Arbeitskreis Hardware

Prof. Dr. Michael Rohs, Dipl.-Inform. Sven Kratz

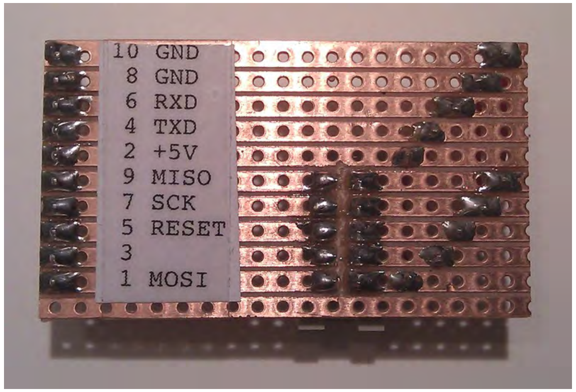
michael.rohs@ifi.lmu.de

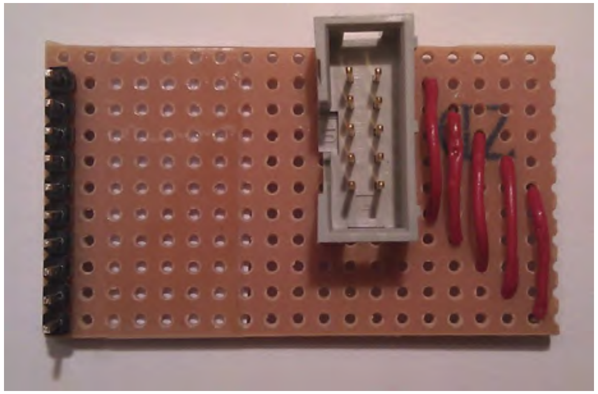
MHCI Lab, LMU München

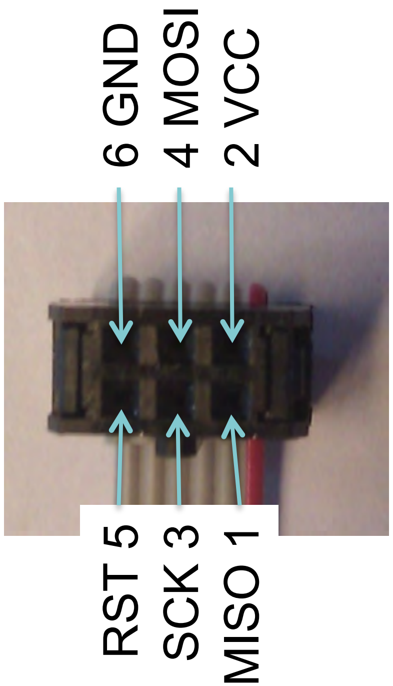
Schedule (preliminary)

Date	Topic (preliminary)
2.5.	Introduction to embedded interaction, microcontrollers, hardware & software tools
9.5.	*keine Veranstaltung (CHI)*
16.5.	soldering ISP adapter, AVR architecture, LED multiplexing/charlieplexing
23.5.	AVR architecture, AVR assembler, sensors: light, force, capacity, acceleration, etc.
30.5.	Electronics basics, PCB design & fabrication, EAGLE, 3D printing
6.6.	Displays (character LCDs, graphics LCDs), audio (speakers, amplification, op-amps)
13.6.	*keine Veranstaltung (Pfingsten)*
20.6.	I2C: interfacing to other chips (EEPROM, real-time clock, digital sensors)
27.6.	Actuation: stepper motors, servo motors
4.7.	Communication: fixed-frequency RF, ZigBee, Bluetooth
11.7.	Project
18.7.	Project
25.7.	Project

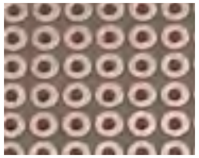
Mini Project: Programming Adapter







- strip board (Lochrasterplatine), 2.54 mm (0.1 inch) spacing

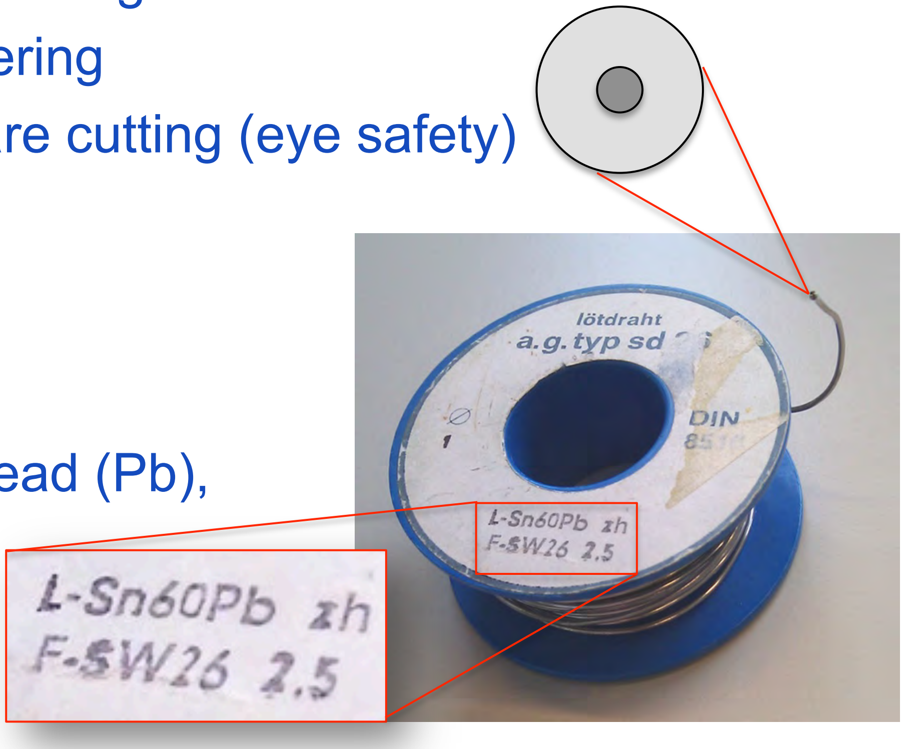


- 6-pin header for plugging into breadboard

- label for the six pins (2.54 mm spacing)
- two 3-pin headers for plug
- soldering wires to connect headers

Soldering

- Consult “Soldering is easy – Here's how to do it”
  mightyohm.com/files/soldercomic/FullSolderComic_20110409.pdf
- Safety tips
  1. Don’t touch the tip of the soldering iron
  2. Wash your hands after soldering
  3. Hold or cover the lead you are cutting (eye safety)
  4. Don’t breathe the smoke
- Solder is hollow and
  filled with flux (Flussmittel)
  - Example: 60% tin (Sn), 40% lead (Pb), flux: rosin (Kolophonium)
  - Lead-free solder in commercial applications

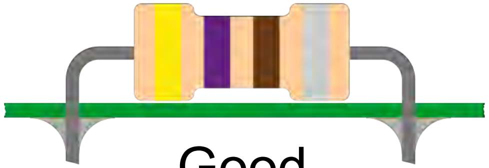


Soldering

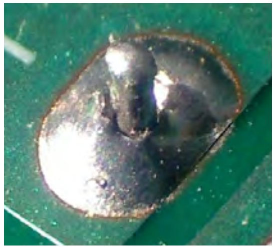
- Place components close to printed circuit board (PCB), leads should be short
  - Long wires lead introduce capacitances
  - Cut leads after soldering, small wire cutter with flat edge
- Placing components
  - Pay attention to component polarity!
  - Bend out leads to keep component in place
  - Excessive heat destroys components
- Soldering iron
  - Heat up, wait, clean on wet sponge
  - Put back iron into stand at all times
  - Remove power plug when done

Soldering

- Soldering (quickly)
  - Press tip of soldering iron against lead and PCB contact pad (lead and pad need to heat up) for 2s
  - Add 1-3mm of solder (very little) where tip touches lead and PCB contact pad, pull solder away
  - Wait 1s, pull soldering iron away
- Control (and fix bad connections)
  - Control connections with a multimeter
  - A good solder connection has a flat profile
  - Bad solder connections can be fixed by repeating the process
  - Remove excess solder using a vacuum pump or copper braid



Good

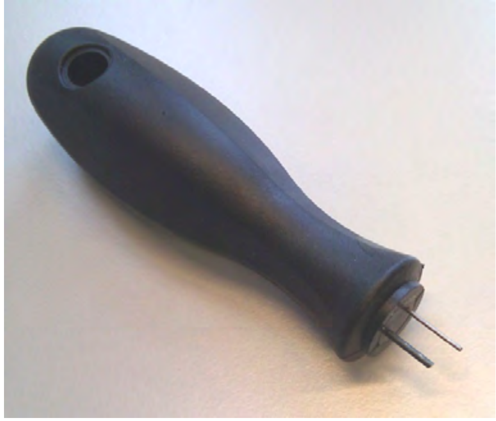


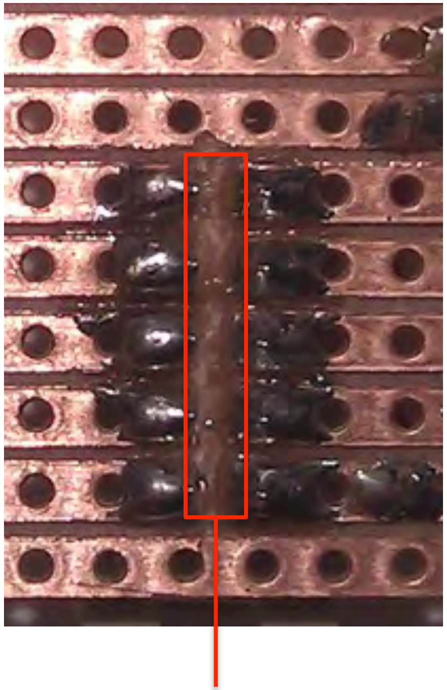
CC-BY-SA-3.0, GFDL

Bad

Handling Stripboards

- Remove copper using scratching tool



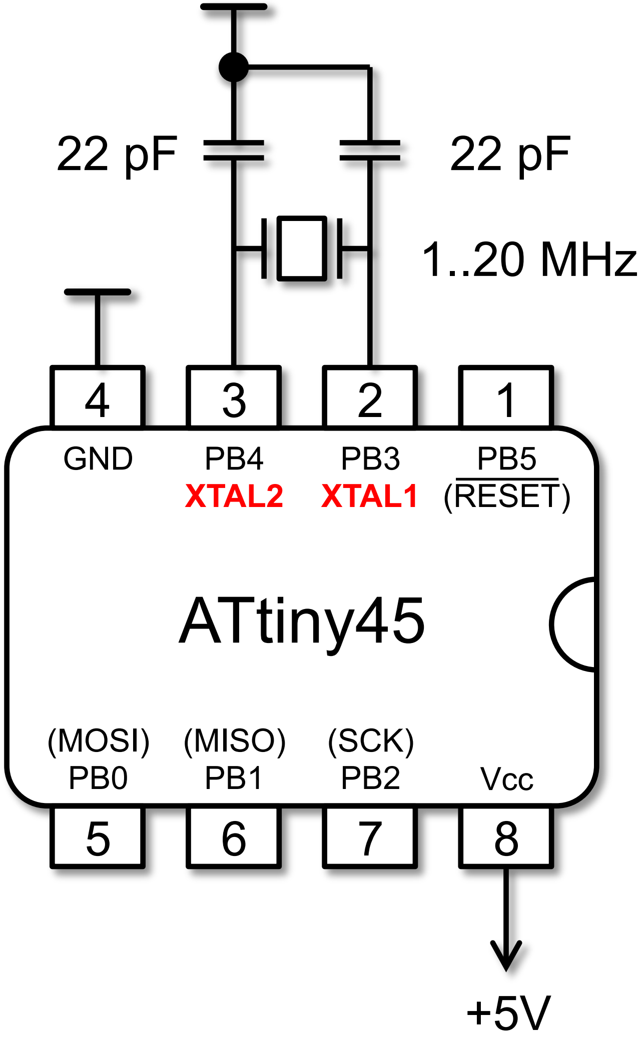


copper stripes

broken copper stripes,
copper removed with tool

- Cut PCB using a small saw
  – We will use milling machine to do this
- For soldering, fix with a “third hand” or bench vise

External Clock: Quartz Crystal Oscillators



- More precise than internal oscillators
- Quartz 1..20 MHz

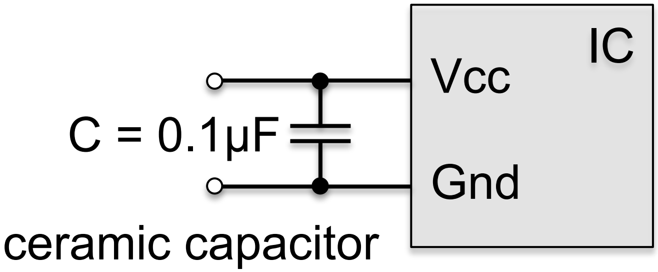
  ATtiny13 cannot use external quartz, ATtiny45 can
- Ceramic capacitors 12-22pF
- Place quartz and capacitors close to AVR pins
- Change CLKSEL fuse bits

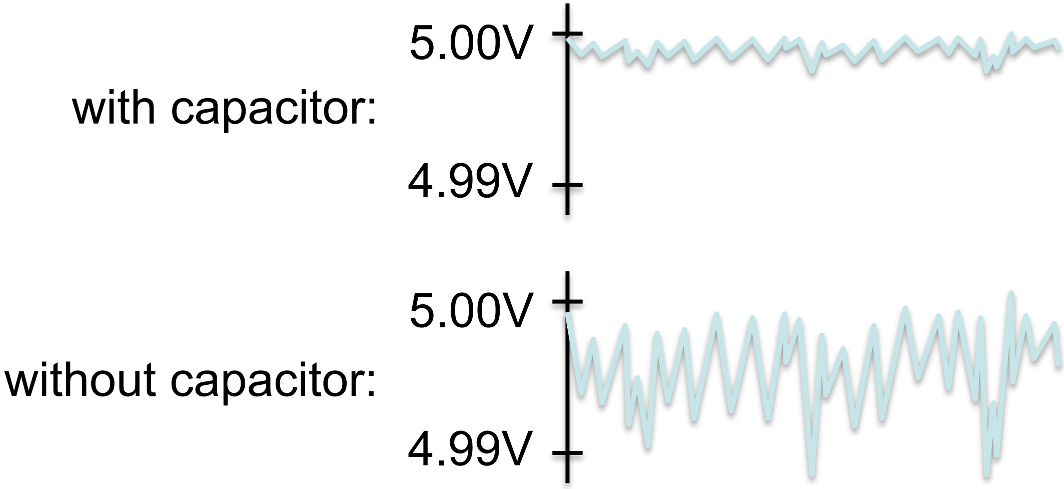
Stabilizing and Decoupling Capacitors (Stütz- und Abblockkondensatoren)

- Sudden fluctuation in current caused by
  - switching on/off LEDs, motors, relays causes
  - changing state of AVR pins
- Power supply alone cannot compensate for these
- Solution: stabilizing capacitors between VCC and ground
  - no current flow through them after charging (if voltage stable)
  - local energy source
  - filter spikes
- Higher frequency ripple requires smaller capacitor

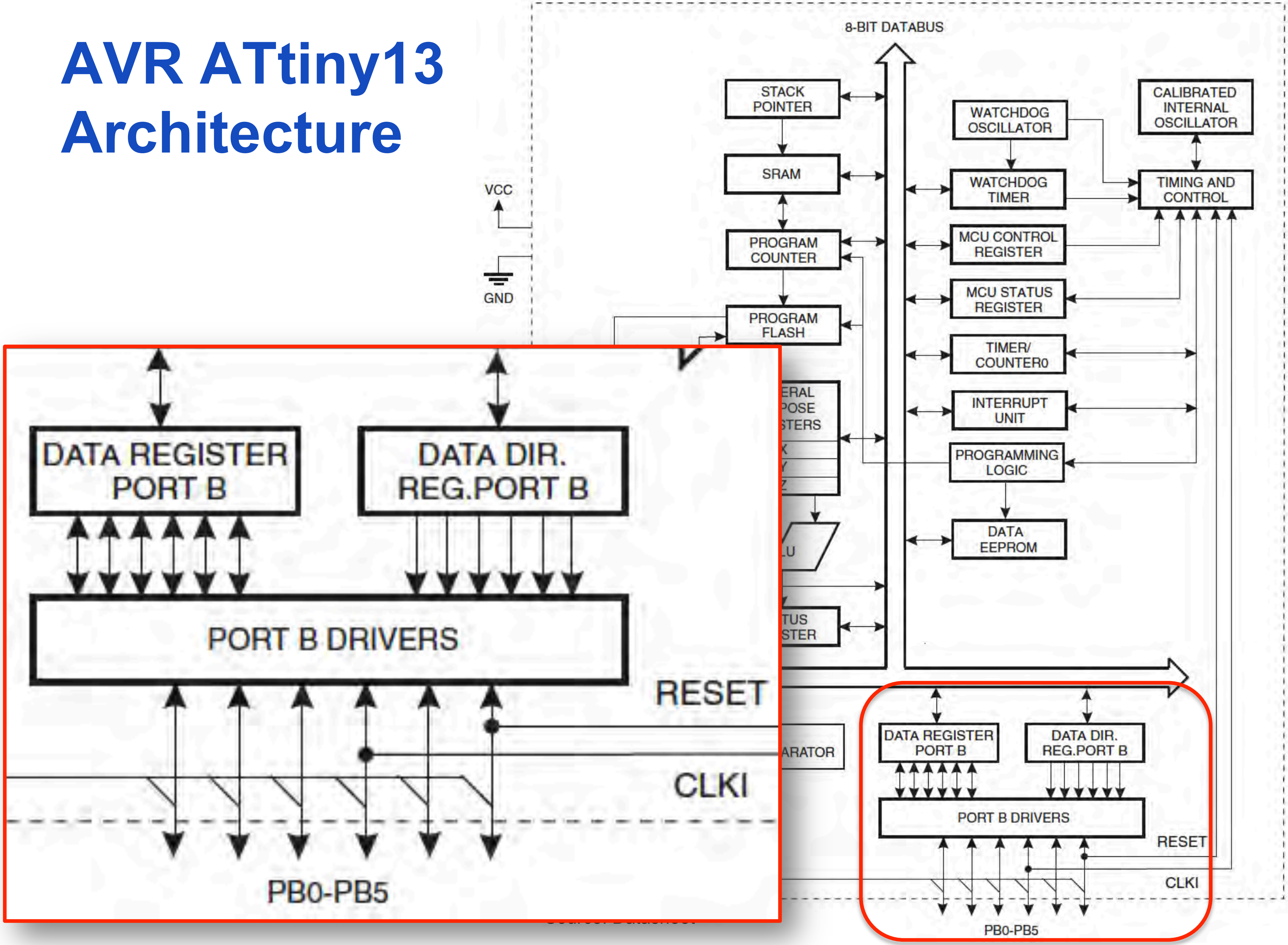
Stabilizing and Decoupling Capacitors (Stütz- und Abblockkondensatoren)

- Larger capacitors (10µF..100µF) as a local energy source
  - Electrolytic, high capacitance, high leakage, not suited for high frequencies, polar (!)
- Smaller capacitors (10nF..100nF) for filtering spikes
  - Ceramic, low capacitances, suited for high frequencies
- Place capacitors between GND and VCC of ICs
  - Place close to IC pins

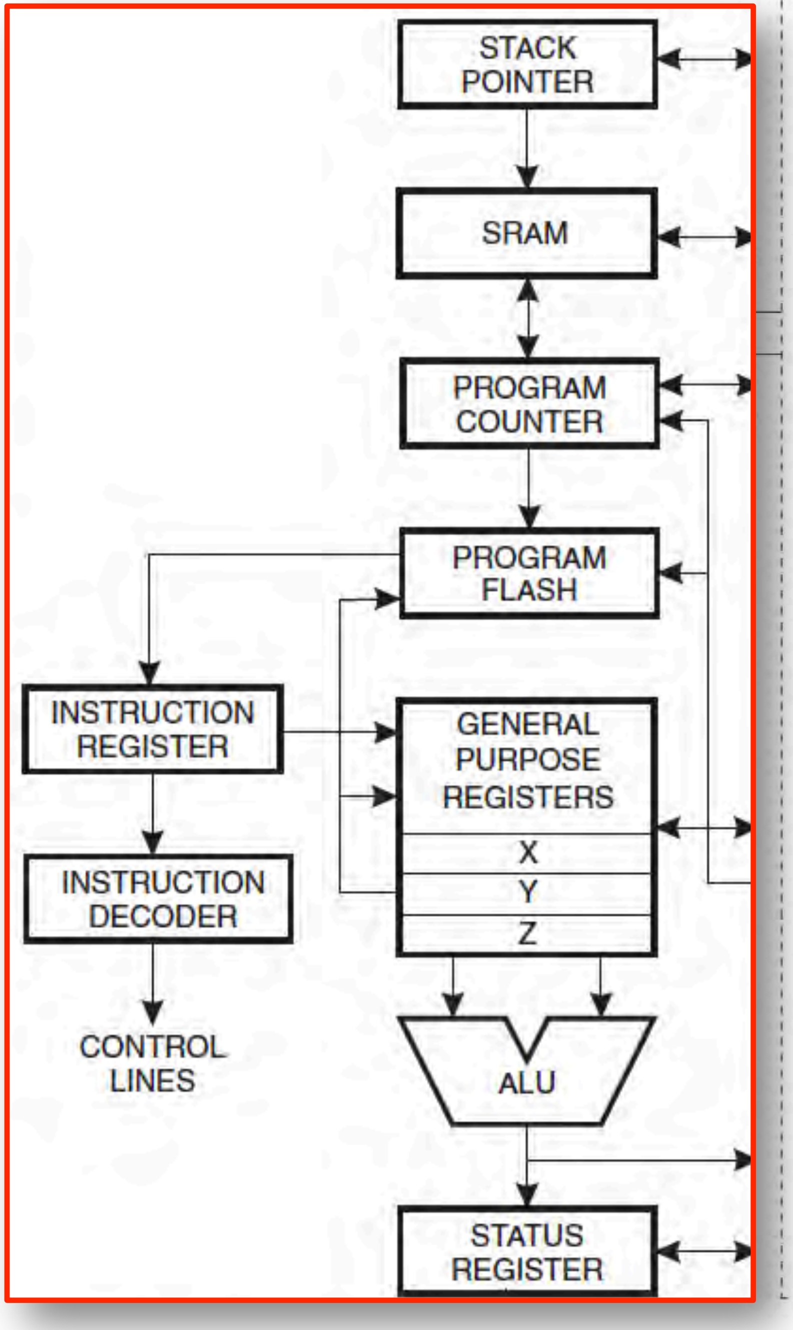


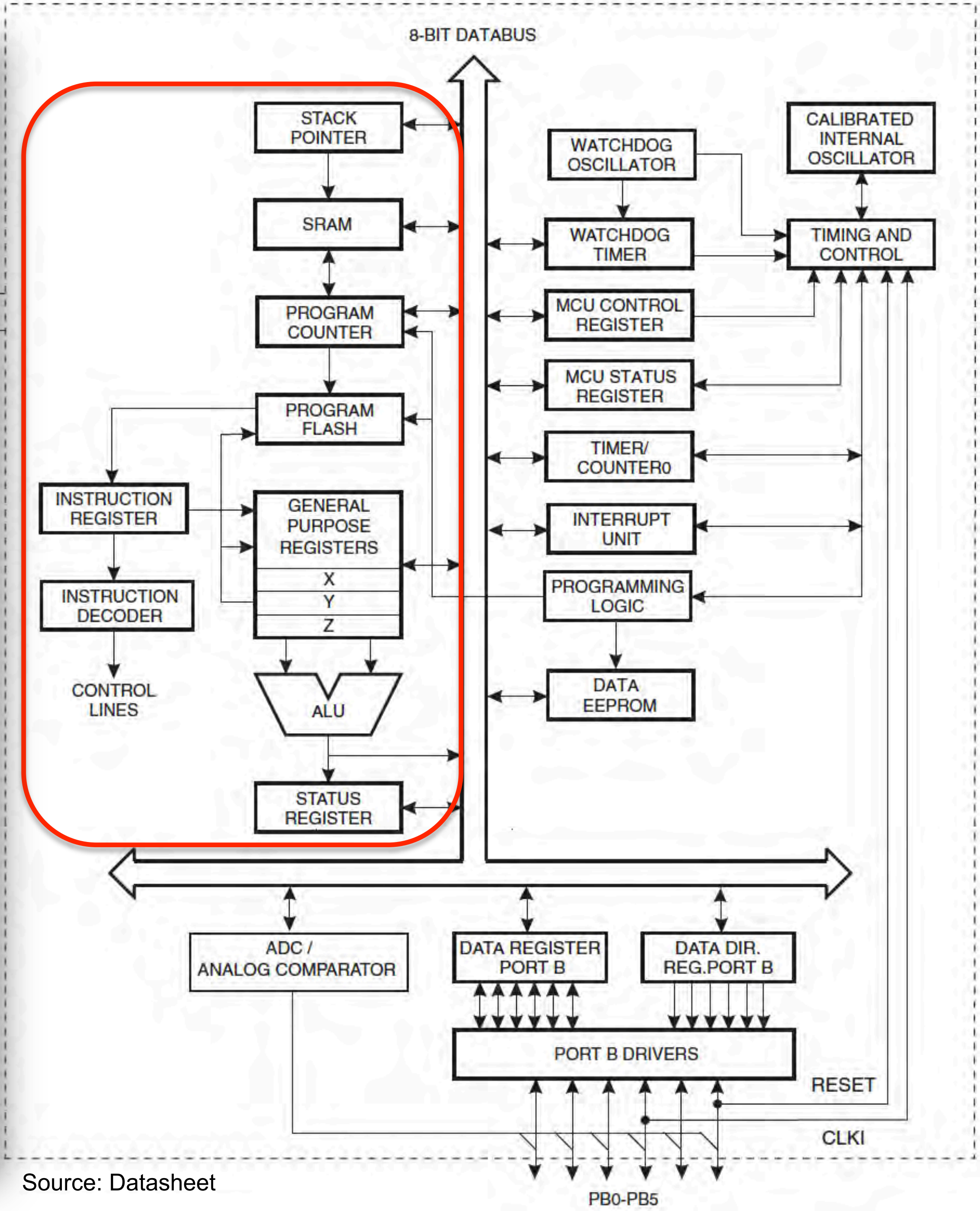


AVR ATtiny13 Architecture



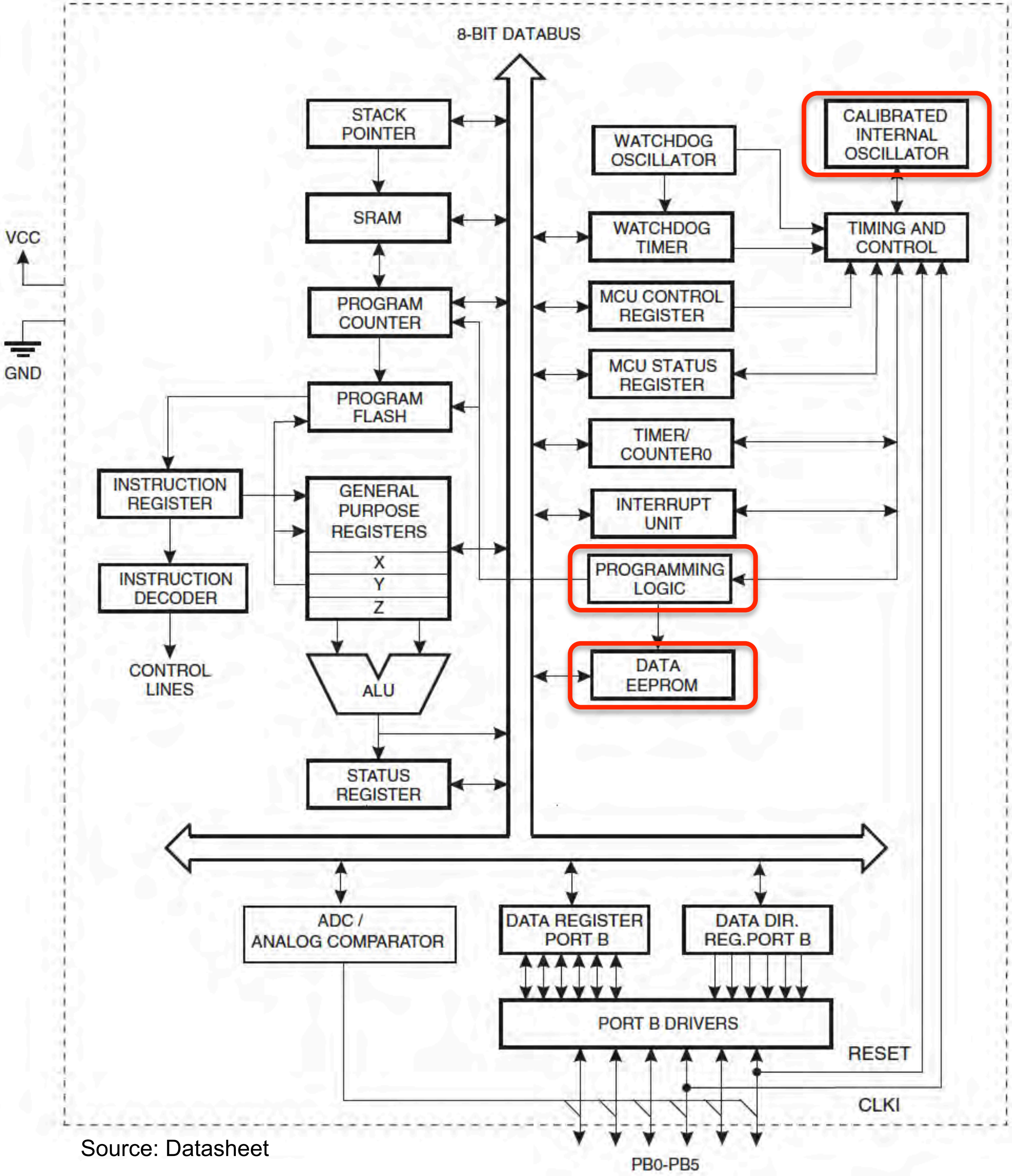
Source: Datasheet





Source: Datasheet

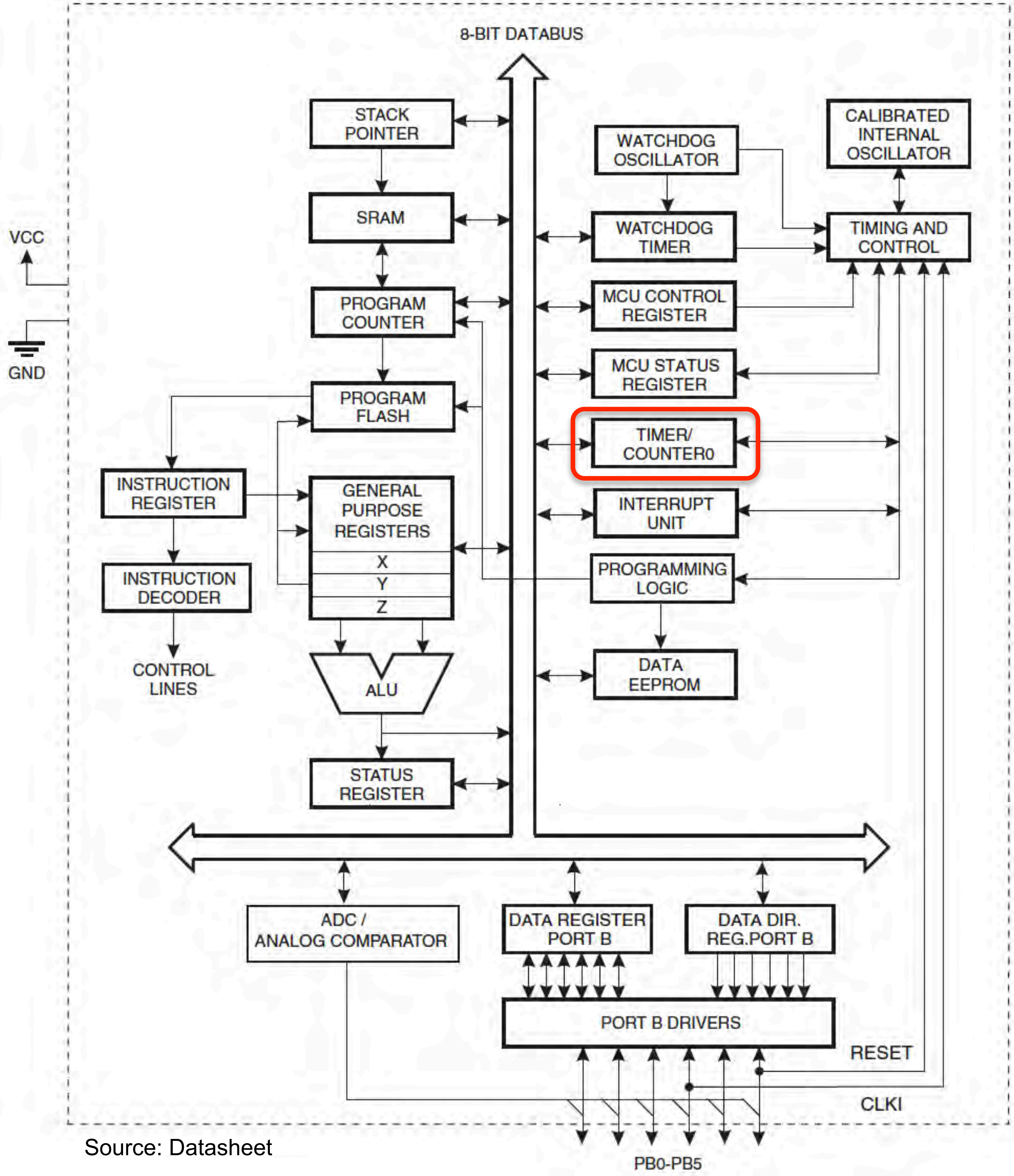
AVR ATtiny13 Architecture



Source: Datasheet

AVR ATtiny13 Architecture

- Timer / Counter 0

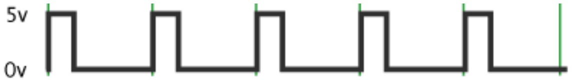


Source: Datasheet

AVR Timers

- Tasks: Generating periodic events, PWM
  - Pulse-width modulation (PWM) on I/O pins
  - Timers can generate interrupts

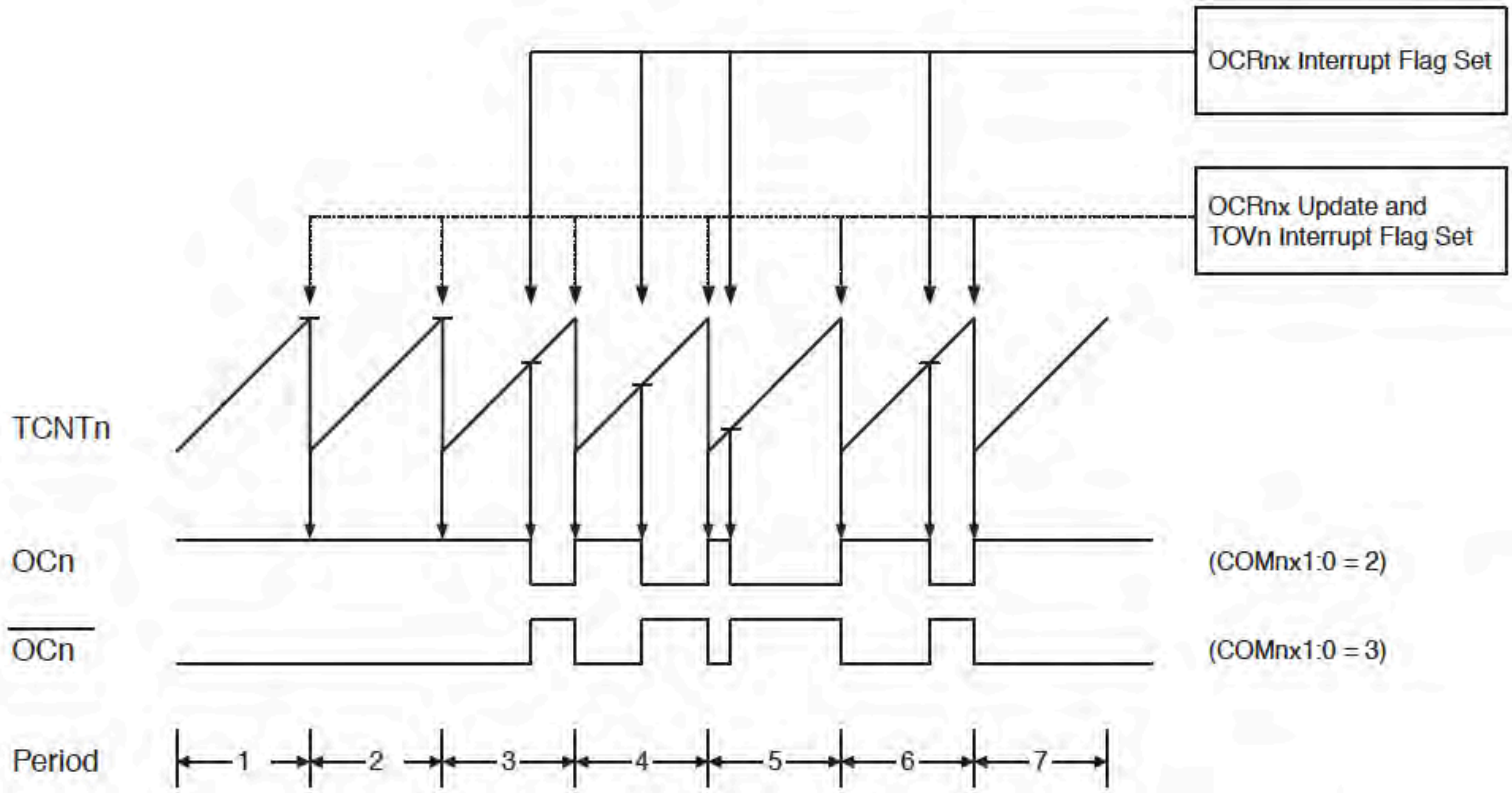
PWM with 25% duty cycle



- Synchronous clock source: device clock
  - divided by prescaler, if necessary
- Asynchronous clock source: external clock or PLL
  - PLL = phase-locked loop
- Modes
  - normal: count to 2^8-1 = 255, generate interrupt, continue at 0
  - clear-timer-on-compare: count to value,
  - fast PWM: single slope, count to 255 or value, set/clear pin on match
  - phase-correct PWM: dual slope, 50% speed of fast PWM

Fast PWM (single-slope operation)

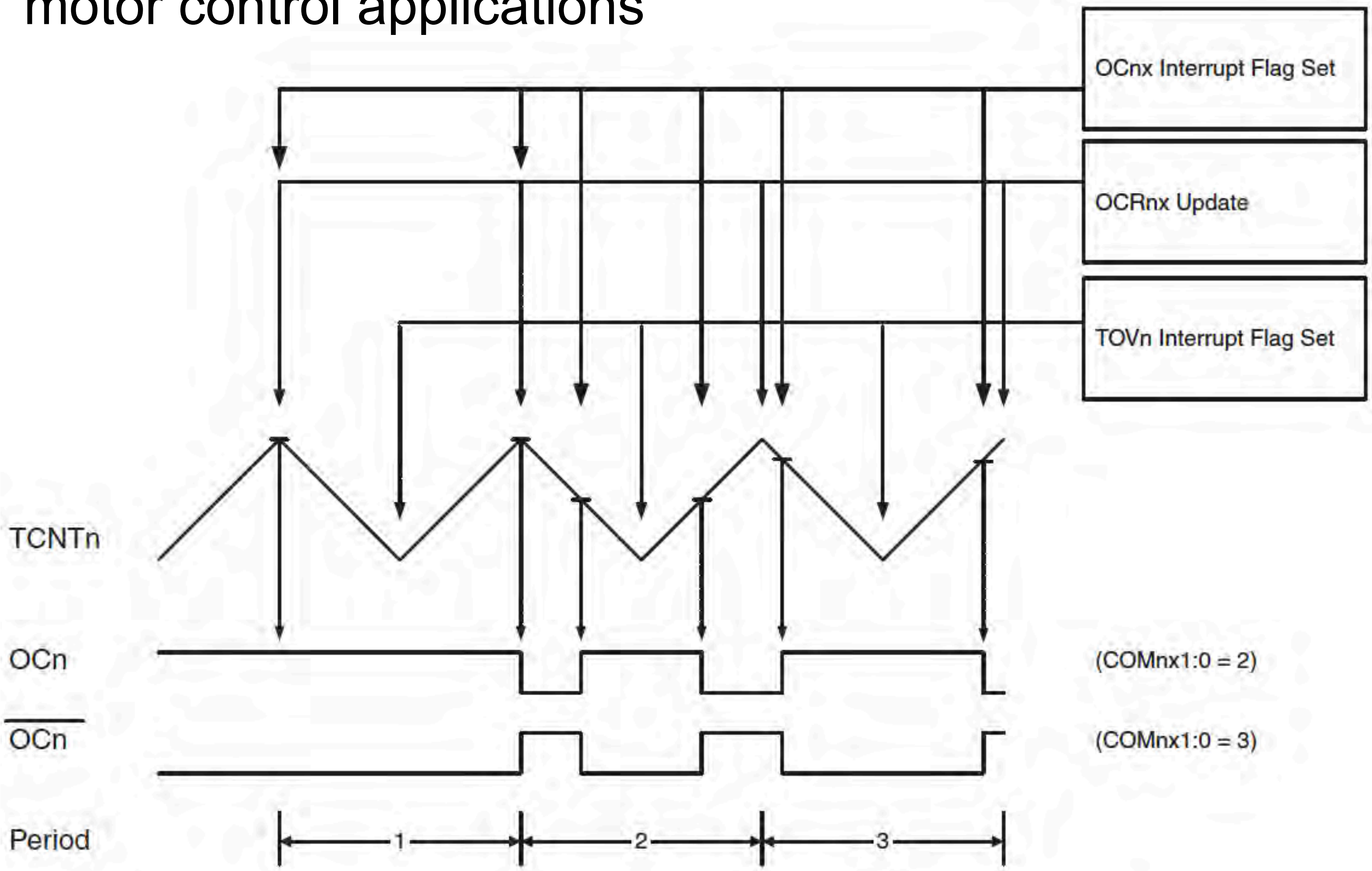
- power regulation, rectification, and DAC applications



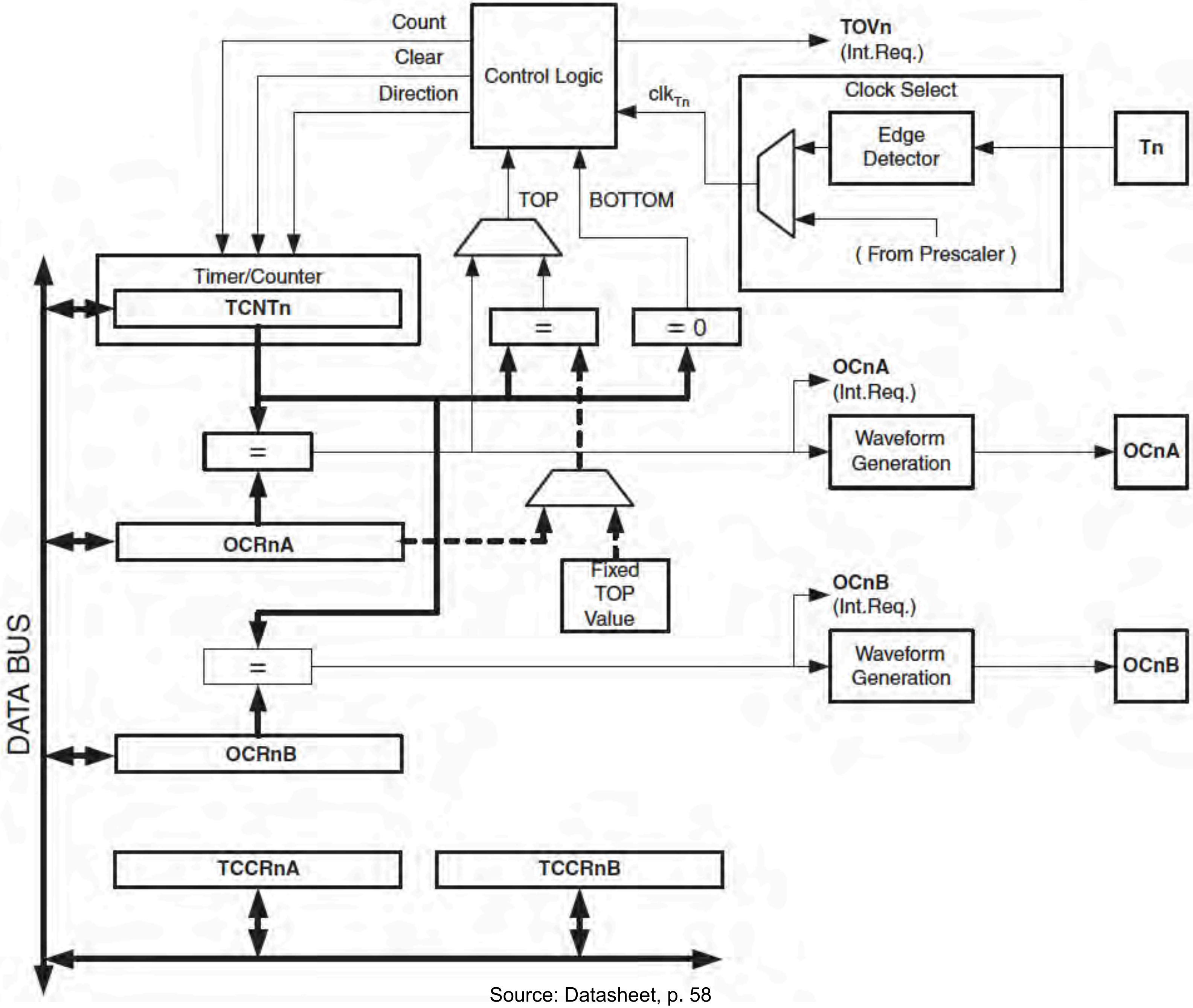
Source: Datasheet

Phase-Correct PWM (dual-slope operation)

- motor control applications



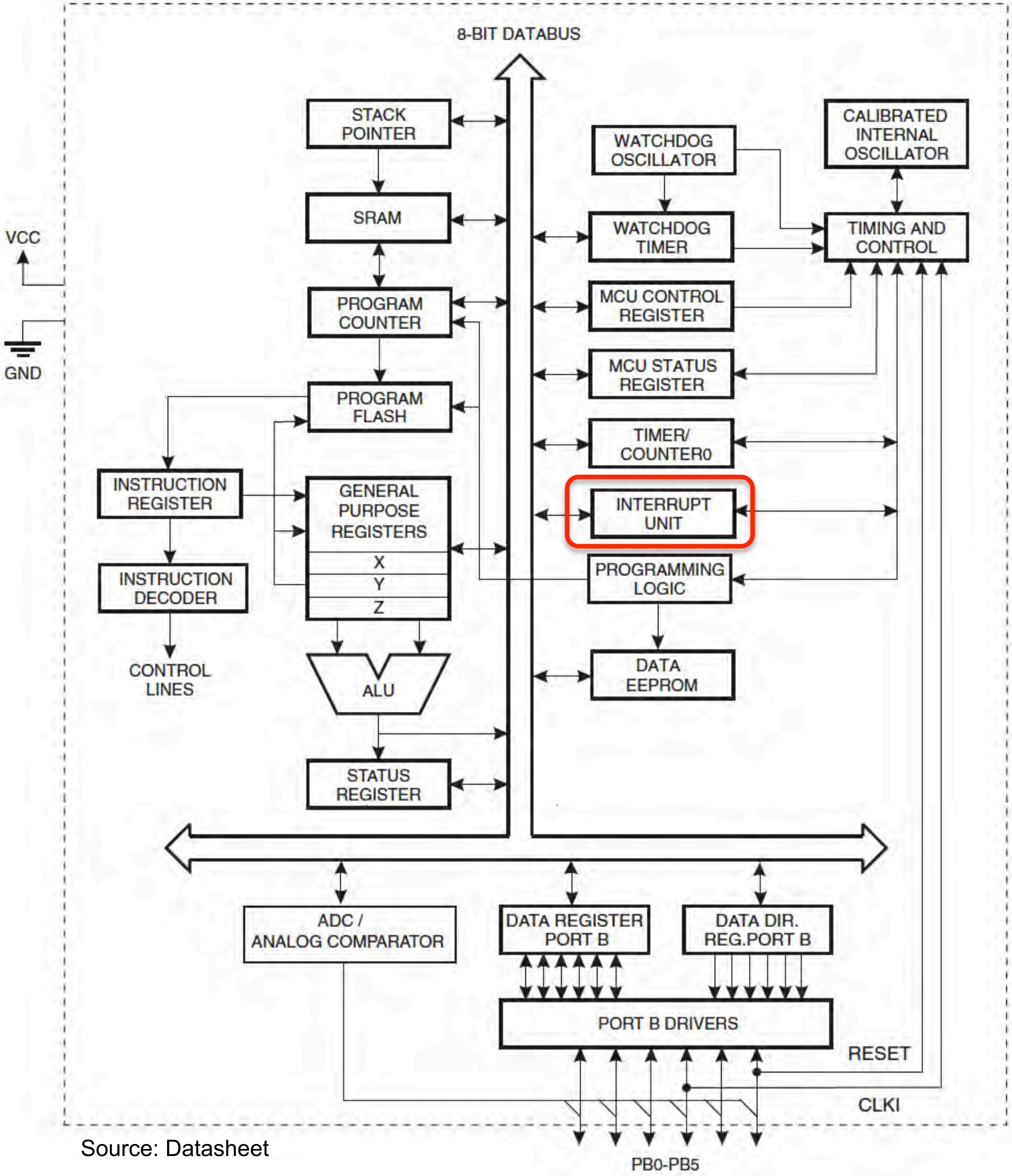
Source: Datasheet



Source: Datasheet, p. 58

AVR ATtiny13 Architecture

- Interrupt Unit



Source: Datasheet

AVR Interrupts

- Interrupt normal execution, jump to interrupt service routine (ISR), resume normal execution
- Interrupt vectors at start of program memory space (Flash)
  - intr. vectors = jump instructions to interrupt services routines
  - Lower address = higher priority
- Memory layout

```
0x0000 rjmp RESET ; Reset Handler
0x0001 rjmp EXT_INT0 ; IRQ0 Handler
0x0002 rjmp PCINT0 ; PCINT0 Handler
0x0003 rjmp TIM0_OVF ; Timer0 Overflow Handler
0x0004 rjmp EE_RDY ; EEPROM Ready Handler
0x0005 rjmp ANA_COMP ; Analog Comparator Handler
0x0006 rjmp TIM0_COMPA ; Timer0 CompareA Handler
0x0007 rjmp TIM0_COMPB ; Timer0 CompareB Handler
0x0008 rjmp WATCHDOG ; Watchdog Interrupt Handler
0x0009 rjmp ADC ; ADC Conversion Handler

0x000A RESET: ldi r16, low(RAMEND); Start
0x000B out SPL,r16 ; Stack Pointer to RAM end
0x000C sei ; Enable interrupts
0x000D <instr> xxx
```

AVR Interrupts

- Interrupts must be enabled
  - Various registers enable/disable interrupts
  - Global interrupt enable bits, individual interrupt enable bits
  - MCUCR, GIMSK and PCMSK registers control interrupts
  - GIFR shows interrupt states
- External interrupts
  - INT0 pin or PCINT5..0 pins
  - even if configured as outputs (software interrupt)
  - pin change interrupts: trigger if PCINT5..0 pin toggles
  - can wake the part from sleep modes
  - level interrupt: triggers as long as INT0 pin low
- Timers, ADC, analog comparator, etc. generate interrupts

LEDs

- LEDs are quickly switchable, power-efficient light sources, emitting light over the entire visible spectrum
  - (and beyond, e.g. IR LEDs)
- Anode (long lead) goes to positive potential
- Cathode (short lead) goes to negative potential
- LEDs operate like voltage-controlled switches
  - little current below turn-on voltage (silicon: 0.7V)
  - very high current above → LEDs need current-limiting resistors
- LEDs are diodes: no current in reverse direction
- Typical forward current: 20mA, typical forward voltage 2V

Exercise: Controlling LEDs

- Control brightness of two LEDs with PWM
  - Attach 2 LEDs to an ATtiny13
  - Periodically
    - Over 2s: increase brightness of LED_1 from dark to maximum, decrease brightness LED_2 from maximum to dark
    - Over 2s: increase brightness of LED_2 from dark to maximum, decrease brightness LED_1 from maximum to dark
    - Over 2s: no change in LED brightness
  - Use timers and interrupts as needed
- Control brightness of an LED using PWM and two buttons
  - Attach LED to an ATtiny13
  - While $button_1$ is pressed, slowly increase brightness
  - While $button_2$ is pressed, slowly decrease brightness