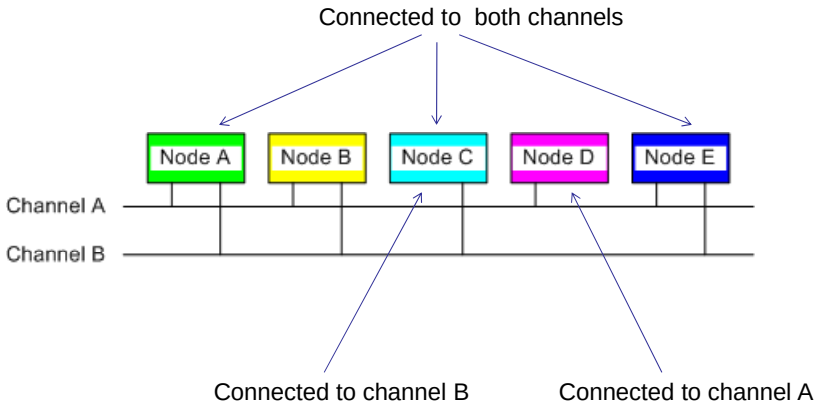
FlexRay

FlexRay

- Developed by Daimler Chrysler in collaboration with BMW
- Based on the BMW protocol Byteflight
- A fault tolerant dual channel bus
- A node can be connected to one or both of the busses
- A node connected to both busses can send the same or different messages on the two busses
  - Sending the same message on both busses increases the fault tolerancy
- Bit rate 10 Mbits/s

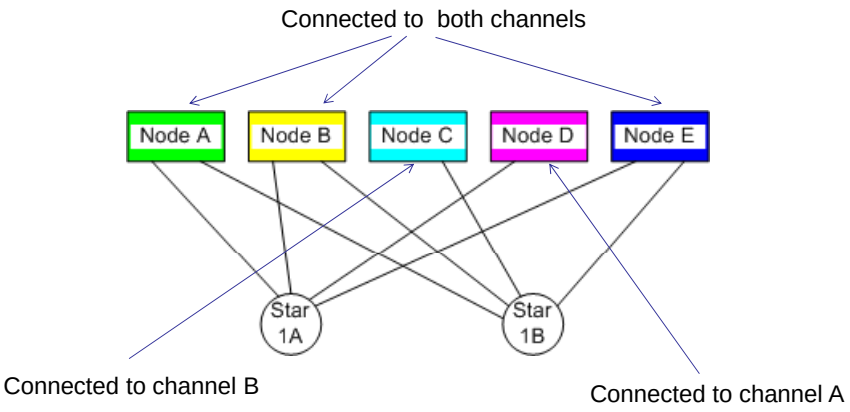
Bus topologies

Passive dual channel bus



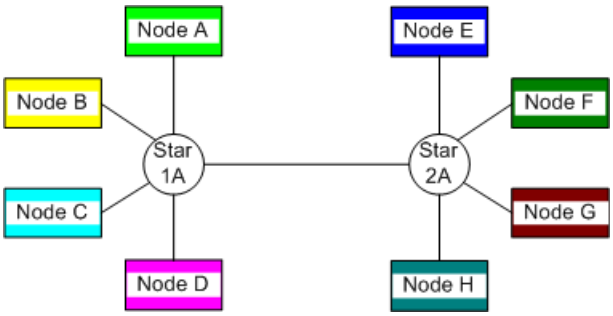
Bus topologies cont.

Dual channel single star bus



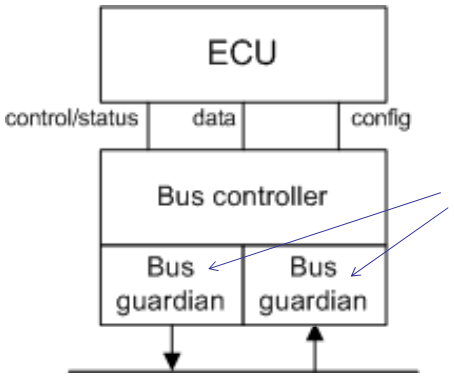
Bus topologies cont.

Single channel cascaded star bus



All nodes connected to channel A

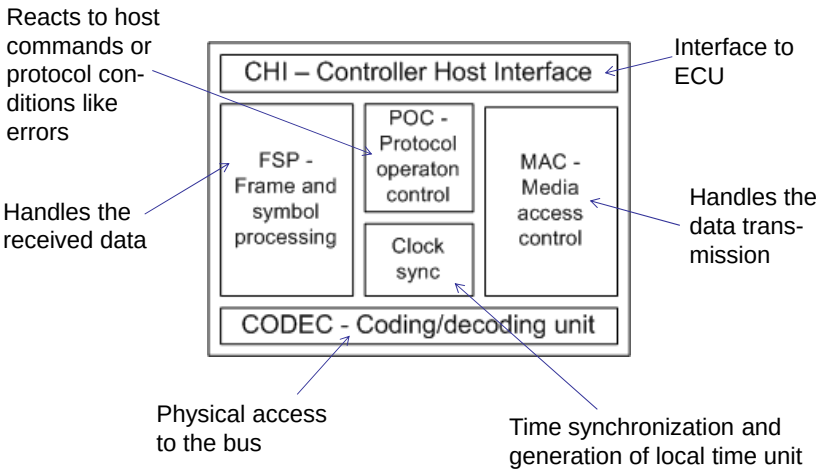
Node structure



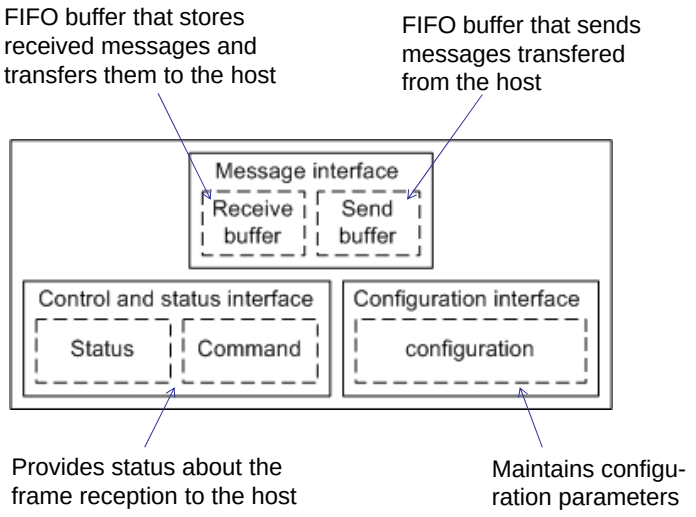
Disconnects the node from the bus when the node is sending or receiving garbage

ECU – Electronic Control Unit

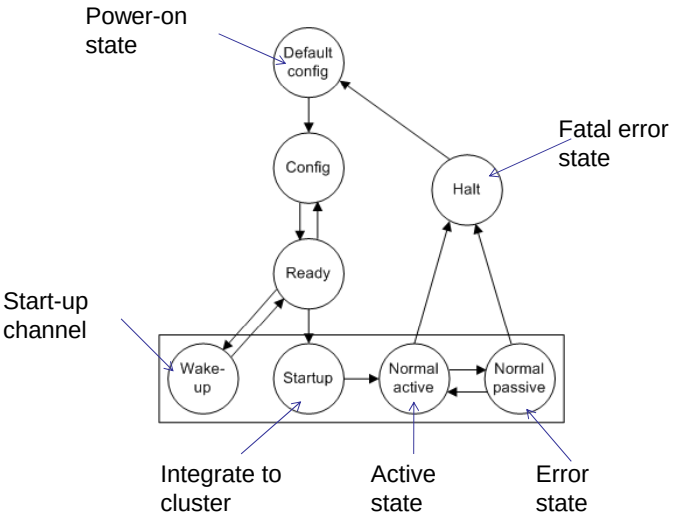
Bus controller



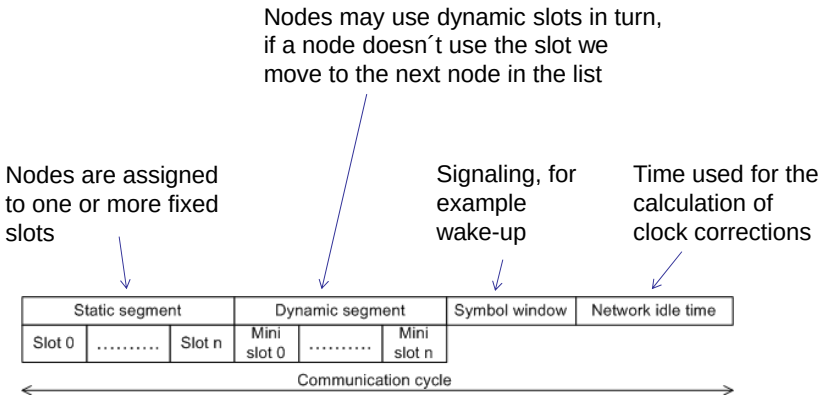
Controller host interface



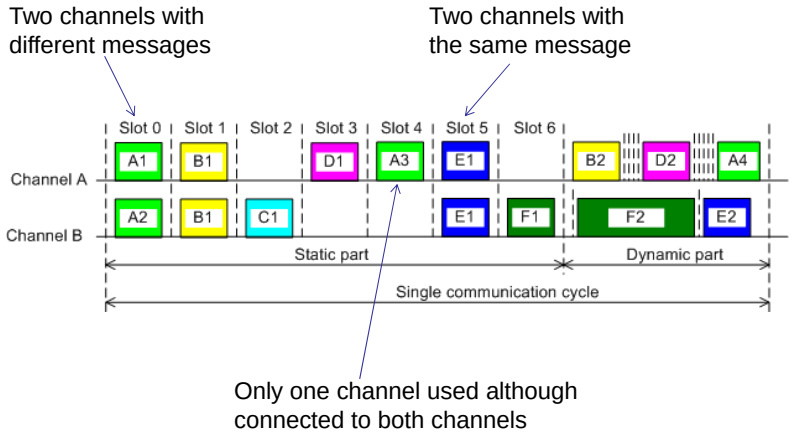
Protocol operation control



Communication cycle



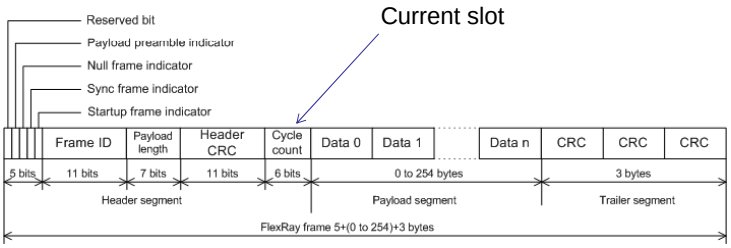
Communication cycle cont.



Communication cycle cont.

- Static segment
  - nodes are assigned slots where they have exclusive right to the bus
  - the nodes only know the number(s) of their slot(s) but nothing about the other slots
- Dynamic segment
  - nodes are offered slots according to the list of nodes. If a node doesn´t grab the slot within a short time we move on to the next node in the list
  - nodes can take up as much time as they need as long as the segment still lasts
  - This means that nodes with lower slot numbers assigned can block out nodes with higher slot numbers, we have a hierarchy
  - A node will not start using a slot if there is not enough time left in the segment

Frame format



The payload length field gives the number of data bytes in the payload segment divided by two

Clock synchronization

The nodes need to synchronize there clocks so that they only use their own time slots.

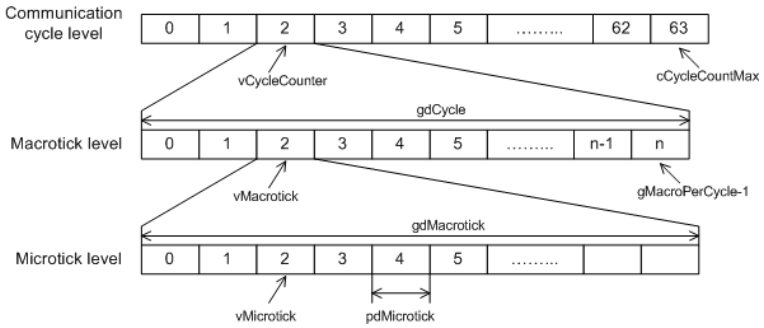
The static segment of a communication cycle contains synchronization slots.

A communication cycle is divided into an integer number of macroticks of equal length. There can be up to 64 macroticks in a cycle. These ticks are directly related to our slots.

The macroticks are synchronized at cluster (bus) level. Within tolerances the length of a microtick is identical for all nodes in the cluster.

Each macrotick is divided into an integer number of microticks. The number of microticks in a macrotick is controller specific

Clock synchronization cont.



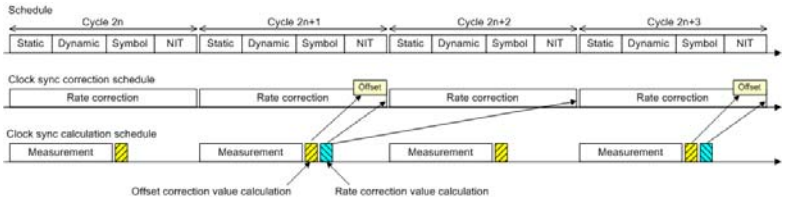
Clock synchronization cont.

The mission of the synchronization process is to make sure that the length of the macroticks in each node is the same and that the macroticks start at the same time in all nodes.

To do this we use two synchronization processes

- Rate correction where we use the microticks to correct the length of the node´s macroticks by adding or deleting microticks
  - Calculated over two communication cycles
- Offset correction where we insert microticks at one moment in time to correct the starting time of a new macrotick
  - Inserted on odd communication cycles

Clock synchronization cont.

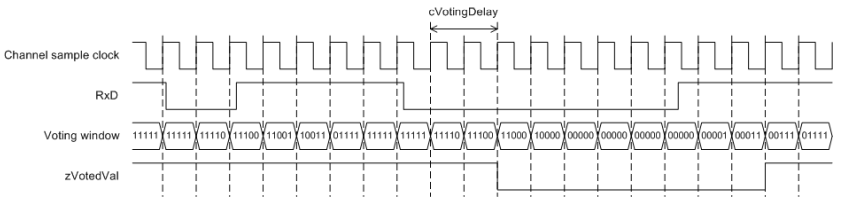


Reading of data bits

When a receiver is reading a data value it uses a majority vote scheme over five consecutive samples to decide the value

- If the last five samples gives more zeros than ones then the result is interpreted as a zero (’0’)
- If the last five samples gives more ones than zeros then the result is interpreted as a one (’1’)

Reading of data bits cont.



The number of zeros never exceeds the number of ones

The number of zeros exceeds the number of ones

The number of ones exceeds the number of zeros

Further reading

Manuals and other material can be downloaded from
www.flexray.com