Daniel Werner

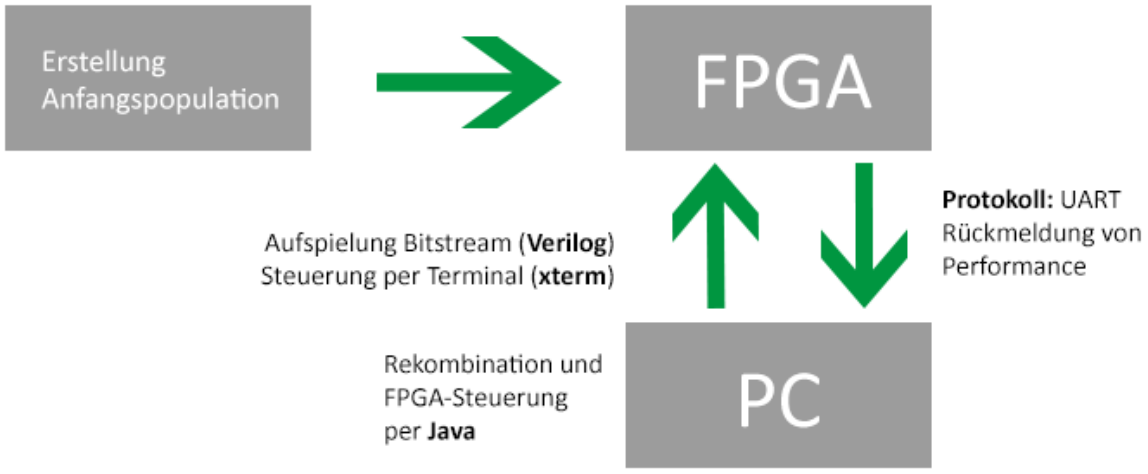
Projekt „Evolvable Hardware“

Projektziel

Das Ziel des Projekts war es, mithilfe von evolutionären Algorithmen und neuronalen Netzen vollautomatisch einen FPGA-Bauplan anhand von vorgegebenen Trainingsdaten zu erstellen.

Zum Schluss sollte es dann mithilfe dieser Technik möglich sein, einen FPGA basierten Bitcoin-Miner zu entwickeln, der wesentlich leistungsfähiger ist, als aktuelle auf dem Markt erhältliche Miner. Die Projektdauer betrug 3 Monate.

Ablauf



Zuerst wird zufällig eine Anzahl x an FPGA-Bitstreams erstellt, die die Anfangspopulation darstellt. Danach wird jeder Bitstream einzeln auf seine Performance hin getestet und anschließend diejenigen, welche am besten abgeschnitten haben, untereinander rekombiniert. Die Rekombination wird parallel von einem neuronalen Netz (RNN) überwacht, das mit der Zeit eine Regel hinter der jeweiligen Rekombination erkennen soll, um so den evolutionären Prozess enorm zu beschleunigen. Entscheidend ist, die benötigte Zeit pro Iteration so gering wie möglich zu halten und zusätzlich eine maximale

Verbesserung zur Vorgängergeneration zu erreichen, um auch bei komplexen Problemen nach einer vertretbaren Zeit zu einem brauchbaren Ergebnis kommen zu können.

Aufbau des Bitcoin-Miner

Um ein Bitcoin zu „schürfen“, muss man einen „Block-Header“ mit einer Länge von 640 Bit finden, der doppelt SHA256-gehasht unter dem aktuellen Schwellwert im Netzwerk liegt. Normale Miner sind deswegen gezwungen, zufällig einen entsprechenden Block-Header unter diesem Wert zu suchen, was nur mit einer möglichst hohen Anzahl an probierten Headern pro Sekunde möglich ist.

Der mithilfe von evolutionären Algorithmen erstellte FPGA-Miner hingegen geht den umgekehrten Weg, er wird mit zufällig generierten Hashs unter dem Schwellwert gespeist und versucht, den passenden Block-Header dahinter zu „minen“. Passt also ein gefundener Block-Header zum Eingangshash ist automatisch ein gültiger Block gefunden worden. Dies ist jedoch nur möglich, da der FPGA-Miner hunderttausende Hashs pro Sekunde prüft und dadurch auch eine sehr geringe Erkennungsrate (0,00000001%) ausreichend ist.