32 位微控制器

HC32M120 系列的 DMA 控制器

适用对象

系列	产品型号
HC32M120	HC32M120J6TB HC32M120F6TB

目　录

1 摘要

本篇应用笔记主要介绍如何使用 HC32M120 系列芯片的 DMA 模块传输数据。

2 DMA 简介

什么是 DMA？

DMA（直接内存访问控制器）功能块可以不通过 CPU 高速传输数据。使用 DMA 能提高系统性能。

DMA 的重要特征？

DMA 总线独立于 CPU 总线，所以即便是在使用 CPU 总线的时候，DMA 也可进行传输操作。

3 HC32M120 系列的 DMA

3.1 简介

华大 HC32M120 系列 MCU 内部集成 DMAC 模块，能够在 CPU 不参与的情况下实现存储器之间，存储器和外围功能模块之间以及外围功能模块之间的数据交换。

3.2 说明

DMAC 总线独立于 CPU 总线，按照 AMBA AHB-Lite 总线协议传输。

拥有 2 个独立通道，可以独立操作不同的 DMA 传输功能。

每个通道的启动请求源通过独立的触发源选择寄存器配置。

每次请求传输一个数据块，数据块最小为 1 个数据，最多为 256 个数据。每个数据的宽度可配置为 8bit，16bit，32bit。

可以配置最多 1023 次传输。

源地址和目标地址可以独立配置为固定、自增、自减、支持以下功能四选一：源地址循环、源地址指定偏移量的跳转、目标地址循环、目标地址指定偏移量的跳转。

可产生 3 种中断：块传输完成中断，传输完成中断，传输错误中断。每种中断都可配置是否屏蔽。其中块传输完成，传输完成可作为事件输出，作为其他外围模块的触发源。

支持连锁传输功能，可实现一次请求传输多个数据块。

不使用时可设置进入模块停止状态以降低功耗。

3.2.1 寄存器介绍

1) DMA_EN：DMA 使能寄存器，使能或关闭 DMA 模块。
2) DMA_CHEN：通道使能寄存器，使能 DMA 通道，bit0~1 分别对应一个通道，写 1 使能，写 0 无效。
3) DMA_CHENCLR：通道使能复位寄存器，通道使能复位，bit0~1 分别对应一个通道，写 1 复位 CHEN 寄存器相应通道，写 0 无效。
4) DMA_INSTAT0~1：中断状态寄存器（传输请求溢出错误中断、传输错误中断、块传输完成中断、传输完成中断）。
5) DMA_INTMASK0~1：中断屏蔽寄存器，配置各中断是否屏蔽。
6) DMA_INTCLR0~1：中断复位寄存器，清空中断状态标志位。
7) DMA_CHSTAT：通道状态观测寄存器。
8) DMA_TRGSEL1~2：触发源选择寄存器，配置各通道启动传输的触发源，配置前需打开 CLK_FCG 寄存器的 AOS 位。
9) DMA_SAR0~1：源地址寄存器，配置传输源地址。
10) DMA_DAR0~1：目标地址寄存器，配置传输目标地址。
11) DMA_CHxCTL0(x=0~1)：数据控制寄存器，配置传输次数、数据块大小、传输数据宽度、连锁传输相关配置（使能、模式、链指针）。
12) DMA_CHxCTL(x=0~1)：通道控制寄存器，配置源和目的地址的更新方式，以及重复和不连续传输功能相关配置。

3.2.2 工作流程介绍

在本章节主要介绍 DMA 传输模式的设置和运行流程。

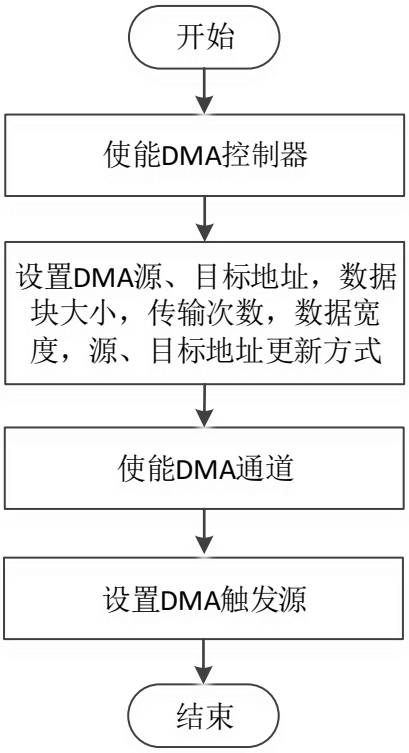
1) 基本传输

不配置重复和不连续功能，且不配置连锁传输的功能的传输。

该传输可以配置源地址、目标地址的更新方式，固定、自增或者自减。存储器之间数据交换、单个寄存器和存储器之前的数据交换、单个寄存器和单个寄存器之间数据交换等可以选择该基本传输模式。

具体样例可参照 DDL 的 DMA 模块样例 dmac_base。

基本的配置流程如下图，其中 DMA 源、目标地址等的配置可以直接调用 DMA_ChannelCfg 函数实现。

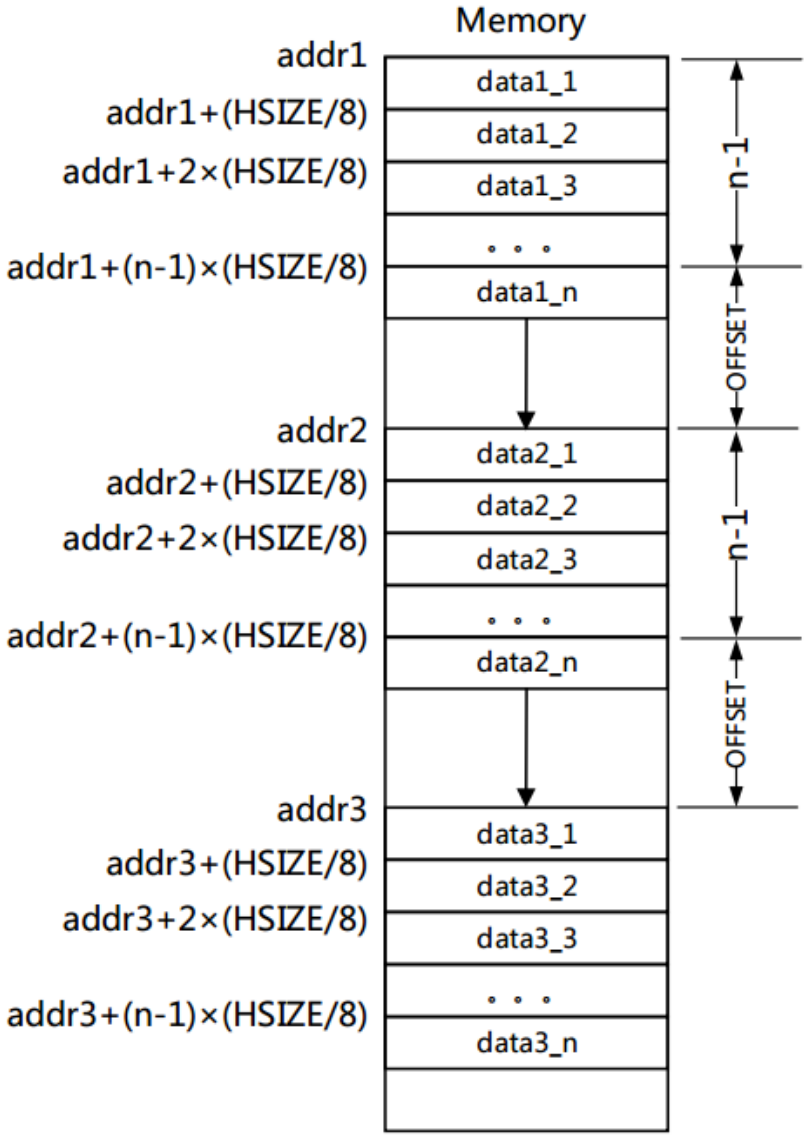


注意：

- 在使能 DMA 控制器之前需确保打开 CLK_FCG 寄存器 DMA 位。
- 在设置触发源之前需确保打开 CLK_FCG 寄存器 AOS 位。
- 上述配置流程是在 DMA 上电初始化的基础上。若已经配置重复、不连续、连锁传输功能，需关闭相应使能位。

2) 不连续传输

该传输可以实现源地址和目标地址在传输完一定量的数据后按照一定的偏移量跳转。跳转的方向由 DMA_CHxCTL.SINC 和 DMA_CHxCTL.DINC 的设定决定。传输过程按照下图方式进行。



其中：n = DMA_CHxCTL.RPTNSCNT

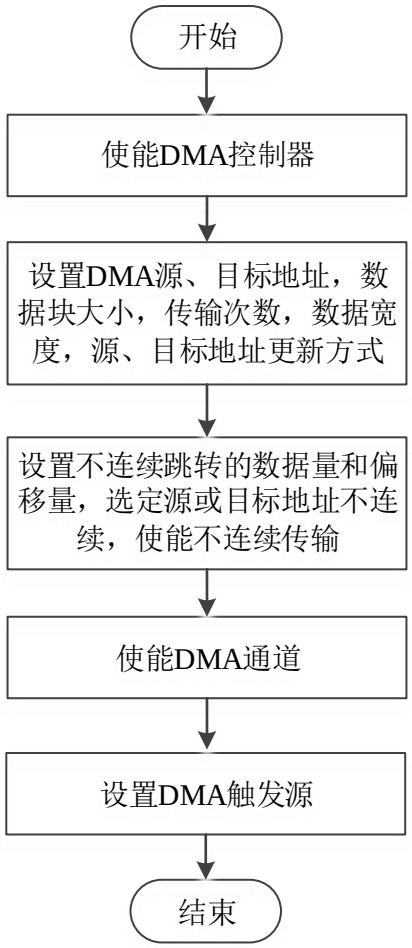
Offset = DMA_CHxCTL.OFFSET

HSIZE 为配置的数据宽度，可以是 8bit，16bit，32bit。

不连续传输的具体样例可以参照 DDL 的 DMA 模块样例 dmac_non_sequence。

基本的配置流程是在基本传输的配置流程基础上添加相关不连续传输的配置，包括传输数据量 n，地址跳转偏移，不连续传输使能，以及选定源地址不连续还是目的地址不连续。可以直接调用 DMA_NonSeqInit 函数实现不连续传输的配置。

基本的配置流程如下图：



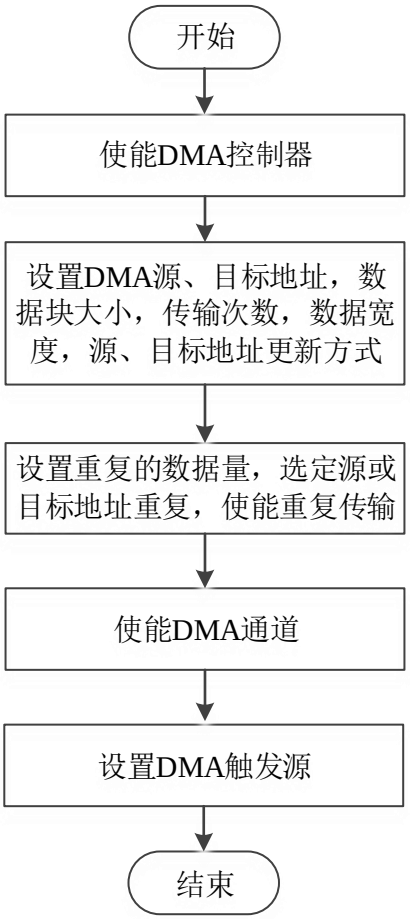
3) 重复传输

该传输可以指定数量的数据后，源地址或目标地址将回到初始配置的地址。当一段指定 ram 或者连续几个寄存器与存储器之间的数据交换，可选择重复传输模式。

重复传输的具体样例可以参照 DDL 的 DMA 模块样例 dmac_repeat。

基本的配置流程是在基本传输的配置流程基础上添加相关重复传输的配置，包括重复数据量 n，重复传输使能，以及选定源地址重复还是目的地址重复。可以直接调用 DMA_RepeatInit 函数实现不连续传输的配置。

基本的配置流程如下图：

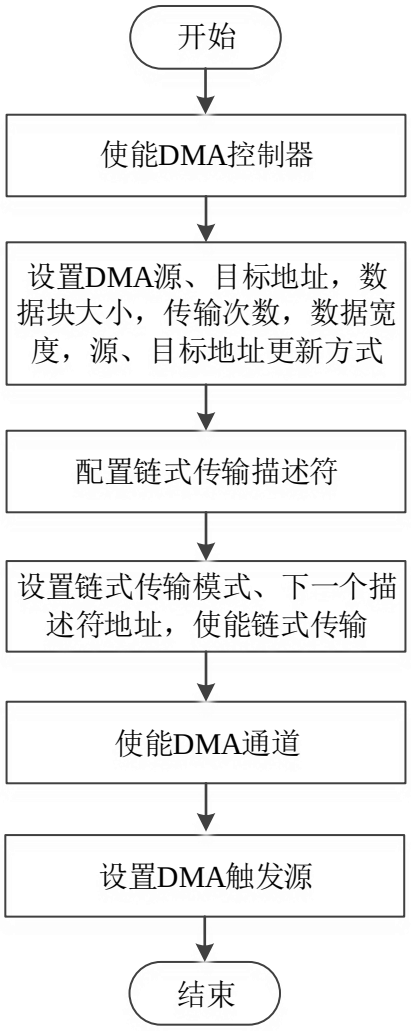


4) 连锁传输

该传输当一个描述符的最后一次传输结束时，LLP 指定的下一个描述符将被从存储器中载入通道配置寄存器。等待下一次传输请求输入，开始新描述符的第一次传输。或者根据寄存器 DMA_CHxCTLx.LLPRUN 的设置，在载入新描述符后直接开始第一次传输。

连锁传输的具体样例可以参照 DDL 的 DMA 模块样例 dmac_link_list_pointer。

基本的配置流程如下图：



5) 传输提前终止

传输过程中通道使能寄存器 DMA_CHEN.CHENx 保持有效，非连锁传输时，数据控制寄存器 DMA_DTCTLx 设定的传输次数完成后自动置为无效，连锁传输时，最后一次传输的传输次数完成后自动置为无效。传输过程中如果软件写 DMA_CHENCLR.CHENCLRx 为 1，则 DMA 将在完成当次数据读写后终止传输。

4 样例代码

4.1 代码介绍

用户可根据上述的工作流程编写自己的代码来学习验证该模块，也可以直接通过华大半导体的网站下载到设备驱动库（Device Driver Library, DDL）的样例代码并使用其中的 DMA 的样例进行验证。

以下部分简要介绍本 AN 基于 DDL 的 DMA 模块样例 dmac_base 代码所涉及的各项配置。

1) 初始化 LED

```
/* Initialize LED */
LedInit();
```

2) 使能 DMA 外设时钟：

```
CLK_FcgPeriphClockCmd(CLK_FCG_DMA, Enable);
```

3) 初始化 DMA 配置：

```
/* Config DMA */
stcDmaChCfg.u32DataWidth = DMA_DATAWIDTH_32BIT;
stcDmaChCfg.u32BlockSize = DMA_BLKSIZE;
stcDmaChCfg.u32TransferCnt = DMA_TRNCNT;
stcDmaChCfg.u32SrcAddr = (uint32_t)(&u32SrcBuf[0]);
stcDmaChCfg.u32DesAddr = (uint32_t)(&u32DstBuf[0]);
stcDmaChCfg.u32SrcInc = DMA_SRCADDRINC_INC;
stcDmaChCfg.u32DesInc = DMA_DESADDRINC_INC;

DMA_ChannelCfg(DMA_CH, &stcDmaChCfg);
```

4) 使能 DMA 及其通道：

```
/* Enable DMA. */
DMA_Cmd(Enable);

/* Enable DMA channel0. */
DMA_ChannelCmd(DMA_CH ,Enable);
```

5) 设置 DMA 触发源：

```
/* Enable AOS clock */
CLK_FcgPeriphClockCmd(CLK_FCG_AOS, Enable);

/* Set DMA trigger source */
DMA_SetTriggerSrc(DMA_CH, EVT_AOS_STRG);AOS_SW_Trigger();
```

6) 比较 DMA 源、目标缓存数据：

```
u8CmpRet = memcmp(u32DstBuf, u32ExpectDstBufData, sizeof(u32DstBuf));
if(0 == u8CmpRet)
{
    LED_G_ON();    /* Meet the expected */
}
else
{
    LED_R_ON();    /* Don't meet the expected */
}
```

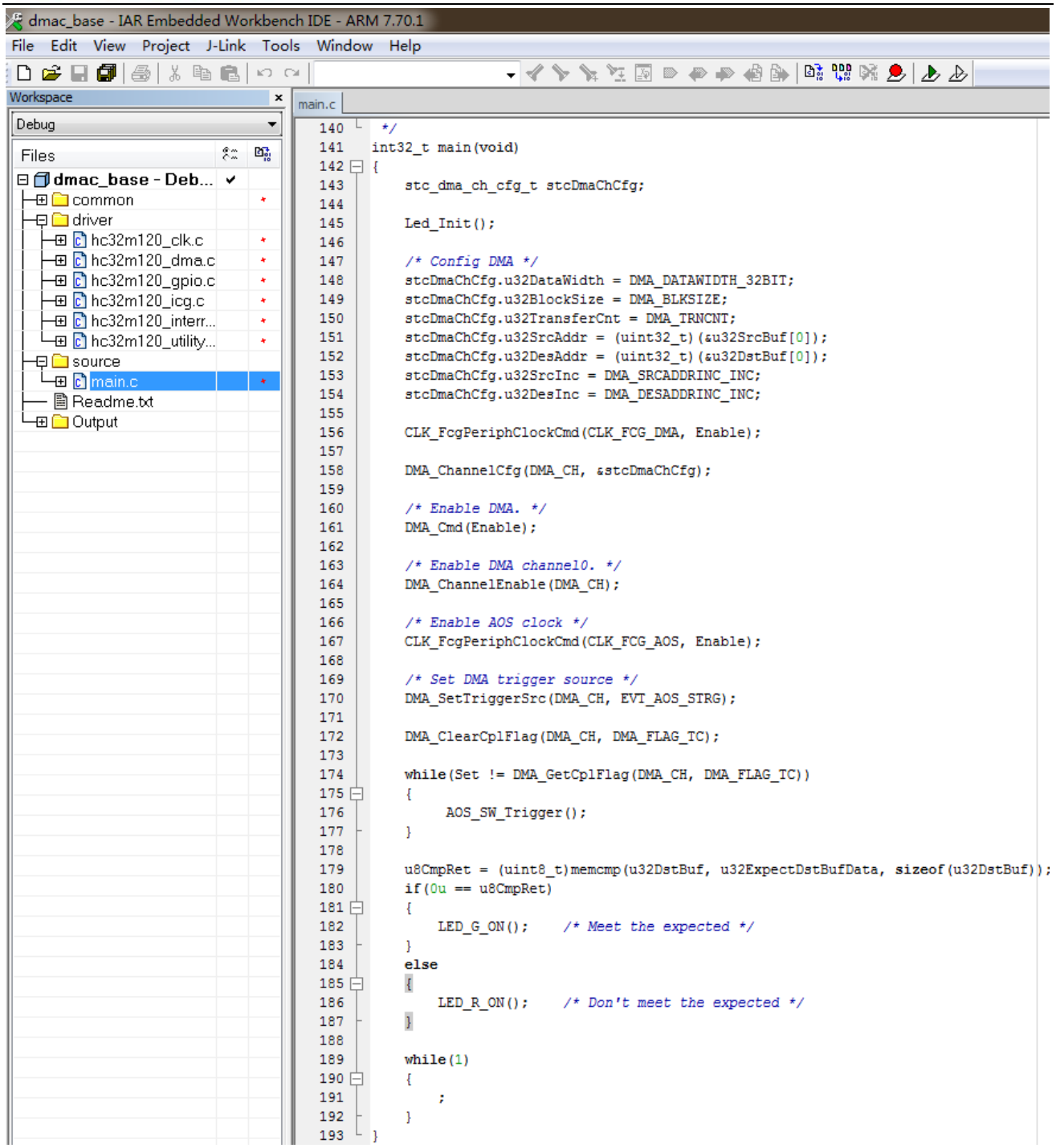
4.2 代码运行

用户可以通过华大半导体的网站下载到 HC32M120 的 DDL 的样例代码（dmac_base），并配合评估用板（EV-HC32M120-LQFP48-050-V1.1）运行相关代码学习使用 DMA 模块。

以下部分主要介绍如何在评估板上运行 DMA 样例代码并观察结果：

- 确认安装正确的 IAR EWARM v7.7 工具（请从 IAR 官方网站下载相应的安装包，并参考用户手册进行安装）。
- 从华大半导体网站下载 HC32M120 DDL 代码。
- 下载并运行 dmac\ dmac_base\中的工程文件：

  1) 打开 dmac_base\工程，并打开‘main.c’如下视图：



2) 点击 重新编译整个项目；

3) 点击 将代码下载到评估板上，全速运行；

4) 绿色 LED 灯点亮。

5 版本信息 & 联系方式

日期	版本	修改记录
2019/10/31	Rev1.0	初版发布

如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email：mcu@hdsc.com.cn

网址：www.hdsc.com.cn

通信地址：上海市浦东新区中科路 1867 号 A 座 10 层

邮编：201203