Modify LS1046ARDB MAC6 from non-Fixed to Fixed Link

On the LS1046ARDB, there are 2 1G SGMII with PHY, but sometimes customer want to get PHY-less connection to evaluate the performance, so they may have to change the non-fixed link properties into fixed-link by reconfiguration the SW configuration.

In this document, it will give details of configuring the LS1046ARDB to support the fixed-link requirement with LSDK2108 focus on the DTS and Linux kernel. The ethernet MAC in this document is FM1 mEMAC6: 1AE_A000h. Because there is no PHY-less connection on board. We only provide the status when the MAC has been configured.

There are change list in detail below.

DTS Modification

Please find the DTS in the default path: */flexbuild_lsdk2108_github/components/linux/linux/arch/arm64/boot/dts/freescale/fsl-ls1046a-rdb.dts.

In the DTS, please update the desired interface with the fixed-link property:

```
ethernet@ea000 {
        //phy-handle = <&sgmii_phy2>;
        fixed-link = <0 1 1000 0 0>;
        phy-connection-type = "sgmii";
};
```

Linux Modification

The next step will be to apply the patch below on the Linux kernel. See path */flexbuild_lsdk2108_github/components/linux/linux/drivers/net/ethernet/freescale. This patch updates PCS config for SGMII (fixed and non-fixed link) and BASEX configuration

Totally 7 files, this will enable the fixed link mode:

```
---
 .../net/ethernet/freescale/sdk_dpaa/mac-api.c |   2 ++
 drivers/net/ethernet/freescale/sdk_dpaa/mac.c |   4 +++-
 .../sdk_fman/Peripherals/FM/MAC/fm_mac.c           |  1 +
 .../sdk_fman/Peripherals/FM/MAC/fm_mac.h           |  1 +
 .../sdk_fman/Peripherals/FM/MAC/memac.c            | 22 +++++++++++++------
 .../sdk_fman/Peripherals/FM/MAC/memac.h            |  1 +
 .../sdk_fman/inc/Peripherals/fm_mac_ext.h          |  1 +
 7 files changed, 24 insertions(+), 8 deletions(-)

diff --git a/drivers/net/ethernet/freescale/sdk_dpaa/mac-api.c
b/drivers/net/ethernet/freescale/sdk_dpaa/mac-api.c
index d0dda69d3ffe..8cc626b7b27a 100644
--- a/drivers/net/ethernet/freescale/sdk_dpaa/mac-api.c
+++ b/drivers/net/ethernet/freescale/sdk_dpaa/mac-api.c
@@ -84,6 +84,7 @@ static const enet_mode_t _100[] = {
 static const enet_mode_t _1000[] = {
 	[PHY_INTERFACE_MODE_GMII]	= e_ENET_MODE_GMII_1000,
 	[PHY_INTERFACE_MODE_SGMII]	= e_ENET_MODE_SGMII_1000,
+	[PHY_INTERFACE_MODE_1000BASEX]	= e_ENET_MODE_SGMII_BASEX_1000,
 	[PHY_INTERFACE_MODE_QSGMII]	= e_ENET_MODE_QSGMII_1000,
 	[PHY_INTERFACE_MODE_TBI]	= e_ENET_MODE_TBI_1000,
 	[PHY_INTERFACE_MODE_RGMII]	= e_ENET_MODE_RGMII_1000,
@@ -140,6 +141,7 @@ static int __cold init(struct mac_device *mac_dev)
 	param.baseAddr =  (typeof(param.baseAddr))(uintptr_t)devm_ioremap(
 		mac_dev->dev, mac_dev->res->start, 0x2000);
 	param.enetMode	= macdev2enetinterface(mac_dev);
+	param.autoneg = (mac_dev->fixed_link) ? false : true;
 	memcpy(&param.addr, mac_dev->addr, min(sizeof(param.addr),
 		sizeof(mac_dev->addr)));
 	param.macId		= mac_dev->cell_index;
diff --git a/drivers/net/ethernet/freescale/sdk_dpaa/mac.c
b/drivers/net/ethernet/freescale/sdk_dpaa/mac.c
index 3092c49e9e1c..8af080332050 100644
--- a/drivers/net/ethernet/freescale/sdk_dpaa/mac.c
+++ b/drivers/net/ethernet/freescale/sdk_dpaa/mac.c
```

```
@@ -67,6 +67,7 @@ static const char phy_str[][11] = {
        [PHY_INTERFACE_MODE_MII]        = "mii",
        [PHY_INTERFACE_MODE_GMII]       = "gmii",
        [PHY_INTERFACE_MODE_SGMII]      = "sgmii",
+       [PHY_INTERFACE_MODE_1000BASEX] = "1000base-x",
        [PHY_INTERFACE_MODE_QSGMII]     = "qsgmii",
        [PHY_INTERFACE_MODE_TBI]        = "tbi",
        [PHY_INTERFACE_MODE_RMII]       = "rmii",
@@ -94,6 +95,7 @@ static const uint16_t phy2speed[] = {
        [PHY_INTERFACE_MODE_MII]        = SPEED_100,
        [PHY_INTERFACE_MODE_GMII]       = SPEED_1000,
        [PHY_INTERFACE_MODE_SGMII]      = SPEED_1000,
+       [PHY_INTERFACE_MODE_1000BASEX]  = SPEED_1000,
        [PHY_INTERFACE_MODE_QSGMII]     = SPEED_1000,
        [PHY_INTERFACE_MODE_TBI]        = SPEED_1000,
        [PHY_INTERFACE_MODE_RMII]       = SPEED_100,
@@ -342,7 +344,7 @@ static int __cold mac_probe(struct platform_device *_of_dev)
        mac_dev->if_support = DTSEC_SUPPORTED;
        /* We don't support half-duplex in SGMII mode */
        if (strstr(char_prop, "sgmii") || strstr(char_prop, "qsgmii") ||
-           strstr(char_prop, "sgmii-2500"))
+           strstr(char_prop, "sgmii-2500") || strstr(char_prop, "1000base-x"))
                mac_dev->if_support &= ~(SUPPORTED_10baseT_Half |
                                         SUPPORTED_100baseT_Half);
```

```
diff --git
a/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/fm_mac.c
b/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/fm_mac.c
index caf3940ad9df..e74a10183b93 100644
--- a/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/fm_mac.c
+++ b/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/fm_mac.c
@@ -80,6 +80,7 @@ t_Handle FM_MAC_Config (t_FmMacParams *p_FmMacParam)

        p_FmMacControllerDriver->h_Fm                  = p_FmMacParam->h_Fm;
        p_FmMacControllerDriver->enetMode              = p_FmMacParam->enetMode;
+       p_FmMacControllerDriver->autoneg               = p_FmMacParam->autoneg;
        p_FmMacControllerDriver->macId                 = p_FmMacParam->macId;
        p_FmMacControllerDriver->resetOnInit           = DEFAULT_resetOnInit;

diff --git
a/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/fm_mac.h
b/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/fm_mac.h
index ba3b9133a88f..804ec3e4b8e6 100644
--- a/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/fm_mac.h
+++ b/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/fm_mac.h
@@ -135,6 +135,7 @@ typedef struct {

        t_Handle                h_Fm;
        t_FmRevisionInfo        fmRevInfo;
+       bool                    autoneg;
        e_EnetMode              enetMode;
        uint8_t                 macId;
        bool                    resetOnInit;
diff --git
a/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/memac.c
b/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/memac.c
index 0d86f5b05a35..651ac8f7d43a 100644
--- a/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/memac.c
+++ b/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/memac.c
@@ -84,17 +84,23 @@ static void SetupSgmiiInternalPhy(t_Memac *p_Memac,
uint8_t phyAddr)
 {
        uint16_t        tmpReg16;
        e_EnetMode      enetMode;
-       bool autoneg_disabled = p_Memac->enetMode == e_ENET_MODE_SGMII_2500;
+       bool autoneg_disabled; //
+

            /* In case the higher MACs are used (i.e. the MACs that should support
10G),
             speed=10000 is provided for SGMII ports. Temporary modify enet mode
             to 1G one, so MII functions can work correctly. */
        enetMode = p_Memac->enetMode;

-       /* SGMII mode + AN enable */
-       tmpReg16 = PHY_SGMII_IF_MODE_AN | PHY_SGMII_IF_MODE_SGMII;
+       if (p_Memac->autoneg) {
+
        /* SGMII mode + AN enable */
+            tmpReg16 = PHY_SGMII_IF_MODE_AN | PHY_SGMII_IF_MODE_SGMII;
+       } else {
+            tmpReg16 = PHY_SGMII_IF_MODE_SGMII |
PHY_SGMII_IF_SPEED_GIGABIT;
+       }
+
        /* unless SGMII 2500 where AN needs to be disabled    */
-       if (autoneg_disabled)
+       if (p_Memac->enetMode == e_ENET_MODE_SGMII_2500)
             tmpReg16 = PHY_SGMII_IF_SPEED_GIGABIT | PHY_SGMII_IF_MODE_SGMII;

        p_Memac->enetMode =
MAKE_ENET_MODE(ENET_INTERFACE_FROM_MODE(p_Memac->enetMode),
e_ENET_SPEED_1000);
@@ -118,7 +124,7 @@ static void SetupSgmiiInternalPhy(t_Memac *p_Memac,
uint8_t phyAddr)
        MEMAC_MII_WritePhyReg(p_Memac, phyAddr, 0x13, 0x0007);
        MEMAC_MII_WritePhyReg(p_Memac, phyAddr, 0x12, 0xa120);

-       if (!autoneg_disabled)
+       if (p_Memac->autoneg && (p_Memac->enetMode != e_ENET_MODE_SGMII_2500))
             /* Restart AN */
             tmpReg16 = PHY_SGMII_CR_DEF_VAL | PHY_SGMII_CR_RESET_AN;
        else
@@ -165,8 +171,10 @@ static void SetupSgmiiInternalPhyBaseX(t_Memac
*p_Memac, uint8_t phyAddr)
        MEMAC_MII_WritePhyReg(p_Memac, phyAddr, 0x13, 0x002f);
        MEMAC_MII_WritePhyReg(p_Memac, phyAddr, 0x12, 0xaf08);

-       /* Restart AN */
-       tmpReg16 = PHY_SGMII_CR_DEF_VAL | PHY_SGMII_CR_RESET_AN;
+       if (p_Memac->autoneg)
+            /* Restart AN */
+            tmpReg16 = PHY_SGMII_CR_DEF_VAL | PHY_SGMII_CR_RESET_AN;
+       else
+           tmpReg16 = PHY_SGMII_CR_DEF_VAL & ~PHY_SGMII_CR_AN_EN;
+
        MEMAC_MII_WritePhyReg(p_Memac, phyAddr, 0x0, tmpReg16);

        /* Restore original enet mode */
diff --git
a/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/memac.h
b/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/memac.h
index d361af42b0c2..4fd0298152f3 100644
--- a/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/memac.h
+++ b/drivers/net/ethernet/freescale/sdk_fman/Peripherals/FM/MAC/memac.h
@@ -70,6 +70,7 @@ typedef struct
        struct memac_regs               *p_MemMap;                        /**<
Pointer to MAC memory mapped registers */
        struct memac_mii_access_mem_map *p_MiiMemMap;
/**< Pointer to MII memory mapped registers */
        uint64_t                         addr;                           /**<
MAC address of device */
+       bool                             autoneg;
        e_EnetMode                        enetMode;                       /**<
Ethernet physical interface    */
        t_FmMacExceptionCallback        *f_Exception;
        int                              mdioIrq;
diff --git
a/drivers/net/ethernet/freescale/sdk_fman/inc/Peripherals/fm_mac_ext.h
b/drivers/net/ethernet/freescale/sdk_fman/inc/Peripherals/fm_mac_ext.h
index be99b7c9f93b..182854325dd2 100644
--- a/drivers/net/ethernet/freescale/sdk_fman/inc/Peripherals/fm_mac_ext.h
+++ b/drivers/net/ethernet/freescale/sdk_fman/inc/Peripherals/fm_mac_ext.h
@@ -230,6 +230,7 @@ typedef struct t_FmMacParams {
                                                                   0 -
FM_MAX_NUM_OF_1G_MACS;
                                                               numbering of
10G-MAC (TGEC) and 10G-mEMAC:
                                                                   0 -
FM_MAX_NUM_OF_10G_MACS */
+       bool                             autoneg;
        e_EnetMode                        enetMode;             /**< Ethernet operation
mode (MAC-PHY interface and speed);
                                                               Note that the
speed should indicate the maximum rate that
                                                               this MAC should
support rather than the actual speed;
```

Check Ethernet Status

After the board has booted make sure that the ethernet node is still in fixed mode. To verify this, please dump the running DTB:

```
dtc -I fs -O dts -o dts /proc/device-tree
```

#check inside the generated DTS for the fixed node

```
cat dts | less
```

In the log, you will get fixed-link in the ethernet@ea000

```
 ethernet@ea000 {
         phy-connection-type = "sgmii";
         ptp-timer = <0x17>;
         local-mac-address = [00 04 9f 07 75 24];
         cell-index = <0x05>;
         pcsphy-handle = <0x2c>;
         fixed-link = <0x00 0x01 0x3e8 0x00 0x00>;
         compatible = "fsl,fman-memac";
         reg = <0xea000 0x1000>;
         phandle = <0x39>;
         fsl,fman-ports = <0x2a 0x2b>;
};
```

Now the MAC6 has been set as fixed-link, it can be appeared with command "ifconfig -a".