

Detaillierter Sachbericht des Projektpartners RPTU (TUK) Kaiserslautern-Landau der Förderung zur Förderbekanntmachung „Elektronik und Softwareentwicklungsmethoden für die Digitalisierung der Automobilität (MANNHEIM)“ bei den BMFTR-Referaten 511 Künstliche Intelligenz und 512 Elektronik und autonomes Fahren; Supercomputing

Titel: “Flexibles KI-Deployment und KI-Plattformen für eingebettete, automotive Anwendungen” (Akronym: „MANNHEIM-FlexKI”)

Laufzeit: 01.09.2022 bis 31.08.2025 (3 Jahre)



Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Forschung, Technologie und Raumfahrt unter dem Förderkennzeichen 16IS22086K gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der Autorin/beim Autor.

Projektkoordinator: Ingo Feldner HW/SW Co-Design; Embedded AI (CR/ASD4), Tel. +49 711 811-33648, Mobil +49 1525 8813118, Ingo.Feldner@de.bosch.com

Tabelle 1: Adressen und Ansprechpartner der Verbundpartner

Verbundpartner mit Adresse	Art ¹	Ansprechpartner / Projektleiter
Robert Bosch GmbH (RB) Robert-Bosch-Campus 1, 71272 Renningen	GU	Ingo Feldner Tel. +49 711 811-33648, Ingo.Feldner@de.bosch.com
Eberhard Karls Universität Tübingen (EKUT) Sand 13, 72076 Tübingen	F&E	Prof. Dr. Oliver Bringmann Tel. +49 7071 29-77348, oliver.bringmann@uni-tuebingen.de
Infineon Technologies AG (IFX) Am Campeon 1-15, 85579 Neubiberg	GU	Prof. Dr.-Ing. Wolfgang Ecker Tel. +49 89 234 45334, wolfgang.ecker@infineon.com
itemis AG (ITE) Industriestr. 6, 70565 Stuttgart	KMU	Andreas Graf Tel. +49 151 10860479, graf@itemis.de
Mercedes Benz AG (MB) Hanns-Klemm-Straße 45, 71034 Böblingen	GU	Andreas Herp Tel. +49 151 58613563, andreas.herp@mercedes-benz.com
MINRES Technologies GmbH (MNRS) Keltenhof 2, 85579 Neubiberg	KMU	Eyck Jentzsch Tel. +49 89 67807688, eyck@minres.com
Neumonda GmbH (NM) Industriestr. 4-6, 61440 Oberursel	MU	Prof. Dr. Peter Pöchmüller Tel. +49 6172 90350, peter.poechmueller@neumonda.com
Ruhr-Universität Bochum (RUB) Universitätsstr. 150, 44780 Bochum	F&E	Prof. Dr.-Ing. Tim Güneysu Tel. +49 234 32-24626, tim.gueneysu@rub.de
Technische Universität Darmstadt (TUDA) Hochschulstr. 10, 64289 Darmstadt	F&E	Prof. Dr.-Ing. Andreas Koch Tel. +49 6151 16 22420, koch@esa.informatik.tu-darmstadt.de
Technische Universität Kaiserslautern (TUK) Erwin-Schrödinger-Str., 67663 Kaiserslautern	F&E	Prof. Dr.-Ing. Norbert Wehn Tel. +49 631 205-4436, wehn@eit.uni-kl.de
Technische Universität München (TUM) Arcisstr. 21, 80333 München	F&E	M. Sc. Johannes Geier Tel.: +49 89 289 -23643, johannes.geier@tum.de
Creonic GmbH (CRE) Bahnhofstr. 26-28, 67655 Kaiserslautern	KMU	Dr.-Ing. Timo Lehnigk-Emden Tel. +49 631 34 35 988-1, timo.lehnigk-emden@creonic.com
edacentrum GmbH (EDAC) Schneiderberg 32, 30167 Hannover	Unterauftragnehmer aller Partner	Boris Strohmeier Tel. +49 511 93 68 74-67, strohmeier@edacentrum.de

1 Ziele

1.1 Gesamtziel des Vorhabens / Motivation

Deutschland ist einer der führenden Standorte für Hersteller eingebetteter Elektroniksysteme, speziell für den Automobilbereich. Dabei ergeben sich bei der Entwicklung von Anwendungen für das autonome Fahren und die damit einhergehenden technologischen Disruptionen (Stichwort Künstliche Intelligenz – KI) neue Herausforderungen für den Software- und Hardware-Entwurf (SW- und HW-Entwurf). Autonome Fahrzeuge werden

¹ GU = Großunternehmen, MU = Mittelständisches Unternehmen, KMU = Klein und Mittelständische Unternehmen, F&E = Forschungspartner

mit einer Vielzahl von Sensoren (Video, Lidar, Radar, Sound, Ultraschall) ausgestattet sein, um ein möglichst genaues Abbild der Umgebung zu erzeugen, diese zu interpretieren, das Verhalten aller beteiligten dynamischen Objekte (u.a. Kraftfahrzeuge, Fahrräder, Fußgänger) vorherzusagen, die Planung der Manöver und Fahrtrajektorien vorzunehmen sowie die Fahrdynamik zu regeln. Dies führt zu explodierenden Anforderungen an die Leistungsfähigkeit, Vertrauenswürdigkeit und Energieeffizienz von automobilen IT-Systemen. Datenströme müssen möglichst früh und dezentral in der Verarbeitungskette in Echtzeit erfasst, fusioniert und analysiert werden. Hauptpfeiler hierfür sind eingebettete KI-Plattformen, die Anwendungen auf Basis mehrerer tiefer neuronaler Netze (DNNs) hocheffizient ausführen. Bestehende KI-Plattformen zur Realisierung von hochkomplexen, KI-basierten Fahrfunktionen besitzen eine elektrische Leistungsaufnahme von mehreren Kilowatt, so dass die Energieeffizienz bei einem Maximum an Verlässlichkeit und Vertrauenswürdigkeit beim Entwurf solcher Systeme für „mission critical“ Anwendungsszenarien eine entscheidende Rolle spielt. Dieser Markt wird vielfältig besetzt, von klassischen GPU-Herstellern, wie Nvidia mit der Xavier-, der Orin- und künftig der Atlan-Architektur über spezialisierte HW-Anbieter wie z.B. Intel Mobileye mit Q5/Q6 sowie Smartphone System-on-Chip (SoC)-Anbieter wie Qualcomm mit Snapdragon Ride und Huawei HiSilicon Ascend bis hin zu Automotive OEMs, wie Tesla mit dem Full-Self Driving Computer-Chip. Ähnliches spiegelt sich auch in den Roadmaps führender deutscher Automotive OEMs wider. Bei der Entwicklung von Anwendungen auf Basis Künstlicher Intelligenz (KI) für autonome Fahrfunktionen werden neuronale Netze zuerst von KI-Experten entworfen und trainiert, im Folgenden für die eingebettete Ziel-Hardware (HW)-Plattform optimiert und schließlich die Software (SW)-Codes generiert oder vielfach noch von Hand implementiert. Die Verwendung unterschiedlicher HW-Architekturen in diesem Kontext erfordert stets eine umfangreiche Anpassung der KI-Anwendungen und bindet oftmals stark an Anbieter-spezifische Entwicklungsabläufe und Entwicklungstools.

Diese Abhängigkeit erzeugt große Herausforderungen bei der Entwicklung von KI-Anwendungen für international agierende, deutsche Automobilunternehmen. Das Projekt MANNHEIM-FlexKI (Flexibles KI-Deployment und KI-Plattformen für eingebettete, automotiv Anwendungen) hat zum Ziel, diese HW-Abhängigkeit zu durchbrechen und einen offenen Referenz-Ansatz für das Deployment von KI- und DSP-Anwendungen zu erforschen, der es erlaubt, KI-Anwendungen schnell auf eine neue HW-Plattform zu portieren (sog. Retargeting). Durch dieses FlexKI Retargeting-Verfahren ergeben sich große wirtschaftliche Vorteile für deutsche Unternehmen, um mit zukünftigen Herausforderungen bei der Entwicklung von KI-Systemen umzugehen:

- Durch **Protektionismus** und **Blacklisting** können zukünftig spezifische KI-Plattformen in einem bestimmten Zielmarkt nicht einsetzbar sein. Die Austauschbarkeit der HW durch den HW-agnostischen Flex-KI-Deployment Ansatz ist hier ein Enabler, um auf allen Märkten präsent zu sein.
- Die **Chip-Krise** hat gezeigt, dass zeitweilig bestimmte Zielplattformen nicht verfügbar sein können. Zukünftige Chip-Krisen können ein Retargeting erzwingen, wenn auf eine alternative KI-Plattform umgestiegen werden muss. Hier mit dem FlexKI-Ansatz schnell agieren zu können, erzeugt einen Wettbewerbsvorteil.
- Durch sogenannte **Lock-In-Effekte** können sich Abhängigkeiten von einem speziellen Hersteller von KI-Plattformen ergeben, da Legacy Code dieser Plattformen benötigt wird. Eine Vermeidung solcher Abhängigkeiten erzeugt einen weiteren Wettbewerbsvorteil für deutsche Unternehmen.

FlexKI verfolgt hierbei zwei Entwicklungspfade: 1) Das flexible Deployment von vernetzten KI-Applikationen auf hoch-performante, heterogene Commercial-off-the-Shelf-HW-Plattformen. 2) Einen HW/SW-Co-Design-Ansatz für das Deployment auf eine neue maßgeschneiderte, energieeffiziente KI-HW-Plattform. Beide Pfade ermöglichen neben der Vermeidung eines Vendor Lock-In eine schnelle und automatische Migration von KI-Anwendungen auf andere HW-Plattformen falls aufgrund von Protektionismus, Blacklisting oder Lieferengpässe (Chip-Krise) bestimmte HW-Komponenten nicht verfügbar sind. Insgesamt stärkt das Projekt alle Ebenen der automobilen Wertschöpfungskette in Deutschland und etabliert einen Weg für zukünftige Standardisierung im Bereich KI-Deployment.

Robert Bosch (RB) und Infineon (IFX) planen durch den Transfer der Projektergebnisse an verschiedene Geschäftsbereiche und die Einbindung in den industriellen Entwurfsablauf hocheffiziente und vertrauenswürdige KI-Applikationen auf Basis von DNNs und DSP zu geringen Kosten und damit wirtschaftlich wettbewerbsfähig zu entwickeln. Durch ein erwartetes Wachstum von ca. 8% im gesamten Halbleitermarkt sehen RB und IFX in dem Projekt MANNHEIM-FlexKI eine hervorragende Möglichkeit, ihre zukünftige Position im Markt zu stärken.

Die Mercedes Benz AG (MB) plant durch den Transfer der Projektergebnisse KI-Lösungen in den verschiedensten Steuergeräten im Fahrzeug einsetzen und zwischen diesen auch verschieben zu können, um zum Beispiel eine bessere Verteilung der Rechenlasten zu erlauben und auf zum Beispiel Lieferengpässe einzelner Komponenten schnell reagieren zu können.

Die itemis AG (ITE) plant auf Basis der Projektergebnisse eine Erweiterung des Produktportfolios (z.B. Verwertung einer Referenzimplementierung, Entwicklung eines „As-A-Service“ Angebots). Damit bildet das Projekt eine wesentliche Säule für das Unternehmenswachstum im Bereich ML/AI.

Neumonda GmbH (NM) möchte das generierte DDR4 PHY IP sowie die Ergebnisse von CRE für die Entwicklung einer DRAM Testplattform verwenden. Damit könnte das Produktportfolio erweitert und die Unabhängigkeit von asiatischen Lieferanten reduziert werden. Insbesondere soll eine innovative Technologie entwickelt werden, welche einen Robustheitstest in der Anwendung erlaubt, um eine höhere Zuverlässigkeit im Betrieb zu gewährleisten.

1.2 Wissenschaftliche und/oder technische Arbeitsziele des Projektpartners RPTU

Die RPTU Kaiserslautern-Landau (RPTU), Lehrstuhl Entwurf mikroelektronischer Systeme, arbeitet an dem Design und der Implementierung eines anwendungsoptimierten Speichercontrollers für DDR4-DRAM-Speicher. Dieser wird hauptsächlich zur Integration in ASICs aber auch für FPGAs vorbereitet und validiert. Zusätzlich wird die RPTU neben der Validierung des DDR4 PHYs an der Modellierung und Integration von Near-/In-Memory Processing für DRAMs (DRAM-PIM) in Bezug auf Deep Learning/Machine Learning (DL/ML) Algorithmen und an der Evaluation des Einsatzes von Near-/In-Memory Computing Architekturen für verschiedene Edge-Anwendungen arbeiten.

Ziele: Aufbauend auf den Vorerfahrungen bei der RPTU (TUK) mit anwendungsspezifischen DRAM-Controllern für Spezialprozessoren oder Video-Anwendungen soll in MANNHEIM-FlexKI ein breiteres Feld von ML/KI-Anwendungen unterstützt werden. Weiter soll durch den Einsatz von Near-/In-Memory Processing für das Edge-Computing (DL/ML) zusätzlich gezeigt werden, dass der Energiebedarf für Edge-Anwendungen stark reduziert werden kann. Mit Hilfe des entwickelten Virtuellen Prototyps können dann die verschiedenen DRAM-PIM Ansätze untersucht und evaluiert werden (Performanz, Energieeffizienz).

2 Stand der Wissenschaft und Technik; relevante Kompetenzen und bisherige Arbeiten

2.1 Stand der Wissenschaft und Technik

2.1.1 *Werkzeuge für das automatisierte Deployment von KI/ML-Anwendungen und enge Verzahnung für plattform-spezifische Optimierung*

Einige Anbieter von KI-HW-Plattformen bieten dazugehörige vendor-spezifische Frameworks für die optimierte Implementation von ML Anwendungen auf ihren Systemen an (z.b. Nvidia – TensorRT, Intel – openVino, Xilinx – Vitis AI, Qualcomm® Neural Processing SDK, ARM VELA), sowohl auf Plattformen für Mobiltelefone als auch auf Mikrocontroller-Plattformen. Dieser Ansatz hebt jedoch die Problematik der technischen Inkompatibilität bzw. Einarbeitungszeit zwischen verschiedene Frameworks. Ein Wechsel zwischen HW-Plattform ist mit dem Wechsel auf neue Frameworks verbunden und eine Portabilität ist somit nicht gegeben. Heterogene, Anwendungs-optimierte Lösungen sind praktisch unmöglich und der Vergleich von alternativen Lösungen auf Basis unterschiedlicher KI-Plattformen aufwendig. Dadurch ergibt sich der bereits in der Motivation bekannte Lock-in Effekt, der kein schnelles Retargeting erlaubt.

Einige dieser Vendor-Frameworks basieren auf quell-offenen/frei verfügbaren ML Frameworks. Zum Beispiel basiert ARM's VELA Deployment Lösung auf dem quelloffenen Google Deployment Projekt „tensorflow-lite“. Diese Vendor-spezifischen Erweiterungen der quelloffenen Frameworks bieten aber keine effektive Basis für die Realisierung eines konfigurierbare plattformneutralen Deployment-framework für heterogene Systeme, da sie darauf ausgelegt sind, nur die vendor-spezifische Hardware zu unterstützen. Durch ihre Entwicklung auf Basis von tensorflow-lite als proprietäre, time-to-market getriebene, Lösungen für etablierte Prozessor/Prozessor IP-Produkt-familien sind sie aus Compiler-technischer Sicht fernab des notwendigen Stands der Technik.

Eine weitere interessante Technologie zur **automatisierten Abbildung neuronaler Netze auf HW-Plattformen und damit einhergehender Optimierung** ist Apache TVM. Seit 2020 unterstützt der ML-Compiler Apache TVM die Generierung von Code für verschiedene Prozessorarchitekturen, wobei die Erweiterung μ TVM auf kleine Edge Geräte abzielt. TVM wird kontinuierlich um Funktionen erweitert, zum Beispiel zur Generierung von statischem Modell-Code (AOT: *Ahead-of-time Compilation*).

Die Erweiterung um die Unterstützung neuer HW-Plattformen ist jedoch nicht automatisiert und mit hohen Aufwänden verbunden. Hierbei wurde im Projekt Scale4Edge ein erster Vorschlag erarbeitet, wie eine generische Schnittstelle für KI-Beschleuniger für TVM aussehen kann. Dieser sogenannte RFC mit Namen *Universal*

Modular Accelerator Interface (UMA) steht kurz vor der Akzeptanz durch die TVM Community. Dieser Beitrag zeigt, dass flexible KI-Lösungen, wie sie von der deutschen Automobilindustrie benötigt werden, über koordinierte Open Source-Beiträge aus Förderprojekten als Quasi-Standard etabliert werden können. Darauf aufbauend existiert aber bisher kein standardmäßiges Backend für die verteilte Berechnung von neuronalen Netzen, entweder auf den verschiedenen Recheneinheiten (Prozessor, Beschleuniger) eines Knotens (*Mixed deployment*) oder auf mehreren Knoten (*Distributed Inference*), wie von MANNHEIM-FlexKI angestrebt wird. Hierzu gibt es nur akademische Ansätze wie zu Beispiel DeeperThings, ADCNN oder MoDNN: Da in der Entwicklungs-Community die Automotive-Industrie noch nicht stark vertreten ist, finden diese spezifischen Anforderungen noch keinen Eingang. Diese Features sollen nun durch MANNHEIM-FlexKI adressiert werden.

2.1.2 Arbeiten zur Trennung von Neuronalen Netzen und HW-Plattformen mit automatisiertem Deployment

Ein Hauptziel von FlexKI ist die HW-agnostische Entwicklung von KI-Applikationen. Eine Übertragung der Ideen von AUTOSAR, modellbasierter SW-Entwicklung und generischen Programmiersprachen auf die generische Unterstützung von KI-HW-Plattformen liegt nahe: Auf standardisierten Plattformen wird via definierter Schnittstellen aus einer standardisierten Beschreibung optimierte spezifische SW generiert.

Dazu liegen bereits erste **standardisierte Beschreibungsformate für neuronale Netze** vor, wie z.B. ONNX, TIR und MLIR. Da die Entwicklung der Methoden und Technologien im Bereich ML schnell voranschreitet, müssen diese Standards kontinuierlich weiterentwickelt werden.

Weiterhin sind **standardisierte Beschreibungsformate für HW-Plattformen** erforderlich. Diese Beschreibungen liegen oft in Textform vor und sind nicht für die automatisierte Verarbeitung geeignet. Dort wo Beschreibungsformate existieren (z.B. IP-XACT, Scale4Edge CoreDSL) fehlen wichtige Informationen, die für ein automatisiertes Deployment erforderlich dies umfasst zum einen die Beschreibung nichtfunktionaler Eigenschaften von Hardwarekomponenten wie z.B. Performance, Energiebedarf, Timing zum anderen fehlen häufig Beschreibungen der Operationssemantik auf einem für ML-Compiler geeigneten Abstraktionsniveau.

Eine weitere Klasse von HW-Beschreibungsformaten sind sogenannte Architecture Description Languages (ADLS) wie Spatial von Stanford oder HeteroCL von Cornell. Diese zielen aber auf eine Synthese von Beschleunigern ab, nicht auf die Generierung der SW-Unterstützung für das Deployment.

2.1.3 Skalierbare Edge-KI-Plattformen

In vielen modernen Anwendungsbereichen ist eine Anpassung der Rechnerarchitektur an die Anforderungen der konkreten Anwendung bzw. Anwendungsdomäne erforderlich, um die im Automotive/Edge-Bereich nötigen Effizienzen (z. B. in Bezug auf Rechenleistung/Latenzen in Relation zu Energieaufnahme oder Kosten) zu erzielen. Bei dieser Vorgehensweise muss aber stets abgewogen werden, ob der Einsatz von zu spezialisierten Bausteinen für Teilfunktionen eventuell das System im Ganzen zu sehr verkompliziert. Eine balancierte Lösung erfordert sowohl die flexible *Komposition* von Komponenten als auch ihr passgenaues *Zuschneiden* auf die konkrete Anwendung durch möglichst freie Parametrisierbarkeit. Ein Beispiel für einen Punkt in diesem Lösungsraum für Edge-Devices sind auf ML-Funktionen hochspezialisierte Recheneinheiten, wie der ARM Ethos-N37 IP-Block oder der Hailo-8 Chip, die für energieeffiziente Inferenzaufgaben ausgelegt sind. Allerdings sind sie nur begrenzt parametrisiert, z.B. ist der N37-Block stets auf 512 8-Bit MAC-Operationen je Takt festgelegt. Im akademischen Bereich wurden in den letzten Jahren ebenfalls eine Vielzahl von KI-Rechenbeschleunigern mit Fokus auf Inferenz von künstlichen neuronalen Netzwerken vorgeschlagen. Einige basieren auf systolischen Arrays oder erweitern diese durch eine rekonfigurierbare Architektur. S2TA ist eine Methode, die strukturierte Sparsamkeit auf systolischen Arrays ausnutzt. Wang et al. zeigten eine FPGA-basierte Lösung, ähnlich wie Sankaradas et al. für CNNs. Innerhalb der Gruppe der ASIC-Beschleuniger kann zwischen Processing-Element-basierten (PE-basierten), eng gekoppelten und Coprozessor-basierten unterschieden werden. PE-basierte Beschleuniger sind für hochparallele Berechnungen ausgelegt. Einige wurden speziell für die Berechnung von Daten in komprimiertem Format entwickelt um eine effiziente Verarbeitung und Leistungseffizienz zu erreichen. Es ist zu beachten, dass weder die ARM Ethos-Architektur und die Hailo-Lösung noch die genannten akademischen Ansätze anspruchsvollere konventionelle SW ausführen können, die ja nach wie vor für viele Aufgaben benötigt wird. Mit der anpassbaren FlexKI HW-Plattform sollen *beide* dieser Schwächen adressiert werden: So sollen durch Parametrisierung zum einen hochgradig flexible Lösungen entstehen, zum anderen durch die werkzeugunterstützte Komposition von heterogenen Recheneinheiten im richtigen Mix passgenaue *anwendungsspezifische HW-Systeme* geschaffen werden. Das Endergebnis soll damit eher vergleichbar dem GreenWaves GAP8-SoC sein, das auf seinen neun programmierbaren Kernen sowohl ML-Anwendungen als auch (einfache) Steuer-SW ausführen kann. Diese allgemeine Verwendbarkeit wird auf dem GAP8 aber erkauft durch eine potenziell reduzierte Energieeffizienz, da für die Kommunikation

zwischen den Kernen immer *Speicher* genutzt wird. Dieser kann wegen der Notwendigkeit von Adressberechnungen gegenüber spezialisierteren Kommunikationslösungen, wie z.B. Punkt-zu-Punkt FIFOs oder Atomic Updates, im Betrieb mehr Energie benötigen. In der FlexKI HW-Plattform werden daher nicht nur die Recheneinheiten, sondern auch die Kommunikationsmechanismen dazwischen anwendungsspezifisch optimiert. Mit diesem Ansatz lassen sich nicht nur Architekturen wie der für Anwendungen des autonomen Fahrens gedachte NSITEXE DR1000C Chip realisieren, der vier skalare Rechenkerne mit einer Vektoreinheit für umfangreichere Berechnungen kombiniert, sondern darüber hinaus auch noch als skalierbare Kacheln ausgelegte ML-Beschleuniger für effiziente ML/KI-Inferenz einbinden. Damit strebt MANNHEIM-FlexKI in Bezug auf die neue anpassbare HW-Plattform eine deutlich verbesserte Lösungsqualität in den für die deutsche und europäische Industrie besonders relevanten Bereichen der Prozessoren für Spezialanwendungen, wie Automotive, Medizin und Industrie 4.0 an.

2.1.4 HW/SW-Co-Design

Zudem sind neue Co-Design-Ansätze erforderlich, die eine gemeinsame Optimierung der ML-Systeme passend zur zugrundeliegenden HW-Plattform (Prozessorkerne, KI-Rechenbeschleunigern und spezialisierte Speicherarchitektur) ermöglichen. Daraus hat sich in den letzten zwei Jahren das neue und sich sehr dynamisch entwickelnde Forschungsfeld "AI System Hardware/Software Co-Design" bzw. "Hardware-ML Model Co-Design" entwickelt. Dabei gilt es, DNN-Optimierungen wie "Pruning", "Quantization", "Knowledge Distillation", "Conditional Computing" und "Neural Architecture Search" in Zusammenhang mit der Optimierung und Konfiguration der KI-Rechenbeschleuniger (u.a. Wortbreitenoptimierung, Topologiebestimmung und Dimensionierung der MAC-Arrays sowie der Speicherarchitektur) für intelligente Sensorsysteme nutzbar zu machen. Ein neuer industrieller Ansatz wurde von Google bereitgestellt (CFU Playgrounds: <https://github.com/google/CFU-Playground>), wobei hier aber nur sehr kleine, maßgeschneiderte RISC-V-basierte KI-Systeme (TinyML) erzeugt werden können. MANNHEIM-FlexKI zielt hier auf eine skalierbare Plattform ab, die auch größere ML Workloads unterstützt. Der optimierte Einsatz dieser Methoden erfordert die Bereitstellung der Systemarchitektur in einem geeigneten Format. Mit der FlexKI Modellierung wird hier eine Grundlage geschaffen, die es erlaubt HW-Eigenschaften beim Design der Neuronalen Netzwerke zu berücksichtigen und die HW/SW-Architekturen optimiert aufeinander abzustimmen.

2.1.5 Safety/Security

In jüngster Zeit wurde mehrfach demonstriert, wie mithilfe physikalischer Implementierungsangriffe (Seitenkanal-/Fehlerangriffe) die Vertraulichkeit und Integrität von KI-HW Beschleunigern korrumpiert werden kann. Hierbei analysiert der Angreifer den Stromverbrauch, die elektromagnetische Abstrahlung oder das Zeitverhalten der KI-HW Primitive um Rückschlüsse auf die verwendete Netzarchitektur (Anzahl und Art der Layer, Aktivierungsfunktion, usw.) und/oder die Trainingsgewichte zu ziehen. Solch ein Reverse Engineering von KI-Netzen würde dem Angreifer, der auch ein Mitbewerber sein kann, erlauben kommerzielles IP zu replizieren, was zu einem erheblichen wirtschaftlichen Schaden führen kann. Desweiteren wurde in der Literatur auch schon gezeigt, dass genaue Informationen über das verwendete KI-Netz sogenannte „adversarial attacks“ erleichtern. Dabei werden Eingabedaten minimal manipuliert, um das KI-Netz zu einer falschen Klassifikation zu verleiten. So ist es z. B. möglich, eine automatische Verkehrszeichenerkennung von Fahrzeugen durch manipulierte Schilder zu täuschen, was im Kontext vom autonomen Fahren zu erheblichen Problemen im Bereich Safety führen kann. Die Härtung der KI-Netze in Bezug auf Security ist daher ein elementarer Bestandteil zur Aufrechterhaltung der Betriebssicherheit (Safety) und muss somit fest im Designprozess verankert sein. Zum Schutz gegen Seitenkanalangriffen können auf HW-Ebene bekannte Verfahren aus dem Bereich kryptografischer Schaltungen wie Maskierung oder Verschleierung (Hiding) angewendet werden. Hierzu gibt es auch schon Arbeiten im Kontext von KI-Netzen. Die vorgestellten Techniken erzeugen jedoch einen sehr großen Chip-Flächenverbrauch, welchen einen praktischen Einsatz erschweren. Auch der Schutz gegen Reverse Engineering Angriffe auf HW-Ebene rückt zunehmend in den Fokus. Erste frei-verfügbare Werkzeuge für (teil-)automatisiertes HW Reverse Engineering befinden sich in der Entwicklung. Im Angesicht der Dynamik, mit der die Entwicklung dieser Werkzeuge voranschreiten bedarf die Entwicklung und Evaluation von Gegenmaßnahmen wie z. B. HW Obfuskation weiterer Erforschung. Im Bereich von SW sind sowohl Werkzeuge zum Reverse Engineering als auch Gegenmaßnahmen weiter vorangeschritten. Ein sehr interessantes Verfahren zur algorithmischen Erkennung von Fehlern in Neuronalen Netzen wurde von NVIDIA vorgestellt, das Convolution-Kernel der neuronalen Netzwerke über Checksummen absichert. Dies erfordert jedoch manuelle Änderungen der Kernels. Eine solche Fehlererkennung wird benötigt, um Safety Aspekte im operativen Betrieb sicherzustellen. In MANNHEIM-FlexKI soll untersucht werden, wie solche Härtungsmethoden automatisiert über den KI-Compiler eingefügt werden können.

Heute gibt es keine Methoden, um die Robustheit von Sekundärkomponenten wie DRAM in der Applikation zu testen. Ein Baustein mag im Produktionstest marginal "pass" sein, fällt in der Applikation aufgrund unterschiedlicher Signalintegrität aus. In MANNHEIM-FlexKI soll ein DRAM PHY entwickelt werden der mit Hilfe von künstlichem Jitter und anderer Maßnahmen die Robustheit des Speichersystems austesten kann. Da die Bausteine im Betrieb altern kann durch diesen Robustheitstest eine wesentliche Degradation frühzeitig im System erkannt werden was die Ausfallsicherheit signifikant erhöht.

2.2 Relevante Kompetenzen und bisherige Arbeiten des Antragstellers

Kurzdarstellung der Rheinland-Pfälzischen Technischen Universität Kaiserslautern-Landau

Der Lehrstuhl für den Entwurf Mikroelektronischer Systeme (EMS) der RPTU Kaiserslautern-Landau (seit 1.1.2023 RPTU, ehemals TU Kaiserslautern) arbeitet auf dem Gebiet des mikroelektronischen Schaltungs- und Systementwurfs. Schwerpunkte sind hierbei insbesondere die Mobilkommunikation, Kanalcodierung, Hardwarebeschleuniger für ML/KI-Anwendungen, neuronale Netze, Optimierung des Leistungs- und Energieverbrauchs, fortgeschrittene Rechner- und Speicherarchitekturen sowie entsprechende Entwurfsmethoden. Der Lehrstuhl ist an mehreren nationalen und internationalen Forschungsprojekten beteiligt. Der Lehrstuhl hat darüber hinaus eine große Expertise im Bereich der Speicherarchitekturen und Speichercontroller. Dies betrifft sowohl die Modellierung als auch die Optimierung von Speicherarchitekturen. Dabei werden sowohl On-Chip Speicher als auch externe Speicher, insbesondere DRAMs, aber auch neue Speicheransätze wie RRAMs modelliert. Bezüglich Optimierungen liegt der Schwerpunkt vor allem auf der Energie- und Speicherbandbreitenoptimierung. Insbesondere im Bereich von KI-Architekturen spielen Speicher eine zentrale Rolle, da große Datenmengen/Gewichte effizient abgespeichert werden müssen

3 Ausführliche Beschreibung des Arbeitsplans

3.1 Aufbau und Gesamtstruktur des Projekts

Der Arbeitsplan von MANNHEIM-FlexKI besteht aus 6 Arbeitspaketen AP1 beinhaltet Arbeitspaketübergreifende Aufgaben wie das Management, den Aufbau der Metadaten-Infrastruktur und der Technologietransferunterstützung. In AP2 geht es um die Modellierung der ML-Workloads und der FlexKI HW-Plattform. Diese Modellierung wird in AP3 für das FlexKI Deployment verwendet und basiert auf der zu entwickelnden FlexKI HW-Plattform aus AP4. AP3 und AP4 untersuchen sowohl Methoden zum IP-Schutz als auch Safety in SW und HW. Durch die Modellierung des ML Workloads und der FlexKI HW-Plattform wird zudem eine HW/SW-Optimierung ermöglicht, die in AP5 untersucht wird. In AP6 werden die Ergebnisse anhand von zwei Use Cases demonstriert und evaluiert. Die MANNHEIM-FlexKI Gesamtstruktur und Zusammenwirken der Arbeitspakete (AP) und Aufgaben (A) ist in Abb. 1 dargestellt.

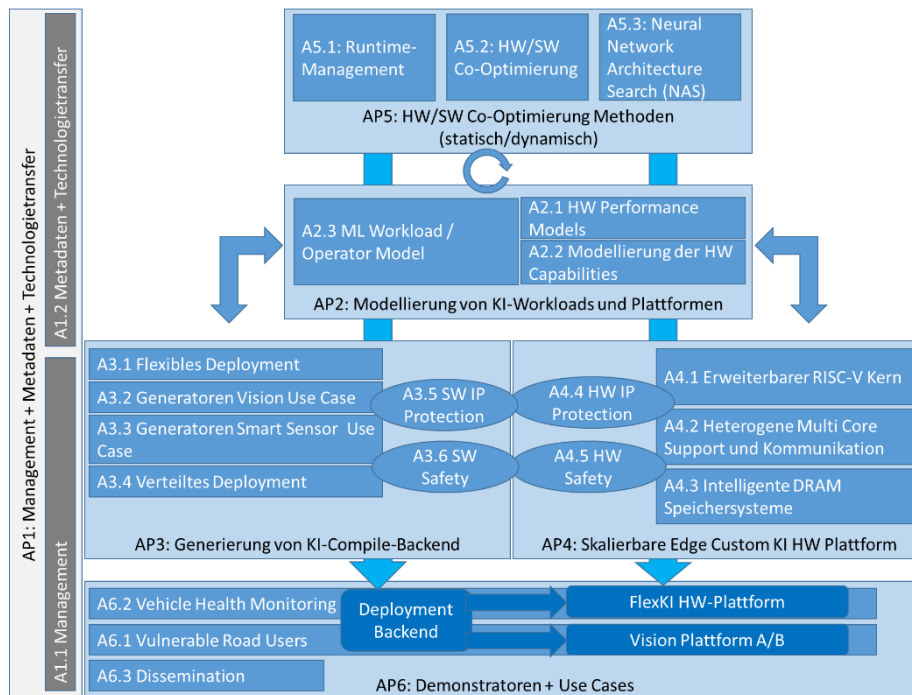


Abb. 1: MANNHEIM-FlexKI Gesamtstruktur und Zusammenwirken der Arbeitspakete (AP) und Aufgaben (A)

3.2 Ausführlicher Arbeitsplan des Projektpartners

Der detaillierte Arbeitsplan von MANNHEIM-FlexKI ist im Folgenden dargestellt. Der Arbeitsplan ist in 6 Arbeitspakete (AP). Der Hauptbeitrag der RPTU (TUK) hat in dem Arbeitspaket AP4.3 stattgefunden. Ein weiterer Beitrag wurde in AP6.3 geleistet.

3.2.4 Arbeitspaket AP4: FlexKI HW-Plattform

Projektpartner: EKUT, IFX, ITE, MB, MNRS, NM, RB, RUB, TUDA, **RPTU (TUK)**

Ziele:

Die FlexKI HW-Plattform ist eine skalierbare, heterogene, RISC-V-basierte HW-Plattform für Edge-Anwendungen. Diese enthält spezialisiert RISC-V-Rechenkerne, optimierte Beschleuniger für KI/ML Aufgaben und anwendungsspezifisch konfigurierbare Speichersysteme. Sie lässt sich flexibel und generatorbasiert an spezifische KI-Anforderungen anpassen. In AP4 wurden flexible konfigurier- und komponierbare HW-Basiskomponenten entwickelt, die durch übergeordnete Werkzeugflüsse maßgeschneidert auf die konkreten Anforderungen einer Anwendung hin zu vollständigen HW-Plattformen automatisiert zusammengesetzt werden können. Angestrebt wird das Abdecken eines kosten- und energieeffizienten Leistungsspektrums von der Vorverarbeitung bzw. Regelung an einzelnen Sensor/Aktorknoten bis hin zur Low-Rate Datenfusion über mehrere Sensoren, *unterhalb* der Fähigkeiten von Central Compute-Knoten. Während der Plattformgenerierung entstand neben den HW-Beschreibungen, z.B. für die Realisierung der HW in Form von eigenen ASIC-Chips, auch detaillierte maschinenlesbare Charakterisierungen der HW-Eigenschaften, die in AP3 und AP5 ausgewertet wurden, um dort ebenfalls automatisiert die für die Plattform passenden Programmierwerkzeuge zu erzeugen und das Gesamtsystem zu optimieren.

Aufgaben:

A4.1: Energieeffizienter anwendungsspezifisch erweiterbarer RISC-V Prozessorkern

A4.2: Unterstützung für heterogene Multi-Cores und Kommunikationsmechanismen

A4.3: Anwendungsoptimierbare intelligente DRAM-basierte Speichersysteme mit Applikationsrobustheitstest

A4.4: IP-Schutz der neuronalen Netze durch HW-seitige Maßnahmen

A4.5: Safety-Aspekte der FlexKI HW-Plattform

Leitung:

TUDA

Meilensteine:

M4.1: Systemarchitektur der FlexKI HW-Plattform

M4.2: Initiale Basisfunktionalität der verschiedenen Blöcke

M4.3: Plattform ebenen übergreifend evaluiert und optimiert, Test und Demonstrator ASICs realisiert und in DDR4-Demonstratorboard integriert

Tabelle 2: Ressourcenplanung zu Arbeitspaket AP4

Partner	A4.1 [PM]	A4.2 [PM]	A4.3 [PM]	A4.4 [PM]	A4.5 [PM]	Summe [PM]
EKUT	0,00	18,00	0,00	0,00	0,00	18,00
IFX	0,00	36,00	0,00	12,00	12,00	60,00
ITE	0,00	0,00	0,00	5,00	0,00	5,00
MB	0,00	0,00	0,00	0,00	4,00	4,00
MNRS	32,00	0,00	0,00	0,00	0,00	32,00
NM	0,00	0,00	53,00	0,00	0,00	53,00
RB	18,00	48,00	0,00	8,00	20,00	94,00
RUB	8,00	0,00	0,00	16,00	3,00	27,00
TUDA	15,00	35,00	34,00	0,00	0,00	84,00
RPTU (TUK)	0,00	0,00	44,00	0,00	0,00	44,00
TUM	0,00	0,00	0,00	0,00	0,00	0,00
CRE	0,00	0,00	0,00	0,00	0,00	0,00
Summe	73,00	137,00	131,00	41,00	39,00	421,00

Ergebnisse:

Die RPTU kooperierte in Arbeitspaket 4.3 eng mit der TU Darmstadt (TUDA) und Neumonda GmbH (NM). NM und TUDA bearbeiteten gemeinsam die Beiträge B4.3.1 (NM) sowie B4.3.2-B4.3.4 (TUDA) für die Realisierung und das Design eines DDR4 DRAM PHYs zur direkten Ansteuerung eines DDR4-DRAM Bausteins.

Im Beitrag B4.3.5 unterstützte die RPTU die Validierung des PHYs von TUDA und NM.

In B4.3.6 wurden PIM-Architekturen von der RPTU modelliert, untersucht und evaluiert.

Final in B4.3.7 entwickelte die RPTU einen entsprechenden DDR4-DRAM Controller Module zur Ansteuerung des DDR4 DRAM PHYs.

A4.3 Anwendungsoptimierbare intelligente DRAM-basierte Speichersysteme

Inhalt von AP4.3: (a) Anwendungsspezifisch optimierbarer DRAM-Speichercontroller, (b) physikalische DDR4 PHY-Schnittstellen auf Basis eines prozessportablen Phase-to-Digital Converters und (c) Unterstützung für Processing-in-Memory sowie besonders robuste Speicheranbindung für kritische Automobilanwendungen.

B4.3.5: (RPTU) Unterstützung bei der Validierung des DDR4-PHY IP-Blocks

Es konnte in diesem Zusammenhang eine detaillierte Evaluierung und Review der Behavioral Verilog-Modelle des DDR4-PHY IP Blocks durchgeführt werden. Durch die abschließende Kopplung von DDR4 Controller (RTL) und PHY-Modell konnte simulativ die Funktionalität validiert werden, siehe auch Abb. 2 und Abb. 3.

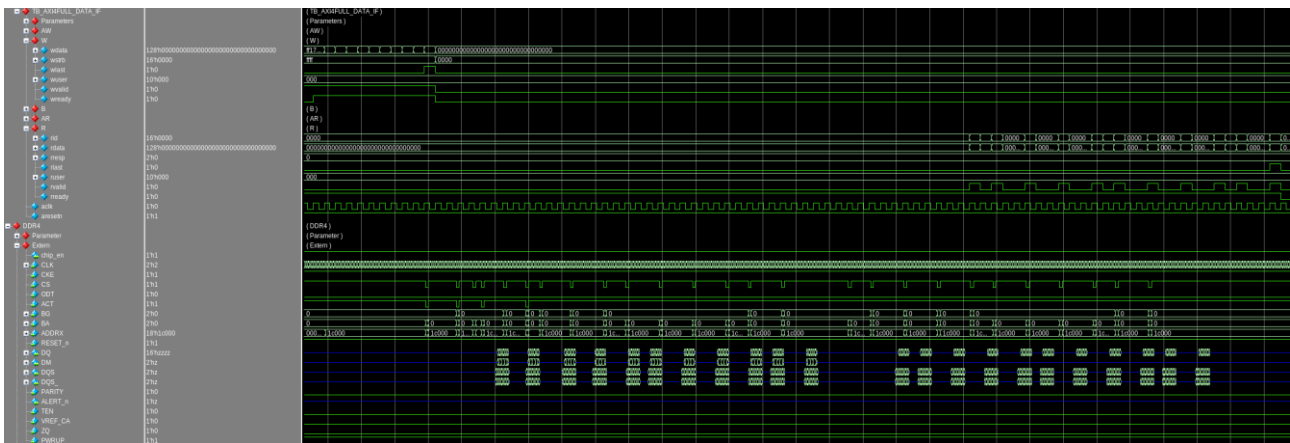


Abb. 2: DDR4 Subsystem Simulation (Controller + PHY): 10x Writes / 10x Reads

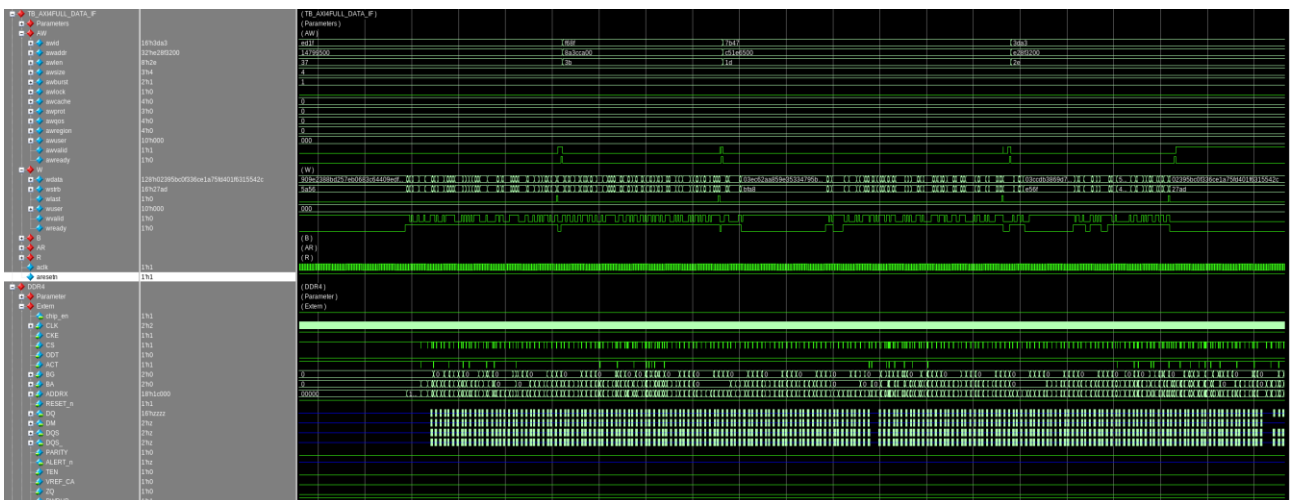


Abb. 3: DDR4 Subsystem Simulation (Controller + PHY): 100x Writes / 100x Reads

B4.3.6: (RPTU) Near und In-Memory-Processing-Architekturen (Modellierung und Evaluierung)

Es zeigt sich jedoch in unserer zunehmend datenorientierten Welt, dass die erreichbare Performanz von DNNs und ML-Implementierungen zunehmend weniger durch die verfügbare Rechenleistung als vielmehr durch die endliche Speicherbandbreite der eingesetzten DRAMs begrenzt wird. Eine mögliche Lösung für dieses Problem ist die Verwendung von Processing-In-Memory (PIM), das einen Teil der Datenverarbeitung direkt in den Speicher verlagert. Hier wurde eine reale PIM-Implementierung namens Function-In-Memory DRAM (FIM-DRAM) des Speicherherstellers Samsung mithilfe eines entwickelten virtuellen Prototyps und unter Verwendung der Simulationsplattform *gem5* und des Speichersimulators *DRAMSys* (siehe auch Abb. 4) analysiert.

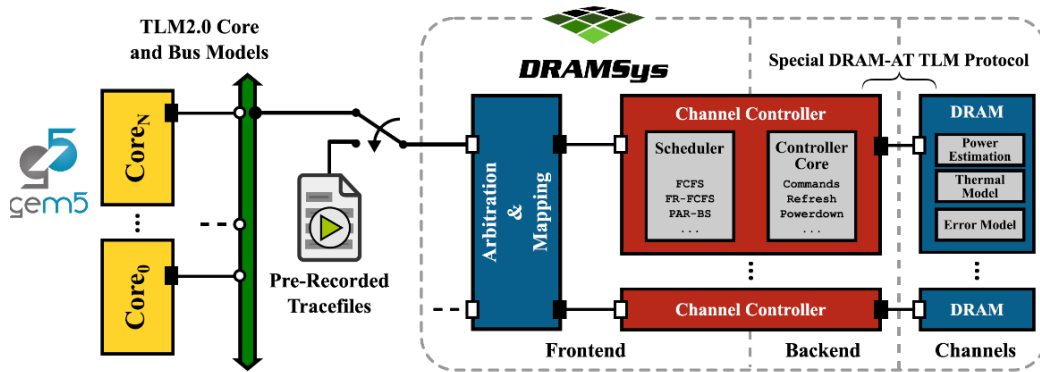


Abb. 4: DRAMSys Simulationsframework

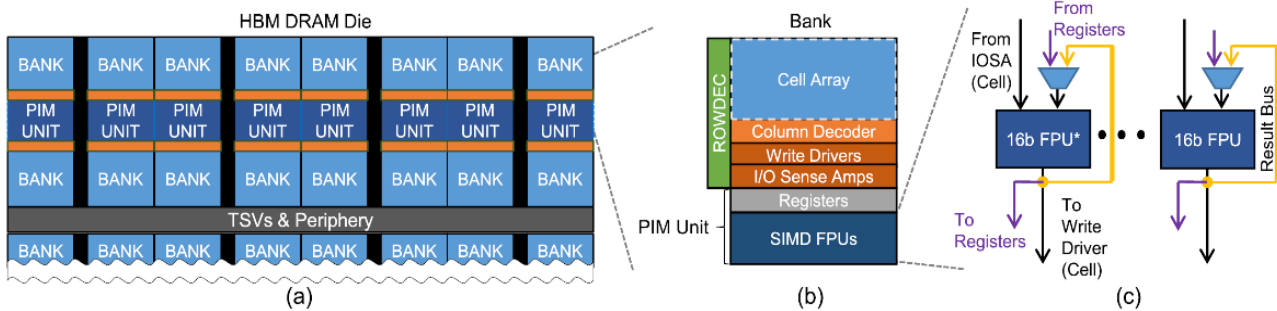


Abb. 5: Samsung's FIMDRAM/PIM-HBM HW Architektur

Wie der Name PIM-HBM bereits andeutet, basiert es auf dem HBM2-Speicherstandard und integriert 16-bit breite SIMD-Engines (Single-Instruction-Multiple-Data) direkt in die Speicherbänke, wobei die große Parallelität auf Bankebene ausgenutzt wird, während die hoch optimierten lokalen Leseverstärker erhalten bleiben und nicht modifiziert werden. PIM-HBM erfordert keine Änderungen an Komponenten moderner Prozessoren, wie z. B. dem Speicher-Controller, d. h., es ist unabhängig von bestehenden HBM2-Plattformen. Folglich ist für PIM-HBM eine Mode-Umschaltung erforderlich, was es für verschachtelten PIM- und Nicht-PIM-Traffic/Transfers und kleine Batchgrößen weniger nützlich macht. Glücklicherweise ermöglicht die Architektur von HBM viele unabhängige Speicherkanäle in einem einzigen Stapel (3D-Stack), so dass eine saubere Trennung des Speichers in einen PIM-aktivierten Bereich und einen normalen HBM-Bereich möglich ist. Das Herzstück des PIM-HBM sind die PIM-Ausführungseinheiten, die von jeweils zwei Bänken eines pCH gemeinsam genutzt werden. Sie umfassen 16 16-bit breite SIMD-Gleitkommaeinheiten (FPUs), Befehlsregisterdateien (CRFs), allgemeine Registerdateien (GRFs) und Skalarregisterdateien (SRFs). Diese allgemeine Architektur ist in Abb. 5 detailliert dargestellt, mit (a) der Platzierung der PIM-Einheiten zwischen den Speicherbänken eines DRAM-Die, wobei (b) eine Bank mit ihrer PIM-Einheit gekoppelt ist, und (c) der Datenpfad der Eingänge, Ausgänge und zeitlichen Ergebnisse innerhalb der PIM-Einheit. Wie in (c) zu sehen ist, können die Eingangsdaten für die FPU entweder direkt aus der Speicherbank, aus einem GRF/SRF oder aus dem Ergebnisbus einer vorangegangenen Berechnung stammen. Die 16 breiten SIMD-Einheiten entsprechen der 256-Bit-Prefetch-Architektur von HBM2, bei der 16 16-Bit-Gleitkomma-Operanden direkt von den sekundären Sense-Amplifiern an die FPUs in einem einzigen Speicherzugriff weitergeleitet werden. Da alle PIM-Einheiten parallel arbeiten, mit 16 Bänken pro pCH, lädt ein einziger Speicherzugriff insgesamt 256 Bit - 8 Verarbeitungseinheiten = 2.048 Bit in die FPUs. Infolgedessen ist die theoretische interne Bandbreite des PIM-HBM 8-mal höher als die externe Busbandbreite zum Host-Prozessor.

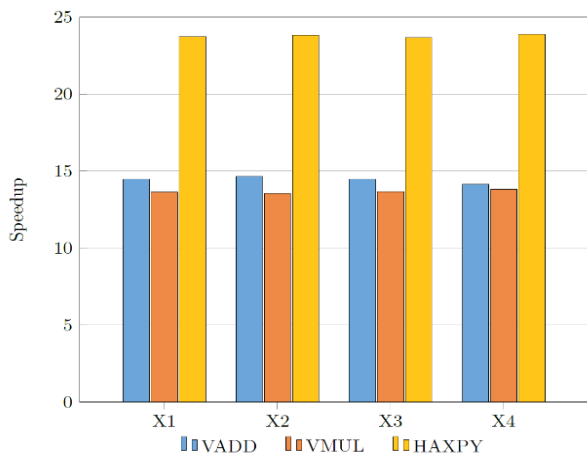


Abb. 6: Vergleich zwischen non-PIM und PIM für die Vektor-Benchmarks, evaluiert mit einer CPU-Frequenz von 3 GHz

Ein erster Satz von Benchmarks analysiert die Beschleunigung von PIM-HBM für verschiedene Vektoroperationen, nämlich eine elementweise Vektoraddieroperation (VADD), eine elementweise Vektormultiplikationsoperation (VMUL) und eine HAXPY-Operation. Die Vektordimensionen wurden von X1 ($2^{21}=2M$) bis X4 ($2^{24}=16M$) gewählt. In Abb. 6 sind die Ergebnisse der Vektorbenchmarks abgebildet. Es konnten 13.6× bis 23.9× Performanceverbesserungen (Laufzeitverbesserung - Speedup) erreicht werden. Diese relativen Performanceverbesserungen konnten für die Vektor-Benchmarks, auf einem generischen ARM-basierenden System (gem5) mit einer typische Taktfrequenz von 3 GHz gezeigt werden.

Zusammenfassend wurde die Studie bzgl. In-Memory Processing, die die Samsung HBM2 DRAM ML-Beschleuniger Architektur nutzt, finalisiert. Die ersten Ergebnisse hierzu sehen durchaus vielversprechend aus.

B4.3.7: (RPTU) Anwendungsspezifisch parametrisierbarer DDR4 Memory Controller (MC)

Zu Beginn wurden erste Spezifikationen bzgl. der HW-Architektur des Memory Controllers mit den Partnern ausgetauscht. Diese Definition der Spezifikationen für den DDR4 Memory Controller wurde im weiteren Verlauf des Projekt verfeinert und abgeschlossen. Somit wurden hier die Schnittstellen, wie in Abb. 7 gezeigt, zum DRAM (DDR4 x16), zum DDR4 PHY (NM; TUDA) und zum „User-Interface“ wurden festgelegt.



Abb. 7: DDR4 Memory Controller im DRAM-Subsystem

Im Einzelnen sind dies:

- Ein DRAM **DDR4 Interface** in „by 16“ Modus (**x16**), Burstlänge (BL) = 8,
- ein **DFI4.0 Interface** zum PHY (128 bit Nutzdaten) und
- ein **AXI4 Burst User-Interface** mit 128 bit Nutzdaten bei einer Burstlänge von 1/2/4.

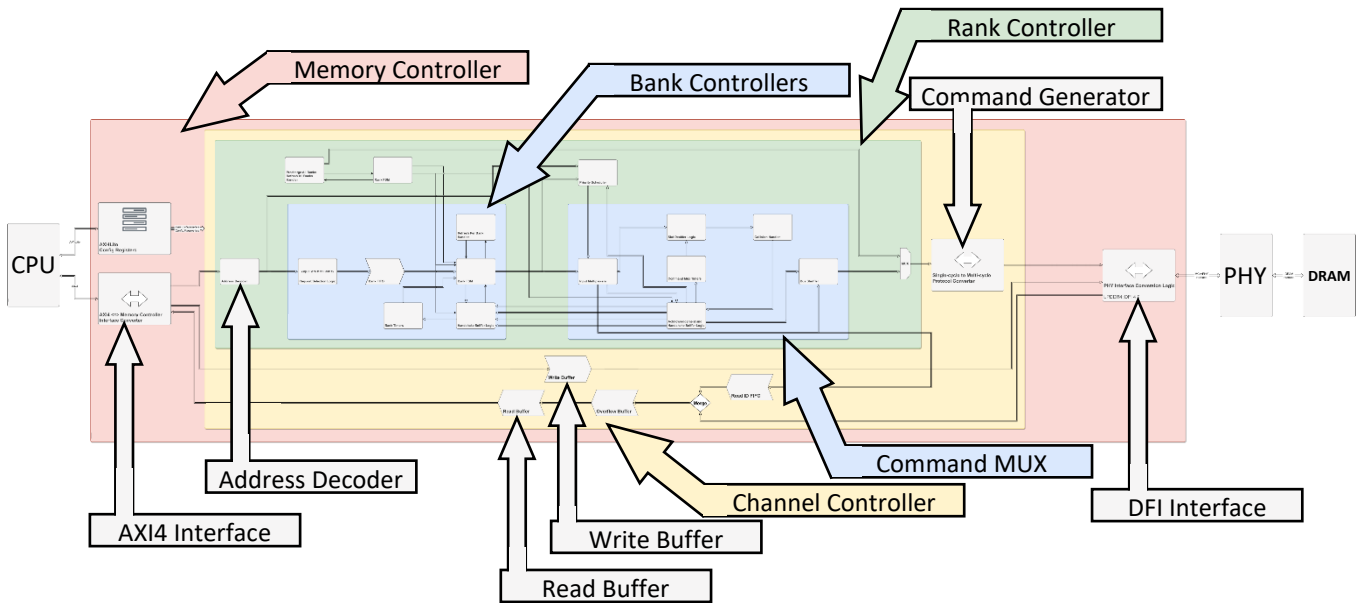


Abb. 8: Details der DDR4 Memory Controller HW Architektur

Der entwickelte DDR4 Controller wurde zuerst auf einer FPGA HW implementiert, um ihn detailliert testen zu können. Mit Hilfe des Config-Busses, siehe auch Abb. 8, werden die entsprechenden Speed-Sort abhängigen Latenzen und Timings eingestellt.

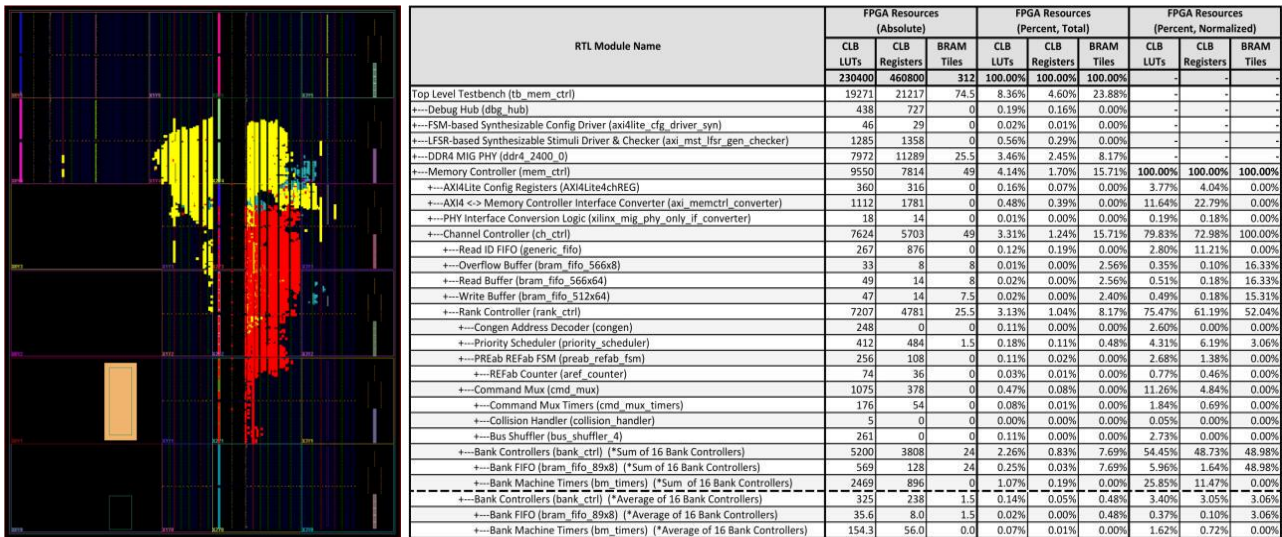


Abb. 9: DDR4 Memory Controller auf dem FPGA und die hierarchische HW-Ressourcenbenutzung

Die Ressourcenverwendung wird hauptsächlich durch Write/Read FIFOs und anderen Write/Read-Speichern (FFs) dominiert.

Die Ergebnisse der Synthese wurden unter folgenden Randbedingungen erzielt:

- Als Constraint wurde DDR4 Controller Frequenz auf $f_{ctrl} = 300$ MHz (FPGA) gesetzt.
- 4 : 1 Clock Ratio (PHY : Controller) → Datenrate von DDR4-2400 (2.4 Gbps/pin).

Run #	Clock Ratio	MC Clock Period	PHY Clock Period	DRAM SPEED_BIN	Boundary Optimizations	Optimizations	Timings
01	4:1	3332 ps	833 ps	DDR4_2400	Enabled	No optimization, initial version.	Violated
02	4:1	3332 ps	833 ps	DDR4_2400	Disabled	After some optimizations.	Met !

Tabelle 3: DDR4 Controller Implementierungsergebnisse auf FPGA HW

Folgende Interfaces wurden definiert und umgesetzt:

Main Interfaces	Purpose	Protocol
CPU <-> MC	Data	AXI4
CPU <-> MC	Config	AXI4Lite
MC <-> PHY	Data	Xilinx MIG PHY

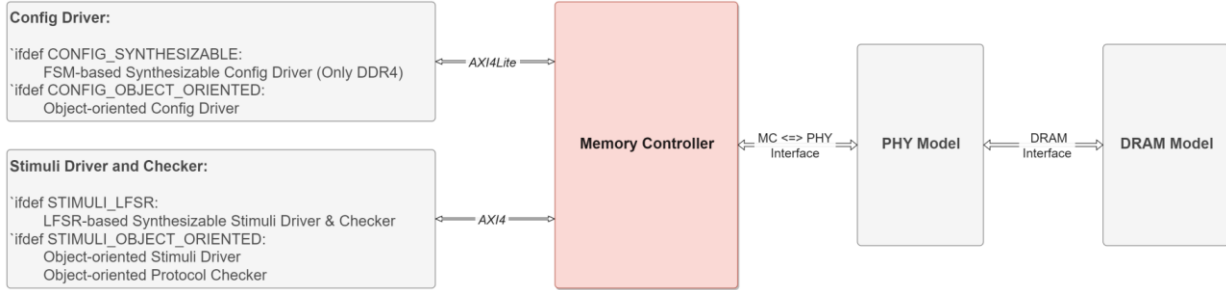


Abb. 10: DDR4 Memory Controller Main Interfaces & Top-Level Testbench

Abb. 10 zeigt die Top-Level Testbench, die zur Validierung auf FPGA und zur Simulation benutzt worden ist. Via diverser Tests und Simulationen wurde die Funktionalität (Write und Read Transactions) sichergestellt, siehe auch Abb. 11 und Tabelle 4.

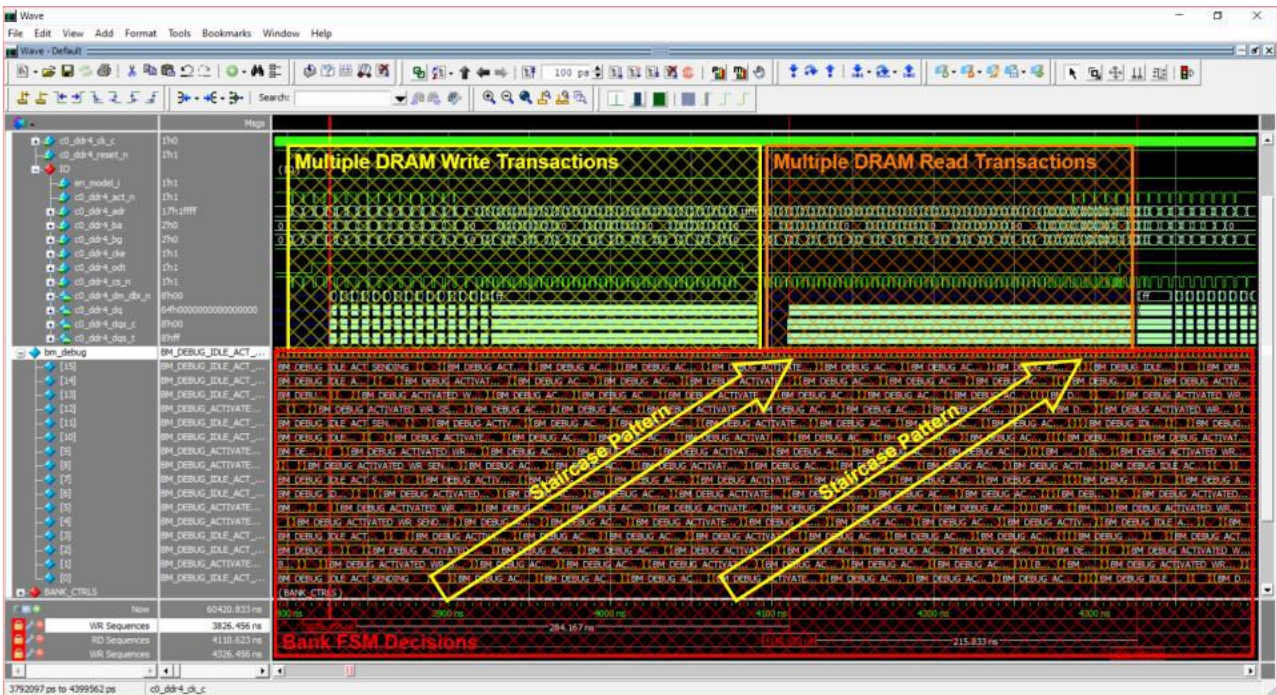


Abb. 11: DDR4 Memory Controller Simulation

Timing Scope	Timing Purpose	Timing Abbreviation	Configuration Value	Waveform Value	Unit	Expectations Met?	Note
Bank Timers	ACT => CAS	tRCD	39	39	DRAM tCK	Yes	-
	ACT => PREpb	tRAS	90	90	DRAM tCK	Yes	-
	WR => PREpb	tWRtoPRE	67	67	DRAM tCK	Yes	-
	WRap => ACT	tWRtoPRE + tRPpb	-	-	-	-	1
	RD => PREpb	tRTP	16	16	DRAM tCK	Yes	-
	RDap => ACT	tRTP (+8) + tRPpb	-	-	-	-	1
	PREpb => ACT	tRPpb	39	41	DRAM tCK	Yes	2
Command Mux Timers	ACT => ACT	tRRD	16	16	DRAM tCK	Yes	3
	CAS => CAS	tCCD	8	8	DRAM tCK	Yes	4
	WR => RD	tWRtoRD	49	49	DRAM tCK	Yes	-
	RD => WR	tRDtoWR	39	39	DRAM tCK	Yes	-
	PREpb => PREpb	tPPD	4	4	DRAM tCK	Yes	-
	PREab => ACT	tRPab	45	46	DRAM tCK	Yes	5
Refresh Timers	REFpb => REFpb	tpbR2pbR	192	192	DRAM tCK	Yes	-
	tREFIpb	tREFIpb	1041	?	DRAM tCK	Yes	6
	tREFIab	tREFIab	8328	8332	DRAM tCK	Yes	5, 7
	tRFCpb	tRFCpb	192	194	DRAM tCK	Yes	2
	tRFCab	tRFCab	384	400	DRAM tCK	Yes	5, 7
PHY Timings	Read Latency	RL	42	42.5	DRAM tCK	Yes	-
	Write Latency	WL	20	20	DRAM tCK	Yes	-

Tabelle 4: DDR4 Timer und Timing Validierung

Anhand der Tabelle konnten die Wichtigsten für den DDR4 Controller relevanten Timings und Timer-Settings validiert werden.

DDR4 DRAM Controller für die ASIC-Integration

Die finale Spezifikation für die ASIC-Integration des DDR4 DRAM Controllers wurde festgelegt. Der DDR4 Controller wurde mit diesen Vorgaben und Randbedingungen für die 22nm ASIC-Technologie design, implementiert und evaluiert. Da der DDR4-PHY nicht in den aktuellen ASIC integriert werden konnte, wurde eine FIFO basierte Schnittstelle für das ASIC-FPGA Interface entwickelt und validiert, und dann den Partnern zur Verfügung gestellt (EKUT + TUDA).

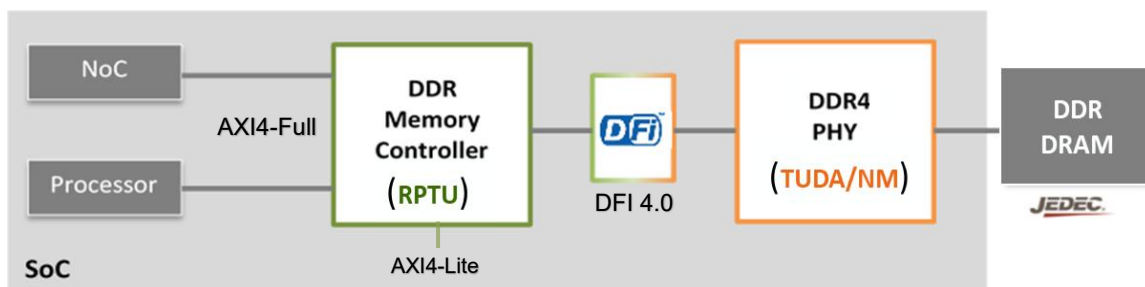


Abb. 12: DDR4 Memory Controller Integration – Überblick

Der DDR4 Memory Controller wurde mit Hilfe von Synthesis und Place&Route in der Zieltechnologie (GF 22nm FDx) evaluiert. Mit einer minimalen Zielfrequenz von 400MHz wurde die Synthese und P&R gestartet. Das Ergebnis ist in Abb. 13 und Abb. 14 zu sehen.

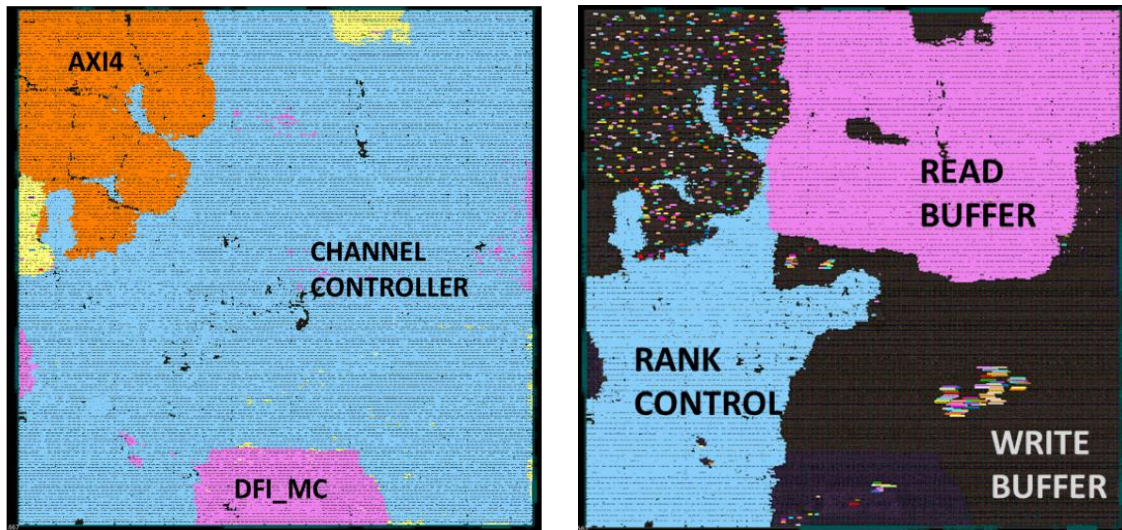


Abb. 13: DDR4 Memory Controller Implementierung in GF 22nm FDX

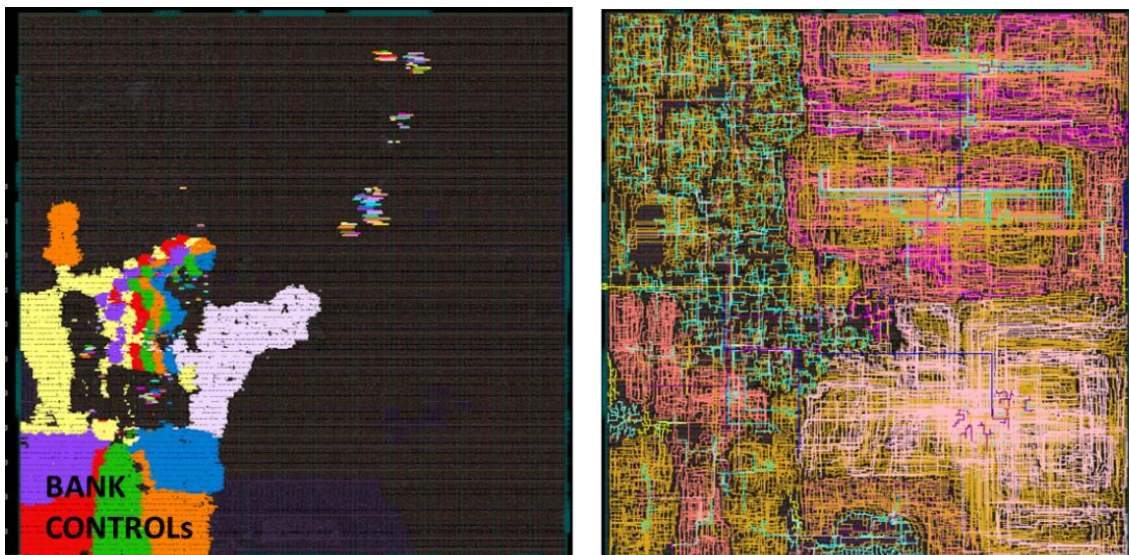


Abb. 14: Details der DDR4 Memory Controller Implementierung in GF 22nm FDX

- Eine Frequenz von über 500 MHz konnte unter Worst-Case Bedingungen erreicht werden. Die Controller-Fläche kann mit circa $A = 0.1 \text{ mm}^2$ bei einer Standard-Zell-Utilization von $\sim 70\%$ angegeben werden. Bei einem Clock-Ratio von 1:4 ermöglicht dies Datenraten von mehr als 3.200 Gbps/pin, z.B. für den DRAM Standard DDR4-3200.
- Die Ressourcenverwendung (Fläche) wird hauptsächlich durch Write/Read FIFOs (Buffer) und anderen Write/Read-Speichern (FFs) dominiert. Dies umfasst circa 70% der Fläche.
- Der DDR4 Controller ist von Benutzerseite mit einem AXI4-Full Burst-Interface ausgestattet und benutzt zur Konfiguration ein AXI4-Lite Interface. Zur Kommunikation mit dem DDR4-PHY wurde ein DFI4.0-Interface entworfen und implementiert. Dieses wurde anhand eines Behavioral-Modells des DDR4-PHYs validiert und getestet.

Memory Controller Instanzierung:

```
import pkg_axi_if::*;
import pkgAXI4::*;
import pkg_mem_ctrl::*;
module mem_ctrl #(
) (
    axi4lite_cfg_slv ...,
    axi4full_slv_ ...,
    dfi_*_pN ...
    ...
);
```

Abschließend kann der DDR4 DRAM Controller für ASIC-Implementierungen nun vollständig verwendet und zusammen mit den DDR4-PHY integriert werden.

FMC I/O ASIC-to-FPGA Schnittstelle - Design und Implementierung

Um eine reibungslose und einfache Anbindung des geplanten ASICs an einen FPGA zu ermöglichen, wurde ein FMC I/O ASIC-to-FPGA Interface entwickelt. Dies soll mit einer maximalen Frequenz von 250 MHz über Standard-I/Os (GPIOs) Daten in beide Richtungen übertragen können. Dazu können 68 Low-speed Pins des Standard FMC Steckers / Interfaces verwendet werden bzw. sind verfügbar. Damit wurde die I/O-Pin-Anzahl erheblich eingeschränkt. In Leserichtung wurden doppelt so viele I/Os (32) wie in Schreibrichtung (16) verwendet, da üblicherweise die Menge der zu übertragenden Lesedaten größer ist als die der Schreibdaten. Das Streaming-Interface wurde von TUDA ESA zur Verfügung gestellt und damit wurde hier eine einfachere Integration des Design möglich, siehe auch Abb. 15.

Das Modul „ASIC_FMC_top“ wurde beispielhaft implementiert. Im Zuge dessen konnte ein Floorplan mit Hilfe von EDA-Tools der Fa. Synopsys erstellt werden. Ein nahezu vollständiger PAD-Ring konnte aufgebaut werden, um das entwickelte Modul bzgl. Fläche und Performance zu evaluieren. Dies konnte hier nun erfolgreich gezeigt werden. Das Modul genügt den Anforderungen und erreicht bereits im ersten Implementierungsdurchlauf eine max. Frequenz von 200 MHz (extern - GPIO - Pins) und 400 MHz (intern – ASIC). Abb. 16 zeigt das in 22nm FDX implementierte „ASIC_FMC_top“ Modul.

```

module ASIC_FMC_top (
    input wire          clk,
    input wire          rst_n,
    input wire [3:0]    clock_dev_setting,
    input wire          RETCIN,

    // Receiver
    output wire         stream_valid_o,
    input wire         stream_ready_i,
    output wire [15:0]  stream_data_o,
    output wire [ 1:0]  stream_byte_enable_o,
    output wire         stream_last_o,

    // Sender
    output wire         stream_ready_o,
    input wire         stream_valid_i,
    input wire [15:0]  stream_data_i,
    input wire [ 1:0]  stream_byte_enable_i,
    input wire         stream_last_i,

    // Sender to Pads
    output wire         SERIAL_SEND_clk,
    input wire         SERIAL_SEND_ready,
    output wire         SERIAL_SEND_valid,
    output wire [15:0]  SERIAL_SEND_data,
    output wire [ 1:0]  SERIAL_SEND_byte_en,
    output wire         SERIAL_SEND_last,

    // Receiver to Pads
    input wire         SERIAL_RECV_clk,
    output wire        SERIAL_RECV_ready,
    input wire         SERIAL_RECV_valid,
    input wire [31:0]  SERIAL_RECV_data,
    input wire [ 3:0]  SERIAL_RECV_byte_en,
    input wire         SERIAL_RECV_last
);

```

Abb. 15: I/O-Signale des ASIC_FMC_top Moduls



Abb. 16: Floorplan und Implementierung (Layout nach Place&Route) des ASIC_FMC_top Moduls

3.2.6 Arbeitspaket AP6: Demonstratoren + Use Cases + Standardisierung

Projektpartner: EKUT, IFX, ITE, MB, MNRS, NM, RB, RUB, TUDA, **RPTU (TUK)**, TUM, CRE

Ziele:

In AP6 sollte die Anwendbarkeit des automatisierten FlexKI Retargeting-Verfahrens für KI-Compiler und der HW/SW-Co-Design Verfahren anhand von zwei realen Demonstratoraufbauten nachgewiesen werden. Der erste Demonstrator verwendet eine Anwendung mit dem Ziel der Detektion von Vulnerablen Road Usern aus dem Bereich des automatisierten Fahrens und zeigt eine optimierte Abbildung auf zwei heterogene am Markt erhältliche Systeme der Klasse 1, wie NVIDIA Orin oder Qualcomm Snapdragon. Der zweite Demonstrator zeigt die Verwendbarkeit der FlexKI HW-Plattform anhand eines Use-Cases aus dem Bereich des Vehicle Health Monitoring. Darüber hinaus wurde durch das MANNHEIM-FlexKI-Konsortium eine offene Referenzimplementierung des MANNHEIM-FlexKI Deployment und Retargeting Flows zur Verfügung gestellt.

Aufgaben:

A6.1: Auswahl geeigneter Netzwerke zur Detektion von Vulnerable Road Users und geeignete Wahl der Demonstrator Plattformen der Klasse 1 Off-the-Shelf.

A6.2: Vehicle Health Monitoring (Demonstrator Plattformen der Klasse 2)

A6.3: Weiterentwicklung bestehender Standards um die Inhalte des Flex-KI Projekts

Leitung:

MB

Meilensteine:

M6.1: Konzept für Demonstratoren Use Case 1 und 2

M6.2: Demonstration von Teilaspekten des Deployment und Optimierungsflow

M6.3: Demonstration der Use Cases auf den Zielplattformen

M6.4: Zusammenfassung der Standardisierungsaktivitäten

Tabelle 5: Ressourcenplanung zu Arbeitspaket AP6

Partner	A6.1 [PM]	A6.2 [PM]	A6.3 [PM]	Summe [PM]
EKUT	6,00	30,00	3,00	39,00
IFX	15,00	25,00	4,00	44,00
ITE	0,00	0,00	1,00	1,00
MB	31,00	6,00	1,00	38,00
MNRS	0,00	6,00	1,00	7,00
NM	0,00	0,00	1,00	1,00
RB	0,00	8,00	1,00	9,00
RUB	2,00	2,00	1,00	5,00
TUDA	0,00	5,00	1,00	6,00
RPTU (TUK)	0,00	0,00	1,00	1,00
TUM	0,00	6,00	1,00	7,00
CRE	0,00	11,00	1,00	12,00
Summe	54,00	99,00	17,00	170,00

Ergebnisse:

A6.3: Standardisierungsaktivitäten

Die Projektergebnisse wurden nachhaltig über den Kreis des Projektkonsortiums hinaus in der Industrie verankert, indem die relevanten Standards mitgestaltet wurden. Dabei wurden formelle und informellere / de-facto Standards betrachtet, wie z.B. Accellera, RISC-V International, OpenHW Group, die Apache TVM-Community, TinyML Foundation, openXSAM und die Eclipse-Foundation. Alle Partner in MANNHEIM-FlexKI beteiligen sich an Standardisierungsaktivitäten.

Die RPTU war mit einem Monat in Arbeitspaket 6 Beitrag B6.3.10 für die Standardisierungsaktivitäten Robustheitstest mitwirkend.

B6.3.10 : (RPTU) Unterstützung der Diskussion eines neuen Standards für Robustheitstest und Degradationsdefinition von Halbleitern

Zusammen mit der Neumonda GmbH unterstützte die RPTU die Standardisierungsbemühungen, um hier Erweiterungen vorzuschlagen. Insbesondere bei der Signalintegrität oder der Konfiguration der I/O Transceiver konnten Erkenntnisse gewonnen werden.

4 Voraussichtlicher Nutzen, insbesondere Verwertbarkeit der Ergebnisse

Die Ergebnisse des Projekts wurden durch Seminare, Workshops und wissenschaftliche Publikationen sowie durch Präsentationen auf Konferenzen verbreitet. Somit wurde die wissenschaftliche Verwertung der durchgeführten Untersuchungen und deren Ergebnissen sichergestellt. Des Weiteren erfolgte innerhalb des Projekts eine enge Zusammenarbeit mit Industriepartnern. Im Fall der RPTU fand auch eine enge Kooperation mit der Firmen Creonic GmbH und Neumonda GmbH statt. Die innerhalb dieses Projektes gesammelten Erkenntnisse können damit auch in zukünftige Projekte im Speicherbereich (Controller, DRAM PHYs) einfließen.

Im speziellen sind die Ergebnisse bezüglich des ASIC und FPGA-basierten DDR4 DRAM Controller sehr vielversprechend, da aufgezeigt werden konnte, dass der von der RPTU entwickelte Controller nicht nur bezüglich der Performanz besser, sondern auch bezüglich der Implementierungskomplexität konkurrenzfähig ist. Die im Projekt in Zusammenarbeit mit den Hauptpartnern TUDA, EKUT, NM und Creonic entwickelten IPs sowie dem FlexKI-SoC Demonstrator wird auch zukünftig für die Präsentation der wissenschaftlichen Erkenntnisse des Projekts auf Messen und Konferenzen genutzt und trägt damit zur wissenschaftlichen Verwertung der durchgeführten Untersuchungen bei.

5 Fortschritte auf dem Gebiet des Vorhabens bei anderen Stellen

Im Projekt MANNHEIM-FlexKI wurde regelmäßig der Stand der Wissenschaft ermittelt und in die Forschungsarbeiten der einzelnen Partner integriert. Hierzu haben regelmäßig Informationsrecherchen auf den einschlägigen Onlineplattformen stattgefunden.

Ein Austausch mit anderen Stellen fand im Rahmen von projektinternen Workshops sowie wissenschaftlichen Konferenzen statt.

6 Anhang

6.1 Geplante Veröffentlichungen

[1] Multi-Partner Project: Flexible AI Deployment to Flexible Platforms - from MOPS to TOPS

[2] An OpenSource DDR4 DRAM Controller adaptable for AI Workloads