

Entwicklung einer Auslesekarte zur schnellen Verarbeitung von Detektordaten

ALICE RUN2 Auslese-Upgrade und Evaluierung von Datenfluss-Hardwarebeschreibung für Ausleseanwendungen der Hochenergiephysik¹

Heiko Engel²

Abstract: Im Rahmen dieser Arbeit wurde eine FPGA-basierte Ausleselösung für die ALICE Data Acquisition und High Level Trigger Systeme in RUN 2 entwickelt, die durch ihre generische Architektur weit über die geplanten Anwendungsfälle und Zeiträume hinaus zum Einsatz kommt. Ein zentraler Bestandteil ist hierbei die hardwareseitige Vorverarbeitung von Detektordaten bereits im FPGA der Auslesekarte, ohne die eine Online-Aufbereitung der Daten in diesem Maße nicht möglich gewesen wäre. Durch die Umsetzung des verwendeten Vorverarbeitungsalgorithmus in einer Datenflussbeschreibung konnte darüber hinaus eine Implementierung erarbeitet werden, bei der sich Ressourcenverbrauch und Latenz kaum von der händischen Implementierung unterscheiden. Die deutliche reduzierte Entwicklungszeit, der geringere Umfang, die leichtere Wartbarkeit, sowie die einfachere Partitionierung konnten somit zeigen, dass Hardwarebeschreibungslösungen auf höheren Abstraktionsebenen auch in der Hochenergiephysik enorme Einsparungen bringen können.

1 Einleitung

Die Grundlagenforschung versucht mit einer Vielzahl von Experimenten zu unterschiedlichsten physikalischen Aspekten ein tieferes Verständnis unserer Welt zu fördern. Die Teilchenkollisionsexperimente der Hochenergiephysik tragen hierbei zum zentralen Verständnis der Struktur der Materie bei. All diese Experimente sind hoch-komplexe Systeme mit einer Vielzahl von Komponenten, Detektoren und Technologien. Sie nutzen unterschiedliche physikalische Eigenschaften der Kollisionsprodukte aus, um diese zu detektieren, deren Spuren, Energien oder Impulse zu messen und daraus die physikalischen Stoßprozesse zu rekonstruieren. Eine gemeinsame Herausforderung dieser Experimente ist die Koordination, Synchronisation und Auslese dieser massiv-parallelen Systeme bei Ereignisraten von mehreren Kilohertz. Eine Technologie, die bei vielen Ausleseketten an verschiedenen Stellen eingesetzt wird, sind programmierbare Hardware-Bausteine in der Form von “Field Programmable Gate Arrays” (FPGAs). Mit FPGAs kann benutzerdefinierte Logik umgesetzt werden, die für kleine bis mittlere Stückzahlen deutlich günstiger als ASICs ist und jederzeit an geänderte Bedingungen angepasst werden kann. Außerdem können FPGAs durch ihre fast beliebige Parallelität eine Rechenleitung bieten, die in vielen Anwendungsfällen auch von CPU, GPU oder DSP Systemen nicht erreichbar

¹ Englischer Titel der Dissertation: Development of a Read-Out Receiver Card for Fast Processing of Detector Data – ALICE HLT Run 2 Readout Upgrade and Evaluation of Dataflow Hardware Description for High Energy Physics Readout Applications [En19]

² Infrastruktur und Rechnersysteme in der Informationsverarbeitung, Goethe-Universität Frankfurt a. M.

ist. FPGAs haben in den letzten Jahren ihren festen Platz in den Ausleseketten von Physikexperimenten gefunden. Mit immer größeren und schnelleren Chips werden außerdem immer komplexere Verarbeitungsschritte in Hardware möglich.

Der Large Hadron Collider (LHC, [EB08]) an der Europäischen Organisation für Kernforschung (CERN) ist derzeit der leistungsstärkste Teilchenbeschleuniger der Welt. Dutzende verschiedene Experimenten sind daran angebunden. ALICE [Aa08] ist eines der vier größten Experimente am LHC und auf die Erforschung von Schwerionenkollisionen sowie des dabei entstehenden Quark-Gluon-Plasma spezialisiert. ALICE besteht aus 19 Sub-Detektoren, die mit Hilfe eines komplexen Synchronisations-, Trigger- und Auslesesystems zeitgleich betrieben werden. Über 500 serielle optische Links stellen den Datentransport zwischen den Detektoren und dem Data Acquisition (DAQ) System sicher. Parallel wird eine Kopie der Detektordaten an den High Level Trigger (HLT, [Ri07]) geschickt, der die Daten rekonstruiert, kalibriert und komprimiert. Durch die Kompression und Speicherung vorverarbeiteter Daten können die ursprünglichen Rohdaten verworfen und damit die Datenmenge zur dauerhaften Speicherung signifikant reduziert werden. Nur diese Kompression macht es möglich, die bei den Schwerionenkollisionen in ALICE anfallenden Datenmengen überhaupt speichern zu können.

Eine zentrale Komponente dieser Auslesearchitektur ist die Schnittstelle zwischen den optischen Detektorlinks und den Rechnerclustern der DAQ und HLT Systeme. In der ersten LHC Runperiode (RUN 1) zwischen 2009 und 2012 wurden hierfür speziell entwickelte FPGA-basierte Auslesekarten eingesetzt, die Read-Out Receiver Cards (RORCs). In der HLT-Anwendung war neben der reinen Datenauslese zudem eine Datenvorverarbeitung in Form eines Hardware Cluster Finder Algorithmus integriert. Diese hoch-parallele FPGA-basierte Vorverarbeitung ersparte beträchtliche Mengen an CPU Rechenleistung im Vergleich zu äquivalenten Verarbeitungsschritten auf CPUs in Software.

2 Motivation und Ziele der Arbeit

Nach dem Ende der ersten LHC Runperiode wurden mehrere ALICE Detektorsysteme weiter ausgebaut und für höhere Daten- und Ereignisraten in der zweiten LHC Runperiode (RUN 2) ab 2015 vorbereitet. Die DAQ und HLT Systeme wurden zum größten Teil mit neuer Hardware ersetzt, um den Anforderungen von RUN 2 gerecht werden zu können. Die bisherigen FPGA Auslesekarten konnten auf Grund der höheren Linkraten sowie einer veralteten Host-Schnittstelle nicht mehr mit der verbesserten Detektorhardware und den neuen Server-Maschinen verwendet werden. Ein Ziel dieser Arbeit war daher die Bereitstellung und der Betrieb einer Ausleselösung für ALICE in RUN 2, die den erhöhten Anforderungen gerecht wird, mit den bestehenden Schnittstellen kompatibel ist, eine zeitgemäße und zukunftssichere Technologie bietet und für Erweiterungen der Verarbeitungsschritte in Hardware gerüstet ist. Die Auswahl der Hardware war von Anfang an darauf ausgelegt, dass dieselbe Plattform in den DAQ und HLT Systemen in RUN 2 zum Einsatz kommt. Firmware und Software Implementierungen erfolgten für beide Gruppen separat, da die Anwendungen und die aus RUN 1 wiederverwendeten Technologien deutlich

verschieden sind. Die Firmware-Entwicklungen für den HLT Produktivbetrieb sowie die Integration ins HLT Datenverarbeitungssystem waren ebenfalls Ziele dieser Arbeit.

In einem ersten Schritt musste eine Hardware gefunden werden, die die erhöhten Anforderungen erfüllen kann, in entsprechender Stückzahl rechtzeitig verfügbar ist und in die bestehende Experiment-Infrastruktur integriert werden kann. Hierzu wurden Hardware-Technologien evaluiert, Prototypen erstellt und getestet sowie eine experimentweite Beschaffung organisiert und koordiniert werden.

Ein zweiter Aspekt war die Integration der neuen Auslesekarten in die Hardware- und Softwareumgebung des HLT. Dies umfasste die Entwicklung der Auslesefirmware sowie die Einbindung in das bestehende verteilte HLT Datenverarbeitungssystem. Weiterhin war die Integration, Wartung und Weiterentwicklung der FPGA-basierten Datenvorverarbeitung in Form des Hardware Cluster Finders erforderlich, der auch in RUN 2 einen zentralen Bestandteil der HLT Datenverarbeitungskette bildet. Diese beiden Arbeitspakete waren streng vom Zeitplan des Experiments abhängig. Rechtzeitig vor Beginn der zweiten LHC Runperiode musste eine zuverlässige Hardwarelösung in den Produktivsystemen von DAQ und HLT installiert sein. Diese musste in die Auslese- und Datenverarbeitungssoftware integriert sein und gründlich getestet für die Aufzeichnung von Experimentdaten bereitstehen.

Ein dritter Aspekt dieser Arbeit betrifft die Beschreibung von Datenverarbeitungsschritten in Hardware. Die gängigen Hardwarebeschreibungsmethoden arbeiten üblicherweise auf der Register-Transfer-Ebene, die Datenbewegungen und -Verarbeitungsschritte zwischen Registern Takt-synchron beschreibt. Die Implementierung von Algorithmen auf dieser Ebene ist möglich und wird seit Jahren so betrieben. Auch der Hardware Cluster Finder ist auf dieser Ebene beschrieben. Jedoch bringt die Beschreibung eines Algorithmus auf dieser Ebene einen hohen Aufwand an Entwicklung, Verifikation und Wartung mit sich. Bereits kleine Änderungen am Algorithmus können große Arbeiten an der Implementierung bedeuten. In den letzten Jahren wurden einige Werkzeuge zur Hardwarebeschreibung auf algorithmischer Ebene in Form von High-Level Synthese (HLS) veröffentlicht, die insbesondere den Aufwand von Entwicklung, Verifikation und Wartung vereinfachen sollen. Die meisten dieser Ansätze verwenden Teilmengen imperativer Programmiersprachen, die mit Anweisungen erweitert wurden, um die zeitliche und räumliche Parallelität einer Hardware-Implementierung auszudrücken. Ein vielversprechender Ansatz, der sich konzeptionell von den populärsten Ansätzen etwas abgrenzt, ist die Datenfluss-Programmierung. Datenfluss-Architekturen hatten es zwar nie in kommerziell erfolgreiche Rechnersysteme geschafft, die Konzepte lassen sich jedoch vergleichsweise einfach auf FPGAs abbilden, um damit hoch-effektive Datenverarbeitungssysteme zu entwickeln. Am Beispiel des Hardware Cluster Finder sollte im Rahmen dieser Arbeit der Datenfluss-Ansatz zur High-Level Synthese untersucht werden. Die bestehende Implementierung des Hardware Cluster Finders auf Register-Transfer-Ebene diente hierbei als direkte Referenz, um den Datenfluss-Ansatz zu evaluieren und die Realisierbarkeit einer algorithmischen Hardwarebeschreibung für die Datenverarbeitung einer Experimentauslese zu untersuchen.

3 Die C-RORC Hardware Plattform

Im Rahmen dieser Arbeit wurde die Common Read-Out Receiver Card (C-RORC) als eine gemeinsame Auslesekarte für die ALICE DAQ und HLT Systeme in RUN 2 entwickelt. Diese Karte musste in der Lage sein, die optischen Detektorlinks sowohl mit der in RUN 1 verwendeten Si-

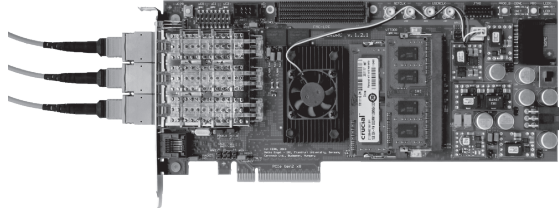


Abb. 1: Foto der C-RORC Karte.

gnalrate, als auch mit erhöhten Raten für die in RUN 2 erweiterten Detektoren auslesen zu können. Die Schnittstelle zum Host-Rechner musste hierbei genügend Bandbreite zur Verfügung stellen, um die eingehenden Detektordaten weiterzuleiten. Im Zuge des technischen Fortschritts war eine Lösung mit erhöhter Link-Dichte gewünscht, da dies die Anzahl der benötigten Host-Rechner zur Installation der Karten reduziert. Weiterhin mussten ausreichend FPGA Ressourcen verfügbar sein, um die aus RUN 1 im HLT bereits eingesetzte Datenvorverarbeitung in Form des Hardware Cluster Finders in RUN 2 mit den erhöhten Anforderungen weiterführen und erweitern zu können. Für die HLT-Anwendung wurde Speicher auf der Karte benötigt, um Detektordaten zu Testzwecken abspielen zu können. Eine spezielle elektrische Schnittstelle wurde von der DAQ-Anwendung benötigt, um die bestehende Flusskontrolle mit dem Triggersystem weiterverwenden zu können. Neben diesen unbedingt notwendigen technischen Voraussetzungen waren eine kompakte Bauform, eine dynamische Auswahloption der seriellen Signalaraten sowie ein zuverlässiges und flexibles Konfigurationssystem für die effiziente Integration und den Betrieb in den Produktivsystemen vorgesehen.

Weder zum Zeitpunkt der Hardware-Auswahl noch zu einem späteren Zeitpunkt war eine kommerzielle Hardware-Plattform verfügbar, die diese Anforderungen auch nur annähernd erfüllen konnte. Hardware-Entwicklungen anderer Experimente waren entweder noch nicht weit genug fortgeschritten, oder nicht kompatibel mit den ALICE Anforderungen. Aus diesem Grund wurde die C-RORC als eine eigene FPGA-basierte Auslesekarte konzipiert und entwickelt. Die Link-Dichte, die generischen Komponenten für den anwendungsspezifischen Einsatz, sowie die Innovationen im Bereich Remote-Wartbarkeit waren zum damaligen Zeitpunkt in diesem Formfaktor einzigartig. Ein Foto der Karte ist in Abbildung 1 gezeigt.

Die C-RORC wurde im Rahmen dieser Arbeit als FPGA Auslesekarte für ALICE konzipiert, umgesetzt und integriert. Eine Entwicklung von eigenen Karten bringt immer mehr Herausforderungen insbesondere für kleine Entwicklerteams mit sich. Die Komplexität steigt durch immer mehr Funktionsumfang und schnellere Schnittstellen der Komponenten. Über tausend Pins allein am FPGA und Firmwareentwicklung bereits für die Stromversorgung einer Karte sind keine Seltenheit. Im selben Maße wächst der Umfang der Dokumentation sowie die Kosten und das erforderliche Wissen für passende Entwicklungs-, Simulations- und Messwerkzeuge. Nicht selten ist die nächste Generation FPGAs verfügbar, bevor eine Eigenentwicklung wirklich in Produktion gehen kann.

Die Konzeption der ALICE Auslesehardware für RUN 2 wurde 2010 begonnen. Die ersten beiden Prototypen-Serien waren Anfang und Mitte 2013 verfügbar. Zu diesem Zeitpunkt hat sich das ATLAS Experiment dazu entschieden, dieselbe Hardware für ihr Auslesesystem in RUN 2 einzusetzen. Die Serienproduktion der Karten erfolgte daraufhin in Zusammenarbeit zwischen den ALICE und ATLAS Gruppen. Knapp 400 C-RORCs wurden bis Mitte 2014 gefertigt, wobei Teile der Koordination und ein großer Teil der Entwicklung einer Testumgebung zur Verifikation aller Karten bereits beim Hersteller im Rahmen dieser Arbeit geleistet wurden. Die Karten wurden im Herbst 2014 in die ALICE und ATLAS Produktivsysteme eingebaut und waren bis zum Ende von RUN 2 im Dezember 2018 zentrale Bestandteile der jeweiligen Experiment-Auslese.

Neben den Hauptanwendungen in den RUN 2 Auslesesystemen von ALICE und ATLAS hat die Karte eine Anwendung in verschiedenen Test- und Evaluationssystemen gefunden. Teilweise sollen die Karten auch in RUN 3 weiterverwendet werden. Die nach RUN 2 in ALICE nicht mehr benötigten Karten sind bereits für andere Experimente vorgesehen.

4 Integration in den ALICE High-Level Trigger

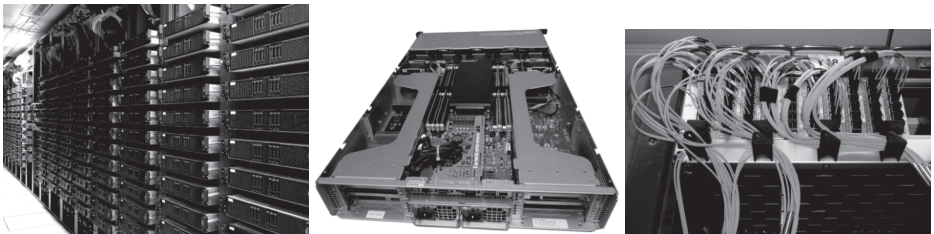


Abb. 2: Links: Foto des ALICE HLT Clusters zu Beginn von RUN 2. Mitte: Offener HLT Rechnerknoten mit C-RORC in rot. Rechts: Optische Aufteilkabel zur Glasfaser-Installation.

Nach Ende der ersten LHC Runperiode wurde die gesamte HLT-Hardware ersetzt um die Daten- und Ereignisraten in RUN 2 verarbeiten zu können. Der HLT besteht in RUN 2 aus 180 identischen Rechner-Knoten die jeweils eine GPU enthalten und mit InfiniBand und Ethernet miteinander verbunden sind. 74 dieser Knoten enthalten außerdem eine C-RORC als zentrale Ein- und Ausgangsschnittstelle für Detektordaten in und aus dem HLT. Die 74 C-RORCs bilden hierbei die Verbindung zwischen über 500 Glasfaserleitungen und den HLT Rechnerknoten. Bis zu 12 Detektorlinks werden pro C-RORC ausgelesen. Glasfaser-Aufteilkabel verbinden die parallel-optischen Schnittstellen der C-RORCs mit der bestehenden Experiment-Infrastruktur. Fotos des Clusters, eines HLT Rechners sowie der Glasfaserinstallation sind in Abbildung 2 gezeigt. Die C-RORCs sind in zwei Datenflussrichtungen im HLT integriert. Sie bilden die Eingangsschnittstelle des HLT, um die Detektordaten von den optischen Links entgegenzunehmen, gegebenenfalls bereits im FPGA mit dem Hardware Cluster Finder vorzuverarbeiten, und an den Host-Rechner weiterzugeben. Weiterhin stellen die C-RORCs die Ausgangsschnittstelle des HLT, indem sie die vom HLT komprimierten und rekonstruierten Daten vom Host-Rechner lesen und über die optischen Links an das DAQ System schicken. Die Signalrate der optischen Links ist zur Laufzeit konfigurierbar. Die Schnittstelle zum Host-Rechner ist mit einer eigenen Direct Memory

Access (DMA) Implementierung zum direkten Zugriff auf den Host-Speicher realisiert. Im Host-Rechner bildet ein Kernelmodul die Grundlage für die Kommunikation mit der Karte per Software sowie die Reservierung von DMA Speicher. Sämtliche Interaktionen mit der Karte erfolgen aus dem Benutzer-Kontext mit einer Treiber-Bibliothek. Die C-RORCs sind ins HLT Datenverarbeitungssystem als Datenquellen und -Senken unter Verwendung dieser Bibliothek integriert. Jeder Detektorlink ist von einem eigenständigen Prozess verwaltet. Dies ermöglicht es, beliebige Link-Konfigurationen unabhängig voneinander zu betreiben. Sämtliche Firmware-Varianten, die gesamte Integration ins HLT-System sowie große Teile der Treiber-Bibliothek wurden im Rahmen dieser Arbeit entwickelt.

5 Hardware Cluster Finding

Der Hardware Cluster Finder ist ein Algorithmus, der in den Rohdaten des größten Detektors in ALICE, der Zeit-Projektionskammer (TPC, [A110]), nach charakteristischen Signaturen, sogenannten “Clustern”, sucht und deren Parameter berechnet und ausgibt. Die Rohdaten werden nach der Auslese verworfen und nur die komprimierten Cluster werden gespeichert. Eine erste Implementierung dieses Algorithmus als parallele Verarbeitungspipeline direkt im Auslese-FPGA war bereits während der ersten LHC Runperiode in den HLT Auslesekarten integriert.

Die Hardware-Vorverarbeitung der Daten war hierbei schon deutlich effizienter als eine vergleichbare Software-Implementierung auf CPUs und konnte die benötigte CPU Rechenleistung im HLT Cluster signifikant reduzieren. Der Algorithmus kann mit laufenden gewichteten Summen beschrieben werden, die sich effizient mit FPGAs berechnen lassen. Abbildung 3 zeigt ein Beispiel der aus Rohdaten berechneten Cluster-Informationen.

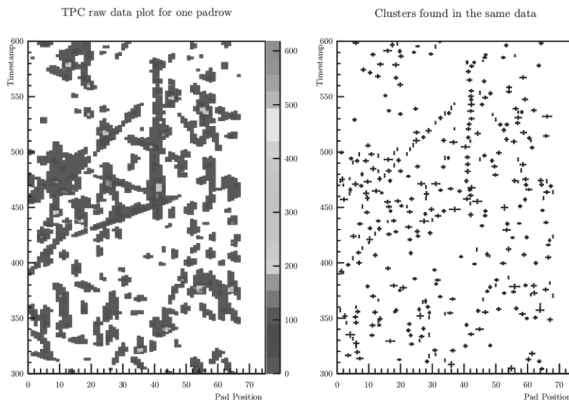


Abb. 3: TPC Rohdaten mit farbkodierten Ladungswerten (links) und die daraus extrahierten Clusterpositionen und -Breiten (rechts).

Die bestehende FPGA-Implementierung aus den RUN 1 Auslesekarte wurde in einem ersten Schritt auf die C-RORC portiert. Sechs Cluster Finder Instanzen sind hierbei pro C-RORC integriert. Die Daten von 216 TPC Ausleselinks werden von 36 C-RORCs im HLT vorverarbeitet. Eine zentrale Erweiterung für ALICE in RUN 2 war das Upgrade der

TPC Ausleseelektronik mit der Readout Control Unit 2 (RCU2, [A113]). Mit einer höher parallelisierten Auslesearchitektur konnte etwa die doppelte Datenrate vom Detektor zu den RCUs ausgelesen werden. Um diese Datenrate an die DAQ und HLT Onlinesysteme weiterreichen zu können, wurde die Signalrate der optischen Links entsprechend erhöht. Dies hatten deutliche Auswirkungen auf die Datenvorverarbeitung mit dem Cluster Finder in den HLT C-RORCs. Zum einen musste der Cluster Finder mit einer überproportional erhöhten Taktrate betrieben werden, da die verfügbare Bandbreite mit der neuen Detektor Hardware deutlich effektiver ausgenutzt wird. Dies hatte direkten Einfluss auf die maximale Latenz der Berechnungsschritte in Hardware und erforderte eine Optimierung sowohl der Datenpuffer als auch der Pipeline-Architektur um die erforderlichen Taktraten zu erreichen. Zum anderen war eine Abwärtskompatibilität mit dem vorherigen Datenformat gewünscht, um weiterhin beliebige Daten zu Testzwecken vom Kartenspeicher abspielen zu können. Beide Aspekte stellten hohe Ansprüche an die Firmwareentwicklung, da die erforderlichen Taktraten zusammen mit der kombinatorischen Komplexität des Algorithmus bereits nahe an die technischen Grenzen des FPGAs kamen. Diese Firmware-Variante wurde mit der Installation der RCU2 Karten ins Experiment Anfang 2016 im HLT Produktionssystem eingesetzt.

In Zusammenarbeit zwischen den HLT und TPC Gruppen konnte zudem durch die Simulation und Analyse der Roh- und Clusterdaten ein verbesserter Cluster Finding Algorithmus erarbeitet werden, der rauschinduzierte Detektorsignale deutlich besser unterdrücken kann. Die Hardware-Implementierung dieses Algorithmus wurde im Rahmen dieser Arbeit umgesetzt. Messungen haben gezeigt, dass sich die Anzahl gefundener Cluster dadurch um etwa 21 % (Daten mit der Gasmischung von 2017) bzw. 31 % (Daten mit der Gasmischung von 2016) im Vergleich zur vorherigen Implementierung reduziert. Die Qualität der Physikresultate ist von der geringeren Anzahl Cluster nicht betroffen und liefert teilweise sogar bessere Resultate als zuvor. Die reduzierte Anzahl an Clustern bedeutet direkt eine erhöhte Datenkompression im HLT und damit ein reduziertes Datenvolumen für die dauerhafte Datenspeicherung. Diese Firmwarevariante war von Mitte 2017 an im HLT aktiv und hat zusammen mit softwareseitigen Anpassungen die Datenkompressionsrate des HLT auf einen Faktor von über acht erhöht.

Die Bestimmung der Rechenleistung des Hardware Cluster Finders im Vergleich zu einer reinen Softwarelösung benötigt eine definierte Testumgebung. Die Software Referenz-Implementierung in Bezug auf die Physik-Resultate ist der Offline Cluster Finder, der volle Events auf einem einzelnen Knoten verarbeitet. Der Hardware Cluster Finder ist im Gegensatz dazu auf 216 Instanzen in 36 C-RORCs verteilt, die parallel die Daten von jeweils nur einem der 216 optischen Links pro Instanz verarbeiten. Ein Durchsatz-Vergleich von Hardware und Software-Lösung ist jedoch auf Knoten-Ebene möglich. Eine C-RORC mit sechs Cluster Finder Instanzen kann volle TPC-Ereignisse bearbeiten. Der Durchsatz dieses Setups kann mit dem maximalen Durchsatz von parallelen Offline Cluster Finder Instanzen auf demselben Knoten verglichen werden. Die Hardware Implementierung in einer einzelnen C-RORC unter Benutzung von vier CPU Kernen hat sich dabei als etwa einen Faktor zehn schneller als die reine Softwareimplementierung, oder äquivalent zu 480 CPU Kernen erwiesen. Die enormen Einsparungen an CPU Rechenleistung im HLT durch die Datenvorverarbeitung in Hardware sind damit offensichtlich.

6 Datenfluss-Implementierung des Cluster Finders

Die Detektoren und das Experiment wurden über RUN 2 hinweg kontinuierlich weiterentwickelt. Die Firmware-Entwicklungen, insbesondere am Hardware Cluster Finder, haben auch im Rahmen dieser Arbeit aufgezeigt, wie aufwendig bereits kleinere algorithmische Änderungen umzusetzen sein können. Nicht selten müssen dabei Firmware-Teile von Grund auf neu implementiert und verifiziert werden. Ein Grund hierfür ist, dass die Hardwarebeschreibung in der Regel auf Register-Transfer-Ebene stattfindet und damit die Partitionierung eines Algorithmus in Raum und Zeitrichtung komplett manuell erfolgen muss. Kleine algorithmische Änderungen können daher große strukturelle Auswirkungen auf diese Partitionierung haben. Ansätze zur High-Level Synthese versprechen dieses Problem zu lösen, indem sie die Hardwarebeschreibung auf die algorithmische Ebene heben und die Partitionierung auf Register-Transfer-Ebene einem Compiler überlassen. Im Rahmen dieser Arbeit wurde hierzu der Datenfluss-Ansatz untersucht. Ein Framework aus FPGA-Karte, einem Compiler sowie Softwarebibliotheken erlaubt es, Algorithmen aus einer Datenflussbeschreibung heraus unkompliziert auf der Hardware zu betreiben. Implementierungsdetails wie der Datentransfer zu und vom FPGA, Speicherreservierung, Flusskontrolle in der Firmware usw. werden funktionsfertig bereitgestellt und deren Interna bleiben vor dem Nutzer verborgen. Die Datenflussbeschreibung des Algorithmus wird vom Compiler in einen Datenfluss-Graphen übersetzt und mit Hilfe einer internen Bibliothek von Firmwarebausteinen auf das FPGA abgebildet. Die Datenfluss-Beschreibung erfolgt in dem hier verwendeten Framework in einer Sprache mit Java-ähnlichem Syntax, die einerseits auf eine in Hardware abbildbare Teilmenge reduziert und andererseits um Aspekte der Steuerung von räumlicher und zeitlicher Parallelität auf der Hardware erweitert wurde. Der Compiler stellt mit Regeln sicher, dass die somit beschriebene Funktionalität als effiziente Verarbeitungs-Pipeline im FPGA implementiert werden kann. Die Verarbeitungsschritte werden in "Kernels" eingebunden. "Manager" beschreiben die Struktur auf Systemebene und verbinden Kernels untereinander, mit Schnittstellen zum Host-Rechner, zum Speicher auf der Karte oder zu Netzwerkkomponenten. Das Framework bringt eine Simulations- und Entwicklungsumgebung mit, in der sowohl der Datenfluss-, als auch der Host-Code entwickelt, verifiziert und auf der Zielhardware ausgeführt werden kann.

Der Hardware Cluster Finder Algorithmus wurde in dieser Arbeit mit dem Datenfluss-Framework umgesetzt. Die Struktur dieser Implementierung hat sich dabei im Vergleich zur bereits bestehenden hoch-optimierten low-level Implementierung kaum verändert. Der Algorithmus konnte unkompliziert in der Datenfluss-Sprache beschrieben werden. Die Beschreibung von mathematischen Operationen unabhängig von den Datentypen, Latenzen und mit den gängigen Operatoren aus der Softwareprogrammierung vereinfachte und beschleunigte die Entwicklung beträchtlich. Zur Integration der Datenfluss-Implementierung des Cluster Finders in die HLT-Umgebung wurden verschiedene Möglichkeiten untersucht. Eine Ersetzung der C-RORC mit einer reinen Datenfluss-Implementierung der gesamten Ausleselösung war auf Grund von Beschränkungen der Datenfluss-Hardware sowie des Frameworks nicht möglich. Die Auslagerung des Cluster Findings in dedizierte Datenfluss-Hardware wäre nur mit deutlich erhöhten Kosten und Komplexität möglich gewesen. Ein vielversprechender Ansatz war jedoch die Integration des Datenfluss Cluster Finders in die bestehende HLT C-RORC Hardware-, Firmware- und Softwareumgebung

über die eigenständigen Netzlisten. Dies konnte exemplarisch gezeigt werden. Der Durchsatz der händischen Implementierungen und der mit der Datenfluss-Umgebung entwickelten Implementierungen ist konstruktionsbedingt identisch. Auch die maximal möglichen Taktraten sind ähnlich. Der Ressourcenverbrauch beider Varianten hat sich als beinahe identisch erwiesen. Der Quellcode-Umfang der Datenfluss-Implementierung ist auf Grund der höheren Abstrahierung deutlich geringer und damit leichter zu entwickeln, zu erweitern und zu warten.

Ein großer Vorteil des Datenfluss-Ansatzes ist die Möglichkeit, Erweiterungen und Änderungen am Algorithmus schnell und einfach zu evaluieren. In dieser Arbeit wurden beispielhaft die Auswirkungen auf den Ressourcenverbrauch durch die Variation der Fixed-Point Genauigkeit sowie das Verschieben von Verarbeitungsschritten zwischen Software und Hardware untersucht. Beides sind essentielle Schritte für die Planung einer Hardware-Implementierung, die mit der üblichen Register-Transfer Beschreibung der Hardware extrem zeitaufwändig sind. Auf der Datenfluss- / Algorithmusebene waren diese Evaluationen in kurzer Zeit möglich.

7 Ergebnisse, Schlüsse und Ausblick

Die Ziele der Arbeit waren die Entwicklung einer gemeinsamen Ausleselösung für die ALICE DAQ und HLT Systeme in RUN 2, die Integration und Betreuung dieser in der HLT Anwendung sowie die Untersuchung von Ansätzen zur Hardwarebeschreibung auf algorithmischer Ebene. Diese Ziele wurden erreicht und teilweise übertroffen. Es wurde eine Hardwareplattform entwickelt, die neben den eigentlichen Zielanwendungen in den ALICE DAQ und HLT Systemen außerdem im ATLAS Experiment, sowie zahlreichen weiteren Testsystemen und Experimenten ihre Anwendung gefunden hat. Die Parallelnutzung von Hardwareentwicklungen über LHC-Großexperimente hinweg war zu diesem Zeitpunkt ein Novum. Weiterhin ist die Karte für den Einsatz in RUN 3 vorgesehen. Die Karte wurde im Rahmen dieser Arbeit über den gesamten Produktlebenszyklus hinweg von der Konzeption an über Entwicklung, Koordination, Produktion, Test, Firmware- und Softwareentwicklung sowie Integration und Produktivbetrieb im HLT betreut. Sie war rechtzeitig für den Betrieb in den RUN 2 Produktivsystemen verfügbar, erfüllt alle Anforderungen und hatte bis zum Ende von RUN 2 keine technischen Ausfälle zu verzeichnen.

Die Firmware und Software der HLT Anwendung wurde für RUN 2 vorbereitet und über die gesamte Betriebsdauer hinweg kontinuierlich weiterentwickelt. Ein zentraler Aspekt in der HLT Anwendung ist die Datenvorverarbeitung mit dem Hardware Cluster Finder im Auslese-FPGA. Dieser Algorithmus konnte auf die neue Hardware portiert werden, wurde im Rahmen des TPC-Elektronikupdates auf den doppelten Durchsatz optimiert und hat mit einem verbesserten Algorithmus zu einer deutlich erhöhten HLT Datenkompressionsrate bei teilweise sogar verbesserten Physik-Parametern beigetragen. Im Vergleich zu äquivalenten Verarbeitungsschritten in Software, ist eine HLT FPGA-Auslesekarte mit Hardware Cluster Finder etwa einen Faktor 10 schneller als eine parallelisierte Softwareverarbeitung auf einem HLT Knoten.

Die Firmware-Entwicklungen um den Hardware Cluster Finder waren durchweg erfolgreich und mit sehr positiven Resultaten für das Gesamtexperiment verbunden, jedoch auch sehr aufwändig. Änderungen an der Auslesearchitektur sowie dem Algorithmus hatten die Neuentwicklung mehrerer zentraler Komponenten mit hohem Verifikationsaufwand erfordert. Die Beschreibung des Algorithmus mit dem Datenfluss-Ansatz hat sich hierbei als effiziente, kompakte und leicht wartbare Alternative gezeigt. Sowohl der Ressourcenverbrauch, als auch der Durchsatz waren vergleichbar mit einer händischen Implementierung. Beides konnte mit einem Bruchteil der Entwicklungszeit und des Code-Umfangs erreicht werden. Evaluationen von Verarbeitungsschritten in Hardware waren damit sehr einfach möglich. Für die Integration ins Gesamtsystem konnte die Datenfluss-Beschreibung in eine eigenes Firmware- und Softwareumgebung eingebunden werden. Dieser Ansatz verdeutlicht, wie konventionelle Firmware-Entwicklung und HLS-Ansätze zukünftig schrittweise zusammengebracht werden können, um die Vorteile der Hochsprachen mit bestehenden Bausteinen zu kombinieren. Am Beispiel des Hardware Cluster Finders konnte gezeigt werden, wie eine algorithmische Hardwarebeschreibung den Entwicklungsaufwand beträchtlich vereinfacht. Sie reduzierte die Entwicklungszeit sowie den Umfang der Beschreibung. Damit erleichterte sie außerdem die Wartung und Weiterentwicklung. Die Umsetzung des Algorithmus mit der Datenfluss-Beschreibung war dies in diesem Fall sogar ohne Einbußen bei Ressourcenverbrauch oder Taktrate möglich.

Literaturverzeichnis

- [Aa08] Aamodt, K. et al.: The ALICE experiment at the CERN LHC. JINST, 3:S08002, 2008.
- [Al10] Alme, J. et al.: The ALICE TPC, a large 3-dimensional tracking device with fast readout for ultra-high multiplicity events. NIM-A, 622(1):316–367, Oct 2010.
- [Al13] Alme, J et al.: RCU2 – The ALICE TPC readout electronics consolidation for Run2. JINST, 8(12):C12032, 2013.
- [EB08] Evans, Lyndon; Bryant, Philip: LHC Machine. JINST, 3(08):S08001, 2008.
- [En19] Engel, H.: Development of a Read-Out Receiver Card for Fast Processing of Detector Data – ALICE HLT Run 2 Readout Upgrade and Evaluation of Dataflow Hardware Description for High Energy Physics Readout Applications. Dissertation, Goethe-University Frankfurt a. M., 2019.
- [Ri07] Richter, M. et al.: High Level Trigger Applications for the ALICE Experiment. IEEE Transactions on Nuclear Science, 55(1):133–138, 2007.



Dr. Heiko Engel hat an der Ruprecht-Karls Universität in Heidelberg Physik studiert und das Studium mit Diplom abgeschlossen. Im Anschluss entwickelte und betreute er an der Johann Wolfgang Goethe-Universität in Frankfurt a. M. Auslesehardware für das ALICE Experiment am CERN. Im Kontext dieser Arbeiten schloss er 2019 am Lehrstuhl von Prof. Dr. Udo Kebschull seine Promotion in Informatik mit Auszeichnung ab. Seit Abschluss der Arbeiten am CERN arbeitet er an programmierbaren Hardwarelösungen für die Industrie.