

# System Clock

System Clock and Clock Options

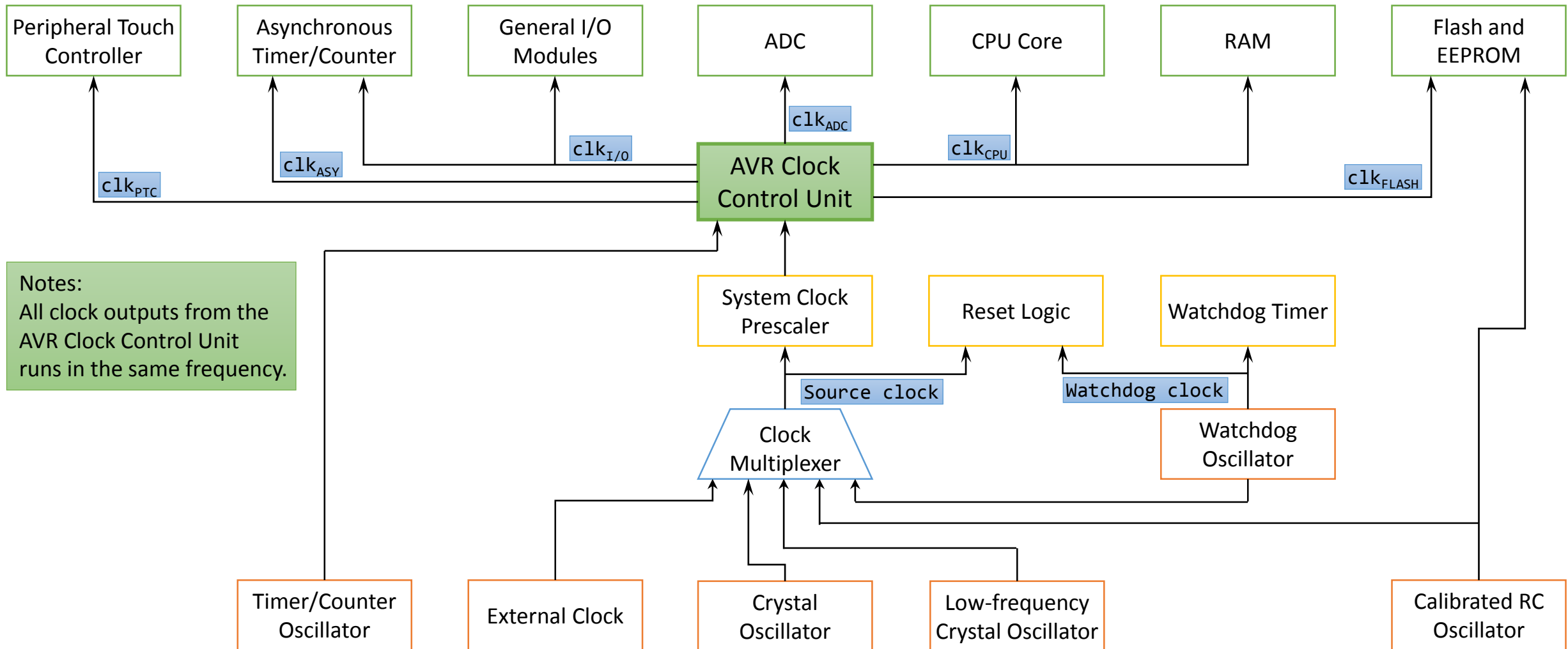
# Objectives

---

- Clock Systems and Their Distribution
- Clock Sources
- System Clock Prescaler

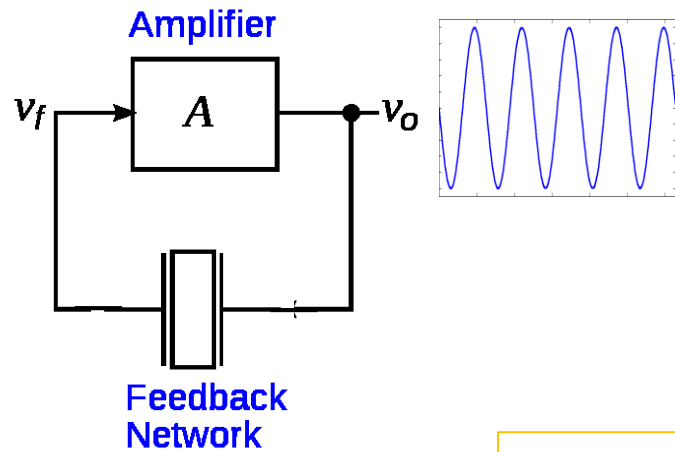
# Clock Systems and Their Distribution

# Clock Systems and Their Distribution

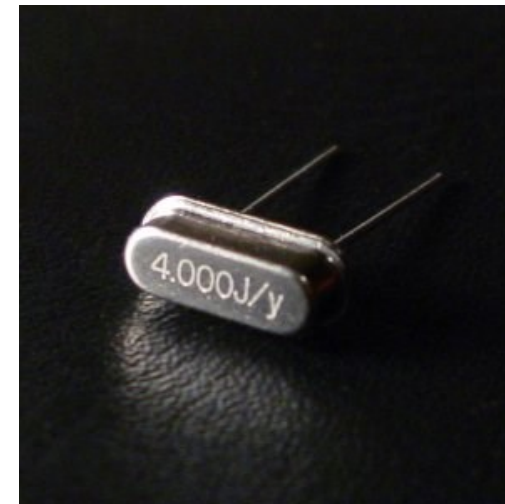
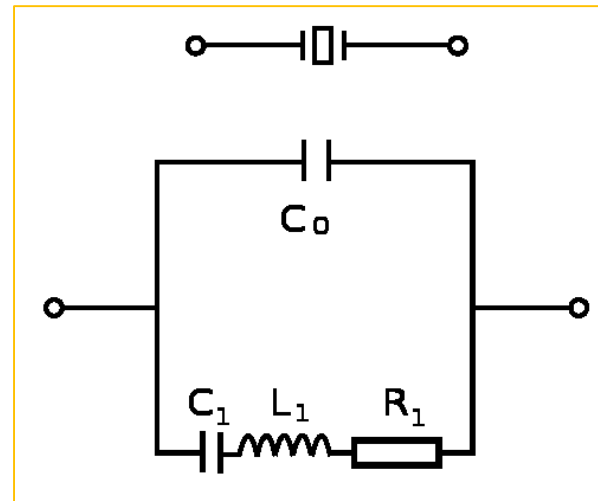


# Clock Sources

# Crystal Oscillator



An electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency.



# Clock Sources (1)

- Clock source is selected by Flash Fuse bits, **CKSEL[3:0]** (Table 32-7, p387, 328PB datasheet)

## I. Calibrated Internal RC Oscillator 8 MHz

- Default Clock Source
- **CKSEL[3:0]** = 0010
- **SUT[1:0]** = 10
- **CKDIV8** = 0 (programmed)
- The system clock is **1.0** MHz
- **OSCCAL** register: for calibration

Device Clocking Option	CKSEL[3:0]
Low Power Crystal Oscillator	1111 – 1000
Low Frequency Crystal Oscillator	0101 - 0100
Internal 128kHz RC Oscillator	0011
Calibrated Internal RC Oscillator 8 MHz	0010
External Clock	0000
Reserved	0001

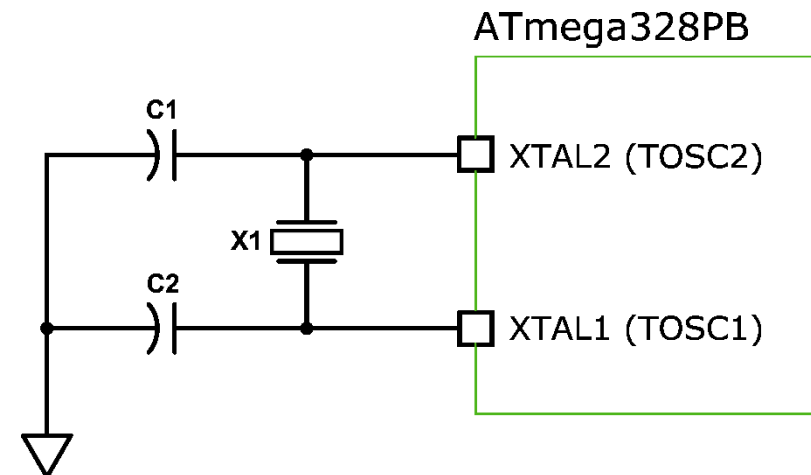
# Clock Sources (2)

## II. Low Power Crystal Oscillator

- A quartz crystal or a ceramic resonator may be used.
- Low Power Crystal Oscillator Operating Modes
- CKSEL Fuses: **1111** - **1000**



Frequency Range [MHz]	Capacitance C1 and C2 [pF]	CKSEL[3:1]
0.4 – 0.9 (ceramic resonators only)	-	100
0.9 – 3.0	12-22	101
3.0 – 8.0		110
8.0 – 16.0		111





# Clock Sources (3)

---

## III. Low Frequency Crystal Oscillator

- 32.768kHz watch crystal
- CKSEL Fuses: 0100 or 0101

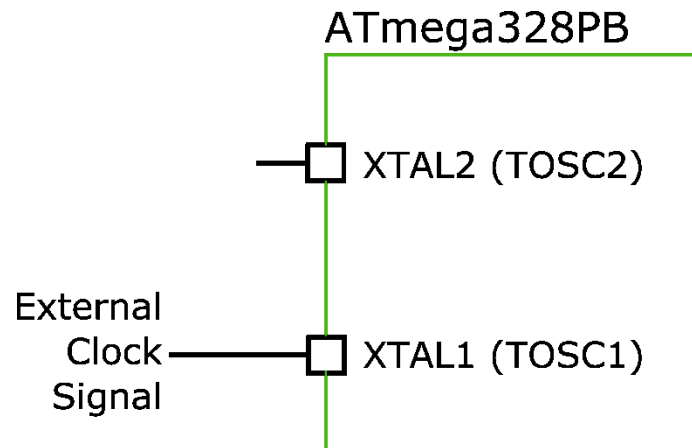
## IV. 128kHz Internal Oscillator

- CKSEL Fuses: 0011
- Used for clock source of Watchdog Timer(WDT)
- Used for clock source of Clock Failure Detection(CFD) mechanism

# Clock Sources (4)

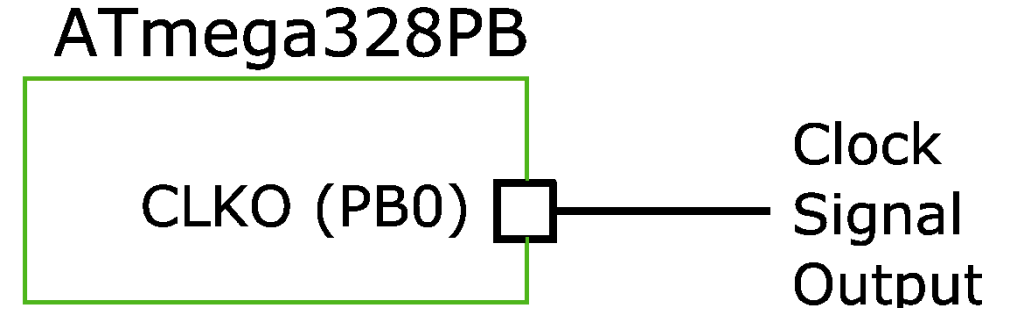
## V. External Clock

- CKSEL Fuses: 0000



# Clock Output

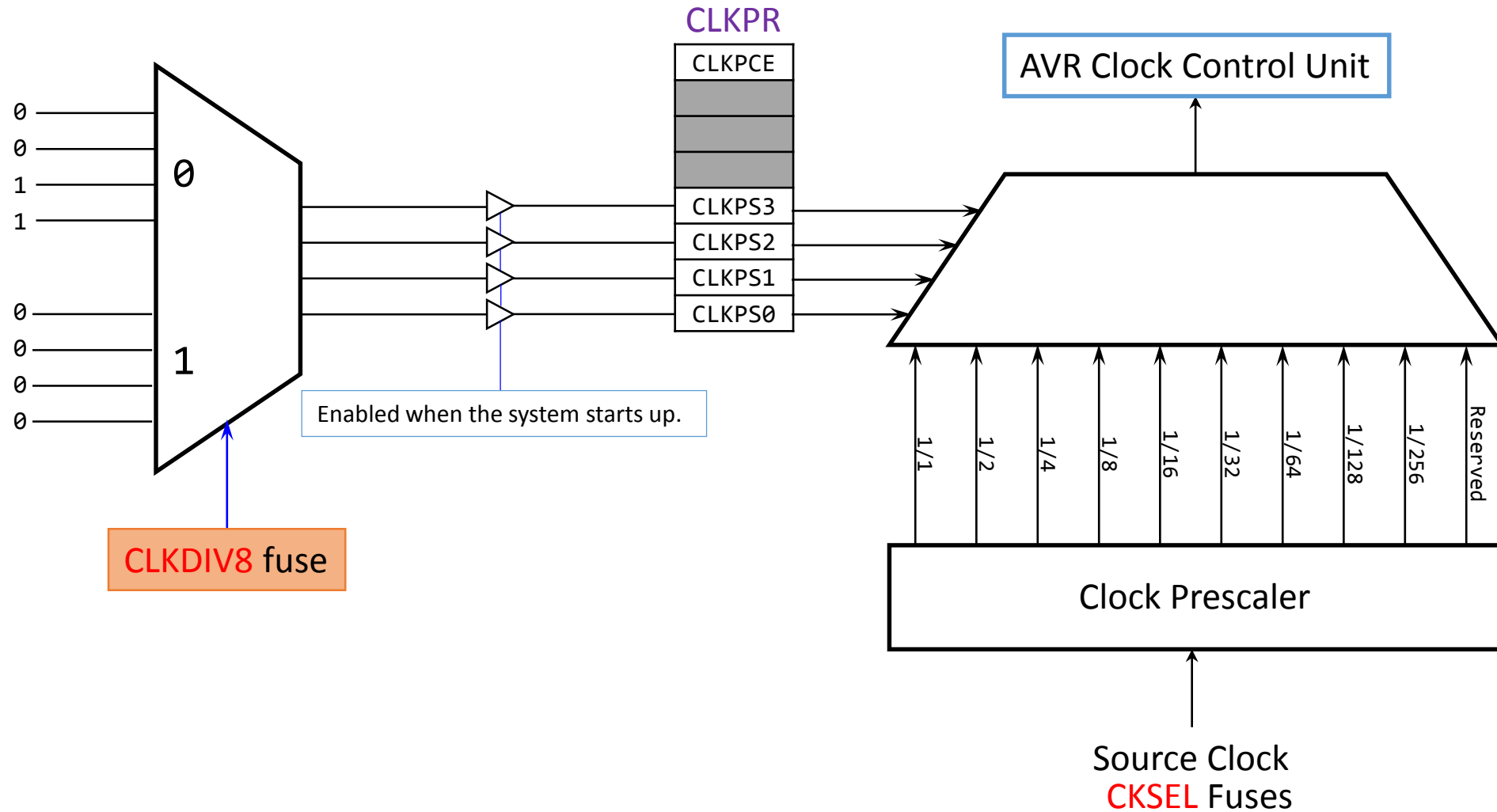
- CKOUT Fuse: 0 (Programmed)
- This mode is suitable when the chip clock is used to drive other circuits on the system.
- The clock also will be output during reset.
- Normal operation of I/O pin will be overridden when the fuse is programmed.
- Any clock source, including the internal RC Oscillator, can be selected when the clock is output on CLKO.
- If the System Clock Prescaler is used, it is the divided system clock that is output.



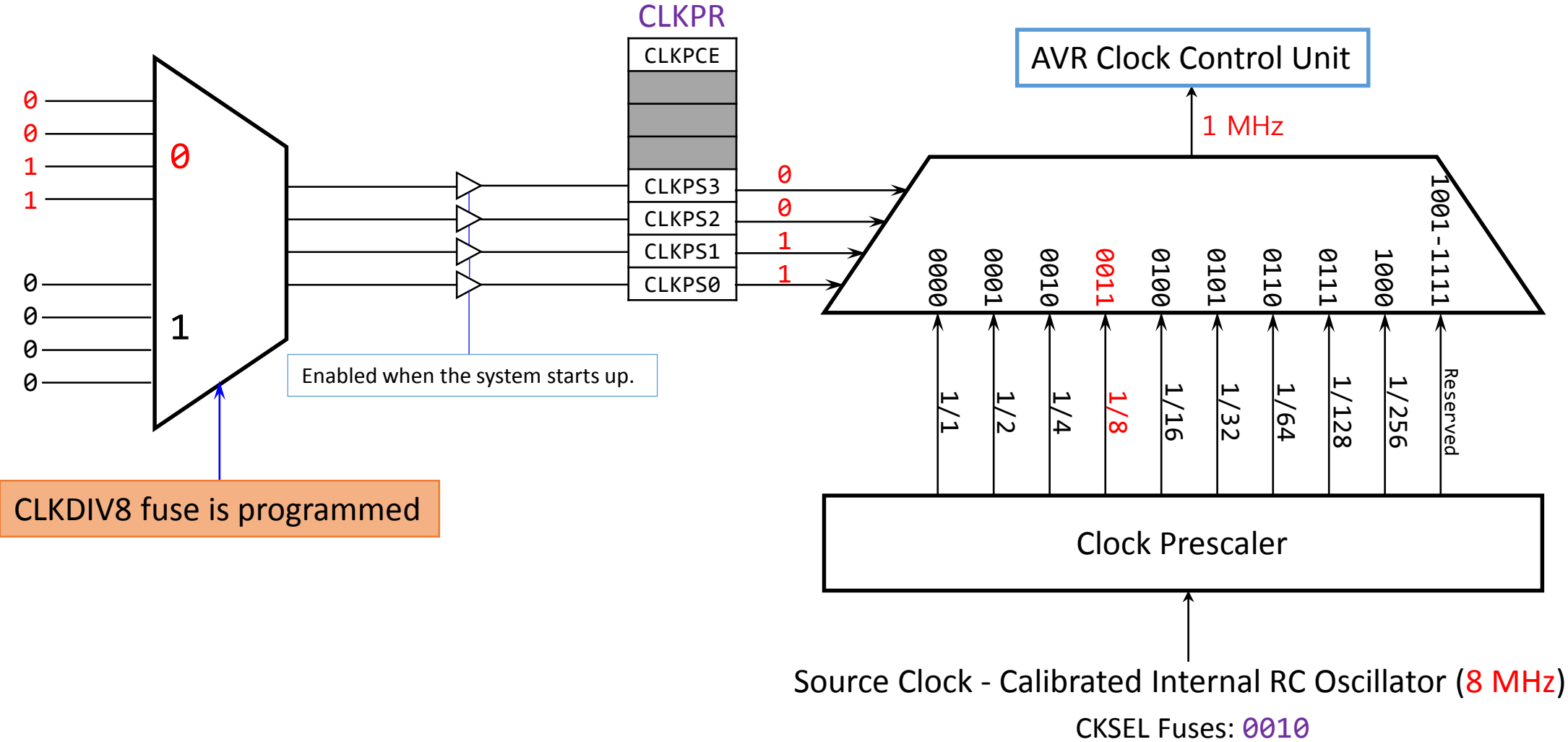
# System Clock Prescaler (1)

- System clock can be divided by configuring the **Clock Prescale Register (CLKPR)**.
- To decrease the power consumption when the requirement for processing power is low.
- Can be used with all clock source options, and it will affect the clock frequency of the CPU and all synchronous peripherals.
- $\text{clk}_{I/O}$ ,  $\text{clk}_{ADC}$ ,  $\text{clk}_{CPU}$ , and  $\text{clk}_{FLASH}$  are divided by a factor as shown in the **CLKPR** description.
- A special write procedure must be followed to change the **CLKPS[3:0]** bits:
  1. Write the Clock Prescaler Change Enable (CLKPCE) bit to '1' and all other bits in CLKPR to zero: **CLKPR=0x80**.
  2. Within four cycles, write the desired value to CLKPS[3:0] while writing a zero to CLKPCE: **CLKPR=0x0N**
- Interrupts must be disabled when changing prescaler setting to make sure the write procedure is not interrupted.

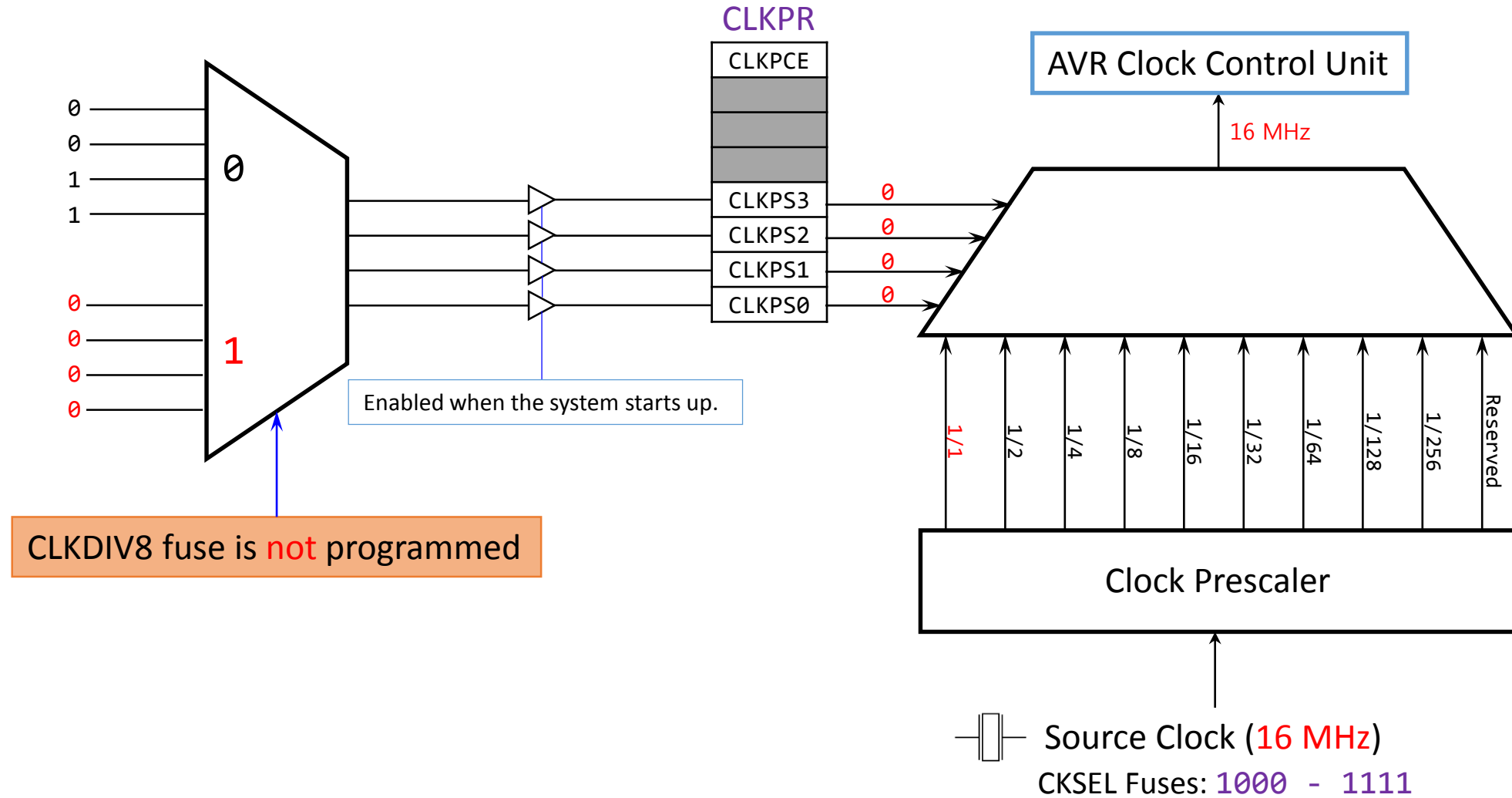
# System Clock Prescaler (2)



# System Clock Prescaler (Default Values)



# System Clock (Example)



# System Control and Reset

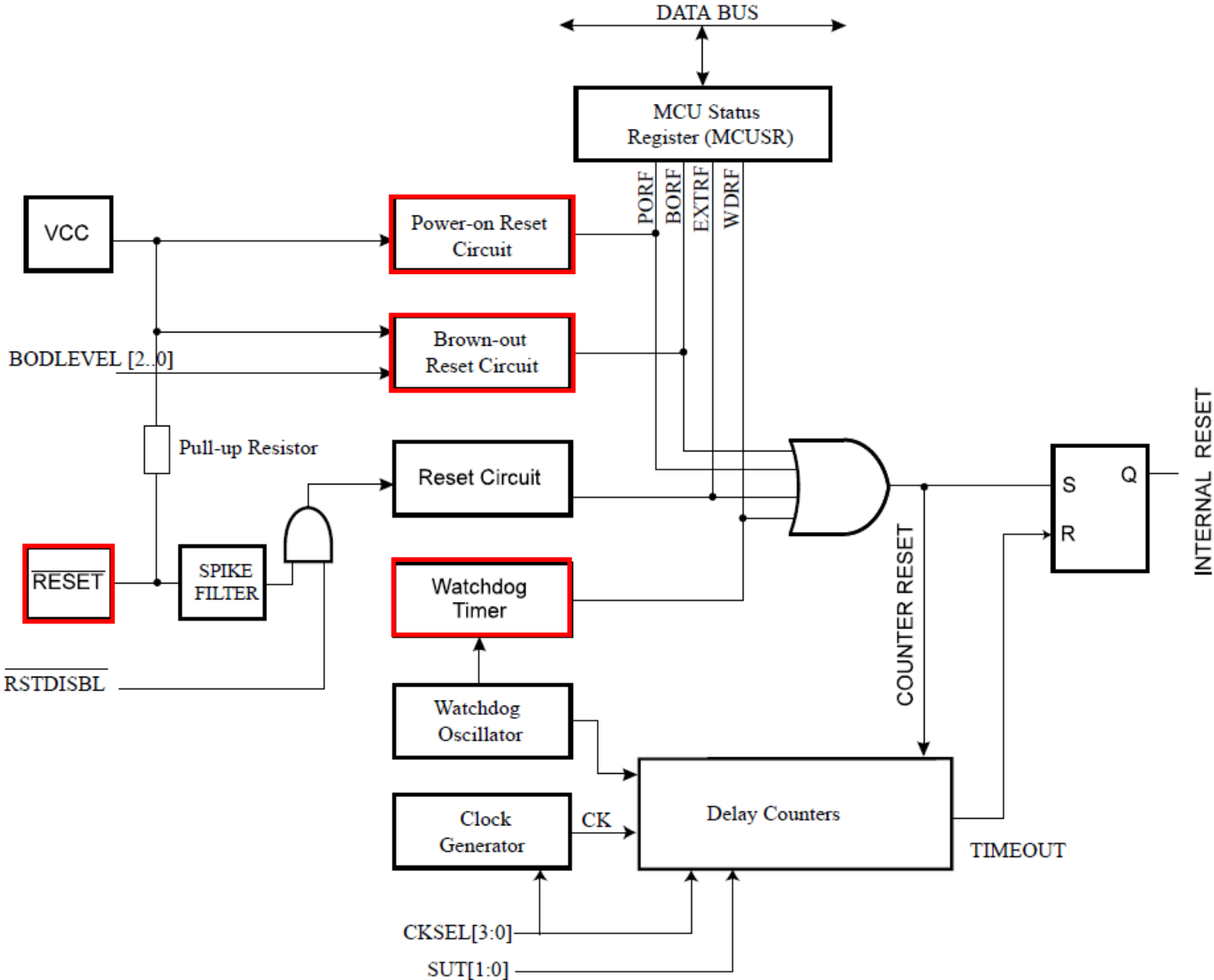


# Reset Sources (1)

---

- Power-on Reset
  - When the supply voltage is less than the Power-on Reset threshold (VPOT).
- External Reset
  - When a low level is present on the RESET pin for longer than the minimum pulse length.
- Watchdog System Reset
  - When the Watchdog Timer period expires and the Watchdog System Reset mode is enabled.
- Brown-out Reset
  - When the supply voltage VCC is less than the Brown-out Reset threshold (VBOT) and the Brown-out Detector is enabled.

# Reset Sources (2)



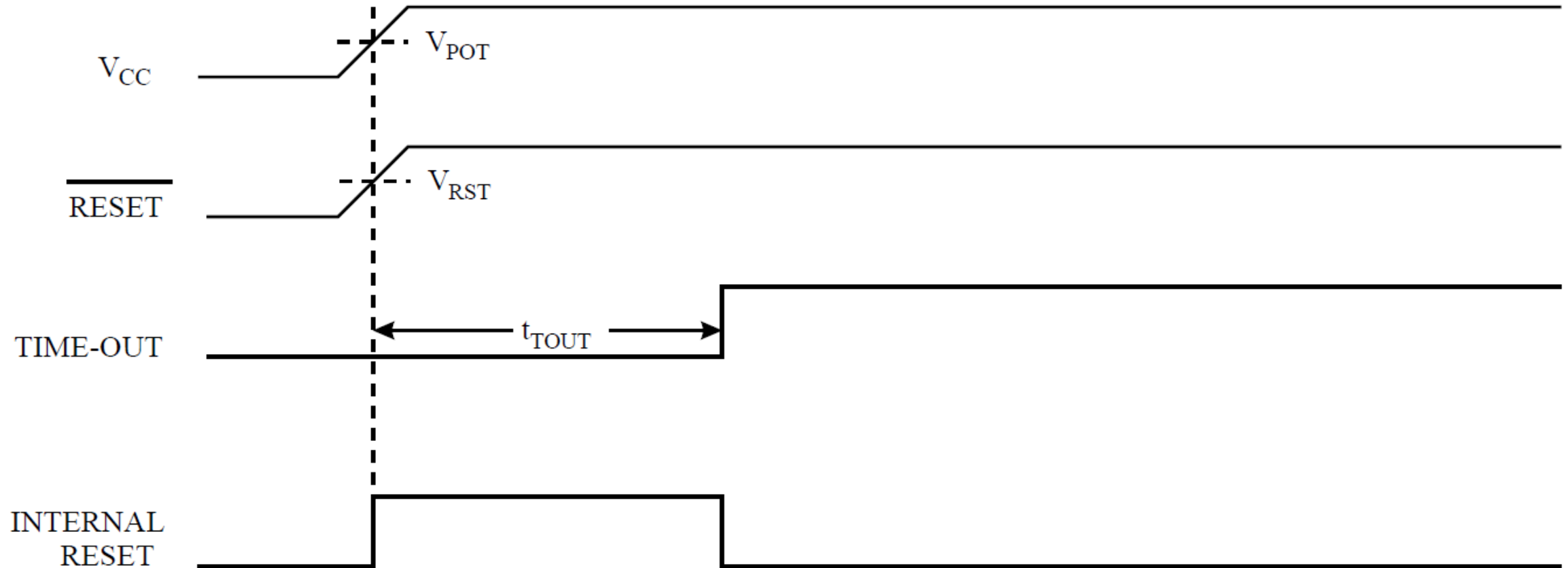
# Power-on Reset (1)

---

- A Power-on Reset (POR) pulse is generated by an On-chip detection circuit.
- The POR is activated whenever  $V_{CC}$  is below the detection level.
- The POR circuit can be used to trigger the start-up Reset, as well as to detect a failure in supply voltage.
- A Power-on Reset (POR) circuit ensures that the device is reset from Power-on.
- Reaching the Power-on Reset threshold voltage invokes the delay counter, which determines how long the device is kept in Reset after  $V_{CC}$  rise.
- The Reset signal is activated again, without any delay, when  $V_{CC}$  decreases below the detection level.

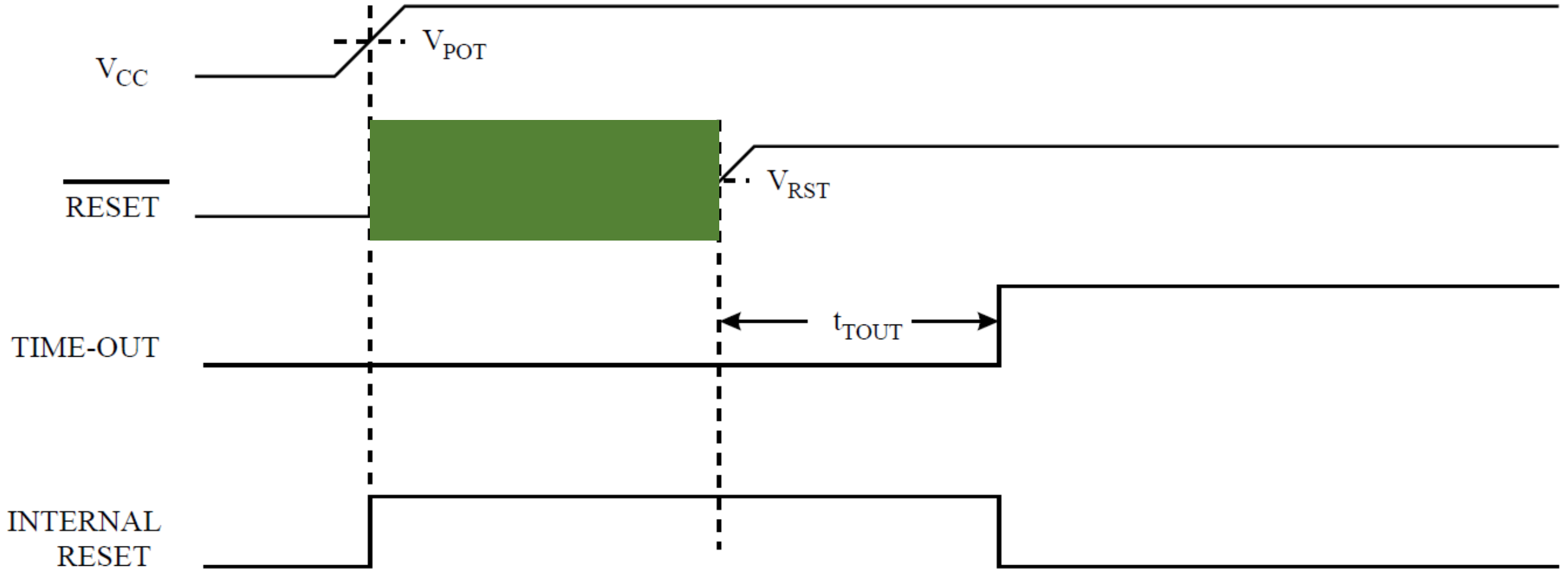
# Power-on Reset (2)

MCU Start-up,  $\overline{\text{RESET}}$  Tied to  $V_{CC}$



# Power-on Reset (3)

MCU Start-up,  $\overline{\text{RESET}}$  Extended Externally



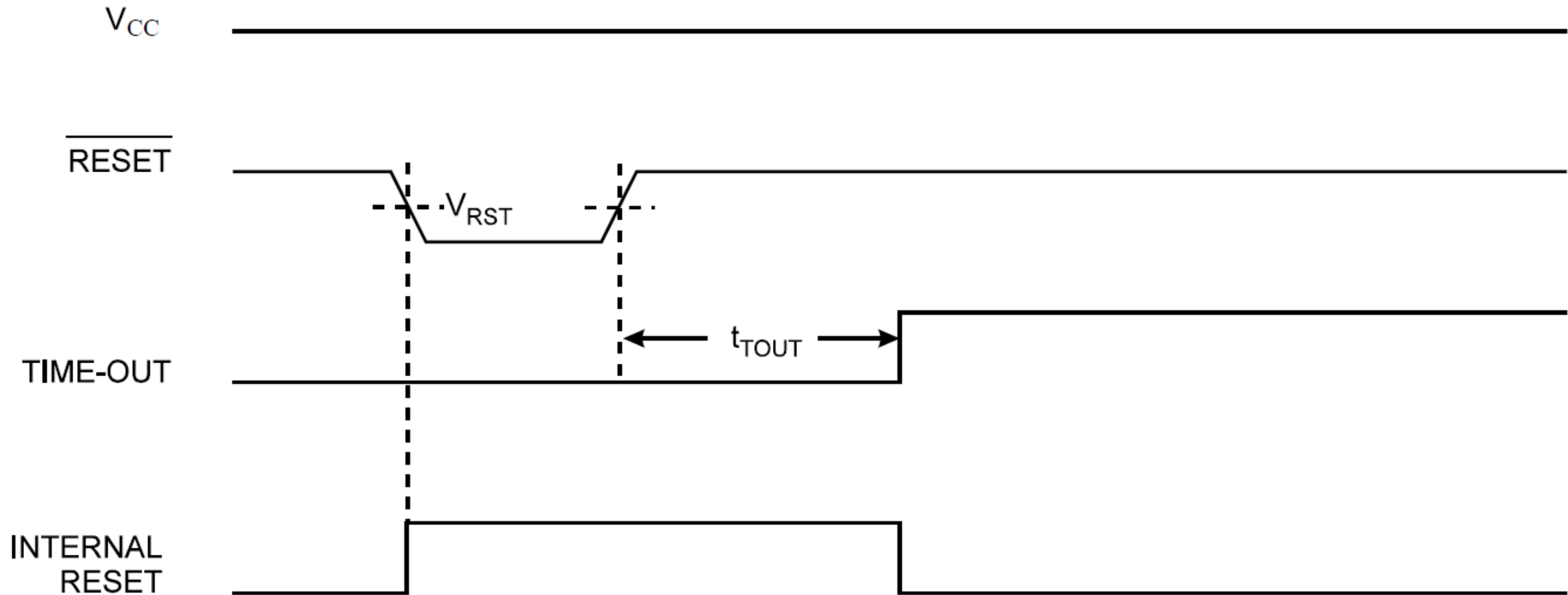
# External Reset (1)

---

- An External Reset is generated by a low level on the  $\overline{\text{RESET}}$  pin.
- Reset pulses longer than the minimum pulse width will generate a reset, even if the clock is not running.
- Shorter pulses are not guaranteed to generate a reset.
- When the applied signal reaches the Reset Threshold Voltage ( $V_{\text{RST}}$ ) on its positive edge, the delay counter starts the MCU after the Time-out period ( $t_{\text{TOU}}$ ) has expired.
- The External Reset can be disabled by the **RSTDISBL** fuse.

# External Reset (2)

## External Reset During Operation



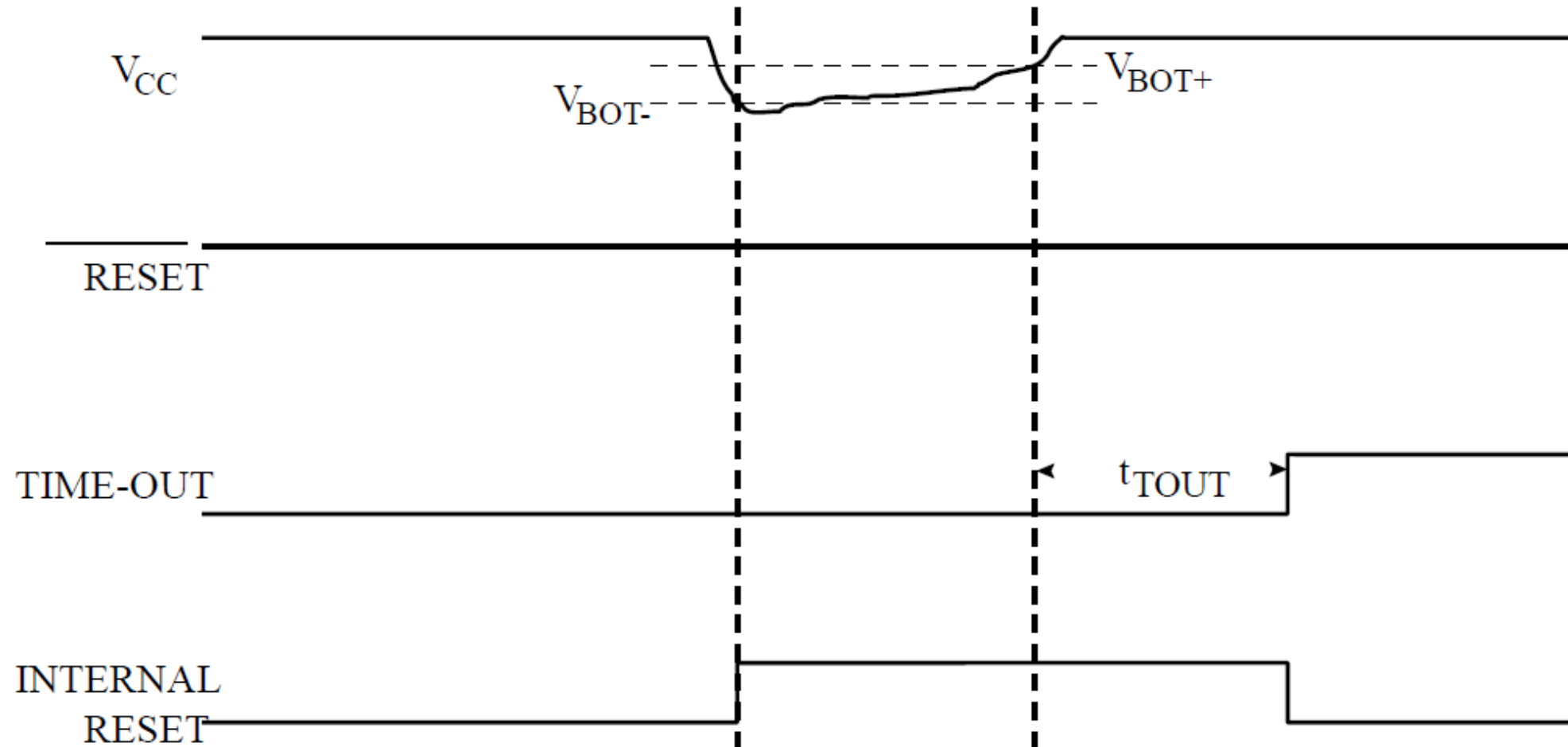
# Brown-out Detection (1)

- The device has an On-chip Brown-out Detection (BOD) circuit for monitoring the VCC level during operation by comparing it to a fixed trigger level.
- The trigger level for the BOD can be selected by the **BODLEVEL** Fuses.
- The trigger level has a hysteresis to ensure spike free Brown-out Detection.
  - $V_{\text{BOT+}} = V_{\text{BOT}} + V_{\text{HYST}}/2$  and  $V_{\text{BOT-}} = V_{\text{BOT}} - V_{\text{HYST}}/2$ .
- When the BOD is enabled, and  $V_{\text{CC}}$  decreases to a value below the trigger level ( $V_{\text{BOT-}}$ ), the Brown-out Reset is immediately activated.
- When  $V_{\text{CC}}$  increases above the trigger level ( $V_{\text{BOT+}}$ ), the delay counter starts the MCU after the Time-out period  $t_{\text{TOUT}}$  has expired.
- The BOD circuit will only detect a drop in  $V_{\text{CC}}$  if the voltage stays below the trigger level for longer than  $t_{\text{BOD}}$ .



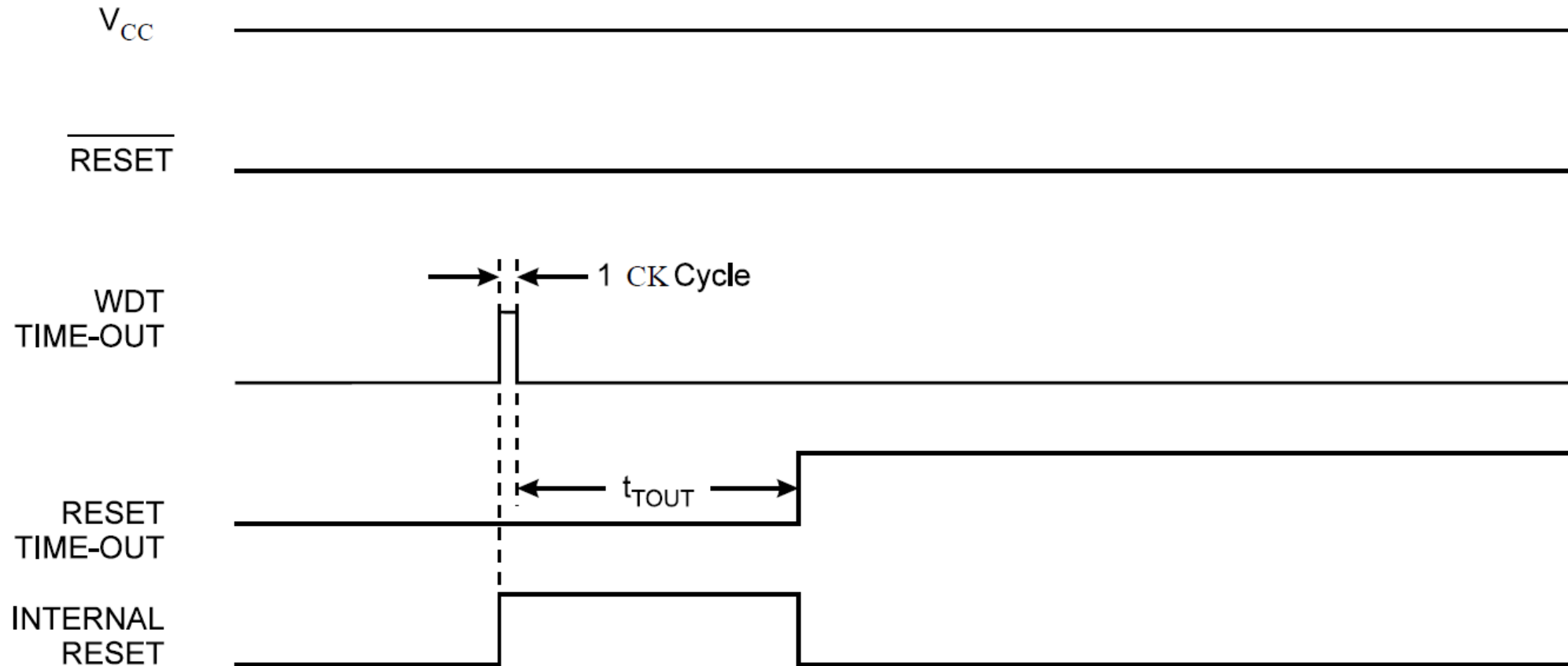
# Brown-out Detection (2)

## Brown-out Reset During Operation



# Watchdog System Reset

- When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration.
- On the falling edge of this pulse, the delay timer starts counting the Time-out period  $t_{TOUT}$ .



# Internal Voltage Reference

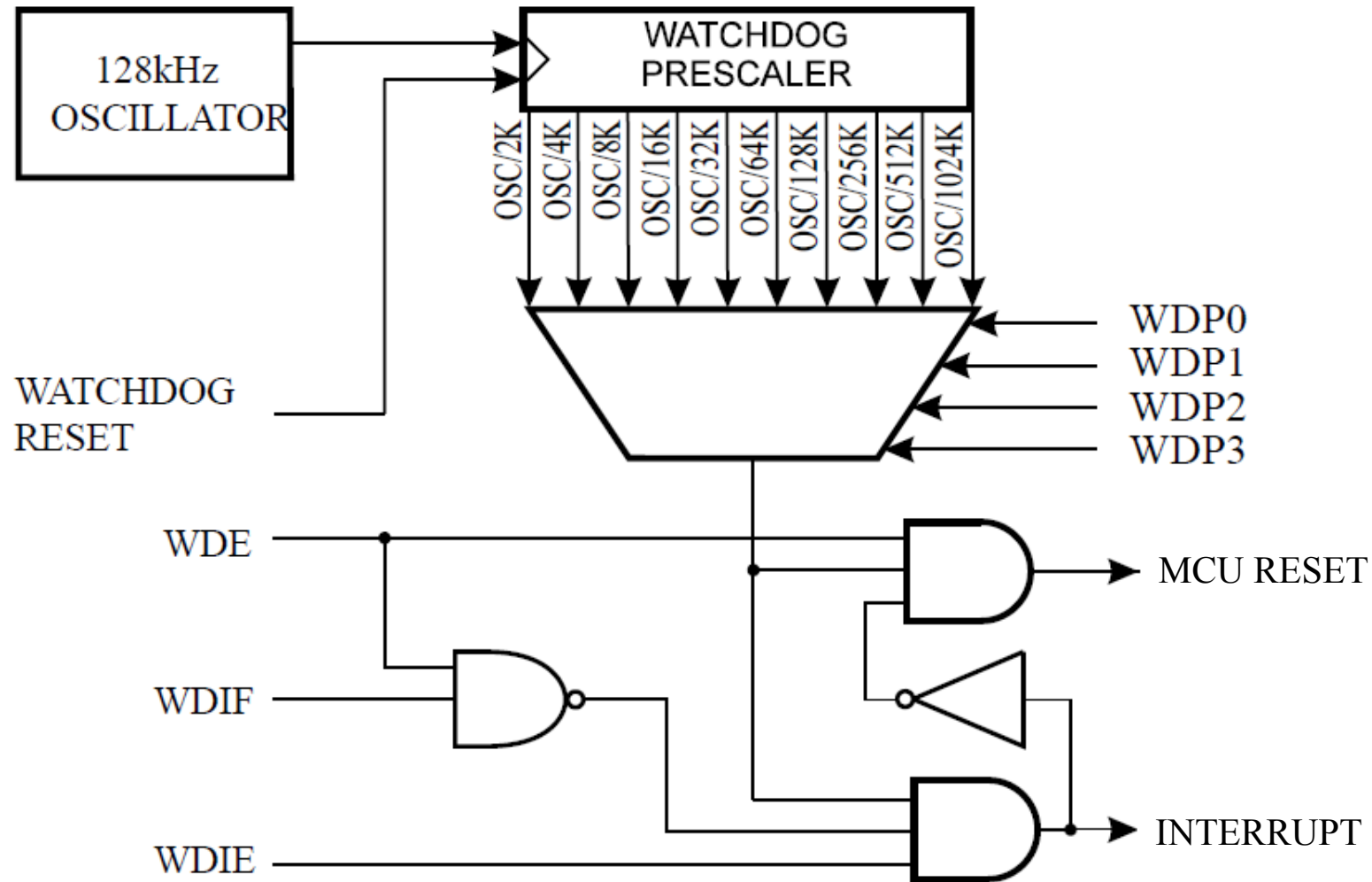
- The device features an internal bandgap reference.
- This reference is used for Brown-out Detection, Analog Comparator, ADC.
- The voltage reference has a start-up time that may influence the way it should be used.
- To save power, the reference is not always turned on. The reference is on during the following situations:
  1. When the BOD is enabled (by programming the [BODLEVEL \[2:0\]](#) Fuses).
  2. When the bandgap reference is connected to the Analog Comparator (by setting the ACBG bit in ACSR (ACSR.AC BG)).
  3. When the ADC is enabled.
- Thus, when the BOD is not enabled, after setting ACSR.AC BG or enabling the ADC, the user must always allow the reference to start up before the output from the Analog Comparator or ADC is used.
- To reduce power consumption in Power-Down mode, the user can avoid the three conditions above to ensure that the reference is turned off before entering Power-Down mode.

# Watchdog Timer (WDT) (1)

---

- The WDT is a timer counting cycles of a separate on-chip **128 kHz** oscillator.
- The WDT gives an interrupt or a system reset when the counter reaches a given time-out value.
- In normal operation mode, it is required that the system uses the Watchdog Timer Reset (**WDR**) instruction to restart the counter before the time-out value is reached.
- If the system doesn't restart the counter, an **interrupt** and/or **system reset** will be issued.

# Watchdog Timer (WDT) (2)



# Watchdog Timer (WDT) (3)

- Interrupt mode
  - The WDT gives an interrupt when the timer expires.
  - This interrupt can be used to wake the device from sleep-modes, and also as a general system timer.
  - One example is to limit the maximum time allowed for certain operations, giving an interrupt when the operation has run longer than expected.
- System Reset mode
  - The WDT gives a reset when the timer expires.
  - This is typically used to prevent system hang-up in case of runaway code.
- Interrupt and System Reset mode
  - Combined of the other two modes by first giving an interrupt and then switch to System Reset mode.
  - This mode will, for instance, allow a safe shutdown by saving critical parameters before a system reset.

# Watchdog Timer (WDT) (4)

## Watchdog Timer Configuration

WDTON Fuse	WDE Bit	WDIE Bit	Mode	Action on Timeout
1	0	0	Stopped	None
1	0	1	Interrupt Mode	Interrupt
1	1	0	System Reset Mode	System Reset
1	1	1	Interrupt and System Reset Mode	Interrupt, then go to System Reset Mode
0	X	X	System Reset Mode	System Reset

**WDTON** Fuse: Watchdog Timer Always On Fuse (Default: Unprogrammed, 1).

Refer to Table 33-5, p359 of the ATmega328PB datasheet.

# Watchdog Timer (WDT) (5)

## How to Use WDT

```
_disable_interrupt();    // cli

_watchdog_reset();      // wdr

/* Start timed sequence */
WDTCSR |= (1 << WDCE) | (1 << WDE);

/* Set new prescaler(time-out) value = 64K cycles (~0.5 s) */
WDTCSR = (1 << WDE) | (1 << WDP2) | (1 << WDP0);

_enable_interrupt();
```



# Watchdog Timer (WDT) (5)

## How to Turn Off WDT

```
_disable_interrupt();           // cli

_watchdog_reset();             // wdr

/* Clear WDRF in MCUSR first to clear WDE */
MCUSR &= ~(1 << WDRF);

/* Write logical one to WDCE and WDE */
/* Keep old prescaler setting to prevent unintentional time-out */
WDTCSR |= (1 << WDCE) | (1 << WDE);

/* Turn off WDT */
WDTCSR = 0x00;

_enable_interrupt();
```

# Watchdog Timer (WDT) (Example)

## How to Use WDT (AtmelStudio7)

```
// WDTON Fuse: Unprogrammed(1)

#define F_CPU 16000000UL // for _delay_ms()
#include <util/delay.h> // for _delay_ms()

#include <avr/io.h>
#include <avr/wdt.h> // for Watchdog Timer

int main(void)
{
    DDRB = (1 << DDB5); // Set PB5 as OUTPUT mode
    PORTB = (1 << PORTB5); // Set PB5 to HIGH level
    _delay_ms(2000); // This takes about 2.0 sec

    wdt_enable(WDTO_2S); // WDT will reset the system if WDR is not reset within 2 sec.
    //wdt_enable(WDTO_250MS); // WDT will reset the system if WDR is not reset within 250 msec.

    while (1)
    {
        wdt_reset(); // Reset WDT to avoid system reset.
        PINB |= (1 << PINB5); // Toggle PB5
        _delay_ms(500); // This takes about 500 msec
    }
}
```

wdt\_normal\_operation

# Register Description

# Oscillator Calibration Register (OSCCAL)

7	6	5	4	3	2	1	0
CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7:0	<p><b>Oscillator Calibration Value</b></p> <ul style="list-style-type: none"><li>• This register is used to adjust the <b>Calibrated Internal RC Oscillator</b> output frequency.</li><li>• A pre-programmed calibration value is automatically written to this register during chip reset, giving the Factory calibrated frequency. (Refer to p360 of the datasheet.)</li><li>• The application software can write this register to change the oscillator frequency.</li><li>• Note that this oscillator is used to time EEPROM and Flash write accesses, and these write times will be affected accordingly. The CAL7 bit determines the range of operation for the oscillator.<ul style="list-style-type: none"><li>✓ 0: gives the lowest frequency range</li><li>✓ 1: gives the highest frequency range.</li><li>✓ The two frequency ranges are overlapping</li></ul></li><li>• The CAL6..0 bits are used to tune the frequency within the selected range.<ul style="list-style-type: none"><li>✓ A setting of 0x00 gives the lowest frequency in that range, and a setting of 0x7F gives the highest frequency in the range.</li></ul></li></ul>
--------	---

# Clock Prescaler Register (CLKPR) (1)

7	6	5	4	3	2	1	0
CLKPCE				CLKPS3	CLKPS2	CLKPS1	CLKPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7	<p><b>CLKPCE: Clock Prescaler Change Enable</b></p> <ul style="list-style-type: none"> <li>The CLKPCE bit must be written to logic one to enable change of the CLKPS bits.</li> <li>The CLKPCE bit is only updated when the other bits in CLKPR are simultaneously written to zero.</li> <li>CLKPCE is cleared by hardware four cycles after it is written or when CLKPS bits are written.</li> <li>Rewriting the CLKPCE bit within this time-out period does neither extend the time-out period, nor clear the CLKPCE bit.</li> </ul>
Bit3:0	<p><b>CLKPSn: Clock Prescaler Select n [n = 3:0]</b></p> <ul style="list-style-type: none"> <li>These bits define the division factor between the selected clock source and the internal system clock.</li> <li>These bits can be written run-time to vary the clock frequency to suit the application requirements.</li> <li>The CKDIV8 Fuse determines the initial value of the CLKPS bits. <ul style="list-style-type: none"> <li>✓ If CKDIV8 is unprogrammed: the CLKPS bits will be reset to “0000”, giving a division factor of 1 at start up.</li> <li>✓ If CKDIV8 is <b>programmed</b>: the CLKPS bits are reset to “0011”, giving a division factor of 8 at start up.</li> <li>✓ The device is shipped with the CKDIV8 Fuse <b>programmed</b>.</li> </ul> </li> <li>Any value can be written to the CLKPS bits regardless of the CKDIV8 Fuse setting.</li> </ul>

# Clock Prescaler Register (CLKPR) (2)

## Clock Prescaler Select

CLKPS[3:0]	Clock Division Factor
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001-1111	Reserved

END