Project : Advanced Encryption Standard (AES)

GA(EE4415/4415E): Saurabh Jain, Lin Longyang, Trinh Quang Kien, Su Hanyang
Lecturer(s): Prof. Massimo Alioto, Prof. Xu Yong Ping

National University of Singapore (NUS)
ECE Department
Green IC group

Coverage of Presentation

- Introduction
- Description
- Implementation (RTL)
- Interface with Testbench
- Details at Testbench Level
- Signal Timings
- Synthesis Constraints
- Marks distribution
- Quick Start

Coverage of Presentation

- Introduction
- Description
- Implementation (RTL)
- Interface with Testbench
- Details at Testbench Level
- Signal Timings
- Synthesis Constraints
- Marks distribution
- Quick Start

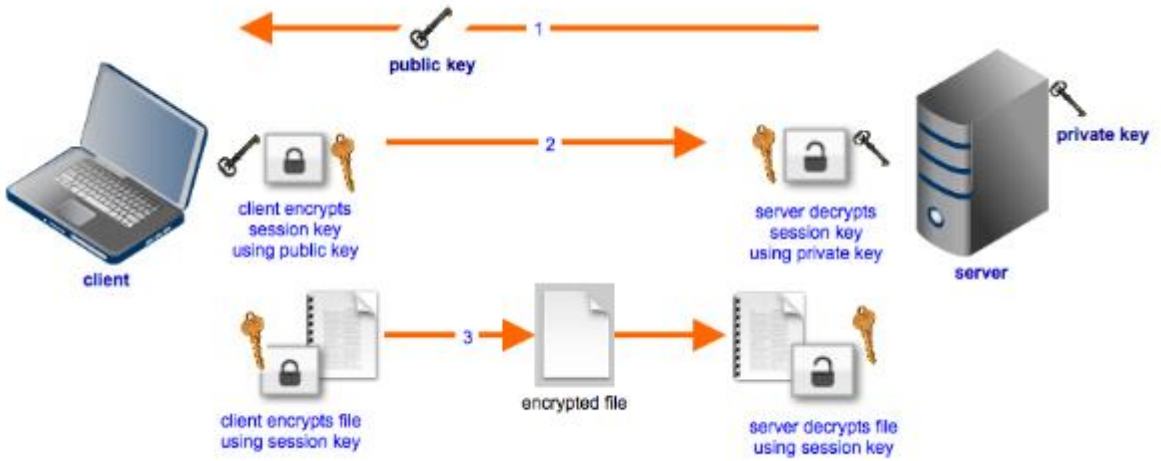
Introduction

What is AES

- AES: ***A***dvanced ***E***ncryption ***S***tandard
  - First introduced in 2001 for encrypting electronic data
  - Introduced by US. National Institute of Standard and Tech. (NIST)
  - Block length: 128 bits, key length: 128/192/256
  - Symmetric key algorithm (i.e. encryption and decryption uses same key)
  - First publically available cipher
  - AES also called “Rijndael”

Where is AES Used

- Secure file transfers protocols FTPS/ HTTPS/ SFTP etc.



- Read more:

  https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Coverage of Presentation

- Introduction
- Description
- Implementation (RTL)
- Interface with Testbench
- Details at Testbench Level
- Signal Timings
- Synthesis Constraints
- Marks distribution
- Quick Start

Description

Overall Flow

- http://poincare.matf.bg.ac.rs/~ezivkovm/nastava/rijndael_animacija.swf

Coverage of Presentation

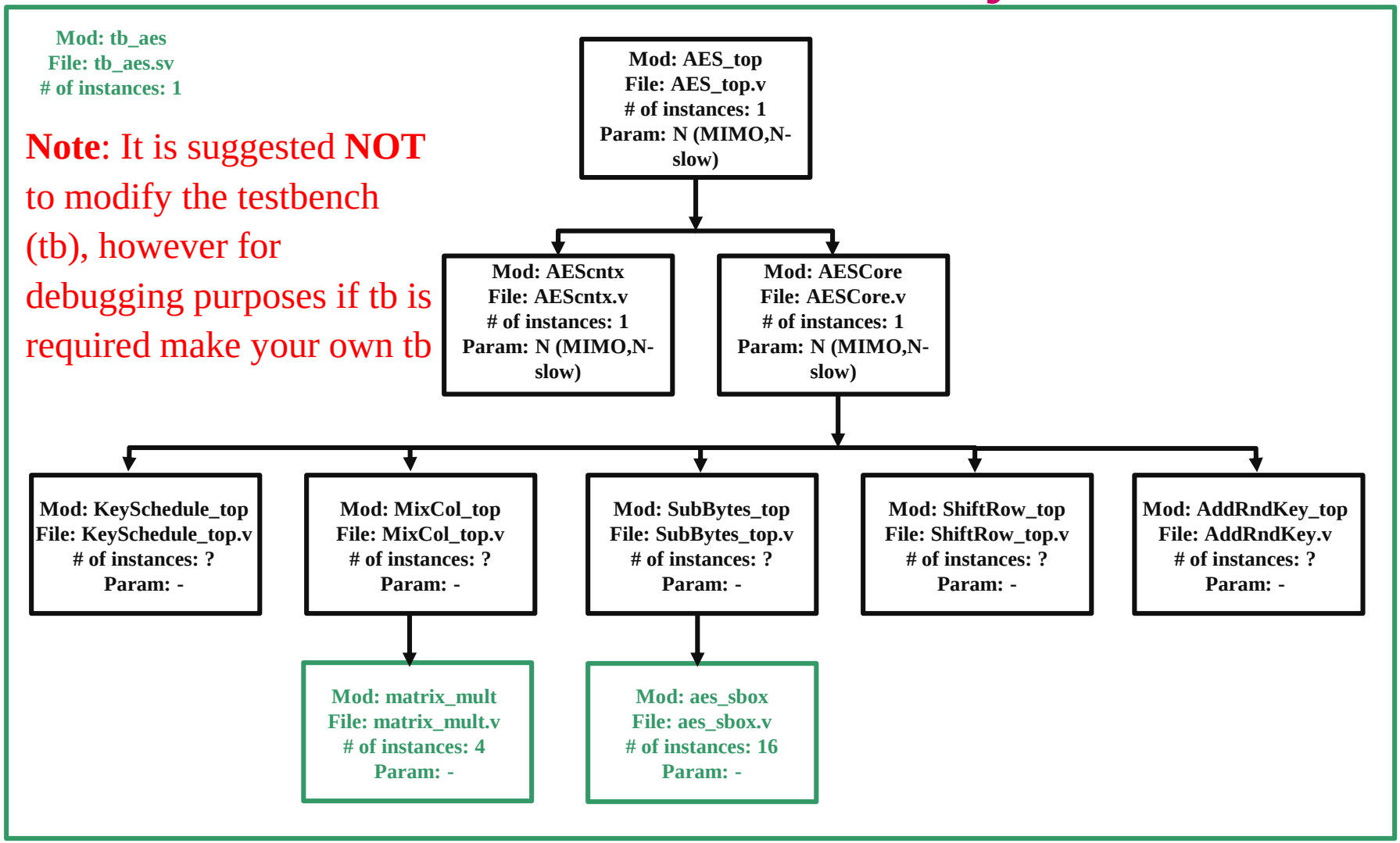
- Introduction
- Description
- Implementation (RTL)
- Interface with Testbench
- Details at Testbench Level
- Signal Timings
- Synthesis Constraints
- Marks distribution
- Quick Start

Implementation (RTL)

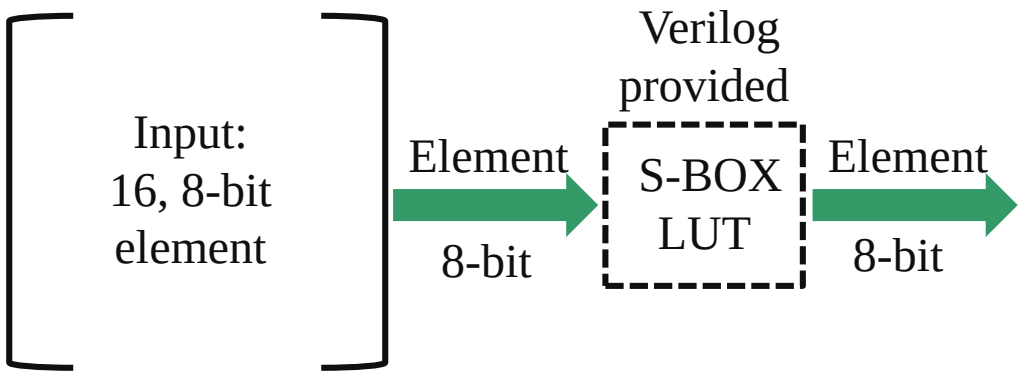
Problem Statement

- RTL implementation of AES with following specifications:
  - 128 bit key
  - 128 bit plain text
  - 128 bit cipher text
- Variants:
  - SISO: Single Input (key & plain text) Single Output (cipher text)
  - MIMO (N): Multiple Input (key & plain text) Multiple Output (cipher text)
  - N-slowing: N time multiplexed I/O streams

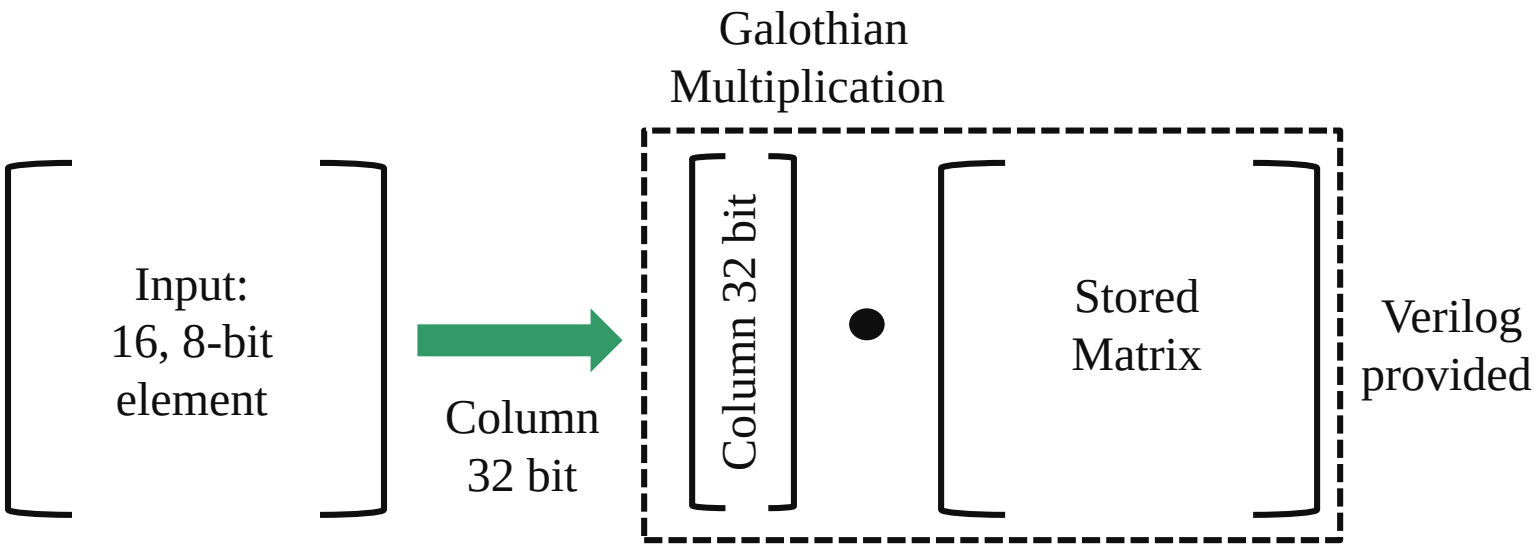
Block Level Hierarchy



Provided Resources: Sub-Bytes



Provided Resources: Mix Column

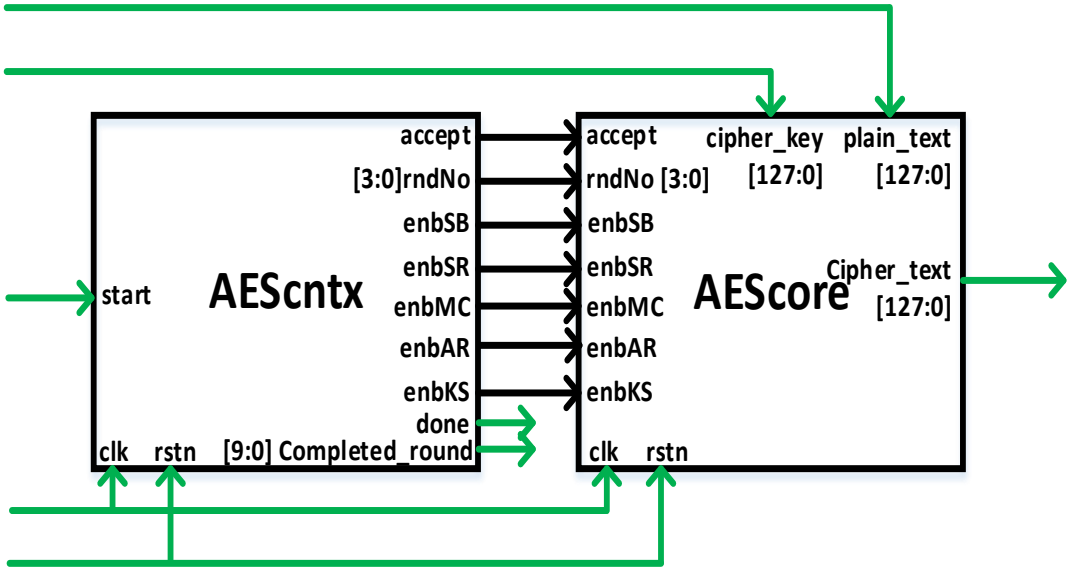


Coverage of Presentation

- Introduction
- Description
- Implementation (RTL)
- Interface with Testbench
- Details at Testbench Level
- Signal Timings
- Synthesis Constraints
- Marks distribution
- Quick Start

Interface with Testbench

Interface between Core, Controller & TB



Accept: Notifies the core to accept input from testbench
rndNo: Inform the core about the ongoing round number

enbSB: Inform the core when to enable SUB-BYTES block

enbSR: Inform the core when to enable SHIFT-ROWS block

enbMC: Inform the core when to enable MIX-COULUMN block

enbAR: Inform the core when to enable ADD-ROUND-KEY block

enbKS: Inform the core when to enable KEY-SCHEDULING block

done: Notifies testbench that AES of 1 input completed
completed_round: Notifies testbench the completion status of a given round
cipher_key: incoming key from testbench
plain_text: incoming data from testbench
cipher_text: encrypted data

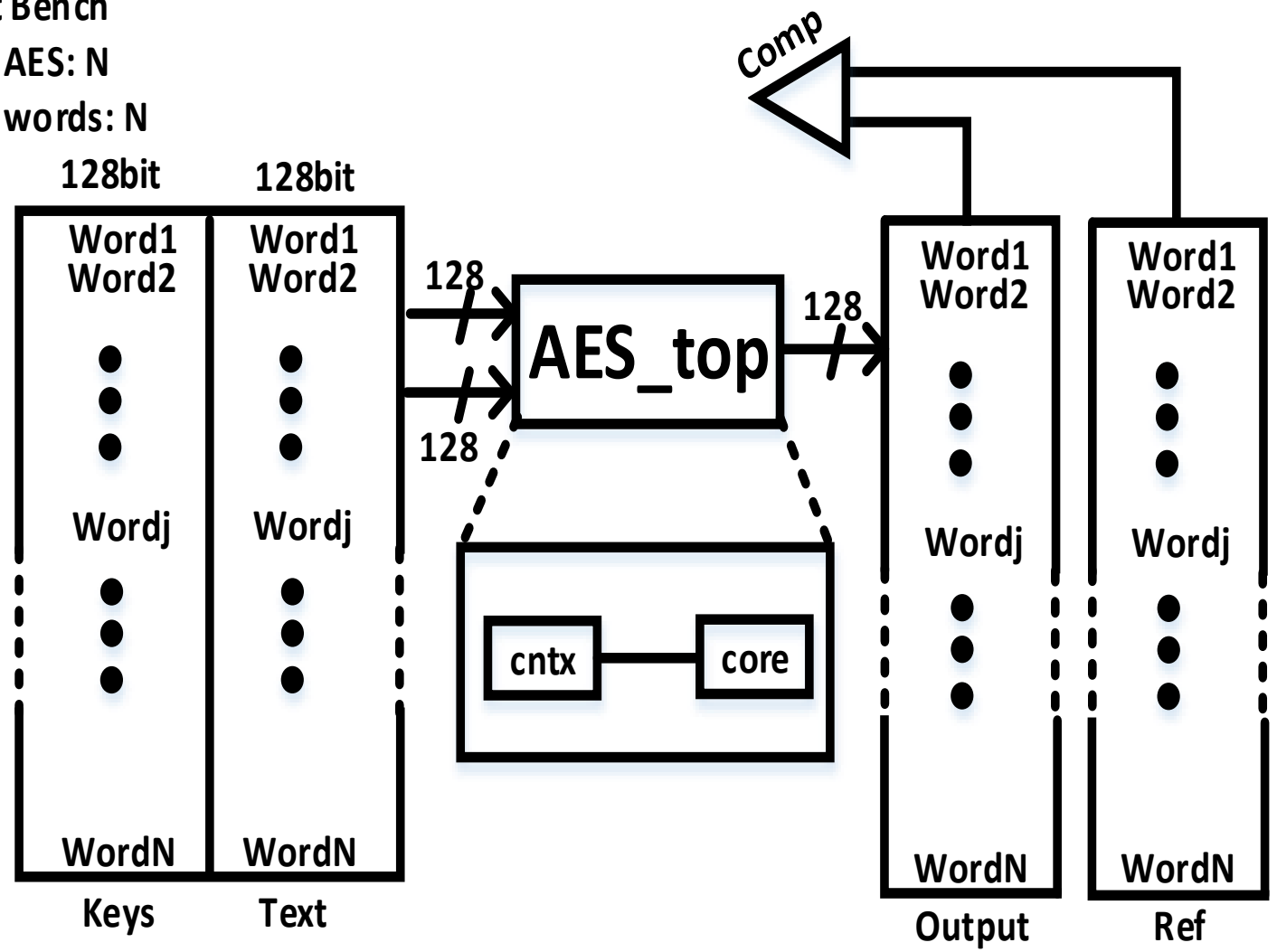
Coverage of Presentation

- Introduction
- Description
- Implementation (RTL)
- Interface with Testbench
- Details at Testbench Level
- Signal Timings
- Synthesis Constraints
- Marks distribution
- Quick Start

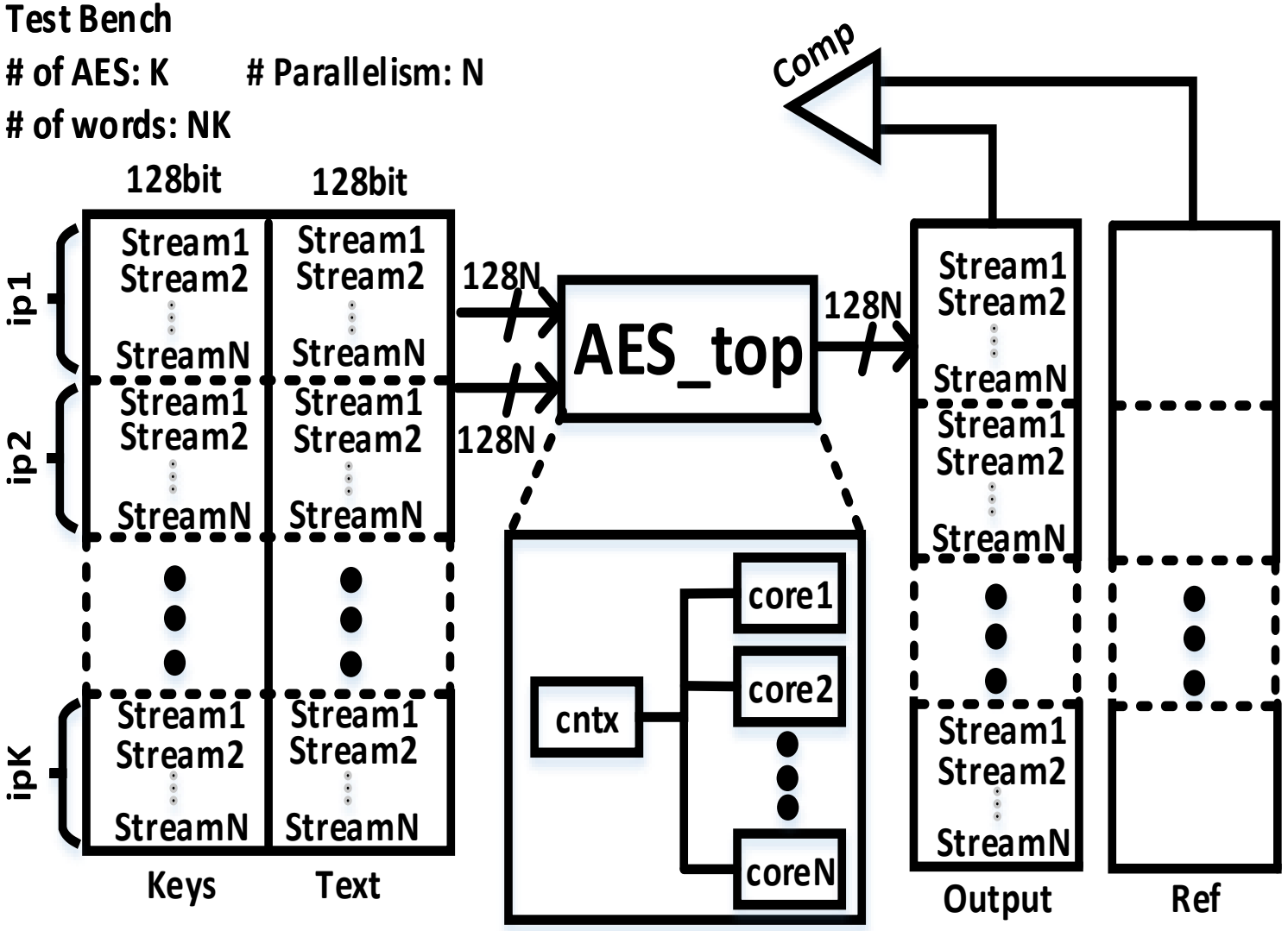
Details at Testbench Level

SISO

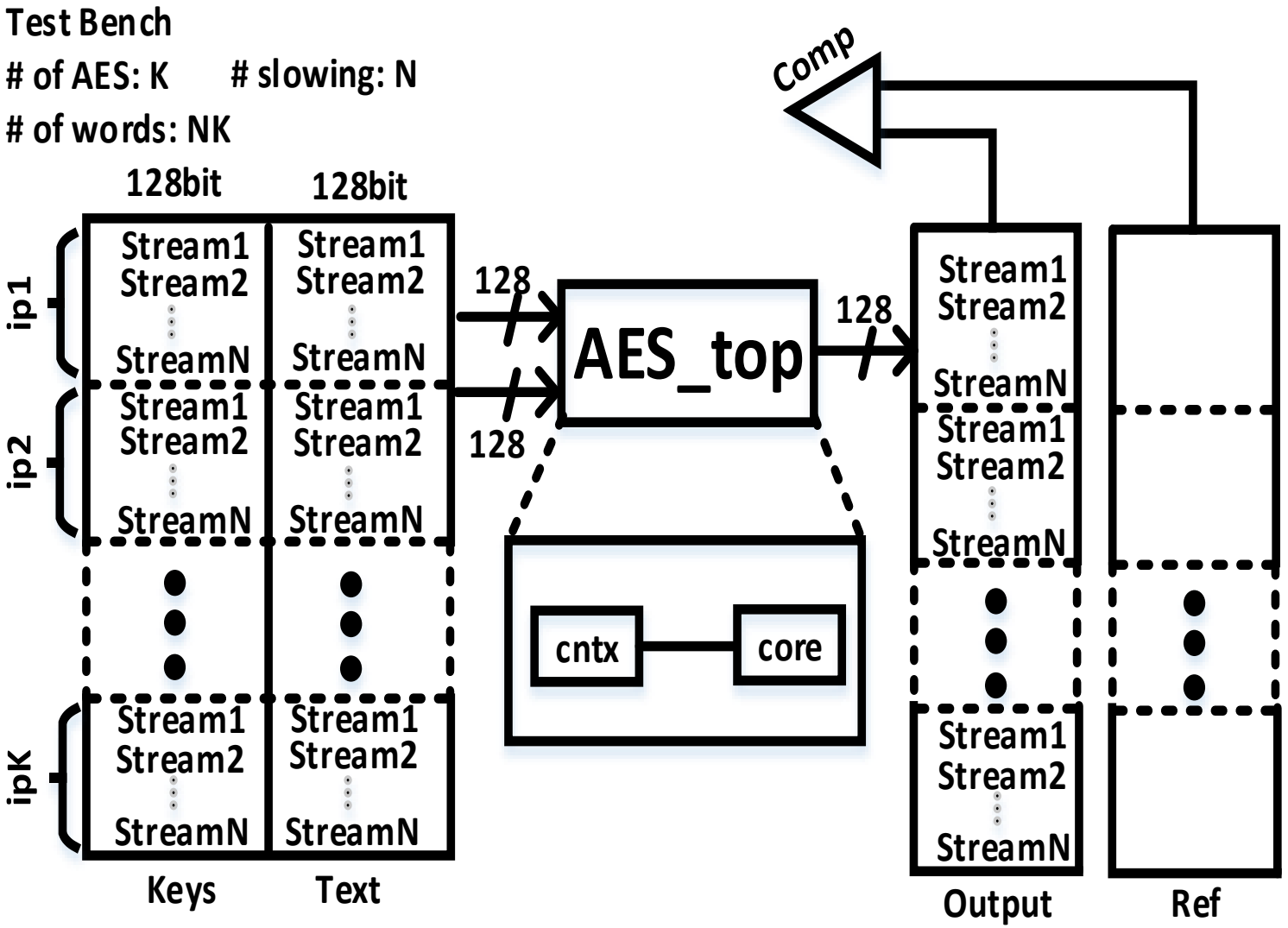
Test Bench
# of AES: N
# of words: N



MIMO


Test Bench
# of AES: K
Parallelism: N
of words: NK
128bit
128bit
ip1
ip2
ipK
Stream1
Stream2
StreamN
Keys
Text
128N
128N
AES_top
cntx
core1
core2
coreN
Comp
Output
Ref

N-Slowing

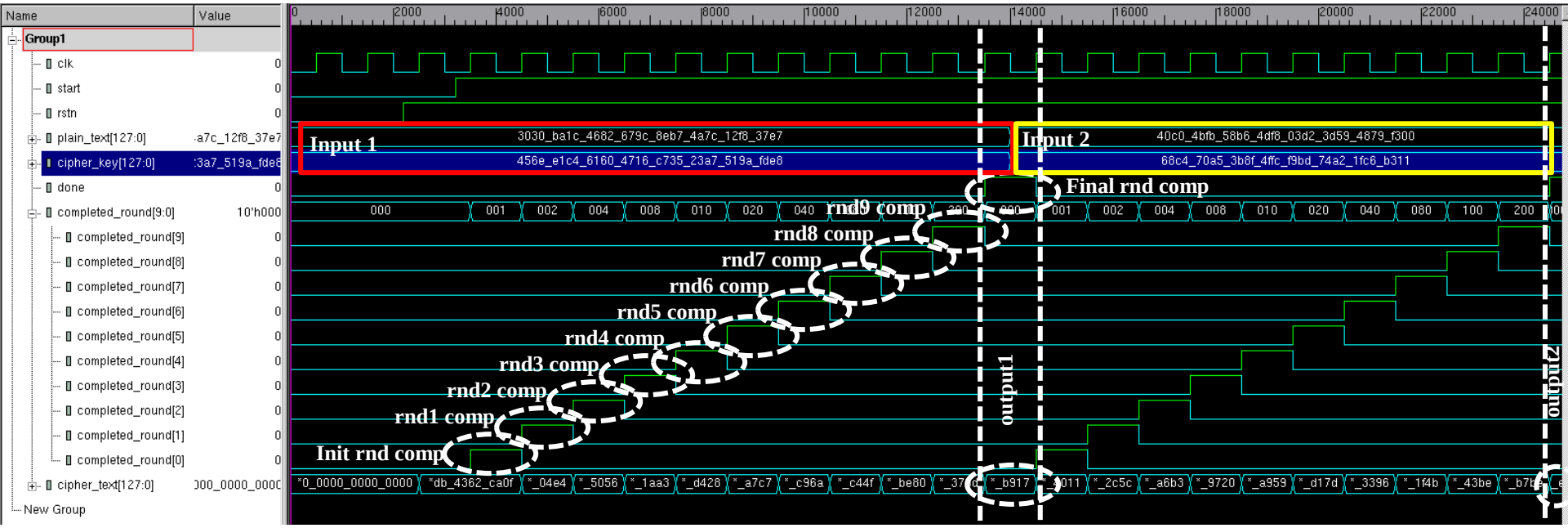

Test Bench
# of AES: K
slowing: N
of words: NK
128bit
128bit
ip1
ip2
ipK
Stream1
Stream2
StreamN
Keys
Text
128
128
AES_top
cntx
core
Comp
Output
Ref

Coverage of Presentation

- Introduction
- Description
- Implementation (RTL)
- Interface with Testbench
- Details at Testbench Level
- Signal Timings
- Synthesis Constraints
- Marks distribution
- Quick Start

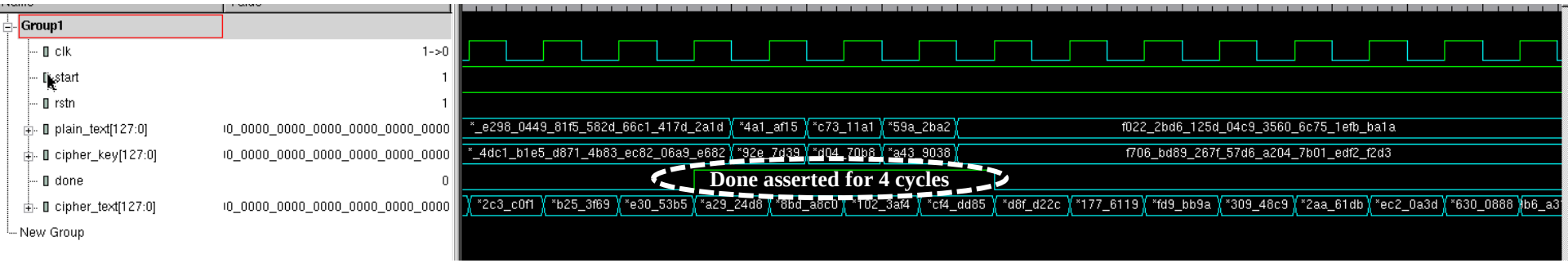
Signal Timings

Timing of signals (SISO and MIMO)



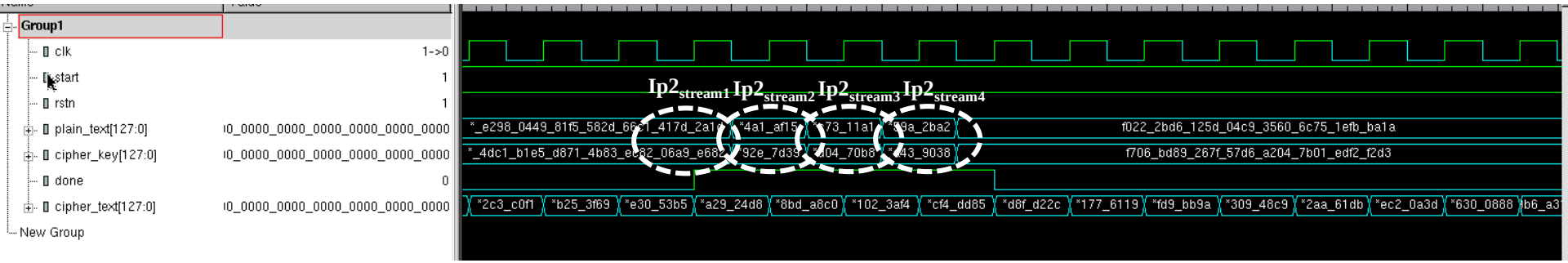
Timing of signals (N-Slowing)

- N=4



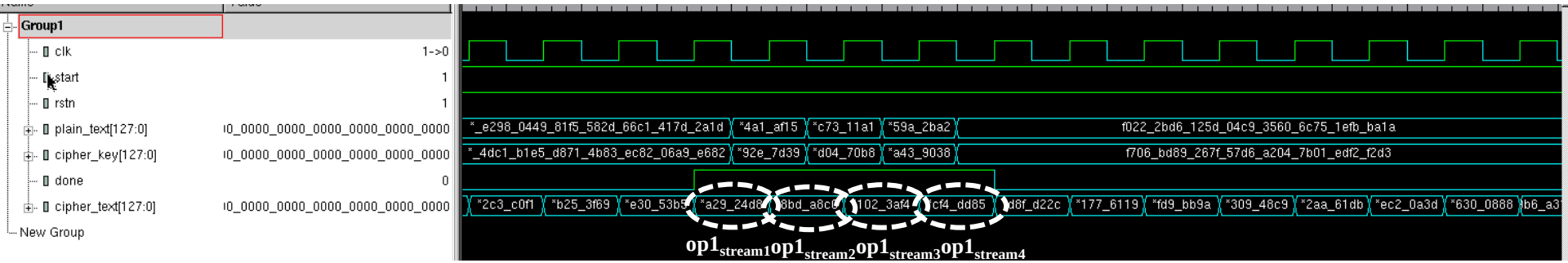
Timing of signals (N-Slowing)

- N=4



Timing of signals (N-Slowing)

- N=4



Coverage of Presentation

- Introduction
- Description
- Implementation (RTL)
- Interface with Testbench
- Details at Testbench Level
- Signal Timings
- Synthesis Constraints
- Marks distribution
- Quick Start

Synthesis Constraints

- SISO:
  - Clock period of 2 ns
  - Input delay (exclude clock) of 0.2 ns
  - Output delay of 0.2 ns
  - Load capacitance (all outputs) of 5fF
- MIMO (N=4):
  - Clock period of 2 ns
  - Input delay (exclude clock) of 0.2 ns
  - Output delay of 0.2 ns
  - Load capacitance (all outputs) of 5fF
- N-slowing (N=4):
  - Clock period of 0.5 ns
  - Input delay (exclude clock) of 0.1 ns
  - Output delay of 0.1 ns
  - Load capacitance (all outputs) of 5fF

Note: All constraint should be written in "constraint.tcl" file of respective parts

Coverage of Presentation

- Introduction
- Description
- Implementation (RTL)
- Signal Timings
- Interface with Testbench
- Details at Testbench Level
- Synthesis Constraints
- Marks distribution
- Quick Start

Marks Distribution

Submissions

- SISO (Single Input Single Output) (10%)
- MIMO (Multiple Input Multiple Output) (5%)
- N-slowing (10%)
- Report (total 5%)

SISO (part I)

- "src" directory:
  - Compiled *.v files (7.5%)
  - Comments in the code
- "syn" directory:
  - "constraint.tcl" (else 2.5% won't be awarded)
  - "synthesis.tcl" (2.5%):
    - Commands to analyze all *.v files in "src" folder (0.5%)
    - Command to elaborate "AES_top" (0.5%)
    - Command to source "constraint.tcl"
    - Command to compile (or synthesize) "AES_top" (0.5%)
    - Command to report timing (0.5%)
    - Command to report area (0.5%)

MIMO (part II)

- "src" directory:
  - Compiled *.v files (3.5%)
  - Comments in the code

NOTE : Your code will be tested for arbitrary value of N

- "syn" directory:
  - "constraint.tcl" (else 1.5% won't be awarded)
  - "synthesis.tcl" (1.5%):
    - Commands to analyze all *.v files in "src" folder (0.3%)
    - Command to elaborate "AES_top" (0.3%)
    - Command to source "constraint.tcl"
    - Command to compile (or synthesize) "AES_top" (0.3%)
    - Command to report timing (0.3%)
    - Command to report area (0.3%)

NOTE : N = 4 for synthesis

N-Slowing (part III)

- "src" directory:
  - Compiled *.v files (7%)
  - Comments in the code

NOTE : Your code will be tested for arbitrary value of N

- "syn" directory:
  - "constraint.tcl" (else 3% won't be awarded)
  - "synthesis.tcl" (3%):
    - Commands to analyze all *.v files in "src" folder (0.5%)
    - Command to elaborate "AES_top" (0.5%)
    - Command to source "constraint.tcl" file
    - Command to compile (or synthesize) "AES_top" (0.5%)
    - Command to retime the design (0.5%)
    - Command to report timing (0.5%)
    - Command to report area (0.5%)

NOTE : N = 4 for synthesis

Report (part IV)

- Architecture and indicate following (2%):
  - Interconnection between blocks (i.e. Subbytes, Shift-Rows, Mix-columns, Add-round-key and key-scheduling) (1%)
  - Registers in your design (1%)
- Area (um^2), critical path delay (ns) and calculated maximum throughput (MS/s) for following (2%):
  - SISO (0.5%)
  - MIMO, N=4 (0.5%)
  - 4-slowing (1%)
- Observation and discussion (1%)
  - Which (0.5%) strategy is best (for throughput) and why (0.5%)

Throughput Calculation

- X_{max} (MS/s) = $Freq_{max}$ (MHz)x$\left(\frac{\#\ of\ \boldsymbol{valid}\ samples}{\#\ of\ cycles}\right)$
- Hints for completing project!!
  - Part 2: If part 1 completed part 2 doesn't require any effort
  - Part 3: For N-slowing **every** register in your design has to be replaced by N cascaded registers

Coverage of Presentation

- Introduction
- Description
- Implementation (RTL)
- Signal Timings
- Interface with Testbench
- Details at Testbench Level
- Synthesis Constraints
- Marks distribution
- Quick Start

Quick Start

Quick Start

- Make a new directory in your home

mkdir ~/<new_dir_name>

- Copy src to your new directory

cp –r /app11/lab-session/ee4415_part2/* ~/<new_dir_name>/.

- To Compile your source:

cd ~/<new_dir_name>/<part>/vcs

make compile

- To run your code on terminal

make sim **or** ./simv

- To run/debug your code in gui mode

./simv –gui

- Before starting synthesis (in design compiler)

cd ~/<new_dir_name>/<part>/syn/

cp /app11/lab-session/ee4415_part1/.synopsys_dc.setup ~/<new_dir_name>/<part>/syn/.

Note: carry out synthesis of each part in “syn” folder of respective parts

THANK YOU!
Questions?