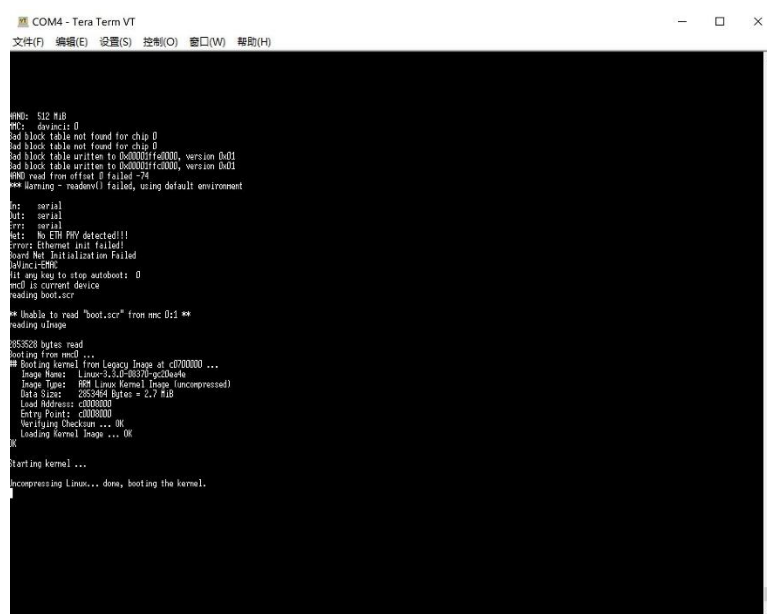


Dear TI,

Currently, I designed an evaluation board using OMAPL138EZWTD450 chip. The OMAPL138EZWTD450 chips are bought from retail .

The DSP core can run the DSP demo examples from starterware. Also, I want to use both cores, the arm core runs linux OS, and the DSP core runs signal processing. I put the uboot program and linux kernel to SD card, and set OMAPL138 boot from SD card. The SD card can start the uboot image, shown in Fig.1



```
COM4 - Tera Term VT
文件(F) 编辑(E) 设置(S) 控制(O) 窗口(W) 帮助(H)

HND: 512 M8
MC: davinci 0
Bad block table not found for chip 0
Bad block table not found for chip 0
Bad block table written to 0x0000fffa000, version 0x01
Bad block table written to 0x0000fffa000, version 0x01
HND read from offset 0 failed -74
** Warning - readenv() failed, using default environment
set: serial
set: serial
set: serial
set: No ETH PHY detected!!!
Error Ethernet unit failed!
Board Net Initialization Failed
Init:GMC
Hit any key to stop autoboot: 0
mc0 is current device
reading boot.scr
** Unable to read "boot.scr" from mmc 0:1 **
reading uImage
385320 bytes read
Booting from mmc ...
Booting kernel from Legacy Image at c0000000 ...
Image Name: Linux-3.6.9-0-gc0b5a4e
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 2053494 Bytes = 2.7 Mib
Load Address: c0000000
Entry Point: c0000000
Verifying Checksum ... OK
Loading kernel image ... OK
Starting kernel ...
Uncompressing Linux... done, booting the kernel.
```

Fig.1 The information printed by uboot.

Then the program is forever stopped at “uncompressing linux... done, booting the kernel”. I use the same SD card to launch two other evaluation boards successfully, one evaluation board is designed by Chinese TL company, the other is OMAPL138 LCDK designed by TI. So the I am sure the uboot and linux kernel program in the SD card

is correct. Then I compare the environment parameters in uboot, even the parameters are the same, my board still do not start the linux kernel, it still stopped at “uncompressing linux... done, booting the kernel”.

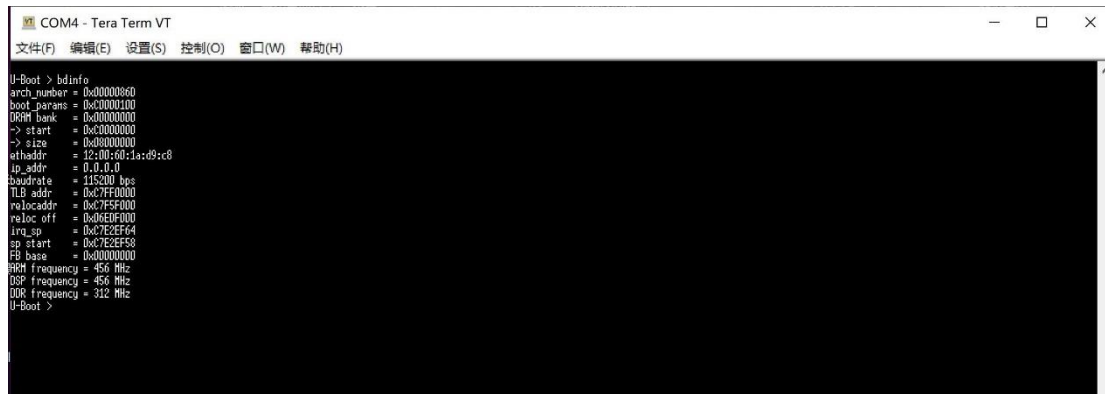
Fig.2 shows the environment from my board.

```
COM4 - Tera Term VT
文件(F) 编辑(E) 设置(S) 控制(O) 窗口(W) 帮助(H)
printenv
baudrate=115200
bootcount=run boottestcount; run setcommonargs; run mmcboot; run nandboot;
bootcount=0
bootdelay=1
bootfile=boot
bootfile=boot.scr
bootpart=0:2
bootscr=boot.scr
boottest=0
boottestcount=0
boottestcount=if test $boottest = 1; then setexpr bootcount $bootcount + 1; saveenv; echo boottest=1 bootcount: $bootcount; fi;
console=ttyS2,115200n8
da850-panel=TL0709
enac-mode=nil
ethact=0:0:0:0:0:0
ethaddr=08:00:00:08:00:00
hwconf=igdp:uake:yes
loadaddr=0xc0000000
loadbootscr=fatload mmc 0:1:1 $loadaddr $bootscr
loadimage=fatload mmc 0:1:1 $loadaddr $bootfile
man_args=mem=32M boot=32M boot=32M
mxcargs=setenv bootargs $commonargs root=${mxcroot} rootfstype=${mxcrootfstype}
mxcboot=mxc dev $mxcdev; if mmc rescan; then if run loadbootscr; then echo Loaded bootscript from $bootscr; source $scraddr; fi; if run loadimage; then echo Booting from mmc($mxcdev) ...; run mxcargs; boot $loadaddr; fi; fi;
mxcdev=0
mxcroot=/dev/mchlk02 ru
mxcrootfstype=ext3 rootwait
nandboot=if boot $nandbootpart; then echo Booting from nand ...; nand info; run nandargs; bootn $loadaddr; fi;
nandbootpart=nand.kernel
nanddev=0
nandrootfstype=ubifs
netargs=setenv bootargs $commonargs ip=dhcp root=/dev/nfs nfsroot=${serverip}:${rootpath},${nfspts} ru
netboot=echo Booting from network ...; run netargs; mmc dev $mxcdev; if mmc rescan; then if run loadimage; then bootn $loadaddr; fi; fi; if nboot $nandbootpart; then bootn $loadaddr; fi; fi;
nfspts=noLock
rootpath=/export/rootfs
scraddr=0xc0000000
setcommonargs=setenv commonargs console=${console} $man_args eth=${ethaddr} $optargs da850-panel=${da850-panel} da850-enac=${enac-mode}
spiboot=if sf probe $spidev; then sf read $loadaddr $spibootaddr $spibootsize; run nandargs; bootn $loadaddr; fi;
spibootaddr=0x00000
spibootsize=0x40000
spidev=0
stderr=serial
stdin=serial
stdout=serial
ver=U-Boot 2012.04.01-00078-gdccc9f5 (Aug 28 2018 - 17:17:05)
Environment size: 2299/131068 bytes
U-Boot >
```

Figure 2 the environment printed from uboot

Then I print the board information using “bdinfo” instructions in uboot, Fig.3 shows the information for evaluation Board designed by TL company, Fig.5 shows the information for OMAPL138 LCDK, and Fig.3 shows the information for my board. From the three figures, when the parameters (TLB addr, relocaddr, reloc off, irq_sp, sp start, FB base) are the same, shown in fig.3 and fig.5, the linux kernel can boot successfully. Compared with my board shown in fig. 4, the parameters “TLB addr”, “relocaddr”, “reloc off”, “irq_sp” are

different. Is the reason that my board cannot launch the linux kernel?
How to address the problems? I am waiting for your suggestions,
thank you very much



```
COM4 - Tera Term VT
文件(F) 编辑(E) 设置(S) 控制(O) 窗口(W) 帮助(H)

U-Boot > binfo
arch_number = 0x00000860
boot_params = 0xc0000100
DRAM bank = 0x00000000
-> start = 0xc0000000
-> size = 0x00000000
ethaddr = 12:00:00:1a:d9:c8
ip_addr = 0.0.0.0
baudrate = 115200 bps
TLB addr = 0xc7ff0000
relocaddr = 0xc7f5f000
reloc off = 0x06e0f000
irq_sp = 0xc7e2ef64
sp start = 0xc7e2ef58
FB base = 0x00000000
ARM frequency = 456 Mhz
DSP frequency = 456 Mhz
DDR frequency = 312 Mhz
U-Boot >
```

Fig.3 One evaluation Board designed by TL company



```
COM4 - Tera Term VT
文件(F) 编辑(E) 设置(S) 控制(O) 窗口(W) 帮助(H)

arch_number = 0x00000860
boot_params = 0xc0000100
DRAM bank = 0x00000000
-> start = 0xc0000000
-> size = 0x10000000
ethaddr = 02:01:50:34:31:03
ip_addr = 0.0.0.0
baudrate = 115200 bps
TLB addr = 0xcfff0000
relocaddr = 0xcfff0000
reloc off = 0xcfeef000
irq_sp = 0xcfe2ef64
sp start = 0xcfe2ef58
FB base = 0x00000000
ARM frequency = 456 Mhz
DSP frequency = 456 Mhz
DDR frequency = 312 Mhz
U-Boot >
```

Fig.4 My Board



The image shows a screenshot of a Tera Term VT window titled "COM5 - Tera Term VT". The window has a menu bar with options: 文件(F), 编辑(E), 设置(S), 控制(O), 窗口(W), and 帮助(H). The main area displays the output of a U-Boot boot process, listing various parameters in hexadecimal and decimal formats. The parameters include arch_number, boot_params, DRAM bank, start, size, ethaddr, ip_addr, baudrate, TLB addr, relocaddr, reloc off, irq_sp, sp_start, FB base, ARM frequency, DSP frequency, and DDR frequency. The output ends with "U-Boot >".

```
arch_number = 0x00000860
boot_params = 0xc0000100
DRAM bank   = 0x00000000
-> start    = 0xc0000000
-> size     = 0x08000000
ethaddr     = 16:01:20:38:64:4f
ip_addr     = 0.0.0.0
baudrate    = 115200 bps
TLB addr    = 0xc7ff0000
relocaddr   = 0xc7ef0000
reloc off   = 0x06e0f000
irq_sp      = 0xc7e2ef64
sp_start    = 0xc7e2ef58
FB base     = 0x00000000
ARM frequency = 456 MHz
DSP frequency = 456 MHz
DDR frequency = 312 MHz
U-Boot >
```

Fig.5 OMAPL138 LCDK designed by TI