ST65 macro-memories

Already ordered memories are found in: /usr/local-eit/cad2/cmpstm/mem201*
If no memory of the desired size is available, new sizes can be ordered but delivery time is 1-2 weeks.
The memory types that available are:

- SPREG - Single Port Register File
- DPREG - Dual Port Register File
- SPHD - Single Port High Density SRAM
- SPHS - Single Port High Speed SRAM
- DPHD - Dual Port High Density SRAM
- DPHS - Dual Port High Speed SRAM

In general, Register Files have smaller area footprint than SRAM for small memory sizes. Dual port memories are much bigger than Single port. High Density are smaller than High Speed. Mux options varies between different sizes and depending on memory size different mux options results in different area footprint.

Memory files of interest

behaviour - contains files used for behavioral simulation:

- behaviour/verilog/Memory_date.v - behavorial model for Modelsim
- behaviour/verilog/Memory_date.verilog.map - sdf map file for correct sdf annotation CADENCE - contains files for Virtuoso & Soc Encounter (EDI)
- CADENCE/LEF/Memory_date_soc.lef - LEF used for Place & Route
- doc - documentation related to memory
- ugnC65_ST_MemoryTypeModels - User Manual, Release Notes, and Known Problems & Solutions
- PttV1200T025 - Data sheet with Timing diagrams for the specific corner

LEF - generic LEF - DO NOT USE!
libs - libs to use for Synthesis and Place & Route

- *.db - use together with Design Compiler
- *.lib - use with Cadence Tools, i.e., Soc Encounter & RC-compiler

SDF annotation

To correctly annotate and use the a ST65 memory, SDF should be generated in the following procedure:

Synthesis

1. Synthesize in Design Compiler using .db files in libs directory
2. Save hierarchical DDC (Design Compiler internal data format) "write_file -format ddc -hierarchy -output file.ddc"
3. Load DDC in PrimeTime "read_ddc file.ddc"
4. Write SDF using sdf-map file supplied by ST in the behaviour directory of the memory "write_sdf -map behaviour/verilog/memory_data.verilog.map -context Verilog -output file.sdf"
5. Use the "mapped" sdf file in Modelsim as usual, in case of null values use Max & Min instead of typical.

Place and Route

1. Place and route with .lib files in libs directory
2. Write SDF using sdf-map supplied by ST in the behaviour directory of the memory (Only works in version EDI 11 and onwards) "write_sdf -map_file behaviour/verilog/memory_data.verilog.map file.sdf"
3. Use the "mapped" sdf file in Modelsim as usual, in case of null values use Max & Min instead of typical.