I.MX IEEE 1588 1PPS TEST

MAGGIE JIANG

2022 JULY

Background

- Some customer need to use IEEE 1588 1PPS signal. (PPS= Pulse per second)
- This article take i.MX8MP as example, because i.MX8MP have two ethernet ports: one is normal ENET port, another is EQOS port which also support TSN.

Two Ethernet controllers, capable of simultaneous operation

- One Gigabit Ethernet controller with support for EEE, Ethernet AVB and IEEE1588
- One Gigabit Ethernet controller with support for TSN, EEE, Ethernet AVB and IEEE1588

- For normal FEC port test procedure. it is also suitable for other i.MX silicon, including i.MX6, i.MX8QM, i.MX8QXP, i.MX8M, etc.
- HW environment: i.MX8MP EVK
- SW environment : L5.10.72_2.2.0

1588 output PIN

- ENET port 1588 output pin

ENET1_1588_EVENT0_OUT	SAI1_RXC	ALT4
ENET1_1588_EVENT1_OUT	SAI1_RXD1	ALT4

- EQOS port 1588 output pin

ENET_QOS_1588_EVENT0_OUT	GPIO1_IO09	ALT1
ENET_QOS_1588_EVENT1_OUT	I2C2_SDA	ALT1
ENET_QOS_1588_EVENT2_OUT	SAI2_RXD0	ALT2
ENET_QOS_1588_EVENT3_OUT	SAI2_TXFS	ALT2

Add FEC port 1588 output support in dts

imx8mp-evk.dts：

```
pinctrl_fec: fecgrp {
        fsl,pins = <
            MX8MP_IOMUXC_SAI1_RXD2__ENET1_MDC             0x3
            MX8MP_IOMUXC_SAI1_RXD3__ENET1_MDIO            0x3
            MX8MP_IOMUXC_SAI1_RXD4__ENET1_RGMII_RD0       0x91
            MX8MP_IOMUXC_SAI1_RXD5__ENET1_RGMII_RD1       0x91
            MX8MP_IOMUXC_SAI1_RXD6__ENET1_RGMII_RD2       0x91
            MX8MP_IOMUXC_SAI1_RXD7__ENET1_RGMII_RD3       0x91
            MX8MP_IOMUXC_SAI1_TXC__ENET1_RGMII_RXC        0x91
            MX8MP_IOMUXC_SAI1_TXFS__ENET1_RGMII_RX_CTL    0x91
            MX8MP_IOMUXC_SAI1_TXD0__ENET1_RGMII_TD0       0x1f
            MX8MP_IOMUXC_SAI1_TXD1__ENET1_RGMII_TD1       0x1f
            MX8MP_IOMUXC_SAI1_TXD2__ENET1_RGMII_TD2       0x1f
            MX8MP_IOMUXC_SAI1_TXD3__ENET1_RGMII_TD3       0x1f
            MX8MP_IOMUXC_SAI1_TXD4__ENET1_RGMII_TX_CTL    0x1f
            MX8MP_IOMUXC_SAI1_TXD5__ENET1_RGMII_TXC       0x1f
            MX8MP_IOMUXC_SAI1_RXD0__GPIO4_IO02            0x19
    +       MX8MP_IOMUXC_SAI1_RXC__ENET1_1588_EVENT0_OUT 0xd6
```

Add FEC port 1588 output support in FEC driver

```
--- a/drivers/net/ethernet/freescale/fec_ptp.c
+++ b/drivers/net/ethernet/freescale/fec_ptp.c
@@ -185,7 +185,8 @@ static int fec_ptp_enable_pps(struct fec_enet_private *fep, uint enable)
            val |= (1 << FEC_T_TF_OFFSET | 1 << FEC_T_TIE_OFFSET);
            val &= ~(1 << FEC_T_TDRE_OFFSET);
            val &= ~(FEC_T_TMODE_MASK);
-           val |= (FEC_HIGH_PULSE << FEC_T_TMODE_OFFSET);
+           //val |= (FEC_HIGH_PULSE << FEC_T_TMODE_OFFSET);
+           val |= (FEC_TMODE_TOGGLE << FEC_T_TMODE_OFFSET);
```

Because the default BSP uses HIGH_PULSE setting, the pulse width is 8ns. We modify to set as TOGGLE mode for the test. Then it's easy to capture signal with normal oscilloscope.

FEC port 1588 1PPS test procedure

root@imx8mpevk:~# **ptp4l -A -4 -H -m -i eth0 &**

[2] 837

ptp4l[58.048]: selected /dev/ptp0 as PTP clock

ptp4l[58.049]: port 1: INITIALIZING to LISTENING on INIT_COMPLETE

ptp4l[58.049]: port 0: INITIALIZING to LISTENING on INIT_COMPLETE

ptp4l[58.049]: port 1: link down

ptp4l[58.049]: port 1: LISTENING to FAULTY on FAULT_DETECTED (FT_UNSPECIFIED)

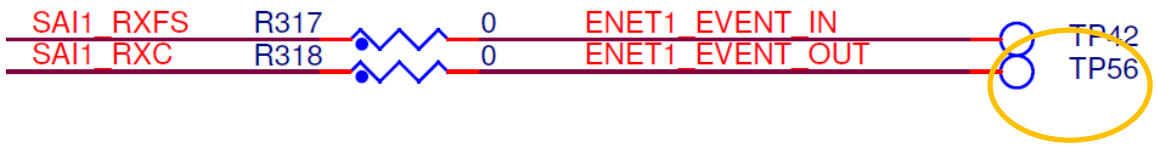
ptp4l[58.049]: selected local clock 00049f.fffe.070f72 as best master

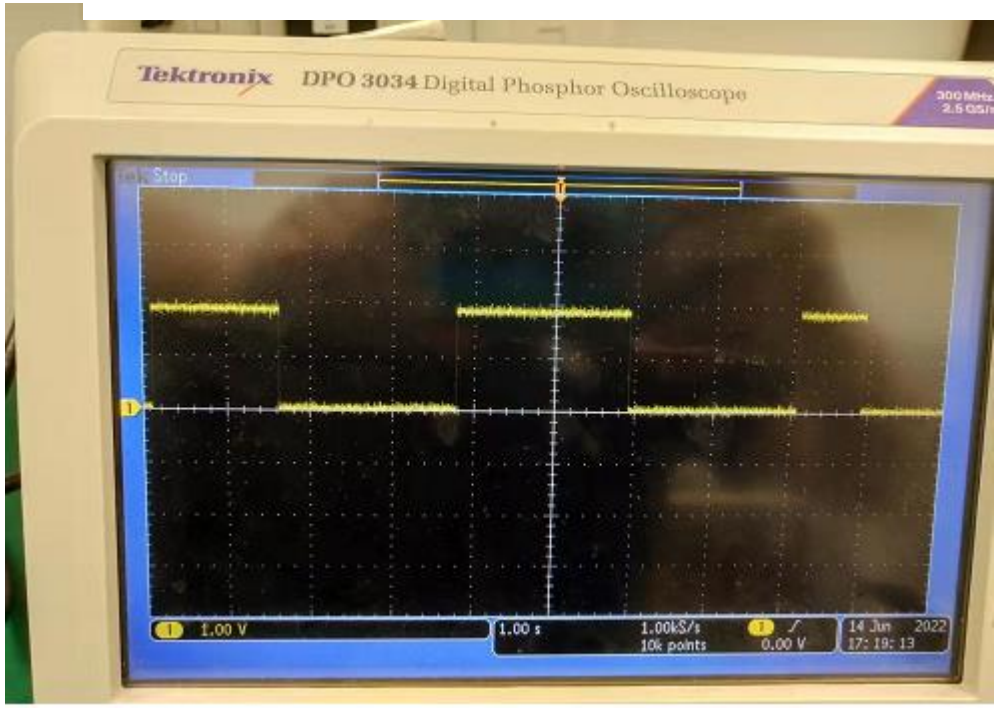
ptp4l[58.049]: port 1: assuming the grand master role

root@imx8mpevk:~# **echo 1 > /sys/class/ptp/ptp0/pps_enable**

i.MX8MP FEC port test point

- i.MX8MP EVK test point is following TP56 in baseboard:





Other i.MX serials IEEE 1588 1PPS test tips

For other i.MX which support 1588 FEC port, such as i.MX6, i.MX8QXP/QM, i.MX8M serials, etc

1. dts setting is different on 1588 ENENT out pin. Please refer to RM for it.

2. Driver part modification is same on : drivers/net/ethernet/freescale/fec_ptp.c

3. This demo test command line use eth0, because FEC port is first ethernet port in i.MX8MP and both ports are loading. Be careful of this sequence.

root@imx8mpevk:~# ptp4l -A -4 -H -m -i eth0 &

root@imx8mpevk:~# echo 1 > /sys/class/ptp/ptp0/pps_enable

imx8mp.dtsi :

```
aliases {
        ethernet0 = &fec;
        ethernet1 = &eqos;
```

i.MX8MP EQOS PORT 1588 output in dts

imx8mp-evk.dts：

```
pinctrl_eqos: eqosgrp {
        fsl,pins = <
            ……
            MX8MP_IOMUXC_ENET_RD2__ENET_QOS_RGMII_RD2      0x91
            MX8MP_IOMUXC_ENET_RD3__ENET_QOS_RGMII_RD3      0x91
            MX8MP_IOMUXC_ENET_RXC__CCM_ENET_QOS_CLOCK_GENERATE_RX_CLK      0x91
            MX8MP_IOMUXC_ENET_RX_CTL__ENET_QOS_RGMII_RX_CTL 0x91
            MX8MP_IOMUXC_ENET_TD0__ENET_QOS_RGMII_TD0      0x1f
            MX8MP_IOMUXC_ENET_TD1__ENET_QOS_RGMII_TD1      0x1f
            MX8MP_IOMUXC_ENET_TD2__ENET_QOS_RGMII_TD2      0x1f
            MX8MP_IOMUXC_ENET_TD3__ENET_QOS_RGMII_TD3      0x1f
            MX8MP_IOMUXC_ENET_TX_CTL__ENET_QOS_RGMII_TX_CTL 0x1f
            MX8MP_IOMUXC_ENET_TXC__CCM_ENET_QOS_CLOCK_GENERATE_TX_CLK      0x1f
            MX8MP_IOMUXC_SAI2_RXC__GPIO4_IO22           0x19
    +       MX8MP_IOMUXC_GPIO1_IO09__ENET_QOS_1588_EVENT0_OUT 0xd6
        >;
pinctrl_i2c2_synaptics_dsx_io: synaptics_dsx_iogrp {
        fsl,pins = <
    -       MX8MP_IOMUXC_GPIO1_IO09__GPIO1_IO09          0x16
```

i.MX8MP EQOS PORT 1588 1pps support in driver

- EQOS port driver is different with normal FEC port, following is driver location, which need to add additional patch to support 1588 1pps test.
- drivers/net/ethernet/stmicro/stmmac/stmmac_main.c

- 0001-net-stmmac-retain-PTP-clock-time-during-SIOCSHWTSTAM.patch

This patch has three modifications, it will fail in stmmac_platform.c, just ignore it. It is OK to just apply stmmac.h and stmmac_main.c.

```
drivers/net/ethernet/stmicro/stmmac/stmmac.h   |   1 +
.../net/ethernet/stmicro/stmmac/stmmac_main.c | 125 ++++++++++++++-------
.../ethernet/stmicro/stmmac/stmmac_platform.c |   2 +-
3 files changed, 81 insertions(+), 47 deletions(-)
```

i.MX8MP EQOS PORT 1588 1pps test procedure

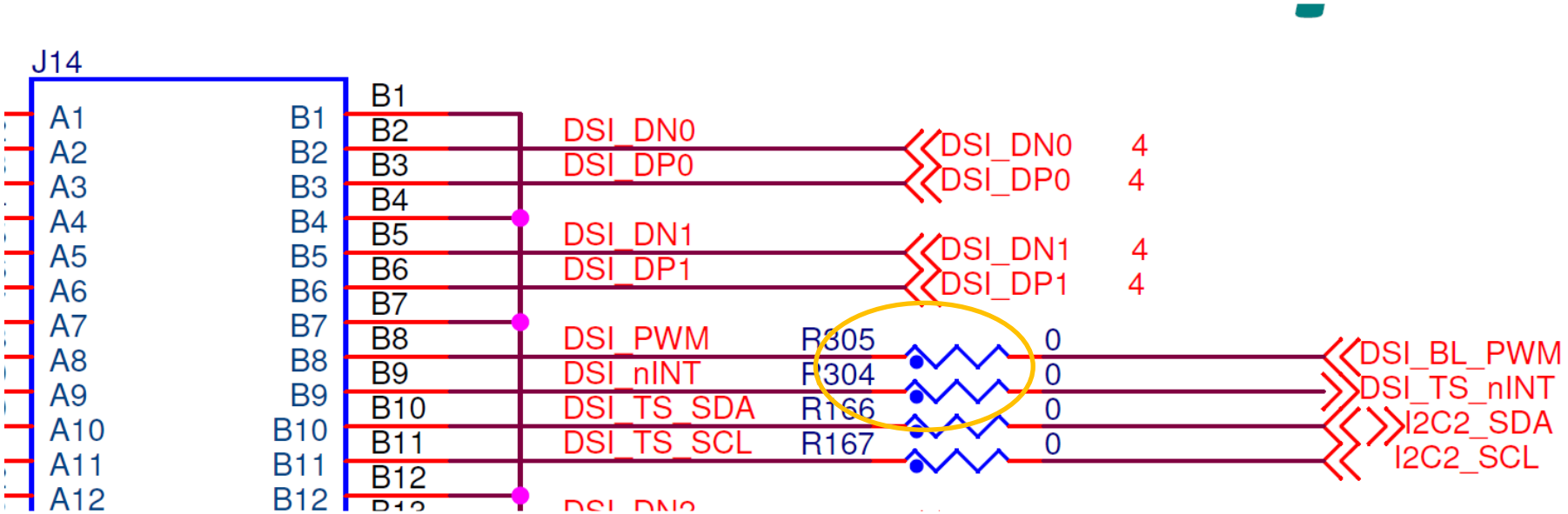
- root@imx8mpevk:~# **echo 0 0 0 1 0 > /sys/class/ptp/ptp1/period**

We can get command line meaning from IO control code: drivers/ptp/ptp_sysfs.c

```
static ssize_t period_store(struct device *dev,
                            struct device_attribute *attr,
                            const char *buf, size_t count)
{
        cnt = sscanf(buf, "%u %lld %u %lld %u", &req.perout.index,
                     &req.perout.start.sec, &req.perout.start.nsec,
                     &req.perout.period.sec, &req.perout.period.nsec);
                ……
        enable = req.perout.period.sec || req.perout.period.nsec;
        err = ops->enable(ops, &req, enable);
        if (err)
```

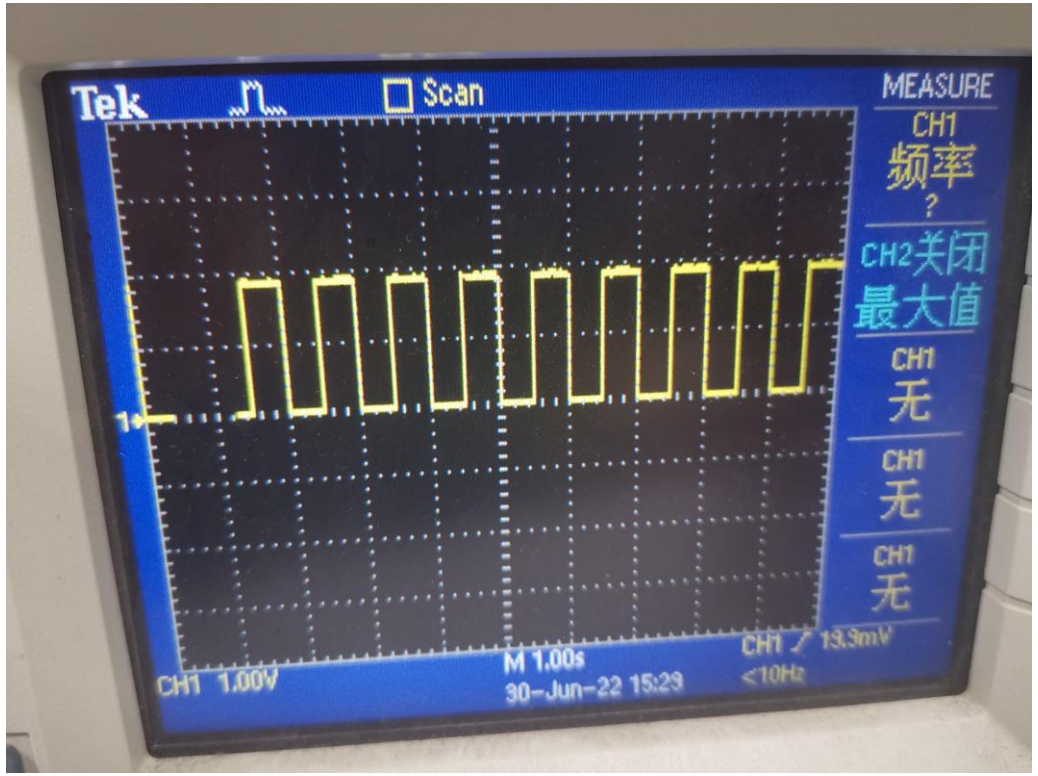
i.MX8MP EVK EQOS 1588 out test point

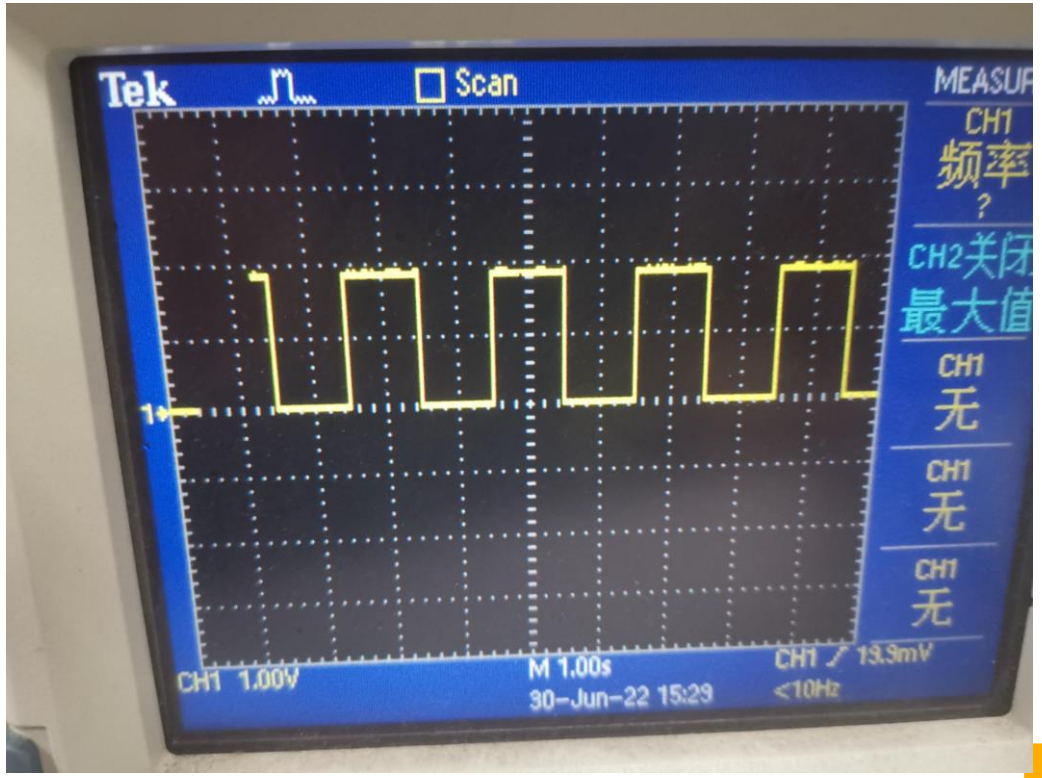
- Test point is baseboard J14, R304



Different PPS waveform result

- echo 0 0 0 1 0 > /sys/class/ptp/ptp1/period // 1PPS signal
- echo 0 0 0 2 0 > /sys/class/ptp/ptp1/period //2 PPS signal



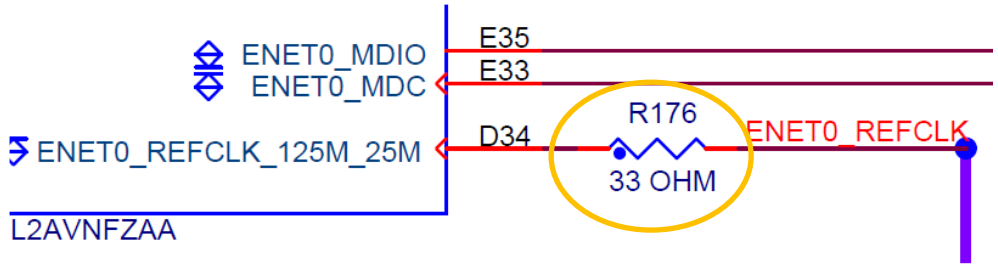


i.MX8DXL EQOS 1588 1PPS test

- Currently besides i.MX8MP, i.MX8DXL also support EQOS.
- Following is i.MX8DXL dts and test command:

CONN_EQOS_PPS_OUT	ENET0_REFCLK_125M_25M	ALT3

- IMX8DXL_ENET0_REFCLK_125M_25M_CONN_EQOS_PPS_OUT



- Please note that i.MX8DXL use channel 1 to test.
- **echo 1 0 0 1 0 > /sys/class/ptp/ptp0/period**

linux/Documentation/ABI/testing/sysfs-ptp

Description:

This write-only file enables or disables periodic outputs. To enable a periodic output, write five integers into the file: **channel index, start time seconds, start time nanoseconds, period seconds, and period nanoseconds.** To disable a periodic output, set all the seconds and nanoseconds values to zero.

Conclusion

- In all i.MX serials, it is similar on FEC port 1588 1PPS test procedure, It is also similar on EQOS port 1588 1PPS test procedure.

SECURE CONNECTIONS
FOR A SMARTER WORLD