Welcome

to the presentation

Modular Graphical Programming (MGP) of 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

AVT ILMENAU

XPERTS
Xilinx Program for Engineering Resources from Third Parties

Topics

1. Introduction
2. Programming of FPGAs
3. New concept of FPGA programming
   - 6 steps of Modular Graphical Programming (MGP)
4. Conclusion

Introduction

FPGAs (Field Programmable Gate Arrays = free programmable hardware circuits)

The most important **advantages** of the use **of FPGAs** are:

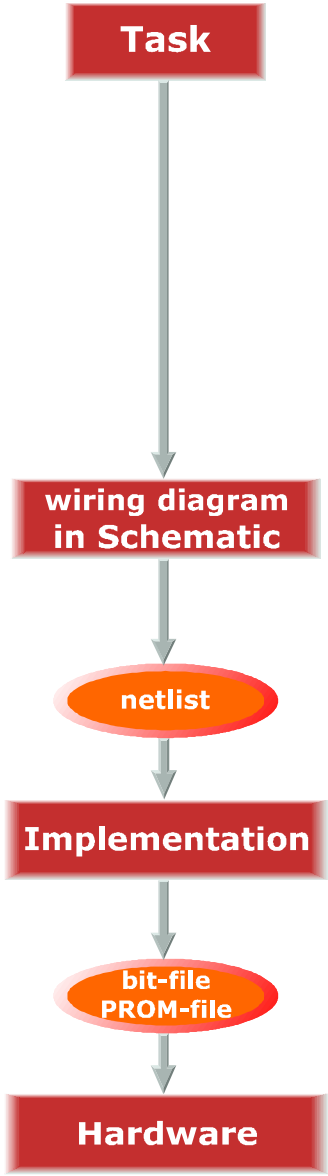
- Extremely fast for intensive calculations by the use of parallel processing
- Small dimensions, small energy consumption and low thermal load
- Flexible by the use of reprogrammable logic blocks
- A lot of free definable inputs and outputs (more than 1100)
- No or few redesigns of printed circuit boards necessary
- Fast development of designs with comfortable software tools shortens time-to-market
- Many additional components can be integrated in the FPGA, e.g. CPU, I/O control, RAM control, filters, ...
- Application of completed Soft-IP-Cores for implementation of expert knowledge

Programming of FPGAs

Essential programming methods of FPGAs:

1. **<u>Former FPGA programming methods</u> based on Netlists of logic elements and system gates**

   - **produce giant netlists for big FPGAs**
   - **ECS/Schematics is one of the graphical programming tool of design entry**
   - **following implementation tool consist of translate, mapping, place & route and bit-file generation**

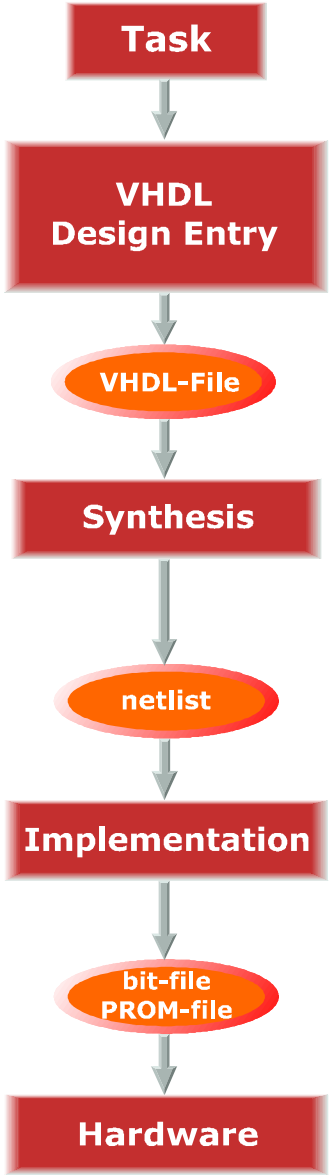


Programming of FPGAs

2. FPGA programming with VHDL/Verilog
Design entry

Advantages:
- easy description and programming of complex functions
- programming of big FPGAs possible

Disadvantages:
- command-line oriented, difficult documentation for good overview
- good knowledge of description language necessary, long period of vocational adjustment of application engineers



Programming of FPGAs

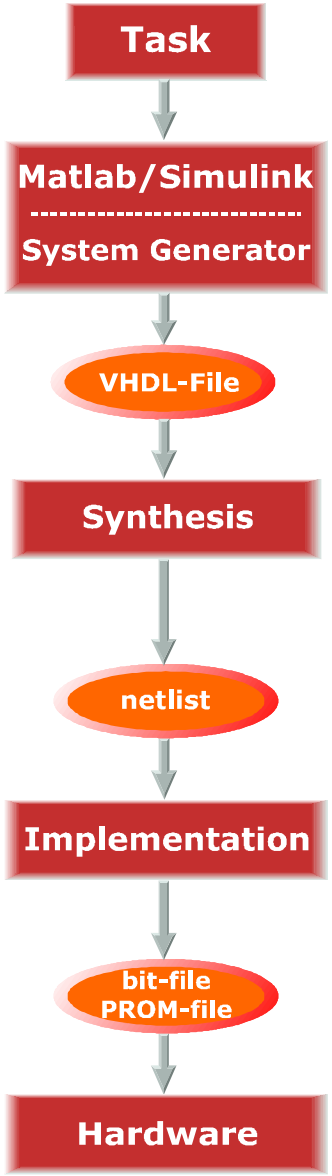
3. FPGA programming with graphical design entry on high level

Advantages:

- easy graphical programming on high level
- a good overview and documentation of the design is guaranteed
- bit and cycle true simulation
- programming of big FPGAs possible
- automatic generation of HDL and netlists

Disadvantages:

- knowledge of basic HDL syntax, Matlab language and Simulink blocksets necessary, long period of vocational adjustment of application engineers
- expensive software costs



Programming of FPGAs

Problems of actual programming methods especially for newcomer:

- **high costs of comfortable programming software**
- **high costs for training period of application engineers**

Additional hardware problems for newcomer:

- **unknown technology of FPGA design in PCB layout**

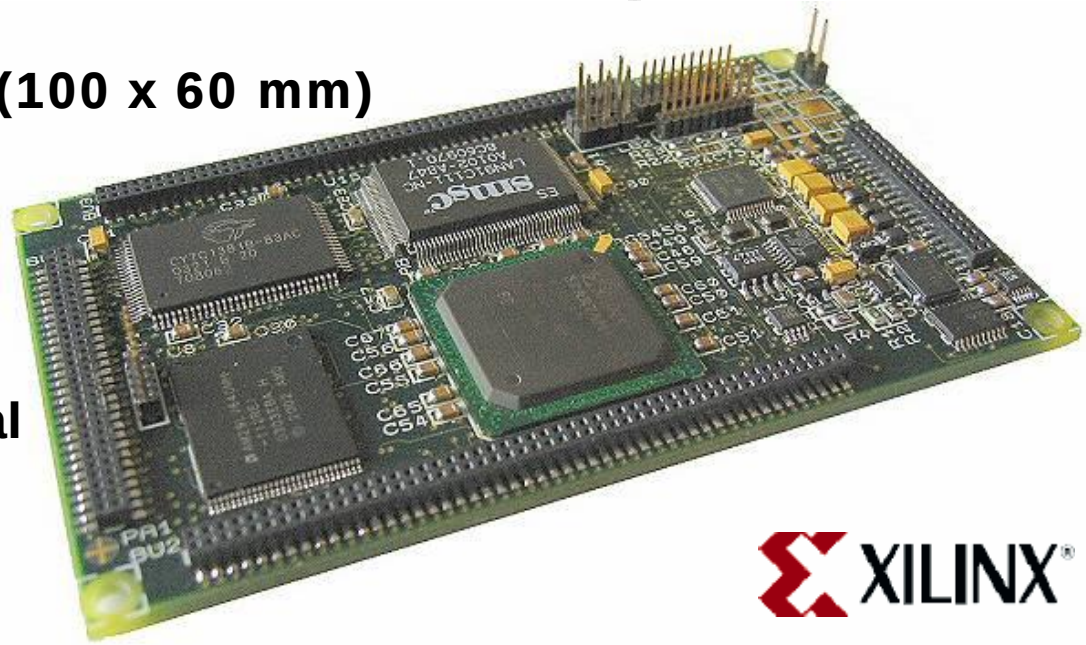
These are important problems of small and medium enterprices and build high walls to use this technology.

Solving the problems

Hardware problems can be solved by use of small and universal FPGA development kits.

Example of a development kit that can also be used as final product for small and medium series:

- **Development Kit = final product**
- **space-saving design (100 x 60 mm)**
- **use of established FPGA from XILINX**
- **widespread peripheral devices onboard**

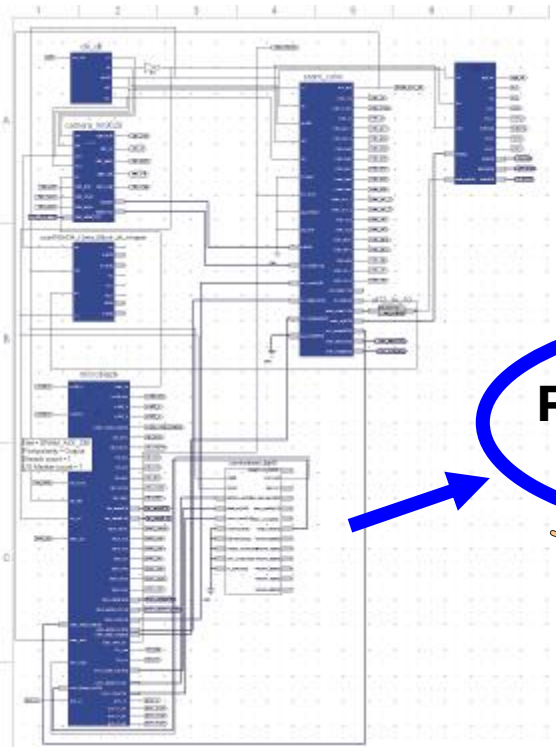


XILINX®

Modular Graphical Programming (MGP)

Application Programming with XILINX ISE and AVT Toolbox

- **vocational adjustment in very short time**
- **graphical programming with complex functional blocks (cores)**
- **free software for implementation and graphical input**
- **extension of functions with basic elements, IP-Cores or HDL**



Task

Fast realization of application

Core Toolbox

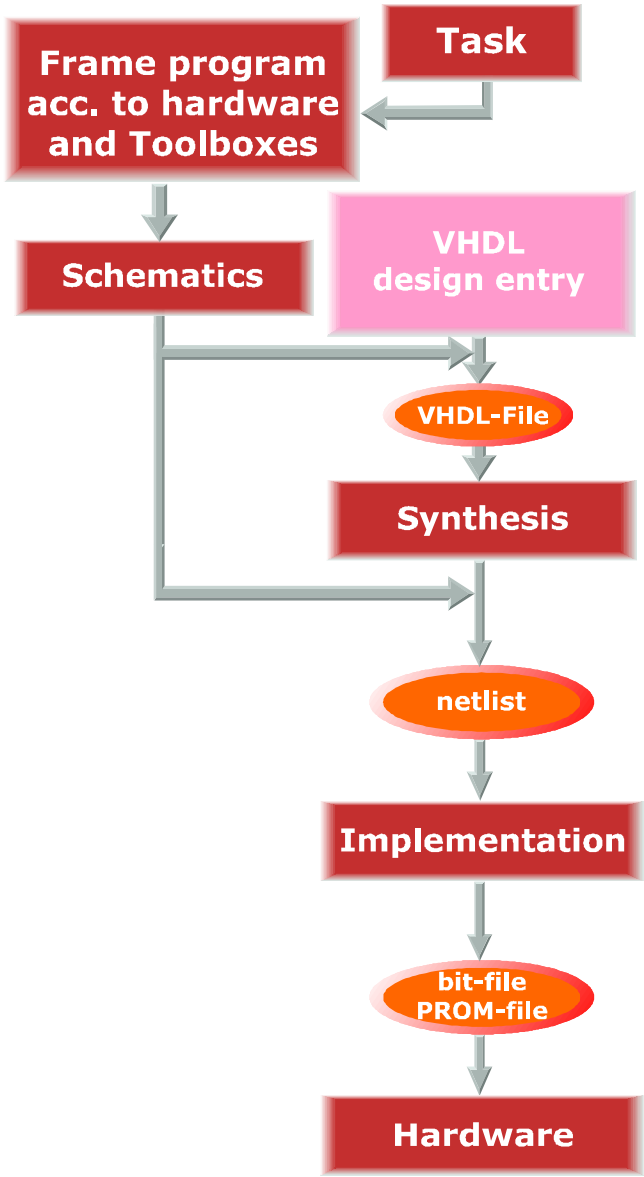
Software Dokumentation

Modular Graphical Programming (MGP)

Design flow of MGP:

- application engineer must have the task and realization idea
- application engineer use graphical frame program according hardware
- add for application necessary functions as graphical modules of toolbox
- generation of FPGA programming file by using implementation tool

MGP consists of 6 steps à



Modular Graphical Programming (MGP)

1. Step: Calculation of design resources

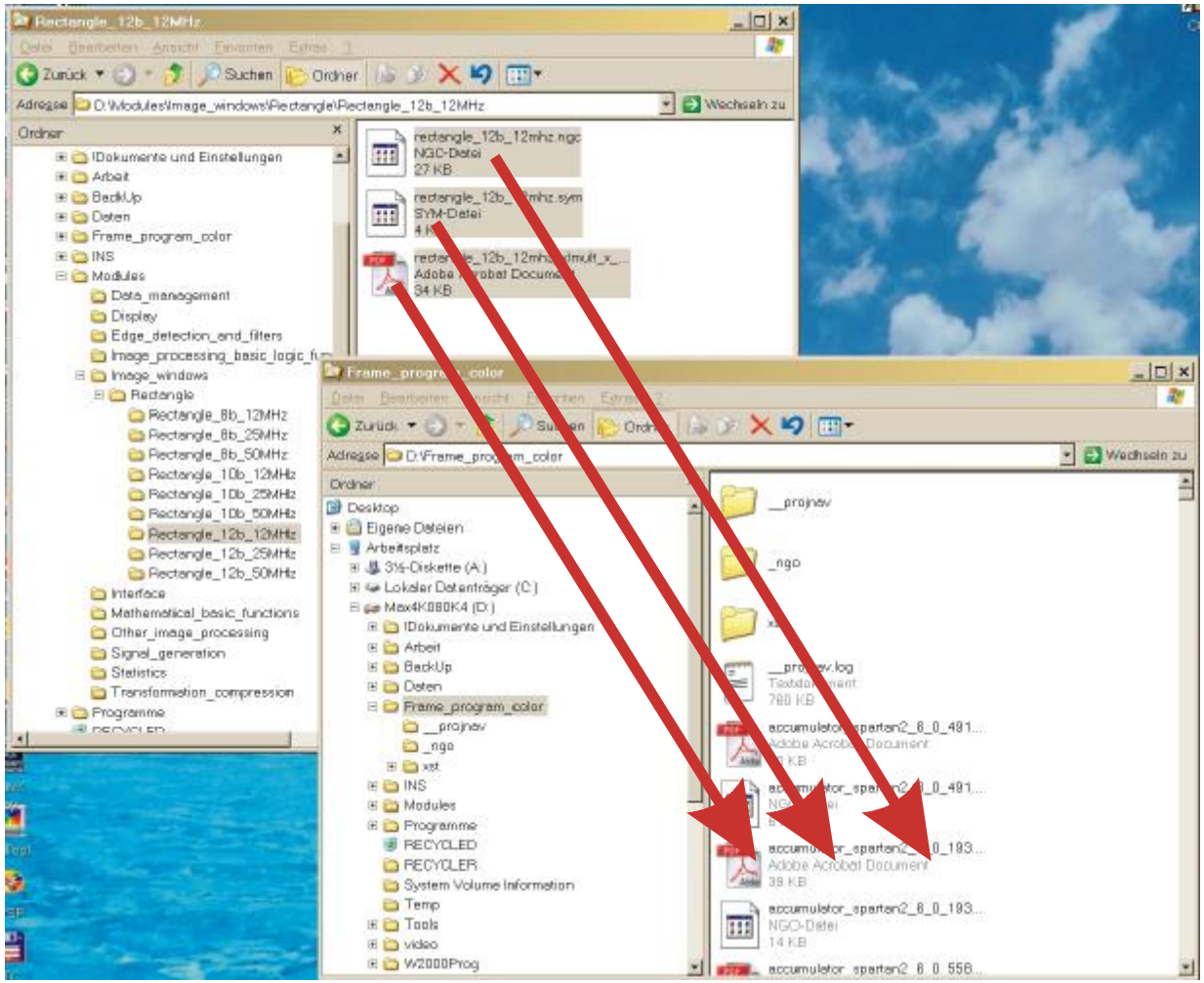
- choice of design functions according the application
- enter of bit width and number of functions in an Excel sheet
- output is the sum of necessary design resources (slices and BlockRAM) comparable with chosen FPGA

	A	G	I	J	K
1	**Functions in XILINX-FPGAs**	**Input**		**Output**	
2					
3					
4					
5	**Mathematical basic functions**				
6	Constant			0	0
7	Add			0	0
8	Sub			0	0
9	Mult	5	10	0	0
10	Abs			0	0
11	Scale bit width of vector			1	1
12	Limit			0	0
13	log_2			0	0
14				1	1
15	**Signal generation**			0	0
16	Pulse			0	0
17	counter bit 1 (50MHz)			1	1
18	counter bit 1_25MHz			0	0
19	counter bit 1_50MHz			0	0
20	counter bit 2 (50MHz)			1	1
21	counter bit 2_25MHz			0	0
22	counter bit 2_50MHz			0	0
23	counter bit adjustable with button			1	1
24	edge detection Bank_increasing			0	0
25	edge detection Bank_synchron			0	0
26	Pattern (50MHz)			1	1
27	Pattern_50MHz			0	0
28	counter bit			0	0
29	counter bit with button			1	1
30	counter bit vagrantly			0	0
31				0	0
32	**Image windows**			1	1
33	rectangle_xxx_12mHz	5	12	70	0	350
34	rectangle_xxx_25mHz			0	0
35	rectangle with Filter			1	1
36				0	0
37	**Edge detection and filters**			0	0
38				1	1
39	binary image + edge detection (3x 4/3/0/10 px)			0	0
40	Sobel (8x; x/y/xy/ag)			0	0
41	Laplace (4x, 1/2)			1	1
42	low_pass		10	10	0	50
43	high_pass			0	0
44	Dilation (8x, OP1, xOP, yOP, xOP, yOP)			1	1
45	Prewitt (8x, x/y)			0	0
46	Roberts (8x, OP1/OP2)			0	0

Modular Graphical Programming (MGP)

2. Step: Copy of necessary module files

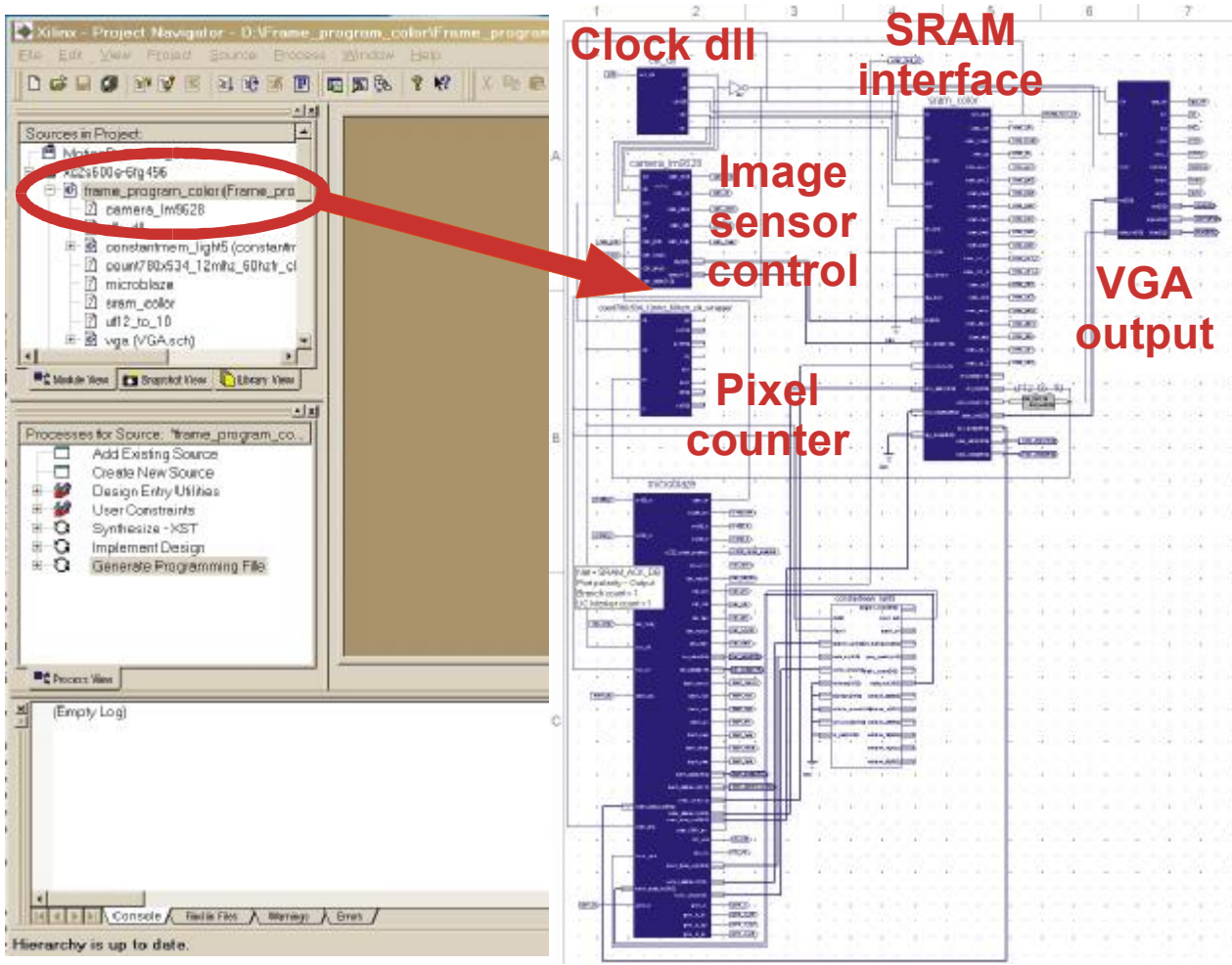
- copy of module files from tool-box directory to working directory
- module files are netlists and graphical symbols of functions
- additional report files consist of space and timing information



Modular Graphical Programming (MGP)

3. Step: Start of frame program inside implementation tool

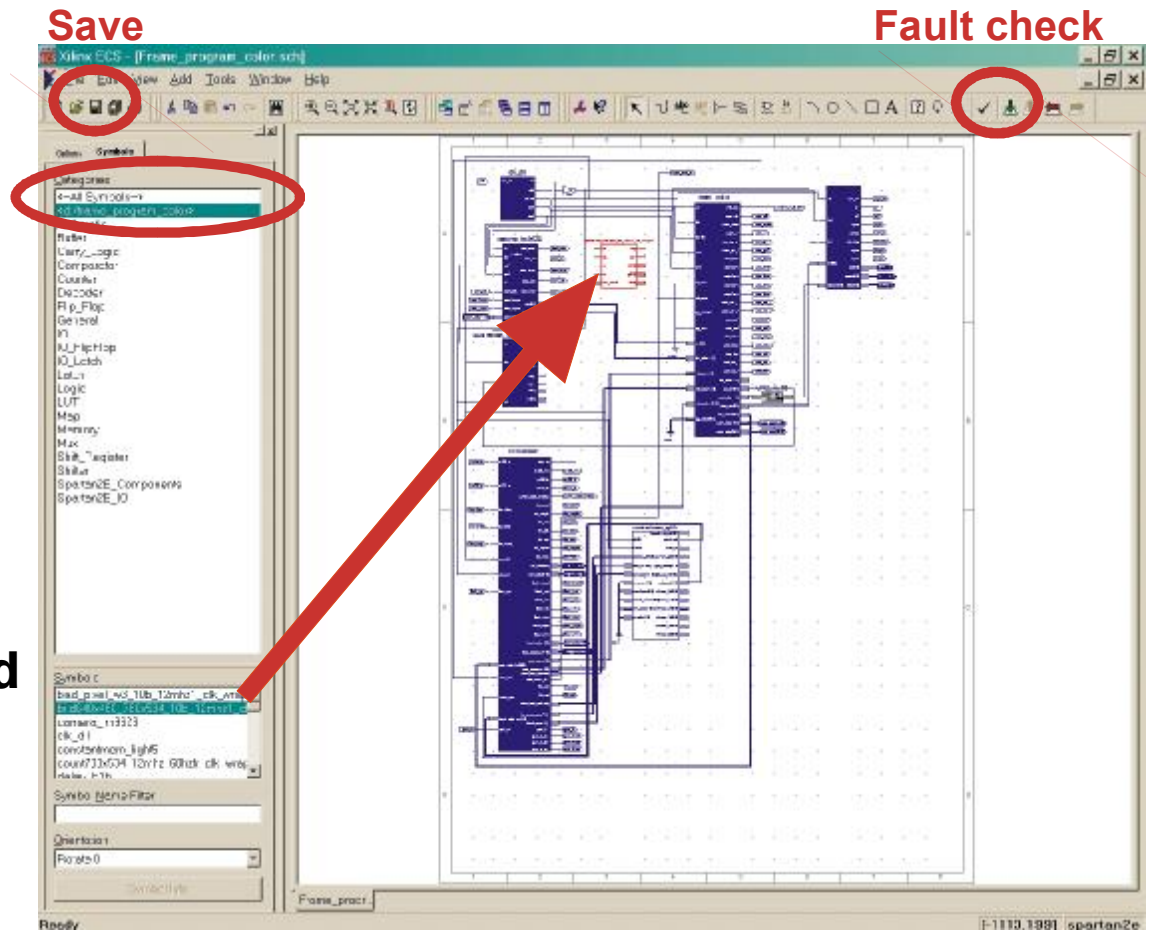
- Xilinx ISE project navigator shows parts of design of the frame program and implementation steps
- open frame program in ECS/ Schematics



Modular Graphical Programming (MGP)

4. Step: Assembling of module symbols in application schematic

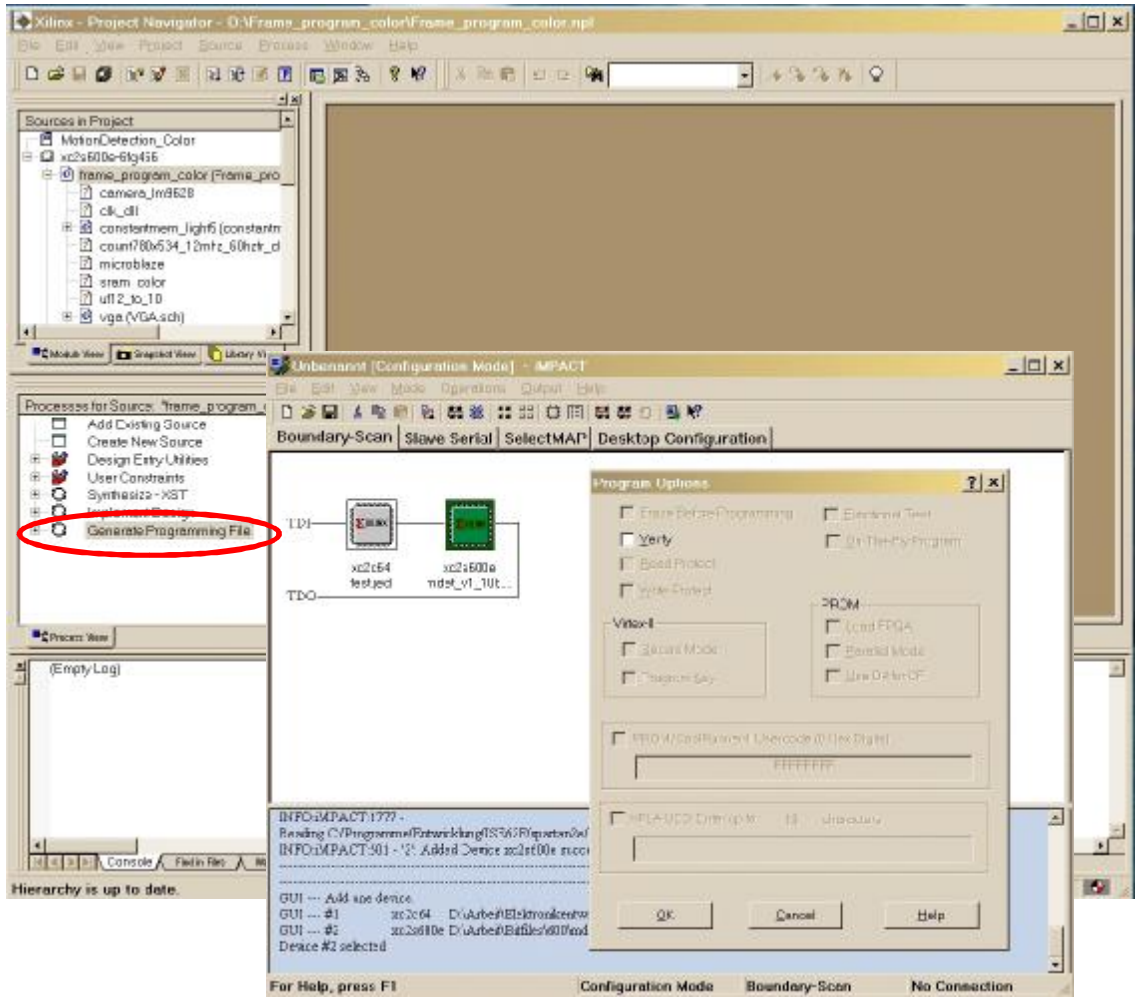
- Symbols of module functions can be found in working directory in ECS/Schematic. After choice of modules, they can be assembled inside application schematic
- Data lines between frame program modules can be separated and re-connected with modules of predefined functions between them
- fault check and save



Modular Graphical Programming (MGP)

5. and 6. Step: Program file generation and hardware programming

- Automatic generation of program file by double-clicking last implementation step; all steps will run automatically
- the “IMPACT” program (part of Xilinx ISE) load generated bit-file inside FPGA



how does it look like à

Modular Graphical Programming (MGP)

- MGP toolbox consist of more than 800 modules with a center on video processing, digital signal processing and communication

- Module functions are arranged in different categories (mathematical basic functions, signal generation, edge detection and filters, transformation and compression, statistics, display functions, interface control, image processing and data management)

- Modules contain also of very complex functions like JPEG compression, use of XILINX MicroBlaze 32 bit RISC processor inside FPGA and mixed hardware/software modules connected to MicroBlaze processor.

Conclusion

- **Modular Graphical Programming (MGP) is made for an easy, cost efficient and fast *entry* in FPGA design and/or a fast design *extension* of existing designs with cores**

- **MGP is easy to use and fast to learn à save costs**

- **MGP make fast finish of applications possible à save costs**

- **MGP is seamless integrated in implementation tool, no special programming software necessary à save costs**

- **MGP is also extendable by own functions à save costs**
  (MGP make different design entries possible:
  Schematic modules, VHDL modules, modules of Core/System Generator and IP Cores)

- **After *start* working with FPGAs MGP will expand the knowledge about FPGA programming systematically**

Our offer

- A lot of modules of MGP are free of charge with development kit of AVT GmbH
- Bigger and more complex functions will deliver for low costs
- Generation of newer versions and more functions is in progress
- Access of FPGA knowledge and MGP programming method is supported by modular training courses

MGP is also ideal suited for System-on-Chip designs on FPGA with hardware and software libraries of MicroBlaze processor, we offer our knowledge in training courses

For more information please contact us
or come to me in coffee break à

Contact Information

AVT GmbH Ilmenau

Automatisierungs- und Verfahrenstechnik

Am Hammergrund 1

D-98693 Ilmenau

- GERMANY -

Email and Telephone Numbers:

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**

Contact Person:

Dr.-Ing. Jörg Pospiech	**CEO, Director of Development**
Herr Wilfried Geyer	**Constructions**

Programming of FPGAs