An Introduction to TTCAN
(Time Triggered Controller Area Network)

Rahul Shah & Xuanming Dong

EE290O Class Discussion

March 7, 2002

Brief History

- Developed by Bosch
- Standardized in ISO 11898
- First deployed in 1986
- Used in:
  - Automotive applications
  - Industrial automation
  - Embedded systems

CAN standards

- CAN version 1.0
- Standard CAN (version 2.0A)
- Extended CAN (version 2.0B)
- Time-Triggered CAN

CAN vs. LAN

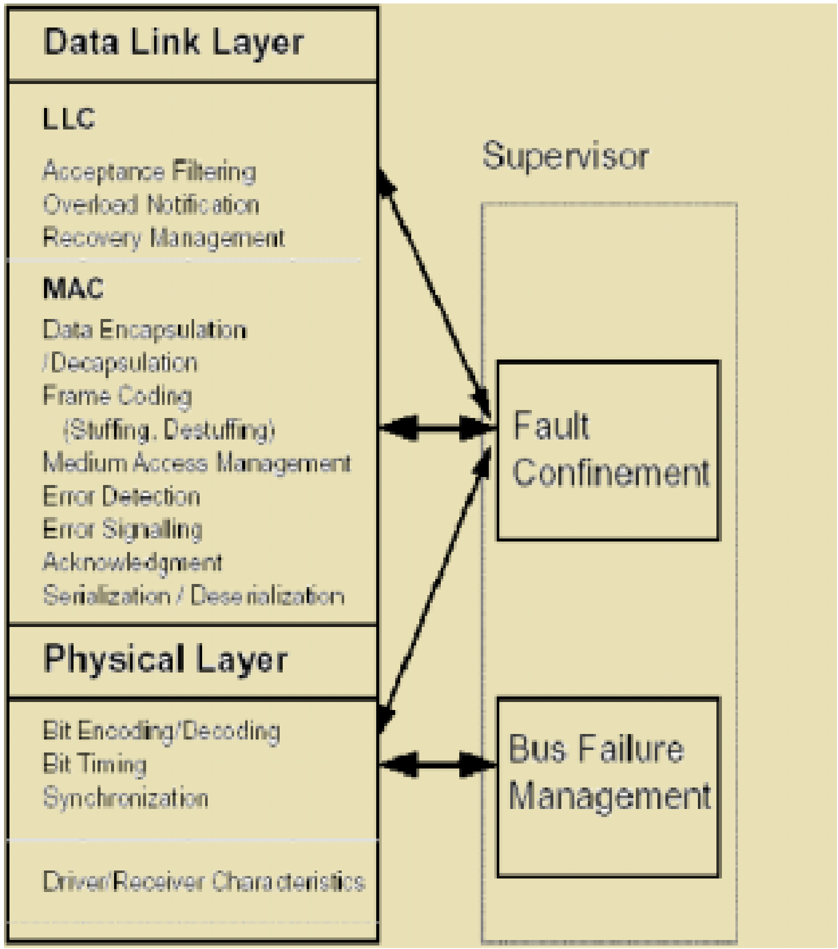
- Content addressing vs. node addressing
- Strict priority based system
- In case of collision, packet with higher priority gets through
- Better fault tolerance
- Engineered for known traffic patterns

Controller Area Network (CAN)

Layered Structure of a CAN Node

Application Layer
Object Layer –Message Filtering –Message and Status Handling
Transfer Layer –Fault Confinement –Error Detection and Signaling –Message Validation –Acknowledgement –Arbitration –Message Framing –Transfer Rate and Timing
Physical Layer –Signal Level and Bit Representation –Transmission Medium

CAN Protocol Stack



CAN Bit Rates

Bus length (m)	Max. bit rate (b/s)
40	1M
100	500k
200	250k
500	125k
6 km	10k

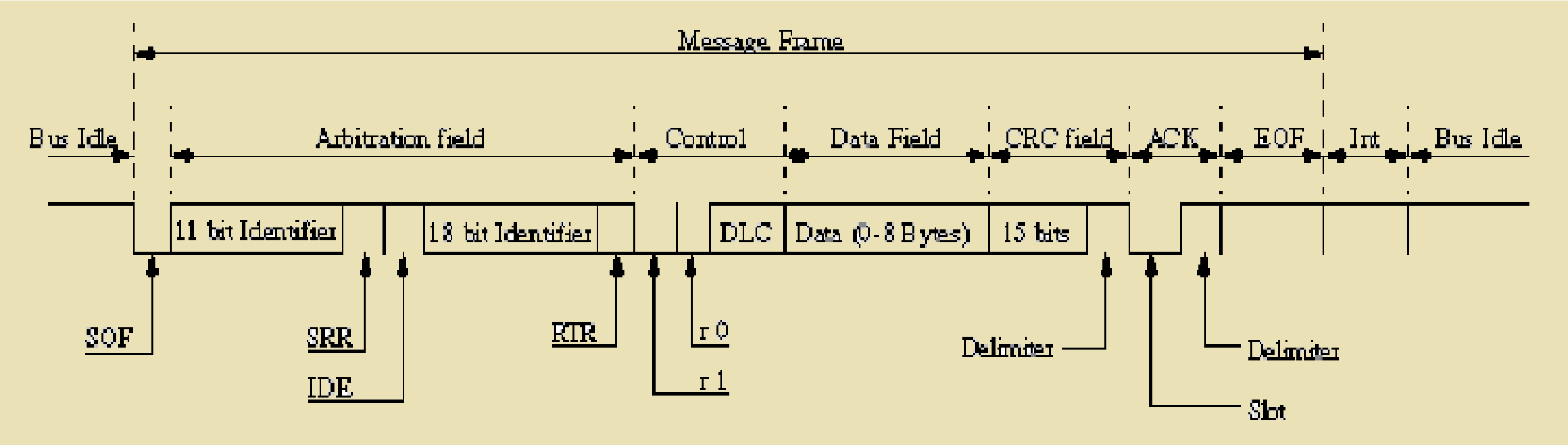
CAN Bus

- Broadcast bus connected in wired-AND
- Non-return to zero with bit stuffing
- Dominant bit – 0
- Recessive bit – 1
- Bit wise arbitration
- Tagged priority for every message

Message Types

- Data Frame

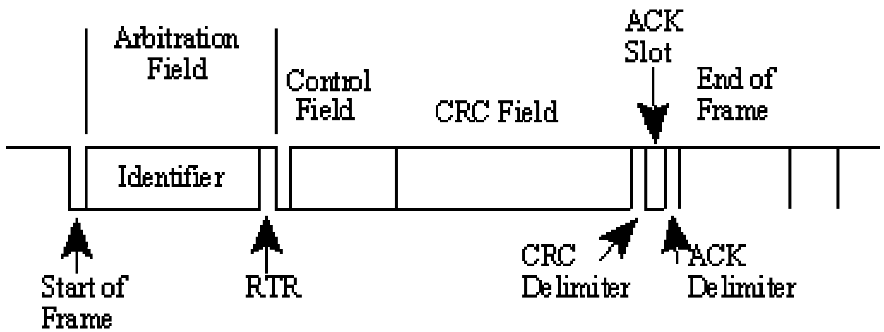
"Hello everyone, here's some data labeled X, hope you like it!"



Message Types

- Remote Frame

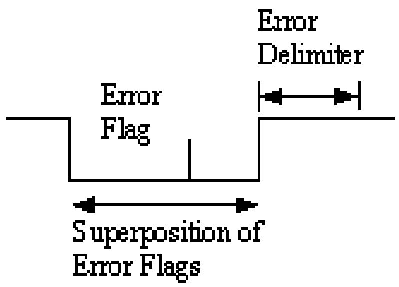
“Hello everyone, can somebody please produce the data labeled X?”



Message Types

- Error Frame

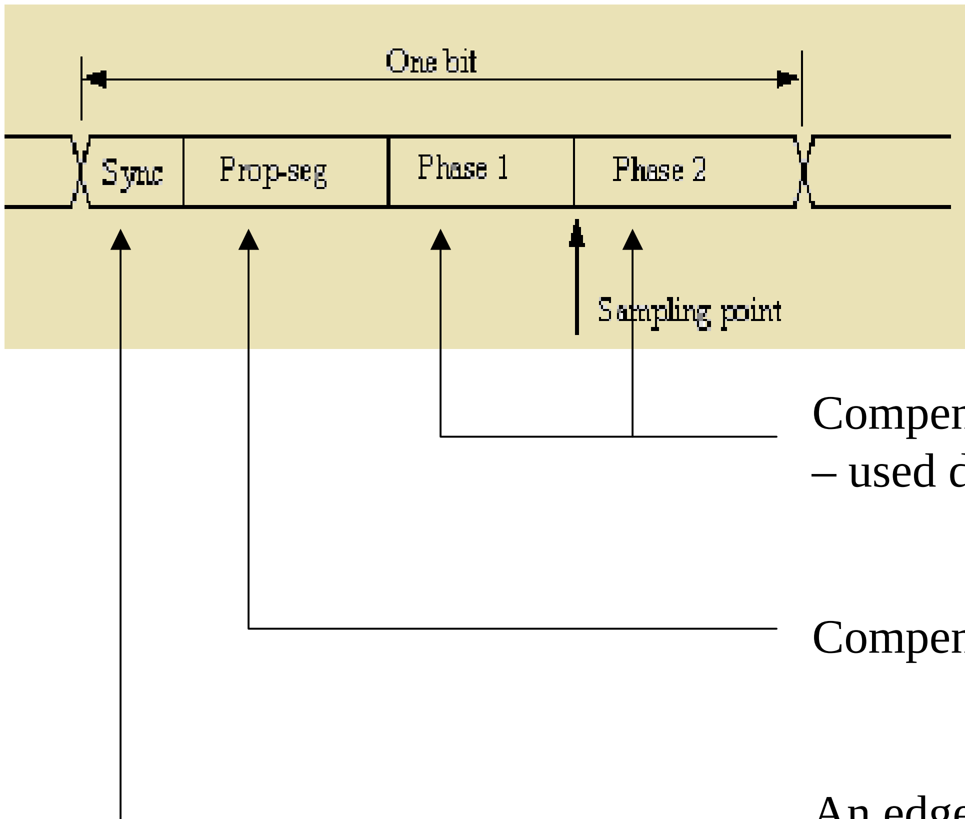
  (everyone, aloud) "OH DEAR, LET'S TRY AGAIN"



- Overload Frame

  "I'm a very busy little guy, could you please wait for a moment?"

Bit Timing



Compensates for edge phase errors
– used during resynchronization

Compensates for physical delay on the bus

An edge is expected in this segment

Error Detection Mechanisms

- Bit monitoring
- Bit stuffing
- Frame check
- Acknowledgement check
- Cyclic redundancy check (CRC)

Fault Confinement

- Unit may be in three states:
  - Error active
  - Error passive
  - Bus off
- Two registers are maintained in every unit:
  - Transmit error count
  - Receive error count

CAN Advantages

- Mature standard
- Hardware implementation of the protocol
- Simple transmission medium
- Excellent error handling
- Fine fault confinement

Time Triggered Controller Area Network (TTCAN)

Motivation of TTCAN

- Under the bitwise arbitration of CAN, the access may be delayed, if some other message is already in the process of transmission or if another message with higher priority also competes for the bus.
- Even the message with the highest priority may experience a small latency.
- The lower the priority of a message is, the higher the latency jitter for the media access

The Goals of TTCAN

- Reduce latency jitters
- Guarantee a deterministic communication pattern on the bus
- Use the physical bandwidth of a CAN network much more efficiently

TTCAN: Two Level Extension

- Extension level 1
  - Guarantees the time triggered operation of CAN based on the reference message of a time master
  - Fault tolerance of that functionality is established by redundant time masters (potential time masters)
- Extension level 2
  - A globally synchronized time base is established
  - A continuous drift correction among the CAN controllers is realized

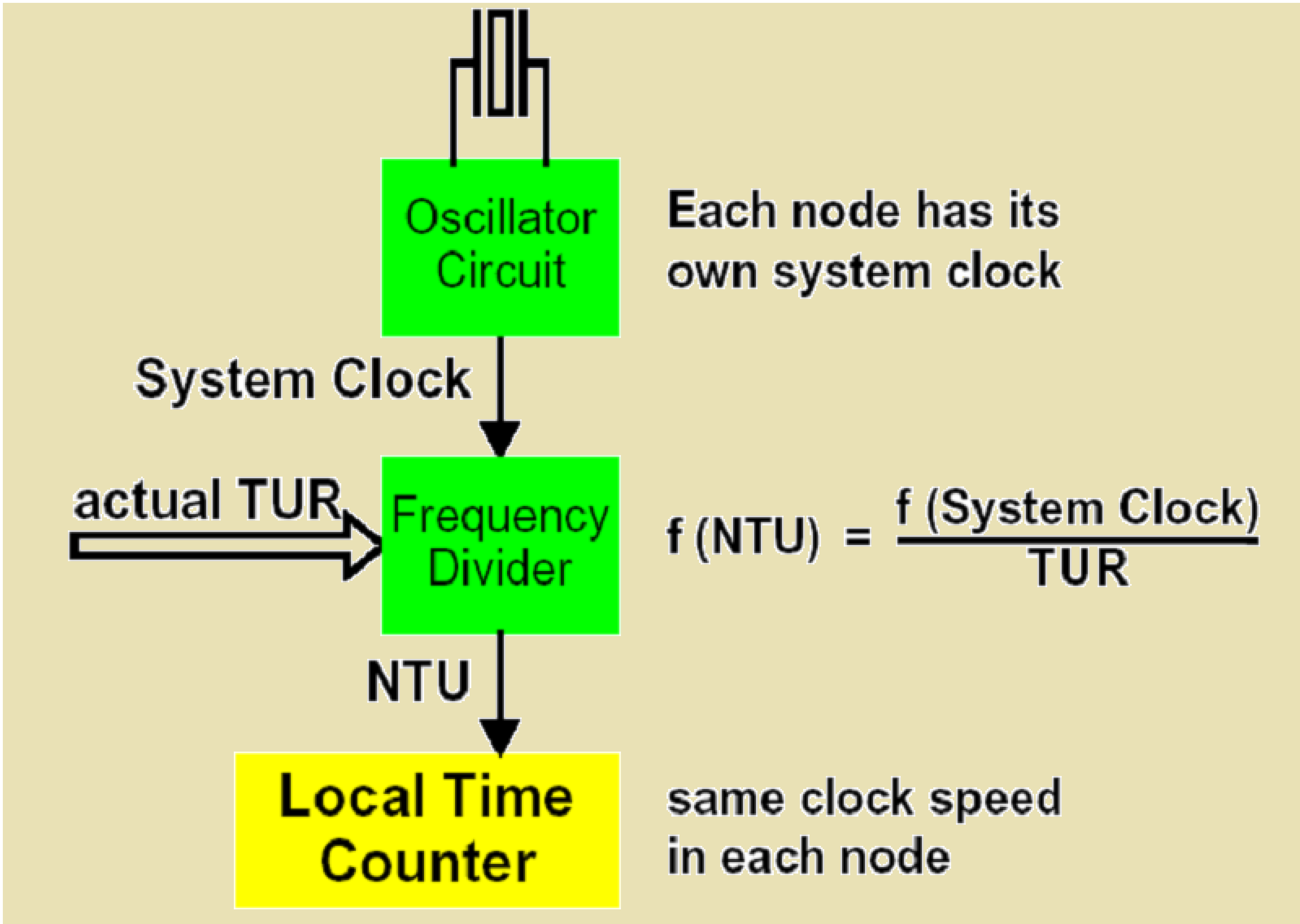
Overview of TTCAN Mechanisms

- Time bases, provided either by an internal or by an external clock
  - Local_Time, Cycle_Time, and Global_Time
- Three different types of time windows
  - Exclusive time window, arbitrating time window, and free time window
- Time-triggered and periodic communications clocked by a time master's reference message
- The system matrix specifies the sequence of messages transmitted in each basic cycle
- At least one event-trigger should be freely programmable by the control unit

Local Time

- Local_Time, a 16 bit integer value that is incremented each Network Time Unit (NTU)
- The length of the NTU is the same for all nodes
- It is generated locally, based on the local system clock period t_{sys} and the local Time Unit Ratio (TUR), $NTU=TUR*t_{sys}$
- Different system clocks in the nodes require different (non-integer) TUR values
  - TUR is a non-integer value and may be adapted to compensate for clock drift or to synchronize to an external time base.

Local Time

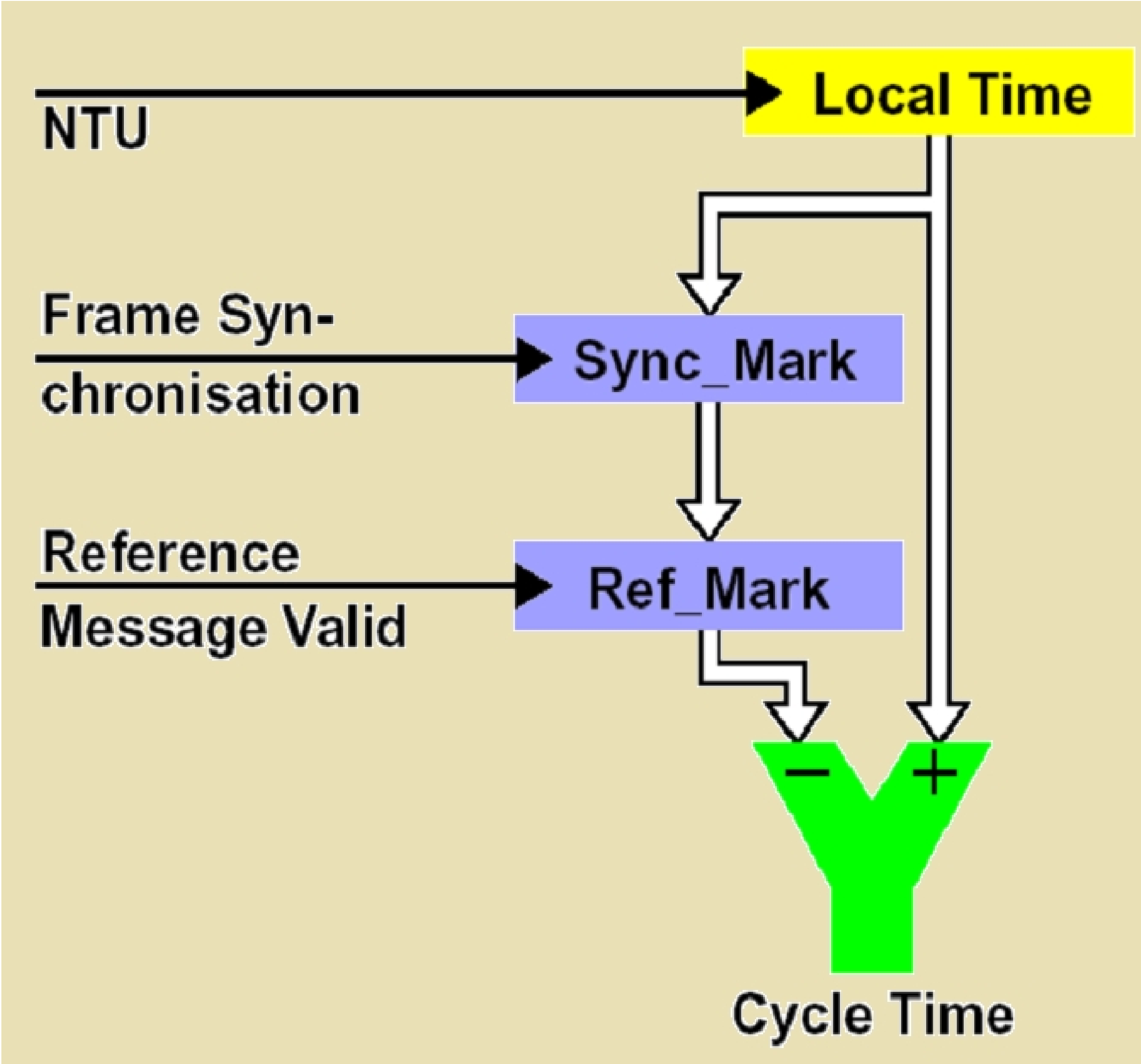


Oscillator Circuit
Each node has its own system clock
System Clock
actual TUR
Frequency Divider
f (NTU) = f (System Clock) / TUR
NTU
Local Time Counter
same clock speed in each node

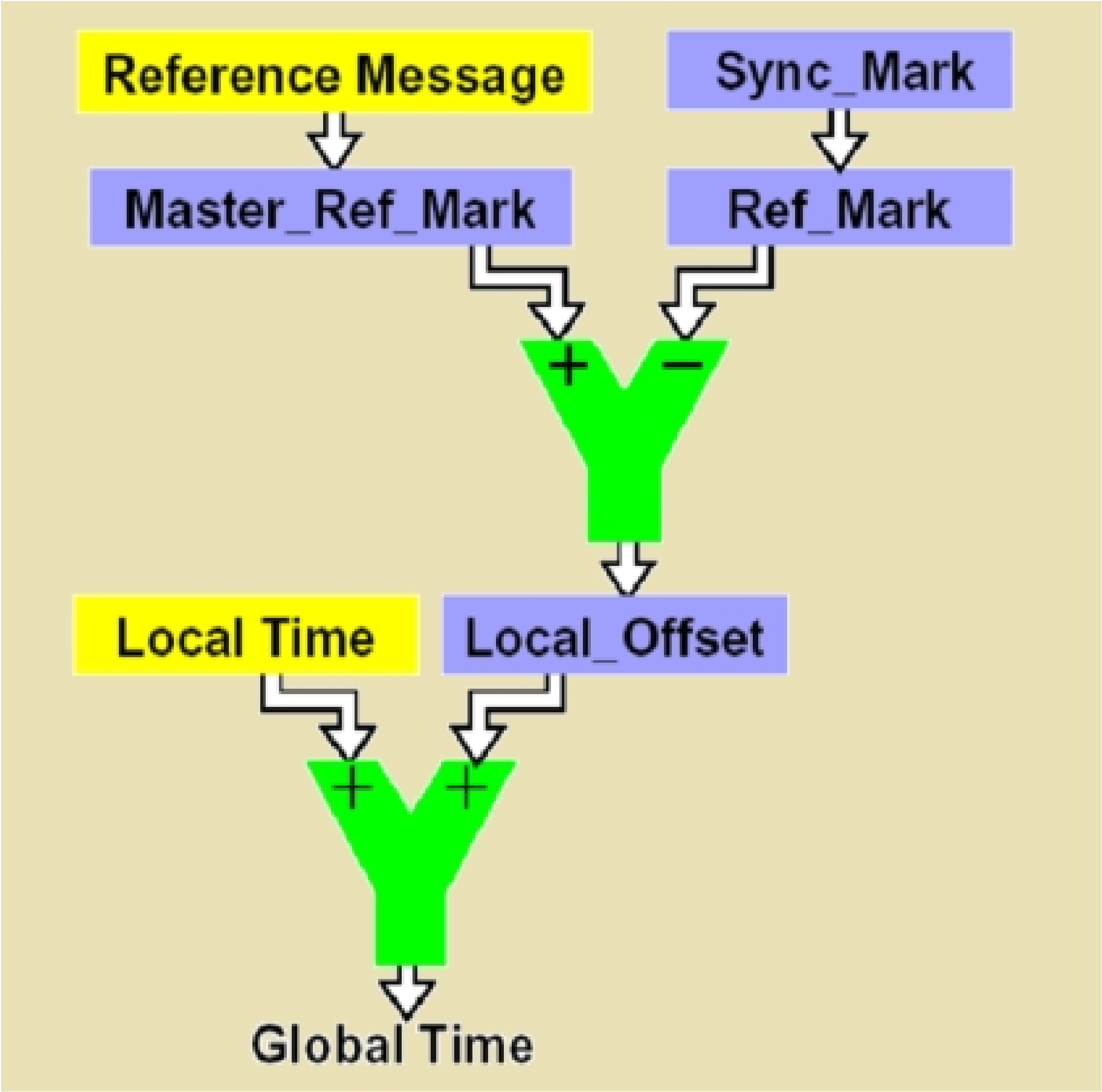
Cycle Time

- Each valid Reference Message starts a new basic cycle and causes a reset of each node’s Cycle_Time.
- The value of Local_Time is captured as Sync_Mark at the start of frame (SOF) bit of each message. When a valid Reference Message is received, this message’s Sync_Mark becomes the new Ref_Mark
- Cycle_Time is the actual difference between Local_Time and Ref_Mark, restarting at the beginning of each basic cycle when Ref_Mark is reloaded
- Even in a software implementation of TTCAN, the capturing of Local_Time into Sync_Mark at each SOF must be done in hardware

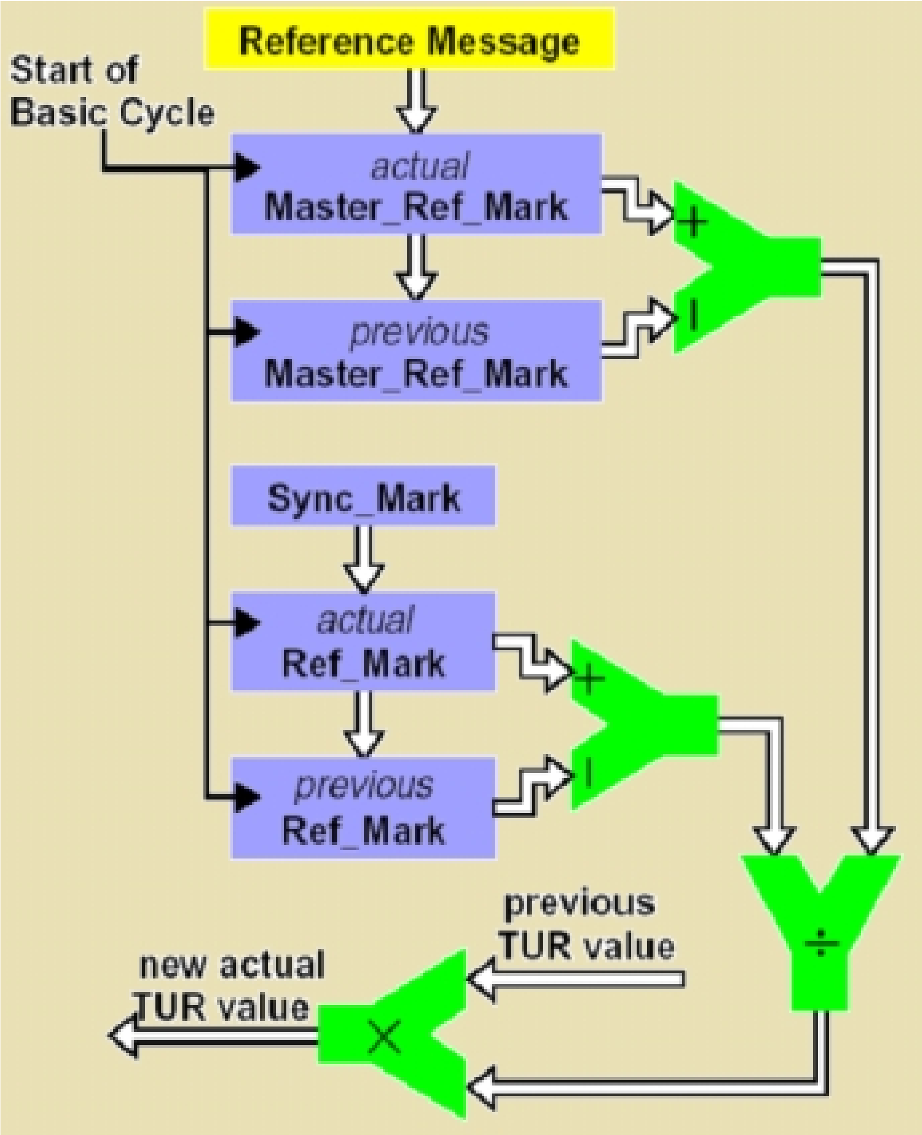
Cycle Time


NTU
Local Time
Frame Syn-
chronisation
Sync_Mark
Reference
Message Valid
Ref_Mark
−
+
Cycle Time

Global Time


Reference Message
Sync_Mark
Master_Ref_Mark
Ref_Mark
+
−
Local Time
Local_Offset
+
+
Global Time

Drift Compensation



Time Windows

- Exclusive window
  - Exclusively reserved for one message, without competition for the CAN network access
  - The automatic retransmission of messages that could not be transmitted successfully is disabled, guaranteeing that messages in exclusive time windows are not delayed
- Arbitrating window
  - During which messages can compete for the bus by the bitwise arbitrating mechanism of CAN
- Free window
  - Reserved for further extensions of the network

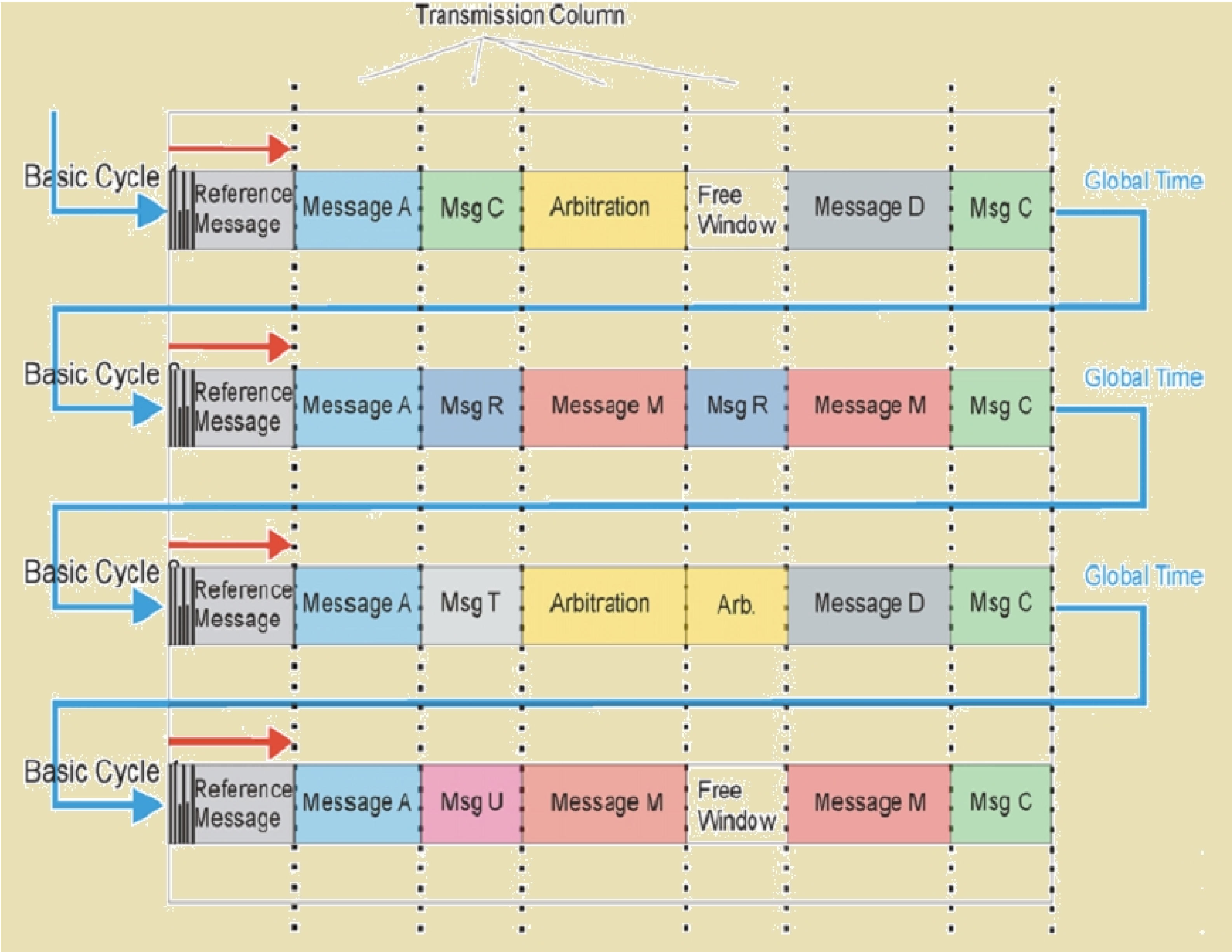
The Reference Message

- The reference message can be easily recognized by its identifier
- In extension level 1
  - The reference message only holds some control information of one byte, the rest of a CAN message can be used for data transfer
- In extension level 2
  - The reference message holds additional control information, e.g. the global time information of the current TTCAN time master
  - The reference message of level 2 covers 4 bytes while downwards compatibility is guaranteed. The remaining 4 bytes are open for data communication as well

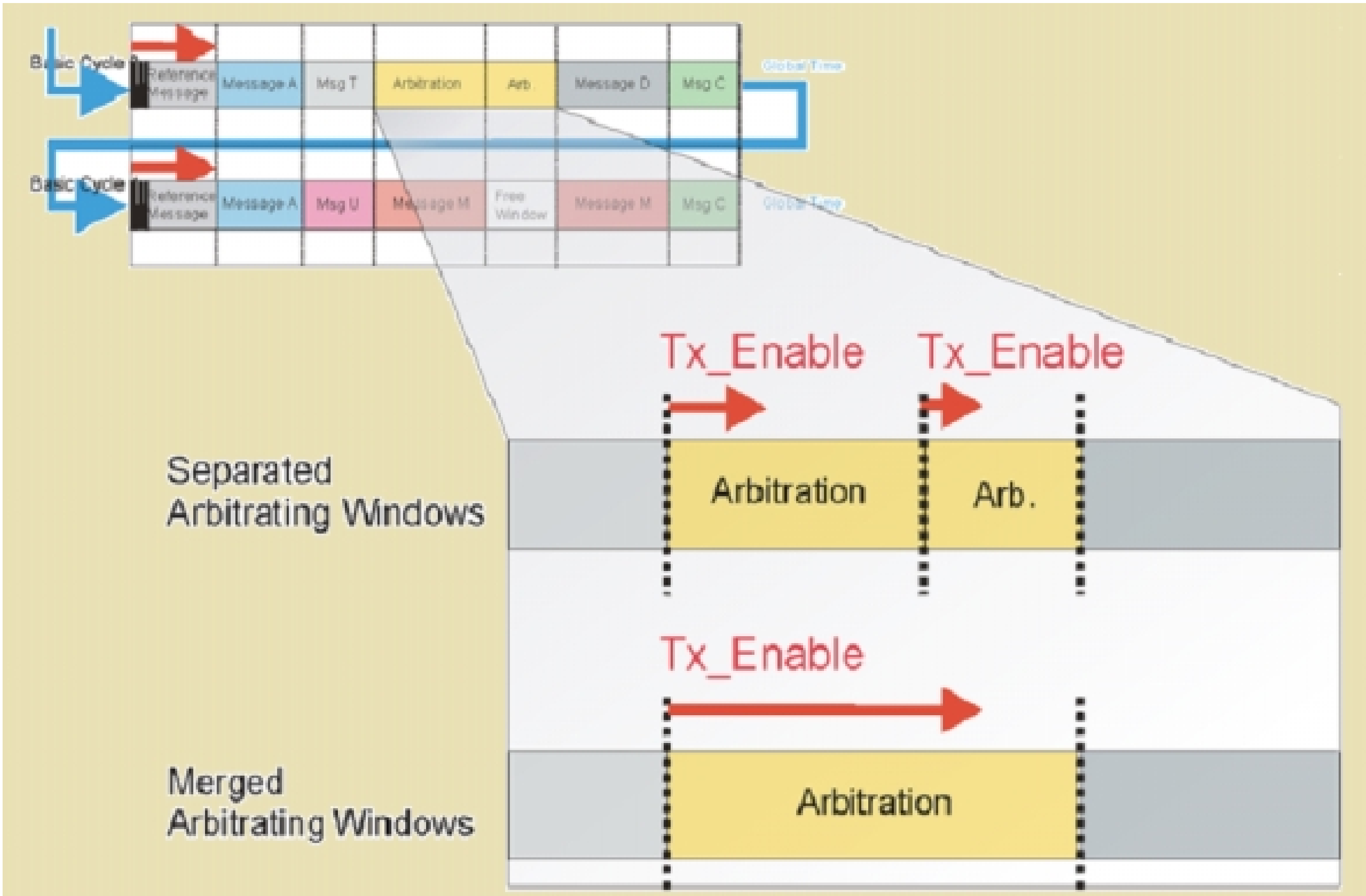
The System Matrix

- In TTCAN not all basic cycles necessarily have to be the same
- Different basic cycles are distinguished by the cycle count
  - A cycle count is incremented each cycle up to the maximum value after which it is restarted again
- System matrix is obtained by combining all those different cycles
  - It represents the communication overview of a TTCAN network
- System matrix allows another useful exception
  - Ignore the columns in the case of two or more arbitrating time windows in series

Example of a TTCAN System Matrix

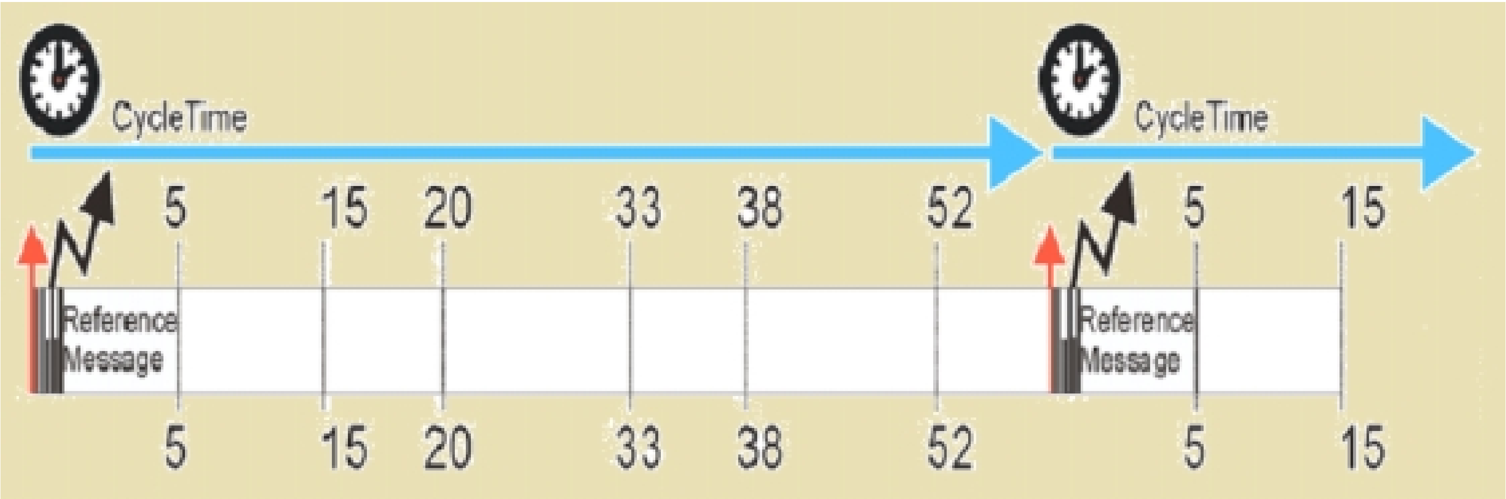


Merged Arbitrating Windows



Cycle Time and Time Marks

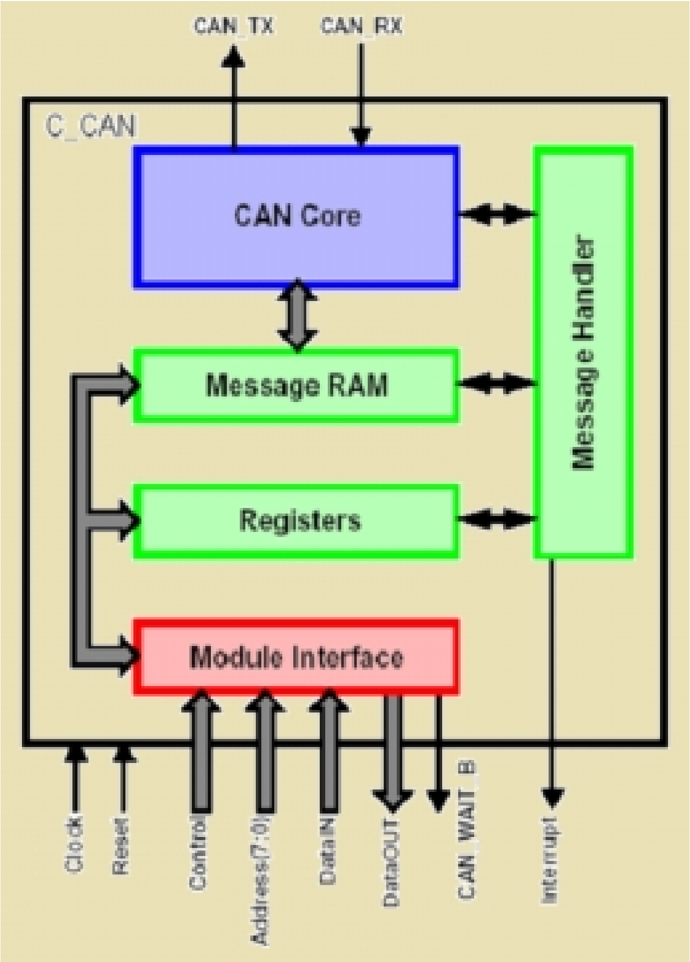
- A time mark furthermore consists of the base mark and the repeat count information.
  - The base mark determines the number of the first basic cycle after the beginning of the matrix cycle in which the message must be sent/received.
  - The repeat count determines the number of basic cycles between two successive transmissions/receptions of the message.



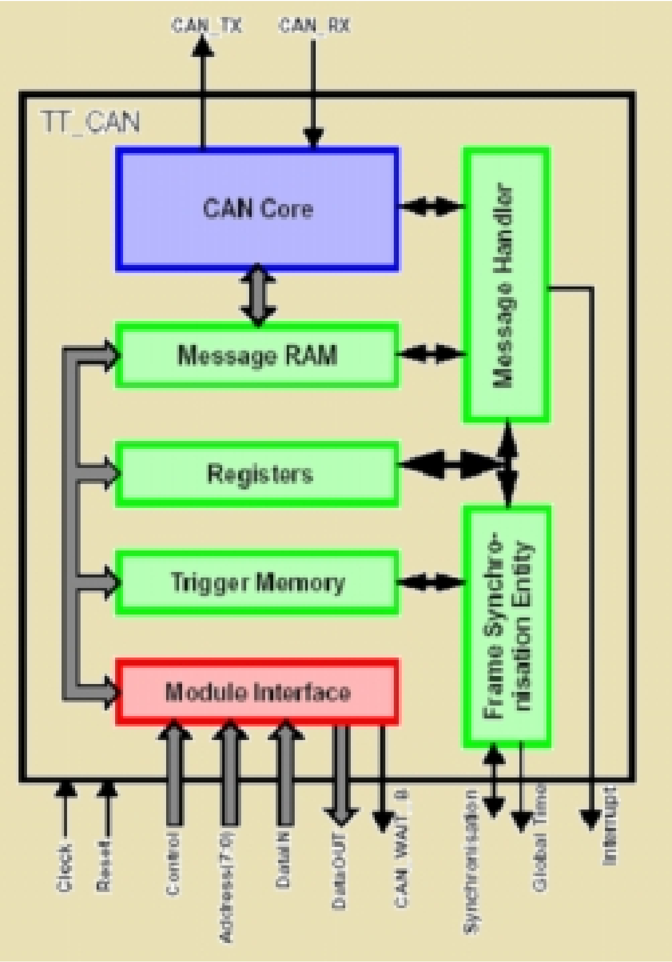
TTCAN Implementation

- The TTCAN is expanded by two functional blocks: the Trigger Memory and the Frame Synchronization Entity (FSE)
- The Trigger Memory stores the time marks of the system matrix that are linked to the messages in the Message RAM; the data is provided to the Frame Synchronization Entity
- The Frame Synchronization Entity is the state machine that controls the time triggered communication. It synchronizes itself to the reference messages on the CAN bus, controls the cycle time, and generates Time Triggers. It is divided into six blocks:
  - TBB: Time Base Builder
  - CTC: Cycle Time Controller
  - TSO: Time Schedule Organizer
  - MSA: Master State Administrator
  - AOM: Application Operation Monitor
  - GTU: Global Time Unit

TTCAN Implementation



C_CAN



TT_CAN