

Tips & Tricks > Trace > Utilizing TRACE32 Mixed-Signal Probe to Trace Task Switches in Real-Time Operating Systems (RTOS)

Utilizing TRACE32 Mixed-Signal Probe to Trace Task Switches in Real-Time Operating Systems (RTOS)

2024-12-13 - Comments (0) - Trace

Today, nearly every application runs on an operating system. To effectively analyze the timing behavior of the OS, it is crucial to trace the currently running task. Unfortunately, the tracing infrastructure of the core architecture sometimes lacks both data tracing and context ID tracing. In other cases, the available on-chip trace is so small that a thorough analysis of timing behavior is not possible.

For use cases like these, where long-term task runtime analysis is required, you can leverage GPIO signals available on the target board along with a TRACE32 Mixed-Signal Probe. For example, a solution has been developed for the PolarFire SoC (RISC-V, 64-bit) running FreeRTOS on a single U54 hart. While the PolarFire SoC offers on-chip and off-chip program flow trace, it provides neither data trace nor context ID tracing. FPGA-based SoCs typically have many GPIOs, which makes implementing such a solution easier.

The basic idea is to associate certain GPIO switches with the task switch operations of an RTOS. In this case, the FreeRTOS task switch hook function is utilized to add code that writes the task number to specific GPIOs.

```
480⊖ void gpio trace(uint32 t current task)
481 {
         GPIO TypeDef * gpio;
482
483
         uint32_t delay;
484
485
         gpio = GPIO2 LO;
486
487
         /* set 4 leds */
488
         gpio->GPI0_SET_BITS = ((uint32_t)0x0F << MSS_GPI0_16);</pre>
489
         for (delay = 0; delay < 10; delay++); // ~1us</pre>
490
491
         /* clear leds according with current task */
492
         gpio->GPIO CLR BITS = ((uint32 t)(current task & 7) << MSS GPIO 16);</pre>
493
         for (delay = 0; delay < 10; delay++); // ~1us</pre>
494
         /* toggle led 4 as clock */
495
496
         gpio->GPIO_CLR_BITS = ((uint32_t)0x01 << MSS_GPI0_19);</pre>
497 }
```

The GPIO signals can be recorded using a Lauterbach logic analyzer (Mixed-Signal Probe). A DisConfig command in TRACE32 allows the recorded GPIO signals to be mapped to the task number. To understand the following screenshots, it's important to know that the signals recorded by the Mixed-Signal Probe are displayed in TRACE32 using the TRACE32 CIProbe or IProbe main command groups. The commands IProbe.Chart.TASK and IProbe.STATistic.TASK enable now the desired long-term task runtime analysis.

B::IProbe.Chart.TASK								
🔑 Setup 👔 Grou	os 🔡 Config	Goto	🔒 Goto	🛉 Find 🔳	ln In Out	🕀 Full		
	. 5 (00s	-6.000s	-5.50	0 s	-5.000s	-4.50)0s
(un T Stac Que Sie	range k known) k mr Svc k kEater k ueCons k veDemo k							
B::IProbe.STATistic.TA	_ <	W > <						>
🔑 Setup 👔 Grou	ps 🚼 Config	, 🗾 Detailed	Resting	사 Chart 🛛 📕	Profile			
	tasks: 5.	to	otal: 6.51	.8s				
range	total r	nin	max	avr	count	rat	tio% 1%	2%
(unknown) Tmr Svc StackEater	0.000us 18.645us 265.095us	0.000us 18.645us 18.425us	- 18.645us 25.235us	- 18.645us 18.935us	0	. <	0.000% 0.001% ← 0.004% ←	

Now, for those interested in a deeper understanding of the solution, here are some additional details.

<

FreeRTOS provides a "task num" for each created task. It's a progressive number starting from 1.

277	<pre>#if (configUSE_TRACE_FACILITY == 1)</pre>
278	<code>UBaseType_t uxTCBNumber;</code> $/*<$ Stores a number that increments each time a TCB is created.
279	* It allows debuggers to determine when a task has been deleted
280	* and then recreated. */
281	UBaseType_t uxTaskNumber; /*< Stores a number specifically for use by third party trace code. */
282	#endif



The "task num" is encoded on several GPIOs during each task switch. In this implementation, three data GPIOs are used (allowing for up to seven available tasks) along with one clock GPIO to sample valid data on the Mixed-Signal Probe. This configuration effectively functions like a bus trace with a single clock line.

쁎 B::IProbe.Timing							
🌽 Setup 📑 Name 🔃 Goto	🛉 Find 🔳	In I Out 🕕	Full O Off	Arm	⊗ Init	Snapshot us	ed:
-4.37450	0000s		-4.374	450000s			
line							
1 IP.LED1		,					<u> </u>
			<u> </u>				_
0 IP.040							_
0 IP.05							×
< 11 > <							>
-							
H B::IProbe.Chart.TASK /Track							×
B::IProbe.Chart.TASK /Track	ſt Goto [1	. Goto 🎁 Find	l ∢⊡ ∙ In •⊡•	Out 🕕 Full			
B::IProbe.Chart.TASK /Track	₽ Goto ₽ 4500000s	. Goto 🏥 Find	l ∢⊡• In ▶⊡ • -4.37	Out • Full _			
B::IProbe.Chart.TASK /Track	Q Goto Q 45 00000s	. Goto 🏦 Finc	i 4⊡• in •⊡ -4.37	Out Full 4450000s			
B::IProbe.Chart.TASK /Track	Q Goto Q 4500000s	, Goto 🎽 Finc		Out			
B::IProbe.Chart.TASK /Track	Goto A	. Goto 🎽 ji Finc	I ▲D In →D -4.37 . I	Out			
B:IProbe.Chart.TASK /Track	G Goto A 4500000s	. Goto 🧃 Finc	i 4D in ▶D -4.37 	Out Full 4450000s			
B:IProbe.Chart.TASK /Track	€ Goto 4500000s	. Goto j j Finc	I ▲① In →① -4.37 	Out Full 4450000s			
B::IProbe.Chart.TASK /Track	€ Goto 4500000s	. Goto j finc		4450000s			
B::IProbe.Chart.TASK /Track	Goto Q	Goto jij Find	I 4D In ▶D -4.37	4450000s			

Several CIProbe/IProbe.DisConfig.CYcle commands are configured to detect write cycles, along with the appropriate 'Strobe', 'Address', and 'Data' fields:

- The Address is the task number
- The values from the GPIOs are used as the Strobe.
- The Data is encoded to retrieve the original FreeRTOS task number based on the "task num" encoded on the GPIOs.

PRIVATE & &task_mag	task_mag jic=TASK.	ic CONFIG (m	agic)								
; TASK2 1	s SieveD	emo									
NAME.Word	W.TASK2	IP.04 L	OW LOW LO	W HIGH H	IGH HIG	HIGH I	OW HIGH 1	LOW HIGH	LOW HIGH	LOW LOW LC	W
HIGH LOW	LOW LOW	LOW LOW	LOW LOW L	OW LOW H	IGH LOW	LOW LOW	LOW LOW	LOW LOW	LOW LOW 1	LOW LOW LOW	LOW
LOW LOW I	OW LOW L	OW LOW L	OW LOW LO	W LOW LO	W LOW LO	DW LOW I	OW LOW LO	DW LOW L	OM LOW LOW	W LOW LOW	
PRIVATE &	strobeta	sk2									
&strobeta	sk2="Str	obe TP.L	EDI LOW S	trobe TP	LED2 H	IGH Stro	be TP.LEI	D3 LOW"			
TProbe.Di	sConfig.	CYcle "w	r-mad.2"	Write S	trobe Il	P.CLK Ri	sing &sti	robetask	2 Address	LOW Addres	sBase
stask mag	ric Data	W TASK2	2 quad,2	11100 0	01000 1		.5111g 4501	Concourse	2 Hadress	Lon Maarco	obuoc
acask_mag	IC Data	W. IADRE									
and a second second											
B::IProbe.L	st										×
🔑 Setup	Config	🔒 Goto	👘 Find	- Chart	🚺 Profile	🚺 MIPS	More	Less			
record	run addr	ess		cycle	data		symbol			ti.back	
-00000058			D:080147C	0 wr-quad	00000000	08022AF0	freertos	\tasks\px	CurrentTCB	17.115us	^
-00000055			D:08014/C	0 wr-quad	00000000	080225E0	freertos	\tasks\px \tasks\px	CurrentICB	9.014ms	=
-00000049			D:080147C	0 wr-guad	00000000	080225E0	freertos	tasks\px	CurrentTCB	958.815us	\checkmark
-00000046			D:0801470	0 wr-quad	00000000	08022AF0	freertos	\tasks\px	CurrentTCB	17.110us	^
-00000043			D:080147C	0 wr-quad	00000000	080225E0	freertos	\tasks\px	CurrentTCB	9.993ms	
-00000040			D:0801470	0 wr-quad	00000000	08022AF0	treertos	\tasks\px	CurrentTCB	17.130us	
-00000036			D:08014/C	0 wr-quad	000000000	080225E0		\tasks\px \tasks\py	CurrentICB	9.995ms 17.120us	
-00000030			D:080147C	0 wr-guad	000000000	080225F0	freertos	tasks\px	CurrentTCB	9,993ms	
-00000027			D:0801470	0 wr-quad	00000000	080224E0	froortos	tacks pr	CurrentTCR	17 140uc	
-00000024			D.00014/C	o wi-quau	00000000	00022AFU		\Lasks \px	Currentics	1/.140u5	
00000024			D:080147C	0 wr-quad	00000000	080225E0	freertos	\tasks\px \tasks\px	CurrentTCB	9.993ms	
-00000020			D:080147C D:080147C D:080147C	0 wr-quad 0 wr-quad 0 wr-quad	00000000	080225E0	freertos	\tasks\px \tasks\px \tasks\px	CurrentTCB CurrentTCB	9.993ms 17.105us	
-00000020 -00000016			D:080147C D:080147C D:080147C D:080147C	0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad		08022AF0 080225E0 08022AF0 080225E0	freertos freertos freertos	\tasks\px \tasks\px \tasks\px \tasks\px	CurrentTCB CurrentTCB CurrentTCB CurrentTCB	9.993ms 17.105us 9.015ms	
-00000020 -00000016 -00000013 -0000008			D:080147C D:080147C D:080147C D:080147C D:080147C D:080147C	0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad		080224F0 080225E0 080224F0 080225E0 080224F0 080224F0	freertos freertos freertos freertos freertos	\tasks\px \tasks\px \tasks\px \tasks\px \tasks\px \tasks\px	CurrentTCB CurrentTCB CurrentTCB CurrentTCB CurrentTCB	9.993ms 17.105us 9.015ms 19.660us 13.600us	
-00000020 -00000016 -00000013 -00000008 -00000005			D:080147C D:080147C D:080147C D:080147C D:080147C D:080147C D:080147C	0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad		080225E0 080225E0 08022AF0 080225E0 08022AF0 080223080 080225E0	freertos freertos freertos freertos freertos freertos freertos	\tasks\px \tasks\px \tasks\px \tasks\px \tasks\px \tasks\px \tasks\px	CurrentTCB CurrentTCB CurrentTCB CurrentTCB CurrentTCB CurrentTCB CurrentTCB	9.993ms 17.105us 9.015ms 19.660us 13.600us 18.435us	
-00000020 -00000016 -00000013 -00000008 -00000005 -00000002			D:080147C D:080147C D:080147C D:080147C D:080147C D:080147C D:080147C D:080147C	0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad	00000000 00000000 00000000 00000000 0000	0080225E0 0080225E0 0080224F0 0080225E0 0080224F0 0080223080 0080225E0 0080224F0	freertos freertos freertos freertos freertos freertos freertos	<pre>\tasks \px \tasks \px</pre>	CurrentTCB CurrentTCB CurrentTCB CurrentTCB CurrentTCB CurrentTCB CurrentTCB CurrentTCB	9.993ms 9.9015ms 9.015ms 19.660us 13.600us 18.435us 17.065us	
-00000020 -00000016 -00000013 -00000008 -00000005 -00000002			D:080147C D:080147C D:080147C D:080147C D:080147C D:080147C D:080147C D:080147C	0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad 0 wr-quad		080225E0 080224E0 080225E0 080225E0 080224E0 080223080 080225E0 080225E0	freertos freertos freertos freertos freertos freertos freertos	<pre>\tasks \px \tasks \px</pre>	CurrentTCB CurrentTCB CurrentTCB CurrentTCB CurrentTCB CurrentTCB CurrentTCB	9.993ms 9.015ms 9.015ms 19.660us 13.600us 18.435us 17.065us	v

Please note that more than one DisConfig command is required—specifically, one command for each task. Each DisConfig command uses a distinct strobe to identify a specific task.

The information obtained from CIProbe/IProbe.List allows the standard FreeRTOS awareness to generate the CIProbe/IProbe.Chart.TASK view. This approach is designed for a single-core RTOS.

Additionally, the PolarFire SoC generates a program flow trace. Since both the PowerTrace (Analyzer), which records the program flow, and the Mixed-Signal Probe (IProbe/CIProbe), which captures the GPIOs, tag their trace records with the same - across all TRACE32 trace tools - timestamp, these recordings can be easily synchronized. The Analyzer.Chart Window below shows which functions are executed when the change from task SieveDemo to task StackEater occurred (blue cursor in both windows).

🕂 B::IProbe.Chart.TASK																												×
🔑 Setup 👔 Groups 🚦 Co	onfig.		U,	Goto	o	A	, Got	to		jij Fi	ind		•□•	In	•[]4	Out	: •	🕨 Fi	III									
	. 3	00	ns	-	391	. 20	Om	s	-	391	.1	00m	s	-	- 3 9:	1.0	00	ms	-	- 3 9	0.9	00	ms	- 3 9	10.8	300	ms	\square
range	•• 1					1					1					1					1				1			
(unknown)	•																											~
QueueCons													_ _	÷.			Ľ.											
SieveDemo														1														1
StackEater	•															Ĺ.												
	· ·	•				•			-										•				•	•				
	1																											0
	10																1	1										
<u> </u>	1.	ш	2	٩.																							~	.::

B::Analyzer.Chart /Track									×
🌽 Setup 👖 Groups 🚼 Config 🔒 Goto) (Goto	👘 Find	•D• In →D•	Out 🕕 Full				
	00ms	-388	3.200ms	-388.000	ms - 38	7.800ms	-387.	600ms	
address 🕢			1 .			1			
eertos\Global\handle_synchronous					11 .				~
\Global\test_if_environment_call					<u> U</u> .				
reertos (tasks (vlaskSwitchContext									
TS_gp10_Treertos\Global\Cl2dl2									
pio_freertos\gueue\vOueuePeceive									
ertos/queue/prvCopyDataEromOueue					1				
apio freertos\midi\vQueueConsume			•						
freertos\midi\vOueueConsumeHook					1				
o_freertos\queue\prvIsQueueEmpty				· · · · · ·					
_gpio_freertos\tasks\vTaskResume									
tos\tasks\prvTaskIsTaskSuspended 🚻									
gpio_freertos\tasks\vTaskSuspend				. !!					
gpio_freertos\midi\vStackEatTask									_
s_gpio_freertos\midi\vStackEater				. !					
mpts_gplo_treertos\Global\memset []									
pro_rreercos\rist\VL1stInsertEnd				· · ·					
asks (pr vkesechekchaskonbrockrine	< III								× .
	、 Ш	/ `							