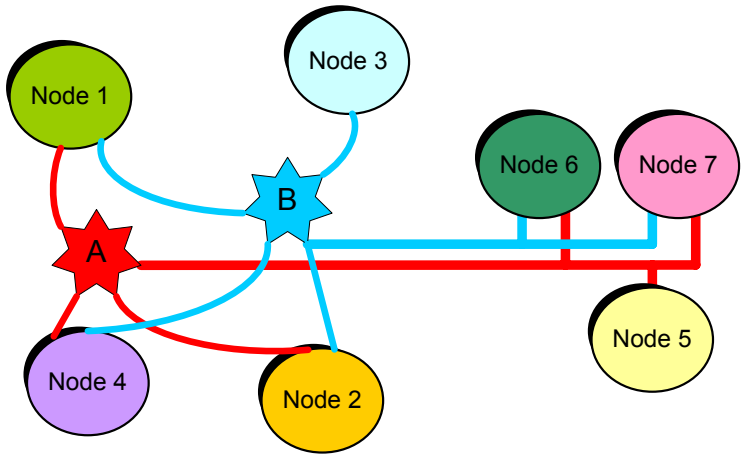
FlexRay

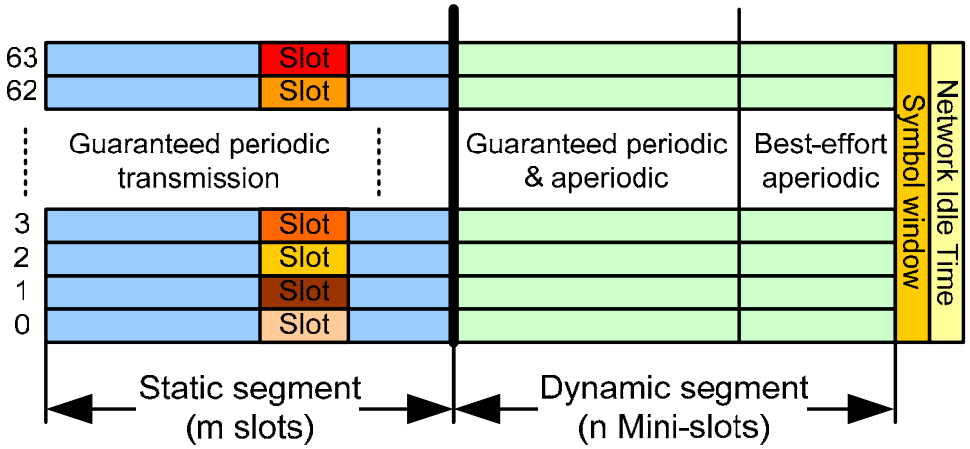


- 10 Mbit/s
- Dual or Single Bus or Star
- Electrical / optical PHY
- Flexible configuration
- Energy management

CEDES
Cost Efficient Dependable Electronic Systems

FlexRay contd.

- Static message segment
  - Periodic statically scheduled messages
  - Deterministic communication behaviour
- Dynamic segment
  - Event type signalling
  - Guaranteed service (low frame id:s)
  - Best effort service (high frame id:s)
  - Varying message times



- Design space:
- 64 cycles x 2048 IDs =...
- (including dynamic slots)

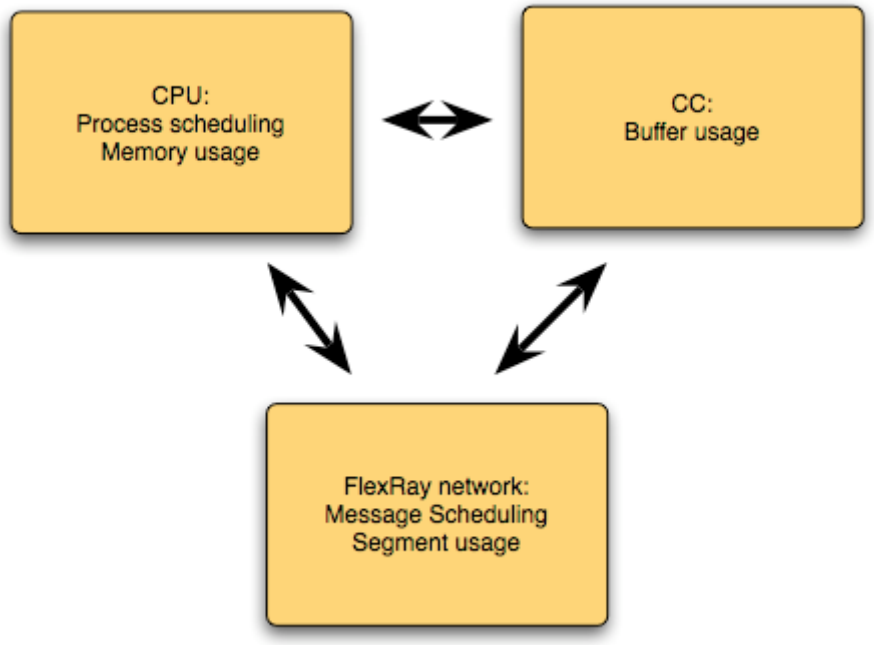
FlexRay - addressing

Ch A	1	2	3	4	5	6	7	8			9	10	11	12
Ch B	1	2	3	4	5	6	7	8	9	10	11			
	Static segment							Dynamic segment						

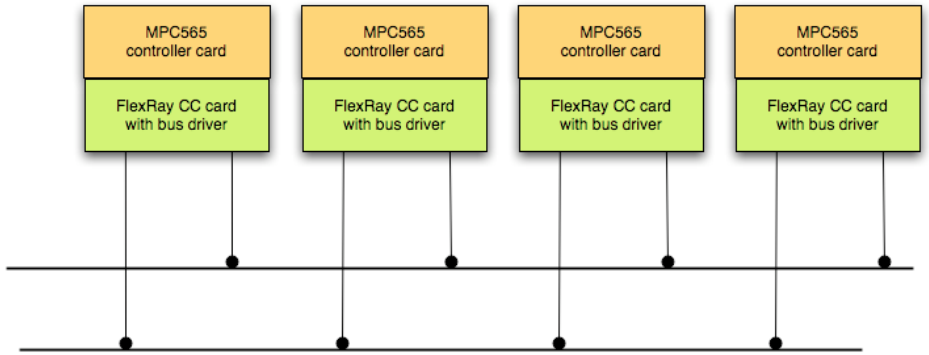
- Addressing based on frame-ID representing certain time slots
- IDs are assigned to the different nodes in the configuration state
- Only one node may transmit in a certain time slot
- Slot counters in static segment are incremented simultaneously on both channels at the end of each slot
- Slot counters in dynamic segment are incremented at the end of each minislot when there is no transmission or at the end of a frame when transmission occurs
- Slot counters in the dynamic segment are incremented individually on each channel.

Building a FlexRay system

- High design effort
- Global time allows for smart CPU scheduling



Physical platform



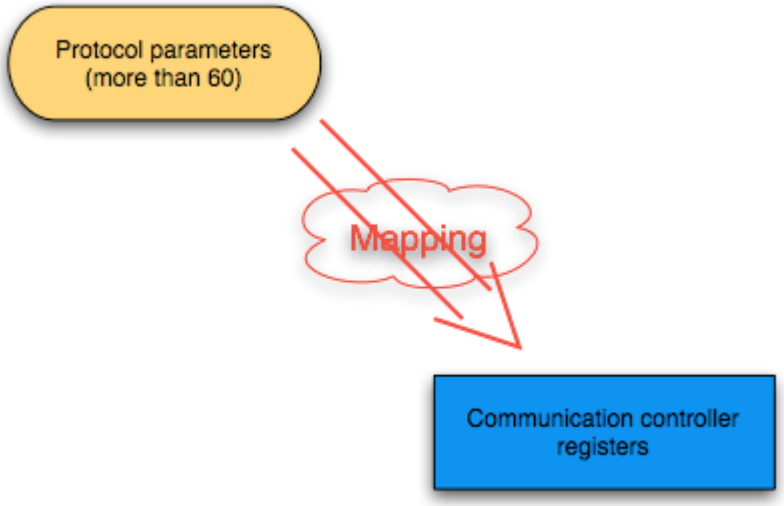
- Four nodes based on MPC565 processors and MFR4200 FlexRay controllers.
- Each node consists of a processor card and a CC-card with bus driver.
- Initial problem – integrate node components.

Node development approach

- Establish initial communication between the host and communication controller
- Write simple configuration to communication controller and transmit wakeup
- Develop driver software to write complete FlexRay configuration to communication controller
- Connect two communication controllers and try to get a sync on the FlexRay network
- Develop simple message buffer configuration routines for testing purposes

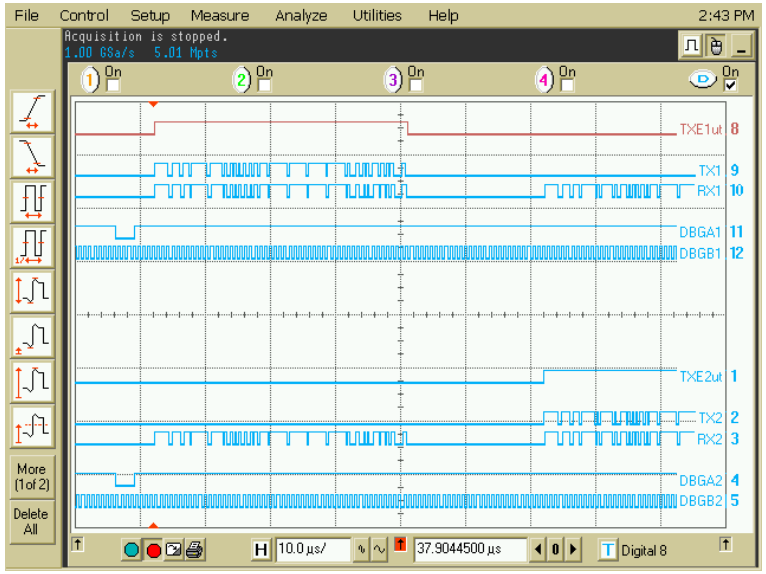
Controller Host Interface

- Controller addressed as external memory
- Read/Write timing on the CHI-bus can cause trouble
- No direct protocol implementation, mapping is needed
- The FlexRay controller requires a correct configuration to transmit anything
- Many dependent parameters, must fulfil protocol constraints

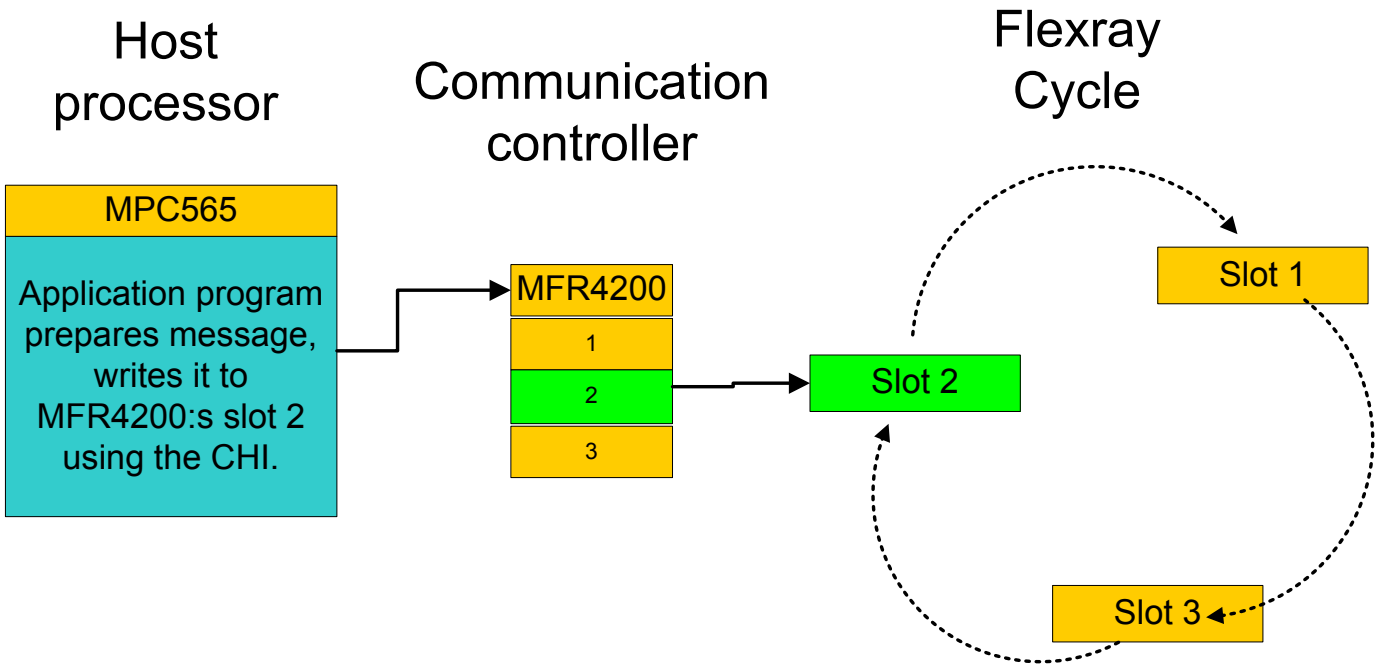


Cluster startup – lessons learned

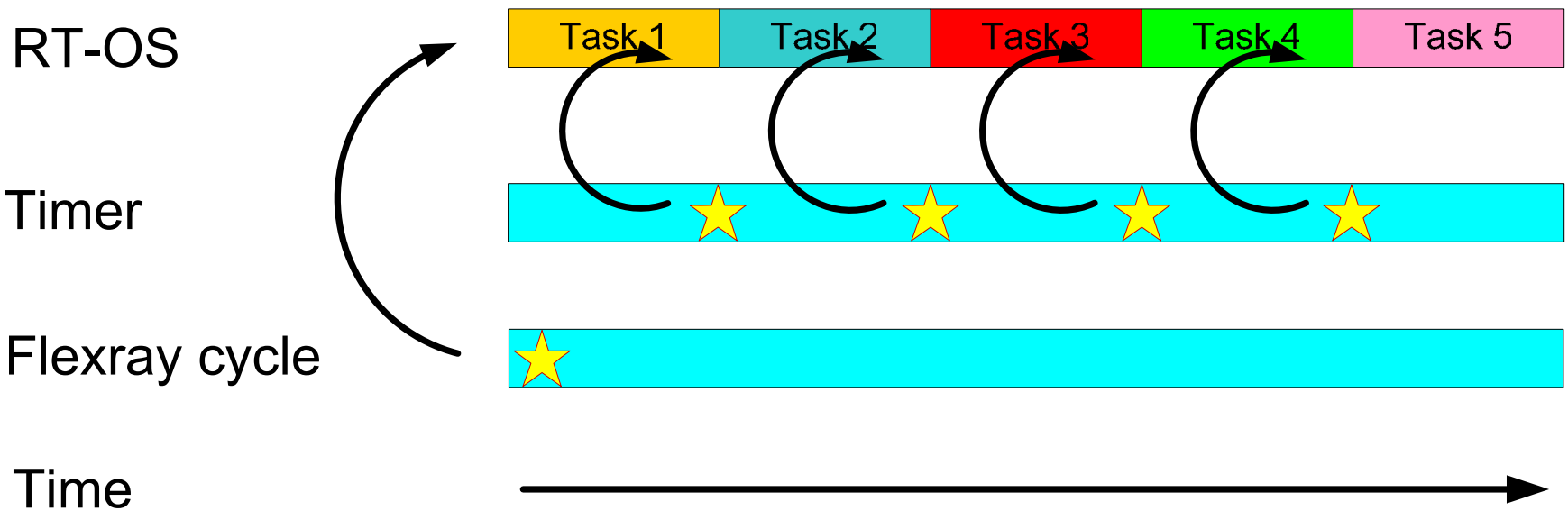
- At least two nodes must be configured as startup nodes
- All parameters must be correct and constraints fulfilled – configuration tool recommended
- Make sure the system is configured to send frames in certain slots – sync frames are not automatic
- There must be at least one odd frame and one even frame configured as sync frame
- Analysing the bus is very helpful to see what parameters may be wrong



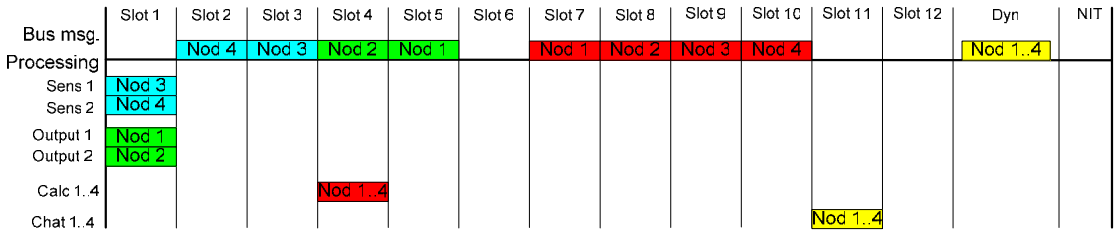
Using MFR4200 to send messages



Cluster global time - implementation

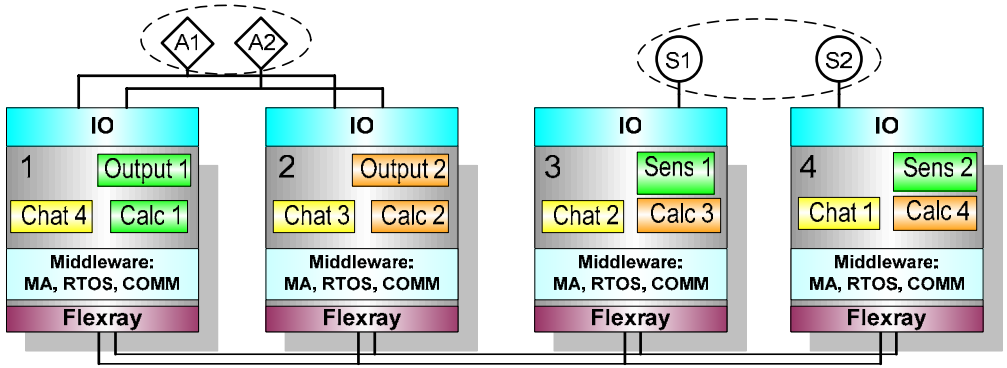


Design decisions



- Simple schedule for demonstration and implementation purposes
- Schedule all messages and processes based on FlexRay cycle
- Prioritize ease of implementation before performance when choosing parameters (slot duration etc.)
- Use static segment for all applications to guarantee message arrival times.
- Provide dynamic segment and a few empty slots to allow for minor system changes

Our system



- Three simple applications, sensor, actuator and calculation
- Membership service for all entities
- Standby/active mode of calculation application based on membership information
- Fault tolerant behaviour to accept loss of 1-2 nodes and one communications channel

Thank you for your attention