PROJECT LEARN IDE

saved a few seconds ago

EDITOR NAV-TLV LOG

DIAGRAM VIZ WAVEFORM

```
\m4_TLV_version 1d: tl-x.org
\SV
   m4_include_lib(['https://raw.githubusercontent.com/BalaDhinesh/RISC-V_MYTH_Workshop/master/tlv_lib/risc-v_shell_
\SV
   m4_makerchip_module   // (Expanded in Nav-TLV pane.)
\TLV
   // TL-Verilog Code for RISC-V RV32I CPU Architecture with all instruction set (Using Load and Store through Jump
   // Complete RISC-V RV32I CUP Design using TL-Verilog Abstraction.....
   // /====================\
   // | Sum 1 to 9 Program |
   // \====================/

   // External to function:
   m4_asm(ADD, r10, r0, r0)             // Initialize r10 (a0) to 0.
   // Function:
   m4_asm(ADD, r14, r10, r0)            // Initialize sum register a4 with 0x0
   m4_asm(ADDI, r12, r10, 1010)         // Store count of 10 in register a2.
   m4_asm(ADD, r13, r10, r0)            // Initialize intermediate sum register a3 with 0
   // Loop:
   m4_asm(ADD, r14, r13, r14)           // Incremental addition
   m4_asm(ADDI, r13, r13, 1)            // Increment intermediate register by 1
   m4_asm(BLT, r13, r12, 1111111111000) // If a3 is less than a2, branch to label named <loop>
   m4_asm(ADD, r10, r14, r0)            // Store final result to register a0 so that it can be read by main progra
   m4_asm(SW, r0, r10, 100)
   m4_asm(LW, r15, r0, 100)

   // Optional:
   // m4_asm(JAL, r7, 00000000000000000000) // Done. Jump to itself (infinite loop). (Up to 20-bit signed immediat
   m4_define_hier(['M4_IMEM'], M4_NUM_INSTRS)

   |cpu
      @0
         $reset = *reset;
      // YOUR CODE HERE
      // Stimulus for RISC-V CPU Architecture.....
         $pc[31:0] = (>>1$reset) ? 32'b0 :
                     (>>3$is_load) ? (>>3$int_pc) :
                     (>>3$is_jal) ? (>>3$br_tgt_pc) :
                     (>>3$is_jalr) ? (>>3$jalr_tgt_pc) :
                     (>>3$taken_br) ? (>>3$br_tgt_pc) :  (>>1$int_pc)  :
```