

Virtualisation in Use An AURIX/**T1** demo

Dipl.-Ing. (BA) **Peter Gliwa**, Gliwa GmbH Dipl.-Ing. **Jens Harnisch**, Infineon Technologies AG





- Motivation: analyse variants of a function
- How it works: virtual function substitution
- How it works: virtual function migration
- Summary



- Need to gather *actual performance data* in order to select between a number of variants of a function
 - Performance of the function itself
 - Impact of the function
 - stack usage
 - shared memory conflicts
 - cache usage
- A build, flash and run cycle takes several hours
- Virtual function substitution allows a large number of variants to be trialled with one build



- Any computationally intensive function will demonstrate the principles.
- Naïve algorithm tests successive number *N* to see if they have a factor such that $1 < \text{factor} <= \sqrt{N}$
- Alternative algorithm uses lazy, sparse Sieve of Eratosthenes with fewer arithmetic operations but more memory accesses
- Which performs better in a real system?



- Infineon AURIX with 3 TriCore CPUs:
 - CPU0, V1.6E (Efficiency) core, mostly event-driven schedule
 - CPU1, V1.6P (Performance) core, mostly periodic schedule
 - CPU2, V1.6P (Performance) core, reserved in this demo
- Tasking v4.1r1 TriCore compiler
- Gliwa *T1* triggers demo phases and visualises timing effects



Function substitution disabled





Function substitution enabled





Used for example analysis: T1 timing suite

le Edit Keve Book theb								
	S I Bi I T I det tax		2017		_			
base	• a deself sig / braceld	- C	a une	and in passion designed	1 m			
Charles I and Ch	A 08	N ON						
· Link o	Errolaut de la com	The Make (Mar Mart Chart, Dave Other Dave						I DD
king a)	- brokecrub-kacnight			1	-			Solen
	EISE SPI							S - CRIVE THE
143				141.5				* 🐨 en 35a (tim 11 attes)
scope	ISR CAN			1111			_	The interaction of the line of
hundrand out		-		6 - 1 - A				The beings and Che More
Owned for Discourse	DRIVE INS					-	+	www.age 380er 200er [1] access
- Inneres State			-	B. Contractor			-	sample ork (25 [71 (cope]
	DRIVE SNS		-	and the state of t			Day No.	# officiate
Show Traces Save Traces	(artis)		-		-			t in gots execute the
-	COMMON_10NS							a prescher
10 flex					-		_	a management and a second and a
topo poten amount	DRIVE_10MS							I perod
		_			0010			 After Shathid loss home instant of minut
	DRIVE_20MS				1000		1000	a w Longest the base to a set of a res
								 posed MVS_BC_MO_TH_0 posed MVS_BC_MO_TH_0
	DRIVE_SOMS							a - Diselectedary WEIR
Bentre line Score	ANJ							B. Soton Imag Information
pit	SPI handing							S monistr
	4	0		Are Stiller	time.	The Sile	11mm	and the second se
daubien Jest	time	Sile beine	311		1014			DRIVE INS.
Halt Measurement on Accura	<u>s</u>		_					- June 1
End of the State o	Selection Sylanvation							N a Ballbor
mail door of Press.	projecting entrope	CPUU	LINGS		476an 377ct	946741		188
ture data access on	5%	CPUU	usgs (including one-head	meteral .	0e	8 0083		
		N	and a second second	Running -	2(0PU) P	respond SWs	oom levet	
		DF	WE BHS	1324 9 6 6	30,540	AGus Citine 38er 1834		
A and Features		01	WE THS	75.8 Giller	15,041	456429441		1 1 1
A long	NY .	154	R GAN	\$5at 24245	12,986	Bra		
1 Delay		155	R 971 curstets 179	TJUE 20016	0.000	Bre		Wentstork and New
a cont			cash 180	0.0	0,000			Testes Dig. and executer 2010

- Runtime measuring, debugging, verification and optimisation
- System and code level timing analysis
- Oscilloscope-like visualisation of runtime scenarios
- Net run times for tasks, interrupts, functions or any code fragment
- CPU load measurement
- On-target measurement and supervision
- On-line instrumentation of code
- Easy connection to target hardware no HW modification required
- Interfaces to static code- and scheduling analysis tools
- Embedded in AUTOSAR processes





Results of substitution on CPU1



- Naïve Primes max CET = 825µs
- Sieve Primes max CET = 560µs
 - 32% reduction in execution time



- If we can substitute one function with another, we can just as well migrate the whole function to another core
- The optimised prime calculator is fast enough to run on CPU0's slower 1.6E core
- So let us migrate sievePrimes to CPU0, freeing CPU load on CPU1



Function migration









- Sieve Primes on CPU1 max CET = 560µs
- Sieve Primes on CPU0 max CET = 680µs
 CPU load increased by about 35%
- CPU0 can manage the extra CPU load
- Prioritised interrupts mean that CPU0 is unaffected with regard to previous interrupts, which have higher priority than the migration interrupt



- Using simple HW features, we can
 - substitute one function with another
 - migrate a function from one core to another
- The replacement function could equally be compiled, located and loaded to RAM *while the system is operating*