MAC Address usage

This TE modification was done for TE0720 only.

Zynq FSBL

As standard option Zynq FSBL for TE0720 does read and display on primary UART the MAC address that is stored on the module, this MAC Address is written to registers of the Primary Ethernet IP Core. If u-boot (or another second stage bootloader) does not overwrite it, then it will persist and be used in Linux (or other O/S) as well.

This is done in fsbl_hooks.c in the FsblHookBeforeHandoff function. When using TE supplied example design for Petalinux the FSBL code is already including this. When creating FSBL from scratch using Xilinx SDK Wizard then the hook code must be inserted manually (or the fsbl_hooks.c replaced completely). The way MAC Address is read from Hardware is module dependent, so proper files should be used.

MAC Address in u-boot

U-boot Ethernet driver guide says that:

```
        /* if your device has dedicated hardware storage for the
         * MAC, read it and initialize dev->enetaddr with it
         */
```

Unfortunately Xilinx u-boot Ethernet drivers (xilinx_axi_emac, xilinx_axilite, zynq_gem) do not implement this functionality as of Petalinux 2014.4 version.

In u-boot documentataion README.ethernet it says:

```
Correct flow of setting up the MAC address (summarized):

1. Read from hardware in initialize() function
2. Read from environment in net/eth.c after initialize()
3. The environment variable will be compared to the driver initialized
   struct eth_device->enetaddr. If they differ, a warning is printed, and the
   environment variable will be used unchanged.
   If the environment variable is not set, it will be initialized from
   eth_device->enetaddr, and a warning will be printed.
4. Program the address into hardware if the following conditions are met:
        a) The relevant driver has a 'write_addr' function
        b) The user hasn't set an 'ethmacskip' environment variable
        c) The address is valid (unicast, not all-zeros)
```

Xilinx u-boot and u-boot drivers for Zynq and MicroBlaze do not support any kind of reading of the MAC, they do not support the setting of MAC in Ethernet driver.

Xilinx document UG983 explains how to set MAC address from u-boot prompt.

To disable u-boot to overwrite the MAC Address that was read and set in FSBL environment variable ethmacskip needs to be set. This prevents u-boot to write the MAC from environment into Ethernet Hardware.

As only option to use proper MAC Address (read from hardware MAC Address EEPROM in FSBL or other first stage bootloader) shared memory can be used. This method is supported by TE supplied FSBL hooks code. This MAC Address info is loaded into u-boot using a script - there is no need to patch u-boot source code.

/* PREBOOT */
#undef CONFIG_PREBOOT
#define CONFIG_PREBOOT<>"echo U-BOOT for ${hostname} preboot is loading environment from uEnv.txt; set autoexec echo; if mmc rescan; then if fatload mmc 0 0x01000000 uEnv.txt; then env import -t 0x01000000; fi; fi; "\

"echo importing env from FSBL shared area at 0xFFFFFC00; if itest *0xFFFFFC00 == 0xCAFEBABE; then env import -t 0xFFFFFC04; fi; echo running autoexec script; run autoexec"

Example use of the MAC Address import in u-boot

> ⚠ FSBL hooks.c are available to provide support for this MAC address forwarding to u-boot