Multicore Debugging for GTM Virtual Prototype

Matthias Noack, Jens Braunes
PLS Programmierbare Logik & Systeme GmbH

PLS Development Tools

- Debug, test and trace tools for microcontrollers and embedded processors
- Made in Germany
- Easy-to-use, efficient and reliable development tools since 1990

Multicore Debugging for GTM Virtual Prototype

pls Development Tools

UDE® Universal Debug Engine

Complete tool set for debugging and trace

- On real target hardware
  - Microcontrollers, embedded systems, multicore SoCs
- On **Virtual Prototypes** / Simulators
- Real multicore debugging
  - System centric debugger environment, not core centric
  - One common user interface
  - Integrated Core Debuggers instead of separate debugger UI instances for cores

Architecture and Controller Support

Infineon	STMicroelectronics	NXP	arm	Renesas	Texas Instruments	RISC-V	Synopsys
AURIX TC4x, TC3xx, TC2xx	Stellar automotive microcontroller family	S32N55 Vehicle Super-Integration Processor	Cortex-M (M7, M4, M4F, M3, M1, M0)	RH850/U, RH850/E, RH850/C	Jacinto 7 TDA4, DRA8	SiFive	ARC EM
Automotive RISC-V Prototype	SPC58xx, SPC57xx, SPC56xx	S32G, S32G3 Vehicle Network Processors	Cortex-M23, Cortex-M33	RH850/F1, RH850/P1, RH850/D1	Sitara AM64x	GreenWaves	ARC EV
Traveo T2G	STM32 series	S32Z, S32E Real-Time Processors	Cortex-R52, Cortex-R4, Cortex-R5	R-Car	Sitara AM243x	GigaDevice	ARC HS
XMC7000		S32V S32V2 Processors for Vision	Cortex-A8, Cortex-A9 Cortex-A53	RZ/T, RZ/N			ARC-V
XMC4xxx, XMC1xxx		S32K1, S32K3, S32M	ARM7 / ARM9 / ARM11	RA4, RA6			Various Virtualizer Development Kits
TriCore TC17xx, TC11xx		S32R Radar Processing		SH-2A			
XC2000 / XE166		i.MX-RT					
		MPC57xx, MPC56xx					
		Kinetis, LPC					
		KW45 Wireless MCU					

GTM IP implemented

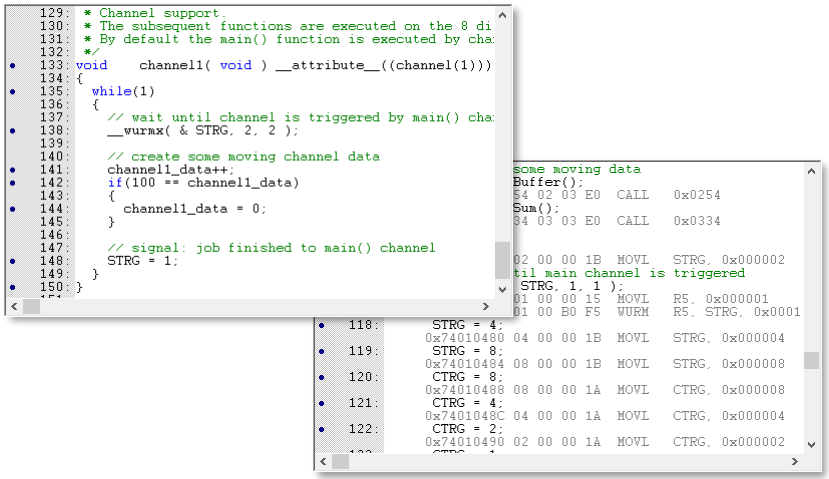
UDE GTM Support
- Basic debug support for GTM version < 4.1
- MCS run-control for GTM 4.1
- Trace support

UDE Support for Virtual Prototypes / Virtual Targets
- Synopsys VDK incl. GTM
- **COSEDA COSIDE®**

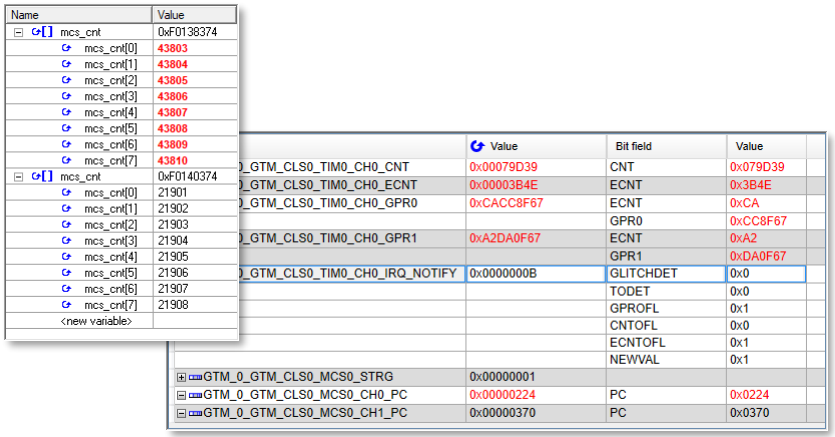
pls Development Tools

Basic GTM MCS Debugging

- Display of assembler code executed by MCS channels
- Display of C sources if MCS code is compiled by C-compiler



- Display of channel registers and module registers (TIM, TOM, SPE, etc.)
- Watch variables
- Real-time watch of variables, registers, memory
- Modifiable variables, registers, memory

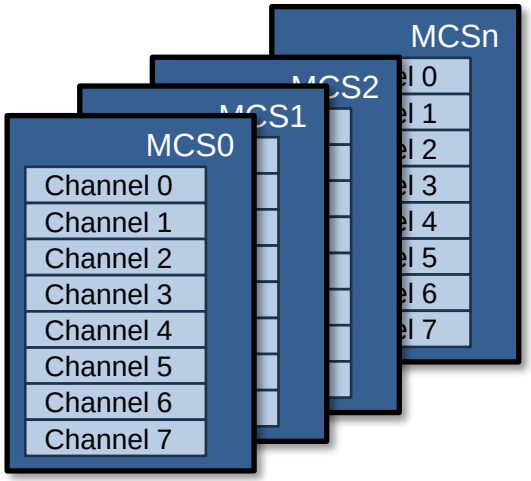


pls Development Tools

MCS Instances and Channels

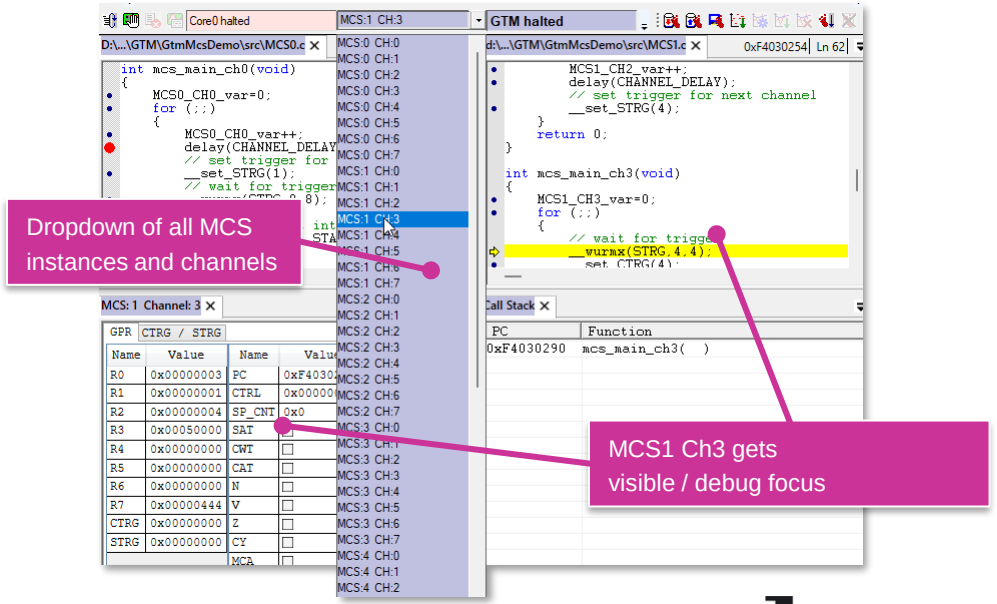
GTM Architecture

- *n* MCS instances
- *8* programmable channels per MCS instance



Debugger View

- Focus on single channel of MCS*n*
- Switching between channels:

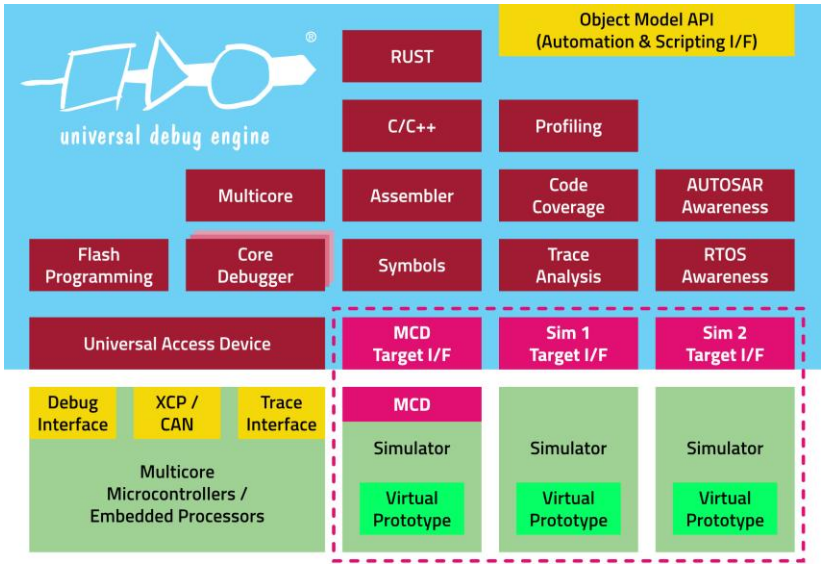


Debugging on Virtual Prototypes with UDE®

- Unified debugging environment for both virtual prototypes and real hardware
- Start software development early (pre-silicon)
  - E.g. for GTM5
  - Smooth transition to real hardware — same UDE interface, same workflows
  - Shorter time-to-market
- HW-less automated regression testing
  - Repeatable runs, and reproducible scenarios
  - Cost reduction
  - Higher software quality

Simulator integration

- MCD – Multi-Core Debug API (e.g. COSIDE)
- Integrated instruction set simulators
- Proprietary simulator I/F



Debugging GTM Virtual Prototype vs. Real Silicon

GTM in Real Silicon

- GTM needs to be enabled and initialized by host application

GTM prior version 4.1

- No HW debug support for breakpoints
- No breakpoints, no single stepping

- Suspend / release synchronized with run-control of host cores
  - GTM can be suspended when host core(s) hits breakpoint

GTM version 4.1 (and higher)

- Two hardware breakpoints introduced with GTM v4.1
- “Normal” debugging of MCS code now possible
- Breakpoints + Single stepping
- Break options for breakpoint or manual break
- Only MCS channel in focus
  - Entire GTM

Virtual GTM (COSIDE®)

- Independent from host MCU
- Host application by COSIDE®

- No enabling and initializing needed
- GTM is ready right from the beginning

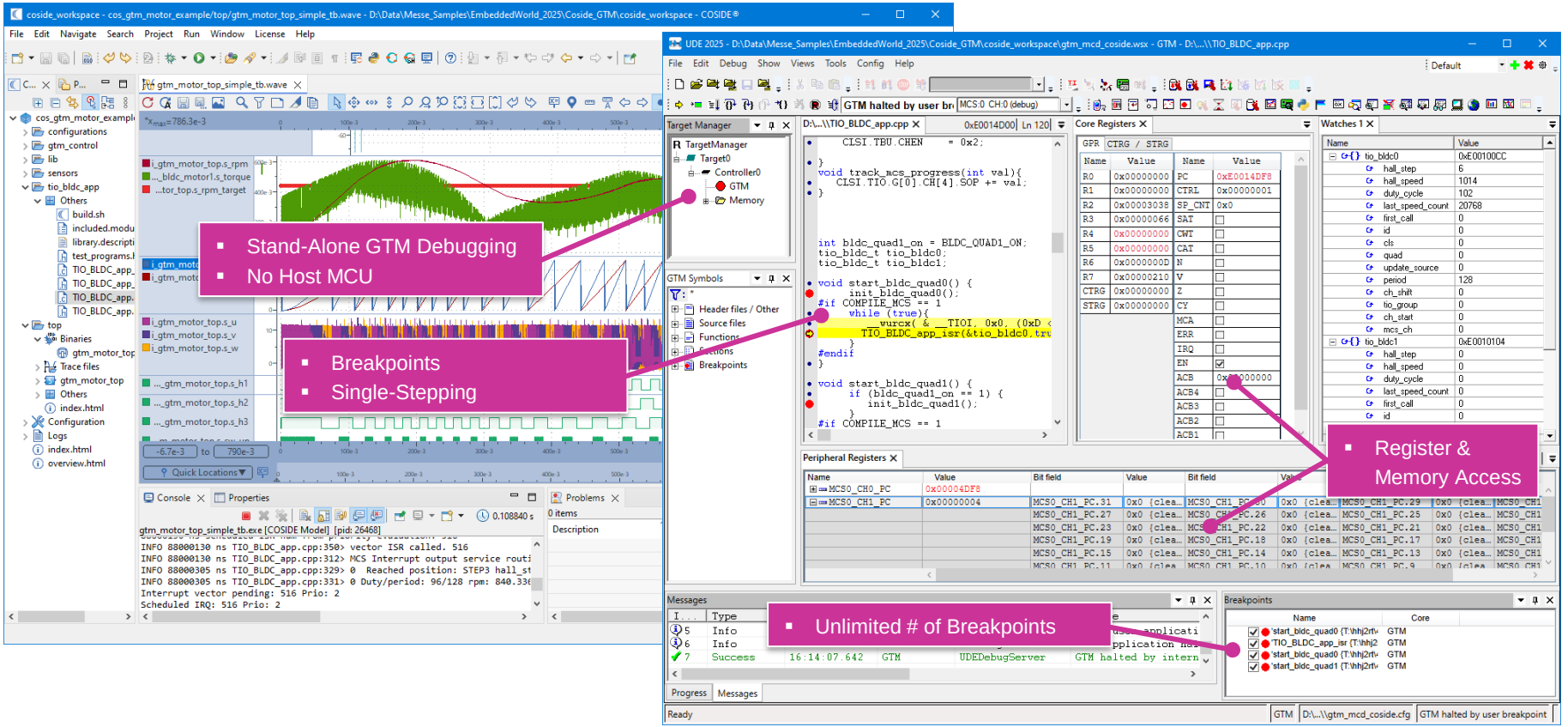
- Unlimited number of breakpoints

- Entire simulation model is halted on break, “Time is frozen”
- Breaking has no influence on system behavior
- 100% reproducible

UDE as frontend for simulator

- Virtual target in UDE, no debug probe
- Same look and feel and comparable functionality
- Seamless switch from pre-silicon development to real HW

Debugging GTM Virtual Prototype



UDE® as Frontend for COSIDE®

- ☑ Start development in pre-silicon phase (e.g. for GTM5)
- ☑ Better project partitioning → development of pure GTM functionality independent from host SoC
- ☑ Continued use of the same tooling when switching to real hardware

Debug ▪ Trace ▪ Test

pls
Development Tools

info@pls-mc.com
www.pls-mc.com