author wuxu.wu <wuxu.wu@huawei.com> 2020-01-01 11:39:41 +0800
committer Mark Brown <broonie@kernel.org> 2020-01-03 00:59:40 +0000
commit 19b61392c5a852b4e8a0bf35aecb969983c5932d (patch)
tree ea43e3a3d0498d517ea7c6b7a9c3d70c2bcb4b8e
parent ca59d5a51690d5b9340343dc36792a252e9414ae (diff)
download linux-19b61392c5a852b4e8a0bf35aecb969983c5932d.tar.gz

diff options
context: 3
space: include
mode: unified

spi: spi-dw: Add lock protect dw_spi rx/tx to prevent concurrent calls

```
dw_spi_irq() and dw_spi_transfer_one concurrent calls.

I find a panic in dw_writer(): txw = *(u8 *)(dws->tx), when dw->tx==null,
dw->len==4, and dw->tx_end==1.

When tpm driver's message overtime dw_spi_irq() and dw_spi_transfer_one
may concurrent visit dw_spi, so I think dw_spi structure lack of protection.

Otherwise dw_spi_transfer_one set dw rx/tx buffer and then open irq,
store dw rx/tx instructions and other cores handle irq load dw rx/tx
instructions may out of order.

        [ 1025.321302] Call trace:
        ...
        [ 1025.321319]  __crash_kexec+0x98/0x148
        [ 1025.321323]  panic+0x17c/0x314
        [ 1025.321329]  die+0x29c/0x2e8
        [ 1025.321334]  die_kernel_fault+0x68/0x78
        [ 1025.321337]  __do_kernel_fault+0x90/0xb0
        [ 1025.321346]  do_page_fault+0x88/0x500
        [ 1025.321347]  do_translation_fault+0xa8/0xb8
        [ 1025.321349]  do_mem_abort+0x68/0x118
        [ 1025.321351]  el1_da+0x20/0x8c
        [ 1025.321362]  dw_writer+0xc8/0xd0
        [ 1025.321364]  interrupt_transfer+0x60/0x110
        [ 1025.321365]  dw_spi_irq+0x48/0x70
        ...

Signed-off-by: wuxu.wu <wuxu.wu@huawei.com>
Link: https://lore.kernel.org/r/1577849981-31489-1-git-send-email-wuxu.wu@huawei.com
Signed-off-by: Mark Brown <broonie@kernel.org>
```

Diffstat

-rw-r--r-- drivers/spi/spi-dw.c 15
-rw-r--r-- drivers/spi/spi-dw.h 1

2 files changed, 13 insertions, 3 deletions

```
diff --git a/drivers/spi/spi-dw.c b/drivers/spi/spi-dw.c
index 9387f60eb496e..c547ae38ed697 100644
--- a/drivers/spi/spi-dw.c
+++ b/drivers/spi/spi-dw.c
@@ -172,9 +172,11 @@ static inline u32 rx_max(struct dw_spi *dws)

 static void dw_writer(struct dw_spi *dws)
 {
-       u32 max = tx_max(dws);
+       u32 max;
        u16 txw = 0;

+       spin_lock(&dws->buf_lock);
+       max = tx_max(dws);
        while (max--) {
                /* Set the tx word if the transfer's original "tx" is not null */
                if (dws->tx_end - dws->len) {
@@ -186,13 +188,16 @@ static void dw_writer(struct dw_spi *dws)
                dw_write_io_reg(dws, DW_SPI_DR, txw);
                dws->tx += dws->n_bytes;
        }
+       spin_unlock(&dws->buf_lock);
 }

 static void dw_reader(struct dw_spi *dws)
 {
-       u32 max = rx_max(dws);
+       u32 max;
        u16 rxw;

+       spin_lock(&dws->buf_lock);
+       max = rx_max(dws);
        while (max--) {
                rxw = dw_read_io_reg(dws, DW_SPI_DR);
                /* Care rx only if the transfer's original "rx" is not null */
@@ -204,6 +209,7 @@ static void dw_reader(struct dw_spi *dws)
                dws->rx += dws->n_bytes;
        }
+       spin_unlock(&dws->buf_lock);
 }

 static void int_error_stop(struct dw_spi *dws, const char *msg)
@@ -276,18 +282,20 @@ static int dw_spi_transfer_one(struct spi_controller *master,
 {
        struct dw_spi *dws = spi_controller_get_devdata(master);
        struct chip_data *chip = spi_get_ctldata(spi);
+       unsigned long flags;
        u8 imask = 0;
        u16 txlevel = 0;
        u32 cr0;
        int ret;

        dws->dma_mapped = 0;
-
+       spin_lock_irqsave(&dws->buf_lock, flags);
        dws->tx = (void *)transfer->tx_buf;
        dws->tx_end = dws->tx + transfer->len;
        dws->rx = transfer->rx_buf;
        dws->rx_end = dws->rx + transfer->len;
        dws->len = transfer->len;
+       spin_unlock_irqrestore(&dws->buf_lock, flags);

        spi_enable_chip(dws, 0);
```

```
@@ -470,6 +478,7 @@ int dw_spi_add_host(struct device *dev, struct dw_spi *dws)
        dws->type = SSI_MOTO_SPI;
        dws->dma_inited = 0;
        dws->dma_addr = (dma_addr_t)(dws->paddr + DW_SPI_DR);
+       spin_lock_init(&dws->buf_lock);

        spi_controller_set_devdata(master, dws);
```

diff --git a/drivers/spi/spi-dw.h b/drivers/spi/spi-dw.h
index 38c7de1f0aa94..1bf5713e047d3 100644
--- a/drivers/spi/spi-dw.h
+++ b/drivers/spi/spi-dw.h

```
@@ -119,6 +119,7 @@ struct dw_spi {
        size_t                  len;
        void                    *tx;
        void                    *tx_end;
+       spinlock_t              buf_lock;
        void                    *rx;
        void                    *rx_end;
        int                     dma_mapped;
```