

[about](#) [summary](#) [refs](#) [log](#) [tree](#) [commit](#) [diff](#) [stats](#)

log msg



search

author Shawn Guo <shawn.guo@linaro.org> 2022-10-12 20:13:53 +0800
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2022-11-25 17:45:43 +0100
commit [0a3160f4ffc70ee4bfa1521f698dace06e6091fd](#) (patch)
tree [145ff42fe13d184c58b78d21a38fe8c98c7b54f3](#)
parent [7e1f908e65c56b06c2ccdc1c8c8034bbb1e2de62](#) (diff)
download [linux-0a3160f4ffc70ee4bfa1521f698dace06e6091fd.tar.gz](#)

diff options

context:

3 ▼

space:

include ▼

mode:

unified ▼

serial: imx: Add missing .thaw_noirq hook

[Upstream commit 4561d8008a467cb05ac632a215391d6b787f40aa]

The following warning is seen with non-console UART instance when system hibernates.

```
[ 37.371969] -----[ cut here ]-----
[ 37.376599] uart3_root_clk already disabled
[ 37.380810] WARNING: CPU: 0 PID: 296 at drivers/clock/clock.c:952 clk_core_disable+0xa4/0xb0
...
[ 37.506986] Call trace:
[ 37.509432] clk_core_disable+0xa4/0xb0
[ 37.513270] clk_disable+0x34/0x50
[ 37.516672] imx_uart_thaw+0x38/0x5c
[ 37.520250] platform_pm_thaw+0x30/0x6c
[ 37.524089] dpm_run_callback.constprop.0+0x3c/0xd4
[ 37.528972] device_resume+0x7c/0x160
[ 37.532633] dpm_resume+0xe8/0x230
[ 37.536036] hibernation_snapshot+0x288/0x430
[ 37.540397] hibernate+0x10c/0x2e0
[ 37.543798] state_store+0xc4/0xd0
[ 37.547203] kobj_attr_store+0x1c/0x30
[ 37.550953] sysfs_kf_write+0x48/0x60
[ 37.554619] kernfs_fop_write_iter+0x118/0x1ac
[ 37.559063] new_sync_write+0xe8/0x184
[ 37.562812] vfs_write+0x230/0x290
[ 37.566214] ksys_write+0x68/0xf4
[ 37.569529] __arm64_sys_write+0x20/0x2c
[ 37.573452] invoke_syscall.constprop.0+0x50/0xf0
[ 37.578156] do_el0_svc+0x11c/0x150
[ 37.581648] el0_svc+0x30/0x140
[ 37.584792] el0t_64_sync_handler+0xe8/0xf0
[ 37.588976] el0t_64_sync+0x1a0/0x1a4
[ 37.592639] ---[ end trace 56e22eec54676d75 ]---
```

On hibernating, pm core calls into related hooks in sequence like:

```
.freeze
.freeze_noirq
.thaw_noirq
.thaw
```

With `.thaw_noirq` hook being absent, the clock will be disabled in a unbalanced call which results the warning above.

```
imx_uart_freeze()
    clk_prepare_enable()
imx_uart_suspend_noirq()
    clk_disable()
imx_uart_thaw
    clk_disable_unprepare()
```

Adding the missing `.thaw_noirq` hook as `imx_uart_resume_noirq()` will have the call sequence corrected as below and thus fix the warning.

```
imx_uart_freeze()
    clk_prepare_enable()
imx_uart_suspend_noirq()
    clk_disable()
imx_uart_resume_noirq()
    clk_enable()
imx_uart_thaw
    clk_disable_unprepare()
```

Fixes: 09df0b3464e5 ("serial: imx: fix endless loop during suspend")

Reviewed-by: Martin Kaiser <martin@kaiser.cx>

Signed-off-by: Shawn Guo <shawn.guo@linaro.org>

Link: <https://lore.kernel.org/r/20221012121353.2346280-1-shawn.guo@linaro.org>

Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

```
-rw-r--r-- drivers/tty/serial/imx.c 1
```

1 files changed, 1 insertions, 0 deletions

diff --git a/drivers/tty/serial/imx.c b/drivers/tty/serial/imx.c

index cf3d531657762e..164597e2e00443 100644

--- a/drivers/tty/serial/imx.c

+++ b/drivers/tty/serial/imx.c

```
@@ -2626,6 +2626,7 @@ static const struct dev_pm_ops imx_uart_pm_ops = {
    .suspend_noirq = imx_uart_suspend_noirq,
    .resume_noirq = imx_uart_resume_noirq,
    .freeze_noirq = imx_uart_suspend_noirq,
+   .thaw_noirq = imx_uart_resume_noirq,
    .restore_noirq = imx_uart_resume_noirq,
    .suspend = imx_uart_suspend,
    .resume = imx_uart_resume,
```