DM9000A Migrated to DM9000B

Application Note
Hardware

**Scenario 1:** If the design is based on DM9000A application schematic provided by Davicom, there is no change to DM9000B.

**Scenario 2:** If the design is using external 2.5V to Transformer’s Center Tap (CT) instead of 2.5V provided by DM9000A, this external 2.5V must be changed to 1.8V while DM9000B is applied.

Software

**Scenario 1:** Davicom’s Driver is primitively applied on the design. If the driver version applied to DM9000A is older than the following series of newer version numbers, a newer Davicom’s Driver must be required.

- Linux platform: 2.50
- WinCE platform: 2.50
Scenario 2: If Davicom’s Driver is not primitively applied on the design with DM9000A, there will be some differences to DM9000B in your own driver as following:

1. Register 38H is different.
2. When Register IFH: Bit 0 changes status from 1 to 0, a delay is required at least 1 ms.
3. PHY Register 1BH is different.
4. Processor I/O Read Timing is different.

Instructions of driver modification:

- **Linux Driver**

  1. **Set Reg 38H to 0x01 (eliminate about 2ns IOR spike)**
     
     ```
     #define DM9KS_BUSCR 0x38
     #define DM9KS_CHIPR 0x2c
     #define DM9000B_REV 0x1A
     db->chip_revision = ior(db, DM9KS_CHIPR);
     if (db->chip_revision == DM9000B_REV)  // if chip is DM9000B
        iow(db, DM9KS_BUSCR, 0x01); /* default: 2mA */
     ```

  2. **After power up PHY/power down PHY, make 20ms delay.**
     
     ```
     #define DM9KS_GPR 0x1f
     iow(db, DM9KS_GPR, 0); /* GPR (reg_1F h)bit GPIO0=0 pre-activate PHY */
     mdelay(20);  /* wait for PHY power-on ready */
     ```

  3. **Set PHY Reg1bH to 0xE100**
     
     ```
     #define PHY_DSP_Control 0x1b
     if (db->chip_revision == DM9000B_REV)  // if chip is DM9000B
        phy_write(db, PHY_DSP_Control, 0xE100);
     ```

- **Linux Kernel**

  1. Processor I/O Read Timing
     
     Max. of T3 is 19ns, and Max. of T4 is 19ns.
     
     *19ns for bus driving 2mA, 12ns for 4mA, 10ns for 6mA, 10ns for 8mA

- **Uboot**

  No modification
WinCE Driver

1. Set Reg 38H to 0x01 (eliminate about 2ns IOR spike)
   
   ```
   #define DM9_PBCR 0x38
   #define DM9_CHIPREV 0x2c
   #define DM9000B_REV 0x1A
   
   If(DeviceReadPort(DM9_CHIPREV) == DM9000B_REV)
   DeviceWritePort(DM9_PBCR,0x01);
   ```

2. After power up PHY/ power down PHY, make 20ms delay.

   ```
   #define DM9_GPR 0x1f
   DeviceWritePort(DM9_GPR,0);
   NdisMSleep(20000);  //20ms
   ```

3. Set PHY Reg1bH to 0xE1

   ```
   #define PHY_DSP_Control 0x1b
   
   If(DeviceReadPort(DM9_CHIPREV) == DM9000B_REV)
   DeviceWritePhy(0,PHY_DSP_Control,0xE100);
   ```

EBoot

No modification