

s4

CAN – ett kommunikationsprotokoll för realtidssystem

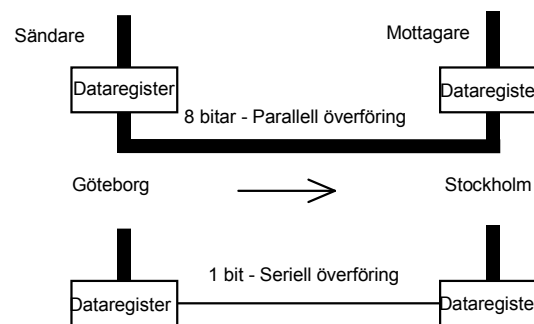
Seriekommunikation – Synkronisering -
Datanät – CAN

Rolf Snedsböl

MOP

1

Seriekommunikation

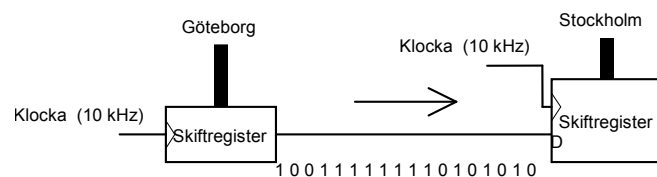


MOP

2

Seriekommunikation -forts

s6

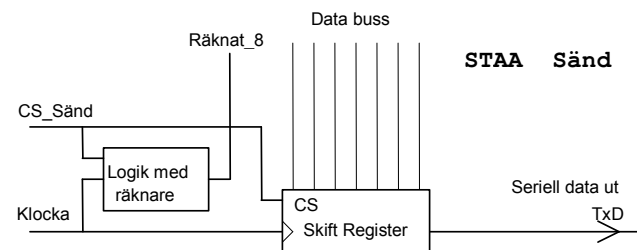


MOP

3

Seriekommunikation -forts

s6



MOP

4

Seriekommunikation -forts

s7

```

Cnt8    equ    %00000001    Def Räknat 8 bitar
        ldx    #Text        Pekare
NextCh  ldaa    1,x+          Läs tecken
        cmpa   #EOT         Sista?
        beq    Slut         ..hoppa om JA

R8är1   ldab    InPort        Läs Status, Redo?
        bitb   #Cnt8        Räknat8=1?
        beq    R8är1        ..hoppa om NEJ

        staa   ShiftReg      Sänd ett tecken

R8är0   ldab    InPort        Läs Status, Upptagen?
        bitb   #Cnt8        Räknat8=0?
        bne    R8är0        ..hoppa om NEJ

        bra    NextCh

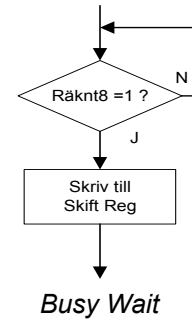
Slut    nop
        bra    Slut
    
```

MOP

5

Seriekommunikation -forts

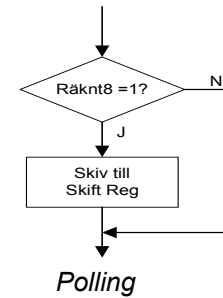
s7



Busy Wait

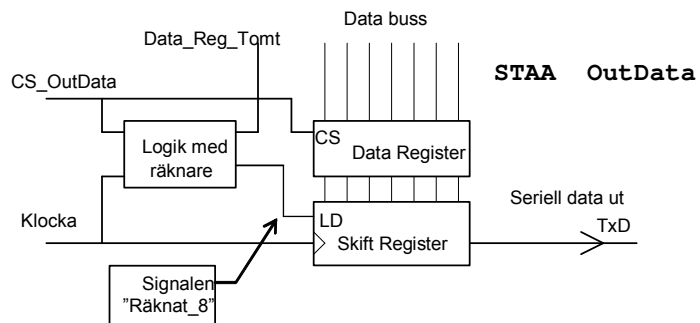
MOP

6



Polling

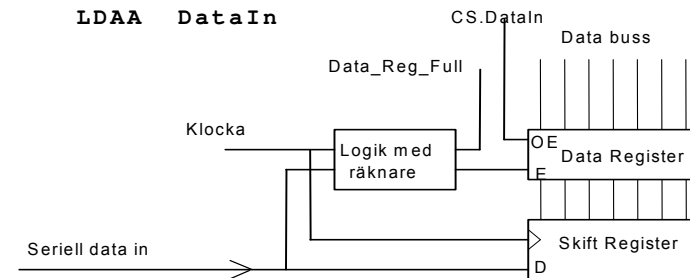
Seriekommunikation - Buffrad sändning



MOP

7

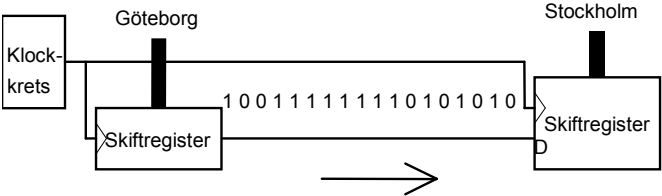
Seriell IO – Buffrad mottagning



MOP

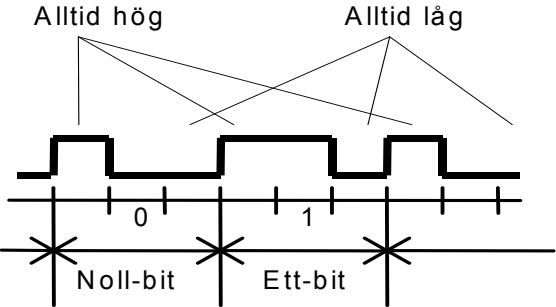
8

Seriekomunikation –Synkron överföring s9



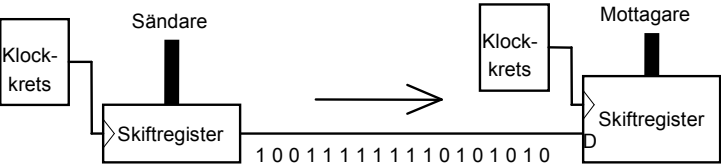
MOP 9

Seriekomunikation –Synkron överföring s9



MOP 10

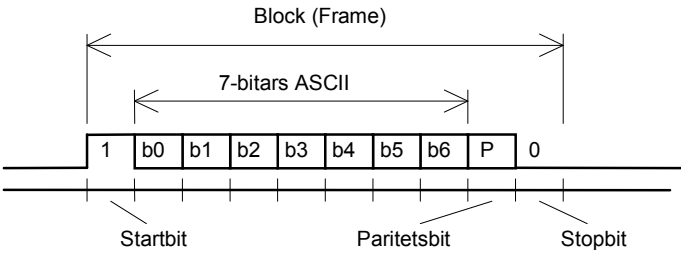
Seriekomunikation –Asynkron överföring s10



BAUD, BAUDRATE

MOP 11

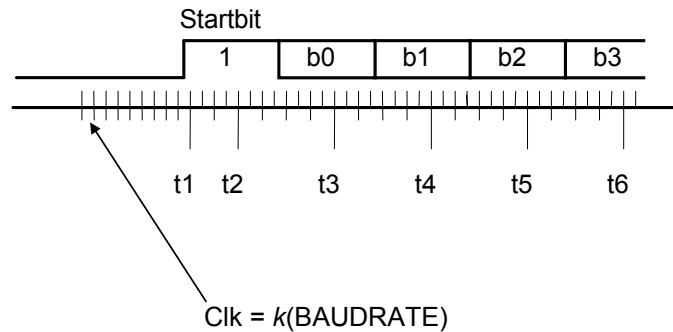
Seriekomunikation –Asynkron överföring



Init: Ant bitar, Paritet, Stopbitar, BAUDRATE, etc

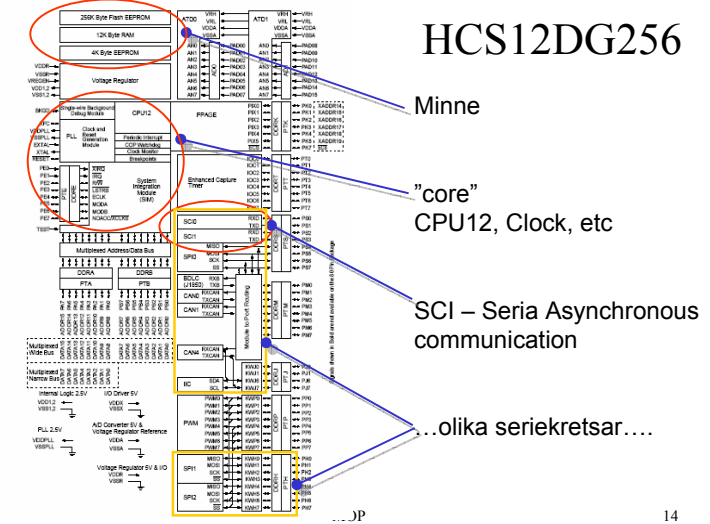
MOP 12

Seriekommunikation –Asynkron överföring



MOP

13



14

SCI – Serial Communication Interface

Two identical devices:
 SCI 0 = Offset 0xC8
 SCI1 = Offset 0xD0

Offset	Use	Access
\$_0	SCI Baud Rate Register High (SCIBDH)	Read/Write
\$_1	SCI Baud Rate Register Low (SCIBDL)	Read/Write
\$_2	SCI Control Register 1 (SCICR1)	Read/Write
\$_3	SCI Control Register 2 (SCICR2)	Read/Write
\$_4	SCI Status Register 1 (SCISR1)	Read
\$_5	SCI Status Register 2 (SCISR2)	Read/Write
\$_6	SCI Data Register High (SCIDRH)	Read/Write
\$_7	SCI Data Register Low (SCIDRL)	Read/Write

BaudRate = BusClock/(16 x BR)

BusClock = 8×10^6 , $1 \leq BR \leq 8192$

MOP

15

SCI – Initialization example, 9600 Baud

```
typedef struct sSCI{
    volatile unsigned short scibd;
    volatile unsigned char scicr1;
    volatile unsigned char scicr2;
    volatile unsigned char scisr1;
    volatile unsigned char scisr2;
    volatile unsigned char scidrh;
    volatile unsigned char scidrl;
}SCI, PSCI*;

#define SCI0_BASE    0x00C8
#define BAUD         8000000/(16*9600)

...
/* init SCI */
PSCI sci = (PSCI) SCI0_BASE;
sci->scibd = BAUD;
sci->scicr1 = 0;
sci->scicr2 = 0xC;
...
```

MOP

16

SCI – Input and output

```

/* send a character through SCI0 */
void _outchar( int c )
{
    /* wait for TDRE==1 */
    while( ( sci->scisr1 & 0x80 )== 0 ) {};
    /* send the character */
    sci->scidr1 = (unsigned char) c;
}

/* receive a character through SCI0 */
int _inchar( void )
{
    /* wait for character, RDRF==1 */
    while( ! ( sci->scisr1 & 0x20 ) ) {};

    /* return the character */
    return (int) ( sci->scidr1 );
}

```

MOP

17

Seriell IO – Datanät

Datanäten grupperas i

- lokala datanät (**LAN, Local Area Network**)
- kontinental (**WAN, Wide Area Network**).

Topologi – Media - Accessmetoder

MOP

18

Seriekomunikation- Datanät- Media

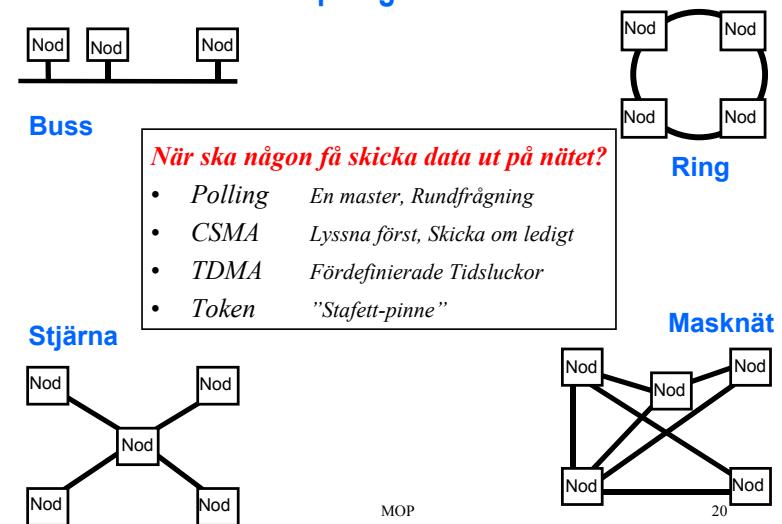
s15

	Tvinnad partråd	Koaxialkabel	Fiberoptik
Kapacitet	Låg	Medium	Mycket hög
Flexibilitet	Mycket hög	Medium	Låg
Kontaktering	Billig	Medium	Mycket dyr
Störökänslighet	Låg	Bra	Mycket hög

MOP

19

Datanät- Topologi - Accessmetoder



MOP

20

CAN Controller Area Network

- Meddelanden, allmänt
- Bussåtkomst
- Synkronisering
- Olika ramar
- Felhantering
- Identifierare

1984, 1991 CAN C-Spec, Bosch och Intel

- Kommunikationsstandard för bilindustrin
- Standard CAN, senare Extended CAN

Implementerat i hårdvara

- INTEL, MOTOROLA, PHILIPS, Siemens, NCS, SGS, etc,etc

Protokoll för distribuerad styrning i händelsestyrda system med små mängder data (1-8 bytes) och ett datanät med fysisk liten utsträckning.

MOP

21

CAN-protokollets egenskaper

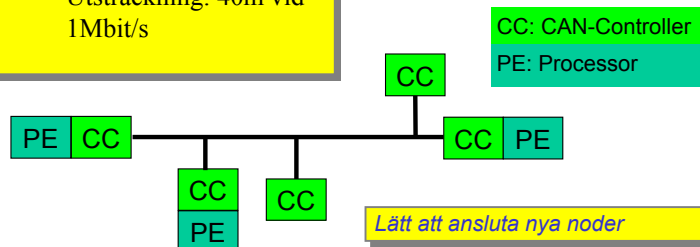
- Flexibelt
- Multimaster protokoll
- Multicast protokoll
- Hög överföringshastighet (1 Mbit/s)
- Kort svarstid (200 µs)
- Automatisk omsändning av störda överföringar
- "Atomic Broadcast"
- Stöd för synkroniserad exekvering i olika noder
- Avlastar processorn/användaren med meddelandeöverföringen

MOP

22

Ett typiskt CAN nät

- Busstopologi
- Media: twisted pair, koaxial, fiber
- Utsträckning: 40m vid 1Mbit/s

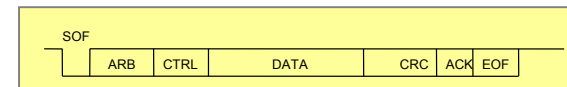


MOP

23

CAN-meddelandet

- Objekt (meddelanden)
 - Ex: broms, ljus, fönsterhiss
 - Skicka/efterfråga ett objekt
- Unik prioritet knuten till identifieraren
 - Specificeras innan "run-time"
 - Prioriteten kan tolkas som en "adress"



- ARB Arbitrering (identifierare)
- CTRL Kontrollinformation
- DATA 0-8 bytes
- CRC Kontrollsumma
- ACK Kvitto (Inbyggd handskakning)
- EOF Slutmarkering

MOP

24

Bussåtkomst

CSMA/CD (Ethernet)

- Carrier Sense: Lyssna Först
- Multiple Access: Vem som helst kan skicka ett meddelande när bussen är ledig.
- Collision Detect: Buss-krock upptäcks.

Algoritm: Börja om vid fel

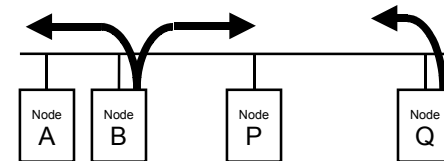
MOP

25

Bussåtkomst forts

Algoritm (CSMA/CD)

- Lyssna på bussen;
- Om bussen är ledig starta sändning av ett meddelande



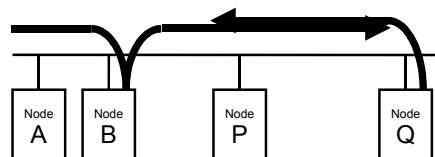
MOP

26

Bussåtkomst forts

Algoritm (CSMA/CD)

- Lyssna på bussen och kontrollera om busskrock uppstår.



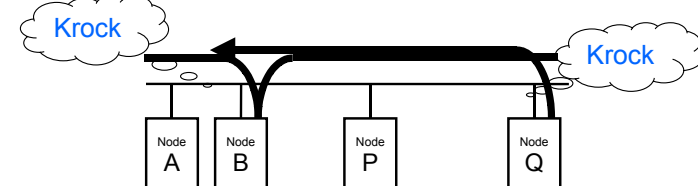
MOP

27

Bussåtkomst forts

Algoritm (CSMA/CD)

- Om busskrock uppstått, avsluta då sändningen och försök igen senare.



MOP

28

Bussåtkomst forts

CSMA/CR (CAN)

- Carrier Sense: Lyssna först
 - Multiple Access: Vem som helst kan skicka ett meddelande när bussen är ledig.
 - Collision Resolution: Buss-krocksupplösning.
- Algoritm: Starkast vinner
 - Multimaster

MOP

29

Bussåtkomst forts

Algoritm (CAN)

- Lyssna på bussen. Om bussen är ledig starta sändning av ett meddelande.
- Lyssna på bussen och jämför bit för bit av sändt data.
- Om mottagen bit skiljer sig från sändt bit indikerar detta att någon med högre prioritet skickar data samtidigt. Avsluta då sändningen och försök igen när "högprioritets-meddelandet" har skickats i sin helhet.

• Outage time

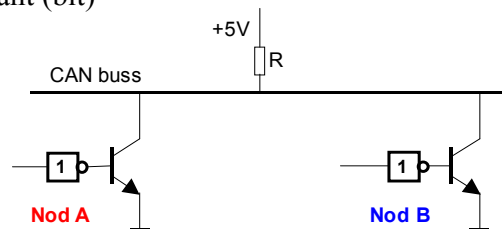
• Kort svarstid (200 μ s vid 1Mbit)

MOP

30

Bussåtkomst forts

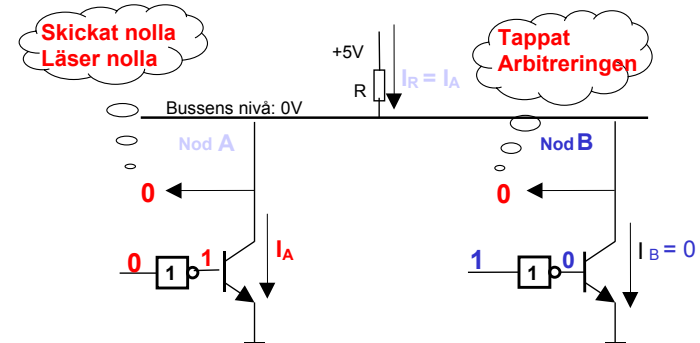
- Bussdrivare: Princip "Open collector"
- Bussens nivå:
 - Recessiv (bit)
 - Dominant (bit)



MOP

31

Bussåtkomst forts



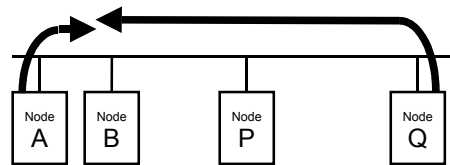
MOP

32

Bussåtkomst forts

Inskränkningar

- Bitsynkronisering
- En enskild bit måste vara "giltig i hela nätet"



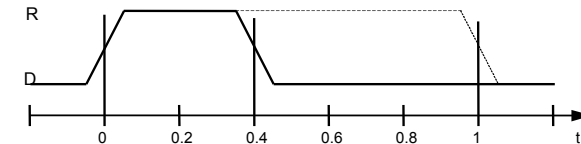
MOP

33

Bussåtkomst forts

Inskränkningar

- Nätets utsträckning/bithastighet



- $\tau_{\text{vänta}} = 2 l/v = 2 \cdot 40 \text{ m} / 2 \cdot 10^8 = 0,4 \mu\text{s}$
- Max 40m vid 1 Mbit/s

MOP

34

Bussåtkomst (arbitrering)

- Tre noder skickar samtidigt

Nod A sänder: \$257 (0010 0101 0111)

Nod B sänder: \$360 (0011 0110 0000)

Nod C sänder: \$25F (0010 0101 1111)

- Bussens nivå

Bit nummer	SOF	1	2	3	4	5	6	7	8	9	10	11	12	13
Bussens nivå	D	D	D	R	D	R	D	R	D	R	R	R	R	R
Nod A skickar	0	0	0	1	0	0	1	0	1	0	1	1	1	1
Nod B skickar	0	0	0	1	1	Slutar sända								
Nod C skickar	0	0	0	1	0	0	1	0	1	1	Slutar sända			

MOP

35

Bussåtkomst (arbitrering)

- Identifierare anger prioriteten
- **OBS!!!** Endast EN nod får skicka data med en speciell identifierare

MOP

36

Synkronisering i CAN

- Asynkront protokoll
- Okodade bitar

• Synkroniseringsmekanism

- Mottagare:
 - Hård synkronisering på SOF
 - Synkronisera på flanker i mottaget data
- Sändaren:
 - Inför STUFF BITAR vid långa sekvenser av nollor / ettor

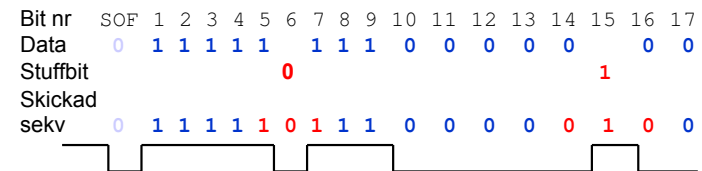
MOP

37

Synkronisering i CAN forts

• Stuffbitar

- Skickar ett meddelande som börjar med \$FF03 binärt
1111 1111 0000 0011.

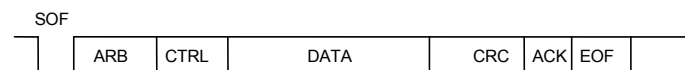


MOP

38

Olika ramar

- Data Frame (skicka data)
 - SOF Start Of Frame. 1 bit
 - ARB Arbitration. 12 bitar(Stand CAN)
 - CTRL Control. 6 bitar
 - DATA Användar data. 1-8 bitar (max 64 b)
 - CRC Kontrollsumma 16 bitar
 - ACK Kvitto 2 bitar
 - EOF End Of Frame 7 bitar
 - IFS Inter Frame Space. 3 bitar
- Totalt 108 b



MOP

39

Olika ramar forts

• Remote Frame (efterfråga en dataframe)

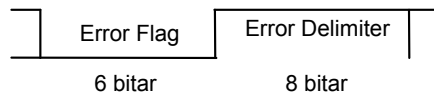
- En remoteframe saknar datafält (se data-frame).
- Remote- och dataframe har samma identifierare!???
- Ex efterfråga hastighet; skicka hastighet
- ID är 11 bitar, ARB-fältet är 12 bitar
- RTR-bit (bit 12 i ARB-fältet)
 - Remote frame recessiv RTR-bit
 - Data frame dominant RTR-bit
- Innebär att vem som helst kan skicka en remoteframe (men endast EN kan skicka en data-frame)

MOP

40

Olika ramar forts

- Error frame (skriv sönder pågående utskick)
 - 6 st dominanta bitar (Error flag)
 - 7 recessiva bitar (Error delimiter)



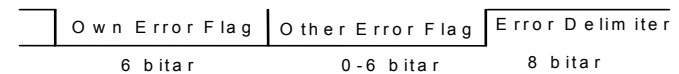
- Alla kan skicka en Error frame
- Automatisk omsändning av störda överföringar
- Robust

MOP

41

Feldetektering

- Skicka Error Frame vid felupptäckt
 - Bit-fel
 - Stuff-fel
 - CRC-fel
 - Fixed Form-fel



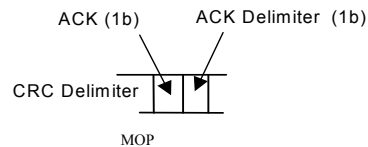
- Möjliggör samtidig exekvering i noder
- Atomic broadcast

MOP

42

Kvitto (ACK)

- En frame innehåller ett två-bitars ACK-fält för handskakning
- Sändaren av en frame skickar två recessiva bitar i detta fält
- En adresserad fungerande mottagare skriver över den första med en dominant bit
- Sändaren detekterar dominant eller recessiv bit i ACK-fältet



MOP

43

Kvitto forts

- Vad innebär ACK-funktionen?
 - Sändaren “ser” att hans frame tas emot
 - Sändaren detekterar att den inte är “ensam i nätet”
 - Observera att sändaren inte kan detektera *vem* som korrekt tagit emot utskickad meddelande utan endast att *någon* korrekt tagit emot utskickad meddelande.

- Underlättar konsistens

MOP

44

Identifierare

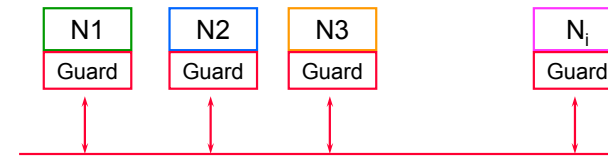
- Identifieraren “identifierar” meddelandet
 - ID x kan endast skickas av en viss nod pga arbitrerings
- Standard CAN
 - 11 bitar- 2048 olika identifierare
- Enskilda meddelanden
 - \$04B Bromskommando
 - \$1A2 Ljus meddelande, fram
- Extended CAN (29 bitar - 0.5 10⁹ identifierare)
 - Försök till standardisering (anpassning J1939)
 - \$1A29 92D0 HF parklampa
 - \$1A29 22D1 HF halvljuslampa
 - \$1A29 32D2 HF helljuslampa

MOP

45

Identifierare forts

- Filtrering
 - En CAN-krets initieras (i mask-registret) med ett antal identifierare som den skall igenkänna



- Multicast
- Belastar ej processorn med önskad data

MOP

46

Identifierare forts

- Filtrering och prioriteter
 - \$1A29 92D0 HF parklampa
 - \$1A29 22D1 HF halvljuslampa
 - \$1A25 32D2 HF helljuslampa
 - \$FFF0 0FF0 Maskregister
- Möjliggör dynamisk prioritetsändring
- Outage times

MOP

47

Sammanfattning CAN

- Flexibelt
- Multimaster protokoll
- Multicast protokoll
- Hög överföringshastighet (1 Mbit/s)
- Kort svarstid (200 µs)
- Automatisk omsändning av störda överföringar
- ”Atomic Broadcast”
- Stöd för synkroniserad exekvering i olika noder
- Avlastar processorn/användaren med meddelandeöverföringen

MOP

48