

# **HERMESS Measurment System Specification (MESS) II**

**Documentation**

Kai Robin Grimsman, M.Eng.

23. Februar 2021

# Inhaltsverzeichnis

# 1 Blockdiagramm

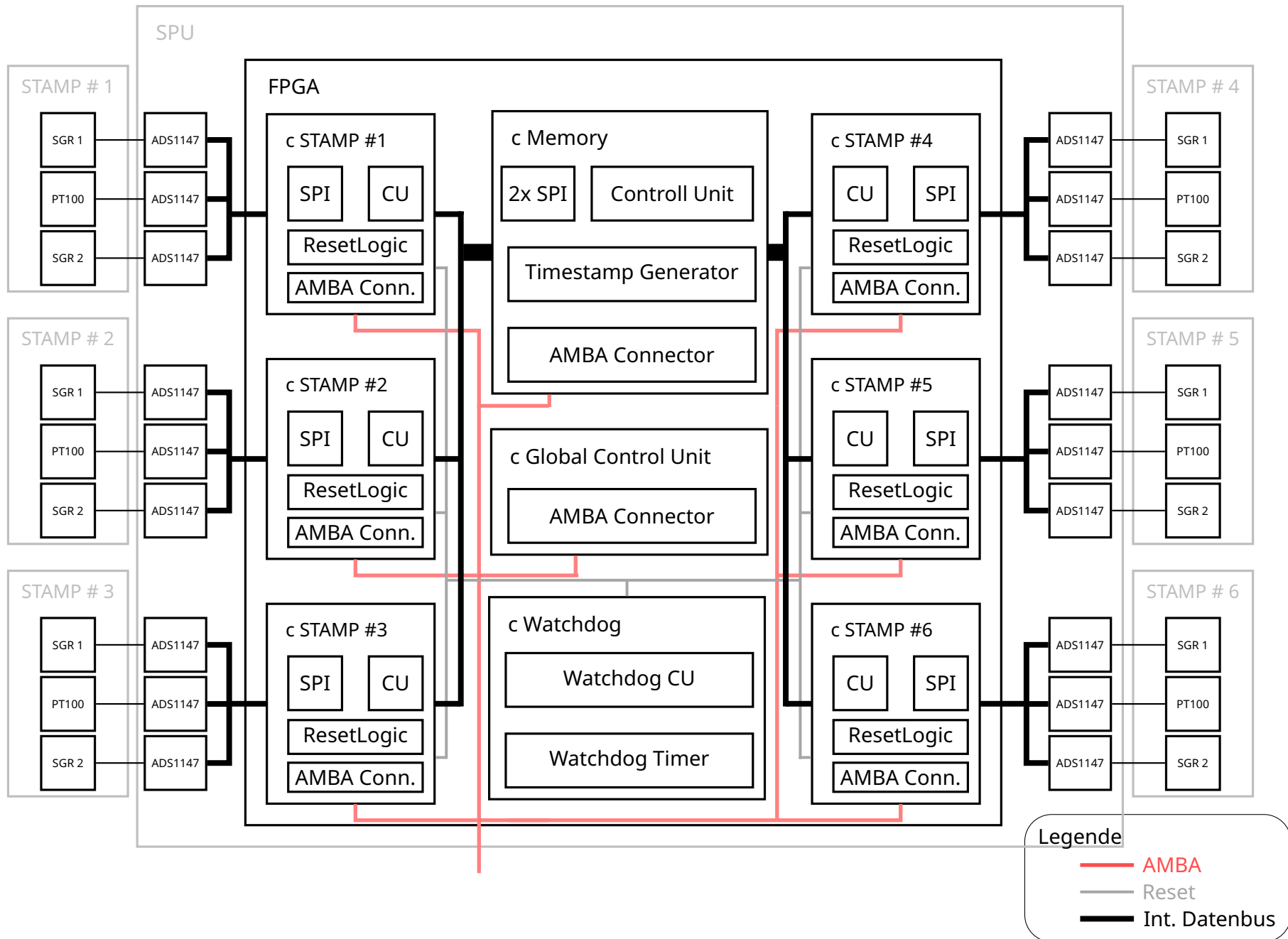


Abbildung 1.1: Systemübersicht

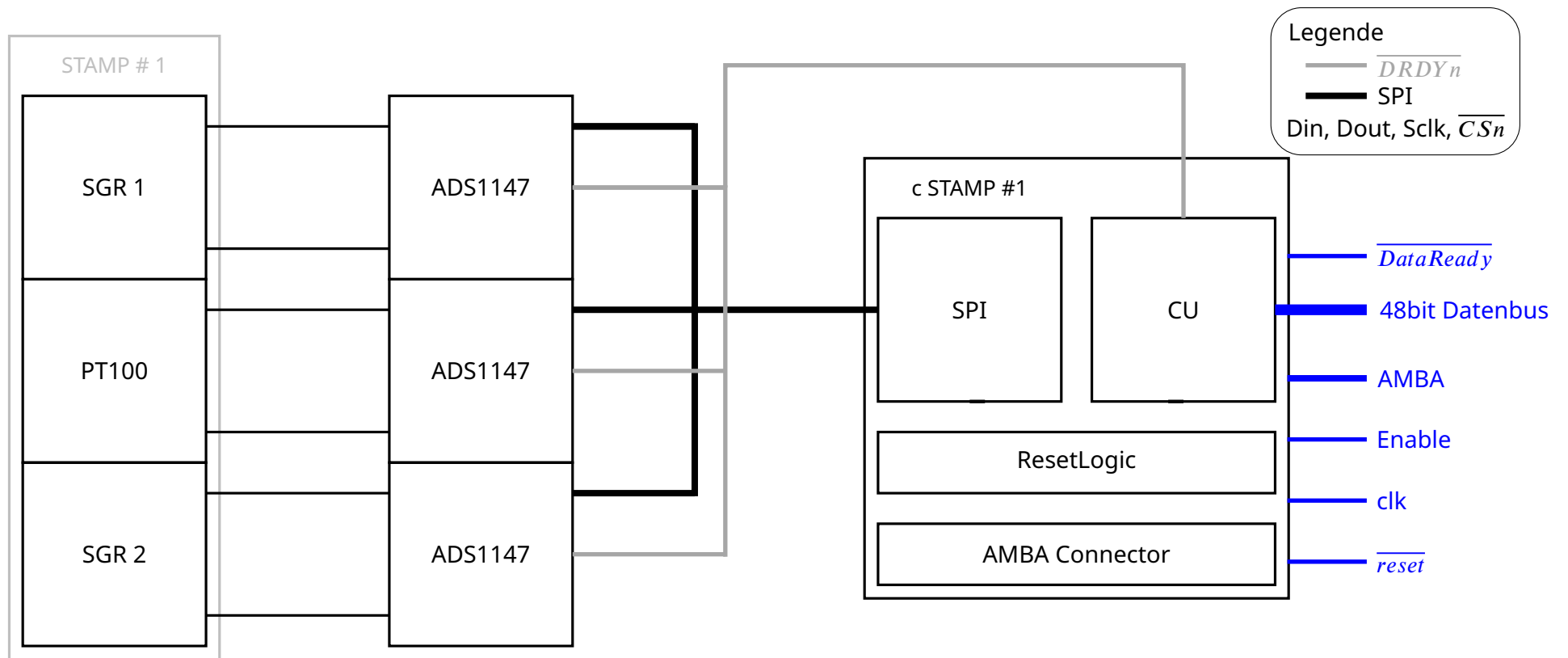


Abbildung 1.2: Messstelle, Digitalisierung, Datenverarbeitung

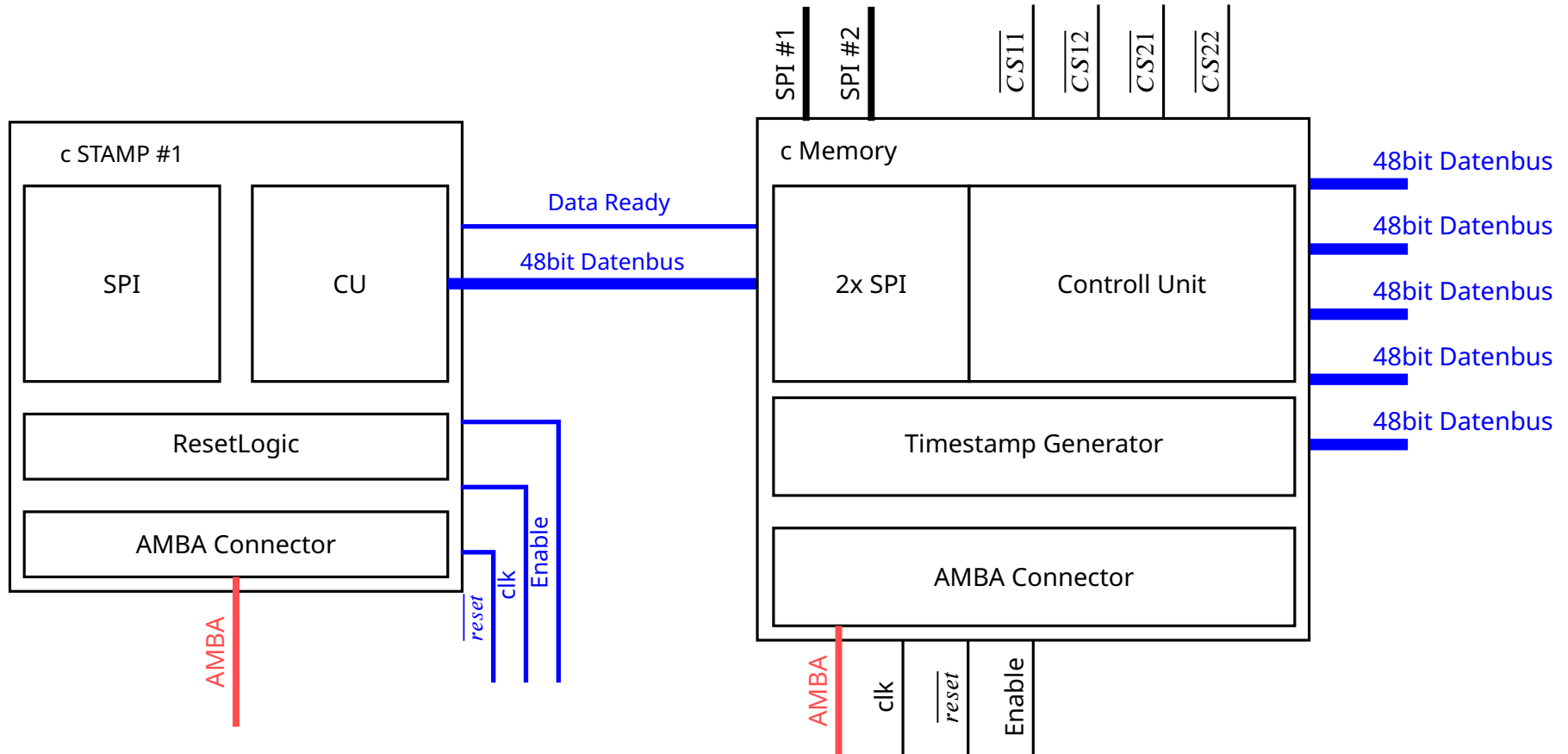


Abbildung 1.3: Interface c Stamp zu c Memory

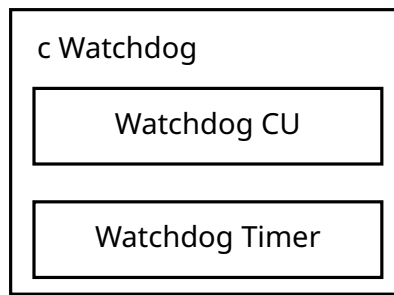


Abbildung 1.4: Watchdog

## 2 Beschreibung

### 2.1 Komponente: STAMP

Die STAMP Systemkomponente beschreibt einen Logikblock, der für die Kommunikation mit dem ADC 1147 zuständig ist. Um diese Kommunikation zu gewährleisten wird in jeder STAMP - Komponente ein SPI Core mit einer 8 - 16 Bit breiten Busanbindung implementiert. Im Vergleich zum bisherigen System würde die Anzahl der genutzten SPI - Systeme nicht erhöht werden. Die Systeme bedienen sich der FPGA inne liegenden Parallelität.

#### 2.1.1 Beschreibung

##### In- / Outputs

- SPI [3]: Der klassische SPI Bus bestehend aus DIN, DOUT, SCLK.

##### Input

- $\overline{DRDY1,2,3}$  Interrupt vom ADC1147, fallende Flanke. Löst internen Process aus.

##### Outputs

- $\overline{CS1,2,3}$  Chip Select der ADC 1147
- $\overline{RESET1,2,3}$  Asynchroner Reset der ADC 1147
- Data [48] Bus zur Memory Unit.
- Finished, steigende Flanke. Alle Daten wurden über SPI geholt und liegen an dem Data - Bus an.

#### 2.1.2 Beschreibung STAMP.Cu ControllUnit

Die ControllUnit, kurz CU, steuert das Finished Signal. Ebenso erfolgt die Steuerung des SPI Cores über die ControllUnit. Die CU registriert die fallende Flanke der ADC1147, auf Basis der fallenden Flanke des Interrupts wird der SPI Process gestartet. Hierbei wird der entsprechende Befehl an den ADC1147 gesendet und auf die empfangenden Daten gewartet. Ebenfalls stellt die CU sicher, dass der PT100 Sensor des STMAPs min. mit einer Frequenz von 20 Hz gelesen wird. Die Daten der DMS ADCs plus das Datum des PT100 ADCs werden in einem 48 bit breiten Bus parallel zur Verfügung gestellt. Bei dem deutlich



langsamerem PT100 Datum wird das bestehenden Datum weiterverwendet. Nach dem alle drei Datums ermittelt wurden wechselt das Signal Finished auf High.

Ingesamt wird diese Konstruktion drei mal, für drei STAMP - Messstellen implementiert.

#### **Datenformant auf data[48]**

Bit 47 - 24: PT 100

Bit 23 - 16: DMS 2

Bit 15 - 0: DMS 1

### **2.1.3 ResetLogic**

Wird durch den Watchdog eine Unregelmäßigkeit festgestellt wird das asynchrone Signal, IntReset ausgelöst. Dieses Signal wird durch den STAMP Core an den ADC1147 weitergeleitet.

## **2.2 Komponente: Watchdog**

### **2.2.1 Beschreibung**

Der Watchdog implementiert einen internen Timer, wird dieser Timer ausgelöst, wird ein internes Reset - Signal generiert. Die Logik des Watchdogs lässt sich wie folgt beschreiben:

Wird durch einen STAMP binnen von 500ms kein Signal, Finished wechselt auf high, generiert, so wird der Reset für das Teilsystem ausgelöst.

Der Timer für das System muss entsprechend hoch dimensioniert werden. Eine entsprechend implementierter Prescaler ist vom Vorteil, aber nicht zwingend nötig.

## **2.3 Komponente: Memory**

### **2.3.1 Beschreibung**

Die Komponente hält 512 Byte Speicher vor, dieser Speicher entspricht der Größe zweier Speicher - Pages des Flashspeichers.

Ein Datenframe besteht aus 20 Bytes. Bytes 0 - 17: Daten der drei Messstellen. Bytes 18-19: 16 Bit CRC.

Ingesamt passen auf eine 256 Byte große Speicherseite 12 Datensätze. Insgesamt 16 Byte

bleiben für Metadaten über. Die Metadaten halten mindestens den Verweis auf den folgenden Datensatz in dem Format, insgesamt vier Byte: 2 Byte: Speichereinheits ID, Adresse auf der Einheit.

Des Weiteren können auf in den verbleibenden 12 Byte der Timestamp des ersten Datensatzes sowie der Timestamp des letzten Datensatzes gespeichert werden.

Timestamp, siehe Abschnitt ??.

### Inputs

- 3 x data [48]
- 3 x Finished

### Output

- 2 x SPI (DIN, DOUT, SCLK) je Flash double Die IC.
- 4 x  $\overline{CS1.1, 1.2, 2.1, 2.2}$

## 2.4 Komponente: Timestampgenerator

### 2.4.1 Beschreibung

Hauptaufgabe ist es das erstellen eines 32 bit langen Timestamps. 0 = SOE. Auflösung:  $100\mu s$ .

$$\frac{Timestamps}{s} = \frac{1s}{100\mu s} = 10000$$

$$T = \frac{Auflösung}{\frac{Timestamps}{s}} = \frac{2^{32} - 1}{10^{\frac{k}{s}}} = 423496s \approx 119h$$

Entsprechend der Auflösung sind ca. 119 Stunden Aufnahme mit dem System möglich. Zudem ermöglicht es die Auflösung von  $100\mu s$  der Unterscheidung, von wann der Eintag mit Referenz zu  $T = 0$ .

### Output

- Timestamp [32] Der ermittelte Timestamp.

## 2.5 Resoucesbedarf, geschätzt

Tabelle 2.1: Geschätze Anzahl der LogicElemente LEs

Name	LEs	Anzahl	Gesamt
SPI	300	4	1200
CU	400	3	1200
WDT	300	1	300
RESET	200	1	200
Memory	1500	1	1500
APB	400	1	400
<b>Gesamt</b>			<b>4800</b>

Ingesamt wird mit dem Design 48 % des FPGAs ausgenutzt. Grundlage für die Zahl ist die Anzahl der Logikeinheiten des FPGAs. In diesem Fall ist diese mit 10k angenommen.