

Using virtualGTM with Coside –

advantages, opportunities

AE/PAI-IP 06.12.2023 Jürgen Hanisch



Bosch Semiconductors.

We make the world safe, smart and accessible.





Using virtualGTM with Coside Agenda



- What is the GTM-IP and where is it used
- What is the virtualGTM
- Who should know/ use the virtualGTM ?
- Opportunities of/ with virtualGTM

Introduction to use cases and using virtualGTM in Coside



Using virtualGTM with Coside What is the GTM-IP and where is it used







GTM-IP – Legal Framework From requirements to solutions



5 Internal | AE/PAI-IP | 2023-12-06

© Robert Bosch GmbH 2023. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

BOSCH

GTM-IP – Overview GTM Gen 1 – Gen 4 Delivered device configuration 2010 - 2022



Delivered GTM configurations **GTM IP Deliveries** (by GTM IP Generation) First GTM IP (Gen 1) delivered in 2010 Meanwhile more than 40 different configurations delivered to multiple semiconductor vendors Delivery of GTM IP Gen 4 starting in 2021 Gen 1 Gen 2 Gen 3 Gen 3.5 Gen 4.1 GTM Gen 4.1 devices dominating the list with more then Gen 3/3.5 in less than 2 years



GTM-IP – Status and Roadmap New markets – application driven growth





GTM-IP GTM IP 4.1 microcontrollers

Microcontrollers from semiconductor companies

Chipower THA6 (GTM 3)

Infineon Aurix TC4x family

NXP <u>S32Z2/S32E2</u>

Renesas <u>RH850/U2B</u>

8

ST Microelectronics Stellar G/P family





GTM-IP – Architecture Key features and structures

Real-time oriented I/O co-processor



Modular design approach – tailorable to the target application



Rich set of multi-purpose and specialized modules



RISC-based internal cores with 8x multi-threading



2 levels of GTM internal bus systems to connect sub-modules (intra- and inter-cluster communication)



06

9

 μ C bus master interface enables offloading of workloads from μ C cores to GTM



Programmability by external CPUs



GTM-IP – Architecture Scalability and massive parallel processing

Highly scalable and modular approach, supporting a wide range of μ C, system and application requirements

- 01 Low-End support with single instance of basic I/O function modules
- 02 High-End support with multiple instances of complex I/O function modules, various special function modules as well as RISC cores
- 03 Autonomous and massive parallel real-time task processing support
- 04 Typical clock frequency up to 200 MHz



10 Internal | AE/PAI-IP | 2023-12-06

© Robert Bosch GmbH 2023. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



GTM-IP – Architecture virtualGTM – actual content

VirtualGTM available for download:Uses fully equipped 1 cluster GTM

01 Every GTM function available

02 Ideal suited for education/ training example

03 Complex applications using multiple GTM resources incl. MCS and ARU can be put in place

04 VirtualGTM variants (more GTM resources/ clusters) can be made available on request



11 Internal | AE/PAI-IP | 2023-12-06

© Robert Bosch GmbH 2023. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



Using virtualGTM with Coside Example for microcontroller with GTM Example: IFX TC4xx

GTM is a sub unit in the micro controller system



https://www.infineon.com/export/sites/default/_images/product/microcontroller/Aurix/TAURIX-TC4x-Evolution.png_1016265805.png





Using virtualGTM with Coside GTM only simulation environment

What will be available for use in Coside simulation

virtualGTM: Cycle accurate SystemC model of GTM_IP

GTM_IP is embedded in Coside module, which models simplified uC- functionalities:

- CPU with interrupt controller
- Local memory
- ADC's
- Analog functionality for High Resolution PWM



Internal | AE/PAI-IP | 2023-12-06

© Robert Bosch GmbH 2023. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as



Opportunities of/ with virtualGTM

Deep in depth debugging/ analysis of internal GTM functions Validation of realtime critical applications using multiple GTM internal resources Platform to train/ educate GTM functionality

> Bosch AE provides GTM examples for self study / modification

> > BOSCH

GTM MCS code development/ trouble shooting

Coside / Matlab simulink integration for system validation Easy adaptable due to SystemC and existing Coside libraries

C-Code developed on virtualGTM is usable in any uC EDA environment

Each GTM user will have the opportunity to enhance/adapt the usage of the virtualGTM to his needs/ purpose

15 Internal | AE/PAI-IP | 2023-12-06

© Robert Bosch GmbH 2023. All rights reserved, also regarding any disposal, exploitation, reproduction, edunt, distribution, as we'l as in the event of a plications on indus net for one rty rights.

Using virtualGTM with Coside Getting access



- COSEDA website for virtualGTM download
 - <u>https://www.coseda-tech.com/bosch-gtm-systemc-model</u>
- Bosch AE website for individual use case download
 - <u>https://www.bosch-semiconductors.com/ip-modules/gtm-platform/gtm-virtual-prototyping/</u>

Follow instructions in downloaded document: Install_virtualGTM.pdf





Using virtualGTM with Coside Starting with examples in cos_gtm_example



Using virtualGTM with Coside Typical schematic and C code of a training use case

G

🕻 COSIDE® Explorer 🗙 🏠 Project Explorer 🗧 🗖	🖻 mcs_atom_tb_simple_tb.cpp 🛛 🔂 *mcs_atom_tb_mcs0_ch0.wave 🛛 🔂 mcs_atom.cpp 🗙	🗖 🗖 😫 Outline 🗙 📲 Hierarchy	
	96		Testhench
m > bosch_gtm [bosch_gtm coseda 1.6.7_RMdpllext1 17]	98 // main thread:	쥠*mcs_atom_tb.scaml ×	
cos_gtm_example [cos_gtm_example sim_1.6.7_mcs_atom 1]	99	s_result_adc1	loophack allows to
> 🕞 atom_somp	2101 uint32 t = 0;	s s ready_adcl s s ready_adcl s s ready_adcl s s ready_adcl s s s ready_adcl s s s ready_adcl s s s s s s s s s s s s s s s s s s s	IUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
▶ 🕞 cmu_clk	<pre>GAL_INFO("Configure cluster clocks switch off cluster 1n");</pre>	s_result_adc0	
End configurations	$\begin{bmatrix} 103 & \text{int } \text{cls}_{\text{div}} = 1; \\ \hline \infty 104 & \text{GTM.cLs[0].ARCH.CTRL} = 0x0; \\ \end{bmatrix}$	i const arc sel a cost a	generate input
✓ gr > mcs_atom	GTM.CLS[0] ARCH.CLK_CFG = cls_div & 0x03 ;	add_ready add_ready add_ready add_ready	Beilerate input
👻 🐝 Binaries	$\begin{bmatrix} 106 \\ 0 \end{bmatrix} = 0 \\ \hline 0 \end{bmatrix} = 0 \\ \hline 0 \end{bmatrix} = 0 \\ \hline 0 \end{bmatrix} \\ \hline 0 \hline 0 \\ \hline 0 \end{bmatrix} \\ \hline 0 \hline \hline 0 \\ \hline 0 \hline 0$	i pwc src sc]	cignals for the
<pre>@ mcs_atom_tb_simple_tb.exe</pre>	108	<pre></pre>	Signals for the
→ A > Trace files	GAL_INFO("configure CMU");	gtm_tio0_g0_out_r gtm_tio0_g0_out_r gtm_tio0_g0_out_r	
<pre>> mcs_atom_tb_mcs0_ch0.wave</pre>	111 // Set Count Value for clock devider	init val = (7)10 gtm tin 0 instruction gtm atom0 out n	GIM by GIM
C mcs_atom_tb_object_trace.vcd	GTM.CLS[0].CMU.CLK 0 CTRL = CLK_RES_SCALE -1 ;	repeat = falgen .cdtm0_dtm3tgou n9_dtm1_aux_in	
✓ mcs_atom_to	113 114 // Enable CMU CLK0	gtm_cdtm0_dtm3fgu_m9_dm2_sov_m	internal recourses
mcs_atom_tb_simple_tb.cpp	<pre>\$115 GTM.CLS[0].CMU.CLK EN = 0 \$00000002;</pre>	gtm_cdtm0_dtm4tepumP_dtm4_oux_inCOS_GTM_CIIgtm_brc_ingbrc_ing	internal resources
mcs_atom_tb.b		gtm_cdtm0_dtm8tev_m2.sex_instdti_irq	
A mcs atom thiscam	<pre>Solution State ("configure TBU");</pre>	gtm_cdtmo_dtm?stepum?sten7_oux_in	
mcs atom tb.xml	GAL_INFO("======""""""""""""""""""""""""""""""	gun uou gu in gan uno in gan uno ing gun tino ing gan tino ing	• • • • • • • • • • • • • • •
▼ R > Others	121 // setup and enable TBU channel 0	gtim med ing gtim meso i	q \
axim.mem	T22 GTM.CLS[0].TBU.CH0 CTFL = 0x00; // no LOW_RES, chose CMU_CLK0	gtm tom0 irq gtm tom0.irq	
🕼 build.sh		gtm tio0 g0 irq gtm tio0 g0 irq gtm stom0 irg gtm stom0 irg	
📑 included.modules	125 #if CH_CNT == 1	dbg.mc20 hbp_irq dbg.mc30 hbp_irq	
🔒 library.description	126 GAL_INFO("configure Alow channel 3"); 127 GAL INFO("====================================	port_int_table = port_int_table_gtm4 test_program = mcs_atom	program
🗟 mcs_atom_core.cpp	128	i pwc src sc2 verder dev de umbrelle dev de	
💦 mcs_atom_core.h	129 // Enable update of compare register 130 GTM.CLS[0].ATOM.AGC.GLB_CTRL = 0x008000000:	<pre>sc_bice_3_2>(sc_bice_3_2>);gtm_tim0_in)</pre>	
mcs_atom_mcs_v4.cpp	131	#include "/configurations/port	_irq_table_gtm4,h"conv_del
la > mcs_atom_mcs_v4.h	132 // SOMP (Bit 1:0 \neq 0b10) 133 GTM CLS[0] ATOM CH[3] CTBL = 0x00000002: // c]k src = CML CLK 0	init vol = 42	(qtm atom0 out,7) $(boother tim0 in 7)$
M mcs_atom_mcs.log	134 #else	repeat = falce	
> mcs_atom_mcs.mcs	135 for(int i=0;i<(H_CNT;i++) {	(1_5C_US,0x00), (3_5C_US,0x35)	i convide2
mcs_atom_program.cpp	137 // Enable update of compare register	(* 30. US. 00.24) (* 35. US. 00.24) (* 35. US. 00.24)	$\langle \overline{bool}, b \Delta p \rangle$ $DE \rightarrow \delta E$
\sim > mcs_atom_programmin	[3138 GTM.CLS[0].ATOM.AGC.GLB_CTRL = 0x2 << (16 + 2*i);	(1) 5C_US(0(55), (10.5C_US(0x2a),	tm_atom0_out,6) == intermediate and the second seco
> mcs atom.h	140 // SOMP (Bit 1:0 = 0b10)	(11 SC US, by 28), (12 SC US, by 28), (13 SC US, by 28),	
Na > mcs_atom.log	[14] GTM.CLS[0].ATOM.CH[i].CTRL = 0x00000002; // clk_src = CMU_CLK_0	All inputs/ outputs	
📑 test_programs.h	142 } 143 #endif ••••		
	(I his code will be)	to a 1 cluster GTM	
	avacuted on a uC		
		can be used in	
Internal AE/PAI-IP 2023-12-06	core and configure	testhench hookun	
© Robert Bosch GmbH 2023. All rights reserved	also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in	the event of applications for industrial property rights.	
S Hoser Beech empirization in hjird rederred.	the GTM		

Using virtualGTM with Coside Typical wave gui of a use case

Hierarchical module



trace

COSIDE® Explorer X 🏠 Project Explorer 🖓 🗖	د mcs_atom_tb_simple_tb.cpp الله *mcs_atom_tb_mcs0_ch0.wave :	×			🗄 Outline 🗙 📲 Hierarchy	8 🗖 🗖
🗉 🖻 😫 🛃 👬	C . 0	2.5e-6 5e	-6 7.5e-6	10e-	type filter text	
main > bosch_gtm [bosch_gtm coseda_1.6.7_RMdpllext1 17]	SystemC.ATOM0_OUT(0) -				The most atom the object trace yed	
▼ 🚮 > cos_gtm_example [cos_gtm_example sim_1.6.7_mcs_atom ↑1]	SystemC.ATOM0_OUT(1) -				▼ [™] SystemC	
🕨 🕞 atom_somp	SystemC.ATOM0_OUT(2) -				▼ [™] i mcs atom tb	
🕨 📷 cmu_clk		CTM ID signals			v i cos gtm	
🕨 🕞 configurations	SystemC.ATOM0_OUT(4)	I G I W_IF SIGNAIS			i cos irq ctrl	
▼ 💏 > mcs_atom	SystemC.ATOM0_OUT(5) -				👻 🔅 i_gtm_top1	
👻 🎲 Binaries	SystemC.ATOM0_OUT(6) -				▼ 👺 atom0	
mcs_atom_tb_simple_tb.exe	SystemC.ATOM0_OUT(7) -				🕨 🏥 atom0_ch0	
▼ h/ > Trace files		(5000		🕨 ই atom0_ch1	
🎀 > mcs_atom_tb_mcs0_ch0.wave	i cos gtm.i gtm top1.mcs0.MCS CTRL STAT	'h02000000	X 'h000	00000	🕨 🔅 atom0_ch2	
<pre>mcs_atom_tb_object_trace.vcd</pre>					🕨 🔅 atom0_ch3	
▼ 🛃 > mcs_atom_tb	tm top].mcs0.MCS CTBL STATEN TIM FOUT				atom0_ch4	
mcs_atom_tb_simple_tb.cpp	ui gtm top].mcs0.MCS_CTBL_STAT:SCD_MODE				🕨 🤹 atom0_ch5	
mcs_atom_to.cpp	mi atm top1 mcs0 MCS_CTBL_STAT:SCD_CH	'h	0		atom0_ch6	
mes_atom_tb.caml	i gtm_top1_mcs0_MCS_CTRL_STAT:BAM_BST				atom0_ch7	
mcs_atom_tb.xml					AGC_ACT_TB	
▼	atm top1 mcs0 MCS CTRL STATERR SRC ID		0		AGC_ENDIS_CIRL	
axim.mem	thi cos atm i atm tanl mcs0 mcs0 ch0 PC	/ IP internal signals	×	'b0000134		
G build.sh	atmi.gtm_top1.mcs0.mcs0_cn0.pc -					
🔒 included.modules	a stm i stm topl mos0 mos0 sh0 instruge 2				AGC INT TRIG	
🔒 library.description		CS() channel()			AGC OUTEN CTRL	
🗟 mcs_atom_core.cpp						
🚹 mcs_atom_core.h		houddodd	houddood	Nobologia Nobologia	● Build Targets ×	
mcs_atom_mcs_v4.cpp	tm_top1.mcs0.mcs0_ch0.instr_reg.a_in_xoreg					-> 📐 👔
📑 > mcs_atom_mcs_v4.h	tm_top1.mcs0.mcs0_ch0.instr_reg.b_in_xoreg				(IDS (SINGLECORE)	-
🕅 mcs_atom_mcs.log	tb.i_cos_gtm.i_gtm_top1.mcs0.mcs0_ch0.ACB	'h000i	00000		() tb	
> mcs_atom_mcs.mcs	b.i_cos_gtm.i_gtm_top1.mcs0.mcs0_ch0.MHB	'h0000000	hoo X hooooooaa X hoo)	'h000000AA X 'h00 X 'h0000	▼ Scos_gtm_example	
] > mcs_atom_program.cpp	tb.i_cos_gtm.i_gtm_top1.mcs0.mcs0_ch0.R(0)	'h0000000	X 'h00000028 X	'h0000003C X 'h00000024	() all	
🔝 mcs_atom_program.h	tb.i_cos_gtm.i_gtm_top1.mcs0.mcs0_ch0.R(1)	'h0000000	X 'h00000014 X	1600000028 X 16000000A	Il (singlecore)	
ka > mcs_atom.cpp	tb.i_cos_gtm.i_gtm_top1.mcs0.mcs0_ch0.R(2)	'h000	00000		() clean	
ling > mcs_atom.n	tb.i_cos_gtm.i_gtm_top1.mcs0.mcs0_ch0.R(3)	'h0000000	X 'h00 Y 'h00AA0000 X 'h00)	'h00AA0000 X h00 X 'h00AA	(i) distclean	
Incs_atom.og Test programs b	tb.i_cos_gtm.i_gtm_top1.mcs0.mcs0_ch0.R(4)	'h0000000	X 'h000	0000008	() exe <lib file=""></lib>	
🔝 cosc programs.n	thi cos atmi atm top] mcc0 mcc0 ch0 P(S)	140000000	V - 160000588	thonorosca Vitkonon	(IDS (Singlecore)	
				· · · · · · · · · · · · · · · · · · ·		

Tracable signals



Using virtualGTM with Coside Typical application schematic and MCS code



🔂 signal	_generat	ion_fifo	o.cpp	📄 adc_trigger_mo	s.mcs ×	(· · · · · · · · · · · · · · · · · · ·	-
56							
57							
58							
59							
60	######	#####	ATOM Ch	annels for DTM-	ADC Trig	gger ###########	
61	## AT0	M Char	nnel 4				
62	movl	R0, 5	60		; R0 =	SR0	
63	movl	R1, 5	50 0		; R1 =	= SR1	
64	movl	R2, 0	xFFFF00			; R2 = CN0 (-> Offset)	
65	movl	R3, 2	2		;		
66							
67	bwr	R1, A	ТОМ_СН4	_SR1			
68	bwr	R0, A	ТОМ_СН4	_SR0			
69	bwr	R3, A	ТОМ_СН4	_CM1			
70	bwr	R0, A	ТОМ_СН4	_CM0			
71	bwr	R2, A	ТОМ_СН4	_CN0			
72							
73							
74			1				
/5	## ena	ble Al	OM-Chan	nels to start s	ynchrono	ous	
/6	movl	R5, 6	X000200	OUTEN OTAT			
//	DWF	R5, A	TOM_AGC	_OUTEN_STAT			
78	DWF	кз, А	ATOM_AGC	_ENDIS_STAT			
79							
80							
01	it ada						
02 L_W	arc_auc	во т					
0.0	addl	P2 2	00_150				
85	WILLEMY	R2 T	BU TSA				
86	WUTINX	112, 1	00_100				
87	## rea	d ADC	value a	nd check for ne	w data		
88 pol	l data:	a Abe	vacae a	ind check for he	waaca		
89	brd	R3. A	ADC CH0	DATA			
90	ibc	MHB.	6. poll	data	: MHB[6]] = ADC NEW DATA -> == 0 : no new va	l
91	Juna	,	-,		,		
92	## han	dle AD)C value				
93							
94	## jum	p to w	ait for	next ADC value			
95	imp	l wai	t adc				
96	Second C	-	-				
97							
98 l_d:	isable	ch:					
99	## dis	able M	1CS - Chan	nel			
100	movl	STA,	0×0				
101							



Using virtualGTM with Coside Typical application schematic wave forms





© Robert Bosch GmbH 2023. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Using virtualGTM with Coside Typical application schematic wave forms





© Robert Bosch GmbH 2023. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Thank You

Get in contact with us:

Mail: gregor.sunderdiek@de.bosch.com

https://www.bosch-semiconductors.com/ip-modules/gtm-platform/gtm-generic-timer-ipmodule/



