CM3 FPGA Register System IP

Wishbone-based multi-controller, multi-target, configuration-driven register fabric for FPGA designs

Parameter	Description
Category	FPGA IP - register fabric / control plane
Bus	Wishbone (B4-compatible command/response model)
Language	VHDL
Scope	Registers, peripheral targets, multi-controller arbitration, memory maps
Typical use	GenICam/feature maps, debug access, on-board managers, PHYs, sensors
Used by Tecphos IP	CoaXPress, GigEVision, EEVideo GenICam / feature registers
License	MIT (delivered to Tecphos IP customers)
Document	Datasheet DS-CM3-001 | 2026

1. Overview

CM3 is a controller-based framework for managing multiple capability registers, multiple peripheral targets, and multiple peripheral controllers inside FPGA designs. It organizes status, configuration, memory windows, and peripheral engines into a single address space so every function is reachable by the same access process.

All Tecphos FPGA designs use CM3 in project-specific configurations to manage status and configuration registers and to interface with peripherals such as I2C and SPI. Tecphos CoaXPress, GigEVision, and EEVideo IP cores use CM3 for defined GenICam and feature registers. Customers who license those cores work in the same register model Tecphos uses internally.

2. Why CM3

Most FPGA systems need a CSR map, peripheral bridges, and more than one control path (protocol stack, debug port, on-board configuration manager). Building that fabric ad hoc burns schedule and creates inconsistent access models. CM3 provides reusable, configuration-driven modules so teams launch or expand designs without re-implementing arbitration, decoding, and register plumbing on every project.

- Single map for registers, targets, and controllers
- Add registers and targets from declaration tables - core logic stays frozen
- Multiple masters share one feature set with round-robin arbitration
- Structure-based register I/O avoids port explosion through the hierarchy
- Library of controllers and targets for imaging and board bring-up

3. Architecture

3.1 Controllers (masters)

One or more Wishbone controllers issue cycles into CM3. Controllers are declared in a project package enumeration and connected at the top level. CM3 arbitrates concurrent access with a round-robin grant mechanism so masters share the same map safely. Typical roles: GenICam/protocol soft logic, SPI slave bridge, soft CPU (e.g. RISC-V) via async Wishbone CDC, UART debug, and on-board configuration managers.

3.2 Targets (slaves)

Targets are memory-mapped windows or peripheral engines. Each target is declared with a base address, number of address bits to decode, and data width, then wired at the top level. Registers and memories share the same decode path.

3.3 Local register fabric

Local registers are generated from a configuration list using looping VHDL. Each field specifies type, bit width, MSB location (for packing multiple fields in one word), write-enable class, default value, and address. Adding a register is typically one configuration line - no new hand-written register process. Field inputs/outputs are collected in array structures for clean assignment throughout the design.

4. Register model

CM3 supports a richer set of register behaviors than simple R/W CSRs:

Type	Behavior
r	Read-only static (default / constant)
ra	Read-only; updated from local logic asynchronously
ru	Read-only; updated from local logic with synchronous enable
rw	Read/write from the bus
rwu	Bus R/W plus local synchronous update
rwt	Bus R/W plus local reset
rsc	Read plus bus set/clear
rscu	Set/clear plus local update
rsct	Set/clear plus local reset

Additional controls include optional write-enable gating per field and bit-field packing via BitWidth and MSBLoc. Remote access indicators (remote read/write strobes) support side-effect-aware logic.

5. Library modules

Module class	Examples / role
Core	CM3 fabric: decode, arbiter, register bank, debug taps, optional timeout
SPI controller	SPI pin interface into the Wishbone map (address/command/data; multi-word sequences)
I2C target	I2C engine with FIFOs; SCL rate derived from Wishbone clock configuration
UART target	FIFO-backed serial path for debug and command
SMI / MDIO target	Ethernet PHY management bus access from the common map
Async Wishbone	Clock-domain crossing bridge for masters on a different clock (e.g. soft CPU)
Dual-port memory	Init-capable DP memory windows; optional write-protect on MSBs
Logic analyzer	On-chip capture macro for bring-up
CM3 monitor	Bus transaction capture into readable debug memory
Support	Sync FIFOs, Ethernet CRC-16 helpers

6. Project workflow

CM3 is designed so project differentiation lives in a package file, not in forked bus logic:

- Copy a CM3 package template/example into the project (controllers, targets, register list).
- Edit only the marked customization sections (lists and parameter tables).
- Keep CM3.vhd (core) unmodified across products.
- Wire controllers and targets at the top level; map I2C/SPI/UART/MDIO/memory as needed.
- Simulate with provided testbench patterns (e.g. GHDL scripts in the tree).

Wishbone parameters such as address width, data width, and optional decode pipelining (DecodeRegister) are project constants. Example imaging packages use 12-bit address / 32-bit data; the template also supports narrower buses.

7. Core options

Option	Purpose
TimeOutEn / TimeOutWait	Optional transaction watchdog; recovers hung cycles (with WbClkFreq)
DecodeRegister	Optional extra decode stage for routing/timing closure

WbAddrBits / WbDataBits	Configurable map width (project package)
DebugCmd / DebugRsp / IDs	Observability of active command, response, granted controller, target

8. Integration sketch

- Masters (controllers) -> arbitration -> internal Wishbone command
- Decode -> local register bank and/or target window enable
- Targets return Ack/Data; responses muxed back to the granted controller
- Local FPGA logic reads/writes register structures without per-field top-level ports

Because GenICam, SPI debug, and on-board managers share one map, feature addresses stay consistent whether accessed over the video protocol stack or a board test path.

9. Deliverables

- VHDL sources: core, example/template packages, controllers, targets, reg macros, memory, support
- Simulation aids (testbench / GHDL scripts where provided)
- Wishbone B4 specification PDF (reference)
- This datasheet

10. License

CM3 is provided under the MIT License (Copyright Tecphos Inc.). Tecphos delivers CM3 to IP customers so they can extend product designs on the same foundation used by Tecphos CoaXPress, GigEVision, and EEVideo cores. Include the copyright and permission notice in all copies or substantial portions of the Software. The software is provided "as is," without warranty.

Commercial Tecphos video IP (CoaXPress, GigEVision, EEVideo, etc.) remains under separate project license terms. CM3 does not by itself convey a license to those protocol cores.

11. Contact

CM3 is included with applicable Tecphos IP engagements. For evaluation, customization support, or questions about integrating CM3 with GenICam feature maps and board peripherals:

Field	Value
Web	https://www.tecphos.com/products/cm3-fpga-register-system-ip/
Email	info@tecphos.com
Phone	+1 719 426 2080
Address	4040 E Bijou St., Colorado Springs, Colorado, USA