7 Fazit

Ziel der Arbeit war es, durch die Implementierung eines digitalen Filters sowohl auf einem FPGA als auch auf einem Mikrocontroller, die beiden Systeme miteinander zu vergleichen. Hierfür wurde ein digitaler Filter, genauer gesagt ein FIR-Filter, umgesetzt. Damit die Gegenüberstellung unter möglichst gleichwertigen Rahmenbedingungen stattfindet, wurden durch eine Python Anwendung digitale Referenzdaten für beide Implementierungen erzeugt. Der Filterentwurf erfolgte ebenfalls über Python und dessen Filterkoeffizienten wurden auf beiden Systemen gleichermaßen eingebunden. Somit konnte eine solide Grundlage für die Untersuchung geschaffen werden.

Die Vergleichsanalyse enthält zum einen die Unterschiede in der Implementierung der Systeme und zum anderen die Messung der Latenz, des Durchsatzes, des Jitters sowie der Leistungsaufnahme. Als Hardware repräsentiert der STM32H747I-DISCO die Mikrocontroller, während als FPGA der ARTY-S7-25 gewählt wurde. Bei der Implementierung konnte als Hauptunterschied festgestellt werden, dass diese beim Mikrocontroller softwarebasiert erfolgt, während der Entwicklungsprozess des FPGAs eher hardwarebasiert abläuft. Aufgrund dessen ist die Umsetzung auf dem FPGA mit deutlich höherem Entwicklungsaufwand und fundiertem Wissen im Bereich digitaler Schaltungstechnik verbunden. Zudem müssen bei der Realisierung des FPGAs deutlich mehr Aspekte, unter anderem Timing-Anforderungen und Ressourcenverbrauch, beachtet werden. Der hohe Entwicklungsaufwand ist jedoch mit einer höheren Berechnungsleistung und geringerem Energieaufwand verbunden, wie durch die Messungen gezeigt wurde. Das liegt unter anderem daran, dass beim FPGA durch die Hardwareparallelisierung geringere Latenzen erreicht werden. Trotz einer um den Faktor 40 geringeren Taktfrequenz erreicht der FPGA im untersuchten Versuchsaufbau eine etwa 47-fach geringere Berechnungslatenz als der Mikrocontroller

Mikrocontroller bieten hingegen eine einfachere Entwicklung und höhere Flexibilität. Vor allem durch die STM32CubeIDE und der STM32CubeMX Software lassen sich Kommunikationsschnittellen und Funktionsblöcke des Mikrocontrollers leicht einbinden und ansteuern. Außerdem ist zu beachten, dass STMicroelectronics je nach Anwendungsgebiet passende Mikrocontroller anbietet. Hierbei gibt es ein Vielzahl an möglichen Mikrocontroller-Familien zur Auswahl. Für Anwendungen bei denen die Leistungsaufnahme eine höhere Priorität als ein maximaler Durchsatz besitzt, bietet STMicroelectronics darüber hinaus speziell auf einen geringen Energiebedarf ausgelegte Ultra-low-power Mikrocontroller an [42].

Für den hier betrachteten Anwendungsfall, ist der FPGA insbesondere hinsichtlich Latenz, Durchsatz und Energiebedarf von Vorteil. Digitale Filter werden oft im Bereich der Audio- und

Bildbearbeitung verwendet [27]. In diesen Fällen möchte man eine geringe Latenz zwischen dem Empfangen des Signals und der Ausgabe der gefilterten Werte. Dies ist insbesondere dann von Bedeutung, wenn die Verarbeitung in Echtzeit erfolgen soll. Als Beispielanwendung könnte ein solcher FIR-Filter zum Herausfiltern von Rauschsignalen eines Mikrofons verwendet werden. Die Filterberechnung sollte möglichst schnell erfolgen, damit keine unerwünschten Verzögerungen auftreten. Für 8000 Samples ergibt sich unter Berücksichtigung der gemessenen beziehungsweise geschätzten Leistungswerte zudem ein näherungsweise 36-fach geringerer Energiebedarf des FPGA, wobei der Leistungswert des FPGAs auf einer Schätzung des Vivado Power Reports basiert.

Der Mikrocontroller bietet dennoch wesentliche Vorteile hinsichtlich der Flexibilität und des Entwicklungsaufwands. Gerade aufgrund der einfachen Implementierung und der Vielzahl an Mikrocontroller-Familien werden diese in zahlreichen Anwendungen eingesetzt. Für Anwendungen mit moderaten Anforderungen an Latenz und Durchsatz stellt der Mikrocontroller daher eine geeignete Alternative dar.

Weiterführend könnte untersucht werden, wie sich unterschiedliche Filterordnungen und stärker optimierte Implementierungen auf die Ergebnisse auswirken. Insbesondere eine gepipelinte FIR-Architektur auf dem FPGA sowie eine weitgehende Optimierung der Mikrocontroller Implementierung wären hierfür interessante Ansätze.