GMM-7550 – Cologne Chip GateMate FPGA Module

Anton Kuzmin

ORConf 2023, September 15th to 17th, Munich

Who am I. . .

Embedded, modular, and real-time systems developer for almost 30 years

- i8051, i8080, i960, Digital Alpha, x86, MIPS, ARM, RISC-V
- CAMAC, VME, CompactPCI, AdvancedTCA, μTCA, SoMs
- FPGA and SoC-FPGA (Altera/Intel, Microsemi/Microchip)

Introduction

Motivation and Current State

Technical Details

- Cologne Chip GateMate FPGA
- GMM-7550 Module
- Additional Modules
- Software and RTL Code Examples

What's next

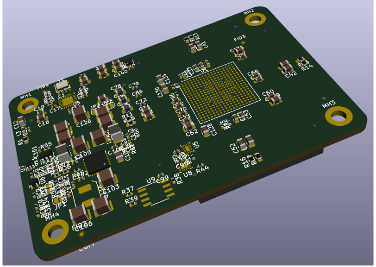
Contact info

Why GateMate FPGA

- a new kid on the block
- FOSS commitment (Yosys for synthesis right now, `nextpnr` announced)
- semi-documented (and semi-open) bitstream format
- possibility to change configuration at run-time from inside

Why design a Module

- 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 enought 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

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)

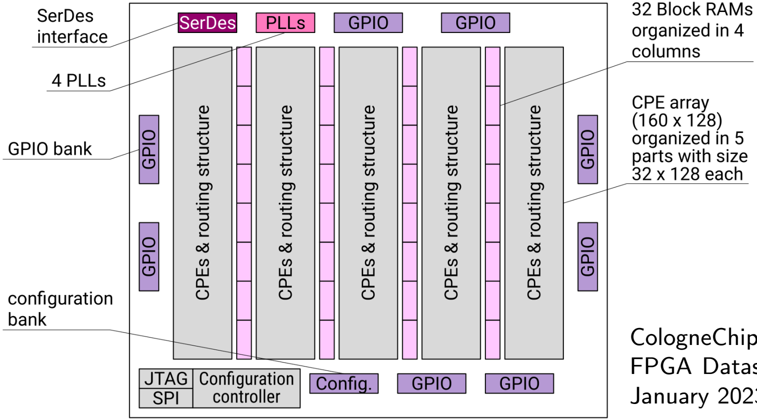


Figure 2.1: *Simplified architecture overview*

CologneChip GateMate FPGA Datasheet, DS1001, January 2023, page 21

GateMate FPGA Architecture (3/3)

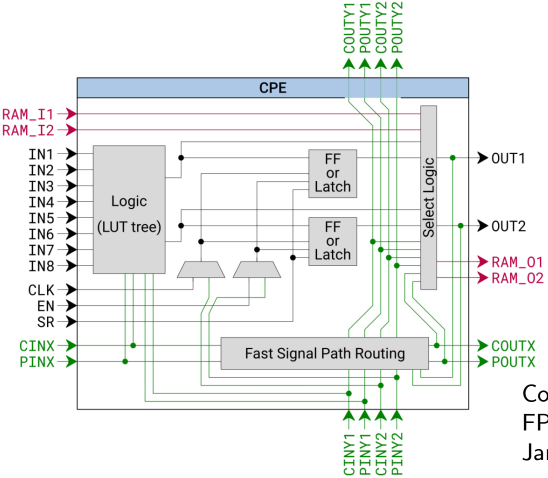
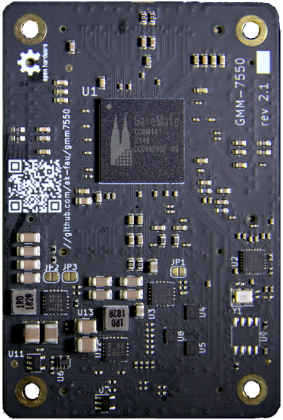


Figure 2.2: *Cologne Programmable Element (CPE)*

CologneChip GateMate FPGA Datasheet, DS1001, January 2023, page 22

GMM-7550 Module – Key Features

- Cologne Chip GateMate FPGA CCGM1A1
- wide-range input power
- module control: discrete signals and I^{2}C
- 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)



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_{core} (build-time option)
- V_{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_{io} (2.5V) and V_{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

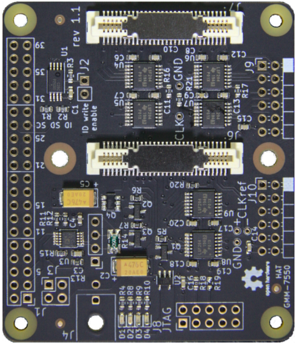
- Texas Instruments CDCE6214 PLL with internal EEPROM
- 25 MHz crystal on the module
- LVDS reference clock input
- single-ended (LVCMOS 2.5V) and two differential output clocks
- default 100 MHz differential clock to the FPGA `SER_CLK` input
- dedicated output for DC-DC synchronization

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 I^{2}C
- SPI-NOR on the module is accessible from a baseboard
- **default mode:** Active Serial from SPI-NOR on the module

Raspberry-Pi 40-pin GPIO HAT Adapter

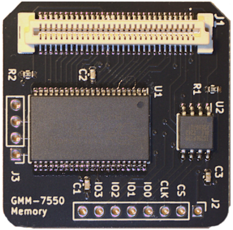
- power for the module from R-Pi 5V or a separate power connector
- current and voltage monitoring (ADM1177)
- module control signals, I^2C, SPI, and UART
- 2x5 .1" JTAG connector
- 2.5V/3.3V I/O level converters
- two 12-pin P-mod connectors for extension modules
- access to 3 I/O connectors
- 4x LEDs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



Memory Extension Module

- SRAM 512K x8 (CY7C1049GN30-10ZSXI)
- QSPI-NOR 128Mb (16MiB, IS25LP128-JBLE)

- mechanical design validation



Control Application

- Raspberry-Pi, VisoinFive
- ArchLinux ARM, `buildroot`
- CLI and web interface (Python panel)
- . . . live demo

VHDL Examples

- Yosys with ghdl-yosys-plugin, Cologne Chip `p_r`, GNU `make`
- GateMate primitives VHDL library
- blinking LED *(with different clock inputs and PLL settings)*
- serial loopback
- SPI bridge to access SPI-NOR chips on the memory module from the baseboard SPI
- export examples as stand-alone projects
- . . . live demo

Third-party Frameworks

- FuseSoC
  - blinky
  - servant *not working yet*
- LiteX – TODO

Next steps – for myself

To keep in mind in the future, that creating an open source hardware design takes the same effort and responsibilities as doing it "for money", just a bit more demanding, and no doubt is much more fun.

- test remaining things (SerDes, SRAM, ID EEPROM, operation at different V_core)
- stress test
- more examples, integration into frameworks
- RISC-V cores, software
- partial self-reconfiguration
- ...

Next steps – for everybody

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

- build modules, design baseboards and extension modules
- report problems
- create custom designs
- experiment
- share your results

open source hardware

Thank you!

Anton Kuzmin
ak-fau+orconf@no-problem.cc
https://github.com/ak-fau/

Questions?..