A proved model for the hot standby – an overlay over two ClockSlaves

Rodrigo Ferreira Coelho, Siemens AG
Dragan Obradovic, Siemens AG
Günter Steindl, Siemens AG

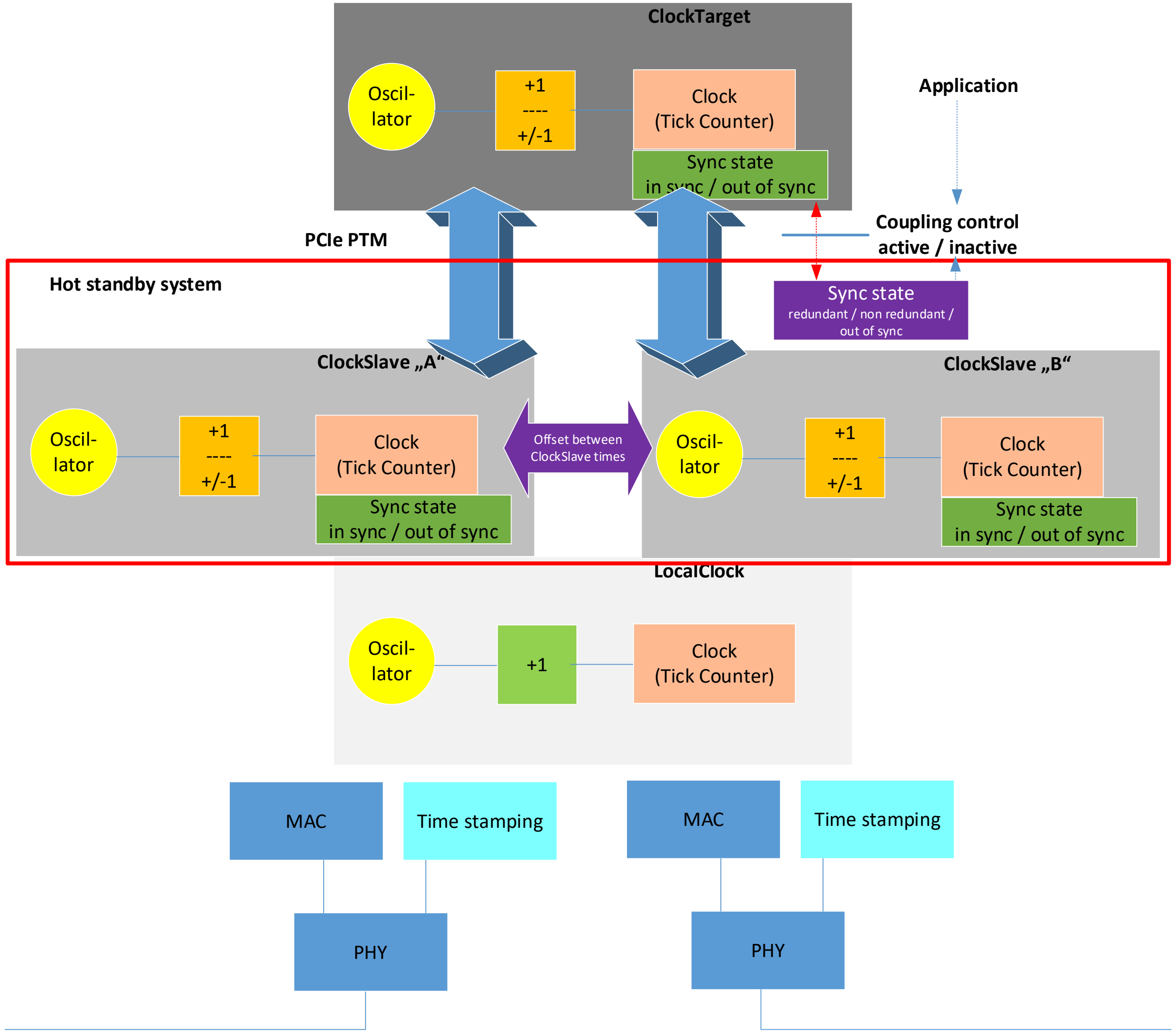
V01

Model for the hot standby
Log

V01: Initial version

Reference:
- 60802-steindl-Sync-Model-2211-v02.pdf

Implementation model
assumed for this presentation



The statement of this presentation are in general independent from this implementation model.

A hot standby system contains two independent controlled ClockSlaves which maintains their sync state.

The hot standby system itself maintains a sync state which is dependent on the two ClockSlave sync states.

Its sync state can be
- Redundant
- Non-redundant
- Out of sync

The next slide contains a truth table combining the different states.

Model for the hot standby Sync states

ClockSlave A Sync state	ClockSlave B Sync state	Offset between A and B are below defined treshold	Hot standby system Sync state[2)]	Coupling control State	ClockTarget Sync state[3)]
Out of sync	Out of sync	---	Out of sync	Inactive	Out of sync
In sync	Out of sync	---	Non redundant	Inactive	Out of sync
In sync	Out of sync	---	Non redundant	Active	In sync
In sync	In sync	Yes	Redundant	Inactive	Out of sync
In sync	In sync	Yes	Redundant	Active	In sync
In sync	In sync	No[1)]	Non redundant	Inactive	Out of sync
In sync	In sync	No[1)]	Non redundant	Active	In sync

[1)] In this case, the ClockSlave with the attribute Primary is assumed to be correct.
[2) and3)] Table should be read „left to right“. The hot standby system sync state and the ClockTarget state are the result of the columns to their left.

See 60802-steindl-Sync-Model-2211-v02.pdf for ClockSlave sync states

Model for the hot standby
Splitting and combining of machines

A ClockTarget either used both ClockSlave times (in state redundant) or the Primary ClockSlave time while the coupling control is active.

A ClockTarget stays with the chosen ClockSlave / Grandmaster if the other Grandmaster is lost / the other ClockSlave is out of sync. It uses the Primary only, if hot standby is again in state redundant.

Thus, splitting a machine into two machines (one active Grandmaster per machine) and combining the two machines to one machine (two active Grandmasters per machine) is working.

The offset between the two Grandmasters in case of combining the two machines is “removed” by power-on/-off one of the two Grandmasters.
While the second Grandmaster is gone, all ClockSlaves bound to this Grandmaster are “out of sync” and the coupling control is in state inactive, thus the corresponding ClockTarget sync state is “out of sync”, too.

One of the two ClockSlaves enter state “in sync” and the coupling control can thus switch to state active and thus, move the ClockTarget into sync state “in sync”.

The second Grandmaster would first synchronize to the first Grandmaster and then start transmitting time.

Model for the hot standby
Conclusion

The hot standby system switches freely between the different states. Its states are the outcome of the combination of the sync states of the ClockSlaves and the offset between the ClockSlave times.

The attributes added by the hot standby system are

- Defining the relation between the two ClockSlaves
- Defining a hot standby system state
- Defining the behavior depending on the sync states and the offset between the ClockSlave times

Any locking or prohibiting of state changes of the hot standby system doesn’t fit to the industrial requirements.

Conclusion

Hot standby – P802.1ASdm – needs to be updated with the next draft.

This update is needed to cover the requirements for industrial automation!