第 3 章　LCD 显示与触摸

LCD 是 Liquid Crystal Display 的缩写，即液晶显示器，是一种支持全彩显示的显示设备。GD32F3 苹果派开发板上的 LCD 显示模块尺寸为 4.3 寸，能够显示丰富的内容，如彩色文本、图片、波形及 GUI 等。LCD 显示模块上还集成了触摸传感器，这样的模块我们通常称为触摸屏，触摸屏支持多点触控，能够减少对机械按键的使用，提高设备的便携性和交互性。基于 LCD 显示模块可以呈现出更为直观的结果，我们能够设计更加丰富的嵌入式实例。使用 emWin 开发 GUI 需要使用 LCD 显示模块驱动程序，且最终的界面也需要通过 LCD 进行显示。本章将学习 LCD 显示原理和使用方法，以及触摸屏工作原理，实现模拟手写板的功能。

3.1　LCD 显示原理

LCD 按工作原理不同可分为 2 种：被动矩阵式，常见的有 TN-LCD、STN-LCD 和 DSTN-LCD；主动矩阵式，通常为 TFT-LCD。GD32F3 苹果派开发板上使用的 LCD 为 TFT-LCD，在 LCD 的每个像素点上都设置了一个薄膜晶体管（TFT），可有效克服非选通时的串扰，使液晶屏的静态特性与扫描线数无关，极大地提高了图像质量。

LCD 的结构如图 3-1 所示，主要由背光层、垂直偏光片、正极电路、液晶层（包含液晶分子）、负极电路、水平偏光片和彩色滤光片构成。最底层的背光层用于发出白光，通过为液晶层施加电场，影响液晶分子的排列，并结合偏光片对光线的过滤效果，控制输出到彩色滤光片的红色、绿色和蓝色部分的白光光强（发光强度）。不同强度的白光透过彩色滤光片后，形成不同强度的红、绿、蓝三原色光，三原色光混合即可使 LCD 的每个像素点显示不同的颜色，多个像素点组合即可实现在 LCD 上显示文字和图片等效果。

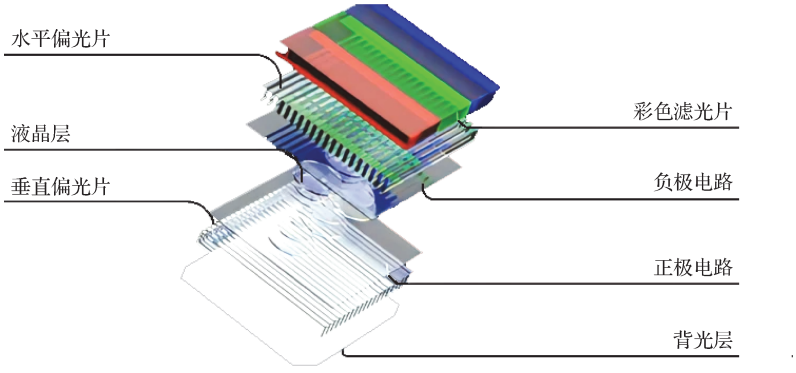

图 3-1　LCD 的结构

为了精确控制像素点的颜色，需要对像素点的红、绿、蓝三种颜色分量进行量化。下面简要介绍本书使用到的 RGB565 和 RGB888 两种颜色格式。

在使用 RGB565 格式时，每个像素点的颜色由 16bit（2 字节）来控制。其中，R 值（红色分量）占最高 5bit，G 值（绿色分量）占中间 6bit，B 值（蓝色分量）占最低 5bit。R 值、G 值、B 值共同组成一个像素点的 RGB 值，如图 3-2 所示。R 值、G 值、B 值越大，表示对应的红色、绿色、蓝色的光强越强。在程序设计过程中，可以通过设置每个像素点的 RGB 值来使像素点呈现对应的颜色。

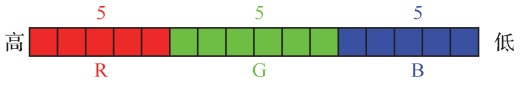

图 3-2　RGB565 颜色格式

而在使用 RGB888 格式时，每个像素点的颜色由 24bit（3 字节）来控制，其中 R 值、G 值、B 值各占 8bit，如图 3-3 所示。

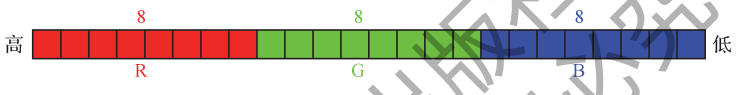

图 3-3　RGB888 颜色格式

在嵌入式开发中使用的主流 LCD 按接口类型不同分为 2 种：RGB 屏（使用 RGB 接口）和 MCU 屏（使用 MCU 接口）。两种屏的主要区别是显存的位置不同。RGB 屏的显存由系统内存（通常为外部 SRAM、SDRAM 等存储器）充当，通过提升系统内存容量，RGB 屏的显存大小也可以灵活提升，因此可以实现大尺寸 RGB 屏。而对于 MCU 屏，在设计之初，由于微控制器的内存较小，将显存内置于 LCM（指液晶显示模块，通常包括 LCD 的显示驱动电路、背光板、接口电路等）中，然后通过特定的命令更新显存，因此 MCU 屏的尺寸往往不会很大。

RGB 屏和 MCU 屏的数据传输模式也不同。对于 RGB 屏，用户需要先将待显示的数据写入系统内存为 LCD 显存分配的空间，启动显示后，微控制器的 DMA（直接存储器访问）机制会自动将显存数据通过 RGB 接口发送到 LCM，RGB 屏即可正常显示。而对于 MCU 屏，需要微控制器发送命令来修改 MCU LCM 内部的 RAM，因此 MCU 屏的显示速度会比 RGB 屏慢。MCU 接口的 LCM 内部通常自带 LCD 控制器，可对微控制器发送过来的数据和命令进行处理，从而得到每个像素点的 RGB 值，最终在显示屏上显示。而对于 RGB 接口的 LCM，微控制器发送过来的数据为每个像素点的 RGB 值，无须转换即可显示。RGB 屏和 MCU 屏的框架分别如图 3-4 和图 3-5 所示。

GD32F3 苹果派开发板上使用的 LCD 显示模块是一款集 NT35510 驱动芯片、4.3 寸（480 像素×800 像素分辨率）触摸屏及驱动电路于一体的 MCU 屏，可以通过 GD32F303ZET6 芯片上的外部存储器控制器 EXMC 来控制 LCD 显示模块。

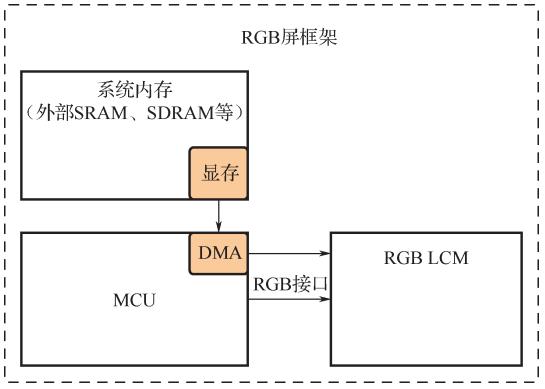

图 3-4　RGB 屏框架

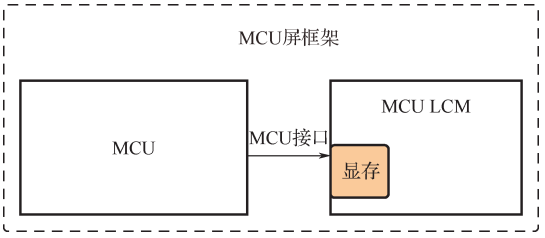

图 3-5　MCU 屏框架

3.2　LCD 显示模块接口

开发板上的 LCD 显示模块接口电路原理图如图 3-6 所示。LCD 显示模块的 EXMC_D[0:15]引脚分别与 GD32F303ZET6 芯片的 PD[14:15]、PD[0:1]、PE[7:15]和 PD[8:10]引脚相连，LCD_CS 引脚连接到 PG9 引脚，LCD_RD 引脚连接到 PD4 引脚，LCD_WR 引脚连接到 PD5 引脚，LCD_RS 引脚连接到 PF0 引脚，LCD_BL 引脚连接到 PB0 引脚，LCD_IO4 引脚连接到 NRST 引脚。

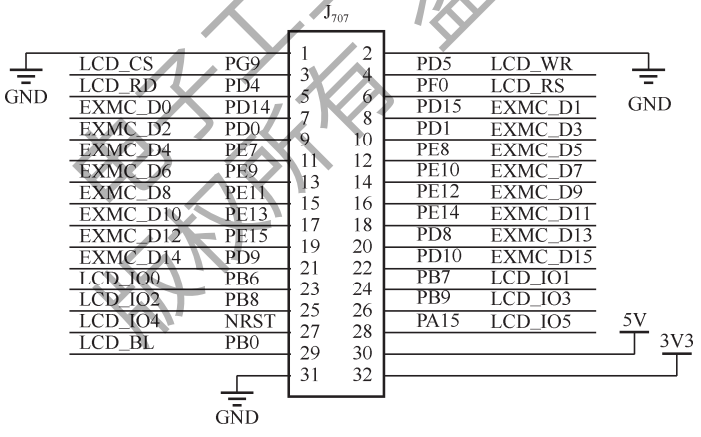

图 3-6　LCD 显示模块接口电路原理图

MCU 接口分为多种类型，常用的有 8080 并行接口、6800 并行接口、3 线 SPI 接口、4 线 SPI 接口等。与 GD32F3 苹果派开发板配套的 LCD 显示模块采用 16 位的 8080 并行接口来传输数据，用到了 4 条控制线（LCD_CS、LCD_WR、LCD_RD 和 LCD_RS）和 16 条双向数据线。LCD 显示模块接口定义如表 3-1 所示。

表 3-1　LCD 显示模块接口定义

序　号	名　称	说　明	引　脚
1	LCD_CS	片选信号，低电平有效	PG9

续表

序　号	名　称	说　明	引　脚
2	LCD_WR	写入信号，上升沿有效	PD5
3	LCD_RD	读取信号，上升沿有效	PD4
4	LCD_RS	指令/数据标志（0-读/写指令，1-读/写数据）	PF0
5	LCD_BL	背光控制信号，高电平有效	PB0
6	LCD_IO4	硬件复位信号，低电平有效	NRST
7	EXMC_D[0:15]	16 位双向数据线	PE[7:15]、PD[0:1]、PD[8:10]、PD[14:15]

LCD 显示模块通过 8080 并行接口传输的数据有两种，分别为 NT35510 芯片的控制指令和 LCD 像素点显示的 RGB 颜色数据。这两种数据都涉及读取和写入，下面根据 LCD 的信号线来简单介绍 LCD 读取、写入控制指令和 RGB 颜色数据的时序图。

读取、写入数据首先需要拉低片选信号 LCD_CS，然后根据是读取还是写入数据，配置 LCD_RD 和 LCD_WR 的电平。如果是写入数据，则将 LCD_WR 的电平拉低，LCD_RD 的电平拉高；读数据则相反。数据通过 EXMC_D 的 16 位双向数据线进行传输。

（1）写入数据：在 LCD_WR 的上升沿，将数据写入 NT35510 芯片，如图 3-7 所示。

（2）读取数据：在 LCD_RD 的上升沿，读取数据线上的数据（EXMC_D[0:15]），如图 3-8 所示。

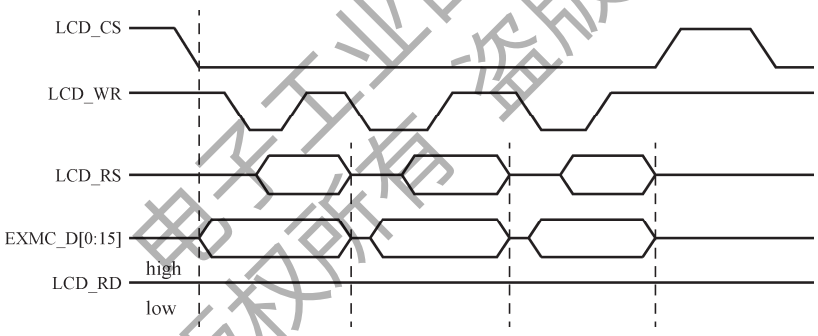

图 3-7　写入数据时序图

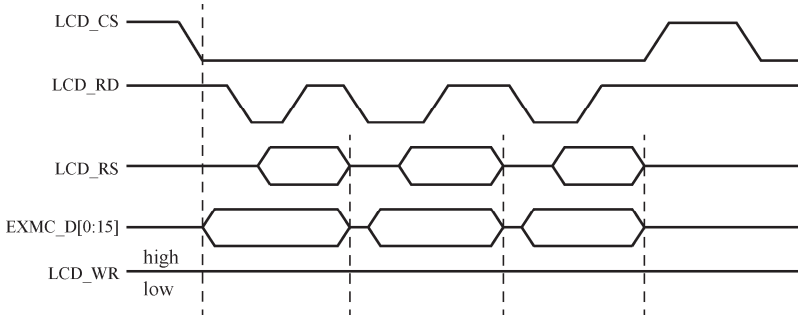

图 3-8　读取数据时序图

3.3 LCD 控制原理

3.3.1 NT35510 芯片的显存及常用指令

NT35510 驱动芯片为液晶控制器，自带显存，显存大小为 1152000 字节（480×800×24/8 字节），即 24 位模式下的显存容量。在 16 位模式下，NT35510 芯片采用 RGB565 格式存储颜色数据，此时 NT35510 芯片的 24 位数据线、GD32F30x 系列 MCU 的 16 位数据线和 LCD GRAM 的对应关系如表 3-2 所示。

表 3-2 数据线与 LCD GRAM 的对应关系

名　称	对 应 关 系				
NT35510（24 位）	D17～D13	D12	D11～D6	D5～D1	D0
MCU（16 位）	D15～D11	NC	D10～D5	D4～D0	NC
LCD GRAM（16 位）	R[4]～R[0]	NC	G[5]～G[0]	B[4]～B[0]	NC

NT35510 芯片在 16 位模式下，使用的数据线为 D17～D13 和 D11～D1，D0 和 D12 未使用，D17～D13 和 D11～D1 分别对应 GD32F303ZET6 MCU 上的 16 个 GPIO。16 位的 RGB 颜色数据，低 5 位为蓝色分量，中间 6 位为绿色分量，高 5 位为红色分量。数值越大，表示颜色越深。注意，NT35510 芯片的所有指令均为 16 位，且读/写 GRAM 时也是 16 位。

微控制器通过向 NT35510 芯片发送指令来设置 LCD 的显示参数，NT35510 芯片提供了一系列指令供用户开发，关于这些指令具体的定义和描述请参考文件 *NT35510 Data Sheet*（《NT35510 数据表》，位于本书配套资料包“09.参考资料”文件夹下）的第 255～257 页。设置 NT35510 芯片的过程如下：先向 NT35510 芯片发送某项设置对应的指令，目的是告知 NT35510 芯片接下来将进行该项设置，然后发送此项设置的参数完成设置。例如，设置 LCD 的扫描方向为从左到右、从下到上，应先向 NT35510 芯片发送设置读/写方向的指令（0x3600），然后发送从左到右、从下到上的读写方向参数（0x0080），即可完成设置。

下面简要介绍 NT35510 芯片的几条常用指令：0xDA00～0xDC00、0x3600、0x2A00～0x2A03、0x2B00～0x2B03、0x2C00 和 0x2E00。

1. 0xDA00～0xDC00

指令 0xDA00～0xDC00 为读 ID 指令，分别用于读取 LCD 产品的 ID、控制器版本的 ID 及控制器的 ID，每个指令输出一个参数，每个 ID 以 8 位数据（即指令后的参数）的形式输出（高 8 位固定为 0）。将 3 条指令的输出进行组合即可得到芯片 ID。例如，读出的 ID 为 8000H，表示芯片型号为 NT35510。

上述 3 条读 ID 指令的具体描述如表 3-3 所示，下面以 0xDA00 为例进行介绍。要完成读 ID 指令的操作，就要先向 NT35510 芯片写入指令 0xDA00。写指令操作需要将 LCD_RS 的电平拉低，将 LCD_RD 的电平拉高，然后在 LCD_WR 的上升沿通过 EXMC_D[0:15]写入 0xDA00。NT35510 芯片接收并识别到指令后，会发送 ID 参数，接下来需要将 LCD_RS 和 LCD_WR 的电平拉高，然后在 LCD_RD 的上升沿通过 EXMC_D[0:15]读取 ID 参数。通过以上操作即可完成读 ID 指令的操作。

表 3-3　读 ID 指令

顺序	控　制			各 位 描 述									HEX
	RS	RD	WR	D15	D14	D13	D12	D11	D10	D9	D8	D7～D0	
指令	0	1	↑	1	1	0	1	1	0	1	0	00H	DA00
参数	1	↑	1	0	0	0	0	0	0	0	0	00H	00
指令	0	1	↑	1	1	0	1	1	0	1	1	00H	DB00
参数	1	↑	1	0	0	0	0	0	0	0	0	80H	80
指令	0	1	↑	1	1	0	1	1	1	0	0	00H	DC00
参数	1	↑	1	0	0	0	0	0	0	0	0	00H	00

2．0x3600

指令 0x3600 为存储访问控制指令，用于控制 NT35510 芯片存储器的读/写方向。在连续写 GRAM 时，可以通过该指令控制 GRAM 指针的增长方向，从而控制显示方式，读 GRAM 类似。关于该指令的具体描述如表 3-4 所示。

表 3-4　存储访问控制指令

顺序	控　制			各 位 描 述								HEX
	RS	RD	WR	D15～D8	D7	D6	D5	D4	D3	D2	D1	D0	
指令	0	1	↑	36H	0	0	0	0	0	0	0	0	3600
参数	1	1	↑	00H	MY	MX	MV	ML	RGB	MH	RSMX	RSMY	

其中，ML 用于控制 TFT-LCD 的垂直刷新方向；RGB 用于控制 R、G、B 的排列顺序：0（RGB）或 1（BGR）；MH 用于控制水平刷新方向；RSMX 用于左右翻转图像（该位为 1 时有效）；RSMY 用于上下翻转图像（该位为 1 时有效）。

另外，通过设置 MY、MX 和 MV 这 3 位参数，可以控制 LCD 的扫描方向，如表 3-5 所示。例如，在显示 BMP 格式的图片时，BMP 解码是从图片的左下角开始到右上角结束的，如果设置 LCD 的扫描方向为从左到右，从下到上，则只需要设置一次原点坐标，然后不断向 NT35510 芯片发送颜色数据就可以了，这样可以大大提高显示速率。

表 3-5　MY、MX 和 MV 参数的取值及其效果

控　制　位			效　果
MY	MX	MV	LCD 扫描方向（GRAM 自增模式）
0	0	0	从左到右，从上到下
1	0	0	从左到右，从下到上
0	1	0	从右到左，从上到下
1	1	0	从右到左，从下到上
0	0	1	从上到下，从左到右
0	1	1	从上到下，从右到左
1	0	1	从下到上，从左到右
1	1	1	从下到上，从右到左

3．0x2A00～0x2A03

指令 0x2A00～0x2A03 为列地址设置指令。在默认的扫描方式（从左到右，从上到下，即竖屏显示）下，这 4 条指令用于设置横坐标（*X* 轴坐标）的范围。因为 GD32F3 苹果派开发板上使用的 LCD 分辨率为 480 像素×800 像素，所以 NT35510 芯片给出了 *X* 轴和 *Y* 轴坐标的范围限制：0≤*x*≤479，0≤*y*≤799，此范围适用于竖屏情况。若为横屏显示，则 *X* 轴和 *Y* 轴坐标范围互换。

指令 0x2A00～0x2A03 各带有一个参数，用于设置两个坐标值，即列地址的起始值 XS 和结束值 XE（XS 和 XE 都为 16 位，且都由两个参数的低 8 位组合而成），这两个坐标值的范围需满足 0≤XS≤XE≤479（竖屏）。一般在设置 *X* 轴坐标范围时，只需要设置 XS，因为 XE 在初始化的时候已被设置了一个固定值。关于列地址设置指令的具体描述如表 3-6 所示。

表 3-6　列地址设置指令

顺序	控　制			各 位 描 述									HEX
	RS	RD	WR	D15～D8	D7	D6	D5	D4	D3	D2	D1	D0	
指令 1	0	1	↑	2AH	0	0	0	0	0	0	0	0	2A00
参数 1	1	1	↑	00H	XS15	XS14	XS13	XS12	XS11	XS10	XS9	XS8	XS[15:8]
指令 2	0	1	↑	2AH	0	0	0	0	0	0	0	1	2A01
参数 2	1	1	↑	00H	XS7	XS6	XS5	XS4	XS3	XS2	XS1	XS0	XS[7:0]
指令 3	0	1	↑	2AH	0	0	0	0	0	0	1	0	2A02
参数 3	1	1	↑	00H	XE15	XE14	XE13	XE12	XE11	XE10	XE9	XE8	XE[15:8]
指令 4	0	1	↑	2AH	0	0	0	0	0	0	1	1	2A03
参数 4	1	1	↑	00H	XE7	XE6	XE5	XE4	XE3	XE2	XE1	XE0	XE[7:0]

4．0x2B00～0x2B03

与列地址设置指令类似，指令 0x2B00～0x2B03 为行地址设置指令。在默认扫描方式下，这 4 条指令用于设置纵坐标（*Y* 轴坐标）范围，也各带有一个参数，用于设置行地址的起始值 YS 和结束值 YE（YS 和 YE 都为 16 位，且都由两个参数的低 8 位组合而成），这两个坐标值的范围需满足 0≤YS≤YE≤799（竖屏）。一般在设置 *Y* 轴坐标范围时，只需要设置 YS，因为 YE 在初始化的时候已被设置了一个固定值。

5．0x2C00

指令 0x2C00 为写 GRAM 指令。在向 NT35510 芯片发送该指令之后，即可向 LCD 的 GRAM 中写入颜色数据，该指令支持连续写，具体描述如表 3-7 所示。

在收到指令 0x2C00 后，数据有效位宽变为 16 位，可以连续写入 LCD GRAM 值（16 位的 RGB565 值），GRAM 的地址将根据 MY、MX 和 MV 设置的扫描方向进行自增。例如，如果设置的扫描方向为从左到右，从上到下，那么设置好起始坐标(XS,YS)后，每写入一个颜色值，GRAM 地址将会自增 1（XS++）。如果写到 XE，则重新回到 XS，此时 YS++，即先显示完一行，然后列数加 1，再显示下一行，一直写到结束坐标(XE,YE)，其间无须再次设置其他坐标，从而提高写入速度。

表 3-7　写 GRAM 指令

顺序	控　制			各 位 描 述									HEX
	RS	RD	WR	D15	D14	D13	D12	D11	D10	D9	D8	D7～D0	
指令	0	1	↑	0	0	1	0	1	1	0	0	0	2C00
参数 1	1	1	↑	D1[15:0]									
……	1	1	↑	D2[15:0]									
参数 *N*	1	1	↑	D3[15:0]									

6．0x2E00

指令 0x2E00 为读 GRAM 指令，如表 3-8 所示。该指令用于读取 GRAM。NT35510 芯片在收到该指令后，第一次输出的为 dummy 数据，即无效数据，从第二次开始，输出的才是有效的 GRAM 数据（从起始坐标(XS,YS)开始），输出方式为，每个颜色分量占 8 位，一次输出两个颜色分量。例如，第一次输出 R1G1，随后的规律为，B1R2→G2B2→R3G3→B3R4→G4B4→R5G5，以此类推。如果只需要读取一个点的颜色值，那么只需要接收至参数 3，后面的参数不需要接收；若要连续读取，则按照上述规律接收颜色数据。

表 3-8　读 GRAM 指令

顺序	控　制			各 位 描 述												HEX
	RS	RD	WR	D15～D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	
指令	0	1	↑	2EH				0	0	0	0	0	0	0	0	2E
参数 1	1	↑	1	××												dummy
参数 2	1	↑	1	R1[4:0]	××			G1[5:0]						××		R1G1
参数 3	1	↑	1	B1[4:0]	××			R2[4:0]					××			B1R2
参数 4	1	↑	1	G2[5:0]		××		B2[4:0]					××			G2B2
参数 5	1	↑	1	R3[4:0]	××			G3[5:0]						××		R3G3
参数 *N*	1	↑	1	按以上规律输出												

以上就是 NT35510 芯片常用的一些指令，通过这些指令可以控制 LCD 进行简单的显示。

3.3.2　EXMC 简介

LCD 显示可以通过 GPIO 引脚模拟 8080 接口时序来控制，但由于使用 GPIO 引脚模拟时序较慢，且 CPU 的占用率较高，因此通常使用微控制器的 EXMC 接口来驱动 LCD 显示模块。下面介绍 EXMC 的基本原理。

1．EXMC 功能框图

EXMC 是外部存储器控制器，主要用于访问各种外部存储器。通过配置寄存器，EXMC 可以把 AMBA 协议转换为专用的片外存储器通信协议。GD32F30x 系列微控制器的 EXMC 可访问的存储器包括 SRAM、ROM、NOR Flash、NAND Flash 和 PC Card 等。用户还可以通过调整、配置寄存器中的时间参数来提高通信效率。EXMC 的访问空间被划分为多个块（Bank），每个块支持特定的存储器类型，用户可以通过配置 Bank 的控制寄存器来控制外部存储器。