



CAN

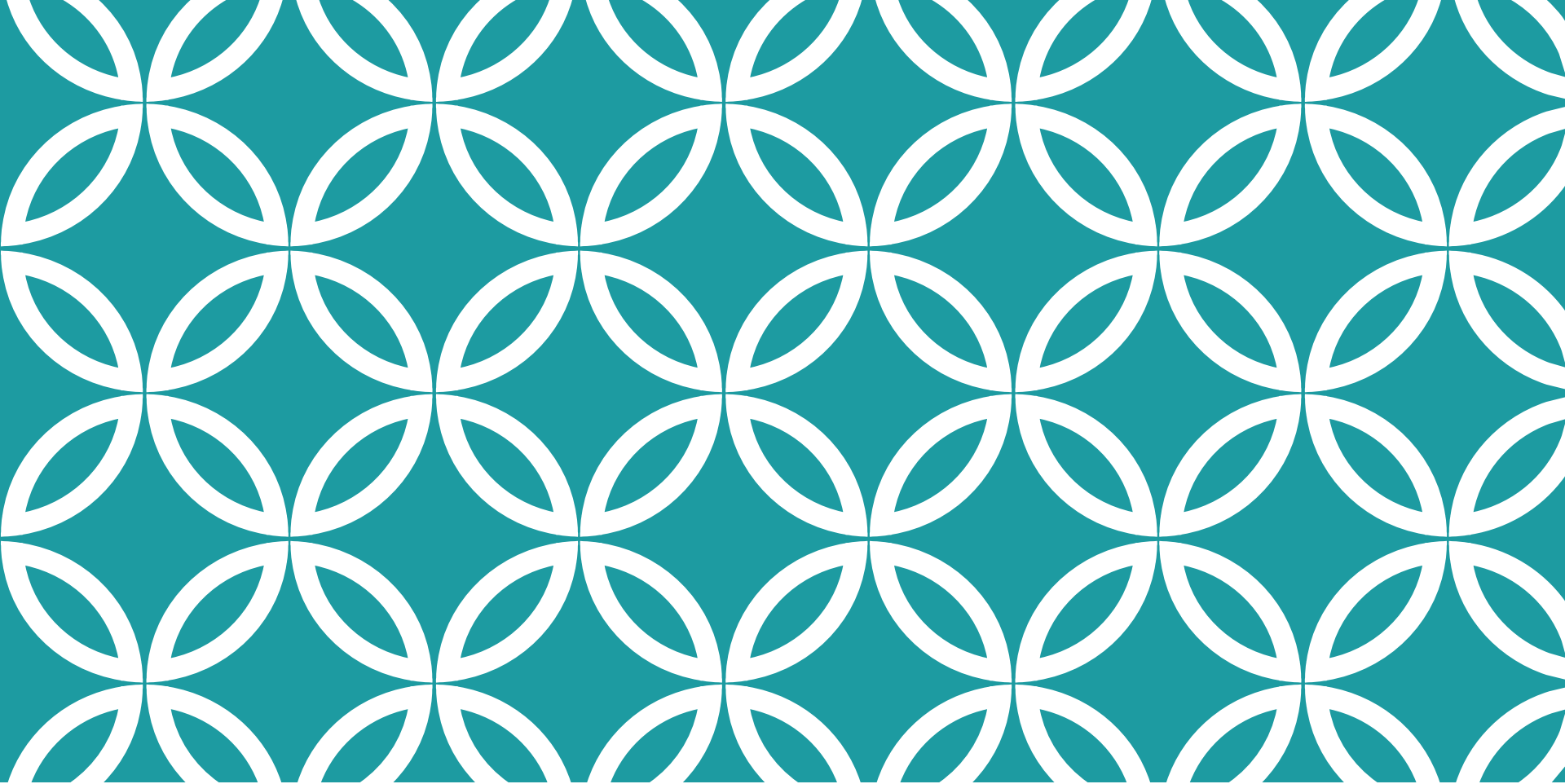
Најпознатији
аутомобилски
протокол

ЦИЉЕВИ

- Након завршетка овог предавања имаћете:
 - Боље разумевање *CAN* магистрала:
 - Историјат и еволуцију развоја
 - Градивни елементи: од физичког преко *МАС* до транспортног
 - Принципи који су омогућили побољшања
 - Поређења са осталим магистралама

ПРЕГЛЕД

- Шта смо радили:
 - LIN – аутомобилска асинхрона серијска комуникација
- Овај час: *CAN*
 - Преглед, принципи, еволуција, побољшања
 - Архитектура: топологија, физички слој, арбитража
 - Формати пакета и ефикасност
 - Препознавање грешака



ПРЕГЛЕД

CAN (CONTROLLER AREA NETWORK)

ПРЕГЛЕД



- Сви актери су равноправни и могу произвољно да комуницирају.
У случају колизије користи се посредовање да се одреди приоритет.
- Брза серијска магистрала: ефикасно, поуздано, и исплативо
- Линијска магистрала са више руководилица: окидање на догађаје
- Елегантно посредовање: на основу најмање битске вредности
- Дељење информација и ресурса између дистрибуираних апликација

CAN

ПРИМЕНА



- Развијено 1980-их: користи се у свим деловима (мотор, шасија, тело)
- Распрострањено у Европи, Америци, и у Азији
- Уграђује се и у остале чипове (нпр. степ мотори, помоћ за паркирање)
- Користи се и у индустријске примене: <http://www.can-cia.de/>

CAN НАМЕНСКИ СИСТЕМИ

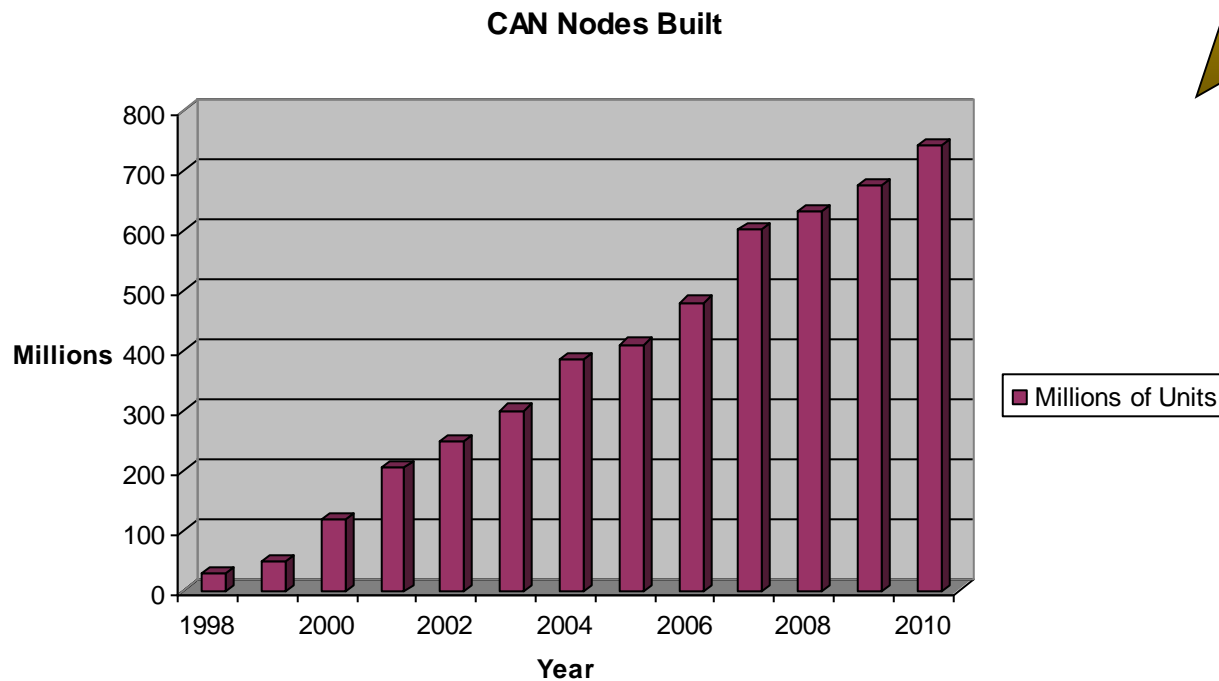
- Главни разлози за примену:

- Пријемчиво
 - Ниске цене ожичења и опреме
- Поузданост:
 - Комуникација без грешака
 - Отпорност на ЕМС
- Доступност:
 - Стандардизовано
 - Брдо микроконтролера са подршком
- Проширивост



БРОЈ НАПРАВЉЕНИХ ЧВОРОВА

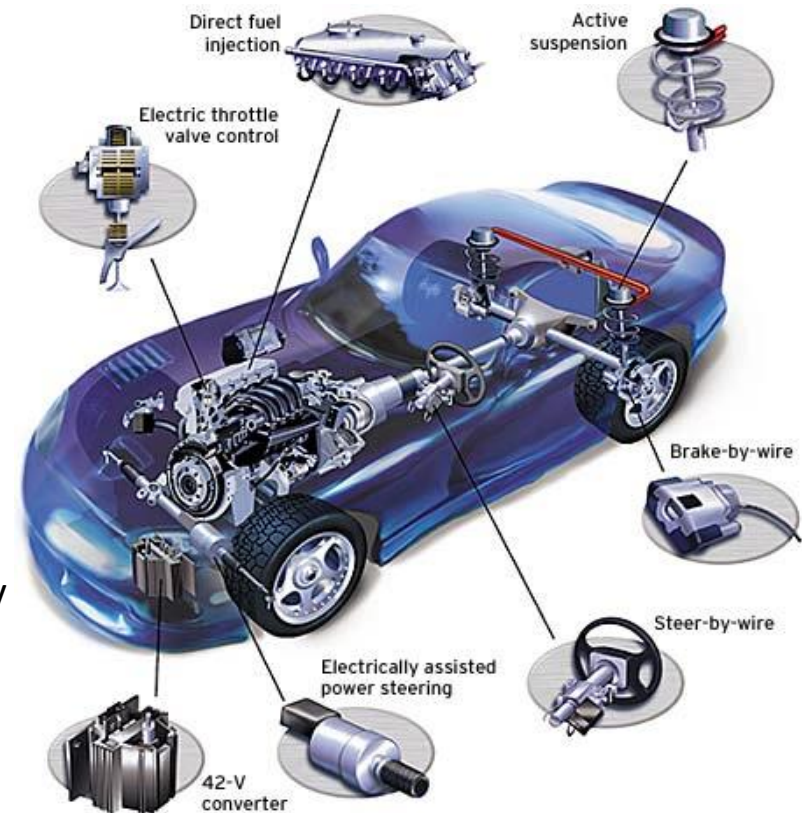
~Преко 2
милијарде!!!*



CAN

СТРУКТУРА

- Стандарди:
 - CAN 2.0A (ISO11519) | Can 2.0B(ISO11898)
- Слојеви:
 - Физички слој: засновано на доминантном биту
 - Транспортни слој: арбитража бит по бит
 - Слој за филтрирање
- Критични аспекти:
 - Приоритети порука
 - Капацитет уређаја: колико могуће смањити без прекорачења спремника
 - Компромиси



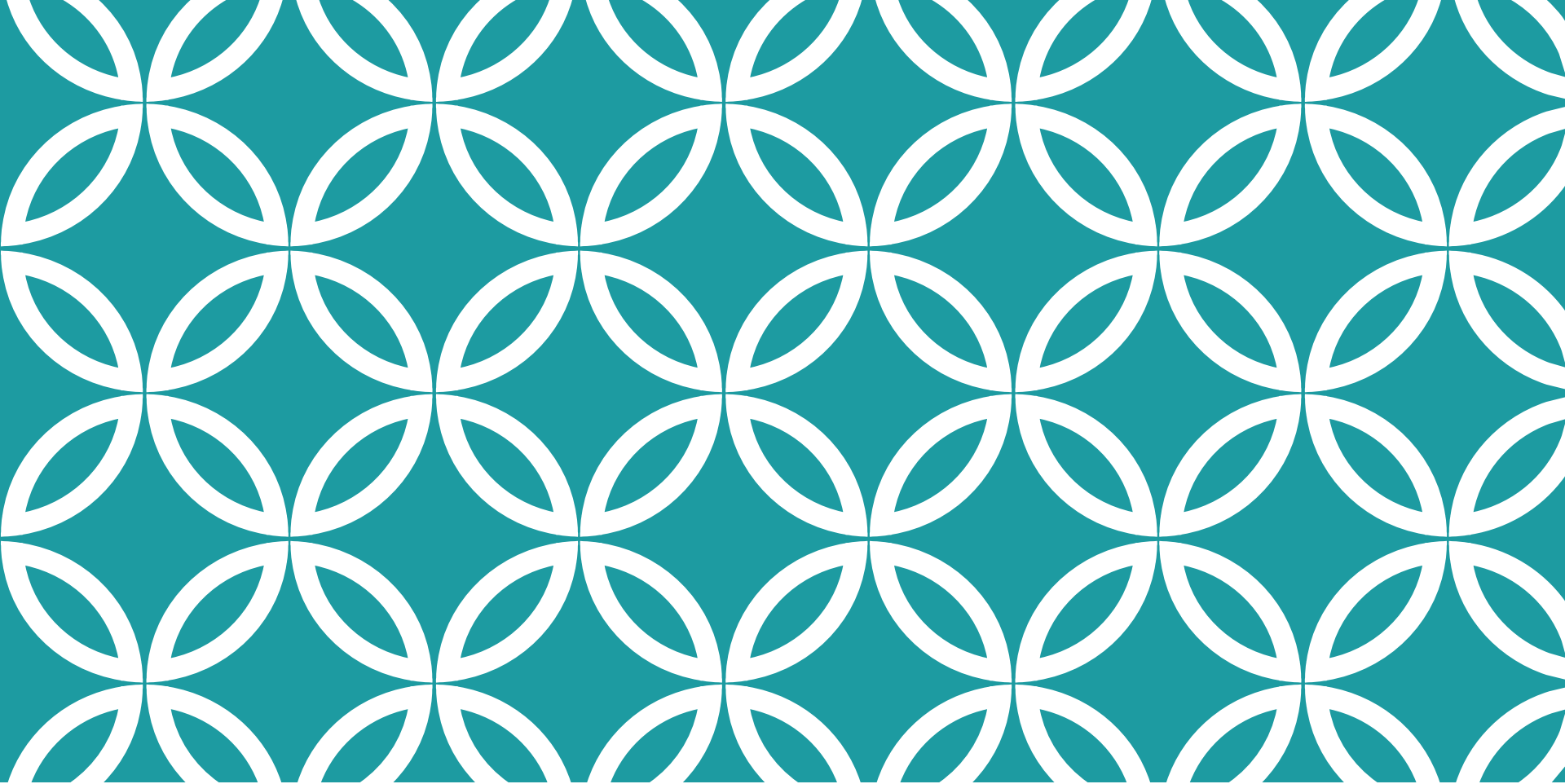
CAN

УВОДНИ ДЕТАЉИ

- Асинхрона комуникација (окидање на догађаје)
- Либеран приступ: било ко приступа када је медијум тих
- Недеструктивна арбитража:
 - 100% искоришћеност пропусне моћи
- Приоритет на основу идентификатора: 11 бита (проширено 29 бита)
 - Већа кашњења за поруке ниског приоритета
 - Мала кашњења за поруке високог приоритета
- Аутоматско препознавање грешака, сигнализација, понављање
- Упредена парица – брзине до 1 Mbps - домет 40 m
- ISO 11898 spec – 32 уређаја

ФУНКЦИОНАЛНОСТИ И ПРЕДНОСТИ

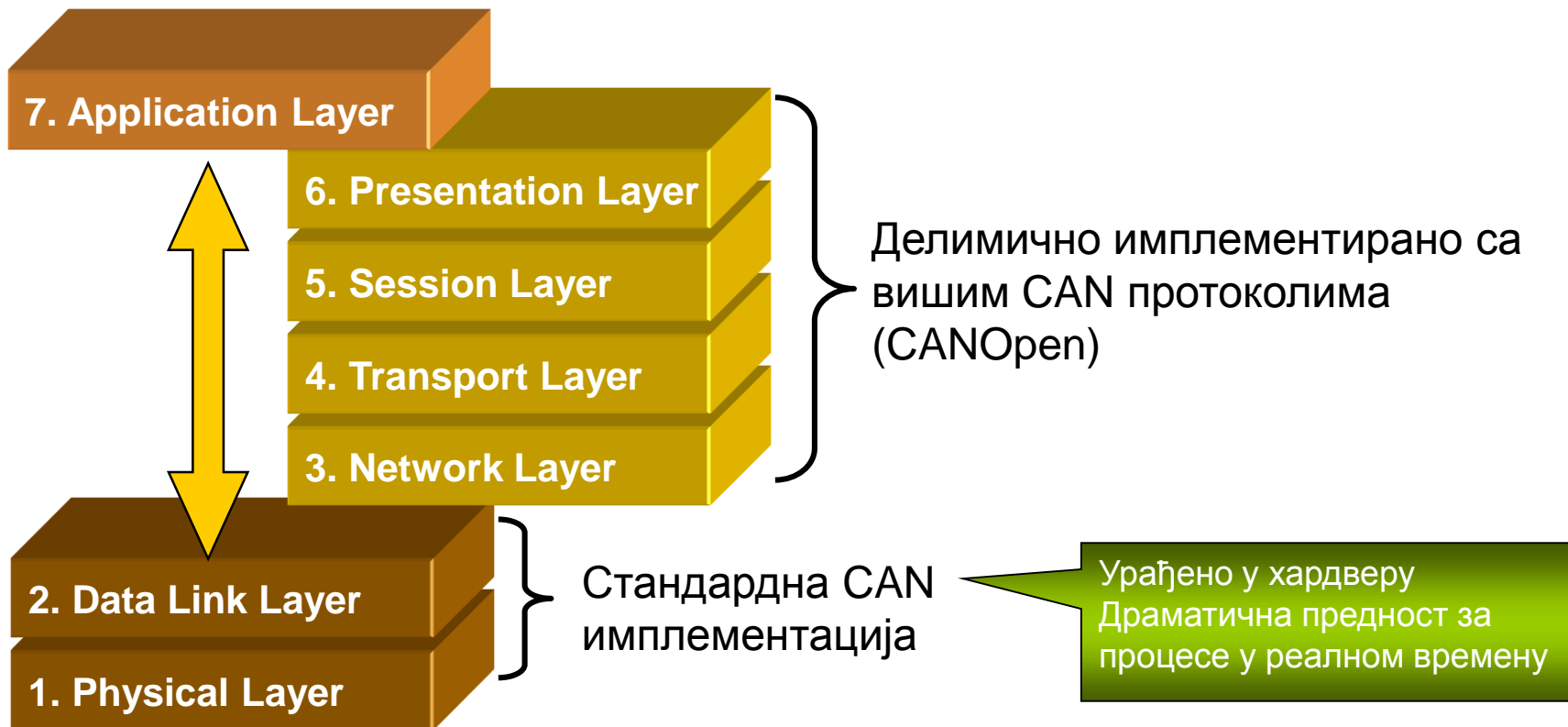
- Хијерархија уз више руководиоца
- Брзина: 1 Mbs
- Величина пакета: 0-8 октета
- Јединствено адресирање
- Филтрирање на пријему
- Препознавање грешака
- Мере за ограничавање грешака (гашење физички оштећених)
- Аутоматско понављање
- Редундантан систем
- Одзив у реалном времену
- Упрошћени захтеви
- Флексибилност у осмишљавању
- Арбитрирање и приоритети
- Висока поузданост
- Неометен саобраћај
- Стабилана веза



АРХИТЕКТУРА И МОДЕЛ

CAN

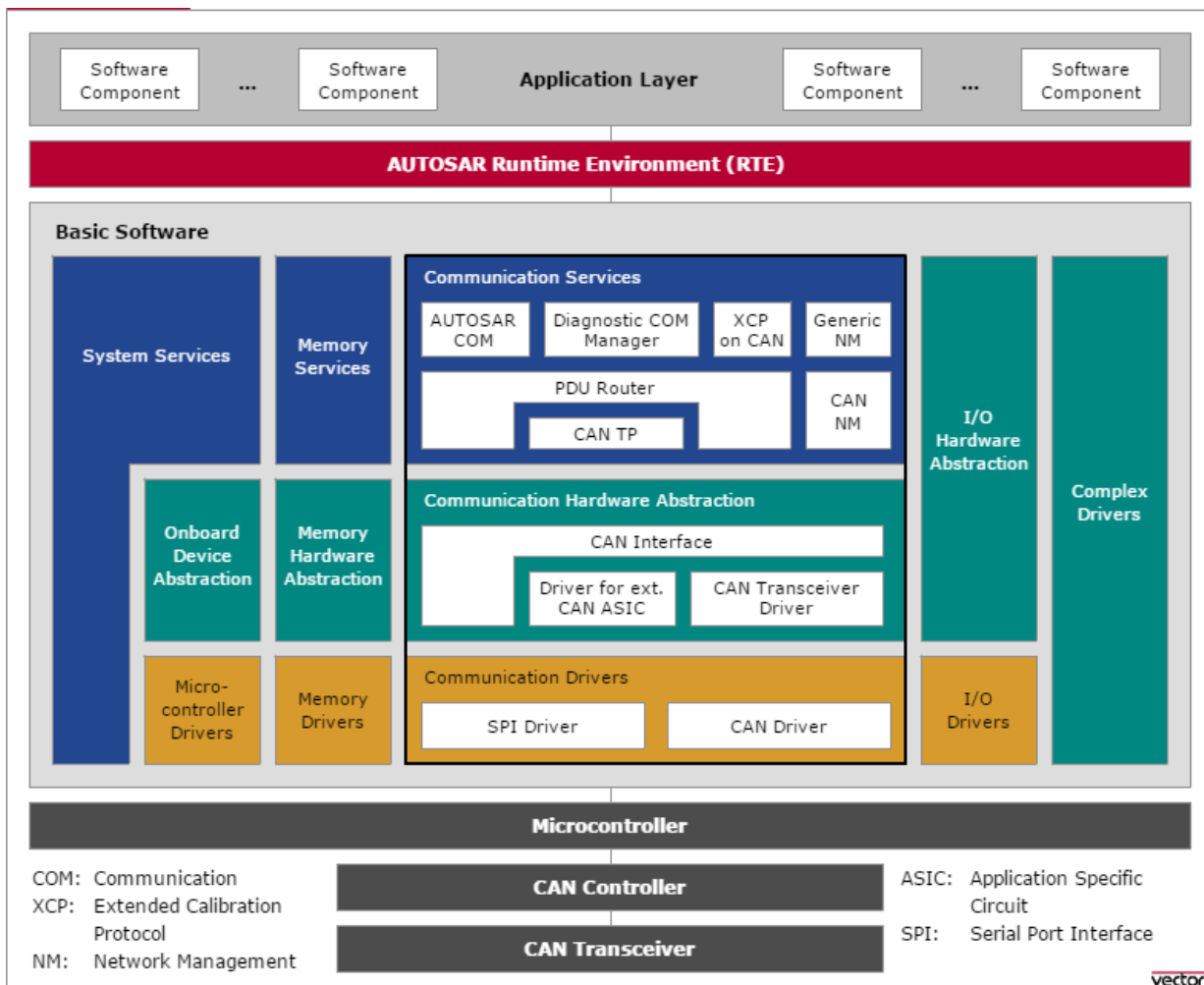
ИСА/ОСИ МОДЕЛ (7 СЛОЈЕВА)



AUTOSAR

- **AUTOSAR** (AUTomotive Open System ARchitecture)
 - Референтна софтверска архитектура за ECU
- Кључни централни део чини:
 - AUTOSAR Runtime Environment (RTE)
 - Одвајање апликације од мрежних функција
 - Униформисани сервиси и компоненте
 - Три нивоа: од доле ка горе
 - Microcontroller Abstraction Layer
 - ECU Abstraction Layer
 - Service Layer

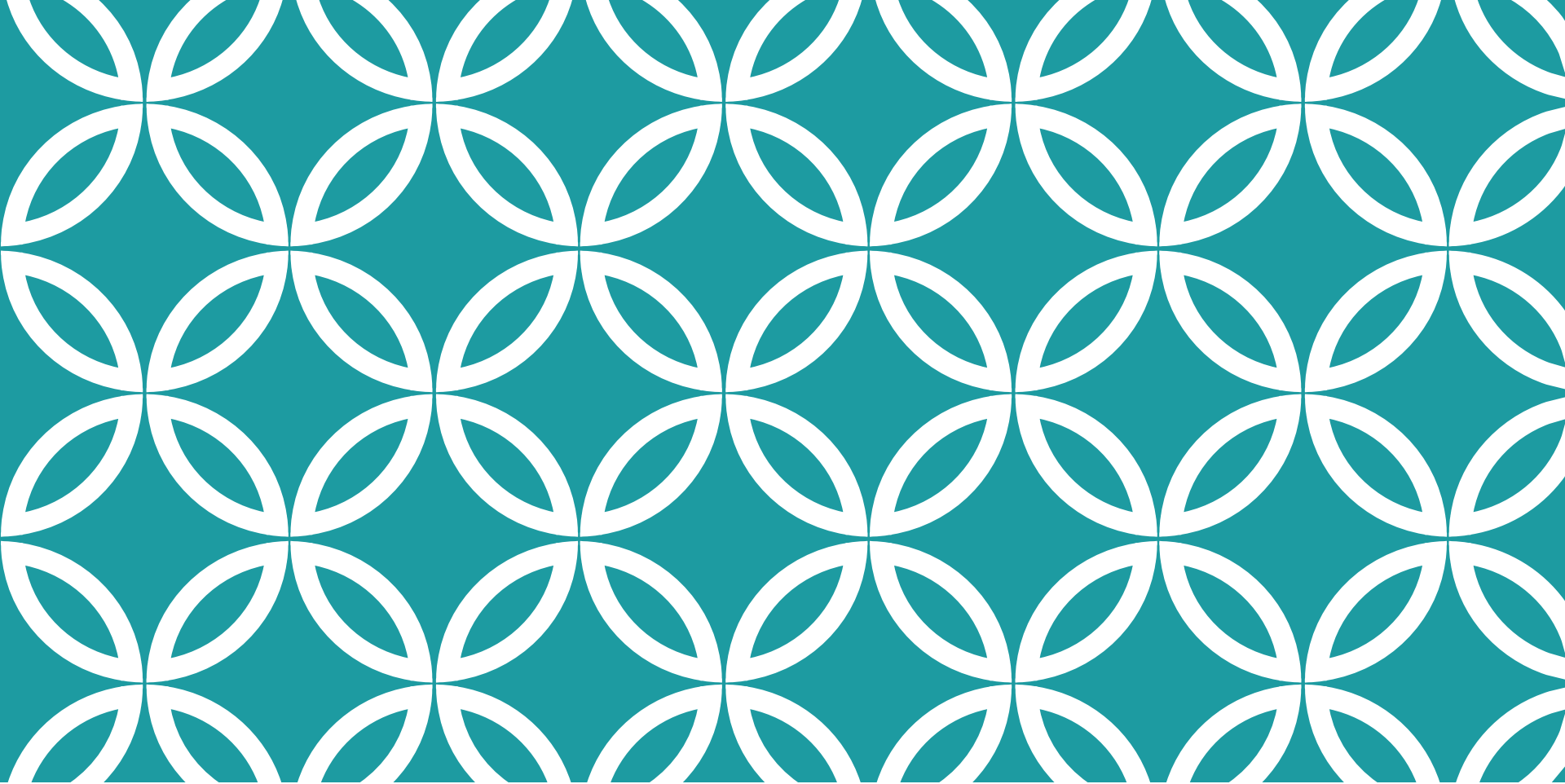
AUTOSAR ПРИМЕР



AUTOSAR CAN

СЕРВИСИ

- Представљена структура модерног CAN уређаја
- Један од битнијих блокова *Communication Services*
 - AUTOSAR COM (Communication) - стандардни комуникациони сервисиси
 - Diagnostic COM Manager – сервиси за дијагностику комуникације
 - Network Management (Generic NM/CAN NM) - сервиси за управљање мрежом
 - CAN TP (Transport Protocol) – сервиси транспортног слоја
 - PDU Router (Protocol Data Unit) – координација између интерних и спољашњих слојева



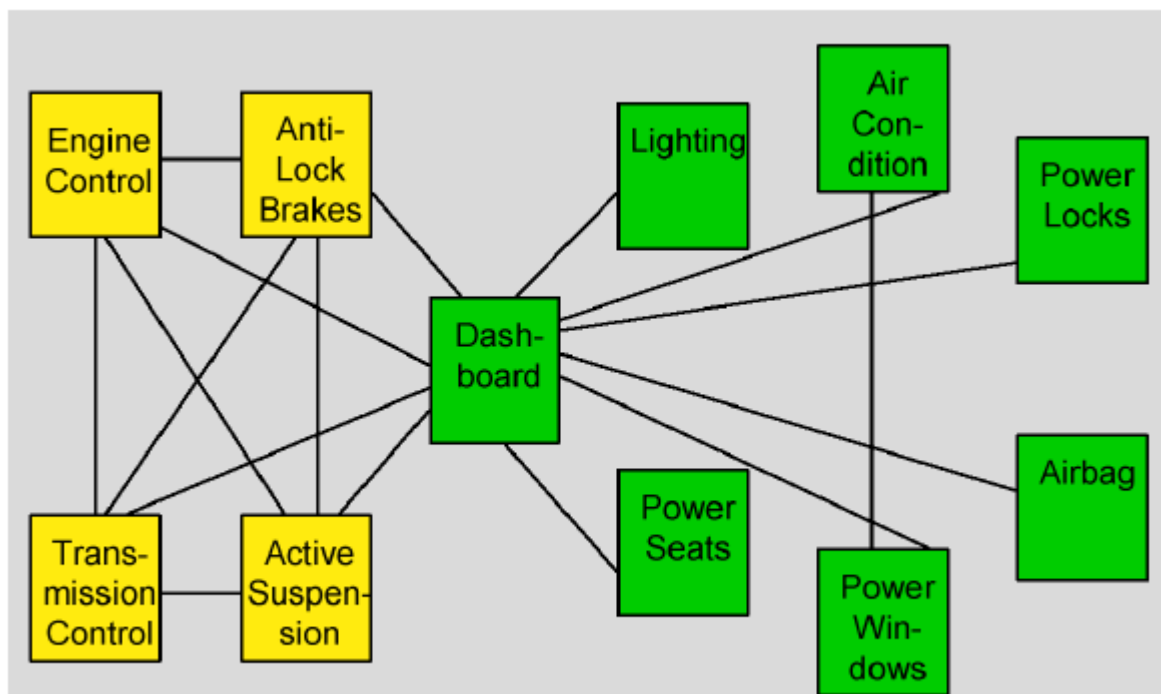
ТОПОЛОГИЈА



ТОПОЛОГИЈА

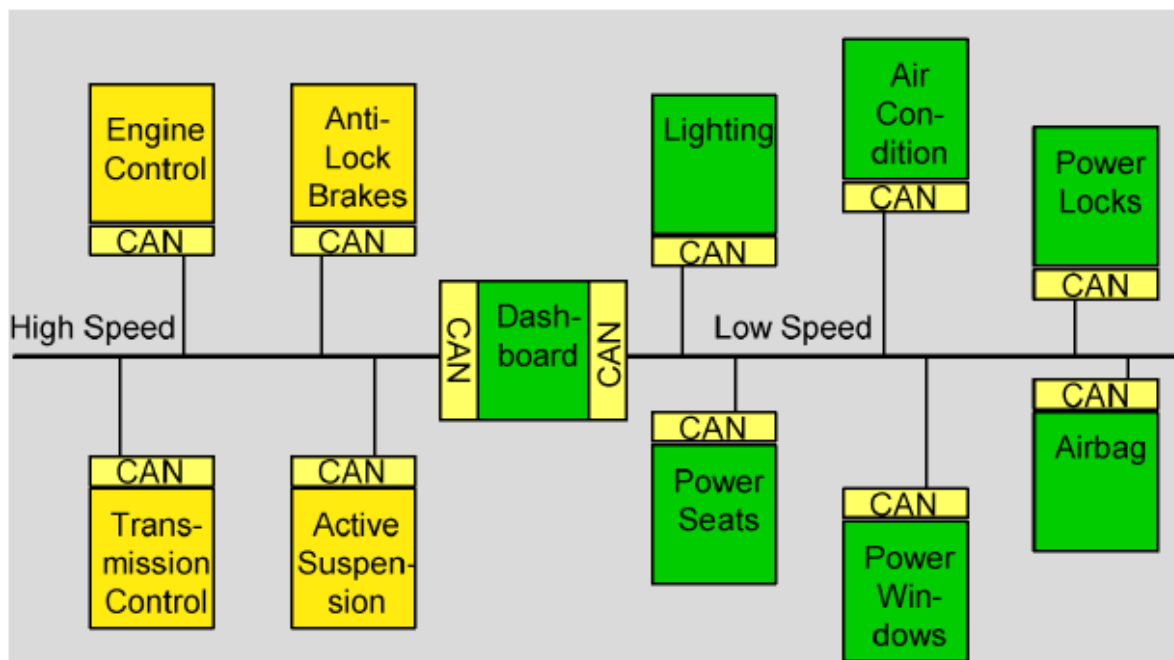
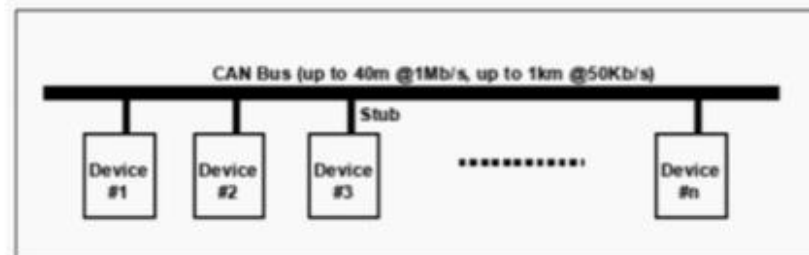
ПОЈЕДИНАЧНЕ ВЕЗЕ

- Пре појаве CAN



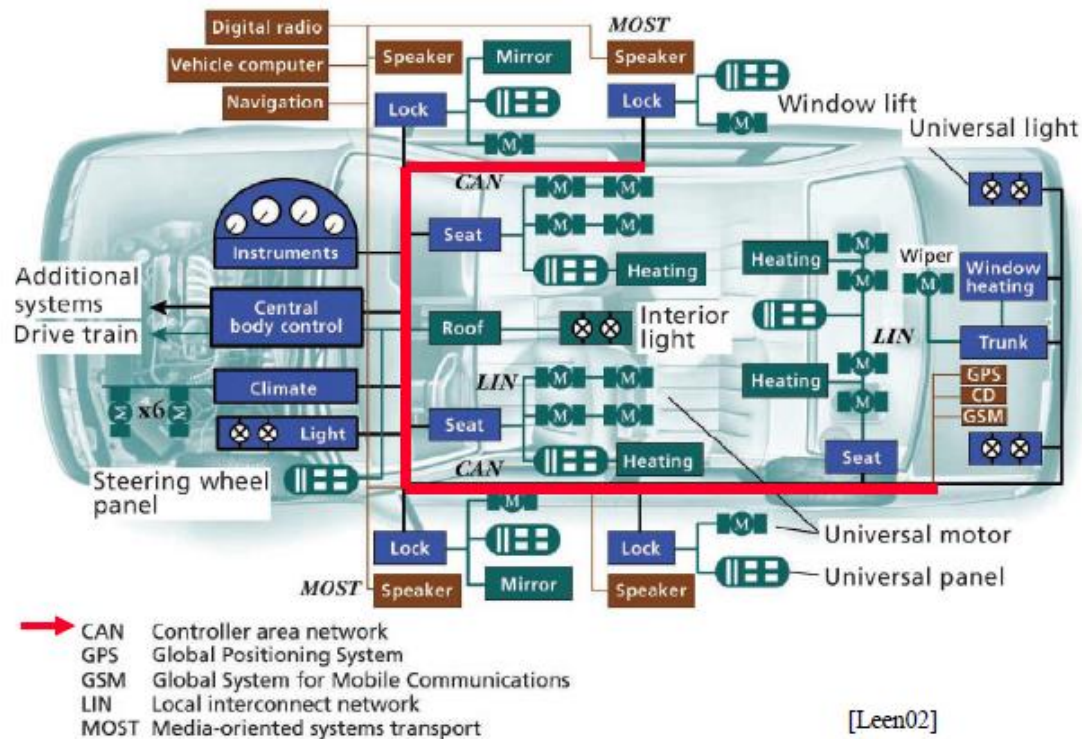
ТОПОЛОГИЈА

ДЕЉЕНА МАГИСТРАЛА



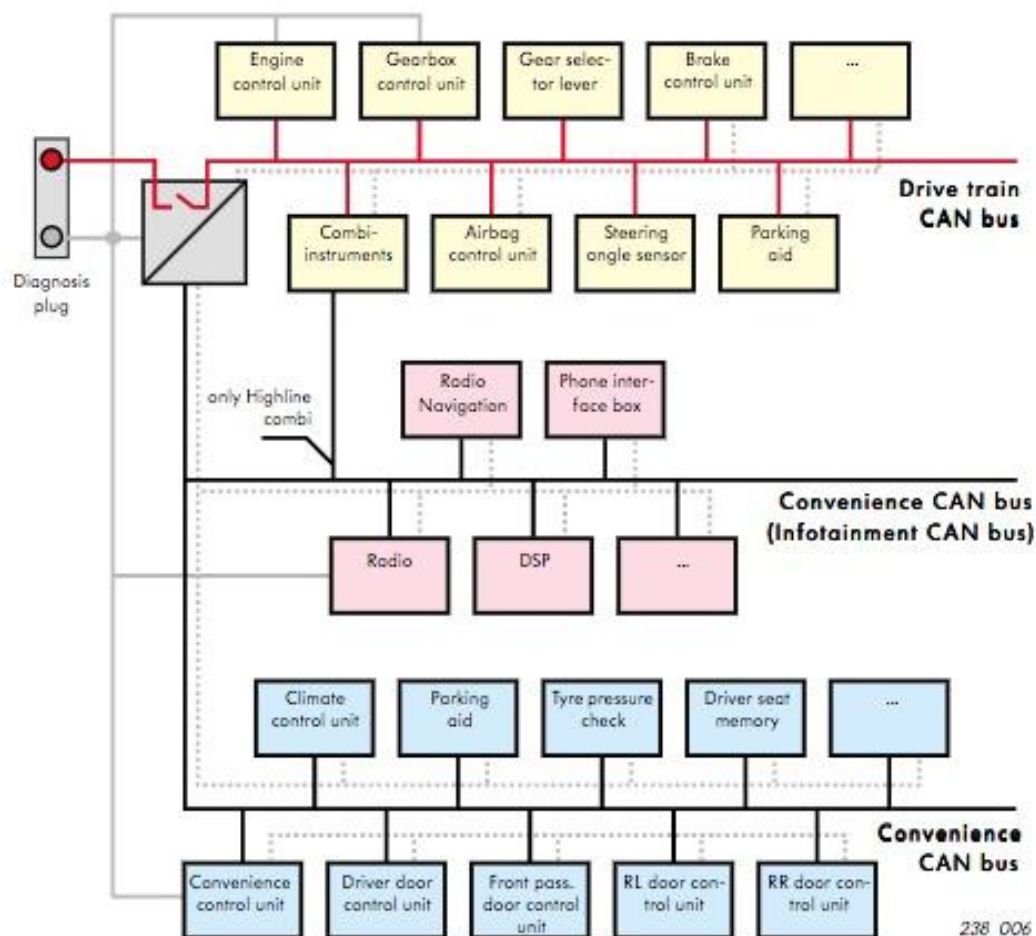
ОКОСНИЦА МРЕЖЕ

- CAN има централну улогу



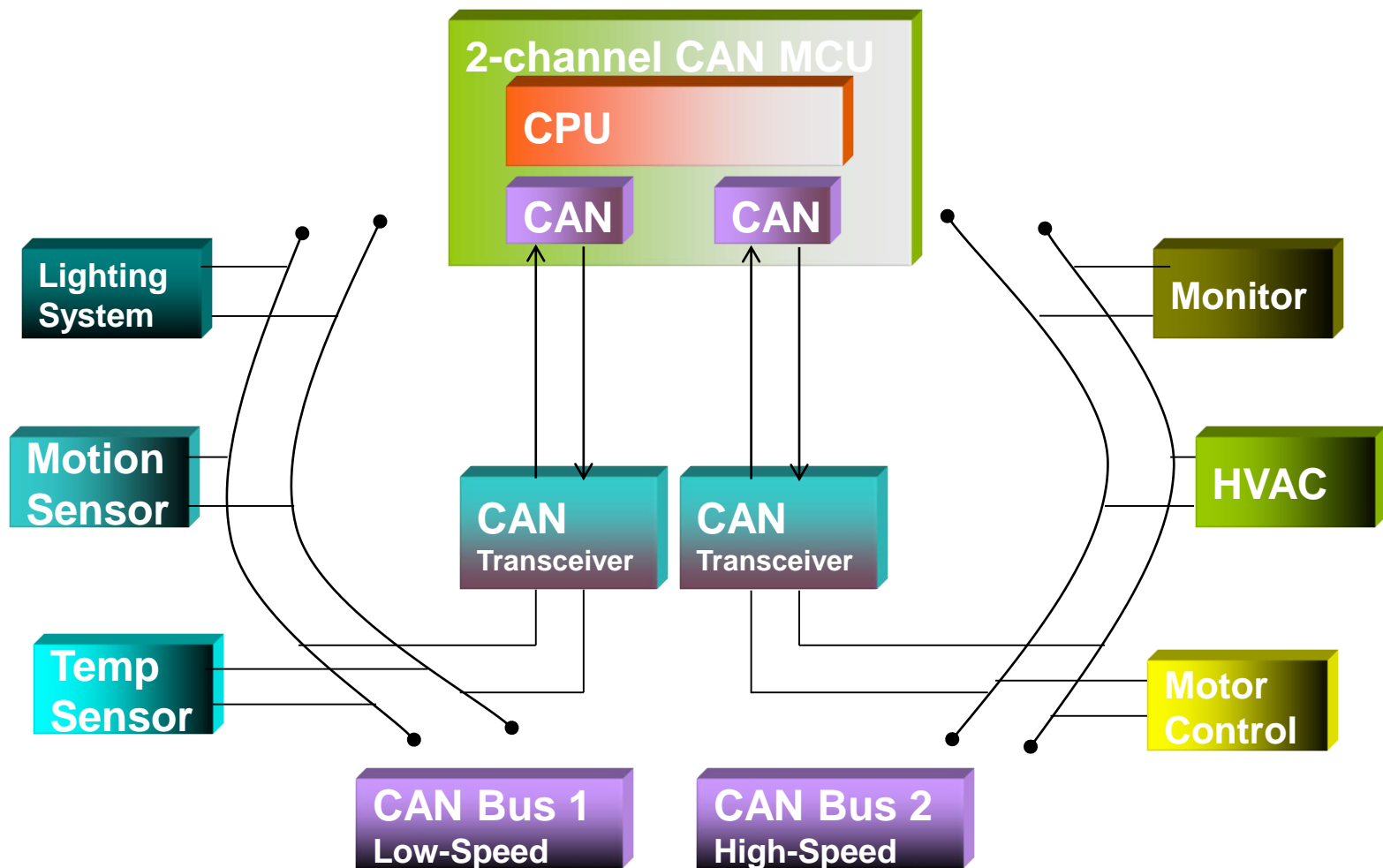
CAN

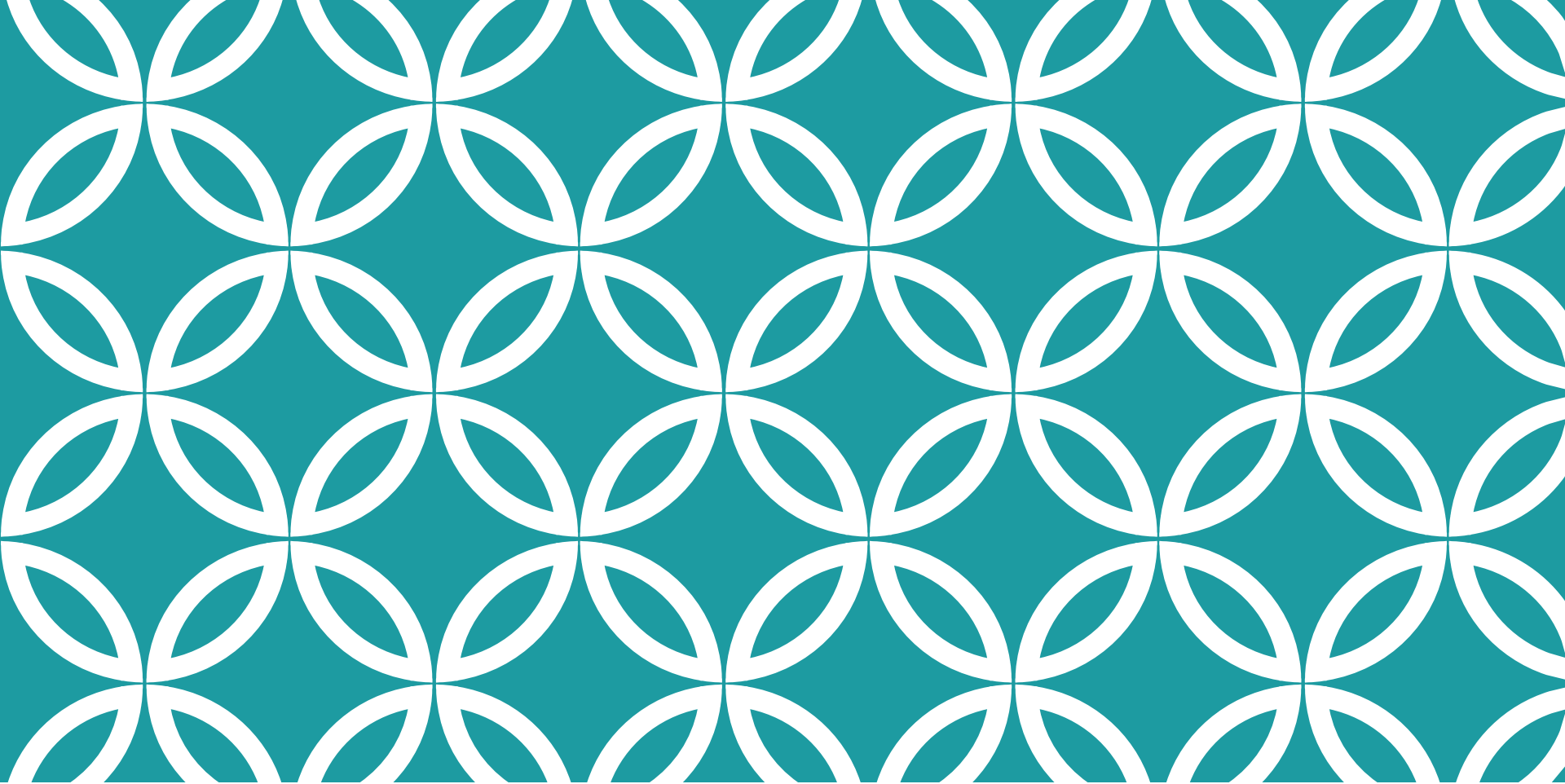
ВРСТЕ МАГИСТРАЛА



ТИПИЧНА МАГИСТРАЛА

ДВА КАНАЛА





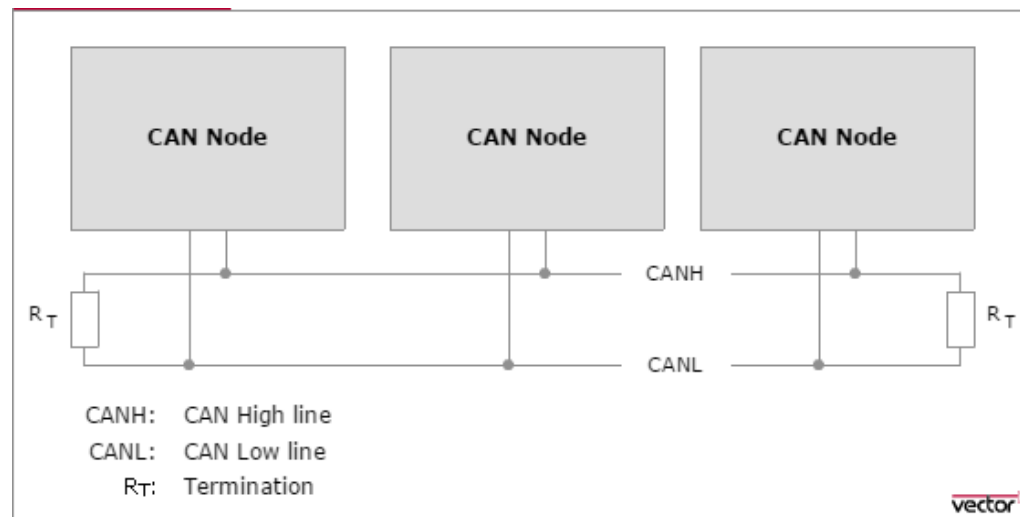
ФИЗИЧКИ СЛОЈ



CAN

ФИЗИЧКИ СЛОЈ

- Упредена параца: (не)оклопљена
- Одвојене линије за 2 сигнала: CANH и CANL
- Потребно омеђити линију - ISO 11898 стандард: $120 \pm 12 \Omega$



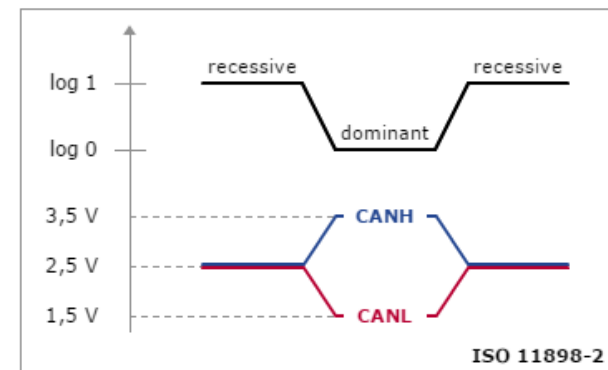
ДИФЕРЕНЦИЈАЛНА МАГИСТРАЛА

- Логичка стања се процењују на основу разлике напона

- Напонски нивои зависе од конкретног сучеља:

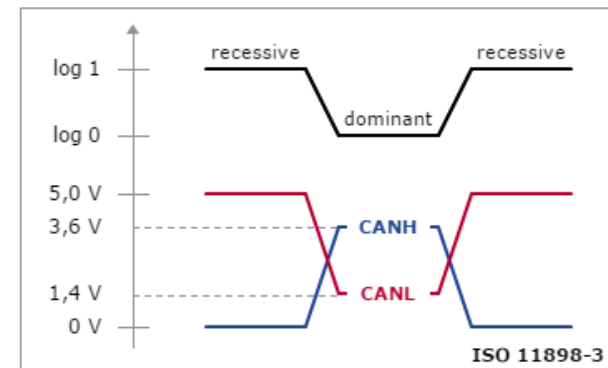
- High-speed CAN bus interface (**ISO 11898-2**)

- Логичка 1: разлика од 0 V (већ од 0.5 V)
- Логичка 0: разлика од 2 V (чак и преко 0.9 V уколико је напајање +-12 V)



- Low-speed bus interface (**ISO 11898-3**)

- Логичка 1: разлика од 5 V
- Логичка 0: разлика од 2 V

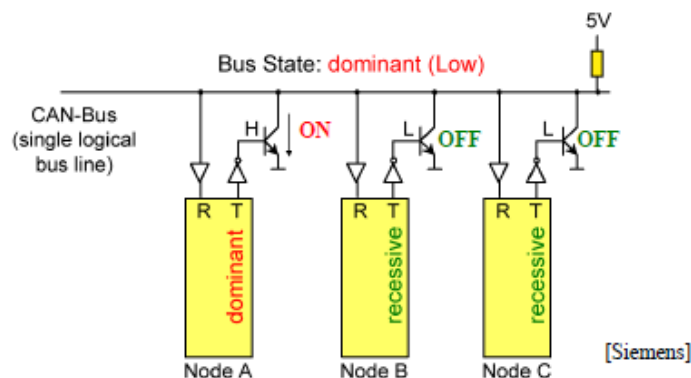


- Хистерезис коло се додаје да повећа имуност на сметње

ЛОГИЧКО „И“ МАГИСТРАЛА

- **Концепт: рецесивних и доминантних бита**

- Магистрала ожичена као „и“ коло
- Подразумевано високо стање (pull-up)
- Стање се активно обара на ниско



- **Високо стање: рецесиван бит**

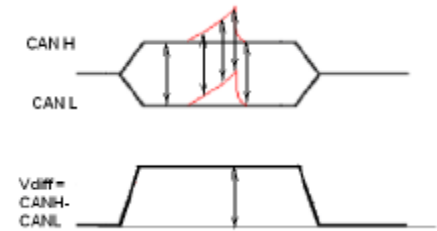
- Немогућност преиначења стања магистрале уколико се шаље "1"

- **Ниско стање: доминантан бит**

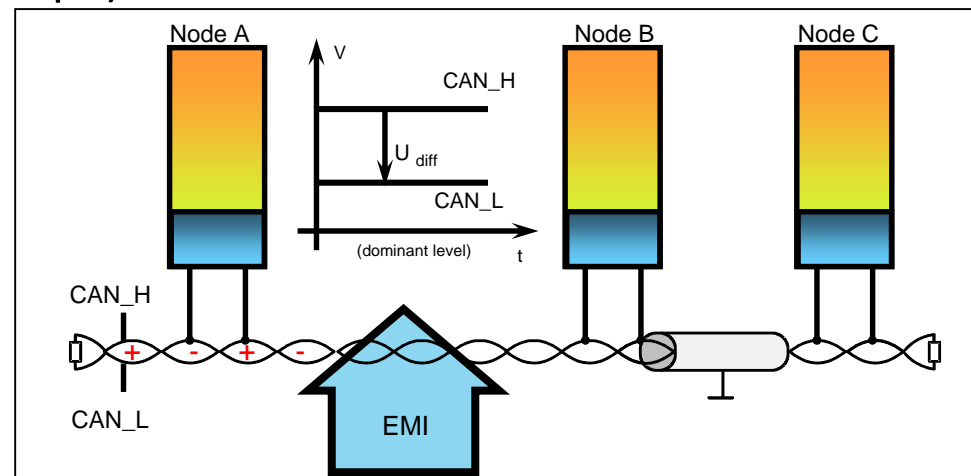
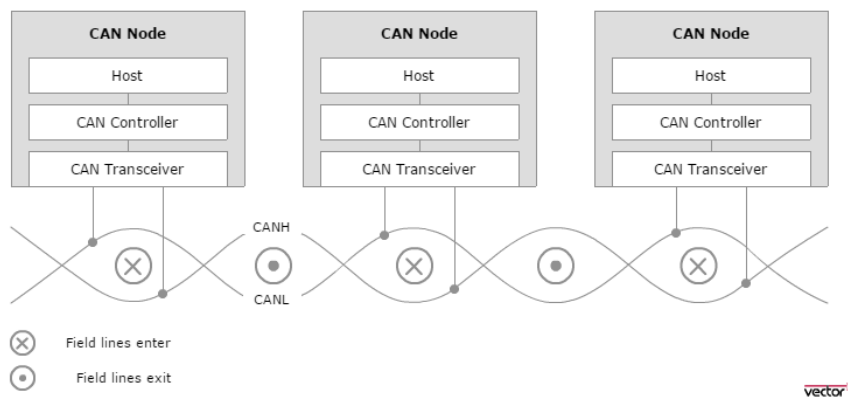
- Стање магистрале се преиначује слањем "0" независно од осталих

- **Опасност: покварен пренос на доминантном стању**

ЗАШТИТА ОД СМЕТЊИ



- Симетрични **диференцијални** сигнали: подједнако на оба – разлика 0
- **Уплетена** парица: сабирање супростављених магнетских поља
- Резултат: повећана ЕМК – усклађеност и отпорност
 - Широк опсег радних услова
 - Елимисан утицај индукованих сметњи од мотора, алтернатора, и прекидача
- Низак ниво емисија (прљање спектра)



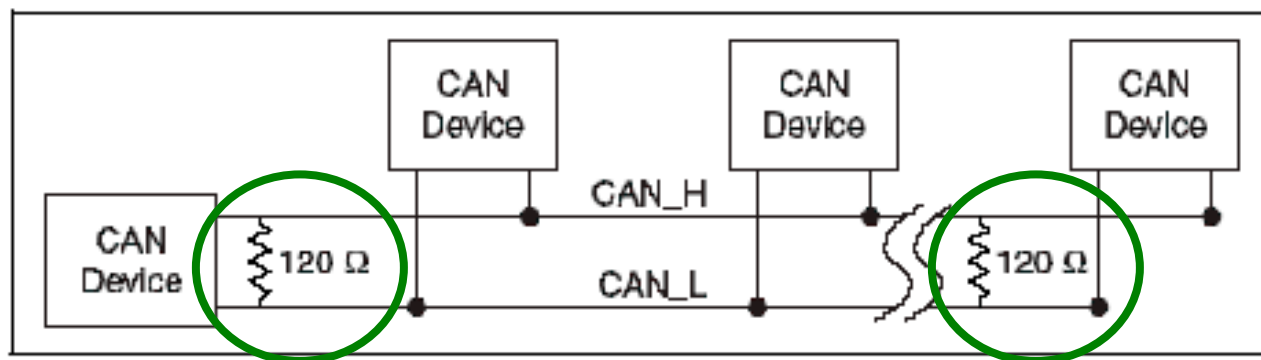
УТИЦАЈ ОДБИЈАЊА СИГНАЛА

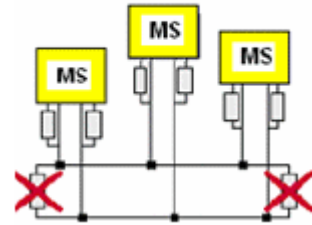
- Ограничена брзина простирања сигнала у медијуму: рефлексije
- Утицај рефлексija расте са:
 - повећањем брзине
 - дужином магистрале
- Потреба за омеђавањем отпорницима (енг.terminating):
 - Теорија максималног преноса снаге
 - Подударање са карактеристичном импедансом линије:

ОМЕЂАВАЊЕ ЛИНИЈЕ

HIGH-SPEED CAN (ISO 11898-2)

- Брзина: од 125 Kbs до 1 Mbs
- Оба краја две линије (CAN_H и CAN_L) морају да се омеђе (ограде)
- Само крајни уређаји омеђени: средина слободна
- ISO 11898 диктира номиналну импедансу: 120 Ω

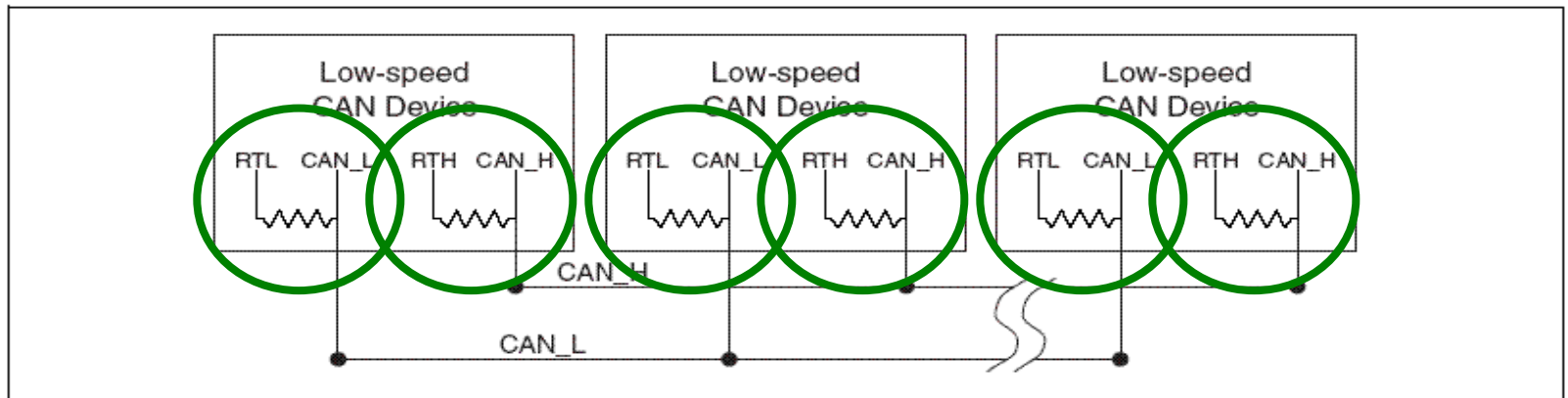




ОМЕЂАВАЊЕ ЛИНИЈЕ

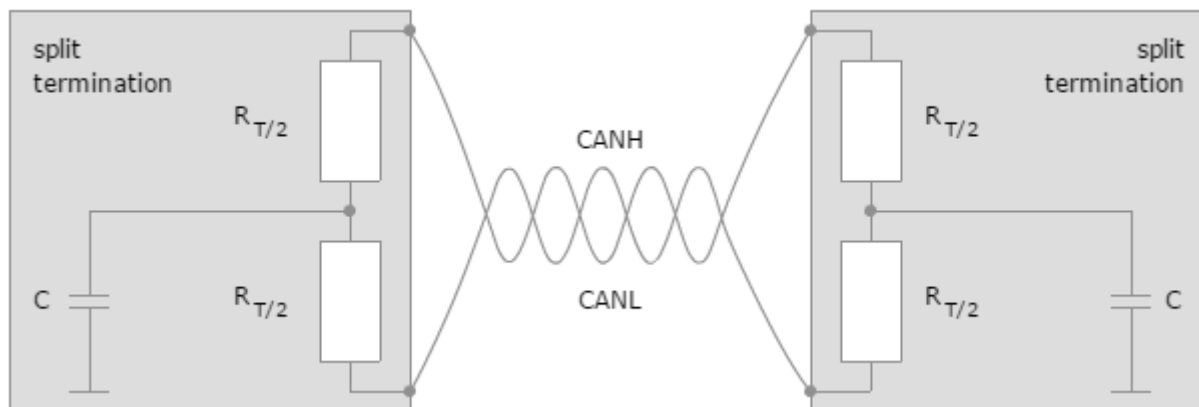
LOW-SPEED CAN (ISO 11898-3)

- Брзина: мања од 125 Kbs
- Нема потребе за омеђавањем крајева линије
- Омеђавају се уређаји и свака линија посебно: RTH-CAN_H и RTL-CAN_L
- Отпорност се индивидуално израчунава



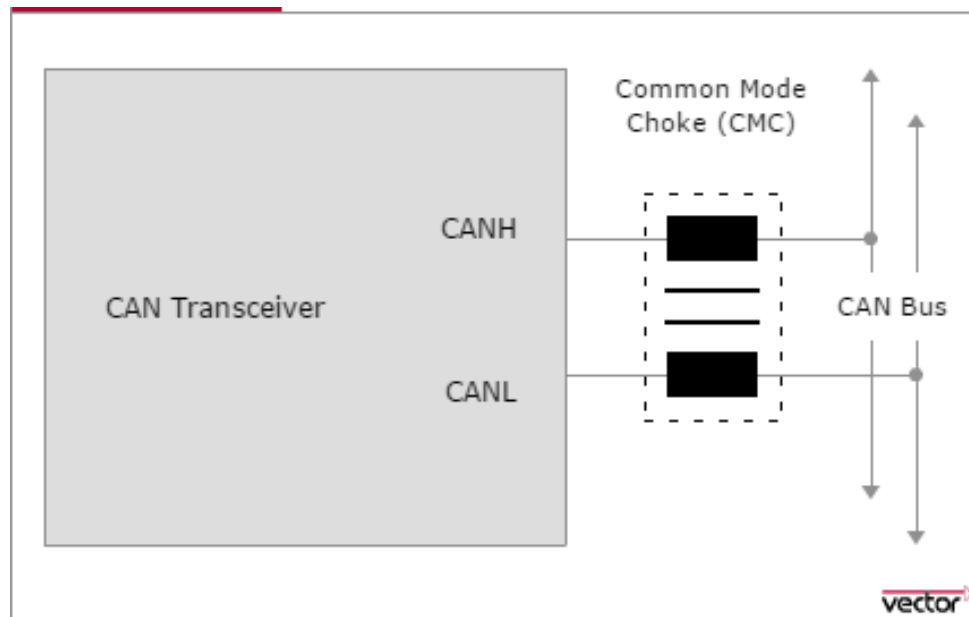
РАЗДВОЈЕНО ОМЕЂАВАЊЕ

- Служи као нископропусни филтер
 - Оборити високовректентне компоненте на уземљење
 - Једносмерна компонента се не дира
- Принцип:
 - РЦ коло
 - 2 идентична отпорника ($60\ \Omega$) и капацитивност (типично $4.7\ \text{nF}$)
- Мерења показала ефектност: емисије и отпорност на сметње



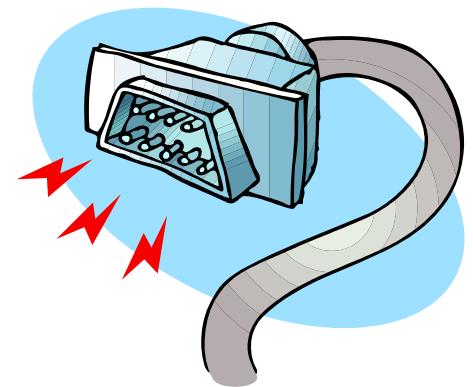
БАЛАНСИРАНИ ПРИГУШИВАЧ

- енг. *Common Mode Choke (CMC)*
- Улога:
 - смањење симултаних високофреквентних компоненти на излазу
 - Могућност смањења штетних емисија



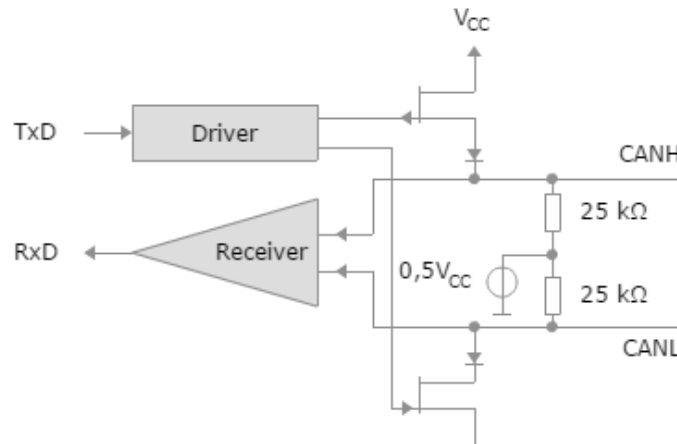
КАБЛОВИ И УТИЧНИЦЕ

- CAN не одређује врсту физичког медијума (слобода избора)
- Типични каблови:
 - Упредена парица Twisted pair
 - Оклопљена упредена парица
- Број каблова:
 - Нормално један
 - Уколико има потреба за напајањем: додатни пар
- Типичне утичнице
 - 9-pin Dsub
 - 5-pin mini style
 - Terminal blocks
 - Зависно од апликације (нпр. телефонске утичнице)

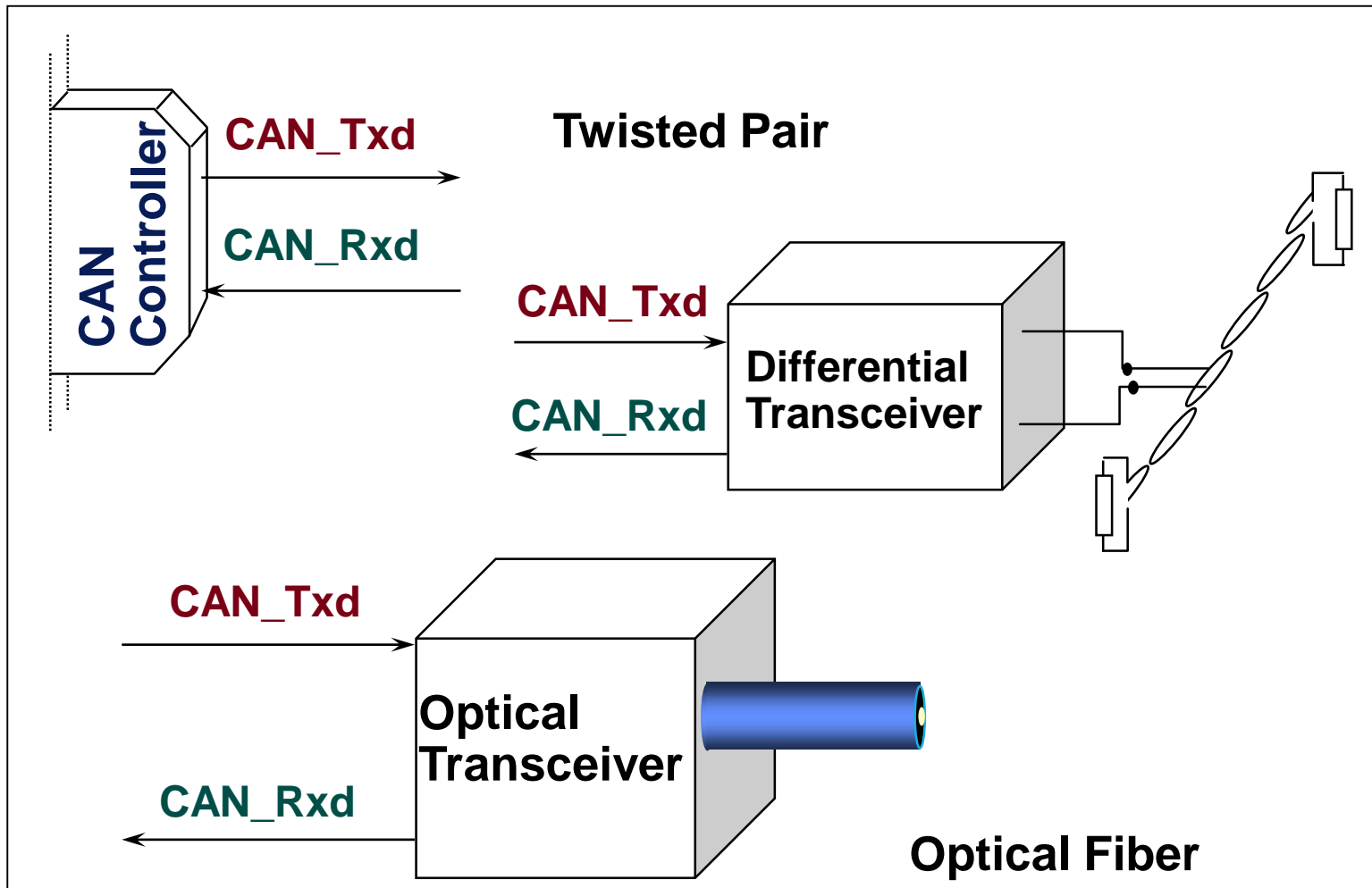


ПРИМОПРЕДАЈНИЦИ

- Заштита од електростатичког пражњења (Electro Static Discharge (ESD)) до 8 kV
- low-speed CAN:
 - робусност: отказивање (кратак спој, прекид) једне линије – могућ наставак рада
- high-speed CAN transceiver
 - Два заједнички управљана транзистора
 - Блокирајући мод: обе линије на истом напону ($0.5 \cdot V_{CC}$), излаз 0
 - Проводни мод: диференцијални напон у складу са отпором оптерећења (2 V, 35 mA)

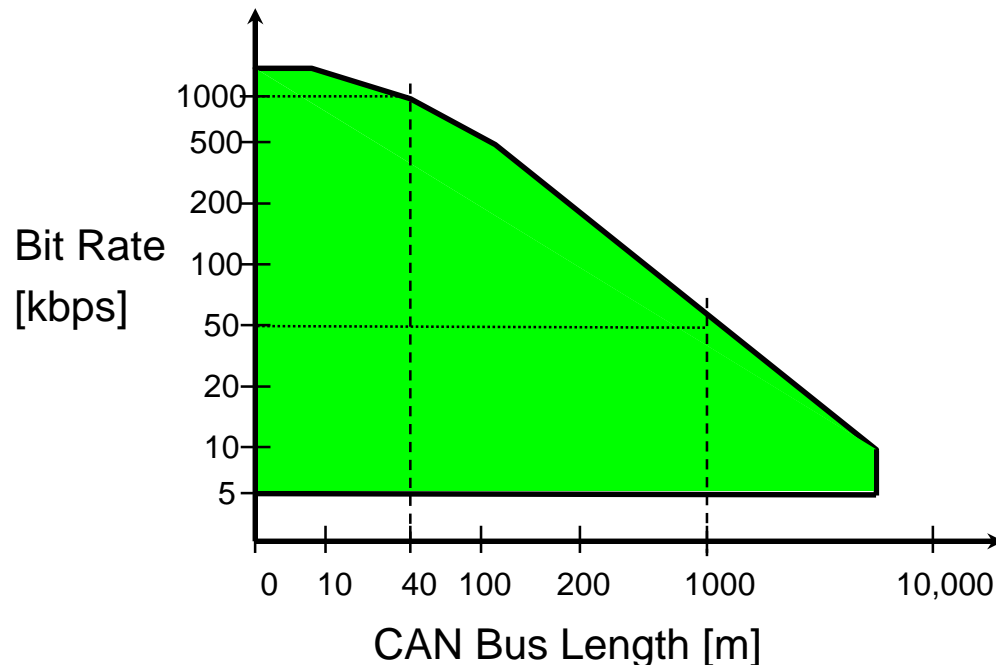


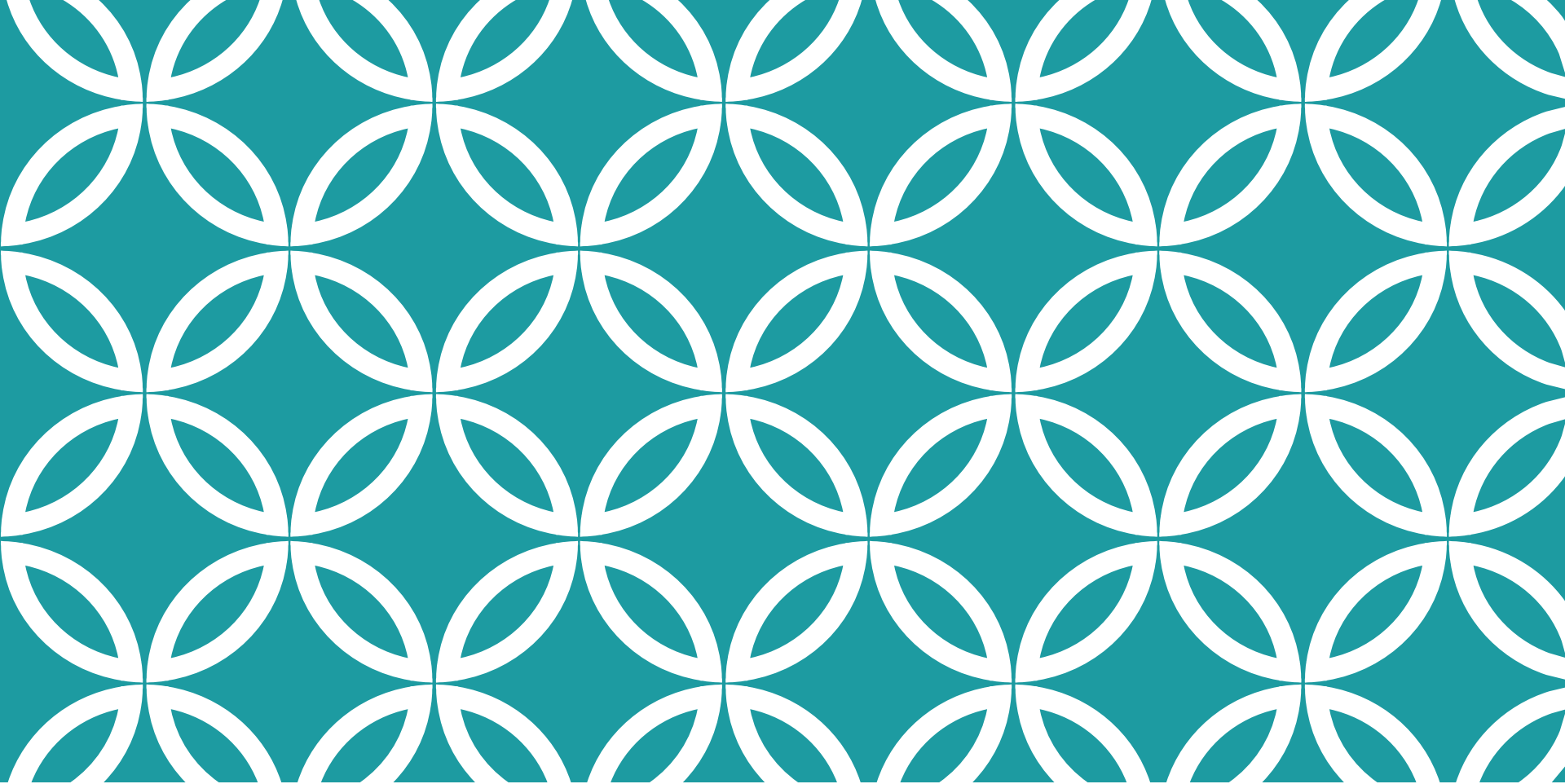
ФИЗИЧКИ МЕДИЈУМИ



ПРОПУСНА МОЋ И ДУЖИНА

- Прорачун заснован на електричном медијуму
- Рачун узима трајање бита, повратни пут, и препознавање судара
- Ефикасност компоненти утиче на зависност





ФОРМАТ ОКВИРА

Заглавља
Подаци
Поља за проверу

ПРЕГЛЕД ФОРМАТА

- **Заглавље:**

- Идентификатор у стандардном или проширеном формату (11 или 29 бита)
- Приоритет пошиљаоца
- Заглавље често означава пошиљаоца, примаоца, и врсту поруке

- **Подаци:**

- Апликација или виши слојеви прослеђују вредности
- Дужина: 0 до 8 октета – **за шта служи 0?**

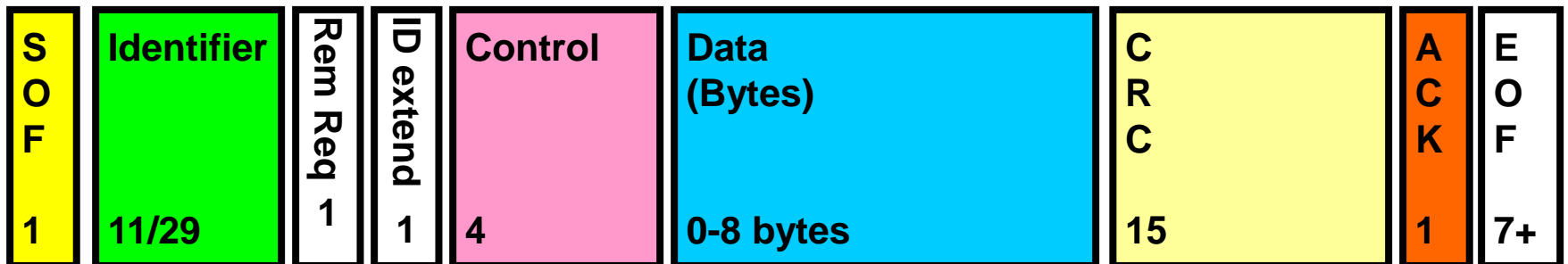
- **Препознавање грешке:**

- Коришћење 15 бита CRC



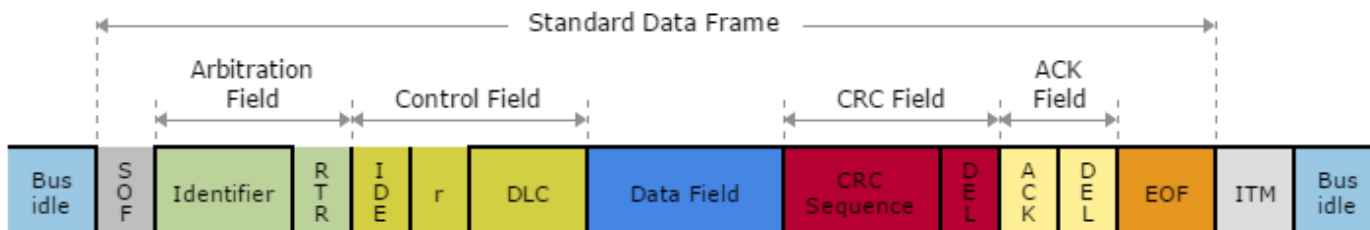
ПОЉА У ОКВИРУ

- Start of Frame – 1-bit
- Arbitration Field – 11-bits/29-bits
- Control Field – 6 bits (2 reserved, 4 representing number of Data Field bytes)
- Data Field – 0 to 8 BYTES
- CRC – 15-bits
- ACK Field – 1-bit/variable
- End of Frame – 7-bits (recessive)



ОБЈАШЊЕЊЕ ПОЉА

- **SOF – Start of frame (SYNC symbol):** 1 доминантни бит (0)
- **Arbitration field:** ниска 11/29 бита, приоритет, поставља апликација
 - Може да садржи RTR (remote transmission) за атомичне трансакције
- **Control field:** 6 бита
 - Дужина података: описано са 4 бита – вредности 0, 1, 2, 4, 8
 - Продужени формат: 1 бит | неискоришћен 1 бит
- **Data field:** 0-8 октета
- **CRC field:** 15 бита праћено са рецесивним битом за одвајање
- **Ack field:** 1 бит потврде (доминантно уколико је примљено)
- **END of frame delimiter:** 7 рецесивних бита (нарушавање уметања)



ОКВИР ПОДАКА

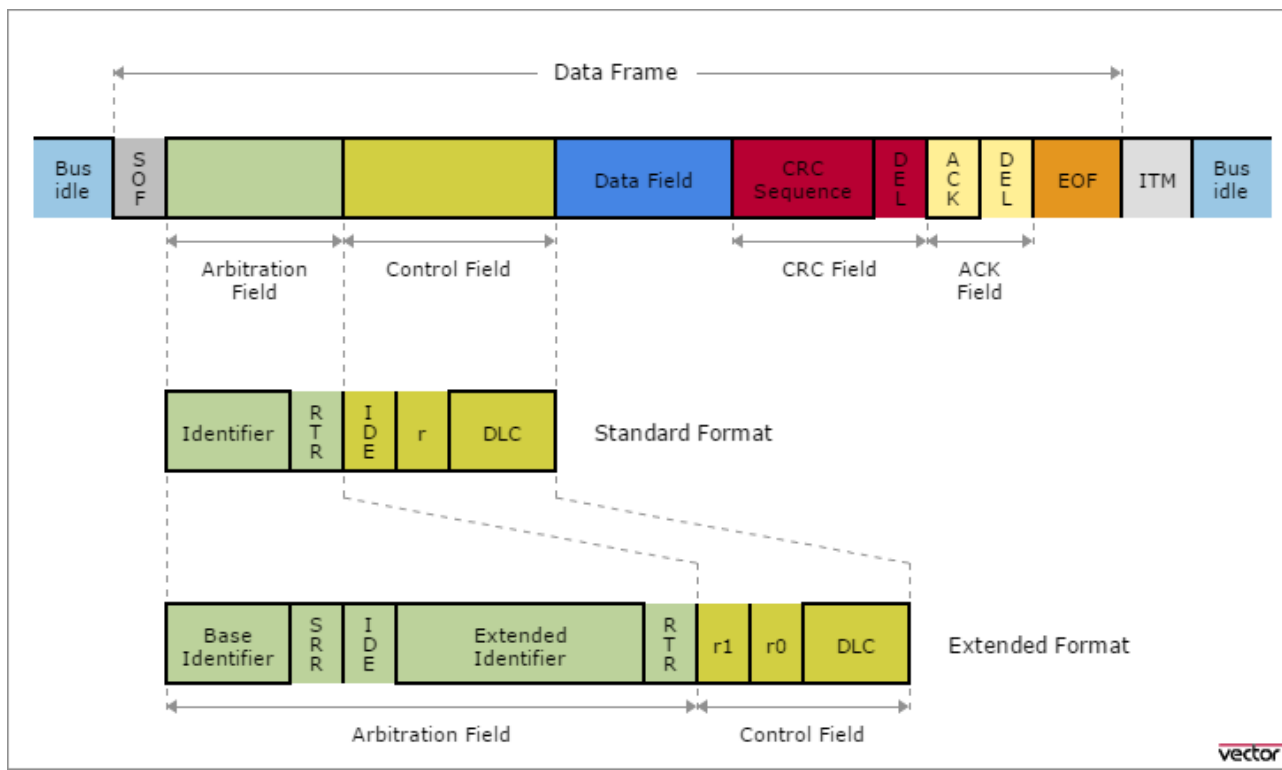
ДОДАТНИ ДЕТАЉИ

- Свако поље има тачан задатак
- Задаци: усклађивање такта, успостављање везе, слање и заштита података
- Почетак: (**Start of Frame — SOF**) – ивица преласка из стања мировања – усклађивање целе мреже
- Усклађивање такта: свака силазна ивица (рецесивно-доминантно) – фазно прилагођавање
- Идентификатор: приоритет и прихватање везе у матрици
- RTR (Remote Transmission Request): доминантно за податке а рецесивно за удаљено слање
- IDE бит: стандардни и проширени формат идентификатора
- DLC (Data Length Code): број октета података
- Заштита података: Cyclic redundancy check (**CRC**) завршава се са разделник битом
- Пријемник на основу провере CRC убацује бит за потврду (ACK) и разделник
- Завршетак: 7 рецесивних бита (End Of Frame —**EOF**).

ИДЕНТИФИКАТОР

ДВЕ ВЕРЗИЈЕ ПРОТОКОЛА

- Стандардни идентификатор - 11 бита - 2048 ID – V2.0A
- Проширени идентификатор - 29 бита – 536M ID – V2.0B

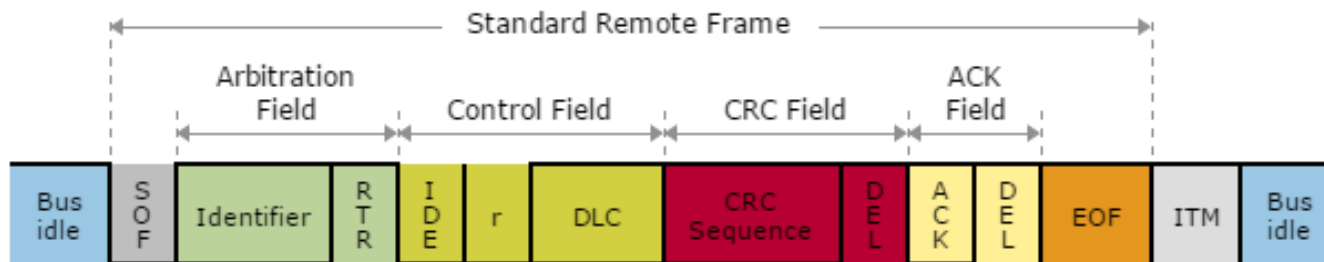


КВИЗ

- Зашто су нам потребни проширени идентификатори?
- Који се идентификатори превасходно користе у пракси?
 - CAN 2.0A (11-bit identifiers)
 - CAN 2.0B (29-bit identifiers)?
- Разлози
 - Губитак ефикасности и брзине
 - Смањење поузданости (више бита за грешке)
 - Повећање горњег прага кашњења

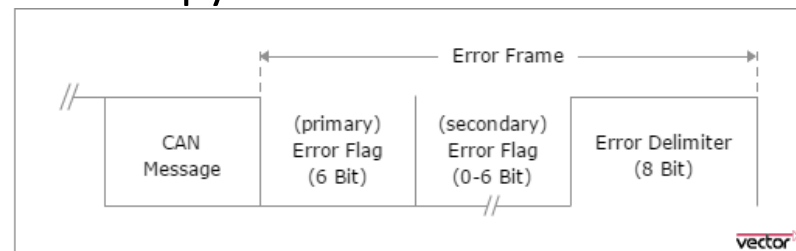
УДАЉЕНИ ОКВИР

- енг. Remote frame
- Употреба: захтевање података од удаљених уређаја
- Структура истоветна са оквиром података: недостају подаци



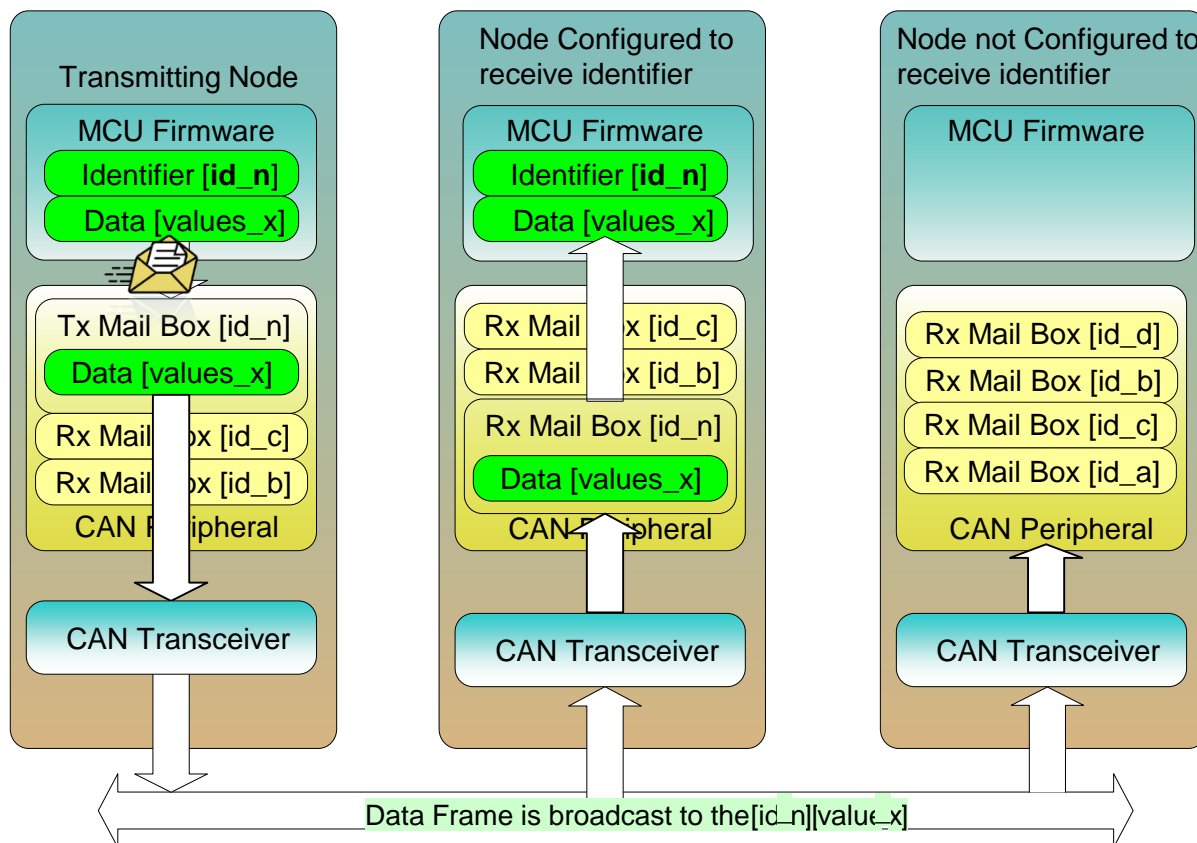
ОКВИР ГРЕШКИ

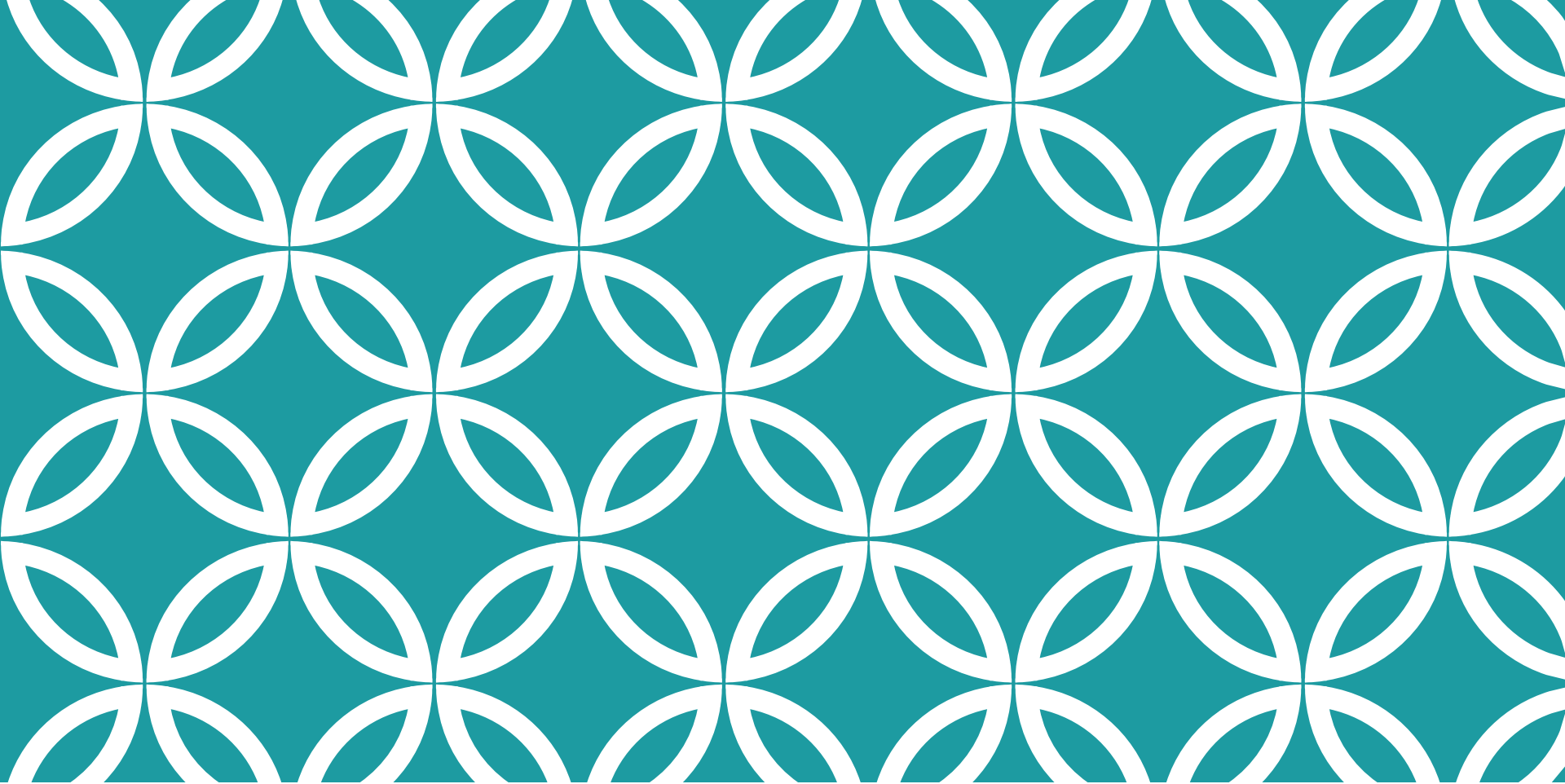
- енг. **Error frame**
- Употреба: упозорење пошиљаоцу да је пријем изобличен
- Шаље се одмах по препознавању грешке а текући пренос се прекида
- Важно: извор недетерминистичког понашања: нарушава се распоред
- Структура: упрошћено - два поља
 - Заставица (енг. flag): 6 доминантних бита
 - Граничник: 8 рецесивних бита
- Изобличења: грешка са уметањем бита или грешка са CRC
- Напомене:
 - Заставица нарушава правило уметања (5 бита) осим ако подлегне и она грешци
 - Пошиљалац након губитка арбитражу (претпражњење) покушава поновно слање



ТОК ПОДАТАКА

- Филтрирање и маске: вежба





АРБИТРАЖА

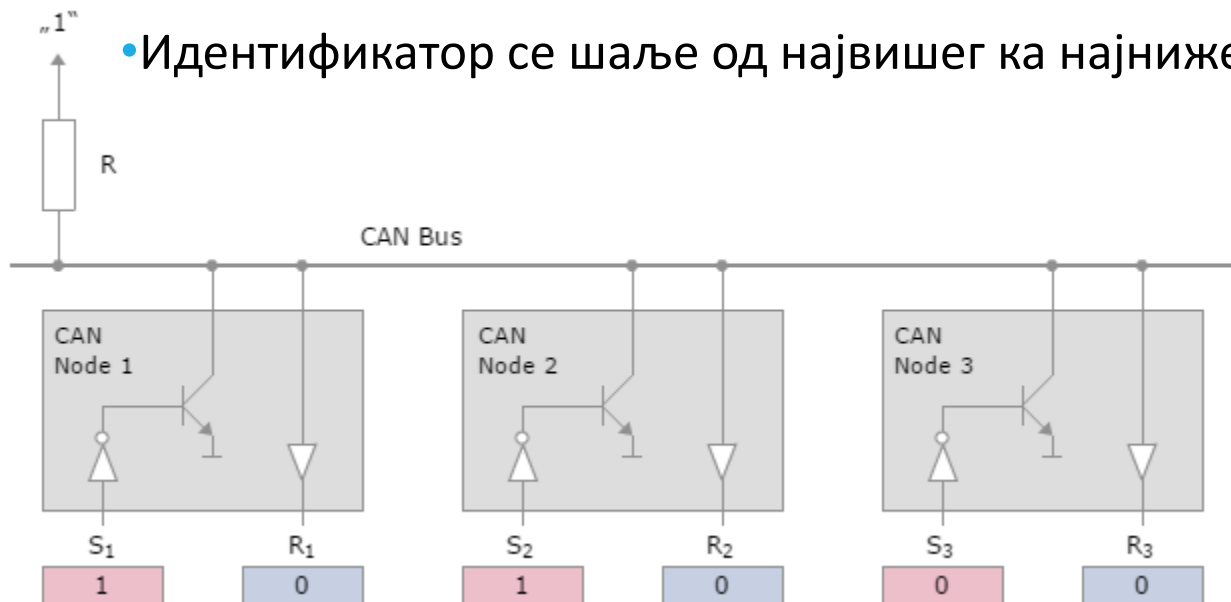
Контрола приступа
Решавање судара

ПРИСТУП МЕДИЈУМУ

- Користи се CSMA/CD
- Код диференцијалног се зове CSMA/BA (енг. Bitwise Arbitration)
- Судар не доводи до губитка порука: нема бацања пропусне моћи
- Означавање порука са идентификатором: приоритет
- Сваки уређај има јединствени идентификатор

АРБИТРАЖА

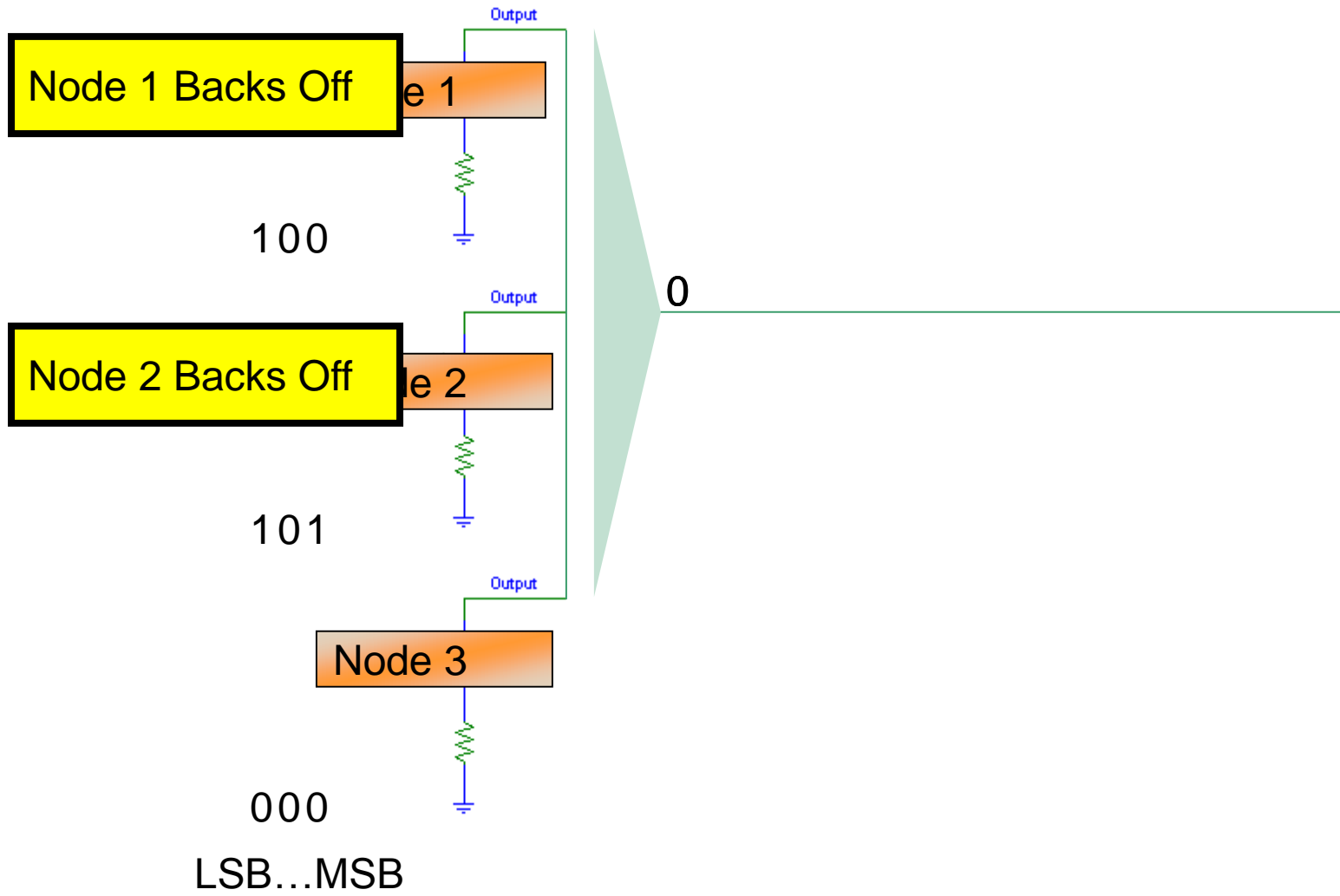
- Слање почиње слањем сопственог идентификатора
- Одступање од слања:
 - уколико се препозна доминантан бит док се шаље пасиван бит
- Побеђују уређаји са најмањим идентификатором: приоритет
- Идентификатор се шаље од највишег ка најнижем биту – **зашто?**



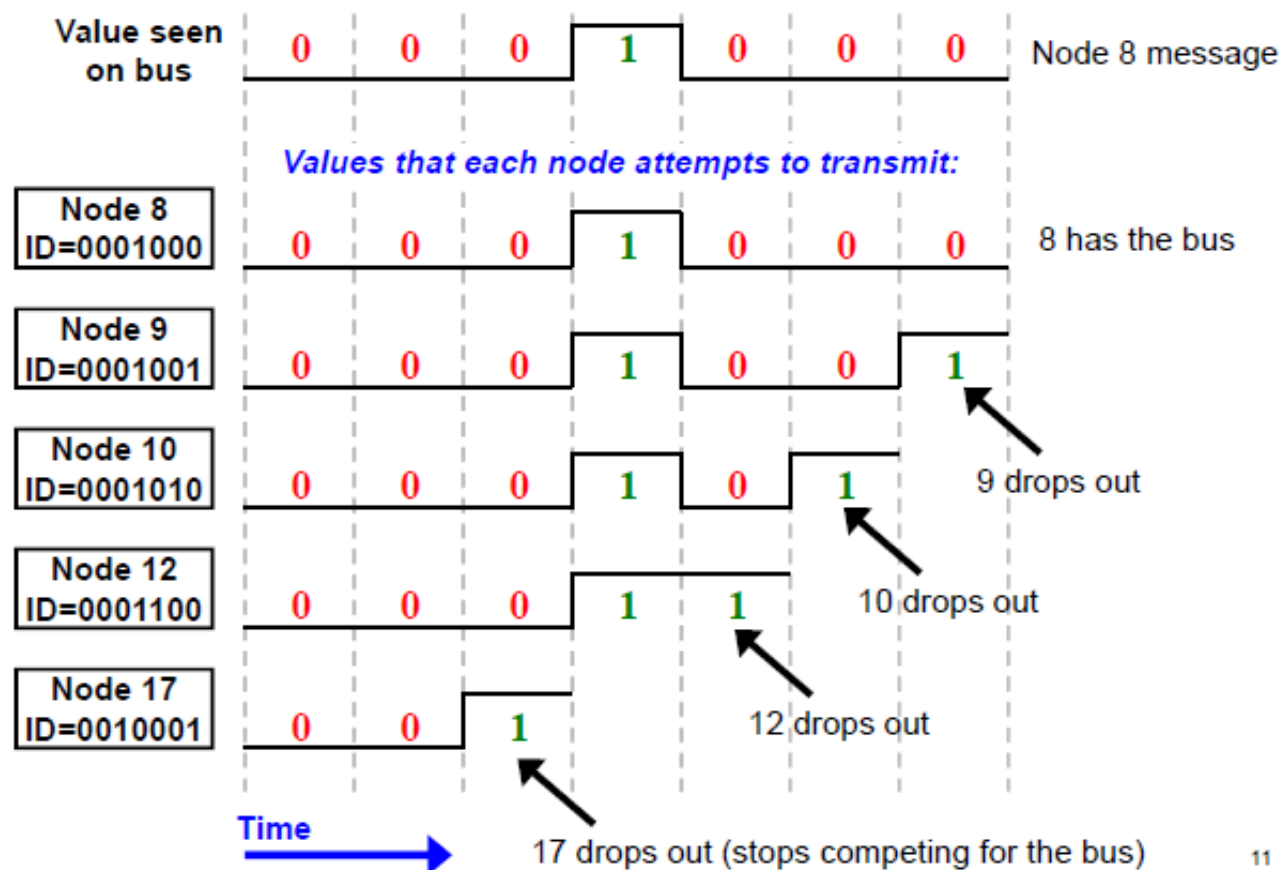
Ко осваја арбитражу?

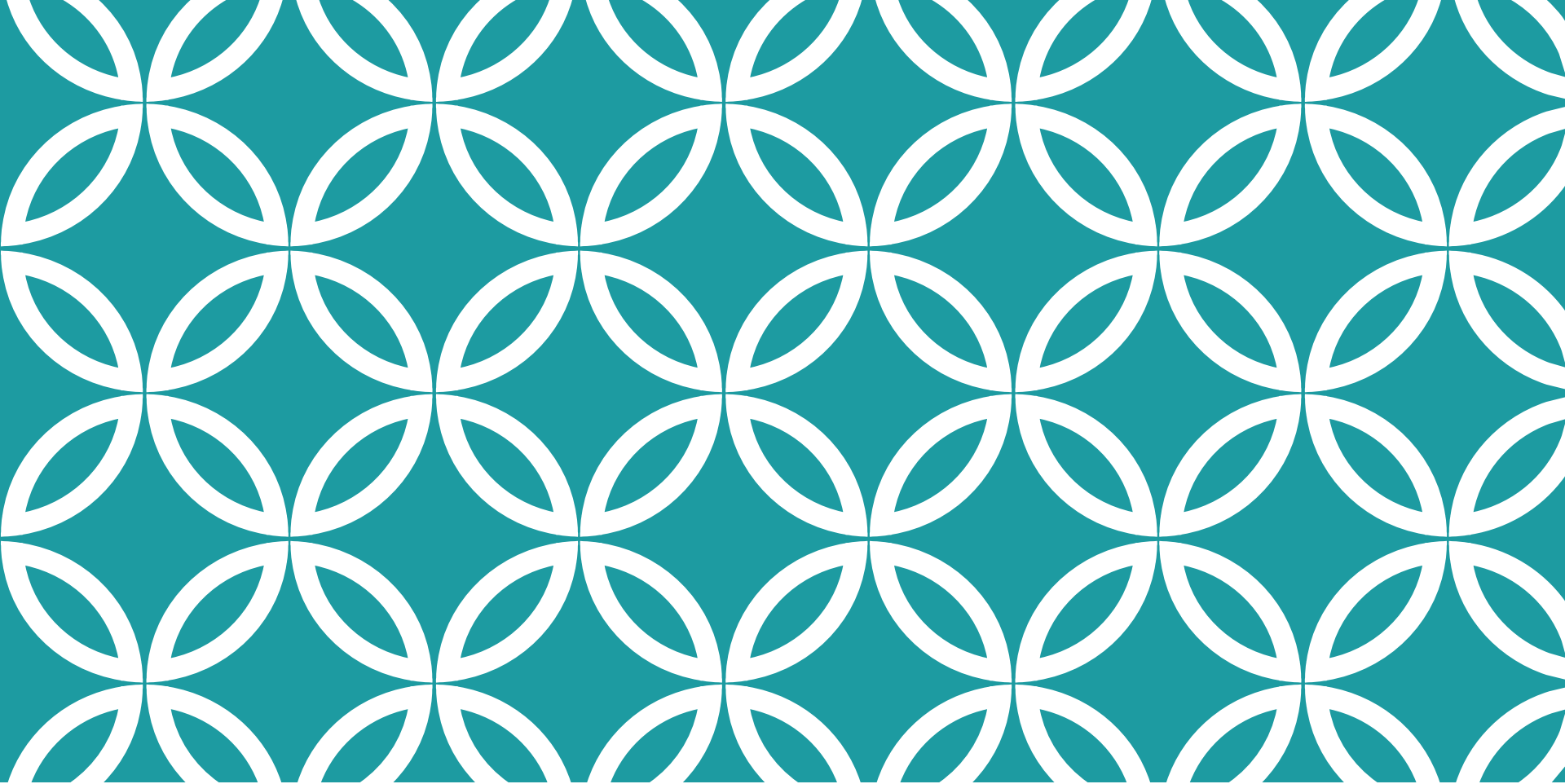
ПРИМЕР

ДОМИНИРАЈУЋИ БИТ



ПРИМЕР



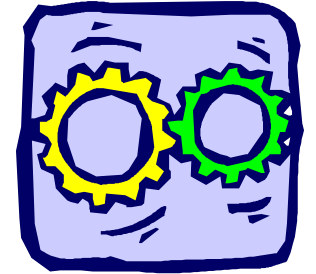


ПРЕПОЗНАВАЊЕ ГРЕШАКА

CRC

Убацивање бита

Каскадне грешке



УЧЕСТАНОСТ ГРЕШАКА

- Статистички учестаност грешака зависи од целог окружења:
 - Укупног броја уређаја
 - Физичког ожичења и расореда
 - Спољашњих ЕМ сметњи
- Пример:
 - 2000 сати/годишње, 500 Kbs, 25% оптерећење магистрале
 - Учестаност непрепознатих грешака: 1 у 1000 година

ТОК ПРЕПОЗНАВАЊА ГРЕШАКА

- Дистрибуирано између пошиљаоца (Tx) и примаоца (Rx)
- Постоји 5 одвојених корака (механизма) препознавања грешака:
 1. Надгледање појединачних бита (Tx)
 2. Провера структуре
 3. Провера кодовања (уметања бита)
 4. Провера потврде (Tx)
 5. Провера контролног поља
- Независно од филтрирања и маскирања на пријему

КОРАЦИ ПРЕПОЗНАВАЊА ГРЕШАКА

- Надгледање појединачних бита:
 - препознавање локалних и глобалних грешака код пошиљаоца
 - упоређује се стање на магистрали са послатим битом
 - не примењује се на поље за арбитражу – **зашто?**
- Провера структуре: пријемник
 - Типична заједничка поља: CRC delimiter, ACK delimiter и EOF – увек рецесивни бити

КОРАЦИ ПРЕПОЗНАВАЊА ГРЕШАКА

- Провера кодовања (уметања бита): пријемник проверава ток бита
 - На сваких 5 узастопних истоветних бита мора да дође убачена промена
 - Укључује све бите од SOF до краја CRC
- Провера контролног поља: **cyclic redundancy check (CRC)**
 - Пошиљалац рачуна пре слања, пријемник проверава подударање при пријему
- Провера потврде: обавеза пошиљаоца
 - Потребно бар да пристигне једна потврда
 - Пошиљалац подеси рецесивни бит, потребно да пријем обори са доминантним

ЦИКЛИЧНА РЕДУНДАНТНА ПРОВЕРА

- Принцип:

- полином од 16 бита $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + x^0$
- Хамингово растојање: 6

- Одлике (важи за фиксне дужине података):

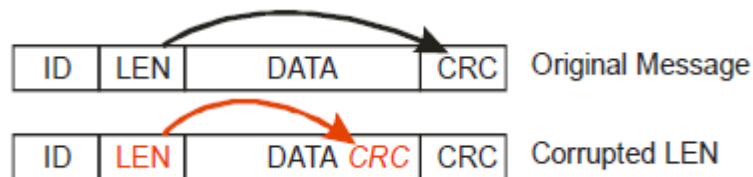
- Препознаје рафалне грешке до 15 бита заредом
- Препознаје грешку од 5 насумичних бита

- Ограничења: уметање бита (променљива дужина података)

- Хамингово растојање: 2 – могућност препознавање 2 бита
- Увек препознаје грешку од насумичног 1 бита

НЕЗАШТИЋЕНО ЗАГЛАВЉЕ

- CAN не штити поље дужине пакета
 - Могуће грешке од 1 бита
 - Грешке доводе до погрешног показивања позиције CRC!



- Пример решења: FlexRay - посебна CRC заглавља

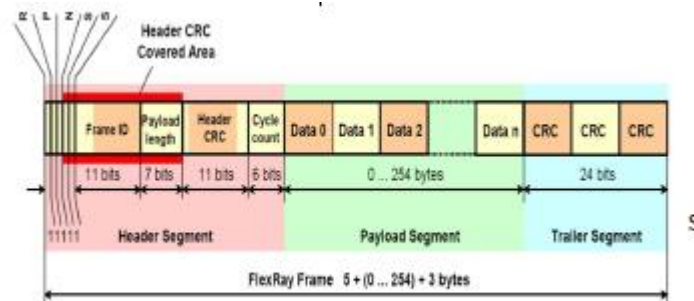
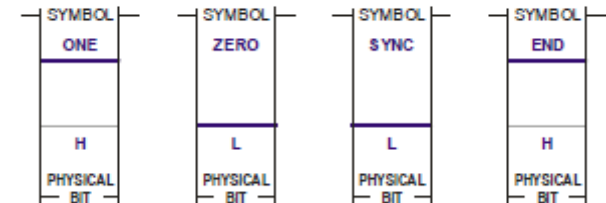


Figure 4-1: FlexRay frame format.

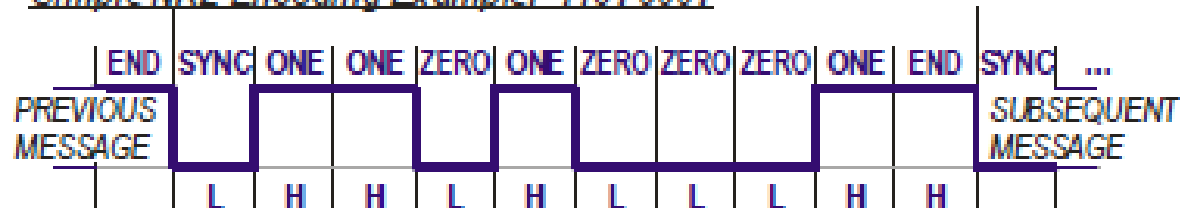
NON-RETURN TO ZERO (NRZ)

- Кодовање као код серијских магистрала
- Велике брзине, мале емисије
- Означавање бита: два стања: високо (1) и ниско (0)
- Означавање почетка и краја: супротна стања (1/0)
- Могућност губитка синхронизације:
 - У најгорем случају: исте вредности бита
 - Најлакше да се ограничи подаци – 8 бита

Simple NRZ Bit Encoding



Simple NRZ Encoding Example: 1101 0001

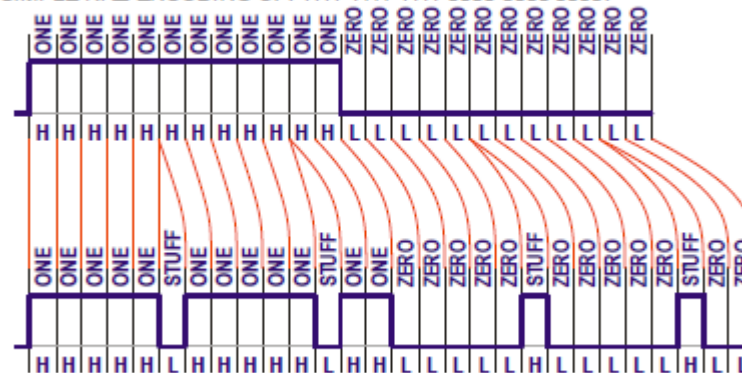


УБАЦИВАЊЕ БИТА

- енг. *Bit Stuffing*
- Побољшање усклађивања:
 - Додавање бита који праве промену – ивица за усклађивање
- Додати бити не мењају податке (HW)
- Приступ: после 5 идентичних убацити супротан бит
- Генерално није потребно: врло ретко иду истоветне ниске

BIT STUFF IDEA:

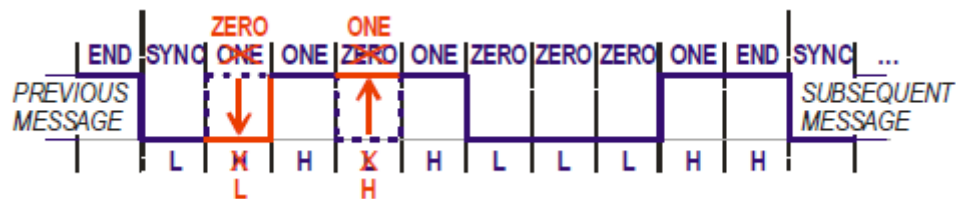
SIMPLE NRZ ENCODING OF: 1111 1111 1111 0000 0000 0000:



BIT-STUFFED NRZ ENCODING OF: 1111 1111 1111 0000 0000 0000:

ПОДЛОЖНОСТ ГРЕШКАМА

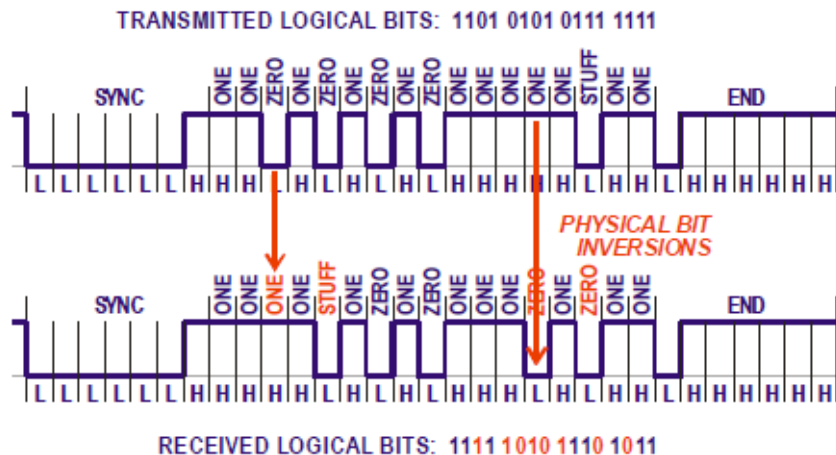
- **Појединачни инвертовани бити се тешко могу препознати**
 - Висока ефикасност долази по цени тешког препознавања грешке
 - Понекад CRC може да помогне (уз ограничења)
- **Препознавање грешака уметања:**
 - Нарушавање правила уметања означава грешку
 - Нпр. 6 идентичних бита заредом
 - Додуше постоје суптилнији проблеми са уметањем



КАСКАДНЕ ГРЕШКЕ

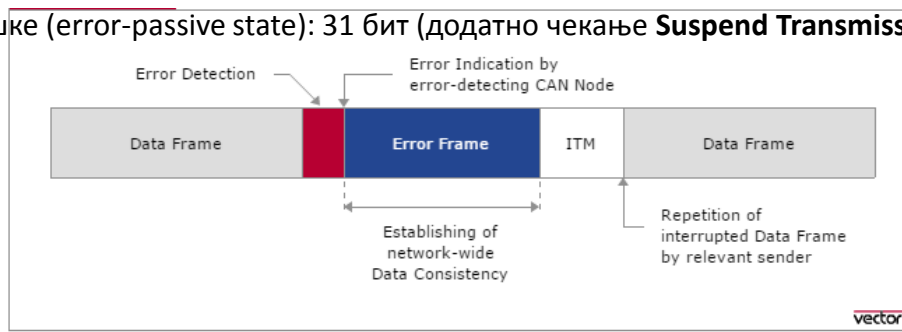
- **Комбинација инверзије и уметања бита:**
 - Могуће збуњивање механизма за декодовање уметнутих бита
- Најгоре грешке долазе у пару:
 - Прекид дугог низа: бит у средини инвертован
 - Подаци збуњени за убачени бит и обрнуто
 - Подаци задржавају исту дужину али остају погрешно замењени бити
 - Проблем: могуће да се грешка провуче кроз CRC – како?

- **Пример:**

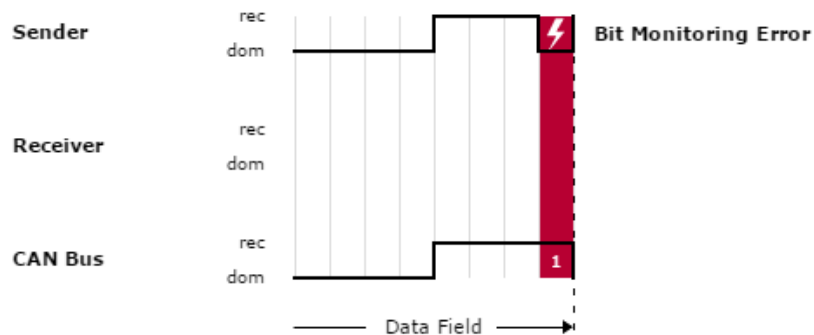


ГЛОБАЛНА ДОСЛЕДНОСТ

- Гаранција - локални поремећај потребно глобално огласити:
 - оквир грешке са 6 доминантних бита нарушава правило уметања
- Примарна заставица служи за пошиљаоца а секундарна за остале пошиљаоце
 - Могуће преклапање
- Након грешке постоји чекање до следећег слања које зависи:
 - Најбољи случај: 17 бита (примарна заставица, граничник, интермецо пауза) (ITM — Intermission)).
 - Без поклапања заставица: 23 бита
 - Пасивно стање грешке (error-passive state): 31 бит (додатно чекање **Suspend Transmission Time**)



ПРИМЕР РУКОВАЊА ГРЕШКИ



Event 1

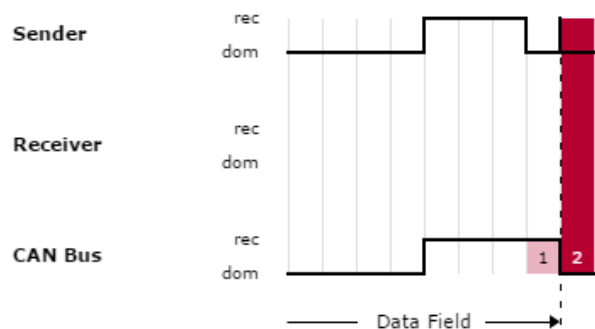
During the transmission of a data frame's data field the **sender** detects a discrepancy between the dominant level that was sent and the recessive level that was detected on the CAN bus (bit monitoring error, see lightning).



Effect

After the detection of the bit monitoring error the **sender** transmits an error flag during the next bit time interval. Thereby terminating the transmission of the data frame.

ПРИМЕР РУКОВАЊА ГРЕШКИ



Event 2

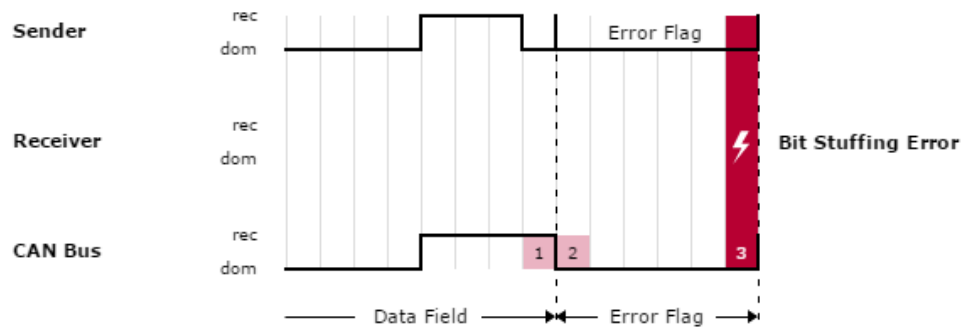
The **sender** transmits the first dominant bit of the error flag. Thereby the transmission of the data frame has been terminated.



Effect

The **sender** transmits five more dominant bits (an error flag always consists of six dominant bits).

ПРИМЕР РУКОВАЊА ГРЕШКИ



Event 3

After the sixth dominant bit has been transmitted by the **sender**, the error flag is complete.

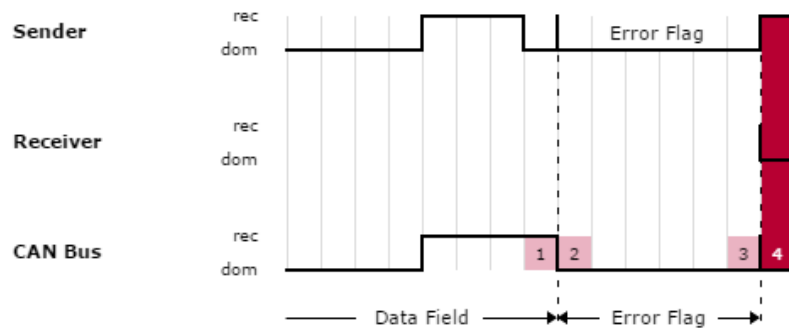
With receipt of the sixth dominant bit the **receiver** detects a bit stuffing error (see lightning).

Effect

After the detection of the bit stuffing error the **receiver** transmits an error flag in the next bit time interval. Thereby terminating the receipt of the data frame.

In the next bit time interval the **sender** transmits the error delimiter.

ПРИМЕР РУКОВАЊА ГРЕШКИ



Event 4

The **receiver** transmits the first dominant bit of its error flag. Thereby it terminates the receipt of the data frame.

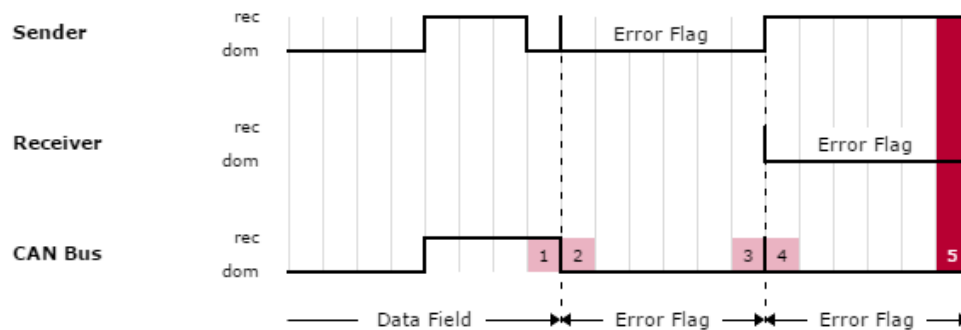
The **sender** transmits the first recessive bit of its error delimiter. However, the recessive bit does not appear on the CAN bus.

Effect

The **sender** will continue to transmit recessive bits until one of it appears on the CAN bus. If this happens the **sender** will transmit seven more recessive bits, as the error delimiter always consists of eight recessive bits.

The **receiver** continues the transmission of the error flag (an error flag always consists of six dominant bits).

ПРИМЕР РУКОВАЊА ГРЕШКИ



Event 5

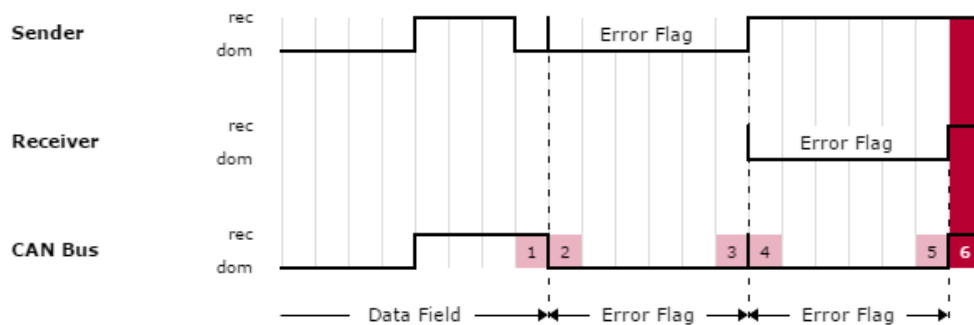
The transmission of the error flag sent by the **receiver** is completed after the sixth dominant bit.

Effect

During the next bit time interval the **receiver** transmits its error delimiter.

The **sender** continues the transmission of recessive bits.

ПРИМЕР РУКОВАЊА ГРЕШКИ



Event 6

The **receiver** transmits the first recessive bit of its error delimiter.

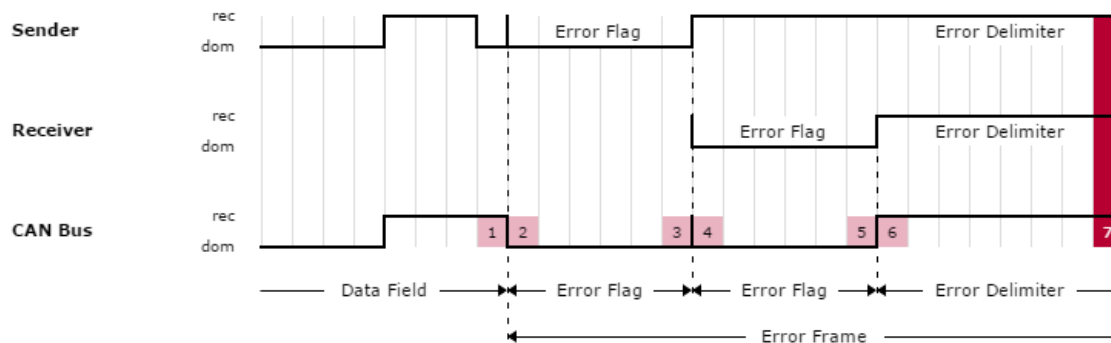
For the first time a recessive bit transmitted by the **sender** appears on the CAN bus.



Effect

Sender and **receiver** transmit seven more recessive bits in order to complete their respective error delimiters.

ПРИМЕР РУКОВАЊА ГРЕШКИ



Event 7

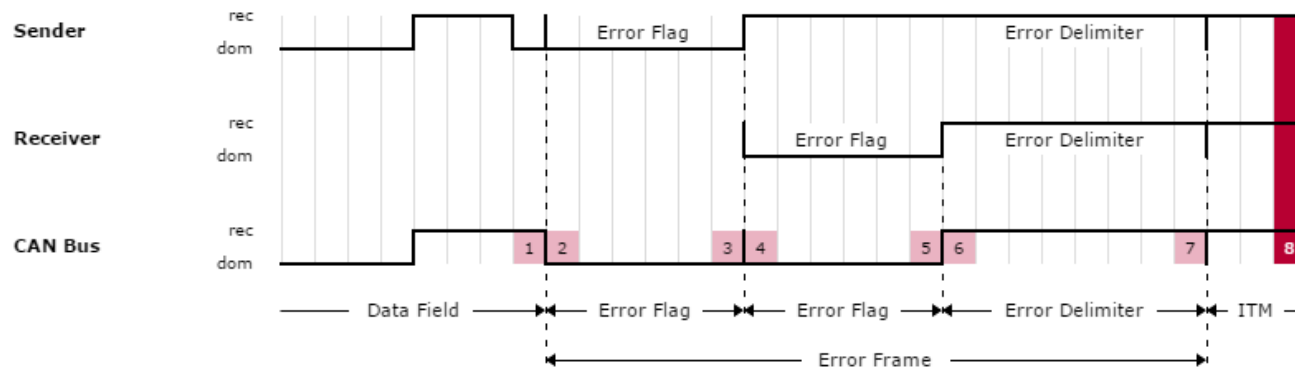
Sender and **receiver** have completed the transmission of their error delimiters. Both of the error flags together with the error delimiters add up to the error frame. The first error flag is called primary error flag, the second is referred to as secondary error flag.



Effect

The **sender** has to wait for three recessive bit time intervals (ITM: Intermission), before it can access the CAN bus to try to send the aborted data frame again.

ПРИМЕР РУКОВАЊА ГРЕШКИ



Event 8

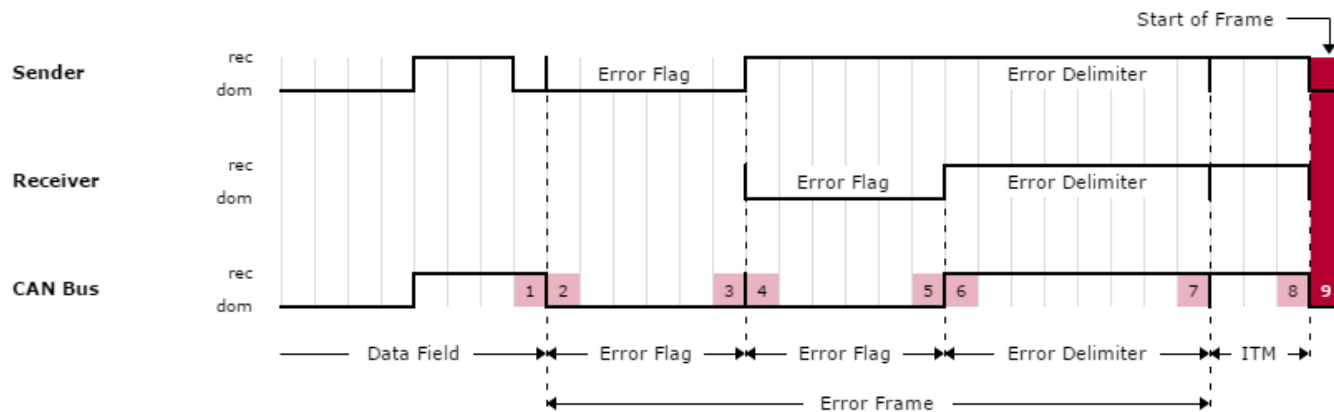
ITM (Intermission) is complete and the CAN bus is idle.



Effect

The **sender** accesses the CAN bus in order to transmit the aborted data frame again.

ПРИМЕР РУКОВАЊА ГРЕШКИ



Event 9

The **sender** gets access to the CAN bus to transmit the aborted data frame again. In order to do so the **sender** transmits the SOF (Start of Frame) bit first.



Effect

If there are no other **senders** accessing the CAN bus, the aborted data frame can be transmitted immediately. In the case of simultaneous bus access the data frame's priority determines the point in time of the transmission.

ПРАЋЕЊЕ ГРЕШАКА

УВОД

- Уређаји увек имају право да глобално огласе препознату грешку
 - Подложно ефекту „лажна узбуна“ – могуће рушење мреже код неисправног препознавања
- Механизам који прати и класификује грешке:
 - Избегавање загушења магистрале
 - Препознавање пролазне грешке и сталне
- Бројачи: TEC (Transmit Error Counter) и REC (Receive Error Counter).
 - Успешно слање/пријем: смањење за 1 бројача ($TEC=TEC-1$; $REC=REC-1$)
 - Препознавање грешке: повећање бројача према правилима
 - Пошиљалац: $TEC=TEC+8$.
 - Прималац који препозна грешку: $REC=REC+1$
 - Прималац који изазове грешку: $REC=REC+8$

ПРАЋЕЊЕ ГРЕШАКА

- Дијаграм стања: прелази дефинисани

- Нормално стање: Error Active

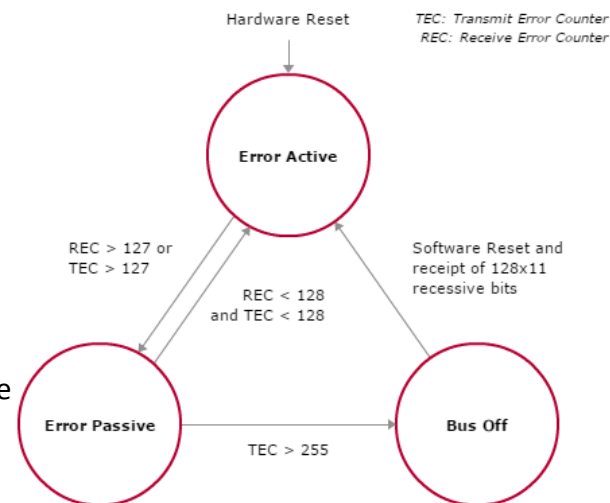
- Подразумевано по покретању
 - Шаљу се редовне заставице од 6 доминантних бита
 - После прекорачења ($TEC > 127$; $REC > 127$) прелази се у пасивно стање

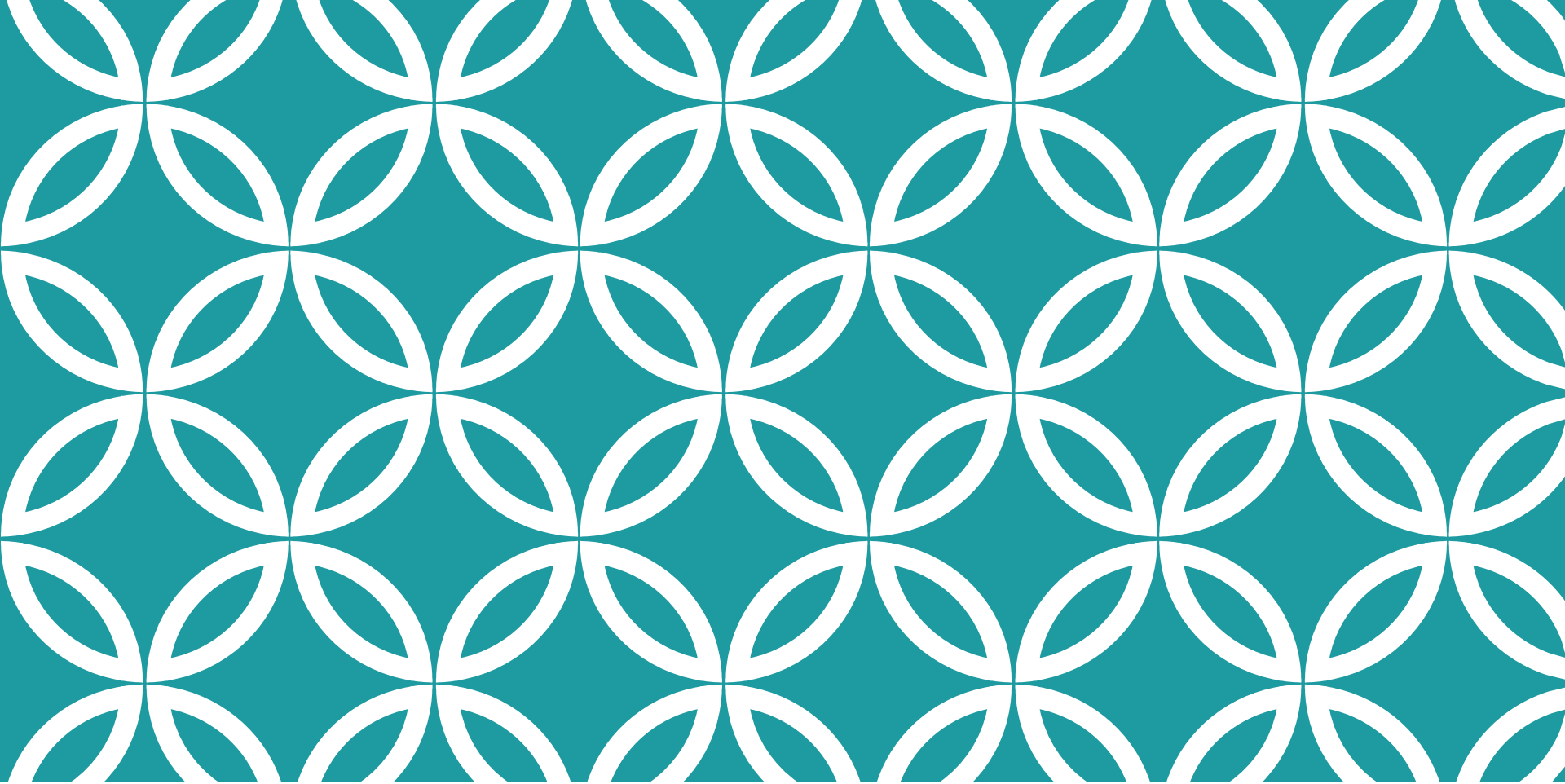
- Пасивно стање: Error Passive

- Шаљу се измењене заставице од 6 рецесивних бита
 - Спречавање пријемника који препознају грешку да се глобално огласе
 - Додатно потребно Suspend Transmission Time" (8 bits).

- Искључено стање: Bus-Off

- Прелази се након прекомерног гомилања грешака ($TEC > 255$) (обично у кратком року)
 - Управљачки CAN модул се одспоји са магистрале
 - Могући излаз:
 - Након локалне интервенције и после обавезног чекања од 128 x 11 бита
 - Након хардверског ресетовања





ЕФИКАСНОСТ

Бита
Оквир
Распреда

ЛИМИТИ ЕФИКАСНОСТИ

- **Ефикасност оквира**

- Пребачај мора да се минимизује

- **Ефикасност кодовања:**

- Обара се са додавањем бита: усклађивање и заштита од грешки

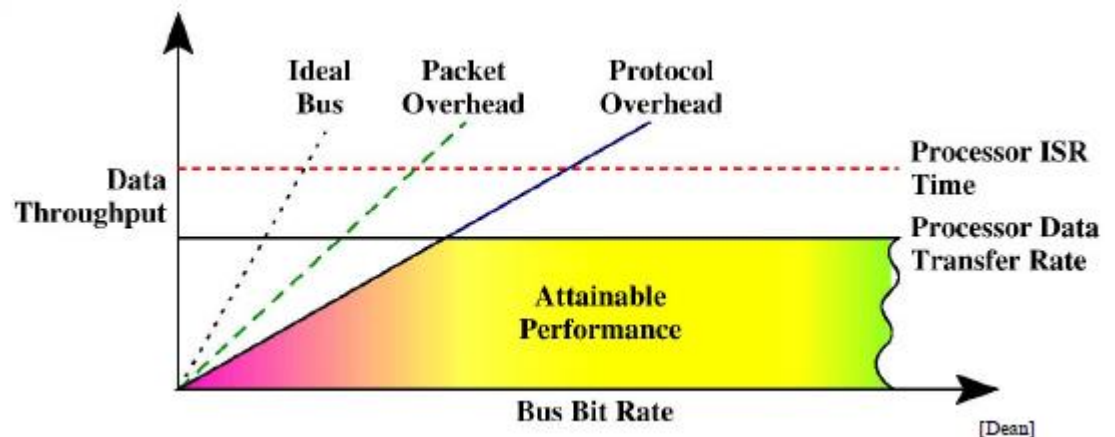
- **Арбитража:**

- Колизиије, понављања, грешке

- **Процесорска брзина**

- **Прављење распореда**

- **Заказивање**



БИТСКА ЕФИКАСНОСТ

- Ефикасност: проценат корисних бита података наспрам укупно послатог
- Премашај: „некорисни“ контролни бити)
- Пример: 64 бита података, заглавља (премашај) 76 бита -> ефикасност = 46%
- Најгора процена за CAN
 - Премашај: 67 бита
 - Заглавље: идентификатор 29 бита (блага промена за ID од 11 бита)
 - CRC: 15 бита | Дужина: 4 бита
 - Старт и статуси: 9 бита | Стоп и чекања: 10 бита
 - Подаци: s_m – број октета
 - Убацавање бита:
 - у најгорем случају 1 бит на 5 бита поруке
 - У обзир долазе само 54 бита премашаја и подаци

$$\#bits = \left(\left\lfloor \frac{54 + 8s_m}{5} \right\rfloor + 67 + 8s_m \right)$$

ЕФИКАСНОСТ ОКВИРА

- Како проценити?
 - Оквири са подацима наспрам свих оквира (контролни, усклађивање)
 - Претпоставка: униформна дужина свих порука (донекле тачно)
- Системи са руководиоцем и подређеним
 - Груба процена ефикасности: ~50% (пола су сами захтеви)
- Систем са жетоном (токеном)
 - Ефикасност расте са оптерећењем (више података)
 - Ефикасност зависи од спајања токена и података
- Системи подложни сударима:
 - Ефикасност опада са оптерећењем јер расте број симултаних преноса
- CAN:
 - Ефикасност: близу 100% - **зашто?** (наговештај Remote Frames)

ПОВЕЋАВАЊЕ ЕФИКАСНОСТИ

ТРИКОВИ

- Комбиновање више података у веће оквире
 - Пажња: ради само код спајања оквира истог пошиљаоца
 - Опрез: окидачи догађаја се не спајају – губитак догађаја
- Планирање размака – смањен премашај арбитраже
 - Насумично смицање почетака слања
 - Временско окидање: TDMA
 - Могућа ефикасност од 100%
 - Потребан почетни контролни премашак
 - Захтева прецизно усклађивање такта

ОПТЕРЕЋЕЊЕ

ПРОСЕЧНО И У ШПИЦУ

- **Просечно оптерећење:** засновано на просечним периодима

Total = 55 messages / 30 time units

- Периодични саобраћај: дефинисан период
- Спорадични саобраћај: окидање у оквиру дефинисаног периода
- Претпоставка да неће доћи до нагомилавања саобраћаја

Message Name	Type	Mean Period	# in 30 units
A	Periodic	1	30
B	Sporadic	5	6
C	Periodic	5	6
D	Periodic	10	3
E	Periodic	5	6
F	Sporadic	10	3
G	Periodic	30	1

- **Оптерећење у шпицу:** засновано на просецима и минималним временима

- Периодични саобраћај: дефинисан период
- Спорадични саобраћај: окидање на минимални период
- Претпоставка да неће доћи до нагомилавања саобраћаја

Message Name	Type	Mean Period	Peak Period	# in 30 units
A	Periodic	1		30
B	Sporadic	5	2	15
C	Periodic	5		6
D	Periodic	10		3
E	Periodic	5		6
F	Sporadic	10	10	3
G	Periodic	30		1

Total = 64 messages / 30 time units

- **За критичне системе потребно поопштрити критеријуме**

ПРАВЉЕЊЕ РАСПОРЕДА

ТРИВИЈАЛНА ОГРАНИЧЕЊА

- Пронаћи најкраће ограничење и покушату све сместити

- Пример:

- Најкраће ограничење за прорез: 5 ms
- Величина овира: 90 бита
- Укупно оквира: 20 (укупно величина 1800 бита)
- Потребна брзина: $1800 \text{ бита} / 5 \text{ ms} = 360 \text{ kbs}$
- Тривијалан проблем уколико обезбедимо брзину

<i>Signal Description</i>	<i>Size /bits</i>	<i>Deadline /ms</i>	<i>From</i>	<i>bits/ message</i>	<i>bits/ second</i>
Accel_Posn	8	5	Driver	90	18000
Brake_Master	8	5	Brakes	90	18000
Brake_Line	8	5	Brakes	90	18000
Trans_Clutch	8	5	Trans	90	18000
Contactora	4	5	Battery	90	18000
Key_Run	1	20 -> 5	Driver	90	18000
Key_Start	1	20 -> 5	Driver	90	18000
Accel_Switch	2	20 -> 5	Driver	90	18000
Brake_Switch	1	20 -> 5	Brakes	90	18000
Emer_Brake	1	20 -> 5	Driver	90	18000
Shift_Lever	3	20 -> 5	Driver	90	18000
T_Batt_V	8	100 -> 5	Battery	90	18000
T_Batt_C	8	100 -> 5	Battery	90	18000
A_Batt_V	8	100 -> 5	Battery	90	18000
A_Batt_C	8	100 -> 5	Battery	90	18000
Trans_Lube	8	100 -> 5	Trans	90	18000
Speed	8	100 -> 5	Brakes	90	18000
T_Batt_Tave	8	1000 -> 5	Battery	90	18000
T_Batt_Tmax	8	1000 -> 5	Battery	90	18000
T_Batt_GF	1	1000 -> 5	Battery	90	18000
Total bits/sec =					360,000

ПРАВЉЕЊЕ РАСПОРЕДА

МОНОТНИ РОКОВИ

- Додела приоритета на основу вредности временског рока
- Претпоставка: оптерећење < ~100%
- Пример:
 - Могуће заказивање
 - Потребна брзина: ~125 kbs
 - Н.Б. Рокови су хармонични

Signal Number	Priority	Signal Description	Size /bits	J /ms	Period /ms	Periodic /Sporadic	Deadline /ms	From	bits/ message	bits/ second
7	1	Accel_Posn	8	0.1	5	P	5	Driver	90	18000
8	2	Brake_Master	8	0.1	5	P	5	Brakes	90	18000
9	3	Brake_Line	8	0.2	5	P	5	Brakes	90	18000
11	4	Trans_Clutch	8	0.1	5	P	5	Trans	90	18000
14	5	Contactor	4	0.1	50	S	5	Battery	90	18000
15	6	Key_Run	1	0.2	50	S	20	Driver	90	4500
16	7	Key_Start	1	0.3	50	S	20	Driver	90	4500
17	8	Accel_Switch	2	0.4	50	S	20	Driver	90	4500
18	9	Brake_Switch	1	0.3	20	S	20	Brakes	90	4500
19	10	Emer_Brake	1	0.5	50	S	20	Driver	90	4500
20	11	Shift_Lever	3	0.6	50	S	20	Driver	90	4500
1	12	T_Batt_V	8	0.6	100	P	100	Battery	90	900
2	13	T_Batt_C	8	0.7	100	P	100	Battery	90	900
4	14	A_Batt_V	8	0.8	100	P	100	Battery	90	900
6	15	A_Batt_C	8	0.9	100	P	100	Battery	90	900
10	16	Trans_Lube	8	0.2	100	P	100	Trans	90	900
12	17	Speed	8	0.4	100	P	100	Brakes	90	900
3	18	T_Batt_Tave	8	1.0	1000	P	1000	Battery	90	90
5	19	T_Batt_Tmax	8	1.1	1000	P	1000	Battery	90	90
13	20	T_Batt_GF	1	1.2	1000	P	1000	Battery	90	90
Total bits/sec =									122,670	

<i>Signal Number</i>	<i>Signal Description</i>	<i>Size /bits</i>	<i>J /ms</i>	<i>Period /ms</i>	<i>Periodic /Sporadic</i>	<i>Deadline /ms</i>	<i>From</i>	<i>bits/ message</i>	<i>bits/ second</i>	
14	Contactior	4	0.1	50	S	5	Battery	90	18000	
8	Brake Master	8	0.1	5	P	5	Brakes	90	18000	} Brake msg
9	Brake Line	8	0.2	5	P	5	Brakes	90	18000	
7	Accel_Posn	8	0.1	5	P	5	Driver	90	18000	
11	Trans_Clutch	8	0.1	5	P	5	Trans	90	18000	
18	Brake_Switch	1	0.3	20	S	20	Brakes	90	4500	
15	Key_Run	1	0.2	50	S	20	Driver	90	4500	} Driver msg
16	Key_Start	1	0.3	50	S	20	Driver	90	4500	
17	Accel_Switch	2	0.4	50	S	20	Driver	90	4500	
19	Emer_Brake	1	0.5	50	S	20	Driver	90	4500	
20	Shift_Lever	3	0.6	50	S	20	Driver	90	4500	
1	T_Batt_V	8	0.6	100	P	100	Battery	90	900	} Batt msg1
2	T_Batt_C	8	0.7	100	P	100	Battery	90	900	
4	A_Batt_V	8	0.8	100	P	100	Battery	90	900	
6	A_Batt_C	8	0.9	100	P	100	Battery	90	900	
12	Speed	8	0.4	100	P	100	Brakes	90	900	
10	Trans_Lube	8	0.2	100	P	100	Trans	90	900	
3	T_Batt_Tave	8	1.0	1000	P	1000	Battery	90	90	} Batt msg2
5	T_Batt_Tmax	8	1.1	1000	P	1000	Battery	90	90	
13	T_Batt_GF	1	1.2	1000	P	1000	Battery	90	90	
							Total bits/sec =	122,670		

ПРАВЉЕЊЕ РАСПОРЕДА

ГРУПИСАЊЕ СЛИЧНИХ ОКВИРА

- **Предност:** уштеда у брзини 36 kbs - оптимално
- **Мане (нема бесплатног ручка):**
 - Промена структуре мреже: више извора за исту врсту поруке
 - Промена рокова за једну од комбинованих порука
 - Промена врсте саобраћаја: периодично и спорадично

Priority	Signal Number	Signal Description	Size /bits	J /ms	Period /ms	Periodic /Sporadic	Deadline /ms	From	bits/ message	bits/ second
1	14	Contactora	4	0.1	50	S	5	Battery	90	18000
2	8+9	Brake_msg	16	0.1	5	P	5	Brakes	100	20000
3	7	Accel_Posn	8	0.1	5	P	5	Driver	90	18000
4	11	Trans_Clutch	8	0.1	5	P	5	Trans	90	18000
5	18	Brake_Switch	1	0.3	20	S	20	Brakes	90	4500
6	15-17,19-20	Driver_msg	8	0.2	50	S	20	Driver	90	4500
7	1,2,4,6	Batt_msg1	32	0.6	100	P	100	Battery	120	1200
8	12	Speed	8	0.4	100	P	100	Brakes	90	900
9	10	Trans_Lube	8	0.2	100	P	100	Trans	90	900
10	3,5,13	Batt_msg2	17	1.0	1000	P	1000	Battery	110	110
Total bits/sec =									86,110	

КАШЊЕЊА

- Мрежа је серијски медијум: неко мора последњи у реду
- Кашњења са краја на крај имају више чиниоца:
 - Генерисање података: узорковање сензора, обрада апликације
 - Редови чекања: минимално – дужина саме поруке, макс дужина реда
 - Арбитража: предња ивица: оптимално иначе се чека следећи круг
 - Провера грешака на пријему и филтрирање
 - Реаговање на податке: актуатори, апликација на пријему
 - Грешке и понављање оквира: срећом је ACK/NACK 1 бит

ЗАКАЗИВАЊЕ СЛАЊА

ВРТЕШКА

- Сви извори се испитују у круг: енг. Round-Robin
- Најбољи случај: одмах се шаље
 - Празна мрежа
 - Погодан распоред
- Најгори случај: поновно чекање свог редоследа
 - Порука генерисана после редоследа: чекај све уређаје
 - Чекање свих претходних n порука да заврши

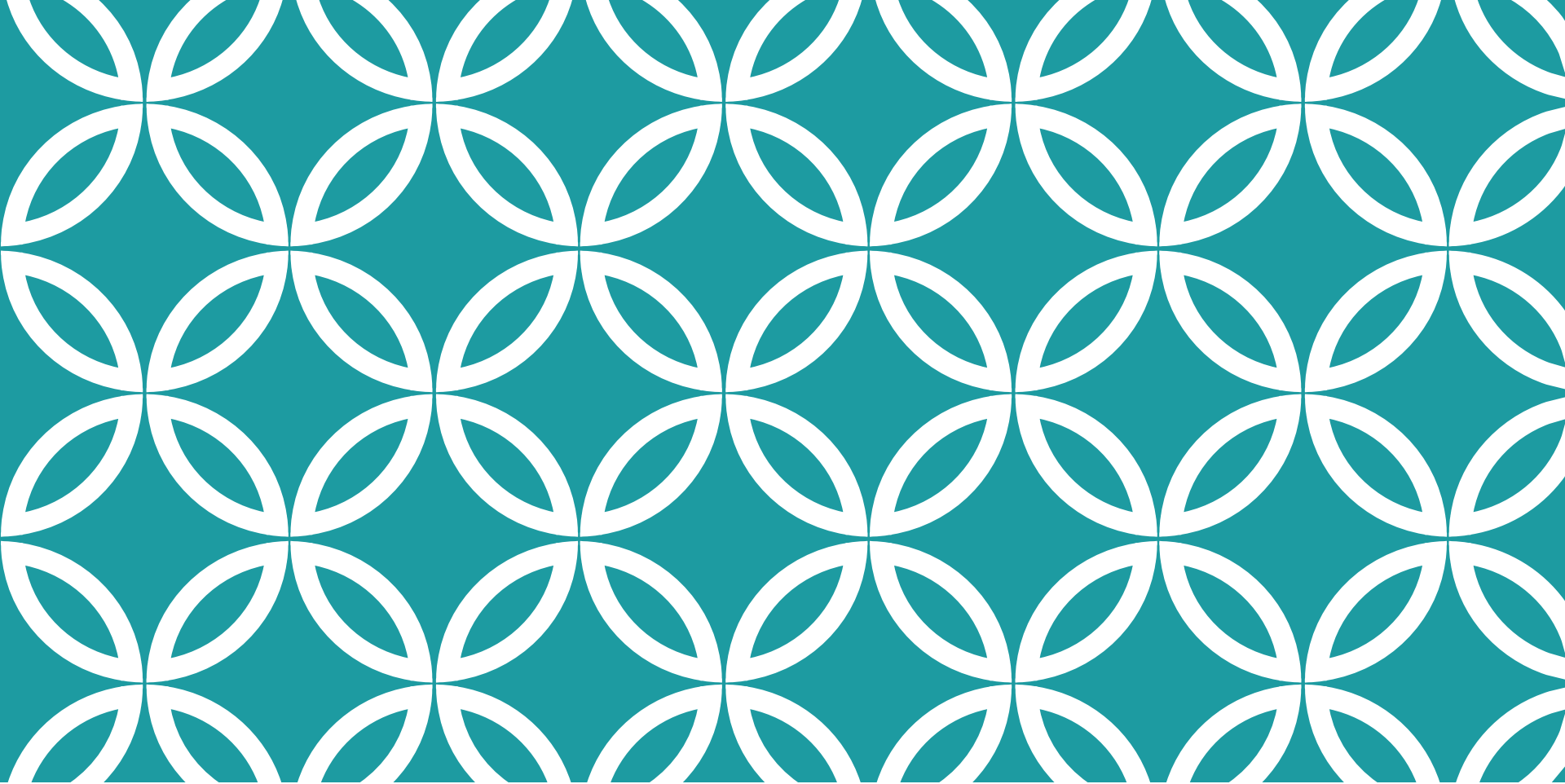
ЗАКАЗИВАЊЕ СЛАЊА

ПРИОРИТЕТИ

- Приоритети према идентификатору оквира
- Најбољи случај: слање одмах
- Најгори случај:
 - Заузет медијум: чекање на завршетак текућег преноса
 - Предност имају сви остали оквири

ПРЕКОРАЧЕЊЕ ОБРАДЕ

- **Подложност: спори пријемници а брзо слање порука**
 - Хардвер „ухвати“ све поруке а мК споро „вади“ поруке из spremника
- **Могућа решења:**
 - Постојање интермецо периода између слања
 - Постојање посебних сандучића за различите оквири
 - Груписање више порука у један оквир
 - Наменско заказивање оквира тако да нема прокорачења

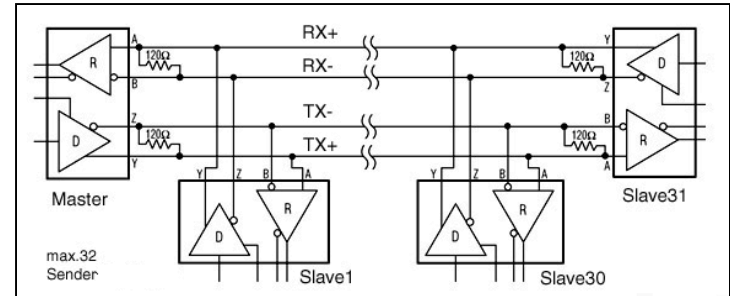


ПОРЕЂЕЊА



RS-485 VS. CAN

- Јесу то исте магистрале?
 - Сличне цене
 - Слични домет
 - Слична имуност на сметње
 - Слична доступност чипова
 - Сличне утичнице
 - Истоветан број уређаја: до 32
 - Дуплекс: 4 жице или полу дуплекс: 2 жице



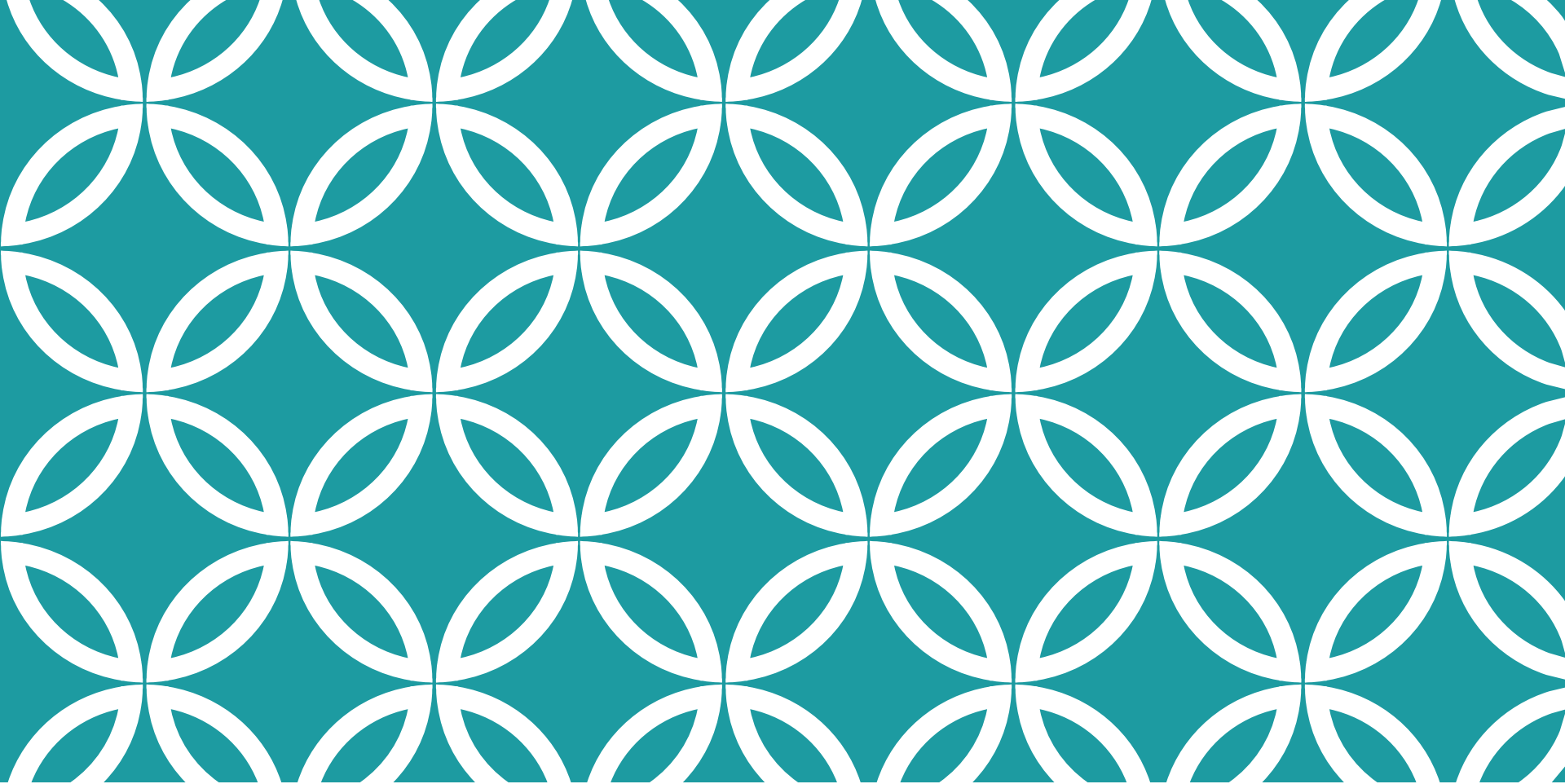
**RS-485 се користи због јаког наслеђа
Сећате ли се 8051?**

LIN versus CAN

	LIN <i>versus</i> CAN	
Access Control	Single Master	Multiple Master
Max Bus Speed	20 Kbps	1 Mbps
Typical # nodes	2 to 16	4 to 20
Message Routing	6-bit Identifier	11/29-bit Identifier
Data byte/frame	2,4,8 bytes	0-8 bytes
Error detection	8-bit checksum	16-bit CRC
Physical Layer	Single-wire	Twisted-pair

КВИЗ

- Када је LIN боља солуција од CAN?
 - To save the bandwidth of another main bus
 - Size of Network is 16 nodes or less
 - When lower speed is acceptable
 - Economical
 - Single Master with multiple slaves



ЗАКЉУЧЦИ ЗА ПОНЕТИ

Шта смо радили

CAN ПРЕГЛЕД

- Један од најраспрострањенијих ауто и индустријских стандарда
- Предности: прилагођеност за асинхрони саобраћај
 - Висока пропусна моћ под малим оптерећењем
 - Ефикасно коришћење пропусног опсега
 - Могуће дефинисати локалне и глобалне приорите
 - Константан премашај: битска арбитрација део поруке
- Мане: немогућност окидања на време
 - Потребна доминантно битско стање: немогућност трансформаторског одвајања
 - Брзина простирања ограничава дужину магистрале
 - Неправедан приступ: високи приоритет може да окупира мрежу
 - Могуће избећи са пажљивим планирањем
 - Могуће додавање чувара магистрале да брине о радном циклусу
 - Повећана кашњења за низак приоритет: изгладњивање
 - Потребно филтрирање да се не би преливали spremнике