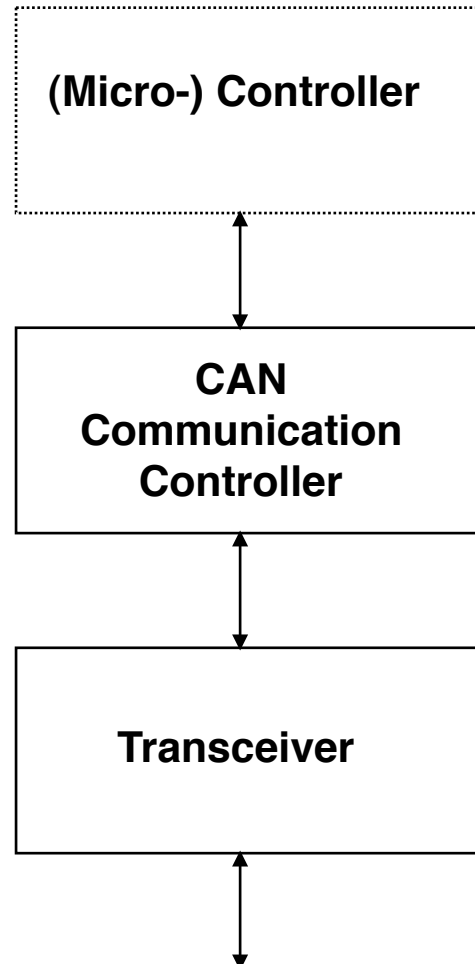


CAN Bausteine



- Stand-alone
- Microcontroller mit CAN-Komponente
- I/O-Bausteine (SLIO)
- Transceiver Bausteine

Full CAN Bausteine
Basic CAN Bausteine

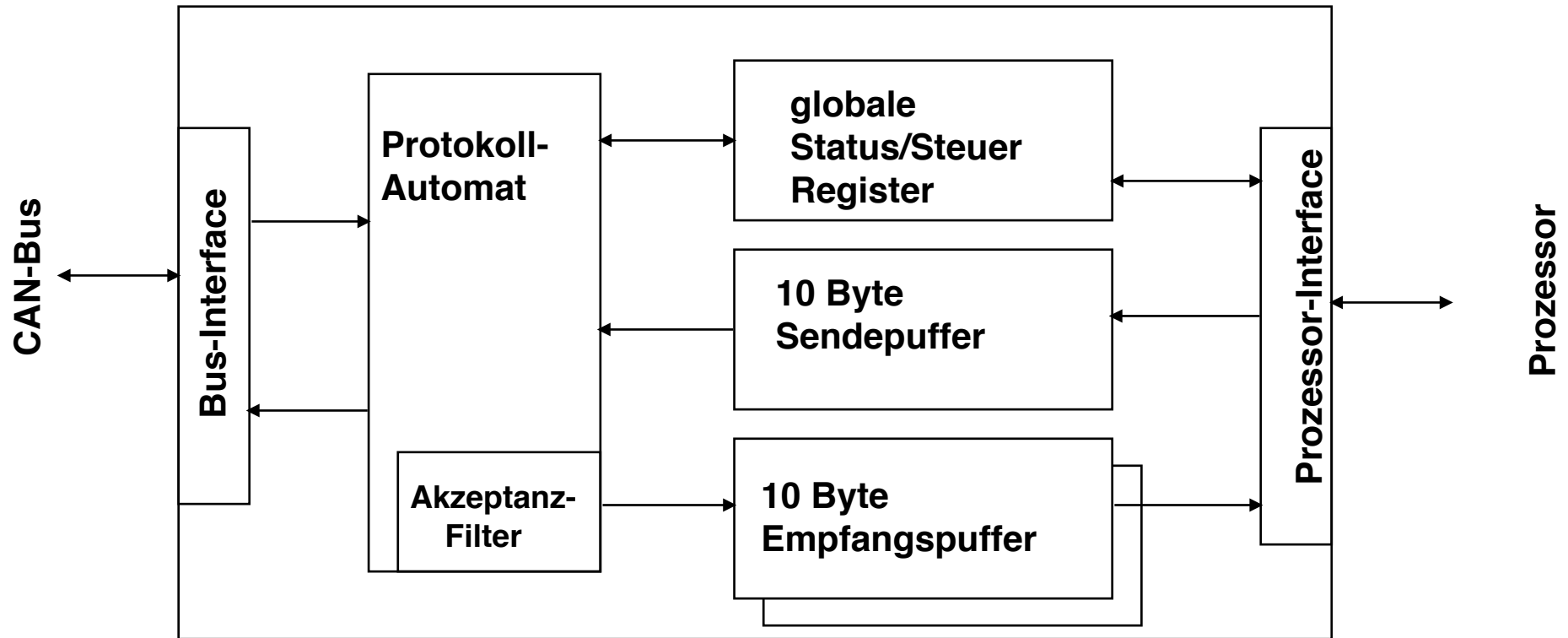


Aufgaben des CAN Controllers

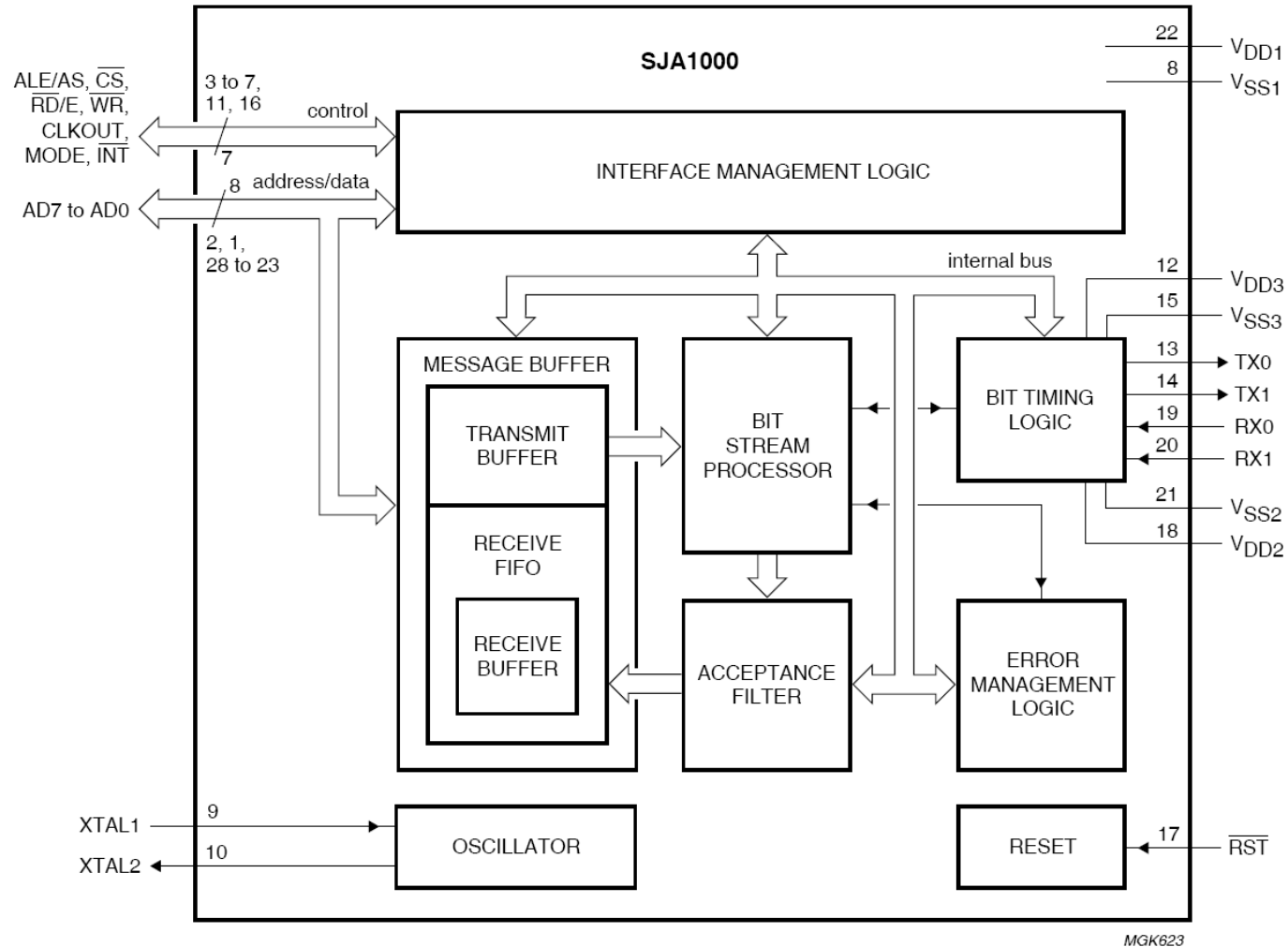
- **Busarbitrierung**
- **Serialisierung der zu sendenden Telegramme**
- **Assemblierung der empfangenen Telegramme**
- **Berechnung bzw. Überprüfung der Checksumme**
- **Filterung von Nachrichten**
- **Fehlererkennung und Fehlersignalisierung**
- **Bildung der CAN-Nachrichtenformate**
- **Einfügen bzw. Entfernen der zusätzlichen Bits beim Bit-Stuffing**
- **Erzeugen bzw. Überprüfen des Acknowledge-Bits**
- **Synchronisation des empfangenen Bitstroms**



Basic CAN



SJA1000 (Philips)



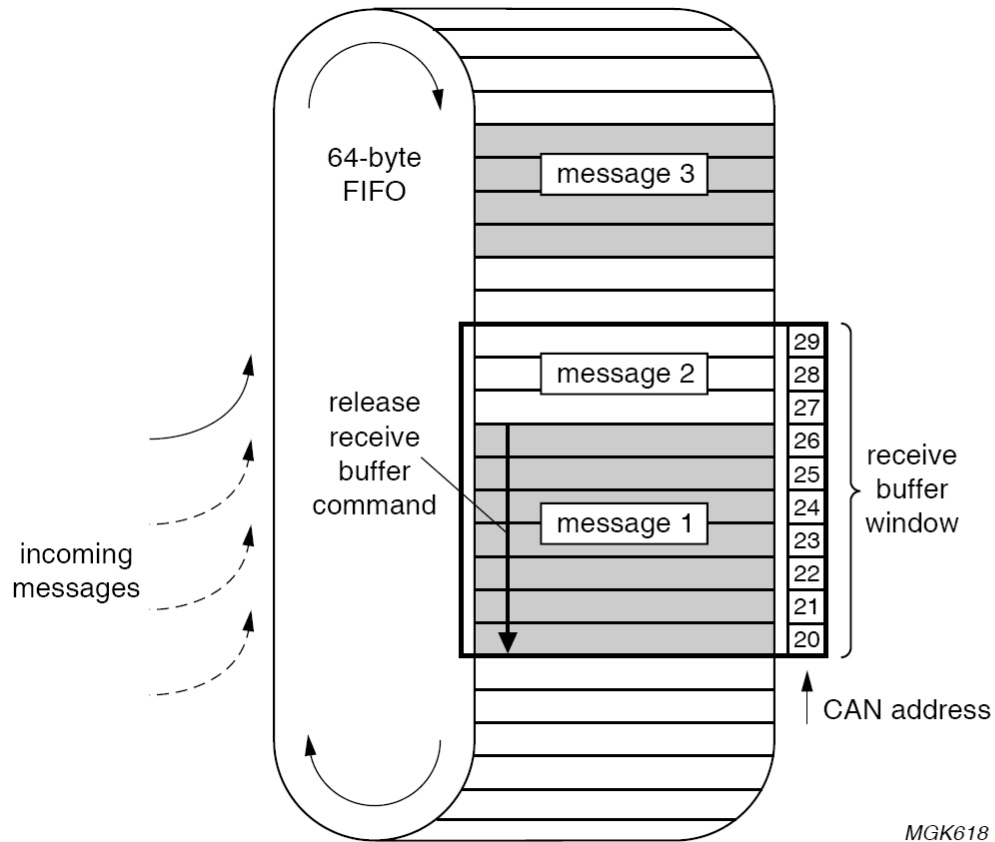
SJA1000 Transmit Buffer Layout

CAN ADDRESS	FIELD	NAME	BITS							
			7	6	5	4	3	2	1	0
10	descriptor	identifier byte 1	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3
11		identifier byte 2	ID.2	ID.1	ID.0	RTR	DLC.3	DLC.2	DLC.1	DLC.0
12	data	TX data 1	transmit data byte 1							
13		TX data 2	transmit data byte 2							
14		TX data 3	transmit data byte 3							
15		TX data 4	transmit data byte 4							
16		TX data 5	transmit data byte 5							
17		TX data 6	transmit data byte 6							
18		TX data 7	transmit data byte 7							
19		TX data 8	transmit data byte 8							

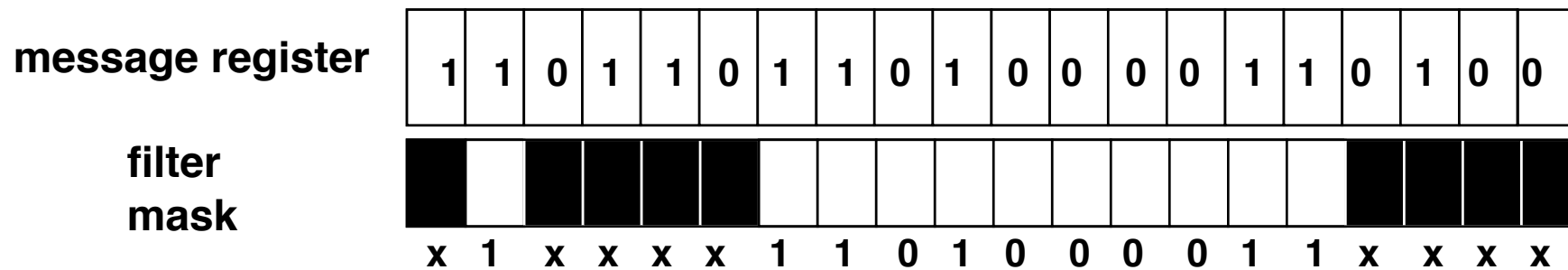
For extended CAN the Address range is extended accordingly to 4 Bytes of ID Descriptor.



SJA1000 (Philips)



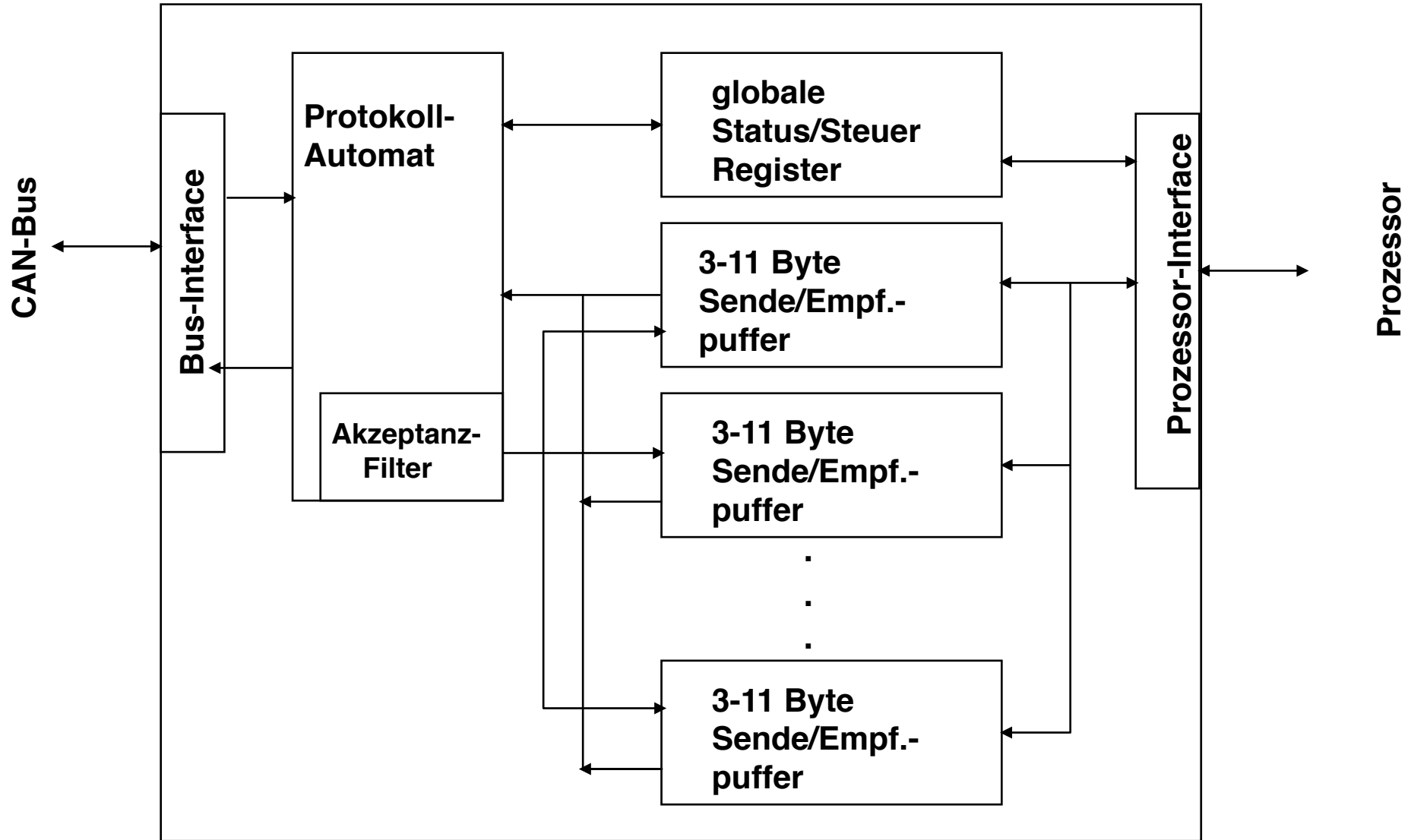
message filtering



Anzahl der Nachrichtenregister, Konfigurationsmöglichkeiten und Möglichkeiten der Nachrichtenfilterung sind abhängig vom verwendeten Kommunikationskontroller.

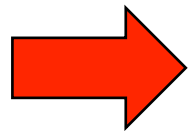


Full CAN



What CAN can't

- **All-or-nothing property under all single (crash/omission) fault conditions**
- **Consistent order of messages**
- **Temporal guarantees for message transmissions**



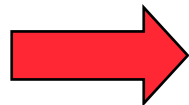
TTCAN
FTTCAN



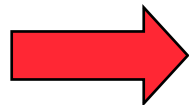
Consequences from the validation protocol

J. Rufino, P. Veríssimo, C. Almeida , L. Rodrigues: „Fault-Tolerant Broadcasts in CAN“, *Proc. FTCS-28, Munich, Germany, June 1998.*

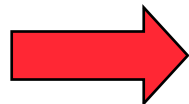
J. Kaiser, Mohammad Ali Livani: “Achieving Fault-Tolerant Ordered Broadcasts in CAN” *Proc. of the 3rd European Dependable Computing Conference, (EDCC-3), Prague, Sept. 1999*



inconsistent message duplicates



inconsistent omission faults



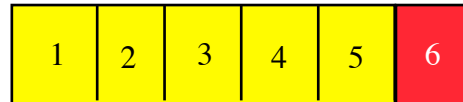
(potentially) unbounded latencies



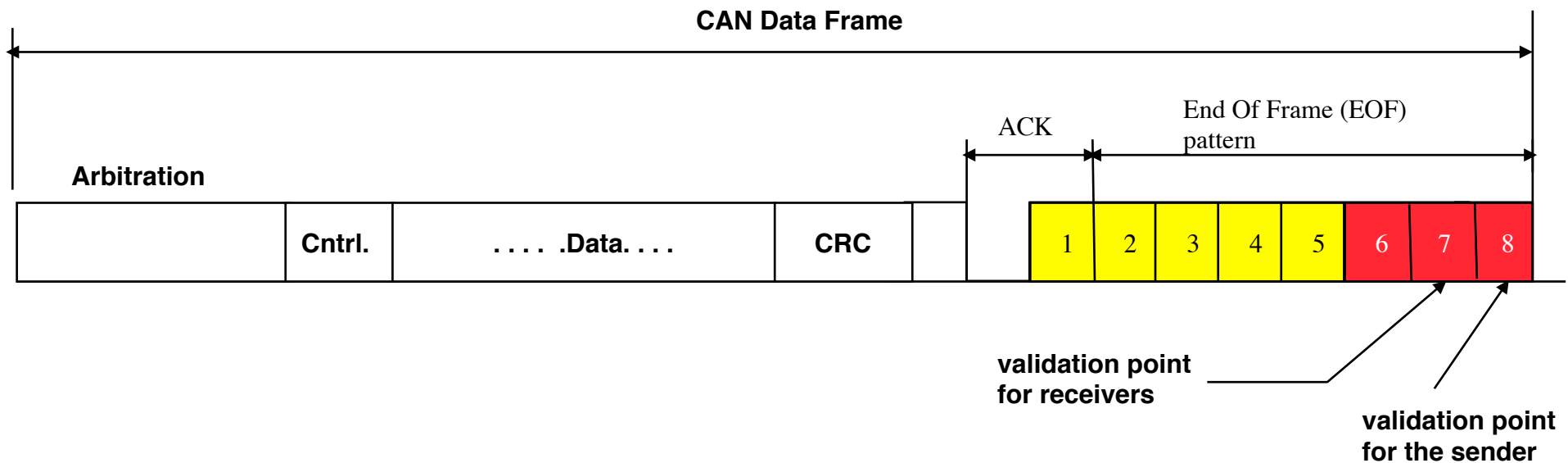
Error Detection and Error Signalling in CAN

The Case for Inconsistencies

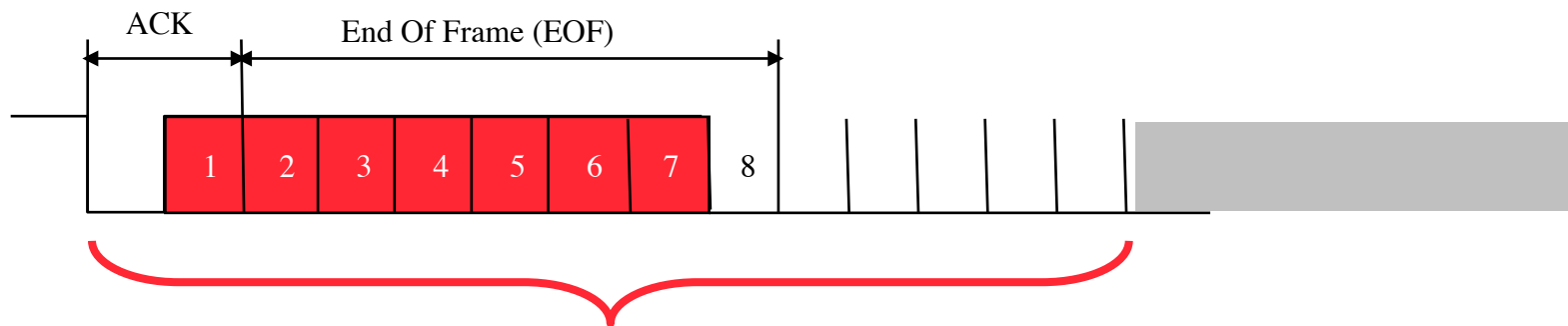
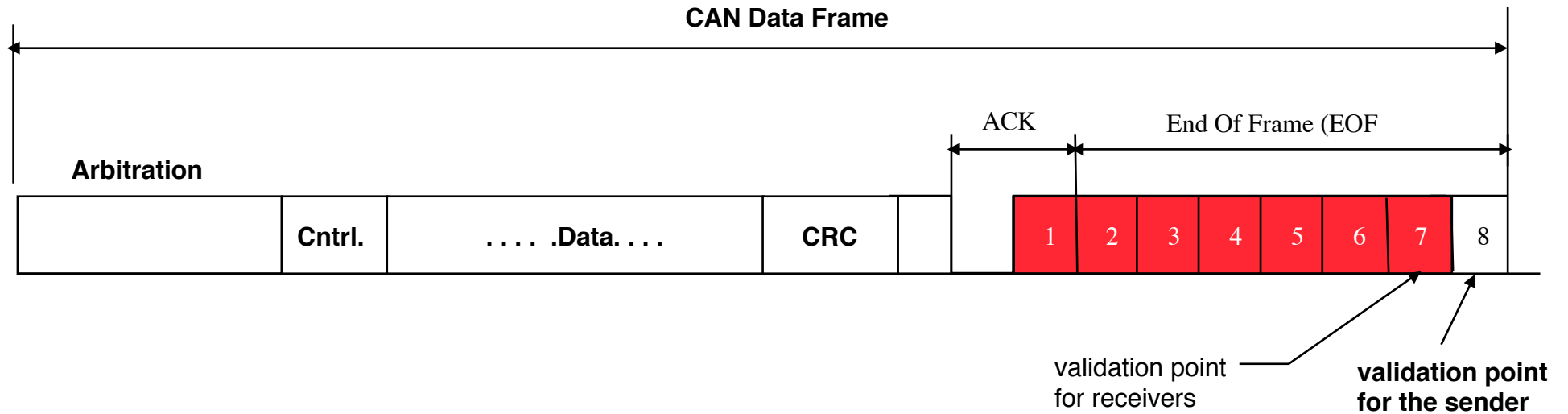
Violation of the Bit-Stuffing Rule:
Used for Error Detection and Signalling



Bit-Stuffing enforces the following rule:
A sequence of 5 identical bit levels
is followed by a complementary bit level



The Case for SHARE: Inconsistent Omissions



unique pattern : 1 dominant, 7 recessive, 6 dominant !



The Architecture of SHARE

