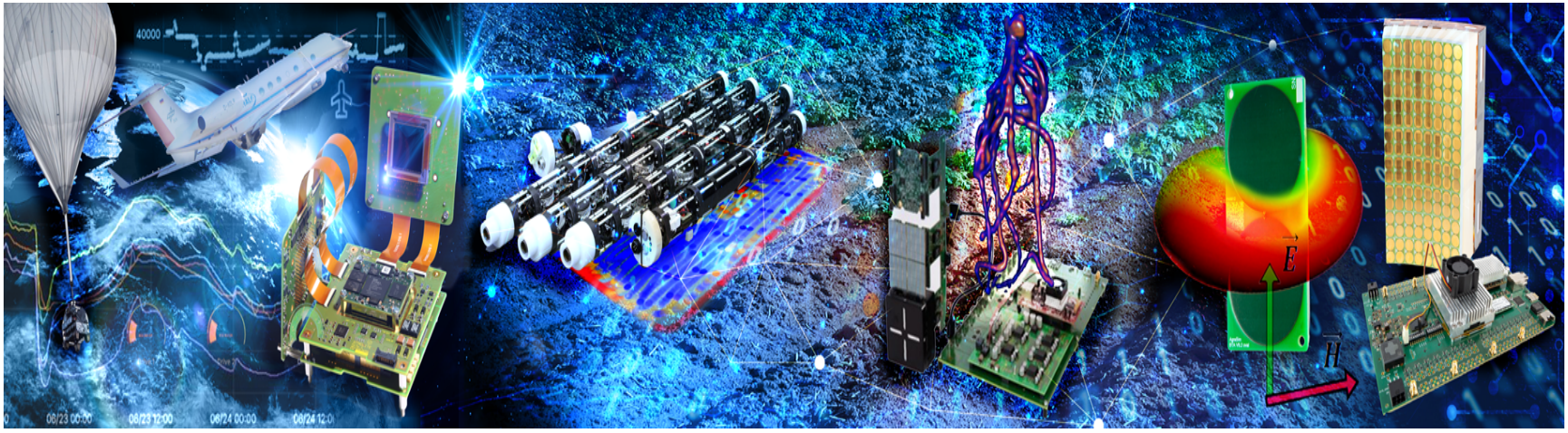


IMPLEMENTIERUNG EINES SKALIERBAREN RADAR-SYSTEMS

mit verteilter Datenerfassung auf Basis Xilinx RFSoC ZU25DR

26.03.2025 | Mathias Bachner, Ilja Bekman, Achim Mester, Georg Schardt | ITE

ÜBERSICHT

Motivation

Konzept

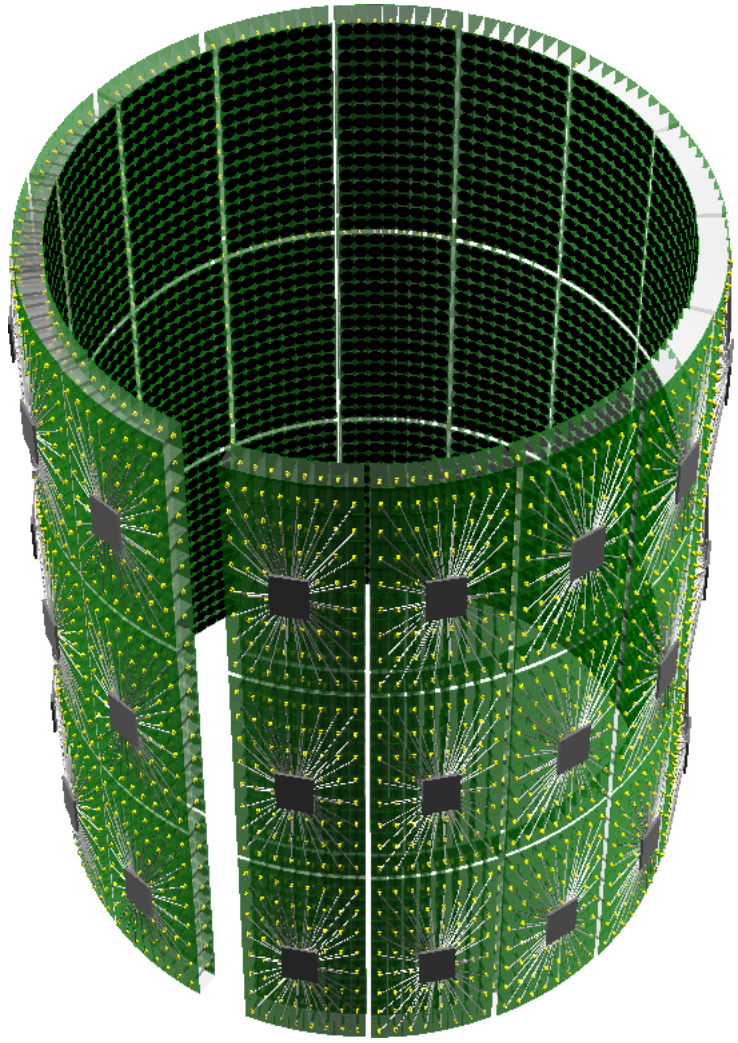
Aufbau

Zusammenfassung

Ausblick

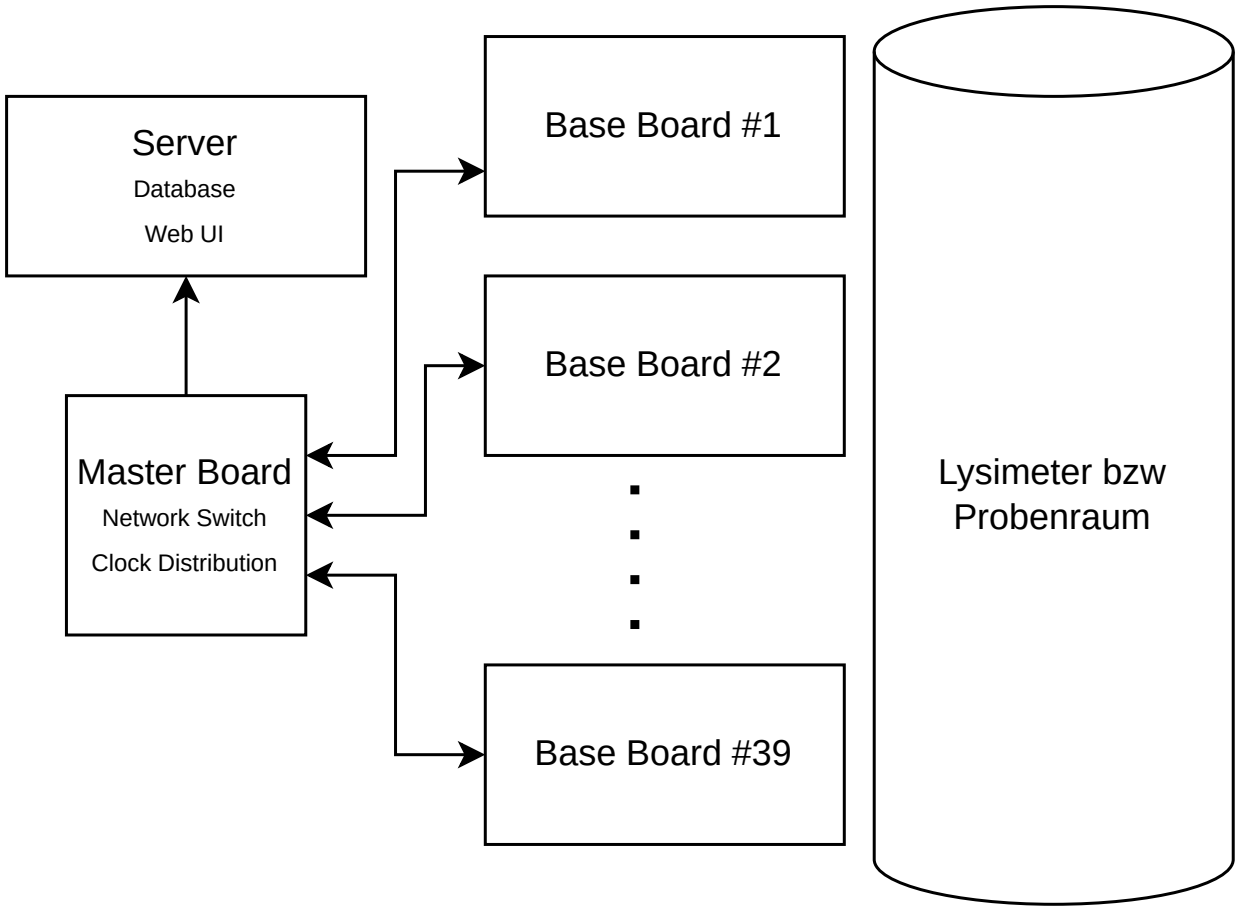
MOTIVATION

- Tomografisches Bodenradar, zylindrischer Aufbau, Durchmesser 1 m, Höhe 1,5 m
- Messung mit Ricker-Pulsen, Frequenzbereich 250 MHz – 1 GHz
- 2496 Antennen (13 Sektoren*3 Ringe*8 Zeilen*8 Spalten)
- Messung von jeder Antenne zu jeder Antenne (6 Millionen Messungen)
- Genauigkeit $<$25 ps, Messzeit $<$10 s



SMUCT-GPR

Scalable Multi-Channel Tomography Ground Penetrating Radar



SMUCT-GPR

Aufbau:

- Ein
  System für alle 2496 Antennen

SMUCT-GPR

Aufbau:

- Ein
  System für alle 2496 Antennen
- 1. Vereinfachung: Aufteilung auf
  Kacheln mit je 8x8 Antennen

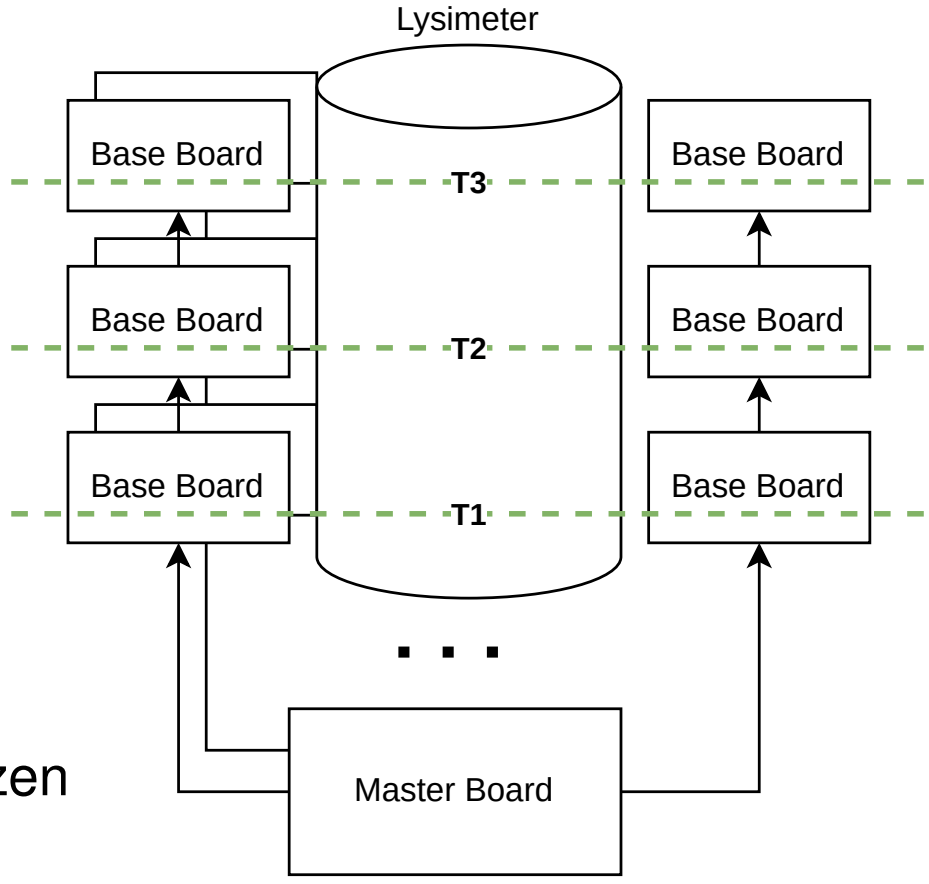
SMUCT-GPR

Aufbau:

- Ein
  System für alle 2496 Antennen
- 1. Vereinfachung: Aufteilung auf Kacheln mit je 8x8 Antennen
- 2. Vereinfachung: 13 Verbindungen vom Master Board zu jedem Sector, Reihen per Daisy-chain.

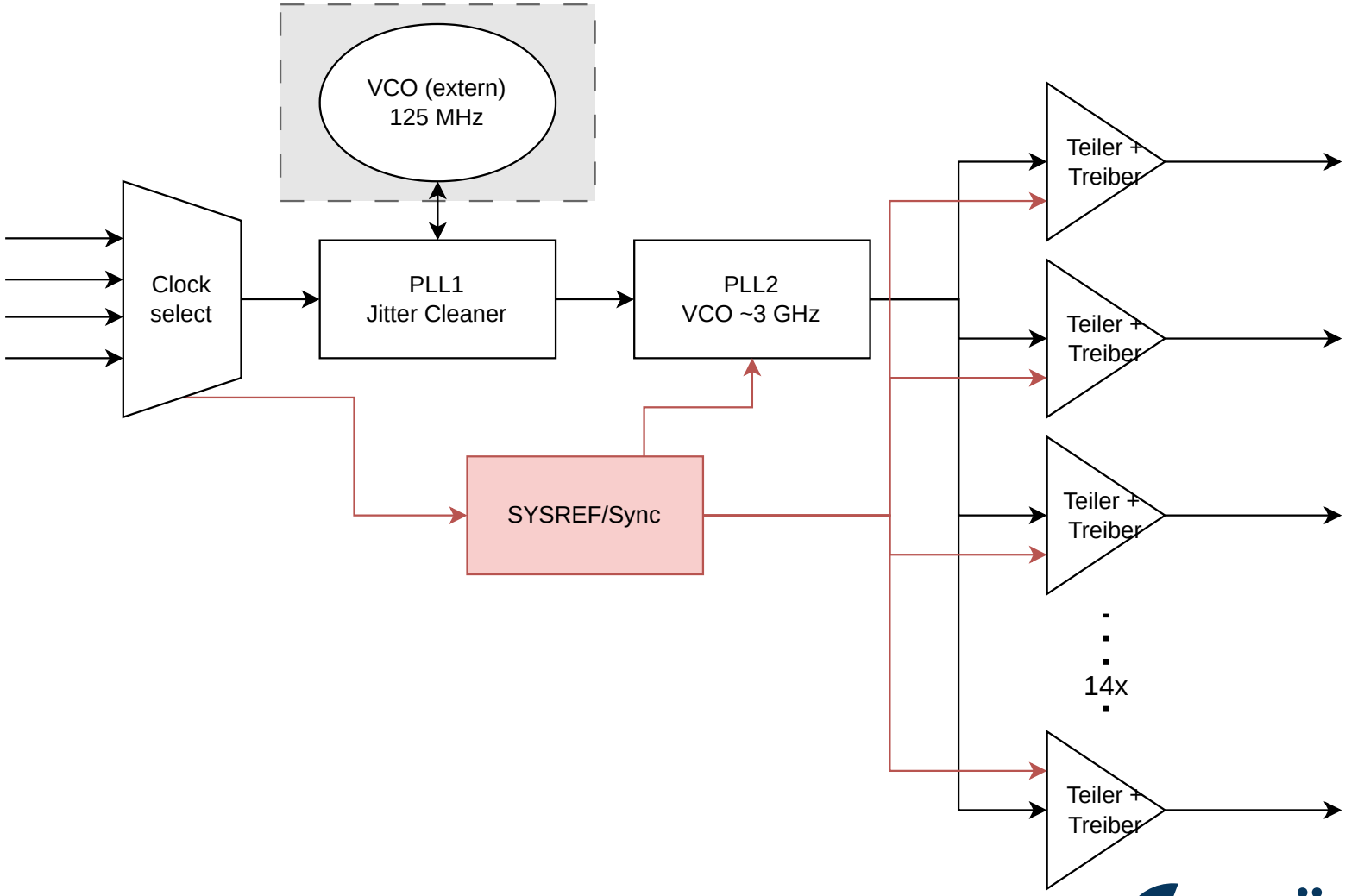
Zeitebenen:

- Verkettung ergibt 3 Zeitebenen
- finale Kalibration der Zeitdifferenzen



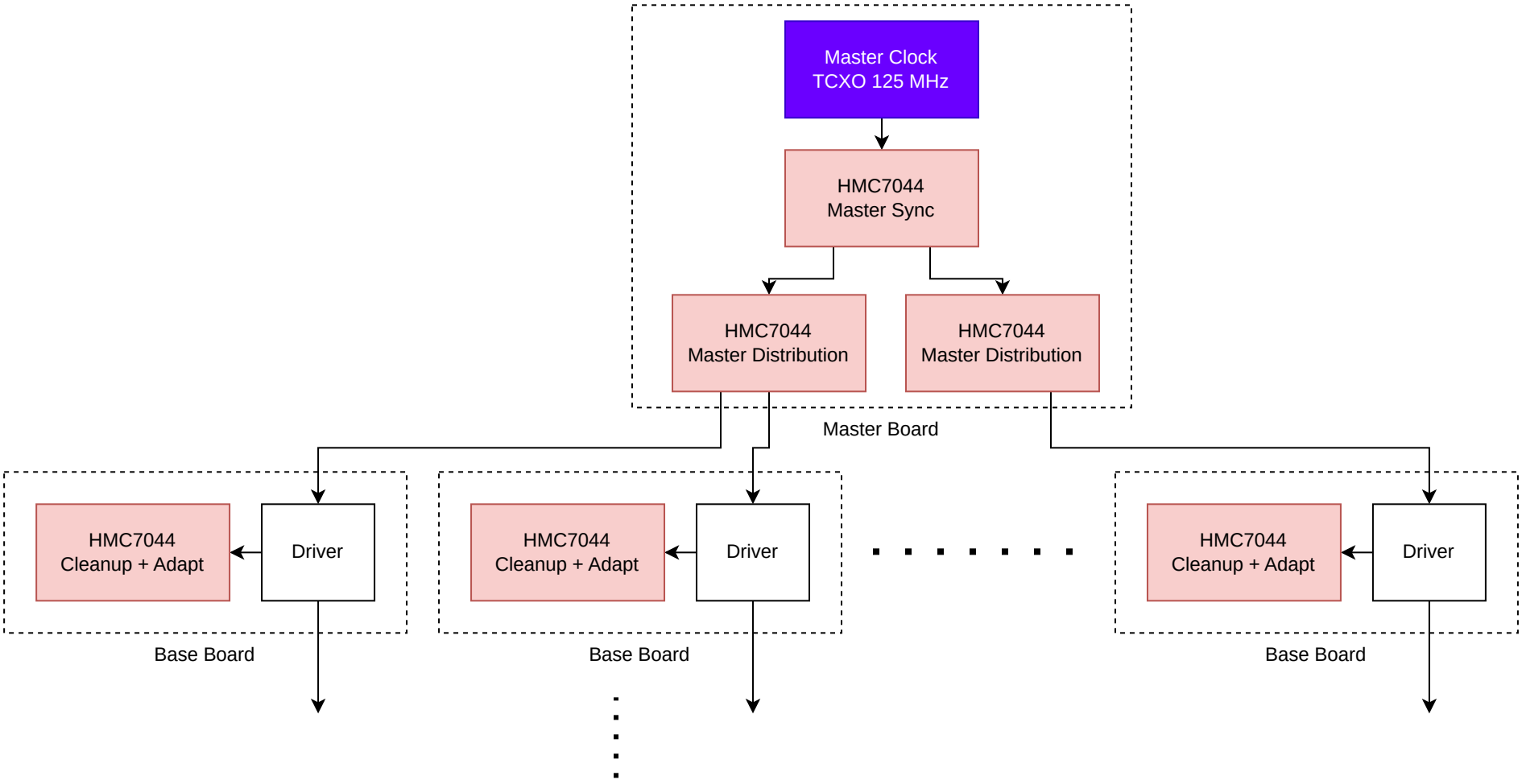
CLOCK

HMC7044 mit Doppel PLL Aufbau



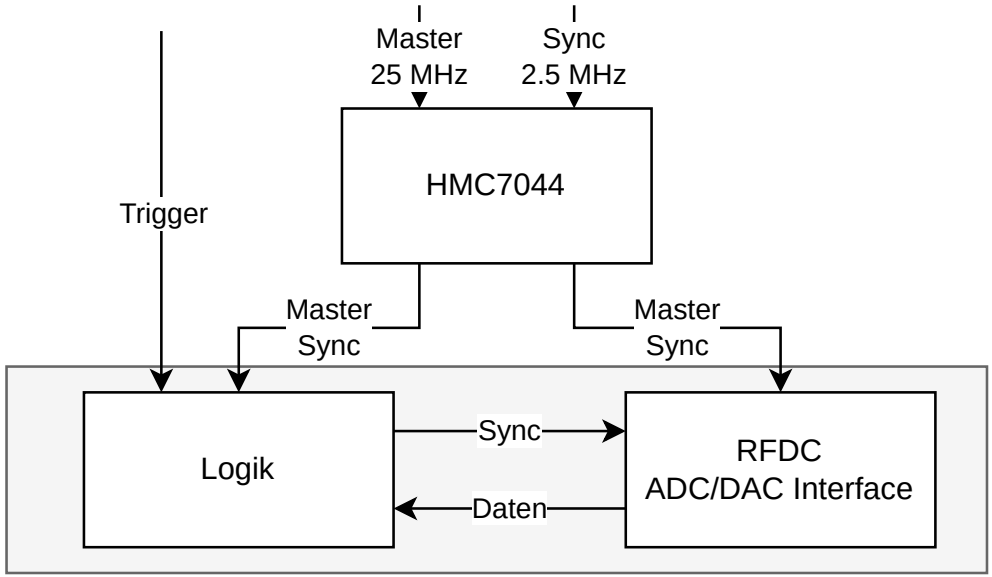
CLOCK

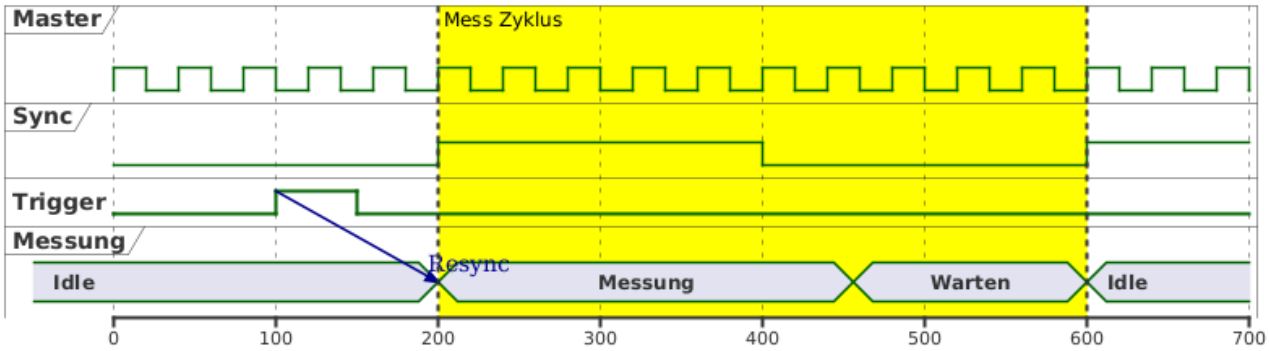
Clock Tree



CLOCK

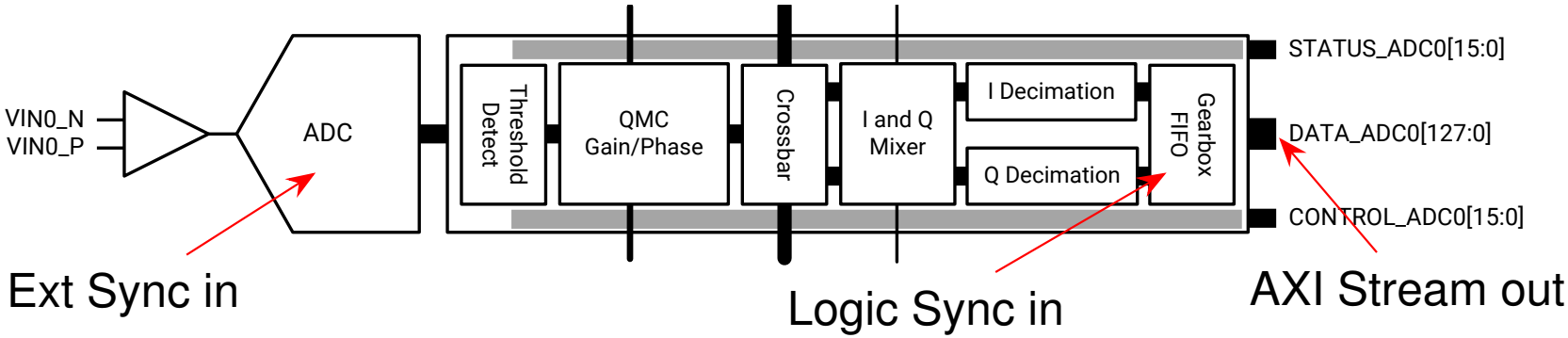
Synchronisierung





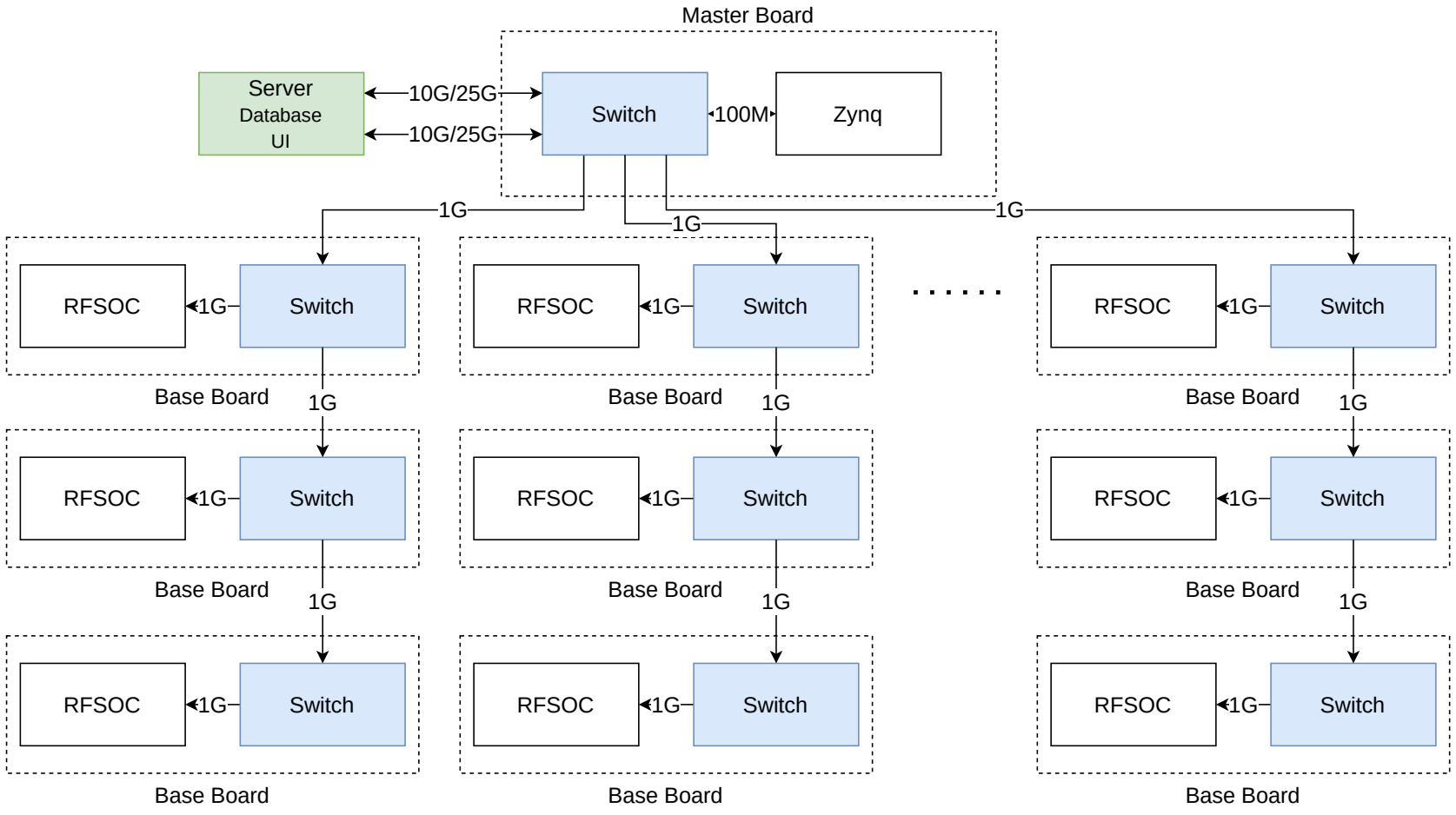
CLOCK

Multi-Tile-Sync (MTS) im Xilinx RFSOC RFDC

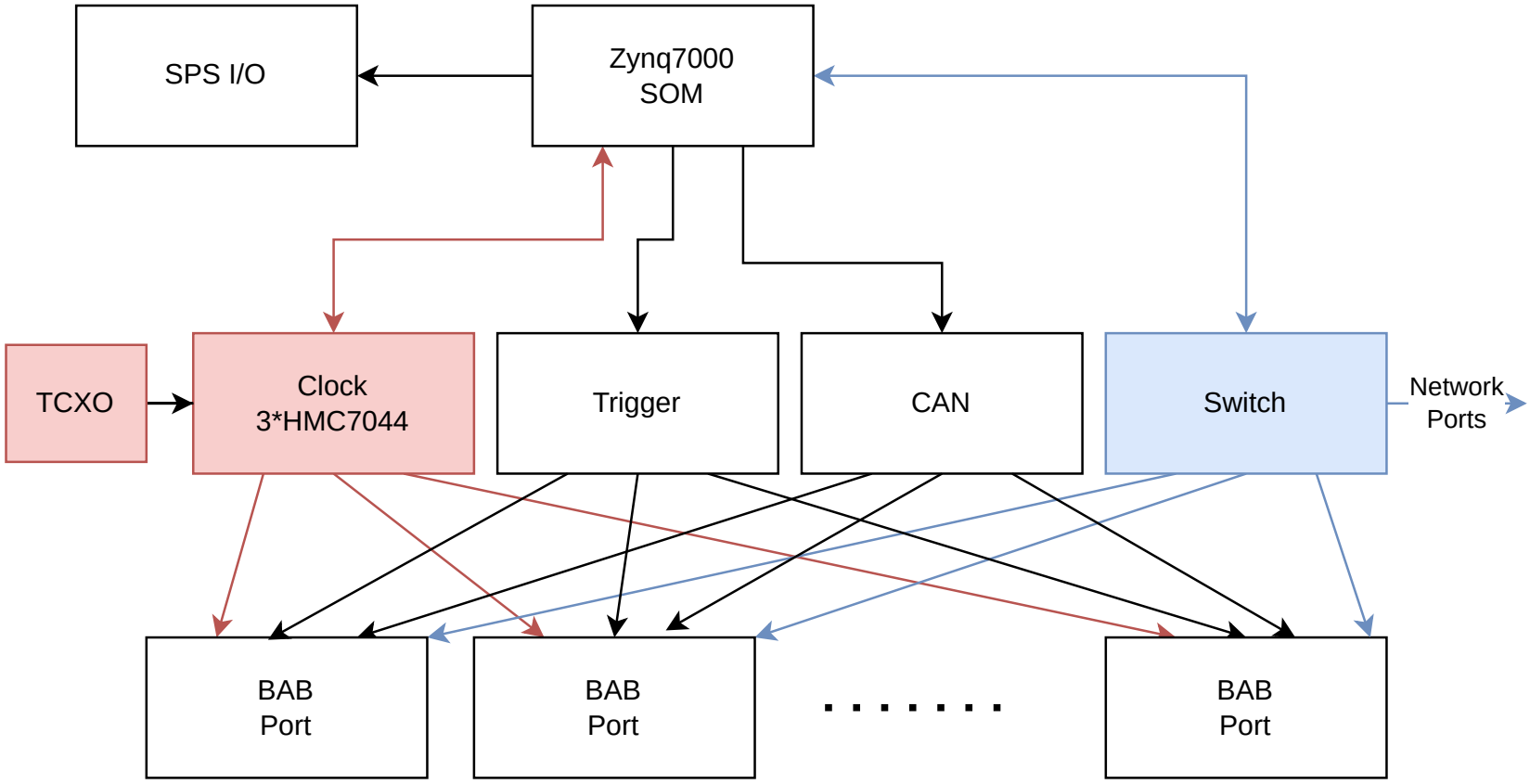


- synchronisiert alle internen ADC+DAC Datenströme zum Sync signal
- das Sync-Signal in der Logik synchronisiert den FIFO Ausgang (AXI Stream)
- Verzögerung zwischen ADC Eingang und FIFO Ausgang kann nach MTS ausgelesen werden oder durch MTS eingestellt werden

NETZWERK

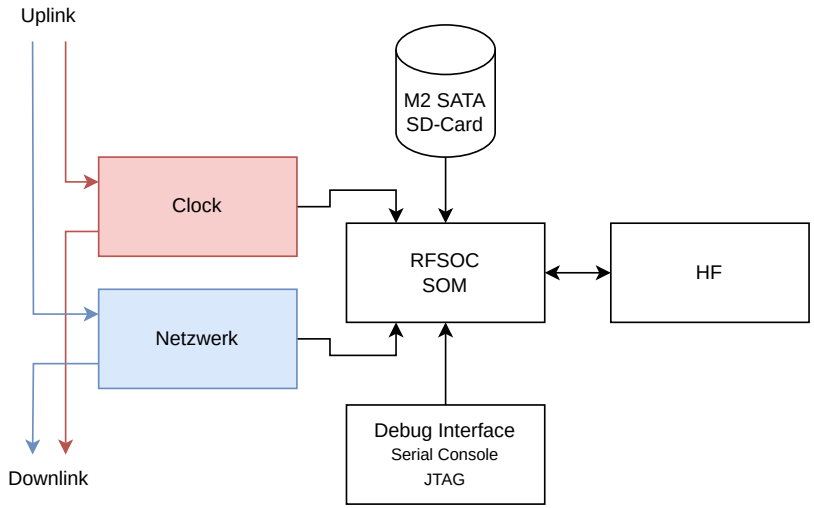

Master Board
Server
Database
UI
10G/25G
10G/25G
Switch
100M
Zynq
1G
1G
1G
RFSOC
1G
Switch
Base Board
1G
RFSOC
1G
Switch
Base Board
1G
RFSOC
1G
Switch
Base Board
1G
RFSOC
1G
Switch
Base Board
1G
RFSOC
1G
Switch
Base Board
1G
RFSOC
1G
Switch
Base Board
1G
RFSOC
1G
Switch
Base Board
1G
RFSOC
1G
Switch
Base Board

MASTER BOARD



BASE BOARD

Hardware

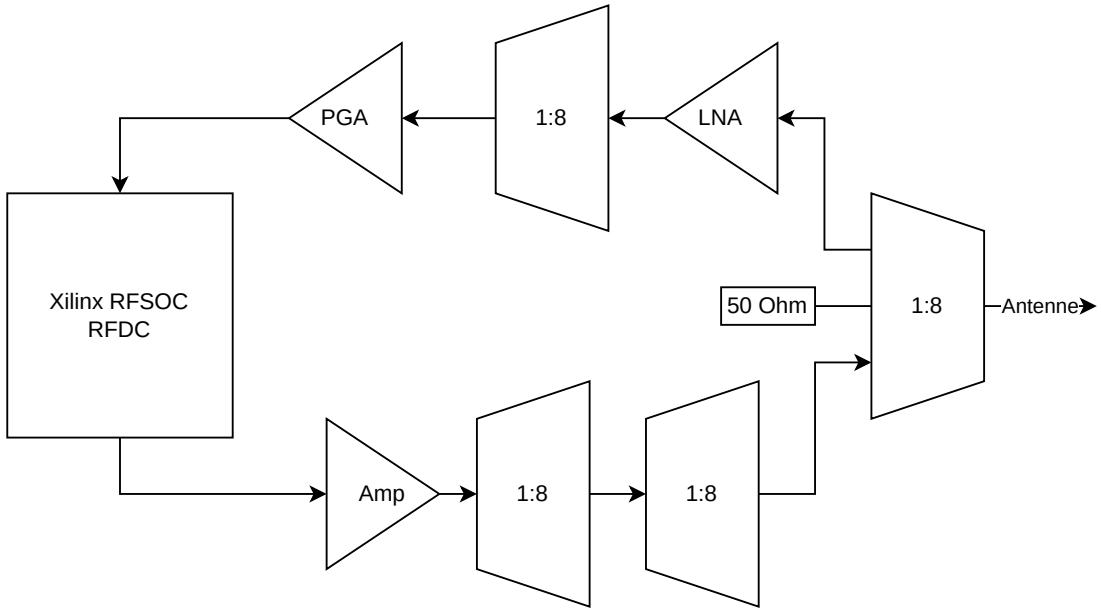


BASE BOARD

Hardware

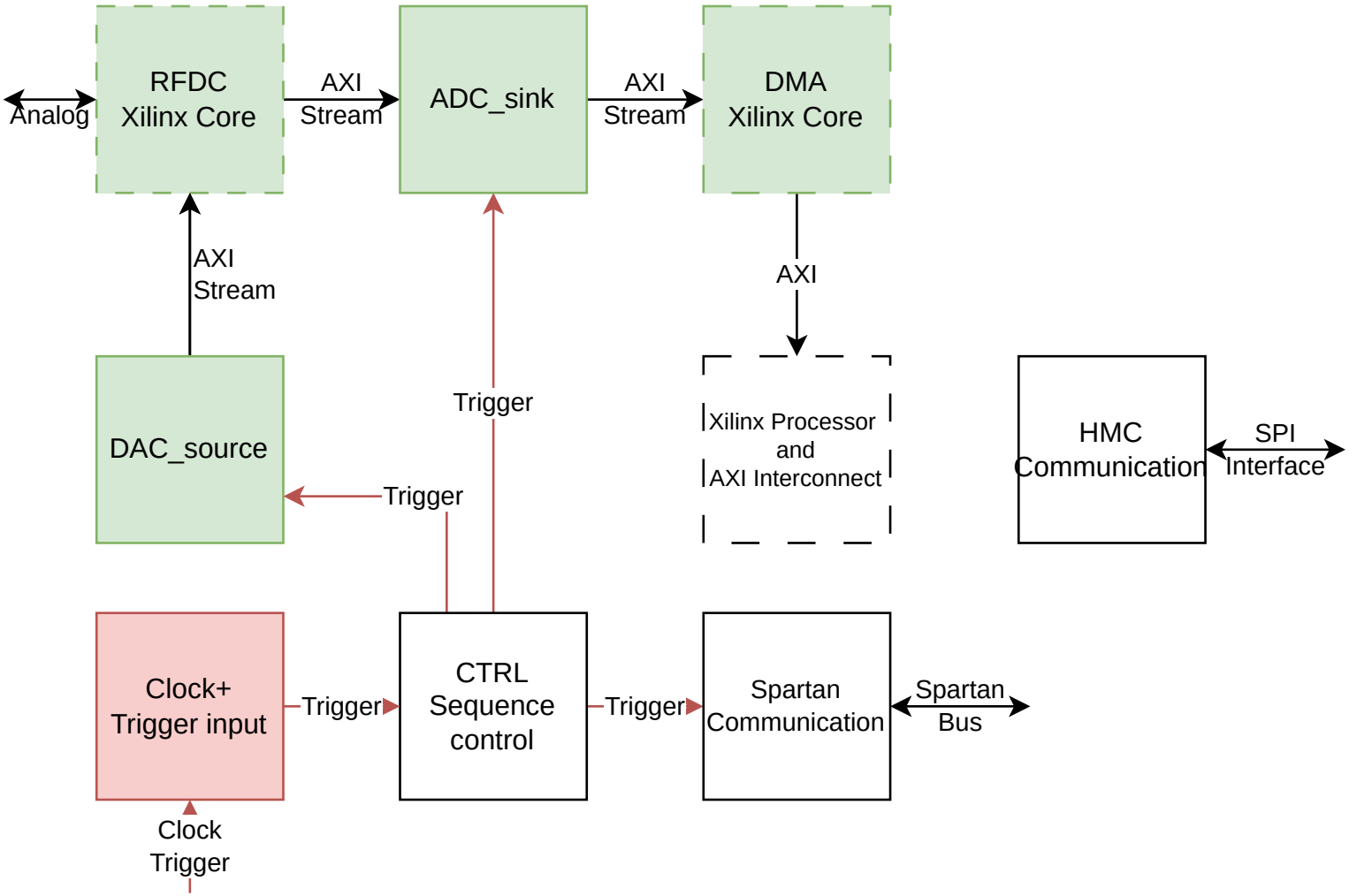
HF Signalpfad:

- 8 ADC Pfade
  für parallele Messung
  von 8 Antennen
- 1 DAC Pfad, nur
  eine Antenne sendet
- Steuerung
  durch extra FPGA
  - ein SPI Bus für 8
    PGAs
  - Pro Antenne 2 DOs
    pro Multiplexer = 128
    DO
  - 17 8fach Multiplexer
    mit je 3 DO = 91 DO



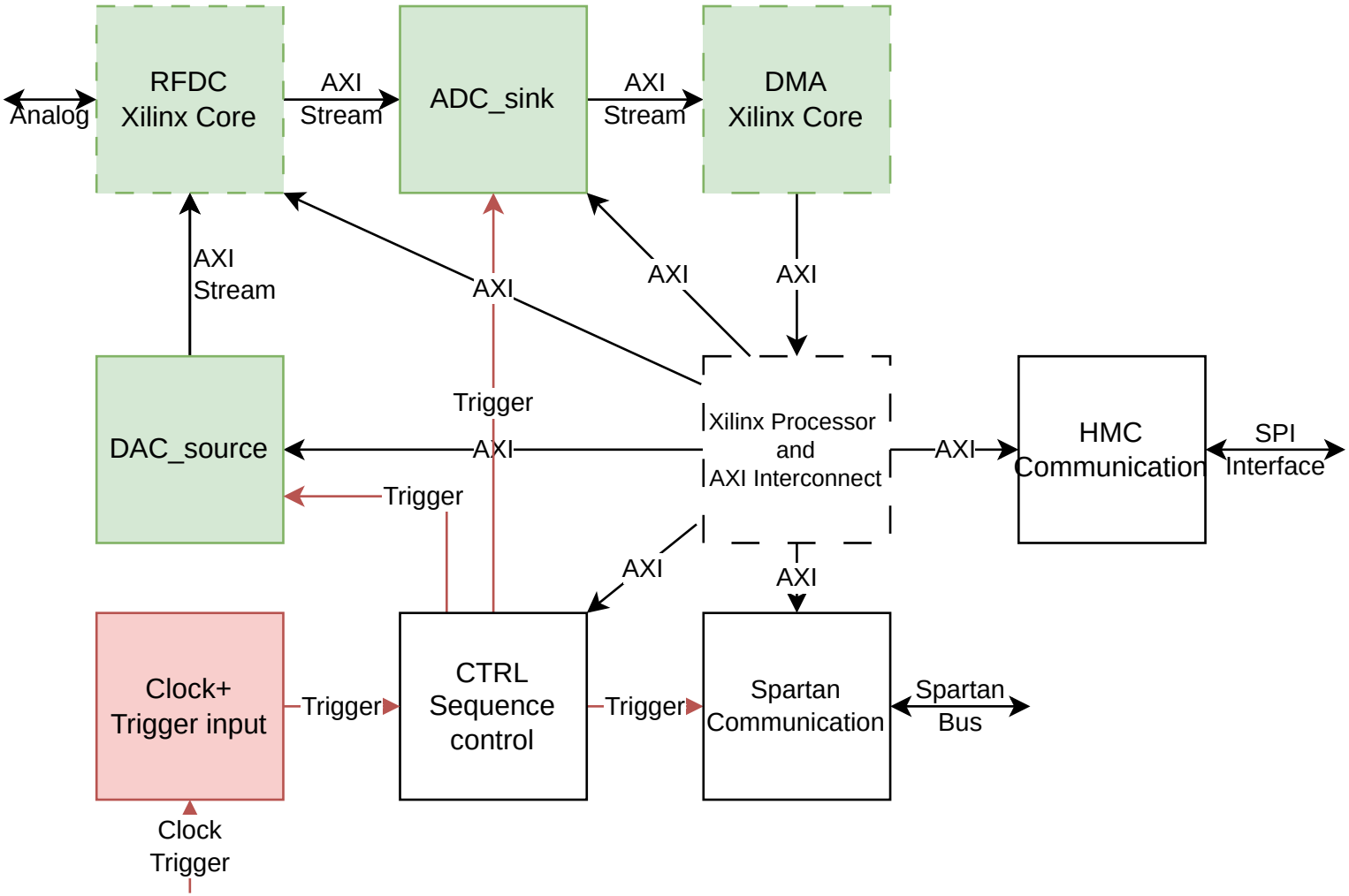
BASE BOARD

Logik



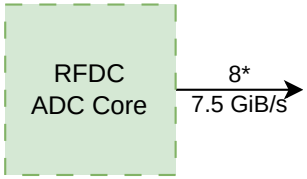
BASE BOARD

Logik



BASE BOARD

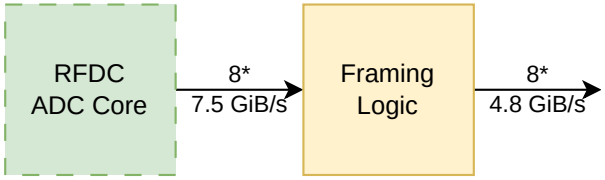
ADC



- 8 ADCs * 16 Bit * 4 GS/s = 60 GiB/s

BASE BOARD

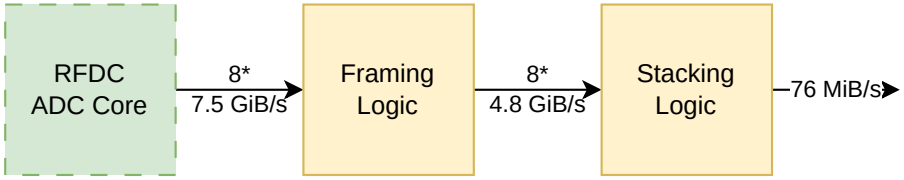
ADC



- 8 ADCs * 16 Bit * 4 GS/s = 60 GiB/s
- 400 ns/Frame $\Rightarrow$ 2,5 MHz, 1024 Samples/Frame
- 256 ns Messzeit, 144 ns Pause
- 8 ADCs * 1024 Samples * 16 Bit / 400 ns = 38 GiB/s

BASE BOARD

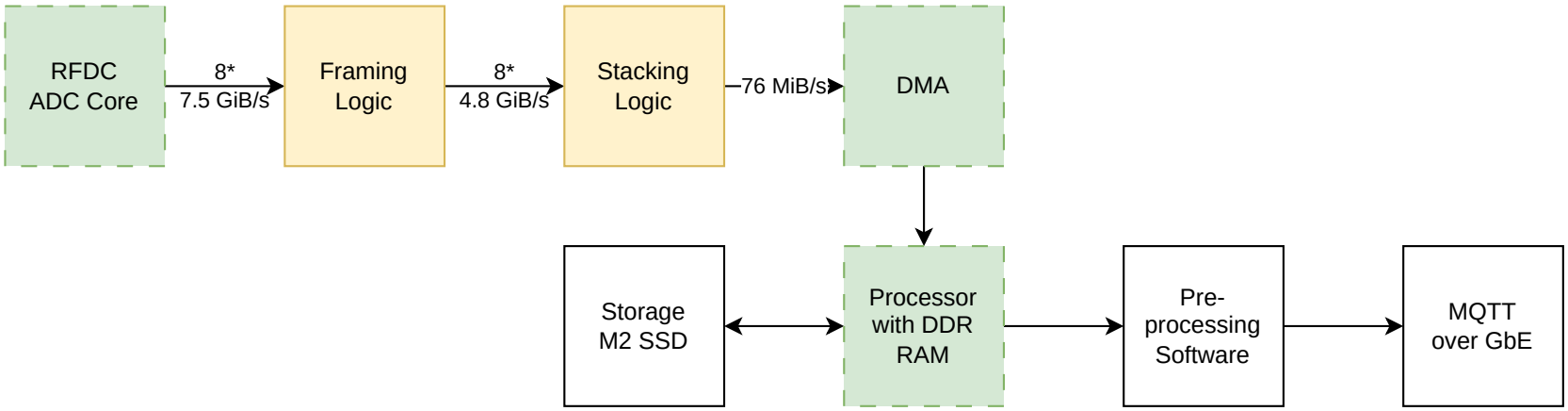
ADC



- 8 ADCs * 16 Bit * 4 GS/s = 60 GiB/s
- 400 ns/Frame $\Rightarrow$ 2,5 MHz, 1024 Samples/Frame
- 256 ns Messzeit, 144 ns Pause
- 8 ADCs * 1024 Samples * 16 Bit / 400 ns = 38 GiB/s
- 1024 Stacks, Erweiterung auf 32 Bit = 76 MiB/s

BASE BOARD

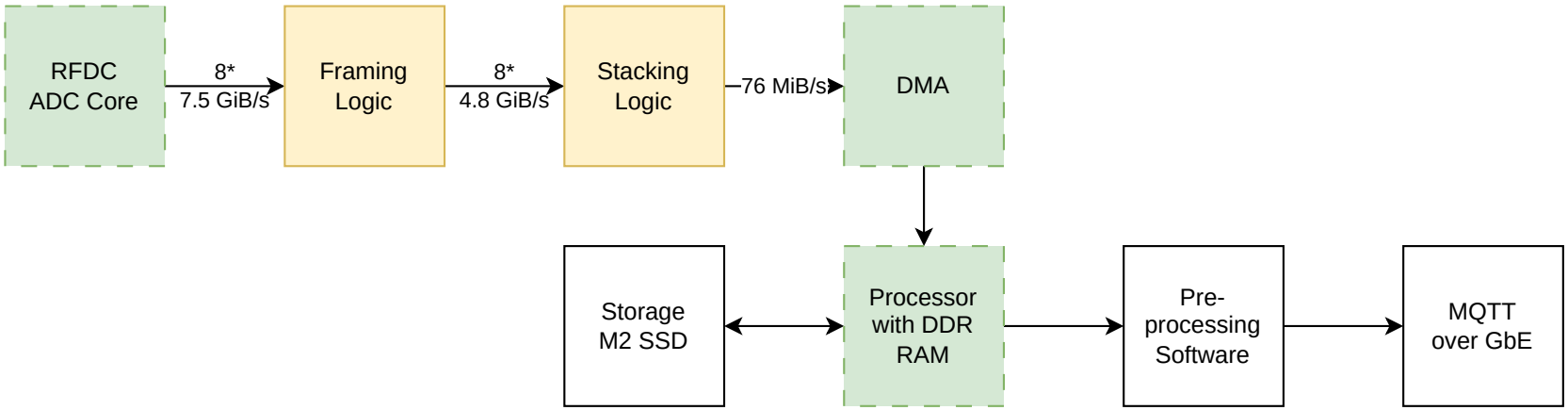
ADC



- 8 ADCs * 16 Bit * 4 GS/s = 60 GiB/s
- 400 ns/Frame $\Rightarrow$ 2,5 MHz, 1024 Samples/Frame
- 256 ns Messzeit, 144 ns Pause
- 8 ADCs * 1024 Samples * 16 Bit / 400 ns = 38 GiB/s
- 1024 Stacks, Erweiterung auf 32 Bit = 76 MiB/s
- von allen 39 Base Boards: 2,9 GiB/s

BASE BOARD

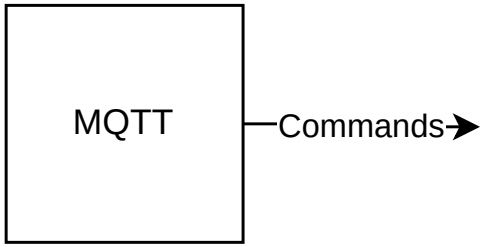
ADC



- 8 ADCs * 16 Bit * 4 GS/s = 60 GiB/s
- 400 ns/Frame $\Rightarrow$ 2,5 MHz, 1024 Samples/Frame
- 256 ns Messzeit, 144 ns Pause
- 8 ADCs * 1024 Samples * 16 Bit / 400 ns = 38 GiB/s
- 1024 Stacks, Erweiterung auf 32 Bit = 76 MiB/s
- von allen 39 Base Boards: 2,9 GiB/s
- Messzeit: 2496 Antennen * 8 Zeilen/Board * 1024 Stacks * 400 ns = 8,2 s

BASE BOARD

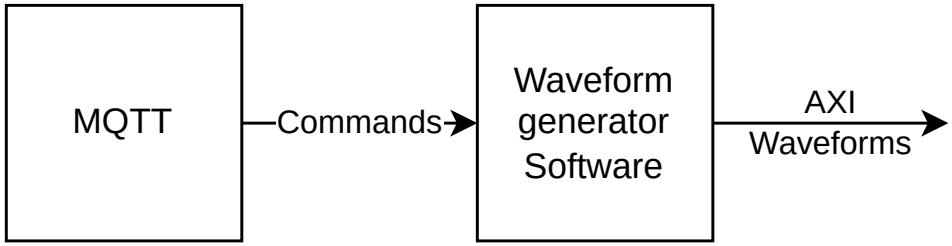
DAC



- MQTT überträgt vorab Parameter für Pulse (Form, Frequenz, Amplitude)

BASE BOARD

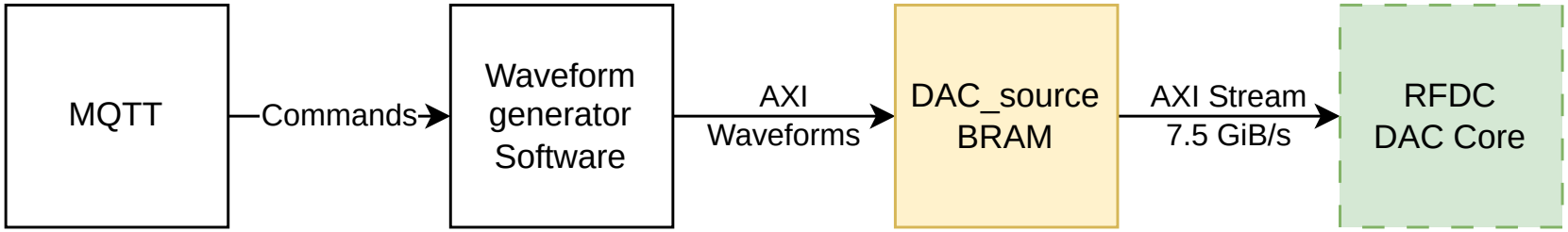
DAC



- MQTT überträgt vorab Parameter für Pulse (Form, Frequenz, Amplitude)
- Software generiert vor der Messung die Wellenformen und schreibt sie ins BRAM

BASE BOARD

DAC



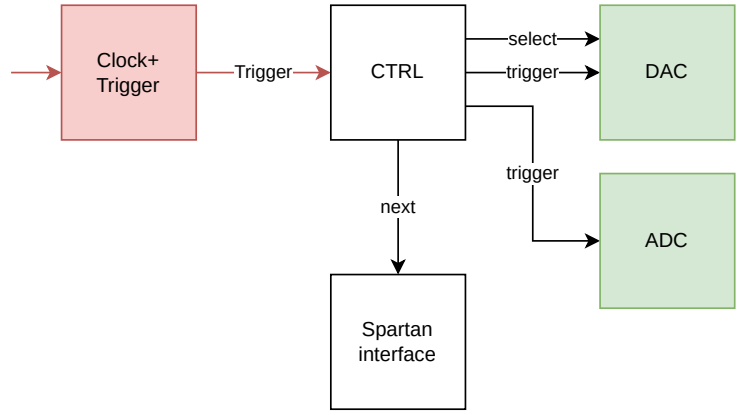
- MQTT überträgt vorab Parameter für Pulse (Form, Frequenz, Amplitude)
- Software generiert vor der Messung die Wellenformen und schreibt sie ins BRAM
- BRAM wird ständig ausgelesen und an DAC übertragen

BASE BOARD

Timing

Messablauf in
FPGA-Logik, kein Eingriff vom Prozessor.

- CTRL
  synchronisiert Trigger auf Clock-Signal

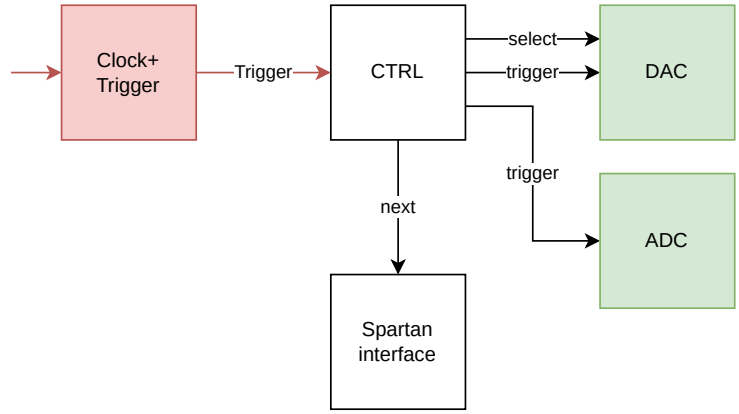


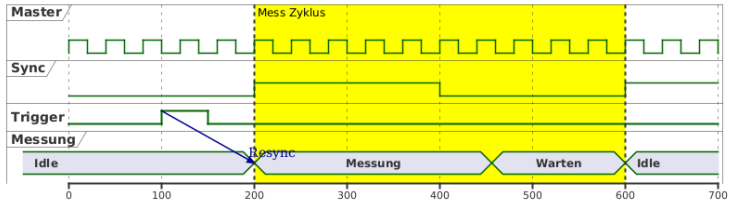
BASE BOARD

Timing

Messablauf in
FPGA-Logik, kein Eingriff vom Prozessor.

- CTRL
  synchronisiert Trigger auf Clock-Signal
- Trigger an DAC zur Pulsgenerierung
- Trigger an ADC als “Frame”-Start



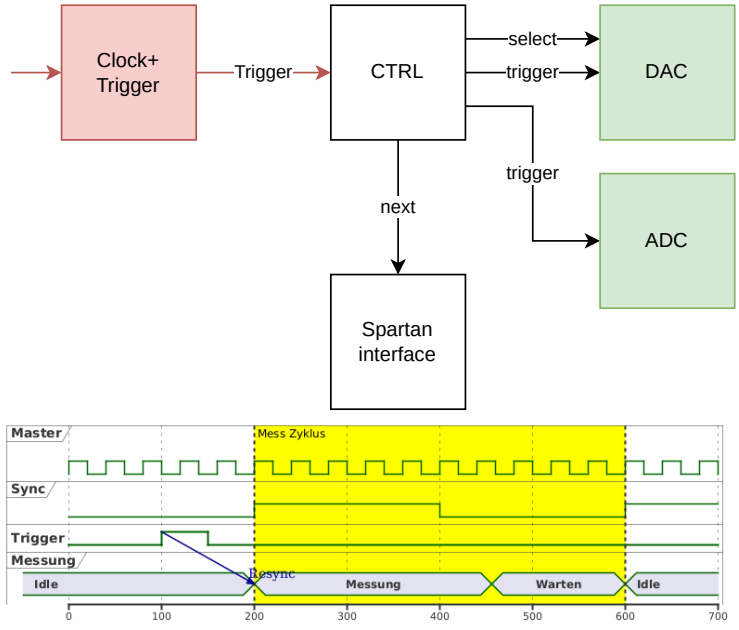


BASE BOARD

Timing

Messablauf in
FPGA-Logik, kein Eingriff vom Prozessor.

- CTRL
  synchronisiert Trigger auf Clock-Signal
- Trigger an DAC zur Pulsgenerierung
- Trigger an ADC als “Frame”-Start
- nach 1024 Stacks Signal an Spartan
  um auf nächste Antennengruppe zu
  schalten (“next”)

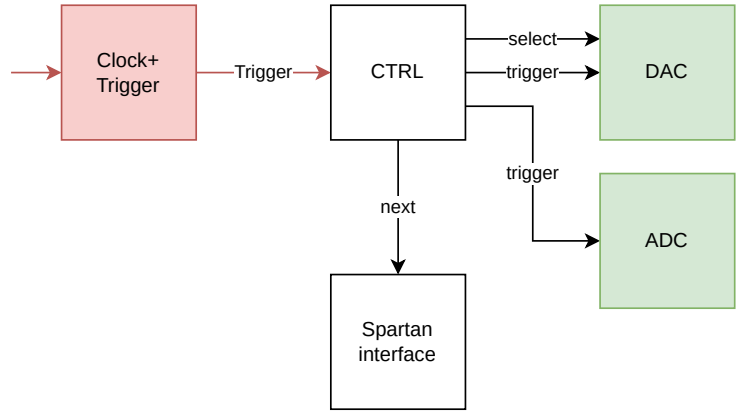


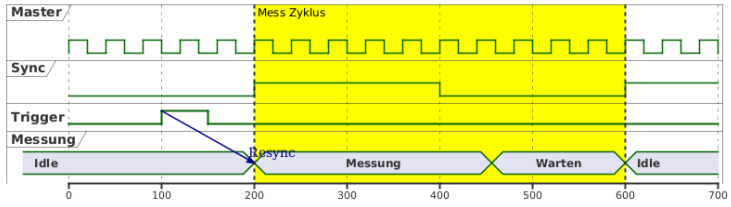
BASE BOARD

Timing

Messablauf in
FPGA-Logik, kein Eingriff vom Prozessor.

- CTRL
  synchronisiert Trigger auf Clock-Signal
- Trigger an DAC zur Pulsgenerierung
- Trigger an ADC als “Frame”-Start
- nach 1024 Stacks Signal an Spartan um auf nächste Antennengruppe zu schalten (“next”)
- nach 8 Zyklen Signal an DAC um nächste Wellenform zu laden (“select”)



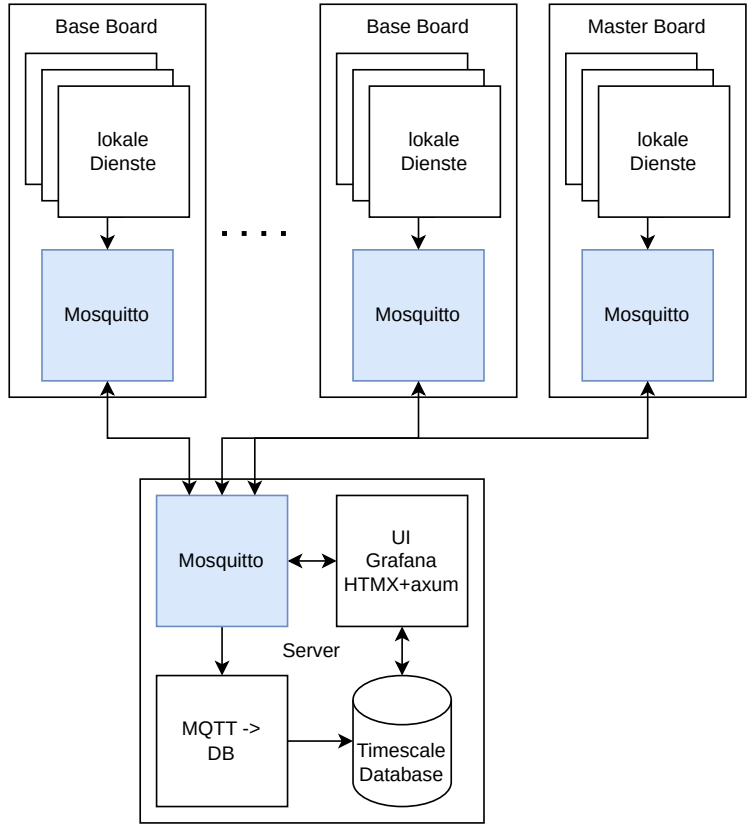


Prozessor bereitet die Sequenz vor und kümmert sich während der Messung nur um den Datentransfer.

MQTT

MQTT Broker:

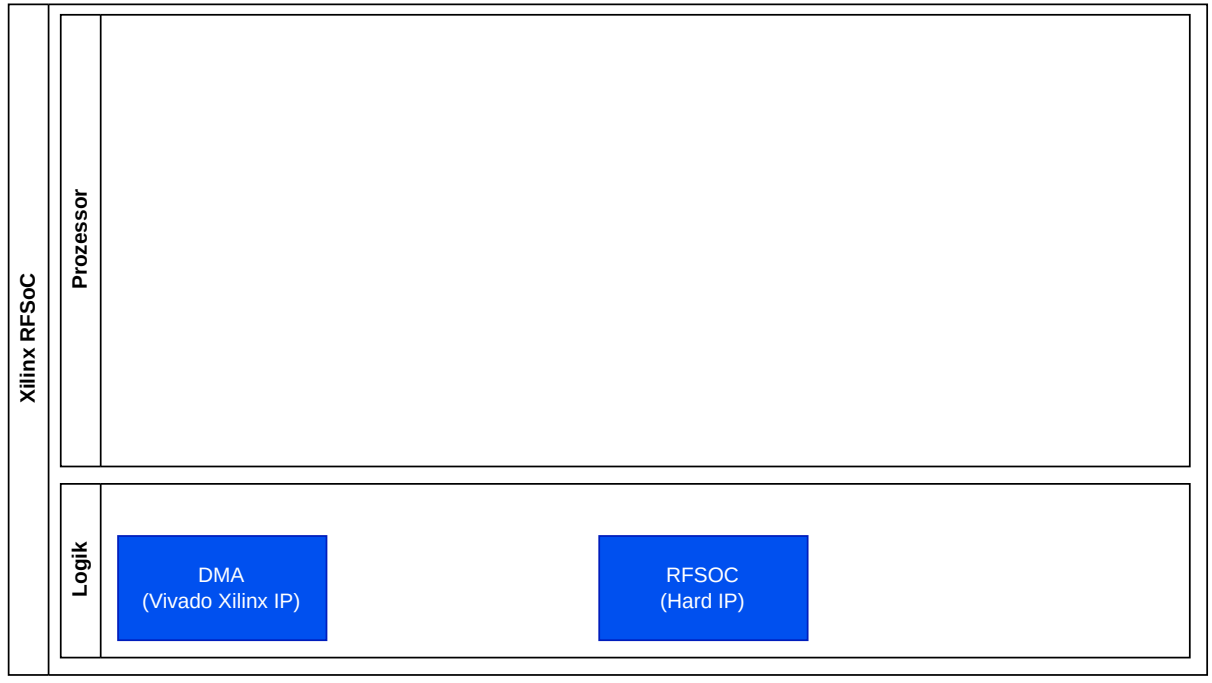
- Server: zentraler Broker
- Master Board: Verbindung zum Server, lokale Kommunikation innerhalb des Master Board
- pro Base Board: Verbindung zum Server, lokale Kommunikation innerhalb des Base Board



SOFTWARE

Build Pipeline:

- Vivado erzeugt Logik



SOFTWARE

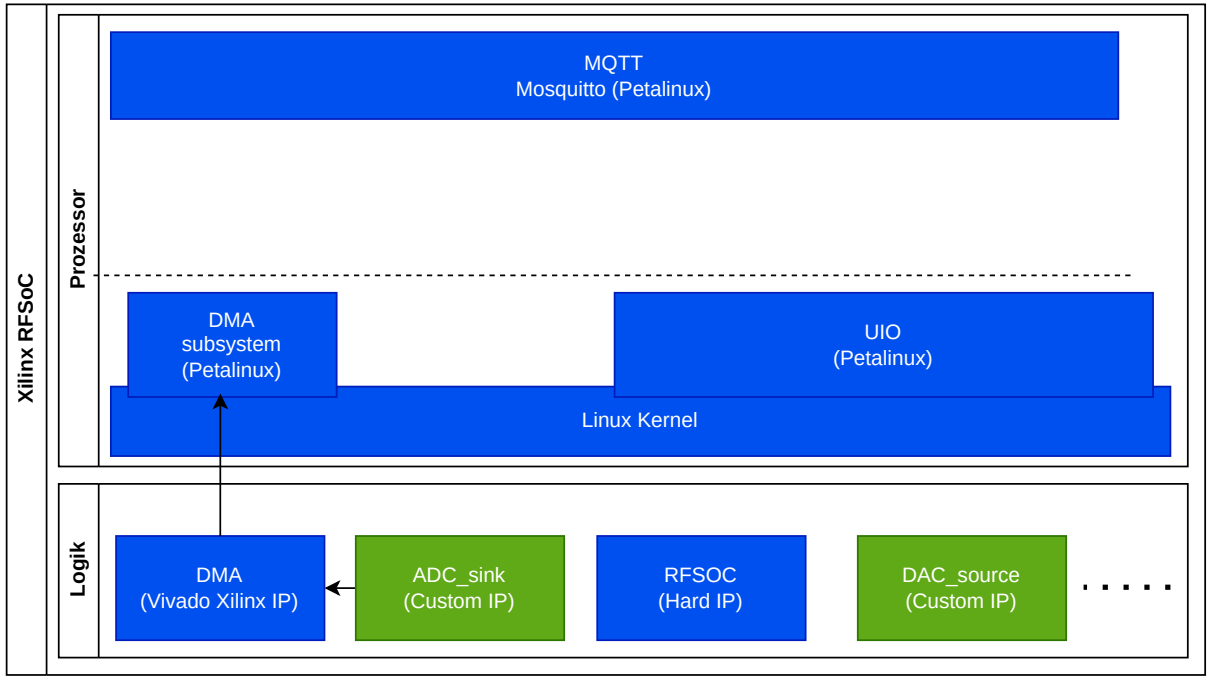
Build Pipeline:

- Vivado
  erzeugt Logik
- eigene
  IP Cores für Daten
  und Steuerung

SOFTWARE

Build Pipeline:

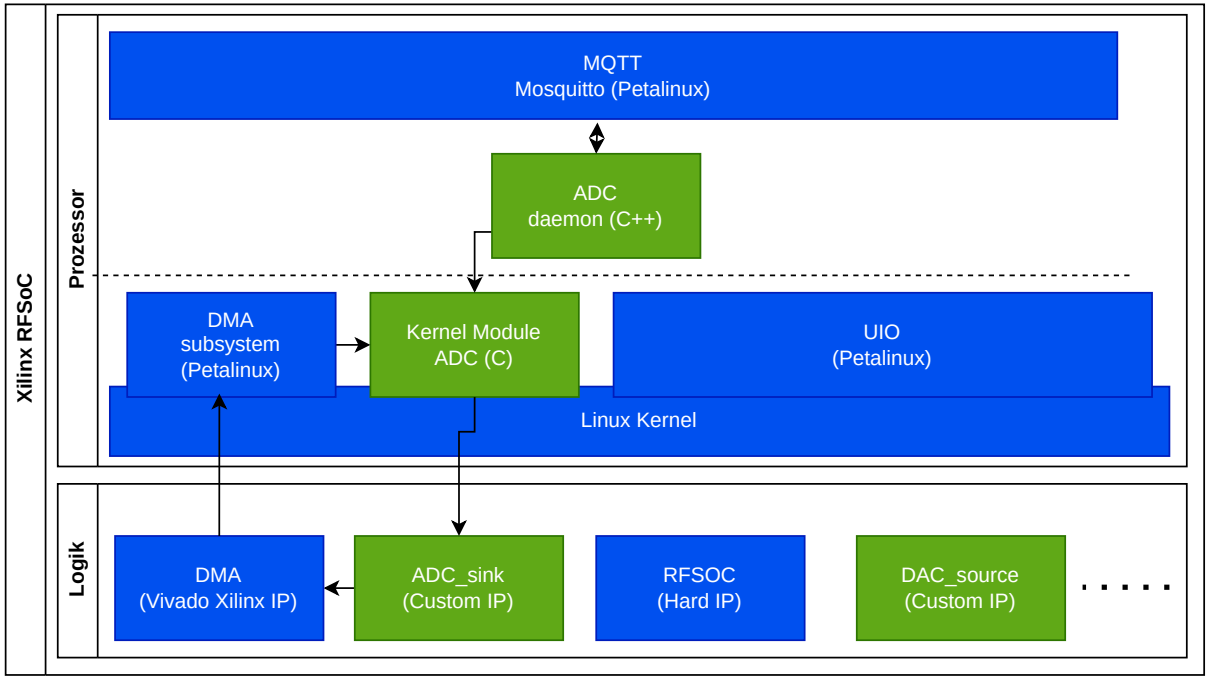
- Vivado erzeugt Logik
- eigene IP Cores für Daten und Steuerung
- Petalinux erzeugt Linux System



SOFTWARE

Build Pipeline:

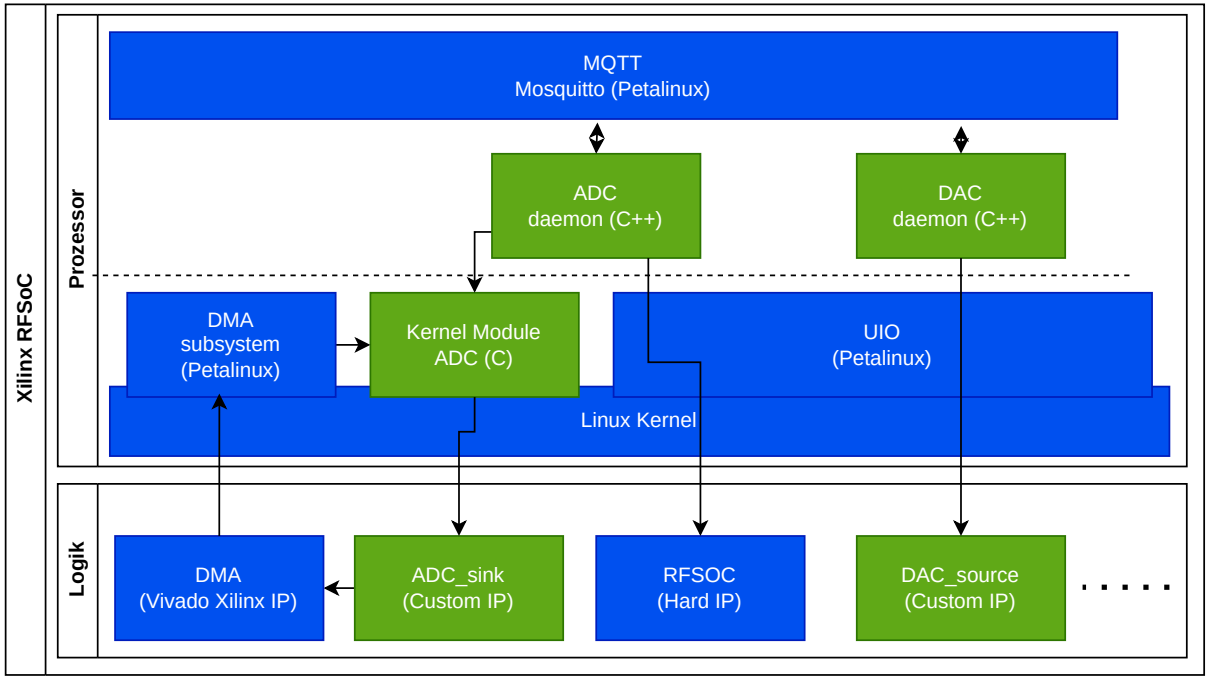
- Vivado erzeugt Logik
- eigene IP Cores für Daten und Steuerung
- Petalinux erzeugt Linux System
- Kernel Modul für DMA Integration



SOFTWARE

Build Pipeline:

- Vivado
  erzeugt Logik
- eigene
  IP Cores für Daten
  und Steuerung
- Petalinux erzeugt
  Linux System
- Kernel Modul für
  DMA Integration
- UIO
  für restliche Logik



ZUSAMMENFASSUNG

- Modularer Aufbau erlaubt skalierbares System
- geforderte Spezifikation erreichbar
- Messablauf in Logik erlaubt sehr schnellen Betrieb
- MQTT Steuerung und Überwachung funktioniert

AUSBLICK

- Störsignale müssen noch genauer untersucht werden
- Timing-Stabilität des gesamten Systems unter Betriebsbedingungen muss validiert werden
- Datentransfer während der Messung muss noch validiert werden

Work in progress....