

AG10K FPGA 调试的建议

1. PHY

PHY 芯片时钟，一般在 Tx 方向，数据与 clk 会使用边沿对齐，Rx 方向是中央对齐的方式。

a) Tx:

Edge Aligned，除了在数据端使用 DDIO_OUT，在 clk 上也要用上 DDIO_OUT，以保证时钟和数据的对齐。

```
ddio_out inst (  
    .aclr(aclr),  
    .datain_h(1'b1),  
    .datain_l(1'b0),  
    .outclock(outclock),  
    .outclocken(outclocken),  
    .dataout(dataout)  
);
```

Center Aligned，不用 DDIO，把时钟反向，如：assign p0gtxc = !pll_clk;

然后 asf 文件中加入延迟：

```
set_instance_assignment -name OUT_DELAY -to p0tx* 1'b1 -extension
```

```
set_instance_assignment -name OUT_DELAY -to p1tx* 1'b1 -extension
```

b) Rx:

在 asf 文件中加入：

```
set_instance_assignment -name IN_DATA_DELAY -to p*rx* 3'b110 -extension
```

p*rx*应包括所有的 rx 数据输入（每个 PHY 有 4 个 data 和 1 个 ctrl）

加入以上设置，在 sdc 文件中可以省略输入输出延迟的约束。

这只针对 rxc 使用 Global Clock 时的情况。其它需具体根据时序分析延迟。

DDIO 的应用除 data 外，dv 信号有可能不会被 Supra 识别，这时，需在 ve 文件中对 dv 信号

加入 DDIO_IN 参数，如：

```
P0_rxdv pin_xx DDIO_IN
```

2. SDRAM

——方法（1）

a) 输出端：

sdc 文件中加入以下时序约束（以下红色标示的需改）：

```
set_sdr_src_pin {mclk_inst|altpll_component|auto_generated|pll1|clk[2]}
```

```
set_sdr_clk sdrclk_output
```

```
set_sdr_ctrl_data {sdramcs sdramba* sdrmcas sdramras sdramwe sdramadd* sdramdatam*}
```

```
create_generated_clock -name $sdr_clk -source [get_pins $sdr_src_pin] [get_ports $sdr_clk]
```

```
set_output_delay -clock $sdr_clk -max 1.5 [get_ports $sdr_ctrl_data]
```

```
set_output_delay -clock $sdr_clk -min -1 [get_ports $sdr_ctrl_data]
```

sdr_clk 是输出到 SDRAM 的时钟，不再需要相位移动；

mclk_inst|altpll_component|auto_generated|pll1|clk[2] 替换为实际的 sdr_clk 时钟管脚名称；
sdram* 替换为实际的 SDRAM 数据和控制端口。

对于 16 位 SDRAM，每一组 SDRAM 的时钟和数据控制端口都要加入以上约束。

b) 输入端：

数据输入端需加入一级流水线 Pipeline。在数据端口使用 FAST_INPUT_REGISTER 设置，来优化 IO 到寄存器的时序；

Sdc 时序约束可以不加。

——方法（2）

同时在 SDRAM 的端口使用 IO register，Sdc 时序约束可以不加。

输入端：在数据端口使用 FAST_INPUT_REGISTER 设置；

输出端：DATA 端口使用 FAST_OUTPUT_REGISTER 和 FAST_OUTPUT_ENABLE_REGISTER 设置，其它控制管脚使用 FAST_OUTPUT_REGISTER 设置。

输出给 SDRAM 的时钟相对于 Controller 的时钟加一个反向，即相移 180 度。

3. PLL

AGM FPGA 在配置成功时，PLL 已经完成锁定，lock 信号已经变高；如果原设计中用 lock 信号输出实现系统 reset 的复位功能，就不能正确完成上电复位；同时，为了保证 PLL 相移的稳定，我们需要在 PLL 启动后做个延时的复位，设计中可以参考下面 Verilog 代码，这里需要 PLL 的 areset 或 pfdena 引出，通过复位信号控制。

```
reg [5:0] reset_init = 6'b0 /* synthesis syn_preserve = 1*/;
wire init = reset_init[5];
always @ (posedge inclk) begin
    if (!init) begin
        reset_init <= reset_init + 1'b1;
    end
end

PLL inst (
    .inclk(inclk),
    .areset(!init),
    .pfdena(init),
    ....
);
```

4. Flash 管脚

如果在设计中对配置 FLASH 进行读写，则要在 Quartus II 中把 4 个 SPI 配置管脚设置为“Use

as regular I/O”。

5. 上升和下降沿采样建议

如果在设计中用时钟下降沿采集的，而他的上一级是上升沿，这样两者之间就只有半个时钟 period。这样时序会紧张，建议用同沿。

6. VccINT

如果 FPGA 的资源利用率达到 90%以上，且时序要求较高，建议把 1.2V 的 VccINT 电压，升高到正常电压范围的 1.25V，会有效提高 FPGA 性能和稳定性。

7. FLASH 编程文件

烧写配置用 SPI FLASH，使用 XXX_master.bin，可通过 FLASH 编程器烧写。如使用其它兼容 Altera POF 文件的烧写工具，需注意 Altera 文件 MSB-LSB 为反向，需倒置后才可正确烧写。

8. AS 在线烧写 FLASH

如果使用 USB-Blaster 通过 AS 接口在线烧写配置用 SPI FLASH，由于 AG10K 的 DCLK、nCSO、ASDO 管脚并未输出三态，需要对 FLASH 加载管脚串接电阻使用；

方法一：FLASH 加载管脚分别串接 1KΩ 电阻，使得 USB-Blaster 驱动能力高过这些管脚的输出能力，且 DCLK 时钟降低频率，即可实现正常烧写和配置。

降低 DCLK 时钟频率方法：在 DesignName.asf 文件中添加如下约束可降低 DCLK 加载频率

```
set_global_assignment -name OSC_FREQUENCY 0 -extension
```

（其中 default 值为 3）

方法二：将 FLASH 加载管脚 DCLK 使用 500 欧姆电阻串接，其余 3 个管脚使用 1K 欧姆电阻串接，不需要降低 DCLK 时钟频率，即可实现正常烧写和配置；

如 FPGA 有 JTAG 接口，则可直接通过 Supra 软件烧写 XXX_master.svf 或 XXX_master.prg 文件到已连接的 SPI FLASH 中，无需通过 AS 接口，更为方便。

9. CLK 输入管脚

CLK 专用输入管脚，没有内部上拉电阻，所以在悬空没有输入状态时，会得到 0 的电平数据。这和 Altera EP4C 的高电平结果不同。如果正常输入时钟或信号，则没有影响。

10. Latch 锁存器

支持 latch，但建议设计不用 latch，异步电路时序不容易分析操作，建议不使用。

11. conf_done

所有 fpga 逻辑必须发生在 fpga 配置完成之后（conf_done 为高），否则 fpga 无法稳定接受开始工作。

12. AG10KSDE176 IO 驱动电流

由于片内集成 3.3V SDRAM，只能 3.3V IO 供电，IO 电源 pin 供电负载较大，需把 SDRAM 连接的相关 IO 驱动电流设为最小值 4mA，可在 asf 文件中加入：

```
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to *
```

一般 PHY 的 Tx_CLK 由于频率高，需要较强驱动能力，驱动电流最好设为最大值 16mA。

(Notes: 需确认在 Quartus II 软件中已将 IO 电压设置为 3.3V，否则使用 default 的 2.5V，

AG10KSDE176 的电流设置为正常的 2 倍)

13. 除法器

如设计中使用了除法器，需对除法器的路径设置 `multicycle`。除法器的 `latency` 即为 `multicycle` 的时钟数。如无法满足 `timing`，建议增加除法器的 `latency`。

14. 烧写文件压缩

AG10K 支持 AS 模式的烧写文件压缩，以减少存储空间。但不同设计压缩出的文件大小不一，如需统一文件存储空间，可选非压缩方式。

AG10K 不支持 PS 模式的烧写文件压缩。原设计里如果选了压缩，请取消；或者在 AGM 项目中的 `DesignName.asf` 中加入：

```
set_global_assignment -name ON_CHIP_BITSTREAM_DECOMPRESSION OFF
```

15. ve 文件的更新

AG10K 用 176 封装，我们会用 `ve` 文件分配管脚，管脚名称后面可以加入对 IO 寄存器布局的约束，比如：

```
p1rxdv PIN_110 DDIO_DIST:8
p1rx[3] PIN_122 DDIO_DIST:8
p1rx[2] PIN_120 DDIO_DIST:8
p1rx[1] PIN_119 DDIO_DIST:8
p1rx[0] PIN_115 DDIO_DIST:8
```

这样就把 IO 的寄存器故意设为距离管脚 8 个 Tile 的位置，或者取其它数值，用于加入 IO 的不同延迟。默认不用这个参数的话，是 `DDIO_DIST:1`。

16. bin 文件的组合与转换

AG10K 编译完生成的 FLASH 烧写文件 `DesignName_master.bin`，如果要加入其它 FLASH 的 `bin` 文件（如 MCU 程序等），可以通过 `dos` 指令，把 2 个文件连接，生成 `xxx.bin`：

```
copy /b DesignName_master.bin+123.bin xxx.bin
```

`bin` 文件如果希望从 JTAG 写入 SPI FLASH，可以执行：

```
af_gen --source_bits xxx.bin --from_binary
```

生成 `xxx.svf` 和 `prg`，用于通过 JTAG 烧写。

现在也可使用 `Supra` 中的 `Generate` 功能（Generate programming files from a binary file）。

17. 3.3V IO

软件中要按实际 IO 电压正确设置。`Quartus II` 中对 IO standard 的默认为 2.5V，如果实际采用 3.3V，需要修改 IO 默认值为 3.3V。

AG10KSDE176，由于 SDRAM 为 3.3V，VCCIO 必须使用 3.3V，软件中也要设置正确。

18. 电源上电顺序

AG10K 3.3V VCCIO 电源应略先于 VCCINT 1.2V 和 2.5V 上电，这样保证 FPGA 在启动配置前，使得配置 FLASH（3.3V）启动完成，从而避免上电配置失败。

19. PLL 时钟输出

为了避免时钟域 skew 或 jitter，如果需要同频率同相位的时钟驱动不同功能，尽可能合并成

同一个时钟进行处理，不要采用多个同样的输出时钟这种方式，避免跨时钟域的情况出现。不同频的时钟，尽可能采用 RAM 或者 FIFO 交互，如果寄存器直接交互，尽可能通过设计保证时钟采样在数据中间位置。

20. 逻辑资源优化

当逻辑资源比较紧张时，推荐将同步复位改为异步复位，可以节省一定的逻辑资源；