User Manual For Embedded Study

Porting U-boot-2012.10 & Linux 3.9-rc6 To OK6410-A.

East China Institute Of Technology

E-mail:houstar@foxmail.com

Houstar

Part 1.U-boot

Repository.
You can download here.
https://github.com/houstar/u-boot-2012.10

Rule

$ this color means command execute in shell(or terminal)

this color means OK6410's bootload information

-: means delete current line
+: means add code in current line

Prepare 1

1,Crosstool: arm-unknown-linux-uclibceabi-gcc
2,OS: Archlinux
3,Board:OK6410-A,4G Nand flash updated.
4,SD:4G Kingstone
5,Serial line & USB to Serial line
If you want to download my tools,please contact with me.E-mail:houstar@foxmail.com
6.You must install git which known as Version Control System that you can find where I modified and so on.
7.Also you must install lib named lusb.
8.My OS: Archlinux
9,My Cross tool:arm-unknown-linux-uclibceabi-gcc

```
$ arm-unknown-linux-uclibceabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-unknown-linux-uclibceabi-gcc
COLLECT_LTO_WRAPPER=/usr/local/clfs/cross-tools/libexec/gcc/arm-unknown-linux-uclibceabi/4.7.0/lto-wrapper
```

```
Target: arm-unknown-linux-uclibceabi
Configured with: ../gcc-4.7.0/configure --prefix=/usr/local/clfs/cross-tools
--build=x86_64-cross-linux-gnu --target=arm-unknown-linux-uclibceabi
--host=x86_64-cross-linux-gnu --with-sysroot=/usr/local/clfs --disable-nls --enable-
shared --enable-languages=c,c++ --enable-c99 --enable-long-long --with-
mpfr=/usr/local/clfs/cross-tools --with-gmp=/usr/local/clfs/cross-tools --with-
mpc=/usr/local/clfs/cross-tools --disable-multilib --with-abi=aapcs --with-
cpu=arm1176jzf-s --with-mode=arm --with-float=hard --with-fpu=vfp
Thread model: posix
gcc version 4.7.0 (GCC)
```

Prepare 2

Assume that you are already know how to use Linux,such as these commands:ls,tar,rm,mkdir,vim,gcc,make.Now let's us enjoy the road to embedded world.As far as we known,If we have a skillfull PM ,we will contribute ourself to the society.

```
$make ARCH=arm CROSS_COMPILE=arm-unknown-linux-uclibceabi-
smdk6400_config
```

If you can find u-boot.bin & u-boot-nand.bin,it proves that you're already installed cross tool successfully!

Chapter 1,Start

Step 1.Copy directory.

Aussume that you have been downloaded the u-boot into root directory.Now you should change directory to the u-boot.

```
$git init
Initialized empty Git repository in /home/houstar/GitRepo/u-boot/.git/
$git add .
$git commit  -m 'origin edition'
```

Now we new directory,you can find difference as followed which use git status command:

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   arch/arm/include/asm/arch-s3c64xx/s3c6410.h
#       new file:   board/samsung/smdk6410/.gitignore
#       new file:   board/samsung/smdk6410/Makefile
#       new file:   board/samsung/smdk6410/config.mk
#       new file:   board/samsung/smdk6410/lowlevel_init.S
#       new file:   board/samsung/smdk6410/smdk6400.c
```

```
#       new file:   board/samsung/smdk6410/smdk6400_nand_spl.c
#       new file:   board/samsung/smdk6410/u-boot-nand.lds
#       new file:   include/configs/smdk6410.h
#       new file:   nand_spl/board/samsung/smdk6410/Makefile
#       new file:   nand_spl/board/samsung/smdk6410/config.mk
#       new file:   nand_spl/board/samsung/smdk6410/u-boot.lds
#
```

Step 2.Rename those files,We use git mv in /board/samsung/smdk6410/

```
$ git mv smdk6400.c smdk6410.c
```

```
$ git mv smdk6400_nand_spl.c smdk6410_nand_spl.c
```

Check the status.

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   ../../../arch/arm/include/asm/arch-s3c64xx/s3c6410.h
#       new file:   .gitignore
#       new file:   Makefile
#       new file:   config.mk
#       new file:   lowlevel_init.S
#       new file:   smdk6410.c
#       new file:   smdk6410_nand_spl.c
#       new file:   u-boot-nand.lds
#       new file:   ../../../include/configs/smdk6410.h
#       new file:   ../../../nand_spl/board/samsung/smdk6410/Makefile
#       new file:   ../../../nand_spl/board/samsung/smdk6410/config.mk
#       new file:   ../../../nand_spl/board/samsung/smdk6410/u-boot.lds
#
```

Step 3,In these directory.

1,/board/samsung/smdk6410,smdk6410.c & lowlevel_init.S

/arch/arm/cpu/arm1176/s3c64xx, cpu_init.S,reset.S,speed.S&timer.S
/drivers/mtd/nand, s3c64xx.c
/drivers/serial,s3c64xx.c
/drivers/usb/host,s3c64xx-hcd.c

```
modify #include <asm/arch/s3c6400.h> as #include<asm/arch/s3c6410.h>
```

2,/nand_spl/board/samsung/smdk6410/ Makefile. Through all file replace smdk6400 as smdk6410
3,/include/configs/smdk6410.h

#define CONFIG_S3C6400 -->#define CONFIG_S3C6410
#define CONFIG_SMDK6400 -->#define CONFIG_SMDK6410
4,/arch/arm/include/asm/s3c64xx/s3c6410.h
-#ifndef S3C6400_H
-#define S3C6400_H
+#ifndef S3C6410_H

5,Makefile.
Add these under smdk6400_noUSB_config:

```
smdk6410_noUSB_config      \
smdk6410_config         :       unconfig
        @mkdir -p $(obj)include $(obj)board/samsung/smdk6410
        @mkdir -p $(obj)nand_spl/board/samsung/smdk6410
        @echo "#define CONFIG_NAND_U_BOOT" > $(obj)include/config.h
        @echo "CONFIG_NAND_U_BOOT = y" >> $(obj)include/config.mk
        @if [ -z "$(findstring smdk6410_noUSB_config,$@)" ]; then
        \
                echo "RAM_TEXT = 0x57e00000" >> $
(obj)board/samsung/smdk6410/config.tmp;\
        else                                                    \
                echo "RAM_TEXT = 0xc7e00000" >> $
(obj)board/samsung/smdk6410/config.tmp;\
        fi
        @$(MKCONFIG) smdk6410 arm arm1176 smdk6410 samsung s3c64xx
        @echo "CONFIG_NAND_U_BOOT = y" >> $(obj)include/config.mk
```

Now we check the status.Verify.....Yeah , Hope we can successed.

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   arch/arm/include/asm/arch-s3c64xx/s3c6410.h
#       new file:   board/samsung/smdk6410/.gitignore
#       new file:   board/samsung/smdk6410/Makefile
#       new file:   board/samsung/smdk6410/config.mk
#       new file:   board/samsung/smdk6410/lowlevel_init.S
#       new file:   board/samsung/smdk6410/smdk6410.c
#       new file:   board/samsung/smdk6410/smdk6410_nand_spl.c
#       new file:   board/samsung/smdk6410/u-boot-nand.lds
#       new file:   include/configs/smdk6410.h
#       new file:   nand_spl/board/samsung/smdk6410/Makefile
#       new file:   nand_spl/board/samsung/smdk6410/config.mk
#       new file:   nand_spl/board/samsung/smdk6410/u-boot.lds
```

```
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   Makefile
#       modified:   arch/arm/cpu/arm1176/s3c64xx/Makefile
#       modified:   arch/arm/cpu/arm1176/s3c64xx/cpu_init.S
#       modified:   arch/arm/cpu/arm1176/s3c64xx/reset.S
#       modified:   arch/arm/cpu/arm1176/s3c64xx/speed.c
#       modified:   arch/arm/cpu/arm1176/s3c64xx/timer.c
#       modified:   arch/arm/include/asm/arch-s3c64xx/s3c6410.h
#       modified:   board/samsung/smdk6410/Makefile
#       modified:   board/samsung/smdk6410/lowlevel_init.S
#       modified:   board/samsung/smdk6410/smdk6410.c
#       modified:   drivers/mtd/nand/s3c64xx.c
#       modified:   drivers/serial/s3c64xx.c
#       modified:   drivers/usb/host/s3c64xx-hcd.c
#       modified:   include/configs/smdk6410.h
#       modified:   nand_spl/board/samsung/smdk6410/Makefile
#
```

Verify we are right now or not.Compiling.....

Step 4, configuration.

```
$ make ARCH=arm CROSS_COMPILE=arm-unknown-linux-uclibceabi-
smdk6410_config
```

```
warning: Please migrate to boards.cfg.  Failure to do so will
        mean removal of your board in the next release.

Configuring for smdk6410 board...
```

Step 5 ,compiling.

```
$ make ARCH=arm CROSS_COMPILE=arm-unknown-linux-uclibceabi-
.
.
.
.
.
arm-unknown-linux-uclibceabi-objcopy -O binary hello_world hello_world.bin
2>/dev/null
make[1]: Leaving directory `/home/houstar/GitRepo/u-boot/examples/standalone'
make -C examples/api all
make[1]: Entering directory `/home/houstar/GitRepo/u-boot/examples/api'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/houstar/GitRepo/u-boot/examples/api'
```

3,Now we test.

1,Firstly,we should have dnw client that we can transform files through USB.
Dnw.c

```
/*you should use lsusb to find out the actual vender ID & product ID of board.
 * make CFLAGS= -lusb
 *
 * Author:     houstar <houstar@foxmail.com>
 * License:    GPL
 *
 */

#include <stdio.h>
#include <usb.h>
#include <errno.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

#define OK6410_VENDOR_ID        0x04e8
#define OK6410_PRODUCT_ID       0x1234
#define OK6410_RAM_BASE         0x50200000
/*

// FS2410

#define RAM_BASE            FS2410_RAM_BASE

#define VENDOR_ID           FS2410_VENDOR_ID

#define PRODUCT_ID          FS2410_PRODUCT_ID

*/

// OK6410
#define RAM_BASE            OK6410_RAM_BASE
#define VENDOR_ID           OK6410_VENDOR_ID
#define PRODUCT_ID          OK6410_PRODUCT_ID
struct usb_dev_handle * open_port()
{
        struct usb_bus *busses, *bus;
        usb_init();
        usb_find_busses();
        usb_find_devices();

        busses = usb_get_busses();

        for(bus=busses;bus;bus=bus->next)
        {
                struct usb_device *dev;
                for(dev=bus->devices;dev;dev=dev->next)
                {
                        if( VENDOR_ID==dev->descriptor.idVendor
                                        &&  PRODUCT_ID==dev->descriptor.idProduct)
                        {
                                printf("Target usb device found!\n");
```

```
                                struct usb_dev_handle *hdev = usb_open(dev);
                                if(!hdev)
                                {
                                        perror("Cannot open device");
                                }
                                else
                                {
                                        if(0!=usb_claim_interface(hdev, 0))
                                        {
                                                perror("Cannot claim interface");
                                                usb_close(hdev);
                                                hdev = NULL;
                                        }
                                }
                                return hdev;
                        }
                }
        }
        printf("Target usb device not found!\n");
        return NULL;
}
void usage()
{
        printf("Usage: dnw2 <file>\n\n");
}
unsigned char* prepare_write_buf(char *filename, unsigned int *len)
{
        unsigned char *write_buf = NULL;
        struct stat fs;
        int fd = open(filename, O_RDONLY);
        if(-1==fd)
        {
                perror("Cannot open file");
                return NULL;
        }
        if(-1==fstat(fd, &fs))
        {
                perror("Cannot get file size");
                goto error;
        }
        write_buf = (unsigned char*)malloc(fs.st_size+10);
        if(NULL==write_buf)
        {
                perror("malloc failed");
                goto error;
        }
        if(fs.st_size != read(fd, write_buf+8, fs.st_size))
        {
                perror("Reading file failed");
                goto error;
        }
        printf("Filename : %s\n", filename);
        printf("Filesize : %d bytes\n", fs.st_size);

        *((u_int32_t*)write_buf) = RAM_BASE;        //download address
        *((u_int32_t*)write_buf+1) = fs.st_size + 10;    //download size;
```

```
        *len = fs.st_size + 10;
        return write_buf;

error:
        if(fd!=-1) close(fd);
        if(NULL!=write_buf) free(write_buf);
        fs.st_size = 0;

        return NULL;

}

int main(int argc, char *argv[])
{
        if(2!=argc)
        {
                usage();
                return 1;
        }
        struct usb_dev_handle *hdev = open_port();
        if(!hdev)
        {
                return 1;
        }
        unsigned int len = 0;
        unsigned char* write_buf = prepare_write_buf(argv[1], &len);
        if(NULL==write_buf) return 1;
        unsigned int remain = len;
        unsigned int towrite;

        printf("Writing data ...\n");
        while(remain)
        {
                towrite = remain>512 ? 512 : remain;
                if(towrite != usb_bulk_write(hdev, 0x02, write_buf+(len-remain),
towrite, 3000))
                {
                        perror("usb_bulk_write failed");
                        break;
                }
                remain-=towrite;
                printf("\r%d%\t %d bytes     ", (len-remain)*100/len, len-remain);
                fflush(stdout);
        }
        if(0==remain) printf("Done!\n");
        return 0;
}
```

```
$ gcc  -o dnw dnw.c -lusb
$ sudo cp dnw /usr/local/bin
```

Then we flash OK6410_SDboot.nb0 to SD card which can let board boot from USD download. Set OK6410's switch boot from SD card which can print such information to terminal through serial by minicom.

```
+-------------------------------------------+
|  TE6410 USB OTG Downloader v0.1 (2009.12.22)  |
|  Forlinx embedded          www.witech.com.cn  |
+-------------------------------------------+
ARMCLK: 532.00MHz  HCLKx2: 266.00MHz  HCLK: 133.00MHz  PCLK:
66.50MHz
VIC mode / Sync Mode

USB host is not connected yet.
Waiting for USB host connection.

!!! USB host is connected !!!
 - Bulk In EP : 1
 - Bulk Out EP : 2
 - Speed : High
 - Op Mode : DMA mode

Download & Run is selected

Select a file to download in DNW
```

Step 6,Then execute..

```
$ sudo dnw u-boot.bin
Target usb device found!
Filename : u-boot.bin
Filesize : 244816 bytes
Writing data ...
100%     244826 bytes     Done!
```

And the terminal will print such things.

```


[ADDRESS:57e00000h,TOTAL:244826(0x3bc50)]
(2.905MB/s,0.080s)

Checksum is being cal culated....
(If you want to skip, press 'x' key)

Checksum Value => MEM:91fb DNW:0
Checksum failed.
y??

U-Boot 2012.10-g16fc28c-dirty (Mar 29 2013 - 05:45:49) for SMDK6400
```

```
CPU:     S3C6400@533MHz
         Fclk = 533MHz, Hclk = 133MHz, Pclk = 66MHz (ASYNC Mode)
Board:   SMDK6400
DRAM:  128 MiB
WARNING: Caches not enabled
Flash: *** failed ***
### ERROR ### Please RESET the board ###
```

There is an error ocurred cause we havn't flash on the board.
So we uncomment function under line 541 & 301 like this //hang();
Compile and test again.
But still now we have a problem like this,too.
raise: Signal # 8 caught looped never stop.

arch/arm/cpu/arm1176/s3c64xx/timer.c

View file @ 794deb4

```
@@ -37,13 +37,18 @@
  * Foundation, Inc., 59 Temple Place, Suite 330, Boston,
  * MA 02111-1307 USA
  */
-
 #include <common.h>
 #include <asm/proc-armv/ptrace.h>
 #include <asm/arch/s3c6410.h>
 #include <div64.h>

-static ulong timer_load_val;
+//static ulong timer_load_val;
+DECLARE_GLOBAL_DATA_PTR;
+/* modified bugs:raise:Signal 8# caught*/
+#define timer_load_val (gd->timer_rate_hz)
+#define timer_clk     (gd->tbl)
+#define timestamp      (gd->timer_reset_value)
+#define lastdec        (gd->lastinc)

 #define PRESCALER  167

@@ -62,9 +67,9 @@ static inline ulong read_timer(void)

 /* Internal tick units */
 /* Last decremneter snapshot */
-static unsigned long lastdec;
+//static unsigned long lastdec;
 /* Monotonic incrementing timer */
-static unsigned long long timestamp;
+//static unsigned long long timestamp;
```

```
U-Boot 2012.10-g16fc28c-dirty (Mar 29 2013 - 06:29:06) for SMDK6400
```

```
CPU:     S3C6400@533MHz
        Fclk = 533MHz, Hclk = 133MHz, Pclk = 66MHz (ASYNC Mode)
Board:   SMDK6400
DRAM:  128 MiB
WARNING: Caches not enabled
Flash: *** failed ***
NAND:  No oob scheme defined for oobsize 218
4096 MiB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial                          U-Boot 2012.10-g16fc28c-dirty (Mar 29 2013 -
06:29:06) for SMDK6400


CPU:     S3C6400@533MHz
        Fclk = 533MHz, Hclk = 133MHz, Pclk = 66MHz (ASYNC Mode)
Board:   SMDK6400
DRAM:  128 MiB
WARNING: Caches not enabled
Flash: *** failed ***
NAND:  No oob scheme defined for oobsize 218
4096 MiB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   CS8900-0
Hit any key to stop autoboot:  0
SMDK6400 #
```

Lastly,Conclusion:

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   Makefile
#       modified:   arch/arm/cpu/arm1176/s3c64xx/Makefile
#       modified:   arch/arm/cpu/arm1176/s3c64xx/cpu_init.S
#       modified:   arch/arm/cpu/arm1176/s3c64xx/reset.S
#       modified:   arch/arm/cpu/arm1176/s3c64xx/speed.c
#       modified:   arch/arm/cpu/arm1176/s3c64xx/timer.c
#       new file:   arch/arm/include/asm/arch-s3c64xx/s3c6410.h
#       modified:   arch/arm/lib/board.c
```

```
#       new file:   board/samsung/smdk6410/.gitignore
#       new file:   board/samsung/smdk6410/Makefile
#       new file:   board/samsung/smdk6410/config.mk
#       new file:   board/samsung/smdk6410/lowlevel_init.S
#       new file:   board/samsung/smdk6410/smdk6410.c
#       new file:   board/samsung/smdk6410/smdk6410_nand_spl.c
#       new file:   board/samsung/smdk6410/u-boot-nand.lds
#       modified:   drivers/mtd/nand/s3c64xx.c
#       modified:   drivers/serial/s3c64xx.c
#       modified:   drivers/usb/host/s3c64xx-hcd.c
#       new file:   include/configs/smdk6410.h
#       new file:   nand_spl/board/samsung/smdk6410/Makefile
#       new file:   nand_spl/board/samsung/smdk6410/config.mk
#       new file:   nand_spl/board/samsung/smdk6410/u-boot.lds
#
```

Chapter 2.Net Driver

Step 1,Modified files use git status to check.

```
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   arch/arm/lib/cache.c
#       modified:   board/samsung/smdk6410/lowlevel_init.S
#       modified:   board/samsung/smdk6410/smdk6410.c
#       modified:   include/configs/smdk6410.h
#       modified:   nand_spl/u-boot-spl
#       modified:   nand_spl/u-boot-spl.map
#
```

Step 2.Use git diff to see in detail.

```
$ git diff
diff --git a/arch/arm/lib/cache.c b/arch/arm/lib/cache.c
index b545fb7..44f0a7d 100644
--- a/arch/arm/lib/cache.c
+++ b/arch/arm/lib/cache.c
@@ -43,10 +43,10 @@ void  __flush_cache(unsigned long start, unsigned long
size)
 void  flush_cache(unsigned long start, unsigned long size)
         __attribute__((weak, alias("__flush_cache")));

-/*
- * Default implementation:
```

```
- * do a range flush for the entire range
- */
+	/*
+	 * Default implementation:
+	 * do a range flush for the entire range
+	 */
 void __flush_dcache_all(void)
 {
 	flush_cache(0, ~0);
@@ -55,13 +55,17 @@ void flush_dcache_all(void)
 	__attribute__((weak, alias("__flush_dcache_all")));


-/*
- * Default implementation of enable_caches()
- * Real implementation should be in platform code
- */
+	/*
+	 * Default implementation of enable_caches()
+	 * Real implementation should be in platform code
+	 */
 void __enable_caches(void)
 {
-	puts("WARNING: Caches not enabled\n");
+	icache_enable();
+	if(!icache_status())
+	{
+		puts("WARNING: iCaches not enabled\n");
+	}
 }
 void enable_caches(void)
 	__attribute__((weak, alias("__enable_caches")));
diff --git a/board/samsung/smdk6410/lowlevel_init.S
b/board/samsung/smdk6410/lowlevel_init.S
index 5800755..58a546b 100644
--- a/board/samsung/smdk6410/lowlevel_init.S
+++ b/board/samsung/smdk6410/lowlevel_init.S
@@ -51,16 +51,31 @@ _TEXT_BASE:
 lowlevel_init:
 	mov	r12, lr

-	/* LED on only #8 */
-	ldr	r0, =ELFIN_GPIO_BASE
-	ldr	r1, =0x55540000
-	str	r1, [r0, #GPNCON_OFFSET]
-
-	ldr	r1, =0x55555555
-	str	r1, [r0, #GPNPUD_OFFSET]
```

```
-
-       ldr     r1, =0xf000
-       str     r1, [r0, #GPNDAT_OFFSET]
+         ldr   r0, =ELFIN_GPIO_BASE
+         ldr   r1, =0x55555555
+         str   r1, [r0, #GPKCON0_OFFSET]
+
+         ldr   r1, =0x55555555
+         str   r1, [r0, #GPKCON1_OFFSET]
+
+         ldr   r1, =0x22222666
+         str   r1, [r0, #GPLCON0_OFFSET]
+
+         ldr   r1, =0x04000000
+         str   r1, [r0, #GPFCON_OFFSET]
+         ldr   r1, =0x2000
+         str   r1, [r0, #GPFDAT_OFFSET]
+
+        /* LED on only #8 */
+        ldr   r0, =ELFIN_GPIO_BASE
+        ldr   r1, =0x00111111
+        str   r1, [r0, #GPMCON_OFFSET]
+
+        ldr   r1, =0x00000555
+        str   r1, [r0, #GPMPUD_OFFSET]
+
+        ldr   r1, =0x0027
+        str   r1, [r0, #GPMDAT_OFFSET]

        /* Disable Watchdog */
        ldr     r0, =0x7e000000         @0x7e004000
@@ -132,9 +147,9 @@ wakeup_reset:
        str     r1, [r0]

        /* LED test */
-       ldr     r0, =ELFIN_GPIO_BASE
-       ldr     r1, =0x3000
-       str     r1, [r0, #GPNDAT_OFFSET]
+       //ldr   r0, =ELFIN_GPIO_BASE
+       //ldr   r1, =0x0025
+       //str   r1, [r0, #GPMDAT_OFFSET]

        /* Load return address and jump to kernel */
        ldr     r0, =(ELFIN_CLOCK_POWER_BASE + INF_REG0_OFFSET)
@@ -184,7 +199,7 @@ check_syncack:
         * This was unconditional in original Samsung sources, but it doesn't
         * seem to make much sense on S3C6400.
         */
```

```
-#ifndef CONFIG_S3C6400
+//#ifndef CONFIG_S3C6410
        ldr     r1, [r0, #OTHERS_OFFSET]
        bic     r1, r1, #0xC0
        orr     r1, r1, #0x40
@@ -195,7 +210,7 @@ wait_for_async:
        and     r1, r1, #0xf00
        cmp     r1, #0x0
        bne     wait_for_async
-#endif
+//#endif

        ldr     r1, [r0, #OTHERS_OFFSET]
        bic     r1, r1, #0x40
@@ -222,7 +237,7 @@ wait_for_async:
        str     r1, [r0, #MPLL_CON_OFFSET]

        /* FOUT of EPLL is 96MHz */
-       ldr     r1, =0x200203
+       ldr     r1, =0x80200203//zxd 0x200203
        str     r1, [r0, #EPLL_CON0_OFFSET]
        ldr     r1, =0x0
        str     r1, [r0, #EPLL_CON1_OFFSET]
@@ -242,7 +257,7 @@ wait_for_async:
        ldr     r1, [r0, #OTHERS_OFFSET]
        orr     r1, r1, #0x20
        str     r1, [r0, #OTHERS_OFFSET]
-#elif !defined(CONFIG_S3C6400)
+#elif !defined(CONFIG_S3C6410)
        /* According to 661558um_S3C6400X_rev10.pdf 0x20 is reserved */
        ldr     r1, [r0, #OTHERS_OFFSET]
        bic     r1, r1, #0x20
@@ -311,13 +326,13 @@ mmu_table:

        /* 128MB for SDRAM 0xC0000000 -> 0x50000000 */
        .set __base, 0x500
-       .rept 0xC80 - 0xC00
+       .rept 0xD00 - 0xC00
        FL_SECTION_ENTRY __base, 3, 0, 1, 1
        .set __base, __base + 1
        .endr

        /* access is not allowed. */
-       .rept 0x1000 - 0xc80
+       .rept 0x1000 - 0xD00
        .word 0x00000000
        .endr
 #endif
```

```
diff --git a/board/samsung/smdk6410/smdk6410.c
b/board/samsung/smdk6410/smdk6410.c
index a45b516..4c4b0ee 100644
--- a/board/samsung/smdk6410/smdk6410.c
+++ b/board/samsung/smdk6410/smdk6410.c
@@ -46,8 +46,8 @@ DECLARE_GLOBAL_DATA_PTR;
 static inline void delay(unsigned long loops)
 {
        __asm__ volatile ("1:\n" "subs %0, %1, #1\n"
-                         "bne 1b"
-                         : "=r" (loops) : "0" (loops));
+                        "bne 1b"
+                        : "=r" (loops) : "0" (loops));
 }

 /*
@@ -86,7 +86,7 @@ void dram_init_banksize(void)
 int dram_init(void)
 {
        gd->ram_size = get_ram_size((long *)CONFIG_SYS_SDRAM_BASE,
-                                   PHYS_SDRAM_1_SIZE);
+                        PHYS_SDRAM_1_SIZE);

        return 0;
 }
@@ -94,7 +94,7 @@ int dram_init(void)
 #ifdef CONFIG_DISPLAY_BOARDINFO
 int checkboard(void)
 {
-       printf("Board:   SMDK6400\n");
+       printf("Board:   SMDK6410\n");
        return 0;
 }
 #endif
@@ -129,6 +129,9 @@ int board_eth_init(bd_t *bis)
 #ifdef CONFIG_CS8900
        rc = cs8900_initialize(0, CONFIG_CS8900_BASE);
 #endif
+#if defined(CONFIG_DRIVER_DM9000)
+       rc = dm9000_initialize(bis);
+#endif
        return rc;
 }
 #endif
diff --git a/include/configs/smdk6410.h b/include/configs/smdk6410.h
index b965636..d25ad60 100644
--- a/include/configs/smdk6410.h
+++ b/include/configs/smdk6410.h
```

```
@@ -65,7 +65,7 @@
 /*
  * Architecture magic and machine type
  */
-#define CONFIG_MACH_TYPE            1270
+#define CONFIG_MACH_TYPE            1626

 #define CONFIG_DISPLAY_CPUINFO
 #define CONFIG_DISPLAY_BOARDINFO
@@ -78,10 +78,24 @@
 /*
  * Hardware drivers
  */
-#define CONFIG_CS8900               /* we have a CS8900 on-board      */
-#define CONFIG_CS8900_BASE          0x18800300
-#define CONFIG_CS8900_BUS16         /* follow the Linux driver      */
-
+//#define CONFIG_CS8900             /* we have a CS8900 on-board      */
+//#define CONFIG_CS8900_BASE                0x18800300
+//#define CONFIG_CS8900_BUS16               /* follow the Linux driver      */
+/*
+ * DM9000 Drivers
+ */
+ #define CONFIG_DRIVER_DM9000 1
+#define CONFIG_DM9000_BASE (0x18000300) /*XM0CSN1*/
+#define DM9000_IO (CONFIG_DM9000_BASE)
+#define DM9000_DATA (CONFIG_DM9000_BASE+0x4) /*ADDR2*/
+#define CONFIG_DM9000_DEBUG 1
+#define CONFIG_DM9000_USE_16BIT 1
+
+#define CONFIG_ETHADDR                  08:08:10:12:10:27
+#define CONFIG_NETMASK                  255.255.255.0
+#define CONFIG_IPADDR               192.168.1.232
+#define CONFIG_SERVERIP                 192.168.1.101
+#define CONFIG_GATEWAYIP            192.168.1.1
 /*
  * select serial console configuration
  */
@@ -163,7 +177,7 @@
 /* SMDK6400 has 2 banks of DRAM, but we use only one in U-Boot */
 #define CONFIG_NR_DRAM_BANKS    1
 #define PHYS_SDRAM_1            CONFIG_SYS_SDRAM_BASE/* SDRAM Bank
#1    */
-#define PHYS_SDRAM_1_SIZE   0x08000000      /* 128 MB in Bank #1  */
+#define PHYS_SDRAM_1_SIZE   0x10000000      /* 128 MB in Bank #1  */

 #define CONFIG_SYS_FLASH_BASE           0x10000000
 #define CONFIG_SYS_MONITOR_BASE         0x00000000
```

```
@@ -235,11 +249,11 @@
 #define CONFIG_SYS_NAND_U_BOOT_SIZE (252 * 1024)        /* Size of RAM U-
Boot image   */

 /* NAND chip page size                  */
-#define CONFIG_SYS_NAND_PAGE_SIZE       2048
+#define CONFIG_SYS_NAND_PAGE_SIZE     (2048 * 2)
 /* NAND chip block size                 */
-#define CONFIG_SYS_NAND_BLOCK_SIZE    (128 * 1024)
+#define CONFIG_SYS_NAND_BLOCK_SIZE  (128 * 4 * 1024)
 /* NAND chip page per block count  */
-#define CONFIG_SYS_NAND_PAGE_COUNT  64
+#define CONFIG_SYS_NAND_PAGE_COUNT(64 * 2)
 /* Location of the bad-block label */
 #define CONFIG_SYS_NAND_BAD_BLOCK_POS      0
 /* Extra address cycle for > 128MiB */
```

Step 3,Testing...

```
U-Boot 2012.10-g17a0520-dirty (Mar 29 2013 - 21:52:32) for SMDK6410


CPU:     S3C6400@533MHz
         Fclk = 533MHz, Hclk = 133MHz, Pclk = 66MHz (ASYNC Mode)
Board:   SMDK6410
DRAM:  256 MiB
Flash: *** failed ***
NAND:  No oob scheme defined for oobsize 218
4096 MiB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   dm9000
Hit any key to stop autoboot:  0
SMDK6410 # ping $serverip
Host 192.162.1.101 is alive.
SMDK6410 #
```

Chapter 3.Nand Related

Step 1,Slove oob error

As we know from above,Nand:No oob scheme defined for oobsize 218.

1,Add these code after nand_oob_128 in drivers/mtd/nand/nand_base.c

```
static struct nand_ecclayout nand_oob_218 = {
    .eccbytes = 104,
    .eccpos = {
        24,25,26,27,28,29,30,31,32,33,
        34,35,36,37,38,39,40,41,42,43,
        44,45,46,47,48,49,50,51,52,53,
        54,55,56,57,58,59,60,61,62,63,
        64,65,66,67,68,69,70,71,72,73,
        74,75,76,77,78,79,80,81,82,83,
        84,85,86,87,88,89,90,91,92,93,
        94,95,96,97,98,99,100,101,102,103,
        104,105,106,107,108,109,110,111,112,113,
        114,115,116,117,118,119,120,121,122,123,
        124,125,126,127},
    .oobfree =
      {
        {.offset = 2,
         .length = 22
        }
      }
};
```

Test:

```
U-Boot 2012.10-g9a787ba-dirty (Mar 29 2013 - 22:57:58) for SMDK6410


CPU:     S3C6400@533MHz
         Fclk = 533MHz, Hclk = 133MHz, Pclk = 66MHz (ASYNC Mode)
Board:   SMDK6410
DRAM:  256 MiB
Flash: *** failed ***
NAND:  4096 MiB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   dm9000
Hit any key to stop autoboot:  0

NAND read: device 0 offset 0x60000, size 0x1c0000
 1835008 bytes read: OK
Wrong Image Format for bootm command
ERROR: can't get kernel image!
SMDK6410 #
```

2,In function add these after case 128 :drivers/mtd/nand/nand_base.c

```
case 218:
        chip->ecc.layout = &nand_oob_218;
        break;
```

Step 2,8 Bit ECC(Error Checking and Correction)

1,smdk6410.h /include/configs/

```
index d25ad60..d4dc707 100644
@@ -260,9 +260,15 @@
 #define CONFIG_SYS_NAND_5_ADDR_CYCLE


 /* Size of the block protected by one OOB (Spare Area in Samsung terminology)
*/
-#define CONFIG_SYS_NAND_ECCSIZE CONFIG_SYS_NAND_PAGE_SIZE
+#define CONFIG_SYS_NAND_ECCSIZE
        512//CONFIG_SYS_NAND_PAGE_SIZE
 /* Number of ECC bytes per OOB - S3C6400 calculates 4 bytes ECC in 1-bit
mode */
+#define CONFIG_NAND_BL1_8BIT_ECC
+
+#ifdef CONFIG_NAND_BL1_8BIT_ECC
+#define CONFIG_SYS_NAND_ECCBYTES     13
+#else
 #define CONFIG_SYS_NAND_ECCBYTES      4
+#endif
 /* Size of a single OOB region */
 #define CONFIG_SYS_NAND_OOBSIZE64
 /* ECC byte positions */
```

2./arch/arm/include/asm/arch-s3c64xx/s3c6410.h

```
---------- arch/arm/include/asm/arch-s3c64xx/s3c6410.h ----------
index eef8308..ea96791 100644
@@ -574,6 +574,17 @@
```

```
 #define NFSECC_OFFSET          0x3c
 #define NFMLCBITPT_OFFSET    0x40


+
+#define NF8ECCERR0_OFFSET 0x44
+#define NF8ECCERR1_OFFSET 0x48
+#define NF8ECCERR2_OFFSET 0x4c
+#define NFM8ECC0_OFFSET           0x50
+#define NFM8ECC1_OFFSET           0x54
+#define NFM8ECC2_OFFSET           0x58
+#define NFM8ECC3_OFFSET           0x5c
+#define NFMLC8BITPT0_OFFSET       0x60
+#define NFMLC8BITPT1_OFFSET       0x64
+
 #define NFCONF                  (ELFIN_NAND_BASE + NFCONF_OFFSET)
 #define NFCONT                  (ELFIN_NAND_BASE + NFCONT_OFFSET)
 #define NFCMMD                  (ELFIN_NAND_BASE + NFCMMD_OFFSET)
@@ -591,6 +602,15 @@
 #define NFMECC1                 (ELFIN_NAND_BASE + NFMECC1_OFFSET)
 #define NFSECC                  (ELFIN_NAND_BASE + NFSECC_OFFSET)
 #define NFMLCBITPT       (ELFIN_NAND_BASE + NFMLCBITPT_OFFSET)
+#define NF8ECCERR0     (ELFIN_NAND_BASE+NF8ECCERR0_OFFSET)
+#define NF8ECCERR1
       (ELFIN_NAND_BASE+NF8ECCERR1_OFFSET)
+#define NF8ECCERR2
       (ELFIN_NAND_BASE+NF8ECCERR2_OFFSET)
+#define NFM8ECC0         (ELFIN_NAND_BASE+NFM8ECC0_OFFSET)
+#define NFM8ECC1         (ELFIN_NAND_BASE+NFM8ECC1_OFFSET)
+#define NFM8ECC2         (ELFIN_NAND_BASE+NFM8ECC2_OFFSET)
+#define NFM8ECC3         (ELFIN_NAND_BASE+NFM8ECC3_OFFSET)
+#define NFMLC8BITPT0
       (ELFIN_NAND_BASE+NFMLC8BITPT0_OFFSET)
+#define NFMLC8BITPT1
       (ELFIN_NAND_BASE+NFMLC8BITPT1_OFFSET)
```

```
 #define NFCONF_REG              __REG(ELFIN_NAND_BASE + NFCONF_OFFSET)
 #define NFCONT_REG              __REG(ELFIN_NAND_BASE + NFCONT_OFFSET)
```

3,drivers/mtd/nand/s3c64xx.c

```
---------- drivers/mtd/nand/s3c64xx.c ----------
index bafc1a7..acb4659 100644
@@ -122,6 +122,218 @@ static int s3c_nand_device_ready(struct mtd_info *mtdinfo)
  * This function is called before encoding ecc codes to ready ecc engine.
  * Written by jsgood
  */
+
+#if defined(CONFIG_NAND_BL1_8BIT_ECC) && (defined(CONFIG_S3C6410) || defined(CONFIG_S3C6430))
+ /*********************************************************************
+  * jsgood: Temporary 8 Bit H/W ECC supports for BL1 (6410/6430 only)
+  *********************************************************************/
+ int cur_ecc_mode=0;
+       /*
+        * Function for checking ECCEncDone in NFSTAT
+        * Written by jsgood
+        */
+       static void s3c_nand_wait_enc(void)
+       {
+               while (!(readl(NFSTAT) & NFSTAT_ECCENCDONE)) {}
+       }
+
+       /*
+        * Function for checking ECCDecDone in NFSTAT
+        * Written by jsgood
+        */
+       static void s3c_nand_wait_dec(void)
+       {
```

```
+               while (!(readl(NFSTAT) & NFSTAT_ECCDECDONE)) {}
+       }
+
+ static void s3c_nand_wait_ecc_busy_8bit(void)
+ {
+       while (readl(NF8ECCERR0) & NFESTAT0_ECCBUSY) {}
+ }
+ void s3c_nand_enable_hwecc_8bit(struct mtd_info *mtd, int mode)
+{
+    u_long nfcont, nfconf;
+
+    cur_ecc_mode = mode;
+
+    /* 8 bit selection */
+    nfconf = readl(NFCONF);
+
+    nfconf &= ~(0x3 << 23);
+    nfconf |= (0x1 << 23);
+
+    writel(nfconf, NFCONF);
+
+    /* Initialize & unlock */
+    nfcont = readl(NFCONT);
+    nfcont |= NFCONT_INITECC;
+    nfcont &= ~NFCONT_MECCLOCK;
+
+    if (mode == NAND_ECC_WRITE)
+        nfcont |= NFCONT_ECC_ENC;
+    else if (mode == NAND_ECC_READ)
+        nfcont &= ~NFCONT_ECC_ENC;
+
+    writel(nfcont, NFCONT);
+}
+int s3c_nand_calculate_ecc_8bit(struct mtd_info *mtd, const u_char *dat,
```

```
u_char *ecc_code)
+{
+    u_long nfcont, nfm8ecc0, nfm8ecc1, nfm8ecc2, nfm8ecc3;
+
+    /* Lock */
+    nfcont = readl(NFCONT);
+    nfcont |= NFCONT_MECCLOCK;
+    writel(nfcont, NFCONT);
+
+    if (cur_ecc_mode == NAND_ECC_READ)
+        s3c_nand_wait_dec();
+    else {
+        s3c_nand_wait_enc();
+
+        nfm8ecc0 = readl(NFM8ECC0);
+        nfm8ecc1 = readl(NFM8ECC1);
+        nfm8ecc2 = readl(NFM8ECC2);
+        nfm8ecc3 = readl(NFM8ECC3);
+
+        ecc_code[0] = nfm8ecc0 & 0xff;
+        ecc_code[1] = (nfm8ecc0 >> 8) & 0xff;
+        ecc_code[2] = (nfm8ecc0 >> 16) & 0xff;
+        ecc_code[3] = (nfm8ecc0 >> 24) & 0xff;
+        ecc_code[4] = nfm8ecc1 & 0xff;
+        ecc_code[5] = (nfm8ecc1 >> 8) & 0xff;
+        ecc_code[6] = (nfm8ecc1 >> 16) & 0xff;
+        ecc_code[7] = (nfm8ecc1 >> 24) & 0xff;
+        ecc_code[8] = nfm8ecc2 & 0xff;
+        ecc_code[9] = (nfm8ecc2 >> 8) & 0xff;
+        ecc_code[10] = (nfm8ecc2 >> 16) & 0xff;
+        ecc_code[11] = (nfm8ecc2 >> 24) & 0xff;
+        ecc_code[12] = nfm8ecc3 & 0xff;
+    }
+
```

```
+	return 0;
+}
+
+int s3c_nand_correct_data_8bit(struct mtd_info *mtd, u_char *dat, u_char
*read_ecc, u_char *calc_ecc)
+{
+	int ret = -1;
+	u_long nf8eccerr0, nf8eccerr1, nf8eccerr2, nfmlc8bitpt0, nfmlc8bitpt1;
+	u_char err_type;
+
+	s3c_nand_wait_ecc_busy_8bit();
+
+	nf8eccerr0 = readl(NF8ECCERR0);
+	nf8eccerr1 = readl(NF8ECCERR1);
+	nf8eccerr2 = readl(NF8ECCERR2);
+	nfmlc8bitpt0 = readl(NFMLC8BITPT0);
+	nfmlc8bitpt1 = readl(NFMLC8BITPT1);
+
+	err_type = (nf8eccerr0 >> 25) & 0xf;
+
+	/* No error, If free page (all 0xff) */
+	if ((nf8eccerr0 >> 29) & 0x1)
+		err_type = 0;
+
+	switch (err_type)
+	{
+	case 8: /* 8 bit error (Correctable) */
+		dat[(nf8eccerr2 >> 22) & 0x3ff] ^= ((nfmlc8bitpt1 >> 24) & 0xff);
+		printk("s3c-nand: %d bit(s) error detected, corrected successfully\n",
err_type);
+
+	case 7: /* 7 bit error (Correctable) */
+		dat[(nf8eccerr2 >> 11) & 0x3ff] ^= ((nfmlc8bitpt1 >> 16) & 0xff);
+		printk("s3c-nand: %d bit(s) error detected, corrected successfully\n",
err_type);
```

```
+
+    case 6: /* 6 bit error (Correctable) */
+        dat[nf8eccerr2 & 0x3ff] ^= ((nfmlc8bitpt1 >> 8) & 0xff);
+        printk("s3c-nand: %d bit(s) error detected, corrected successfully\n",
err_type);
+
+    case 5: /* 5 bit error (Correctable) */
+        dat[(nf8eccerr1 >> 22) & 0x3ff] ^= (nfmlc8bitpt1 & 0xff);
+        printk("s3c-nand: %d bit(s) error detected, corrected successfully\n",
err_type);
+
+    case 4: /* 4 bit error (Correctable) */
+        dat[(nf8eccerr1 >> 11) & 0x3ff] ^= ((nfmlc8bitpt0 >> 24) & 0xff);
+        printk("s3c-nand: %d bit(s) error detected, corrected successfully\n",
err_type);
+
+    case 3: /* 3 bit error (Correctable) */
+        dat[nf8eccerr1 & 0x3ff] ^= ((nfmlc8bitpt0 >> 16) & 0xff);
+        printk("s3c-nand: %d bit(s) error detected, corrected successfully\n",
err_type);
+
+    case 2: /* 2 bit error (Correctable) */
+        dat[(nf8eccerr0 >> 15) & 0x3ff] ^= ((nfmlc8bitpt0 >> 8) & 0xff);
+        printk("s3c-nand: %d bit(s) error detected, corrected successfully\n",
err_type);
+
+
+    case 1: /* 1 bit error (Correctable) */
+        printk("s3c-nand: %d bit(s) error detected, corrected successfully\n",
err_type);
+        dat[nf8eccerr0 & 0x3ff] ^= (nfmlc8bitpt0 & 0xff);
+        ret = err_type;
+        break;
+
+    case 0: /* No error */
+        ret = 0;
```

```
+		break;
+
+	}
+
+	return ret;
+}
+
+void s3c_nand_write_page_8bit(struct mtd_info *mtd, struct nand_chip *chip,
+				const uint8_t *buf)
+{
+	int i, eccsize = 512;
+	int eccbytes = 13;
+	int eccsteps = mtd->writesize / eccsize;
+	uint8_t *ecc_calc = chip->buffers->ecccalc;
+	uint8_t *p = buf;
+
+	for (i = 0; eccsteps; eccsteps--, i += eccbytes, p += eccsize) {
+		s3c_nand_enable_hwecc_8bit(mtd, NAND_ECC_WRITE);
+		chip->write_buf(mtd, p, eccsize);
+		s3c_nand_calculate_ecc_8bit(mtd, p, &ecc_calc[i]);
+	}
+
+	for (i = 0; i < eccbytes * (mtd->writesize / eccsize); i++)
+		chip->oob_poi[i+24] = ecc_calc[i];
+	chip->write_buf(mtd, chip->oob_poi, mtd->oobsize);
+}
+
+int s3c_nand_read_page_8bit(struct mtd_info *mtd, struct nand_chip *chip,
+				uint8_t *buf)
+{
+	int i, stat, eccsize = 512;
+	int eccbytes = 13;
+	int eccsteps = mtd->writesize / eccsize;
+	int col = 0;
```

```
+    uint8_t *p = buf;
+
+    /* Step1: read whole oob */
+    col = mtd->writesize;
+    chip->cmdfunc(mtd, NAND_CMD_RNDOUT, col, -1);
+    chip->read_buf(mtd, chip->oob_poi, mtd->oobsize);
+
+    col = 0;
+    for (i = 0; eccsteps; eccsteps--, i += eccbytes, p += eccsize) {
+        chip->cmdfunc(mtd, NAND_CMD_RNDOUT, col, -1);
+        s3c_nand_enable_hwecc_8bit(mtd, NAND_ECC_READ);
+        chip->read_buf(mtd, p, eccsize);
+        chip->write_buf(mtd, chip->oob_poi + 24 + (((mtd->writesize / eccsize) -
eccsteps) * eccbytes), eccbytes);
+        s3c_nand_calculate_ecc_8bit(mtd, 0, 0);
+        stat = s3c_nand_correct_data_8bit(mtd, p, 0, 0);
+
+        if (stat == -1)
+            mtd->ecc_stats.failed++;
+
+        col = eccsize * ((mtd->writesize / eccsize) + 1 - eccsteps);
+    }
+
+    return 0;
+}
+
+/**************************************************************/
+#endif
+
 static void s3c_nand_enable_hwecc(struct mtd_info *mtd, int mode)
 {
         u_long nfcont, nfconf;
@@ -131,8 +343,9 @@ static void s3c_nand_enable_hwecc(struct mtd_info *mtd,
int mode)
```

```
         * those with non-zero ID[3][3:2], which anyway only holds for ST
         * (Numonyx) chips
         */
-        nfconf = readl(NFCONF) & ~NFCONF_ECC_4BIT;

+        nfconf = readl(NFCONF) & ~NFCONF_ECC_4BIT;
+
         writel(nfconf, NFCONF);

         /* Initialize & unlock */
@@ -157,7 +370,6 @@ static int s3c_nand_calculate_ecc(struct mtd_info *mtd,
const u_char *dat,
                                 u_char *ecc_code)
 {
         u_long nfcont, nfmecc0;
-
         /* Lock */
         nfcont = readl(NFCONT);
         nfcont |= NFCONT_MECCLOCK;
@@ -169,7 +381,7 @@ static int s3c_nand_calculate_ecc(struct mtd_info *mtd,
const u_char *dat,
         ecc_code[1] = (nfmecc0 >> 8) & 0xff;
         ecc_code[2] = (nfmecc0 >> 16) & 0xff;
         ecc_code[3] = (nfmecc0 >> 24) & 0xff;
-
+
         return 0;
 }

@@ -274,10 +486,19 @@ int board_nand_init(struct nand_chip *nand)
 #endif

 #ifdef CONFIG_SYS_S3C_NAND_HWECC
+    #ifdef CONFIG_NAND_BL1_8BIT_ECC
```

```
+       printf("USE HWECC 8BIT\n");//zxd
+       nand->ecc.hwctl         = s3c_nand_enable_hwecc_8bit;
+       nand->ecc.calculate     = s3c_nand_calculate_ecc_8bit;
+       nand->ecc.correct       = s3c_nand_correct_data_8bit;
+       nand->ecc.read_page = s3c_nand_read_page_8bit;
+     nand->ecc.write_page = s3c_nand_write_page_8bit;
+     #else
+       printf("USE HWECC Default\n");//zxd
        nand->ecc.hwctl         = s3c_nand_enable_hwecc;
        nand->ecc.calculate     = s3c_nand_calculate_ecc;
        nand->ecc.correct       = s3c_nand_correct_data;
-
+     #endif
        /*
         * If you get more than 1 NAND-chip with different page-sizes on the
         * board one day, it will get more complicated...
@@ -285,6 +506,7 @@ int board_nand_init(struct nand_chip *nand)
        nand->ecc.mode          = NAND_ECC_HW;
        nand->ecc.size          = CONFIG_SYS_NAND_ECCSIZE;
        nand->ecc.bytes         = CONFIG_SYS_NAND_ECCBYTES;
+     printf("ECC Size:%d ECC Bytes:%d\n",nand->ecc.size,nand->ecc.bytes);//zxd
  #else
        nand->ecc.mode          = NAND_ECC_SOFT;
  #endif /* ! CONFIG_SYS_S3C_NAND_HWECC */
```

4.Testing..

```
U-Boot 2012.10-g9a787ba-dirty (Mar 30 2013 - 00:26:21) for SMDK6410



CPU:     S3C6400@533MHz
         Fclk = 533MHz, Hclk = 133MHz, Pclk = 66MHz (ASYNC Mode)
Board:   SMDK6410
DRAM:  256 MiB
Flash: *** failed ***
```

```
NAND:  USE HWECC 8BIT
ECC Size:512 ECC Bytes:13
4096 MiB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   dm9000
Hit any key to stop autoboot:  0
SMDK6410 #
```

5.Conclusion

```
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add/rm <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   arch/arm/include/asm/arch-s3c64xx/s3c6410.h
#       modified:   drivers/mtd/nand/nand_base.c
#       modified:   drivers/mtd/nand/s3c64xx.c
#       modified:   include/configs/smdk6410.h
#       deleted:    nand_spl/board/samsung/smdk6410/cpu_init.S
#       deleted:    nand_spl/board/samsung/smdk6410/lowlevel_init.S
#       deleted:    nand_spl/board/samsung/smdk6410/nand_base.c
#       deleted:    nand_spl/board/samsung/smdk6410/nand_boot.c
#       deleted:    nand_spl/board/samsung/smdk6410/nand_ecc.c
#       deleted:    nand_spl/board/samsung/smdk6410/s3c64xx.c
#       deleted:    nand_spl/board/samsung/smdk6410/smdk6410_nand_spl.c
#       deleted:    nand_spl/board/samsung/smdk6410/start.S
#       deleted:    nand_spl/u-boot-spl
#       deleted:    nand_spl/u-boot-spl.map
#       deleted:    nand_spl/u-boot.lds
```

#
no changes added to commit (use "git add" and/or "git commit -a")

Chapter 4,DNW

Step 1.Prepare files.

```
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   arch/arm/include/asm/arch-s3c64xx/s3c6410.h
#       modified:   common/Makefile
#       modified:   drivers/usb/host/Makefile
#       modified:   include/configs/smdk6410.h
#       modified:   nand_spl/u-boot-spl
#       modified:   nand_spl/u-boot-spl.map
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       common/cmd_usbd.c
#       drivers/usb/host/s3c6410-usbd-otg.c
#       drivers/usb/host/s3c6410-usbd-otg.h
no changes added to commit (use "git add" and/or "git commit -a")
```

You can see we need cmd_usbd.c,s3c6410-usbd-otg.c & s3c6410-usbd-otg.h .

copy cmd_usbd.c to /common/

copy s3c6410-usbd-otg.c & s3c6410-usbd-otg.h to /drivers/usb/host/

Step 2. Modify head file

```
---------- arch/arm/include/asm/arch-s3c64xx/s3c6410.h ----------
index ea96791..db2d894 100644
@@ -904,6 +904,102 @@ enum s3c64xx_uarts_nr {
```

```
        S3C64XX_UART2,
 };


+/*
+ * USB2.0 HS OTG (Chapter 4)
+ */
+#define USBOTG_LINK_BASE    (0x7C000000)
+#define USBOTG_PHY_BASE             (0x7C100000)
+
+/* Core Global Registers */
+#define S3C_OTG_GOTGCTL             (USBOTG_LINK_BASE + 0x000)    /*
OTG Control & Status */
+#define S3C_OTG_GOTGINT             (USBOTG_LINK_BASE + 0x004)    /*
OTG Interrupt */
+#define S3C_OTG_GAHBCFG             (USBOTG_LINK_BASE + 0x008)    /*
Core AHB Configuration */
+#define S3C_OTG_GUSBCFG             (USBOTG_LINK_BASE + 0x00C)    /*
Core USB Configuration */
+#define S3C_OTG_GRSTCTL             (USBOTG_LINK_BASE + 0x010)    /*
Core Reset */
+#define S3C_OTG_GINTSTS             (USBOTG_LINK_BASE + 0x014)    /*
Core Interrupt */
+#define S3C_OTG_GINTMSK             (USBOTG_LINK_BASE + 0x018)    /*
Core Interrupt Mask */
+#define S3C_OTG_GRXSTSR             (USBOTG_LINK_BASE + 0x01C)    /*
Receive Status Debug Read/Status Read */
+#define S3C_OTG_GRXSTSP             (USBOTG_LINK_BASE + 0x020)    /*
Receive Status Debug Pop/Status Pop */
+#define S3C_OTG_GRXFSIZ             (USBOTG_LINK_BASE + 0x024)    /*
Receive FIFO Size */
+#define S3C_OTG_GNPTXFSIZ   (USBOTG_LINK_BASE + 0x028)    /* Non-
Periodic Transmit FIFO Size */
+#define S3C_OTG_GNPTXSTS    (USBOTG_LINK_BASE + 0x02C)    /* Non-
Periodic Transmit FIFO/Queue Status */
+
+#define S3C_OTG_HPTXFSIZ    (USBOTG_LINK_BASE + 0x100)    /* Host
Periodic Transmit FIFO Size */
+#define S3C_OTG_DPTXFSIZ1   (USBOTG_LINK_BASE + 0x104)    /* Device
```

Periodic Transmit FIFO-1 Size */

+#define S3C_OTG_DPTXFSIZ2 (USBOTG_LINK_BASE + 0x108) /* Device Periodic Transmit FIFO-2 Size */

+#define S3C_OTG_DPTXFSIZ3 (USBOTG_LINK_BASE + 0x10C) /* Device Periodic Transmit FIFO-3 Size */

+#define S3C_OTG_DPTXFSIZ4 (USBOTG_LINK_BASE + 0x110) /* Device Periodic Transmit FIFO-4 Size */

+#define S3C_OTG_DPTXFSIZ5 (USBOTG_LINK_BASE + 0x114) /* Device Periodic Transmit FIFO-5 Size */

+#define S3C_OTG_DPTXFSIZ6 (USBOTG_LINK_BASE + 0x118) /* Device Periodic Transmit FIFO-6 Size */

+#define S3C_OTG_DPTXFSIZ7 (USBOTG_LINK_BASE + 0x11C) /* Device Periodic Transmit FIFO-7 Size */

+#define S3C_OTG_DPTXFSIZ8 (USBOTG_LINK_BASE + 0x120) /* Device Periodic Transmit FIFO-8 Size */

+#define S3C_OTG_DPTXFSIZ9 (USBOTG_LINK_BASE + 0x124) /* Device Periodic Transmit FIFO-9 Size */

+#define S3C_OTG_DPTXFSIZ10 (USBOTG_LINK_BASE + 0x128) /* Device Periodic Transmit FIFO-10 Size */

+#define S3C_OTG_DPTXFSIZ11 (USBOTG_LINK_BASE + 0x12C) /* Device Periodic Transmit FIFO-11 Size */

+#define S3C_OTG_DPTXFSIZ12 (USBOTG_LINK_BASE + 0x130) /* Device Periodic Transmit FIFO-12 Size */

+#define S3C_OTG_DPTXFSIZ13 (USBOTG_LINK_BASE + 0x134) /* Device Periodic Transmit FIFO-13 Size */

+#define S3C_OTG_DPTXFSIZ14 (USBOTG_LINK_BASE + 0x138) /* Device Periodic Transmit FIFO-14 Size */

+#define S3C_OTG_DPTXFSIZ15 (USBOTG_LINK_BASE + 0x13C) /* Device Periodic Transmit FIFO-15 Size */

+

+/* Host Global Registers */

+#define S3C_OTG_HCFG (USBOTG_LINK_BASE + 0x400) /* Host Configuration */

+#define S3C_OTG_HFIR (USBOTG_LINK_BASE + 0x404) /* Host Frame Interval */

+#define S3C_OTG_HFNUM (USBOTG_LINK_BASE + 0x408) /* Host Frame Number/Frame Time Remaining */

+#define S3C_OTG_HPTXSTS (USBOTG_LINK_BASE + 0x410) /* Host Periodic Transmit FIFO/Queue Status */

+#define S3C_OTG_HAINT (USBOTG_LINK_BASE + 0x414) /* Host All Channels Interrupt */

+#define S3C_OTG_HAINTMSK (USBOTG_LINK_BASE + 0x418) /* Host All Channels Interrupt Mask */

+

+/* Host Port Control & Status Registers */

+#define S3C_OTG_HPRT (USBOTG_LINK_BASE + 0x440) /* Host Port Control & Status */

+

+/* Host Channel-Specific Registers */

+#define S3C_OTG_HCCHAR0 (USBOTG_LINK_BASE + 0x500) /* Host Channel-0 Characteristics */

+#define S3C_OTG_HCSPLT0 (USBOTG_LINK_BASE + 0x504) /* Host Channel-0 Split Control */

+#define S3C_OTG_HCINT0 (USBOTG_LINK_BASE + 0x508) /* Host Channel-0 Interrupt */

+#define S3C_OTG_HCINTMSK0 (USBOTG_LINK_BASE + 0x50C) /* Host Channel-0 Interrupt Mask */

+#define S3C_OTG_HCTSIZ0 (USBOTG_LINK_BASE + 0x510) /* Host Channel-0 Transfer Size */

+#define S3C_OTG_HCDMA0 (USBOTG_LINK_BASE + 0x514) /* Host Channel-0 DMA Address */

+

+

+/* Device Global Registers */

+#define S3C_OTG_DCFG (USBOTG_LINK_BASE + 0x800) /* Device Configuration */

+#define S3C_OTG_DCTL (USBOTG_LINK_BASE + 0x804) /* Device Control */

+#define S3C_OTG_DSTS (USBOTG_LINK_BASE + 0x808) /* Device Status */

+#define S3C_OTG_DIEPMSK (USBOTG_LINK_BASE + 0x810) /* Device IN Endpoint Common Interrupt Mask */

+#define S3C_OTG_DOEPMSK (USBOTG_LINK_BASE + 0x814) /* Device OUT Endpoint Common Interrupt Mask */

+#define S3C_OTG_DAINT (USBOTG_LINK_BASE + 0x818) /* Device All Endpoints Interrupt */

+#define S3C_OTG_DAINTMSK (USBOTG_LINK_BASE + 0x81C) /* Device All Endpoints Interrupt Mask */

+#define S3C_OTG_DTKNQR1	(USBOTG_LINK_BASE + 0x820)	/* Device IN Token Sequence Learning Queue Read 1 */

+#define S3C_OTG_DTKNQR2	(USBOTG_LINK_BASE + 0x824)	/* Device IN Token Sequence Learning Queue Read 2 */

+#define S3C_OTG_DVBUSDIS	(USBOTG_LINK_BASE + 0x828)	/* Device VBUS Discharge Time */

+#define S3C_OTG_DVBUSPULSE	(USBOTG_LINK_BASE + 0x82C)	/* Device VBUS Pulsing Time */

+#define S3C_OTG_DTKNQR3	(USBOTG_LINK_BASE + 0x830)	/* Device IN Token Sequence Learning Queue Read 3 */

+#define S3C_OTG_DTKNQR4	(USBOTG_LINK_BASE + 0x834)	/* Device IN Token Sequence Learning Queue Read 4 */

+

+/* Device Logical IN Endpoint-Specific Registers */

+#define S3C_OTG_DIEPCTL0	(USBOTG_LINK_BASE + 0x900)	/* Device IN Endpoint 0 Control */

+#define S3C_OTG_DIEPINT0	(USBOTG_LINK_BASE + 0x908)	/* Device IN Endpoint 0 Interrupt */

+#define S3C_OTG_DIEPTSIZ0	(USBOTG_LINK_BASE + 0x910)	/* Device IN Endpoint 0 Transfer Size */

+#define S3C_OTG_DIEPDMA0	(USBOTG_LINK_BASE + 0x914)	/* Device IN Endpoint 0 DMA Address */

+

+/* Device Logical OUT Endpoint-Specific Registers */

+#define S3C_OTG_DOEPCTL0	(USBOTG_LINK_BASE + 0xB00)	/* Device OUT Endpoint 0 Control */

+#define S3C_OTG_DOEPINT0	(USBOTG_LINK_BASE + 0xB08)	/* Device OUT Endpoint 0 Interrupt */

+#define S3C_OTG_DOEPTSIZ0	(USBOTG_LINK_BASE + 0xB10)	/* Device OUT Endpoint 0 Transfer Size */

+#define S3C_OTG_DOEPDMA0	(USBOTG_LINK_BASE + 0xB14)	/* Device OUT Endpoint 0 DMA Address */

+

+/* Power & clock gating registers */

+#define S3C_OTG_PCGCCTRL	(USBOTG_LINK_BASE + 0xE00)

+

+/* Endpoint FIFO address */

+#define S3C_OTG_EP0_FIFO	(USBOTG_LINK_BASE + 0x1000)

```
+
+
+
+/* OTG PHY CORE REGISTERS */
+#define S3C_OTG_PHYPWR              (USBOTG_PHY_BASE+0x00)
+#define S3C_OTG_PHYCTRL             (USBOTG_PHY_BASE+0x04)
+#define S3C_OTG_RSTCON              (USBOTG_PHY_BASE+0x08)
 #include "s3c64x0.h"

 static inline s3c64xx_uart *s3c64xx_get_base_uart(enum s3c64xx_uarts_nr nr)
```

Step 4,Include the object file to make.

```
---------------------------- common/Makefile -------------------------------
index 973f05a..3a9e66c 100644
@@ -167,6 +167,7 @@ ifdef CONFIG_CMD_USB
 COBJS-y += cmd_usb.o
 COBJS-y += usb.o usb_hub.o
 COBJS-$(CONFIG_USB_STORAGE) += usb_storage.o
+COBJS-$(CONFIG_S3C_USBD) += cmd_usbd.o
 endif
 COBJS-$(CONFIG_CMD_XIMG) += cmd_ximg.o
 COBJS-$(CONFIG_YAFFS2) += cmd_yaffs2.o

---------------------------- drivers/usb/host/Makefile --------------------------
index bcb4662..a1fe639 100644
@@ -26,6 +26,7 @@ include $(TOPDIR)/config.mk
 LIB   := $(obj)libusb_host.o

 # ohci
+COBJS-$(CONFIG_S3C_USBD) += s3c6410-usbd-otg.o
 COBJS-$(CONFIG_USB_OHCI_NEW) += ohci-hcd.o
 COBJS-$(CONFIG_USB_ATMEL) += ohci-at91.o
 COBJS-$(CONFIG_USB_OHCI_DA8XX) += ohci-da8xx.o
```

---------- include/configs/smdk6410.h ----------
index d4dc707..3eb5b11 100644
@@ -306,6 +306,7 @@
 #define CONFIG_SYS_USB_OHCI_CPU_INIT		1

 #define CONFIG_USB_STORAGE 1
+#define CONFIG_S3C_USBD
 #endif
 #define CONFIG_DOS_PARTITION	1

Lastly,Testing..

U-Boot 2012.10-g3141653-dirty (Mar 30 2013 - 01:05:20) for SMDK6410

CPU: S3C6400@533MHz
        Fclk = 533MHz, Hclk = 133MHz, Pclk = 66MHz (ASYNC Mode)
Board: SMDK6410
DRAM: 256 MiB
Flash: *** failed ***
NAND: USE HWECC 8BIT
ECC Size:512 ECC Bytes:13
4096 MiB
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
Net: dm9000
Hit any key to stop autoboot: 0
SMDK6410 # dnw 50008000
Insert a OTG cable into the connector!
OTG cable Connected!
Now, Waiting for DNW to transmit data

```
Download Done!! Download Address: 0x50008000, Download Filesize:0x4015c
Checksum is being calculated.
Checksum Value => MEM:3f27 DNW:0
Checksum failed.

SMDK6410 #
```

Chapter 5,MMC driver

Step 1,smdk6410.h

```
------------------------- include/configs/smdk6410.h -------------------------
index 4fa1c81..d103e4a 100644
@@ -41,7 +41,7 @@
  * MMC Support
  */
 #define CONFIG_GENERIC_MMC          1
-#define CONFIG_MMC          1
+#define CONFIG_MMC                  1
 #define CONFIG_S3C64X0_MMC  1
 #define CONFIG_CMD_MMC          /* MMC support  */
```

Step 2,s3c6410.h

```
---------------- arch/arm/include/asm/arch-s3c64xx/s3c6410.h ----------------
index a65ebd0..ea3935f 100644
@@ -111,17 +111,17 @@
 #define AHB_CON1_REG            __REG(ELFIN_CLOCK_POWER_BASE +
AHB_CON1_OFFSET)
 #define AHB_CON2_REG            __REG(ELFIN_CLOCK_POWER_BASE +
AHB_CON2_OFFSET)
 #define SELECT_DMA_REG          __REG(ELFIN_CLOCK_POWER_BASE + \
-                                      SELECT_DMA_OFFSET)
+               SELECT_DMA_OFFSET)
 #define SW_RST_REG          __REG(ELFIN_CLOCK_POWER_BASE +
SW_RST_OFFSET)
```

```
 #define SYS_ID_REG          __REG(ELFIN_CLOCK_POWER_BASE +
SYS_ID_OFFSET)
 #define MEM_SYS_CFG_REG             __REG(ELFIN_CLOCK_POWER_BASE
+ \
-                               MEM_SYS_CFG_OFFSET)
+               MEM_SYS_CFG_OFFSET)
 #define QOS_OVERRIDE0_REG   __REG(ELFIN_CLOCK_POWER_BASE + \
-                               QOS_OVERRIDE0_OFFSET)
+               QOS_OVERRIDE0_OFFSET)
 #define QOS_OVERRIDE1_REG   __REG(ELFIN_CLOCK_POWER_BASE + \
-                               QOS_OVERRIDE1_OFFSET)
+               QOS_OVERRIDE1_OFFSET)
 #define MEM_CFG_STAT_REG    __REG(ELFIN_CLOCK_POWER_BASE + \
-                               MEM_CFG_STAT_OFFSET)
+               MEM_CFG_STAT_OFFSET)
 #define PWR_CFG_REG             __REG(ELFIN_CLOCK_POWER_BASE +
PWR_CFG_OFFSET)
 #define EINT_MASK_REG           __REG(ELFIN_CLOCK_POWER_BASE +
EINT_MASK_OFFSET)
 #define NOR_CFG_REG             __REG(ELFIN_CLOCK_POWER_BASE +
NOR_CFG_OFFSET)
@@ -129,19 +129,19 @@
 #define SLEEP_CFG_REG           __REG(ELFIN_CLOCK_POWER_BASE +
SLEEP_CFG_OFFSET)
 #define OSC_FREQ_REG            __REG(ELFIN_CLOCK_POWER_BASE +
OSC_FREQ_OFFSET)
 #define OSC_CNT_VAL_REG             __REG(ELFIN_CLOCK_POWER_BASE
+ \
-                               OSC_CNT_VAL_OFFSET)
+               OSC_CNT_VAL_OFFSET)
 #define PWR_CNT_VAL_REG             __REG(ELFIN_CLOCK_POWER_BASE
+ \
-                               PWR_CNT_VAL_OFFSET)
+               PWR_CNT_VAL_OFFSET)
 #define FPC_CNT_VAL_REG             __REG(ELFIN_CLOCK_POWER_BASE
+ \
-                               FPC_CNT_VAL_OFFSET)
```

```
+               FPC_CNT_VAL_OFFSET)
 #define MTC_CNT_VAL_REG                __REG(ELFIN_CLOCK_POWER_BASE
+ \
-                               MTC_CNT_VAL_OFFSET)
+               MTC_CNT_VAL_OFFSET)
 #define OTHERS_REG             __REG(ELFIN_CLOCK_POWER_BASE +
OTHERS_OFFSET)
 #define RST_STAT_REG           __REG(ELFIN_CLOCK_POWER_BASE +
RST_STAT_OFFSET)
 #define WAKEUP_STAT_REG                __REG(ELFIN_CLOCK_POWER_BASE
+ \
-                               WAKEUP_STAT_OFFSET)
+               WAKEUP_STAT_OFFSET)
 #define BLK_PWR_STAT_REG       __REG(ELFIN_CLOCK_POWER_BASE + \
-                               BLK_PWR_STAT_OFFSET)
+               BLK_PWR_STAT_OFFSET)
 #define INF_REG0_REG           __REG(ELFIN_CLOCK_POWER_BASE +
INF_REG0_OFFSET)
 #define INF_REG1_REG           __REG(ELFIN_CLOCK_POWER_BASE +
INF_REG1_OFFSET)
 #define INF_REG2_REG           __REG(ELFIN_CLOCK_POWER_BASE +
INF_REG2_OFFSET)
@@ -811,26 +811,26 @@
 #define STARTUP_APLLDIV        0


 #define CLK_DIV_VAL((STARTUP_PCLKDIV << 12) | (STARTUP_HCLKX2DIV
<< 9) | \
-       (STARTUP_HCLKDIV << 8) | (STARTUP_MPLLDIV<<4) |
STARTUP_APLLDIV)
+               (STARTUP_HCLKDIV << 8) | (STARTUP_MPLLDIV<<4) |
STARTUP_APLLDIV)
 #define MPLL_VAL     ((1 << 31) | (STARTUP_MDIV << 16) | \
-       (STARTUP_PDIV << 8) | STARTUP_SDIV)
+               (STARTUP_PDIV << 8) | STARTUP_SDIV)
 #define STARTUP_MPLL     (((CONFIG_SYS_CLK_FREQ >> STARTUP_SDIV) / \
-       STARTUP_PDIV) * STARTUP_MDIV)
+                       STARTUP_PDIV) * STARTUP_MDIV)
```

```
 #if defined(CONFIG_SYNC_MODE)
 #define APLL_VAL      ((1 << 31) | (STARTUP_MDIV << 16) | \
-       (STARTUP_PDIV << 8) | STARTUP_SDIV)
+               (STARTUP_PDIV << 8) | STARTUP_SDIV)
 #define STARTUP_APLL      (((CONFIG_SYS_CLK_FREQ >> STARTUP_SDIV) / \
-       STARTUP_PDIV) * STARTUP_MDIV)
+                       STARTUP_PDIV) * STARTUP_MDIV)
 #define STARTUP_HCLK     (STARTUP_MPLL / (STARTUP_HCLKX2DIV + 1) / \
-       (STARTUP_HCLKDIV + 1))
+               (STARTUP_HCLKDIV + 1))
 #else
 #define APLL_VAL      ((1 << 31) | (STARTUP_AMDIV << 16) | \
-       (STARTUP_PDIV << 8) | STARTUP_SDIV)
+               (STARTUP_PDIV << 8) | STARTUP_SDIV)
 #define STARTUP_APLL      (((CONFIG_SYS_CLK_FREQ >> STARTUP_SDIV) / \
-       STARTUP_PDIV) * STARTUP_AMDIV)
+                       STARTUP_PDIV) * STARTUP_AMDIV)
 #define STARTUP_HCLK     (STARTUP_MPLL / (STARTUP_HCLKX2DIV + 1) / \
-       (STARTUP_HCLKDIV + 1))
+               (STARTUP_HCLKDIV + 1))
 #endif


@@ -994,7 +994,19 @@ enum s3c64xx_uarts_nr {
 /* Endpoint FIFO address */
 #define S3C_OTG_EP0_FIFO       (USBOTG_LINK_BASE + 0x1000)

+/* S3C6400 device base addresses */
+#define ELFIN_DMA_BASE                 0x75000000
+#define ELFIN_LCD_BASE         0x77100000
+#define ELFIN_USB_HOST_BASE        0x74300000
+#define ELFIN_I2C_BASE         0x7f004000
+#define ELFIN_I2S_BASE         0x7f002000
```

```
+#define ELFIN_ADC_BASE         0x7e00b000
+#define ELFIN_SPI_BASE         0x7f00b000
+#define ELFIN_HSMMC_0_BASE        0x7c200000
+#define ELFIN_HSMMC_1_BASE        0x7c300000
+#define ELFIN_HSMMC_2_BASE        0x7c400000

+#define ELFIN_CLOCK_POWER_BASE0x7e00f000

 /* OTG PHY CORE REGISTERS */
 #define S3C_OTG_PHYPWR         (USBOTG_PHY_BASE+0x00)
@@ -1006,5 +1018,17 @@ static inline s3c64xx_uart
*s3c64xx_get_base_uart(enum s3c64xx_uarts_nr nr)
 {
        return (s3c64xx_uart *)(ELFIN_UART_BASE + (nr * 0x400));
 }
+static inline void *samsung_get_base_mmc(void)
+{
+  return (void *)(ELFIN_HSMMC_0_BASE);
+}
+
+struct mmc_host {
+       struct s3c64x0_mmc *reg;
+       unsigned int version;    /* SDHCI spec. version */
+       unsigned int clock;      /* Current clock (MHz) */
+};
+
+int s3c64x0_mmc_init(int dev_index);
 #endif
 #endif /*__S3C6410_H__*/
```

Step 3,s3c64x0.h

```
---------- arch/arm/include/asm/arch-s3c64xx/s3c64x0.h ----------
index 7c1467c..0dc8a4a 100644
@@ -86,4 +86,40 @@ typedef struct {
```

```
        volatile u32        TCNTB4;
        volatile u32        TCNTO4;
 } s3c64xx_timers;
+struct s3c64x0_mmc {
+       unsigned int        sysad;
+       unsigned short      blksize;
+       unsigned short      blkcnt;
+       unsigned int        argument;
+       unsigned short      trnmod;
+       unsigned short      cmdreg;
+       unsigned int        rspreg0;
+       unsigned int        rspreg1;
+       unsigned int        rspreg2;
+       unsigned int        rspreg3;
+       unsigned int        bdata;
+       unsigned int        prnsts;
+       unsigned char       hostctl;
+       unsigned char       pwrcon;
+       unsigned char       blkgap;
+       unsigned char       wakcon;
+       unsigned short      clkcon;
+       unsigned char       timeoutcon;
+       unsigned char       swrst;
+       unsigned int        norintsts;    /* errintsts */
+       unsigned int        norintstsen;        /* errintstsen */
+       unsigned int        norintsigen;        /* errintsigen */
+       unsigned short      acmd12errsts;
+       unsigned char       res1[2];
+       unsigned int        capareg;
+       unsigned char       res2[4];
+       unsigned int        maxcurr;
+       unsigned char       res3[0x34];
+       unsigned int        control2;
+       unsigned int        control3;
```

```
+       unsigned int        control4;
+       unsigned char       res4[0x6e];
+       unsigned short      hcver;
+       unsigned char       res5[0xFFF02];
+};
 #endif /*__S3C64XX_H__*/
```

step 4,Makefile & smdk6410.c

```
*/

---------------------- board/samsung/smdk6410/smdk6410.c ----------------------
index 8cb1f79..45da7e9 100644
@@ -136,4 +136,11 @@ int board_eth_init(bd_t *bis)
 }
 #endif

+#ifdef CONFIG_GERNERIC_MMC
+int board_mmc_init(bd_t *bis)
+{
+       return s3c64x0_mmc_init(0);
+}
+#endif
+



----------------------------- drivers/mmc/Makefile -----------------------------
index 565ba6a..9f7ace9 100644
@@ -47,6 +47,7 @@ COBJS-$(CONFIG_SDHCI) += sdhci.o
 COBJS-$(CONFIG_S5P_SDHCI) += s5p_sdhci.o
 COBJS-$(CONFIG_SH_MMCIF) += sh_mmcif.o
 COBJS-$(CONFIG_TEGRA_MMC) += tegra_mmc.o
+COBJS-$(CONFIG_S3C64X0_MMC) += s3c64x0_mmc.o

 COBJS       := $(COBJS-y)
```

```
SRCS        := $(COBJS:.o=.c)
```

---

step 5,s3c64x0_mmc.c

```
---------- drivers/mmc/s3c64x0_mmc.c ----------
new file mode 100644
index 0000000..bc7a614
@@ -0,0 +1,485 @@
+/*
+ * (C) Copyright 2009 SAMSUNG Electronics
+ * Minkyu Kang <mk7.kang@samsung.com>
+ * Jaehoon Chung <jh80.chung@samsung.com>
+ *
+ * modified by Tekkaman Ninja for s3c64x0 <tekkamanninja@gmail.com>
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
+ * GNU General Public License for more details.
+ *
+ * You should have received a copy of the GNU General Public License
+ * along with this program; if not, write to the Free Software
+ * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
+ */
+//#define DEBUG
+#include <common.h>
+//#undef DEBUG
+#include <mmc.h>
```

```
+#include <asm/io.h>
+//#include <asm/arch/mmc.h>
+#include <asm/arch/s3c64x0.h>
+#include <asm/arch/s3c6410.h>
+
+/* support 3 mmc hosts */
+struct mmc mmc_dev[3];
+struct mmc_host mmc_host[3];
+
+static inline struct s3c64x0_mmc *s3c64x0_get_base_mmc(int dev_index)
+{
+	unsigned long offset = dev_index * sizeof(struct s3c64x0_mmc);
+	return (struct s3c64x0_mmc *)(samsung_get_base_mmc() + offset);
+}
+
+static int mmc_card_detect(struct mmc_host *mmc_host)
+{
+	return (mmc_host->reg->prnsts & (0x5 << 16));
+}
+
+static void mmc_prepare_data(struct mmc_host *host, struct mmc_data *data)
+{
+	unsigned char ctrl;
+
+	debug("data->dest: %08x\n", (u32)data->dest);
+	writel((u32)data->dest, &host->reg->sysad);
+	/*
+	 * DMASEL[4:3]
+	 * 00 = Selects SDMA
+	 * 01 = Reserved
+	 * 10 = Selects 32-bit Address ADMA2
+	 * 11 = Selects 64-bit Address ADMA2
+	 */
+	ctrl = readb(&host->reg->hostctl);
```

```
+	ctrl &= ~(3 << 3);
+	writeb(ctrl, &host->reg->hostctl);
+
+	/* We do not handle DMA boundaries, so set it to max (512 KiB) */
+	writew((7 << 12) | (data->blocksize << 0), &host->reg->blksize);
+	writew(data->blocks, &host->reg->blkcnt);
+}
+
+static void mmc_set_transfer_mode(struct mmc_host *host, struct mmc_data
*data)
+{
+	unsigned short mode;
+
+	/*
+	 * TRNMOD
+	 * MUL1SIN0[5]  : Multi/Single Block Select
+	 * RD1WT0[4]    : Data Transfer Direction Select
+	 *	1 = read
+	 *	0 = write
+	 * ENACMD12[2]: Auto CMD12 Enable
+	 * ENBLKCNT[1] : Block Count Enable
+	 * ENDMA[0]     : DMA Enable
+	 */
+	mode = (1 << 1) | (1 << 0);
+	if (data->blocks > 1)
+		mode |= (1 << 5);
+	if (data->flags & MMC_DATA_READ)
+		mode |= (1 << 4);
+
+	writew(mode, &host->reg->trnmod);
+}
+
+static int mmc_send_cmd(struct mmc *mmc, struct mmc_cmd *cmd,
+			struct mmc_data *data)
```

```
+{
+	struct mmc_host *host = (struct mmc_host *)mmc->priv;
+	int flags, i;
+	unsigned int timeout;
+	unsigned int mask;
+	unsigned int retry = 0x100000;
+
+	/* Wait max 10 ms */
+	timeout = 10;
+
+	/*
+	 * PRNSTS
+	 * CMDINHDAT[1]		: Command Inhibit (DAT)
+	 * CMDINHCMD[0]		: Command Inhibit (CMD)
+	 */
+	mask = (1 << 0);
+	if ((data != NULL) || (cmd->resp_type & MMC_RSP_BUSY))
+		mask |= (1 << 1);
+
+	/*
+	 * We shouldn't wait for data inihibit for stop commands, even
+	 * though they might use busy signaling
+	 */
+	if (data)
+		mask &= ~(1 << 1);
+
+	while (readl(&host->reg->prnsts) & mask) {
+		if (timeout == 0) {
+			printf("%s: timeout error\n", __func__);
+			return -1;
+		}
+		timeout--;
+		udelay(1000);
+	}
```

```
+
+	if (data)
+		mmc_prepare_data(host, data);
+
+	debug("cmd->arg: %08x\n", cmd->cmdarg);
+	writel(cmd->cmdarg, &host->reg->argument);
+
+	if (data)
+		mmc_set_transfer_mode(host, data);
+
+	if ((cmd->resp_type & MMC_RSP_136) && (cmd->resp_type &
MMC_RSP_BUSY))
+		return -1;
+
+	/*
+	 * CMDREG
+	 * CMDIDX[13:8]: Command index
+	 * DATAPRNT[5]  : Data Present Select
+	 * ENCMDIDX[4]: Command Index Check Enable
+	 * ENCMDCRC[3]	: Command CRC Check Enable
+	 * RSPTYP[1:0]
+	 *	00 = No Response
+	 *	01 = Length 136
+	 *	10 = Length 48
+	 *	11 = Length 48 Check busy after response
+	 */
+	if (!(cmd->resp_type & MMC_RSP_PRESENT))
+		flags = 0;
+	else if (cmd->resp_type & MMC_RSP_136)
+		flags = (1 << 0);
+	else if (cmd->resp_type & MMC_RSP_BUSY)
+		flags = (3 << 0);
+	else
+		flags = (2 << 0);
```

```
+
+	if (cmd->resp_type & MMC_RSP_CRC)
+		flags |= (1 << 3);
+	if (cmd->resp_type & MMC_RSP_OPCODE)
+		flags |= (1 << 4);
+	if (data)
+		flags |= (1 << 5);
+
+	debug("cmd: %d\n", cmd->cmdidx);
+
+	writew((cmd->cmdidx << 8) | flags, &host->reg->cmdreg);
+
+	for (i = 0; i < retry; i++) {
+		mask = readl(&host->reg->norintsts);
+		/* Command Complete */
+		if (mask & (1 << 0)) {
+			if (!data)
+				writel(mask, &host->reg->norintsts);
+			break;
+		}
+	}
+
+	if (i == retry) {
+		printf("%s: waiting for status update\n", __func__);
+		return TIMEOUT;
+	}
+
+	if (mask & (1 << 16)) {
+		/* Timeout Error */
+		debug("timeout: %08x cmd %d\n", mask, cmd->cmdidx);
+		return TIMEOUT;
+	} else if (mask & (1 << 15)) {
+		/* Error Interrupt */
+		debug("error: %08x cmd %d\n", mask, cmd->cmdidx);
```

```
+		return -1;
+	}
+
+	if (cmd->resp_type & MMC_RSP_PRESENT) {
+		if (cmd->resp_type & MMC_RSP_136) {
+			/* CRC is stripped so we need to do some shifting. */
+			for (i = 0; i < 4; i++) {
+				unsigned int offset =
+					(unsigned int)(&host->reg->rspreg3 - i);
+				cmd->response[i] = readl(offset) << 8;
+
+				if (i != 3) {
+					cmd->response[i] |=
+						readb(offset - 1);
+				}
+				debug("cmd->resp[%d]: %08x\n",
+					i, cmd->response[i]);
+			}
+		} else if (cmd->resp_type & MMC_RSP_BUSY) {
+			for (i = 0; i < retry; i++) {
+				/* PRNTDATA[23:20] : DAT[3:0] Line Signal */
+				if (readl(&host->reg->prnsts)
+					& (1 << 20))	/* DAT[0] */
+					break;
+			}
+
+			if (i == retry) {
+				printf("%s: card is still busy\n", __func__);
+				return TIMEOUT;
+			}
+
+			cmd->response[0] = readl(&host->reg->rspreg0);
+			debug("cmd->resp[0]: %08x\n", cmd->response[0]);
+		} else {
```

```
+                       cmd->response[0] = readl(&host->reg->rspreg0);
+                       debug("cmd->resp[0]: %08x\n", cmd->response[0]);
+               }
+       }
+
+       if (data) {
+               while (1) {
+                       mask = readl(&host->reg->norintsts);
+
+                       if (mask & (1 << 15)) {
+                               /* Error Interrupt */
+                               writel(mask, &host->reg->norintsts);
+                               printf("%s: error during transfer: 0x%08x\n",
+                                               __func__, mask);
+                               return -1;
+                       } else if (mask & (1 << 3)) {
+                               /* DMA Interrupt */
+                               debug("DMA end\n");
+                               break;
+                       } else if (mask & (1 << 1)) {
+                               /* Transfer Complete */
+                               debug("r/w is done\n");
+                               break;
+                       }
+               }
+               writel(mask, &host->reg->norintsts);
+       }
+
+       udelay(1000);
+       return 0;
+}
+
+static void mmc_change_clock(struct mmc_host *host, uint clock)
+{
```

```
+	int div;
+	unsigned short clk;
+	unsigned long timeout;
+	unsigned long ctrl2;
+
+	/*
+	 * SELBASECLK[5:4]
+	 * 00/01 = HCLK
+	 * 10 = EPLL
+	 * 11 = XTI or XEXTCLK
+	 */
+	ctrl2 = readl(&host->reg->control2);
+	ctrl2 &= ~(3 << 4);
+	ctrl2 |= (2 << 4);
+	writel(ctrl2, &host->reg->control2);
+
+	writew(0, &host->reg->clkcon);
+
+	/* XXX: we assume that clock is between 40MHz and 50MHz */
+	if (clock == 0)
+		goto out;
+	else if (clock <= 400000)
+		div = 0x100;
+	else if (clock <= 20000000)
+		div = 4;
+	else if (clock <= 26000000)
+		div = 2;
+	else
+		div = 1;
+	debug("div: %d\n", div);
+
+	div >>= 1;
+	/*
+	 * CLKCON
```

```
+	 * SELFREQ[15:8]	: base clock divied by value
+	 * ENSDCLK[2]		: SD Clock Enable
+	 * STBLINTCLK[1]	: Internal Clock Stable
+	 * ENINTCLK[0]		: Internal Clock Enable
+	 */
+	clk = (div << 8) | (1 << 0);
+	writew(clk, &host->reg->clkcon);
+
+	/* Wait max 10 ms */
+	timeout = 10;
+	while (!(readw(&host->reg->clkcon) & (1 << 1))) {
+		if (timeout == 0) {
+			printf("%s: timeout error\n", __func__);
+			return;
+		}
+		timeout--;
+		udelay(1000);
+	}
+
+	clk |= (1 << 2);
+	writew(clk, &host->reg->clkcon);
+
+out:
+	host->clock = clock;
+}
+
+static void mmc_set_ios(struct mmc *mmc)
+{
+	struct mmc_host *host = mmc->priv;
+	unsigned char ctrl;
+	unsigned long val;
+
+	debug("bus_width: %x, clock: %d\n", mmc->bus_width, mmc->clock);
+
```

```
+	/*
+	 * SELCLKPADDS[17:16]
+	 * 00 = 2mA
+	 * 01 = 4mA
+	 * 10 = 7mA
+	 * 11 = 9mA
+	 */
+	writel(0x3 << 16, &host->reg->control4);
+
+	val = readl(&host->reg->control2);
+	val &= (0x3 << 4);
+
+	val |=		(1 << 31) | /* write status clear async mode enable */
+			(1 << 30) | /* command conflict mask enable */
+			(1 << 14) | /* Feedback Clock Enable for Rx Clock */
+			(1 << 8);    /* SDCLK hold enable */
+
+	writel(val, &host->reg->control2);
+
+	/*
+	 * FCSEL1[15] FCSEL0[7]
+	 * FCSel[1:0] : Rx Feedback Clock Delay Control
+	 *	Inverter delay means10ns delay if SDCLK 50MHz setting
+	 *	01 = Delay1 (basic delay)
+	 *	11 = Delay2 (basic delay + 2ns)
+	 *	00 = Delay3 (inverter delay)
+	 *	10 = Delay4 (inverter delay + 2ns)
+	 */
+	writel(0x8080, &host->reg->control3);
+
+	mmc_change_clock(host, mmc->clock);
+
+	ctrl = readb(&host->reg->hostctl);
+
```

```
+	/*
+	 * WIDE4[1]
+	 * 1 = 4-bit mode
+	 * 0 = 1-bit mode
+	 */
+	if (mmc->bus_width == 4)
+		ctrl |= (1 << 1);
+	else
+		ctrl &= ~(1 << 1);
+
+	/*
+	 * OUTEDGEINV[2]
+	 * 1 = Riging edge output
+	 * 0 = Falling edge output
+	 */
+	ctrl &= ~(1 << 2);
+
+	writeb(ctrl, &host->reg->hostctl);
+}
+
+static void mmc_reset(struct mmc_host *host)
+{
+	unsigned int timeout;
+
+	/*
+	 * RSTALL[0] : Software reset for all
+	 * 1 = reset
+	 * 0 = work
+	 */
+	writeb((1 << 0), &host->reg->swrst);
+
+	host->clock = 0;
+
+	/* Wait max 100 ms */
```

```
+	timeout = 100;
+
+	/* hw clears the bit when it's done */
+	while (readb(&host->reg->swrst) & (1 << 0)) {
+		if (timeout == 0) {
+			printf("%s: timeout error\n", __func__);
+			return;
+		}
+		timeout--;
+		udelay(1000);
+	}
+}
+
+static int mmc_core_init(struct mmc *mmc)
+{
+	struct mmc_host *host = (struct mmc_host *)mmc->priv;
+	unsigned int mask;
+#if 0
+	if (mmc_card_detect(host) == 0) {
+		printf("NO SD/MMC detected!\n");
+		return -1;
+	}
+#endif
+	mmc_reset(host);
+
+	host->version = readw(&host->reg->hcver);
+
+	/* mask all */
+	writel(0xffffffff, &host->reg->norintstsen);
+	writel(0xffffffff, &host->reg->norintsigen);
+
+	writeb(0xe, &host->reg->timeoutcon);	/* TMCLK * 2^27 */
+
+	/*
```

```
+	 * NORMAL Interrupt Status Enable Register init
+	 * [5] ENSTABUFRDRDY : Buffer Read Ready Status Enable
+	 * [4] ENSTABUFWTRDY : Buffer write Ready Status Enable
+	 * [1] ENSTASTANSCMPLT : Transfre Complete Status Enable
+	 * [0] ENSTACMDCMPLT : Command Complete Status Enable
+	*/
+	mask = readl(&host->reg->norintstsen);
+	mask &= ~(0xffff);
+	mask |= (1 << 5) | (1 << 4) | (1 << 1) | (1 << 0);
+	writel(mask, &host->reg->norintstsen);
+
+	/*
+	 * NORMAL Interrupt Signal Enable Register init
+	 * [1] ENSTACMDCMPLT : Transfer Complete Signal Enable
+	 */
+	mask = readl(&host->reg->norintsigen);
+	mask &= ~(0xffff);
+	mask |= (1 << 1);
+	writel(mask, &host->reg->norintsigen);
+
+	return 0;
+}
+
+static int s3c64x0_mmc_initialize(int dev_index)
+{
+	struct mmc *mmc;
+
+	mmc = &mmc_dev[dev_index];
+
+	sprintf(mmc->name, "SAMSUNG SD/MMC");
+	mmc->priv = &mmc_host[dev_index];
+	mmc->send_cmd = mmc_send_cmd;
+	mmc->set_ios = mmc_set_ios;
+	mmc->init = mmc_core_init;
```

```
+	mmc->detect_mmc = mmc_card_detect;
+
+	mmc->voltages = MMC_VDD_32_33 | MMC_VDD_33_34 |
MMC_VDD_165_195;
+	mmc->host_caps = MMC_MODE_4BIT | MMC_MODE_HS_52MHz |
MMC_MODE_HS;
+
+	mmc->f_min = 400000;
+	mmc->f_max = 52000000;
+
+	mmc_host[dev_index].clock = 0;
+	mmc_host[dev_index].reg = s3c64x0_get_base_mmc(dev_index);
+
+	mmc_register(mmc);
+
+	return 0;
+}
+
+int s3c64x0_mmc_init(int dev_index)
+{
+	return s3c64x0_mmc_initialize(dev_index);
+}
```

Chapter 6 ,lcd driver

Step 1.Nand boot problem.

I found that my u-boot no response when boot from Nand.

As we kown,u-boot-nand.bin contains u-boot-spl-16k.bin and u-boot.bin.

You can do it like this:

```
cat nand-spl/u-boot-spl-16k.bin u-boot.bin > u-boot-nand.bin
```

```
---------- board/samsung/smdk6410/smdk6410_nand_spl.c ----------
index a023284..fa671d9 100644
@@ -32,6 +32,9 @@

 void board_init_f(unsigned long bootflag)
```

```
 {
-       relocate_code(CONFIG_SYS_TEXT_BASE - TOTAL_MALLOC_LEN, NULL,
+       /*relocate_code(CONFIG_SYS_TEXT_BASE - TOTAL_MALLOC_LEN, NULL,
+                       CONFIG_SYS_TEXT_BASE);
+                       */
+       relocate_code(8 * 1024, NULL,
                        CONFIG_SYS_TEXT_BASE);
 }

---------- include/configs/smdk6410.h ----------
index d103e4a..cae9f34 100644
@@ -254,8 +254,8 @@
 #define CONFIG_SYS_NAND_U_BOOT_DST   CONFIG_SYS_PHY_UBOOT_BASE
        /* NUB load-addr      */
 #define CONFIG_SYS_NAND_U_BOOT_START
        CONFIG_SYS_NAND_U_BOOT_DST       /* NUB start-addr     */

-#define CONFIG_SYS_NAND_U_BOOT_OFFS(4 * 1024)   /* Offset to RAM U-
Boot image */
-#define CONFIG_SYS_NAND_U_BOOT_SIZE (252 * 1024)       /* Size of RAM U-
Boot image   */
+#define CONFIG_SYS_NAND_U_BOOT_OFFS       (4 * 4 * 1024)      /* Offset to
RAM U-Boot image */
+#define CONFIG_SYS_NAND_U_BOOT_SIZE       (496 * 1024)       /* Size of
RAM U-Boot image   */

 /* NAND chip page size                 */
 #define CONFIG_SYS_NAND_PAGE_SIZE      (2048 * 2)

---------- nand_spl/board/samsung/smdk6410/config.mk ----------
index 8bea498..da144c5 100644
@@ -33,7 +33,7 @@ include $(TOPDIR)/board/$(BOARDDIR)/config.mk

 # PAD_TO used to generate a 4kByte binary needed for the combined image
 # -> PAD_TO = CONFIG_SYS_TEXT_BASE + 4096
-PAD_TO     := $(shell expr $$[$(CONFIG_SYS_TEXT_BASE) + 4096])
```

```
+PAD_TO   := $(shell expr $$[$(CONFIG_SYS_TEXT_BASE) + 8192])

 ifeq ($(debug),1)
 PLATFORM_CPPFLAGS += -DDEBUG
```

Step 2.Modified this file.

chapter 6,lcd driver

```
---------- arch/arm/include/asm/arch-s3c64xx/s3c6410.h ----------
index ea3935f..06d14bb 100644
@@ -256,6 +256,7 @@
 #define GPJPUD_OFFSET           0x128
 #define GPJCONSLP_OFFSET        0x12C
 #define GPJPUDSLP_OFFSET        0x130
+#define SPCON_OFFSET            0x1A0
 #define MEM0DRVCON_OFFSET       0x1D0
 #define MEM1DRVCON_OFFSET       0x1D4
 #define GPKCON0_OFFSET          0x800

---------- common/lcd.c ----------
index b6be800..63ddbb4 100644
@@ -1,26 +1,6 @@
 /*
  * Common LCD routines for supported CPUs
  *
- * (C) Copyright 2001-2002
- * Wolfgang Denk, DENX Software Engineering -- wd@denx.de
- *
- * See file CREDITS for list of people who contributed to this
- * project.
- *
- * This program is free software; you can redistribute it and/or
- * modify it under the terms of the GNU General Public License as
```

```
- * published by the Free Software Foundation; either version 2 of
- * the License, or (at your option) any later version.
- *
- * This program is distributed in the hope that it will be useful,
- * but WITHOUT ANY WARRANTY; without even the implied warranty of
- * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
- * GNU General Public License for more details.
- *
- * You should have received a copy of the GNU General Public License
- * along with this program; if not, write to the Free Software
- * Foundation, Inc., 59 Temple Place, Suite 330, Boston,
- * MA 02111-1307 USA
  */

 /*****************************************************************************/
@@ -32,6 +12,7 @@
 #include <config.h>
 #include <common.h>
 #include <command.h>
+#include <version.h>
 #include <stdarg.h>
 #include <linux/types.h>
 #include <stdio_dev.h>
@@ -40,27 +21,14 @@
 #endif
 #include <lcd.h>
 #include <watchdog.h>
-
-#if defined(CONFIG_CPU_PXA25X) || defined(CONFIG_CPU_PXA27X) || \
-        defined(CONFIG_CPU_MONAHANS)
-#define CONFIG_CPU_PXA
-#include <asm/byteorder.h>
-#endif
-
```

```
-#if defined(CONFIG_MPC823)
-#include <lcdvideo.h>
-#endif
-
-#if defined(CONFIG_ATMEL_LCD)
-#include <atmel_lcdc.h>
-#endif
+#include <asm/io.h>
+#ifdef CONFIG_LCD

 /************************************************************************/
 /* ** FONT DATA                                                         */
 /************************************************************************/
 #include <video_font.h>              /* Get font data, width and height  */
 #include <video_font_data.h>
-
 /************************************************************************/
 /* ** LOGO DATA                                                         */
 /************************************************************************/
@@ -72,46 +40,61 @@
 # endif
 #endif

-DECLARE_GLOBAL_DATA_PTR;

 ulong lcd_setmem (ulong addr);

-static void lcd_drawchars(ushort x, ushort y, uchar *str, int count);
-static inline void lcd_puts_xy(ushort x, ushort y, uchar *s);
-static inline void lcd_putc_xy(ushort x, ushort y, uchar  c);
+DECLARE_GLOBAL_DATA_PTR;

-static int lcd_init(void *lcdbase);
+vidinfo_t panel_info = {
```

```
+  S3CFB_HRES,  //行点数 320
+  S3CFB_VRES,  //行数  240
+          0,  //显示宽度
+          0,  //显示高度
+   4,//PIXELBITS, //每像素点使用位数 16bpp
+};

-static void *lcd_logo (void);
+char lcd_is_enabled = 0;
+int lcd_line_length;
+int lcd_color_fg;
+int lcd_color_bg;

-static int lcd_getbgcolor(void);
-static void lcd_setfgcolor(int color);
-static void lcd_setbgcolor(int color);
+void *lcd_base;          /* Start of framebuffer memory      */
+void *lcd_console_address; /* Start of console buffer       */

-char lcd_is_enabled = 0;
+/*
+ * Frame buffer memory information
+ */

-#ifdef       NOT_USED_SO_FAR
-static void lcd_getcolreg(ushort regno,
-                          ushort *red, ushort *green, ushort *blue);
-static int lcd_getfgcolor(void);
-#endif       /* NOT_USED_SO_FAR */

-/************************************************************************/
+static int lcd_init (void *lcdbase);
```

```
-/*---------------------------------------------------------------------*/
+extern ulong get_HCLK(void);
+static void lcd_hline(int x,int y,int width,uint16_t color);
+static void lcd_vline(int x,int y,int height,uint16_t color);
+
+static void lcd_drawchars (ushort x, ushort y, uchar *str, int count);
+static inline void lcd_puts_xy (ushort x, ushort y, uchar *s);
+static inline void lcd_putc_xy (ushort x, ushort y, uchar  c);
+static int lcd_getbgcolor (void);
+static void lcd_setfgcolor (int color);
+static void lcd_setbgcolor (int color);
+static void *lcd_logo (void);
+
+short console_col;
+short console_row;

-static void console_scrollup(void)
+static void console_scrollup (void)
 {
        /* Copy up rows ignoring the first one */
-       memcpy(CONSOLE_ROW_FIRST, CONSOLE_ROW_SECOND,
CONSOLE_SCROLL_SIZE);
+       memcpy (CONSOLE_ROW_FIRST, CONSOLE_ROW_SECOND,
CONSOLE_SCROLL_SIZE);


        /* Clear the last one */
-       memset(CONSOLE_ROW_LAST, COLOR_MASK(lcd_color_bg),
CONSOLE_ROW_SIZE);
+       memset (CONSOLE_ROW_LAST, COLOR_MASK(lcd_color_bg),
CONSOLE_ROW_SIZE);
 }

 /*---------------------------------------------------------------------*/


-static inline void console_back(void)
```

```
+static inline void console_back (void)
 {
        if (--console_col < 0) {
                console_col = CONSOLE_COLS-1 ;
@@ -120,13 +103,14 @@ static inline void console_back(void)
                }
        }

-       lcd_putc_xy(console_col * VIDEO_FONT_WIDTH,
-               console_row * VIDEO_FONT_HEIGHT, ' ');
+       lcd_putc_xy (console_col * VIDEO_FONT_WIDTH,
+                    console_row * VIDEO_FONT_HEIGHT,
+                    ' ');
 }

 /*----------------------------------------------------------------------*/

-static inline void console_newline(void)
+static inline void console_newline (void)
 {
        ++console_row;
        console_col = 0;
@@ -134,67 +118,61 @@ static inline void console_newline(void)
        /* Check if we need to scroll the terminal */
        if (console_row >= CONSOLE_ROWS) {
                /* Scroll everything up */
-               console_scrollup();
+               console_scrollup () ;
                --console_row;
        }
 }

-/*----------------------------------------------------------------------*/
```

```
-void lcd_putc(const char c)
+void lcd_putc (const char c)
 {
        if (!lcd_is_enabled) {
                serial_putc(c);
-
                return;
        }

        switch (c) {
-       case '\r':
-               console_col = 0;
+       case '\r':      console_col = 0;
+                       return;

-               return;
-       case '\n':
-               console_newline();
+       case '\n':      console_newline();
+                       return;

-               return;
        case '\t':      /* Tab (8 chars alignment) */
-               console_col +=  8;
-               console_col &= ~7;
+                       console_col +=  8;
+                       console_col &= ~7;

-               if (console_col >= CONSOLE_COLS)
-                       console_newline();
+                       if (console_col >= CONSOLE_COLS) {
+                               console_newline();
+                       }
+                       return;
```

```
-               return;
-       case '\b':
-               console_back();
+       case '\b':      console_back();
+                       return;

-               return;
-       default:
-               lcd_putc_xy(console_col * VIDEO_FONT_WIDTH,
-                       console_row * VIDEO_FONT_HEIGHT, c);
-               if (++console_col >= CONSOLE_COLS)
-                       console_newline();
+       default:        lcd_putc_xy (console_col * VIDEO_FONT_WIDTH,
+                                    console_row * VIDEO_FONT_HEIGHT,
+                                    c);
+                       if (++console_col >= CONSOLE_COLS) {
+                               console_newline();
+                       }
+                       return;
        }
+       /* NOTREACHED */
 }

-/*-----------------------------------------------------------------------*/
-
-void lcd_puts(const char *s)
+void lcd_puts (const char *s)
 {
-       if (!lcd_is_enabled) {
-               serial_puts(s);
-
+       /*if (!lcd_is_enabled) {
+               serial_puts (s);
```

```
                return;
-       }
-
+       }*/
+    serial_puts (s);
        while (*s) {
-               lcd_putc(*s++);
+               lcd_putc (*s++);
        }
 }

-/*----------------------------------------------------------------------*/
-
 void lcd_printf(const char *fmt, ...)
 {
        va_list args;
@@ -210,23 +188,43 @@ void lcd_printf(const char *fmt, ...)
 /************************************************************************/
 /* ** Low-Level Graphics Routines                                       */
 /************************************************************************/
+static void lcd_hline(int x,int y,int width,uint16_t color)
+{
+  uint16_t *pp;
+  int i;
+
+  pp = (uint16_t *)((uint32_t)lcd_base + y * lcd_line_length + x *
panel_info.vl_bpix / 8 );
+  for(i=0;i<width;i++)
+  {
+    *pp = color;
+    pp++;
+  }
+}
+
```

```
+static void lcd_vline(int x,int y,int height,uint16_t color)
+{
+  uint16_t *pp;
+  int i;
+
+  for(i=0;i<height;i++)
+  {
+    pp = (uint16_t *)((uint32_t)lcd_base + (y + i) * lcd_line_length + x *
NBITS(panel_info.vl_bpix)/ 8 );
+    *pp = color;
+  }
+}


-static void lcd_drawchars(ushort x, ushort y, uchar *str, int count)
+static void lcd_drawchars (ushort x, ushort y, uchar *str, int count)
 {
        uchar *dest;
        ushort row;


-#if defined(CONFIG_LCD_LOGO) && !
defined(CONFIG_LCD_INFO_BELOW_LOGO)
-       y += BMP_LOGO_HEIGHT;
-#endif
-
 #if LCD_BPP == LCD_MONOCHROME
        ushort off  = x * (1 << LCD_BPP) % 8;
 #endif


-       dest = (uchar *)(lcd_base + y * lcd_line_length + x * (1 << LCD_BPP) / 8);
+       dest = (uchar *)(lcd_base + y * lcd_line_length + x * (1<<LCD_BPP)/ 8);


-       for (row = 0; row < VIDEO_FONT_HEIGHT; ++row, dest +=
lcd_line_length) {
+       for (row=0;  row < VIDEO_FONT_HEIGHT;  ++row, dest +=
lcd_line_length)  {
```

```
                uchar *s = str;
                int i;
 #if LCD_BPP == LCD_COLOR16
@@ -239,7 +237,7 @@ static void lcd_drawchars(ushort x, ushort y, uchar *str,
int count)
                uchar rest = *d & -(1 << (8-off));
                uchar sym;
 #endif
-               for (i = 0; i < count; ++i) {
+               for (i=0; i<count; ++i) {
                        uchar c, bits;


                        c = *s++;
@@ -247,18 +245,18 @@ static void lcd_drawchars(ushort x, ushort y, uchar
*str, int count)


 #if LCD_BPP == LCD_MONOCHROME
                        sym  = (COLOR_MASK(lcd_color_fg) & bits) |
-                               (COLOR_MASK(lcd_color_bg) & ~bits);
+                              (COLOR_MASK(lcd_color_bg) & ~bits);


                        *d++ = rest | (sym >> off);
                        rest = sym << (8-off);
 #elif LCD_BPP == LCD_COLOR8
-                       for (c = 0; c < 8; ++c) {
+                       for (c=0; c<8; ++c) {
                                *d++ = (bits & 0x80) ?
                                                lcd_color_fg : lcd_color_bg;
                                bits <<= 1;
                        }
 #elif LCD_BPP == LCD_COLOR16
-                       for (c = 0; c < 8; ++c) {
+                       for (c=0; c<8; ++c) {
                                *d++ = (bits & 0x80) ?
```

```
                                        lcd_color_fg : lcd_color_bg;
                                bits <<= 1;
@@ -273,55 +271,198 @@ static void lcd_drawchars(ushort x, ushort y, uchar
*str, int count)

 /*------------------------------------------------------------------------*/

-static inline void lcd_puts_xy(ushort x, ushort y, uchar *s)
+static inline void lcd_puts_xy (ushort x, ushort y, uchar *s)
 {
-       lcd_drawchars(x, y, s, strlen((char *)s));
+#if defined(CONFIG_LCD_LOGO) && !
defined(CONFIG_LCD_INFO_BELOW_LOGO)
+       lcd_drawchars (x, y+BMP_LOGO_HEIGHT, s, strlen ((char *)s));
+#else
+       lcd_drawchars (x, y, s, strlen ((char *)s));
+#endif
 }

 /*------------------------------------------------------------------------*/

-static inline void lcd_putc_xy(ushort x, ushort y, uchar c)
+static inline void lcd_putc_xy (ushort x, ushort y, uchar c)
 {
-       lcd_drawchars(x, y, &c, 1);
+#if defined(CONFIG_LCD_LOGO) && !
defined(CONFIG_LCD_INFO_BELOW_LOGO)
+       lcd_drawchars (x, y+BMP_LOGO_HEIGHT, &c, 1);
+#else
+       lcd_drawchars (x, y, &c, 1);
+#endif
 }

-/************************************************************************/
-/** Small utility to check that you got the colours right              */
```

```
-/*****************************************************************************/
-#ifdef LCD_TEST_PATTERN
-
-#define     N_BLK_VERT      2
-#define     N_BLK_HOR       3


-static int test_colors[N_BLK_HOR*N_BLK_VERT] = {
-       CONSOLE_COLOR_RED,      CONSOLE_COLOR_GREEN,
       CONSOLE_COLOR_YELLOW,
-       CONSOLE_COLOR_BLUE,     CONSOLE_COLOR_MAGENTA,
       CONSOLE_COLOR_CYAN,
-};
-
-static void test_pattern(void)
+ulong calc_fbsize (void)
 {
-       ushort v_max  = panel_info.vl_row;
-       ushort h_max  = panel_info.vl_col;
-       ushort v_step = (v_max + N_BLK_VERT - 1) / N_BLK_VERT;
-       ushort h_step = (h_max + N_BLK_HOR  - 1) / N_BLK_HOR;
-       ushort v, h;
-       uchar *pix = (uchar *)lcd_base;
-
-       printf("[LCD] Test Pattern: %d x %d [%d x %d]\n",
-               h_max, v_max, h_step, v_step);
-
-       /* WARNING: Code silently assumes 8bit/pixel */
-       for (v = 0; v < v_max; ++v) {
-               uchar iy = v / v_step;
-               for (h = 0; h < h_max; ++h) {
-                       uchar ix = N_BLK_HOR * iy + (h/h_step);
-                       *pix++ = test_colors[ix];
-               }
-       }
```

```
+  ulong size;
+
+  int line_length = (panel_info.vl_col * NBITS(panel_info.vl_bpix)) / 8;
+  size = line_length * panel_info.vl_row;
+//#ifdef LCD_FRAMEBUFFER
+//  size = 4096;
+//#endif
+
+  return size;
 }
-#endif /* LCD_TEST_PATTERN */
+/*****************************************************************************/
+/*   根据给定的显示缓冲，初始化 lcd                                          */
+/*   uboot 使用内存有限，只能开辟一个基本窗口                                */
+void lcd_ctrl_init(void *lcdbase)
+{
+  ulong freq_lcdclk;
+  ulong freq_Hclk;
+  ulong fb_size;
+  unsigned char nn;
+  unsigned short *pp;
+  int i;
+
+  printf("************************************************************\n");
+  printf("initial lcd controller\n");
+  //配置管脚
+  GPICON_REG = 0xaaaaaaaa;
+  GPIPUD_REG = 0xaaaaaaaa;
+  GPJCON_REG = 0xaaaaaaaa;
+  GPJPUD_REG = 0xaaaaaaaa;
+
+  lcd_disable();
+  S3C_WINCON0 &= (~(S3C_WINCONx_ENWIN_F_ENABLE));
```

```
+
+
+  // (1)MOFPCON:SEL_BYPASS[3] value@0x7410800C 必须设置为'0'.
+  MIFPCON_REG &= (~SEL_BYPASS_MASK);
+  //(2)SPCON:LCD_SEL[1:0]value@0x74F0081A0 必须设置为'00',使用主机 I/F 类型,或者设置为'01' 使用 RGB I/F 类型。
+  SPCON_REG &= (~LCD_SEL_MASK);
+  SPCON_REG |= (RGB_IF_STYLE_MASK);
+  //(3)VIDCON0:配置视频输出格式和显示使能/禁止。
+
+  freq_lcdclk = S3CFB_PIXEL_CLOCK;
+  freq_Hclk = get_HCLK();
+  nn = (unsigned char)(freq_Hclk / freq_lcdclk) - 1;
+
+  if(freq_lcdclk < freq_Hclk/2)
+  {
+
+    S3C_VIDCON0 = S3C_VIDCON0_INTERLACE_F_PROGRESSIVE + S3C_VIDCON0_VIDOUT_RGB_IF + \
+          S3C_VIDCON0_PNRMODE_RGB_P + S3C_VIDCON0_CLKVALUP_ST_FRM + S3C_VIDCON0_CLKVAL_F(nn) + \
+          S3C_VIDCON0_CLKDIR_DIVIDED + S3C_VIDCON0_CLKSEL_F_HCLK;
+  }else
+  {
+    S3C_VIDCON0 = S3C_VIDCON0_INTERLACE_F_PROGRESSIVE + S3C_VIDCON0_VIDOUT_RGB_IF + \
+          S3C_VIDCON0_PNRMODE_RGB_P + S3C_VIDCON0_CLKVALUP_ST_FRM + S3C_VIDCON0_CLKVAL_F(0) + \
+          S3C_VIDCON0_CLKDIR_DIRECTED  + S3C_VIDCON0_CLKSEL_F_HCLK;
+  }
+  printf("  clk_freq:%d MHz,  div_freq:%d ,rea_freq:%d MHz \n",freq_lcdclk/1000000,nn,freq_Hclk/(nn+1)/1000000);
+  //(4)VIDCON1:RGB I/F 控制信号。
+  nn = 0;
```

```
+
+  if(S3CFB_IVCLK)
+  {
+    nn += S3C_VIDCON1_IVCLK_RISE_EDGE;
+  }
+  if(S3CFB_IHSYNC)
+  {
+    nn += S3C_VIDCON1_IHSYNC_INVERT;
+  }
+  if(S3CFB_IVSYNC)
+  {
+    nn += S3C_VIDCON1_IVSYNC_INVERT;
+  }
+  if(S3CFB_IVDEN)
+  {
+    nn += S3C_VIDCON1_IVDEN_INVERT;
+  }
+  S3C_VIDCON1 = (unsigned int)nn;
+  S3C_VIDCON2 = 0;
+  //(5)I80IFCONx:i80 系统 I/F 控制信号。
+  //(6)ITUIFCON0:ITU(BT.601)接口控制
+  //(7)VIDTCONx:配置视频输出时序和显示尺寸。
+  S3C_VIDTCON0 = S3C_VIDTCON0_VBPD(S3CFB_VBP - 1) |
S3C_VIDTCON0_VFPD(S3CFB_VFP - 1) | S3C_VIDTCON0_VSPW(S3CFB_VSW -
1);
+  S3C_VIDTCON1 = S3C_VIDTCON1_HBPD(S3CFB_HBP - 1) |
S3C_VIDTCON1_HFPD(S3CFB_HFP -1) | S3C_VIDTCON1_HSPW(S3CFB_HSW -
1);
+  S3C_VIDTCON2 = S3C_VIDTCON2_LINEVAL(S3CFB_VRES - 1) |
S3C_VIDTCON2_HOZVAL(S3CFB_HRES - 1);
+
+  printf("\n HBP = %d HFP = %d HSW = %d,Hpixs:
%d",S3CFB_HBP,S3CFB_HFP,S3CFB_HSW,S3CFB_HRES);
+  printf("\n VBP = %d VFP = %d VSW = %d,Vpixs:
%d",S3CFB_VBP,S3CFB_VFP,S3CFB_VSW,S3CFB_VRES);
```

```
+
+  //(8)WINCONx:窗口格式设置
+  S3C_WINCON0 = S3C_WINCONx_BPPMODE_F_16BPP_565|
S3C_WINCONx_BYTSWP_ENABLE;
+  S3C_WINCON1 = 0;
+  S3C_WINCON2 = 0;
+  S3C_WINCON3 = 0;
+  S3C_WINCON4 = 0;
+  //(9)VIDOSDxA ,VIDOSDxB:窗口位置设置
+
+  S3C_VIDOSD0A = S3C_VIDOSDxA_OSD_LTX_F(0) +
S3C_VIDOSDxA_OSD_LTY_F(0);
+  S3C_VIDOSD0B = S3C_VIDOSDxB_OSD_RBX_F(S3CFB_HRES) |
S3C_VIDOSDxB_OSD_RBY_F(S3CFB_VRES);
+  S3C_VIDOSD0C = S3C_VIDOSD0C_OSDSIZE(S3CFB_HRES*S3CFB_VRES);
+
+  S3C_VIDOSD1A = 0;
+  S3C_VIDOSD1B = 0;
+  S3C_VIDOSD1C = 0;
+  S3C_VIDOSD1D = 0;
+  S3C_VIDOSD2A = 0;
+  S3C_VIDOSD2B = 0;
+  S3C_VIDOSD2C = 0;
+  S3C_VIDOSD2D = 0;
+  S3C_VIDOSD3A = 0;
+  S3C_VIDOSD3B = 0;
+  S3C_VIDOSD3C = 0;
+  S3C_VIDOSD4A = 0;
+  S3C_VIDOSD4B = 0;
+  S3C_VIDOSD4C = 0;
+
+  //(10)VIDOSDxC:alpha 值设置
+
+  //(11)VIDWxxADDx:源图像地址设置
```

```
+
+  fb_size = calc_fbsize();
+//#ifdef LCD_FRAMEBUFFER
+ // fb_size =    (panel_info.vl_col * panel_info.vl_bpix) / 8 * panel_info.vl_row;
+//#endif
+ // S3C_VIDW00ADD0B0 = (unsigned int)(lcdbase) |
S3C_VIDWxxADD0_VBANK_F((unsigned int)lcdbase);
+  S3C_VIDW00ADD0B0 = virt_to_phys(lcdbase);
+  S3C_VIDW00ADD0B1 = 0;
+  S3C_VIDW01ADD0B0 = 0;
+  S3C_VIDW01ADD0B1 = 0;
+  S3C_VIDW02ADD0 = 0;
+  S3C_VIDW03ADD0 = 0;
+  S3C_VIDW04ADD0 = 0;
+
+ // S3C_VIDW00ADD1B0 = S3C_VIDWxxADD1_VBASEL_F((unsigned int)
(lcdbase) + fb_size);
+  S3C_VIDW00ADD1B0 = virt_to_phys((unsigned int)(lcdbase) + fb_size);
+  S3C_VIDW00ADD1B1 = 0;
+  S3C_VIDW01ADD1B0 = 0;
+  S3C_VIDW01ADD1B1 = 0;
+  S3C_VIDW02ADD1 = 0;
+  S3C_VIDW03ADD1 = 0;
+  S3C_VIDW04ADD1 = 0;
+
+  S3C_VIDW00ADD2 = S3C_VIDWxxADD2_OFFSIZE_F(0) |
(S3C_VIDWxxADD2_PAGEWIDTH_F(panel_info.vl_col*panel_info.vl_bpix/8));
+  S3C_VIDW01ADD2 = 0;
+  S3C_VIDW02ADD2 = 0;
+  S3C_VIDW03ADD2 = 0;
+  S3C_VIDW04ADD2 = 0;
+  //(12)WxKEYCONx:色键值寄存器
+  //(13)WINxMAP:窗口颜色控制
+  //(14)WPALCON:调色板控制寄存器
```

```
+  //(15)WxPDATAxx:索引窗口调色板数据
+  printf("\nFrameBuff:%08x",(unsigned int)lcdbase);
+#if 1
+  memset(lcdbase,0x55,fb_size);
+#else
+  pp = lcdbase;
+  for(i=0;i< S3CFB_HRES * S3CFB_VRES;i++)
+  {
+    *pp = 0xf100;
+    pp++;
+  }
+#endif
+  lcd_enable();
+  S3C_WINCON0 |= S3C_WINCONx_ENWIN_F_ENABLE;
+  printf("\n  LCD initialization Finished. \n");


+}
+
 /************************************************************************/
 /* ** GENERIC Initialization Routines                                   */
 /************************************************************************/
@@ -335,12 +476,12 @@ int drv_lcd_init (void)

         lcd_line_length = (panel_info.vl_col * NBITS (panel_info.vl_bpix)) / 8;

-        lcd_init(lcd_base);             /* LCD initialization */
+        lcd_init (lcd_base);            /* LCD initialization */

         /* Device initialization */
-        memset(&lcddev, 0, sizeof(lcddev));
+        memset (&lcddev, 0, sizeof (lcddev));

-        strcpy(lcddev.name, "lcd");
```

```
+       strcpy (lcddev.name, "lcd");
        lcddev.ext   = 0;                       /* No extensions */
        lcddev.flags = DEV_FLAGS_OUTPUT;      /* Output only */
        lcddev.putc  = lcd_putc;                  /* 'putc' function */
@@ -352,8 +493,7 @@ int drv_lcd_init (void)
 }

 /*------------------------------------------------------------------------*/
-static
-int do_lcd_clear(cmd_tbl_t *cmdtp, int flag, int argc, char *const argv[])
+static int do_lcd_clear(cmd_tbl_t *cmdtp, int flag, int argc, char *const argv[])
 {
        lcd_clear();
        return 0;
@@ -367,36 +507,37 @@ void lcd_clear(void)

 #elif LCD_BPP == LCD_COLOR8
        /* Setting the palette */
-       lcd_setcolreg(CONSOLE_COLOR_BLACK, 0, 0, 0);
-       lcd_setcolreg(CONSOLE_COLOR_RED, 0xFF, 0, 0);
-       lcd_setcolreg(CONSOLE_COLOR_GREEN, 0, 0xFF, 0);
-       lcd_setcolreg(CONSOLE_COLOR_YELLOW, 0xFF, 0xFF, 0);
-       lcd_setcolreg(CONSOLE_COLOR_BLUE, 0, 0, 0xFF);
-       lcd_setcolreg(CONSOLE_COLOR_MAGENTA, 0xFF, 0, 0xFF);
-       lcd_setcolreg(CONSOLE_COLOR_CYAN, 0, 0xFF, 0xFF);
-       lcd_setcolreg(CONSOLE_COLOR_GREY, 0xAA, 0xAA, 0xAA);
-       lcd_setcolreg(CONSOLE_COLOR_WHITE, 0xFF, 0xFF, 0xFF);
+       /*lcd_setcolreg  (CONSOLE_COLOR_BLACK,       0,    0,    0);
+       lcd_setcolreg  (CONSOLE_COLOR_RED, 0xFF,    0,    0);
+       lcd_setcolreg  (CONSOLE_COLOR_GREEN,       0, 0xFF,    0);
+       lcd_setcolreg  (CONSOLE_COLOR_YELLOW,   0xFF, 0xFF,    0);
+       lcd_setcolreg  (CONSOLE_COLOR_BLUE,        0,    0, 0xFF);
+       lcd_setcolreg  (CONSOLE_COLOR_MAGENTA,         0xFF,    0, 0xFF);
+       lcd_setcolreg  (CONSOLE_COLOR_CYAN,          0, 0xFF, 0xFF);
```

```
+       lcd_setcolreg  (CONSOLE_COLOR_GREY,       0xAA, 0xAA, 0xAA);
+       lcd_setcolreg  (CONSOLE_COLOR_WHITE,     0xFF, 0xFF, 0xFF);*/
 #endif

 #ifndef CONFIG_SYS_WHITE_ON_BLACK
-       lcd_setfgcolor(CONSOLE_COLOR_BLACK);
-       lcd_setbgcolor(CONSOLE_COLOR_WHITE);
+       lcd_setfgcolor (CONSOLE_COLOR_WHITE);
+       lcd_setbgcolor (CONSOLE_COLOR_BLACK);
 #else
-       lcd_setfgcolor(CONSOLE_COLOR_WHITE);
-       lcd_setbgcolor(CONSOLE_COLOR_BLACK);
+       lcd_setfgcolor (CONSOLE_COLOR_BLACK);
+       lcd_setbgcolor (CONSOLE_COLOR_WHITE);
 #endif      /* CONFIG_SYS_WHITE_ON_BLACK */

 #ifdef      LCD_TEST_PATTERN
        test_pattern();
 #else
        /* set framebuffer to background color */
-       memset((char *)lcd_base,
+       memset ((char *)lcd_base,
                COLOR_MASK(lcd_getbgcolor()),
                lcd_line_length*panel_info.vl_row);
 #endif
        /* Paint the logo and retrieve LCD base address */
-       debug("[LCD] Drawing the logo...\n");
+       //debug ("[LCD] Drawing the logo...\n");
        lcd_console_address = lcd_logo ();
+       //lcd_console_address = (void *)lcd_base;

        console_col = 0;
        console_row = 0;
@@ -410,12 +551,12 @@ U_BOOT_CMD(
```

```
 /*----------------------------------------------------------------------*/

-static int lcd_init(void *lcdbase)
+static int lcd_init (void *lcdbase)
 {
        /* Initialize the lcd controller */
-       debug("[LCD] Initializing LCD frambuffer at %p\n", lcdbase);
+       debug ("[LCD] Initializing LCD frambuffer at %p\n", lcdbase);

-       lcd_ctrl_init(lcdbase);
+       lcd_ctrl_init (lcdbase);
        lcd_is_enabled = 1;
        lcd_clear();
        lcd_enable ();
@@ -442,13 +583,13 @@ static int lcd_init(void *lcdbase)
  *
  * Note that this is running from ROM, so no write access to global data.
  */
-ulong lcd_setmem(ulong addr)
+ulong lcd_setmem (ulong addr)
 {
        ulong size;
-       int line_length = (panel_info.vl_col * NBITS(panel_info.vl_bpix)) / 8;
+       int line_length = (panel_info.vl_col * NBITS (panel_info.vl_bpix)) / 8;

-       debug("LCD panel info: %d x %d, %d bit/pix\n", panel_info.vl_col,
-               panel_info.vl_row, NBITS(panel_info.vl_bpix));
+       debug ("LCD panel info: %d x %d, %d bit/pix\n",
+               panel_info.vl_col, panel_info.vl_row, NBITS (panel_info.vl_bpix) );

        size = line_length * panel_info.vl_row;

@@ -458,21 +599,21 @@ ulong lcd_setmem(ulong addr)
```

```
        /* Allocate pages for the frame buffer. */
        addr -= size;

-       debug("Reserving %ldk for LCD Framebuffer at: %08lx\n", size>>10,
addr);
+       debug ("Reserving %ldk for LCD Framebuffer at: %08lx\n", size>>10,
addr);

-       return addr;
+       return (addr);
 }

 /*----------------------------------------------------------------------*/

-static void lcd_setfgcolor(int color)
+static void lcd_setfgcolor (int color)
 {
        lcd_color_fg = color;
 }

 /*----------------------------------------------------------------------*/

-static void lcd_setbgcolor(int color)
+static void lcd_setbgcolor (int color)
 {
        lcd_color_bg = color;
 }
@@ -480,7 +621,7 @@ static void lcd_setbgcolor(int color)
 /*----------------------------------------------------------------------*/

 #ifdef       NOT_USED_SO_FAR
-static int lcd_getfgcolor(void)
+static int lcd_getfgcolor (void)
 {
```

```
        return lcd_color_fg;
 }
@@ -488,91 +629,90 @@ static int lcd_getfgcolor(void)

 /*----------------------------------------------------------------------*/

-static int lcd_getbgcolor(void)
+static int lcd_getbgcolor (void)
 {
        return lcd_color_bg;
 }

 /*----------------------------------------------------------------------*/

-/************************************************************************/
-/* ** Chipset depending Bitmap / Logo stuff...                          */
-/************************************************************************/
-static inline ushort *configuration_get_cmap(void)
+void lcd_disable (void)
 {
-#if defined CONFIG_CPU_PXA
-       struct pxafb_info *fbi = &panel_info.pxa;
-       return (ushort *)fbi->palette;
-#elif defined(CONFIG_MPC823)
-       immap_t *immr = (immap_t *) CONFIG_SYS_IMMR;
-       cpm8xx_t *cp = &(immr->im_cpm);
-       return (ushort *)&(cp->lcd_cmap[255 * sizeof(ushort)]);
-#elif defined(CONFIG_ATMEL_LCD)
-       return (ushort *)(panel_info.mmio + ATMEL_LCDC_LUT(0));
-#elif !defined(CONFIG_ATMEL_HLCD) && !defined(CONFIG_EXYNOS_FB)
-       return panel_info.cmap;
-#else
-#if defined(CONFIG_LCD_LOGO)
-       return bmp_logo_palette;
```

```
-#else
-       return NULL;
-#endif
-#endif
+  S3C_VIDCON0 &= (~(S3C_VIDCON0_ENVID_ENABLE |
S3C_VIDCON0_ENVID_F_ENABLE));
+}
+
+void lcd_enable (void)
+{
+  S3C_VIDCON0 |= (S3C_VIDCON0_ENVID_ENABLE |
S3C_VIDCON0_ENVID_F_ENABLE);
 }


+
+void lcd_panel_disable(void)
+{
+  MIFPCON_REG |= SEL_BYPASS_MASK;
+}
+/************************************************************************/
+/************************************************************************/
+/* ** Chipset depending Bitmap / Logo stuff...                          */
+/************************************************************************/
 #ifdef CONFIG_LCD_LOGO
-void bitmap_plot(int x, int y)
+void bitmap_plot (int x, int y)
 {
 #ifdef CONFIG_ATMEL_LCD
-       uint *cmap = (uint *)bmp_logo_palette;
+       uint *cmap;
 #else
-       ushort *cmap = (ushort *)bmp_logo_palette;
+       ushort *cmap;
 #endif
```

```
        ushort i, j;
        uchar *bmap;
        uchar *fb;
        ushort *fb16;
-#if defined(CONFIG_MPC823)
-       immap_t *immr = (immap_t *) CONFIG_SYS_IMMR;
-       cpm8xx_t *cp = &(immr->im_cpm);
+#if defined(CONFIG_CPU_PXA)
+       struct pxafb_info *fbi = &panel_info.pxa;
+#elif defined(CONFIG_MPC823)
+       volatile immap_t *immr = (immap_t *) CONFIG_SYS_IMMR;
+       volatile cpm8xx_t *cp = &(immr->im_cpm);
 #endif

-       debug("Logo: width %d  height %d  colors %d  cmap %d\n",
+       debug ("Logo: width %d  height %d  colors %d  cmap %d\n",
                BMP_LOGO_WIDTH, BMP_LOGO_HEIGHT, BMP_LOGO_COLORS,
-               ARRAY_SIZE(bmp_logo_palette));
+               (int)(sizeof(bmp_logo_palette)/(sizeof(ushort))));

        bmap = &bmp_logo_bitmap[0];
        fb   = (uchar *)(lcd_base + y * lcd_line_length + x);

        if (NBITS(panel_info.vl_bpix) < 12) {
-               /* Leave room for default color map
-                * default case: generic system with no cmap (most likely 16bpp)
-                * cmap was set to the source palette, so no change is done.
-                * This avoids even more ifdefs in the next stanza
-                */
-#if defined(CONFIG_MPC823)
-               cmap = (ushort *) &(cp->lcd_cmap[BMP_LOGO_OFFSET *
sizeof(ushort)]);
+               /* Leave room for default color map */
+#if defined(CONFIG_CPU_PXA)
```

```
+               cmap = (ushort *)fbi->palette;
+#elif defined(CONFIG_MPC823)
+               cmap = (ushort *)&(cp->lcd_cmap[BMP_LOGO_OFFSET*sizeof(ushort)]);
 #elif defined(CONFIG_ATMEL_LCD)
-               cmap = (uint *)configuration_get_cmap();
+               cmap = (uint *) (panel_info.mmio + ATMEL_LCDC_LUT(0));
 #else
-               cmap = configuration_get_cmap();
+               /*
+                * default case: generic system with no cmap (most likely 16bpp)
+                * We set cmap to the source palette, so no change is done.
+                * This avoids even more ifdef in the next stanza
+                */
+               cmap = bmp_logo_palette;
 #endif

                WATCHDOG_RESET();

                /* Set color map */
-               for (i = 0; i < ARRAY_SIZE(bmp_logo_palette); ++i) {
+               for (i=0; i<(sizeof(bmp_logo_palette)/(sizeof(ushort))); ++i) {
                        ushort colreg = bmp_logo_palette[i];
 #ifdef CONFIG_ATMEL_LCD
                        uint lut_entry;
 #ifdef CONFIG_ATMEL_LCD_BGR555
                        lut_entry = ((colreg & 0x000F) << 11) |
-                                       ((colreg & 0x00F0) <<  2) |
-                                       ((colreg & 0x0F00) >>  7);
+                                   ((colreg & 0x00F0) <<  2) |
+                                   ((colreg & 0x0F00) >>  7);
 #else /* CONFIG_ATMEL_LCD_RGB565 */
                        lut_entry = ((colreg & 0x000F) << 1) |
-                                       ((colreg & 0x00F0) << 3) |
```

```
-                                       ((colreg & 0x0F00) << 4);
+                                      ((colreg & 0x00F0) << 3) |
+                                      ((colreg & 0x0F00) << 4);
 #endif
                        *(cmap + BMP_LOGO_OFFSET) = lut_entry;
                        cmap++;
@@ -587,8 +727,8 @@ void bitmap_plot(int x, int y)

                WATCHDOG_RESET();

-               for (i = 0; i < BMP_LOGO_HEIGHT; ++i) {
-                       memcpy(fb, bmap, BMP_LOGO_WIDTH);
+               for (i=0; i<BMP_LOGO_HEIGHT; ++i) {
+                       memcpy (fb, bmap, BMP_LOGO_WIDTH);
                        bmap += BMP_LOGO_WIDTH;
                        fb   += panel_info.vl_col;
                }
@@ -596,8 +736,8 @@ void bitmap_plot(int x, int y)
        else { /* true color mode */
                u16 col16;
                fb16 = (ushort *)(lcd_base + y * lcd_line_length + x);
-               for (i = 0; i < BMP_LOGO_HEIGHT; ++i) {
-                       for (j = 0; j < BMP_LOGO_WIDTH; j++) {
+               for (i=0; i<BMP_LOGO_HEIGHT; ++i) {
+                       for (j=0; j<BMP_LOGO_WIDTH; j++) {
                                col16 = bmp_logo_palette[(bmap[j]-16)];
                                fb16[j] =
                                        ((col16 & 0x000F) << 1) |
@@ -611,8 +751,6 @@ void bitmap_plot(int x, int y)

        WATCHDOG_RESET();
 }
-#else
-static inline void bitmap_plot(int x, int y) {}
```

```
 #endif /* CONFIG_LCD_LOGO */

 /*----------------------------------------------------------------------*/
@@ -624,47 +762,8 @@ static inline void bitmap_plot(int x, int y) {}

 #ifdef CONFIG_SPLASH_SCREEN_ALIGN
 #define BMP_ALIGN_CENTER      0x7FFF
-
-static void splash_align_axis(int *axis, unsigned long panel_size,
-                              unsigned long picture_size)
-{
-       unsigned long panel_picture_delta = panel_size - picture_size;
-       unsigned long axis_alignment;
-
-       if (*axis == BMP_ALIGN_CENTER)
-               axis_alignment = panel_picture_delta / 2;
-       else if (*axis < 0)
-               axis_alignment = panel_picture_delta + *axis + 1;
-       else
-               return;
-
-       *axis = max(0, axis_alignment);
-}
 #endif

-#if defined(CONFIG_MPC823) || defined(CONFIG_MCC200)
-#define FB_PUT_BYTE(fb, from) *(fb)++ = (255 - *(from)++)
-#else
-#define FB_PUT_BYTE(fb, from) *(fb)++ = *(from)++
-#endif
-
-#if defined(CONFIG_BMP_16BPP)
-#if defined(CONFIG_ATMEL_LCD_BGR555)
-static inline void fb_put_word(uchar **fb, uchar **from)
```

```
-{
-	*(*fb)++ = (((*from)[0] & 0x1f) << 2) | ((*from)[1] & 0x03);
-	*(*fb)++ = ((*from)[0] & 0xe0) | (((*from)[1] & 0x7c) >> 2);
-	*from += 2;
-}
-#else
-static inline void fb_put_word(uchar **fb, uchar **from)
-{
-	*(*fb)++ = *(*from)++;
-	*(*fb)++ = *(*from)++;
-}
-#endif
-#endif /* CONFIG_BMP_16BPP */
-
 int lcd_display_bitmap(ulong bmp_image, int x, int y)
 {
 #if !defined(CONFIG_MCC200)
@@ -679,48 +778,58 @@ int lcd_display_bitmap(ulong bmp_image, int x, int y)
 	unsigned long width, height, byte_width;
 	unsigned long pwidth = panel_info.vl_col;
 	unsigned colors, bpix, bmp_bpix;
+#if defined(CONFIG_CPU_PXA)
+	struct pxafb_info *fbi = &panel_info.pxa;
+#elif defined(CONFIG_MPC823)
+	volatile immap_t *immr = (immap_t *) CONFIG_SYS_IMMR;
+	volatile cpm8xx_t *cp = &(immr->im_cpm);
+#endif

-	if (!bmp || !((bmp->header.signature[0] == 'B') &&
-		      (bmp->header.signature[1] == 'M'))) {
-		printf("Error: no valid bmp image at %lx\n", bmp_image);
-
+	if (!((bmp->header.signature[0]=='B') &&
+		      (bmp->header.signature[1]=='M'))) {
```

```
+               printf ("Error: no valid bmp image at %lx\n", bmp_image);
                return 1;
        }

-       width = le32_to_cpu(bmp->header.width);
-       height = le32_to_cpu(bmp->header.height);
+       width = le32_to_cpu (bmp->header.width);
+       height = le32_to_cpu (bmp->header.height);
        bmp_bpix = le16_to_cpu(bmp->header.bit_count);
        colors = 1 << bmp_bpix;

        bpix = NBITS(panel_info.vl_bpix);

-       if ((bpix != 1) && (bpix != 8) && (bpix != 16) && (bpix != 32)) {
+       if ((bpix != 1) && (bpix != 8) && (bpix != 16)) {
                printf ("Error: %d bit/pixel mode, but BMP has %d bit/pixel\n",
                        bpix, bmp_bpix);
-
                return 1;
        }

        /* We support displaying 8bpp BMPs on 16bpp LCDs */
-       if (bpix != bmp_bpix && (bmp_bpix != 8 || bpix != 16 || bpix != 32)) {
+       if (bpix != bmp_bpix && (bmp_bpix != 8 || bpix != 16)) {
                printf ("Error: %d bit/pixel mode, but BMP has %d bit/pixel\n",
                        bpix,
                        le16_to_cpu(bmp->header.bit_count));
-
                return 1;
        }

-       debug("Display-bmp: %d x %d  with %d colors\n",
+       debug ("Display-bmp: %d x %d  with %d colors\n",
                (int)width, (int)height, (int)colors);
```

```
 #if !defined(CONFIG_MCC200)
        /* MCC200 LCD doesn't need CMAP, supports 1bpp b&w only */
        if (bmp_bpix == 8) {
-               cmap = configuration_get_cmap();
+#if defined(CONFIG_CPU_PXA)
+               cmap = (ushort *)fbi->palette;
+#elif defined(CONFIG_MPC823)
+               cmap = (ushort *)&(cp->lcd_cmap[255*sizeof(ushort)]);
+#elif !defined(CONFIG_ATMEL_LCD)
+               cmap = cmap;//panel_info.cmap;
+#endif
+
                cmap_base = cmap;

                /* Set color map */
-               for (i = 0; i < colors; ++i) {
+               for (i=0; i<colors; ++i) {
                        bmp_color_table_entry_t cte = bmp->color_table[i];
 #if !defined(CONFIG_ATMEL_LCD)
                        ushort colreg =
@@ -755,7 +864,8 @@ int lcd_display_bitmap(ulong bmp_image, int x, int y)
         * specific.
         */
 #if defined(CONFIG_MCC200)
-       if (bpix == 1) {
+       if (bpix==1)
+       {
                width = ((width + 7) & ~7) >> 3;
                x     = ((x + 7) & ~7) >> 3;
                pwidth= ((pwidth + 7) & ~7) >> 3;
@@ -765,16 +875,23 @@ int lcd_display_bitmap(ulong bmp_image, int x, int y)
        padded_line = (width&0x3) ? ((width&~0x3)+4) : (width);
```

```
 #ifdef CONFIG_SPLASH_SCREEN_ALIGN
-       splash_align_axis(&x, pwidth, width);
-       splash_align_axis(&y, panel_info.vl_row, height);
+       if (x == BMP_ALIGN_CENTER)
+               x = max(0, (pwidth - width) / 2);
+       else if (x < 0)
+               x = max(0, pwidth - width + x + 1);
+
+       if (y == BMP_ALIGN_CENTER)
+               y = max(0, (panel_info.vl_row - height) / 2);
+       else if (y < 0)
+               y = max(0, panel_info.vl_row - height + y + 1);
 #endif /* CONFIG_SPLASH_SCREEN_ALIGN */

-       if ((x + width) > pwidth)
+       if ((x + width)>pwidth)
                width = pwidth - x;
-       if ((y + height) > panel_info.vl_row)
+       if ((y + height)>panel_info.vl_row)
                height = panel_info.vl_row - y;

-       bmap = (uchar *)bmp + le32_to_cpu(bmp->header.data_offset);
+       bmap = (uchar *)bmp + le32_to_cpu (bmp->header.data_offset);
        fb   = (uchar *) (lcd_base +
                (y + height - 1) * lcd_line_length + x * bpix / 8);

@@ -790,7 +907,11 @@ int lcd_display_bitmap(ulong bmp_image, int x, int y)
                        WATCHDOG_RESET();
                        for (j = 0; j < width; j++) {
                                if (bpix != 16) {
-                                       FB_PUT_BYTE(fb, bmap);
+#if defined(CONFIG_CPU_PXA) || defined(CONFIG_ATMEL_LCD)
+                                       *(fb++) = *(bmap++);
+#elif defined(CONFIG_MPC823) || defined(CONFIG_MCC200)
```

```
+                                       *(fb++) = 255 - *(bmap++);
+#endif
                                } else {
                                        *(uint16_t *)fb = cmap_base[*(bmap++)];
                                        fb += sizeof(uint16_t) / sizeof(*fb);
@@ -805,37 +926,33 @@ int lcd_display_bitmap(ulong bmp_image, int x, int y)
        case 16:
                for (i = 0; i < height; ++i) {
                        WATCHDOG_RESET();
-                       for (j = 0; j < width; j++)
-                               fb_put_word(&fb, &bmap);
-
-                       bmap += (padded_line - width) * 2;
-                       fb   -= (width * 2 + lcd_line_length);
-               }
-               break;
-#endif /* CONFIG_BMP_16BPP */
-
-#if defined(CONFIG_BMP_32BPP)
-       case 32:
-               for (i = 0; i < height; ++i) {
                        for (j = 0; j < width; j++) {
+#if defined(CONFIG_ATMEL_LCD_BGR555)
+                               *(fb++) = ((bmap[0] & 0x1f) << 2) |
+                                       (bmap[1] & 0x03);
+                               *(fb++) = (bmap[0] & 0xe0) |
+                                       ((bmap[1] & 0x7c) >> 2);
+                               bmap += 2;
+#else
                                *(fb++) = *(bmap++);
                                *(fb++) = *(bmap++);
-                               *(fb++) = *(bmap++);
-                               *(fb++) = *(bmap++);
+#endif
                        }
                }
                break;
```

```
                        }
-                       fb  -= (lcd_line_length + width * (bpix / 8));
+                       bmap += (padded_line - width) * 2;
+                       fb   -= (width * 2 + lcd_line_length);
                }
                break;
-#endif /* CONFIG_BMP_32BPP */
+#endif /* CONFIG_BMP_16BPP */
+
        default:
                break;
        };

-       return 0;
+       return (0);
 }
 #endif

-static void *lcd_logo(void)
+static void *lcd_logo (void)
 {
 #ifdef CONFIG_SPLASH_SCREEN
        char *s;
@@ -848,15 +965,13 @@ static void *lcd_logo(void)

                addr = simple_strtoul (s, NULL, 16);
 #ifdef CONFIG_SPLASH_SCREEN_ALIGN
-               s = getenv("splashpos");
-               if (s != NULL) {
+               if ((s = getenv ("splashpos")) != NULL) {
                        if (s[0] == 'm')
                                x = BMP_ALIGN_CENTER;
                        else
-                               x = simple_strtol(s, NULL, 0);
```

```
+                               x = simple_strtol (s, NULL, 0);

-                       s = strchr(s + 1, ',');
-                       if (s != NULL) {
+                       if ((s = strchr (s + 1, ',')) != NULL) {
                                if (s[1] == 'm')
                                        y = BMP_ALIGN_CENTER;
                                else
@@ -865,12 +980,25 @@ static void *lcd_logo(void)
                }
 #endif /* CONFIG_SPLASH_SCREEN_ALIGN */

-               if (bmp_display(addr, x, y) == 0)
-                       return (void *)lcd_base;
+#ifdef CONFIG_VIDEO_BMP_GZIP
+               bmp_image_t *bmp = (bmp_image_t *)addr;
+               unsigned long len;
+
+               if (!((bmp->header.signature[0]=='B') &&
+                     (bmp->header.signature[1]=='M'))) {
+                       addr = (ulong)gunzip_bmp(addr, &len);
+               }
+#endif
+
+               if (lcd_display_bitmap (addr, x, y) == 0) {
+                       return ((void *)lcd_base);
+               }
        }
 #endif /* CONFIG_SPLASH_SCREEN */

-       bitmap_plot(0, 0);
+#ifdef CONFIG_LCD_LOGO
+       bitmap_plot (0, 0);
+#endif /* CONFIG_LCD_LOGO */
```

```
 #ifdef CONFIG_LCD_INFO
        console_col = LCD_INFO_X / VIDEO_FONT_WIDTH;
@@ -879,11 +1007,13 @@ static void *lcd_logo(void)
 #endif /* CONFIG_LCD_INFO */

 #if defined(CONFIG_LCD_LOGO) && !
defined(CONFIG_LCD_INFO_BELOW_LOGO)
-       return (void *)((ulong)lcd_base + BMP_LOGO_HEIGHT * lcd_line_length);
+       return ((void *)((ulong)lcd_base + BMP_LOGO_HEIGHT * lcd_line_length));
 #else
-       return (void *)lcd_base;
+       return ((void *)lcd_base);
 #endif /* CONFIG_LCD_LOGO && !CONFIG_LCD_INFO_BELOW_LOGO */
 }

 /*----------------------------------------------------------------------*/
 /*----------------------------------------------------------------------*/
+
+#endif /* CONFIG_LCD */

---------------------------- include/configs/smdk6410.h ----------------------------
index cae9f34..d1f7daa 100644
@@ -36,6 +36,13 @@
  * High Level Configuration Options
  * (easy to change)
  */
+/*
+ * LCD support
+ */
+#define CONFIG_LCD
+#ifdef CONFIG_LCD
+#define CONFIG_LCD_LOGO
+#endif
```

```
 /*
  * MMC Support

------------------------------- include/lcd.h ------------------------------
index 42070d7..a2ba81d 100644
@@ -29,6 +29,8 @@
 #ifndef _LCD_H_
 #define _LCD_H_

+#include <s3cfb_LCD.h>//zxd
+
 extern char lcd_is_enabled;

 extern int lcd_line_length;
@@ -262,6 +264,17 @@ typedef struct vidinfo {
 } vidinfo_t;

 void init_panel_info(vidinfo_t *vid);
+//zxd -->
+#elif CONFIG_S3C6410
+typedef struct vidinfo {
+       ushort       vl_col;                /* 行点数 */
+       ushort       vl_row;                /* 行数 */
+       ushort       vl_width;     /* 显示宽度 */
+       ushort       vl_height;    /* 显示高度 */
+
+       u_char       vl_bpix;
+} vidinfo_t;
+//zxd <--

 #else
```

```
@@ -318,7 +331,8 @@ void lcd_show_board_info(void);
 #define LCD_COLOR4         2
 #define LCD_COLOR8         3
 #define LCD_COLOR16        4
-
+//zxd
+#define LCD_BPP LCD_COLOR16
 /*----------------------------------------------------------------------*/
 #if defined(CONFIG_LCD_INFO_BELOW_LOGO)
 # define LCD_INFO_X         0

---------------------------- include/s3cfb-RegLCD.h ----------------------------
new file mode 100644
index 0000000..5be9593
@@ -0,0 +1,527 @@
+/* linux/arch/arm/plat-s3c/include/plat/regs-lcd.h
+ *
+ * Copyright (c) 2003 Simtec Electronics <linux@simtec.co.uk>
+ *              S3C_VIDCON0     http://www.simtec.co.uk/products/SWLINUX/
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License version 2 as
+ * published by the Free Software Foundation.
+*/
+
+
+#ifndef ___ASM_ARCH_REGS_LCD_H
+#define ___ASM_ARCH_REGS_LCD_H
+
+#include <common.h>
+#include <asm/arch/s3c6410.h>
+
+/**************************************************************************/
+/* LCD Registers for S3C2443/2450/S3C6400/6410 */
```

```
+
+#define S3C_LCDREG(x)           __REG((x) + ELFIN_LCD_BASE)
+
+/* LCD control registers */
+#define S3C_VIDCON0             S3C_LCDREG(0x00)    /* Video control 0
register */
+#define S3C_VIDCON1             S3C_LCDREG(0x04)    /* Video control 1
register */
+
+#if defined(CONFIG_CPU_S3C2443)||defined(CONFIG_CPU_S3C2450) ||
defined(CONFIG_CPU_S3C2416)
+#define S3C_VIDTCON0            S3C_LCDREG(0x08)    /* LCD CONTROL 1 */
+#define S3C_VIDTCON1            S3C_LCDREG(0x0C)    /* LCD CONTROL 1 */
+#define S3C_VIDTCON2            S3C_LCDREG(0x10)    /* LCD CONTROL 1 */
+#define S3C_WINCON0             S3C_LCDREG(0x14)    /* LCD CONTROL 1 */
+#define S3C_WINCON1             S3C_LCDREG(0x18)    /* LCD CONTROL 1 */
+#define S3C_VIDOSD0A            S3C_LCDREG(0x28)    /* LCD CONTROL 1 */
+#define S3C_VIDOSD0B            S3C_LCDREG(0x2C)    /* LCD CONTROL 1 */
+#define S3C_VIDOSD0C            S3C_LCDREG(0x30)    /* LCD CONTROL 1 */
+#define S3C_VIDOSD1A            S3C_LCDREG(0x34)    /* LCD CONTROL 1 */
+#define S3C_VIDOSD1B            S3C_LCDREG(0x38)    /* LCD CONTROL 1 */
+#define S3C_VIDOSD1C            S3C_LCDREG(0x3C)    /* LCD CONTROL 1 */
+#define S3C_VIDW00ADD0B0        S3C_LCDREG(0x64)    /* LCD CONTROL 1 */
+#define S3C_VIDW00ADD0B1        S3C_LCDREG(0x68)    /* LCD CONTROL 1 */
+#define S3C_VIDW01ADD0              S3C_LCDREG(0x6C)    /* LCD
CONTROL 1 */
+#define S3C_VIDW00ADD1B0        S3C_LCDREG(0x7C)    /* LCD CONTROL 1 */
+#define S3C_VIDW00ADD1B1        S3C_LCDREG(0x80)    /* LCD CONTROL 1 */
+#define S3C_VIDW01ADD1              S3C_LCDREG(0x84)    /* LCD
CONTROL 1 */
+#define S3C_VIDW00ADD2B0        S3C_LCDREG(0x94)    /* LCD CONTROL 1 */
+#define S3C_VIDW00ADD2B1        S3C_LCDREG(0x98)    /* LCD CONTROL 1 */
+#define S3C_VIDW01ADD2              S3C_LCDREG(0x9C)    /* LCD
CONTROL 1 */
+#define S3C_VIDINTCON           S3C_LCDREG(0xAC)    /* LCD CONTROL 1 */
+#define S3C_W1KEYCON0           S3C_LCDREG(0xB0)    /* LCD CONTROL 1 */
```

```
+#define S3C_W1KEYCON1		S3C_LCDREG(0xB4)	/* LCD CONTROL 1 */
+#define S3C_WIN0MAP		S3C_LCDREG(0xD0)	/* LCD CONTROL 1 */
+#define S3C_WIN1MAP		S3C_LCDREG(0xD4)	/* LCD CONTROL 1 */
+#define S3C_WPALCON		S3C_LCDREG(0xE4)	/* LCD CONTROL 1 */
+#define S3C_SYSIFCON0		S3C_LCDREG(0x130)	/* LCD CONTROL 1 */
+#define S3C_SYSIFCON1		S3C_LCDREG(0x134)	/* LCD CONTROL 1 */
+#define S3C_DITHMODE		S3C_LCDREG(0x138)	/* LCD CONTROL 1 */
+#define S3C_SIFCCON0		S3C_LCDREG(0x13C)	/* LCD CONTROL 1 */
+#define S3C_SIFCCON1		S3C_LCDREG(0x140)	/* LCD CONTROL 1 */
+#define S3C_SIFCCON2		S3C_LCDREG(0x144)	/* LCD CONTROL 1 */
+#define S3C_CPUTRIGCON2		S3C_LCDREG(0x160)	/* LCD
CONTROL 1 */
+
+#else //if defined(CONFIG_CPU_S3C6400) || defined(CONFIG_CPU_S3C6410)
|| defined(CONFIG_CPU_S5P6440) || defined(CONFIG_CPU_S5PC100)
+#define S3C_VIDCON2		S3C_LCDREG(0x08)	/* Video control 2
register */
+#define S3C_VIDTCON0		S3C_LCDREG(0x10)	/* Video time control 0
register */
+#define S3C_VIDTCON1		S3C_LCDREG(0x14)	/* Video time control 1
register */
+#define S3C_VIDTCON2		S3C_LCDREG(0x18)	/* Video time control 2
register */
+#define S3C_VIDTCON3		S3C_LCDREG(0x1C)	/* Video time control 3
register */
+
+#define S3C_WINCON0		S3C_LCDREG(0x20)	/* Window control 0
register */
+#define S3C_WINCON1		S3C_LCDREG(0x24)	/* Window control 1
register */
+#define S3C_WINCON2		S3C_LCDREG(0x28)	/* Window control 2
register */
+#define S3C_WINCON3		S3C_LCDREG(0x2C)	/* Window control 3
register */
+#define S3C_WINCON4		S3C_LCDREG(0x30)	/* Window control 4
register*/
+
```

```
+
+#define S3C_VIDOSD0A            S3C_LCDREG(0x40)    /* Video Window 0
position control register */
+#define S3C_VIDOSD0B            S3C_LCDREG(0x44)    /* Video Window 0
position control register1 */
+#define S3C_VIDOSD0C            S3C_LCDREG(0x48)    /* Video Window 0
position control register */
+
+#define S3C_VIDOSD1A            S3C_LCDREG(0x50)    /* Video Window 1
position control register */
+#define S3C_VIDOSD1B            S3C_LCDREG(0x54)    /* Video Window 1
position control register */
+#define S3C_VIDOSD1C            S3C_LCDREG(0x58)    /* Video Window 1
position control register */
+#define S3C_VIDOSD1D            S3C_LCDREG(0x5C)    /* Video Window 1
position control register */
+
+#define S3C_VIDOSD2A            S3C_LCDREG(0x60)    /* Video Window 2
position control register */
+#define S3C_VIDOSD2B            S3C_LCDREG(0x64)    /* Video Window 2
position control register */
+#define S3C_VIDOSD2C            S3C_LCDREG(0x68)    /* Video Window 2
position control register */
+#define S3C_VIDOSD2D            S3C_LCDREG(0x6C)    /* Video Window 2
position control register */
+
+#define S3C_VIDOSD3A            S3C_LCDREG(0x70)    /* Video Window 3
position control register */
+#define S3C_VIDOSD3B            S3C_LCDREG(0x74)    /* Video Window 3
position control register */
+#define S3C_VIDOSD3C            S3C_LCDREG(0x78)    /* Video Window 3
position control register */
+
+#define S3C_VIDOSD4A            S3C_LCDREG(0x80)    /* Video Window 4
position control register */
+#define S3C_VIDOSD4B            S3C_LCDREG(0x84)    /* Video Window 4
position control register */
+#define S3C_VIDOSD4C            S3C_LCDREG(0x88)    /* Video Window 4
position control register */
```

```
+
+#define S3C_VIDW00ADD2B0    S3C_LCDREG(0x94)    /* LCD CONTROL 1 */
+#define S3C_VIDW00ADD2B1    S3C_LCDREG(0x98)    /* LCD CONTROL 1 */
+
+#define S3C_VIDW00ADD0B0    S3C_LCDREG(0x0A0)  /* Window 0 buffer
start address register, buffer 0 */
+#define S3C_VIDW00ADD0B1    S3C_LCDREG(0x0A4)  /* Window 0 buffer
start address register, buffer 1 */
+#define S3C_VIDW01ADD0B0    S3C_LCDREG(0x0A8)  /* Window 1 buffer
start address register, buffer 0 */
+#define S3C_VIDW01ADD0B1    S3C_LCDREG(0x0AC)  /* Window 1 buffer
start address register, buffer 1 */
+#define S3C_VIDW02ADD0          S3C_LCDREG(0x0B0)  /* Window 2
buffer start address register */
+#define S3C_VIDW03ADD0          S3C_LCDREG(0x0B8)  /* Window 3
buffer start address register */
+#define S3C_VIDW04ADD0          S3C_LCDREG(0x0C0)  /* Window 4
buffer start address register */
+#define S3C_VIDW00ADD1B0    S3C_LCDREG(0x0D0)  /* Window 0 buffer end
address register, buffer 0 */
+#define S3C_VIDW00ADD1B1    S3C_LCDREG(0x0D4)  /* Window 0 buffer end
address register, buffer 1 */
+#define S3C_VIDW01ADD1B0    S3C_LCDREG(0x0D8)  /* Window 1 buffer end
address register, buffer 0 */
+#define S3C_VIDW01ADD1B1    S3C_LCDREG(0x0DC)  /* Window 1 buffer end
address register, buffer 1 */
+#define S3C_VIDW02ADD1          S3C_LCDREG(0x0E0)  /* Window 2
buffer end address register */
+#define S3C_VIDW03ADD1          S3C_LCDREG(0x0E8)  /* Window 3
buffer end address register */
+#define S3C_VIDW04ADD1          S3C_LCDREG(0x0F0)  /* Window 4
buffer end address register */
+#define S3C_VIDW00ADD2          S3C_LCDREG(0x100)  /* Window 0
buffer size register */
+#define S3C_VIDW01ADD2          S3C_LCDREG(0x104)  /* Window 1
buffer size register */
+
+#define S3C_VIDW02ADD2          S3C_LCDREG(0x108)  /* Window 2
buffer size register */
```

```
+#define S3C_VIDW03ADD2              S3C_LCDREG(0x10C)  /* Window 3
buffer size register */
+#define S3C_VIDW04ADD2              S3C_LCDREG(0x110)  /* Window 4
buffer size register */
+
+#define S3C_VIDINTCON0        S3C_LCDREG(0x130)  /* Indicate the Video
interrupt control register */
+#define S3C_VIDINTCON1        S3C_LCDREG(0x134)  /* Video Interrupt
Pending register */
+#define S3C_W1KEYCON0         S3C_LCDREG(0x140)  /* Color key control
register */
+#define S3C_W1KEYCON1         S3C_LCDREG(0x144)  /* Color key value
( transparent value) register */
+#define S3C_W2KEYCON0         S3C_LCDREG(0x148)  /* Color key control
register */
+#define S3C_W2KEYCON1         S3C_LCDREG(0x14C)  /* Color key value
(transparent value) register */
+
+#define S3C_W3KEYCON0         S3C_LCDREG(0x150)  /* Color key control
register      */
+#define S3C_W3KEYCON1         S3C_LCDREG(0x154)  /* Color key value
(transparent value) register  */
+#define S3C_W4KEYCON0         S3C_LCDREG(0x158)  /* Color key control
register      */
+#define S3C_W4KEYCON1         S3C_LCDREG(0x15C)  /* Color key value
(transparent value) register  */
+#define S3C_DITHMODE          S3C_LCDREG(0x170)  /* Dithering mode
register.     */
+
+#define S3C_WIN0MAP           S3C_LCDREG(0x180)  /* Window color
control       */
+#define S3C_WIN1MAP           S3C_LCDREG(0x184)  /* Window color
control       */
+#define S3C_WIN2MAP           S3C_LCDREG(0x188)  /* Window color
control       */
+#define S3C_WIN3MAP           S3C_LCDREG(0x18C)  /* Window color
control       */
+#define S3C_WIN4MAP           S3C_LCDREG(0x190)  /* Window color
control       */
```

```
+#define S3C_WPALCON             S3C_LCDREG(0x1A0)  /* Window Palette
control register   */
+
+#define S3C_TRIGCON             S3C_LCDREG(0x1A4)  /* I80 / RGB Trigger
Control Regiter   */
+#define S3C_I80IFCONA0          S3C_LCDREG(0x1B0)  /* I80 Interface control
0 for Main LDI    */
+#define S3C_I80IFCONA1          S3C_LCDREG(0x1B4)  /* I80 Interface control
0 for Sub LDI     */
+#define S3C_I80IFCONB0          S3C_LCDREG(0x1B8)  /* I80 Inteface control
1 for Main LDI    */
+#define S3C_I80IFCONB1          S3C_LCDREG(0x1BC)  /* I80 Inteface control
1 for Sub LDI     */
+#define S3C_LDI_CMDCON0             S3C_LCDREG(0x1D0)  /* I80 Interface
LDI Command Control 0       */
+#define S3C_LDI_CMDCON1             S3C_LCDREG(0x1D4)  /* I80 Interface
LDI Command Control 1       */
+#define S3C_SIFCCON0            S3C_LCDREG(0x1E0)  /* LCD i80 System
Interface Command Control 0     */
+#define S3C_SIFCCON1            S3C_LCDREG(0x1E4)  /* LCD i80 System
Interface Command Control 1     */
+#define S3C_SIFCCON2            S3C_LCDREG(0x1E8)  /* LCD i80 System
Interface Command Control 2     */
+
+#define S3C_LDI_CMD0            S3C_LCDREG(0x280)  /* I80 Inteface LDI
Command 0       */
+#define S3C_LDI_CMD1            S3C_LCDREG(0x284)  /* I80 Inteface LDI
Command 1       */
+#define S3C_LDI_CMD2            S3C_LCDREG(0x288)  /* I80 Inteface LDI
Command 2       */
+#define S3C_LDI_CMD3            S3C_LCDREG(0x28C)  /* I80 Inteface LDI
Command 3       */
+#define S3C_LDI_CMD4            S3C_LCDREG(0x290)  /* I80 Inteface LDI
Command 4       */
+#define S3C_LDI_CMD5            S3C_LCDREG(0x294)  /* I80 Inteface LDI
Command 5       */
+#define S3C_LDI_CMD6            S3C_LCDREG(0x298)  /* I80 Inteface LDI
Command 6       */
+#define S3C_LDI_CMD7            S3C_LCDREG(0x29C)  /* I80 Inteface LDI
```

```
Command 7        */
+#define S3C_LDI_CMD8          S3C_LCDREG(0x2A0)  /* I80 Inteface LDI
Command 8        */
+#define S3C_LDI_CMD9          S3C_LCDREG(0x2A4)  /* I80 Inteface LDI
Command 9        */
+#define S3C_LDI_CMD10         S3C_LCDREG(0x2A8)  /* I80 Inteface LDI
Command 10       */
+#define S3C_LDI_CMD11         S3C_LCDREG(0x2AC)  /* I80 Inteface LDI
Command 11       */
+
+#define S3C_W2PDATA01         S3C_LCDREG(0x300)  /* Window 2 Palette
Data of the Index 0,1    */
+#define S3C_W2PDATA23         S3C_LCDREG(0x304)  /* Window 2 Palette
Data of the Index 2,3    */
+#define S3C_W2PDATA45         S3C_LCDREG(0x308)  /* Window 2 Palette
Data of the Index 4,5    */
+#define S3C_W2PDATA67         S3C_LCDREG(0x30C)  /* Window 2 Palette
Data of the Index 6,7    */
+#define S3C_W2PDATA89         S3C_LCDREG(0x310)  /* Window 2 Palette
Data of the Index 8,9    */
+#define S3C_W2PDATAAB         S3C_LCDREG(0x314)  /* Window 2 Palette
Data of the Index A, B   */
+#define S3C_W2PDATACD         S3C_LCDREG(0x318)  /* Window 2 Palette
Data of the Index C, D   */
+#define S3C_W2PDATAEF         S3C_LCDREG(0x31C)  /* Window 2 Palette
Data of the Index E, F   */
+#define S3C_W3PDATA01         S3C_LCDREG(0x320)  /* Window 3 Palette
Data of the Index 0,1    */
+#define S3C_W3PDATA23         S3C_LCDREG(0x324)  /* Window 3 Palette
Data of the Index 2,3    */
+#define S3C_W3PDATA45         S3C_LCDREG(0x328)  /* Window 3 Palette
Data of the Index 4,5    */
+#define S3C_W3PDATA67         S3C_LCDREG(0x32C)  /* Window 3 Palette
Data of the Index 6,7    */
+#define S3C_W3PDATA89         S3C_LCDREG(0x330)  /* Window 3 Palette
Data of the Index 8,9    */
+#define S3C_W3PDATAAB         S3C_LCDREG(0x334)  /* Window 3 Palette
Data of the Index A, B   */
+#define S3C_W3PDATACD         S3C_LCDREG(0x338)  /* Window 3 Palette
```

```
Data of the Index C, D  */
+#define S3C_W3PDATAEF          S3C_LCDREG(0x33C)  /* Window 3 Palette
Data of the Index E, F  */
+#define S3C_W4PDATA01          S3C_LCDREG(0x340)  /* Window 3 Palette
Data of the Index 0,1   */
+#define S3C_W4PDATA23          S3C_LCDREG(0x344)  /* Window 3 Palette
Data of the Index 2,3   */
+
+
+#endif
+
+#define S3C_TFTPAL2(x)         S3C_LCDREG((0x300 + (x)*4))
+#define S3C_TFTPAL3(x)         S3C_LCDREG((0x320 + (x)*4))
+#define S3C_TFTPAL4(x)         S3C_LCDREG((0x340 + (x)*4))
+#define S3C_TFTPAL0(x)         S3C_LCDREG((0x400 + (x)*4))
+#define S3C_TFTPAL1(x)         S3C_LCDREG((0x800 + (x)*4))
+
+/*------------------------------------------------------------*/
+/* Video Main Control 0 register - VIDCON0 */
+#define S3C_VIDCON0_INTERLACE_F_PROGRESSIVE        (0<<29)
+#define S3C_VIDCON0_INTERLACE_F_INTERLACE          (1<<29)
+#define S3C_VIDCON0_INTERLACE_F_MASK               (1<<29)
+#define S3C_VIDCON0_VIDOUT(x)                      (((x)&0x7)<<26)
+#define S3C_VIDCON0_VIDOUT_RGB_IF                  (0<<26)
+#define S3C_VIDCON0_VIDOUT_TV                      (1<<26)
+#define S3C_VIDCON0_VIDOUT_I80IF0                  (2<<26)
+#define S3C_VIDCON0_VIDOUT_I80IF1                  (3<<26)
+#define S3C_VIDCON0_VIDOUT_TVNRGBIF                (4<<26)
+#define S3C_VIDCON0_VIDOUT_TVNI80IF0               (6<<26)
+#define S3C_VIDCON0_VIDOUT_TVNI80IF1               (7<<26)
+#define S3C_VIDCON0_VIDOUT_MASK                    (7<<26)
+#define S3C_VIDCON0_L1_DATA16(x)                   (((x)&0x7)<<23)
+#define S3C_VIDCON0_L1_DATA16_SUB_16_MODE          (0<<23)
+#define S3C_VIDCON0_L1_DATA16_SUB_16PLUS2_MODE     (1<<23)
+#define S3C_VIDCON0_L1_DATA16_SUB_9PLUS9_MODE      (2<<23)
```

```
+#define S3C_VIDCON0_L1_DATA16_SUB_16PLUS8_MODE		(3<<23)
+#define S3C_VIDCON0_L1_DATA16_SUB_18_MODE		(4<<23)
+#define S3C_VIDCON0_L0_DATA16(x)		(((x)&0x7)<<20)
+#define S3C_VIDCON0_L0_DATA16_MAIN_16_MODE		(0<<20)
+#define S3C_VIDCON0_L0_DATA16_MAIN_16PLUS2_MODE		(1<<20)
+#define S3C_VIDCON0_L0_DATA16_MAIN_9PLUS9_MODE		(2<<20)
+#define S3C_VIDCON0_L0_DATA16_MAIN_16PLUS8_MODE		(3<<20)
+#define S3C_VIDCON0_L0_DATA16_MAIN_18_MODE		(4<<20)
+#define S3C_VIDCON0_PNRMODE(x)		(((x)&0x3)<<17)
+#define S3C_VIDCON0_PNRMODE_RGB_P		(0<<17)
+#define S3C_VIDCON0_PNRMODE_BGR_P		(1<<17)
+#define S3C_VIDCON0_PNRMODE_RGB_S		(2<<17)
+#define S3C_VIDCON0_PNRMODE_BGR_S		(3<<17)
+#define S3C_VIDCON0_PNRMODE_MASK		(3<<17)
+#define S3C_VIDCON0_CLKVALUP_ALWAYS		(0<<16)
+#define S3C_VIDCON0_CLKVALUP_ST_FRM		(1<<16)
+#define S3C_VIDCON0_CLKVAL_F(x)		(((x)&0xFF)<<6)
+#define S3C_VIDCON0_VCLKEN_ENABLE		(1<<5)
+#define S3C_VIDCON0_CLKDIR_DIVIDED		(1<<4)
+#define S3C_VIDCON0_CLKDIR_DIRECTED		(0<<4)
+#define S3C_VIDCON0_CLKSEL(x)		(((x)&0x3)<<2)
+#define S3C_VIDCON0_CLKSEL_F_HCLK		(0<<2)
+#define S3C_VIDCON0_ENVID_ENABLE		(1 << 1)	/* 0:Disable
1:Enable LCD video output and logic immediatly */
+#define S3C_VIDCON0_ENVID_DISABLE		(0 << 1)	/* 0:Disable
1:Enable LCD video output and logic immediatly */
+#define S3C_VIDCON0_ENVID_F_ENABLE		(1 << 0)	/*
0:Dis 1:Ena wait until Current frame end. */
+#define S3C_VIDCON0_ENVID_F_DISABLE		(0 << 0)	/* 0:Dis
1:Ena wait until Current frame end. */
+
+/* Video Main Control 1 register - VIDCON1 */
+#define S3C_VIDCON1_IVCLK_FALL_EDGE		(0<<7)
+#define S3C_VIDCON1_IVCLK_RISE_EDGE		(1<<7)
+#define S3C_VIDCON1_IHSYNC_NORMAL		(0<<6)
```

```
+#define S3C_VIDCON1_IHSYNC_INVERT		(1<<6)
+#define S3C_VIDCON1_IVSYNC_NORMAL		(0<<5)
+#define S3C_VIDCON1_IVSYNC_INVERT		(1<<5)
+#define S3C_VIDCON1_IVDEN_NORMAL		(0<<4)
+#define S3C_VIDCON1_IVDEN_INVERT		(1<<4)
+
+/* Video Main Control 2 register - VIDCON2 */
+#define S3C_VIDCON2_EN601_DISABLE		(0<<23)
+#define S3C_VIDCON2_EN601_ENABLE		(1<<23)
+#define S3C_VIDCON2_EN601_MASK		(1<<23)
+#define S3C_VIDCON2_TVFORMATSEL0_HARDWARE	(0<<14)
+#define S3C_VIDCON2_TVFORMATSEL0_SOFTWARE	(1<<14)
+#define S3C_VIDCON2_TVFORMATSEL0_MASK	(1<<14)
+#define S3C_VIDCON2_TVFORMATSEL1_RGB	(0<<12)
+#define S3C_VIDCON2_TVFORMATSEL1_YUV422		(1<<12)
+#define S3C_VIDCON2_TVFORMATSEL1_YUV444		(2<<12)
+#define S3C_VIDCON2_TVFORMATSEL1_MASK	(0x3<<12)
+#define S3C_VIDCON2_ORGYUV_YCBCR		(0<<8)
+#define S3C_VIDCON2_ORGYUV_CBCRY		(1<<8)
+#define S3C_VIDCON2_ORGYUV_MASK		(1<<8)
+#define S3C_VIDCON2_YUVORD_CBCR		(0<<7)
+#define S3C_VIDCON2_YUVORD_CRCB		(1<<7)
+#define S3C_VIDCON2_YUVORD_MASK		(1<<7)
+
+/* VIDEO Time Control 0 register - VIDTCON0 */
+#define S3C_VIDTCON0_VBPDE(x)		(((x)&0xFF)<<24)
+#define S3C_VIDTCON0_VBPD(x)		(((x)&0xFF)<<16)
+#define S3C_VIDTCON0_VFPD(x)		(((x)&0xFF)<<8)
+#define S3C_VIDTCON0_VSPW(x)		(((x)&0xFF)<<0)
+
+/* VIDEO Time Control 1 register - VIDTCON1 */
+#define S3C_VIDTCON1_VFPDE(x)		(((x)&0xFF)<<24)
+#define S3C_VIDTCON1_HBPD(x)		(((x)&0xFF)<<16)
+#define S3C_VIDTCON1_HFPD(x)		(((x)&0xFF)<<8)
```

```
+#define S3C_VIDTCON1_HSPW(x)                    (((x)&0xFF)<<0)
+
+/* VIDEO Time Control 2 register - VIDTCON2 */
+#define S3C_VIDTCON2_LINEVAL(x)                 (((x)&0x7FF)<<11) /* these
bits determine the vertical size of lcd panel */
+#define S3C_VIDTCON2_HOZVAL(x)                  (((x)&0x7FF)<<0) /* these
bits determine the horizontal size of lcd panel*/
+
+
+/* Window 0~4 Control register - WINCONx */
+#define S3C_WINCONx_WIDE_NARROW(x)              (((x)&0x3)<<26)
+#define S3C_WINCONx_ENLOCAL_DMA                 (0<<22)
+#define S3C_WINCONx_ENLOCAL                     (1<<22)
+#define S3C_WINCONx_ENLOCAL_MASK                (1<<22)
+#define S3C_WINCONx_BUFSEL_0                    (0<<20)
+#define S3C_WINCONx_BUFSEL_1                    (1<<20)
+#define S3C_WINCONx_BUFSEL_MASK                 (1<<20)
+#define S3C_WINCONx_BUFAUTOEN_DISABLE           (0<<19)
+#define S3C_WINCONx_BUFAUTOEN_ENABLE            (1<<19)
+#define S3C_WINCONx_BUFAUTOEN_MASK              (1<<19)
+#define S3C_WINCONx_BITSWP_DISABLE              (0<<18)
+#define S3C_WINCONx_BITSWP_ENABLE               (1<<18)
+#define S3C_WINCONx_BYTSWP_DISABLE              (0<<17)
+#define S3C_WINCONx_BYTSWP_ENABLE               (1<<17)
+#define S3C_WINCONx_HAWSWP_DISABLE              (0<<16)
+#define S3C_WINCONx_HAWSWP_ENABLE               (1<<16)
+#define S3C_WINCONx_WSWP_DISABLE                (0<<15)
+#define S3C_WINCONx_WSWP_ENABLE                 (1<<15)
+#define S3C_WINCONx_INRGB_RGB                   (0<<13)
+#define S3C_WINCONx_INRGB_YUV                   (1<<13)
+#define S3C_WINCONx_INRGB_MASK                  (1<<13)
+#define S3C_WINCONx_BURSTLEN_16WORD             (0<<9)
+#define S3C_WINCONx_BURSTLEN_8WORD              (1<<9)
+#define S3C_WINCONx_BURSTLEN_4WORD              (2<<9)
```

```
+#define S3C_WINCONx_BLD_PIX_PLANE		(0<<6)
+#define S3C_WINCONx_BLD_PIX_PIXEL		(1<<6)
+#define S3C_WINCONx_BLD_PIX_MASK		(1<<6)
+#define S3C_WINCONx_BPPMODE_F_1BPP		(0<<2)
+#define S3C_WINCONx_BPPMODE_F_2BPP		(1<<2)
+#define S3C_WINCONx_BPPMODE_F_4BPP		(2<<2)
+#define S3C_WINCONx_BPPMODE_F_8BPP_PAL		(3<<2)
+#define S3C_WINCONx_BPPMODE_F_8BPP_NOPAL	(4<<2)
+#define S3C_WINCONx_BPPMODE_F_16BPP_565		(5<<2)
+#define S3C_WINCONx_BPPMODE_F_16BPP_A555	(6<<2)
+#define S3C_WINCONx_BPPMODE_F_18BPP_666		(8<<2)
+#define S3C_WINCONx_BPPMODE_F_24BPP_888		(11<<2)
+#define S3C_WINCONx_BPPMODE_F_24BPP_A887	(0xc<<2)
+#define S3C_WINCONx_BPPMODE_F_25BPP_A888	(0xd<<2)
+#define S3C_WINCONx_BPPMODE_F_28BPP_A888	(0xd<<2)
+#define S3C_WINCONx_BPPMODE_F_MASK		(0xf<<2)
+#define S3C_WINCONx_ALPHA_SEL_0			(0<<1)
+#define S3C_WINCONx_ALPHA_SEL_1			(1<<1)
+#define S3C_WINCONx_ALPHA_SEL_MASK		(1<<1)
+#define S3C_WINCONx_ENWIN_F_DISABLE		(0<<0)
+#define S3C_WINCONx_ENWIN_F_ENABLE		(1<<0)
+
+/* Window 1-2 Control register - WINCON1 */
+#define S3C_WINCON1_LOCALSEL_TV			(0<<23)
+#define S3C_WINCON1_LOCALSEL_CAMERA		(1<<23)
+#define S3C_WINCON1_LOCALSEL_MASK		(1<<23)
+#define S3C_WINCON2_LOCALSEL_TV			(0<<23)
+#define S3C_WINCON2_LOCALSEL_CAMERA		(1<<23)
+#define S3C_WINCON2_LOCALSEL_MASK		(1<<23)
+
+/* Window 0~4 Position Control A register - VIDOSDxA */
+#define S3C_VIDOSDxA_OSD_LTX_F(x)		(((x)&0x7FF)<<11)
+#define S3C_VIDOSDxA_OSD_LTY_F(x)		(((x)&0x7FF)<<0)
+
```

```
+/* Window 0~4 Position Control B register - VIDOSDxB */
+#define S3C_VIDOSDxB_OSD_RBX_F(x)                ((x)&0x7FF)<<11)
+#define S3C_VIDOSDxB_OSD_RBY_F(x)                ((x)&0x7FF)<<0)
+
+/* Window 0 Position Control C register - VIDOSD0C */
+#define  S3C_VIDOSD0C_OSDSIZE(x)             ((x)&0xFFFFFF)<<0)
+
+/* Window 1~4 Position Control C register - VIDOSDxC */
+#define S3C_VIDOSDxC_ALPHA0_R(x)                 ((x)&0xF)<<20)
+#define S3C_VIDOSDxC_ALPHA0_G(x)                 ((x)&0xF)<<16)
+#define S3C_VIDOSDxC_ALPHA0_B(x)             ((x)&0xF)<<12)
+#define S3C_VIDOSDxC_ALPHA1_R(x)                 ((x)&0xF)<<8)
+#define S3C_VIDOSDxC_ALPHA1_G(x)                 ((x)&0xF)<<4)
+#define S3C_VIDOSDxC_ALPHA1_B(x)             ((x)&0xF)<<0)
+
+/* Window 1~2 Position Control D register - VIDOSDxD */
+#define  S3C_VIDOSDxD_OSDSIZE(x)             ((x)&0xFFFFFF)<<0)
+
+/* Frame buffer Start Address register - VIDWxxADD0 */
+#define S3C_VIDWxxADD0_VBANK_F(x)                ((x)&0xFF)<<23) /*
the end address of the LCD frame buffer. */
+#define S3C_VIDWxxADD0_VBASEU_F(x)
        ((x)&0xFFFFFF)<<0) /* Virtual screen offset size (the number of byte). */
+
+/* Frame buffer End Address register - VIDWxxADD1 */
+#define S3C_VIDWxxADD1_VBASEL_F(x)               ((x)&0xFFFFFF)<<0)
/* the end address of the LCD frame buffer. */
+
+/* Frame buffer Size register - VIDWxxADD2 */
+#define S3C_VIDWxxADD2_OFFSIZE_F(x)              ((x)&0x1FFF)<<13) /*
Virtual screen offset size (the number of byte). */
+#define S3C_VIDWxxADD2_PAGEWIDTH_F(x)
        ((x)&0x1FFF)<<0) /* Virtual screen page width (the number of byte). */
+
+/* VIDEO Interrupt Control 0 register - VIDINTCON0 */
```

```
+#define S3C_VIDINTCON0_FIFOINTERVAL(x)
        (((x)&0x3F)<<20)
+#define S3C_VIDINTCON0_SYSMAINCON_DISABLE          (0<<19)
+#define S3C_VIDINTCON0_SYSMAINCON_ENABLE           (1<<19)
+#define S3C_VIDINTCON0_SYSSUBCON_DISABLE           (0<<18)
+#define S3C_VIDINTCON0_SYSSUBCON_ENABLE                  (1<<18)
+#define S3C_VIDINTCON0_SYSIFDONE_DISABLE           (0<<17)
+#define S3C_VIDINTCON0_SYSIFDONE_ENABLE            (1<<17)
+#define S3C_VIDINTCON0_FRAMESEL0_BACK              (0<<15)
+#define S3C_VIDINTCON0_FRAMESEL0_VSYNC             (1<<15)
+#define S3C_VIDINTCON0_FRAMESEL0_ACTIVE            (2<<15)
+#define S3C_VIDINTCON0_FRAMESEL0_FRONT             (3<<15)
+#define S3C_VIDINTCON0_FRAMESEL0_MASK              (3<<15)
+#define S3C_VIDINTCON0_FRAMESEL1_NONE              (0<<13)
+#define S3C_VIDINTCON0_FRAMESEL1_BACK              (1<<13)
+#define S3C_VIDINTCON0_FRAMESEL1_VSYNC             (2<<13)
+#define S3C_VIDINTCON0_FRAMESEL1_FRONT             (3<<13)
+#define S3C_VIDINTCON0_INTFRMEN_DISABLE            (0<<12)
+#define S3C_VIDINTCON0_INTFRMEN_ENABLE             (1<<12)
+#define S3C_VIDINTCON0_FRAMEINT_MASK                     (0x1F<<12)
+#define S3C_VIDINTCON0_FIFOSEL_WIN4              (1<<11)
+#define S3C_VIDINTCON0_FIFOSEL_WIN3              (1<<10)
+#define S3C_VIDINTCON0_FIFOSEL_WIN2              (1<<9)
+#define S3C_VIDINTCON0_FIFOSEL_WIN1              (1<<6)
+#define S3C_VIDINTCON0_FIFOSEL_WIN0              (1<<5)
+#define S3C_VIDINTCON0_FIFOSEL_ALL               (0x73<<5)
+#define S3C_VIDINTCON0_FIFOLEVEL_25              (0<<2)
+#define S3C_VIDINTCON0_FIFOLEVEL_50              (1<<2)
+#define S3C_VIDINTCON0_FIFOLEVEL_75              (2<<2)
+#define S3C_VIDINTCON0_FIFOLEVEL_EMPTY                   (3<<2)
+#define S3C_VIDINTCON0_FIFOLEVEL_FULL                    (4<<2)
+#define S3C_VIDINTCON0_INTFIFOEN_DISABLE         (0<<1)
+#define S3C_VIDINTCON0_INTFIFOEN_ENABLE                  (1<<1)
+#define S3C_VIDINTCON0_INTEN_DISABLE                     (0<<0)
```

```
+#define S3C_VIDINTCON0_INTEN_ENABLE		(1<<0)
+#define S3C_VIDINTCON0_INTEN_MASK		(1<<0)
+
+/* VIDEO Interrupt Control 1 register - VIDINTCON1 */
+#define S3C_VIDINTCON1_INTI80PEND		(0<<2)
+#define S3C_VIDINTCON1_INTFRMPEND		(1<<1)
+#define S3C_VIDINTCON1_INTFIFOPEND		(1<<0)
+
+/* WIN 1~4 Color Key 0 register - WxKEYCON0 */
+#define S3C_WxKEYCON0_KEYBLEN_DISABLE		(0<<26)
+#define S3C_WxKEYCON0_KEYBLEN_ENABLE		(1<<26)
+#define S3C_WxKEYCON0_KEYEN_F_DISABLE		(0<<25)
+#define S3C_WxKEYCON0_KEYEN_F_ENABLE		(1<<25)
+#define S3C_WxKEYCON0_DIRCON_MATCH_FG_IMAGE	(0<<24)
+#define S3C_WxKEYCON0_DIRCON_MATCH_BG_IMAGE	(1<<24)
+#define S3C_WxKEYCON0_COMPKEY(x)		(((x)&0xFFFFFF)<<0)
+
+/* WIN 1~4 Color Key 1 register - WxKEYCON1 */
+#define S3C_WxKEYCON1_COLVAL(x)			(((x)&0xFFFFFF)<<0)
+
+/* Dithering Control 1 register - DITHMODE */
+#define S3C_DITHMODE_RDITHPOS_8BIT		(0<<5)
+#define S3C_DITHMODE_RDITHPOS_6BIT		(1<<5)
+#define S3C_DITHMODE_RDITHPOS_5BIT		(2<<5)
+#define S3C_DITHMODE_GDITHPOS_8BIT		(0<<3)
+#define S3C_DITHMODE_GDITHPOS_6BIT		(1<<3)
+#define S3C_DITHMODE_GDITHPOS_5BIT		(2<<3)
+#define S3C_DITHMODE_BDITHPOS_8BIT		(0<<1)
+#define S3C_DITHMODE_BDITHPOS_6BIT		(1<<1)
+#define S3C_DITHMODE_BDITHPOS_5BIT		(2<<1)
+#define S3C_DITHMODE_RGB_DITHPOS_MASK		(0x3f<<1)
+#define S3C_DITHMODE_DITHERING_DISABLE		(0<<0)
+#define S3C_DITHMODE_DITHERING_ENABLE		(1<<0)
+#define S3C_DITHMODE_DITHERING_MASK		(1<<0)
```

```
+
+/* Window 0~4 Color map register - WINxMAP */
+#define S3C_WINxMAP_MAPCOLEN_F_ENABLE		(1<<24)
+#define S3C_WINxMAP_MAPCOLEN_F_DISABLE		(0<<24)
+#define S3C_WINxMAP_MAPCOLOR			(((x)&0xFFFFFF)<<0)
+
+/* Window Palette Control register - WPALCON */
+#define S3C_WPALCON_PALUPDATEEN			(1<<9)
+#define S3C_WPALCON_W4PAL_16BIT_A		(1<<8)		/*
A:5:5:5 */
+#define S3C_WPALCON_W4PAL_16BIT			(0<<8)
	/*  5:6:5 */
+#define S3C_WPALCON_W3PAL_16BIT_A		(1<<7)		/*
A:5:5:5 */
+#define S3C_WPALCON_W3PAL_16BIT			(0<<7)
	/*  5:6:5 */
+#define S3C_WPALCON_W2PAL_16BIT_A		(1<<6)		/*
A:5:5:5 */
+#define S3C_WPALCON_W2PAL_16BIT			(0<<6)
	/*  5:6:5 */
+#define S3C_WPALCON_W1PAL_25BIT_A		(0<<3)		/*
A:8:8:8 */
+#define S3C_WPALCON_W1PAL_24BIT			(1<<3)
	/*  8:8:8 */
+#define S3C_WPALCON_W1PAL_19BIT_A		(2<<3)		/*
A:6:6:6 */
+#define S3C_WPALCON_W1PAL_18BIT_A		(3<<3)		/*
A:6:6:5 */
+#define S3C_WPALCON_W1PAL_18BIT			(4<<3)
	/*  6:6:6 */
+#define S3C_WPALCON_W1PAL_16BIT_A		(5<<3)		/*
A:5:5:5 */
+#define S3C_WPALCON_W1PAL_16BIT			(6<<3)
	/*  5:6:5 */
+#define S3C_WPALCON_W0PAL_25BIT_A		(0<<0)		/*
A:8:8:8 */
+#define S3C_WPALCON_W0PAL_24BIT			(1<<0)
	/*  8:8:8 */
```

```
+#define S3C_WPALCON_W0PAL_19BIT_A                (2<<0)          /*
A:6:6:6 */
+#define S3C_WPALCON_W0PAL_18BIT_A                (3<<0)          /*
A:6:6:5 */
+#define S3C_WPALCON_W0PAL_18BIT                      (4<<0)
        /*  6:6:6 */
+#define S3C_WPALCON_W0PAL_16BIT_A                (5<<0)          /*
A:5:5:5 */
+#define S3C_WPALCON_W0PAL_16BIT                      (6<<0)
        /*  5:6:5 */
+
+/* I80/RGB Trigger Control register - TRIGCON */
+#define S3C_TRIGCON_SWFRSTATUS_REQUESTED         (1<<2)
+#define S3C_TRIGCON_SWFRSTATUS_NOT_REQUESTED           (0<<2)
+#define S3C_TRIGCON_SWTRGCMD                     (1<<1)
+#define S3C_TRIGCON_TRGMODE_ENABLE                   (1<<0)
+#define S3C_TRIGCON_TRGMODE_DISABLE                  (0<<0)
+
+/* LCD I80 Interface Control 0 register - I80IFCONA0 */
+#define S3C_I80IFCONAx_LCD_CS_SETUP(x)               (((x)&0xF)<<16)
+#define S3C_I80IFCONAx_LCD_WR_SETUP(x)               (((x)&0xF)<<12)
+#define S3C_I80IFCONAx_LCD_WR_ACT(x)             (((x)&0xF)<<8)
+#define S3C_I80IFCONAx_LCD_WR_HOLD(x)                (((x)&0xF)<<4)
+
+
+/*************************************************************************/
+/*HOST IF registers */
+/* Host I/F A - */
+
+/* Host I/F B - Modem I/F */
+#define S3C64XX_HOSTIFB_ADDR                 0x74100000
+#define S3C_HOSTIFBREG(x)                    __REG((x) +
S3C64XX_HOSTIFB_ADDR)
+
+/*LCD 特殊寄存器*/
```

```
+#define MIFPCON_REG                 __REG(0x7410800C)
+  #define  SEL_BYPASS_MASK          0x00000008          // 为 LCD 旁路选择控制
模式
+#define SPCON_REG                   __REG(ELFIN_GPIO_BASE +
SPCON_OFFSET)
+  #define LCD_SEL_MASK              0x00000003          //LCD 管脚输出格式
+  #define RGB_IF_STYLE_MASK         0x00000001          //RGB 格式
+
+#define S3C_HOSTIFB_INT2AP                  S3C_HOSTIFBREG(0x8000)
+#define S3C_HOSTIFB_INT2MSM
        S3C_HOSTIFBREG(0x8004)
+#define S3C_HOSTIFB_MIFCON                  S3C_HOSTIFBREG(0x8008)
+#define S3C_HOSTIFB_MIFPCON
        S3C_HOSTIFBREG(0x800C)
+#define S3C_HOSTIFB_MSMINTCLR
        S3C_HOSTIFBREG(0x8010)
+
+#define S3C_HOSTIFB_MIFCON_INT2MSM_DIS          (0x0<<3)
+#define S3C_HOSTIFB_MIFCON_INT2MSM_EN           (0x1<<3)
+#define S3C_HOSTIFB_MIFCON_INT2AP_DIS           (0x0<<2)
+#define S3C_HOSTIFB_MIFCON_INT2AP_EN            (0x1<<2)
+#define S3C_HOSTIFB_MIFCON_WAKEUP_DIS           (0x0<<1)
+#define S3C_HOSTIFB_MIFCON_WAKEUP_EN            (0x1<<1)
+
+#define S3C_HOSTIFB_MIFPCON_SEL_VSYNC_DIR_OUT       (0x0<<5)
+#define S3C_HOSTIFB_MIFPCON_SEL_VSYNC_DIR_IN        (0x1<<5)
+#define S3C_HOSTIFB_MIFPCON_INT2M_LEVEL_DIS         (0x0<<4)
+#define S3C_HOSTIFB_MIFPCON_INT2M_LEVEL_EN          (0x1<<4)
+#define S3C_HOSTIFB_MIFPCON_SEL_NORMAL              (0x0<<3)
+#define S3C_HOSTIFB_MIFPCON_SEL_BYPASS              (0x1<<3)
+
+#define S3C_HOSTIFB_MIFPCON_SEL_RS0             0
+#define S3C_HOSTIFB_MIFPCON_SEL_RS1             1
+#define S3C_HOSTIFB_MIFPCON_SEL_RS2             2
```

```
+#define S3C_HOSTIFB_MIFPCON_SEL_RS3		3
+#define S3C_HOSTIFB_MIFPCON_SEL_RS4		4
+#define S3C_HOSTIFB_MIFPCON_SEL_RS5		5
+#define S3C_HOSTIFB_MIFPCON_SEL_RS6		6
+
+#define S3C_WINCONx_ENLOCAL_POST		(1<<22)
+#endif

------------------------------ include/s3cfb_LCD.h ------------------------------
new file mode 100644
index 0000000..e105615
@@ -0,0 +1,39 @@
+#ifndef S3C6410_LCD_H
+#define S3C6410_LCD_H
+
+
+#include "s3cfb-RegLCD.h"
+
+#define CFG_HIGH  1
+#define CFG_LOW   0
+
+//1376 //1178
+#define S3CFB_HFP		2	/* front porch */
+#define S3CFB_HSW		41	/* hsync width */
+#define S3CFB_HBP		2	/* back porch */
+//805 //807
+#define S3CFB_VFP		2	/* front porch */
+#define S3CFB_VSW		10	/* vsync width */
+#define S3CFB_VBP		2	/* back porch */
+
+#define S3CFB_HRES		480	/* horizon pixel  x resolition */
+#define S3CFB_VRES		272	/* line cnt       y resolution */
+#define S3CFB_VFRAME_FREQ	60	/* frame rate freq */
+#define PIXELBITS		16
```

```
+
+#define S3CFB_IVCLK            CFG_LOW
+#define S3CFB_IHSYNC           CFG_HIGH
+#define S3CFB_IVSYNC           CFG_HIGH
+#define S3CFB_IVDEN            CFG_LOW
+
+
+#define S3CFB_HRES_VIRTUAL (S3CFB_HRES)   /* horizon pixel  x resolition
*/
+#define S3CFB_VRES_VIRTUAL  (S3CFB_VRES*2)/* line cnt       y resolution */
+
+#define S3CFB_HRES_OSD                (S3CFB_HRES)   /* horizon pixel  x
resolition */
+#define S3CFB_VRES_OSD                (S3CFB_VRES)   /* line cnt       y
resolution */
+#define S3CFB_PIXEL_CLOCK   (S3CFB_VFRAME_FREQ * (S3CFB_HFP +
S3CFB_HSW + S3CFB_HBP + S3CFB_HRES) * (S3CFB_VFP + S3CFB_VSW +
S3CFB_VBP + S3CFB_VRES))
+//malloc 和 uboot 最多只能分配 512K 空间
+#define LCD_FRAMEBUFFER        (CONFIG_SYS_TEXT_BASE - 0x300000)
+
+#endif
```

Chapter 7. Boot from SD

Modified This file:

```
---------- arch/arm/include/asm/arch-s3c64xx/s3c6410.h ----------
index 06d14bb..e65175e 100644
@@ -961,6 +961,43 @@ enum s3c64xx_uarts_nr {
 #define S3C_OTG_HCTSIZ0         (USBOTG_LINK_BASE + 0x510)    /* Host
Channel-0 Transfer Size */
 #define S3C_OTG_HCDMA0                (USBOTG_LINK_BASE + 0x514)    /*
Host Channel-0 DMA Address */


+/*
+ * HS MMC Interface
```

```
+ */
+#define ELFIN_HSMMC_BASE   0x7C200000
+
+#define HM_SYSAD          (0x00)
+#define HM_BLKSIZE              (0x04)
+#define HM_BLKCNT               (0x06)
+#define HM_ARGUMENT             (0x08)
+#define HM_TRNMOD               (0x0c)
+#define HM_CMDREG               (0x0e)
+#define HM_RSPREG0              (0x10)
+#define HM_RSPREG1              (0x14)
+#define HM_RSPREG2              (0x18)
+#define HM_RSPREG3              (0x1c)
+#define HM_BDATA          (0x20)
+#define HM_PRNSTS               (0x24)
+#define HM_HOSTCTL              (0x28)
+#define HM_PWRCON               (0x29)
+#define HM_BLKGAP               (0x2a)
+#define HM_WAKCON               (0x2b)
+#define HM_CLKCON               (0x2c)
+#define HM_TIMEOUTCON                (0x2e)
+#define HM_SWRST          (0x2f)
+#define HM_NORINTSTS            (0x30)
+#define HM_ERRINTSTS            (0x32)
+#define HM_NORINTSTSEN               (0x34)
+#define HM_ERRINTSTSEN               (0x36)
+#define HM_NORINTSIGEN               (0x38)
+#define HM_ERRINTSIGEN               (0x3a)
+#define HM_ACMD12ERRSTS              (0x3c)
+#define HM_CAPAREG              (0x40)
+#define HM_MAXCURR              (0x48)
+#define HM_CONTROL2             (0x80)
+#define HM_CONTROL3             (0x84)
+#define HM_CONTROL4             (0x8c)
```

```
+#define HM_HCVER          (0xfe)

 /* Device Global Registers */
 #define S3C_OTG_DCFG              (USBOTG_LINK_BASE + 0x800)    /* Device
Configuration */
```

include/configs/smdk6410.h

```
index d1f7daa..0c4ad00 100644
@@ -305,8 +305,9 @@
 #define CONFIG_NAND_S3C64XX
 /* Unimplemented or unsupported. See comment above.
 #define CONFIG_ONENAND
-#define CONFIG_MOVINAND
 */
+#define CONFIG_MOVINAND
+

 /* Settings as above boot configuration */
 #define CONFIG_ENV_IS_IN_NAND
```

include/movi.h

```
new file mode 100644
index 0000000..96ceea1
@@ -0,0 +1,99 @@
+#ifndef __MOVI_H__
+#define __MOVI_H__
+
+#define HSMMC_CHANNEL               0
+#define MOVI_INIT_REQUIRED  0
+
+#if defined(CONFIG_S3C6400) || defined(CONFIG_S3C6410) ||
defined(CONFIG_S3C6430)
+#define    TCM_BASE         0x0C004000
```

```
+#define BL2_BASE            0x57E00000
+#elif defined(CONFIG_S3C2450) || defined(CONFIG_S3C2416)
+#define    TCM_BASE         0x40004000
+#define BL2_BASE            0x33E00000
+#else
+# error TCM_BASE or BL2_BASE is not defined
+#endif
+
+/* TCM function for bl2 load */
+#if defined(CONFIG_S3C6400)
+#define CopyMovitoMem(a,b,c,d,e)      (((int(*)(uint, ushort, uint *, uint, int))
(*((uint *)(TCM_BASE + 0x8))))(a,b,c,d,e))
+#elif defined(CONFIG_S3C6410) || defined(CONFIG_S3C6430)
+#define CopyMovitoMem(a,b,c,d,e)      (((int(*)(int, uint, ushort, uint *, int))
(*((uint *)(TCM_BASE + 0x8))))(a,b,c,d,e))
+#elif defined(CONFIG_S3C2450) || defined(CONFIG_S3C2416)
+#define CopyMovitoMem(a,b,c,d)                (((int(*)(uint, ushort, uint *, int))
(*((uint *)(TCM_BASE + 0x8))))(a,b,c,d))
+#endif
+
+/* size information */
+#if defined(CONFIG_S3C6400)
+#define SS_SIZE                     (4 * 1024)
+#define eFUSE_SIZE          (2 * 1024)   // 1.5k eFuse, 0.5k reserved
+#else
+#define SS_SIZE                     (8 * 1024)
+#define eFUSE_SIZE          (1 * 1024)   // 0.5k eFuse, 0.5k reserved`
+#endif
+
+/* movinand definitions */
+#define MOVI_BLKSIZE              512
+
+//#ifdef CONFIG_BOOT_MOVINAND
+#define MOVI_TOTAL_BLKCNT  *((volatile unsigned int*)(TCM_BASE - 0x4))
+#define MOVI_HIGH_CAPACITY *((volatile unsigned int*)(TCM_BASE - 0x8))
```

```
+//#else
+//#define MOVI_TOTAL_BLKCNT7864320 // 7864320 // 3995648 // 1003520 /* static movinand total block count: for writing to movinand when nand boot */
+//#define MOVI_HIGH_CAPACITY	0
+//#endif
+
+/* partition information */
+#define PART_SIZE_BL		(512 * 1024)
+#define PART_SIZE_KERNEL	(4 * 1024 * 1024)
+#define PART_SIZE_ROOTFS	(8 * 1024 * 1024)
+
+#define MOVI_LAST_BLKPOS	(MOVI_TOTAL_BLKCNT - (eFUSE_SIZE / MOVI_BLKSIZE))
+#define MOVI_BL1_BLKCNT		(SS_SIZE / MOVI_BLKSIZE)
+#define MOVI_ENV_BLKCNT		(CONFIG_ENV_SIZE / MOVI_BLKSIZE)
+#define MOVI_BL2_BLKCNT		(PART_SIZE_BL / MOVI_BLKSIZE)
+#define MOVI_ZIMAGE_BLKCNT	(PART_SIZE_KERNEL / MOVI_BLKSIZE)
+#define MOVI_BL2_POS		(MOVI_LAST_BLKPOS - MOVI_BL1_BLKCNT - MOVI_BL2_BLKCNT- MOVI_ENV_BLKCNT)//
+#define MOVI_ROOTFS_BLKCNT	(PART_SIZE_ROOTFS / MOVI_BLKSIZE)
+
+struct movi_offset_t {
+	uint	last;
+	uint	bl1;
+	uint	env;
+	uint	bl2;
+	uint	kernel;
+	uint	rootfs;
+};
+
+/* external functions */
+
+extern void hsmmc_set_gpio(void);
+extern void hsmmc_reset (void);
+extern int hsmmc_init (void);
```

```
+extern int movi_init(void);
+extern void movi_set_capacity(void);
+extern int movi_set_ofs(uint last);
+extern void movi_write (uint addr, uint start_blk, uint blknum);
+extern void movi_read (uint addr, uint start_blk, uint blknum);
+extern void movi_write_env(ulong addr);
+extern void movi_read_env(ulong addr);
+
+#if defined(CONFIG_S3C2450)
+extern ulong virt_to_phy_smdk2450(ulong addr);
+#elif defined(CONFIG_S3C6400)
+extern ulong virt_to_phy_smdk6400(ulong addr);
+#elif defined(CONFIG_S3C6410)
+extern ulong virt_to_phy_ok6410(ulong addr);
+#elif defined(CONFIG_S3C6430)
+extern ulong virt_to_phy_smdk6430(ulong addr);
+#elif defined(CONFIG_S3C2416)
+extern ulong virt_to_phy_smdk2416(ulong addr);
+#endif
+
+extern void test_hsmmc (uint width, uint test, uint start_blk, uint blknum);
+
+/* external variables */
+extern uint movi_hc;
+extern struct movi_offset_t ofsinfo;
+
+#endif /*__MOVI_H__*/
```

nand_spl/board/samsung/smdk6410/Makefile

```
index f95e307..0c11bdb 100644
@@ -38,7 +38,7 @@ AFLAGS        += -DCONFIG_NAND_SPL
 CFLAGS    += -DCONFIG_NAND_SPL -ffunction-sections
```

```
 SOBJS       = start.o cpu_init.o lowlevel_init.o
-COBJS       = nand_boot.o nand_ecc.o s3c64xx.o smdk6410_nand_spl.o
nand_base.o
+COBJS      = nand_boot.o nand_ecc.o s3c64xx.o smdk6410_nand_spl.o
nand_base.o movi_boot.o

 SRCS        := $(addprefix $(obj),$(SOBJS:.o=.S) $(COBJS:.o=.c))
 OBJS        := $(addprefix $(obj),$(SOBJS) $(COBJS))
@@ -79,12 +79,13 @@ $(obj)cpu_init.S:
 $(obj)lowlevel_init.S:
        @rm -f $@
        @ln -s $(TOPDIR)/board/samsung/smdk6410/lowlevel_init.S $@
-
 # from nand_spl directory
 $(obj)nand_boot.c:
        @rm -f $@
        @ln -s $(TOPDIR)/nand_spl/nand_boot.c $@
-
+$(obj)movi_boot.c:
+       @rm -f $@
+       @ln -s $(TOPDIR)/nand_spl/movi_boot.c $@
 # from drivers/mtd/nand directory
 $(obj)nand_ecc.c:
        @rm -f $@

---------------------------- nand_spl/movi_boot.c ----------------------------
new file mode 100644
index 0000000..3877ff0
@@ -0,0 +1,81 @@
+#include <common.h>
+
+#ifdef CONFIG_MOVINAND
+
+#if defined(CONFIG_SMDK6410)
```

```
+#include <asm/arch/s3c6410.h>
+#endif
+#include <nand.h>
+#include <movi.h>
+#include <asm/io.h>
+
+/*
+uint movi_hc = 0;
+
+void movi_set_capacity(void)
+{
+#if defined(CONFIG_S3C6400)
+	if (MOVI_HIGH_CAPACITY == 2)
+#else
+	if (MOVI_HIGH_CAPACITY & 0x1)
+#endif
+		movi_hc = 1;
+}
+
+int movi_set_ofs(uint last)
+{
+	int changed = 0;
+
+	if (ofsinfo.last != last) {
+		ofsinfo.last = last - (eFUSE_SIZE / MOVI_BLKSIZE);
+		ofsinfo.bl1  = ofsinfo.last - MOVI_BL1_BLKCNT;
+		ofsinfo.env = ofsinfo.bl1 - MOVI_ENV_BLKCNT;
+		ofsinfo.bl2  = ofsinfo.bl1 - (MOVI_BL2_BLKCNT +
MOVI_ENV_BLKCNT);
+		ofsinfo.kernel	= ofsinfo.bl2 - MOVI_ZIMAGE_BLKCNT;
+		ofsinfo.rootfs	= ofsinfo.kernel - MOVI_ROOTFS_BLKCNT;
+		changed = 1;
+	}
+
```

```
+	return changed;
+}
+
+int movi_init(void)
+{
+	hsmmc_set_gpio();
+	hsmmc_reset();
+	if (hsmmc_init()) {
+		printf("\nCard Initialization failed.\n");
+		return -1;
+	}
+	return 1;
+}
+
+void movi_write_env(ulong addr)
+{
+	movi_write((uint)addr, ofsinfo.env, MOVI_ENV_BLKCNT);
+}
+
+void movi_read_env(ulong addr)
+{
+	movi_read((uint)addr, ofsinfo.env, MOVI_ENV_BLKCNT);
+}
+*/
+void movi_bl2_copy(void)
+{
+  //__attribute__((noreturn)) void (*uboot)(void);
+  //writel(0x0023,0x7f008824);
+
+
+#if defined(CONFIG_S3C6400)
+	CopyMovitoMem(MOVI_BL2_POS, MOVI_BL2_BLKCNT, (uint *)BL2_BASE,
CONFIG_SYS_CLK_FREQ, MOVI_INIT_REQUIRED);
+#else
```

```
+       writel(readl(HM_CONTROL4) | (0x3 << 16), HM_CONTROL4);
+       CopyMovitoMem(HSMMC_CHANNEL, MOVI_BL2_POS,
MOVI_BL2_BLKCNT, (uint *)BL2_BASE, MOVI_INIT_REQUIRED);
+#endif
+
+/* Jump to U-Boot image */
+       //uboot = (void *)(CONFIG_SYS_PHY_UBOOT_BASE);
+       //(*uboot)();
+}
+
+#endif
```

Part 2. Linux 3.9-rc6

Chapter 8.Nand flash driver & NFS configuration.

Conclusion:We modified there files .

```
$ git status
# On branch develop
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   Makefile
#       modified:   arch/arm/mach-s3c64xx/Kconfig
#       modified:   arch/arm/mach-s3c64xx/Makefile
#       new file:   arch/arm/mach-s3c64xx/mach-ok6410.c
#       new file:   arch/arm/plat-samsung/include/plat/nand.h
#       modified:   arch/arm/plat-samsung/include/plat/regs-nand.h
#       modified:   arch/arm/tools/mach-types
#       modified:   drivers/mtd/nand/Kconfig
#       modified:   drivers/mtd/nand/Makefile
#       modified:   drivers/mtd/nand/nand_base.c
#       new file:   drivers/mtd/nand/s3c_nand.c
#       modified:   scripts/Makefile.lib
```