

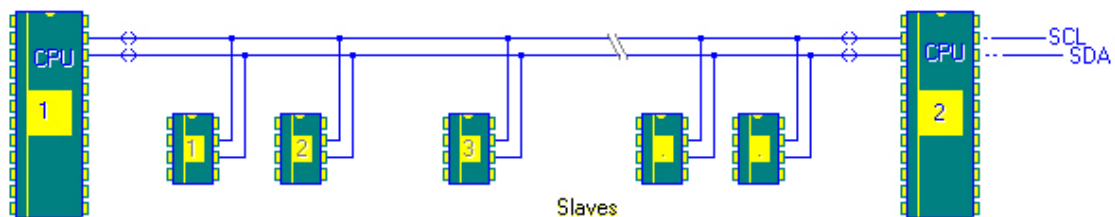
## I2C 버스 중재

이제까지 우리는 마스터의 관점에서 그리고 버스 상의 단 한 개 마스터를 사용하는 것에서 버스 실행을 살펴보았습니다.

I2C 버스는 원래 멀티-마스터 버스로 개발된 것입니다. 이것은 전송을 시작하는 한 개 이상의 디바이스가 시스템에서 활동할 수 있다는 것을 뜻합니다.

버스에서 한 개의 마스터만 사용할 때는, 슬레이브 디바이스에 기능 장애가 있거나 SDA/SCL 버스 라인을 포함하는 결함 조건이 없다면 실질적인 데이터 파괴의 위험이 없습니다.

이 상황은 2 개 MCU 에 따라 변합니다:



MCU 1 이 **start condition** 을 보내고 주소를 전송할 때, 모든 슬레이브들은 (이 때 마찬가지로 하나의 슬레이브로 간주되는 **MCU2** 도 포함) 받아들입니다. 이 주소가 **CPU2** 의 주소와 맞지 않으면, 이 디바이스는 버스가 **stop condition** 후에 다시 유희가 될 때까지 어떤 활동에도 참가하지 않아야 합니다.

두 개의 MCU 들이 버스상에서 진행되고 있는 것을 모니터하는 한(start 와 stop) 그리고 최종 전송 명령이 STOP 이 아니었기 때문에 처리가 계속되고 있는 것을 그들이 알고 있는 한, 문제는 없습니다.

MCU 중의 하나가 START 조건을 놓치고 그 버스가 여전히 유희 상태라고 생각하고 있다가, 또는 실제 시나리오에서 매우 잘 발생할 수 있는 리셋에서 빠져나와 버스에서 교신을 시작하기를 원한다고 가정해 봅시다. 이것은 문제를 야기할 수 있습니다.

버스에서 어떤 다른 디바이스가 전송하고 있다면, 우리는 그것을 어떻게 알 수 있을까요?

다행히도, 물리적 버스 조직이 우리를 도와줍니다. 버스 구조는 wired AND 이므로 (한 디바이스가 라인을 low 로 끌면 이것은 low 를 유지), 사용자는 버스가 유향 상태인지 사용중인지 확인할 수 있습니다.

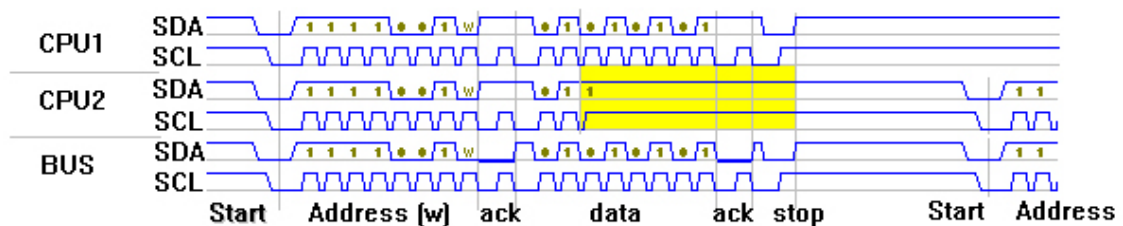
마스터가 라인의 상태를 **HIGH** 로 변경하면, 이것은 반드시 그 라인이 실제로 **HIGH** 로 갔는지 항상 확인해야 합니다. 만약 이것이 **low** 를 유지하면 이것은 버스가 사용 중이며 어떤 다른 디바이스가 그 라인을 **low** 로 끌어당기고 있다는 표시입니다.

따라서 일반적인 경험에 의한 방법으로: 마스터가 high 로 갈 수 있는 확실한 라인을 가질 수 없다면 이것은 중재에서 지며 off 로 돌아가서 전송을 시작하려는 다른 시도를 하기 전에 stop 조건이 보일 때까지 기다려야 합니다.

## 데이터 파괴의 위험은 어떻습니까?

이전 규칙이 마스터가 필요할 때 **SCL** 이나 **SDA** 를 high 로 가게 할 수 없을 때 중재에서 진다고 하므로, 이 문제는 존재하지 않습니다. 버스를 지배하는 것은 “0”을 보내는 디바이스입니다. 한 마스터는 라인들 중에서 high 로 가고, 다시 off 되는 것을 탐지할 수 없기 때문에 다른 마스터의 전송을 방해할 수 없으며, 이것이 그렇게 할 수 없는 다른 쪽 마스터라면, 이것도 동일하게 행동할 것입니다.

이러한 유형의 Back-off 조건은 두 개 마스터에 의해 전송되는 두 개의 레벨들이 동일하지 않은 경우에만 발생합니다. 그러므로 두 개의 MCU 들이 동시에 전송을 시작하는, 다음 그림을 살펴봅시다.



두 개 MCU 들은 주소 1111001 의 write 모드에 있는 슬레이브를 액세스하고 있습니다. 슬레이브는 이것을 확인 응답합니다. 지금까지, 두 개 마스터들은 그들이 버스를 “소유”하고 있다는 생각을 하고 있습니다.

이제 MCU1 은 01010101 을 슬레이브로 전송하고자 하며, 반면 MCU2 는 01100110 을 슬레이브로 보내기를 원합니다. 데이터 비트가 더 이상 맞지 않는 순간 (MCU 가 보내는 것이 버스에 있는 것과 다르기 때문) 그들 중 하나는 중재에서 지고 off 로 돌아갑니다. 확실한 건, 이는 버스에서 데이터를 얻지 못했던 MCU 입니다. 버스에 STOP 시점이 없는 한, 이것은 버스를 건드리지 않으며 SDA 와 SCL 라인들을 홀로 남겨둡니다 (yellow zone). STOP 이 발견되는 순간, MCU2 는 다시 전송을 시도할 수 있습니다.

위의 예에서 우리는 중재 상황에서 LOW 라인을 당기고 있는 마스터가 항상 중재에서 이긴다는 결론을 내릴 수 있습니다. 다른 마스터에 의해 low 로 당겨지고 있을 때 그 라인이 HIGH 가 되기를 원했던 마스터는 그 버스를 빼앗깁니다. 우리는 이것을 loss of arbitration 또는 back-off condition 이라 부릅니다.

MCU 가 중재에서 질 때, 이것은 버스에서 STOP 조건이 나타나기를 기다려야 합니다. 그러고 나면 앞선 전송이 완료되었다는 것을 알게 됩니다.