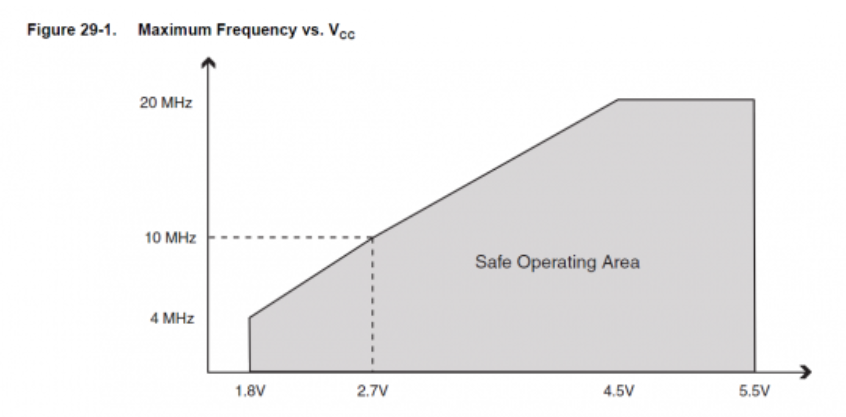
**Power-saving techniques for the Atmega328P processor.**

**Summary of methods**

Run the processor at a lower frequency 🡪 We could change the FOSC (8Mhz)

Run the processor at a lower voltage 🡪 With a LDO regulator we will run the end-device at 3.3V



Turn off unneeded internal modules in software (eg. SPI, I2C, Serial, ADC)

#include <avr/power.h> LIBRARY

**Enabling:**

power\_adc\_enable(); // ADC converter

power\_spi\_enable(); // SPI

power\_usart0\_enable(); // Serial (USART)

power\_timer0\_enable(); // Timer 0

power\_timer1\_enable(); // Timer 1

power\_timer2\_enable(); // Timer 2

power\_twi\_enable(); // TWI (I2C)

**Disabling:**

power\_adc\_disable(); // ADC converter

power\_spi\_disable(); // SPI

power\_usart0\_disable();// Serial (USART)

power\_timer0\_disable();// Timer 0

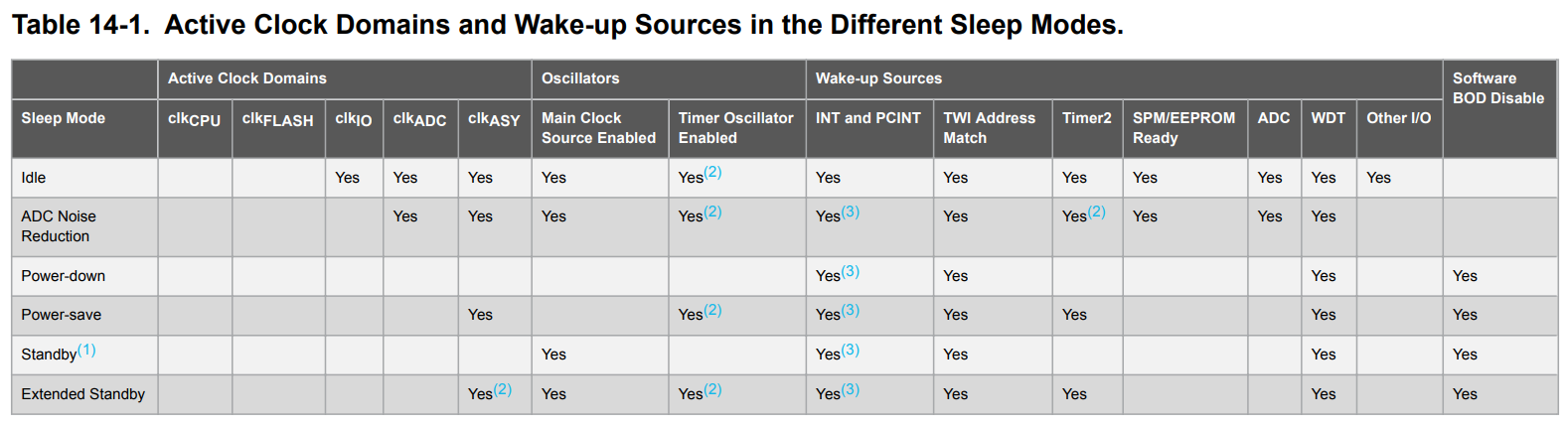
power\_timer1\_disable();// Timer 1

power\_timer2\_disable();// Timer 2

power\_twi\_disable(); // TWI (I2C)

**POWER DOWN SLEEP TO SAVE ENERGY**

The sleep modes differ in what parts remain active, by the sleep duration and the time needed to wake-up (wake-up period).



Library # include <avr/sleep.h>

Library #include <avr/wdt.h>