Controller Area Network

CAN

part 1

Advanced CAN Solutions

We will talk about:

- General features
- CAN messages
- Bitwise arbitration
- CAN synchronisation

Controller Area Network

CAN

General features

KVASER

Advanced CAN Solutions

CAN features

- Bus-access by message priority
  - CSMA/CR
    Carrier Sense Multiple Access / Collision Resolution
- Bus access conflicts resolved by arbitration
  - Bit-wise
  - Non-destructive
  - Allows for guaranteed latency time
- Message identifier
  - CAN has no node addresses
    Every node receive every message and decides itself whether to use it or not

KVASER
Advanced CAN Solutions

CAN features

- Extensive ERROR checking
  - Five different checks
  - Every connected node participate
- Data consistency secured
  - A message is accepted by all nodes or none
- Different Bus Management Methods can be applied for CAN systems, e.g.,
  - Bit-wise arbitration
  - Master/Slave
  - Daisy Chain
  - TDMA

CAN features

- A Higher Layer Protocol is always required
  - CAN is only a low level specification
- The capability of CAN is restricted by the Higher Layer Protocol chosen
  - Market segment
  - Real-time requirements
  - Product Administration requirements
  - etc

KVASER

Advanced CAN Solutions

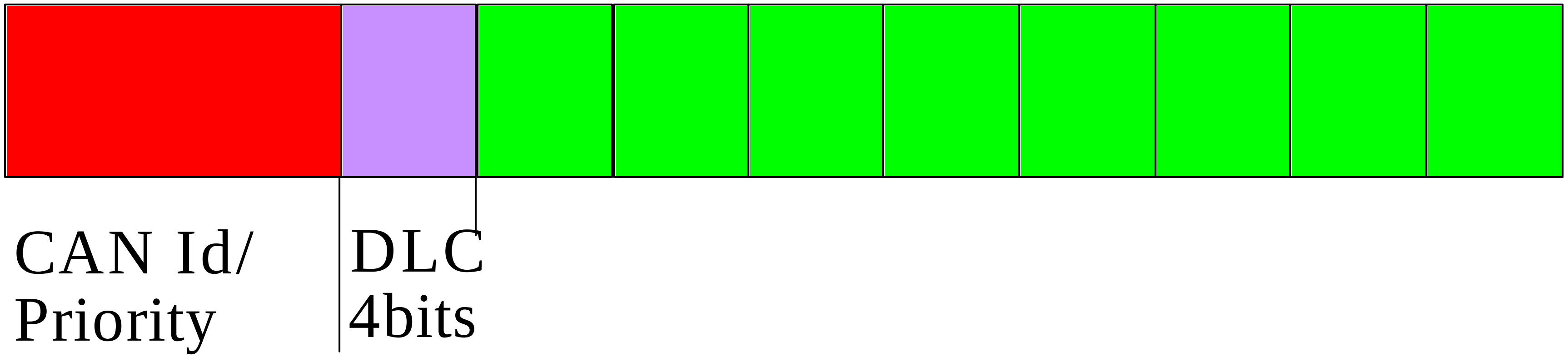
Controller Area Network

CAN messages

KVASER

Advanced CAN Solutions

CAN message



11 or 29 bits

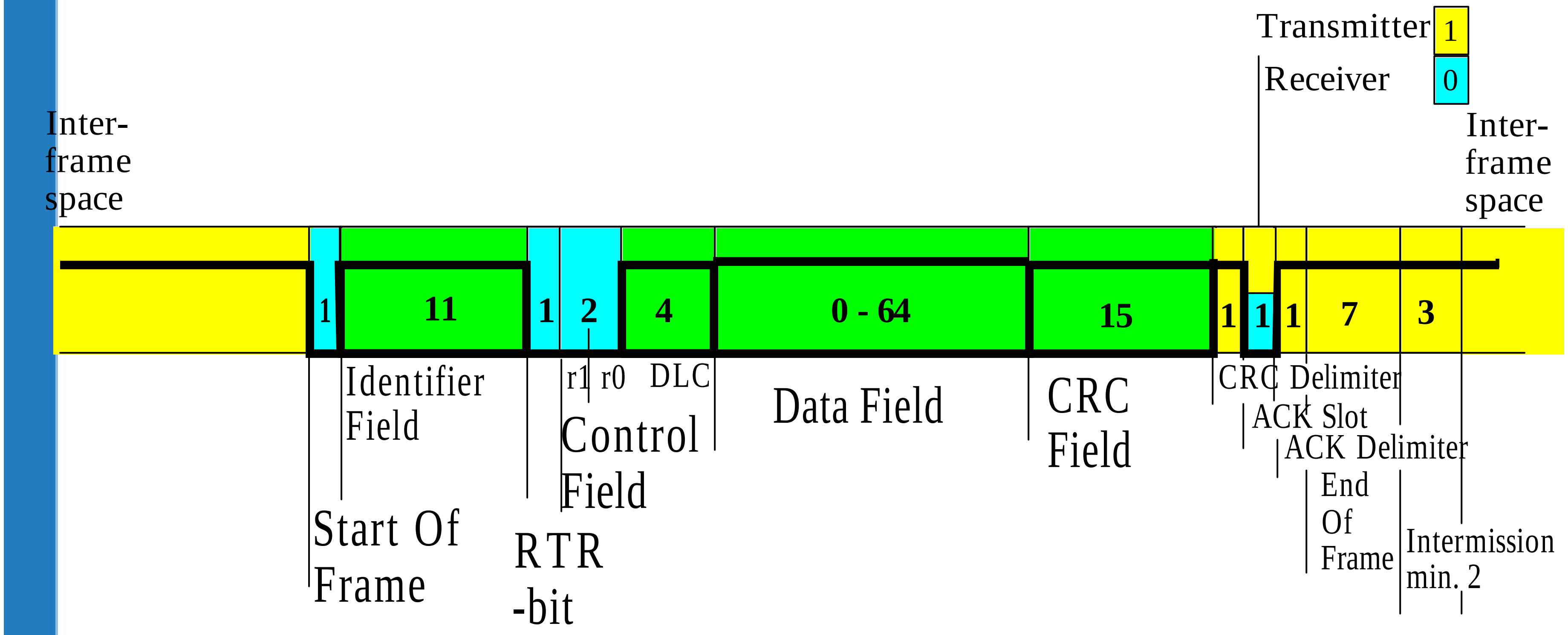
0 - 8 bytes

CAN Id/
Priority

DLC
4bits

Data Frame

CAN Data Frame Std



Transmitter 🟨 1

Receiver 🟦 0

Inter-
frame
space

1 11 1 2 4 0 - 64 15 1 1 1 7 3

Start Of Frame

Identifier Field

RTR -bit

r1 r0 DLC

Control Field

Data Field

CRC Field

CRC Delimiter

ACK Slot

ACK Delimiter

End Of Frame

Intermission min. 2

Inter-
frame
space

Bit values

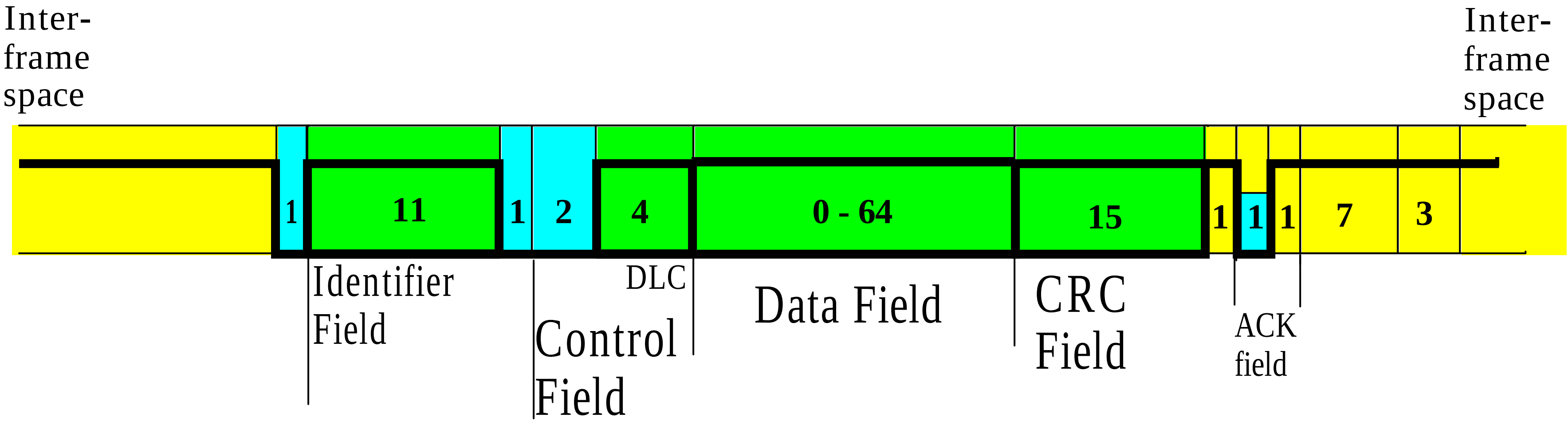
🟦 0

🟩 0/1

🟨 1

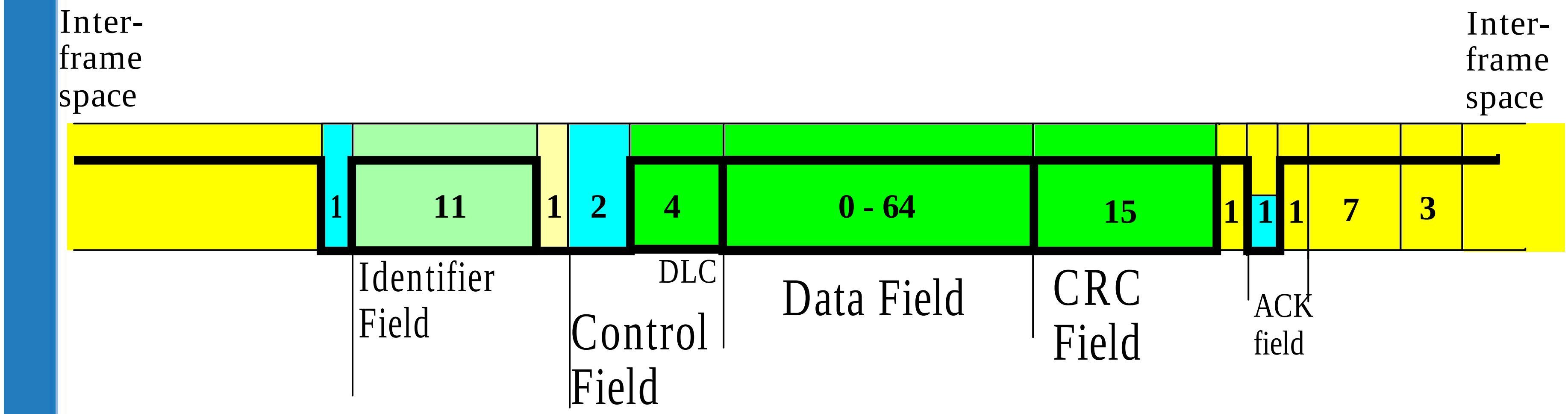
KVASER
Advanced CAN Solutions

A CAN frame



- Priority and Identification field.
- Control field.
- Data field.
- CRC field.
- Acknowledgement field.
- Fixed part.

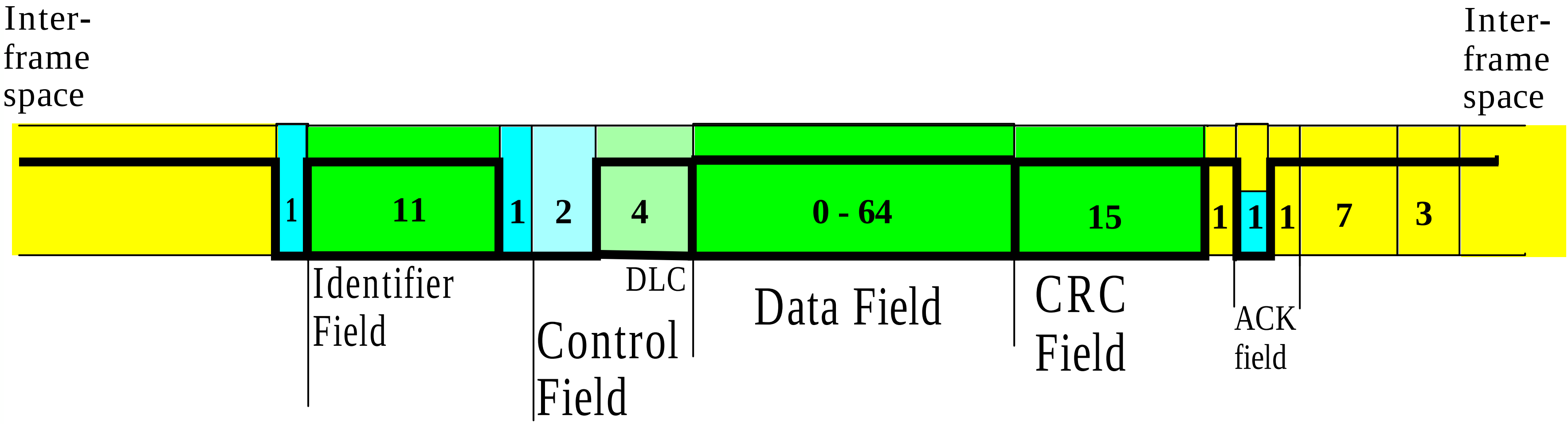
Priority and/or Identification



Identifier Field

- 11 bit or 29 bit. 11 bit is shown above
- Arbitration is done in this part
- This part sets the priority of the message in case of collision
- The Remote Transmit Request bit (RTR) is a part of this field
- CAN-controllers support this part of the message as an identification by which filtration can be made

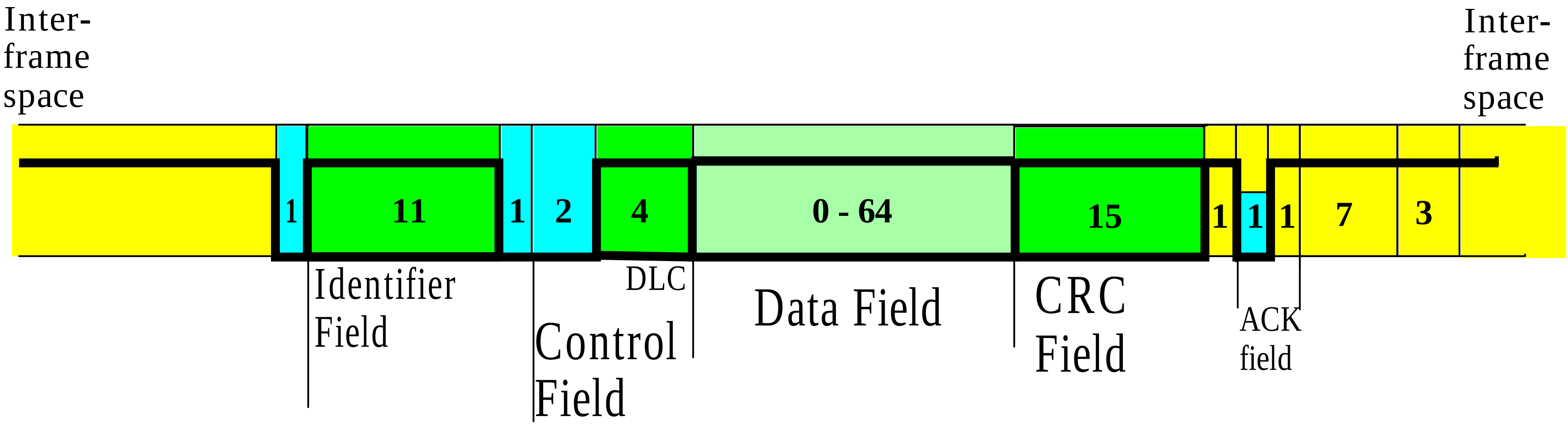
Control and Data length field



Control Field

- Main function Data Length Code DLC
- DLC can have the value 0..8 (Values above 8 are interpreted as 8)
- Two bits are reserved and used to indicate Extended frames
- In Standard frames the reserved bits are fix dominant bits

Data field

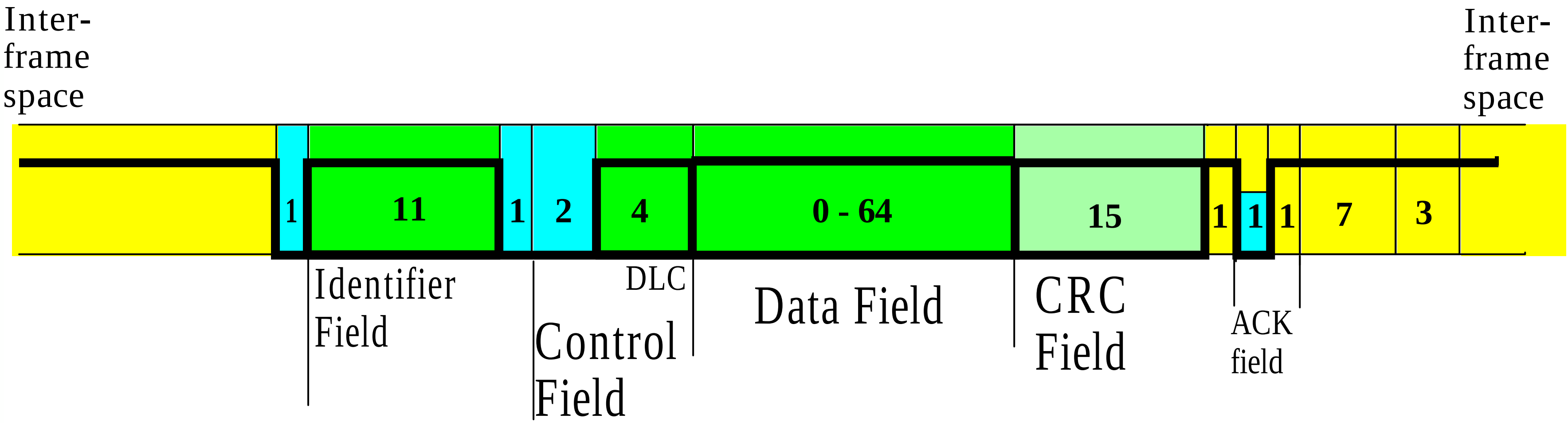


Data Field

- The field can be from zero up to eight byte
- It is always full 8 bit bytes
- The bytes can have any value
- Some CAN controllers can extend ID filtration into the data field

KVASER
Advanced CAN Solutions

CRC field

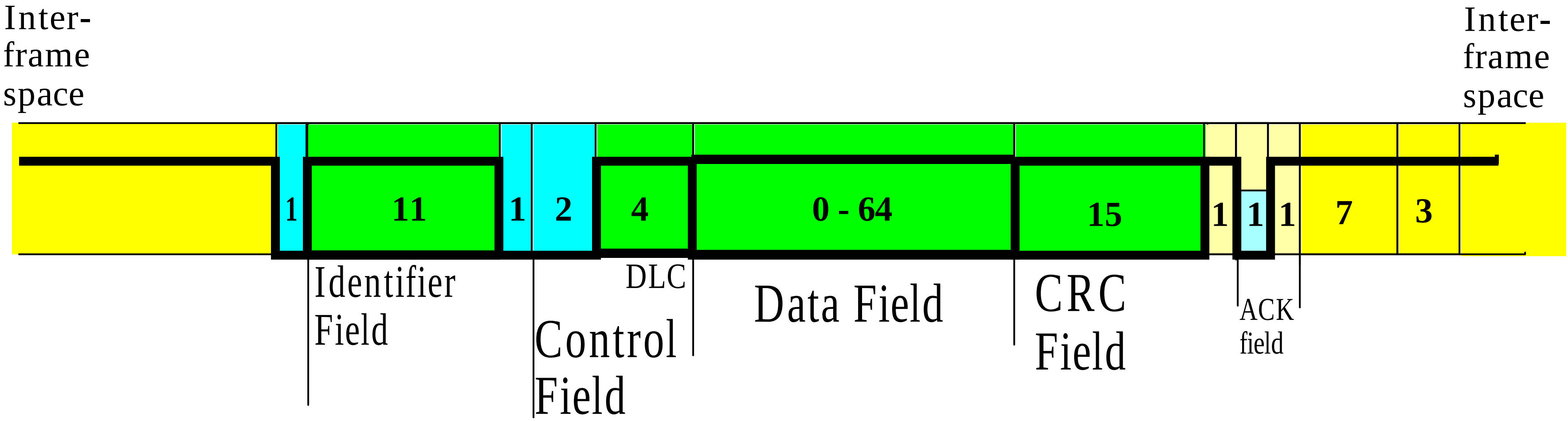


CRC Field

- A Checksum of the bits in the message
- The CRC is optimised for this short type of message
- The CRC check is only one of the error checks in the CAN communication

KVASER
Advanced CAN Solutions

Acknowledgement field

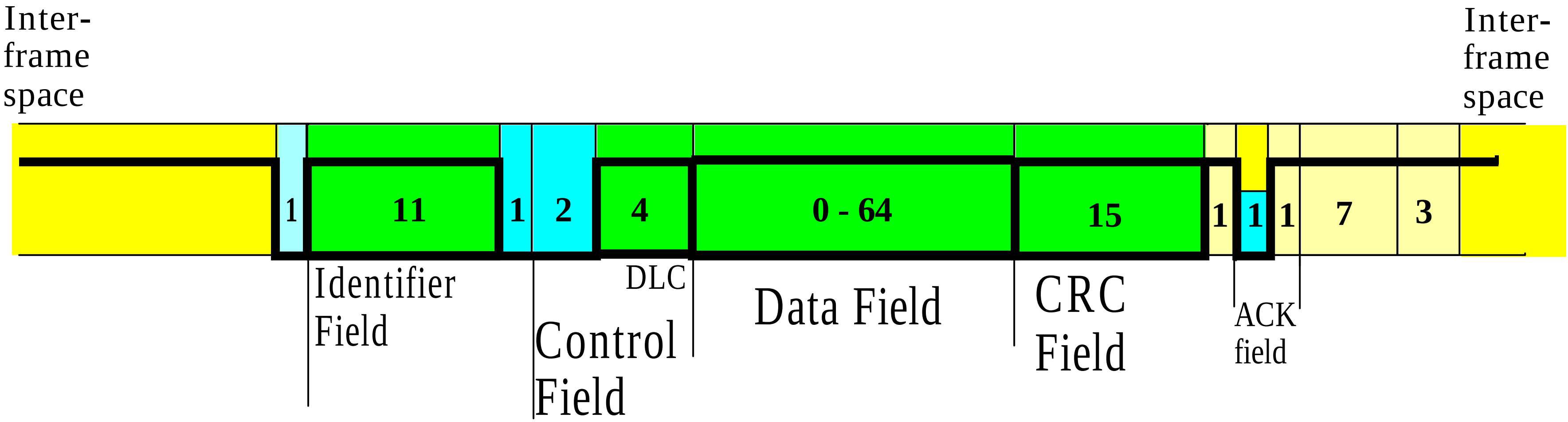


Acknowledgement Field

- It is an acknowledgement of reception, securing at least one receiver has got the message OK
- The transmitter sets the ACK bit to 1
- Any receiver set the ACK bit to 0 when the message is found OK

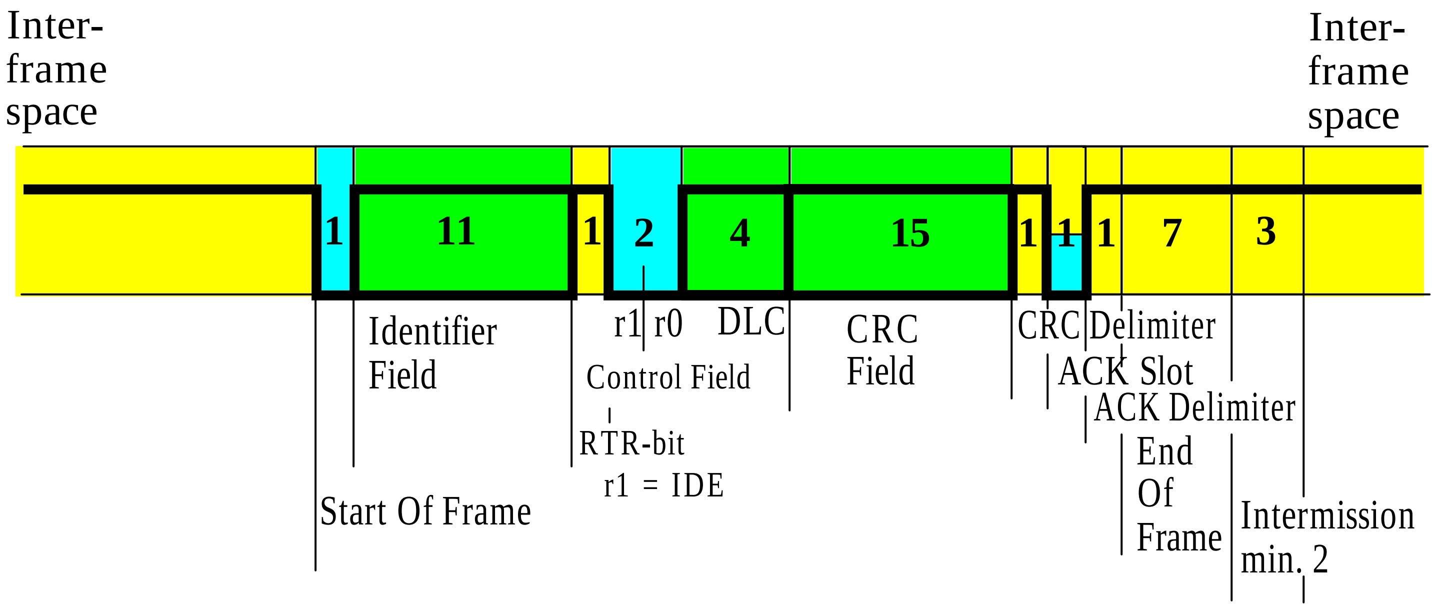
KVASER
Advanced CAN Solutions

Fixed value bits



- These bits have a fixed value for all message frames
- There is additional rules for the intermission bits

CAN Remote Frame Std

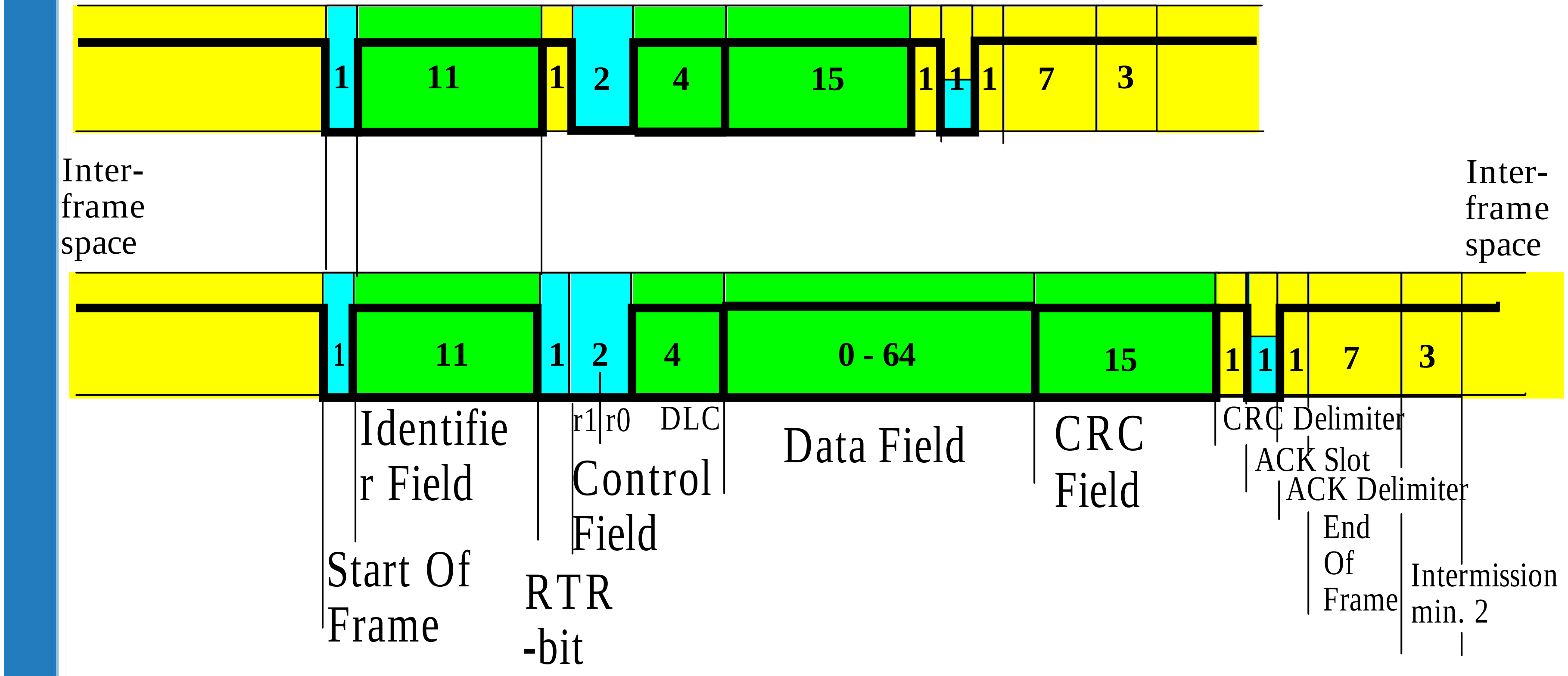


Bit values

- 🟦 0
- 🟩 0/1
- 🟨 1

KVASER
Advanced CAN Solutions

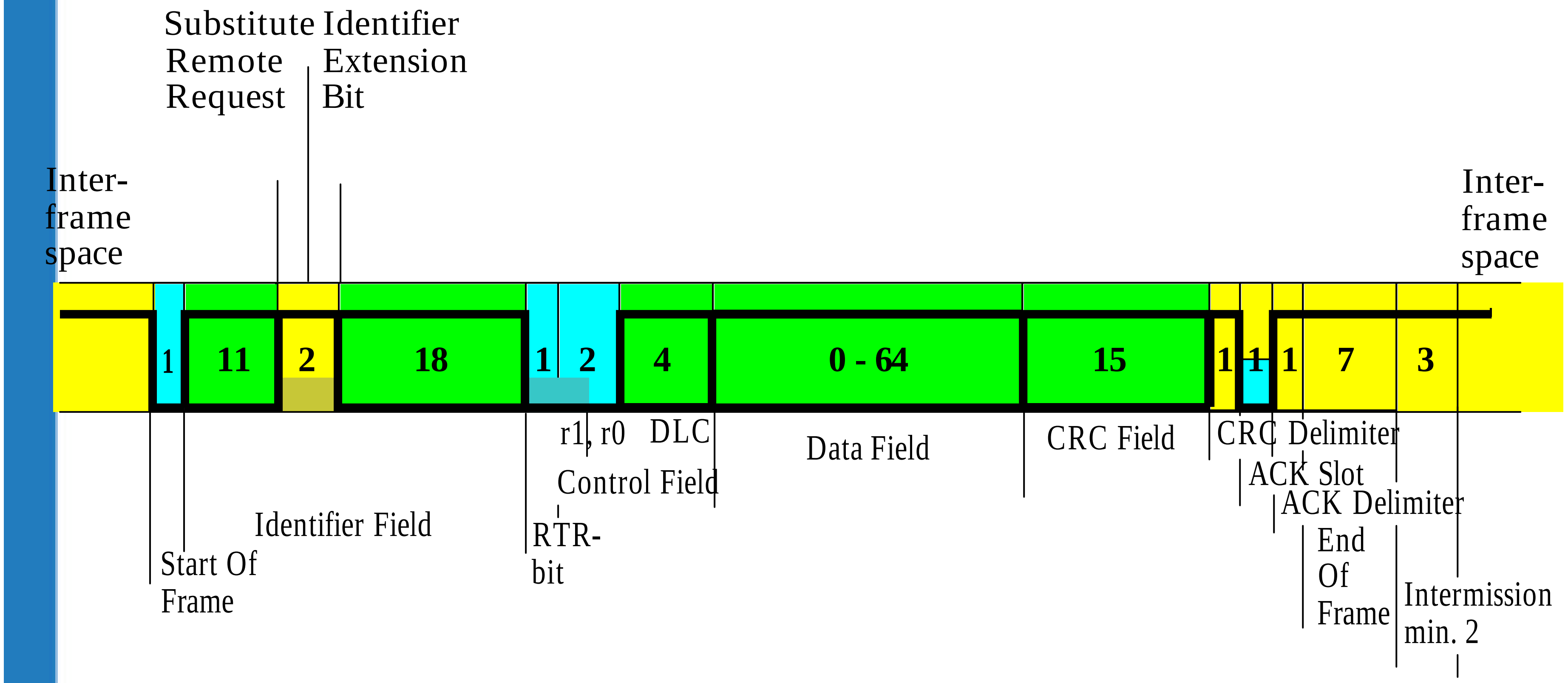
Remote compared to data frame, Std



NOTE: DLC in all remote requests must be identical to DLC in corresponding DATA Message!

KVASER
Advanced CAN Solutions

CAN Data Frame Ext.



Bit values

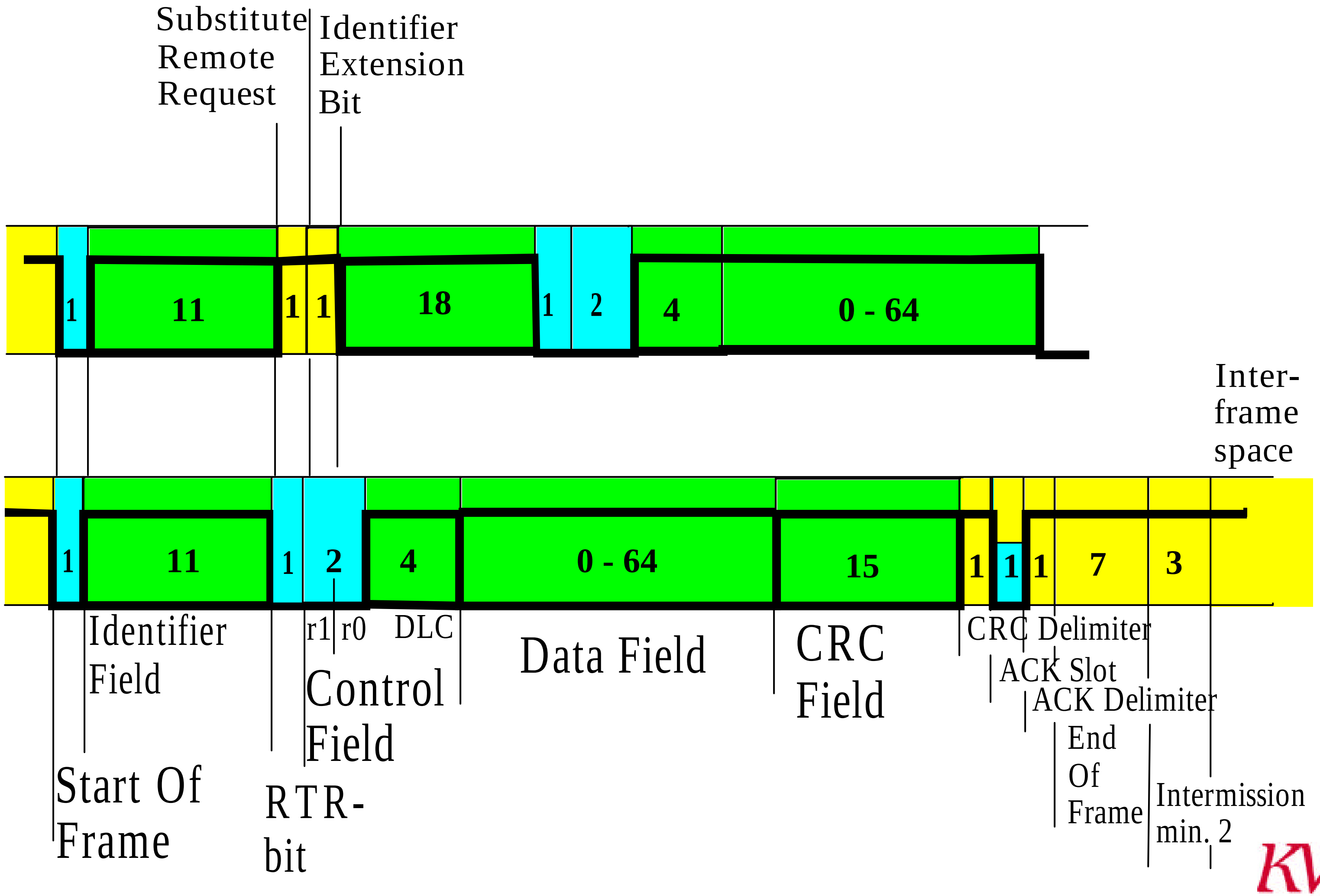
🟦 0

🟩 0/1

🟨 1

KVASER
Advanced CAN Solutions

CAN Data Frame Ext. and Std



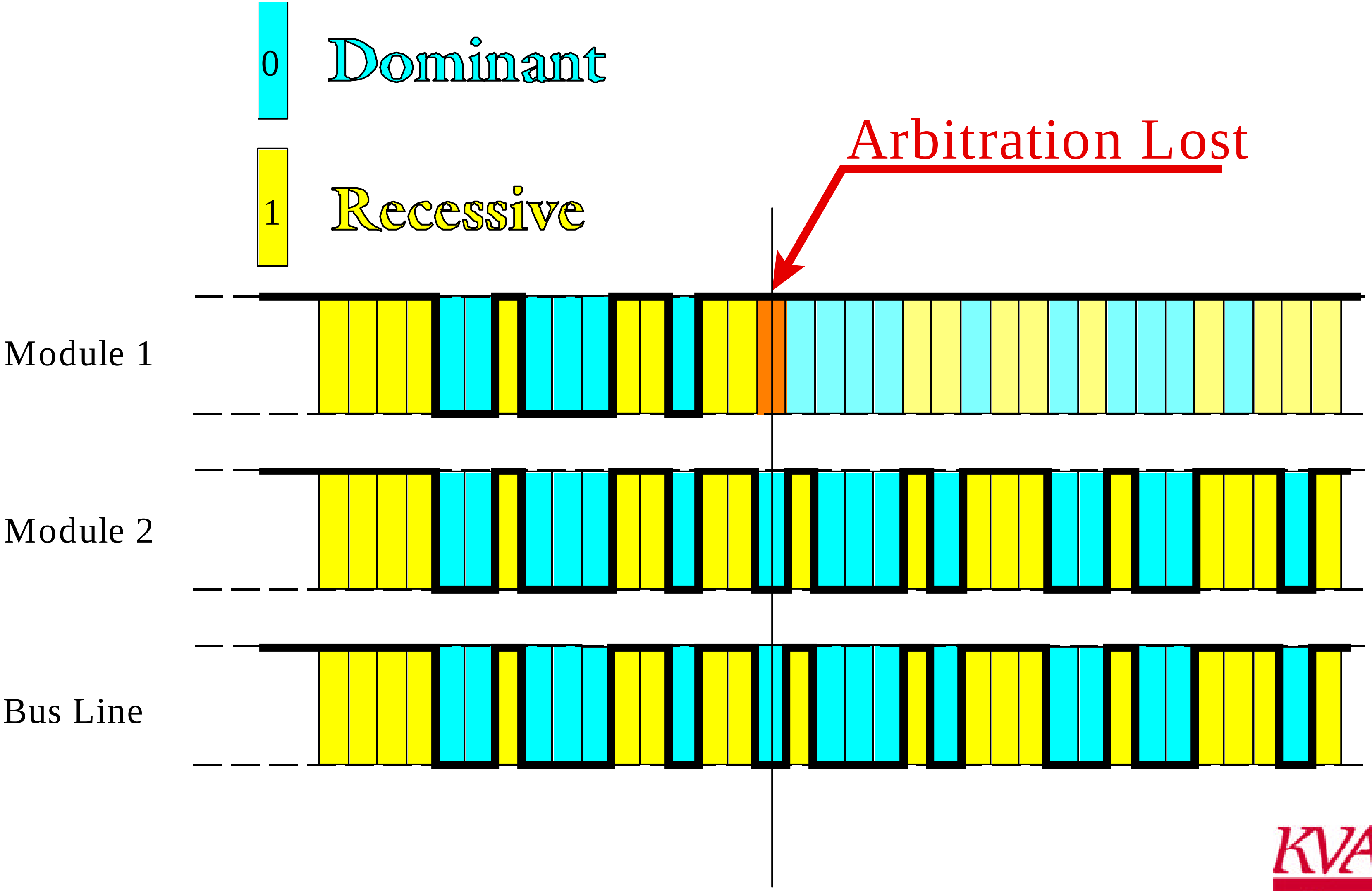
Advanced CAN Solutions

Controller Area Network

CAN

Arbitration

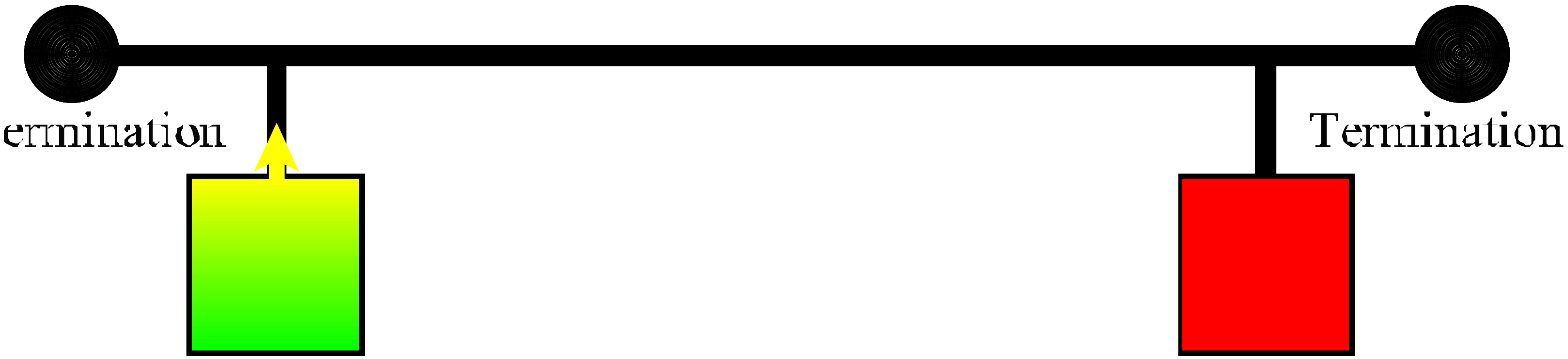
Bit-wise Arbitration



Advanced CAN Solutions

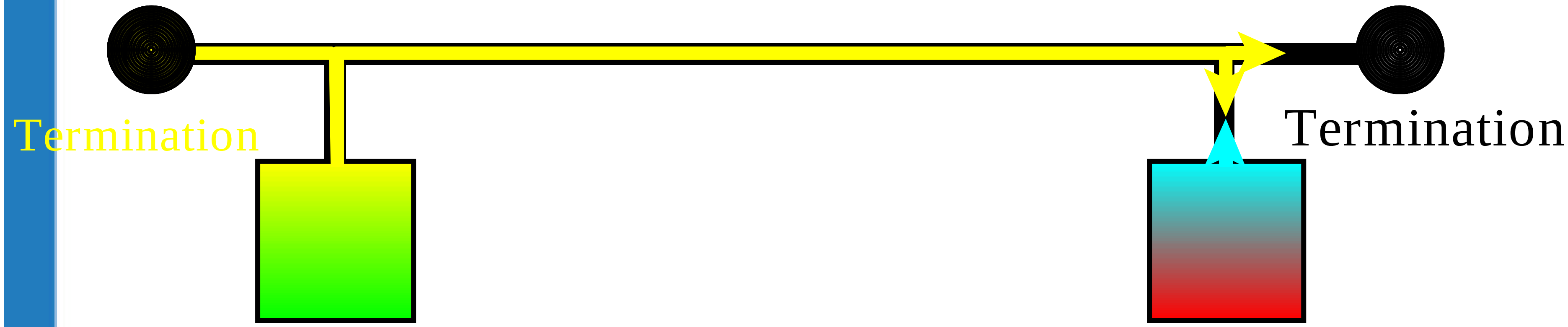
Bit-wise arbitration

The green node starts transmisson of a recessive bit on the idle bus.



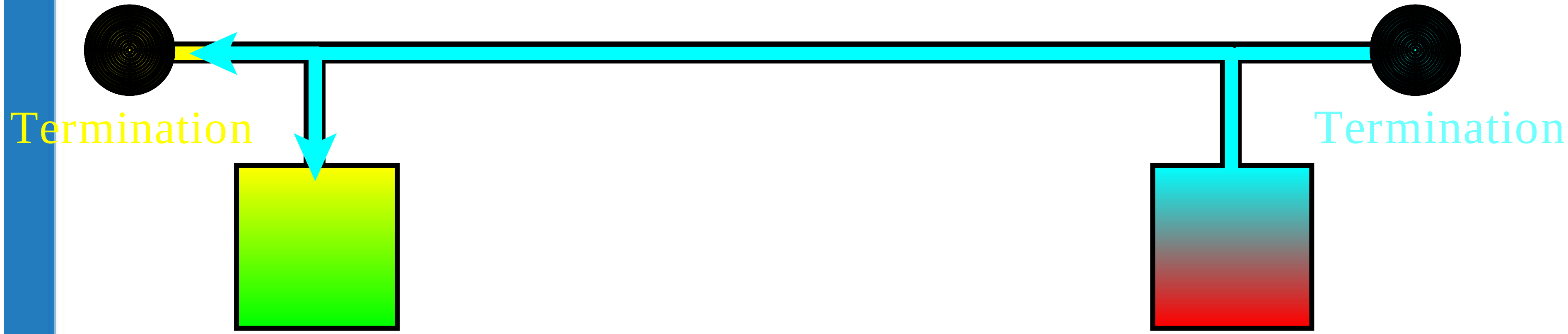
Bit-wise arbitration

The wave from the green node has not yet reached the red node. To this the bus is still free and the red node starts transmitting a dominant bit.



Bit-wise arbitration

Now the green node can see that there is a dominant bit on the bus and that it has lost arbitration.
Thus a transmitter has to wait until the wave has reached the most distant node and **back** (plus internal delays) before judging the bus-line level



KVASER
Advanced CAN Solutions

Bit-wise arbitration

Maximum bit rate is depending on wave propagation delays

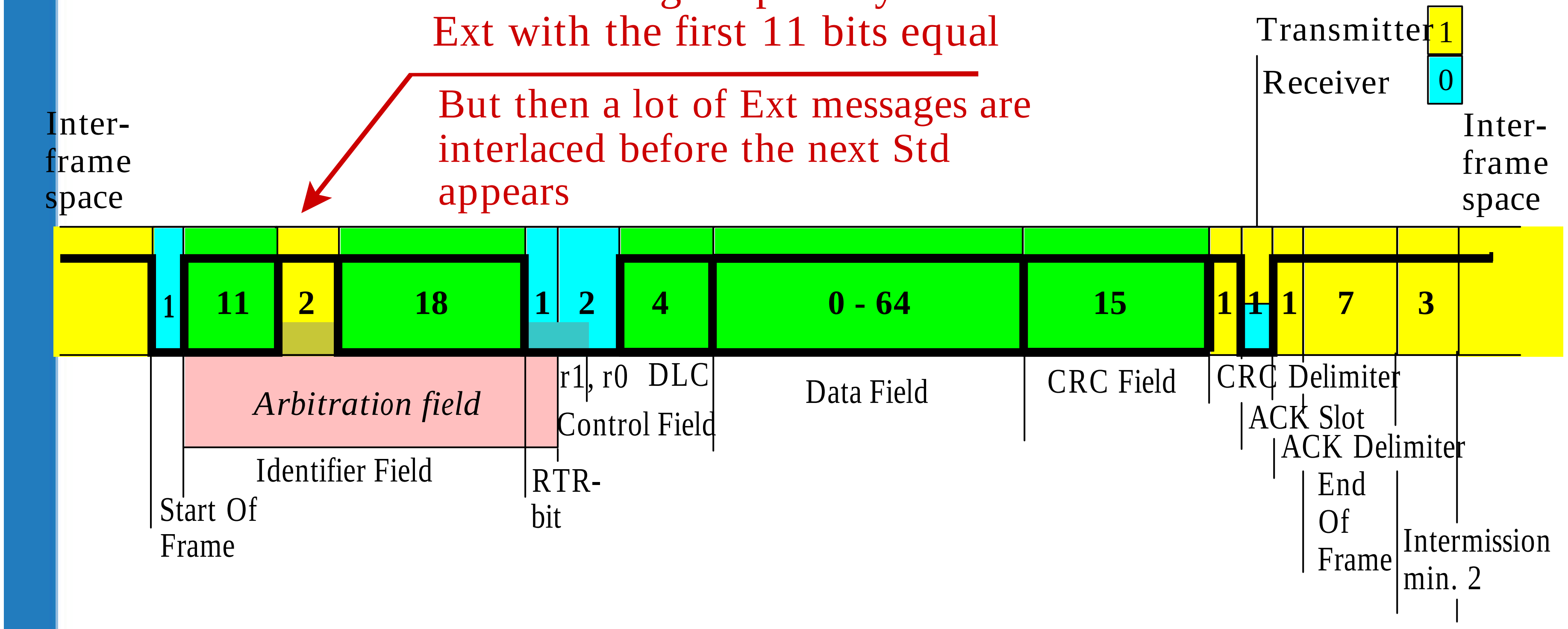
- Bus length
- Opto couplers
- Internal delays
- Oscillator accuracy

(Often not specified)

CAN Data Frame Ext.

A Std has higher priority than an Ext with the first 11 bits equal

But then a lot of Ext messages are interlaced before the next Std appears



Bit values

🟦 0

🟩 0/1

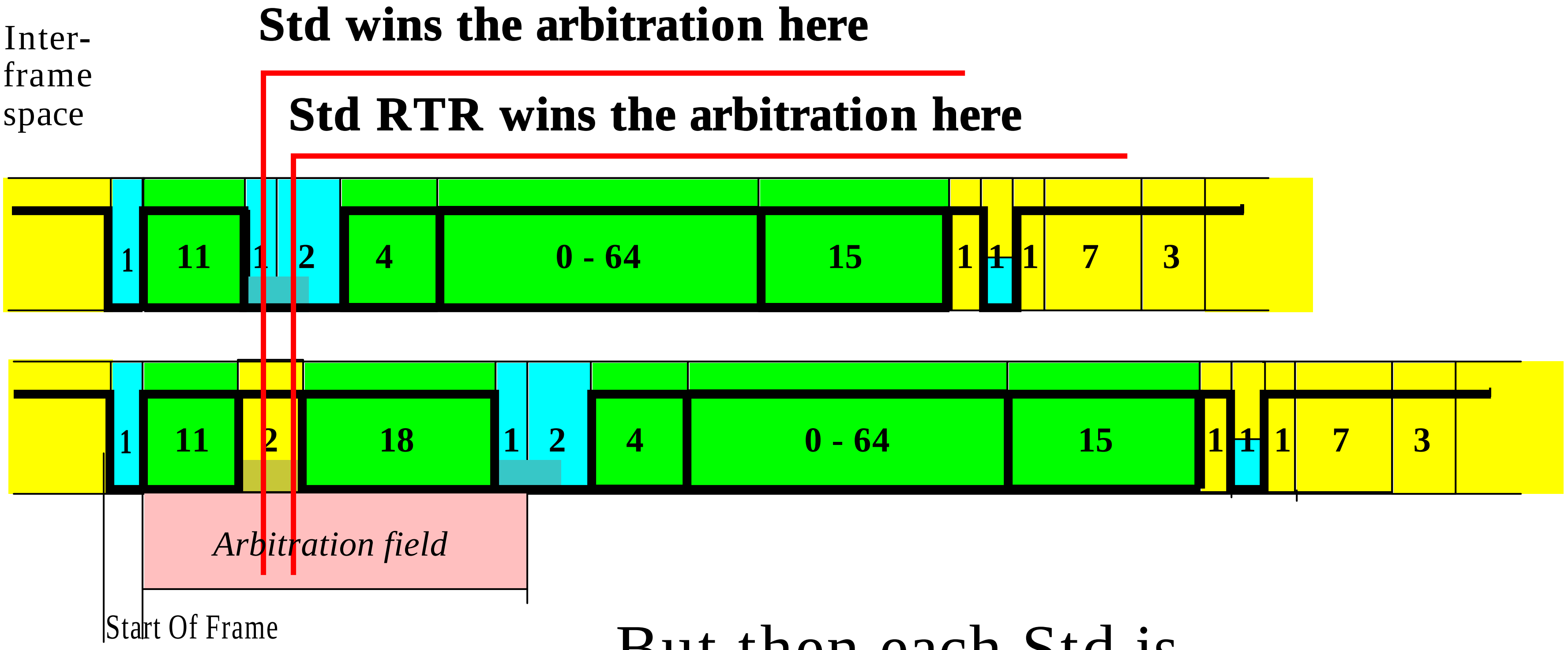
🟨 1

KVASER

Advanced CAN Solutions

Std Identifier wins over Ext. Identifier

when the first 11 bits are equal



Std wins the arbitration here

Std RTR wins the arbitration here

Inter-frame space

1 11 1 2 4 0 - 64 15 1 1 1 7 3

1 11 2 18 1 2 4 0 - 64 15 1 1 1 7 3

Arbitration field

Start Of Frame

Bit values

🟦 0

🟩 0/1

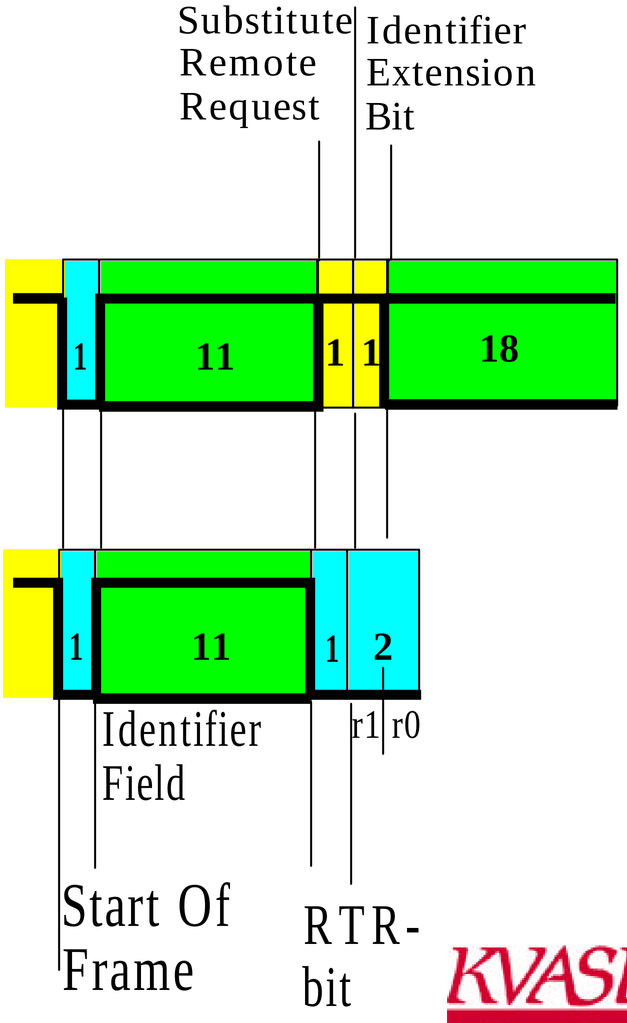
🟨 1

But then each Std is interlaced by 2^{18} Ext messages

KVASER
Advanced CAN Solutions

Std vs. Ext. priority

Prio	Bit -29 - -19	Bit -18 - -1
Hi	0	std
	0	0
	0	1
	0	…
	0	262143
	1	std
	1	0
	1	1
	…	…
Low	2047	262143



KVASER
Advanced CAN Solutions

Controller Area Network

CAN

Synchronisation

KVASER

Advanced CAN Solutions

BIT TIMING

A bit time is built up by four parts:
Synch_Seg, Prop_Seg, Phase_Seg1 and Phase_Seg2

Those parts are built up by a number of time quantas

Bit Time = Synch_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2

Often alternatively expressed as

TBIT = TSYNC + TSEG1 + TSEG2

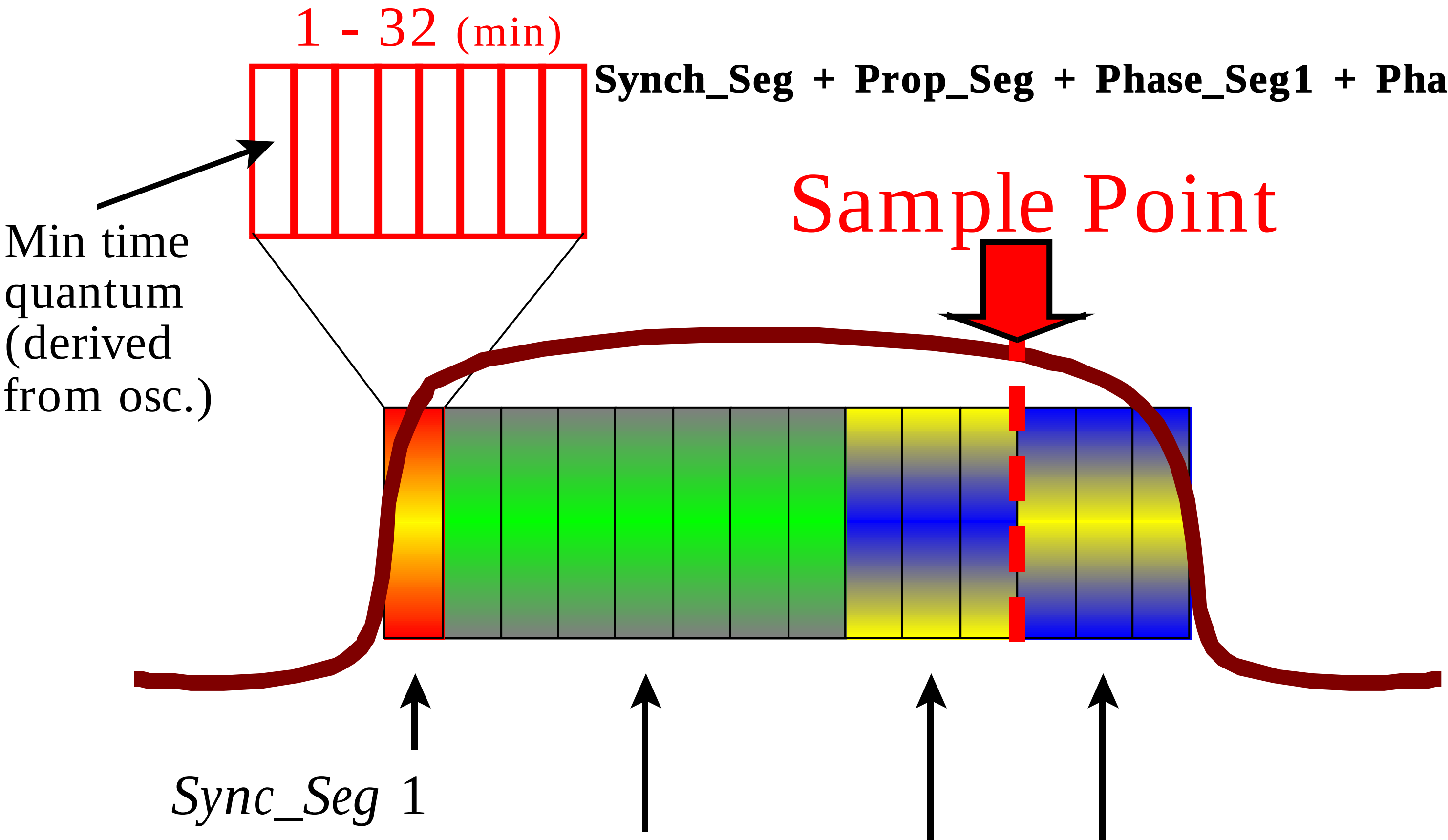
where

- TSYNC = 1
- TSEG1 = [2..16]
- TSEG2 = [1..8]
- TBIT = [4..25]

KVASER
Advanced CAN Solutions

Bit Time =

Synch_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2



Sync_Seg 1

Prop_Seg 1 - 8 (min)

Phase_Seg 1 1 - 8 (min)

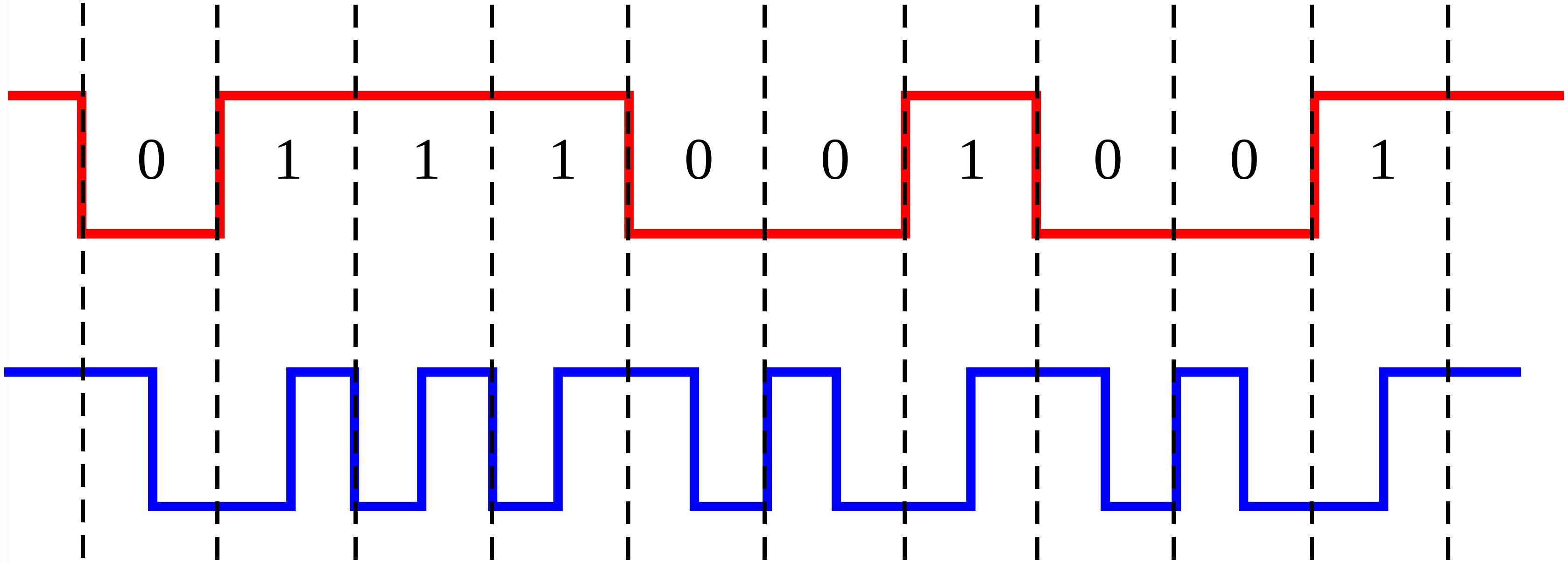
Phase_Seg 2 1 - 8 (min)

Total 4 - 25 Time quanta (min)

KVASER Advanced CAN Solutions

NRZ

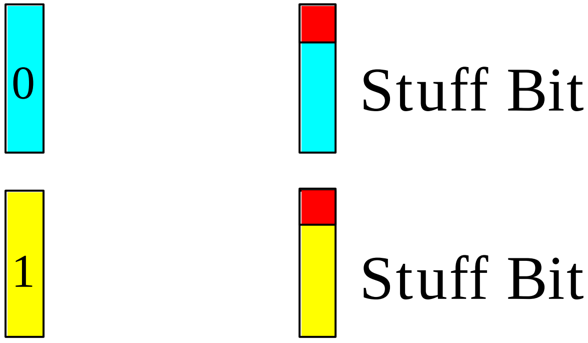
Non Return to Zero



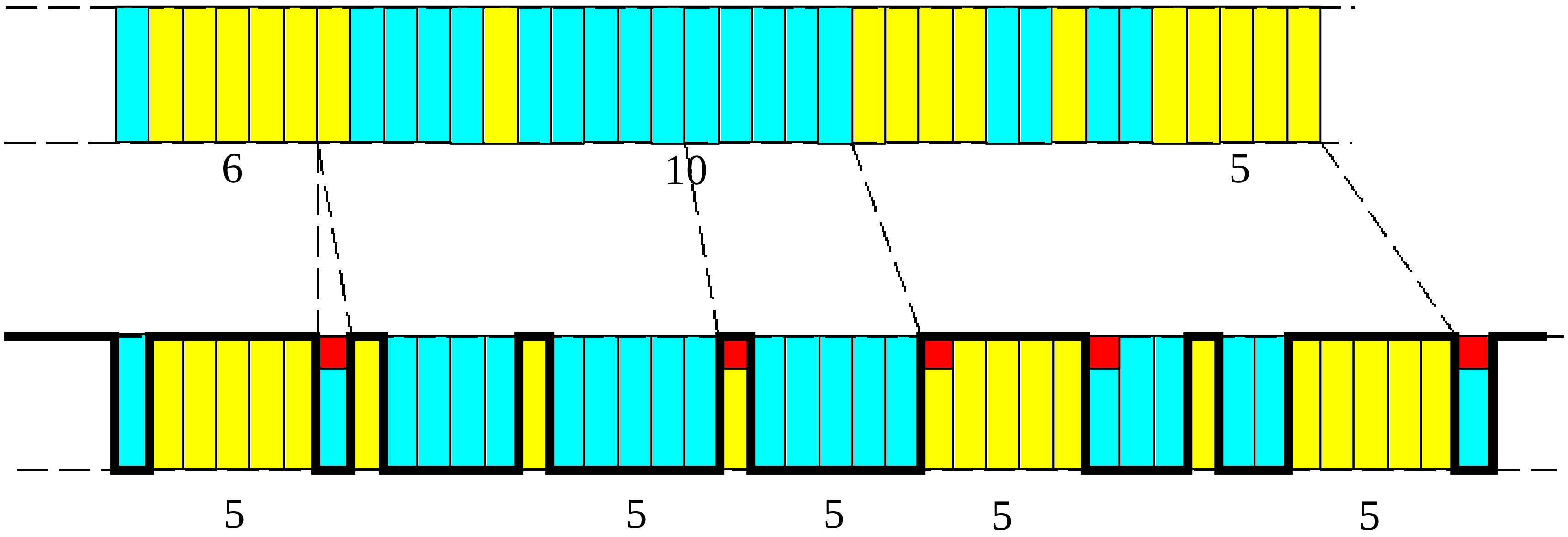
(Manchester Coding)

CAN is NRZ which has EMC advantages compared with MC

Bit Stuffing



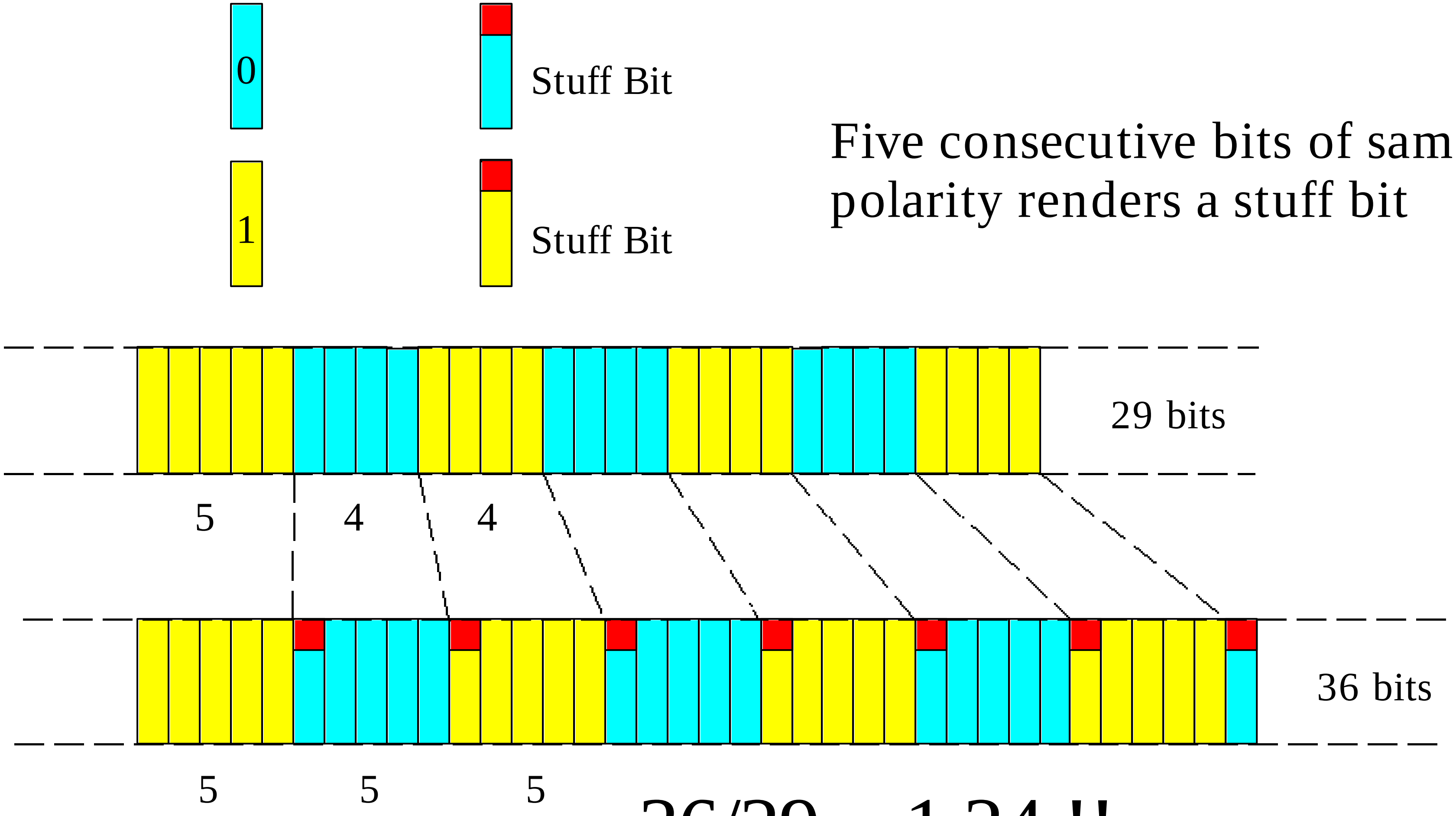
Five consecutive bits of sam
polarity render a stuff bit



KVASER

Advanced CAN Solutions

Bit Stuffing

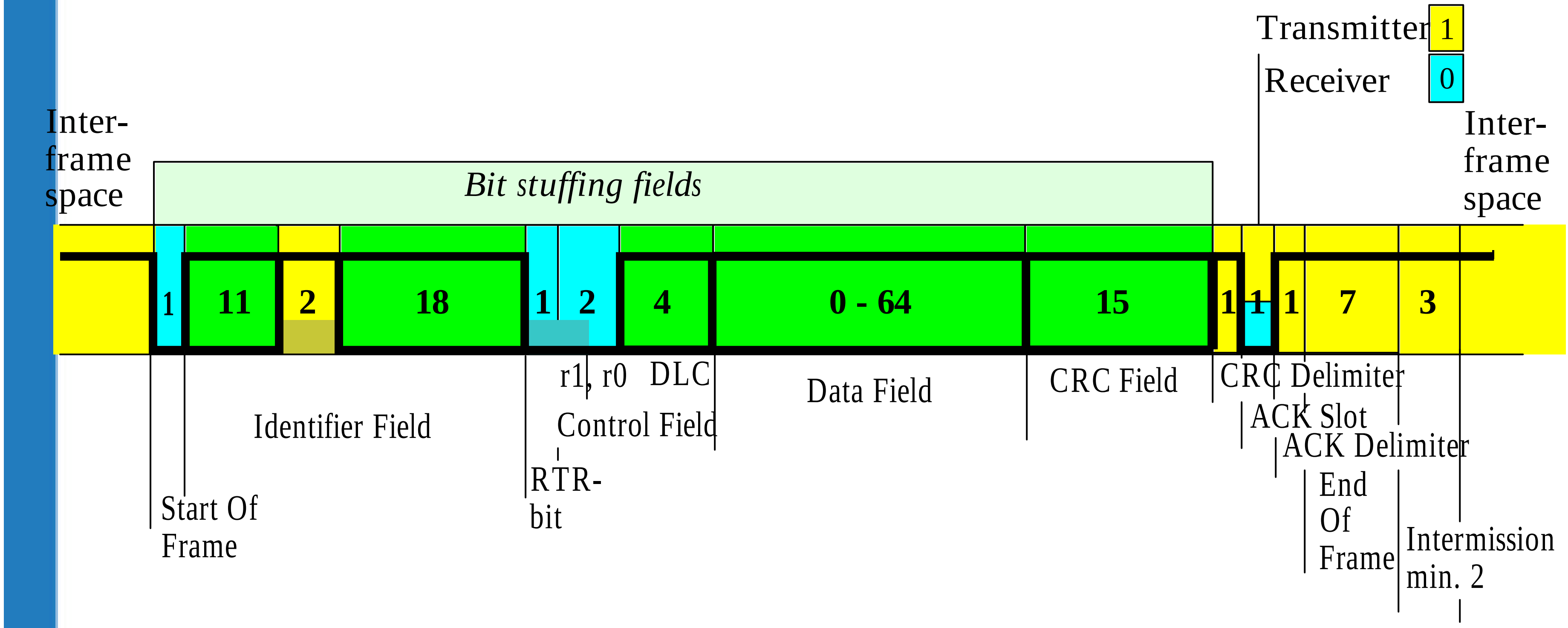


Five consecutive bits of same polarity renders a stuff bit

$$36/29 = 1.24 \;!!$$

KVASER Advanced CAN Solutions

CAN Data Frame Ext.



Bit values

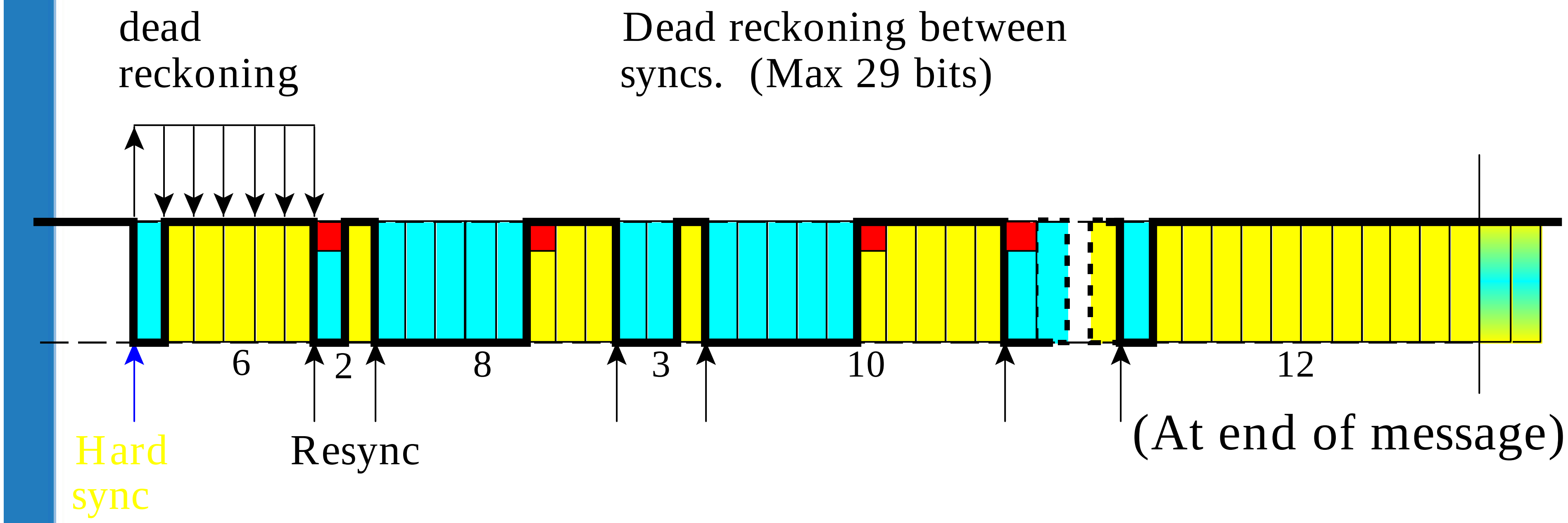
🟦 0

🟩 0/1

🟨 1

KVASER
Advanced CAN Solutions

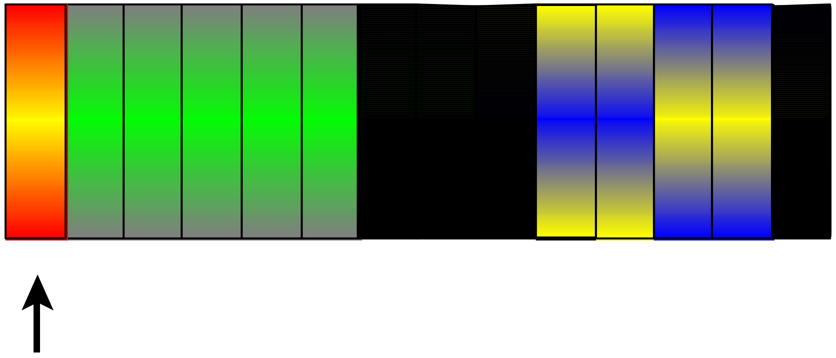
Synchronisation



↑ Sync. flank at falling edges

Sync_Seg

Synchronize the various CAN nodes on the bus. An edge is expected within this segment.

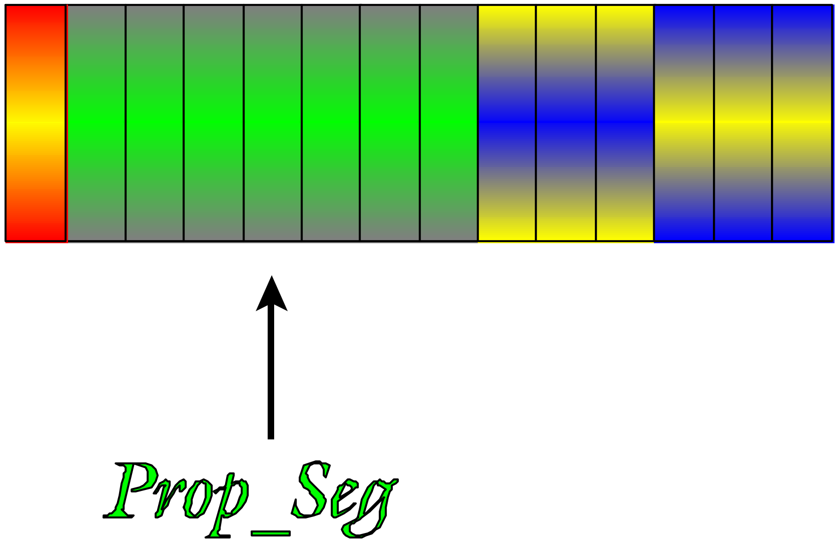


Sync_Seg 1

1 Time quantum

Prop_Seg

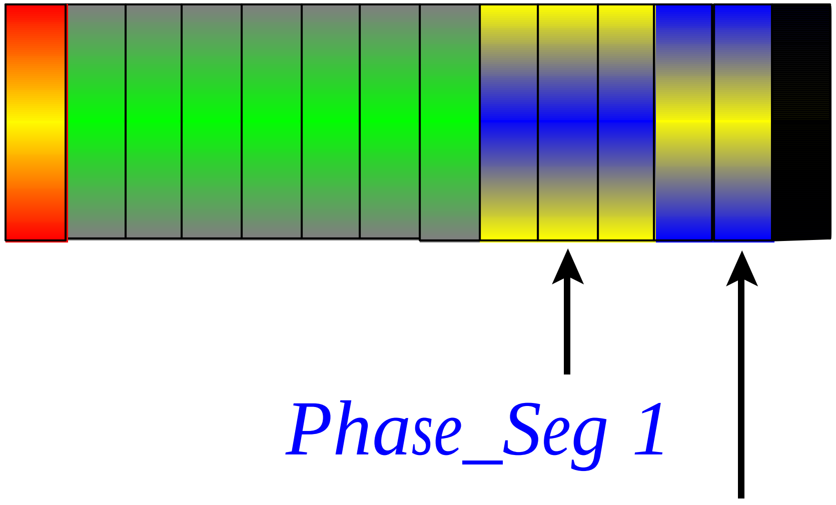
Compensate for physical delay times on the bus and node interfaces



1 - 8 Time quanta (min)

Phase_Seg1 & 2

Compensate for phase errors



Phase_Seg 1

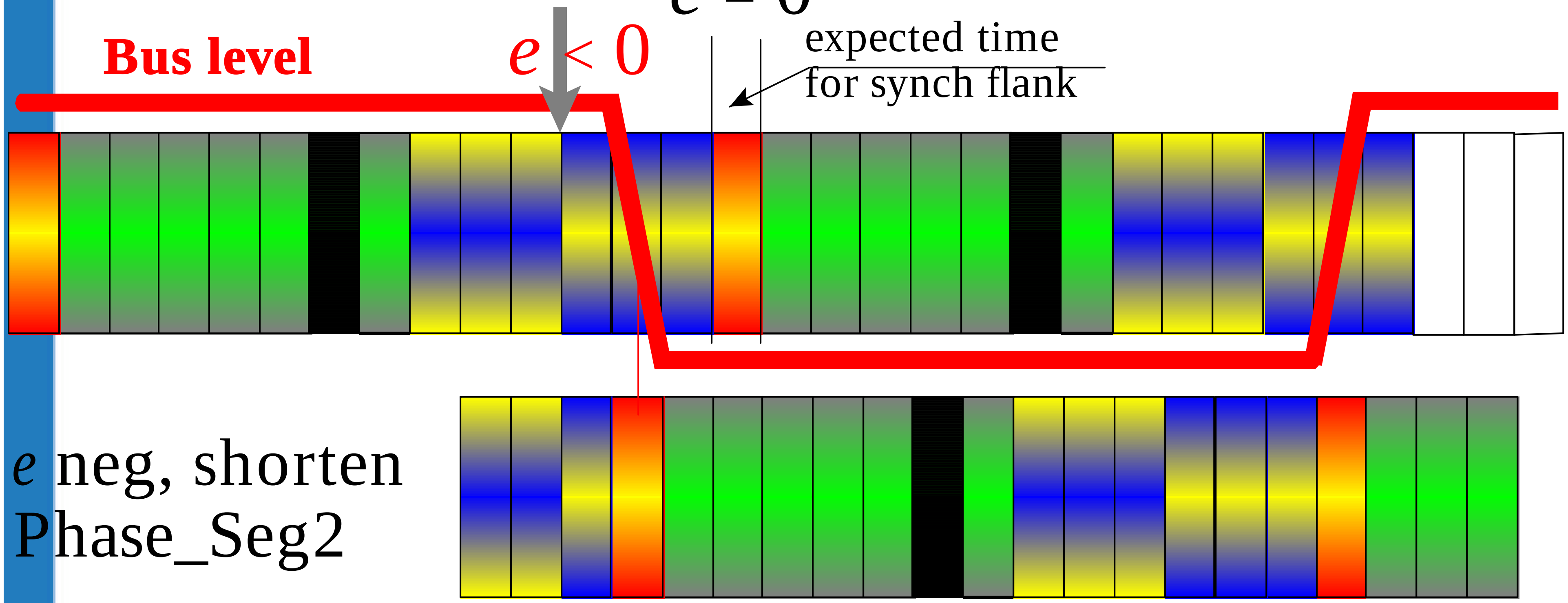
Phase_Seg 2

Phase_Seg1 1 - 8 Time quanta (min)
Phase_Seg2 = Phase_Seg1 (or information processing time)

Resynchronisation

Synch flank detected after sampling point (prematurely)

e = Phase error



$e = 0$

Bus level

$e < 0$

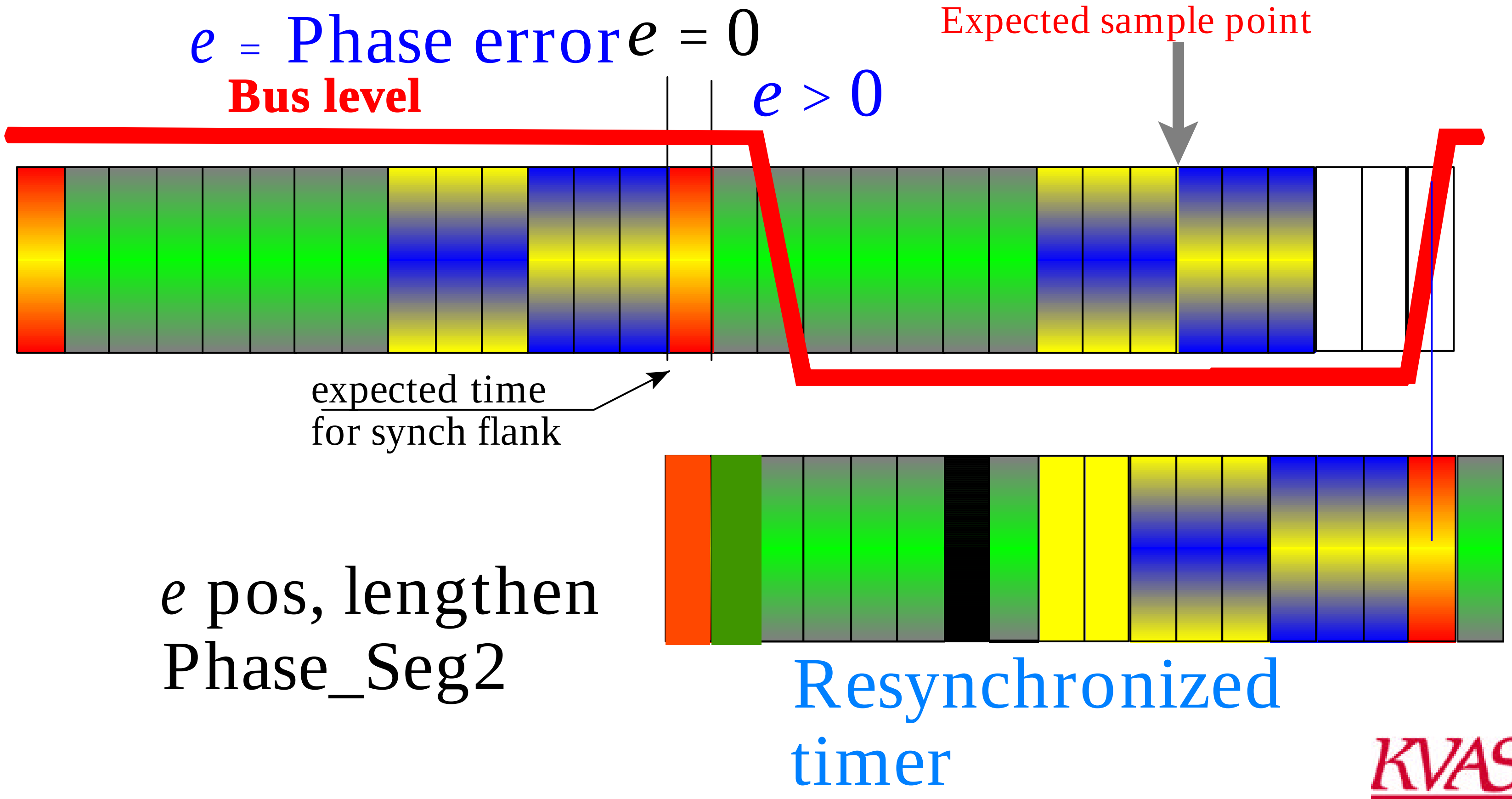
expected time for synch flank

e neg, shorten Phase_Seg2

Resynchronized timer

Resynchronisation

Synch flank detected before sampling point (later than expected)

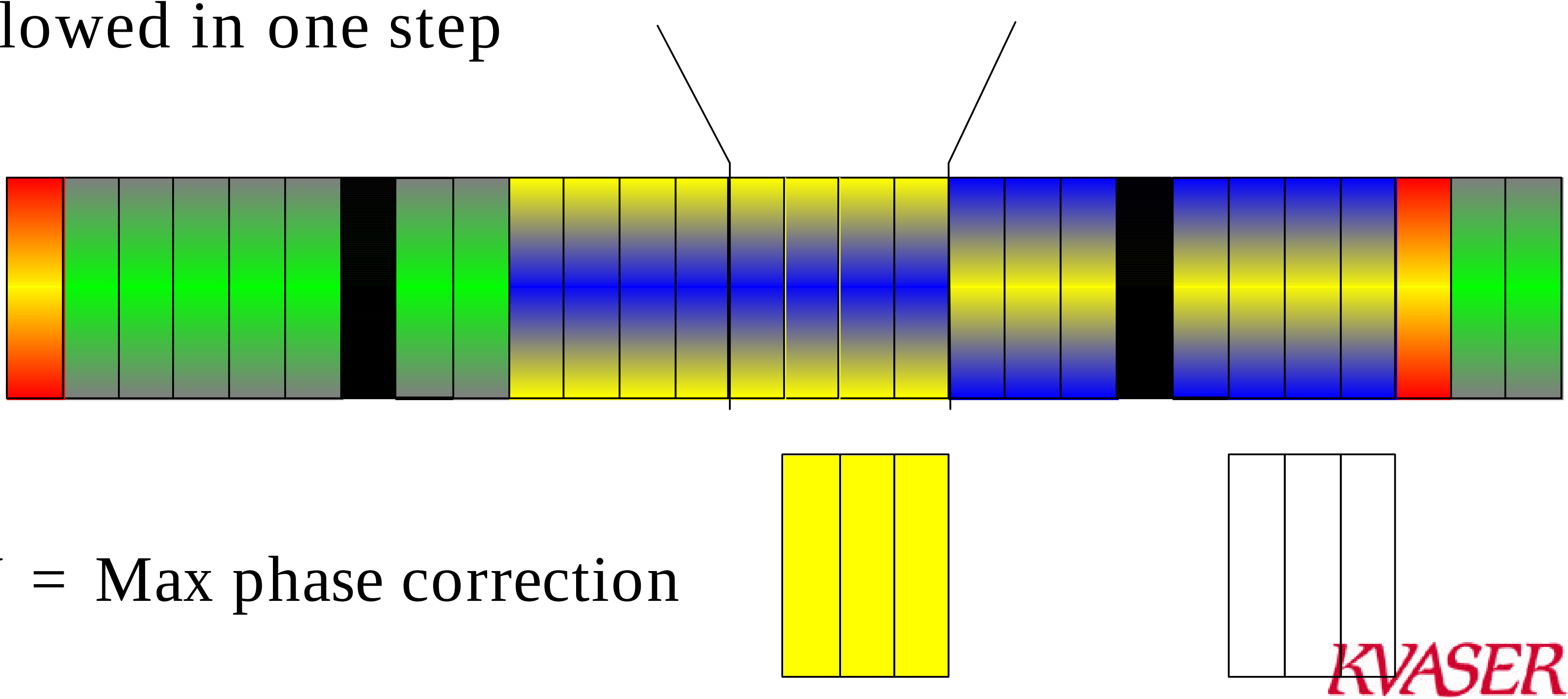


e pos, lengthen Phase_Seg2

Sync Jump Width

Maximum synch compensation allowed in one step

Max 4(min), Phase seg 2



SJW = Max phase correction

Advanced CAN Solutions

Oscillator tolerance range

$$(1\text{-}df)\cdot f_{nom} \quad f_{osc} \quad (1+df)\cdot f_{nom}$$

Conditions:

- df min(Phase_Seg1, Phase_Seg2)/[2(13·TBIT - Phase_Seg2)]
- df SJW/(20TBIT)
- Max diff. between two osc. is 2dffnom

CALCULATIONS

- Tscl = Tclk * BRP * 2 = Tclk*(BRP + 1)*2
  - BRP the value in CAN-controller
  - (clk = 16 MHz and BRP = 0: Tclk = 62.5ns and Tscl =125 ns)
- Tseg1 = Tscl * (TSEG1) = Tscl * (TSEG1 + 1)
  - TSEG1 the value in CAN-controller
- Tseg2 = Tscl * (TSEG2) = Tscl * (TSEG2 + 1)
  - TSEG2 the value in CAN-controller
- Tsjw = Tscl * (TSJW1) = Tscl * (TSJW + 1)
  - TSJW the value in CAN-controller
- SJW = [1..4]

RULES

- TProp_Seg > (All delays) * 2
- TSeg2 > = 1 Tscl, CAN controller may demand minimum 2 Tscl.
- TSeg2 > = Tsjw
- TSeg1 > = Tsjw + TProp

Suggestion: Keep Phase segment Tsjw+1

KVASER
Advanced CAN Solutions