Creating a Tessent Memory Library

To create a Tessent memory library (memlib) file for a 1024x32 SRAM, we need to define the memory's characteristics in a format compatible with Tessent's memory BIST (Built-In Self-Test) tools. A memlib file typically describes the memory's size, configuration, and parameters such as address width, data width, and access details.

Below is a sample memlib file for a 1024x32 SRAM (1024 words, 32-bit data width). I'll explain the key parameters and assumptions, followed by the file content.

Assumptions

- *Memory Type*: Single-port SRAM.

- *Address Width*: 1024 words require 10 address bits (2^10 = 1024).

- *Data Width*: 32 bits.

- *No Redundancy*: No repairable features (e.g., spare rows/columns) unless specified.

- *Default Timing*: Generic timing parameters (modify as needed for your process).

- *Tessent Format*: The file follows the Tessent memlib syntax for memory description.

Key Parameters

- *Words*: 1024 (number of memory locations).

- *Bits*: 32 (data width per word).

- *Address Bits*: 10 (log2(1024)).

- *Ports*: Single read/write port.

- *Operation*: Synchronous SRAM with read and write capabilities.

Sample Tessent Memlib File

plaintext

```
// Tessent Memory Library File for 1024x32 SRAM

// Generated on April 24, 2025


MEMORY SRAM_1024x32 {

  TYPE = SRAM;

  DESCRIPTION = "Single-port SRAM, 1024 words, 32-bit data width";


  SIZE {
```

```
  WORDS = 1024;
  BITS = 32;
  ADDRESS_BITS = 10;
}

PORTS {
  PORT rw {
    TYPE = READ_WRITE;
    PINS {
      ADDRESS = A[9:0];
      DATA_IN = D[31:0];
      DATA_OUT = Q[31:0];
      WRITE_ENABLE = WE;
      CHIP_ENABLE = CE;
      CLOCK = CLK;
    }
  }
}

OPERATION {
  READ {
    ENABLE = (CE == 1, WE == 0, CLK == rising);
    OUTPUT = Q[31:0];
  }
  WRITE {
    ENABLE = (CE == 1, WE == 1, CLK == rising);
    INPUT = D[31:0];
  }
}

TIMING {
    CLOCK_PERIOD = 5.0ns; // Example: 200 MHz clock
    SETUP_ADDRESS = 0.5ns;
    HOLD_ADDRESS = 0.2ns;
    SETUP_DATA = 0.5ns;
    HOLD_DATA = 0.2ns;
    SETUP_CONTROL = 0.5ns;
    HOLD_CONTROL = 0.2ns;
    ACCESS_TIME = 2.0ns; // Time from clock edge to valid Q
  }

  BIST {
    ALGORITHM = MARCH_C; // Example: Standard March C algorithm
    DIAGNOSTIC = FALSE;
    REDUNDANCY = NONE;
  }
}
```

Explanation of Sections

1. *MEMORY Block*:
   - Defines the memory name (SRAM_1024x32) and type (SRAM).
   - Includes a description for clarity.

2. *SIZE Block*:
   - Specifies the memory dimensions: 1024 words, 32 bits per word, 10 address bits.

3. *PORTS Block*:
   - Defines a single read/write port (rw) with:
     - Address pins (A[9:0]) for 10-bit addressing.
     - Data input (D[31:0]) and output (Q[31:0]) for 32-bit data.
     - Control signals: Write Enable (WE), Chip Enable (CE), and Clock (CLK).

4. *OPERATION Block*:

   - Describes read and write operations:

     - *Read*: Enabled when CE = 1, WE = 0, on rising clock edge.

     - *Write*: Enabled when CE = 1, WE = 1, on rising clock edge.

5. *TIMING Block*:

   - Provides example timing parameters (modify based on your SRAM datasheet):

     - Clock period: 5 ns (200 MHz).

     - Setup/hold times for address, data, and control signals.

     - Access time for read operation.

6. *BIST Block*:

   - Specifies the BIST algorithm (e.g., March C, a common memory test algorithm).

   - Indicates no redundancy or diagnostic mode (adjust if your SRAM supports repair).

Notes

- *Customization*: Adjust the timing parameters, BIST algorithm, or redundancy settings based on your SRAM's specifications or process technology.

- *Verification*: Validate the memlib file with Tessent tools (e.g., Tessent Shell) to ensure compatibility.

- *Additional Features*: If your SRAM has special features (e.g., multiple ports, bit-write masks, or power-down modes), add corresponding sections to the file.

- *File Extension*: Save the file with a .memlib extension (e.g., sram_1024x32.memlib).