Einheit zur anwendungsbezogenen Leistungsmessung für die RISC-V-Architektur

Tobias Scheipel
tobias.scheipel@gmail.com

Institut für Technische Informatik
Embedded Automotive Systems Group
Technische Universität Graz

16. November 2017

Agenda

- Einleitung und Motivation
- CPU Evaluation
- Performance Monitoring Unit – Hardware
  - Grundsatzüberlegungen
  - Architektur
  - Ausführung
- Eingliederung und Anwendung in *mosartMCU*-OS
- Messungen und Vergleiche

Einleitung und Motivation

- Performance Monitoring Unit (PMU) für das *mosartMCU*-Projekt
- Messung von Ausführungszeiten und Ereignissen
- Minimaler Softwareoverhead
  - Einfaches Interface zum Konfigurieren und Auslesen
  - Verwendung zur Optimierung des Task Schedule und der Performance
- Messungen ausschließlich in Hardware
  - Kein Eingriff in das bestehende System → externer Beobachter
  - Skalierbar und einfach erweiterbar (Verilog)
  - Wiederverwendbarkeit einzelner Hardwareelemente

CPU Evaluation

RISC-V und Peripherie

- Offene und freie Instruction Set Architecture (ISA)
- Entwickelt von der University of California, Berkeley
- Viele verschiedene Implementierungen
- Verwendete Variante: V-scale
  - 32-Bit-Core mit Integerunterstützung und Mult/Div-Einheit (RV32IM)
  - Implementierung in Verilog
- Zusätzliche Peripherie (nicht in V-scale vorhanden):
  - Dual-Port-RAM
  - GPIO
  - UART

[1],[2],[3]

Performance Monitoring Unit – Hardware

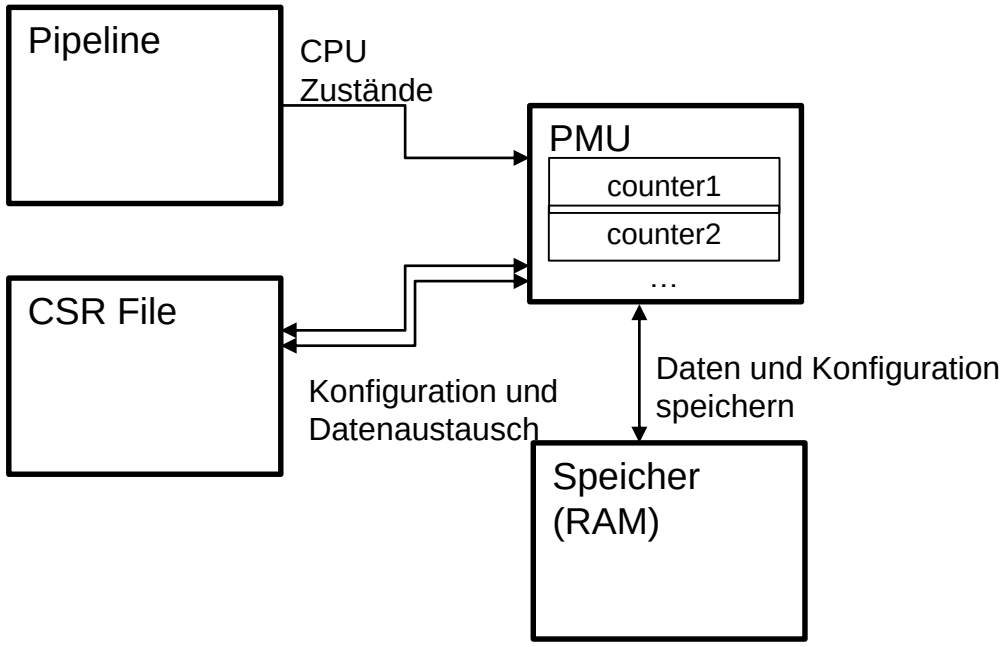
Grundsatzüberlegungen – Gewünschte Funktionen

- Diverse konfigurierbare Zählregister
- Zähler durch verschiedene Ereignisse getriggert:
  - Veränderung von Task Pointer (TP) oder Program Counter (PC)
  - Interrupts und externe Ereignisse
- Zählerarten
  - Allgemeine, globale Zähler
  - Taskabhängige Zähler
  - Taskzugehörige Zähler
- Skalierbar im Bezug auf Anzahl der Zählregister und der Konfigurationsregister pro Zähler
- Zählerstände beim Taskwechsel ohne Verzögerung im Speicher ablegen (Task-Awareness)

Performance Monitoring Unit – Hardware

Architektur

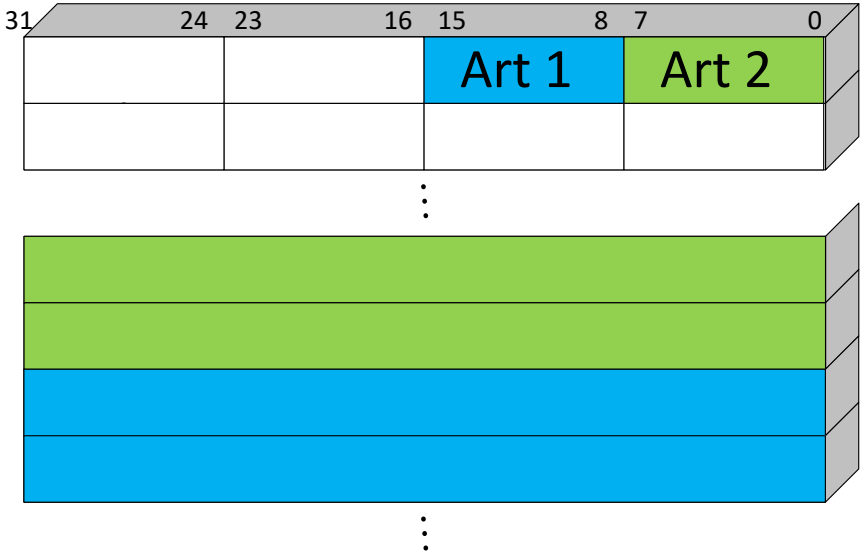
- Positionierung innerhalb des Pipeline-Moduls
  - Statusabfragen
- Control and Status Register (CSR) File [4]
  - Schnittstelle nach außen
  - R/W Konfiguration
  - R/(W) Zählwerte
- Speicherzugriff
- Zähleinheiten bestehend aus:
  - 64-Bit-Zählregister
  - Konfigurationsstruktur
  - Kombinatorische Logik



Performance Monitoring Unit – Hardware

Ausführung: Konfiguration

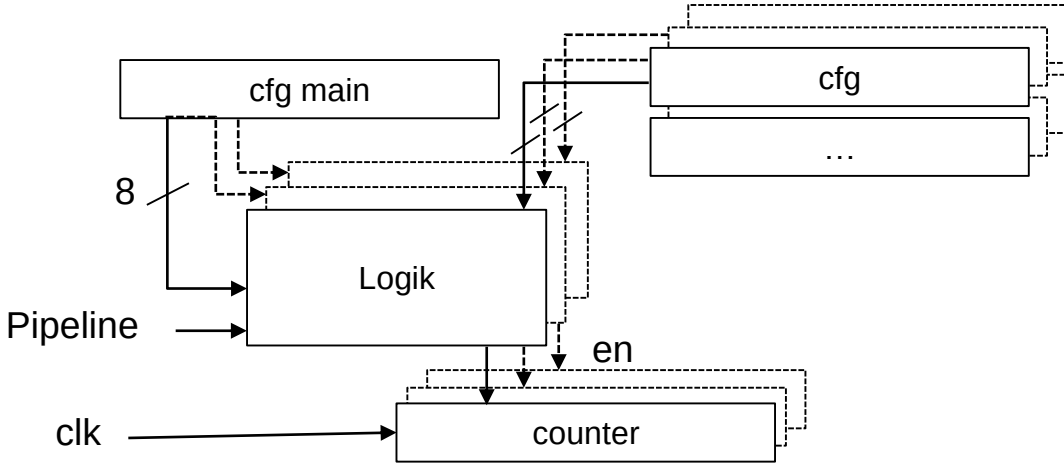
- Ausgeführt als CSR (read/write)
- Art des Zählers als Byte im Hauptkonfigurationsregister
  - Aufgeteilt in allgemeine, taskabhängige und taskzugehörige Zähler
  - Skaliert und padded 32-Bit-Register abhängig von der Anzahl der Zähler
- Zugehörige Konfigurationsregister
  - Konfiguration des Zählers
  - Skaliert
- Zugriff über CSR-Operationen



Performance Monitoring Unit – Hardware

Ausführung: Kombinatorische Logik der Zähler

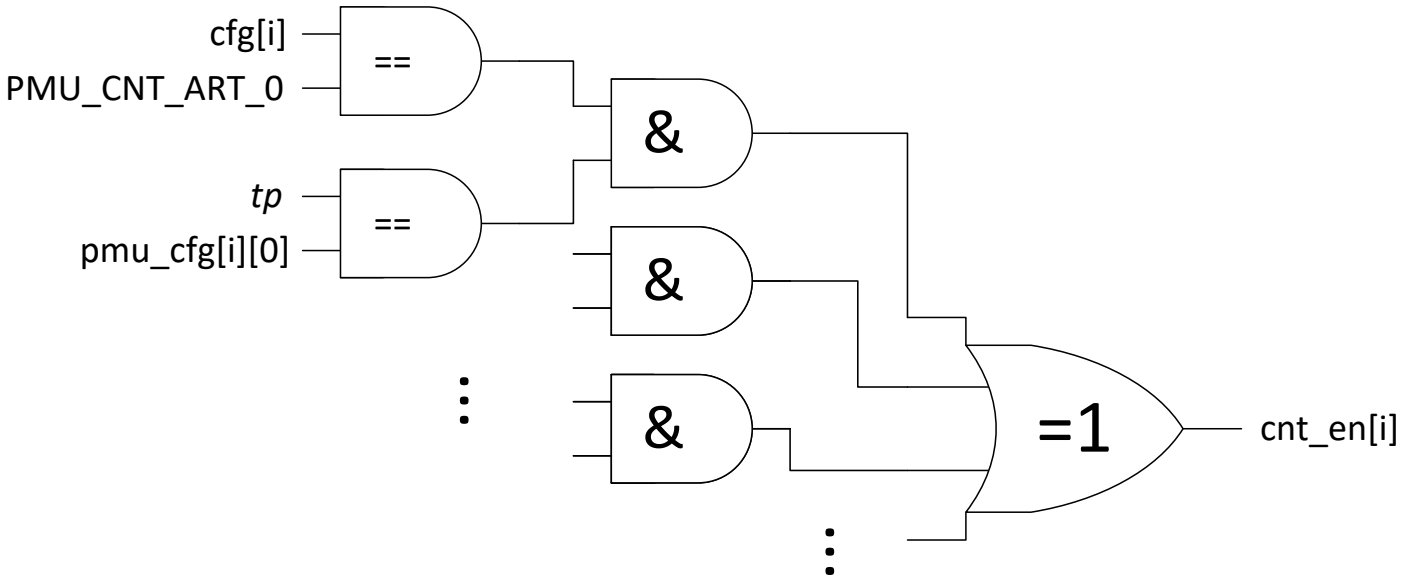
- Jede Zähleinheit hat eigenen Logikblock
- Zugriff auf
  - Statusinformationen von der Pipeline
  - Zugehöriges Byte aus dem Hauptkonfigurationsregister (Zählerart)
  - Alle Konfigurationsregister des korrespondierenden Zählers
- Erzeugt Enable-Flag um Zählregister zu inkrementieren
- Zählregister via CSR-Operationen auslesbar



Performance Monitoring Unit – Hardware

Ausführung: Kombinatorische Logik der Zähler

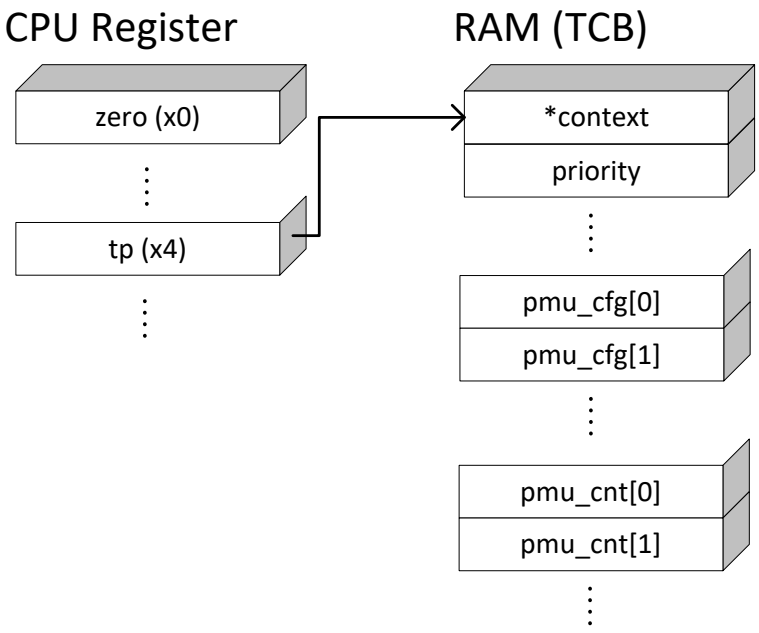
- Rein kombinatorischer Logikblock
- Skaliert je nach Anzahl der Zähleinheiten
- Einfach erweiterbar durch hinzufügen beliebiger Logikausdrücke
- Jede Zählerart abgebildet



Performance Monitoring Unit – Hardware

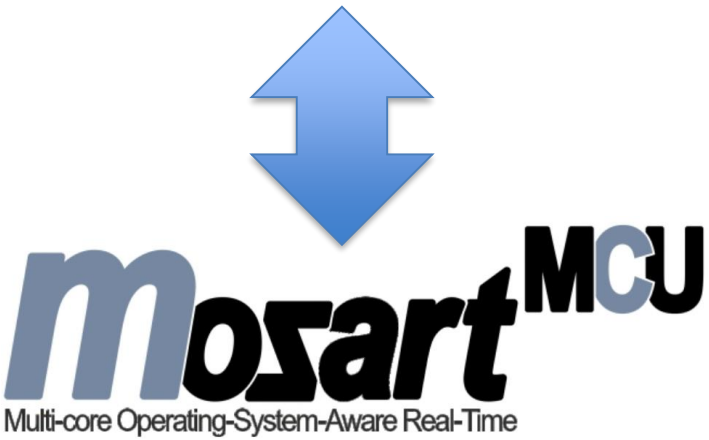
Ausführung: Speichern von Zählwerten und Konfigurationen

- CPU Register *tp (x4)* als Schnittstelle
- Betriebssystem legt Taskkontrollblock (TCB) an
- Bei Taskwechsel Werte aus-/einlagern
  - Zustandsautomat in Hardware
  - Eigenständiger Zugriff auf den Speicher
- Dual-Port-RAM [5]
  - Port A für Betriebssystem (Standardport)
  - Port B für Hardware/PMU



Eingliederung und Anwendung in *mosartMCU*-OS

- Betriebssystem verwaltet Taskkonstrukte über
  - Aktueller Task Pointer in CPU-Register *tp (x4)*
  - Taskkontrollblöcke beinhalten Werte der PMU
- Konfiguration erfolgt über CSR-Zugriffe
- Diverse Funktionen zum
  - Konfigurieren von Zählern
  - Auslesen der Zählwerte
  - Überwachen eines konkreten Tasks

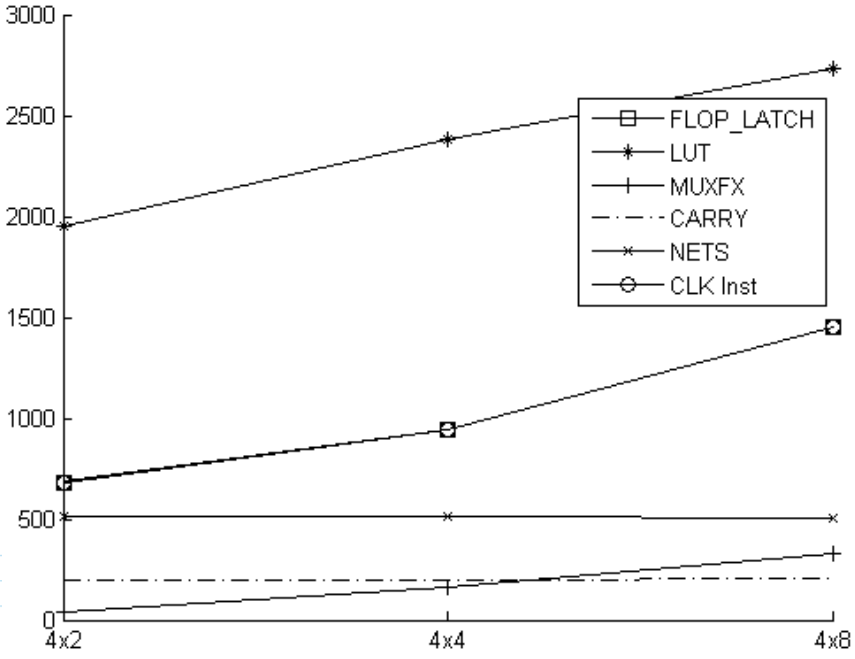


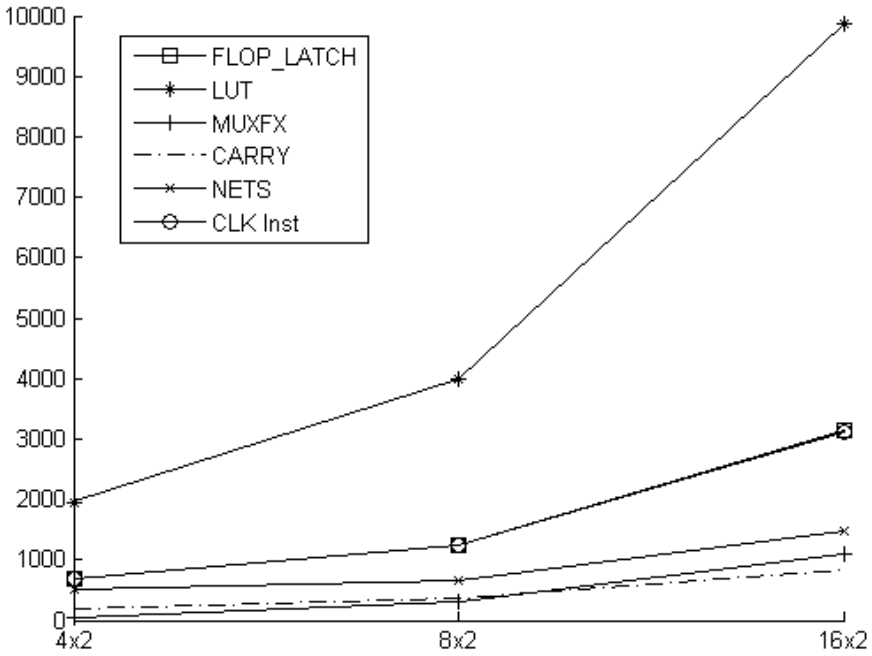
[1]

Messungen und Vergleiche

Ressourcenverbrauch am FPGA

Variation der Konfigurationsregister pro Zähleinheit





Variation der Zähleinheiten

Nexys4 Artix-7 FPGA Board

Messungen und Vergleiche

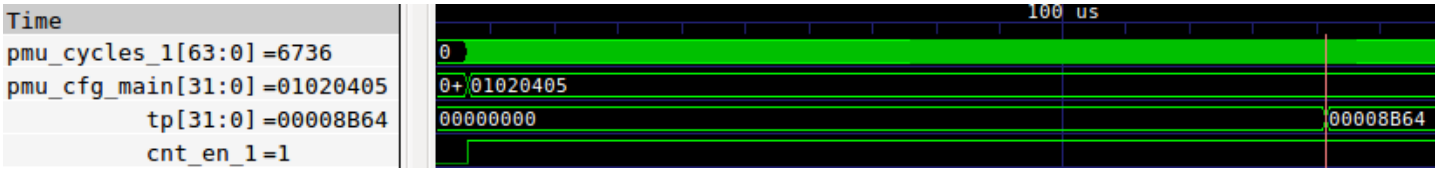
Messungen an Testsystemen: Messaufbau

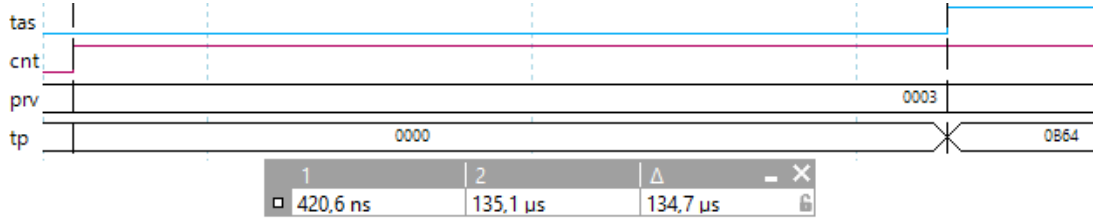
- Messung mit PicoScope 2205 MSO
- Simulationsvergleich mit GTKWave und Vivado 2016.2

$$t = Cnt \cdot \frac{1}{f_{CPU}} = Cnt \cdot \frac{1}{50\ MHz}$$

Messungen und Vergleiche

Messungen an Testsystemen: Allgemeine Zähler



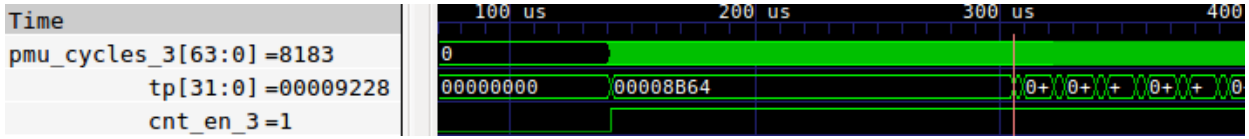


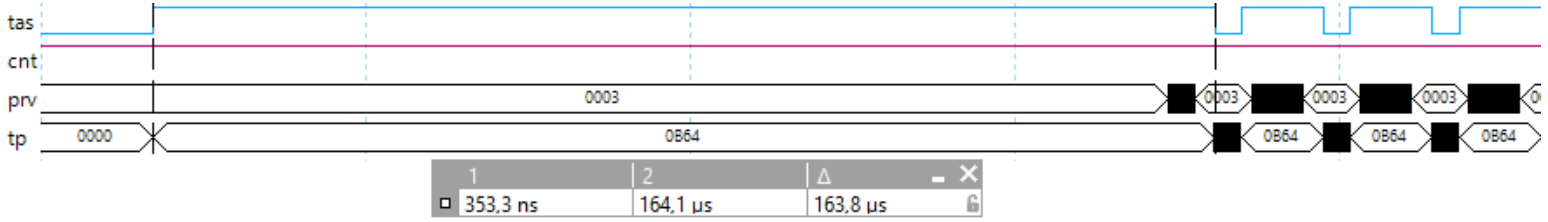
PMU_ALL_COUNTER

6736 ≙ 134,72 µs 134,7 µs

PMU_TASKTIME_COUNTER

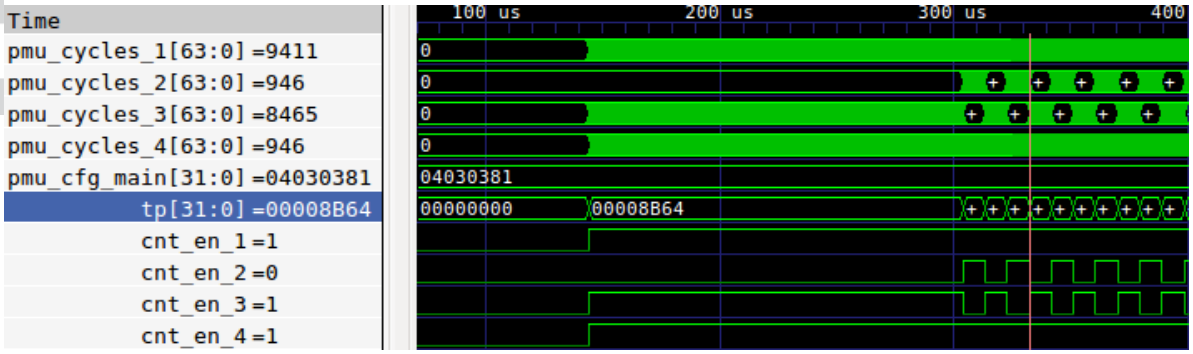
8183 ≙ 163,66 µs 163,8 µs





Messungen und Vergleiche

Messungen an Testsystemen: Taskzähler



9411 ≙ 188,22 µs 188,6 µs

946 ≙ 18,92 µs 19,099 µs

8465 ≙ 169,3 µs

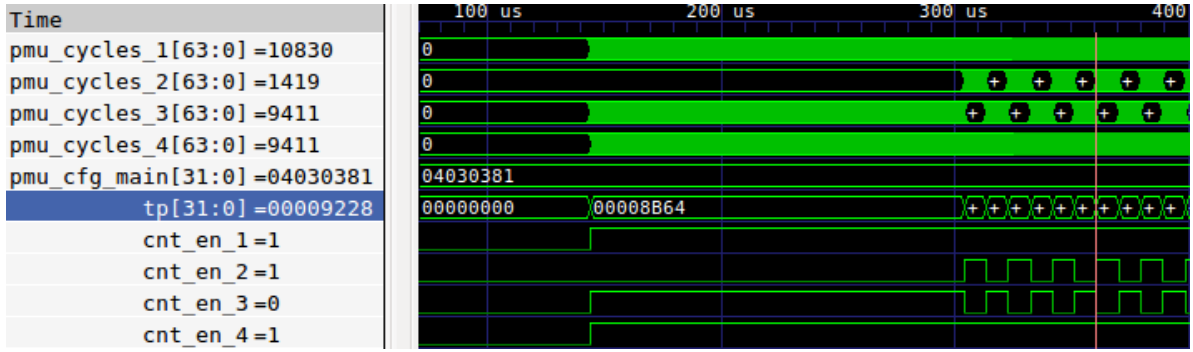
8465 + 946 = 9411

10830 ≙ 216,6 µs 216,8 µs

1419 ≙ 28,38 µs

9411 ≙ 188,22 µs 188,614 µs

9411 + 1419 = 10830



Messungen und Vergleiche

Vergleich mit ähnlichen Systemen und Ausblick

- Maßgebliche Unterschiede zu bestehenden Systemen:
  - PMU fügt sich in das Gesamtsystem ein und versteht dieses
  - Wiederverwendung der Zählelemente → Hardwareaufwand bleibt minimal
- Durch Konfigurierbarkeit einfach auf andere Systeme übertragbar

- Zukünftig realisierbare Funktionalitäten:
  - Sequentielle Zähler
  - Selbstoptimierung des Speicherauslagerungsprozesses hinsichtlich benötigter Laufzeit

[6],[7]

Danke für die Aufmerksamkeit

- [1] https://riscv.org/
- [2] Andrew Waterman, Yunsup Lee, David A. Patterson, and Krste Asanovic. The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.1. Technical Report UCB/EECS-2016-118, EECS Department, University of California, Berkeley, May 2016.
- [3] Yunsup Lee, Albert Ou, and Albert Magyar. Z-scale: Tiny 32-bit RISC-V Systems. https://riscv.org/wp-content/uploads/2015/06/riscv-zscale-workshop-june2015.pdf, 2015.
- [4] Andrew Waterman, Yunsup Lee, Rimas Avizienis, David A. Patterson, and Krste Asanovic. The RISC-V Instruction Set Manual Volume II: Privileged Architecture Version 1.9.1. Technical Report UCB/EECS-2016-161, EECS Department, University of California, Berkeley, Nov 2016.
- [5] Xilinx. 7 Series FPGAs Memory Resources. Xilinx, ug473 (v1.12) edition, September 2016.
- [6] Jude Angelo Ambrose, Vito Cassisi, Daniel Murphy, Tuo Li, Darshana Jayasinghe, and Sri Parameswaran. Scalable Performance Monitoring of Application Specific Multiprocessor Systems-on-Chip. 2013 IEEE 8th International Conference on Industrial and Information Systems, August 2013.
- [7] Nam Ho, Paul Kaufmann, and Marco Platzner. A hardware/software infrastructure for performance monitoring on LEON3 multicore platforms. 24th International Conference on Field Programmable Logic and Applications, 2014.