Developer: Abhishek Basak, Graduate Student, Case Western Reserve University, Cleveland, OH, USA, funded in part by Semiconductor Research Corporation

And Dr. Swarup Bhunia, University of Florida, Gainesville, FL, USA

This directory provides a complete System-on-Chip (SoC) model written in Verilog, which includes description of all IP blocks. This example SoC design with the constituent IPs can be used to evaluate and benchmark various security architectures and policies.

Questions or suggestions can be sent to the authors to Abhishek Basak axb594@case.edu

The File/Module Hierarchy in the SoC RTL (In Verilog) Model:

test (topTESTBENCHSPIandPMC16.v)

- TOPmost (TOPmostSPIandPMC15.v)
  - FFT128wrapper (fft128wrapper3.v)
    - FFT128 (fft128.v) (After this, hierarchy of modules not defined)
      - BUFRAM128C_1 (bufram128c_1.v)
      - RAM2x128C (ram2x128.v)
      - RAM128 (ram128.v)
      - FFT8 (fft8_3.v)
      - CNORM_1 (cnorm_1.v)
      - ROTATOR128 (rotator128_v.v)
      - WROM128 (WROM128.v)
      - BUFRAM128C_2 (bufram128c_2.v)
      - FFT16 (fft16.v)
      - MPUC707 (mpuc707.v)
      - MPUC541 (mpuc541.v)
      - MPUC1307 (mpuc1307.v)
      - MPUC924_383 (mpuc924_383.v)
      - CNORM_2 (cnorm_2.v)
      - CNORM (cnorm.v)
      - BUFRAM128C (bufram128c.v)
      - debug_FFT (debug_FFT.v)
  - clkgen (clkgen.v)
  - together (together2Mwrapperextra4.v)
    - dlx (topM.v)
    - Icache (IcacheME.v)
    - SystemMem (DcacheME2.v)

debug_DLX1 (debug_DLX2.v)
debug_mem (debug_mem.v)
AEStopwrapper (AEStopwrapper2.v)
aes_encoder_decoder (AES_encoder_decoderF1.v)
debug_AES (debug_AES.v)
spi_top (spi_topMY1.v)
spi_clgen (spi_clgen.v)
spi_shift (spi_shift.v)
debug_SPI (debug_SPI.v)

PMC (power_management.v)
securitypolicy (securitypolicytop2.v)
IcacheSP (IcacheM.v)
dlx (topM.v)
Dcache (DcacheM2ALT.v)
DAP (DAP.v)

Files that are included as part of various modules are dlx.defines, FFT128_CONFIG.inc ,timescale.v and spi_defines.v

Example program that is used to test SPC functioning is test11BRANCH3.DAT (machine level)

Along same lines, some programs (at the machine level) that are used to see functioning of the DLX core include TOPprog.DAT and test11onlystoreandload.DAT

<u>The major Functional Modules or IP Blocks:</u>

128 point FFT engine

DLX RISC μP (5 stage pipeline and Load/Store architecture) & associated register based instruction memory

128 bit AES engine (encrypt + decrypt)

Serial Peripheral Interface (SPI) Control Module

Power Management Controller (Toy model – Just the Interface Signal Modelling)

SPC (Security Policy Controller) – implemented using the DLX μP core and register based dedicated instruction and data memory

System Data Memory (using registers) – Read/Write interface to above first 4 functional modules via appropriate memory mapping/segmentation

Important Points to Note:

The files fft128wrapper3.v, together2Mwrapperextra4.v, AEStopwrapper2.v all contain corresponding security wrappers of associated functional modules, boundary scan wrappers (IEEE P1500 which is verified via appropriate test benches) to all functional I/O pads and the debug trace macrocells, local to each module or subsystem like debug_DLX1, debug_mem, debug_FFT, debug_AES etc (along lines of ARM Coresight, but in much smaller scale).

All the debug modules are programmed through configuration register interface, controlled by the Debug Access Port (DAP). The SPC interaction with these trace modules of IPs for program/configuration, is through the address mapped debug bus via control at the DAP. All IPs are also interconnected wherever necessary (like IP to system data memory) via point-to-point links (need to extend it to NoC based approach). Each trace cell has interface with the corresponding IP core security wrapper through which it sends data/information to SPC when an event of interest is detected.

HOW TO EXEUTE THE MODEL:

Add all the files except for dlx.defines, FFT128_CONFIG.inc and the machine level simulation programs (test11..DAT files) to project and compile. Simulate test and run for 300 – 400 ns (300000 – 400000 ps in this case as there is a latent factor of 1000 here). The higher level program (test in this case) basically inputs to FFT engine, the DLX executes program based on the FFT o/p data, the AES encrypts DLX output data and sends through SPI module for external communication. All through the execution, the SPC is active, monitoring security controls such as SPCDIS and SPCREQ of various modules based on the wrapper supplied event lists.

Accordingly different overhead scenarios are estimated in DC_COMPILER in 32 nm tech library.

MPF, MTI and WLF files in the folder are project/simulation files here in ModelSIM (you can start another project separately or use the existing one depending on the tool being used for functional simulation) . Similarly work is the project folder for simulation here