|  |  |  |
| --- | --- | --- |
| Article | Key Notes | Thoughts/comments |
| 具有优先级队列的最短最坏响应时间的饥饿避免CAN调度 | 核心：利用ID最高位设置反转低优先级，一次仲裁就可以反转  将11位消息ID分为10位静态ID字段（用于编码优先级）和1位动态ID字段（已知）  最坏情况响应时间(WCRT)：描述了一条消息的所有实例中的最大响应时间  可调度性分析：我们需要确保所有消息都是可调度的，换句话说，消息的WCRT小于其截止日期。  FIFO队列（FQ）先进先出，或优先级队列（PQ） | 将这一个反转的思想运用到自己观点的一部分 |
| 一种无饥饿控制器局域网络优先级反转方案的分析与建模 | 核心：需要一个ID最高位和一个软件计数器相结合的算法，  计数器用来计算失败次数，一旦满足设置的失败次数的条件则自动的将标志位反转（默认反转前为1） | 软件计时器要学会使用，或者不使用，但是要换一种方法来描述 |
| 控制器局域网总线仲裁机制的改进 | 利用ID的最高有效位将消息分为静态和动态，其中静态具有更高的优先级  消息设置时间：当消息发送时，节点首先将消息标识符，数据字节和控制位加载到发送消息组合寄存器中。然后，该节点将数据传输到CAN协议引擎。协议引擎通过插入帧元素（例如开始和停止位以及帧间空间位）来创建实际帧。协议引擎还处理总线仲裁，循环冗余校验和计算，并查找传输错误[1]。因此，当从节点发送消息时，需要时间来准备要发送的下一条消息。发送两个连续的消息之间存在一些空闲时间。 | 利用ID最高位将消息区分为关键消息（静态）和非关键消息（动态），然后利用混合算法来使用，借鉴了flexray通信方式，其中提出的消息设置时间有的莫名其妙 |