Cologne Chip

GateMate™ FPGA Datasheet

CCGM1A1

DS1001 Pre-Release Datasheet December 2020

Cologne Chip AG
Eintrachtstr. 113
50668 Köln

Tel.: +49 (0) 221 / 91 24-0
Fax: +49 (0) 221 / 91 24-100

https://colognechip.com
info@colognechip.com

Contents

List of Figures

List of Tables

Revision History

This Datasheet is constantly updated. The latest version of the document can be found following the link below:

- **DS1001** - GateMate™ FPGA CCGM1A1 Datasheet

Date	Remarks
December 2020	Expanded JTAG interface documentation in Section 2.2.5. Updated package height and tolerances in Figure 6.1.
October 2020	Updated 324-ball FBGA package to final pinout: • FBGA pinout in Figures 5.1 and 5.2. • Pad types and pad lists in Tables 5.1 and 5.2.
February 2020	Initial pre-release.

Chapter 1

Introduction

1.1 Introduction

The Cologne Chip GateMate™ FPGA Series is a family of small to medium-sized FPGAs that addresses all requirements of typical applications. The programmable silicon can be used from low-power to high-speed applications and thus in a wide range of fields: industry, automation, communication, security, automotive, IoT, AI, and lots more.

Logic capacity, power consumption, package size and PCB compatibility are optimized for a wide range of applications. As these FPGAs combine various features with lowest cost in the market, the devices are well suited from university projects to high volume applications. Because of the outstanding ratio of circuit size to cost, even price-sensitive applications can also now take advantage of FPGAs. The devices are manufactured using Globalfoundries™ 28 nm SLP process in Dresden, Germany. The GateMate™ FPGA program is supported by the German Federal Ministry for Economic Affairs and Energy as part of the Important Project of Common European Interest (IPCEI) on Microelectronics project, which is also supported by the European Commission.

Supported by:

on the basis of a decision by the German Bundestag

Cologne Chip is a semiconductor company based in Cologne, Germany. The company, which celebrates its 25th anniversary in 2020, has excellent industry knowledge and an experienced team of developers. The design and manufacturing location *Made in Germany* represents a unique selling proposition of globally competitive FPGAs in the market.

Cologne Chip offers and supports a variety of development tools:

- The GateMate™ Evaluation Board is a feature-rich, ready-to-use development platform that serves as a reference design and for a direct entry into application development.
- With the help of the GateMate™ Programmer, the FPGA can be configured in various ways.

Cologne Chip provides a comprehensive technical support. Please contact our support and sales teams or visit our website for more information.

1.2 Features

- Novel Programmable Element Architecture
  - 20,480 programmable elements for combinatorial and sequential logic
  - 20,480 8-input LUT-trees / 40,960 4-input LUT-trees
  - 40,960 Latches / Flip-flops within programmable elements
  - Each programmable element configurable as:
    - 1-bit full adder
    - 2-bit full adder or
    - 2x2-bit multiplier
  - Dedicated logic and routing for fast arithmetic and arbitrary-sized multipliers
- 9 GPIO Banks
  - 162 user-configurable GPIOs
  - All GPIOs configurable as single-ended or LVDS differential pairs
  - Double Data Rate (DDR) registers in I/O cells
  - I/O voltage range from 1.2 to 2.5 V
- 4 Clock Generators (PLL)
  - max. oscillator output frequency from 500 MHz to 1,250 MHz
- 2.5 Gb/s Serializer/Deserializer (SerDes) Controller
- Flexible Memory Resources
  - 1,310,720 total RAM bits distributed over 32 SRAM blocks
  - Each RAM block configurable as two independent 20K blocks or single 40K block
  - Simple or True Dual Port (SDP/TDP) or FIFO mode
  - Data width from 1 bit up to 40 bits (TDP) or 80 bits (SDP)
  - Bit-wide write enable
  - Error Checking and Correction (ECC) for certain bit widths
- Flexible Device Configuration
  - Configuration bank switchable to user I/O
  - JTAG interface with bypass to SPI interface
  - active/passive SPI interface for singe-, dual- and quad-mode operation
  - SPI flash interface in active mode
  - Multi-Chip configuration from single source
- Application modes: *low power, economy, speed*
  - Mode adjustable on each chip via core voltage
  - Core voltage range from 0.9 to 1.1 V
- Package
  - 15x15 mm 324 balls 0.8 mm Fine-pitch Ball Grid Array (FBGA) package
  - Only 2 signal layers required on PCB

Chapter 2

Architecture

2.1 GateMate™ FPGA Overview

The basic functional elements of GateMate™ FPGA are set up in an array structure of 128×160 elements called Central Programming Element (CPE) as described in Section 2.2.1.

All CPEs are interconnected by a routing structure of 132×164 so called Switch Boxes (SBs) as described in Section 2.3.

Additional functional blocks are Dual Port SRAMs, Phase Locked Loops (PLLs), GPIO cells, SPI configuration and data flash interface, JTAG interface and Serializer/Deserializer (SerDes) interface.

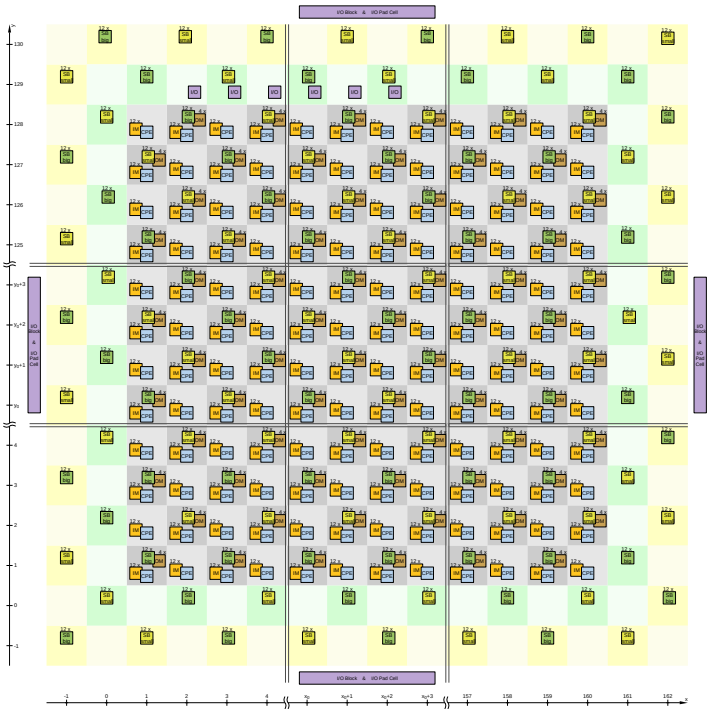


Figure 2.1: Simplified Architecture Overview

2.2 FPGA Functional Elements

2.2.1 CPE

General purpose combinatorial and sequential circuits are implemented using Central Programming Elements (CPEs). The CCGM1A1 has 20,480 CPEs arranged in a 160 × 128 matrix. Each CPE can be set up to the following combinatorial functions:

- 8 inputs with LUT-2 tree
- 6 inputs with MUX-4 function
- Dual 4 inputs with LUT-2 tree each
- 1-bit or 2-bit full adder, expandable to any length in horizontal or vertical arrangement
- 2x2-bit multiplier, expandable to any multiplier size

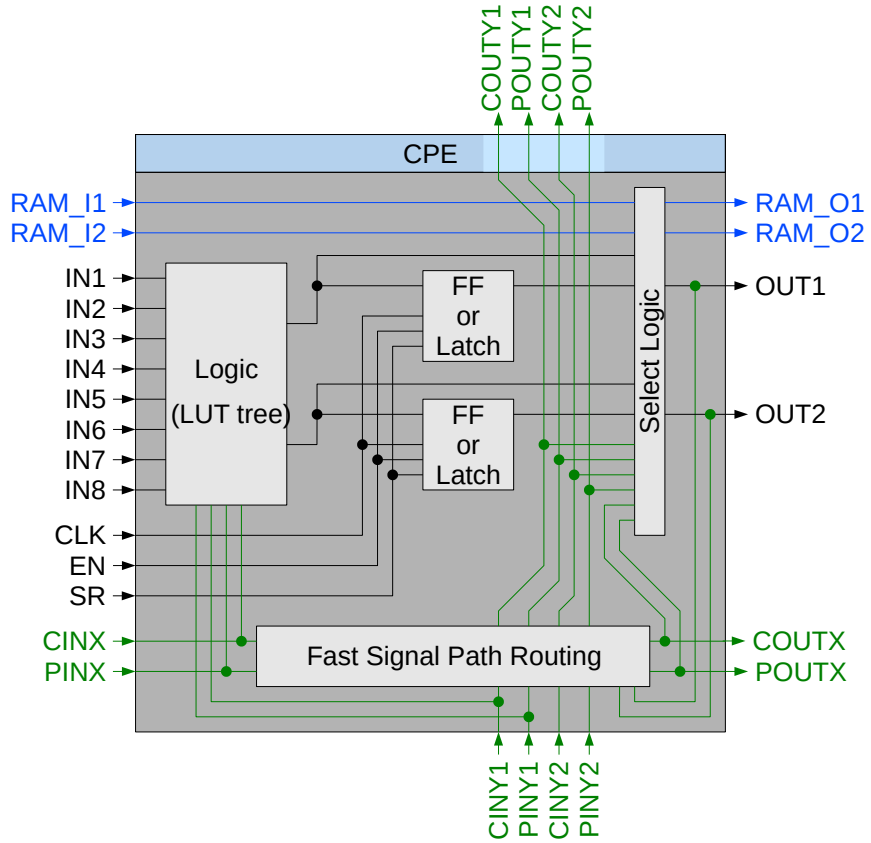


Figure 2.2: Central Programming Element

Two CPE outputs OUT1 and OUT2 are available and each can use a Flip-Flop or Latch function.

Furthermore, the CPEs include fast signal routing paths, typically used for fast clock and carry propagation in adder and multiplier functions, e.g., fast CPE to adjacent CPE connections or even fast signal propagation over any span of CPEs.

Figure 2.2 illustrates the general structure of a CPE. General combinatorial and sequential functions are supported from fast signal path routing. In addition, the Dual Port SRAM blocks of GateMate™ FPGA, as described in Chapter 2.2.3, require some CPE ports RAM_I[2:1] and RAM_O[2:1] for CPE-to-SRAM connection.

2.2.2 General Purpose Input/Output (GPIO)

The GateMate™ FPGA offers up to 162 general purpose input / output pins. These are organized in signal pairs, so called *pad cells*, which can either operate as two independent, bidirectional single-ended signals or they can be set up as bidirectional Low-Voltage Differential Signaling (LVDS). Figures 2.3 and 2.4 show the structure of the pad cells and their interconnection to the FPGA elements.

Nine pad cells build up a GPIO bank as shown in Figure 2.5 on page 15. Eight banks and another optional bank [1] are available.

The single-ended GPIO support the LVCMOS standard up to 2.5 V nominal supply voltage, compliant to the standards JESD8-5 (2.5 V) and JESD8-7 (1.8 V and 1.2 V).

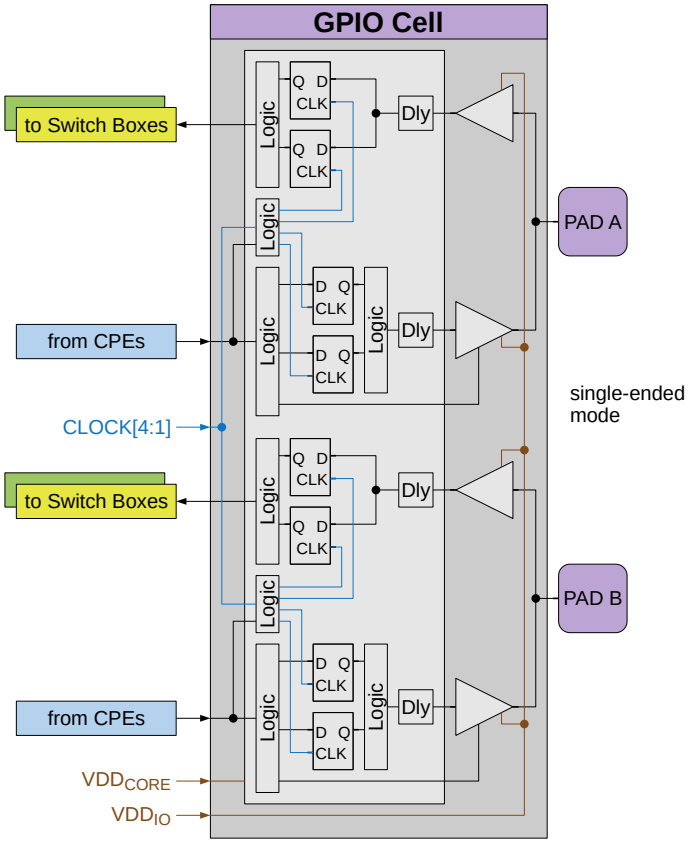


Figure 2.3: General Purpose Input/Output cell in single-ended mode

The single-ended GPIO pins have the following output features:

- 0 / 1 / high-Z
- Slew-rate control
- Driver strength 3 mA / 6 mA / 9 mA and 12 mA
- 2 Flip-Flops
- Programmable delay line

[1] The FPGA configuration pins are arranged within the ninth GPIO bank. These pins can be used as normal GPIO bank after configuration process.

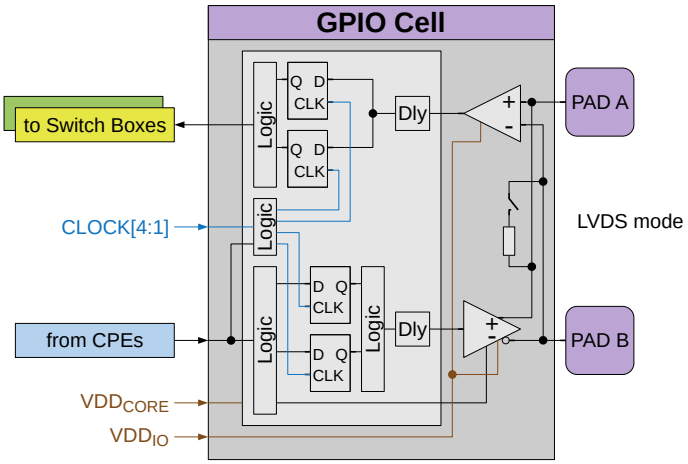


Figure 2.4: General Purpose Input/Output cell in LVDS mode

The single-ended GPIO pins have the following input features:

- Schmitt trigger input
- Pull-up / pull-down resistors or keeper functionality
- Input receiver disable for power reduction
- 2 Flip-Flops
- Programmable delay line

If a pad cell is used in its LVDS mode, both pads can be used together only. The LVDS pad is compliant to the LVDS 2.5 V standard. It further can operate down to 1.8 V nominal supply voltage, with common mode voltage $V_{DD}/2$. The LVDS input receiver can be used as voltage comparator.

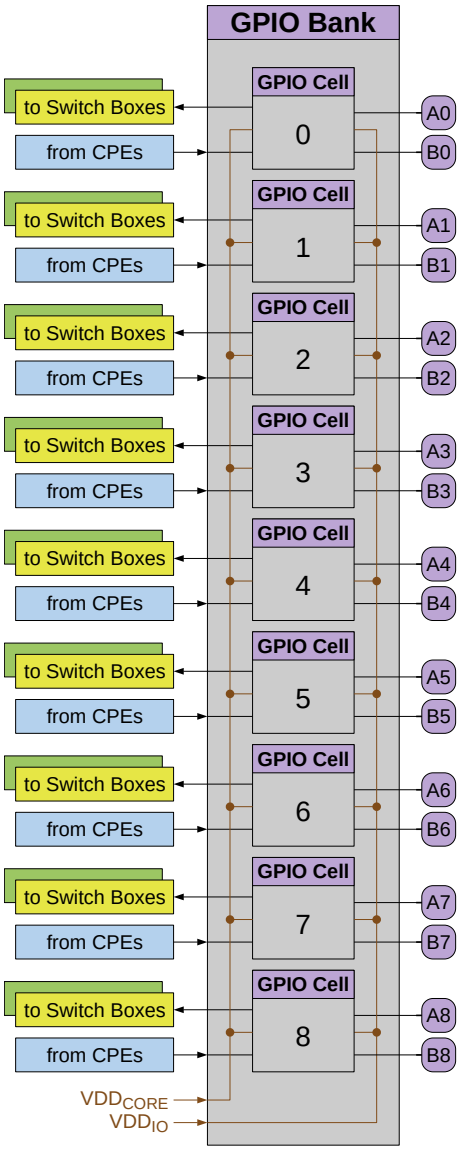


Figure 2.5: General Purpose Input/Output bank

2.2.3 Dual Port SRAM

The GateMate™ FPGA has SRAM blocks which are organized as configurable 40 Kbits Dual Port SRAM (DPSRAM). 32 DPSRAM blocks are available. They are arranged in 4 columns as shown in Figure 2.6. The DPSRAM ports are all connected to CPE outputs RAM_O[2:1] and CPE inputs RAM_I[2:1] as shown in Figure 2.2 on page 12. This means, DPSRAM inputs are feed by CPE outputs and DPSRAM output signals are connected to the Switch Boxes of the routing structure.

Each DPSRAM block allows usage of the memory in four different modes:

- True dual port non-split mode (TDP non-split)
- True dual port split mode (TDP split)
- Simple dual port non-split mode (SDP non-split)
- Simple dual port split mode (SDP split)

True dual port (TDP) mode versus **Simple dual port (SDP) mode**

The DPSRAM has two independent access ports. All combinations of operations at these two ports are possible: two reads, two writes or one read and one write. The ports can have two independent clock frequencies and different data widths.

One port can only be used for write access while the other port can only do read access. By that, data width can be increased up to 80 bit. The ports can have two independent clock frequencies and different data widths.

Non-Split mode versus **Split mode**

Both ports have access to the entire 40 Kbit memory. Input and output data widths can be different.

The DPSRAM is split into two completely independent DPSRAMs of half size 20 Kbit each.

The DPSRAM supports several data bit widths. The smaller the bit width, the larger the available address space. Table 2.1 shows all available DPSRAM configurations.

Each data bit has it's own write enable signal.

Table 2.1: Dual Port SRAM configurations

DPSRAM configuration	Non-Split TDP 40 Kbit	Non-Split SDP 40 Kbit	DPSRAM configuration	Split TDP 2x20 Kbit	Split SDP 2×20 Kbit
32 K × 1 bit	✓	✓	(RAM size per 20 Kbit block)		
16 K × 2 bit	✓	✓	16 K × 1 bit	✓	✓
8 K × 5 bit	✓	✓	8 K × 2 bit	✓	✓
4 K × 10 bit	✓	✓	4 K × 5 bit	✓	✓
2 K × 20 bit	✓	✓	2 K × 10 bit	✓	✓
1 K × 40 bit	✓	✓	1 K × 20 bit	✓	✓
512 × 80 bit	✗	✓	512 × 40 bit	✗	✓

Based on these four modes, additional features and functions can be configured:

- Different data widths ranging from 1 bit up to 40 bits (in case of TDP mode) or up to 80 bits (in case of SDP mode).
- Flexible selective forwarding of clocks, address and control signals.
- Write enable can be set for each bit separately.
- Usage as First-In First-Out (FIFO) memory in asynchronous or synchronous mode.
- Interconnection of several DPSRAMs in order to support larger bit widths (more than 40 bits in TDP mode, more than 80 bits in SDP).
- Interconnection of adjacent DPSRAMs (in case of 1-bit data words) in order to support more addresses (cascade mode).
- Error checking and correction (ECC) for certain bitwidths.

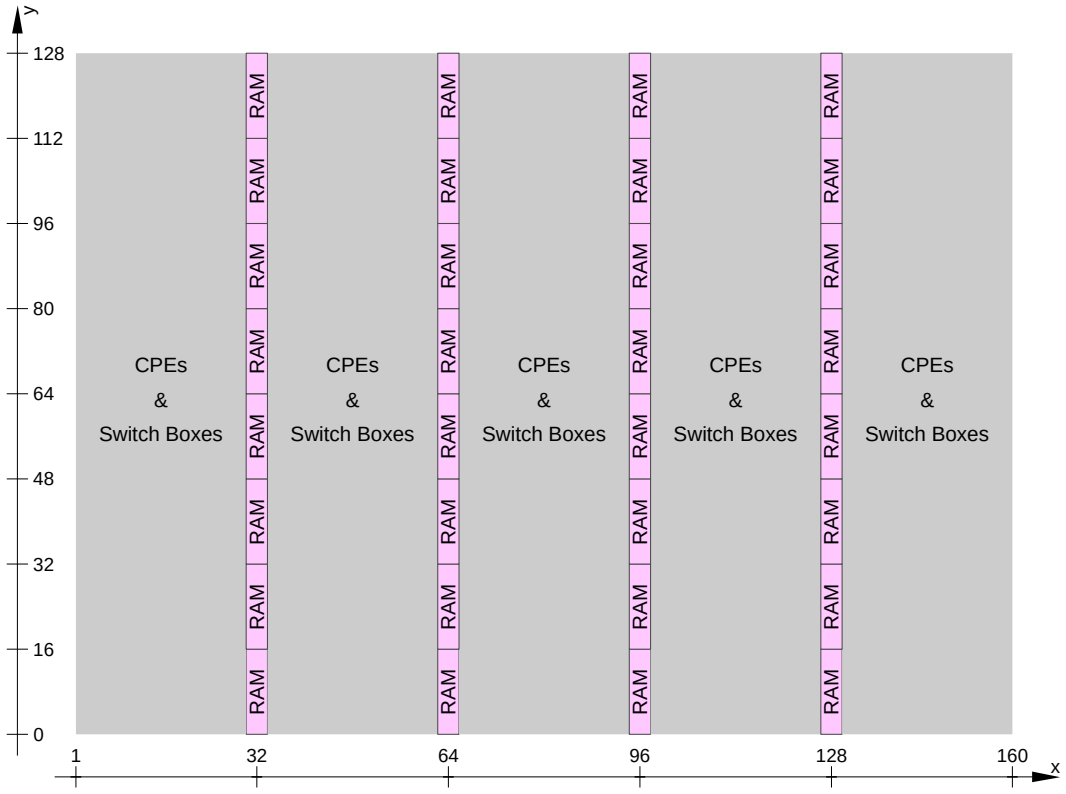


Figure 2.6: Dual Port SRAM blocks

The DPSRAM blocks contain a FIFO controller which can be configured in asynchronous or synchronous mode. The FIFO functionality is available for the following SRAM modes:

- TDP non-split, arbitrary but equal input and output bitwidth
- SDP non-split, 80 bit input and output bitwidth

If data shall be protected against unintended changes, the DPSRAM provides error checking and correction (ECC) encoding. For this feature the user can only use 32 or 64 bits data width since the remaining 8 or 16 bits are required for storage of parity bits. ECC is only available for the following configurations:

- TDP non-split, configured to 40 bit of which user can use only 32
- SDP non-split, configured to 80 bit of which user can use only 64
- SDP split, configured to 40 bit of which user can use only 32

For ECC the Hamming-code with 7 parity bits is used. It can correct one bit error and detect two bit errors.

2.2.4 Clock Generators (PLLs)

Four independent Phase Locked Loop (PLL) clock generators are available:

- All-digital phase locked loop (ADPLL) clock generator for versatile FPGA clock generation.
- Clock generation based on a digitally controlled oscillator DCO running with a typical frequency tuning range from 1 GHz to 2 GHz.
- Programmable clock frequency dividers for reference clock input, PLL loop divider and core clock outputs, enabling wide output and reference frequency ranges.
- Core clock outputs available at four 90°-spaced phases.
- Fast lock-in by binary frequency search.
- Autonomous free-running oscillator mode for FPGA configuration clock after power-up.

The maximum oscillator output frequency depends on the FPGA supply voltage as follows:

Low power mode	($VDD_{PLL} = 0.9\,V \pm 50\,mV$): 500 MHz
Economy mode	($VDD_{PLL} = 1.0\,V \pm 50\,mV$): 1000 MHz
Speed mode	($VDD_{PLL} = 1.1\,V \pm 50\,mV$): 1250 MHz

There are different reference clock pins selectable individually for each PLL:

- SerDes clock input SER_CLK (can also be used as general purpose clock input)
- SPI clock input SPI_CLK
- JTAG clock input JTAG_TCK
- Clock input IO_SB_A5
- Clock input IO_SB_A6
- Clock input IO_SB_A7
- Clock input IO_SB_A8

The same reference clock can be used from several PLLs at the same time.

2.2.5 JTAG Interface

The GateMate™ FPGA configuration bank provides a standard IEEE 1149.1-2001 compliant Test Access Point (TAP) that can be used for the following functions:

- Load configuration via JTAG [2]
- Full access to SPI interface using JTAG-SPI bypass
- Utilities to access external SPI dataflashes
- Boundary-Scan Testability: SAMPLE/PRELOAD and EXTEST
- Access to supplementary internal registers

Figure 2.7 illustrates the typical wiring between a single FPGA and a JTAG host controller. By connecting devices via the data input and output signals it is possible to chain multiple TAPs in a serial configuration. The operating range of the supply voltage on the configuration bank is shown in Table 4.3. The JTAG interface is always available after global reset. It will no longer be available once the configuration bank is switched to user I/O after configuration.

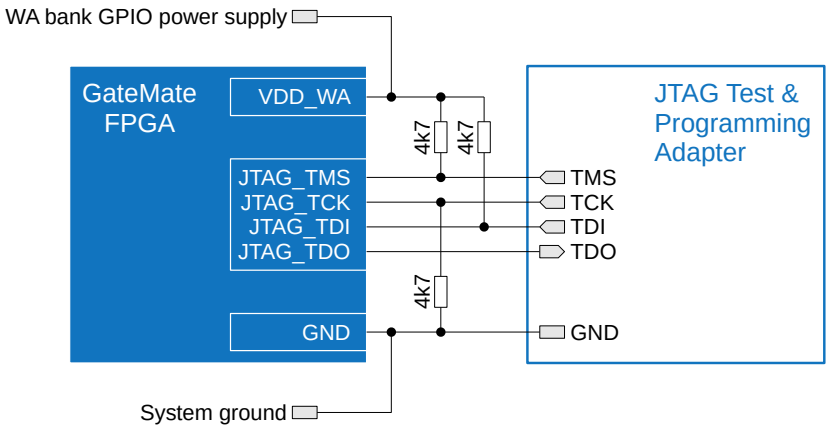


Figure 2.7: JTAG connection scheme

Four signals support the TAP according to the IEEE 1149.1-2001 requirements:

Test Mode Select (TMS) controls the TAP controller state machine. The signal is sampled on the rising edge of TCK.

Test Clock (TCK) provides a clock signal. The mode TMS is sampled on the rising edge of TCK. Input data on TDI is shifted into the instruction (IR) or data (DR) registers on the rising edge of TCK. Output data on TDO is shifted out on the falling edge of TCK.

Test Data Input (TDI) is the serial input to all JTAG instruction (IR) and data (DR) registers depending on the sequence previously applied at TMS.

Test Data Output (TDO) is the serial output for all JTAG instruction (IR) and data (DR) registers depending on the sequence previously applied at TMS.

[2] See Section 3.3.1 for more ways to load the FPGA configuration.

JTAG Usage

The TMS signal is evaluated on the rising edge of TCK. Its value decides the next state into which the Finite State Machine (FSM) changes. The FSM states are shown in Figure 2.8.

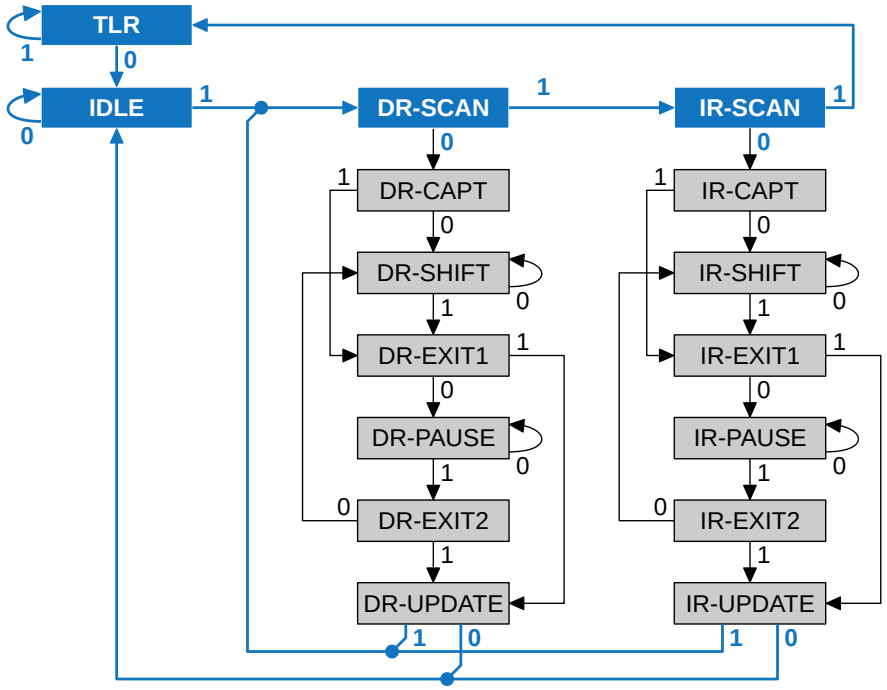


Figure 2.8: Finite State Machine of the JTAG interface

The typical procedure is to first load the instruction and then write or read the data associated with this instruction. The JTAG host controller must drive the signals TCK, TMS and TDI in the according way as illustrated shown in Figures 2.9 and 2.10. The instruction register (IR) size is 6 bits. Instructions have to be shifted into the JTAG TAP with LSB first.

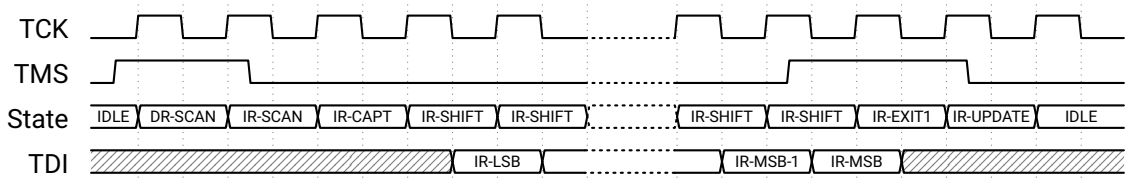


Figure 2.9: JTAG instruction shift

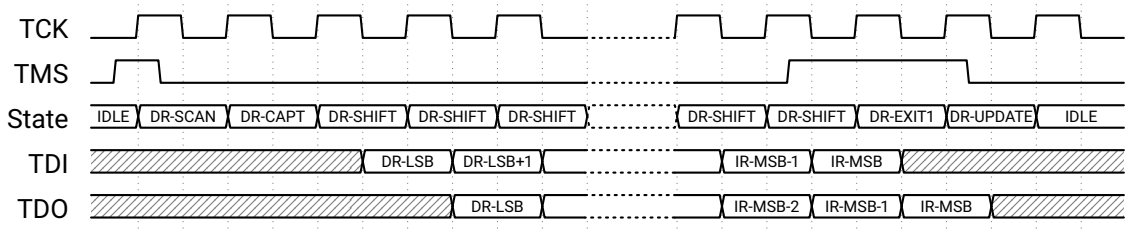


Figure 2.10: JTAG data shift

Common JTAG Instructions

According to IEEE 1149.1-2001, the IDCODE register is optional and thus only contains a placeholder value.

The bypass mode connects the JTAG TDI and TDO lines via a single-bit pass-through register and allows the testing of other devices in a JTAG chain. In bypass mode, the TDI input is delayed by one clock cycle before outputting to TDO.

Instruction:	**0x3F**	**JTAG_BYPASS**	Shift width:	1 bit
		Configure JTAG bypass mode. The bypass register is set to zero when the TAP controller is in the CAPTURE-DR state.		
Shift-in bits:	–			
Shift-out bits:	–	Input bit pattern delayed by one clock cycle.		

JTAG Configuration

The JTAG interface can be used to configure the FPGA as well. The configuration stream must have a length of a multiple of 8 bits. The JTAG TAP automatically combines every 8 received bits to a single byte and forwards it to the configuration processing block. JTAG configuration works independently of the configuration mode pins CFG_MD[3:0]. However, in order to avoid conflicts, the pins should be set to 0xC. Further information on the configuration modes can be found in Section 3.3.1. The configuration byte stream is in big endian while the bytes are in little endian. If the configuration bank is not switched to user I/O after configuration, the status can be read as usual using the CONFIG_DONE and CONFIG_FAILED pins.

Instruction:	**0x06**	**JTAG_CONFIGURE**	Shift width:	M · 8 bits
		Configure the FPGA with a configuration stream of M bytes.		
Shift-in bits:	M·[7:0]	The configuration byte stream is in big endian while the bytes are in little endian.		
Shift-out bits:	M·[7:0]	Don't care		

Access to SPI Bus

The SPI bus can be accessed directly using the JTAG interface. This way, bitstreams may be written to an external dataflash without having to access the bus with additional hardware. There are two ways of acting on the SPI bus via the JTAG interface:

- Full access to SPI interface using JTAG-SPI bypass
- Utilities to access external SPI dataflashes

Figure 2.11 illustrates the wiring between a single FPGA connected to an external SPI component and a JTAG host controller.

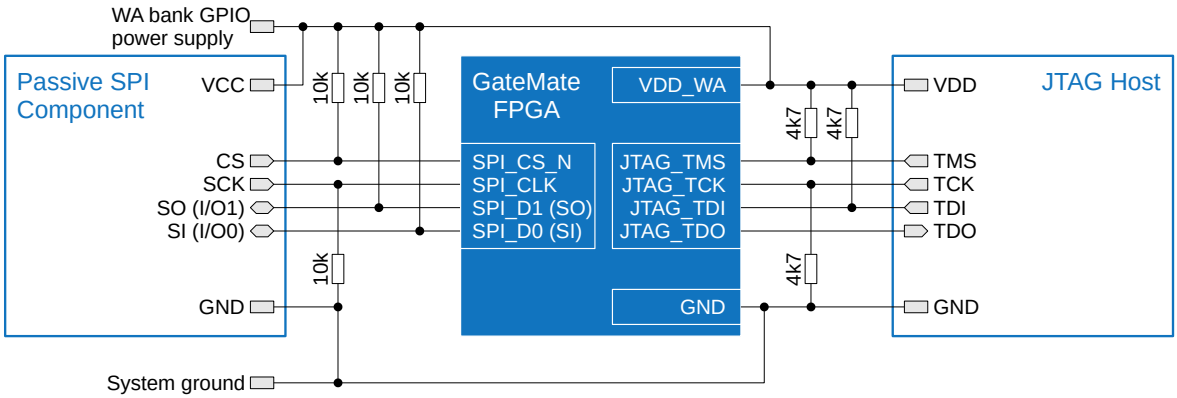


Figure 2.11: JTAG-SPI bypass connection scheme.

In active JTAG-SPI bypass mode all JTAG signals TCK, TDI and TDO are directly connected to the SPI pins. Due to the direct connection of TCK and SCK, the clock frequency on the SPI bus is the same as that of the JTAG clock. Before accessing the SPI bus via active bypass it must be ensured that the SPI bus is not busy, e.g. by checking the JTAG_GET_SPI_BUSY command. There is no restriction on the length of a SPI transfer. The SPI bypass mode works for all four SPI modes (CPOL, CPHA), but only in single-IO mode as set via the SPI configuration mode pins CFG_MD[1:0] described in Table 3.1.

With each DR-shift one bit at TDI is shifted into the JTAG TAP and directly forwarded to SPI_D0 output. Simultaneously, one bit is shifted into SPI_D1 input. The bit will be available on the JTAG output TDO at the next falling edge of TCK. There must be one more DR-shift cycle than bits to transmit. The last bit shifted into the TAP will not be shifted to the external SPI device and the very first bit shifted out of the TAP is an invalid bit.

Instruction:	**0x05**	**JTAG_SPI_BYPASS**	Shift width:	N bits
		Active access to SPI bus via JTAG.		
Shift-in bits:	–	Data to be shifted into external SPI device		
Shift-out bits:	–	Data to be shifted from external SPI device		

Figure 2.12 illustrates the function of the active JTAG-SPI bypass in all four available SPI single-IO modes.

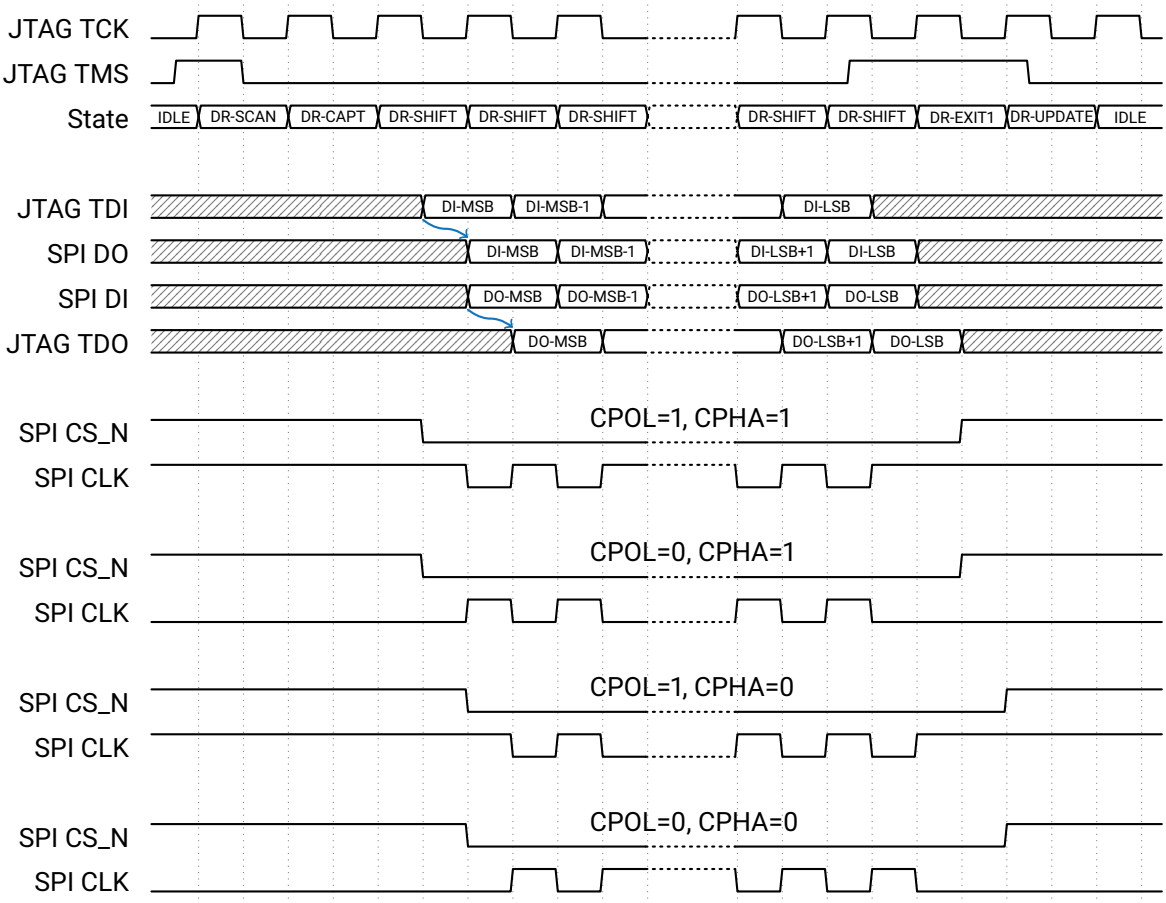


Figure 2.12: Timing diagram of the JTAG-SPI bypass mode.

Access to SPI Dataflash

The JTAG_SPI_ACCESS provides access to external SPI dataflashes with a conventional command structure, in which command, mode, address and bidirectional data bits can be exchanged. In order to support a wide range of SPI dataflashes, all command fields can be adjusted. First, the instructions have to be shifted into the JTAG TAP register with LSB first. The actual SPI transfer start as soon as the DR-UPDATE state in JTAG FSM was executed. In contrast to the JTAG-SPI bypass mode, the SPI clock rate is determined by the clock frequency of the internal oscillator.

The JTAG_SPI_ACCESS command allows transmissions of at most 32 TX data bits. The number of TX bits must be specified and TX words must be set to 1. Moreover, up to 32 bits of data can be received via the JTAG_GET_SPI_RXDATA command after execution of JTAG_SPI_ACCESS with the corresponding read command and number of bits to be received. The RX data flag is only valid if the read command was previously executed.

In order to transfer more than 32 bits, the JTAG TAP contains a register file of 256 bytes that can be filled via the JTAG command JTAG_FLASH_PAGE_PREP before the execution of an JTAG_SPI_ACCESS with a number of TX data word greater than 1. The total number of bits to be transmitted from the register file is the product of TX data bits times TX data words.

The internal file register for transmitting more than 32 bits contains a 256 byte memory that can be pre-loaded by using the JTAG_FLASH_PAGE_PREP command. An entire 32-bit word must always be followed by a DR-UPDATE. This procedure can be repeated up to 64 times to load a full page of 256 bytes. As long as no test logic reset command is executed, the file register retains its content, but can be overwritten. Its address counter is set to zero each time the command is called. For correct functioning it is necessary to transmit the first word of the register file again during JTAG_SPI_ACCESS.

It is highly recommended to verify the idle state before accessing the SPI bus by executing the JTAG_GET_SPI_BUSY command.

Instruction:	**0x01**	**JTAG_SPI_ACCESS** Shift width: 119 bits Access external SPI dataflash via JTAG.
Shift-in bits:	[118:115]	Number of **command bits** to be shifted to SPI dataflash
	[114:109]	Number of **address bits** to be shifted to SPI dataflash
	[108:105]	Number of **mode bits** to be shifted to SPI dataflash
	[104: 99]	Number of **TX data bits** to be shifted to SPI dataflash
	[98: 92]	Number of **TX data words** to be shifted to SPI dataflash
	[91: 86]	Number of **dummy cycles** to be shifted to SPI dataflash
	[85: 80]	Number of **RX data bits** to be shifted to SPI dataflash
	[79: 72]	**Command** to be transferred to SPI dataflash
	[71: 64]	**Mode** to be transferred to SPI dataflash
	[63: 32]	**Address** to be transferred to SPI dataflash
	[31: 0]	**TX data** word to be transferred to SPI dataflash. The byte stream is in big endian while the bytes are in little endian. Must contain the first word when transmitting data using the JTAG_FLASH_PAGE_PREP command.
Shift-out bits:	[118: 0]	Don't care

Instruction:	**0x02**	**JTAG_FLASH_PAGE_PREP** Shift width: K · 32 bits Load data for writing up to one 256-byte flash page into FPGA. Must be followed by 0x1 – JTAG_SPI_ACCESS.
Shift-in bits:	K·[31:0]	K 32-bit words to be written to one page in the SPI dataflash with $1 \leq K \leq 64$. Each word must be followed by a DR-UPDATE. The byte stream is in big endian while the bytes are in little endian.
Shift-out bits:	K·[31:0]	Don't care

Instruction:	**0x03**	**JTAG_GET_SPI_RXDATA**	Shift width:	33 bits
		Get data read from SPI dataflash via after 0x1 – JTAG_SPI_ACCESS.		
Shift-in bits:	[32:0]	Don't care		
Shift-out bits:	[32:1]	RX data read from SPI dataflash		
	[0]	RX data valid flag. Only true if the preceding command was JTAG_SPI_ACCESS.		

Instruction:	**0x04**	**JTAG_GET_SPI_BUSY**	Shift width:	1 bit
		Checks if the SPI controller is busy. Recommended to use before any SPI access.		
Shift-in bits:	[0:0]	Don't care		
Shift-out bits:	[0:0]	Obtain SPI bus state: 0b0: SPI idle 0b1: SPI busy		

Boundary-Scan Test

A JTAG IEEE 1149.1-2001 compliant boundary-scan allows to test pin connections without physically probing pins by shifting a boundary-scan register that is composed of cells connected between the FPGA logic and IOs.

The SAMPLE instruction allows to take a snapshot of the input and output signal states without interfering with the device pads. The snapshot is taken on the rising edge of TCK in the DR-CAPT state. The data can then be accessed by shifting through the TDO output.

The PRELOAD instruction allows scanning of the boundary-scan register without causing interference to the FPGA logic. It allows an initial pattern to be shifted into the boundary-scan register cells.

Both instructions SAMPLE and PRELOAD are combined in the JTAG_SAMPLE_PRELOAD command. While preloading the boundary-scan register cells by shifting data into the input TDI the sample results are shifted through the TDO output.

The EXTEST instruction allows access to the external circuitry through the device pads. The boundary-scan register cells connected to the output pins are used to apply test stimuli, while those at the input pins capture external signals from the device pads.

The boundary-scan register length is 317 bits. To enable boundary-scan test, the testmode register must be set first by using the JTAG_SET_TESTMODE instruction. Please refer to the official BSDL files for more information.

Instruction:	**0x20**	**JTAG_SET_TESTMODE**	Shift width:	2 bits
		Set value in testmode register.		
Shift-in bits:	[1:0]	Testmode to be set: 0b00: Disable testmode 0b01: reserved 0b10: reserved 0b11: Boundary-Scan Test		
Shift-out bits:	[1:0]	Don't care		

Instruction:	**0x3D**	**JTAG_SAMPLE_PRELOAD**	Shift width:	317 bits
		Boundary-scan SAMPLE and PRELOAD instruction. The register 0x20 – Set Testmode must be set to 0b11.		
Shift-in bits:	[316:0]	Preload value for boundary-scan shift register.		
Shift-out bits:	[316:0]	Sample value of boundary-scan shift register.		

Instruction:	**0x3E**	**JTAG_EXTEST**	Shift width:	317 bits
		Boundary-scan EXTEST instruction. The register 0x20 – Set Testmode must be set to 0b11.		
Shift-in bits:	[316:0]	Don't care.		
Shift-out bits:	[316:0]	Value of boundary-scan shift register.		

2.3 Routing Structure

All functional elements of the GateMate™ FPGA are interconnected by the routing structure. This consist of so called *Switch Boxes* mainly and is supplemented by additional *Input Multiplexers* and *Output Multiplexers*. All Switch Boxes are arranged in a matrix $(x, y) = (-1, -1) \ldots (162, 130)$ as shown in Figure 2.13.

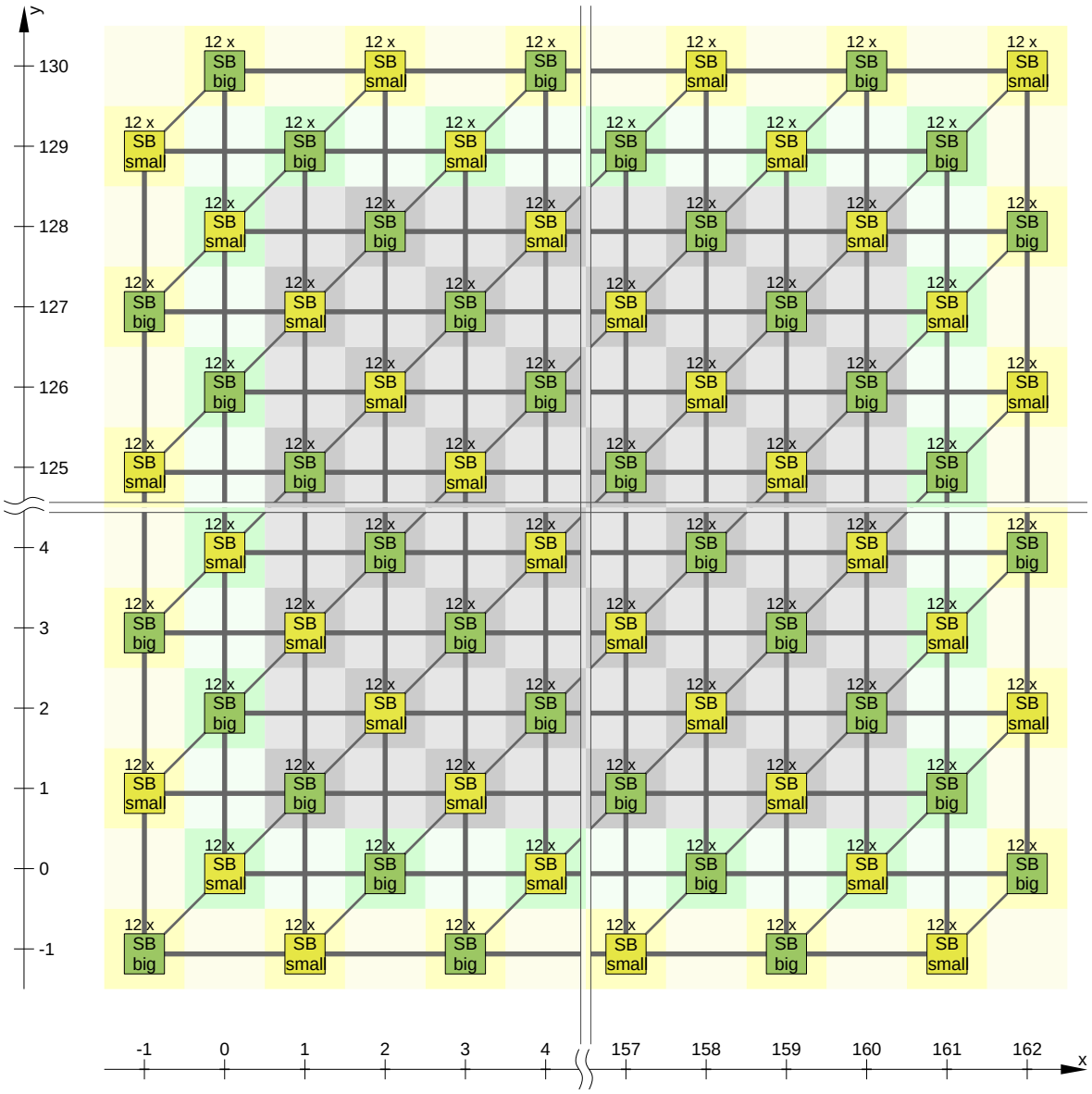


Figure 2.13: Switch Box interconnection scheme

Every second coordinate is populated by Switch Boxes. There are two types of Switch Boxes which differ only in the span they can forward into the routing matrix. Mainly, Switch Boxes connect bidirectional to other Switch Boxes in horizontal and vertical direction. Big Switch Boxes connect six other Switch Boxes in every direction and Small Switch Boxes only two others. Direction change from vertical to horizontal and vice versa is possible. From an arbitrary point (x_n, y_m) all coordinates in row y_m or column x_n are reachable.

Furthermore, diagonal neighbors to upper right $(x_n + 1, y_m + 1)$ and lower left $(x_n - 1, y_m - 1)$ coordinate can be connected. This allows change to neighbored rows or columns.

(a) Small Switch Box **(b)** Big Switch Box

Figure 2.14: Small and Big Switch Boxes

Signals are feed into the routing structure either directly from CPE outputs or, for more flexibility, through so called *Output Multiplexers*. GPIO input signals are connected to the Switch Boxes as well.

Switch Boxes decouple all signals through so called *Input Multiplexers* to the CPEs.

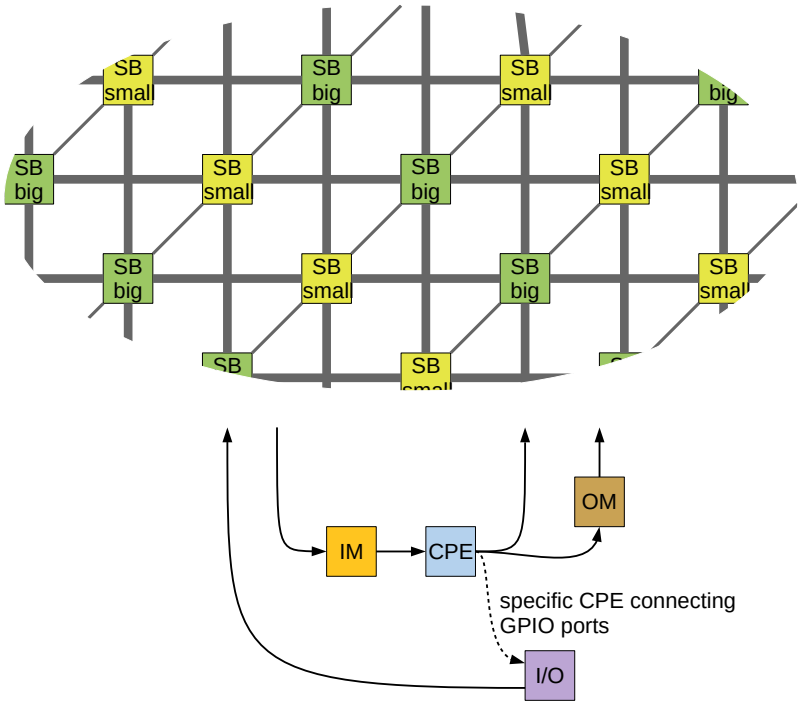


Figure 2.15: CPE connection into the routing structure

2.4 Power Supply

The GateMate™ FPGA has a single power supply for the chip core and GPIO power supply pins for each GPIO bank. There are some more power pins for dedicated parts of the FPGA:

VDD Core supply voltage.

VDD_CLK Supply voltage for the input pins SER_CLK, SER_CLK_N, RST_N and POR_ADJ.

VDD_PLL Supply pin for PLLs.

VDD_SER Supply pin for the SerDes module.

VDD_SER_PLL Supply pin for the SerDes PLL.

VDD_EA, VDD_EB, VDD_NA, VDD_NB, VDD_SA, VDD_SB, VDD_WA, VDD_WB and VDD_WC Supply pins for GPIO banks.

Core voltage can be chosen to select the application mode:

Low power mode	$\mathrm{VDD} = 0.9\,\mathrm{V} \pm 50\,\mathrm{mV}$
Economy mode	$\mathrm{VDD} = 1.0\,\mathrm{V} \pm 50\,\mathrm{mV}$
Speed mode	$\mathrm{VDD} = 1.1\,\mathrm{V} \pm 50\,\mathrm{mV}$

VDD_PLL has the same core voltage range. VDD_SER and VDD_SER_PLL have a voltage range from $1.0\,\mathrm{V} \pm 50\,\mathrm{mV}$ to $1.1\,\mathrm{V} \pm 50\,\mathrm{mV}$. All voltages should be connected to noise filters.

The voltage range of GPIO banks and VDD_CLK can be set from 1.1 V to 2.7 V.

Chapter 3

Workflow and Hardware Setup

3.1 FPGA Workflow

The workflow from the design to the GateMate™ FPGA configuration file is shown in Figure 3.1. Typically, a software framework for FPGA design is already installed at the customer side.

The design flow starts with HDL source generation with direct HDL implementation or any higher-level programming language.

Logic synthesis generates a Verilog netlist containing an abstraction of the HDL sources using low-level primitives. The resulting Verilog netlist is then passed to EasyConvert™to convert the primitives netlist to the data format required by the GateMate™ FPGA Place and Route software.

In the first step of Place and Route a procedure for speed or area optimization is executed and these results are passed to the mapping module. After placement and routing, the static timing analysis (STA) might lead to further optimization steps. This design closure is an iterative process of constraint-driven re-placement and re-routing steps to finally achieve user requirements.

Finally, a Verilog netlist and SDF file are written. These can be used by the customer testbench for post implementation simulation. The FPGA configuration file is written as well. This can either be written into a Flash memory or directly into the FPGA configuration controller.

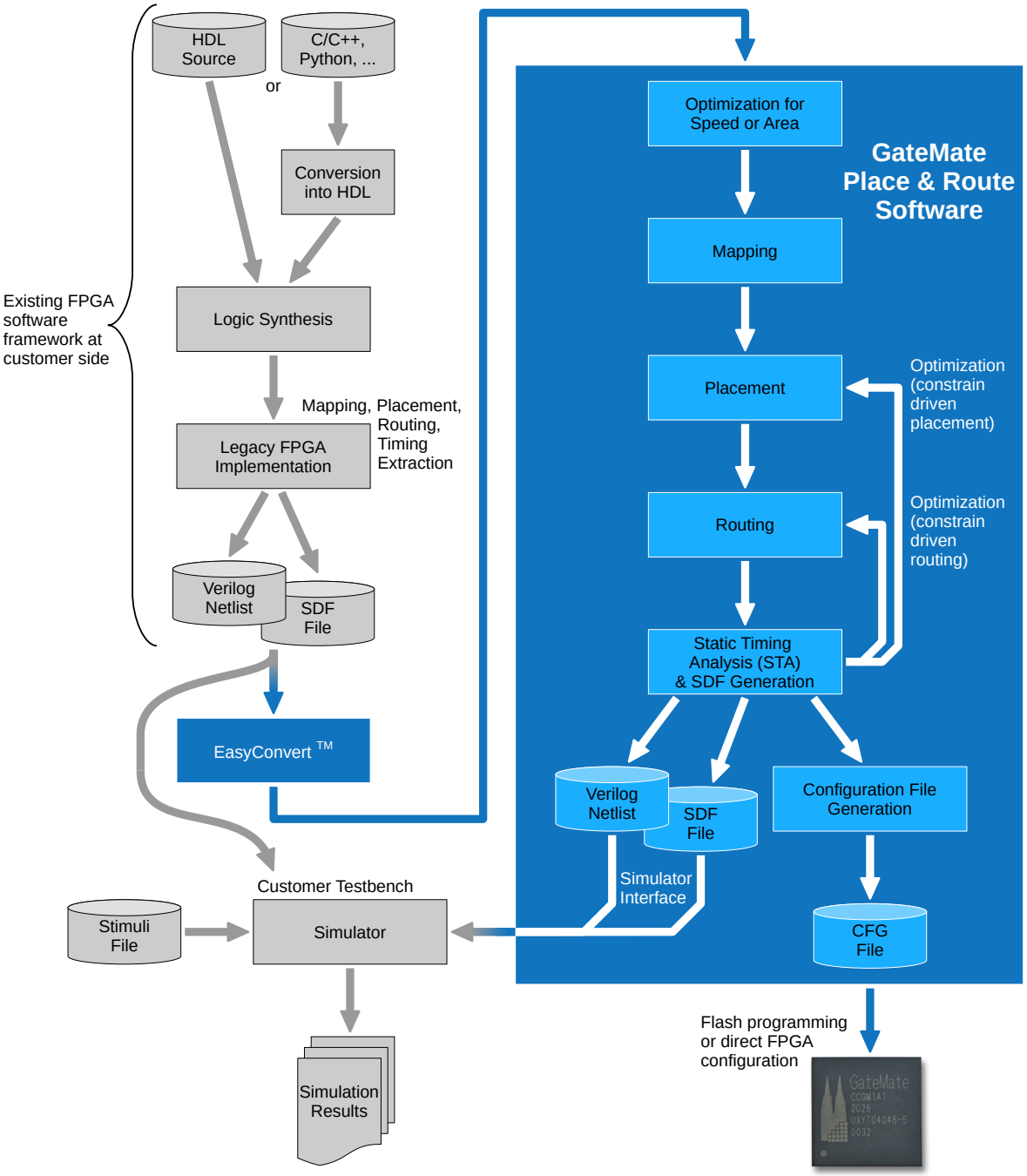


Figure 3.1: GateMate™ FPGA Workflow

3.2 Hardware Setup

3.2.1 Power-on Reset

The GateMate™ FPGA has a reset pin RST_N with internal pull-up resistor. Signal polarity is active low.

The power-on reset module (POR) ensures a safe reset as soon as VDD_CORE and VDD_CLK supply voltages are stable. The POR has the following features:

- Safe reset generation after stable VDD_CORE and VDD_CLK voltages.
- Voltage drop detection for VDD_CORE and VDD_CLK voltages.
- Adjustable VDD_CLK voltage threshold.
- No restrictions concerning order of voltage switch-on.
- Temperature compensated voltage reference.

The POR must be enabled with POR_EN signal:

0: Reset state depends only on the RST_N signal.

1: FPGA changes from reset to operation state when RST_N is 1 and POR indicates VDD_CORE and VDD_CLK voltages are stable.

POR_EN has an internal pull-down resistor and can be left open if not used.

3.3 Configuration Procedure

3.3.1 Configuration Modes

The GateMate™ FPGA can be configured from four different sources:

1. The configuration process is controlled by the FPGA itself where it acts in SPI active mode, e.g. after reset. Configuration data is read from an external Flash memory.
2. A processor unit operating as SPI in active mode feeds the configuration data to the FPGA which acts in SPI passive mode.
3. The JTAG interface of the FPGA connects to a JTAG test & programming adapter.
4. User configured logic of the FPGA can feed configuration data to the FPGA itself.

With the rising edge of the reset signal the configuration mode setup at pins CFG_MD[3:0] gets captured as shown in Table 3.1. These pins must always be connected to either GND or VDD_WA.

Table 3.1: Configuration mode setup

CFG_MD[3:0]*		Configuration mode	
0x0	0000	SPI Active Mode	CPOL = 0, CPHA = 0
0x1	0001	SPI Active Mode	CPOL = 0, CPHA = 1
0x2	0010	SPI Active Mode	CPOL = 1, CPHA = 0
0x3	0011	SPI Active Mode	CPOL = 1, CPHA = 1
0x4	0100	SPI Passive Mode	CPOL = 0, CPHA = 0
0x5	0101	SPI Passive Mode	CPOL = 0, CPHA = 1
0x6	0110	SPI Passive Mode	CPOL = 1, CPHA = 0
0x7	0111	SPI Passive Mode	CPOL = 1, CPHA = 1
0xC	1100	JTAG	

* Modes not mentioned in the list may not be selected and lead to a malfunction of the device.

3.3.2 SPI Configuration

The SPI bus of GateMate™ FPGA can be used to load the configuration either in SPI active or passive mode. Both modes use the same pins. Typically, a Flash memory is connected to the SPI bus to provide the configuration data. Figure 3.2 shows a typical circuitry which uses only single-I/O or dual-I/O mode. The Flash signals I/O_2 and I/O_3 can have further functions depending on the used device and should have pull-up resistors in most cases.

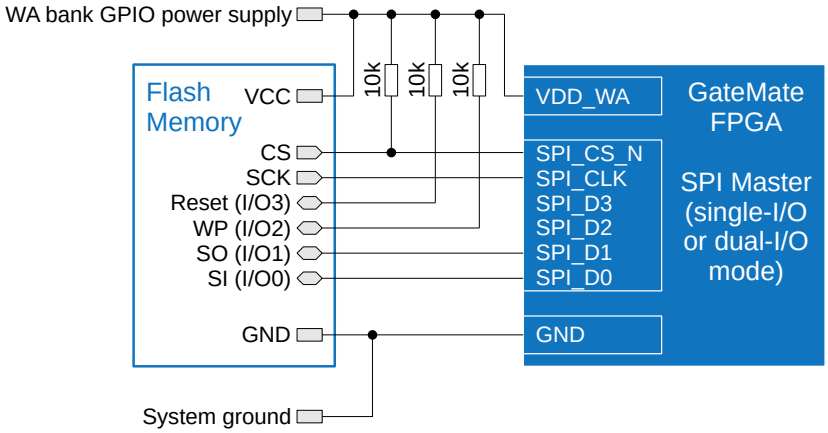


Figure 3.2: Configuration data from Flash memory in SPI single-I/O or dual-I/O mode

When SPI quad-I/O mode is used, Figure 3.3 shows the typical circuitry. Now, all data lines are connected and no pull-up resistors are required. Depending on the used Flash device, pull-up resistors are recommended at I/O_2 and I/O_3 signals.

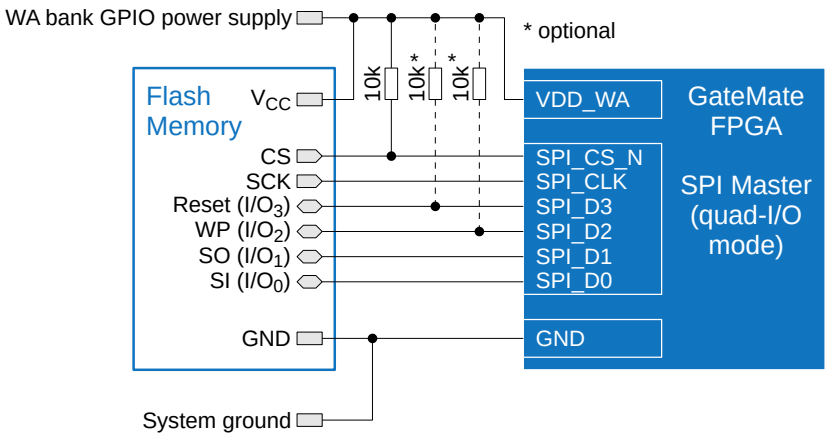


Figure 3.3: Configuration data from Flash memory in SPI quad-I/O mode

Finally, Figure 3.4 shows two GateMate™ FPGAs using the same SPI configuration data source. This is done using the SPI forward function. Only SPI single-I/O mode can be used in this case.

The GateMate™ FPGA is able to load it's configuration automatically after reset. No external clock is required. When SPI active Mode is set up, the rising edge of RST_N starts reading the configuration data from the Flash memory. The internal clock needs no external reference clock. If an external reference clock is available, the configuration stream can set up a PLL to any configuration clock frequency.

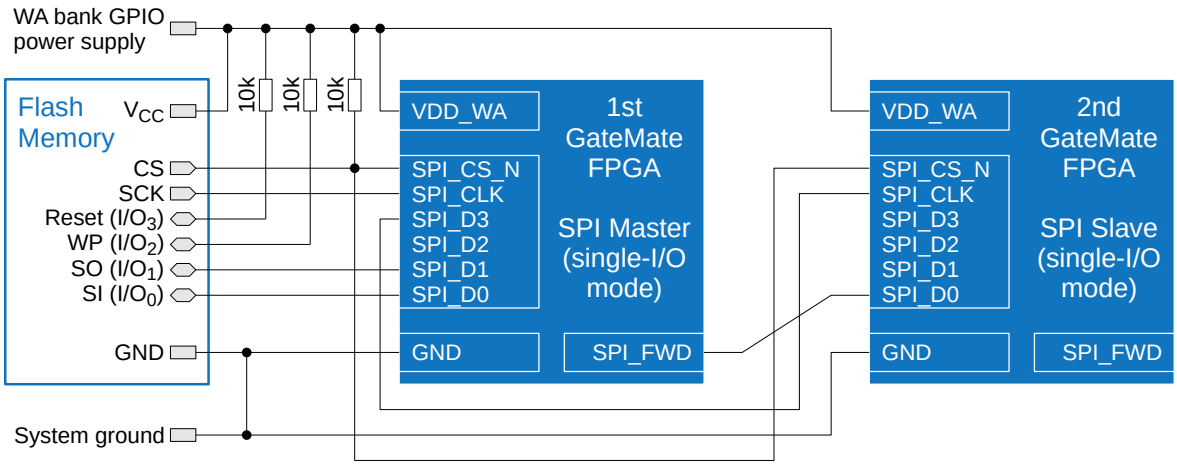


Figure 3.4: Configuration data stream from a single Flash memory

When operating in SPI active mode, error detection and correction is done automatically. In this case a short CFG_FAILED_N pulse is given and the failed configuration sequence will be re-read.

When operating in SPI passive mode, the configuration controller can only react on the incoming data stream while the external device in SPI active mode is in control over the bus.

The signals CFG_DONE and CFG_FAILED_N indicate the end of the configuration process as described in Table 3.2.

Table 3.2: Configuration process state

CFG_DONE	CFG_FAILED_N	Description
0	0 *	Invalid state
0	1	Configuration procedure is still in progress
1	0 *	Unable to load configuration. Data stream might be invalid
1	1 **	Configuration successful

* Only long CFG_FAILED_N pulses. Short pulses have no significance.
** CFG_DONE might not be retrievable in case the configuration bank is set to user mode.

CFG_FAILED_N requires a pull-up resistor. If multiple FPGAs are configured from a single Flash memory, all CFG_FAILED_N have to be connected as shown in Figure 3.5.

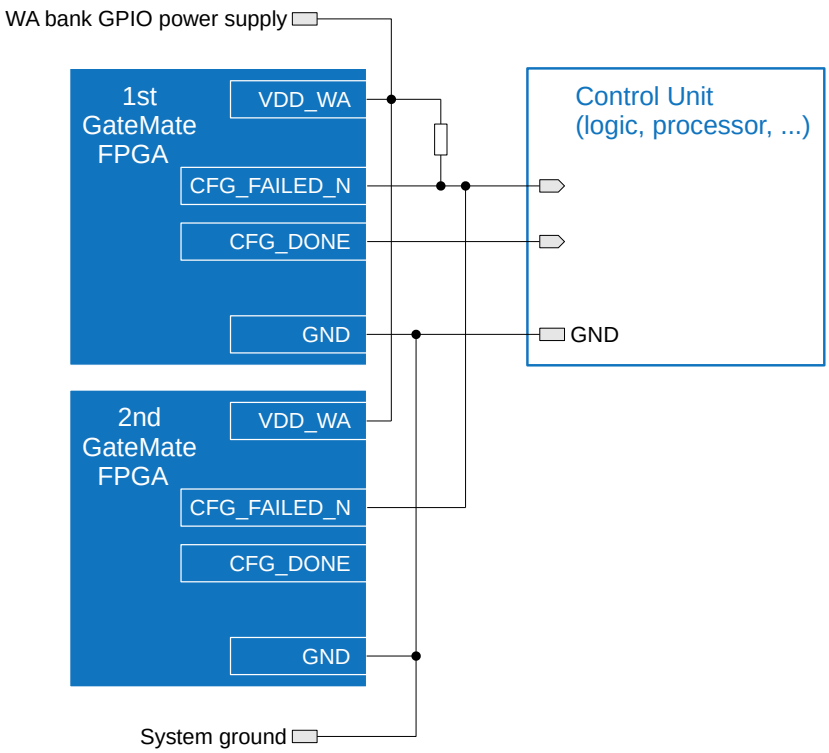


Figure 3.5: Connection of CFG_DONE and CFG_FAILED_N signals in multi-FPGA applications with a single Flash memory

Chapter 4

Electrical Characteristics

Table 4.1: Absolute maximum ratings

Symbol	Min	Typ	Max	Unit	Description
	-40		125	°C	Junction temperature
VDD_{IO}			2.75	V	I/O supply voltage (GPIO banks and VDD_CLK)
VDD_{core}			1.20	V	Core supply voltage (VDD_CORE, VDD_PLL, VDD_SER and VDD_SER_PLL)

Table 4.2: Operation range

Symbol	Min	Typ	Max	Unit	Description
	-20		85	°C	Operating temperature
$VDD_{low\ power}$	0.85	0.9	0.95	V	Core supply voltage (VDD_CORE, VDD_PLL) in low power mode
$VDD_{economy}$	0.95	1.0	1.05	V	Core supply voltage (VDD_CORE, VDD_PLL) in economy mode
VDD_{speed}	1.05	1.1	1.15	V	Core supply voltage (VDD_CORE, VDD_PLL) in speed mode
VDD_{SER}	0.95		1.15	V	SerDes supply voltage (VDD_SER)
VDD_{SER_PLL}	0.95		1.15	V	SerDes PLL supply voltage (VDD_SER_PLL)

Table 4.3: GPIO characteristics in single-ended mode

Symbol	Min	Typ	Max	Unit	Description
VDD_{IO}	1.1		2.7	V	I/O supply voltage (GPIO banks and VDD_CLK)
V_{IN}	-0.4		VDD_{IO}+0 4	V	Input voltage range
Schmitt-trigger function disabled:					
V_{IH}	0.43		0.51	VDD_{IO}	Input high threshold voltage
V_{IL}	0.45		0.51	VDD_{IO}	Input low threshold voltage
V_{HYST}		0		V	Hysteresis
Schmitt-trigger function enabled:					
V_{IH}	0.61		0.67	VDD_{IO}	Input high threshold voltage
V_{IL}	0.31		0.39	VDD_{IO}	Input low threshold voltage
V_{HYST}	0.26		0.33	VDD_{IO}	Hysteresis
Output driver disabled:					
I_{IL}			1	µA	Input pin current when driven active low
I_{IH}			1	µA	Input pin current when driven active high
R_{PU}		50		kΩ	Pull-up resistance
R_{PD}		50		kΩ	Pull-down resistance
$I_{DD, max}$			158	µA	Maximum supply current at GPIO input at transition point

Table 4.4: GPIO characteristics in LVDS mode

Symbol	Min	Typ	Max	Unit	Description
VDD_{IO}	1.62		2.75	V	I/O supply voltage
I_{out}		3.2		mA	Output current when LVDS output current boost is set to nominal current
I_{out}		6.4		mA	Output current when LVDS output current boost is set to increased output current
VCM_{TX}		$VDD_{IO}/2$		V	
R_{term}	90	100	130	Ω	

Table 4.5: PLL characteristics

Symbol	Min	Typ	Max	Unit	Description
t_{lock}			1100	Number of ref. clock cycles	Lock in time (coarse tune, fast-lock mode) with lock detection
t_{lock}			57	Number of ref. clock cycles	Lock in time (coarse tune, fast-lock mode) without lock detection
$f_{DCO,\,low\ power}$	500		1000	MHz	DCO frequency in low power mode
$f_{DCO,\,economy}$	1000		2000	MHz	DCO frequency in economy mode
$f_{DCO,\,speed}$	1250		2500	MHz	DCO frequency in speed mode
$T_{DCO,\,step}$			10	ps	DCO period tuning step size

Chapter 5

Pinout

5.1 CCGM1A1 324-Ball FBGA Pinout

The GateMate™ FPGA CCGM1A1 has a 18×18 pin 0.8 mm Fine-pitch Ball Grid Array (FBGA) package with 324 balls.

GPIO ball arrangement is defined for 4-layer PCB layout with only 2 signal layers.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
A	GND	VDD_WC	IO_NA_A0	IO_NA_A1	VDD_NA	IO_NA_A4	GND	IO_NA_A7	IO_NB_B0	GND	IO_NB_B2	IO_NB_B4	GND	IO_NB_B7	IO_EB_B8	VDD_EB	IO_EB_B5	GND	A
B	IO_WC_A8	IO_WC_B8	IO_NA_B0	IO_NA_B1	IO_NA_A2	IO_NA_B4	VDD_NA	IO_NA_B7	IO_NB_A0	VDD_NB	IO_NB_A2	IO_NB_A4	VDD_NB	IO_NB_A7	IO_EB_A8	GND	IO_EB_A5	VDD_EB	B
C	GND	VDD_WC	IO_WC_A7	IO_WC_B7	IO_NA_B2	IO_NA_A3	IO_NA_A5	IO_NA_A6	IO_NA_A8	IO_NB_B1	IO_NB_B3	IO_NB_B5	IO_NB_B6	IO_NB_B8	IO_EB_B7	IO_EB_B6	IO_EB_B4	IO_EB_A4	C
D	IO_WC_A5	IO_WC_B5	IO_WC_A6	IO_WC_B6	VDD_WC	IO_NA_B3	IO_NA_B5	IO_NA_B6	IO_NA_B8	IO_NB_A1	IO_NB_A3	IO_NB_A5	IO_NB_A6	IO_NB_A8	IO_EB_A7	IO_EB_A6	IO_EB_B2	IO_EB_A2	D
E	IO_WC_A3	IO_WC_B3	IO_WC_A4	IO_WC_B4	GND	VDD_NA	GND	VDD_NA	GND	VDD_NB	GND	VDD_NB	GND	VDD_EB	IO_EB_B3	IO_EB_A3	VDD_EB	GND	E
F	GND	VDD_WC	IO_WC_A2	IO_WC_B2	VDD_WC	GND	VDD_NA	GND	VDD	GND	VDD_NB	GND	VDD_EB	GND	IO_EB_B1	IO_EB_A1	IO_EB_B0	IO_EB_A0	F
G	IO_WC_A0	IO_WC_B0	IO_WC_A1	IO_WC_B1	GND	VDD	GND	VDD	GND	VDD	GND	VDD	GND	VDD_EA	IO_EA_B8	IO_EA_A8	IO_EA_B7	IO_EA_A7	G
H	IO_WB_A7	IO_WB_B7	IO_WB_A8	IO_WB_B8	VDD_WB	GND	VDD	GND	VDD	GND	VDD	GND	VDD	GND	IO_EA_B6	IO_EA_A6	VDD_EA	GND	H
J	GND	VDD_WB	IO_WB_A6	IO_WB_B6	GND	VDD	GND	VDD	GND	VDD	GND	VDD	GND	VDD_EA	IO_EA_B5	IO_EA_A5	IO_EA_B4	IO_EA_A4	J
K	IO_WB_A5	IO_WB_B5	IO_WB_A4	IO_WB_B4	VDD_WB	GND	VDD	GND	VDD	GND	VDD	GND	VDD	GND	IO_EA_B3	IO_EA_A3	IO_EA_B2	IO_EA_A2	K
L	IO_WB_A3	IO_WB_B3	IO_WB_A2	IO_WB_B2	GND	VDD	GND	VDD	GND	VDD	GND	VDD	GND	VDD_EA	IO_EA_B1	IO_EA_A1	VDD_EA	GND	L
M	GND	VDD_WB	IO_WB_A1	IO_WB_B1	VDD_WB	GND	VDD	GND	VDD	GND	VDD	GND	VDD	IO_EA_B0	IO_EA_A0	GND	IO_SB_A3	IO_SB_B3	M
N	IO_WB_A0	IO_WB_B0	SPI_CS_N	SPI_CLK	VDD_WA	VDD	GND	VDD	GND	VDD	VDD_SB	GND	VDD_SB	IO_SB_A8	IO_SB_B8	N.C.	GND	VDD_SB	N
P	SPI_D1	SPI_D0	VDD_WA	GND	VDD_WA	VDD_SA	GND	VDD_SA	GND	VDD_SA	IO_SB_A4	IO_SB_A7	IO_SB_B7	IO_SB_A6	IO_SB_B6	VDD_PLL	IO_SB_A2	IO_SB_B2	P
R	SPI_D3	SPI_D2	JTAG_TCK	SPI_FWD	CFG_MD0	IO_SA_A1	IO_SA_A2	IO_SA_A4	IO_SA_A6	IO_SA_A7	IO_SB_B4	GND	IO_SB_A5	IO_SB_B5	VDD_SB	GND	IO_SB_A1	IO_SB_B1	R
T	VDD_WA	JTAG_TDI	JTAG_TMS	GND	CFG_MD1	IO_SA_B1	IO_SA_B2	IO_SA_B4	IO_SA_B6	IO_SA_B7	GND	SER_CLK	SER_CLK_N	VDD_CLK	RST_N	VDD_SER_PLL	GND	VDD_SB	T
U	POR_EN	JTAG_TDO	VDD_WA	CFG_MD2	IO_SA_A0	VDD_SA	IO_SA_A3	IO_SA_A5	VDD_SA	IO_SA_A8	SER_RX_P	VDD_SER	SER_TX_P	GND	GND	GND	IO_SB_A0	IO_SB_B0	U
V	GND	CFG_FAILED_	CFG_DONE	CFG_MD3	IO_SA_B0	GND	IO_SA_B3	IO_SA_B5	GND	IO_SA_B8	SER_RX_N	SER_RTERM	SER_TX_N	GND	POR_ADJ	GND	VDD_SER	GND	V
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	

Figure 5.1: CCGM1A1 FBGA pinout

	1	2	3	4	5
N	IO_WB_A0	IO_WB_B0	IO_WA_A8	IO_WA_B8	VDD_WA
P	IO_WA_A7	IO_WA_B7	VDD_WA	GND	VDD_WA
R	IO_WA_A6	IO_WA_B6	IO_WA_A5	IO_WA_B5	IO_WA_A0
T	VDD_WA	IO_WA_A4	IO_WA_B4	GND	IO_WA_B0
U	IO_WA_A3	IO_WA_B3	VDD_WA	IO_WA_A1	IO_SA_A0
V	GND	IO_WA_A2	IO_WA_B2	IO_WA_B1	IO_SA_B0

Figure 5.2: Configuration pads of CCGM1A1 BGA pinout for use as GPIO after configuration

5.2 CCGM1A1 Pinout Description

Table 5.1: Pad Types

Color	Pad Types	Pads
	North GPIO banks NA and NB Each bank has 18 GPIO signals and both have separate power supply balls and can be driven individually. GPIO signals are grouped by pairs and each pair can get configured to be either two single ended or a differential pair signal.	36
	East GPIO banks EA and EB Each bank has 18 GPIO signals and both have separate power supply balls and can be driven individually. GPIO signals are grouped by pairs and each pair can get configured to be either two single ended or a differential pair signal.	36
	South GPIO banks SA and SB Each bank has 18 GPIO signals and both have separate power supply balls and can be driven individually. GPIO signals are grouped by pairs and each pair can get configured to be either two single ended or a differential pair signal.	36
	Configuration pads or west GPIO bank WA alternatively The configuration interface consists of 18 pads. After configuration procedure, these balls can be configured to be normal GPIO signals. They are grouped by pairs and each pair can get configured to be either two single ended or a differential pair signal.	18
	West GPIO banks WB and WC Each bank has 18 GPIO signals and both have separate power supply balls and can be driven individually. GPIO signals are grouped by pairs and each pair can get configured to be either two single ended or a differential pair signal.	36
	SerDes signal pads The SerDes interface consists of differential transmit and receive signals and a termination input pad. Furthermore, dedicated power supply exists to the SerDes interface and the SerDes PLL function.	5
	Power supply pads for FPGA core and GPIO GateMate™ FPGAcore gets supplied from a single power source. The GPIO banks have own power supply pads each. Additionally, some further power supply pads are available for SerDes and other functions.	78
	Ground pads GateMate™ FPGAhas a single ground level for all supply voltages. All ground pads must be connected.	74
	Any other pads Clock input (single ended or differential), Reset input and power-on reset adjustment input are available in this pad group.	5

Table 5.2: CCGM1A1 Pad list sorted by ball name

Ball	Signal name	Signal group	Description
A1	GND	Ground	
A2	VDD_WC	Power	3rd GPIO west bank power supply
A3	IO_NA_A0	GPIO	1st GPIO north bank signal A0
A4	IO_NA_A1	GPIO	1st GPIO north bank signal A1
A5	VDD_NA	Power	1st GPIO north bank power supply
A6	IO_NA_A4	GPIO	1st GPIO north bank signal A4
A7	GND	Ground	
A8	IO_NA_A7	GPIO	1st GPIO north bank signal A7
A9	IO_NB_B0	GPIO	2nd GPIO north bank signal B0
A10	GND	Ground	
A11	IO_NB_B2	GPIO	2nd GPIO north bank signal B2
A12	IO_NB_B4	GPIO	2nd GPIO north bank signal B4
A13	GND	Ground	
A14	IO_NB_B7	GPIO	2nd GPIO north bank signal B7
A15	IO_EB_B8	GPIO	2nd GPIO east bank signal B8
A16	VDD_EB	Power	2nd GPIO east bank power supply
A17	IO_EB_B5	GPIO	2nd GPIO east bank signal B5
A18	GND	Ground	
B1	IO_WC_A8	GPIO	3rd GPIO west bank signal A8
B2	IO_WC_B8	GPIO	3rd GPIO west bank signal B8
B3	IO_NA_B0	GPIO	1st GPIO north bank signal B0
B4	IO_NA_B1	GPIO	1st GPIO north bank signal B1
B5	IO_NA_A2	GPIO	1st GPIO north bank signal A2
B6	IO_NA_B4	GPIO	1st GPIO north bank signal B4
B7	VDD_NA	Power	1st GPIO north bank power supply
B8	IO_NA_B7	GPIO	1st GPIO north bank signal B7
B9	IO_NB_A0	GPIO	2nd GPIO north bank signal A0
B10	VDD_NB	Power	2nd GPIO north bank power supply
B11	IO_NB_A2	GPIO	2nd GPIO north bank signal A2
B12	IO_NB_A4	GPIO	2nd GPIO north bank signal A4
B13	VDD_NB	Power	2nd GPIO north bank power supply
B14	IO_NB_A7	GPIO	2nd GPIO north bank signal A7
B15	IO_EB_A8	GPIO	2nd GPIO east bank signal A8

Continued on next page

Table 5.2: CCGM1A1 Pad list sorted by ball name

Continued from previous page

Ball	Signal name	Signal group	Description
B16	GND	Ground	
B17	IO_EB_A5	GPIO	2nd GPIO east bank signal A5
B18	VDD_EB	Power	2nd GPIO east bank power supply
C1	GND	Ground	
C2	VDD_WC	Power	3rd GPIO west bank power supply
C3	IO_WC_A7	GPIO	3rd GPIO west bank signal A7
C4	IO_WC_B7	GPIO	3rd GPIO west bank signal B7
C5	IO_NA_B2	GPIO	1st GPIO north bank signal B2
C6	IO_NA_A3	GPIO	1st GPIO north bank signal A3
C7	IO_NA_A5	GPIO	1st GPIO north bank signal A5
C8	IO_NA_A6	GPIO	1st GPIO north bank signal A6
C9	IO_NA_A8	GPIO	1st GPIO north bank signal A8
C10	IO_NB_B1	GPIO	2nd GPIO north bank signal B1
C11	IO_NB_B3	GPIO	2nd GPIO north bank signal B3
C12	IO_NB_B5	GPIO	2nd GPIO north bank signal B5
C13	IO_NB_B6	GPIO	2nd GPIO north bank signal B6
C14	IO_NB_B8	GPIO	2nd GPIO north bank signal B8
C15	IO_EB_B7	GPIO	2nd GPIO east bank signal B7
C16	IO_EB_B6	GPIO	2nd GPIO east bank signal B6
C17	IO_EB_B4	GPIO	2nd GPIO east bank signal B4
C18	IO_EB_A4	GPIO	2nd GPIO east bank signal A4
D1	IO_WC_A5	GPIO	3rd GPIO west bank signal A5
D2	IO_WC_B5	GPIO	3rd GPIO west bank signal B5
D3	IO_WC_A6	GPIO	3rd GPIO west bank signal A6
D4	IO_WC_B6	GPIO	3rd GPIO west bank signal B6
D5	VDD_WC	Power	3rd GPIO west bank power supply
D6	IO_NA_B3	GPIO	1st GPIO north bank signal B3
D7	IO_NA_B5	GPIO	1st GPIO north bank signal B5
D8	IO_NA_B6	GPIO	1st GPIO north bank signal B6
D9	IO_NA_B8	GPIO	1st GPIO north bank signal B8
D10	IO_NB_A1	GPIO	2nd GPIO north bank signal A1
D11	IO_NB_A3	GPIO	2nd GPIO north bank signal A3

Continued on next page

Table 5.2: CCGM1A1 Pad list sorted by ball name

Continued from previous page

Ball	Signal name	Signal group	Description
D12	IO_NB_A5	GPIO	2nd GPIO north bank signal A5
D13	IO_NB_A6	GPIO	2nd GPIO north bank signal A6
D14	IO_NB_A8	GPIO	2nd GPIO north bank signal A8
D15	IO_EB_A7	GPIO	2nd GPIO east bank signal A7
D16	IO_EB_A6	GPIO	2nd GPIO east bank signal A6
D17	IO_EB_B2	GPIO	2nd GPIO east bank signal B2
D18	IO_EB_A2	GPIO	2nd GPIO east bank signal A2
E1	IO_WC_A3	GPIO	3rd GPIO west bank signal A3
E2	IO_WC_B3	GPIO	3rd GPIO west bank signal B3
E3	IO_WC_A4	GPIO	3rd GPIO west bank signal A4
E4	IO_WC_B4	GPIO	3rd GPIO west bank signal B4
E5	GND	Ground	
E6	VDD_NA	Power	1st GPIO north bank power supply
E7	GND	Ground	
E8	VDD_NA	Power	1st GPIO north bank power supply
E9	GND	Ground	
E10	VDD_NB	Power	2nd GPIO north bank power supply
E11	GND	Ground	
E12	VDD_NB	Power	2nd GPIO north bank power supply
E13	GND	Ground	
E14	VDD_EB	Power	2nd GPIO east bank power supply
E15	IO_EB_B3	GPIO	2nd GPIO east bank signal B3
E16	IO_EB_A3	GPIO	2nd GPIO east bank signal A3
E17	VDD_EB	Power	2nd GPIO east bank power supply
E18	GND	Ground	
F1	GND	Ground	
F2	VDD_WC	Power	3rd GPIO west bank power supply
F3	IO_WC_A2	GPIO	3rd GPIO west bank signal A2
F4	IO_WC_B2	GPIO	3rd GPIO west bank signal B2
F5	VDD_WC	Power	3rd GPIO west bank power supply
F6	GND	Ground	
F7	VDD_NA	Power	1st GPIO north bank power supply

Continued on next page

Table 5.2: CCGM1A1 Pad list sorted by ball name

Continued from previous page

Ball	Signal name	Signal group	Description
F8	GND	Ground	
F9	VDD	Power	Core power supply
F10	GND	Ground	
F11	VDD_NB	Power	2nd GPIO north bank power supply
F12	GND	Ground	
F13	VDD_EB	Power	2nd GPIO east bank power supply
F14	GND	Ground	
F15	IO_EB_B1	GPIO	2nd GPIO east bank signal B1
F16	IO_EB_A1	GPIO	2nd GPIO east bank signal A1
F17	IO_EB_B0	GPIO	2nd GPIO east bank signal B0
F18	IO_EB_A0	GPIO	2nd GPIO east bank signal A0
G1	IO_WC_A0	GPIO	3rd GPIO west bank signal A0
G2	IO_WC_B0	GPIO	3rd GPIO west bank signal B0
G3	IO_WC_A1	GPIO	3rd GPIO west bank signal A1
G4	IO_WC_B1	GPIO	3rd GPIO west bank signal B1
G5	GND	Ground	
G6	VDD	Power	Core power supply
G7	GND	Ground	
G8	VDD	Power	Core power supply
G9	GND	Ground	
G10	VDD	Power	Core power supply
G11	GND	Ground	
G12	VDD	Power	Core power supply
G13	GND	Ground	
G14	VDD_EA	Power	1st GPIO east bank power supply
G15	IO_EA_B8	GPIO	1st GPIO east bank signal B8
G16	IO_EA_A8	GPIO	1st GPIO east bank signal A8
G17	IO_EA_B7	GPIO	1st GPIO east bank signal B7
G18	IO_EA_A7	GPIO	1st GPIO east bank signal A7
H1	IO_WB_A7	GPIO	2nd GPIO west bank signal A7
H2	IO_WB_B7	GPIO	2nd GPIO west bank signal B7
H3	IO_WB_A8	GPIO	2nd GPIO west bank signal A8

Continued on next page

Table 5.2: CCGM1A1 Pad list sorted by ball name

Continued from previous page

Ball	Signal name	Signal group	Description
H4	IO_WB_B8	GPIO	2nd GPIO west bank signal B8
H5	VDD_WB	Power	2nd GPIO west bank power supply
H6	GND	Ground	
H7	VDD	Power	Core power supply
H8	GND	Ground	
H9	VDD	Power	Core power supply
H10	GND	Ground	
H11	VDD	Power	Core power supply
H12	GND	Ground	
H13	VDD	Power	Core power supply
H14	GND	Ground	
H15	IO_EA_B6	GPIO	1st GPIO east bank signal B6
H16	IO_EA_A6	GPIO	1st GPIO east bank signal A6
H17	VDD_EA	Power	1st GPIO east bank power supply
H18	GND	Ground	
J1	GND	Ground	
J2	VDD_WB	Power	2nd GPIO west bank power supply
J3	IO_WB_A6	GPIO	2nd GPIO west bank signal A6
J4	IO_WB_B6	GPIO	2nd GPIO west bank signal B6
J5	GND	Ground	
J6	VDD	Power	Core power supply
J7	GND	Ground	
J8	VDD	Power	Core power supply
J9	GND	Ground	
J10	VDD	Power	Core power supply
J11	GND	Ground	
J12	VDD	Power	Core power supply
J13	GND	Ground	
J14	VDD_EA	Power	1st GPIO east bank power supply
J15	IO_EA_B5	GPIO	1st GPIO east bank signal B5
J16	IO_EA_A5	GPIO	1st GPIO east bank signal A5
J17	IO_EA_B4	GPIO	1st GPIO east bank signal B4

Continued on next page

Table 5.2: CCGM1A1 Pad list sorted by ball name

Continued from previous page

Ball	Signal name	Signal group	Description
J18	IO_EA_A4	GPIO	1st GPIO east bank signal A4
K1	IO_WB_A5	GPIO	2nd GPIO west bank signal A5
K2	IO_WB_B5	GPIO	2nd GPIO west bank signal B5
K3	IO_WB_A4	GPIO	2nd GPIO west bank signal A4
K4	IO_WB_B4	GPIO	2nd GPIO west bank signal B4
K5	VDD_WB	Power	2nd GPIO west bank power supply
K6	GND	Ground	
K7	VDD	Power	Core power supply
K8	GND	Ground	
K9	VDD	Power	Core power supply
K10	GND	Ground	
K11	VDD	Power	Core power supply
K12	GND	Ground	
K13	VDD	Power	Core power supply
K14	GND	Ground	
K15	IO_EA_B3	GPIO	1st GPIO east bank signal B3
K16	IO_EA_A3	GPIO	1st GPIO east bank signal A3
K17	IO_EA_B2	GPIO	1st GPIO east bank signal B2
K18	IO_EA_A2	GPIO	1st GPIO east bank signal A2
L1	IO_WB_A3	GPIO	2nd GPIO west bank signal A3
L2	IO_WB_B3	GPIO	2nd GPIO west bank signal B3
L3	IO_WB_A2	GPIO	2nd GPIO west bank signal A2
L4	IO_WB_B2	GPIO	2nd GPIO west bank signal B2
L5	GND	Ground	
L6	VDD	Power	Core power supply
L7	GND	Ground	
L8	VDD	Power	Core power supply
L9	GND	Ground	
L10	VDD	Power	Core power supply
L11	GND	Ground	
L12	VDD	Power	Core power supply
L13	GND	Ground	

Continued on next page

Table 5.2: CCGM1A1 Pad list sorted by ball name

Continued from previous page

Ball	Signal name	Signal group	Description
L14	VDD_EA	Power	1st GPIO east bank power supply
L15	IO_EA_B1	GPIO	1st GPIO east bank signal B1
L16	IO_EA_A1	GPIO	1st GPIO east bank signal A1
L17	VDD_EA	Power	1st GPIO east bank power supply
L18	GND	Ground	
M1	GND	Ground	
M2	VDD_WB	Power	2nd GPIO west bank power supply
M3	IO_WB_A1	GPIO	2nd GPIO west bank signal A1
M4	IO_WB_B1	GPIO	2nd GPIO west bank signal B1
M5	VDD_WB	Power	2nd GPIO west bank power supply
M6	GND	Ground	
M7	VDD	Power	Core power supply
M8	GND	Ground	
M9	VDD	Power	Core power supply
M10	GND	Ground	
M11	VDD	Power	Core power supply
M12	GND	Ground	
M13	VDD	Power	Core power supply
M14	IO_EA_B0	GPIO	1st GPIO east bank signal B0
M15	IO_EA_A0	GPIO	1st GPIO east bank signal A0
M16	GND	Ground	
M17	IO_SB_A3	GPIO	2nd GPIO south bank signal A3
M18	IO_SB_B3	GPIO	2nd GPIO south bank signal B3
N1	IO_WB_A0	GPIO	2nd GPIO west bank signal A0
N2	IO_WB_B0	GPIO	2nd GPIO west bank signal B0
N3	SPI_CS_N	GPIO	1st function: Configuration SPI chip select
	IO_WA_A8	GPIO	2nd function: 1st GPIO west bank signal A8
N4	SPI_CLK	GPIO	1st function: Configuration SPI clock
	IO_WA_B8	GPIO	2nd function: 1st GPIO west bank signal B8
N5	VDD_WA	Power	1st GPIO west bank power supply
N6	VDD	Power	Core power supply
N7	GND	Ground	
N8	VDD	Power	Core power supply

Continued on next page

Table 5.2: CCGM1A1 Pad list sorted by ball name

Continued from previous page

Ball	Signal name	Signal group	Description
N9	GND	Ground	
N10	VDD	Power	Core power supply
N11	VDD_SB	Power	2nd GPIO south bank power supply
N12	GND	Ground	
N13	VDD_SB	Power	2nd GPIO south bank power supply
N14	IO_SB_A8	GPIO	1st function: 2nd GPIO south bank signal A8
	CLK1	GPIO	2nd function: 1st clock input
N15	IO_SB_B8	GPIO	2nd GPIO south bank signal B8
N16	N.C.	Other	Not connected and must be left open
N17	GND	Ground	
N18	VDD_SB	Power	2nd GPIO south bank power supply
P1	SPI_D1	GPIO	1st function: Configuration SPI data bit 1
	IO_WA_A7	GPIO	2nd function: 1st GPIO west bank signal A7
P2	SPI_D0	GPIO	1st function: Configuration SPI data bit 0
	IO_WA_B7	GPIO	2nd function: 1st GPIO west bank signal B7
P3	VDD_WA	Power	1st GPIO west bank power supply
P4	GND	Ground	
P5	VDD_WA	Power	1st GPIO west bank power supply
P6	VDD_SA	Power	1st GPIO south bank power supply
P7	GND	Ground	
P8	VDD_SA	Power	1st GPIO south bank power supply
P9	GND	Ground	
P10	VDD_SA	Power	1st GPIO south bank power supply
P11	IO_SB_A4	GPIO	2nd GPIO south bank signal A4
P12	IO_SB_A7	GPIO	1st function: 2nd GPIO south bank signal A7
	CLK2	GPIO	2nd function: 2nd clock input
P13	IO_SB_B7	GPIO	2nd GPIO south bank signal B7
P14	IO_SB_A6	GPIO	1st function: 2nd GPIO south bank signal A6
	CLK3	GPIO	2nd function: 3rd clock input
P15	IO_SB_B6	GPIO	2nd GPIO south bank signal B6
P16	VDD_PLL	Power	PLL power supply
P17	IO_SB_A2	GPIO	2nd GPIO south bank signal A2
P18	IO_SB_B2	GPIO	2nd GPIO south bank signal B2

Continued on next page

Table 5.2: CCGM1A1 Pad list sorted by ball name

Continued from previous page

Ball	Signal name	Signal group	Description
R1	SPI_D3	GPIO	1st function: Configuration SPI data bit 3
	IO_WA_A6	GPIO	2nd function: 1st GPIO west bank signal A6
R2	SPI_D2	GPIO	1st function: Configuration SPI data bit 2
	IO_WA_B6	GPIO	2nd function: 1st GPIO west bank signal B6
R3	JTAG_TCK	GPIO	1st function: Configuration JTAG clock
	IO_WA_A5	GPIO	2nd function: 1st GPIO west bank signal A5
R4	SPI_FWD	GPIO	1st function: Configuration SPI data forward
	IO_WA_B5	GPIO	2nd function: 1st GPIO west bank signal B5
R5	CFG_MD0	GPIO	1st function: Configuration mode bit 0
	IO_WA_A0	GPIO	2nd function: 1st GPIO west bank signal A0
R6	IO_SA_A1	GPIO	1st GPIO south bank signal A1
R7	IO_SA_A2	GPIO	1st GPIO south bank signal A2
R8	IO_SA_A4	GPIO	1st GPIO south bank signal A4
R9	IO_SA_A6	GPIO	1st GPIO south bank signal A6
R10	IO_SA_A7	GPIO	1st GPIO south bank signal A7
R11	IO_SB_B4	GPIO	2nd GPIO south bank signal B4
R12	GND	Ground	
R13	IO_SB_A5	GPIO	1st function: 2nd GPIO south bank signal A5
	CLK4	GPIO	2nd function: 4th clock input
R14	IO_SB_B5	GPIO	2nd GPIO south bank signal B5
R15	VDD_SB	Power	2nd GPIO south bank power supply
R16	GND	Ground	
R17	IO_SB_A1	GPIO	2nd GPIO south bank signal A1
R18	IO_SB_B1	GPIO	2nd GPIO south bank signal B1
T1	VDD_WA	Power	1st GPIO west bank power supply
T2	JTAG_TDI	GPIO	1st function: JTAG data input
	IO_WA_A4	GPIO	2nd function: 1st GPIO west bank signal A4
T3	JTAG_TMS	GPIO	1st function: JTAG test mode select
	IO_WA_B4	GPIO	2nd function: 1st GPIO west bank signal B4
T4	GND	Ground	
T5	CFG_MD1	GPIO	1st function: Configuration mode bit 1
	IO_WA_B0	GPIO	2nd function: 1st GPIO west bank signal B0
T6	IO_SA_B1	GPIO	1st GPIO south bank signal B1
T7	IO_SA_B2	GPIO	1st GPIO south bank signal B2

Continued on next page

Table 5.2: CCGM1A1 Pad list sorted by ball name

Continued from previous page

Ball	Signal name	Signal group	Description
T8	IO_SA_B4	GPIO	1st GPIO south bank signal B4
T9	IO_SA_B6	GPIO	1st GPIO south bank signal B6
T10	IO_SA_B7	GPIO	1st GPIO south bank signal B7
T11	GND	Ground	
T12	SER_CLK	Other	SerDes clock: positive LVDS signal (or common use clock single ended)
T13	SER_CLK_N	Other	SerDes clock: negative LVDS signal
T14	VDD_CLK	Power	Clock signal power supply
T15	RST_N	Other	Reset
T16	VDD_SER_PLL	Power	SerDes PLL power supply (1.0 V to 1.1 V ± 50 mV)
T17	GND	Ground	
T18	VDD_SB	Power	2nd GPIO south bank power supply
U1	POR_EN	GPIO	1st function: Enable power-on reset
	IO_WA_A3	GPIO	2nd function: 1st GPIO west bank signal A3
U2	JTAG_TDO	GPIO	1st function: JTAG data output
	IO_WA_B3	GPIO	2nd function: 1st GPIO west bank signal B3
U3	VDD_WA	Power	1st GPIO west bank power supply
U4	CFG_MD2	GPIO	1st function: Configuration mode bit 2
	IO_WA_A1	GPIO	2nd function: 1st GPIO west bank signal A1
U5	IO_SA_A0	GPIO	1st GPIO south bank signal A0
U6	VDD_SA	Power	1st GPIO south bank power supply
U7	IO_SA_A3	GPIO	1st GPIO south bank signal A3
U8	IO_SA_A5	GPIO	1st GPIO south bank signal A5
U9	VDD_SA	Power	1st GPIO south bank power supply
U10	IO_SA_A8	GPIO	1st GPIO south bank signal A8
U11	SER_RX_P	SerDes	Receive data line: positive LVDS signal
U12	VDD_SER	Power	SerDes core power supply (1.0 V to 1.1 V ± 50 mV)
U13	SER_TX_P	SerDes	Transmit data line: positive LVDS signal
U14	GND	Ground	
U15	GND	Ground	
U16	GND	Ground	
U17	IO_SB_A0	GPIO	2nd GPIO south bank signal A0
U18	IO_SB_B0	GPIO	2nd GPIO south bank signal B0

Continued on next page

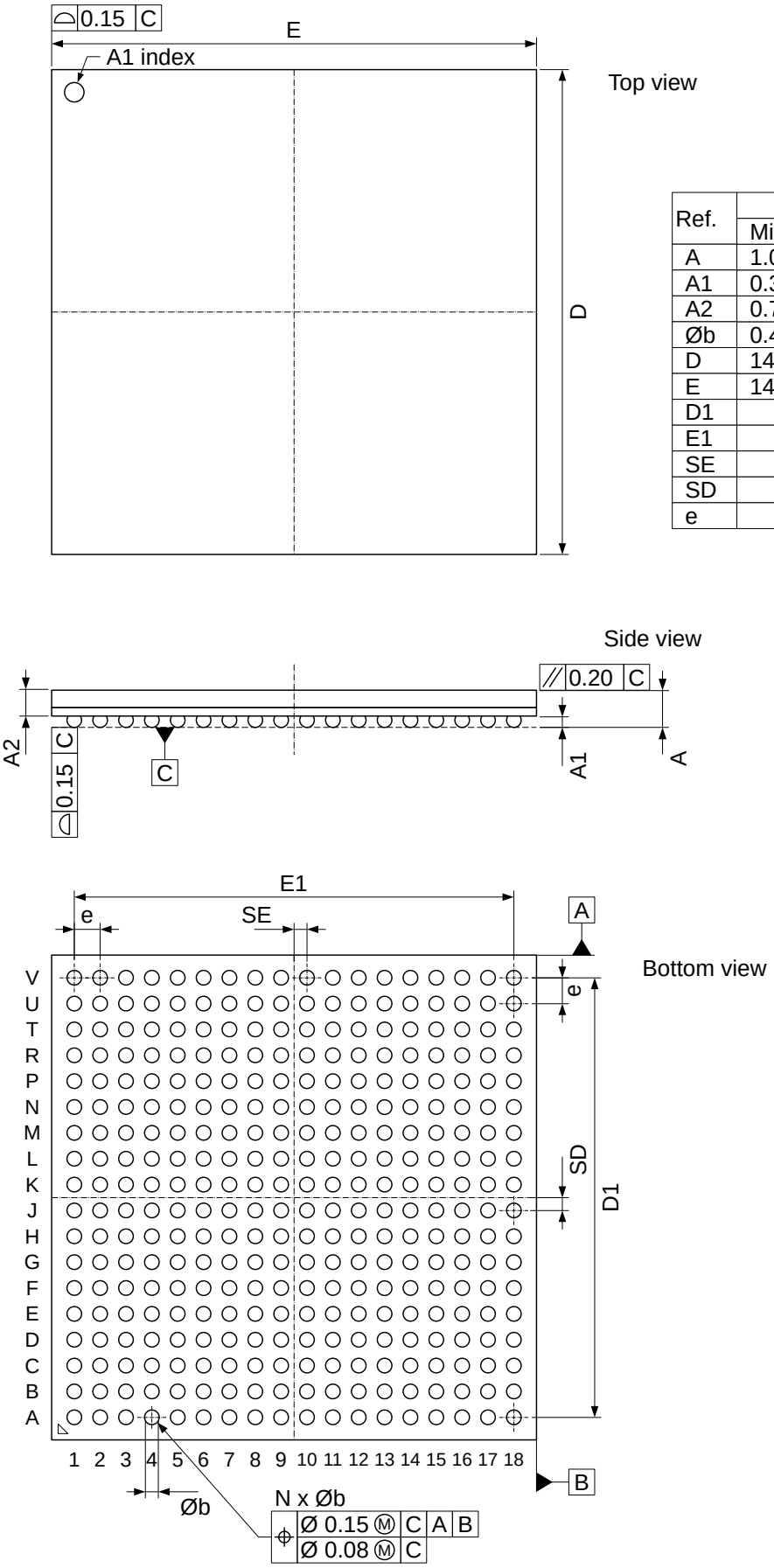
Table 5.2: CCGM1A1 Pad list sorted by ball name

Continued from previous page

Ball	Signal name	Signal group	Description
V1	GND	Ground	
V2	CFG_FAILED	GPIO	1st function: Configuration failed signal
	IO_WA_A2	GPIO	2nd function: 1st GPIO west bank signal A2
V3	CFG_DONE	GPIO	1st function: Configuration done signal
	IO_WA_B2	GPIO	2nd function: 1st GPIO west bank signal B2
V4	CFG_MD3	GPIO	1st function: Configuration mode bit 3
	IO_WA_B1	GPIO	2nd function: 1st GPIO west bank signal B1
V5	IO_SA_B0	GPIO	1st GPIO south bank signal B0
V6	GND	Ground	
V7	IO_SA_B3	GPIO	1st GPIO south bank signal B3
V8	IO_SA_B5	GPIO	1st GPIO south bank signal B5
V9	GND	Ground	
V10	IO_SA_B8	GPIO	1st GPIO south bank signal B8
V11	SER_RX_N	SerDes	Receive data line: negative LVDS signal
V12	SER_RTERM	SerDes	Line termination
V13	SER_TX_N	SerDes	Transmit data line: negative LVDS signal
V14	GND	Ground	
V15	POR_ADJ	Other	Power-on-Reset level adjustment
V16	GND	Ground	
V17	VDD_SER	Power	SerDes core power supply (1.0 V to 1.1 V $\pm$ 50 mV)
V18	GND	Ground	

Chapter 6

Mechanical Dimensions



Ref.	Dimensions in mm		
	Min	Nom	Max
A	1.080	1.190	1.300
A1	0.300	0.350	0.400
A2	0.780	0.840	0.900
Øb	0.400	0.450	0.500
D	14.900	15.000	15.100
E	14.900	15.000	15.100
D1	13.600 BSC		
E1	13.600 BSC		
SE	0.400 BSC		
SD	0.400 BSC		
e	0.800 BSC		

Figure 6.1: FBGA 324 package dimensions

Acronyms

BSDL Boundary Scan Description Language.

CPE Central Programming Element.

DDR Double Data Rate.

DPSRAM Dual Port SRAM.

ECC Error Checking and Correction.

FBGA Fine-pitch Ball Grid Array.

FSM Finite State Machine.

GPIO General Purpose Input/Output.

IPCEI Important Project of Common European Interest.

LSB Least Significant Bit.

LVDS Low-Voltage Differential Signaling.

PLL Phase Locked Loop.

SB Switch Box.

SerDes Serializer/Deserializer.

TAP Test Access Point.

TCK Test Clock.

TDI Test Data Input.

TDO Test Data Output.

TMS Test Mode Select.

GateMate™ FPGA Datasheet
DS1001
December 2020

Cologne Chip