Starterware 在 BeagleBone Black 上的移植

Starterware 是针对 AM335x 的 no-OS 软件平台，既可以用于 AM335x 的裸跑应用，又可以做一些底层驱动调试。目前只支持 GPEVM，SK，BeagleBone 三个平台，不支持 BeagleBone Black。BeagleBone Black 是 BeagleBone 的升级版，AM335x 从 PG1.0 的 720M 升级到 PG2.1 的 1G，DDR 由 256M DDR2 升为 512M DDR3L，PMU 由 TPS65217B 改为 TPS65217C，去掉了 FTDI JTAG-USB 芯片和 USB Hub 芯片，增加了 HDMI 转换芯片。目前越来越多客户基于 BaggelBone Black 作参考设计，因此下面介绍下将 starterware 移植到 BaggelBone Black 上的几个需要修改的地方。

目前最新的 Starterware 版本是 02.00.01.01，用宏定义"beaglebone"来包含 BeagleBone 专用的板级配置。针对 BaggelBone Black 的移植，最简单快捷的办法就是把"beaglebone"宏定义中的板级配置内容，改为 beagleboneblack 的配置，主要是修改 bootloader 部分内容。如下列出的需要修改的地方用粗体标出。对于 Gel 文件，可参考 bootloader 做对应修改。

1. 修改\bootloader\src\armv7a\am335x\bl_platform.c 中的 ConfigVddOpVoltage()函数，这个函数是启动后调节 PMU 电压的，TPS65217C 的 LDO3 和 LDO4 电压已经确定了，无需再调节。

```
void ConfigVddOpVoltage(void)
{
   SetupI2C();

#ifdef beaglebone

   unsigned char pmic_status = 0;

   /* Configure PMIC slave address */
   I2CMasterSlaveAddrSet(SOC_I2C_0_REGS, PMIC_TPS65217_I2C_SLAVE_ADDR);

   TPS65217RegRead(STATUS, &pmic_status);

   /* Increase USB current limit to 1300mA */
   TPS65217RegWrite(PROT_LEVEL_NONE, POWER_PATH, USB_INPUT_CUR_LIMIT_1300MA,
                USB_INPUT_CUR_LIMIT_MASK);

   /* Set DCDC2 (MPU) voltage to 1.275V */
   TPS65217VoltageUpdate(DEFDCDC2, DCDC_VOLT_SEL_1100MV);

   /* Set LDO3, LDO4 output voltage to 3.3V */
 //  TPS65217RegWrite(PROT_LEVEL_2, DEFLS1, LDO_VOLTAGE_OUT_3_3,
LDO_MASK);//for tps65217C


 //   TPS65217RegWrite(PROT_LEVEL_2, DEFLS2, LDO_VOLTAGE_OUT_3_3, LDO_MASK);//
for tps65217C

#elif  defined (evmAM335x) || defined (evmskAM335x)
```

```
    /* Configure PMIC slave address */
    I2CMasterSlaveAddrSet(SOC_I2C_0_REGS, PMIC_CNTL_I2C_SLAVE_ADDR);

            /* Select SR I2C(0) */
    SelectI2CInstance(PMIC_DEVCTRL_REG_SR_CTL_I2C_SEL_CTL_I2C);

    /* Configure vdd1- need to validate these parameters */
    ConfigureVdd1(PMIC_VDD1_REG_VGAIN_SEL_X1, PMIC_VDD1_REG_ILMAX_1_5_A,
                    PMIC_VDD1_REG_TSTEP_12_5, PMIC_VDD1_REG_ST_ON_HI_POW);

    /* Select the source for VDD1 control */
    SelectVdd1Source(PMIC_VDD1_OP_REG_CMD_OP);

#else

    #error Unsupported EVM !!

#endif
}
```

2. 修改\bootloader\src\armv7a\am335x\bl_platform.c 中的 DDR3 参数配置，Beagleboneblack 是镁光 512MB 的 DDR3L，型号：MT41K256M16HA

```
/* DDR3 init values */
#ifdef evmskAM335x
#define DDR3_CMD0_SLAVE_RATIO_0          (0x40)
#define DDR3_CMD0_INVERT_CLKOUT_0        (0x1)
#define DDR3_CMD1_SLAVE_RATIO_0          (0x40)
#define DDR3_CMD1_INVERT_CLKOUT_0        (0x1)
#define DDR3_CMD2_SLAVE_RATIO_0          (0x40)
#define DDR3_CMD2_INVERT_CLKOUT_0        (0x1)

#define DDR3_DATA0_RD_DQS_SLAVE_RATIO_0    (0x3B)
#define DDR3_DATA0_WR_DQS_SLAVE_RATIO_0    (0x85)
#define DDR3_DATA0_FIFO_WE_SLAVE_RATIO_0   (0x100)
#define DDR3_DATA0_WR_DATA_SLAVE_RATIO_0   (0xC1)

#define DDR3_DATA0_RD_DQS_SLAVE_RATIO_1    (0x3B)
#define DDR3_DATA0_WR_DQS_SLAVE_RATIO_1    (0x85)
#define DDR3_DATA0_FIFO_WE_SLAVE_RATIO_1   (0x100)
#define DDR3_DATA0_WR_DATA_SLAVE_RATIO_1   (0xC1)

#define DDR3_CONTROL_DDR_CMD_IOCTRL_0      (0x18B)
#define DDR3_CONTROL_DDR_CMD_IOCTRL_1      (0x18B)
#define DDR3_CONTROL_DDR_CMD_IOCTRL_2      (0x18B)

#define DDR3_CONTROL_DDR_DATA_IOCTRL_0     (0x18B)
#define DDR3_CONTROL_DDR_DATA_IOCTRL_1     (0x18B)

//#define DDR3_CONTROL_DDR_IO_CTRL          (0x0fffffff)
#define DDR3_CONTROL_DDR_IO_CTRL          (0xefffffff)

#define DDR3_EMIF_DDR_PHY_CTRL_1          (0x06)
#define DDR3_EMIF_DDR_PHY_CTRL_1_DY_PWRDN        (0x00100000)
```

```
#define DDR3_EMIF_DDR_PHY_CTRL_1_SHDW      (0x06)
#define DDR3_EMIF_DDR_PHY_CTRL_1_SHDW_DY_PWRDN    (0x00100000)
#define DDR3_EMIF_DDR_PHY_CTRL_2          (0x06)

#define DDR3_EMIF_SDRAM_TIM_1             (0x0888A39B)
#define DDR3_EMIF_SDRAM_TIM_1_SHDW        (0x0888A39B)

#define DDR3_EMIF_SDRAM_TIM_2             (0x26337FDA)
#define DDR3_EMIF_SDRAM_TIM_2_SHDW        (0x26337FDA)

#define DDR3_EMIF_SDRAM_TIM_3             (0x501F830F)
#define DDR3_EMIF_SDRAM_TIM_3_SHDM        (0x501F830F)

#define DDR3_EMIF_SDRAM_REF_CTRL_VAL1     (0x0000093B)
#define DDR3_EMIF_SDRAM_REF_CTRL_SHDW_VAL1 (0x0000093B)

#define DDR3_EMIF_ZQ_CONFIG_VAL           (0x50074BE4)
#define DDR3_EMIF_SDRAM_CONFIG            (0x61C04AB2)//termination = 1 (RZQ/4)
                                          //dynamic ODT = 2 (RZQ/2)
                                          //SDRAM drive = 0 (RZQ/6)
                                          //CWL = 0 (CAS write latency = 5)
                                          //CL = 2 (CAS latency = 5)
                                          //ROWSIZE = 5 (14 row bits)
                                    //PAGESIZE = 2 (10 column bits)
#else
#define DDR3_CMD0_SLAVE_RATIO_0           (0x80)
#define DDR3_CMD0_INVERT_CLKOUT_0         (0x0)
#define DDR3_CMD1_SLAVE_RATIO_0           (0x80)
#define DDR3_CMD1_INVERT_CLKOUT_0         (0x0)
#define DDR3_CMD2_SLAVE_RATIO_0           (0x80)
#define DDR3_CMD2_INVERT_CLKOUT_0         (0x0)

#define DDR3_DATA0_RD_DQS_SLAVE_RATIO_0    (0x38)//(0x3B)
#define DDR3_DATA0_WR_DQS_SLAVE_RATIO_0    (0x44)//(0x3C)
#define DDR3_DATA0_FIFO_WE_SLAVE_RATIO_0   (0x94)//(0xA5)
#define DDR3_DATA0_WR_DATA_SLAVE_RATIO_0   (0x7D)//(0x74)

#define DDR3_DATA0_RD_DQS_SLAVE_RATIO_1    (0x38)// (0x3B)
#define DDR3_DATA0_WR_DQS_SLAVE_RATIO_1    (0x44)// (0x3C)
#define DDR3_DATA0_FIFO_WE_SLAVE_RATIO_1   (0x94)// (0xA5)
#define DDR3_DATA0_WR_DATA_SLAVE_RATIO_1   (0x7D)// (0x74)

#define DDR3_CONTROL_DDR_CMD_IOCTRL_0      (0x18B)
#define DDR3_CONTROL_DDR_CMD_IOCTRL_1      (0x18B)
#define DDR3_CONTROL_DDR_CMD_IOCTRL_2      (0x18B)

#define DDR3_CONTROL_DDR_DATA_IOCTRL_0     (0x18B)
#define DDR3_CONTROL_DDR_DATA_IOCTRL_1     (0x18B)

#define DDR3_CONTROL_DDR_IO_CTRL          (0xefffffff)

#define DDR3_EMIF_DDR_PHY_CTRL_1          (0x07)// (0x06)
#define DDR3_EMIF_DDR_PHY_CTRL_1_DY_PWRDN        (0x00100000)
#define DDR3_EMIF_DDR_PHY_CTRL_1_SHDW     (0x07)// (0x06)
```

```
#define DDR3_EMIF_DDR_PHY_CTRL_1_SHDW_DY_PWRDN    (0x00100000)
#define DDR3_EMIF_DDR_PHY_CTRL_2          (0x07)// (0x06)

#define DDR3_EMIF_SDRAM_TIM_1             (0x0AAAD4DB)// (0x0888A39B)
#define DDR3_EMIF_SDRAM_TIM_1_SHDW        (0x0AAAD4DB)// (0x0888A39B)

#define DDR3_EMIF_SDRAM_TIM_2             (0x266B7FDA)// (0x26517FDA)
#define DDR3_EMIF_SDRAM_TIM_2_SHDW        (0x266B7FDA)// (0x26517FDA)

#define DDR3_EMIF_SDRAM_TIM_3             (0x501F867F)// (0x501F84EF)
#define DDR3_EMIF_SDRAM_TIM_3_SHDM        (0x501F867F)// (0x501F84EF)

#define DDR3_EMIF_SDRAM_REF_CTRL_VAL1     (0x00000C30)// (0x0000093B)
#define DDR3_EMIF_SDRAM_REF_CTRL_SHDW_VAL1 (0x00000C30)// (0x0000093B)

#define DDR3_EMIF_ZQ_CONFIG_VAL           (0x50074BE4)

/*
** termination = 1 (RZQ/4)
** dynamic ODT = 2 (RZQ/2)
** SDRAM drive = 0 (RZQ/6)
** CWL = 0 (CAS write latency = 5)
** CL = 2 (CAS latency = 5)
** ROWSIZE = 7 (16 row bits)
** PAGESIZE = 2 (10 column bits)
*/
#define DDR3_EMIF_SDRAM_CONFIG            (0x61C05332)// (0x61C04BB2)
#endif
```

3. 修改\bootloader\src\armv7a\am335x\bl_platform.c 中的 BlPlatformConfig()，将 BeagleboneBlack 的 DDR 配置改为 DDR3：

```
#ifdef evmskAM335x
  freqMultDDR = DDRPLL_M_DDR3;
#elif evmAM335x
  if(BOARD_ID_EVM_DDR3 == BoardIdGet())
  {
    freqMultDDR = DDRPLL_M_DDR3;
  }
  else if(BOARD_ID_EVM_DDR2 == BoardIdGet())
  {
    freqMultDDR = DDRPLL_M_DDR2;
  }
#else
  freqMultDDR = DDRPLL_M_DDR3;//for Beaglebone Black DDR3
#endif
```

4. 修改\bootloader\src\armv7a\am335x\bl_platform.h 中的 DDRPLL_M_DDR3，将其由 PG1.0 芯片最高 303M 改为 PG2.1 芯片的最高 400M

```
#define DDRPLL_M_DDR2             (266)
#define DDRPLL_M_DDR3              (400)// for Beaglebone Black DDR3
#define DDRPLL_N                   23
#define DDRPLL_M2                  1
```