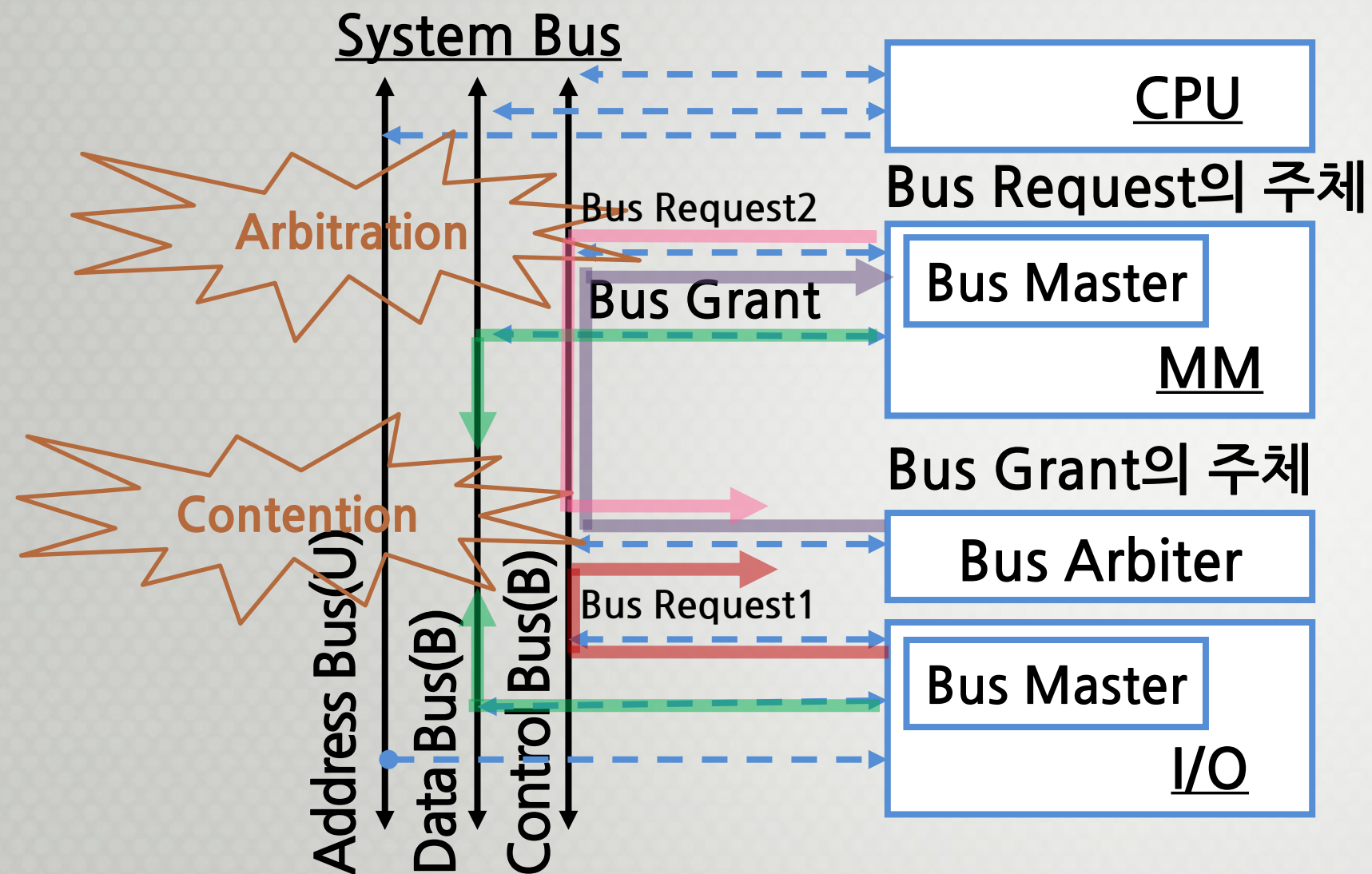


■ 왜 Bus 사용을 중재(Arbitration) 해야 할까? 구조는?

- **Bus Contention** : 두 개 이상의 구성요소가 Bus의 사용을 동시에 요구할 경우 발생하는 충돌
- **Arbitration** : Bus Contention 발생을 막기 위해 Bus 사용 우선권을 중재



■ 왜 Bus 사용을 중재(Arbitration) 해야 할까? 구조는?

① Bus Arbiter가 집중화 되어 있는가?

Bus Master가 직렬로 배치되어 있는가?, '병렬로 배치되어 있는가?' 에 따라 Arbitration Scheme이 분류됨

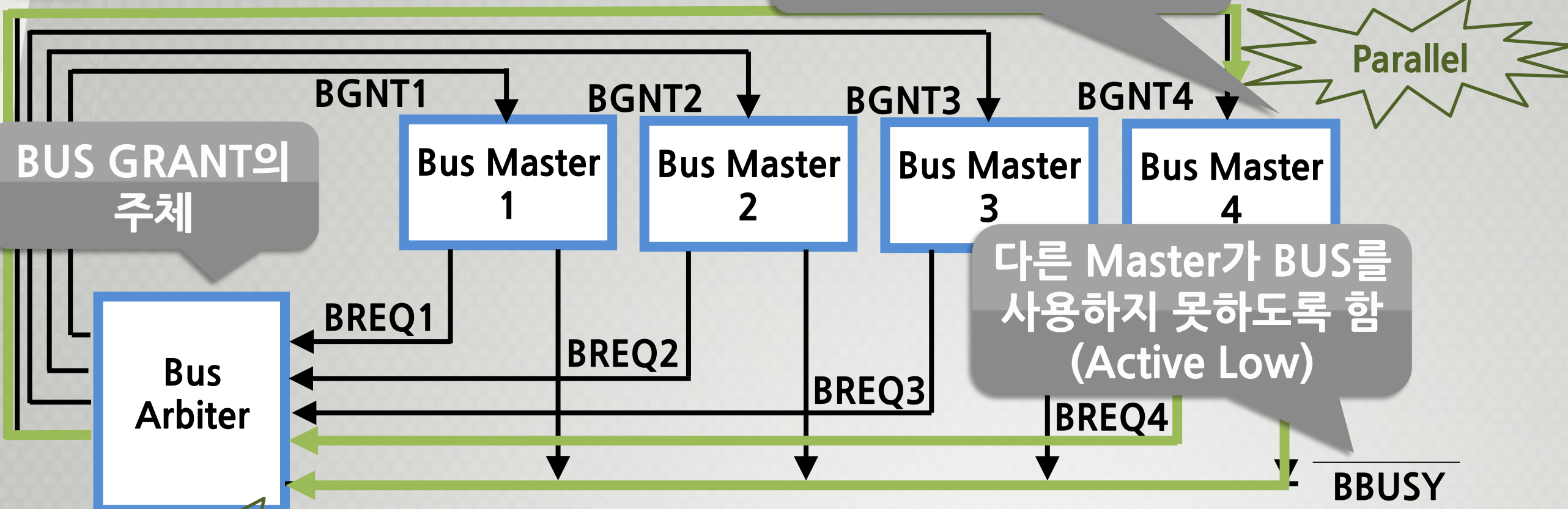
분류(①)]	분류(②)	Arbitration Scheme
Centralized	Parallel	Centralized Parallel Arbitration(CPA)
	Serial	Centralized Serial Arbitration(CSA)
Decentralized	Parallel	Decentralized Parallel Arbitration(DPA)
	Serial	Decentralized Serial Arbitration(DSA)
Polling		H/W Polling
		S/W Polling



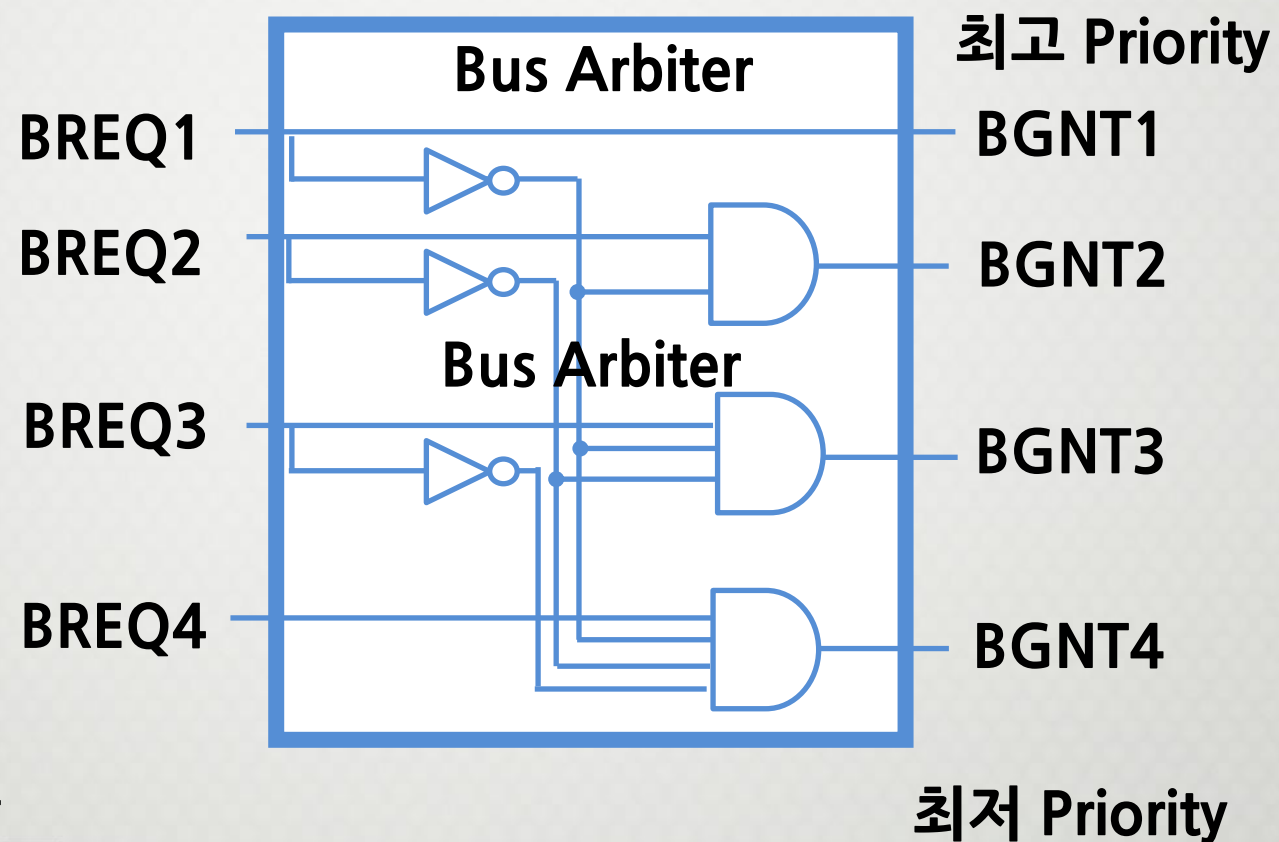
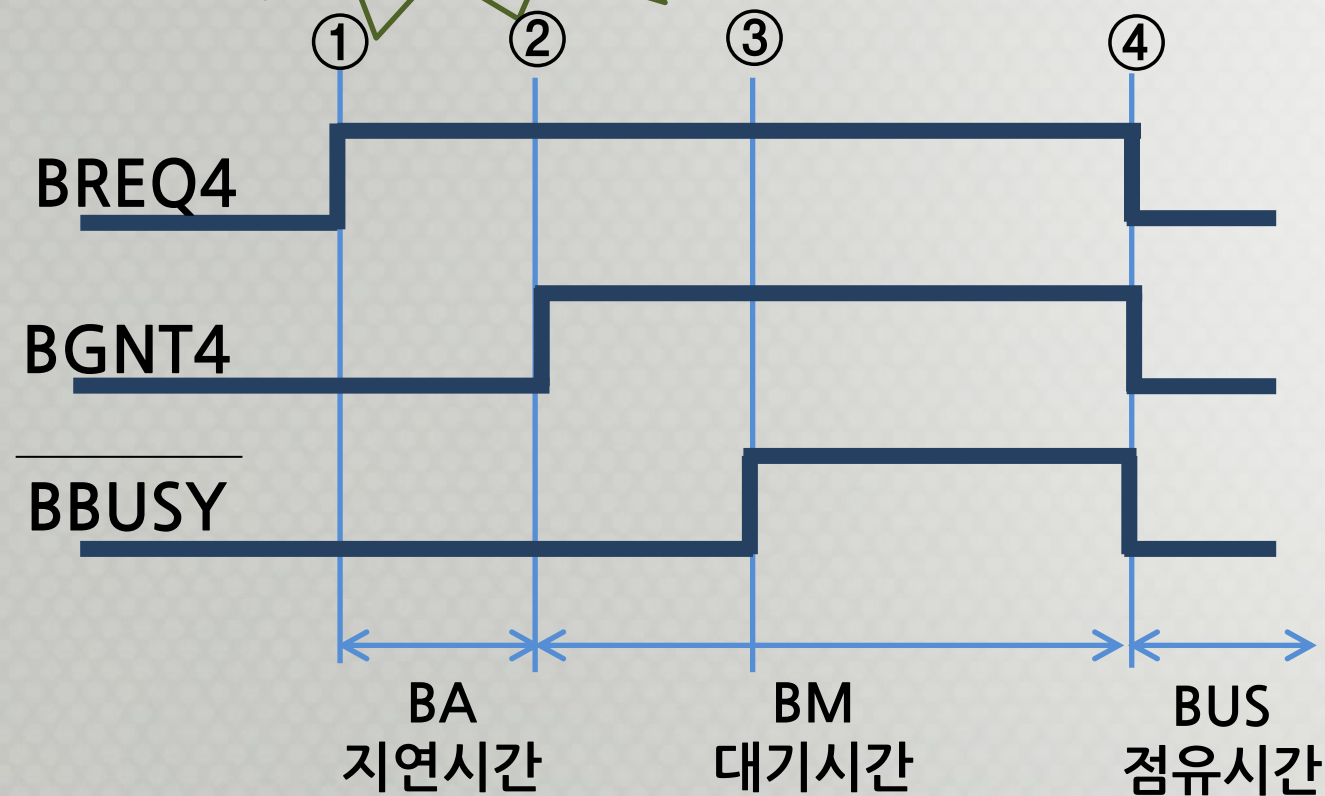
CPA

BUS REQUEST의 주체

BUS GRANT의 주체



Centralized



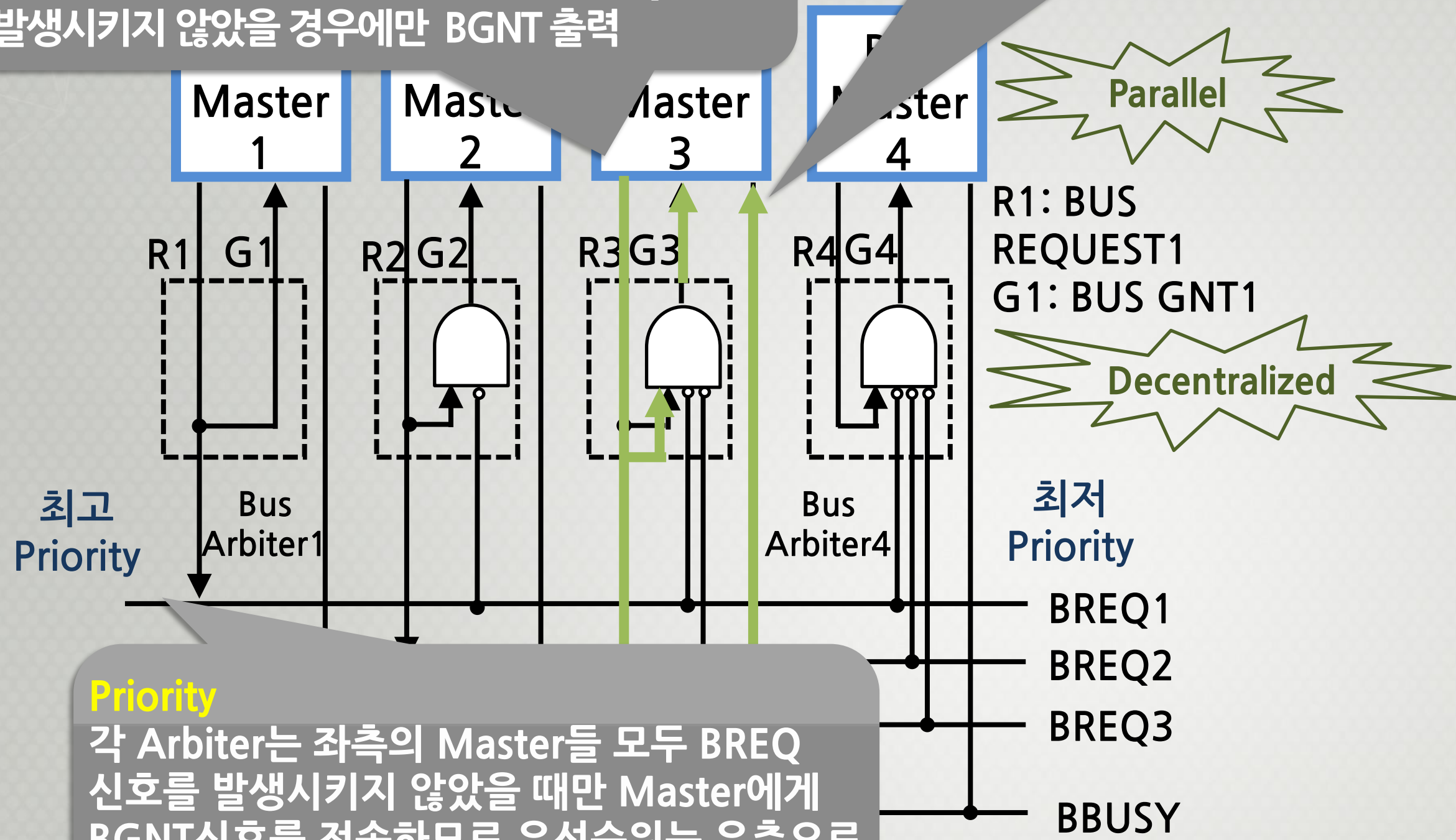
DPA

BREQ & BGNT

- 각 Master는 자신의 BREQ신호를 발생시킴
- 자신보다 우선순위가 높은 Master가 BREQ신호를 발생시키지 않았을 경우에만 BGNT 출력

BGNT

BGNT를 받은 Master는 BBUSY가 해제된 상태일 때 Bus 사용 시작



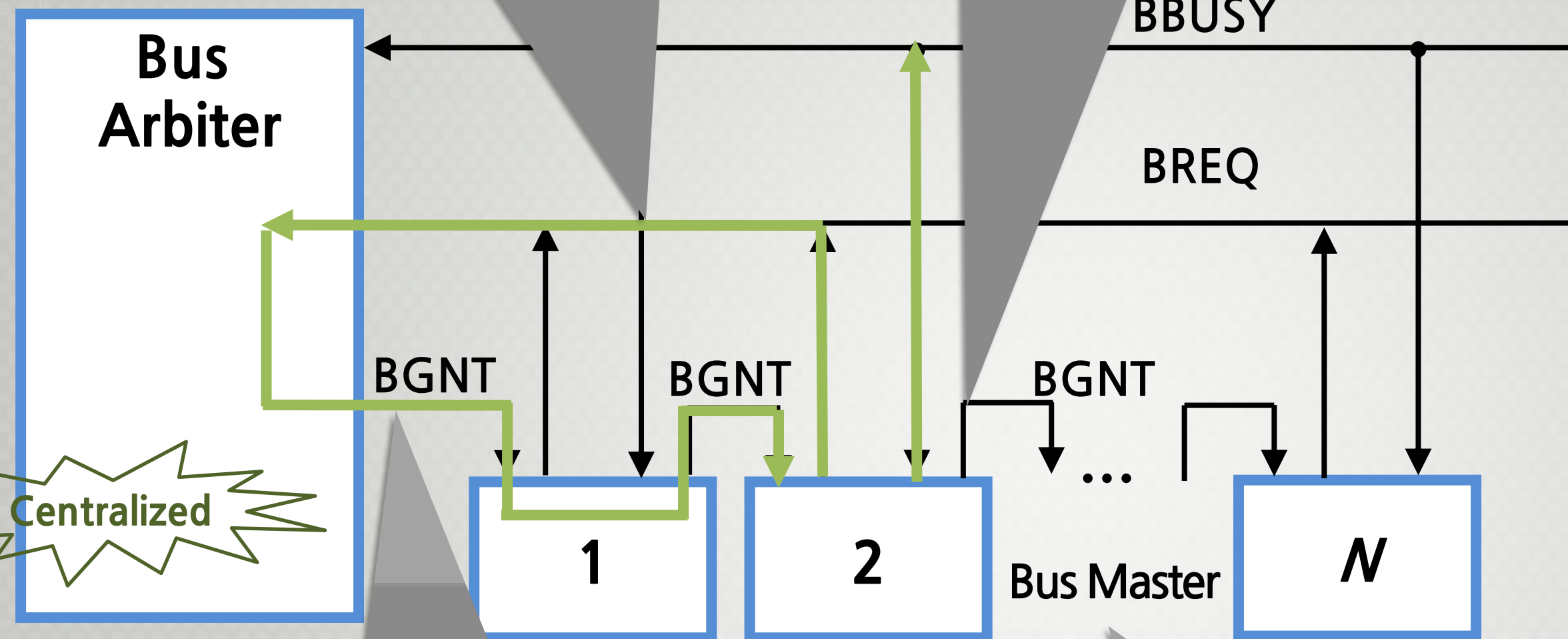
Priority

각 Arbiter는 좌측의 Master들 모두 BREQ 신호를 발생시키지 않았을 때만 Master에게 BGNT신호를 전송하므로 우선순위는 우측으로 갈수록 낮아짐

CSA(Daisy-Chain 기반)

Arbiter는 Master들 중 누군가로부터 Bus사용요청(BREQ, 공통선 연결)을 전송 받으며, BREQ를 발생시킨 Master가 누구인지는 모름

BREQ를 발생시킨 Master에게 도달하면 해당 Master는 전달을 멈추고 BBUSY를 활성화 시킨 후 Bus를 사용



Centralized

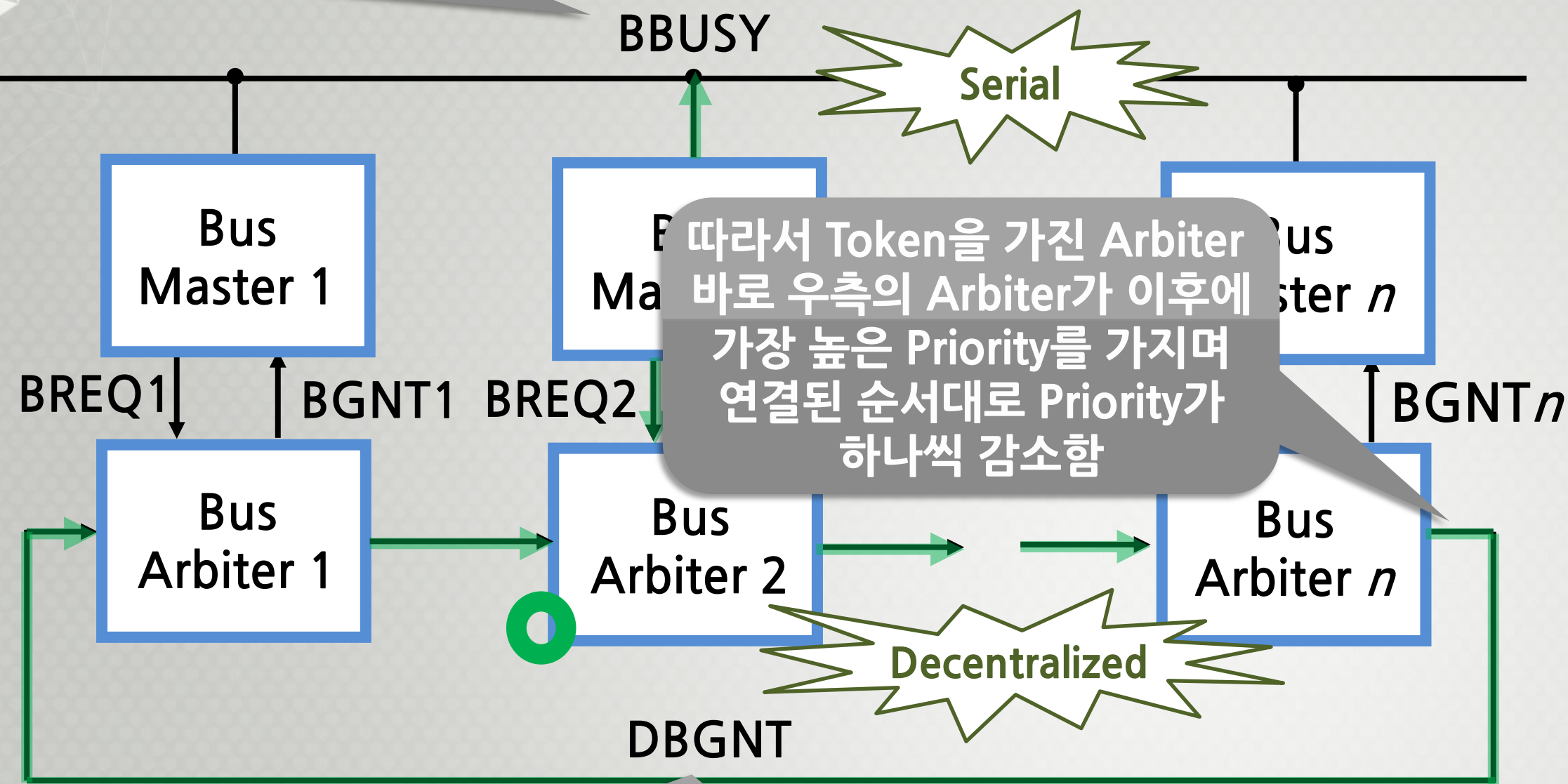
Arbiter는 Master들에게 Bus Priority(Token, BGNT)을 순서대로 전달 (Shift Right)

Serial

따라서 우선순위는 BGNT가 연결된 순서

■ DSA(Daisy-Chain 기반)

Token을 가진 Arbiter에 연결된 Master가
BREQ를 요구하면 Arbiter는 Master에게
BGNT 허용 → DBGNT를 우측 Master로 전달



기본적으로 Arbiter끼리 D(Daisy-Chained)
BGNT(Token, Bus Priority)돌리기를
수행(Circular Shift Right)