Victor’s training

* Launch program from terminal
* AES low latency include:

+ cropto\_AES: AES\_Dshare: do encryption algorithm

+ control part: control, data inout from computer (key, plaintext, read cipher from crypto part), control datapath, and PRGN to generate random input for masking purpose

+ 2 part with 2 separate subprojects.

* Important signals

+ state: write, read (to and from crypto), idle…

+ coin\_reg : to choose for which group will be encrypt (1 or 2)

+ total\_count: the order of encryption (count down)

+ instruction: which type of plaintext and encryption we do (fix or random, mask-off or mask-on)

+ howmany: number of encryptions

+ check simu with: state, unshare in and out, coin\_reg, total\_count

* Synthesis Steps:

+ connect them together, after simulation => iSIM with window, should save for next time don’t need to add the list of signals anymore.

+ at .ucf file (care for FPGA pins, not physical pins)

+ after synthesis => map => generate bit file => open iMPACT to transfer it to FPGA (for separate 2 parts). Initiate and program

+planahead: important for adjust constrainst about area

+ notice in warning: serious warning affect implementation result

* Test:

+ use matlab file to control and test\_com (compare result => ok)

+ notice with type of input signals

+ reorder input bytes (column or row prior)

+ the number of RNG created??? Should be adapt?

+ file .c (encrypt AES 128) for what?

* Measurement

+ measurement: got traces and show on oscilloscope using matlab, notice on fastframe

+ notice to adjust scale and the number of points on screen

+ can capture for each encryption separately

+ each trace calculated following byte unit

**+ the number of traces = the number of encryptions**

+ traces got will be devided to groups (which encryptions)