

Günstiger Einstieg in die FPGA Programmierung

Als Field Application Engineer (FAE) beim dem weltweit größten Bauteiledistributor ARROW empfehle ich meistens Microcontroller für alle Arten von Steuerungen. Es gibt aber auch Fälle, bei denen FPGA's besser die Anforderung für schnellen Datentransfer und Auswertung abdecken oder - gerade für FORTH-Programmierer interessant - auch wieder ältere Prozessorarchitekturen bzw. FORTH-Prozessoren zum Leben erwecken können. Für den schnellen Einstieg haben Arrow zusammen mit Trezz Electronic diverse günstige FPGA-Boards entwickelt, die ich hier vorstellen will.

Vorgeschichte

Entwicklungsboards für FPGA's gibt es viele, weil die Hersteller selbst schon Hardware für ihre Chips und Software brauchen. Jedoch haben diese oft den Nachteil des höheren Preises und dienen nur der Evaluierung, was industriellen Temperaturbereich und einer Garantie für den Serieneinsatz ausschließt. In einigen Fällen ist auch kein Debugger bzw. Programmer integriert und muss teuer dazugekauft werden. In der FPGA Community gibt es auch viele Boards, die aber meist für bestimmte Anwendungen gedacht sind und deshalb bezüglich Speicher- und Schnittstellenausstattung limitiert sind. Auch diese dürfen normalerweise nicht für industrielle Produkte eingesetzt werden.

Deshalb hat sich das Arrow Engineering Team zusammen mit Trezz überlegt, was man benötigt, um einen günstigen Einstieg in die FPGA-Programmierung zu beginnen und danach auch dieses Tool für Projekte in kleinen Serien zu nutzen. Da Arrow sowohl Altera (jetzt Intel), als auch Lattice und Microsemi (jetzt Microchip) FPGA's vertreiben, sollte daraus eine Familie von Boards werden.

Gewünschte Features waren:

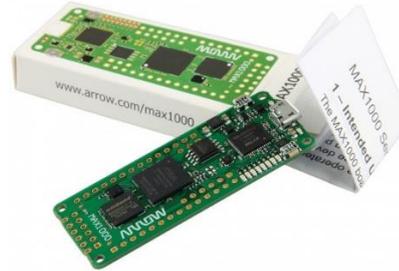
1. Günstiger Preis
2. Integrierter Debugger mit serieller Schnittstelle
3. Kleiner Formfaktor (Nano Arduino)
4. Tasten und LED's zur Bedienung und Statusanzeige
5. Zusätzlicher Programmspeicher
6. Zusätzliches RAM
7. 16-Pin-Pmod¹-Konnektor

Innerhalb der letzten zwei Jahre wurden 4 Boards realisiert, welche bei Preisen unter 50€ die gewünschten Features realisiert haben. Da alle Hersteller für die kleineren FPGA-Derivate kostenlose Software und auch entsprechende IP² für die Schnittstellen und sogar Prozessoren (NIOS II, LatticeMICO8/32 und jetzt auch RISC V) zur Verfügung stellen, ist dies alles was man investieren muss.

Diese Boards werden über den MicroUSB Konnektor versorgt und programmiert. In dem damit verbundenen - auch separat verfügbaren - Arrow USB Programmer 2 ist auch eine USB-UART-Bridge integriert und damit immer eine serielle Schnittstelle für die Prozessor IP verfügbar. Das Board kann auch über zwei Taster bedient werden und haben 8 LEDs als Anzeige. Die zusätzlich bestückbaren Nano Arduino- und Pmod-Konnektoren erlauben den Anschluss externer Hardware.

MAX1000 (TEI0001)

<https://www.arrow.de/products/max1000/arrow-development-tools>
<https://shop.trenz-electronic.de/de/Produkte/Trenz-Electronic/MAX1000-Intel-MAX10/>



- Intel MAX10 mit 8000 LE³ (integrierter Flash und A/D-Wandler)
- Arrow USB Programmer 2 mit USB-UART-Bridge
- 8Mbyte QuadSPI
- 8Mbyte SDRAM
- 2 Buttons & 8 LED's
- 3-Achsen Beschleunigungsaufnehmer

Für ca. 26€ (bei Arrow aktuell mit kostenlosem Versand) erhalten Sie die 8kLE-Version. Aber auch größere Versionen z.B. mit 16 bzw. 32kLE und 32MByte SDRAM sind verfügbar. Die MAX10-Serie hat ein integriertes Flash und einen 1MHz 12-Bit A/D-Wandler. Der Chip ist deshalb sofort nach dem Start ohne zusätzlichen Download aktiv, kann aber auch das externe SDRAM als Programmspeicher nutzen und dafür auch große Programme aus dem hier verfügbaren QuadSPI laden. Über die Trezz Webseite sind alle Informationen wie User Guide, Schematics und Beispielprogramme, meist für den auf MAX10 üblichen NIOS II-Prozessor, verfügbar.

CYC1000 (TEI0003)

<https://www.arrow.de/products/cyc1000/arrow-development-tools>
<https://shop.trenz-electronic.de/de/Produkte/Trenz-Electronic/CYC1000-Intel-Cyclone-10/>



- Intel Cyclone 10 LP (25kLE)
- Arrow USB Programmer 2 mit USB-UART-Bridge
- 2Mbyte Flash
- 8Mbyte SDRAM
- 2 Buttons & 8 LED's
- 3-Achsen Beschleunigungsaufnehmer

Ca. 35€ muss man investieren, wenn man die neuere Cyclone 10 LP Familie testen will. Da mehr LUT's verfügbar, können auch größere IP's geladen werden. Ansonsten ist das Board ähnlich dem MAX1000.

Günstiger Einstieg in die FPGA Programmierung

SMF2000 (TEM0001)

<https://www.arrow.de/products/smf2000/trenz-electronic-gmbh>
<https://shop.trenz-electronic.de/de/Produkte/Trenz-Electronic/SMF2000-Microsemi-SF2/>



- Microchip SmartFusion2 (integrierter Flash, Cortex-M3 und 12kLE)
- Arrow USB Programmer 2 mit USB-UART-Bridge
- 8Mbyte QuadSPI
- 8Mbyte SDRAM
- 2 Buttons
- 8 LED's

Mit 42€ schon etwas teuer ist der auf Microsemi (jetzt Microchip) SmartFusion2 basierenden SMF2000. Dafür erhält man eine schon im Chip eine integrierten 166MHz Cortex-M3 Hard-IP, 256 KByte Flash, 80 KByte SRAM sowie einen 12kLUT FPGA-Teil. Dies wird dann extern um 8MByte QuadSPI Flash, 8MByte SDRAM und den üblichen Tasten und LED's ergänzt. Dadurch könnte man eine eigene Processor-IP im FPGA durch den Cortex-M3 bedienen und Features wie Programm-Download, Debugging und Singlestep relativ einfach realisieren. Auch hier sind wieder alle Dokumente und Beispielprogramme auf der Trenz Webseite zu finden.

LXO2000 (TEL0001)

<https://www.arrow.de/products/lxo2000/trenz-electronic-gmbh>
<https://shop.trenz-electronic.de/de/Produkte/Trenz-Electronic/LXO2000-Lattice-XO2-4000/>



- Lattice MachXO2 (4kLE)
- Arrow USB Programmer 2 mit USB-UART-Bridge

Eigentlich nur zur Vollständigkeit hier das vierte Board basierend auf einem Lattice MACHXO2. Mit nur 4kLE ist er fast zu klein für eine Prozessor-IP, aber selbst damit kann man 16-Befehle-Cores oder Sequenzer programmieren. Dafür kann man auch die bis über 200kBit internes RAM verteilt auf den Chip nutzen.

Prozessor-IP's

Hier bewege ich mich auf Glatteis, da die Programmierung von FPGA's nicht meine Kernkompetenz ist und ich deshalb wenig Erfahrung mit den diversen IP's habe. Trotzdem hier einige Hinweise auf verfügbare Cores.

Ich denke, dass OpenCores (<http://opencores.org>) die

wichtigste Quelle für freie IP sind. Unter anderem werden dort ca. 200 Prozessoren wie 6502, 6809, 68HC11, Z80, 8051 oder sogar ein 32-Bit FORTH Prozessor mit Java Compiler gelistet.

Es gibt eine lange Liste von FORTH-Cores für FPGA. Hier möchte ich nur die vermutlich bekanntesten erwähnen:

1. The J1 (H2) Forth CPU (siehe Gameduino)
2. C. H. Ting's EP16/EP32
3. Don Golding's FP1
4. Richard E. Haskell's FP16 Forth Core
5. Bernd Paysan's B16
6. MPE's RTXcore (RTX-2000 für FPGA)
7. Klaus Scheisik's MicroCore

Da man mit den meisten Entwicklungsumgebungen sowohl VHDL als auch Verilog verwenden kann, gibt es bei allen FORTH-Cores nur die Limitierungen durch den FPGA oder der angehängten externen Speicher. Viele der Implementierungen sind auf Xilinx FPGA Boards getestet, aber vermutlich leicht auf Intel-, Lattice oder Microchip-FPGA übertragbar. Da man kaum um die Entwicklungsumgebung der Hersteller herumkommt, muss man sich zuerst einige 100MByte bis 5GByte aus dem Internet laden und installieren. Für die Einarbeitung helfen natürlich die vielen Einführungen beim Hersteller im Web. Für die erwähnten Boards gibt es jeweils Beispiele, welche die Schnittstelle initialisieren oder sogar schon eine Prozessor-IP implementieren.

Schlusswort

Ich weiß, dass dieser Artikel sich eher nach Werbung für unsere Produkte anhört. Aber ich hoffe doch, dass es eine Anregung ist, um mit möglichst kleiner Investition die alten FORTH-Microcontroller wie 6502, 6809, 68HC11 oder Z80 wieder zum Leben zu erwecken oder selbst ein FORTH- Prozessoren zu implementieren. Die 8 bis 25kLE des MAX10, Cyclone 10 LP oder SmartFusion2 sind dafür ausreichend und bieten wegen des zusätzlichen externen Flash's und dem SDRAM genügend Speicher für die Emulation der meisten Systeme. Über die vielen Pins sind dazu auch Anbindungen an (PS2-)Tastatur und Displays möglich, falls eine UART über die Debug-Schnittstelle nicht ausreicht. Leider kann ich Ihnen die Verwendung der Hersteller-Tools nicht abnehmen, da es sonst nur wenige freie FPGA-Entwicklungsumgebungen gibt und die entsprechenden Compiler am besten auf die Chips angepasst sind.

Ich würde mich über Interesse freuen und wie auch bei „meinen“ Microcontroller soweit möglich auch diese Tools für Projekte zur Verfügung stellen. Dazu bitte ein Email an klaus.kohl-schoepe@arrow.com möglichst mit Informationen über die geplante Anwendung.

- Ende

¹ Pmod: 6-Pin oder hier 12-Pin Konnektor definiert von Digilent für Erweiterungen wie I/O, SPI, UART ...

² IP = Intellectual Property: vorgefertigter Funktionsblock eines Chipdesigns

³ LE = Logic Elements: Logik-Block meist mit einer (4-Bit-)Lookup-Tabelle (LUT) und einem D-FlipFlop (1k LE sind 1000 LE)