

# 曙光 BIOS Porting Guide 用户手册

DAWNING INFORMATION INDUSTRY CO.,LTD.

## 声明

本手册的用途在于帮助您正确地使用曙光公司 BIOS Porting Guide 产品(以下称“本产品”), 在使用本产品前, 请您务必先仔细阅读随机配送的所有资料, 特别是本手册中所提及的注意事项。这会有助于您更好和安全地使用本产品。请妥善保管本手册, 以便日后参阅。

---

本手册的描述并不代表对本产品规格和软硬件配置的任何说明。有关本产品的实际规格和配置, 请查阅相关协议、装箱单、产品规格配置描述文件, 或向产品的销售商咨询。

---

如您不正确地或未按本手册的指示和要求安装、使用或保管本产品, 或让非曙光公司授权的技术人员修理、变更本产品, 曙光公司将不对由此导致的损害承担任何责任。

---

本手册中所提供照片、图形、图表和插图, 仅用于解释和说明目的, 可能与实际产品有些差别, 另外, 产品实际规格和配置可能会根据需求需要不时变更, 因此与本手册内容有所不同。请以实际产品为准。

---

本手册中所提及的非曙光公司网站信息, 是为了方便起见而提供, 此类网站中的信息不是曙光公司产品资料的一部分, 也不是曙光公司服务的一部分, 曙光公司对这些网站及信息的准确性和可用性不做任何保证。使用此类网站带来的风险将由您自行承担。

---

本手册不用于表明曙光公司对其产品和服务做了任何保证, 无论是明示的还是默示的, 包括(但不限于)本手册中推荐使用产品的适用性、安全性、适销性和适合某特定用途的保证。对本产品及相关服务的保证和保修承诺, 应按可适用的协议或产品标准保修服务条款和条件执行。在法律法规的最大允许范围内, 曙光公司对于您的使用或不能使用本产品而发生的任何损害(包括, 但不限于直接或间接的个人损害、商业利润的损失、业务中断、商业信息的遗失或任何其他损失), 不负任何赔偿责任。

---

对于您在本产品之外使用本产品随机提供的软件, 或在本产品上使用非随机软件或经曙光公司认证推荐使用的专用软件之外的其他软件, 曙光公司对其可靠性不做任何保证。

---

曙光公司已经对本手册进行了仔细的校勘和核对, 但不能保证本手册完全没有任何错误和疏漏。为更好地提供服务, 曙光公司可能会对本手册中描述的产品软件和硬件及本手册的内容随时进行改进或更改, 恕不另行通知。如果您在使用过程中发现本产品的实际情况与本手册有不一致之处, 或您想得到最新的信息或有任何问题和想法, 欢迎致电 400-810-0466 或登录曙光公司服务网站 [www.sugon.com](http://www.sugon.com) 垂询。

---

## 商标和版权

“SUGON”及图标是曙光信息产业股份有限公司的商标或注册商标。

“中科曙光”及图标是曙光信息产业股份有限公司的商标或注册商标，文中“曙光信息产业股份有限公司”简称“曙光公司”。

---

上面未列明的本手册提及的其他产品、标志和商标名称也可能是其他公司的商标或注册商标，并由其各自公司、其他性质的机构或个人拥有。

---

在本用户手册中描述的随机软件，是基于最终用户许可协议的条款和条件提供的，只能按照该最终用户许可协议的规定使用和复制。

---

版权所有©2013 曙光信息产业股份有限公司，所有权利保留。

---

本手册受到著作权法律法规保护，未经曙光信息产业股份有限公司事先书面授权，任何人士不得以任何方式对本手册的全部或任何部分进行复制、抄录、删减或将其编译为机读格式，以任何形式在可检索系统中存储，在有线或无线网络中传输，或以任何形式翻译为任何文字。

---

## 目录

|                               |     |
|-------------------------------|-----|
| 声明 .....                      | ii  |
| 商标和版权 .....                   | iii |
| 插图目录 .....                    | v   |
| 表格目录 .....                    | vi  |
| 变更记录 .....                    | vii |
| 1 产品简介 .....                  | 8   |
| 1.1 产品概述 .....                | 8   |
| 2 产品功能 .....                  | 8   |
| 2.1 CRB 项目 .....              | 8   |
| 2.2 代码准备 .....                | 8   |
| 2.3 内存 Porting .....          | 9   |
| 2.4 USB Porting .....         | 12  |
| 2.5 DXIO 与 GPIO Porting ..... | 12  |
| 2.5.1 DXIO porting .....      | 13  |
| 2.5.2 GPIO Porting .....      | 14  |
| 2.6 BoardId Porting .....     | 14  |
| 2.7 UART 及其他 Porting .....    | 15  |
| 2.8 编译脚本 Porting .....        | 18  |
| 2.9 功能 Porting .....          | 18  |
| 2.10 Smbios Porting .....     | 18  |
| 附录一 缩略语与缩写 .....              | 20  |

## 插图目录

|  |    |
|--|----|
| 图 2-1 G30 项目 CRB 配置 .....                  | 8  |
| 图 2-2 G40 项目 CRB 配置 .....                  | 8  |
| 图 2-3 XorcoreBoardPkg 文件结构 .....           | 9  |
| 图 2-4 内存 SPD 地址配置 .....                    | 9  |
| 图 2-5 ChannelId 与 CPU 引脚对应关系 .....         | 10 |
| 图 2-6 MuxChannel 与 Channel 对应关系 .....      | 10 |
| 图 2-7 内存 I2C Mux 地址配置 .....                | 10 |
| 图 2-8 内存 SPD 地址默认对应关系 .....                | 11 |
| 图 2-9 内存 SPD 地址配置示例 .....                  | 11 |
| 图 2-10 内存 Channel 对应转化配置 .....             | 11 |
| 图 2-11 内存 Dimm 对应转化配置 .....                | 12 |
| 图 2-12 USB OC 映射配置 .....                   | 12 |
| 图 2-13 DXIO 配置 .....                       | 13 |
| 图 2-14 BoardId 定义 .....                    | 14 |
| 图 2-15 PlatformId 定义 .....                 | 15 |
| 图 2-16 PlatformId 安装 .....                 | 15 |
| 图 2-17 默认 Porting 的 Die 数量 .....           | 15 |
| 图 2-18 Uart 配置 .....                       | 15 |
| 图 2-19 Uart Pin 配置 .....                   | 15 |
| 图 2-20 Uart 默认 Pin 配置 .....                | 15 |
| 图 2-21 ABL 阶段 UART 配置 .....                | 16 |
| 图 2-22 I2C 配置 .....                        | 16 |
| 图 2-23 AST2500 功能配置 .....                  | 16 |
| 图 2-24 AST2500 I0decode 及 port 80 配置 ..... | 16 |
| 图 2-25 SIO 配置 .....                        | 17 |
| 图 2-26 SIO 配置 .....                        | 17 |
| 图 2-27 PCI_SLOT_MAP_LIST 配置 .....          | 17 |
| 图 2-28 PCI_SLOT_MAP_LIST 结构体定义 .....       | 17 |
| 图 2-29 BIOS 名称配置 .....                     | 18 |
| 图 2-30 编译脚本主要定义内容 .....                    | 18 |
| 图 2-31 功能 Token 重定义 .....                  | 18 |
| 图 2-32 Smbios 默认值定义 .....                  | 19 |

表格目录

表 2-1 内存 SPD 地址配置变量定义 ..... 9

表 2-2 内存配置宏定义 ..... 10

表 2-3 HEADER TABLE ..... 13

表 2-4 PCIE TABLE ..... 13

表 2-5 GPIO 配置变量定义 ..... 14

变更记录

| No | 修改日期       | 修 定 内 容   | 修前<br>版本 | 修后<br>版本 |
|----|------------|---|----------|----------|
| 1  | 2020-12-30 | 首次制定  | /        | 0.1      |
| 2  | 2022-02-26 | 更新基础内容，增加 XorcoreBoardPkg、BoardId<br>Porting、编译脚本 Porting、功能 Porting、Smbios<br>Porting 等，针对 G40 和 3 号 CPU 做了改动。 | 0.1      | 0.2      |

# 1 产品简介

在本文档中，您将可以了解到中科曙光自主开发的 BIOS 适配不同主板硬件设计的使用说明。

## 1.1 产品概述

为适配新的主板，根据主板的硬件设计规格，对内存，PCIE，SATA，GPIO，USB 等硬件设计进行适配。

# 2 产品功能

## 2.1 CRB 项目

| CRB         | Package 类型 | Socket 数量 | Channels/Socket | DIMMs/Channel |
|-------------|------------|-----------|-----------------|---------------|
| Hygon35N16  | SL1        | 1         | 8               | 2             |
| Hygon65N32  | SL1        | 2         | 8               | 2             |
| HygonDM1SLT | DM1        | 1         | 2               | 2             |
| Hygon52D16  | SL1r2      | 2         | 4               | 2             |
| Hygon35N16  | SL1r2      | 1         | 4               | 2             |
| Hygon65N32  | SL1r2      | 2         | 4               | 2             |

图 2-1 G30 项目 CRB 配置

| CRB         | Package类型 | Socket数量 | Channels/Socket | DIMMs/Channel |
|-------------|-----------|----------|-----------------|---------------|
| Hygon62DB32 | SL1/SL1r2 | 2        | 8/4             | 2             |
| Hygon52DB16 | SL1r2     | 2        | 4               | 2             |
| Hygon32DB16 | SL1/SL1r2 | 1        | 8/4             | 2             |

图 2-2 G40 项目 CRB 配置

可根据当前主板 Socket 型号数目及内存配置选择相似配置的 CRB（如图 2-1 和图 2-2）在对应 CRB 项目基础上进行 porting。G30 支持海光 1 号和 2 号 CPU，G40 支持海光 1、2、3、4 号 CPU，若项目要支持 3 号或 4 号 CPU，需要参考 G40 的 CRB 项目。如果项目本身是 CRB，可以参考同代的其他 CRB。

## 2.2 代码准备

选择相似配置的 CRB 以后，在 XorcoreBoardPkg 下建立对应项目的文件夹，文件夹名称为“Hygonxxxx”，其中 xxxx 代表主板名称，如 32DB16。项目文件夹下包含 AcpiTables、Apcb、BoardId、DxioAndGpio、Include、USB 文件夹以及 HygonCpm.dxe.inc.fdf、HygonCpm.inc.dsc、HygonCpm.pei.inc.fdf、HygonCpmDefines.inc.dsc 文件，项目文件结构如图 2-3；

参考 CRB 项目在根目录下新建 Hygonxxxx.bat；

为主板申请一个 BoardId；

为主板申请一个 BIOS 名称；



将以上三条信息更新到云桌面 Wiki>UEFI>项目-主板-bios 页面下。

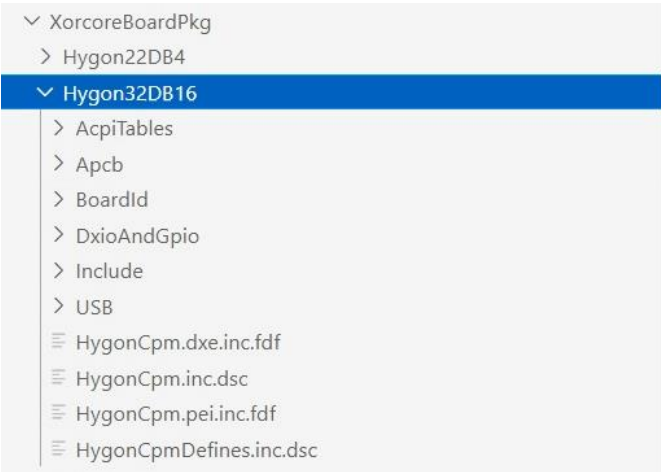


图 2-3 XorcoreBoardPkg 文件结构

2.3 内存 Porting

在 XorcoreBoardPkg\Hygonxxxx\Apcb 路径 ApcbData\_ZP\_GID\_0x1704\_Type\_DimmInfoSpd.c 文件配置对应 SPD 地址信息，图 2-4 为 CRB DM1SLT 内存配置，DM1SLT 内存数目为 4，因此图 2-4 内存 SPD 地址信息配置为 4 组。

```
DIMM_INFO_SMBUS DimmInfoSmbus [] =
{
    TRUE, 0, 0, 0, 0xa0, 0xff, 0xff, 0xff,
    TRUE, 0, 0, 1, 0xa2, 0xff, 0xff, 0xff,

    TRUE, 0, 1, 0, 0xa4, 0xff, 0xff, 0xff,
    TRUE, 0, 1, 1, 0xa6, 0xff, 0xff, 0xff,
};
```

图 2-4 内存 SPD 地址配置

图 2-4 中 SPD 地址信息详细定义 DIMM\_INFO\_SMBUS 见表 2-1。

表 2-1 内存 SPD 地址配置变量定义

| 变量名               | 定义                        |
|-------------------|---------------------------|
| DimmSlotPresent   | 内存是否在位                    |
| SocketId          | 内存所在 Socket, 0-1          |
| ChannelId         | 内存所在 Channel, 0-7         |
| DimmId            | 内存所在 Dimm, 0-1            |
| DimmSmbusAddress  | 内存的 I2C 地址                |
| I2CMuxAddress     | I2C Mux 的地址               |
| MuxControlAddress | 内存对应 I2C Mux 的控制地址        |
| MuxChannel        | 内存对应 I2C Mux 的 channel 通道 |

其中，ChannelId 是指 DIMM 连接到 CPU 的引脚所属 Channel，对应关系如图 2-5；

MuxControlAddress 是指访问 MUX 时，SMBUS “Write Byte Data”协议发送的 command，对于 PCA9546APWR 芯片，会忽略这个值，设置为 0xFF 即可；MuxChannel 是指访问 MUX 时，SMBUS “Write Byte Data”协议发送的 Data，填入的是选择 MUX 的 Channel 所填入的值，对于 PCA9546APWR 芯片，需符合规则如图 2-6。

| Channel | A | B | C | D | E | F | G | H |
|---------|---|---|---|---|---|---|---|---|
| ID      | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

图 2-5 ChannelId 与 CPU 引脚对应关系

| Channel                           | 0   | 1   | 2   | 3   |
|-----------------------------------|-----|-----|-----|-----|
| MuxChannel = 2 <sup>Channel</sup> | 0x1 | 0x2 | 0x4 | 0x8 |

图 2-6 MuxChannel 与 Channel 对应关系

根据主板内存配置还需修改路径 XorcoreBoardPkg\Hygonxxxx\Apcb\Include 路径下 ApcbCustomizedDefinitions.h 中配置与表 2-2 对应内存定义。

表 2-2 内存配置宏定义

| 宏                            | 定义                             |
|------------------------------|--------------------------------|
| NUMBER_OF_DIMMS_PER_CHANNEL  | 每个 channel 的 Dimm 数目           |
| NUMBER_OF_CHANNEL_PER_SOCKET | 每个 Soceket 的 channel 数目        |
| BLDCFG_I2C_MUX_ADDRESS       | I2C Mux 的地址，如无 I2C Mux 则为 0xFF |

当硬件设计主板内存 SPD 连接到 I2C Mux 时，根据 I2C Mux 硬件设计及 I2C Mux 规范需在路径 XorcoreBoardPkg\Hygonxxxx\DxioAndGpio\Pei\HygonCpmOemInitPeim.c 图 2-7 处修改配置。例如图 2-7 示例硬件设计 I2C Mux 地址为 0xE0 采用 I2C Mux 芯片为 PCA9546，I2CMuxAddr 也就是 I2C Mux 地址 0xE0 为硬件设计定义，I2CCommand 与 MuxControlAddress 在 I2C Mux 规范文档定义，图 2-7 处修改并非与内存数目对应，配置相同可合并。

```
typedef struct _MUX_DIMM_INFO_SMBUS{
    UINT8    I2CMuxAddr;
    UINT8    I2CCommand;
    UINT8    I2CChannelAddr;
} MUX_DIMM_INFO_SMBUS;

MUX_DIMM_INFO_SMBUS    mMuxDimmSmbus[]={
    {0xE0, 0xFF, 0x01},
    {0xE0, 0xFF, 0x02},
    {0xE0, 0xFF, 0x04},
    {0xE0, 0xFF, 0x08},
    {0xFF, 0xFF, 0xFF}
};
```

图 2-7 内存 I2C Mux 地址配置

当前 CRB 默认采用图 2-8 的内存配置与 Smbus 地址对应关系，数组变量 3 个 1 组，定义分别为

{ChannelId, DimmId, SmbusAddress}, 如主板内存 ChannelId 和 DimmId 与 SmbusAddress 的对应关系与下图不一致, 例如 Channel 2 Dimm 0 与 Smbus Address 0xA0 这种对应关系与图 2-8 不一致, 需要根据硬件设计在路径 XorcoreBoardPkg\Hygonxxxx\DxioAndGpio\Pei\HygonCpmOemInitPeim.c 图 2-9 处修改, 图 2-9 处变量 4 个 1 组, 定义分别为{SocketId, ChannelId, DimmId, SmbusAddress}。

```

AGESA_DIMM_INFO_SMBUS DimmSmbus[] =
{
    // DIMM_SPD_MAP
    {0, 0, 0xA0},
    {0, 1, 0xA2},
    {1, 0, 0xA4},
    {1, 1, 0xA6},
    {2, 0, 0xA8},
    {2, 1, 0xAA},
    {3, 0, 0xAC},
    {3, 1, 0xAE},
    {4, 0, 0xA0},
    {4, 1, 0xA2},
    {5, 0, 0xA4},
    {5, 1, 0xA6},
    {6, 0, 0xA8},
    {6, 1, 0xAA},
    {7, 0, 0xAC},
    {7, 1, 0xAE},
    {0xFF, 0xFF, 0xFF}
};

```

图 2-8 内存 SPD 地址默认对应关系

```

OEM_DIMM_INFO_SMBUS      mOemDimmSmbus[]={
|  {0xFF,0xFF,0xFF,0xFF}    //End
};

```

图 2-9 内存 SPD 地址配置示例

当前 CRB 默认 CPU 端与主板端内存 Channel 是一一对应的关系, 例如 CPU 端 Channel 0/A 对应的是主板端 Channel 0/A 如图 2-10, 当主板硬件设计为 CPU 端 Channel 与主板 channel 定义不一致时, 例如 CPU 端 channel 3/D 对应的是主板 channel 1/B, 而默认为 CPU 端 channel 3/D 对应的是主板 channel 3/D 需在路径 XorcoreBoardPkg\Hygonxxxx\DxioAndGpio\Pei\HygonCpmOemInitPeim.c 处修改, 图 2-10 数组变量 2 个 1 组定义为 RequestedChannelId 和 TranslatedChannelId, RequestedChannelId 为 CPU 端 Channel ID (0-7), TranslatedChannelId 为主板 channel ID (0-7), 如非一一对应关系则根据硬件设计修改。

```

UINT8  HygonDM1SLTChanXLat[]      = {0x0,0x0, 0x1,0x1, 0xFF,0xFF};

```

图 2-10 内存 Channel 对应转化配置

当前 CRB 默认 CPU 端 Dimm 和主板 Dimm 定义相反, 例如 CPU 端 DImm 0 对应主板端 Dimm 1, 若 CPU 端内存 Dimm 和主板 Dimm 一致即 CPU 端 DImm 0 对应主板端 Dimm 0, 需在路径 XorcoreBoardPkg\Hygonxxxx\DxioAndGpio\Pei\HygonCpmOemInitPeim.c 图 2-11 处修改, 图示数组

4 个 1 组定义为{ SocketId, ChannelId, DimmId, BoardDimmId}。

```
OEM_DIMM_INFO_XLAT      HygonDM1SLTDimmXLat[]={
|   {0xFF,0xFF,0xFF,0xFF}    //End
};
```

图 2-11 内存 Dimm 对应转化配置

## 2.4 USB Porting

USB 需要配置 OC Pin 与 USB 端口的对应关系，配置在文件 TSXhciService.c 图 2-12 处，数组定义为 Ocport (bit0:3-USB 端口, bit4:7-USB 端口所在 Die) 和 Ocpin, 例如{0x01,0x01}是 Die0 上的 USB2.0 port1 与 Ocpin 1 对应，需要根据硬件设计进行配置。

```
USB_OC_DATA USBOCCfgTableDM1SLT[]={
|   //CPU0
|   {0x01,0x01},
|   {0x02,0x02},
|   {0x03,0x03},
|   {0x04,0x02},
|   {0x05,0x01},
|   {0x07,0x03},
|   //End
|   0xFF
};
```

图 2-12 USB OC 映射配置

1. USB OC 主要是起到过流保护的作用，Hygon CPU 中 DM1、SL1R2、SL1 引出的 port 数量都不一样，具体如下：  
DMI (1Die): Die0 上有 1 个 USB 控制器，引出 4 个 USB2.0 和 4 个 USB3.0;  
SL1R2(2Die): Die0 和 Die1 上各引出 4 个 USB2.0 和 4 个 USB3.0;  
SL1 (4Die): Die0 和 Die1 上各有一个 XHCI 控制器，海光 1 号和 2 号每个 XHCI 支持 2 个 USB2.0 和 2 个 USB3.0，海光 3 号每个 XHCI 支持 4 个 USB2.0 和 2 个 USB3.0;  
Porting 时 USB Port 号从 Port0 开始递增先将所有的 USB2.0 编完号再到 USB3.0。
2. USB\_CONF\_DATA 的 Porting 主要是配置 USB 信号，此处配置偏差过大可能会导致 USB 设备无法识别，或者 KVM 键盘无法使用。

## 2.5 DXIO 与 GPIO Porting

DXIO 及 GPIO 配置是在路径 XorcoreBoardPkg\Hygonxxxx\DxioAndGpio 的 PortingGuide-V101-<项目>.xlsx 文档修改，配置方式见 2.5.1 和 2.5.2，修改完将该文件路径加入根目录下文件 PortingGuide.bat 中，执行 PortingGuide.bat，执行完在 XorcoreBoardPkg\Hygonxxxx\DxioAndGpio\Pei\HygonCpmOemInitPei 目录下会生成 PcieBifurcationxxx.h 和 AutoGpioTablexxx.h 文件。

PortingGuide-V101-<项目>.xlsx 文档包含 4 个页面，HEADER TABLE，PCIE TABLE，GPIO

TABLE, XXX DXIO, HEADER TABLE 为基础配置页面, PCIE TABLE 为 PCIE 配置页面, GPIO TABLE 为 GPIO 配置页面, XXX DXIO 为硬件 DXIO 配置页面（此页面无需修改）。

HEADER TABLE 配置如表 2-3, 除 Item 其余均为必填项, Skuid 没有可填 N/A, 根据主板配置修改 Nsocket 和 DieNum。

表 2-3 HEADER TABLE

| Item | Ver | BoardId | BoardName | Skuld | BoardRev | Nsockets | DieNum |
|------|-----|---------|-----------|-------|----------|----------|--------|
| 0    | 101 | 11      | DM1SLT    | 1,4,5 | 0        | 1        | 1      |

2.5.1 DXIO porting

图 2-13 为 DXIO 配置 Lane 与 PCIE Controller 对应的关系图。

| Die | Lanes | Package Group | PCie controller | BIOS Mapping |
|-----|-------|---------------|-----------------|--------------|
| 0   | 0-15  | P0            | Pcie0 (Type A)  | 0-15         |
| 0   | 16-31 | G0            | Pcie1 (Type B)  | 16-31        |
| 1   | 0-15  | P1            | Pcie0 (Type A)  | 32-47        |
| 1   | 16-31 | G1            | Pcie1 (Type B)  | 48-63        |
| 2   | 0-15  | G2            | Pcie0 (Type A)  | 64-79        |
| 2   | 16-31 | P2            | Pcie1 (Type B)  | 80-95        |
| 3   | 0-15  | G3            | Pcie0 (Type A)  | 96-111       |
| 3   | 16-31 | P3            | Pcie1 (Type B)  | 112-127      |

图 2-13 DXIO 配置

DXIO 配置在文档 PortingGuide-V101-< CRB 项目>.xlsx 中 PCIE TABLE 页面修改, PCIE TABLE 页面包含表 DefaultPciePortTable, PortTableValue, HotPlugCfgTable, PortPhyTable, 主要配置在 DefaultPciePortTable。

表 2-4 PCIE TABLE

| Skid | Socket | DieNum | StartLane | EndLane | PortValueName                | Decrsiption |
|------|--------|--------|-----------|---------|------------------------------|-------------|
| 0    | 0      | 0      | 0         | 0       | PcieV01:A03S00D01F04<br>"NA" | BMC VGA     |

下述为表 2-4DefaultPciePortTable 各项定义简述,

Skid: 区分相同 Lane 复用功能的标志。

Socket: PCIE 或 SATA 配置所在 Socket;

DieNum: PCIE 或 SATA 配置所在 Die;

StartLane: PCIE 或 SATA 的 Lane 起始地址;

EndLane: PCIE 或 SATA 的 Lane 结束地址;

PortValueName: PortValueName 的配置和 PortTableValue 配置对应, 例如: PcieV01:

A03S00D01F04, PcieV01 对应配置位于 PortTableValue 的 ValueName 中 PcieV01（可根据硬件设计选择对应配置）所在行, A03 的定义为 Bit0: 0 为非板载, 1 为板载, Bit1: 0 为在显示设备, 1 为隐藏设备, S00 定义为 Slot ID, 此 ID 不可重复且只需外接 PCIE 端口配置, D01F04 定义为 PCIE 设备指定的 Root Port 的 Device 和 Function 地址（参考 PPR）。

Description: 描述 PCIE 或 SATA 的注释。

1. DXIO 配置表中配置为 PCIe Unued 的 lane 一定要为其指定 root port，如果将 Dev 和 Fun 号设置为 0 0，会导致相邻的一些 lane 上的设备识别或功能异常；
2. 对于属于同一个 PHY 的 lane 必须配置为同一种协议，否则有可能导致该 PHY 上的设备无法正常识别；
3. 配置 GPIO 时需要留意 GPIO 所在 DIE 的对应关系，二者对应关系出错可能会导致 hang 机。

### 2.5.2 GPIO Porting

GPIO 配置在文档 PortingGuide-V101-<CRB 项目>.xlsx 中 GPIO TABLE 页面修改。

GPIO 页面包含 GpioCfgTable 配置表，配置项简述如表 2-5。

表 2-5 GPIO 配置变量定义

|                |                  |
|----------------|------------------|
| Socket         | GPIO 所在 Socket   |
| Die            | GPIO 所在 Die      |
| PinNum         | GPIO 编号          |
| GPIO Ball Name | GPIO 描述          |
| Muti Function  | GPIO 复用功能        |
| Function       | 选择硬件设计采用 GPIO 功能 |
| Output         | GPIO 输入输出配置      |
| Pull           | GPIO 高低电平配置      |
| Descrption     | GPIO 主板设计功能描述    |

除上述修改外还需定义一些特殊功能的 GPIO pin，

PcdGpioRecoverPin 为 BIOS Recovery 功能对应的 GPIO 所在 Socket，Die，Pin；

PcdGpioPasswordClearPin 定义了清除密码所选 GPIO 所在 Socket，Die，Pin。

PcdGpioSkuPin 为 BIOS 对应的 SkuId 所在的 GPIO 所在 Socket，Die，Pin，三个一组；

PcdGpioCpldResetBmcPin，为 BIOS 控制 cpld reset bmc 的 GPIO 所在 Socket，Die，Pin；

PcdBmcReadyPin，为 BIOS 读取 BMC ready 状态的 GPIO 所在 Socket，Die，Pin；

PcdGpioBIOSPostCmpPin 为 Complete 信号所在的 GPIO 所在 Socket，Die，Pin；

PcdGpioBIOSPostCmpValue 为 Completemin 的 Func。

如果 Post CompletePin 使用了串口 0/1 的流控 pin，还必须设置

PcdUart0FlowControlPin/PcdUart1FlowControlPin 为 FALSE，以避免 PI 代码修改此 Pin。

正常 Power On 后，需检查 GPIO 是否都配置正确，使用 HART 工具，执行 HART -gpio -dump 命令，读取所有 GPIO 设定，对照 Porting Guide，检查所有 GPIO。

### 2.6 BoardId Porting

将申请的 BoardId 定义在 XorcoreBoardPkg\Hygonxxxx\Include\Board.h 中，如图 2-14

```
typedef enum {  
    HYGON_32DB16 = 43,  
} BOARD_ID;
```

图 2-14 BoardId 定义





ABL 的串口是单独控制的，在个项目的 ApcbCustomizedDefinitions.h 中 AmdABLDebug\_SUPPORT 为 1 表示输出 ABL log，AmdABLDebug\_Select 用于选择使用哪个串口输出，1 代表 UART0，2 代表 USRT1，UART 的选择要与各项目编译脚本中的串口选择一致。

```
#define AmdABLDebug_SUPPORT 0
#define AmdABLDebug_Select 2
```

图 2-21 ABL 阶段 UART 配置

当主板使用 I2C 与外设连接时，需要在 XorcoreBoardPkg\Hygonxxxx\HygonCpm.inc.dsc 中 porting 对应的 pcd 为 enable，如图 2-22，此选项为 enable 会做如下事情：

1. I2C 设备以 ACPI 的形式上报给 OS，这样 Linux 和 Windows 才会加载 I2C 设备的驱动；
2. 将 I2C 与 GPIO 的服用引脚配置为 I2C；
3. 修改 I2C 控制器到设备的保持时长。

```
gUefiHygonPkgTokenSpaceGuid.DefaultI2c0Config|0x01
gUefiHygonPkgTokenSpaceGuid.DefaultI2c1Config|0x01
gUefiHygonPkgTokenSpaceGuid.DefaultI2c3Config|0x01
gUefiHygonPkgTokenSpaceGuid.DefaultI2c4Config|0x01
gUefiHygonPkgTokenSpaceGuid.DefaultI2c5Config|0x01
```

图 2-22 I2C 配置

对于 SIO 的 Porting，CRB 默认只是用了 AST2500 的 Port 80 功能，其他需要的功能可以在 XorcoreModulePkg\IO\AST2500\AST2500PeiInit.h 文件中设置，如图 2-23。

```
#define AST2500_KEYBOARD_PRESENT 0
#define AST2500_MOUSE_PRESENT 0
#define AST2500_SERIAL_PORT1_PRESENT 0
#define AST2500_SERIAL_PORT2_PRESENT 0
#define AST2500_SERIAL_PORT3_PRESENT 0
#define AST2500_SERIAL_PORT4_PRESENT 0
#define AST2500_GPIO_PORT_PRESENT 1
#define AST2500_SWC_PRESENT 0
#define AST2500_ILPC2AHB_PRESENT 1
#define AST2500_MAILBOX_PRESENT 1
```

图 2-23 AST2500 功能配置

IO decode 及 port 80 配置在项目的 bat 文件中配置，如图 2-24。其中 AST2500\_CONFIG\_INDEX 和 AST2500\_CONFIG\_DATA 的配置需查看原理图 2500 HW Strap 引脚 RGMII1TXD2\_GPIO4，为低电平时选 0x2E, 0x2F，为高电平时选 0x4E, 0x4F，曙光 G40 硬件规范应选 0x4E, 0x4F。

AST2500\_CONFIG\_GPIO\_PORT80 为选择 Port 80 等使用那组 GPIO，需查看原理图确认 LED 的连接方式，若是使用 GPIOM 组设为 0xC，若是使用 GPIOH 组则设为 0x7，曙光 G40 硬件规范应选 0x7。

```
echo DEFINE AST2500_CONFIG_INDEX = 0x4e >> %PLATFORMID_FILE%
echo DEFINE AST2500_CONFIG_DATA = 0x4f >> %PLATFORMID_FILE%
echo DEFINE AST2500_CONFIG_GPIO_PORT80 = 0x07 >> %PLATFORMID_FILE%
```

|   |      |      |
|---|------|------|
| AST2500_CONFIG_INDEX <sup>1</sup>       | 0x2E | 0x4E |
| AST2500_CONFIG_DATA                     | 0x2F | 0x4F |
| AST2500_CONFIG_GPIO_PORT80 <sup>2</sup> | 0x7  | 0xC  |

图 2-24 AST2500 IOdecode 及 port 80 配置



NCT 系列的 SIO 需要在批处理文件<项目>\_Dhyana.bat 中图 2-25 处为 SIO 地址端口配置。

```
echo DEFINE NCT6796D_CONFIG_INDEX = 0x2E    >> %PLATFORMID_FILE%
echo DEFINE NCT6796D_CONFIG_DATA  = 0x2F    >> %PLATFORMID_FILE%
echo DEFINE NCT6796D_SIO_SUPPORT   = 1      >> %PLATFORMID_FILE%
```

图 2-25 SIO 配置

FastVideo 可以在 PEI 阶段点亮 VGA，目前主要支持 AST2500 和景嘉微显卡，主要在 XorcoreBoardPkg\Hygonxxxx\HygonCpm.inc.dsc 中配置如下 Pcd。

|                             |   |
|-----------------------------|---|
| <b>PcdFastVideoEnable</b>   | 1: 使能FastVideo<br>0: 关闭FastVideo                                      |
| <b>PcdFastVideoTextMode</b> | 0: 关闭显卡Text模式<br>1: 打开显卡Text模式  |
| <b>PcdFastVideoDieNum</b>   | FastVideo显卡所在的Die编号   |
| <b>PcdFastVideoBusNum</b>   | 设为0，代码会动态更新   |
| <b>PcdFastVideoDevNum</b>   | 需与PortingGuide*.xlsx中<br>PCIE TABLE设置的一致                              |
| <b>PcdFastVideoFunNum</b>   | 需与PortingGuide*.xlsx中<br>PCIE TABLE设置的一致                              |
| <b>PcdFastVideoPeiMmio</b>  | MMIO范围，低16bit为base，高16bit为limit<br>如0xDD00DC00代表0xDC000000-0xDD000000 |
| <b>PcdFastVideoPeiMmio1</b> | VGA BAR1的MMIO基址，若PcdFastVideoPeiMmio为0xDD00DC00<br>此处填0xDC000000      |
| <b>PcdFastVideoPeiMmio2</b> | VGA BAR2的MMIO基址，若PcdFastVideoPeiMmio为0xDD00DC00<br>此处填0xDD000000      |

图 2-26 SIO 配置

PCI\_SLOT\_MAP\_LIST 这个 Token 放在项目的编译脚本中，一般用来设置 PCIE 的属性，最终会体现在 Setup 下 Advanced->PCIe UEFI Driver Policy->PCIe Device Per-Cnreol 下，如图 2-27 和图 2-28。

```
::Define by platform,example Hygon{skid,sltatr,socket,die,dev,func,portindex,DevDefValue,PortDefValue,SriovDefValue}
:: {          onboard vga and onboard I350 ,                                ,over }
echo DEFINE PCI_SLOT_MAP_LIST = {0xFF,0x03,0,0,1,5,0xFF,0xFF,0xFF},{0xFF,0x01,0,0,1,3,0xFF,0xFF,0xFF},{0,0,0,0,0,0,0xFF,0xFF,0xFF}
```

图 2-27 PCI\_SLOT\_MAP\_LIST 配置

| Skld  | Sltr   | Socket               | Die                   | Dev   | Func   | PortIndex  | DevDefValue                                      | PortDefValue   | SriovDefValue   |
|---|--|----------------------|-----------------------|---|--|--|--|--|---|
| Skuld<br>需与<br>PortingGuide*.xlsx中<br>保持一致<br>0xFF表示任意SKU | BIT0<br>0: 设备为外插<br>1: 板载<br>BIT1<br>0: 在Setup下不<br>隐藏<br>1: 在Setup下隐<br>藏 | 设备所在的<br>CPU<br>从0开始 | 设备所在的<br>Die<br>取值0-4 | 设备上游PCI桥的Dev<br>号<br>需与<br>PortingGuide*.xlsx中<br>PCIE TABLE设置的一<br>致 | 设备上游PCI桥的<br>Func<br>号<br>需与<br>PortingGuide*.xlsx中<br>PCIE TABLE设置的一<br>致 | 设备的第几个port<br>一般为PCI设备的<br>Func号<br>0xFF表示任意port | 设备默认开关<br>0: 关闭<br>1: 打开<br>0xFF: 使用Setup默<br>认值 | 设备Port的OpRom默认<br>开关<br>0: 关闭<br>1: 打开<br>0xFF: 使用Setup默<br>认值 | 设备SRIOV默认开<br>关<br>0: 关闭<br>1: 打开<br>0xFF: 使用Setup默<br>认值 |

图 2-28 PCI\_SLOT\_MAP\_LIST 结构体定义

## 2.8 编译脚本 Porting

参考 CRB 项目建立对应项目的编译脚本（Hygonxxxx.bat），主要定义 BIOS 和 Platform 等名称，如图 2-29 和图 2-30，部分定义也可放在 XorcoreBoardPkg\Hygonxxxx\HygonCpmDefines.inc.dsc 中。

```
set CUSTOMER_NUMBER1=0
set PROJECT_NUMBER=HCB
set CUSTOMER_NUMBER23=00
set SUBPROJECT_NUMBER=4
set PLATFORM_UPGRADE_EXT_NUMBER=0
set VERSION_NUMBER=12
```

图 2-29 BIOS 名称配置

```
echo DEFINE PLATFORM_SELECT      = 43      > %PLATFORMID_FILE%
echo DEFINE AST2500_CONFIG_INDEX = 0x4e    >> %PLATFORMID_FILE%
echo DEFINE AST2500_CONFIG_DATA  = 0x4f    >> %PLATFORMID_FILE%
echo DEFINE AST2500_CONFIG_GPIO_PORT80 = 0x07 >> %PLATFORMID_FILE%
echo DEFINE NSOCKETS              = 1      >> %PLATFORMID_FILE%
echo DEFINE BIOS_TAG              = %BIOS_NAME% >> %PLATFORMID_FILE%
echo DEFINE FCH_UART_PORT_SELECT  = 2      >> %PLATFORMID_FILE%
echo DEFINE FAST_READ_SUPPORT     = 0      >> %PLATFORMID_FILE%
echo DEFINE FLASH_MMIO_READ_SUPPORT = 1    >> %PLATFORMID_FILE%
echo DEFINE PLATFORM_NAME_STR     = L"R3240H0" >> %PLATFORMID_FILE%
echo DEFINE BMC_LAN_COUNT         = 0x02    >> %PLATFORMID_FILE%
echo DEFINE BMC_LAN1_CHANNEL_NUMBER = 0x8    >> %PLATFORMID_FILE%
echo DEFINE BMC_LAN2_CHANNEL_NUMBER = 0x1    >> %PLATFORMID_FILE%
echo DEFINE PROJECT_UNIFIED_PORTING = TRUE    >> %PLATFORMID_FILE%
echo DEFINE PROJECT_PORTING_PATH   = Hygon32DB16 >> %PLATFORMID_FILE%
set PROJECT_ID=Hygon32DB16
```

图 2-30 编译脚本主要定义内容

## 2.9 功能 Porting

对于特殊机型来说，并不需要主干上所有的功能，比如工作站，因此需要对主干上一些功能的开关进行 Override，可以在 XorcoreBoardPkg\Hygonxxxx\HygonCpmDefines.inc.dsc 中进行 Token 的重定义，如图 2-31。

```
DEFINE DEBUG_FUNCTION_SUPPORT      = FALSE
DEFINE POST_COMPLETE_SUPPORT      = TRUE
DEFINE HDDPASSWORD_SUPPORT        = FALSE
DEFINE SECURE_BOOT_ENABLE         = FALSE
DEFINE TPM_SUPPORT                = FALSE
DEFINE BMC_WARM_RESET_SUPPORT     = FALSE
DEFINE BMC_INIT_DELAY             = FALSE
DEFINE BMC_TIMEZONE_SUPPORT       = FALSE
DEFINE HARDWARE_MONITOR_SUPPORT   = FALSE
DEFINE DEBUG_ALWAYS_DEBUG_ENABLE  = FALSE
DEFINE S3_SUPPORT_ENABLE          = TRUE
```

图 2-31 功能 Token 重定义

## 2.10 Smbios Porting

如果要对 smbios 的默认值进行定义，可以在

XorcoreBoardPkg\Hygonxxxx\DxioAndGpio\Dxe\OemSmbios\OemSmbios.c 中对相应的 type 进行 pcd

的重新赋值，如图 2-32。

```
Status = GpioGetSkuId(&BoardSkuId);
switch (BoardSkuId) {
    case 0:
        ManufacturerStr = L"Sugon";
        ProductNameStr = L"W330-H35A1";
        break;
    case 1:
        ManufacturerStr = L"Sugon";
        ProductNameStr = L"W330-H35A2";
        break;
    case 4:
        ManufacturerStr = L"Suma";
        ProductNameStr = L"W330H0";
        break;
    case 5:
        ManufacturerStr = L"Suma";
        ProductNameStr = L"W335H0";
        break;
}
```

图 2-32 Smbios 默认值定义

附录一 缩略语与缩写

| 缩略语  | 解释说明   |
|------|--|
| BIOS | 基本输入/ 输出系统（BASIC INPUT/OUTSYSTEM）的缩写。            |
| GPIO | 通用型之输入输出的（General-purpose input/output）简称。       |
| CRB  | （custom reference board）客户参考板。                   |
| DXIO | （Distributed IO Crossbar Subsystem）分布式 IO 交叉子系统。 |