20140116
上午分析了BCM和F601D配置SMI的接口，尝试了下，都无法配置5461S的phy。
上午还验证了从BRCM版本切换到CSP版本。替换cferom.bin和cferam.bin后，升级bcm96838GWO_nand_cferom_fs_image_128_jffs2.w可切换。

1. 目前9026-2的68380软件版本已经可以切换到CSP版本，PC可以ping通单板，但是由于没有裁剪语音，所以页面打不开。 ----李瑞。（先做这个）
注：68380烧结版本，目前可在南京ONU上启动，但是在9026-2上仍然无法启动。认为是硬件的问题，仍需要继续分析。
2. 5461S phy，参考F601D的方式操作SMI接口，无法读取和写入。今天下午重点和硬件量一下，配置过程中，有没有信号过去。必要时，需要技术支持指导。 ----刘金成/荆杰
1）如果无法读取，那么找F601D确认phy的操作方式。（硬件和软件）
---
硬件可以读取到RGMII的时钟，但是在操作命令时，MDIO没有反应，可能是操作接口不对，也可能是硬件问题。
8313的SMI接口连接带外phy，通过GPIO模拟SMI接口，硬件将5461S的SMI接口飞线到8313SMI，或者完全仿照SEGB那样，将5461S的SMI接口飞线到相应的GPIO，看phy能否正常初始化。
还大概浏览了5461S phy的datasheet，对比bsp_mdio_reg_get函数，有32bit的PRE。
提了个CSP，和BRCM确认接口是否正确。

20140117
上午开会。
将5461S的SMI飞线到8313，使用bsp mdio测试函数，turnround_MDIO失败。
和南京交流，了解到南京的RGMII也还没有调通，所以重点调试phy5461S直接对接phy2

20140118
对比使用bsp mdio测试函数，在9026上使用SEGB作对比，9026上正常，9026-2上不正常。
之后硬件分析波形，发现由于5461S没有reset，所以找不到这个phy，读出来的值不正常。硬件接地reset这个phy，5461S的SMI飞线到8313上，能够管理5461S的寄存器。
问题1。但是通过实验，发现即使手动reset 5461S，68380仍然无法读取phy的状态。下周一和南京同事确认，F601D是否有配置SMI接口的工作模式（张维松分析这两个管脚和GPIO复用，默认为GPIO模式）
问题2。此外，手动reset 5461S，68380上和5461S相连的phy仍然无法link up。对比9026-2和9026上，5461S的所有32个寄存器，发现只有一两个状态bit位不同，其余完全相同。（shadom寄存器没有读取）。下周一，需要找新蕾或BRCM或上海支持，看有没有人熟悉这个phy芯片。

```
port ge3
PORT: Status (* indicates PHY link up)
  ge3 LS(SW) Auto(no link) Ability (fd = 1000MB,2500MB hd = intf = gmii,sgmii medium = pause = pause_tx,pause_rx,pause_asymm lb =
none,MAC,PHY flags = autoneg )Local (fd = 1000MB,2500MB hd = intf = medium = pause = pause_tx,pause_rx lb = flags = )STP(Forward)
Lrn(ARL,FWD) UtPri(0) Pfm(FloodNone) IF(SGMII) Max_frame(1522) MDIX(ForcedNormal, Normal) Medium(Fiber)

BCM.0> ps ge3
          ena/ speed/ link auto STP lrn inter max loop
      port link duplex scan neg? state pause discrd ops face frame back
       ge3 down - SW Yes Forward TX RX None FA SGMII 1522

BCM.0> phy ge3
Port ge3 (PHY addr 0x84): COMBO65 (Internal 2.5G 65nm SERDES PHY Driver)
        0x00: 0x0000 0x01: 0x0000 0x02: 0x0000 0x03: 0x0000
        0x04: 0x0000 0x05: 0x0000 0x06: 0x0000 0x07: 0x0000
        0x08: 0x0000 0x09: 0x0000 0x0a: 0x0000 0x0b: 0x0000
        0x0c: 0x0000 0x0d: 0x0000 0x0e: 0x1140 0x0f: 0x0000
        0x10: 0x1140 0x11: 0x0119 0x12: 0x0143 0x13: 0xbff0
        0x14: 0x00a0 0x15: 0x6120 0x16: 0x0006 0x17: 0x2001
        0x18: 0x0000 0x19: 0x0000 0x1a: 0x0000 0x1b: 0x0000
        0x1c: 0x0000 0x1d: 0x0000 0x1e: 0x0000 0x1f: 0xffe0
```

参考SEGB，配置寄存器
bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x1400 -->0x141f
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0x141e <--Disable CLK125 output
show c c没有计数

通过镜像，让ge3可以向外发包
dmirror ge2 mode=all DP=ge3
show c c显示，ge3的tx有计数增加

设置RGMII Timing Mode

bsp_mdio_reg_get_shadom 0,0x19,0x18,0x7007 ----》0x70e7
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0x71e7 ----》0x721c

确认上行方向，rdpa有收到包，但是全部被emac4丢弃。
bs /b/e system
Object: system. Object type: system. Owned by: root
==========
init_cfg : {wan_type=gpon,emac_mode=
{group_mode=qsgmii,emac4_mode=rgmii,emac5_mode=sgmii},num_wan=1,num_lan=5,enabled_emac=emac0+emac1+emac2+emac3+emac4,wan_emac=emac4,switching_mode=none,ip_class_method=none,runner_ext_sw={enabled=no,emac_id=none,hdr_type=none}}

wan_emac=emac5

设置SGMII自环，下行发包不通，但是硬件量，还是有很多报文。

bs /bdmf/ex system children:yes class:config max_prints:-1

1月24日
硬件将68380到5461S的时钟断开，结果，5461S和68380还是可以link up为1000FD，很奇怪。
此外，由于修改了init_cfg中的wan_emac，通过56024镜像向68380发包，收包个数正常，收包行为正常上CPU，收包可打印。

现在怀疑，68380和5461S之间的时钟有问题。

调试SMI接口
# ethctl phy ext reg 0x19 4
ext
ext
mii register 25 is 0xffff
反复执行这个命令，硬件量68380的SMI接口，没有信号。说明68380没有将相应的信号发出来。

# ethctl eth1 reg 0
mii (phy addr 0x2) register 0 is 0x1140
# bs /m mr 0xb4e00600 w 1
b4e00600: db24ffff *$.....*

硬件修改后，CFE中硬件自检失败，在DIE0处。

bsp_mdio_reg_set_shadom 0,0x19,0x1C,0xf9e3
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0x79e3
bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x7c03

dmirror ge1 mode=all DP=ge3

bsp_mdio_reg_get_shadom 0,0x19,0x1C,0xfc00 确认fiber signal是否为1，如果不是1，则需要反一下SD
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0xf9e3 反一下SD，signal detect
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0xfc03 fiber模式，选择fiber寄存器空间
bsp_mdio_reg_set 0,0x19,0,0x4140 关闭自动协商，对fiber环回。设置环回时必须关闭自动协商。
bsp_mdio_reg_get_test 0,0x19,0 确认环回成功
bsp_mdio_reg_get_test 0,0x19,1 查看端口状态，第一次读不算数
bsp_mdio_reg_get_test 0,0x19,1 查看端口状态，第二次读
bsp_mdio_reg_get_shadom 0,0x19,0x1C,0xfc00 看fiber模式有没有up

选择serdes或者SGMII模式，可通过配置ge3的内部phy地址来实现。
首先要选择合适的block，然后再配置寄存器。可采用类似命令
phy ge3 0x1f 0

切换到coper模式
bsp_mdio_reg_get_shadom 0,0x19,0x1C,0xfc00 0x7d7a
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0xfc02

设置rx delay
bsp_mdio_reg_get_shadom 0,0x19,0x18,0x7007 0x70e7
bsp_mdio_reg_set_shadom 0,0x19,0x18,0x71e7 0x71e7
bsp_mdio_reg_set_shadom 0,0x19,0x18,0x70e7

设置tx delay
bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x0c00 0
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0x0e00

从下面的描述可以看出，TXCLK指的是Input timing。之前的理解有问题，导致不通。

1月18日
9026初始化phy5461S

```
void bspUpCardInit(void)^M
{^M
    UINT16 data=0;^M
^M
    /*disable BCM5461S clk125*/^M
    if(PORTTYPE_SEGB1 == bspGetMainType())^M
    {^M
        taskLock();^M
        ^M
        bspMdioSelect(4);^M
    ^M
        bsp_mdio_reg_set(0,0x19,0x1c,0x1400); /*enable read bit9:0*/^M
        bsp_mdio_reg_get(0,0x19,0x1c,&data);^M
        data &= ~0x01; /*disable clk125*/^M
        data |= 0x8000; /*enable write bit9:0*/^M
        bsp_mdio_reg_set(0,0x19,0x1c,data);^M
^M
        taskUnlock();^M
    }^M
}^M
```

9806/BSPS/CDZ/bspHWinterface.c

F601D初始化phy50612

```
 /*B50612: GTXCLK[9]set 0 : normal mode */
 reg_addr = 0x1c;
 reg_data = 0x8c00; 配置RGMII?
 mdio_write(EXT_PHY_ADDR,reg_addr,reg_data);
```

```
int
soc_miim_write(int unit, uint16 phy_id,
           uint8 phy_reg_addr, uint16 phy_wr_data)
{
   int rv = SOC_E_NONE;
#if defined(_INSTALL_CFZ)
   if((0x19 == phy_id) || (0x1A == phy_id))
   {
      hwBcmConfLock();
      bsp_mdio_reg_set(0,phy_id,phy_reg_addr,phy_wr_data);
      hwBcmConfUnLock();
      return rv;
   }
#endif
```

miim.c 交换芯片的驱动做了适配。

```
LOCAL UINT16 bcm_port_phy_addr[HW_BCM_PORT_NUM] =
           {
              HW_BCM_NULL_PHY, /* GE0: not used */
              0xC, /* GE1: 调试口 */
              HW_BCM_NULL_PHY, /* GE2:GMII口 */
              HW_BCM_NULL_PHY, /* GE3: 接上联子卡 */
              HW_BCM_NULL_PHY, /* GE4: 接上联子卡 */
           };
```

```
/*缺省bcm_port_phy_addr中记录的是没有上联子卡的情况，有上联小卡发生变化，则需要修改phy_id。*/
void hw_bcm_phy_add_init(void)
{
    if(HW_UPLINK_OGSDA == UpLinkCardType || HW_UPLINK_OGSDC ==UpLinkCardType)
    {
        printf("hw_bcm_phy_add_init: OGSDC.\n");
        bcm_port_phy_addr[3] = 0x19;
        bcm_port_phy_addr[4] = 0x1A;
    }
    /* start -- added by djj 120904 */
    else if(HW_UPLINK_SEGB == UpLinkCardType || HW_UPLINK_SEGB10 == UpLinkCardType)
    {
        bcm_port_phy_addr[3] = 0x19;
    }
    /* end -- added by djj 120904 */
}
```

hwBcmConfig.c 对于SEGB子卡，配置phy_addr

```
UINT16 hw_bcm_get_port_phy_addr(bcm_port_t bcm_port)
{
    UINT16 phyID = HW_BCM_NULL_PHY;

    if ((bcm_port <= HW_BCM_PORT_NUM) && (bcm_port >= 1))
    {
        phyID = bcm_port_phy_addr[bcm_port - 1];
    }

    return phyID;
}
```

hwBcmConfig.c 提供给交换芯片的接口，用于控制phy。

```
->bsp_mdio_reg_get_test 0,0x19,0
MDIO Error 1!
0xffff
static inline int turnaround_MDIO(void)
```

1月19日

SGMII到5461S的phy，此时有两种选择，走RGMII或者走phy2，这里是二选一的关系，应该通过SMI去选择。
目前RGMII在工作，有时钟信号，是否意味着5461S的默认工作状态是RGMII？

H168N V2.0上有类似设计，63168通过RGMII连接到外部的phy上。这里用到的phy和F601D是同一款。因此，也可以在H168N V2.0上验证，我们使用的SMI接口操作命令是否正确。

1. If your GE is rgmii mode, please change this line.
from
{bp_ulPhyId4, .u.ul = 0x18 },
to
{bp_ulPhyId4, .u.ul = 0x18 | MAC_IFACE_VALID | MAC_IF_RGMII},

从8313上读出来的phy寄存器情况
1140 0169 0020 60c1 0060 c0a0 0002 0000
0000 0000 0000 0000 0000 0000 0000 c000
0000 2000 0000 0000 0000 0000 0000 0000
0400 1000 0000 ffff 0000 0000 0000 0000

9026 SEGB卡，phy ge3
0x10: 0x1140 0x11: 0x012d 0x12: 0x0143 0x13: 0xbff0
0x14: 0x00a0 0x15: 0x4060 0x16: 0x0004 0x17: 0x2001
0x18: 0x0000 0x19: 0x0000 0x1a: 0x0000 0x1b: 0x0000
0x1c: 0x0000 0x1d: 0x0000 0x1e: 0x0000 0x1f: 0xffe0

9026 SEGB卡，bsp_mdio_reg_get_test 0,0x19,X

1140 016d 0020 60c1 0060 c0a0 0000 0000
0000 0000 0000 0000 0000 0000 0000 c000
0000 2000 0000 0000 0000 0000 0000 0000
0400 1000 0000 ffff 0000 0000 0000 0000

01h Table 27: “1000BASE-T/100BASE-TX/10BASE-T MII Status Register (Address 01h),” on page 69
06h Table 31: “1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Expansion Register (Address 06h),” on page 76
01h寄存器：
-->9026-2：0000 0001 0110 1001 Link down
-->9026 ：0000 0001 0110 1101 Link up
06h寄存器：
-->9026-2：0000 0000 0000 0010 1 = New page has been received from link partner
-->9026 ：0000 0000 0000 0000 0 = New page has not been received
Page Received
The BCM5461S returns a 1 in bit 1 of the 1000BASE-T/100BASE-TX/10BASE-T Auto-negotiation Expansion register when a new link code word has been received from the link partner since the last time this register was read; otherwise, it returns a 0.

1月20日
完成shadom寄存器读写接口

```
void bsp_mdio_reg_get_test(UINT32 dev_id, UINT32 phy_addr, UINT32 reg)
{
 UINT16 data ;
 bsp_mdio_reg_get(dev_id, phy_addr, reg, &data) ;
 printf("0x%02x\n", data) ;
}

void bsp_mdio_reg_get_shadom(UINT32 dev_id, UINT32 phy_addr, UINT32 reg, UINT16 shadom)
{
 UINT16 data ;
// if(reg == 0x1C)
 {
          bsp_mdio_reg_set(dev_id, phy_addr,reg, shadom); /*enable read bit9:0*/
       bsp_mdio_reg_get(dev_id, phy_addr, reg,&data);
            printf("get 0x%02x\n", data) ;
 }
}

void bsp_mdio_reg_set_shadom(UINT32 dev_id, UINT32 phy_addr, UINT32 reg, UINT16 data)
{
// if(reg == 0x1C)
 {
       printf("set:\n") ;
             data |= 0x8000; /*enable write bit9:0*/
             bsp_mdio_reg_set(dev_id, phy_addr,reg,data);

  data &= 0x7FFF ;
  bsp_mdio_reg_get_shadom(dev_id, phy_addr, reg, data) ;
 }
}
```

bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x1400
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0x141e

1Ch Shadow Value 11111 2:1
bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x7c00
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0x7c6d -->0x7c2d

1Ch Shadow Value 11111 0
bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x7c00 -->0x7c6b fiber link up
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0x7c6d <--SGMII Mode
bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x7c00 -->0x7c2d fiber link down，但是copper没有link up

bsp_mdio_reg_get_test 0,0x19,0 -->0x1940 ？？两次不一样
bsp_mdio_reg_set 0,0x19,0,0x1140
bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x7c00 -->0x7c2c？

00h Isolate
bsp_mdio_reg_set 0,0x19,0,0x1540
bsp_mdio_reg_get_test 0,0x19,0 -->0x1540 Isolate

09h Master/Slave
bsp_mdio_reg_get_test 0,0x19,9

To enable 100BASE-FX SGMII path mode:
• Set register 1Ch, shadow 00010, bit 4 = 1 to enable 100BASE-FX. 1 = Enable 100FX mode on MDI pairs 0 = Normal copper operation on MDI pairs
• Set register 00h to 2100h to disable auto-negotiation and to force the speed to 100 Mbps full duplex.
bsp_mdio_reg_set 0,0x19,0,0x2100
bsp_mdio_reg_get_test 0,0x19,0 -->0x0140，reset 68380 phy，down
• Set register 18h to 0430h to set the edge rate control.

bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x7800 -->0x7862
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0x7860 0 = Copper selected when both media are active自动介质检测禁用

Auxiliary 1000BASE-X Status Register (Address 1Ch, Shadow Value 11100)
bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x7000 -->0x721c

AUTODET

1月21日

2:1 Mode Select R/W 00 = Copper
01 = Fiber
10 = SGMII
11 = GBIC
INTF_SEL[1:0]

Address 1Ch, Shadow Value 00010 bit4

一、重新考虑SGMII mode
1. 设置1Ch，shadom 1F，bit1~2=10，bit0 =0。
bit0=0，表示disable 1000BASE-X，enable copper registers for address 00h~0Fh。
---->bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x7c00 ==0x7c6b
---->bsp_mdio_reg_set_shadom 0,0x19,0x1C,0x7c6c ==0x7c2c
---->bsp_mdio_reg_get_test 0,0x19,1 ==0x7949
---->bsp_mdio_reg_get_test 0,0x19,0 ==0x1940 bit11=1，power down
2. 检查Address 1Ch, Shadow Value 00010 bit4
bit4=1 enable 100FX mode on MDI pairs，bit4=0，Normmer copper operation on MDI paris
---->bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x0800 ==0x0800
3. 设置01h bit11 power <---------完成此步骤，需要硬件量是否有信号出来（能否通过寄存器去判断是否有信号发出？）
---->bsp_mdio_reg_set 0,0x19,0,0x1140 设置bit11=0，normer mode
---->bsp_mdio_reg_get_test 0,0x19,1 ==0x7949。eth0 link up，01h为0x79c9。Link Fail state。
Page7 1000M-T master和slave在1微妙内检测到双方status，那么就进入到link-pass state
4. 分析master和slave模式
Page 37 master和slave可由auto-negotiation中自动产生。
09h，Page 79，控制Master、Slave、Test mode等。
0ah，09h的状态信息。如Master、Slave是否设置成功等。
5. 对比正常和异常时的收发包计数
6. Address 18h, Shadow Value 111。 Misc Control Register。禁用RGMII
bit9 Force Auto-MDIX mode。只有当auto-negotiation disable时，才有效。
bit7 RGMII mode 1 = eanble ，0 = disable
bit3 MDIO All PHY Select， 1 = All PHY selected during MDIO writes when PHY Address = 00000
7. 考虑RGMII和TRD的优先级

8. 19h，Auxiliary Status Summary Register
比如自动协商是否完成，是否收到自动协商请求等。
1Ah、1Bh，phy产生中断及中断掩码

二、设置copper mode
1. 设置1Ch，shadom 1F，bit1~2=00，bit0 =0。
---->bsp_mdio_reg_get_shadom 0,0x19,0x1C,0x7c00 ==0x7c2c
---->bsp_mdio_reg_set_shadom 0,0x19,0x1C,0x7c28 ==0x7c28

三、RESET CPLD
1.68380_RST_N ：68380的硬复位信号，CPLD的0x0b寄存器，bit0，0-复位，1-不复位。整机复位的时候建议将该信号复位
2.68380_RST1_N: 68380的软复位信号，CPLD的0x0b寄存器，bit1,0-复位，1-不复位。
3.5641S_RST_N: 5461S的硬复位信号，CPLD的0x0b寄存器, bit2,0-复位，1-不复位。整机复位的时候建议将该信号复位

结论：
1.5461s的phy口可以通；
但是需要加变压器，然后中心抽头到2.5v的电压，因为5461s是电压型的phy；

//我们验证了5461s应该是电压型的；
断开5461s跟68380的连接，直接从5461s飞线到变压器，然后中心抽头接2.5v的电源；
这个时候跟电脑网口对接，是可以协商的。//
bsp_mdio_reg_set_shadom 0,0x19,0x1C,0x7c6c
bsp_mdio_reg_set 0,0x19,0,0x1140
需要设置1000BASE-T的power为nomer mode，默认为power down

---

RGMII初始化
1. 编译boot
cd cfe/build/broadcom/bcm63xx_rom
make clean
make BRCM_CHIP=6838 BLD_NAND=1 CFG_COPY_PSRAM=1 INC_PMC_DRIVER=1 BLD_NAND=1 CONFIG_ZTE_DEMOGPON=1
cp cfe6838rom.bin ../bcm63xx_ram/cfe6838ram.bin ../../../../targets/cfe/
cd -

2. make menuconfig
去掉USB相关的所有功能。

3. 修改boardparam.c
{bp_ulPhyId4, .u.ul = 0x19 | MAC_IF_RGMII|PHY_EXTERNAL|PHY_INTEGRATED_VALID},

4. 修改bcm63xx_dev.c
增加
set_pinmux(PINMUX_RGMII_MDIO, PINMUX_RGMII_FUNC); //PINMUX_RGMII_MDIO=47

boot启动时，有如下打印：
pE[0].sw.phy_id[4] = 0x2180019
Configuring RGMII pinpux
ddr_tm_base_address = 0xa0800000

5. 合入BRCM修改
相关脚本在userspace\public\apps\bdmf_shell\scripts目录下。
---
1）.
in rdpa_gpon_init.sh:
In system object:
bs /bdmf/new system/init_cfg={wan_type=gpon, emac_mode={group_mode=qsgmii,emac4_mode=rgmii,emac5_mode=sgmii},num_lan=5,
enabled_emac=emac0+emac1+emac2+emac3+ emac4,switching_mode=none, ip_class_mode=none}

2）.
rdpa_common_init.sh:
Add one more lan port object as:
bs /bdmf/new port/index=lan4, cfg={emac=emac4}

When creating DS egress_tm objects and priority queues, create egress_tm and priority queue for Lan4 as well.

bs /bdmf/new egress_tm/dir=ds,index=4,level=queue,mode=sp port/index=lan4
bs /bdmf/configure egress_tm/dir=ds,index=4,queue_cfg[0]={weight=0,drop_alg=dt,drop_threshold=128,red_high_threshold=0,red_lowthreshold=0
}

3）. In The rdpa_common_filter_init.sh.
Add the common filter for Lan4 as well.

6. 尝试调试SMI