Filling a Gap Between Hardware and Software
Cologne Chip GateMate FPGA

Anton Kuzmin

FOSDEM’24, 3 & 4 February 2024, Brussels
Who am I...
Embedded, modular, and real-time systems developer for almost 30 years

- not really a software developer ... but write code sometimes
- i8051, i8080, i960, Digital Alpha, x86, PowerPC, MIPS, ARM, RISC-V
- CAMAC, VME, CompactPCI, AdvancedTCA, μTCA, SoMs
- FPGA and SoC-FPGA (Altera/Intel, Microsemi/Microchip)

Largely disappointed by the ever-growing gap between software development pace of innovations and RTL FPGA design approach frozen in the past century.
Intro

FPGA and software development

Hardware that made it possible

Road ahead

Contact info

Backup slides
Why software developers should care about FPGA

- conventional hardware architectures are stuck
- the only two mainstream HDLs represent software technology level of a stone age (well, last century)

There should be a way...

- for reconfigurable hardware to catch up with the progress made in the software development world for the last four decades
- bring fun and younger generation to FPGA development
Current state of FPGA development (for the last 40 years)

• RTL and a rise of Verilog and VHDL were a revolution
• ...and nothing can compare to it for 40 years
• may be it is still fine for ASIC
• ...but FPGAs are different
Two reasons for disappointment (1/2)

Benefits of reconfigurability at run-time are under-explored and virtually unexploited

IBM 402 plug-board. Chris Shrigley, 2003, CC-BY
Two reasons for disappointment (2/2)

The only available way is a cross-development

IBM PC 5150, Ruben de Rijcke, 2010, CC-BY-SA
Demo...
Perspective

One small LUT run-time modification to demonstrate, but a huge leap towards a self-hosted FPGA development.

- run-time ISA extensions
- JIT to hardware
- iterative, interactive, and incremental development
- encourage exploration and experiments (education)
- native FPGA development (native is not naïve)
GateMate FPGA Architecture (1/3)

- novel CPE architecture (8-input LUT tree, two flip-flops or latches)
- low power consumption (GlobalFoundries 28 nm SLP (Super Low Power) process)
- 4 programmable PLL
- dual-ported block RAM
- 5 Gbps SerDes
- configuration via QSPI up to 100 MHz
- all pins configurable as single-ended (1.2 .. 2.5V) or LVDS
- all GPIO blocks support DDR
- 324-ball BGA, 15x15mm
GateMate FPGA Architecture (2/3)

Figure 2.1: Simplified architecture overview

CologneChip GateMate FPGA Datasheet, DS1001, January 2023, page 21
GateMate FPGA Architecture (3/3)

![Diagram of GateMate FPGA Architecture](image)

**Figure 2.2:** Cologne Programmable Element (CPE)

CologneChip GateMate FPGA Datasheet, DS1001, January 2023, page 22
Why to design a module

- CCGM1A1 is the best thing since XC6200
- an Evaluation Kit was not available back in mid-2020
- a module is smaller and reusable
- freedom for experiments (e.g. to interconnect several FPGAs)
- best way to get to know a new chip
- fun and easy exercise with KiCAD (at least it seemed so at the beginning)
Current Status

• three boards designed and manufactured
• the boards are functional (from the very first versions and with just a few minor problems)
• schematic symbol and PCB footprint for GateMate FPGA are accepted into KiCAD v7 libraries
• control application is functional enough to debug, test, and configure the module
• several VHDL examples are running
• support for the module is (to be) added to the FuseSoC and LiteX (work in progress, nothing is upstreamed yet)
• ...it took roughly 5 times longer than initially estimated
GMM-7550 Module

- Cologne Chip GateMate FPGA CCGM1A1
- wide-range input power
- module control: discrete signals and I²C
- 8 I/O banks available on 4 connectors with identical pinout
- 5 Gbps SerDes
- programmable clock
- all configuration modes are supported (Active and Passive SPI, and JTAG)
Raspberry-Pi 40-pin GPIO HAT Adapter

- power for the module from R-Pi 5V or a separate power connector
- current and voltage monitoring (ADM1177)
- module control signals, I²C, SPI, and UART
- 2x5 .1” JTAG connector
- 2.5V/3.3V I/O level converters
- two 12-pin P-mod connectors for extension modules
- access to 3 I/O connectors
- 4x LEDs
- R-Pi HAT ID EEPROM
Memory Extension Module

- SRAM 512K x8 (CY7C1049GN30-10ZSXI)
- QSPI-NOR 128Mb (16MiB, IS25LP128-JBLE)
- mechanical design validation
It doesn’t have to be...

It has to be Cologne Chip GateMate (at least for now), but it does not have to be...

- RISC-V (it is still a good choice)
- C (anything else would be better)
  Forth, Lua, microPython, Scheme, Scala, Rust, ...
- GMM-7550 module
  CologneChip Evaluation board, Trenz, Olimex
You are invited to innovate... 

The greatest engineering reward and pleasure is to see the results used, so, please:

- create software for FPGAs and with FPGAs
- build modules, design baseboards and extension modules
- report problems
- create custom designs
- experiment
- share
Thank you!

Anton Kuzmin
ak@gmm7550.dev
https://github.com/gmm-7550/

Questions?..
GMM-7550 Module – Power

- wide input range (2.9 .. 6.5V), may be powered directly from a 3.3V baseboard, 5V USB, or single cell Li-Pol
- DC-DC are synchronized to the base clock and run in counter-phase (default 1.25 MHz, PLL programming option)
- $0.9/1.0/1.1V$ $V_{\text{core}}$ (build-time option)
- $V_{\text{io}}$ may be supplied directly on the module (2.5V, build-time option) or from a baseboard (individually for each I/O bank)
- ADP2164 step-down DC-DC, $V_{\text{io}}$ (2.5V) and $V_{\text{core}}$ are rated up to 4A
- separate LDO (ADP1753) for SerDes and SerDes PLL, 1.0/1.1V, 800mA
- input voltage monitor/reset generator on the module, an external reset input, and $I^2C$ controllable reset
GMM-7550 Module – Clock

- Texas Instruments CDCE6214 PLL with internal EEPROM
- 25 MHz crystal on the module
- LVDS reference clock input
- single-ended (LVCMOS 2.5V) and two differential output clocks
- default 100 MHz differential clock to the FPGA SER_CLK input
- dedicated output for DC-DC synchronization
GMM-7550 Module – FPGA Configuration

- JTAG (2.5V) available on the module connector
- Active Serial mode from SPI-NOR on the module or on a baseboard
- Passive Serial mode from a baseboard
- configuration mode and SPI connection configurable via I²C
- SPI-NOR on the module is accessible from a baseboard
- default mode: Active Serial from SPI-NOR on the module