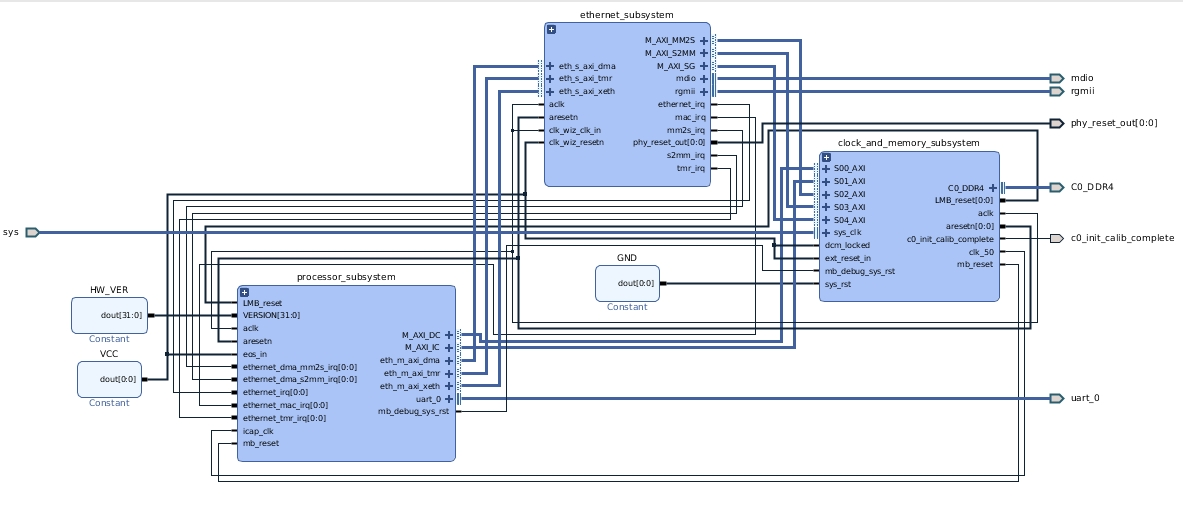
Microblaze golden镜像制作要点

仓库在

<https://gitee.com/figkey/xcku040_golden>

1. 要有升级功能

这样就要用到网络和DDR，在040上基本上bit文件就要有8MB以上。



最小系统的vivado模块，包括Microblaze(1M片上IC、DC)、MDM、串口、DDR（MIG）、以太网子系统（含一个32位定时器和DMA控制器）

1. Vivado方面的有关约束设置

我们一般采用的

set\_property CFGBVS VCCO [current\_design]

set\_property CONFIG\_VOLTAGE 3.3 [current\_design]

set\_property BITSTREAM.GENERAL.COMPRESS true [current\_design]

set\_property BITSTREAM.CONFIG.CONFIGRATE 50 [current\_design]

set\_property BITSTREAM.CONFIG.SPI\_BUSWIDTH 4 [current\_design]

set\_property BITSTREAM.CONFIG.SPI\_FALL\_EDGE Yes [current\_design]

set\_property BITSTREAM.CONFIG.CONFIGFALLBACK ENABLE [current\_design]

set\_property CONFIG\_MODE SPIx4 [current\_design]

就已经可以了。

这里不需要定义timer。因为给小了一启动就认为是fallback，给大了要等比较久。实际上可以在vitis这边动态设置这个timer。

如果实在需要在bit流定义timer，这样添加

set\_property BITSTREAM.CONFIG.TIMER\_CFG 32'h01FFFFFF [current\_design]

这个定义按xapp1296是15s

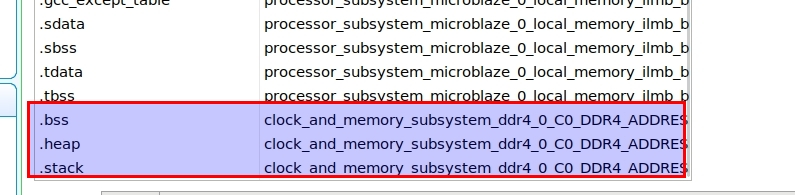
如果update镜像要在16MB以上地址，还应设置32位寻址，默认是24位寻址

set\_property BITSTREAM.CONFIG.SPI\_32BIT\_ADDR Yes [current\_design]

1. Golden镜像最好是不使用bootloader，指令运行在片上

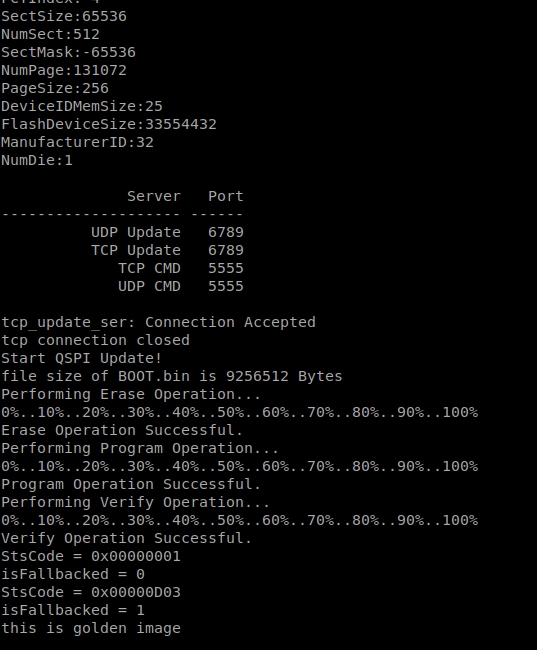
关键是.bss区要定位到DDR，heap和stack也一起定位到DDR，一般来说heap和stack大小可以定义为大于等于0xA000

而接收镜像用malloc动态分配实际上接收不了很大的镜像(即便给和很大的heap，大于8MB板端程序就挂了)，按照以前定义成数组就可以了.



1. Golden镜像烧录其他镜像，应刷到update镜像区。

目前还是从0开始刷的



1. 执行逻辑



启动之后检查BOOTSTS Register的fallback位

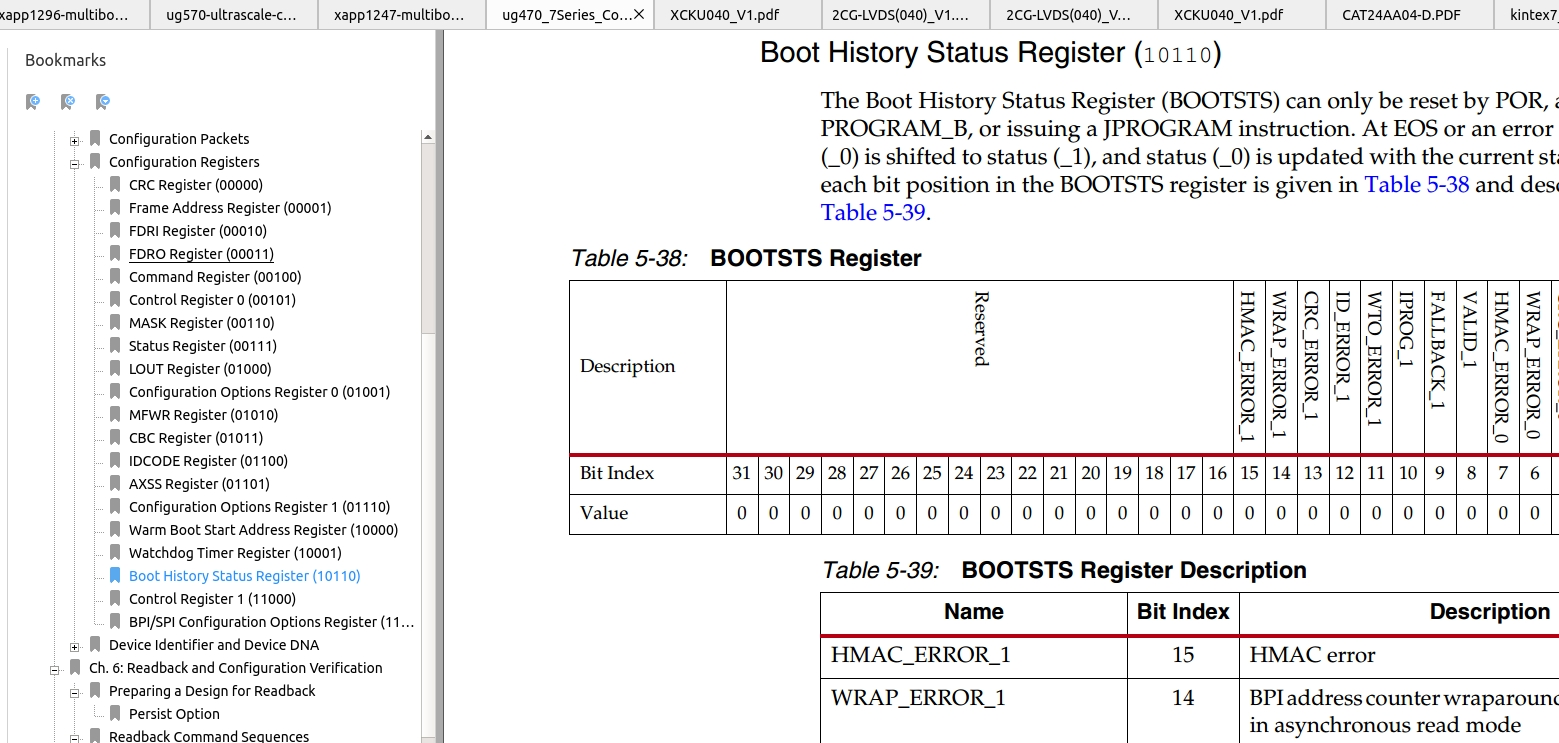
打印是否是fallback状态(后面的while是为了打印完整)

然后检查是否timer设置为0, 如果为0要给一个定时

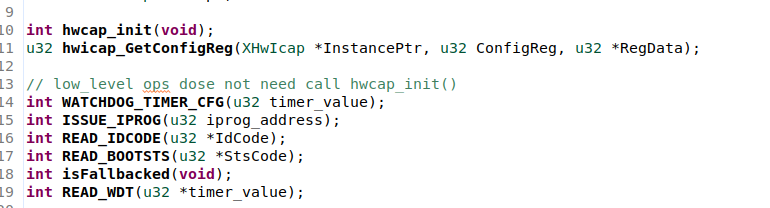
最后跳转到预定义的镜像地址

如果没有镜像，在定时器的定时到，就fallback回来goledn镜像

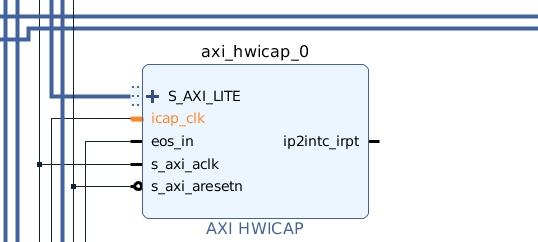
如果预定义的镜像损坏，crc报错也会fallback回来goledn镜像



1. Hwicap模块



提供icap操作接口来访问FPGA的配置寄存器，用于读取timer及镜像转跳地址设置



对应这个vivado模块，icap\_clk不大于50MHz

1. 对公共库的影响

由于xhwcap的vitis库定义的宏和bitmanip.h定义的一些库冲突，所以bitmanip.h进行了修改，连带xgpio\_i2c的库也需要修改，已经更新到公共库。

1. 参考资料

xapp1296,xapp1247,ug570,ug470,pg134