

Willkommen

zur Präsentation

Modulare Grafische Programmierung (MGP) von FPGAs

Dr.-Ing. J. Pospiech

AVT GmbH Ilmenau

Am Hammergrund 1

98693 Ilmenau

Tel: +49 (0)3677 / 64 79 0

Fax: +49 (0)3677 / 64 79 69

Email: info@avt-ilmenau.de

Web: www.avt-ilmenau.de



Themen

- 1. Einleitung**
- 2. Programmierung von FPGAs**
- 3. Neues Konzept der FPGA Programmierung**
6 Schritte der Modularen Grafischen Programmierung (MGP)
- 4. Zusammenfassung**

Einleitung

FPGAs (Field Programmable Gate Arrays = frei programmierbare Schaltkreise)

Die wichtigsten **Vorteile von FPGAs** sind:

- Sehr schnell für rechenintensive Anwendungen durch parallele Berechnungen
- kleine Abmessungen, geringer Energiebedarf und geringe thermische Beanspruchung
- flexibel durch beliebig oft programmierbare Logikblöcke
- sehr viele frei definierbare Ein- und Ausgänge (mehr als 1100)
- keine bzw. wenige Redesigns der Leiterplatten notwendig
- schnelle Entwicklung von Designs mit komfortablen Software-Werkzeugen verkürzen "time-to-market"
- Integration vieler zusätzlicher Komponenten in den FPGA, z.B. CPU, I/O Steuerung, RAM Steuerung, Filter,...
- Einbindung von Soft-IP-Cores als Expertenwissen in die Anwendung



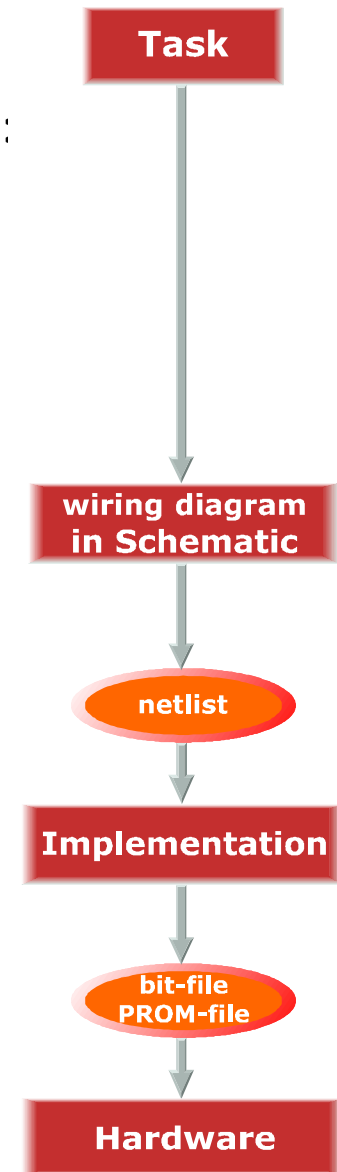
Programmierung von FPGAs

Wesentliche Programmiermethoden von FPGAs:

1. Frühere FPGA Programmiermethoden

basierend auf Netzlisten logischer Elemente und Systemgatter

- Produziert riesige Netzlisten für große FPGAs
- ECS/Schematics ist eins der grafischen Programmierertools für den Design Entry
- Das folgende Implementierungstool besteht aus Translate, Mapping, Place & Route und Bitfile-Generation



Programmierung von FPGAs

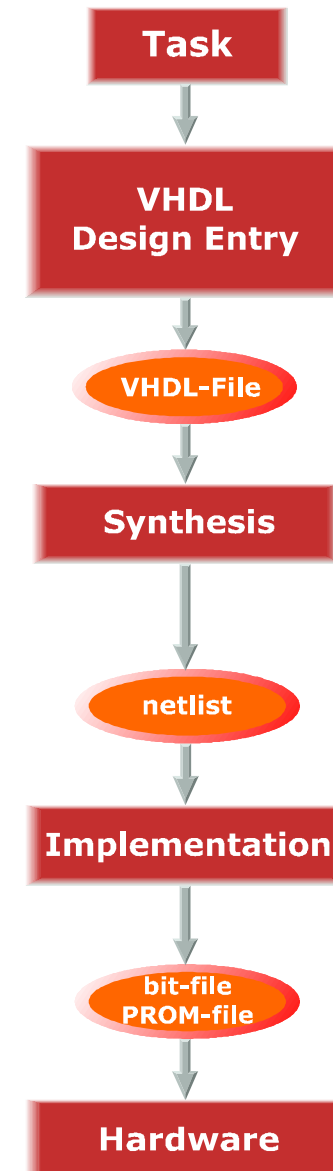
2. FPGA-Programmierung mit VHDL/Verilog als Design Entry

Vorteile:

- leichte Beschreibung und Programmierung von komplexen Funktionen
- Programmierung von großen FPGAs möglich

Nachteile:

- kommandozeilenorientiert, schwierige Dokumentation für einen guten Überblick
- gute Kenntnisse der Beschreibungssprache notwendig, langwierige Einarbeitung von Applikationsingenieuren



Programmierung von FPGAs

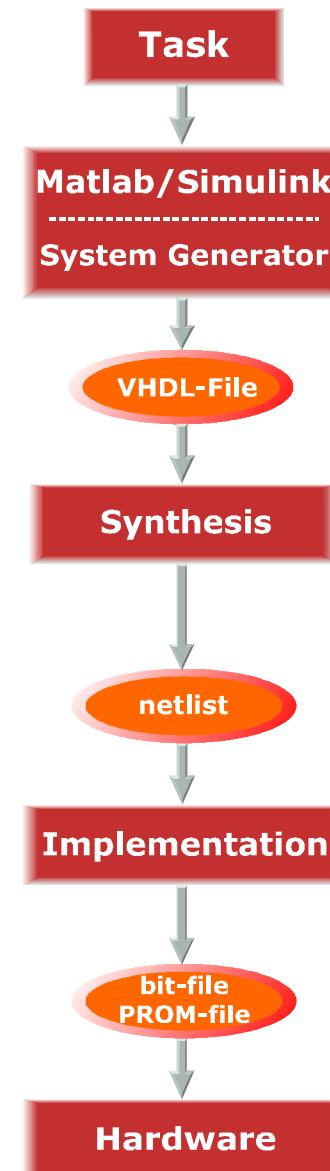
3. FPGA Programmierung mit grafischer Programmieroberfläche auf hohem Niveau

Vorteile:

- leichte grafische Programmierung auf hohem Niveau
- guter Überblick und Dokumentation des Designs sind garantiert
- bit- und zeitgenaue Simulation
- Programmierung großer FPGAs möglich
- automatische Generierung von HDL und Netzlisten

Nachteile:

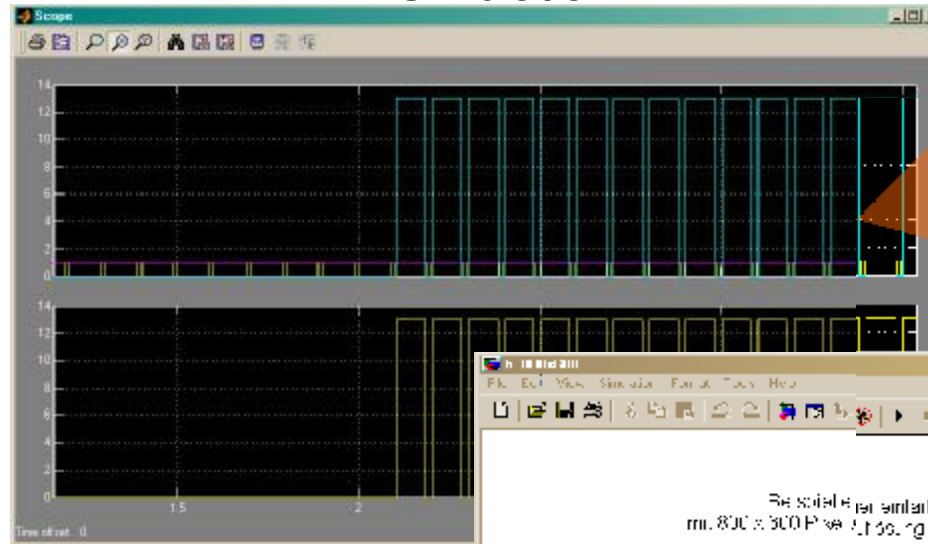
- Kenntnisse grundlegender HDL syntax, sowie Matlab Scriptsprache und Simulink blocksets notwendig, langwierige Einarbeitung von Applikationsingenieuren
- kostenintensive Software



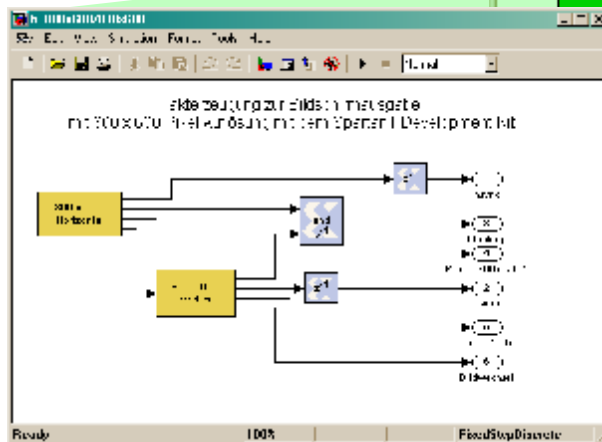
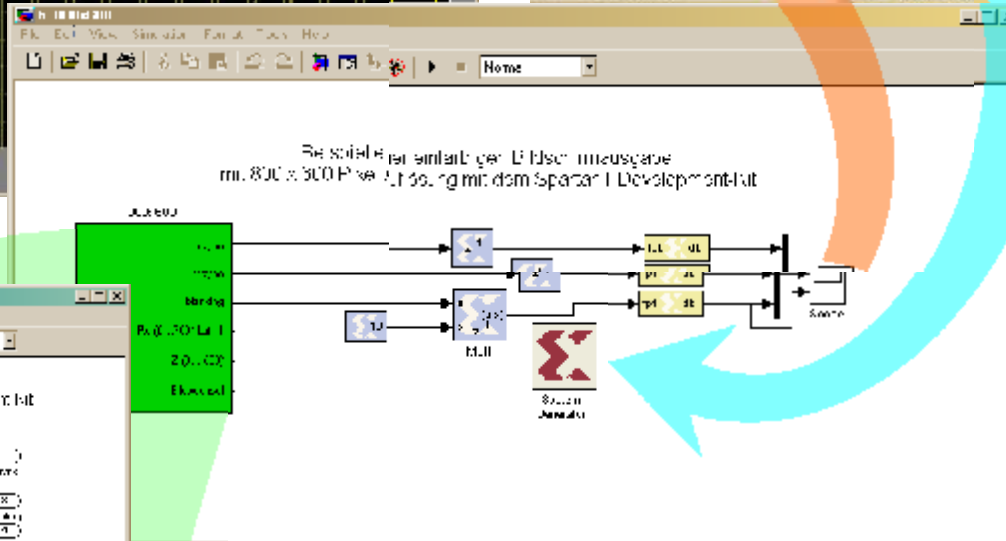
Programmierung von FPGAs

Simulation

XILINX-Library



Simulink Library Browser window showing the Xilinx Library. The left pane lists categories: Continuous, Discrete, Functions & Tables, Math, Nonlinear, Signals & Systems, Sinks, Sources. The right pane shows a search for 'System Generator' and lists components like 'System Generator', 'Addressable Shift Register', 'Black Box', and 'Control'.



Simulink

Simulink library browser showing the DSP and Math sections. The DSP section includes 'DSP', 'Math', 'Matlab IO', 'Memory', and 'State Machine'. The Math section includes 'Data Valid', 'Set Valid Bit', 'Slice', and 'Sync'.

Programmierung von FPGAs

Probleme aktueller Programmiermethoden, besonders für Neueinsteiger:

- hohe Kosten der komfortablen Programmiersoftware
- hohe Kosten durch Einarbeitungsphasen der Applikationsingenieure

Zusätzliche Hardwareprobleme für Neueinsteiger:

- unbekannte Technologie des FPGA-Designs im PCB Layout

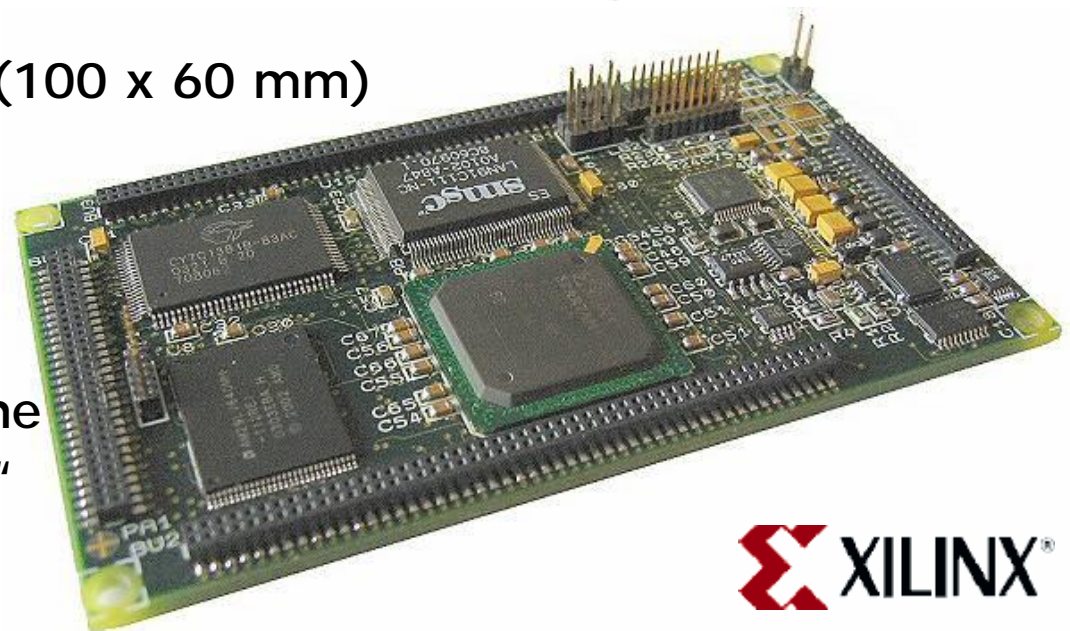
Diese Probleme stellen insbesondere für Universitäten, sowie kleine und mittlere Unternehmen hohe Hürden für die Nutzung dieser Technologie dar.

Die Lösung der Probleme

Die Hardwareprobleme können durch die Nutzung kleiner und universeller FPGA Development Kits gelöst werden.

Beispiel eines Development Kits, dass auch als Endprodukt für kleine und mittlere Serien genutzt werden kann:

- Development Kit = Endprodukt
- Platzsparendes Design (100 x 60 mm)
- Gebrauch passender FPGAs von XILINX
- umfangreiche zusätzliche Bauelemente „onboard“



XILINX®

Modulare Grafische Programmierung (MGP)

Applikationsprogrammierung mit XILINX ISE und AVT Toolbox

- sehr kurze Einarbeitungszeit
- grafische Programmierung mit komplexen funktionalen Blöcken (Cores)
- freie Software für Implementierung und grafischen Design Entry
- Funktionserweiterung mit Grundelementen, IP-Cores oder HDL

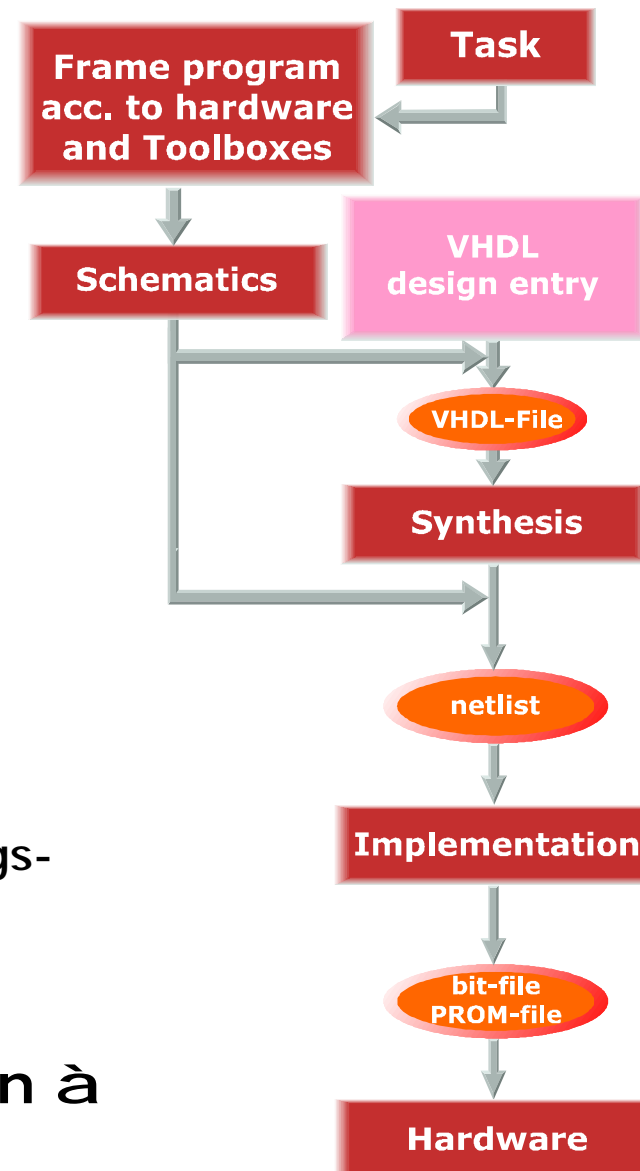


Modulare Grafische Programmierung (MGP)

Design flow von MGP:

- Applikationsingenieur muss die Aufgabe und die Realisationsidee haben
- Applikationsingenieur benutzt grafisches Rahmenprogramm entsprechend der Hardware
- Hinzufügen der für die Applikation notwendigen Funktionen als grafische Module der Toolbox
- Generierung der FPGA-Programmierungsdatei mit dem Implementierungstools

MGP besteht aus 6 Schritten à



Modulare Grafische Programmierung (MGP)

1. Schritt: Berechnung der Design-Ressourcen

- Wahl der notwendigen Funktionen entsprechend der Anwendung
- Eingabe der Bitbreite und Anzahl der Funktionen in einer Excel-Tabelle
- Ausgabewerte sind die Summe der notwendigen Design-Ressourcen (Slices und BlockRAM) im Vergleich mit den Ressourcen des ausgewählten FPGA

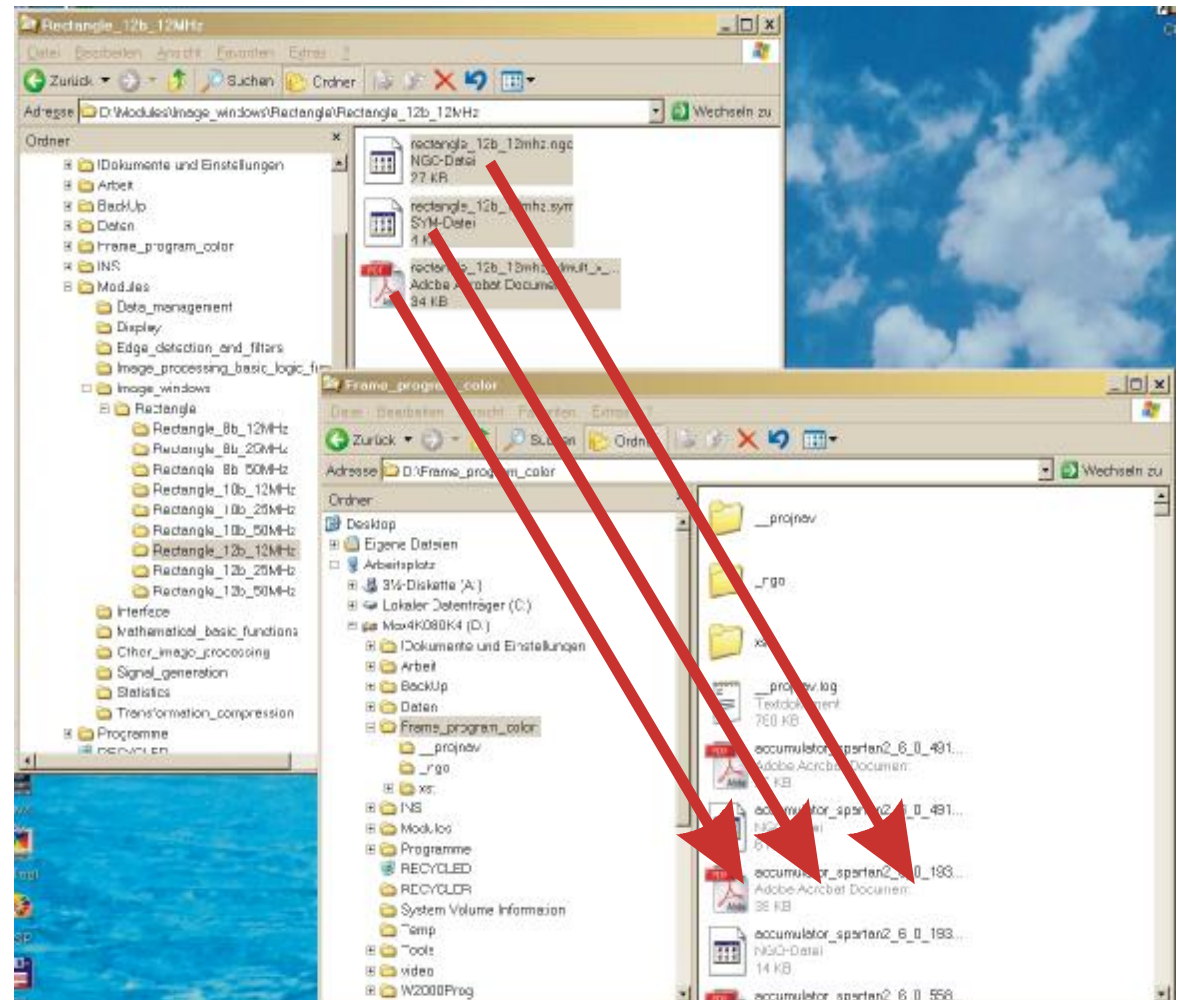
	A	B	C	D	E	F	G	H	I	J	K
<	Functions in XILINX-FPGAs										
2											
3											
4											
5	Mathematical basic functions										
6	Divisor										
7	Acc										
8	Bit										
9	Yot										
10	And										
11	Block RAM (Kbit)										
12	in 11										
13	in 12										
14	in 13										
15	Signal generation										
17	in 14										
18	coloc_bsr01_2MHz										
19	coloc_bsr01_25MHz										
20	coloc_bsr01_100MHz										
21	coloc_bsr02_100MHz										
22	coloc_bsr02_50MHz										
23	coloc_bsr02_50MHz										
24	edge detect on bank 0 (ring)										
25	edge detect on bank 0 (asynchr)										
26	edge detect on bank synchron										
27	coloc_bsr03_10MHz										
28	coloc_bsr03_10MHz										
29	coloc_bsr04										
30	coloc_bsr04_invert										
31	coloc_bsr04_invert										
32	coloc_bsr04_invert										
33	Image windows										
34	rectangle_2bit										
35	rectangle_2bit										
36	rectangle_2bit										
38											
39	Edge detection and filters										
40											
41	in 15										
42	coloc_bsr05_10MHz										
43	coloc_bsr05_10MHz										
44	coloc_bsr05_10MHz										
45	coloc_bsr05_10MHz										
46	Difference (0; 0; 0) (0; 0; 0) (0; 0; 0)										
47	coloc_bsr06_10MHz										
48	coloc_bsr06_10MHz										

Input **Output**

Modulare Grafische Programmierung (MGP)

2. Schritt: Kopieren der notwendigen Modul-Dateien der Funktionen

- Kopieren der Modul-Dateien vom Toolbox-Verzeichnis in das Arbeitsverzeichnis
- Modul-Dateien sind Netzlisten und graphische Symbole von Funktionen
- ergänzende Report-Dateien bestehen aus Platz- und Geschwindigkeitsinformationen

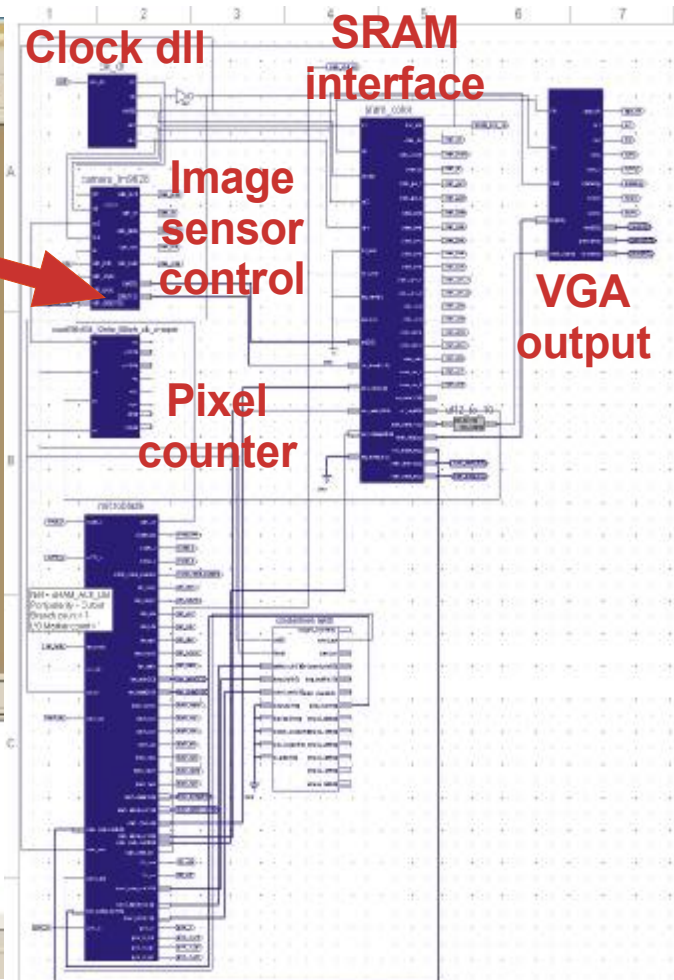
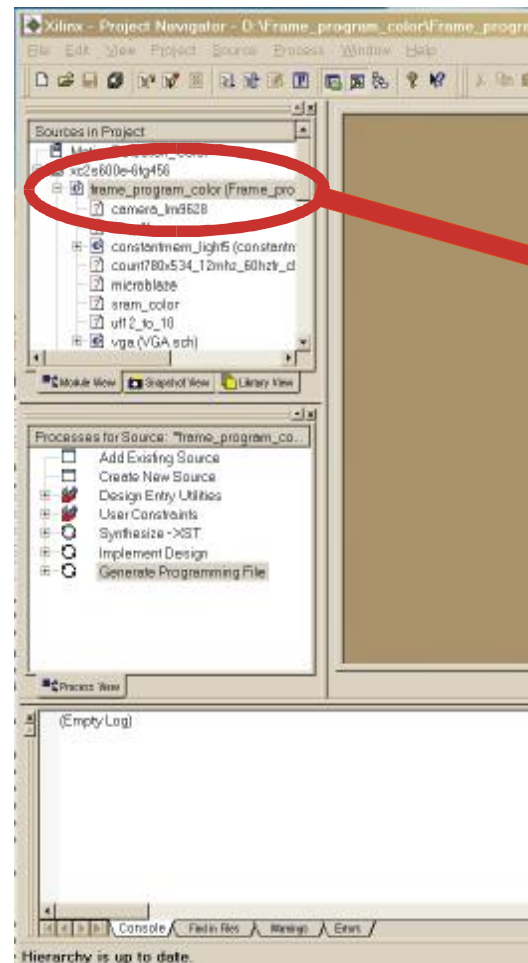


Modulare Grafische Programmierung (MGP)

3. Schritt:

Start des Rahmenprogramms innerhalb des Implementierungstools

- Xilinx ISE Project Navigator zeigt die Teile des Rahmenprogramms und die Implementierungsschritte
- Öffnen des Rahmenprogramms in ECS/ Schematics

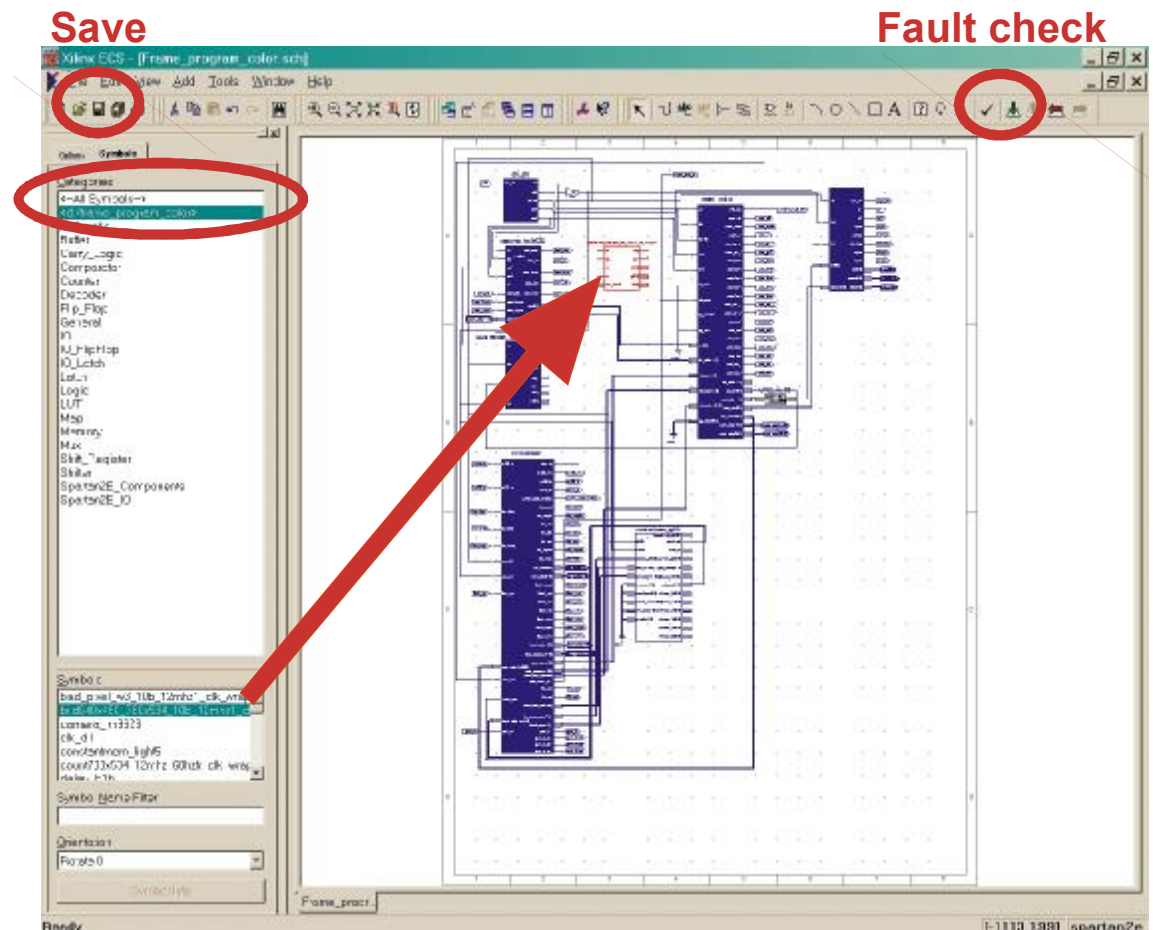


Modulare Grafische Programmierung (MGP)

4. Schritt:

Einfügen der Modulsymbole zu einer Applikation

- Symbole der Modulfunktionen befinden sich im Arbeitsverzeichnis in ECS/Schematic. Nach der Wahl der Module, werden diese ins Rahmenprogramm eingefügt
- Datenleitungen zwischen Modulen des Rahmenprogramms können getrennt und wieder mit Modulen der vorgefertigten Funktionen verbunden werden
- Fehlerüberprüfung und Speichern

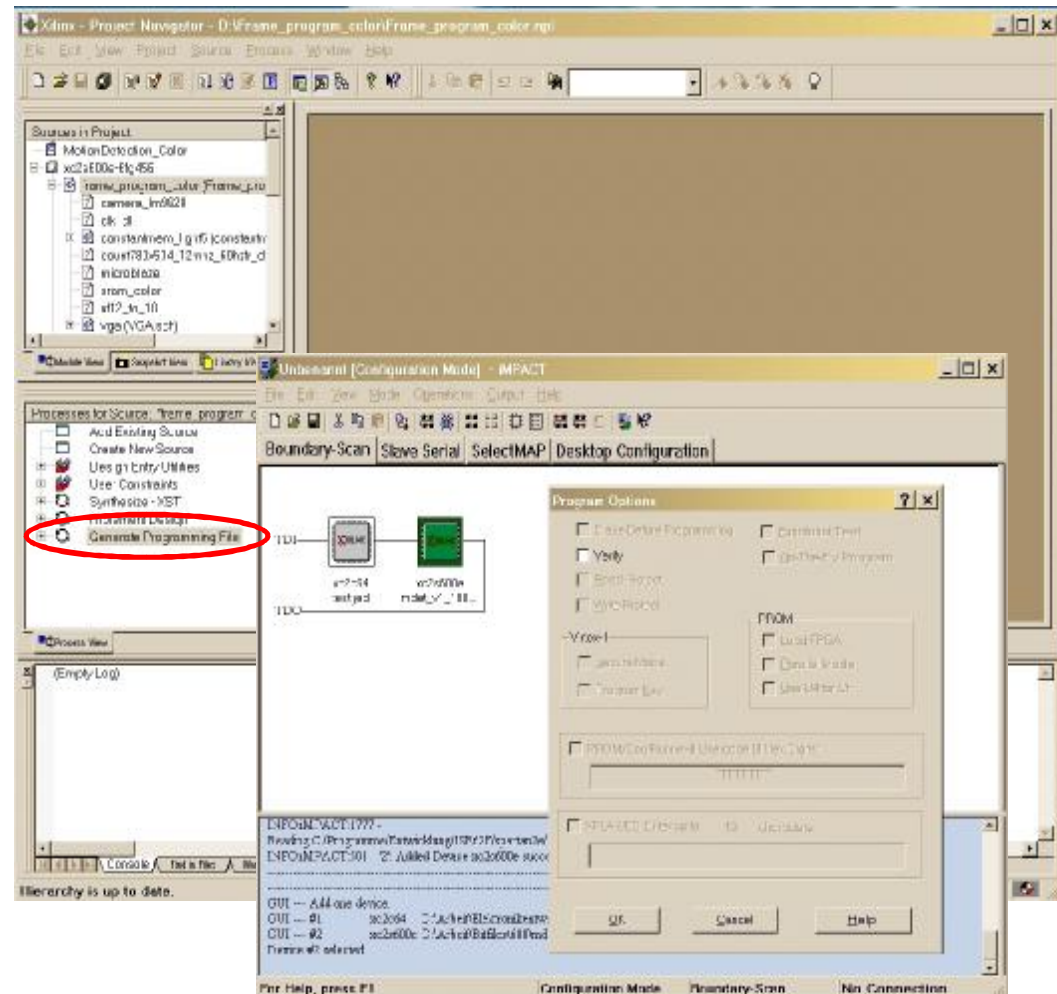


Modulare Grafische Programmierung (MGP)

5. und 6. Schritt: Generieren der Programmdatei und Hardwareprogrammierung

- Automatische Generierung der Programmierdatei durch Doppelklick auf den letzten Implementierungsschritt; alle Schritte laufen automatisch
- Das Programm "IMPACT" (Teil der Xilinx ISE) lädt das generierte bit-file in den FPGA

Wie sieht so eine Applikation aus ? à



Modulare Grafische Programmierung (MGP)

- Die MGP-Toolbox besteht aus über 800 Modulen, hauptsächlich für Videobearbeitung, digitale Signalverarbeitung und Kommunikation
- Modul-Funktionen sind in verschiedenen Kategorien angeordnet (mathematische Grundfunktionen, Signalgenerierung, Flankenerkennung und Filter, Transformation und Kompression, Statistik, Displayfunktionen, Interfacekontrolle Bildverarbeitung und Datenmanagement)
- Module beinhalten zudem sehr komplexe Funktionen wie JPEG-Komprimierung, Nutzung des XILINX MicroBlaze 32 bit RISC Prozessors im FPGA, sowie Hard- und Software Module, die an den MicroBlaze Prozessor angeschlossen sind

Zusammenfassung

- Modulare Grafische Programmierung (MGP) ist geeignet für den leichten, kostengünstigen und schnellen *Einstieg* in die FPGA-Programmierung und/oder für eine schnelle Design-Erweiterung von bestehenden Designs mit Funktionen als Cores
- MGP ist leicht zu benutzen und schnell zu lernen → spart Kosten
- MGP macht die schnelle Realisierung von Anwendungen möglich → spart Kosten
- MGP ist nahtlos in das Implementierungstool integriert, damit ist keine spezielle Programmiersoftware notwendig → spart Kosten
- MGP ist durch eigene Funktionen erweiterbar → spart Kosten
- MGP macht verschiedene Programmierungen möglich: Schematic Module, VHDL Module, Module von Core/System Generator und IP Cores)
- Nach *Beginn* der Arbeit mit FPGAs wird MGP die Kenntnisse über FPGA-Programmierung systematisch erweitern

Unser Angebot

- Viele Module von MGP sind kostenfrei in Verbindung mit dem Development Kit der AVT GmbH
- Größere und komplexere Funktionen werden kostengünstig angeboten
- Eine Programmierung neuer Versionen und mehr Funktionen befinden sich in ständiger Entwicklung (auch auf Kundenwunsch)
- Einstieg in FPGA Kenntnisse und MGP-Programmiermethoden werden unterstützt durch modulare Seminare
- MGP ist weiterhin ideal anwendbar für System-on-Chip-Designs auf FPGAs mit Hardware- und Softwarebibliotheken für MicroBlaze-Prozessor; auch dazu bieten wir unsere Kenntnisse in Seminaren an.

Unser Angebot

Vielen Dank für Ihre Aufmerksamkeit !

**Für weitere Informationen stehe ich
Ihnen gerne zur Verfügung**

Kontaktinformationen

AVT GmbH Ilmenau

Automatisierungs- und Verfahrenstechnik

Am Hammergrund 1

98693 Ilmenau



Sie erreichen uns unter:

Tel: +49 (0) 36 77 / 64 79 0

Fax: +49 (0) 36 77 / 64 79 69

Email: info@avt-ilmenau.de

Web: www.avt-ilmenau.de



Ansprechpartner:

Dr.-Ing. Jörg Pospiech

Geschäftsführer, Entwicklungsleiter

Herr Wilfried Geyer

Konstruktion