Media Technologie
Hardware Engineering

Vincent Bakker

Les 4

If Microsoft made cars..

1. For no reason whatsoever, your car would crash twice a day.

2. Every time they repainted the lines in the road, you would have to buy a new car.

3. Occasionally your car would die on the freeway for no reason. You would have to pull over to the side of the road, close all of the windows, shut off the car, restart it, and reopen the windows before you could continue. For some reason you would simply accept this.

4. Occasionally, executing a maneuver such as a left turn would cause your car to shut down and refuse to restart, in which case you would have to reinstall the engine.

5. Macintosh would make a car that was powered by the sun, was reliable, five times as fast and twice as easy to drive, but would run on only five percent of the roads.

hogeschool Leiden

Les 4

- Een belangrijk toepassingsgebied van embedded is real-time systems

- Timing is alles!!

  - PWM & Multiplexing
  - Communicatie
  - Human Interaction
  - Animatie & POV
  - Beveiligingscircuits

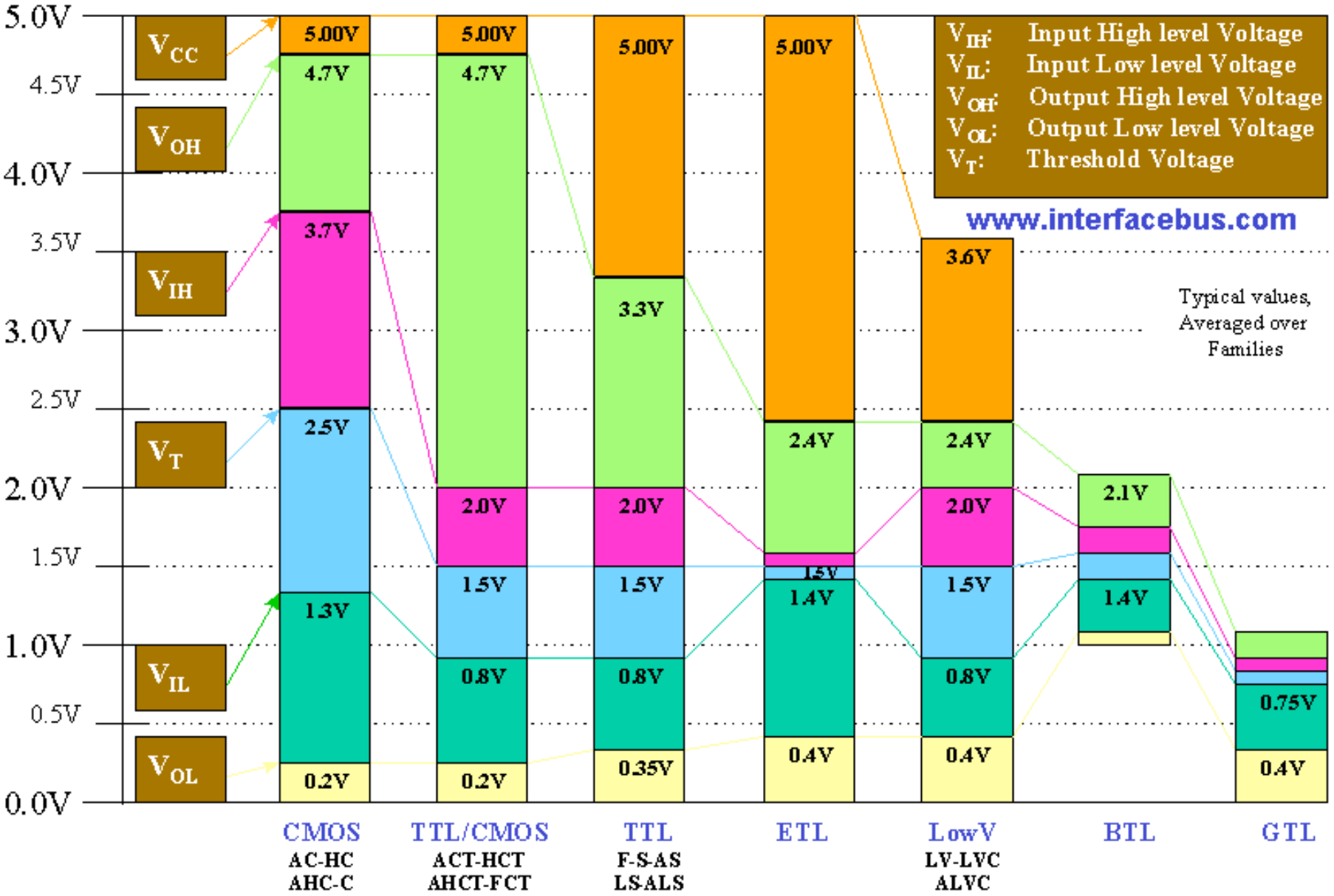
Les 4

- Doelstellingen Les 4
  - Kunnen benoemen welke mogelijkheden er zijn qua timing van microcontrollers
  - Kunnen uitleggen wat een ISR is
  - Kunnen uitleggen wat het nut is van Interrupts
  - Kunnen benoemen welke typen interrupts er zijn
  - Kunnen uitleggen hoe timers gebruikt kunnen worden
  - Een ISR kunnen programmeren

Les 4 Onderwerpen en Planning

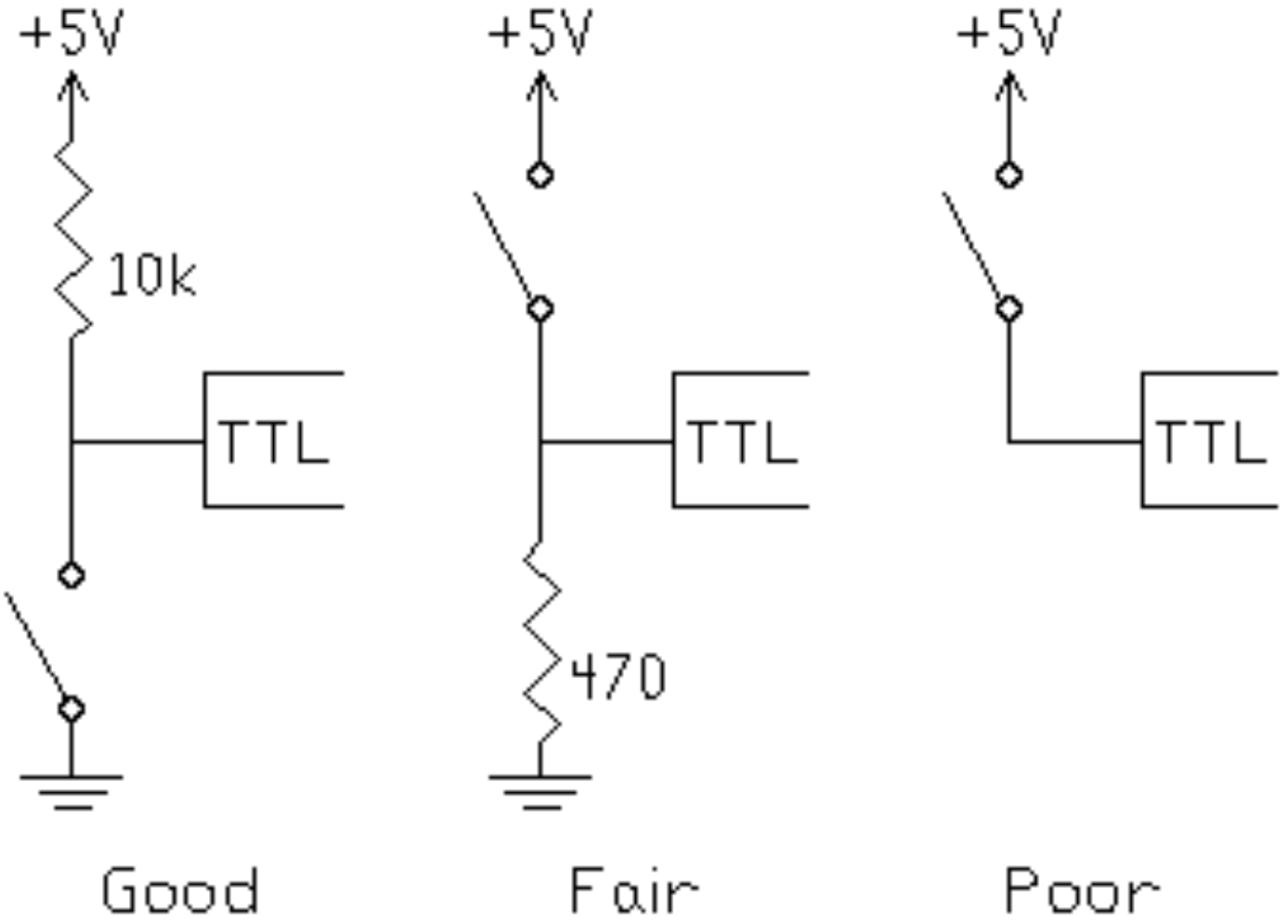
- Vragen les 3 en herhaling (5 min)
- Clock en timing (15 min)
  - Watchdog timer
- Clock Oefening (20 min)
- Pauze (15 min)
- Interrupts theorie (20 min)
- ISR implementatie oefening (20 min)
- Evaluatie (5 min)

hogeschool Leiden



hogeschool Leiden

Schakelaar 2



hogeschool Leiden

Interne Pull-up

- Vraag: niveau op PB5

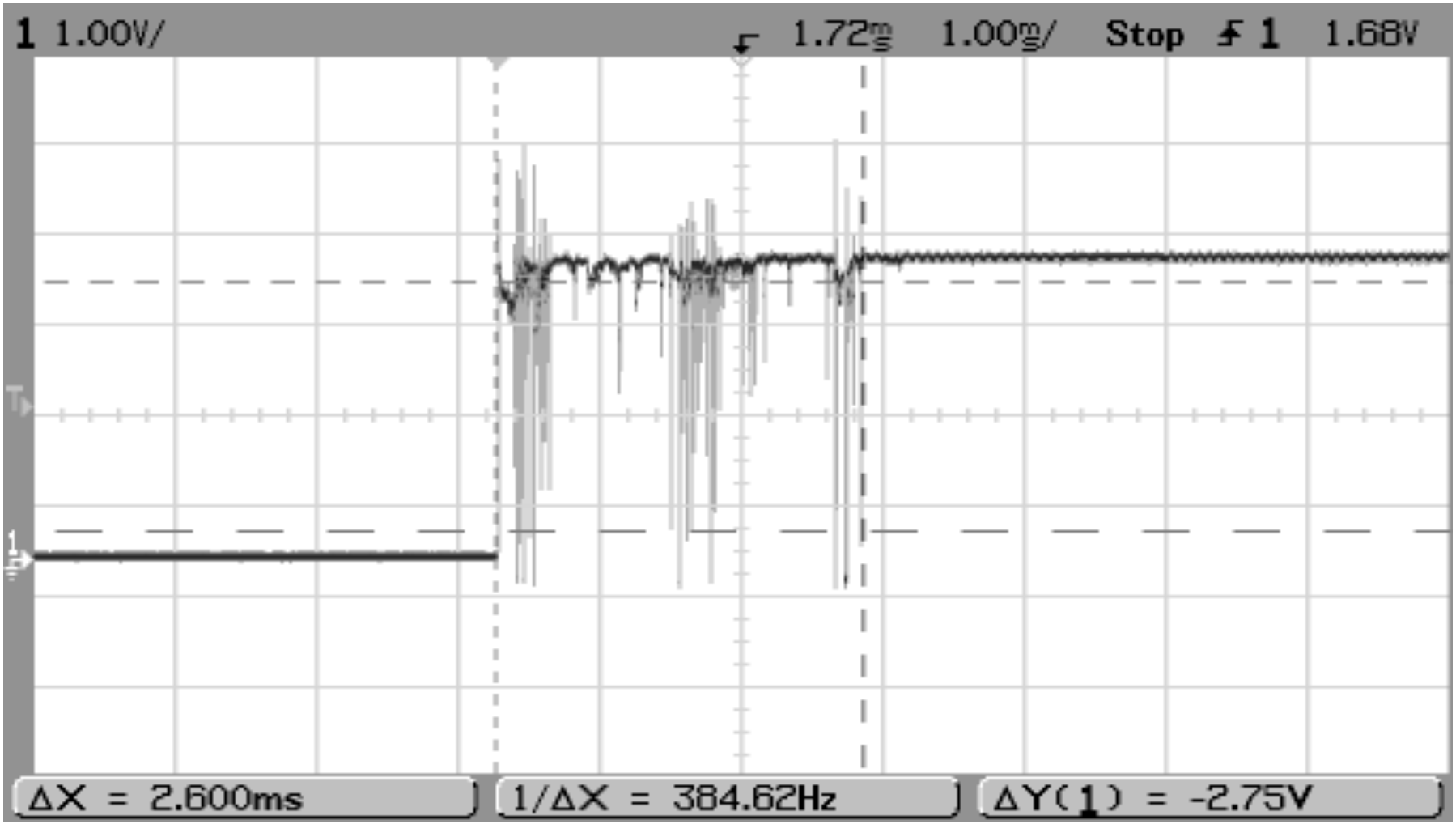
```
PORTB |= (1 << 5); // activeer Pull-up
DDR &= _BV(5); // _BV(5) = (1 << 5)

if (PINB && (1 << 5))
{
    ledOn();
}
else
{
    ledOff();
}
```

NB: weak pull-up

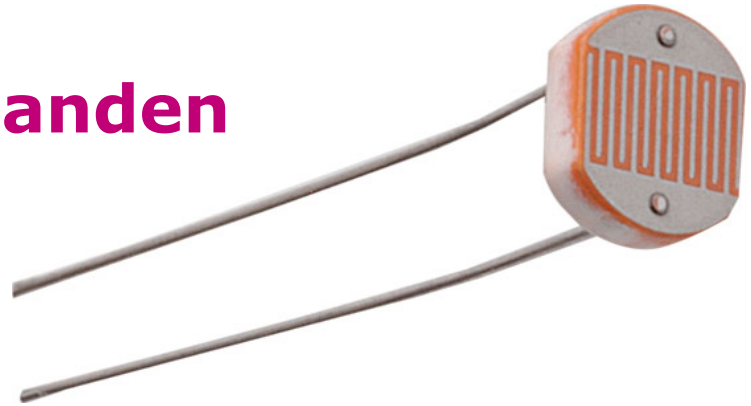
hogeschool Leiden

Denderende schakelaars (bounce)

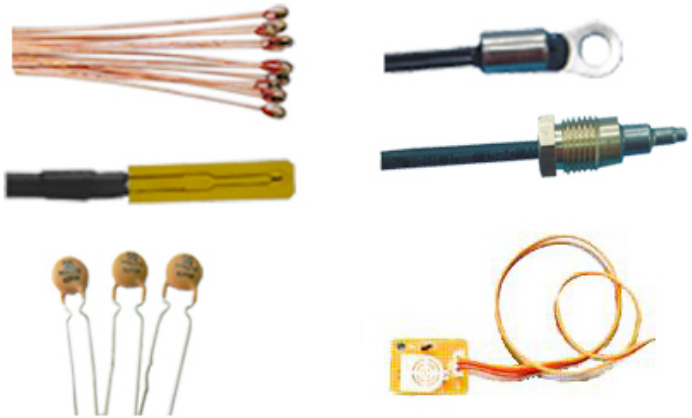


hogeschool Leiden

Analoge sensoren - weerstanden

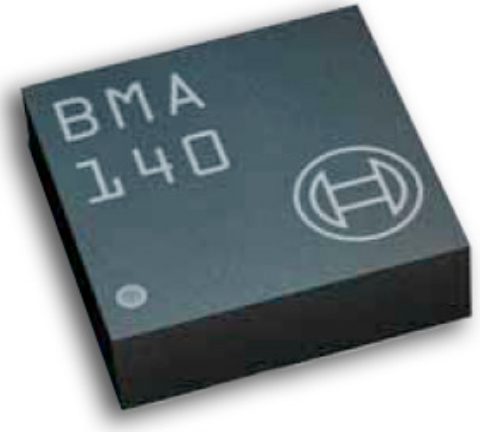


- Light Dependent Resistor
- NTC en PTC weerstand
- Forse Sensitive Resistor
  - DIY
- Reed Switch (magneet)
- Thermostaat (temperatuur)
  - Thermokoppel: speciaal IC
- E.v.a.



hogeschool Leiden

Analoge sensoren – Halfgeleiders



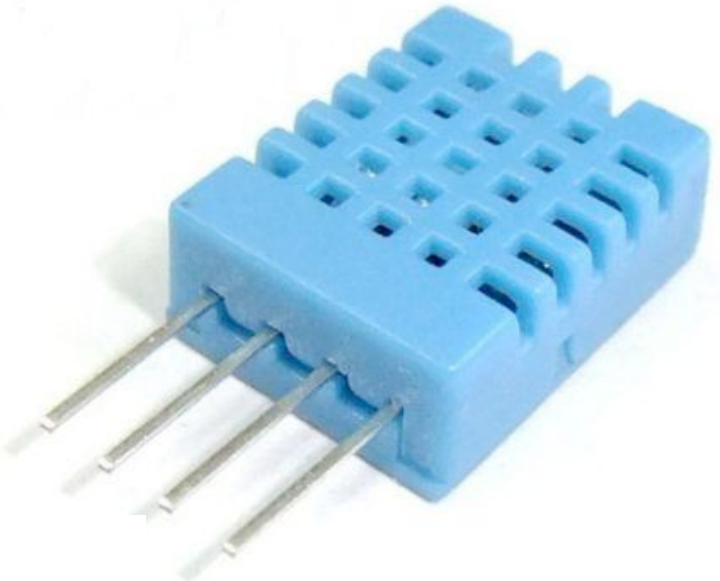
- Light Dependent Diode/Transistor
- Temperatuursensor
  - geijkt
- Hall Sensor (magneetveld)
  - Ratiometric
  - Switching
- Druk
  - Geluid/Microfoon
  - Piezo
- Acceleration

- E.v.a.

- Check de datasheet / voedingsspanning !!

hogeschool Leiden

Digitale sensoren



- Temperatuur
- Licht/Kleur
  - CMOS cam
- Druk
- Vochtigheid

- Versnelling/G-sensor

- Chemisch
  - Gas/rook

- Check de datasheet / voedingsspanning !!
- Protocollen
  - PWM
  - SPI,I2C en allerhande custom protocollen

hogeschool Leiden

Stappenmotor

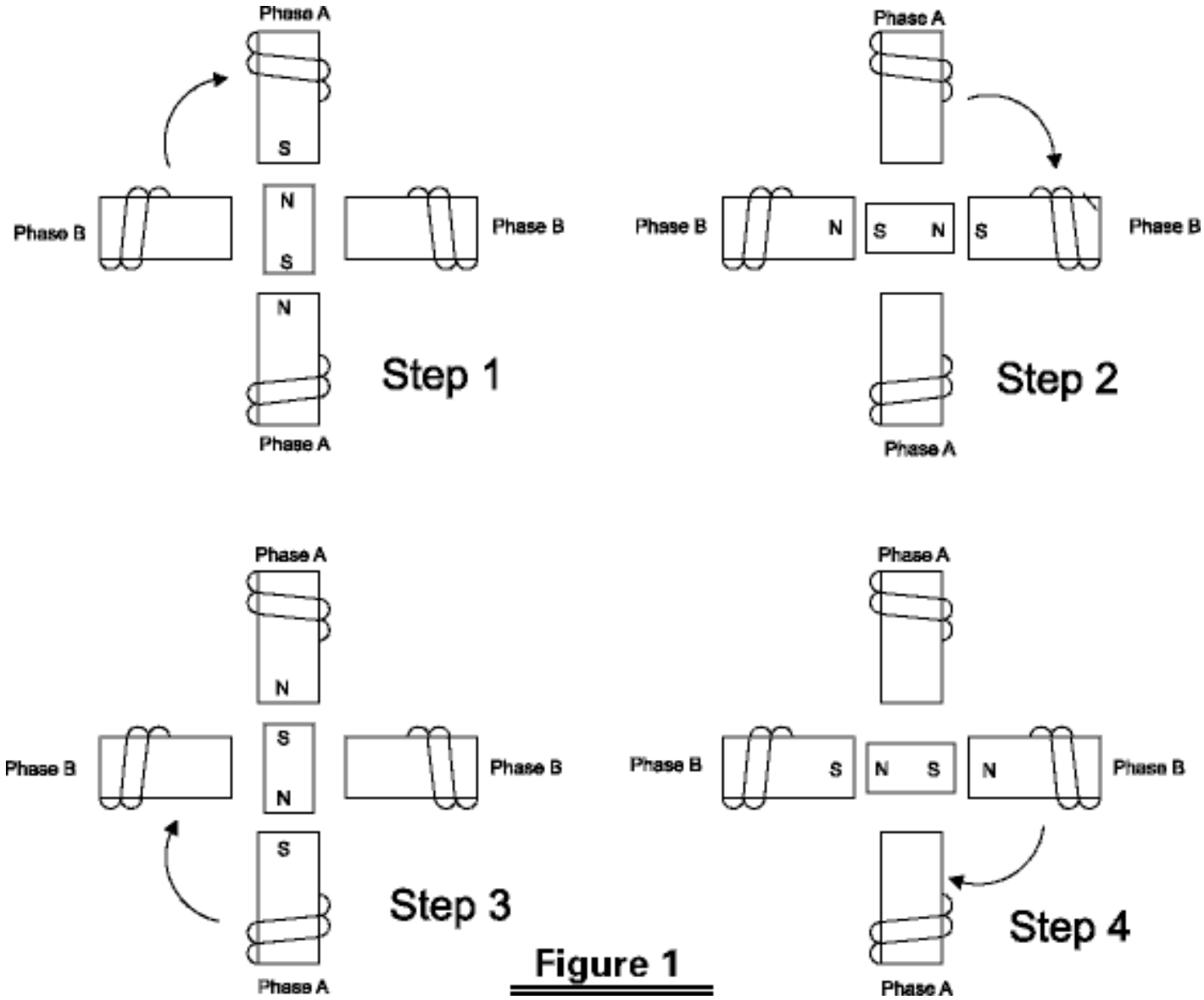


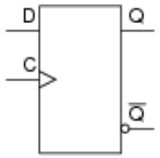
Figure 1

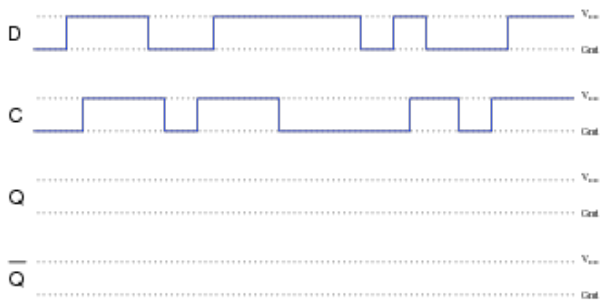
hogeschool Leiden

Week 4: Klok en Timing

hogeschool Leiden

De Processor Clock

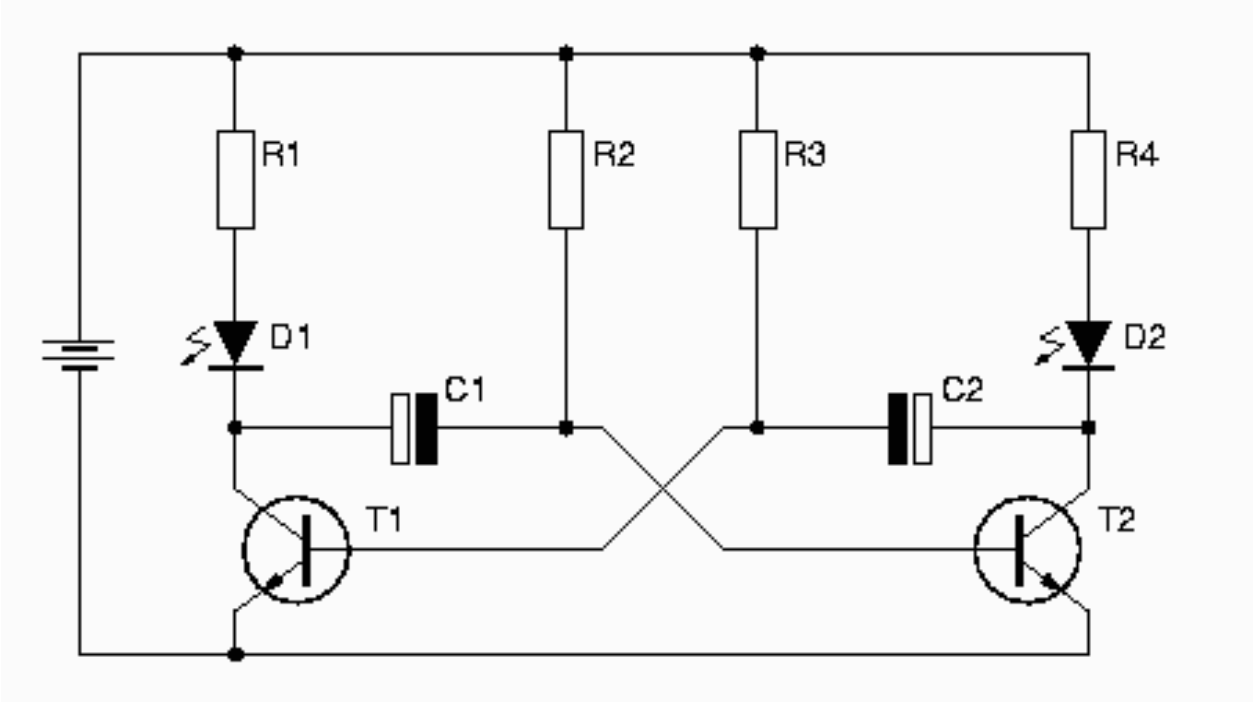




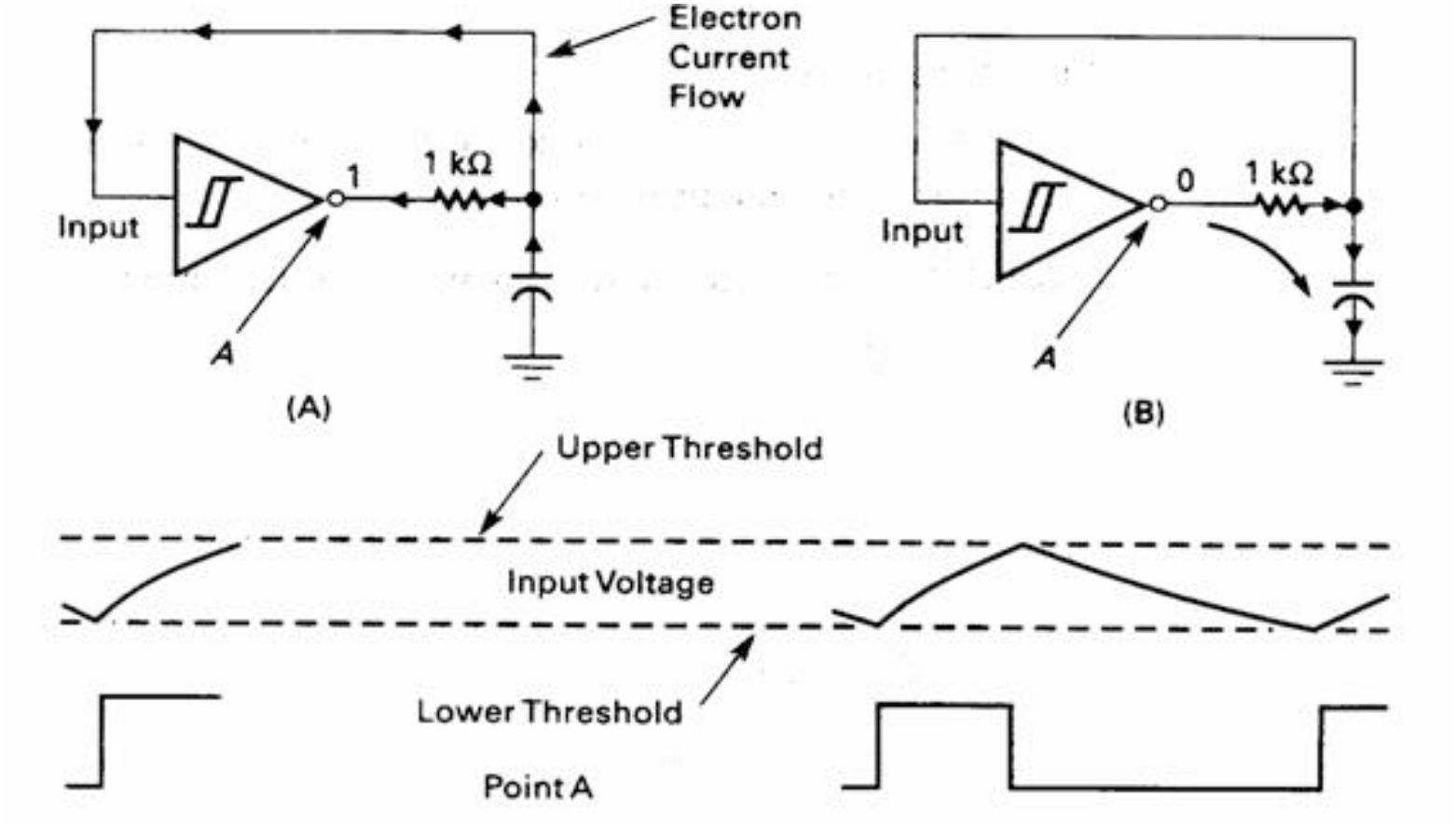
- Het clock-signaal is het hart van de processor
  - Denk aan de flip-flop
  - Normaliter 1 (machine)instructie per puls
  - Sommige instructies kosten meer ‘cycles’
- Min en max clockspeed
  - Energieverbruik
  - Spanning beinvloedt max clockspeed
    - overclocken
- Klokfrequentie is afhankelijk van de toepassing

hogeschool Leiden

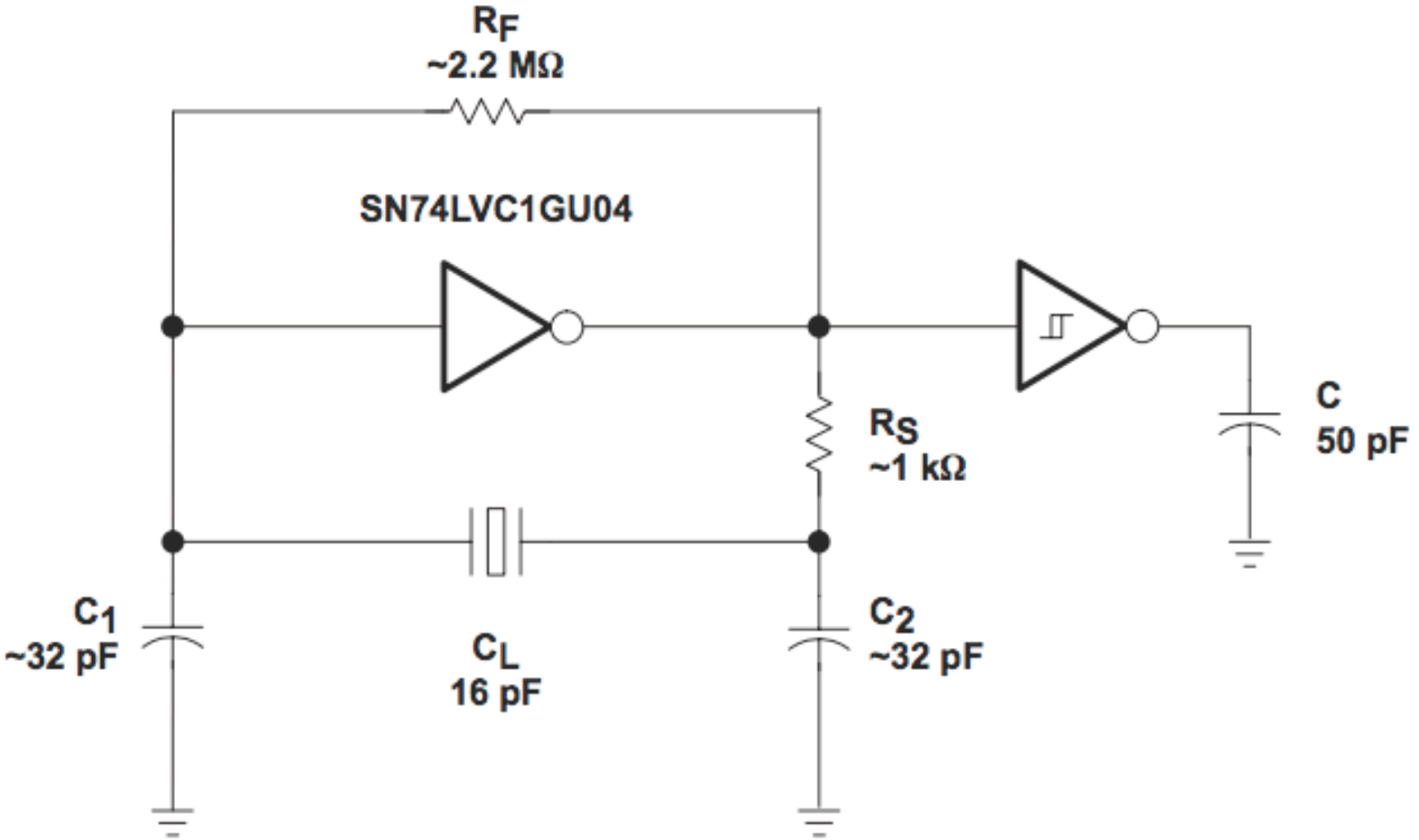
Astabiele Multivibrator

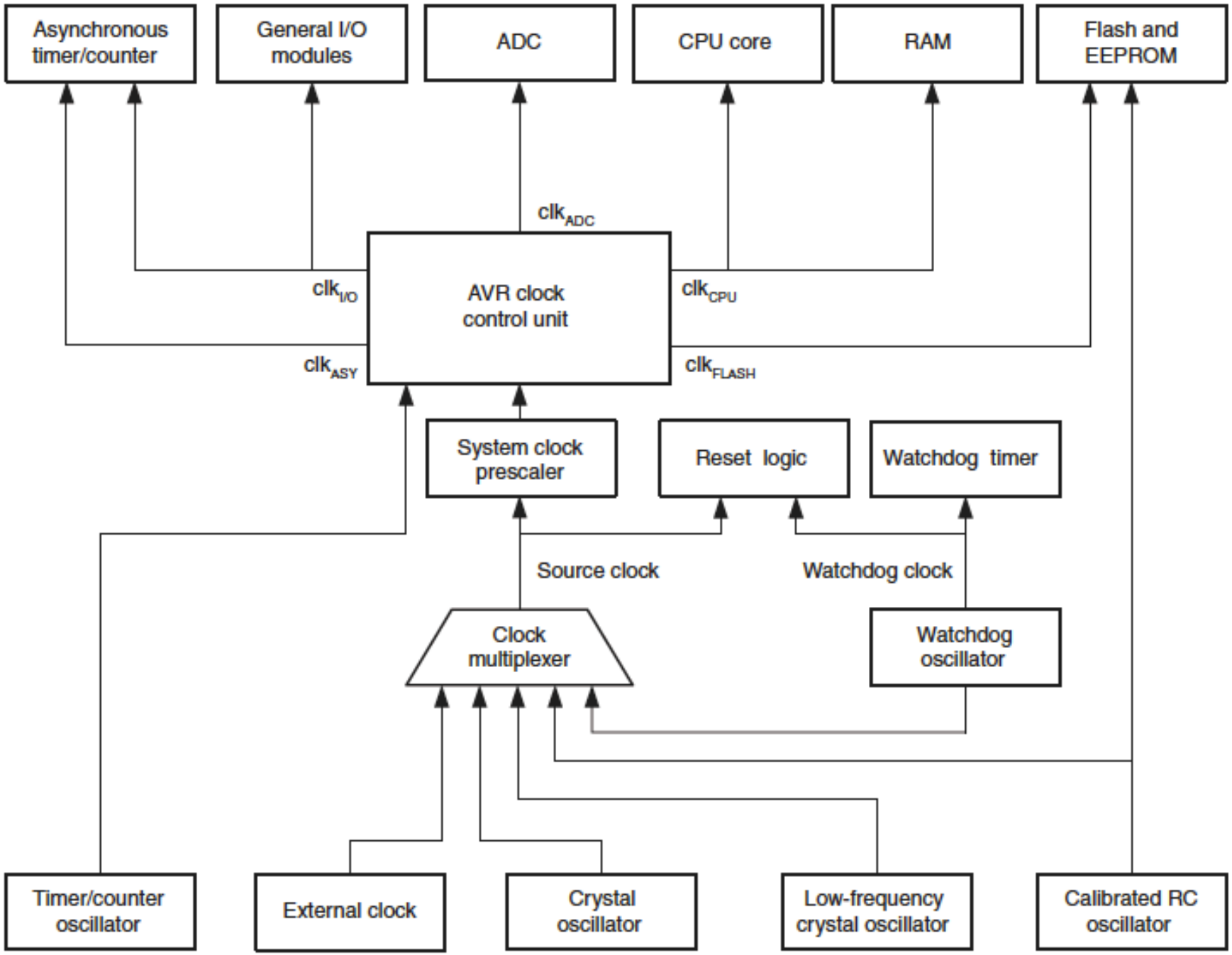


Het maken van een kloksignaal



Het maken van een kloksignaal




Asynchronous timer/counter
General I/O modules
ADC
CPU core
RAM
Flash and EEPROM
clk_ADC
clk_I/O
AVR clock control unit
clk_CPU
clk_ASY
clk_FLASH
System clock prescaler
Reset logic
Watchdog timer
Source clock
Watchdog clock
Clock multiplexer
Watchdog oscillator
Timer/counter oscillator
External clock
Crystal oscillator
Low-frequency crystal oscillator
Calibrated RC oscillator

AVR Clock

- Intern
  - Lofreq RC (128kHz)
  - Calibrated RC (7.3 – 8.1Mhz)
    - DIV8 Fuse
- Extern
  - Kristal
    - HF
    - LF (bv 32.768kHz kristal)
  - RC netwerk
  - Pulse
- Check of de compiler settings overeenstemmen met de fuse settings!!

- NB als de klok niet werkt kun je ook niet flashen met de USBasp!! Op 128kHz flashen lukt niet zonder kunstgrepen

hogeschool Leiden

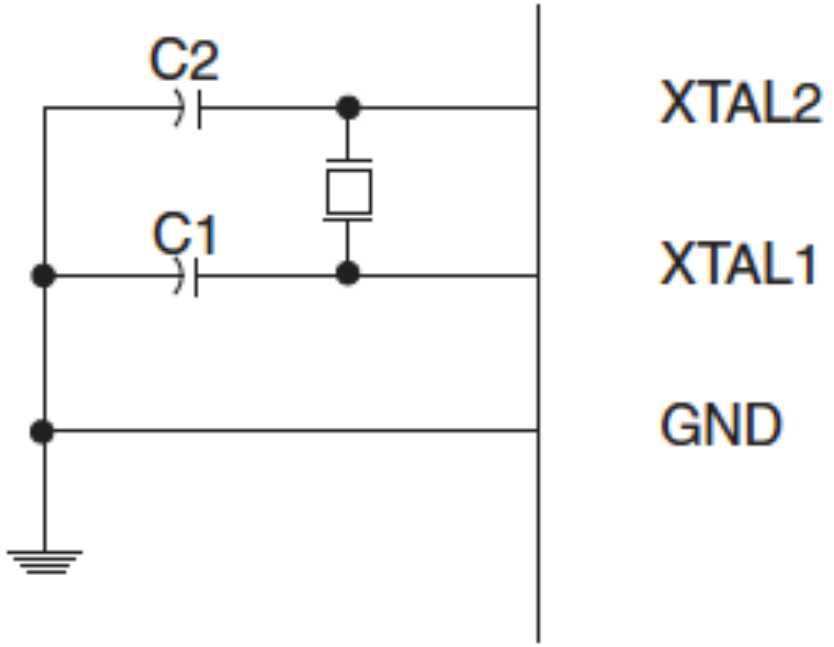
Prescaler

Table 9-14. Clock prescaler select.

CLKPS3	CLKPS2	CLKPS1	CLKPS0	Clock division factor
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256

Kristal aansluiten

Figure 9-2. Crystal oscillator connections.



Watchdog Timer

- Que?

hogeschool Leiden

Vormen van Reset

- Power-on reset. The MCU is reset when the supply voltage is below the power-on reset threshold (VPOT)
- External reset. The MCU is reset when a low level is present on the RESET pin for longer than the minimum pulse length
- Watchdog system reset. The MCU is reset when the watchdog timer period expires and the watchdog system reset mode is enabled
- Brown-out reset. The MCU is reset when the supply voltage VCC is below the Brown-out Reset threshold (VBOT) and the brown-out detector is enabled

Opdracht Blinkled & Clock Fuse

- Pak een BlinkLed met 1 sec knippertijd
- Klokfreq in de settings staat op 1Mhz
- Flash de Blinkled
- Zet de DIV8 fuse uit
  - Verander geen andere fuses!!
- Flash de Blinkled opnieuw
- Wat gebeurt er?

hogeschool Leiden

Bevindingen Opdracht?

hogeschool Leiden

Interrupts (volgens Wikipedia)

In computing, an **interrupt** is an asynchronous signal indicating the need for attention or a synchronous event in software indicating the need for a change in execution.

Interrupts are a commonly used technique for computer multitasking, especially in real-time computing. Such a system is said to be interrupt-driven.

An act of *interrupting* is referred to as an interrupt request (IRQ)

The part of a program (usually firmware, driver or operating system service) that deals with the interrupt is referred to as an interrupt service routine (ISR) or interrupt handler.

Voorbeeld programma (opdracht 4)

```
main()
{
    while(1)
    {
        doPWM();
    }
}
```

hogeschool Leiden

Voorbeeldprogramma

```
main()
{
    while(1)
    {
        rood = leesPotmeter(0);
        groen = leesPotmeter(1);
        blauw = leesPotmeter(2);
        doPWM();
    }
}
```

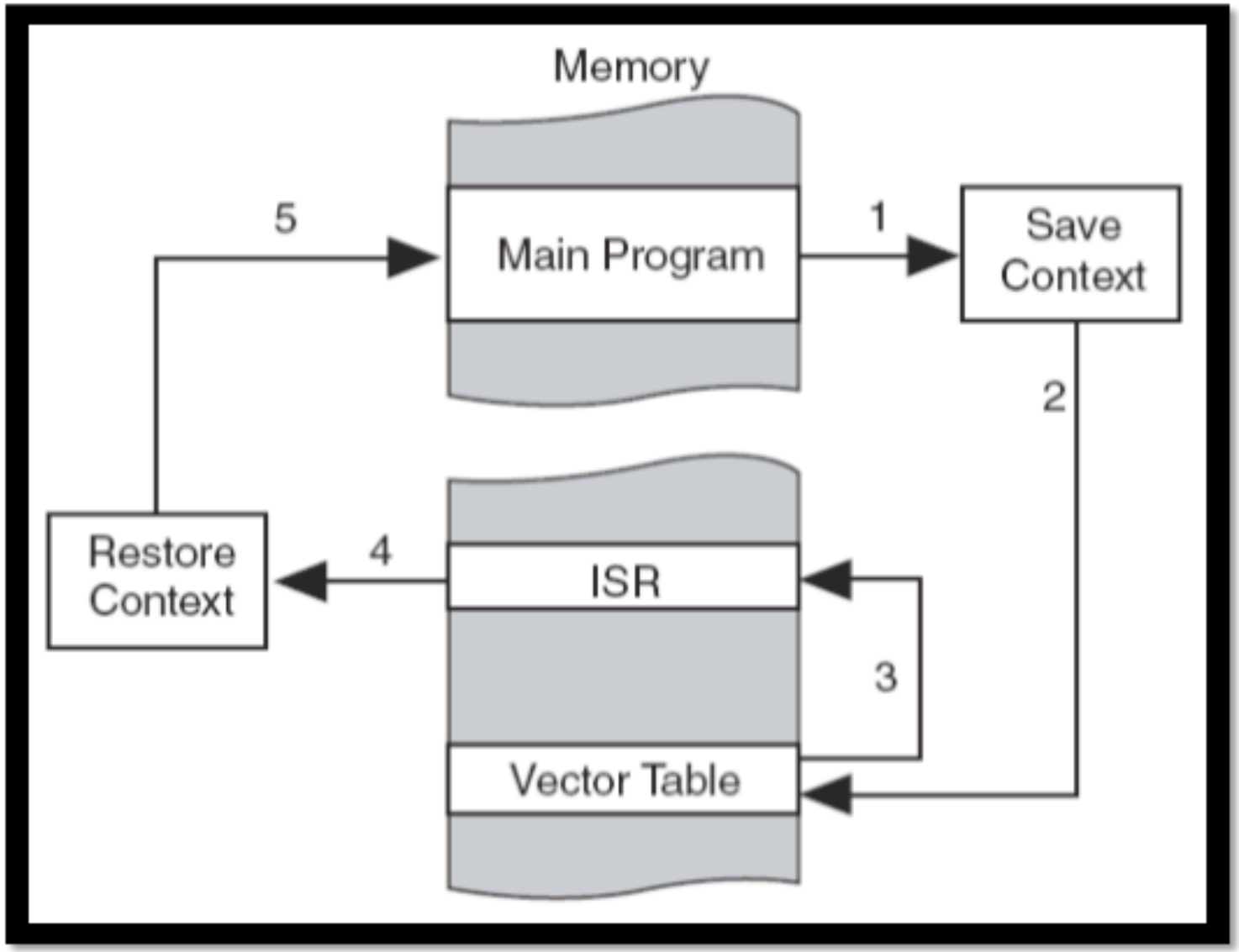
hogeschool Leiden

Voorbeeldprogramma 3: stopwatch

```
main()
{
    while(1)
    {
        for(i=0;i<100;i++)
        {
            schrijfDisplayLinks(secs/10);
            _delay_ms(5);
            schrijfDisplayRechts(secs%10);
            _delay_ms(5);
        }
        secs++;
    }
}
```

hogeschool Leiden

Interrupt execution



AVR Interrupts

- Interne interrups
  - Timer
  - Watchdog
  - UART/SPI/TWI (communicatie)
  - ADC ready
  - EEPROM ready

- Externe interrupts
  - Pinnen INT0 en INT1
  - Pins PCINT

- Een externe interrupt kan de chip laten ontwaken

Interrupt vectors

Vector no.	Program address	Source	Interrupt definition
1	0x000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x001	INT0	External interrupt request 0
3	0x002	INT1	External interrupt request 1
4	0x003	PCINT0	Pin change interrupt request 0
5	0x004	PCINT1	Pin change interrupt request 1
6	0x005	PCINT2	Pin change interrupt request 2
7	0x006	WDT	Watchdog time-out interrupt
8	0x007	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x008	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x009	TIMER2 OVF	Timer/Counter2 overflow
11	0x00A	TIMER1 CAPT	Timer/Counter1 capture event
12	0x00B	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x00C	TIMER1 COMPB	Timer/Coutner1 compare match B
14	0x00D	TIMER1 OVF	Timer/Counter1 overflow
15	0x00E	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x00F	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x010	TIMER0 OVF	Timer/Counter0 overflow
18	0x011	SPI, STC	SPI serial transfer complete
19	0x012	USART, RX	USART Rx complete
20	0x013	USART, UDRE	USART, data register empty
21	0x014	USART, TX	USART, Tx complete

Voorbeeldprogramma 4

```
// TIMER2 telt van 0 -> 255
// dus Overflow is elke 256 pulsen
// stel F_CPU is 32.768 kHz
// dan TIMER2 prescaler van 32768/256 = 128

ISR(TIMER2_OVF_vect)
{
   secs++;
}

// dit is dus de reden dat 32.768kHz kristallen veel voorkomen
// worden ook wel horlogekristallen genoemd, omdat ze in digitale
   horloges zitten
```

hogeschool Leiden

Stopwatch revisited

```
volatile uint8_t seconds;

main() {
    while(1) {
        schrijfDisplayLinks(seconds/10);
        schrijfDisplayRechts(seconds%10);
    }
}

ISR(TIMER2_OVF_vect) {
    seconds++;
    if(seconds > 99)
        seconds = 0;
}
```

hogeschool Leiden

Stopwatch revisited 2

```
main() {
    while(1);
}

ISR(TIMER2_OVF_vect) {
    seconds = (seconds + 1) % 100;
}

ISR(TIMER0_OVF_vect) {
    if(LEFT) {
            schrijfDisplayLinks(seconds/10);
            LEFT = FALSE;
    } else {
            schrijfDisplayRechts(seconds%10);
            LEFT = TRUE;
    }
}
```

hogeschool Leiden

Advanced Timing

```
// stel F_CPU is 8000000
// dan TIMER2 prescaler van 8000000/256 = 31250
// bestaat niet, max prescaler = 1024;
// laten we millisecs doen
// 8000 / 256 = 31,25 => bestaat ook niet
// met een prescaler van 64 => 8000 / 64 = 125
// => bij een prescaler van 64 moet de counter tot 125 tellen (ipv
    256)
// dus gebruik een TIMER_COMPARE met 125 in CTC mode

ISR(TIMER2_OVF_vect) {
    millisecs++;
    if(millisecs > 999) {
          milliSecs = 0;
          seconds ++;
    }
}
```

hogeschool Leiden

Interrupts moeten wel aangezet worden

```
#include <interrupt.h>

//configureer registers (i.e. TIMSK0)

sei(); // zet globale interupts aan (nadat alle registers goed staan)

cli(); // zet globale interrupts uit
```

Timers extra

- Timer overflow en compare kan ook aan externe pinnen gehangen worden; op die manier ‘hardwarematig’ PWM doen
- Lees de datasheet!!

Figure 15-5. CTC mode, timing diagram.

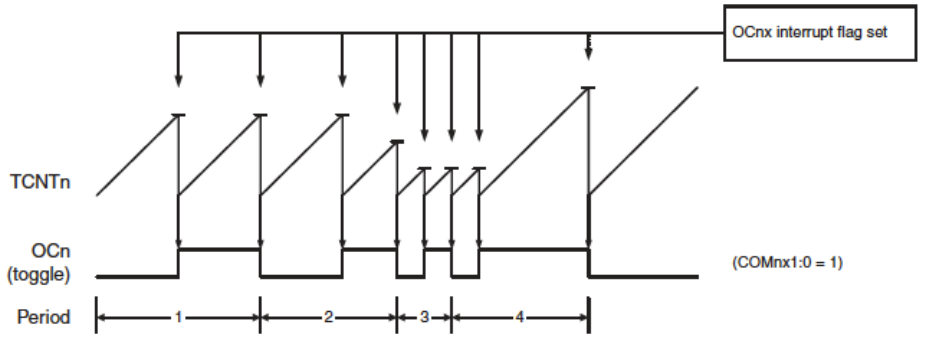
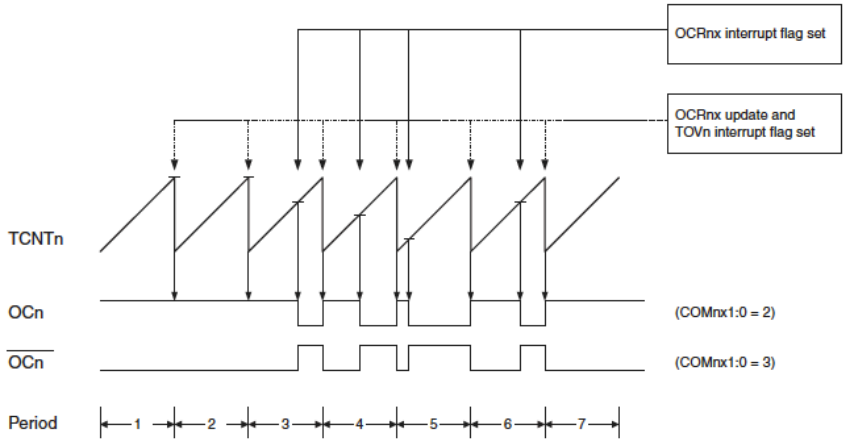


Figure 15-6. Fast PWM Mode, timing diagram.



hogeschool Leiden