Bridge of Life Education

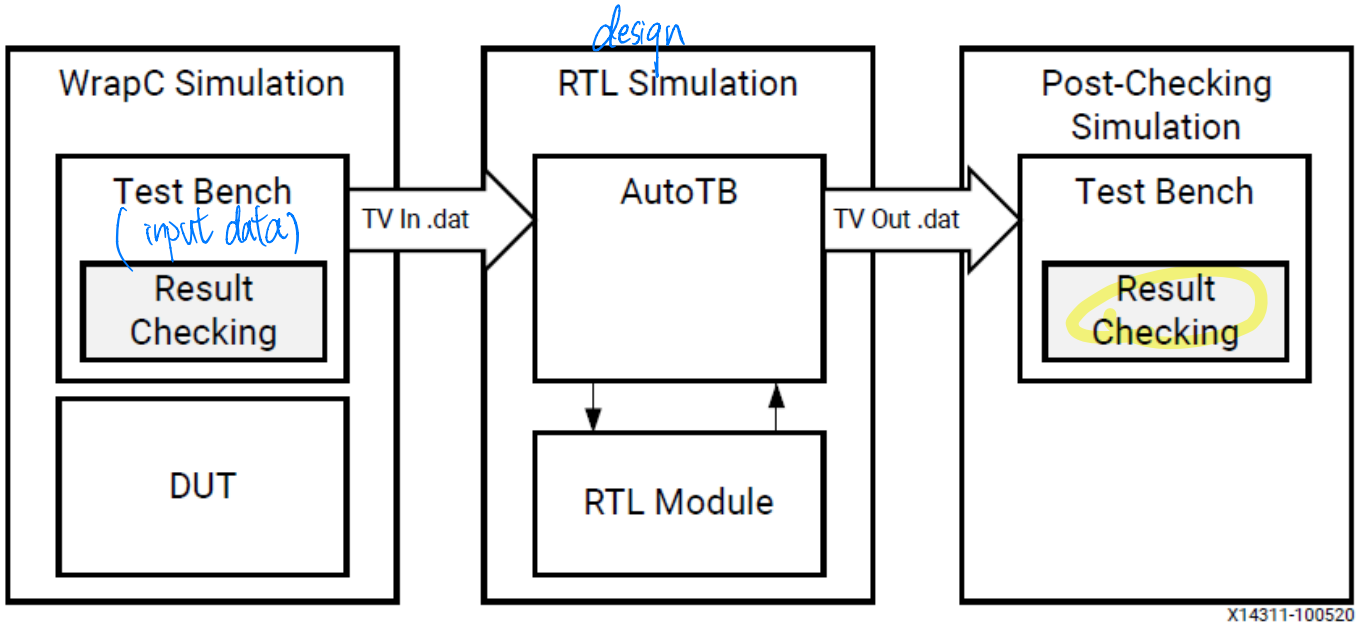
Guideline for HLS Cosim

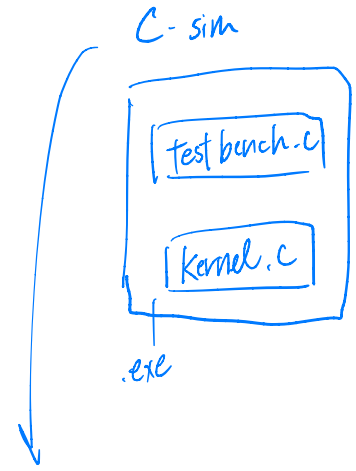
Guideline for Writing a Test Bench

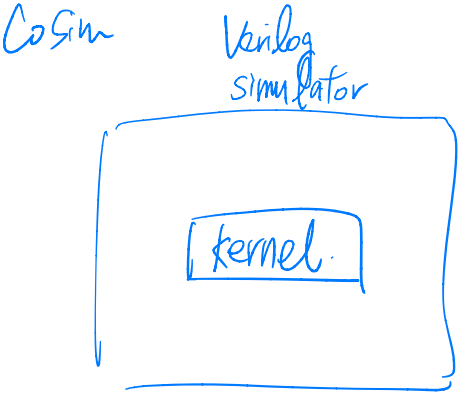
- Separate the top function for synthesis from the test bench and header files.
- If there are multiple kernel functions, group them in one top-level function for synthesis.
- The test bench should execute the top-level function for multiple transactions, allowing different data values to be applied and verified. (test coverage)
- You will use the same test bench for C/RTL Co-simulation. If you want to calculate II during the RTL simulation, you also need to provide multiple transactions to calculate II accurately.
- Checking correctness of kernel run is important. The return value of function main() indicates the following:
  - Zero: Results are correct.
  - Non-zero value: Results are incorrect. The non-zero results indicate the type of failure.

C/RTL Co-simulation

0. Different from Event-Driven Hardware Simulator
1. Phase#1: C simulation is executed to prepare the “input vectors” to the top-level function.
2. Phase#2: RTL simulation starts; it takes the input vectors and generates the “output vectors”
3. Phase#3: C simulation of the test bench main() function continues. It takes the “output vectors” returned from the RTL simulation and performs verification of the result.







Stream fifo depth ∞

↕

Co-sim 有問題 → buffer depth ↑

Requirement for Co-simulation

```
@E [SIM-345] Cosim only supports the following 'ap_ctrl_none' designs: (1)
combinational designs; (2) pipelined design with task interval of 1; (3) designs with
array streaming or hls_stream ports.
@E [SIM-4] *** C/RTL co-simulation finished: FAIL ***
```

- Interface Synthesis Requirement
  - Top-level function using ap_ctrl_chain or ap_ctrl_hs (can not use ap_ctrl_none)
  - Or design must be purely combinational (evaluate in one cycle)
  - Top function must have an initiation interval of 1. No handshake in the data argument. Input test vector are fed into RTL all at once
  - Interface must be all arrays streaming with axis or ap_hs
- **Provide sufficient FIFO depth to avoid stall or deadlock**
  - **Depth declared for the interface is too small. Create Deadlock**
- Certain unsupported Optimization for array or structs
  - Multiple transformation on array on the interface or array within structs
    - Vertical mapping, Reshape, Partition
  - Not support –
    - Horizontal mapping, Vertical mapping
    - Conditional access on the AXI4-Stream with register slice enabled
    - Mapping arrays to streams
- Use Waveform Viewer or Co-simulation deadlock viewer

- https://www.boledu.org/textbooks/hls-textbook/application-acceleration-development-flow/crtl-co-simulation#crtl-co-simlation
- https://docs.xilinx.com/r/en-US/ug1399-vitis-hls/Interface-Synthesis-Requirements

Provide Sufficient FIFO Depth to Prevent Co-sim Deadlock

```
#pragma HLS interface mode=m-axi port=<name> bundle=<string> \
register register_mode=<mode> depth=<int> offset=<string> latency=<value>\
clock=<string> name=<string> storage_type=<value>\
num_read_outstanding=<int> num_write_outstanding=<int> \
max_read_burst_length=<int> max_write_burst_length=<int>
```

```
#pragma HLS stream variable=<variable> type=<type> depth=<int>
```

- **depth: Specifies the maximum number of samples for the test bench to process.**