A detailed wireframe model of a particle accelerator, showing a large, oval-shaped ring structure with various internal components and a smaller, more complex structure in the background.

# **Clock Synthesizer 3 Firmware (CLOS3)**

Martin Mitkov

# AGENDA

- CLOS Y Projekt
- Lösungsweg
- CLOS Y Blockdiagram
- Plan & Status: Bridge
- Bridge API Beispiel
- Plan & Status: All-in-One
- Probleme

# CLOS Y Projekt

## Das Ziel des Projektes:

- Konfigurierbar über einer USB Schnittstelle
- Automatische Wiederherstellung der letzten Konfiguration nach Neustart

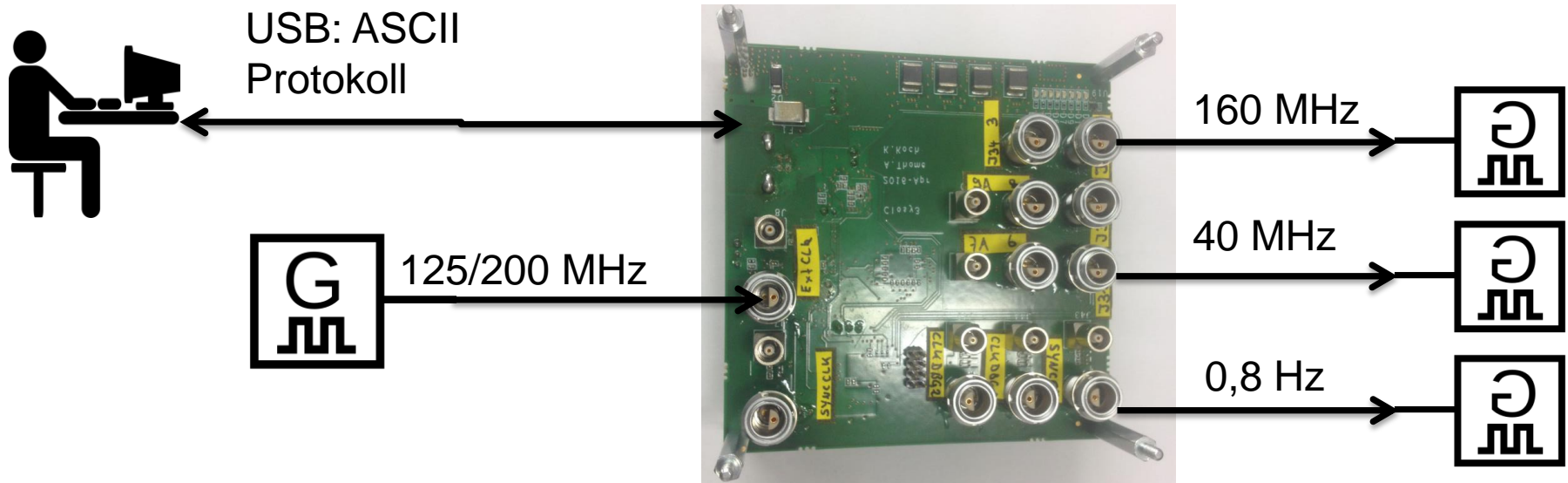
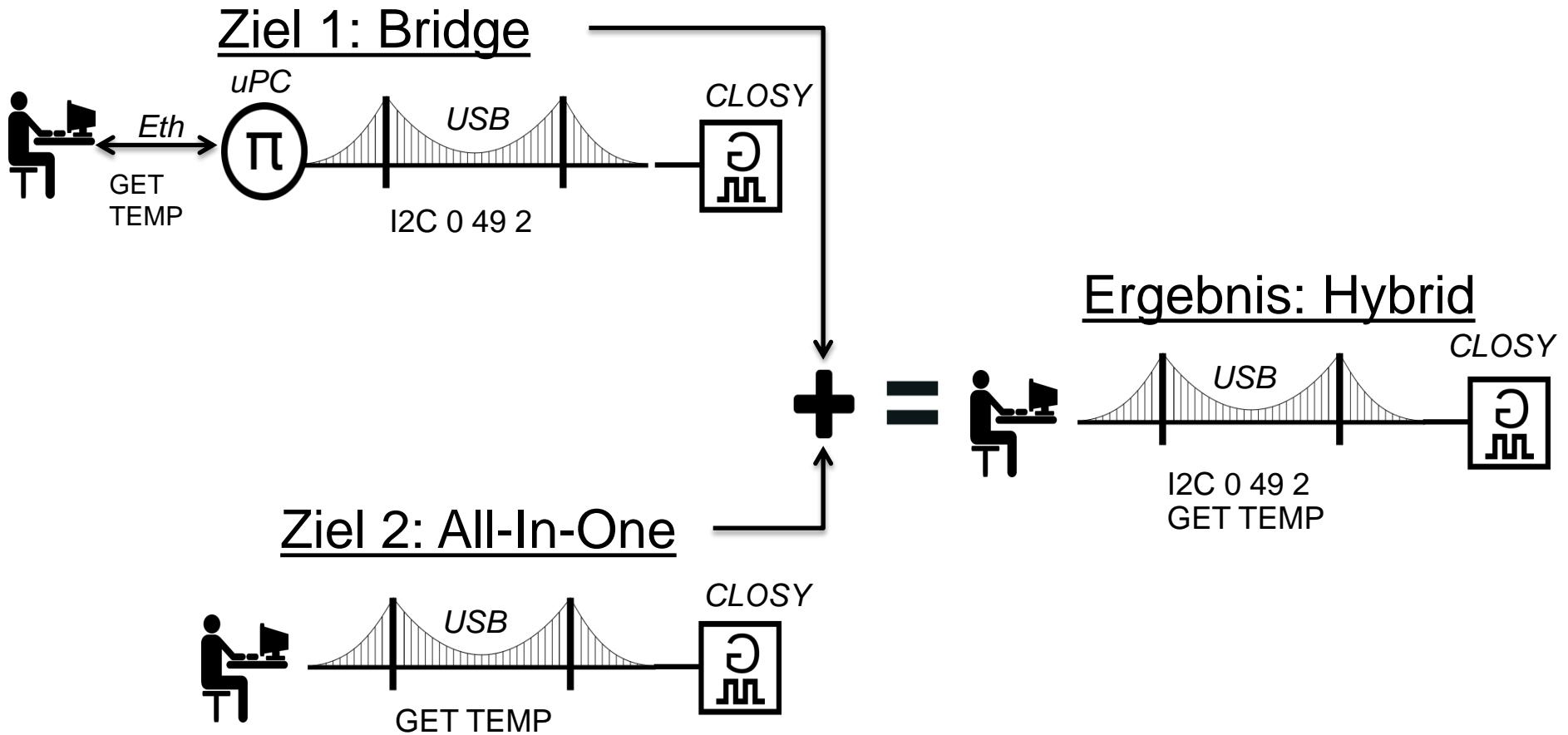


Abbildung 1: CLOS Y

# Lösungsweg



# CLOSY Blockdiagram

## Notizen

**I<sup>2</sup>C:** Inter-Integrated Circuit  
**SPI:** Serial Peripheral Interface  
**CPLD:** Complex Programmable Logic Devices

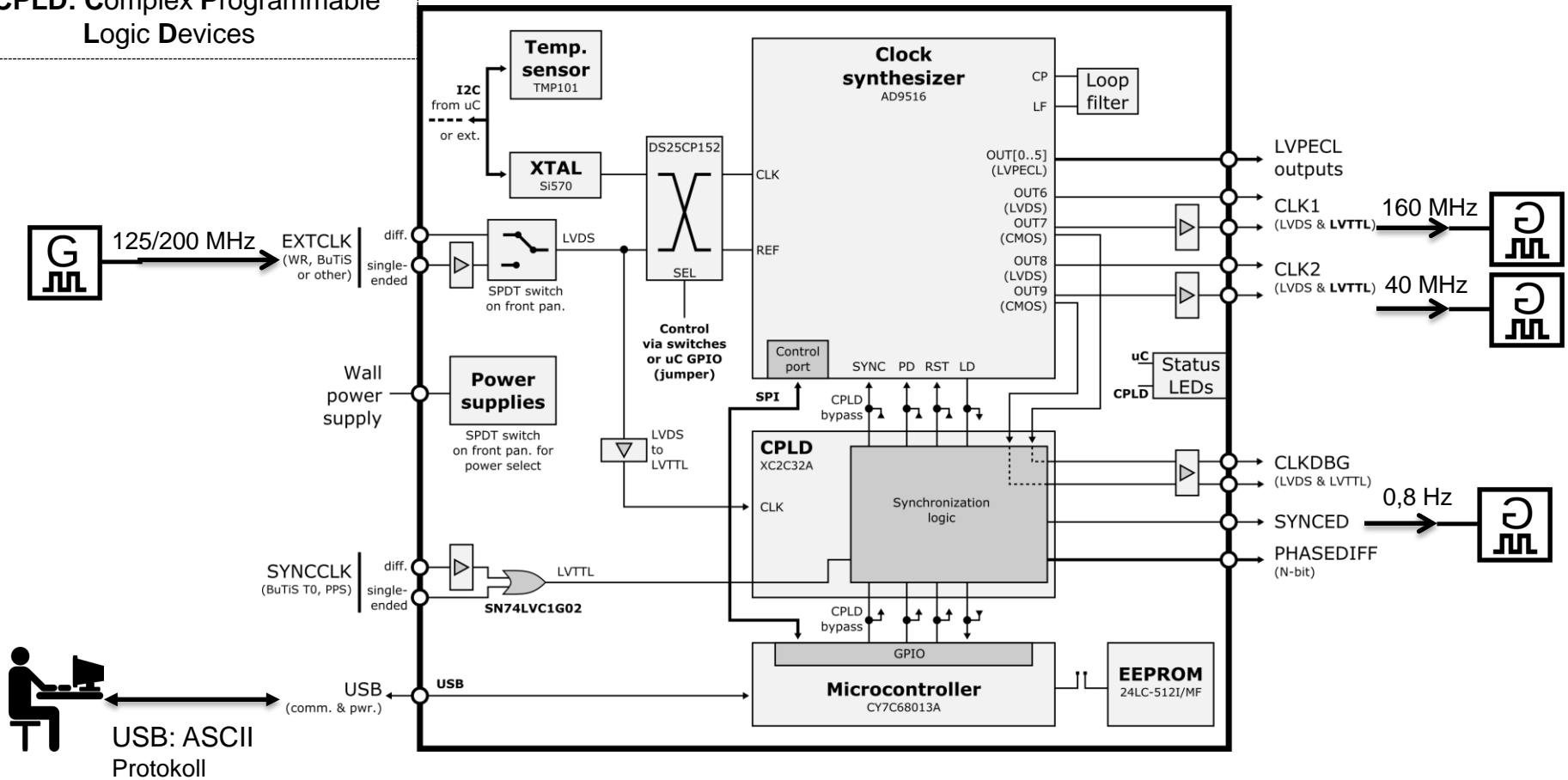


Abbildung 2: CLOSY Blockdiagram

# Plan & Status: Bridge

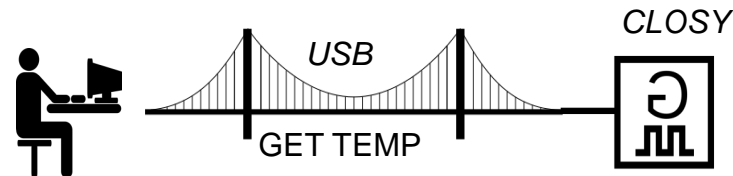
- Entwicklungssystem einsetzen:
  - EZ USB Suite IDE & Compiler SDCC
  - Mikrokontroller kennenlernen: *Hallo World Beispiel*
- CLOSY Firmware Entwicklung
  - Kommunikationsschnittstelle für die Peripherie
    - USB: OK
    - I2C: OK
    - SPI: OK
  - Anwendungsprogrammierschnittstelle:
    - USB ↔ I2C/SPI
    - Befehle: I2C, SPI
    - wie HadCon2:  
<https://wiki.gsi.de/foswiki/bin/view/EE/HadCon2#Introduction>
  - Konfiguration in den externen Speichern schreiben/lesen

## Bridge API Beispiel:

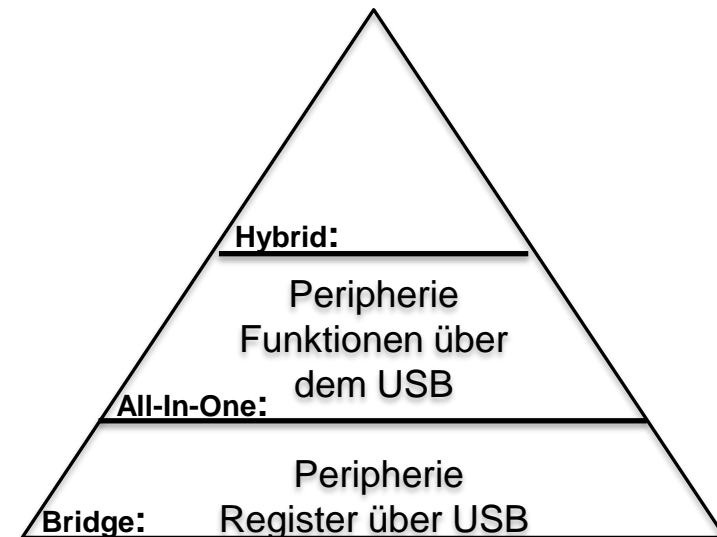
- **Form:** I2C <0|1> <I2C address> <data length> <byte1 ... byte8>  
 <0|1>: used to execute write or read operation. This is the **Least Significant Bit (LSB)** from the I2C address byte.
- **<I2C address>:**  
 Here has to be written the hexadecimal value on the address byte, which is shifted to right with one bit. The reason for the right shift is because, the LSB bit ist already used in the first field, where the working operation is choosen.
- **<data length>:** Anzahl Data Bytes
- **<byte1 ... byte8>:** Data
- **Example command:**  
 send: I2C 0 49 2  
 receive: RECV I2C 0 49 01 08 -OK-

# Plan & Status: All-in-One

## Ziel 2. All-In-One



- Anwendungsprogrammierschnittstelle erweitern:
  - Von USB ⇔ I2C/SPI (Bridge) zu USB ⇔ Menu ⇔ I2C/SPI
- erweitern des Befehlssatz:
  - MODE < EXPERIMENT | LABOR >
  - OUTPUT <CHANNEL> <ON|OFF>
  - REF <FREQ>
  - PLL <CONTRIL ID> <VALUE>
  - DIVIDER <CHANNEL> <VALUE>
  - TEMP
- Beispiel:
  - Bridge: I2C 0 49 2 => RECV I2C 0 49 01 08 -OK-
  - All-In-One: TEMP => RECV TEMP 0108 -OK-

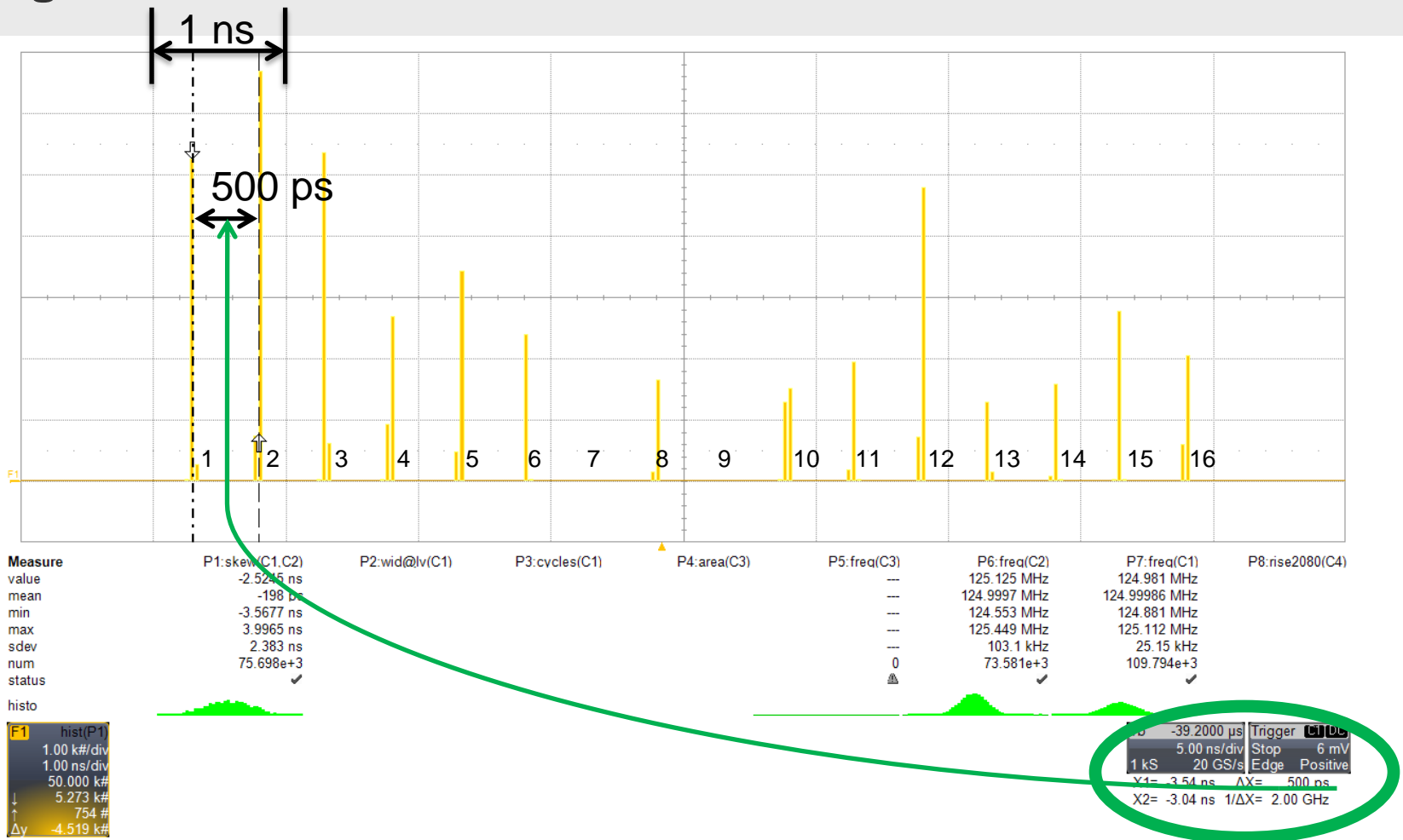




# Probleme

- Hauptproblem Speicherbedarf:
  - USB als Virtuelle COM Schnittstelle: *7/16KB*
  - SPI + **I2C** schreiben/lesen: *1/16KB*
  - Funktionen: *8/8KB*
    - USB  $\leftrightarrow$  I2C/SPI Befehle: ?
    - USB  $\Rightarrow$  EEPROM: ?
    - EEPROM  $\Rightarrow$  SPI/I2C/Zustand der Ausgänge: ?

# Clock Synthesizer Phasenverschiebung Möglichkeiten



XA2C32A → XA2C64A

**Macrocells Used/Tot:** 26/32 (82%)  
**Product Terms Used/Tot:** 37/112 (34%)  
**Function Block Inps Used/Tot::** 41 /80 ( 51%)  
**Registers Used/Tot::** 23 /32 ( 72%)  
**Pins Used/Tot:** 7 /33 ( 21%)

**Macrocells Used/Tot:** 26 /64 ( 41%)  
**Product Terms Used/Tot:** 37 /224 ( 17%)  
**Function Block Inps Used/Tot::** 41 /160 ( 26%)  
**Registers Used/Tot::** 23 /64 ( 36%)  
**Pins Used/Tot:** 7 /33 ( 21%)

1. Detektieren eine nicht statische Phasenverschiebung

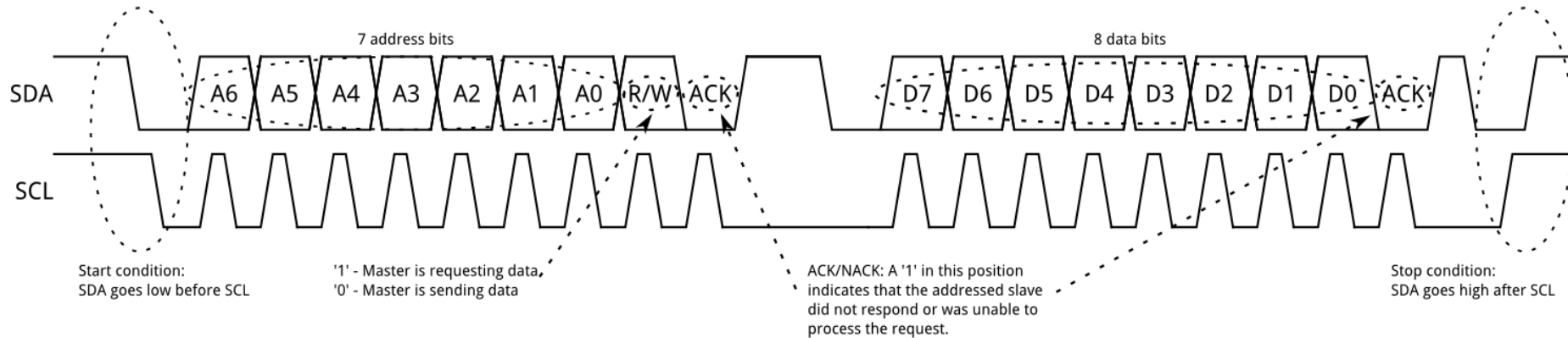
1. Detektieren eine nicht statische Phasenverschiebung
2. SYN Signal selbst generieren
3. Externen SYN Signal weiterleiten

Performance Summary	
<b>Min. Clock Period</b>	12.000 ns .
<b>Max. Clock Frequency (<a href="#">fSYSTEM</a>)</b>	83.333 MHz.
Limited by Clock Pulse Width for marker3<1>_MC.Q	
<b>Clock to Setup (<a href="#">tCYC</a>)</b>	5.000 ns .
<b>Pad to Pad Delay (<a href="#">tPD</a>)</b>	7.800 ns .
<b>Setup to Clock at the Pad (<a href="#">tSU</a>)</b>	-2.300 ns .
<b>Clock Pad to Output Pad Delay (<a href="#">tCO</a>)</b>	19.500 ns .

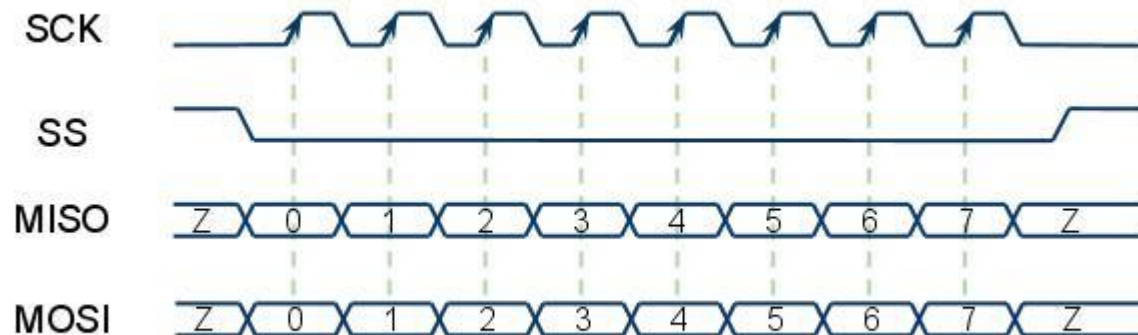
Vielen Dank!

# Was sind I2C & SPI?

## Inter-Integrated Circuit (I2C)



## Serial Peripheral Interface (SPI)



# Quellen

- [https://wiki.gsi.de/foswiki/bin/view/Epics/HadConMultipurposeControlsProtocolCmndSpi#SPI\\_spi\\_enable](https://wiki.gsi.de/foswiki/bin/view/Epics/HadConMultipurposeControlsProtocolCmndSpi#SPI_spi_enable)
- [https://commons.wikimedia.org/wiki/File:Signal\\_Generator\\_rectangle.svg](https://commons.wikimedia.org/wiki/File:Signal_Generator_rectangle.svg)
- <https://pixabay.com/de/computer-b%C3%BCro-arbeitnehmer-eingabe-146329/>
- [https://www.iconfinder.com/icons/430087/check\\_checkmark\\_circle\\_icon](https://www.iconfinder.com/icons/430087/check_checkmark_circle_icon)
- [https://commons.wikimedia.org/wiki/File:Suspension\\_bridge\\_icon.svg](https://commons.wikimedia.org/wiki/File:Suspension_bridge_icon.svg)
- [https://commons.wikimedia.org/wiki/File:Letter\\_Pi.svg](https://commons.wikimedia.org/wiki/File:Letter_Pi.svg)
- <https://commons.wikimedia.org/wiki/File:Toicon-icon-afiado-equal.svg>
- <https://pixabay.com/de/pi-symbol-mathematik-wissenschaft-2718206/>
- <https://cdn.sparkfun.com/assets/6/4/7/1/e/51ae0000ce395f645d000000.png>
- <https://commons.wikimedia.org/wiki/File:SPI-protocollo.jpg>