数字缓冲器基本介绍 DigBuf V 1.3

Copyright © 2012 Cypress Semiconductor Corporation. All Rights Reserved.

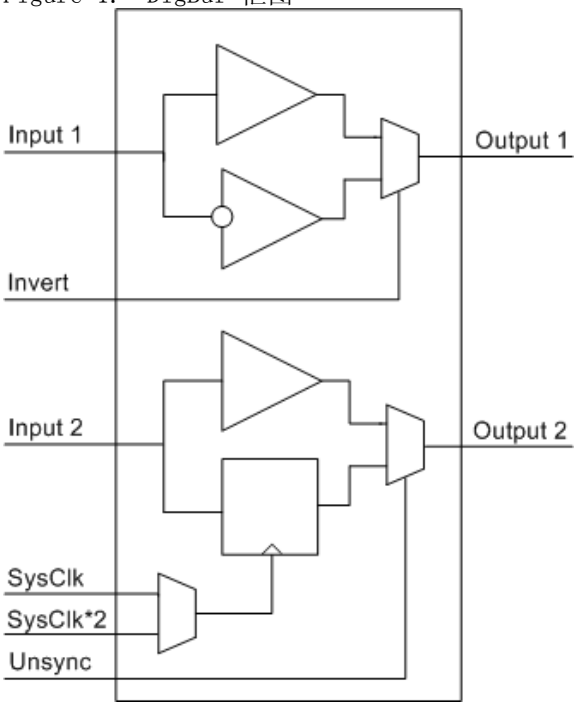
资源	PSoC® 模块			API 存储器 （字节）		引脚 （每个外部 I/O）
	数字	模拟 CT	模拟 SC	Flash （闪存）	RAM	
CY8C29/27/24/22/21xxx, CY8C23x33, CY8CLED02/04/08/16, CY8CLED0xD, CY8CLED0xG, CY8CTST110, CY8CTMG110, CY8CTST120, CY8CTMG120, CY8CTMA120, CY8CTMA140, CY8C21x45, CY8C22x45, CY8CTMA30xx, CY8C28x45, CY8CPLC20, CY8CLED16P01, CY8C28xxx						
	0	0	0	25	0	1 到 4
CYWUSB6953	0	0	0	25	0	1 到 4

功能和概述

- 两个数字缓冲器
- 可以将 Input1 反相
- 可用于产生 Output1 上升沿上的中断。

DigBuffer 用户模块是一个简单的双输入双输出数字缓冲器。 输出等于输入信号。

Figure 1. DigBuf 框图



功能描述

DigBuf 是两个数字缓冲器的组合。 可将其映射至任何数字 PSoC 模块。 API 提供启用或禁用 Output1 中断功能的函数。

直流和交流电气特性

Table 1. DigBuf 直流和交流电气特性

参数	条件和注释	典型值	限制	单位
来自全局总线的输入 F_{max}			12	MHz
来自内部连接的输入 F_{max}			48	MHz
输入到输出转换		< 25		ns
输出到全局总线的 F_{max}			12	MHz
输出到内部连接的 F_{max}			48	MHz

时序

当输入或输出与全局总线连接时，DigBuf 用户模块转换速率被限制在 12 MHz 以内。 为达到更高的转换速率，将 DigBuf 用户模块紧挨提供高速输出的用户模块，则该模块即可选作 Input2 信号的源。

放置

可以将 DigBuf 放置在任何数字 PSoC 模块中。

参数和资源

在载入用户模块时，确定该模块默认处于启用或禁用的状态。 如果该参数被设置为启用 (Enable)，则无需调用启动 (Start) 函数来启用该用户模。 如果设置为禁用 (Disable)，则不打开缓冲器功能，直至调用了启动 (Start) 函数。

Input1

Input1/Output1 数字缓冲器是非同步缓冲器。

Input1 源的数量根据放置用户模块的数字模块的不同而变化。 这些源包括模拟比较器 (Analog Comparator) 总线、行输入和输出总线。

Input2

Input2 源的数量根据放置用户模块的数字模块的不同而变化。 这些源包括 48 MHz 振荡器输出、从 24-MHz 系统时钟细分出来的较低频率 (VC1、VC2 及 VC3)、其他 PSoC 模块，以及经由全局输入和输出的外部输入。

Input2 ClockSync

如果该参数被设置为 “Unsynchronized”，则 Input2/Output2 数字缓冲器作为非同步缓冲器运行。 如果该参数被设置为 “Sync to SysClk” 或 “Sync to SysClk*2”，则 Input2/Output2 电路将作为一个触发器进行工作，该触发器采用针对同步而选择的相应时钟作为时钟。

如果要将 DigBuf 用户模块的 Output2 信号用作输入信号或 PSoC 内其他模块的时钟，我们建议使 Input2 与内部系统时钟之一进行同步。 时钟的选择取决于输出的内部路径，在下表中进行详述。

ClockSync 值	使用说明
Sync to SysClk（同步到系统时钟）	在输出经由使用 24 MHz (SysClk) 、或二分或多分衍生 SysClk 时钟源的模块时，使用该设置。 例如 VC1、VC2、VC3 （当 VC3 由 SysClk 驱动时），32 KHz。
Sync to SysClk*2	在输出经由使用 48 MHz (SysClk*2) 、或使用基于 SysClk*2 的时钟的模块时，使用该设置。
Unsynchronized	在需要未同步输入时使用。 通常，只有在计划将输出直接馈送到引脚、或为了产生中断时，才建议使用。

Output1

可以使输出经由四个全局输出信号之一。

Output2

可以使输出经由四个全局输出信号之一。

InvertInput1

反转 Input1 的值。

中断生成控制

当选中 PSoC Designer 中的 “ 启用中断生成控制 ” 复选框时，有两个附加参数变为可用。 **启用中断生成控制** 复选框时，会有一个附加参数变为可用。 可以在以下菜单下找到此复选框： **项目 > 设置 > 芯片编辑器**. 当外覆层的多个用户模块所共享的中断用于多个外覆层时，中断生成控制非常重要：

- 中断 API
- IntDispatchMode

InterruptAPI

InterruptAPI 参数允许有条件生成用户模块的中断处理程序和中断矢量表条目。 选择 “ 启用 ” 以生成中断处理程序和中断矢量表条目。 选择 “ 禁用 ” 可不生成中断处理程序和中断矢量表条目。 在那些拥有多个外覆层而且有多个外覆层使用同一模块资源的项目中，特别推荐要正确地选择是否要生成中断 API。 仅在必要时选择生成中断 API，这样可以避免生成中断调度代码，从而减少开销。

IntDispatchMode

IntDispatchMode 参数用于指定中断请求的处理方式，这些中断由同一模块不同外覆层中的多个用户模块共享。 选择 “ActiveStatus” 会导致固件在为共享的中断请求提供服务之前测试哪一个外覆层正处于活动状态。 每次请求共享中断时，都会进行此测试。 这会增加延迟，还会产生为共享中断请求提供服务的不确定过程，但是不需要任何 RAM。 选择 “OffsetPreCalc” 参数会导致固件只在最初载入一个重叠层时计算共享中断请求的来源。 这种计算可减少中断延迟，并产生为共享中断请求提供服务的确定过程，但会占用一个字节的 RAM 空间。

应用程序编程接口

提供的应用程序编程接口 (API) 子程序作为用户模块的一部分，允许设计人员能够采用更高级的方式处理模块。 本节指定每个函数的接口以及 “include” 文件提供的相关常量。

Note

在这里，如同所有用户模块 API 中的一样，A 和 X 寄存器的值可能通过调用 API 函数发生更改。 如果在调用后需要 A 和 X 的值，则调用函数负责在调用前保留 A 和 X 的值。 选择此 “ 寄存器易失 ” 策略旨在提高效率，自 PSoC Designer 1.0 版起已强制使用此策略。 C 编译器自动遵循此要求。 汇编语言程序

员也必须确保其代码遵守这一策略。 虽然一些用户模块 API 函数可以保留 A 和 X 不变，但是无法保证它们将来也会如此。

对于大储存器模式器件， 保存 CUR_PP、IDX_PP、MVR_PP 以及 MVW_PP 寄存器中的所有值也是调用程序的职责。 尽管部分寄存器现在可能不可修改，但是无法保证在将来的版本中也会如此。

以下是为 DigBuf 提供的 API 编程子程序。

DigBuf_Start

说明：

启动模块内的数字缓冲器。

C 原型：

```
void DigBuf_Start(void);
```

汇编：

```
lcall DigBuf_Start
```

参数：

None

返回值：

None

副作用：

可通过该函数修改寄存器 A 和 X。

DigBuf_Stop

说明：

停止模块内的数字缓冲器。 输出被驱动为低电平。

C 原型：

```
void DigBuf_Stop(void);
```

汇编：

```
lcall DigBuf_Stop
```

参数：

None

返回值：

None

副作用：

可通过该函数修改寄存器 A 和 X。

DigBuf_EnableInt

说明：

启用中断模式运行。

C 原型：

```
void DigBuf_EnableInt(void);
```

汇编：

```
lcall DigBuf_EnableInt
```

参数：

None

返回值：

None

副作用：

可通过该函数修改寄存器 A 和 X。

DigBuf_DisableInt

说明：

禁用中断模式运行。

C 原型：

```
void DigBuf_DisableInt(void);
```

汇编：

```
lcall DigBuf_DisableInt
```

参数：

None

返回值：

None

副作用：

用户可通过该函数修改寄存器 A 和 X。

固件源代码示例

以下汇编语言源代码说明了 API 的使用方法。

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;  Start the Digital Buffers user module.
;
;  Parameters: None
;  Returns:    None
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

include    "M8C.inc"               ; include the global include file

include    "DigBuf.inc"            ; include the DigBuf API include file

StartBuffers:
    M8C_EnableGInt
    call  DigBuf_EnableInt
    call  DigBuf_Start
```

```
    ret
```

同一编码用 C 语言表示如下。

```
/*****************************************************************************
*  Start the Digital Buffers user module.
*
*  Parameters: None
*  Returns:    None
*****************************************************************************/

#include    "M8C.h"
#include    "DigBuf.h"

void StartBuffers(void)
{
   M8C_EnableGInt;
   DigBuf_EnableInt();
   DigBuf_Start();
}
```

配置寄存器

下面几个表格中描述了用于配置 DigBuf 用户模块的 PSoC 数字模块寄存器。 仅解释参数化符号。

Table 2. 模块 DigBuf 寄存器函数

位	7	6	5	4	3	2	1	0
值	0	0	1	0	0	0	1	0

Table 3. Block DigBuf: 寄存器输入

位	7	6	5	4	3	2	1	0
值	Input1 Select				Input2 Select			

Input1 Select 从各源中设置输入，并在器件编辑器中进行设置。 Input2 Select 从各源中设置输入，并在器件编辑器中进行设置。

Table 4. Block DigBuf: 寄存器输出

位	7	6	5	4	3	2	1	0
值	Input2 ClockSync Select		Output2 Enable（输出 2 启用）	Output2 Select		Output1 Enable（输出 1 启用）	Output1 Select	

Outputx Enable 标志用于表示输出已被启用。 Outputx Select 标志指示 DigBuf 的输出将经由哪里。 这两个参数都是在器件编辑器中设置的。

Table 5. Block DigBuf: LFSR Register DR0

位	7	6	5	4	3	2	1	0
值	0	0	0	0	0	0	0	0

Table 6. Block DigBuf: Polynomial Register DR1

位	7	6	5	4	3	2	1	0
值	0	0	0	0	0	0	0	0

Table 7. Block DigBuf: Seed/Residual Register DR2

位	7	6	5	4	3	2	1	0
值	0	0	0	0	0	0	0	0

Table 8. Block DigBuf: 控制寄存器 CR0

位	7	6	5	4	3	2	1	0
值	0	0	0	0	0	0	1	启动 (Start)

当启动 (Start) 位被启用时，DigBuf 功能被启用；当它被清除时，模块的输出被驱动至低电平。

版本历史记录

版本	创作者	说明
1.3	DHA	添加了版本历史
1.3.b	DHA	1. 更新了用户模块框图。 2. 更新了对 “Input1” 和 “Input2 ClockSync” 用户模块参数的说明。

Note PSoC Designer 5.1 在所有用户模块基本介绍中都引入了 “ 版本历史 ”。 本数据表详细介绍了当前和先前用户模块版本之间的区别。

Copyright © 2012 Cypress Semiconductor Corporation. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC Designer™ and Programmable System-on-Chip™ are trademarks and PSoC® is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.