实验2请在实验1文档的基础上阅读

# 1 DMA模块介绍及其要求

## 1.1 DMA概述

DMA，Direct Memory Access，直接内存访问，是一种不经过CPU而直接从内存存取数据的数据交换模式。在DMA模式下，CPU只需要向DMA控制器下达指令（配置DMA寄存器），传输数据由DMA来完成，数据传送完再把信息反馈给CPU，这样能够减少CPU的资源占有率。DMA由硬件实现，从共用系统数据总线的角度看，DMA和CPU是竞争对手的关系，当完成一批数据的传输工作之后，可以向CPU发出中断请求，报告本次传输结束的同时，请示下一步的操作要求。

图2. 1 DMA传送系统示意图

DMA的典型工作模式有以下几个阶段：

⑴DMAC的初始化：在DMA操作开始之前，用户应根据需要先对DMAC编程，把要传送的数据字节数、数据在存储器的起始地址、数据的传送方式、要使用的DMAC通道号等信息送到DMAC，这就是DMAC的初始化。

⑵申请阶段：在初始化工作完成之后，若外部设备要求系统以DMA方式为之服务，则向DMAC发出请求信号DREQ；DMAC如果允许外部设备的请求，发出总线请求信号HRQ（HOLD REQUEST），申请占用总线。

⑶响应阶段：当检测到DMA发送的总线请求信号后，总线总裁器把总线的控制权给DMA，同时让CPU一侧的总线“浮空”，CPU释放总线控制权，同时以总线保持回答信号HLDA（HOLDACK）表明总线已让出；这样DMAC一侧则可以接管总线，成为系统的主控者。

⑷数据传输阶段：DMAC接管总线之后，一方面以DMA请求回答信号DACK通知发出请求的外部设备，使之成为被选中的DMA传送设备；同时DMAC行使总线控制权，向存储器发送地址信号、向存储器及外设发送读写控制信号，控制数据按初始化设定的方向传送。

⑸传送结束阶段：当数据传送完毕后，DMAC就产生一个“计数已到”或“过程结束”的信号，并发送给外设。外设收到此信号，认为它请求传送的数据已完毕，于是就撤销DMA请求信号DREQ，从而使得总线请求信号HRQ和总线回答信号HLDA相继变为无效，DMAC释放总线控制权，CPU接管总线。至此，一次DMA传送结束。如果需要，还可以用“过程结束”信号引发一个中断请求，由CPU去处理DMA传送结束后的事宜。

dma\_doc.pdf介绍了一个具体的DMA IP，比较复杂，大家可以适当参考。

## 1.2 DMA实验要求

本次实验对DMA设计要求如下：

首先同学根据学号尾数选择特定的图片，利用Matlab将图片转换成50×50×1约2KB的灰度图像矩阵供DMA使用，转换代码（img2txt.m）已经给出，同学们只需要直接执行代码并通过readmemh函数读入SoC即可。

在开始传输前，DMA接收CPU对于源地址，目的地址，搬运数据长度的配置信息，当这些寄存器信息更新后，DMA开始自行进行数据搬运。在搬运时，要求完成原图像的复原。另外如果有需要，同学们可以选择处理更大的图像矩阵，只需要在matlab代码中将50改为你需要的即可。

搬运结束后，将DMA中断拉高。要求至少实现下面寄存器的维护：（如果需要可自行维护其它寄存器）

1，源地址寄存器，指示搬运的起始地址，可读可写

2，目的地址寄存器，指示搬运的目的地址，可读可写

3，数据长度寄存器，指示搬运的数据长度，可读可写

4，状态寄存器，只读，指示配置完成，搬运完成等

在完成搬运后，在SRAM2中可通过fwrite函数将数据全部写入txt文件中，建议写成2500X1的矩阵形式。最后通过所给的另外一个Matlab代码（txt2img.m）将txt文件重新转换为图片。

在软件代码中，配置DMA寄存器以及编写中断服务程序；

要求在报告文档里解释清楚dma硬件模块的设计思路，需要有详细的波形介绍以及所维护的寄存器列表同时附上最后转换后Matlab中的Figure截图；并以附件形式附上具体的代码（硬件+软件）。

# 2 实验SOC介绍

## 2.1 基本框架



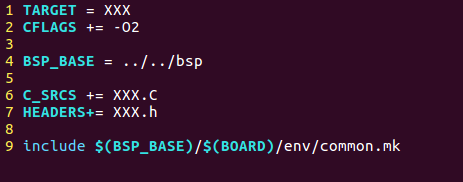
整个系统包含两部分，一部分是对存储器的访问，一部分是对外设的访问。

存储器方面，CPU和DMA都可以访问存储器。两者的访问请求通过ICB仲裁后再根据地址区间不同而分发给不同的模块。但是本实验中CPU对存储器的访问仅仅在于上电的时候（复位），因为复位时，指令PC会复位成0x0000\_1000，也就是对应ROM的地址空间。当程序正常运行时，CPU不会发起对存储ICB总线的请求，换句话说，本实验中没有总线仲裁理论上也可以正常工作；当DMA开始工作时（在SRAM之间搬运数据），会发起对SRAM的请求。对应的ICB请求会通过一个ICB转SRAM控制器的桥后接到SRAM。

外设方面，本实验中保留了I2C，UART，DMA三个外设。从CPU的请求同样会根据地址区间不同通过ICB-splt桥分发给对应的外设模块（这部分桥没有在上图中画出来）。其中，I2C和UART的代码已经给出，DMA需要大家自己写。

### 2.2 软件编写

大家在编写DMA的软件代码之前，记得参照demo\_i2c先在demo\_e200/hbird-e-sdk-master/bsp/hbird-e200/software目录下新建名为XXX的文件夹（名字随便），并在该文件夹中新建XXX.c和XXX.h（如果需要的话）的文件。然后新建Makefile文件，Makefile的写法如下：（注意TARTGET的值和文件夹名字要一致）



后续运行的方法请参考实验1说明文档中的2.4.4节。