

# Classic CAN 2.0 总线网络“负载率”计算

## 1 Classic CAN帧长度

## 2 Interframe space帧间空间的介绍

### 2.1 Intermission间歇

### 2.2 Bus idle总线空闲

### 2.3 Suspend transmission

## 3 Classic CAN 2.0总线网络“负载率”计算

### 3.1 单个帧的“负载率”

### 3.2 通过CAN DBC/Arxml来计算一个CAN网络的理论“负载率”

## 4 拓展

## 1 Classic CAN帧长度

不管是**Classic CAN Standard Frame**还是**Classic CAN Extended Frame**，其帧结构都由以下7个段组成：

- SOF 帧起始;
- arbitration field 仲裁段;
- control field 控制段;
- data field 数据段;
- CRC field;
- ACK field;
- EOF.



**Classic CAN Standard Frame标准帧**（不考虑位填充）共：108Bit

帧起始（1bit）、仲裁段（12bit）、控制段（6bit）、数据段（8×8bit）、循环冗余码段（16bit）、应答段（2bit）和帧结束（7bit）

**Classic CAN Extended Frame扩展帧**（不考虑位填充）共：128Bit

帧起始（1bit）、仲裁段（32bit）、控制段（6bit）、数据段（8×8bit）、循环冗余码段（16bit）、应答段（2bit）和帧结束（7bit）

## 2 Interframe space帧间空间的介绍

Data frame数据帧和remote frame远程帧应通过一个称为**interframe space帧间空间**的bit field与前面的帧分开，不管是什么类型的帧(data frame数据帧、remote frame远程帧、error frame错误帧、overload frame过载帧)。

注意：

- overload frame过载帧和error frame错误帧的前面不应有interframe space帧间空间；
- 多个overload frame过载帧之间也不应有interframe space帧间空间。

interframe space帧间空间应包含bit field：intermission和bus idle，并对作为前一帧的发送方的error-passive node暂停发送(见图2-1和图2-2)。

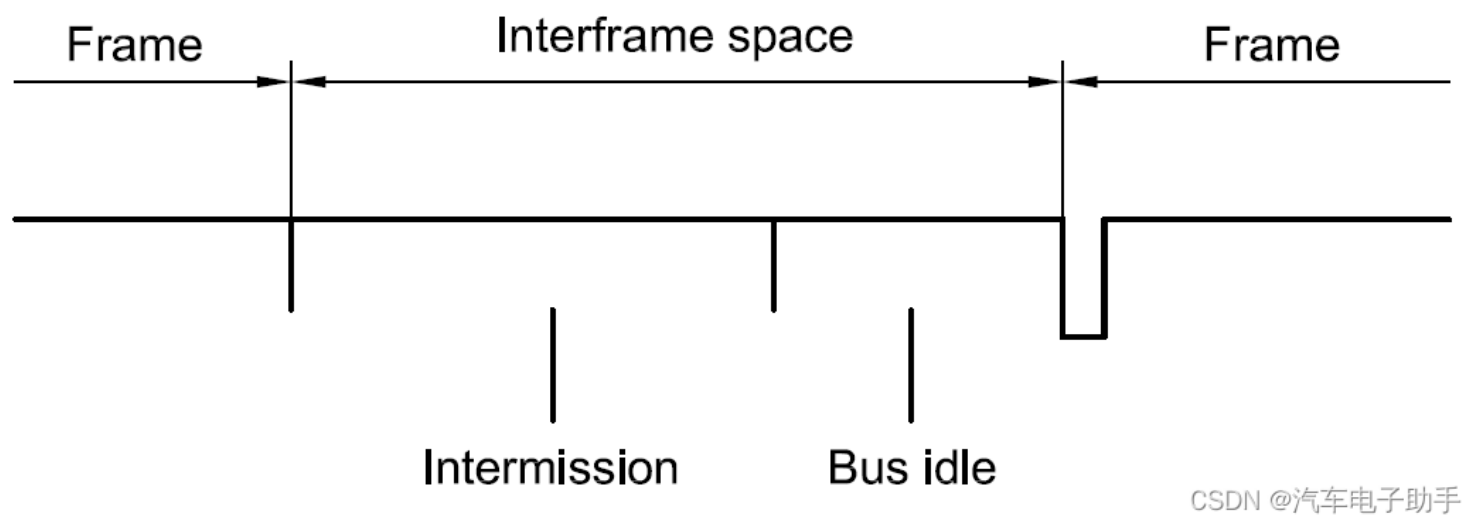


图2-1 非前一帧的error-passive或receiver的节点的帧间空间

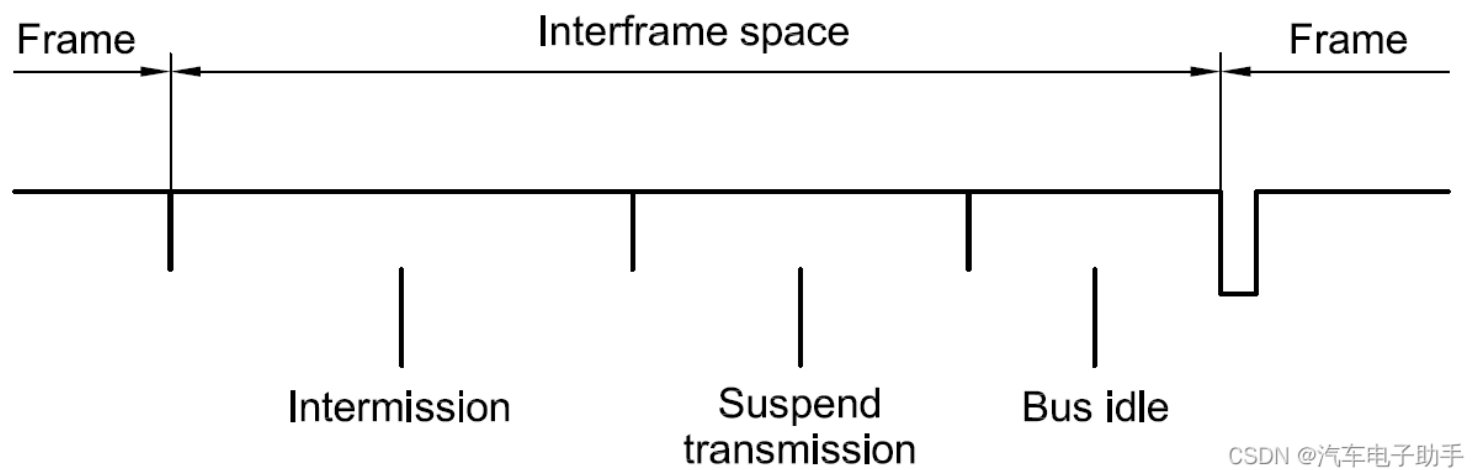


图2-2 作为前一帧发送方的error-passive node的Interframe space

### 2.1 Intermission间歇

intermission field应包括三个**隐性位**。在intermission期间，任何节点不得开始发送**data frame数据帧**或**remote frame远程帧**。只允许发出overload条件的信号。

在intermission期间的第三位检测到总线上的**显性位**应解释为SOF。

## 2.2 Bus idle总线空闲

**bus idle总线空闲**期间的长度可以是任意的。总线应识别**bus idle总线空闲**，任何节点都可以进入总线进行发送。在发送另一帧的过程中，等待发送的帧，应在intermission后的第一Bit开始。

在**bus idle总线空闲**期间，对总线上一个**显性位**的检测应被解释为SOF。

## 2.3 Suspend transmission

一个error-passive node，如果是前一帧的发送方，在开始发送另一帧之前，应在intermission后发送8个**隐性位**后。

# 3 Classic CAN 2.0总线网络 “负载率” 计算

通过第2章我们可以知悉，CAN总线网络正常的发送过程中，帧与帧之间至少有3个**隐性位**。

## 3.1 单个帧的 “负载率”

对**Classic CAN Standard Frame标准帧**来说，发送一帧实际长度（**不考虑位填充；帧间隔3Bit；DLC = 8**）：

$108+3=111\text{Bit};$

比特率/波特率	一个Bit的位时间	单个帧的“负载率”
250 Kbps	4000纳秒	$((111 * 4000\text{纳秒}) / 1\text{秒}) * 100\% = 0.0444 \%$
500 Kbps	2000纳秒	$((111 * 2000\text{纳秒}) / 1\text{秒}) * 100\% = 0.0222 \%$
1 Mbps	1000纳秒	$((111 * 1000\text{纳秒}) / 1\text{秒}) * 100\% = 0.0111 \%$

对**Classic CAN Extended Frame扩展帧**来说，发送一帧实际长度（**不考虑位填充；帧间隔3Bit；DLC = 8**）：

$128+3=131\text{Bit};$

比特率/波特率	一个Bit的位时间	单个帧的“负载率”
250 Kbps	4000纳秒	$((131 * 4000\text{纳秒}) / 1\text{秒}) * 100\% = 0.0524 \%$
500 Kbps	2000纳秒	$((131 * 2000\text{纳秒}) / 1\text{秒}) * 100\% = 0.0262 \%$

1 Mbps	1000纳秒	$((131 * 1000\text{纳秒}) / 1\text{秒}) * 100\% = 0.0131 \%$
--------	--------	---

### 3.2 通过CAN DBC /Arxml来计算一个CAN网络的理论“负载率”

如果已知一个CAN总线网络所有CAN ID的DBC/ Arxml，并且知道它们的发送方式，我们可以粗略地估算出CAN网络的“负载率”：CAN总线负载率是各个帧占用总线带宽百分比之和。

下面以比特率/波特率：500kbps为例（不考虑位填充；帧间隔3Bit；DLC = 8）：

Num	CAN ID	发送周期	帧类型	DLC	对应帧一秒内的负载率
1	0x115	10ms	标准帧	8	$((1000\text{ms} / 10\text{ms}) * (111 * 2000\text{纳秒}) / 1\text{秒}) * 100\% = 2.22\%$
2	0x217	20ms	扩展帧	8	$((1000\text{ms} / 20\text{ms}) * (131 * 2000\text{纳秒}) / 1\text{秒}) * 100\% = 1.31\%$
3	0x3B4	50ms	扩展帧	8	$((1000\text{ms} / 50\text{ms}) * (131 * 2000\text{纳秒}) / 1\text{秒}) * 100\% = 0.524\%$
4	0x475	500ms	标准帧	8	$((1000\text{ms} / 500\text{ms}) * (111 * 2000\text{纳秒}) / 1\text{秒}) * 100\% = 0.0444\%$
...	...	...		...	...
CAN总线网络的理论“负载率”					4.0984 % +...

### 4 拓展

如果有一个已知的CAN Trace Log，我们也可以按照上述的方法，统计1秒以内发送的帧数，来粗略计算该1秒以内的Classic CAN负载率。