SoC Design and Practice

Jungrae Kim (dale40@skku.edu)
Semiconductor Systems Engineering

Lab 7: Coverage

Recap

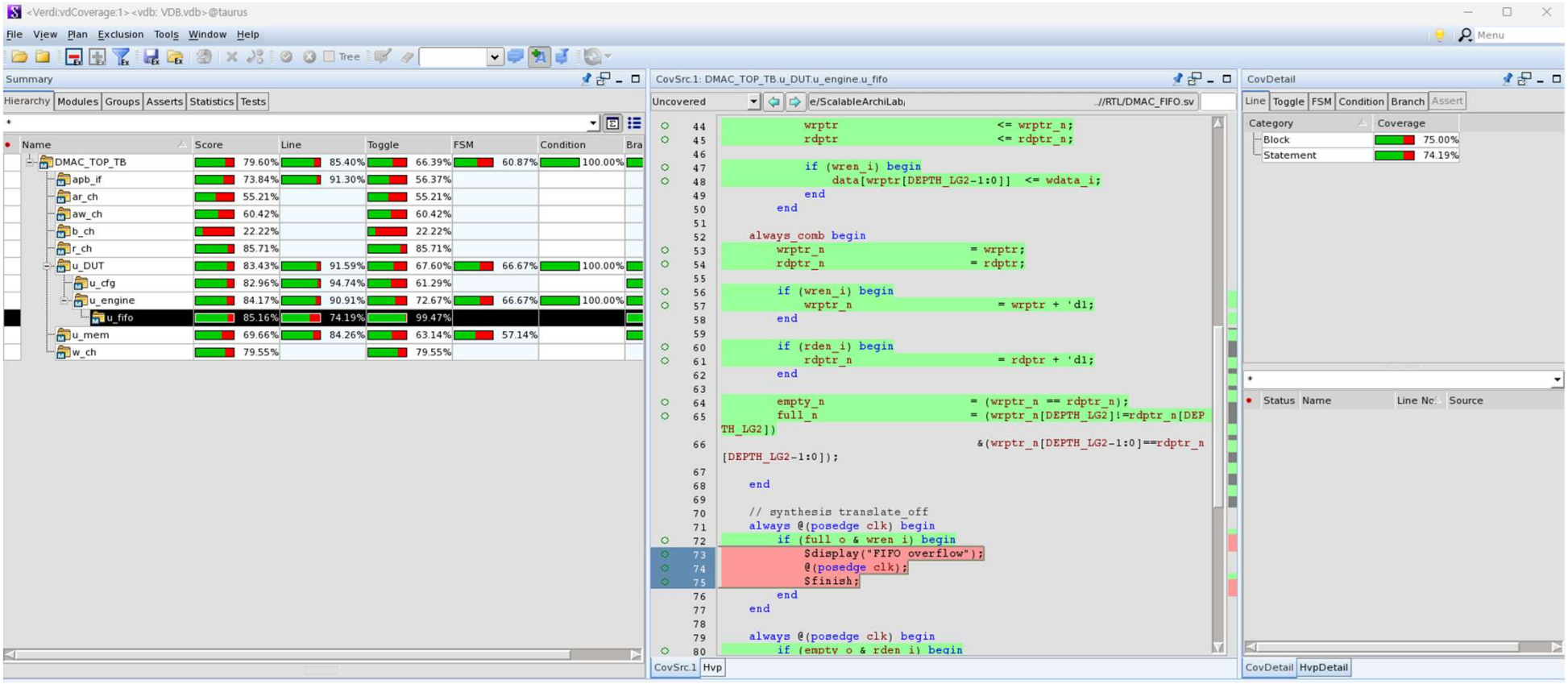
- Verification
  - Directed vs. constrained random
  - Coverage-oriented
- Coverage
  - Code coverage
  - Functional coverage

Outline

- Code coverage measurement
- Functional coverage measurement

Code Coverage

Code Coverage in Verdi



Login and git update

- `$ git clone https://github.com/2025-Spring-SoC/lab5.git`
- `$ cd lab5/DMAC`

Simulator Options for Code Coverage

- VCS (Synopsys simulator)
  - vcs -cm line+cond+fsm+tgl+branch …
    - Included in run.compile4cov and run.sim4cov
    - You can simply run `run.compile4cov` and `run.sim4cov`

- NCVerilog (Cadence simulator)
  - Ncverilog –coverage <coverage_types> …
    - Coverage type
      - B (Block)
      - E (Expression)
      - F (FSM)
      - T (Toggle)
      - U (Functional)
      - A(All)

After Compilation and Simulation

- We have a new directory containing coverage database
  - (default) simv.vdb
  - (in our environment) VDB.vdb

- Reading the coverage database
  - 1) Create a HTML-based report using Synopsys urg

    ```
    $ urg -dir VDB.vdb
    ```

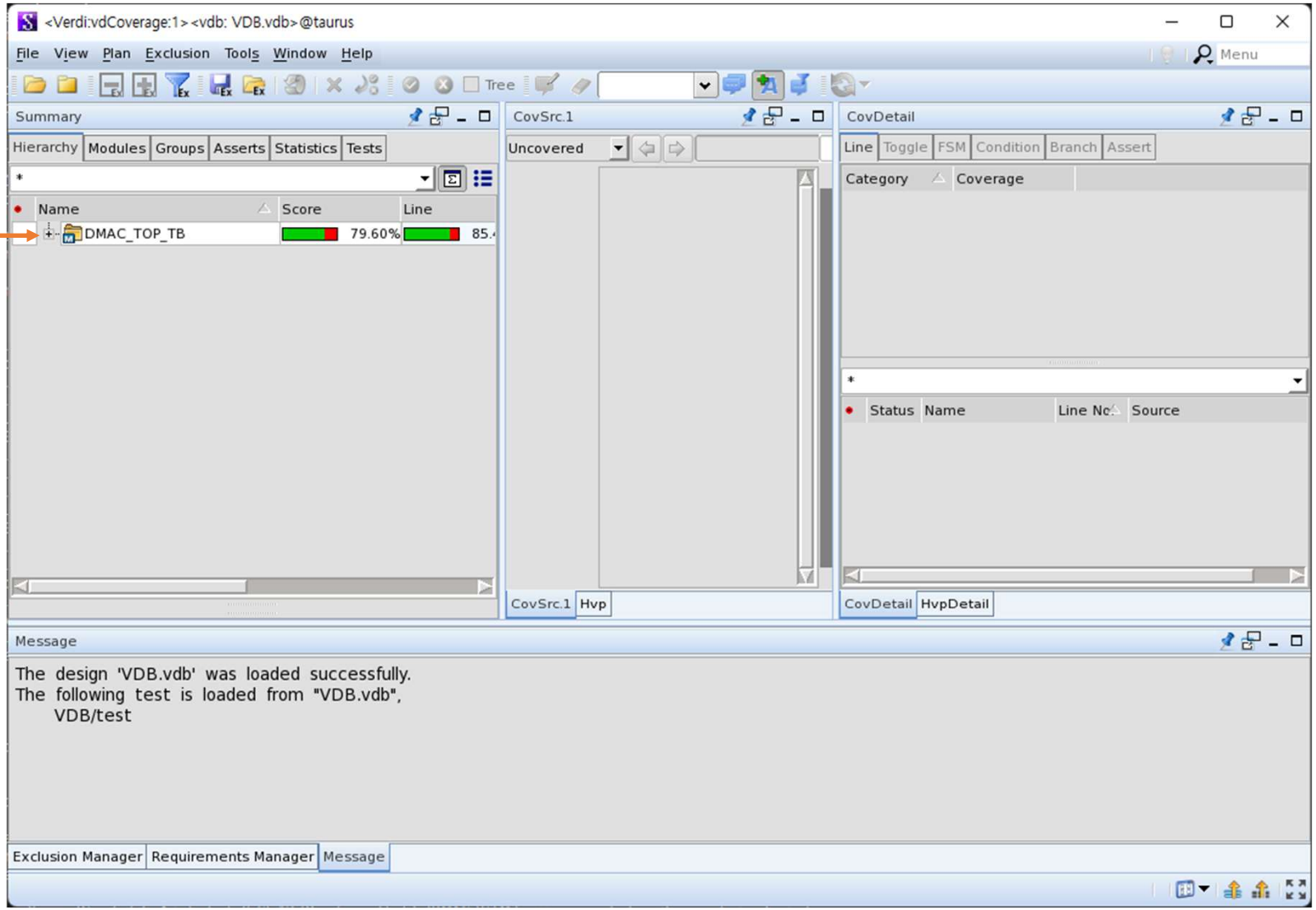
  - 2) Launch Verdi

    ```
    $ verdi -cov -covdir VDB.vdb
    ```

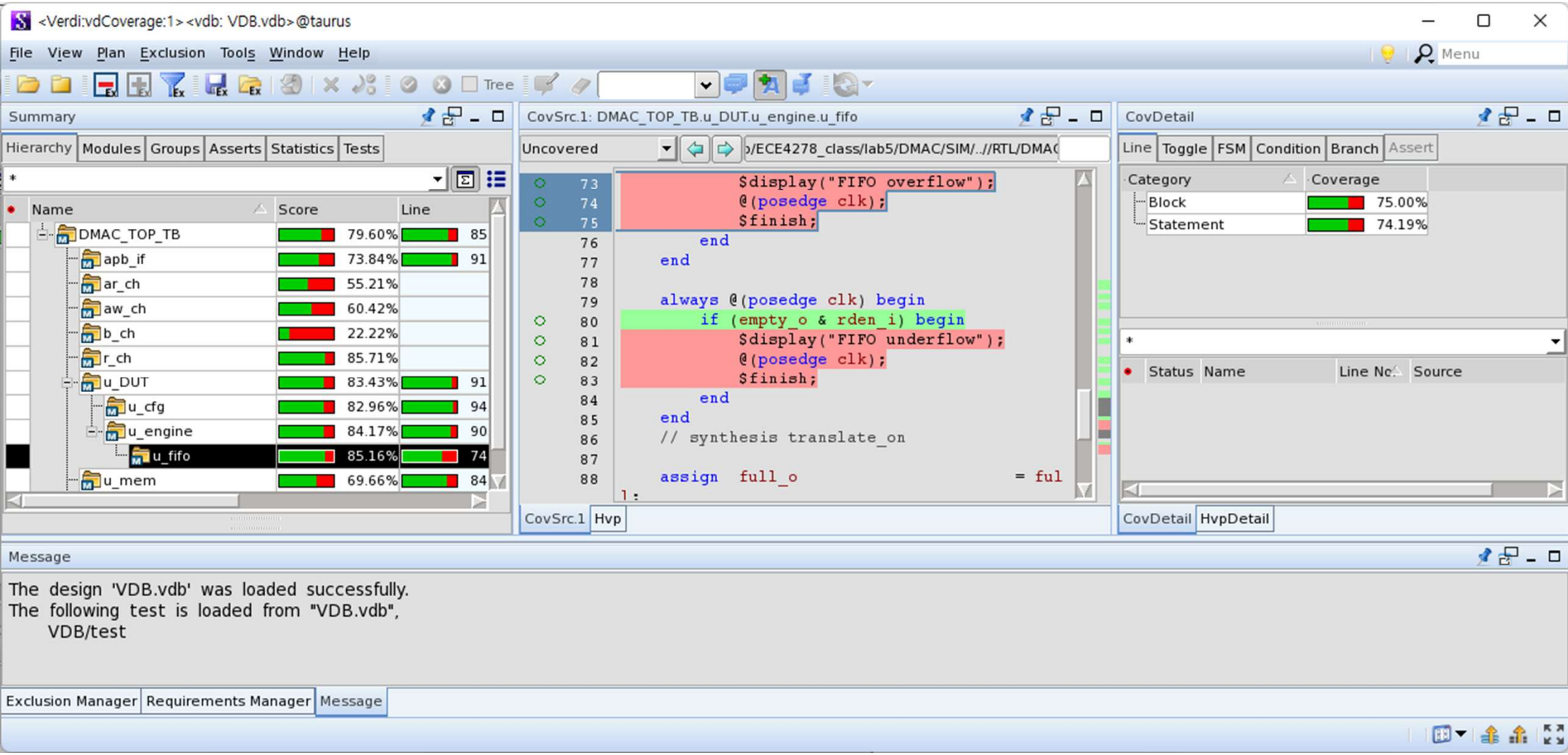
    Or simply type `./run.verdi4cov`

Code Coverage in Verdi

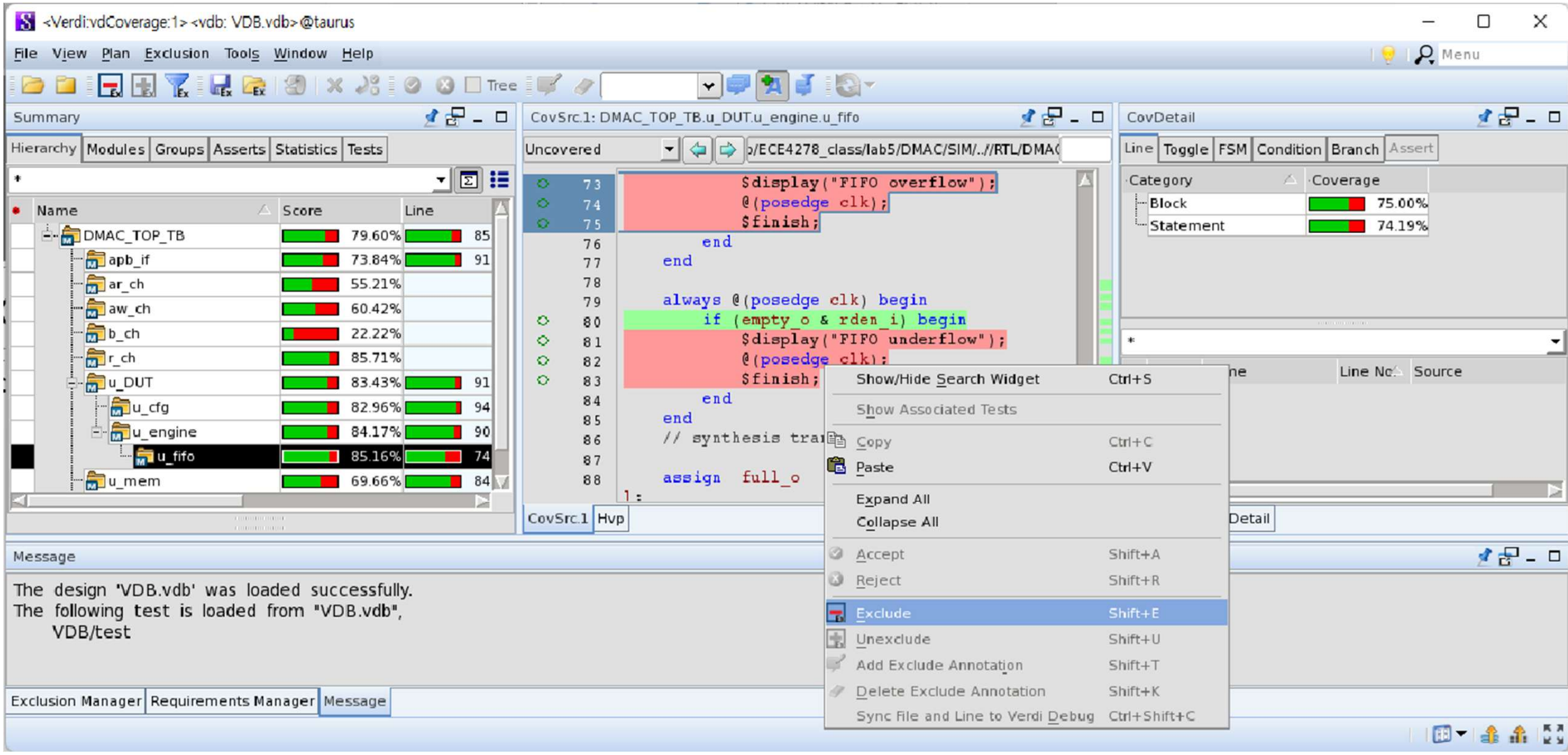
Click + to expand



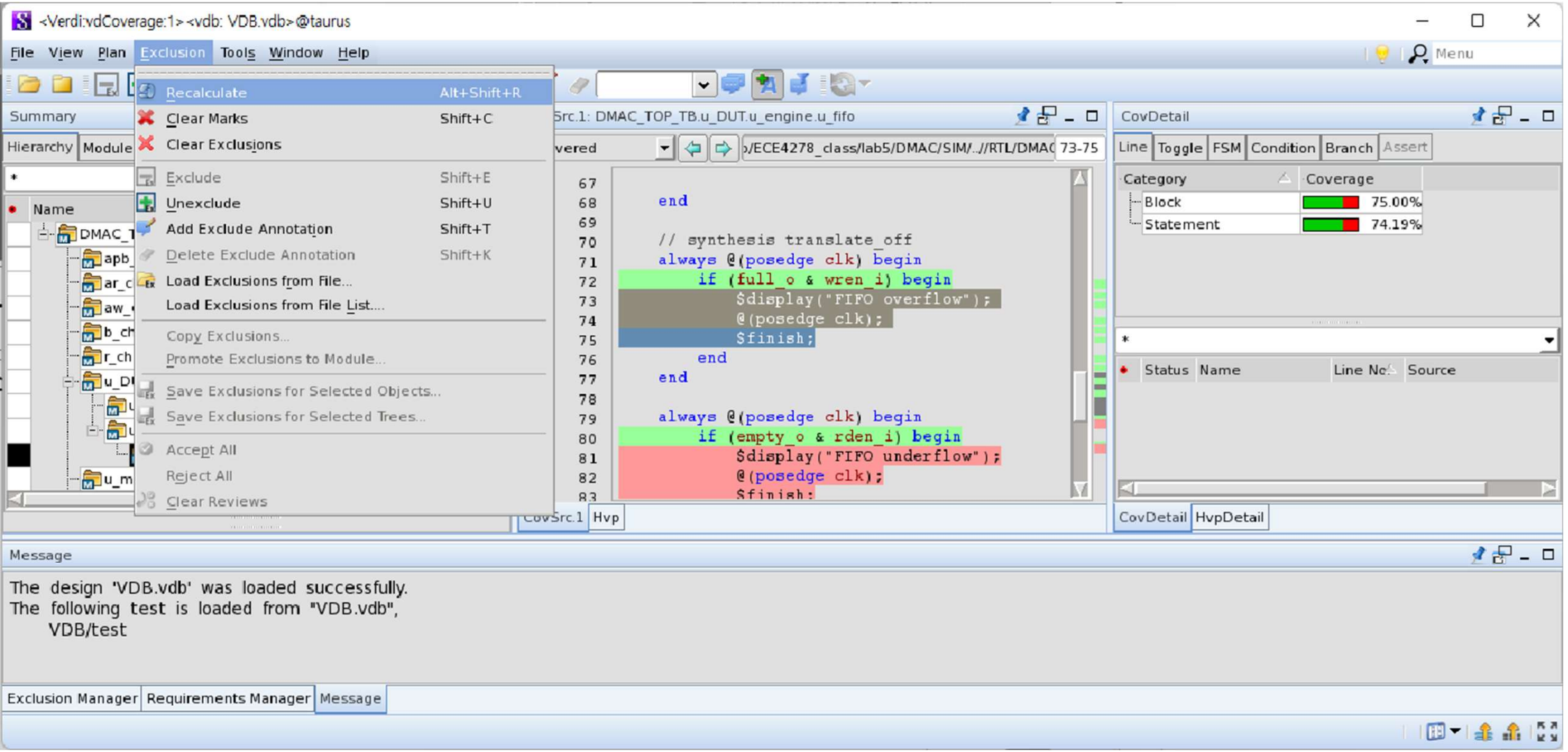
Code Coverage in DMAC_FIFO



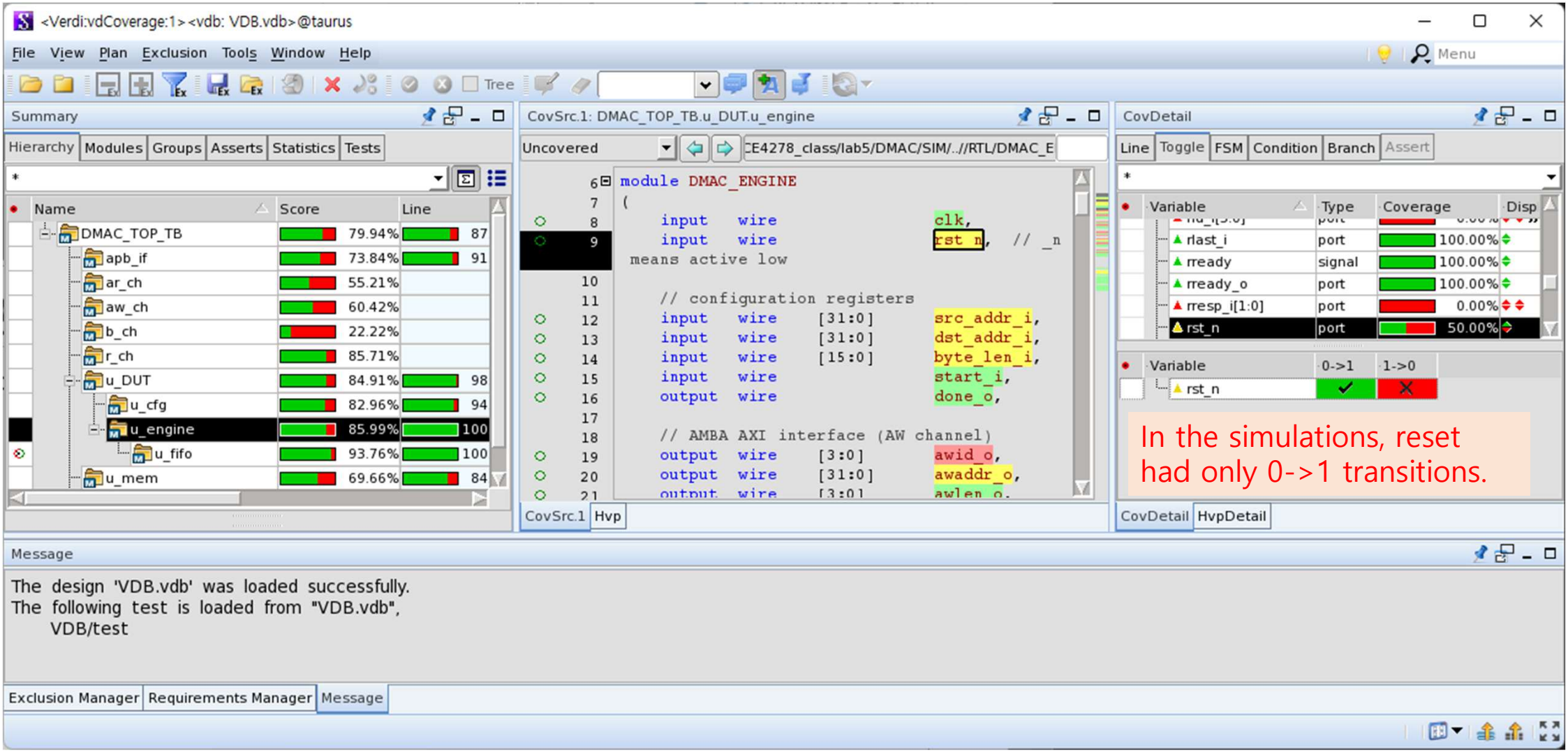
Excluding Codes from Calculation



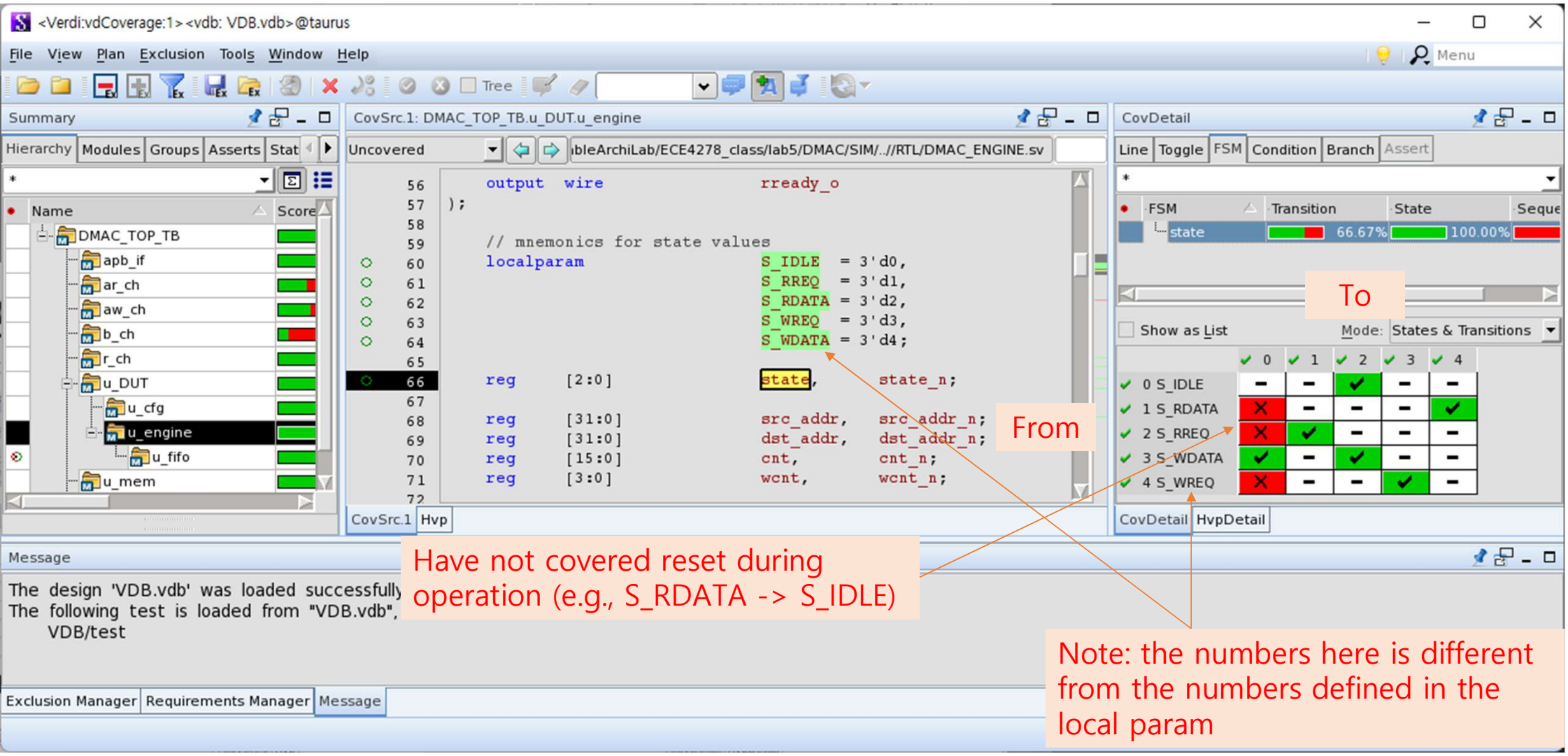
Recalculate after Exclusion



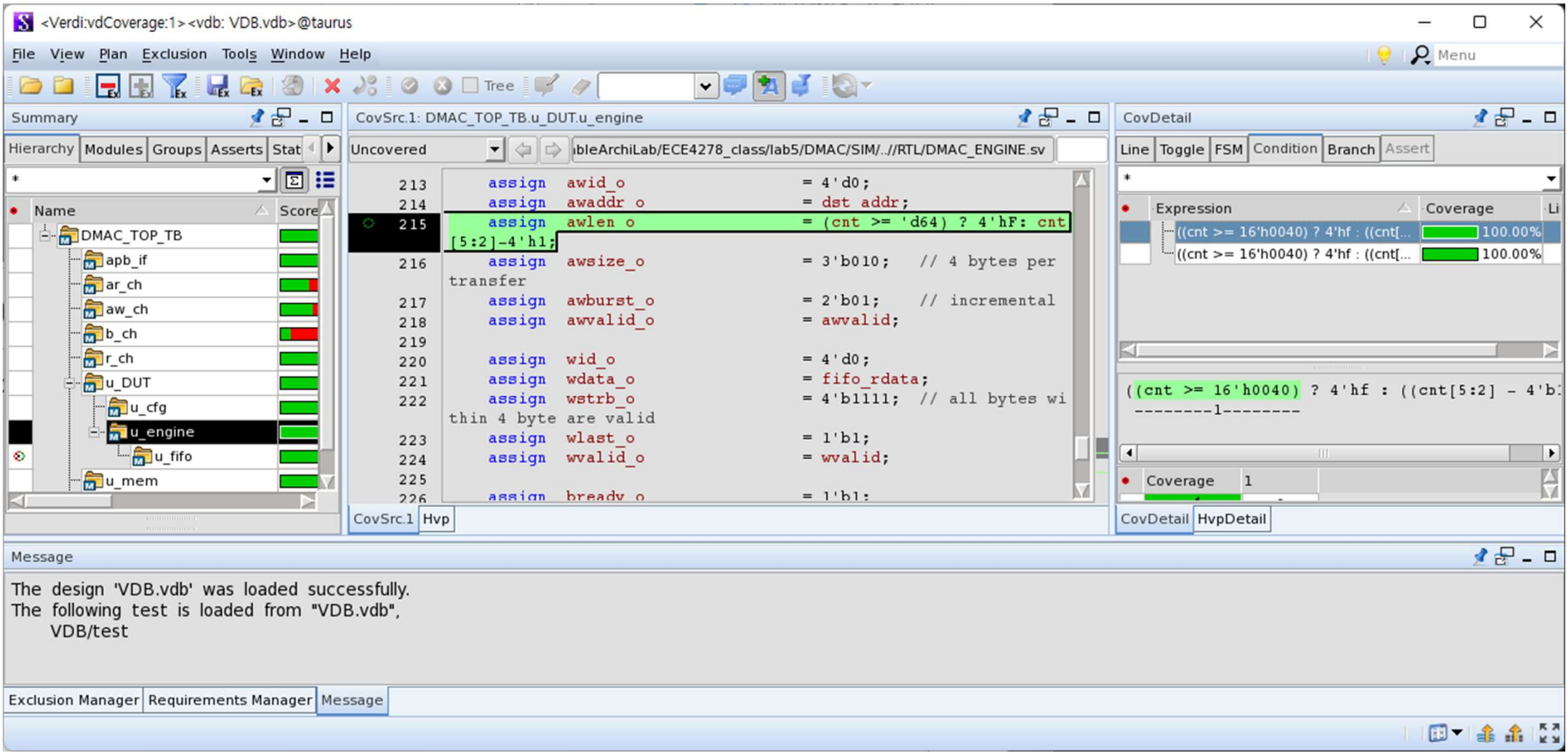
Toggle Coverage



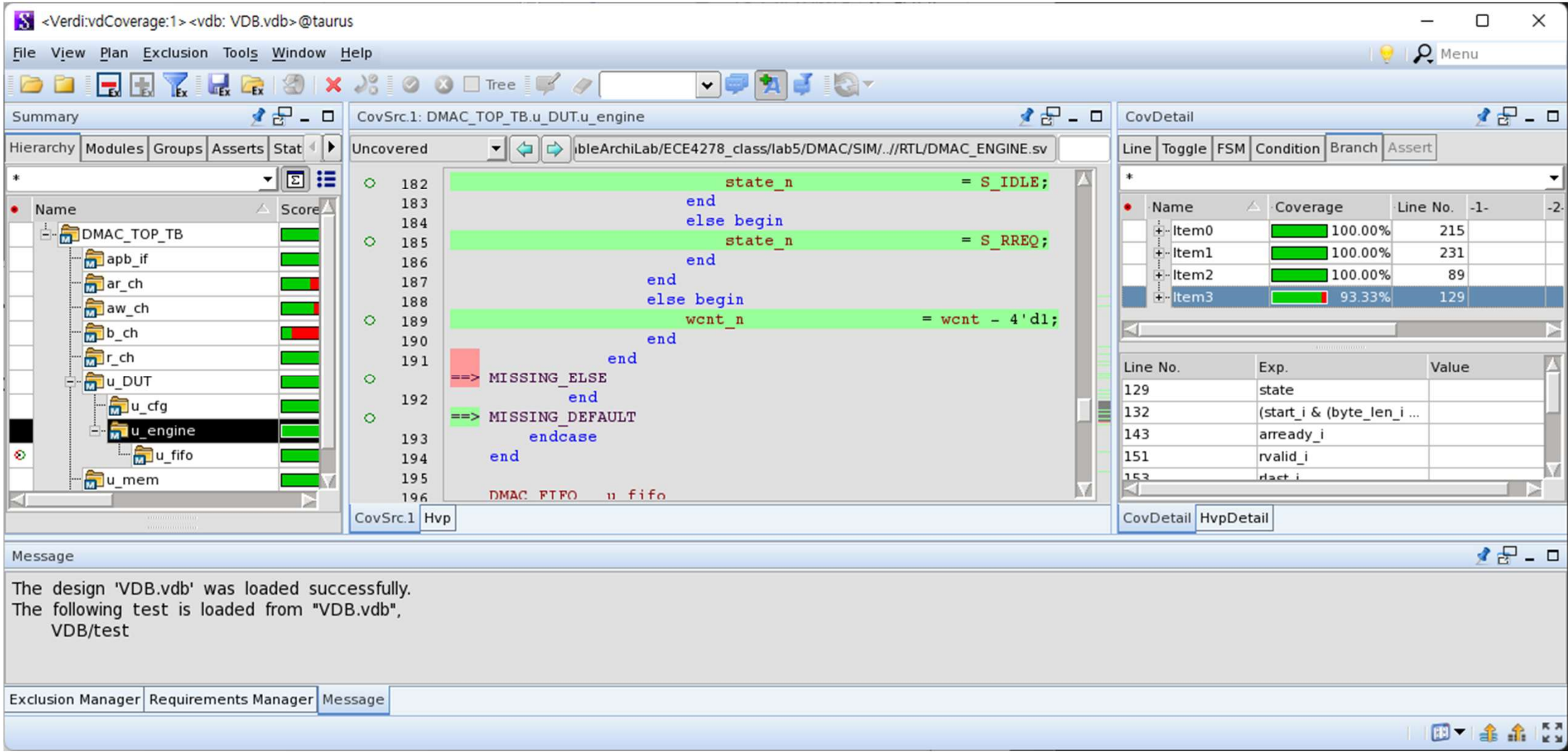
FSM Coverage



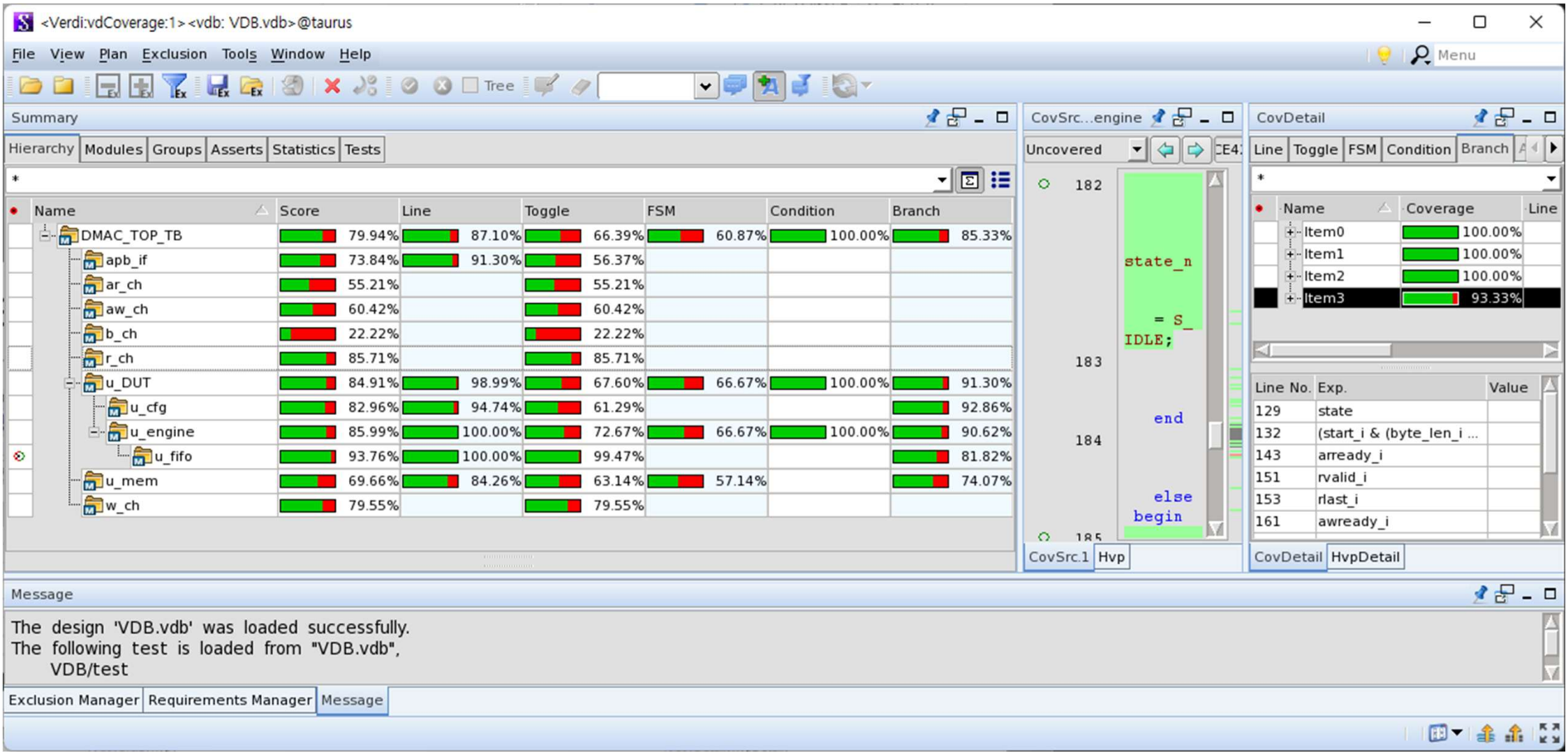
Condition Coverage



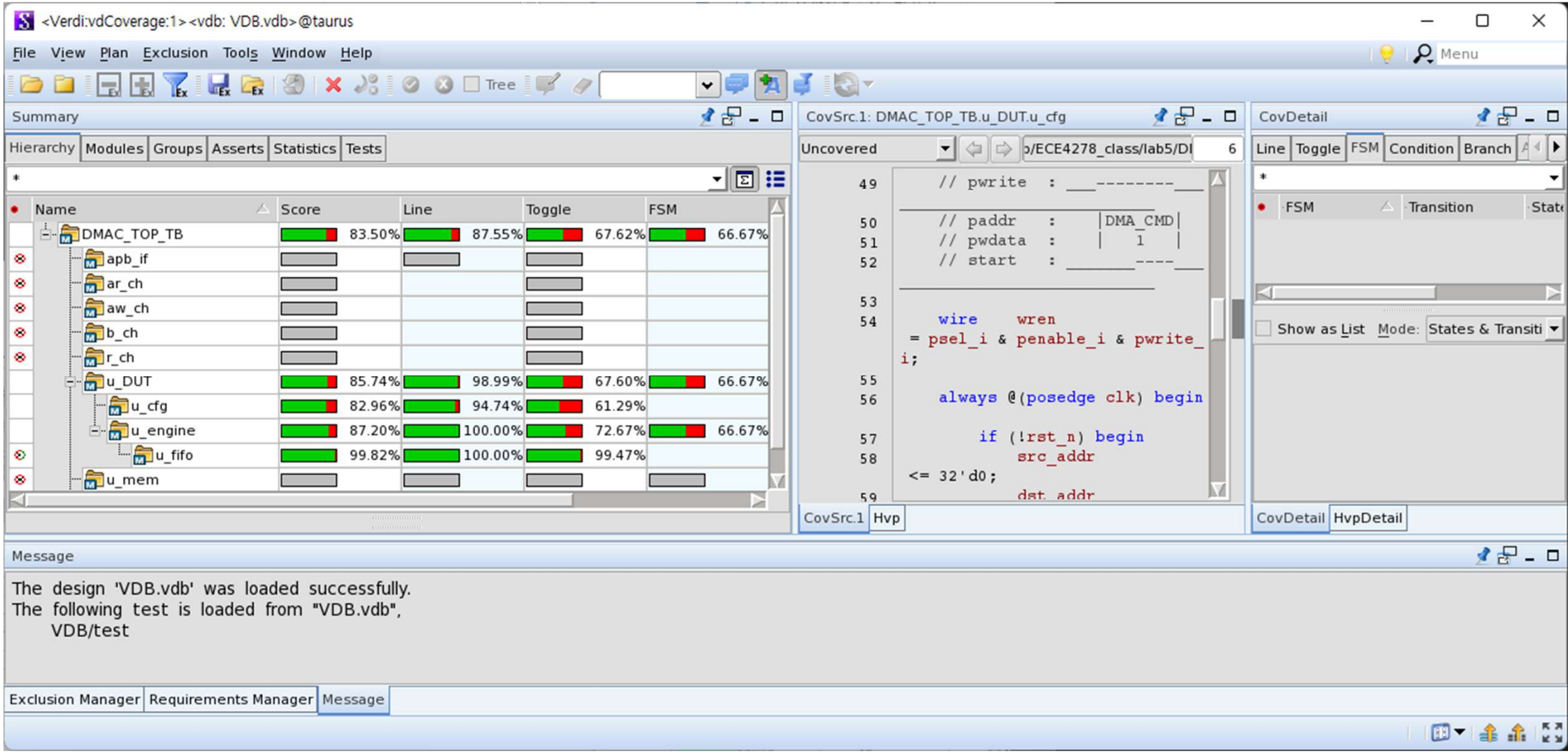
Branch Coverage



Putting Them Together



After Excluding TB and Assertions



Functional Coverage

Function Coverage Items

- You need to manually create

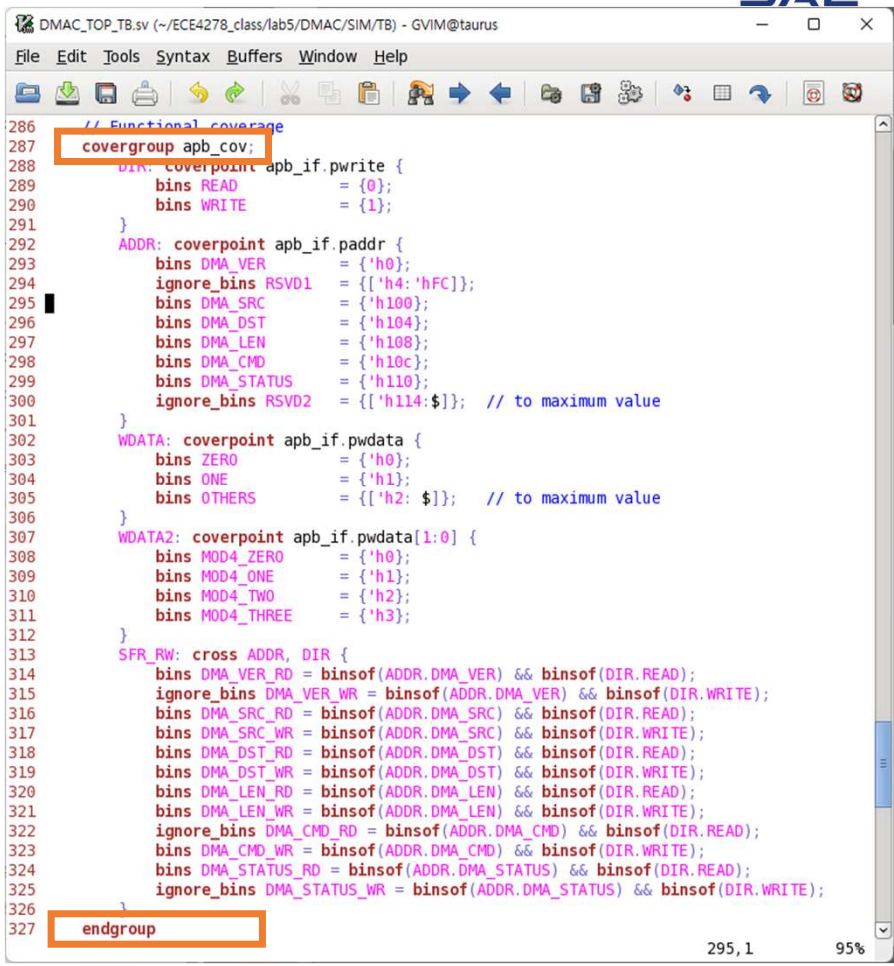
DMAC_TOP_TB.sv (~/ECE4278_class/lab5/DMAC/SIM/TB) - GVIM@taurus

File Edit Tools Syntax Buffers Window Help

```
286     // Functional coverage
287     covergroup apb_cov;
288         DIR: coverpoint apb_if.pwrite {
289             bins READ           = {0};
290             bins WRITE          = {1};
291         }
292         ADDR: coverpoint apb_if.paddr {
293             bins DMA_VER        = {'h0};
294             ignore_bins RSVD1   = {['h4:'hFC]};
295             bins DMA_SRC        = {'h100};
296             bins DMA_DST        = {'h104};
297             bins DMA_LEN        = {'h108};
298             bins DMA_CMD        = {'h10c};
299             bins DMA_STATUS     = {'h110};
300             ignore_bins RSVD2   = {['h114:$]};  // to maximum value
301         }
302         WDATA: coverpoint apb_if.pwdata {
303             bins ZERO           = {'h0};
304             bins ONE            = {'h1};
305             bins OTHERS         = {['h2: $]};   // to maximum value
306         }
307         WDATA2: coverpoint apb_if.pwdata[1:0] {
308             bins MOD4_ZERO      = {'h0};
309             bins MOD4_ONE       = {'h1};
310             bins MOD4_TWO       = {'h2};
311             bins MOD4_THREE     = {'h3};
312         }
313         SFR_RW: cross ADDR, DIR {
314             bins DMA_VER_RD = binsof(ADDR.DMA_VER) && binsof(DIR.READ);
315             ignore_bins DMA_VER_WR = binsof(ADDR.DMA_VER) && binsof(DIR.WRITE);
316             bins DMA_SRC_RD = binsof(ADDR.DMA_SRC) && binsof(DIR.READ);
317             bins DMA_SRC_WR = binsof(ADDR.DMA_SRC) && binsof(DIR.WRITE);
318             bins DMA_DST_RD = binsof(ADDR.DMA_DST) && binsof(DIR.READ);
319             bins DMA_DST_WR = binsof(ADDR.DMA_DST) && binsof(DIR.WRITE);
320             bins DMA_LEN_RD = binsof(ADDR.DMA_LEN) && binsof(DIR.READ);
321             bins DMA_LEN_WR = binsof(ADDR.DMA_LEN) && binsof(DIR.WRITE);
322             ignore_bins DMA_CMD_RD = binsof(ADDR.DMA_CMD) && binsof(DIR.READ);
323             bins DMA_CMD_WR = binsof(ADDR.DMA_CMD) && binsof(DIR.WRITE);
324             bins DMA_STATUS_RD = binsof(ADDR.DMA_STATUS) && binsof(DIR.READ);
325             ignore_bins DMA_STATUS_WR = binsof(ADDR.DMA_STATUS) && binsof(DIR.WRITE);
326         }
327     endgroup
                                                        295,1         95%
```

Covergroup

- SystemVerilog user-defined type
- Usually contains
  - When to sample
  - What to sample
    - A set of cover points
    - Cross coverage between cover points
  - Other options



When to Sample

```
// Functional coverage
covergroup apb_cov @(posedge clk);  // sample at posedge
    // more here
endgroup

apb_cov apb_cov_inst        = new();
```

OR

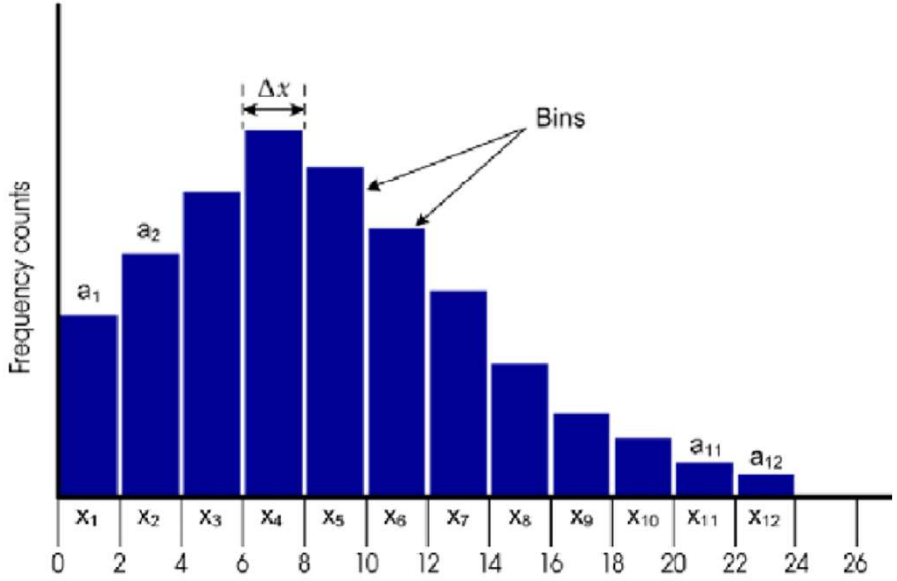
```
// Functional coverage
covergroup apb_cov;
    // more here
endgroup

apb_cov apb_cov_inst        = new();

always @(posedge clk)       // specify when to sample
    if (rst_n & apb_if.psel & apb_if.penable)
        apb_cov_inst.sample();
```

What to Sample: Coverpoint

- Specify which variable to be sampled
  - Can have a label (with ':')
  - Can have bins



DMAC_TOP_TB.sv (~/ECE4278_class/lab5/DMAC/SIM/TB) - GVIM@taurus

File Edit Tools Syntax Buffers Window Help

```
286     // Functional coverage
287     covergroup apb_cov;
288         DIR: coverpoint apb_if.pwrite {
289             bins READ           = {0};
290             bins WRITE          = {1};
291         }
292         ADDR: coverpoint apb_if.paddr {
293             bins DMA_VER        = {'h0};
294             ignore_bins RSVD1   = {['h4:'hFC]};
295             bins DMA_SRC        = {'h100};
296             bins DMA_DST        = {'h104};
297             bins DMA_LEN        = {'h108};
298             bins DMA_CMD        = {'h10c};
299             bins DMA_STATUS     = {'h110};
300             ignore_bins RSVD2   = {['h114:$]};  // to maximum value
301         }
302         WDATA: coverpoint apb_if.pwdata {
303             bins ZERO           = {'h0};
304             bins ONE            = {'h1};
305             bins OTHERS         = {['h2: $]};   // to maximum value
306         }
307         WDATA2: coverpoint apb_if.pwdata[1:0] {
308             bins MOD4_ZERO      = {'h0};
309             bins MOD4_ONE       = {'h1};
310             bins MOD4_TWO       = {'h2};
311             bins MOD4_THREE     = {'h3};
312         }
313         SFR_RW: cross ADDR, DIR {
314             bins DMA_VER_RD = binsof(ADDR.DMA_VER) && binsof(DIR.READ);
315             ignore_bins DMA_VER_WR = binsof(ADDR.DMA_VER) && binsof(DIR.WRITE);
316             bins DMA_SRC_RD = binsof(ADDR.DMA_SRC) && binsof(DIR.READ);
317             bins DMA_SRC_WR = binsof(ADDR.DMA_SRC) && binsof(DIR.WRITE);
318             bins DMA_DST_RD = binsof(ADDR.DMA_DST) && binsof(DIR.READ);
319             bins DMA_DST_WR = binsof(ADDR.DMA_DST) && binsof(DIR.WRITE);
320             bins DMA_LEN_RD = binsof(ADDR.DMA_LEN) && binsof(DIR.READ);
321             bins DMA_LEN_WR = binsof(ADDR.DMA_LEN) && binsof(DIR.WRITE);
322             ignore_bins DMA_CMD_RD = binsof(ADDR.DMA_CMD) && binsof(DIR.READ);
323             bins DMA_CMD_WR = binsof(ADDR.DMA_CMD) && binsof(DIR.WRITE);
324             bins DMA_STATUS_RD = binsof(ADDR.DMA_STATUS) && binsof(DIR.READ);
325             ignore_bins DMA_STATUS_WR = binsof(ADDR.DMA_STATUS) && binsof(DIR.WRITE);
326         }
327     endgroup
```

295,1 95%

What to Sample: Cross Coverage

- Derived item from coverpoints
  - Check whether a combination of the cover points are hit
  - Can have bins

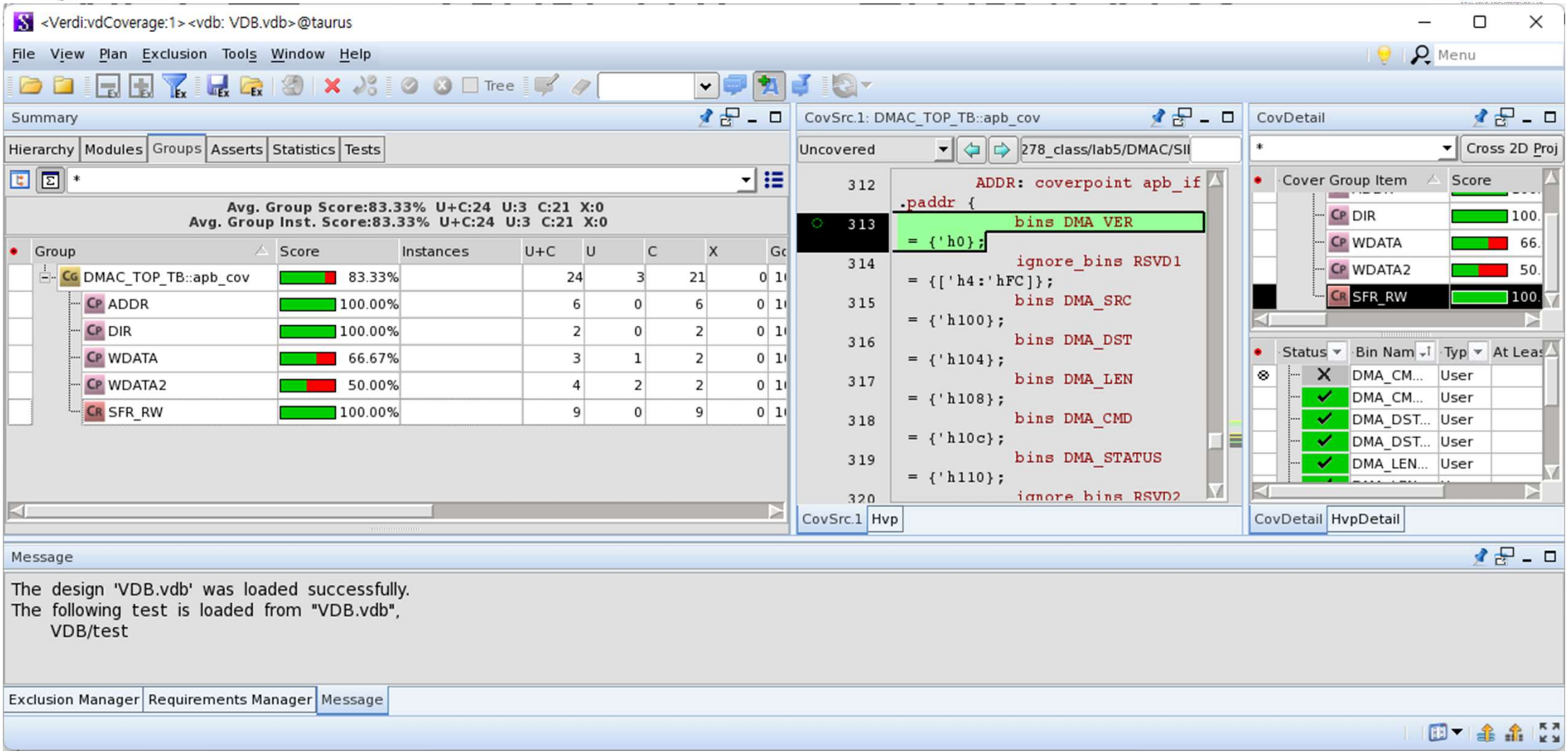
DMAC_TOP_TB.sv (~/ECE4278_class/lab5/DMAC/SIM/TB) - GVIM@taurus

File Edit Tools Syntax Buffers Window Help

```
286     // Functional coverage
287     covergroup apb_cov;
288         DIR: coverpoint apb_if.pwrite {
289             bins READ           = {0};
290             bins WRITE          = {1};
291         }
292         ADDR: coverpoint apb_if.paddr {
293             bins DMA_VER        = {'h0};
294             ignore_bins RSVD1   = {['h4:'hFC]};
295             bins DMA_SRC        = {'h100};
296             bins DMA_DST        = {'h104};
297             bins DMA_LEN        = {'h108};
298             bins DMA_CMD        = {'h10c};
299             bins DMA_STATUS     = {'h110};
300             ignore_bins RSVD2   = {['h114:$]};  // to maximum value
301         }
302         WDATA: coverpoint apb_if.pwdata {
303             bins ZERO           = {'h0};
304             bins ONE            = {'h1};
305             bins OTHERS         = {['h2: $]};   // to maximum value
306         }
307         WDATA2: coverpoint apb_if.pwdata[1:0] {
308             bins MOD4_ZERO      = {'h0};
309             bins MOD4_ONE       = {'h1};
310             bins MOD4_TWO       = {'h2};
311             bins MOD4_THREE     = {'h3};
312         }
313         SFR_RW: cross ADDR, DIR {
314             bins DMA_VER_RD = binsof(ADDR.DMA_VER) && binsof(DIR.READ);
315             ignore_bins DMA_VER_WR = binsof(ADDR.DMA_VER) && binsof(DIR.WRITE);
316             bins DMA_SRC_RD = binsof(ADDR.DMA_SRC) && binsof(DIR.READ);
317             bins DMA_SRC_WR = binsof(ADDR.DMA_SRC) && binsof(DIR.WRITE);
318             bins DMA_DST_RD = binsof(ADDR.DMA_DST) && binsof(DIR.READ);
319             bins DMA_DST_WR = binsof(ADDR.DMA_DST) && binsof(DIR.WRITE);
320             bins DMA_LEN_RD = binsof(ADDR.DMA_LEN) && binsof(DIR.READ);
321             bins DMA_LEN_WR = binsof(ADDR.DMA_LEN) && binsof(DIR.WRITE);
322             ignore_bins DMA_CMD_RD = binsof(ADDR.DMA_CMD) && binsof(DIR.READ);
323             bins DMA_CMD_WR = binsof(ADDR.DMA_CMD) && binsof(DIR.WRITE);
324             bins DMA_STATUS_RD = binsof(ADDR.DMA_STATUS) && binsof(DIR.READ);
325             ignore_bins DMA_STATUS_WR = binsof(ADDR.DMA_STATUS) && binsof(DIR.WRITE);
326         }
327     endgroup
```

295,1 95%

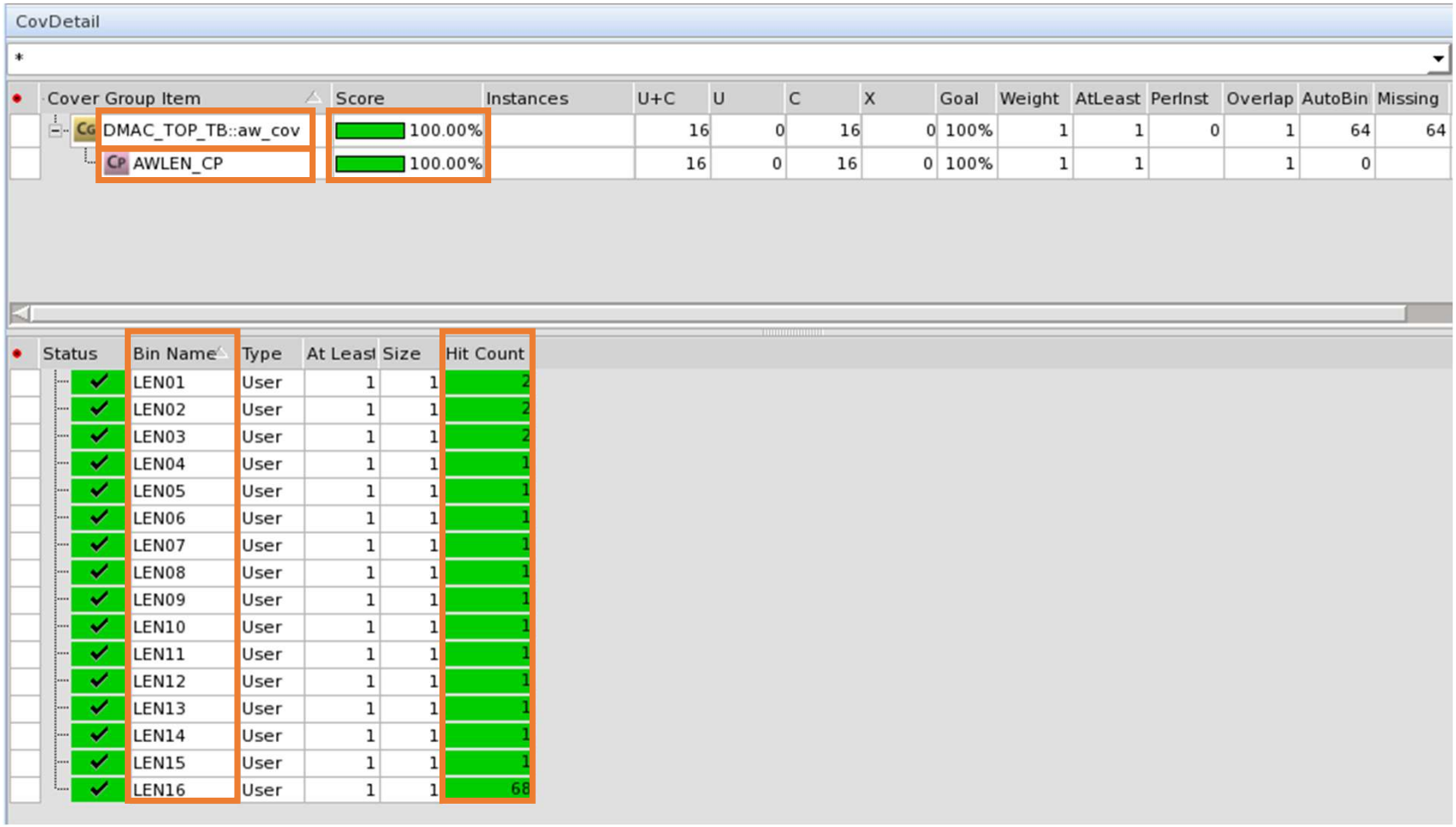
Functional Coverage in Verdi



Assignment: AWLEN Functional Coverage

- Goal:
  - Check whether our simulations cover all AWLEN values
    - AWLEN=0 → burst 1, AWLEN=1 → burst 2, ..., AWLEN=15 → burst 16
- How
  - Add coverage items to SIM/TB/DMAC_TOP_TB.sv
  - Steps
  1. Create a coverage group
  2. Create a coverpoint for AWLEN (label: AWLEN_CP)
  3. Create 16 bins (one for each AWLEN) for the coverpoint
     - Bins name : AWLEN=0 → LEN01, AWLEN=1 → LEN02, ..., AWLEN=15 → LEN16
  4. Instantiate the coverage group (covergroup name: aw_cov)
  5. Make the coverage group to be sampled at AW handshake
     - i.e., rst & AWVALID & AWREADY

Expected Output



Submission

- Upload DMAC_TOP_TB.sv on icampus
  - Do not change the file name
  - It is okay to have -1, -2, … at the end of the file name

- Due: Sun. 4/13, 23:59