

8 Kundennutzen, Fazit und Ausblick

8.1 Rückblick und Kundennutzen

Für das Projekt *deep* steht mit dem Zybo ein kostengünstiges Experimentierboard mit einem leistungsstarken Prozessor und FPGA zur Verfügung.

Das "*OpenOCDInterface*" kann mit dem Zybo gleich eingesetzt werden, wie das bestehende "*Abatron-Interface*" für PowerPCs. Damit kann die Entwicklung des *deep*-Kernel mit dem neuen ARM-Prozessor gleich weiter geführt werden wie bisher.

Die in dieser Arbeit entwickelte Toolchain ermöglicht es, ohne zusätzliche Hardware, ein von *deep* kompiliertes Programm auf den Zynq des Zybos zu laden.

Mit dieser Arbeit konnte auch aufgezeigt werden, dass der *gdb* grundsätzlich genutzt werden kann, um eine *deep*-Applikation direkt auf dem Prozessor zu debuggen. Dabei können auch *gdb*-Features wie Sourcecode-Lookup verwendet werden, obwohl der *gdb* Java nicht mehr direkt unterstützt.

Die Toolchain kann ebenfalls als Hardware-Debugger verwendet werden. Es ist möglich Prozessorregister und Speichersegmente zu lesen und zu schreiben und es können Hardware-Breakpoints gesetzt werden.

8.2 Ausblick

Die entwickelte Toolchain bietet eine gute Ausgangslage für die weitere Entwicklung von *deep*. Damit der Debugger aber effizient genutzt werden kann, fehlen noch zwei Kernelemente:

1. Die Debug-Informationen müssen automatisch beim kompilieren der *deep*-Applikation erzeugt werden. Dabei sollten nicht die veralteten STABS verwendet werden, wie in dieser Arbeit, sondern das modernere DWARF-Format.
2. Ein *gdb*-Plugin für *Eclipse* muss noch entwickelt werden. Das "*GNU MCU Eclipse plug-ins for ARM & RISC-V C/C++ developers*"¹ ist ein Plugin für Eclipse, dass genau dieses Problem löst. Allerdings muss es noch so angepasst werden, dass es in die OpenOCD-Toolchain von dieser Arbeit integrierte werden kann.

¹<https://github.com/gnu-mcu-eclipse/eclipse-plugins>