

TI 安全微控制器

FlexRay 模块培训

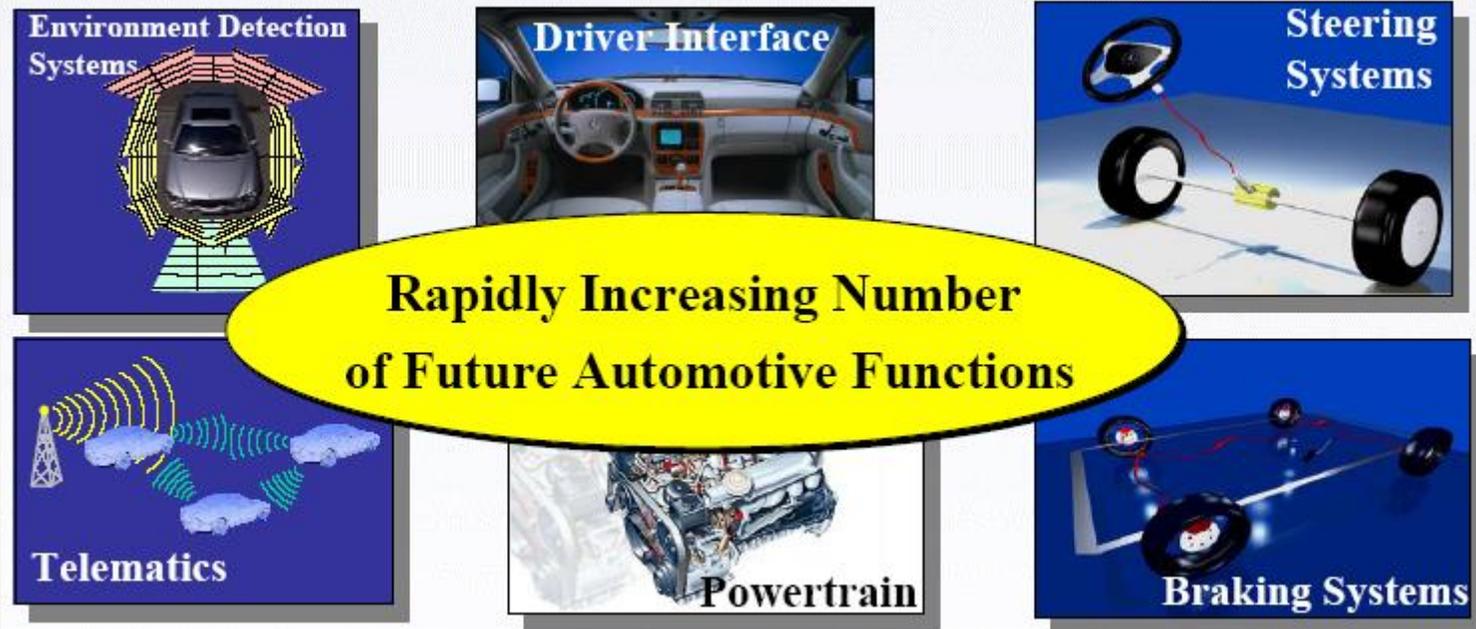


FlexRay 摘要

FlexRay 概述	3
TI 的 FlexRay 实施	27
FlexRay 传输单元 (FTU)	45
FlexRay 中断结构	59

FlexRay 概述

未来的网络需求



未来应用对总线系统有额外的要求

- 可靠性
- 稳健性
- 确定性
- 容错性

FlexRay 主要特性

- 开放总线系统
- 支持冗余传输通道
- 每通道 10 兆位/秒的数据速率
- 支持容错同步全球时基
- 静态和动态数据传输（可扩展）
 - 确定性数据传输
 - 无仲裁的传输
- 在硬件中实施的容错服务和时间触发型服务
- 支持光学和电子物理层

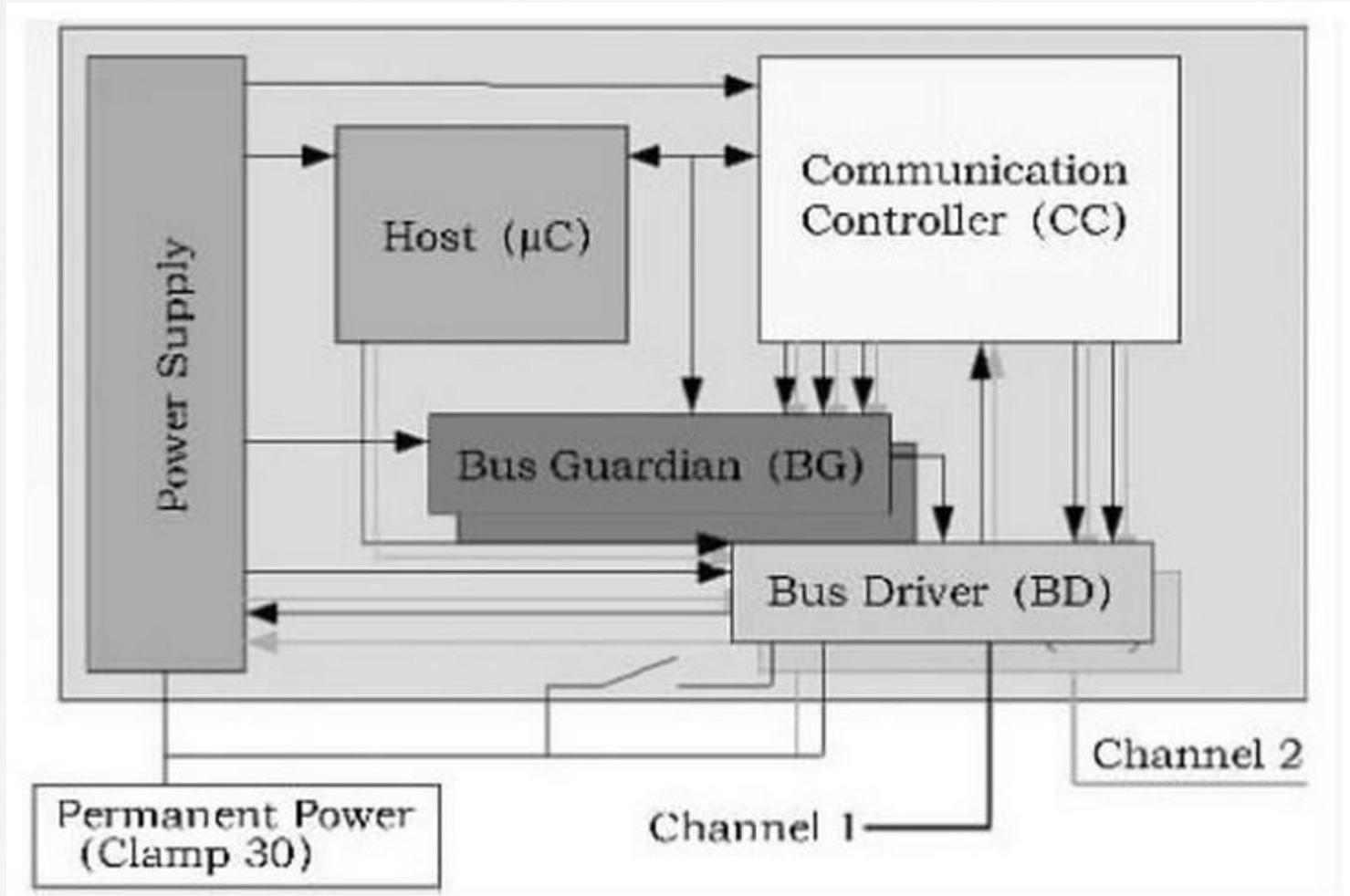
FlexRay 协议硬件



演示：在我们的网络中有 13 个节点。1 是 TMS570LS3137，其他是 TMS570LS20216

FlexRay 架构 -- ECU

- FlexRay 节点：ECU



FlexRay 架构 – 协议运行控制

FlexRay 控制器状态：

- 1 – 默认配置
- 2 – 配置
- 3 – 就绪
- 4 – 唤醒
- 5 – 启动
- 6 – 正常主动
- 7 – 正常被动
- 8 – 停止

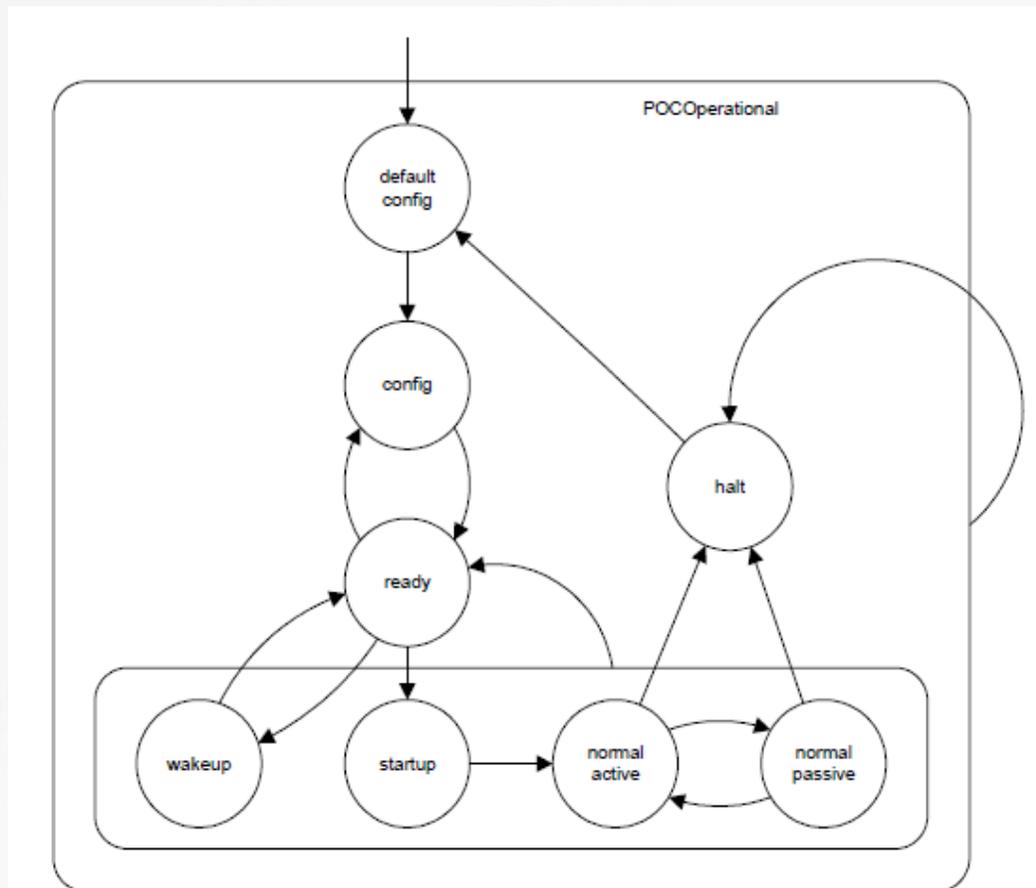


Figure 2-3: Overview of protocol operation control.

FlexRay 架构 – 网络拓扑

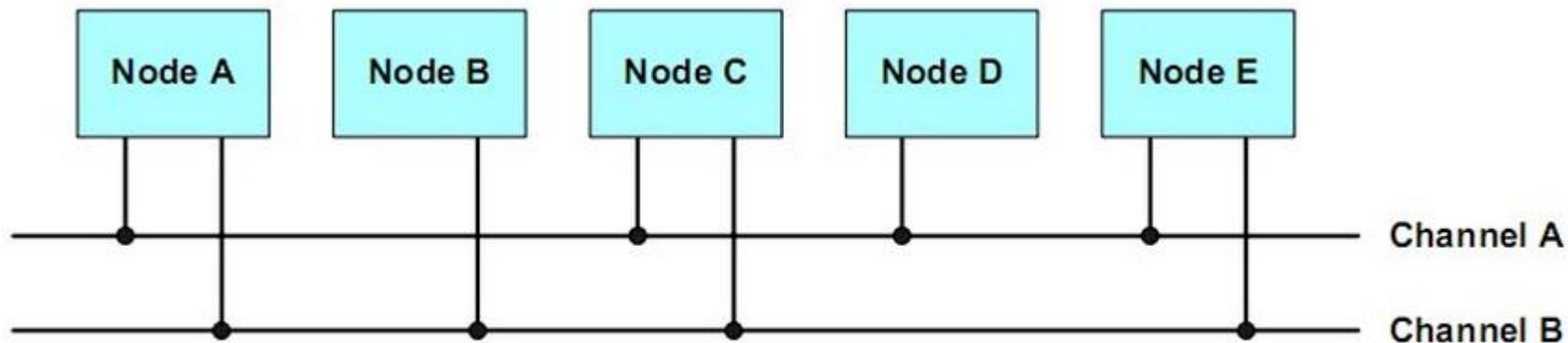


Figure 1-1: Dual channel bus configuration.

- 无环且不具备有源元件的结构被称为“线性无源总线”。

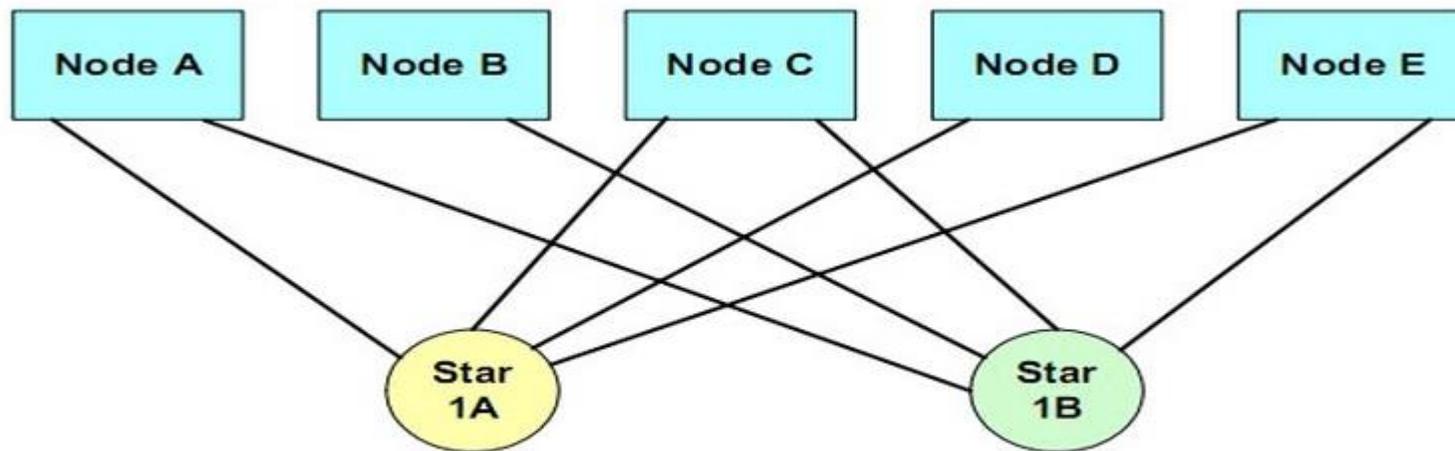


Figure 1-2: Dual channel single star configuration.

FlexRay 架构 – 网络拓扑

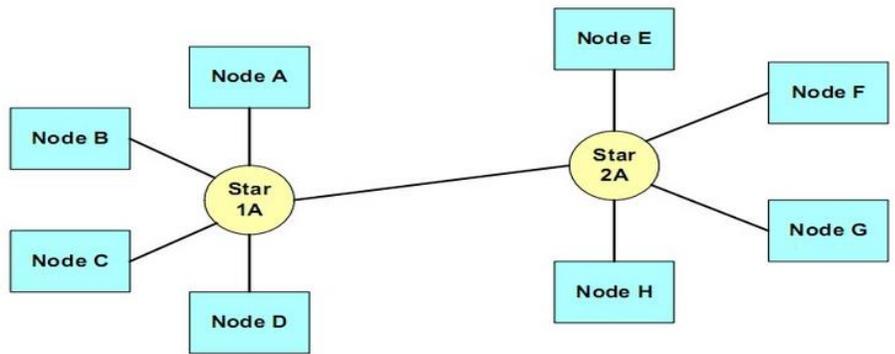


Figure 1-3: Single channel cascaded star configuration.

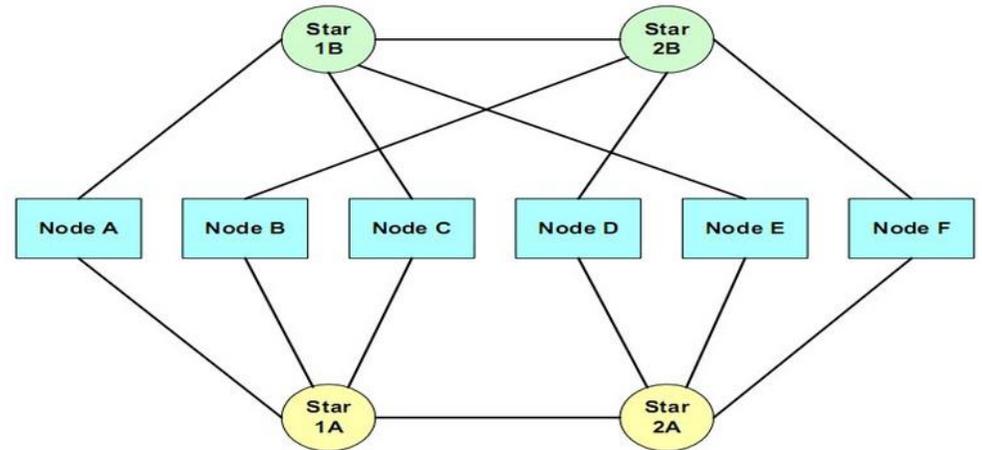


Figure 1-4: Dual channel cascaded star configuration.

- 有源星型可简化分布式协调的某些方面

通过星型的最大延迟为 250 ns，因此未缓冲全部消息

在有源星型网络中，有源星型的一个或多个分支可能设计为线性无源总线或无源星型。

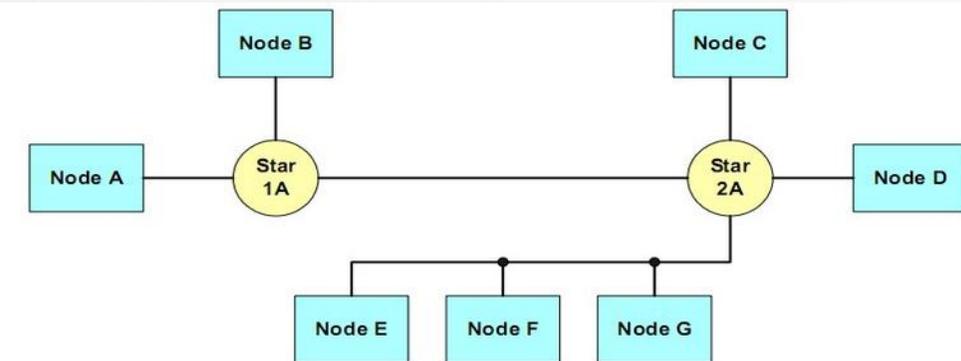


Figure 1-5: Single channel hybrid example.

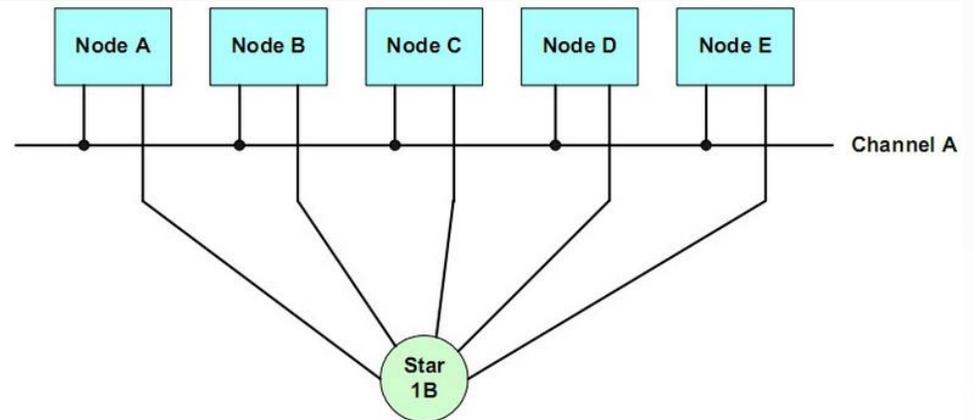
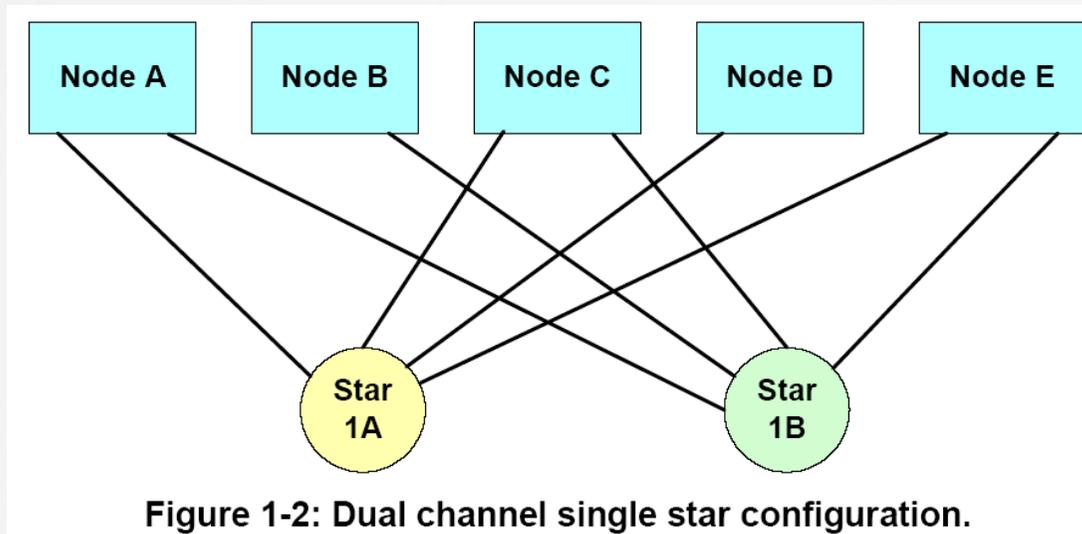


Figure 1-6: Dual channel hybrid example.

FlexRay 架构 – 网络拓扑

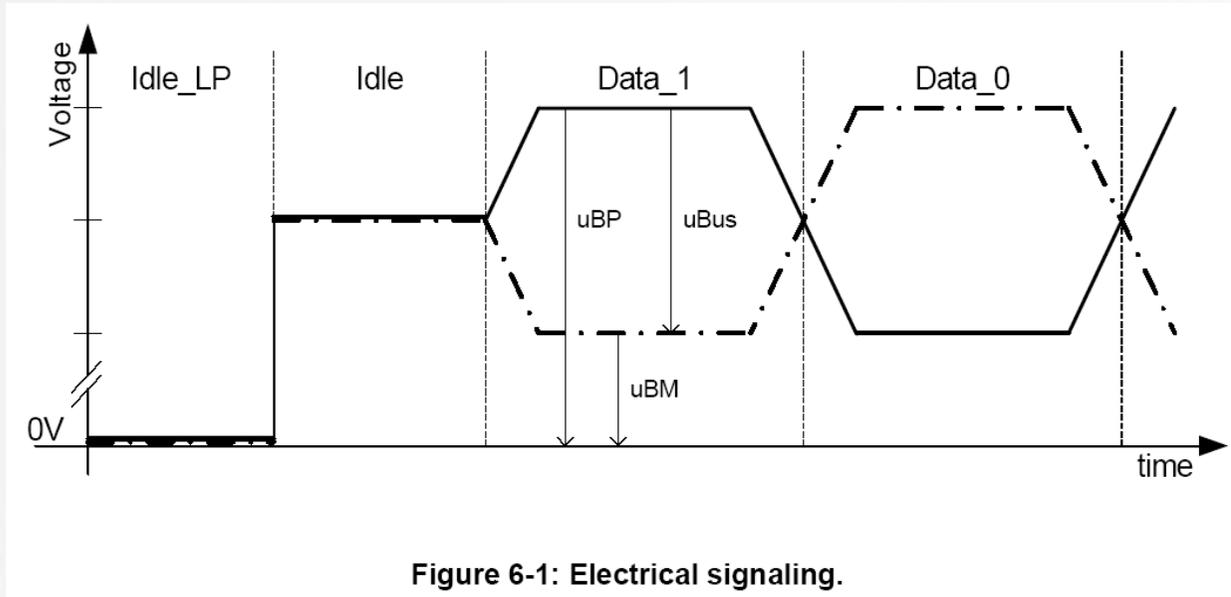
- 旨在消除关键系统的单点故障
 - 这似乎是 FlexRay 线控系统最可能的配置



- 发现 TTP 具有某些分散性总线守护程序问题
 - 与侦听故障网络启动消息的节点相关的问题
 - 最新建议是同样转移到 TTP 的双通道星形配置

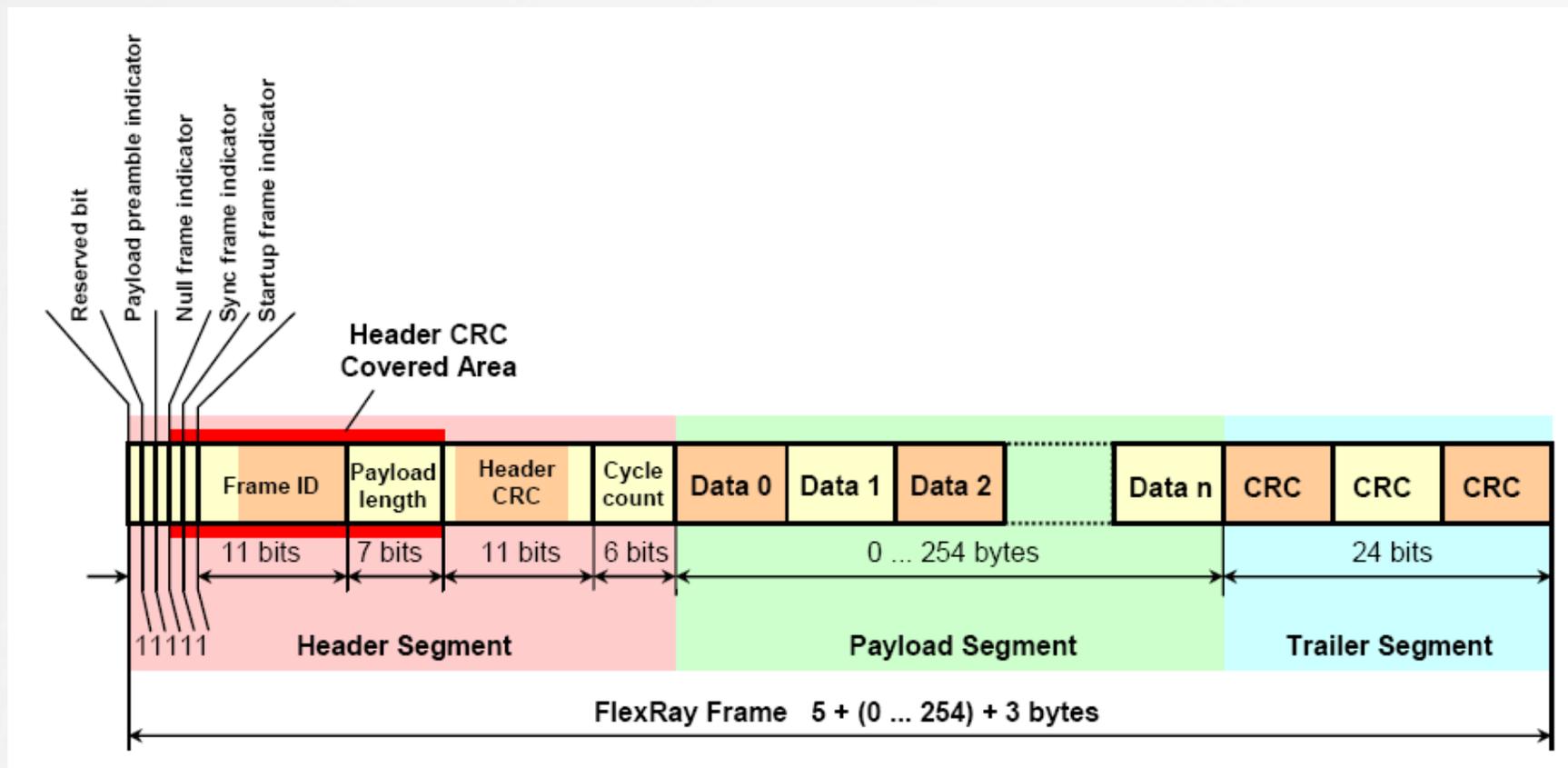
FlexRay – FR 总线上的信号

- 差分 NRZ 编码



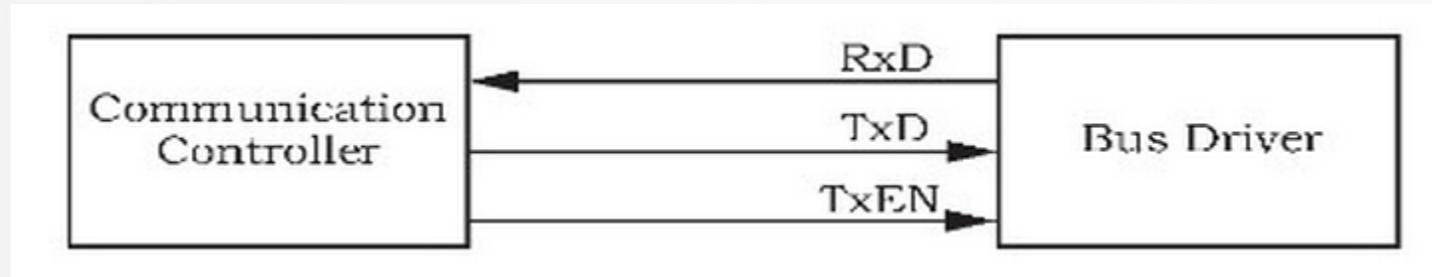
- 10 Mbps 运行速度
 - 与网络长度无关，因为与 CAN 不同，其不使用位仲裁

FlexRay 协议 -- 帧格式



- 根据编码格式将此数据编码为 NRZ 字节

FlexRay 协议 -- 帧编码



FlexRay 协议 -- 编码

- 以 NRZ 字节形式发送的数据

- TSS = 传输启动顺序 (对 5-15 位而言为 LOW)
- FSS = 帧启动顺序 (一个 HI 位)
- BSS = 字节启动顺序 (类似于其他 NRZ 中的启动/停止位)
- FES = 帧结束顺序 (帧的结束符号 - LO + HI)

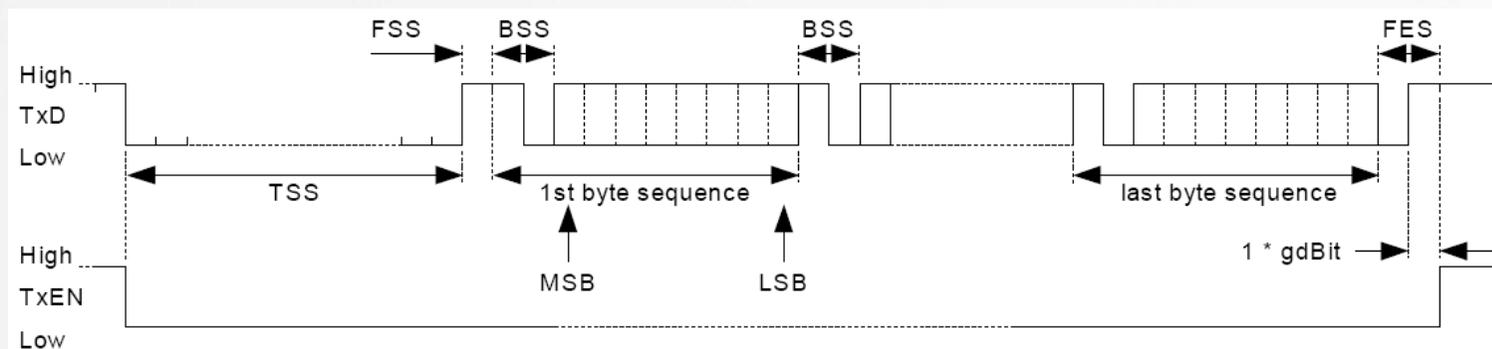
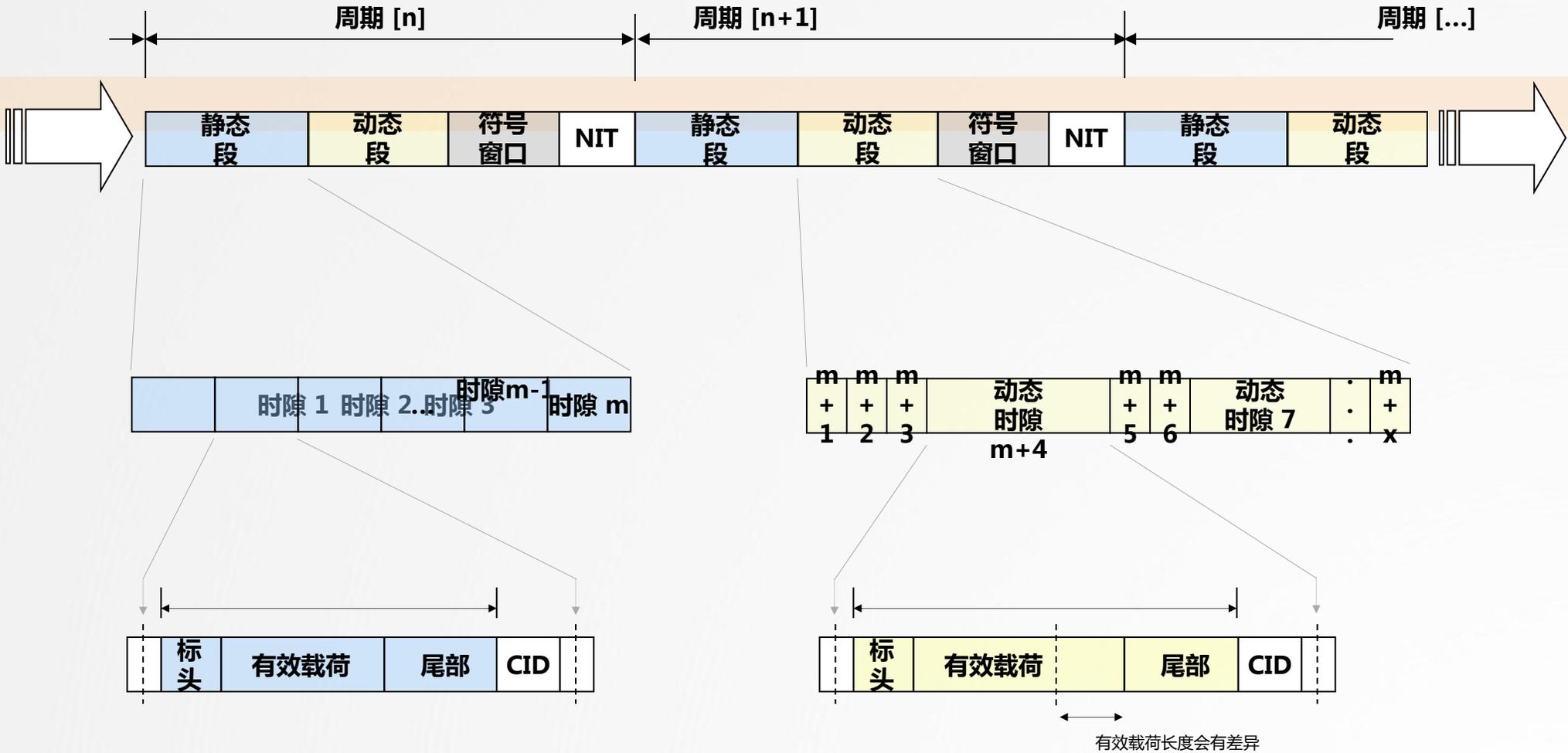


Figure 3-2: Frame encoding in the static segment.

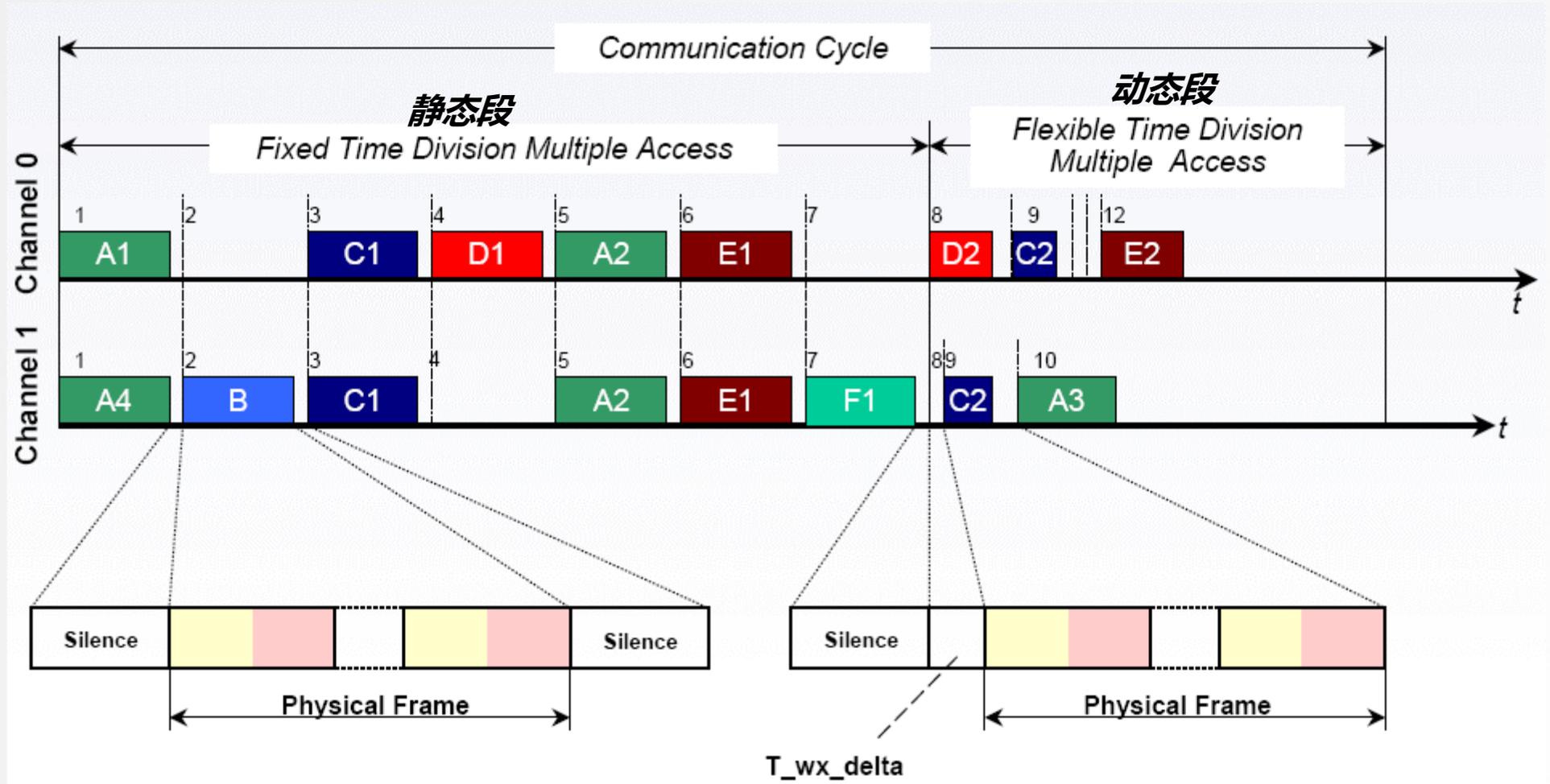
- 动态段帧是相似的

- 添加 DTS = 动态尾部序列字段；有助于将微时隙排队

FlexRay 通信结构



FlexRay 通信周期



FlexRay 消息周期

- 两个主要阶段：静态和动态
 - “临时防火墙” – 两个阶段之间的分区保护每个阶段的时序

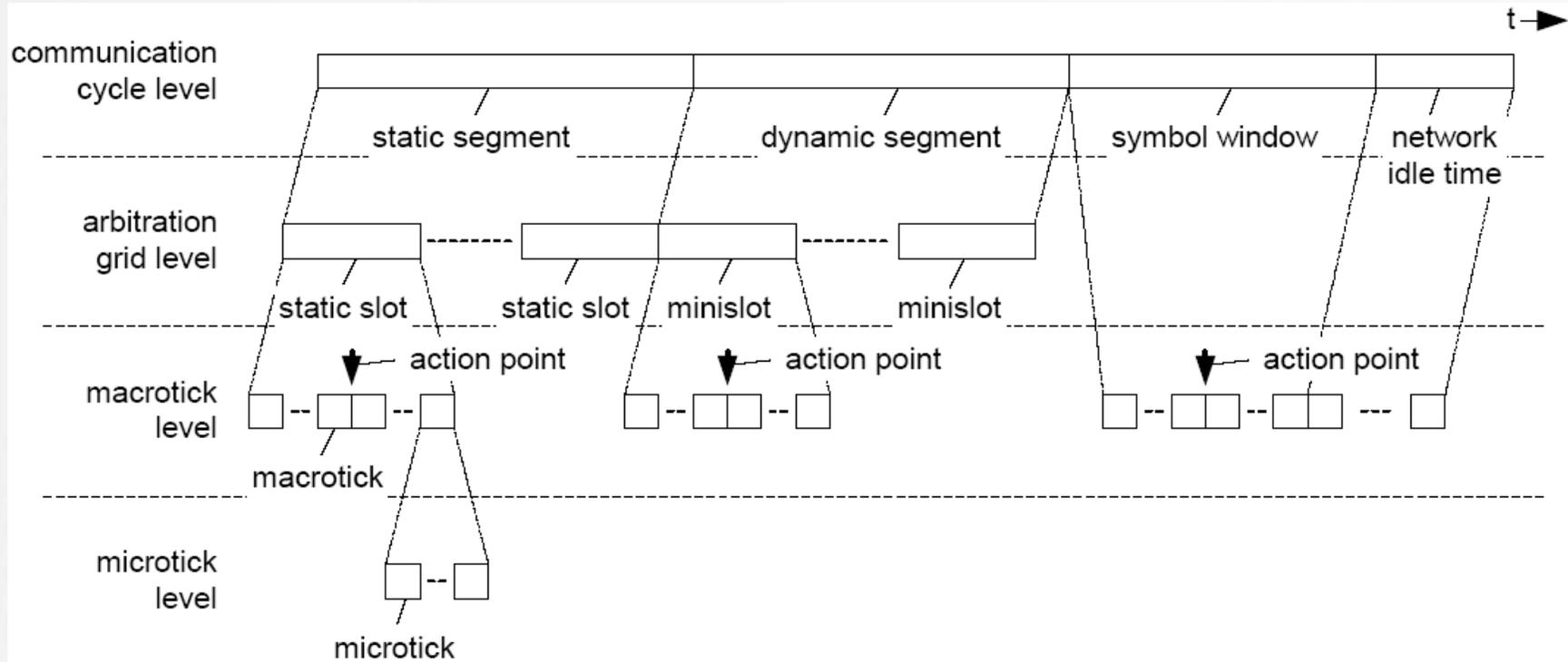


Figure 5-1: Timing hierarchy within the communication cycle.

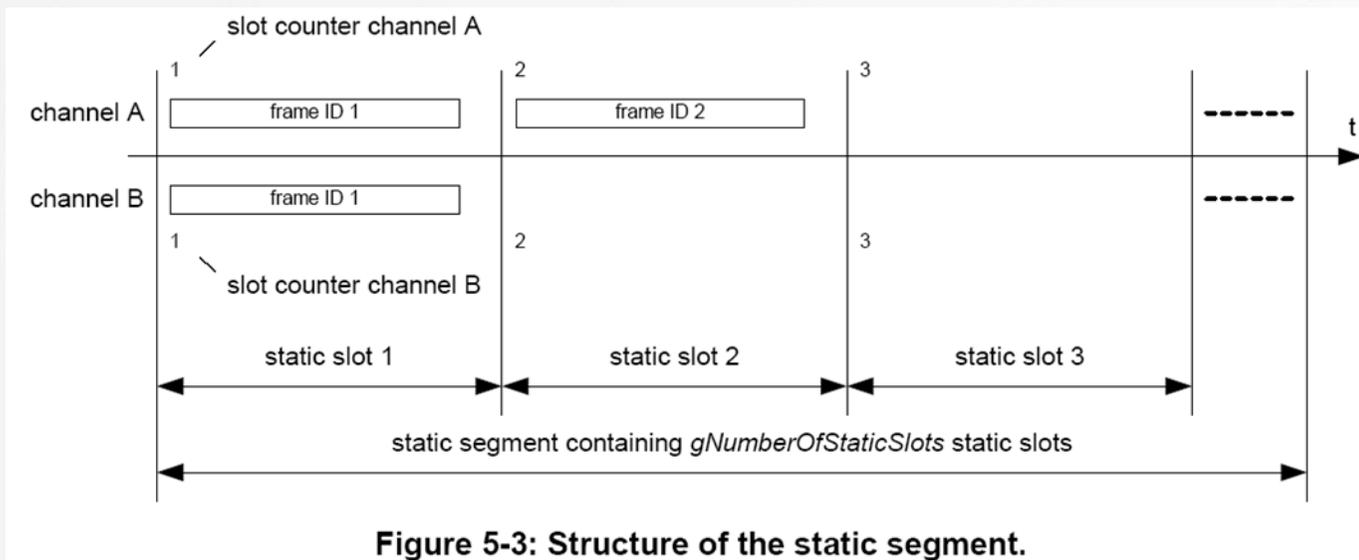
微周期和宏周期

- **微周期级别**
 - 节点自己的内部时基
 - 来自本地振荡器或计数器/计时器的直接或调节值
 - 不与系统的其余部分同步 – 本地自由运行的振荡器
- **宏周期级别**
 - 从群集范围时钟同步算法派生的时间间隔
 - 始终为整数个微周期
 - 但是，不一定每个节点的微周期数相同
 - 微周期数量在运行时改变以实施时钟同步
- **指定的宏周期边界为“动作点”**
 - 传输从此处开始 – 静态；动态；符号窗口
 - 传输在此处结束 – 动态段

静态段

TDMA 消息，最可能用于关键消息

- 所有静态时隙具有相同的微周期数
- 每个通信周期中按顺序重复所有静态时隙
- 无论是否使用，周期内的所有静态时隙时间都将耗费
- 可以为系统配置静态时隙数量；最多为 1023 个时隙



静态段详细信息

两通道工作方式

- 两个通道上的同步帧；其他帧，（可选）1 或 2 个通道
- 不太关键/不太昂贵的节点可能仅连接到一个通道
- 时隙在两个通道上按顺序锁定步进

TDMA 顺序按帧 ID 号升序排列

- 帧编号用于通过软件确定时隙编号
 1. 这不是一种二进制倒计时仲裁机制 – 一次仅一个发射极
 2. （可选）有效载荷区域中存在可能与时隙编号无关的消息 ID
 3. 示例用法：每个节点将其节点号用作帧号并将其消息多路复用到单个时隙上（按消息 ID 来区分）
- 相反，TTP 具有一个 MEDL，后者可以具有子周期
 1. 既不需要帧 ID，也不需要消息 ID
 2. 要管理和协调额外信息

动态段

主要理念是基于事件的通信通道

- 需要仲裁，但是必须是确定性的
- 未使用二进制倒计时（此外，限制可能的介质）

“微时隙”方法

- 可以考虑作为时间压缩 TDMA 方法（下一幻灯片上详细说明）
- 两个通道可以使用独立的消息队列

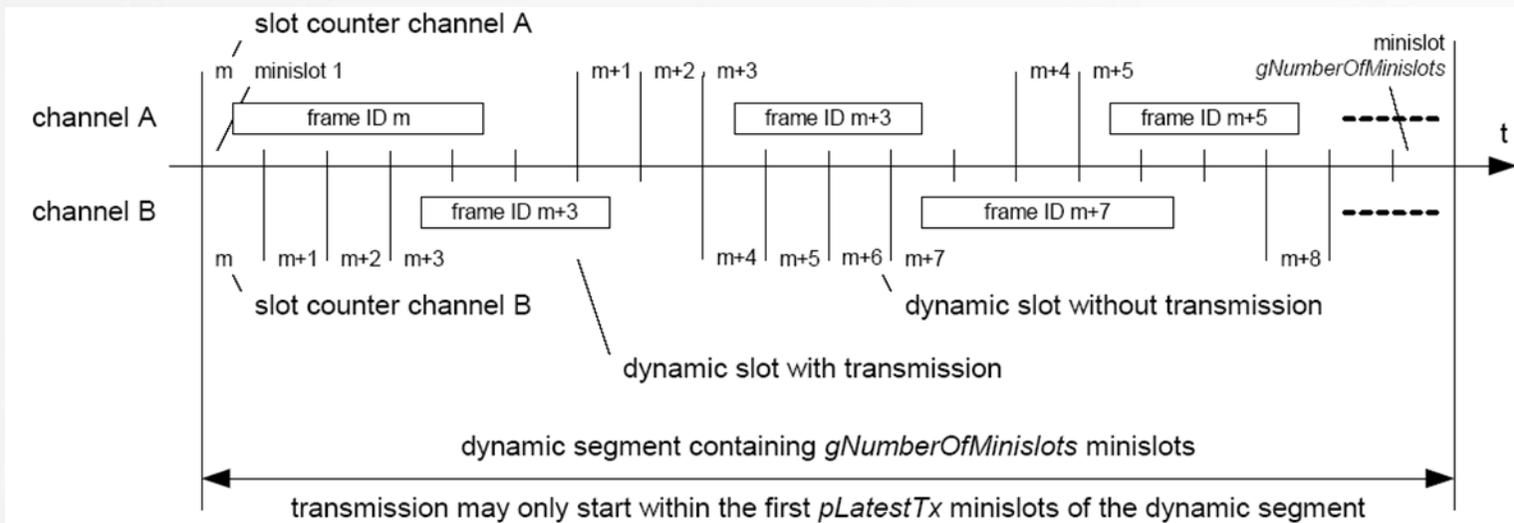
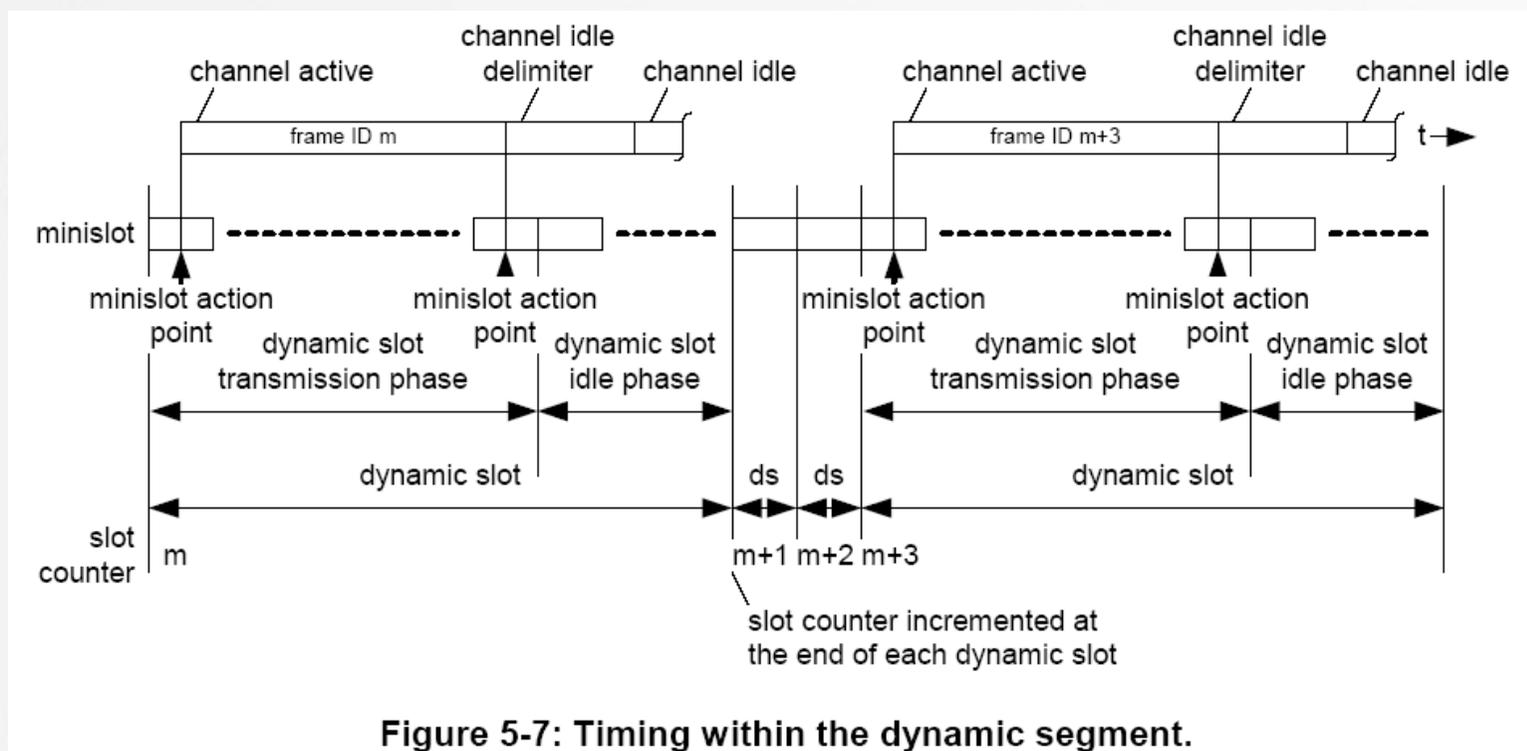


Figure 5-5: Structure of the dynamic segment.

动态段详细信息

主要理念是每个微时隙是一个发送消息的机会

- 如果发送了消息，则微时隙扩展到消息传输
- 如果未发送消息，微时隙将作为短空闲时期不被使用而耗费
- 所有发送器将观察是否发送了消息，以便它们可以对微时隙进行计数



微时隙性能

帧 ID 号用于进行时隙编号

- 第一个动态帧 ID = 最后一个静态帧 ID + 1

动态段有固定的时间长度

- 固定数量的宏周期（划分为微时隙）
- 不一定有足够时间来发送所有动态消息
- 动态段时间结束时，未发送的消息等待下一个周期

网络影响：事件触发的消息

- 首先发送帧 ID 最低的消息
- 每个帧 ID 号只能在每个周期内发送一条消息
- 要纳入动态段的所有消息都将发送
- 这意味着在每个周期中仅发送队列中优先级最高的消息
- 请注意，空闲微时隙会耗用动态段带宽
 - 但是微时隙比消息小得多

FlexRay 系统启动

首先是一个唤醒过程

- 将控制器从“睡眠”转换到“唤醒”
- （不一定由冷启动节点启动）

随后是冷启动，它实际启动 TDMA 操作

- 系统中至少两个冷启动节点
- 对于有 3 个或更多节点的系统，理想数字是三个冷启动节点
 - ✓ 若超过 3 个节点，可能在组成单个组时发生问题（启动时的派系避免功能仅对 3 个节点有效）
- 在系统中预指定冷启动节点

唤醒模式

在一个通道上发送唤醒模式来唤醒其他节点

- 一般会多次发送唤醒模式 – 示例中显示发送两次
- 一次仅在一个通道上发送
 - ✓ 必须处理发送唤醒为单点故障的情况
- 唤醒未得到节点应答
 - ✓ 该情况在启动期间进行处理

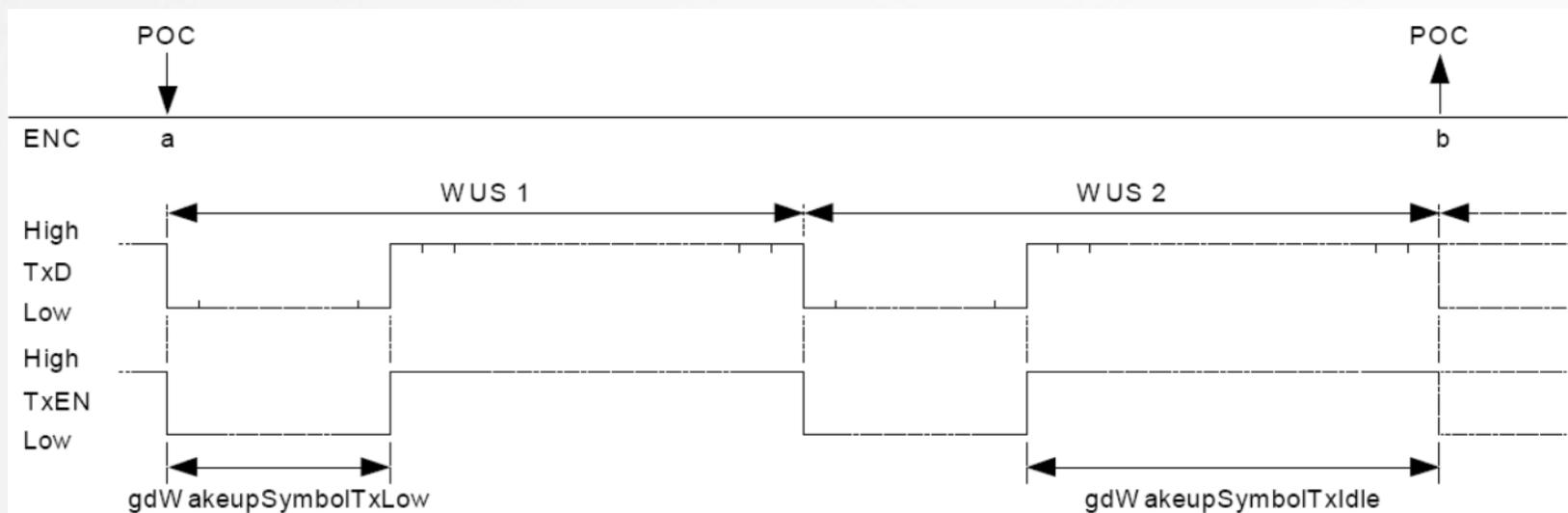


Figure 3-5: Wakeup pattern consisting of two wakeup symbols.

TI 的 FlexRay 实施

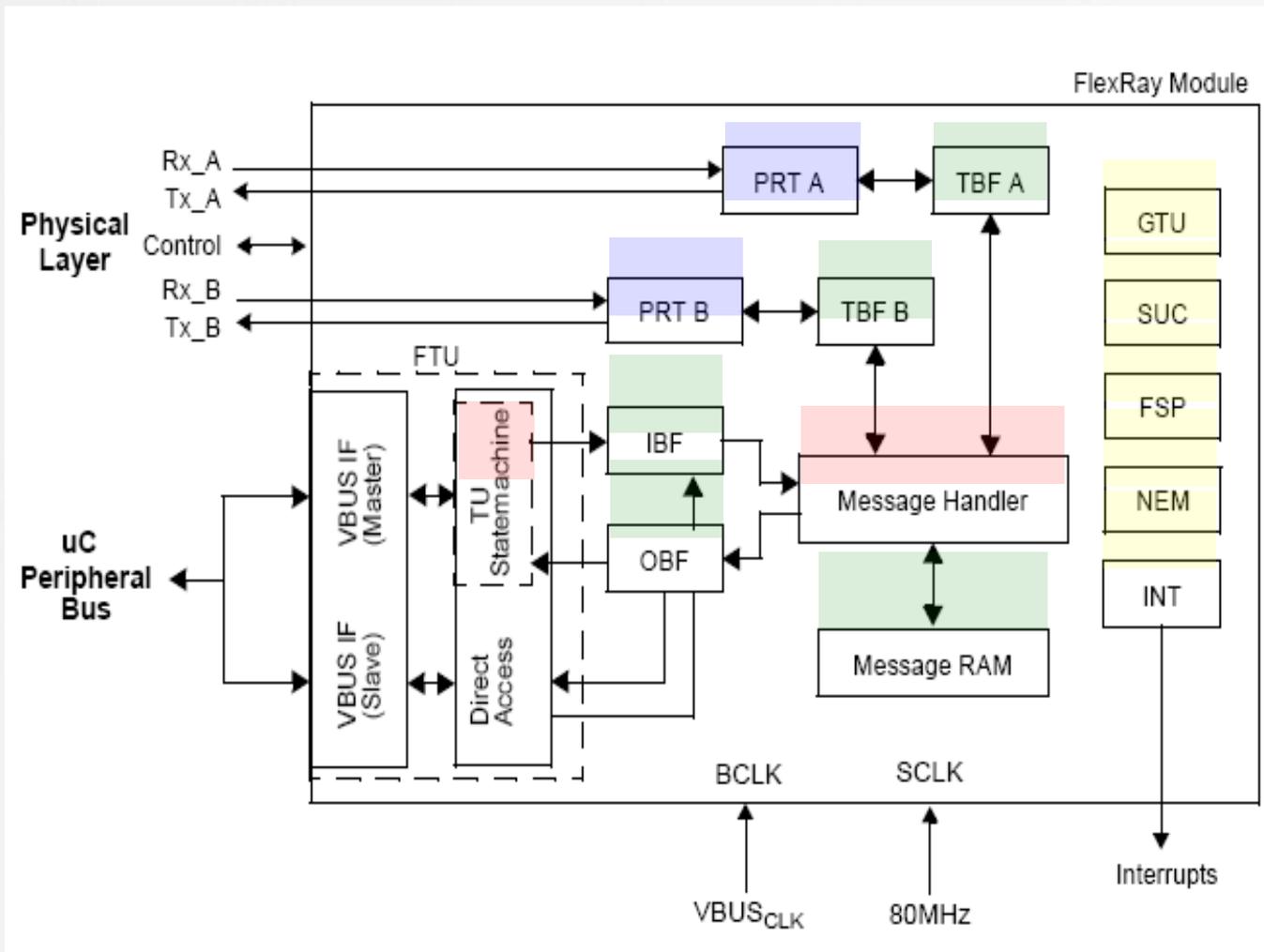
主要特性 (1/2)

- FlexRay 核心 (E-Ray)
- 符合 FlexRay 协议规范 V2.1 修订版 A
- 2 个通道上的数据速率分别最快为 10 兆位/秒
- 8 千字节的消息 RAM 存储如下内容
 - 具有最大 48 字节数据段的 128 个消息缓冲器，或者
 - 具有 254 字节数据段的 30 个消息缓冲器
 - 有效载荷长度可能不同
- 消息 RAM 的奇偶校验保护
- 消息处理器控制
 - 消息 RAM 访问仲裁
 - 接受过滤
 - 维护传输时间表
 - 提供状态信息

主要特性 (2/2)

- 每个消息缓冲器可以配置为
 - 接收缓冲器
 - 传输缓冲器
- 每个消息缓冲器可以分配到
 - 通信周期的静态段
 - 通信周期的动态段
 - 接收 FIFO 的一部分
- 2 个模块接口
 - CPU 通过输入和输出缓冲器对消息缓冲器进行直接访问
 - 在没有 CPU 交互的情况下，通过传输单元在数据存储器与消息缓冲器之间自动进行数据传输
- 过滤帧 ID、通道 ID 和周期计数器
- 可屏蔽模块中断
- 支持网络管理

方框图



FTU :
FlexRay 传输单元

IBF :
输入缓冲器 (与消息处理器相关)

OBF :
输出缓冲器

INT :
中断控制

TBF A/B :
瞬态缓冲器 RAM

PRT A/B :
FlexRay 通道协议控制器

GTU :
全球时间单元

SUC :
系统通用控制

FSP :
帧和符号处理

NEM :
网络管理

消息 RAM 中的消息缓冲器

消息 RAM 具有右图中所示的结构

允许数据分区从消息 RAM 字编号开始：
 $(MRC.LCB + 1) \cdot 4$

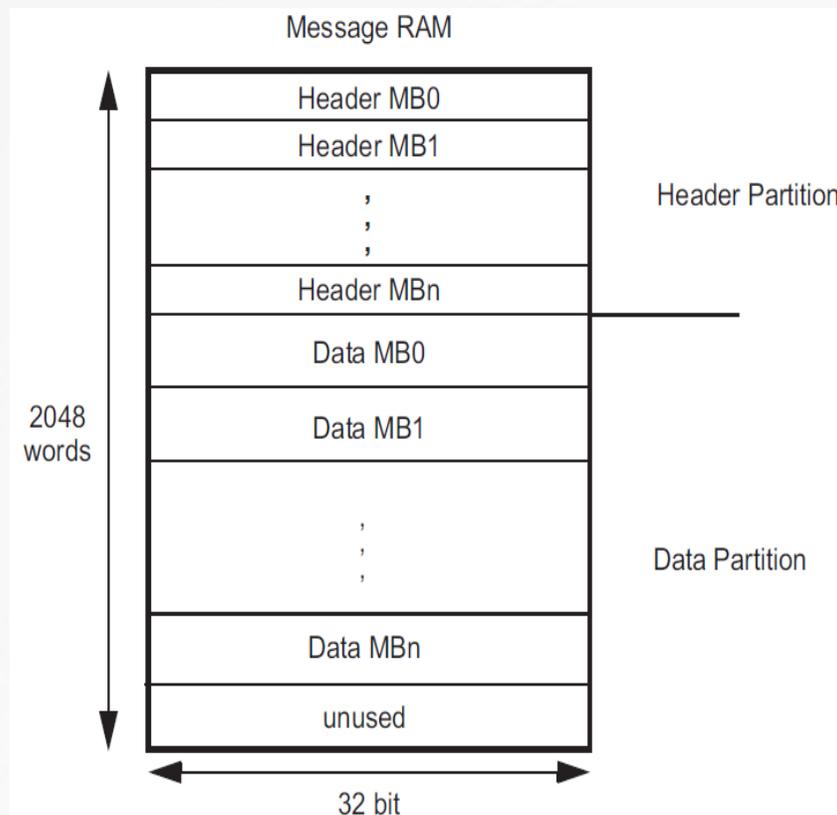
标头分区： 存储 FlexRay 帧的标头段。每个消息缓冲器有一个由四个 32 位字组成的标头。

数据分区： 灵活存储不同长度的数据段。

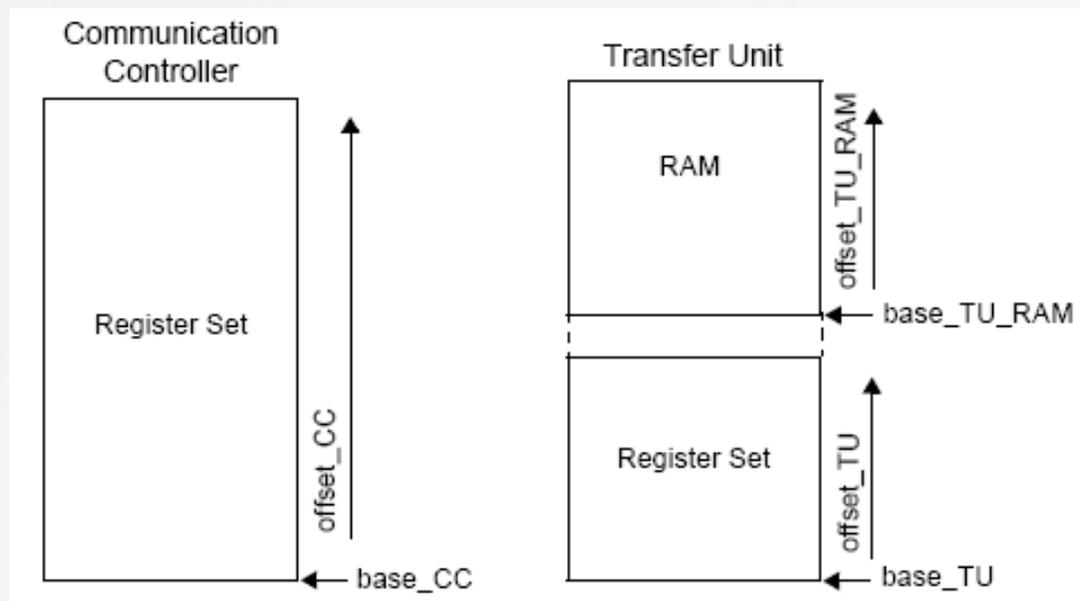
限制： 标头分区 + 数据分区不可占用超过 2048 x 32 位字。

注意：

通信控制器的消息 RAM 未进行内存映射



模块方块内存映射

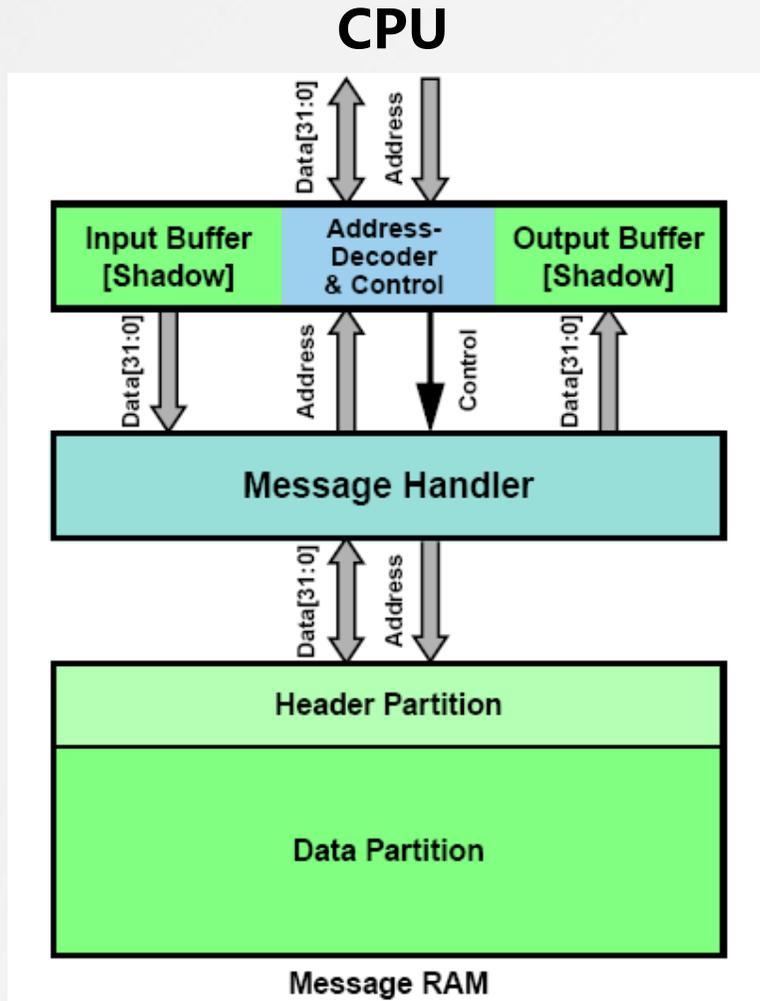


Module	Address Range
FlexRay Communication Controller	0xFFF7_C800 - 0xFFF7_CFFF
FlexRay TU	0xFFF7_A000 - 0xFFF7_A1FF
FlexRay TU RAM	0xFF50_0000 - 0xFF51_FFFF

注意：

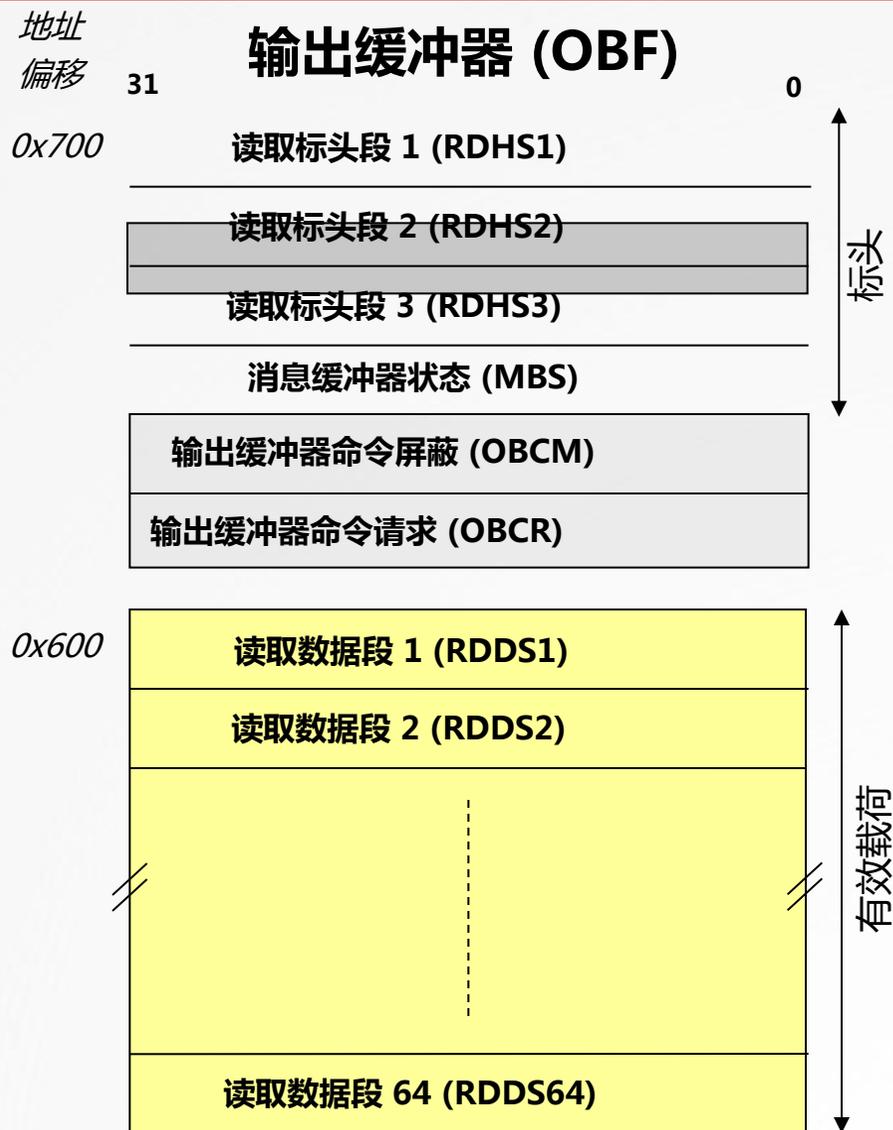
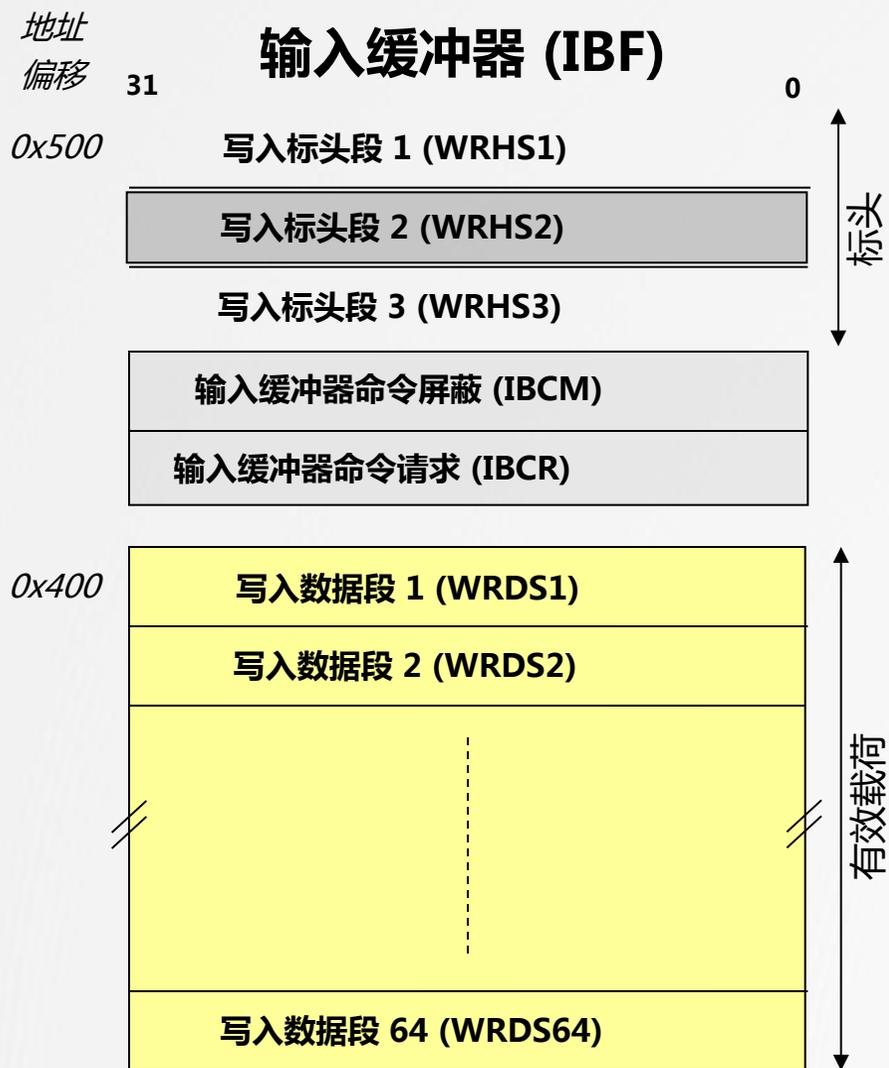
通信控制器的消息 RAM 未进行内存映射

CPU 对消息 RAM 的访问

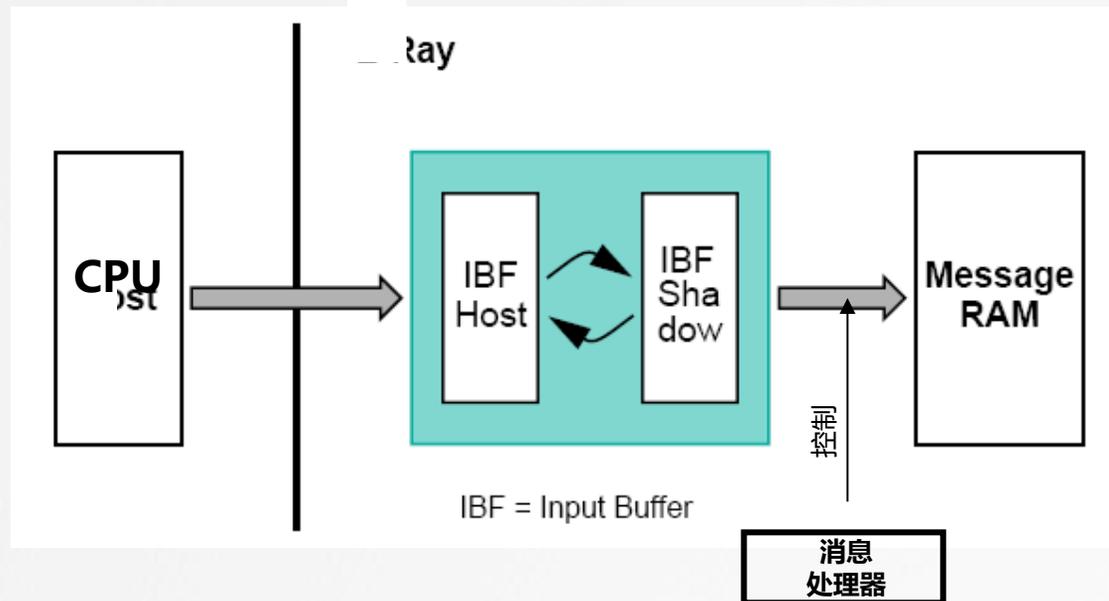


- CPU 与消息缓冲器之间的数据传输通过一个输入和输出缓冲器寄存器组进行
- 每个输入和输出缓冲器寄存器组可以容纳一个消息帧（标头 + 有效载荷）
- 输入缓冲器 (IBF) 和输出缓冲器 (OBF) 组成为双重缓冲器结构
- 消息传输由 CPU 触发

接口寄存器中的数据组织

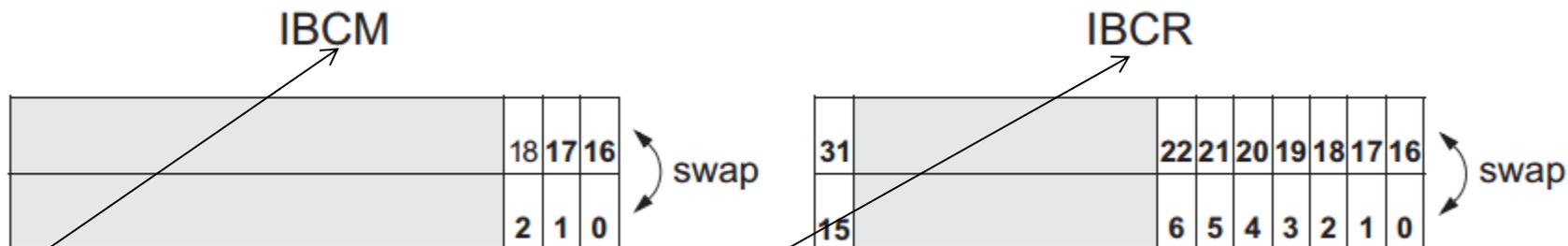


从输入缓冲器到消息 RAM 的数据传输 1/3



- 主机将消息 RAM 中的目标消息缓冲器编号写入输入缓冲器命令寄存器时，IBF 主机和 IBF 影子将交换
- 消息处理器将数据从 IBF 影子传输到消息 RAM 中的目标消息缓冲器时，CPU 可能将下一条消息写入 IBF 主机。

从输入缓冲器到消息 RAM 的数据传输 2/3



Position	Access	Bit	Function
18	r	STXRS	Set Transmission Request shadow ongoing or finished
17	r	LDSS	Load Data Section shadow ongoing or finished
16	r	LHSS	Load Header Section shadow ongoing or finished
2	r/w	STXRH	Set Transmission Request Host
1	r/w	LDSH	Load Data Section Host
0	r/w	LHSH	Load Header Section Host

Pos.	Access	Bit	Function
31	r	IBSYS	IBF Busy Shadow, signals ongoing transfer from IBF shadow to message RAM
22...16	r	IBRS(6-0)	IBF Request Shadow, number of message buffer currently / last updated
15	r	IBSYH	IBF Busy Host, transfer request pending for message buffer referenced by IBRH(6-0)
6...0	r/w	IBRH(6-0)	IBF Request Host, number of message buffer to be updated next

从输入缓冲器到消息 RAM 的数据传输 3/3

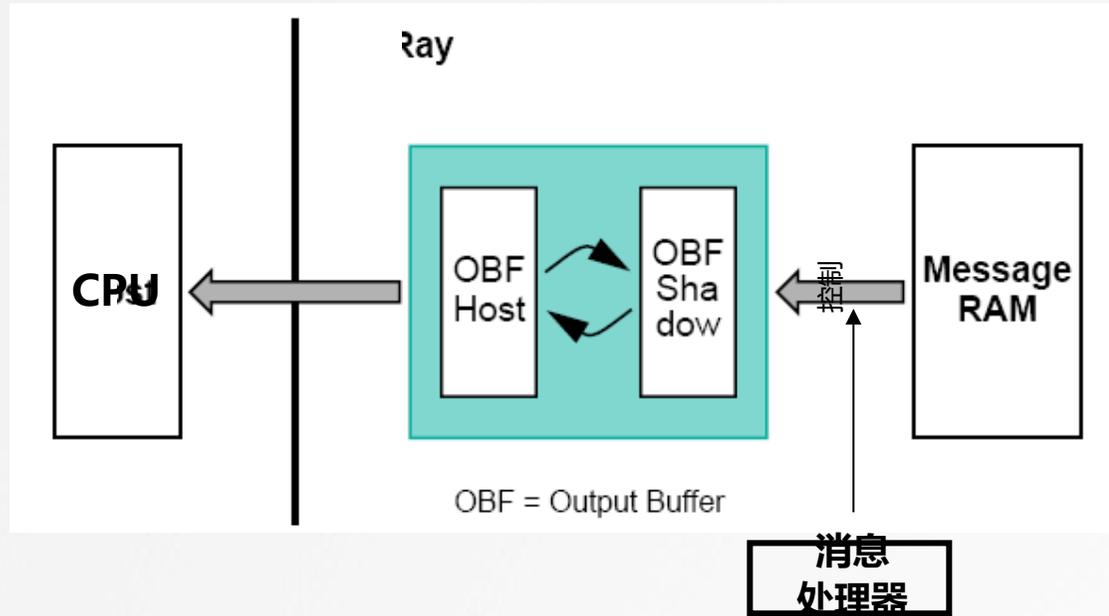
- **8/16/32 位主机访问序列的示例：**
 1. 通过 IBF 配置/更新第 n 个消息缓冲器
 2. 等到 IBCR.IBSYH 复位为止
 3. 将数据段写入 WRDSn
 4. 将标头段写入 WRHS1,2,3
 5. 写入命令屏蔽：写入 IBCM.STXRH、IBCM.LHSH、IBCM.LDSH
 6. 命令数据传输到目标消息缓冲器：写入 IBCR.IBRH(6-0)
- **同样通过 IBF 配置/更新更多消息缓冲器**

```
write_buffer->stxrh = 1; // set transmission request
write_buffer->ldsh = 1; // load data section
write_buffer->lhsh = 0; // load header section
write_buffer->ibsys = 0; // check for input buffer busy shadow
write_buffer->ibsyh = 1; // check for input buffer busy host

// wait for cycle start interrupt flag
Fray_PST->SIR_UN.SIR_UL = 0xFFFFFFFF; // clear all status int. flags
while ((Fray_PST->SIR_UN.SIR_UL & 0x4) == 0x0); // wait for CYCS interrupt flag
Fray_PST->SIR_UN.SIR_UL = 0xFFFFFFFF; // clear all status int. flags

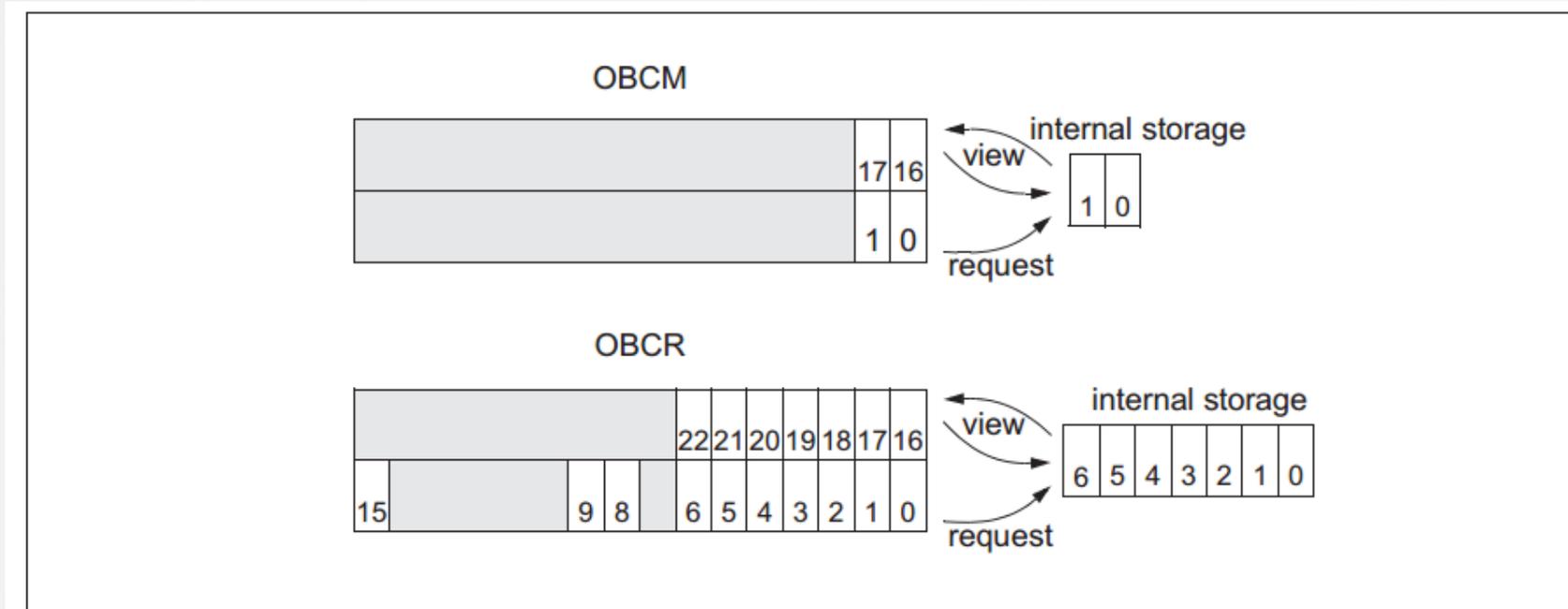
// write payload for buffers
if(node <= SYNC_NODE)
{
    // buffer #0
    write_buffer->ibrh = 0; // input buffer number, sync buffer
    (Fray_PST->WRDS[0] = sync_data[node-1][0]); // Data 1
    (Fray_PST->WRDS[1] = sync_data[node-1][1]); // Data 2
    Fr_TransmitTxLPdu(Fray_PST, write_buffer);
}
}
```

从消息 RAM 到输出缓冲器的数据传输 1/3



- 为了从消息 RAM 读取消息缓冲器，CPU 必须写入输出缓冲器命令寄存器以便触发从目标消息缓冲器到 OBF 影子的数据传输
- 在消息 RAM 与 OBF 影子之间的传输完成之后，CPU 必须设置 **VIEW 位** 来交换 OBF 影子和 OBF 主机
- CPU 从 OBF 主机读取所传输的消息缓冲器时，消息处理器可能将下一条消息从消息 RAM 传输到 OBF 影子。

从消息 RAM 到输出缓冲器的数据传输 2/3



- 将输出缓冲器命令请求寄存器中的位 **OBCR.REQ** 写入到 1 会将来自输出缓冲器命令屏蔽寄存器的位 OBCM.RHSS、OBCM.RDSS 以及来自输出缓冲器命令请求寄存器的位 OBCR.OBRS(6-0) 复制到内部存储器

从消息 RAM 到输出缓冲器的数据传输 3/3

主机对单个消息缓冲器的访问示例：

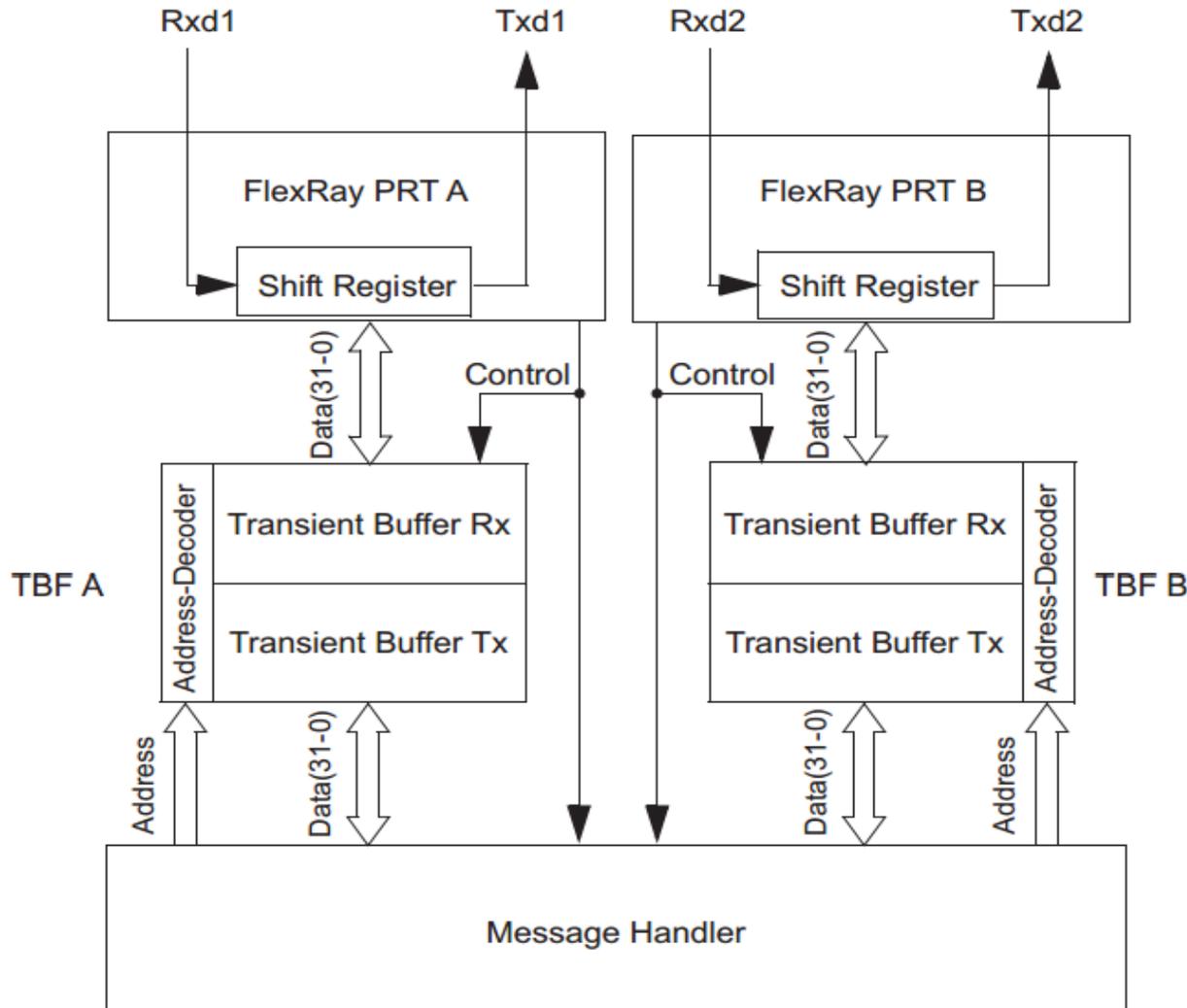
如果必须读出单个消息缓冲器，需要对 OBCR.REQ 和 OBCR.VIEW 进行两次独立的写访问：

- 等到 OBCR.OBSYS 复位为止
- 写入输出缓冲器命令屏蔽 OBCM.RHSS、OBCM.RDSS
- 通过写入 OBCR.OBRS(6-0) 和 OBCR.REQ 来请求从消息缓冲器到 OBF 影子的传输（如果是 8 位主机接口，则必须在 OBCR.REQ 之前写入 OBCR.OBRS(6-0)）。
- 等到 OBCR.OBSYS 复位为止
- 通过写入 OBCR.VIEW = 1 来切换 OBF 影子和 OBF 主机
- 通过读取 RDDSn、RDHS1,2,3 和 MBS 来读出传输的消息缓冲器

```
void Fr_ReceiveRxLPdu(FRAY_ST *Fray_PST, bc *Fr_LSduPtr)
{
    // ensure no transfer in progress on shadow registers
    while (((Fray_PST->OBCR_UN.OBCR_UL) & 0x00008000) != 0);
    Fray_PST->OBCM_UN.OBCM_UL=(((Fr_LSduPtr->rdss & 0x1) << 1) | (Fr_LSduPtr->rhss & 0x1));
    Fray_PST->OBCR_UN.OBCR_UL=((1 << 9) | (Fr_LSduPtr->obrs & 0x7F)); //req=1, view=0
    // wait for completion on shadow registers
    while (((Fray_PST->OBCR_UN.OBCR_UL) & 0x00008000) != 0);

    Fray_PST->OBCM_UN.OBCM_UL=(((Fr_LSduPtr->rdss & 0x1) << 1) | (Fr_LSduPtr->rhss & 0x1));
    Fray_PST->OBCR_UN.OBCR_UL=((1 << 8) | (Fr_LSduPtr->obrs & 0x7F)); //req=0, view=1
}
```

FlexRay 协议控制器对消息 RAM 的访问

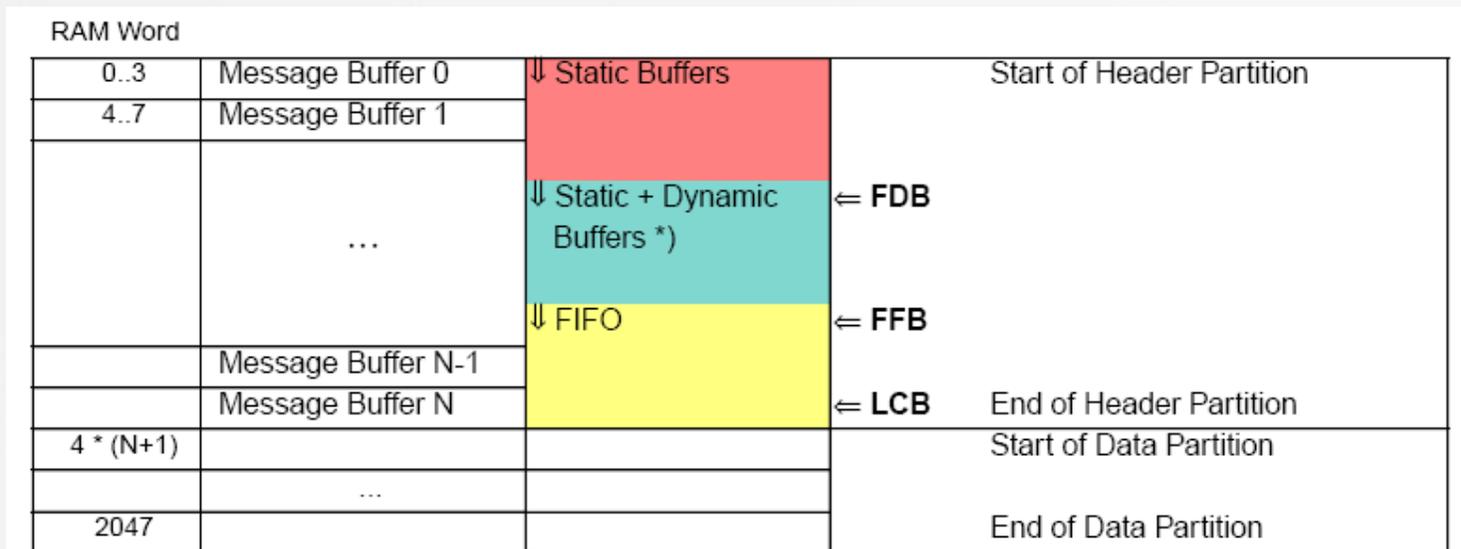


FlexRay 模块包含以下 RAM 部分：

- 1.消息 RAM
- 2.瞬态缓冲器 RAM 通道 A (TBF A)
- 3.瞬态缓冲器 RAM 通道 B (TBF B)
- 4.输入缓冲器 (IBF)
- 5.输入缓冲器影子 (IBFS)
- 6.输出缓冲器 (OBF)
- 7.输出缓冲器影子 (OBFS)
- 8.传输配置 RAM (TCR)
- 9.所有 RAM (除了 TCR) 都是通信控制器核心的一部分。

消息 RAM 配置

每个 FlexRay 缓冲器包含一个标头和一个数据段



标头：

FDB：第一个动态缓冲器

FFB：FIFO 的第一个缓冲器

LCB：上一个已配置的缓冲器

数据：

消息缓冲器的数据段被标头段中配置的数据指针引用。

标头分区

Bit Word	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	P			M B I	T X M	P P I T	C F G	C H B	C H A			Cycle Code													Frame ID								
1	P		Payload Length Received								Payload Length Configured														Tx Buffer: Header CRC Configured Rx Buffer: Header CRC Received								
2	P		R E S	P I S	N F I S	S Y N S	S F I S	S F I S	R C I S			Receive Cycle Count													Data Pointer								
3	P		R E S S	P I S S	N F I S S	S Y N S S	S F I S S	S F I S S	R C I S S			Cycle Count Status						F T B	F T A		M L S T	E S B	E S A	T C I B	T C I A	S V O B	S V O A	C E O B	C E O A	S E O B	S E O A	V F R B	V F R A
...	P		...																														
...	P		...																														

- Frame Configuration
- Filter Configuration
- Message Buffer Control
- Message RAM Configuration
- Updated from received Data Frame
- Message Buffer Status MBS
- Parity Bit
- unused

数据（有效载荷）分区

Bit / Word		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
:		MB0 Data3								MB0 Data2								MB0 Data1								MB0 Data0							
:		unused								unused								MB0 Data5								MB0 Data4							
:		MB1 Data3								MB1 Data2								MB1 Data1								MB1 Data0							
:		unused								unused								MB1 Data5								MB1 Data4							
:		MB2 Data3								MB2 Data2								MB2 Data1								MB2 Data0							
:		unused								unused								MB2 Data5								MB2 Data4							
:		MB3 Data3								MB3 Data2								MB3 Data1								MB3 Data0							
:		o								o								o								o							
:		MB3 Data(k)								MB3 Data(k-1)								MB3 Data(k-2)								MB3 Data(k-3)							
:		MBn Data3								MBn Data2								MBn Data1								MBn Data0							
:		o								o								o								o							
:		o								o								o								o							
:		MBn Data(m)								MBn Data(m-1)								MBn Data(m-2)								MBn Data(m-3)							
:		unused								unused								unused								unused							
:		unused								unused								unused								unused							
2046		unused								unused								unused								unused							
2047		unused								unused								unused								unused							

- 数据分区在标头分区的最后一个字之后开始
- 消息缓冲器数据段的开头和结尾是由数据指针以及消息缓冲器标头段中配置的有效载荷长度确定的
- 程序员必须确保数据指针指向数据分区内的地址

FlexRay 帧的过滤和屏蔽

允许使用以下用于接受/传输过滤的过滤器组合：

- 时隙计数器 + 通道 ID
- 时隙计数器 + 周期计数器 + 通道 ID

接收过滤：

为了将接收的消息存储在消息缓冲器中，所有配置的过滤器必须匹配。

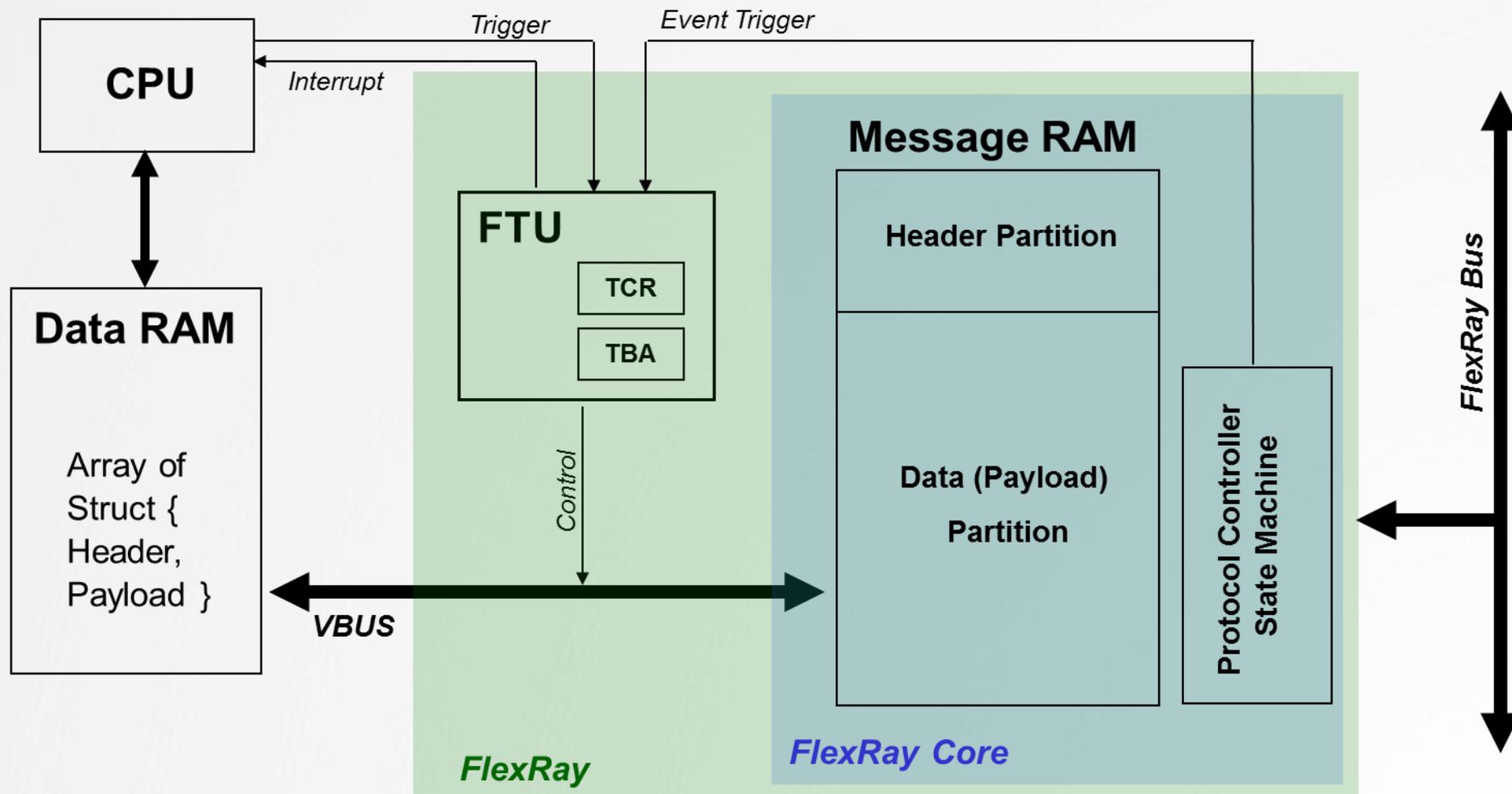
传输过滤：

在同已配置通道上的已配置帧 ID 相对应的时隙中将传输一条消息。

如果启用了周期计数器过滤，已配置的周期过滤器值也必须匹配。

FlexRay 传输单元 (FTU)

FTU 数据传输方案



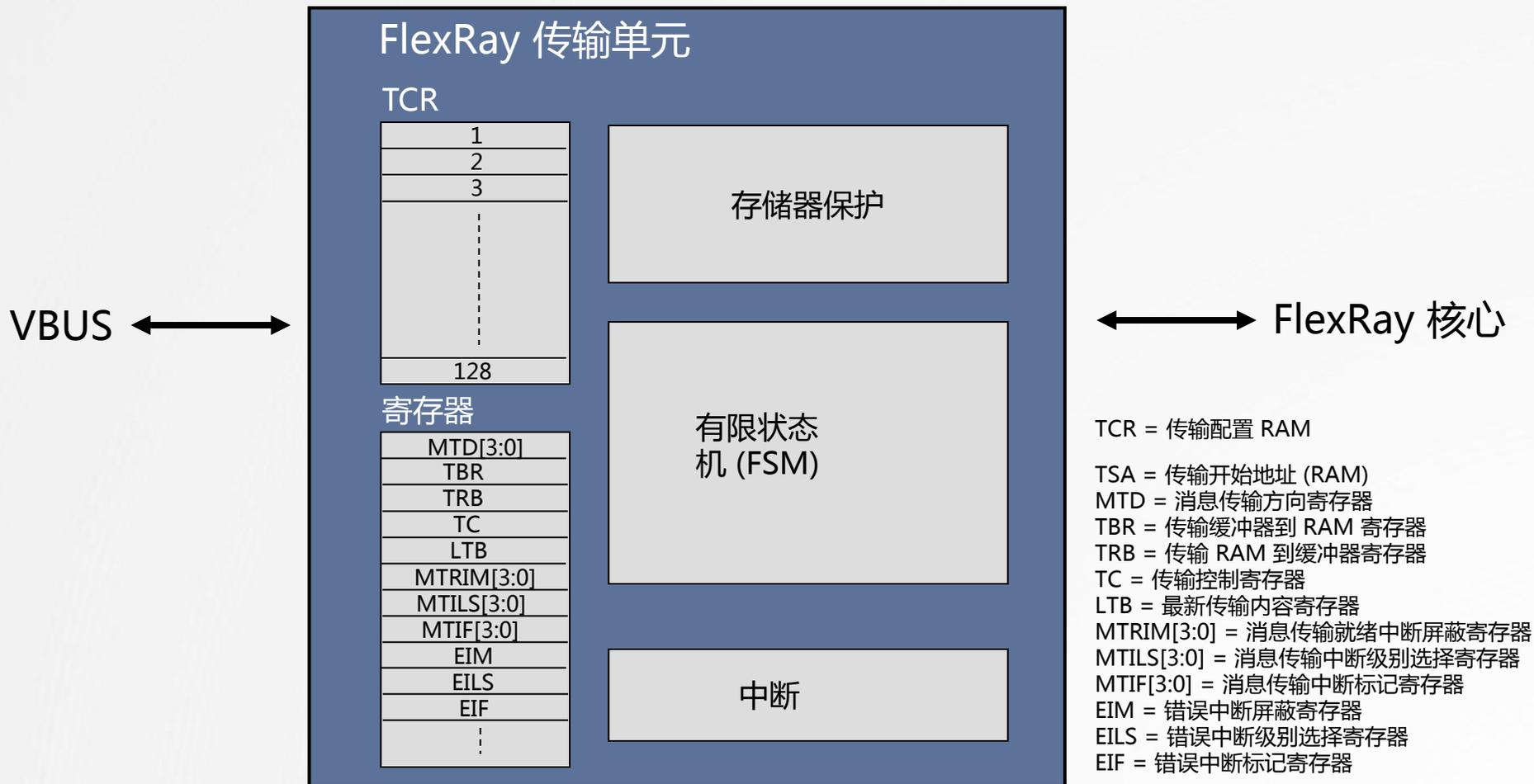
FlexRay 传输单元主要特性 (1/2)

- 无需 CPU 交互即可进行数据传输
 - 从 FlexRay 消息 RAM 到数据 RAM (读取)
 - 从数据 RAM 到 FlexRay 消息 RAM (写入)
- 传输类型
 - 数据和标头段
 - 仅限标头段
 - 仅限数据段
- 传输配置 RAM (具有奇偶校验)
 - 配置传输序列
 - 奇偶校验保护
- 通过触发器启动传输
 - 由 CPU 驱动 (单个传输序列)
 - 由事件驱动 (单个或连续传输序列)

FlexRay 传输单元主要特性 (2/2)

- 不同传输条件
 - 如果各消息缓冲器的状态标记（标头段）已更新
 - 如果各消息缓冲器的数据段已更新
 - 始终
- 消息缓冲器传输完成时生成可屏蔽的中断
- 存储器保护单元
 - 可以定义一个存储器部分（开始和结束地址）
 - 在复位之后未设置存储器部分

FlexRay 传输单元 (FTU) 子块



传输配置 RAM (TCR)

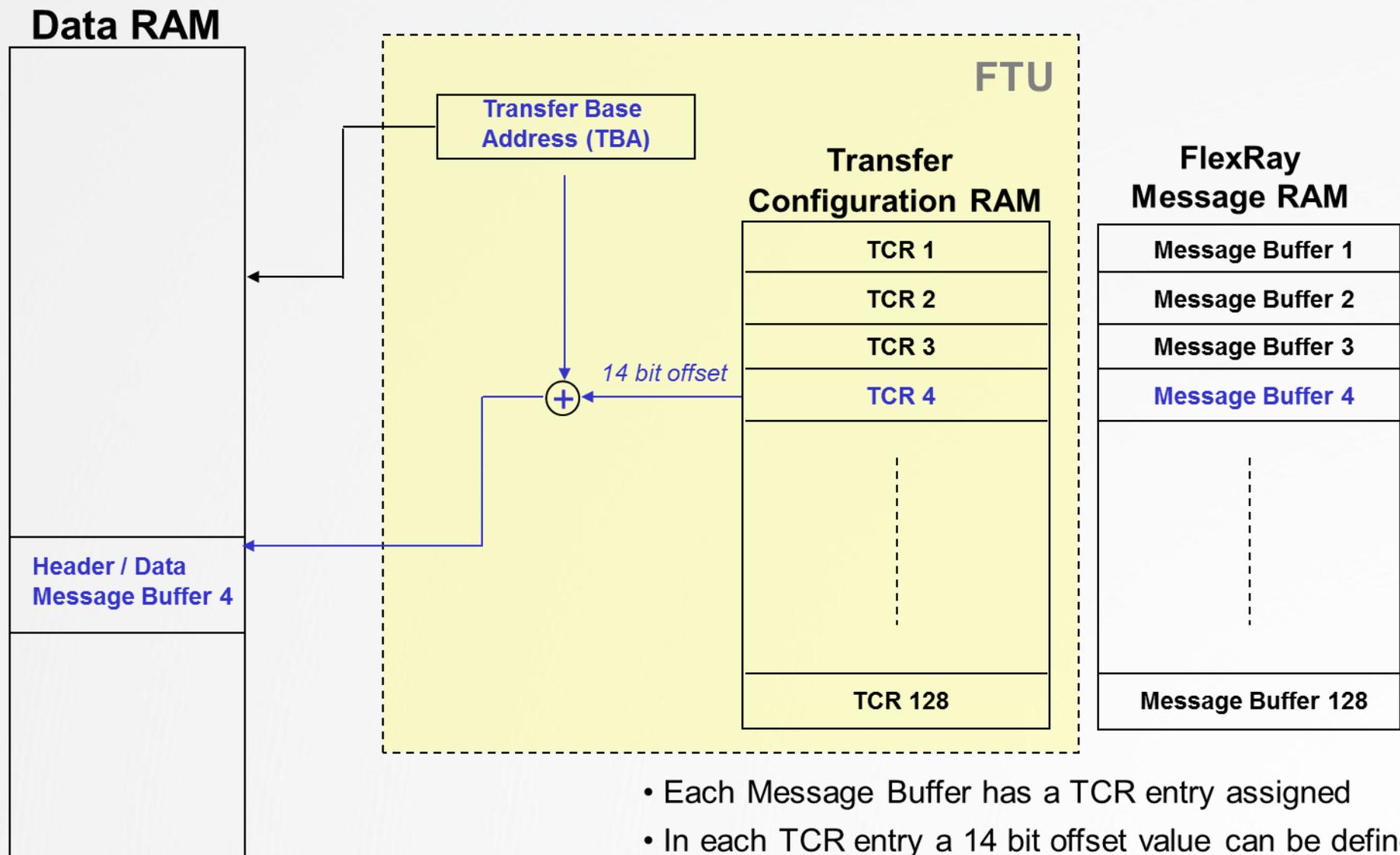
- 用于设置传输序列
- 条目号等于 FlexRay 的消息缓冲器编号
- 按条目号从低到高执行传输周期
- 传输配置 RAM 具有奇偶校验保护



每个条目定义了：

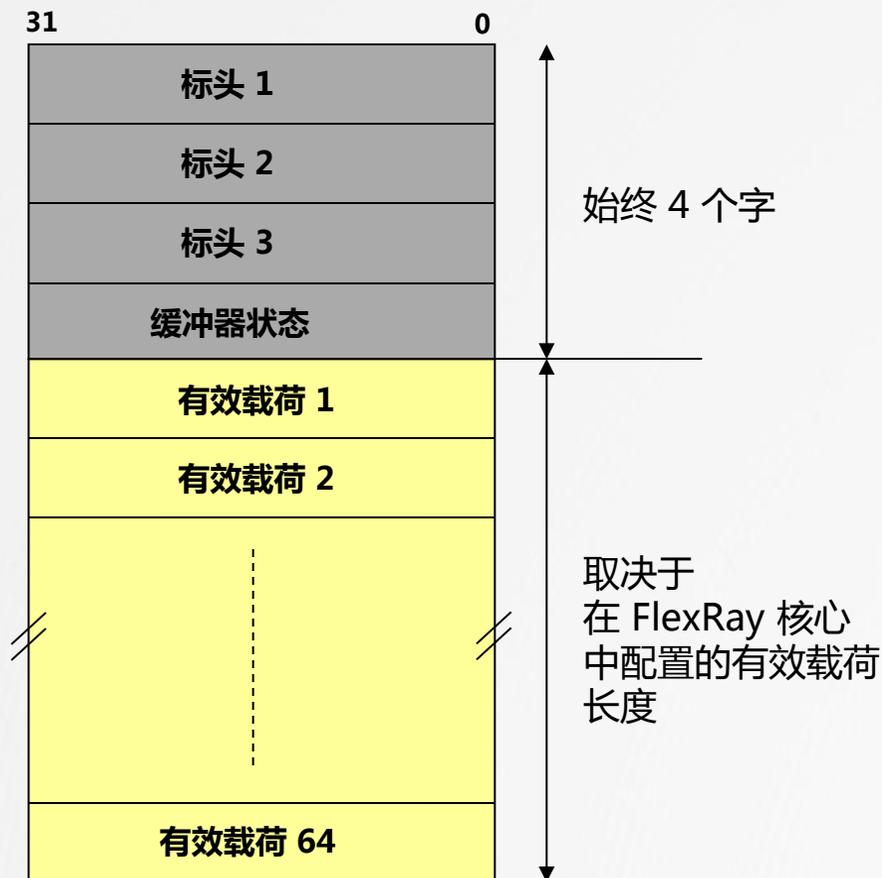
- 传输地址偏移
- 传输方向
- 标头和/或有效载荷的传输
- 针对 FlexRay 核心的自动发送触发器

FTU 数据寻址



- Each Message Buffer has a TCR entry assigned
- In each TCR entry a 14 bit offset value can be defined

数据 RAM 中的数据包存储顺序



- 标头段始终为 4 个字。
- 有效载荷长度在 FlexRay 核心中配置
- TU 传输的数据可选为：
 - 数据和标头段
 - 仅限标头段
 - 仅限数据段
- 缓冲器状态仅从通信控制器传输到系统存储器

可能的传输序列模式

- 手动通过设置触发器传输到系统存储器寄存器 (TTSM) 或触发器传输到通信控制器寄存器 (TTCC) 中的相应位
- 事件驱动使用 “允许发生事件时传输到系统存储器寄存器” (ETESM)
 - 仅从 FlexRay 通信控制器传输到系统存储器
 - 通过 FlexRay 总线完成帧的接收或传输时触发传输
- 连续使用 “在发生事件时准许传输到系统存储器” (CESM)

传输状态指示

有 3 个指示传输状态的寄存器

- 传输状态当前缓冲器 (TSCB) 显示当前传输缓冲器状态
- 上次缓冲器传输到通信控制器 (LTBCC) 显示上次完成的从缓冲器到通信控制器的传输
- 上次缓冲器传输到系统存储器 (LTBSM) 显示上次完成的从缓冲器到系统存储器的传输

传输优先级

- TU 按消息缓冲器编号从低到高地传输消息缓冲器。
 - 如果同一缓冲器在
 - 触发器传输到通信寄存器 (TTCC)
 - 触发器传输到系统存储器寄存器 (TTSM)
- ⇒ 中均未定，则 TTCC 与 TTSM 之间的优先级由 TU 全局控制寄存器中的传输优先级位 (GC.PRIO) 确定。

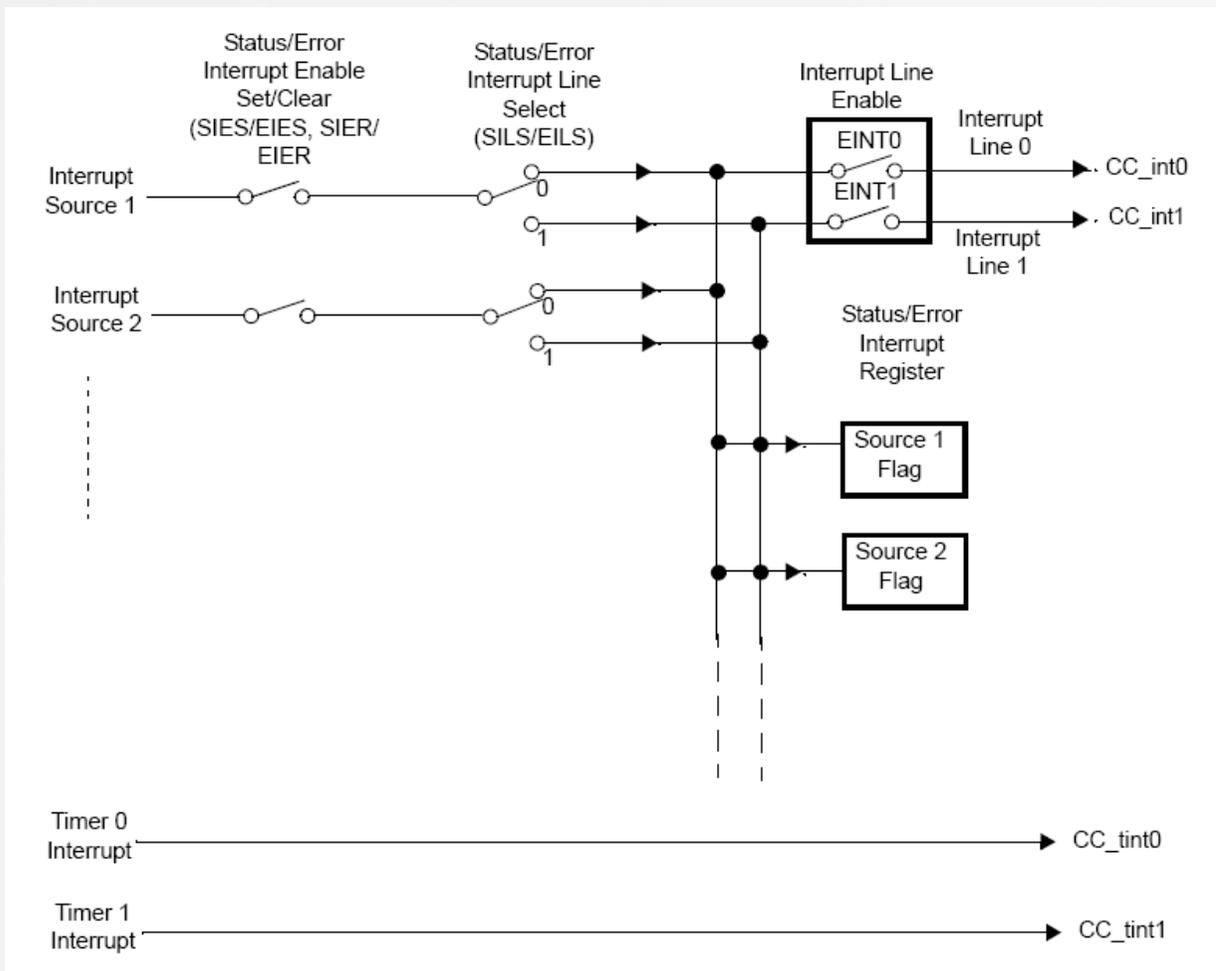
存储器保护机制

- 可以在数据 RAM 中定义一个存储器部分，其允许传输单元状态机进行读取和写入访问。
- 如果发生保护违例
 - 将不会进行传输
 - 将设置一个标记
 - 将激活存储器保护违例中断
 - 将禁用传输单元状态机
- 将保护地址范围的设置默认为：
 - 0x00000000 (对于开始地址)
 - 0x00000000 (对于结束地址)

⇒ 必须先设置有效地址范围，然后才可以使用传输单元。

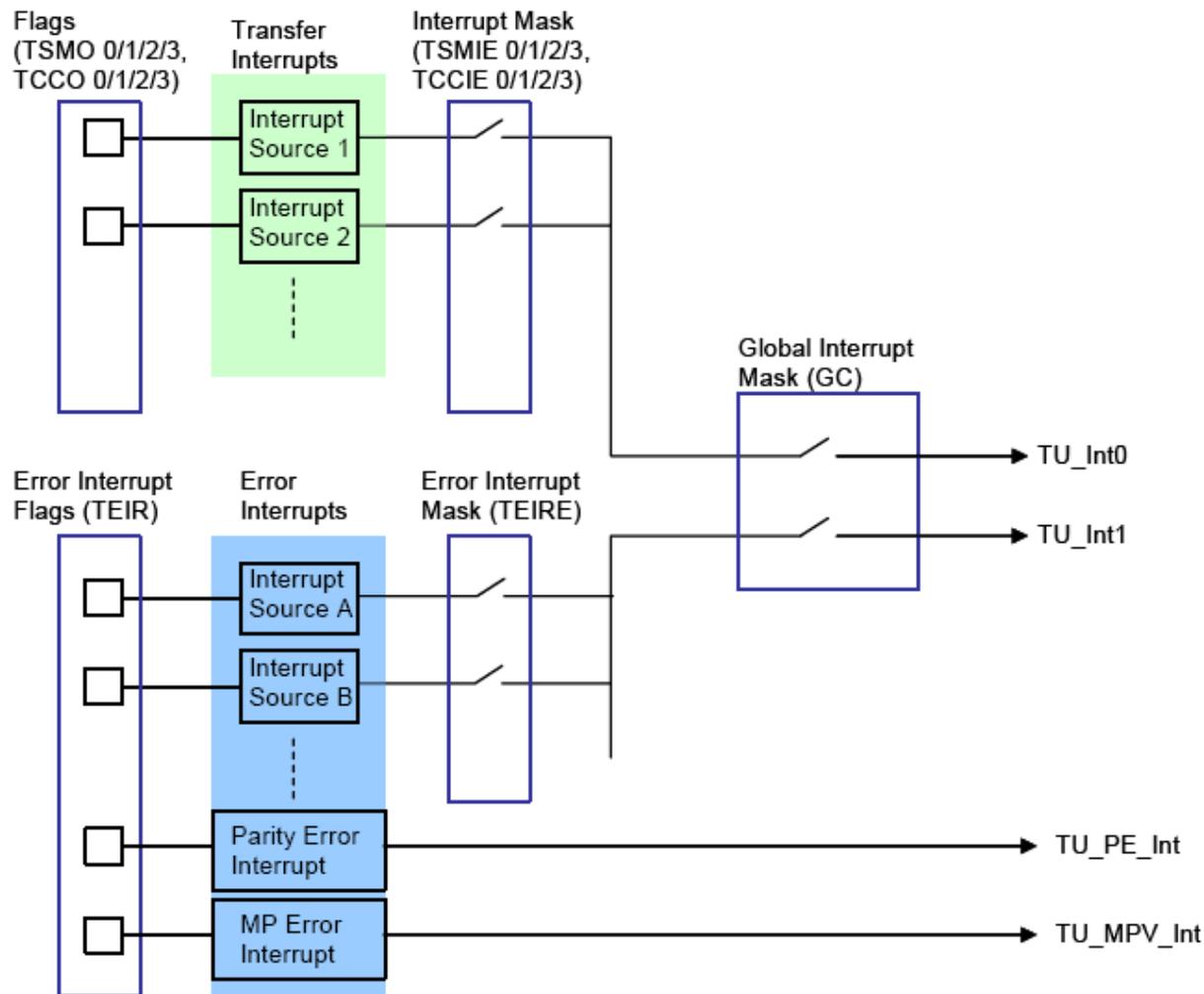
FlexRay 中断结构

FlexRay 核心中断结构



- 可以生成错误和状态中断
- 每个中断源可以映射到 2 条中断线之一
- Timer0 和 Timer1 中断具有专用中断线
- 每个中断源的中断标记

FTU 中断结构



- 每个传输通道可以生成一个传输中断
- 奇偶校验和存储器保护错误具有专用中断线

- 更多的错误中断可能为：

- 读取传输错误
- 写入传输错误
- 传输未就绪
- CPU 被禁止访问 IBF 或 OBF

- 每个中断源的中断标记

重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 对所提供的产品和服务进行更正、修改、增强、改进或其它更改, 并有权根据 JESD48 最新标准中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。仅在 TI 保证的范围内, 且 TI 认为有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定, 否则没有必要对每种组件的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用 TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险, 客户应提供充分的设计与操作安全措施。

TI 不对任何 TI 专利权、版权、屏蔽作品权或其它与使用了 TI 组件或服务的组合设备、机器或流程相关的 TI 知识产权中授予的直接或间接版权限作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息, 不能构成从 TI 获得使用这些产品或服务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可, 或是 TI 的专利权或其它知识产权方面的许可。

对于 TI 的产品手册或数据表中 TI 信息的重要部分, 仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时, 如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分, 则会失去相关 TI 组件或服务的所有明示或暗示授权, 且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意, 尽管任何应用相关信息或支持仍可能由 TI 提供, 但他们将独自负责满足与其产品及其应用中使用 TI 产品相关的所有法律、法规和安全相关要求。客户声明并同意, 他们具备制定与实施安全措施所需的全部专业技术和知识, 可预见故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理造成的任何损失。

在某些场合中, 为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特有的可满足适用的功能安全性标准和要求的终端产品解决方案。尽管如此, 此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III (或类似的生命攸关医疗设备) 的授权许可, 除非各方授权官员已经达成了专门管控此类使用的特别协议。

只有那些 TI 特别注明属于军用等级或“增强型塑料”的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同意, 对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用, 其风险由客户单独承担, 并且由客户独自负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品, 这些产品主要用于汽车。在任何情况下, 因使用非指定产品而无法达到 ISO/TS16949 要求, TI 不承担任何责任。

	产品		应用
数字音频	www.ti.com.cn/audio	通信与电信	www.ti.com.cn/telecom
放大器和线性器件	www.ti.com.cn/amplifiers	计算机及周边	www.ti.com.cn/computer
数据转换器	www.ti.com.cn/dataconverters	消费电子	www.ti.com.cn/consumer-apps
DLP® 产品	www.dlp.com	能源	www.ti.com.cn/energy
DSP - 数字信号处理器	www.ti.com.cn/dsp	工业应用	www.ti.com.cn/industrial
时钟和计时器	www.ti.com.cn/clockandtimers	医疗电子	www.ti.com.cn/medical
接口	www.ti.com.cn/interface	安防应用	www.ti.com.cn/security
逻辑	www.ti.com.cn/logic	汽车电子	www.ti.com.cn/automotive
电源管理	www.ti.com.cn/power	视频和影像	www.ti.com.cn/video
微控制器 (MCU)	www.ti.com.cn/microcontrollers		
RFID 系统	www.ti.com.cn/rfidsys		
OMAP应用处理器	www.ti.com.cn/omap		
无线连通性	www.ti.com.cn/wirelessconnectivity	德州仪器在线技术支持社区	www.deyisupport.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2016, Texas Instruments Incorporated