国际标准 ISO 11898 - 1 第二版2015-12-15

道路车辆-控制器区域网络(CAN)一第1部分:数据链路层和物理信令

# 前言

ISO(国际标准化组织)是由各国标准机构(ISO成员机构)组成的世界性联合会。制订国际标准的工作通常是通过ISO技术委员会进行的。每一个对已设立技术委员会的某一主题感兴趣的成员机构都有权参加该委员会。政府和非政府的国际组织也与ISO保持联系，参与这项工作。ISO与国际电工委员会(IEC)在电工技术标准化的所有问题上密切合作。

ISO/IEC指令第1部分描述了用于编制本文件和用于进一步维护的程序。特别需要注意的是，不同类型的ISO文件需要不同的批准标准。本文件是根据ISO/IEC指令第2部分的编辑规则起草的(见www.iso.org/directives)。

提请注意的是，本文件的一些内容可能是专利权的主题。国际标准化组织不负责识别任何或所有此类专利权。在文件开发过程中确定的任何专利权的详细信息将在介绍和/或收到的ISO专利声明列表中(见www.iso.org/patents)。

本文档中使用的任何商品名称都是为方便用户而提供的信息，并不构成背书。

关于ISO有关合格评定的具体术语和表达的含义的解释，以及关于ISO在技术性贸易壁垒(TBT)中遵守WTO原则的信息，请参阅以下网址:前言-补充信息。

负责本文件的委员会是ISO/TC 22，道路车辆，小组委员会SC 31，数据通信。

这个第二版取消并取代了第一版(ISO 11898-1:2003)，它已经经过了技术上的修订。它还包含了勘误表ISO 11898-1:2003/Cor 1:2006。

ISO 11898由以下部分组成，总标题为道路车辆-控制器区域网络(CAN):

-第1部分:数据链路层和物理信令

-第2部分:高速媒体接入单元

-第3部分:低速，容错，依赖介质的接口

-第4部分:时间触发通信

-第5部分:低功耗模式的高速媒体接入单元

-第6部分:具有选择性唤醒功能的高速介质访问单元

注：

1)第2、5、6部分正在修改中。它们将合并为第二部分的新版本。

# 介绍

ISO 11898于1993年首次作为一个文件发布。它涵盖了CAN数据链路层，以及高速物理层。

在审查和重组的ISO 11898系列:

-第1部分定义了数据链路层，包括逻辑链路控制(LLC)子层和媒体访问控制(MAC)子层，以及物理信令(PHS)子层;-第2部分定义了高速物理介质连接(PMA);

-第3部分定义了低速容错物理介质连接(PMA);

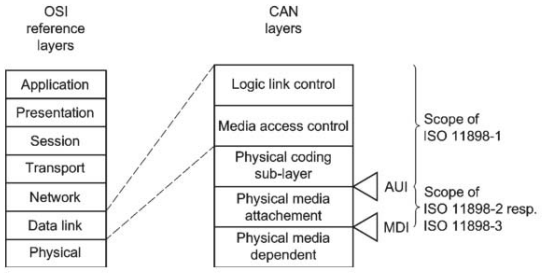
-第4部分定义了时间触发通信;

-第5部分定义了高速物理介质连接(PMA)的电源模式;

-第6部分定义了高速物理介质附件(PMA)的选择性唤醒功能。

注:ISO 11898-2与ISO 11898的这部分更新同步更新，以合并ISO 11898-2、ISO 11898-5和ISO 11898-6中描述的功能。(ISO 11898-2的未来版本将取消并取代当前的ISO 11898-2:2003、ISO 11898-5:2007和ISO 11898-6:20 03)

图1显示了OSI参考层与ISO 11898系列各部分之间的关系。



说明ISO 11898-2是指将现行ISO 11898-2 - 2003、ISO 11898-5 - 2007和ISO 11898-6 - 2013取消并替换的未来版本。

图1-CAN数据链路和物理层与OSI模型的关系

# 道路车辆-控制器区域网络(CAN) - 第1部分: 数据链路层和物理信令

# 1范围

iso 11898的这一部分规定了建立数字信息交换的特征

实现CAN数据链路层的模块之间。控制器局域网是串行的

通信协议，支持分布式实时控制和多路复用，用于道路车辆和其他控制应用。

ISO 11898的这一部分规定了经典的CAN帧格式和新引入的CAN灵活帧格式

数据速率帧格式。经典的CAN帧格式允许高达1mbit /s的比特率和有效载荷

每帧最多8字节。灵活的数据帧格式允许比特率高于1mbit /s和

有效载荷大于8字节每帧。

ISO 11898的这一部分描述了CAN的层次结构

根据ISO/IEC7498-1的开放系统互连(OSI)参考模型。CAN链路层按照ISO/IEC 8802-2和ISO/IEC 8802-3规范进行配置。

ISO 11898的这一部分包含以下详细规范(见图2):

- 逻辑链路控制子层;

- 介质访问控制子层;

- 物理编码子层。

有三个实现选项。它们如下所示。

- 支持经典的CAN帧格式，不支持灵活的数据速率帧格式;

- 支持经典的CAN帧格式，并容忍灵活的数据速率帧格式;

- 支持经典的CAN帧格式和灵活的数据速率帧格式。

建议在新的设计中实现最后一个选项。

注:仅在不使用灵活数据速率帧格式时，第一选项的实现可以与第三选项的实现通信;否则，将生成错误帧。在使用灵活的数据速率帧格式的CAN网络中也有实现第一种选择的机会，但这些不在ISO 11898这一部分的范围内。

# 2一致性

数据链路层一致性测试计划不在ISO 11898这一部分的范围内。为了实现符合ISO 11898的这一部分，逻辑链路控制子层和媒体访问控制子层应符合ISO 11898的这一部分给出的所有强制性规范和值。如果实现了可选的规范和值，它们也应遵守。

# 3规范参考

以下文件，全部或部分，在本文件中作为规范参考，是本文件应用中不可缺少的。对于有日期的引用，仅引用的版本适用。对于未注明日期的引用，适用最新版本的引用文件(包括任何修订)。

ISO/IEC 7498-1, Information technology—Open Systems Interconnection — Basic Reference Model: The Basic Model— Part1

ISO/IEC/IEEE 8802-3:2014,Standard for Ethernet— Part3

# 4术语和定义

在is011898的这一部分中，下列术语和定义适用。

## 4.1仲裁阶段 - arbitration phase

使用标称位时间的阶段

## 4.2 bit填充 - bit stuffing

当使用NRZ bit表示时，提供周期性重同步所需的总线状态变化的帧编码方法

注1:每当传输逻辑遇到数据中一定数量 (填充宽度) 的连续相等的bit时，它会自动地将一些互补的值(一个比特位)填充到输出比特流中。。接收器对数据帧和远程帧进行处理，即执行反向过程。

## 4.3总线 - bus

通信网络的拓扑结构，其中所有节点都通过被动链路到达，允许双向传输

## 4.4总线比较器 - bus comparator

将物理信号通过通信介质转换回逻辑信息或数据信号的电子电路

## 4.5总线驱动 - bus driver

把信息或数据信号转换成物理信号的电子线路，以便这些信号能在通信媒介上传输

## 4.6总线状态 - bus state

两种互补状态之一:显性隐性

注1:显性状态表示逻辑0，隐性状态表示逻辑1。在显性比特和隐性比特同时传输时，得到的总线状态是显性的。当没有传输时，总线是空闲的。空闲时，处于隐性状态

## 4.7经典基帧格式 - Classical Base Frame Format

数据帧或使用11位标识符的远程帧的格式，以一个比特速率传输，最多包括8个数据字节

## 4.8经典扩展帧格式 - Classical Extended Frame Format

使用29位标识符的数据帧或远程帧的格式，以一个比特速率传输，最多包括8个数据字节

## 4.9经典帧 - Classical Frame

数据帧或远程帧使用经典的基本帧格式或经典的扩展帧格式

## 4.10基于内容的仲裁 - content-based arbitration

当多个节点同时访问总线时，解决总线竞争的CSMA仲裁程序

## 4.11数据比特率 - data bit rate

在数据阶段的比特数时间，独立于比特编码/解码

## 4.12数据位时间 - data bit time

数据阶段一个比特的持续时间

## 4.13数据帧 - data Frame

包含用户数据的帧(例如:一个或多个信号或一个或多个可疑参数的一个或多个进程数据)

## 4.14数据段 - data phase

使用数据位时间的段

## 4.15边缘 - edge

两个连续时间量子之间总线状态的差异

## 4.16错误帧 - error Frame

表示检测到错误条件的帧

## 4.17 FD使能 - FD enabled

能够接收和发送FD帧，以及经典帧

## 4.18 FD基本帧格式 - FD Base Frame Format

使用11位标识符的数据帧格式，以灵活的比特率传输，最多可包含64个数据字节

## 4.19 FD扩展帧格式 - FD Extended Frame Format

使用29位标识符的数据帧格式，以灵活的比特率传输，最多包括64个数据字节

## 4.20 FD帧 - FD Frame

数据帧使用FD基本帧格式或FD扩展帧格式

## 4.21 FD不兼容 - FD intolerant

只能接收或发送经典帧，干扰FD帧

## 4.22 FD兼容 - FD tolerant

不能接收或传输FD帧，但不干扰它们

## 4.23帧 - frame

协议数据链路层的数据单元，规定在传输介质的序列中位或位域的排列和含义

## 4.24 handle

硬件对象一个或多个LLC帧的标签(LPDU)

## 4.25高层协议 - higher-layer protocol

数据链路层之上的协议根据开放系统互连模型的协议[来源:ISO/IEC7498-1]

## 4.26标识符 - identifier

不表明帧的目标，但反映了特定帧的优先级，并表示数据的含义

## 4.27闲置 - idle

网络状态，当有隐性状态时，完成一帧

## 4.28空闲状态 - idle condition

检测11个连续的采样隐性比特序列

## 4.29集成 - integrating

在总线断开恢复或协议异常事件中，节点启动协议操作后等待空闲状态

## 4.30最小时间量 - minimum time quantum

可为特定实现配置的最小时间量

## 4.31多播 - multicast

寻址，即单个帧同时寻址到一组节点

注1: Broadcast是组播的一种特殊情况，即同时向所有节点发送一个帧。

## 4.32多主 - multi master

具有多个节点的网络，其中每个节点都能够临时控制其他节点的行为

## 4.33节点 - node

连接到通信网络的程序集，能够根据通信协议规范跨网络通信

注1: CAN节点是通过CAN网络通信的阳极节点。

## 4.34节点时钟 - node clock

time base用来协调CAN实现中与位时间相关的状态机

## 4.35标称比特率 - nominal bit rate

仲裁阶段每次的比特数，与比特编码/解码无关

## 4.36标称位时间 - nominal bit time

仲裁阶段一个比特的持续时间

## 4.37不归零制 - Non-Return-to-Zero

表示二进制信号的方法，即在同一比特时间内，信号电平不变，其中具有相同逻辑值的比特流不提供边

## 4.38过载帧 - Overload Frame

指示过载情况的帧

## 4.39优先 - priority

属性在仲裁期间控制其排名

注1:高优先级增加了帧赢得仲裁过程的概率。

## 4.40协议 - protocol

节点间信息交换的正式约定或规则集，包括帧管理、帧传输和PL的规范

## 4.41协议异常事件 - protocol exception event

例外的正式约定或规则集能够容忍未来新的帧格式

## 4.42接收方 - receiver

总线未空闲时，非发射机或集成的任何节点

## 4.43远程帧 - remote Frame

请求传输专用数据帧的帧

## 4.44填充位计数 - stuff bit count

CRC字段之前的帧中的内容位的数量，不包括固定的内容位

## 4.45时间触发通信 - time-triggered communication

选项，帧可以在定义的时隙中传输，也提供了网络范围内的时钟同步，以及禁用自动重传帧，以便专用数据帧和远程帧避免与其他节点传输的数据帧和远程帧冲突

## 4.46收发器 - transceiver

将CAN节点连接到CAN网络的电子电路，由总线比较器和总线驱动程序组成

## 4.47发射机 - transmitter

发起数据帧或远程帧的节点，在总线再次空闲或节点失去仲裁之前一直处于发送端

# 5符号和缩写术语

|  |  |  |
| --- | --- | --- |
| **缩写** | **术语** | **备注** |
| **ACK** | Acknowledgement | 确认 |
| **AUI** | Attachment Unit Interface | 附件接口 |
| **BCH** | Bose-Chaudhuri-Hocquenghem | Bose-Chaudhuri-Hocquenghem |
| **BRS** | Bit Rate Switch | 比特率开关 |
| **CAN** | Controller area network | 控制器局域网 |
| **CBFF** | Classical Base Frame Format | 经典基帧格式 |
| **CEFF** | Classical Extended Frame Format | 经典扩展帧格式 |
| **CRC** | Cyclic RedundancyCheck | 循环RedundancyCheck |
| **CSMA** | Carrier Sense Multiple Access | 载波侦听多路存取 |
| **DF** | Data Frame | 数据帧 |
| **DLC** | Data Length Code | 数据长度码 |
| **DLL** | Data Link Layer | 数据链路层 |
| **EF** | Error Frame | 错误帧 |
| **EOF** | End Of Frame | 帧结束 |
| **ESI** | ErrorState Indicator | ErrorState指示器 |
| **FBFF** | FD Base Frame Format | FD基帧格式 |
| **FCE** | Fault Confinement Entity | 故障限制实体 |
| **FD** | Flexible Data Rate | 灵活数据率 |
| **FDF** | FD Format indicator | FD格式指示器 |
| **FEFF** | FD Extended Frame Format | FD扩展帧格式 |
| **HLP** | Higher-Layer Protocols | 更高级别的协议 |
| **IDE** | IDentifier Extension | 标识符扩展 |
| **IPT** | Information Processing Time | 信息处理时间 |
| **LAN** | Local Area Network | 局域网 |
| **LLC** | Logical Link Control | 逻辑链路控制 |
| **LME** | Layer Management Entity | 层管理实体 |
| **LPDU** | LLC Protocol Data Unit | LLC协议数据单元 |
| **LSDU** | LLC Service Data Unit | LLC服务数据单元 |
| **MAC** | Medium Access Control | 介质访问控制 |
| **MAU** | Medium Attachment Unit | 介质连接装置 |
| **MDI** | Medium Dependent Interface | 介质相关界面 |
| **MPDU** | MAC Protocol Data Unit | MAC协议数据单元 |
| **MSB** | Most Significant Bit | 最高有效位 |
| **MSDU** | MAC Service Data Unit | MAC业务数据单元 |
| **NRZ** | Non-Return-to-Zero | 不归零制 |
| **OF** | Overload Frame | 过载帧 |
| **OSI** | Open Systems Interconnection | 开放系统互连 |
| **OVLD** | Overload | 过载 |
| **PCI** | Protocol Control Information | 协议控制信息 |
| **PCS** | Physical CodingSub-layer | 物理CodingSub-layer |
| **PDU** | Protocol Data Unit | 协议数据单元 |
| **PL** | Physical Layer | 物理层 |
| **PMA** | Physical Medium Attachment | 物理介质连接 |
| **r0** | Reserved bit in Classical Extended Frame Format | 经典扩展帧格式中的保留位 |
| **res** | Reserved bit in FD Frames | FD帧中的预留比特位 |
| **RF** | Remote Frame | 远程帧 |
| **RRS** | Remote Request Substitution | 远程请求替换 |
| **RTR** | Remote Transmission Request | 远程传输请求 |
| **SAP** | Service Access Point | 业务接入点 |
| **SDU** | Service Data Unit | 业务数据单元 |
| **SJW** | Synchronization Jump Width | 同步跳变宽度 |
| **SOF** | Start Of Frame | 帧的起始 |
| **SP** | Sample Point | 样本点 |
| **SRR** | Substitute Remote Request | 替换远程请求 |
| **SSP** | Secondary Sample Point | 二次采样点 |
| **TDC** | Transmitter Delay Compensation | 发射机延迟补偿 |

# 6 CAN的基本概念

## 6.1 CAN属性

CAN具有以下属性。

- 基于优先级的多主总线访问;

- 基于内容的非破坏性仲裁;

- 所有帧传输都以广播的形式进行;

- 通过接受过滤进行多播帧传输;

- 远程数据请求;

- 配置的灵活性;

- 全网数据一致性;

- 错误检测和错误信号;

- 自动重传仲裁失败、未被确认或在传输过程中因错误而销毁的帧;

- 区分节点的临时错误和永久故障，以及缺陷节点的自主切换。

## 6.2帧

总线上的信息以固定格式的帧发送，帧的长度不同但有限制。当总线空闲时，允许任何连接的节点开始发送DF或RF。在没有帧传输时，总线空闲。此外，任何连接节点都可以通过专用帧(EF和of)启动错误和过载信息的指示。

## 6.3总线接入方式

如果两个或多个节点同时开始传输DFs或RFs，总线访问冲突通过使用该标识符进行基于内容的仲裁解决。仲裁机制确保信息和时间都不会丢失。具有最高优先级的DF或RF的发射机获得总线接入。与RF ID相同的DF将赢得总线仲裁。

## 6.4信息路由

节点不使用任何有关网络配置的信息(例如节点地址)。相反，接收方根据帧接受过滤(frame acceptance filtering)来决定接收到的信息是否相关。接收者不需要知道信息的发送者，反之亦然。

## 6.5网络灵活性

在can网络中，只要所增加的节点不是任何DF的发送节点，并且不需要任何额外的传输数据，就可以在不改变任何节点软硬件的情况下增加节点。

## 6.6数据一致性

在CAN网络中，一个帧可以同时被所有节点接受，也可以不被任何节点接受。因此，数据一致性是CAN网络的属性，通过广播和错误处理的概念来实现。

## 6.7远程数据请求

通过发送RF，需要数据的节点可以请求另一个节点发送相应的DF。rf和对应的DF使用相同的标识符命名。

**注意1**拥有请求ID的消息的节点决定是否产生新数据或发送数据到传输缓冲区。

**注意2**拥有被请求数据消息的节点决定如何响应DLC不匹配的RF。

## 6.8错误检测

为了检测错误，提供了以下措施:

- 监控(发射器将传输的比特级别与网络上检测到的比特级别进行比较);

- 经典帧的15位CRC, FD帧的17位CRC, FD帧的20到64位数据字段字节;

- 检查FD帧的数据位计数;

- 可变位填充，填充宽度为5 (FD帧的CRC字段除外);

- 帧格式检查;

- ACK。

## 6.9错误信令与恢复时间

损坏的帧由任何发送节点和任何正常工作(错误活动)的接收节点标记。这样的帧会被中止，并根据实现的恢复过程重传(参见8.3.4)。如果没有进一步的错误，从检测错误到可能开始下一帧的恢复时间通常是17到23标称比特次数(在错误被动模式的节点的情况下，最多可达31标称比特次数)。

## 6.10 ACK

所有接收端对接收到的DFs和RFs进行一致性检查，确认一致的帧，并通过EF标记不一致的帧。发送节点认为未被确认为损坏的DF或RF。

## 6.11自动重传

丢仲裁的帧、没有得到确认的帧、传输过程中被错误干扰的帧都会自动重传，直到传输成功完成或不再请求传输为止(见8.3.4和10.9.6)。可选地，自动重传可以被禁用(参见9.3)。可选地，自动重传可以限制为一定的尝试次数(参见10.9.6)。

## 6.12故障限制

CAN节点能够区分短暂的干扰和永久的故障。故障的发射节点被关闭。“关闭”意味着节点逻辑上与总线断开，既不能发送也不能接收任何帧(参见12.1.4.4)。

## 6.13 Error-active

错误活动节点通常参与总线通信，并在检测到错误时发送主动错误标志。主动错误标志由6个连续的显性位组成，违反了比特填充规则和DF和RF中出现的所有固定格式(见12.1.4.2)。

## 6.14 Error-passive

错误被动节点不发送主动错误标志。它参与总线通信，但当检测到错误时，就发送被动错误标志。被动错误标志由6个连续的隐性位组成。传输后，错误被动节点在发起进一步传输之前会等待一段额外的时间(参见10.4.6.4和12.1.4.2中的暂停传输)。

## 6.15 Bus-off

当一个节点由于FCE请求从总线断开时，它处于总线断开状态。在bus-off状态下，节点既不发送也不接收帧。在总线断开状态下，节点不发送任何显性比特。

# 7 CAN的分层架构

## 7.1 OSI模型参考

根据OSI参考模型(见ISO/IEC 7498-1)， IS0 11898这部分的CAN结构有两层(见图2)，

- DLL和

- PL的PCS。

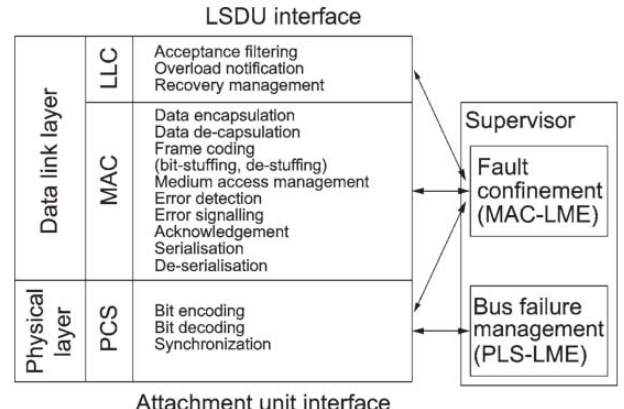


图2 - CAN的分层结构

根据ISO/IEC 8802-2和ISO/IEC 8802-3, DLL又细分为

- LLC 和

- MAC

PL被细分为

- PCS,

- PMA，还有

- MDI。

MAC子层的操作由FCE监督。故障限制是一种自检机制，用于区分短时干扰和永久故障(见12.1)。

PL也可以由一个实体监督，该实体负责检测和管理物理介质的故障(例如总线线路短路或中断，参见12.2)。

## 7.2协议规范

两个对等协议实体通过交换帧或pdu进行通信。

(N)层协议数据单元(PDUN)由(N)层特定协议控制信息(PCIN)和(N)层用户数据组成。PDUN应该通过SAPN-1传递到(N-1)层实体。PDUN应通过SAPN-1传递到(N-1)层，该层的业务允许PDUN的传递。SDU是(N)层实体之间保持身份的接口数据，即表示业务所传输的逻辑数据单元。CAN协议的DLL既不能提供将一个SDU映射到多个PDUs的方法，也不能提供将多个SDUs映射到一个PDU的方法，即直接从关联的SDUN和特定层的控制信息PCIN构建一个PDUN。图3说明了数据链路子层的交互。

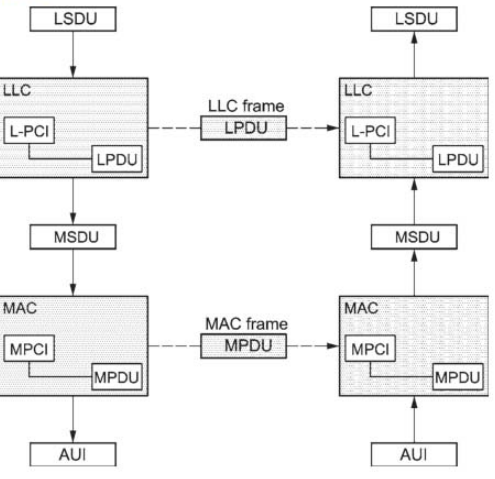


图3 -协议层交互

## 7.3服务格式说明

### 7.3.1业务原语的格式描述

服务要素应写成:

service.type (

[parameterl, ...]

）

where

Service 服务名称。L\_Data用于LLC子层提供的数据传输服务;

Type 表示服务原语的类型(参见Z.3.2);

[parameterl, ...] 是传递给服务原语的值列表。

方括号表示这个参数列表可能为空。

### 7.3.2服务原语的类型

服务原语应为三种通用类型。

**a)Service.Request**

请求原语应从(N)用户(服务用户)传递到(N)层(服务提供者)，以请求启动服务。

**b)Service.Indication**

指示原语应从(N)层传递给(N)用户，以指示对(N)用户有意义的内部(N)层(或子层)事件。该事件可能与远程服务请求在逻辑上相关，也可能是由(N)层(或子层)内部的事件引起的。

**c)Service.Confirm**

确认原语应从(N)层(或子层)传递给(N)用户，以传递一个或多个先前相关服务请求的结果。这个原语可以表示不遵守或某种程度的遵守。它不一定表明远程对等接口上的任何活动。

## 7.4 LLC接口

LLC子层应为LLC用户提供两种无连接传输服务:

- 未确认的数据传输服务;

- 未确认的远程数据请求服务。

从用户发送或发送给用户的接口业务数据应按8.2.2中规定。表1和表2指定了LLC用户和LLC子层之间发送的消息。

表1 - LLC用户发送给LLC子层的消息

|  |  |
| --- | --- |
| 信息 | 描述 |
| Reset\_Request | 将结点设置为初始状态的请求 |

表2 -从LLC发送的消息子层发送给LLC用户

|  |  |
| --- | --- |
| 信息 | 描述 |
| Reset\_Response | 对Reset\_Request的响应 |
| Node\_Status | 表示节点的当前状态，即它指示节点是否处于bus-off状态。 |

从主管FCE发送和发送到主管FCE的LLC接口消息应按照12.1.3的规定

# 8子层描述

## 8.1通用

LLC子层根据ISO/IEC8802-2描述了动态链接库的上层。它与那些与介质访问方法类型无关的协议问题有关。

## 8.2 LLC子层业务

### 8.2.1无连接模式传输业务类型

LLC子层应提供两种类型的无连接模式传输服务:

**未确认的数据传输服务**

本服务将提供LLC用户交换LSDU而无需建立数据链路连接的方法。数据传输可以是点对点，组播或广播。

**未确认的远程数据请求服务**

该服务将提供LLC用户在不建立数据链路连接的情况下请求远程节点进行LSDU传输的方法。

远程节点主要通过以下两种方式为数据请求提供服务。

- 请求的数据可能由远程用户准备传输。在这种情况下，数据应位于远程节点缓冲区中，并由远程用户LLC实体在接收到远程请求帧后进行传输。

- 远程用户收到远程请求帧后，发送请求数据。

根据两种不同的LLC业务，发送节点和接收节点之间的通信可以使用六种类型的帧。

- LLC数据帧经典的基础帧格式;

- 经典扩展帧格式LLC数据帧;

- FD基本帧格式LLC数据帧;

- FD扩展帧格式LLC数据帧;

- LLC远程帧经典基帧格式;

- LLC远程帧经典扩展帧格式。

LLC DFs应将数据从发射机传送到接收机。LLC RFs应被传输以请求从(单个)远程节点传输DF(具有相同标识符)。在这两种情况下，LLC子层应通知LLC用户成功发送或接收了DF或RF。

### 8.2.2业务基本规格

#### 8.2.2.1通用

本子句的服务原语规范详细描述了LLC服务原语及其相关参数。LLC服务原语的完整列表如表3所示。

表3 - LLC服务原语概述

|  |  |  |
| --- | --- | --- |
| 服务 | 服务原语 | 描述 |
| 未经确认的数据传输 | L\_Data.Request | 数据传输请求 |
| L Data.Indication | 数据传输指示 |
| L\_Data.Confirm | 确认数据传输 |
| L\_Data.AbortRequest (optional) | 请求终止数据传输 |
| 未确认的远程数据请求 | L\_Remote.Request | 请求远程数据请求 |
| L\_Remote.Indication | 远程数据请求的指示 |
| L\_Remote.Confirm | 确认远程数据请求 |
| L\_Remote.AbortRequest(optional) | 请求终止远程数据请求 |

表4 - LLC服务原语参数列表

|  |  |
| --- | --- |
| 范围 | 描述 |
| Identifier | 标识帧的内容 |
| Format | 帧格式(CBFF,CEFF,FBFF,FEFE，可选给出ESI和BRS位值) |
| DLC | 数据长度 |
| Data | 用户想要传输的数据 |
| Transfer\_Status | 确认参数 |
| Handle (optional) | 标识用于事务处理的硬件元素 |

#### 8.2.2.2 L\_Data.Request

**Function**

L\_Data.Request原语应从LLC用户传递到LLC子层，以请求将一个LSDU发送到一个或多个远程LLC实体。

**L\_Data.Request源语语义**

原语应提供如下参数:

L\_Data.Request(

Identifier

Format

DLC

Data

Handle

)

如果关联的LLC DF的数据长度为零，则参数数据不重要。可选地，用于传输的硬件元素(消息存储单元)由句柄标识。

**Effect on receipt**

接收此原语将导致LLC子层使用MAC子层提供的数据传输服务发起LLC DF的传输(见表6)。如果在此期间没有出现错误帧，则请求应不迟于请求后的第二次SOF处理。

#### 8.2.2.3 L\_Data.indication

**Function**

L\_Data.indication原语应从LLC子层传递给LLC用户，以表明LSDU的到达。

**L\_Data.indication源语语义**

原语应提供如下参数:

L\_Data.Indication(

Identifier Format

DLC

Data

)

如果关联的LLC DF的数据长度为零，则参数数据不重要。

**Effect on receipt**

LLC用户接收此原语的效果未指定。

#### 8.2.2.4 L\_Data.Confirm

**Function**

L\_Data.Confirm原语应从本地LLC子层传递到LLC用户，以传递上一个L\_Data的结果。原始的请求。该原语应为当地确认，即。这并不意味着远程有限责任公司实体或实体已将相关指示原语传递给相应的有限责任公司用户。

**L\_Data.Confirm源语语义**

原语应提供如下参数:

L\_Data.Confirm(

Identifier

Transfer\_Status Handle

)

Transfer\_Status应使用表示交易状态的完成

- 由之前的L\_Data.Request原语发起(如果可选句柄不存在)，或

- 由Handle给出的可选引用硬件元素的。

Transfer\_Status:[已完成，未完成，终止]

如果提供并支持可选服务L\_Data.AbortRequest ，则传输状态为Aborted。

**Effect on receipt**

LLC用户接收此原语的效果未指定。

#### 8.2.2.5 L\_Data.AbortRequest (optional)

**Function**

可选的L\_Data.AbortRequest原语将从LLC用户传递到LLC子层，以终止之前请求的LSDU传输请求。

**L\_Data.AbortRequest源语语义**

原语应提供如下参数:

L\_Data.AbortRequest(

Handle

)

通过句柄，识别出传输应中止的硬件元件(消息存储单元)。

**Effect on receipt**

接收此原语将导致LLC子层中止在指定的消息存储单元中传输LLC DF。待决的传输(已经传递到MAC子层)只有在以下情况下才会终止

- 传输过程中MAC子层错误 或

- 仲裁丢失在MAC子层，

这会导致LSDU等待另一次传输尝试。

这意味着终止请求必须在LLC层保持待决状态，直到上述任意一种情况发生，或直到传输完成。

LLC子层不能立即中止已经提交到MAC子层的传输(由于由Handle指示的请求硬件元素的优先调度)。如果在此期间没有出现错误帧，则任何L\_Data.AbortRequest应在请求后的第二次SOF之前进行处理。在此期间出现错误帧时，任何L\_Data.AbortRequest应在请求后的第三次SOF之前进行处理。

#### 8.2.2.6 L\_Remote.Request

**Function**

L\_Remote.Request原语应从LLC用户传递到LLC子层，以请求单个远程LLC实体传输一个LSDU。

**L\_Remote.Request源语语义**

原语应提供如下参数:

L\_Remote.Request(

Identifier Format

DLC

Handle

)

DLC的值等于所请求DE的数据字段的长度。通过可选句柄，确定用于传输的硬件元素(消息存储单元)。

**Effect on receipt**

收到此原语后，LLC子层将使用由MAC子层提供的远程数据传输服务，在可选指定的消息存储单元中启动一个LSDU的传输(见表6)。

#### 8.2.2.7 L\_Remote.Indication

**Function**

L\_Remote.Indication原语应从LLC子层传递给LLC用户，以表明传输LSDU请求的到达。

**L\_Remote.Indication源语语义**

原语应提供如下参数:

L\_Remote.Indication(

Identifier Format

DLC

)

identifier将标识要发送的LSDU。DLC的值等于所请求DF的数据字段的长度。

**Effect on receipt**

LLC用户接收此原语的效果未指定。

#### 8.2.2.8 L\_Remote.Confirm

**Function**

L\_Remote.Confirm原语应从本地LLC子层传递给LLC用户，以传达上一个L\_Remote.Request原语的结果。此原语应为本地确认，这并不意味着远程LLC实体已经将关联的指示原语传递给相应的LLC用户。

**L\_Remote.Confirm源语语义**

原语应提供如下参数:

L\_Remote.Confirm(

Identifier

Transfer\_Status

Handle

)

Transfer\_Status 应表明事务状态

- 先前的L\_Remote.Request原语(如果可选句柄不存在)，或者

- 由Handle给出的可选引用硬件元素的。

Transfer\_Status应表明前一个L\_Remote.Request原语发起的事务完成。

Transfer\_Status:[已完成，未完成，终止]

在传输请求仍在等待中或正在进行中，或一个可选的中止请求正在等待中时，应给出传输状态Not\_Complete。

如果提供且支持可选服务L\_Remote.AbortRequest，则传输状态为“已终止”。。

**Effect on receipt**

LLC用户接收此原语的效果未指定。

#### 8.2.2.9 L\_Remote.AbortRequest (optional)

**Function**

可选的L\_Remote.AbortRequest原语应从LLC用户传递到LLC子层，以中止向单个远程LLC实体发送LSDU(即RF)的请求，该请求之前已经请求过。

**L\_Remote.AbortRequest源语语义**

原语应提供如下参数:

L\_Remote.AbortRequest(

Handle

)

通过句柄，识别出传输应中止的硬件元件(消息存储单元)。

**Effect on receipt**

接收此原语将导致LLC子层中止在指定的消息存储单元中传输LLC RF。挂起的传输(已经传递到MAC子层)只能在以下情况下中止

- 传输过程中MAC子层错误或

- 仲裁丢失在MAC子层，

这会导致LSDU等待另一次传输尝试。

这意味着终止请求必须在LLC层保持待决状态，直到上述情况之一发生，或直到传输完成。

LLC子层不能立即中止已经提交到MAC子层的传输(这是由句柄指示的所请求硬件元素的优先级调度造成的)。

## 8.3 LLC子层功能

### 8.3.1通用

LLC子层应提供以下功能:

- 帧接受过滤;

- 超载通知;

- 恢复管理。

### 8.3.2 Frame acceptance filtering - 帧接受过滤

在LLC子层发起的帧交易应该是独立于先前帧交易的独立的、独立的操作。帧的内容应当由其标识符命名。每个接收方通过帧接收滤波来判断帧是否相关。

### 8.3.3 Overload notification - 超载通知

如果接收端内部条件要求下一个LLC DF或LLC RF延迟，则由LLC子层发起MAC的传输。如果实施CAN的内部条件导致MAC的启动，则这些条件应被记录在案。

最多可以产生两个MAC，以延迟下一个DF或RF。

### 8.3.4 Recovery management - 恢复管理

LLC子层应提供自动重传丢失仲裁的帧、未被确认的帧或在传输过程中被错误干扰的帧的方法。然而，自动重传可以通过可选的LLC AbortRequests终止。在传输成功完成或(可选)中止之前，不得向用户报告已确认或(可选)中止的帧传输服务。当不再请求传输该帧时，应禁用该帧的自动重传。所有帧的自动重传可能被禁用。

## 8.4 LLC帧的结构

### 8.4.1通用

LLC帧应该是对等LLC实体(LPDU)之间交换的数据单元。LLC DF和RF的结构和格式应随后指定。LLC帧的可选句柄保存在LLC本身内，作为与上层通信使用的标签。因此，它的表示在MAC层上是不可见的，因此在CAN总线上也是不可见的。下面不再考虑LLC帧句柄。

### 8.4.2 LLC DF规范

#### 8.4.2.1通用

LLC DF由四个位域组成(见图4):

- identifier field;

- format field;

- DLC field;

- LLC data field.



**图4-LLC DF**

#### 8.4.2.2 Identifier field - 标识字段

标识符字段应该由两个部分组成，基本标识符和标识符扩展。

基本标识符长度为11位(ID-28 ~ ID-18)，扩展标识符长度为18位(ID-17 ~ ID-0)。CBFF和FBFF中的帧应该忽略标识符扩展。

#### 8.4.2.3 Format field - 格式字段

该字段用于区分CBFF、CEFF、FBFF和FEFE中的帧。在FBFF和FEFF中，包括ESI位和BRS位(见表4)。

#### 8.4.2.4 DLC field - DLC字段

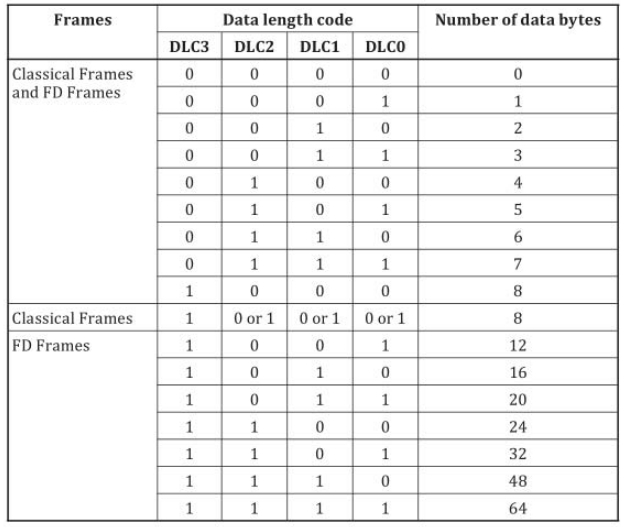
数据字段中的字节数应由DLC表示;见表5。这个DLC应该由4位组成。

经典帧允许的数据字节数应该在0到8之间。0 ~ 7的dlc表示长度为0 ~ 7字节的字段。

在经典的帧中，所有其他的dlc应该表明数据字段是8字节长。

在FD帧中，0 ~ 8的dlc表示长度为0 ~ 8字节的数据字段，其他的DLC编码长度更长的数据字段，如表5所示。

**表5 DLC数据字节数编码说明**



#### 8.4.2.5 Data field - 数据字段

数据字段应包括在DF内传输的数据。它可以在经典帧中包含0字节到8字节，也可以在FD帧中包含0字节到64字节，其中每个字节应包含8位。

### 8.4.3 LLC RF规范

LLC RF应由三个位域组成(见图5):

- identifierfield;

- format field;

- DLC field.



**图5-LLC RF**

LLC RF identifier字段和DLC字段的格式与LLC DF identifier字段(见8.4.2.2)和DLC字段(见8.4.2.4)的格式相同。没有数据字段，与DLC的值无关。LLC RF的格式字段只区分CBFF和CEFF格式的帧。在FBFF或FEFF中不得有LLC RF

RFs只能通过一个全网确定的DLC进行传输，这个DLC是对应DF的DLC(见10.9.9)。

## 8.5 LLC帧限制

不需要实现所有可能的标识符或dlc。

如果LLC子层被限制使用一个子范围的标识符(例如，只有11位标识符)，那么它应该被限制使用该子范围的标识符的LLC DFs和LLC RFs(例如:标识符的IDE位设置为逻辑0，忽略其标识符扩展)。

如果一个LLC子层被限制使用小于最大数据字节数，那么它应该被限制到LLC DFs与该限制范围的数据字节数。如果DLC表示更高的数据字节数，则超出限制范围的数据字节将在LLC DF中替换为CCHEX(“padding”)值的字节，无论是接收帧还是传输帧。可选地，CAN实现可以支持这样一种配置，即当帧的DLC指示超出限制范围的数据字节数时，CAN节点将不会传输帧。

**注意** 接收到的消息填充不需要在LLC层实现。

# 9 LLC和MAC接口

## 9.1服务

MAC子层应向本地LLC提供服务

- (MAC-) LLC帧的确认传输，以及

- MAC OFs传输。

接口业务数据从LLC子层发出或发送至LLC的信息应如8.3所述。

## 9.2时间和时间触发

### 9.2.1描述

可选地，启用了CAN的实现可以支持全网时钟同步，如果是这样，它们还可以额外支持时间触发通信。时钟同步选项描述了校准网络中节点时钟所需的先决条件。为了使网络内节点的时钟同步，需要一个公共参考点。以任意消息应使用的SOF位或EOF的最后一位的采样点作为参考点。节点时钟同步有助于在高层协议中建立全网时间基础。如果实现，时间触发通信选项将促进帧在定义的时隙中传输。

建立全网络时基所需的硬件应包括在LLC和MAC之间。

### 9.2.2时基

任何支持时间和时间触发选项的节点都应提供一个时间基准。时基是一个至少16位的循环向上计数器，以由内部或外部时钟发生器产生的时钟时标作为反馈。

### 9.2.3时间参考点

接收或传输的任何信息应调用在SOF识别各自信息时或在EOF最后位采样点时获取的时基。成功接收消息后，捕获的值应至少提供给CPU一个消息，并且在接收到下一个消息之前应是可读的。

### 9.2.4事件产生

应能够从上述时基产生至少一个可编程事件触发器。该触发器应由CPU自由编程，范围至少为0到(216-1)×时间滴答。

## 9.3禁用自动重传功能

### 9.3.1重传帧

可以禁用自动重传(参见6.11)。

## 9.4消息时间戳

可选地，是否启用FD实现可以为接收和传输的数据帧支持消息时间戳功能。

时间戳的宽度应为8位、16位或32位。用于时基的时钟源应在节点内部生成或由LLC用户提供。时基计数器应增加，并应超过零。

LLC用户在任何时候都可以读取时间基准计数器。时间基准值应在每个数据框的参考点上获取。对于经典帧，参考点是对应帧的SOF采样点或根据10.7取帧有效时的时间点。对于FD帧，即当FDF为隐性且后续res位为显性时，参考点为SOF的采样点，或res位的采样点，或10.7中认为该帧有效的时间点。

在EOF之后，捕获的时间戳值将被LLC用户读取。

# 10 MAC子层描述

## 10.1通用

MAC子层表示OSI DLL的下部。它应服务于LLC子层和PL的接口，并包含相关的功能和规则

- 发送/接收数据的封装/解封装，

- 错误检测和信号

- 管理传输/接收媒体访问。

## 10.2 MAC子层业务

### 10.2.1服务描述

由MAC子层提供的服务应允许本地LLC子层实体与对等LLC子层实体交换MSDU。MAC子层服务如下:

a)确认数据传输

本服务应提供有LLC实体在没有建立数据链路连接的情况下交换MSDUs的方法。数据传输可以是点对点、组播或广播。

b)确认远程数据请求

此服务将提供一种方法，通过该方法LLC实体请求另一个远程节点传输一个LSDU，而无需建立数据链路连接。远程LLC实体应使用MAC服务“确认数据传输”来传输请求的数据。服务的ACK应由远程节点的MAC子层产生。ACK不应包含远程节点用户的任何数据。

c)OF 转移

该服务应提供一种方式，使LLC实体发起传输OF，一种特殊的固定格式LPDU，造成下一个DF或RF的延迟。

### 10.2.2业务原语规范

#### 10.2.2.1通用

提供给LLC子层的MAC子层的服务原语应在表6中给出。

**表6-MAC子层业务原语**

|  |  |
| --- | --- |
| **服务** | **服务原语** |
| 确认数据传输 | MA\_Data.Request  MA\_Data.Indication  MA\_Data.Confirm |
| 确认远程数据请求 | MA\_Remote.Request  MA\_Remote.Indication  MA\_Remote.Confirm |
| OF 转移 | MA\_OVLD.Request  MA\_0VLD.Indication  MA\_0VLD.Confirm |

#### 10.2.2.2 MA\_Data.Request

**Function**

MA\_Data.Request 原语将从LLC子层传递到MAC子层,将MSDU发送到一个或多个远程MAC实体。

**MA\_Data.Request源语语义**

原语应提供如下参数:

MA\_Data.Request(

Identifier

Format

DLC

Data

)

对数据长度为零的MAC DFs来说,参数数据无关紧要。

**Effect on receipt**

接收此原语将导致MAC子层以包括所有MAC特定的控制信息(SOF, SRR bit, IDE bit, RTR (or RRS) bit, FDF bit, res [or r0) bit, BRS bit, ESI bit, CRC, ACK域的隐性位, EOF) 到来自 LLC 子层的MSDU。详见10.4.2.3和10.4.2.4。MPDU应被序列化,并将其传递给PL,以转移到对等MAC的子层实体或实体。

#### 10.2.2.3 MA\_Data.Indication

**Function**

MA\_Data.Indication原语应从MAC子层传递到LLC子层，以表明MSDU的到达。

**MA\_Data.Indication源语语义**

原语应提供如下参数:

MA\_Data.Indication(

Identifier

Format

DLC

Data

)

如果相关的MAC DF是数据长度为零,则参数数据是无关紧要的。只有正确接收到MSDU时，才应向LLC子层指示MSDU的到达。

**Effect on receipt**

由LLC子层对该原始元素的接收效果未说明。

#### 10.2.2.4 MA\_Data.Confirm

**Function**

MA\_Data.Confirm原语应从本地MAC子层传递到LLC子层，以传递上一个MA\_Data.Request原语的结果。此原语是一个远程确认，即它应表明远程MAC实体或实体已将相关的指示原语传递给相应的用户。

**MA\_Data.Confirm源语语义**

原语应提供如下参数:

MA\_Data.Confirm(

Identifier

Transmission\_Status

)

Transmission\_Status 应用来表示前一个MA\_Data的成功或失败。

**请求源语**

Transmission\_Status:[Success, No\_Success]

故障是在传输过程中发生的错误或仲裁失败。

**Effect on receipt**

LLC子层对该原语的效果接收是未指定的。

#### 10.2.2.5 MA\_Remote.Request

**Function**

MA\_Remote.Request 原语应从LLC子层传递到MAC子层，以请求单个远程MAC实体传输MSDU。

**MA\_Remote.Request源语语义**

原语应提供如下参数:

MA\_Remote.Request(

Identifier

Format

DLC

)

标识符应标识要发送的MSDU。DLC的值应等于所请求的MSDU数据的长度。

**Effect on receipt**

接收到该原语后，MAC子层将把所有MAC特定的控制信息(SOF, SRR bit, IDE bit, RTR bit, FDF bit, rO bit, CRC, ACK域的隐性位, EOF)包含到来自LLC子层的MSDU以准备PDU。具体请参见10.4.2.3和10.4.2.4。MPDU应被串行化，并作为SDU逐位传递给PL，以便传输给对等的MAC子层实体或实体。

#### 10.2.2.6 MA\_Remote.Indication

**Function**

MA\_Remote.Indication原语应从MAC子层传递到LLC子层，以表明MSDU传输请求的到达。

**MA\_Remote.Indication源语语义**

原语应提供如下参数:

MA\_Remote.Indication(

Identifier

Format

DLC

)

只有正确接收到MSDU传输请求时，才应向LLC子层指示MSDU传输请求的到达。

**Effect on receipt**

LLC子层对该原语的接收效果未指明。

#### 10.2.2.7 MA\_Remote.Confirm

**Function**

MA\_Remote.Confirm原语将从本地MAC子层传递到LLC子层，以传递上一个MA\_Remote.Request的结果。此原语是一个远程确认，即它应表明远程MAC实体或实体已将相关的指示原语传递给相应的用户。

**MA\_Remote.Confirm源语语义**

原语应提供如下参数:

MA\_Remote.Confirm(

Identifier

Transmission\_Status

)

Transmission\_Status应使用表示前一个MA\_Remote.Request原语的成功或失败。

**Transmission\_Status:[Success, No\_Success]**

故障是在传输过程中发生的错误或仲裁失败。

**Effect on receipt**

LLC子层对该原语的接收效果未指明。

#### 10.2.2.8 MA\_OVLD.Request

**Function**

MA\_OVLD.Request原语从LLC子层传递到MAC子层，请求传输MAC OVLD帧(见10.4.5)。OVLD帧应为固定格式帧，并在MAC子层中完全构建。

**MA\_OVLD.Request源语语义**

原语应提供如下参数:

MA\_OVLD.request(

)

**Effect on receipt**

收到本原语将使MAC子层形成OF。OF应被传递到较低的协议层，以便传递给对等MAC子层实体。

#### 10.2.2.9 MA\_OVLD.Indication

**Function**

MA\_OVLD.Indication原语应从MAC子层传递到LLC子层，以表明已收到OF(见10.4.5)。

**MA\_OVLD.Indication源语语义**

原语应提供如下参数:

MA\_OVLD.Indication(

)

**Effect on receipt**

LLC子层对该原语的接收效果未指明。

#### 10.2.2.10 MA\_OVLD.Confirm

**Function**

MA\_0VLD.Confirm原语应从MAC子层传递到LLC子层，以表明OF已发送。该确认应是本地的，这并不意味着远程对等协议实体已正确接收到OF。

**MA\_0VLD.Confirm源语语义**

原语应提供如下参数:

MA\_OVLD.Confirm(

Transmission\_Status

)

Transmission\_Status应使用表示上一个MA\_OVLD.request原语的成功或失败。

**Transmission\_Status:[Success, No\_Success]**

**Effect on receipt**

LLC子层对该原语的接收效果未指明。

## 10.3 MAC子层体系结构功能模型

### 10.3.1 Capability

MAC子层的功能能力由 ISO/IEC 8802-3 中指定的功能模型描述(参见图6)。在该模型中，MAC子层被分为两个完全独立运行的部分，即:发送部分和接收部分。发送和接收部分的功能应按本条款和图6所示。

### 10.3.2帧传输

帧传输应满足以下要求:

**a)传输数据封装**

- LLC帧和接口控制信息的接受;

- CRC序列计算，包括FD帧的数据位计数;

- 通过向LLC帧添加SOF, SRR位(如果使用帧格式)，IDE位，RTR(或RRS)位，FDF位，res位，BRS位(如果使用帧格式)，ESI位(如果使用帧格式)，CRC, ACK和EOF来构建MAC帧(受限制的LLC子层不能请求传输超出其限制的标识符或数据字段的MAC帧;见8.5)。

**b)传输介质接入管理**

-识别总线空闲(符合帧间空间)后启动传输过程;

-序列化MAC帧;

-插入填充bit(bit stuffing);

-仲裁并在仲裁失败时转入接收模式;

-错误检测(监控、格式检查);

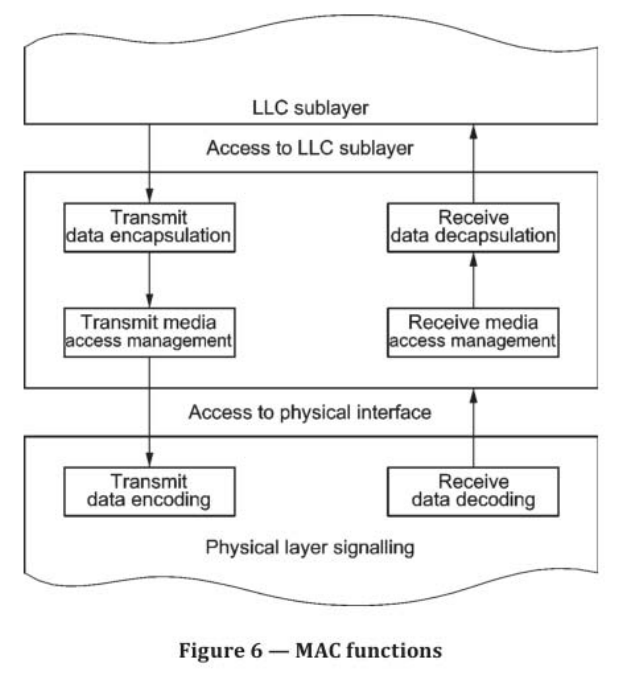
-ACK校验;

-过载情况的识别;

-OF 创建和开始传播;

-EF 创建和开始传播;

-向PL展示用于传输的串行比特流。



### 10.3.3帧接收

帧接收应满足以下要求:

**a)接收媒体接入管理:**

-从PL接收串行比特流;

-帧结构的反序列化和重新编译;

-删除填充bit(bit de-stuffing);

-错误检测(CRC，数据位计数校验，格式校验，数据规则校验);

-发送ACK;

-EF 创建和启动传播;

-超载情况的识别;

-反应 OF 构造和启动传播。

**b)接收数据解封装**

-从接收到的帧中删除MAC相关信息;

-向LLC子层展示LLC帧和接口控制信息(对于受限LLC子层，只展示LLC帧的有限部分;见8.5)。

## 10.4 mac帧的结构

### 10.4.1描述

CAN网络节点间的数据传输和接收由四种不同的帧类型进行控制:

- 将数据从发射器传送到所有接收器的DF;

- 节点发送RF，请求传输具有相同标识符的DF;

- 当检测到总线错误时，由任何节点(发送器或接收器)发送的EF;

- 用于在前面和后面的DFs或RFs之间提供额外延迟的OF。

DFs和RFs应为总线访问进行仲裁，并通过帧间空间与前面的帧分开

在CAN中有四种不同的DFs:

- DF 在 经典的基础帧格式，

- DF 在 经典扩展帧格式，

- DF 在 FD基本帧格式，

- DF 在 FD扩展帧格式。

CAN中有两种不同的RFs:

- 经典基帧格式的RF，

- 经典扩展帧格式的RF。

### 10.4.2 MAC DF规范

#### 10.4.2.1描述

在传输时，LLC DF(见图4)应转换为MAC DF。接收时，MAC DF应转换为LLC DF。MAC DFs由7个不同的位域组成;请参见图7。

- SOF;

- 仲裁字段(包含标识符字段和部分格式字段);

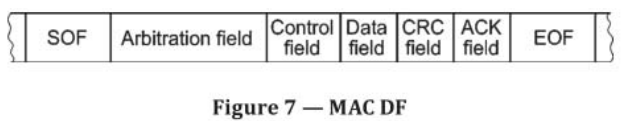
- 控制字段(包含DLC字段和部分格式字段);

- 数据字段(包含LLC数据字段);

- CRC field;

- ACK field;

- EOF.



#### 10.4.2.2 SOF

SOF应标志着DFs和RFs的开始。它应该由一个显性位组成。

只有在总线空闲时，节点才发送SOF(参见10.4.6.3)。在暂停传输期间或在间歇的第三位对显性位进行采样的节点应将其作为SOF接收。

如果一个节点在帧间隔的第3个比特对显性比特进行采样，如果该节点正在等待传输，且该节点是主动错误或已接收上一帧，则该节点应以其标识符的第1个比特从下一个比特开始传输其消息，而不首先传输SOF比特，也不成为接收方。

所有节点应同步到由SOF引起的电平变化沿第一个节点开始传输。

#### 10.4.2.3仲裁字段

仲裁字段应由标识符字段(从LLC子层传递)和RTR位(CBFF 和 CEFF)或RRS位(FBFF 和 FEFF)组成。在一个 MAC DF 中，RTR位和RRS位的值应该是显性的。根据控制域的IDE位和FDF位的不同，仲裁域的结构对于四种格式是不同的。

- 在CBFF (IDE标志为显性的)，仲裁字段应该由11位基本标识符和RTR位组成。标识位用ID-28 ~ ID-18表示。

- 在CEFF (IDE标志为隐性的)，仲裁字段由基本标识符(ID-28到ID-18)、SRR和IDE位(都是隐性的)、标识符扩展(ID-17到ID-0)和RTR位组成。

- 在FBFF (IDE标志为显性的)，仲裁字段应该由11位基本标识符和显性RRS位组成。标识位用ID-28 ~ ID-18表示。

- 在FEFF (IDE标志为隐性的)，仲裁字段由基本标识符(ID-28到ID-18)、SRR和IDE位(都是隐性的)、标识符扩展(ID-17到ID-0)和显性RRS位组成。

**SRR bit[仅在CEFF和FEFF中使用]**

SRR比特位在ID-18比特位之后，在CBFF中的RTR比特位或FBFF中的RRS比特位的位置，以CEFF和FEFF的形式传输。SRR比特应以隐性方式传输，但接收方应接受隐性和显性SRR比特。

**RTR bit[仅在CBFF和CEFF中使用]**

RTR位的值在MAC DF中应占显性位。

**RRS bit[仅在FD帧中]**

在FD帧中，RRS比特应在经典帧中RTR比特的位置上传输。发送的RRS比特应是显性的，但接收方应接受隐性和显性的RRS比特。

**注意：接收SRR和RRS比特的两种状态意味着这两种状态都不会被视为格式错误。**

**IDE bit**

IDE bit 应区分CBFF和CEFF或FBFF和FEFF，即:是否属于

- CEFF和FEFF的仲裁场地，或

- CBFF和FBFF的控制字段。

在CEFF和FEFF中，IDE位应是隐性传输，而在CBFF和FBFF中，IDE位应是显性传输。应在 base identifier 与RTR位(CBFF)之后、 base identifier 与SRR位(CEFF)之后、 base identifier 与RRS位(FBFF)之后或 base identifier 与SRR位(FEFF)之后传输。

在IDE位上进行仲裁，以解决CBFF或FBFF格式的帧与CEFF或FEFF格式的帧之间具有相同 base identifier 的碰撞，使CBFF或FBFF格式的帧优于CEFF或FEFF格式的帧。

#### 10.4.2.4控制字段

控制域应该由6位(CBFF和CEFF)或8位(FEFF)或9位(FBFF)组成，其中最后4位应该是DLC，从LLC子层传递(见8.4.2.4)。

**CBFF**

在这种格式中，控制域的前两位是IDE位和FDF位，两者都应传输显性位。此帧格式向后兼容ISO 11898这部分的以前版本。

**CEFF**

在这种格式中，控制字段的前两位是FDF位和rO位，都是传输的显性位。此帧格式向后兼容ISO 11898这部分的以前版本。

**FBFF**

在这种格式中，前5位是IDE位(传输显性)、FDF位(传输隐性)和保留res位(传输显性)，然后是BRS位和ESI位。

**FEFF**

在这种格式中，前4位是FDF位(传输隐性)和res位(传输显性)，然后是BRS位和ESI位。

**FDF bit**

该比特位用于区分经典帧和全双工帧。在FD帧中它是隐性的，在经典帧中它是显性的。在具有11位标识符的帧中，FDF位于IDE位之后。在29位标识符的帧中，它是控制字段的第一个位。当检测到FD帧的隐性FDF比特而不是经典帧的显性FDF比特时，FD容忍接收机将检测到协议异常事件(见10.9.5)。

FDF位对应于标识符为11位的帧中的rO位，以及标识符为29位的帧中的r1位，这在ISO 11898这一部分的之前版本中有所规定。FD不容忍的CAN节点无法正确解码FD帧。

仲裁不会发生在FDF位上。一个给定的标识符应使用，用于经典帧或FD帧，但在特定的实现中不能同时用于两者。这并不禁止给定节点在发送经典帧和发送FD帧之间切换。

**r0 bit**

在CEFF中，FDF位之后是r0位，这是为协议未来扩展保留的。r0比特应以显性方式发送，但接收方应以隐性/显性接收。

注意：接收r0位的隐性状态意味着隐性状态不会被视为形式错误。。

**res bit**

在FD帧中，FDF比特之后是res比特，这是为协议未来扩展保留的。res比特应作为传输主导。启用FD的接收端在检测到res比特为隐性而不是预期的显性值时，应检测到协议异常事件(见10.9.5)。这是一个实现选项。如果启用该选项，可以禁用协议异常事件检测功能。禁用协议异常事件检测功能后，启用CAN节点检测FDF位后面的res位为隐性时，应将其视为形式错误(见10.11)。

**BRS bit**

该比特位表示是否在FD帧内切换比特率。如果检测到比特是隐性的，比特率应该从仲裁阶段的标称比特率切换到数据阶段的预配置数据比特率。在一个网络中，所有FD帧的BRS比特位都不需要相同。节点不需要发送具有特定BRS比特值的特定LLC帧。BRS在经典框架中不存在。

**ESI bit**

该标志应由错误主动节点发送为显性信号，由错误被动节点发送隐性信号。ESI在经典框架中不存在。

可选地，error active 的CAN节点可以在LLC用户的控制下，传输隐性状态的ESI位。

#### 10.4.2.5数据字段

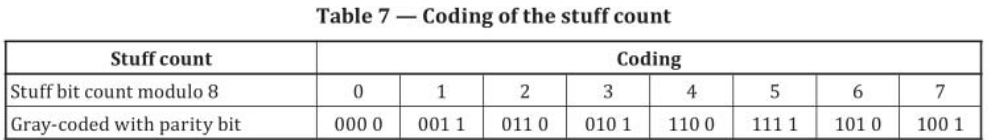
MAC帧数据字段应该等同于LLC数据字段(见8.4.2.5)。

#### 10.4.2.6 CRC字段

CRC字段应包含CRC序列，然后是一个隐性的CRC定界符。对于FD帧，CRC字段还应包含材料计数。

**Stuff count**

在FD帧中，stuff count应在CRC字段的开始处。它应该由3位格雷码模8的料位计数和一个奇偶校验位组成，如表7所示。



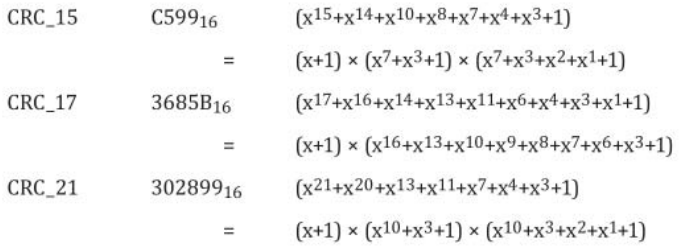
帧的发送端和接收端都应在帧中第一个固定的信息比特之前计算信息比特数。发射机应在CRC场开始时，在CRC序列之前传输其编码为stuff count的stuff bit count。接收方应检查收到的stuff count是否与自己的stuff count计算的值相匹配。

**CRC sequence**

帧校验序列应由CRC(BCH-code)导出。对于不同的帧格式，CAN节点应使用不同的CRC生成器-多项式。第一个多项式CRC\_15用于经典框架。第二个是CRC\_17，用于数据字段长度不超过16字节的FD帧。第三种是CRC\_21，用于数据字段超过16字节的FD帧。每个多项式的 hamming 距离为6。

在帧开始时，同时计算所有3个CRC序列;在所有节点，包括发射机。获得仲裁的节点发送由帧的FDF位和DLC值选择的CRC序列。接收方应仅考虑选定的CRC多项式以检查CRC-error。

CRC序列的长度(ncRc，生成器-多项式的阶)对于CRC\_15设置为15，对于CRC\_17设置为17，对于CRC\_21设置为21。



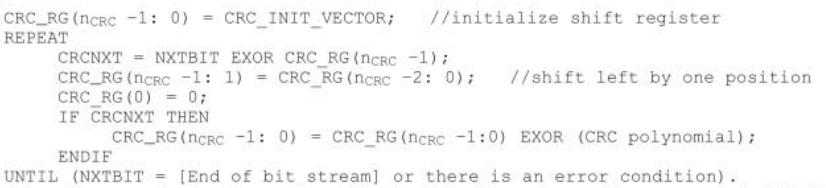
如何用十六进制表示法表示生成器多项式有不同的约定。ISO 11898这部分的表示法包括高阶位。

CRC\_15的 CRC\_INIT\_VECTOR 应为(0，..，0)，CRC\_17和CRC\_21的 CRC\_INIT\_VECTOR 应为(1,0，..，0);其中单个“1”位于最高位。

用于CRC计算的相关比特流是由SOF、仲裁字段、控制字段和(如果存在)数据字段组成的比特流，并辅以 “0” 的ncrc位。 在经典帧中，填充比特不应包含在相关比特流中进行CRC计算。在FD帧中，除固定信息比特外，填充计数和填充比特均应包括在相关比特流中进行CRC计算。

为了进行CRC计算，需要划分的多项式由相关比特流的系数定义。该多项式被生成器多项式除(系数按模-2计算)。

这个多项式除法的剩余部分是在总线上传输的CRC序列。为了实现这个函数，可以使用一个ncrc位移位寄存器CRC\_RG(ncrc -1:0)。每个CRC序列在单独的移位寄存器块中计算。如果 NXTBIT 表示比特流的下一个比特位，由SOF到数据字段结束的相关比特流给出，则CRC序列计算如下:



在相关比特流的最后一个比特发送/接收后，每个CRC\_RG包含三个CRC序列中的一个。

**CRC delimiter (CRC分隔符)**

CRC序列后面应跟着CRC定界符。在经典帧中，CRC定界符是一个隐性比特。在FD帧中，CRC分隔符可能由一个或两个隐性位组成。发送端应该只发送一个隐性比特作为CRC定界器，但是它应该在从隐性到显性的边缘开始ACK slot之前接受两个隐性比特。接收方将在第一个CRC分隔符位之后发送应答位。

**注意: 当FD帧到达CRC定界符(第一个比特)的采样点时，CAN的实现会从数据阶段切换回仲裁阶段。**

#### 10.4.2.7 ACK字段

ACK字段应包含ACK槽和ACK分隔符。在ACK域，发送节点发送隐性比特。所有接收方检查接收到的DF或RF是否一致，并确认接收到的帧是否一致，并通过EF标记不一致的帧(见10.12)。未被确认的DF或RF应被视为损坏，并应由发射节点用EF标记。

**ACK slot(ACK槽)**

所有接收到匹配CRC序列的节点(FD帧中为匹配数)都应在ACK时隙内发送ACK，其方式是用一个显性比特覆盖发送端的隐性比特(它们发送ACK)。在FD帧中，所有节点都需要接收一个长度不超过2比特的重叠ACK时隙主相位作为有效ACK，以补偿接收端之间的相位偏移。在经典帧中，在单个ACK槽位之后的显性位被认为是 form error。

**ACK delimiter(ACK分隔符)**

ACK分隔符是ACK字段的最后一个位，应该是一个隐性位。因此，ACK槽被隐性位包围(CRC定界符、ACK定界符)。

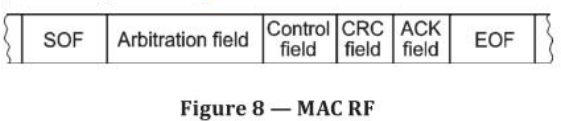
#### 10.4.2.8 EOF

每个DF和RF由一个由七个隐性比特组成的标志序列分隔，形成EOF。

### 10.4.3 MAC RF规范

#### 10.4.3.1描述

作为某些数据接收器的节点可以通过发送如图8所示的射频来发起其源节点各自数据的传输:



#### 10.4.3.2 MAC DF 和 MAC RF的相同场

SOF、CRC、ACK、EOF位域应等效于MAC DF的对应位域(见图Z)，射频中不得有数据域。

#### 10.4.3.3仲裁字段

仲裁字段应该由标识符字段组成，从LLC子层传递，和RTR位。在CBFF和CEFF中，MAC RF中RTR比特的值应该是隐性的。FBFF和FEFF中没有RF。

#### 10.4.3.4控制字段

在CBFF和CEFF中，MAC RF的控制场应等效于MAC DF的控制场(见8.4.2.4)。冲突解决(见10.9.9)要求射频的DLC的值等于请求的DF的DLC。

### 10.4.4 EF的规范

#### 10.4.4.1描述

EF由两个不同的字段组成。第一个字段由不同节点的错误标志叠加而成。第二个字段应该是错误分隔符。

#### 10.4.4.2错误标志

可以使用两种形式的错误标志，active error flag 和passive error flag，其中

- active error flag 由6个连续的主比特和组成

- passive error flag应该由6个连续的隐性比特组成，除非它被其他节点的显性比特覆盖。

error-active节点检测到错误情况时，应发送一个active error flag来表示这一点。错误标志的形式违反了位填充规则或破坏了需要固定形式的位域。因此，所有其他节点也会检测到错误，并开始发送错误标志。因此，总线上实际监控到的显性比特序列，是由各个节点发送的不同错误标志叠加而成的。这个序列的总长度可以在最小6位和最大12位之间变化。

由发送端发起的无源错误标志在比特填充编码的帧场中启动时，会在接收端引起错误(有两个例外)，因为这样会导致接收端检测到填充错误。第一个异常是一个passive error flag，在仲裁期间开始，另一个节点继续传输;第二个异常是一个passive error flag，在CRC序列结束前不到6位开始，并且CRC序列的最后一位刚好都是隐性的。

由接收器发起的passive error flag不能胜过总线上的任何活动。因此，错误被动接收机在检测到错误情况后，总是要等待6个后续的相等比特。当检测到这6个相等的比特位时，passive error flag完成。

#### 10.4.4.3 Error delimiter(错误分隔符)

错误分隔符应该由8个隐性位组成。在发送错误标志后，每个节点应发送隐性比特并监视总线，直到它检测到一个隐性比特。之后，它将开始发送更多的7个隐性比特。

### 10.4.5 OF 的规格

#### 10.4.5.1类型

以下类型的OF应具有相同的格式。

- LLC requested OF

LLC子层要求该OF表示内部过载情况(见10.13)。

- Reactive OF

Reactive OF的传输应在一定的错误条件下由MAC子层发起(见10.13)。

OF应包含两个位域，过载标志和过载定界符。过载标志应与活动错误标志对应。过载定界符应与错误定界符相同。

#### 10.4.5.2过载标志

过载标志由6个主导位组成。它会破坏中场休息字段的固定形式(见10.4.6节)。因此，所有其他节点也会检测到过载情况，并开始发送过载标志。

#### 10.4.5.3 重载定界符

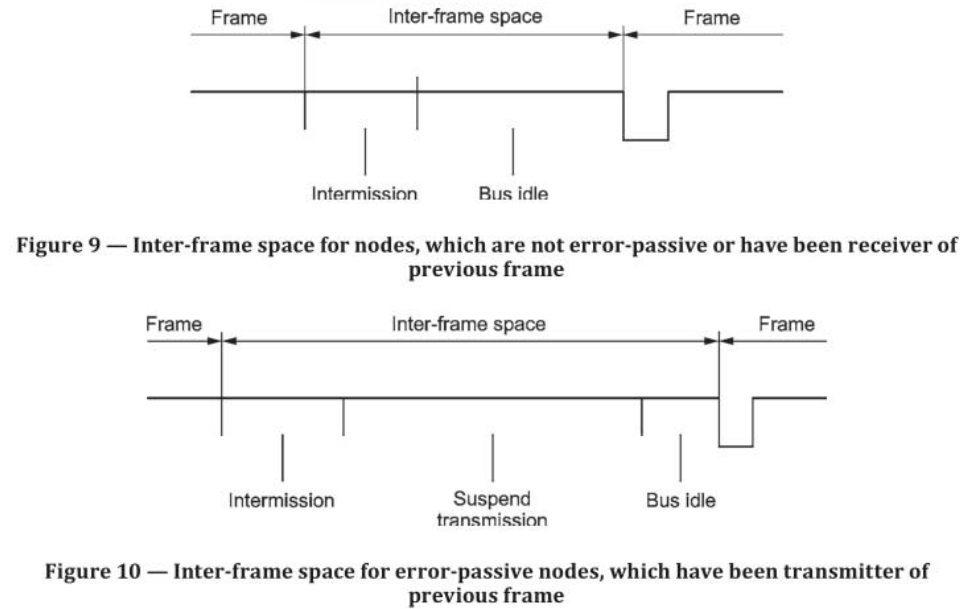
过载定界器应由8位隐性位组成。在发送过载标志后，每个节点将监视总线直到它检测到一个隐性比特。此时，每个节点应完成过载标志的发送，所有节点应同时开始多发送7个隐性比特，以完成8位长的过载分隔符。

### 10.4.6帧间空间规范(Specification of inter-frame space)

#### 10.4.6.1描述

DFs和RFs应该与之前的帧分开，无论它们是什么类型的帧(DF, RF, EF, OF)，通过一个称为帧间空间的时间周期。与此相反，EFs和OFs之前不能有帧间空间，多个OFs之间不能被帧间空间分隔。

帧间空间应包含位域间歇和总线空闲时间。对于已经是前一帧发送的错误被动节点，帧间空间还应包含该节点的暂停传输时间(见图9和图10)。



#### 10.4.6.2幕间休息

间歇场应由3个隐性位组成。在间歇期间，任何节点不得开始发送DF或RF。只允许过载状态的信号。

在间歇的第三位处检测显性位应被解释为SOF(见10.4.2.2)。

#### 10.4.6.3总线空闲

总线空闲的时间可以是任意长度。当第三位间歇被视为隐性时，总线应由接收器和错误主动发射器识别为空闲;由无源发射机误差时，第8位被暂停的传输时间可见隐性;或总线集成状态处于左侧时(参见10.9.4)。当总线空闲时，任何节点都可以访问总线进行传输。

在传输另一帧期间，等待传输的帧应在间歇后的第一个比特中开始。

总线空闲时对总线上显性位的检测应被解释为SOF。

#### 10.4.6.4暂停传输

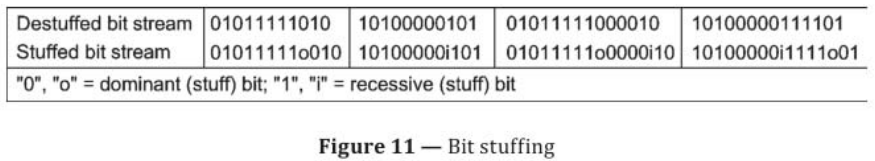
一个已经发送前一帧的错误被动节点，将在间歇后暂停进一步帧传输的开始8比特次。如果另一个节点在该暂停传输时间内开始传输，该节点将成为该DF或RF帧的接收方。

### 10.5帧编码

帧内的比特流应按照NRZ编码方法进行编码。这意味着生成的比特级别在总比特时间内是恒定的。

为了限制可用于同步的边之间的最大距离，对SOF、仲裁场、控制场、数据场和CRC序列等帧段采用比特填充的方法进行编码。当发送器在要传输的比特流中检测到五个连续的相同值的比特(包括填充比特)时，它将自动插入一个互补比特(称为填充比特)到实际传输的比特流中(见图11)。接收方应识别一个由五个相同值的连续比特组成的序列，并应丢弃下列比特。

bit填充如图11所示。



在FD帧的CRC域中，将填充位插入固定位置;它们被称为固定的填充。在计数的第一个比特之前，应该有一个固定的比特位，即使前一个字段的最后一个比特位不是五个连续相同值的比特位的序列。如果前一个字段的最后一个比特是连续五个相同值的比特，则只有固定的比特，不应该有两个连续的比特。在CRC字段的每四个比特位之后，应插入一个进一步的固定内容比特位。该固定比特位的值应为该固定比特位之前比特位的倒数。接收方应从比特流中丢弃固定的填充位元以进行CRC检查。如果固定的填充位与它的前一个位具有相同的值，它将检测一个格式错误。FD帧的CRC校验域中固定的信息比特数等于应用经典帧的比特填充方法所能产生的最大信息比特数。

DF或RF的剩余位域(CRC分隔符、ACK域和EOF)应是固定形式且不填充。

EF和OF也应该是固定形式，并且不能用bit填充的方法进行编码。

## 10.6帧确认

所有接收方应检查所有接收的DFs和RFs的一致性，并确认所有一致的DFs和RFs。确认不应依赖于帧的标识符。

## 10.7帧验证

帧被认为有效的时间点对于帧的所有接收端应该是相同的，但对于帧的发送端应该是不同的。

**接收机**

如果在EOF的最后一个比特之前没有错误，则该帧对接收方是有效的。EOF的最后一位值不应抑制帧验证，主导值不应导致格式错误。

在EOF的最后一个比特位检测到主比特位的接收机应以 OF 响应(见10.13)。

**发射机**

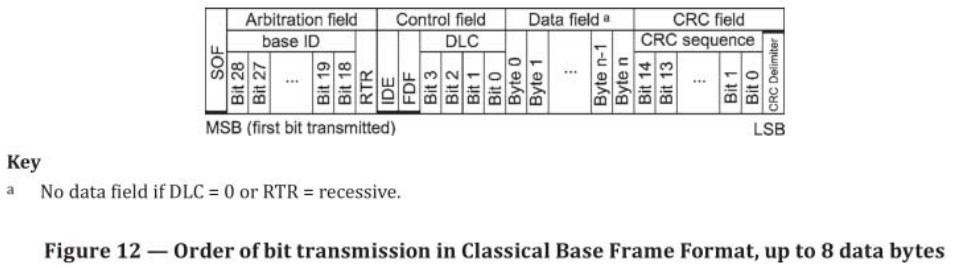
如果在EOF结束前没有错误，则帧对发射机是有效的。如果帧损坏，恢复将按照10.9.6中描述的处理。

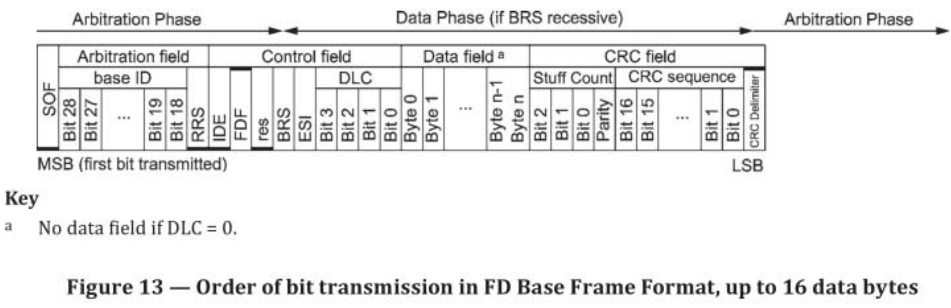
如果发送端在EOF的最后一个比特(发送端的全局或局部错误)采样了一个显性比特，则该帧对接收端有效，但对发送端无效。发射器放置一个EF并重新启动帧的传输，帧将被接收两次。接收端将显性比特位视为OF，并将下一帧作为独立的第二帧接收。

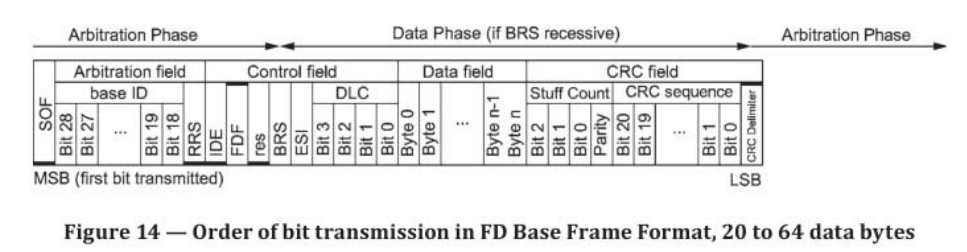
## 10.8位传输顺序

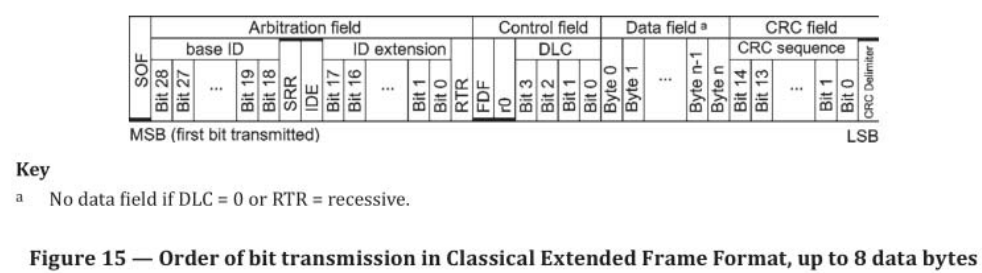
DFs和RFs需要逐位传输，从显性的SOF位开始。在一个领域内，应首先传输MSB。在data字段中(如果有的话)，字节从0 转移到n (n+1是数据字节的数量，如表5所定义)。在每个字节中，比特从第7位转移到第0位。

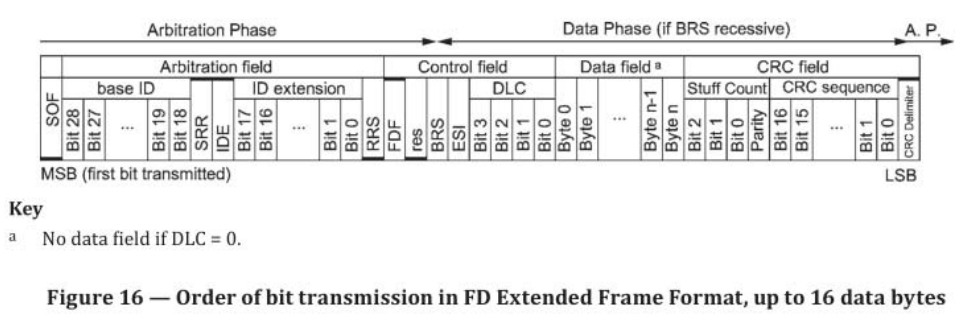
在图12到图17中，比特底部的宽线表示该比特应以显性方式传输，比特顶部的宽线表示该比特应以隐性方式传输。填充位没有显示在图中。

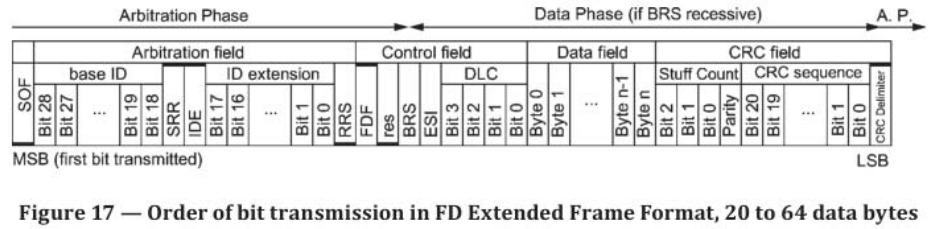












## 10.9介质访问方法

### 10.9.1 通用

本章描述了CAN介质访问方式的相关功能和特点。

#### 10.9.2多主机

在传输过程中，每个发送DF或RF的节点都应是总线主节点。

### 10.9.3总线访问

在总线空闲时(参见10.4.6.3)，一个活动错误的结点可以访问总线。当总线空闲时，接收当前帧或前一帧的错误被动节点可以访问总线。当前帧的发送节点或曾经是前一帧的发送节点，在其暂停传输时间结束后，只要没有其他节点同时开始传输，就可以访问总线。当多个节点同时开始传输时，此时传输优先级最高的节点将成为总线主节点。解决总线访问冲突的机制是基于内容的仲裁。

### 10.9.4总线集成状态(Bus integration state)

CAN节点在启动协议操作后，在总线恢复过程中，或(对于允许全双工或全双工的节点)在检测到协议异常状态后进入总线集成状态。检测到空闲状态(见4.28)时，CAN节点应离开总线集成状态。

当总线集成状态进入或在采样点检测到CAN总线显性时，应该有一个位计数器复位。当CAN总线在采样点被检测为隐性时，位计数器将增加。当该位计数器达到值11时，将检测到空闲状态。对于总线断开恢复状态的检测(见12.1.4.4)，应该有第二个计数器，每次检测到空闲状态时该计数器加1。

处于bus-off状态的节点在检测到空闲状态后，如果bus-off恢复条件尚未满足，应立即重新进入总线集成状态。

对于FD容忍或FD启用的节点，位计数器有第三个重置条件。当检测到引起同步的边缘时，它应该被重置。当同步发生时，将重新开始计算连续11个隐性比特的序列。有关可选的边缘过滤，请参见11.3.2.3。

当一个FD容忍或FD启用的节点正在集成时，它作用于比标称比特时间短的主导脉冲，以确保FD帧的数据相位的比特不会被误认为是空闲状态。

### 10.9.5协议异常事件

FD容忍的CAN节点和支持FD的CAN节点检测10.4.2.4中指定的协议异常事件。作为对协议异常事件的反应，错误计数器不改变，硬同步使能，节点发送隐性比特，进入总线集成状态(见10.9.4)。

### 10.9.6 MAC帧传输

根据10.9.3的规定，在允许节点访问总线时，可能启动MAC DFs和MAC RFs。MAC EF应按照10.12的规定传送。MAC OF应按照10.13的规定传送。

仲裁失败的MAC DFs和MAC RFs，未被确认的MAC DFs和MAC RFs，或在传输过程中被错误干扰的MAC DFs和MAC RFs，应自动重传。被重传的MAC DF或MAC RF应被处理为任何其他MAC DF或MAC RF，即它参与仲裁过程以获得总线访问。当不再请求传输该帧时，应禁用该帧的自动重传。可选地，可以对CAN节点进行配置，以将重传尝试限制为特定的次数。所有帧的自动重传可能被禁用。

### 10.9.7基于内容的仲裁 (Content-based arbitration)

在仲裁期间，每个发送器应比较传输的比特的水平与总线上监视的水平。如果这些级别相等，节点可以继续发送。当发送一个隐性层而监测到一个显性层时，该节点将失去仲裁，不再发送任何比特而退出仲裁。当发送显性电平并监测隐性电平时，节点应检测到比特错误。

对于基本格式消息，**从标识符的第一个比特到IDE比特，以及扩展格式消息的标识符后面的RTR比特或FD帧中的RRS比特，都需要进行基于内容的仲裁。**

### 10.9.8帧优先级

在两个标识符不同的帧中，对标识符二进制值较低的帧赋予较高的优先级。

如果同时发起一个DF和一个具有相同标识的RF，则DF的优先级高于RF。这应该通过给RTR位分配值来实现。

### 10.9.9碰撞解析(Collision resolution)

传输只能在总线空闲时发起。当两个或多个帧同时开始时，这称为碰撞(collision)。CAN位仲裁方法解决了具有不同标识符或不同帧类型的DFs和RFs之间的冲突。如果碰撞的帧不相同，则未解决的碰撞会导致EFs。

### 10.9.10禁用帧格式

可选地，实现可以具有配置接口，该接口允许禁用经典的CAN帧格式或灵活的Data Rate帧格式。如果一种帧格式被禁用，该格式的帧将被视为无效帧，从而导致错误帧。一个启用FD的实现不应该被设置为一种模式，在这种模式下它的行为与一个FD容忍的实现一样。

## 10.10 MAC数据一致性

要传输的消息由LLC用户准备，通过节点的控制器-主机接口和数据链路层的LLC子层传输到负责消息分帧的MAC子层。消息可以存储在共享内存中。从共享内存中传输的消息的数据一致性应由两种方法中的至少一种确保。

- MAC子层将整个要传输的消息存储在一个临时缓冲区中，该缓冲区在传输开始前填满。

- LLC子层在将消息传输到MAC子层时检查数据错误。如果检测到数据错误，则不启动传输。如果在检测到数据错误时已经启动，则该节点应切换到总线监控模式，见10.14，或切换到受限运行模式，见10,15。接收节点将看不到有效的消息。

**注意： 数据错误，例如:RAM字中的奇偶校验错误，未及时提供的数据，或在传输过程中由LLC用户部分更新的数据。如果使用第一种确保数据一致性的方法，则可选择实现操作模式总线监控模式和受限操作模式。**

## 10.11错误检测

MAC子层应提供以下错误检测机制:

— monitoring;

— stuff rule check;

— frame check;

— stuff count check in FD Frames,

— 15-bit,17-bit, or 21-bit CRC;

— ACK check.

有5种不同的错误类型，它们并不互斥。

1. **Bit error**

在总线上发送比特的节点也应该监视总线。当被监控的比特值与发送的比特值不同时，在该比特时刻检测比特错误。

例外情况:当在仲裁期间发送隐性信息位或在ACK时隙发送隐性信息位时，显性比特不会导致比特错误。一个节点发送一个被动错误标志并且检测到一个显性比特不能将其解释为一个比特错误。

**b) Stuff error**

在采用比特填充方法编码的帧场中，在连续第六个等比特级别的比特时，应检测到填充错误。当FD帧的CRC域中有固定的stuff位没有达到其期望值时，认为是一种格式错误，而不是stuff错误。

**c) CRC error**

CRC序列应由发射机的CRC计算结果组成。接收机应以与发射机相同的方式计算CRC。当计算出的CRC序列与接收到的CRC序列不相等时，应检测CRC错误。在FD帧中，计数的数据位与接收到的数据位之间的不匹配被认为是CRC错误。

**d)Form error**

当固定格式的位字段包含一个或多个非法位时，将检测到form error。

例外: 接收端在EOF的最后一位监视显性位，或任何节点在错误分隔符或过载分隔符的最后一位监视显性位，不得将此解释为form error。

**e) ACK error**

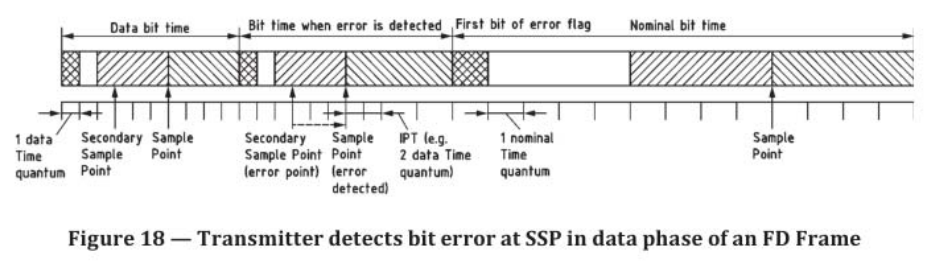
当发射机在ACK时隙中没有监视主比特时，ACK error将被发射机检测。

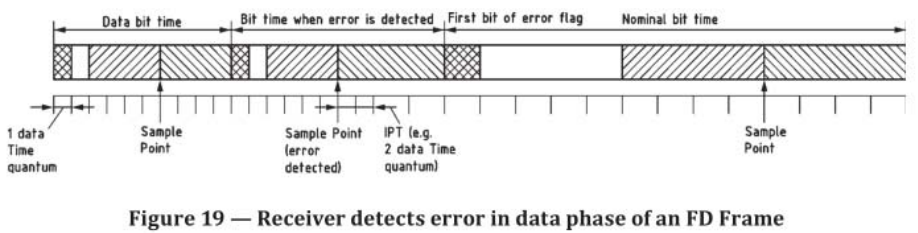
当检测到这些错误之一时，应通知LLC子层。因此，MAC子层将发起一个错误标志的传输。

## 10.12错误信令

当任何节点检测到bit error, stuff error, form error, 或者 ACK error时，对应的节点应在下一个比特处启动一个错误标志。当在FD帧的数据阶段检测到错误时，在它开始错误标志之前，节点应该从数据位时间切换回仲裁阶段的标称位时间。使用TDC(见11.3.3)的变送器应按图18所示的方式切换位时间。接收机(和不使用TDC的发射机)应按图19所示切换位时间。图18和图19中的阴影表示图22所示的比特时间的部分。

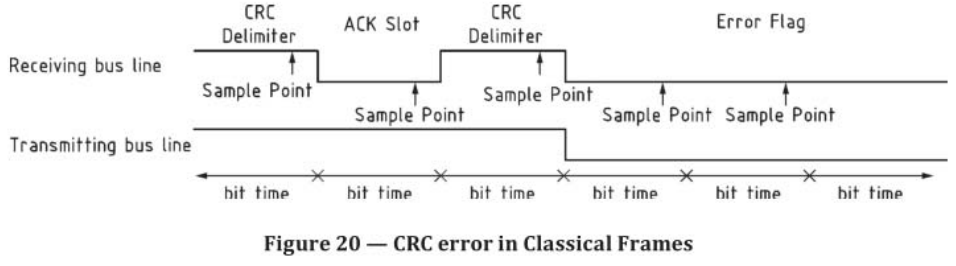
在SSP检测到错误之后，在SP和IPT之后，比特计时被设置回名义比特速率，但在IPT期间，在启动下一个比特并开始发送错误标志之前，已经对Phase\_Seg2进行了计数，而Phase\_Seg2必须通过。

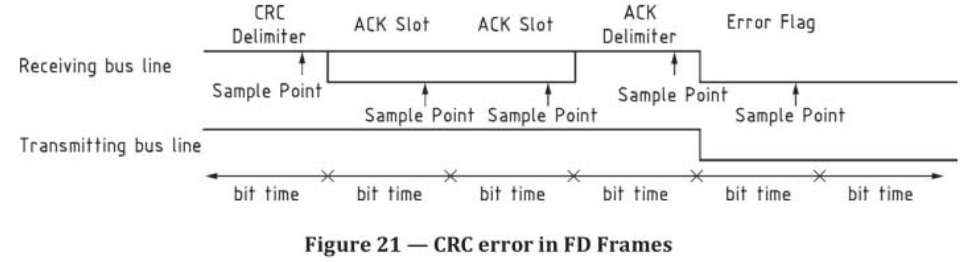




接收端接收到经典帧并检测到CRC错误，应在ACK定界器之后发送EF。接收端接收到FD帧并检测到CRC错误后，应在CRC定界器之后发送3位后的EF。经典帧如图20所示，FD帧如图21所示。

在CRC分隔符和EF开始之间看到的显性位不应被视为错误。





## 10.13 过载信号

以下条件将导致 OF 的传输。

a) LLC-requested OF(由LLC子层发起):接收端内部条件，要求下一个MAC DF或MAC RF有延迟

b)Reactive OF(由MAC子层发起):在间歇的前两位中的任何一位检测一个显性比特，接收方在EOF的最后一位检测一个显性比特，或任何节点在错误分隔符或过载分隔符的最后一位检测一个显性比特。

由于上述条件b)， LLC-requested OF 只在预期间歇的第1位启动，而 reactive OFs 则在检测到主导位后启动1位。上述a)条件下的 start of LLC-requested OFs 是允许的，但不要求实施。

最多产生两个LLC OFs来延迟下一个MAC DF或MAC RF。

## 10.14总线监控

可选的，CAN实现可以提供总线监控模式，在这种模式下，它们应该能够接收有效的DFs和有效的RFs，但它只在CAN网络上发送隐性比特，而不启动传输。

如果MAC子层需要发送一个显性位(ACK位、过载标志、主动错误标志)，该位必须在内部重新路由，以便MAC子层监视这个显性位，尽管CAN网络可能仍然处于隐性状态。

## 10.15操作限制

可选的，CAN实现可以提供受限的操作模式，在这种模式下，它们应该能够接收DFs和RFs，并且应该给有效的帧ACK，但是它不应该发送EFs或OFs。在出现错误状态或者过载状态时，不发送主比特;相反，它应将此视为协议异常事件，并应进入总线集成状态(见10.9.4)。当CAN节点处于限制运行模式时，错误计数器不能增加或减少。如果受限运行模式下的CAN节点是网络中潜在的时间主人，则应能够传输时间参考消息启动网络;其他帧不得传输。

# 11 PL规格

## 11.1一般和功能模型

PL实现将CAN节点连接到总线。总线上的电力负载和CAN数据链路层协议限制了节点的数量。

PL根据ISO/IEC 8802-3建模。它有三个子层。

a) PCS 应包括位编/解码、定时、同步以及总线故障检测等相关功能。它在后续的子句中指定。

b) PMA 子层包括总线传输/接收的功能电路。它不在ISO 11898这一部分的范围内。

c) PMD 子层包含物理介质和PMA子层之间的机械和电气界面。它不在ISO 11898这一部分的范围内。

## 11.2 PL服务

### 11.2.1描述

PL的服务应允许本地MAC子层实体与对等MAC子层实体交换比特。

PL应向MAC子层提供以下服务原语:

- PCS\_Data.Request;

- PCS\_Data.Indicate.

对于启用FD的实现，应该有两个额外的可选服务原语:

- PCS\_Status.Transmitter;

- PCS\_Status.Receiver.

### 11.2.2 PCS\_Data.Request

PCS\_Data.Request原语应从MAC子层传递到PL以请求传输一个显性或隐性比特。该原语提供以下参数:

PCS\_Data.Request(

Output\_Unit

)

Output\_Unit参数应采用两个值之一:显性或隐性。

### 11.2.3 PCS\_Data.Indicate

PCS\_Data.Indicate 原语应该从PL传递到MAC子层，以便指示显性或隐性比特的到来。原语应提供以下参数。

PCS\_Data.Indicate(

Iput\_Unit

)

Input\_Unit参数应该采用两个值中的一个，每个值代表一个比特:显性或隐性。

### 11.2.4 PCS\_Status.Transmitter

PCS\_Status.Transmitter原语应从MAC子层传递到PL，以表明MAC子层传输FD帧的数据相位。原语应提供以下参数。

PCS\_status.Transmitter(

FD\_Transmit

)

FD\_Transmit参数的取值有两种:active表示MAC子层传输FD帧的数据阶段;passive表示MAC子层不传输FD帧的数据阶段。

### 11.2.5 PCS\_Status.Receiver

PCS\_Status.Receiver原语应从MAC子层传递到PL，以表明MAC子层接收到FD帧的数据相位。原语应提供以下参数。

PCS\_Status.Receiver(

FD\_Receive

)

FD\_Receive 参数的取值有两种:active表示MAC子层接收到FD帧的数据相位;passive表示MAC子层没有接收到FD帧的数据相位。

## 11.3 PCS规格

### 11.3.1比特编码/解码

#### 11.3.1.1比特时间（bit time）

在位时间帧内执行的总线管理功能，如CAN节点同步行为、网络传输延迟补偿、样本点定位等，应由CAN实现的可编程位时序逻辑给出。

FD使能实现应该支持两种比特率，名义比特率和数据比特率。不启用FD的实现被限制在经典CAN指定的名义比特率。定义为第二个比特率，数据比特率与数据比特时间，需要一个单独的配置寄存器集。数据位时间应与标称位时间相同或短于标称位时间。

数据位时间只能在FD帧的数据阶段使用。如果检测到BRS位是隐性的，则数据相位应从BRS位的采样点开始。当到达CRC定界器的第一个采样点或当CAN实现看到导致开始EF的错误条件时，此数据阶段应结束。数据阶段之外的阶段应为仲裁阶段。所有经典帧、EFs、OFs、空闲时间以及FD帧中具有显性BRS比特的所有部分都属于仲裁阶段。仲裁阶段内的标称比特时间应使用。在隐性BRS比特的采样点，比特率应从名义比特率切换为数据比特率。在CRC定界器的第一个采样点或检测到错误条件时，应将比特率从数据比特率切换到标称比特率。当检测到误码而切换码率时，切换时间应在采样点之后偏移，偏移幅度不大于两个时间量子;参见图18和图19。

**注意： 由于比特率是在BRS位和CRC定界符位的采样点切换的，因此这两个位的长度是中间的。这两个比特的长度之和等于标称比特时间和数据比特时间之和;参见图22。**

**时间量**

时间量子应该是一个从节点时钟周期导出的固定时间单位。应存在至少一个可编程分频器，其值为整数，范围至少为1至32。最小时间量子为一个节点时钟周期。时间量的长度为

- 时间量子(N)=m(N)· 标称比特时间的最小时间量子，以及

- 时间量子(D)=m(D)· 数据位时间的最小时间量子，

其中m(N)和m(D)是分频器的值。

也应该有两个实现选项

- 两个独立的分频器，m(N)表示标称位时间，m(D)表示数据位时间，或

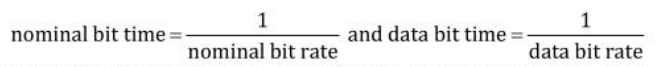
- m(N)=m(D)的共享分频器。

两个独立的分频器使时间量子(N)和时间量子(D)具有不同的长度。在分频器共享的情况下，时间量子(N)与时间量子(D)具有相同的长度。比特时间的长度取决于时间量子的长度和比特中时间量子的数量。如果不同的参数组合得到的比特时间长度相同，则组合的时间量子应使用较短。

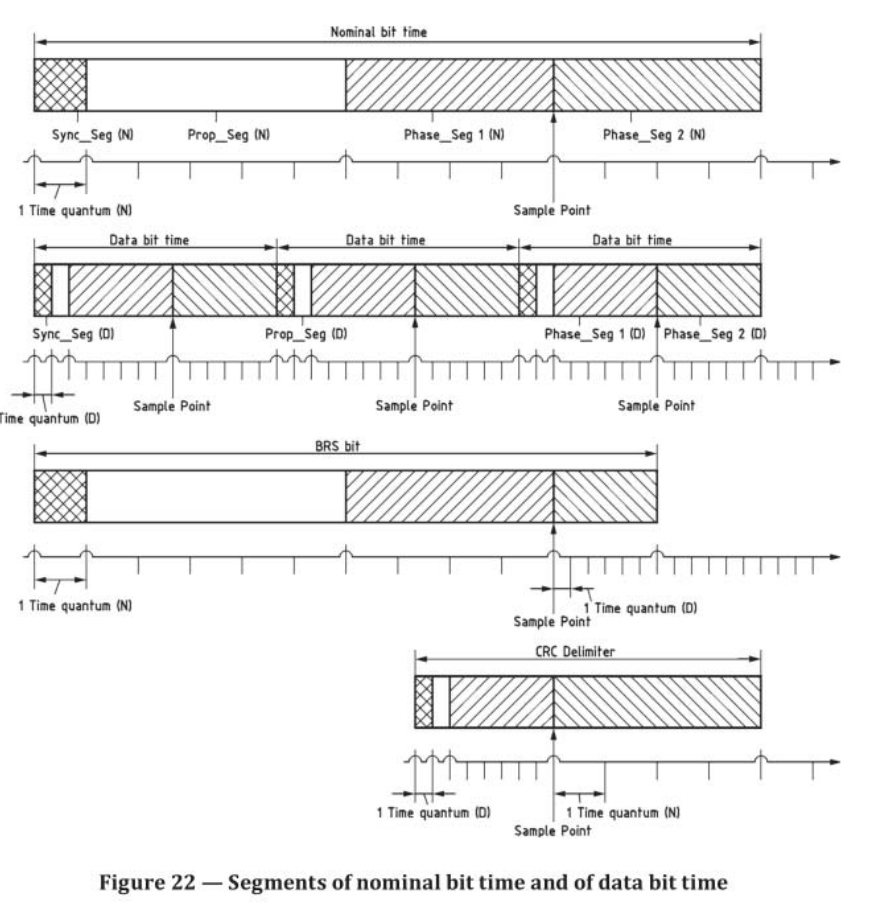
不启用FD的实现仅限于基于时间量子(N)的名义比特时间。

**比特率和比特时间**

比特率给出了理想发送端在没有重新同步的情况下每秒传输的比特数;比特速率和比特时间之间的关系是



两个比特时间都应该由互不重叠的时间段组成。分段应形成如图22所示的位时间。



**Sync\_Seg**

这部分位时间，即同步段，应使用用于同步总线上的各个CAN节点。我们希望在这一段中检测到边缘。

**Prop\_Seg**

这部分比特时间，即传播时间段，以补偿应使用在网络内的物理延迟时间。这些延迟时间包括总线上的信号传播时间和CAN节点内部的延迟时间;参见图23。

**Phase\_Seg1, Phase\_Seg2**

这些相位缓冲段1和2用于补偿边缘相位误差。这些段可以通过重新同步延长或缩短。

**Sample point**

采样点应该是总线级被读取和解释为相应位值的时间点。它的位置应该在Phase\_Seg1的末尾。

**Information processing time(信息处理时间)**

信息处理时间为后续比特级计算所需的时间量子数。该计算从采样点开始，必须小于或等于Phase\_Seg2(其他限制参见11.3.2.1)。

**同步跳跃宽度(SJW)**

重新同步的结果是，Phase\_Seg1可以延长，Phase\_Seg2可以缩短。同步跳变宽度对相位缓冲段的延长和缩短有一个上限。

**内部时延**

相对于CAN实现的位时序逻辑单元，CAN节点的内部延迟时间tnode应该是在传输和接收路径上发生的所有异步延迟的总和。有关更多细节，请参见图23。

结合图23

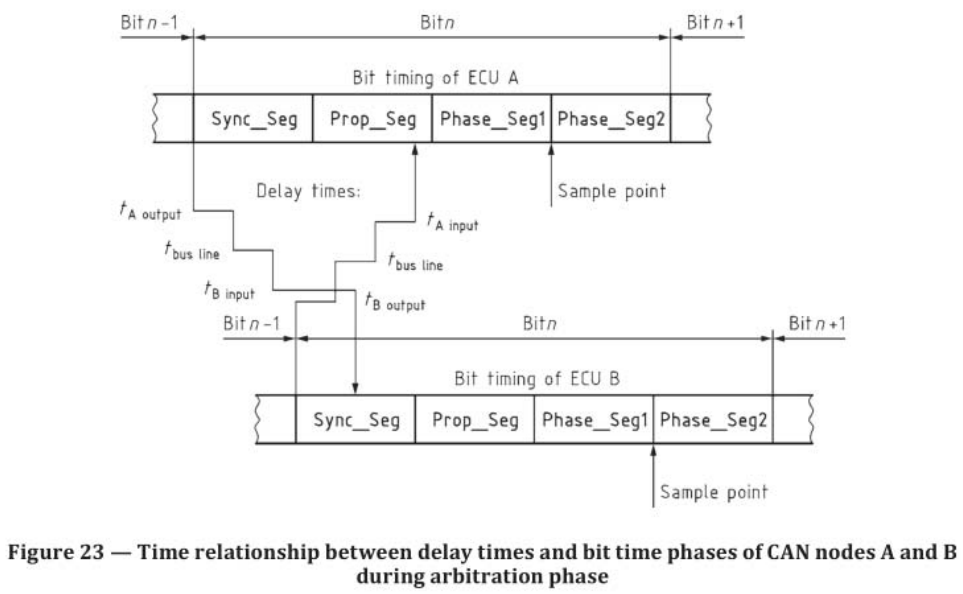
- 输出和输入CAN节点延迟之和相对于标称位时间的配置是至关重要的。CAN节点的重要特征参数由式(1)给出

(1) tnode = toutput + tinput

- 适当仲裁应当符合下列条件

(2)tprop\_Seg≥tnode\_A + tnode\_b + 2 × tbusline

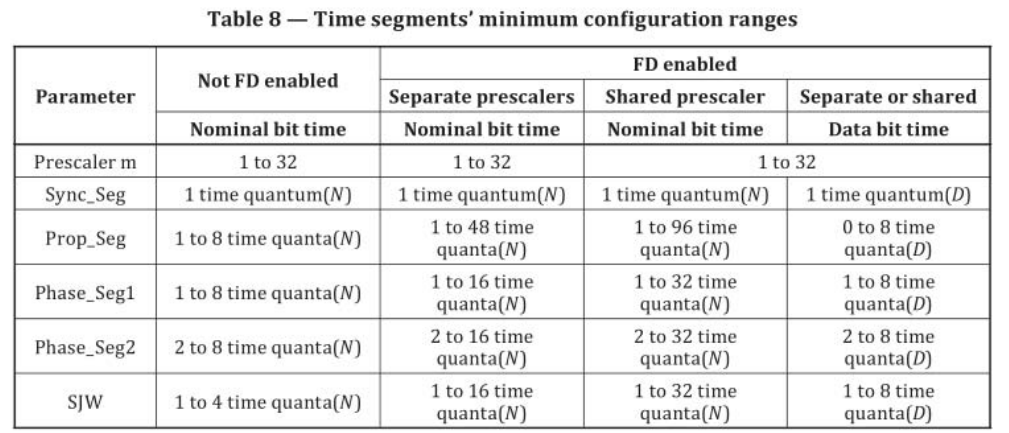
- 领先的传输位时序逻辑对于CAN的同步，节点A应该能够知道采样点位n的正确总线电平。tnode的可容忍值取决于总线所需的比特率和线路长度(任意两个节点之间的最大距离)以及仲裁条件所示的可能的比特时序。



#### 11.3.1.2位时间参数配置

位时间的配置应使用以下时间段进行:Sync\_Seg, Prop\_Seg, Phase\_Seg1, Phase\_Seg2，和SJW，所有代表基于最小时间量值和分频器m的整数时间量值。对于启用FD的实现，有两组配置值，每个比特时间一个;请参见图22。

对于启用FD的实现和未启用FD的实现，位时间段长度的配置范围应该不同。除了同步段，它应该恰好是一个时间量子长，实现可能允许时间段超过表8中指定的最小要求配置范围。



位时间段的配置应满足以下限制。

—信息处理时间应小于等于2个量子数。

—在数据位时间上，Phase\_Seg2应大于等于最大信息处理时间。

—在标称比特时间中，Phase\_Seg2应大于等于SJW和信息处理时间这两个项的最大值。

—在标称位时间和数据位时间中，SJW应小于或等于Phase\_Seg1和Phase\_Seg2这两项中的最小值。

在同步的情况下，Phase\_Seg1可能比程序设定的值长，Phase\_Seg2可能比程序设定的值短。所述采样点的位置可以在两个位定时配置中不同;在数据比特率的配置中，Prop\_Seg的长度可以为零。

基于m(N)=3的分频器和时间段Prop\_Seg(N)=5, Phase\_Seg1(N) =4, Phase\_Seg2(N)=4的标称位时间配置，结合基于m(D) = 1的分频器和时间段Prop\_Seg(D) = 1,Phase\_Seg1(D)= 6, Phase\_Seg2(D) = 6的数据位时间配置，以及中间长度BRS和CRC分隔符的结果位，如图22所示。

在CAN实现中，Prop\_Seg和Phase\_Seg1不需要单独编程;编写Prop\_Seg和Phase\_Seg1的总和就足够了。对于未启用FD的实现，在标称位时间内的时间量子总数应至少从8到25可编程。对于启用FD的实现，数据位时间内的时间量子总数应至少从5到25可编程，并且在标称位时间内至少从8到80可编程。

如果在标称位时间和数据位时间中采用相同的时间量子长度，且标称位时间内的采样点在网络中所有CAN节点的位置相同，则实现了FD帧网络的最优时钟容差。

不同CAN节点的节点时钟振荡器的频率应该进行协调，以便为标称位时间和FD使能实现的数据位时间提供一个全网指定的时间量。协议实现的可接受振荡器公差(参见11.3.2.5)和不正确同步的潜在可能性由Phase\_Seg1、Phase\_Seg2和SJW决定。

当CAN实现的位时间配置不可编程时，其固定位时间配置应满足本章对位时间段配置的限制。

### 11.3.2同步

#### 11.3.2.1描述

将CAN实现的操作同步到CAN总线上的信号的状态机应按一个时间量子的时间步长操作。在每个时间量上，需要分析总线状态是隐性的还是显性的。在某位采样点检测到的总线状态，视为该位的值。两个连续时间量子之间总线状态的差异是一条边。与未受干扰帧同步的接收节点仅在一比特时间的Sync\_Seg中检测边缘。在Sync\_Seg之外检测到的边可能导致CAN节点将其操作同步到该边。

说明将物理信号转换为数据信号的总线比较器可以独立于节点时钟运行。在被时钟状态机视为数据信号之前，将数据信号同步到节点时钟所需的时间是can节点输入延迟时间的一部分(参见图23)。

硬同步和重同步是同步的两种形式。他们应当遵守以下规则。

a)一个比特时间内(两个采样点之间)只允许一次同步。在检测到边缘后，同步将被禁用，直到下一次在采样点检测到的总线状态是隐性的。

b)只有在前一个采样点检测到的总线状态(前一个读总线状态)是隐性的时，边缘才会引起同步。如果发射机使用发射机延迟补偿(见11.3.3)，在CRC定界器的采样点之后，从隐性到显性的第一个检测边缘也会引起同步。在这种情况下，CAN实现可以选择性地选择在第一个检测边缘上同步，从隐性到显性，至少由发送端延迟加上CRC定界器的采样点之后的一个时间量子的量所看到。具有正相位误差的边(见11.3.2.2)不会导致发送主比特的节点同步。

c)在帧间空间、节点处于总线集成状态时，以及在FD帧内边缘(FDF比特与后续支配res比特之间)进行硬同步(间歇第一个比特除外)。

d)所有其他隐性到主导边满足规则 a)和 b)应使用，用于重新同步，但有一个例外:传输FD帧的节点在传输该帧的数据阶段时不进行同步。

时钟信息应从一个位值到另一个位值的转换中得到。由于比特填充，只有固定的最大连续比特数具有相同的值，这一特性将提供在帧期间重新同步CAN节点到比特流的可能性。

#### 11.3.2.2边缘的相位误差

边缘的相位误差e由边缘相对于Sync\_Seg的位置给出，用时间量子测量。相位误差的符号如下所示:

- e=0，如果边在Sync\_Seg内;

- e>0，如果边缘位于Sync\_Seg和当前比特的采样点之间;

- e<0，如果边缘位于当前比特的采样点和下一个比特的Sync\_Seg之间。

#### 11.3.2.3硬同步

在硬同步之后，在Sync\_Seg完成的情况下，每个位定时逻辑单元将重新启动位时间。因此，硬同步将迫使导致硬同步的边缘处于重启比特时间的同步段内。硬同步的作用不受同步跳变宽度的限制。

可选地，CAN实现可以在总线集成状态期间执行边缘过滤;看10.9.4。当配置使能该选项时，需要两个具有主导总线状态的连续标称时间量来检测引起同步的边缘。同步发生在第二次发现具有支配总线状态的量子之后。

在进行边缘滤波时，应忽略小于两个标称时间量子的显性总线状态。

#### 11.3.2.4位重同步

重新同步将导致比特时间的缩短或延长，以纠正样本点相对于被检测边缘的位置的位置。当引起重同步的边缘相位误差的大小小于或等于同步跳变宽度的规划值时，重同步的效果与硬同步相同。

当相位误差的大小大于同步跳变宽度的程序值时

-如果相位误差e为正，则将Phase\_Seg1延长与同步跳变宽度相等的长度;

-如果相位误差e为负，则Phase\_Seg2应缩短一个与同步跳变宽度相等的量。

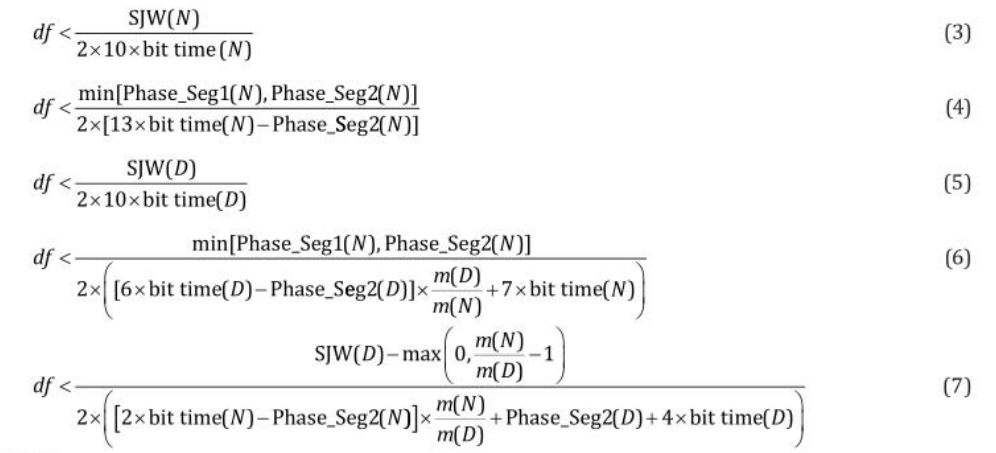
如果将Phase\_Seg2缩短到一个小于信息处理时间的值，则可以在Phase\_Seg2结束后完成后续比特级的计算。

由于同步仅在总线状态为读隐性时启用，并且在两个采样点之间最多只允许一次同步，因此只有从隐性到显性的边与同步相关。第一个边缘将用于同步，之后，在检测到第一个边缘时同步被禁用，即使该边缘没有引起同步，例如，因为它在Sync\_Seg内部检测到

#### 11.3.2.5振荡器频率的公差范围

节点时钟振荡器频率 fosc 在公称频率 fnom 附近的误差范围为 [(1-df) × fnom ≤ fosc ≤ (1+df) × fnom]。容差df取决于时间量的长度、比特时间的分段以及同步跳变宽度。任意两个节点的节点时钟振荡器之间的最大差值应为 2 × df × fnom

fosc的最大公差df应满足以下条件:



Where

(N)表示仲裁阶段的数值;

(D)表示数据阶段的值。

对于经典框架，满足公式(3)、(4)中的条件;FD帧应满足公式(3)至公式(7)的所有条件。

要考虑的是，SJW不能大于相位缓冲器段的较小值，并且传播时间段限制了可用于相位缓冲器段的部分比特时间。

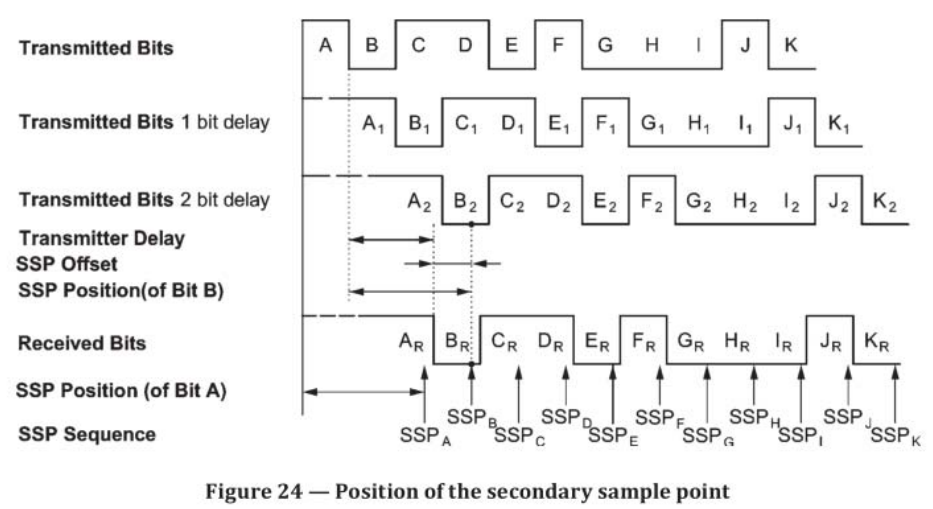
### 11.3.3发射机延迟补偿

CAN节点通过电路(即收发器)连接到CAN网络，在协议控制器输出信号上发送比特信号的时间点和协议控制器输入信号上出现相同比特信号的时间点之间，不可避免地存在一个时间延迟。在没有发送端延迟补偿的情况下，FD帧数据相位的比特率会受到限制，因为如果发送端在该比特的采样点最近没有接收到自己发送的比特，那么发送端就会检测到比特错误。

FD使能实现应支持发射机延迟补偿机制，用于数据阶段的比特时间长度小于从PCS到PMA输出消息(见11.4.2.1)到PMA到PCS输入消息(见11.4.3.1)的应用，这在ISO 11898的其他部分中有所规定。当BRS是隐性时，只有在FD帧的数据阶段发射机才能使用这种机制。无论是否使用该机构，都应是可编程的。当使用这种机制时，数据时间量子m(D)的分频器的值应该是1或2。该实现应能补偿发射机至少两个数据位元时间的延迟。

PCS to PMA输出消息对应于收发机总线驱动输入引脚 RxD, PMA to PCS输入消息对应于收发机总线比较器输出引脚 TxD。

发射机延迟补偿机制定义了一个二次采样点SSP。使用时，发射机应忽略采样点检测到的比特错误。在SSP处，接收的比特值应与(延迟的)传输的比特值进行比较。如果在SSP检测到比特错误，发射机应在随后的采样点对该比特错误作出反应。比特错误检测应为那些比特位在数据阶段的结束被禁用，在那里比特位的ssp将在以下仲裁阶段。

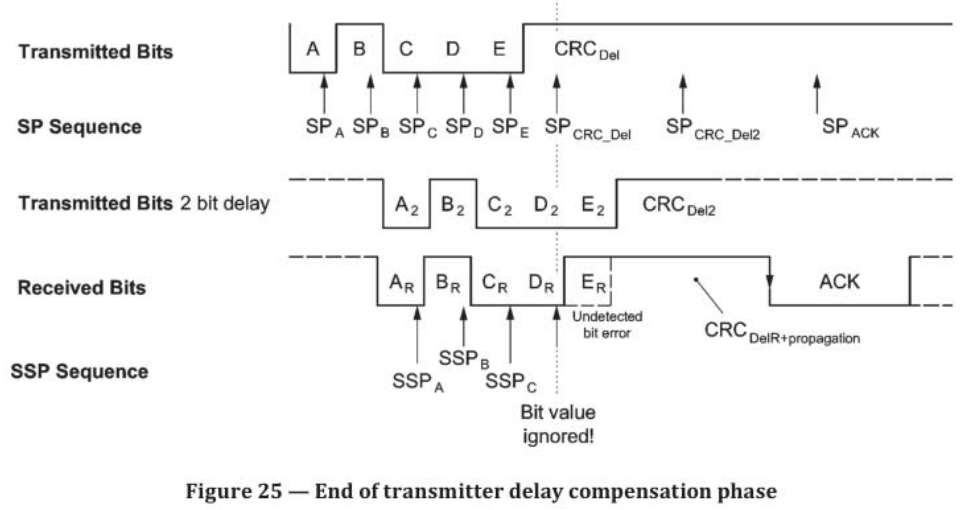


SSP的位置应该由它从比特时间开始的距离来指定。此SSP位置的配置范围应至少为0到63最小时间量程。SSP位置要么设置为一个固定值，要么设置为一个从实际发射机延迟测量得到的值。在每个传输帧中，从FDF位到res位的隐性到显性边缘处测量发送端延迟。将测量值应用于SSP位置将在同一帧的数据阶段完成。当发射机开始发送发送输出处的主要res位时，计数器将启动。该计数器应每最小时间量子加1，直到在接收输入处检测到主信号为止;然后计数器将停止。计数器值是测量的发射机延迟时间。当SSP位置由延迟测量得出时，该位置应为测量的发射机延迟值和可配置的SSP偏移量之和。

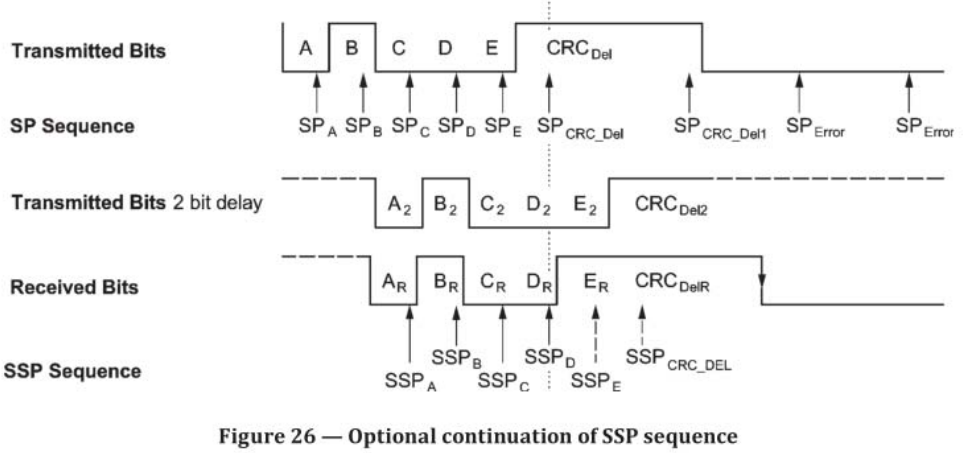
比特流[A, B,...K]图24显示了发射机延迟与SSP位置的关系。在这个例子中，发射机延迟几乎是数据位的两倍长。因此，SSP被放置在发射机延迟到该比特位开始后的3个数据比特位的范围内;输入信号预计已经稳定的点。在这种情况下，接收到的比特流与延迟两个数据位的发送比特流进行比较:在SSPA，接收到的比特流AR与延迟发送的比特流Az进行比较，以此类推。

可选地，如果SSP位置值为奇数，且数据时间量子的分频器值为2，则SSP位置可以用SSP位置值除以2并将结果向下取整来确定。

图25显示了带有发射机延迟补偿的数据阶段的结束。SSP序列在发射机的(第一个)CRC定界位的采样点停止，在那里数据阶段结束。如果在采样点忽略实际接收到的比特值，则SSP序列停止。在该示例中，CRC定界符之前的最后一个CRC序列比特被局部干扰(接收比特Er被认为是隐性的，但传输时是显性的)。在接收到主ACK之前，发送端容忍第二个隐性CRC定界符位。如果比特E的扰动不受局部限制，则会有一个EF而不是ACK。



CAN实现可以选择继续SSP序列超过数据阶段的限制，检查传输节点的本地错误，如图26所示。



## 11.4 AUI规范

### 11.4.1通用

附件单元接口是指在ISO 11898的本部分中指定的PCS与ISO 11898的其他部分或其他ISO标准(如ISO 11992-1)中指定的PMA之间的接口。

### 11.4.2 PCS到PMA消息

#### 11.4.2.1输出信息

当PMA子层收到来自MAC子层的Output\_Unit时，PCS应向PMA子层发送输出消息。输出消息使PMA发送一个显性或隐性比特。

#### 11.4.2.2 Bus\_off消息

当PCS收到来自supervisor的bus\_off\_request时(见12.1)，应向PMA子层发送bus\_off消息。

#### 11.4.2.3 Bus\_off\_release消息

当PCS收到来自supervisor的bus\_off\_release\_request时(见12.1)，PCS应向PMA子层发送bus\_off\_release消息。

#### 11.4.2.4 FD\_Transmit消息

当pc收到来自MAC子层的FD\_Transmit消息时，就向PMA子层发送一个FD\_Transmit消息。此消息对于启用FD的实现是可选的。

#### 11.4.2.5 FD\_Receive消息

当pc从MAC子层接收到FD\_Receive时，它将发送一个FD\_Receive消息给PMA子层。对于启用fd的实现，此消息是可选的。

### 11.4.3 PMA to PCS消息

#### 11.4.3.1输入消息

当PMA从介质接收到比特时，PMA子层应向pc发送输入信息。输入信号向pc表明显性或隐性位的到来。

# 12主管FCE描述

## 12.1故障限制

### 12.1.1目标

故障限制的目的是即使在存在故障节点的情况下仍能保持数据传输网络的高可用性。因此，故障限制策略应在以下情况下被证明是可靠的:

A)区分暂时性错误和永久性故障;

B)故障节点的定位与关闭。

### 12.1.2策略

所有节点应包括一个发送错误计数器和一个接收错误计数器。发送错误计数器应记录传输期间的错误数，接收错误计数器应记录接收帧期间的错误数。

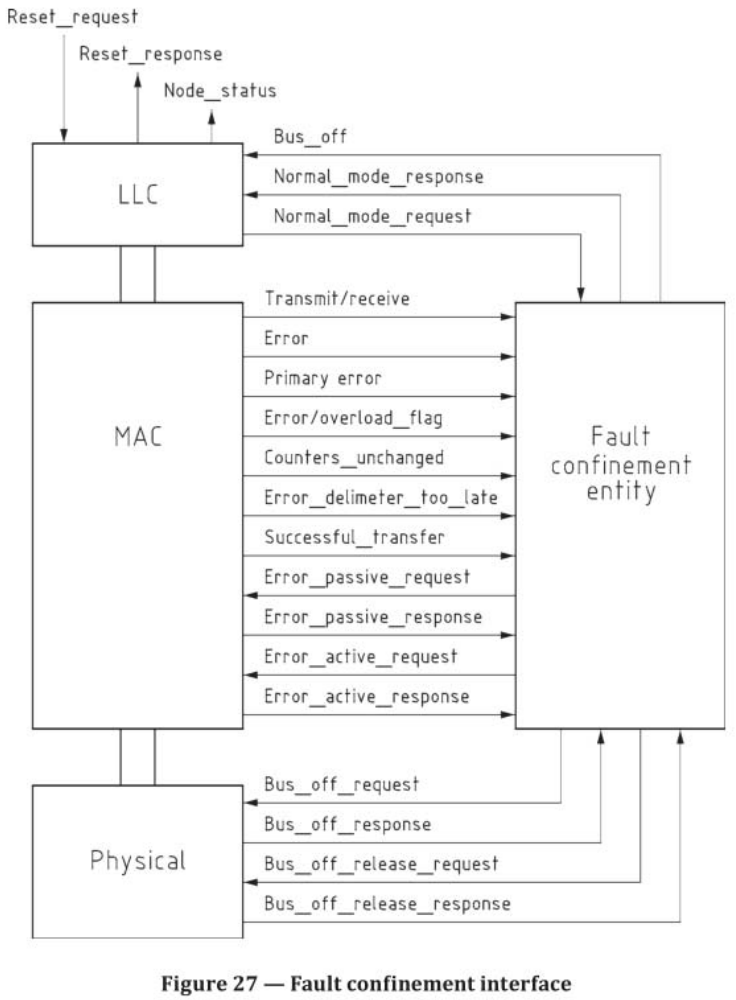
当正确发送或接收帧时，计数器将减少。当帧发送或接收有错误时，计数器的增加应该大于没有错误时计数器的减少。计数器加/减的比例取决于总线上无效/有效帧的可接受比例。在任何时候，错误计数器的水平反映了先前错误的相对频率。

根据预先确定的计数器值，节点在错误情况下的行为将被修改。从禁止发送错误标志到取消帧，到关闭经常发送无效帧的节点。

### 12.1.3故障约束接口规范

#### 12.1.3.1描述

故障约束界面如图27所示。



#### 12.1.3.2 LLC子层/FCE接口

FCE和LLC子层之间交换的消息应如表9和表10所示。

表9

|  |  |
| --- | --- |
| Message | 描述 |
| Normal\_Mode\_Request | 重置FCE到初始状态(当FCE进入初始状态时，发送错误计数器和接收错误计数器的值设置为零) |

表10

|  |  |
| --- | --- |
| Message | 描述 |
| Normal\_Mode\_Response | 对Normal\_Mode\_Request的响应。 |
| Bus\_Off | 表示该节点处于off状态。 |

#### 12.1.3.3 MAC子层/FCE接口

FCE和MAC子层之间的消息交换应如表11和表12所示。

表11

|  |  |
| --- | --- |
| Message | 描述 |
| Transmit/receive | 节点当前的传输模式。 |
| Error | 表示MAC子层检测到错误 (bit error, stuff error, CRC error, form error, ACK error) |
| primary\_error | MAC子层在发送错误标志后检测到主比特的信号(表示MAC子层检测到的是主错误，而不是由另一个节点的错误标志引起的错误) |
| Error/overload flag | 表示MAC子层正在发送错误标志或过载标志。 |
| Counters\_unchanged | 表示FCE计数器保持不变(由于特殊情况;参见12.1.4.2)中的规则c)。 |
| Error\_delimiter\_too\_late | 表示MAC子层等待错误定界符的时间过长。在发送错误标志后，每次接收到8个连续的主比特序列后，都会设置该信号。 |
| Successful\_transfer | 发送/接收成功。 |
| Error\_passive\_response | 节点被设置为错误被动状态。 |
| Error\_active response | 节点重新设置为错误活动状态。 |

表12

|  |  |
| --- | --- |
| Message | 描述 |
| Error\_passive\_request | 请求将节点设置为错误被动状态。 |
| Error\_active\_request | 请求将节点设置为错误活动状态 |

#### 12.1.3.4 PL/FCE接口

FCE和PL之间交换的消息应如表13和表14所示。

表13

|  |  |
| --- | --- |
| Message | 描述 |
| Bus\_off\_request | 请求从总线断开该节点。 |
| Bus\_off\_release\_request | 请求将节点设置为正常的发送/接收节点。 |

表14

|  |  |
| --- | --- |
| Message | 描述 |
| Bus\_off\_response | 响应Bus\_off\_request |
| Bus\_off\_release\_response | 响应Bus\_off\_release\_request |

### 12.1.4故障限制规则

#### 12.1.4.1描述

对于故障限制，节点可以处于以下三种状态之一，具体取决于错误计数器的级别(见6.13和6.15)。

— error-active;

— error-passive;

— bus-off.

#### 12.1.4.2错误计数

错误计数器应根据以下规则进行修改(在给定的帧传输期间可能应用多个规则)。

a)当接收端检测到错误时，接收错误计数器应加1，除非检测到的错误是在发送主动错误标志或过载标志期间的比特错误。

b)当接收端发送错误标志后检测到主频位作为第一个比特时，接收错误计数器加8。

c)当发送器发送错误标志时，发送错误计数器加8。

例外1:

如果发送端是错误被动的，并且由于没有检测到显性ACK而检测到ACK错误，并且在发送其被动错误标志时没有检测到显性比特。

例外2:

如果发送端因为仲裁过程中发生了材料错误而发送一个错误标志，此时填充位应该是隐性的，并且被发送为隐性的，但被监控为显性的。

在异常1和异常2中，传输错误计数器保持不变。

d)如果发射机在发送主动错误标志或过载标志时检测到比特错误，则发送错误计数器加8。

e)如果接收方在发送主动错误标志或过载标志时检测到比特错误，则接收错误计数器加8。

f)任何节点在发送主动错误标志、被动错误标志或过载标志后，应容忍多达7个连续的显性比特。在检测到14个连续的显性比特后(在有主动错误标志或过载标志的情况下)，或在无源错误标志之后检测到8个连续的显性比特后，在每增加8个连续的显性比特序列后，每个发送端应将其发送错误计数器加8，每个接收端应将其接收计数器加8。

g)在成功传输一帧后(收到ACK，直到EOF结束才检测到错误)，发送错误计数器减1，除非它已经为0。

h)在成功接收到一帧(到ACK时隙为止无错误接收，并且ACK比特发送成功)后，如果接收错误计数器在1到127之间，则将其减1。如果接收错误计数器是0，它应该保持在0，如果它大于127，那么它应该设置为119到127之间的值。

#### 12.1.4.3 error-active和error-passive状态的转换

如果节点的发送错误计数器或接收错误计数器超过127(接收错误计数器为7比特时为进位条件)，则监督者应请求MAC子层将相应节点设置为错误被动状态。

导致一个节点变为错误被动状态的错误条件将导致该节点发送一个主动错误标志。

当发送错误计数器和接收错误计数器都小于或等于127时(见图28)，错误被动节点将再次变为错误主动。

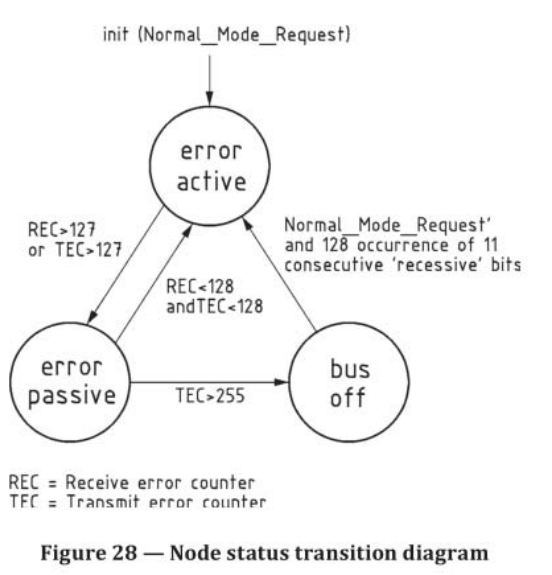
当节点的接收错误计数器超过了127的被动错误限制时，该接收错误计数器的进一步增加将受到计数器宽度的限制。在下一次成功接收帧时(过渡到错误主动)，接收错误计数器应设置为低于错误被动限制的值[参见12.1.4.2中的错误计数规则h]。

#### 12.1.4.4总线退出管理

如果节点的发送错误计数器大于255(8位发送错误计数器为进位条件)，则监督者应请求PL将节点设置为bus-off状态。

处于bus-off状态的节点对总线没有影响。它不应发送任何帧或确认DFs或RFs。结点是否接受来自总线的DFs取决于实现。

在重新启动请求时，处于bus-off状态的节点将集成到CAN通信中(见10.9.4)，在监测到总线上128次空闲情况(见4.28)后(见图28)，节点的错误计数器都设置为零，并可能变为错误活跃(不再断开总线)。



### 12.1.5网络启动

如果在网络启动过程中，只有一个节点在线，并且该节点发送了一些帧，那么它将得不到ACK。在这种情况下，它将检测到错误并重复帧。根据12.1.4.2例外1中的规则c)，它可以成为错误被动，但不能bus-off。

一个已经被关闭的节点将被重新打开

- 在开始传输之前，与已有的节点同步，以集成到总线通信中(参见4.29)。当11个隐性比特位等价于:

ACK分隔符+ EOF +间歇符

或

检测到错误/过载分隔符 + 间歇符;

- 如果当前没有其他节点可用，请等待其他节点，不要成为bus-off。

## 12.2总线故障管理

在正常运行过程中，可能会发生一些总线故障，影响总线运行。这些故障和由此产生的网络行为应符合所使用的PMA。