



arm

EL3 Exception vector improvements

Manish Pandey, TF-A Tech Forum
13-07-2023

AArch64 exception vector table

- When exception occurs, PE must execute handler corresponding to exception.
- The location in memory where the handler is stored is called the *exception vector*.
- For ARM architecture, exception vectors are stored in a table, called the *exception vector table*.
- Each EL has its own vector table.
- VBAR_ELn register stores the base of vector table.
- + In these slides we will be focusing on Exception from Lower EL (EL3 point of view).

Address	Exception type	Description
VBAR_ELn + 0x000	Synchronous	Current EL with SP0
+ 0x080	IRQ/vIRQ	
+ 0x100	FIQ/vFIQ	
+ 0x180	SError/vSError	Current EL with SPx
+ 0x200	Synchronous	
+ 0x280	IRQ/vIRQ	
+ 0x300	FIQ/vFIQ	
+ 0x380	SError/vSError	Lower EL using AArch64
+ 0x400	Synchronous	
+ 0x480	IRQ/vIRQ	
+ 0x500	FIQ/vFIQ	
+ 0x580	SError/vSError	Lower EL using AArch32
+ 0x600	Synchronous	
+ 0x680	IRQ/vIRQ	
+ 0x700	FIQ/vFIQ	
+ 0x780	SError/vSError	

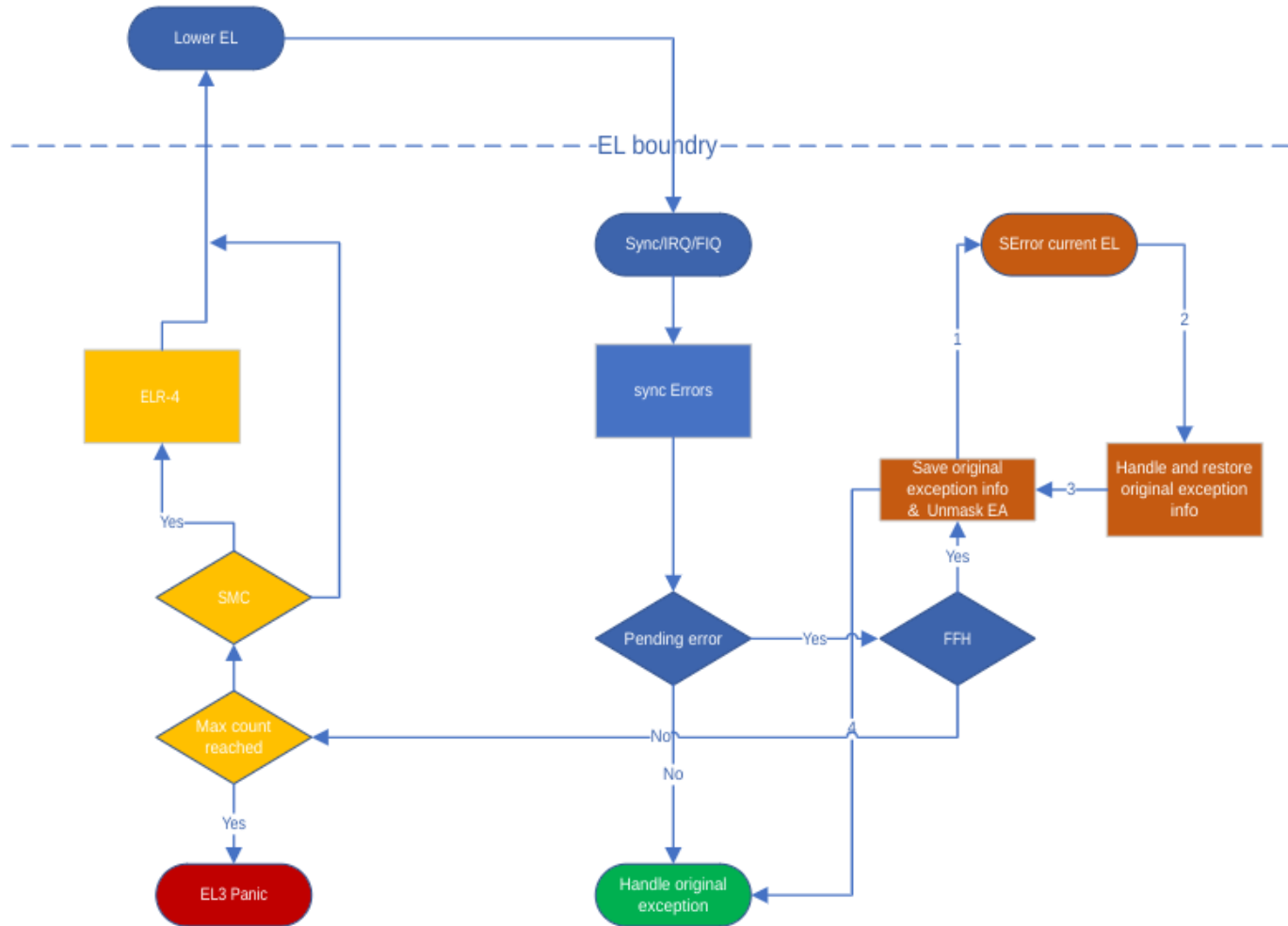
Lower EL exception handling Models

- + Lower EL can route exceptions to EL3 by setting SCR_EL3 (EA, FIQ, IRQ bits).
- + We are interested in EA bit, When this bit is
 - Set, External aborts and SErrors are routed to EL, Firmware First handling (FFH)
 - Else, they are handled in lower EL itself, Kernel First handling (KFH)
- + FFH and KFH paradigm is only applicable to Non-secure lower ELs, Secure/Realm lower EL's are always KFH.
- + The default model in TF-A is KFH, to enable FFH platform need to set "HANDLE_EA_EL3_FIRST_NS" macro (sets SCR_EL3.EA for NS).
- + RAS errors signalled by EA can also use above feature to route errors to EL3.
- + RAS support in TF-A need "ENABLE_FEAT_RAS" to be enabled which defaults to KFH
- + To get FFH support enable "RAS_FFH_SUPPORT" macro.

Error Synchronization during EL3 entry

- + On exception boundaries, we must synchronize errors (async EA) to isolate them.
- + There may be errors in the system which are not yet signalled to PE.
- + When Synchronizing errors, it triggers at incoming EL even though cause of error was in outgoing EL.
- + Following above reasoning EL3 should also ensure that errors pertaining to lower EL are identified in its vector entries.
- + On detecting a pending async EA
 - Reflect it back to lower EL (KFH)
 - Handle the EA in EL3 (FFH)
 - In both the scenarios, original exception which caused EL3 entry is deferred.
 - + For FFH it is handled after synchronized EA is handled (nested Exception handling in EL3)
 - + For KFH, Reflection is done without handling original exception.

Lower EL Sync/IRQ/FIQ exceptions



- + For KFH reflection, If the priority of EA is lower than original exception there will be back and forth. Max count is used to break from the loop
- + In future, if EL3 gets capability of virtual SError injection this will be solved.

Miscellaneous







- + For SError exceptions from lower EL
 - It is only expected in FFH mode
 - Hitting this vector for KFH mode is unexpected, panic
- + Error synchronization instruction
 - Without RAS extension, use "dsb"
 - With RAS extension
 - + we could use "esb" The problem with this instruction is, along with synching errors it might also consume the error. Which is not ideal for KFH mode.
 - + Use FEAT_IESB which provides controls to insert an implicit Error synchronization event at exception entry and exception return.
 - + Assumption in TF-A is, if RAS extension is present, we assume FEAT_IESB is also implemented (any concerns??)
- + Patches under review
 - [https://review.trustedfirmware.org/q/topic:%22mp%252Fseerror_reflection%22+\(status:open%20OR%20status:merged\)](https://review.trustedfirmware.org/q/topic:%22mp%252Fseerror_reflection%22+(status:open%20OR%20status:merged))

Testing on FVP

+ Test description

- Fvp-ea-ffh : FFH mode Without RAS extension (PLATFORM_TEST_EA_FFH)
- Fvp-ras-kfh : Basic KFH testing, no EL3 involvement
- **FVp-ras-kfh-reflect** : special test, explained in next slide
- Fvp-single-fault : RAS FFH mode test, gracefully handle error PLATFORM_TEST_RAS_FFH.
- Fvp-uncontainable : Uncontainable error, panic
- **FVP-ras-ffh-nested** : special test, explained in further slide

+ Special tests use few CI patches which are applied on the fly.

Test Group	TF Build Config	TFTF Build Config	Run Config	Status
tf-l3-boot-tests-ras	fvp-ras-ffh	fvp-single-fault	fvp-tftf-fip.tftf-aemv8a.fi-debug	SUCCESS <input type="checkbox"/> 
		fvp-uncontainable	fvp-tftf.fault-fip.tftf-aemv8a.fi-debug	SUCCESS <input type="checkbox"/> 
		fvp-ras-ffh-nested	fvp-tftf-fip.tftf-ras_ffh_nested-aemv8a.fi-debug	SUCCESS <input type="checkbox"/> 
	fvp-ras-kfh	fvp-ras-kfh-reflect	fvp-tftf-fip.tftf-ras_kfh_reflection-aemv8a.fi-debug	SUCCESS <input type="checkbox"/> 
		fvp-ras-kfh	fvp-tftf-fip.tftf-aemv8a.fi-debug	SUCCESS <input type="checkbox"/> 
	fvp-ea-ffh	fvp-ea-ffh	fvp-tftf-fip.tftf-aemv8a-debug	SUCCESS <input type="checkbox"/> 

KFH reflection test

- + Works in conjunction with "**ras_kfh_reflection.patch**"
- + One test each for IRQ and SMC, there is SError handler registered in TF-A tests
- + For IRQ reflection
 - Make SMC version call along with changing SCR_EL3.EA = 1 (applied through CI patch)
 - Disable SError and inject RAS error and wait till it gets pended
 - Inject IRQ
 - On EL3 entry there will be a pending EA, which reflected to TF-A tests
 - During ERET change (SCR_EL3.EA = 0).
 - Entering TF-A tests there will be pending IRQ and pending SError. On FVP IRQ gets handled first and then SError handler
- + SMC is like above except
 - ERET to ELR-4
 - On entering TF-A tests, only SError pending, which gets handled first and then original SMC version gets printed

FFH nested exception

- Works in conjunction with **ras_ffh_nested.patch**
- Register/enable SDEI event notification for RAS error.
- Make an SMC version call along with that it also changes SCR_EL3.EA=0 to route SError to TFTF (temporarily). This allow SError to be pended when next SMC call is made.
- + Disable SError (PSTATE.A= 1) and inject RAS error
- + When SError is pended make SMC call
- + On entering EL3, change "SCR_EL3.EA = 1"
- + On finding pending EA : save ESR_EL3, ELR_EL3, SPSR_EL3 & LR to restore original exception later on.
- + Unmask EA (PSTATE.A), which will cause SError exception for current EL.
- + Call platform handler to handle lower EL EA exception handler and eret back to original exception.

arm

Thank You

Danke

Gracias

Grazie

谢谢

ありがとう

Asante

Merci

감사합니다

धन्यवाद

Kiitos

شكراً

ধন্যবাদ

תודה