Simulation of the TMU

1. Disclaimer

2. General

This contribution includes a MATLAB simulation of the timing synchronization unit (TMU) part of the USB 4.0 standard. The TMU spec includes mathematical equations and variable definitions which are not trivial to understand. The model

implements all these variables and equations in a high-level programming language (MATLAB) and enables a newcomer to the standard to:

1. Look at the code and see if his understanding of the variables and equations is correct.
2. Run the code, tune various parameters (filters, time windows, etc.) and see their system-level effect on convergence time, accuracy, etc. The code enables simulation of very long time periods (e.g. 10's of seconds) which are impractical to simulate with hardware description language (HDL) tools.
3. Take real-life inputs (taken from a real device or from a HDL Silicon simulation), run the simulation model using these inputs, and compare the model outputs to the actual outputs (from the real device or the HDL simulation) for debugging purposes.

3. Simulation description

Only the TMU part of the standard is simulated. The basic event of the TMU is the handshake between two routers, so the simulation works in an event-driven mode where actions are taken only at handshakes.
The simulated network has a specific topology, as described in Figure 1. However, the code is flexible and can be easily adapted to any other topology by using generic functions which are described in comments within the code.
The simulation starts by generating all the handshakes. Since the TMU works in a feed-forward manner, there is no feedback from TMU calculations to the timing accumulators of the routers. Therefore, the t1, t2, t3, t4 values of a handshake between 2 routers do not depend on past handshake results, and all the handshakes can be generated before the TMU calculations are performed. This enables the simulation to work in a vectorized manner and significantly accelerates its run time (several Seconds of TMU operations can be simulated during a few minutes on an average computer).
Each router is assumed to have an accumulator which holds its local time, as defined by the TMU spec. For every handshake, these accumulators are updated according to the assumed local clock of each router.
After generating the handshake results, The handshakes between different pairs of routers are first aligned to each other such that each handshake is aligned with the latest available handshakes of the relevant router pairs that should be used for the TMU calculations.
After aligning the handshake results between all relevant pairs of routers, the TMU calculations can be performed in a vectorized manner.

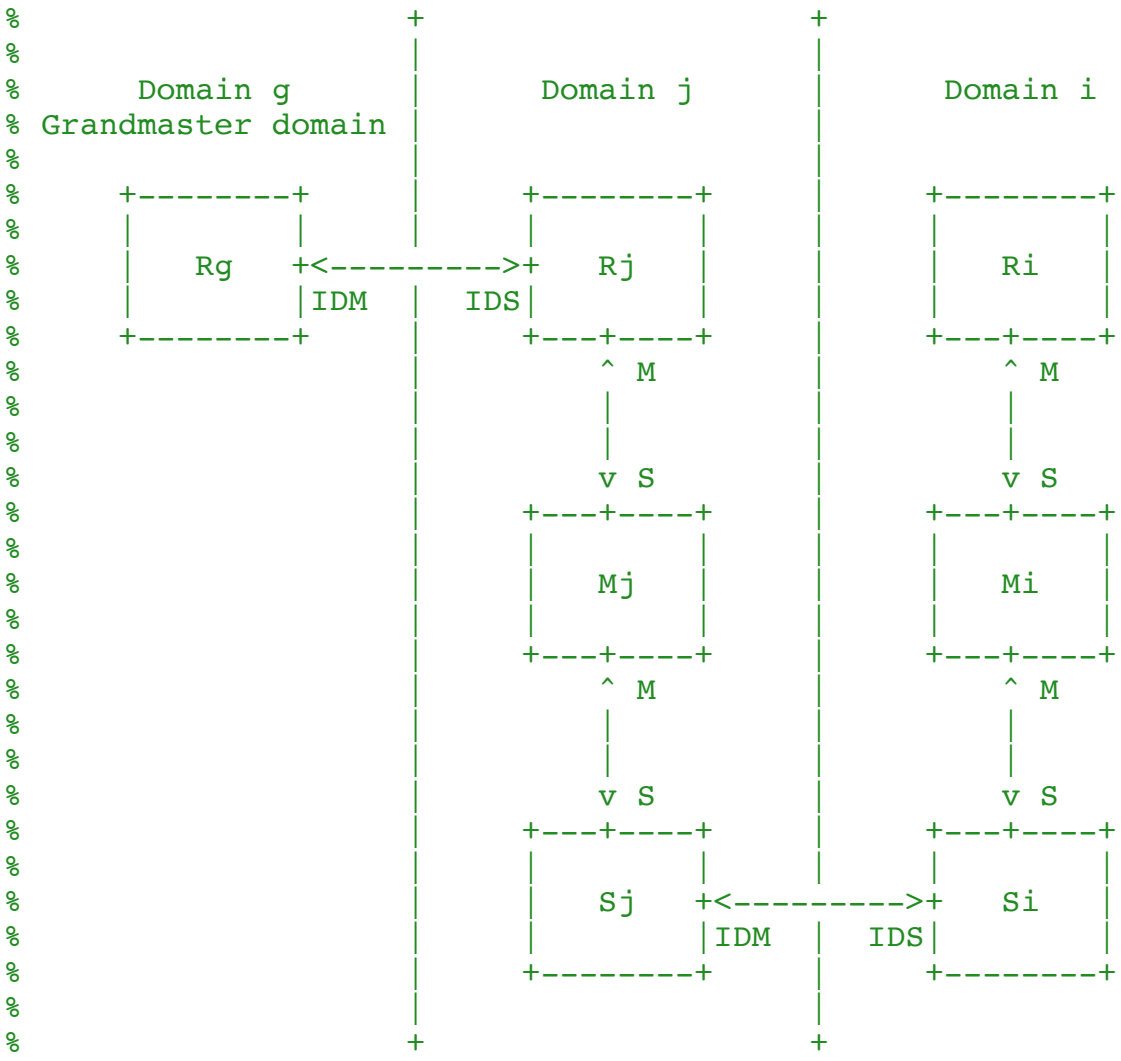

Figure 1 – network topology

4. Description of the main files

- Tmu.m – Top simulation file, where the network topology is defined
- Init_params.m – Definition of all router parameters, including clocks, dppm offsets, filter coefficients, sync periods, etc.
- Time_sync_domain.m – TMU calculations for a domain with 3 routers (grandmaster, master & slave)
- Time_sync_pair.m - TMU calculations for a pair of routers (master & slave)

- Handshake.m - Creates t1, t2, t3, t4 values for all handshakes between a pair of routers
- IIR.m - An IIR low pass filter, as defined by the spec in equation 7.18, for smoothing an estimated parameter which is approximately constant over time (e.g. frequency offset, propagation delay)
- IIR_AC.m – An IIR low pass filter with filter attenuation compensation, as shown in Figure 7.17 in the spec, for smoothing an estimated parameter which changes linearly over time (e.g. time offset)
- Get_index_offset_for_t_vec_sync.m - Time vectors alignment tool. Aligns the handshake parameters of several router pairs such that the most recent handshakes of the other pairs are aligned with the handshake of the target pair.
- Calc_tidg.m - Implements equation 7-10 and 7-16 in the spec, for calculating the host (either inter domain or intra domain) local clock at given slave times.
- Quant.m – A fixed point quantization function.

5. Simulation Output

Any intermediate variable can be plotted and inspected, including frequency offsets, timing offsets, timing references, etc. Two example output plots are shown in Figures 2 and 3 below, for the TMU parameters which are set in the given init_params.m file. Figure 2 shows the time offset between a slave and a grandmaster of the same domain as calculated from Equation 7.9 of the TMU spec. Figure 3 shows the timing error between a slave of a domain and a grandmaster of another domain, as calculated from equation 7-17 of the TMU spec. It can be seen that the TMU converged after ~200mS and the steady-state timing error jitter is ~1nS peak to-peak.

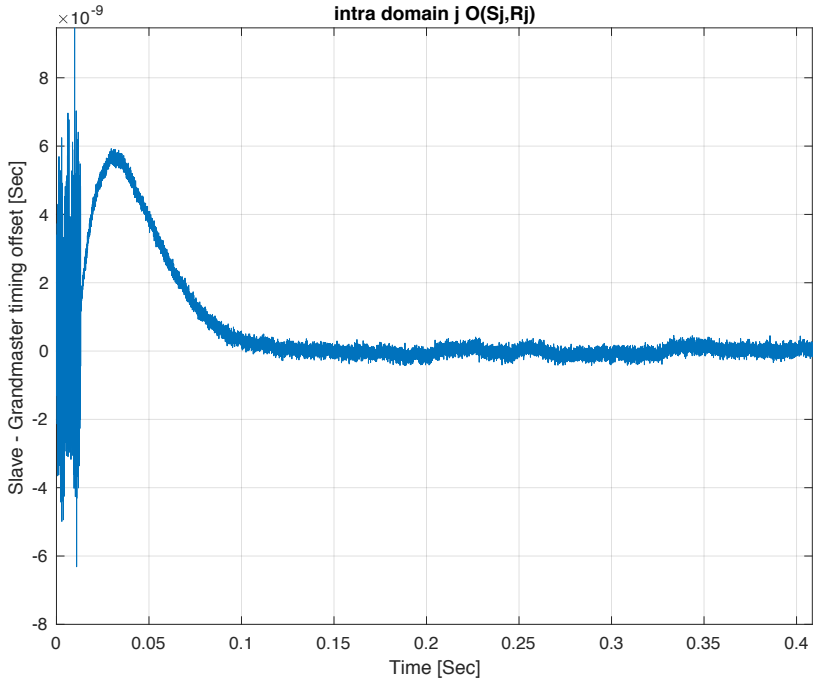

Figure 2 – Example simulation output – timing offset vs. time

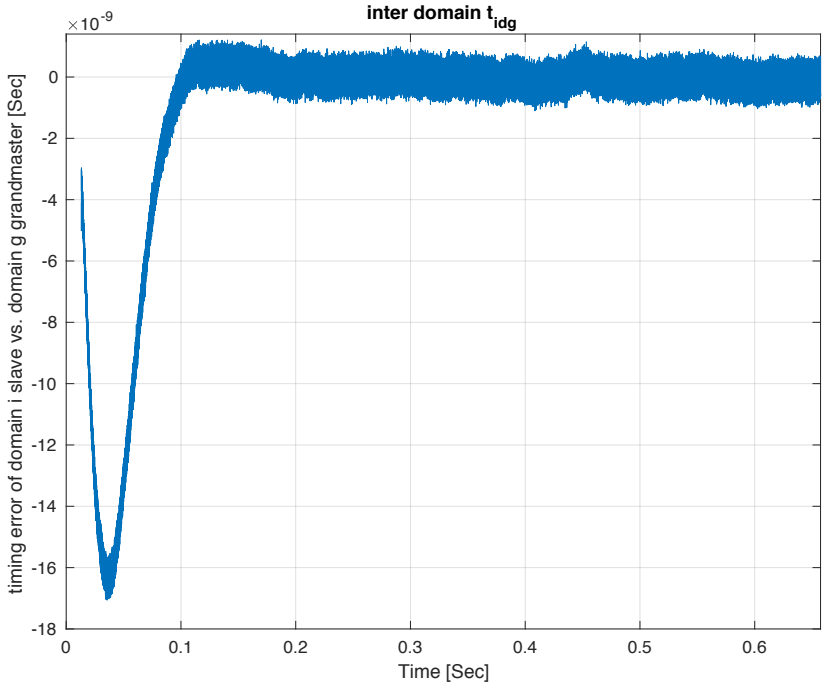

Figure 3 – Example simulation output – inter domain timing error vs. time