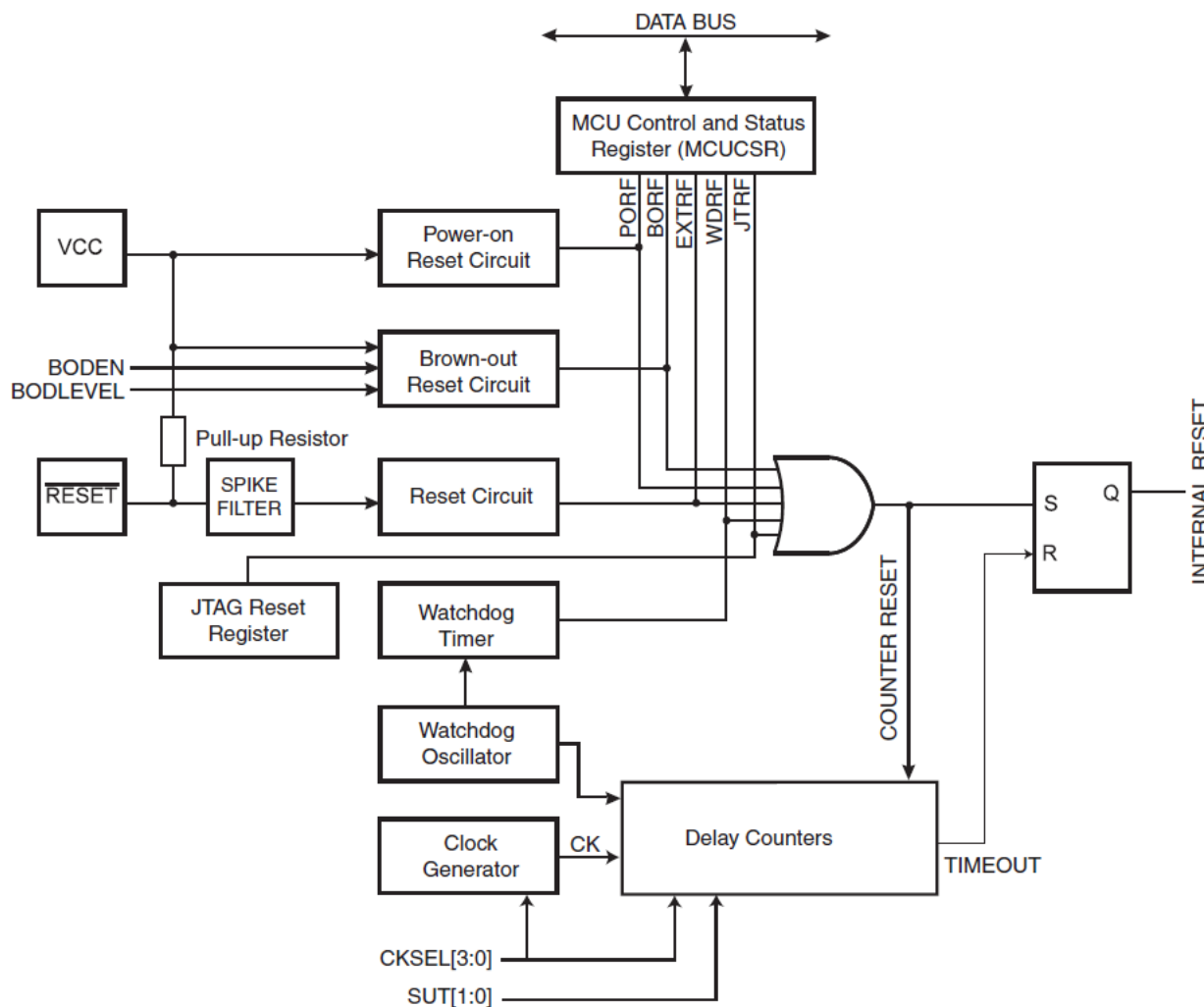
بیت ۲: پرچم بازنشانی افت ولتاژ تغذیه (BORF)

بیت ۱: پرچم بازنشانی خارجی (EXTRF)

بیت ۰: پرچم بازنشانی Power-on (PORF)

نمودار جعبه‌ای سیستم بازنشانی میکروکنترلر (یادآوری)

• واحد بازنشانی



منبع ولتاژ داخلی

- میکروکنترلر ATmega16 دارای یک منبع ولتاژ مرجع bandgap درونی است.
- این منبع برای شناسایی افت ولتاژ تغذیه استفاده می‌شود و می‌تواند به عنوان ورودی مقایسه کننده‌های آنالوگ یا ADC به کار رود.
- منبع ولتاژ مرجع در حین وضعیت‌های زیر روشن می‌باشد:
 - هنگامی که BOD فعال شده باشد (به وسیله برنامه‌ریزی فیوز BODEN).
 - هنگامی که ولتاژ مرجع bandgap به یک مقایسه کننده آنالوگ متصل شده باشد (از طریق یک کردن بیت ACBG در ACSR).
 - هنگامی که ADC فعال باشد.

مشخصات منبع ولتاژ مرجع درونی

Symbol	Parameter	Min	Typ	Max	Units
V_{BG}	Bandgap reference voltage	1.15	1.23	1.4	V
t_{BG}	Bandgap reference start-up time		40	70	μs
I_{BG}	Bandgap reference current consumption		10		μA

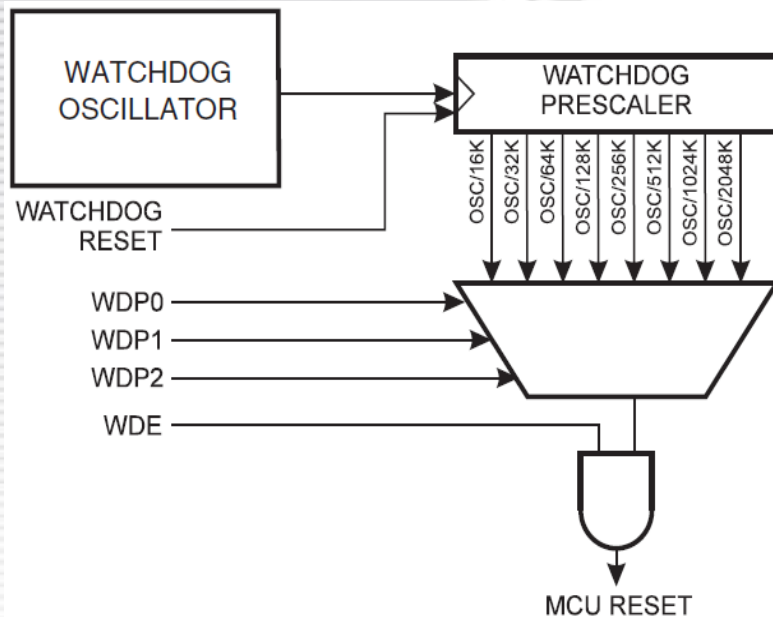
زمان سنج نگهبان

- زمان سنج نگهبان از طریق یک نوسان ساز سوار بر تراشه جداگانه که با سرعت ۱ مگاهرتز نوسان می کند (وضعیت معمول بازا $V_{CC} = 5V$)، سیگنال ساعت را دریافت می نماید.
- دستور WDR زمان سنج نگهبان را بازنشانی می کند.
- زمان سنج نگهبان همچنین هنگامی که آنرا غیرفعال کنیم و نیز زمانی که تراشه بازنشانی شود، بازنشانی می شود.
- اگر زمان بازنشانی بدون یک بازنشانی نگهبان به پایان برسد، میکروکنترلر ATmega16 توسط زمان سنج نگهبان بازنشانی می شود.

زمان سنج نگهبان

- ۸ انتخاب مختلف می تواند برای تعیین فرکانس ساعت زمان سنج نگهبان انتخاب شود.

- این انتخاب توسط بیت های **WDP2**، **WDP1** و **WDP0** (پیش مقیاس گذارهای ۰ و ۱ و ۲ متعلق به زمان سنج نگهبان) که در ثبات **WDTCR** قرار دارند انجام می شود.



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	

نحوه انتخاب پیش تقسیم ساعت زمان سنج نگهبان

- محاسبه زمانی که بعد از سپری شدن آن، زمان سنج نگهبان اقدام به بازنشانی میکروکنترلر می نماید با توجه به مقادیر بیت های WDP2، WDP1 و WDP0

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

ثبات کنترلی زمان سنج نگهبان (WDTCR)

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	

- بیت‌های ۰ تا ۲: **WDP0**، **WDP1** و **WDP2** (پیش مقیاس‌گذارهای ۰ و ۱ و ۲ متعلق به زمان‌سنج نگهبان)
- بیت ۳: **WDE** (برای فعال کردن زمان‌سنج نگهبان این بیت باید یک شود)
- بیت ۴: **WDTOE** (هنگامی که صفر در بیت **WDE** نوشته می‌شود، این بیت باید یک شود. در غیر این صورت زمان‌سنج نگهبان غیرفعال نخواهد شد. یکبار که این بیت یک شود، سخت‌افزار، بعد از ۴ چرخه ساعت این بیت را مجدداً صفر می‌کند).
- بیت‌های ۵ تا ۷: بیت‌های رزرو شده

برنامه خاموش کردن زمان سنج نگهبان

- برنامه‌های زیر یک تابع اسمبلی و یک تابع C برای خاموش کردن زمان سنج نگهبان WDT را نشان می‌دهند.
- در این مثال‌ها فرض شده است که در حین اجرای این تابع‌ها وقفه‌ای رخ نمی‌دهد (برای مثال از طریق غیر فعال کردن سراسری وقفه‌ها).

مراحل غیر فعال سازی ساعت نگهبان

۱- یک کردن WDE و WDTOE

۲- صفر کردن WDE

Assembly Code

```
WDT_off:
;Reset WDT
WDR
; Write logical one to WDTOE and WDE
in r16, WDTCR
ori r16, (1<<WDTOE)|(1<<WDE)
out WDTCR, r16
; Turn off WDT
ldi r16, (0<<WDE)
out WDTCR, r16
ret
```

C Code

```
void WDT_off(void)
{
/* Reset WDT*/
_WDR();
/* Write logical one to WDTOE and WDE */
WDTCR |= (1<<WDTOE) | (1<<WDE);
/* Turn off WDT */
WDTCR = 0x00;
}
```